Computer Architecture (Lab). Week 13

Muhammad, Munir, Vladislav, Alena, Hamza, Manuel

Innopolis University

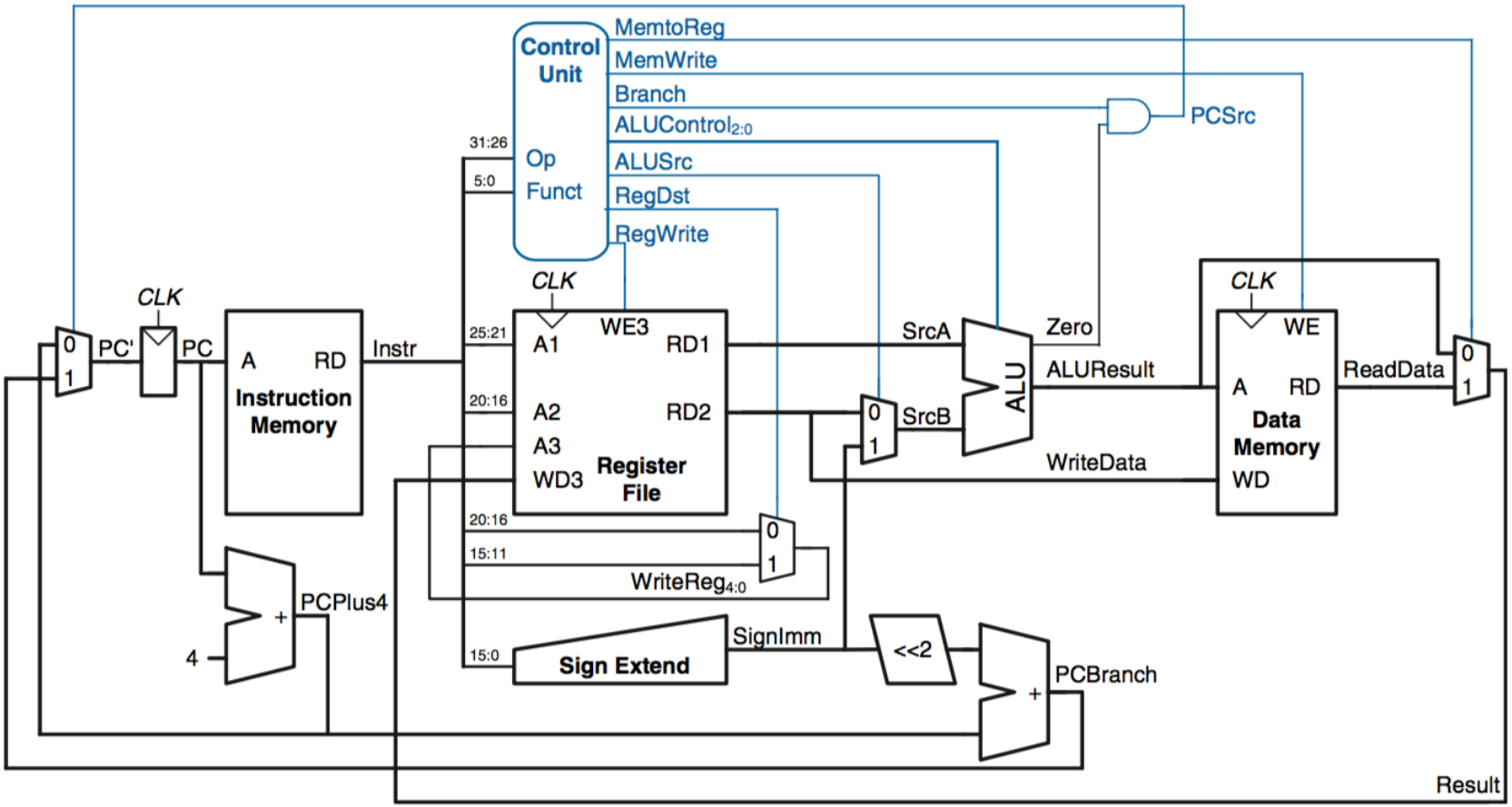
m.fahim@innopolis.ru
m.makhmutov@innopolis.ru
v.ostankovich@innopolis.ru
a.yuryeva@innopolis.ru
h.salem@innopolis.university
m.rodriguez.osuna@innopolis.university

November 26, 2020

Topic of the Lab

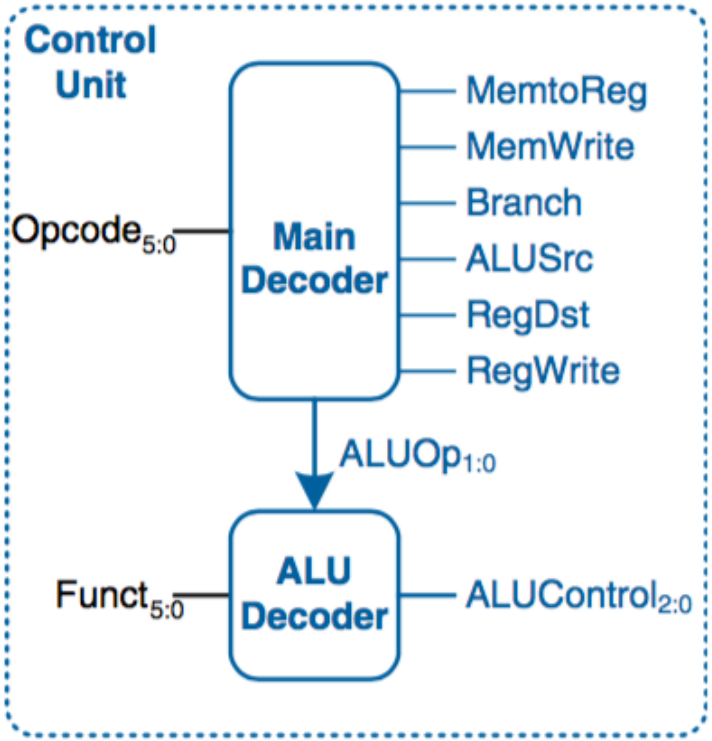
- Understanding control signals in a single-cycle processor
- Implementing control unit

Control Signals



Overview of a single-cycle processor architecture. Can you explain the purpose of these blocks? *Image courtesy of Harris & Harris*

Control Unit



Control unit decodes instructions and produces control signals.

Image courtesy of Harris & Harris

Control Signals (1/2)

Signal	Description
MemtoReg	Set when reading to register from memory
RegDst	Set when instruction uses $rd
ALUSrc	Set if SrcB input is chosen from immediate field
MemWrite	Set when the instruction writes to memory
Branch	Set if the instruction can rewrite the value of PC
RegWrite	Set if the register value is to be updated
ALUOp	Determines what ALU should do

Control Signals (2/2)

Part 1

Instruction	Opcode	RegWrite	RegDst	ALUSrc
R-type	0x00	1	1	0
lw	0x23	1	0	1
sw	0x2B	0	X	1
beq	0x04	0	X	0

Part 2

Instruction	Branch	MemWrite	MemtoReg	ALUOp	ALUControl
R-type	0	0	0	10	?
lw	0	0	1	00	?
sw	0	1	X	00	?
beq	1	0	X	01	?

ALUOp Description

ALUOp is fed to the ALU and helps it to determine current operation

ALUOp	Meaning
00	add
01	subtract
10	look at funct field
11	prohibited

ALU Decoder

ALUOp together with funct determine the ALU operation

ALUOp	Funct	ALUControl
00	X	010 (Add)
X1	X	110 (Subtract)
1X	0x20	010 (Add)
1X	0x22	110 (Subtract)
1X	0x24	000 (And)
1X	0x25	001 (Or)
1X	0x2A	111 (Set less than)

Exercise

1. Write a module in Verilog that implements main decoder. It accepts Opcode as an input and has 7 outputs (see slide 4).
2. Write a module that implements ALU Decoder. It accepts ALUOp and Funct as input and outputs ALUControl signal.
3. Write a module that combines main decoder and ALU decoder in Control Unit.
4. Run the testbench and compare output.

Acknowledgements

- This lab was created and maintained by Vitaly Romanov, Aidar Gabdullin, Munir Makhmutov, Ruzilya Mirgalimova, Muhammad Fahim, Vladislav Ostankovich, Alena Yuryeva, Hamza Salem, Manuel Rodriguez