	Universidade Federal de Minas Gerais – Escola de Engenharia	
	Departamento de Engenharia Eletrônica – Laboratório de Sistemas Digitais	
	Autor: Equipe de professores de Laboratório de Sistemas Digitais	
Aulas 11 a 15	Versão 1.3	Concepção, projeto e documentação de um Sistema Digital RTL com FSM Alto Nível

## TRABALHO FINAL EM EQUIPE – PROJETO RTL

### Lógica Combinacional

Recordando, a metodologia de projeto de circuitos combinacionais está resumida na Tabela 1 a seguir:

**TABELA 1: METODOLOGIA PARA PROJETO DE CIRCUITOS COMBINACIONAIS**

Passo 1	Identifique a função	Crie a tabela verdade ou equações, <i>o que for mais natural para o problema em questão</i> , para descrever o comportamento do circuito lógico combinacional.
Passo 2	Converta para equações	Esse passo só é necessário se você no passo 1 descreveu o problema em uma tabela verdade. Crie uma equação para cada saída, fazendo a soma de produtos, por exemplo, ou outro método que julgar mais adequado. Simplifique as equações se necessário, usando álgebra booleana ou mapas de Karnaugh.
Passo 3	Implemente as equações	Para cada saída (equação), crie o circuito correspondente. Em VHDL codifique as equações ou use conexão entre portas/funções lógicas.

### Lógica Sequencial


A Metodologia em Cinco Passos, mostrada na Tabela 2, é recomendada para projetar Controladoras, i.e., as máquinas de estados finitos de baixo nível (FSM booleanas).

**TABELA 2: METODOLOGIA PARA PROJETO DE CONTROLADORAS**

Passo 1	Capture o comportamento da FSM	Desenhe o diagrama da FSM booleana que descreva seu comportamento. Lembre-se de se certificar que apenas uma condição de transição de estado seja verdadeira e que haja pelo menos uma condição verdadeira para cada estado a cada ciclo de <i>clock</i> . <sup>1</sup>
Passo 2	Crie a arquitetura	Neste curso vimos a Arquitetura Canônica (arquitetura de Huffman-Moore). No caso de programação em VHDL, vimos a arquitetura na forma canônica na modalidade <i>TPM – Two-process-model</i> (modelo de dois processos).
Passo 3	Codifique os estados	Diferentes escolhas de codificação de estados levarão a diferentes lógicas combinacionais no passo seguinte, com maior ou menor complexidade na implementação, dependendo de suas escolhas. No caso da metodologia sugerida aqui, atribua estados usando a palavra “type” em VHDL para definir um tipo que assume os estados desejados. Você pode definir o mapeamento para uma codificação específica, como o <i>one-hot-encoding</i> por exemplo ou deixar a atribuição binária de cada estado ser feita automaticamente pelo compilador. Crie então um <i>process (clear, clock)</i> para atualizar o estado.
Passo 4	Crie a tabela de transição de estados e de saída	Nesta etapa são criadas a <i>Tabela de Transição de Estado</i> e a <i>Tabela de Saída</i> , que são tabelas-verdade contextualizadas para a lógica necessária à FSM, de forma que gere os sinais de próximo estado e os sinais de saída, respectivamente, a partir do estado atual e das entradas. Essas tabelas são a realização do projeto da lógica combinacional da FSM e podem ser implementadas aqui na forma de equações booleanas e estruturas de decisão de fluxo codificadas em VHDL (passo 5). Na metodologia, esse é o momento de aplicar métodos de simplificação (mapas de <i>Karnaugh</i> , álgebra booleana, algoritmos, etc.), mas atualmente grande parte dessas simplificações é executada pela própria ferramenta de síntese.
Passo 5	Implemente a lógica combinacional (metodologia da Tabela 1)	Implemente a lógica combinacional usando qualquer método apropriado – no caso da metodologia TPM, implementa-se um <i>process (entradas, estado atual)</i> para a rede combinacional. Certifique-se que este <i>process</i> não contenha <i>latches</i> gerados inadvertidamente (afinal é um circuito combinacional !!!). <sup>1,2</sup>

<sup>1</sup> Ao codificar em VHDL lembre-se: codifique também o óbvio! Indefinições nas transições também podem gerar *latches* espúrios.

<sup>2</sup> *Latches* presentes na região combinacional do circuito consistem em um erro de codificação e levam a comportamentos imprevisíveis. Uma das vantagens de se usar a metodologia *TPM* é poder isolar facilmente a parte combinacional e desaparecer com os *latches* indesejáveis. Sugestão: você

	Universidade Federal de Minas Gerais – Escola de Engenharia	
	Departamento de Engenharia Eletrônica – Laboratório de Sistemas Digitais	
	Autor: Equipe de professores de Laboratório de Sistemas Digitais	
Aulas 11 a 15	Versão 1.3	Concepção, projeto e documentação de um Sistema Digital RTL com FSM Alto Nível

A metodologia na Tabela 3 é a metodologia RTL, para implementação de Máquinas de Estado de Alto Nível usando controladoras e caminho de dados.

**TABELA 3: METODOLOGIA RTL (PARA PROJETAR FSM DE ALTO NÍVEL)**

Passo 1	Capture o comportamento da FSM alto nível	Descreva o comportamento desejado do sistema na forma de uma máquina de estados de alto nível. A máquina é de “alto nível” porque as condições para as transições e as ações dos estados são mais do que simplesmente operações booleanas.
Passo 2	Crie o caminho de dados (bloco operacional)	Partindo da máquina de estados de alto nível do passo anterior, crie um bloco operacional (caminho de dados) capaz de realizar as operações que envolvem dados e testar condições de transição de alto nível, convertendo-as em informações e comandos booleanos a serem ligados à controladora correspondente.
Passo 3	Conecte o caminho de dados à controladora	Conecte o bloco “caminho de dados” ao bloco “controladora” (FSM baixo nível com transições e operações de alto nível convertidas para operações booleanas). Conecte também as entradas e saídas a tais blocos e destes com o mundo exterior.
Passo 4	Obtenha a FSM da controladora (Metodologia da Tabela 2)	Converta a máquina de estados de alto nível na máquina de estados finitos de baixo nível. Para isso, substitua as operações que envolvem dados e as transições de alto nível por sinais de controle booleanos, que são ativados ou lidos pela controladora em colaboração com o caminho de dados.

Referência: Frank Vahid. *Sistemas Digitais: Projeto, Otimizações e HDLs*. Bookman. 2008.

Disponível no Google Books em:

<https://books.google.com.br/books?id=8xT9sD0kpfUC&lpg=PR2&dq=Frank%20vahid&hl=pt-BR&pg=PR3#v=onepage&q=Frank%20vahid&f=false>

Projeto de circuitos combinacionais (Tabela 1):

- Pág. 46 a 103 e pág. 182 a 198 e pág. 207 a 221

Projeto de circuitos sequenciais e controladora (Tabela 2):

- Pág. 111 a 156 e pág. 166 a 182 e pág. 198 a 207 e pág. 221 a 226.

Metodologia RTL, passos 1 e 2 (Tabela 3).

Captura do diagrama da FSM Alto Nível e Projeto do Caminho de Dados:

- Pág. 243 a 253 e pág. 493 a 495.

Metodologia RTL, passos 3 e 4 (Tabela 3).

Derivação da controladora (FSM baixo nível) a partir da FSM Alto Nível e do caminho de dados.


Interconexão entre controladora e caminho de dados

- Pág. 253 a 255 e pág. 498 a 502.

## **MATERIAL A SER ENTREGUE:**

Espera-se que o grupo pense na descrição, aplicação, projeto, implementação e simulação de uma FSM alto nível de livre escolha onde, na sua realização, o caminho de dados deve ter pelo menos 4 entradas digitais e pelo menos 4 saídas digitais, sendo no mínimo 2 entradas de 16 a 32 bits e no mínimo 2 saídas de 16 a 32 bits. A controladora deve implementar uma máquina de estados finitos de pelo menos 8 estados.

pode, por exemplo, comentar a parte sequencial e inspecionar o circuito da parte combinacional usando o RTL-Viewer do Quartus. Se houver algum *latch*, corrija seu código.

	Universidade Federal de Minas Gerais – Escola de Engenharia	
	Departamento de Engenharia Eletrônica – Laboratório de Sistemas Digitais	
	Autor: Equipe de professores de Laboratório de Sistemas Digitais	
Aulas 11 a 15	Versão 1.3	Concepção, projeto e documentação de um Sistema Digital RTL com FSM Alto Nível

Cada equipe deverá desenvolver o projeto que propôs e documentá-lo da seguinte forma:

1. Descrição do problema e da solução implementada, que compreende:
  - a. Descrição do produto e do ambiente
  - b. Requisitos do sistema e entradas/saídas.
  - c. Diagrama de blocos do sistema como um todo, acompanhado de sua descrição textual <sup>3</sup>.
  - d. Máquina de estado de alto nível:
    - i. Importante: pede-se aqui a FSM que o grupo usou para “capturar” a ideia de funcionamento do sistema <sup>4</sup>.
    - ii. Não é pedida aqui a controladora (FSM de baixo nível, booleana, incompleta, emitida pelo “State Machine Viewer” do Quartus).
  - e. Desenho do caminho de dados com ilustração de todos seus componentes e a forma de interligação entre eles.
  - f. Sub-diagrama de blocos conceitual mostrando os dois blocos: caminho de dados e controladora. Este diagrama trará as diversas linhas de interligação entre estes blocos e deles com o mundo exterior (entradas e saídas). Todas as linhas deverão ter os mesmos nomes usados na codificação VHDL. Portanto procure usar mnemônicos os mais autoexplicativos possíveis.
2. Programas VHDL, acompanhados dos respectivos circuitos e máquinas de estado gerados (disponível em Tools > Netlist Viewers > RTL Viewer e (...) > State Machine Viewer).
3. Formas de onda da simulação (estímulos e saídas), para no mínimo um cenário de teste, acompanhado de explicação textual da simulação (se possível acompanhada de balões numerados que facilitem a identificação das transições descritas).
4. Diretório de arquivos do projeto zipados.

A documentação postada deverá conter as correções eventualmente necessárias observadas durante a validação quando da demonstração prática ao professor. O grupo deverá postar no Moodle a documentação correspondente ao projeto em no prazo estipulado pelo professor:

#### Entrega 1:

- Descrição do problema e da solução implementada
- Diagrama de Máquina de Estados de Alto Nível (diagrama conceitual)
- Desenho conceitual de projeto do Caminho de Dados mostrando interligação entre seus componentes internos.
- Descrição e pelo menos uma simulação de cada componente individual que compõe o caminho de dados.

#### Entrega 2:

- Projeto da Controladora e Interligação com o Caminho de Dados.
- Simulação de um cenário demonstrando funcionamento do conjunto Controladora + Caminho de Dados
- Postar documentação final completa conforme definido acima com as eventuais correções após demonstração.

<sup>3</sup> O objetivo do diagrama de blocos é descrever as partes físicas componentes do sistema e a forma como se interligam. Não se trata de entregar os RTL Views gerados pelo Quartus e sim desenhar os blocos que compõem sua ideia de sistema, como entradas e saídas, kits utilizados, sensores externos, etc.

<sup>4</sup> Diagrama conceitual de captura da FSM: trata-se do diagrama que o grupo usou para conceber o projeto e não aquele gerado automaticamente pelo Quartus. A critério do grupo, pode-se usar qualquer ferramenta para documentar este diagrama. Para citar um exemplo, existem ferramentas que seguem o padrão UML – Unified Modeling Language ([www.uml.org](http://www.uml.org)). Por exemplo, na iniciativa de código livre temos a ferramenta StarUML: [staruml.sourceforge.net](http://staruml.sourceforge.net). A UML conta em seu conjunto com inúmeros diagramas disponíveis, dentre eles o subconjunto de diagramas comportamentais (*behavioral diagrams*). Dentre eles está o diagrama de estados (*Statechart Diagram* ou *State Machine Diagram*). Para projetistas de sistemas embarcados, temos como exemplo a ferramenta de modelagem da Quantum Leaps: <https://www.state-machine.com/qm/sm.html>. Você pode visitar o site [en.wikipedia.org/wiki/List\\_of\\_UML\\_tools](http://en.wikipedia.org/wiki/List_of_UML_tools) para ver uma comparação entre diversas ferramentas UML gratuitas e pagas.