New Multiply-Accumulate Circuits Based on Variable Latency Speculative Architectures with Asynchronous Data Paths

Hoda Ghabeli[1], Amir Sabbagh Molahosseini [1*], Azadeh Alsadat Emrani Zarandi[2]
1- Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran.
Email: h.ghabeli@iauk.ac.ir
Email: sabbagh@iauk.ac.ir (Corresponding author)
2- Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.
Email: a.emrani@uk.ac.ir

Received: October 2021 Revised: January 2022 Accepted: March 2022

ABSTRACT:
This paper proposes the Variable Latency Speculative (VLS) Multiply-Accumulate (MAC) architectures. The proposed VLS architectures, unlike conventional MAC with fixed latencies, consists of two short and long data paths and a circuit is used to select a suitable path with minimum overhead. Two methods are considered to design the proposed VLS MAC. The first one considers the general structure of the VLS MAC with integrating the result vectors of multiplier with the accumulator, and the second method uses a novel VLS 4:2 compressor design. To investigate the proposed VLS MACs performance, all architectures have been synthesized using a CMOS 90 nm technology library, for operand lengths 8, 16 and 32 bits. Obtained results show that the proposed MAC architectures provide a variety of trade-offs in the power-delay-area space that outperform the existing designs that use only the integration technique. Moreover, the VLS MAC with the proposed VLS 4:2 compressor, in the short data path, has a delay equal to MAC with previously proposed approximate 4:2 compressor with an error recovery module. In comparison to MAC with the approximate 4:2 compressor, on average, the VLS MAC with the proposed VLS 4:2 compressor resulted in 11.26% and 13.59% lower area and power consumption.

KEYWORDS: Arithmetic Circuits, Multiply-Accumulator Unit, Variable Latency Speculative Circuits, 4:2 Compressor.

1. INTRODUCTION

Fast and real-time data analysis has been previouse coming the main challenge for digital signal processors (DSP) applications and deep learning (DL) recently. For example, many filters, convolutional operation, computer vision, images and speech recognition, they all face this problem [1]-[4]. The Multiply-Accumulate (MAC) units are widely used in both DSP and DL. Therefore, increasing the performance of MAC directly affects the performance of DSP and DL applications. The MAC includes both multiplier and adder units, which require high energy and delay, thereby affecting the two important parameters of power and speed [5], [6]. Approximate computing is a promising approach for improving performance, allowing to achieve high energy efficiency with controlled loss of accuracy. This technique can be applied in a variety of adder and multiplier circuits [7]-[9]. Approximate techniques can be used in a variety of ways to achieve area, delay and power trade-offs. However, unreliability and difficulty of detecting output error are challenges of the approximate computations. Speculative techniques are a subset of approximate techniques which can generate exact results. This method is usually referred as Variable Latency Speculative (VLS) [10]-[18]. A VLS computational circuit is a speculative design based on approximate techniques. The VLS circuit is designed to speculate a short path with fewer computations that allows for early completion [10]. Wrong speculation can be corrected by an error detection and correction circuit. In other words, in a VLS circuit, a short path is embedded as an alternative to the critical path that is probably to generate an accurate result, otherwise, the additional circuit will compensate for the erroneous probability [15]. There are two challenges in designing speculative circuits: 1) replacing the critical path with a

Paper type: Research paper
DOI: https:// 10.30486/mjee.2022.696494
How to cite this paper: H. Ghabeli, A. Sabbagh Molahosseini and A. Alsadat Emrani Zarandi," **New Multiply-Accumulate Circuits Based on Variable Latency Speculative Architectures with Asynchronous Data Paths**", Majlesi Journal of Electrical Engineering, Vol. 16, No. 2, pp. 41-53, 2022.

shorter path that is faster and it is speculated that the short path can generate an accurate result, along with an additional circuit for controlling the correctness of the speculation, 2) the area, power and delay overhead of the additional circuit for controlling the correctness of the speculation. Any error detection and correction unit in approximate designs can lead to area, power and even delay overhead. The synchronization of the error detection signal and the short path is also a part of the second challenge.

The first challenge for adders is almost solved, but it still exists for other arithmetic circuits. Despite the variety of VLS adders in recent years [10]-[17], there are not many solutions for VLS multiplier [18]. Its main reason is due to having only one carry propagation chain in the adder structures for computing worst-case delay. Here, the short path can be obtained by ignoring part of this carry propagation chain, and the ignored parts are added as an error correction circuit. But in the multiplier there is not only one carry propagation chain [18]. In other words, there is data dependency in both rows and columns. Due to this, there are limited works in the field of the VLS multipliers. Besides, although a variety approximate MACs that are based on approximate adders or multipliers for error tolerant applications have introduced in recent years [19]-[22], a VLS MAC has not been provided yet. In most approximate MAC designs, the general structure of the MAC is ignored [19]-[21]. In approximate MAC of [19], an approximate carry-propagate adder (CPA) is used. In the approximate MACs introduced in [20] and [21], an approximate multiplier is used. Adders in all of them is in the MAC repetition cycle and this leads to an increase in delay. Although, high-speed and optimized architectures for MAC have been proposed in [23]-[26], the use of approximate components in the MAC increases the error rate in the final result due to the repetition of the MAC cycle.

This paper proposed VLS MAC by two approaches that generate accurate results in each cycle. In the first one, an appropriate general architecture for VLS MAC is developed. A high-speed MAC architecture is a design with integration technique to use the approximation and VLS techniques. In the first approach, VLS MACs by using the integration technique are proposed. The second approach is developed based on building VLS 4:2 compressor for the VLS MAC. Compressors are commonly used in the design of high speed multipliers [27]. Although many approximate 4:2 compressors have been proposed [28]-[30], VLS 4:2 compressors have not yet been proposed. The previous approximate 4:2 compressor is designed with short paths along with an additional circuit for error recovery [30], in contrast to the proposed VLS 4:2 compressor that is designed with a critical path, which becomes shorter if some multiplier operand bits are zero. The VLS MAC with the VLS 4:2 compressor satisfies the mentioned challenges 1 and 2.

The structure of this paper is as follows. Section 2 is a review of the merging technique. The three types of a VLS MACs and the design strategy of the combining the integration technique and speculative technique are presented in section 3. Two types, Type-I and Type-II of a VLS MACs are based only on the integration technique and one type, Type-III of a VLS MAC is based on a new VLS 4:2 compressor, and the integration technique are presented in Section 3. Section 4 experimentally evaluates the merging technique of various levels of multiplier with the accumulator of MAC and combines with the speculative methods. Finally, section 5 concludes the paper.

2. THE INTEGRATION TECHNIQUE REVIEW

The structure of a general MAC consists of multiplier and accumulator. General architecture of MAC is based on a multiplier where output is added to the previous result in an accumulator. The MAC corresponds to the sum of the products. An integration technique is merging multiplication results before reaching the final addition of multiplier with accumulator to speed up MAC [23]-[26], as it is shown in Fig. 1 [23]. The advantage of this method is removing both the final addition of MAC and final addition of multiplier from the iteration cycle. It is also possible to extend the merging technique to achieve less delay [25], [26].

The binary multiply of two n-bit operands $A = a_{n-1}, a_{n-2}, \dots, a_0$ and $B = b_{n-1}, b_{n-2}, \dots, b_0$ can be done as a Wallace tree [31]. Any multiplication operation consists of three phases including Partial-Products Generation Phase (PPGP), Partial-Products Reduction Phase (PPRP), and final adder. In the PPGP, partial-products are computed from the point multiplication of multiplier and multiplicand.

For instance, b_0 should be multiplied to all bits of A and the same should be done for other bits of B. For each $n \times n$ multiplier n levels of partial-products are generated which partial-products of each level is denoted by pps_i where i is the index of the level and the number of levels is in the range of 1 to n. The concept of the PPRP is the summation of each column of partial-products which can be done with a Carry-Save Adder (CSA) or some half-adders and full-adders, as can be seen in Fig. 1. The CSA consists of a row of independent full-adders to reduce the three inputs of binary numbers to two outputs of sum and carry vectors. The total delay of a CSA is equal to the total delay of a single full-adder cell. The addition of each three-levels of pps_i is considered as one part. In each stage, based on the number of levels, there are some parts that can be done in parallel and considered as one stage, and some other

rows remain which are not belong to any part since there are less than three [32].

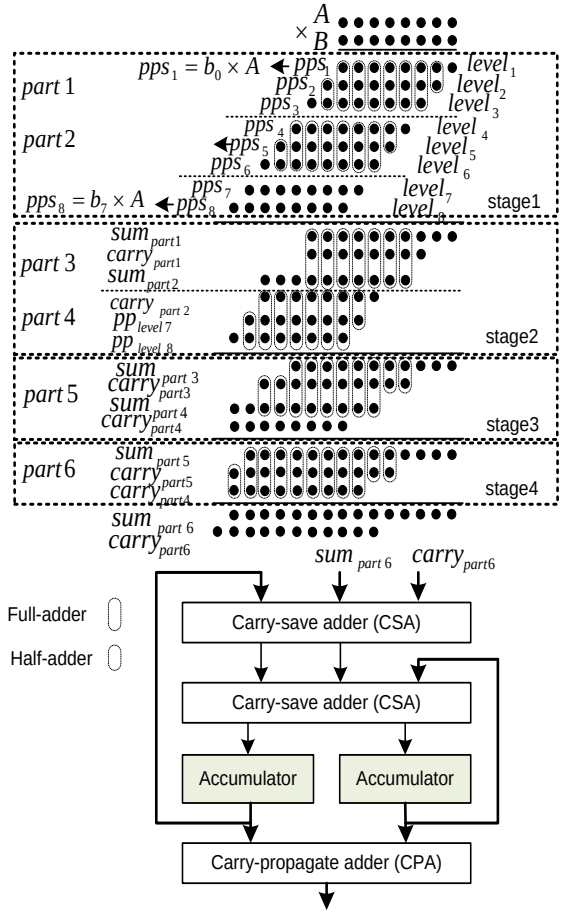

Fig. 1. conventional high-speed MAC (Integrate levels of result, the fourth stage of PPRP in pairs with the accumulator of 8×8 MAC [23]).

The sum and carry of each part (denote by sum_{parti}, $carry_{parti}$ and i corresponds to the number of part) are passed to the next stage until achieving the final sum and carry vectors. In the final phase of multiplication, a Carry-Propagate Adder (CPA) adder is required to add two values of sum and carry. It should be mentioned that different types of adders with different characteristics can be used. In the traditional MACs, the result can be achieved by adding the previous value of the accumulator with the output of multiplier after its final addition (third phase of multiplier). In other words, the output of the register is fed back to one input of the adder of the third phase of multiplication. A high-speed MAC by integration technique can be achieved by integrating two vectors sum and carry of multiplier with the accumulator in the PPRP of a multiplier as it is shown in Fig. 1 for 8×8 MAC. This means that instead of using the 2n-bit register, two 2n-bit registers can be used and the two outputs of the registers of sum and carry vectors are fed back to two inputs of the CSAs of the PPRP. However, MAC speed can be increased more by removing final phase of multipliers from the cycle of MAC and integrate them with accumulators. These deleted stages are moved after the repetition cycle of MAC.

2.1. Integration PPRP of Multiplier from One to All Stages with the Accumulator

In the extensive integration technique, one or all stages of a multiplier, in PPRP can be eliminated from cycle of MAC and merged with accumulator. In other words, the outputs of multipliers are added with the corresponding previously-stored results. Further parallelization is possible with this method which requires more registers. Although more parallelization, can increase area and power consumption besides improving the speed. In the MAC with extensive integration, instead of performing all reduction stages, only a few stages of the reduction operation are performed, and instead of the two vectors of sum and carry, some vectors of result are added with the corresponding previous results of accumulator. This operation can be performed by merged the number levels of results of PPRP with accumulator in two ways in pair and separately. In a pair, "double" means for every two output levels of result, there are two rows of CSA. Sum and carry vectors of the second row of CSA are restored in two accumulators and the two outputs of the registers are fed back to inputs of the first and second row CSAs. In other words, levels of result are merged in pairs with two accumulators. Separately, "single" means for every one output level of result, there is a row of CSA that sum and carry vectors of CSA are restored in two accumulators and the two outputs of the registers are fed back to inputs CSA. For example, Fig. 2.(a) shows integrate levels of result of the third stage of PPRP with the accumulator of 8×8 MAC, whereas the number of levels of result from third stage of PPRP of 8×8 multiplier is 3. So three levels of output results sum_{part5}, $carry_{part5}$ and $carry_{part4}$ are added with accumulators. The sum_{part5}, $carry_{part5}$ are merged in pairs and $carry_{part4}$ is merged separately. The removed stage of the PPRP is transferred to the next iteration cycle, which is done only once. Fig. 2(b) shows integrating levels of result from the second stage in pairs with the accumulator of 8×8 MAC. Whereas the number levels of result from second stage of 8×8 multipliers is 4 and thus the four results sum_{part3}, $carry_{part3}$, sum_{part4} and $carry_{part4}$ are added with the accumulators in pair. Finally, Fig. 2(c) shows the integration of levels of result from the first stage in pair with the accumulator of 8×8 MAC. Whereas the number of levels of result, from the first stage of 8×8 multiplier

is 6 and more results sum_{part1}, $carry_{part1}$, sum_{part2}, $carry_{part2}$, pps_7 and pps_8 in pair are added with the accumulator bits.

2.2. Integrating the PPGP with the Accumulator

The merging of the levels can be done in all stages of PPRP, up to the PPGP. In other words, without performing any PPRP, the pps_i of each multiplication operation are added to the previously stored values. The merging of pps_i levels in the PPGP can be done in two ways in pair and separately. For example, Fig. 3(a) shows the integration of pps_i levels in PPGP in pairs with the accumulator for the 8×8 MAC, whereas the number of pps_i levels, in the PPGP of the 8×8 multiplier is 8.

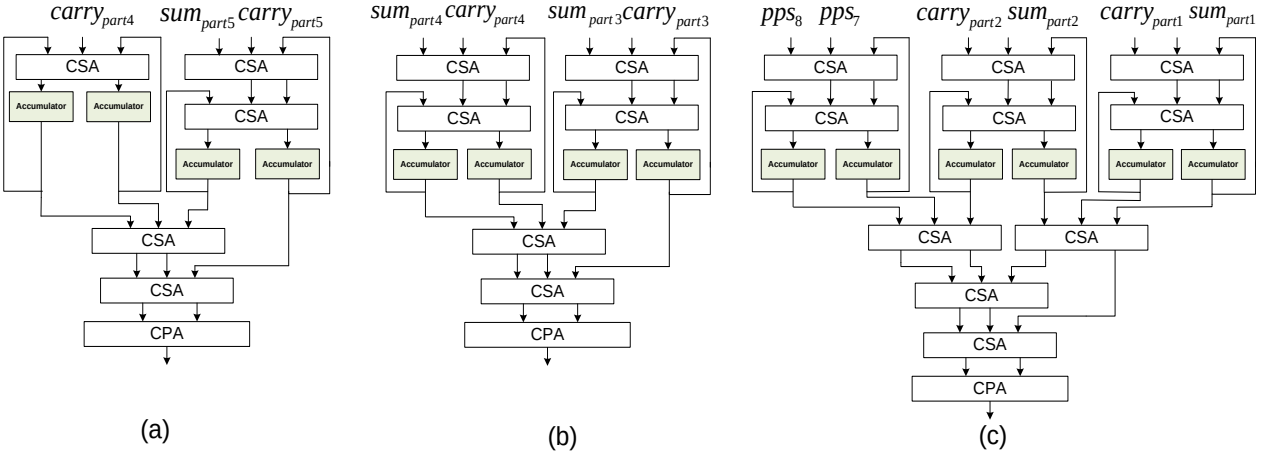

Fig. 2. (a) Integrate levels of result from the third stage of PPRP in pairs with the accumulator of 8×8 MAC, (b) Integrate levels of result from the second stage of PPRP in pairs with the accumulator of 8×8 MAC, (c) Integrate levels of result from the first stage of PPRP in pairs with the accumulator for the 8×8 MAC.

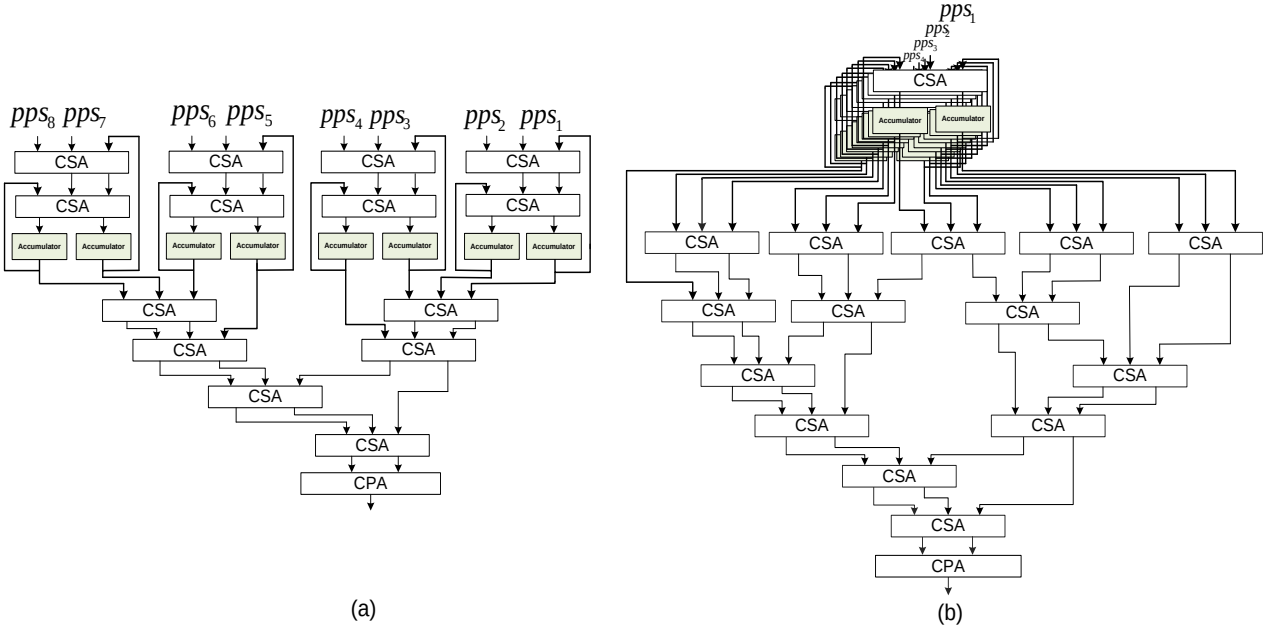

Fig. 3. (a) Integrate pps_i levels of PPGP in pairs with the accumulator for 8×8 MAC, (b) Integration of the separate pps_i levels from the PPGP with the accumulator for 8×8 MAC.

Therefore, 8 levels of result pps_1, pps_2, pps_3, pps_4, pps_5, pps_6, pps_7 and pps_8 are merged in pair with the accumulators. Fig. 3(b) shows integrating the separate pps_i levels in the PPGP with the accumulator for 8×8 MAC. Whereas, the difference between this method and the previous one is merging independently each level with the accumulator. Thus, the smaller number of CSA rows in iteration cycle of MAC is obtained by merging most levels of result with the accumulator as a result most of the CSA rows are transferred to the next cycle,

and this decreases the delay.

The integration technique separates the different stages of the multiplier and thus it becomes possible to find short path for a VLS multiplier especially for MAC.

3. PROPOSED VLS MAC

Many solutions have been proposed to reduce latency, including approximate calculation techniques and provide acceptable results for fault tolerable applications. The VLS technique can be effective for computational short paths selection without losing data accuracy, because at the same time the short path detection logic is used to detect the accuracy of the speculated short path.

In this way, in the VLS computation, two paths are predicted for calculations, a short path (non-critical path) and a long path (critical path). In this section, three designs for the VLS MAC are proposed. The main point for making short paths is to consider that in multiplying two values of $A = a_{n-1},a_{n-2},\dots,a_0$ and $B = b_{n-1},b_{n-2},\dots,b_0$, if each bit of b_i is zero, the pps_{i+1} will be completely zero (there is a whole level of zeros). In principle, a VLS MAC can be created by recognizing and ignoring these zero levels of pps_i. The designs are based on the integration technique of the multiplication phase with the accumulator as mentioned in the previous section. Type-1 architecture aims to find the short path by detecting scattered zero levels of pps_i, and Type-II architecture aims to find the short path by recognizing the last zero levels of pps_i. In the Type-III a novel VLS 4:2 compressor for using in MAC is proposed.

3.1. Type-I

The goal of Type-I, is to detect distributed zero levels of pps_i. In the following, two subclasses of Type-I are defined. Since the purpose of the VLS calculation is to find a shorter path than the critical path, the logical solution is where the speculative technique is used for the fastest possible states of MAC with extensive integration technique.

In MAC with integrating pps_i levels in the PPGP with the accumulator separately, it is not possible to satisfy the challenge 1. Because to find the short path with zero-level detection, each level of pps_i that is to be merged with the accumulator, must be checked to have a value of zero. A short path is activated when all levels of pps_i are zero, or value of B is zero, whilst this probability is very low. Hence Type-I-A with integrating the pps_i levels in the PPGP with the accumulator in pair and Type-I-B with integrating of levels of result from the first stage of PPRP with accumulator in pair are considered.

Type-I-A is integrating the pps_i levels in the PPGP in pair with the accumulator which can be combined with speculative technique. For example in Fig. 3(a), in the integration of pps_i levels pps_1, pps_2, pps_3, pps_4 pps_5, pps_6, pps_7 and pps_8 in pairs with the accumulator if each of the $b_0.b_1$, $b_2.b_3$, $b_4.b_5$ and $b_6.b_7$ logical expressions are zero, then there exists at least one zero levels from each of the two levels integrated with the accumulators. With this check and ensuring that one input from two inputs is zero, two inputs are converted to one input through an OR gate. In this way, the two rows of CSA will be critical path and the OR gate and CSA will be a short path. Zero level detection signal $b_i.b_{i+1}$, is selector of multiplexer for to select the appropriate path. Type-I-A is as shown in Fig. 4 for 8×8 MAC. It can lead to the deletion of a CSA row and replacing it with OR gate as a short path (red dotted line in Fig. 4). It should be mentioned that with the delay of a multiplexer to select a path, the final delay of short path becomes equal to the corresponding critical path of non-speculative MAC with integrating the pps_i levels of PPGP in pair (Delay OR gate + Delay full-adder + Delay multiplexer ≅ Delay full-adder + Delay full-adder). Besides, if the level is not zero and the critical path is selected, the overload contains a multiplexer delay. The delay related to the repetition cycle of MAC of each path of Type-I-A is determined as:

- The short path, shown with the red dotted line in Fig. 4 (non-critical path) = Delay $_{\text{OR gate}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_i.b_{i+1}= 0$.
- The long path, shown with the blue dotted line in Fig. 4 (critical path) = Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_i.b_{i+1} \neq 0$.
- The critical-path of non-speculative of the integration of the pps_i levels of PPGP in pair with the accumulator: delay related to the repetition cycle of MAC = Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$.

After explaining the previous case, Type-I-B is MAC with integrating of levels of result from the first stage of PPRP with accumulator in pair. For example, in Fig. 2(c), in the way that every three levels of pps_i levels, pps_1, pps_2, pps_3 and pps_4, pps_5, pps_6 in first stage of PPRP are reduced to two levels of result through the CSA. Then levels of results and the rest of the pps_i levels pps_7 and pps_8, in pair are added with values of the accumulators. In this case, for a VLS MAC, the detection of one zero level of pps_i is not effective due to the reasons mentioned earlier. But distinguishing two zero levels of pps_i from three levels pps_i is effective in reducing latency and Fig. 5 shows the Type-I-B. Detecting at least two zero levels from each three levels can lead to the deletion of two CSA rows and their replacement with two OR gate as a short path (red dotted line in Fig. 5). The three rows of CSA will be critical path (blue dotted line in Fig. 5). Zero levels detection signal $b_i.b_{i+1} + b_i.b_{i+2} + b_{i+1}.b_{i+2}$, is selector of multiplexer to select the appropriate path. The delay

related to the repetition cycle of MAC of each path of Type-I-B is determined as:

- The short path is shown with the red dotted line in Fig. 5 (non-critical path) = Delay $_{\text{OR gate}}$ + Delay $_{\text{OR gate}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_i.b_{i+1} + b_i.b_{i+2} + b_{i+1}.b_{i+2} = 0$.
- The long path is shown with the blue dotted line in Fig. 5 (critical path) = Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_i.b_{i+1} + b_i.b_{i+2} + b_{i+1}.b_{i+2} \neq 0$.
- The critical path of non-speculative integration the levels of result from the first stage of PPRP in pair with the accumulator: delay related to the repetition cycle of MAC = Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$ + Delay $_{\text{full-adder}}$.

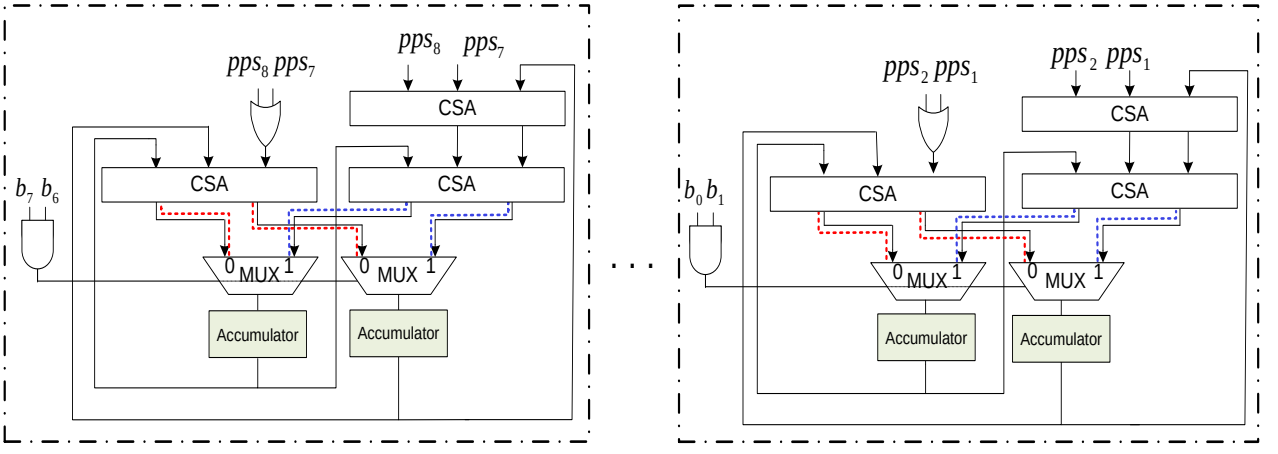

Fig. 4. The proposed Type-I-A architecture for n=8.

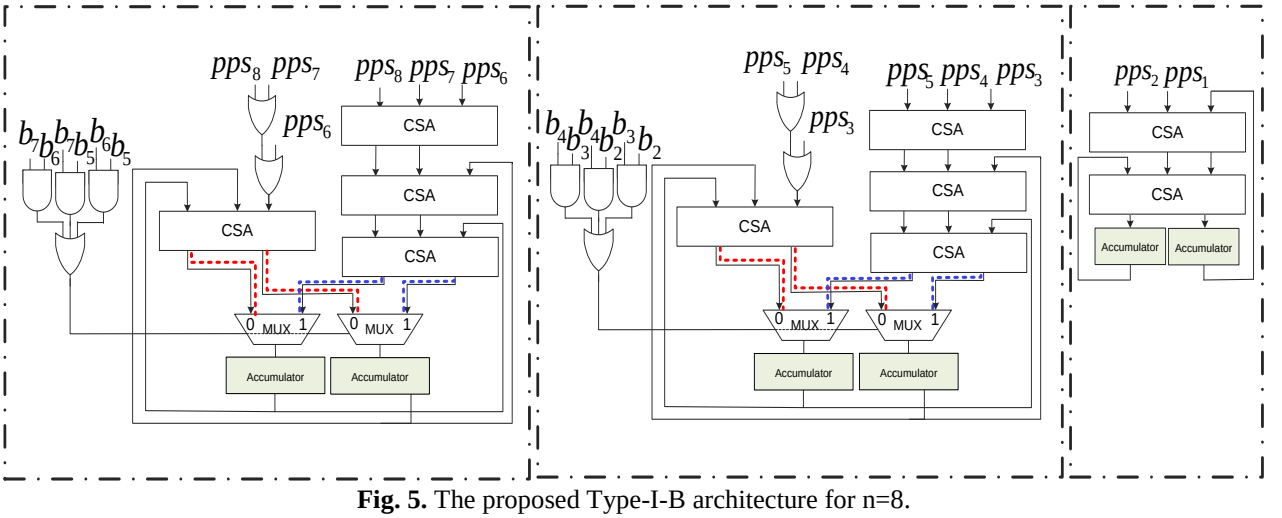

Fig. 5. The proposed Type-I-B architecture for n=8.

Delay reduction is when the short path is activated on all blocks at the same time. (integration every three levels of pps_i with accumulator is as a block). Although a short path may not be selected in all blocks simultaneously at once multiplication operation, but due to the repetition of multiplication operations in MAC, there is a probability of activating a short path and as a result, the total delay can be less than the non-speculative MAC depending on the input data.

3.2. Type-II

Second design of a VLS MAC is proposed to detection zero levels of pps_i from n×n multiplier if the $(\frac{n}{2})$ bits of the MSBs of multiplier operand B are all zeros. In Type-II-A, a shorter path is predicted in addition to the critical path, which, if the short path activation condition is true ($(\frac{n}{2})$ MSBs from operand B are all zeros) the results will be entered in the accumulator from a short path. Fig. 6 shows the 8×8 MAC with four stages of PPRP. If four bits of the MSBs of the B multiplier are zero ($b_7 + b_6 + b_5 + b_4 = 0$), then the last four levels of pps_i will be zero (pps_8, pps_7, pps_6 and pps_5 will be zero).

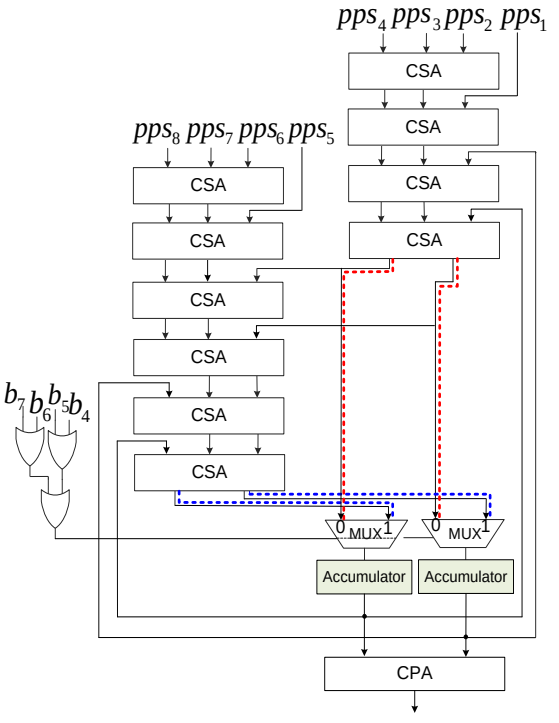

Fig. 6. The proposed Type-II-A architecture for n=8.

Detecting the last $(\frac{n}{2})$ zero levels can be effective in reducing latency because it eliminates two stages of PPRP as a short path (red dotted line in Fig. 6). If less than $(\frac{n}{2})$ zero levels are detected, only one stage is reduced compared to the critical path which is not a shorter path considering the addition of a multiplexer delay. Also, by recognizing zero levels more than $(\frac{n}{2})$, the probability of activating a short path will be reduced.

The architecture for Type-II-B offers a modified Type-II-A that leads to a reduction both in term of circuit area and power. Type-II-B shares the two final rows of CSA of Type-II-A between short and critical paths. Fig.7 shows the design of the Type-II-B for n=8.

The delay related to short and critical path of Type-II is determined as:

- The short path is shown with the red dotted line in Fig.7 (non-critical path) = number of stages of PPRP for $\frac{n}{2}$ levels of pps_i × Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_{n-1} + b_{n-2} + \cdots + b_{\frac{n}{2}} = 0$.
- The critical path is shown with the blue dotted line in Fig.7 (critical path) = number of stages of PPRP for n levels of pps_i × Delay $_{\text{full-adder}}$ + Delay $_{\text{multiplexer}}$ if $b_{n-1} + b_{n-2} + \cdots + b_{\frac{n}{2}} \neq 0$.

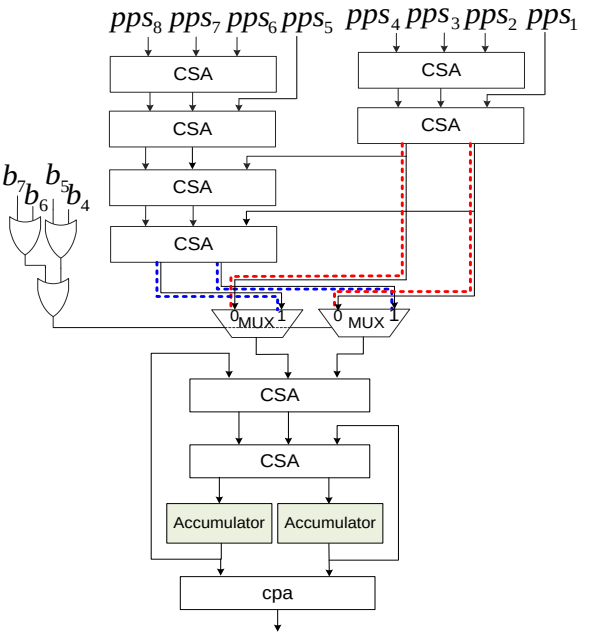

Fig. 7. The proposed Type-II-B architecture for n=8.

3.3. Type-III

One method to reduce Wallace tree delay is to use n:2 compressors instead of CSAs [27]. The 4:2 compressors are a widely used structure of this type. Fig. 8 shows the conventional structure for a 4:2 compressors.

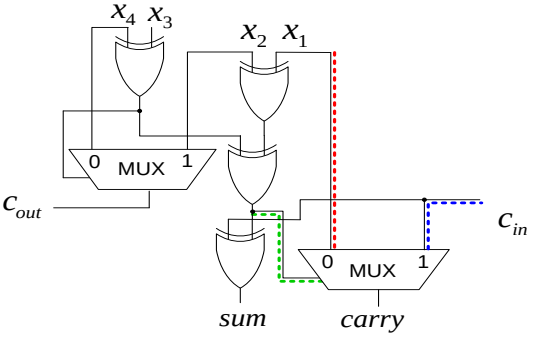

Fig. 8. The conventional 4:2 compressor.

A module of 4:2 compressor consists of four inputs x_1, x_2, x_3 and x_4, plus an input denote by c_{in} resulting from the right module of lower significant order and reciprocally an output called c_{out} to the left module of higher significant order and two outputs sum and $carry$. To compare different compressor designs, critical path delay is considered based on the number of gates [27], [33]. There are two paths to generate a $carry$. A carry generation path starts from the right 4:2 compressor module of the c_{in} generator through the multiplexer and another carry generation path begins from the x_1. The selection line between the two carry generation paths passes through two XOR gates. The

three outputs c_{out}, $carry$ and sum are obtained as follows:

$$c_{out} = (x_4 \oplus x_3).x_2 + \overline{(x_4 \oplus x_3)}.x_4 \quad (1)$$

$$carry = (x_4 \oplus x_3 \oplus x_2 \oplus x_1).c_{in} + \overline{(x_4 \oplus x_3 \oplus x_2 \oplus x_1)}.x_1 \quad (2)$$

$$sum = x_4 \oplus x_3 \oplus x_2 \oplus x_1 \oplus c_{in}. \quad (3)$$

The delay related to the repetition conventional 4:2 compressor is determined as:

- The first carry generation path delay (blue dashed line) (critical path) = Delay $_{multiplexer}$ + Delay $_{multiplexer}$.
- The second carry generation path delay (red dashed line) = Delay of select line (green dashed line) = Delay $_{XOR}$ + Delay $_{XOR}$ + Delay $_{multiplexer}$.
- The sum generation path delay = Delay $_{XOR}$ + Delay $_{XOR}$ + Delay $_{XOR}$.

As it can be seen, both carry generation paths have the same delay. In recent years, approximate compressor designs have been proposed to further increase compressor speed [28]-[30]. Simpler logical implementation is achieved by accepting a number of errors [28]. In these designs, the competition is for less error rate with respect to area, delay and power trade-offs. For this purpose, an error recovery module is embedded in the previous approximate 4:2 compressor [30]. The approximate 4:2 compressor [30], shown in Fig. 9, has and two outputs approximate (sum and $carry$) along with an error compensation resulting from the OR of two error detection signals belonging to two parts of a stage.

$$carry = x_1.x_2 + x_1.x_3 + x_1.x_4 + x_2.x_3 + x_2.x_4 \quad (4)$$

$$sum = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \quad (5)$$

$$error\ detection = x_3.x_4 \quad (6)$$

$$error\ compensation = (x_3.x_4)\ of\ part1 + (x_3.x_4)\ of\ part2. \quad (7)$$

For every two parts, a vector of error compensation results is added to the next stage of PPRP, which increases the number of stages and the area and power consumption, although $carry$ is generated from a shorter path than a conventional 4:2 compressor. And finally the result will be approximate.

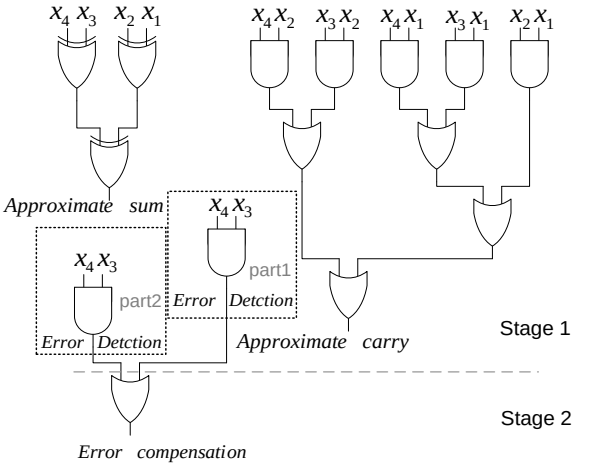

Fig. 9. The approximate 4:2 compressor with error recovery module [30].

In this paper VLS 4:2 compressor has been devised to speculate a shorter parallel path to the critical path to possibly increase speed with low overhead, while the result is exact. As mentioned before if each bit of b_i is zero, a row of pp_{i+1} will be completely zero. A VLS compressor design must consider this feature. In Type-III for each 4:2 compressors, if the b_i and b_{i-1} related to pp_{i+1} and pp_i respectively inputs to the compressor are zero, then there will be two complete levels of zero. In this case, x_3 and x_4, and follow them c_{in} and c_{out} in all modules will be zero. With this condition, sum is $x_1 \oplus x_2$ and as well as the generation of the $carry$ depends only on $x_1.x_2$. Otherwise the critical path of the conventional 4:2 compressors is selected. The $carry$ and sum are calculated through two multiplexers that get at the inputs:

$$carry\ of\ short\ path = x_1.x_2 \quad if\ \ b_i + b_{i+1} = 0,$$

$$carry\ of\ critical\ path = (x_4 \oplus x_3 \oplus x_2 \oplus x_1).c_{in} + \overline{(x_4 \oplus x_3 \oplus x_2 \oplus x_1)}.x_1 \quad otherwise \quad (8)$$

and

$$sum\ of\ short\ path = x_1 \oplus x_2 \quad if\ \ b_i + b_{i+1} = 0,$$

$$sum\ of\ critical\ path = x_4 \oplus x_3 \oplus x_2 \oplus x_1 \oplus c_{in} \quad otherwise \quad (9)$$

Where $b_i + b_{i+1}$ is assigned to the selection line of multiplexers. Fig. 10 shows the gate level

implementation of the proposed VLS 4:2 compressor design.

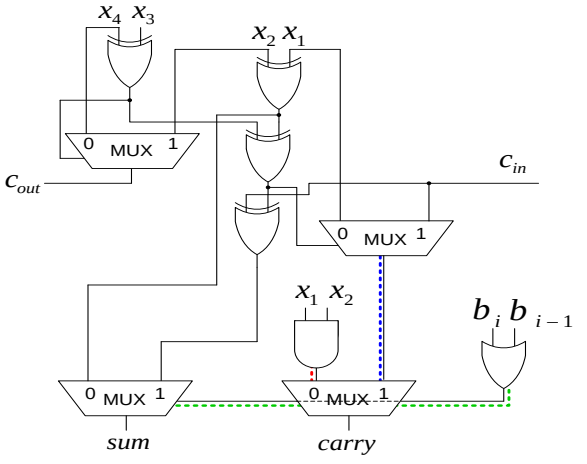

Fig. 10. The proposed VLS 4:2 compressor.

The delay related to the proposed VLS 4:2 compressor can be theoretically modelled as follow:

- The carry generation long path (blue dashed line) delay (critical path) = Delay $_{\text{multiplexer}}$ + Delay $_{\text{multiplexer}}$ + Delay $_{\text{multiplexer}}$ if $b_i + b_{i+1} \neq 0$.
- The carry generation short path (red dashed line) delay = Delay $_{\text{AND}}$ + Delay $_{\text{multiplexer}}$ if $b_i + b_{i+1} = 0$.
- The sum generation long path (critical path) delay = Delay $_{\text{XOR}}$ + Delay $_{\text{XOR}}$ + Delay $_{\text{XOR}}$ + Delay $_{\text{multiplexer}}$ if $b_i + b_{i+1} \neq 0$.
- The sum generation short path delay = Delay $_{\text{XOR}}$ + Delay $_{\text{multiplexer}}$ if $b_i + b_{i+1} = 0$.

In Type-III, the VLS 4:2 compressor proposed are used in the first stage of PPRP. So the general structure of the MAC is the design with integration levels of the result from the first stage of PPRP in pairs with the accumulator. Fig. 11 shows the proposed Type-III for n=8. The proposed VLS 4:2 compressor can only be used in the first stage of PPRP, where if both b_i and b_{i+1} are zero, the condition of activating a short path is true. The proposed VLS 4:2 compressor is comparable to the approximate 4:2 compressor [30] that uses an error recovery module. In this comparison, like the Type-III, the approximate 4:2 compressor is used only for the first stage of PPRP, and the error coverage is applied to all n columns (most and least significant column).

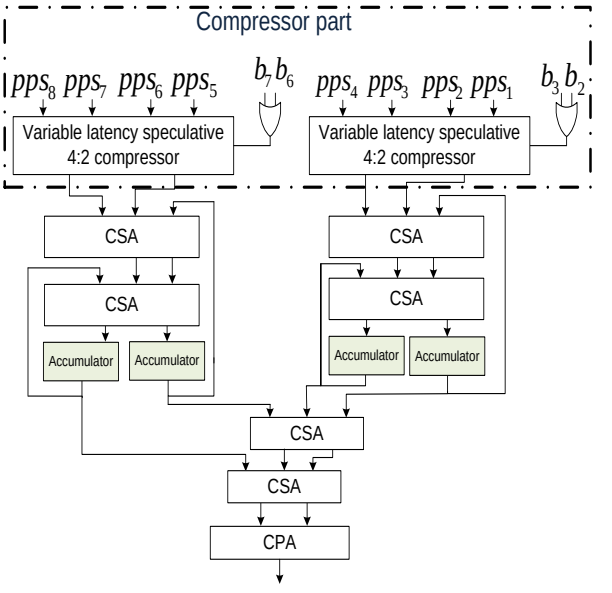

Fig. 11. The MAC with the proposed VLS 4:2 compressor (Type-III) for *n*=8.

4. RESULTS AND DISCUSSION

The MAC in the integration of the resulting levels in different multiplier phases and stages with the accumulator (section 2) and Type-I-B, Type-II-A, Type-II-B, Type-III (section 3) have been described in structural VHDL and synthesized with the Synopsys design compiler for the CMOS 90nm technology library, considering different values of *n* (8, 16 and 32 bits). Experimental results for the delay, power-consumption, and circuit area are presented in Tables 1 and Table 2. The shaded cells indicate the results of the proposed VLS MAC. Table 1 shows a comparison of delay, power-consumption, and circuit area between Type-I-B, Type-II-A, Type-II-B and the non-speculative MACs in the integration of the resulting levels in different multiplier phases and stages with the accumulator. Table 2 compares different compressor-based MACs including approximate [30], conventional and proposed VLS 4:2 compressors, in the first stage of PPRP. Besides, Table 3 shows the delay, area and power-consumption obtained for the relevant 4:2 compressor modules. The results of Table 1 show that, integrating with the accumulator in the higher stages, results to less delay, but integrating more levels with more registers, increases area, and power consumption. For 8×8 MAC of Type-I-B, the required power consumption and area are on average 18.11% and 9.26% less than the two fast modes (i.e. merging the pps_i of PPGP with the accumulator in pairs and separately), respectively. Whereas the delay in the short path (non-critical path) is 4.65% less than the integration pps_i of PPGP with the accumulator in pairs.

The Type-II-B shares similar delay to the Type-II-A, but in Type-II-B, the consumed area and power are

reduced on average 5.95% and 5.07% for different sizes, respectively, compared to Type-II-A. For 8×8 MAC of Type-II-A and Type-II-B less area and power consumption than the 8×8 MAC with merging the resulting levels in the stage 1 of PPRP and integration pps_i levels of PPGP in pairs and separately, are achieved. Whereas the delay in the short path is less than the 8×8 MAC with merging the resulting levels in the stages 4 and 3 of PPRP with accumulator. But the delay in the short path is not less than the 8×8 MAC with merging the resulting levels in the stage 2 of PPRP with accumulator. As a result, 8×8 MAC of Type-II-A and Type-II-B are not suitable alternatives for the 8×8 MACs with merging the resulting levels with the accumulator in the lower stages and with the smaller n_t.

For the 16×16 MAC of Type-I-B, the area is on average 18.11% lower than the two fast modes. But the delay in the short path is not less than the delay of the two faster modes. For 16×16 MAC of Type-II-A and Type-II-B, area and power consumption are less than the 16×16 MAC with merging the resulting levels of the stages 4, 3, 2 and 1 of PPRP and merging pps_i levels in PPGP with accumulator in pairs and separately. whereas the delay in the short path is reduced with gain 12.94% and 8.64% over the 16×16 MACs with merging the resulting levels in the sixth and fifth stages of PPRP with accumulator respectively.

The 32×32 MAC of Type-I-B has no improvement in terms of short path delay, area and power. It can be seen that the 16×16 and 32×32 MACs of Type-I-B are not suitable alternatives for the MACs with merging the resulting levels with the accumulator in the higher stages and with the larger n_t. In the cases of 32×32 MAC of Type-II-A and Type-II-B, the results are similar to the 16×16 MAC of Type-II-A and Type-II-B. The 32×32 MAC of Type-II-A and Type-II-B lead to 12.72%, and 9.43% faster in the short path compared to 32×32 MACs with merging the resulting levels in the stages 8 and 7 of PPRP with accumulator respectively. In summary, the Type-I-B for smaller values of n can be optimized to run fast with less power and area.

Table 1. Area, delay and power-consumption results for different MACs.

Architecture	Integration stage number of PPRP and PPGP	Number of merged levels for each two accumulators	Area (μm^2)	Power (mW)	Critical path delay (ns)	Short path delay (ns)
8×8 conventional high-speed MAC [23] Fig. 1	4	Double (in pairs)	4983.10	15.28	0.66	□
8×8 MAC Fig. 2.(a)	3	Double (in pairs)	5218.30	15.73	0.59	□
8×8 MAC Fig. 2.(b)	2	Double (in pairs)	5774.16	17.33	0.54	□
8×8 MAC Fig. 7	Type-II-B	Double (in pairs)	5907.43	16.54	0.73	0.58
8×8 MAC Fig. 6	Type-II-A	Double (in pairs)	6834.13	18.60	0.73	0.58
8×8 MAC Fig. Fig. 2.(c)	1	Double (in pairs)	7039.53	20.84	0.47	□
8×8 MAC Fig. 5	Type-I-B	Double (in pairs)	9818.03	23.41	0.51	0.41
8×8 MAC Fig. Fig. 3.(a)	PPGP	Double (in pairs)	9843.12	25.48	0.43	□
8×8 MAC Fig. Fig. 3.(b)	PPGP	Single (Separately)	12013.23	32.56	0.37	□□
16×16 conventional high-speed MAC [23]	6	Double (in pairs)	17879.11	55.17	0.85	□
16×16 MAC	5	Double (in pairs)	18656.84	57.90	0.81	□
	Type-II-B	Double (in pairs)	20018.65	60.34	0.91	0.74
	Type-II-A	Double (in pairs)	20812.85	62.36	0.91	0.74
	4	Double (in pairs)	21110.76	62.48	0.71	□
	3	Double (in pairs)	21781.08	65.41	0.66	□□
	2	Double (in pairs)	24518.81	72.79	0.57	□
	1	Double (in pairs)	28817.48	82.74	0.51	□
	Type-I-B	Double (in pairs)	37592.01	107.80	0.53	0.46
	PPGP	Double (in pairs)	38098.48	104.83	0.47	□
	PPGP	Single (Separately)	48005.10	137.43	0.43	□
32×32 conventional high-speed MAC [23]	8	Double (in pairs)	66070.82	202.64	1.10	□
32×32 MAC	7	Double (in pairs)	67028.86	209.14	1.06	□
	Type-II-B	Double (in pairs)	68268.37	215.88	1.15	0.96
	Type-II-A	Double (in pairs)	68593.73	217.86	1.15	0.96
	6	Double (in pairs)	71722.67	229.36	0.94	□
	5	Double (in pairs)	75602.68	236.45	0.88	□
	4	Double (in pairs)	77502.31	249.32	0.79	□
	3	Double (in pairs)	78316.11	254.14	0.72	□
	2	Double (in pairs)	81337.64	285.41	0.60	□

	1	Double (in pairs)	87082.01	289.87	0.52	□
	PPGP	Double (in pairs)	94183.48	341.05	0.49	□
	Type-I-B	Double (in pairs)	100901.58	293.37	0.58	0.50
	PPGP	Single (Separately)	108421.31	359.68	0.47	□

Table 2. Area, delay and power-consumption results for different compressor-based MACs.

Architecture	compressor part for first stage of PPRP	Area	Power (mw)	Critical path delay (ns)	Short path delay (ns)
8×8 MAC with integration levels of result from the first stage of PPRP	VLS 4:2 compressor (Type-III)	7792.96	18.87	0.58	0.49
	Conventional 4:2 compressor	6958.78	18.75	0.53	□
	approximate 4:2 compressor [30]	9008.16	22.31	0.50	□
16×16 MAC with integration levels of result from the first stage of PPRP	VLS 4:2 compressor (Type-III)	25328.68	59.63	0.63	0.54
	Conventional 4:2 compressor	21427.50	59.61	0.57	□
	Approximate 4:2 compressor [30]	25620.33	61.55	0.54	□
32×32 MAC with integration levels of result from the first stage of PPRP	VLS 4:2 compressor (Type-III)	81790.01	223.81	0.68	0.57
	Conventional 4:2 compressor	67956.33	212.86	0.61	□
	Approximate 4:2 compressor [30]	101170.49	295.17	0.57	□

For larger values of n, the Type-II-B is expected to perform comparatively better performance in both power and area. The choice between these can be based on the need for area, power and delay.

For the proposed MAC of Type-III, the short path delay of the MAC using the proposed VLS 4:2 compressor has a delay similar to the critical path of the MAC using the approximate 4:2 compressor for all MAC widths. Whereas area and power parameters of the MACs using the proposed VLS 4:2 compressor are lower compared with the MAC using the approximate 4:2 compressor [30] on average, by 11.26% and 13.59% respectively, for all MAC width. The VLS MAC with the VLS 4:2 compressor, satisfies challenges 1 and 2. Although as shown in Table 3, the approximate compressor [30] consumes an average of 24.45% less power than the conventional and proposed compressor on average. However, error recovery leads to the addition of error vectors in the later stages of PPRP, resulting in more stages and an increase in area and power consumption.

The advantage of proposed circuits depends on the probability of activating the short path. The probability of activating the short path is subject to the input data. In some applications a multiplier must perform multiplications for different word lengths depending on the operation mode. In the simpler case a n-bit multiplier is used for one k-bit multiplication, where ($k \leq n$), the proposed VLS MACs of Type-II and Type-III can be useful. Furthermore, in applications using 2's complement representation, there is a long chain of zero values if the operands are small and positive where the proposed VLS MACs of Type-II and Type-III can be useful. The non-speculative MAC, is the best option if the probability of occurrence of the mentioned cases is low, otherwise the average delay of the proposed VLS MACs is either equal to the critical path or less than the critical path of non-speculative MACs.

5. CONCLUSION

This work addresses the challenge of designing asynchronous data paths for the MAC. The proposed VLS MACs can outperform the existing fixed-latency and approximate MACs. The MAC architecture by integration technique features, can be used for VLS computing. On the basis of this technique, three types, Type-I, Type-II and Type-III of VLS MACs with alternative area, power and delay characteristics have been presented. VLS computations give a chance to increase the computational speed by guaranteeing accurate result. In the Type-I and Type-II, it first focuses on general MAC structure with the extensive integration technique. In the Type-III, an attempt is made to provide a VLS MAC by the VLS components used in the MAC. Therefore, the novel VLS 4:2 compressor is designed to generate VLS MAC. The proposed VLS 4:2 compressor can replace the high precision approximate 4:2 compressor when high speed and accurate results are required. The proposed Type-III offers different advantages in terms of speed, area and power consumption. Finally, the increase in speed of the proposed designs depends on the input data of the MAC.

Table 3. Area, delay and power-consumption experimental results for different 4:2 compressors.

4:2 compressor design	Area	Power (μw)	Critical path delay (ns)	Short path delay (ns)
Proposed VLS 4:2 compressor	1081.1	439.25	0.23	0.11
Conventional 4:2 compressor	729.90	304.45	0.18	□
Approximate 4:2 compressor [30]	1005.8	271.69	0.15	□

REFERENCES

[1] B.H. Lee, & S.M. Kuo, **"Real Time Digital Signal Processing, Implementations, Applications and Experiments with the TMS320C55x"** *John Wiley & Sons LTD*, New York (2001) p. 330

[2] V. Gierenz, C. Panis, J. Nurmi, **"Parameterized MAC unit generation for a scalable embedded DSP core,"** *Microprocessors and Microsystems*, Vol. 34 (5), pp. 138–150, 2010.

[3] K. Benkrid, S. Belkacemi, **"Design and implementation of a 2D convolution core for video applications on FPGAs,"** *Digital and Computational Video, DCV 2002. Proceedings. Third International Workshop on, (2002)*, pp.85-92.

[4] M. Verhelst and B. Moons, **"Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices,"** *IEEE Solid-State Circuits Magazine*, Vol. 9(4), pp.55-65, 2017.

[5] J. Chang, H. Lee, and C. Choi, **"A power-aware variable-precision multiply-accumulate unit,"** *in International Symposium on Communications and Information Technology*, pp. 1336–1339, 2009.

[6] H. Lee, **"Power-Aware Scalable Booth Multiplier,"** *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E88-A, No. 11, pp.3230-323, 20054.

[7] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, **"Approximate arithmetic circuits: A survey, characterization and recent applications,"** *Proceedings of the IEEE*, Vol. 108, No. 12, pp. 2108-2135, Dec. 2020.

[8] L. Sousa, **"Nonconventional Computer Arithmetic Circuits, Systems and Applications,"** *IEEE Circuits and Systems Magazine*, Vol. 21, No 1, pp. 6-40, March 2021.

[9] J. .Hu, Z. Li, M.Yang, Z.Huang, and W. Qian, **"A high-accuracy approximate adder with correct sign calculation,"** *Integration, the VLSI Journal*, Vol. 65, pp. 370-388, March 2019.

[10] K. Verma et al., **"Variable latency speculative addition: a new paradigm for arithmetic circuit design,"** *in Proc. Design, Automation and Test in Europe*, pp. 1250—1255, 2008.

[11] K. Du, P. Varman, and K. Mohanram, **"High performance reliable variable latency carry select addition,"** *Proc. Design, Autom. Test Eur*., pp. 1257–1262, 2012.

[12] A. Cilardo, **"A new speculative addition architecture suitable for two's comple- ment operations,"** *in Proc. Design, Automation and Test in Europe*, pp. 664—669, 2009.

[13] D. Kelly and J. Phillips, **"Arithmetic data value speculation,"** *Adv. Comput. Syst. Architecture, Lecture Notes Comput. Sci*., pp. 353–366, 2005.

[14] S. M. Nowick, K. Y. Yun, P. A. Beerel, and A. E. Dooply, **"Speculative completion for the design of high-performance asynchronous dynamic adders,"** in Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, Apr. 1997, pp. 210–223.

[15] D. Esposito, D. De Caro, A.G.M. Strollo, **"Variable Latency Speculative Parallel Prefix Adders for Unsigned and Signed Operands, "** IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 63, n. 8, pp. 1200-1209, Aug. 2016.

[16] I.-C. Lin, Y.-M. Yang, and C.-C. Lin, **"High-performance low-power carry speculative addition with variable latency,"** *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*., Vol. 23, No. 9, pp. 1591–1603, Sep. 2015.

[17] Y. Choi and E. E. Swartzlander, **"Speculative Carry Generation with Prefix Adder,"** *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 16, No. 3, pp. 321-326, March 2008.

[18] A. Cilardo, D. De Caro, N. Petra, F. Caserta, N. Mazzocca, E. Napoli, and A. G. M. Strollo, **"High speed speculative multipliers based on speculative carry-save tree,"** *IEEE Trans. Circuits Syst. I, Reg. Papers*, Vol. 61, No. 12, (2014), pp. 3426–3435.

[19] D. Esposito, D. De Caro, E. Napoli, N. Petra and A. G. M. Strollo, **"On the use of approximate adders in carry-save multiplieraccumulators,"** *IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD*, pp. 1-4, 2017.

[20] D. Esposito, A. G. M. Strollo, and M. Alioto**, "Low-power approximate MAC unit,"** *inProc. IEEE PRIME, Giardini Naxos, Italy*, pp. 81–84, 2017.

[21] G. A. Gillani, M. A. Hanif, M. Krone, S. H. Gerez, M. Shafique, andA. B. J. Kokkeler, **"Designing approximate MAC accelerators with internal-self-healing,"***IEEE Access*, Vol. 7, pp. 142–77, 2019.

[22] M. Masadeh, O. Hasan, and S. Tahar, **"Input-Conscious Approximate Multiply-Accumulate (MAC) Unit for Energy-Efficiency,"** *IEEE Access*, Vol. 7, pp. 129–147, 2019.

[23] B. Parhami **"Computer arithmetic, algorithms and hardware designs. "** *New York: Oxford Press*; 2000.

[24] H. Parandeh-Afshar, S.M. Fakhraie, and O.Fatemi, **"Parallel Merged Multiplier-Accumulator Coprocessor Optimized for Digital Filters"**, *Elsevier Journal of Computers and Electrical Engineering*, No.36, pp.864-873, 2008.

[25] AA. Fayed, MA. Bayoumi **"A merged multiplier–accumulator for high speed signal processing applications,"** *IEEE Trans VLSI*, Vol. 3(2), 2002.

[26] J. Wang, L. Xu, H. Wang and C. Choy, **"A high-speed pipeline architecture of squarer-accumulator (SQAC),"** *IEEE Region 10 Conference (TENCON), Singapore*, pp. 3429-3432, 2016.

[27] C. H. Chang, J. Gu, and M. Zhang, **"Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits,**" *IEEE Trans. Circuits Syst. I, Reg. Papers*, Vol. 51, No. 10, pp. 1985–1997, Oct. 2004.

[28] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, **"Design and analysisof approximate compressors for multiplication,"** *IEEE Trans. Comput.*, Vol. 64, No. 4, pp. 984-994, Apr. 2015.

[29] Z. Yang, J. Han, and F. Lombardi, **"Approximate compressors for error-resilient multiplier**

design," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Amherst*, MA, 2015.

[30] M. Ha and S. Lee, **"Multipliers with approximate 4–2 compressors and error recovery modules,"** *IEEE Embedded Syst. Lett.*, Vol. 10, No. 1, pp. 6–9, Mar. 2018.

[31] L. S. Wallace, **"A suggestion for fast multipliers,"** *IEEE Trans.Comput.*, Vol. EC-13, pp. 14–17, 1964.

[32] R. S. Waters and E. E. Swartzlander, **"A reduced complexity Wallace multiplier reduction,"** *IEEE Transactions on Computers*, Vol. 59, No. 8, pp. 1134–1137, August 2010.

[33] A. Rahnamaei and A.K. Sarkaleh, "**High Performance Low Latency 16×16 bit Booth Multiplier using Novel 4-2 Compressor Structure**," *Majlesi Journal of Electrical Engineering*, Vol. 14, No. 2, pp. 1-9, 2020.