Considerations for Algorithm Selection and C Programming Style for the SRC-6E Reconfigurable Computer

Russ Duren and ***Douglas Fouts***
Naval Postgraduate School

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. In this paper we discuss additional aspects of programming the SRC-6E. In particular we examine classes of algorithms that best benefit from the SRC-6E's architecture. Additionally, we examine how the SRC-6E compiler interprets various C language constructs.

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 current version of the SRC-6E allows the programmer to access two Xilinx Virtex II XC2V6000TM FPGAs. The two FPGAs share 24 MB of RAM referred to as onboard memory or OBM. The Pentium processors and FPGAs communicate over a 64-bit interface with an 800 MB/s peak bandwidth. This bus can be used for message passing and for DMA transfers between the PC's memory and OBM. Each FPGA is connected to the OBM with six buses that provide an aggregate peak bandwidth of 4800 MB/s. The two FPGAs are interconnected by three buses with an aggregate peak bandwidth of 2400 MB/s. Additional interface buses are provided to connect each FPGA to off board data sources and sinks. The latter interfaces can be used for real-time embedded processing.

The architecture of the SRC-6E allows programmers to realize significant speed improvements when executing certain classes of algorithms. The speed improvements result from the use of the resources provided by the FPGAs. In practice, the speedup depends greatly on the properties of the algorithm. Certain algorithms are better suited for implementation in an FPGA. Such algorithms will benefit most from the SRC-6E. Although the SRC-6E provides very high bandwidth interconnections, certain algorithms can saturate these resources limiting performance gains. This paper examines these issues.

The SRC-6E compiler supports programming in two high order languages (HOL). Programs are developed and debugged in C or FORTRAN in a Linux environment. The intent is to support programming entirely in the HOL. When performance demands additional speed, the SRC-6E compiler allows the user to develop highly optimized macros using Verilog, VHDL or Schematic capture. These macros are accessed by function calls within the HOL 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 HOL user code to highly pipelined operations on the FPGAs. Pipelining allows complicated HOL 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. Still, certain HOL constructs can lead to vastly different performance. This paper addresses this further.

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 discus their application to defense-related processing.

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.