

Performance Comparison CORDIC Implementations on the SRC-6E Reconfigurable Computer

Russ Duren, Douglas Fouts, and Dan Zulaica
Naval Postgraduate School, Monterey, California

Abstract: The architecture and programming environment of the SRC-6E Reconfigurable Computer was presented in the 2002 MAPLD International Conference [1]. That paper described how the programmer could trade off chip area for execution speed. The chief advantage of the SRC-6E environment is that programmers need very little knowledge of the reconfigurable hardware. The environment allows code written in a high level language, either C or FORTRAN, to achieve significant performance improvement with little modification. This article compares the performance of a CORDIC algorithm written in C and compile using the SRC-6E compiler to the performance obtained using a highly optimized hardware IP (Intellectual Property) core.

Introduction

The SRC-6E reconfigurable computer provides a state of the art platform for developing and executing programs that take advantage of multi-million gate Xilinx FPGAs combined with Intel Pentium Processors. The architecture of the SRC-6E has been designed so that it can be rapidly modified to leverage advances in both personal computers and FPGAs. The programming environment of the SRC-6E allows programmers to utilize the resulting computing power with minimal consideration of the hardware itself [2].

The SRC-6E compiler supports programming in two high level languages (HLL). Programs are developed and debugged in C or FORTRAN in a Linux environment. The intent is to support programming entirely in the HLL. When performance demands additional speed, the SRC-6E compiler allows the user to develop highly optimized macros using Verilog, VHDL or Schematic capture or macros that incorporate commercially available Intellectual Property (IP) Cores. These macros are accessed by function calls within the HLL source program. From the programmer's point of view this is similar to calling assembly language routines for increased performance on a standard microprocessor.

The SRC-6E compiler allows the programmer to specify portions of code to be executed on either FPGA. The compiler translates the HLL user code to highly pipelined operations on the FPGAs. Pipelining allows complicated HLL routines to produce results every 10 nanoseconds. Additional speed increases may be obtained through the use of parallel computations. To a large extent, the SRC-6E compiler frees the programmer from having to think about the target hardware when developing code.

A traditional compiler produces object code that executes on a processor; the SRC-6E compiler produces both object code and a hardware design. The hardware design is contained in FPGA configuration files that are linked together with the object code in one executable file. Therefore, the performance of the resulting "code" depends on how well the HLL is translated into hardware. An experiment was developed to test the performance of the SRC-6E compiler. A program was written implementing a

CORDIC arctangent function. Initially, the CORDIC routine was written in C. Next, a CORDIC IP core was substituted for the C version. The CORDIC IP core was generated using the Xilinx CORE Generator program. The IP core was integrated into the second version of the program using the SRC-6E macro capabilities.

The results of the experiment are described in the paper. The description discussed the steps required to develop both programs and compares the performance of the resulting code. Both timing and area (FPGA utilization) are examined.

RELATED WORK

FPGAs and ASICs have been used for high performance digital signal processing for many years. Parhi discusses VLSI design techniques that trade off speed, area, and power dissipation [3]. Chodowiec discusses performance gains made possible by reconfigurable computing [4]. He examines gains and limitations obtained from pipelining, parallel circuitry, and loop unrolling. Peterson and Drager list a number of reconfigurable computing platforms and discuss their application to defense-related processing [5].

REFERENCES:

1. Fidanci, O.D., Diab, H., El-Ghazawi, T., Gaj, K. and Alexandridis, N. "Implementation trade-offs of Triple DES in the SRC-6e Reconfigurable Computing Environment," 5th MAPLD International Conference, Laurel, Maryland, September, 2002.
2. SRC-6E C Programming Environment v1.3 Guide, SRC Computers, Inc. April 11, 2003
3. Parhi, K.K., "VLSI Digital Signal Processing Systems, Design and Implementation," John Wiley & Sons, NY, 1999.
4. Chodowiec, P.R., "Comparison of the Hardware Performance of the AES Candidates Using Reconfigurable Hardware," Master's Thesis, George Mason University, Spring 2002.
5. Peterson, G.D. and Drager, S.L., "Accelerating Defense Applications Using High Performance Reconfigurable Computing," GOMAC, 2003.