



**Universidade Federal de Santa Catarina**  
**Centro Tecnológico**  
Departamento de Informática e Estatística  
**Curso de Graduação em Ciências da Computação**



# **Sistemas Digitais**

**INE 5406**

## **Aula 1-P**

**Introdução ao Fluxo de Projeto com ferramentas de EDA.  
Introdução à Linguagem VHDL. Descrição de um somador  
completo (*full adder*) em VHDL e síntese com o Quartus II.**

**Prof. José Luís Güntzel**  
**[guntzel@inf.ufsc.br](mailto:guntzel@inf.ufsc.br)**

**Colaboração: Vinícius Livramento (Est. Docência 2010/1)**  
**[vini@inf.ufsc.br](mailto:vini@inf.ufsc.br)**

**[www.inf.ufsc.br/~guntzel/ine5406/ine5406.html](http://www.inf.ufsc.br/~guntzel/ine5406/ine5406.html)**

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Número de Transistores Integrados Processadores Intel (até 2004)

Microprocessor	Year of Introduction	Transistors
4004	1971	2,300
8008	1972	2,500
8080	1974	4,500
8086	1978	29,000
Intel286	1982	134,000
Intel386™ processor	1985	275,000
Intel486™ processor	1989	1,200,000
Intel® Pentium® processor	1993	3,100,000
Intel® Pentium® II processor	1997	7,500,000
Intel® Pentium® III processor	1999	9,500,000
Intel® Pentium® 4 processor	2000	42,000,000
Intel® Itanium® processor	2001	25,000,000
Intel® Itanium® 2 processor	2003	220,000,000
Intel® Itanium® 2 processor (9MB cache)	2004	592,000,000

Source: Intel Corporation.

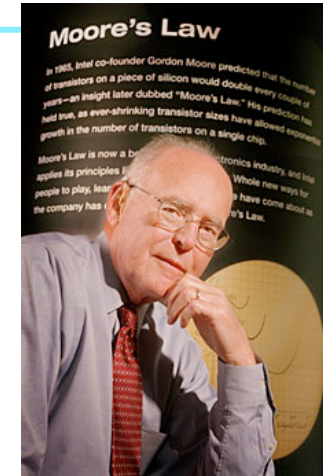
[http://www.intel.com/museum/archives/history\\_docs/mooreslaw.htm](http://www.intel.com/museum/archives/history_docs/mooreslaw.htm)

# Projeto de Sistemas Digitais com Ferramentas EDA

## ▶ A Lei de Moore

**“The number of transistors incorporated in a chip will approximately double every 24 months.”**

**Gordon Moore, Co-Founder Intel Co., 1965**



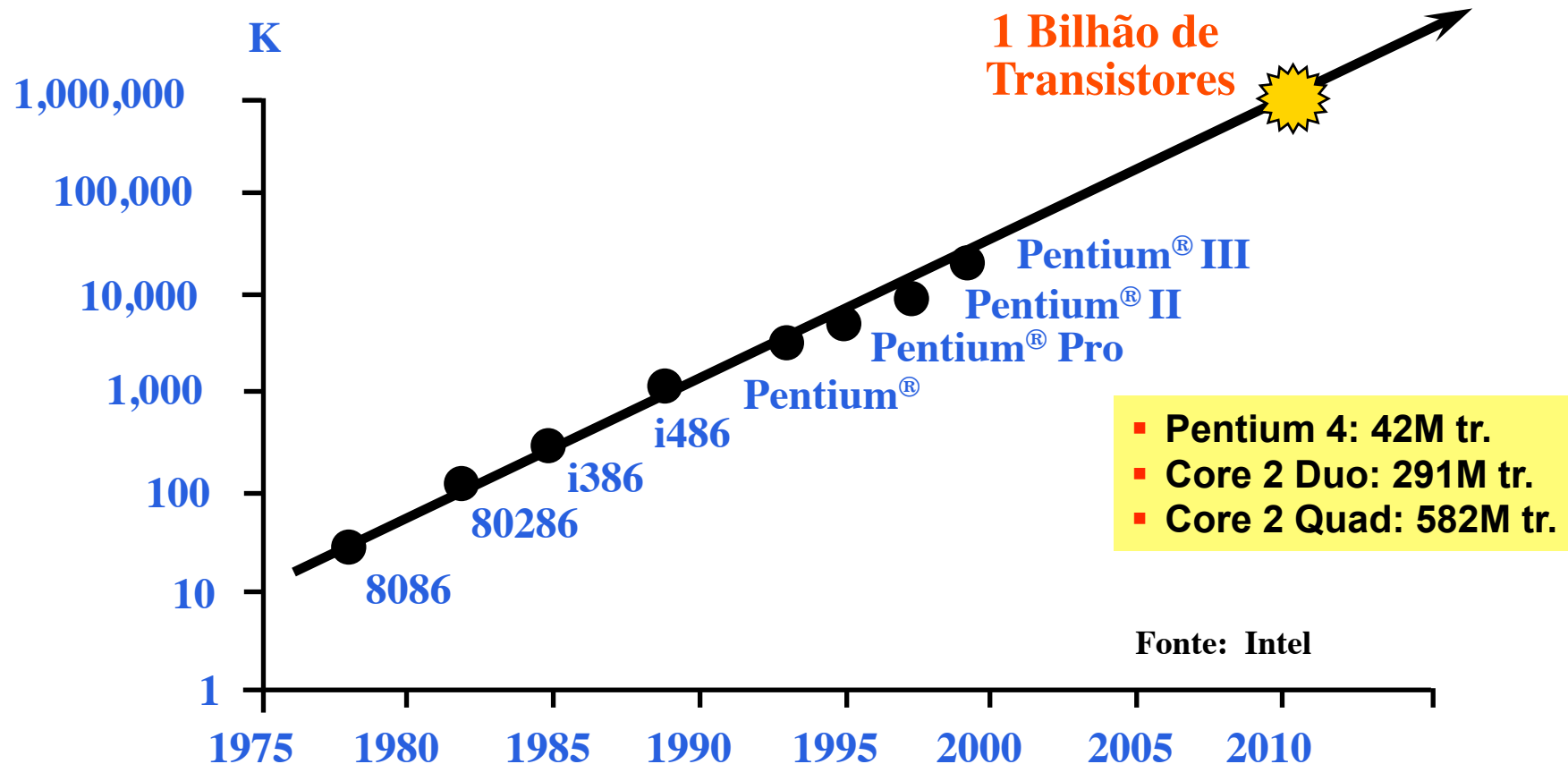
**Gordon E. Moore, Co-founder, Intel Corporation.**

Source: [http://www.intel.com/museum/archives/history\\_docs/mooreslaw.htm](http://www.intel.com/museum/archives/history_docs/mooreslaw.htm)

**Em 1965, Gordon Moore (co-fundador da Intel) previu que o número de transistores integrados por chip dobraria a cada 24 meses.**

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Número de Transistores Integrados

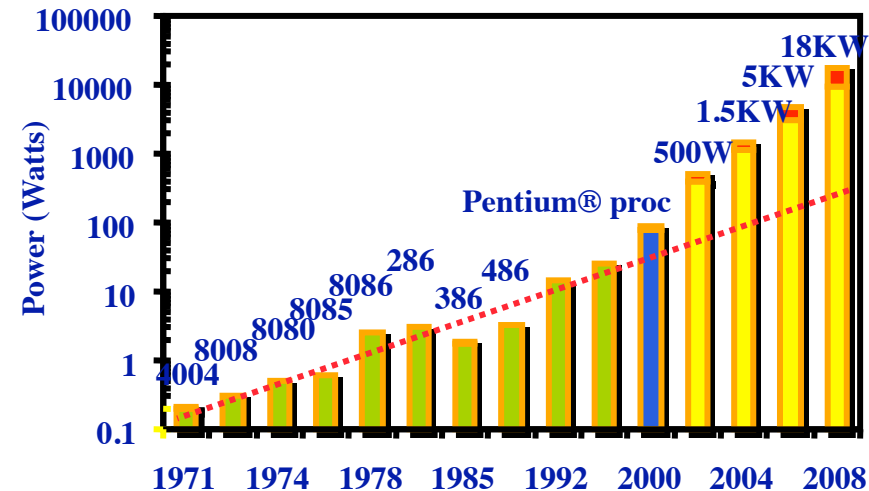
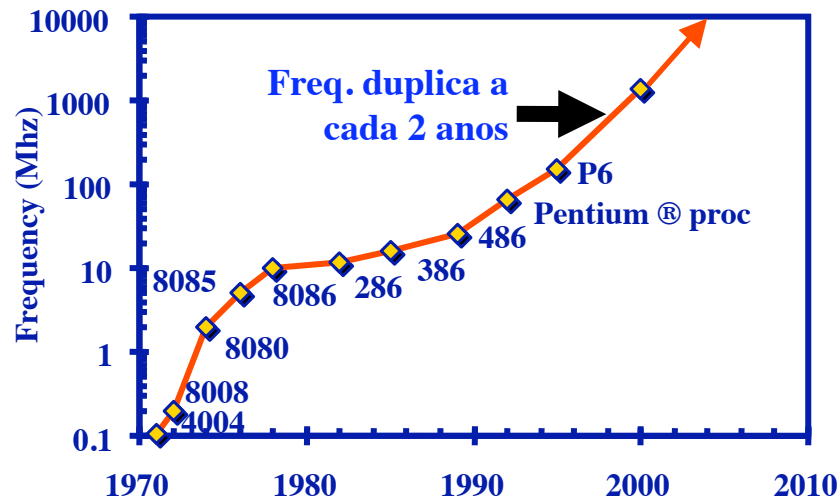
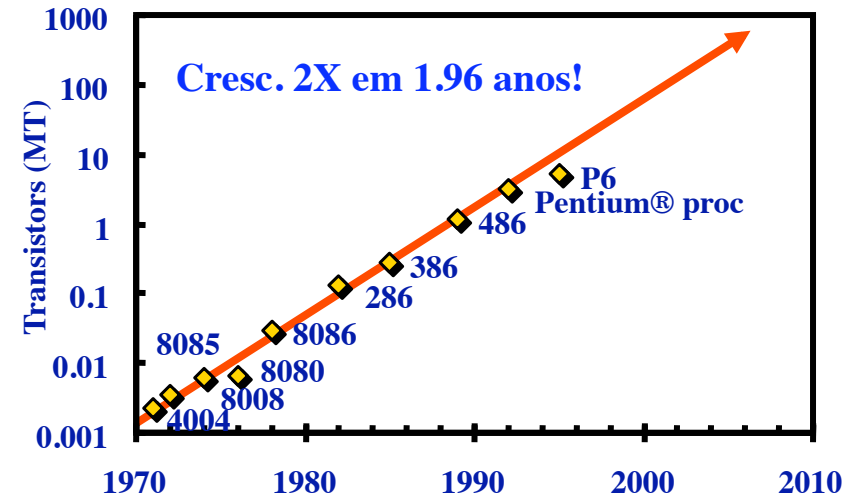


Fonte: Rabaey; Chandrakasan; Nikolic, 2003

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► A Lei de Moore nos Processadores

- No de transistores
- Potência
- Frequência



Fonte: Ravi, Chandra, Nikolic, 2003

INE/CTC/UPSC

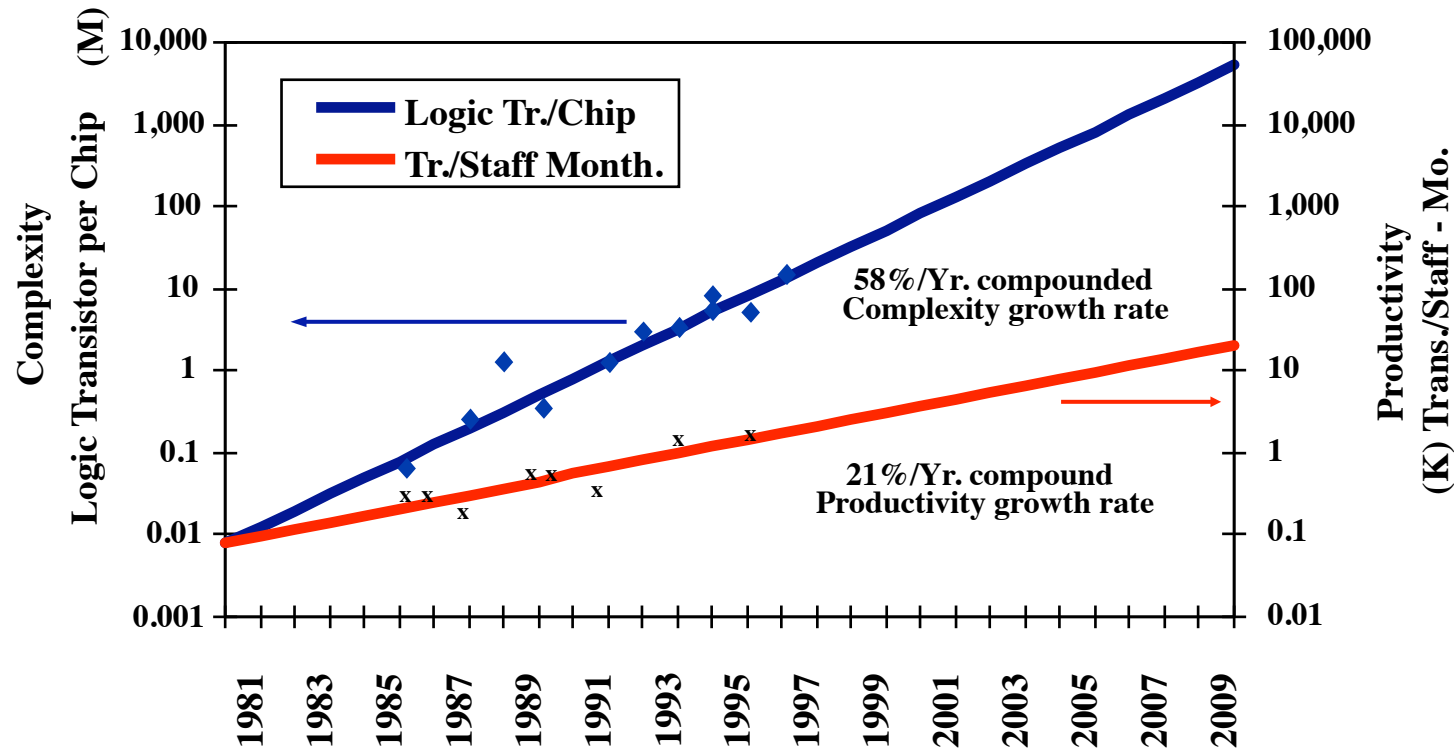
Sistemas Digitais - semestre 2010/2

slide 1P.5

Prof. José Luís Güntzel

# Projeto de Sistemas Digitais com Ferramentas EDA

## ▶ O “Gap” de Produtividade

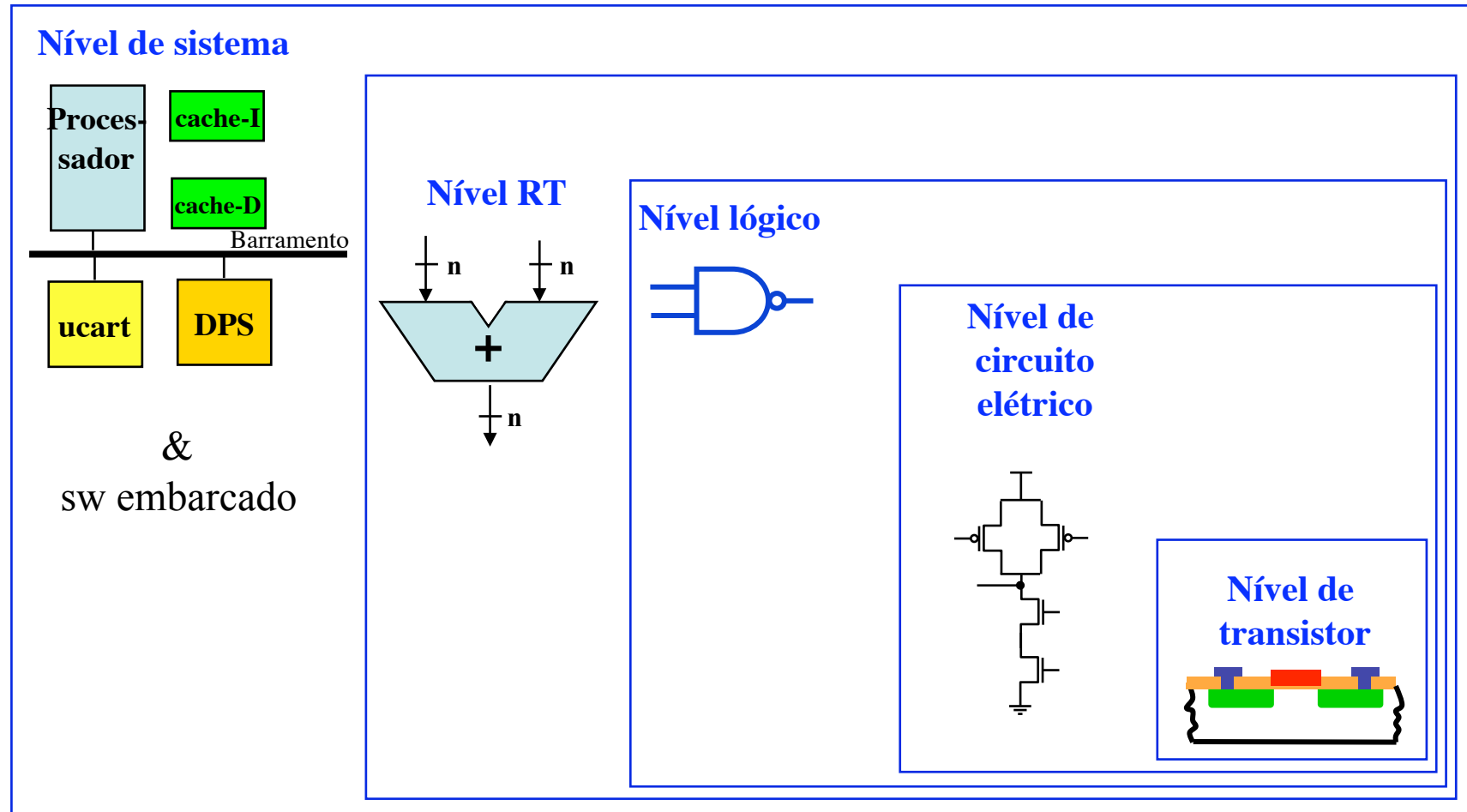


Source: Sematech

Courtesy, ITRS Roadmap

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Níveis de Abstração



# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► Visões de Projeto

### **Descrição Estrutural**

- Utiliza-se um conjunto de blocos e conexões que representam uma possível implementação do sistema eletrônico. Pode-se usar linguagem de descrição de hardware (HDL) ou esquemáticos (em papel ou usando algum editor de esquemático).

### **Descrição Comportamental**

- Faz uso de texto em linguagem natural, HDL ou equações para descrever como o sistema eletrônico se comporta (i.e., funciona).

### **Descrição Física**

- usada para implementar o sistema eletrônico. No caso de fabricação do chip com tecnologia CMOS, descrição das geometrias das máscaras que serão usadas no processo de litografia fina.



# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► **Níveis de Abstração de Sistemas Digitais**

### **Nível de Transistores:**

- Transistores e materiais utilizados na fabricação do circuito integrado

### **Nível de Circuito Elétrico:**

- Transistores,
- Resistores,
- Capacitores,
- Indutores e
- Fios.

### **Nível Lógico:**

- Portas lógicas,
- Latches e
- Flip-flops.

# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► Níveis de Abstração de Sistemas Digitais

### Nível RT (*Register Transfer*):

- Unidades funcionais (somadores, subtratores, somadores /subtratores, multiplicadores etc)
- Rede de interconexão (fios, multiplexadores, decodificadores, barramentos, buffers tri-state)
- Registradores e blocos de memória RAM, ROM etc

### Nível de Sistema:

- Processadores de uso genéricos (CPUs),
- Processadores para domínios específicos (ASIPs),
- Processadores específicos (ASICs),
- Barramentos,
- Memórias,
- Software embarcado.

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Custo dos Circuitos Integrados

### 1. Custo Fixo ou Não-Recorrente (NRE)

- Independe do volume (quantidade de peças a serem produzidas), mas depende da complexidade do projeto
  - Tempo de Projeto ( $\propto$  complexidade, rigor dos requisitos, produtividade da equipe)
  - Produção das máscaras
  - Investimento em pesquisas..

### 2. Custo Variável ou Recorrente:

- Proporcional ao volume do produto e à área do chip
  - Processamento do silício, encapsulamento (*packaging*), teste

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Custo dos Circuitos Integrados

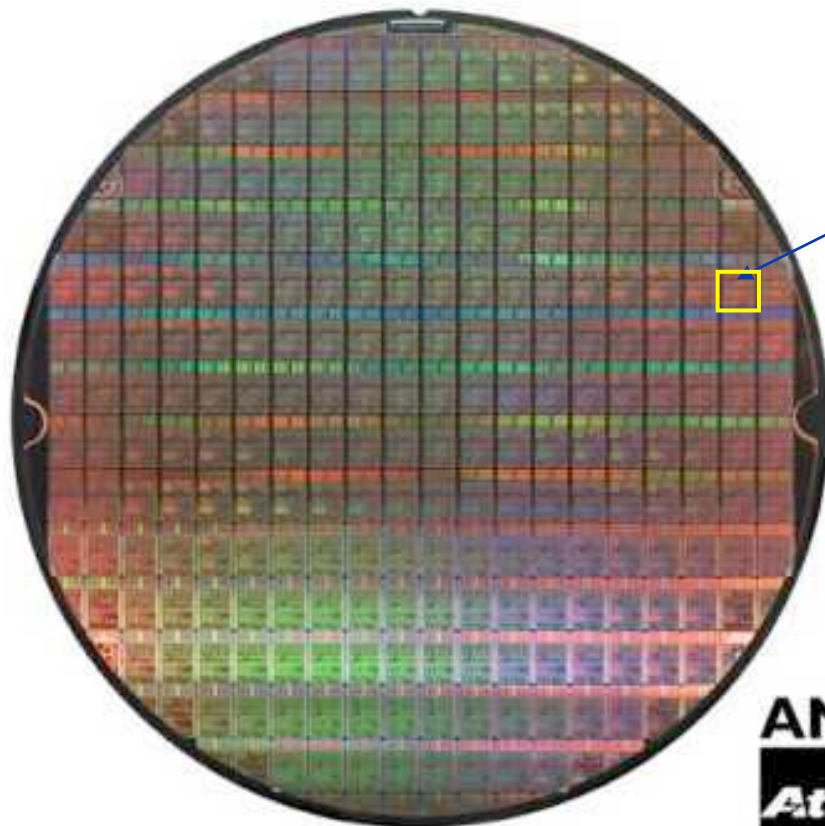
$$\text{Custo por CI} = \text{custo var.} + \frac{\text{custo fixo}}{\text{volume}}$$

$$\text{custo var.} = \frac{\text{custo do die} + \text{custo do teste do die} + \text{custo do } \textit{packaging}}{\text{Yield pós-teste do die}}$$

Obs: Yield é o rendimento da fabricação

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Custo do Die (Chip)



Single die

Wafer

Diâmetro= 30cm (12")



From <http://www.amd.com>

# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► Síntese e Síntese Automática

### **Síntese:**

Traduz uma dada descrição de um sistema eletrônico para uma nova descrição (sendo esta nova descrição em um nível inferior de abstração) por meio da adição de detalhes de implementação.

### **Síntese Automática:**

Síntese realizada com o auxílio de ferramentas computacionais (atualmente referenciadas por ferramentas de **EDA- *Electronic Design Automation***).

# Introdução à Linguagem VHDL

---

## Histórico

- Criada sob encomenda do Departamento de Defesa dos EUA (DoD)
  - 1981: DoD patrocina encontro de especialistas para discutir métodos para projeto de CIs.
  - 1983: Definição dos requisitos da linguagem. DoD assina contrato com IBM + TI + Intermetrics para desenvolvimento da linguagem e ferramentas.
- Padronizada pelo IEEE (*The Institute of Electrical and Electronics Engineers*)
  - Padrão IEEE 1076-1987 (primeiro padrão industrial)
  - Padrão IEEE 1164-1993 (introduz novos tipos de dados, tal como `std_logic` e `std_logic_vector`)

# Introdução à Linguagem VHDL

---



## Características

- **O nome:**
  - **VHDL** = **VHSIC** **H**ardware **D**escription **L**anguage
  - **VHSIC** = **V**ery **H**igh **S**peed **I**ntegrated **C**ircuits
- **A sintaxe: similar à linguagem Ada**
- **Objetivos iniciais:**
  - Permitir a **especificação** de circuitos de forma não ambígua (**modelagem**).
  - Facilitar a **documentação** de circuitos complexos.
  - Servir de entrada para ferramentas computacionais de **simulação**.
- **Objetivo contemporâneo:**
  - Serve de entrada para ferramentas de **síntese automática** e de **validação** com métodos formais.



# Introdução à Linguagem VHDL

---

## ► Características: Síntese a Partir de VHDL

- **Descrições VHDL no nível RT são 100% sintetizáveis**
- **Síntese a partir de descrições VHDL comportamentais dependem:**
  - Das construções VHDL utilizadas no código.
  - Da ferramenta de síntese utilizada.

# Introdução à Linguagem VHDL

---

## ► Primeiros Conceitos

Uma descrição VHDL é dividida em **duas partes fundamentais**:

- 1) **Entidade (Entity)** – Descreve a interface do sistema digital descrito com o mundo externo. Apresenta a definição dos pinos de entrada e saída.
- 2) **Arquitetura (Architecture)** – Descreve o comportamento ou a estrutura do sistema digital. Define como a função do sistema é realizada.

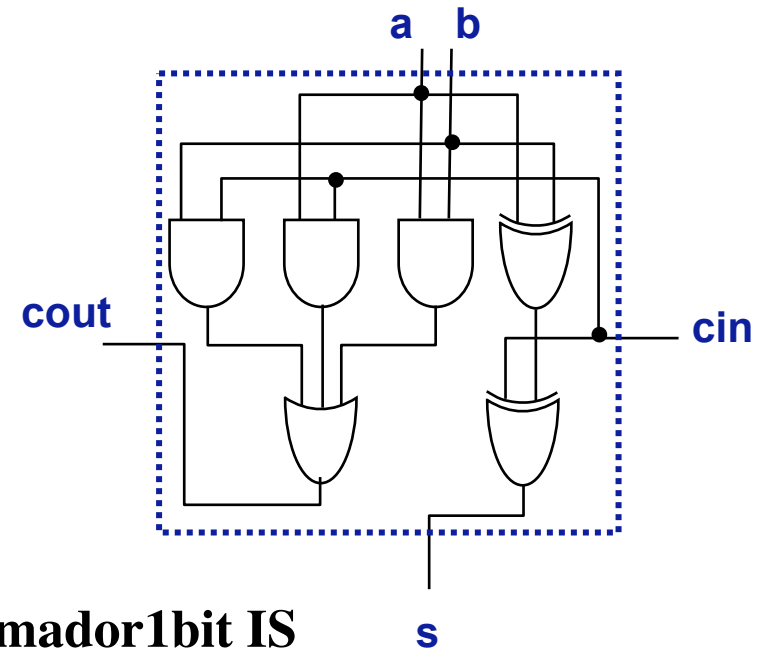
# Introdução à Linguagem VHDL

## ► Exemplo: um Full Adder

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;
```

```
ENTITY somador1bit IS  
    PORT (cin, a, b : IN STD_LOGIC;  
          s, cout : OUT STD_LOGIC);  
END somador1bit ;
```

```
ARCHITECTURE comportamento OF somador1bit IS  
BEGIN  
    s <= a XOR b XOR cin;  
    cout <= (a AND b) OR (a AND cin) OR (b AND cin);  
END comportamento;
```



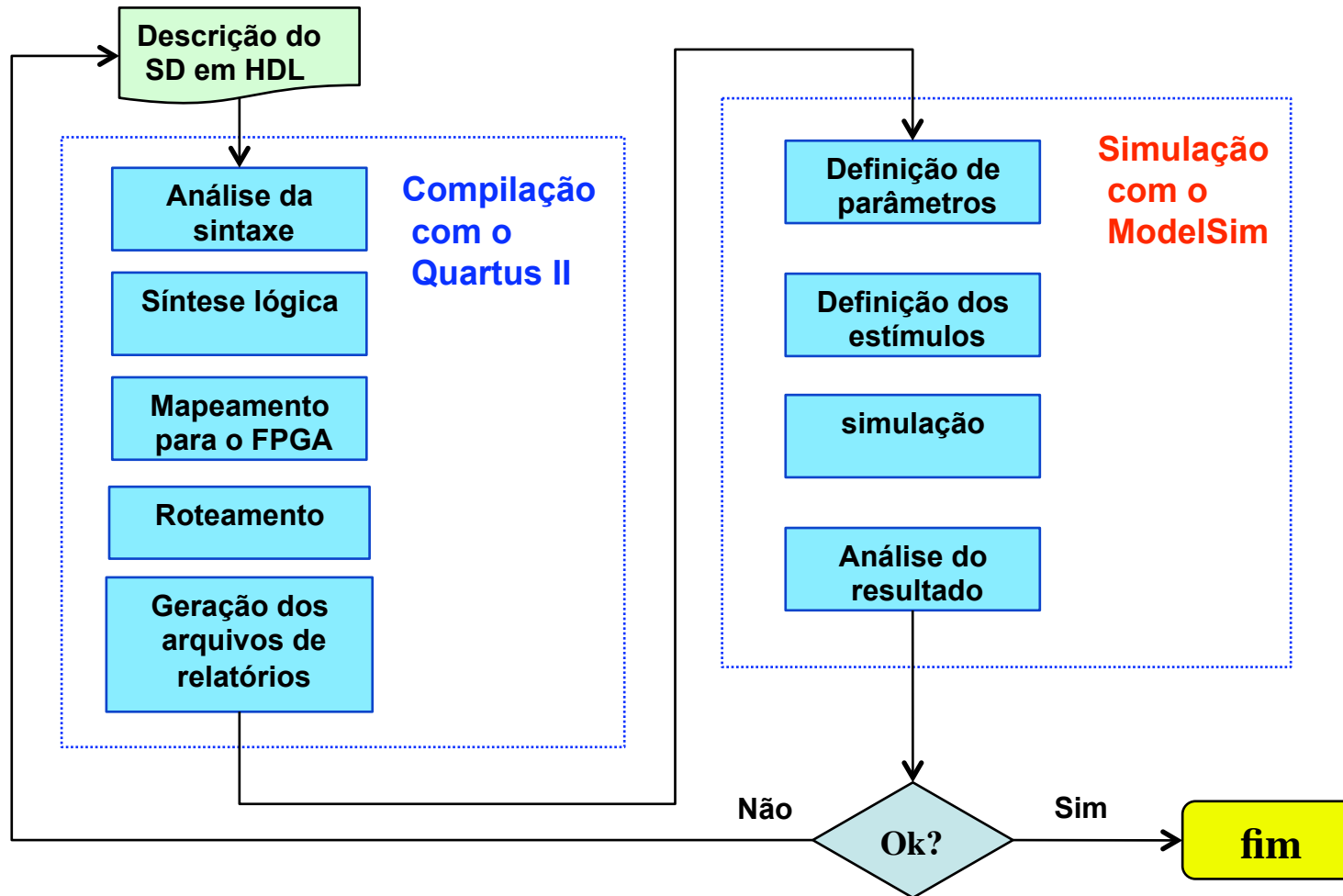
# Introdução à Linguagem VHDL

## ► Alguns Tipos de Dados em VHDL

tipo	valores	comentário
boolean	{false, true}	Nativa da linguagem
bit	{0,1}	Nativa da linguagem
std_logic	{0, 1, - , Z}	Implementada no pacote std_logic_1164
std_logic_vector	{0, 1, - , Z}	Implementada no pacote std_logic_1164; Vetor de std_logic

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Fluxo de Projeto para FPGAs



# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► **Experimento 1:** descrição/compilação e simulação de um SC **Organizando o Ambiente de Trabalho no Computador**

1. Na pasta Meus\_documentos, criar uma pasta com o seu nome (p. ex., “Paulo”). Na pasta “Paulo”, criar uma pasta com nome de “somador1bit”.

**Obs: jamais crie seus projetos na mesma pasta onde o Quartus II ou o ModelSim estão instalados!**

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► **Experimento 1:** descrição/compilação e simulação de um SC

### **Invocando o Quartus II e Criando um Projeto**

- 2 Invocar o Quartus II (a partir do ícone na área de trabalho, ou a partir do “Iniciar->Programas” do windows, sub-menu “Altera”).
- 3 Na janela “Get Started With Quartus II Software”, selecionar “Create New Project”
- 4 Clicar em “Next”.
- 5 Selecionar o caminho para a pasta criada no passo 1 (clicando no botão identificado com “...”).
- 6 Na caixa de diálogo identificada por “What is the name of this project”, escrever “somador1bit”.
- 7 Clicar em “Next”. Clicar em “Next” novamente.
- 8 Na caixa de diálogo “Device Family”, selecionar “Cyclone II”. Na lista identificada por “Available Devices”, selecionar EP2C35F672C6. Clicar em “Next”. (Ver próximo slide.)

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Experimento 1: descrição/compilação e simulação de um SC

Selecionar “Cyclone II”



Selecionar “EP2C35F672C6”



Após, clicar em “Next”



New Project Wizard: Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family  
Family: Cyclone II  
Devices: All

Target device  
☐ Auto device selected by the Fitter  
☒ Specific device selected in 'Available devices' list

Show in 'Available device' list  
Package: Any  
Pin count: Any  
Speed grade: Any  
☒ Show advanced devices  
☐ HardCopy compatible only

Available devices:

Name	Core v...	LEs	User I/...	Memor...	Embed...	PLL
EP2C20F484C8	1.2V	18752	315	239616	52	4
EP2C20F484I8	1.2V	18752	315	239616	52	4
EP2C20Q240C8	1.2V	18752	142	239616	52	4
EP2C35F484C6	1.2V	33216	322	483840	70	4
EP2C35F484C7	1.2V	33216	322	483840	70	4
EP2C35F484C8	1.2V	33216	322	483840	70	4
EP2C35F484I8	1.2V	33216	322	483840	70	4
EP2C35F672C6	1.2V	33216	475	483840	70	4
EP2C35F672C7	1.2V	33216	475	483840	70	4

Companion device  
HardCopy:   
☒ Limit DSP & RAM to HardCopy device resources

< Back Next > Finish Cancel



# Projeto de Sistemas Digitais com Ferramentas EDA

---

## ► **Experimento 1:** descrição/compilação e simulação de um SC **Invocando o Quartus II e Criando um Projeto (cont.)**

9. Na caixa de diálogo “Simulation”, selecionar “ModelSim-Altera”. Clicar em **Next**.
10. Clicar em “**Finish**”. (Ver próximo slide.)

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► Experimento 1: descrição/compilação e simulação de um SC

Selecionar “ModelSim-Altera”



Após, clicar em “Next”



New Project Wizard: EDA Tool Settings [page 4 of 5]

Specify the other EDA tools -- in addition to the Quartus II software -- used with the project.

Design Entry/Synthesis

Tool name: <None>

Format:

☐ Run this tool automatically to synthesize the current design

Simulation

Tool name: ModelSim-Altera

Format: VHDL

☒ Run gate-level simulation automatically after compilation

Timing Analysis

Tool name: <None>

Format:

☐ Run this tool automatically after compilation

< Back Next > Finish Cancel

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► **Experimento 1:** descrição/compilação e simulação de um SC

### **Criando um Arquivo VHDL**

11. No menu “File” (canto superior esquerdo da janela do Quartus II), selecionar “New”. Selecionar “VHDL File” e clicar em “OK”.
12. Copiar o arquivo VHDL do slide 1P.19 (usar copy-paste).
13. No menu “File” (canto superior esquerdo da janela do Quartus II), selecionar “Save as”. Certificar-se que o nome do arquivo seja igual ao nome da entidade (neste caso, “somador1bit”). Clicar em “OK”.

Obs: este projeto terá apenas um arquivo VHDL. Porém, projetos mais complexos podem ter diversos arquivos VHDL. Neste caso, cada arquivo VHDL deverá ser editado e salvo e será importante certificar-se de que todos os arquivos foram incluídos no projeto. Para isso, deve-se clicar na opção “Project” do menu superior do Quartus e selecionar “Add/Remove Files in Project...”.

# Projeto de Sistemas Digitais com Ferramentas EDA

## ► **Experimento 1:** descrição/compilação e simulação de um SC

### **Compilando um Projeto**

14. No menu “Processing” (aba superior da janela do Quartus II), selecionar “Start Comoilation”. (Ou clicar no triângulo roxo, na aba superior).
15. Aguardar a mensagem “Full Compilation was Succesfull” (*warnings* são normais) ou a mensagem de erros (quando houver erros no VHDL).
16. Anotar os seguintes dados mostrados na janela “Compilation Report – Flow Summary”:
  - Total combinational functions:
  - Dedicated logic elements:
17. Anotar os seguintes dados mostrados na janela “Message” (procurar pela linha que inicia por “Longest tpd from ...”):
  - tpd:
  - Source pin
  - Destination pin:

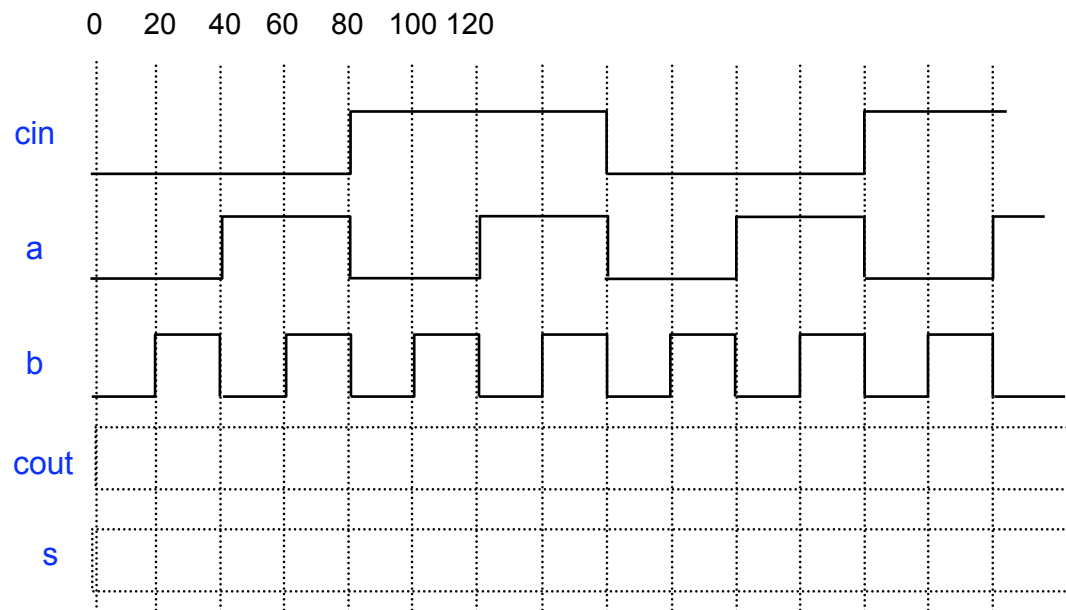
# Projeto de Sistemas Digitais com Ferramentas EDA

## ► **Experimento 1:** descrição/compilação e simulação de um SC

### **Preparação dos Estímulos para a Simulação**

**Solução trivial (ingênua): Transformar a tabela-verdade em formas de onda**

cin	a	b	cout	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



#### Observações:

1. Preencher a mão as waveforms (formas de onda) esperadas para as saídas para confrontá-la com o resultado da simulação.
2. T deve ser maior que “longest tpd” reportado pelo Quartus II.