Design of a 32-bit RISC Processor with Reduced Area

Hyo-Su Jeong Young-Kyu Kim Byungin Moon

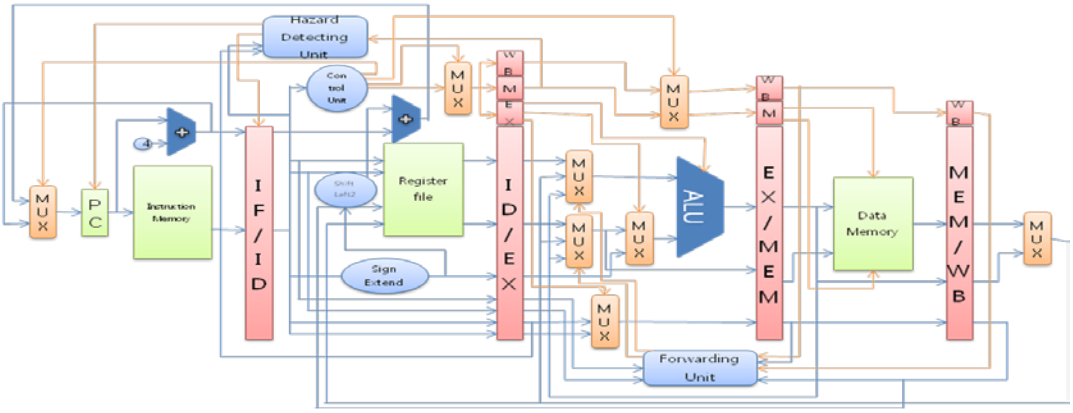
Kyungpook National University, Daegu, Republic of Korea
E-mail: {softjung, kyk79, bihmoon}@ee.knu.ac.kr

Extended Abstract

Design strategy of RISC-type processors is based on the insight that simplified instructions can provide higher performance. Besides, there is trade-off between gate counts and operating performance in embedded environment. So, we implement a processor design with reduced hardware resource. The proposed design is a 32-bit MIPS processor in which the data and instruction word sizes are 32 bits. It is pipelined to a depth of 5 stages and the implemented instruction set includes types of Register, Immediate, Jump, and Floating-point. Also we have developed an assembler and a compiler for the processor.

Main idea of the design is mixed-register architecture to reduce the gates sizes through simply configured Register File and the resource sharing between integer and floating-point units. Conventional processors with floating point support implement integer registers and floating-point registers separately. But this design uses only one register file with 32 registers. We choose to use a Xilinx ML403 evaluation board for our implementation.

Processor main features and specification :



(1) 5-stage Pipeline

Fetch ->Decode ->Execute ->Memory ->Write-back

Each pipeline register has the latency of one clock cycle and combinational logics are located between the pipeline stages.

Data and instruction memories are separately implemented. A control block generates data memory enable signal.

Register File with 2-read port and 1-write port is implemented using D flip-flops. To prevent data hazards, we provide bypassing paths.

The same logic blocks are shared integer and floating-point units. A CLA (carry-look-ahead) adder consists of 4-bit fixed-point adder blocks, which are shared between the 32-bit CLA adder of the arithmetic logic unit, and the 24-bit and 28-bit CLA adders of the floating-pint unit. Also bitwise operation and logic operation blocks shares resource blocks.

(2) Instruction Set Architecture

Implementing a processor with the complex instruction set would be extremely demanding of the available hardware resource. So, four types of instructions shown below are simply implemented.

R-type	Opcode(6)	Operand2(5)	Operand3(5)	Operand1(5)	Not reserved(11)
I-type	Opcode(6)	Operand2(5)	Operand1(5)	Immediate(16)	
J-type	Opcode(6)	Address(26)			
F-type	Opcode(6)	Operand1(5)	Sign(1)	Exponent(8)	Mantissa(12)

Only, arithmetic (*add*, *sub*, *addi*, and *subi*), logical (*and*, *or*, *xor*, and *not*), data transfer (*lw* and *sw*), bitwise (*srl*, *sll*, *sllf*, and *srlf*), conditional branch (*bne* and *beq*), unconditional branch (*jal* and *j*), and floating point (*addf*, *subf*, *sllf*, *srlf*, and *addfi*) instructions are implemented

(3) Register File

We have designed Register File that is capable of supporting integer and floating-point instructions. Some reserved registers are used for floating-point instructions and other registers are used for integer instructions. When an instruction tries to access an improper register, the assembler and the compiler generate a warning or error message.

0	Zero
1~3	Assemble Temporaries
4~7	Floating Point
8~15	Temporaries
16~25	Saved Temporaries
26~31	Reserved

We have implemented 5-stage pipelined processor, using only 13% of the LUTs and 19% of the BRAM units. We verified the compiler through translation of a test code. The processor operation was also verified.

References

[1] J. Hennessy, D. Patterson, Computer Organization and Design: The Hardware/Software Interface Fourth Edition, MORGAN KAUFMANN, San Francisco, 2009.

[2] M. Holland, J. Harris, S. Hauck, “Harnessing FPGAs for Computer Architecture Education” Conference on Microelectronic Systems Education, 2003

[3] J. Gray, “Designing a Simple FPGA-Optimized RISC CPU and System-on-a-Chip”, DesignCon’2001, online at http://www.fpgacpu.org/gr/index.html, 2001.

[4]B. Zivkov, B. Ferguson, M. Gupta, “R4200: A High-Performance MIPS Microprocessor for Portables”, Conference on IEEE International Computer Society,1994