



# CIRCUITOS LÓGICOS

## APRESENTAÇÃO DA DISCIPLINA

Marco A. Zanata Alves

# PROFESSOR

**Luis Allan Künzle**

[www.inf.ufpr.br/kunzle](http://www.inf.ufpr.br/kunzle)

[kunzle@inf.ufpr.br](mailto:kunzle@inf.ufpr.br)

Gabinete 84 – Departamento de Informática

# PROCEDIMENTOS DIDÁTICOS

## Aulas Expositivas

- Slides + Quadro & Giz
- Laboratório (poucas vezes)

## Trabalhos Práticos

## Lista de Exercícios

# AVALIAÇÃO

2 Provas (P1 e P2)

1 Trabalho prático (TRAB)

$$\text{Média} = (0,30 \cdot P1) + (0,35 \cdot P2) + (0,35 \cdot \text{TRAB})$$

**Aprovado**  $\leftrightarrow$  **Média**  $\geq$  **7.0** & **Presença**  $\geq$  **75%**

*Direito ao Exame*  $\leftrightarrow$  *Média*  $\geq$  4.0 & *Presença*  $\geq$  75%

**Aprovado**  $\leftrightarrow$   $\frac{\text{Exame} + \text{Média}}{2} \geq 5.0$  & **Presença**  $\geq$  **75%**

# BIBLIOGRAFIA

## Básica

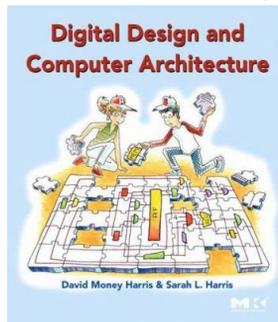
Digital Design and Computer Architecture

Edição: 1ª ou 2ª

Autores: David Money Harris,  
Sarah L. Harris

Editora: Morgan Kaufmann /  
Elsevier

Ano: 2007



## Auxiliar

Sistemas Digitais  
Fundamentos e Aplicações

Edição: 9ª

Autores: Thomas L. Floyd

Editora: Bookman

Ano: 2011



Slides no Moodle





# DO'S, DON'TS & MAYBES

# DO'S

Leia o livro texto.

Faça os exercícios LOGO, não apenas antes das provas.

Mande e-mails e peça ajuda.

Pergunte, participe.

Lembre-se: a aula é local de trabalho, seja ativo!

Chegue na hora

# DON'TS

## Evite perguntar ao colega.

- Atrapalha e pode ser fonte de informações erradas.
- Dirija perguntas diretamente ao professor.

## Atender (ou sair para atender) celular, comer em aula.

- Beber café pode (se a regra da sala permitir)

## Usar

- Celulares, smartphones, tablets, netbooks, notebooks, ultrabooks.

## Chegar atrasado.

- Se perde o raciocínio (meu e de seus colegas)
- Quando a aula for mais curta, eu liberarei antes

## Se, por obra do destino você chegar atrasado...

- Seja invisível.
- Assine a lista de chamada no final da aula (sua responsabilidade).

# MAYBES

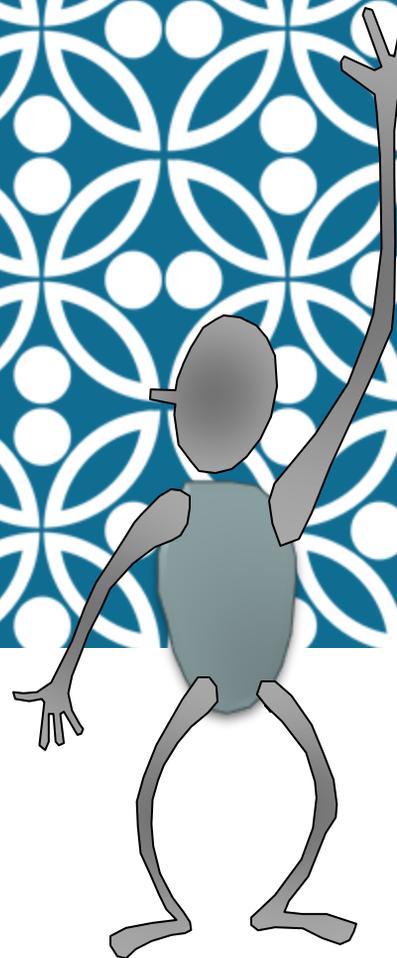
Dormir

Bocejar com a mão servindo de reparo

**Evite roncar**



**NÃO TENHA  
MEDO DE  
PERGUNTAR!!!**





# APRESENTAÇÃO DA DISCIPLINA



# APRESENTAÇÃO

O que é um computador digital?

# APRESENTAÇÃO

O que é um computador digital?

- Aquilo que computa, ou seja, que processa informação digital, ou seja, aquela que é **quantizada** ou **discreta** (ao invés de contínua)
- As informações podem ser representadas por números (que possuem dígitos).

Como um computador digital processa informação?

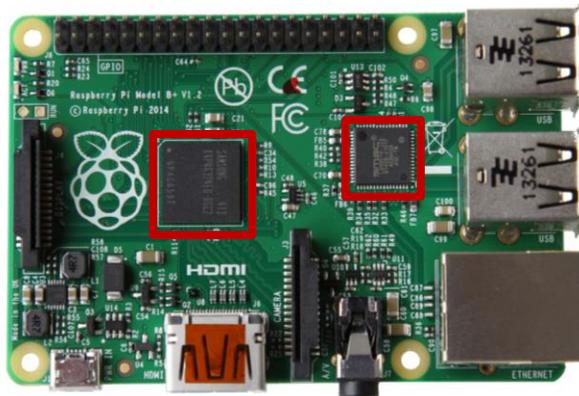
# APRESENTAÇÃO

O que é um computador digital?

- Aquilo que computa, ou seja, que processa informação digital, ou seja, aquela que é **quantizada** ou **discreta** (ao invés de contínua)
- As informações podem ser representadas por números (que possuem dígitos).

Como um computador digital processa informação?

- Por meio de circuitos digitais



# APRESENTAÇÃO

○ que são circuitos digitais?

Para que me interessa saber o que há em um computador?

# UM PROBLEMA...

Considere o seguinte programa:

```
public class prog 1 {  
    public static void main(String args[ ]) {  
        int i = 2147483647;  
        System.out.println(i);  
        i++;  
        System.out.println(i);  
    }  
}
```

**O que será impresso na tela?**

# UM PROBLEMA...

Considere o seguinte programa:

```
public class prog 1 {  
    public static void main(String args[ ]) {  
        int i = 2147483647;  
        System.out.println(i);  
        i++;  
        System.out.println(i);  
    }  
}
```

**O que será impresso na tela?**

2147483647

# UM PROBLEMA...

Considere o seguinte programa:

```
public class prog 1 {  
    public static void main(String args[ ]) {  
        int i = 2147483647;  
        System.out.println(i);  
        i++;  
        System.out.println(i);  
    }  
}
```

**O que será impresso na tela?**

2147483647  
-2147483648

# OUTRO PROBLEMA...

```
int m[][] = new int[7000][7000];
long start = System.currentTimeMillis();
for (int i=0; i < 7000; i++)
    for (int j=0; j < 7000; j++) {
        m[i][j] = 1;
    }
long stop = System.currentTimeMillis();
System.out.println("Tempo :" + (stop - start) +
"ms");
```

**Resultado (em um Atom 1.8GHz):**

# OUTRO PROBLEMA...

```
int m[][] = new int[7000][7000];
long start = System.currentTimeMillis();
for (int i=0; i < 7000; i++)
    for (int j=0; j < 7000; j++) {
        m[i][j] = 1;
    }
long stop = System.currentTimeMillis();
System.out.println("Tempo :" + (stop - start) +
"ms");
```

**Resultado (em um Atom 1.8GHz):**

**Tempo: 148ms**

# OUTRO PROBLEMA...

```
int m[][] = new int[7000][7000];
long start = System.currentTimeMillis();
for (int i=0; i < 7000; i++)
    for (int j=0; j < 7000; j++) {
        m[j][i] = 1; // Troquei j & i !!!
    }
long stop = System.currentTimeMillis();
System.out.println("Tempo :" + (stop - start) +
"ms");
```

**Resultado (em um Atom 1.8GHz):**

# OUTRO PROBLEMA...

```
int m[][] = new int[7000][7000];
long start = System.currentTimeMillis();
for (int i=0; i < 7000; i++)
    for (int j=0; j < 7000; j++) {
        m[j][i] = 1; // Troquei j & i !!!
    }
long stop = System.currentTimeMillis();
System.out.println("Tempo :" + (stop - start) +
"ms");
```

**Resultado (em um Atom 1.8GHz): 41x mais lento!**

**Tempo: 148ms**

**Tempo: 6117ms**

# UM DESASTRE

On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after lift-off (918K QuickTime movie).

The rocket was on its first voyage, after a decade of development costing \$7 billion. The destroyed rocket and its cargo were valued at \$500 million.

A board of inquiry investigated the causes of the explosion and in two weeks issued a report. It turned out that the cause of the failure was a software error in the inertial reference system.

Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer. The number was larger than 32,768, the largest integer storable in a 16 bit signed integer, and thus the conversion failed.



# UMA FALHA

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks and killed 28 soldiers.

It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors. Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by  $1/10$  to produce the time in seconds. This calculation was performed using a 24 bit fixed point register. In particular, the value  $1/10$ , which has a non-terminating binary expansion, was chopped at 24 bits after the radix point. The small chopping error, when multiplied by the large number giving the time in tenths of a second, lead to a significant error. Indeed, the Patriot battery had been up around 100 hours, and an easy calculation shows that the resulting time error due to the magnified chopping error was about 0.34 seconds.

A Scud travels at about 1,676 meters per second, and so travels more than half a kilometer in this time. This was far enough that the incoming Scud was outside the "range gate" that the Patriot tracked.

Ironically, the fact that the bad time calculation had been improved in some parts of the code, but not all, contributed to the problem, since it meant that the inaccuracies did not cancel.



# PLANEJAMENTO DAS DISCIPLINAS DE HARDWARE

## Circuitos (1º Semestre)

- 1.1) sistemas de numeração
- 1.2) Bits para representar sinais digitais
- 1.3) portas lógicas
- 1.4) circuitos com portas lógicas, somadores, muxes, seletores
- 1.5) latches e flip-flops
- 1.6) registradores e contadores

## Projetos (2º Semestre)

- 2.1) revisão de combinacionais - com VHDL
- 2.2) introdução a CMOS – temporização
- 2.3) revisão de sequenciais - com VHDL (setup, hold, skew)
- 2.4) assembly do MIPS
- 2.5) implementação do MIPS ciclo longo
- 2.6) memória
- 2.7) FPGAs e dispositivos programáveis

## Arquitetura (3º Semestre)

- 3.1) aritmética de inteiros e ponto flutuante
- 3.2) implementação do MIPS
  - 3.2.1) multiciclo
  - 3.2.2) pipeline
  - 3.2.3) superescalar (intro)
- 3.3) memória cache
- 3.4) memória virtual
- 3.5) periféricos e E/S
- 3.6) interrupções (hw)
- 3.7) intro ao processamento paralelo

## Software Básico (3º Semestre)

- 4.1) Unix como conjunto de abstrações, Bash como ferramenta de programação
- 4.2) representação de informação (números, ponto flutuante, ASCII)
- 4.3) assembly para adultos
- 4.4) esqueleto de um sistemas operacionais
- 4.5) compilação em fases
- 4.6) periféricos e interrupções (sw)



# ENTRANDO NO UNIVERSO DIGITAL

# DESVENDANDO MISTÉRIOS

Os comportamentos que vimos nos slides anteriores são devidos à forma como nossos computadores atuais são construídos.

**Quais as suas causas?**

**Podemos prevê-los?**

**Podemos evitá-los?**

# DESVENDANDO MISTÉRIOS

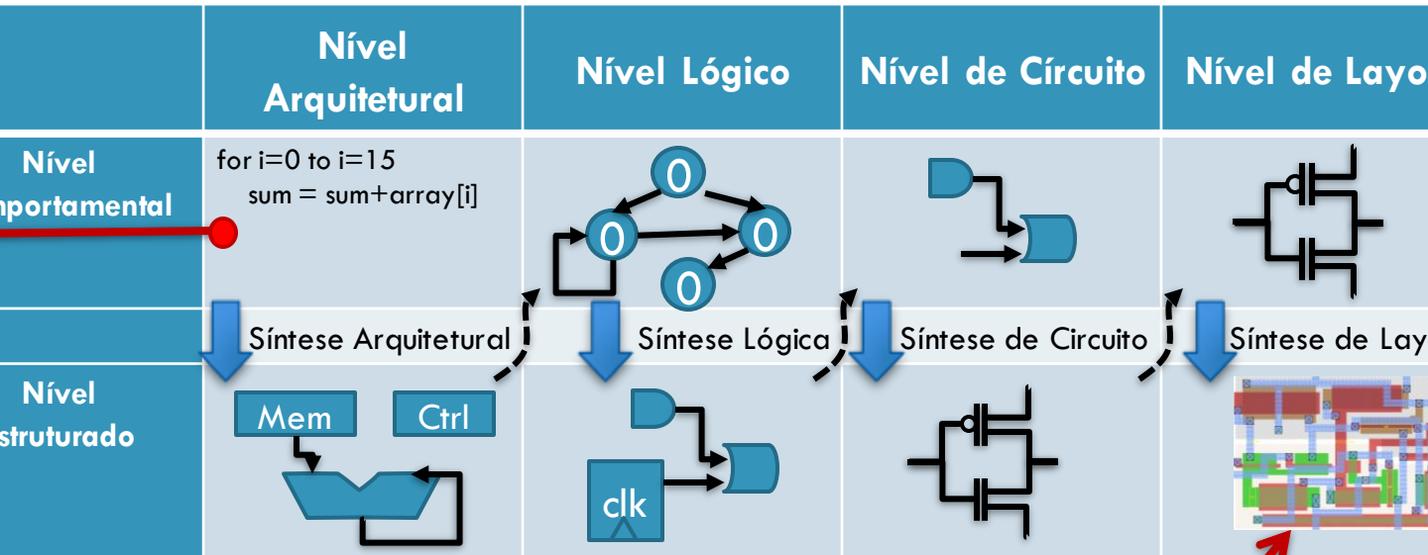
Para responder essas perguntas, precisamos saber a arquitetura do computador sendo usado.

Para sabermos arquitetura de computadores digitais, precisamos primeiro entender os seus blocos básicos, os circuitos digitais.

Para conhecermos circuitos digitais, precisamos voltar aos sistemas de numeração posicionais, que utilizam algarismos (dígitos).

# NÍVEIS DE ABSTRAÇÃO E SÍNTESE

## “A ARTE DE GERENCIAR COMPLEXIDADE”



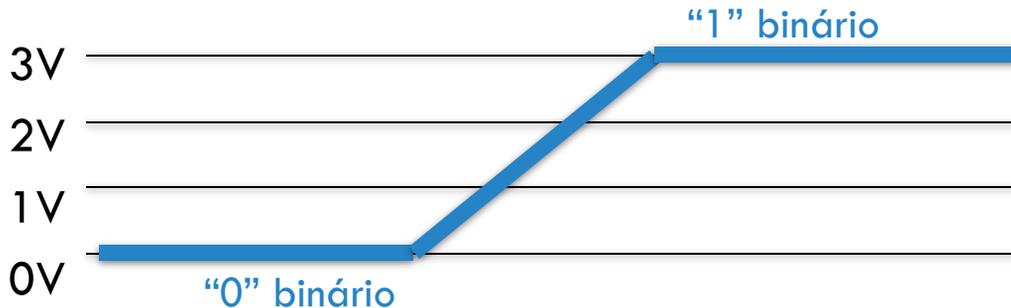
Compilação para silício não é um grande sucesso

# LÓGICA DIGITAL

Lógica digital utiliza a representação binária 0s e 1s

- 1 como ligado ou máxima tensão de alimentação (ex. 3V, 5V)
- 0 como desligado ou mínima tensão de alimentação (Ex. terra ou ground, -1V)

Cada projeto de circuito pode convencionar as tensões e seus respectivos valores



# SISTEMAS DIGITAIS

Uma vez que os sinais do mundo físico são analógicos, é necessários convertê-los para sinais digitais e vice-versa sempre que os sinais digitais tenham que interagir com os sinais do meio físico



# SINAIS DIGITAIS

Sequência discreta (descontínua) no tempo (definido apenas para determinados instantes)

- Representação aproximada do mundo real (analógico)
- Entretanto técnicas de processamento de sinais podem ser adotadas
- Procedimentos de compressão podem ser usados

# PRINCIPAIS PROPRIEDADES DA DISCRETIZAÇÃO

1. **Amostragem:** Discretização do sinal analógico original no tempo.
2. **Quantização:** Discretização da amplitude do sinal amostrado em níveis.
3. **Codificação:** Atribuição de códigos (geralmente binários) às amplitudes do sinal quantizado.

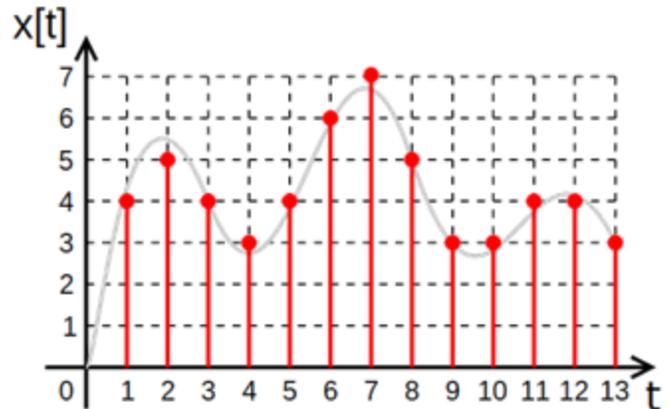
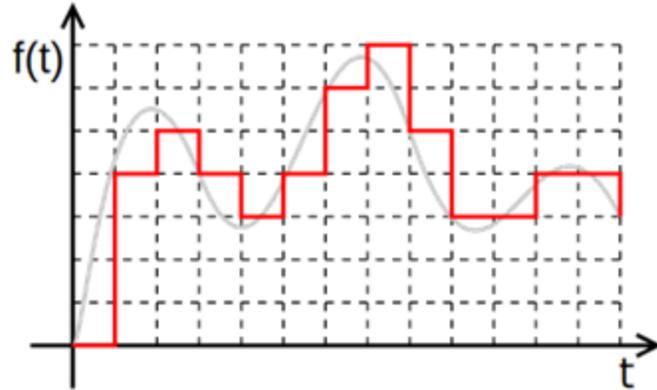
**Exemplo: MP3 1.35Mbps**

44.100Hz (44kHz) em

16bits (65.536 volumes)

stereo (2 canais)

Não confundir com bit rate = taxa de compressão

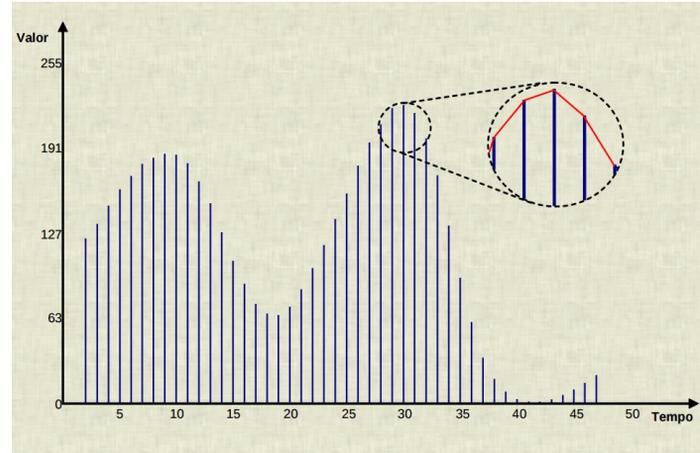
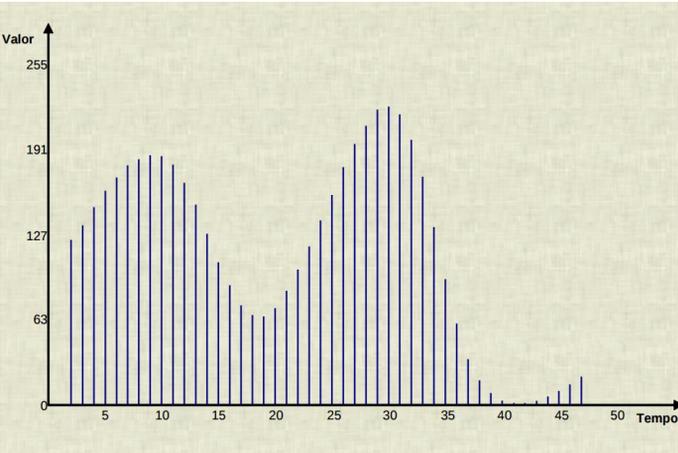


# PROCESSAMENTO

Quando comparado com sistemas analógicos

- Facilidade para armazenar
- Menor susceptibilidade a interferências
- Processamento simplificado

# CONVERSÃO ENTRE SISTEMA ANALÓGICO E DIGITAL (LINEARIZAÇÃO)



# CONVERSÃO ENTRE SISTEMA ANALÓGICO E DIGITAL

Um sinal digital consegue transportar toda a informação de um sinal analógico?

- Se consegue, como faz isto?
- Se não consegue, qual a limitação?

Como posso aumentar o espectro de frequências a ser capturada pelo mundo digital? Quais as consequências (memória, tempo, energia, ...)?