EA772 Circuitos Lógicos
Sistemas Combinacionais

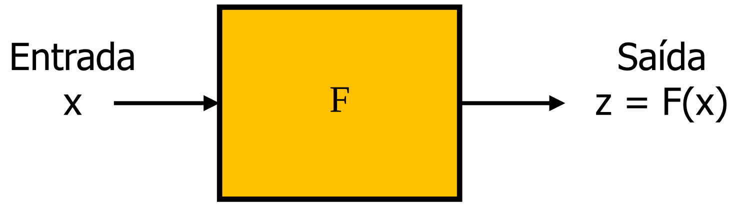
Prof. José Mario De Martino
Departamento de Engenharia de Computação e Automação Industrial
Faculdade de Engenharia de Elétrica e de Computação
Universidade Estadual de Campinas

Sala 317A - FEEC
martino@dca.fee.unicamp.br

UNICAMP

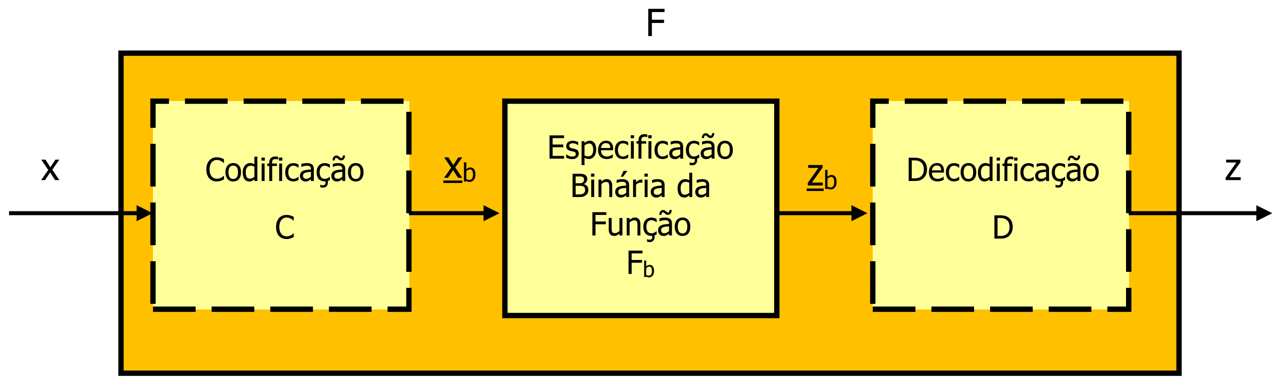
Sistema combinacional

- Conceito
  - sistema no qual o valor da saída em um determinado instante de tempo depende apenas do valor da entrada naquele instante de temp.



Especificação

- Dois tipos de especificação:
  - Especificação de alto nível (mais próxima do problema)
  - Especificação binária (mais próxima da implementação)



Especificação de alto nível

- Especificação de alto nível abrange 3 aspectos:
  - Especificação da entrada
  - Especificação da saída
  - Especificação da função (transformação da entrada na saída)

Especificação de alto nível da entrada e da saída

- Lista dos elementos:

  {0, 1, 2}

- Propriedade que identifique os elementos:

  $\{x \mid (5 \le x \le 10^4) \text{ e } (x \bmod 3 = 0)\}$

- Elementos compostos podem ser especificados como vetores de elementos primitivos:
  - Vetor de dígitos

    $\underline{x} = (x_{n-1}, x_{n-2}, \ldots, x_0)$ $\quad x_i \in \{0, 1, 2 \ldots, 9\}$ $\quad$ Exemplo: (4, 8, 0, 3) = 4803
  - Vetor de caracteres ("string")

    $\underline{c} = (c_{n-1}, c_{n-2}, \ldots, c_0)$ $\quad c_i \in \{A, B, \ldots, Z\}$ $\quad$ Exemplo: (F, E, E, C) = FEEC
  - Vetor de bits

    $\underline{y} = (y_{n-1}, y_{n-2}, \ldots, y_0)$ $\quad y_i \in \{0, 1\}$ $\quad$ Exemplo: (1, 1, 0, 1, 1, 1) = 110111

Especificação de alto nível da função

- Tabela

x	z
A	65
B	66
C	67

- Expressão aritmética

  z = 2 x + 4 (+ representa a soma)

- Expressão condicional

$$z = \begin{cases} a + b & se\ c > d \\ a - b & se\ c = d \\ 0 & se\ c < d \end{cases}$$

- Expressão lógica

  z = a + b (+ representa a soma lógica OR)

- Composição de funções mais simples

Especificação de alto nível: exemplo

- Sistema cuja entrada é uma "string" (cadeia de caracteres) de 4 caracteres, um único caractere e um número inteiro entre 0 e 3 (inclusive). A saída é um "string" de 4 caracteres. O sistema realiza a substituição, na "string" de entrada, do caractere da posição indicada pelo número inteiro .
  - Entradas

    $\underline{x} = (x_3, x_2, x_1, x_0) \quad x_i \in \{A, B, \ldots Z, a, b, \ldots z\}\ i = 0,\ldots,3$

    $y \in \{A, B, \ldots Z, a, b, \ldots z\}$

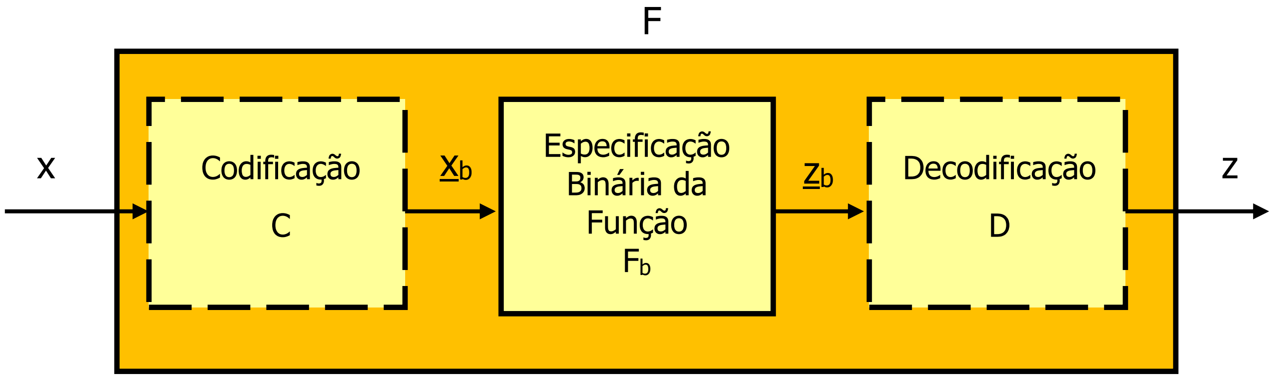
    $k \in \{0, 1, 2, 3\}$
  - Saídas

    $\underline{z} = (z_3, z_2, z_1, z_0) \quad z_i \in \{A, B, \ldots Z, a, b, \ldots z\}$
  - Função

$$z_j = \begin{cases} x_j & se\ j \neq k \quad j \in \{0, 1, 2, 3\} \\ y & se\ j = k \end{cases}$$

Especificação binária

- Para a especificação binária é necessário codificar o conjunto de entrada e decodificar o conjunto saída para/da representação binária.
- Códigos estabelecem a correspondência entre os conjuntos de alto nível e a representação binária.



Códigos: exemplo

Entrada	Código 1	Código 2
PAULO	000	01
JOSÉ	001	001
JOÃO	010	0001
CARLOS	011	00001
PEDRO	100	000001

Código para a representação de caracteres

- ASCII (American Standard Code for Information Interchange) 7 bits:

A	100 0001
B	100 0010
C	100 0011
...	
Y	101 1001
Z	101 1010
0	011 0000
1	011 0001
...	
9	011 1001
SP	010 0000
.	010 1110
(	010 1000
+	010 1011

Código para a representação de caracteres

- EBCDIC (Extended Binary-Coded Decimal Interchange Code)

A 1100 0001

B 1100 0010

C 1100 0011

...

Y 1110 1000

Z 1110 1001

0 1111 0000

1 1111 0001

...

9 1111 1001

SP 0100 0000

. 0100 1011

(0100 1101

+ 0100 1110

Código para a representação de números inteiros

- Sistema de numeração (código binário)
- Código de Gray para representar valores de 0 a 15 (valores consecutivos diferem em apenas 1 bit)

0	0000	11	1110
1	0001	12	1010
2	0011	13	1011
3	0010	14	1001
4	0110	15	1000
5	0111		
6	0101		
7	0100		
8	1100		
9	1101		
10	1111		

Códigos para a representação de decimais (0 a 9)

	C1	C2	C3	C4
0	0000	0000	0011	00011
1	0001	0001	0100	11000
2	0010	0010	0101	10100
3	0011	0011	0110	01100
4	0100	0100	0111	10010
5	0101	1011	1000	01010
6	0110	1100	1001	00110
7	0111	1101	1010	10001
8	1000	1110	1011	01001
9	1001	1111	1100	00101

C1 – Código BCD : ponderado 8421.

C2 – Código 2421: ponderado; para complemento de 9 basta complementar bits.

C3 – Código Excesso-3: para complemento de 9 basta complementar bits.

C4 – Código 2-entre-5: representação sempre com 2 bits iguais a 1; detecção de erro em um bit; utiliza 5 bits.

Especificação binária

- Como a especificação de alto-nível, descreve:
  - Entrada
  - Saída
  - Função (transformação da entrada na saída)
- Especificação binária da entrada e da Saída
  - A especificação advém da codificação/decodificação adotada.
- Função
  - Um sistema combinacional de n entradas (binárias) e m saídas (binárias) é representado por m funções (funções booleanas ou de chaveamento).
  - Cada função booleana tem como domínio o conjunto de n-tuplas (vetor de n bits) e como imagem o conjunto {0, 1}.

Especificação binária da função

- Tabela verdade

x_2 x_1 x_0	$f(x_2, x_1, x_0)$
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	1
1 1 1	1

j	$f(j)$
0	0
1	0
2	1
3	1
4	0
5	0
6	1
7	1

OBS.: j representa de forma compacta o valor em decimal das variáveis de entrada.

Especificação binária da função

- Especificação do conjunto-um (ou conjunto-verdade)

  $f(x_2, x_1, x_0)$ = conjunto-um {2, 3, 6, 7}

- Especificação do conjunto-zero (ou conjunto-falso)

  $f(x_2, x_1, x_0)$ = conjunto-zero {0, 1 , 4, 5}

- Expressão booleana

  $f(x_2, x_1, x_0) = x_1$

Funções booleanas importantes

- Uma Variável

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

$f_0(x) = 0$ → constante 0

$f_1(x) = x$ → identidade

$f_2(x) = x$ (ou $x´$) → NOT (complemento)

$f_3(x) = 1$ → constante 1

Funções booleanas importantes

- Duas variável

	$x_1 x_0$ 00	01	10	11	
f_0	0	0	0	0	
f_1	0	0	0	1	→ AND (E)
f_2	0	0	1	0	
f_3	0	0	1	1	
f_4	0	1	0	0	
f_5	0	1	0	1	
f_6	0	1	1	0	→ XOR (eXclusive OR)
f_7	0	1	1	1	→ OR (OU)
f_8	1	0	0	0	→ NOR
f_9	1	0	0	1	→ XNOR (Equivalência)
f_{10}	1	0	1	0	
f_{11}	1	0	1	1	
f_{12}	1	1	0	0	
f_{13}	1	1	0	1	
f_{14}	1	1	1	0	→ NAND
f_{15}	1	1	1	1	

Funções booleanas importantes

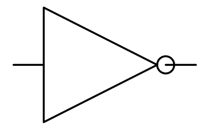
- Duas variável (expressões booleanas)
  - AND $x_1\ x_0$
  - NAND $(x_1\ x_0)'$
  - OR $x_1 + x_0$
  - NOR $(x_1 + x_0)'$
  - XOR $x_1\ x'_0 + x'_1\ x_0$
  - XNOR $x'_1\ x'_0 + x_1\ x_0$

Funções booleanas: representação esquemática

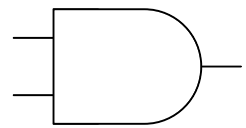
- Representação esquemática utilizando portas.

- Porta é um módulo que implementa uma função de booleana simples, possuindo apenas 1 saída.

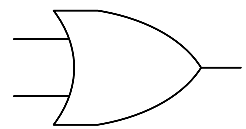
Funções booleanas: representação esquemática



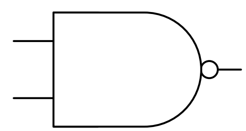
NOT



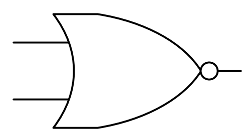
AND



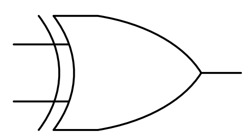
OR



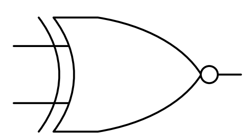
NAND



NOR



XOR



XNOR

Funções booleanas incompletas

- Além dos estados (valores) 0 e 1, pode existir o estado "don´t care" (não importa).
- O estado "don´t care" ocorre sempre que algum valor da entrada não é permitido (não ocorre) e portanto não importa ("don´t care") o valor da saída. Esta pode, portanto, ser associada ao valor 0 ou ao 1 dependendo da conveniência do projeto.
- A representação por conjunto, quanto existe o conjunto-dc (conjunto "dont´t care"), deve ser expressa por pares de conjuntos:
  - conjunto-um e conjunto-zero
  - conjunto-um e conjunto-dc
  - conjunto-zero e conjunto-dc

Funções booleanas incompletas: exemplo

- A entrada é um conjunto de 4 bits, representando um valor BCD. A saída deve indicar 1 sempre que o número na entrada for ímpar e 0 caso contrário.

Funções booleanas incompletas: exemplo

Entrada	Saída
0 0 0 0	0
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	0
0 1 1 1	1
1 0 0 0	0
1 0 0 1	1
1 0 1 0	X (“don´t care”)
1 0 1 1	X
1 1 0 0	X
1 1 0 1	X
1 1 1 0	X
1 1 1 1	X

Expressões booleanas

- Duas expressões booleanas são equivalente se representam a mesma função booleana

$W = x_1\ x_0 + x'_1$

$Z = x'_1 + x_0$

$x_1\ x_0$	W	Z
0 0	1	1
0 1	1	1
1 0	0	0
1 1	1	1

- É possível obter expressões booleanas equivalentes através da manipulação algébrica (utilizando a álgebra de Boole)

$x_1\ x_0 + x'_1 = (x_1 + x'_1)\ \ (x_0 + x'_1) = x_0 + x'_1 = x'_1 + x_0$

Expressões booleanas: Soma de mintermos

- Um mintermo de *n* variáveis é o produto (AND) em que todas as variáveis aparecem uma única vez na forma complementada ou na forma não-complementada.

- Exemplos de mintermos de 4 variávies $x_3\ x_2\ x_1\ x_0$

  $x'_3\ x'_2\ x'_1\ x'_0$

  $x'_3\ x'_2\ x'_1\ x_0$

  $x'_3\ x'_2\ x_1\ x_0$

  $x'_3\ x_2\ x_1\ x_0$

- Para n variáveis $\rightarrow 2^n$ mintermos

Expressões booleanas: Soma de mintermos

- Os mintermos são denotados por:
  - $m_0 \; m_1 \; m_2 \; \ldots \; m_{2^n-1}$
  - $m_j \quad j = 0, 1, 2, \ldots, 2^n-1$
  - j é o valor decimal obtido associando "1" à variável não-complementada e "0" à variável complementada (lógica positiva)
  - Cada mintermo tem o valor igual a "1" para uma única atribuição de valores para as variáveis, a saber, a atribuição associada ao valor decimal j.

$m_0 = 1$	$0 = 0\,0\,0\,0_2$	$m_0 = x'_3 \; x'_2 \; x'_1 \; x'_0$
$m_{11} = 1$	$11 = 1\,0\,1\,1_2$	$m_{11} = x_3 \; x'_2 \; x_1 \; x_0$
$m_{15} = 1$	$15 = 1\,1\,1\,1_2$	$m_{15} = x_3 \; x_2 \; x_1 \; x_0$

Expressões booleanas: Soma de mintermos

- O mintermo m_j representa uma função booleana cujo conjunto-um tem somente o elemento j.

- Uma expressão booleana dada pela soma de mintermos representa uma função booleana cujo conjunto-um é formado pelos estados da variáveis de entrada associados aos mintermos
  - Exemplo

$$F(x_2, x_1, x_0) = x'_2\, x'_1\, x'_0 + x'_2\, x_1\, x'_0 + x_2\, x_1\, x'_0$$

$$F(x_2, x_1, x_0) = m_0 + m_2 + m_6$$

$$F(x_2, x_1, x_0) = \text{conjunto-um}\{0, 2, 6\}$$

- Toda expressão booleana pode ser representada por um soma de mintermos (soma de produtos canônica).

Soma de mintermos: exercícios

- Qual a soma de mintermos que representa a função booleana especificada pela tabela verdade abaixo:

x_2	x_1	x_0	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- Qual a soma de mintermos equivalente à expressão:

$$F(x_2, x_1, x_0) = x_2 (x_1 x_0)' + x_2 x'_0$$

Expressões booleanas: Produto de Maxtermos

- Um Maxtermo de *n* variáveis é a soma (OR) em que todas as variáveis aparecem uma única vez na forma complementada ou na forma não-complementada.

- Exemplos de maxtermos de 4 variávies $x_3\ x_2\ x_1\ x_0$

  $x'_3 + x'_2 + x'_1 + x'_0$

  $x'_3 + x'_2 + x'_1 + x_0$

  $x'_3 + x'_2 + x_1 + x_0$

  $x'_3 + x_2 + x_1 + x_0$

- Para n variáveis $\rightarrow$ 2^n maxtermos.

Expressões booleanas: Produto de Maxtermos

- Os maxtermos são denotados por:
  - $M_0\ M_1\ M_2\ \ldots\ M_{2^n-1}$
  - $M_j \quad j = 0, 1, 2, \ldots, 2^n-1$
  - j é o valor decimal obtido associando "0" à variável não-complementada e "1" à variável complementada (lógica negativa)
  - Cada maxtermo tem o valor igual a "0" para uma única atribuição de valores para as variáveis, a saber, a atribuição associada ao estado j.

$M_0 = 0$	$0 = 0\ 0\ 0\ 0_2$	$M_0 = x_3 + x_2 + x_1 + x_0$
$M_{11} = 0$	$11 = 1\ 0\ 1\ 1_2$	$M_{11} = x'_3 + x_2 + x'_1 + x'_0$
$M_{15} = 0$	$15 = 1\ 1\ 1\ 1_2$	$M_{15} = x'_3 + x'_2 + x'_1 + x'_0$

Expressões booleanas: Produto de Maxtermos

- O maxtermo M_j representa uma função booleana cujo conjunto-zero tem somente o elemento j.

- Uma expressão booleana dada pelo produto de maxtermos representa uma função booleana cujo conjunto-zero é formado pelos estados das variáveis associados aos maxtermos
  - Exemplo

    $F(x_2, x_1, x_0) = (x_2 + x_1 + x_0)\,(x_2 + x_1' + x_0')\,(x_2' + x_1 + x_0')$

    $F(x_2, x_1, x_0) = M_0\, M_3\, M_5$

    conjunto-zero = {0, 3, 5}

- Toda expressão booleana pode ser representada por um produto de maxtermos (produto de somas canônico).

Produto de Maxtermos: exercícios

- Qual o produto de Maxtermos que representa a função booleana especificada pela tabela verdade abaixo:

x_2	x_1	x_0	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- Qual ao produto de maxtermos equivalente à expressão:

$$F(x_2, x_1, x_0) = x_2\ (\ x_1\ x_0)' + x_2\ x_0'$$

Especificação: exercício

- Especificar um sistema combinacional para comparar dígitos da base 4. O sistema deve comparar dois dígitos da base 4, x e y, e produzir a saída G (greater) se $x > y$, E (equal) se $x = y$ e L (less) se $x < y$.

Especificação: exercício

- Um sistema combinacional tem entrada x, a qual representa um dígito decimal. A saída z é o quadrado de x se x for maior do que 4; caso contrário, a saída z é duas vezes x.
  - Dê uma descrição de alto nível do sistema utilizando expressões matemáticas;
  - Apresente também uma tabela da função entrada-saída realizada pelo sistema.

Especificação: exercício

- Um incrementador/decrementador combinacional tem como entradas um número inteiro na faixa de 0 a $2^{16} - 1$ e um sinal de controle binário. Se o sinal de controle tiver o valor 1, o sistema incrementa módulo 2^{16} (isto é, ele computará $z = (x+1) \bmod 2^{16}$); caso contrário, ele decrementa módulo 2^{16}.
  - Dê uma descrição de alto nível do sistema em termos expressões aritméticas condicionais.
  - A representação na forma de tabela da função entrada-saída é viável?

Especificação: exercício

- Um sistema combinacional computa a distância entre dois 1's no vetor de bits de entrada x = (x3, x2, x1, x0), o qual tem exatamente dois 1's. Por exemplo, para x = (0, 1, 0, 1), a distância é 2. Ou seja, se os dois 1's estiverem nas posições i ej, a distância será |i – j|.
  - Apresente um especificação de alto nível do sistema
  - Apresente uma especificação em nível binário. A função e entrada-saída deve ser especificada utilizando expressões lógicas de soma de produtos e produtos de soma.