# Xilinx and Modelsim Habitat for Design of ECC Co-Processor Architecture

B.MuthuKumar[1], S.Jeevananthan[2]

[1] Research Scholar, Sathyabama University, Chennai, India. anbmuthusba@yahoo.co.in
[2]Assistant Professor, Department of Electrical and Electronics Engineering,
Pondicherry Engineering College, Pondicherry, India. jeeva_officials@rediffmail.com

*Abstract*— Xilinx is a most significant synthesizing tool for controller design in various engineering field specifically industrial engineering, instrumentation engineering, robotics, telecommunication, computer science engineering etc. To verify the working functionality of devices in these engineering fields, Modelsim simulator is preferred, where the other synthesizing tools are Lattice, Altera, Actel etc. ALDEC simulator and ISE simulators are competitors of Modelsim. Uncomplicated implementation of the device, dynamic power calculation, trouble-free verification of functionality and energetic area calculation can be achieved charmingly in Xilinx and Modelsim compared to their competitors. Architecture of any very large scale integration (VLSI) design, the Xilinx and Modelsim play an imperative role to achieve the hardware efficiency of the design and functionality verification of the design. This paper establishes how these tools create a habitat for designing architecture, by explaining the typical steps involved in elliptic curve cryptography (ECC) processor using Modelsim 5.7 and Xilinx 9.2i. The performance measure options viz. the hardware efficiency, functionality efficiency, area and power calculations are also explored.

***Index Terms—*** Xilinx, Modelsim, Elliptic curve cryptography, Simulator, Synthesize, ECC processor design.

## I. INTRODUCTION

Xilinx was launched in 1984 by two semiconductor engineers, Ross Freeman and Bernard Vonderschmitt [1][2], who were both working in manufacturing of integrated circuit and solid-state device. The initial idea of Freeman was to create chips that acted like a blank tape, allowing users to program the technology themselves. At the time, the concept was paradigm-changing. This institutive notion has led to produce massive volumes of generic circuits and hence enjoyment of strong profits by big semiconductor manufacturers. Xilinx designs and develops the programmable logic products including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support [3]. Designing and manufacturing dozens of different circuits for low price and required greater manufacturing complexity. Presently, the Field Programmable Gate Arrays (FPGA) allows the circuits to be produced in quantity to be tailored by individual market segments [4]. Modelsim 5.7 introduced by mentor Graphics Corporation [5] in the year of 2003. Modelsim is used to perform the simulation operations for digital designs and became a common tool used by both the beginners and experts. Simulation is a process of verifying the functionality of digital design after having completed the design block.

Christian Beckhoff el al [6] has described procedure to designing the digital application using the powerful Xilinx Design Language (XDL), with plenty of practical examples and use cases. Jacques L Athow et al [7] has implemented Large-Integer hardware multiplier using the vendor synthesis/place and route software tool and the design solutions are multiplier circuits based on embedded arithmetic blocks built in the Xilinx Virtex-4(V4) family of FPGA, which reduce the delay. Nor Fadzilah Mokhtar et al [8] has developed teaching aid to optimize learning and teaching process in designing simple application of digital system using VHDL(text entry) on programming PLDs such as CPLD, FPGA and etc. The interactive and animated teaching aid has been developed to cater to the constraints such as space, time and software facility. Fabrizio Ferrandi et al [9] have described a methodology that allows an easy implementation of IP-Cores focusing only on their functionalities rather than their interfaces and their integration. It has been implemented in classical Xilinx design flow using EDK and ISE. The hardware based Face detection method with Reversible Component Transformation (RCT) colour space algorithm and Xilinx Virtex-II has been implemented by Melanie Po-Leen Ooi [10].

Vıctor Montano and Manuel Jimenez [11] has presented the design of a scalable, floating-point (FP) Fast Fourier using the FPGA, and this bottom-up methodology uses a radix-2 Pease formulation scalable in the number of points, operand precision, number of butterflies, and transform direction. Bahram Rashidi et al [12] has presented the methods to reduce dynamic power consumption of a digital Finite Impulse Response (FIR) filter these methods include low power serial multiplier and serial adder, combinational booth multiplier, shift/add multipliers, folding transformation in linear phase architecture and applied to fir filters to power

consumption reduced and the FIR filters were synthesized implemented using Xilinx ISE Virtex IV FPGA and power is analized using Xilinx XPower analyzer.

Saeid Taherkhani et al [13] have proposed pipelined and non-pipelined implementation of one of the most commonly used symmetric encryption algorithm, Data Encryption Standard (DES), the Very High Speed Integrated Circuit Hardware Description Language (VHDL) is used to program the design and Modelsim is used to simulation. Bin Zhou and David Hwang[14] has presented optimized implementations of two different pipeline for Fast Fourier Transform (FFT)FFT processors , the radix-4 Single-Path Delay-Commutator (R4SDC) and radix-22 Single Path Delay Feedback (R22SDF) architecture provide the highest computational efficiency and it is designed using the Xilinx Spartan-3 and Virtex-E FPGAs and simulated using the Modelsim. J.-Y. Lai et al[15] has designed a parallel and scalable high-throughput dual-field elliptic curve cryptography processor that features all ECC functions with the programmable field and curve parameters over both the prime and binary fields. J.-Y. Lai et al[16] has presented a word-serial finite field arithmetic unit (AU) with the optimized operation scheduling and bit-parallel modular reduction for Elliptic Curve Cryptography (ECC) over binary field, based on the Montgomery scalar multiplication algorithm

This paper explores the features of trendy synthesizing and simulation software tools namely the Modelsim 5.7 and the Xilinx 9.2i respectively and also creates environment for VLSI based design. The established habitat will be useful material and guide for graduate engineers and practitioners involve in VLSI based design. To describe the working flow efficiently, a case study design of ECC processor over prime filed is considered which involved Montgomery inversion algorithm [17] using efficient adder and data selector. The design is implemented and verified in Xilinx spartan3E family.

## II. XILINX 9.2i

Xilinx is a one of the major synthesizing tool [4] used in digital designs with low cost mode. Synthesis is the process of converting a high-level description of the design into an optimized gate-level representation, given a standard cell library and certain design constraints. A standard cell library can have simple cells, such as basic logic gates like and, or, and nor, or macro cells, such as adders, mux, and special flip flops. A standard cell library is also known as the technology library. A gate-level net list is a description of the circuit in terms of gates and connections between them. Synthesis tools ensure that the gate-level net list meets timing, area, and power specifications. The gate-level net list is input to an Automatic Place and Route tool, which creates a layout. The layout is verified and then fabricated on a chip. Some major applications used in Xilinx are image compression, image transformation, secure communication, testing application etc. Xilinx tool has numerous beneficial and features in synthesis process [4]. Some of them are specifying the different source types, performing the synthesis, simulation, post simulation process, implementation details of device, generating the program file for implementing in the kit and test bench etc.

### *A. Types of Sources*

Xilinx permit us to adding the range of file types to the project. Xilinx tool perform the synthesis operation depending upon the file type to be preferred by the user. Implementing the digital design in the form of language, VHSIC (very-high-speed integrated circuits) hardware description language (VHDL) module and Verilog module source type is used. Fig.1 shows that types of sources used in Xilinx tool to synthesis the digital design and implementing in the hardware.

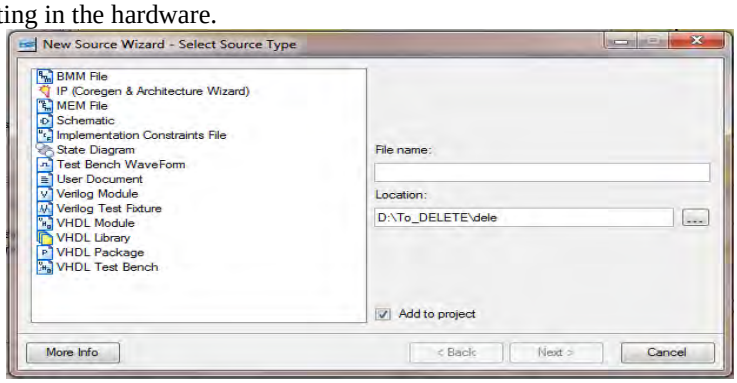


Fig. 1 Types of sources used in Xilinx 9.2i

### *B. Mixture of Operations*

Xilinx tool abides to perform the assortment of operation to design the digital circuits and implementing in the hardware. Fig.2 shows that mixture of operation in Xilinx. The varieties of operations that can be performed are Behavioral simulation, Synthesis/Implementation, Post translate, Post map and Post route. Behavioral simulation used to perform the initial simulation of the design using various simulators namely Modelsim, ISE simulator etc.

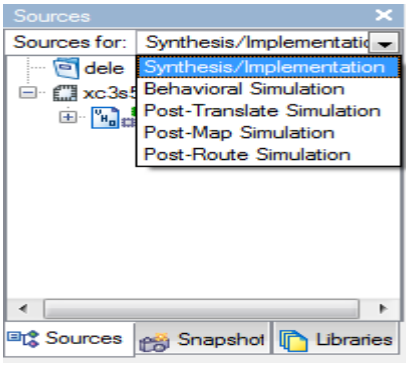


Fig.2 Mixture of operation in Xilinx 9.2i

Synthesis/Implementation process is for verifying the design during hardware implementation. Post translate, Post Map and Post route simulation process for verifying the Translate, Map and Route process after the Hardware design verification respectively.

### *C. Synthesizing and Implementation*

Synthesis is a manner of verifying the digital design whether suitable to be adapted to specified hardware or not. In Xilinx tool, the synthesis process is performing the various operations viz. generating synthesis report in text file, generating RTL schematic, generating technology schematic and check the syntax errors of the design. Fig.3 shows that synthesis and implements operation in Xilinx. Fig.4 shows that RTL schematic view of simple digital design. Fig.5 shows the Technology Schematic view of digital design**.**

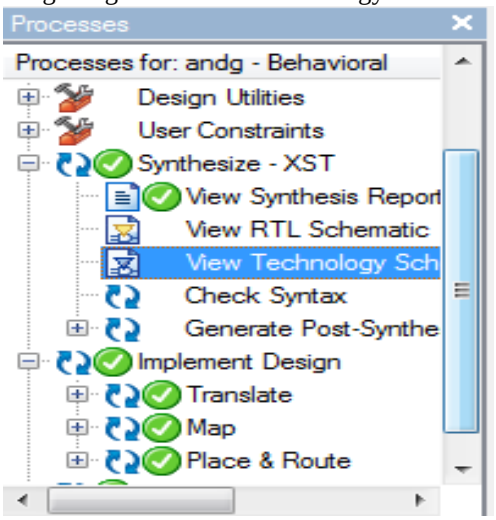


Fig. 3 Synthesis and implement in Xilinx 9.2i

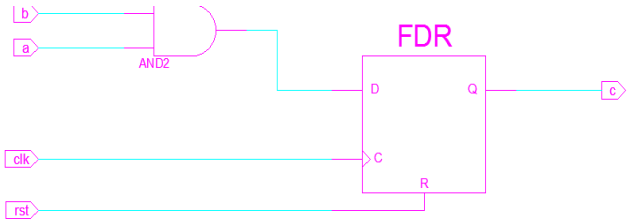


Fig.4 RTL schematic view of simple design

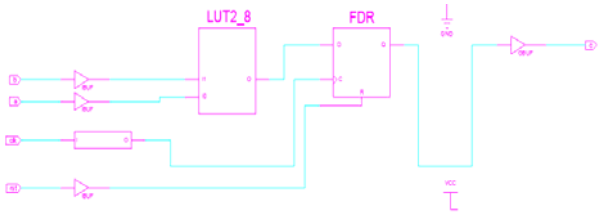


Fig.5 Technology schematic view of simple design

*D. Generating a Program File*

Xilinx allows the practitioner to generate a program file for the entire design of the project. It contains the Programming file generation report, Generate PROM, ACE or JTAG file and Configure device (iMPACT).

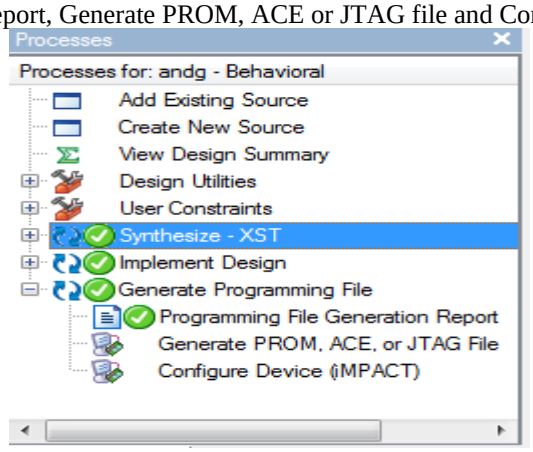


Fig.6 Generating a program files

Programming file generation report is the process for generating the BIT extension file of the project**.** It generates the bit files in the name of top module entity (project name). Generating PROM, ACE for establish the system interface with device. Configure device for download the bit file into specified device.

*E. Test Bench*

Hardware design engineers using any VHDL often need to test RTL code using a test bench. Given an entity declaration writing a test bench skeleton is a standard text manipulation procedure. Every design unit in a project needs a test bench. Generating test bench skeletons automatically can save hours per project. However, a little Perl programming can reduce that time to seconds in future. Fig.7 shows that process of test bench creation in Xilinx tool.

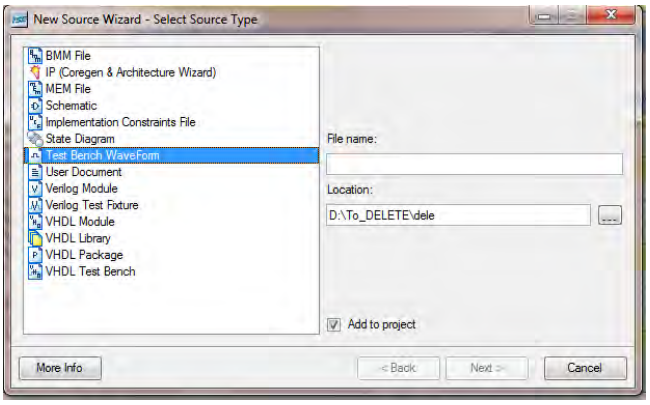


Fig. 7 Test bench creation in Xilinx 9.2i

Comparing to other competitors of Xilinx namely Lattice, Altera and Actel, it achieves the user friendly process in synthesis and implementation. The Xilinx tool allows the variety of file types to obtain the digital design and different level of verification in functional and hardware is achieved. RTL and technology schematic views are easy to obtain in Xilinx than other competitors. Xilinx provides the separate file for test bench wave form.

## III. MODELSIM 5.7

Modelsim5.7 tool is mainly used to verify the functionality of designed work [5]. It has several advantageous and features over the other competitors. Some of the features are simulation options, easy to force the value and user friendly waveform. The functionality of the design block can be tested by applying stimulus and checking results. There are two distinct components in a simulation: a design block and a stimulus block. A stimulus block is used to test the design block. Once the stimulus block is completed, the design is ready to run the simulation and verify the functional correctness of the design block. Different test benches can be used to thoroughly test the design block.

### *A. Simulation Option*

Modelsim5.7 provides the various options for compilation and simulation operation. Fig.8 shows the variety of compilation operations namely compile select for compiling the selected files, compile all for compiling all the files in the project and compile the files in hierarchical order using compile order option. Fig.9 shows the variety of option in simulation. It provides the option for radix type, executing time and iteration limits of execution.

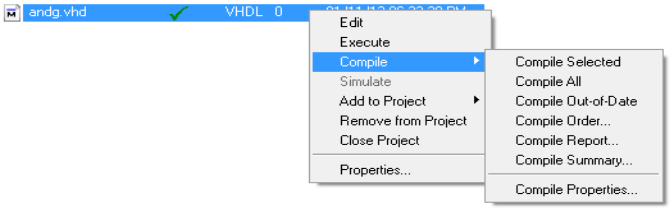


Fig. 8 Compilation options in modelsim5.7

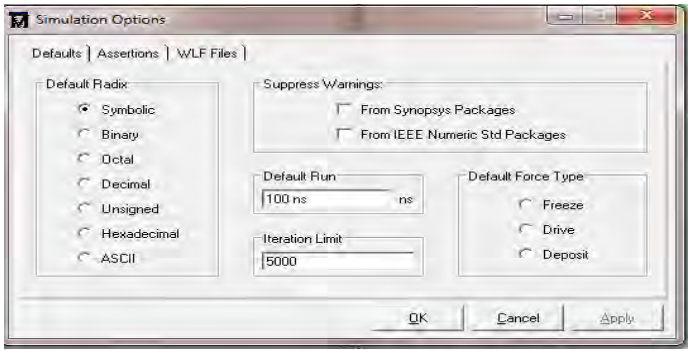


Fig.9 Simulation options in modelsim5.7

### *B. Value Forcing*

Fig.10 shows the window for forcing the values to input of the digital design. User defined delay of signal to derive is possible by using the force selected signal window., which needs to assign the signal value to the Value option in dialog box.

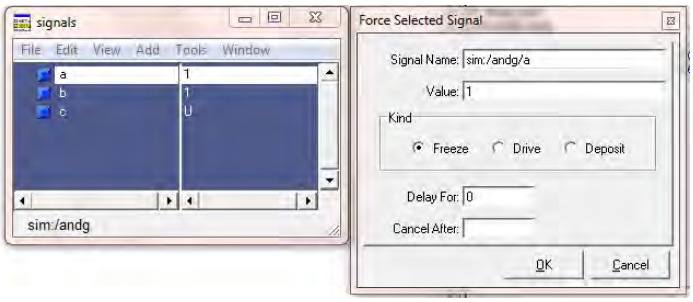


Fig. 10 Simulation options in modelsim5.7

### *C. Waveform Window*

Modelsim5.7 provides the user friendly waveform window for easy access of signal value and executing time. Fig.11 shows that waveform window of modelsim5.7. Using radix user can change the type of the signal to be availed in wave form window. User can run and reset the signal values by using tools menu.

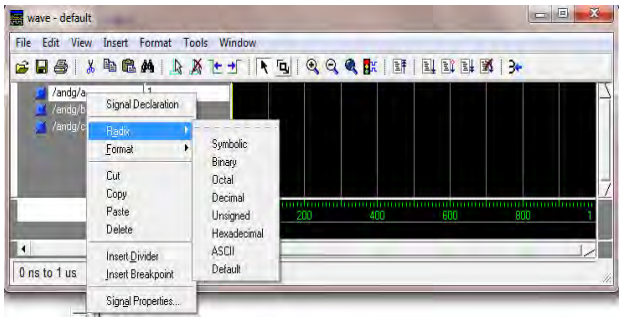


Fig.11 Waveform in modelsim5.7

Modelsim5.7 act as user friendly tool in terms of obtaining the signal value, forcing the value, compilation and simulation process in contrast to other competitors namely ISE simulator and ALDEC simulators.

## IV. XIINX 9.2i AND MODELSIM5.7 – DESIGN FLOW

### *A. Xilinx9.2i*

Fig.12 shows the executing methodology of Xilinx 9.2i tool [4]. Initially user requirements are implemented using HDL language. Using check syntax option in Xilinx tool, any syntax error of HDL programming can be identified. Then synthesis and implement the design using implement design option in Xilinx process. Any errors in synthesis or implementation process may be corrected and then the design is reconstructed. BIT file to be generated using programming file generation option. The design to be implemented using that generating BIT file.

### *B. Modesim5.7*

The executing methodology of Modelsim5.7 simulation tool is shown in Fig.13 [5]. Initially user requirements are implemented using HDL language. Compiling of the HDL program is done using compile option for verifying the syntax of HDL language. If any violation in syntax is found, it must be re-constructed by rectifying the errors. Simulation process also helps in finding the language violation and the violation needs the changes of design methodology. Waveform and signal window used to verify the functionality of the design.

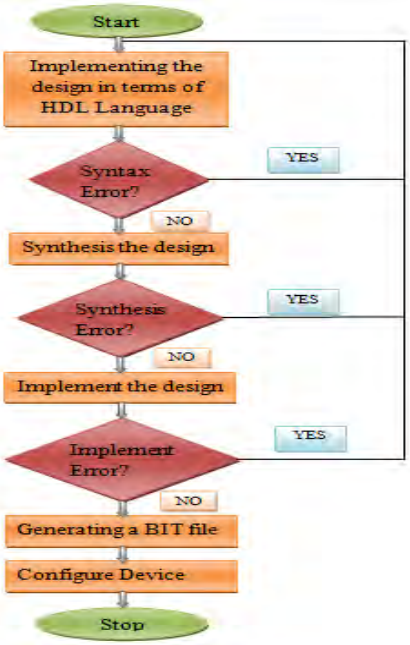


Fig.12 Flow chart of Xilinx flow

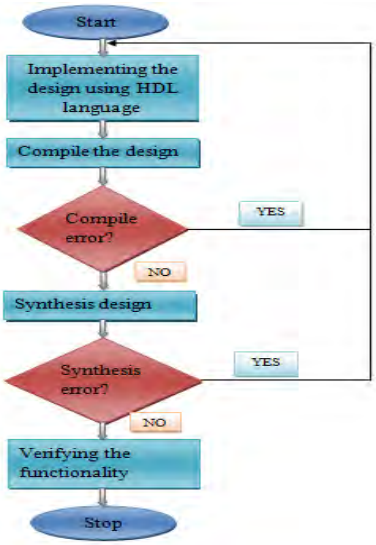


Fig. 13 Flow chart of Modelsim flow

## V. CASE STUDY: ECC CO-PROCESSOR

The most common public key schemes today are based on RSA [18] and Elliptic Curve Cryptography (ECC) [19], [20]. Elliptic curve cryptography is used to provide the security in communication system with small length key. ECC contains various numbers of operations and properties, among which scalar multiplication is a most timing consuming and complex operation. The numbers of research works are being carried out to reduce the time require performing the scalar multiplication operations. Since ECC is used in the portable device, it needs less area, low power and less time consuming operations are require to generating the key. Various methodologies such as parallelism, pipelining and robust can be used to design the ECC processors. These should be used in scalar multiplication while designing the ECC processor because of complexity and time constraints. To perform the scalar multiplication operation proficiently in ECC, an efficient algorithm is required. Montgomery scalar multiplication algorithm [23] mostly used in ECC processor.

Although ECC offers shorter bit-lengths and, thus, faster calculations, RSA will stay with us for the foreseeable future for legacy reasons. A good candidate for the next popular public key variety is Pairing Based Cryptography (PBC) [21], which seems to outgrow its research phase. The abstraction levels shown in Fig.14

are a variant of the classification extended by PBC. The lowest level constitutes the Modular Arithmetic as basis for all schemes. For ECC and PBC the level labeled as Intermediate Algebraic Structure follows, containing operations in the elliptic curve group and the extension field. Upon this builds the Cryptographic Main Function, which is directly required in the Cryptographic Scheme on the fourth level. This scheme, in turn, is used by the application on the System level.

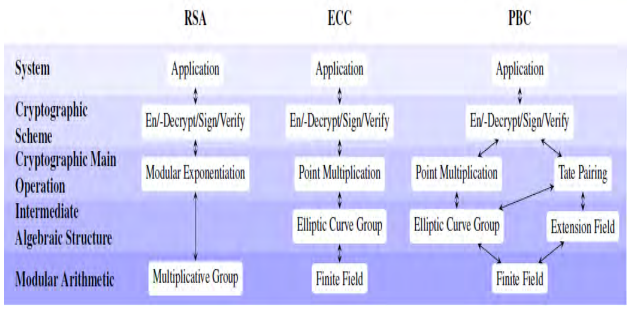


Fig.14 Abstraction levels for Public-Key Cryptography

The points required to be considered while designing an efficient ECC processor are:

- Since ECC is based on the discrete logarithmic problem, needs an efficient methodology to select a point 'P' on the curve and scalar value k.
- Used different methodology (Multiplier) to improve the scalar multiplication such as Q=kP. Where Q is the point on the elliptic curve.
- Using the key value Q, perform the elliptic curve cryptography operation.

The various methodologies are used in scalar multiplication to improve performance in terms of area, power and speed of ECC processor. So a powerful tool is required in predicting the performance measurements of ECC. Based on the performance prediction the better methodologies are identified. Xilinx is one of the powerful and simple tool to predict the performance, using this can identified the better methodology. Modelsim used to verify the functionality of ECC processor in various methodologies.

Modular multiplier, modular adder/subtractor and inverters are the very functional blocks of a typical ECC co-processor. Fig.15 shows the typical slice distribution of the various blocks within the functional unit. The modular multiplier is the largest circuit occupying 4 times more area than the modular adder/subtractor.

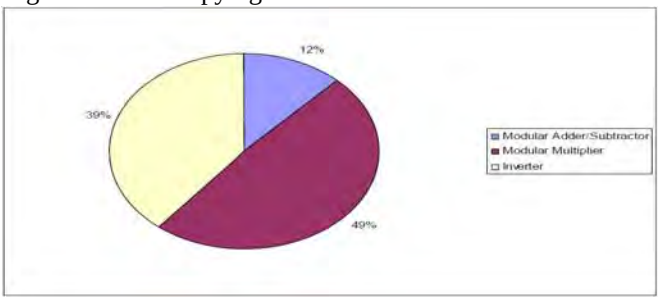


Fig. 15 .Slice distribution of blocks within functional unit.

The Fig.16 [22] shows the sample device utilization summary of ECC processor over GF(p). It is designed with LFSR as data selector and carry look-ahead adder (CLA) to perform the addition operation. A 160 bit prime field operation over GF(p) on parallel mode is written in VHDL language and synthesized in Xilinx 9.2i and stimulated using Modelsim 5.7. Point key generation of 160-bit with 200 MHz frequency is performed at look-up-tables (LUTs) of 2008, flip-flops (FFs) of 682 and Gate count is 22,689.

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** | **Note(s)** |
| Number of Slice Flip Flops | 682 | 9,312 | 7% | |
| Number of 4 input LUTs | 1,987 | 9,312 | 21% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 1,034 | 4,656 | 22% | |
| Number of Slices containing only related logic | 1,034 | 1,034 | 100% | |
| Number of Slices containing unrelated logic | 0 | 1,034 | 0% | |
| **Total Number of 4 input LUTs** | 2,008 | 9,312 | 21% | |
| Number used as logic | 1,987 | | | |
| Number used as a route-thru | 21 | | | |
| Number of bonded IOBs | 18 | 232 | 7% | |
| IOB Flip Flops | 8 | | | |
| Number of GCLKs | 1 | 24 | 4% | |
| **Total equivalent gate count for design** | 22,689 | | | |
| Additional JTAG gate count for IOBs | 864 | | | |

Fig 16 : Sample Device Utilization Summary

Fig.17 [22] shows ECC architecture power summary, which includes dynamic and static power consumed. Dynamic power is defined as amount of power consumed by switching activities of FF, where as static power is power consumed by leakage current. In 200MHz operation the Coprocessor consumes 79mW in static and 96mW in dynamic in the total summation of 175mW. The various power summaries can be attained by changing the frequency values. Thus Xilinx and Modelsim perform a critical role in designing complex architectures and they could create the habitat for the user domain.

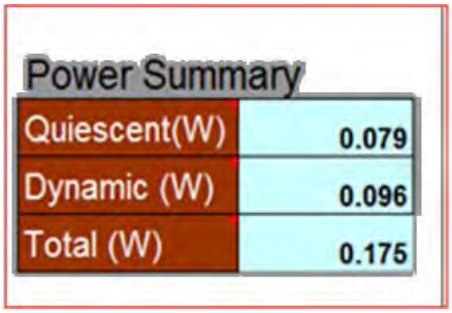


Fig.17 ECC architecture power summary

## VI. CONCLUSION

Xilinx is powerful synthesizable software for implementing the various types of digital design applications namely Bio-metric, Face detection, neural networks and etc in hardware.This paper has explored the features of trendy synthesizing and simulation software tools namely the Modelsim 5.7 and the Xilinx 9.2i respectively and also creates environment for VLSI based design. Elliptic curve cryptography (ECC) is widely used public key encryption techniques and it is fabulously and easily implemented in hardware using Xilinx and the functionality of ECC processor is verified using Modelsim. The key parameters of ECC design such as area utilization, clock cycle, time analysis and power summary are generator in Xilinx. The adapted case study of design an ECC processor could be useful guide for beginners and practicing cryptographists.

## REFERENCES

[1] http://en.wikipedia.org/wiki/Bernard_V._Vonderschmitt
[2] http://en.wikipedia.org/wiki/Ross_Freeman
[3] Samir Palnitkar, " Verilog HDL: A Guide to Digital Design and Synthesis", Second Edition, Prentice Hall PTR, 2003
[4] www.xilinx.com
[5] http://mentor.com
[6] Christian Beckhoff, Dirk Koch, and Jim Torresen, "The Xilinx Design Language (XDL): Tutorial and use cases", *Proceeding of the 6th IEEE international workshop on Reconfigurable Communication-centric Systems-on-Chip* (ReCoSoC'11), Montpellier, Jun 2011, pp.1-8.
[7] Jacques L Athow, and Asim J Al-Khalili, "Implementation of large-integer hardware multiplier in Xilinx FPGA", *Proceeding of the 15th IEEE International Conference on Electronics, Circuits and Systems* (ICECS'08), St.Julien's, Sep 2008, pp.1300-1303.
[8] Nor Fadzilah Mokhtar, AfafRozan Mohd Radzol, and Suzana Ab. Rahim, "Xilinx ISE software teaching aid for Diploma's students", *Proceeding of the IEEE International Conference on Engineering Education* (ICEED'09), Kuala Lumpur, Dec 2009, pp.56-59.
[9] Fabrizio Ferrandi, Giovanna Ferrara, Roberto Palazzo, Vincenzo Rana, and Marco D. Santambrogio, " VHDL to FPGA automatic IP-Core generation: A case study on Xilinx design flow", *Proceeding of the 20th IEEE International Conference on Parallel and Distributed Processing Symposium (IPDPS'06)*, Rhodes Island, Apr 2006, pp.4.
[10] Melanie Po-Leen Ooi, "Hardware Implementation for Face Detection on Xilinx Virtex-II FPGA using the Reversible Component Transformation Colour Space", *Proceeding of the 3rd IEEE International Workshop on Electronic Design, Test and Applications(DELTA'06)*, Kuala Lumpur , Jan 2006, pp.41-46.

[11] Vıctor Montano, Manuel Jimenez, "Design and Implementation of a Scalable Floating-point FFT IP Core for Xilinx FPGAs", *Proceeding of the 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS'10)*, Seattle, Aug 2010, pp-533-536.

[12] Bahram Rashidi, Bahman Rashidi, Majid Pourormazd, "Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on xilinx FPGA", *Proceeding of the 3rd IEEE International Conference on Electronics Computer Technology (ICECT'11)*, Kanyakumari ,Apr 2011, pp.18-22.

[13] Saeid Taherkhani, Enver Ever, Orhan Gemikonakli, "Implementation of Non-Pipelined and Pipelined Data Encryption Standard (DES) Using Xilinx Virtex-6 FPGA Technology", *Proceeding of the 10th IEEE International Conference on Computer and Information Technology (CIT'10)*, Bradford, Jul 2010, pp.1257-1262.

[14] Bin Zhou, David Hwang, "Implementations and Optimizations of Pipeline FFTs on Xilinx FPGAs", *Proceeding of the IEEE International Conference on reconfigurable Computing and FPGAs* (ReConFig'08), Cancun, Dec 2008, pp-325-330.

[15] J.-Y. Lai and C.-T. Huang, "A highly efficient cipher processor for dual-field elliptic curve cryptography", *IEEE Transactions on Circuits System- II, Expr. Briefs*, vol. 56, no. 5, pp. 394–398, May 2009.

[16] J.-Y. Lai and C.-T.Huang, "High-Performance Architecture for Elliptic Curve Cryptography over Binary Field", *Proceeding on IEEE international Symposium on Circuits and System (ISCAS)*, paris, Jun 2010, pp.3933-3936.

[17] IEEE 1363, Standard Specifications for Public key Cryptography, 2000.

[18] N.Koblitz, "Elliptic Curve Cryptosystems", *Mathematics of Computation*, vol.48, pp.203–209, November 1987.

[19] V.S.Miller, "Use of elliptic curves in cryptography," *in CRYPT0 '85*, pp.417-426, 1986.

[20] NIST, Recommended elliptic curves for federal government use, May 1999. http://csrc.nist.gov/encryption

[21] Ralf Laue, H. Gregor Molter, Felix Rieder, Sorin A. Huss and Kartik Saxena, "A Novel Multiple Core Co-Processor Architecture for Efficient Server-based Public Key Cryptographic Applications", *Proceeding of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'08)*, Montpellier, Apr 2008, pp.87-02.

[22] B.Muthukumar and S.Jeevananthan, "Performance Enhanced Co-Processor for Elliptic Curve Cryptography over GF (p)", *European journal of Scientific Research*, Vol.68, No.4, pp. 544-555, January 2012.

[23] B.Muthukumar and S.Jeevananthan, "Hybrid Low Power Encoded Multiplier for Montgomery Modular Multiplication and Efficient ECC Processor", *CIIT International Journal of Networking and Communication Engineering*, Vol.4, No.1, pp.8-15, January 2012.