```
1   //
2   // Part 4
3   // Basic Sequential Logic Lab
4   // Neil Nie, (c) Nov 7th, 2018
5   //
6
7   module gated_d_latch(
8      clk,
9      D,
10     Q
11  );
12
13  input clk, D;
14  output reg Q;
15
16  always @ (D, clk)
17     if (clk)
18        Q = D;
19
20  endmodule
21
22  module ped_flip_flop(
23     clk,
24     D,
25     Q
26  );
27
28  input D, clk;
29  output Q;
30
31  wire Qm, Qs;
32
33  gated_d_latch(~clk, D, Qm);
34  gated_d_latch(clk, Qm, Qs);
35
36  assign Q = Qs;
37
38  endmodule
39
40  module ned_flip_flop(
41     clk,
42     D,
43     Q
44  );
45
46  input D, clk;
47  output Q;
48
49  wire Qm, Qs;
50
51  gated_d_latch(clk, D, Qm);
52  gated_d_latch(~clk, Qm, Qs);
53
54  assign Q = Qs;
55
56  endmodule
57
58  module Part4(
59     SW,
60     LEDR
61  );
62
63  input [1:0] SW;
64  output [2:0] LEDR;
65
66  gated_d_latch(SW[0], SW[1], LEDR[0]);
67  ped_flip_flop(SW[0], SW[1], LEDR[1]);
68  ned_flip_flop(SW[0], SW[1], LEDR[2]);
69
70  endmodule
71
```