ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 11, November 2015

A New Method for TDM Calculation in Speculative Multipliers

Amrutha Balan, Hima Sara Jacob

***Abstract*— Integer multiplication is the basic building block of digital designs. This paper presents high speed integer multiplication based on speculation, a technique which performs faster-but gives wrong result only in the rare case of error and shifting to an error correction circuit. The proposed speculative multiplier uses new method for TDM carry-save tree calculation and the partial products tree reduction can be done using three steps: partial products recoding, partial products partitioning and speculative compression. For speculative compression uses *(m :2)* speculative counters, with *m >3*, which are faster than conventional counters using full-adders and half-adders and a correction block is needed for each speculative counter if *m <3*. In speculative tree, final carry-save addition is done using a fast speculative adder. The speculative multiplier is simulated using Xilinx 13.2 and synthesis report shows that the proposed speculative multiplier is faster than the conventional speculative multiplier and also consumes less power.**

***Index Terms*— Digital arithmetic, multiplication, speculative functional units, speculative multipliers.**

I. INTRODUCTION

Variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together. Integer multiplication is the fundamental building block of digital systems.

Fast multiplier circuits can be obtained by using speculative method [1]. Speculative circuits have high speed but only in rare case of error shifting to a multi-cycle error correction circuit. Fast non-booth algorithms use logarithmic methods such as Wallace, Dadda or Three Dimensional Method TDM [2]. They all use carry-save compression tree, which consists of full-adders and half-adders to convert multi-operand sum in to two operand addition completed by a, fast carry-propagate adder. Speculative method recently applied to addition [3]. The calculation of speculative adder takes two cycles. In the first cycle adder computes the sum and an error flag while in the second cycle corrects the speculative result.

In this paper presents a method to design high speed speculative multiplier. The proposed speculative multiplier introduces a new approach for TDM carry-save tree not calculation and uses three steps for carry-save tree reduction such as partial products recoding, partial products partitioning and speculative compression. For speculative compression uses speculative *(m :2)* counters, with *m >3*. Speculative counters are faster than conventional counters. The TDM carry-save tree is completed with a fast speculative adder and an error correction circuit.

Manuscript received Aug 15, 2012.

***Amrutha Balan**, ECE, Mangalm College of Engineering, Ettumanoor , Kottayam.*

***Hima Sara Jacob** ECE, Mangalm College of Engineering, Ettumanoor , Kottayam.*

II. STEPS FOR SPECULATIVE CARRY-SAVE TREE REDUCTION

The partial products matrix (PPM) for a 16×16 multiplier is shown in Fig.1. The rightmost and leftmost columns of the PPM consist of small number of partial products, but more number of partial products in inner columns. The calculation of delay of the circuit is according to the height of the PPM: higher the matrix, higher the delay. By deleting some partial products from the inner columns of the PPM speculative carry-save tree reduction can be obtained. Although this would leads to misprediction. Therefore uses three steps for partial products reduction: partial products recoding, partial products partitioning and speculative compression.

A. Partial Products Recoding

Let a_ib_j and a_jb_i are the two partial products of the i + j-th column of the PPM and also introduce modified partial products shown below:

$$A_{i,j} = a_ib_j \ AND \ a_jb_i$$
$$O_{i,j} = a_ib_j \ OR \ a_jb_i \qquad (1)$$

As from the above equations: $A_{i,j} + O_{i,j} = a_ib_j + a_jb_i$. Thus, substitute modified partial products $A_{i,j}$ and $O_{i,j}$ to the partial products which belongs to the inner column of the PPM.

B. Partial Products Partitioning

Fig.1. (b) shows the partial products after recoding. Partial products which have higher delay are recoded.

C. Speculative Compression

For achieving reduction in carry-save tree simple deletion of $A_{i,j}$ terms would give large misprediction. Thus, uses speculative counters for compressing $A_{i,j}$ terms. An *(m :2)* speculative counters have *m* inputs $(x_0, x_1, \ldots\ldots, x_{m-1})$ and two outputs: sum *S* and carry *C*. The function of speculative counter is to count number of input bit that are "1" and encode the result on *S* and *C*.

$$2C + S = x_0 + \ldots\ldots + x_{m-1} \text{ for: } x_0 + \ldots\ldots + x_{m-1} \ m \leq 3 \qquad (2)$$

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 11, November 2015

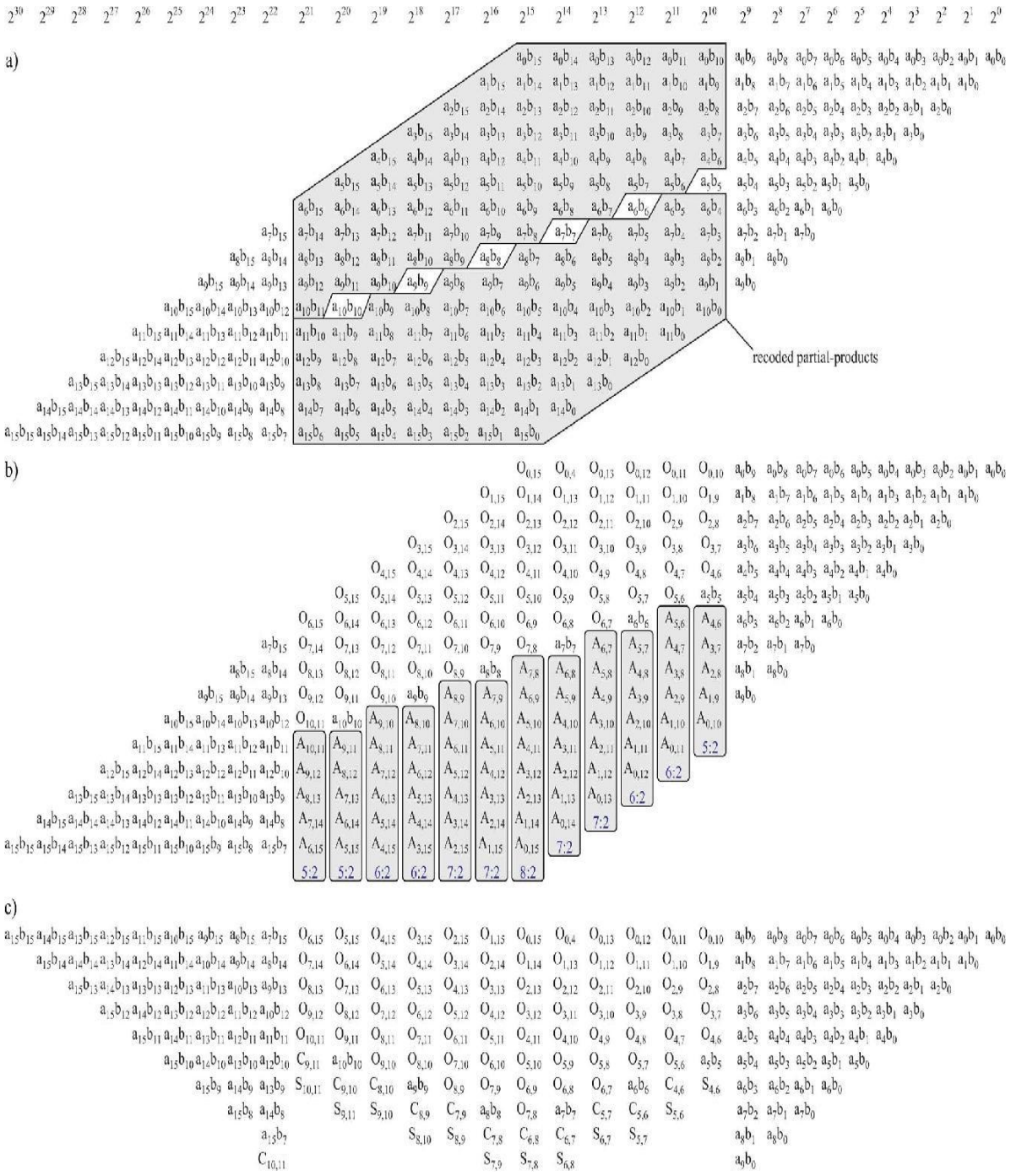

Fig.1. 16 × 16 bit Multiplier partial products matrixes: (a) initial partial products matrix; (b) partial products matrix after recoding; (c) partial products matrix after speculative compression. [1]

For $m > 3$, the sum $(x_0, x_1, \ldots\ldots, x_{m-1})$ could not represent with the S and C signals for all input configurations. Speculative counter gives error result if more than three inputs are high and it must be corrected in the next cycle. The PPM uses three (5:2), four (6:2), four (7:2) and one (8:2) speculative counters for reduction and is shown in Fig. 1(c).

III. SPECULATIVE MULTIPLIER DESIGN

Fig.2. sows the design of a speculative multiplier. Let A and B are the two inputs of the multiplier. Initially the inputs are processed by the partial products generation and recoding block. This block generates all partial products and recodes partial products which belong to the inner columns of the PPM producing $A_{i,j}$ and $O_{i,j}$ terms.

For further reduction in the PPM the $A_{i,j}$ terms are given to speculative counters which generates $S_{i,j}$ and $C_{i,j}$ terms. The counter outputs $S_{i,j}$ and $C_{i,j}$, the un-recoded $a_i b_j$ and the recoded $O_{i,j}$ terms are added together by using TDM carry-save tree[2]. The two outputs of the TDM is computed by speculative adder [3] resulting speculative result *Ys*.

If the number of inputs is more than three an error can be generate for each speculative counter. The correction block accepts same inputs of corresponding speculative compressor and produces two outputs: an error flag *E* and an error correction word *EW*. The error flag E is high if four or more

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 11, November 2015

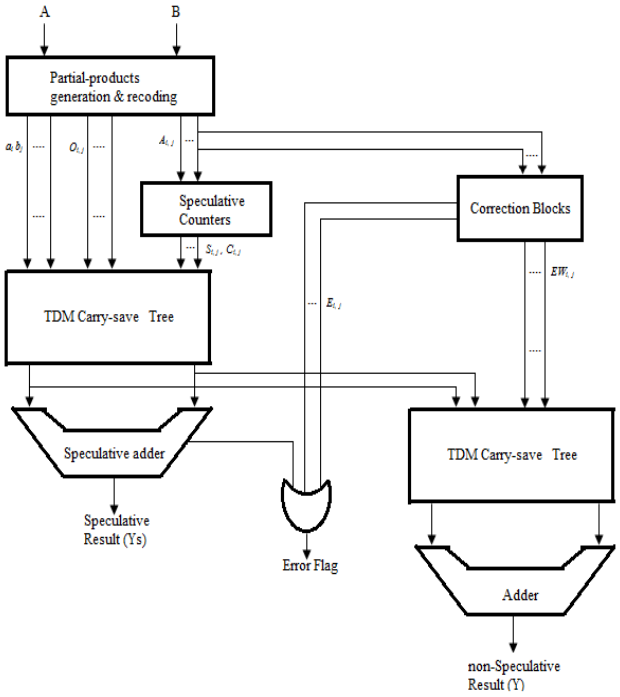

Fig. 2. Design of Speculative Multiplier [1]

inputs of the speculative compressor are "1". Therefore, for obtaining correct result add correction word *EW* to the speculative compressor output.

For computing error flag of the speculative multiplier, all error flags produced by each correction block and error flag of the speculative adder are OR-ed together. By adding the correction words $EW_{i,j}$ and the outputs of the TDM carry-save tree involved in the speculative part of the multiplier gives the non-speculative result *(Y).*

A. *Speculative Counter*

An *(m :2)* speculative counter is a component which have *m* inputs $(x_0, x_1, \ldots\ldots, x_{m-1})$ and two outputs: sum *S* and carry *C*. the *S* output can be easily calculated as a tree of XOR gates of the inputs. The estimation of the *C* signal has some difficulties. Let $f \geq 2\ (x_0, x_1, \ldots\ldots, x_{m-1})$ be the binary function and gives the *C* output of the *(m :2)* speculative counter.

For two, three and four inputs the function $f \geq 2$ can be simply calculated. In case of two inputs:

$$f \geq 2\ (x_0, x_1) = x_0 \,.\, x_1 \qquad (3)$$

The function $f \geq 2$ given to three inputs similar to the carry function of a full-adder:

$$f \geq 2\ (x_0, x_1, x_2) = x_0 \,.\, x_1 + x_0 \,.\, x_2 + x_1 \,.\, x_2 \qquad (4)$$

The function $f \geq 2$ applied to four inputs can be calculated as:

$$f \geq 2(x_0, x_1, x_2, x_3) = x_0 \,.\, x_1 + x_2 \,.\, x_3 + (x_0 + x_1) \,.\, (x_2 + x_3) \qquad (5)$$

For $m \geq 5$ use of the function $f \geq 2$ would lead to a large area and delay. To avoid this problem, divide-and-conquer method is used to compute the function $f \geq 2(x_0, x_1, \ldots. \ldots, x_{m-1})$ for $m \geq 5$. Consider the case of $m = 5$. Here the five inputs are grouped in to two subsets: the first subset consists of x_0, x_1, x_2, the second subset includes x_3, x_4 and is shown below:

$$f \geq 2(x_0, x_1, x_2, x_3, x_4) = f \geq 2(x_0, x_1, x_2) + f \geq 2(x_3, x_4) + f \geq 2(x_0 + x_1 + x_2 + x_3 + x_4) \qquad (6)$$

The implementation of the last equation is shown in Fig.3. The "modified full-adder" in the circuit calculates the carry output as $f \geq 2\ (x_0, x_1, x_2)$ and the sum output as the OR between x_3 and x_4.

The (6:2) and (7:2) can be implement by partitioning the inputs in two subsets of three bits or in two subsets of three and four bits respectively. The implementation of the (8:2) counter can be done either in two ways: partitioning the inputs in two subsets of four elements or by partitioning the inputs in three subsets, which includes three, three and two inputs each. It can be computed as:

$f \geq 2(x_0, \ldots\ldots\ldots, x_7)$

$$= f \geq 2(x_0, x_1, x_2) + f \geq 2(x_3, x_4, x_5) + f \geq 2(x_6, x_7) + f \geq 2(x_0 + x_1 + x_2,\ x_3 + x_4 + x_5,\ x_6 + x_7) \qquad (7)$$

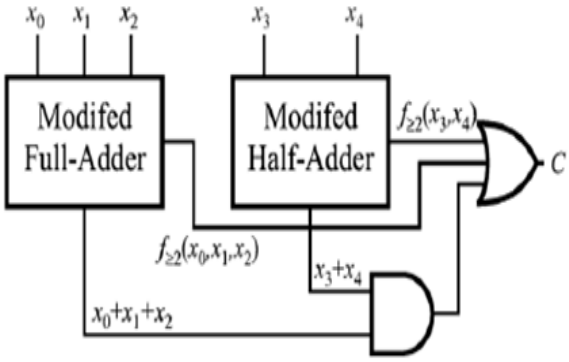

Fig. 3. Implementation of the carry output *(C)* of the (5:2) speculative com-pressor. [1]

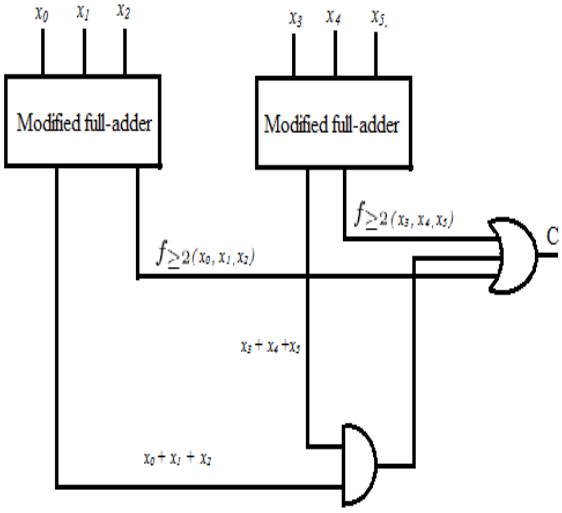

Fig. 4. Implementation of carry output (C) of (6:2) speculative compressor

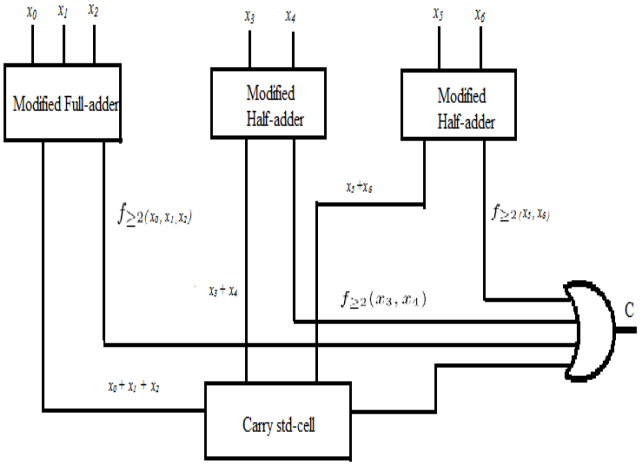

Fig 5. Implementation of carry output (C) of (7:2) speculative compressor

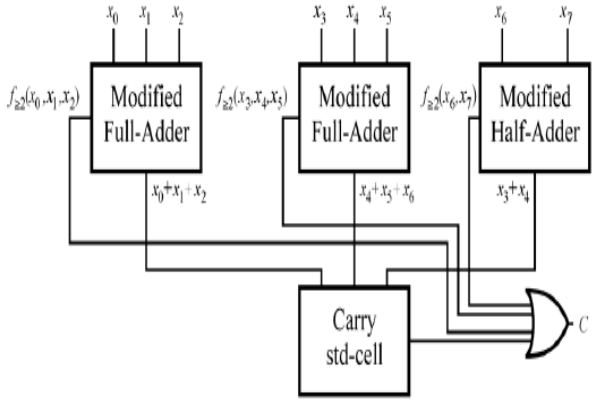

Fig 6. Implementation of carry output (C) of (8:2) speculative compressor [1]

B. *TDM Carry-Save Tree*

The TDM carry-save tree is a method for partial products reduction. Similar to carry propagate adder (CPA), TDM also has carry propagation. The minimum delay partial product reduction tree (PPRT) can be obtained by TDM using full-adders and few numbers of half-adders and it reduces the partial products into sum and carry.

The TDM considers the arrival time of the inputs for calculation. In the conventional speculative multiplier the computation of TDM for five inputs as follows:

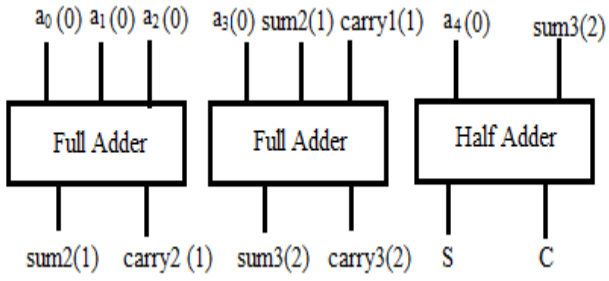

Fig. 7. Implementation of the TDM carry-save tree for five inputs

Fig. 7.shows the TDM calculation of five inputs, a_0, a_1, a_2, a_3 *and* a_4 and the arrival time of the inputs are shown in brackets ( ). Here the intermediate carry signals: carry2 and carry3 are not used for calculation since it has propagated to the next column of the tree. The inputs a_0, a_1, a_2, a_3 and a_4 have zero time delay, the intermediate signals sum2, carry2 and the carry signal: carry1 from the previous column of the tree have time delay one and have arrival time of two for intermediate signals sum3 and carry3. First three inputs a_0, a_1 and a_2 are given to a full-adder and the balance two inputs a_3 and a_4 are given to the separate components. The input a_3, the signals which have time delay one such as the carry signal: carry1 from the previous column of the tree and the sum signal, sum3 is computed next using a full-adder and then the input a_4 and sum4 signal with time delay two is computed using a half adder.

The proposed speculative multiplier uses new approach for TDM calculation and is shown below:

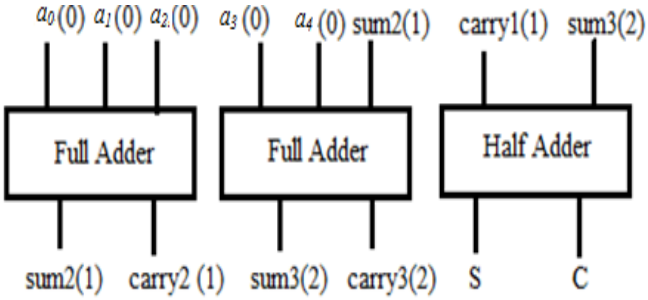

Fig. 8. Implementation of the proposed TDM carry-save tree for five inputs

The implementation of proposed TDM carry-save tree for five inputs is illustrated in Fig. 8. Here also inputs a_0, a_1, a_2, a_3 and a_4 have zero time delay, the intermediate signals sum2, carry2 have time delay one and the intermediate signals sum3 and carry3 have time delay two. In this method, inputs are computed according to the arrival time. The inputs with time delay zero is calculated first, next intermediate signals which have time delay one is computed, and then signals with arrival time of two are calculated. Here the input signals a_0, a_1, a_2, a_3 and a_4 have first preference. The inputs a_0, a_1 and a_2 are firstly computed using a full-adder, next the remaining inputs: a_3, a_4 and the intermediate signal sum2 which have time delay one is calculated. Finally, intermediate carry signal: carry1 from the previous column of the tree and sum3 signal is realized using a half-adder. Thus, the proposed method for calculation of TDM carry-save tree has less delay and power than the conventional method.

C. *Correction Block*

In the correction circuit error correction word *EW* are added together and the error flag *E* are OR-ed together. The structure of *(m :2)* correction circuits resemble to the *m*-bit speculative counter. Fig.9. illustrates the design of the (5:2) correction block. In this case, an error flag should assert when 4 or 5 inputs of the counter is high. This condition becomes false if at least two inputs of the counter are zero. Therefore, the error flag *E* can be calculated as:

$$E = \overline{f_{\geq 2}(\overline{x}_0, \overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4)} \quad (8)$$

This equation can be implemented as the carry output of a

(5:2) speculative compressor with inverted inputs and output. When the *E* is high, the speculative counter output is either 2 (when four inputs are equal to 1) or 3 (when the five inputs are equal to 1).

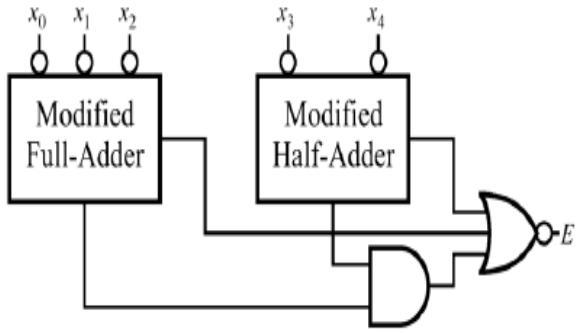

Fig. 9. Implementation of the (5:2) correction circuit. [1]

For the design of (6:2), (7:2) and (8:2) correction circuits, find all combination of inputs that contains number of one's more than three. Then check the sum and carry outputs and sort out the input combinations which have same sum and carry. From that combinations using full-adders and half-adders build the correction block for corresponding speculative counters.

IV. RESULT

Simulation was done in Xilinx 13.2. The result shows that the proposed speculative multiplier has less delay and power than conventional speculative multiplier.

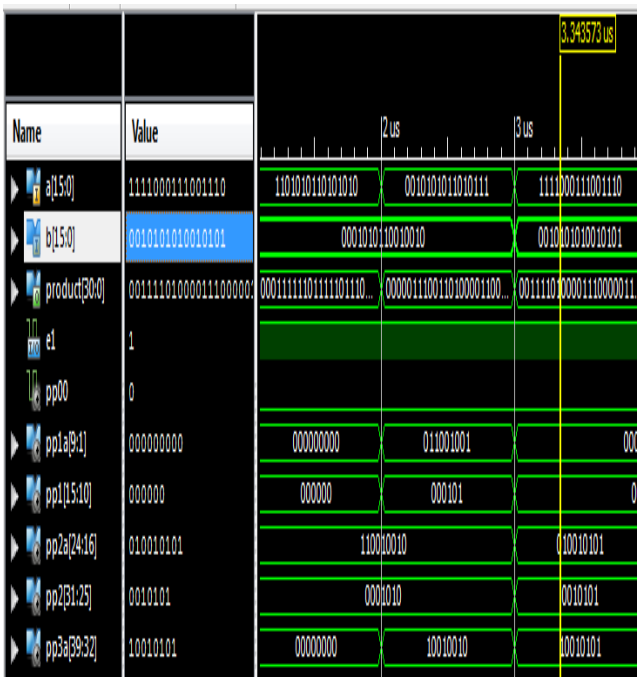

Fig. 10. Conventional 16 × 16 bit Speculative Multiplier output waveform.

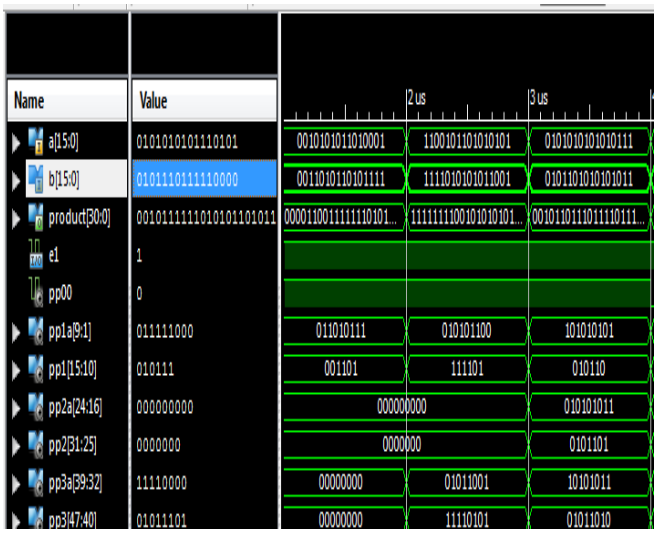

Fig. 11. Proposed 16 × 16 bit Speculative Multiplier output waveform.

V. COMPARISON OF CONVENTIONAL AND PROPOSED SPECULATIVE MULTIPLIER

Table 1: Comparison of delay and power of conventional and proposed Speculative multipliers

Multiplier	Delay (ns)	Power (mW)
Conventional Speculative Multiplier	38.518	819
Proposed Speculative Multiplier	34.598	786

VI. CONCLUSION

In this paper a method for TDM carry-save tree calculation in speculative multiplier is proposed. Both conventional and proposed speculative multipliers are simulated in Xilinx 13.2. Simulation result shows that the proposed speculative multiplier has less delay and power than conventional speculative multiplier.

REFERENCES

[1] Alessandro Cilardo, Davide De Caro, Nicola Petra, Frances- co Caserta, Nicola Mazzocca, Ettore Napoli, and Antonio Giusep- pe Maria Strollo, "High Speed Speculative Multipliers Based on Speculative Carry-Save Tree" IEEE TRANSACTI ONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAP ERS, VOL. 61, NO. 12, DECEMBER 2014

[2] J. Um and T. Kim, "An optimal allocation of carry-save- adders in arithmetic circuits," IEEE Trans. Comput., vol. 50, no. 3, pp. 215–233, Mar. 2001

[3] A. Cilardo, "A new speculative addition architecture suitable for two's complement operations," in Proc. Design, Autom, Test Eur. Conf. Exhib. (DATE), 2009, pp. 664–669.

Ms. Amrutha Balan received her B.Tech degree in Electronics and Communication Engineering from Jawaharlal College of Engineering and Technology, Palakkadu, India in 2012 and pursuing her M.Tech in VLSI and Embedded system at Mangalam College of Engineering, Kottayam, India.

Ms. Hima Sara Jacob, Assistant professor at Mangalam College of Engineering, Kottayam, India. She completed her B.Tech in Applied Electronics and Instrumentation from Saint Gits College of Engineering, Kottayam, India and received her M.Tech in VLSI and Embedded system from Saint Gits College of Engineering, Kottayam, India.