A SoC design platform "Redefis" using Dynamically Reconfigurable Processor "Vulcan"

1 Introduction

As System on a Chip(SoC) design is becoming more complex, design's weight is increasing. Designing a SoC with the popular approach consisting of integrating a set of ASICs coupled with a core processor is effective, but ASICs are expensive to design and time consuming to map an application on. Consequently we propose a new approach to design SoC using "Redefis", a novel SoC design platform.

Redefis conciliates ease of design with high performance by exploiting the advantages of both software and hardware approaches. Redefis compiler generates custom instruction set from application program described in C language, and maps it on a reconfigurable hardware. We are also developing the reconfigurable hardware for Redefis named "Vulcan". Vulcan is a processor has dynamically reconfigurable data path.

2 SoC design Platform "Redefis"

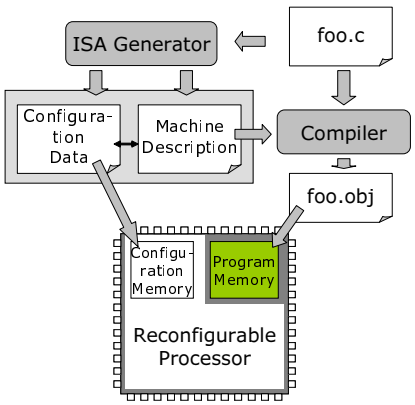

Figure 1: Redefis compiling flow

Redefis consists of Redefis toolchain that generate object code from application program and reconfigurable hardware that execute it. The overall structure for implementing application on Redefis shows in figure1. Firstly, ISA Generator generates apprication-specific Instruction set from analyzing source code, configuration data and machine description to realize application-specific Instruction for the reconfigurable processor. Secondly, The redefis compiler generates object code for the reconfigurable processor. Finally, Reconfigurable Processor embedded on SoC executes application program. Currently ISA Generator is under development.

3 Dynamically Reconfigurable Processor "Vulcan"

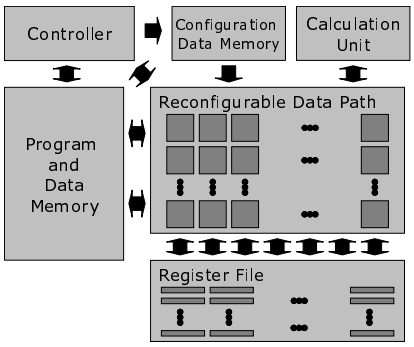

Figure 2: The overall view of Vulcan architecture

The overall view of Vulcan architecture is shown in Figure2. Configuration Memory has configuration data for every instructions. The structure of Reconfigurable Data Path (RDP) is an array of Programmable Elements (PE). PE is fine grain reconfigurable logic. Vulcan also includes a Calculation Unit consisting of multiplier and adder in order to accelerate arithmetic operations. In the execution flow of Vulcan, the Controller firstly fetches the instruction and loads configuration data into RDP. Secondly RDP or the Calculation Unit loads data and executes the instruction. Finally, RDP outputs calculated data. Currently we have Vulcan implemented on FPGA and plan to produce a Vulcan test chip in future.