

RELATÓRIO – Máquina RTL Ordenador

Dados de Identificação	
Componente Curricular	ELE2715.0 – Circuitos Digitais – Teoria
Data do experimento	18/11/2018
Componente	2016012955 - Atyson Jaime de Sousa Martins

1 - Introdução

Nesse relatório, estará presente todo o procedimento para escolha e construção da máquina RTL Ordenação. Mas, antes vamos 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 meche com dados do processador). Para se projetar uma máquina dessa é preciso seguir alguns passos, como: Obter a máquina de estados de nível alto; criar o bloco operacional; obter a máquina de estados finito do bloco de controle (FSM).

2 – Desenvolvimento

2.1 – Escolha dos Processos

Para a idealização do projeto da Máquina RLT, pensamos em um modo onde a máquina iria possuir uma menor lógica e facilitasse nosso trabalho como projetista, sendo assim, pensamos na técnica do Trono para a ordenação dos números que serão colocados na máquina. Essa técnica consiste da seguinte forma, a partir de um vetor de valores, pegamos a primeira posição desse vetor e dizemos que ele é o maior, e vamos comparando com os demais valores do vetor, caso exista, trocamos eles de posição, quando se chega ao final do vetor, passamos para a próxima posição e fazemos a comparação de novo, até que todos os valores do vetor estejam ordenados da maneira correta.

2.2 – Controlador

Para fazer o Controlador da Máquina RTL, definimos quatro estágios, inicial, wait, load e ordenação. No estado inicial a máquina irá dar clear em todos os registradores do DataPath, após isso, ela irá automaticamente para o estado de wait, nesse estado ele irá esperar o usuário apertar no botão de Load ou de Go, se ele apertar no botão de Load, ele irá para o estado de load no qual como o próprio nome diz, ele dará load nos registradores, como também, mandará um sinal para os seletores dos mux, fazendo o valor que você colocar na chave entrar para o registrador, caso o usuário aperte o botão de Go, ele irá começar a ordenação dos Registradores indo para o estado de ordenação, a partir daí, ele fica esperando o sinal de done, mas como segue uma lógica inversa para que um led fique aceso enquanto o datapath estiver fazendo a ordenação, enquanto done for um, ele continua no estado, mas se o done for 0, ele volta para o estado de wait e espera novamente o usuário decidir o que irá ser feito. A partir disso, foi desenhado a seguinte máquina de estados:

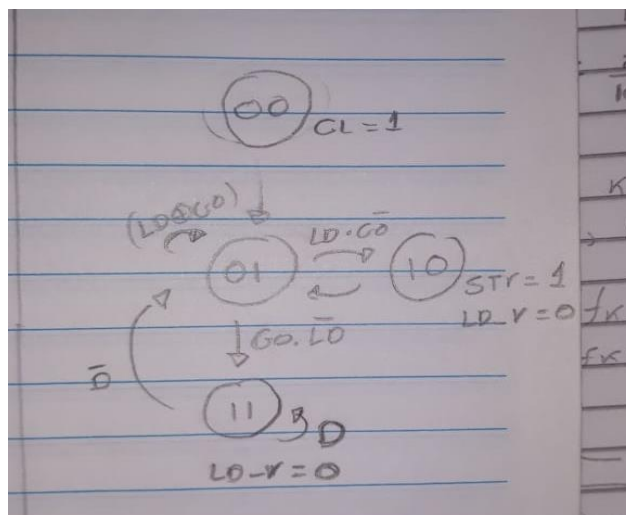


Foto 1 – Máquina de Estados

No qual, Ld (Load), Go (Go), D (Done), Cl (Clear), Str (Sinal para os Seletores do Mux). Já os estados 00 – inicial, 01 – wait, 10 – load, 11 – ordenador. Como são quatro estados, iremos precisar de dois flip flop JK, com isso em mão, codificamos os estados para encontrar a lógica combinacional fazendo o mapa de karnaugh. Assim, considerando 1 como JK mais significativo, e um vetor S (1 down to 0) como as saídas do JK, encontramos as seguintes lógicas:

$J1 = S0 \text{ AND } (LD \text{ XNOR } GO);$

$K1 = \text{Not}(S0) \text{ AND } \text{Not}(D),$

$J0 = 1,$

$K0 = \text{Not}(S1) \text{ AND } LD \text{ AND } \text{Not}(GO),$

$LD_R = \text{Not}(S1),$

$CL_R = \text{Not}(S1) \text{ AND } \text{Not}(S0),$

$STR = S1 \text{ AND } \text{Not}(S0),$

Por conseguinte, criamos a maquina de estados no computador primeiramente para fazer os testes devidos e assim passar para protoboard ficando dessa forma finalmente:

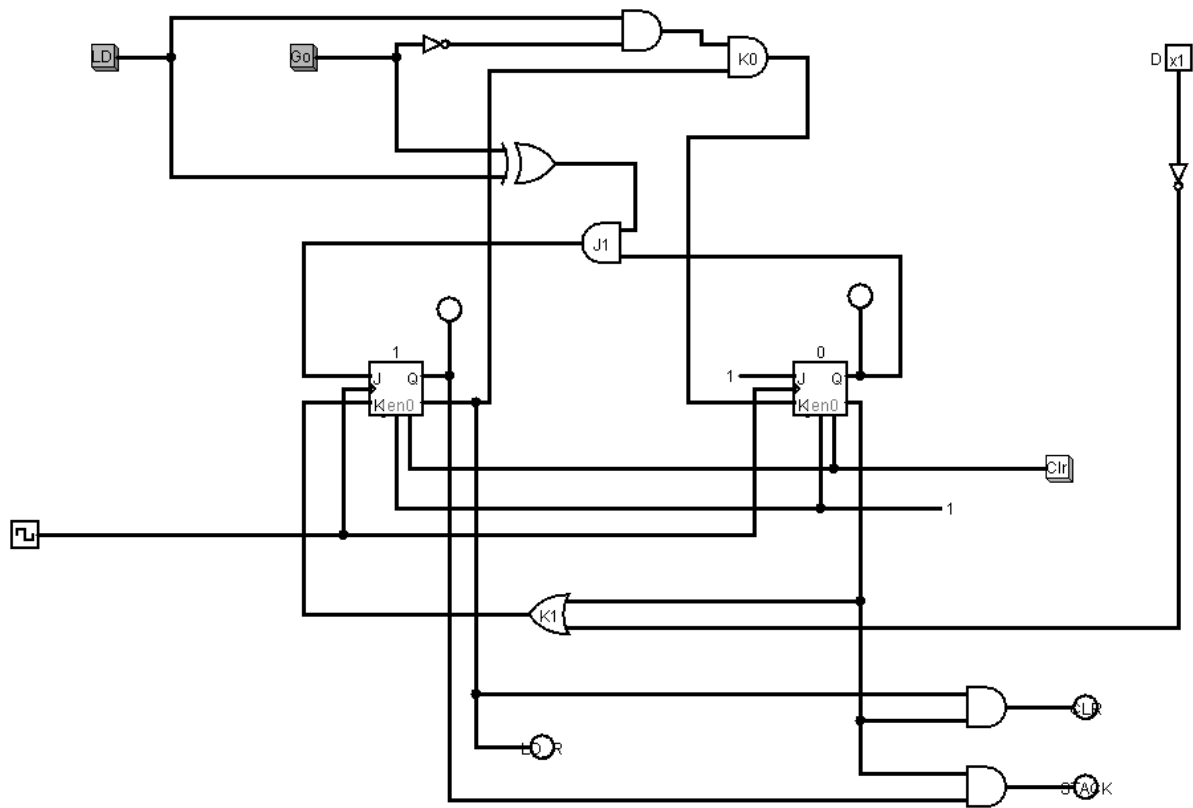


Foto 2 – Maquina de Estados

Um adendo, colocamos um D como uma constante nesse momento para poder fazer o teste na maquina e saber se todos os estados estavam funcionando perfeitamente.

2.3 – Datapath

Para fazer o datapath usamos registradores, mux, comparadores e os display para mostrar os valores que estavam sendo colocados. A primeira parte do data path, consistia nas entradas paralelas, ou seja, um registrador e para cada registrador havia um mux de quatro entradas para uma saída, a partir do load dado na maquina de estados, o registrador pegava o valor que vinha da entrada “00”. Eram necessários dar três loads para finalizar o processo de carregamento para começar o processo de ordenação, por que, quando se dava o primeiro load, o que colocamos na entrada ia para o primeiro registrador, quando dava o segundo load, o que tava no primeiro ia para o segundo e novamente o primeiro pegava o que havia na entrada e no ultimo load, o que estava no segundo ia para o terceiro registrador, o que estava no primeiro ia para o segundo e o primeiro pegava o que havia na entrada. Sendo assim, os registradores estavam carregados e esperando o botão de go ser acionado.

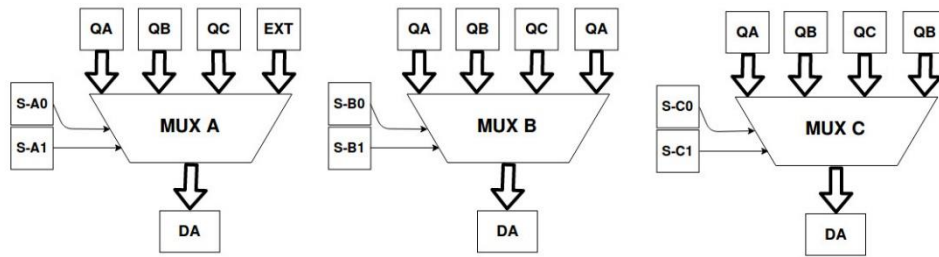


foto 3 – Mux dataPath

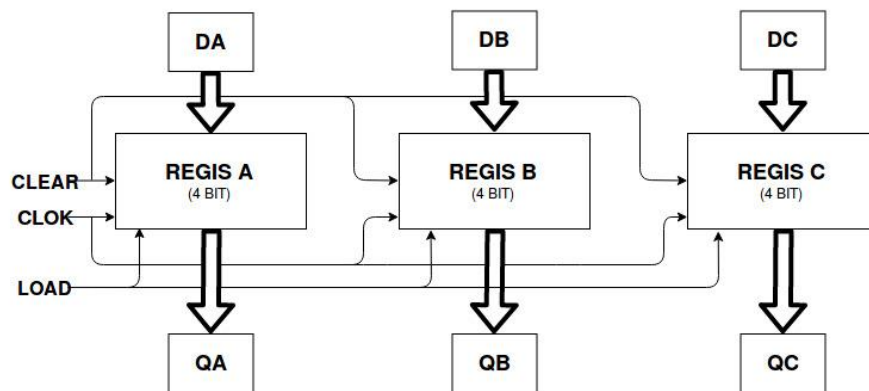


Foto 4 – Reg. DataPath

Após tudo isso e o botão go ser pressionada, as saídas dos registradores entram nos comparadores, no qual se verifica se $A < B$ e se $B < C$, a saída desse comparadores entram em um loop interno a partir das portas logicas que mandam o sinal das entradas do mux, que ao dar o pulso de clock vão atualizando os valores dos registradores até que esteja sendo mostrado no display $A > B > C$, enquanto o datapath estiver em seu loop interno um led fica aceso, quando o loop termina, esse led apaga e um sinal de Done e enviado a máquina de estados, que o mesmo volta para o estado de Wait, esperando novamente você dar os loads ou clear na máquina.

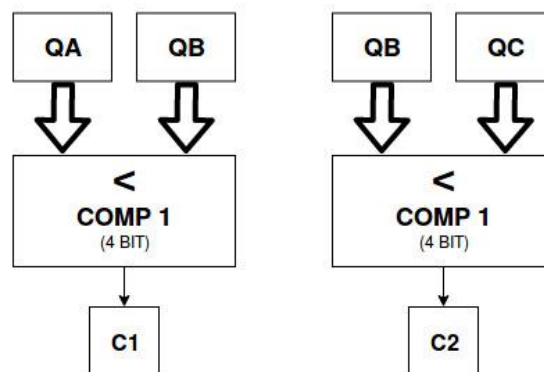


Foto 5 - Comparadores

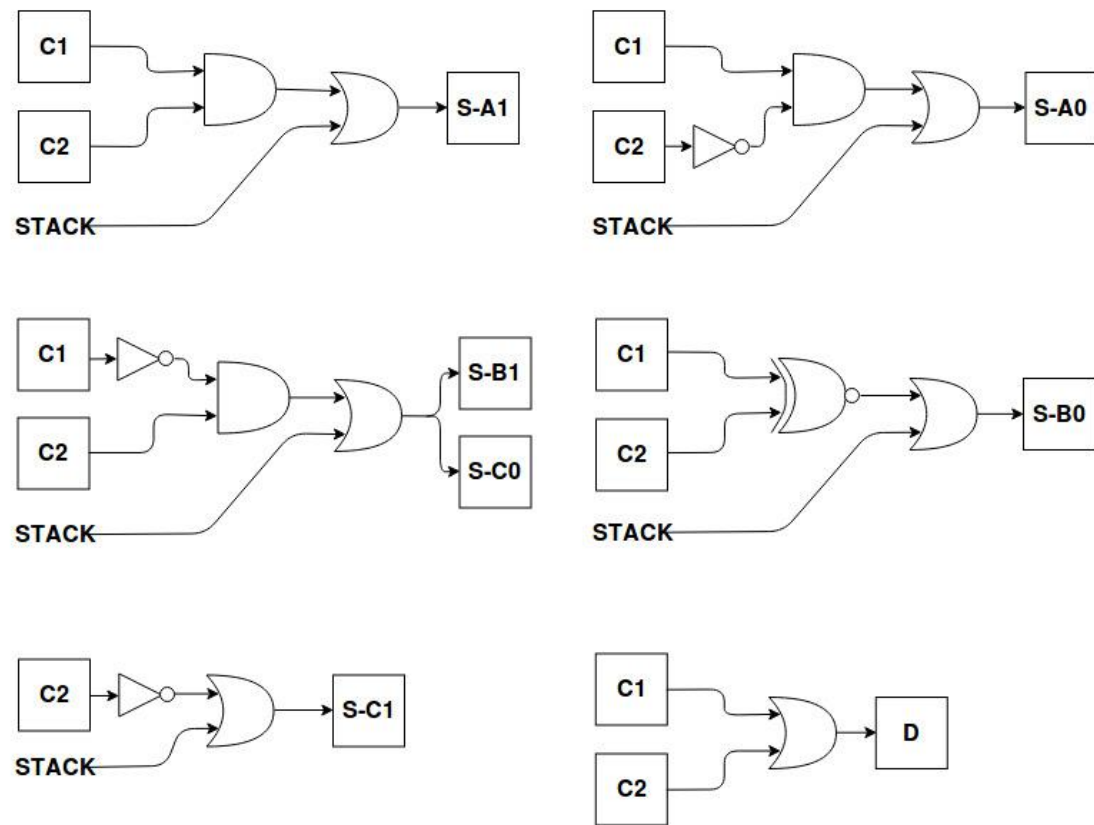


Foto 6 – Portas Logicas para os Mux