```verilog
1.  `timescale 1ns / 1ps
2.
3.  ///////////fields of IR
4.  `define oper_type IR[31:27]
5.  `define rdst      IR[26:22]
6.  `define rsrc1     IR[21:17]
7.  `define imm_mode  IR[16]
8.  `define rsrc2     IR[15:11]
9.  `define isrc      IR[15:0]
10.
11.
12. /////////////////arithmetic operation
13. `define movsgpr        5'b00000
14. `define mov            5'b00001
15. `define add            5'b00010
16. `define sub            5'b00011
17. `define mul            5'b00100
18.
19.
20.
21.
22. module top();
23.
24.
25.
26.
27.
28.
29. reg [31:0] IR;           ////// instruction register  <--ir[31:27]--><--ir[26:22]--><--
    ir[21:17]--><--ir[16]--><--ir[15:11]--><--ir[10:0]-->
30.                          //////fields               <---  oper  --><--   rdest --><--
    rsrc1 --><--modesel--><--  rsrc2 --><--unused  -->
31.                          //////fields               <---  oper  --><--   rdest --><--
    rsrc1 --><--modesel--><--  immediate_date     -->
32.
33. reg [15:0] GPR [31:0] ;   ///////general purpose register gpr[0] ....... gpr[31]
34.
35.
36.
37. reg [15:0] SGPR ;      ///// msb of multiplication --> special register
38.
39. reg [31:0] mul_res;
40.
41.
42.
43. always@(*)
44. begin
45. case(`oper_type)
46. /////////////////////////////////
47. `movsgpr: begin
48.
49.    GPR[`rdst] = SGPR;
50.
51. end
52.
53. /////////////////////////////////
54. `mov : begin
55.    if(`imm_mode)
56.        GPR[`rdst]  = `isrc;
57.    else
58.        GPR[`rdst]   = GPR[`rsrc1];
```

```verilog
59. end
60.
61. //////////////////////////////////////////////////
62.
63. `add : begin
64.      if(`imm_mode)
65.          GPR[`rdst]   = GPR[`rsrc1] + `isrc;
66.      else
67.          GPR[`rdst]   = GPR[`rsrc1] + GPR[`rsrc2];
68. end
69.
70. /////////////////////////////////////////////////////
71.
72. `sub : begin
73.      if(`imm_mode)
74.          GPR[`rdst]  = GPR[`rsrc1] - `isrc;
75.      else
76.          GPR[`rdst]   = GPR[`rsrc1] - GPR[`rsrc2];
77. end
78.
79. //////////////////////////////////////////////////////////
80.
81. `mul : begin
82.      if(`imm_mode)
83.          mul_res   = GPR[`rsrc1] * `isrc;
84.      else
85.          mul_res   = GPR[`rsrc1] * GPR[`rsrc2];
86.
87.      GPR[`rdst]   =  mul_res[15:0];
88.      SGPR         =  mul_res[31:16];
89. end
90.
91. ////////////////////////////////////////////////////////
92. endcase
93. end
94. endmodule
95.
96. /////////////////////////////////////////////////////////////////////
```