

Laboratório sobre Minimização Automatizada de Funções Combinacionais e Diagramas de Transição de Estados

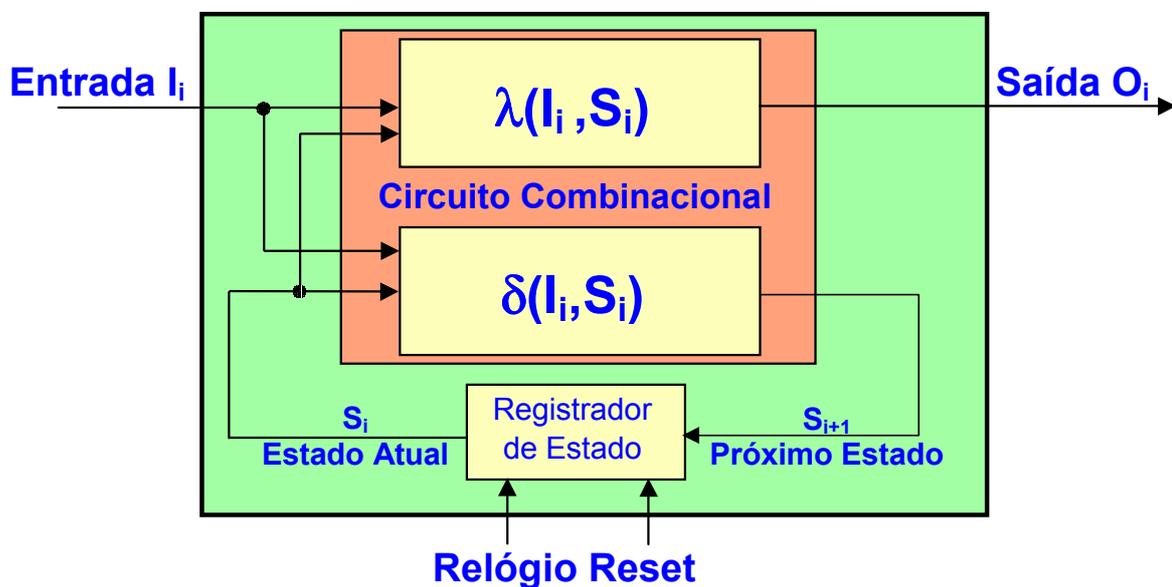
Prática: O exemplo da máquina de venda de refrigerantes em lata: especificação e projeto usando diagramas de transição de estados

Recursos: Sistema de CAD Foundation da empresa Xilinx, Inc., e o programa de domínio público Espresso, implementado na Universidade de Berkeley

Parte I - Objetivos

O **Laboratório sobre Circuitos Seqüenciais e Máquinas de Estados Finitas** envolveu a segunda das duas classes fundamentais de circuitos digitais, os **circuitos seqüenciais**. Lembre-se que qualquer circuito digital pode ser modelado como um circuito seqüencial, desde um flip-flop até um Pentium III (Leia, Pense e Responda: até os circuitos combinacionais podem ser vistos como um caso especial de circuito seqüencial. Como isto pode ser feito?). Foi dito no Laboratório citado que a atenção seria restrita apenas aos circuitos seqüenciais síncronos (aqueles onde todas as ações internas são controladas por um sinal especial denominado, normalmente, relógio ou clock). Ainda naquela ocasião, um modelo abstrato útil foi introduzido para representar qualquer circuito seqüencial, as **máquinas de estados finitas** ou **FSMs** (sigla derivada da nomenclatura em inglês, *Finite State Machines*). A seguir, introduziu-se um modelo estrutural geral para representar FSMs síncronas (existem também modelos para FSMs assíncronas, mas estes estão fora do escopo deste curso). Foi dito ainda que a partir deste modelo, e de uma codificação dos estados internos da FSM, basta gerar o hardware combinacional que implementa os dois conjuntos de funções (λ e δ) para ter trivializada a implementação da FSM. O modelo aparece abaixo.

Modelo Estrutural para FSMs Síncronas



O objetivo específico deste laboratório é, a partir da especificação formalizada da máquina de vender refrigerantes em lata, via sua tabela de transição de estados, **apresentar ferramentas de projeto alternativas para a implementação**. A dificuldade de implementação durante o uso de editores de esquemáticos surge, muitas vezes, por se lidar com níveis de abstração baixos para representar e manipular as funções. A primeira parte deste laboratório lidará com ferramentas para **facilitar a manipulação de funções Booleanas complexas**, e a segunda parte com ferramentas para **aumentar o nível de abstração ao representar circuitos seqüenciais**. A ferramenta básica de implementação otimizada proposta anteriormente foi o mapa de Karnaugh computado

manualmente, que é inadequado e altamente propenso a erros quando o tamanho das funções ultrapassa 5 ou 6 variáveis. Ver-se-á aqui o emprego de um programa de minimização automatizada de múltiplas funções, o software **espresso**. Mesmo usando programas para automaticamente minimizar funções, a especificação destas é complicada e igualmente propensa a erros. Se houvesse uma ferramenta que permitisse a representação direta do conceito de estado, e da relação de transição entre estes e de sua influência no cálculo das saídas, o trabalho seria facilitado. Este será o tema da segunda parte deste laboratório, o uso de um editor de **Diagramas de Transição de Estados**, uma forma gráfica para representar FSMs síncronas, disponível no CAD Foundation.

Parte II – Melhorando a Capacidade de implementar Funções Booleanas - Espresso

O programa **espresso**, desenvolvido na década de 80 por pesquisadores da Universidade de Berkeley (Califórnia, USA) é de domínio público, inclusive sendo seu código fonte disponível. Este programa recebe uma especificação de **n** funções discretas, todas dependentes de um conjunto de **p** variáveis discretas, e fornece como saída uma descrição equivalente sob a forma de uma soma de produtos, onde o número de produtos é minimizado em relação à descrição original (se possível), e onde há um compartilhamento máximo de utilização de produtos por funções distintas.

Este programa deve estar instalado nas máquinas do laboratório sob o diretório **C:\XSTOOLS\BIN**. Neste diretório deve existir o programa (**espresso.exe**), um diretório com diversos exemplos (diretório **PLA_ex**). Além destes, devem existir os dois manuais da ferramenta (arquivos **user_manual.txt** e **io_manual.txt**). Espresso é um programa DOS, que trabalha a partir de uma interface de linha de comando. Como o programa opera com entrada e saída padrão, a maneira mais simples de executá-lo é digitar na linha de comando DOS o seguinte:

DOS_Prompt> espresso Esta linha dispara o espresso e espera que o usuário entre com as funções linha a linha via teclado (ou seja, a entrada padrão), até a linha contendo **.e** (comando que indica o final da descrição). A seguir o programa é executado e o resultado é jogado na saída padrão (normalmente a tela). Para usar arquivos de entrada e saída digita-se a linha DOS:

DOS_Prompt> espresso arq_entrada >arq_saída Neste caso, o programa lê a descrição do arquivo **arq_entrada** e coloca a saída minimizada no arquivo **arq_saída**.

Nesta parte, usem o arquivo disponível na homepage da disciplina (arquivo **Refri.pla.parcial**), contendo o início da descrição das funções combinacionais da máquina de vender refrigerantes da aula passada. Tomem este arquivo, e leiam a introdução em comentários sobre o formato de especificação de entrada do **espresso**. A partir daí, completem a especificação de entrada para que reflita exatamente a especificação da parte combinacional da aula passada e executem o **espresso** sobre este arquivo, gerando um arquivo de saída. Analisem este arquivo para verificar qual o resultado da simplificação conduzida pelo programa. Comparem a implementação de vocês com a do **espresso** em termos de total de portas lógicas e número de entradas de cada porta lógica.

Parte III – Aumentando o Nível de Abstração – o Editor de FSMs do Foundation

Execute a implementação do diagrama de transição de estados da aula passada sob a forma de um grafo ou diagrama de transição de estados. Use a ferramenta de edição de FSMs disponível no sistema de CAD Foundation. Para auxiliá-lo, existe um tutorial no Menu **Help → Foundation Help Contents → State Editor Tutorial**, que explica o uso do software passo a passo. Não se esqueça de gerar um símbolo de esquemático após editar o grafo de transição de estados.

Parte IV - A fazer:

1. Siga as instruções da **Parte II**, e gere o arquivo resultante da execução do espresso.
2. Implemente a máquina de vender refrigerantes como um grafo de transição de estados.
3. Simule a nova descrição, usando o mesmo script do laboratório anterior.