IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DESIGN OF A FULL ADDER WITH FAULT-TOLERANT USING A PARALLEL ADDER

[1]S TEJASWINI, [2]DR. BINDU. S

[1]PG Student, VLSI & Embedded Systems, Dept. of ECE, B.N.M Institute of Technology, Bangalore, India

[2]Professor, Dept. of ECE, B.N.M Institute of Technology, Bangalore, India

Abstract: The creation of mobile complementary metal oxide semiconductors for signal and image processing applications requires reduced transistor count and switching. The adder serves as the foundation for all mathematical operations performed by the processing units. The accuracy and size of an error-tolerant application for a parallel adder are optimized using a faithful approximation. The parallel adder suggested the carry select addition generates the vast majority of n/2m blocks with the right logic. The least significant approximation component of the adder is constructed using the EFA and FTFA cell**.**

Index Terms - Carry select adder, Ripple carry adder, EFA design with parallel adder, FTFA1, FTFA2.

I. INTRODUCTION

The basic unit of arithmetic operations in signal processing, digital image processing, multimedia processing, and other applications in addition. All of these applications require multi-bit adders and using long ripple carry adders presents significant carry propagation delay problems (RCA). The resident parallel adders in digital systems carry out a quick and dependable function.

For VLSI implementation, parallel adders are used as simple cells with regular connections between them. Adder suggested the method of reducing the delay propagation and it is independently manufacturing the multiple carries it generates the bit of the sum simultaneously. It is suggested that parallel addition employing a CSLA can reduce the latency of the carry propagation in the area. Due to twin RCA architectures' tendency to consume space in standard CSLA, gate-level adjustments are made to minimize silicon area.

A proposal is made for an add-one circuit, a single RCA array, a carry select parallel adder, and Mux units. The design is suggested to use an add-one circuit with a single multiplexer-based RCA configuration.

1.1 RIPPLE CARRY FULL ADDER

A ripple carry adder is a digital circuit that adds two binary integers arithmetically. It can be built with full adders connected in a cascade, each full adder's carry output feeding into the carry input of the one behind it. Four full adders (FA) circuits are connected to create a 4-bit ripple carry adder.

Due to the fact that the first block represents the least significant bit, the input comes from the LSB. The important bits of the numbers to be added in the example are represented by a0 and b0. The bits s0 to s3 represent the output sum.

1.2 CARRY SELECT ADDER

The Carry Select Adder is a fast adder utilized in digital communication and memory architectures. One ripple carry adder's Carry will be "0" and another will be "1," respectively. The 2 to 1 multiplexer in this case identifies the output sum and carry. Carry can serve as a representation of the multiplexer's control signal (Cin).

The fundamental building block of a carry-select adder, with a block size of 4, is shown below Figure 1.1. The carry and sum bits produced by the multiplexing of two 4-bit ripple-carry adders are chosen by the carry-in. The intended outcome is obtained by choosing which ripple-carry adder has the correct assumption via the actual carry-in since one ripple-carry adder assumes a carry-in of 0 while the other assumes a carry-in of 1.

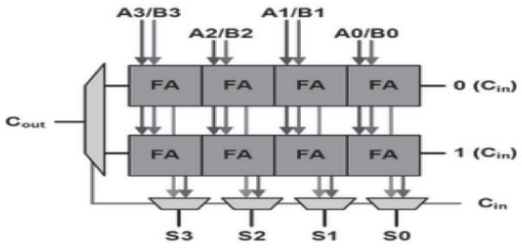


Figure 1.1: Carry Select Adder

II. IMPLEMENTATION

2.1 PROPOSED SYSTEM

We provide an approximation of the parallel adder that can while keeping the largest deviation within bounds, enhance area and accuracy. Due to its small hardware footprint and regular arrangement of full adder cells, traditional RCA is better suited for multi-bit addition, although it has a large propagation delay. RCA uses the Concatenated n bit of full adder cells can take two n-bit operands as input and output is n + 1 bits by connecting the carry-out bit of one FA cell to a carry-in bit of the subsequent FA cell.

$$t_{RCA} = n \cdot t_{FA}$$

Where tFA is delay carry propagation of 1-bit full adder.

2.2 PROPOSED PARALLEL ADDER

The addition of two one-bit values plus an input carry is handled by a single complete adder. A Parallel Adder, on the other hand, is a digital circuit that can operate on equivalent pairs of bits in parallel to calculate the arithmetic sum of two binary integers that are longer than one bit. It is shown in below Figure 2.1

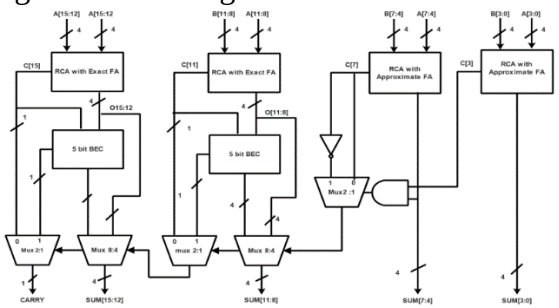


Figure 2.1: Block diagram of the parallel adder for n-16

A 16-bit complete adder is being created adders group together at 4 and is separated into 4, 4, 4, 4. Small models make up the larger model. Here, instead of splitting into 4,4,4,4, the first block should be made up of 0 to 3 bits, the second block of 4 to 7 bits, the third block of 8 to 11 bits, and the fourth block of 12 to 16 bits. For an extra conversion, these outputs are routed to BEC units. By using the multiplexer, it chooses either the output adder or output of the BEC depending upon the preceding the carry block to the output comes of the sum [15:12] and [11:8], and output carry. Parallel adder suggested the two scenarios are EFA and the FTFA designs.

2.3 EFA DESIGN WITH PARALLEL ADDER

In the suggested full adder design, also known as suggested EFA, the exact component and the least significant approximation portion are implemented with exact FA cells. The sum and carry logic equations for the precise full adder are given by,

$$Sum = A \wedge B \wedge C$$

$$Carry = A \cdot B + B \cdot Cin + A \cdot Cin$$

The implementation of gate level exact FA is shown in Figure 2.2,

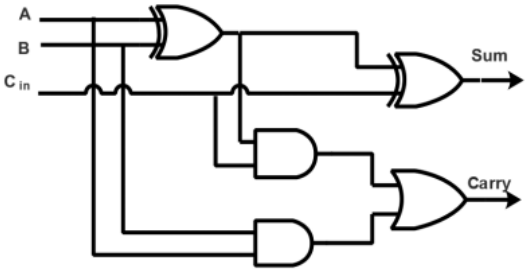


Figure 2.2: Logic diagram of exact FA

2.4 EXACT PARALLEL ADDER DESIGN WITH FTFA1

The design of the FTFA cells is created by using an EFA cell, while an approximate is created using the FTFA cells. Two different FTFA cells are utilized to demonstrate the efficiency of the suggested approach in maximizing area and accuracy. The probabilities are P(Serror)=1/4 and P(Cerror)=1/8, The design of the parallel adder FTFA1 architecture has one error in Carry and two faults in Sum introduced by the FTFA. Logic statements for the FTFA cell's Sum and Carry outputs are provided by,

$$Sum = A \wedge (B + Cin)$$
$$Carry = A + (B \cdot Cin)$$

The implementation of gate level logic FTFA1 is shown in Figure 2.3

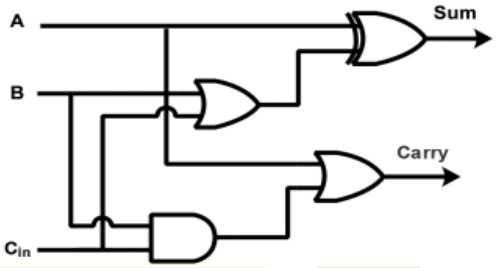


Figure 2.3: Logic diagram of FTFA1

2.5 EXACT PARALLEL ADDER WITH FTFA2

The design of parallel adder FTFA2 has a probability of P(Serror/Cerror) of 1/4, and the FTFA2 has two errors in the Sum and Carry output.

The equation of the FTFA2 cell is given by,

$$Sum = (A \wedge B) * (\sim Cin)$$
$$Carry = Cin$$

The implementation of gate level logic FTFA2 is shown in Figure 2.4

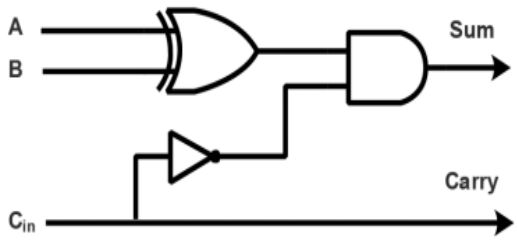


Figure 2.4: Logic diagram of FTFA2

III RESULTS AND DISCUSSION

The designed Verilog code is simulated, verified in Xilinx Isim, and synthesized in the simulator.

3.1 RIPPLE CARRY ADDER (4-bit)

RCA is defined as a digital circuit it produces the sum of two binary numbers. The carry output from each full adder is connected to the input of the following FA in the chain, forming a cascaded connection between the full adders. Figure 3.1 shows the waveform,

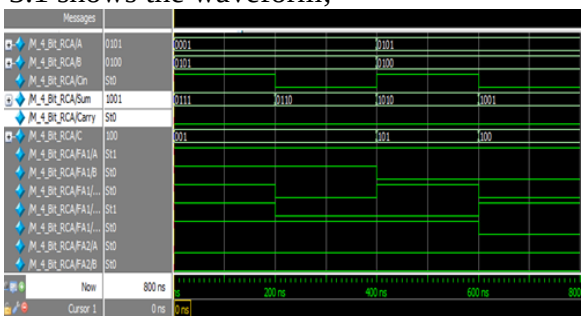


Figure 3.1: Simulation of Ripple Carry Adder

3.2 CARRY SELECT ADDER WITH BEC N=16

The binary format's three inputs and two outputs are added to create the CSA. It is utilized in the fast multipliers last stages to add the partial products and produce the final result with a 16-bit binary to the excess converter. The waveform is shown in below Figure 3.2,

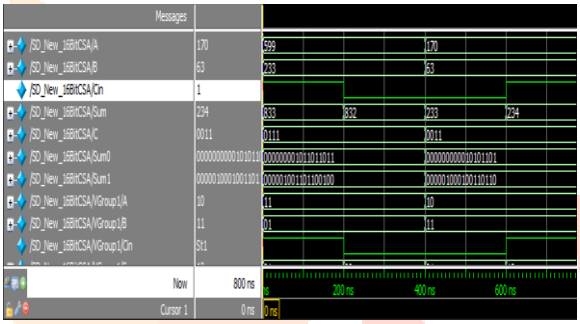


Figure 3.2: Simulation of proposed to carry select adder BEC N=16

3.3 CARRY SELECT ADDER WITH BEC N=16 APPROXIMATE FTFA1

The inputs are A, B, Cin, and outputs are sum, carry less approximation, and differentiation are performed on the n=16 bits of the FTFA 1 in the proposed carry select adder employing BEC. The waveform is shown in below Figure 3.3,

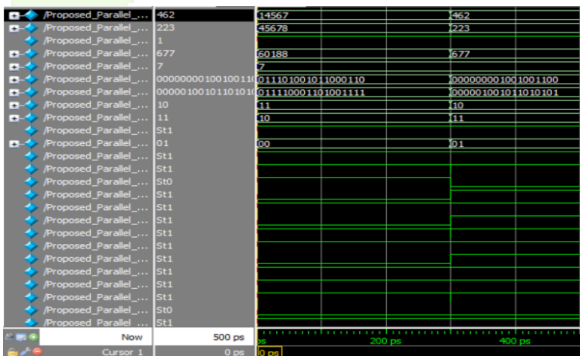


Figure 3.3: Simulation of Proposed Carry Select Adder BEC N=16 Approximate FTFA1

3.4 CARRY SELECT ADDER WITH BEC N=16 APPROXIMATE FTFA2

The accuracy of the FTFA 2 design is superior to the FTFA 1 design in this simulation of the proposed carry select adder BEC N=16 bits. The waveform is shown in below Figure 3.4,

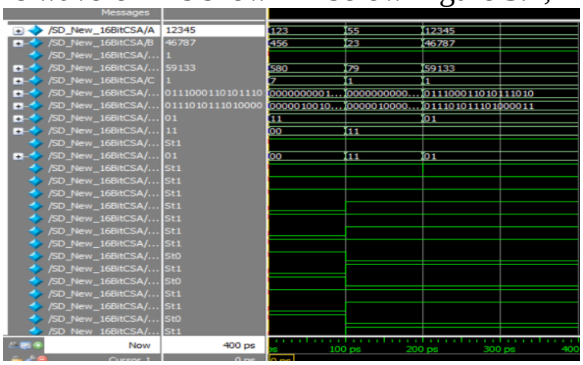


Figure 3.4: Simulation of Proposed Carry Select Adder BEC N=16 Approximate FTFA2

3.5 SYNTHESIS RESULTS

The project simulation is carried out solely using the ModelSim software, which verifies if the code is accurate or not in order to find the area and delay using the Xilinx ISE tool. The next step in the synthesis process involves the use of Xilinx software to calculate Area, Delay, and Power

AREA

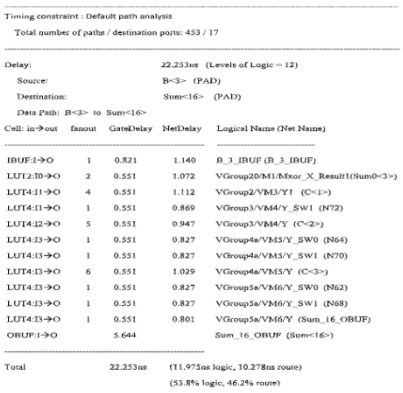
The parameters of the region are the number of LUTs, gates, and slices. These are the three factors that go into calculating the area.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Notes(s)**
Number of Slice Flip Flops	1	55,296	1%	
Number of 4 input LUTs	302	55,296	1%	
Logic Distribution				
Number of occupied Slices	156	27,648	1%	
Number of Slices containing only related logic	156	156	100%	
Number of Slices containing unrelated logic	0	156	0%	
Total Number of 4 input LUTs	306	55,296	1%	
Number used as logic	302			
Number used as a route-thru	4			
Number of bonded IOBs	178	633	28%	
IOB Flip Flops	125			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	3,705			
Additional JTAG gate count for IOBs	8,544			

Figure 3.5: General timing report specifies the report of the total area

DELAY

Overall delay is the result of the gate delay and the path delay. The latency was found to be 22.253ns. This is the 16-bit CSLA delay



```
Timing constraint : Default path analysis
  Total number of paths / destination ports: 453 / 17
--------------------------------------------------------------------------------
Delay:                  22.253ns  (Levels of Logic = 12)
  Source:               B<3>  (PAD)
  Destination:          Sum<16>  (PAD)
  Data Path: B<3> to Sum<16>
Cell: in->out  fanout  GateDelay  NetDelay  Logical Name (Net Name)
----------------------------------------  ------------
IBUF:I->O        1     0.821     1.140   B_3_IBUF (B_3_IBUF)
LUT2:I0->O       2     0.551     1.072   VGroup20/M1/Mxor_X_Result1(Sum0<3>)
LUT4:I1->O       4     0.551     1.112   VGroup2/VM3/Y1  (C<1>)
LUT4:I1->O       1     0.551     0.869   VGroup3/VM4/Y_SW1  (N72)
LUT4:I2->O       5     0.551     0.947   VGroup3/VM4/Y  (C<2>)
LUT4:I3->O       1     0.551     0.827   VGroup4a/VM5/Y_SW0  (N64)
LUT4:I3->O       1     0.551     0.827   VGroup4a/VM5/Y_SW1  (N70)
LUT4:I3->O       6     0.551     1.029   VGroup4a/VM5/Y  (C<3>)
LUT4:I3->O       1     0.551     0.827   VGroup5a/VM6/Y_SW0  (N62)
LUT4:I3->O       1     0.551     0.827   VGroup5a/VM6/Y_SW1  (N68)
LUT4:I3->O       1     0.551     0.801   VGroup5a/VM6/Y  (Sum_16_OBUF)
OBUF:I->O              5.644             Sum_16_OBUF  (Sum<16>)
----------------------------------------  ------------
Total                 22.253ns  (11.975ns logic, 10.278ns route)
                                (53.8% logic, 46.2% route)
```

Figure 3.6: General timing report specifies the report of the overall delay

POWER

The 16-bit CSLA's power consumption is specified in the timing report. The estimated power is 37 pW.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		37
Vccint 1.20V:	10	12
Vccaux 2.50V:	10	25
Vcco25 2.50V:	0	0
Inputs:	0	0
Logic:	0	0
Outputs:		
Vcco25	0	0
Signals:	0	0
Quiescent Vccint 1.20V:	10	12
Quiescent Vccaux 2.50V:	10	25

Figure 3.7: General timing report specifies the report of the total estimated power

3.6 COMPARISON OF ADDERS

Table 1: Comparison of adders

Spartan 3 XC3S 5000 4FG 1156	AREA			DELAY		
	LUT	Slices	Gates	Overall Delay	Gate Delay	Path Delay
Propsed Carry Select Adder BEC n=16	48	28	294	23.646ns	12.596ns	11.050ns
Proposed Carry Select Adder BEC n=16 Approx FTFA_1	46	27	276	23.977ns	12.526ns	11.451ns
Proposed Carry Select Adder BEC n=16 Approx FTFA_2	44	24	264	22.253ns	11.975ns	10.278ns

The accuracy of the FTFA 2 design is superior to the FTFA 1 design in this simulation of the proposed carry select adder BEC N=16 bits. Overall delay is the result of the gate delay and the path delay. The latency was found to be 22.253ns. This is the 16-bit CSLA delay.

Area

When compared to FTFA1 and EFA utilizing the BEC n=16 of carry select adder, the FTFA2 design is more advanced. The representation of the area is shown in above Figure 3.8

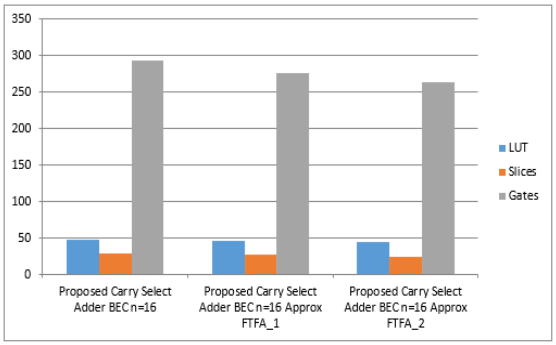


Figure 3.8: Pictorial representation of area

Delay

Overall delay is the combination of the gate delay and the path delay. The latency was found to be 22.253ns. The FTFA 2 design is superior to the FTFA 1 design in this simulation of the proposed carry select adder BEC N=16 bits. The representation of the delay is shown in below Figure 3.9,

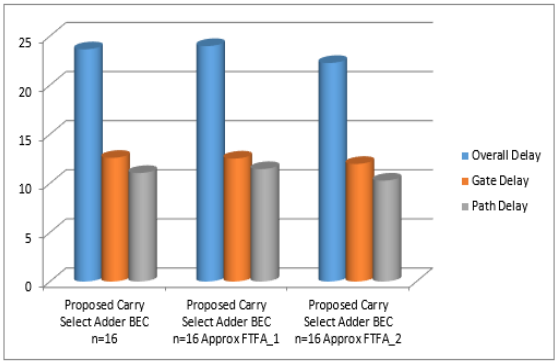


Figure 3.9 : Pictorial representation of delay

IV CONCLUSION

For error-tolerant applications, the efficiency of the adder combination of the carry bypass and CSA choose the algorithm for addition with the faithful approximation. The FTFA2 design performs better in the efficiency of the area and the delay whereas the design of exact FA design is better in accuracy, according to the implementation of the algorithm in n=16 and m=4. The design of FTFA1 can improve both accuracy and delay. The FTFA2 design performs the approximate logic of the area and delay compared to both FTFA1 and exact FA design. The FTFA2 design is advanced using the CSLA of the BEC n=16.

REFERENCES

1] High-Performance Error Tolerant Adders for Image Processing Applications, C. Vasanthanayaki, R. Jothin, 2018, IETE Journal of Research Volume 67, 2021 – Issue 2.

2] High-Speed Energy Efficient Static Segment Adder for Approximate Computing Applications, R. Jothin, C. Vasanthanayaki, 2017, Journal of Electronic Testing: Theory and ApplicationsVol. 33, No. 1

3] A power constrained approximate multiplier with a high level of configurability, Mostafa Abbasmollaei, Fahimeh Hajizadeh SiamakMohammadiaMohammadreza BineshMarvastibSeyyed AmirAsgharibBehrouzSamieiyanc, 2022, Microprocessors and Microsystems Volume 90, April 2022, 104519

4] Investigation of Error-Tolerant Approximate Multipliers for Image Processing Applications, D. Tilak Raju1 and Y. Srinivasa Rao, 2019, Researchgate

5] Configurable DSI partitioned approximate multiplier,FahimehHajizadehaMohammadrezaBinesh MarvastiaSeyyed AmirAsghariaMostafa Abbas Mollaei Amir M. Rahmanic, 2020, Future Generation Computer Systems Volume 115, February 2021, Pages 100-114

6] Performance Evaluation of Fault-Tolerant Approximate Adder, Tanya Mendez; Subramanya G. Nayak, 2022, 2022 6th International Conference on Devices, Circuits and Systems (ICDCS).

7] Design and Implementation of Low Power and Area Efficient Architecture for High-Performance ALU, Usthulamuri Penchalaiah and V. G. Siva Kumar, 2021, IETE Journal of Research