```
 1  module control_unit( input[2:0] opcode,
 2                       input reset,
 3                       output reg reg_dst,mem_to_reg,
 4                       output reg jump,branch,mem_read,mem_write,alu_src,reg_write

 5   );
 6   always @(*)
 7   begin
 8        if(reset == 1'b1) begin
 9                  reg_dst = 1'b0;  //0
10                  mem_to_reg = 1'b0;
11                  jump = 1'b0;
12                  branch = 1'b0;
13                  mem_read = 1'b0;
14                  mem_write = 1'b0;
15                  alu_src = 1'b0;
16                  reg_write = 1'b0;
17        end
18        else begin
19        case(opcode)
20      3'b000: begin  reg_dst = 1'b1;  // R-type
21                     mem_to_reg = 1'b0;
22                     jump = 1'b0;
23                     branch = 1'b0;
24                     mem_read = 1'b0;
25                     mem_write = 1'b0;
26                     alu_src = 1'b0;
27                     reg_write = 1'b1;  // 28/7
28              end
29      3'b010: begin    // LB //0
30                     reg_dst = 1'b0;
31                     mem_to_reg = 1'b1;
32                     jump = 1'b0;
33                     branch = 1'b0;
34                     mem_read = 1'b1;
35                     mem_write = 1'b0;
36                     alu_src = 1'b1;
37                     reg_write = 1'b1;
38              end
39      3'b011: begin    // SB
40                     reg_dst = 1'b0;
41                     mem_to_reg = 1'b0;
42                     jump = 1'b0;
43                     branch = 1'b0;
44                     mem_read = 1'b0;
45                     mem_write = 1'b1;
46                     alu_src = 1'b1;
47                     reg_write = 1'b0;
48              end
49      3'b001: begin    // Addi
50                     reg_dst = 1'b0;
51                     mem_to_reg = 1'b0;
52                     jump = 1'b0;
53                     branch = 1'b0;
54                     mem_read = 1'b0;
55                     mem_write = 1'b0;
56                     alu_src = 1'b1;
57                     reg_write = 1'b1;
58                      //
59              end
60     3'b100:  begin    //beq //0
61                     reg_dst = 1'b0;
```

```
62                      mem_to_reg = 1'b0;
63                      jump = 1'b0;
64                      branch = 1'b1;
65                      mem_read = 1'b0;
66                      mem_write = 1'b0;
67                      alu_src = 1'b0;
68                      reg_write = 1'b0;
69              end
70      3'b101: begin    //jump
71                      reg_dst = 1'b0;
72                      mem_to_reg = 1'b0;
73                      jump = 1'b1;
74                      branch = 1'b0;
75                      mem_read = 1'b0;
76                      mem_write = 1'b0;
77                      alu_src = 1'b0;
78                      reg_write = 1'b0;
79              end
80        default: begin
81                      reg_dst = 1'b1;
82                      mem_to_reg = 1'b0;
83                      jump = 1'b0;
84                      branch = 1'b0;
85                      mem_read = 1'b0;
86                      mem_write = 1'b0;
87                      alu_src = 1'b0;
88                      reg_write = 1'b1;
89                  end
90      endcase
91      end
92  end
93  endmodule
94
```