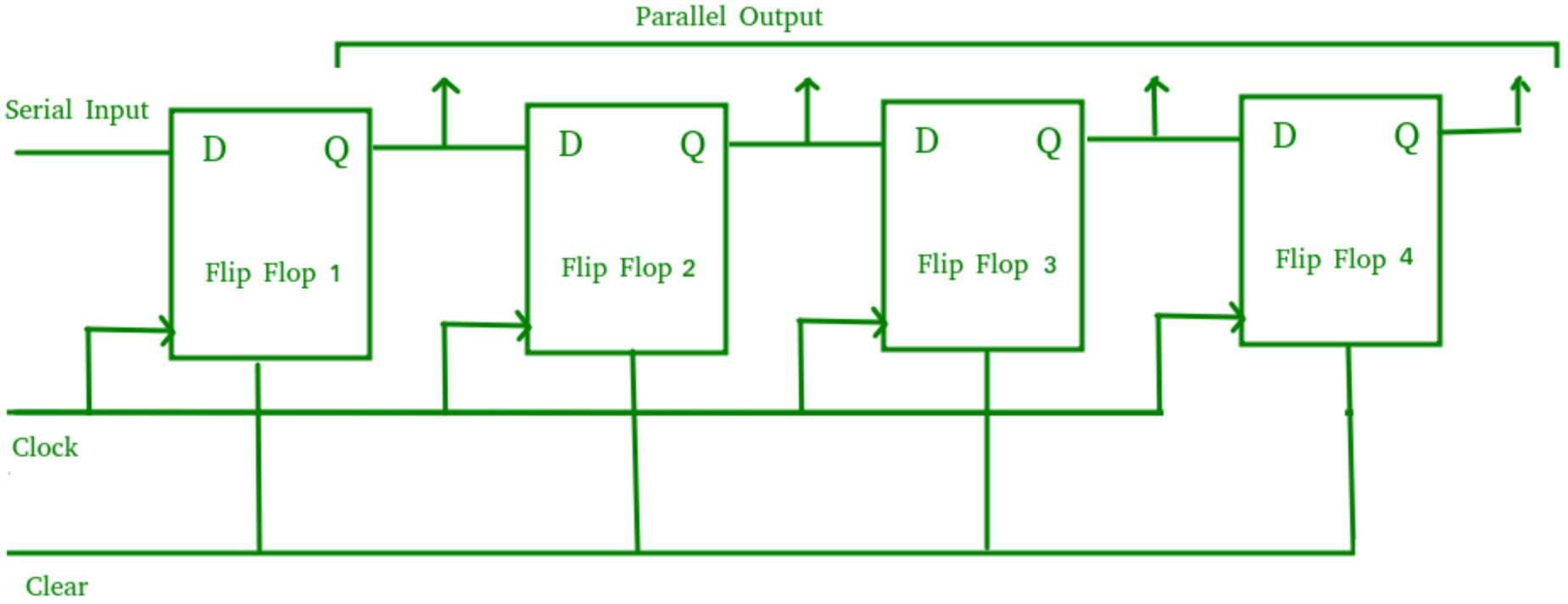
# Lab03 Code Review

# 參考架構 (SIPO Shift register)



# Reference code

```
//---------------------------------------------------------------------
//   LOGIC DECLARATION
//---------------------------------------------------------------------
logic [2:0] cnt;
logic [3:0] shift_r;
```

```
always_ff @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        shift_r <= 4'b0;
    else if(in_valid)
        shift_r <= {shift_r[3:0], s_in};
    else
        shift_r <= 0;
end

always_comb begin
    if(in_valid && cnt < 4)
        cnt_next = cnt + 1;
    else
        cnt_next = 0;
end

always_ff @(posedge clk or negedge rst_n)begin
    if(!rst_n)
        cnt <= 0;
    else if(in_valid)
        cnt <= cnt_next;
    else
        cnt <= 0;
end

assign out_valid = (cnt==4) ? 1 : 0;
assign p_out = shift_r;
```

# Bad coding example

- Combinational loop

```
always_comb begin
    if (!in_valid) count_new = 0;
    else count_new = count_new + 1;
  if(count == 3) begin
    out_new = 1;
  end
  else out_new = 0;
end
```

# Bad coding example

- Separate combinational and sequential blocks

```
always @(posedge clk, negedge rst_n)
begin
    if(rst_n == 0)
        begin
            out_valid<=0;
            p_out<=0;
            c<=0;
        end
    else
    begin
        if(in_valid)
        begin
            c <=c+1;
            p_out <= (p_out *2) +s_in;
            if(c >= 3)
            begin
            out_valid <=1;
            c <=0;
            end
            else
            out_valid <=0;
        end
        else
            begin
            out_valid <=0;
            p_out <=0;
            end
    end
end
```