Project Report

1. Objective

The objective of this project is to design a whole CPU module, which is a single-cycle processor which can execute MIPS instructions with Verilog. This CPU supports the following instructions:

"lw", "sw", "add", "sub", "addu", "subu", "addi", "addiu", "and", "or", "nor", "xor", "andi", "ori", "beq", "bne", "slt", "j", "jr", jal.

2. I/O

This ALU takes **4** inputs:
Two 1-bit inputs: *start, clock*

"start" is to start the module as it becomes 1;
"clock" is to trigger the execution of data path at every positive edge.

Two 32-bit input: *i_datain, d_datain*

"i_datain" is the machine code of instruction reading from instruction memory according to PC;

"d_datain" is the data reading from data memory when loading the data to register file;

And it generates **1** 32-bit outputs: *d_dataout*

"d_dataout" is the data which is going to be stored into data memory when storing data.

```
module CPU(

    input wire clock,
    input wire start,
    input [31:0] i_datain,  // Instruction as input
    input wire [31:0] d_datain, // The data should be loaded from data memory as input
    output wire [31:0] d_dataout    // The data should be stored into data memory.
    );
```

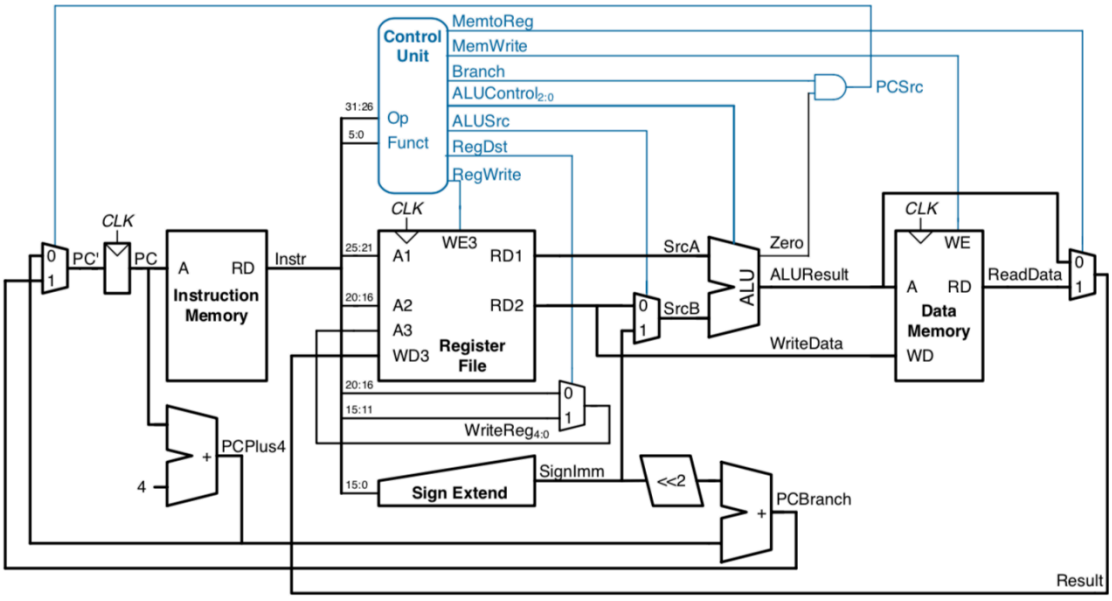
And there are some variables in should be noticed:

```
reg [31:0]gr[31:0]; // 32 registers
reg [15:0]pc = 32'h0000_0000; // address of pc

reg [31:0]reg_A;    // SrcA
reg [31:0]reg_B;    // SrcB
reg [31:0]reg_C;    // ALUResult
reg [31:0]reg_C1;   // Result
reg [31:0]write_data;   // WD for data memory
reg mem_write;  // WE for data memeory

reg [5:0]opcode;
reg [5:0]functcode;
```

The name in the comment is corresponding to the block diagram given by the project introduction (as shown below).



3. Logic

The logic of this module is playing as “Control Unit”:

First, it reads the instruction code and find opcode and function code, so that it can identify which instruction it is.

```
//********************* Control Unit ***********************//

opcode = i_datain[31:26];
functcode = i_datain[5:0]; // For R-type use
```

After identifying the instruction, it can correctly control every multiplexer in the CPU. This program mainly deals with these control parts, as shown in the block diagram above: MemtoReg, MemWrite, Branch, ALUControl, ALUSrc, RegDst, RegWrite.

So the code is divided into several parts to show this logic. Each part deals with all instruction cases and they work together to integrate as a strong CPU. Each part was specified by a line like this:

```
// *************** Control Unit -- ALUSrc ****************//
```

Then, make sure the whole data path is completed so that it can successfully transfer the data. And this work is done inside every control part.

The comments in the code is very detailed and clear. You can find detailed logic by reading the comments of code.

4. Test

The file “test_CPU” gives a test to every supported instruction.

There are 32 registers, but only 4 are used: gr1, gr2, gr3 are used for testing arithmetical and logical instructions, and gr31, which represents $ra is used for testing instructions “jr” and “jal”.

For every test, the comments in the code provides the clear instruction in MIPS form and one of expected output. Please read detailed comments in the code.

```
/*Test:*/

#5 start = 1; // posedge, print the second line

#5
// lw test
d_datain <= 32'h0000_00ab;
i_datain <= {6'b100011, `gr0, `gr1, 16'h0001};  // lw gr1 1(gr0); gr1 = ab

#period d_datain <= 32'h0000_3c00;
i_datain <= {6'b100011, `gr0, `gr2, 16'h0002};  //lw gr2 2(gr0); gr2 = 3c00

// add test
#period i_datain <= {6'b000000, `gr1, `gr2, `gr3, 5'b00000, 6'b100000}; //add gr3 gr1 gr2; gr3 = 3cab

// sub test
#period i_datain <= {6'b000000, `gr3, `gr2, `gr3, 5'b00000, 6'b100010}; //sub gr3 gr3 gr2; gr3 = ab
```

5. How to Run

To run the program, you need to write this in command line under the directory including module files:

1. Compile:

```
iverilog –o test CPU.v test_CPU.v
```

2. Run:

```
./test
```

3.Result:

```
yihongpengdeMacBook-Pro:project4 yihongpeng$ iverilog -o test CPU.v test_CPU.v
yihongpengdeMacBook-Pro:project4 yihongpeng$ ./test
pc  :        instruction            : reg_A  : reg_B  : reg_C  : reg_C1 :d_datain:d_dataout:  gr0   :  gr1   :  gr2   :  gr3   :gr31($ra):mem_write
0000:00000000000000000000000000000000:xxxxxxxx:xxxxxxxx:xxxxxxxx:xxxxxxxx:00000000:xxxxxxxx :00000000:xxxxxxxx:xxxxxxxx:xxxxxxxx:xxxxxxxx : x
0004:00000000000000000000000000000000:00000000:00000000:xxxxxxxx:xxxxxxxx:00000000:00000000 :00000000:xxxxxxxx:xxxxxxxx:xxxxxxxx:xxxxxxxx : 0
0004:10001100000000010000000000000001:00000000:00000000:xxxxxxxx:xxxxxxxx:000000ab:00000000 :00000000:xxxxxxxx:xxxxxxxx:xxxxxxxx:xxxxxxxx : 0
0008:10001100000000010000000000000001:00000000:00000001:00000001:000000ab:000000ab:xxxxxxxx :00000000:000000ab:xxxxxxxx:xxxxxxxx:xxxxxxxx : 0
0008:10001100000000010000000000000010:00000000:00000001:00000001:000000ab:00003c00:xxxxxxxx :00000000:000000ab:xxxxxxxx:xxxxxxxx:xxxxxxxx : 0
000c:10001100000000010000000000000010:00000000:00000002:00000002:00003c00:00003c00:xxxxxxxx :00000000:000000ab:00003c00:xxxxxxxx:xxxxxxxx : 0
000c:00000000001000100001100000100000:00000000:00000002:00000002:00003c00:00003c00:xxxxxxxx :00000000:000000ab:00003c00:xxxxxxxx:xxxxxxxx : 0
0010:00000000001000100001100000100000:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0010:00000000011000100001100000100010:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0014:00000000011000100001100000100010:00003cab:00003c00:000000ab:000000ab:00003c00:00003c00 :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
0014:00000000001000100001100000100000:00003cab:00003c00:000000ab:000000ab:00003c00:00003c00 :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
0018:00000000001000100001100000100000:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0018:00000000011000100001100000100010:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
001c:00000000011000100001100000100010:00003cab:00003c00:000000ab:000000ab:00003c00:00003c00 :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
001c:10101100000000010000000000000010:00003cab:00003c00:000000ab:000000ab:00003c00:00003c00 :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
0020:10101100000000010000000000000010:00000000:00000002:00000002:00000002:00003c00:000000ab :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 1
0020:00100000001000110011111000010001:00000000:00000002:00000002:00000002:00003c00:000000ab :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 1
0024:00100000001000110011111000010001:000000ab:00003c11:00003cbc:00003cbc:00003c00:000000ab :00000000:000000ab:00003c00:00003cbc:xxxxxxxx : 0
0024:00100000001000110011111000000000:000000ab:00003c11:00003cbc:00003cbc:00003c00:000000ab :00000000:000000ab:00003c00:00003cbc:xxxxxxxx : 0
0028:00100000001000110011111000000000:000000ab:00003c00:00003cab:00003cab:00003c00:00003cbc :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0028:00000000010000110001100000100100:000000ab:00003c00:00003cab:00003cab:00003c00:00003cbc :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
002c:00000000010000110001100000100100:00003c00:00003cab:00003c00:00003c00:00003c00:00003cab :00000000:000000ab:00003c00:00003c00:xxxxxxxx : 0
002c:00000000001000110001100000100101:00003c00:00003cab:00003c00:00003c00:00003c00:00003cab :00000000:000000ab:00003c00:00003c00:xxxxxxxx : 0
0030:00000000001000110001100000100101:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0030:00000000000000000000110000100111:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0034:00000000000000000000110000100111:00000000:00000000:ffffffff:ffffffff:00003c00:00000000 :00000000:000000ab:00003c00:ffffffff:xxxxxxxx : 0
0034:00000000001000100001100000100110:00000000:00000000:ffffffff:ffffffff:00003c00:00000000 :00000000:000000ab:00003c00:ffffffff:xxxxxxxx : 0
0038:00000000001000100001100000100110:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0038:00110000001000110011111001010111:000000ab:00003c00:00003cab:00003cab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
003c:00110000001000110011111001010111:000000ab:00003cab:000000ab:000000ab:00003c00:00003cab :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
003c:00110100010000110000000010101011:000000ab:00003cab:000000ab:000000ab:00003c00:00003cab :00000000:000000ab:00003c00:000000ab:xxxxxxxx : 0
0040:00110100010000110000000010101011:00003c00:000000ab:00003cab:00003cab:00003c00:000000ab :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0040:00010000001000100000000000000010:00003c00:000000ab:00003cab:00003cab:00003c00:000000ab :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0044:00010000001000100000000000000010:000000ab:00003c00:ffffc4ab:ffffc4ab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0044:00010000001000010000000000000010:000000ab:00003c00:ffffc4ab:ffffc4ab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0050:00010000001000010000000000000010:000000ab:000000ab:00000000:00000000:00003c00:000000ab :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0050:00010100001000100000000000000010:000000ab:000000ab:00000000:00000000:00003c00:000000ab :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
005c:00010100001000100000000000000010:000000ab:00003c00:ffffc4ab:ffffc4ab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
005c:00000000001000100001100000101010:000000ab:00003c00:ffffc4ab:ffffc4ab:00003c00:00003c00 :00000000:000000ab:00003c00:00003cab:xxxxxxxx : 0
0060:00000000001000100001100000101010:000000ab:00003c00:00000001:00000001:00003c00:00003c00 :00000000:000000ab:00003c00:00000001:xxxxxxxx : 0
0060:00001000000000000000000100000000:000000ab:00003c00:00000001:00000001:00003c00:00003c00 :00000000:000000ab:00003c00:00000001:xxxxxxxx : 0
0400:00001000000000000000000100000000:00000000:00003c00:00000001:00000001:00003c00:00000000 :00000000:000000ab:00003c00:00000001:xxxxxxxx : 0
0400:00001100000000000000000100000000:00000000:00003c00:00000001:00000001:00003c00:00000000 :00000000:000000ab:00003c00:00000001:xxxxxxxx : 0
0800:00001100000000000000000100000000:00000000:00003c00:00000001:00000001:00003c00:00000000 :00000000:000000ab:00003c00:00000001:00000400 : 0
0800:00000011111000000000000000001000:00000000:00003c00:00000001:00000001:00003c00:00000000 :00000000:000000ab:00003c00:00000001:00000400 : 0
0400:00000011111000000000000000001000:00000400:00000000:00000001:00000001:00003c00:00000000 :00000000:000000ab:00003c00:00000001:00000400 : 0
yihongpengdeMacBook-Pro:project4 yihongpeng$
```