Computer Architecture and Programming Languages
Jacobs University Bremen
Fjolla Dedaj

Course: CO20-320241
18th of November, 2019

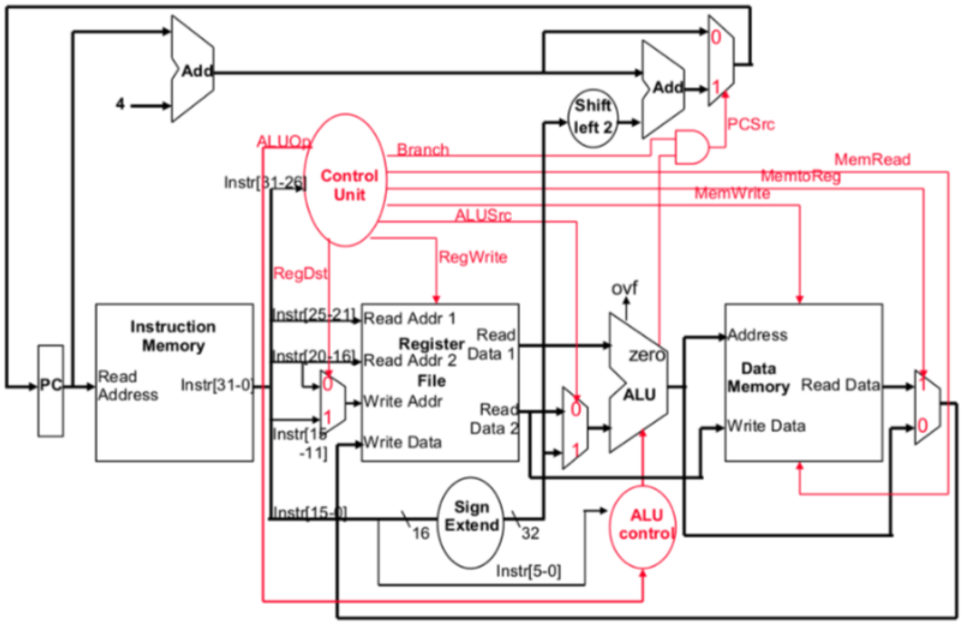
Homework 9

Problem 9.1

(a) Why does the PC not need an explicit write signal in a single-cycle datapath?
Solution:
If you check the single cycle control signals flowing in this question, the Jump and Branch control signals feed the muxes at the end of the pipeline, which determines the input to the PC. In the absence of these control inputs, the PC will be incremented by its default value, 4. Therefore there is no need of an explicit write control for PC in a single cycle data path.

(b) Why is an explicit write control signal needed in a multicycle datapath?
Solution:
This happens because there are a lot of operations happening in parallel, therefore, explicit control is necessary to determine which path to pick in determining the address of the next instruction.

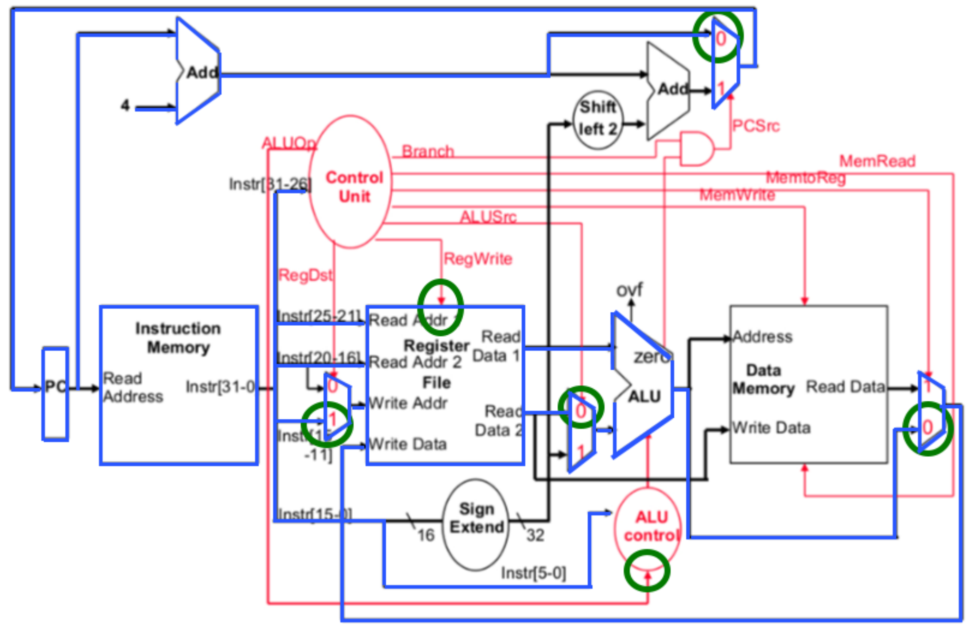
Problem 9.2



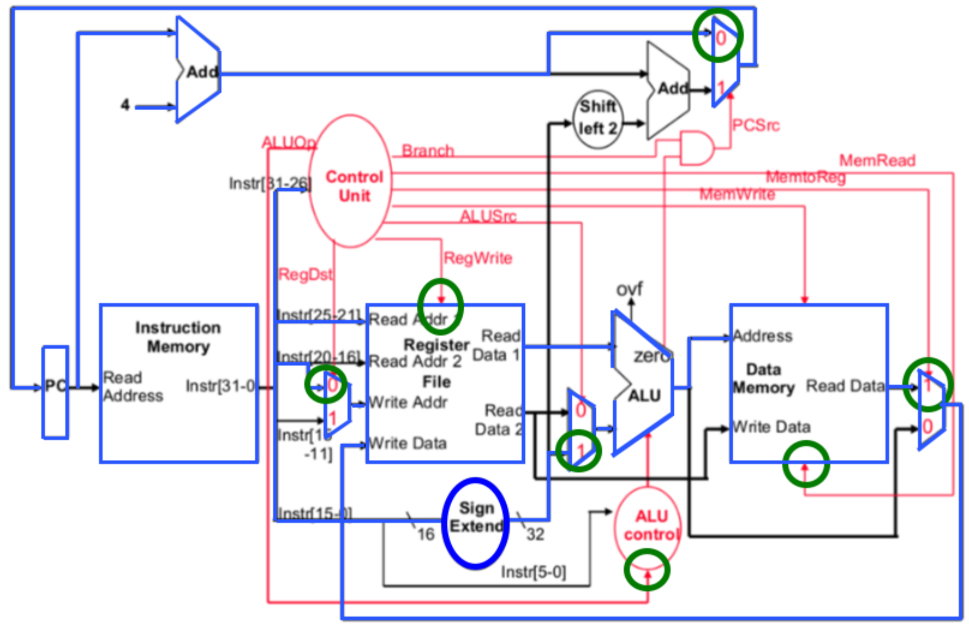
(a) By marking (by e.g., a thicker line or another color) the active lines and circling the active selectors in the figure above, show the single-cycle datapath for the add $s0 $s1 $s2 and lw $s3 16 ($s2)instructions. Use two copies of the figure for the two instructions. Then with your findings write down the values of the control lines for the instructions into the table below.

Solution:

add $s0 $s1 $s2:



lw $s3 16 ($s2):



Instruction	RegDst	ALUSrc	Memto Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp
add	1	0	0	1	0	0	0	10
lw	0	1	1	1	1	0	0	00

(b) When does the ALU need to add its inputs? Give and explain detailed examples from at least two different instruction classes.

R-type instruction:
⟶ Add/sub operations:
These instructions are executed when the ALUOp is 10. When these operations happen, the operands in the ALU go through addition, but for different usages. For add it will only do the sum of two values and put the result into the destination register.

I-type instruction:
⟶ sw $rt, offset($rs):
The ALU adds the value read from the register with the offset, which is in the lower 16 bits of the instruction. The sum calculated by the ALU is used as the address for the data memory which is read and written into register $rt, whose address is given in the instruction.