

PCS3635 – Laboratório Digital I

Planejamento e Relatório do Experimento 3

Primeiro sistema digital



Turma 3 – Bancada 3

Professores:

Paulo Sergio Cugnasca

Edson Midorikawa

Integrantes:

Arthur Pires da Fonseca – 10773096

Lucas Lopes de Paula Junior - 9344880

19 de janeiro de 2020

INTRODUÇÃO

Nesta experiência será estudado como organizar o projeto de um sistema digital simples, composto por um fluxo de dados e uma unidade de controle. A parte experimental consiste no projeto de um circuito que mede a largura de um pulso de entrada.

1. PARTE TEÓRICA (RESUMO)

O projeto de um circuito digital simples pode ser dividido basicamente em 2 partes: a Unidade de Controle (UC) e o Fluxo de Dados (FD). Para que se tenha maior êxito no projeto e na sua execução é imprescindível seguir uma metodologia a fim de dividir (dividir para conquistar) o todo em módulos menores com o intuito de facilitar a implementação de cada um de forma que todos se comuniquem entre si depois.

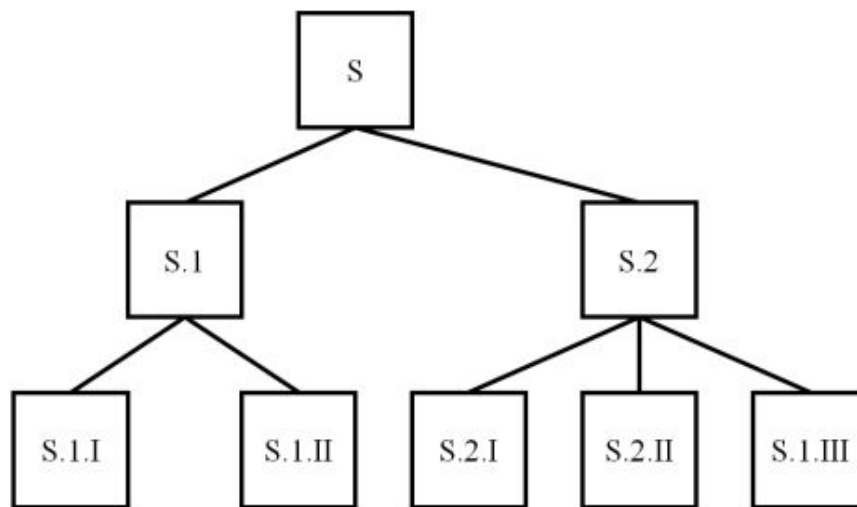


Figura 1 – Decomposição de um sistema em subsistemas.

ETAPAS DE UM PROJETO DE SISTEMAS DIGITAIS

O desenvolvimento do projeto de sistemas digitais pode ser dividido em sete etapas [Midorikawa, 2017]:

1. Escrever uma descrição verbal do sistema;
2. Desenvolver um pseudocódigo do funcionamento;
3. Elaborar um diagrama ASM de alto nível;
4. Selecionar os elementos do fluxo de dados para as operações;
5. Identificar sinais de status e de controle dos elementos do fluxo de dados;
6. Elaborar um diagrama de transição de estados da unidade de controle a partir do fluxo de dados;
7. Verificar conexão do fluxo de dados e da unidade de controle para formar o sistema digital.

2. PARTE EXPERIMENTAL

2.1. Atividade 1 – Projeto de um Circuito Simples

O objetivo é criar um circuito digital simples que mede a largura de um pulso de entrada. Partiremos de um pseudo-código (figura 3) e também usaremos além do contador de 8 bits desenvolvido nos experimentos anteriores também iremos usar uma máquina de estados para controlar o funcionamento do sistema digital.

```
Algoritmo: medidor de largura
entradas: liga, pulso
saídas: largura, pronto
{
1.      while (verdadeiro) {
2.          espera acionamento do sinal LIGA
3.          espera início do PULSO de entrada
4.          while (pulso em 1)
5.              habilita contagem de largura
6.          ativa sinal PRONTO
7.      }
}
```

Figura 3 – Pseudocódigo do medidor de largura de pulso.

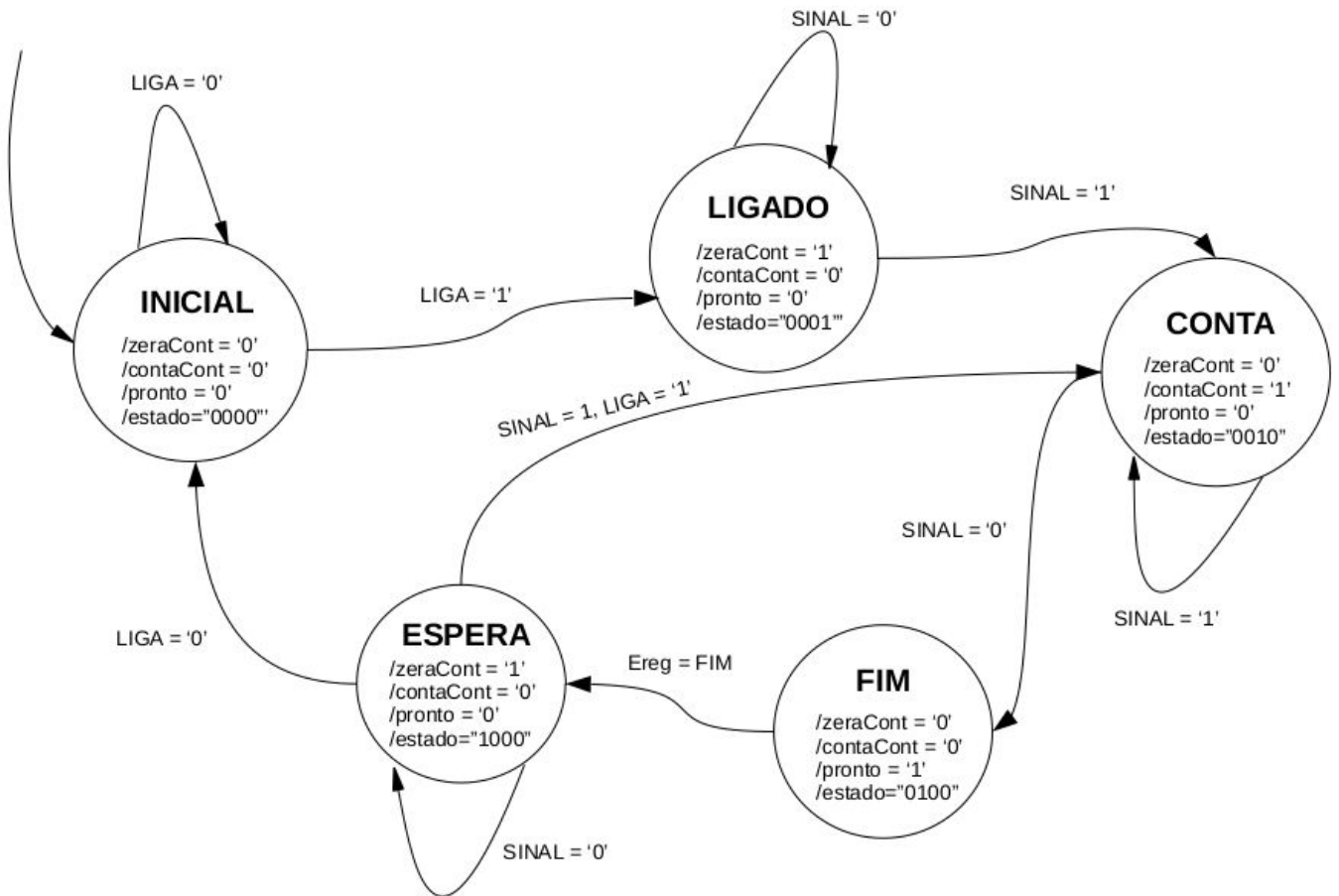
2.1.1. Estudo de uma Máquina de Estados

Considere a descrição VHDL apresentada na figura 4. Ela será utilizada no projeto do medidor de largura de pulso.

```
1  -- Arquivo: controlador.vhd
2  -- Projeto: Jogo do Tempo de Reacao
3  -- Data: 09/01/2020
4
5  -- Descricao: controlador (unidade de controle)
6  -- do medidor do tempo de reacao (largura de pulso)
7  -- descricao VHDL comportamental
8
9
10 -- sinais de condicao: liga - aciona medida da largura
11 --                    sinal - pulso a ser medido
12 -- sinais de controle: zeraCont - zera contagem
13 --                    contaCont - habilita contagem
14 --                    pronto - indica final da medida
15
16 -- Revisoes:
17 -- Data: Versao: Autor: Descricao
18 -- 09/01/2020 1.0 Edson Midorikawa criacao
19
20
21 library IEEE;
22 use IEEE.std_logic_1164.all;
23
24 entity controlador is
25     port (
26         clock, reset: in STD_LOGIC;
27         liga, sinal: in STD_LOGIC;
28         zeraCont, contaCont: out STD_LOGIC;
29         pronto: out STD_LOGIC;
30         estado: out STD_LOGIC_VECTOR(3 downto 0)
31     );
32 end controlador;
33
34 architecture controlador_arch of controlador is
35     type tipo_estado is (INICIAL, LIGADO, CONTA, ESPERA, FIM);
36     signal Ereg, Eprox: tipo_estado;
37 begin
38     -- mudanca de estado
39     process (clock, reset)
40     begin
41         if reset = '1' then
42             Ereg <= INICIAL;
43         elsif clock'event and clock = '1' then
44             Ereg <= Eprox;
45         end if;
46     end process;
47
48     -- logica de proximo estado
49     process (LIGA, SINAL, Ereg)
50     begin
51         case Ereg is
52             when INICIAL => if LIGA = '0' then Eprox <= INICIAL;
53                             else Eprox <= LIGADO;
54                             end if;
55             when LIGADO => if SINAL = '0' then Eprox <= LIGADO;
56                             else Eprox <= CONTA;
57                             end if;
58             when CONTA => if SINAL = '1' then Eprox <= CONTA;
59                             else Eprox <= FIM;
60                             end if;
61             when FIM => Eprox <= ESPERA;
62             when ESPERA => if LIGA = '0' then Eprox <= INICIAL;
63                             elsif SINAL = '0' then Eprox <= ESPERA;
64                             else Eprox <= CONTA;
65                             end if;
66             when others => Eprox <= INICIAL;
67         end case;
68     end process;
69
70     -- sinais de controle ativos em alto
71     with Ereg select
72         zeraCont <= '1' when LIGADO | ESPERA,
73         '0' when others;
74     with Ereg select
75         contaCont <= '1' when CONTA,
76         '0' when others;
77     with Ereg select
78         pronto <= '1' when FIM,
79         '0' when others;
80     with Ereg select
81         estado <= "0000" when INICIAL,
82         "0001" when LIGADO,
83         "0010" when CONTA,
84         "0100" when FIM,
85         "1000" when ESPERA,
86         "1111" when others;
87 end controlador_arch;
```

Figura 4 – Descrição VHDL da máquina de estados a ser estudada.

O desenho do diagrama de transição de estados está representado abaixo.



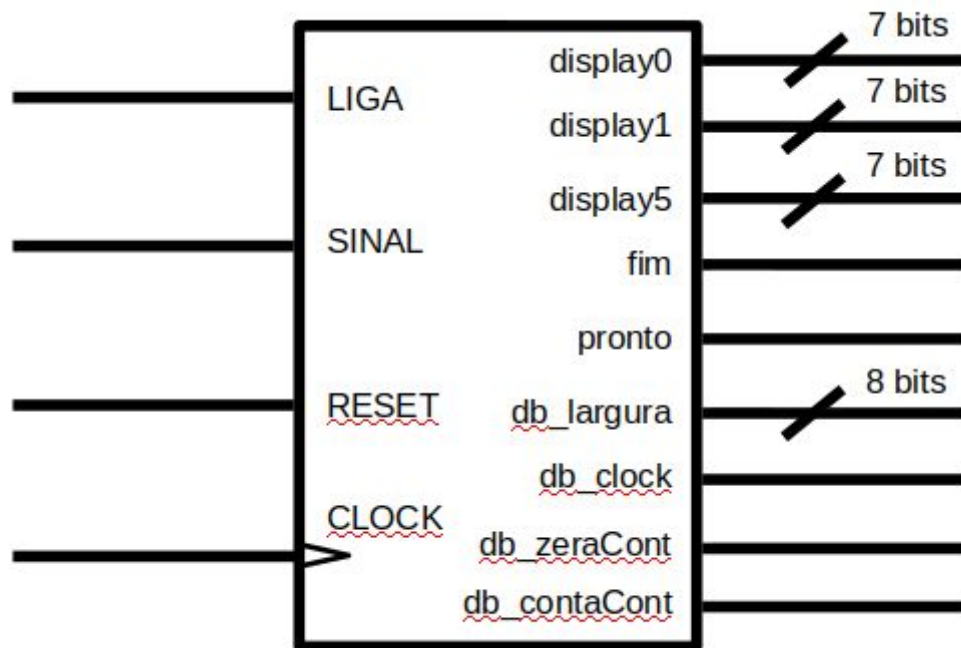
Estado	Comando
INICIAL	espera acionamento do sinal LIGA
LIGADO	espera início do PULSO de entrada
CONTA	while(pulso em 1) habilita contagem de largura
FIM	ativa sinal PRONTO
ESPERA	espera início do PULSO de entrada

Respostas às primeiras perguntas:

1. Que tipo de máquina de estados é descrita pelo código VHDL (Mealy ou Moore)?

A máquina de estados descrita é de Moore, pois os sinais de saída dependem exclusivamente do estado atual.

2. Mostre o símbolo lógico do circuito representado, apresentando seus sinais de entrada e saída.



3. O sinal RESET é síncrono ou assíncrono? Por quê?

Ele é assíncrono, pois independe do sinal de *clock* e pode ser acionado a qualquer momento (está na lista de sensibilidade do *process*), modificando o estado atual.

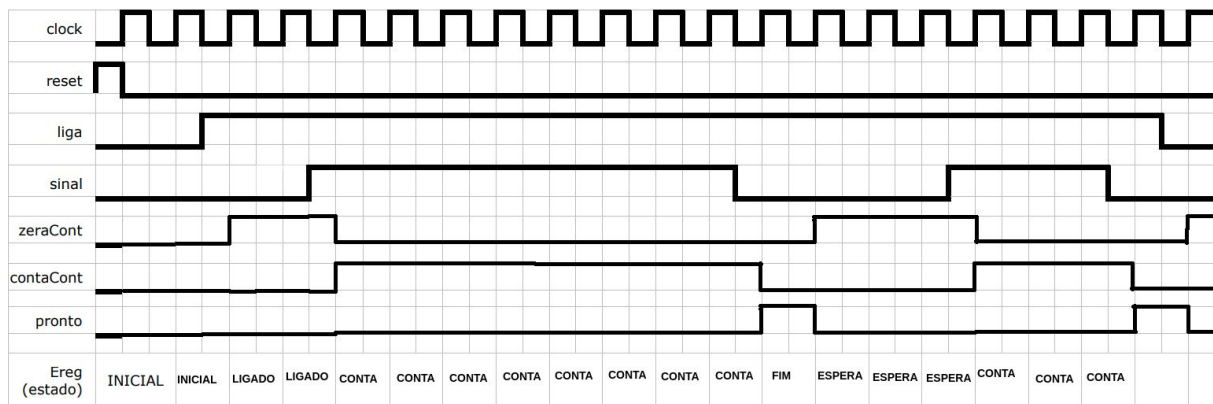
4. É possível a máquina de estados mudar para um estado que não seja um dos estados válidos (INICIAL ou LIGADO ou CONTA ou FIM ou ESPERA)? Explique sua resposta.

Sim, é possível a máquina mudar para um estado inválido, pois o arquivo *controlador.vhd* não especifica uma configuração inicial para o sinal *Ereg*, portanto ele pode começar com um valor não previsto no projeto do circuito. Esse fato não é, contudo, um problema, pois se um estado inválido for alcançado, a máquina mudará para o estado inicial (linha 71).

5. É possível a saída de depuração “estado” apresentar um valor diferente de “0000” ou “0001” ou “0010” ou “0100” ou “1000” ou “1111”? Explique sua resposta.

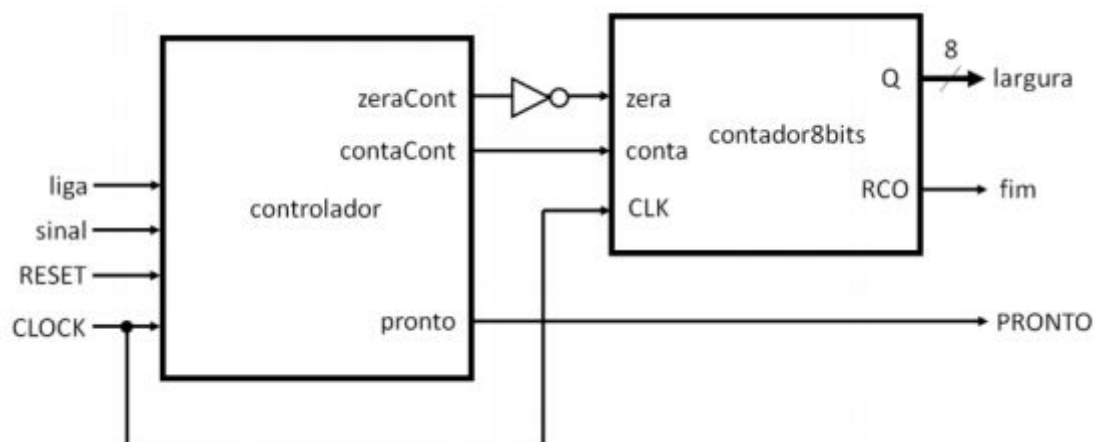
Não, o valor do sinal “estado” é restringido pelo multiplexador definido entre as linhas 85 e 91 do *controlador.vhd*, que define os valores que o “estado” deve assumir quando a máquina estiver em um estado válido e impõe o valor “1111” para qualquer outro estado (inválido) ao usar a sintaxe “others”.

Tabela de forma de onda esperada dos sinais do circuito:

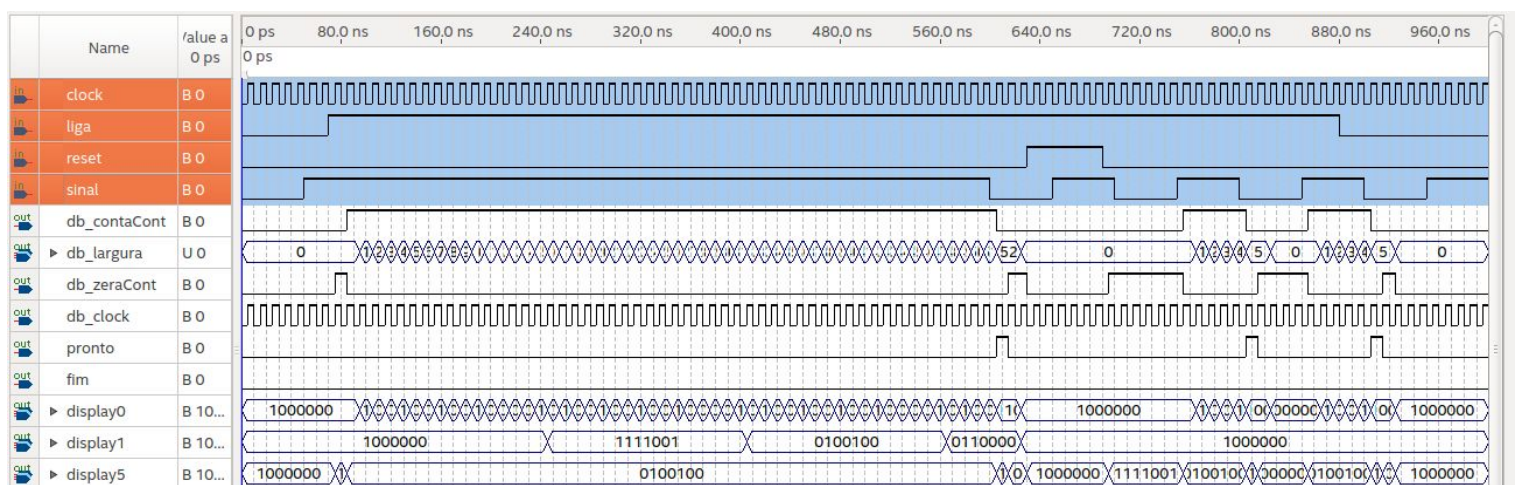


2.1.2. Projeto do Medidor de Largura de Pulso e Simulação

Nesta seção, acoplamos o contador8bits.vhd com a descrição VHDL desta experiência, controlador.vhd, conforme o esquema abaixo.



O arquivo de projeto chama-se `medidor_largura.vhd` e está em anexo com este projeto. Com ele foi gerada a imagem da simulação abaixo no Quartus:



Plano de testes do circuito

A seguir está descrita a sequência de testes a serem feitos durante o laboratório e como os sinais de depuração do arquivo `medidor_largura.vhd` podem ser usados para ajustar a nossa descrição VHDL.

Os sinais de depuração são `display5`, `db_zeraCont` e `db_contaCont` poderão ser utilizados conjuntamente para informar o estado atual do sistema digital e se os sinais `zeraCont` e `contaCont` estão com os respectivos valores previstos para aquele estado.

O sinal `db_clock` permitirá a visualização da borda de subida de clock.

O sinal `db_largura` será usado para verificar a passagem do tempo contada pelo circuito (deverá mudar apenas quando há borda de subida do *clock*) e será amostrado no `display` de 7 segmentos.

Associação entre os sinais da entidade `medidor_largura` e os pinos da placa FPGA

Designação			
sinal	nome	número do pino	na placa
clock	KEY0	PIN_U7	botão KEY0
reset	SW0	PIN_U13	chave SW0
liga	SW1	PIN_V13	chave SW1
sinal	SW2	PIN_T13	chave SW2
largura0[0]	HEX00	PIN_U21	7 Seg Digit 0 [0]
largura0[1]	HEX01	PIN_V21	7 Seg Digit 0 [1]
largura0[2]	HEX02	PIN_W22	7 Seg Digit 0 [2]
largura0[3]	HEX03	PIN_W21	7 Seg Digit 0 [3]
largura0[4]	HEX04	PIN_Y22	7 Seg Digit 0 [4]
largura0[5]	HEX05	PIN_Y21	7 Seg Digit 0 [5]
largura0[6]	HEX06	PIN_AA22	7 Seg Digit 0 [6]
largura1[0]	HEX10	PIN_AA20	7 Seg Digit 1 [0]
largura1[1]	HEX11	PIN_AB20	7 Seg Digit 1 [1]
largura1[2]	HEX12	PIN_AA19	7 Seg Digit 1 [2]
largura1[3]	HEX13	PIN_AA18	7 Seg Digit 1 [3]
largura1[4]	HEX14	PIN_AB18	7 Seg Digit 1 [4]
largura1[5]	HEX15	PIN_AA17	7 Seg Digit 1 [5]
largura1[6]	HEX16	PIN_U22	7 Seg Digit 1 [6]
estado[0]	HEX40	PIN_U20	7 Seg Digit 4 [0]

estado[1]	HEX41	PIN_Y20	7 Seg Digit 4 [1]
estado[2]	HEX42	PIN_V20	7 Seg Digit 4 [2]
estado[3]	HEX43	PIN_U16	7 Seg Digit 4 [3]
estado[4]	HEX44	PIN_U15	7 Seg Digit 4 [4]
estado[5]	HEX45	PIN_Y15	7 Seg Digit 4 [5]
estado[6]	HEX46	PIN_P9	7 Seg Digit 4 [6]
fim	LEDR0	PIN_AA2	led LEDR0
db_clock	LEDR1	PIN_AA1	led LEDR1
db_zeraCont	LEDR2	PIN_W2	led LEDR2
db_contaCont	LEDR3	PIN_Y3	led LEDR3
pronto	LEDR9	PIN_L1	led LEDR9

2.2. Atividade 2 – Síntese na Placa FPGA e Teste de Funcionamento

2.2.1. Circuito com sinal de clock em botão

Utilizando as descrições VHDL citadas anteriormente e a configuração de pinos da placa FPGA, programamos a placa DE0-CV com a ajuda do *software* Altera Quartus Prime.

A execução do plano de testes resultou na seguinte configuração:

Teste	Passos	Resultado observado
procurar estado inválido	-CLK (quantas vezes for necessário)	-nenhum estado inválido observado (mantém o estado 0, INICIAL) -LEDR1 apaga quando o botão KEY0 é pressionado e acende quando está solto
inicializar a máquina de estados	-Reset(H), CLK	-a máquina se mantém no estado 0 (é resetada com sucesso)
transferir para o estado LIGADO	-Reset(L), Liga(H), CLK	-estado atual = 1 (LIGADO) -LEDR2 aceso
testar permanência no estado LIGADO	-Liga(H), Sinal(L), CLK (3 vezes)	-nenhuma mudança observada (mantém o estado LIGADO)
transferir para o estado CONTA	-Sinal(H), CLK	-estado atual = 2 (CONTA) -LEDR2 apagado -LEDR3 aceso

testar permanência no estado CONTA	-Sinal(H), CLK (3 vezes)	-permanece o estado, o display da contagem incrementa até 0x03 (partindo do 0)
testar contagem em si	-Sinal(H), CLK (15 vezes)	-o display de contagem mostra 0x12
transferir para o estado FIM	-Sinal(L), CLK	-estado atual = 4 (FIM) -LEDR3 apagado -LEDR9 aceso
ir do estado FIM para o ESPERA	-CLK	-estado atual = 8 (ESPERA) -LEDR2 aceso -LEDR9 apagado
testar permanência no estado ESPERA	-Sinal(L), CLK (3 vezes)	-ao primeiro <i>clock</i> , o <i>display</i> zerou a contagem, depois o estado foi mantido em 8 (ESPERA)
voltar para o estado CONTA	-Sinal(H), CLK	-estado atual = 2 (CONTA) -LEDR2 apagado -LEDR3 aceso
voltar para o estado ESPERA	-Sinal(L), CLK, CLK	-estado atual = 8 (ESPERA) -LEDR2 aceso -LEDR3 apagado
voltar para o estado INICIAL	-Liga(L), CLK	-estado atual = 0 (INICIAL) -LEDR2 apagado
ir até o estado CONTA e depois resetar	-Liga(H), Sinal(H), CLK, CLK, Reset(H)	-o estado avançou até 2 (CONTA) e depois, ao empurrarmos a alavanca, todos os <i>displays</i> foram resetados para 0 -Todos os LEDs desligados, exceto o LEDR1 (<i>clock</i>)

2.2.2. Circuito com sinal de clock de 1Hz

Modificamos a pinagem do sinal de CLOCK para permitir o uso de um sinal digital de 1Hz emitido por um Analog Discovery da Digilent.

Designação		
sinal	nome	número do pino
CLOCK	GPIO_0_D0	PIN_N16

Usando o *software* Waveforms, programamos o nosso gerador de pulsos para produzir um sinal de *clock* periódico com frequência de 1Hz.

Teste	Passos	Resultado observado
procurar estado inválido	-Conectar o CLK do Analog Discovery ao FPGA	-nenhum estado inválido encontrado (estado 0, INICIAL foi mantido durante os 30 segundos de execução) -LEDR1 piscando com frequência de 1Hz
inicializar a máquina de estados	-Reset(H)	-máquina inicializada, estado = 0 (INICIAL)
transferir para o estado LIGADO	-Liga(H)	-estado atual = 1 (LIGADO) -LEDR2 aceso
testar permanência no estado LIGADO	-Liga(H), Sinal(L)	-nenhuma alteração
transferir para o estado CONTA	-Sinal(H)	-display contando a cada segundo, estado = 2 (CONTA) -LEDR2 apagado -LEDR3 aceso
testar permanência no estado CONTA	-Sinal(H)	-nenhuma mudança (continua contando)
testar contagem em si	-Sinal(H)	-funcionando
transferir para o estado FIM	-Sinal(L)	-LEDR3 apagado -LEDR9 aceso -estado = 4 (FIM)

ir do estado FIM para o ESPERA	-Esperar 1 segundo	-LEDR3 aceso -LEDR9 apagado -estado = 8 (ESPERA)
testar permanência no estado ESPERA	-Sinal(L)	-nenhuma mudança
voltar para o estado CONTA	-Sinal(H)	- <i>display</i> volta a contar
voltar para o estado ESPERA	-Sinal(L), esperar 2 segundos	-funcionando
voltar para o estado INICIAL	-Liga(L)	- estado atual = 0 (INICIAL) -LEDR3 apagado
ir até o estado CONTA e depois resetar	-Liga(H), Sinal(H), esperar 2 segundos, Reset(H)	-contador para no valor (0x03) (passou do ponto de espera de 2 segundos)

Mais perguntas:

6. Quando SINAL é acionado por 5 segundos, qual é a saída apresentada na saída Q?

A saída é 0x05.

7. O que acontece quando o sinal LIGA é desativado antes do SINAL?

Se o estado atual for LIGADO, FIM ou CONTA, não há efeito, mas se for INICIAL ou ESPERA, transfere-se para o estado INICIAL (máquina de estados “desligada”).

8. Qual é a saída do contador se SINAL fica acionado por mais de 255 segundos?

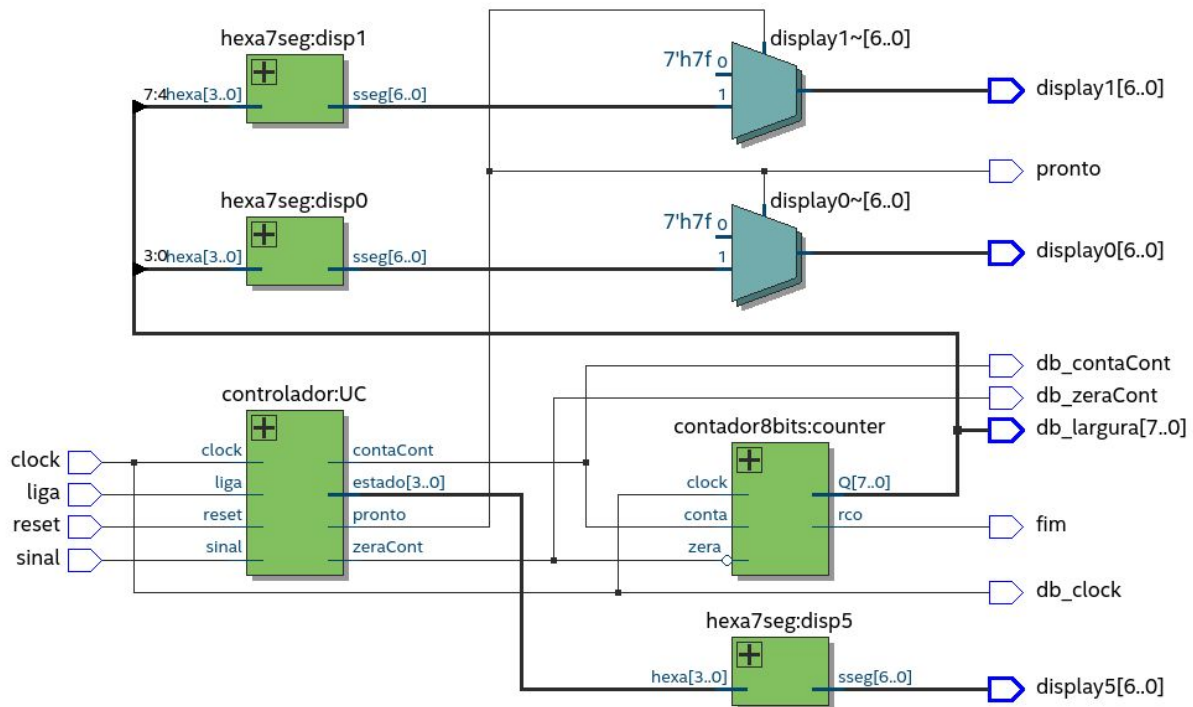
A saída chega a 0xFF e depois zera.

9. O que acontece quando SINAL é acionado várias vezes por intervalos de tempo diferentes antes do sinal LIGA ser desativado?

O estado da máquina é mudado quando SINAL fica desligado e o contador é desativado e acontece um ciclo entre os estados CONTA, ESPERA e FIM sempre que se aciona SINAL várias vezes.

Desafio - apenas amostrar a contagem quando a máquina sai do estado CONTA

Modificamos o arquivo `medidor_largura.vhd` de forma que a contagem apresentada nos *displays* 0 e 1 apenas quando a contagem fosse desabilitada. O RTL gerado pelo *software* Quartus pode ser visto abaixo.



A mudança necessária para essa nova funcionalidade foi adicionar um multiplexador para cada um dos *displays* de contagem, desligando os *displays* de 7 segmentos enquanto o sinal de fim não estivesse ativo e mostrando a contagem feita durante a permanência no estado CONTA quando o fim da contagem chegasse de fato.

Resultados alcançados

Nesta experiência, conseguimos emular na placa FPGA o circuito de contagem, usando uma máquina de estados finita. Conseguimos também implementar o desafio com êxito, tendo os dois circuitos funcionando como planejamos antes da experiência.

Pontos positivos:

- O nosso planejamento estava bem completo e pudemos realizar a montagem do circuito sem grandes dificuldades.

- Trabalhamos mais em grupo que nas últimas experiências, durante os quais dividimos mais as tarefas.

Pontos negativos:

- Cometemos um erro utilizando o Quartus - mudamos a pinagem da placa e não compilamos de novo o projeto -, o que nos fez perder bastante tempo conferindo o sinal de *clock* com o auxílio de um osciloscópio presente na nossa bancada.

- Esquecemos de fotografar o nosso circuito para anexar uma imagem ilustrativa neste relatório.

Lições aprendidas:

- Aprendemos a usar um osciloscópio para medir o sinal de *clock* do nosso gerador de pulsos.

- Descobrimos mais uma peculiaridade do Quartus quando se quer simular um circuito digital em um FPGA.