International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 2, April 2023, pp. 1367~1378
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i2.pp1367-1378

Delay-efficient 4:3 counter design using two-bit reordering circuit for high-speed Wallace tree multiplier

Madaka Venkata Subbaiah, Galiveeti Umamaheswara Reddy
Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Sri Venkateswara University, Andhra Pradesh, India

Article Info

Article history:

Received Apr 11, 2022
Revised Oct 15, 2022
Accepted Nov 1, 2022

Keywords:

16-bit multiplier
4:3 counter
7:3 counter
8-bit multiplier
Full adder
Half adder

ABSTRACT

In many signal processing applications, multiplier is an important functional block that plays a crucial role in computation. It is always a challenging task to design the delay optimized multiplier at the system level. A new and delay-efficient structure for the 4:3 counter is proposed by making use of a two-bit reordering circuit. The proposed 4:3 counter along with the 7:3 counter, full adder (FA), and half adder (HA) circuits are employed in the design of delay-efficient 8-bit and 16-bit Wallace tree multipliers (WTMs). Using Xilinx Vivado 2017.2, the designed circuits are simulated and synthesized by targeting the device 'xc7s50fgga484-1' of Spartan 7 family. Further, in terms of lookup table (LUT) count, critical path delay (CPD), total on-chip power, and power-delay-product (PDP), the performance of the proposed multiplier circuit is compared with the existing multipliers.

Corresponding Author:

Madaka Venkata Subbaiah
Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Sri Venkateswara University
Tirupati, Andhra Pradesh, India-517502
Email: venkatmadaka@gmail.com

1. INTRODUCTION

Multiplication is the prominent operation in many signal and image processing applications such as filtering and convolution [1]–[3]. In the literature, there are many techniques to perform the multiplication operation. All the techniques that had been proposed for multiplication have been classified into two categories such as combinational and sequential schemes. The sequential multiplier [4] requires less hardware but more time to complete the computation. Also, its energy consumption is more. An N-bit sequential multiplier requires N clock cycles to complete the computation. In each cycle the circuit consumes power P and offers a delay of t then the energy consumption of the circuit in each cycle is Pt. The total energy consumption of N-bit sequential multiplier is NPt. Hence it consumes more energy even it has less hardware. However, the combinational multiplier requires more hardware and less time to complete the computation as compared to the sequential multiplier since all the partial products (PPs) are simultaneously generated here. Typically, the two groups of binary bits in combinational multiplier are multiplied in three steps. The PP bits are computed in step 1 while in step 2, the PP bits are accumulated hence this step is named as PP reduction (PPR) stage. Finally, the resultant product is obtained in step 3. A high-speed multiplier is possible to design by decreasing the delay incurred by the PPR stage. There are several multiplier types such as array multiplier (AM) [4], [5], carry save multiplier (CSM) [6], [7], Wallace tree multiplier (WTM) [8]–[10], Vedic multiplier (VM) [11]–[15] and many others [16], [17] comes under the category of combinational multipliers. Among all, WTM and VM are flexible in structure hence huge research is going on them. In AM, all the PPs are accumulated one after the other to get the final product,

hence it provides huge delay. An N-bit AM requires (N-1) number of N-bit parallel adders. In CSM, the carry save addition (CSA) process is used to accumulate the PPs i.e. the carry generated in j^{th} bit location of step k is given to the $(j+1)^{th}$ bit location of step $(k+1)$ and finally a ripple carry adder (RCA) is used to generate the final product. An N-bit CSM has (N-1) stages of CSA and finally (N-1)-bit RCA. The CSM offers less delay when compared to AM since it employs CSA process in PPR stage. Both AM and CSM are of row reduction type multipliers. The multiplier designed by Urdhva Tiryagbhyam Vedic sutra is flexible in structure hence it can be designed by many ways. An N-bit VM requires four (N/2)-bit multipliers and three N-bit parallel adders. In VM both step 2 and 3 are clubbed together to attain the final result.

In the literature, several techniques are proposed for the reduction of PPs for WTM. Out of all, the counters and compressors [18]–[20] are popular. The notable distinction between compressor and counter is that the weights of carry bits C_1 and C_2 produced by them. For example, a 3:2 compressor will be designed using one full adder (FA) and one half adder (HA) as shown in Figure 1(a). The 3:2 compressor accepts three binary bits A_2, A_1, and A_0 from a particular bit position along with carry input C_{in} from previous lower significant compressor and it produces two output bits S and C_1 along with carry output C_2 to the compressor in the next significant stage. The relation between inputs and outputs of the 3:2 compressor is expressed as in (1) [21].

$$A_2 + A_1 + A_0 + C_{in} = 2^1.(C_2 + C_1) + 2^0 . S \quad (1)$$

Figure 1(b) shows the block diagram of 4:3 counter [22]. It consists of one FA and two HAs. The relation between inputs and outputs of the 4:3 counter is expressed as in (2) [23].

$$A_4 + A_3 + A_2 + A_1 = 2^2.C_2 + 2^1.C_1 + 2^0.S \quad (2)$$

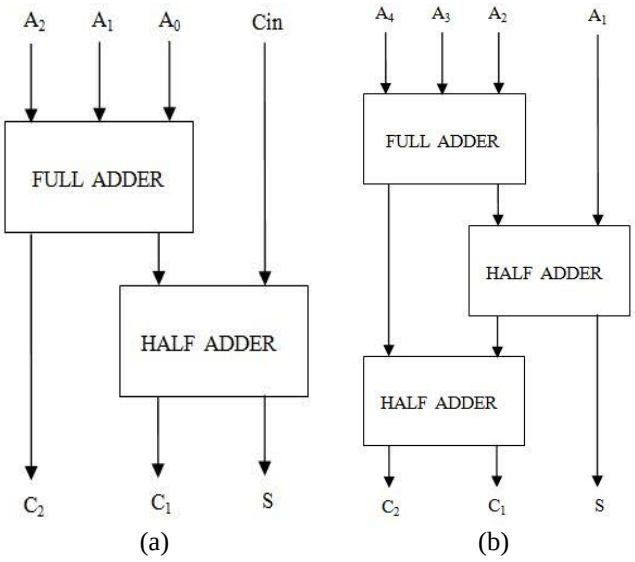


Figure 1. Block diagram of (a) 3:2 compressor and (b) 4:3 counter [22]

From (1) and (2), it is observed that the weights of carry bits C_1 and C_2 are same for compressor but is different for counter circuit. However, the functionality of both the circuits are exactly same i.e. both the circuits are used to count the number of 1's present in accepted inputs. For instance, consider $A_2=1$, $A_1=1$, $A_0=1$ and $C_{in}=1$ as inputs for 3:2 compressor then it produces the result as $\{C_2, C_1, S\}=110$ (the equivalent decimal value is four based on (1)) and consider $A_4=1$, $A_3=1$, $A_2=1$ and $A_1=1$ as inputs for 4:3 counter that will produce the result as $\{C_2, C_1, S\}=100$ (value is four as per the (2)).

This paper proposes a new and delay-efficient structure for 4:3 counter using two-bit reordering circuit. The proposed circuit requires less number of gates for its implementation hence it occupies less area. The delay-efficient 8 and 16-bit multipliers are designed using the 7:3 counter along with the proposed 4:3 counter, FA and HA.

The rest of the paper is sequenced as follow. The literature review of 4:3 counter and other counters is described in section 2. Section 3 presents information about the proposed 4:3 counter, 7:3 counter, and finally methodology of the proposed multiplier design. Section 4 describes the simulation and synthesis results and followed by conclusion is given in section 5.

2. LITERATURE REVIEW

The basic 4:3 counter [22] is implemented using two HAs and one FA as shown in Figure 1(b). Each HA is designed by one EX-OR gate and one AND gate while FA is designed by two EX-OR gates, two AND gates and one OR gate. As a whole, 4:3 counter is implemented with one OR gate, four AND gates, and four EX-OR gates. The basic 4:3 counter offers more delay since there are three EX-OR gates in series between inputs and sum (S) output. Asif and Kong [24] have proposed 6:3, 5:3 and 4:3 counters using carry generate (G) and carry propagate (P) blocks for the implementation of WTM of various bit sizes. Figure 2 shows the 4:3 counter designed by them. It has only two EX-OR gates in series between inputs and sum (S) output and hence, it offers less delay than the basic 4:3 counter. Figure 3 illustrates the 3-1-1-2 counter proposed by Sivanandam and Kumar [25], is designed with 4X1 multiplexers. Authors have proposed it for the implementation of modified VM.

Fritz and Fam [26] have proposed 6:3 and 7:3 counters that uses 3-bit stacking circuits. The circuits offer less delay than the existing designs. However, the circuit complexity and occupied area are more. Krishna *et al*. [23] have designed a novel 5:3 counter, thereby designed 15:4 counter using 5:3 counter. The designed 5:3 and 15:4 counters are used in the implementation of 16-bit multiplier. The designed 16-bit multiplier offers less delay but occupies large area. However, higher bit counters would be helpful in diminishing the number of stages required for computation. The 16-bit WTM were designed using 7:3 and 5:3 counters in [27] that offer more delay and occupied less area.

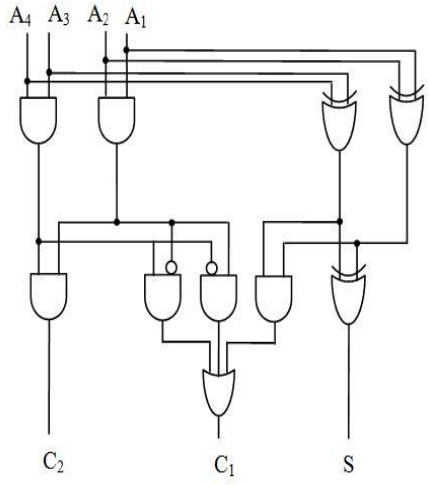


Figure 2. Logic diagram of 4:3 counter [24]

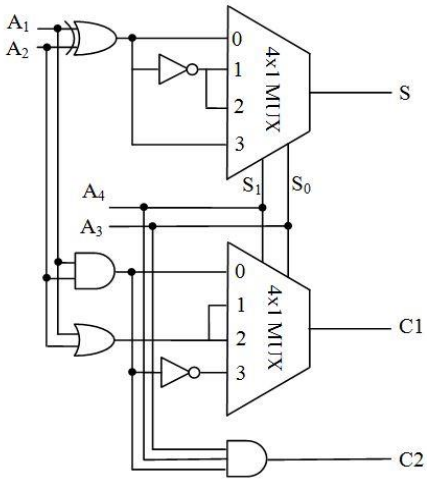


Figure 3. Logic diagram of 3-1-1-2 counter [25]

3. PROPOSED HIGH SPEED WALLACE TREE MULTIPLIER

This section presents the details of the proposed 4:3 counter design, Elmore delay estimation of 4:3 counter, 7:3 counter design. Also, the design methodology that is the number of adder elements such as FA, HA, 4:3 counter, and 7:3 counters. That used in different stages of computation of the 16-bit proposed WTM is described.

3.1. Proposed 4:3 counter

The functionality of the 4:3 counter is to calculate the number of 1's present in the accepted four binary bits from a particular bit position and it produces three binary bits as output. The inputs are represented as A_4, A_3, A_2, and A_1 while the outputs are represented as C_2, C_1, and S. The output bit S is given into the same bit position while the C_2 and C_1 outputs are given into the next two immediate bit positions of a particular bit position from where the inputs are accepted. The Boolean functions for S, C_1, and C_2 can be written as in (3)-(5) respectively.

$$S = A_4 \oplus A_3 \oplus A_2 \oplus A_1 \quad (3)$$

$$C_1 = A_4^1 A_3^1 (A_2 A_1) + A_4^1 A_3 (A_2 + A_1) + A_4 A_3^1 (A_2 + A_1) + A_4 A_3 (A_2 A_1)^1 \quad (4)$$

$$C_2 = A_4 A_3 A_2 A_1 \quad (5)$$

The proposed 4:3 counter uses a two-bit reordering circuit [28] on the input side. The two-bit reordering circuit will have two inputs designated by X_2, X_1, and two outputs designated by Y_2, Y_1. The circuit rearranges its inputs as all 1's to the left followed by all 0's in the order, and the relation between its inputs and outputs is given by the Table 1.

Table 1. Functional table of two-bit reordering circuit

Inputs		Outputs	
X_2	X_1	Y_2	Y_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

The Boolean functions of Y_2 and Y_1 are given by (6) and (7) respectively.

$$Y_2 = X_2 + X_1 \quad (6)$$

$$Y_1 = X_2 X_1 \quad (7)$$

Using (6) and (7), the (3) can be written as given by (8).

$$S = (A_4 + A_3)(A_4 A_3)^1 \oplus (A_2 + A_1)(A_2 A_1)^1 \quad (8)$$

The logic diagram of proposed 4:3 counter is illustrated in Figure 4.

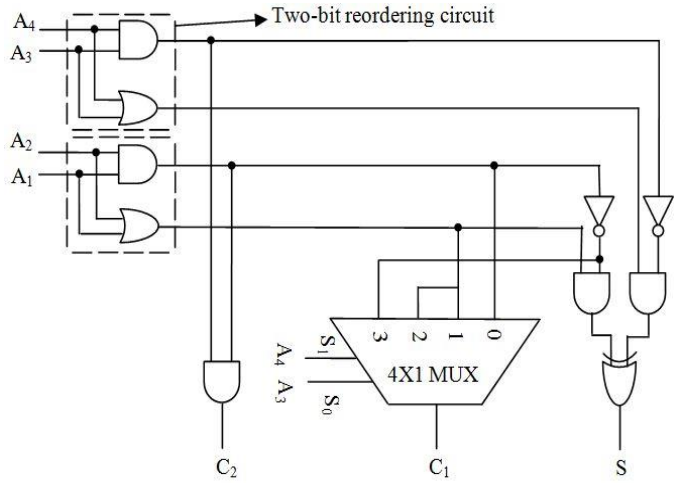


Figure 4. Logic diagram of proposed 4:3 counter

The delay introduced by the proposed 4:3 counter from its inputs to outputs S, C_1, and C_2 is given by (9)-(11) respectively.

$$\tau_S = \tau_{OR} + \tau_{AND} + \tau_{XOR} \quad (9)$$

$$\tau_{C1} = \tau_{OR} + \tau_{MUX} \quad (10)$$

$$\tau_{C2} = 2\,\tau_{AND} \quad (11)$$

Where, τ_S, τ_{C1}, and τ_{C2} represent the delay offered by the proposed 4:3 counter from its inputs to S, C_1, and C_2 outputs respectively, τ_{OR}, τ_{AND}, and τ_{XOR} represent the delay introduced by 2-input OR, AND, and EX-OR gates from its inputs to output, and τ_{MUX} is the delay introduced by 4X1 multiplexer from its inputs to output.

3.1.1. Elmore delay estimation of 4:3 counter

The propagation delay (PD) from a source node *s* to a node *i* in a circuit is given by (12) [29].

$$t_{pd} = \sum_i R_{is} C_i \quad (12)$$

Where R_{is} is the resistance on the desired path from source node *s* to node *i* and C_i is the capacitance at the node *i*. For example, let us consider the circuit of 4-input NAND gate [29] as depicted in Figure 5. The PD of 4-input NAND gate can be estimated for both falling and rising output by taking the equivalent RC circuit as shown in Figures 6(a) and 6(b) respectively. For the rising output, in the worst-case, the three inner inputs A, B, C are one and the outer input D is zero. Then the output Y is pulled up to V_{DD} through a single pMOS on transistor and the nMOS on transistors provide parasitic capacitance that slows down the transition.

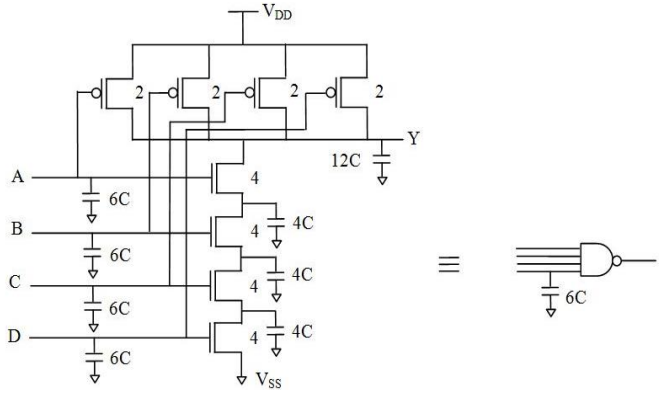


Figure 5. Circuit diagram of 4-input NAND gate and its logic symbol with input capacitance

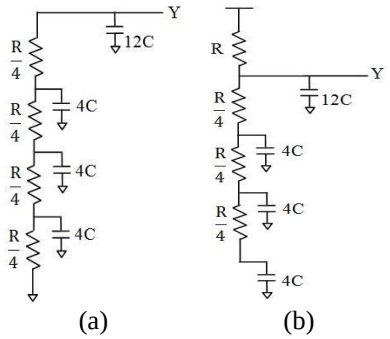


Figure 6. Equivalent RC circuit of 4-input NAND gate for (a) falling and (b) rising output

For the falling output, the PD is calculated as the sum of product of resistance R on the desired path from source node to a node *i* and capacitance *C* at node *i* of each node. It is given by (13).

$$t_{pdf} = \frac{R}{4}(4C) + \left(\frac{R}{4} + \frac{R}{4}\right)(4C) + \left(\frac{R}{4} + \frac{R}{4} + \frac{R}{4}\right)(4C) + \left(\frac{R}{4} + \frac{R}{4} + \frac{R}{4} + \frac{R}{4}\right)(12C)$$
$$t_{pdf} = 18RC \quad (13)$$

For the rising output, the PD is calculated as the sum of product of resistance R on the desired path from source node to a node *i* and capacitance *C* at node *i* of each node. It is given by (14).

$$t_{pdr} = R\,(12C) + R\,(4C) + R(4C) + R(4C)$$
$$t_{pdr} = 24RC \quad (14)$$

Figure 7 depicts the logic diagram of 4:3 counter with the details of falling output delay t_{pdf} and rising output delay t_{pdr} of each gate along with input capacitance. The input capacitance is same on all the inputs of a gate. From the figure, the worst-case falling output delay and rising output delay of the outputs C_2, C_1, and S of 4:3 counter is given by the (15)-(20) respectively. In Table 2, the number of gates used in proposed 4:3 counter is compared with existing 4:3 counter designs.

$$t_{pdf-C2} = t_{pdr} \text{ of NAND} + t_{pdf} \text{ of NOR} = 25\ RC + 10\ RC$$
$$t_{pdf-C2} = 35\ RC \quad (15)$$

$$t_{pdr-C2} = t_{pdf} \text{ of NAND} + t_{pdr} \text{ of NOR} = 24\ RC + 8\ RC$$
$$t_{pdr-C2} = 32\ RC \quad (16)$$

$$t_{pdf-C1} = t_{pdr} \text{ of NAND} + t_{pdf} \text{ of NOT} + t_{pdr} \text{ of NAND} + t_{pdf} \text{ of NAND}$$
$$t_{pdf-C1} = 25\ RC + 8\ RC + 21\ RC + 18\ RC$$
$$t_{pdf-C1} = 72\ RC \quad (17)$$

$$t_{pdr-C1} = t_{pdf} \text{ of NAND} + t_{pdr} \text{ of NOT} + t_{pdf} \text{ of NAND} + t_{pdr} \text{ of NAND}$$
$$t_{pdr-C1} = 24\ RC + 8\ RC + 18\ RC + 24\ RC$$
$$t_{pdr-C1} = 74\ RC \quad (18)$$

$$t_{pdf-S} = t_{pdf} \text{ of NOR} + t_{pdr} \text{ of NOT} + t_{pdr} \text{ of NAND} + t_{pdf} \text{ of NOT} + t_{pdf} \text{ of XOR}$$
$$t_{pdf-S} = 13\ RC + 17\ RC + 11\ RC + 12\ RC + 34\ RC$$
$$t_{pdf-S} = 87\ RC \quad (19)$$

$$t_{pdr-S} = t_{pdf} \text{ of NOR} + t_{pdr} \text{ of NOT} + t_{pdr} \text{ of NAND} + t_{pdf} \text{ of NOT} + t_{pdr} \text{ of XOR}$$
$$t_{pdr-S} = 13\ RC + 17\ RC + 11\ RC + 12\ RC + 33\ RC$$
$$t_{pdr-S} = 86\ RC \quad (20)$$

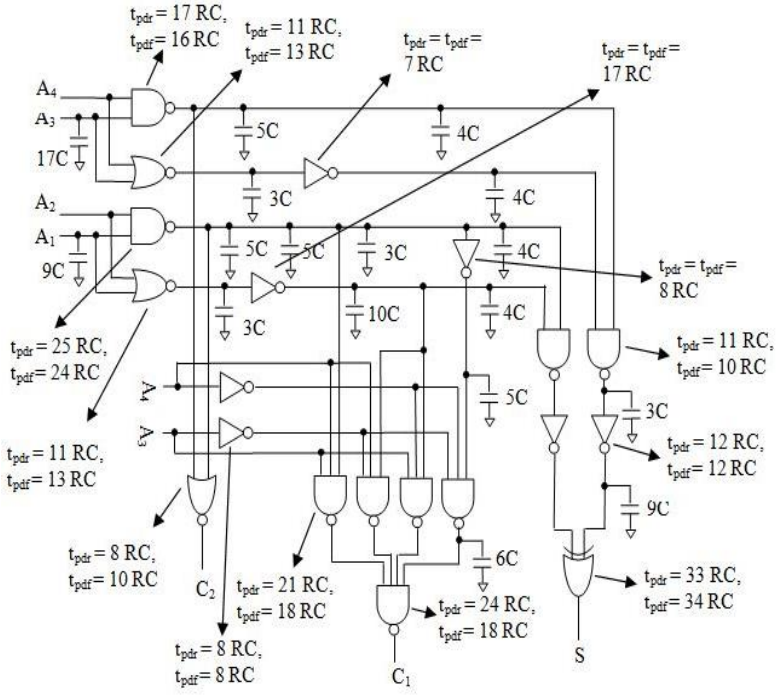


Figure 7. Logic diagram of 4:3 counter with rising and falling output delays of each gate

Table 2. Gate utilization of different 4:3 counters

4:3 Counter	2-input EX-OR	2-input AND	2-input OR	NOT	4X1 MUX
Conventional [22]	4	4	1	-	-
[24]	3	6	2	2	-
[25]	1	3	1	2	2
Proposed	1	5	2	2	1

3.2. 7:3 counter

A 7:3 counter [18], [27] is a circuit that calculates the number of 1's present in seven binary bits which are accepted from a particular bit position and it produces three binary bits as output. The output value ranges from zero to seven when the circuit's inputs range from all 0's to all 1's respectively. The inputs are represented as A_7, A_6, A_5, ..., A_1 while the outputs are represented as C_2, C_1, and S. The output bit S is given into the same bit position while the C_2 and C_1 outputs are given into the next two immediate bit positions of a particular bit position from where the inputs are accepted. The Boolean functions for S, C_1, and C_2 are given by (21)-(23) respectively. Figure 8 depicts the logic diagram of the 7:3 counter.

$$S = x_1 \oplus x_3 \oplus A_4 \tag{21}$$

$$C_1 = x_2 \oplus x_4 \oplus x_5 \tag{22}$$

$$C_2 = x_4 x_5 + (x_4 + x_5) x_2 \tag{23}$$

Where the expressions for x_1, x_2, x_3, x_4, and x_5 are given by (24)-(28) respectively.

$$x_1 = A_7 \oplus A_6 \oplus A_5 \tag{24}$$

$$x_2 = A_6 A_5 + (A_6 + A_5) A_7 \tag{25}$$

$$x_3 = A_3 \oplus A_2 \oplus A_1 \tag{26}$$

$$x_4 = A_2 A_1 + (A_2 + A_1) A_3 \tag{27}$$

$$x_5 = x_3 A_4 + (x_3 + A_4) x_1 \tag{28}$$

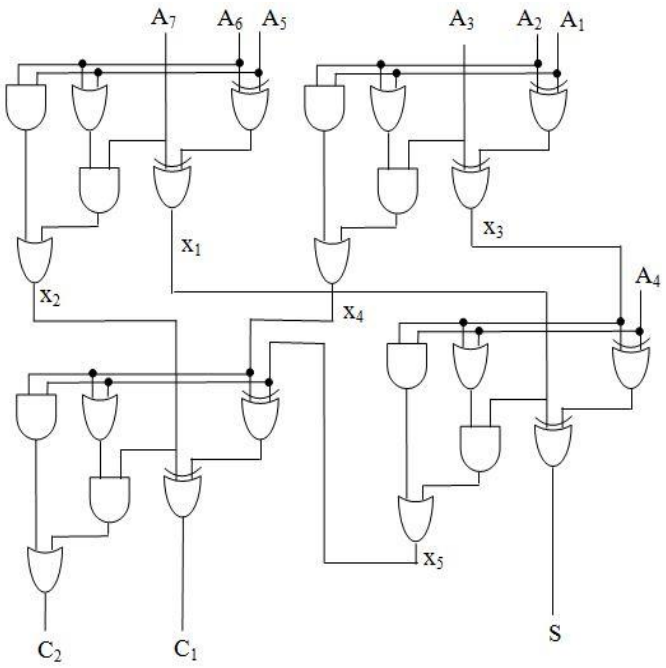


Figure 8. Logic diagram of 7:3 counter [18], [27]

3.3. Design methodology of proposed 16-bit multiplier

Figure 9 shows the design of 16-bit WTM using 7:3 counter, 4:3 counter along with FA and HA. In the diagram, each dot represents a single binary bit. The designed circuit has a depth of five stages, where in ten 7:3 counters, eight 4:3 counters, forty-four FAs and one HA is employed in the first stage of computation. In stage 1, the 7:3 counter at 13th column takes seven bits and it produces three output bits C_2, C_1 and S. The S bit is fed in to the same column and C_1, C_2 are fed in to the 14th and 15th column respectively. Similarly, the 4:3 counter at 14th column takes four bits and it produces three output bits C_2, C_1 and S. The S bit is fed in to the same column and the remaining C_1, C_2 bits are fed in to the column 15 and 16 respectively. In the same fashion, the circuits used in remaining columns will perform the computation. Twelve 7:3 counters, three 4:3 counters, fourteen FAs and three HAs are used in the second stage of computation. The third stage uses twenty-five FAs and one HA as well as the fourth stage uses Eighteen FAs and nine HAs to complete the computation. After the fourth stage of computation, the result has only two rows of binary bits. By adding these two rows of binary bits, the final product is obtained. In each stage, a circuit is selected and used in every column of the multiplier in such a way that to reduce the number of stages in the computation. In each stage, the right most single bits represented with black colored dots are the computed final product bits and the bits which are not taken for computation are represented with red colored dots.

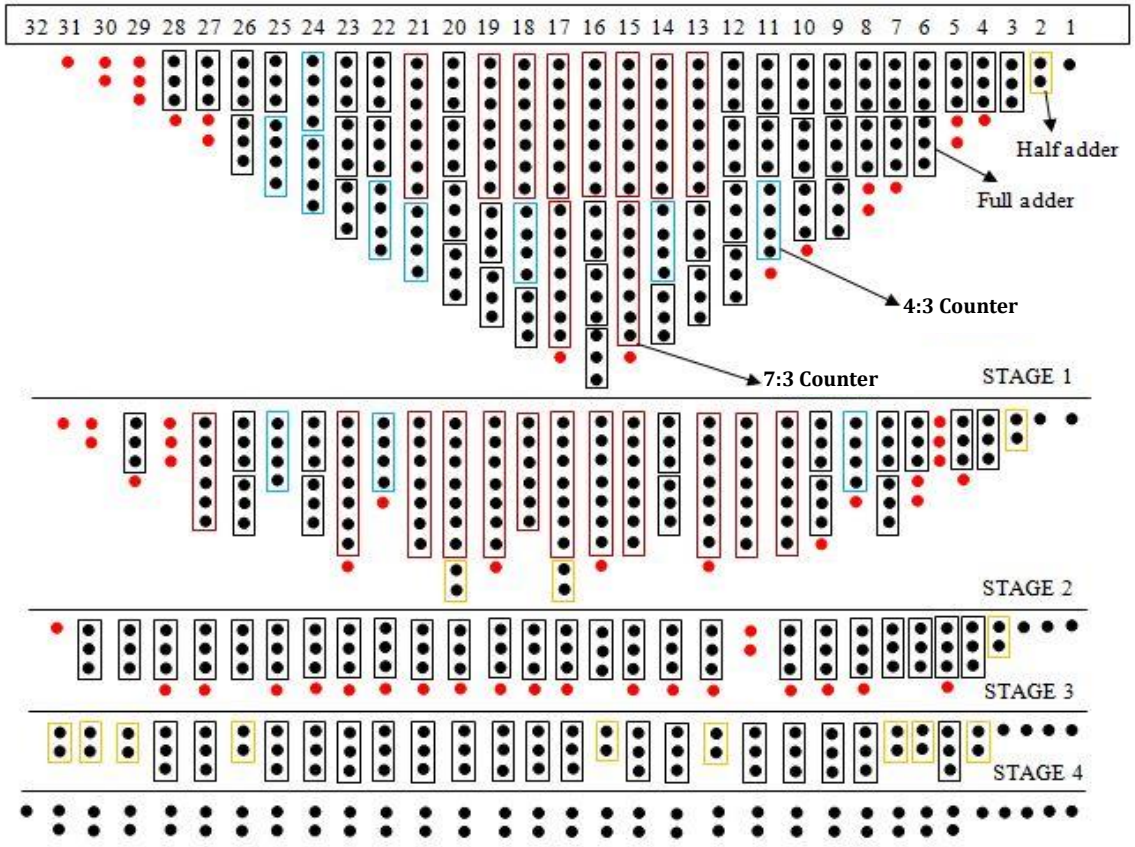


Figure 9. 16-bit WTM using proposed 4:3 counter, 7:3 counter, FA and HA

4. RESULTS AND DISCUSSION

All the proposed circuits and existing designs are programmed in Verilog hardware description language (HDL) gate level modeling and verified their functional behavior by a test bench having a few particular input test vectors. The simulation result of proposed 16-bit multiplier is depicted in Figure 10. By targeting the device 'xc7s50fgga484-1' of Spartan-7 family, the designed circuits were synthesized using Xilinx Vivado 2017.2.

The synthesized results i.e. LUT count, CPD in *ns*, and power consumption in Watt of the existing and proposed 4:3 counter are shown in Table 3. The LUT count denotes the circuit area occupied on the device and CPD denotes the delay of the longest path of the circuit. The total on-chip power includes static

and dynamic power. The dynamic power is a combination of logic, signal and I/O power, which depends on the switching activity of the input test vectors that are applied to the circuit as well as the internal signals and the resultant outputs. The power consumption values are obtained from the tool based on the default settings such as default toggle rate = 12.5, default static probability = 0.5, and some other parameters that are as shown in Figures 11(a) and 11(b).

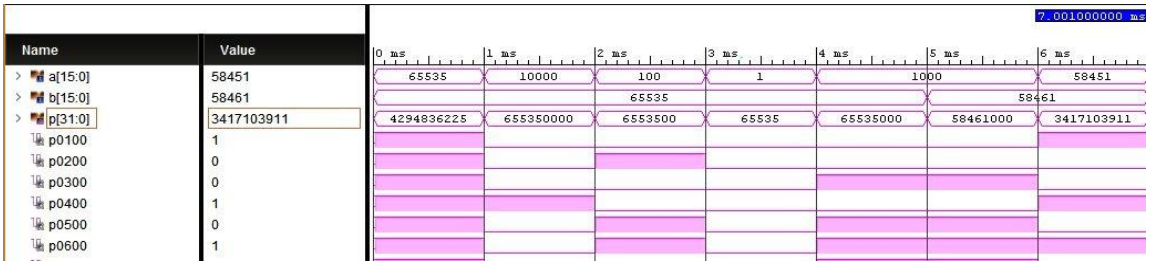
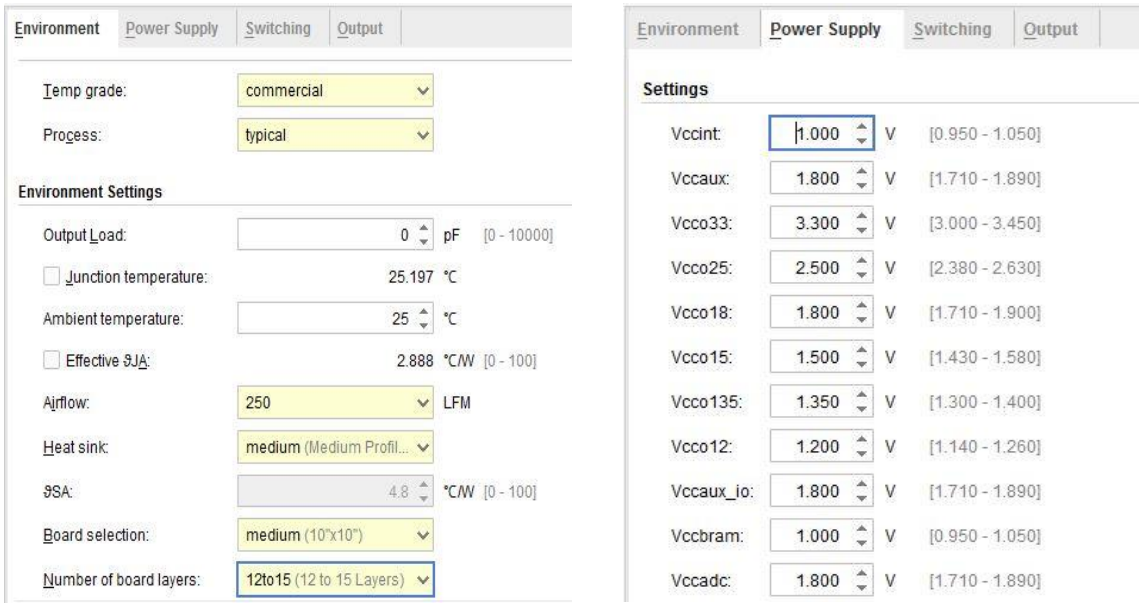


Figure 10. Simulation result of proposed 16-bit multiplier

Table 3. Performance metrics of existing and proposed 4:3 counters

4:3 Counter	Number of LUTs	CPD (ns)	Power Consumption (W)		
			Static Power	Dynamic Power	Total On-chip power
Conventional [22]	2	7.319	0.072	1.195	1.266
[24]	2	7.286	0.072	1.195	1.267
[25]	3	7.074	0.072	1.197	1.269
Proposed	2	7.178	0.072	1.196	1.268



(a) (b)

Figure 11. Default settings of the tool to obtain the power report (a) environment and (b) power supply

The dynamic power dissipation [29] in a CMOS circuit is caused due to the charging and discharging the load capacitance as well as due to the short circuit current flowing when both pMOS and nMOS subcircuits are partially on. The former one is called as switching power and the latter one is called as short circuit power. Thus, dynamic power dissipation is given by (29).

$$P_{dynamic} = P_{switching} + P_{short-circuit} \quad (29)$$

Where $P_{dynamic}$, $P_{switching}$, and $P_{short\ circuit}$ are the dynamic, switching and short circuit powers respectively.

The short circuit power is normally less than 10% of the total dynamic power [29]. The switching power is calculated by (30).

$$P_{switching} = \alpha C_L V_{DD}^2 f \quad (30)$$

Where α is the activity factor, C_L is the load capacitance, V_{DD} is the supply voltage, and f is the clock frequency.

From Table 3, it is noticed that the CPD of proposed 4:3 counter is reduced by 1.93% and 1.48% as well as increased by 1.47% as compared to conventional [22], [24], [25] 4:3 counters respectively. However, the delay to other outputs of proposed 4:3 counter is less than 4:3 counter [25]. The LUTs occupied by proposed 4:3 counter is same as [22], [24] but reduced by 33.3% when compared to [25] 4:3 counter. The power consumption of the proposed 4:3 counter is almost same as compared to the existing 4:3 counter designs.

The synthesized results of the proposed 8 and 16-bit multipliers are compared with the existing multiplier designs in Tables 4 and 5 respectively. One more parameter i.e. PDP is also included in the tables to determine the effectiveness of proposed 8 and 16-bit multipliers when compared with existing designs. The tables also include percentage variation of the parameters of existing multiplier designs w.r.t. the WTM using counters (proposed 4:3 counter).

Table 4. Performance comparison of proposed 8-bit multiplier with existing multipliers

Type of 8-bit Multiplier	Number of LUTs	CPD (ns)	%reduction of CPD	Total On-chip Power (W)	%reduction of total On-chip power	PDP (nJ)	%reduction of PDP
AM [5]	129	18.33	22.97	14.837	6.48	271.96	27.96
Conv. CSM [7]	81	16.21	12.89	13.737	-1.00*	222.68	12.02
WTM [5]	99	14.80	4.59	13.911	0.26	205.88	4.84
WTM using counters [27]	95	14.74	4.21	13.845	-0.22*	204.08	4.00
WTM using counters (Conv. 4:3 counter [22])	97	14.96	5.61	13.806	-0.50*	206.54	5.15
WTM using counters (4:3 counter [24])	97	14.87	5.04	13.889	0.10	206.53	5.14
WTM using counters (3-1-1-2 counter [25])	100	14.46	2.35	13.848	-0.19*	200.24	2.16
WTM using counters (Proposed 4:3 counter)	98	14.12	-	13.875	-	195.91	-

*Negative sign indicates the %increment of that parameter of the WTM using counters (Proposed 4:3 counter) w.r.t. the existing designs.

Table 5. Performance comparison of proposed 16-bit multiplier with existing multipliers

Type of 16-bit Multiplier	Number of LUTs	CPD (ns)	%reduction of CPD	Total On-chip Power (W)	%reduction of total On-chip power	PDP (nJ)	%reduction of PDP
AM [5]	600	36.08	40.52	48.488	20.92	1749.45	52.97
Conv. CSM [7]	550	27.35	21.53	45.526	15.78	1245.14	33.92
WTM [5]	437	23.33	8.01	40.006	4.16	933.34	11.84
WTM using counters [27]	369	22.48	4.54	38.455	0.29	864.47	4.82
WTM using counters (Conv. 4:3 counter [22])	386	22.15	3.11	38.298	-0.11*	848.30	3.00
WTM using counters (4:3 counter [24])	386	21.81	1.60	38.612	0.70	842.13	2.29
WTM using counters (3-1-1-2 counter [25])	383	21.75	1.33	38.232	-0.29*	831.55	1.05
WTM using counters (Proposed 4:3 counter)	382	21.46	-	38.342	-	822.82	-

*Negative sign indicates the %increment of that parameter of the WTM using counters (Proposed 4:3 counter) w.r.t. the existing designs.

5. CONCLUSION

In this paper, a new and delay-efficient structure is proposed for the 4:3 counter using two-bit reordering circuit. The 8 and 16-bit WTMs are designed using the proposed 4:3 counter along with 7:3 counter, FA, and HA. All the designed circuits are coded in Verilog HDL and their functionality is verified. Using Xilinx Vivado 2017.2, the designed circuits are synthesized by targeting the device 'xc7s50fgga484-1' of Spartan 7 family. From the results, it is noted that the 8 and 16-bit multipliers designed using the proposed

4:3 counter along with other circuits offer less CPD when compared to existing multiplier designs. Using the proposed 4:3 counter, it would be possible to design a delay-efficient 12:4 counter that will be helpful in designing the higher bit multipliers with fewer stages in computation.

REFERENCES

[1] J. G. Proakis, *Digital signal processing-principles, algorithms and applications*. Pearson Education India, 2001.
[2] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 3rd ed. Pearson Education International, 2008.
[3] S. L. Freeny, "Special-purpose hardware for digital filtering," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 633–648, 1975, doi: 10.1109/PROC.1975.9797.
[4] M. M. Mano and M. D. Ciletti, *Digital design: with an introduction to the Verilog HDL*. Pearson Education, New Jersey, 2013.
[5] G. C. Ram, D. S. Rani, R. Balasaikesava, and K. B. Sindhuri, "Design of delay efficient modified 16 bit Wallace multiplier," in *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, May 2016, pp. 1887–1891, doi: 10.1109/RTEICT.2016.7808163.
[6] A. Habibi and P. A. Wintz, "Fast multipliers," *IEEE Transactions on Computers*, vol. C–19, no. 2, pp. 153–157, Feb. 1970, doi: 10.1109/T-C.1970.222881.
[7] S. Suman, N. P. Singh, R. Selvakumar, and H. Saini, "Design of 32-bit cell-based carry-save combinational multiplier with reduced area and propagation delay," *Journal of Physics: Conference Series*, vol. 1804, no. 1, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012195.
[8] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964, doi: 10.1109/PGEC.1964.263830.
[9] A. Sundhar, S. D. Tharshini, G. Priyanka, S. Ragul, and C. Saranya, "Performance analysis of Wallace tree multiplier with kogge stone adder using 15-4 compressor," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2019, pp. 903–907, doi: 10.1109/ICCSP.2019.8697981.
[10] T. Satish and K. S. Pande, "Multiplier using NAND based compressors," in *2019 3rd International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, Aug. 2019, pp. 1–6, doi: 10.1109/IEMENTech48150.2019.8981067.
[11] M. C. Hanumantharaju, H. Jayalaxmi, R. K. Renuka, and M. Ravishankar, "A high speed block convolution using ancient Indian vedic mathematics," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, Dec. 2007, pp. 169–173, doi: 10.1109/ICCIMA.2007.332.
[12] A. Eshack and S. Krishnakumar, "Pipelined Vedic multiplier with manifold adder complexity levels," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2951–2958, Jun. 2020, doi: 10.11591/ijece.v10i3.pp2951-2958.
[13] V. Anbumani, S. Soviya, S. Sneha, and L. Saran, "Speed and power efficient Vedic multiplier using adders with MUX," in *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Nov. 2021, pp. 1–5, doi: 10.1109/i-PACT52855.2021.9696992.
[14] J. Kuppili, M. Abhiram, and N. A. Manga, "Design of Vedic mathematics based 16 bit MAC unit for power and delay optimization," in *2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, Jan. 2021, pp. 1–4, doi: 10.1109/ICNTE51185.2021.9487570.
[15] J. S. S. Bharath and K. Tirathji, "Vedic mathematics or sixteen simple sutras from the Vedas," *Motilal Banarsidas, Varanasi (India)*, vol. 6, no. 7, 1986.
[16] L. Dadda, "Some schemes for parallel multipliers," *Alta frequenza*, vol. 34, pp. 349–356, 1965.
[17] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," *IEEE Transactions on Computers*, vol. 44, no. 8, pp. 962–970, 1995, doi: 10.1109/12.403712.
[18] M. Mehta, V. Parmar, and E. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits," in *Proceedings 10th IEEE Symposium on Computer Arithmetic*, 1991, pp. 43–50, doi: 10.1109/ARITH.1991.145532.
[19] E. E. Swartzlander, "A review of large parallel counter designs," in *IEEE Computer Society Annual Symposium on VLSI*, 2004, pp. 89–98, doi: 10.1109/ISVLSI.2004.1339513.
[20] S. Veeramachaneni, A. Lingamneni, M. K. Krishna, and M. B. Srinivas, "Novel architectures for efficient (m, n) parallel counters," in *Proceedings of the 17th great lakes symposium on Great lakes symposium on VLSI-GLSVLSI '07*, 2007, pp. 188–191, doi: 10.1145/1228784.1228833.
[21] 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 Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004, doi: 10.1109/TCSI.2004.835683.
[22] D. KavyaShree, P. Samundiswary, and K. V Gowreesrinivas, "High speed multipliers using counters based on symmetric stacking," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2019, pp. 1–6, doi: 10.1109/ICCCI.2019.8822185.
[23] L. H. Krishna, M. Neeharika, V. Janjirala, S. Veeramachaneni, and N. Mahammad S, "Efficient design of 15:4 counter using a novel 5:3 counter for high-speed multiplication," *IET Computers and Digital Techniques*, vol. 15, no. 1, pp. 12–19, Jan. 2021, doi: 10.1049/cdt2.12002.
[24] S. Asif and Y. Kong, "Design of an algorithmic Wallace multiplier using high speed counters," in *2015 Tenth International Conference on Computer Engineering and Systems (ICCES)*, Dec. 2015, pp. 133–138, doi: 10.1109/ICCES.2015.7393033.
[25] K. Sivanandam and P. Kumar, "Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor," *Microprocessors and Microsystems*, vol. 65, pp. 97–106, Mar. 2019, doi: 10.1016/j.micpro.2019.01.002.
[26] C. Fritz and A. T. Fam, "Fast binary counters based on symmetric stacking," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2971–2975, Oct. 2017, doi: 10.1109/TVLSI.2017.2723475.
[27] S. Kumar and T. N. Sasamal, "Verilog implementation of high-speed Wallace tree multiplier," in *Lecture Notes in Networks and Systems*, Springer Singapore, 2021, pp. 457–469.
[28] W. Guo and S. Li, "Fast binary counters and compressors generated by sorting network," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1220–1230, Jun. 2021, doi: 10.1109/TVLSI.2021.3067010.
[29] N. Weste, *CMOS VLSI design-A circuits and systems perspective*, 4th ed. Pearson, 2010.

BIOGRAPHIES OF AUTHORS

Madaka Venkata Subbaiah received B.Tech degree in Electronics and Communication Engineering from K. S. R. M. College of Engineering, S. V. University, Kadapa in 2005, and M.Tech in VLSI System Design from A.I.T.S., J.N.T.U.A., Rajampeta in 2012. He is currently a Research Scholar in the Department of Electronics and Communication Engineering, S.V.U. College of Engineering, S.V. University, Tirupati, Andhra Pradesh, India. His current research interests include VLSI System Design, VLSI signal processing. He can be contacted by email: venkatmadaka@gmail.com.

Galiveeti Umamaheswara Reddy received B.Tech Degree in Electronics and Communication Engineering, M.Tech in Instrumentation and Control Systems, and obtained Ph.D. from Sri Venkateswara University, Tirupati. He is a member of the ISTE, IE, and BMSI. Currently, he is Professor at Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India. He has a teaching experience of more than 25 years and has 30 technical publications in national/international journals. His areas of interest include VLSI and signal processing. He can be contacted by email: umaskit@gmail.com.