Henrique Soares Assumpção e Silva
Matrícula: 2020006620
henriquesoares@dcc.ufmg.br

Trabalho em Verilog II
Introdução a Sistemas Lógicos

2021-08-16

1 Introdução

O programa foi desenvolvido na linguagem Verilog, compilada utilizando a ferramenta Icarus Verilog, no sistema operacional Windows 10. O programa foi inicialmente implementado e compilado em uma máquina com 16GB de RAM em um processador Intel Core I5-9400F.
Todo código deste trabalho está disponível no seguinte repositório do GitHub:
https://github.com/HenrySilvaCS/Logical-Systems/tree/main/tp2
Para compilar e executar os arquivos contidos no repositório utilizando-se o Icarus Verilog, basta realizar os seguintes passos:

```
>>> iverilog -o test_bench_<NUM>.vvp test_bench_<NUM>.v
>>> vvp test_bench_<NUM>.vvp
>>> gtkwave
```

Onde <NUM> indica uma determinada atividade.

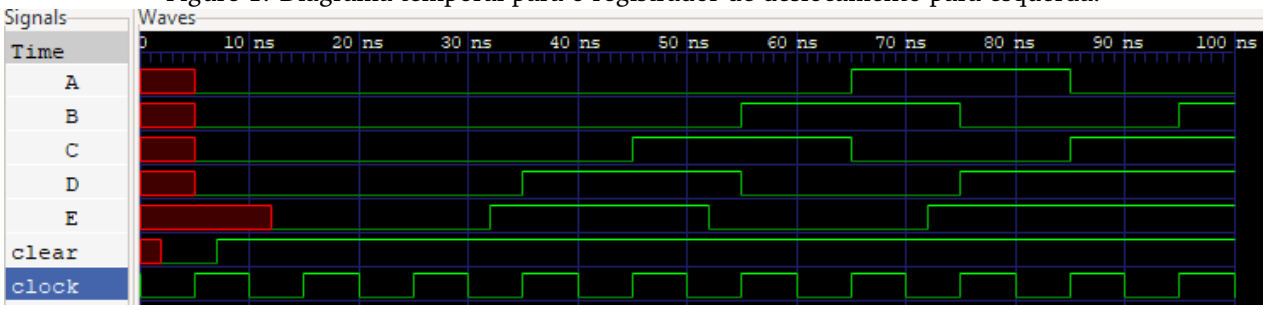
2 Primeira atividade

A primeira atividade requisita a implementação de um registrador de deslocamento para esquerda. O código verilog implementado é apresentado a seguir:

```
module shifting_4bit(clock,clear,E,A);
  input clock,clear,E;
  output A;
  reg A;
  reg B,C,D;
  always @(posedge clock)
    begin
      if(!clear) begin A<=0;B<=0;C<=0;D<=0;end
      else begin
        D<=E;
        C<=D;
        B<=C;
        A<=B;
      end
    end
endmodule
```

Além do código, também foi produzido um diagrama temporal:

Figure 1: Diagrama temporal para o registrador de deslocamento para esquerda.



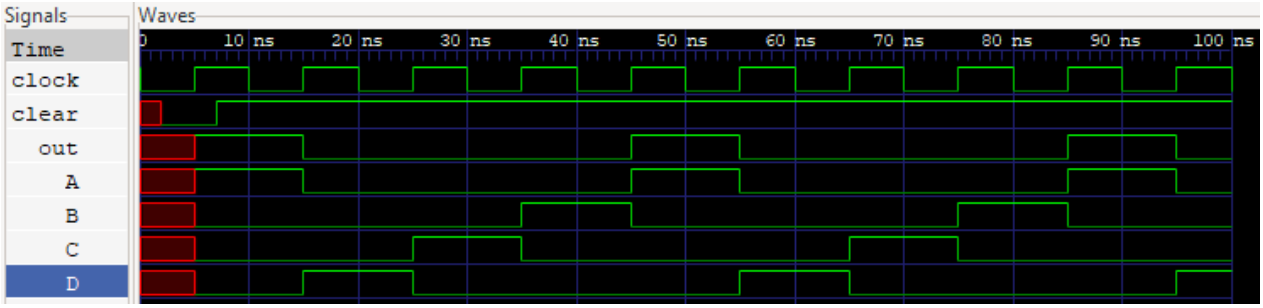
3 Segunda atividade

A segunda atividade requisita a implementação de um contador de 4 bits em anel. O código verilog implementado é apresentado a seguir:

```
module feedback_register(clock, clear,out);
  input clock,clear;
  output out;
  reg A,B,C,D;
  always @(posedge clock)
    begin
      if(!clear) begin A<=1;B<=0;C<=0;D<=0;end
```

```
8       else begin
9         D<=A;
10        C<=D;
11        B<=C;
12        A<=B;
13      end
14      end
15    assign out=A;
16  endmodule
```

Figure 2: Diagrama temporal para o contador de 4 bits em anel.



4 Terceira atividade

A segunda terceira requisita a implementação de um Contador Johnson. O código verilog implementado é apresentado a seguir:

```
1   module johnson_counter(clk,clr,out);
2     input clk,clr;
3     outpput out;
4     reg A,B,C,D;
5     always @(posedge clk)
6       begin
7         if(!clr) begin A<=0;B<=0;C<=0;D<=0;end
8         else begin
9           D<=C;
10          C<=B;
11          B<=A;
12          A<=(~D);
13        end
14      end
15    assign out=D;
16  endmodule
```

Figure 3: Diagrama temporal para o Contador Johnson.