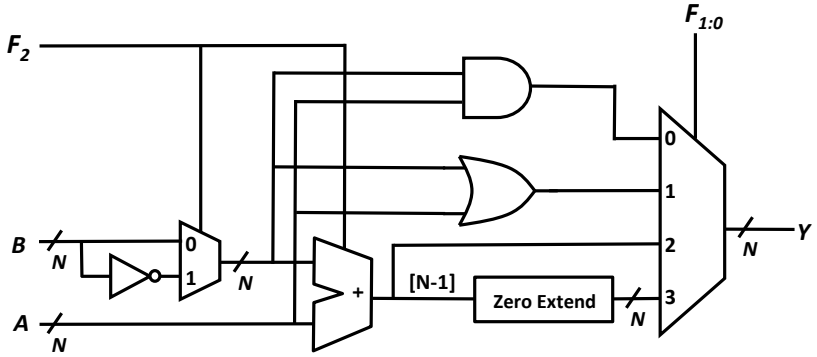
Exercises 4
Processor Design

Computer Organization and Components / Datorteknik och komponenter (IS1500), 9 hp
Computer Hardware Engineering / Datorteknik, grundkurs (IS1200), 7.5 hp

KTH Royal Institute of Technology
December 18, 2020

1. Consider the following circuit that shows an *Arithmetric Logic Unit (ALU)*.



   (a) Explain the meaning of signals A, B, F and Y.

   (b) Why is an ALU designed to have the capability to perform several different functions?

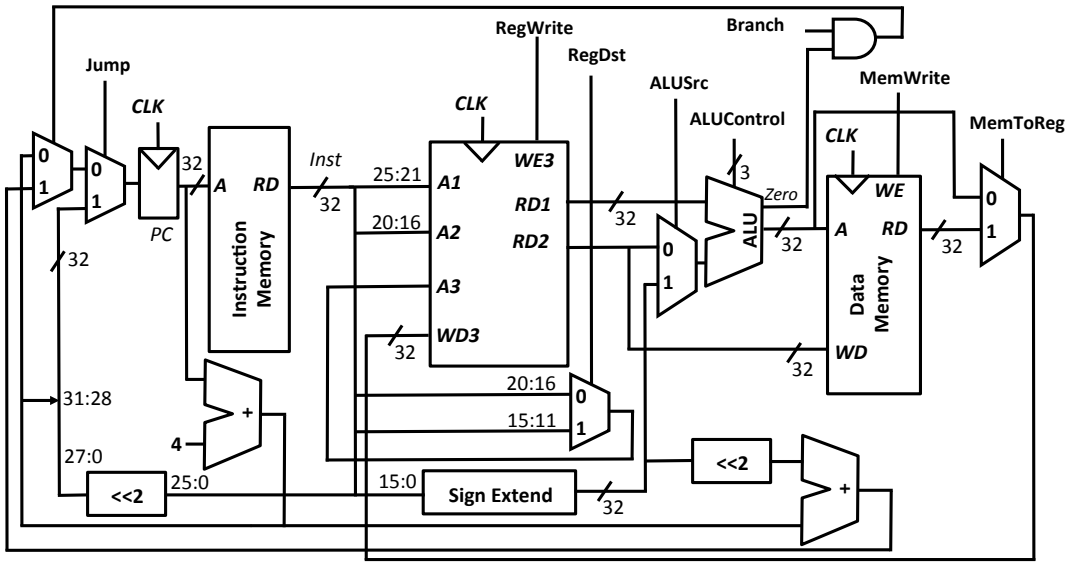
   (c) If $F = 5$, $A = \texttt{ff}_{16}$, and $N = 8$, what is then Y? Does the value of B matter? Why/why not?

   (d) What value has the F signal, if the ALU should compute integer subtraction, that is, $A - B$?

   (e) Which operation/function is performed if $F = 111_2$? Note that $[N - 1]$ means "select the most significant bit of the N-bit bus".

   (f) If $F = 101_2$, $N = 4$, $A = 10_{10}$, and $B = C_{16}$, what is then Y?

2. Consider the following Figure that shows the datapath of a single-cycle MIPS processor.



(a) If we assume that instruction `add $t3,$s0,$s1` executes, what is then the values of $A1$, $A2$, $A3$, and $RegDst$ after that all signals have stabilized?

(b) Assume that instruction `lw $t2,-4($s4)` executes and that register `$s4` contains value `0x0000ff08`. Assume also that we use the same ALU as in exercise 1. What are then the values of the control signals $Jump$, $RegWrite$, $RegDst$, $ALUSrc$, $ALUControl$, $Branch$, $MemWrite$, and $MemToReg$? Also, what values have the address signal A and the write data signal WD of the data memory?

3. Consider the figure in Exercise 2.

If the current machine code that executes is `0x214bfffd` and the values of the registers in the processor are as shown below, what is then the value of the input $WD3$? Answer as a 32-bit hexadecimal value.

```
$at = 0x00011021
$v0 = 0x5234f1a0
$v1 = 0x1114f111
$a0 = 0xff001231
$a1 = 0xffffffff
$a2 = 0x32252341
$a3 = 0xff1245ee
$t0 = 0xffff12ff
$t1 = 0xffffffff
$t2 = 0xfffffff5
$t3 = 0xfffff67f
$t4 = 0x0121ffff
$t5 = 0x55f7fff5
```

Assume that all other registers in the register file have value 0.

4. More on control signals. Consider the figure in Exercise 2. For each of the following statements, answer if the statement is true or false. Motivate your answer.

   (a) Control signal $RegWrite$ must be 0 when $MemToReg$ is 0, because $WD3$ can (in this cycle) otherwise get the wrong value.

   (b) The control signal $Branch$ must always be 1 when a `beq` instruction is executed.

   (c) The $RegDst$ control signal is only dependent on the 6 most significant bits of machine code of a MIPS instruction.

   (d) The control signal $ALUControl$ is only dependent on the 6 least significant bits of a machine code of a MIPS instruction.

5. Performance analysis and pipelining. For each of the following statements, answer if the statement is true or false. Motivate your answer.

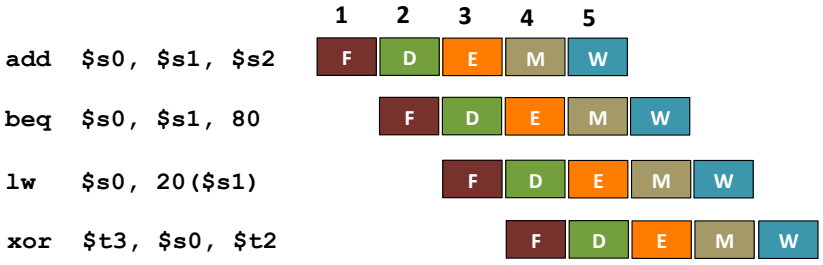
   (a) If a processor is using pipelining, the CPI (cycles per instruction) can be reduced compared to a single-cycled processor that is not pipelined.

   (b) By reducing the critical path in a processor, the processor can get better performance (shorter execution time on a selected set of benchmarks) because it might be possible to clock the processor at a higher frequency.

   (c) A 5-stage pipeline is the optimal way of designing a processor so that both execution time and energy consumption become as good as theoretically possible.

   (d) In a 5-stage MIPS processor, the decode stage is usually responsible for reading out register values from the register file.

6. Consider the following MIPS assembly instructions that are executed on a 5-stage MIPS pipeline.



   (a) Assume that comparison of register values for the `beq` instruction is done by the ALU. In such a case, will the `beq` result in any hazards? If so, what kind of hazard? How can it be resolved?

   (b) Are there any more data hazards in the example? If so, how can they be solved?

   (c) Assume that comparison of register values for the `beq` instruction is still done by the ALU. Assume further that the processor does not have any branch predictor and that the branch is statically assumed to be branch-not-taken. What is then the branch misprediction penalty for taking or not taking the branch at the `beq` instruction? Is there a hazard? What is this kind of hazard called? How can it be solved?

(d) Assume now that the comparison of register values for the `beq` instruction is done in the decode (D) stage, instead of in the execute (E) stage. Assume further that if the branch is taken, the program counter can be updated at the end of the decode stage. In what way changes the different kind of hazards that can occur, compared to the scenario when the comparison for `beq` was done by the ALU?