

Circuitos Lógicos

Aula 8

cruz@gta.ufrj.br <http://gta.ufrj.br/~cruz>

Na última aula

- Mapa de Karnaugh
- Codificadores



Hoje

- Método de projeto
- Universalidade NAND
- High Z
- Don't care



Hoje

- Método de projeto
- Universalidade NAND
- High Z (alta impedância)
- Don't care



Hoje

- Método de projeto
- Universalidade NAND
- High Z (alta impedância)
- Don't care (tanto faz)



Método de projeto



Engenharia

- Problema
 - Deve ser compreendido
- Projeto
 - Deve resolver o problema
- Implementação
 - Deve refletir o projeto
- Teste
 - Deve garantir que o problema está resolvido



Projeto



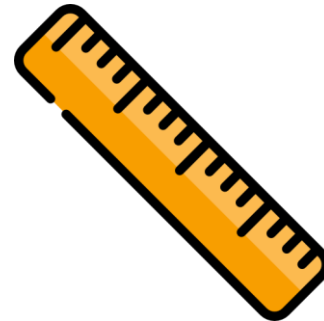
Implementação



Teste

Objetivo: resolver o problema

- Atender aos requisitos do problema
 - Funcionais
 - Lógica booleana
 - Não-funcionais
 - Custo
 - Eficiência energética
 - Tempo de resposta
 - Tamanho
 - ...
- Reduzir o número de portas lógicas



Objetivo: resolver o problema

- Atender aos requisitos do problema

- Explícitos

- Im

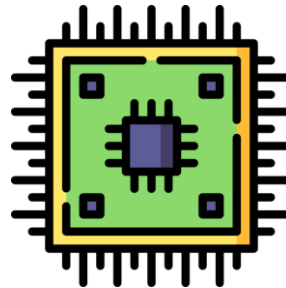
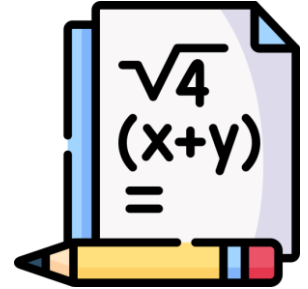
Método garante o menor circuito possível
(atendendo aos requisitos implícitos)

- Reduzir o número de portas lógicas



Método de projeto

- Compreender problema
 - Listar entradas e saídas
 - Gerar tabela verdade
 - Relacionar entradas e saídas
- Obter expressão lógica para cada saída
 - Simplificar expressões lógicas
- Desenhar circuito



Universalidade NAND



NAND é demais

- É possível construir as portas básicas utilizando apenas portas NAND
 - NOT
 - OR
 - AND
 - XOR



Porta NOT

- Faz a função NOT de uma variável

$$\overline{A \cdot 1} = \bar{A}$$

$$\overline{A \cdot A} = \bar{A}$$

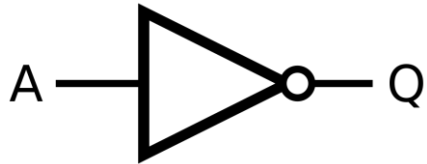
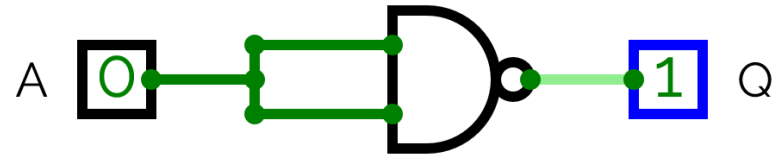
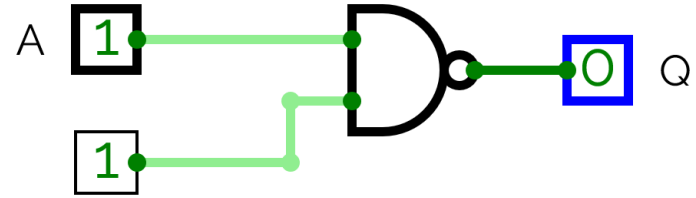


Imagem retirada da Wikipedia



Porta AND

- Faz a função AND entre duas ou mais variáveis

$$A \cdot B = \overline{\overline{A \cdot B}} = \overline{\overline{A \cdot B} \cdot \overline{A \cdot B}}$$

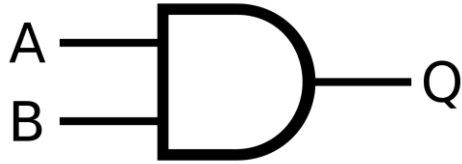
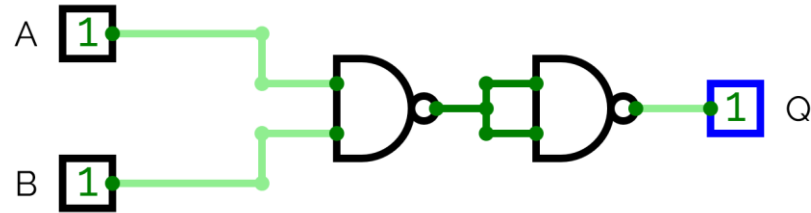


Imagem retirada da Wikipedia



Porta OR

- Faz a função OR entre duas ou mais variáveis

$$A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}}$$

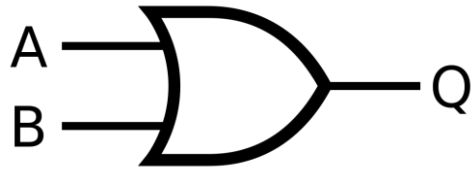
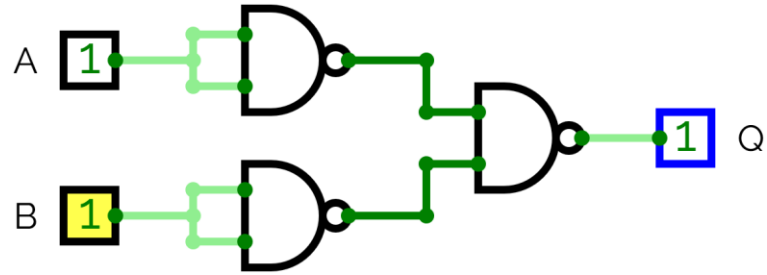


Imagem retirada da Wikipedia



Porta XOR

- Faz a função XOR entre duas ou mais variáveis

$$\begin{aligned}A \oplus B &= (A + B) \cdot (\overline{A \cdot B}) \\&= (A \cdot \overline{A \cdot B}) + (B \cdot \overline{A \cdot B}) \\&= \overline{\overline{A \cdot \overline{A \cdot B}} + \overline{B \cdot \overline{A \cdot B}}} \\&= \overline{\overline{A \cdot \overline{A \cdot B}} \cdot \overline{B \cdot \overline{A \cdot B}}}\end{aligned}$$

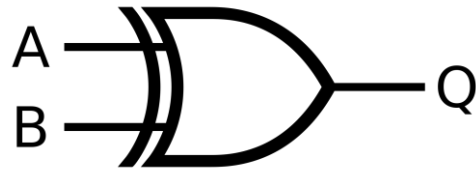
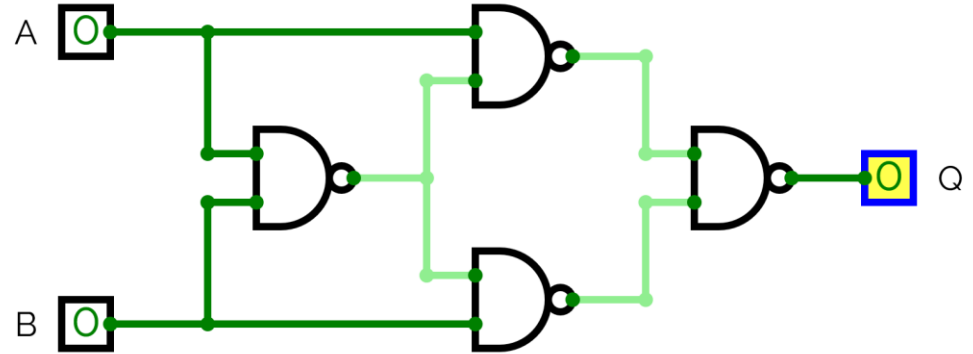


Imagem retirada da Wikipedia



E o NOR?

- Podemos construir qualquer porta usando NOR
 - Temos universalidade NOR



E o NOR?

- Podemos construir qualquer porta usando NOR
 - Temos universalidade NOR

Oba, mais uma lista de exercícios!



Terceiro estado lógico: High-Z



Impedância

- Impedância é a capacidade de resistir à corrente elétrica
- Resistência ôhmica
 - Material resiste “segura” elétrons e impede sua passagem
- Impedância capacitiva
 - Campo elétrico impede passagem de corrente
- Impedância indutiva
 - Campo magnético impede passagem de corrente



Impedância

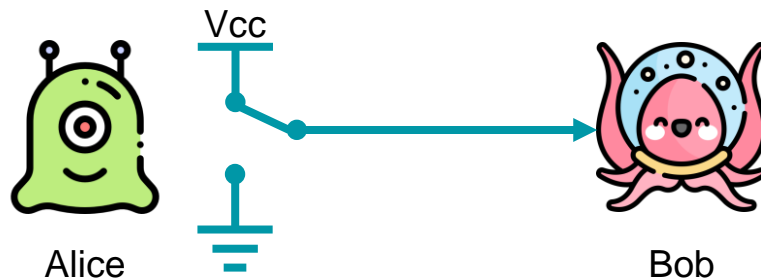
- Impedância é a capacidade de resistir à corrente elétrica
- Resistência ôhmica
 - Material resiste “segura” elétrons e impede sua passagem
- Impedância capacitiva
 - Campo elétrico impede passagem de corrente
- Impedância indutiva
 - Campo magnético impede passagem de corrente

Se a impedância é alta o suficiente, a corrente se aproxima de zero
É como se um interruptor estivesse aberto
(pode ser, de fato, um interruptor aberto)



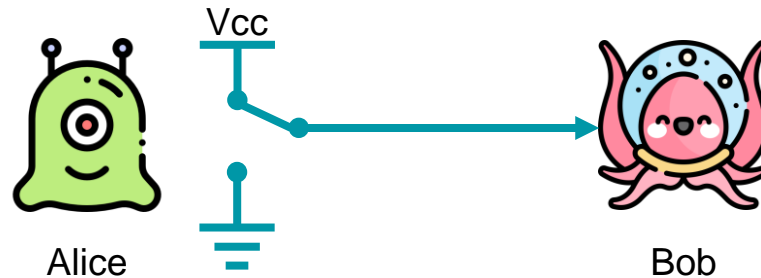
Comunicação digital

- Alice quer falar com Bob
- Canal é estabelecido
 - “Fio” é colocado entre eles
- Dados transmitidos representam dados gerados por Alice
 - 5v representa 1
 - 0v representa 0
- Como representar o ‘silêncio’?



Representando o 'silêncio'

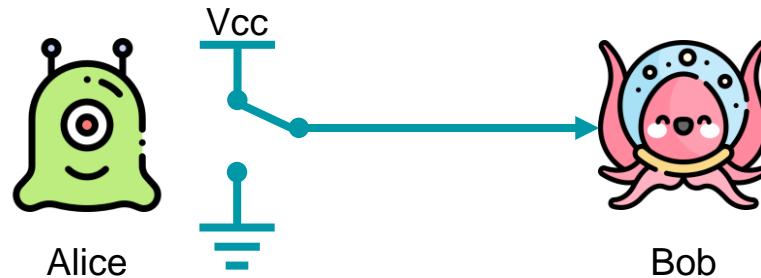
- Dados transmitidos representam dados gerados por Alice
 - 5v representa 1
 - Alice gera corrente na direção de Bob
 - 0v representa 0
 - Alice recebe corrente vinda de Bob
 - Silêncio é representado por alta impedância
 - Alice não recebe nem gera corrente



Representando o 'silêncio'

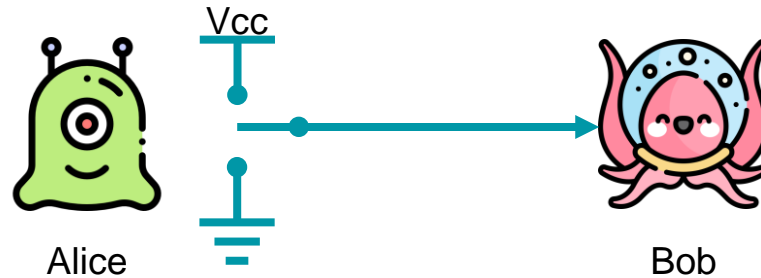
- Dados transmitidos representam dados gerados por Alice
 - 5v representa 1
 - Alice gera corrente na direção de Bob
 - 0v representa 0
 - Alice recebe corrente vinda de Bob
 - Silêncio é representado por alta impedância
 - Alice não recebe nem gera corrente

Há muitas outras formas de representar o silêncio!!!



Alta impedância

- Alice se desconecta de Bob
 - Bob não recebe nem envia corrente
 - Bob não sabe se é 0 ou 1
 - Bob sabe que é silêncio



3º estado lógico: high-z

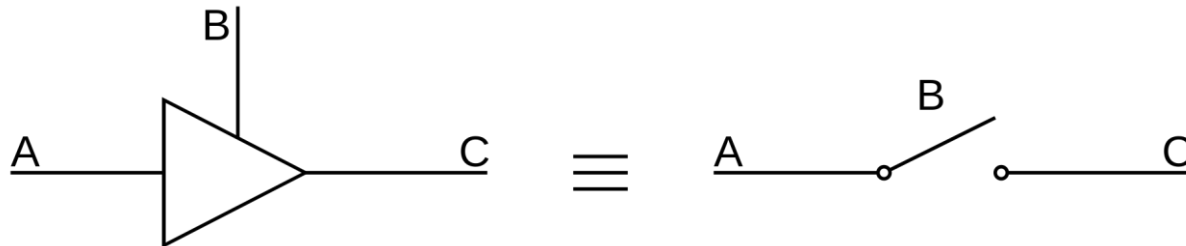
- Estados lógicos bem definidos
 - 1
 - 5v com relação a um referencial comum
 - 0
 - 0v com relação a um referencial comum
- Estado lógico representado desconexão
 - High-z



Tri-state buffer

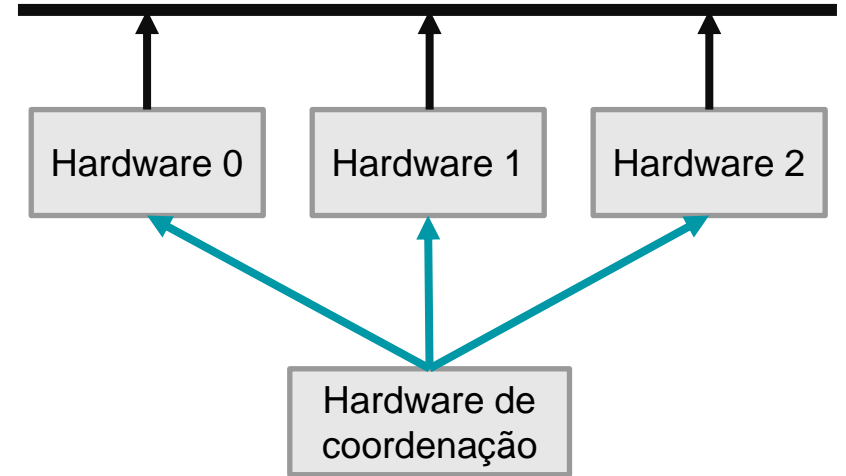
- Duas entradas
 - A
 - Controla dois estados lógicos da saída
 - 1
 - 0
 - B
 - Controla impedância da saída (*enable*)
 - High-z

B	A	C
0	0	High-Z
0	1	High-Z
1	0	0
1	1	1



Output Enable

- Várias peças de hardware estão conectadas ao mesmo barramento
 - Só uma pode estar “falando” ao mesmo tempo
- Saída de cada peça tem um tri-state buffer
- *Output Enable* ativa saída
 - \overline{OE}^*
- Saídas de hardwares diferentes nunca estão ativas ao mesmo tempo

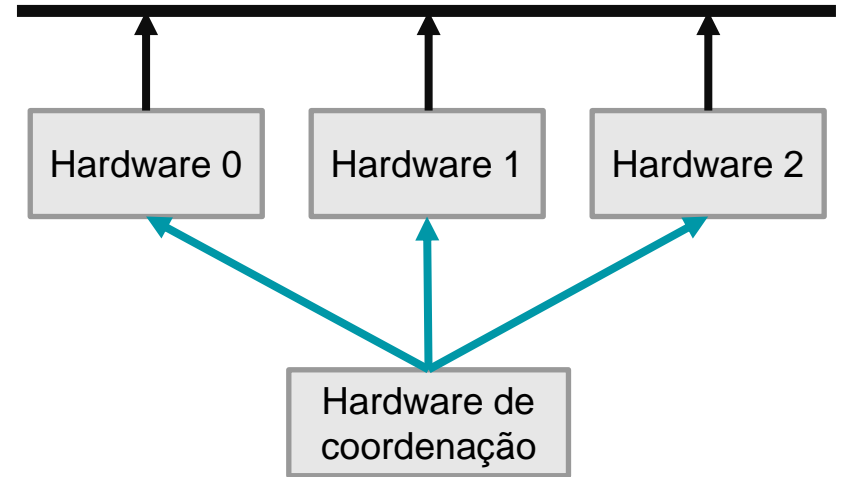


* Negação indica que o OE é ativo em nível lógico baixo



Chip Select

- Várias peças de hardware estão conectadas ao mesmo barramento
 - Só uma pode estar “falando” ao mesmo tempo
- Cada hardware tem um tri-state buffer **que desativa o hardware**
- *Chip Select* ativa o hardware como um todo
 - \overline{CS}^*
- Hardwares diferentes nunca estão ativos ao mesmo tempo



* Negação indica que o CS é ativo em nível lógico baixo



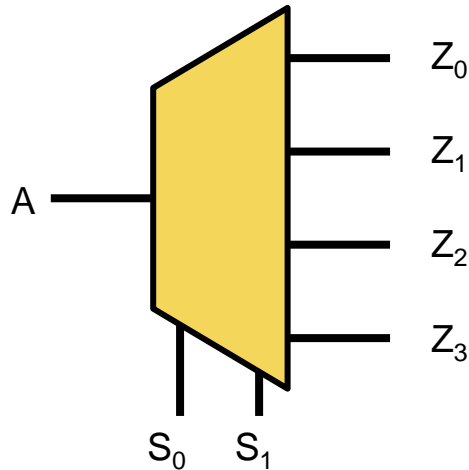
Memento flashback



Demultiplexador (DEMUX)



- Dois conjuntos de entradas
 - n seletores
 - 1 linhas de dados
- 2^n saídas
- Seletor “chaveia” a linha de dados entre diferentes saídas



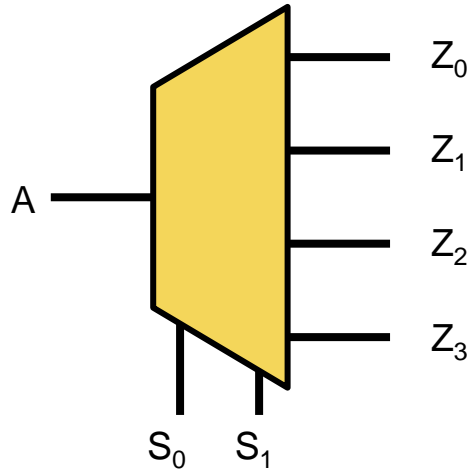
S_1	S_0	A	Z_3	Z_2	Z_1	Z_0
0	0	0				0
0	0	1				1
0	1	0			0	
0	1	1			1	
1	0	0		0		
1	0	1		1		
1	1	0	0			
1	1	1	1			

Se a gente lembrar



- $x \cdot 1 = x$
 - AND com 1 “deixa passar”
- $x + 0 = x$
 - OR com 0 “deixa passar”

$$\begin{aligned}Z_0 &= (\overline{S_1} \cdot \overline{S_0} \cdot A) \\Z_1 &= (\overline{S_1} \cdot S_0 \cdot A) \\Z_2 &= (S_1 \cdot \overline{S_0} \cdot A) \\Z_3 &= (S_1 \cdot S_0 \cdot A)\end{aligned}$$

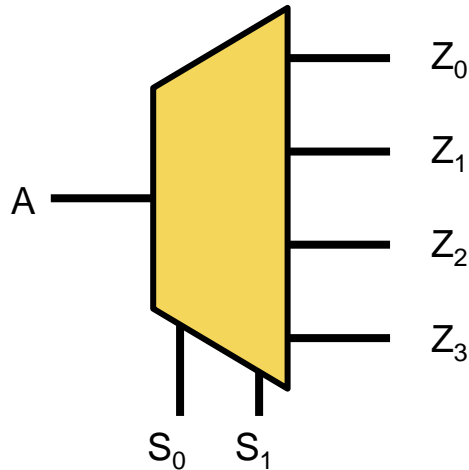


Problema!



- Saídas recebem 0 quando deveriam estar inativas!!!
- Como resolver?

Saberemos em aulas futuras



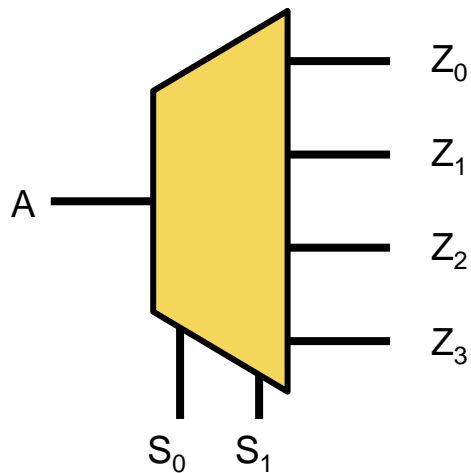
S ₁	S ₀	A	Z ₃	Z ₂	Z ₁	Z ₀
0	0	0				0
0	0	1				1
0	1	0			0	
0	1	1			1	
1	0	0		0		
1	0	1		1		
1	1	0	0			
1	1	1	1			

O futuro é agora!



O futuro é agora!

- Saídas recebem 0 quando deveriam estar inativas!!!
- Como resolver?
 - High-Z



S_1	S_0	A	Z_3	Z_2	Z_1	Z_0
0	0	0	High-Z	High-Z	High-Z	0
0	0	1	High-Z	High-Z	High-Z	1
0	1	0	High-Z	High-Z	0	High-Z
0	1	1	High-Z	High-Z	1	High-Z
1	0	0	High-Z	0	High-Z	High-Z
1	0	1	High-Z	1	High-Z	High-Z
1	1	0	0	High-Z	High-Z	High-Z
1	1	1	1	High-Z	High-Z	High-Z



Don't care



Outro flashback



Outro flashback

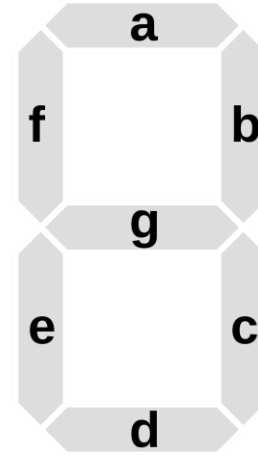


Imagem retirada do clipe de Don't Care – Ed Sheeran e Justin Bieber

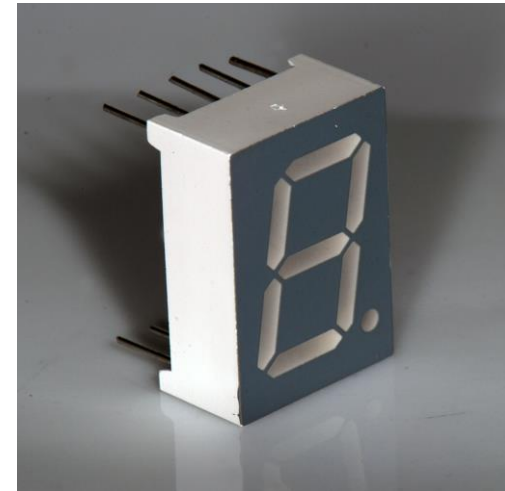


Display 7 segmentos

- Leds organizados em 8
- Leds nomeados a, b, c, d, e, f, g
- GND comum
- Terminal para cada um dos leds



Imagens retiradas da Wikipedia



Codificador BCD – 7 segmentos

- Recebe 4 variáveis
 - Representam um número de 4 bits
- Codifica o número para exibição em um display de 7 segmentos

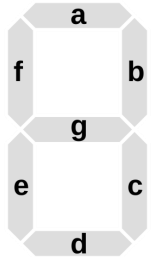
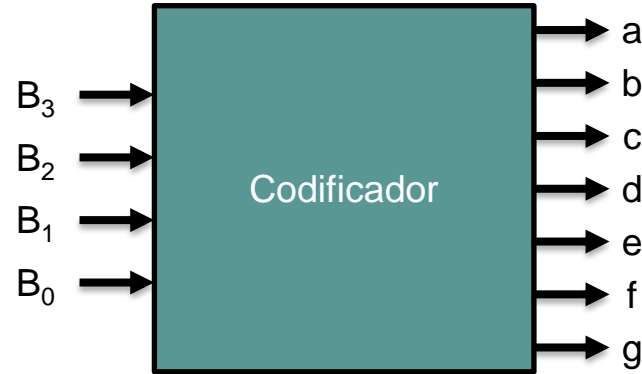
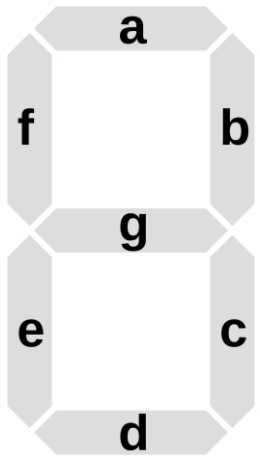


Tabela verdade

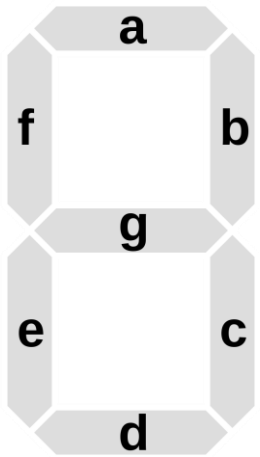


Imagens retiradas da Wikipedia

B ₃	B ₂	B ₁	B ₀	a	b	c	d	e	f	g	Dec.	Display
0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	0	1	1	0	0	0	0	1	1
0	0	1	0	1	1	0	1	1	0	1	2	2
0	0	1	1	1	1	1	1	0	0	1	3	3
0	1	0	0	0	1	1	0	0	1	1	4	4
0	1	0	1	1	0	1	1	0	1	1	5	5
0	1	1	0	1	0	1	1	1	1	1	6	6
0	1	1	1	1	1	1	0	0	0	0	7	7
1	0	0	0	1	1	1	1	1	1	1	8	8
1	0	0	1	1	1	1	1	0	1	1	9	9
1	0	1	0	X	X	X	X	X	X	X	X	Não tem
...	Não tem



Tabela verdade



Imagens retiradas da Wikipedia

B ₃	B ₂	B ₁	B ₀	a	b	c	d	e	f	g	Dec.	Display
0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	0	1	1	0	0	0	0	1	1
0	0	1	0	1	1	0	1	1	0	1	2	2
0	0	1	1	1	1	1	1	0	0	1	3	3
0	1	0	0	0	1	1	0	0	1	1	4	4
0	1	0	1	1	0	1	1	0	1	1	5	5
0	1	1	0	1	0	1	1	1	1	1	6	6
0	1	1	1	1	1	1	0	0	0	0	7	7
1	0	0	0	1	1	1	1	1	1	1	8	8
1	0	0	1	1	1	1	1	0	1	1	9	9
1	0	1	0	X	X	X	X	X	X	X	X	Não tem
...	Não tem



Olhando a saída 'a'

B ₃	B ₂	B ₁	B ₀	a	Dec.	Display
0	0	0	0	1	0	0
0	0	0	1	0	1	1
0	0	1	0	1	2	2
0	0	1	1	1	3	3
0	1	0	0	0	4	4
0	1	0	1	1	5	5
0	1	1	0	1	6	6
0	1	1	1	1	7	7
1	0	0	0	1	8	8
1	0	0	1	1	9	9
1	0	1	0	X	X	X
...



Olhando a saída 'a'

	$\overline{B_0}$	B_0		$\overline{B_0}$	
$\overline{B_2}$	0	1	3	2	$\overline{B_3}$
B_2	4	5	7	6	
	$\overline{B_2}$	12	13	15	14
B_2		8	9	11	10
	$\overline{B_1}$		B_1		



B_3	B_2	B_1	B_0	a	Dec.	Display
0	0	0	0	1	0	0
0	0	0	1	0	1	1
0	0	1	0	1	2	2
0	0	1	1	1	3	3
0	1	0	0	0	4	4
0	1	0	1	1	5	5
0	1	1	0	1	6	6
0	1	1	1	1	7	7
1	0	0	0	1	8	8
1	0	0	1	1	9	9
1	0	1	0	X	X	X
...	X	X	X

Olhando a saída 'a'

	$\overline{B_0}$		B_0				$\overline{B_0}$		
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	
B_2	12	X	13	X	15	X	14	X	B_3
	$\overline{B_2}$	8	1	9	1	11	X	10	
	$\overline{B_1}$				B_1				



B_3	B_2	B_1	B_0	a	Dec.	Display
0	0	0	0	1	0	0
0	0	0	1	0	1	1
0	0	1	0	1	2	2
0	0	1	1	1	3	3
0	1	0	0	0	4	4
0	1	0	1	1	5	5
0	1	1	0	1	6	6
0	1	1	1	1	7	7
1	0	0	0	1	8	8
1	0	0	1	1	9	9
1	0	1	0	X	X	X
...	X	X	X

O Don't care

- Configurações 10, 11, 12, 13, 14 e 15 nunca serão alcançados
 - Variáveis de entrada não assumem configurações
- Saídas não importam
 - Se 0, tudo bem
 - Se 1, tudo bem

TANTO FAZI!

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
	12	X	13	X	15	X	14	X	
$\overline{B_2}$	8	1	9	1	11	X	10	X	B_3
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	
	12	X	13	X	15	X	14	X	B_3
$\overline{B_2}$	8	1	9	1	11	X	10	X	
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
$\overline{B_2}$	12	X	13	X	15	X	14	X	B_3
B_2	8	1	9	1	11	X	10	X	$\overline{B_3}$
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3$$

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
	12	X	13	X	15	X	14	X	
$\overline{B_2}$	8	1	9	1	11	X	10	X	B_3
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3$$

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
	12	X	13	X	15	X	14	X	
$\overline{B_2}$	8	1	9	1	11	X	10	X	B_3
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3 + B_1$$

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
	12	X	13	X	15	X	14	X	
$\overline{B_2}$	8	1	9	1	11	X	10	X	B_3
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3 + B_1$$

	$\overline{B_0}$		B_0		$\overline{B_0}$				
$\overline{B_2}$	0	1	1	0	3	1	2	1	$\overline{B_3}$
B_2	4	0	5	1	7	1	6	1	B_3
	12	X	13	X	15	X	14	X	
$\overline{B_2}$	8	1	9	1	11	X	10	X	B_3
	$\overline{B_1}$				B_1				



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3 + B_1 + (B_0B_2)$$

	$\overline{B_0}$	B_0	$\overline{B_0}$	
$\overline{B_2}$	0 1	1 0	3 1	2 1
B_2	4 0	5 1	7 1	6 1
$\overline{B_2}$	12 X	13 X	15 X	14 X
B_2	8 1	9 1	11 X	10 X
	$\overline{B_1}$	B_1		



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3 + B_1 + (B_0 B_2)$$

	$\overline{B_0}$	B_0	$\overline{B_0}$	
$\overline{B_2}$	0 1	1 0	3 1	2 1
B_2	4 0	5 1	7 1	6 1
	12 X	13 X	15 X	14 X
$\overline{B_2}$	8 1	9 1	11 X	10 X
	$\overline{B_1}$	B_1		



Otimizando com don't care

- Se tanto faz
 - Podemos escolher a configuração mais conveniente
- Procuramos qual valor mais conveniente no mapa de Karnaugh

$$a = B_3 + B_1 + (B_0 \cdot B_2) + (\overline{B_0} \cdot \overline{B_2})$$

	$\overline{B_0}$	B_0	$\overline{B_0}$	
$\overline{B_2}$	0 1	1 0	3 1	2 1
B_2	4 0	5 1	7 1	6 1
$\overline{B_2}$	8 1	9 1	11 X	10 X
	$\overline{B_1}$	B_1		

The Karnaugh map shows a 4x4 grid of cells. The columns are labeled $\overline{B_0}$, B_0 , $\overline{B_0}$ and the rows are labeled $\overline{B_2}$, B_2 , $\overline{B_2}$. The right side of the map is labeled $\overline{B_3}$ for the top two rows and B_3 for the bottom two rows. The cells contain values: 0, 1, 1, 0, 3, 1, 2, 1 in the first row; 4, 0, 5, 1, 7, 1, 6, 1 in the second row; 12, X, 13, X, 15, X, 14, X in the third row; 8, 1, 9, 1, 11, X, 10, X in the fourth row. The cells with 'X' are don't care values. The map is annotated with colored boxes: orange boxes around (0,1), (2,1), (8,1), and (10,X); a blue box around (5,1), (7,1), (13,X), and (15,X); and a yellow box around (3,1), (7,1), (11,X), and (15,X).



Conclusões

- Porta NAND e porta NOR são universais
 - Podem construir as outras portas
- 3º estado lógico pode resolver determinados problemas
 - “Não há informação”
 - “Não posso falar agora”
- *Don't care* é uma ferramenta importante para otimização
 - Necessário identificar onde usar



Créditos

Os ícones desta apresentação foram feitos por Freepic e retirados de www.flaticon.com.





GTA / UFRJ

GRUPO DE TELEINFORMÁTICA E AUTOMAÇÃO

www.gta.ufrj.br