JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An implementation of FIR filters using Digit-Serial Montgomery multipliers and Asute Compressors for Digital signal processing applications

Sub Title : An Efficient implementation of FIR filters using high speed multipliers and compressors for Signal processings applications

Mrs. Manikanta Umadevi, M.Tech Student, Department of ECE, Viswam Engineering College (JNTUA), Madanapalli, Chittor(dist), AP, India.
manikantaumadevi94@gmail.com
Mrs. T. Reddy Rani, Assistant professor, Department of ECE, Viswam Engineering College, Madanapalli, Chittor(dist),AP, India.
sai.reddyrani@gmail.com
Mr.N.Nagendra ,Assistant professor,Viswam Engineering College, Madanapalli, Chittor(dist),AP,India.
nnagendra1993@gmail.com

Abstract—In the field of VLSI, enhancement is prominent. Arithmetic circuits are one of the influential sectors in today's end products of electronics, where multipliers are one of the deciding factors of efficiency. Multiplier plays an important role in different applications such as digital signal processing in which it acts as a key hardware block. As time rolls down, the technology exposed the ways for the initiation of many hardware and software implementations of the faster multipliers. One among them is the Montgomery multiplier. The fundamental operation in the Montgomery multiplier is the modular multiplication. It is mainly used in FIR filters, which in-turn has numerous applications such as speech analysis, multi-rate signal processing, adaptive filters, and averaging filters. With the usage of proposed compressor in the conventional design of the multiplier, the number of transistor count has been declined by a significant amount and made the design into an optimal area design. This paper presents a modified Montgomery multiplier design and its implementation in the 5 th order FIR filter.

Index Terms— Cryptography, Montgomery Multiplier, Compressor, FIR Filter

I. INTRODUCTION

In the world of smart technology, data security is a crucial entity for every individual. The techniques for securing data are in high demand and became challenging for the designers to come up with efficient ways based on the requirements. The expanding areas of networks and security demand more number of efficient algorithms for use in cryptography [1]. Many researchers have done phenomenal work and proposed various architectures with the advancements [2]. Encryption is an essential criteria in cryptography for data handling and processing. When explicit information from one environment is to send to the other, then encryption of the data is necessary for security purposes. Cryptography provides pavement for this restraint.

It deals with encryption and decryption of the data such that the fundamental information transfer from the sender to the recipient remains confidential. The cryptographic algorithms which are used in advance systems has motive principles of authentication and data secrecy. Performing modular multiplication is one of the core operations in cryptography. Given two inputs P and Q of k bits each, where P is the multiplier and Q is the multiplicand and modulus N of k bits, modular multiplication is given by eq. (1), in which C represents the result of the multiplication.

P = pk−1.....p1p0 Q = qk−1.....q1q0 N = nk−1 n1n0 C = P ∗ Q ∗ mod(N)(1)

In most of the cryptographic applications, modular multiplication algorithm tends to be a time-consuming process. Therefore, for the efficient functioning of the algorithm and to decrease time consumption, modular multiplication reduction is necessary [3].

As the Montgomery multiplication speedup modular multiplication algorithm, it is one of the adaptable choices for the designers as it evades trial division, which is the strenuous algorithm for integer

factorization. Embedding the Carry-Save Adders (CSA) reduces the critical path delay [4]. Performing addition and division arithmetic operations by the powers of two substitutes the trail division in Montgomery multiplication. Computing multiplications take a transformation from time-consuming to time-saving process with the Montgomery multiplication as there are no division and subtraction operations involved.

Instead, the addition of multiples of N takes place for the reduction of least significant bits and discard them in the final product. The represenatation of multiplication in Montgomery domain is as eq. (2) $C = P * Q * r^{-1} \bmod(N)$ (2) where, r is a positive integer and co-prime of N. The condition of r>N must be satisified all the time while performing the multiplication. In general, r is considered as 2 m, where m is a positive integer. This is because shift and bit operations are convenient to consider for the efficient implementation. N is always an odd prime number. The implementation of modular multiplication consists of 3 steps. 1) The first step is the partial product generation for the given inputs P and Q. 2) The second step in the process is the division. 3) The final step is the reduction of the least significant bits to produce the exact results. There are several ways to implement the Montgomery multiplication algorithm, and many algorithms are proposed by different designers to scale down the complexity and to improve efficiency. The two ways for the implementation of Montogmery multiplier are bit-serial [5] and bit-parallel. The selection depends on the type of application.

BASICS OF DIGIT-SERIAL MONTGOMERY MULTIPLIER

The conventional Montgomery multiplier [6] is constructed for m=6 and d=3, where m is the digit size of each input and d is the bit size. The first stage in computing the results of the multiplier, is the processing stage. The partial product generation for the given multiplicand and multiplier takes place in this stage with the combination of And-network and Adder-network. The multiplicand and the multiplier are of 6-bit length each. The individual bit representation of multiplicand, multiplier and modulus N is given by eq. (3). The complete Montgomery multiplication algorithm takes place in 2 cycles. In the first cycle of the algorithm, 6 bits of multiplicand P and first 3 bits of multiplier Q are given as the inputs. In the second cycle, the next 3 bits of the multiplier Q will be acting as the inputs. To choose the bits of multiplier according to the cycle, 2:1 multiplexers are used. For each bit of the Q, a multiplexer is used and based on the selection line, the required bits are chosen. If the selection line is high for all the 3 multiplexers, first 3 bits of Q i.e, (q0,q1,q2) will be propagated as inputs. If the selection line is low, the next 3 bits of Q i.e, (q3,q4,q5) are propagated as inputs. In the first cycle of the algorithm, all the carry bits which are given as the inputs to the processing element are considered as zeros i.e, C=0. 6 bits of multiplicand and first 3 bits of the multiplier generates partial products as shown in eq. (4). The outputs of the processing element are the Sum and Carry bits of full adders that are present in the adder network, and they will be driving the next stage. P = (pm−1, pm−2 p0) Q = ((qm−1, qm−2 qm−d),(qd−1, qd−2.....q0)) N = (nm−1, nm−2 n0) C = 0 (3) $C = X_{d\,i=0}^{m-1}{}_{j=0} p_j q_i + C$ (4) The second stage is the division stage. An array of full adders performs the division operation in the algorithm. Bits of N operates as one of the inputs to the full adders and the other inputs are Sum and Carry bits of the adders involved in the successive stages. The outputs of adders after performing the division operation are the Carry bits from C0 to C6 and Cc1 to Cc6. The Carry bits will be acting as inputs to the final stage of the design. 3 D-flipflops in which 2 are of 6 bit input size and 1 single D-flipflop are used to store Carry bits generated from the division cell and to give them back as inputs to the processing element in the second stage. The usage of D- flipflops in the circuit reduces the critical path delay. A D-flipflop works under the presence of a clock signal and operates with a delay in input by one clock cycle. After the generation of Carry bits, the second cycle starts and in the second cycle of the multiplier, the next 3 bits of multiplicand and the Carry bits from the D-flipflop will repeat the process and produces the final output of the multiplication from the compressor. The output produced is of 6 bit length which is equal to the bit size of the given inputs to the multiplier.

II. MODIFIED MONTGOMERY MULTIPLIER

The modified Montgomery multiplier consists of 3 stages as shown in Fig. (1). The first two stages perform the similar operation as in the conventional design [6]. The reduction cell in the conventional design is replaced with the proposed 12:6 Astute Compressor, making the design into a modified one. This modification resulted in area and power efficient multiplier by reducing the complexity involved in the usage and design of multiplexers in the reduction cell of the conventional design and by declining the number of transistor count.

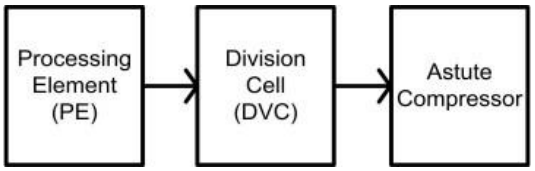

Fig. 1: Block Diagram of Modified Montgomery Multiplier

The modified Montgomery multiplier design works for any 6 bit length input P and Q, but the value of N is restricted to 3 and 7. The results produced by the multiplier are accurate. the design of modified Montgomery multiplier with a combination of CMOS and PTL logic [7] resulted in low power and area efficient design. The proposed Astute

compressor lessened the number transistors to a notable amount.

III. PROPOSED 12:6 COMPRESSOR: ASUTE COMPRESSOR

The compressor is a circuit that is used to depreciate the area by down turning the transistors count and reduce the overall power and delay for more reliable performance of a design [8]. This paper presents a 12:6 Astute Compressor which compresses 12-bit input to 6-bit output as shown in Fig. (3). The compressor conisists of a Half Adder (HA) and a series of Modified Full Adders (MFA).

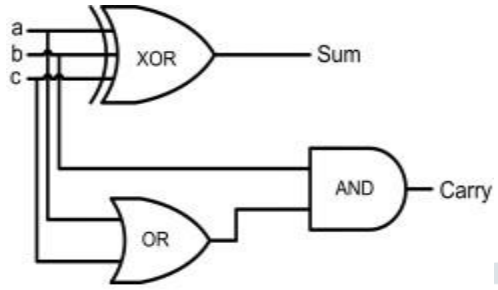

Fig. 2: Modified Full Adder (MFA)

The modified full adder works for all the input combinations except for the input combination of 101 which is binary equivalent of 5. All the adders are connected in series and the Carry of each adder will be propagated to the next successive adder as one of the inputs. The other inputs to the adders are the previous stage Carry bits C0 to C6 and Cc1 to Cc5. Propagation of the Carry bits to the next successive adder takes place till the end. The final adder in the compressor consists of only Sum part and the Carry part is neglected for the exact output of 6 bit length. The output of the compressor is the final output of the multiplier after completion of 2 nd cycle. The Sum bits of each adder will constitute the output bits. The first adder Sum bit is the Least Significant Bit (LSB) of the final result, and the last adder Sum bit is the Most Significant Bit (MSB). The design of the compressor has been done with a combination of PTL and CMOS logic style with 78 transistors, which is equal to 42% reduction in the transistor count by CMOS logic style. The compressor produces an average power of 0.449 uW and a delay of 4.455 ns. The comparison of the proposed compressor with the other compressors on FPGA [9] in terms of power, delay, and number of logic gates used is shown in Table I. The graphical representation of power and delay comparison of the proposed Astute compressor with the others compressors are shown in Fig. (4), Fig. (5) respectively.

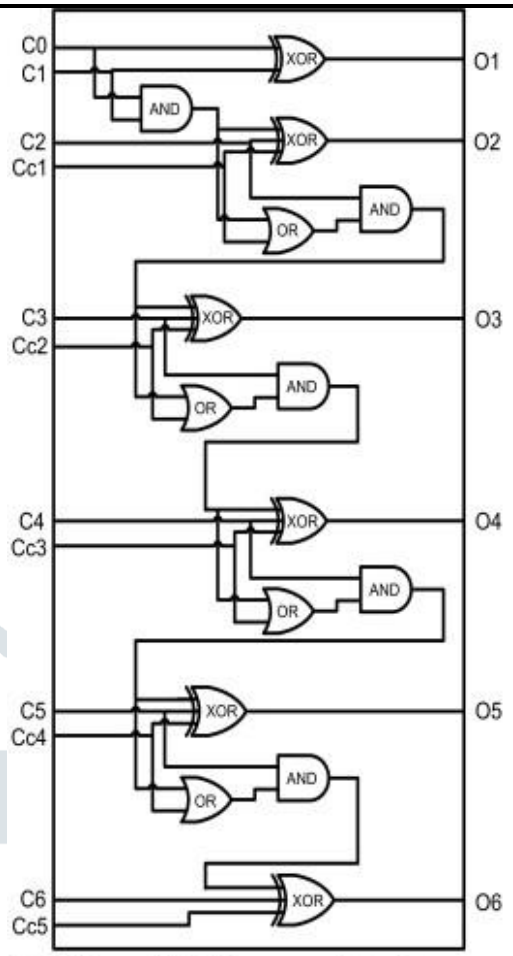

Fig. 3: Proposed 12:6 Compressor:Astute Compressor

IV. PROPOSED MONTGOMERY MULTIPLIER DESIGN USING ASTUTE COMPRESSOR

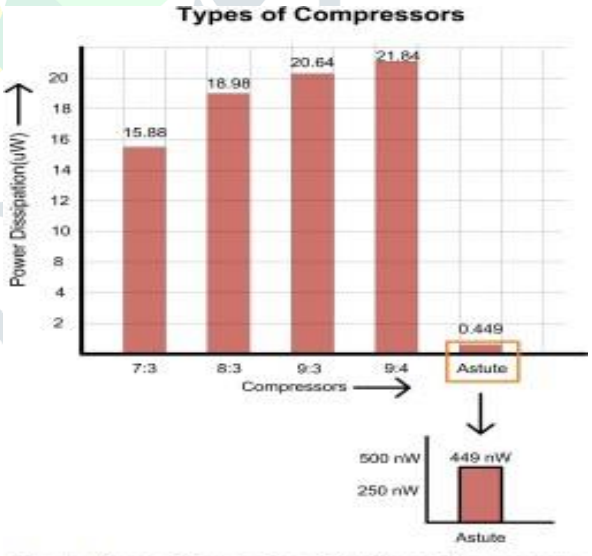

Fig. 4: Power Comparison of Various Compressors

The proposed multiplier is verified with the other multipliers designed using different techniques to demonstrate its effectiveness.

Analysis of Proposed Multiplier in terms of Power Dissipation and transistor count: The proposed multiplier has been compared with the various multipliers, designed using different techniques in terms of power and number of transistor count as shown in table II. The usage of pass transistor logic in combination with conventional CMOS logic resulted in yielding

low power and area efficient design.

V. APPLICATION OF THE PROPOSED MULTIPLIER

In the epoch of propelling technology, processing of the signal to establish proper communication plays a pivotal role [14]. To have better communication, one must take many factors into account. Usage of filters in digital signal processing can reduce the complexity and helps to achieve better performance in various factors such as noise cancellation, removal of unwanted signals, and proper attenuation. Restoration and separation of the signals while signal processing are the two important undertakings by a filter. The classification of filters depends on whether it is an analog or digital filter. The digital filters can be classified into Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. The analog filters can be classified into Butterworth filter, Chebyshev filter etc. A filter can be extended to any number of stages based on the requirement. The complexity depends on number of stages present. More number of stages results in high complexity filter design. With increase in complexity, the cost of the filters will also increase. Therefore, the design of higher order filters with reduced complexity and cost, is on high demand in the market. A 5 th order low pass FIR filter is adopted for the implementation of the proposed multiplier in this paper. FIR filters are mainly used in the applications where there is a need for linear phase response. In general, FIR filters are stable filters. To design a FIR filter, one must choose the design method. Three methods are known for the design of FIR filter namely, 1) Frequency sampling techniques 2) window methods 3) optimal filter design methods

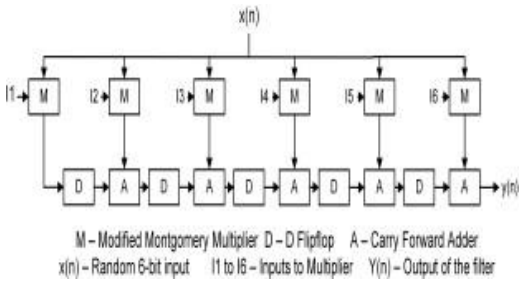

Fig. 7: 5th Order Low Pass FIR

Filter $y(n) = I1.x(n)+I2.x(n-1)$ ..... $+I6.x(n-5)$ (5)

The general equation of the filter is given by eq. (5). The filter design consists of multipliers, D-flipflops, and a carry-forward adder [15] as the main elements as shown in Fig. (7). D-flipflop is used in order to reduce the critical path delay of the filter. The carry forward adder which is used in the filter consists of a half adder and full adders as its functional elements in which carry of each adder is propagated to the next adder in series. The inputs to the adder are the multiplier output and the D-flipflop output each of 6-bit length each. The adder produces a 6-bit output, and the final output of the circuit is the output of last adder present in the filter.

VI. Simulation Results

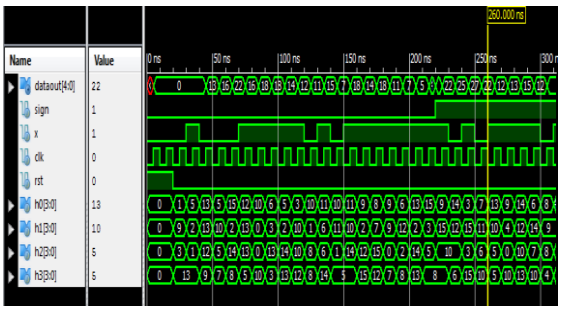

Figure 8: Simulation outcome

Figure 8 represents the simulation outcome of 5 stage fir filter. Here, Ho,h1,h2,h3 and h4 are the impulse inputs, x is the data input and resultant outcome is stored into data out. With sign control

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	12	408000	0%
Number of Slice LUTs	66	204000	0%
Number of fully used LUT-FF pairs	9	69	13%
Number of bonded IOBs	25	600	4%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 9: Design summary

Figure 9 represents area utilization by the extension method, out of available 408000 slice registers only 12 are used. Out of available 20400 slice look up tables only 66 are used, so the area consumption is reduced as compared to proposed method.

```
                                   Gate     Net
Cell:in->out      fanout   Delay    Delay  Logical Name (Net Name)
----------------------------------------  ------------
  IBUF:I->O            7   0.000   0.578  h3_2_IBUF (h3_2_IBUF)
  LUT6:I0->O           5   0.043   0.569  u2/Madd_n0072_xor<2>11 (u2/n0072<2>)
  LUT6:I0->O           1   0.043   0.289  u2/Madd_BUS_0013_GND_4_o_add_15_OUT_cy<2>11 (u
  LUT5:I4->O           1   0.043   0.550  u2/Madd_BUS_0013_GND_4_o_add_15_OUT_xor<3>11 (
  LUT6:I0->O           1   0.043   0.428  u2/Mmux_d[3]_q[4]_wide_mux_19_OUT_83 (u2/Mmux_
  LUT6:I3->O           1   0.043   0.000  u2/d<3>31 (u2/d[3]_q[4]_wide_mux_19_OUT<3>)
  FDRE:D              -0.001          u2/q_3
----------------------------------------
Total                      2.630ns (0.215ns logic, 2.415ns route)
                                   (8.2% logic, 91.8% route)
```

Figure 10: Time summary

Figure 10 represents time utilization by the extension method, and the proposed method consumes total 2.630ns of path delays, which includes 0.215 ns logical delays and 2.415ns of route delays, and these are reduced as compared to proposed method respectively.

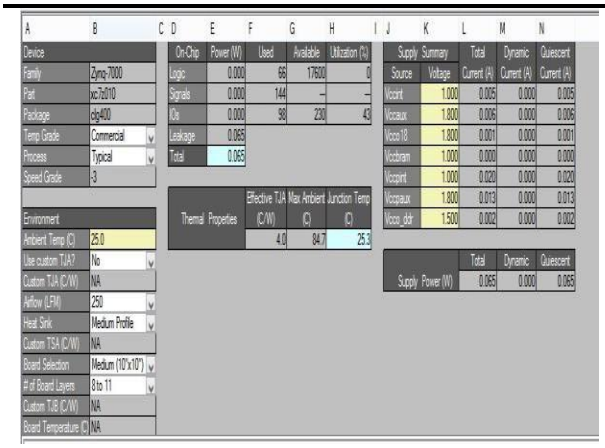

Figure 11: power summary

Figure 11 represents power utilization by the extension method, and the proposed method consumes total 0.065 watts of power, respectively.

VII. CONCLUSION

In today's smart products of electronics and their applications in different fields, the speed and area constraints are met by the different multipliers proposed by several designers. The upgrading technology still demands the efficient design of the multipliers in different constraints. A modified Montgomery multiplier, which plays a crucial role in cryptography, has been proposed in this paper with low complexity and enhancements in power and area factors. The design of the multiplier by embedding the proposed Astute compressor made it into a modified one and an area efficient design. The combination of CMOS and pass transistor logic for designing the logic gates involved in the multiplier declined the number of transistors and yielded in low power design. A 5th order low pass FIR filter is designed with the proposed multiplier and the results are very encouraging.

REFERENCES

[1] Tohru Hisakado, Nobuyuki Kobayashi, Takeshi Ikenga, et.al, "N bit-wise Modular Architecture for Public Key Cryptography", Annual 2004 International Carnahan Conference on Security Technology, 2004.

[2] Kakde, S., Somulu, G., et.al, "Performance analysis of Montgomery multiplier for public key cryptosystem", Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013.

[3] Vollala, S., Varadhan, V. V., Geetha, K., et.al, "Efficient modular multiplication algorithms for public key cryptography", IEEE International Advance Computing Conference (IACC), 2014.

[4] Sassaw, G., Jimenez, C. J, et.al, "High radix implementation of Montgomery multipliers with CSA", International Conference on Microelectronics, 2010.

[5] Bertoni, G., Breveglieri, et.al, "Efficient finite field digit-serial multiplier architecture for cryptography applications", Proceedings Design, Automation and Test in Europe. Conference and Exhibition 2001, 1530-1591.

[6] Sahar Fatemi, Maryam Zare, Amir Farzad Khavari, et.al, "Efficient implementation of digit-serial Montgomery modular multiplier architecture", IET Circuits Devices Syst., 2019, Vol. 13 Iss. 7, pp. 942-949.

[7] Kamenskih, "The Research into Fault-Tolerant Design Usign Pass Transistor Logic", IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2019.

[8] R. Marimuthu, Y. Elsie Rezinold, et.al, "Design and Analysis of Multiplier Using Approximate 15-4 Compressor", IEEE Access, 5, 1027-1036.

[9] Yuhao Leong, HaiHiung Lo, Micheal Drieberg, et.al, "Performance Comparison Review of 8-3 Compressor on FPGA", Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.

[10] Harasimranjit Kaur and N. R. Prakash, et.al, "Area-Efficient low PDP 8-bit Vedic Multiplier Design using Compressors", IEEE Trans RAECS UIET, Dec. 2015.

[11] Praveen, et.al, "Low Power Approximate Multipliers With Truncated Carry Propogation for LSBs", IEEE Trans. on Comp., 2018.

[12] Pravan kumar and A. Radhika, et.al, "FPGA Implementation of High Speed 8-bit Vedic Multiplier using barrel shifter", IEEE Trans., 2013.

[13] Shashank S. Meti, et.al, "Design and Implementation of 8-bit Vedic Multiplier using mGDI Technique", IEEE Trans., 2017.

[14] Zhixi Yang, Jun Yang, et.al, "Approximate compressor based multiplier design methodology for error-resilient digital signal processing", IEEE International Conference on Signal and Image Processing (ICSIP), 2016. Manish B. Trimale, et.al, " FIR Filter Implementation on FPGA Using MCM Design Technique", Proceeding of Second International conference on Circuits, Controls and Communications, 2017, pp. 213-217.