

```

1. module tb;
2.
3.
4. integer i = 0;
5.
6. top dut();
7.
8. ///////////////updating value of all GPR to 2
9. initial begin
10. for( i = 0; i < 32; i = i + 1)
11. begin
12. dut.GPR[i] = 2;
13. end
14. end
15.
16.
17.
18. initial begin
19. ////////////// immediate add op
20. $display("-----");
21. dut.IR = 0;
22. dut.`imm_mode = 1;
23. dut.`oper_type = 2;
24. dut.`rsrc1 = 2;///gpr[2] = 2
25. dut.`rdst = 0;///gpr[0]
26. dut.`isrc = 4;
27. #10;
28. $display("OP:ADI Rsrc1:%0d Rsrc2:%0d Rdst:%0d",dut.GPR[2], dut.`isrc, dut.GPR[0]);
29. $display("-----");
30. ///////////////register add op
31. dut.IR = 0;
32. dut.`imm_mode = 0;
33. dut.`oper_type = 2;
34. dut.`rsrc1 = 4;
35. dut.`rsrc2 = 5;
36. dut.`rdst = 0;
37. #10;
38. $display("OP:ADD Rsrc1:%0d Rsrc2:%0d Rdst:%0d",dut.GPR[4], dut.GPR[5], dut.GPR[0] );
39. $display("-----");
40.
41. ///////////////immediate mov op
42. dut.IR = 0;
43. dut.`imm_mode = 1;
44. dut.`oper_type = 1;
45. dut.`rdst = 4;///gpr[4]
46. dut.`isrc = 55;
47. #10;
48. $display("OP:MOVI Rdst:%0d imm_data:%0d",dut.GPR[4],dut.`isrc );
49. $display("-----");
50.
51. ///////////////register mov
52. dut.IR = 0;
53. dut.`imm_mode = 0;
54. dut.`oper_type = 1;
55. dut.`rdst = 4;
56. dut.`rsrc1 = 7;///gpr[7]
57. #10;
58. $display("OP:MOV Rdst:%0d Rsrc1:%0d",dut.GPR[4],dut.GPR[7] );
59. $display("-----");
60.
61.

```

```

62.
63.
64.
65. //////////////////////////////////////////////////logical and imm
66. dut.IR = 0;
67. dut.`imm_mode = 1;
68. dut.`oper_type = 6;
69. dut.`rdst = 4;
70. dut.`rsrc1 = 7;//gpr[7]
71. dut.`isrc = 56;
72. #10;
73. $display("OP:ANDI Rdst:%8b Rsrc1:%8b imm_d :%8b",dut.GPR[4],dut.GPR[7],dut.`isrc );
74. $display("-----");
75.
76. //////////////////////////////////////////////////logical xor imm
77. dut.IR = 0;
78. dut.`imm_mode = 1;
79. dut.`oper_type = 7;
80. dut.`rdst = 4;
81. dut.`rsrc1 = 7;//gpr[7]
82. dut.`isrc = 56;
83. #10;
84. $display("OP:XORI Rdst:%8b Rsrc1:%8b imm_d :%8b",dut.GPR[4],dut.GPR[7],dut.`isrc );
85. $display("-----");
86.
87. ////////////////////////////////////////////////// zero flag
88. dut.IR = 0;
89. dut.GPR[0] = 0;
90. dut.GPR[1] = 0;
91. dut.`imm_mode = 0;
92. dut.`rsrc1 = 0;//gpr[0]
93. dut.`rsrc2 = 1;//gpr[1]
94. dut.`oper_type = 2;
95. dut.`rdst = 2;
96. #10;
97. $display("OP:Zero Rsrc1:%0d Rsrc2:%0d Rdst:%0d",dut.GPR[0], dut.GPR[1], dut.GPR[2] );
98. $display("-----");
99.
100. //////////////////////////////////////////////////sign flag
101. dut.IR = 0;
102. dut.GPR[0] = 16'h8000; //1000_0000_0000_0000
103. dut.GPR[1] = 0;
104. dut.`imm_mode = 0;
105. dut.`rsrc1 = 0;//gpr[0]
106. dut.`rsrc2 = 1;//gpr[1]
107. dut.`oper_type = 2;
108. dut.`rdst = 2;
109. #10;
110. $display("OP:Sign Rsrc1:%0d Rsrc2:%0d Rdst:%0d",dut.GPR[0], dut.GPR[1], dut.GPR[2] );
111. $display("-----");
112.
113. //////////////////////////////////////////////////carry flag
114. dut.IR = 0;
115. dut.GPR[0] = 16'h8000; //1000_0000_0000_0000 <0
116. dut.GPR[1] = 16'h8002; //1000_0000_0000_0010 <0
117. dut.`imm_mode = 0;
118. dut.`rsrc1 = 0;//gpr[0]
119. dut.`rsrc2 = 1;//gpr[1]
120. dut.`oper_type = 2;
121. dut.`rdst = 2; //0000_0000_0000_0010 >0
122. #10;

```

```
123.
124.     $display("OP:Carry & Overflow Rsrc1:%0d  Rsrc2:%0d Rdst:%0d",dut.GPR[0], dut.GPR[1],
    dut.GPR[2] );
125.     $display("-----");
126.
127.     #20;
128.     $finish;
129.     end
130.
131.     endmodule
```