CSED311 Lab3: Multi-Cycle CPU

Hyunuk Cho

gusdnr9779@postech.ac.kr

Contact the TAs at csed311-ta@postech.ac.kr

Contents

- Objectives
- Multi-cycle CPU
- Assignment
- Announcements

Objectives

- Understand why a multi-cycle CPU is better than single-cycle implementation
- Design and implement a multi-cycle CPU, which has its own datapath and control unit

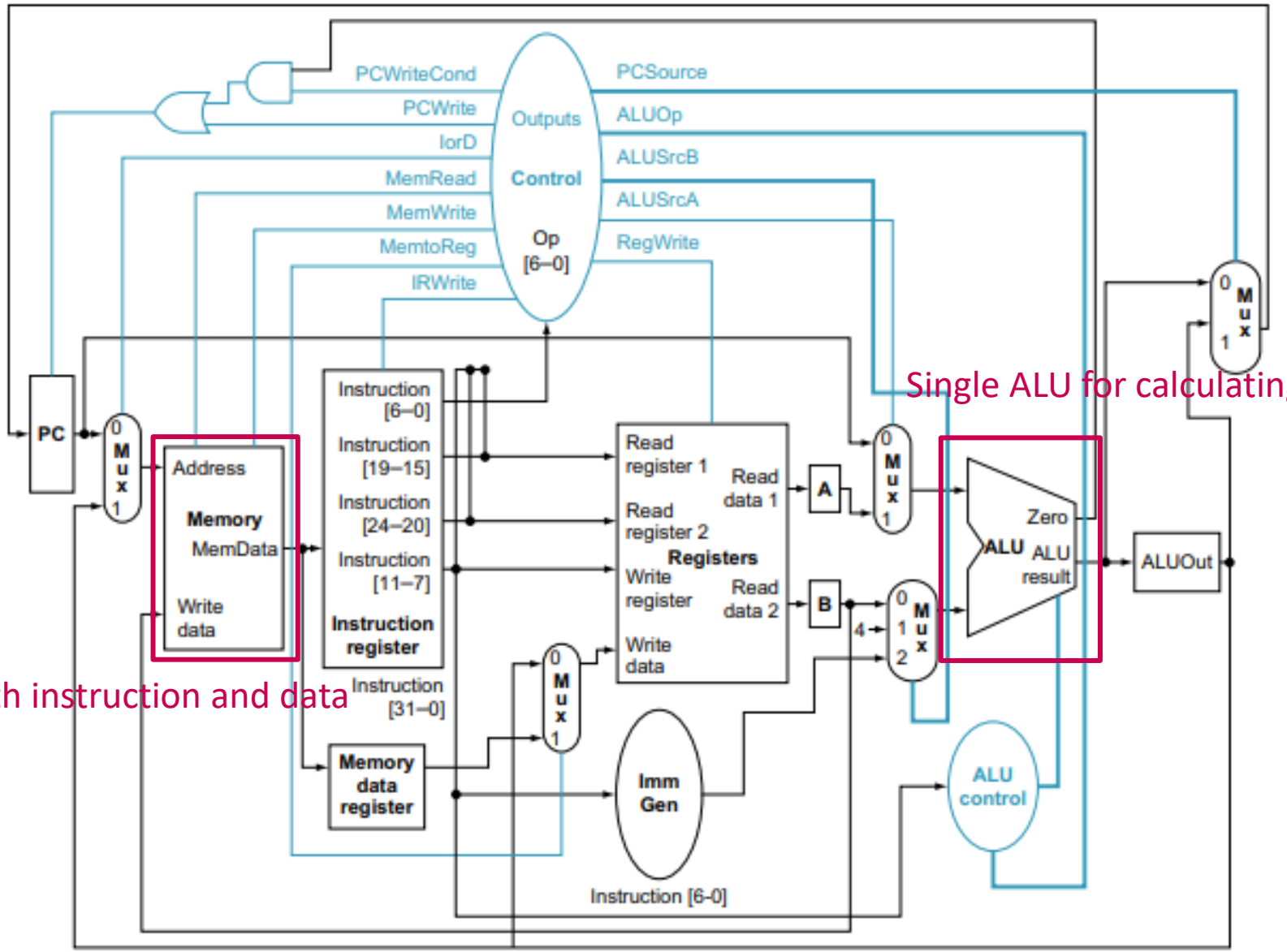
Why Multi-Cycle CPU?

- Problem of the single-cycle CPU: underutilization of resources (ALU, memory, register file, etc.)
- **Solution**: use higher clock frequency and allocate a different number of cycles for each instruction type

Memory units (read or write): 200 ps
ALU (add op) : 100 ps
Register file (read or write): 50 ps
Other combinational logic: 0 ps

Steps	IF	ID	EX	MEM	WB	Delay
Resources	mem	RF	ALU	mem	RF	
R-type	200	50	100		50	400
I-type	200	50	100		50	400
LD	200	50	100	200	50	600
SD	200	50	100	200		550
Bxx	200	50	100			350
JAL	200		100		50	350
JALR	200	50	100		50	400

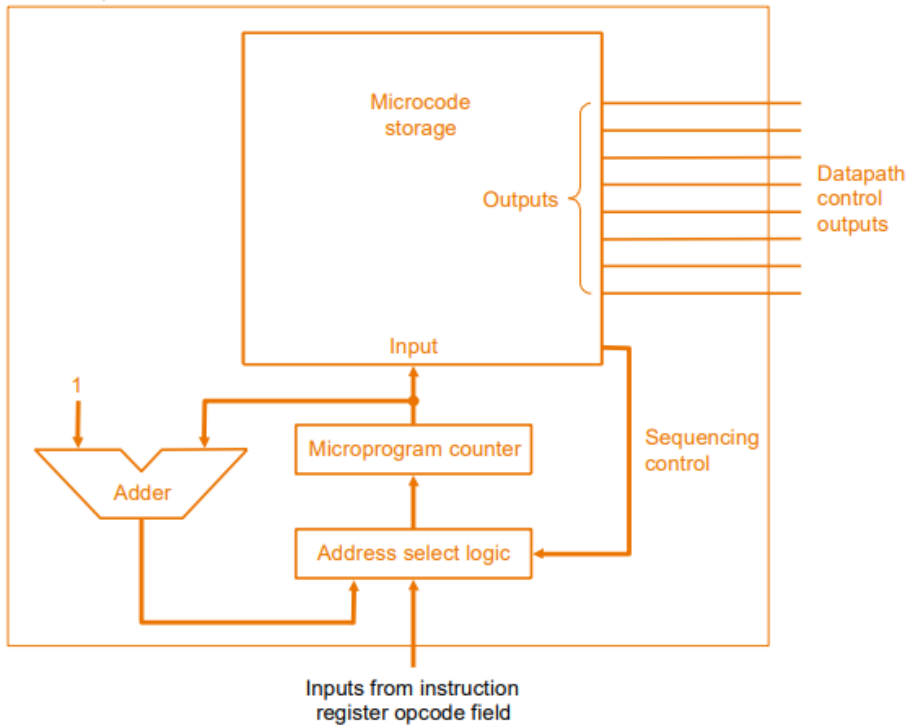
Multi-Cycle CPU (Datapath with Resource Reuse)

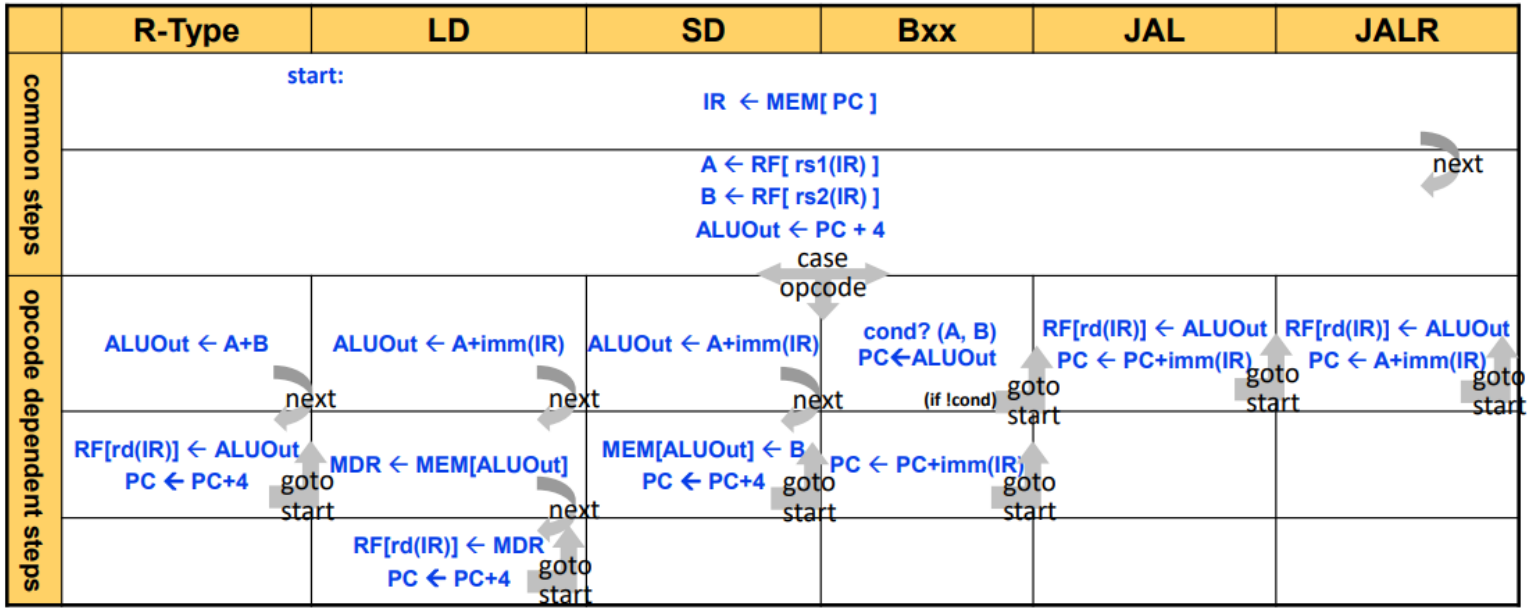


Single ALU for calculating both data and PC

Single memory for both instruction and data

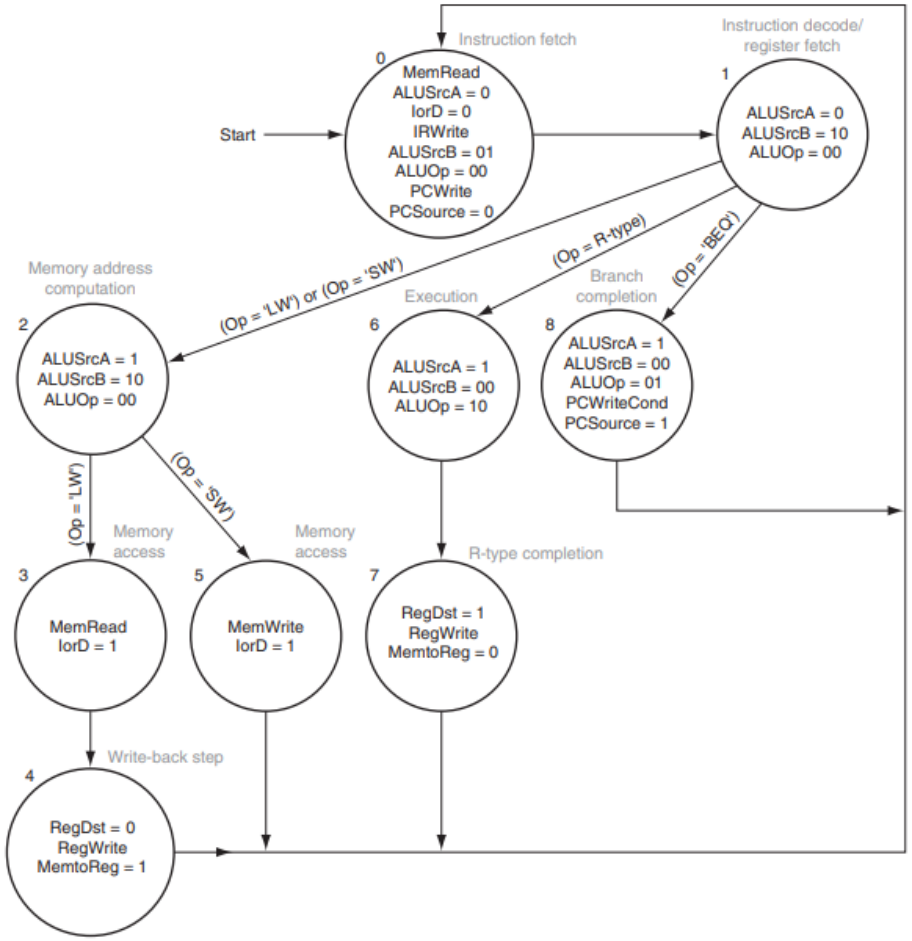
Multi-Cycle CPU (Microcode Controller)





POSTECH

Multi-Cycle CPU (Finite State Machine)



- State transition diagram example in the textbook (appendix C)
- You can use this diagram and the RT sequencing table in the last slide as references for your FSM
- More details for multi-cycle CPU are given in the lecture note and textbook
  - Appendix C can be helpful
- Please read those materials yourself to work on the assignment

Assignment

- Implement a multi-cycle RISC-V CPU (RV32I)
  - Datapath
    - ALU
    - Register file
  - Control unit
    - Microcode controller
    - Generate the control signals used in the datapath
  - Please keep the components modularized. It's okay to make minor changes (e.g., adding or renaming ports)
- Skeleton code updated
  - cpu.v, RegisterFile.v, and memory.v are updated
  - You can take other modules (e.g., ALU) from your single-cycle CPU to implement the multi-cycle CPU
  - Do not modify these files: top.v, RegisterFile.v, and memory.v

Assignment (cont'd)

- Implement the same instructions required in the single-cycle CPU

imm[20|10:1|11|19:12]				rd	1101111	JAL
imm[11:0]		rs1	000	rd	1100111	JALR
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
imm[12|10:5]	rs2	rs1	001	imm[4:1|11]	1100011	BNE
imm[12|10:5]	rs2	rs1	100	imm[4:1|11]	1100011	BLT
imm[12|10:5]	rs2	rs1	101	imm[4:1|11]	1100011	BGE
imm[11:0]		rs1	010	rd	0000011	LW
imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	SW
imm[11:0]		rs1	000	rd	0010011	ADDI
imm[11:0]		rs1	100	rd	0010011	XORI
imm[11:0]		rs1	110	rd	0010011	ORI
imm[11:0]		rs1	111	rd	0010011	ANDI
0000000	shamt	rs1	001	rd	0010011	SLLI
0000000	shamt	rs1	101	rd	0010011	SRLI
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND
000000000000		00000	000	00000	1110011	ECALL

Evaluation Criteria (Implementation)

- The score will be calculated based on the final register values (x1-x31) of the Verilog RTL after (unshared) test cases for evaluation are executed (same as single-cycle CPU)
  - You can check right register values with single-cycle CPU Ripes simulation (Ripes doesn't support multi-cycle simulation)
  - For custom test cases, keep the number of instructions smaller than 1024
- Implementation guideline
  - Your control unit should be a well-implemented state machine
    - Each state should generate its control signals
  - All your circuits (datapath + control unit) should be clock-synchronous
  - All storage units (registers, PC, etc.) must be updated only at the clock's positive edges
  - Your code should reuse resources (e.g., use a single ALU for R-type ops and calculating next PC values), and the control units should be implemented accordingly
  - If you don't follow the guideline, you will get penalty
  - Read the lab_guideline.pdf before starting implementation

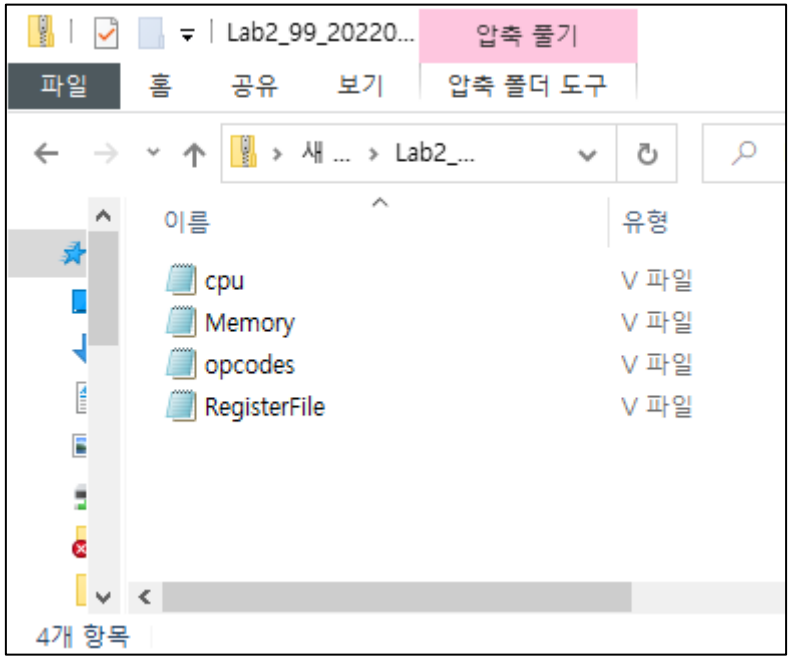
Evaluation Criteria (Report)

- The report should include (1) introduction, (2) design, (3) implementation, (4) discussion, and (5) conclusion sections
- Attach **screenshots of your microcode controller, control unit, register file code** in the report
- Please describe/discuss these aspects in your report:
  - Difference between single-cycle CPU and multi-cycle CPU
  - Why multi-cycle CPU is better?
  - Multi-cycle CPU design and implementation
    - Description of whether each module (RF, memory, PC, control unit, ..) is clock synchronous or asynchronous
    - Microcode controller state design
    - Resource reuse design and implementation
  - Number of clock cycles it took to run basic_ripes, and loop_ripes examples

Submission

- Submit your report and source code on PLMS with filename:
  - Lab3_{TeamID}_{StudentID1}_{StudentID2}.pdf
    - PDF file of your report
  - Lab3_{TeamID}_{StudentID1}_{StudentID2}.zip
    - Zip file of your source code (without top.v)
    - Do not create a folder within the zip file

Zip file contents
(note there is no folder):



- **If you don't follow the guidelines, you will get penalty**

POSTECH

Deadline

- 1st week, 2nd week (optional) submission (Non-control flow instructions only)
  - Deadline: 2022. 4. 5 09:00 a.m., 2022. 4. 12 09:00 a.m.
  - This is not mandatory. The result will not be reflected in the lab score.
  - We will run the TBs for testing non-control flow instructions and let you know the score
- 3rd week final submission (All instructions required in this lab)
  - **Code: 2022. 4. 19 / 09:00 a.m.**
  - **Report: 2022. 4. 19 / 23:59 p.m.**
  - This is the mandatory part that will be reflected in the lab score
  - Evaluation will be done with both non-control flow and control flow instructions

Announcements

- From this lab, we will use Vivado for lab assignments
  - Vivado install and use guide will be uploaded on PLMS in a few days

- About lab 1 scoring
  - Demo score 80% (weights: ALU 30%, vending machine 70%)
    - Number of tests passed
    - Whether there are implementation guideline violations
  - Report score 20%
  - The final score will be re-calculated with the weights and announced again

Questions?