

# Efficient Synthesis Approaches over Reconfigurable Computers

Esmail Chitalwala<sup>1</sup>, Tarek El-Ghazawi<sup>1</sup>, Kris Gaj<sup>2</sup>, and Nikitas Alexandridis<sup>1</sup>

April 25<sup>th</sup>, 2003

<sup>1</sup>George Washington University

<sup>2</sup>George Mason University

## Introduction:

Over the last few years many advances have been made in the world of Reconfigurable computing. With the evolution of high performance FPGAs, which not only offer speed but also offer a high gate count, it has become possible to develop high performance computers based on close interoperability between microprocessors and FPGAs.

Such an environment demands a quantum change in the programming model used in such a kind of environment. The HC-36 Hypercomputer coupled with VIVA<sup>®</sup> forms a highly versatile environment using which we can program the underlying FPGAs to perform tasks in parallel, which would normally have executed sequentially on a processor. Using this kind of an environment it is possible to cut down the time to solution and create an environment whereby an application can be created and run immediately on a Reconfigurable Computing machine.

In this paper, we overview the architecture and programming model used in the HC-36 “Hypercomputer<sup>®</sup>”. With help of the RC5 Encryption, Decryption and Key Generation Algorithms we compare the efficiency of Viva<sup>®</sup>, the high level programming environment, as a synthesis tool against conventional synthesis tools used to synthesize and map designs written in Verilog onto FPGAs. We demonstrate how Viva transforms high-level code into logic circuitry that runs on one or more FPGAs. We also show as to how the size of the design impacts the time taken by VIVA to synthesize a schematic constructed in VIVA.

We also assess the impact of this programming paradigm on time to solution as well as ease of use and level of hardware design knowledge for application developers.

## A Brief Summary of Experiments:

Unlike other environments, which use HDLs to program their FPGAs, VIVA<sup>®</sup> uses a “schematic capture type” language, referred to as *Implementation Independent Algorithmic Description Language (Refer to Appendix A)*, to describe the kind of circuit to be created on the FPGAs. VIVA<sup>®</sup> aims to shield the programmer from the complexity of the underlying FPGAs making it easy for a person who is not familiar with HDL to program them in the least possible amount of time.

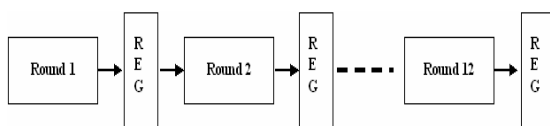


Fig.1 RC5 Encryption Structure with Pipelining between the rounds  
(13 Pipeline stages in all)

In this paper we try and compare the efficiency with which VIVA<sup>®</sup> can synthesize and place and route a design as compared to a conventional synthesis tool like Synplify Pro 7.2 combined with Xilinx<sup>®</sup> Place and route Tool 4.2.

To achieve this end we make use of the RC-5 Encryption Algorithm, and compare the performance and efficiency of synthesis in VIVA<sup>®</sup> as against a third party tool like Synplify and Xilinx. We compare two different versions of the algorithm written in both

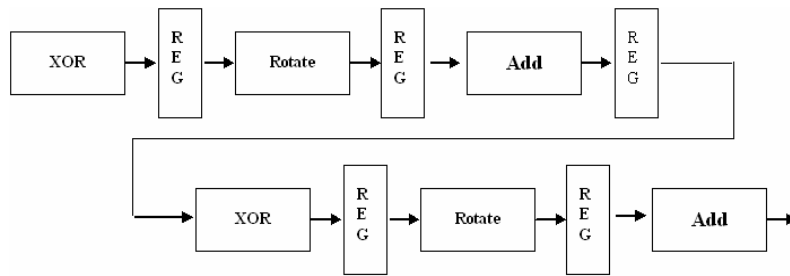


Fig.5. Registers present within one round of the RC5 Encryption module to give a net of 7 pipeline stages.

VIVA<sup>®</sup> and Verilog. Both the implementation have been kept similar to each other as far as possible in terms of pipeline latency and I/O. We also assess as to how the size of a design impacts the efficiency of the tools under consideration.

### Conclusion:

Two different versions of the RC5 Encryption algorithm were designed and tested using two varying approaches and programming paradigms. Also both the Encryption and Decryption have been synthesized for both the pipelined versions in VIVA<sup>®</sup> and tested. The programming environment of VIVA<sup>®</sup> shows a lot of potential in fast design and implementation of applications although as a tool it is has yet to reach its full potential. In some cases VIVA<sup>®</sup> performs as good as Synplify/Xilinx, but VIVA<sup>®</sup> has yet to mature as a synthesis tool and many features have yet to be incorporated like Incremental Synthesis techniques and retiming. Although the programming paradigm for VIVA<sup>®</sup> is pretty different from what the conventional hardware designer is used to, once mastered it has the potential to cut down the time from concept to implementation of designs drastically. Designs in VIVA can run at any speed below the PCI-X bus speed, which itself is pre-settable at 4 different frequencies.

**Keywords:** Hypercomputer<sup>®</sup>, VIVA<sup>®</sup>, RC5 Encryption, Synthesis

### Corresponding Author:

Tarek El-Ghazawi  
 E-mail: tarek@seas.gwu.edu  
 Phone: +1 202 994 2607

Address:  
 ECE Department  
 George Washington University  
 801 22<sup>nd</sup> Street NW,  
 Washington DC 20052, U.S.A.

### Areas:

- Reconfigurable/ Adaptive Computing Systems
- Hardware and Software: The Line is Blurring
- Encryption Systems
- Systems and Software