Lab02 registerfile 实验报告

PB16051448 赵敏帆

实验目的

- 编写寄存器文件模块，实现 64*32bit 寄存器文件，并具备一组读端口和一组写端口
- 利用寄存器文件和 ALU 模块实现斐波那契数列计算

实验平台

Vivado 2016.4

实验过程

总体设计思想

- 编写寄存器文件模块
- 利用 control 模块产生控制信号，控制寄存器的读写和 ALU 的数据输入

  > 由于 ALU 模块是组合逻辑，故只需要控制其输入数据即可

设计寄存器文件

```
module regfile(
input clk,
input rst_n,
input [5:0]rAddr,
input [5:0]wAddr,
input [31:0]wDin,
output [31:0]rDout,
input wEna
    );
    reg [31:0]regfiles[0:63];
    reg [5:0] i;
    always@( negedge rst_n or posedge clk)
    if(!rst_n)
      begin
      regfiles[0]<=2;
      regfiles[1]<=2;
      end
    else
    if(wEna)
        regfiles[wAddr]<=wDin;
    assign rDout=regfiles[rAddr];
endmodule
```

设计 control 模块

control 模块通过一个 `count` 计数器，以三个时钟周期为一个寄存器读写周期，读两次寄存器，做加法计算后写一次寄存器。

```
module control(
input clk,
input rst_n,
input [31:0]read_data,
output reg [31:0]out1,
output reg [31:0]out2,
output reg [5:0]read,
output reg wena,
output reg [5:0]write
    );
    reg [1:0]count;
    always @(posedge clk or negedge rst_n)
    if(!rst_n)
    begin
    count<=0;
    read<=0;
    wena<=0;
    out1<=0;
    out2<=0;
    end
    else if(read<63)
    case(count)
    2'h0:
    begin
    out1<=read_data;
    read<=read+1;
    count<=count+1;
    wena<=0;
    end
    2'h1:
    begin
    out2<=read_data;
    write<=read+1;
    count<=count+1;
    wena<=0;
    end
    2'h2:
    begin
    wena<=1;
    count<=0;
    end
    endcase
endmodule
```

ALU 模块

```
module ALU(
    input signed  [31:0] alu_a,
    input signed  [31:0] alu_b,
    input  [4:0] alu_op,
    output reg [31:0] alu_out
    );
        parameter     A_NOP= 5'h00;
        parameter    A_ADD= 5'h01;
        parameter    A_SUB= 5'h02;
        parameter    A_AND= 5'h03;
        parameter    A_OR = 5'h04;
        parameter    A_XOR= 5'h05;
        parameter    A_NOR= 5'h06;
          always@(*)
           case(alu_op)
            A_NOP: alu_out = 0;
            A_ADD: alu_out = alu_a+alu_b;
            A_SUB: alu_out = alu_a-alu_b;
            A_AND: alu_out = alu_a &alu_b;
            A_OR:  alu_out = alu_a|alu_b;
            A_XOR: alu_out = alu_a^alu_b;
            A_NOR: alu_out = ~(alu_a|alu_b);
            default: alu_out=0;
          endcase
endmodule
```

top 顶层模块

```
module top(
    input clk,
    input rst_n
    );
    wire [5:0]read;
    wire [5:0]write;
    wire [31:0]out1;
    wire [31:0]out2;
    wire [31:0]Din;
    wire [31:0]read_data;
    wire wen;
     control control1(
     .clk (clk),
     .rst_n (rst_n),
     .read_data (read_data),
     .out1 (out1),
     .out2 (out2),
     .read (read),
      .wena (wen),
     .write (write)

     );

     regfile  regfile1(
```

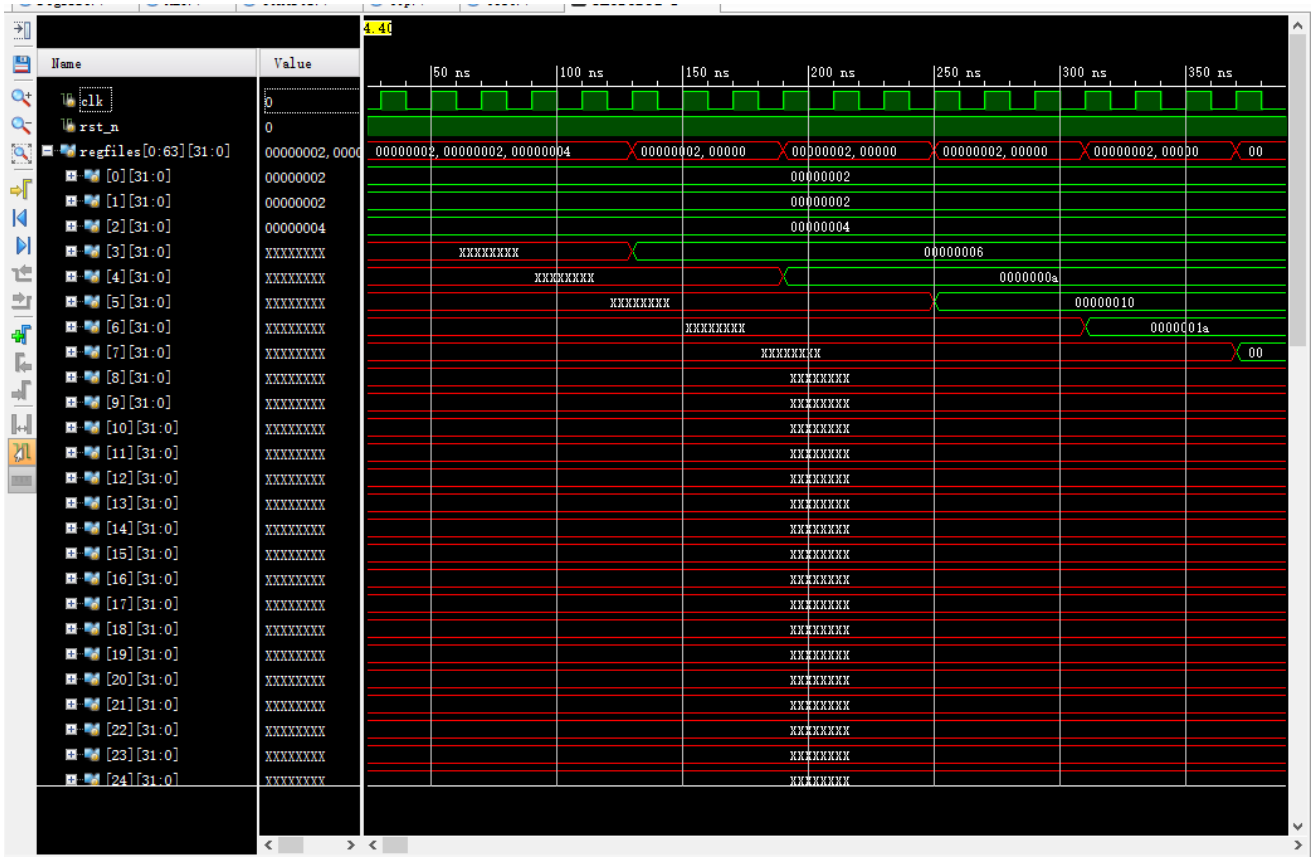
```
            .clk (clk),
            .rst_n (rst_n),
            .rAddr (read),
            .wAddr (write),
            .wDin (Din),
            .rDout (read_data),
            .wEna (wen)
            );
     ALU alu(
     .alu_a (out1),
     .alu_b (out2),
     .alu_op (5'h1),
     .alu_out (Din)
     );
endmodule
```

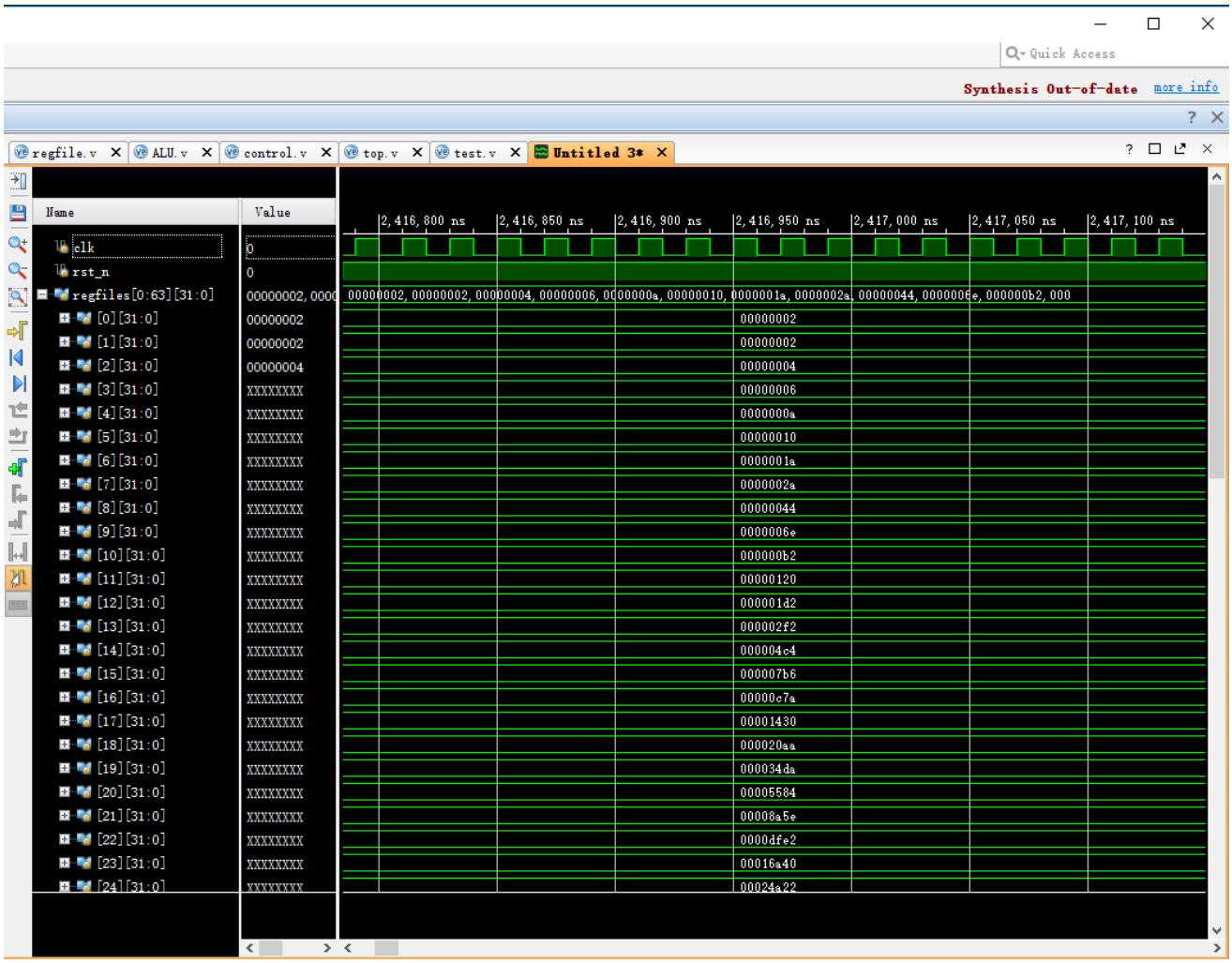
仿真文件设计

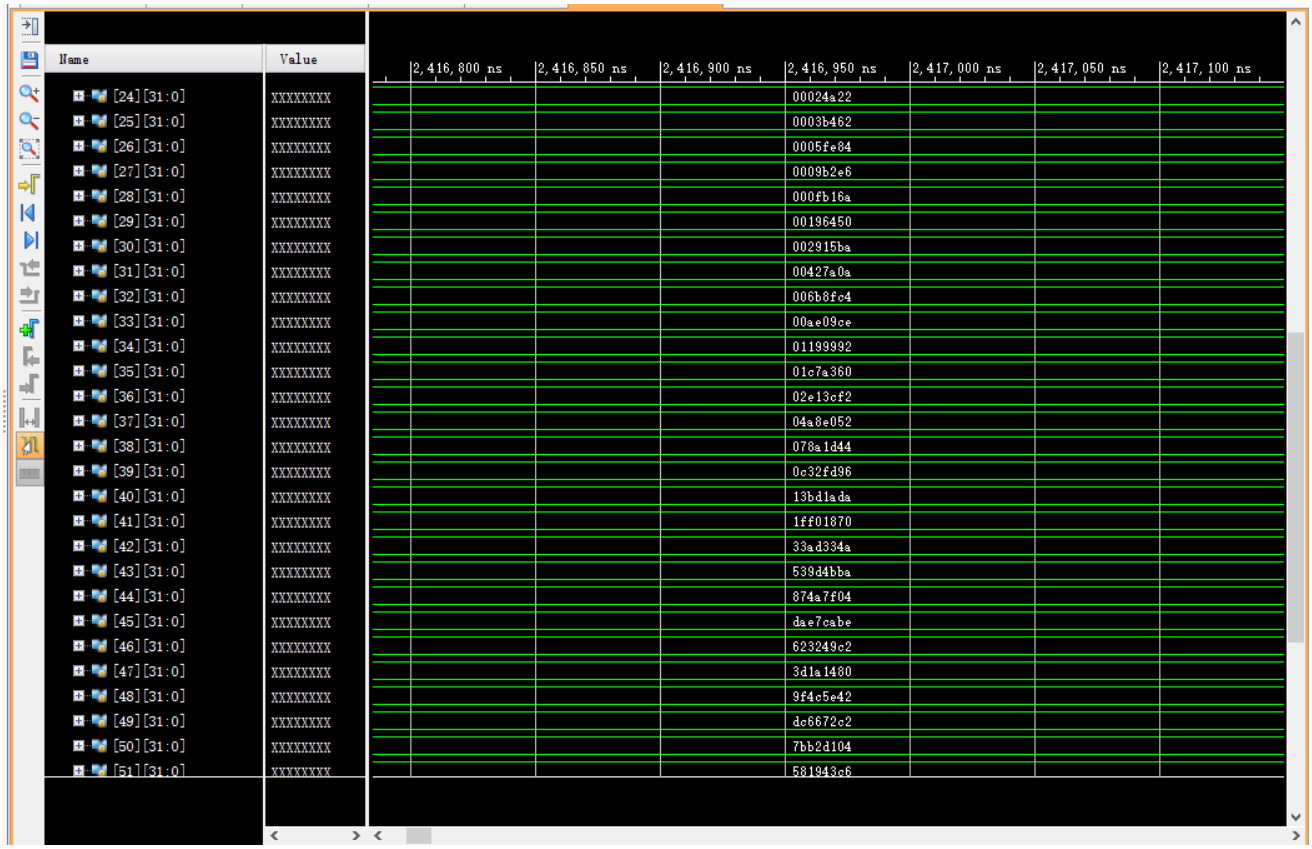
```
module test(

    );
    reg clk;
    reg rst_n;

    top test(
    .clk (clk),
    .rst_n (rst_n)
    );
    always #10 clk=~clk;
    initial begin
    clk=0;
    rst_n=0;
    #10
    rst_n=1;
    end
endmodule
```

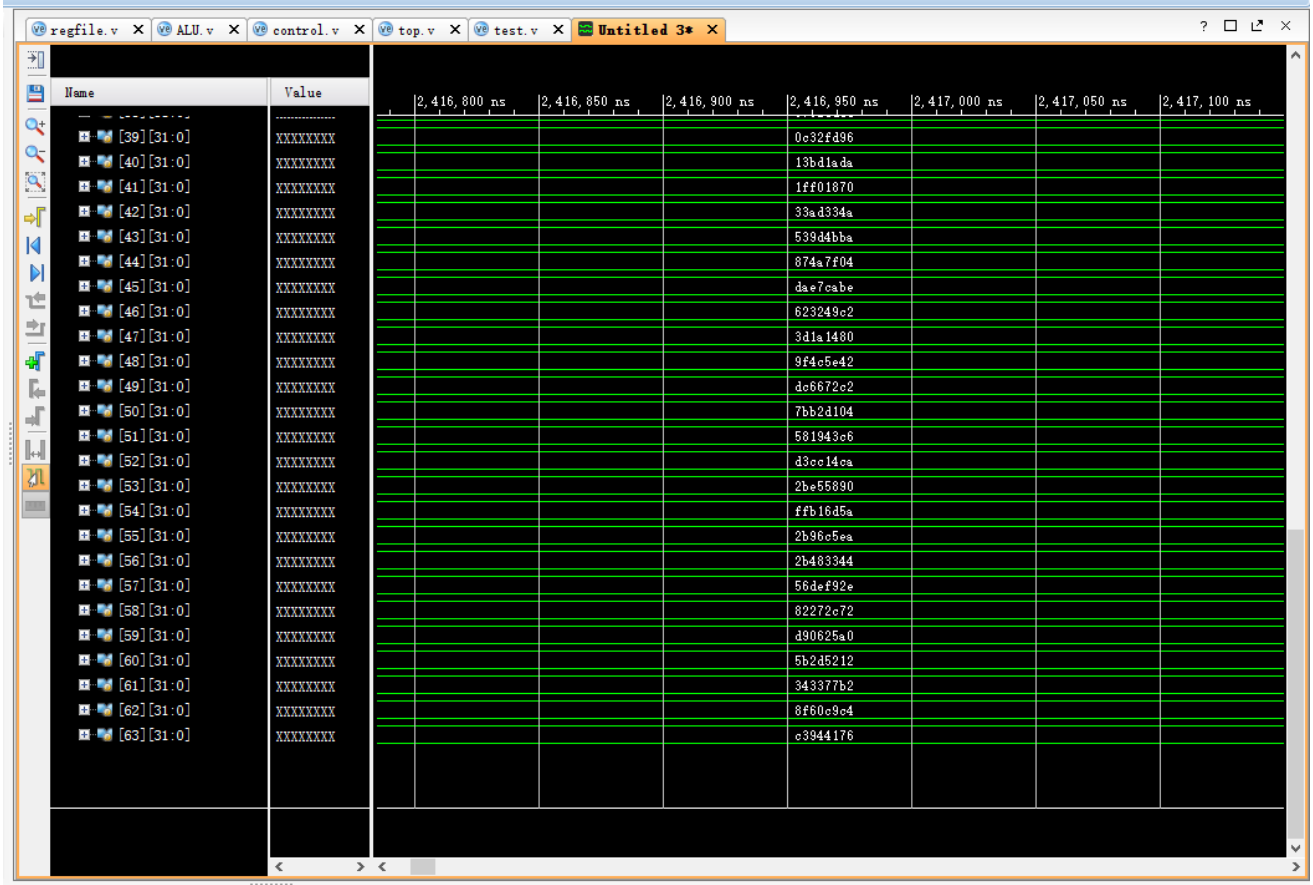
实验结果

仿真结果


Name
Value
clk
rst_n
regfiles[0:63][31:0]
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns


Quick Access
Synthesis Out-of-date
more info
regfile.v
ALU.v
control.v
top.v
test.v
Untitled 3*
Name
Value
clk
rst_n
regfiles[0:63][31:0]
2,416,800 ns
2,416,850 ns
2,416,900 ns
2,416,950 ns
2,417,000 ns
2,417,050 ns
2,417,100 ns


Name
Value
2,416,800 ns
2,416,850 ns
2,416,900 ns
2,416,950 ns
2,417,000 ns
2,417,050 ns
2,417,100 ns
[24][31:0] XXXXXXXX 00024a22
[25][31:0] XXXXXXXX 0003b462
[26][31:0] XXXXXXXX 0005fe84
[27][31:0] XXXXXXXX 0009b2e6
[28][31:0] XXXXXXXX 000fb16a
[29][31:0] XXXXXXXX 00196450
[30][31:0] XXXXXXXX 002915ba
[31][31:0] XXXXXXXX 00427a0a
[32][31:0] XXXXXXXX 006b8fc4
[33][31:0] XXXXXXXX 00ae09ce
[34][31:0] XXXXXXXX 01199992
[35][31:0] XXXXXXXX 01c7a360
[36][31:0] XXXXXXXX 02e13cf2
[37][31:0] XXXXXXXX 04a8e052
[38][31:0] XXXXXXXX 078a1d44
[39][31:0] XXXXXXXX 0c32fd96
[40][31:0] XXXXXXXX 13bd1ada
[41][31:0] XXXXXXXX 1ff01870
[42][31:0] XXXXXXXX 33ad334a
[43][31:0] XXXXXXXX 539d4bba
[44][31:0] XXXXXXXX 874a7f04
[45][31:0] XXXXXXXX dae7cabe
[46][31:0] XXXXXXXX 623249c2
[47][31:0] XXXXXXXX 3d1a1480
[48][31:0] XXXXXXXX 9f4c5e42
[49][31:0] XXXXXXXX dc6672c2
[50][31:0] XXXXXXXX 7bb2d104
[51][31:0] XXXXXXXX 581943c6


regfile.v
ALU.v
control.v
top.v
test.v
Untitled 3*
Name
Value
2,416,800 ns
2,416,850 ns
2,416,900 ns
2,416,950 ns
2,417,000 ns
2,417,050 ns
2,417,100 ns
[39][31:0] XXXXXXXX 0c32fd96
[40][31:0] XXXXXXXX 13bd1ada
[41][31:0] XXXXXXXX 1ff01870
[42][31:0] XXXXXXXX 33ad334a
[43][31:0] XXXXXXXX 539d4bba
[44][31:0] XXXXXXXX 874a7f04
[45][31:0] XXXXXXXX dae7cabe
[46][31:0] XXXXXXXX 623249c2
[47][31:0] XXXXXXXX 3d1a1480
[48][31:0] XXXXXXXX 9f4c5e42
[49][31:0] XXXXXXXX dc6672c2
[50][31:0] XXXXXXXX 7bb2d104
[51][31:0] XXXXXXXX 581943c6
[52][31:0] XXXXXXXX d3cc14ca
[53][31:0] XXXXXXXX 2be55890
[54][31:0] XXXXXXXX ffb16d5a
[55][31:0] XXXXXXXX 2b96c5ea
[56][31:0] XXXXXXXX 2b483344
[57][31:0] XXXXXXXX 56def92e
[58][31:0] XXXXXXXX 82272c72
[59][31:0] XXXXXXXX d90625a0
[60][31:0] XXXXXXXX 5b2d5212
[61][31:0] XXXXXXXX 343377b2
[62][31:0] XXXXXXXX 8f60c9c4
[63][31:0] XXXXXXXX c3944176

心得体会

- 通过在仿真中添加 `regfile` 的波形图来观察结果
- 可以在初始化的时候使用一个 `for` 循环来将 regfiles 的所有值都初始化，这样就可以在波形图中不出现 `x`