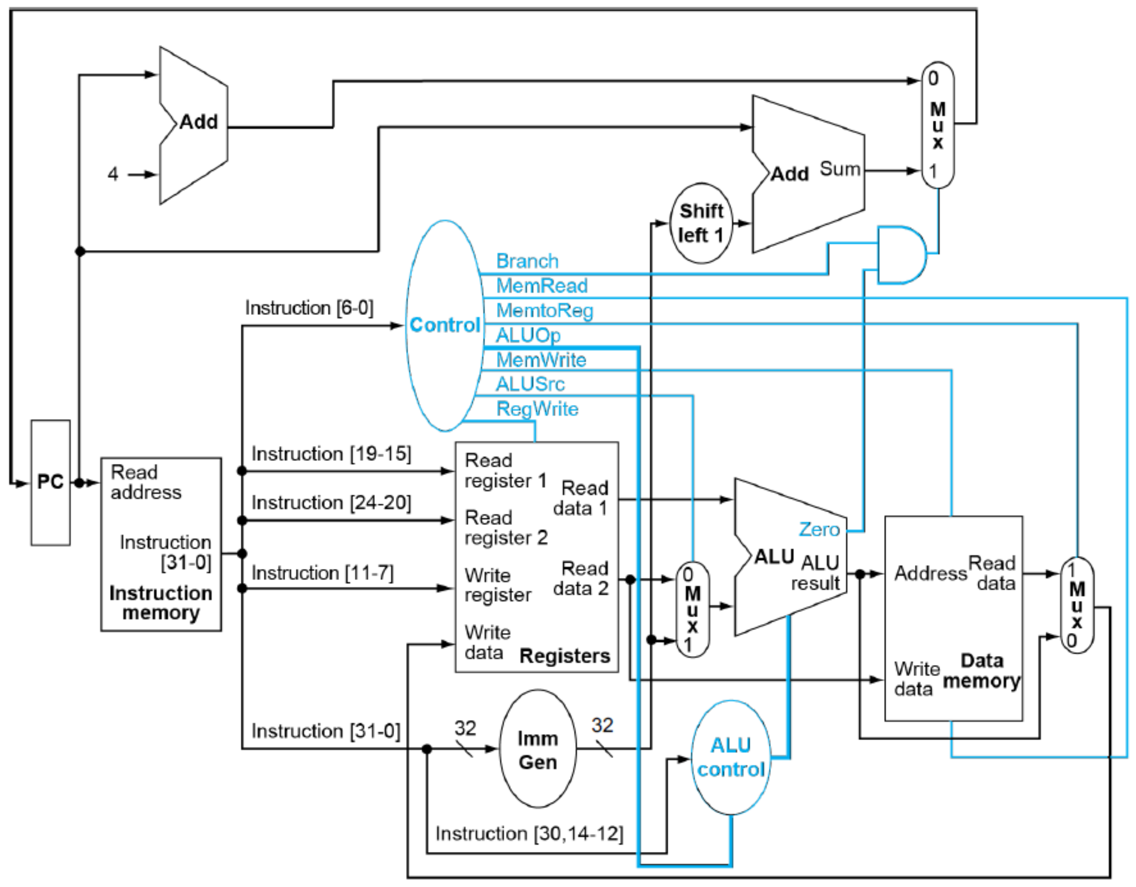
# VE370 Lab 3 Report

Lan Wang 519370910084

## Description

The modeling of the processor basically follows the graph below.



To support `addi`:

```
ALUOp=2'b11;

ALUControl=4'b0010;
```

To support `bne`:

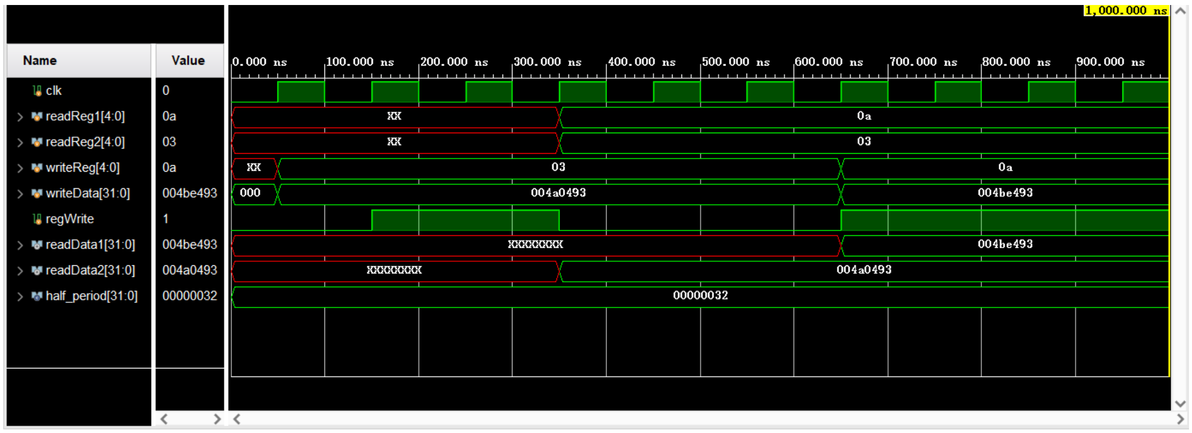
```
ALUOp=2'b01;

ALUControl=4'b0111; // so that it can differ from beq
```

Other control signals are same to the lecture slides.

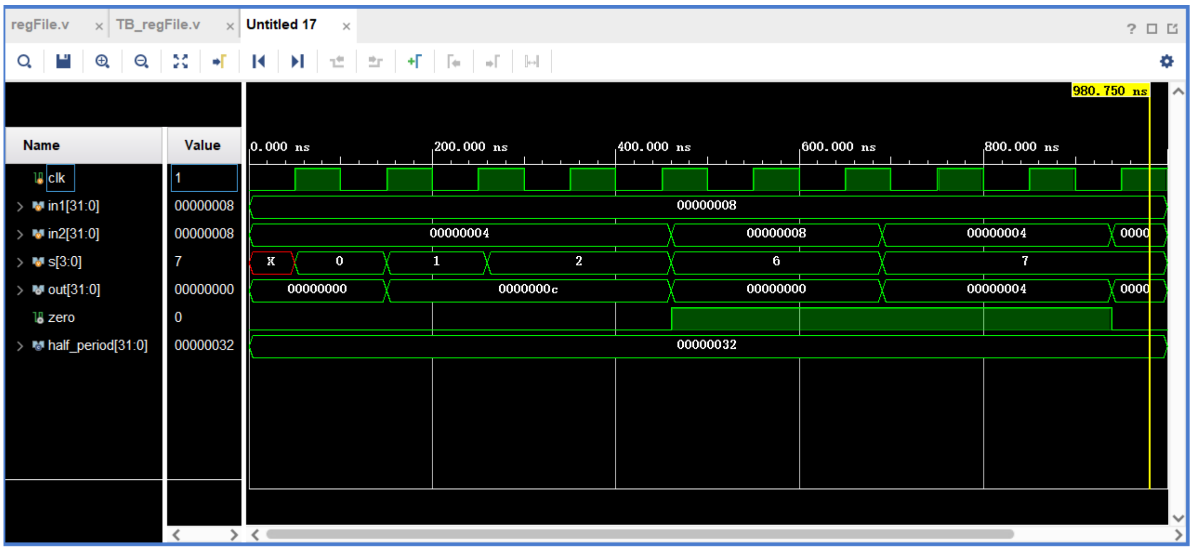
| Inst. | ALUSrc | ALUOp | MemWrite | MemRea | Branch | MemtoReg | RegWrite |
|---|---|---|---|---|---|---|---|
| add | 0 | 10 | 0 | 0 | 0 | 0 | 1 |
| addi | 1 | 11 | 0 | 0 | 0 | 0 | 1 |
| lw | 1 | 00 | 0 | 1 | 0 | 1 | 1 |
| sw | 1 | 00 | 1 | 0 | 0 | 0 | 0 |
| beq | 0 | 01 | 0 | 0 | 1 | 0 | 0 |
| bne | 0 | 01 | 0 | 0 | 1 | 0 | 0 |

## Simulations

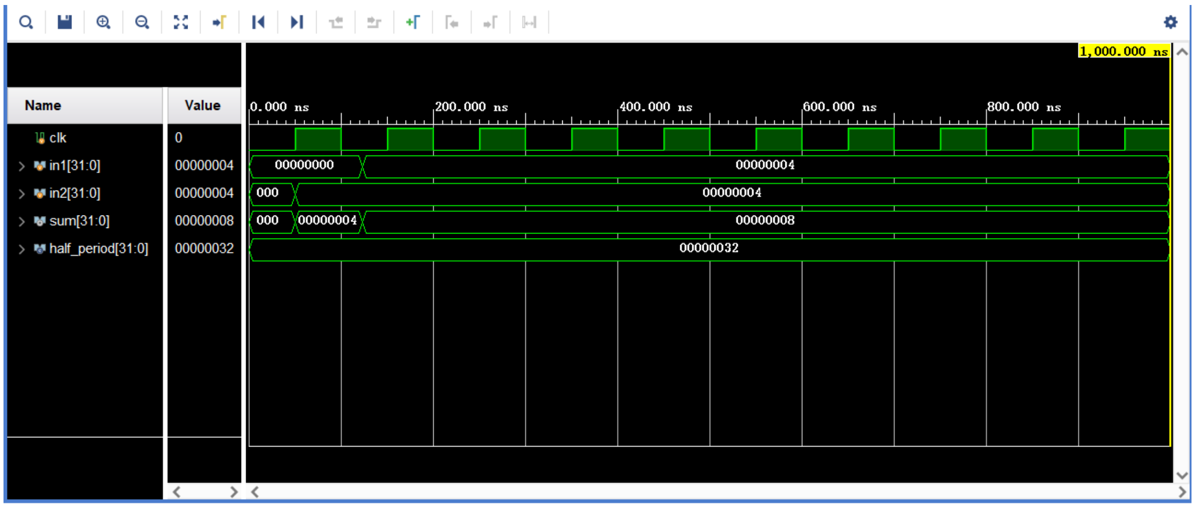
### Register File



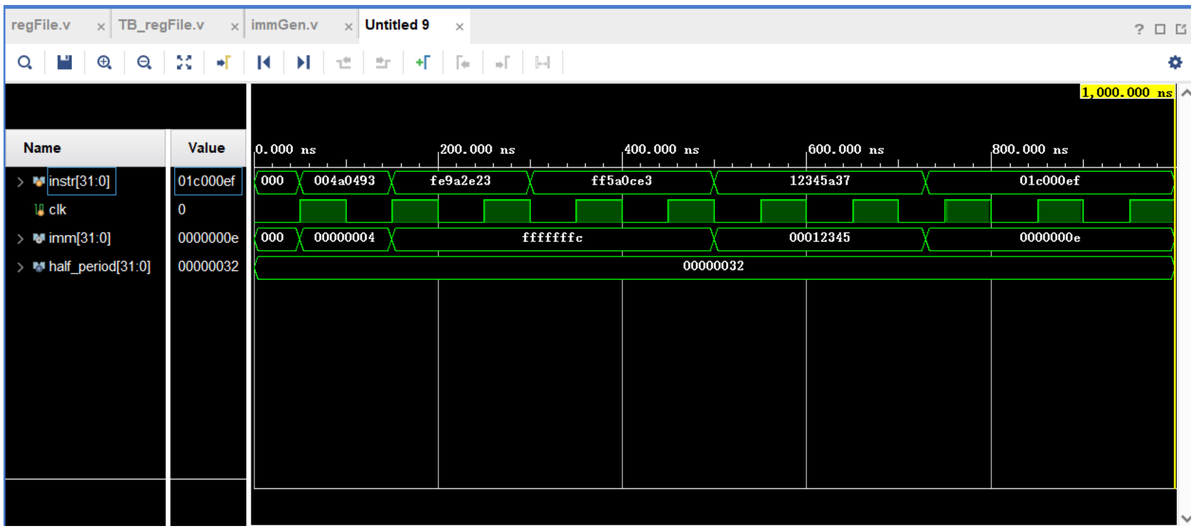
### 32-bit ALU



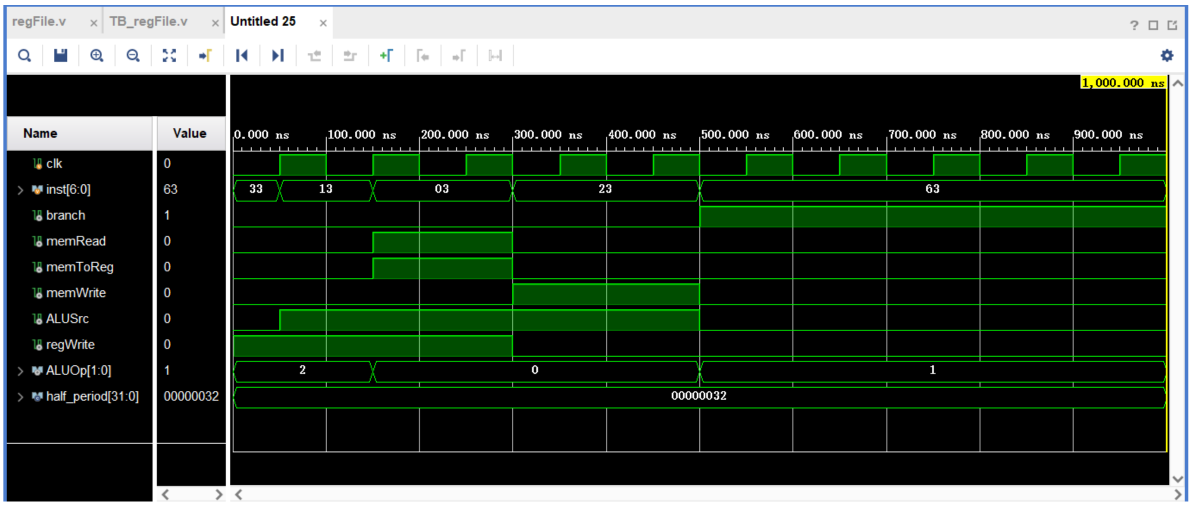
### 32-bit Adder



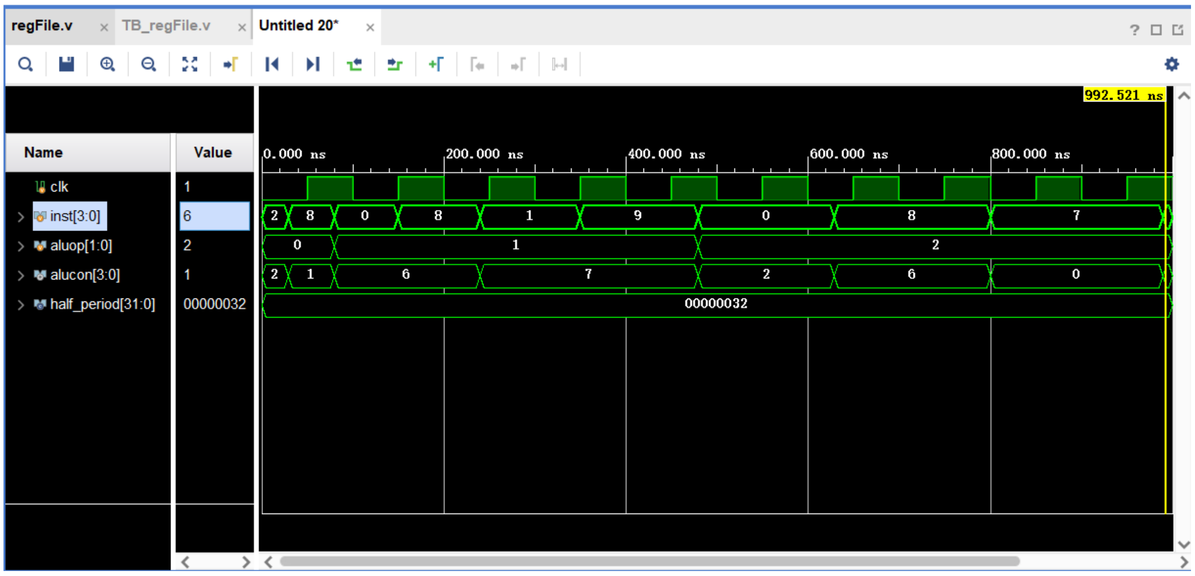
## Immediate Generator



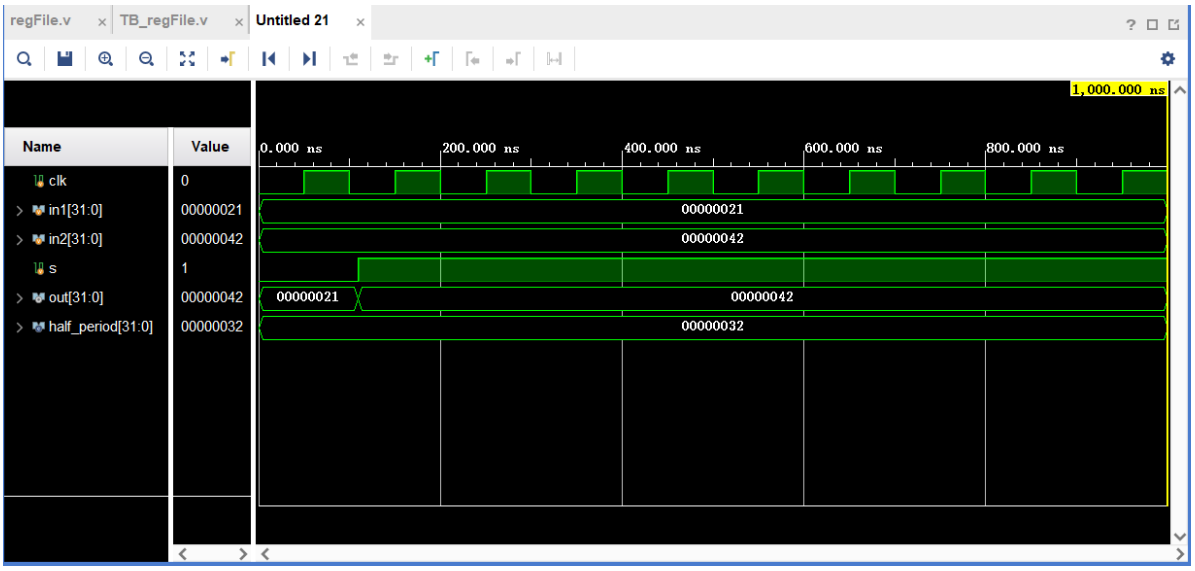
## Control Unit



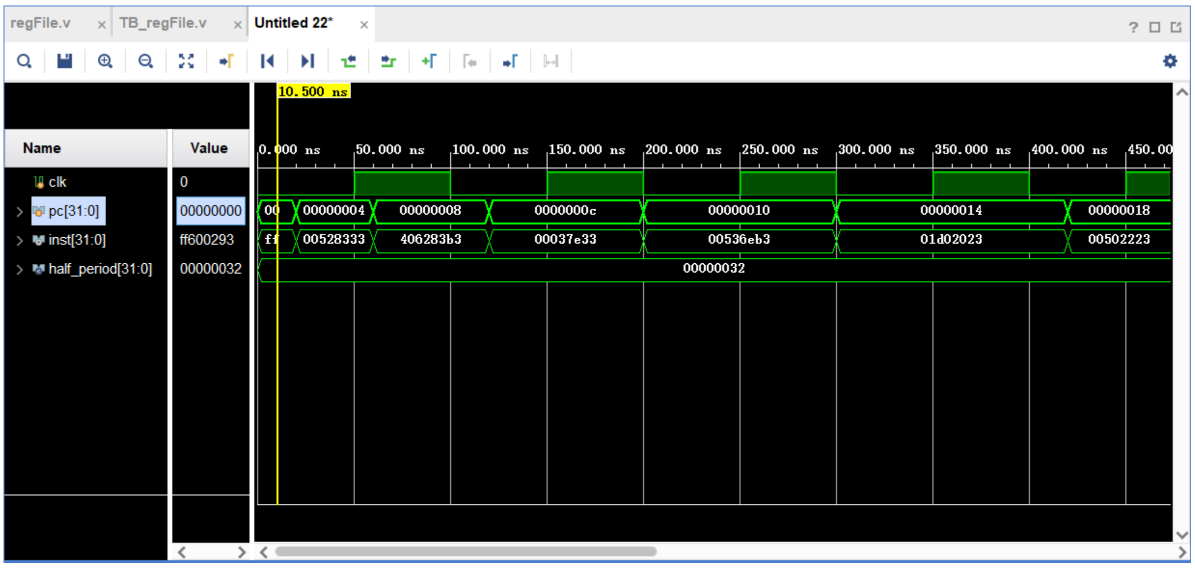
## ALU Control



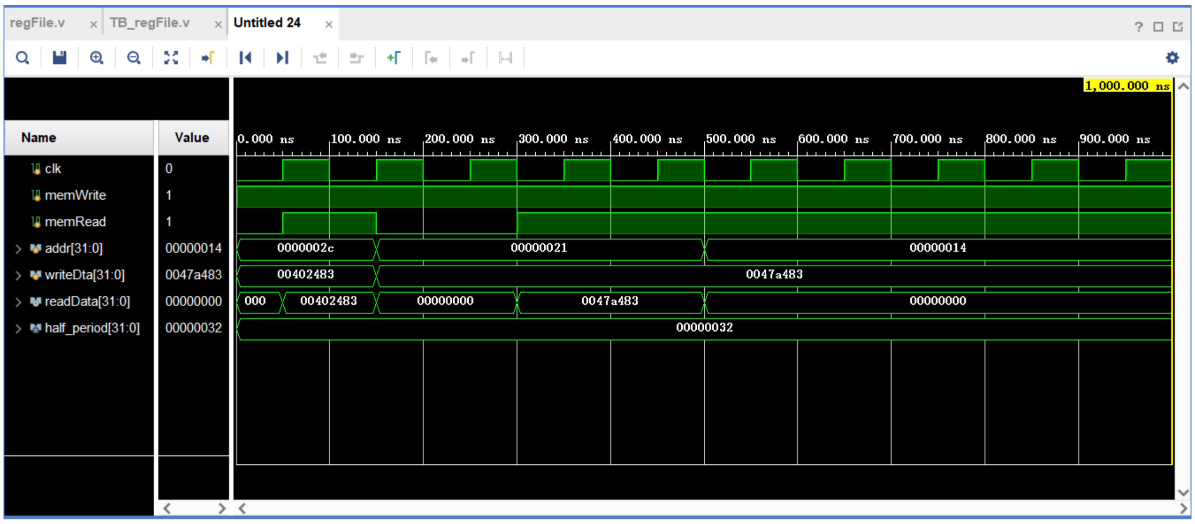
## 2-to-1 MUX



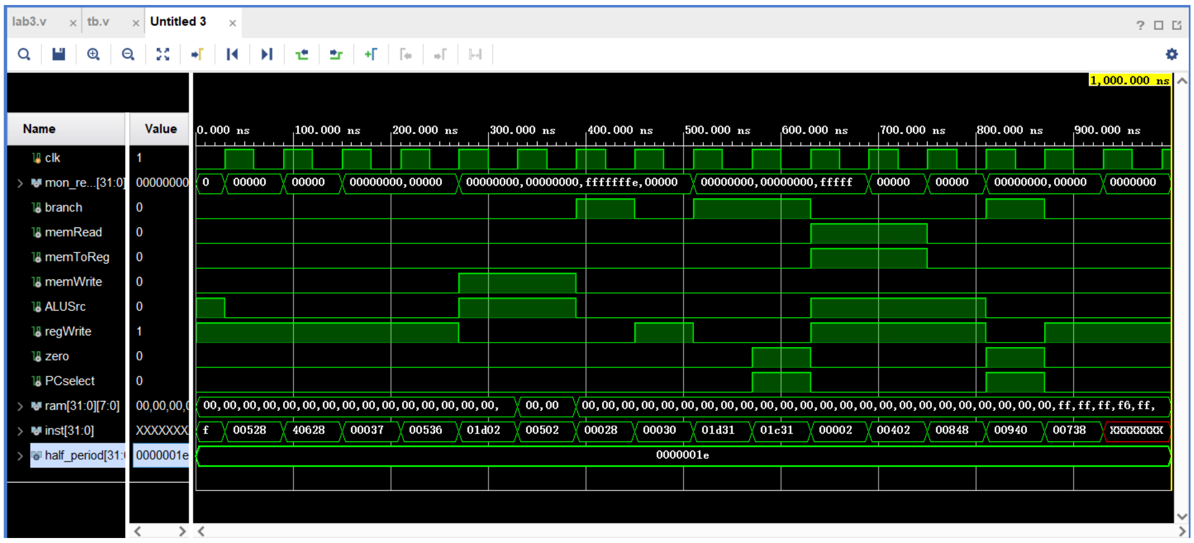
## Instruction Memory



## Data Memory

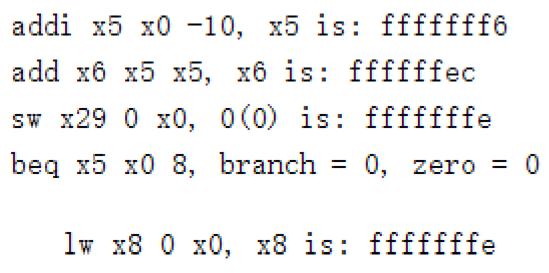


## simulation results for instructions



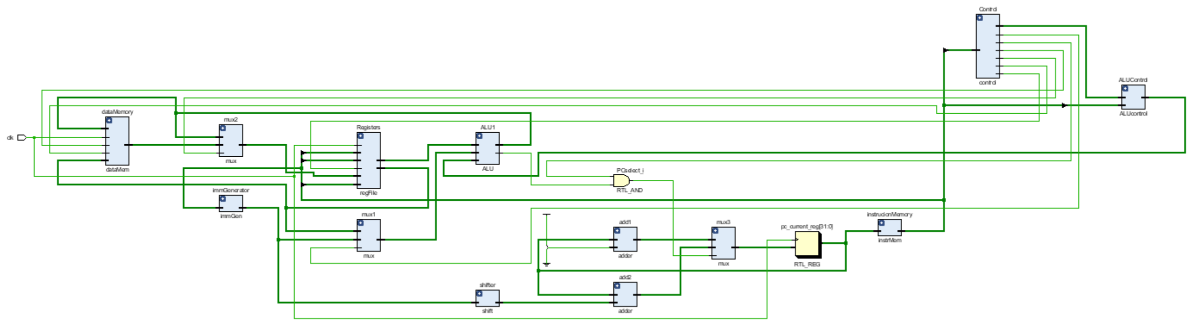
A clearer simulation result is given below:

```
#35 $display("addi x5 x0 -10, x5 is: %h",mon_regs[5]);
#95 $display("add x6 x5 x5, x6 is: %h",mon_regs[6]);
#335 $display("sw x29 0 x0, 0(0) is: %h",{ram[3],ram[2],ram[1],ram[0]});
#455 $display("beq x5 x0 8, branch = %b, zero = %b \n",branch,zero);
#695 $display("lw x8 0 x0, x8 is: %h",mon_regs[8]);
```



```
addi x5 x0 -10, x5 is: fffffff6
add x6 x5 x5, x6 is: ffffffec
sw x29 0 x0, 0(0) is: fffffffe
beq x5 x0 8, branch = 0, zero = 0

  lw x8 0 x0, x8 is: fffffffe
```

## RTL schematic



For a clearer version, see `./schematic.pdf`.