Assignment_4_Extended

(under supervision of Eng/Karim Wassem)

Question1]

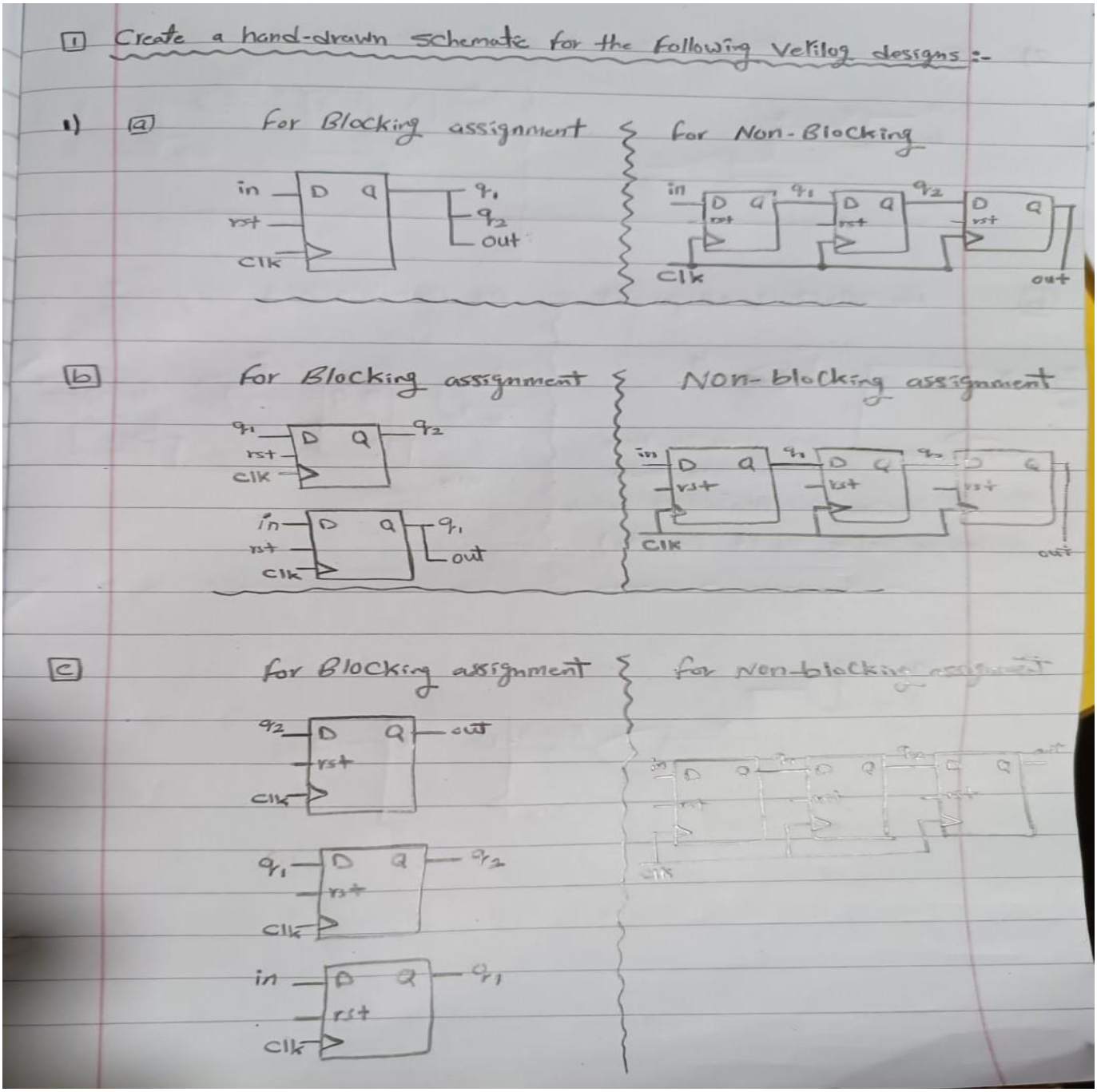

Figure (1) : question1 “part1”

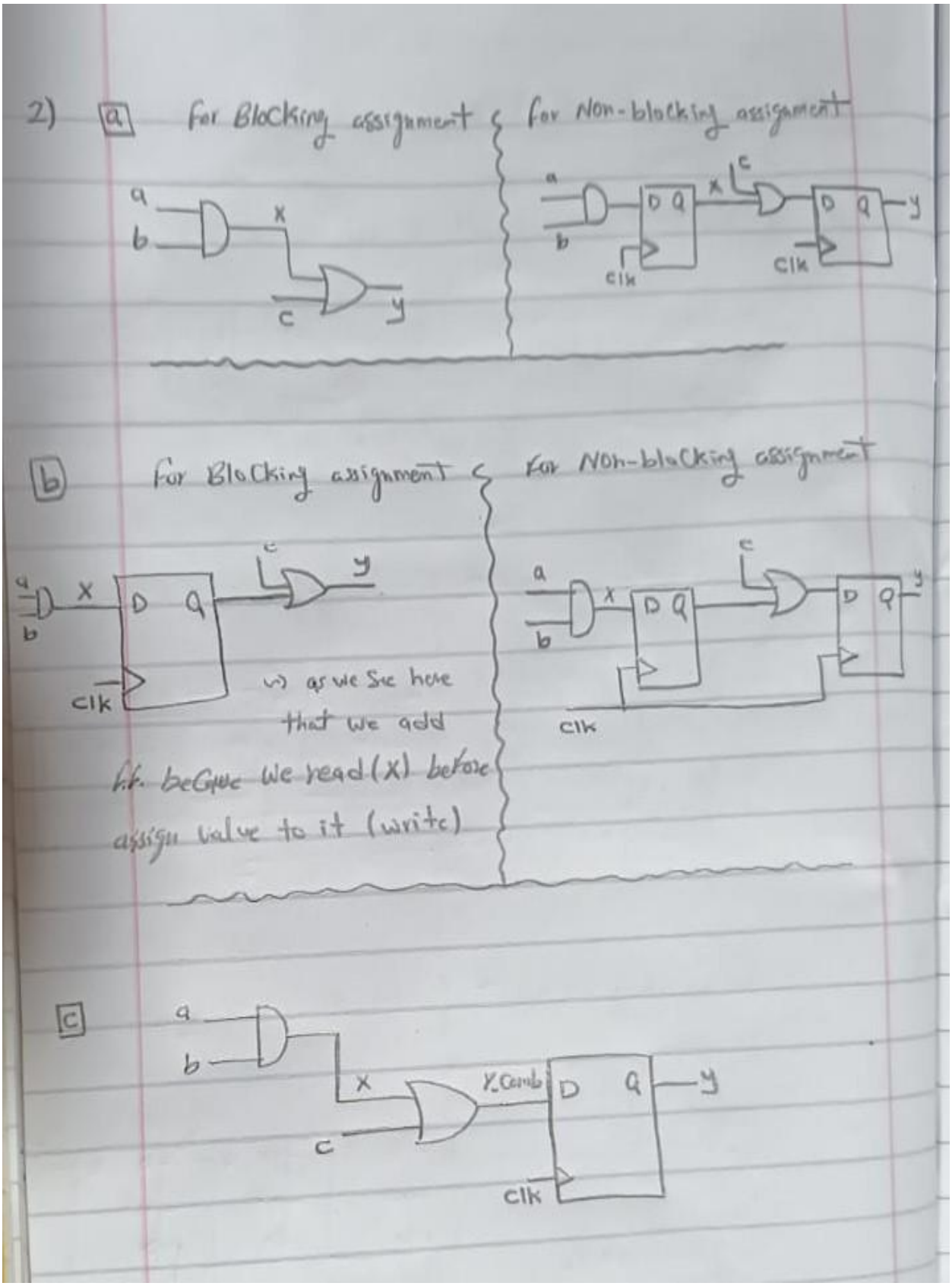

Figure (2) : question1 “part2”

Question2]

```
1   module q2_ext(clk,rst,load,load_value,po);
2
3   parameter SHIFT_DIRECTION="LEFT";
4   parameter SHIFT_AMOUNT=1;
5   input clk,rst,load;
6   input [7:0] load_value;
7   output reg [7:0] po;
8
9   always @(posedge clk or posedge rst) begin
10
11      if(rst)
12      po<=0;
13      else if(load)
14      po<=load_value;
15      else begin
16          case(SHIFT_DIRECTION)
17          "LEFT": begin
18              po<=po<<SHIFT_AMOUNT;
19          end
20          "RIGHT": begin
21              po<=po>>SHIFT_AMOUNT;
22          end
23
24          endcase
25      end
26  end
27
28  endmodule
```

Figure(3) : design code

```
1  module q2_ext_tb1();
2
3  parameter N1="LEFT";
4  parameter N2=1;
5  reg clk,rst,load;
6  reg [7:0] load_value;
7  wire [7:0] po;
8  integer i=0;
9
10 q2_ext #(.SHIFT_DIRECTION(N1),.SHIFT_AMOUNT(N2)) DUT(.clk(clk),.rst(rst),.load(load),.load_value(load_value),.po(po));
11
12 initial begin
13     clk=0;
14     forever
15     #25 clk=~clk;
16 end
17
18 initial begin
19     rst=1;
20     #100;
21     rst=0;
22     for(i=0;i<20;i=i+1) begin
23         load=$random;
24         load_value=$random;
25             #50;
26     end
27
28     $stop;
29 end
30
31 endmodule
```

Figure (4) : first testbench with shift left (1 bit)

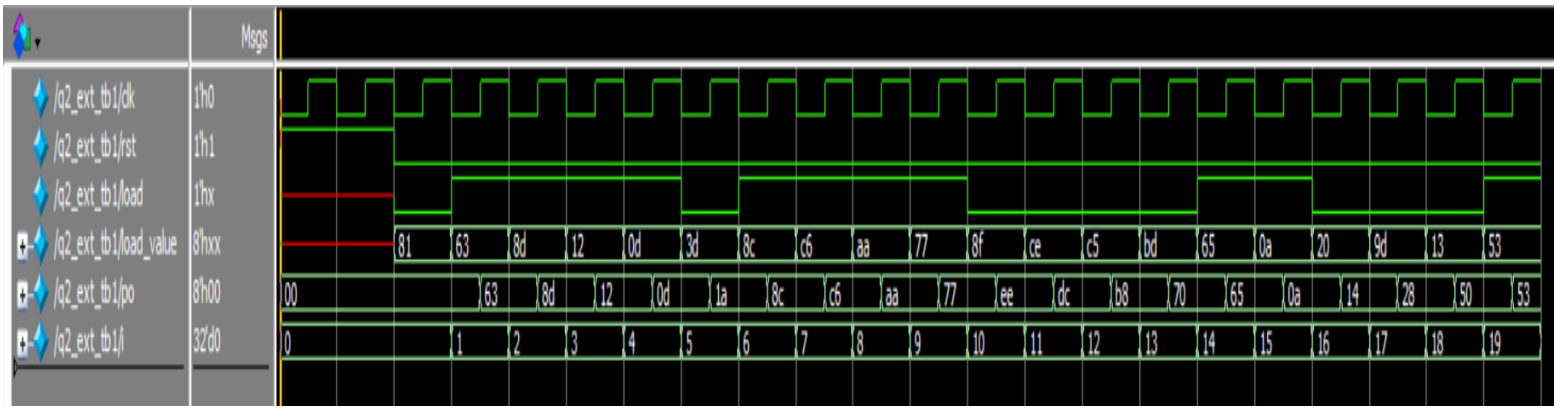

Figure (5) : waveform snippet for the above test bench (shift left)

```
1  module q2_ext_tb2();
2
3  parameter N3="RIGHT";
4  parameter N4=2;
5  reg clk,rst,load;
6  reg [7:0] load_value;
7  wire [7:0] po;
8  integer i=0;
9
10 q2_ext #(.SHIFT_DIRECTION(N3),.SHIFT_AMOUNT(N4)) DUT(.clk(clk),.rst(rst),.load(load),.load_value(load_value),.po(po));
11
12 initial begin
13     clk=0;
14     forever
15     #25 clk=~clk;
16 end
17
18 initial begin
19     rst=1;
20     #100;
21     rst=0;
22     for(i=0;i<20;i=i+1) begin
23         load=$random;
24         load_value=$random;
25             #50;
26     end
27
28     $stop;
29 end
30
31 endmodule
```

Figure (6) : second testbench with shift right (2 bits)

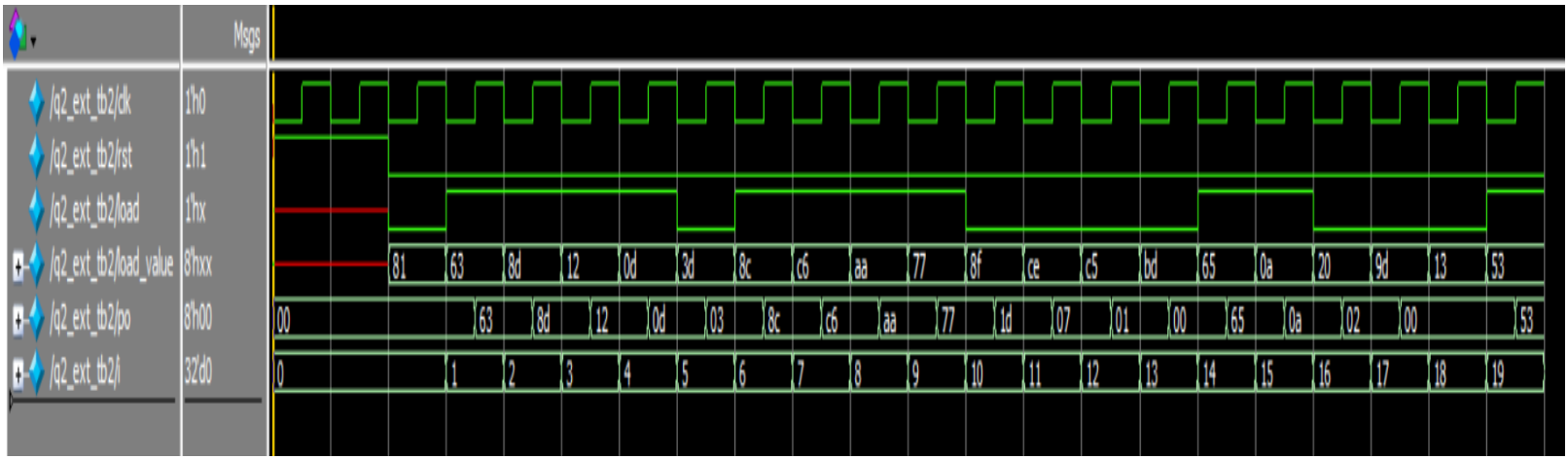

Figure (7) : waveform snippet for the above test bench (shift right)

Question3]

```
module q3_ext(clk,rst,gray_out);

input clk,rst;

output [1:0] gray_out;

reg [1:0] bin_out;

always @(posedge clk or posedge rst) begin

    if(rst)
    bin_out<=0;
    else
    bin_out<=bin_out+1;

end

assign gray_out[0]=^bin_out;

assign gray_out[1]=bin_out[1];

endmodule
```

Figure (8) : design code

```
module q3_ext_tb();

reg clk,rst;
wire [1:0] gray_out;

q3_ext DUT(.clk(clk),.rst(rst),.gray_out(gray_out));

initial begin
    clk=0;
    forever
    #25 clk=~clk;
end

initial begin

    rst=1;
    #100;
    rst=0;
    #500;
      $stop;
end

initial
$monitor("the gray_counter =%b",gray_out);


endmodule
```

Figure (9) : testbench code

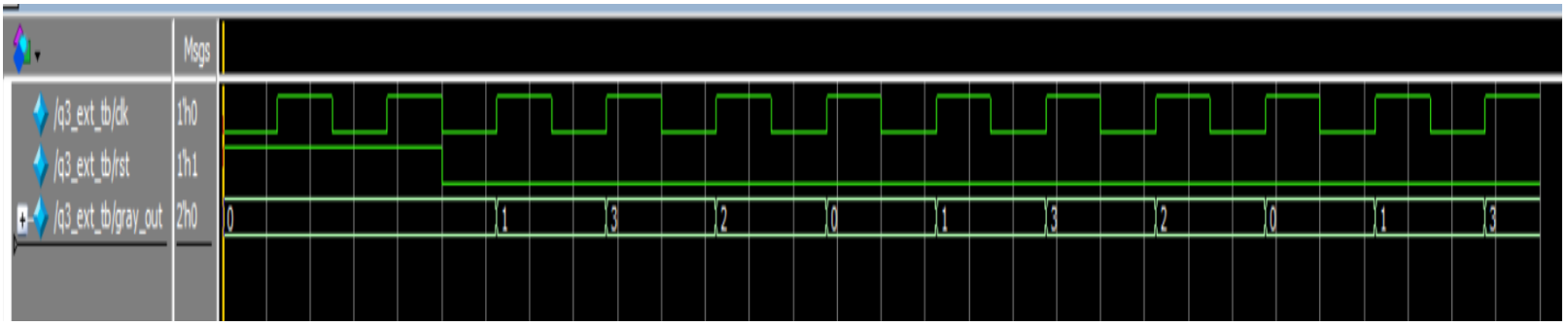

Figure (10) : waveform snippet