International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.8, pp : 1102-1105 1 Aug 2014

VLSI Implementation of Montgomery Multiplier in Finite Field Arithmetic for ECC over GF(2^163)

Shemina A Karim, Reneesh C.Zacharia, Rijo Sebastian
Department of Electronics and Communication Engineering, Mangalam College of Engineering, Kerala
Email: sheminakarim@yahoo.com

Abstract—An efficient architecture for Finite Field Arithmetic with Montgomery multiplier is presented. Efficient implementation of Montgomery multiplier in the finite field arithmetic yields less area, power and delay. The advantage of ECC (Elliptic Curve Cryptography), it is more secure for wireless communication. Implemented with Xilinx ISE 13.2 and simulated with Modelsim.

Keywords—Finite field Arithmetic, Montgomery Multiplier Elliptic Curve Cryptography, FPGA.

I. Introduction

ECC (Elliptic Curve Cryptography)which is a more secure public key cryptography, ECC is having the smaller key size than RSA. For the rapid growth of electronics in nowadays it needs more secure data transmission. Confidential data exchange over public computer network needs authentication[i]. Digital signatures and secure channels are meant for valid data exchange and cryptography gives a solution for this. Modular multiplication is for polynomial multiplication in finite field arithmetic. This paper introduces the FPGA implementation of Montgomery modular multiplication over GF(2^163)[v].

Montgomery multiplier which is used in finite field arithmetic meant for ECC Processor The modular multiplication which gives high speed of operation. Finite field is also known as Galois field, which involves addition, multiplication squaring and division. The basic operation of an ECC is the point multiplication and is realised with finite field arithmetic. The Montgomery modular multiplier and squarer circuit is used to realize the finite field arithmetic, it shows more efficient for high speed applications in the ECC implementation. Multiplication is the more complex and time consuming process in finite field arithmetic ,with most important one. Montgomery multiplier replaced with interleaved multiplier for the conventional ECC[i] gives less power, area and delay. The field arithmetic for polynomial basis can select the corresponding irreducible polynomial. The basic conversion is simple if the adequate selection of polynomial . By the proper polynomial, it reduces the complexity of modular multiplication. Experimental results are carried out with Xilinx ISE and Modelsim

II. Finite Field Arithmetic (FFA)Architecture.

We can propose the finite field arithmetic as a specialised ALU which computes the addition, multiplication, squaring and inversion. The ALU architecture is shown in Fig.1.The conventional ECC processor having interleaved multiplication circuit and can be replace with Montgomery multiplication, which acquires more speed than the conventional one.

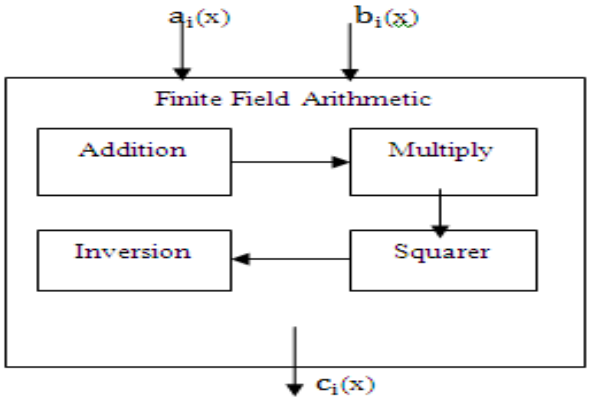

Fig.1 Architecture of FFA

Finite Field Arithmetic consists of :-
a) Addition
b) Multiplication
c) Squarer
d) Inversion/Division
Where a(x) and b(x) are the inputs and the modular output for the finite field is obtained at the C(x).Here the Galois field is defined over (2^163)[iv] and is computed for the adequate multiplier selection. The interleaved multiplier for the Conventional ECC is replaced with more efficient Montgomery multiplier. For cryptography the implementation of fast multiplier is a must, for high speed applications like wireless and mobile communication[ii].

a) Addition

It is defined as a simple modular xor operation
C(x) = A(x) xor B(x) mod f(x)
Where f(x) is the defined polynomial over GF(2^163)
f(x)= $x^{163} + x^7 + x^6 + x^3 + 1$.

b) Multiplication

For a conventional ECC Processor using an Interleaved multiplier and is replace with fast multiplier Montgomery multiplier, which reduces the power, delay and area.[ii]

i) Interleaved Multiplier

The shift and add method having interleaved reduction step is involved with a simple algorithm .multiplication of a(x)and b(x)in G(2^163)
can be computed as:
C(x)=a(x)b(x)mod f(x).
Algorithm-1

```
for i in 0 . . m-1 loop c(i) :=0;end loop;
```

```
for  i  in 0 . .  m-1 loop
c := m2xvv(m2abv(b(i),a),c);
a := Product _ alpha _ A(a,f);
end loop;
```

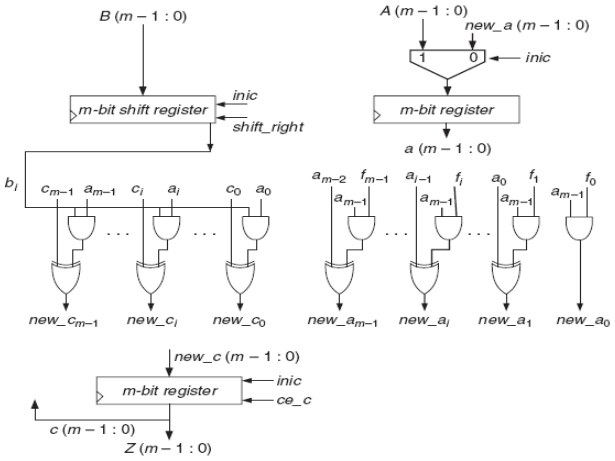

Fig 2.Interleaved multiplier data path

ii) Montgomery Multiplier

Montgomery multiplier extended in finite field arithmetic for modular multiplication. It calculated as follows:[iii]

C(x) = a(x)b(x) r^{-1}(x) mod f(x)

where r(x) is a fixed element and gcd(r(x),f(x))=1.

Algorithm 2

Input :a(x),b(x),f(x)

Output :c(x) = a(x)b(x) x^{-m} mod f(x)

1. c(x) := 0
2. for i = 0 to m-1 do
3. c(x) := c(x) +aib(x)
4. c(x) : = c(x) +c0f(x)
5. c(x) := c(x)/x

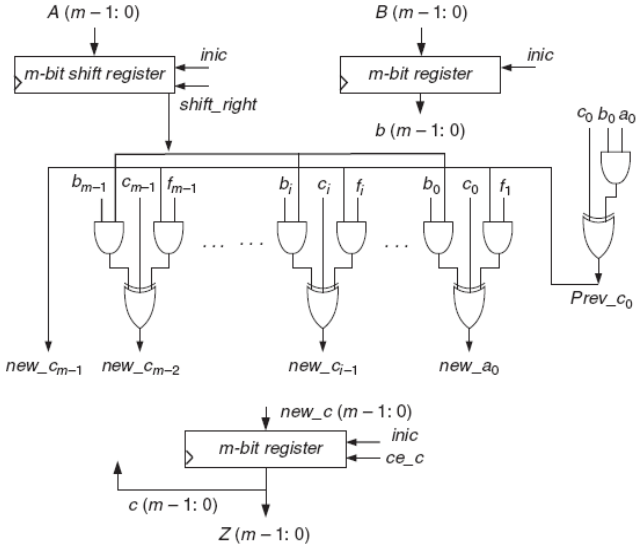

Fig 3. Montgomery multiplier datapath

c) Montgomery Squarer

Squaring can be done by modifying multiplication algorithm.

Algorithm 3

```
for  i in 0 . . m-1 loop c(i) :=0;
end loop;
for  i in 0 . . m-1 loop
c :=m2xvv(c,m2abv(a(i),a));
if c(0) =1 then
 c :=m2xvv(c,m2abv(c(0),f));
c :=lshift(c);
c(m-1) :=1;
else;
c := lshift(c);
end if;
end loop;
```

d) Inversion/Division

Algorithm 4

```
for i in 0 . .  m-1  loop
s(i) :=  f (i);  r(i) :=  a(i);  v(i) := 0;
u(i)  :=0;  auxm (i)  := 0;
end loop;
u(0) := 1; d:=0;
for I in 1 . . 2*m loop
if r(m)=0 then
r  := rshiftm(r);
u  := rshiftm(u);
d  := d +1;
else
if s(m) =1 then
s := m2xvvm (s, r);
y :=m2xvvm(y, u);
end if;
s  := rshiftm (s);
if d:=0 then
auxm  :=  s;  s  :=  r; r  := auxm;
auxm  :=  v;  v  := u;
u := rshiftm(auxm);
d :=1;
else u  :=  lshiftm(u); d:= d-1;end if;
end if;
end loop;
```

III. Results and Tables

A. Simulation Results

The simulation analysis of Interleaved multiplier obtained in Modelsim is shown in fig.4.

International Journal of Scientific Engineering and Technology
Volume No.3 Issue No.8, pp : 1102-1105
(ISSN : 2277-1581)
1 Aug 2014

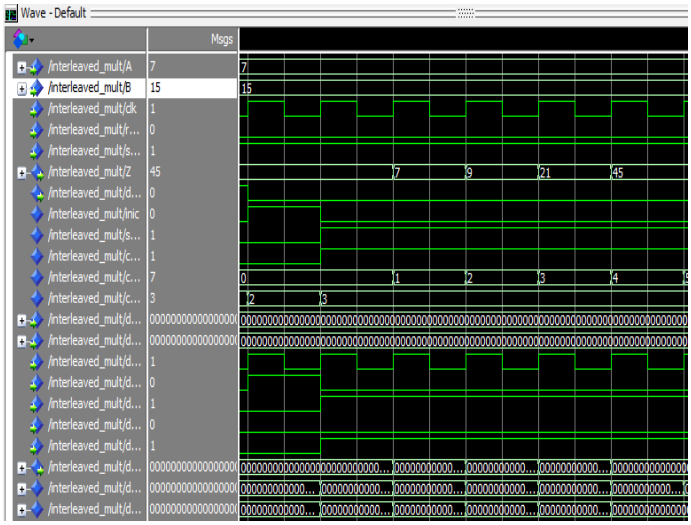

Fig 4.Generated waveform of interleaved multiplier

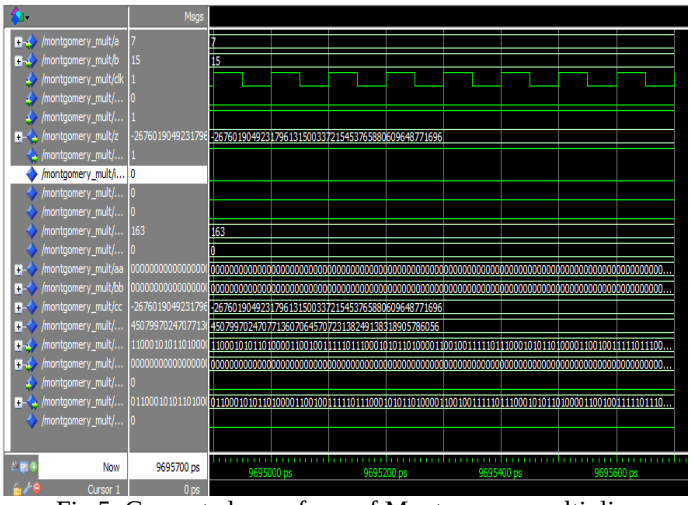

Fig 5. Generated waveform of Montgomery multiplier

B. Synthesis Results

Synthesisation is done in Xilinx ISE 13.2,and results analyzed. The parameters enhanced with power, area ,delay are given in the synthesis report.

```
                                   Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 FDC:C->Q             18   0.626   1.499  current_state_FSM_FFd2 (current_state_FSM_FFd2)
 LUT3:I0->O            1   0.479   0.681  current_state_FSM_Out31 (done_OBUF)
 OBUF:I->O                 4.909          done_OBUF (done)
----------------------------------------
Total                      8.194ns (6.014ns logic, 2.180ns route)
                           (73.4% logic, 26.6% route)
```

Fig 7: Delay obtained in Xilinx for interleaved multiplier

```
Delay:               6.818ns (Levels of Logic = 1)
  Source:            current_state_FSM_FFd2 (FF)
  Destination:       bb_128 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising

  Data Path: current_state_FSM_FFd2 to bb_128
                                   Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDC:C->Q             186   0.720   3.074  current_state_FSM_FFd2 (current_state_FSM_FFd2)
     LUT2:I0->O            35   0.551   1.870  current_state_FSM_Out11 (shift_r)
     FDCE:CE                    0.602          cc_128
    ----------------------------------------
    Total                      6.818ns (1.873ns logic, 4.945ns route)
                               (27.5% logic, 72.5% route)
```

Fig 8: Delay obtained for Montgomery multiplier

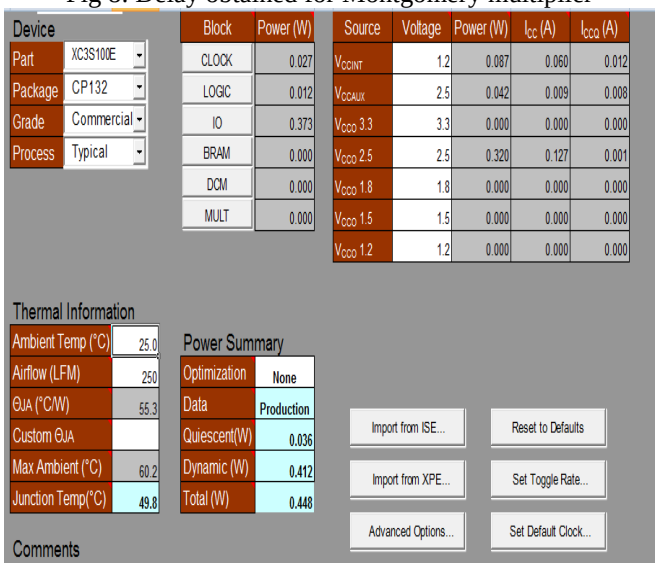

Fig 6. Power of Interleaved multiplier

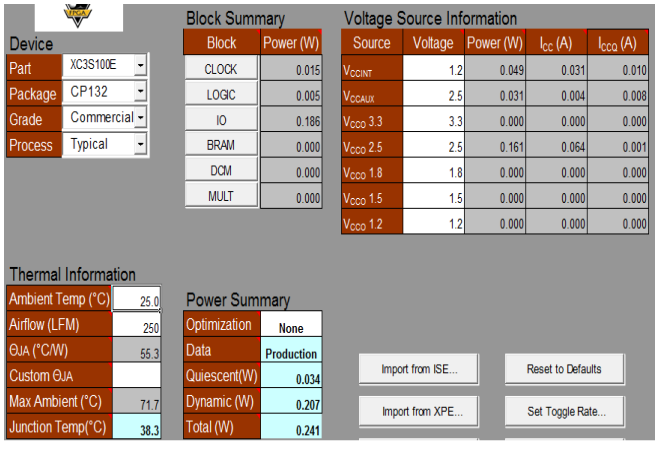

Fig 7. Power of Montgomery multiplier

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	293	1920	15%
Number of Slice Flip Flops	507	3840	13%
Number of 4 input LUTs	512	3840	13%
Number of bonded IOBs	493	173	284%
Number of GCLKs	1	8	12%

Fig 8.Device Utilization Summary of Interleaved multiplier

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	288	1920	15%
Number of Slice Flip Flops	499	3840	12%
Number of 4 input LUTs	353	3840	9%
Number of bonded IOBs	493	173	284%
Number of GCLKs	1	8	12%

Fig 9.Device Utilization Summary of Montgomery multiplier

C. Comparison

From the results, Table 1 shows Mongomery multiplier having less power, area,and delay compared to interleaved multiplier.

Table1.
Comparison of Interleaved multiplier and Montgomery multiplier

Parameter	Delay (ns)	Power (W)	No.of Slices
Interleaved	8.194	0.448	293
Montgomery	6.818	0.241	288

IV. Conclusion

Proposed Montgomery multiplier is implemented and compared the different parameters with the conventional interleaved multiplier.

Acknowledgement

I use this opportunity to express my gratefulness to all of them who helped to completing this research works.

References

i. *"Modular multiplication without trial division,".P.L.Montgomery ,in.Math.Comp..1985,vol.44.pp 519.521*

ii. *P.Schaumont,"A parallel implementation of Montgomery multiplication on multicore systems:Algorithm,analysis and prototype"IEEE Trans.Comput.,vol.60,no.12,pp.1692-1703,Dec 2011.*

iii. *C.K.Koc,T.Acar, and B.s.Kaliski,Analyzing and compairing Montgomery multiplication algorithms"IEEE Micre,vol.16,no.3,pp.26-33,Jun. 1996.*

iv. *N.Gura,S.C.Shantz,H.Eberle, S. Gupta, V.Gupta,D.Finchelstein, E.Goupy, and D.Stebila,"An end-to-end systems approach to elliptic curve cryptography,"in Proc.Cryptograph.Hardw.Embedded Syst.,Redwood Shores,CA,2002,pp.349-365.*

v. *F. Rodriguez-Henriquez, N.A.Saqib, and A.Diaz-Perez,"A fast parallel implementation of elliptic curve point multiplication over GF(2^M),Micreprocess Microsyst.,vol.28,nos.5-6, pp.329-339,2004.*

vi. *K.Jarvinen,"Optimized FPGA-based elliptic curve cryptography processor for high-speed applications,"v Integer., VLSI J.,vol.44,no.4,pp.270-279,Sep.2011.*

vii. *M.Rosing,"Implementing elliptic curve cryptography"N Manning,1999.*

viii. *N.Koblitz, "Elliptic curve cryptosystems",Mathematics of Computation 48,pp.203-209,1987. .*

ix. *D.Hankerson, A.Menezes, S. Vanstone, Guide to elliptic curve cryptography,Springer,2004.*

x. *W.Stallings,Cryptography and Network Security,4th Ed.,Prentice-Hall,2006.*