Assignment #5

Single Cycle CPU

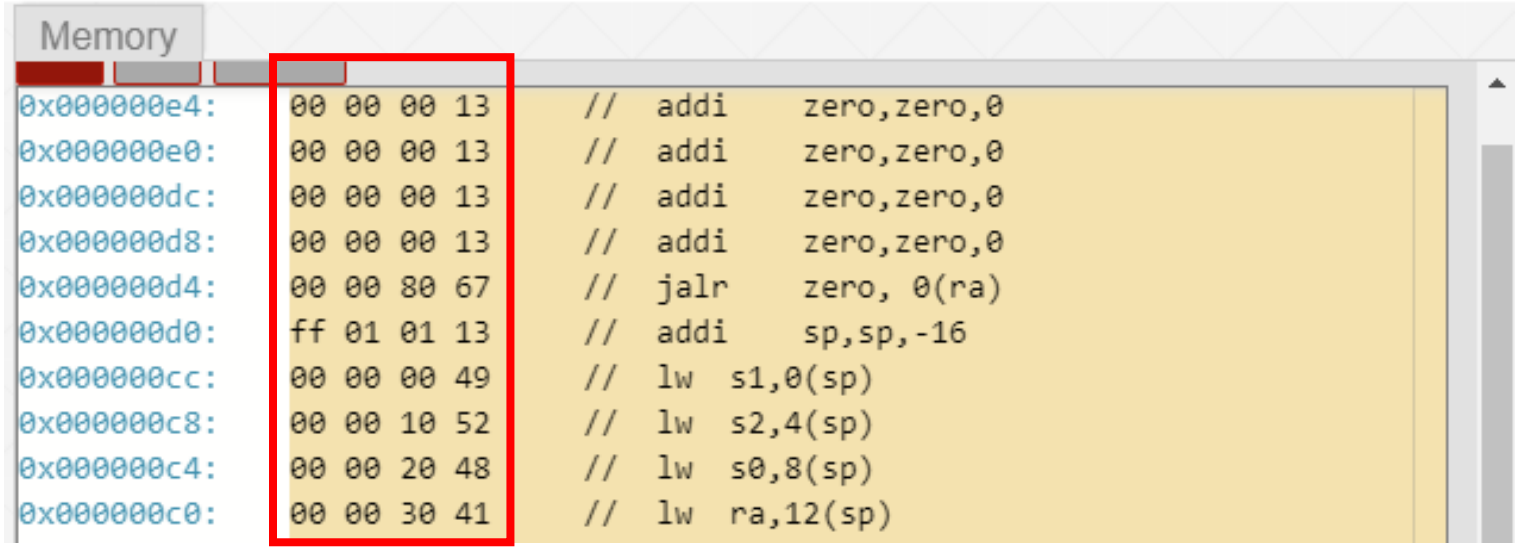
Hanhwi Jang <hanhwi@ajou.ac.kr>

Requirement

- Implement a single-cycle CPU
  - Refer P&H textbook and our lecture slides for design details.
  - ISA: RISC-V 32-bit instructions
    - You do not need to implement the whole instructions
    - Just implement ones required for the following test workload.
  - Test workload (You already implemented this?!)
    - Bubble sort algorithm
    - Put the instructions in Vr_inst_mem.v
    - Before use your instructions, make sure that the instructions refer data addresses correctly.
- Submit your CPU module, testbench, and a report to explain your design decision and evaluation results.

Install “instructions” in the instruction memory (1)

- Use the simulator to get the instructions’ binary representation.



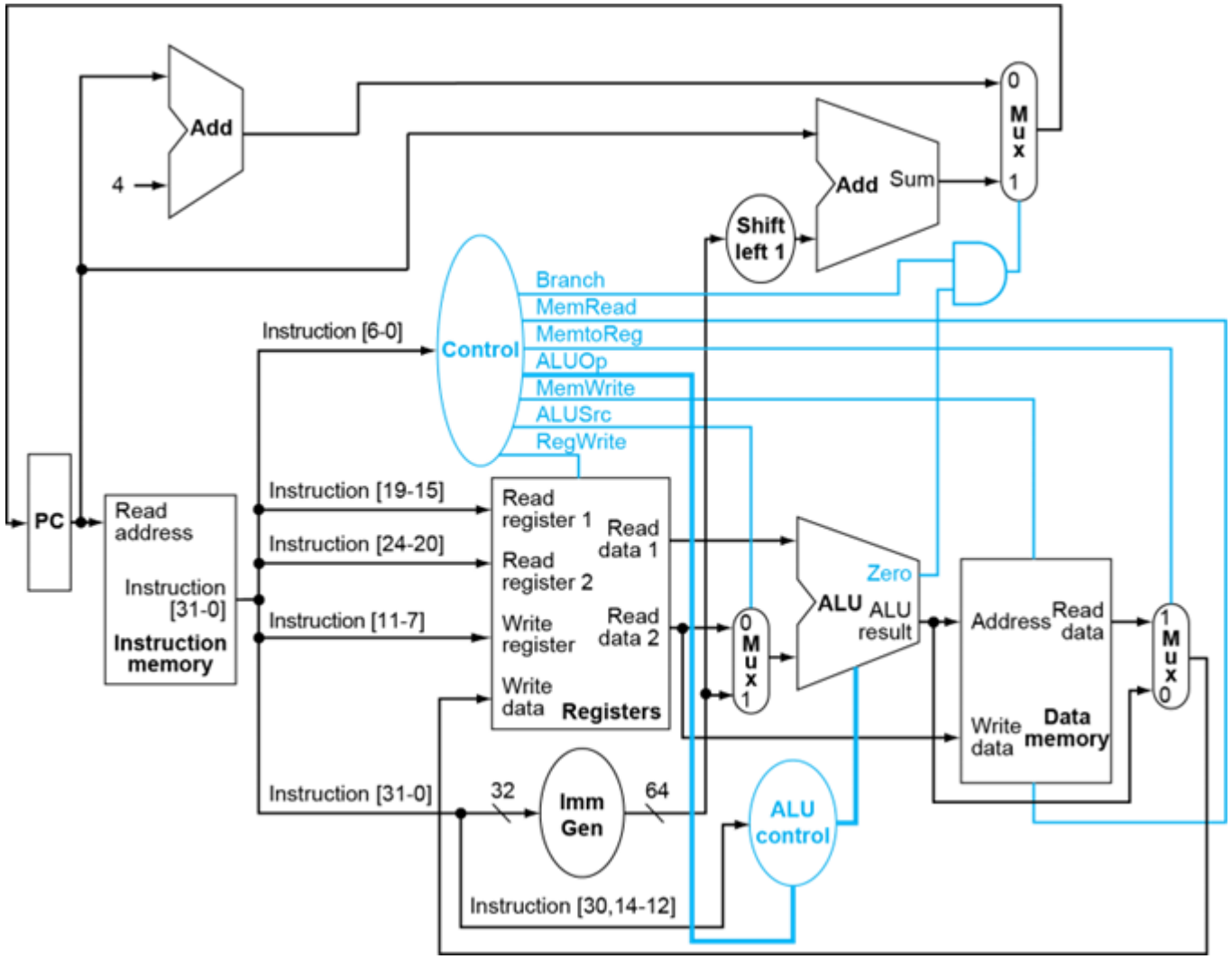
- Put the instructions in the instruction memory (Vr_inst_mem.v)

Install “instructions” in the instruction memory (2)

- Modify the encoded data addresses according to the actual data locations on your implementation.
  - To identify the data location, check “Vr_data_mem.v”
- Example) addi x1, x0, DATA

immediate	rs1	funct3	rd	opcode		**Binary Representation**
12 bits	5 bits	3 bits	5 bits	7 bits		
0000 0000 0000	00000	000	00001	0010011	→	**00 0**0 00 83
DATA_ADDR = 0	**x0**	**addi**	**x1**	**addi**		
0000 0001 0000	00000	000	00001	0010011	→	**01 0**0 00 83
DATA_ADDR = 16	**x0**	**addi**	**x1**	**addi**		

Single-cycle CPU Design



“Magic” Memory and Register

- Instruction/data memory and register file are provided
  - Vr_data_mem.v
  - Vr_inst_mem.v
  - Vr_register_file.v

- Combinational Read
  - The output of the read data port is a combinational function of the register file contents and the corresponding read select port

- Synchronous write
  - The selected register is updated on the positive-edge clock transition when write enable is asserted

Submission Deadline:
- 06/15