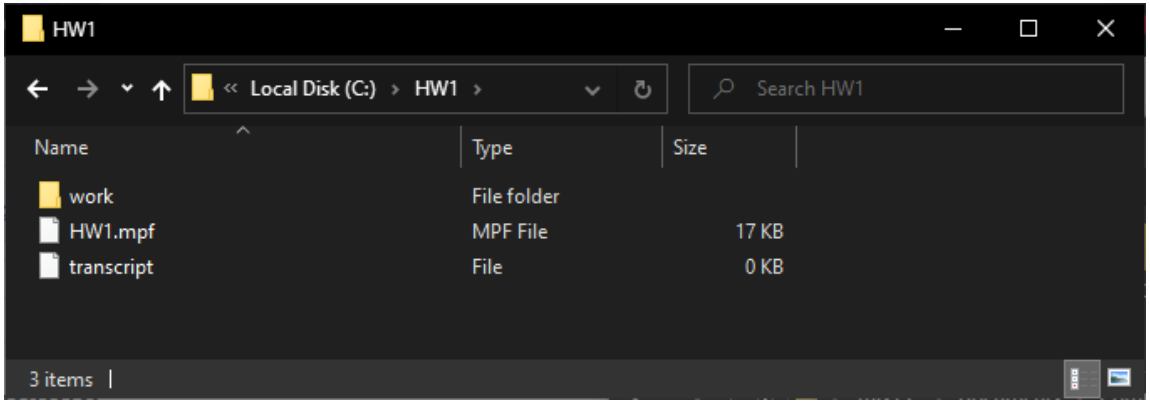
Example of HW1

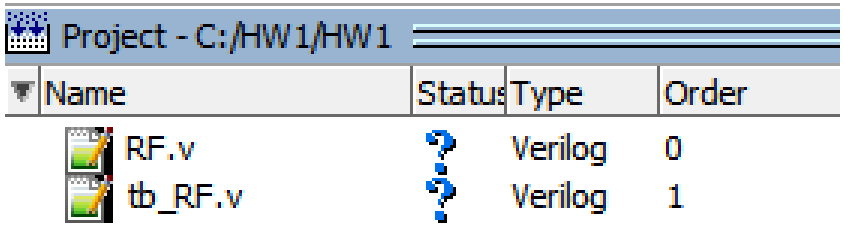
Create a project

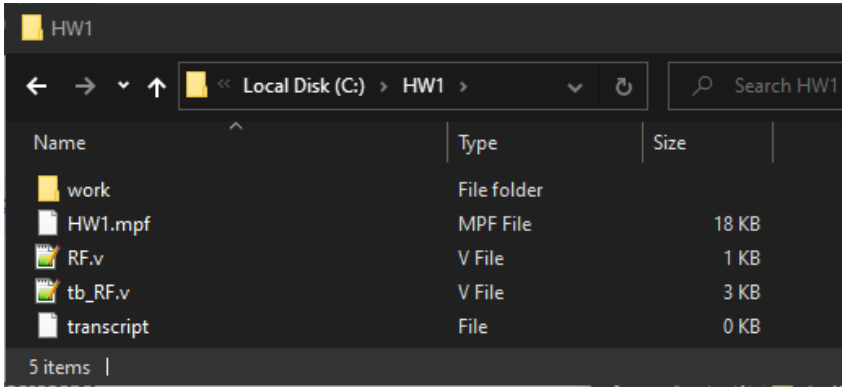
1. Click “File” → “New” → “Project…”.
2. Key in Project Name = “HW1”.
3. Key in or Select “Project Location” = “C:/HW1”.
4. Click “OK”.



Add RF.v to project

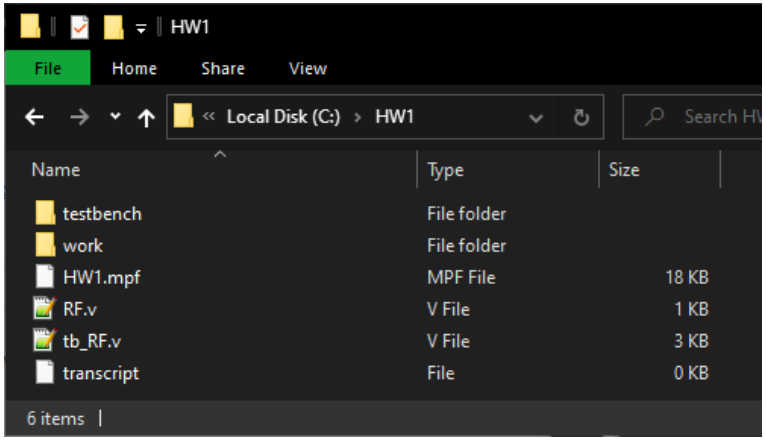
1. Right click in “Project” tab.
2. Click “Add to Project” → “Existing File…”.
3. Check “Copy to project directory” and “Browse..” tb_RF.v and RF.v files.

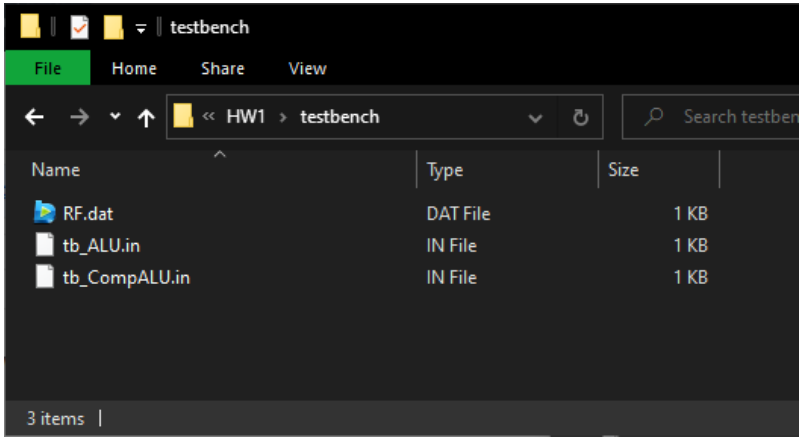




Copy “testbench” folder

1. Copy “testbench” folder and paste at “C:/HW1/”.
2. Inside “testbench” folder must have:
   a) RF.dat
   b) tb_ALU.in
   c) tb_CompALU.in





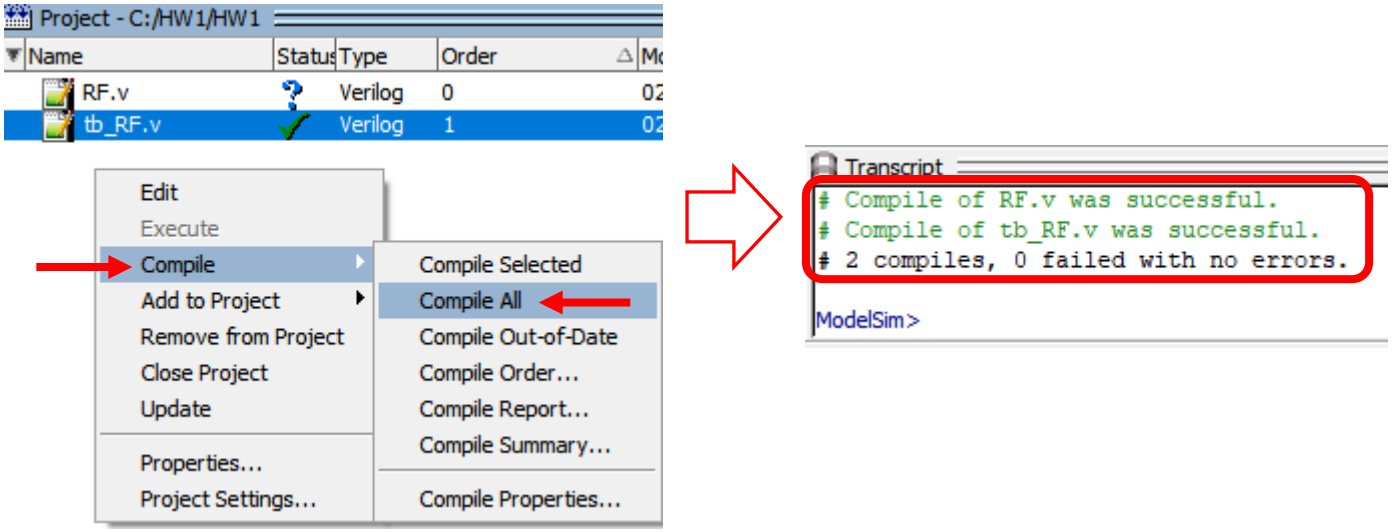
Edit RF.v

```
25  /*
26   * Declaration of Register File for this project.
27   * CAUTION: DONT MODIFY THE NAME.
28   */
29  module RF();
30
31      /*
32       * Declaration of inner register.
33       * CAUTION: DONT MODIFY THE NAME AND SIZE.
34       */
35      reg [31:0]R[0:31];
36
37  endmodule
38
```

```
26   * Declaration of Register File for this project.
27   * CAUTION: DONT MODIFY THE NAME.
28   */
29  module RF(
30          //Inputs
31          input [4:0] Rs_addr,
32          input [4:0] Rt_addr,
33          //Outputs
34          output [31:0] Rs_data,
35          output [31:0] Rt_data
36
37  );
38          /*
39           * Declaration of inner register.
40           * CAUTION: DONT MODIFY THE NAME AND SIZE.
41           */
42          reg [31:0]R[0:31];
43
44          //Regsiter Muxs
45          assign Rs_data = R[Rs_addr];
46
47  endmodule
48
```

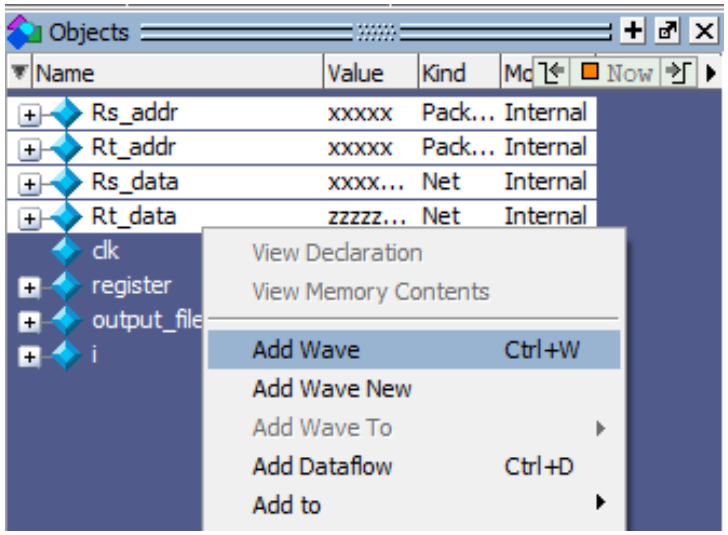
Compile the project

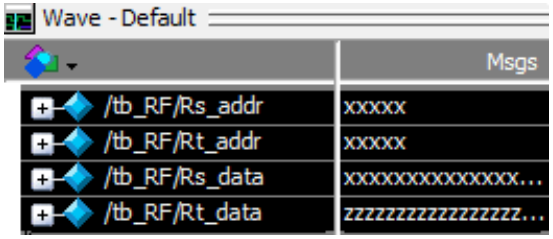
1. Right click in “Project” tab.
2. Click “Compile” → “Compile All”.
3. Done successful as shown in “Transcript” tab.



Simulate tb_RF.v (1)

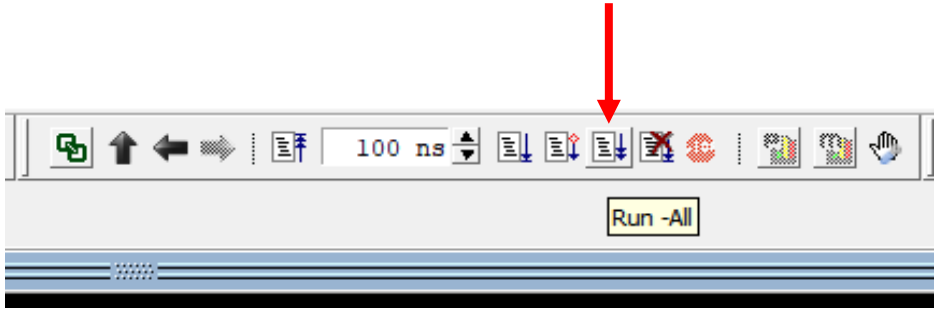
1. Click “Simulate” → “Start Simulation”.
2. Select “tb_RF” in “work”.
3. Click “OK”.
4. Add Wave for “Rs_addr”, “Rt_addr”, “Rs_data”, “Rt_data”.

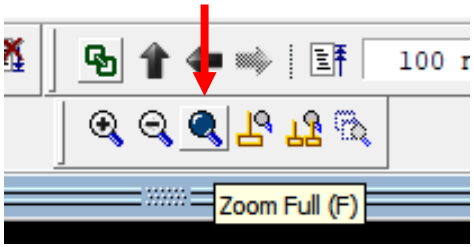




Simulate tb_RF.v (2)

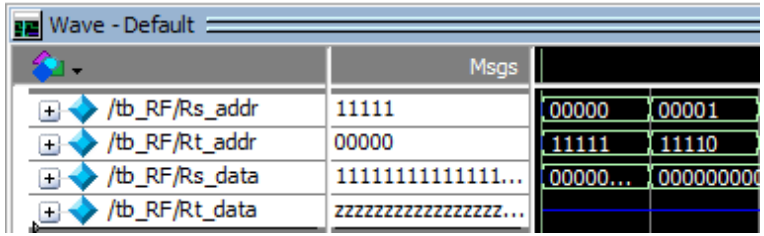
5. Click “Run -All” button on “Simulate” toolbar.
6. Click “Zoom Full” button on “Zoom” toolbar to arrange all wave form signal with window size.

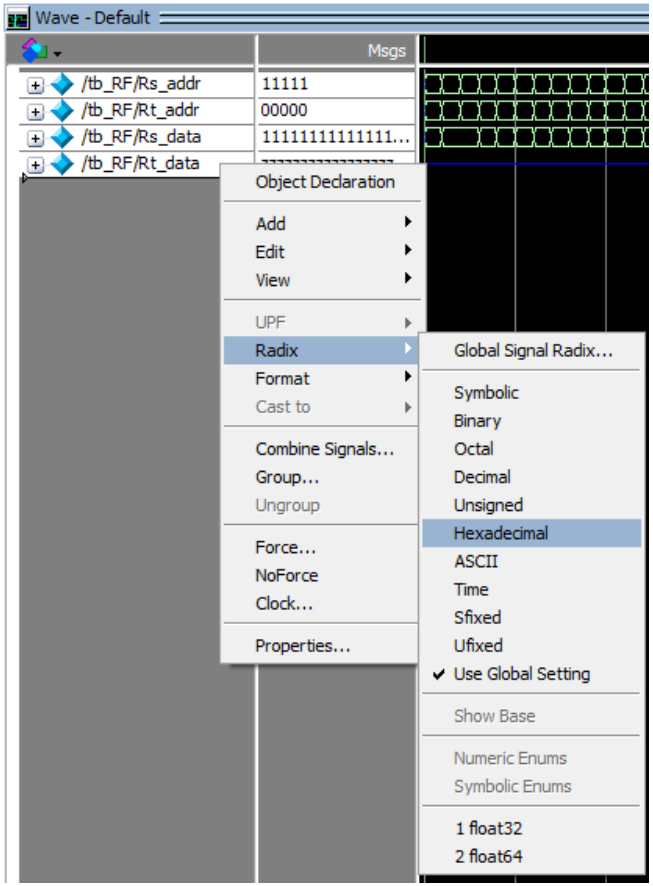




Simulate tb_RF.v (3)

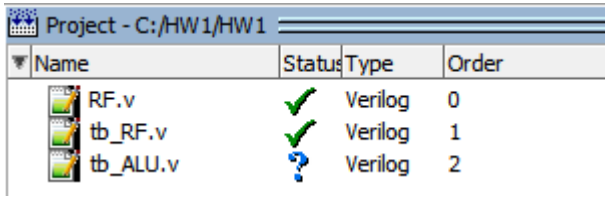
7. Right click on signal in “Wave” tab.
8. Click “Radix” → “Hexadecimal”.

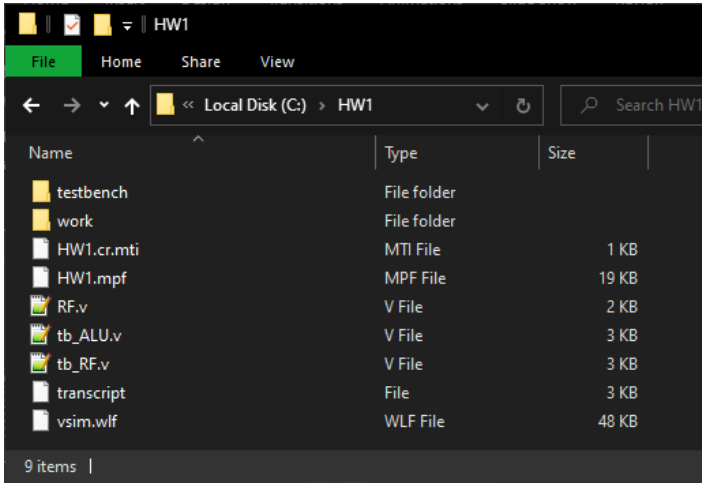




Add tb_ALU.v to project

1. Click “Simulate” → “End Simulation”.
2. Right click in “Project” tab.
3. Click “Add to Project” → “Existing File…”.
4. Check “Copy to project directory” and “Browse..” tb_ALU.v file.





Create ALU.v to project

1. Right click in “Project” tab.
2. Click “Add to Project” → “New File”.
3. Key in “File Name” = “ALU.v”.
4. Select “Add file as type” = “Verilog”.
5. Click “OK”.

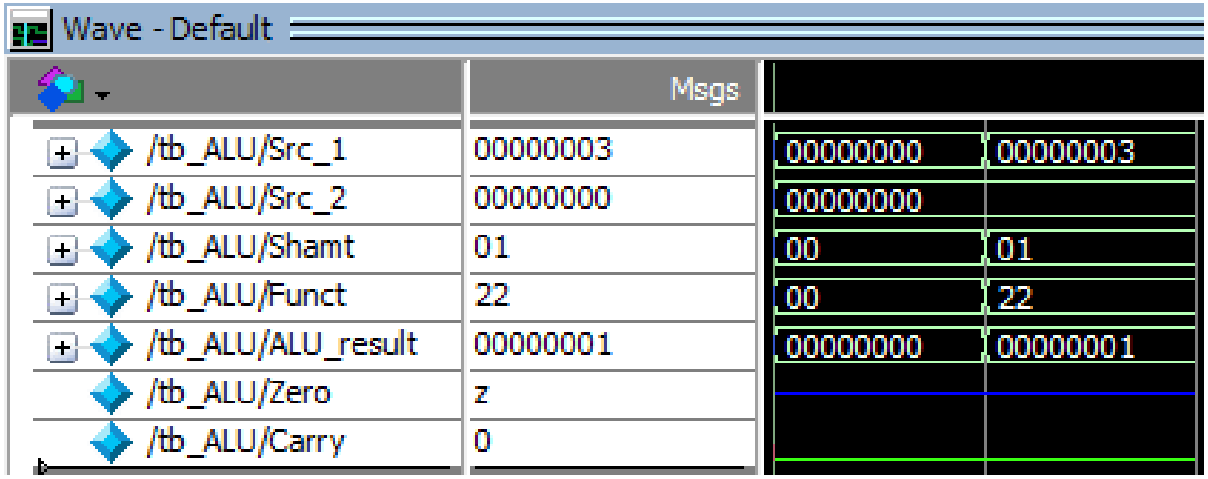
Edit ALU.v and Compile All

1. Edit “ALU.v” then “Compile –All”.

```
1   `define SRL     6'b100010
2
3   module ALU(
4           //      Inputs
5           input   [31:0] Src_1,
6           input   [31:0] Src_2,
7           input   [4:0] Shamt,
8           input   [5:0] Funct,
9           //      Outputs
10          output  reg     [31:0] ALU_result,
11          output  wire    Zero,
12          output  reg     Carry
13  );
14
15          always @(Src_1, Src_2, Shamt, Funct) begin
16                  case(Funct)
17                          `SRL:           {Carry, ALUResult} <= Src_1 >> Shamt;
18                          default:        {Carry, ALUResult} <= 0;
19                  endcase
20
21          end
22
23  endmodule
24
```

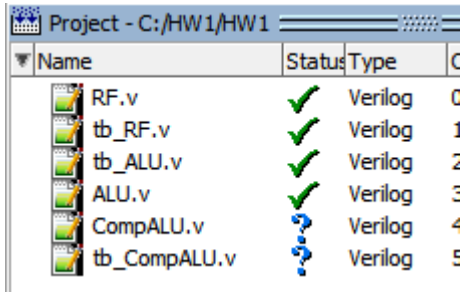
Simulate tb_ALU.v

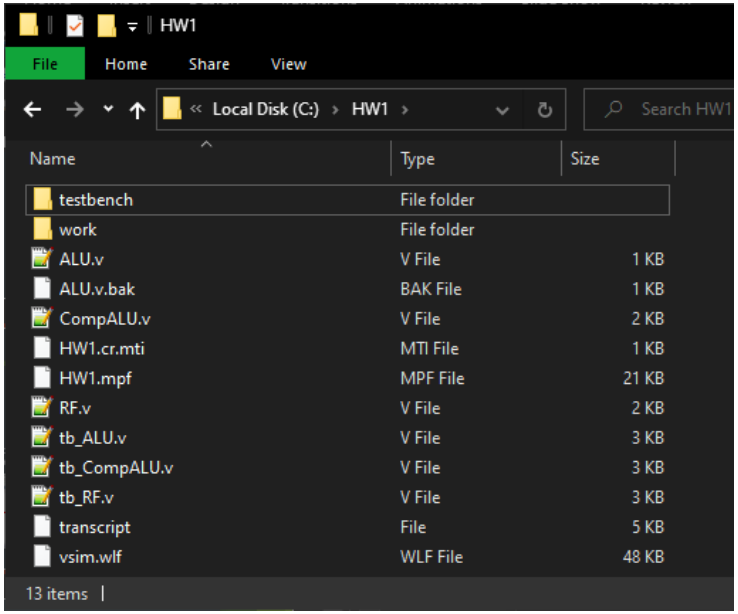
1. Simulate “tb_ALU” in “work”.
2. Add Wave for “Src1”, “Src2”, “Shamt”, “Funct”, “ALU_result”, “Zero”, “Carry”.



Add CompALU.v to project

1. Click “Simulate” → “End Simulation”.
2. Add tb_CompALU.v and CompALU.v files to project.





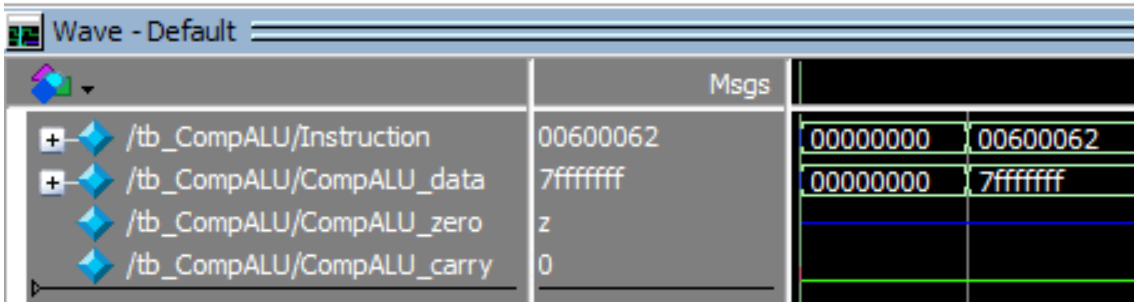
Edit CompALU.v

```
25  /*
26   * Declaration of top entry for this project.
27   * CAUTION: DONT MODIFY THE NAME AND I/O DECLARATION.
28   */
29  module CompALU(
30          //      Inputs
31          Instruction,
32          //      Outputs
33          CompALU_data,
34          CompALU_zero,
35          CompALU_carry
36  );
37
38          /*
39           * Declaration of Register File.
40           * CAUTION: DONT MODIFY THE NAME.
41           */
42
43          RF Register_File();
44
45  endmodule
```

```
25  /*
26   * Declaration of top entry for this project.
27   * CAUTION: DONT MODIFY THE NAME AND I/O DECLARATION.
28   */
29  module CompALU(
30          //      Inputs
31          input [31:0] Instruction,
32          //      Outputs
33          output [31:0] CompALU_data,
34          output CompALU_zero,
35          output CompALU_carry
36  );
37
38          /*
39           * Declaration of Register File.
40           * CAUTION: DONT MODIFY THE NAME.
41           */
42
43          wire [31:0] Inner_Src_1;
44          wire [31:0] Inner_Src_2;
45
46          RF Register_File(
47                  // Inputs
48                  .Rs_addr(Instruction[25:21]),
49                  .Rt_addr(Instruction[20:16]),
50                  // Outputs
51                  .Rs_data(Inner_Src_1),
52                  .Rt_data(Inner_Src_2)
53          );
54
55          ALU Arithmetic_Logical_Unit(
56                  //      Inputs
57                  .Src_1(Inner_Src_1),
58                  .Src_2(Inner_Src_2),
59                  .shamt(Instruction[10:6]),
60                  .funct(Instruction[5:0]),
61                  //      Outputs
62                  .ALUResult(CompALU_data),
63                  .Zero(CompALU_zero),
64                  .Carry(CompALU_carry)
65          );
66
67  endmodule
```

Simulate tb_CompALU.v

1. Simulate “tb_CompALU” in “work”.
2. Add Wave for “Instruction”, “CompALU_data”, “CompALU_ zero”, “CompALU_ carry”.



Disclaimer

Use of this documentation is your acknowledgment that you agree to and will comply with the following notices and disclaimers. You are advised to seek appropriate legal, engineering, and other professional advice regarding the use, interpretation, and effect of this documentation.

This document only provides students who take this course (Computer Organization). Any modification or spreading of this document without permission is not allowed.

Copyright (C) 2024 Embedded System Software Lab.

All Right Reserved.