```verilog
//
// Part 4
// Basic Sequential Logic Lab
// Neil Nie, (c) Nov 7th, 2018
//

module gated_d_latch(
   clk,
   D,
   Q
);

input clk, D;
output reg Q;

always @ (D, clk)
   if (clk)
      Q = D;

endmodule

module ped_flip_flop(
   clk,
   D,
   Q
);

input D, clk;
output Q;

wire Qm, Qs;

gated_d_latch(~clk, D, Qm);
gated_d_latch(clk, Qm, Qs);

assign Q = Qs;

endmodule

module ned_flip_flop(
   clk,
   D,
   Q
);

input D, clk;
output Q;

wire Qm, Qs;

gated_d_latch(clk, D, Qm);
gated_d_latch(~clk, Qm, Qs);

assign Q = Qs;

endmodule

module Part4(
   SW,
   LEDR
);

input [1:0] SW;
output [2:0] LEDR;

gated_d_latch(SW[0], SW[1], LEDR[0]);
ped_flip_flop(SW[0], SW[1], LEDR[1]);
ned_flip_flop(SW[0], SW[1], LEDR[2]);

endmodule

```