

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
**INSTITUTO DE INFORMÁTICA**  
**Cursos: Engenharia de Computação e Ciência da Computação**

TRABALHO 3  
Disciplina: INF1058 – Circuitos Digitais

Prof. Sergio Bampi  
Turma B

**TRABALHO PRÁTICO III**

Descrição e Implementação em Placa com FPGA de uma Máquina de Controle de Vendas

## 1) OBJETIVO

Neste trabalho o aluno deverá desenvolver uma máquina de estados para controlar uma máquina de venda de bebidas. A descrição dessa máquina deve ser feita em duas etapas:

- 1- Etapa Manual. O aluno deve implementar a máquina sozinho, com o método de síntese explicado em aula. O entregável desta etapa é o o grafo de estados completo e o diagrama lógico completo com Flip-Flops tipo D e portas lógicas.
- 2- Etapa com especificação em código VHDL e *prototipada* no Kit de Desenvolvimento DE0 utilizando o ambiente de software de simulação e de síntese de sistemas digitais, o Quartus II. O mapeamento dos periféricos, assim como o funcionamento da máquina estão descritos nas próximas seções.

Ao final deste trabalho o aluno deverá ser capaz de demonstrar o correto funcionamento da máquina de bebidas no Kit DE0 em aula e, também, de explicar as técnicas/metodologia utilizada para desenvolvê-la. Além da implementação em FPGA da máquina de bebidas, o aluno deverá fazer, também, o desenvolvimento e síntese manuais utilizando flip-flops do tipo D.

## 2) ESPECIFICAÇÃO

### Comportamento

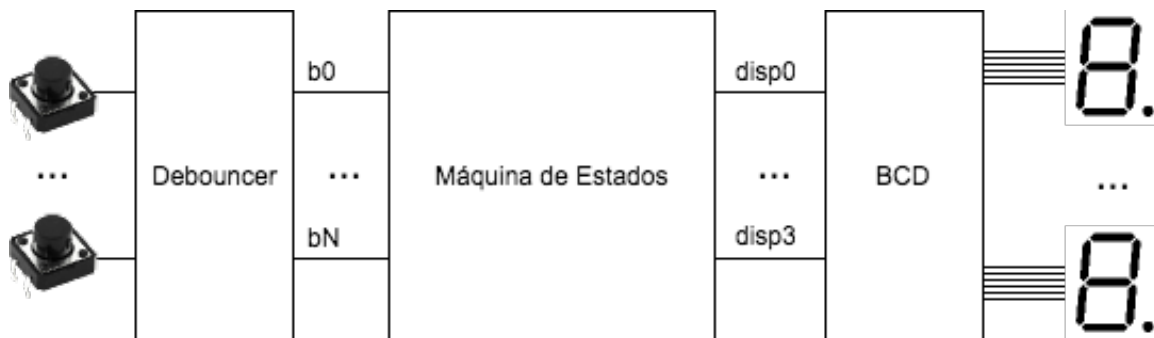
A máquina de bebidas aceita três moedas distintas (de R\$1,00 , R\$0,50 , R\$0,25) para comprar ou um suco por R\$0,75 ou uma refrigerante por R\$1,50. Para que o usuário tenha controle sobre o valor já inserido na máquina, a cada moeda inserida, o saldo atual deverá ser atualizado nos *displays* de 7 segmentos. Os primeiros dois *displays* representam o valor à esquerda da vírgula, enquanto os outros dois, à direita da vírgula.

Quando o usuário atingir o saldo desejado ele poderá efetuar a compra de um dos itens oferecidos. O projetista, no entanto, deve conferir se há saldo suficiente para que a compra seja efetuada com sucesso. Além de inserir moedas e comprar os

insumos oferecidos, o usuário também pode cancelar a operação e, assim, recuperar as moedas inseridas.

### Visão do Sistema

O sistema é composto de três grandes blocos: um *debouncer*, a FSM e o controlador BCD. O *debouncer* é responsável por filtrar os cliques nos botões e garantir que a máquina de estados funcione corretamente. Sua implementação será discutida em aula. A saída desse é interpretada pela FSM, bloco de controle de nosso sistema. A saída da FSM, por sua vez, controla o funcionamento do display de 7 segmentos. Como veremos em aula, para controlar estes displays é necessário um módulo que configure cada segmento individualmente. Logo, utilizaremos um bloco BCD para converter um número binário para os valores corretos de configuração de cada display. Este bloco também será discutido em aula.



**Figura 1 – Diagrama de blocos da arquitetura topo.**

### Entradas do Circuito

1. Mapeados em chaves (*switch*) separadas:
  - Moeda de R\$0,25
  - Moeda de R\$0,50
  - Moeda de R\$1,00
  - Cancela operação
  - Comprar suco
  - Comprar refrigerante
2. Mapeamento em botões separados:
  - Sinal de reset do sistema

### Saídas do Circuito

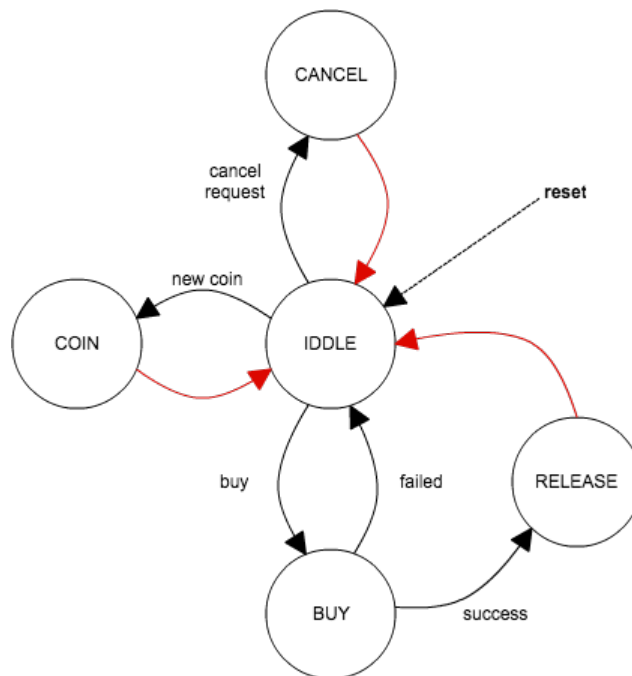
3. Mapeados nos displays de 7 segmentos
  - Saldo atual do usuário:
    - Os dois displays à esquerda representam os valores à esquerda do display.
    - Os dois displays à direita, por sua vez, representam os valores a direita da vírgula

## Máquina de estados

Como pode ser observado na Figura 2, a máquina de estados sequenciais é composta de cinco estados:

- **IDDLE** → Neste estado o sistema espera a interação do usuário, i.e. que injete-se alguma entrada;
- **COIN** → Quando o usuário inserir uma moeda, o sistema avança para este estado e, dependendo da moeda, atualiza o valor do saldo atual;
- **CANCEL** → Quando o usuário solicitar o cancelamento da compra o saldo deve ser zerado;
- **BUY** → Quando o usuário solicitar a compra de um item (suco ou refrigerante) o sistema entra neste estado para verificar se há saldo suficiente para efetuar-se a compra. Caso essa condição não seja satisfeita, retorna-se para o estado de *iddler*.
- **RELEASE** → O sistema atinge este estado quando o usuário solicitou algum item (suco ou refrigerante) e possui saldo disponível suficiente.

Inicialmente o sistema deve ser posto em um estado conhecido através do sinal de *reset*. Esse estado é o **IDDLE**. A partir desse, o sistema espera por uma entrada do usuário. Caso entre uma nova moeda, a máquina troca para o estado **COIN**, atualiza o valor do saldo atual e retorna para o estado **IDDLE**. Observe que essa condição de retorno ao estado **IDDLE** ocorre independentemente de alguma ação do usuário. Essa e outras flechas em vermelho são chamadas de transições incondicionais (sem condições de ativação) e são utilizadas para que o sistema volte no estado inicial (esperando entradas do usuário). Além do estado **COIN**, a partir do **IDDLE** é possível atingir o estado **CANCEL** e **BUY**. No primeiro, o saldo do usuário é zerado para simular o retorno do dinheiro inserido por ele, enquanto no segundo a máquina irá atender uma solicitação de compra. No estado **BUY**, verifica-se se há saldo suficiente para completar a compra. Caso essa condição falhe, a máquina retorna para o estado **IDDLE** sem alterar o saldo. Caso a condição seja verdadeira, a máquina avança para o estado de **RELEASE** e reduz o saldo do usuário pela quantia do produto selecionado e retorna para o estado inicial.



**Figura 2 – Máquina de Estados do sistema.**

### 3) METODOLOGIA

Devido a complexidade do sistema recomenda-se que o aluno siga o fluxo proposto abaixo. Esse apresenta uma metodologia procedural no qual todos os blocos poderão ser testados individualmente e, assim, garantir o correto funcionamento do trabalho.

1. Aprender a utilizar o display de 7 segmentos. Recomenda-se fazer algumas descrições de VHDL para configurar o displays com valores diferentes para entender o seu funcionamento.
2. Aprender a utilizar o *debouncer*. Esse módulo é de vital importância para o sistema, uma vez que ele garante que as entradas do sistema pulsem apenas uma vez (conforme será visto em aula).
3. Estudar a descrição da máquina de estados. Antes de iniciar a sua descrição, é de extrema importância que o aluno tenha entendido o funcionamento da FSM.
4. Implementar a FSM. Para que o seu trabalho fique mais organizado, recomenda-se utilizar um arquivo para cada módulo.
5. Testar o funcionamento da FSM por simulação. Durante a simulação é possível observar todos os sinais do sistema, desta forma você pode identificar erros facilmente e corrigi-los.
6. Testar o funcionamento do sistema na placa.

### 4) ENTREGA

A demonstração da ULA operando corretamente em chip FPGA modelo *Cyclone III*, na placa DE0 da Altera Inc., será feita em aula no dia **05 de julho**, durante a aula de laboratório.

Os resultados obtidos devem ser escritos na forma de relatório, cujo *deadline* de entrega via *moodle.inf.ufrgs.br* será no dia **05 de julho**.

Os itens abaixo devem estar presentes do relatório para que o aluno possa atingir a nota máxima:

- Síntese **manual** da máquina de estados (FSM) utilizando FF-D. Deve estar presente no relatório todo o desenvolvimento desta etapa, e.g. tabelas e circuito.
- Síntese e extração de resultados de utilização de recursos (número de LUTs, de registradores, de pinos de I/O etc).
- Criação do arquivo de teste Vector Waveform File (VWF) com as formas de onda e plotagem dos resultados. **OBS:** Faça gráficos legíveis para o revisor, caso contrário haverá descontos na avaliação do trabalho.
- Figuras contendo os blocos desenvolvidos.
- Plotagem do Netlist gerado pelo Quartus II.
- Programação em FPGA do circuito da ULA com a pinagem detalhada em relatório.