

Design-ul unui înmulțitor secvențial pentru numere reprezentate în Semn-Marime

Probleme propuse

Oprîtoiu Flavius
flavius.opritoiu@cs.upt.ro

December 9, 2023

Problema 1

Implementați registrul Q din arhitectura înmulțitorului secvențial pentru numere reprezentate în Semn-Mărire.

Porniți de la tiparul de mai jos (accesibil [▶ aici](#)):

```
1  module reg_q(  
2      input clk, rst_b, clr_lsb, ld_ibus, ld_obus, sh_r,  
3      input sh_i, [7:0] ibus,  
4      output reg [7:0] obus, [7:0] q  
5  );  
6      always @ (posedge clk, negedge rst_b)  
7          //treat inputs rst_b, clr_lsb, ld_ibus, sh_r here  
  
9      always @ (*) //write content to obus when ld_obus==1  
10         obus = (ld_obus) ? q : 8'bz;  
11 endmodule
```

Problema 2

Construiți registrul A al arhitecturii înmulțitorului secvențial pentru numere reprezentate în Semn-Mărime.

Porniți de la tiparul de mai jos (accesibil [▶ aici](#)) știind că livrarea conținutului la Outbus se face similar cazului registrului Q.

```
1 module reg_a (  
2     input clk, rst_b, clr, sh_r, ld_sgn, ld_obus, ld_sum,  
3     input sh_i, sgn, [7:0] sum,  
4     output reg [7:0] obus, [7:0] q  
5 );  
6     //implementation here  
  
8     //write content to obus  
9 endmodule
```

Problema 3

Proiectați unitatea de control a arhitecturii înmulțitorului secvențial pentru numere reprezentate în Semn-Mărime, pornind de la tiparul de mai jos (accesibil [aici](#)).

```
1 module ctrl_u(  
2     input  clk, rst_b, bgn, q_0, cnt_is_7,  
3     output c0, c1, c2, c3, c4, c5, c6, fin  
4 );  
5     //implementation here  
6 endmodule
```

Problema 4

Completați arhitectura înmulțitorului secvențial pentru numere reprezentate în Semn-Mărime pornind de la tiparul de mai jos (accesibil [▶ aici](#)), care include un testbench pentru înmulțirea operanzilor $-23 * 2^{-7}$ and $-3 * 2^{-7}$.

```
1 module sm_unit (
2     input clk, rst_b, bgn, [7:0] ibus,
3     output fin, [7:0] obus
4 );
5     //implementation here
6 endmodule

8 module sm_unit_tb;
9     reg clk, rst_b, bgn; reg [7:0] ibus; wire fin; wire [7:0] obus;

11    sm_unit test (.clk(clk), .rst_b(rst_b), .bgn(bgn), .ibus(ibus),
12                .fin(fin), .obus(obus));
13    localparam CLK_PERIOD=100, CLK_CYCLES=17, RST_PULSE=25;
14    localparam X=8'b10010111/*=-23*2^(-7)*/, Y=8'b10000011;/*=-3*2^(-7)*/
15    initial begin clk=1'd0; repeat (CLK_CYCLES*2) #(CLK_PERIOD/2) clk=~clk; end
16    initial begin rst_b=1'd0; #(RST_PULSE); rst_b=1'd1; end
17    initial begin bgn=1'd1; #200; bgn=1'd0; end
18    initial begin ibus=0; #100 ibus=X; #100 ibus=Y; end
19 endmodule
```