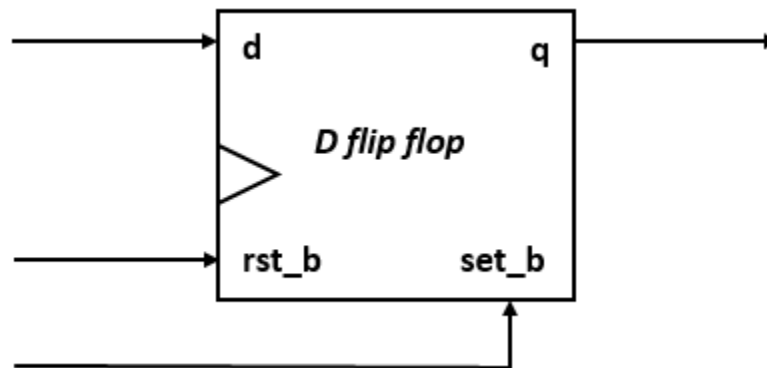


Laborator S10 AC

Arhitecturi de detecție a erorilor

P.10.1 Proiectați un bistabil de tip D cu o linie de set asincronă adițională, așa cum este descris în arhitectura de mai jos:



- Implementați, folosind limbajul Verilog, modulul ***dff_ar*** atașat design-ului dat.
- Redactați un testbench care verifică funcționalitatea modulului ***dff_ar*** folosind evaluarea intrărilor de tip non-exhaustiv.

```
// Soluție a):  
  
module dff_ar (  
    input d, rst_b, set_b, clk,  
    output reg q  
);  
  
always @(posedge clk, negedge rst_b) begin  
    if (set_b==0)  
        q <= 1'b1;  
    else  
        if (rst_b==0)  
            q <= 1'b0;  
        else q <= d;  
    end  
endmodule  
  
// Soluție b):  
  
module dff_ar_tb (  
    output reg d, rst_b, set_b, clk,  
    output q  
);  
  
initial begin  
    clk=1'b0;  
    repeat (4) #20 clk = ~clk;  
end  
  
initial begin  
    rst_b=1'b0;  
    #5 rst_b = 1'b1;  
end
```

```
initial begin
set_b=1'b1;
#60 set_b = 1'b0;
#10 set_b=1'b1;
end
initial begin
d=1'b0;
#40 d = 1'b1;
#20 d = 1'b0;
end
endmodule
```

P10.2. Proiectați un registru serial de deplasare la dreapta pe 4 biți, cu feedback, a cărui ieșiri sunt generate în tabelul de adevăr dat ca suport. Pentru aceasta se cer următoarele aspecte:

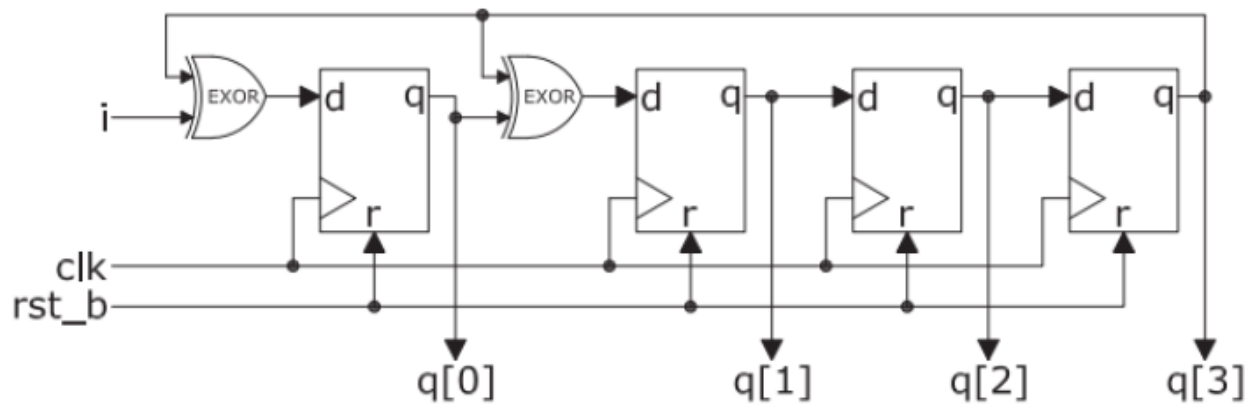
- Completați tabelul de adevăr cu vectorii pe 4 biți lipsă și determinați periodicitatea secvenței de ieșire.
- Implementați, folosind limbajul Verilog, unitatea Linear Feedback Shift Register (LFSR) pe 4 ranguri, utilizând instanțe ale modulului *dff_ar* proiectat la

P10.1

Soluția la a) este furnizată în materialul de laborator.

```
module LFSR (  
    input clk, rst_b, set_b,  
    output [3:0] q  
);  
    // Instantierea  
    dff_ar dff0 ( .d(q[3]),  
                  .clk(clk),  
                  .rst_b (1'b1),  
                  .set_b (set_b),  
                  .q (q[0]) );  
    dff_ar dff1 ( .d(q[3] ^ q[0]),  
                  .clk(clk),  
                  .rst_b (1'b1),  
                  .set_b (set_b),  
                  .q (q[1]) );  
    dff_ar dff2 ( .d(q[1]),  
                  .clk(clk),  
                  .rst_b (1'b1),  
                  .set_b (set_b),  
                  .q (q[2]) );  
    dff_ar dff3 ( .d(q[2]),  
                  .clk(clk),  
                  .rst_b (1'b1),  
                  .set_b (set_b),  
                  .q (q[3]) );  
endmodule
```

P.10.3. Implementați, utilizând limbajul Verilog, un Single Input Signature Register (SISR) pe 4 ranguri, așa cum este ilustrat mai jos.



```

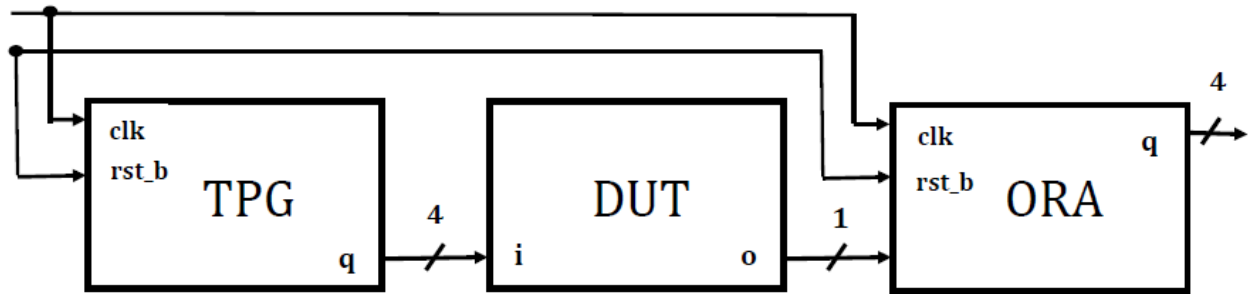
module SISR (
    input i, clk, rst_b, set_b,
    output [3:0] q
);
    // Instantierea
    dff_ar dff0 ( .d(i ^ q[3]),
                  .clk(clk),
                  .rst_b (rst_b),
                  .set_b (1'b1),
                  .q (q[0]) );
    dff_ar dff1 ( .d(q[3] ^ q[0]),
                  .clk(clk),
                  .rst_b (rst_b),
                  .set_b (1'b1),
                  .q (q[1]) );

```

```
dff_ar dff2 ( .d(q[1]),  
             .clk(clk),  
             .rst_b (rst_b),  
             .set_b (1'b1),  
             .q (q[2]) );  
dff_ar dff3 ( .d(q[2]),  
             .clk(clk),  
             .rst_b (rst_b),  
             .set_b (1'b1),  
             .q (q[3]) );  
endmodule
```

P.10.4. Proiectați, utilizând limbajul Verilog, o arhitectură Built-In Self-Test (BIST) care să folosească următoarele componente în structura sa:

- LFSR-ul pe 4 biți implementat la P10.2 va înlocui unitatea TPG.
- DUT-ul va fi substituit de un circuit combinațional care furnizează un bit de 1 dacă intrarea i a blocului este multiplu de 3, și 0 în caz contrar.
- SISR-ul pe 4 biți va substitui unitatea ORA din diagram dată.



```

module BIST (
    input clk, rst_b, set_b,
    output [3:0] q
);
    wire [3:0] w1;
    wire w2;
    // Instantierea
    LFSR TPG ( .clk(clk),
                .rst_b (1'b1),
                .set_b (set_b),
                .q (w1)
    );
    MUL_3 DUT ( .i (w2),
                .o (w1)
    );
    SSIR ORA ( .clk(clk),
                .rst_b (rst_b),
                .set_b (1'b1),
                .q (q)
    );
endmodule

```