

```

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. ///////////////logical and imm

```

```

62. dut.IR = 0;
63. dut.`imm_mode = 1;
64. dut.`oper_type = 6;
65. dut.`rdst = 4;
66. dut.`rsrc1 = 7;//gpr[7]
67. dut.`isrc = 56;
68. #10;
69. $display("OP:ANDI Rdst:%8b Rsrc1:%8b imm_d :%8b",dut.GPR[4],dut.GPR[7],dut.`isrc );
70. $display("-----");
71.
72. //////////////////////////////////logical or imm
73. dut.IR = 0;
74. dut.`imm_mode = 1;
75. dut.`oper_type = 7;
76. dut.`rdst = 4;
77. dut.`rsrc1 = 7;//gpr[7]
78. dut.`isrc = 56;
79. #10;
80. $display("OP:XORI Rdst:%8b Rsrc1:%8b imm_d :%8b",dut.GPR[4],dut.GPR[7],dut.`isrc );
81. $display("-----");
82.
83.
84. end
85.
86. endmodule

```