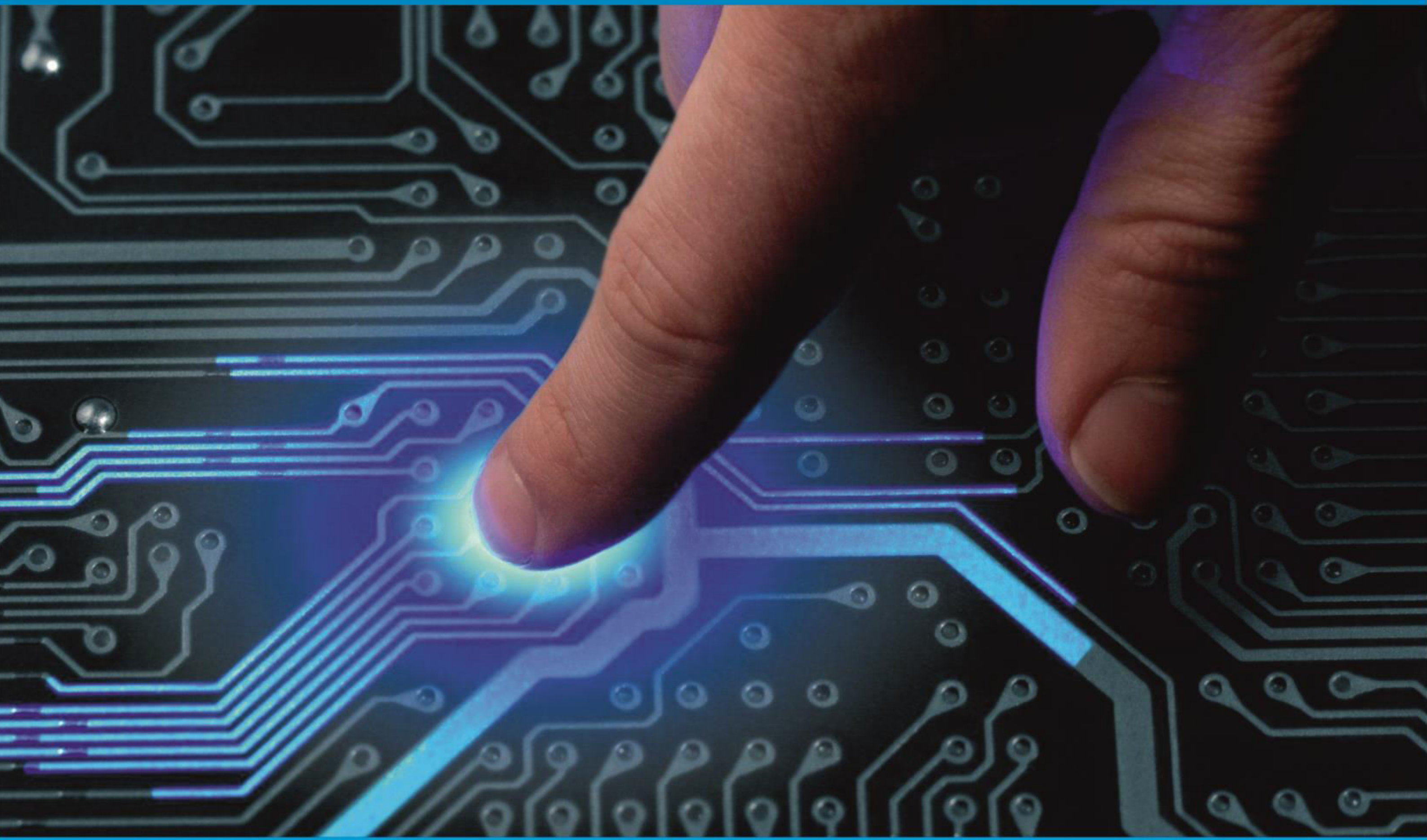
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 3, March 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

Area and Power Efficient Multiplier Based on Multi-bit Compressors and Counters

M.Sathya[1], Dr.C.N.Marimuthu[2]

Department of Electronics and Communication Engineering, Nandha Engineering College, Erode, India[1]

Professor & Dean (R&D), Department of Electronics and Communication Engineering, Nandha Engineering College, Erode, India[2]

ABSTRACT: Design methodology for ultrahigh-speed 5-2 and 7-2 compressors has been illustrated in this article. With the help of introduced procedure, the gate-level delay has been reduced considerably when compared with the previous designs, while the total transistor and gate count remain in a reasonable range. By starting the discussion for the carry rippling problem in n − 2 compressors, the method has been developed for 5-2 compressor and is expanded for 7-2 architecture, which shows 32% and 30% improvement in speed performance for these structures, respectively. Further, higher order 6:3 and 7:3 binary counter based on symmetric stacking can also be added to the design to further reduce the critical path delay. It uses 3-bit stacking circuits, which group all of the "1" bits together, followed by a novel symmetric method to combine pairs of 3-bit stacks into 6-bit stacks. The bit stacks are then converted to binary counts, producing 6:3 counter circuits with no XOR gates on the critical path. This avoidance of XOR gates results in faster designs with efficient power and area utilization. Finally, a typical 16 × 16 bit multiplier has been implemented to investigate the efficiency of the designed compressor and counter blocks. Based on the synthesis results provided using Xilinx ISE, the proposed compressors and counters demonstrate better speed performance and power–delay product (PDP) factor over previous works.

KEYWORDS: 6-3 Counter , 7:3 Binary Counter, 5-2 Compressor, 7-2 Compressor, Bit stacking, Carry propagation, High performance, High speed, Parallel multiplier.

I. INTRODUCTION

Because of their higher efficiency, parallel multipliers are now extremely used in many high-speed systems such as digital signal processors (DSPs), central processing units (CPUs), and multimedia applications. In most of the CPUs, the multiplier lies in the critical path for signal propagation. To decrease the delay and complexity of such systems, practical design considerations have been pursued over recent year. In a parallel multiplier, the multiplication process is divided into three steps. At first, the partial products (PPs) are generated. Then these products are summed, and the process continues until two rows remain, and at the final stage, the two remaining rows will be added by means of, for example, a carry propagation adder.

At the second stage and after generation of PPs, a partial product reduction tree (PPRT) is often employed for efficient summation of the products. Considering the full adder (FA) as the main building block in various multiplication configurations, this block constitutes the basis for much different architecture.

However, the main drawback of FA-based configurations, which limits their usage in today's parallel multiplier design, is the propagation latency for the cascaded cells. Moreover, the most important concern in the design process of a parallel multiplier is the circuit size and power consumption, which is directly related to the number of employed gates used in the various parts of the architecture.

An efficient solution to overcome such drawbacks is to utilize a compressor network instead of FA trees, especially in the PPRT. Furthermore, comprehensive analysis depicts that the most significant part of the total delay and power dissipation will belong to this stage in a parallel multiplier. Therefore, the performance enhancement of this stage can significantly improve the speed and lower the power dissipations of the whole system.

As a universal principle, an $n - 2$ compressor modifies the interconnection of the adjacent cells in the accumulation stage by means of one or more horizontal paths defined as Cout1, Cout2, etc. These paths will transmit their logic value to the adjoining compressor, which has one binary bit higher order in significance known as *C*in1, *C*in2, etc. for the latter compressor, respectively. By reducing the signal transfer load for the vertical trajectories known as Sum and Carry outputs, significant performance enhancement can be achieved. However, the carry rippling will be unfolded as a new problem for such structures that must be taken into account. Such an obstacle has barely been considered in previously reported works.

Considering the above, the focus in this article is to improve the speed performance of $n - 2$ compressors by bringing up the carry rippling problem as a new design consideration. Also, the fact that power and active area on-chip should be comparable to the best-reported designs is focused. Therefore, the main emphasis was on the neutral states of the outputs for horizontal paths. Utilization of the newly proposed truth table has resulted in a fast 303-ps 5-2 compressor (designed in a CMOS standard 0.18-μm process) in which the gate-level delay is fewer than three XOR logic gates. By expanding the same idea for the 7-2 compressor, the gate-level latency of four XOR logic gates is obtained, which is a considerable speed optimization in compressor design criteria.

II. DESIGN CONSIDERATIONS OF N:2 COMPRESSOR

A.LITERATURE SURVEY

The art of summing up the numbers with minimum carry propagation delay is one of the common speed improvement techniques utilized in state-of-the-art digital circuits. The basic idea is to reduce three numbers to two numbers with the help of an FA, which is the general definition of a 3-2 counter block .

The 4:2 compressor is the simplest form for the realization of an n−2 compressor. In conventional form, it is constructed by means of cascaded attachment of two FAs. In a similar fashion, a 5-2 compressor is simply obtained by cascading three FAs. Because three FA blocks have been arranged after each other, at least five XOR logic gate-level delay is expected for such structure. By means of the optimizations reported in the gate-level latency was reduced to four XOR logic gates.There are also other 5-2 compressor circuits which have their own advantages and drawbacks.

The common disadvantage between all of the previous works is the rippling of the horizontal outputs (*C*out1 or *C*out2*)* for at least three consecutive stages. This problem, which has barely been investigated before, drastically degrades the performance of the compressor cell. Hence, the real gate level latency will be equal to five XOR logic gates for the circuits reported Also, the configuration of, which is based on conventional architecture, contains no speed enhancement when compared. Moreover, the corresponding latencies for compressors of will be more than six XOR logic gates.
The rippling problem (for at least three stages) also exists for higher order compressor blocks. The emphasis over the other works reported in the literature proves the hypothesis.

Herein, we are about to demonstrate that if the conventional truth table of each compressor is investigated carefully, then the rippling effect can be reduced considerably. Also, other parameters like power and active area can be maintained at a moderate level.

B.CARRY RIPPLING PROBLEM

One of the essential key factors on the performance enhancement of an $n - 2$ compressor is the reduction of carry rippling issues between adjacent compressor cells (for 5-2 or higher order structures). To clarify such an issue, which directly increases the total delay of the compressor block, we can consider the horizontal cascading of three 5-2 compressor structures.

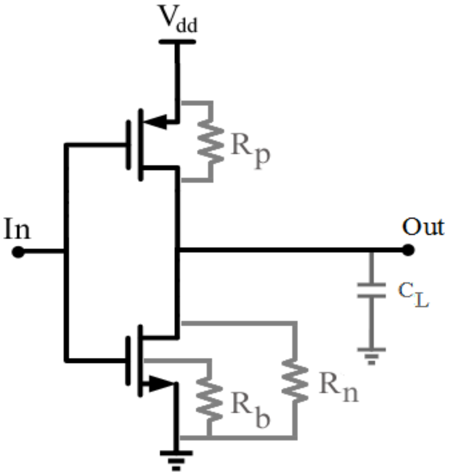


Fig. 1. Configuration of an inverter for propagation delay evaluation.

There is a logical dependence between *C*out2 and *C*in1, which is highlighted with the dashed lines for the middle stage compressor. Therefore, the input of the first-level compressor will be rippled to the third-stage compressor structure due to this dependence. This obstacle will ultimately degrade the speed performance of the designed architecture, which can clearly be seen in all of the previously reported works. On the other hand, if the design considerations will be built on the independency of *C*out1 and *C*out2 outputs from input *C*in1 and *C*in2 bits, then the structure will be obtained. As it is evident, the carry rippling has been reduced for two cascaded stages.

It must be mentioned that this procedure can be utilized for higher order compressors, as well. But as the complexity of the calculations grows along with the circuit-level implementation, the authors only relied on the results obtained for 5-2 and 7-2 compressors. The best solution for the carry rippling obstacle is the modification of the conventional truth table for each compressor cell. The operation starts with the reconsideration of neutral states in the truth table. By assigning the desired logic values to those states, the table can be divided into distinct regions. As a result, logical simplifications can be applied to the corresponding output that lies in the critical path of signal propagation.

In the conventional compressors based on full adder has a carry rippling effect.which is the main drawback of conventional compressor.The existing 5:2 and 7:2 compressors without full adders reduces the critical path delay but still consumes more power and area due to the presence of large number of XOR gates.

III. MULTIPLIER

For digital signal processing and applications involving signal and image processing, multipliers and adders form an important part. So, speed of the multipliers and adders affect speed of the operation. For high speed applications, faster multipliers are recommended. Several techniques to increase multiplier speed are proposed. Multiplication is done by adding partial product terms.Implementation of multiplier comprises three steps:

- Generation of partial products
- Partial products reduction tree
- A vector merge addition to produce final product from the sum and carry rows generated from the reduction tree.

Second step consumes more power. In this approximation is applied in reduction tree stage.

Partial product accumulation in 4 × 4 multipliers can be represented as, consider two 4-bit unsigned operands $\alpha = \sum_{i=0}^{3} \alpha_i 2^i$and $\beta = \sum_{j=0}^{3} \beta_j 2^j$. The partial product array pp is a 4×4-bit array of the partial product bits $pp_{i,j} = \alpha_i \beta_j$, where i, $j\in$ {0, 1, 2, 3}. Table 1.1 gives all the partial products for a 4-bit multiplication and their corresponding product bits.

The product is denoted by $\gamma = \sum_{k=0}^{7} \gamma_k 2^k$. The bits of γ are produced in stages going from the LSB to the MSB. According to Table 1.1,$\gamma_0 = pp_{0,0}$ and there is no further operation in Stage 0. In Stage 1, to generate γ_1, we can simply use a half adder that produces a sum bit $\gamma_1$and a carry bit (c_1) for the next stage.

Table I Original partial product of the multiplication

Stage 7	Stage 6	Stage 5	Stage 4	Stage 3	Stage 2	Stage 1	Stage 0
	$pp_{3,3}$	$pp_{3,2}$	$pp_{3,1}$	$pp_{3,0}$	$pp_{2,0}$	$pp_{1,0}$	$pp_{0,0}$
		$pp_{2,3}$	$pp_{2,2}$	$pp_{2,1}$	$pp_{1,1}$	$pp_{0,1}$	
			$pp_{1,3}$	$pp_{1,2}$	$pp_{0,2}$		
				$pp_{0,3}$			
γ_7	γ_6	γ_5	γ_4	γ_3	γ_2	γ_1	γ_0

Since the half adder circuit is already a simple design, there is no need to approximate it.In Stage 2, there are three pp terms and the carry from the previous stage (c_1) that must be added together. Thus, a 4:2 compressor is required to generateγ_1 and a carry for the next stage.

IV. COMPRESSORS

A.5:2 COMPRESSOR

The existing circuit for 5-2 compressor has been demonstrated in Fig. 2. For the better realization of the latency in different paths from inputs to the outputs, two neighbor 5-2 compressor cells must be connected together as the carry rippling finishes in the second compressor block. This point is discussed in the simulation section. By considering this fact and as shown in Fig.2, the critical path will belong to the generation of Carry output.It must be taken into account that all of the MUX gates except the one that produces Cout1, and the gate which is fed by Gnd, are channel-ready gates. As a result, they will exhibit a latency equal to 0.25 delta. Moreover, Bbar is produced by means of an inverter at the output node of the nonfull swing XOR gate that generates B. However, it does not affect the delay of the whole system.

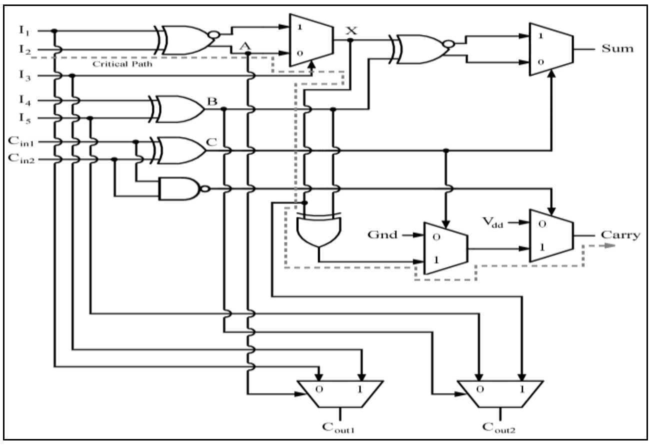


Fig.2. 5:2 Compressor

For high logic values at the input stage of TG, the output capacitor will be charged to Vdd–Vth by means of the NMOS transistor. At the same time, Bbar is transferred to the output of the inverter gate. With the help of Bbar, which enables the PMOS transistor, the output capacitor will fully be charged to Vdd value. For low logic states in the input stage of TG, the NMOS path itself discharges the output capacitor to the zero value.

The two output XOR–XNOR gates generate both outputs simultaneously, which are designed and wholly discussed. Because of the similar paths, their different outputs have little influence on the delay and glitch effect of TG waveforms. Also, static CMOS has been employed to design the architectures of NAND and NOR gates.

If the outputs of these gates are inverted, then the AND/OR gates will be obtained, and since these gates have not been located in the critical path, therefore, they would not affect the latency of the whole system.To calculate the delay of the critical path for the proposed 5-2 compressor in the logic gate level, we refer to the calculations provided in the previous section. As stated above, the critical path belongs to the route starting from I1 and I2 and finishing in Carry output. In the conventional compressors based on full adder has a carry rippling effect.The Existing 5:2 and 7:2 compressors without full adders reduces the critical path delay but still consumes more power and area due to the presence of large number of XOR gates.Recently M.Sathya and Dr.C.N.Marimuthu [2] relied on a similar fasion of designed structure.Critical path delay is high due to the presence of large number of XOR gates.To overcome this drawback in the existing compressor we can go for the 6:3 and 7:3 counter based on symmetric stacking along with 4:2 compressors.

This hybrid design results in the reduction of xor gates by ordering the input of the counter before starting the sum and carry calculation.These Hybrid structure helps in reducing total number of logic gates. Parallelism by partitioning bits increases the performance.These designs are helpful to avoid delay in the critical path by reducing XOR gates.

B.7:2 COMPRESSOR

The literature review depicts that the optimization of design considerations was less considered for 7-2 compressor circuits. Moreover, the carry rippling issue still exists for at least three stages. Only Giuseppe Caruso and Daniela Di Sclafan resolved the rippling problem, although because of careless design, the critical path consists of almost six cascaded XOR gates.By employing the same procedure for the 5-2 compressor, the novel architecture for the 7-2 compressor has been constructed. In addition to higher speed performance when compared with the previous designs, the carry rippling between adjacent blocks has been reduced by one stage. Fig.4. shows the designed scheme, in which the latency from inputs to outputs is reduced to less than four XOR logic gates.

The 7:2 compressor is same as the 5:2 compressor.But slightly different.In the 5:2 compressor circuit, the parameters denoted as I1, I2, I3, I4, and I5 constitute the primary inputs, while Cin1 and Cin2 are the secondary inputs.In the 7:2 compressor circuit the inputs denoted as I1–I7 will organize the primary inputs while Cin1 and Cin2 are the secondary inputs.

In the conventional compressors based on full adder has a carry rippling effect.The Existing 5:2 and 7:2 compressors without full adders reduces the critical path delay but still consumes more power and area due to the presence of large number of XOR gates.Critical path delay is high due to the presence of large number of XOR gates.To overcome this drawback in the existing compressor we can go for the 6:3 and 7:3 counter based on symmetric stacking along with 4:2 compressors.

This hybrid design results in the reduction of xor gates by ordering the input of the counter before starting the sum and carry calculation.These Hybrid structure helps in reducing total number of logic gates. Parallelism by partitioning bits increases the performance.These designs are helpful to avoid delay in the critical path by reducing XOR gates.

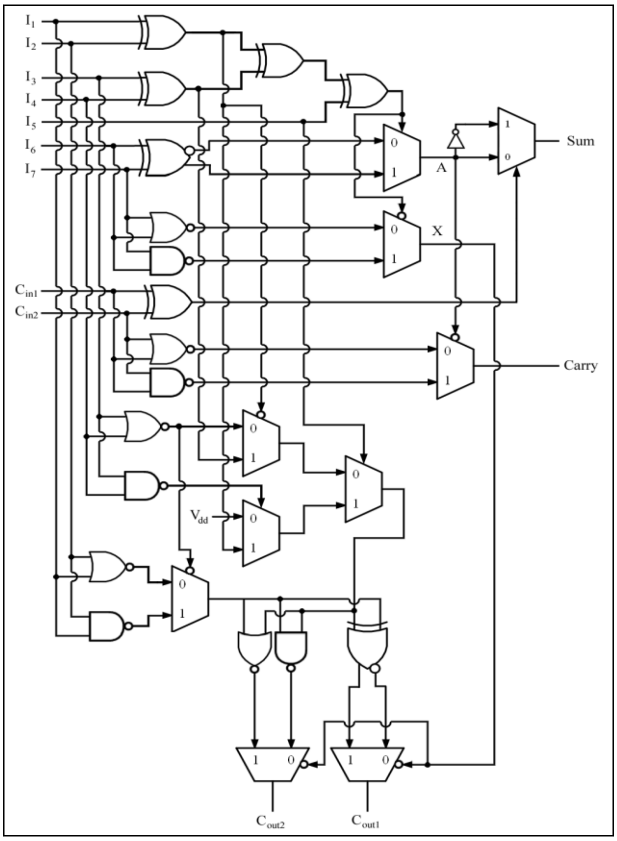


Fig.3. 7:2 Compressor

V. PROPOSED COUNTERS

In proposed method, uses 6:3 and 7:3 counters based on stacking technique along with 4:2 compressors. This hybrid design results in the reduction of xor gates by ordering the input of the counter before starting the sum and carry calculation.6:3 and 7:3 counters uses only, 1 and 2 XOR gates respectively, minimizing the overall gates compared to the existing compressors.The proposed 6:3 counter based on bit stacking has no XOR gates on its critical path, it operates nearly 30% faster than all other counter designs. Thus, this novel method of counting via bit stacking allows construction of a counter for a substantial performance increase without increasing power consumption.

A.SYMMETRIC BIT STACKING

The proposed 6:3 counter is realized by first stacking all of the input bits such that all of the "1" bits are grouped together. After stacking the input bits, this stack can be converted into a binary count to output the 6- bit count. Small 3-bit stacking circuits are first used to form 3-bit stacks. These 3-bit stacks are then combined to make a 6-bit stack using a symmetric technique that adds one extra layer of logic.

B.THREE-BIT STACKING CIRCUIT

Given inputs X0, X1, and X2, a 3-bit stacker circuit will have three outputs Y0, Y1, and Y2 such that the number of "1" bits in the outputs is the same as the number of "1" bits in the inputs, but the "1" bits are grouped together to the left followed by the "0" bits. It is clear that the outputs are then formed by,

Y0 = X0 + X1 + X2 (1)
Y1 = X0X1 + X0X2 + X1X2 (2)

Y2 = X0X1X2 (3)

Namely, the first output will be "1" if any of the inputs is one, the second output will be"1" if any two of the inputs are one, and the last output will be one if all three of the inputs are "1." The Y1 output is a majority function and can be implemented using one complex CMOS gate. The 3-bit stacking circuit is shown in Fig.4.

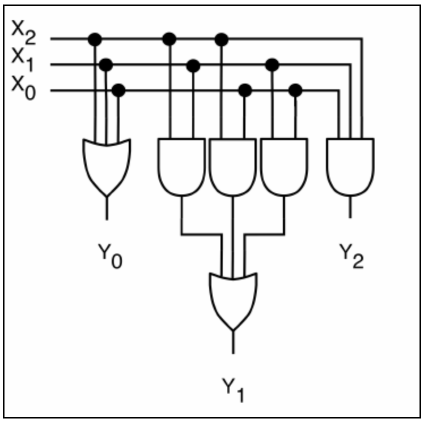


Fig.4. Three-Bit Stacker Circuit.

C.MERGING STACKS

We wish to form a 6-bit stacking circuit using the 3-bit stacking circuits discussed. Given six inputs X0, . . . , X5, we first divide them into two groups of 3 bits which are stacked using 3-bit stacking circuits.

Let X0, X1, and X2 be stacked into signals named H0, H1, and H2 and X3, X4, and X5 be stacked into I0, I, and I2. First, we reverse the outputs of the first stacker and consider the six bits H2,H1,H0, I0, I1,andI 2. See the top of Fig.4.1 for an example of this process. We notice that within these six bits, there is a train of "1" bits surrounded by "0" bits. To form a proper stack, this train of "1" bits must start from the leftmost bit. In order to form the proper 6-bit stack, two more 3-bit vectors of bits are formed called J0, J1, J2 and K0, K1, K2. The idea is to fill the J vector with ones first, before filling the K vector.
So we let,

J0 = H2 + I0 (4)
J1 = H1 + I1 (5)
J2 = H0 + I2 (6)

In this way, the first three "1" bits of the train are guaranteed to fill into the J bits although they may not be properly stacked. Now to ensure no bits are counted twice, the K bits are formed using the same inputs but with the AND gates instead.

K0 = H2 I0 (7)
K1 = H1 I1 (8)
K2 = H0 I2 (9)

If the train of "1"s is no more than three places long, then all of the K bits will be "0" as the AND gate inputs are three positions apart. If the train is longer than three places long, then some of the AND gates will have both inputs as "1"s as the AND gate inputs are three positions apart. The number of AND gates that will have this property will be three less than the length of the train of "1"s. We notice that now J0 J1 J2 and K0K1K2 still contain the same number of "1" bits as the input in total but now J bits will be filled with ones before any of the K bits. We must now stack J0 J1 J2 and

K0K1K2 using two more 3-bit stacking circuits. The outputs of these two circuits can then be concatenated to form the stack outputs Y5, . . . , Y0.

D.CONVERTING BIT STACK TO BINARY NUMBER

In order to implement a 6:3 counter circuit, the 6-bit stack described must be converted to a binary number. For a faster, more efficient count, we can use intermediate values H, I, and K to quickly compute each output bit without needing the bottom layer of stackers. Call the output bits C2, C1, and S in which C2, C1, S is the binary representation of the number of “1” input bits. To compute S, we note that we can easily determine the parity of the outputs from the first layer of 3-bit stackers. Even parity occurs in the H if zero or two “1” bits appear in X0, X1, and X2. Thus, He and Ie, which indicate even parity in the H and I bits, are given by,

He = + H1
Ie = + I1 . (10)

As S indicates odd parity over all of the input bits, and because the sum of two numbers with different parities is odd, we can compute B0 as,

S = He ^ Ie. (11)

Although this does incur one XOR gate delay, it is not on the critical path. To compute C1, we note C1 = 1 when the count is 2, 3, or 6. Therefore, there are two cases. First, we need to check if we have at least two but no more than three total inputs. We can use the intermediate H, I, and K vectors for this. To check for at least two inputs we need to see stacks of length two from either top level stacker, or two stacks of length one, which yields H1 + I1 + H0 I0.

VI. PROPOSED 6:3 COUNTER BASED ON SYMMETRIC STACKING CIRCUIT

The final 6:3 counter circuit can be constructed, as shown in Fig.5 Using larger CMOS gates, the critical path delay is reduced to seven basic gates. As there are no XOR gates on the critical path, this 6:3 counter outperforms existing designs are represented. One drawback of this design is an increase in wiring complexity: The symmetric approach necessitates signals crossing after the first layer of stackers, while traditional counters, as in Fig.7.

The 6:3 counter based symmetric stacking circuit is nothing but the stacking circuit. In this circuit the input X0,X1,X2 and X3,X4,X5 are given to the 3 bit stacker circuit. Then which produce the output K0,K1,K2.The bit stacker are converted the binary bit “0”
bits together and “1” bits together. After the bit stacker conversion, the XOR gates are reduced when the sum and carry calculation takes place. The XOR gates are reduced compared to existing designs. Therefore, due to the XOR gates reduction in the circuit the gate count of the circuit is reduced. So the area of the circuit is reduced compared to existing compressors. The given Fig.4.2 representing the proposed 6:3 counter based on symmetric stacking circuit. Because the proposed 6:3 counter based on bit stacking has no XOR gates on its critical path, it operates nearly 30% faster than all other counter designs. Thus, this novel method of counting via bit stacking allows construction of a counter for a substantial performance increase without increasing power consumption.

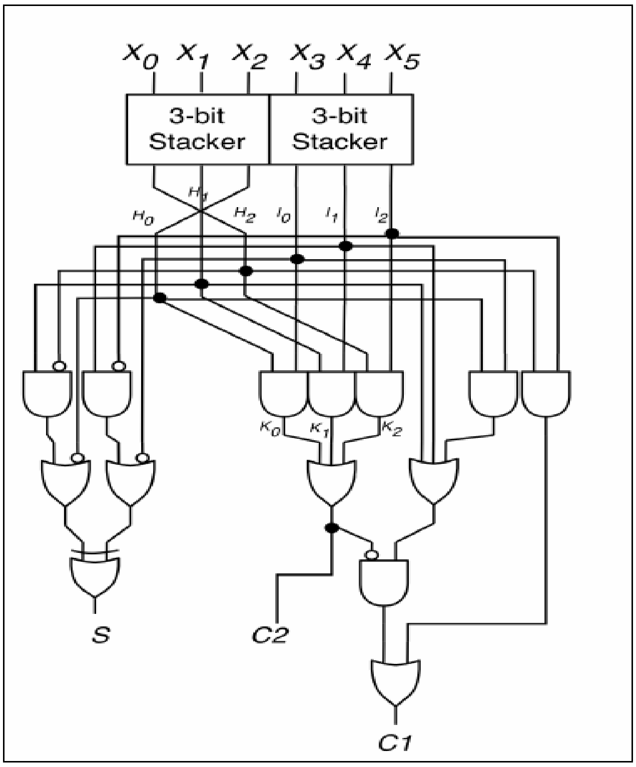


Fig.5. Proposed 6:3 Counter Based on Symmetric Stacking

The proposed 6:3 counter based on bit stacking has no XOR gates on its critical path, it operates nearly 30% faster than all other counter designs. Thus, this novel method of counting via bit stacking allows construction of a counter for a substantial performance increase without increasing power consumption. The 6:3 counter is going to apply in the 6 bit input of the multiplier circuit. The proposed 6:3 counter based on bit stacking has no XOR gates on its critical path,Therefore the gate count is reduced.Then area also reduced. Therefore overall performance of the circuit is increased.

VII. PROPOSED 7:3 COUNTER BASED ON SYMMETRIC STACKING CIRCUIT

The proposed 7:3 counter based on symmetric stacking circuit is nothing but the stacking circuit.Which is represented in Fig.6. The proposed 7:3 counter circuit is same as the 6:3 counter circuit. The main difference between the circuit is the 6:3 counter occupies the 6 bit input, but the 7:3 counter occupies 7 bit inputs in the multiplier circuit. The 7:3 counter based symmetric stacking circuit is nothing but the stacking circuit. In this circuit the input X0,X1,X2 and X3,X4,X5 are given to the 3 bit stacker circuit. Then which produce the output 0,K1,K2.The stacker are converted the binary bit “0” bits together and “1” bits together. After the bit stacker conversion, the XOR gates are reduced when the sum and carry calculation takes place. The XOR gates are reduced compared to existing designs. Therefore, due to the XOR gates reduction in the circuit the gate count of the circuit is reduced.

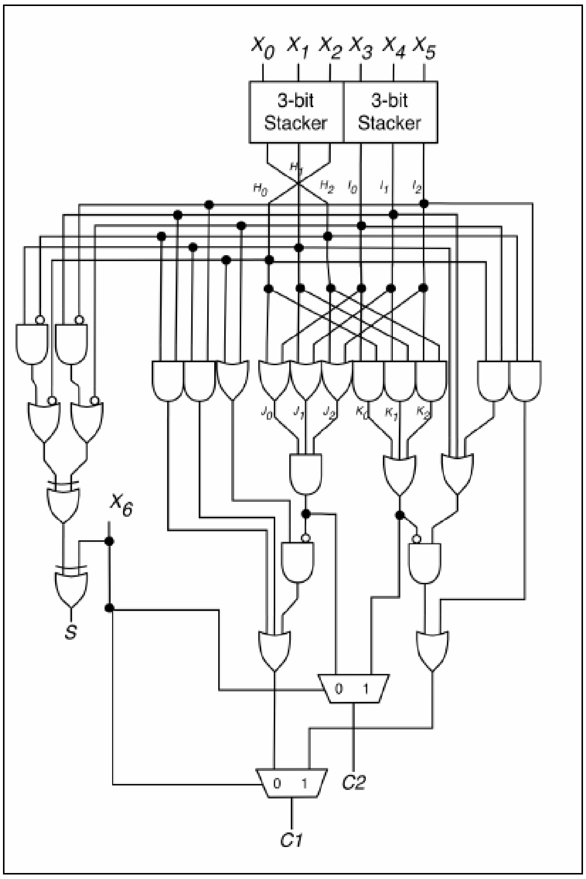


Fig.6. Proposed 7:3 Counter Based on Symmetric Stacking.

So the area of the circuit is reduced compared to existing compressors. The Fig.6 representing the proposed 7:3 counter based on symmetric stacking circuit**.** Because the proposed 7:3 counter based on bit stacking has no XOR gates on its critical path, it operates nearly 30% faster than all other counter designs. Thus, this novel method of counting via bit stacking allows construction of a counter for a substantial performance increase without increasing power consumption.

VIII. SCALING UP TO LARGE MULTIPLIER

In order to construct larger, e.g. 16×16 and 32×32, approximate multipliers, the two proposed 4×4 multipliers are combined in an array structure. For instance, to construct an 8×8 multiplier using a 4× 4 design, the two 8-bit operands A and B are partitioned into two 4-bit nibbles, ,for A and, for B., are the 4 MSBs and,indicate the 4 LSBs of A and B, respectively. Each two of these four nibbles (in total 4 possible combinations) are multiplied using 4×4 multipliers and the partial products are then shifted (based on the nibble's importance) and added together (using a Wallace tree architecture) to produce the final multiplication result. Building 2n×2n multipliers using n×n multipliers is specified in Fig.7.

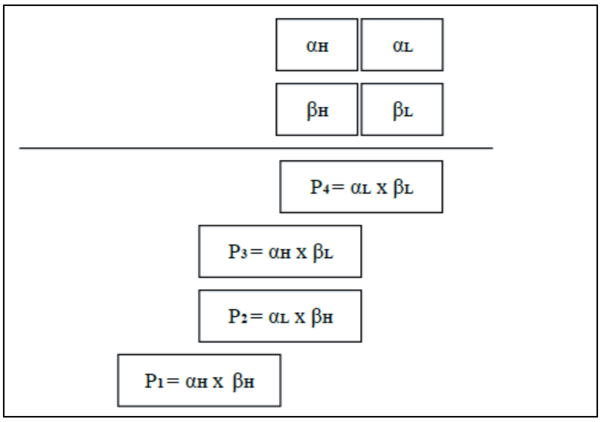


Fig.7. Building 2n×2n multipliers using n×n multipliers

Each partial product P_i where $i\in$ {1, 2, 3, 4} in equation 1 is generated using an n×n multiplier and multiplications by 2^{2n} and 2^n are simply done by $2n$-bit and n-bit left shifts, respectively. Given that $P_4$is the least and P_1 is the most significant partial products, whereas P_2 and P_3 are equivalently significant, multipliers with different accuracies can be designed with different configurations. We propose six 8×8 approximate multipliers, three of which, i.e. M8-1, M8-3, and M8-5, use M1 and the other three use M2 as their main building block.

IX. PROPOSED 8×8 BIT MULTIPLIER

Because of their higher efficiency, parallel multipliers are now extremely used in many high-speed systems such as digital signal processors (DSPs), central processing units (CPUs), and multimedia applications. In most of the CPUs, the multiplier lies in the critical path for signal propagation. To decrease the delay and complexity of such systems, practical design considerations have been pursued over recent years. In a parallel multiplier, the multiplication process is divided into three steps. At first, the partial products (PPs) are generated.

Then these products are summed, and the process continues until two rows remain, and at the final stage, the two remaining rows will be added by means of, for example, a carry propagation adder. At the second stage and after generation of PPs, a partial product reduction tree (PPRT) is often employed for efficient summation of the products. Considering the full adder (FA) as the main building block in various multiplication configurations, this block constitutes the basis for much different architecture.

The Fig.8.representing the architecture of the proposed 8x8 bit multiplier.However, the main drawback of FA-based configurations, which limits their usage in today’s parallel multiplier design, is the propagation latency for the cascaded cells. Moreover, the most important concern in the design process of a parallel multiplier is the circuit size and power consumption, which is directly related to the number of employed gates used in the various parts of the architecture. An efficient solution to overcome such drawbacks is to utilize a compressor network instead of FA trees, especially in the PPRT.

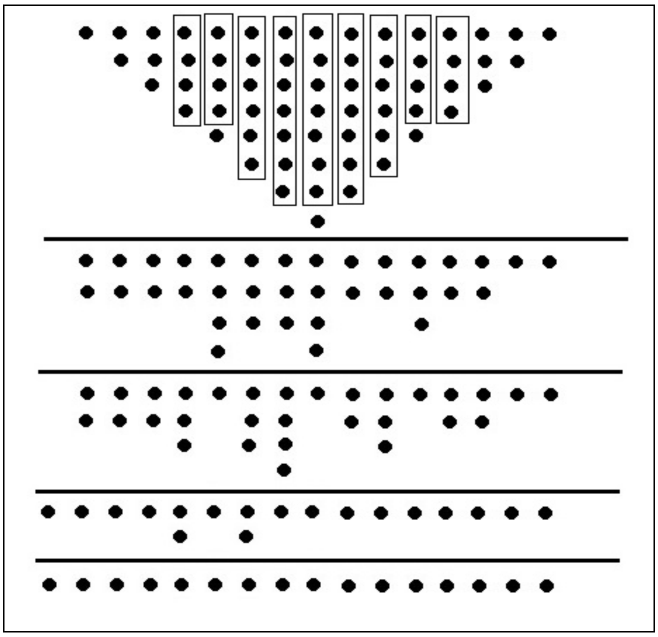


Fig.8. Architecture of the Proposed 8x8 bit Multiplier

. Furthermore, comprehensive analysis depicts that the most significant part of the total delay and power dissipation will belong to this stage in a parallel multiplier. Therefore, the performance enhancement of this stage can significantly improve the speed and lower the power dissipations of the whole system.In this proposed 8 bit multiplier we have to implement the 6:3 counter and 7:3 counter block circuit instead of compressor blocks. By using compressor block critical path delay is reduced. But still consumes more power and area due to the presence of large number of XOR gates. In the conventional compressors based on full adder has a carry rippling effect. The existing 5:2 and 7:2 compressors without full adders reduces the critical path delay but still consumes more power and area due to the presence of large number of XOR gates.

X. PROPOSED 16×16 BIT MULTIPLIER

In order to construct larger, e.g. 16×16 and 32×32, approximate multipliers, the two proposed 8×8 multipliers are combined in an array structure. For instance, to construct an 16×16 multiplier using a 8×8 design, the two 16-bit operands *A* and *B* are partitioned into two 8-bit nibbles, α_H and $\alpha_L$for *A* and β_H and β_L for *B*.α_H and β_H are the 8 MSBs and α_L and β_L indicate the 8 LSBs of *A* and *B*, respectively. Each two of these four nibbles (in total 4 possible combinations) are multiplied using 8×8 multipliers and the partial products are then shifted (based on the nibble's importance) and added together (using a Wallace tree architecture) to produce the final multiplication result. Fig.9.representing the Building 8×8 multipliers using 16×16 multiplier circuit.

Multiplication is a key arithmetic operation that is optimized in digital processors. Parallel multipliers are now extremely used in many high-speed systems. The multiplier lies in the critical path for signal propagation and to decrease the delay and complexity of such systems, parallel accumulation is considered. In a parallel multiplier, the multiplication process is divided into three steps. At first, the partial products (PPs) are generated. Then these products are summed, and the process continues until two rows remain.At the final stage, the two remaining rows will be added by means of, for example, a carry propagation adder.For digital signal processing and applications involving signal and image processing, multipliers and adders form an important part. So, speed of the multipliers and adders affect speed of the operation. For high speed applications, faster multipliers are recommended. Several techniques to increase multiplier speed are proposed.

Multiplication is done by adding partial product terms. Implementation of multiplier comprises three steps:

- Generation of partial products
- Partial products reduction tree
- A vector merge addition to produce final product from the sum and carry rows generated from the reduction tree.

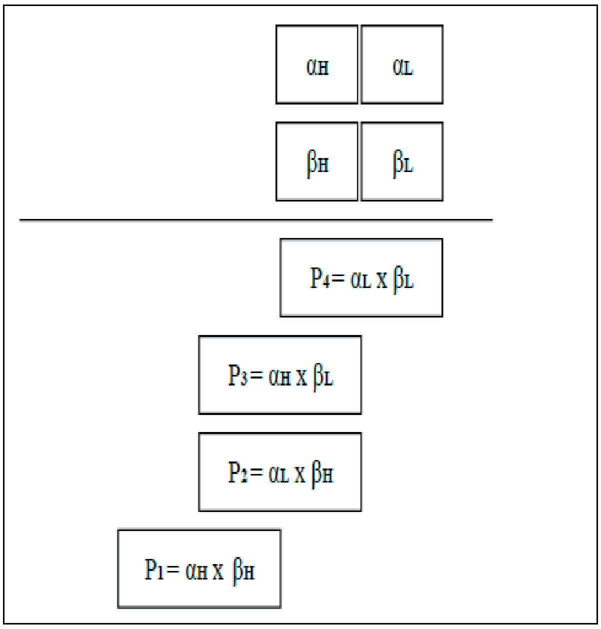
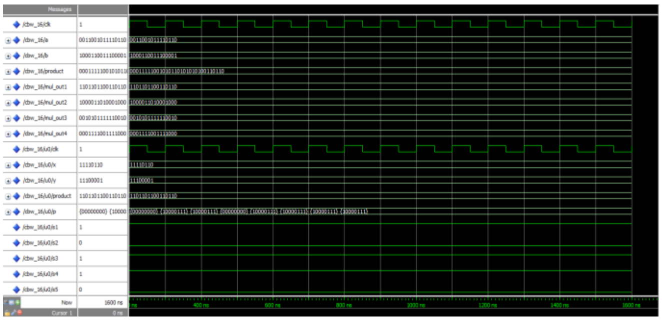


Fig.9.Architecture of the Proposed 16 × 16 bit Multiplier

In this proposed multiplier is the vedic multiplier.The vedic multiplier is the multiplier. Which perform the fastest multiplication adding technique.In this proposed 16 bit multiplier the 16 bits are divided into two 8 bits, then the bits are multiplied. After the partial product generated like P1,P2,P3,P4,The partial product are generated parallelly. Then the product are summed, Finally it will produce the 32 bit outputs. These Hybrid structure helps in reducing total number of logic gates. Parallelism by partitioning bits increases the performance. we have to implement the 6:3 and 7:3 counter circuit in the multiplier.The 6:3 and 7:3 counter circuit reduces the XOR gates compared to existing compressors.

XI. SIMULATION RESULTS AND COMPARISON

The simulation result contains 16 bit input and then we can get the 32 bit output. In this 16x16 mulplier circuit,we have to implement the proposed 6:3 and 7:3 counter blocks in the partial product addition stage. The full adder (FA) acts as the main building block in various multiplication configurations in PPRT. However, the main drawback of FA-based configurations, is the propagation latency for the cascaded cells. An efficient solution to overcome such drawbacks is to utilize a compressor network instead of FA trees, especially in the PPRT.



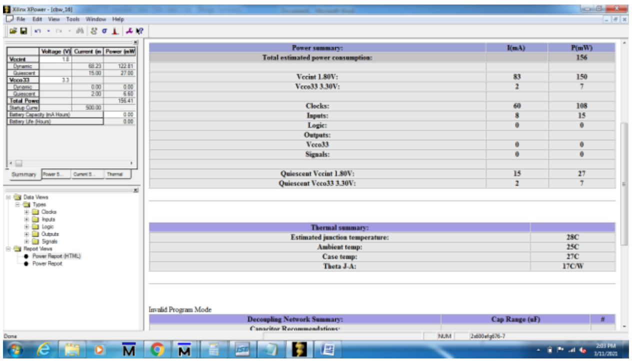
Messages	
/cbw_16/clk	1
/cbw_16/a	0011001011110110
/cbw_16/b	1000110011100001
/cbw_16/product	00011111001010110
/cbw_16/mul_out1	1101101100110110
/cbw_16/mul_out2	1000011010001000
/cbw_16/mul_out3	0010101111110010
/cbw_16/mul_out4	0001111001111000
/cbw_16/u0/clk	1
/cbw_16/u0/x	11110110
/cbw_16/u0/y	11100001
/cbw_16/u0/product	1101101100110110
/cbw_16/u0/p	{00000000} {10000
/cbw_16/u0/s1	1
/cbw_16/u0/s2	0
/cbw_16/u0/s3	1
/cbw_16/u0/s4	1
/cbw_16/u0/s5	0
Now	1600 ns
Cursor 1	0 ns

Fig.10.Simulation result of proposed 16 x 16 bit multiplier

However, the carry rippling will be unfolded as a new problem for such structures. The Existing 5:2 and 7:2 compressors without full adders reduces the critical path delay but still consumes more power and area due to the presence of large number of XOR gates. The proposed 6:3 counter based on bit stacking has no XOR gates on its critical path, it operates nearly 30% faster than all other counter designs. Thus, this novel method of counting via bit stacking allows construction of a counter for a substantial performance increase without increasing power consumption. The Fig.10.shows the simulation result of proposed 16 x 16 bit multiplier..

A.POWER ANALYSIS

Power analysis is the important in the electronic circuit. The power is also measured by the unit of mW.In this we have to measure the power of the proposed multiplier. The power of the proposed multiplier is low when compared to the existing multiplier. The power consumption of the proposed method is 156.41mW. Fig.11.represents the power analysis of proposed multiplier.



Xilinx XPower - [cbw_16]

	Voltage (V)	Current (m	Power (mW
Vccint	1.8		
Dynamic		68.23	122.81
Quiescent		15.00	27.00
Vcco33	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
Total Powe			156.41
Startup Curre		500.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		156
Vccint 1.80V:	83	150
Vcco33 3.30V:	2	7
Clocks:	60	108
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	28C
Ambient temp:	25C
Case temp:	27C
Theta J-A:	17C/W

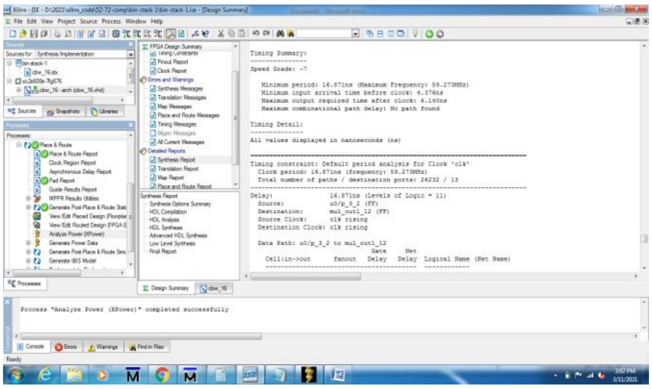
Invalid Program Mode

Decoupling Network Summary:	Cap Range (uF)	#
Capacitor Recommendations:		

Fig.11.Power Analysis of Proposed Multiplier

B.DELAY ANALYSIS

The delay is measured by the unit of ns. Here the Fig 13 shows the delay analysis of proposed multiplier Therefore if the delay is reduced means the speed of the circuit is increased. The delay of the proposed multiplier is low (16.871ns) compared to the existing multiplier. The delay of the circuit is increased means the speed of the circuit is decreased. Delay is the important parameter to measure every electronic circuits. Here we have to measure the delay of proposed multiplier design circuit. The given Fig.12 represents the delay of proposed multiplier design circuit.



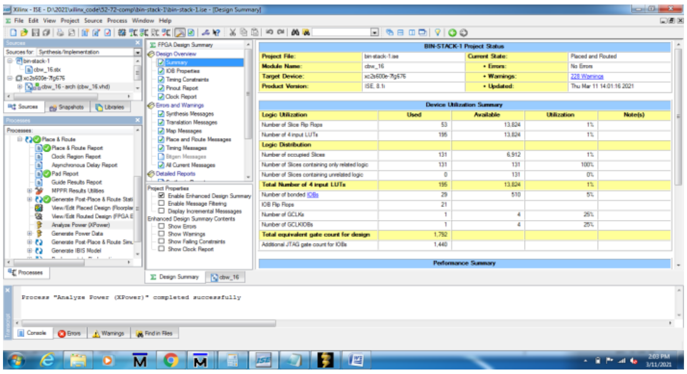
```
Timing Summary:
---------------
Speed Grade: -7

   Minimum period: 16.871ns (Maximum Frequency: 59.273MHz)
   Minimum input arrival time before clock: 4.376ns
   Maximum output required time after clock: 6.140ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 16.871ns (frequency: 59.273MHz)
  Total number of paths / destination ports: 26232 / 13
-------------------------------------------------------------------------
Delay:               16.871ns (Levels of Logic = 11)
  Source:            u0/p_3_2 (FF)
  Destination:       mul_out1_12 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: u0/p_3_2 to mul_out1_12
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
```

```
Process "Analyze Power (XPower)" completed successfully
```

Fig.12. Delay Analysis of Proposed Multiplier

C.AREA ANALYSIS

The area is measured by the unit gate count. Here the Fig.13 shows the design summary of proposed multiplier. Therefore The area of the proposed multiplier (1792) by measuring of gate count. By using Xilinx ISE tool we have to measure area analysis of proposed multiplier.The area analysis is important parameter to measure every electronic circuits.



BIN-STACK-1 Project Status			
Project File:	bin-stack-1.ise	Current State:	Placed and Routed
Module Name:	cbw_16	• Errors:	No Errors
Target Device:	xc2s600e-7fg676	• Warnings:	228 Warnings
Product Version:	ISE, 8.1i	• Updated:	Thu Mar 11 14:01:16 2021

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	53	13,824	1%	
Number of 4 input LUTs	195	13,824	1%	
Logic Distribution				
Number of occupied Slices	131	6,912	1%	
Number of Slices containing only related logic	131	131	100%	
Number of Slices containing unrelated logic	0	131	0%	
Total Number of 4 input LUTs	195	13,824	1%	
Number of bonded IOBs	29	510	5%	
IOB Flip Flops	21			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	1,792			
Additional JTAG gate count for IOBs	1,440			

Performance Summary

```
Process "Analyze Power (XPower)" completed successfully
```

Fig.13. Area Analysis of Proposed Multiplier

D. POWER-DELAY-PRODUCT ANALYSIS

PDP measurements is nothing but, the product of power and delay,Then the result is also called as PDP factor. The unit of power delay product is fJ. The fJ is also called as femtoJoules. 1 femtoJoule is equal to the 10^{-15} Joules. The PDP factor of proposed multiplier (2638.7 x 10^{-15}) is decreased compared to the existing systems.The PDP is explained below,

Power x Delay = power delay product factor
156.41 x 16.871 = 2638.7x 10^{-15}

Therefore the PDP factor is measure by the femto Joule.

Which is also called as fJ.In this power and delay of the proposed method is low. So the power delay product factor also low. If the delay of the circuit is reduced means speed performance of the circuit is increased.

TABLE II

Comparison Between Existing and Proposed Multipliers

Parameters	Power(mW)	Area (Gate count)	Delay(ns)	PDP (fJ(10 $^{-15}$))
Existing Method-1 (16x16 Multiplier)	349.81	5867	18.351	6401.3
Existing Method-2 (16x16 Multiplier)	320.25	5940	17.893	5699.5
Proposed Method (16x16 Multiplier)	156.41	1792	16.871	2638.7

The table II explains the comparison between both existing method and proposed multipliers. In this table the power, area delay and power delay product(PDP) should be mentioned. Therefore, the proposed designs power consumption is 156.41mW.Therefore the proposed multiplier consumes less power compared to the existing multiplier. Then the existing multiplier contains more delay 17.893ns. The delay of the proposed multiplier is low compared to the existing designs. The delay of the proposed designs is 16.871ns respectively. and also the power delay product (PDP) of the proposed designs (2638.7fJ) is less than the existing designs.

XII. CONCLUSION

A new design methodology for area and power reduction using compressor and counter structures has been proposed. The main advantage of the designed structures is the considerable lower area in comparison with previously reported works by minimizing XOR gates through binary stacking. In addition, by considering the reduced carry ripple problem between the cells of a row and total transistor and gate count, it is clear that the PDP proposed architectures are lower than the previous designs.

REFERENCES

[1] Amir Fathi ,Behbood Mashoufi, and Sarkis Azizian, “Very Fast, High-Performance 5-2 and 7-2 Compressors in CMOS Process for Rapid Parallel Accumulations”, IEEE Transactions on Very Large Scale Integration (Vlsi) Systems, vol.28,Issue: 6, June 2020.

[2] M.Sathya and Dr.C.N.Marimuthu, “Very Fast High-Performance 5-2 and 7-2 Compressors in CMOS Process for Rapid Parallel Accumulations” is accepted in International Journal of Intellectual Advancements and Research in Engineering Computations (IJIAREC), ISSN: 2348-2079 Vol.8.Issue-4 pages (759-771),December 2020.

[3] Weiqiang Liu, Senior Member, IEEE, Tingting Zhang, Emma McLarnon, Maire O’Neill, Senior Member, IEEE, Paolo Montuschi, Fellow, IEEE and Fabrizio Lombardi, Fellow, IEEE “Design and Analysis of Majority Logic BasedApproximate Adders and Multipliers”, Ieee Transactions on Emerging Topics In Computing, July 2019.

[4] D. Balobas and N.Konofaos,“Low-power high-performance CMOS 5-2 compressor with 58 transistors,”

Electron. Lett.,vol.54,no.5,pp. 278–280, March 2018.

[5] Christopher Fritz and Adly T. Fam,"Fast Binary Counters Based on Symmetric Stacking" IEEE Transactions on Very Large Scale Integration (Vlsi) Systems,2017.

[6] Suganthi Venkatachalam and Seok-Bum Ko, Senior Member, IEEE "Design of Power and Area Efficient Approximate Multipliers", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, December 19, 2016.

[7] P.Saha,P.Samanta, and D.Kumar,"4:2 and 5:2 decimal compressors,"in Proc. 7th Int. Conf. Intell. Syst., Modelling Simulation (ISMS),Jan. 2016, pp. 424-429.

[8] Srinivasan Narayanamoorthy, Hadi Asghari Moghaddam,Zhenhong Liu, Taejoon Park, and Nam Sung Kim "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems June 5, 2014.

[9] A. Momeni, J. Han, Member, P.Montuschi, Senior Member and F. Lombardi, Fellow "Design and Analysis of Approximate Compressors for Multiplication", IEEE Transactions On Computers, 2013

[10] A. Najafi, S. Timarchi, and A. Najafi, "High-speed energy-efficient 5:2 compressor," in Proc. 37th Int. Conv. Inf. Commun. Technol., Electron. Micro electron., Opatija, Croatia, May 2014, pp. 80–84.

[11] M. Tohidi, M. Mousazadeh, S. Akbari, K. Hadidi, and A. Khoei, "CMOS implementation of a new high speed, glitch-free 5-2 compressor for fast arithmetic operations," in Proc. 20th Int. Conf. Mixed Design Integr.Circuits Syst. (MIXDES), Jun. 2013.

[12] S. Mehrabi, K. Navi, and O. Hashemipour, "Performance comparison of high-speed high-order (n:2) and (n:3) CNFET-based compressors," Int.J. Model. Optim., vol. 3, no. 5, pp. 432–435, Oct. 2013.

[13] C. Pan, Z. Wang, and C. Sechen, "High speed and power efficient compression of partial products and vectors," J. Algorithms Optim., Oct.,vol. 1, no. 1, pp. 39–54, 2013.

[14] A. Fathi, S. Azizian, K. Hadidi, A. Khoei, and A. Chegeni, "CMOS implementation of a fast 4-2 compressor for parallel accumulations," in Proc. IEEE Int. Symp. Circuits Syst., pp. 1476–1479,May 2012.

[15] G. Caruso and D. Di Sclafani, "Analysis of compressor architectures in MOS current-mode logic," in Proc. 17th IEEE Int. Conf. Electron., Circuits Syst., Dec. 2010, pp.13–16

[16] W. Ma and S. Li, "A new high compression compressor for large multiplier,"in Proc. 9th Int. Conf. Solid-State Integrated-Circuit Technol.,Oct. 2008, pp. 1877–1880.

[17] M. Rouholamini, O. Kavehie, A.-P. Mirbaha, S. J. Jasbi, and K. Navi, "A new design for 7:2 compressors," in Proc. IEEE/ACS Int. Conf. Comput.Syst. Appl. (AICCSA), May 2007, pp. 474–478

[18] G. Yang, S.-O. Jung, K.-H. Baek, S. Hwan Kim, S. Kim, and M. Kang, "A 32-bit carry look ahead adder using dual-path all-N logic," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 8,pp. 992–996, Aug. 2005.

[19] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,Feb. 2002.

[20] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in Proc. Conf. Rec. 25th Asilomar Conf. Signals, Syst. Comput., Nov. 2001,pp. 129–133.

[21] A Deepa,C.N.Marimuthu, "High speed VLSI architecture for squaring binary numbers using Yavadunam sutra and bit reduction technique" International Journal of Applied Engineering Research,vol .13,issue 6 Pages 4471-4474.

[22] A Deepa,C.N.Marimuthu,C.Murugesan"An efficient high speed squaring and multiplier architecture using yavadunam sutra and bit reduction technique" Journal of Physics: Conference series,vol.1432,issue 1,Pages 012080.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING