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

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.

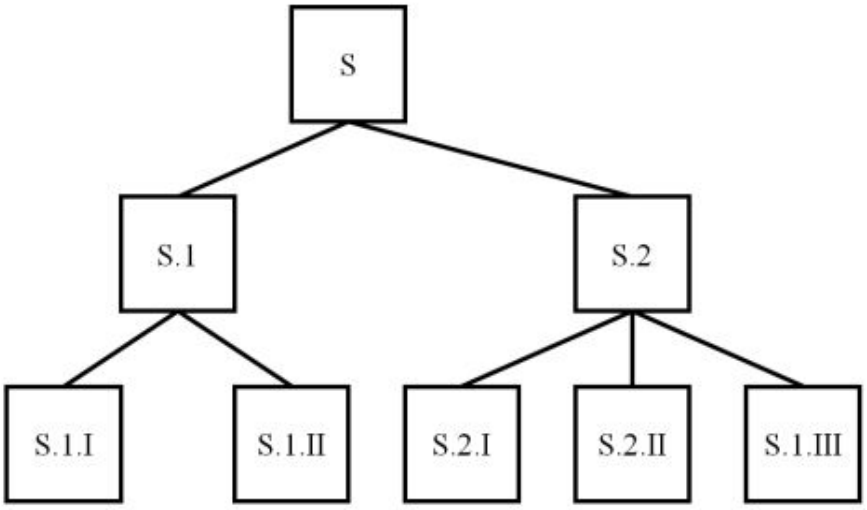


Figura 1 – Decomposição de um sistema em subsistemas.

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.

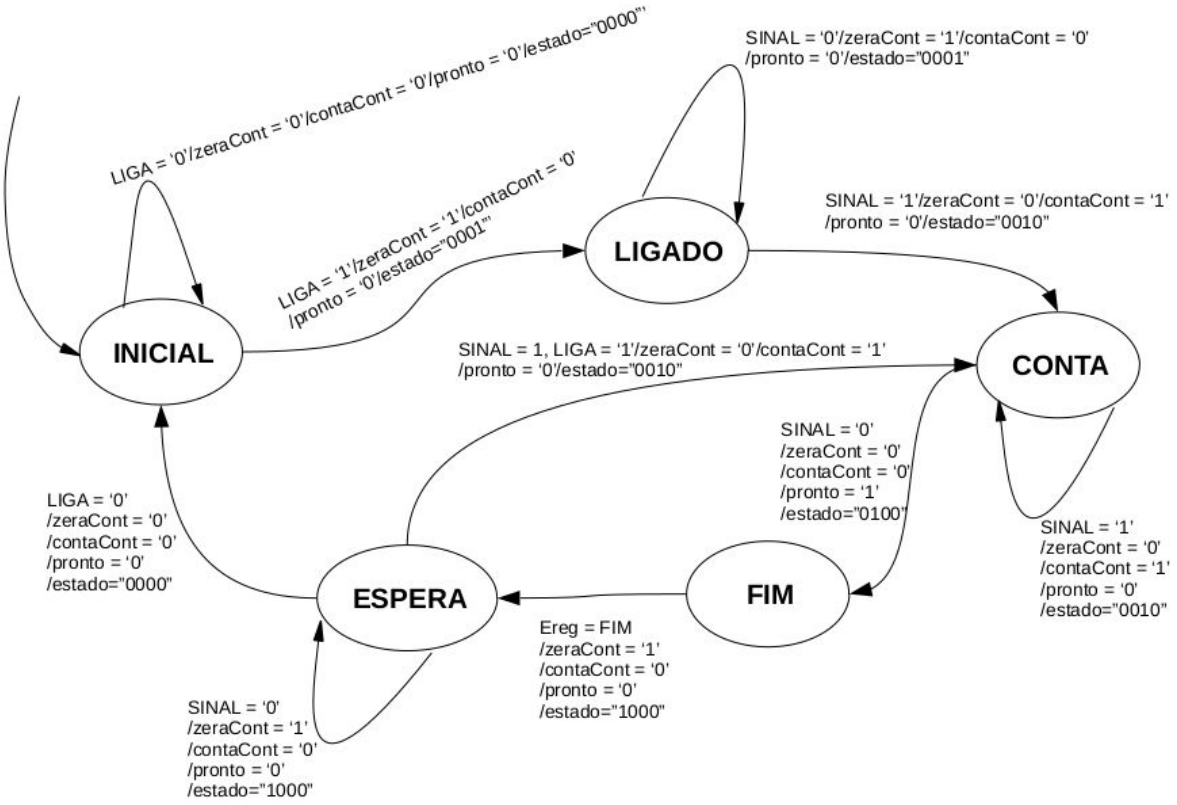
2.1.1. Estudo de uma Máquina de Estados

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

```
1  ----------------------------------------------------------------------
2  -- Arquivo    : controlador.vhd
3  -- Projeto    : Jogo do Tempo de Reacao
4  -- Data       : 09/01/2020
5  ----------------------------------------------------------------------
6  -- Descricao  : controlador (unidade de controle)
7  --              do medidor do tempo de reacao (largura de pulso)
8  --              descricao VHDL comportamental
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 ( clock, reset:      in  STD_LOGIC;
26            liga, sinal:       in  STD_LOGIC;
27            zeraCont, contaCont: out STD_LOGIC;
28            pronto:            out STD_LOGIC;
29            estado:            out STD_LOGIC_VECTOR(3 downto 0)
30          );
31 end controlador;
32
33 architecture controlador_arch of controlador is
34     type tipo_Estado is (INICIAL, LIGADO, CONTA, ESPERA, FIM);
35     signal Ereg, Eprox: tipo_Estado;
36 begin
37
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
56             when LIGADO => if SINAL = '0' then Eprox <= LIGADO;
57                            else               Eprox <= CONTA;
58                            end if;
59
60             when CONTA => if SINAL = '1' then Eprox <= CONTA;
61                           else               Eprox <= FIM;
62                           end if;
63
64             when FIM => Eprox <= ESPERA;
65
66             when ESPERA => if LIGA = '0' then Eprox <= INICIAL;
67                            elsif SINAL = '0' then Eprox <= ESPERA;
68                            else               Eprox <= CONTA;
69                            end if;
70
71             when others => Eprox <= INICIAL;
72         end case;
73     end process;
74
75     -- sinais de controle ativos em alto
76     with Ereg select
77         zeraCont <= '1' when LIGADO | ESPERA,
78                     '0' when others;
79     with Ereg select
80         contaCont <= '1' when CONTA,
81                      '0' when others;
82     with Ereg select
83         pronto <= '1' when FIM,
84                   '0' when others;
85     with Ereg select
86         estado <= "0000" when INICIAL,
87                   "0001" when LIGADO,
88                   "0010" when CONTA,
89                   "0100" when FIM,
90                   "1000" when ESPERA,
91                   "1111" when others;
92 end controlador_arch;
93
```

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

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



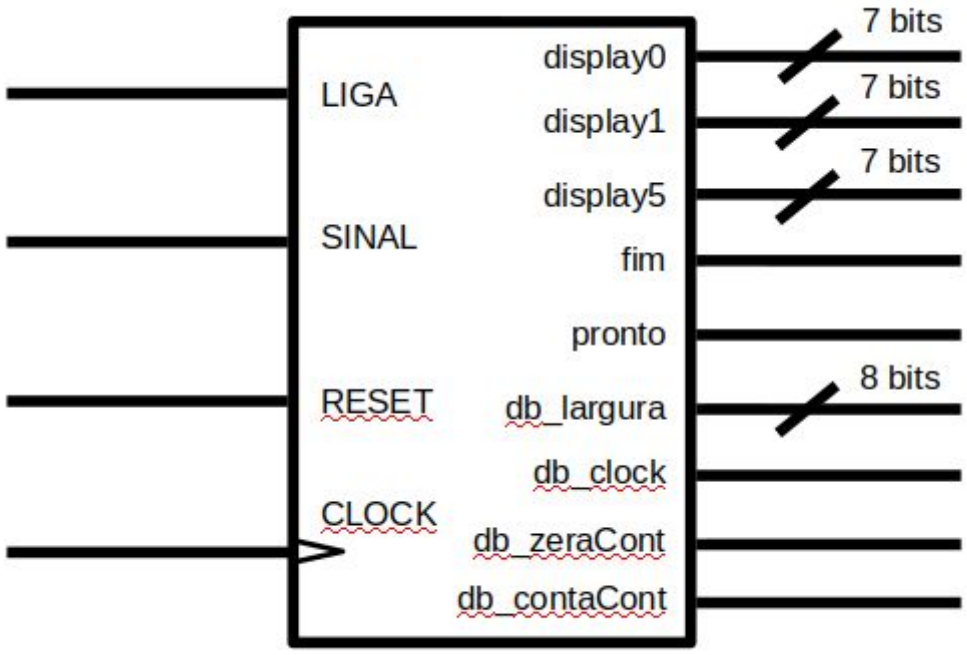
Estado	Comando
INICIAL	espera acionamento do sinal LIGA
LIGADO	espera início do PULSO de entrada
CONTA	while(pulso em 1) habilita contagem de largura
FIM	ativa sinal PRONTO
ESPERA	espera início do PULSO de entrada

Respostas às perguntas:

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

A máquina de estados descrita é de Mealy, pois a transição de um estado para o outro depende não só do estado atual, mas também das entradas da máquina.

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.

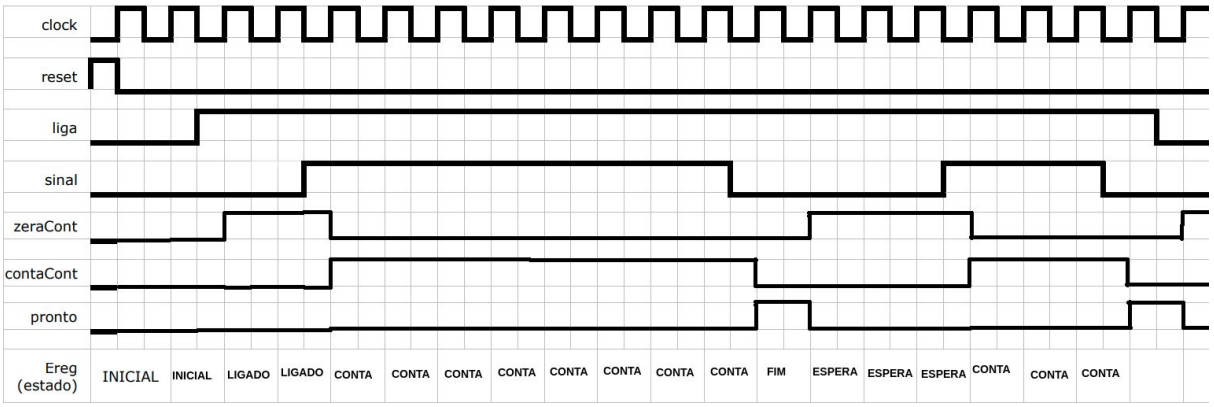
4. É possível a máquina de estados mudar para um estado que não seja um dos estados válidos (INICIAL ou LIGADO ou CONTA ou FIM ou ESPERA)? Explique sua resposta.

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.

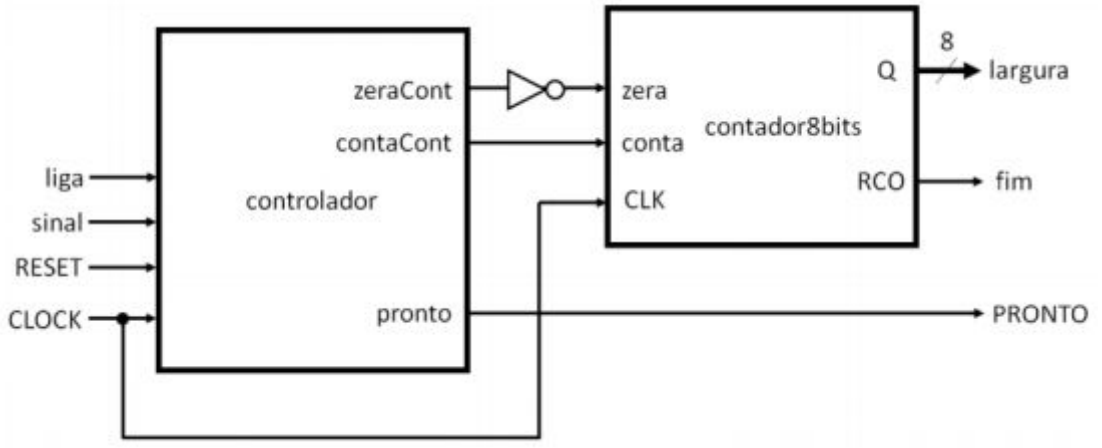
Não, o valor do sinal “estado” é restringido pelo multiplexador definido entre as linhas 85 e 91 do controlador.vhd, que define os valores que o “estado” deve assumir quando a máquina estiver em um estado válido e impõe o valor “1111” para qualquer outro estado (inválido) ao usar a sintaxe “others”.

Tabela de forma de onda esperada dos sinais do circuito:

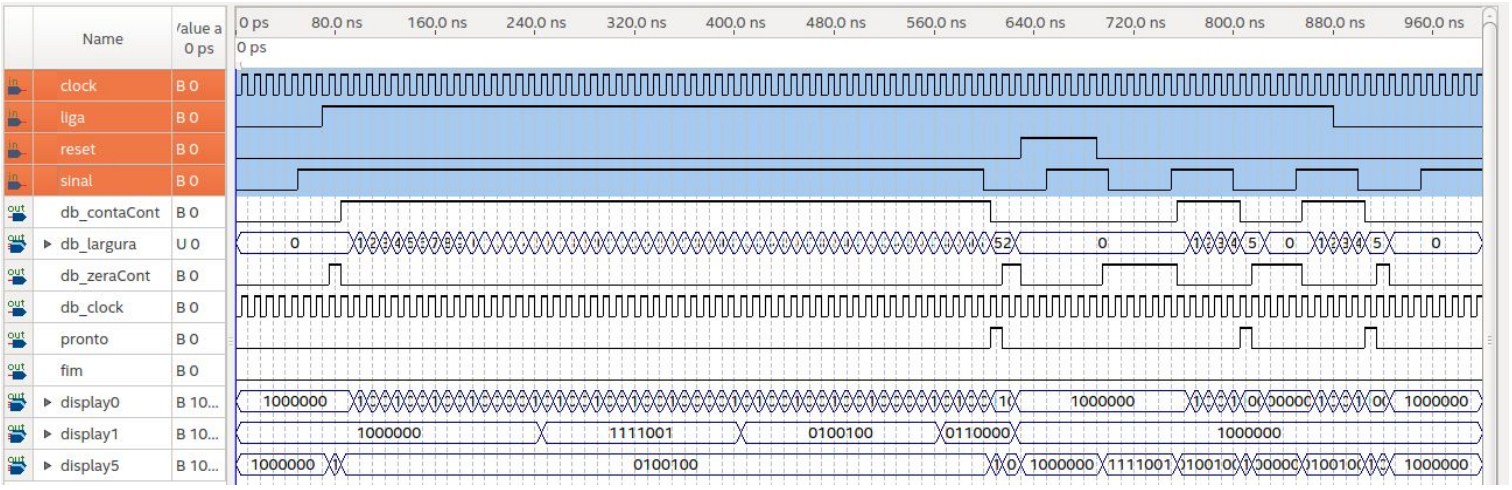


2.1.2. Projeto do Medidor de Largura de Pulso e Simulação

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.

Teste	Passos	Resultado observado
procurar estado inválido	-CLK (quantas vezes for necessário)	
inicializar a máquina de estados	-Reset(H), CLK	
transferir para o estado LIGADO	-Liga(H), CLK	
testar permanência no estado LIGADO	-Liga(H), Sinal(L), CLK (3 vezes)	
transferir para o estado CONTA	-Sinal(H), CLK	
testar permanência no estado CONTA	-Sinal(H), CLK (3 vezes)	
testar contagem em si	-Sinal(H), CLK (15 vezes)	
transferir para o estado FIM	-Sinal(L), CLK	
ir do estado FIM para o ESPERA	-CLK	
testar permanência no estado ESPERA	-Sinal(L), CLK (3 vezes)	
voltar para o estado CONTA	-Sinal(H), CLK	
voltar para o estado ESPERA	-Sinal(L), CLK, CLK	
voltar para o estado INICIAL	-Liga(L), CLK	
ir até o estado CONTA e depois resetar	-Liga(H), Sinal(H), CLK, CLK, Reset(H)	

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
largura	HEX00	PIN_U21	7 Seg Digit 0 [0]
largura	HEX01	PIN_V21	7 Seg Digit 0 [1]
largura	HEX02	PIN_W22	7 Seg Digit 0 [2]
largura	HEX03	PIN_W21	7 Seg Digit 0 [3]
largura	HEX04	PIN_Y22	7 Seg Digit 0 [4]
largura	HEX05	PIN_Y21	7 Seg Digit 0 [5]
largura	HEX06	PIN_AA22	7 Seg Digit 0 [6]
largura	HEX10	PIN_AA20	7 Seg Digit 1 [0]
largura	HEX11	PIN_AB20	7 Seg Digit 1 [1]
largura	HEX12	PIN_AA19	7 Seg Digit 1 [2]
largura	HEX13	PIN_AA18	7 Seg Digit 1 [3]
largura	HEX14	PIN_AB18	7 Seg Digit 1 [4]
largura	HEX15	PIN_AA17	7 Seg Digit 1 [5]
largura	HEX16	PIN_U22	7 Seg Digit 1 [6]
estado	HEX40	PIN_U20	7 Seg Digit 4 [0]
estado	HEX41	PIN_Y20	7 Seg Digit 4 [1]
estado	HEX42	PIN_V20	7 Seg Digit 4 [2]
estado	HEX43	PIN_U16	7 Seg Digit 4 [3]
estado	HEX44	PIN_U15	7 Seg Digit 4 [4]
estado	HEX45	PIN_Y15	7 Seg Digit 4 [5]
estado	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)	
inicializar a máquina de estados	-Reset(H), CLK	
transferir para o estado LIGADO	-Liga(H), CLK	
testar permanência no estado LIGADO	-Liga(H), Sinal(L), CLK (3 vezes)	
transferir para o estado CONTA	-Sinal(H), CLK	
testar permanência no estado CONTA	-Sinal(H), CLK (3 vezes)	
testar contagem em si	-Sinal(H), CLK (15 vezes)	
transferir para o estado FIM	-Sinal(L), CLK	
ir do estado FIM para o ESPERA	-CLK	
testar permanência no estado ESPERA	-Sinal(L), CLK (3 vezes)	
voltar para o estado CONTA	-Sinal(H), CLK	
voltar para o estado ESPERA	-Sinal(L), CLK, CLK	
voltar para o estado INICIAL	-Liga(L), CLK	
ir até o estado CONTA e depois resetar	-Liga(H), Sinal(H), CLK, CLK, Reset(H)	

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	

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

Rascunho: -mostrar como foi feito com prints

Teste	Passos	Resultado observado
procurar estado inválido	-CLK (quantas vezes for necessário)	
inicializar a máquina de estados	-Reset(H), CLK	
transferir para o estado LIGADO	-Liga(H), CLK	
testar permanência no estado LIGADO	-Liga(H), Sinal(L), CLK (3 vezes)	
transferir para o estado CONTA	-Sinal(H), CLK	
testar permanência no estado CONTA	-Sinal(H), CLK (3 vezes)	
testar contagem em si	-Sinal(H), CLK (15 vezes)	
transferir para o estado FIM	-Sinal(L), CLK	
ir do estado FIM para o ESPERA	-CLK	
testar permanência no estado ESPERA	-Sinal(L), CLK (3 vezes)	
voltar para o estado CONTA	-Sinal(H), CLK	
voltar para o estado ESPERA	-Sinal(L), CLK, CLK	
voltar para o estado INICIAL	-Liga(L), CLK	
ir até o estado CONTA e depois resetar	-Liga(H), Sinal(H), CLK, CLK, Reset(H)	

Perguntas: (testar no Waveform)

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 é 0xFF.

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