jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2019, VOL. 65, NO. 3, PP. 477-483
Manuscript received December 19, 2018; revised August, 2019. DOI: 10.24425/ijet.2019.129802

Low Power and Improved Speed Montgomery Multiplier using Universal Building Blocks

P. Velrajkumar*, C. Senthilpari, J. Sheela Francisca, and T. Nirmal Raj

***Abstract*—This paper describes the arithmetic blocks based on Montgomery Multiplier (MM), which reduces complexity, gives lower power dissipation and higher operating frequency. The main objective in designing these arithmetic blocks is to use modified full adder structure and carry save adder structure that can be implemented in algorithm based MM circuit. The conventional full adder design acts as a benchmark for comparison, the second is the modified Boolean equation for full adder and third design is the design of full adder consisting of two XOR gate and a 2-to-1 Multiplexer. Besides Universal gates such as NOR gate and NAND gate, full adder circuits are used to further improve the speed of the circuit. The MM circuit is evaluated based on different parameters such as operating frequency, power dissipation and area of occupancy in FPGA board. The schematic designs of the arithmetic components along with the MM architecture are constructed using Quartus II tool, while the simulation is done using Model sim for verification of circuit functionality which has shown improvement on the full adder design with two XOR gate and one 2-to-1 Multiplexer implementation in terms of power dissipation, operating frequency and area.**

***Keywords*—FPGA, Model Sim, Power dissipation, Speed, Universal Logic**

I. Introduction

AN American mathematician Peter L. Montgomery introduced a way to perform faster modular multiplication by using Montgomery modular multiplication. Montgomery algorithm performs the X*Y mod M fastest even when large values are assigned to X, Y and M [1]. The value of X and Y are transformed into another domain known as the Montgomery form. The Montgomery multiplication A constant R is introduced, which is responsible for the transformation between the Montgomery world and the real world. The Montgomery form of X is X.R mod M. The constant R depends on the architecture of the hardware used and the value of M. All the computation which involves the reduction is done in the Montgomery form. The results are transformed from the Montgomery form to the real world. This gives the modular product of X*Y mod M.

A MM has a complex algorithm and a large hardware configuration. The power dissipation and total gate count of the MM is high. This leads to inefficient performance in terms of power dissipation and speed. The MM circuit is complex network due to the type of carry adder block used and the fundamental design of full adder block used in the carry adder design. The use of complex gates produces a MM circuit with less operating frequency. The main aim of this research is to modify and design an efficient adder circuit which is used in the MM circuit; implement an efficient Carry Save Adder (CSA) to the adder circuit. Various adder circuits like Carry Skip Adder, Manchester Carry Chain Adder, Carry Propagate Adder and Carry Ripple Adder are all investigated but cannot be used to replace the existing CSA block. Implementing Universal Building Block such as NOR gate and NAND gate is to realise the architecture of CSA that is implemented in the MM hardware circuit. After the circuit design of the arithmetic circuit is completed successfully, the schematic and VHDL code of the design is simulated and analysed. The modified circuit is examined with existing circuits and then graphical representation of simulation is analysed. This paper is aided by Quartus II CAD system. This software helps in designing circuits that are implemented by using FPGA devices

P.Velrajkumar is with Department of Electrical & Electronics Engineering, CMRIT, Bangalore, India. (velrajkumar.p@cmrit.ac.in).
C.Senthilpari & J.Sheela Francisca are with Faculty of Engineering, Multimedia University, Malaysia.
T.Nirmal Raj is with School of Electrical & Electronics Engineering, Sastra University, India.

II. Design Overview

Asirbad Behera, et.al (2014), found that the minority based bridge style full adder has greater circuit speed over virtually all testing conditions. Power analysis subsequently showed that the new design dissipated more power than minority based full adder, but it consumes less power than CLRCL adder. Although the silicon area consumed by CLRCL is less, by considering the PDP (energy) the new design is more efficient than other designs. Manoj Kumar et.al. (2013), analysed the Single bit full adder using eight transistors and proposed XNOR cell, which reduced power consumption with less number of transistors. Guilherme Perin et.al. (2010), Montgomery Modular Multiplication on Reconfigurable Hardware: Systolic versus Multiplexed Implementation [9] proved that Systolic architecture produces higher clock frequency compared to multiplexed architecture. According to Ref [10]. High speed radix-2 MM Architecture, Hybrid multiplier can give good results for operands with large number of bits. The main advantage is its high speed in VLSI design. The work can be further extended through analysis of power dissipation and hardware requirements of the multiplier and optimizing the three parameters for an efficient MM design. Sangeeta Rani, et.al. (2014) Comparison Analysis of 16-bit Adders [12], Carry Save Adder, Carry look ahead Adder, Ripple Carry Adder, in ISE XIILINX 10.1 by using HDL Verilog, Carry Save adder has the lowest delay among them and hence it is fastest of them all performance analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry Look Ahead (CLA) Adder [13]. Raminder Preet Pal Singh, Parveen Kumar, Balwinder Singh (2009) CLA logic for addition of partial product terms and CSA in partial product lines. Multipliers were all modelled using VHDL for 32-bit unsigned data. Implementation of Carry Save Adder logic in each partial product lines has better overall performance of multiplier unit as compared to CLA logic. Padma Devi et.al (2010), designed Carry Select Adder with decreased Area and Lower Power Consumption [14]. Carry Select Adders are good in power consumption and area.

III. DESIGN METHODOLOGY

The chronological development of the MM architecture design is discussed from the basic arithmetic elements in the circuits. The schematic design of arithmetic elements are designed initially, where these arithmetic blocks are simulated, compiled and integrated together to form high performance MM architecture. This paper can be classified into two sections, design and implementation. The basic arithmetic blocks of MM architecture is designed with the aid of Computer Aided Design (CAD) tool such as Quartus II. The schematic design of arithmetic circuits are developed on Quartus II and simulated in the same program. The input patterns of designs are verified by the inputs given in the truth tables of the respective designs. The layout of each design is generated by compiling the VHDL file. The layouts are simulated to acquire and evaluate the design parameters and performance factor of the design.

A. Arithmetic Element

The use of CSA block in this architecture is studied and modifications were done to further improve the efficiency of the adder. Different approaches in implementation of full adder architecture are executed, which results in improved adder block in MM architecture. The circuit designs and the functionality of each arithmetic block are verified thoroughly in Quartus II. The integration of the modified block completes the MM architecture.

Adder Design: There are various methodology and techniques used to enforce an adder circuit. In this research, the use of Universal Gate and other adder technique is used in designing the adder block.

Full Adder design I: The Sum circuit is fed with 3 input variables A, B and Cin. The Sum circuit is composed of XOR gate. Meanwhile, the Carry-out circuit is based on AND gate and OR gate [15]. The Carry-out circuit also uses input variables A, B and C_{in}. A total of three AND gates are used to realize this circuit. Each AND gate takes two inputs from the input variables. The use of OR and AND gate in the Cout circuit gives output of C_{out} = AB + AC + BC. The inputs of AB, AC and BC are parallel connection to succumb Carry-output.

***Full Adder design II*:** The full adder II design adopts a different Carry-out equation. The Carry-out formula is manipulated from equation (3) C= AB+BC+CA using algebraic manipulation. The Sum circuit is similar to the full adder design I which gives the output of S = A ⊕ B ⊕ C. Where else, the Carry-out circuit gives the output of C_{out}= AB + C (A ⊕ B). Input A and B are supplied to an AND gate and the output of A ⊕ B which are obtained from the Sum equation is used to supply to the Carry-out. This output is supplied as an input to an AND gate together with the Cin to realize the Carry-out equation. The operation of full adder is based on following alternative equation, given 3 single bit inputs as A, B, Cin and it generates 2 outputs of single bit Cout and Sum [7], where:

$$Cout = AB + (A \oplus B)C \quad (1)$$

This modification reduces the number of AND gates and OR gates used in the Carry-out equation.

Full adder design III: Full adder design III uses two XOR gates and one multiplexer to build a full an adder. A Boolean function that can provide this result is the exclusive-OR, XOR, symbolized by $A \oplus B$, which is 1 if either A or B is 1, but not both. It is easy to verify that $S = A \oplus B \oplus Cin$. The Cout is 1 whenever a majority of inputs are 1. A clever way to implement this is as $C = AB + (A \oplus B)Cin$. The single bit full adder operation can be explained as follows: the addition of 2 single-bit inputs A and B with carry Cin gives 2 single-bit outputs Sum and Cout, where

$$Sum = (A \oplus B) \oplus Cin \quad (2)$$

$$Cout = A.B + Cin(A \oplus B) \quad (3)$$

In the full adder design III, the Boolean expression can also be write as

$$Sum = X \oplus Cin = XCin' + X'Cin \quad (4)$$

where $X = A \oplus B$

$$Cout = A.B + A'.B.Cin + A.B'.Cin$$

$$Cout = (A'.B' + A.B).A + (A'.B + A.B').Cin$$

$$Cout = A \oplus X = A.X' + A'X \quad (5)$$

According to multiplexer, only the control input decides the output. The A⊕B is a control input of the mux circuit, which reduces the number of logic blocks in the design.

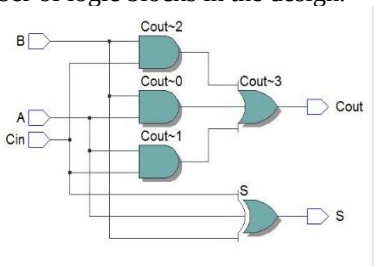

Fig. 1(a). Full Adder design circuit I

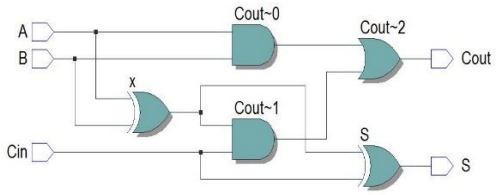

Fig. 1(b). Full Adder design circuit II

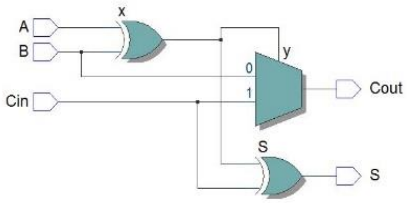

Fig. 1(c). Full Adder design circuit III

B. Universal Gate

Universal gate can be used to implement any Boolean function without the need of any other gate. Since NOR gate and NAND gate are basic gates used in digital logic design, the use of these gates to implement the XOR function in the full adder may help reduce the delay, increase the operating frequency and reduce the power dissipation of a full adder circuit.

***NOR Gate*:** All the full adder designs can be implemented using the NOR gate to replace the XOR function. The use of NOR gate can further reduce the propagation delay. Since XOR gate contributes the highest in propagation delay, the substitution of this gate can improve the design of the full adder.

***NOR gate implementation for full adder design I**:* As per standard equation, the full adder circuit is designed by using NOR gate implementation. Equation (6) and (7) are implemented in to Sum and carry design circuit with input variables A, B and C_{in}:

X = AB NOR (A NOR B) (6)

Sum = XC_{in} NOR (X NOR C_{in}) (7)

The Carry-out equation remains the same since AND gate and OR gate are also basic gates, the overall delay of the Carry-out circuit with the use of these gates is minimal. C_{out} = AB + AC + BC (8)

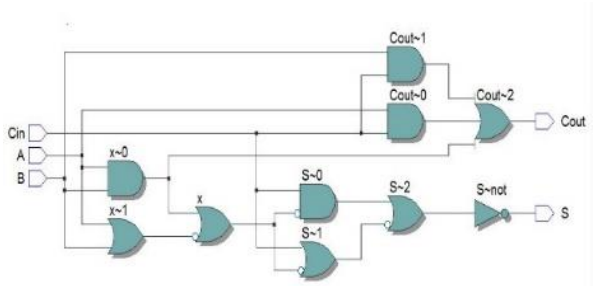

Fig. 2 (a). NOR Implementation for full adder design circuit I

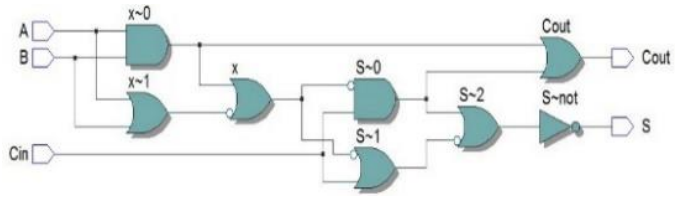

Fig. 2 (b). NOR Implementation for full adder design circuit II

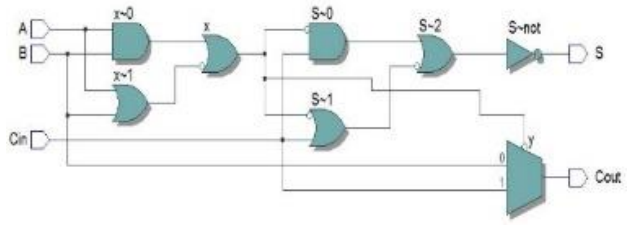

Fig. 2 (c). NOR Implementation for full adder design circuit III

NOR gate implementation for full adder design II: For the full adder design II, NOR gate is implemented in the SUM equation as done with full adder design I. The Carry-out equation remains the same as the one used in equation (3). The A ⊕ B is replaced with the NOR gate implemented equation from (3). Since A ⊕ B is also used in the Carry-out equation, the implementation of NOR gate also affects the performance of Carry-out circuit. The Carry-out equation is, C_{out} = AB + XC, where X = AB NOR (A NOR B).

NOR gate implementation for full adder design III: For the full adder design III, the NOR gate is also implemented in the SUM equation as done with full adder design I and II. The two XOR gate are replaced with NOR gates to realize the function of XOR gate. The equation (6) and (7) are used to realize the Sum circuit. The Carry-out equation remains the same as the one used in equation (5), where mux is used to select the Cout based on the value of the A ⊕ B.

NAND Gate: From other researches, it was found that the use of NAND gate is better compared to the NOR gate with respect to power consumption and propagation delay. NAND gate is used to realize the function of XOR gate in this section for all the full adder designs. One drawback of this design is that it uses three different types of gates, requiring three different IC packages, even though there are only five gates.

***NAND gate implementation for full adder design I*:** Similar to the NOR gate implementation, the NAND gate is implemented in the SUM equation where the use of two XOR gate are replaced with NAND gates to realize the function of XOR gate for full adder design I. Below is the equation of the Sum circuit with input variables A, B and C_{in}:

$$X = ((A \sim \&B) \sim \&A) \sim \&((A \sim \&B) \sim \&B) \quad (9)$$

$$Sum = ((X \sim \&Cin) \sim \&X) \sim \&((X \sim \&Cin) \sim \&Cin) \quad (10)$$

The Carry-out equation remains the same as the use of AND gate and OR gate contribute less propagation delay. C_{out} = AB + AC + BC.

NAND gate implementation for full adder design II: The NAND gate implementation for the full adder design II is also done in SUM circuit. The XOR gates from the full adder design II is replaced with the equivalent functionality with the use of NAND gate. The equation (8) and (9) are used to build the Sum circuit. The Carry-out circuit however differs since the use of NAND gate in equation is applied. The Carry-out circuit has an equation of C_{out} = AB + XC, where $X = ((A \sim \&B) \sim \&A) \sim \&((A \sim \&B) \sim \&B)$.

NAND gate implementation for full adder design III: For the full adder design III, NAND gate is implemented in the SUM circuit as well. The two XOR gate in the full adder design III is replaced with NAND gate implementation of XOR gate. This modification greatly reduces the propagation delay as this full adder architecture consists of only two XOR gates and a multiplexer. The equation (8) and (9) are again used to get the Sum circuit's output. The carry- out circuit uses equation (5) to determine the value supplied to the multiplexer.

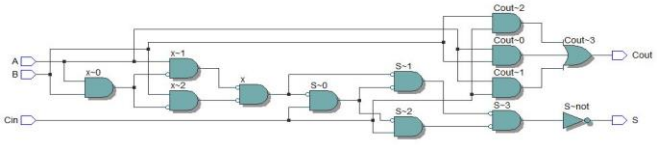

Fig. 3(a). Schematic of NAND implementation for full adder design I

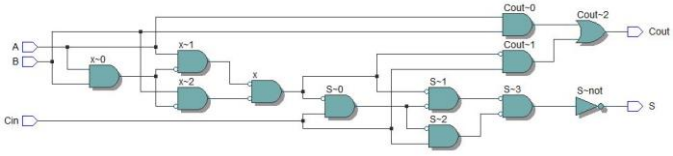

Fig. 3(b). Schematic of NAND implementation for full adder design II

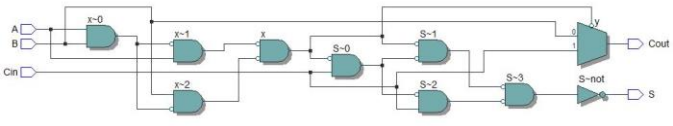

Fig. 3(c). Schematic of NAND implementation for full adder design III

C. Carry Save Adder

Since CSA performs addition without carry propagation, where its architecture uses full adder in parallel and no horizontal connection between the full adders makes CSA the most effective adder. In this paper, three different full adder designs with each design implemented with NOR gate and NAND gate are tested to examine which design is the most effective in the MM circuit.

Carry save adder with full adder design I*:* Full adder design I is the most conventional full adder. This full adder design acts as a benchmark to other full adder designs. The Sum circuit and Carry-out circuit based on equation (1) and (2) respectively are used to design this CSA. The CSA that is used in the Montgomery Multiplier consist of 8-bit binary input, which requires eight full adder block in one CSA architecture.

Carry save adder using full adder design I with NOR implementation: Full adder design I is implemented with using NOR gates to replace the XOR function for the Sum circuit. As mentioned earlier, the Sum circuit are designed based on equation (6) and (7). The use of NOR gate in this architecture will help to improve the propagation delay of the circuit. The Carry-out equation remains the same though the use of AND gate and OR gate have a low propagation delay.

Carry save adder using full adder design I with NAND gate implementation: In this design, NAND gates are used to realize the function of XOR gate. Two XOR gates are used to generate the Sum circuit, ten NAND gates are required to replace the functionality of the two XOR gates. Although the number of gates increase, the implementation of NAND gate will eventually reduce the propagation delay within the circuit since NAND gates are basic gates.

Carry save adder with full adder design II: Full adder design II is just a modified version of full adder design I. The main difference can be seen in the Carry-out circuit, where the circuit of A ⊕ B is supplied to the Carry-out equation. Since the Carry-out circuit incorporates part of the Sum circuit, the use of OR and AND gate in the Cout circuit are reduced. In this design, only two of each AND gate and OR gate are required.

Carry save adder using full adder design II with NOR gate implementation: Since NOR gate is a universal gate, where a universal gate can be used as a building block to create any function, the NOR gate is implemented to replace the function of XOR gate used in the Sum circuit of CSA adder circuit. The new Sum circuit uses the equations (6) and (7) to implement the Sum output with use of NOR gate. Each full adder design consists of four NOR gates to realize the Sum circuit. Since eight full adder blocks are used to build the CSA, the total NOR gate in this architecture is thirty-two.

Carry save adder using full adder design II with NAND implementation: In this circuit, full adder design II with NAND gate are used as a building block to create an 8- bit CSA. NAND gates are used to replace the more complex XOR gate to achieve a lower propagation delay. The total NAND gates used in this circuit is eighty.

Carry save adder with full adder design III: CSA with full adder design III uses two XOR gate and one multiplexer in each full adder block. This architecture creates the smallest full adder design with the least gate usage. Since the CSA takes 8-bit inputs, eight full adder blocks are integrated to produce the Sum and Carry-out. The Sum circuit of this design is based on equation (4) and the Carry-out circuit is based on equation (5). The result of A ⊕ B acts as select input for the multiplexer where if the result of A ⊕ B is '0', the input of B is produced out as the Carry-out and if the result of A ⊕ B is '1', the input of Carry in is supplied to the Carry-out. The Sum circuit consists of a XOR gate that takes two inputs, one from the consequent output of A ⊕ B and the other from the Carry in.

Carry save adder using full adder design III with NOR gate implementation: In this design, NOR gate is implemented to replace the function of XOR gate. The equations (6) and (7) are used to realize the Sum circuit and equation (5) is used to realize the Carry-out circuit. Since the full adder design III only consists of two XOR gates and a multiplexer, the use of universal gate in this architecture further improves the performance of the CSA with respect to power dissipation, operating frequency and propagation delay.

Carry save adder using full adder design III with NAND gate implementation: Since there are two universal gates, this design is built using NAND gate to substitute the use of XOR gate in the full adder circuit. Every full adder of design III has two XOR gates, when these gates are replaced with NAND gates; the total gate count is ten in each full adder block, which is shown in Fig. 4. Since NAND gate is a basic building block, the propagation delay is expected to reduce in this design. Since the input of the CSA is 8-bit, eight full adder blocks are required.

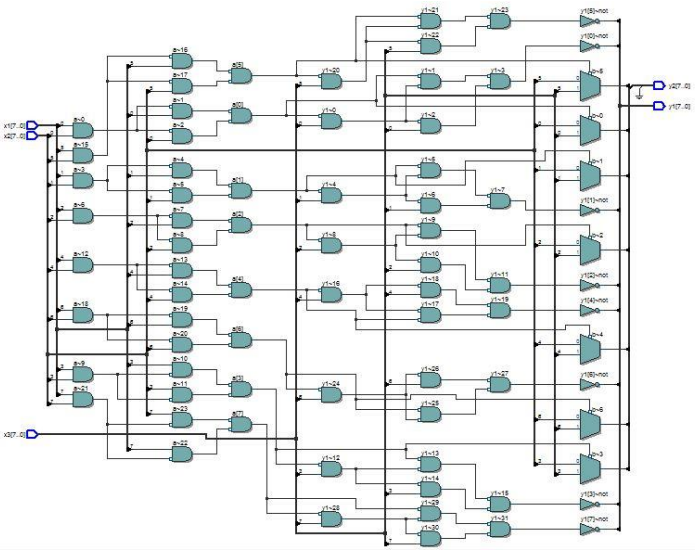

Fig. 4: Schematic diagram of CSA using full adder design III with NAND gate implementation

D. Montgomery Multiplier

One of the most reliable methods for performing modular multiplications in hardware is the MM [9]. Given 2 integers *X* and *Y*, the classical modular multiplication algorithm computes *XY* mod *N*. The MM circuit is build based on the Montgomery Multiplier algorithm.

Montgomery Multiplier Algorithm using two CSA blocks:

$$Inputs: X, Y, M \text{ with } 0 \le X, Y < M$$

$$Outputs: P = (X * Y(2^n)^{-1})\%M$$

$$N: \text{number of bits in } X;$$

$$X_i: i^{th} \text{ bit of } X;$$

s_0 : *Least Significant Bit of* S ;

$S := 0; C := 0;$

$For(i = 0 : i < n : i++)\{$

$S, C := S + C + x_i * Y;$

$S, C := S + C + s_0 * M;$

$S := S / 2; C := C / 2;\}$

$P := S + C;$

$if\ (P \geq M)\ then\ P := P - M$;

The architecture for implementation of loop of the Montgomery Multiplier algorithm is shown in figure 5 [9]. The layout comprises of two CSAs, registers for storing the intermediate results of the carry and sum. CSA's are the dominant occupiers of area in hardware mainly for very large values of *n*.

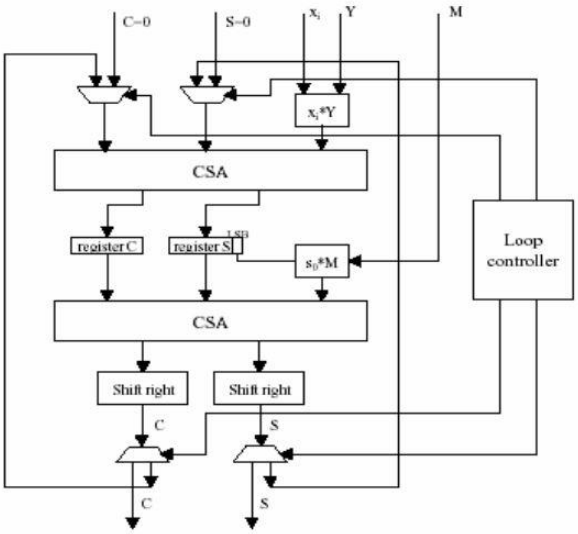

Fig. 5: The architecture of Montgomery Multiplier [3]

Montgomery Multiplier with Carry Save Adder Architecture:

A conventional Montgomery Multiplier comes with two CSA. Each CSA is supplied with three inputs and gives two outputs. A conventional Montgomery Multiplier has two CSA in its architecture. The CSA is supplied with 8-bit three input variables each where input is supplied bit by bit.. The modification done previously can be applied in this architecture to examine the difference in performance, delay, power consumption and power dissipation of each full adder design when applied to CSA architecture. Each and every CSA blocks designed earlier were used to build the MM circuit separately.

IV. RESULTS AND DISCUSSION

The results and discussion describe modified full adder circuit along with 8-bit Montgomery Multiplier architecture that are designed, compiled using Quartus II tool and verified using Model sim. FPGA board level analysis was done to evaluate parameters like operating frequency, power dissipation and number of ALMs used. The schematic designs of the modified full adder blocks and Montgomery Multiplier are constructed using Quartus II tool, Hence the compilation results for the analysis, synthesis and fitting were taken from that tool. The Modelsim tool is used for the timing diagram to verify the full adder design in accordance to the truth table. In accordance to the design methodology, relevant TCL scripts and test bench were created for each design for verification of the gate level behaviour simulation. The power dissipation results of each design were obtained using the Power Play, Power Analyzer tool in Quartus II for analysis purposes.

A. Full Adder

The full adder's designs are first compiled and simulated. The post fitting diagrams are similar for all the full adder design due to similar function of the full adder in generating the Sum and Carry-out. Although the design methods are different, the entire full adders are designed in accordance to the truth table. The full adder design timing diagram is shown in figure 6. Since the functionality of full adder designs is similar for each design, the timing diagram of each full adder design is similar to each other.

***Full Adder design I*:** According to design methodology, the full adder design I has evocated. A logic transition of the circuit has given three fan-in and four fan-out for Carry-out circuit and 1 fan-out for Sum circuit. The imbalanced Sum and Carry-out circuits give parametric results which are described according to Quartus II analysis and the results have been verified using ModelSim. This circuit gives total thermal power dissipation of 354.08 mW as per industry standard values (ISI).

***Full Adder design II*:** The full adder design II is designed using XOR-AND gate method. This method reduces logic count by using the reduction method. This makes the transition period to be very low when compared to the existing adder circuits. This balanced logic circuit gives lower power dissipation compared to full adder design I.

***Full adder design III*:** The full adder design III using XOR-MUX based circuits. Here the three inputs are fed in MUX and outputs are decided by the selection input which is A XOR B according to design concept. A huge reduction of logic blocks are seen in this design. This makes the propagation and logic transition of the circuit to be fast. The circuit delay is tremendously reduced compared to other adder-based circuits. As per view, this circuit seems to be imbalanced, but the Sum circuit is designed according to Boolean identities. According to fan-in and fan-out concept, this adder circuit has equal number which gives better result than other existing circuits.

B. Universal Gate

The full adder designs were implemented using the two universal gates namely the NOR gate and the NAND gate. The XOR gate is replaced using these universal gate; compilation and simulation are done using Quartus II and Model Sim to examine the results in comparison with the existing circuits.

***NOR Gate*:** The NOR gate implementation in full adder circuit is designed by using AND-OR-Invert (AOI) method. According to the full adder's basic truth table, the Sum circuit has to be logic '1' when either inputs are set to logically '1'. The AOI concept is translating the logic signal in the order of Gate Diffusion Input (GDI). Even though GDI logics are used,

the power dissipation and operating frequency are increased due to the increased number of logic block used. According to circuit in figure 4, the Carry-out has less logic gates compared to the Sum circuit. The basic Weinberger chart, the NOR gate implementation reduces the logical delay. The results are shown in table 1.

***NAND Gate*:** The NAND gate implementation in full adder is also based on AOI methods. Here, the OR logic blocks are implemented in terms of bubbled AND circuits. Even though the numbers of logic gates are increased, the power dissipation and operating frequency are moderate values compared with the existing circuits due to negative logic transition used.

TABLE I
SIMULATION RESULTS OF ALL FULL ADDER DESIGN USING NOR GATE IMPLEMENTATION

Adder type	Total thermal power dissipation mW	I/O thermal power dissipation, mW	Delay of hold timing, ns
FA I	354.08	4.63	22
FA II	336.27	4.23	19
FA III	279.50	2.50	14

C. Carry Save Adder

The CSA is simulated and verified using all the full adder design specified earlier. Since the carry save adder consist of 8 bit binary inputs, the total number of full adder required to assemble the carry save adder is eight. Each adder design contributes to different result in terms of power dissipation. Although different types of full adder are used, the functionality of the full adder is the same where the output generated from the input are based on the truth table. Full adder design III shows the best result in the use of carry save adder since the full adder design III only consist of two XOR gate and a 2-to-1 multiplexer. The simplicity of this design can be very useful when used as a basic building block in the construction of an 8-bit CSA. The results of all the full adder designs which are also designed using universal gates are tabulated in table 2.

D. Montgomery Multiplier

These arithmetic blocks are simulated according to the Montgomery Multiplier architecture. Each Montgomery Multiplier requires two adders' blocks, either the carry select adder or carry save adder to compute the addition of the two inputs supplied to the circuit. The MM timing diagram is shown in figure 6. The timing diagram is the same for all the designs used in Montgomery Multiplier since the functional aspect of each design are the same.

Compilation and simulation are done using carry save adder. The results obtained are tabulated in the table 3 shown. For verification of Montgomery Multiplier circuit functionality, two inputs are supplied to the circuit and simulated. The figure 6 illustrates one simulation of the MM where X= 10011111, Y = 11101001 and M= 239. The value of R is 2^8 for 8-bit input. When these values are supplied to the Montgomery Multiplier, the result after the computation is finished is 11010011, where the Montgomery Multiplier computes X.Y/R mod M. The value is verified using other Montgomery simulator to verify the result produced in this Montgomery Multiplier circuit.

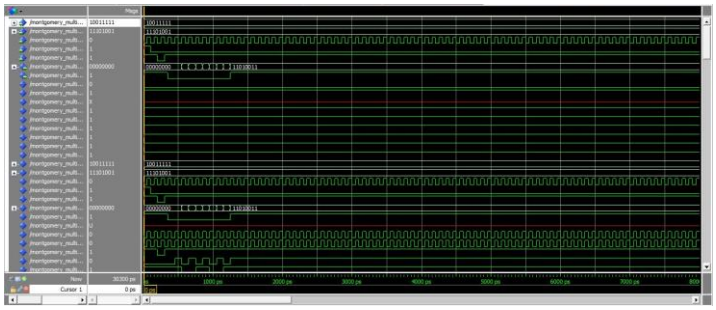

Fig. 6: Timing diagram of Montgomery Multiplier

TABLE II
SIMULATION RESULTS OF CARRY SAVE WITH CARRY SAVE ADDER

Adder type		TTPD, mW	I/O TPD mW	DH ns
Full adder design I	Con	355.86	6.40	22.30
	NOR	355.53	6.24	13.70
	NAND	355.53	6.24	13.70
Full adder design II	Con	355.29	6.08	20.20
	NOR	355.06	5.99	17.30
	NAND	355.06	5.99	17.30
Full adder design III	Con	354.65	6.01	7.90
	NOR	354.70	6.01	14.50
	NAND	354.70	6.01	14.50

TABLE III
SIMULATION RESULTS OF MONTGOMERY MULTIPLIER ADDER FOR ALL FULL ADDER DESIGNS

Adder type		TTPD, mW	MF, MHz	Logic ALM used	Total Labs	Total fan-out
	Con	355.47	211.15	48	7	457
FA1	NOR	355.42	344.95	41	6	202
	NAND	355.42	344.95	41	6	202
FA2	Con	355.47	213.63	48	6	458
	NOR	355.45	293.77	40	6	399
	NAND	355.44	299.22	40	6	399
FA3	Con	355.41	323.00	37	5	391
	NOR	355.47	323.42	37	5	391
	NAND	355.47	323.42	37	5	391

In terms of total thermal power dissipation, it is found that the use of different full adder designs and carry save adder design do not influence the result much. This is due to the 8-bit implementation of Montgomery Multiplier, higher bit implementation of Montgomery Multiplier show higher difference since the use of full adder blocks are directly proportional to the number of bits used. Due to software constrains in terms of higher bit implementation where the number of pins required for the FPGA increases as the number of bit increases. Due to this limitation, the designs were done using only 8 bit inputs. Only slight improvement can be seen in terms of total thermal power dissipation when universal gate were used for the implementation of full adder design I and II. Full adder design III shows the least thermal power dissipation which is 355.41mW for carry save adder implementation.

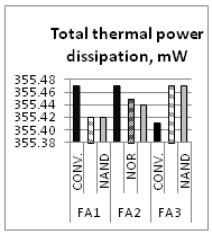

Fig.7 (a)

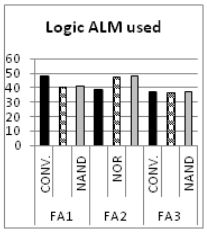

Fig.7 (b)

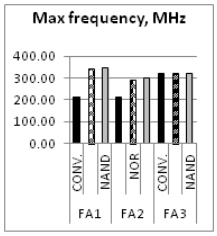

Fig.7 (c)

Fig 7 (a). Bar chart of total thermal power dissipation ofall full adder design in CSA for MM

Fig 7 (b). Bar chart of total logic ALM used for all full adder designs in CSA for MM

Fig 7 (c). Bar chart of maximum frequency of all full adder design in CSA for MM

Adaptive Logic Modules (ALMs) are basic building blocks, used to compose a logic array block. In this paper, ALMs are used to configure and implement arithmetic functions, logic functions and register functions in Quartus II tool. The higher the number of ALMs used reflects on the total area required by the circuit to implement its functionality. From the table 3, it can be observed that full adder design III uses the least amount of ALMs to produce the MM circuit. Since the number of ALMs used is smaller than the other full adder designs, the area required is also small for the full adder design III. The number of ALMs required to build the MM circuit is 37 for the full adder design III. The reduced number of gates and the simplicity of the full adder design III is the main contributing factor in the reduced number of ALMs required. In terms of maximum operating frequency, carry save adder has the highest value with the use of full adder design I implemented using the universal gates, which gives operating frequency is 344.95 MHz. Even though full adder design I with universal gates produces the highest operating frequency, full adder design III is better in terms of producing higher operating frequency for carry save adder design compared to other full adder design using the conventional gate implementation. This is due to the reduced number of gates required to compute the Sum and Carry-out circuit of the full adder. The simplicity of this design, incorporated with the use of 2-to-1 MUX produces a maximum operating frequency of 323.00MHz for CSA design in MM circuit. The use of universal gate implementation in this full adder design further improves the frequency to 323.42MHz for CSA implementation.

V. CONCLUSION

Three separate full adder design cells, full adder design I, design II and design III are designed with different number of gates and architecture. The full adder designs are used as the arithmetic block to build the CSA which is used to form the MM circuit. The full adder design III used in this paper consists of the least amount of gates with two XOR gates and a MUX. The CSA is a more conventional carry adder block used in MM circuit. The best operating frequency obtained from the result is 344.95 MHz with the use of full adder design I with universal gate implementation. But the overall performance of MM circuit in terms of operating frequency is better for the full adder designs used in CSA. It can be concluded that full adder design III applied in CSA block is more efficient in the application of MM circuit to achieve lower gate count of the circuit, lower path delay, lower power dissipation and higher operating frequency. All the schematic designs are designed in Quartus II and the simulation are done using ModelSim tool. Various types of simulation and analysis are carried out to validate and verify the functionality of each design.

REFERENCES

[1] Thomas Blum “Montgomery Modular Exponentiation on Reconfigurable Hardware” ARITH '99 Proceedings of the 14th IEEE Symposium on Computer Arithmetic, IEEE Computer Society Washington, DC, USA , 1999, pp-70-78.

[2] Quartus® II Introduction for VHDL Users, ALTERA® CORPORATION APRIL 2011.

[3] Amanor, David & Paar, C & Pelzl, Jan & Bunimov, V & Schimmler, Manfred. (2005). Efficient hardware architectures for modular multiplication on FPGAs. 2005. 539 - 542. 10.1109/FPL.2005.1515780.

[4] Tang, Guang-Ming & Takagi, Kazuyoshi & Takagi, Naofumi. (2017). 32×32-bit 4-bit Bit-Slice Integer Multiplier for RSFQ Microprocessors. IEEE Transactions on Applied Superconductivity. PP. 1-1. 10.1109/TASC.2017.2662700.

[5] Shinde, Kunjan D., Nidagundi, Jayashree C. (2014).Design of fast and efficient 1-bit full adder and its performance analysis, 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp- 1275 – 1279.

[6] Asirbad Behera, Manas Ranjan Jena, Abhinna Das, Narendra Kumar Pattnayak, (2014),“Design of an Efficient Dedicated Low Power High Speed Full Adder”, Journal of Embedded Systems, Archive, Vol. 2 , issue 3 2014, pp.35-28.

[7] Manoj Kumar , Sandeep K. Arya and Sujata Pandey (2011), “Single bit full adder design using 8 transistors with novel 3 transistors XNOR gate”, International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.4, 2011, pp 47-59.

[8] P. Montgomery (1985), “Modular multiplication without trial division,” Mathematics of Computation, vol. 44, pp. 519–521, 1985.

[9] Guilherme Perin, Daniel Gomes Mesquita, and João Baptista Martins (2011), “Montgomery Modular Multiplication on Reconfigurable Hardware: Systolic versus Multiplexed Implementation”, International Journal of Reconfigurable Computing, Volume 2011, Article ID 127147, 10 pages.

[10] C. Senthilpari, K. Diwakar and A.K. Singh, “Low power, low latency, high throughput 16 bit CSA adder using Non Clocked Pass-transistor logic”, Journal of Circuits, Systems and Computers, May 2009, Vol. 18, No. 03: pp. 581-596.

[11] C Senthilpari, G Ramanamurthy, P Velrajkumar, An efficient EPI and energy consumption of 32 bit ALU using Shannon theorem based adder approach, WSEAS Transactions on Circuits and Systems 10 (7), pp-231-238.

[12] Akbar Bermana (2012)“ A new simulation of a 16-bit Ripple Carry Adder and a 1-bit Skip Carry Adder” International journal of scientific &Technology Research Vol.1 Issue 2, 2012, pp-78-84.

[13] Raminder Preet Pal Singh, Parveen Kumar , Blwinder Singh (2009), “Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry-Look-Ahead Adder”, International Journal of Recent Trends in Engineering, Vol 2, No. 6, 2009, pp 83-86.

[14] Padma Devi, Ashima Girdher, Balwinder Singh, “Improved Carry Select Adder with Reduced Area and Low Power Consumption”, International Journal of Computer Applications (0975 – 8887) Volume 3 – No.4, 2010, pp 14-18.

[15] C Senthilpari A Low-power and High-performance Radix-4 Multiplier Design Using a Modified Pass-transistor Logic Technique, IETE Journal of Research 57 (2), pp 149-155.