International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 3 (March 2014), PP.55-63

Design of FIR Filter Using Modified Montgomery Multiplier with Pipelining Technique

Shobana.J[1], Ms.P.Kavipriya[2], Sarojini.K[3]
[1]Department of ECE, M.Tech VLSI Design, Sathyabama University, Tamilnadu, India
[2]Department of ECE, Senior Lecturer, Sathyabama University, Tamilnadu, India
[3]Department of ECE, M.Tech VLSI Design, Sathyabama University, Tamilnadu, India

Abstract:- This paper implements the FIR filter by using Montgomery multiplier. The main objective of this paper is used to reduce the area, delay and analyze the performance of the Montgomery multiplier. It performs the fastest multiplication of modular operation. Add the pipelining in Montgomery multiplier, it will improve the speed of the operation. The main application of the Montgomery multiplier is cryptography for encryption/decryption purpose.

Keywords:- Montgomery multiplier, FIR filter, Pipelining, Replication, VLSI.

I. INTRODUCTION

Multipliers are one of the key components in digital signal processing applications such as frequency domain filtering, frequency transformations etc. Montgomery multiplier is one of the modular multiplier. It performs modular multiplication operation. Peter Montgomery have been achieved a way to speed up arithmetic operation in which the modulus [3] is used for a long running computation. These methods have been explored as a hardware operation. It is particularly suitable for implementation on general purpose computers like signal processors or microprocessors. However Montgomery multiplication is easier to solve the side-channel attacks, so in some circumstances the Montgomery technique may be preferable.

The main objective is to design a FIR filter based on this Montgomery multiplication. Because FIR filters consist of multiplier, adder and unit delay. It is represented as finite impulse response and it is used to remove the unwanted frequencies from the original signal. The FIR filters mainly used for digital signal processors. To design Montgomery multiplier by using adders and shift registers and to include the pipelining, it will improve the speed of the operation. Because, the pipelining is used to improve the throughput of the design. We show the parametric design using pipelining and replication. The main advantage of the parametric design is their scalability and reusability. By replicating the design, it will increase the latency. Because, replication used to measure the latency and pipelining used to measure the throughput of the design. Modular multiplication is mainly used in encryption/decryption, authentication, cryptography, key distribution and many other applications. Montgomery multiplication also may be used for the basis for Montgomery exponentials.

This paper describes as follows, Section I implements the introduction, section II introduce the previous work and section III describes the pipelining. Section IV employs the Montgomery multiplication. Section V describes the design of FIR filter and section VI introduces results and discussions. Finally section VII concludes this paper.

II. EXISTING METHOD

In our works are represented in Montgomery multiplier with pipelining and replication features. It shows how a general algorithms consisting of a loop dependencies carried from one iteration to the next can be automatically mapped to a parametric hardware designs with pipelining and replication features [1]. In this model used to predict the area taken by the designs with less than 5% of error and their frequencies and throughput with less than 22% of error. They are developing a Montgomery multiplier with carry save adder based design. In this method number of clock cycles is reduced, so it is mainly improve the speed of the design. But, it covers more space in the design space exploration.

III. PIPELINING

Pipelining is an important technique used in several applications such as DSP systems, microprocessors etc. It originates from the idea of a water pipe with continuous water sent in without waiting for the water in the pipe to come out. Accordingly, it provides the results in speed enhancement for the critical path in most DSP systems. For example, it can either increase the clock speed or reduce the power consumption at the same speed in a DSP system. It has an several techniques, like, dynamic scheduling, loop unrolling, software pipelining etc.

Dynamic scheduling is the hardware re-arranging instruction execution to reduce the stalls while maintaining dataflow exception behavior. It is used to simplify the compiler of the operation. The operations over which the compiler has complete control when operations are executed. It eliminates the need for complex circuitry in the CPU, which frees up space and power for the functions including additional execution resources.

IV. MONTGOMERY MULTIPLICATION

The Montgomery multiplication is an efficient method for performing modular multiplications with an odd modulus. It replaced costly division operation with simple shifts [2]. This algorithm is mainly suitable for the implementations on general purpose computers.

Given two integers A, B with odd modulus M. The Montgomery multiplication Y can be represented as, [2]

$$Y = \text{montmulti}\ (A, B) = A \times B \times R^{-1} \bmod M$$

Given A, B < M and R is represented as,

gcd(R, M) = 1. Where $R = 2^n$ and M is an integer in the range $2^{n-1} < M < 2^n$, since R, it is sufficient that the modulus M is an integer. Since Montgomery multiplication is not an ordinary multiplication, there is a process of conversion between the ordinary domain and the Montgomery domain. The conversion between the ordinary domain and the Montgomery domain is given by the relation

$$A \Longleftrightarrow A' \text{ with } A' = A \times 2^{-n} \bmod M.$$

This conversion is represented as in Table I.

Table I: Conversion between ordinary domain and Montgomery domain.

ORDINARY DOMAIN	MONTGOMERY DOMAIN
A	$A' = A \times 2^{-n} \bmod M$
B	$B' = B \times 2^{-n} \bmod M$
AB	$(A * B)' = A{\times}B{\times}2^{-n} \bmod M$

The Montgomery multiplication provides the reduced product using a series of additions. Montgomery multiplication is particularly used in implementations of the RSA cryptosystem or other cryptosystems based on modular arithmetic.

Proposed Work:

To implement the pipelining and replication technique of our design, it is used to improve the speed of the operation. To replicate the block, the delay will be increase. So we add the pipelining in our block, it improve the speed and reduce the number of clock cycles. The 32-bit Montgomery multiplier can be represented in Fig. 1.

To design a new architecture of Montgomery multiplier with pipelining technique, which is used to multiply the two number representations with modulo M. we introduce a pipelining technique in our internal architecture which gives fast operation of our design. To use Montgomery multiplication, we must have the multiplicand (A) and multipliers (B) must be less than modulo M.

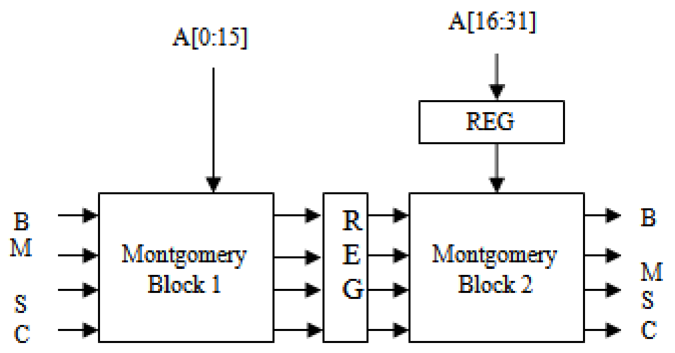


Fig. 1: Structure of a Montgomery multiplier with pipelining stages.

The cost of the modular multiplication is equal to the three integer multiplications plus cost of conversion to or from the Montgomery domain. But the cost of conversion is negligible compared to number of multiplications executed in the Montgomery domain. The internal structure of Montgomery multiplier is represented as shown in Fig. 2.

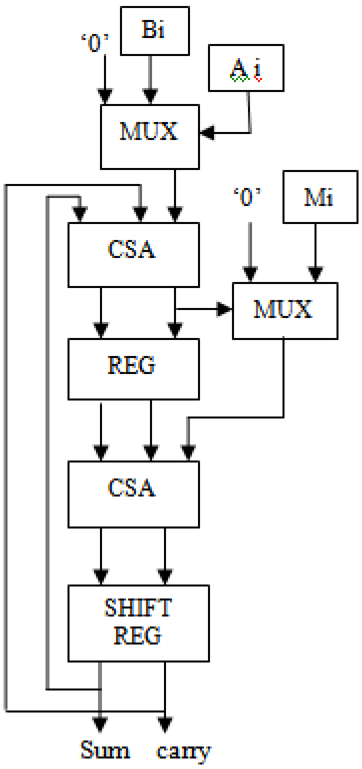


Fig. 2: Internal Structure of Montgomery Block.

- The modulo M is to be a prime to the radix, (i.e.,) there exists no common divisor for M and radix.
- The multiplicand and the multiplier (A, B) must be lesser than M.

Montgomery multiplication mainly used for encryption/decryption, cryptography, key distribution, authentication and many other applications. Montgomery multiplication also used for the Montgomery exponentials.

V. FIR FILTER

Digital filters are mainly used for Digital Signal Processor. It has two main uses, signal separation and signal restoration.

- Signal separation is needed when the signals have been contaminated with noise or other signals.
- Signal restoration is needed when the signal is distorted in some way.

The filters can be represented as in equation,

$$Y[n] = x[n] * f[n]$$
$$= \sum_{k=0} f(k)\, x(n-k)$$
$$= \sum_{k=0} x(k)\, f(n-k).$$

Digital filters having two types and it is represented as FIR filter and IIR filter.

- FIR filter is represented as Finite Impulse Response, whose impulse response or response to any finite length input of finite duration, because it settles to zero in finite time.
- IIR filter is represented as Infinite Impulse Response, whose impulse response is infinite duration.

Design of FIR filter:

FIR filters requires three basic building blocks in multiplication, addition and unit delay.

- DSP system of multiplication must be fast and must have sufficient to support the desired application.
- Signal addition is a very basic DSP function. In an FIR filter additions are required in combination with multiplications.
- The unit delay provides a one sample signal delay. A sample value is stored in a memory slot for one sample clock cycle.

Proposed Work:

To design a FIR filter by using our Montgomery multiplier architecture. We have to design a 8-tap FIR filter. In this filter gives better performance of the filter operation. The FIR filters can be represented as shown in Fig. 3.

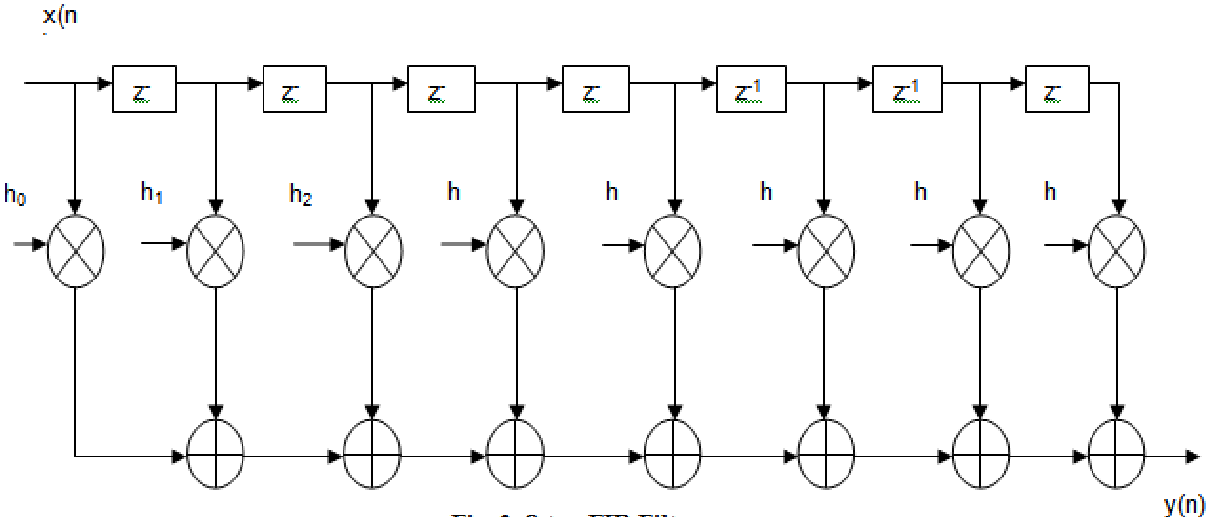


Fig. 3: 8-tap FIR Filter

FIR filters have some properties over IIR filter. it has require no feedback and it is inherently stable. They can be easily designed to be linear phase by making the coefficient sequence symmetric.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency cutoffs are needed. Many digital signal processors gives specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

VI. RESULTS AND DISCUSSIONS

Proposed Method:

Montgomery Multiplier:

The simulation results and device utilization summary of modified 32-bit Montgomery multiplier as shown in Fig. 4 & Fig. 5.

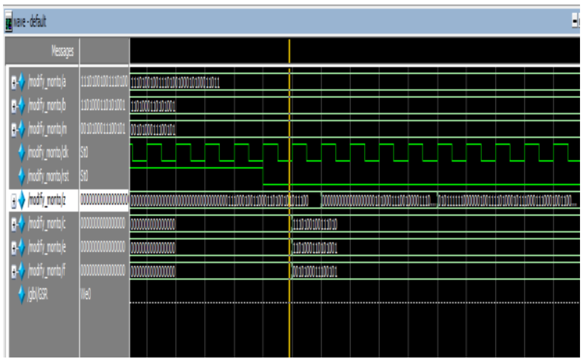
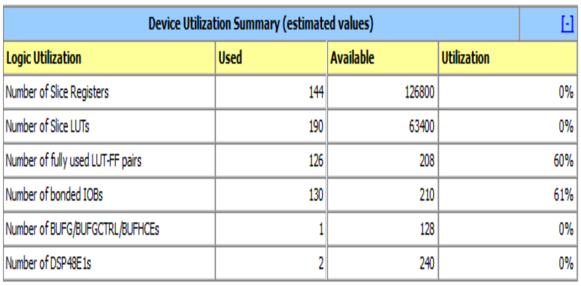


Fig. 4: simulation results of Montgomery multiplier.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	144	126800	0%
Number of Slice LUTs	190	63400	0%
Number of fully used LUT-FF pairs	126	208	60%
Number of bonded IOBs	130	210	61%
Number of BUFG/BUFGCTRL/BUFHCEs	1	128	0%
Number of DSP48E1s	2	240	0%

Fig. 5: Device Utilization Summary

The RTL schematic and synthesis report of modified Montgomery multiplier as shown in **Fig. 6, Fig.7 & Fig.8**.

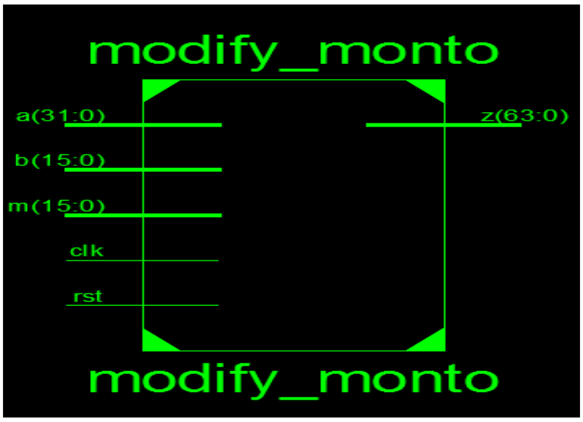


Fig. 6: RTL schematic

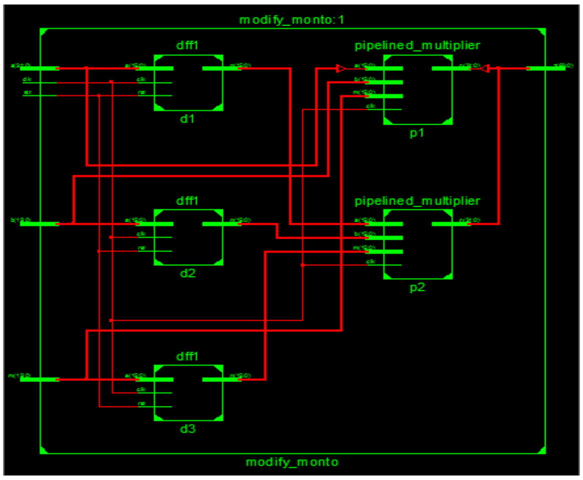
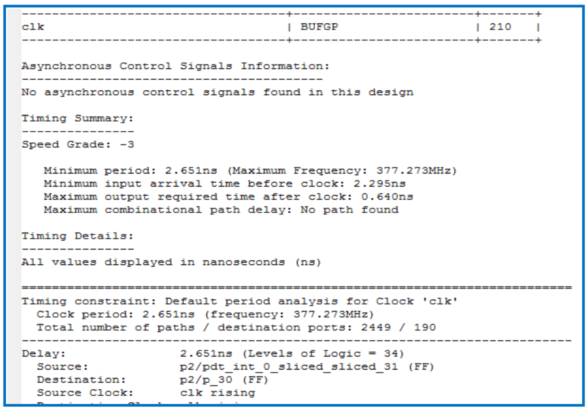


Fig. 7: Internal structure of RTL schematic



```
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 210   |
-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 2.651ns (Maximum Frequency: 377.273MHz)
   Minimum input arrival time before clock: 2.295ns
   Maximum output required time after clock: 0.640ns
   Maximum combinational path delay: No path found

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

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 2.651ns (frequency: 377.273MHz)
  Total number of paths / destination ports: 2449 / 190
-------------------------------------------------------------------------
Delay:               2.651ns (Levels of Logic = 34)
  Source:            p2/pdt_int_0_sliced_sliced_31 (FF)
  Destination:       p2/p_30 (FF)
  Source Clock:      clk rising
```

Fig. 8: Synthesis report

Internal architecture of Montgomery multiplier:

The simulation results and device utilization summary of internal architecture of modified 16-bit Montgomery multiplier as shown in **Fig.9 & Fig. 10.**

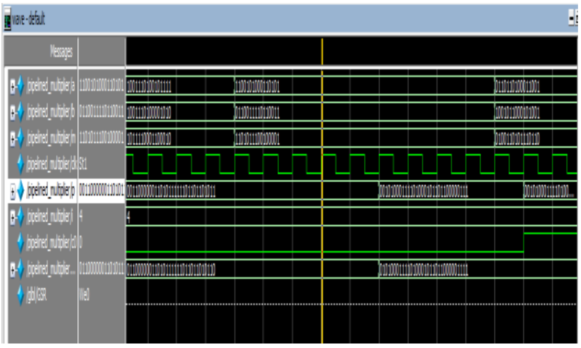
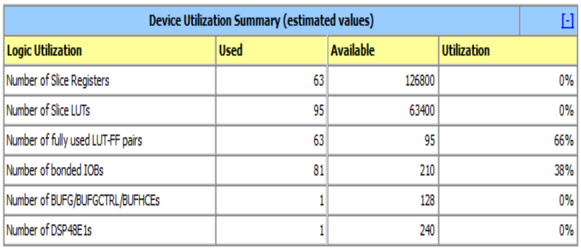


Fig. 9: Montgomery multiplier



Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	63	126800	0%
Number of Slice LUTs	95	63400	0%
Number of fully used LUT-FF pairs	63	95	66%
Number of bonded IOBs	81	210	38%
Number of BUFG/BUFGCTRL/BUFHCEs	1	128	0%
Number of DSP48E1s	1	240	0%

Fig. 10: Device Utilization Summary

The RTL schematic and synthesis report of internal structure of 16-bit modified Montgomery multiplier as shown in **Fig.11, Fig.12 & Fig.13**.

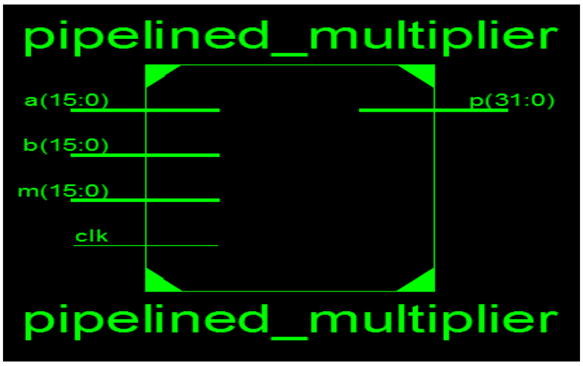


Fig. 11: RTL schematic

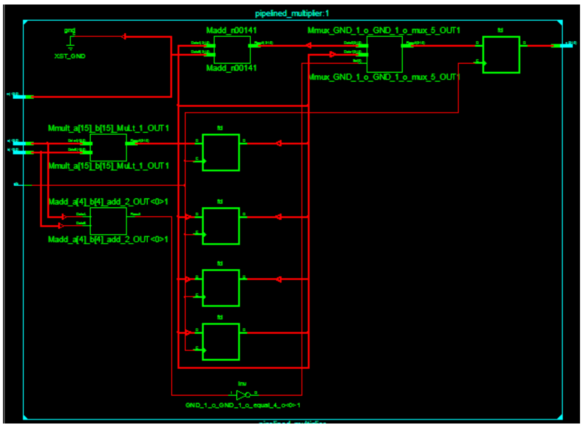
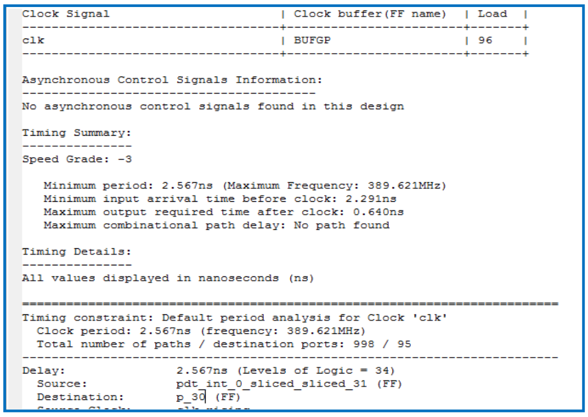

pipelined_multiplier:1
gnd
XST_GND
Madd_n00141
Madd_n00141
Mmux_GND_1_o_GND_1_o_mux_5_OUT1
Mmux_GND_1_o_GND_1_o_mux_5_OUT1
fd
Mmult_a[15]_b[15]_MuLt_1_OUT1
Mmult_a[15]_b[15]_MuLt_1_OUT1
Madd_a[4]_b[4]_add_2_OUT<0>1
Madd_a[4]_b[4]_add_2_OUT<0>1
fd
fd
fd
fd
inv
GND_1_o_GND_1_o_equal_4_o<0>1

Fig. 12: Internal structure of RTL schematic



```
Clock Signal                       | Clock buffer(FF name)  | Load  |
-----------------------------------+------------------------+-------+
clk                                | BUFGP                  | 96    |
-----------------------------------+------------------------+-------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 2.567ns (Maximum Frequency: 389.621MHz)
   Minimum input arrival time before clock: 2.291ns
   Maximum output required time after clock: 0.640ns
   Maximum combinational path delay: No path found

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

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 2.567ns (frequency: 389.621MHz)
  Total number of paths / destination ports: 998 / 95
-------------------------------------------------------------------------
Delay:               2.567ns (Levels of Logic = 34)
  Source:            pdt_int_0_sliced_sliced_31 (FF)
  Destination:       p_30 (FF)
```

Fig. 13: Synthesis Report

FIR Filter:

The simulation results and device utilization summary of FIR filter as shown in Fig. 14 & Fig. 15.

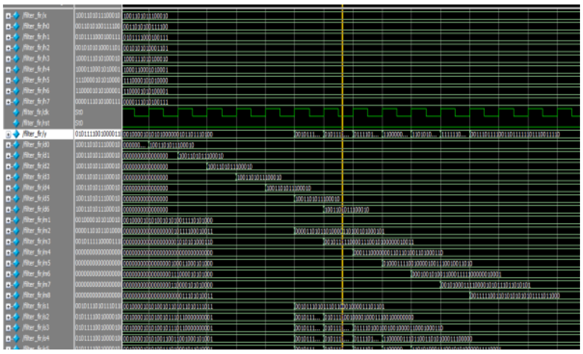
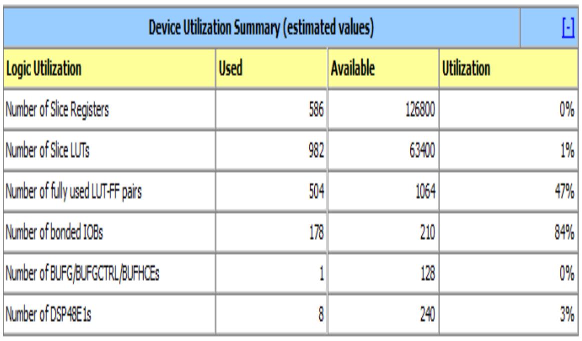


Fig. 14: Simulation results of FIR filter



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	586	126800	0%
Number of Slice LUTs	982	63400	1%
Number of fully used LUT-FF pairs	504	1064	47%
Number of bonded IOBs	178	210	84%
Number of BUFG/BUFGCTRL/BUFHCEs	1	128	0%
Number of DSP48E1s	8	240	3%

Fig. 15: Device Utilization Summary

The RTL schematic diagram and synthesis report of FIR filter as shown in Fig. 16, Fig. 17 & Fig. 18.

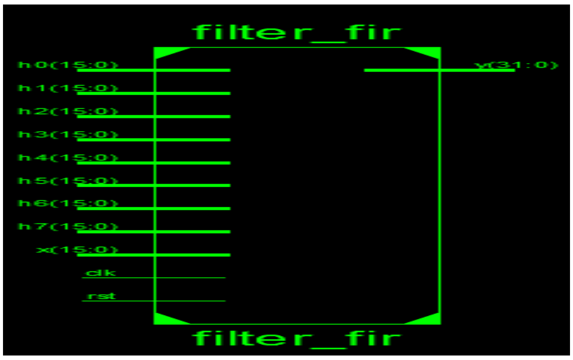


Fig. 16: RTL schematic

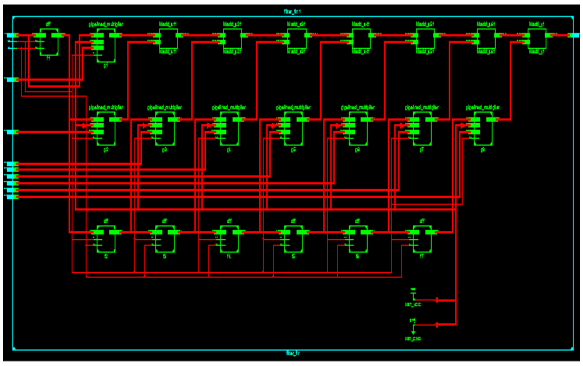
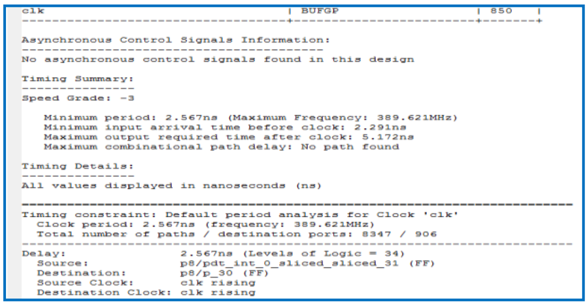


Fig. 17: Internal structure of RTL schematic



```
clk                                   | BUFGP                   | 850    |
-------------------------------------+------------------------+--------+

Asynchronous Control Signals Information:
----------------------------------------
No asynchronous control signals found in this design

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 2.567ns (Maximum Frequency: 389.621MHz)
   Minimum input arrival time before clock: 2.291ns
   Maximum output required time after clock: 5.172ns
   Maximum combinational path delay: No path found

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

=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 2.567ns (frequency: 389.621MHz)
  Total number of paths / destination ports: 8347 / 906
-------------------------------------------------------------------------
Delay:               2.567ns (Levels of Logic = 34)
  Source:            p8/pdt_int_0_sliced_sliced_31 (FF)
  Destination:       p8/p_30 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising
```

Fig. 18: Synthesis Report

COMPARISON TABLE:

Montgomery Multiplier

To compare the Montgomery multipliers of existing and proposed method as shown in Table II.

Table II: comparison table of existing and proposed Montgomery multiplier.

METHOD	Number of Slice Registers		Number of Slice LUTs		Maximum Frequency
	USED	Available	USED	Available	
Existing Method	160	126800	298	63400	225.647MHz
Proposed Method	144	126800	190	63400	377.273MHz

VII. CONCLUSION

The Montgomery multiplication algorithm is an efficient method for modular multiplication with an arbitrary modulus, particularly suitable for implementation on general-purpose computers like signal processors or microprocessors. The proposed system of Montgomery multiplier used to reduce the area and usage of time is less. To implement the pipelining into the Montgomery multiplier architecture, it gives better performance. However many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications. In this proposed multiplier is applied to the 8-tap FIR filter used for fast computation. In this process are implemented by using Xilinx Artix 7.

Future Work:

To implement the Montgomery multiplier can be applied to N number of bits and to design the N-tap FIR filter. The main application of Montgomery multiplier is cryptography like RSA and ECC algorithms etc…. Further the design is implemented in many applications.

REFERENCES

[1]. Adrien Le Masle and Wayne Luk, "Mapping Loop Structures onto Parametrized Hardware Pipelines, ieee transactions on very large scale integration (VLSI) systems Digital Object Identifier 10.1109/TVLSI. 2013.2251430.1063-8210.

[2]. E. Öztürk[a] , B. Sunar[a] , E. Savas[b,*] "A versatile Montgomery multiplier architecture with characteristic three support" Computers and Electrical Engineering xxx (2008) xxx–xxx doi:10.1016/j.compeleceng.2008.05.009

[3]. Peter L. Montgomery. Modular multiplication without trial division. Mathematics of Computation, 44:519–521, 1985.

[4]. T Ramesh Reddy, M. Tech Student, and Dr. K. Soundara Rajan, Professor "Low Power and Low Area Digital FIR Filter Using Different Multipliers and Adders" International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 3, May – 2012 ISSN: 2278-0181.

[5]. 1Gowrishankar , 2ManoranjithamD, 3Jagadeesh P "Efficient FIR Filter Design Using Modified Carry Select Adder & Wallace Tree Multiplier" ISSN: 2278 – 7798 International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 3, March 2013.

[6]. Nadia Nedjah, Luiza de Macedo Mourelle, "Software/Hardware co-design of efficient and secure cryptographic hardware" Journal of Universal Computer Science, vol. 11, no. 1 (2005), pp 66-82

[7]. K. Sakiyama, N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "Reconfigurable Modular Arithmetic Logic Unit for High-Performance Public-Key Cryptosystems", K. Bertels, J.M.P. Cardoso, and S. Vassiliadis (Eds.): ARC 2006, LNCS 3985, 2006. _c Springer-Verlag Berlin Heidelberg 2006 pp. 347–357,

[8]. Shinichi Kawamura1, Masanobu Koike2, Fumihiko Sano2, and Atsushi Shimbo1, "Cox-Rower Architecture for Fast Parallel Montgomery Multiplication", B. Preneel (Ed.): eurocrypt 2000, LNCS 1807, pp. 523-538.

[9]. 1S. Karunakaran and 2N. Kasthuri, "VLSI implementation of FIR filter using computational sharing multiplier based on high speed carry select adder" American Journal of Applied Sciences, 2012, 9 (12), 2028-2045 ISSN: 1546-9239 ©2012 Science Publication.

[10]. A Kamaraj, C. Kalyana Sundaram, J.Senthilkumar, "Pipelined FIR Filter Implementation using FPGA", International Journal of Scientific Engineering and Technology (ISSN : 2277-1581) Volume No.1, Issue No.4, pg : 55-60 01 Oct. 2012

[11]. Adrien Le Masle1, Wayne Luk1, Jared Eldredge2, and Kris Carver2, "Parametric Encryption Hardware Design", P. Sirisuk et al. (Eds.): ARC 2010, LNCS 5992, pp. 68–79.