CS 520 - All Sections - Fall 2021 - Ghose
Homework 1
Due: Wednesday, 9/22/21 by 11:59 pm

Please remember the academic honesty policy, its implications and submit your individual work.

ELECTRONIC SUBMISSION VIA BRIGHTSPACE IS REQUIRED

In the following problems, we also introduce the use a new notation (rather, operator) in the RTL language we are using to describe the semantics of the instructions. These notation is as follows:

The notation <RTL_Statement1>**;** <RTL_Statement2>, where the "**;**" implies that the operations represented by Statement1 and Statement2 are implemented in strict sequence with the operations corresponding to Statement1 done first, followed by the operations corresponding to Statement2. As an example, the notation:

R1 ← Result1 ; R2 ← Result 2, where Result 1 and Result 2 are the names for two latches in the WB stage, implies that the contents of Result1 is first written to R1, followed by the writing of Result2 to R2, both from the WB stage, as all writes to registers take place from the WB stage.

In general, the notation "; " can be used more loosely without explicit mention of where the operations are completed and what have to be done in parallel and where in the pipeline; these have to be inferred with the sequential constraint and pipeline constraints in mind.

The following problems require you to modify the 5-stage APEX pipeline by making changes to the APEX pipeline such as adding new connections, a separate adder that can work in parallel with the ALU in the EX stage and changing the number of register file ports. ***Also, assume that the ALU cannot be used twice in the same cycle.*** In ALL of the following problems, you will need to read all source registers from the same stage (namely, D/RF). You will also need to update register(s) from the same stage (namely, WB).

Please provide brief explanations to justify the design changes you have made to support the instructions for each problem.

Now for the problems….

Problem 1. Consider the following variant of a load instruction is added to the ISA of APEX:

```
LRR <dest> <src1> <src2>          /* Load and reset instruction */
```

where <dest> is the destination register for LRR, while <src1> and <src2> are its two source registers. The semantics of this instruction is as follows:

```
dest ← Mem [src1 + src2]; src1 ← 0
```

Don't forget the implications of the "**;**" in the RTL description of the instruction semantics!

Using a table similar to the one on Page 39 of the notes, describe the information flowing from one stage of the 5-stage APEX pipeline to the next stage. Remember to make sure that all register reads take place from D/RF and all register writes take place from WB.

Using a diagram similar to the one on Page 40 of the notes, draw and highlight the changes that need to be made to the APEX pipeline to support the instruction. Assume that the instruction being processed is **LRR R1, R2, R3** and the contents of the registers R2 and R3 to be 2000 and 1000, respectively, before the instruction is sent over to EX. Assume that the contents of the memory location to be read out to be 1234.

Problem 2. Consider now a variation of a store instruction, symbolically designated as **STR+** with the following format:

```
STR+ <src1> <dsrc2> #<signed_literal>
```

where <src1> and <dsrc2> are its two source registers. For the STR+ instruction, **dsrc2** also serves as a destination register, hence the "d" before "src1". The semantics of this instruction is as follows:

```
Mem [dsrc2 + signed_literal] ← src1; dsrc2 ← dsrc2 + 4
```

Don't forget the implications of the ";" in the RTL description of the instruction semantics!

Using a table similar to the one on Page 39 of the notes, describe the information flowing from one stage of the 5-stage APEX pipeline to the next stage. Remember to make sure that all register reads take place from D/RF and all register writes take place from WB.

Using a diagram similar to the one on Page 40 of the notes, draw and highlight the changes that need to be made to the APEX pipeline to support the instruction. Assume that the instruction being processed is **STR+ R1, R2, # -50** and the contents of the registers R1 and R2 to be 500 and 2000, respectively, before the instruction is sent over to EX.

Problem 3. Consider that the following variant of an ADD instruction is added to the ISA of APEX:

```
ADS <dest> <dsrc1> <src2>          /* Add and subtract instrn */
```

where <dest> is the destination register for ADS, while <dsrc1> and <src2> are its two source registers. The register <dsrc1> also serves as a destination register. The semantics of this instruction is as follows:

```
dest ← dsrc1 + src2; dsrc1 ← src2 – dsrc1
```

Again, don't forget the implications of the "**;**" in the RTL description of the instruction semantics!

Using a table similar to the one on Page 39 of the notes, describe the information flowing from one stage of the 5-stage APEX pipeline to the next stage. Remember to make sure that all register reads take place from D/RF and all register writes take place from WB.

Using a diagram similar to the one on Page 40 of the notes, draw and highlight the changes that need to be made to the APEX pipeline to support the instruction. Assume that the instruction being processed is **ADS R1, R2, R3** and the contents of the registers R1, R2 and R3 to be 500, 2000 and 1000, respectively, before the instruction is sent over to EX.