

Lista de Exercícios 1 - Programando em System Verilog

1. Um circuito combinacional tem uma entrada X, a qual representa dois dígitos decimais. A saída Z é igual a $2X + 5$ se X for maior do que 4; caso contrário a saída Z é igual a $X^3 + 5X$.

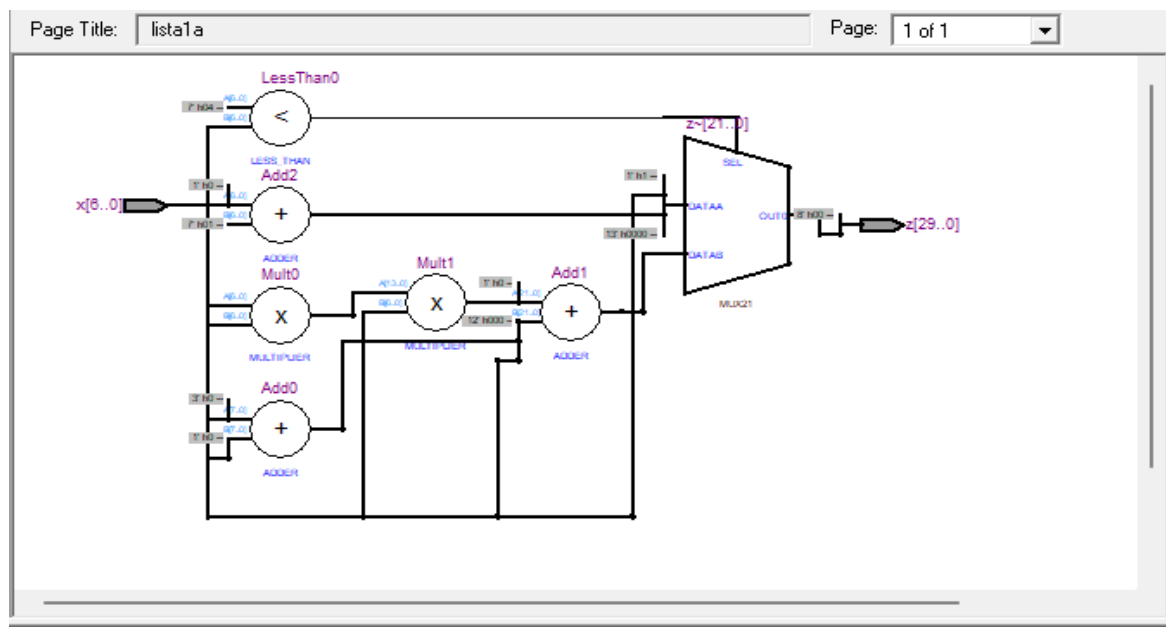
a) Dê uma descrição **System Verilog** do circuito

Resposta:

```
1  module listala
2  (output logic [29:0] z,
3   input logic [6:0] x);
4
5  always @ * begin
6      if (x > 4)
7          z = ( x*x*x)+(5*x);
8      else z = 2*x+5;
9
10 end
11 endmodule
```

b) Obtenha a **RTL Viewer** do circuito

Resposta:



2. Um circuito combinacional deve ser projetado para calcular a distância entre dois 1's em um vetor de bits de entrada (8 bits). Suponha que o vetor tenha exatamente dois 1's. Por exemplo se $x = (1,0,1,0,0,0,0,0)$ então a distância é 2.

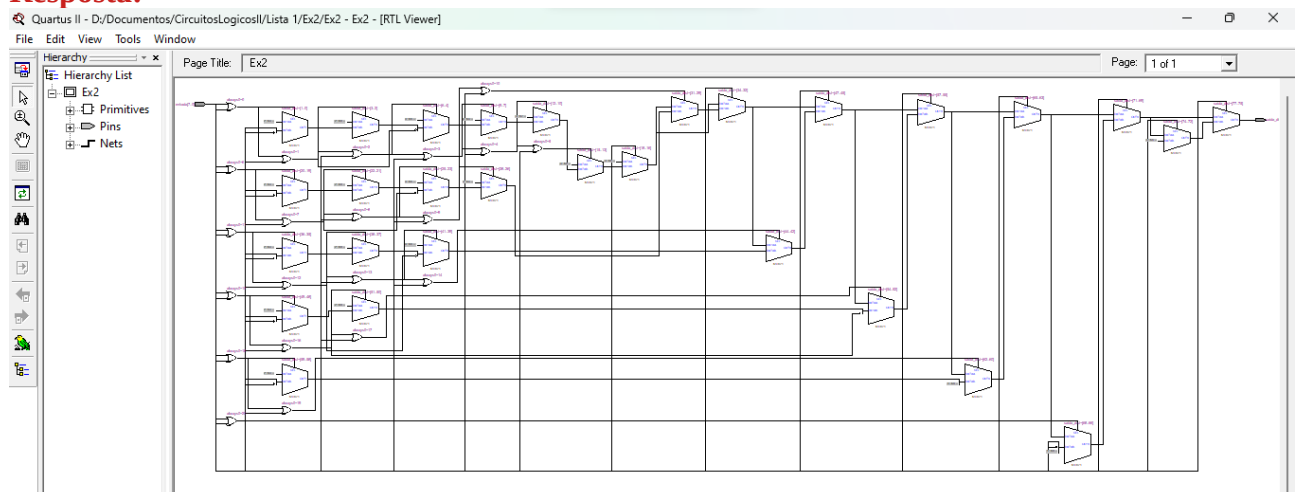
a) Dê uma descrição **System Verilog** do circuito

Resposta:

```
1 //Um circuito combinacional deve ser projetado para calcular a distância entre dois 1's em um vetor de bits de entrada (8 bits). Suponha que o
2 vetor tenha exatamente dois 1's. Por exemplo se x = (1,0,1,0,0,0,0,0) então a distância é 2.
3 module Ex2
4   (input logic [7:0]entrada,
5    output logic [2:0]saida_dist); //O output deve ser de 3 bits, pois a distância máxima é 7, que é representado em binário por 3 bits (111) e
6   0 input deve ser de 8 bits, pois o vetor de bits de entrada tem 8 bits
7
8   always @ * begin //O always é usado para que o código seja executado sempre que houver uma mudança na entrada
9     saida_dist = 0; //A saída é inicializada com 0
10    for (int i = 0; i < 8; i++) begin //O for é usado para percorrer o vetor de entrada
11      if (entrada[i] == 1) begin //Se o valor do vetor for 1, o primeiro 1 é encontrado
12        for (int j = i+1; j < 8; j++) begin //O for é usado para percorrer o vetor de entrada a partir da posição do primeiro 1
13          if (entrada[j] == 1) begin //Se o valor do vetor for 1, o segundo 1 é encontrado
14            saida_dist = j - i; //A distância é calculada
15            break; //O break é usado para sair do for
16          end
17        end
18      end
19    end
20  end
21 endmodule
```

b) Obtenha a **RTL Viewer** do circuito

Resposta:



3. Faça a descrição **System Verilog** de um circuito que conta o número de 1's de um vetor de entrada de 16 bits. Simule a seguir o circuito usando um **testbench simples**.

Resposta:

Descrição:

```
1 // Faça a descrição System Verilog de um circuito que conta o número de 1's de um vetor de entrada de 16 bits. Simule
  a seguir o circuito usando um testbench simples.
2 module Ex3
3   (input logic [15:0] entrada,
4    output logic [15:0] saida_quant_um
5   ); //Entrada de 16 bits como a questão pede, e saída de 16 bits (pois a é a maior quantidade de 1's possível) para
      representar o número de 1's
6
7   always @ * begin //O always é usado para definir um bloco de código que será executado sempre que houver uma mudança
      nos sinais de entrada do módulo.
8     saida_quant_um = 0; //Inicializa a saída com 0
9
10    for (int i = 0; i < 16; i++) begin //Loop para percorrer os 16 bits da entrada
11      if (entrada[i] == 1) begin //Se o bit for 1, incrementa a saída
12        saida_quant_um = saida_quant_um + 1; //Incrementa a saída
13      end
14    end
15
16  end
17
18 endmodule
```

Testbench:

```
1 `timescale 1ns/100ps //Ajusta a escala de tempo para ns
2 module Ex3_tb;
3
4   logic [15:0] entrada, saida_quant_um;
5
6   Ex3 dut(.entrada(entrada), .saida_quant_um(saida_quant_um)); //Instancia o testbench
7
8   initial begin
9     entrada = 16'b0000000000000000; //zero 1's
10    #10
11    assert(saida_quant_um == 0) $display("Test 1 OK");
12    else $error("Test 1 ERROR");
13    $display("Saida esperada: 0");
14    $display("Saida obtida: %d", saida_quant_um);
15
16    entrada = 16'b0000000000000001; //um 1's
17    #10
18    assert(saida_quant_um == 1) $display("Test 2 OK");
19    else $error("Test 2 ERROR");
20    $display("Saida esperada: 1");
21    $display("Saida obtida: %d", saida_quant_um);
22
23    entrada = 16'b0100101001000100; // cinco 1's
24    #10
25    assert(saida_quant_um == 5) $display("Test 3 OK");
26    else $error("Test 3 ERROR");
27    $display("Saida esperada: 5");
28    $display("Saida obtida: %d", saida_quant_um);
29  end
30 endmodule
```

Saida do testbench:

```
Transcript
# .main_pane.objects.interior.cs.body.tree
# run -all
# Test 1 OK
# Saida esperada: 0
# Saida obtida:    0
# Test 2 OK
# Saida esperada: 1
# Saida obtida:    1
# Test 3 OK
# Saida esperada: 5
# Saida obtida:    5

VSIM 2>

Now: 30 ns Delta: 1 sim:/Ex3_tb
```

4. Faça uma descrição em **System Verilog** de um circuito combinacional incrementador/decrementador cuja entrada seja um número inteiro na faixa de **0 a 15**. Uma entrada de **controle X** determina se o sistema incrementa ou decrementa. Faça a seguir a simulação do circuito usando um modelo de testbench que cheque se o resultado está correto ou incorreto (**modelo auto checável**).

Resposta:

5. Descreva um circuito em **System Verilog** para determinar o número de bits "1" e o número de bits "0" de um vetor de entrada de 16 bits. Faça a seguir um **testbench simple para testar o circuito**

Resposta:

Descrição:

```
1 //Descreva um circuito em System Verilog para determinar o número de bits "1" e o número de bits "0" de um vetor de entrada de 16 bits. Faça a
2 seguir um testbench simple para testar o circuito
3 module Ex5(input Logic [15:0]entrada,
4 output Logic [15:0]saida_quant_um, output Logic [15:0]saida_quant_zero); //Entrada de 16 bits como a questão pede, e saída de 16 bits para
5 mostrar a quantidade de 1's e 0's
6
7 always @ * begin //O always é usado para definir um bloco de cód. que será executado sempre que houver uma mudança nos sinais de entrada do
8 módulo.
9   saida_quant_um = 0; //Inicializa a saída com 0
10  saida_quant_zero = 0; //Inicializa a saída com 0
11
12  for (int i=0; i < 16; i++) begin //Loop para percorrer os 16 bits da entrada
13    if (entrada[i] == 1) begin //Se o bit for 1, incrementa a saída correspondente
14      saida_quant_um = saida_quant_um + 1;
15    end
16    else begin //Se o bit for 0, incrementa a saída correspondente
17      saida_quant_zero = saida_quant_zero + 1;
18    end
19  end
20 end
21 endmodule
```

Testbenche:

```
`timescale 1ns/100ps //Ajusta a escala de tempo para ns
module Ex5_tb;

  Logic [15:0] entrada, saida_quant_um, saida_quant_zero;

  Ex5 dut(.entrada(entrada), .saida_quant_um(saida_quant_um), .saida_quant_zero(saida_quant_zero)); //Instancia o testbench

  initial begin
    entrada = 16'b0000000000000000; //zero 1's e dezesseis 0's
    #10
    assert(saida_quant_um == 0 & saida_quant_zero == 16) $display("Test 1 OK");
    else $error("Test 1 ERROR");
    $display("Saida esperada para 1's: 0");
    $display("Saida esperada para 0's: 16");
    $display("Saida obtida para 1's: %d", saida_quant_um);
    $display("Saida obtida para 0's: %d", saida_quant_zero);

    entrada = 16'b0000000000000001; //um 1's e quinze 0's
    #10
    assert(saida_quant_um == 1 & saida_quant_zero == 15) $display("Test 2 OK");
    else $error("Test 2 ERROR");
    $display("Saida esperada para 1's: 1");
    $display("Saida esperada para 0's: 15");
    $display("Saida obtida para 1's: %d", saida_quant_um);
    $display("Saida obtida para 0's: %d", saida_quant_zero);

    entrada = 16'b0100101001000100; //cinco 1's e onze 0's
    #10
    assert(saida_quant_um == 5 & saida_quant_zero == 11) $display("Test 3 OK");
    else $error("Test 3 ERROR");
    $display("Saida esperada para 1's: 5");
    $display("Saida esperada para 0's: 11");
    $display("Saida obtida para 1's: %d", saida_quant_um);
    $display("Saida obtida para 0's: %d", saida_quant_zero);
  end
endmodule
```

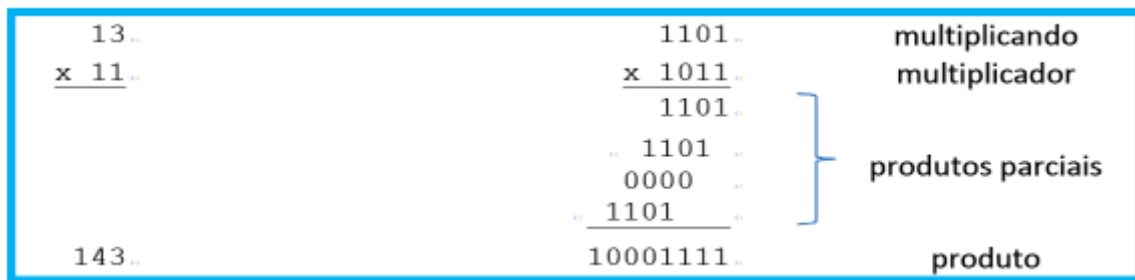
Saida do Testbenche:

```
Transcript
# Test 1 OK
# Saida esperada para 1's: 0
# Saida esperada para 0's: 16
# Saida obtida para 1's:    0
# Saida obtida para 0's:   16
# Test 2 OK
# Saida esperada para 1's: 1
# Saida esperada para 0's: 15
# Saida obtida para 1's:    1
# Saida obtida para 0's:   15
# Test 3 OK
# Saida esperada para 1's: 5
# Saida esperada para 0's: 11
# Saida obtida para 1's:    5
# Saida obtida para 0's:   11

VSIM 2> ]

Now: 30 ns Delta: 1 [8]
```

6. Descreva em **System Verilog** um circuito para executar o algoritmo tradicional de multiplicação binária entre dois números de 4 bits, executado com sucessivos deslocamentos do multiplicando à esquerda e soma dos produtos parciais, segundo a figura abaixo:



Dica: use a instrução **for**, o operador de deslocamento (<<) e a operação soma (+)

Resposta:

7. Dado o programa abaixo em C:

```
#include <stdio.h>
int main()
{
    int fat, n;
    printf("Insira um valor para o qual deseja calcular seu fatorial: ");
    scanf("%d", &n);
    fat = 1;
    for(n; n > 1; n = n - 1)
        fat = fat * n;
    printf("\nFatorial calculado: %d", fat);
    getch();
    return 0;
}
```

Tomando como base o programa acima, desenvolva um circuito com um vetor de entrada de 4 bits e um vetor de saída de 16 bits, para calcular o fatorial da entrada, usando uma descrição comportamental em System Verilog e responda:

a) Quais as modificações que devem ser feitas para adaptar a codificação C acima em System Verilog ?

Resposta:

b) Qual o maior valor de entrada possível que o circuito consegue calcular o fatorial ?

Resposta:

c) Faça um **auto checável testbench** para testar o circuito

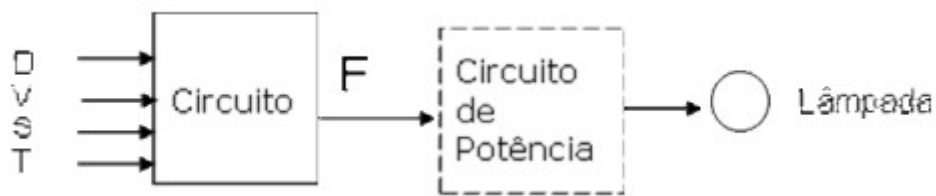
Resposta:

8. Uma escola tem sua diretoria constituída pelos seguintes elementos: Diretor, Vice-Diretor, Secretário e Tesoureiro. Uma vez por mês essa diretoria se reúne para decidir sobre diversos

assuntos, sendo que as propostas são aceitas ou não através de votação. Devido ao número de elementos da diretoria ser par, o sistema adotado é o seguinte:

- 1) Maioria absoluta – a proposta é aceita ou não se no mínimo três elementos são respectivamente a favor ou contra.
- 2) Empate – vence o voto dado pelo diretor

Deseja-se projetar um circuito de votação que acenda uma lâmpada, conforme a proposta seja aprovada ou não de acordo com o esquema abaixo:



Modele o circuito em System Verilog e faça um **autohecável testbench** para testar o circuito

Resposta: