

SISTEMAS DE NUMERAÇÃO

Sistemas Digitais

- **DECIMAL**
 - SÍMBOLOS 0,1 .. 9
 - $1842 \Rightarrow 1 \times 10^3 + 8 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$
- **OCTAL**
 - SÍMBOLOS 0..7
 - $1634 \Rightarrow 1 \times 8^3 + 6 \times 8^2 + 3 \times 8^1 + 4 \times 8^0$
- **HEXADECIMAL**
 - SÍMBOLOS 0.. 9,A,B,C,D,E,F
 - $5F1A0 \Rightarrow 5 \times 16^4 + 15 \times 16^3 + 1 \times 16^2 + 10 \times 16^1 + 0 \times 16^0$
- **BINÁRIO**
 - SÍMBOLOS 0,1
 - $10110 \Rightarrow 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

A **POSICÃO** é que dá importância ou **PESO** ao dígito.
 O dígito **MAIS SIGNIFICATIVO** é o que está mais à esquerda (MSB)
 O dígito **MENOS SIGNIFICATIVO** é o que está mais à direita (LSB)

1

SISTEMA BINÁRIO

Sistemas Digitais

- **IMPORTÂNCIA DO SISTEMA BINÁRIO**
 - Fácil implementação física
 - Implementação com sistemas hidráulicos, eléctricos, luminosos, etc.
- **CONVERSÕES:**
 - DECIMAL \rightarrow BINÁRIO
 - BINÁRIO \rightarrow DECIMAL

$26_d = 11010_b$

$10100110_b = 166_d$

1	0	1	0	0	1	1	0
128	64	32	16	8	4	2	1
+	+	+	+	+	+	+	+
							128+32+4+2=166

2

BASES POTÊNCIAS DE 2

Sistemas Digitais

- As bases que são potências de 2 são facilmente convertidas em binário e vice-versa
 - Octal 1 dígito octal = 3 dígitos binários
 - Hexadecimal 1 dígito hexa = 4 dígitos binários

$2D3_H = 1011010011_b = 1323_{Oct}$

- **Vantagens**
 - Usam menos dígitos para representar um dado número
 - São mais facilmente entendidas por humanos
 - São muito usadas

3

Aritmética binária

Sistemas Digitais

- Basicamente as mesmas regras que a aritmética decimal !
 - Somam-se os números dígito a dígito
 - De um dígito para o seguinte (mais significativo), pode "ir um", ou seja pode haver "CARRY"
 - 1 e 1 são dois (ou seja 10_b)

Exemplo:

Adição	$(11011)_2$ $+ (10011)_2$ <hr style="width: 100%;"/> $(101110)_2$	$(647)_{10}$ $+ (537)_{10}$ <hr style="width: 100%;"/> $(1184)_{10}$
--------	---	--

Multiplicação

	1101 $\times 101$ <hr style="width: 100%;"/> 1101 0000 1101 <hr style="width: 100%;"/> 1000001	152 $\times 231$ <hr style="width: 100%;"/> 152 456 304 <hr style="width: 100%;"/> 35012
--	---	---

São apenas deslocamentos e somas !!!

4

Aritmética binária

Sistemas Digitais

- Numa máquina, o número de dígitos é FINITO
 - Não posso usar todos os dígitos que quiser
 - Há um número MÁXIMO que se pode representar:

- **Consequência da representação com um número FINITO de dígitos**
 - Os números não são representados por uma recta, mas sim por uma circunferência !

5

REPRESENTAÇÃO DE NÚMEROS NEGATIVOS

Sistemas Digitais

- **Problema:**
 - Como indicar que um número é negativo, sem usar o símbolo "-" (usando apenas 0 e 1)
 - Solução: usar uma das posições para representar o sinal
- **SINAL E MÓDULO** (signed integer)
 - O bit mais significativo representa o sinal, e os restantes a magnitude
 - Sinal = 0 => Positivo (representação normal)
 - Sinal = 1 => Negativo

Exemplos:

0100 = 4	1100 = -4
0010 = 2	1011 = -3

6

COMPLEMENTO PARA 2

Sistemas Digitais

- Ideia Base**
 - Facilitar somas e subtrações

$N^{(2)} = M - N$

NÚMERO 'N'
MÓDULO (MÁX Nº DE 1+1)
NÚMERO 'N' EM COMPLEMENTO PARA 2

Regra para fazer as conversões:
COMPLEMENTAR TODOS OS DÍGITOS A PARTIR DO 1º '1'

00110 (6_{10}) \Rightarrow 11010 (-6_{10})

100000
-00110

111010

COMPLEMENTO PARA 2

Sistemas Digitais

- Complemento para 2**
 - Usa o bit mais significativo para representar o sinal (tal como anteriormente)
 - Os restantes bits são calculados de acordo com o algoritmo apresentado
 - Vantagens
 - Permite ver rapidamente se um número é positivo ou negativo
 - Não existem números repetidos (com 2 representações)
 - O número -1 está imediatamente antes do 0
 - As operações de soma e subtração podem ser feitas usando os algoritmos usuais
- Algoritmos para a conversão positivo/negativo em complemento p/2**
 - Subtrair o número positivo ao número 1000... (2^N)
 - Começar do lado direito, e deixar na mesma todos os dígitos até ao primeiro 1 (inclusive). Complementar todos os dígitos a partir desse ponto.

Números não-inteiros

Sistemas Digitais

- Representação em vírgula fixa**
 - Igual a sinal e módulo
 - Número pré-determinado de dígitos são para a parte fraccionária
 - Exemplo:
 - Pré-fixado que há dois dígitos binários para a parte fraccionária
 - $2.5 = 1010$
- Coefficientes da parte fraccionária**
 - Potências negativas da base
 - $2^{-1} (=0.5)$, $2^{-2} (=0.25)$, ...
- Representação em vírgula flutuante**

Representação em vírgula flutuante

Sistemas Digitais

- Permitem uma maior GAMA DE VARIAÇÃO**
- Têm menos precisão que vírgula fixa**
- Representação:**

$$+ 0.43 \times 10^{+6} = 430000$$

Sinal Mantissa Exponente (c/sinal) Base
- Norma IEEE-754 (single precision floating point)**
 - 32 bits - 1 bit de sinal, 8 bits de expoente, 23 bits de mantissa

CÓDIGOS BINÁRIOS - numéricos

Sistemas Digitais

- Para representação de números, sem ser em binário natural**
 - Para simplificar as convenções binário / decimal
 - BCD - Binary coded decimal (natural, ou 8421)
 - Usam-se 4 dígitos binários para cada dígito decimal
 - Perdem-se 6 posições em cada 16
 - Aiken (ou 2421)
 - Os bits têm peso 2421
 - Os números desperdiçados são os "do meio"
 - Permite distinguir facilmente os números maiores que 5
 - É autocomplementar
 - Excesso 3 (não ponderado)
 - Usa os 10 números "do meio" - 3 a 13
 - É autocomplementar
 - 7421 - Minimiza o consumo

CÓDIGOS BINÁRIOS - numéricos

Sistemas Digitais

Dec.	BCD	AIKEN	EXC.3	7421
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0010
3	0011	0011	0110	0011
4	0100	0100	0111	0100
5	0101	1011	1000	0101
6	0110	1100	1001	0110
7	0111	1101	1010	1000
8	1000	1110	1011	1001
9	1001	1111	1100	1010

Gray

0000
0001
0011
0010
0110
0111
0101
... ..
... ..

- Código Gray (binário reflectido)**
 - Serve para minimizar transições
 - Pode resolver problemas de estados transitórios nas mudanças
 - Conversores físicos
 - É um código cíclico
 - Fácil passagem para binário

Representar o mundo com 0's e 1's

Sistemas Digitais

- **Consigo representar números**
 - Binário natural, complemento para 2, vírgula fixa, vírgula flutuante
- **Com números posso representar:**
 - TEXTO
 - IMAGEM
 - SOM
 - Outros
 - Vibração
 - Cheiro
 - ...o que quiser....
- **Para trocar informação**
 - Tem que haver **NORMAS** para interpretar os 0s e 1s
 - Formato PDF, DOCX, JPG, XLS, MP3, etc,etc



13

CÓDIGOS BINÁRIOS - Alfanuméricos

Sistemas Digitais

- **Para representação de caracteres**
 - Código ASCII
 - American Standard Code for Information Interchange
 - Define caracteres normais, símbolos, e caracteres de controlo.
 - Extensões para 8 bits para caracteres especiais
 - Código ebcidic (usado apenas na IBM)
 - Unicode (16 bits, extensão do ASCII que inclui caracteres orientais)

0	16	32	48	64	80	96	112
1	17	33	49	65	81	97	113
2	18	DC2 34	50	66	82	98	114
3	19	DC3 35	51	67	83	99	115
4	20	DC4 36	52	68	84	100	116
5	21	37	53	69	85	101	117
6	22	38	54	70	86	102	118
7	BEL 23	39	55	71	87	103	119
8	BS 24	40	56	72	88	104	120
9	25	41	57	73	89	105	121
10	LF 26	42	58	74	90	106	122
11	27	ESC 43	59	75	91	107	123
12	FF 28	44	60	76	92	108	124
13	CR 29	45	61	77	93	109	125
14	SO 30	46	62	78	94	110	126
15	SI 31	47	63	79	95	111	127

14

Imagens

Sistemas Digitais

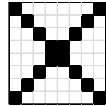
- **Ideia geral**
 - Dividir a imagem em "quadrinhos", ou "Picture Elements"
 - PIXEL
 - Cada PIXEL pode ocupar 1 bit (0,1) ou vários, para ter diferentes cores, intensidades, etc
- **Formatos raster**
 - BMP (Windows Bitmap)
 - 24 bits (3 Bytes), equivalente a R,G,B, por pixel.
 - Não tem compressão
 - TIFF (Tagged Image File Format)
 - 24 ou 32 bits
 - Usa compressão sem perdas (LZW)
 - JPEG (Joint Photographic Experts Group)
 - Compressão com perdas, RFC 1341
 - (RAW) - Sem cabeçalho, formatação, ou compressão
- **Outros formatos**
 - Formato Vectorial (p.ex. PCX)
 - Outros (GIF, PNG, CGM, SVG, JPG(2000), TGA, PDF, CDR, EPS, ODG, WMF, XPS, VML, XPS, DXF, PIC)

Exemplo de um formato de imagem

Sistemas Digitais

- **Consideremos a seguinte imagem:**
- **A codificação (raw) seria:**

```
1 0 0 0 0 0 0 1
0 1 0 0 0 0 1 0
0 0 1 0 0 1 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1
```


- **Representado de outra maneira**
 - Em bits:
 - 10000001010000100010010000011000000110000001001000100001010000001
 - Em números de 8 bits representados em:
 - Decimal: 129,66,36,24,24,36,66,129
 - Hexadecimal: 81,42,24,18,18,24,42,81
 - Com cabeçalho: (3 bytes para profundidade de cor, 2 para largura, 2 para altura: 00 00 01 00 08 00 08 81 42 24 18 18 24 43 81.

16

Sons

Sistemas Digitais

- **Ideia geral**
 - O som é um sinal que representa variações de pressão
 - Em cada instante verifica-se qual o valor (relativo) da pressão
 - Quantificação do sinal determina erro quantificação
 - Periodicamente vai-se repetir o processo
 - Frequência de amostragem determina a gama de frequências que se consegue observar.
- **Formatos de som**
 - WAV
 - Guarda o sinal de pressão
 - MP3
 - Cumprime a informação
 - Calcula a transformada de fourier, e guarda apenas os coeficientes mais importantes

ERROS

Sistemas Digitais

- **O que é um erro**
 - É um 1 passar a 0, ou vice-versa
- **Erros de transmissão**
- **Degradação do meio magnético**
- **Soluções**
 - Mandar informar redundante para confirmação
 - Utilização de BITS DE PARIDADE
 - 1 bit permite detectar se houve um número impar de erros
 - Paridade Par, Ímpar, Mark, e Space
 - Paridade byte a byte, e paridade vertical
 - Utilização de códigos correctores
 - Códigos de Hamming 5/3
 - Utilização de checksums

18

Sistemas Digitais

ALGEBRA DE BOOLE

Regras matemáticas para manipular os 0's e 1's com que representamos o mundo

19

Sistemas Digitais

Álgebra DE BOOLE

- Definição FORMAL

$\{U, +, \cdot\}$ $U =$ Conjunto finito
 $+, \cdot =$ Operações (soma, produto)

1 $\rightarrow a + b \in U$ 4 $\rightarrow a(b + c) = ab + ac$
 $a \cdot b \in U$ $a + bc = (a + b)(a + c)$

2 $\rightarrow a + b = b + a$ 5 $\rightarrow a + X = 1$
 $a \cdot b = b \cdot a$ $a \cdot X = 0$

3 $\rightarrow a + 0 = a$ $X = \bar{a}$ (complemento)
 $a \cdot 1 = a$

20

Sistemas Digitais

UTILIDADE EM SISTEMAS LÓGICOS

- Consideramos $U = \{0,1\}$
 - o conjunto U é apenas os 2 valores binários
 - podemos implementar facilmente este tipo sistemas com: lâmpadas, relés, transistores, actuadores mecânicos e hidráulicos, etc.

Usamos binário porque é fácil fazer máquinas que tenham 2 estados possíveis

- Operação adição
 - Corresponde ao OU lógico
- Operação de multiplicação
 - Corresponde ao E lógico
- Operação complemento
 - É a simples negação

$U = \{0, 1\}$
 $+$ = "OR" (operação OU)
 \cdot = "AND" (operação E)
 Complemento = "NOT" (operação NEGAÇÃO)

21

Sistemas Digitais

TEOREMAS

- Vão ser as ferramentas para toda a manipulação de dados que vamos fazer...
- PRINCÍPIO DA DUALIDADE
 - Se uma dada proposição é verdadeira, então, substituindo os 0 com 1 e os 1 com 0, obtenho também uma proposição verdadeira
- 1 - ELEMENTO ABSORVENTE
 - $A \cdot 0 = 0$ $A + 1 = 1$
- 2 - ELEMENTO NEUTRO
 - $A \cdot 1 = A$ $A + 0 = A$
- 3 - IDEMPOTÊNCIA
 - $A \cdot A = A$ $A + A = A$

22

Sistemas Digitais

TEOREMAS

- 4 - COMPLEMENTARIDADE
 - $A \cdot \bar{A} = 0$ $A + \bar{A} = 1$
- 5 - INVOLUÇÃO
 - $A = \bar{\bar{A}}$
- 6 - COMUTATIVIDADE
 - $A \cdot B = B \cdot A$ $A + B = B + A$
- 7 - ASSOCIATIVIDADE
 - $A \cdot B \cdot C = (A \cdot B) \cdot C = A \cdot (B \cdot C)$
 - $A + B + C = (A + B) + C = A + (B + C)$
- 8 - LEIS DE DeMORGAN
 - $\overline{A \cdot B} = \bar{A} + \bar{B}$ $\overline{A + B} = \bar{A} \cdot \bar{B}$

23

Sistemas Digitais

TEOREMAS

- 9 - DISTRIBUTIVIDADE
 - $A \cdot (B + C) = A \cdot B + A \cdot C$
 - $A + BC = (A + B) \cdot (A + C)$
- 10 - ABSORÇÃO
 - $A + AB = A$ $A(A + B) = A$
- 11 -
 - $A + A\bar{B} = A$ $(A + B) \cdot (A + \bar{B}) = A$
- 12 -
 - $A + \bar{A}B = A + B$ $A \cdot (\bar{A} + B) = A \cdot B$
- 13 - TEOREMA DO TERMO INCLUÍDO
 - $AB + \bar{A}C + BC = AB + \bar{A}C$
 - $(A + B)(\bar{A} + C)(B + C) = (\bar{A} + B)(A + C)$

24

DEMONSTRAÇÕES

Sistemas Digitais

- **USANDO TABELAS DE VERDADE**
 - Demonstra-se para **TODOS** os casos possíveis.
 - Tabela de verdade das funções **AND** e **OR**

A	B	S = A.B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	S = A+B
0	0	0
0	1	1
1	0	1
1	1	1

25

EXEMPLO:

Sistemas Digitais

- Provar que $A \cdot (\bar{A} + B) = A \cdot B$

A	B	\bar{A}	$\bar{A} + B$	$A \cdot (\bar{A} + B)$	$A \cdot B$
0	0	1	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	1	0	1	1	1

26

Funções de 2 variáveis

Sistemas Digitais

- **Quantas funções existem de 2 variáveis ?**
 - É um número finito.
 - 2 variáveis \Rightarrow 4 combinações de entrada $\Rightarrow 2^4=16$ funções
 - 3 delas decorrem imediatamente da definição da álgebra
 - \rightarrow AND (E, \cdot)
 - \rightarrow OR (OU, +)
 - \rightarrow NOT (NEG, $\bar{\quad}$)
 - Há outras funções que são muito usadas: XOR, NAND, NOR
- **Implementação física**
 - Sistemas mecânicos (alavancas, rodas dentadas)
 - Sistemas hidráulicos (usados em certos ambientes perigosos)
 - Sistemas eléctricos (relés)
 - Sistemas electrónicos (transistores, diodos, circuitos integrados)
 - \rightarrow De longe o mais eficiente, logo mais usado !

27

Implementação FÍSICA da Álgebra de Boole

Montar circuitos ou máquinas que façam as operações previstas na álgebra de boole (and, or, not, ...)

28

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

- **PORTA "AND" C/ RELÉS**

Implementação física

Tabela de verdade

A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

Símbolo Lógico

29

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

- **PORTA "OU" C/ RELÉS**

Implementação física

Tabela de verdade

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Símbolo Lógico

30

Sistemas Digitais

Dep.Armas e Electrónica- Escola Naval
V.1.8 V.Lobo 2012

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

● PORTA "NOT" C/ RELÉS

Implementação física

Tabela de verdade

A	\bar{A}
0	1
1	0

Símbolo Lógico

31

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

● PORTA "NAND" C/ RELÉS

Implementação física

Tabela de verdade

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Símbolo Lógico

32

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

● PORTA "NOR" C/ RELÉS

Implementação física

Tabela de verdade

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Símbolo Lógico

33

REALIZAÇÃO FÍSICA COM INTERRUPTORES

Sistemas Digitais

● PORTA "XOR" C/ RELÉS

Implementação física

Tabela de verdade

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Símbolo Lógico

34

Outras simbologias

Sistemas Digitais

● Norma ANSI Y.32.14

- Simplifica a representação das portas lógicas
- É menos "bonita", mas mais eficiente

AND

NAND

OR

NOR

XOR

XNOR

Identidade

NOT

35

Exercícios

Sistemas Digitais

● Problema do alarme de segurança

- Suponha que existem dois sensores de incêndio, e uma lâmpada que deverá acender quando um deles for activado. Projecte o circuito que actua sobre a lâmpada.

● Problema da segurança do navio


- Suponha num dado navio existem 4 pontos onde devem estar sentinelas quando o navio está fundeado: dois em cada bordo, um na alheta, outro na amura. Em cada um desses pontos está um sensor que envia um sinal 1 quando aí se encontra um sentinela, e 0 em caso contrário. Na câmara de oficiais deverão existir duas lâmpadas: uma amarela, outra vermelha. A vermelha deverá acender sempre que não há qualquer sentinela num dos bordos. A amarela deverá acender quando há apenas 2 sentinelas nos seus postos. Projecte o circuito que resolve este problema.

36

Exercícios

Sistemas Digitais

- **PROBLEMA DO SEMÁFORO "POR PEDIDO"**
 - Imagine que num dado local existe um estrangulamento numa estrada onde só passa um automóvel. Existem uns sensores para indicar que há um automóvel em cada lado do estrangulamento, e dois semáforos (verde/vermelho) que controlam o acesso a essa área. Se apenas houver automóveis de um dos lados, o semáforo deverá deixar passar esses automóveis. Se não houver automóveis em qualquer dos lados, os semáforos deverão estar ambos vermelhos. Caso contrário, o semáforo deverá estar verde para apenas um dos lados (à sua escolha).



37

SUFICIÊNCIA DO NAND

Sistemas Digitais

- **Quantas portas diferentes são necessárias para gerar uma função booleana ?**
 - A álgebra é definida com três operações (que por definição geram todas as funções possíveis):
→ AND, OR, NOT
 - Se eu conseguir realizar essas funções com uma só gate, poderei gerar qualquer outra função com essa gate
- **Suficiência do NAND**
 - NOT(A) = A NAND A
 - A AND B = (A NAND B) NAND (A NAND B)
 - A OR B = (A NAND A) NAND (B NAND B)

38

REALIZAÇÕES FÍSICAS

Sistemas Digitais

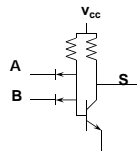
- **FAMÍLIAS LÓGICAS**
 - Permitem ligações directas entre as diversas portas lógicas
 - Exemplos: interruptores, relés, sistema mecânico e hidráulico
- **FAMÍLIAS ELECTRÓNICAS**
 - DTL, RTL
→ Fácil compreensão
 - ECL
→ Muito rápida, consome bastante
 - CMOS
→ Consumo muito baixo, tolerância a diversos níveis de tensão, grande integração (integrados da família 4000)
 - I²L
→ Mais uma alternativa...
 - TTL
→ Barato, simples de usar, compromisso bastante bom de características. É a mais usada (integrados da família 74xx, 54xx)

39

DTL

Sistemas Digitais

- **Diode-Transistor Logic**
 - Usa diodos e transistores
 - Exemplo: gate NAND
 - Hipótese 1: A=0v ou B=0v
→ Os diodos conduzem
→ A tensão na base do transistor é aprox. =0v
→ O transistor não conduz
→ A resistência de saída faz de pull-up: S=5v
 - Hipótese 2: A=B=1
→ Os diodos não conduzem
→ A resistência de entrada faz com que a tensão na base do transistor seja aproximadamente = 5v
→ O transistor conduz
→ A tensão de saída é aprox. =0v

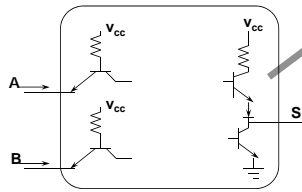


40

TTL

Sistemas Digitais

- **Transistor-Transistor Logic**
 - Transistores de junção bipolar
 - Vamos estudar apenas
→ Andar de entrada (diodos)
→ Andar de saída (totem-pole)



Andar de saída

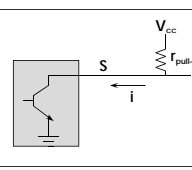
T1	T2	Saída
ON	OFF	V _{cc}
OFF	ON	GND
OFF	OFF	Tri-State
ON	ON	Bumm!

41

TTL

Sistemas Digitais

- **Gates Open-Collector**
 - O andar de saída só tem um transistor (ligado à massa)
 - A gate pode forçar o valor lógico ZERO
 - Tem que haver uma resistência externa de PULL-UP para forçar o nível lógico 1
 - Posso implementar um WIRED-AND, ligando várias saídas O.C.



TRANSISTOR LIGADO
A tensão de saída é 0, e a sendo a corrente
 $I = V_{cc} / R_{pull-up}$

TRANSISTOR DESLIGADO
A corrente é 0, logo a tensão de saída é V_{cc}

42

Sistemas Digitais

Dep.Armas e Electrónica- Escola Naval
V.1.8 V.Lobo 2012

CARACTERÍSTICAS

Sistemas Digitais

- **TTL**
 - FACILIDADE DE FABRICO, E DISPONIBILIDADE
 - ROBUSTEZ E FIABILIDADE
 - BAIXO CUSTO
 - CONSUMO MODERADO (LOGO DISSIPACÃO MODERADA)
 - FAMÍLIA 74xxx e 54xx
 - 54xx tem especificações militares: grande amplitude de temperaturas/humidade/vibração, distribuição otimizada dos pinos
 - VARIACÕES 74S , 74LS , 74L , 74H (consumo, velocidade)
- **CMOS, NMOS e PMOS**
 - TRANSISTORES DE EFEITO DE CAMPO
 - CONSUMO MUITO BOM
 - LENTIDÃO , E PROBLEMAS C/ ESTÁTICA
 - MAIOR FLEXIBILIDADE NOS NÍVEIS DE TENSÃO
 - FAMÍLIA 40xx

43

CARACTERÍSTICAS

Sistemas Digitais

- **FAN-OUT**
 - Nº de portas que podem ser ligadas à saída
 - Pode ser especificado em número de gates que consegue alimentar (da mesma família lógica) ou em corrente máxima de saída (em mA).
- **FAN-IN**
 - Corrente que injecta/consome na entrada
- **MARGEM DE RUIDO**
 - Tolerância entre níveis
 - 0 lógico não é 0v

Nota:
O que é ruído ?
Quais os seus efeitos ?
Quais são as fontes de ruído ?
Como pode ser diminuído ?

44

CARACTERÍSTICAS

Sistemas Digitais

- **FUNÇÃO DE TRANSFERÊNCIA**
 - A transição de 0 lógico para 1 lógico não é perfeita
 - Exemplo: gate NOT
- **TEMPO DE PROPAGAÇÃO**
 - Uma gate leva um certo tempo até que as saídas reflectam o estado das entradas
 - O tempo de propagação quando as saídas têm que passar de 0 para 1 é normalmente diferente de 1 para 0.
- **DISSIPACÃO**
 - As gates consomem corrente que provoca aquecimento
 - O aquecimento é normalmente proporcional à velocidade de processamento

GATE NOT

45

DISPLAYS

Sistemas Digitais

- **Quanto à tecnologia física**
 - Indicadores de descarga de gás
 - são válvulas
 - Leds - diodos emissores de luz
 - Baixo consumo
 - Interface muito simples
 - Grande variedade
 - Cristais Líquidos (lcd)
 - Consumo muitíssimo baixo
 - Mudanças na polarização (provocados pela aplicação de campo eléctrico) fazem com que a luz não seja reflectida
- **Quanto à disposição gráfica**
 - Displays de 7 segmentos
 - Matriz de pontos

46

FUNÇÕES BOOLEANAS

Sistemas Digitais

- **S = F (A) S = F (A,B,C, ...)**
 - Onde A,B,C... podem assumir os valores 0 e 1
- **PARA UM DADO NÚMERO DE VARIÁVEIS, O Nº DE FUNÇÕES POSSÍVEIS É LIMITADO**
 - Exemplo: FUNÇÕES DE 1 VARIÁVEL:

FUNÇÃO	ENTRADAS		DESIGNAÇÃO	EXPRESSÃO DE BOOLE
	0	1		
S0	0	0	Zero	0
S1	0	1	Igualdade	A
S2	1	0	Negação	!A (ou -A)
S3	1	1	Identidade	1

47

FUNÇÃO DUAL E COMPLEMENTO

Sistemas Digitais

- **FUNÇÃO DUAL**
 - G é função dual de F sse $G(A) = (F(A^*))^*$
 - (X^* é o dual de X se em X trocar 1 por 0, e vice-versa)
 - Exemplos
 - O dual da função AND é a função OR
 - O dual da negação é a própria negação
- **FUNÇÃO COMPLEMENTO**
 - G é função complemento de F sse $G(A) = !F(A)$
 - O complemento da função AND é a função NAND
 - O complemento da negação é a igualdade

48

FUNÇÕES DE 2 VARIÁVEIS

Sistemas Digitais

FUNÇÃO	ENIRADAS 00,01,10,11	DESIGNAÇÃO	EXPRESSÃO DE BOOLE	NOTAÇÃO	DUAL	COMPLE MENTO
S0	0 0 0 0	Zero	0		15	15
S1	0 0 0 1	And	A.B	A.B	7	14
S2	0 0 1 0	Inibição ou Nix Igludade	A.B*		11	13
S3	0 0 1 1		A		3	12
S4	0 1 0 0	Inibição ou Nix Igludade	A*.B		13	11
S5	0 1 0 1		B		5	10
S6	0 1 1 0	Or Exclusivo ou Dilema	A*.B+A.B*	A ⊕ B	9	9
S7	0 1 1 1	Or (Inclusivo)	A+B	A+B	1	8
S8	1 0 0 0	Nor ou Função Dagger	(A+B)*	$\overline{A+B}$	14	7
S9	1 0 0 1	Equivalência	A.B+A*.B*	$A \equiv B$	6	6
S10	1 0 1 0	Not (Negação)	B*	10*	5	5
S11	1 0 1 1	Implicação Material	A+B*	$B \Rightarrow A$	2	4
S12	1 1 0 0	Not (Negação)	A*		12	3
S13	1 1 0 1	Implicação Material	A*.B	$\overline{A} \Rightarrow B$	4	2
S14	1 1 1 0	Nand ou Função Stroke	(A.B)*	A . B	8	1
S15	1 1 1 1	Unidade ou Identidade	1		0	0

49

FORMAS CANÓNICAS

Sistemas Digitais

- Como identificar de forma unívoca e normalizada uma dada função?
 - Expressões analíticas podem ter várias formas
 - Tabelas de verdade são muito extensas
 - Formas canónicas: são a solução ideal
 - A tabela de verdade tem na coluna de resultados 0 ou 1
 - Posso identificar a função dizendo que entradas da tabela de verdade são 1 (ou 0)
 - A tabela de verdade tem que ter as entradas por uma determinada ordem

$$A.B + \overline{A}.B = A.B + !A.B = A \oplus B$$

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

As linhas 1 e 2 têm 1

F_{1,2}

50

FORMAS CANÓNICAS

Sistemas Digitais

- Como identificar as linhas da tabela de verdade ?
 - Cada linha corresponde a um produto de todas as variáveis
- MINTERMOS
 - Produtos que englobam todas as variáveis independentes
 - Correspondem às linhas da tabela de verdade, se esta for escrita de modo a que as variáveis formem o código binário
 - São numeradas, atribuindo 0 às variáveis negadas, e 1 às afirmadas
- MAXTERMOS
 - Somatórios que englobam todas as variáveis independentes
 - Podem-se obter a partir dos mintermos, e vice-versa
 - $M_i = m_2^{n-1-i}$

51

FORMAS CANÓNICAS

Sistemas Digitais

- 1ª. FORMA CANÓNICA
 - Soma de mintermos
 - Exemplo: função XOR
 - $XOR(A,B) = A.B + !A.B = m_1 + m_2 = \Sigma(1,2)$
 - Problemas
 - Qual a tabela de verdade da função de 3 variáveis $\Sigma(0,5,7)$?
 - Qual a 1ª forma canónica da função OR de 3 variáveis
- 2ª. FORMA CANÓNICA
 - Produto de maxtermos
 - Exemplo: função XOR
 - $XOR(A,B) = (!A+!B).(A+B) = M_0 . M_3 = \Pi(0,3)$
 - Problemas
 - Qual a tabela de verdade da função de 3 variáveis $\Pi(0,5,7)$?
 - Qual a 2ª forma canónica da função OR de 3 variáveis

52

RESOLUÇÃO DE PROBLEMAS

Sistemas Digitais

- 1) OBTENÇÃO DE UMA FUNÇÃO QUE RESOLVA O PROBLEMA POSTO
 - Métodos analíticos
 - Especificar o problema numa tabela de verdade
 - Obter os mintermos
- 2) SIMPLIFICAR A EXPRESSÃO
 - Métodos analíticos
 - Mapas de Karnaugh
- 3) IMPLEMENTAR O CIRCUITO
 - Escolher os integrados que implementam as gates
 - Pode ser necessário alterar a função obtida em 2 para minimizar o número de integrados usado
 - Desenhar o logigrama (com pinout) do circuito

53

Exemplo

Sistemas Digitais

- Passo 1 para o problema dos vigias do navio:
 - Método analítico: $L = a.b.!c.d + a.!b.c.d + !a.b.c.d + \dots$
 - Tabela de verdade:

a	b	c	d	A
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Mintermos:
0,1,2,3,4,6,8,12

$A(a,b,c,d) = \Sigma(0,1,2,3,4,6,8,12)$

$A(a,b,c,d) = !a.b.c.d + !a...$

54

MAPAS DE KARNAUGH

Sistemas Digitais

- Um mapa de karnaugh é um modo de escrever a tabela de verdade
- Cada quadrícula tem apenas 1 bit diferente dos vizinhos (distância de Hamming=1)

55

MAPAS DE KARNAUGH

Sistemas Digitais

- Método gráfico, baseado nos diagramas de Venn, que permite detectar adjacências
 - 1) Escrever o mapa usando código reflectido, de modo a que 2 quadrículas contíguas diferem em apenas 1 byte.
 - 2) Cada quadrado corresponde a uma linha da tabela de verdade => corresponde a um mintermo da expressão se for 1
 - 3) Como na tabela 2 quadrados contíguos diferem apenas numa das variáveis, podemos escrevê-los como $\Pi x_i y_j$ e $\Pi x_i \bar{y}_j$
 - 4) Se dois quadrados contíguos forem 1, podemos representá-los como $\Pi x_i y_j + \Pi x_i \bar{y}_j = \Pi x_i (y_j + \bar{y}_j) = \Pi x_i$, de onde se conclui que podemos ignorar a variável que troca de valor
- REGRA:
 - 1) Formar quadrados ou rectângulos com 2^m quadrículas
 - 2) Pôr na expressão só as variáveis que se mantêm constantes

56

MAPAS DE KARNAUGH

Sistemas Digitais

- Os grupos resultantes da junção de mintermos chamam-se **IMPLICANTES**
 - Implicante PRIMO
 - Implicante que não pode ser mais alargado
 - Implicante ESSENCIAL
 - Implicante que seja o único (dos primos) que "cobre" um dado mintermo

Problemas:

- Vigias
- Descodificador de 7 Segmentos para BCD
- Semáforos
- Segurança para as portas da cidadela

57

MAPAS DE KARNAUGH

Sistemas Digitais

- Indeterminações
 - Correspondem a casos onde "tanto faz" que a resposta seja 1 ou 0 (pode por exemplo ser uma combinação de entrada que nunca ocorre)
 - Representam-se nos mapas de Karnaugh por X
 - Podemos simplificar os X como 1 ou como 0, conforme nos dê mais jeito
 - Exemplo: descodificador de 7 segmentos BCD (traço do meio)

Alguns X são interpretados como 1 outros como 0

58