

# Máquina de venda de cafés

Universidade de Aveiro

Diogo Mendes, João Tiago Rainho



V 1.0

# Máquina de venda de cafés

D.E.T.I.

Universidade de Aveiro

P3-Grupo 1

Diogo Mendes, João Tiago Rainho

(88801) diogovmendes@ua.pt, (92984) tiago.rainho@ua.pt

3 de Junho de 2019

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Arquitectura</b>	<b>2</b>
<b>3</b>	<b>Implementação</b>	<b>4</b>
3.1	Máquina de estados finitos . . . . .	4
3.2	Arquitectura detalhada . . . . .	5
3.3	Ligação a periféricos . . . . .	8
<b>4</b>	<b>Validação</b>	<b>9</b>
<b>5</b>	<b>Conclusão</b>	<b>10</b>

# Capítulo 1

## Introdução

Este documento está dividido, em quatro capítulos. Depois desta introdução, no Capítulo 2 é apresentada a arquitetura geral do sistema e explicado o processo de criação da mesma. No Capítulo 3 é explicado o diagrama de estados da máquina de estados finita do sistema, sendo também feita uma análise detalhada a vários blocos criados pelos alunos e aos periféricos do kit ligados. No Capítulo 4 são apresentadas algumas simulações de blocos e do sistema geral. Por fim no Capítulo 5 é discutido se o trabalho final vai ou não de acordo com os objetivos iniciais.

A máquina de vendas de cafés tem um processo de funcionamento bastante simples. Primeiro, deve ser escolhido o produto desejado, cujo preço será apresentado nos displays de 7 segmentos, através do uso dos *switches* 3 a 0. De seguida, deve-se introduzir o dinheiro para efetuar o pagamento fazendo uso das *keys* 3 a 0, que correspondem a 50, 20, 10 e 5 cêntimos, respectivamente. O troco (caso exista) será então apresentado nos displays de 7 segmentos, a piscar a uma frequência de 10Hz. Por fim, o utilizador pode observar nos leds vermelhos o número de moedas de 5 cêntimos disponíveis para ser fornecido o troco. Caso não existam moedas suficientes a máquina irá apresentar o troco que falta fornecer e irá parar o seu funcionamento até ocorrer a introdução de mais moedas de troco (através do *switch* 17). Note-se que o switch de escolha do produto deve ficar para cima até ao final visto que, o facto de o utilizador o baixar será interpretado como a recolha do produto e do troco. O *switch* 4 funciona como botão de reset da máquina.

## Capítulo 2

# Arquitectura

Ao longo da criação do sistema foram introduzidas cada vez mais funcionalidades, isto devido ao processo faseado pelo qual o projeto foi feito.

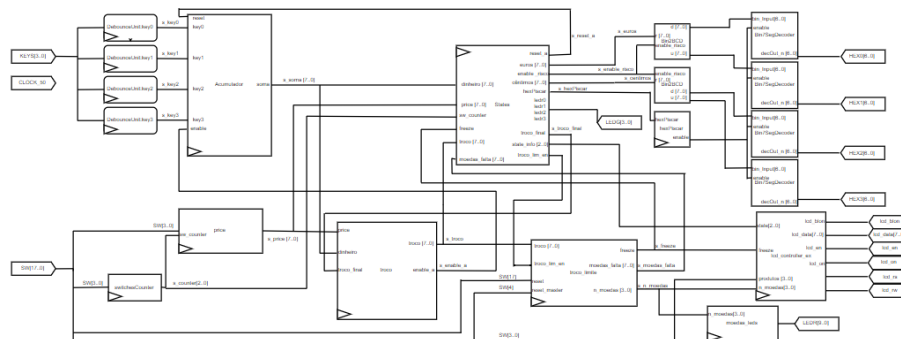
Na fase 1, o sistema apenas é composto pelos *Debouncers* pelos quais passa o valor das *keys*. Este valor é então introduzido num acumulador, com um *reset*, o valor do acumulador (dinheiro) é então introduzido no *states* (máquina de estados). Para além disto, inclui também um bloco *Counter\_SW* que é meramente um contador do número de *switches* que se encontram a 1. O valor dos *switches* passa também para o bloco *price* cuja saída é um vetor com o preço associado ao *switch* ativo e que entra como *input* no *states*. Nas saídas da máquina de estados temos, apenas, os valores (euros e cêntimos) que vão para os blocos *Bin2BCD* e *Bin7SegDecoder* e que destes vão para os displays de 7 segmentos e temos ainda uma saída que serve como *reset* para o acumulador.

Na fase 2, foram ,para além de eventuais sinais e mudanças mínimas no código de blocos já existentes na fase 1, introduzidos dois blocos importantes. O bloco *troco* cujas entradas são as saídas do acumulador e do *price*, ou seja, o dinheiro introduzido pelo utilizador e o preço do produto. Este bloco calcula o troco a ser entregue (dinheiro-preço), para não permitir que o utilizador continue a introduzir mais dinheiro depois do produto estar pago. Foi adicionado um *enable* ao bloco acumulador. Foi criado também o bloco *hexPiscar* que é responsável por colocar os displays de 7 segmentos a piscar a 10 Hz de acordo com o *enable* que provém do *states*.

Finalmente, na fase 3, e novamente ignorando alterações mínimas ao código de blocos previamente referidos, foram criados os blocos: troco-limite e moedas-led. O bloco troco-limite é responsável por calcular o limite de troco, ou seja, vai contando quanto moedas de 5 cêntimos ainda estão disponíveis. Esta informação é passada para o *states* e para o moedas\_led. No moedas\_led a quantidade de moedas restantes é preparada para ser representada em leds. Note-se que nos leds verdes 3 a 0 pode-se observar em que estado a máquina de estados se encontra. Isto permite ao utilizador saber que passo deve fazer a seguir (escolha, pagamento ou levantamento do produto).

Posteriormente, foram adicionados dois blocos, fornecidos pelos docentes da

Abaixo apresenta-se um esquema da arquitectura geral do sistema. Note-se que as ligações do CLOCK\_50 aos vários blocos foram omitidas, sendo representadas como um triângulo nos blocos adequados.



3

## Capítulo 3

# Implementação

### 3.1 Máquina de estados finitos

A abordagem adotada para a elaboração deste sistema implicou a criação de uma máquina de estados finitos com 5 estados, que são:

- I (Inicial) - Estado inicial em que aguarda a introdução de inputs; são efetuados todos os resets necessários neste estado, visto ser o primeiro.
- SB (Stand-By) - Estado em que foi selecionado um SW (o preço do mesmo está a ser demonstrado nos displays) e aguarda o pressionar de uma KEY para passar ao próximo estado (daí stand-by);
- S (Start) - Estado em que o sistema aceita a introdução de moedas através das KEYS. Quando dinheiro maior do que preço é apresentado o valor do troco, a piscar nos displays, e são retiradas as moedas usadas para o troco dos LEDs e prossegue para o estado F.
- F (Final) - Estado "final", é o último estado no caso geral. Neste estado o sistema aguarda que o utilizador coloque o *switch* para baixo de forma a sinalizar a recolha do produto e do troco, regressando então ao estado I. Caso não hajam moedas para fornecer o troco, irá passar para o estado F\_R.
- F\_R (Final Refill) Estado final alternativo, ocorre apenas caso o número de moedas para troco seja menor do que o requerido (necessita de "refill" das moedas para prosseguir daí o nome). Neste estado será apresentado nos displays de 7 segmentos o troco em falta e a máquina estará num estado de "freeze". Após introduzirem-se mais moedas para troco com o *switch* 17 a máquina aguardará que o *switch* do produto seja colocado para baixo antes de regressar ao estado I.

Abaixo encontra-se um esquema da máquina de estados finitos. Note-se que a ordem das entradas é: count\_sw; price; dinheiro; troco; freeze; moedas\_falta. E a das saídas: euros; centimos; enable\_Risco; hexPiscar; reset\_a; troco\_final; troco\_lim\_en. As saídas ligadas aos leds verdes e ao state\_info não são representadas para simplificar o esquema visto que apenas representam o estado em que máquina se encontra.

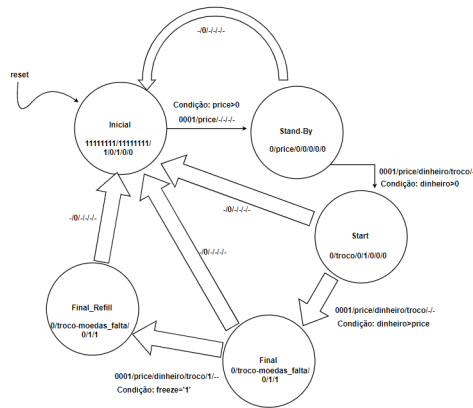


Figura 3.1: Esquema da máquina de estados finitos.

## 3.2 Arquitectura detalhada

Ao longo deste sub-capítulo serão apresentadas as composições de vários blocos elaborados pelos autores com objetivos muito específicos.

Bloco Price.

⇒ Objetivo:

- Calcular o valor do preço, de acordo com o produto selecionado, se nenhum ou mais do que um produto então o preço é 0.

⇒ Composto por:

- 2 comparadores "LessThan";
- 18 somadores ("Adders");
- 9 multiplexers;
- 9 FlipFlops tipo D com enable;



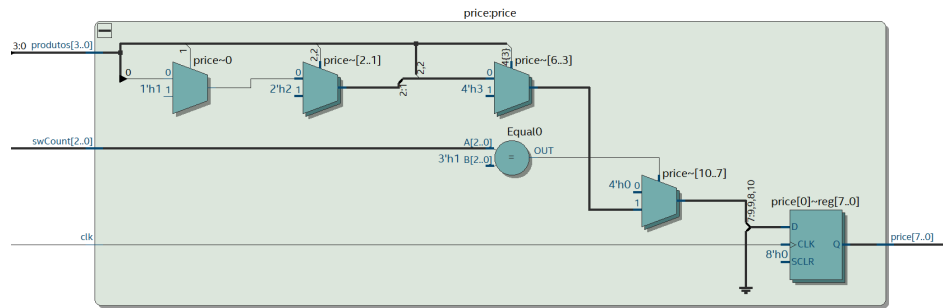


Figura 3.2: Esquema do Price.

Bloco Troco.

⇒ Objetivo:

- Calcular o valor do troco.

⇒ Composto por:

- 2 comparadores "LessThan";
- 18 somadores ("Adders");
- 9 multiplexers;
- 9 FlipFlops tipo D com enable;

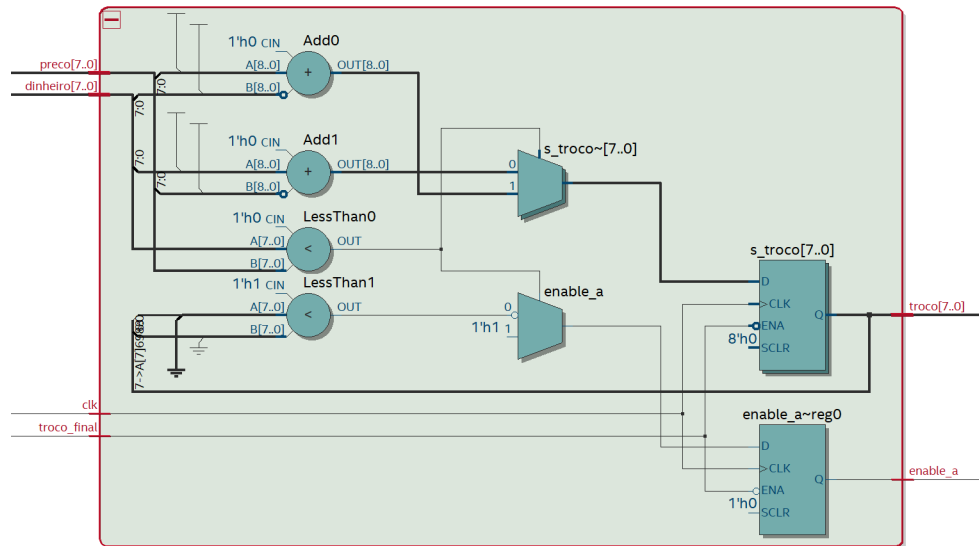


Figura 3.3: Esquema do Troco.

### Bloco Troco\_limite

⇒ Objetivo:

- Calcular o valor de troco ainda disponivel, quer em moedas quer em valor numérico.

⇒ Composto por:

- 1 And;
- 1 Or;
- 8 comparadores "Equal";
- 2 comparadores "LessThan";
- 32 somadores ("Adders");
- 8 divisores
- 102 multiplexers;
- 11 FlipFlops tipo D;
- 17 FlipFlops tipo D com enable;

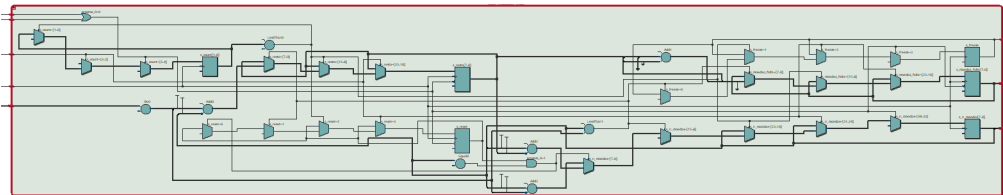


Figura 3.4: Esquema do Troco\_limite.

### Bloco hexPiscar

⇒ Objetivo:

- Enable no modo piscar. Efetua o piscar a uma frequência de 10Hz.

⇒ Composto por:

- 1 comparador "LessThan";
- 32 somadores ("Adders");
- 68 multiplexers;
- 1 FlipFlop tipo D;
- 2 FlipFlops tipo D com enable;

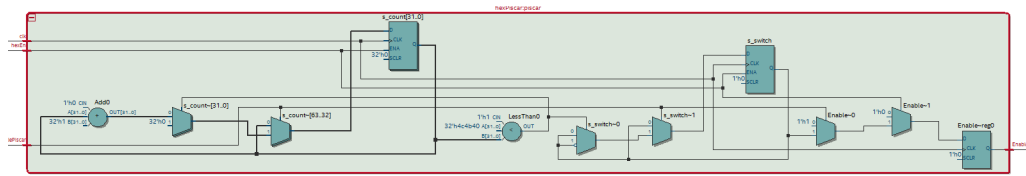


Figura 3.5: Esquema do hexPiscar.

### 3.3 Ligação a periféricos

No final deste projeto, a máquina obtida encontra-se ligada aos seguintes periféricos:

- 6 switches (SW[4 downto 0] e o SW(17));
- 4 keys (KEY[3 downto 0]);
- 3 leds verdes (LEDG[3 downto 0]);
- 10 leds vermelhos (LEDR[9 downto 0]);
- 4 displays de 7 segmentos (HEX0 a HEX3);
- 16 x 2 LCD display;
- Clock de 50 MHz;

## Capítulo 4

# Validação

Todos os blocos utilizados assim como a arquitectura geral do sistema foram testados diretamente no kit. As simulações de blocos que ocorreram aquando da elaboração do trabalho foram efetuadas em blocos com erros e não vão ser portanto apresentadas.

## Capítulo 5

# Conclusão

Todos os objetivos declarados no enunciado fornecido pelos docentes da disciplina foram alcançados. Excetuando um bug de código relativo à apresentação do valor das moedas em falta, todo o restante código encontra-se a trabalhar corretamente e dentro do previsto.

Todo o trabalho efetuado ao longo do projeto foi dividido equitativamente pelos membros do grupo, quer a elaboração deste relatório como a elaboração do código e dos esquemas utilizados na apresentação e neste documento.

Devido ao resultado final os autores deste trabalho julgam que merecem uma nota não inferior a 17 valores. Visto que, apesar de existir um pequeno bug no código relativo à apresentação do valor das moedas em falta em caso de não existir troco disponível, o restante código, relatório e apresentação preparados são merecedores desta nota. Para além disso, os alunos completaram os requerimentos extra pedidos pelos docentes e irão submeter o vídeo requerido até à data indicada.