COFFEE™ RISC Core

Overview

The RISC Team @
Institute of Digital and Computer Systems

The Idea Behind The Project

- Open cores well suitable for our research purposes were unavailable.
- The core can be used in other projects later on
- The core will be open source and freeware
- Design of a core deepends the knowledge on risc architectures and system design overall.
- Core can be extended to a complete processor

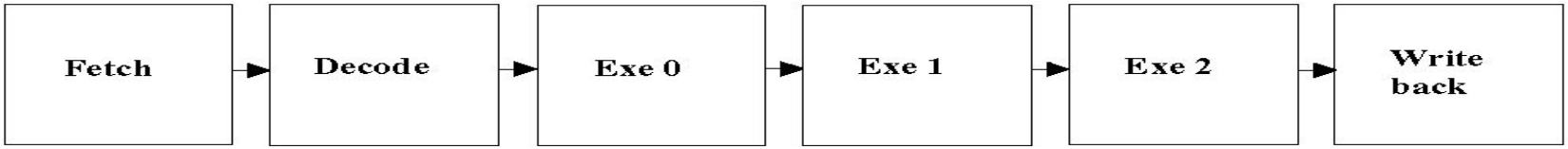
Members of the Team

- **Juha Kylliäinen**
  - **The original designer of the core, currently working elsewhere**
- **Marko Savolainen, Tuukka Kasanko**
  - **Currently debugging, documenting and modifying the core**
- **Markus Moisio**
  - **Porting the GNU C-compiler to this architecture**
- **Winnie Tsang**
  - **Porting libraries for the compiler.**
- **Ugnè Tenikaitienè**
  - **Designer of assembler, disassembler and linker for this architecture**
- **Petri Metsäpuro**
  - **Designing instruction set simulator for the core**
- **Anna Auvinen**
  - **Developing an operating system for the core.**

Instruction Set Architecture Overview

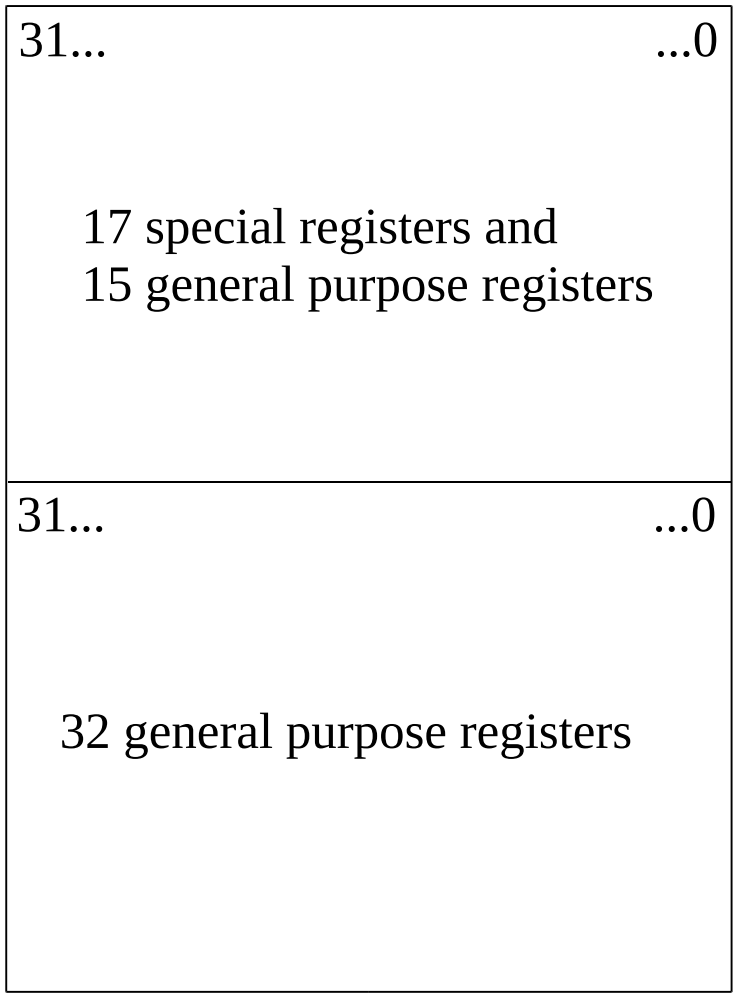
- 6 Stage Pipeline
- 68 Instructions
- Conditional execution of most of the instructions
- Two different encodings 16- and 32-bits
- 2 x 32 registers
- Superuser and user memory areas
- Harward architecture
- Instruction set resembles Arm’s instruction set.
- Co-processor interface (hardware accelerator interface)
  - supports 4 co-processors
  - for example for a floating point unit
- Support for 8 direct and 255 memory mapped interrupts

Pipeline of the Core



- **Decode**
  - 16-bit instructions are expanded to 32 bits
  - Control signals needed in the next cycle are extracted and decoded
  - Execution conditions are verified
  - Register operands are fetched
  - immediate constants are expanded
  - The target address of a PC relative branch is calculated
- **First execution cycle**
  - Condition flags are calculated
  - Data memory addresses are calculated
  - Data address validity is checked
  - Instruction address validity is checked
- **Second execution cycle**
  - Data memory access
- **Third execution cycle**
  - First execution of the *mulhi* instruction, which multiplies the upper 32-bits of a 32 x 32-bit multiplication.

Registers



- **2 x 32 registers (superuser, user)**
- **The superuser has 17 special purpose and 15 general purpose registers.**
- **The user has 32 general purpose registers**

Current Status

- A few architecture specific documents ready
- User guide and other user support documentation under development
- One Master Thesis coming up
- Functional verification in progress

Conclusions

- **Basic pipelined risc architecture**
- **Modular design style, system can be customized according to user's needs.**
- **Can be used as a stand-alone chip in for example embedded systems**
- **Can be used as an IP block in a SoC design**
- **In 0.35 μm standard cell CMOS technology the core area is approximately 2.2 mm² and the clock frequency is approximately 100 MHz.**

Ideas for the Future

- Support for dynamic mapping of the superuser memory area. Other functional blocks around the core
  - e.g. Memory Management Unit, interrupt handler, bus interfaces, peripherals, co-processors, ...
- Different implementations of the core
  - Low power implementation
  - Smaller area implementation
- Developments tools, for example system simulator
- Operating system