

## Laboratório – Projeto, Prototipação e Validação de Circuitos Lógicos Combinacionais

Autor: Prof. Dr. Ney Laert Vilar Calazans, Engenharia de Computação, EPoli (PUCRS) <sup>1</sup>

### 1 Objetivos

Ao final deste Laboratório os alunos devem haver entendido, treinado e dominado o emprego de circuitos integrados (CIs) da família TTL para montar e validar projetos de circuitos lógicos. Os alunos devem estar capacitados a: (1) parcialmente entender o processo de partir de uma especificação de um problema que pode ser resolvido por um circuito lógico e propor o projeto de um circuito que mostra a resolução do problema; (2) reconhecer componentes TTL, incluindo identificar sua função a partir do código do componente e reconhecer suas características de pinagem e uso correto dos mesmos em termos de alimentação, entradas e saídas; (3) construir circuitos lógicos combinacionais simples com portas lógicas disponibilizadas sob a forma de CIs TTL.

### 2 Exercício 1

Para treinar os postulados da álgebra Booleana, a representação dos circuitos digitais, as equivalências lógicas e a análise do funcionamento dos circuitos digitais, esta Seção apresenta a proposta de um exercício prático. Para o circuito apresentado na Figura 1, pede-se:

- Obtenha a tabela verdade que descreve o circuito.
- Represente o comportamento deste circuitos usando um diagrama de portas lógicas usando apenas portas NAND. Para tanto, transforme o circuito usando as equivalências Booleanas que podem ser encontradas em vários textos, tais como por exemplo na página a seguir: [https://www.electronicstutorials.ws/boolean/bool\\_6.html](https://www.electronicstutorials.ws/boolean/bool_6.html).
- Monte o circuito em protoboard e teste seu funcionamento (use a representação de sua escolha, igual à da Figura 1, só que composta usando apenas portas NAND).

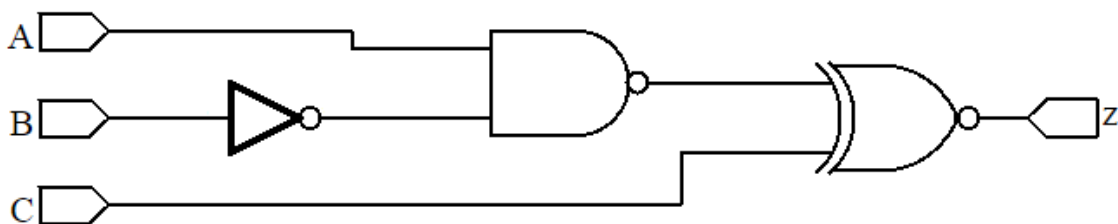


Figura 1 – Um exemplo de circuito lógico que descreve uma função Booleana de 3 entradas Booleanas.

Para o circuito acima, a equação Booleana equivalente a ele é dada por:  $Z = \overline{\overline{A}B} \oplus C$ .

Lembrando, o símbolo + com um círculo em volta é a representação da função Booleana OU-exclusivo; a barra sobre uma letra ou uma expressão representa inversão etc.

A tabela verdade incompleta que descreve o funcionamento do circuito lógico é dada abaixo, termine de preenchê-la corretamente. Note que as colunas denominadas A, B e C (mais à esquerda na tabela) correspondem às entradas, a coluna mais à direita é a saída e a coluna do meio é a saída da porta NAND.

<sup>1</sup> Baseado em material original disponibilizado pelo Prof. Júlio César Marques de Lima, da Escola Politécnica - PUCRS

A B C	$\overline{A\overline{B}}$	$z = \overline{\overline{A\overline{B}}} \oplus C$
0 0 0	1	
0 0 1		
0 1 0		
0 1 1		
1 0 0		
1 0 1		
1 1 0		
1 1 1		1

Depois de fazer a tabela, desenhe o diagrama de portas lógicas equivalente ao da Figura 1, mas que usa apenas portas lógicas NAND (note que isto sempre é possível).

Já para a montagem em protoboard serão necessários fios para a montagem, além de uma Fonte de Alimentação simples, um Multímetro e um protoboard, dois integrados 7400 (Para descobrir os *part numbers* a partir da função levantada experimentalmente, use documentação sobre portas TTL, tal como disponível, por exemplo na página de Internet [https://pt.wikipedia.org/wiki/Lista\\_dos\\_circuitos\\_integrados\\_da\\_s%C3%A9rie\\_7400](https://pt.wikipedia.org/wiki/Lista_dos_circuitos_integrados_da_s%C3%A9rie_7400).), um resistor (sugere-se que este seja de 390 Ohms) e um LED. Uma sugestão de montagem está representada na Figura 2. Os fios indicados com 5V e 0V são conectados à Fonte de Alimentação. Os fios indicados com as letras A, B e C são as entradas do circuito. Conecte-os nas trilhas indicadas por “1” ou com “0” (via setas pretas à direita da montagem), ou use um conjunto de 4 dip-switches adequadamente montado e observe o resultado na saída do circuito, o LED). A condição da saída em “1” significa LED<sup>2</sup> aceso e “0” significa LED apagado. Gere todas as combinações de 1s e 0s da tabela verdade para as entradas A, B e C.

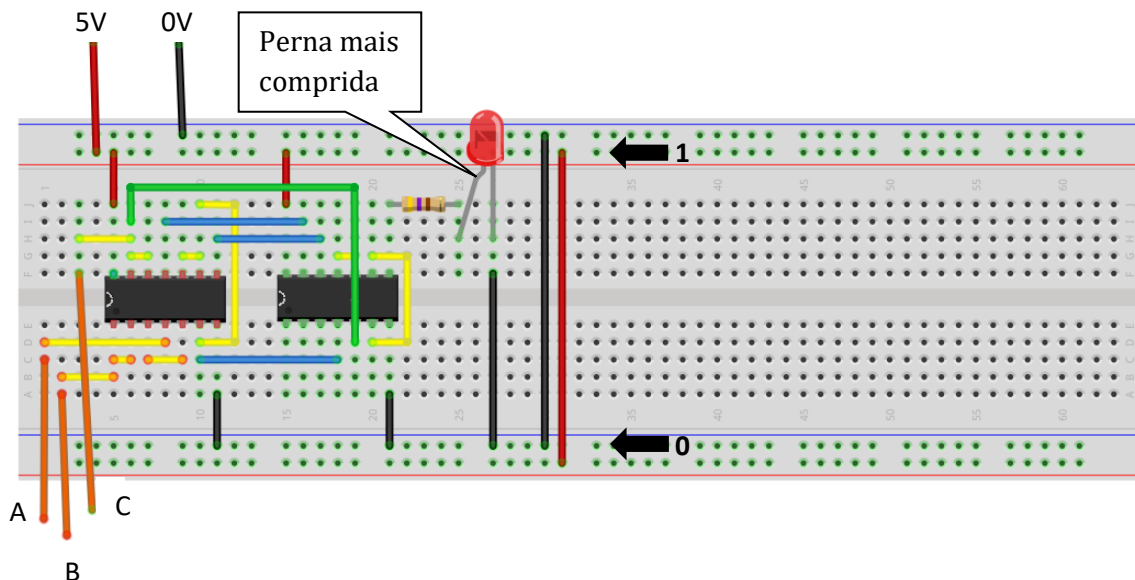


Figura 2 – Exemplo de aparência que terá a montagem do circuito no protoboard.

<sup>2</sup> Notem que um LED é um dispositivo com polaridade, então é essencial conectá-lo considerando o fluxo de corrente que por ele passa, veja a observação na Figura 2.

### 3 Exercício 2

Mostradores ou *displays* de 7 segmentos são utilizados na construção de painéis dos mais variados equipamentos de uso industrial, comercial e residencial. Há vários tipos de *displays* de 7 segmentos, classificados quanto ao tipo de dispositivo de visualização (LCD, LED, Fluorescente, etc.). Os *displays* mais comuns são os do tipo LED, pois não necessitam de nenhum circuito especial de acionamento, apenas um resistor para limitar a corrente em cada segmento. Os *displays* de 7 segmentos do tipo LED podem ainda ser do tipo ANODO COMUM, onde cada segmento é ativado com nível lógico "0" ou CATODO COMUM, onde cada segmento é ativado com nível lógico "1".

*Displays* de 7 segmentos podem ser encontrados em diversos tamanhos, cores e número de dígitos. O aspecto físico dos *displays* de 7 segmentos é exemplificado na Figura 3 dada a seguir, onde está feita a indicação do nome de cada segmento e sua localização.



Figura 3 – Exemplos de mostradores (em inglês *displays*) de 7 segmentos e sua nomenclatura.

O Laboratório dispõe apenas de *displays* de 7 segmentos do tipo ANODO COMUM, de 1 dígito. Para este, pede-se:

- Escreva a tabela verdade de um circuito decodificador de um número binário de 4 bits para acionar um mostrador de 7 segmentos capaz de mostrar os dígitos de 0 a 9. Use a tabela abaixo e a figura que a segue como guia, lembrando que se utilizarão mostradores de 7 segmentos do tipo ANODO comum (LED acende se entrada for 0). Note que a saída ZA corresponde ao segmento A (segmento horizontal no topo do mostrador), a saída ZB corresponde ao segmento B (segmento vertical, à direita na metade de cima do mostrador), e assim por diante.

A	B	C	D	ZA	ZB	ZC	ZD	ZE	ZF	ZG
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							



- Escreva as equações Booleanas para cada segmento do mostrador de 7 segmentos.
- Simplifique as equações e racionalize a representação de termos semelhantes.

- 4) Represente o diagrama elétrico com portas lógicas.
- 5) Monte o circuito e teste o seu funcionamento. Não esqueça de calcular o resistor de limitação de corrente para cada segmento.

## 4 Descrições Textuais e Exercício 3

Circuitos lógicos podem ser construídos a partir de descrições textuais, se estas utilizarem expressões como SE, ENTÃO, SENÃO, E, OU, NÃO, QUANDO, ENQUANTO, que permitam descrever com precisão o funcionamento de um determinado circuito lógico. Com o emprego da álgebra Booleana ou de uma tabela verdade, pode-se traduzir esta descrição para uma linguagem Booleana e por fim para o desenho do circuito elétrico equivalente.

### 4.1 Primeiro Exemplo – Caso de um alarme de regime perigoso de motor

Um motor a combustão tem alarme de **regime perigoso** quando o nível de óleo (N) está baixo e a temperatura do motor (T1) está acima de 85°C ou a temperatura do motor (T2) está acima de 95°C e a rotação do motor (R) está acima de 4000 RPM.

N	Óleo
0	baixo
1	Ok

T1T2	Temperaturas
00	Normal
10	Acima de 85
11	Acima de 95

R	Rotação
0	< 4000
1	> 4000

Obs.: A combinação lógica T1T2 = 01 não existe na descrição. Porém, ela pode existir na tabela verdade e pode ser empregada como condição de inespecificação ou em inglês um *don't care* (X). Através desta descrição, a equação do alarme pode ser descrita (já na forma simplificada) como:

$$\text{Alarme} = \bar{N}T1\bar{T}2 + T1T2R$$

Já a sua tabela verdade poderia ser descrita por:

N	T1	T2	R	Alarme
0	0	0	0	0
0	0	0	1	0
0	0	1	0	X
0	0	1	1	X
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	X
1	0	1	1	X
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Note que as condições X da tabela verdade marcam as condições que não podem ocorrer fisicamente no sensor de temperatura. A equação não simplificada seria a dada abaixo. Note o ganho de usar os *don't-cares* para otimizar a lógica, pois quanto mais letras e mais termos, maior o hardware que implementa uma função:

$$\text{Alarme} = \bar{N} T1\bar{T}2 \bar{R} + \bar{N} T1\bar{T}2 R + \bar{N} T1T2 \bar{R} + \bar{N} T1T2 R + N T1T2 R$$

## 4.2 Segundo Exemplo – Caso da transfusão de sangue

No início do século XX, Karl Landsteiner descobriu os grupos sanguíneos (A, B, AB e O) e desta forma conseguiu explicar e determinar quando e porque uma transfusão sanguínea entre seres humanos é segura. A compatibilidade entre os grupos sanguíneos pode ser explicada por meio de uma tabela que relaciona as entradas (doadores) e as saídas (receptores). As posições onde a tabela indica um “+” significam que existe compatibilidade e as posições onde aparecem um “-” significam que não existe compatibilidade e a transfusão não pode ser feita.

	A	B	AB	O
A	+	-	+	-
B	-	+	+	-
AB	-	-	+	-
O	+	+	+	+

A coluna mais à esquerda indica os grupos doadores possíveis e a linha superior representa os grupos receptores.

### 4.2.1 Exercício – Transfusão de Sangue

Projete um “Painel Eletrônico” capaz de receber como entrada um tipo de doador de sangue disponível e que indica quais receptores são compatíveis com este doador. O painel conta com duas entradas (X e Y), que geram todas as combinações possíveis de doadores (A, B, AB e O) e quatro saídas que indicam todos os possíveis receptores. As saídas irão acionar um LED cada. Não esqueça que cada saída precisa de um resistor limitador de corrente. As entradas também precisam deste resistor, para limitar a corrente que circula nas chaves.

Monte um circuito que mostre o painel funcionando.

## 4.3 Para ir mais Longe

Como muitos devem saber, a compatibilidade sanguínea não depende apenas dos tipos sanguíneos mencionados na Seção 4.2. Ela também depende do chamado fator Rh. Leia sobre este fator, por exemplo, na página <https://www.significados.com.br/fator-rh/>. A partir daí, como o painel eletrônico proposto como exercício poderia ser refinado para considerar este fator na geração da informação de compatibilidade sanguínea?

**Bom trabalho!**