Design of a 32 - bit core based on RISC – V Architecture

Navinkumar K – 19EC1067

Introduction

RISC – V is an open-source RISC processor specification. It was developed at UC, Berkeley. It is the fifth iteration in a series of RISC architectural designs done by UC Berkeley. In this project, 32 – bit Core based on the RISC – V ISA is designed.

Motivation

x86-based instructions (CISC) are difficult to decode and hence slower to execute. Moreover, since the x86 architecture has very less number of processor registers, most of instructions are executed in the memory or cache, causing more power consumption. The RISC – V Architecture has an advantage here with **31** general-purpose registers. All the instructions are of fixed width (32 – bit).

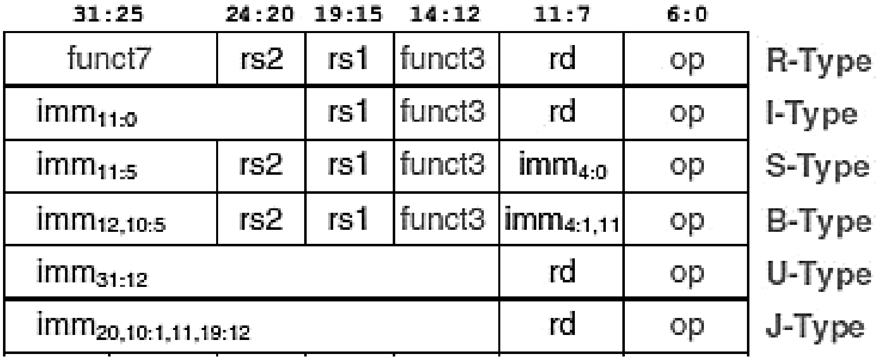
RISC – V is potentially suitable for various applications ranging from micro-power embedded devices to high-performance server-based cloud multiprocessors. Most importantly, RISC – V is an **open-source** ISA unlike most other architectures.

Languages, Tools and Technologies used

Verilog, System Verilog, Intel Quartus Prime Lite, ModelSim - Altera

Instruction Set of the processor

Our processor aims to support the **RV32I** Core instructions.



31:25	24:20	19:15	14:12	11:7	6:0	
funct7	rs2	rs1	funct3	rd	op	R-Type
$imm_{11:0}$		rs1	funct3	rd	op	I-Type
$imm_{11:5}$	rs2	rs1	funct3	$imm_{4:0}$	op	S-Type
$imm_{12,10:5}$	rs2	rs1	funct3	$imm_{4:1,11}$	op	B-Type
$imm_{31:12}$				rd	op	U-Type
$imm_{20,10:1,11,19:12}$				rd	op	J-Type

Figure 1: **RV32I** Instruction types and their formats

Instruction Execution Cycle

It has **five** stages: Fetch, Decode, Execute, Memory Access and Write back

Design Details

32 – bit processor with 5 stage **pipelined** microarchitecture based on RISC - V. It executes an instruction in five stages.

It also has a **Hazard Unit** to handle both Data Hazards and Control Hazards

System Verilog is used for behavioural modelling of the processor.

Intel Quartus Prime Lite will be used as the platform for generating gate – level netlist, and for visualizing the output waveform.

Performance Analysis

$$\boldsymbol{Execution\ Time} = (\#\boldsymbol{instructions}) * \left(\frac{\boldsymbol{cycles}}{\boldsymbol{instructions}}\right) * \left(\frac{\boldsymbol{seconds}}{\boldsymbol{cycles}}\right)$$

CPI (Cycles per Instruction) = 1, for Single Cycle Processor. The clock speed in a single-cycle processor is limited by the slowest instruction.

References

- Computer Organization and Design The Hardware Software Interface [RISC-V Edition], *David A. Patterson, John L. Hennessy*
- Digital Design and Computer Architecture, RISC-V Edition, 1st Edition, *Sarah L. Harris, David Harris*
- Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCs, and cloud servers, *Jim Ledin*
- Computer Architecture: A Quantitative Approach, 6th Edition, *John L. Hennessy, David A. Patterson*