Alex Oswald

EECE 2323

Due: 22 Sep 2020

Pre-Lab Assignment 4

Hi all,

For Prelab 4, please submit a single document which contains:

1) reg\_file.v

2) reg\_file\_tb.v

3) Waveform of reg\_file\_tb.v

4) Create Test Vectors for the ALU-Regfile module in lab 4 (You do not need to code or simulate it, so please consider every necessary cases)

**reg\_file.v**

1. `timescale 1ns / 1ps
3. module reg\_file(
4. //inputs
5. input rst, // rst=reset OR clr=clear
6. input clk,
7. input wr\_en,
8. input [1:0] rd0\_addr,
9. input [1:0] rd1\_addr,
10. input [1:0] wr\_addr,
11. input [8:0] wr\_data, //piped from ALU
12. //outputs
13. output reg [8:0] rd0\_data,
14. output reg [8:0] rd1\_data);
16. reg [8:0] mem[3:0];
17. integer i;
19. always@(rst, negedge clk)
20. begin
21. **if**(rst)
22. **for** (i=0; i<4; i=i+1)
23. mem [i] = 0;
24. **else** **if**(wr\_en)
25. mem[wr\_addr] = wr\_data;
26. **else**
27. begin
28. rd0\_data = mem[rd0\_addr];
29. rd1\_data = mem[rd1\_addr];
30. end
31. end
33. endmodule

**reg\_file\_tb.v**

1. `timescale 1ns / 1ps
3. module reg\_file\_tb();
4. // Inputs
5. reg rst, clk, wr\_en;
6. reg [1:0] rd0\_addr, rd1\_addr, wr\_addr;
7. reg [8:0] wr\_data; //piped from alu
8. // Outputs
9. wire [8:0] rd0\_data, rd1\_data;
11. // Initiate the Unit Under Test (UUT)
12. reg\_file uut (
13. .rst(rst),
14. .clk(clk),
15. .wr\_en(wr\_en),
16. .rd0\_addr(rd0\_addr),
17. .rd1\_addr(rd1\_addr),
18. .wr\_addr(wr\_addr),
19. .wr\_data(wr\_data), //piped from ALU
20. .rd0\_data(rd0\_data),
21. .rd1\_data(rd1\_data));
23. // Initialize Inputs (stimulus)
24. initial
25. // 1) Initial pulse for rst (clr?)
26. begin
27. #0
28. rst = 0;
29. #5
30. rst = 1;
31. #5
32. rst = 0;
33. end
35. // 2) Clock signal
36. initial
37. begin
38. clk  = 1;
39. forever #5  clk = ~clk;
40. end
42. // 3) Values for other inputs
43. initial
44. begin
45. $monitor("%d  ", $time, , " rst=%b ", rst, " clk=%b ", clk, " wr\_en=%b ", wr\_en, " rd0\_addr=%d ", rd0\_addr, " rd1\_addr=%d ", rd1\_addr, " wr\_addr=%d ", wr\_addr, " wr\_data=%d ", wr\_data, " rd0\_data=%d ", rd0\_data, " rd1\_data=%d ", rd1\_data);
46. #0 // Initially
47. rd0\_addr = 0;
48. rd1\_addr = 0;
49. wr\_en = 0;
50. wr\_addr = 0;
51. wr\_data = 9'd0;
53. // Test Things
54. #10 wr\_en = 1;  wr\_addr = 0;    wr\_data = 9'd0;
55. #10 wr\_en = 1;  wr\_addr = 1;    wr\_data = 9'd1;
56. #10 wr\_en = 1;  wr\_addr = 2;    wr\_data = 9'd2;
57. #10 wr\_en = 1;  wr\_addr = 3;    wr\_data = 9'd3;
58. #10 wr\_en = 0;  rd0\_addr = 0;   rd1\_addr = 1;
59. #10 wr\_en = 0;  rd0\_addr = 2;   rd1\_addr = 3;
61. #10 wr\_en = 1;  wr\_addr = 0;    wr\_data = 9'd20;
62. #10 wr\_en = 1;  wr\_addr = 1;    wr\_data = 9'd21;
63. #10 wr\_en = 1;  wr\_addr = 2;    wr\_data = 9'd22;
64. #10 wr\_en = 1;  wr\_addr = 3;    wr\_data = 9'd23;
65. #10 wr\_en = 0;  rd0\_addr = 0;   rd1\_addr = 1;
66. #10 wr\_en = 0;  rd0\_addr = 2;   rd1\_addr = 3;
68. #10 wr\_en = 1;  wr\_addr = 0;    wr\_data = 9'd175;
69. #10 wr\_en = 1;  wr\_addr = 1;    wr\_data = 9'd12;
70. #10 wr\_en = 1;  wr\_addr = 2;    wr\_data = 9'd511;
71. #10 wr\_en = 1;  wr\_addr = 3;    wr\_data = 9'd255;
72. #10 wr\_en = 0;  rd0\_addr = 1;   rd1\_addr = 0;
73. #10 wr\_en = 0;  rd0\_addr = 3;   rd1\_addr = 2;
75. #10 $finish;
76. end
78. endmodule

**Waveform of reg\_file\_tb.v**

**Graphical user interface

Description automatically generated**

**Test Vectors for the ALU-Regfile module in Lab 4**

1. ALUsrc1=0; ALUsrc2=0;  // ReadData1 && ReadData2
2. ALUop = 3'b000; // a + b;
3. #10 RegWrite = 1;   WriteAddr = 1;  WriteData = 9'd1;
4. #10 RegWrite = 1;   WriteAddr = 2;  WriteData = 9'd2;
5. #10 RegWrite = 1;   WriteAddr = 3;  WriteData = 9'd3;
6. #10 RegWrite = 0;   ReadAddr1 = 0;  ReadAddr2 = 1;
7. #10 RegWrite = 0;   ReadAddr1 = 2;  ReadAddr2 = 3;
9. ALUop = 3'b001; // ~b;
10. #10 RegWrite = 1;   WriteAddr = 0;  WriteData = 9'd20;
11. #10 RegWrite = 1;   WriteAddr = 1;  WriteData = 9'd21;
12. #10 RegWrite = 1;   WriteAddr = 2;  WriteData = 9'd22;
13. #10 RegWrite = 1;   WriteAddr = 3;  WriteData = 9'd23;
14. #10 RegWrite = 0;   ReadAddr1 = 0;  ReadAddr2 = 1;
15. #10 RegWrite = 0;   ReadAddr1 = 2;  ReadAddr2 = 3;
17. ALUop = 3'b010; // a & b;
18. #10 RegWrite = 1;   WriteAddr = 0;  WriteData = 9'd175;
19. #10 RegWrite = 1;   WriteAddr = 1;  WriteData = 9'd12;
20. #10 RegWrite = 1;   WriteAddr = 2;  WriteData = 9'd511;
21. #10 RegWrite = 1;   WriteAddr = 3;  WriteData = 9'd255;
22. #10 RegWrite = 0;   ReadAddr1 = 1;  ReadAddr2 = 0;
23. #10 RegWrite = 0;   ReadAddr1 = 3;  ReadAddr2 = 2;
25. ALUsrc1=0; ALUsrc2=1; // ReadData1 && Instr\_i
26. ALUop = 3'b011; // a | b;
27. #10 RegWrite = 1;   WriteAddr = 0;  WriteData = 9'd399;
28. #10 RegWrite = 0;   ReadAddr1 = 0;  Instr\_i = 8'd42;
30. ALUop = 3'b100; // a >>> 1;
31. #10 RegWrite = 1;   WriteAddr = 1;  WriteData = 9'd425;
32. #10 RegWrite = 0;   ReadAddr1 = 1;  Instr\_i = 8'd200;
34. ALUop = 3'b101; // a << 1;
35. #10 RegWrite = 1;   WriteAddr = 2;  WriteData = 9'd1;
36. #10 RegWrite = 0;   ReadAddr1 = 2;  Instr\_i = 8'd23;
38. ALUop = 3'b110; // 0;
39. #10 RegWrite = 1;   WriteAddr = 3;  WriteData = 9'd1;
40. #10 RegWrite = 0;   ReadAddr1 = 3;  Instr\_i = 8'd23;
42. #10 ALUsrc1=1; ALUsrc2=0; // zero\_register && ReadData2
43. ALUop = 3'b110; // 0;
44. #10 Instr\_i = 8'd255;
46. ALUop = 3'b111; // 0;
47. #10 Instr\_i = 8'd255;
49. #10 $finish;