Programa para Excelência em Microeletrônica

Matéria: SystemVerilog

Exercício 02

Henrique Martins Miranda

Sumário

[Introdução iii](#_Toc468640646)

[Questão 01 iii](#_Toc468640647)

[Questão 02 iv](#_Toc468640648)

[Questão 03 v](#_Toc468640649)

[Questão 04 v](#_Toc468640650)

[Questão 05 vi](#_Toc468640651)

[Questão 06 vi](#_Toc468640652)

[Questão 07 vi](#_Toc468640653)

[Questão 08 vi](#_Toc468640654)

[Questão 09 vi](#_Toc468640655)

[Questão 10 vi](#_Toc468640656)

# Introdução

Todas as questões feitas estão disponíveis nesse repositório Github <*github.com/MrH3nry/Henrique\_Martins\_Ex02\_SV*>.

# Questão 01

**module** Registrador(

input logic clock,

input logic [3:0] entrada,

output logic[3:0] saida

);

always\_ff @ (posedge clock)

      saida <= entrada;

**endmodule**

**module** Questao01 (

input swap, enable, clock,

output logic[3:0] UpSaida, DownSaida

);

logic[3:0] UpEntrada, DownEntrada;

Registrador Upcount (.clock(clock), .entrada(UpEntrada), .saida(UpSaida)),

Downcount (.clock(clock), .entrada(DownEntrada), .saida(DownSaida));

always\_comb

begin

if(enable)

if(swap)

UpEntrada = DownSaida;

else

UpEntrada = UpSaida + 4'd1;

else

UpEntrada = UpSaida + 4'd1;

end

always\_comb

begin

if(enable)

if(swap)

DownEntrada = UpSaida;

else

DownEntrada = DownSaida - 4'd1;

else

DownEntrada = DownSaida - 4'd1;

end

**endmodule**

# Questão 02

**module** Count(

input logic clock,

input logic [8:0] entrada,

output logic[8:0] saida

);

always\_ff @ (posedge clock)

      saida <= entrada;

**endmodule**

**module** Questao02(

input clock,

output logic f

);

logic[8:0] CountEntrada, CountSainda;

Count count(.clock(clock), .entrada(CountEntrada), .saida(CountSainda));

always\_comb

begin

if(CountSainda == 8'd499)

CountEntrada = 8'd0;

else

CountEntrada = CountSainda + 8'd1;

end

always\_comb

begin

if((CountSainda > 8'd19) && (CountSainda < 8'd90))

f = 1'd0;

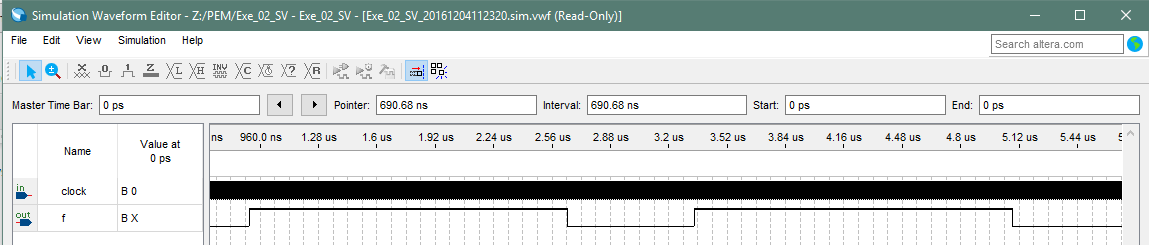
else

f = 1'd1;

end

**endmodule**

* F em 0 passa aproximadamente 700 ns;
* F em 1 passa aproximadamente 1,73 us.



*Figura 1 – Diagrama de Tempo da questão 02.*

# Questão 03

*Figura 2 – Questão 03.*

# Questão 04



*Figura 3 – Questão 04.*

*Figura 4 – Multiplex 2-1 de 8 bits.*



# Questão 05

Sim, pode substituir o laço “for” por um “while”, esses dois laços tem a mesma função, contudo, com sintax diferente, por exemplo a incrementarão da variável “i” tem que ser feita dentro do “while”. O código ficaria assim:

[...]

i =0;

while (i < 8)

**begin**

if(data\_in[i]) msb <= i;

i = i+1;

**end**

[...]



*Figura 5 – Multiplex 2-1 3bits.*



*Figura 6 – Questão 05.*

# Questão 06

C:\Users\Henrique\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Questao06Maior10b.eps

*Figura 7 – Comparador “>” de 10 bits.*



*Figura 8 – Contador de 10 bits parte 1.*



*Figura 9 – Contador de 10 bits parte 2.*

*C:\Users\Henrique\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Questao06.eps*

*Figura 10 – Questão 06.*

# Questão 07

A saída “f” receber o valor de “c” antes de “c” receber “a + b”, pós é bloqueante, contudo, “a” e “b” recebe os valores não bloqueantes para que quando o “c” for receber os valores deles, já receber o “valor atualizado” e assim poder ir contando. O “f” é sempre o valor antigo de “c”.



*Figura 11 – Meio Somador.*



*Figura 12 – Somador Completo.*

*C:\Users\Henrique\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Questao 07 Somador 8b.eps*

*Figura 13 – Somador 8bits.*

*C:\Users\Henrique\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Questao 07.eps*

*Figura 14 – Questão 07.*

# Questão 08

# Questão 09

# Questão 10