

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA - CT  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - DEE  
BACHARELADO ENGENHARIA MECATRÔNICA

RELATÓRIO SISTEMAS DIGITAIS – MÁQUINA DE VENDAS - PROJETO 01

NATAL  
2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA - CT  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA - DEE  
BACHARELADO ENGENHARIA MECATRÔNICA  
SISTEMAS DIGITAIS

ATYSON JAIME DE SOUSA MARTINS

RELATÓRIO SISTEMAS DIGITAIS – MÁQUINA DE VENDAS - PROJETO 01

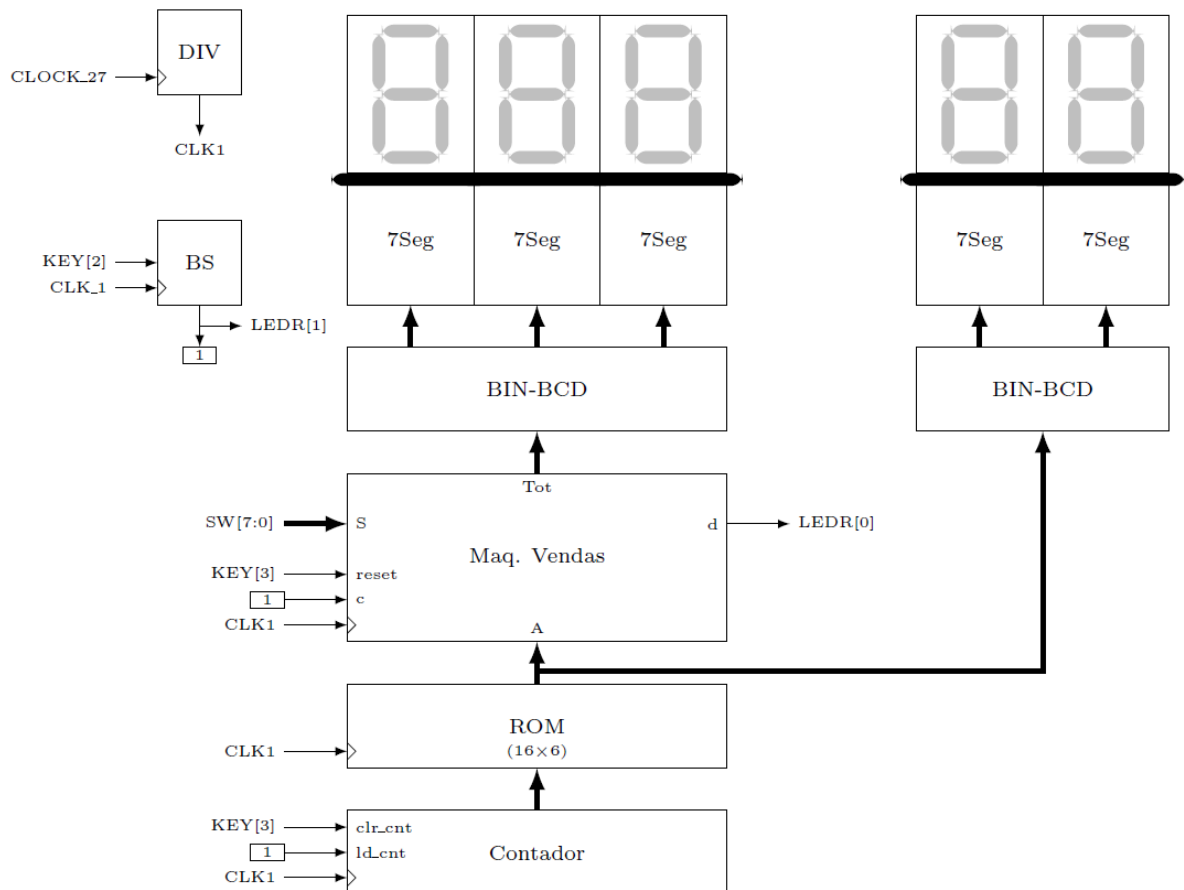
NATAL  
2020

## **Sumário**

<b>1. Introdução</b>	<b>03</b>
<b>2. Desenvolvimento</b>	<b>04</b>
<b>2.1 Divisor de Clock</b>	<b>04</b>
<b>2.2 Botão Síncrono</b>	<b>04</b>
<b>2.3 Maquina de Vendas</b>	<b>06</b>
<b>2.4 BINBCD – Display7Segmentos</b>	<b>08</b>
<b>2.5 ROM</b>	<b>08</b>
<b>2.6 Contador</b>	<b>09</b>
<b>3. Resultados Obtidos</b>	<b>10</b>
<b>4. Conclusão</b>	<b>14</b>
<b>5. Referências Bibliográficas</b>	<b>15</b>
<b>6. Anexos</b>	<b>16</b>

## 1. Introdução

Neste relatório estará presente todo o procedimento para escolha e construção da máquina RTL, Máquina de Vendas em VHDL para ser usado em uma FPGA em que seguiremos a arquitetura desejada pelo professor, apresentada na figura abaixo (imagem 1).



**Imagem 1 - Arquitetura desejada para ser implementada**

Todavia, antes de apresentar o procedimento, iremos falar sobre o que é uma Máquina RTL (Register Transfer Level); essas máquinas são métodos usados para criar processadores que são a junção de um bloco de controle (onde fica toda a parte que controla o processador) com um bloco operacional (onde fica toda a parte que mexe com dados do processador). Para se projetar uma máquina dessa é preciso seguir alguns passos: Obter a máquina de estados de nível alto; criar o bloco operacional; obter a máquina de estados finito do bloco de controle (FSM).

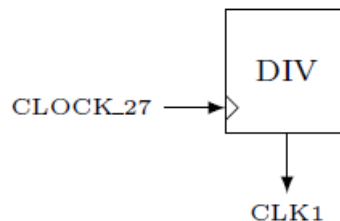
No seguinte estudo, foi necessária a utilização de outros componentes juntos a máquina RTL para a perfeito funcionamento desejado e adequado da estrutura referida na imagem 1.

## 2. Desenvolvimento

Para a realização do projeto proposto decidi fazer de forma estruturada, ou seja, destrinchar o projeto em partes e ao final juntar todas elas em um único arquivo. Desse modo, o código fica mais enxuto e mais fácil de ser testado. Assim, seguindo como roteiro a imagem 1, dividi os blocos da seguinte maneira: divisor de clock, botão síncrono, máquina de vendas (máquina de estado), BIN-BCD com display 7 segmentos, ROM e contador.

### 2.1. Divisor de Clock

Considerando que a FPGA utilizada para os testes possui em sua circuitaria interna um clock de 27MHz, foi necessário a criação de um bloco no qual reduza esse clock de operação da placa para um clock de operação de 10Hz.

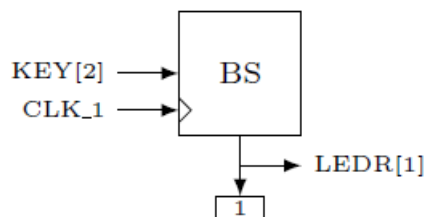


**Imagem 2 – Bloco Divisor de Clock**

Para a confecção desse, utilizou-se o código disponibilizado pelo professor, o qual a posteriori foi implementado em VHDL e simulado na plataforma.

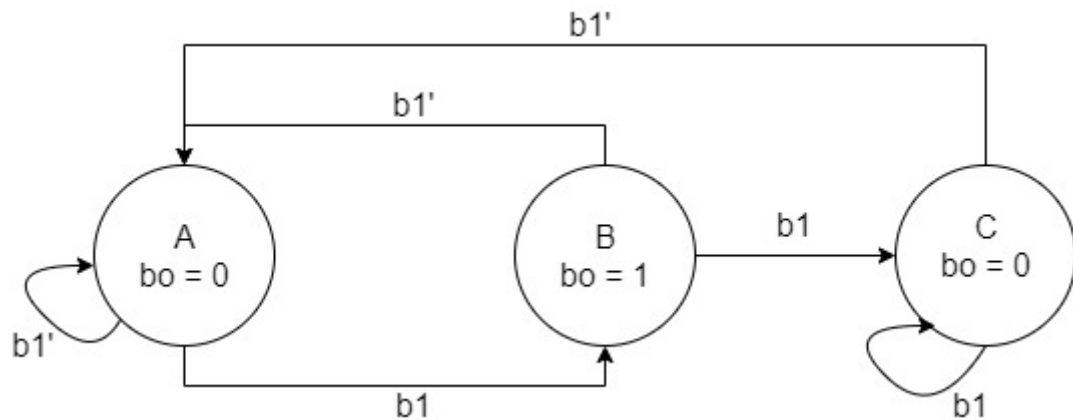
### 2.2 – Botão Síncrono

Umas das exigências apresentadas era que o botão de acionamento utilizado no projeto, como a colocação de uma moeda na máquina, funcionasse síncrono com o clock do projeto. Dessa forma, quando o botão é apertado, o resultado ficará em nível alto por exatamente um ciclo de clock. Isso servirá para que a máquina evite o erro de considerar um aperto contínuo como múltiplos abertos no botão.



**Imagem 3 – Bloco Botão Síncrono**

Para a sua construção, busquei usar a máquina de estado presente no livro Sistemas Digitais – Projeto, Otimização e HDL's pelo autor por Frank Vahid.

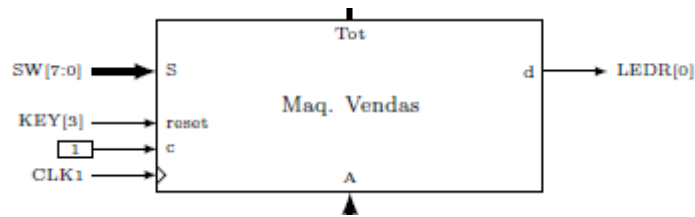


**Imagem 4 – Máquina de Estados Botão Síncrono**

Seu funcionamento se baseia da seguinte forma, quando a máquina está no estado A, ela espera o acionamento do botão b1. Enquanto esse botão não for acionado, ela continuará no estado A. Após o botão b1 ser pressionado, a máquina irá se dirigir ao estado B, cuja função é mandar a saída b0 para nível lógico alto. Caso b1 ainda esteja acionado, a máquina vai para o estado C, caso não esteja, volta para o Estado A. Em ambos os casos, a máquina deixará a saída b0 em nível lógico baixo. No estado C a máquina continuará nesse estado caso o botão fica acionado continuamente, se não, voltará para o estado A. Desse modo, a saída b0 só estará em nível lógico alto por um ciclo de clock apenas.

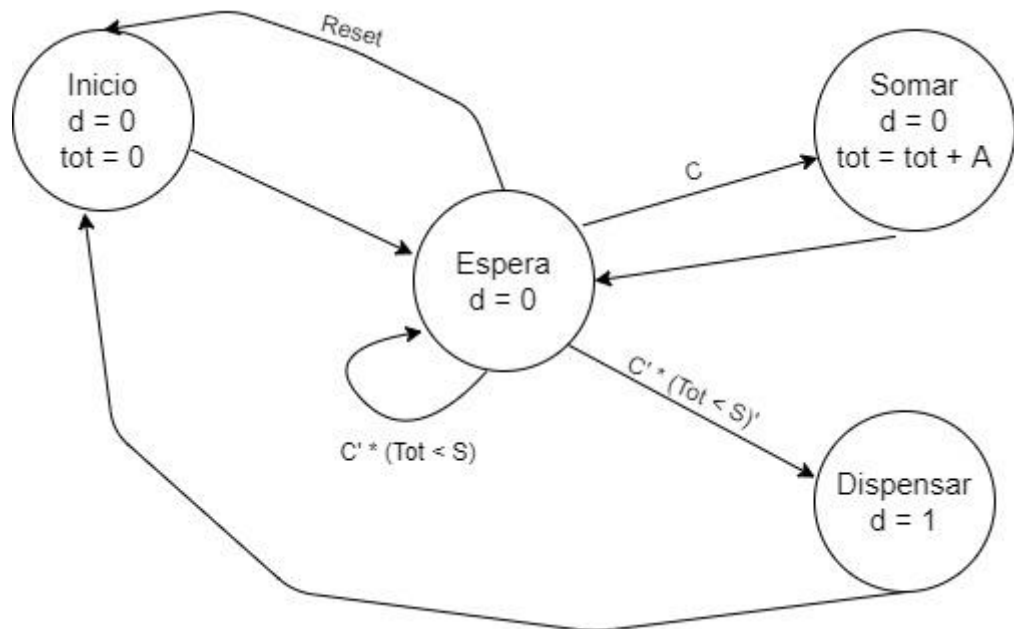
Por sua vez, observando essa máquina de estado, percebemos que ela é do tipo Moore, pois sua saída depende apenas do estado que a máquina se encontra. Sendo assim, realizamos a modificação do código base para máquina de estado do tipo Moore disponibilizado pelo professor para ficar de acordo com a lógica requerida, e assim implementar em VHDL e simular na plataforma.

## 2.3 – Máquina de Vendas



**Imagem 5 – Bloco Máquina de Vendas**

Utilizando também como base o projeto RTL para a máquina de vendas presente no livro Sistemas Digitais – Projeto, Otimização e HDL's pelo autor por Frank Vahid, com uma adicional de um botão de reset na máquina de estados. Em vista disso, ficamos com a seguinte formatação para a máquina de estado de alto nível.



**Imagem 6 – Máquina de estado de alto nível**

Funcionalmente da maquina se dar da seguinte maneira:

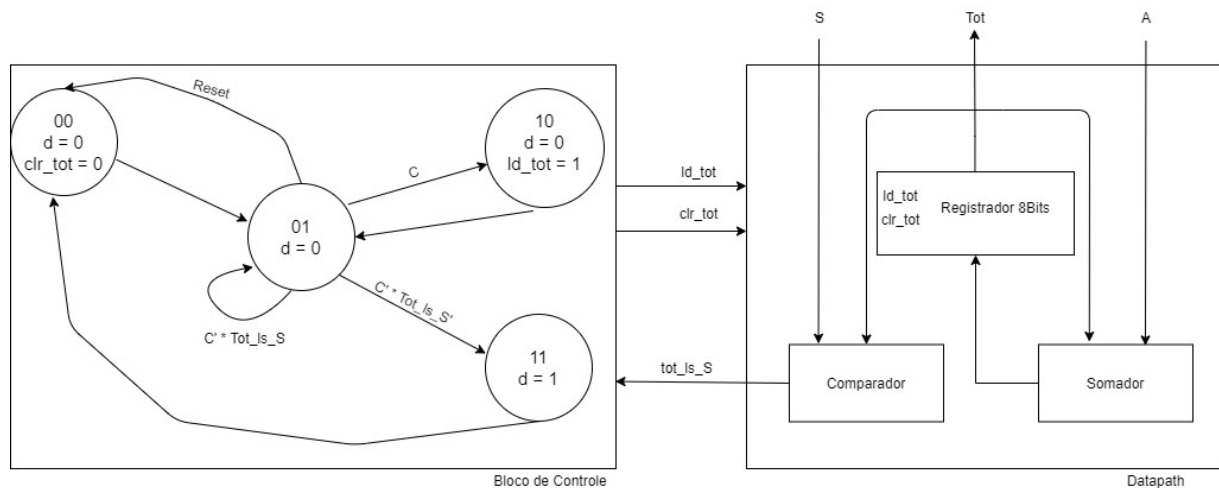
Início: estado no qual a máquina inicia;

Espera: após o pulso de clock a maquina vai para esse estado, no qual, fica esperando o depósito de alguma moeda, enquanto o total acumulado for menor que o valor final do produto, a máquina continuará nesse estado.

Somar: caso o usuário insira alguma moeda, ele irá para esse estado, no qual, basicamente ocorrerá a soma, do valor da moeda inserida com o total acumulado na máquina. Voltando para o estado de Espera após o pulso de clock.

Dispensar: quando o total se igualar ou for maior que o valor final do produto, ele é dispensado pela máquina para o usuário. Voltando para o estado de início após o pulso de clock.

Com esse esquemático pronto, dividimos o projeto RTL em datapath (local onde fica a parte que manipula dados) e bloco de controle (local onde fica a parte que controla a máquina). Para o datapath, foi necessário criar um Somador no qual somara o total já acumulado com a entrada A (valor referente a moeda depositada), como também, a necessidade de um registrador para guarda o valor do total acumulador, além disso, criou-se um comparador para comparar o total acumulado com a entrada S (valor final do produto desejado). Como as entradas são de 8bits, o registrador, somador e acumulador criados são de 8Bits também. Uma exigência para a arquitetura final desejada, era que o total acumulado fosse mostrado em alguns displays de 7 segmentos, por isso teremos uma saída do datapath equivalente ao valor guardado no registrado. Para o bloco de controle, os sinais antes literários foram trocados por sinal de 1bit. Portanto, ficando assim, com a seguinte estrutura para o bloco de controle é o datapath.



**Imagem 7– Bloco de controle e datapath da máquina de vendas**

Após implementação em VHDL, o mesmo foi simulado na plataforma.



## 2.4 – BIN-BCD com display7Segmentos

Se olharmos para a imagem 1 novamente vemos que a exigência da arquitetura é que, a saída tanto da ROM, como do registrador localizada dentro da maquina de vendas, tenha seu resultado mostrado em um display 7 segmentos. Para isso ocorrer, precisou-se primeiramente fazer uma conversão de binário para BCD.

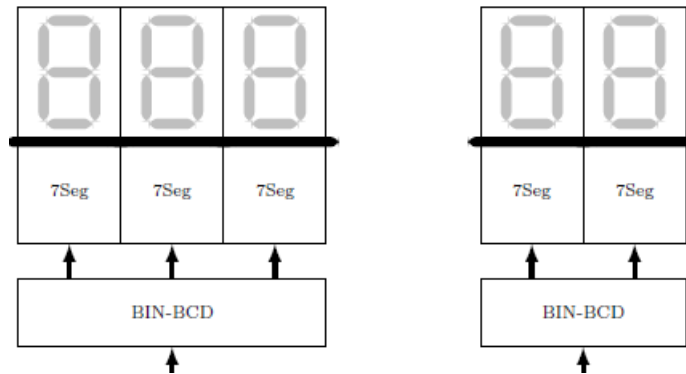


Imagem 8 – Bloco BINBCD – Display7Segmentos

Os códigos utilizados para fazer todas essas partes, tanto para a conversão de binário para BCD, quanto BCD para display 7 segmentos, foram retirados das atividades antes realizadas na matéria de Circuitos Digitais, ah qual, é requisito obrigatório para se pagar Sistemas Digitais.

## 2.5 – ROM

O bloco de memória vista na arquitetura da imagem 1, funcionará basicamente como um guardado dos valores das moedas possíveis para a maquina de vendas, por isso que sua saída vai tanto para entrada da maquina de vendas (A) vista na imagem 5, quando para o display de 7 segmentos (mostrando o valor da moeda) visto na imagem 8. Como no projeto não precisamos gravar nada em alguma memória, só ler o que está salvo nela, a ROM é a melhor escolha, já que, pela sua definição a ROM (*Read-Only Memory*) é uma memória apenas de leitura.

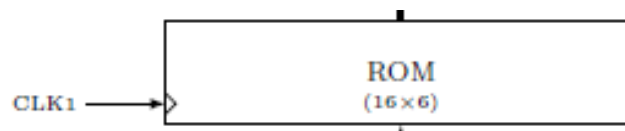


Imagem 9 – Bloco da ROM

Para a implementação da mesma em VHDL na plataforma, utilizou-se do passo a passo em pdf disponibilizado pelo professor por meio da turma virtual.

## 2.6 – Contador

No projeto, o contador foi empregado para acessar a posição da ROM e assim, pegar o valor que será usado como moeda na máquina de vendas.

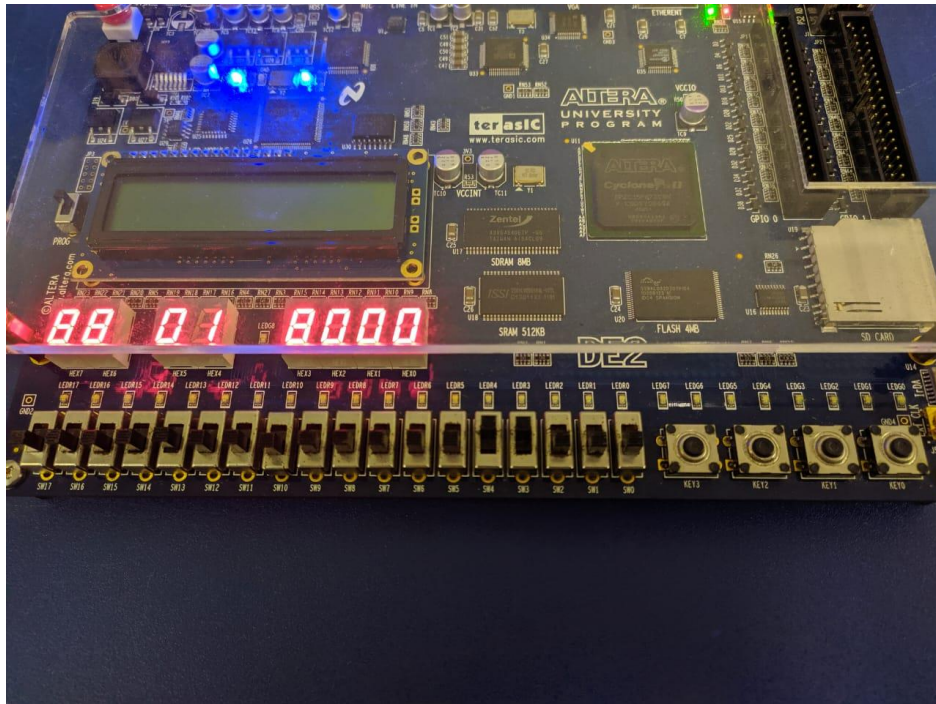


**Imagem 10 – Bloco Contador**

Dado que o tamanho da ROM é possível ter 64 posições diferentes, foi necessário criar um contador de 6Bits, pois quando fazemos  $2^6 = 64$ . Internamente, seu funcionamento se dar por flip flop JK juntos, com pequena diferença empregada devido a imposição quista na arquitetura final, no qual, foi a colocação de um load e um clear no contador. Desse modo, o contador só funcionará quando o botão for apertado, ou seja, quando o usuário inserir uma moeda. Quando isso ocorrer, o valor que está no contador ditará a posição da ROM no qual terá a quantia X para aquela moeda.

### 3. Resultados Obtidos

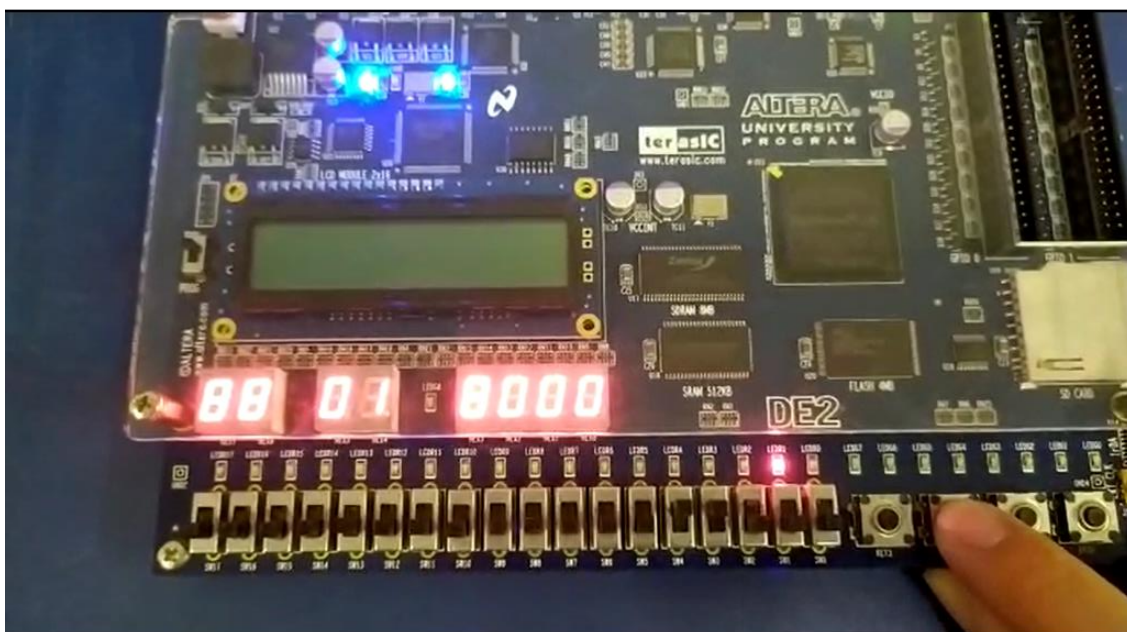
Após testes exaustivos em simulador o código foi posteriormente testado em FPGA, utilizando interfaces de saída de dados destinados aos displays da placa, aliado a um código divisor de clock utilizado para reduzir o clock de operação da placa, 27 MHz, para um clock de operação de 10Hz, previamente citado no projeto. Desse modo, começamos com a seguinte interface no FPGA:



**Imagem 11 – momento inicial do teste em FPGA**

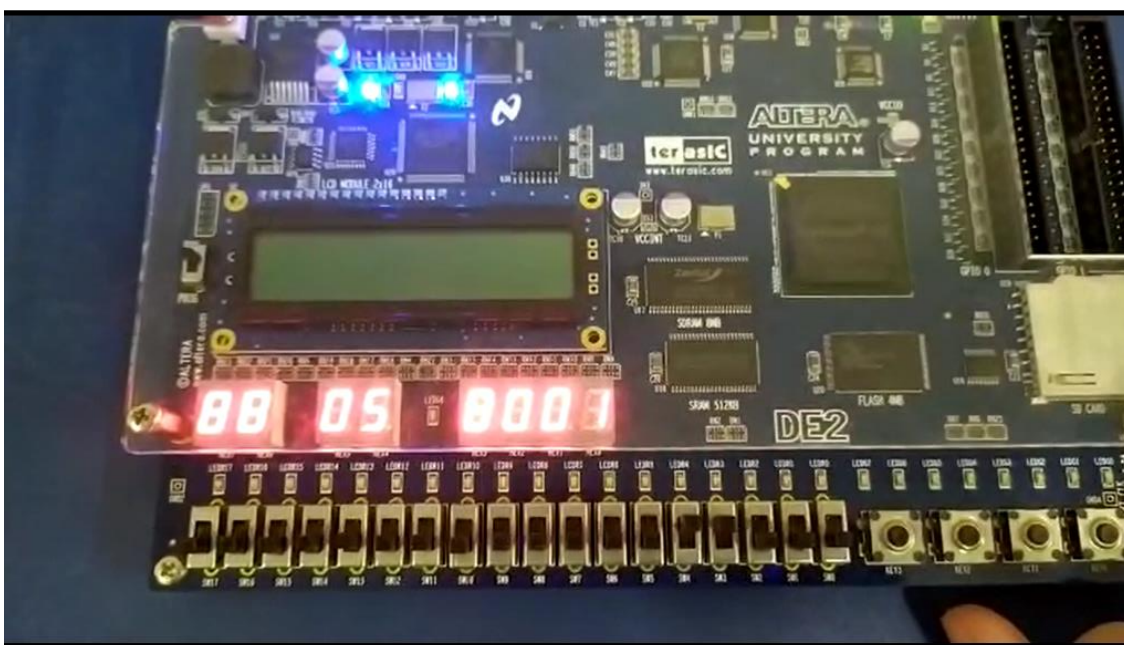
Como podemos ver na imagem 11, os 2 displays que se encontram no meio (o qual possuem valor 01) mostram o valor de entrada da moeda, ou seja, valor guardado na ROM. Já os 3 displays da ponta (o qual possuem valor 000 inicialmente) mostra o valor do total guardado na máquina. Para fins didáticos e de teste, colocamos a entrada SW igual a '00011000' o que gera um produto com um valor de 24.

Quando apertamos o botão KEY2 aciona-se o bloco do botão síncrono fazendo o load do projeto funcionar apenas por um ciclo de clock, quando isso ocorre o led LEDR1 da FPGA acende.



**Imagem 12 – Funcionalmente do botão síncrono**

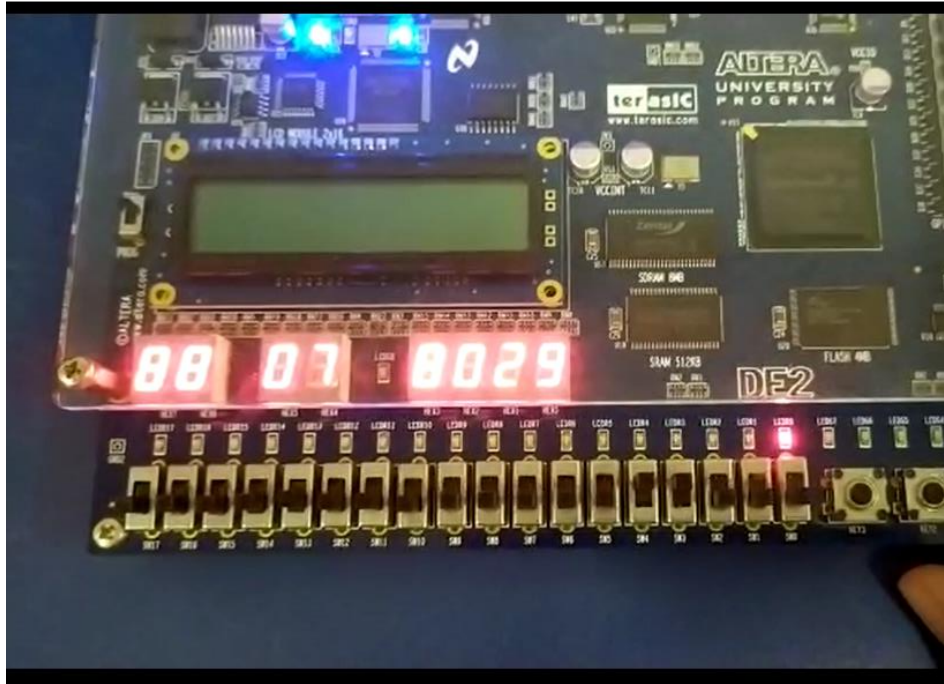
Após o led se apagar, os 3 displays da ponta mostraram o resultado da entrada do valor da moeda presente nos 2 displays do meio somado com o total presente na máquina guardado anteriormente nos 3 displays da ponta. Outrossim, os 2 displays do meio agora apresentaram o próximo valor de moeda, pois a ROM está mostrando outra posição internamente devido ao contador ter sofrido load com o acionamento do botão.



**Imagem 13 – Troca dos valores dos displays 7 segmentos**

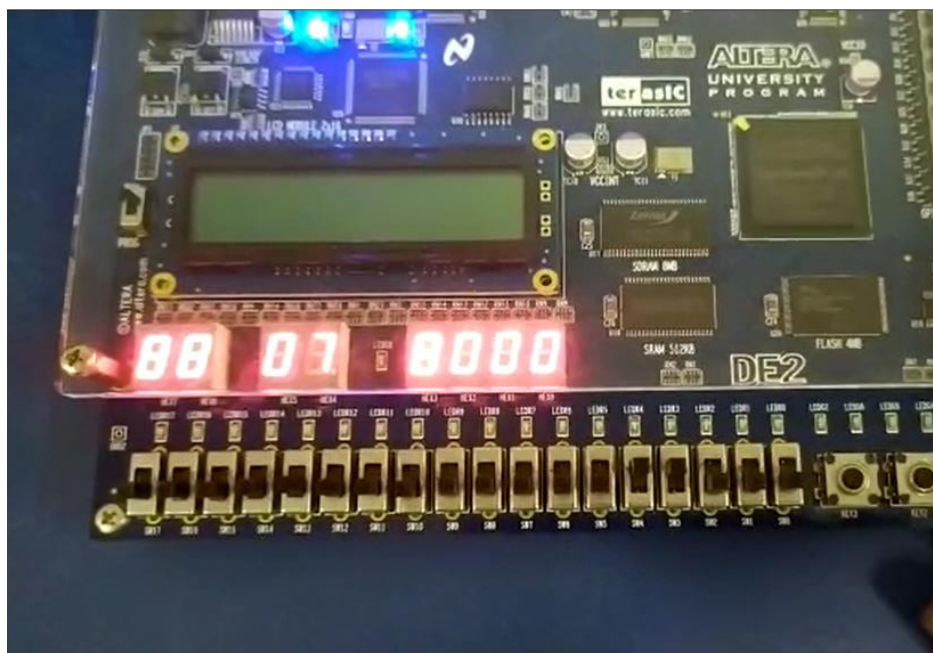


Continuando esse procedimento do acionamento do botão, chegaremos ao momento onde o total presente na maquina é maior ou igual ao valor anteriormente colocado para o preço do produto. Assim, quando isso ocorrer, o led LEDR0 da FPGA é acesso, informando ao usuário que o produto foi dispensado pela máquina.



**Imagem 14 – produto dispensado pela maquina**

Quando o led LEDR0 se apaga, os 3 displays voltaram ao seu estado inicial mostrando zero, pois dado que o produto foi dispensado não tem mais valores guardado na máquina e o total igualou-se a 0 novamente.



**Imagem 15 – Total zerando após dispensa do produto**

Adicionalmente, ao apertamos no botão KEY3 da FPGA independente do local que a maquina de vendas esteja, ou a posição que a ROM se encontra, todo o projeto é reset. Voltando assim, as suas configurações iniciais, presente na imagem 11 para esse respectivo teste.

Portanto, todos os testes físicos retornaram os resultados obtidos em simulador, constituindo um projeto implementável em FPGA.

#### **4. Conclusão**

Nos dias atuais, o projeto circuitos aumentaram significativamente. Como também, a utilização de memória seja RAM ou ROM junto das mesmas. Para tal, a utilização de máquinas RTL para a projeção do circuito tornou sua construção e codificação mais fáceis de serem entendidas e realizadas.

Nesse documento, foi exposto o passo a passo de como projetar uma máquina RTL para uma máquina de vendas, com uma memória ROM para salvar os tipos de moeda. Ademais, foi mostrado os resultados obtidos quando simulado o projeto utilizando a FPGA Cyclone II.

Disto podemos retirar as seguintes conclusões, dado a quantidade de componentes exigidos para a máquina de vendas foi uma importante revisão dos conceitos visto anteriormente na matéria de Circuitos Digitais que agora serão de extrema importância para o que iremos enfrentar pela frente em Sistemas Digitais. Como também, o fato de utilizarmos a FPGA para manipular e olhar nosso código funcionando como se fosse uma situação real, nos mostra a importância da matéria para os desafios que iremos encontrar no futuro da nossa profissão.

## **5. Referências Bibliográficas**

D'AMORE, R.; VHDL: descrição e síntese de circuitos digitais. 2.ed. Rio de Janeiro: LTC, 2015. 292p.

STALLINGS, William. Arquitetura e organização de computadores. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

VAHID, F.; Sistemas Digitais: projeto, otimização e HSLs; tradução Anatólio Laschuk. – Porto Alegre: Artmed, 2008. 560p.