Computer Organization
Lab 3 - Single Cycle CPU

教授:蔡文錦
助教:劉益先、鄭吉呈、黃芷柔、林彧頎

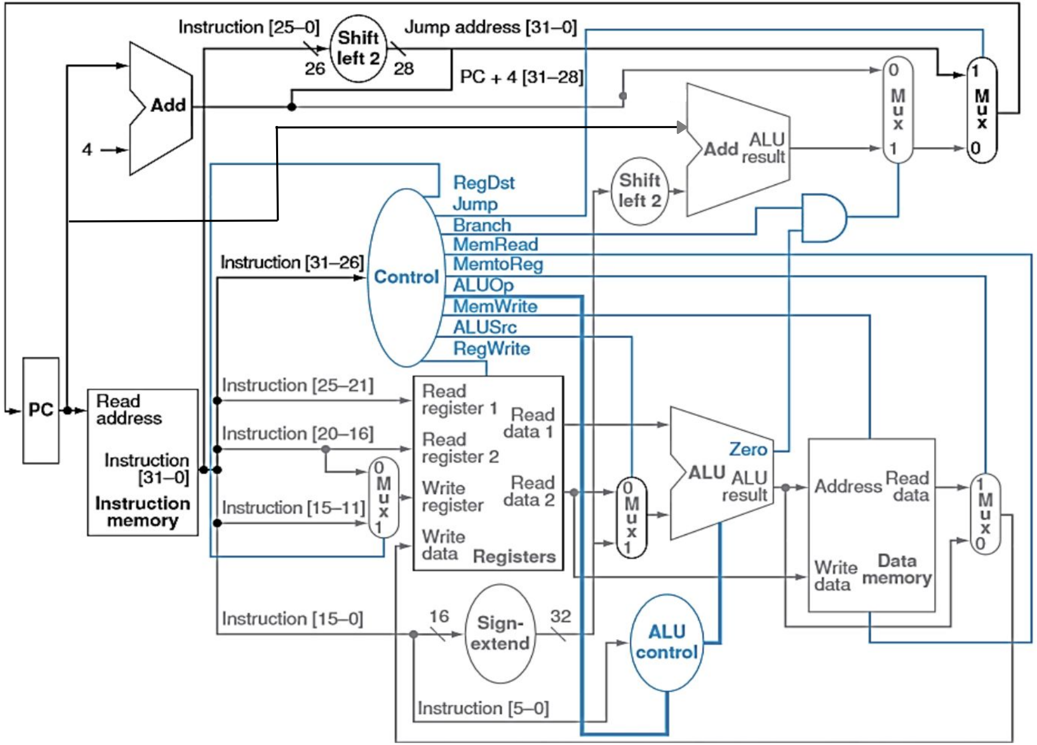
Objectives

In this lab, we are going to implement a **single cycle CPU** with memory unit, which can run **R-type**, **I-type** and **J-type** instructions.

- To realize how to set the control signal in different instruction type.
- To learn how to follow the datapath to form a single cycle CPU

Overview

The following diagram is the datapath of single cycle CPU.



Attached Files

- **TO DO**
  - Adder.v
  - ALU_Ctrl.v
  - ALU.v (You can use the code from Lab 2 or rewrite one)
  - MUX_2to1.v, MUX_3to1.v
  - Decoder.v
  - Sign_Extend.v
  - Simple_Single_CPU.v
  - Shift_Left_Two_32.v

- **DO NOT modify**
  - ProgramCounter.v
  - Reg_File.v,
  - Instr_Memory.v
  - Data_Memory.v

- **For validation - DO NOT modify**
  - testbench.v

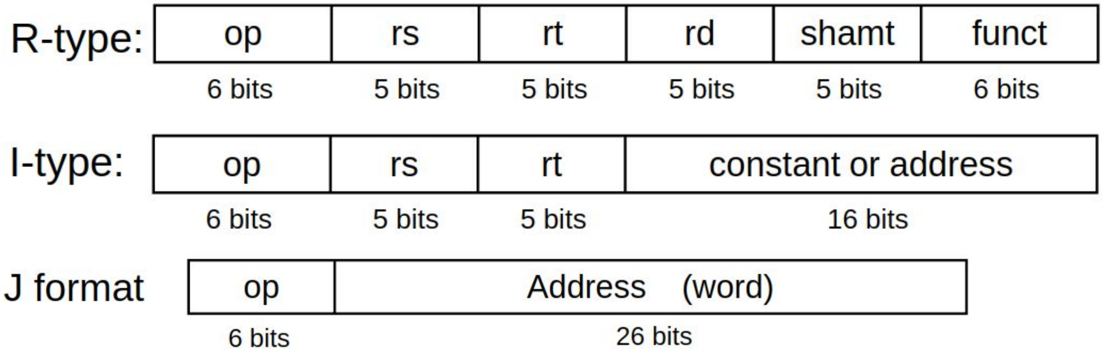
- **Testcase - YOU CAN modify the instructions in it.**
  - *.txt

Instruction Set

You are going to implement these instructions :

- **R-type :** add, sub, AND, OR, NOR, slt, sll, srl, sllv, srlv, jr
- **I-type :** lw, sw, beq, bne, addi
- **J-type :** j, jal

Instruction Format:



R-type:	op	rs	rt	rd	shamt	funct
	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

I-type:	op	rs	rt	constant or address
	6 bits	5 bits	5 bits	16 bits

J format	op	Address (word)
	6 bits	26 bits

Instruction Set (R-type)

Function	Op Field	Function Field
add	6’b000000	6’b100011
sub	6’b000000	6’b100001
AND	6’b000000	6’b100110
OR	6’b000000	6’b100101
NOR	6’b000000	6’b101011
slt	6’b000000	6’b101000
sll	6’b000000	6’b000010
srl	6’b000000	6’b000100
sllv	6’b000000	6’b000110
srlv	6’b000000	6’b001000
jr	6’b000000	6’b001100

You can make your own design for sll, srl, sllv, srlv

Such as

1. adding new modules and data paths

2. extending existing ALU operations

3. or any others ...

Instruction Set (I-type and J-type)

I-type	
Function	**Op Field**
addi	6b’001001
lw	6b’101100
sw	6b’100100
beq	6b’000110
bne	6b’000101

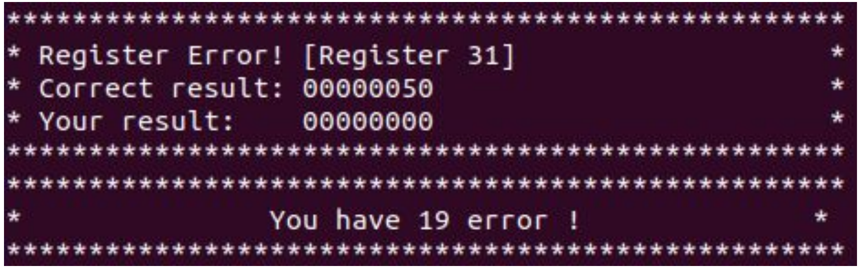
J-type	
Function	**Op Field**
j	6b’000111
jal	6b’000011

Compile & Run

- **Compile**
  - $ iverilog -o lab3 testbench.v

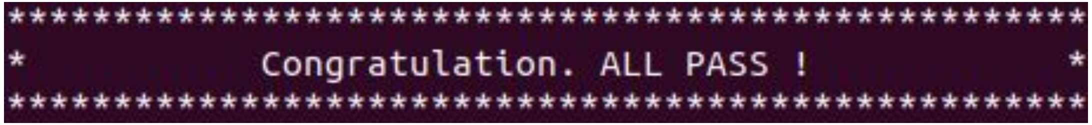
- **Run**
  - $ ./lab3
  - (windows)$ vvp lab3

Wrong results:



```
****************************************************
* Register Error! [Register 31]                    *
* Correct result: 00000050                         *
* Your result:    00000000                         *
****************************************************
****************************************************
*               You have 19 error !               *
****************************************************
```

Correct results:



```
****************************************************
*           Congratulation. ALL PASS !             *
****************************************************
```

Grading Policy

- There are **5 hidden cases with serial several instructions**, and you will get **20** points for each correct testcase, totally 100 points.
- **Any assignment work by fraud will get a zero point！**
- **No late submission！**

Submission

- **Please attach student IDs as comments at the top of each TO DO file.**
- The files you should hand in include:
  - **all *.v files excluding testbench.v**
- Compress all file ***.v** into one zip file **without any extra folder layer**, and **make sure do not add unnecessary files or folders** (like .DS_Store, __MACOSX).
- Name your zip file as **HW3_{studentID}.zip**
  - **e.g.**
    - **HW3_123456789.zip**
      - **{*.v}**
- **Wrong format will have 20% penalty！**
- **Deadline: 8/13 23:55**