Introdução ao projeto de sistemas digitais

Prof. Ilan Sousa Correa

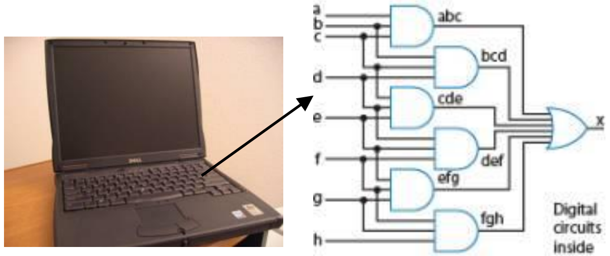
Universidade Federal do Pará (UFPA)

Instituto de Tecnologia (ITEC)

Faculdade de Eng. da Computação e Telecomunicações (FCT)

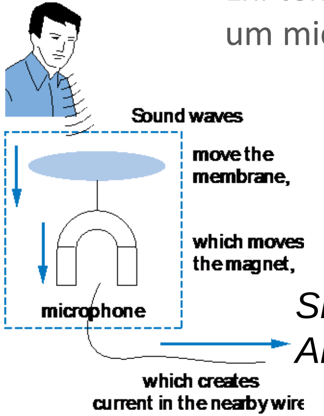
Por que estudar projeto de sistemas digitais?

- Blocos básicos da construção de computadores
  - Mesmo programadores de alto nível precisam conhecer o hardware em algum nível
- Dispositivos eletrônicos estão se tornando digitais
  - Chips menores e mais poderosos
  - Resultado:
    - Dispositivos melhores: Melhores gravadores de som, câmeras, carros, smartphones, equipamentos médicos,...
    - Novos dispositivos: Vídeo games, smartphones, ...
  - Podem ser chamados genericamente de sistemas embarcados



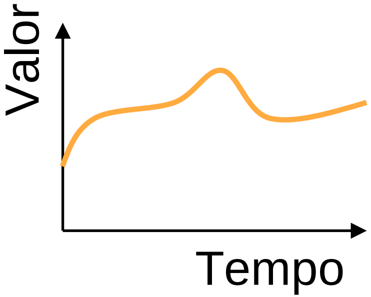
Sistemas digitais e sinais digitais

- Sinais analógicos
  - Infinitas possibilidades de valores
  - Ex: tensão em um condutor gerada por um microfone

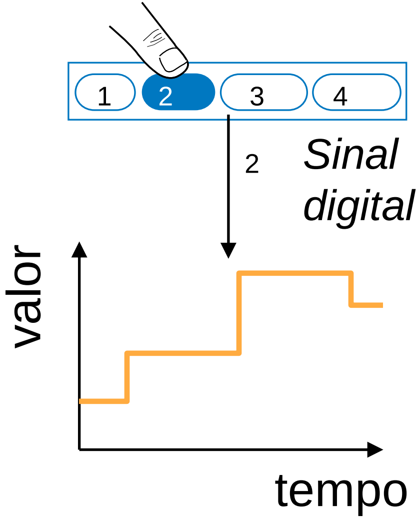


Sinal Analógico

Valores possíveis:
1.00, 1.01, 2.0000009,
... Infinitas possibilidades



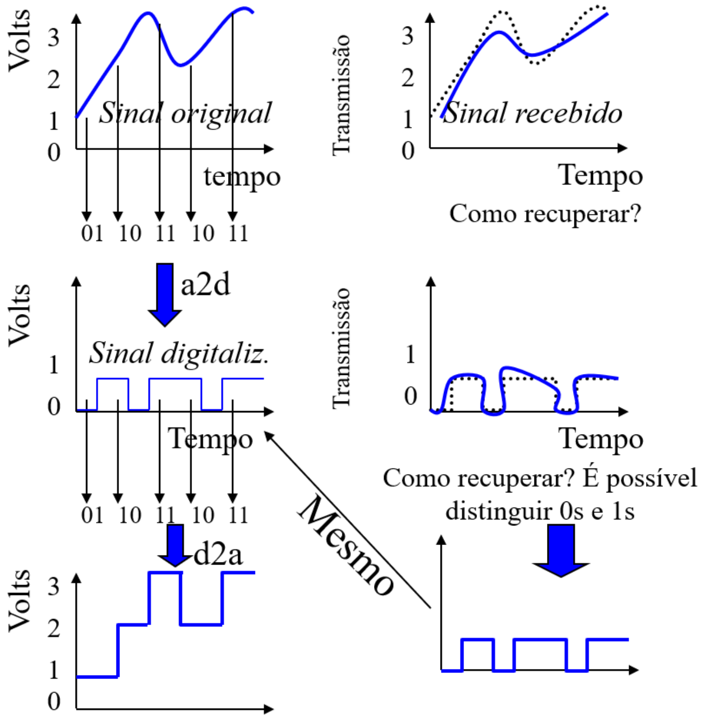
- Sinais digitais
  - Possibilidades finitas de valores
  - Ex: botão pressionado em um teclado



Únicos valores possíveis:
0, 1, 2, 3, ou 4.

Benefícios da utilização de sinais digitais

- Sinais analógicos podem perder qualidade
    - Valores de tensão não são salvos/copiados/transmitidos perfeitamente
- Versão digitalizada permite salvar/copiar/ transmitir quase perfeitamente
    - Amostra da tensão a uma determinada taxa, salva a amostra em uma determinada codificação
    - Há erro a amostragem
    - Mas é possível distinguir entre 0s e 1s.



Benefícios da utilização de sinais digitais

- Sinais digitais como áudio podem ser comprimidos
  - Ex: MP3
  - Um CD pode armazenar:
    - Aproximadamente 20 músicas não comprimidas
    - Aproximadamente 200 músicas comprimidas
  - A compressão também se aplica a imagens (JPEG), vídeos (MPEG), e outros
  - Há ainda outros benefícios

Exemplo de um esquema de Compressão

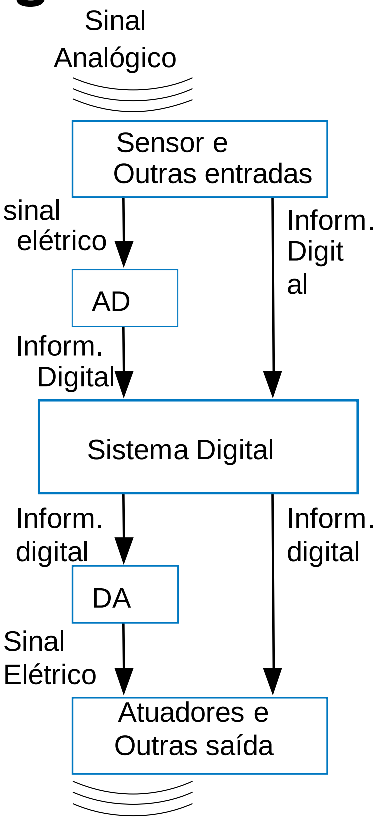
00 --> 0000000000
01 --> 1111111111
1X --> X

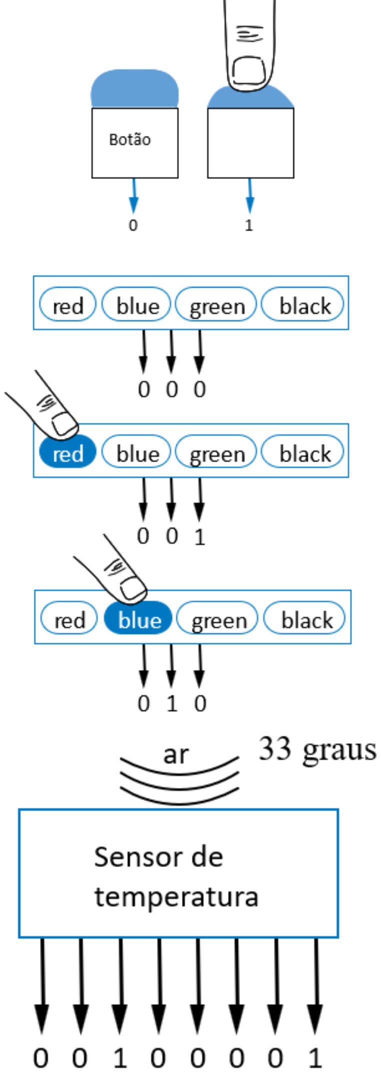
0000000000 0000000000 0000001111 1111111111

00 00 10000001111 01

Codificação da informação digital

- Alguns tipos de entradas de dados são inerentemente digitais
  - Botão: pressionado (1), não pressionado (0);
- Outros tipos de entradas podem são digitais
  - Necessitam de uma codificação
  - Vários botões: vermelho (000), azul (010)
- Sinais analógicos
  - Conversão AD
  - Codificação





Exemplos de codificação de informação: ASCII e UNICODE

- ASCII
  - American Standard Code for Information Interchange
  - Codificação de 7 ou 8 bits para cada letra, número ou símbolo
- Unicode
  - Codificação com 16 bits para cada letra, número ou símbolo
  - Inclui suporte para caracteres de vários alfabetos diferentes

Símbolo	Codificação
R	1010010
S	1010011
T	1010100
L	1001100
N	1001110
E	1000101
0	0110000
.	0101110
<tab>	0001001

Símbolo	Codificação
r	1110010
s	1110011
t	1110100
l	1101100
n	1101110
e	1100101
9	0111001
!	0100001
<space>	0100000

Exemplo:
Representação binária de uma palavra
1001100 0110000 1010100 1000101

LOTE

Exemplos de codificação de informação: números binários

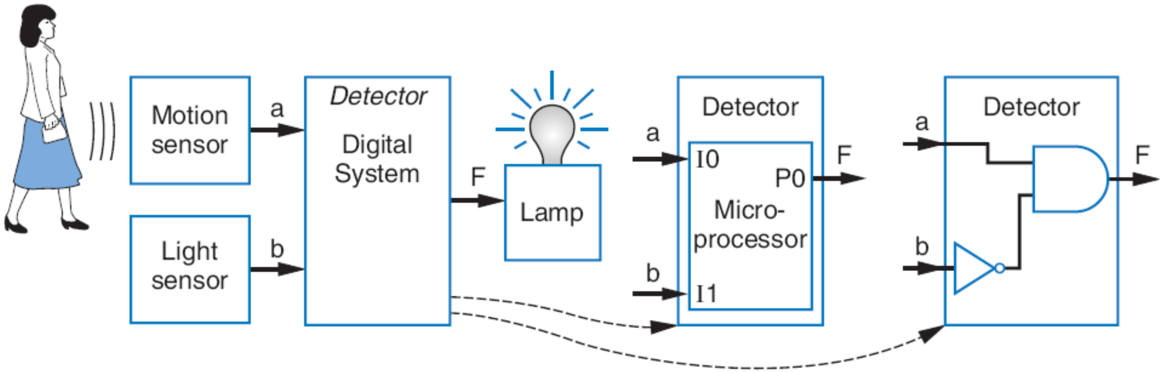
- Cada posição representa uma quantidade
  - Base 10 (decimal):
    - Dez símbolos: 0, 1, ... 9.
  - Base 2 (binário)
    - Dois símbolos: 0 e 1
- Combinação de símbolos para representar mais números

		5	2	3
10^4	10^3	10^2	10^1	10^0

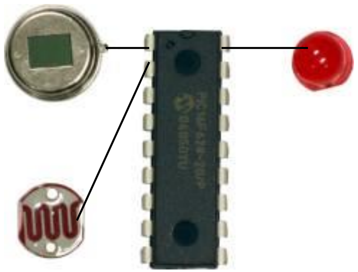
		1	0	1
2^4	2^3	2^2	2^1	2^0

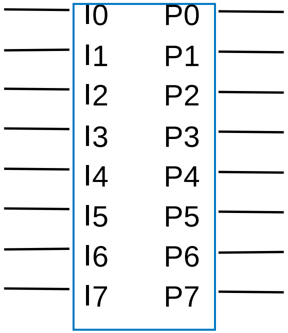
Implementando sistemas digitais: Processador vs Circuitos digitais

Detector de movimento no escuro

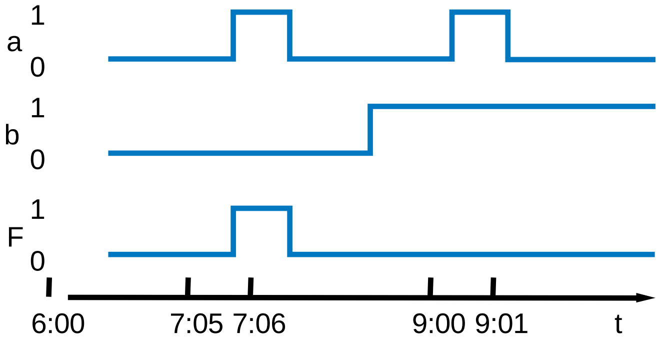


- Processadores são as escolhas mais naturais para implementação de sistemas digitais
  - Fáceis de programar
  - Podem ser baratos
  - Disponibilidade





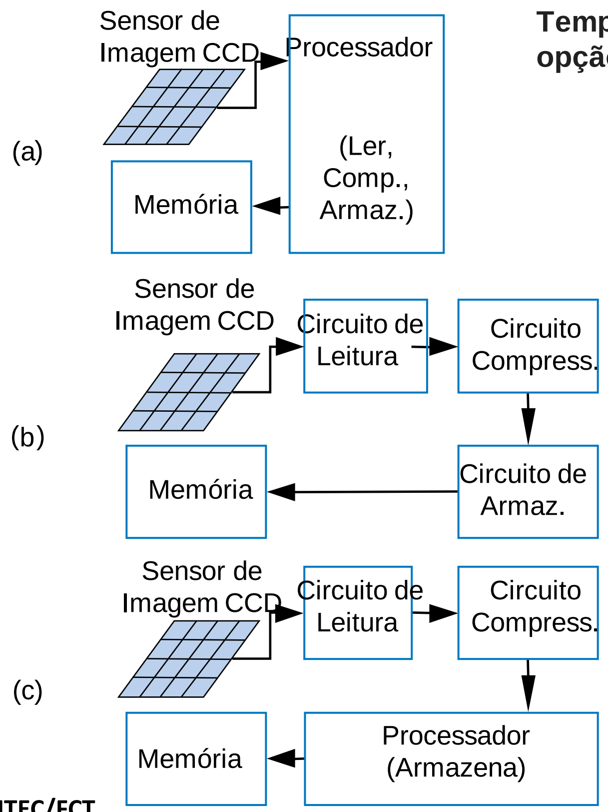

```
void main()
{
   while (1) {
      P0 = I0 && !I1;
      // F = a and !b,
   }
}
```



Implementando sistemas digitais: Processador vs Circuitos digitais

- Porque desenvolver um circuito digital, se já há processadores baratos disponíveis?
  - Processadores podem ser lentos para uma determinada aplicação
  - Outras restrições: energia, preço, tamanho
- Exemplo:

Tarefa	Processador	Circuito digital
Leitura	5	0.1
Compres.	8	0.5
Armazen.	1	0.8



Tempo decorrido em cada opção

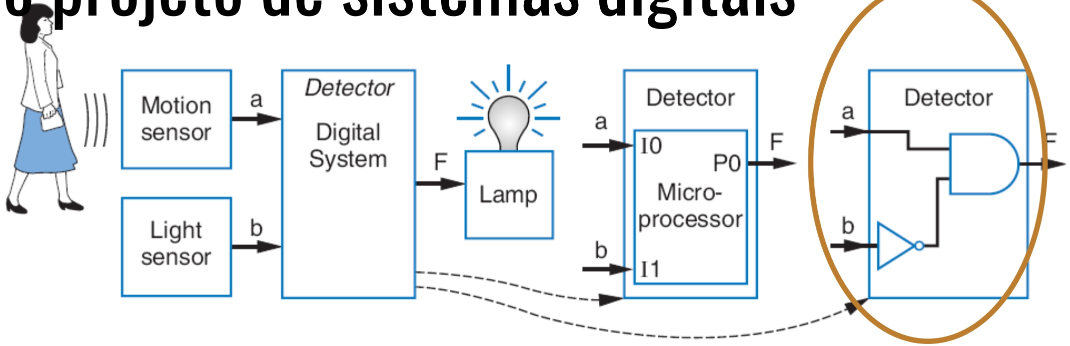
5+8+1
=14 s

.1+.5+.8
=1.4 s

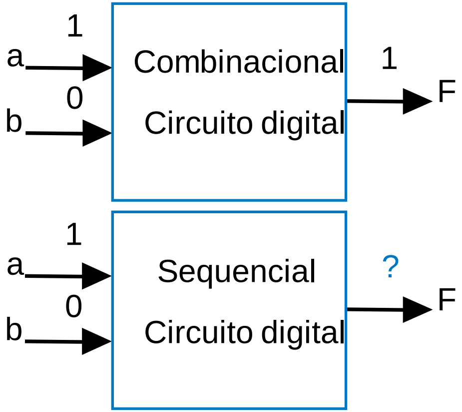
.1+.5+1
=1.6 s

Projeto de sistemas digitais

Introdução ao projeto de sistemas digitais

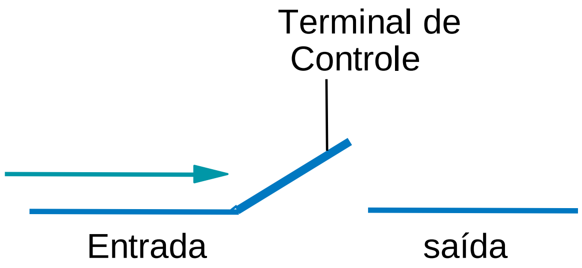


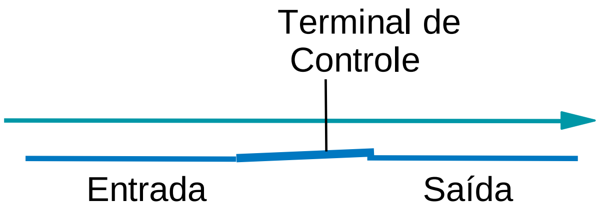
- Como se projetar sistemas digitais
  - Circuitos combinacionais: Circuito digital cuja saída depende somente das entradas
  - Circuito sequencial: Circuito digital cuja saída dependa das entradas e de valores anteriores.



Chaves eletrônicas para implementação de sistemas digitais

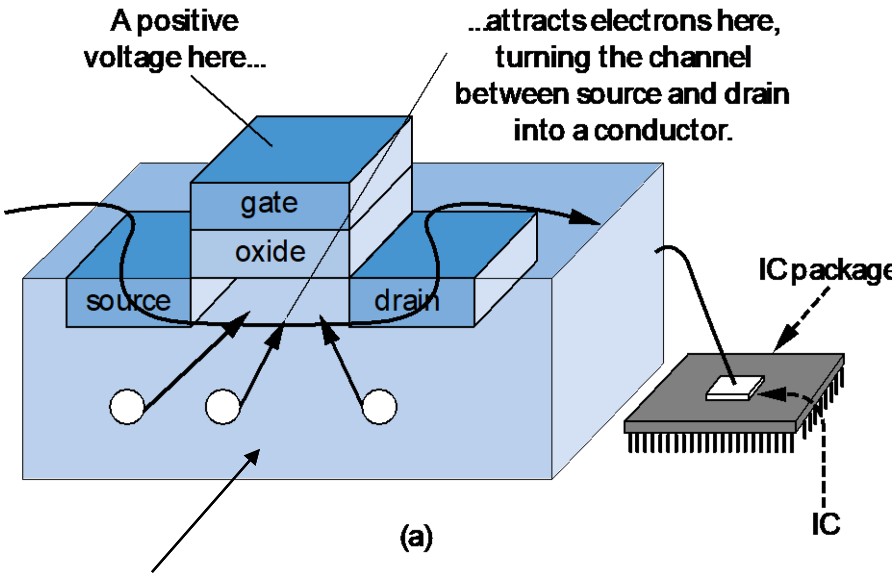
- Uma chave possui três terminais básicos
  - Entrada, saída e controle
    - Corrente flui da entrada para a saída, se o terminal de controle possui tensão



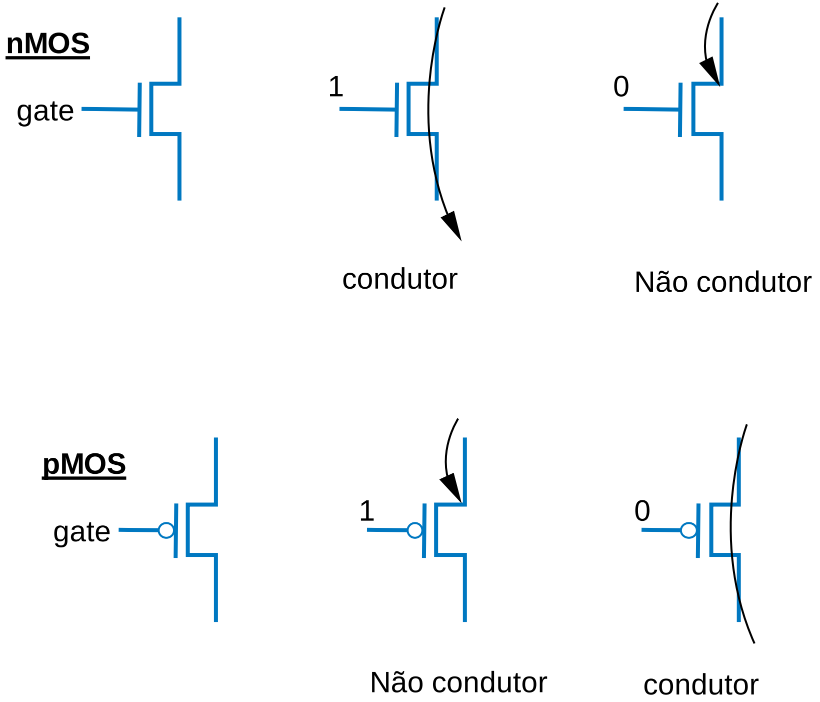


Chaves eletrônicas para implementação de sistemas digitais

- Transistor CMOS



Silício – nem bom nem mal condutor:
Semicondutor

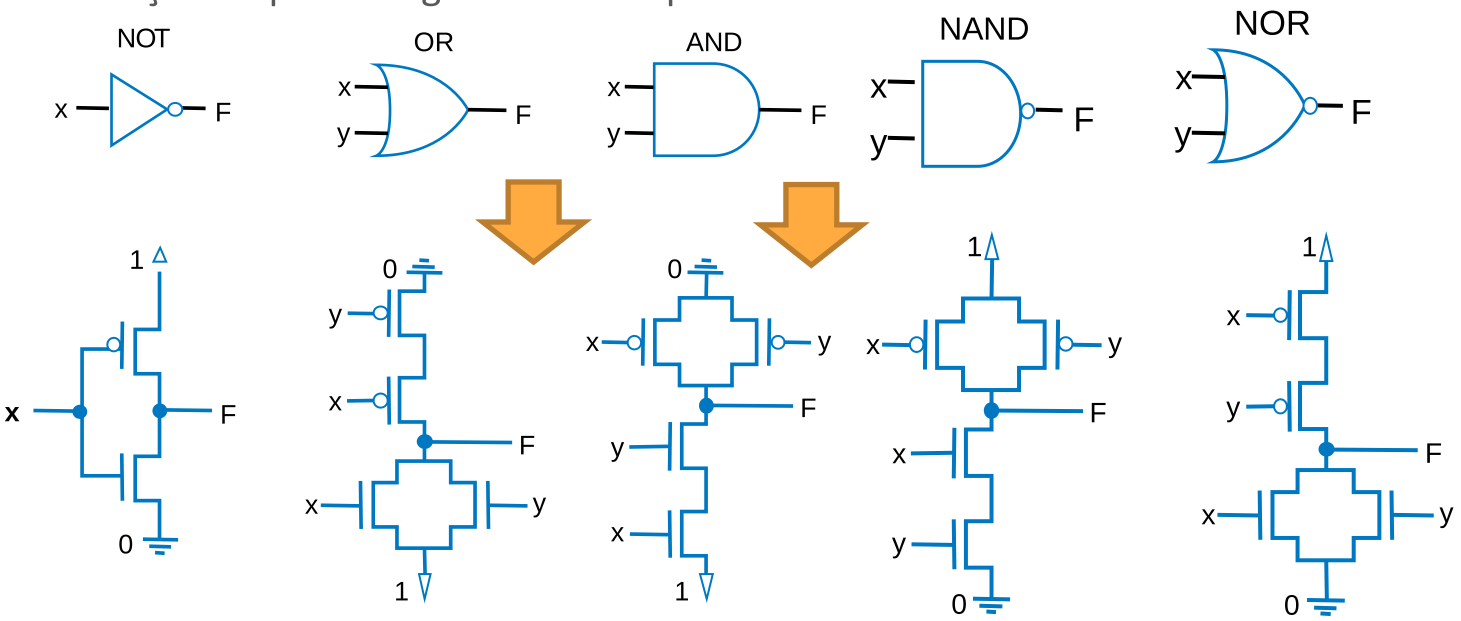


Chaves eletrônicas para implementação de sistemas digitais

- Em circuito digitais empregamos portas lógica e não transistores

Chaves eletrônicas para implementação de sistemas digitais

- Em circuito digitais empregamos portas lógica e não transistores
  - Construção de portas lógicas é feita a partir de transistores

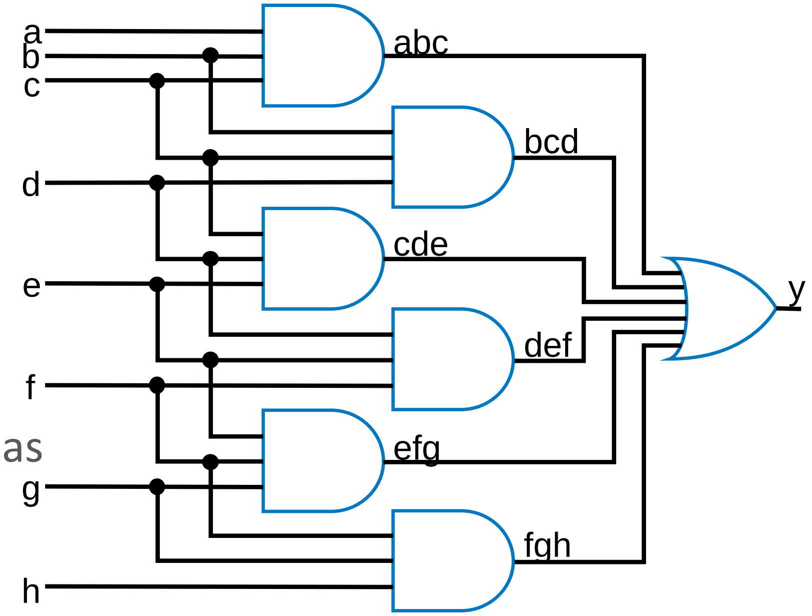


Projeto de um circuito combinacional

	Passo	Descrição
Passo 1	**Captura da função lógica**	Criação de uma tabela verdade ou equações que descreve o comportamento do circuito combinacional
Passo 2	**Conversão para equação**	Em caso de captura na forma de tabela-verdade, deve-se criar uma equação representada por OR de cada um dos mintermos. A equação pode ser simplificada.
Passo 3	**Implementação baseada em portas lógica**	Para cada saída, deve-se criar um circuito lógico correspondente à equação encontrada.

Projeto de um circuito combinacional

- Exemplo: Detector de 1s consecutivos in uma palavra de 8 bits
  - 000**111**01 → 1
  - 10101011 → 0
  - **111**10000 → 1
- Passo 1: Captura da função lógica
- Passo 2: Escrita da equação
  - y = abc + bcd + cde + def + efg + fgh
- Passo3: Implementação baseada em portas lógicas



Projeto de um circuito combinacional

- Conversão de uma tabela-verdade para equação lógica

Entradas		Saída	Regra
a	b	F	F = soma de
0	0	1	a'b'
0	1	1	a'b
1	0	0	
1	1	0	

F = a'b' + a'b

a	b	c	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	1	ab'c
1	1	0	1	abc'
1	1	1	1	abc

F = ab'c + abc' + abc

Projeto de um circuito combinacional

- Simplificação usando álgebra booleana

Comutativa

$a + b = b + a$

$a * b = b * a$

Distributiva

$a * (b + c) = a * b + a * c$

$a + (b * c) = (a + b) * (a + c)$ ← (atenção!)

Associativa

$(a + b) + c = a + (b + c)$

$(a * b) * c = a * (b * c)$

Identidade

$0 + a = a + 0 = a$

$1 * a = a * 1 = a$

Complemento

$a + a' = 1$

$a * a' = 0$

Alguns exemplos:

abc' equivalente a c'ba.

- Propriedade comutativa :

$a*b*c' = a*c'*b = c'*a*b = c'*b*a = c'ba$.

abc + abc' = ab

- Propriedade distributiva:

$abc + abc' = ab(c+c')$.

- Complemento $c+c' = 1$

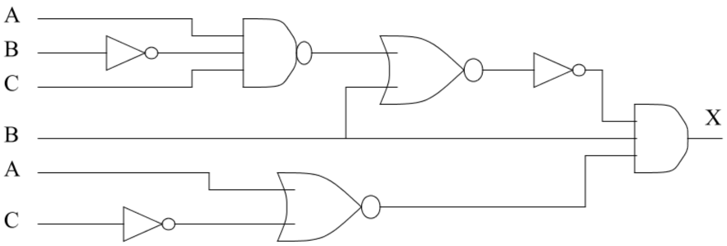
$ab(c+c') = ab(1)$.

- Identidade

$ab(1) = ab*1 = ab$

Projeto de um circuito combinacional

- Simplificação usando álgebra booleana



$$X = \overline{\left(\overline{\overline{(A\cdot\overline{B}\cdot C)}+B}\right)}\cdot B\cdot\overline{(A+\overline{C})}$$

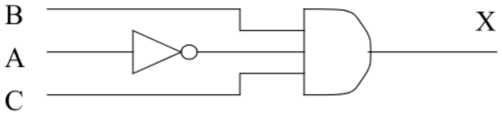
$$X = \overline{\left(\overline{\overline{(A\cdot\overline{B}\cdot C)}+B}\right)}\cdot B\cdot\overline{(A+\overline{C})}$$

$$X = (\overline{A}+B+\overline{C}+B)\cdot B\cdot(\overline{A}\cdot C)$$

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

$$X = B\cdot\overline{A}\cdot C+B\cdot\overline{A}\cdot C+0+B\cdot\overline{A}\cdot C$$

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



Projeto de um circuito combinacional

- Simplificação usando álgebra booleana

A	B	C	x	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\rightarrow \overline{A}BC$
1	0	0	0	
1	0	1	1	$\rightarrow A\overline{B}C$
1	1	0	1	$\rightarrow AB\overline{C}$
1	1	1	1	$\rightarrow ABC$

$$x = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

$$x = \overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

$$x = BC(\overline{A} + A) + AC(\overline{B} + B) + AB(\overline{C} + C)$$

$$x = BC + AC + AB$$

Projeto de um circuito combinacional

- Simplificação usando álgebra booleana

	A	B	C	D	z
(0)	0	0	0	0	0
(1)	0	0	0	1	0
(2)	0	0	1	0	0
(3)	0	0	1	1	0
(4)	0	1	0	0	0
(5)	0	1	0	1	0
(6)	0	1	1	0	0
(7)	0	1	1	1	$1 \rightarrow \bar{A}BCD$
(8)	1	0	0	0	$1 \rightarrow A\bar{B}\bar{C}\bar{D}$
(9)	1	0	0	1	$1 \rightarrow A\bar{B}\bar{C}D$
(10)	1	0	1	0	$1 \rightarrow A\bar{B}C\bar{D}$
(11)	1	0	1	1	$1 \rightarrow A\bar{B}CD$
(12)	1	1	0	0	$1 \rightarrow AB\bar{C}\bar{D}$
(13)	1	1	0	1	$1 \rightarrow AB\bar{C}D$
(14)	1	1	1	0	$1 \rightarrow ABC\bar{D}$
(15)	1	1	1	1	$1 \rightarrow ABCD$

$$z = \overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$$

$$\begin{aligned} z &= \overline{A}BCD + A\overline{B}\,\overline{C}(\overline{D} + D) + A\overline{B}C(\overline{D} + D) + AB\overline{C}(\overline{D} + D) + ABC(\overline{D} + D) \\ &= \overline{A}BCD + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC \\ &= \overline{A}BCD + A\overline{B}(\overline{C} + C) + AB(\overline{C} + C) \\ &= \overline{A}BCD + A\overline{B} + AB \\ &= \overline{A}BCD + A(\overline{B} + B) \\ &= \overline{A}BCD + A \end{aligned}$$

$$z = \overline{A}BCD + A = BCD + A$$

Projeto de um circuito combinacional

- Simplificação usando mapa de Karnaugh

A	B	C	D	X	
0	0	0	0	0	
0	0	0	1	1	$\rightarrow \bar{A}\bar{B}\bar{C}D$
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	$\rightarrow \bar{A}B\bar{C}D$
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	0	
1	0	1	1	0	
1	1	0	0	0	
1	1	0	1	1	$\rightarrow AB\bar{C}D$
1	1	1	0	0	
1	1	1	1	1	$\rightarrow ABCD$

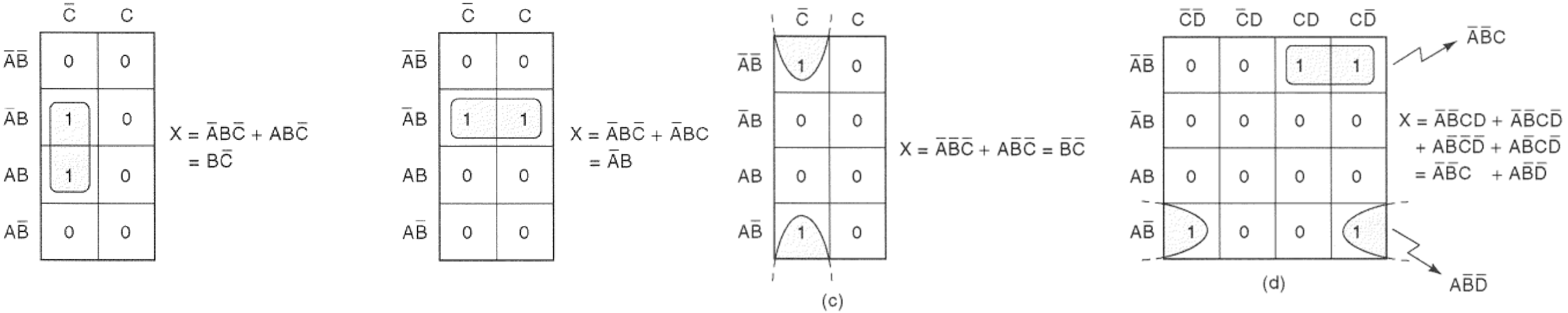
$$\left\{ \begin{aligned} X &= \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D \\ &+ AB\bar{C}D + ABCD \end{aligned} \right\}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Projeto de um circuito combinacional

- Simplificação usando mapa de Karnaugh

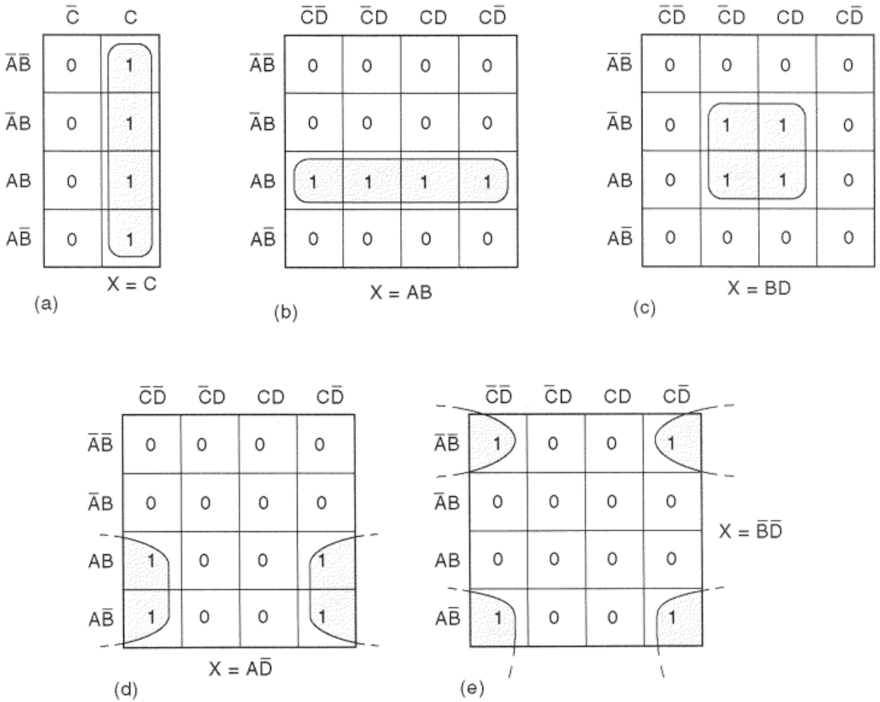
Variáveis que sofrem alteração nos pares saem da equação



Projeto de um circuito combinacional

Variáveis que sofrem alteração nos pares saem da equação

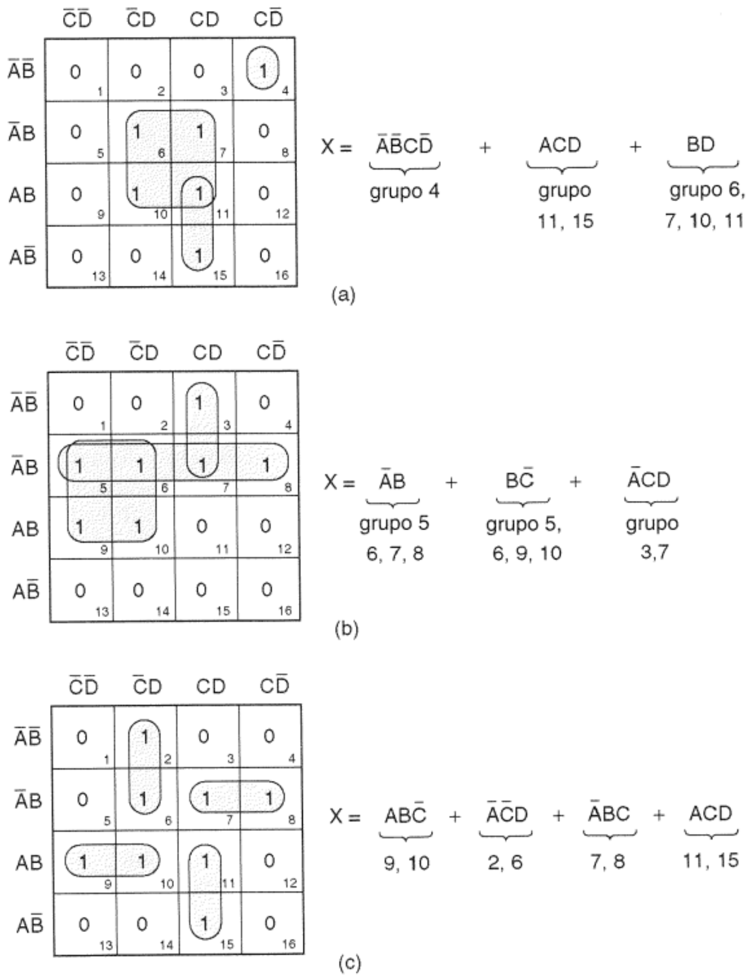
- Simplificação usando mapa de Karnaugh



Projeto de um circuito combinacional

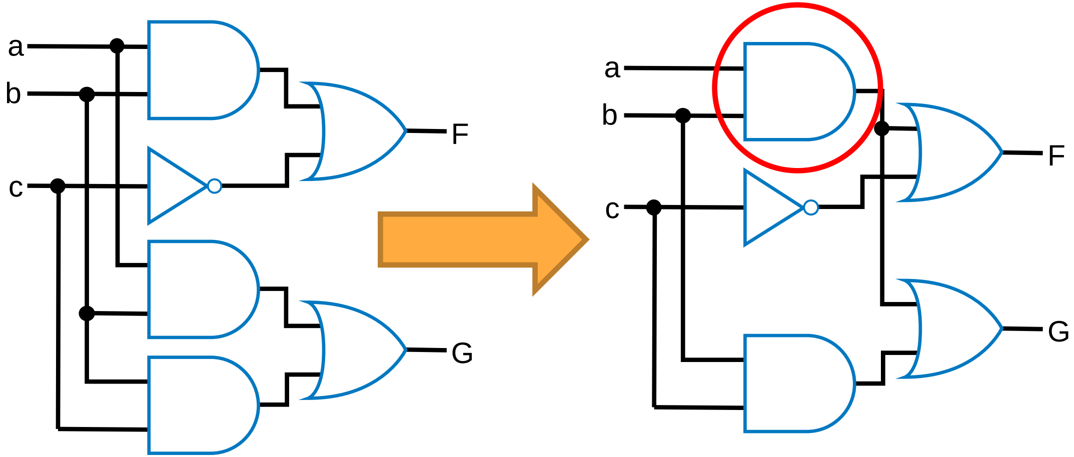
- Simplificação usando mapa de Karnaugh

Variáveis que sofrem alteração nos pares saem da equação



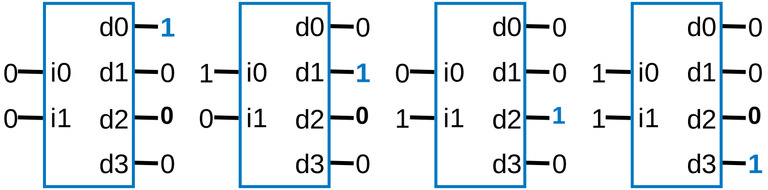
Projeto de um circuito combinacional

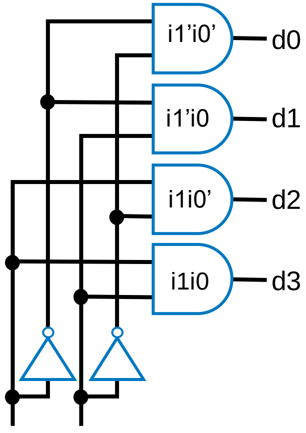
- Circuitos com mais de uma saída
    - Uma equação/tabela-verdade/circuito lógico para cada saída
    - Em alguns casos é possível reutilizar parte das portas lógicas do cálculo de uma das saídas no cálculo de outra saída.
    - Ex:
        - F = ab + c’
        - G = ab + bc

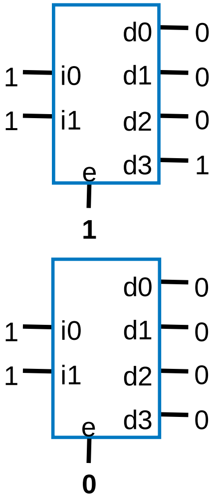


Projeto de um circuito combinacional

- Projeto utilizando blocos funcionais básicos pré-construídos → Decodificadores
    - Converte uma entrada binária para um formato no qual apenas um bit apresenta nível lógico alto
    - 2 bits → 4 bits saída, n bits → 2^n saídas

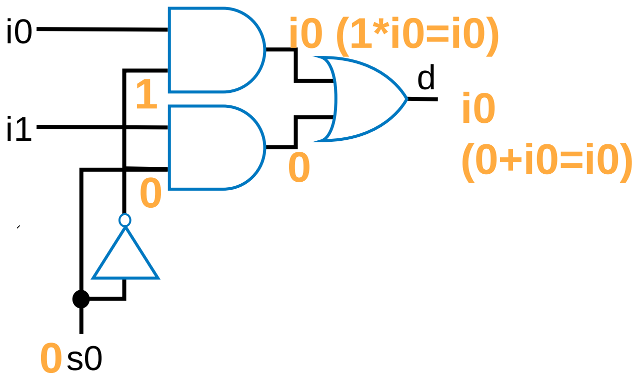


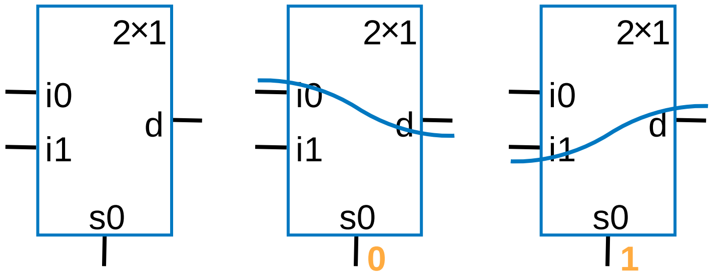




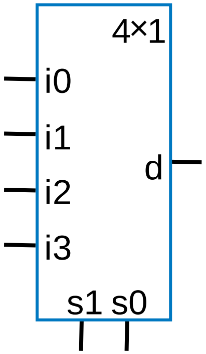
Projeto de um circuito combinacional

- Projeto utilizando blocos funcionais básicos pré-construídos → Multiplexadores
    - “Roteia” uma das entradas para a saída de acordo com uma entrada de controle
    - 2 entradas → 1 bits de seleção, n entradas → log(n) bits de seleção

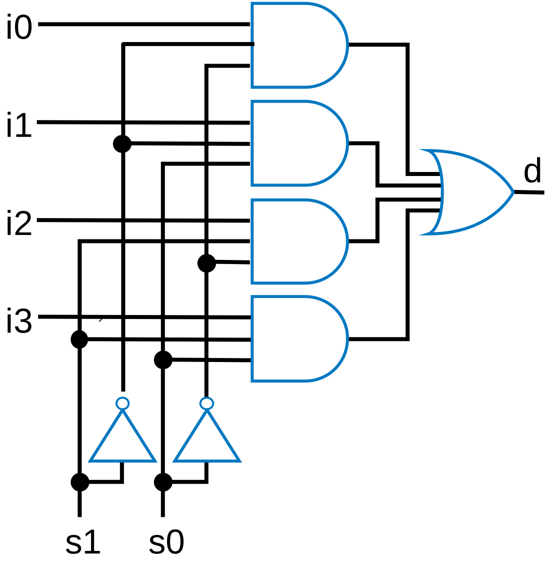




2x1 mux

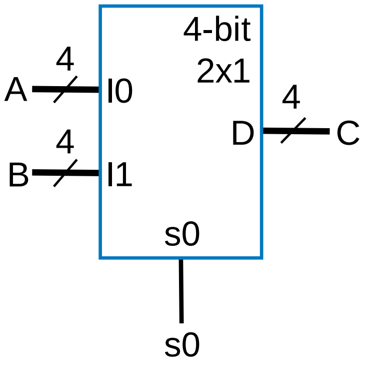


4x1 mux

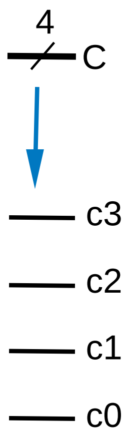


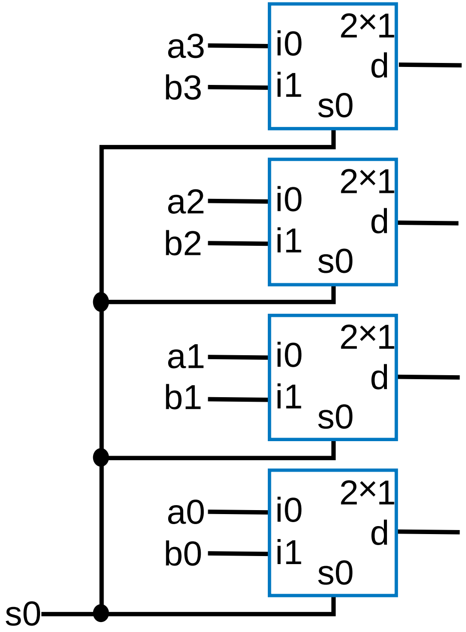
Projeto de um circuito combinacional

- Projeto utilizando blocos funcionais básicos pré-construídos → Multiplexadores
    - Multiplexadores de barramento (ou de n bits)



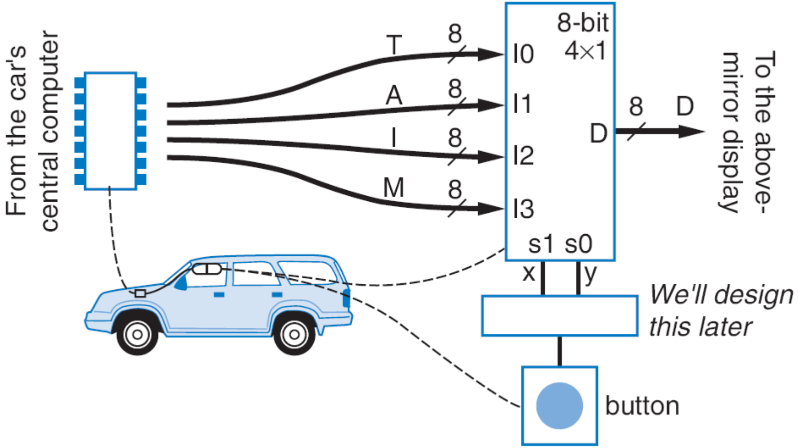
Notação





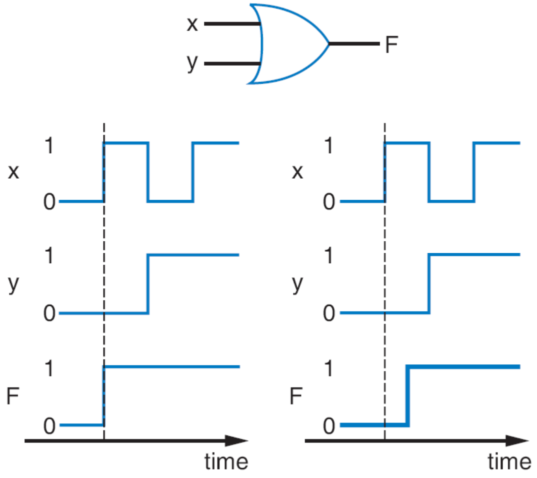
Projeto de um circuito combinacional

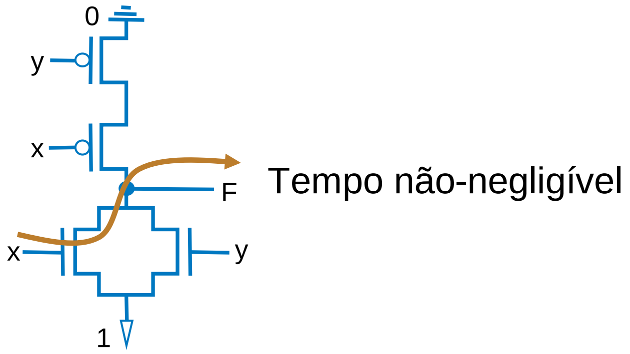
- Projeto utilizando blocos funcionais básicos pré-construídos → Multiplexadores
- Exemplo:
    - Display de um veículo que pode mostrar informações de temperatura do motor (T em °C), rendimento médio (A em km/L), rendimento instantâneo (I em km/L), autonomia (M em km)



Projeto de um circuito combinacional

- Comportamento não-ideal de portas lógicas
  - Geralmente desconsideramos o atraso que um sinal sofre da entrada para a saída de uma porta lógica.
  - Na prática, em alguns casos é necessário considerar o atraso da porta lógica





Bibliografia

- Sistemas digitais: Projeto, Otimização e HDLs, Frank Vahid, Ed. Bookman, 1ª Ed., 2008
- Tocci, R. J., Widmer, N. S. Sistemas digitais. 7. ed. Rio de Janeiro: LTC, 1998.