A 1024 – Bit Implementation of the Faster Montgomery Multiplier Using VHDL

David Narh Amanor
University of Mines and Technology,Tarkwa, Ghana

1. Introduction

This paper presents a 1024-bit implementation of the recently proposed Faster Montgomery algorithm for performing modular multiplication. The object of this paper is to show how complex microelectronic systems or architectures could be modelled, simulated, synthesized and emulated on an FPGA (or fabricated as an ASIC) through the use of the industry standard language VHDL (IEEE-1076) as the design entry. The implementation used higher levels of abstraction to partition the complex design into subsystems or components. These components were implemented as independent functional blocks before being wired together to construct the Faster Montgomery architecture. The VHDL implementation was simulated and synthesized for a Xilinx Virtex FPGA. But, the code could also be used as the design entry for an ASIC. The work concludes by specifying the hardware requirements needed for the fabrication of the Faster Montgomery architecture.

2. Main Results

The design capture of the Faster Montgomery architecture was done using VHDL and the functionality was verified by simulation in Modelsim with 1024-bit input variables generated from a software implementation in JAVA. The reader is referred to [1], [2] and [3] for a detailed treatment of the Faster Montgomery architecture.

After the simulation, the synthesis tool generated the design report for the implemented architecture. The figures for the minimum clock period, number of gates, flip-flops, and function generators are tabulated in Table 1. The percentage of configurable logic block slices used and the overall area requirements for the implementation are available in [3].

Table 1: Design Statistics

No. of Gates	No. of Dff or Latches	No. of Function Generators	Period and Frequency
3163	5134	3180	20.398ns(49.024 MHz)

These figures represent the overall hardware requirements for the complete multiplier including the block of registers for holding the input data bits.

The minimum clock period and absolute time for the implemented architecture is as shown in Table 2. The absolute time is derived from the minimum clock period by

Absolute time = (Minimum period)(No. of clock cycle).*
Where, *No. of clock cycles = 2*bitlength (n) + 1*

The last clock cycle is used to trigger the values of the *sum* and *carry* from the internal registers inside the loop to the outside for display or further post processing. In these implementations, post processing outside the loop is omitted.

Table 2: Absolute Time

Precision	Absolute time (ns)
1024 bits	41795.502

3. Conclusion

In this paper, we performed a VHDL implementation for the Faster Montgomery multiplier that executes a 1024-bit inputs modular multiplication in less than 42 micro-seconds. The paper demonstrates, how a microelectronic system could be modelled, simulated, and synthesized for a Xilinx Virtex 2000E target device through the use of the industry standard language VHDL (IEEE-1076) as the design entry.

References

[1] V. Bunimov and M. Schimmler, "Area and Time Efficient Modular Multiplication of Large Integers," in IEEE 14th International Conference on Application Specific Systems, Architectures and Processors, June (2003).

[2] D. Amanor, V. Bunimov, C. Paar, J. Pelzl and M. Schimmler, "Efficient Hardware Architectures for Modular Multiplication on FPGAs", International Conference on Field Programmable Logic, Reconfigurable Computing, and Applications. August 24-28, (2005), Tampere, Finland.

[3] D. Amanor, "Efficient Hardware Architectures for Modular Multiplication." February (2005) http://www.crypto.rub.de/theses.html