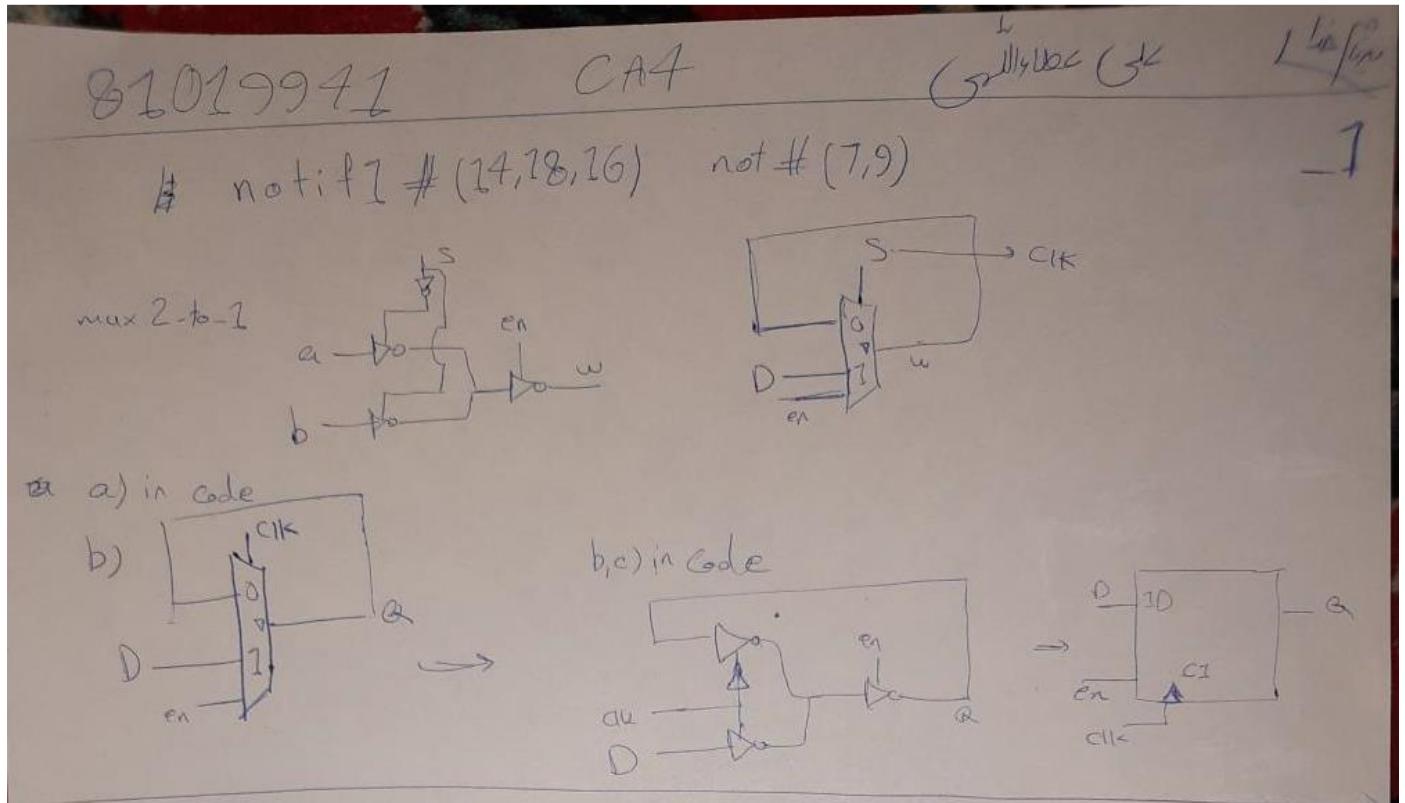


به نام خدا  
على عطا الله  
810199461

1:



```
`timescale 1ns/1ns

module mux2_to_1 (input a,b,s,en,output w);
    wire i,j;
    notif1 #(14,18,16) n1(i,a,j);
    notif1 #(14,18,16) n2(i,b,s);
    not #(7,9) n3(j,s);
    notif1 #(14,18,16) n4(w,i,en);
endmodule

module D_latch(input d,clk,en,output q);
    mux2_to_1 mx(q,d,clk,en,q);
endmodule
```

```
`timescale 1ns/1ns

module CA4_E1_TB_1 ();
    logic dd=1,c=1,enn=1;
```

```

wire qq;

D_latch xx(dd,c,enn,qq);

always #200 c=~c ;

initial begin
    #100 dd=0;
    #200 dd=1;
    #200 $stop;
end

endmodule

module CA4_E1_TB_2 ();

    logic dd=1,c=1,enn=1;

    wire qq;

    D_latch xx(dd,c,enn,qq);

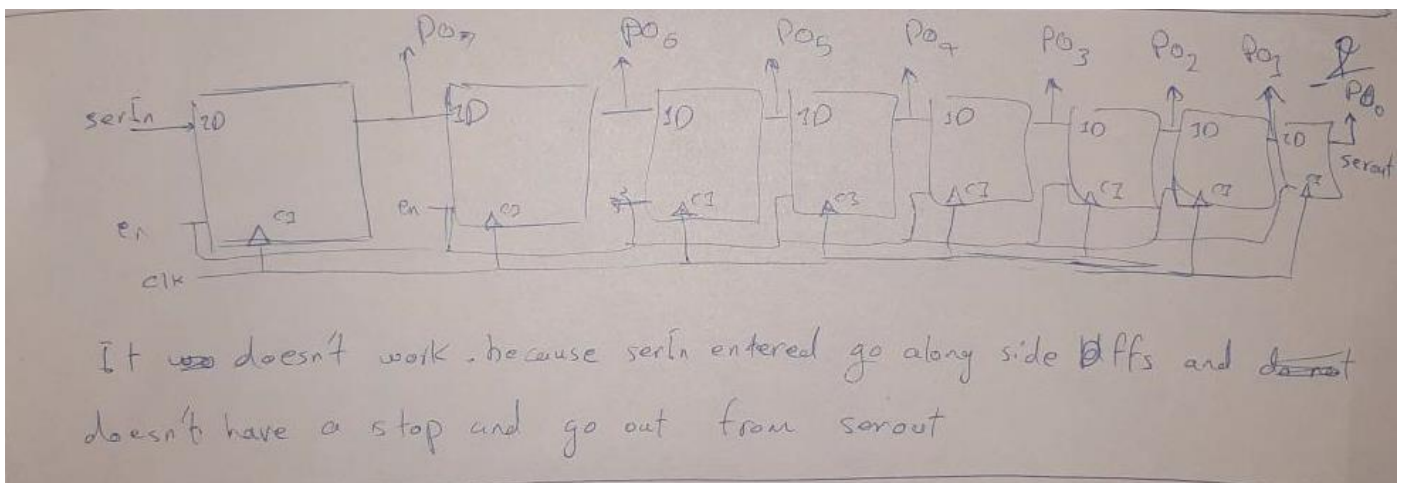
    always #200 c=~c ;

    initial begin
        #200 dd=0;
        #200 dd=1;
        #200 $stop;
    end

end
endmodule

```

2:



```
`timescale 1ns/1ns
```

```

module mux2_to_1 (input a,b,s,en,output w);

    wire i,j;

```

```

        notif1 #(14,18,16) n1(i,a,j);

        notif1 #(14,18,16) n2(i,b,s);

        not #(7,9) n3(j,s);

        notif1 #(14,18,16) n4(w,i,en);

endmodule

module D_latch(input d,clk,en,output q);

    mux2_to_1 mx(q,d,clk,en,q);

endmodule

module register8withDlatch(input serIn,clk,en,output [7:0] po);

    wire [8:0] d;

    assign d[8]=serIn;

    genvar k;

    generate

        for(k=8;k>0;k=k-1) begin : registerGates

            D_latch dd(d[k],clk,en,d[k-1]);

        end

    endgenerate

    assign po=d[7:0];

endmodule

```

```

`timescale 1ns/1ns

module CA4_E2_TB ();

    logic serr=0,c=1,enn=1;

    wire[7:0] ppoo;

    register8withDlatch xx(serr,c,enn,ppoo);

    always #200 c=~c ;

    initial begin

        #1050 serr=1;

        #200 serr=0;

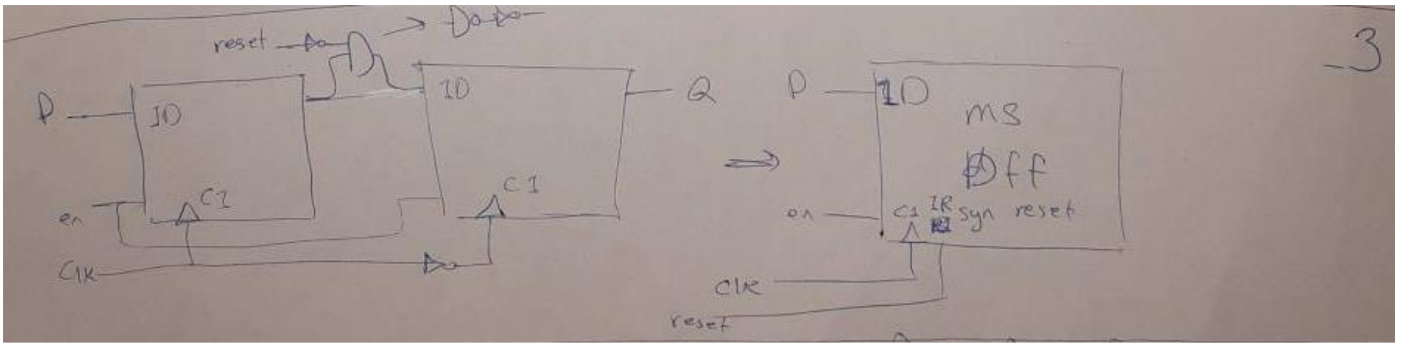
        #1000 $stop;

    end

endmodule

```

3:



```
`timescale 1ns/1ns
```

```
module mux2_to_1 (input a,b,s,en,output w);
```

```
    wire i,j;
```

```
    notif1 #(14,18,16) n1(i,a,j);
```

```
    notif1 #(14,18,16) n2(i,b,s);
```

```
    not #(7,9) n3(j,s);
```

```
    notif1 #(14,18,16) n4(w,i,en);
```

```
endmodule
```

```
module D_latch(input d,clk,en,output q);
```

```
    mux2_to_1 mx(q,d,clk,en,q);
```

```
endmodule
```

```
module ms_DFF(input d,clk,reset,en,output q);
```

```
    wire i,j,k,d2,q1;
```

```
    not #(7,9) n1(i,reset);
```

```
    not #(7,9) n2(j,clk);
```

```
    nand #(14,10) o1(k,i,q1);
```

```
    not #(7,9) n3(d2,k);
```

```
    D_latch dl1(d,clk,en,q1);
```

```
    D_latch dl2(d2,j,en,q);
```

```
endmodule
```

```
`timescale 1ns/1ns
```

```
module CA4_E3_TB_1 ();
```

```
    logic dd=1,c=1,enn=1,reset=0;
```

```
    wire qq;
```

```
    ms_DFF xx(dd,c,reset,enn,qq);
```

```
    always #300 c=~c ;
```

```
    initial begin
```

```

#1000

#1000 dd=0;

#2000 dd=1;

#1900 resett=1;

#2000 $stop;

end

endmodule

module CA4_E3_TB_2 ();

    logic dd=1,c=1,enn=1,resett=0;

    wire qq;

    ms_DFF xx(dd,c,resett,enn,qq);

    always #500 c=~c ;

    initial begin

        #1000

        #1000 dd=0;

        #2000 dd=1;

        #2000 resett=1;

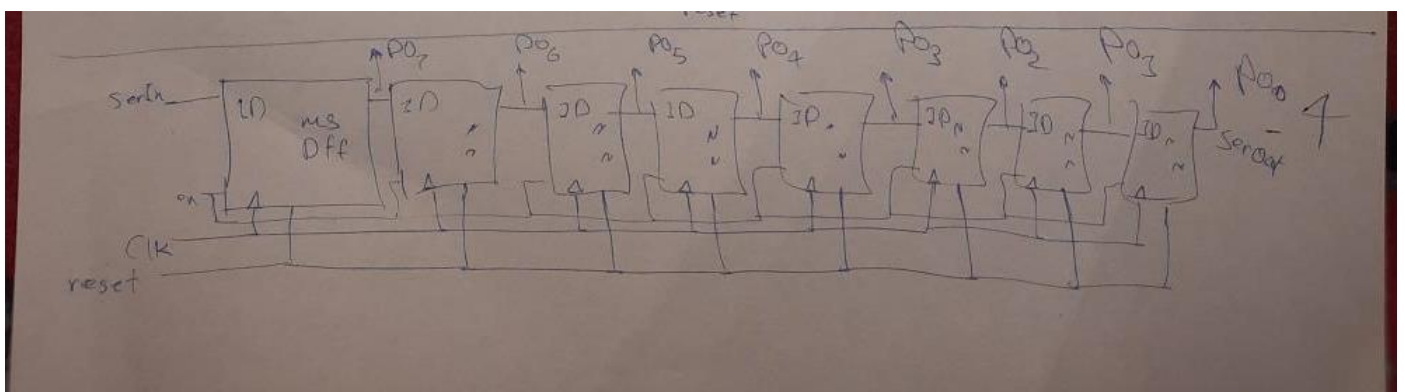
        #2000 $stop;

    end

endmodule

```

4:



a) it works and problem 1 doesn't work because it has stop and serIn can't go along side registers and it goes step by step with clock.

b) in code

```

`timescale 1ns/1ns

module mux2_to_1 (input a,b,s,en,output w);

    wire i,j;

    notif1 #(14,18,16) n1(i,a,j);

    notif1 #(14,18,16) n2(i,b,s);

    not #(7,9) n3(j,s);

    notif1 #(14,18,16) n4(w,i,en);

endmodule

module D_latch(input d,clk,en,output q);

    mux2_to_1 mx(q,d,clk,en,q);

endmodule

module ms_DFF(input d,clk,reset,en,output q);

    wire i,j,k,d2,q1;

    not #(7,9) n1(i,reset);

    not #(7,9) n2(j,clk);

    nand #(14,10) o1(k,i,q1);

    not #(7,9) n3(d2,k);

    D_latch dl1(d,clk,en,q1);

    D_latch dl2(d2,j,en,q);

endmodule

module register8withMSDFF(input serIn,clk,reset,en,output [7:0] po);

    wire [8:0] d;

    assign d[8]=serIn;

    genvar k;

    generate

        for(k=8;k>0;k=k-1) begin : registerGates

            ms_DFF dd(d[k],clk,reset,en,d[k-1]);

        end

    endgenerate

    assign po=d[7:0];

endmodule

module register8withAlways(input serIn,clk,reset,en,output logic [7:0] po);

    always @(posedge clk) begin

        if(reset)

```

```

                #50 po<=8'b0;

            else

                #50 po<={serIn,po[7:1]};

        end

endmodule

```

```

`timescale 1ns/1ns

module CA4_E4_TB ();

    logic serr=0,c=1,enn=1,reset=0;

    wire[7:0] ppoo1,ppoo2;

    register8withMSDFF xx1(serr,c,reset,enn,ppoo1);
    register8withAlways xx2(serr,c,reset,enn,ppoo2);

    always #100 c=~c ;

    initial begin

        #3000

        #50 serr=1;

        #200 serr=0;

        #200 serr=0;

        #200 serr=1;

        #200 serr=1;

        #200 serr=0;

        #200 serr=0;

        #200 serr=1;

        #200 serr=0;

        #1000 $stop;

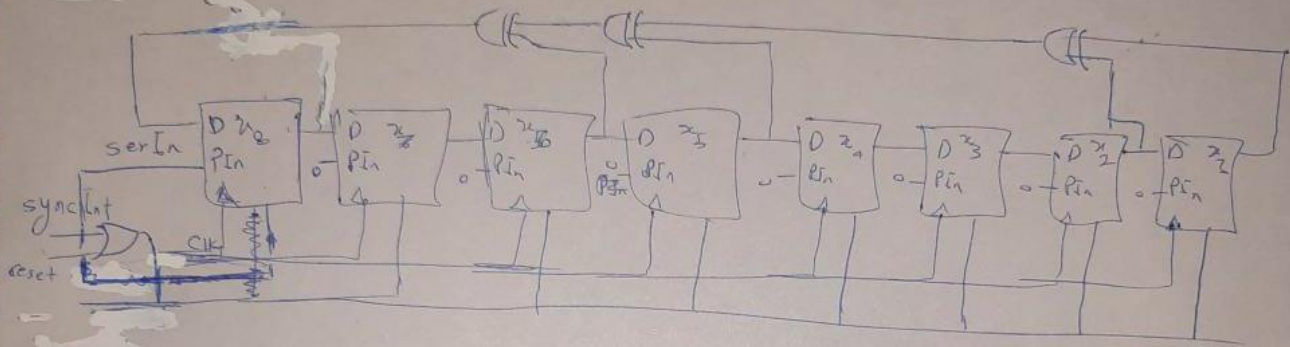
    end

endmodule

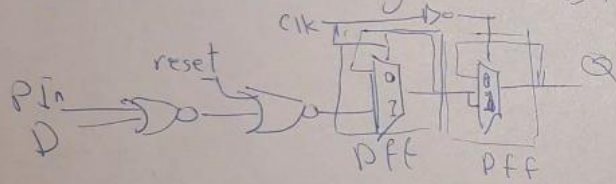
```

5:

$$a) x^8 + x^6 + x^5 + x^2 + x + 1$$



We need to design new DFF that has - PIn



PIn is syncInt

b) because of active reset, all parout are 0 and if we inactivated reset & and clock active, the ~~output~~ output is again 0 because xor of 2 bit zero produces again zero

c) syncInt actived and parout changes constantly

d) with every pulse, we shift 1 bit and serIn is xor of 0,2,5,6 and it repeats by every clock

```
`timescale 1ns/1ns
```

```
module mux2_to_1 (input a,b,s,en,output w);
```

```
    wire i,j;
```

```
    notif1 #(14,18,16) n1(i,a,j);
```

```
    notif1 #(14,18,16) n2(i,b,s);
```

```
    not #(7,9) n3(j,s);
```

```
    notif1 #(14,18,16) n4(w,i,en);
```

```
endmodule
```

```
module D_latch(input d,clk,en,output q);
```

```
    mux2_to_1 mx(q,d,clk,en,q);
```

```
endmodule
```

```
module new_ms_DFF(input d,clk,syncInt,reset,en,output q);
```

```
    wire i,j,clkbar,d2,q1;
```



```

    not #(7,9) n1(clkbar,clk);

    nor #(8,18) n2(i,d,reset);

    nor #(8,18) n3(j,i,syncInt);

    D_latch dl1(j,clk,en,q1);

    D_latch dl2(q1,clkbar,en,q);

endmodule

module new_register8withMSDFF(input clk,syncInt,reset,en,output [7:0] po);
    wire [8:0] d,preset;
    wire reset_;
    assign d[8]=d[0];
    genvar k;
    generate
        for(k=8;k>0;k=k-1) begin : registerGates
            new_ms_DFF dd(d[k],clk,preset[k],reset_,en,d[k-1]);
        end
    endgenerate
    assign po=d[7:0];
    assign reset_ = (reset|syncInt);
    assign preset= 8'b10000000 ? syncInt : 8'b0;
endmodule

module LFSR(input clk,syncInt,reset,en,output [7:0] po);
    wire [8:0] d,preset;
    wire reset_;
    assign d[8]=d[0];
    genvar k;
    generate
        for(k=8;k>0;k=k-1) begin : registerGates
            new_ms_DFF dd(d[k],clk,preset[k],reset_,en,d[k-1]);
        end
    endgenerate
    assign po=d[7:0];
    assign reset_ = (reset|syncInt);
    assign preset= 8'b10000000 ? syncInt : 8'b0;
endmodule

```

```
`timescale 1ns/1ns

module CA4_E5_TB ();

    logic c=1, syncclnt=1, enn=1, resett=0;
    wire[7:0] ppoo1;
    new_register8withMSDFF xx1(c, syncclnt, resett, enn, ppoo1);
    always #100 c=~c ;

    initial begin
        #3000
        #1000 $stop;
    end
endmodule
```

