Lab3: Single Cycle CPU

1.Goal:

Based on lab2, single cycle add memory unit, implement a complete single cycle CPU which can run R-type, I-type and jump instruction.

2.HW requirement:

Deadline: 5/20 11:59 pm

1. ModelSim or Xinlinx as simulation platform
2. **Group member same as Lab1**. Just hand in one assignment for one group. Please attach your student IDs as comments in each of .v source code, otherwise deduct 20 points.
3. Do not copy, or you will get 0.
4. Top module’s name and IO port reference Lab2
5. Put .v source files and report into a compressed file.The compressed file you upload on E3 must have the form of "student ID.zip ". (Ex. 0216310_0216077.zip or 0216310.zip), or you’ll get 0 point.

REGISTER_BANK [29] representation stack pointer register value initial 128,other 0

Decoder may add control signal

-Branch_o

-Jump_o

-MemRead_o

-MemWrite_o

-MemtoReg_o

3.Requirement description:

lw instruction

memwrite is 0，memread is 1，regwrite is 1

Reg[rt]🡐Mem[rs+imm]

Sw instruction

Memwrite is 1，memread is 0

Mem[rs+imm]🡐Reg[rt]

Branch instruction

branch is 1，and decide branch or not by do AND with the zero signal from ALU

PC=PC+4+ (sign_Imm<<2)

Jump instruction

jump is 1

PC={PC[31:28]，address<<2}

3a. Code: (80%)

Basic instruction:(50%)

Lab 2 instruction + lw、sw、j

R-type

Op[31:26]	Rs[25:21]	Rt[20:16]	Rd[15:11]	Shamt[10:6]	Func[5:0]

I-type

Op[31:26]	Rs[25:21]	Rt[20:16]	Immediate[15:0]

Jump

Op[31:26]	Address[25:0]

instruction	Op[31:26]			
lw	6’b100011	Rs[25:21]	Rt[20:16]	Immediate[15:0]
sw	6'b101011	Rs[25:21]	Rt[20:16]	Immediate[15:0]
jump	6'b000010	Address[25:0]		

Advance set :(30%)

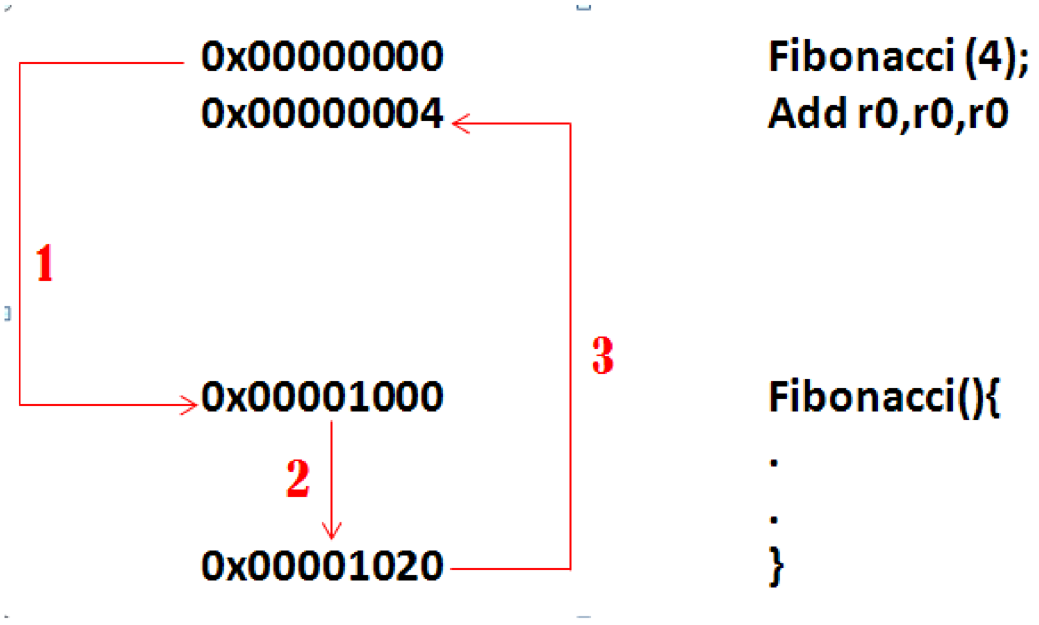
instruction	op	rs	rt	rd	shamt	Func
jal	6’b000011	Address[25:0]				
jr	6’b000000	rs	0	0	0	6’b001000

Jal: jump and link
In MIPS，31th register is used to save return address for function call
Reg[31] save PC+4 and perform jump

Reg[31]=PC+4
PC={PC[31:28],address[25:0]<<2}

Jr: jump to the address in the register rs
PC=reg[rs];
e.g.：In MIPS，return could be used by
jr r31
to jump to return address from **JAL**

Sample: when CPU executes function call,



if you want to execute recursive function, you must use the stack point (REGISTER_BANK [29])。First, store the register to memory and load back after function call has been finished.

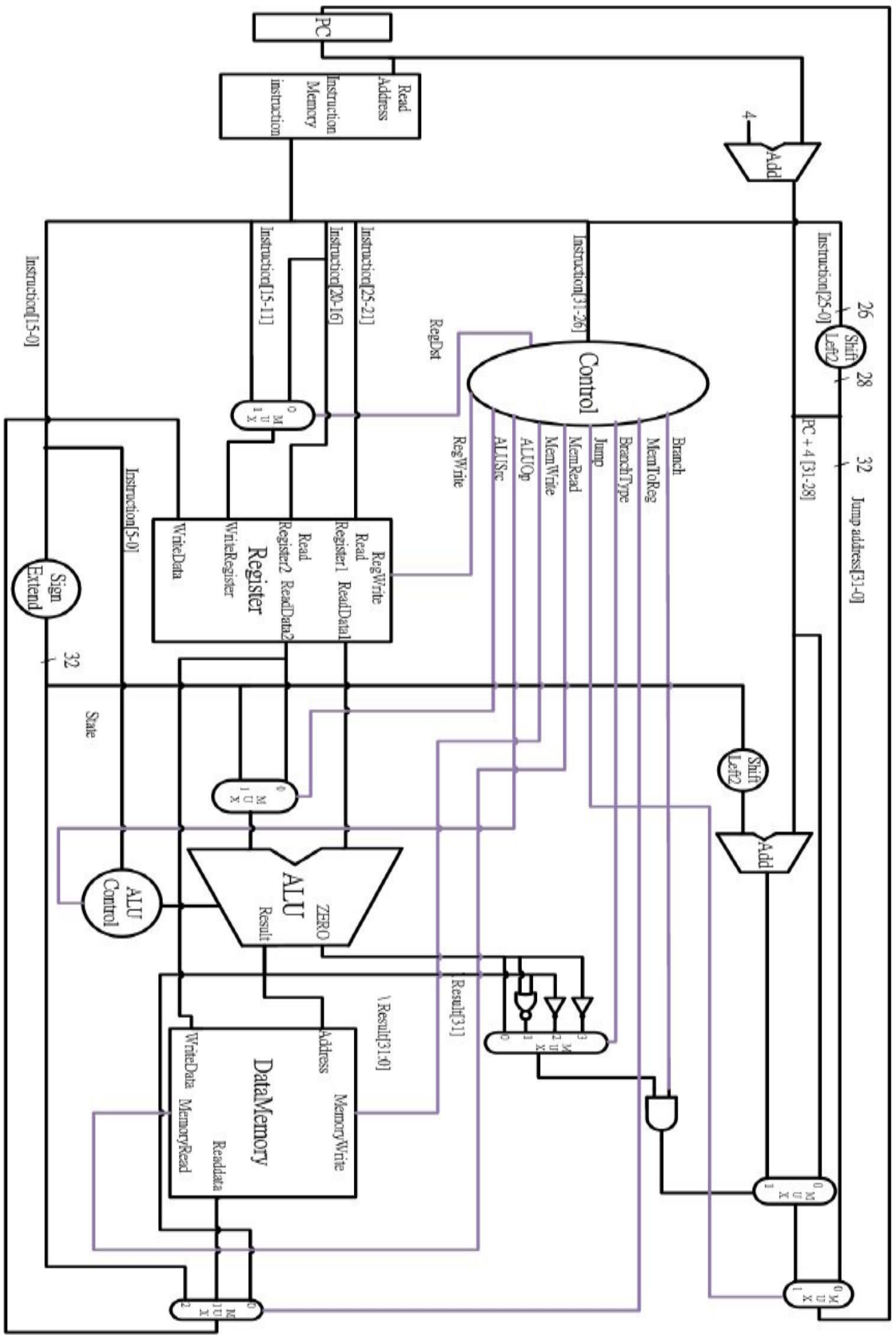
Second testbench CO_P3_test_data2.txt is Fibonacci funcion，if it finished, r2 represent the final answer. Please refer to test2.txt

3b. Report (20%)
If you are two-member teams,you need to explain the division of this work.

4. Reference architecture

This lab might be used extra signal to control. Please draw the architecture you designed on your report.

You may need to modify the architecture.



5. Hand in your assignment:

Please upload the assignment to the E3.

Put all of .v source files and report into same compressed file.(Use your student ID to be the name of your compressed file)

6. Q&A

If you have any question, just send email to TAs.

7. Late Submission

Late submission: Score*0.8 before 5/27. After 5/27,you will get 0 point