

## Roteiro do Projeto 2: Máquina de Estados

### 1. Introdução

O objetivo do projeto 2 do Laboratório de Sistemas Digitais é desenvolver um sistema digital composto por uma máquina de estados, cuja sequência de estados é definida por duas chaves. Os estados percorridos pela máquina de estados estão associados ao número de matrícula do aluno.

O código dos estados percorridos deve ser apresentado em LED's e em um display hexadecimal.

Esse laboratório tem como objetivo exercitar a metodologia de desenvolvimento de projetos de engenharia apoiada em computador (CAE). Esse processo envolve a compreensão do problema, seu planejamento, desenvolvimento da solução lógica, integração dos subsistemas, implementação no ambiente computacional, simulação, testes, implementação física e registro dos resultados.

### 2. Planejamento do Projeto

O processo de elaboração do projeto consiste em quatro etapas básicas:

#### **Aula 10 (AVA):**

- Discussão dos detalhes do projeto;
- Desenvolvimento do projeto lógico do sistema digital: tabelas verdade, mapas de Karnaugh, expressões lógicas simplificadas, tabela de transição de estados do sistema sequencial, equações dos biestáveis da máquina de estados;
- Apresentação individual do projeto lógico do sistema ao professor (**Avaliação Formativa**).

#### **Aula 11 (AVA):**

- Síntese do projeto completo (lógica de seleção, máquina de estados e decodificadores) no ambiente da ferramenta Quartus Prime-Lite (*Block Diagram*);
- Apresentação individual do diagrama esquemático lógico ao professor (**1 ponto**).

#### **Aula 12 (AVA):**

- Elaboração da simulação funcional e temporizada do projeto na ferramenta Quartus Prime;
- Apresentação individual das simulações funcional ao professor para comprovar o correto funcionamento do sistema digital (**2 pontos**);
- Incorporação dos outros módulos que complementam o projeto lógico.

#### **Aula 13 (CGI):**

- Atribuição de pinos e configuração do dispositivo lógico programável (FPGA);
- Apresentação individual do projeto completo e teste projeto na placa **DE10-Lite** para avaliação da implementação pelo professor (**1 ponto**);
- Entrega do relatório do projeto (conforme instruções no final do roteiro) (**2 pontos\***).

**\*Observação:** Na impossibilidade da realização da configuração do FPGA (por indisponibilidade do acesso ao **CGI**) o relatório passará a valer **3 pontos**.

Nas aulas de apresentação o aluno deverá estar preparado para responder às seguintes questões:

- ✓ Demonstrar que a lógica de controle desenvolvida opera de acordo com a especificação;
- ✓ Descrever e exemplificar o funcionamento da lógica de controle (casos de teste considerados);
- ✓ Descrever a função dos componentes utilizados;
- ✓ Descrever as atividades de integração realizadas para a obtenção do projeto;
- ✓ Modificar as formas de onda simuladas para representar situações específicas (casos de teste);

### 3. Descrição Funcional do Projeto

Cada aluno deve implementar uma Máquina de Estados com **seis estados** válidos (A a F), cuja sequência é baseada no seu número de matrícula. O estado inicial da máquina de estados e a sequência dos estados são definidos por duas chaves de controle (S1 e S0), conforme representado na **Tabela 1**.

**Tabela 1:** Tabela funcional da máquina de estados

Chaves de Entrada		Função da máquina de estados
S1	S0	
0	0	Permanece no estado inicial: A
0	1	Executa a sequência no sentido horário: (A→B→C→D→E→F→A...)
1	0	Executa a sequência no sentido anti-horário: (F→E→D→C→B→A→F...)
1	1	Permanece no estado inicial: F

Fonte: Autor.

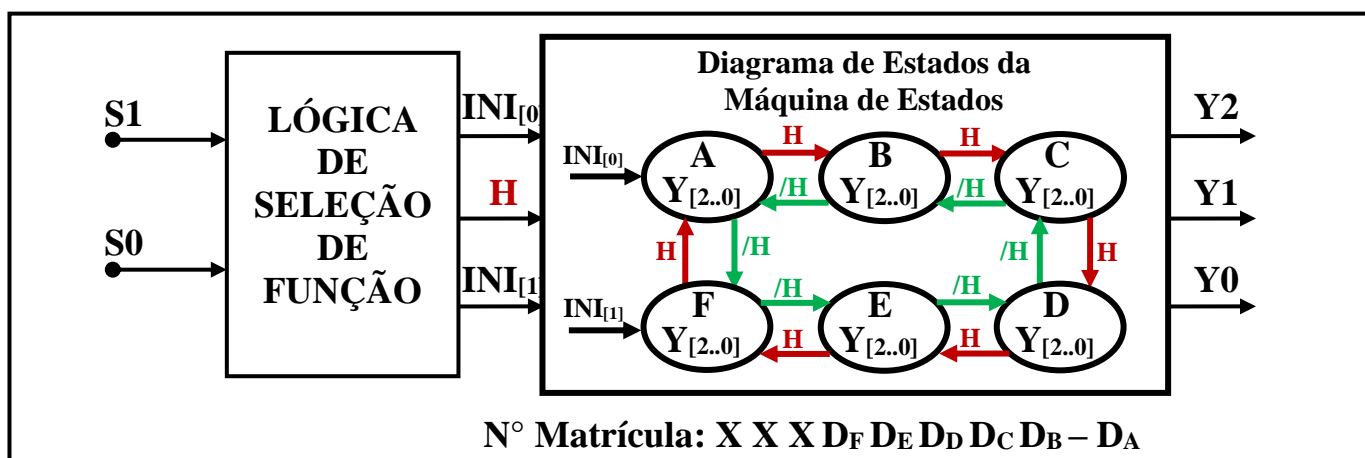
A qualquer momento que as chaves **S1** e **S0** forem ajustadas para um dos estados iniciais (**S1=0** e **S0=0** ou **S1=1** e **S0=1**) a máquina de estados deve ir para o respectivo estado inicial, **independente da existência da borda de clock**, ou seja, a **mudança para um dos estados iniciais é assíncrona**.

A mudança do sentido da sequência da máquina de estados só ocorre a partir de um dos estados iniciais (**A** ou **F**), ou seja, para alterar a sequência deve-se retornar a um dos estados iniciais (selecionando as chaves **S0** e **S1**) e a partir deste estado definir o sentido da sequência desejado.

Esse sistema contém uma saída **Y[2..0]** de três bits (**Y2**, **Y1**, **Y0**), sendo que cada saída ativa um LED vermelho. A saída **Y0** deve acender um LED (em nível lógico 1) quando o número do estado está entre 0 e 2. A saída **Y1** deve acender um LED o número do estado está entre 2 e 5; e a saída **Y2** deve acender um LED quando o número do estado estiver entre 5 e 7. Note que no estado 2 os LED's das saídas **Y0** e **Y1** devem estar acesos, assim como no estado 5 os LED's das saídas **Y1** e **Y2** devem estar acesos.

A **Figura 1** representa o diagrama de blocos do sistema, juntamente com a lógica de seleção de sentido da sequência.

**Figura 1:** Diagrama de estados da Máquina de Estados em função do número do aluno



Fonte: Autor.

### Condições particulares para cada aluno:

Cada aluno deve considerar como valor numérico dos estados **A, B, C, D, E e F** os **seis últimos dígitos do seu número de matrícula** (considerando o dígito de controle). A sequência horária dos estados **deve seguir a ordem numérica dos dígitos no número de matrícula**.

O número do estado inicial (**A**) corresponde ao dígito de controle do número de matrícula do aluno (**D<sub>A</sub>**).

Os demais estados são determinados **sequencialmente** pelos demais dígitos do número de matrícula **a partir do dígito de controle** (sequencialmente de **D<sub>B</sub>** a **D<sub>F</sub>**). Entretanto, **dois estados não podem ter o mesmo número**, assim, **caso ocorra a repetição de valores**, deve-se **substituir o dígito repetido pelo menor valor numérico que ainda não presente na sequência**.

Também se deseja limitar os valores numéricos dos estados ao valor 7, permitindo que sejam utilizados apenas três bits (**Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub>**) para representar os valores dos estados. Caso um dos dígitos do aluno seja maior que 7 deve ser utilizada o mesmo critério para valores repetidos: deve-se **utilizar o menor valor numérico que ainda não presente na sequência**.

A **Tabela 2** apresenta alguns exemplos da determinação dos estados **A** a **F** a partir do número de matrícula de cada aluno. No caso de dúvida **discuta com seu professor** a sequência de numeração.

**Tabela 2:** Exemplos de definição do valor dos estados **A** a **F** a partir do número do aluno.

Número de Matrícula	A	B	C	D	E	F
21212104-6	6	4	0	1	2	3
21116223-7	7	3	2	0	6	1
21229099-0	0	1	2	3	4	5
22210059-9	0	1	5	2	3	4
22214041-3	3	1	4	0	2	5
22115128-4	4	0	2	1	5	3

Fonte: Autor.

## 4. Requisitos de Implementação de Projeto (Aula 10 no AVA)

A Máquina de Estados (saídas **Q<sub>2</sub>, Q<sub>1</sub>, Q<sub>0</sub>, Y<sub>2</sub>, Y<sub>1</sub> e Y<sub>0</sub>**) deve ser desenvolvida utilizando-se o ambiente da ferramenta Quartus Prime Lite (©Intel Corporation). O projeto deve ser implementado no dispositivo lógico programável **MAX10** – modelo: **10M50DAF484C7G** da placa de desenvolvimento **DE10-Lite**.

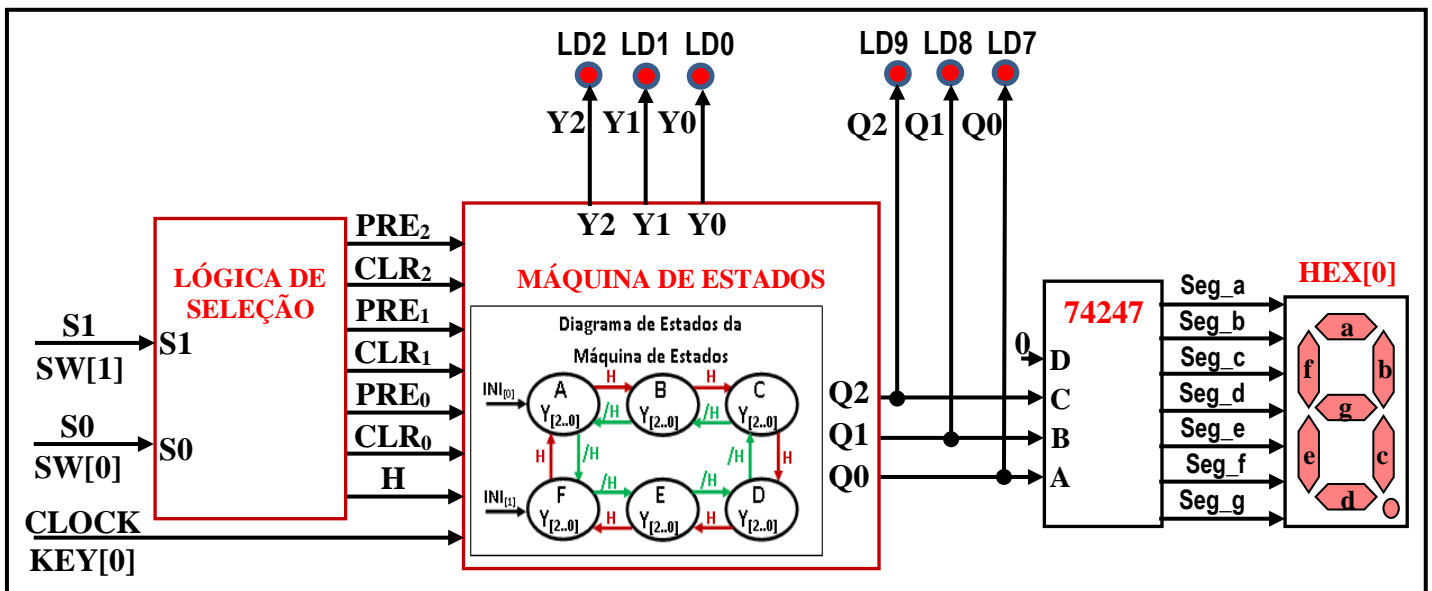
Cada aluno deve desenvolver **individualmente o projeto lógico** (Tabelas Verdade, Tabelas de Transição de Estados, equações lógicas minimizadas) da máquina de estados especificada. Esse projeto deve ser **desenvolvido e discutido com o professor na primeira aula de projeto (Aula 10)**.

Os requisitos específicos desse sistema digital são os seguintes:

- Conforme representado na **Figura 2** o projeto deve ter uma lógica de seleção de função, uma máquina de estados e um decodificador BCD para display de sete segmentos de ânodo comum (**74247**).
- A lógica da máquina de estados deve ser implementada com quatro biestáveis **JK**. Os estados da máquina de estados devem ser representados pelas saídas desses biestáveis **QC, QB e QA**, onde **QC é o bit mais significativo**.
- O estado da máquina de estados deve ser sinalizado de duas formas:
  - a) Em três LED's vermelhos (**LD9 a LD7**), que devem apresentar o código binário das saídas da máquina de estados (**Q<sub>2</sub> a Q<sub>0</sub>**);

- b) Em um display de sete segmentos (**HEX0**) que, através de um decodificador BCD/Display (**74247**), deve apresentar no display **HEX0** (de segmentos: **Seg\_a**, **Seg\_b**, **Seg\_c**, **Seg\_d**, **Seg\_e**, **Seg\_f** e **Seg\_g**) o número decimal do estado da máquina de estados;
- Os três bits da saída **Y2**, **Y1** e **Y0** devem ser apresentados em LED's vermelhos (**LD2** a **LD0**).

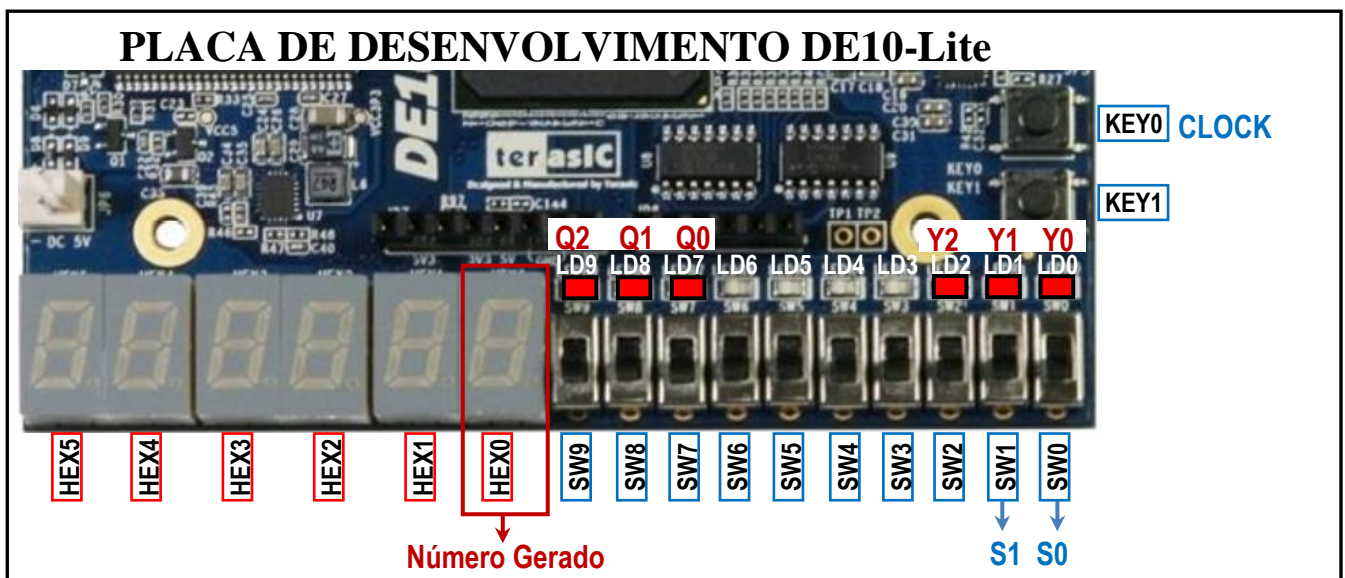
**Figura 2:** Diagrama de blocos global do sistema digital do projeto da máquina de estados.



Fonte: Autor.

- As chaves deslizantes **SW[1]** e **SW[0]** da placa de desenvolvimento **DE10-Lite** (**Figura 3**) devem ser utilizadas para selecionar o modo de funcionamento da máquina de estados (sinais **S1** e **S0**).
- O botão de contato momentâneo **KEY[0]** deve ser utilizado para gerar as transições (entre NL0 e NL1) do sinal **CLOCK**, sendo utilizado para comandar a evolução da máquina de estados, de modo síncrono e sequencial, entre os estados (**A** a **F**).

**Figura 3:** Interfaces do projeto da máquina de estados na placa **DE10-Lite**.



Fonte: Terasic, 2020 (adaptado).

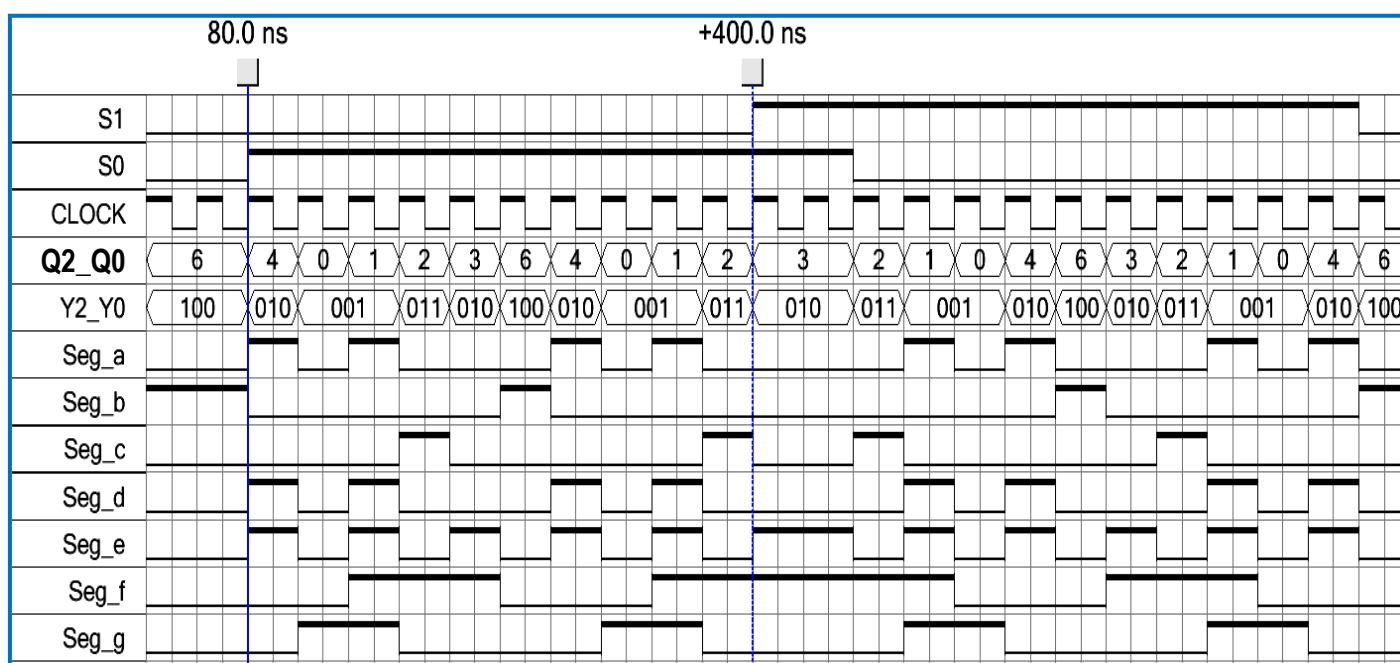
## 5. Apresentação da Implementação do Projeto Lógico (Aula 11 no AVA)

Na **Aula 11** o aluno deve apresentar a **realização do projeto lógico** do sistema, considerando a **Lógica de Seleção** e a **lógica da Máquina de Estados**. O circuito com esses dois componentes deve ser implementado no ambiente da ferramenta **Quartus Prime Lite**. Veja os procedimentos do arquivo: **Tutorial\_Quartus\_Prime\_Captura\_de\_Esquemático** (PRATES, 2019), disponível no Moodle.

## 6. Apresentação da Simulação do Projeto (Aula 12 no AVA)

- Na **Aula 12** o aluno deve apresentar a simulação do circuito com a **Lógica de Seleção** e a **lógica da Máquina de Estados** no ambiente da ferramenta Quartus Prime - Lite, conforme os procedimentos do arquivo: **Tutorial\_Quartus\_Prime\_Captura\_de\_Esquemático** (PRATES, 2019), disponível no Moodle;
- Cada aluno deve **apresentar ao professor** a simulação do projeto da Máquina de Estados de modo a comprovar, através das formas de onda, a correta operação da lógica do sistema digital.
- As formas de onda simuladas devem apresentar todos os sinais de entrada e saída do projeto, com as seguintes características, conforme representado na **Figura 4**.
  - **Tempo total de simulação: End Time = 1us;**
  - **Passo da grade de visualização: Grid Size = 20ns;**
  - **Período do sinal CLOCK: 40ns;**
  - **Sinais S1 e S0:** na condição **00** por **80ns**; na condição **01** por **400 ns**; na condição **11** por **80 ns**; e na condição **10** por **400 ns**;
  - **Saídas Q2, Q1 e Q0:** agrupadas com representação em Decimal Sem Sinal, sendo **Q2** o bit mais significativo (cuidado com a ordem dos bits no agrupamento);
  - **Saídas Y2, Y1 e Y0:** agrupadas com representação em Binário, sendo **Y2** o bit mais significativo (cuidado com a ordem dos bits no agrupamento).

**Figura 4:** Exemplo de resultado de simulação da Máquina de Estados  
(aluno com estados: A=6, B=4, C=0, D=1, E=2, F=3)



Fonte: QUARTUS, 2020 (adaptado).



## 7. Conclusão do Projeto (**Aula 13 no CGI**)

Na **Aula 13** (no **CGI**) o aluno deve incluir todos os componentes (decodificadores **74247** e respectivos pinos), assim como a atribuição da numeração dos pinos do FPGA para cada sinal de entrada e saída do projeto na placa **DE10-Lite**. A associação dos sinais do projeto da máquina de estados com os pinos do FPGA da placa **DE10-Lite** é apresentada na **Tabela 3**.

**Tabela 3:** Numeração dos pinos para os sinais de entrada e saída do FPGA da placa **DE10-Lite**.

SINAL	PINO	DISPLAY HEX0	PINO	LED	PINO
KEY[0] – CLOCK	B8	HEX0 – Seg_g	C17	LD[9] – Q2	B11
SW[1] – S1	C11	HEX0 – Seg_f	D17	LD[8] – Q1	A11
SW[0] – S0	C10	HEX0 – Seg_e	E16	LD[7] – Q0	D14
		HEX0 – Seg_d	C16	LD[2] – Y2	A10
		HEX0 – Seg_c	C15	LD[1] – Y1	A9
		HEX0 – Seg_b	E15	LD[0] – Y0	A8
		HEX0 – Seg_a	C14		

Fonte: Terasic, 2020 (adaptado).

Os procedimentos para a execução da atribuição de pinos do **FPGA** da família **MAX10** - modelo: **10M50DAF484C7G** a um projeto lógico no ambiente do Quartus Prime Lite estão apresentados na **seção 7.3** do arquivo: **Tutorial\_Quartus\_Prime\_Captura\_de\_Esquemático** (PRATES, 2019), disponível no Moodle.

Os demais pinos disponíveis para chaves e LED's da placa **DE10-Lite** estão disponíveis no arquivo: **Associação de Pinos do FPGA da Placa DE10-Lite** (disponível nas referências de laboratório no Moodle).

**Atenção:** Não deixe conectores de saída (OUTPUT) **sem atribuição de pino**. O Quartus Prime associa automaticamente todos os conectores de saída a algum pino de I/O, **podendo ocorrer conflito** de um pino escolhido pela ferramenta com um sinal já utilizado na placa **DE10-Lite**.

Tendo acesso ao **CLE** o aluno pode utilizar a **Sala de Projetos (sala D3-07 ou D3-09)** para testar seu projeto antes da aula de apresentação (a chave da sala e a placa **DE10-Lite** podem ser solicitadas no almoxarifado do **CLE**).

## 8. Critérios de Elaboração do Relatório do Projeto:

O projeto deve ser desenvolvido **individualmente**. Projetos copiados, **total ou parcialmente**, não serão considerados válidos, sendo atribuído zero para todos os projetos copiados;

As avaliações das **apresentações e os relatórios serão individuais**, sendo que cada aluno deve **demonstrar o perfeito conhecimento de todas** as etapas do projeto e teste;

O relatório deve ser entregue no formato de **arquivo em PDF (enviado pelo MOODLE)** e deve conter a documentação detalhada do projeto, **permitindo que este seja entendido e executado por um outro projetista**. O relatório deve apresentar, **no mínimo**, os seguintes itens:

1. **Capa** com o nome, número do aluno e turma;

### 2. Parte 1: Descrição do Problema:

A introdução deve apresentar um resumo dos objetivos do projeto com a especificação das características particulares da implementação realizada pelo aluno (**diagrama de estados** com os dígitos do número de matrícula e os valores dos estados).

### 3. Parte 2: Descrição da Realização do Projeto:

Devem ser descritos os vários elementos que compõem o projeto. Devem ser apresentados **em detalhe no mínimo** os seguintes elementos:

- **Tabelas Verdade**: com as **condições da lógica de seleção de função** para definição das **expressões lógicas de Preset e Clear** e do **sentido da sequência de estados** (lógica do sinal H);
- **Tabela de Transição de Estados**: com as **condições da lógica de transição para as entradas dos biestáveis JK**, conforme o **diagrama de estados apresentado**;
- **Mapas de Karnaugh**: com as **expressões resultantes do projeto das interligações das entradas dos biestáveis JK** para as sequências de estados do aluno;
- **Diagrama Esquemático-Lógico** completo do projeto do sistema, mostrando a interligação entre todos os elementos do projeto e os pinos dos sinais de entrada e saída (gerada com “Print Screen” do editor de blocos do Quartus Prime – arquivo.bdf). Veja exemplo no **Anexo**;
- **Formas de onda da simulação funcional** apresentadas nas aulas de projeto (geradas com “Print Screen” do simulador do Quartus Prime – arquivo.vwf);

### 4. Conclusão:

Na conclusão do relatório podem ser comentados: os objetivos do Projeto 2; as dificuldades encontradas e como foram superadas; o que o aluno faria de diferente se tivesse que executar esse projeto novamente.

**ATENÇÃO**: O *upload* de arquivos no Moodle é de **inteira responsabilidade do aluno**, que **deve verificar** se os arquivos foram efetivamente postados após o *upload*. **Não serão aceitos** arquivos postados **fora do horário** especificado para *upload* da atividade

## 9. Referências Bibliográficas

TOCCI, Ronald J; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas Digitais: Princípios e Aplicações**. Revisão técnica: Renato Giacomini. Tradução: Jorge Ritter. 11. Ed. São Paulo: Pearson Prentice Hall, 2011.

QUARTUS – Intel Quartus Prime – Lite, Version 16.1. Intel-FPGA (©Intel Corporation). Disponível em: (<https://fpgasoftware.intel.com/?edition=lite>). Acesso em: 07/07/2020.

PRATES, R. R. – Tutorial de Quartus Prime para Projeto de CPLD/FPGA (baseado em Captura de Esquemático). Programa de Iniciação Didática – Centro Universitário FEI, 2019.

TERASIC - DE10-LITE Board - User Manual. (©Terasic Inc). Disponível em: (<https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=234&No=1021&PartNo=4>). Acesso em: 06/07/2020.

CURSO DE CIÊNCIA DA COMPUTAÇÃO

CE3512 – LABORATÓRIO DE SISTEMAS DIGITAIS

PROJETO 2 - 1º SEMESTRE DE 2021

**ANEXO – Exemplo de Diagrama Esquemático do Projeto (aluno com estados: A=6, B=4, C=0, D=1, E=2, F=3)**

Fonte: QUARTUS, 2020 (adaptado).

