

# Experimento 7

## Implementação de Circuitos Combinacionais com Multiplexadores

Lucas Mafra Chagas, 12/0126443  
Marcelo Giordano Martins Costa de Oliveira, 12/0037301

<sup>1</sup>Dep. Ciência da Computação – Universidade de Brasília (UnB)  
CiC 116351 - Circuitos Digitais - Turma C

{giordano.marcelo, chagas.lucas.mafra}@gmail.com

**Abstract.** *In this experiment, we focus on building combinational circuits using Multiplexers.*

**Resumo.** *O foco desse experimento é construir Circuitos Combinacionais utilizando Multiplexadores.*

### 1. Objetivos

O objetivo deste experimento é adquirir conhecimento na implementação e também utilização de multiplexadores, que são circuitos combinacionais.

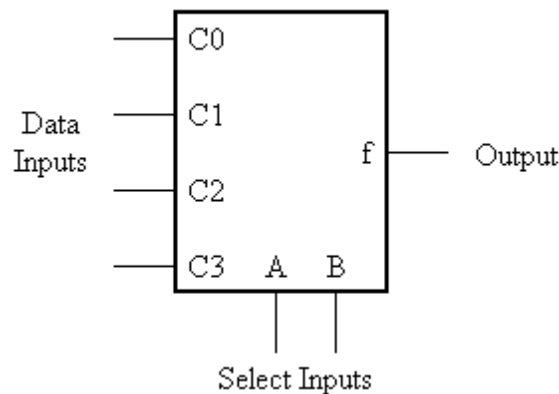
### 2. Materiais

- Painel Digital
- *protoboard*
- Fios conectores
- Portas Lógicas NAND e NOT
- 2 x MUX-4
- DECOD-16

### 3. Introdução

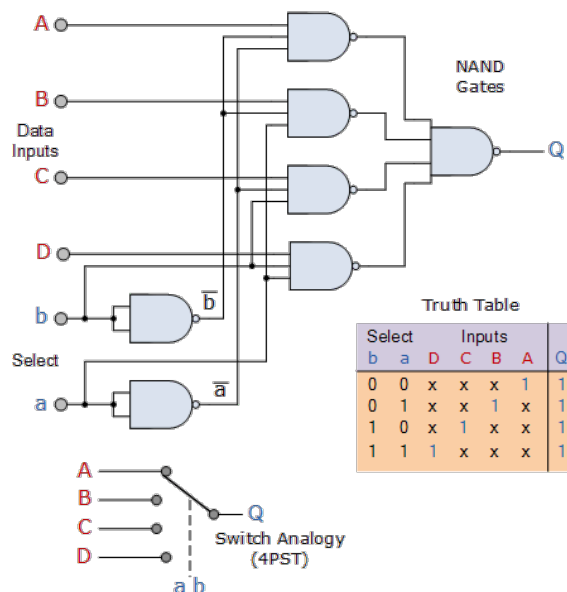
#### 3.1. Multiplexadores e Demultiplexadores

Um multiplexador é um circuito combinacional lógico projetado que tem como saída única e compartilhada um e somente um de seus inputs de dados, a partir da aplicação de um sinal de controle. O demultiplexador, por sua vez, realiza a operação inversa.



**Figure 1. Estrutura de um Multiplexador**

Sob o ponto de vista da implementação de seus circuitos, os multiplexadores e demultiplexadores são simplesmente circuitos combinacionais com diversas entradas e somente uma saída, ou vice-versa.



**Figure 2. Implementação de um multiplexador com 7 portas NAND's**

### 3.2. Aplicações

A eficiência dos sistemas de comunicação pode ser consideravelmente aumentada ao utilizar-se multiplexadores, por permitir a transmissão de diferentes tipos de dados (como áudio e vídeo) ao mesmo tempo, usando uma única linha de transmissão. Eles também são utilizados para implementar grandes quantidades de memória no computador, reduzindo drasticamente a quantidade de fios de cobre necessários para ligar conectar a memória a outras partes do circuito. Isso se deve ao fato de um multiplexador ser capaz de implementar qualquer função booleana genérica. Quando temos uma função em que obter a tabela verdade não é tão simples, a utilização de multiplexadores e demultiplexadores pode simplificar o problema da implementação dessa função.

## 4. Procedimentos

### 4.1. Usar um MUX - 4 duplo em MSI para implementar um somador completo.

Para um somador completo, temos que considerar as entradas A,B e C, com A e B sendo os bits na qual se quer somar e C sendo o carry vindo da soma anterior. Como saídas temos que considerar T e S, com S sendo o valor da soma dos 3 bits de entrada e T sendo o carry gerado por essa soma. Temos portanto, que a tabela verdade para esse circuito seria:

Entradas			Saídas	
A 1º bit	B 2º bit	C Vem Um	T Vai Um	S Soma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 3. Tabela verdade de um Somador Completo

Utilizando a tabela na Figura 5 como referência e tendo em mente que o circuito deveria ser montado a partir de um MUX - 4 duplo, o seguinte esquema para o circuito a ser implementado foi projetado:

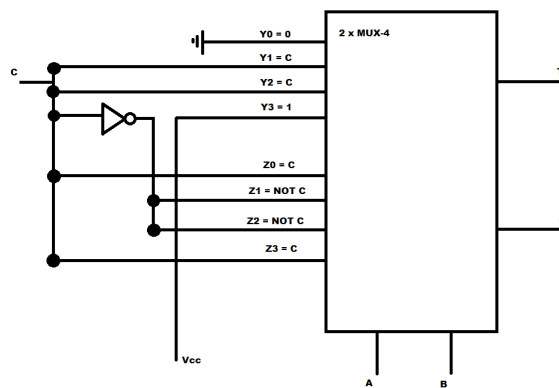


Figure 4. Esquema de um somador completo com 3 entradas e 2 saídas montado com um Multiplexador

O esquema apresentado acima foi montado com base na seguinte ideia: como A e B são as chaves seletoras, precisamos com que as entradas de dados, que terão seus dados selecionados, apresentem os resultados corretos em suas respectivas saídas. Portanto, para a saída T, temos:

Entradas			Saídas	
A 1º bit	B 2º bit	C Vem Um	T Vai Um	S Soma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 5. Entradas de dados para a saída T

Vemos que quando AB selecionar a entrada  $Y_0$ , ou seja, quando  $AB = 00$ , que T possui o valor 0 independente da entrada C. Portanto, temos que a entrada de dados  $Y_0$  precisa apresentar esse valor para que quando AB for igual a 00 a saída apresente o valor correto de T. O mesmo ocorre para o caso em que  $AB = 11$ , porém, ao invés de T possuir o valor 0, ele possui o valor 1. Assim, conectando  $Y_0$  na terra e  $Y_1$  na fonte, conseguimos estes valores. Já para os casos em que  $AB = 01$  e  $AB = 10$ , vemos que a saída T tem que ser igual ao valor de C. Portanto, quando AB selecionar a entrada de dados  $Y_1$  ou  $Y_2$ , a saída tem que possuir o valor de C. Portanto, conectamos C à essas entradas de dados. Já para o caso da saída S temos a seguinte análise:

Entradas			Saídas	
A 1º bit	B 2º bit	C Vem Um	T Vai Um	S Soma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Figure 6. Entradas de dados para a saída S

Quando  $AB = 00$  e  $AB = 11$  temos que a saída S é a mesma que o valor da variável S. Portanto, ao selecionar as entradas de dados  $Z_0$  e  $Z_3$  precisamos que elas apresentem o valor de C, explicando a conexão feita. Quando  $AB = 01$  e  $AB = 10$ , ocorre o oposto: a saída C possui o valor de NOT C. Portanto, temos que as entradas  $Z_1$  e  $Z_2$  precisam conter esse valor. Feito o esquema apresentado, o circuito foi montado e implementado com a ajuda dos materiais apresentados acima:

## 5. Análise dos Resultados

Faça uma análise crítica dos resultados obtidos nos experimentos. Esta análise pode ser feita item a item ou de uma forma geral.

Dica: Use pesquisa na Internet para tirar as dúvidas sobre edição em  $\text{\LaTeX}$ .

## **6. Conclusão**

A realização do experimento resultou num excelente aprendizado a respeito dos multiplexadores e demultiplexadores. Na experiência, os multiplexadores são tomados como circuitos combinacionais e a relação deles com os demultiplexadores também foi estudada. Foi possível concluir que um multiplexador é uma ótima ferramenta para gerar funções booleanas. A simplificação de funções booleanas, como sempre, ajuda a simplificar o experimento, realçando novamente sua importância. A complexidade do experimento e o curto espaço de tempo que se tem para realiza-lo não permite atrasos de nenhum tipo e a simplificação das funções é de extrema ajuda nesse sentido

## Auto-Avaliação

1. C
2. C
3. E
4. C
5. E
6. C
7. C
8. C
9. E
10. E
11. C

**Table 1. Expected values of the obtained circuits' attributes.**

Experiment	Phase 1			Phase 2			$t(s)$
	$n_g$	$n_l$	$n_t$	$n_g$	$n_l$	$n_t$	
1 bit full adder	8.16	3.8	47.6	5.03	3.0	25.93	99.13
2 bit full adder	18.06	5.16	107.13	11.06	4.9	60.06	709.56
2 bit multiplier	14.2	4.03	74.33	7.7	2.2	37.53	357.76
7 segment decoder	47.53	5.83	270.46	32.86	5.0	176.4	740.63
Karnaugh 1 bit full adder	19.0	6.0	102.0	5.03	3.0	24.8	130.73