

Evaluating Application Performance Between DSP Processor and GPP Using Reconfigurable Hardware

Nicola, Eduardo V.
Ruzicki, Julio C.M.
Martins, Luis H.J.
Mattos, Júlio C.B.

Group of Architectures and Integrated Circuits - GACI
Federal University of Pelotas - UFPel
Pelotas - Brasil



Group of Architectures
and Integrated Circuits



- 1** Introduction
- 2** The Reconfigurable Hardware
- 3** Case Study
- 4** Results and Conclusions

Introduction



- The industry and market of electronic devices grows rapidly each day
- Most products now have builtin microprocessors to handle specific tasks
- This is made possible due to the fast technology development, in response to market demands
 - General purpose processors (GPPs)
 - ASICs (Application Specific Integrated Circuits)

Introduction

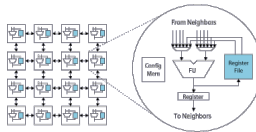


- However, it is possible to perform those tasks in other kinds of processors:
 - **GPPs** were used before the idealization of DSP processors
 - **ASIC** were another option of reduced costs and flexibility

Motivation

■ Flexible Specialization

- A field that is growing in research is Reconfigurable Architectures
- This area of research aims to improve processor performance without increasing its clock rate
- This is made using specialized hardware structures to perform portions of program code



The work was divided in stages:

- 1 Choice of tools to simulate DSP processors and simulate the Mips32 processor;
- 2 Selection of typical DSP applications;
- 3 Use of simulators to generate execution traces from the DSP applications;
- 4 Comparison of results between the DSP processor and the MIPS32, with and without the Reconfigurable Hardware.

Three commercial tools were analysed as possible DSP processor simulators for the work:

Code Composer

- Features
 - Development
 - Debugger
 - Profiler
 - Real-time-operation system



Symphony Studio

- Works with DSP563xx/DSP567xx processors
- Features
 - Can be executed on Eclipse environment
 - GDB (GNU Project Debugger)
 - Project Manager



VisualDSP++

- Features
 - Statistical performance measurements
 - Emulation
 - Simulation
 - Compilation

VISUALDSP++[®]



Group of Architectures
and Integrated Circuits



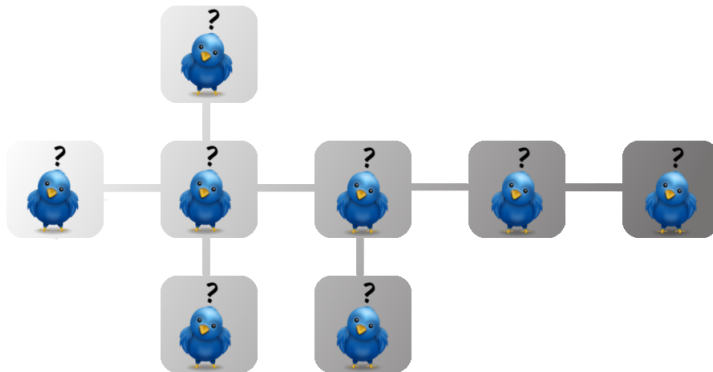
OVPsim

- Features
 - Describing platforms with one or more processors
 - Models are available for many standard processors
 - Compilation
 - Trace Generator



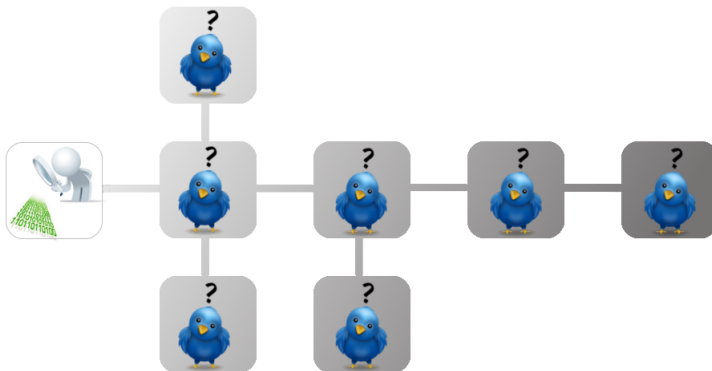
The Reconfigurable Hardware

Pipeline Diagram



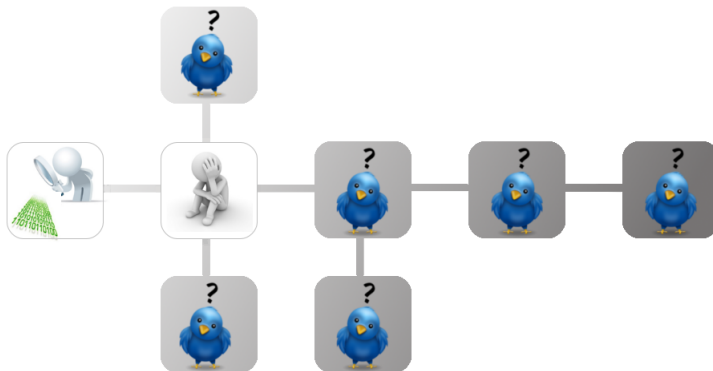
The Reconfigurable Hardware

Instruction Fetch



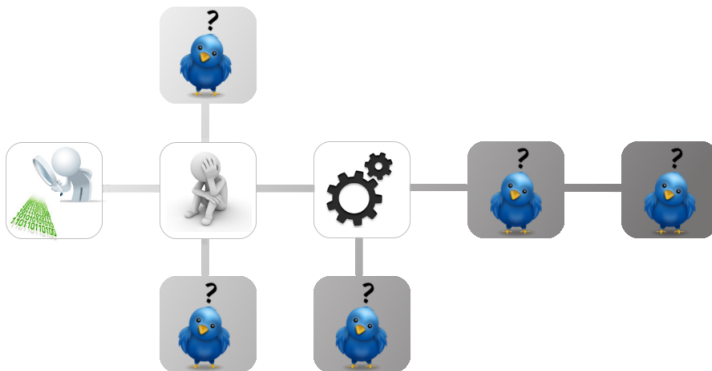
The Reconfigurable Hardware

Instruction Decode



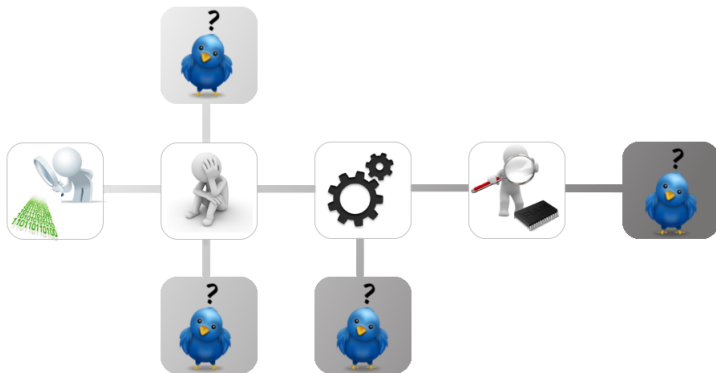
The Reconfigurable Hardware

Execution



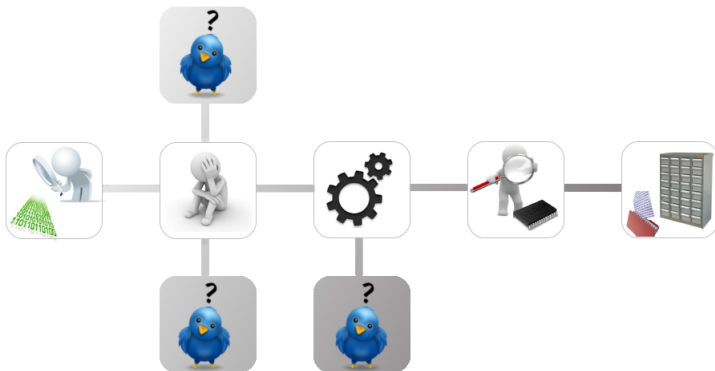
The Reconfigurable Hardware

Memory Access

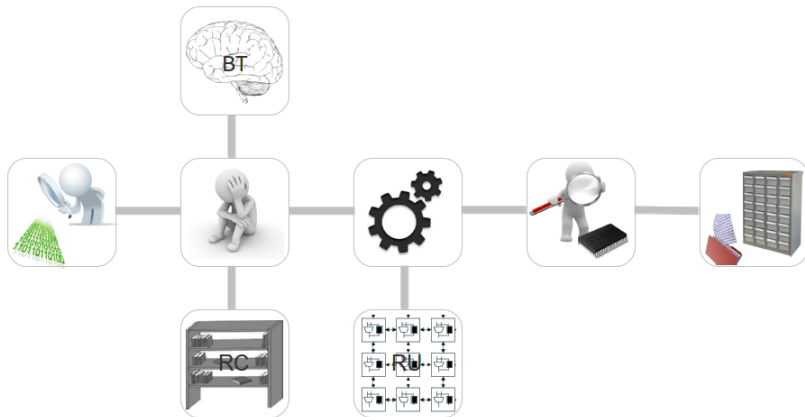


The Reconfigurable Hardware

Write Back



The Reconfigurable Hardware



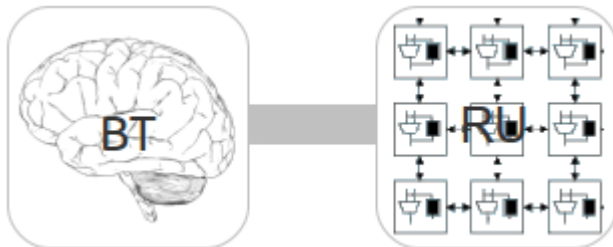
- The Reconfigurable System works as a functional unit

The Reconfigurable Hardware



- The Binary Translator works parallel with MIPS pipeline
 - At DI stage, BT reads the PC indexed address

The Reconfigurable Hardware

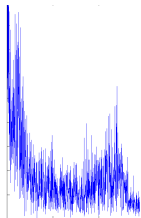


- The Binary Translator works parallel with MIPS pipeline
 - At DI stage, BT reads the PC indexed address
 - After decoding the instruction, BT sends the instruction to RU, when possible, as a configuration to be made on RU

The Reconfigurable Hardware

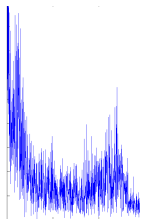


- The Binary Translator works parallel with MIPS pipeline
 - At DI stage, BT reads the PC indexed address
 - After decoding the instruction, BT sends the instruction to RU, when possible, as a configuration to be made on RU
 - BT also stores the instruction configuration or group of instruction configurations on RC
 - When PC points to an existing address on RC, RU takes the execution from CPU.



- A generic and customized benchmark was made with DSP-like operations
 - 1 Performs a multiplication of two real vectors
 - 2 Two matrix addition
 - 3 Matrix subtraction
 - 4 Matrix scalar multiplication
 - 5 Calculates the mean of the input array
 - 6 Calculates the RMS of the elements in the input array
 - 7 Accumulates the two vector product
 - 8 Two matrix addition

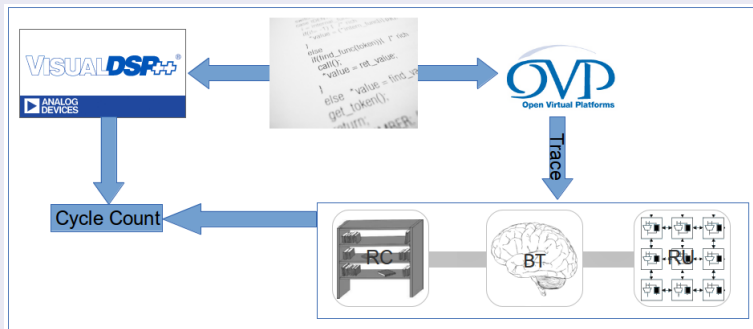
Case Study



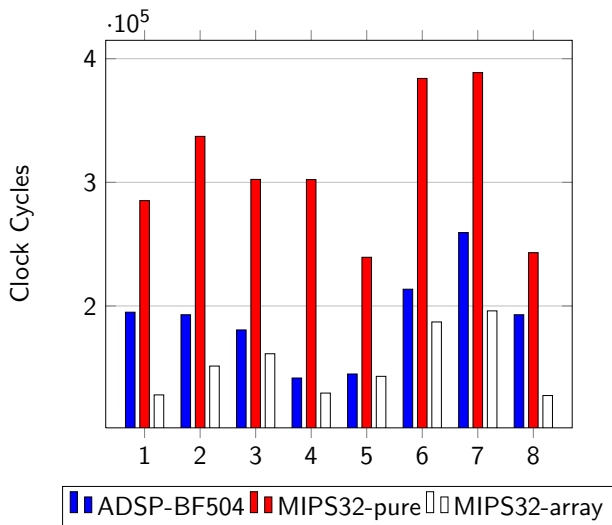
- 9 Executes the Discrete Fourier Transform
- 10 Executes the Inverse Discrete Fourier Transform
- 11 Executes the Fast Fourier Transform
- 12 Executes the Inverse Fast Fourier Transform
- 13 Executes the Finite Impulse Response

Case Study: Methodology

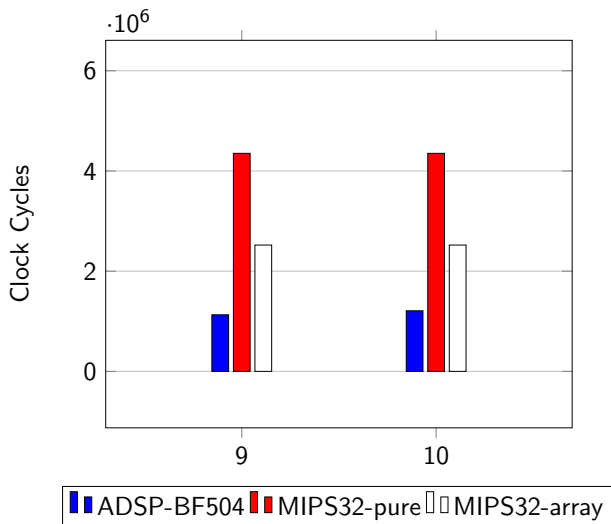
The Work-flow



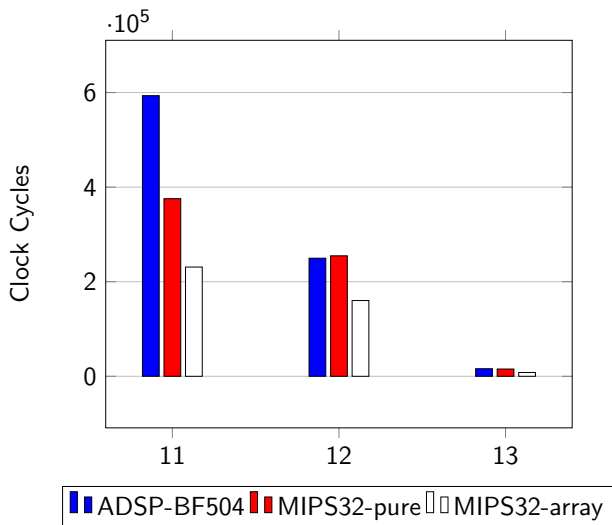
Case Study: Results



Case Study: Results



Case Study: Results



Case Study: Results

- MIPS32 had less clock cycles with the Reconfigurable System
- Blackfin ADSP-BF504 is a specialized DSP processor, comparing to MIPS-pure(without array), but MIPS-array had less clock cycles in most applications

Conclusions

- This work presented the simulation of DSP applications to compare Blackfin and MIPS32
- Even Blackfin ADSP-BF504 being a specialized DSP processor, MIPS-array had less clock cycles in most applications
- A GPP processor with Reconfigurable System can optimise program execution, compared to DSP processors
- For future works: Other kinds of simulations
 - 1 Power estimates
 - 2 Energy consumption

Evaluating Application Performance Between DSP Processor and GPP Using Reconfigurable Hardware

Eduardo Nicola

Julio Ruzicki

Luis Martins

Júlio Mattos

{evnicola, jcmruzicki, lhjmartins, julius}@inf.ufpel.edu.br



Group of Architectures
and Integrated Circuits

