



Introdução a Sistemas Lógicos - Trabalho Prático 2:

Arthur Pontes Nader - 2019022294

Rodrigo Ferreira Araújo - 2020006990

Thales Henrique Silva - 2020007040

1) Introdução

O trabalho consiste na implementação de três circuitos simples na linguagem Verilog: registrador para a esquerda, registrador com feedback e contador de Johnson. Os códigos são dados a seguir.

2) Implementações

- Registrador para Esquerda

Código em Verilog:

```
design.sv
1 // Design
2 module Registrador_Esq(clock,clear,E,A);
3     input clock,clear,E;
4     output A;
5     reg A;
6     reg B,C,D;
7     always @(posedge clock or negedge clear)
8     begin
9         //inicializacao dos flip flops
10        if(!clear) begin A<=0;B<=0;C<=0;D<=0;end
11        //saidas e entradas de cada flip flop dependem do anterior
12        else begin
13            A<=B;
14            B<=C;
15            C<=D;
16            D<=E;
17        end
18    end
19 endmodule
```

testbench.sv



```

1 // Testbench
2 module Bancada_Teste;
3 //declaracao das variaveis
4 reg clk,clr,in;
5 wire out;
6 integer i;
7 Registrador_Esq RE(clk,clr,in,out);
8
9 //frequencia do clock e inicializacao do clear
10 initial
11     begin clk=1'b0;#2 clr=0;#5 clr=1; end
12     always #5 clk=~clk;
13
14 //inicializacao da entrada
15 initial begin #2;
16     repeat(2)
17         begin #10 in=0;#10 in=0;#10 in=1;#10 in=1;end
18     end
19
20 //exibicao das saidas
21 initial
22     begin
23         $dumpfile("Registrador_Esq.vcd");
24         $dumpvars(0,Bancada_Teste);
25         #100 $finish;
26     end
27 endmodule

```

Diagrama Temporal:

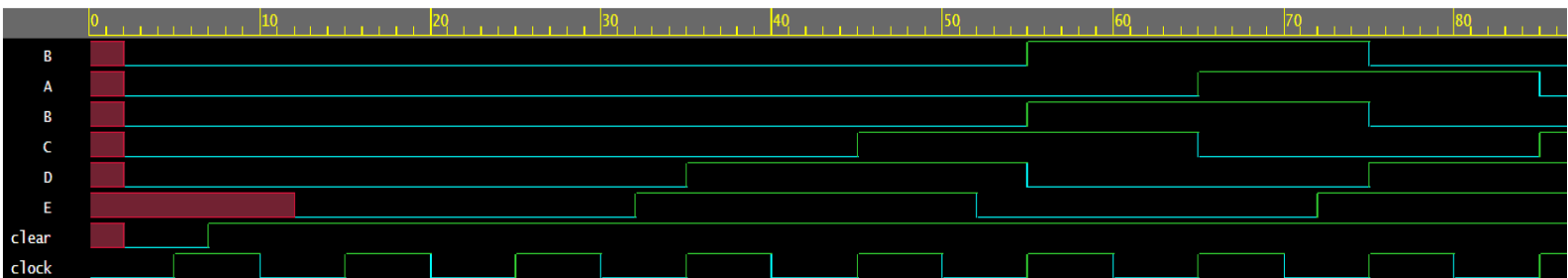
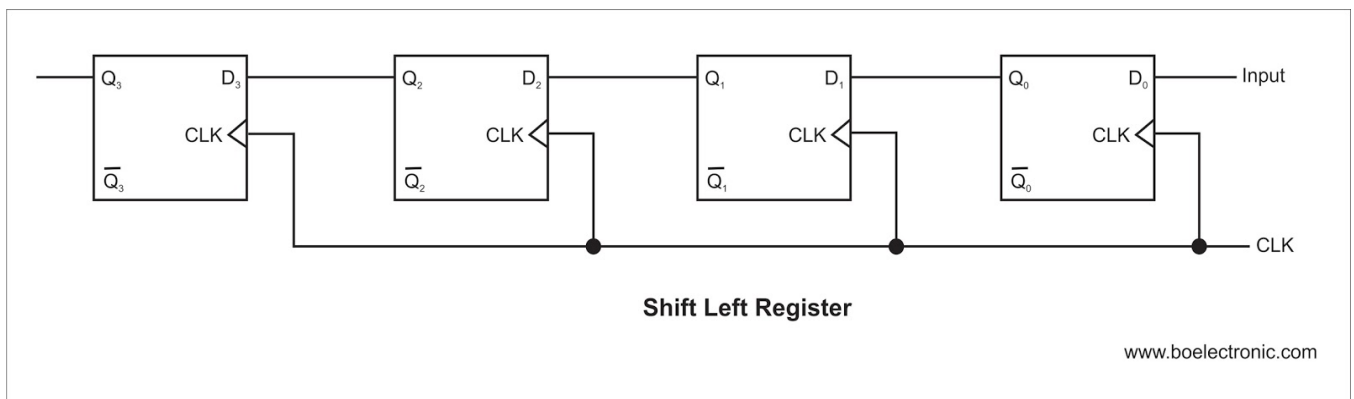


Diagrama esquemático:



- Registrador com feedback

Código em Verilog:

design.sv



```
1 // Design
2 module Registrador_FB(clock,clear,A);
3     input clock,clear;
4     output A;
5     reg A;
6     reg B,C,D;
7     always @(posedge clock or negedge clear)
8     begin
9         //inicializacao dos flip flops
10        if(!clear) begin A<=1;B<=0;C<=0;D<=0;end
11        //saidas e entradas de cada flip flop dependem do anterior
12        else begin
13            A<=B;
14            B<=C;
15            C<=D;
16            D<=A;
17        end
18    end
19 endmodule
```

testbench.sv



```
1 // TestBench
2 module Bancada_Teste;
3     //declaracao das variaveis
4     reg clk,clr;
5     wire out;
6     integer i;
7     Registrador_FB RF(clk,clr,out);
8
9     //frequencia do clock e inicializacao do clear
10    initial
11        begin clk=1'b0;#2 clr=0;#5 clr=1; end
12    always #5 clk=~clk;
13
14    //exibicao das saidas
15    initial
16        begin
17            $dumpfile("Registrador_FB.vcd");
18            $dumpvars(0,Bancada_Teste);
19            #100 $finish;
20        end
21 endmodule
```

Diagrama Temporal:

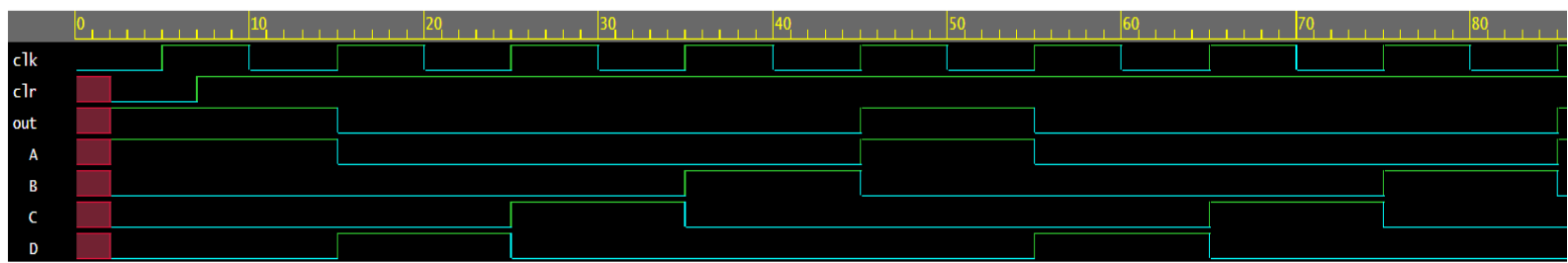
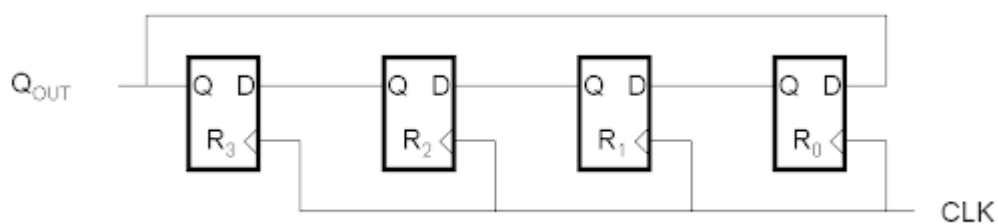


Diagrama esquemático do circuito:



Créditos: http://www.dragonwins.com/domains/getteched/crypto/linear_feedback_shift_registers.htm

- **Contador Johnson**

Código em Verilog:

```
design.sv
1 //Design
2 module Contador_JS(clock,clear,D);
3     input clock,clear;
4     output D;
5     reg A;
6     reg B,C,D;
7     always @(posedge clock or negedge clear)
8     begin
9         //inicializacao dos flip flops
10        if(!clear) begin A<=0;B<=0;C<=0;D<=0;end
11        //saidas e entradas de cada flip flop dependem do anterior
12        else begin
13            A<=~D;
14            B<=A;
15            C<=B;
16            D<=C;
17        end
18    end
19 endmodule
```

testbench.sv



```
1 // Testbench
2 module Bancada_Teste;
3   //declaracao das variaveis
4   reg clk,clr;
5   wire out;
6   integer i;
7   Contador_JS CJ(clk,clr,out);
8
9   //frequencia do clock e inicializacao do clear
10  initial
11    begin clk=1'b0;#2 clr=0;#5 clr=1; end
12  always #5 clk=~clk;
13
14  //exibicao das saidas
15  initial
16    begin
17      $dumpfile("Contador_JS.vcd");
18      $dumpvars(0,Bancada_Teste);
19      #200 $finish;
20    end
21 endmodule
```

Diagrama Temporal:

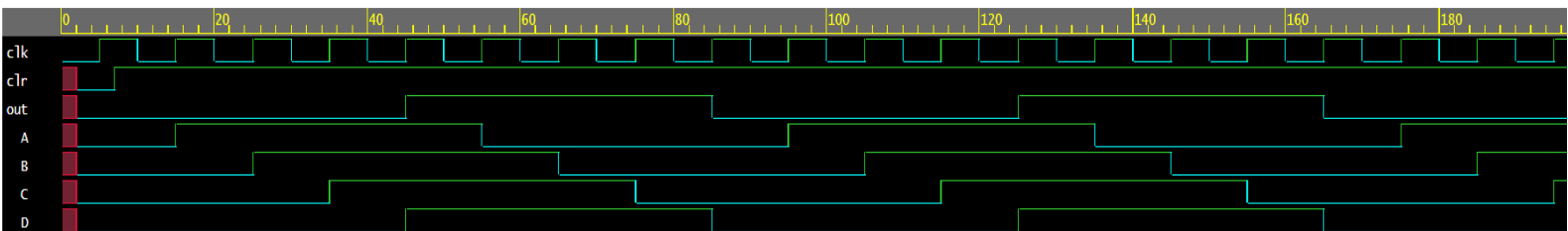
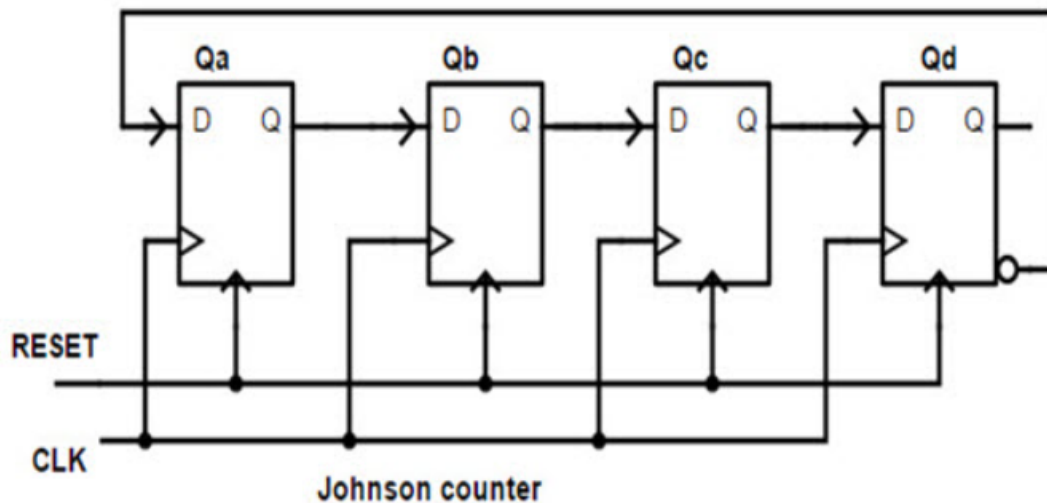


Diagrama esquemático do circuito:



3) Conclusões

- **Registrador para Esquerda**

Para esse registrador, observamos, pelo arquivo de ondas gerado, uma clara movimentação de um bit '1' do bit menos significativo (E) para o mais significativo(A), isto é, shifts à esquerda até o último bit passando por toda a cadeia. Isso pode ser explicado pela troca do input de entrada do bit mais à esquerda pelo bit mais à direita. No entanto, temos, também, mais de um bit '1' por vez na cadeia durante os shifts.

- **Registrador com feedback**

Com o registrador com feedback, observamos o que era esperado no arquivo de ondas: movimentação de um bit da direita para a esquerda (1000 → 0001 → 0010 → 0100 → 1000). Note que, diferente do anterior, não temos mais de um bit por vez na cadeia, o que é explicado pelo fenômeno do feedback (retroalimentação) a partir de entradas pré-definidas.

- **Contador de Johnson**

Esse circuito funciona de maneira semelhante ao anterior, com a diferença de que o bit de retroalimentação é invertido ao entrar no sistema.

4) Referências

- [feedback register | Encyclopedia.com](#)
- [Verilog Ring Counter \(chipverify.com\)](#)
- [Shift Registers in Digital Logic - GeeksforGeeks](#)