Redefis: using dynamically reconfigurable instruction set architecture processor for SoC

1 Overview of the Redefis project

The growing complexity of systems and its related production cost of chips have imposed big constraints in SoC design of new systems. The classical approach of using general purpose processors (GPP) mostly are unable to fit the tight performance and power constraints. The tighter TAT and TTM constraints for a whole ASIC design solution is not affordable either due to its complexity and cost.

The Redefis system is an SoC design platform for high level, fast implementation of ASIPs [1].The platform is composed of a reconfigurable instruction-set processor and a set of design tools. The developed processors can be used as flexible co-processors in a MPSoC [2] systems or as standalone processor/engines.

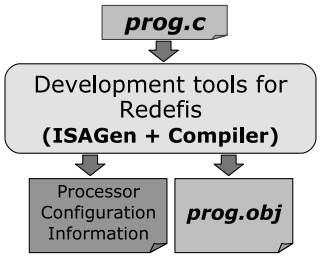
Compared to other approaches based of reconfigurable processor, The Redefis project targets processors where the ISA (Instruction Set Architecture) is fully redefinable and not extended. Redefis also proposes a design flow based on standard C programs (no HDL writing is necessary).

In next section we present the Redefis design tool chain and in the last one, a case-study architecture of an ISA dynamically reconfigurable processor (Vulcan).

In the demonstration we are going to present the full design flow as well the Vulcan working board executing the newly generated ISA for the DES encryption algorithm.

2 The Redefis design tool chain

The Redefis design tool chain is used both to compile an input C program ("the application" written in high level C) and to produce an ISA specifically optimized for the given input application. The chain flow of the design tools is given in the figure bellow.



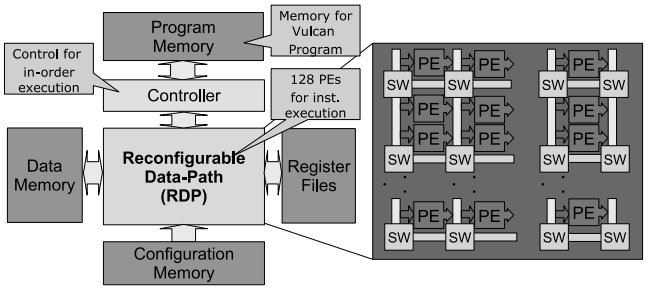
The input is a C program (like other GPPs), and the output is both the binary code of the program and the newly defined ISA as configuration binary streams.

[1] Application Specific Instruction Set Processors
[2] Multi Processor SoC

An important tool in the design flow is the ISAGen (Instruction Set Architecture GENerator) which will produce a new ISA specifically optimized for the input C program; it will also preprocess the C program to reflect the generated set of instructions. Then a retargetable compiler compiles the resulting C program into the target reconfigurable processor's object code.

3 Vulcan: an ISA dynamically reconfigurable processor

Vulcan is an implementation of a dynamically reconfigurable processor where the ISA is fully redefinable (at compile time) and the execution of its instructions is made reconfiguring the processor data-path at every cycle. The figure gives an overview of its architecture.



Vulcan's computation power and flexibility is due to its RDP (Reconfigurable Data Path) module which is a reconfigurable network of processing elements (PE). Additional calculation units are present for more demanding arithmetic computations. The controller is responsible for fetching instructions, reconfiguring the RDP and handling flow changes (e.g. branching).

In the processor every "custom instruction" (identified from the application's source code) is associated with one configuration of the RDP. The full set of configurations loaded into the configuration memory is the current ISA of the processor (defined at compile time).

A typical execution cycle of Vulcan is performed in 3 steps:

- The controller fetches an instruction and reconfigures the RDP.
- The RDP fetches the data and passes it to be computed in the RDP (for every custom instruction).
- The RDP writes back the result of the calculation.

We have implemented an emulator of the Vulcan processor on an FPGA board and a chip implementation of the processor as well.