

CMSC611: Advanced Computer Architecture

Homework 2

Question 1:

(25 points)

Consider the following code from an if-else statement of the form

if (A==0)

 A = B;

else

 A = A + 8;

where A is at 0(R2) and B is at 0(R3):

```
                LD      R1, 0(R2)
                BNEZ   R1, L1
                LD      R1, 0(R3)
                J       L2
L1:             DADDI   R1, R1, #8
L2:             SD      R1, 0(R2)
```

Starting with a standard 5-stage MIPS pipeline with forwarding and branch resolution in the ID stage, you are asked to design a new conditional load instructions “LDZ Rd, x(Rs1), Rs2” and “LDNZ Rd, x(Rs1), Rs2” that do not load unless the value of Rs2 is zero or not zero, respectively.

- Write the code using this new conditional load instruction and show possible stall cycles that would occur in the pipeline.
- Compare the clock cycles used to that of the original code. If implementing the new conditional load instruction would increase the clock cycle by 10%, will it be worth making that change?

Question 2:

(15 points)

The instruction set architecture of MIPS has three 32-bit instruction formats known as I-type, R-type, and J-type. Suppose we want to increase the number of registers in the MIPS architecture from 32 to 64 without changing the instruction size.

- How many total bits should be reassigned to the register fields in each instruction type?
- If the bits are taken exclusively from the opcode, show the new bit layout for each instruction type. How many opcodes will this new architecture support?

Question 3:

(30 points)

Consider the following code in a 5-stage MIPS pipeline with forwarding. Assume the branch is resolved in the ID stage, and there is only one memory port to handle instruction and data. The initial value for R1 is *arrayBase* and for R8 is *arrayBase+40*.

```
Loop: LW      R2, 0(R1)
      SUB     R4, R2, R3
      SW      R4, 0(R1)
      LW      R5, 4(R1)
      SUB     R6, R5, R3
      SW      R6, 4(R1)
      ADDI   R1, R1, 8
      BNE    R1, R8, Loop
```

- a) Show all possible pipeline hazards in the code. Draw a pipeline chart and solve those hazards using stalls.
- b) Suppose there are enough memory ports, and you are allowed to schedule the code by changing the order of instructions, how does the scheduled code compare to the original unscheduled code in clock cycles?

Question 4:

(30 points)

You are asked to design a pipeline for a new instruction architecture. The unpipelined implementation of each instruction in this architecture has 6-cycle execution. The names for the stages in the pipeline are the same as those used for the cycles in the unpipelined implementation: IF = instruction fetch, RF = instruction decode and register fetch, ALU1 = effective address calculation for memory references and branches, MEM = memory access, ALU2 = ALU operations and branches comparison, WB = write back.

- a) Assume all stages in this pipeline are perfectly balanced, and the instruction is started every clock cycle. Draw the pipeline chart with a 6-stage pipeline for 6 instructions.
- b) Find out the number of adders needed. Use the pipeline chart obtained in question a) to show a case that maximizes the adder count.
- c) Find out the number of register read and write ports and memory read and write ports required. Use the pipeline chart obtained in question a) to show possible combinations of instructions and pipeline stage indicating the instruction and the number of read ports and write ports required for that instruction.