UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CIRCUITOS DIGITAIS

MÁQUINA DE TROCOS

ELE2715 - Grupo 01 - Projeto - Problema 05

Eduardo Garcia Zaccharias

Igor Michael Araujo de Macedo

Jose Lindenberg de Andrade

Sthefania Fernandes Silva

Natal, 28 de março de 2021

Eduardo Garcia Zaccharias

Igor Michael Araujo de Macedo

Jose Lindenberg de Andrade

Sthefania Fernandes Silva

MÁQUINA DE TROCOS

Projeto da disciplina de Circuitos Digitais do Departamento de Engenharia Elétrica da Universidade do Rio Grande do Norte para relatório das atividades.

Docente: Samaherni Morais Dias

Natal, 28 de março de 2021

RESUMO

O seguinte relatório tem como objetivo desenvolver uma solução para o projeto de um circuito digital que simula uma máquina de troco que libera, ou não, em moedas, um valor determinado colocado em sua entrada, de acordo com a disponibilidade no cofre. No decorrer do texto serão mencionadas todas as condições de funcionamento de acordo com o que é solicitado, além de uma proposta de solução. A ideia é projetar uma resolução capaz de ser reproduzida, futuramente, em *softwares* de simulação de circuitos digitais, além de linguagens de descrição de *hardware*, como o VHDL. Serão abordados, assim, para o desenvolvimento, conceitos de máquinas de estados finitos, circuitos sequenciais e combinacionais, e tendo como principal foco a realização do projeto em RTL.

Palavras-chave: Máquinas de Estados Finitos; RTL; Circuitos Sequenciais; Blocos Operacionais; Blocos de Controle.

SUMÁRIO

1 INTRODUÇÃO

Em sistemas digitais, o aumento da complexidade dos circuitos gera a necessidade de arranjos lógicos que possam abranger um circuito com múltiplas funções. Um desses arranjos são os blocos de controle, estes são úteis em implementações de sistemas que necessitam de entradas e saídas para controlar um determinado comportamento, como por exemplo, a mudança de estados (VAHID, 2008).

Nesse sentido, o bloco de controle atua em junção com outro bloco construtivo que possua as entradas e saídas de dados do sistema, em particular, esse bloco deve conter registradores para armazenar e unidades funcionais para operar esses dados. Esse arranjo é conhecido como componente do nível de transferência entre registradores, do inglês *Register-Transfer-Level* (RTL). Um circuito composto por tais componentes é nomeado bloco operacional (VAHID, 2008).

A combinação de um bloco operacional com um bloco de controle gera um processador. Há diversos métodos de se projetar um processador, o mais comum é o projeto em nível de transferência entre registradores, ou projeto RTL. Nele são especificados os registradores do circuito, as possíveis transferências e operações que serão feitas com os dados de entrada; saída e dos registradores, além de definir o controle que coordena quando e como transferir e operar dados (VAHID, 2008).

Nesse contexto, para descrever o comportamento de um sistema no método RTL, há as máquinas de estados de alto e baixo nível. As máquinas de alto nível buscam descrever o comportamento do sistema de forma literal, assim, diferente das máquinas de baixo nível, as condições e ações para transição não se limitam ao uso de variáveis booleanas e sim de uma descrição do que aquela variável representa (VAHID, 2008).

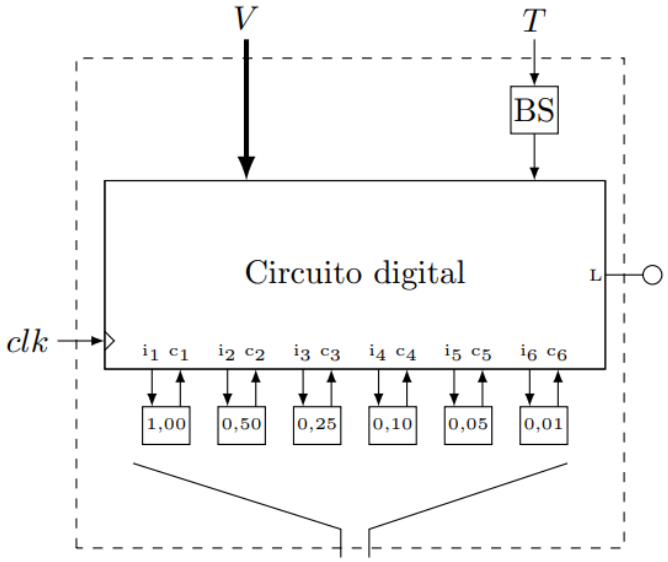
Diante do que foi exposto, o presente relatório irá discutir colocar em prática todos esses conceitos na realização de um projeto de uma máquina de trocos utilizando RTL.

2 DESENVOLVIMENTO

A problemática consiste em projetar uma Máquina de Troco (Figura 1), a qual tem a capacidade de dar um valor, em moedas, determinado pelo usuário. A liberação das moedas ocorre através de um sistema cofre que libera uma moeda sempre que em sua entrada ix (onde x = 1, 2, …, 6) existir um nível lógico alto e ocorrer um pulso de clock. A máquina possui como entrada o valor do troco (V), em binário; e uma entrada T, referente ao pulso gerado a partir da saída do circuito de um botão sincronizado (BS). Terá também uma saída L, um LED no qual, quando está piscando, indica que a máquina está processando a informação para liberar o troco e qualquer outra solicitação de troco será ignorada.

A máquina também é dotada da capacidade de verificar se algum dos cofres de moedas está vazio (cx = 0, onde x = 1, 2, …, 6) e faz o recálculo para fazer a liberação de moedas apenas por onde terá valores. Durante o processo de verificação de troco, como já foi dito, um LED ficará piscando. A máquina também tem a capacidade de informar quando não há troco para o valor fornecido, mantendo a saída LED em nível lógico alto até que um novo valor de troco seja fornecido.

Figura 1 - Diagrama de blocos da máquina de troco.



Fonte: Dados do problema.

A liberação das moedas do cofre (ix, x=1,2 ...6) e a indicação de cofre vazio (cx, x=1,2 ...6) ocorrem da forma apresentada no Quadro 1. Além disso, a entrada do circuito será realizada ao se definir um valor binário V, entre 0 e 10 reais, e fazer T = 1. A máquina só processa um troco por vez.

Quadro 1 - Liberação das moedas e indicação de cofre vazio.

Identificação	Valor	Identificação	Valor
i1 = 1	R$ 1,00	c1 = 1	Há moedas de R$ 1,00
i2 = 1	R$ 0,50	c2 = 1	Há moedas de R$ 0,50
i3 = 1	R$ 0,25	c3 = 1	Há moedas de R$ 0,25
i4 = 1	R$ 0,10	c4 = 1	Há moedas de R$ 0,10
i5 = 1	R$ 0,05	c5 = 1	Há moedas de R$ 0,05
i6 = 1	R$ 0,01	i6 = 1	Há moedas de R$ 0,01

Fonte: Elaboração própria.

2.1 PROJETO RTL

Para projetar a máquina de trocos, foi seguida a metodologia descrita por VAHID (2008), a qual divide o método de projeto RLT em 4 passos: 1º Descrever o comportamento do circuito (máquina de estados de alto nível); 2º criar bloco operacional; 3º conectar bloco operacional a um bloco de controle e 4º obter a máquina de estados finitos (FMS).

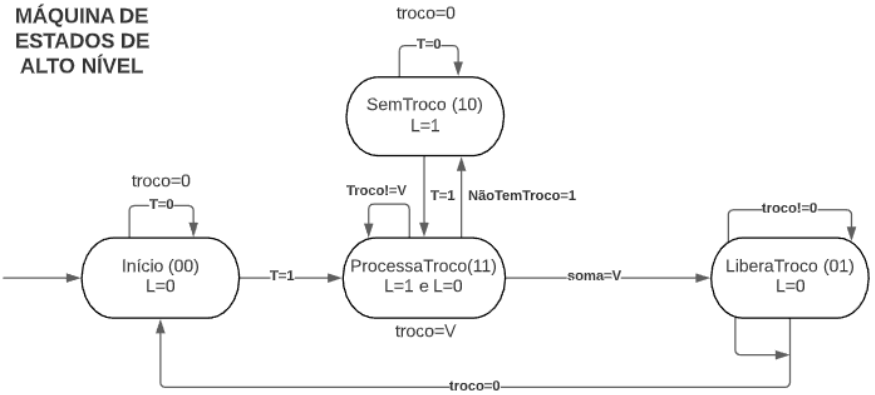
2.1.1 Máquina de estados de alto nível

Primeiramente, a máquina de trocos foi pensada para que na sua entrada seja colocado o valor do troco em centavos, logo, ela irá operar numa faixa de 0 a 1000 centavos. Em binário esse número pode ser representado em até 12 bits. Para saída foi pensada em uma ordem de precedência, assim foi dada prioridade às moedas de valores mais altos.

Para descrever o comportamento desse sistema foi montada a máquina de estados de alto nível mostrada na Figura 2. Esta conta com 4 estados: Início; ProcessaTroco; LiberaTroco e SemTroco. O estado Início significa que nenhum valor foi inserido, ou seja, o botão não foi pressionado, além disso, o LED deve estar apagado. Para sair desse estado o botão deve ser pressionado (T=1) e o próximo estado será ProcessaTroco.

No estado ProcessaTroco o LED fica piscando e uma operação para verificar se a máquina possui ou não troco deve ser realizada. Caso não haja moedas suficientes para o troco (NãoTemTroco=1), o próximo estado será SemTroco (LED aceso); caso haja moedas (soma=V), o próximo estado será LiberaTroco. Em LiberaTroco deverão ser “setadas” as saídas de acordo com a ordem de precedência e o LED tem que ficar apagado.

Figura 2 - Máquina de estados de alto nível da máquina de trocos.



Fonte: Elaboração própria.

2.1.2 Bloco operacional

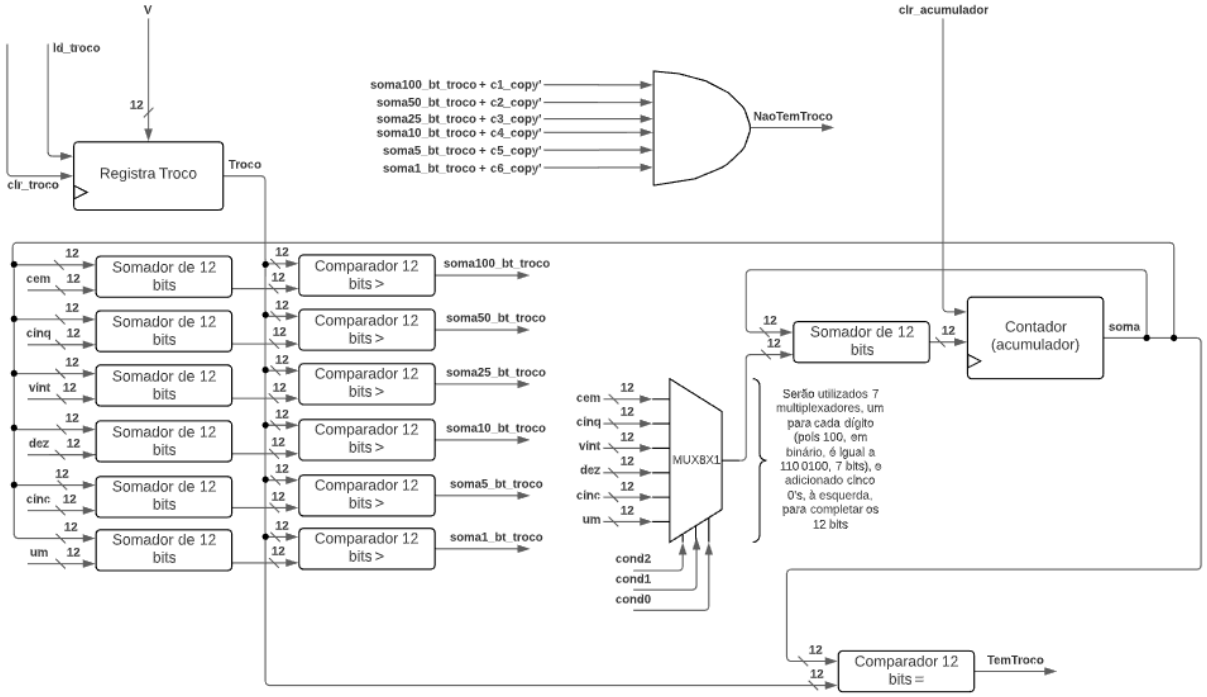
Para que a máquina funcione conforme instruído, foi criado um bloco operacional que contém todas as operações que ocorrem nos estados ProcessaTroco e LiberaTroco.

Em ProcessaTroco, recomenda-se que o valor V seja guardado em um registrador, visando evitar que o usuário altere V durante o processamento. Assim, a verificação será feita com o valor guardado. Para verificar se há ou não troco na máquina, recomenda-se o uso de somadores, subtratores e comparadores mediante condições pré-definidas.

As condições foram criadas para identificar se, com as moedas disponíveis, é possível gerar o troco solicitado. Diante disso, primeiramente, foi definida uma quantidade de moedas para cada cofre, no caso 7 (111, em binário). Então, para verificar se ainda há moedas, recomenda-se o uso de uma porta *OR*, assim, enquanto qualquer um dos 3 bits estiverem em nível lógico alto, há moedas no cofre (ci_copy=1, onde i corresponde ao valor descrito no Quadro 1); caso contrário, não há moedas naquele cofre de moedas.

Além disso, as condições também são resultado da análise da soma das moedas que o cofre possui, visto que, é preciso verificar se a máquina possui moedas suficientes para o troco ser dado. Então, usa-se um contador, em que sua saída irá alimentar 6 somadores (para cada valor de moeda), conforme mostrado na Figura 3, os quais estarão conectados à comparadores. Diante disso, caso a soma realizada seja menor que o valor armazenado (e há moedas) ele continuará somando na mesma condição, quando o valor exceder o troco (ou a moeda acabar), a próxima condição será verificada.

Figura 3 - Diagrama de blocos das operações do estado ProcessaTroco.



Fonte: Elaboração própria.

Enquanto essa verificação é feita, o número de moedas armazenadas irá diminuindo à medida que o acumulador for utilizando os valores necessários para o troco. Assim, se, por exemplo, o troco for de R$2,50 (partindo do pressuposto que todos os cofres estão cheios) serão utilizadas duas moedas de R$1,00 e uma moeda de R$0,50, assim o cofre de R$1,00 ficará com 101 bits e o de R$0,50 com 110 bits. O número de bits subtraído do cofre será guardado em liberai (para i=1, 2, …, 6).

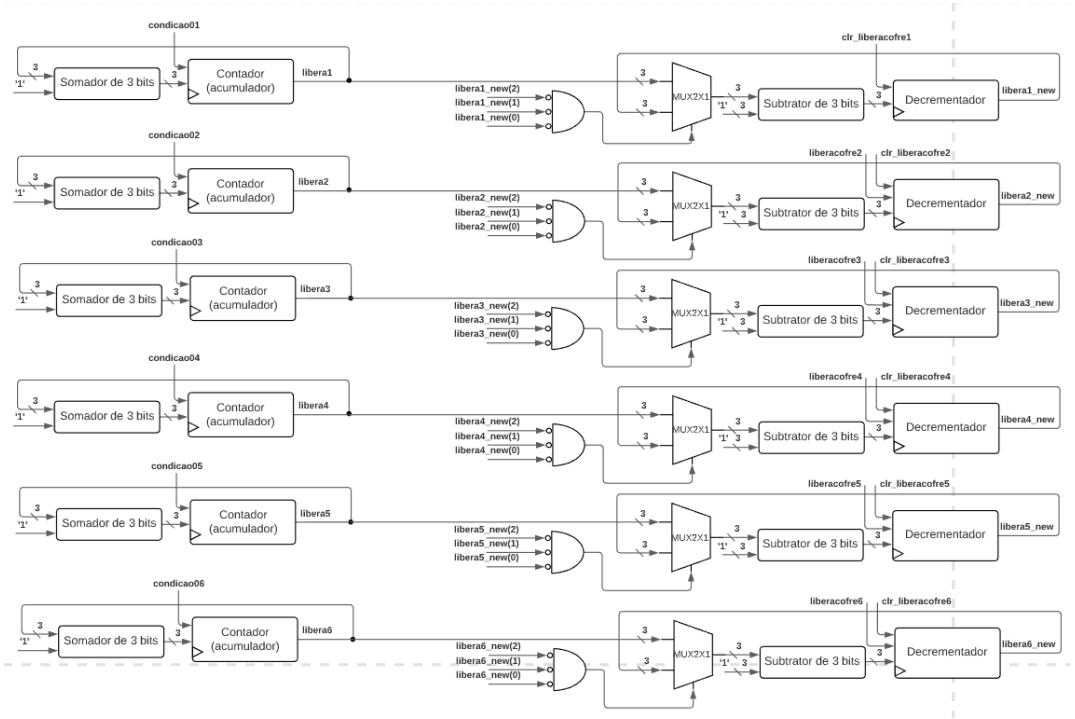
O resultado do contador será comparado com o valor inserido. Caso os valores sejam iguais, TemTroco=1 e o próximo estado será LiberaTroco. Mas, para verificar se não há troco, uma condição diferente precisa ser verificada, pois TemTroco=0 já na primeira verificação, por exemplo, fazendo uma falsa afirmação que não tem troco, enquanto ele ainda nem comparou. Sendo assim,, a condição para não ter troco é caso todas as condições tenham sido verificadas e não haja mais moedas de 1 centavo. O Anexo B mostra todas as condições.

Já no estado LiberaTroco, as saídas serão setadas utilizando a variável liberai, mencionada anteriormente, que informa o número de moedas de cada cofre que deve ser liberado. Isso será feito de forma gradativa, do maior para o menor, diminuindo um bit de cada liberai até todos chegarem em 0. Para identificar se todas as moedas necessárias para dar o troco já estão zeradas, há a variável cofrei_liberado que verifica se liberai é igual a 0.

Assim, a liberação ocorre conforme descrito: primeiro libera as moedas de R$1, em seguida, caso já tenha liberado as moedas do cofre de 1 real (cofre1_liberado), então permite liberar as moedas de 50 centavos (liberacofre2); após liberadas (cofre2_liberado), permite-se liberar as moedas de 25 centavos (liberacofre3); e assim por diante até que todas as moedas necessárias forem liberadas. A variável que define que todo o troco já foi dado é TrocoLiberado, a qual terá nível lógico alto quando todas as moedas dos cofres forem liberadas.

A sinalização da saída será as variáveis ix em nível lógico alto, enquanto estiverem sendo liberadas as moedas do seu respectivo cofre. Se for liberada mais de uma moeda do mesmo valor, o nível lógico deverá ficar alto por quantos pulsos de *clock* forem necessários para todo o valor ser dado. No exemplo do troco de R$2,50, a saída i1 deverá ficar em 1 por dois pulsos de clock e, em seguida, a saída i2 por apenas um.

Figura 4 - Diagrama de blocos das operações do estado LiberaTroco.



Fonte: Elaboração própria

As variáveis condicao0i (com i=1, 2, …, 6) são de acordo com as condições vistas no Anexo B, por exemplo: *condicao01=cond(2)=0 && cond(1)=0 && cond(0)=1; condicao02: cond(2)=0 && cond(1)=1 && cond(0)=0;* e assim por diante. E as variáveis clr_liberacofrei

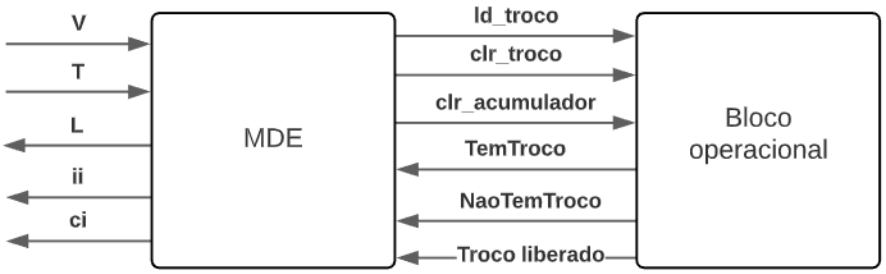
(com i=1, 2, …, 6) são utilizadas para o decrementador fixar em ‘0’ quando não for necessário decrementar, sendo assim, utilizando a variável cofrei_liberado.

2.1.3 Bloco operacional e bloco de controle

O bloco operacional descrito no tópico anterior será conectado a um bloco de controle, o qual possui como entradas V e T; e como saídas L, libera moeda ou não (ix, x=1, 2, …, 6) e os cofres (cx, x=1, 2, …, 6). Já as entradas do bloco operacional são ld_troco, clr_troco e clr_acumulador; e as saídas são TemTroco; NãoTemTroco e TrocoLiberado.

No que tange as entradas do bloco operacional, ld_troco será o responsável por permitir o registro do valor inserido, clr_troco irá limpar esse registrador quando não houver mais necessidade do uso do valor inserido e por último clr_acumulador vai limpar o contador que está acumulando os valores em ProcessaTroco. A Figura 5 ilustra como estão conectados.

Figura 5 - Bloco de controle e bloco operacional da máquina de trocos.

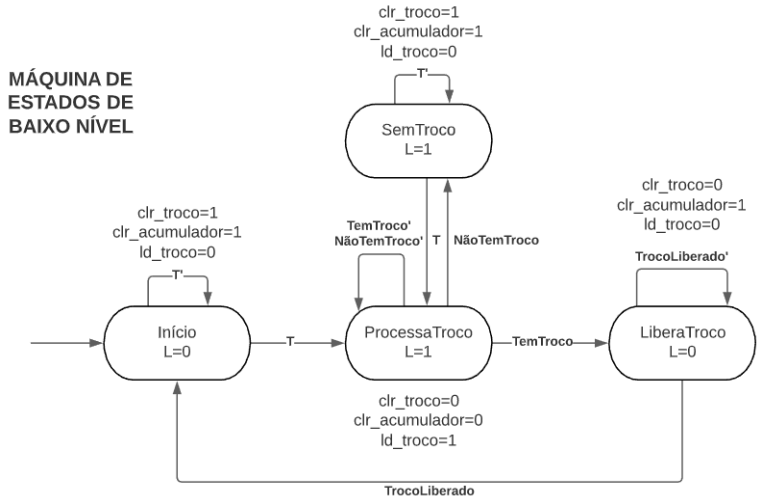


Fonte: Elaboração própria.

2.1.4 Máquina de estados finitos

Após definidos os blocos de controle e operacional, a máquina de estados finita foi montada. Esta é descrita pela máquina de estados de baixo nível mostrada na Figura 6.

Figura 6 - Máquina de estados de baixo nível da máquina de trocos.



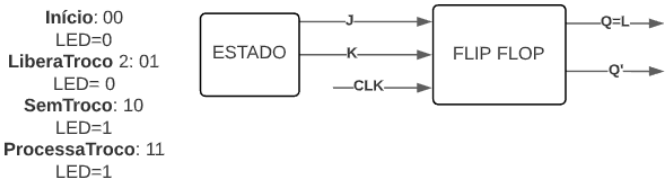
Fonte: Elaboração própria.

A partir dela foi montada a tabela estados mostrada no Anexo C, contendo as variáveis de estado, estados atual e futuro e as saídas.

Note que a saída LED é somente 0 ou 1 na tabela de estados, no entanto, conforme mencionado na problemática, o LED possui 3 comportamentos diferentes de acordo com o estado. Nos estados Início e LiberaTroco o LED deve está apagado, em SemTroco o LED deve está aceso, já em ProcessaSenha o LED deve está piscando. Para tornar isso possível, foi criada uma lógica pautada nos estados e na saída LED de cada um.

Assim, o estado atual *AND* L são colocados na entrada no Flip-flop JK. Devido a configuração do flip-flop o LED será capaz de piscar quando estiver no estado ProcessaTroco. Isso acontece pois, ao inserir 1 nas entradas J e K, o flip-flop fica instável mudando sua saída a cada pulso. Nos demais estados, o LED irá se comportar conforme exigido. A Figura 7 ilustra o que foi descrito.

Figura 7 - Lógica do acendimento do LED.

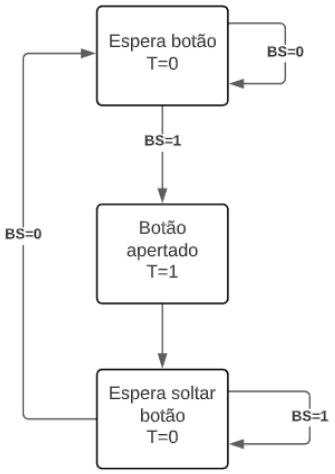


Fonte: Elaboração própria.

2.2 BOTÃO SÍNCRONO

Para a criação do botão síncrono foi feita a máquina de estados mostrada na Figura 8. Esta máquina irá garantir que, quando pressionado o botão, T terá nível lógico alto durante apenas 1 *clock*, solucionando o problema de que, por exemplo, o usuário aperte o botão e fique pressionado por um certo tempo.

Figura 8 - Máquina de estados do botão síncrono.



Fonte: Elaboração própria.

3 CONCLUSÃO

A complexidade dos sistemas digitais exige combinações de circuitos sequenciais, combinacionais e uma padronização/sistematização no processo. Uma dessas combinações, nomeada processador, faz uso de blocos de controle e blocos operacionais. Uma aplicação do dia-a-dia desse conceito é uma máquina de trocos, a qual foi projetada no presente relatório.

A máquina projetada precisa dar um valor em moedas, determinado pelo usuário. A liberação das moedas ocorre através de um sistema cofre que libera uma moeda sempre que em sua entrada ix (onde x = 1, 2, …, 6) existir um nível lógico alto e ocorrer um pulso de clock. A máquina possui como entrada o valor do troco em binário (V), uma entrada T, referente ao pulso gerado a partir da saída do circuito de um botão sincronizado (BS). E uma saída L, na qual quando está piscando indica que a máquina está processando a informação para liberar o troco e qualquer outra solicitação de troco será ignorada.

O circuito também tem a capacidade de informar quando não há troco para o valor fornecido, mantendo a saída LED em nível lógico alto até que um novo valor de troco seja fornecido.

O projeto foi feito embasado no método de projeto RTL, assim foi feito um bloco de controle e um bloco operacional. Além disso, outros circuitos mais simples foram utilizados para compor a máquina, como: somadores, subtratores e comparadores. Ainda assim, o circuito projetado possui limitações e algumas funcionalidades não foram projetadas completamente.

4 REFERÊNCIAS

VAHID, Frank. **Sistemas digitais: projeto, otimização e HDLS**. Rio Grande do Sul: Artmed Bookman, 2008. 558 p.

ANEXOS

ANEXO A - Relato semanal

Líder: Jose Lindenberg de Andrade

A.1 Equipe

Tabela 1 - Identificação da equipe.

Função do grupo:	Discente
Redator	Sthefania Fernandes Silva
Debatedor	Igor Michael Araujo de Macedo
Videomaker	Eduardo Garcia Zaccharias
Auxiliar	-

Fonte: Elaboração própria.

A.2 Defina o problema

Projetar o funcionamento de uma máquina de troco, contendo entrada do valor do troco (V), um botão sincronizado (BS) funcionando com uma entrada T, valores de R$ 1,00, R$ 0,50, R$ 0,25, R$ 0,10, R$ 0,05 e R$ 0,01 que podem serem disponibilizados para o troco. A máquina disponibiliza um troco por vez e informa caso não haja troco disponível. Além disso, possui mecanismo de liberação de moedas utilizando o clock é um mecanismo de indicação de quando o cofre está vazio.

A.3 Registro do *brainstorming*

Inicialmente o grupo voltou-se para entender a situação-problema e estudou os assunto sugeridos para resolução desta. Após, foram discutidas em grupo as ideias para a construção do projeto, sendo verificado as viabilidades para a problemática e aderido às sugestões encontradas por meio de pesquisas na medida que as dificuldades apresentavam-se. Ademais, o grupo construiu sempre um consenso para resolução das demandas.

A.4 Pontos-Chaves

Entendimento de Máquinas de Estados Finitos, RTL, Blocos Operacionais e Controle.

A.5 Questões de pesquisa

Foram realizadas pesquisas para melhor entendimento de registradores, máquinas de estados de alto nível, Projeto RTL, Flip-Flop JK, conversão de bases binárias e divisão em binário, CI's divisores, apesar de não utilizados.

A.6 Planejamento da pesquisa

Inicialmente foi elencado o primeiro dia, 23 de março, para estudo do assunto base para projetar a máquina de moedas, após isso foi dado início as reuniões em grupo e a medida que as dificuldades da semana de projeto iam surgindo, buscava-se a ajuda da monitoria ou as consultas no livro texto e internet.

ANEXO B - Tabela de condições

CONDIÇÕES	Cond 2	Cond 1	Cond 0	Expressão
0	0	0	0	(soma100_bt_troco)! && c_copy_1 ==1
1	0	0	1	((soma100_bt_troco) || c_copy_1==0) && (soma50_bt_troco)! && c_copy_2==1
2	0	1	0	((soma100_bt_troco) || c_copy_1==0) && ((soma50_bt_troco) || c_copy_2==0) && (soma25_bt_troco)! && c_copy_3==1
3	0	1	1	((soma100_bt_troco) || c_copy_1==0) && ((soma50_bt_troco) || c_copy_2==0) && ((soma25_bt_troco) || c_copy_3==0) && (soma10_bt_troco)! && c_copy_4==1
4	1	0	0	((soma100_bt_troco) || c_copy_1==0) && ((soma50_bt_troco) || c_copy_2==0) && ((soma25_bt_troco) || c_copy_3==0) && (soma10_bt_troco)! && c_copy_4==2
5	1	0	1	((soma100_bt_troco) || c_copy_1==0) && ((soma50_bt_troco) || c_copy_2==0) && ((soma25_bt_troco) || c_copy_3==0) && ((soma10_bt_troco) || c_copy_4==0) && ((soma5_bt_troco) || c_copy_5==0)&& (soma1_bt_troco)! && c_copy_6==1
6	1	1	0	((soma100_bt_troco) || c1==0) && ((soma50_bt_troco) || c2==0) && ((soma25_bt_troco) || c3==0) && ((soma10_bt_troco) || c4==0) && ((soma5_bt_troco) || c5==0)&& ((soma1_bt_troco) || c6==0)

Fonte: Elaboração própria.

ANEXO C - Tabela de estados

	ESTADO ATUAL		ENTRADA				ESTADO FUTURO		SAÍDA
ESTADOS	Q1	Q0	T	TEMTROCO	ÑTEMTROCO	TROCOLIBERADO	E1	E0	L
INÍCIO	0	0	0	x	x	x	0	0	0
	0	0	1	x	x	x	1	1	0
PROCESSA TROCO	1	1	x	0	0	x	1	1	1
	1	1	x	1	0	x	0	1	1
	1	1	x	0	1	x	1	0	1
SEM TROCO	1	0	0	x	x	x	1	0	1
	1	0	1	x	x	x	1	1	1
LIBERA TROCO	0	1	x	x	x	0	0	1	0
	0	1	x	x	x	1	0	0	0