**departamento de eletrónica, telecomunicações e informática**

|  |  |
| --- | --- |
| Curso | 8204 - Mestrado Integrado em Engenharia Eletrónica e Telecomunicações |
| Disciplina | 40331- Laboratórios em Eletrónica |
| Ano letivo | 2018/19 |

Relatório

Máquina de vendas

Autores:

|  |  |  |  |
| --- | --- | --- | --- |
| [91372] | [Eduardo Cunha] | | |
| [88803] | [Joana Teixeira] | | |
| Turma | [P10] |  |  |

|  |  |
| --- | --- |
| Data | [12/05/2019] |
| Docente | [Bernardo Cunha] |

Contribuição para o trabalho: Joana Teixeira (50%)

Eduardo Cunha (50%)

**Fase 1:**

Nesta fase pretendemos implementar uma máquina de vendas onde seja feita a escolha do produto, a inserção das moedas, e por fim a ativação do led para sinalizar a saída do produto.

Então, decidimos criar uma máquina de estado (MaqVendas), para ser mais fácil a manipulação destas fases de processo. Como podemos, ver na fig1. temos o diagrama de estados. Para inicializar a máquina utilizamos o SW[1] com a função start e para voltar ao estado inicial é necessário colocar o SW[0] a ‘1’.

Para além disso, criamos um bloco chamado “SelProd”, que consiste na escolha do produto através dos SW[2..3], a escolha deste produto será representada nos leds 0 a 2 em binário, e em simultâneo com os displays de sete segmentos HEX6, este bloco também envia para o “ContMoedas” o valor do produto escolhido. Como a máquina de estados é síncrona e recebe o sinal de clock de 1Hz, não tínhamos tempo hábil para selecionar o produto, decidimos implementar um multiplexador e nessa fase a máquina de estados deixa de receber o sinal de clock anterior, passando a receber o sinal do KEY[0].

Temos também um outro bloco (“ContMoedas”), que recebe informação da moeda introduzida a partir dos SW[15..16] e vai incrementando a moeda introduzida ao valor anterior, quando o valor total de moedas introduzidas for maior ou igual que o valor do produto acenderá um led, pois o produto saiu.

Uma imagem com mapa, texto

Descrição gerada automaticamente

Fig.1 – Diagrama de estados

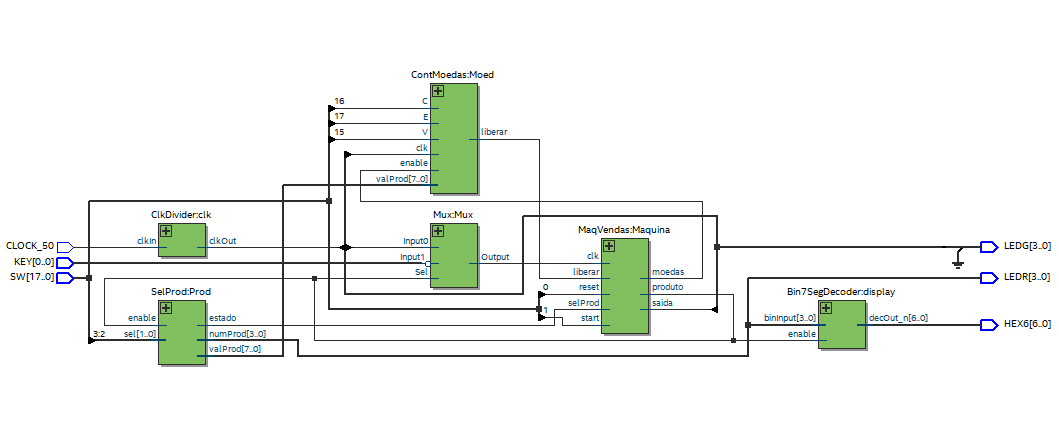


Fig.2 – Esquema de blocos da Fase 1

**Fase 2:**

Na fase 2, para além das características da fase 1, a máquina de vendas terá de dar troco e apresentar desse nos displays de sete segmentos.

Para tal, mantivemos todos os blocos da fase 1 e acrescentamos um novo chamado “Troco”, que tem como objetivo, receber o valor total de moedas inseridas, bem como o valor do produto, e calcular a diferença entre este que será enviada para os displays HEX4 e HEX5.

Como o estado FProduto é o primeiro a ser executado, quando passar para a fase seguinte este será desativado e o valor do produto será apagado, então decidimos guardar esse valor num registo “Holder”, para que este valor chegue ao bloco “Troco”.

Também acrescentamos um bloco “Troco2bcd”, que converte o valor do troco em bcd.

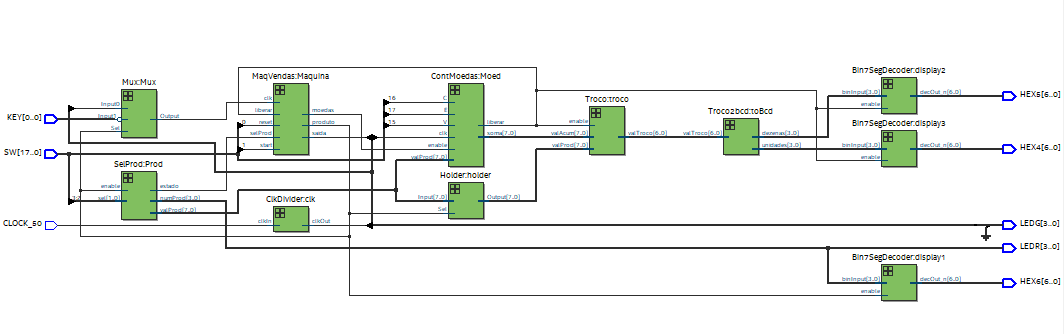


Fig.3 – Esquema de blocos da Fase 2

**Fase 3:**

Na fase 3 é solicitado que seja incluído a quantidade de cada um dos quatro produtos disponíveis na máquina de vendas. Sendo essas quantidades sempre atualizadas quando um produto é liberado. Por tal motivo, mantivemos todos os módulos das fases anteriores e criamos um bloco denominado “QuantProd” que tem a função de disponibilizar em quatro displays de sete segmentos HEX0 a HEX3, o número de unidades de cada produto. Além disso, esse também será o módulo que irá decrescer em uma unidade o produto escolhido assim que o mesmo for liberado.

Como forma de otimização, instanciamos no projeto uma porta AND denominada de “ModAnd” em que as duas entradas estão ligadas na saída “Estado” da entidade “SelProd” e na saída de mesmo nome da entidade “QuantProd”. A saída está ligada a “MaqVenda”, sendo assim só conseguimos avançar à próxima fase se as entradas do “ModAnd” for igual a 1. Com isso não conseguimos selecionar e seguir o processo em objetos que já estejam esgotados.

Foi necessário também a implementação de mais um registo para segurar o valor do número do produto que foi selecionado, visto que o módulo “SelProd” é desligado assim que a máquina de estados muda de estado.

Nesta fase é solicitado que, quando um produto atingir o valor 0 este deve piscar a uma frequência de 1Hz, no entanto, como podemos verificar não é o que está acontecendo no projeto. Contudo, tentamos arranjar uma solução para o problema, o qual não foi possível e decidimos optar por deixar a piscar a uma frequência de 0.5Hz.

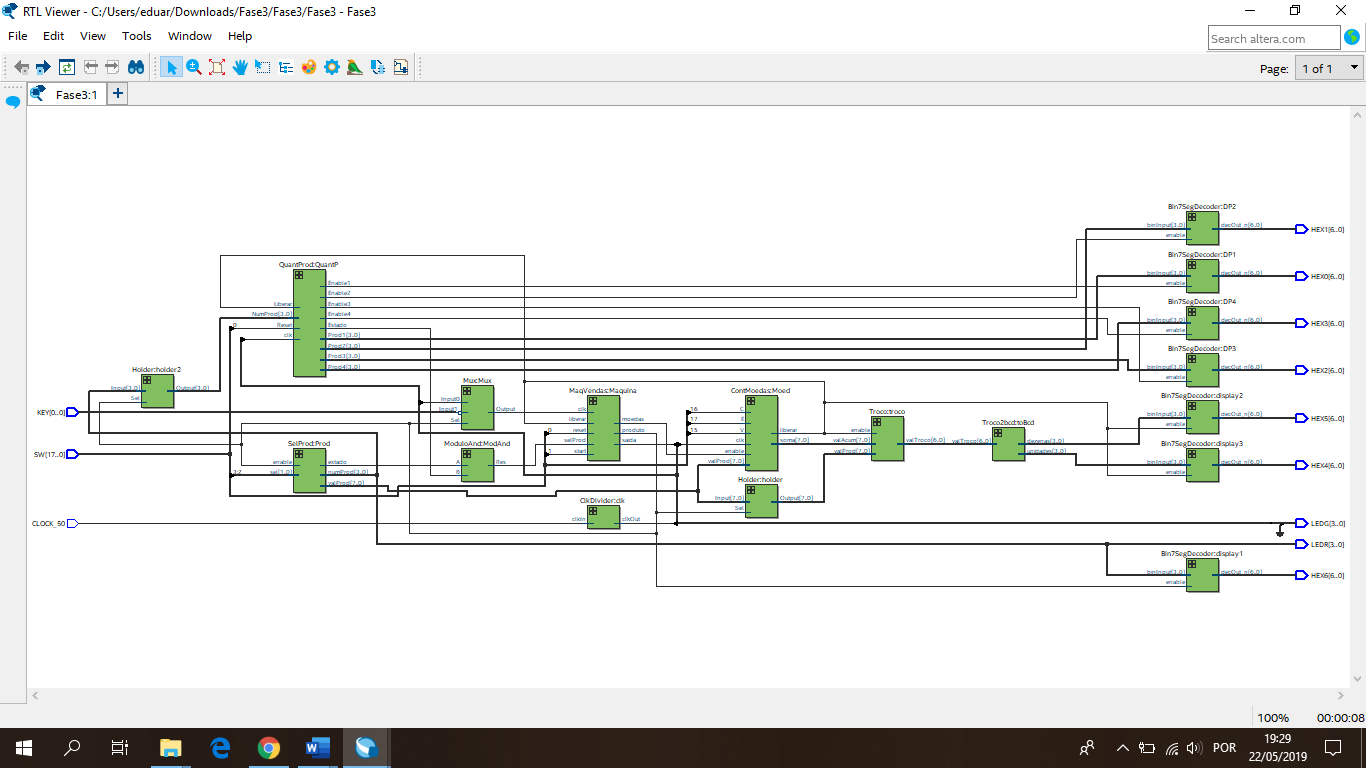


Fig.4 – Esquemas de blocos Fase 3