

**Universidade Federal de São Carlos**

**Departamento de Computação**

**Disciplina:**

Laboratório de Arquitetura e Organização de Computadores 1

**Relatório 3**

Circuitos Reconfiguráveis: Introdução a Verilog.

**Daniel Chaves Macedo      RA: 280844**

**Pedro Gabriel Artiga      RA: 351180**

**Felipe Fiori Campos Martins    RA: 316660**

**São Carlos, 2011.**

# Introdução

Utilizando o software Altera Quartus II, simular um projetos usando a linguagem Verilog, que foi desenvolvida na década de 80, muito similar a linguagem C, esta linguagem facilita extremamente o projeto e o design de projetos, principalmente quando estes são de grande porte.

## Objetivo

Temos como principal objetivo neste experimento, conhecer, utilizar e perceber as vantagens de se utilizar a linguagem Verilog, no projeto e na análise de projetos de circuitos sequenciais.

### **Criando um projeto:**

File -> New Project Wizard -> next -> Salve o arquivo com o nome desejado -> next -> Family: Cyclone II, Available Devices : EP2C20F484C7 -> Finish.

### **Criando o gatenetwork em verilog:**

File -> new -> Verilog HDL file -> ok

# Gatenetwork

Crie um projeto como o nome gatenet.

Crie o arquivo verilog HDL.

Código:

```
module gatenet(A,B,C,D,X,Y);
    input A;
    input B;
    input C;
    input[2:1]D;
    output X,Y;
    reg Y;

    wire X=A & ~(B|C) & (D[1]^D[2]);

    always @(A or B or C or D)
        Y=A & ~(B|C) & (D[1]^D[2]);
endmodule
```

Salve como gatenet.

Compile.

File -> Create/Update -> Create Symbol Files for Current File.

**Criando diagram de blocos:**

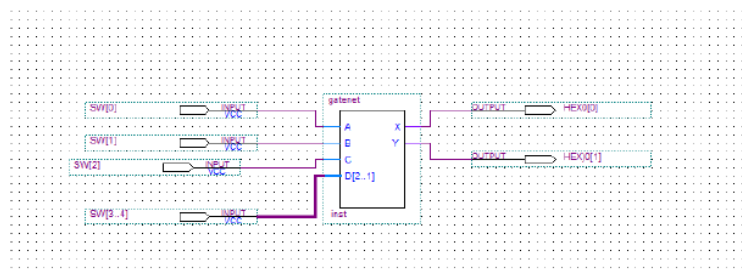
File -> new -> Block diagram/Schematic File

Duplo clique no projeto -> project -> gatenet-> ok e depois na área do diagrama.

Salve como blocogate.

Adicione 4 inputs e 2 outputs ao diagrama.

**NOTA :Os Nomes devem ser mudados de acordo com o manual da placa FPGA:**



Assignments -> Import Assignments -> DE1\_Default.qsf

**Para deixar seu bloco esquemático como top file :**

Files -> botão direito em blocogate.bdf -> Set as top level entity

Start Compilation para compilar.

### Waveform:

File -> New -> Vector Waveform File.

File -> Save: blocogate.vwf

Insert Node or bus -> Node Finder.

Filter -> pins: unassigned -> List.

Botão >> -> OK.

Escolha os valores de entrada -> Start Simulation.



# Decodificador para Display de 7 Segmentos

Crie um projeto como o nome DEC\_7SEG.

Crie o arquivo verilog HDL.

Código:

```
module DEC_7SEG(Hex_digit, segment_a, segment_b, segment_c,
segment_d, segment_e, segment_f, segment_g);
    input [3:0] Hex_digit;
    output segment_a, segment_b, segment_c, segment_d;
    output segment_e, segment_f, segment_g;
    reg [6:0] segment_data;

    always @(Hex_digit)
        case (Hex_digit)
            0: segment_data = 7'b 1111110;
            1: segment_data = 7'b 0110000;
            2: segment_data = 7'b 1101101;
            3: segment_data = 7'b 1111001;
            4: segment_data = 7'b 0110011;
            5: segment_data = 7'b 1011011;
            6: segment_data = 7'b 1011111;
            7: segment_data = 7'b 1110000;
            8: segment_data = 7'b 1111111;
            9: segment_data = 7'b 1111011;
            10: segment_data = 7'b 1110111;
            11: segment_data = 7'b 0011111;
            12: segment_data = 7'b 1001110;
            13: segment_data = 7'b 0111101;
            14: segment_data = 7'b 1001111;
            15: segment_data = 7'b 1000111;
            default:
                segment_data = 7'b 0111110;
        endcase

    /* extract segment data bits and invert */
    /* LED driver circuit is inverted */
    wire segment_a = ! segment_data[6];
    wire segment_b = ! segment_data[5];
    wire segment_c = ! segment_data[4];
    wire segment_d = ! segment_data[3];
    wire segment_e = ! segment_data[2];
    wire segment_f = ! segment_data[1];
    wire segment_g = ! segment_data[0];
endmodule
```

Salve como DEC\_7SEG.

Compile.

File -> Create/Update -> Create Symbol Files for Current File.

### **Criando diagram de blocos:**

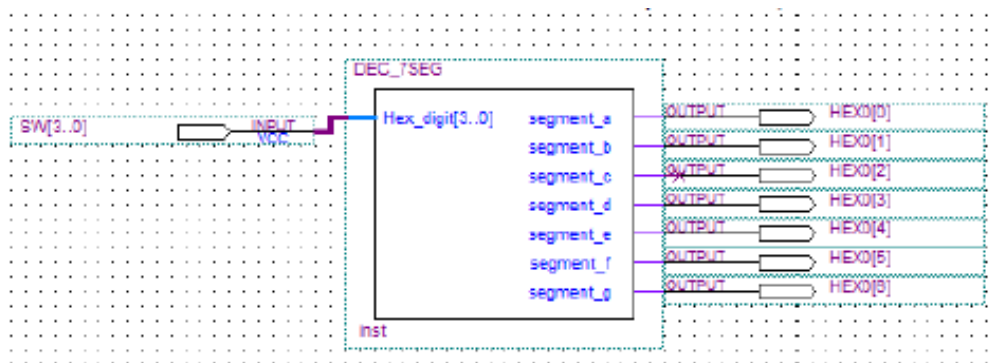
File -> new -> Block diagram/Schematic File

Duplo clique no projeto -> project -> DEC\_7SEG -> ok e depois na área do diagrama.

Salve como blocoDEC.

Adicione 1 input e 7 outputs ao diagrama.

**NOTA :Os Nomes devem ser mudados de acordo com o manual da placa FPGA:**



Assignments -> Import Assignments -> DE1\_Default.qsf

### **Para deixar seu bloco esquemático como top file :**

Files -> botão direito em blocoDEC.bdf -> Set as top level entity

Start Compilation para compilar.

### **Waveform:**

File -> New -> Vector Waveform File.

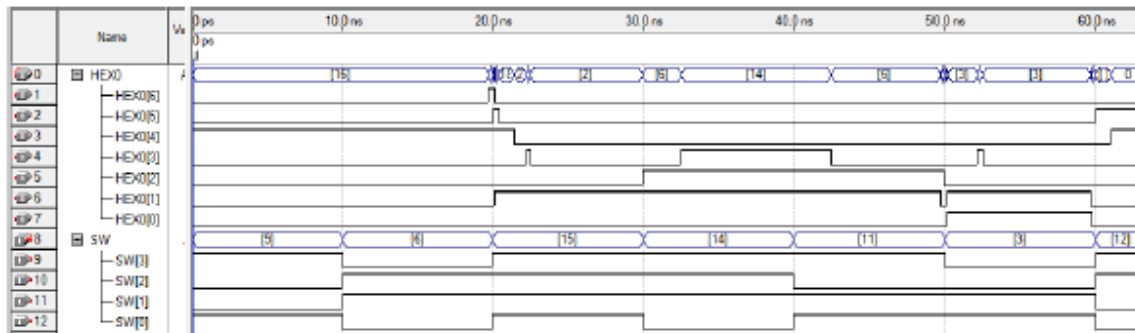
File -> Save: blocoDEC.vwf

Insert Node or bus -> Node Finder.

Filter -> pins: unassigned -> List.

Botão >> -> OK.

Escolha os valores de entrada -> Start Simulation.



## Multiplexador

Crie um projeto como o nome multiplexer.

Crie o arquivo verilog HDL.

Código:

```
module multiplexer(A, B, mux_control, mux_out1,
mux_out2, mux_out3);
```

```
    input A;
    input B;
    input mux_control;
    output mux_out1, mux_out2, mux_out3;
    reg mux_out2, mux_out3;
```

```
    wire mux_out1 = (mux_control)? A:B;
```

```
    always @(A or B or mux_control)
```

```
        if (mux_control)
```

```
            mux_out2 = A;
```

```
        else
```

```
            mux_out2 = B;
```

```
    always @(A or B or mux_control)
```

```
        case (mux_control)
```

```
            0: mux_out3 = B;
```

```
            1: mux_out3 = A;
```

```
            default: mux_out3 = A;
```

```
        endcase
```

```
endmodule
```

Salve com nome de multiplexer.

Start Compilation para compilar.

File -> Create/Update -> Create Symbol Files for Current File

### Criando diagram de blocos:

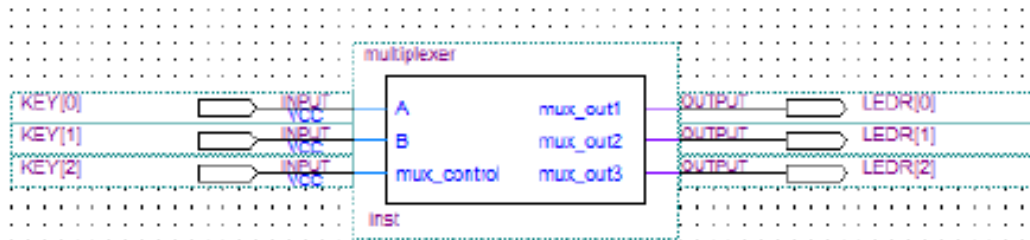
File -> new -> Block diagram/Schematic File

Duplo clique no projeto -> project -> multiplexer -> ok e depois na área do diagrama.

Salve como blocomultiplex.

Adicione 3 inputs e 3 outputs ao diagrama.

**NOTA :Os Nomes devem ser mudados de acordo com o manual da placa FPGA:**



Assignments -> Import Assignments -> DE1\_Default.qsf

### Para deixar seu bloco esquemático como top file :

Files -> botão direito em blocomultiplex.bdf -> Set as top level entity

Start Compilation para compilar.

### Waveform:

File -> New -> Vector Waveform File.

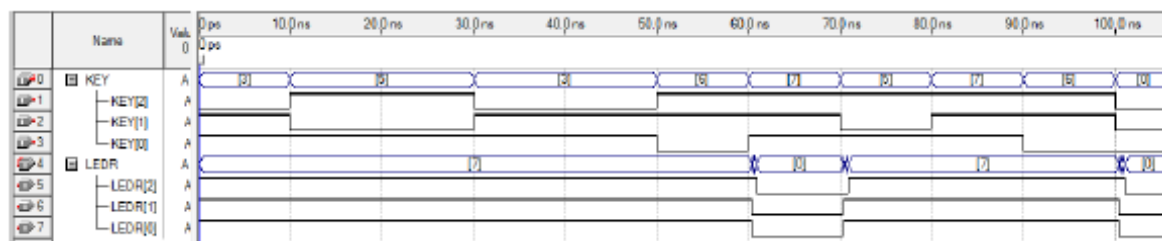
File -> Save: blocomultiplex.vwf

Insert Node or bus -> Node Finder.

Filter -> pins: unassigned -> List.

Botão >> -> OK.

Escolha os valores de entrada -> Start Simulation.





# Tristate

Crie um projeto como o nome tristate.

Crie o arquivo verilog HDL.

Código:

```
module tristate (a, control, tri_out);
    input a;
    input control;
    output tri_out;
    reg tri_out;

    always @(control or a)
    begin
        if (control)
            tri_out = 1'bZ;
        else
            tri_out = a;
    end
endmodule
```

Salve como tristate.

Start Compilation para compilar.

File -> Create/Update -> Create Symbol Files for Current File.

## Criando diagram de blocos:

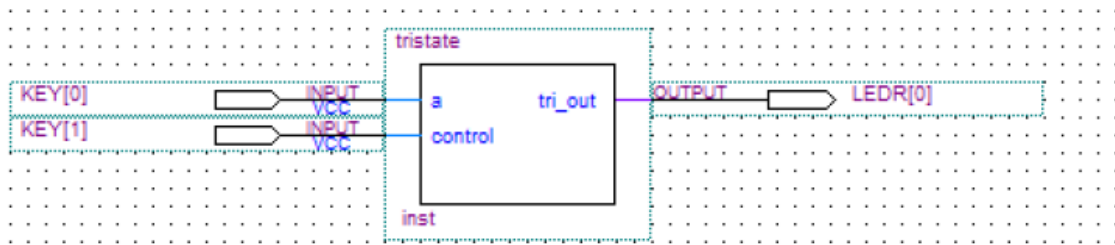
File -> new -> Block diagram/Schematic File

Duplo clique no projeto -> project -> tristate -> ok e depois na área do diagrama.

Salve como blocotristate.

Adicione 4 inputs e 2 outputs ao diagrama.

**NOTA :Os Nomes devem ser mudados de acordo com o manual da placa FPGA:**



Assignments -> Import Assignments -> DE1\_Default.qsf

**Para deixar seu bloco esquemático como top file :**

Files -> botão direito em blocotristate.bdf -> Set as top level entity

Start Compilation para compilar.

### Waveform:

File -> New -> Vector Waveform File.

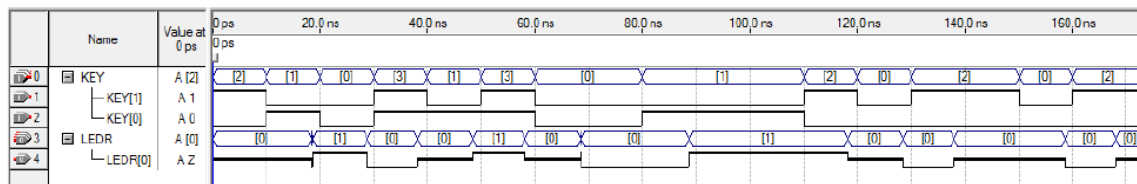
File -> Save: blocotristate.vwf

Insert Node or bus -> Node Finder.

Filter -> pins: unassigned -> List.

Botão >> -> OK.

Escolha os valores de entrada -> Start Simulation.



## Conclusão

A linguagem Verilog, é realmente de grande valia, quando se necessita projetar circuitos mais complexos, além ela possibilita criar um diagrama e utilizar o modulo construido, se necessário fazendo a junção de vários modulos, ainda é possível utilizar a ferramenta de simulações de onda, para testar e avaliar o funcionamento dos modulos projetados.