

DESIGN OF 4-BIT PARTIAL PRODUCTS PARTITIONING ADDER WITH HIGH SPEED AND LOW POWER DELAY PRODUCT

Bhaskara Rao Doddi*, Hari Kanaka Raghu Vamsi Kudulla and Vandana Mahanthi

Received: May 04, 2021; Revised: August 01, 2021; Accepted: October 06, 2021

Abstract

Adder has been designed with uniquely designed XNOR and carry cells for every 4 bits. In Look ahead structure, generate and propagate are designed in inverting methodology unlike the conventional ones. This will give a more parallel structure to reduce the delay, as partial products were partitioned. Proposed adder has reported better results when compared to the many existing designs in the recent literature with respect to power, delay and power delay product. Delay analysis for every block is done by estimating the critical paths for the events in the output by determining the strongly charged and discharged paths. Power analysis has been done by activating the possible power-consuming transitions for every block in the circuit. Tanner EDA tool has been used with 250 nm technology.

Keywords: Adder, speed, power delay product, partial products

Introduction

Adder is an important module in any processing unit. Various adder structures were proposed by many researchers in the literature. Adders can be designed using CMOS logic style (Weste and Eshraghian, 1985). CMOS style has the advantage of low static power and low power delay product (PDP). In recent days, hybrid logic structures became popular, which uses fewer transistors than CMOS logic style (Hamed and Somayeh, 2018).

Based on the output voltage level, digital circuits can be classified into full-swing and non-full-swing categories. Adder has low power but not able to compute quickly when compared to few designs which can effect the Energy Delay Product (Bhattacharya *et al.*, 2015). Hybrid Full adder has

less average power consumption but by needing 24 transistors (Vesterbacka, 1999). Adder has less Power Delay Product with using the multiplexer as the key cell for the design (Aguirre-Hernandez and Linares-Aranda, 2011).

Power efficient design was proposed by minimizing the power-consuming transitions (Hassoune *et al.*, 2010). Number of levels are minimized and also the hardware is optimized to achieve less delay (Chang *et al.*, 2005). Design was proposed to achieve high speed with low voltage which compensates the trade-off with hybrid structure (Kumar and Sharma, 2016). Design was proposed which minimizes the trade-off between the supply voltage and its speed (Wairya *et al.*, 2011).

Faculty of ECE, Raghu Engineering College(A), Visakhapatnam, India. E-mail: bhaskararao.doddi@raghuengcollege.in, raghuvamsi.kudulla@raghuengcollege.in, vandana.mahanthi@raghuengcollege.in

* Corresponding author

Design is proposed with minimal area but with effected noise margins (Chowdhury *et al.*, 2008). Hybrid Adder for ripple carry structure is proposed by using XNOR/XOR cell (Kandpal *et al.*, 2020). High speed hybrid logic adder for ripple carry structure is proposed by using AND/OR cell (Hasan *et al.*, 2020). Adder for carry select structure proposed with high speed and low power (Mohanty and Patel, 2014).

Review of Full Adders

This section reviews one of the hybrid full adder structures, where it has shown the best performance of all the hybrid structures. Design of the full adder is also analyzed for the delay and power consumption.

Hybrid Full Adder (Hamed and Somayeh, 2018)

Hybrid logic is a mix of logic styles where it is intended to achieve some constraints of VLSI by sacrificing other constraints. The design problem is to minimize the dis-advantages and to maximize the advantages. Full swing may be a problem in a few cases or noise may be a problem in other cases,-; as a designer, those things should be addressed properly.

XNOR/XOR blocks have been designed to feed the logic requirements of both the sum and carry output and the rest of the hardware required is the same for both sum and carry output. XNOR/XOR blocks has been designed with 12 number of transistors where the critical path has taken two transistors individually for both XNOR as well as XOR. All the other paths have taken a single transistor, and it will be a total of six transistors, and one not gate in common feeds the logical requirement of XNOR/XOR cell

Delay For Adder

To compute the delay for 1-bit adder sum output, it requires XNOR/XOR cell in the first level and transmission gate in the second level and not gate in the third level. The delay is just the sum of all the hardware blocks delays. To compute 1-bit adder carry output (C0), the critical path is also the same as the sum output. To compute the delay for 2-bit adder sum output, the delay is the sum of the delay of the C0 and not gate, transmission gate and again a not gate. To compute the delay for 2-bit adder carry output (C1), the critical path is also the same as the Sum output. To Compute the delay for N-bit adder, it is the delay of XNOR/XOR cell, N* transmission gates and (3*N)-1 not gates for both sum and carry output.

Power For Adder

To compute Power for adder, a total number of blocks needs to be considered and for every individual block all possible power consuming transitions needs to be activated and then average power consumption will be measured. To compute the average power consumption of the entire circuit consider the number of copies for every individual block and then do the necessary multiplication and addition to find out the power consumption.

Three different cells/blocks have been used and they are XNOR/XOR cell, transmission gate (TG), and not gate. Four XNOR/XOR cells have been used, eight TG have been used and twelve not gates have been used, and all those need to be considered while computing the power consumption

Design of Full Adder

XNOR Cell

XNOR cell generates two outputs, they are half sum and half carry in complemented version. There are six paths possible in the XNOR cell, where only four are active paths. Since output will be known only when both the inputs are known. The propagate (XNOR) and generate (NAND) logic equations are shown below in equations 1 and 2. Values of i can be substituted from 0-3 for 4-bit and 0-(N-1) for N-bit. They are in the complemented form of standard generate and propagate equations, where XOR and AND logic are used. The basic principle of carry look ahead structure is shown in (Miao and Li, 2017). This is what given us the flexibility in modifying the standard equations of look-ahead carry adder. Partial products are also being partitioned to reduce the size of the fan-in to increase the speed. The logic circuit is shown in Figure 1.

$$P_i = A_i \text{XNOR } B_i \quad (1)$$

$$G_i = A_i \text{NAND } B_i \quad (2)$$

Nand gate is used to generate the logic required for xnor output in Figure 1 and the Boolean equation required is

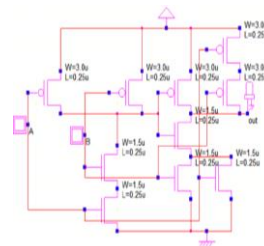


Figure 1. Schematic of XNOR

$$X_{nor} = (\text{nand}.(A+B))' \quad (3)$$

Nor Gate

To generate the carry outputs we require both propagate and generate outputs and the generalized equations are shown below

$$C_{i+1} = G_i' + P_i' . C_i' \quad (4)$$

$$S_i = P_i \text{ XNOR } C_i \quad (5)$$

Since and gate with complemented inputs that is $P_i' . C_i'$ result in a nor gate, we need to design nor gate and it's circuit is shown in Figure 2. The logic of nor gate is when both the inputs are low then the output generated is high and rest of the cases it is low.

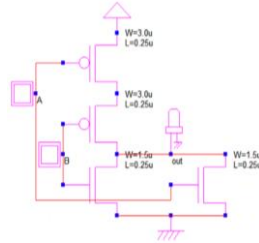


Figure 2. Schematic of nor cell

Carry1 Cell

Figure 3 shows the schematic of carry1 cell to generate carry output of the least significant bit stage of the adder. Boolean expression for carry1 generation is shown in Equation 6. Logic requirement is a NAND gate and one NOT gate. If X_0 is high or G_0 is low then C_1 is high and rest of the cases C_1 is low.

$$C_1 = G_0' + P_0' . C_0' = G_0' + X_0 \quad (6)$$

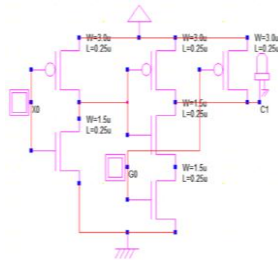


Figure 3. Schematic of carry1 cell

Carry2 Cell

Figure 4 shows the schematic of carry2 cell to generate carry output of the (LSB+1) stage of the adder. Boolean expression for carry2 generation is shown in Equation 7, where Fan-in is four in the second product term. To reduce the fan-in in Equation 7, the expression is modified as in

Equation 8 to reduce the fan-in. Logic requirement is a AOI21 and a NAND gate.

$$C_2 = P_1' . G_0' + P_1' . P_0' . P_0' . C_0 + G_1' \quad (7)$$

$$= X_1 + X_2 . X_0 + G_1' \quad (8)$$

when X_2 and X_0 are high or X_1 is high or G_1 is low then C_2 is high and rest of the input possibilities C_2 is low.

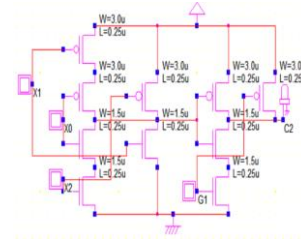


Figure 4. Schematic of carry2 cell

Carry3 Cell

Figure 5 shows the schematic of the carry3 cell to generate the carry output of the (LSB+2) stage of the adder. Boolean expression for carry3 generation is shown in Equation 9, where Fan-in is four in the second product term. To reduce the fan-in in Equation 9, expression is modified as in Equation 10 to reduce the fan-in. The logic requirement is an 8- transistor cell and a NAND gate.

$$C_3 = P_2' . G_1' + P_2' . P_1' (P_1' . G_0' + P_0' . C_0) + G_2' \quad (9)$$

$$= X_3 + X_4 (X_1 + X_0) + G_2' \quad (10)$$

when X_3 is high or G_2 is low or X_4 and X_1 are high or X_4 and X_0 are high then C_3 is high and other than these possibilities of input combinations C_3 is low.

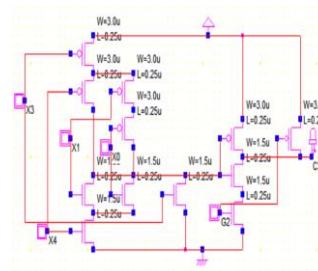


Figure 5. Schematic of carry3 cell

Carry4 Cell

Figure 6 shows the schematic of carry4 cell to generate carry output of the (LSB+3) stage that is for the most significant bit stage of the adder. Boolean expression for carry4 generation is shown in Equation 11, where Fan-in is six in the second product term. To reduce the fan-in in Equation 11, expression is modified as in Equation 12 to reduce

the fan-in. The logic requirement is an 12 transistor cell and a NAND gate.

$$C_4 = P_3'.G_2'+P_3'.P_2'.(P_2'.G_1'+P_1'.G_0'+P_1'.P_0'.P_0'.C_0) + G_3' \quad (11)$$

$$= X_5+X_6(X_3+X_1+X_2X_0)+G_3' \quad (12)$$

when X_5 is high or G_3 is low or X_6 and X_3 are high or X_6 and X_1 are high or X_6 and X_2 and X_0 are high then C_4 is high and other than these possibilities of input combinations C_4 is low.

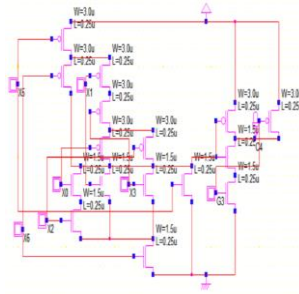


Figure 6. Schematic of carry4 cell

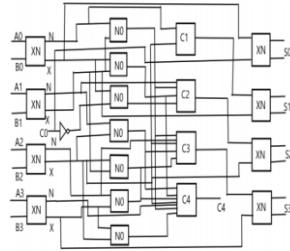


Figure 7. Schematic of 4-bit proposed adder

4-BitFull Adder

As shown in Figure 7, Level1 has four XN(XNOR) cells to produce X (XNOR) and N(NAND) outputs. Level 2 has seven NO(NOR) gates to produce partial products which are required to design the carry outputs. This level 2 NOR gates has the fan-in in two such that carry outputs can be generated faster. Level 3 has four blocks named C_1 , C_2 , C_3 , C_4 , and they are carry blocks which are designed by utilizing the cells/gates from level1 and level 2. Level 4 has XNOR cells to produce full sum outputs S_0 , S_1 , S_2 , S_3 by utilizing the carry blocks from level 3 and also XNOR cells from level 1. In total, four levels are needed to generate the sum output and three levels to generate the carry outputs.

Above waveform from Figure 8 has been generated based on the test pattern applied in the below Table 1. It shows that as per the input's applied, correct functionality can be observed in it.

Table 1 shows sample data inputs applied with rise and fall time equal to 1 ns. Random 8-bit data is applied for 80 ns with a pulse width of 10 ns.

Performance Analysis

Proposed Design

Xnor Cell

Table 2 shows the output for four possibilities of output and their critical path delays. In the first row "00-01" indicates when the previous input is "00", then the output is '1', and now present input applied is "01", output should become '0', now we will measure the delay. In the second row "01-11" indicates when the previous input is "01", then the output is '0', and now the present input applied is "11" output should become '1', now we will measure the delay.

In the third row, "11-10" indicates when the previous input is "11", then the output is '1', and now present input applied is "10" output should become '0', now we will measure the delay. In the fourth row "10-00" indicates when the previous

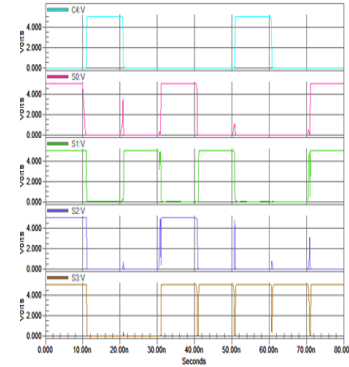


Figure 8. Waveform of 4-bit adder

Table 1. Test vector

Input	Data stream	Rise/Fall time (ns)	Pulse width (ns)
A3	00000110	1	10
A2	00010101	1	10
A1	00011001	1	10
A0	00101001	1	10
B3	11000100	1	10
B2	11011100	1	10
B1	11011001	1	10
B0	11100001	1	10
Cin	01011001	1	10

Table 2. Delay for xnor cell

AB	Output	Critical path delay (ns)	Delay (ns)
00-01	0	0.01	0.17
01-11	1	0.17	
11-10	0	0.13	
10-00	1	0.13	

input is “10”, then the output is ‘0’ and now the present input applied is “00”, the output should become ‘1’, now we will measure the delay. When input applied is from “01-11” the delay is maximum because to get the output for ‘11’ three transistors are there in the path, but for “10-00” the delay is less in comparison due to only two transistors are there in the path.

Table 3. Delay for NOR Cell

AB	Output	Critical path delay (ns)	Delay (ns)
11-00	1	0.25	0.25
00-01	0	0.001	
01-00	1	0.12	
00-10	0	0.001	

Table 4. Delay for carry1 Cell

X0G0	C1	Critical path delay (ns)	Delay (ns)
10-01	0	0.12	0.12
01-11	1	0.09	

Table 5. Delay for carry2 Cell

G1X2X1X0	C2	Critical path delay (ns)	Delay (ns)
0111-1001	0	0.17	0.17
1001-1101	1	0.001	
0111-1100	0	0.16	
1100-1101	1	0.13	

Nor Cell

Table 3 shows the output for four possibilities of output and their critical path delays. In the first row “11-00” indicates when the previous input is ‘11’ then the output is ‘0’ and now present input applied is ‘00’ output should become ‘1’, now we will measure the delay because when the input is ‘11’ it means that nor gate is strongly discharged. In the second row “00-01” indicates when the previous input is ‘00’ then the output is ‘1’ and now present input applied is ‘01’ output should become ‘0’, now we will measure the delay because ‘00’ is the one and only possible charging path.

In the third row, “01-00” indicates when the previous input is ‘01’, then the output is ‘0’ and now the present input applied is ‘00’ output should become ‘1’, now we will measure the delay. In the fourth row, “00-10” indicates when the previous input is ‘00’ then the output is ‘1’ and now present input applied is ‘10’ output should become ‘0’, now we will measure the delay because ‘00’ is the one and only possible charging path. When input applied is from “11-00” the delay is maximum because the previous input has strongly discharged the output. When input applied is from “01-00” the

delay is minimal because previous input has not strongly discharged the output.

Carry1 Cell

Table 4 shows the output for two possibilities of output and their critical path delays. In the first row “10-01” indicates when the previous input is ‘10’, then the output is ‘1’ and now present input applied is ‘01’, output should become ‘0’, now we will measure the delay because when the input is ‘10’, it means that nor gate is strongly charged and also ‘01’ is the one and only critical path for the output of ‘0’.

In the second row, “01-11” indicates when the previous input is ‘01’, then the output is ‘0’, and now present input applied is ‘11’, output should become ‘1’, now we will measure the delay because ‘01’ is the one and only possible discharging path and also ‘11’ is the critical path for the output of ‘1’.

Carry2 Cell

Table 5 shows the output for four possibilities of output and their critical path delays. In the first row “0111-1001” indicates when the previous input is ‘0111’, then the output is ‘1’ and now present input applied is ‘1001’, output should become ‘0’, now we will measure the delay because when the input is ‘0111’ it means that carry2 cell is strongly charged and also ‘1001’ is one of the critical path for the output of ‘0’.

In the second row, “1001-1101” indicates when the previous input is ‘1001’, then the output is ‘0’ and now present input applied is ‘1101’, output should become ‘1’, now we will measure the delay because ‘1001’ is the one of the possible strong discharging path and also ‘1101’ is the critical path for the output of ‘1’.

In the third row, “0111-1100” indicates when the previous input is ‘0111’, then the output is ‘1’, and now present input applied is ‘1100’, output should become ‘0’, now we will measure the delay because when the input is ‘0111’ it means that carry2 cell is strongly charged and also ‘1100’ is one more critical path for the output of ‘0’.

In the fourth row, “1100-1101” indicates when the previous input is ‘1100’, then the output is ‘0’ and now present input applied is “1101”, output should become ‘1’, now we will measure the delay because ‘1101’ is one more possible strong discharging path and also ‘1101’ is the critical path for the output of ‘1’.

Carry3 Cell

Table 6 shows the output for three possibilities of output and their critical path delays. In the first row “01111-11000” indicates when the previous input is ‘01111’, then the output is ‘1’ and now

present input applied is '11000', output should become '0', now we will measure the delay because when the input is '01111' it means that carry3 cell is strongly charged and also '11000' is one and only critical path for the output of '0'.

In the second row, "10000-11010" indicates when the previous input is '10000' then the output is '0' and now present input applied is '11010' output should become '1', now we will measure the delay because '10000' is the one and only possible strong discharging path and also '11010' is one of the critical path for the output of '1'.

In the third row, "10000-11001" indicates when the previous input is '10000', then the output is '0' and now present input applied is '11001', output should become '1', now we will measure the delay because when the input is '10000' it means that carry3 cell is strongly charged and also '11001' is one more critical path for the output of '1'.

Carry4 Cell

Table 7 shows the output for three possibilities of output and their critical path delays. In the first row, "1000000-1100101" indicates when the previous input is '1000000', then the output is '0' and now present input applied is '1100101', output should become '1', now we will measure the delay because when the input is '1000000' it means that carry4 cell is strongly discharged and also '1100101' is one and only critical path for the output of '1'.

In the second row, "0111111-1100100" indicates when the previous input is '0111111', then the output is '1', and now present input applied is '1100100', output should become '0', now we will measure the delay because '0111111' is the one and only possible strong charging path and also '1100100' is one of the critical paths for the output of '0'.

In the third row, "0111111-1100001" indicates when the previous input is '0111111', then the output is '1', and now present input applied is '1100001', output should become '0', now we will measure the delay because when the input is '0111111' it means that carry4 cell is strongly charged and also '1100001' is one more critical path for the output of '0'.

Table 8 shows the number of levels required and the hardware required in those levels for obtaining the outputs of the 4-bit adder. For obtaining the output for S0, we require only two levels and the hardware needed is XNOR in both the levels and delay will be twice the delay needed for XNOR and it is 0.34 ns.

For obtaining the output for S1, we require four levels and the hardware needed is XNOR,NOR and CARRY1 cell and again XNOR to compute the

final carry and delays in all the levels are added to get the delay for S1 and it is 0.71 ns.

For obtaining the output for S2, we require four levels and the hardware needed is XNOR,NOR and CARRY2 cell and again XNOR to compute the final carry and delays in all the levels are added to get the delay for S2 and it is 0.76 ns.

Table 6. Delay for carry3 Cell

G2X4X3X1X0	C3	Critical path delay (ns)	Delay (ns)
01111-11000	0	0.25	0.25
10000-11010	1	0.17	
10000-11001	1	0.23	

Table 7. Delay for carry4 Cell

G3X6X5X3X2X1X0	C4	Critical path delay (ns)	Delay (ns)
1000000-1100101	1	0.31	0.62
0111111-1100100	0	0.62	
0111111-1100001	0	0.58	

Table 8. Delay for 4-bit adder (ns)

Out	Level 1	Level 2	Level 3	Level 4	Delay
S0	Xnor (0.17)	Xnor (0.17)	-	-	0.34
S1	Xnor (0.17)	Nor (0.25)	Carry1 (0.12)	Xnor (0.17)	0.71
S2	Xnor (0.17)	Nor (0.25)	Carry2 (0.17)	Xnor (0.17)	0.76
S3	Xnor (0.17)	Nor (0.25)	Carry3 (0.25)	Xnor (0.17)	0.84
C4	Xnor (0.17)	Nor (0.25)	Carry4 (0.62)	-	1.04

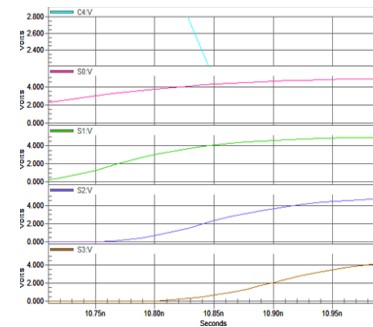


Figure 9. Waveform showing the delay for proposed

For obtaining the output for S3, we require four levels and the hardware needed is XNOR,NOR and CARRY3 cell and again XNOR to compute the final carry and delays in all the levels are added to get the delay for S3 and it is 0.84 ns. For obtaining the output for C4, we require three levels and the hardware needed is XNOR,NOR and CARRY4 cell and delays in all the levels are added to get the delay for C4 and it is 1.04 ns.

The above waveform from Figure 9 has been generated based on the input test patterns for A is “0000” and B is “1111” and Cin is ‘1’ for the first pulse and now for the second pulse the input test patterns for A is “0000” and B is “1111” and Cin is ‘0’. As we can observe C4 output has reached to 50% of the output voltage that is 2.5 V much faster than S3 and S4.

The test pattern is chosen to activate the critical path of the 4-bit full adder because when every bit of A and B are unequal, output depends on the initial carry input. Delay is 0.34ns for the carry output C4.

Existing Design (Hamed and Somayeh, 2018)

Xnor/Xor Cell

Table 9 indicates that the maximum delay for XNOR cell is 0.21 ns, when the applied input is from “10-00” and for the XOR cell is 0.11 ns when the applied input is from “01-11” and the delay for the XNOR/XOR cell is 0.21 ns.

TG Cell

Table 10 indicates that maximum delay for TG cell is 0.13 ns when the applied input is from “10-01” and the minimum delay is 0.05 ns when the applied input is from “01-10” and the delay for the TG cell is 0.13 ns

Not Gate

Table 11 indicates that maximum delay for not gate is 0.125 ns when the applied input is from “1-0” and the minimum delay is 0.123 ns when the applied input is from “0-1” and the delay for the not gate is 0.125 ns.

Table 12 shows the number of levels required and the hardware required in those levels for obtaining the outputs of the 4-bit adder. For obtaining the output for S0, we require three levels and the hardware needed is XNOR/XOR in the first level, TG cell in the level 2 and not gate in the level 3 and delays in all the levels are added to get the delay for S0, and it is 0.465 ns.

For obtaining the output for S1, we require four levels and the hardware needed is identical like S0 in the first level because C1 also requires the same hardware in the level1, not gate in the level 2, TG cell in the level 3 and NOT gate in the level 4 and delays in all the levels are added to get the delay for S1, and it is 0.845 ns.

For obtaining the output for S2, we require four levels, and the hardware needed is identical like S1 in the first level because C2 also requires the same hardware in the level 1, not gate in the level 2, TG cell in the level 3 and NOT gate in the level 4 and delays in all the levels are added to get the delay for S2, and it is 1.225 ns.

For obtaining the output for S3, we require four levels and the hardware needed is identical like S2 in the first level because C2 also requires the same hardware in the level 1, not gate in the level 2, TG cell in the level 3 and NOT gate in the level4 and delays in all the levels are added to get the delay for S3, and it is 1.605 ns.

Table 9. Delay for XNOR/XOR Cell

AB	XNOR/XOR	Critical path delay (ns)	Delay (ns)
00-01	0/1	0.03/0.01	0.21
01-11	1/0	0.01/0.11	
11-10	0/1	0.01/0.03	
10-00	1/0	0.21/0.008	

Table 10. Delay for TG Cell

Inin1	Out	Critical path delay (ns)	Delay (ns)
10-01	1	0.13	0.13
01-10	0	0.05	

Table 11. Delay for Not gate

In	Out	Critical path delay (ns)	Delay (ns)
1-0	0	0.125	0.125
0-1	1	0.123	

Table 12. Delay for 4-bit adder

Output	Level 1	Level 2	Level 3	Level 4	Delay
S0	Xnor/Xor (0.21)	TG (0.13)	Not (0.125)	-	0.465
S1	C1 (0.465)	Not (0.125)	TG (0.13)	Not (0.125)	0.845
S2	C2 (0.845)	Not (0.125)	TG (0.13)	Not (0.125)	1.225
S3	C3 (1.225)	Not (0.125)	TG (0.13)	Not (0.125)	1.605
C4	C4 (1.605)	-	-	-	1.605

For obtaining the delay for C4 it is equivalent to the S3, since they both have the same critical path and it is also 1.605 ns.

The above waveform from Figure 10 has been generated based on the input test patterns for A is “0000” and B is “1111” and Cin is ‘0’ for the first pulse and now for the second pulse the input test patterns for A is “0000” and B is “1111” and Cin is ‘1’. As we can observe C4 output has reached to 50% of the output voltage that is 2.5 V, much slower than S2 and S3.

The test pattern is chosen such that it will activate the critical path of the 4-bit adder, because when every bit of A and B are unequal then output depends on the initial carry input. Delay is almost 0.9ns for carry output C4.

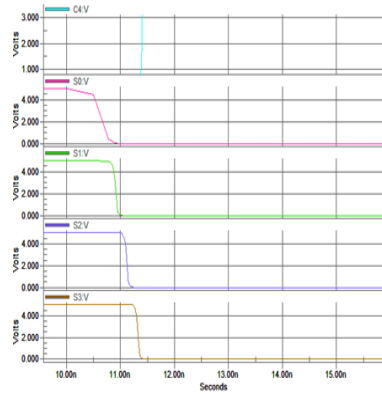


Figure 10. Waveform showing the delay for (Hamed and Somayeh, 2018)

Power Analysis

Proposed Design

Xnor Cell

A two-input XNOR cell has the following possible power-consuming transitions which are shown in the below Table 13, and the test vectors are applied to measure the average power consumption of the XNOR cell.

Table 13 shows four power consuming transitions, when input's go from "01" to "00", "10" to "00", "01" to "11" and "10" to "11".

Nor Gate

A two-input NOR gate has the following possible power-consuming transitions which are shown in the below Table 14 and the test vectors are applied to measure the average power consumption of the Nor gate.

Table 14 shows three power-consuming transitions, when input's go from "01" to "00", "10" to "00" and "11" to "00".

Not Gate

A not gate has the following possible power-consuming transitions which are shown in the below Table 15, and the test vectors are applied to measure the average power consumption of the not gate.

Table 15 shows one power-consuming Transition, when input's go from "1" to "0".

Carry1 Cell

Carry1 cell has the following power-consuming transitions which are shown in the below Table 16, and the test vectors are applied to measure the average power consumption of the carry1 cell.

Table 16 shows three power-consuming transitions, when input's go from "01" to "00", "01" to "10" and "01" to "11".

Table 13. Power consuming transitions

S. No	A	B	Output
1	0	1	0
2	0	0	1
3	1	0	0
4	0	0	1
5	0	1	0
6	1	1	1
7	1	0	0
8	1	1	1

Table 14. Power consuming transitions

S. No	A	B	Output
1	0	1	0
2	0	0	1
3	1	0	0
4	0	0	1
5	1	1	0
6	0	0	1

Table 15. Power consuming transitions

S. No	In	Output
1	1	0
2	0	1

Table 16. Power consuming transitions

S. No	X0	G0	C1
1	0	1	0
2	0	0	1
3	0	1	0
4	1	0	1
5	0	1	0
6	1	1	1

Table 17. Power consuming transitions

X1	X0	X2	G1	C2
0	0	1	1	0
0	0	0	0	1
0	0	1	1	0
1	0	0	1	1
0	0	1	1	0
0	1	1	1	1
0	1	0	1	0
0	0	0	0	1
0	1	0	1	0
1	0	0	1	1
0	1	0	1	0
0	1	1	1	1

Carry2 Cell

Carry2 cell has the following power-consuming transitions which are shown in the below Table 17, and the test vectors are applied to measure the average power consumption of the carry2 cell.

Table 17 shows six power consuming Transitions similar to the methodology applied in the previous Tables 13, 14, 15, 16.

Carry3 Cell

Carry3 cell has the following possible power-consuming transitions which are shown in the below Table 18 and the test vectors are applied to measure the average power consumption of the Carry3 cell.

Table 18 shows eight power consuming Transitions similar to the methodology applied in the previous Tables 13, 14, 15, 16.

Table 18. Power consuming transitions

X3	X4	X1	X0	G2	C3
0	0	1	1	1	0
0	0	0	0	0	1
0	0	1	1	1	0
1	0	0	0	1	1
0	0	1	1	1	0
0	1	0	1	1	1
0	0	1	1	1	0
0	1	1	0	1	1
0	1	0	0	1	0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	0	0	1	0
0	1	1	0	1	1

Carry4 Cell

Carry4 cell has the following possible power consuming transitions which are shown in the below Table 19, and the test vectors are applied to measure the average power consumption of the Carry4 cell.

Table 19 shows fifteen power consuming Transitions similar to the methodology applied in the previous Tables 13, 14, 15, 16.

4-Bit Adder Cell

To design the 4-bit adder, we require the above mentioned cells/gates and the below Table 20 shows how to calculate the average power consumption.

Table 20 shows 8 XNOR's, 7 NOR's, 1 NOT, 1 carry1, 1 carry2, 1 carry3, 1 carry4 cells/gates respectively. Average power consumption per cell/gate is given. For example, the average power consumption of XNOR is 2.136049 e-004w and there are 8 such cells which leads to 17.088 e-004w. Total power is the average overall power of the cells.

Existing Design (Hamed and Somayeh, 2018)**Xnor/Xor Cell**

A two-input XNOR/XOR cell has the following possible power consuming transitions which are shown in the below Table 21, and the test vectors are applied to measure the average power consumption of the XNOR/XOR cell

Table 21 shows four power-consuming transitions, when the input goes from "01" to "00", "10" to "00", "01" to "11", and "10" to "11".

Table 19. Power consuming transitions

X5	X6	X1	X3	X0	X2	G3	C4
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	1
0	0	1	1	1	1	1	0
1	0	0	0	0	0	1	1
0	0	1	1	1	1	1	0
0	1	0	1	0	0	1	1
0	0	1	1	1	1	1	0
0	1	1	0	0	0	1	1
0	0	1	1	1	1	1	0
0	1	0	0	1	1	1	1
0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	1
0	1	0	0	0	1	1	0
1	0	0	0	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	1	0	0	1	1
0	1	0	0	0	1	1	0
0	1	1	0	0	0	1	1
0	1	0	0	0	1	1	0
0	1	0	0	1	1	1	1
0	1	0	0	1	0	1	0
0	0	0	0	0	0	0	1
0	1	0	0	1	0	1	0
1	0	0	0	0	0	1	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	1
0	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1
0	1	1	0	0	0	1	1
0	1	0	0	1	0	1	0
0	1	0	0	1	1	1	1

Table 20. Power consumption of 4-bit adder

Cell/Gate	No	Average power/Cell (e-004w)	Overall power of the Cells (e-004w)	Total power (mw)
Xnor	8	2.136049	17.088	3.925
Nor	7	1.511389	10.579	
Not	1	1.182231	1.182231	
Carry1	1	2.493040	2.493040	
Carry2	1	2.561809	2.561809	
Carry3	1	2.442800	2.442800	
Carry4	1	2.904415	2.904415	

Table 21. Power consuming transitions

S. No	A	B	Xnor
1	0	1	0
2	0	0	1
3	1	0	0
4	0	0	1
5	0	1	0
6	1	1	1
7	1	0	0
8	1	1	1

Not Gate

A not gate has the following possible power consuming transitions which are shown in the below Table 22, and the test vectors are applied to measure the average power consumption of the Not gate.

Table 22 shows one power consuming Transition, when the input goes from '1' to '0'.

Table 22. Power consuming transitions

S. No	A	Output
1	1	0
2	0	1

Table 23. Power consuming transitions

S. No	A	B	C	D	Output
1	0	0	1	1	0
2	1	1	0	0	1
3	0	1	0	1	0
4	1	0	1	0	1

Table 24. Average power consumption of the 4-bit adder(e-004)

Cell/Gate	No	Average power/cell	Overall power/cell	Total power (mw)
Xnor/Xor	4	3.018458	12.073	2.75
not	12	1.182231	14.186	
TG	8	0.1542363	1.233	

Table 25. Performance comparison for various 4-bit adders in terms of power, delay and power delay product

Design	Performance		
	Power (mw)	Delay (ps)	Power delay product(fj)
Proposed	13.070	395	5,162.65
Mohanty and Patel (2014)	26.760	290	7,760.40
Kandpal <i>et al.</i> (2020)	7.482	206	1,541.29
Hasan <i>et al.</i> (2020)	8.455	585	4,946.17
Hamed and Somayeh (2018)	13.800	768	10,598.40

Table 26. Performance comparison for various 8-bit adders in terms of power, delay and powerdelay product

Design	Performance		
	Power (mw)	Delay (ps)	Power delay product (fj)
Proposed	5.269	830	4,373.27
Mohanty and Patel (2014)	8.775	620	5,440.50
Kandpal <i>et al.</i> (2020)	3.610	590	2,129.90
Hasan <i>et al.</i> (2020)	3.546	890	3,155.94
Hamed and Somayeh (2018)	6.640	1,675	1,1122

Table 27. Performance comparison for various 16-bit adders in terms of power, delay and power delay product

Design	Performance		
	Power (mw)	Delay (ps)	Power delay product (fj)
Proposed	10.440	1,860	19,418
Mohanty and Patel (2014)	17.560	1,305	22,915
Kandpal <i>et al.</i> (2020)	10.460	2,050	21,443
Hasan <i>et al.</i> (2020)	7.105	2,580	18,330
Hamed and Somayeh (2018)	13.300	3,450	45,885

TG Cell

A TG (Transmission gate) cell has the following possible power consuming transitions shown in the below Table 23.

Table 23 shows two power consuming Transitions, when the input's go from "0011" to "1100" and "0101" to "1010".

4-Bit Adder Cell

To design the adder, we require the above-mentioned cells and the below Table 24 shows how to calculate the average power consumption.

Table 24 shows 4 XNOR/XOR's, 12 NOT gates, 8 TG's respectively. Average power consumption per cell/gate is given, for example average power consumption of NOT is 1.182231 e-004w and there are 12 such cells which leads to 14.186 e-004w.

Results

To estimate the power consumption of the adder, test pattern with 56 input vector to vector transitions proposed in (Parameshwara and Srinivasaiah, 2017). This test pattern is applied for adders in this section. To find out the delay, carry input of the LSB is given as "010" and primary inputs of A (LSB, LSB+2....LSB+14) and inputs of B(LSB+1,LSB+3,...MSB) are given as "010". Coming to the other inputs of B(LSB,LSB+2...LSB+14) and inputs of A(LSB+1,LSB+3,...MSB) are given as "101" to estimate the final carry output delay. This test pattern will measure propagation delay from low to high and vice-versa.

Table 25 shows proposed design for 4-bit adder has less power consumption, low delay and low power delay product when compared to two existing designs. Power consumption, delay, power delay product is less in (Kandpal *et al.*, 2020) when compared to all the existing designs in the Table 25. All the results were obtained by considering the pulse width of 1ns and with rise and fall time as 100 ps.

Table 26 shows proposed design for 8-bit adder has less power consumption, low delay and low power delay product when compared to two existing designs. Delay, power delay product is less in (Kandpal *et al.*, 2020) when compared to all the existing designs in the Table 26. Power consumption is less in (Hasan *et al.*, 2020) in comparison with existing designs. All the delay results were taken with pulse width of 2 ns and 100 ps as rise and fall time. Power results were obtained with 10 ns pulse width and 1 ns as rise and fall time.

Table 27 shows proposed design for 16-bit adder has less power consumption, low delay and low power delay product when compared to three

existing designs. Delay is less in (Mohanty and Patel, 2014) when compared to all the existing designs in the Table 27. Power consumption and power delay product is less in (Hasan *et al.*, 2020) in comparison with existing designs. All the delay results were taken with pulse width of 4 ns and 100 ps as rise and fall time. Power results were obtained with 10 ns pulse width and 1ns as rise and fall time.

Table 28 shows the proposed design for 32-bit adder has less power consumption, low delay and low power delay product when compared to three existing designs. Delay is less in (Mohanty and Patel, 2014) when compared to all the existing designs in Table 28. Power consumption and power delay product are less in (Hasan *et al.*, 2020) in comparison with existing designs. All the results were taken with pulse width of 32 ns and 1,000 ps as rise and fall time.

Table 28. Performance comparison for various 32-bit adders in terms of power, delay and power delay product

Design	Performance		
	Power (mw)	Delay (ps)	Power delay product (fj)
Proposed	6.63	3,890	25,790
Mohanty and Patel (2014)	11.2	2,660	29,792
Kandpal <i>et al.</i> (2020)	15.5	5,450	84,475
Hasan <i>et al.</i> (2020)	4.61	5,260	24,248
Hamed and Somayeh (2018)	8.65	6,690	57,868

Conclusions

In this paper, partial products of adder were partitioned to have a reduced fan-in at 4-bit stage. This 4-bit adder is used as a basic building block for implementing 8, 16, 32-bit adders. Adder has been analyzed for performance and delay. Proposed adder is highly optimized with respect to delay and for about 26% to 42% reduction in the delay has been achieved when compared with the three of the conventional adders for 32 bit in the recent literature. One of the Conventional adder has better power consumption and one more existing adder has better delay. The proposed adder has employed look ahead structure, where the conventional ones are with ripple carry and carry select structure.

References

- Aguirre-Hernandez, M. and Linares-Aranda, M. (2011). CMOS full-adders for energy-efficient arithmetic applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(4):4.
- Bhattacharya, P., Kundu, B., Ghosh, S., Kumar, V., and Dandapat, A. (2015). Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(10):8.
- Chang, C.H., Gu, J., and Zhang, M. (2005). A review of 0.18- μ m full adder performances for treestructured arithmetic circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(6):10.
- Chowdhury, S.R., Banerjee, A., Roy, A., and Saha, H. (2008). A high speed 8 transistor full adder design using novel 3 transistor XOR gates. *Int. J. Elect. Cir. Sys*, 2(4):7.
- Hamed, N. and Somayeh, T. (2018). Low-power and fast full adder by exploring new XOR and XNOR gates. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(8):13.
- Hasan, M., Hossein, M.J., Hossain, M., Zaman, and H.U., Islam, S. (2020). Design of a scalable low-power 1-bit hybrid full adder for fast computation. *IEEE Transactions On Circuits And Systems -II: Express Briefs*, 67(8):5.
- Hassoune, I., Flandre, D., O'Connor, I., and Legat, J.D. (2010). ULPPA: A new efficient design of a power-aware full adder. *IEEE Transactions on Circuits Systems. I, Reg. Papers*, 57(8):9.
- Kandpal, J., Tomar, A., Agarwal, M., and Sharma, K.K. (2020). High-speed hybrid-logic full adder using high-performance 10-T XOR-XNOR cell. *IEEE transactions on VLSI systems*, 28(6):10.
- Kumar, P. and Sharma, R.K. (2016). Low voltage high performance hybrid full adder. *Engineering, Sci. Technol. Int. J.*, 19(1):7.
- Miao, J. and Li, S. (2017). A novel implementation of 4-bit carry look-ahead adder. In: *International Conference On Electron Devices And Solid-State Circuits (EDSSC)*, Oct 18-20, Hsinchu, Taiwan, p. 2.
- Mohanty, B.K. and Patel, S.J. (2014). Area-delay-power efficient carry-select adder. *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 61(6):5.
- Parameshwara, M.C. and Srinivasaiah, H.C. (2017). Low- Power Hybrid 1-Bit Full-Adder Circuit for Energy Efficient Arithmetic Applications. *Journal of Circuits, Systems, and Computers*, 26(1):1750014.
- Vesterbacka, M. (1999). A 14-transistor CMOS fulladder with full swing nodes. In: *Proc. IEEE Workshop on Signal Process. Systems*; 22-22 Oct, 1999; Taipei, Taiwan, p. 10.
- Wairya, S., Nagaria, R.K., and Tiwari, S. (2011). New design methodologies for high-speed low- voltage 1 bit CMOS Full Adder circuits. *Int. J. Comput. Technol. Appl*, 2(2):9.
- Weste, N. and Eshraghian, K. (1985). *Principles of CMOS VLSI Design*. New York, NY, USA: Addison-Wesley, 713p.