

Universidade Estadual do Oeste do Paraná (UNIOESTE)

Centro de Engenharias e ciências exatas (CECE)

Ciência da Computação

**PROJETO – Sistemas Digitais**

Gabriel José Biudes Lino – RA: 169299

Professor Jorge Habib Hanna El Khouri

Foz do Iguaçu, PR

Maio de 2022

## INTRODUÇÃO

Este documento foi produzido com o intuito de desenvolver os exercícios propostos na segunda avaliação da disciplina de Sistemas Digitais. Sendo assim, será apresentado, passo a passo, o processo de desenvolvimento dos circuitos digitais correspondentes, desde a parte teórica até a implementação prática no circuit.js.

## DESENVOLVIMENTO DOS EXERCÍCIOS

- 1) Implementar uma função (mais\_uns) que avalia uma entrada de 4 bits e detecta se a quantidade de 1s é maior que a quantidade de 0s. Explicar a solução.
  - a) Use Mapa de Karnaugh para obter a função;

Primeiramente, foi montada uma tabela verdade contando todas as 16 possibilidades para as 4 entradas de dados e suas respectivas saídas. A saída X vale 1 caso a quantidade de 1s seja maior que a quantidade de 0s, e 0 caso contrário:

D3	D2	D1	D0	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

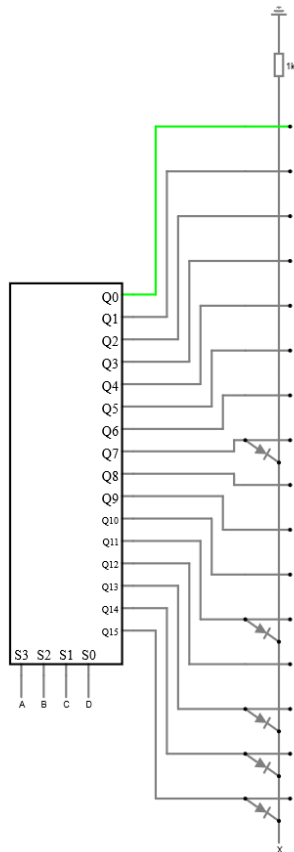
Portanto, pode-se concluir que  $X = \sum(7, 11, 13, 14, 15)$ . A partir disso, é possível montar o Mapa de Karnaugh para a variável X:

AB/CD	00	01	11	10
00				
01			1	
11		1	1	1
10			1	

Assim, os agrupamentos utilizados foram 7 com 15, 11 com 15, 13 com 15 e 14 com 15, produzindo a expressão lógica  $X = BCD + ACD + ABD + ABC$ .

b) Implemente a solução utilizando ROM;

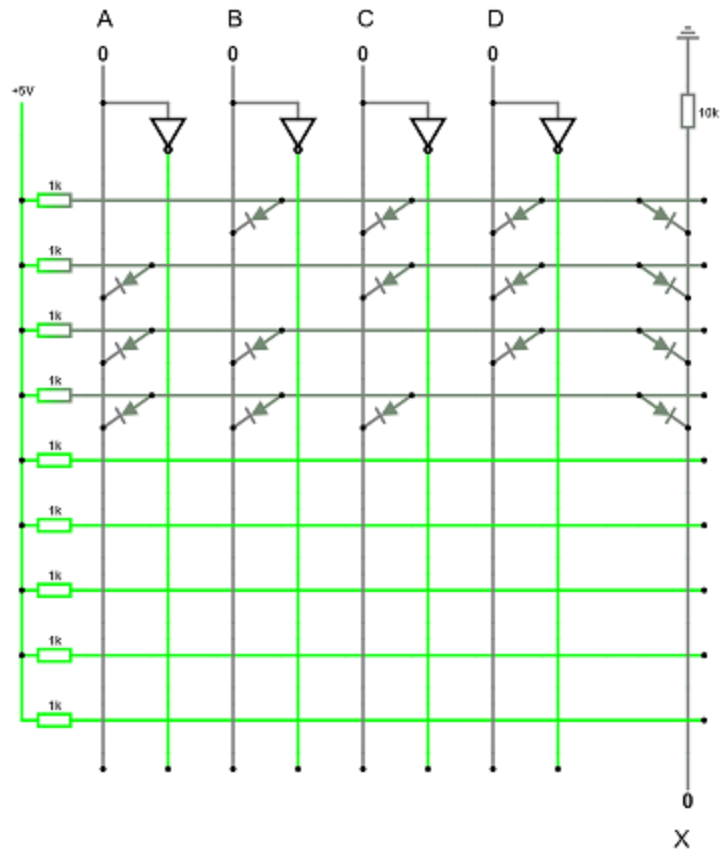
Para o desenvolvimento utilizando ROM, foi utilizado um decoder 4-16. Assim, foi criada uma ROM que guarda 1 para as posições onde a quantidade de 1s é maior que a quantidade de 0s, e 0 caso contrário. Para isso, basta que seja adicionado um diodo onde se queira que o resultado seja 1, ou seja, nas saídas 7, 11, 13, 14 e 15 do decoder:



<https://tinyurl.com/2pj4jsxv>

c) Implemente a solução utilizando PLA;

Utilizando PLA, os diodos da parte esquerda podem ser considerados um AND. Assim, a primeira linha representa BCD, a segunda linha representa ACD, a terceira linha representa ABD e a quarta linha representa ABC. Na parte direita, os diodos podem ser considerados um OR. Assim, nesse caso, X seria  $BCD + ACD + ABD + ABC$ , que é a expressão lógica que retorna 1 caso a quantidade de 1s seja maior que a quantidade de 0s. Portanto, o circuito utilizando PLA fica da seguinte forma:

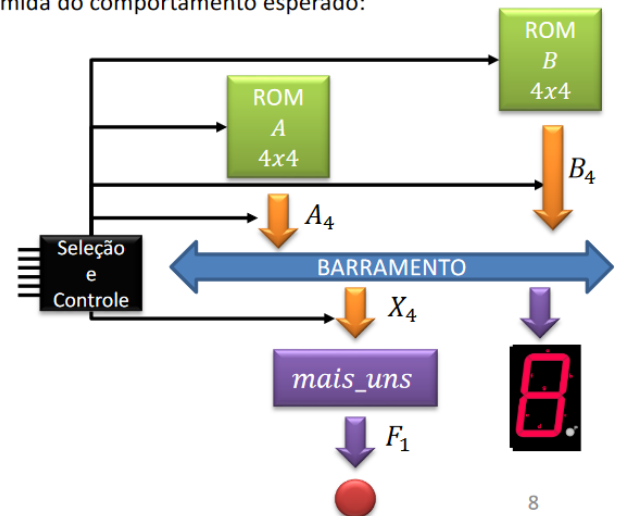


<https://tinyurl.com/2zhj4aag>

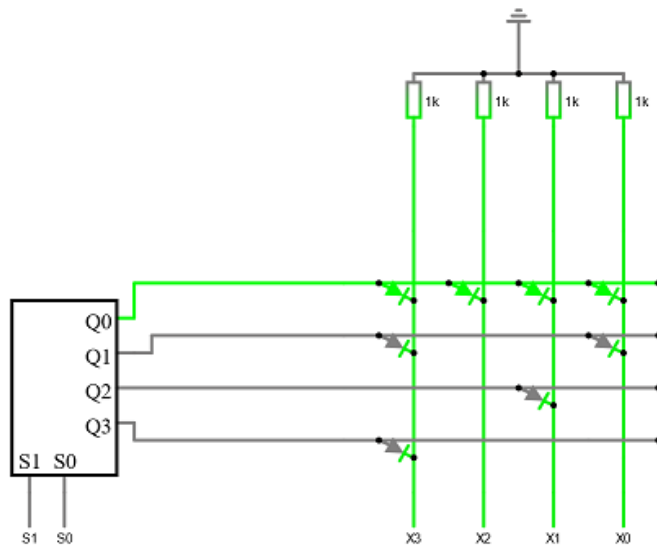
2)

[60] Um determinado circuito deve dispor de duas *ROMs*, um barramento compartilhado, um circuito com uma das lógicas do exercício anterior (*mais\_uns*), um *LED*, um display e elementos de seleção e controle. Projetar e implementar um circuito com a seguinte especificação resumida do comportamento esperado:

- Entrada de Seleção permitirá definir o endereço desejado para a *ROM A* ou *B*;
- Elemento de Controle habilitará que o dado de uma das *ROMs* ( $A_4$  ou  $B_4$ ) acesse o *BARRAMENTO*;
- Elemento de Controle habilitará que a informação que está no *BARRAMENTO* ( $X_4 = A_4 | B_4$ ) seja liberada para o circuito *mais\_uns*;
- O Flag  $F_1$  fará o *LED* acender se  $X_4$  cumprir com a lógica da função *mais\_uns*;
- O Display de 7 Segmentos sempre apresenta o valor que se encontra no *BARRAMENTO*;
- A solução para os Controles deverá contemplar *Three-State Buffers*.

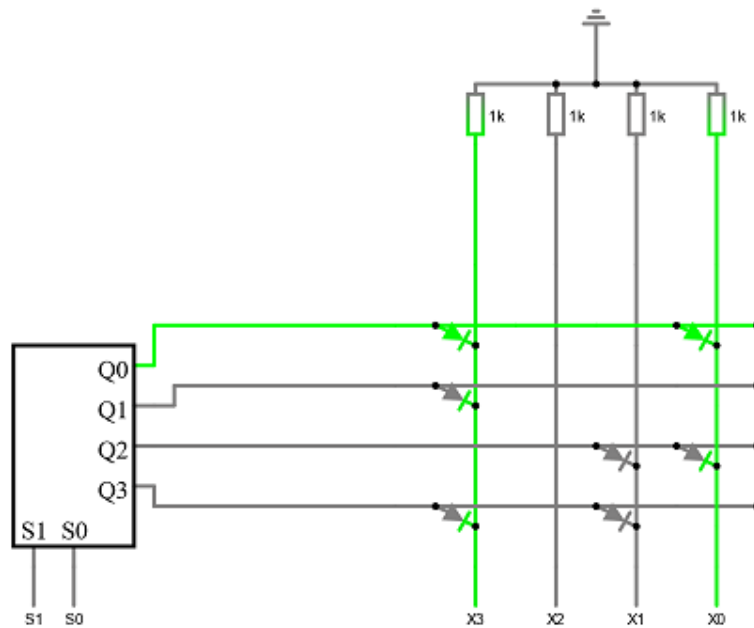


Para começar, foi desenvolvida e configurada a memória ROM A da seguinte forma:



<https://tinyurl.com/2z825aor>

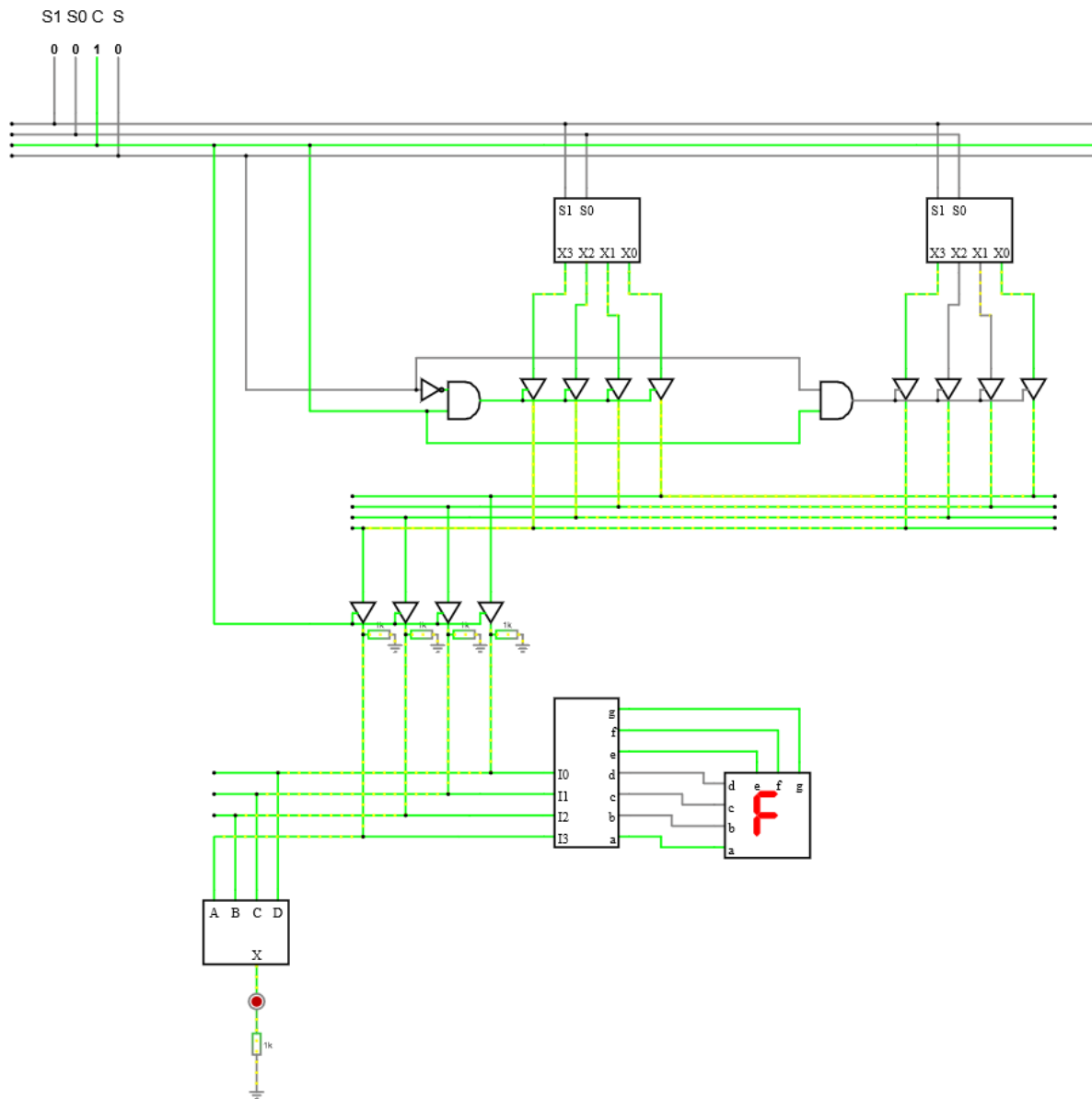
E, então, a memória ROM B:



<https://tinyurl.com/2lj22mxs>

No circuito completo, ambas as memórias recebem as mesmas entradas S1 e S0. Além disso, a entrada C, de controle, serve para controlar a entrada e saída de dados dentro do barramento através do uso de Three-State Buffers. Por fim, temos também a entrada S, de seleção, que define qual das duas memórias ROMs presentes será utilizada. No circuito final, o circuito da função mais\_uns e ambas as memórias ROMs foram colocadas na forma de subcircuito, de modo a deixar o circuito final o mais limpo possível. O circuito completo se encontra logo na figura abaixo, e mostra o resultado do circuito para o primeiro valor armazenado dentro da primeira memória ROM:

Circuito completo:



<https://tinyurl.com/2fab9v6c>

Assim, no resultado, o valor é exibido através do uso de um display de 7 segmentos. Além disso, o LED acende caso a quantidade de 1s seja maior do que a quantidade de 0s; caso contrário, fica apagado, conforme foi solicitado.

## **CONCLUSÃO**

Por fim, após o término do projeto, pode-se entender de maneira mais clara sobre o uso de memória ROM e elementos de controle, estes que são elementos cruciais para o desenvolvimento de determinados sistemas digitais.

## **REFERÊNCIAS**

Falstad:

<https://tinyurl.com/2pj4jsxv>

<https://tinyurl.com/2z825aor>

<https://tinyurl.com/2lj22mxs>

<https://tinyurl.com/2fab9v6c>

<https://tinyurl.com/2zhj4aag>