Introdução aos Sistemas Digitais

Aula 1

Apresentação da disciplina
Introdução aos sistemas digitais
Sistemas de numeração

Apresentação da disciplina

Área científica: Arquitetura de Sistemas Computacionais

Cursos: Mestrado Integrado em Engenharia de Computadores e Telemática, Mestrado Integrado em Engenharia Eletrónica e de Telecomunicações

Escolaridade semanal: 2 horas de aulas teórico-práticas; 2 horas de aulas práticas

Créditos ECTS: 6

Código: 40332

O número de créditos ECTS, atribuído a uma disciplina, **não** indica quantas horas de aulas vão ter. Em vez disso, indica o número de horas espetável que **devem** estudar para esta disciplina.

1 ECTS = 25-30 horas de estudo. 6 ECTS = 150-180 horas de estudo.
O semestre tem ~15 semanas => devem estudar pelo menos 10 horas por semana.

Estas horas incluem: aulas presenciais, leitura de livros, resolução de exercícios, estudo para testes e exames, etc.

Avaliação

Nota final = 0.6 × nota teórica + 0.4 × nota pratica

Nota teórica = nota obtida no exame escrito realizado na época de exames

Nota prática obtém-se através da avaliação do tipo "contínuo", resultante de 5 momentos de avaliação:

- Resolução de (pelo menos 2) problemas durante as aulas - 40%
- Um teste de avaliação realizado na última aula prática - 40%
- Trabalho de casa - 10%
- Qualidade da participação nas aulas - 10%

Avaliação (cont.)

Os alunos repetentes que tenham obtido classificação positiva na componente prática da disciplina no ano letivo de 2016/2017 em época normal mantêm este ano e caso assim o pretendam a sua nota nessa componente de avaliação.

Os alunos que se encontrem nesta situação e que se tenham inscrito, através do PACO, numa das turmas práticas, perdem automaticamente a nota prática obtida anteriormente.

A aprovação à disciplina implica uma avaliação global superior ou igual a 9.5 valores sendo que em nenhuma das componentes (teórica e prática) a nota correspondente (arredondada à décima) pode ser inferior a 7.0 valores.

Avaliação (cont.)

Não haverá registo de faltas nas aulas TP.

Em regime ordinário as aulas práticas são de frequência obrigatória.

Atendendo ao atual regulamento de estudos da UA, todos os estudantes que, não usufruindo do estatuto de trabalhador-estudante no ano letivo corrente, faltem injustificadamente a mais de 3 aulas práticas reprovam automaticamente à disciplina ficando impedidos de apresentar-se a qualquer prova da mesma durante o corrente ano letivo.

A justificação formal das faltas deverá ser feita junto da Secretaria do DETI dentro do prazo regulamentar. Paralelamente e tão cedo quanto possível o aluno deverá enviar cópia da justificação ao respetivo docente da prática.

Avaliação (cont.)

Dado o regime contínuo da avaliação na componente prática não haverá, em época normal, exame global final a esta componente a não ser para os estudantes trabalhadores que comprovadamente não tenham frequentado 80% das aulas práticas. Este exame, apenas para trabalhadores estudantes, será de tipo laboratorial, terá duração de 90 minutos e decorrerá no mesmo dia do exame da componente teórica.

Os alunos com o estatuto de trabalhador-estudante que pretendam ser avaliados em regime de avaliação contínua na componente prática da disciplina, deverão declará-lo por escrito, entregando a respetiva declaração, o mais tardar até à segunda aula prática (texto da declaração).

Aulas práticas

Aconselha-se que os alunos tenham um caderno de registo (*logbook)* das atividades desenvolvidas destinado exclusivamente a esta disciplina.

Este caderno permite sistematizar o estudo e facilita a preparação para testes e o exame final.

Docentes

Regente: Augusto Silva

Aulas teórico-práticas: Augusto Silva, Guilherme Campos, Iouliia Skliarova

Aulas práticas: António Navarro, António Pereira, Arnaldo Oliveira, Augusto Silva, Filipe Silva, Iouliia Skliarova

Bibliografia recomendada

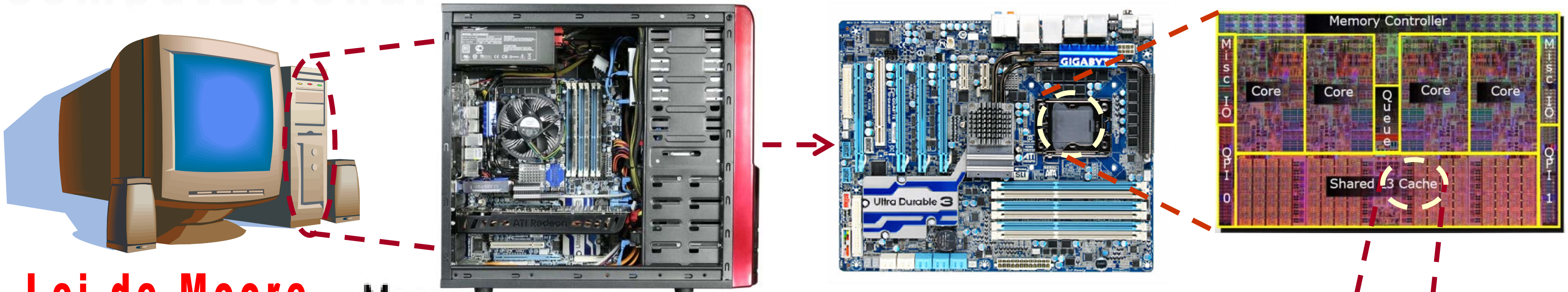
- J.F. Wakerly, Digital design: Principles and practices, 4th edition, 2006, Prentice-Hall;
- M. Mano, Digital design, 4th edition, 2006, Prentice-Hall;
- M. Dias, Sistemas Digitais – Princípios e Prática, 2ª ed., 2011, FCA;
- Z. Kohavi, Niraj K. Jha, Switching and Finite Automata Theory, Cambridge Univ. Press, 2009;
- C. Sêrro, Sistemas Digitais: fundamentos algébricos, IST Press, 2003.

Página da disciplina

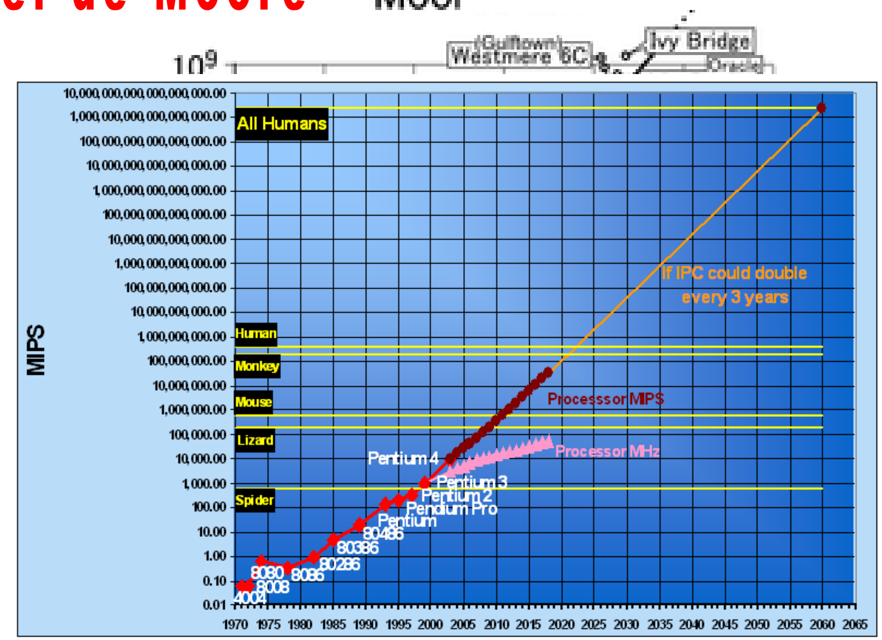
http://elearning.ua.pt/

Consultar o guião da disciplina

Como construir sistemas complexos?



Lei de Moore



Função

+

Tecnologia de implementação

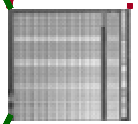


Imagem de http://pc.watch.impress.co.jp/docs/column/

Imagem de http://www.captec-group.com/

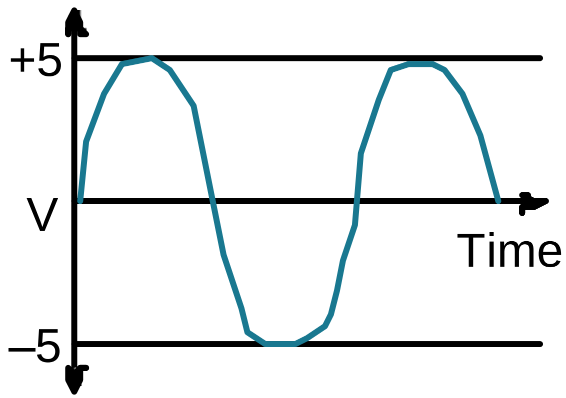
ieeta

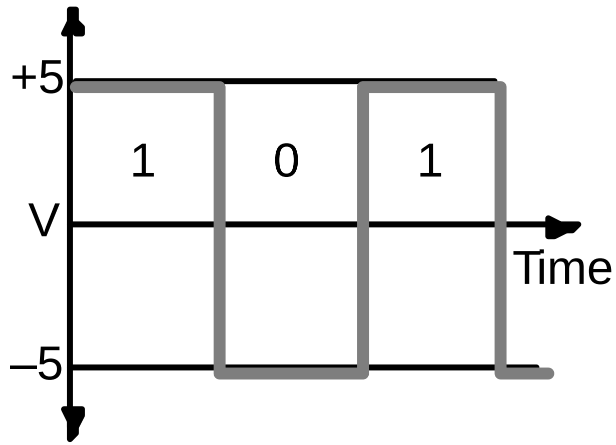
Sistemas digitais

Sistemas analógicos processam sinais que variam no tempo e podem tomar qualquer valor dentro de uma gama.

Em sistemas digitais sinais são modelados como se tomassem sempre um dos (dois) valores discretos.

- reprodução de resultados;
- facilidade de projeto;
- programabilidade;
- desempenho;
- precisão;





Álgebra Booleana

Sistemas digitais binários usam dois valores discretos:

0-1 LOW-HIGH desligado-ligado FALSE-TRUE 0 volts-5 volts

A álgebra Booleana fornece a base matemática rigorosa baseada em lógica.

Variáveis – sinais lógicos

Valores – 0 e 1 (se uma expressão lógica é falsa, então toma o valor 0; caso seja verdadeira, então toma o valor 1)

Operações – AND, OR, NOT

x	y	x and y
		xy
		$x \cdot y$
		$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x or y
		$x+y$
		$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

x	not x
	$\overline{x}$
	x'
0	1
1	0

Realidade física

Componentes eletrónicos físicos, usados para construir sistemas digitais, são contínuos, não discretos.

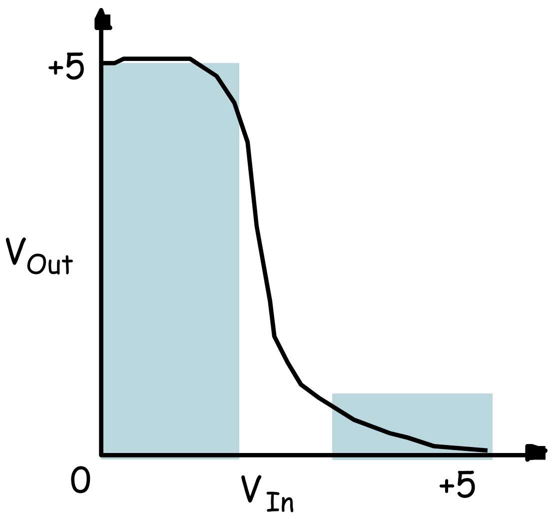
Consequentemente, as transições de estado lógico não são instantâneas, podendo-se observar valores intermédios de curta duração.

Sendo assim, a álgebra Booleana descreve o comportamento de sistemas digitais em regime estacionário e não reflete o aspeto dinâmico correspondente ao seu comportamento variante no tempo.

Exemplo:

Comportamento de uma porta NOT

Entrada varia de 0V a 5V.
Saída mantém-se a 5V para uma certa gama de valores da entrada, e depois varia rapidamente mas não instantaneamente



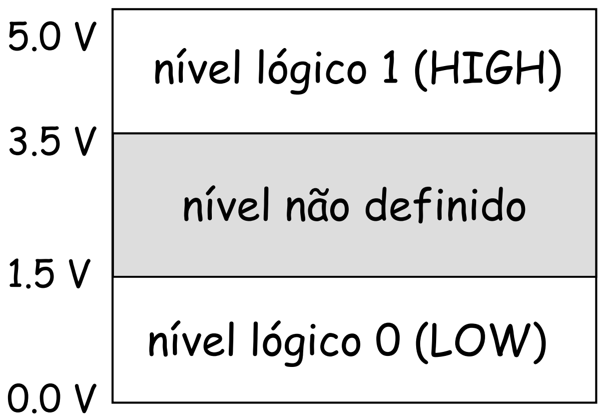
Valores lógicos

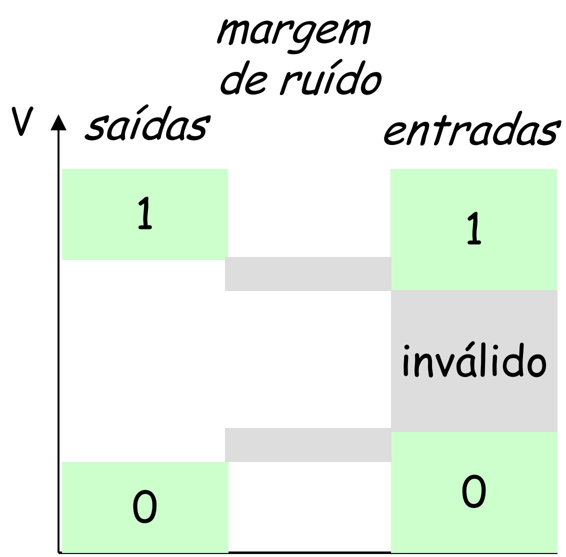
Bit (**bi**nary digi**t**) – dígito que representa valores lógicos 0 e 1.

Em várias tecnologias digitais os bits são representados com a ajuda de fenómenos físicos diferentes.

Exemplo:

Tecnologia CMOS (*Complementary metal-oxide semiconductor*)





Sistemas de numeração: notação

Sistemas digitais são construídos de circuitos que processam dígitos binários.

Problemas reais quase nunca são formulados em termos de números binários.

Num sistema de numeração posicional à posição de cada dígito é atribuído um peso. Para uma base $r \geq 2$, um dígito na posição i tem o peso r^i.

Notação:

$r \geq 2$ – base

$d_i \in \{0,...,r\text{-}1\}$ - conjunto de símbolos (alfabeto)

$p + n$ – número de símbolos (p – parte inteira, n – parte fracionária)

Um número D cuja parte inteira inclui p dígitos e a parte fracionária – n dígitos pode ser representado como:

$$D = d_{p-1} d_{p-2} \ldots d_1 d_0 . d_{-1} d_{-2} \ldots d_{-n} = \sum_{i=-n}^{p-1} d_i * r^i$$

dígito mais significativo

dígito menos significativo

Sistemas de numeração: bases e alfabetos

base	alfabeto
2	0,1
8	0, 1,…, 7
10	0, 1,…, 9
16	0, 1,…, 9, A, B, C, D, E, F

Exemplos:

Sistema decimal

$$D = \sum_{i=-n}^{p-1} d_i * 10^i$$

$2007_{10} = 2*1000 + 0*100 + 0*10 + 7*1$

$19.85_{10} = 1*10 + 9*1 + 8*0.1 + 5*0.01$

Sistema binário

$$D = \sum_{i=-n}^{p-1} d_i * 2^i$$

$1100110_2 = 1*2^6 + 1*2^5 + 1*2^2 + 1*2^1 = 64 + 32 + 4 + 2 = 102_{10}$

$101.0011_2 = 1*2^2 + 1*2^0 + 1*2^{-3} + 1*2^{-4}$

bit mais significativo

bit menos significativo

Sistemas de numeração: exemplos

Exemplos:

Sistema octal

$$D = \sum_{i=-n}^{p-1} d_i * 8^i$$

$3577_8 = 3*8^3 + 5*8^2 + 7*8^1 + 7*8^0 = 1919_{10}$

$35.77_8 = 3*8^1 + 5*8^0 + 7*8^{-1} + 7*8^{-2}$

Sistema hexadecimal

$$D = \sum_{i=-n}^{p-1} d_i * 16^i$$

$2007_{16} = 2*16^3 + 7*16^0 = 8199_{10}$

$7D7_{16} = 7*16^2 + 13*16^1 + 7*16^0 = 2007_{10}$

$A.2C_{16} = 10*16^0 + 2*16^{-1} + 12*16^{-2}$

Correspondência entre sistemas de numeração

binário	decimal	octal	hexadecimal
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	8	10	8
1001	9	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Mudança de base: parte inteira

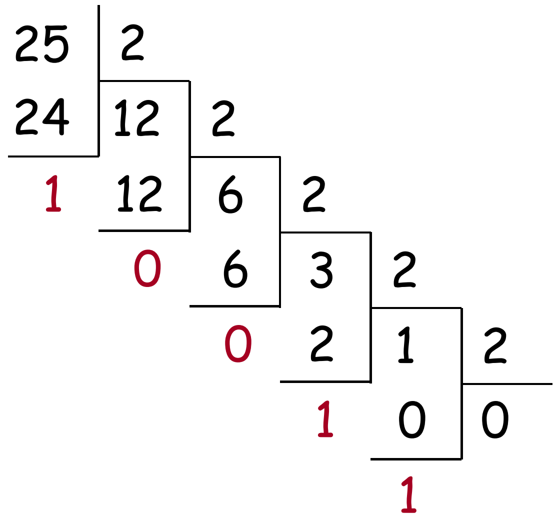
A conversão de um número decimal N inteiro para qualquer outra base r pode ser realizada através de divisões sucessivas do N por r até que o resultado da divisão se torne nulo.

Exemplos:

Conversão para binário

$25_{10} = ???_2$

$25_{10} = \quad\quad _2$

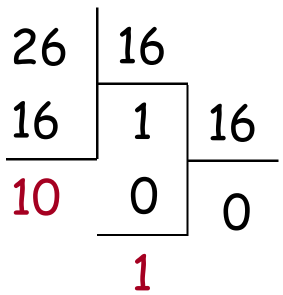


Conversão para hexadecimal

$26_{10} = ???_{16}$

$26_{10} = \quad\quad _{16}$

A



Mudança de base: parte fracionária

A conversão da parte fracionária F de um número decimal para qualquer outra base r pode ser realizada através de multiplicações sucessivas da F por r até que seja atingida a precisão desejada.

Exemplos:

Conversão para binário

$0.6875_{10} = ???_2$

$0.6875_{10} = 0. \quad _2$

```
  0.6875
       2
---------
  1.3750
       2
---------
   0.750
       2
---------
   1. 50
       2
---------
    1.00
```

Conversão para hexadecimal

$0.25_{10} = ???_{16}$

$0.25_{10} = 0. \quad _{16}$

```
    0.25
      16
---------
    4.00
```

Mudança de base: parte fracionária (cont.)

O número de dígitos significativos da parte fracionária deve ser consistente com os erros de representação nas bases inicial e final.

r_1 – base inicial

r_2 – base final

n_1 – número de dígitos fracionários na base inicial r_1

n_2 – número de dígitos fracionários na base final r_2

Para que a mudança de base não traga acréscimo de precisão:

$$n_2 = \lfloor n_1 * \log_{r_2} r_1 \rfloor$$

Exemplos:

$0.6875_{10} = ???_2$ $\quad 0.6875_{10} = 0.1011_2$ $\quad 0.6875_{10} = 0.1011000000000_2$

$A.2C_{16} = 10*16^0 + 2*16^{-1} + 12*16^{-2} = 10 + 0.125 + 0.046875 = 10.171875 = 10.17_{10}$

$101.0011_2 = 1*2^2 + 1*2^0 + 1*2^{-3} + 1*2^{-4} = 4 + 1 + 0.125 + 0.0625 = 5.1875 = 5.2_{10}$

Mudança de base: casos especiais

Quando é necessário converter um número da base r_1 para a base r_2 e se $r_1=r_2^x$, então cada dígito da base r_1 pode ser convertido diretamente para x dígitos em base r_2.

Exemplos:

Conversão de octal para binário

$753.6_8 = 111\ 101\ 011\ .\ 110_2$

$r_1 = 8,\ r_2 = 2,\ 8 = 2^3$

=> cada dígito octal pode ser representado por 3 dígitos binários

Conversão de hexadecimal para binário

$r_1 = 16,\ r_2 = 2,\ 16 = 2^4$

$A5.E_{16} = 1010\ 0101\ .\ 1110_2$

=> cada dígito hexadecimal pode ser representado por 4 dígitos binários

Mudança de base: casos especiais (cont.)

Problema inverso: quando é necessário converter um número da base r_2 para a base r_1 e se $r_1=r_2^x$, então cada subsequência disjunta de x dígitos em base r_2 origina um dígito na base r_1.

Exemplos:

Conversão de binário para octal

$1\ 101\ .\ 010_2 = 15\ .\ 2_8$

$r_1 = 8$, $r_2 = 2$, $8 = 2^3$

=> cada três dígitos binários correspondem a um dígito octal

Conversão de binário para hexadecimal

$r_1 = 16$, $r_2 = 2$, $16 = 2^4$

$110\ 0101\ 1100_2 = 65C_{16}$

=> cada quatro dígitos binários correspondem a um dígito hexadecimal

Exercícios

Explique a Lei de Moore.

Quais são vantagens de sistemas digitais comparando-os com sistemas analógicos?

Quando a saída de uma porta OR está a 0?

Quando a saída de uma porta AND está a 1?

Explique o que é *margem de ruído*?

Exercícios (cont.)

Converta os números seguintes para as bases 2, 8, 10 e 16.

$10111011001_2 = 2731_8 = 5D9_{16} = 1497_{10}$

$1234_8 = 001010011100_2 = 29C_{16} = 668_{10}$

$C0DE_{16} = 1100000011011110_2 = 140336_8 = 49374_{10}$

$108_{10} = 1101100_2 = 154_8 = 6C_{16}$

$15.46_{10} = 1111.011101_2 = 17.35_8 = F.7_{16}$