Implementation of Super-scaler CPU design

March 2021

1 Team Composition

Shubham Kar, 180070058
Garaga V V S Krishna Vamsi, 180070020
Rupak Kundu, 203074008
Alok Kumar, 204076011
Anurag Agarwal, 203070056

2 Motivation

Superscalar CPU design extends the scalar pipeline design for a CPU to achieve ideal instructions per cycle of less than 1. Increasing the **width** of the pipeline helps in issuing more instructions in a single clock cycle. This however complicates the handling of pipeline hazards and more complicated the issuing of instructions become(out-of-order instruction issue), more hardware is required for handling the same. Therefore, there is a tradeoff of performance and hardware while implementing a superscalar CPU design. We had already seen a single cycle data path CPU design in previous labs and have done pipeline implementation on a smaller scale in previous courses. Thus, we attempt to implement an extension of the idea to help us wrap our head around modern processor design.

3 Description

We attempt to extend the MicroMIPS structure as explained in [Par05] by including floating point addition and multiplication using a separate block and also implementing a superscalar CPU design of width 2 with 32 bit instruction word size. The D-cache will be multi-ported in the output as well as the input side. However, we will try to implement interleaving if we get enough time. I-cache will be made multi-output ported. Inter-stage buffers will be made accordingly. The Instruction set we plan to implement is shown in Table 1. The design template we desire to follow is as given in [SL13]. The trap management we desire will handle pipeline hazards and will handle some ALU exceptions like divide by zero etc. Simple 2-bit branch predictor will be made available.

General Type	**Instruction mnemonics**
ALU	add, addi, sub, subbi, slt, slti, srl, sra, sll, and, andi, or, ori, xor, xori
Mult/Divide block	mult, div, multi, divi
Branch	beq, bnz, j, jal, jr
Floating Point	fmult, fadd, faddi, fmulti
Memory	lw, sw

Table 1: Instruction set table

4 Methodology

4.1 First Week

We will be reading up on the CPU design and start with simple blocks like ALU and the other execution blocks for the first week. This is to ensure that all the team members are in perfect tandem for the project.

4.2 Second Week

Next we proceed to cache formation and branch management block in the following week. Trap management block will be started by the end of the week.

4.3 Third Week and afterwards

By the third week, trap management block will be finished and work on connecting the buffers and the logic for other stages will be started. Updates on the progress of the project after this point will be made available in weekly reports. Testbenches will be made with varying composition of instruction types and tested to verify the working of individual blocks in the whole picture by the last week.

References

[Par05] Behrooz Parhami. *Computer Architecture: From Microprocessors to Supercomputers (Oxford Series in Electrical and Computer Engineering)*. USA: Oxford University Press, Inc., 2005. ISBN: 019515455X.

[SL13] John Paul Shen and Mikko H Lipasti. *Modern processor design: fundamentals of superscalar processors*. Waveland Press, 2013.