



## Introdução

### Computadores: Abstrações e Tecnologias

Prof. Edson S. Gomi

PCS - Departamento de Engenharia de Computação e Sistemas Digitais  
Escola Politécnica da Universidade de São Paulo

Agosto, 2020

# Agenda da Aula

- A evolução dos Sistemas Digitais;
- *Design* dos Sistemas Digitais;
- Recapitação: Sistemas Digitais I;
- *Preview*: Sistemas Digitais II;
- Orientações gerais da disciplina;
- Exercício: VHDL e Juiz.

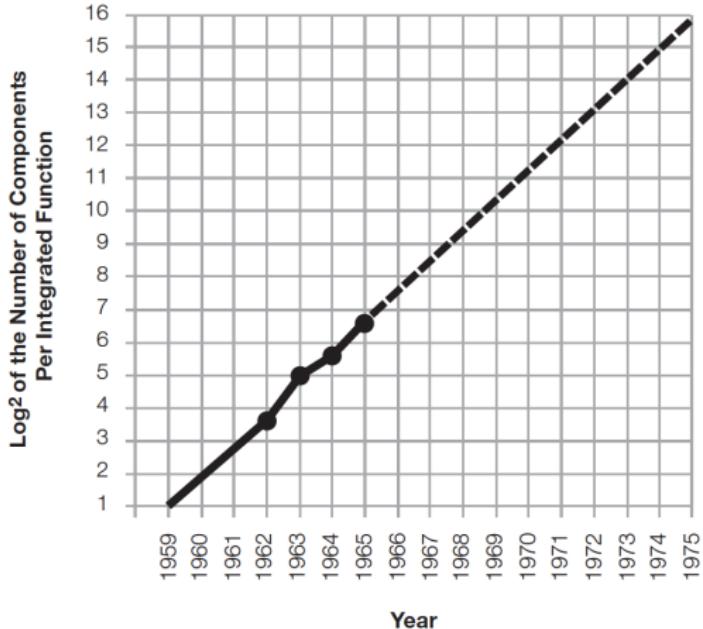
# Ao final da aula você saberá:

- A Lei de Moore;
- A evolução dos Sistemas Digitais e de seus processos de *design*;
- Os níveis de abstração nos Sistemas Digitais;
- Visão geral de Sistemas Digitais I e II;
- Usar o VHDL Judge;
- Como estudar e as regras gerais da disciplina.

# Lei de Moore

- Em 1965, Gordon Moore, co-fundador da Intel, previu que a quantidade de transistores num circuito integrado dobraria a cada 2 anos. Esta constatação, que se comprovou na prática, ficou conhecida como **Lei de Moore**;
- Artigo: Moore, Gordon E., *Cramming more components onto integrated circuits*, Electronics Magazine, Volume 38, Número 8, 1965 [4];
- Veja o vídeo :  
<https://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>.

# Lei de Moore 1/3



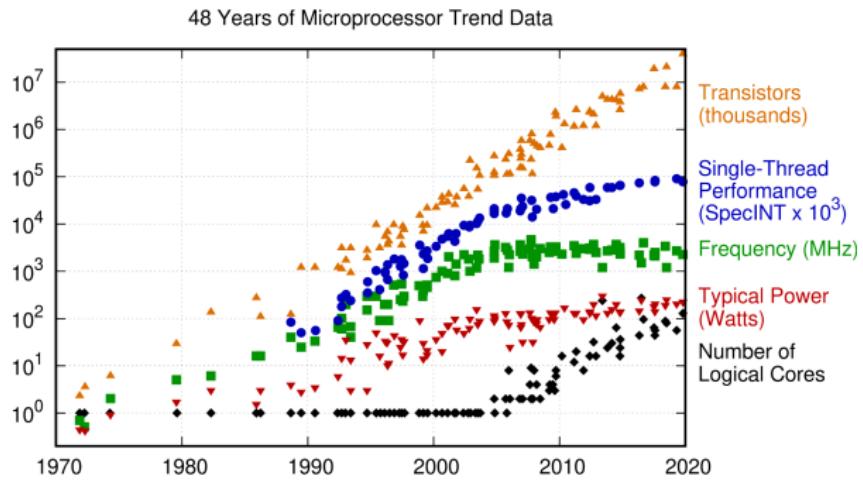
Fonte: [4]

Prof. Edson S. Gomi

Sistemas Digitais II

Agosto, 2020 5/45

# Lei de Moore 2/3

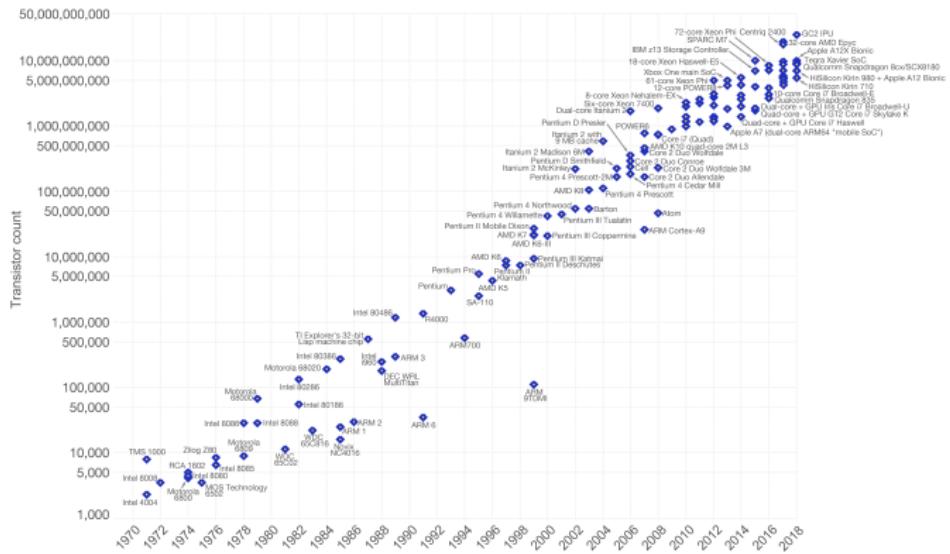


Fonte: <https://github.com/karlrupp/microprocessor-trend-data>

Lei de Moore 3/3

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
The data visualization is available at [OurWorldInData.org](http://OurWorldInData.org). There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Boeser

Fonte: [https://www.visualcapitalist.com/wp-content/uploads/2019/11/Moores\\_Law\\_.jpg](https://www.visualcapitalist.com/wp-content/uploads/2019/11/Moores_Law_.jpg)

Transistor\_Count\_1971-2018.png  
Prof. Edson S. Gomi

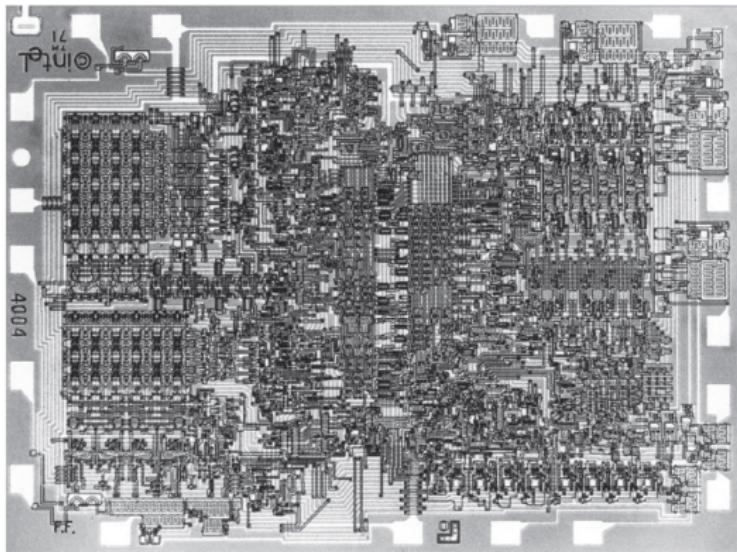
Sistemas Digitais II

Agosto, 2020 7/45

# O que dá para fazer com milhares de transistores?

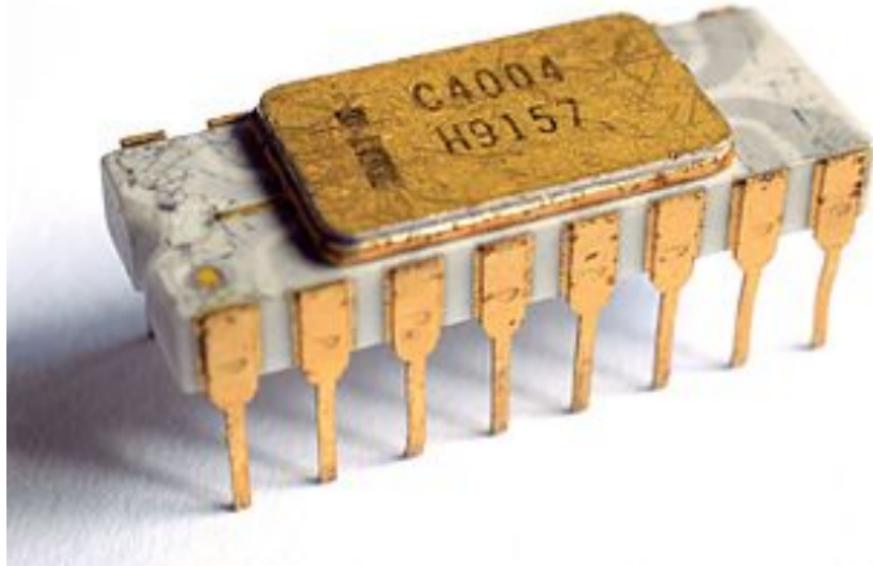
- Microprocessador Intel 4004 (1971);
- Em 1969, a japonesa Busicom encomendou à Intel um conjunto de CLs para uma nova calculadora [2];
- Projetado por Frederico Faggin, a solução entregue foi um microprocessador de 4 bits, com 2.300 transistores CMOS, capaz de executar 60.000 instruções por segundo, com clock de 750 KHz [3];
- Conjunto de CLs:  $4 \times 4001$  (ROM de 2048 bits),  $2 \times 4002$  (DRAM de 320 bits),  $2 \times 4003$  (Registrador de Deslocamento de 10 bits) e  $1 \times 4004$  (CPU de 4 bits), todos em encapsulamento de 16 pinos.

# Imagen do 4004



Fonte: [3]

# Encapsulamento do 4004

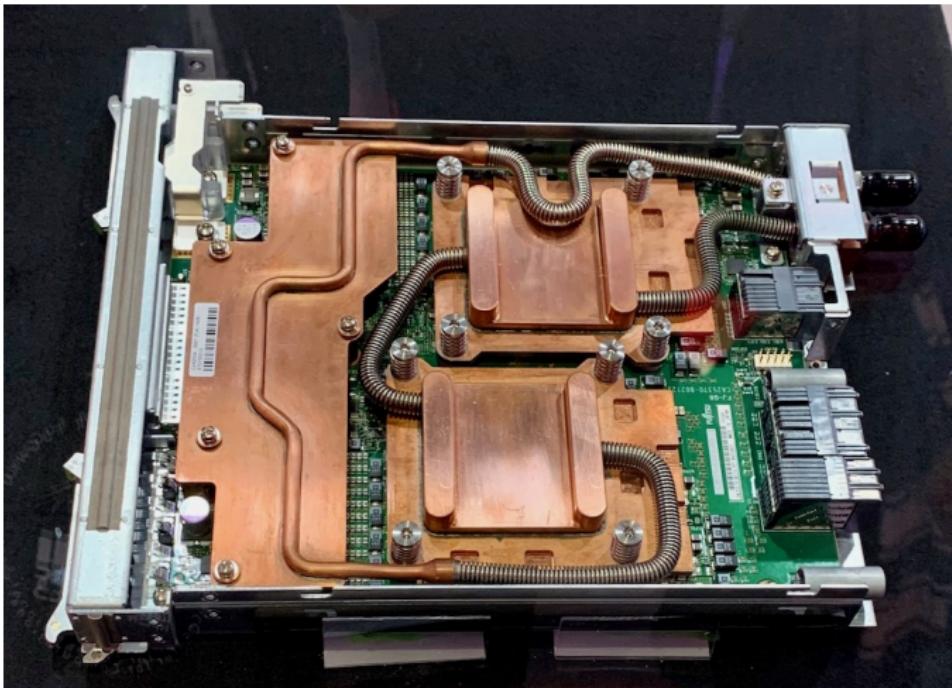


Fonte: [https://en.wikipedia.org/wiki/File:Intel\\_C4004.jpg](https://en.wikipedia.org/wiki/File:Intel_C4004.jpg)

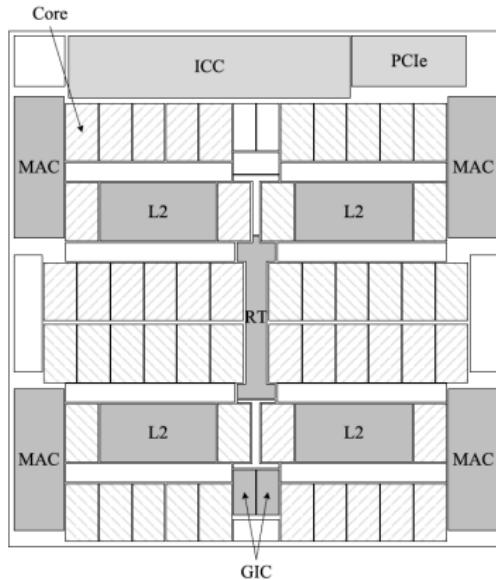
# O que dá para fazer com bilhões de transistores?

- Processador Fujitsu A64FX: ARM v8.2+SVE, 48 núcleos, 2.2 GHz de clock, 3.4 INT8 TOps/s, cache L2 de 32 MiB, 8.7 bilhões de transistores [1];
- Equipa o supercomputador japonês Fukagu, No. 1 na lista TOP500 (22/06/2020): 158,976 nós, cada um com 2 processadores A64FX, com performance de pico de 3,9 INT8 ExaOps ( $3,9 \times 10^{18}$  operações inteiras de 8 bits/segundo) (<https://www.top500.org/news/japan-captures-top500-crown-arm-powered-supercomputer/>).

# Imagen de um nó do Fukagu, com 2 × A64FX



# Diagrama de blocos do A64FX



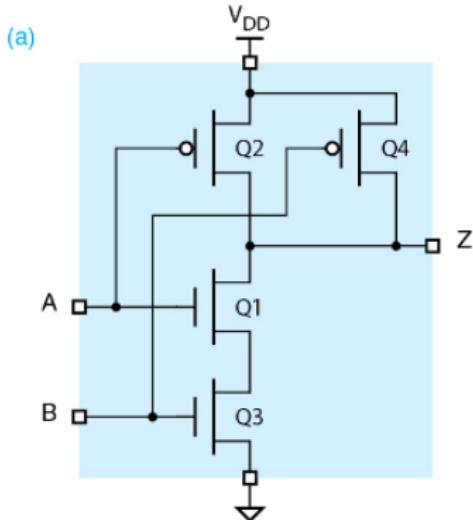
Fonte: [https://github.com/fujitsu/A64FX/blob/master/doc/A64FX\\_Microarchitecture\\_Manual\\_en\\_1.2.pdf](https://github.com/fujitsu/A64FX/blob/master/doc/A64FX_Microarchitecture_Manual_en_1.2.pdf)

//github.com/fujitsu/A64FX/blob/master/doc/A64FX\_Microarchitecture\_Manual\_en\_1.2.pdf

# Como tirar o melhor proveito dos transistores?

- A disponibilidade de transistores é o direcionador das estruturas dos Sistemas Digitais e dos métodos de *design*;
- Como era no início e como é agora?
- Requisitos que são levados em conta: algoritmos, complexidade, espaço, tempo e energia.

# Onde se usa os transistores?



(b)

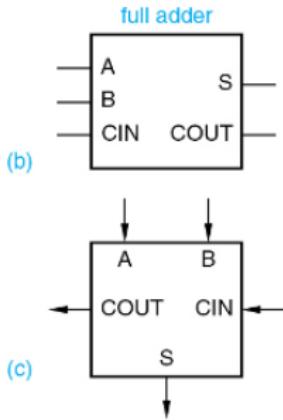
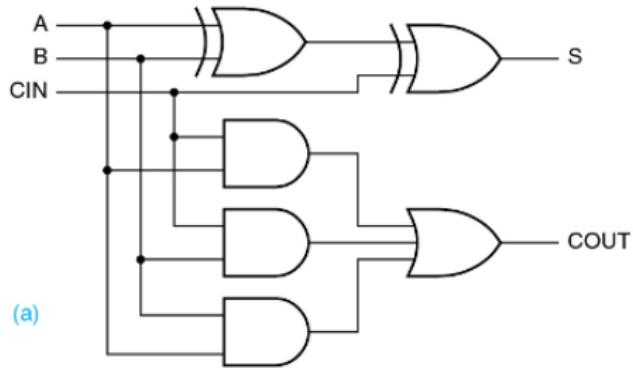
A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	H
H	L	on	off	off	on	H
H	H	on	off	on	off	L

(c)



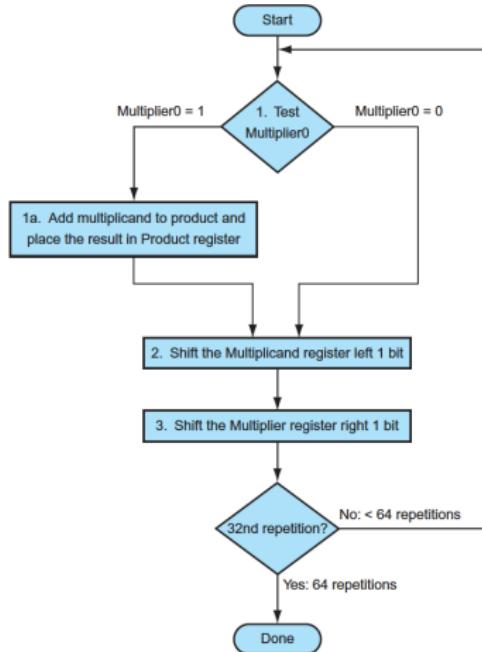
Fonte: [7]

# Bloco - Somador Completo



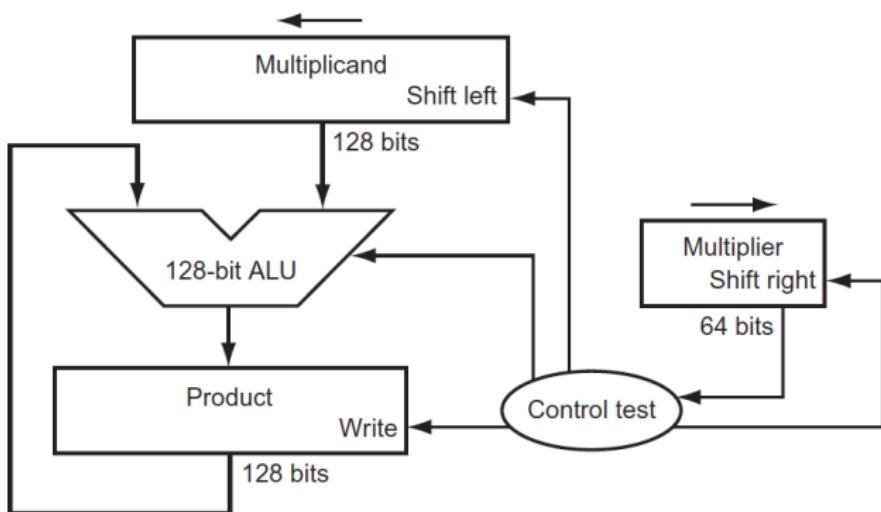
Fonte: [7]

# Algoritmo - Multiplicação



Fonte: [6]

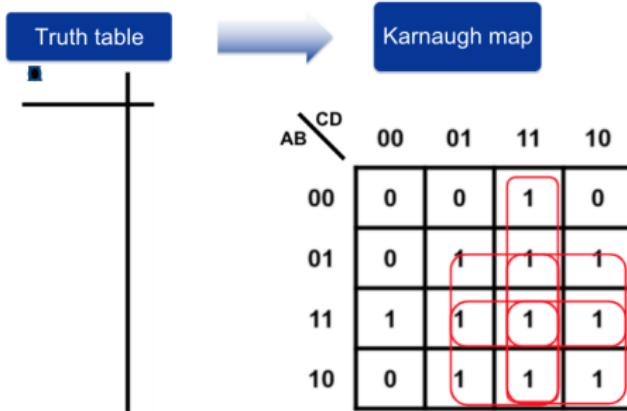
# Implementação - Multiplicador



Fonte: [6]

# *Design para circuitos simples*

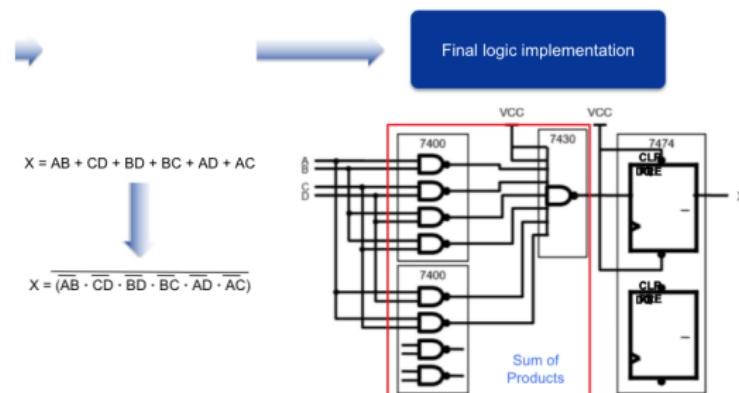
## Digital Design with TTL Logic



Fonte: <https://www.intel.com/content/www/us/en/programmable/support/training/course/odsw1005.html>

# Design com TTL

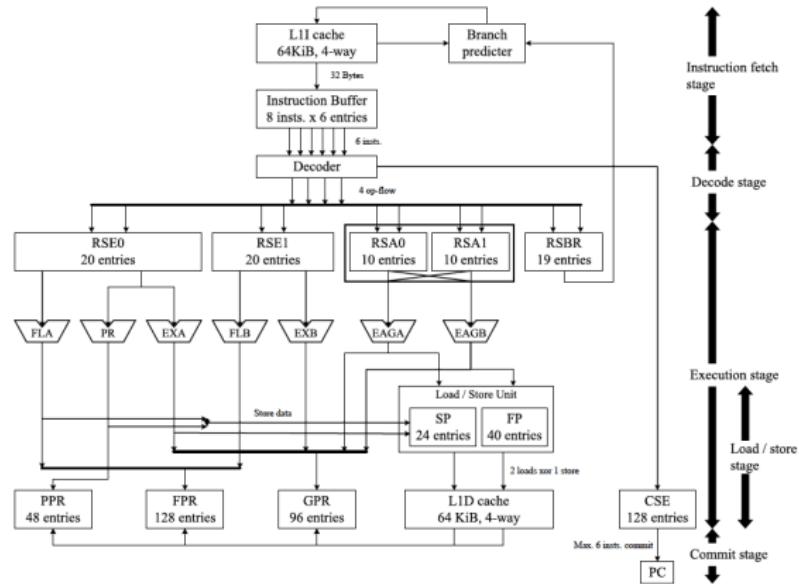
## Digital Design with TTL Logic (cont.)



Fonte: <https://www.intel.com/content/www/us/en/programmable/support/training/course/odsw1005.html>

//www.intel.com/content/www/us/en/programmable/support/training/course/odsw1005.html

# Como projetar o A64FX?

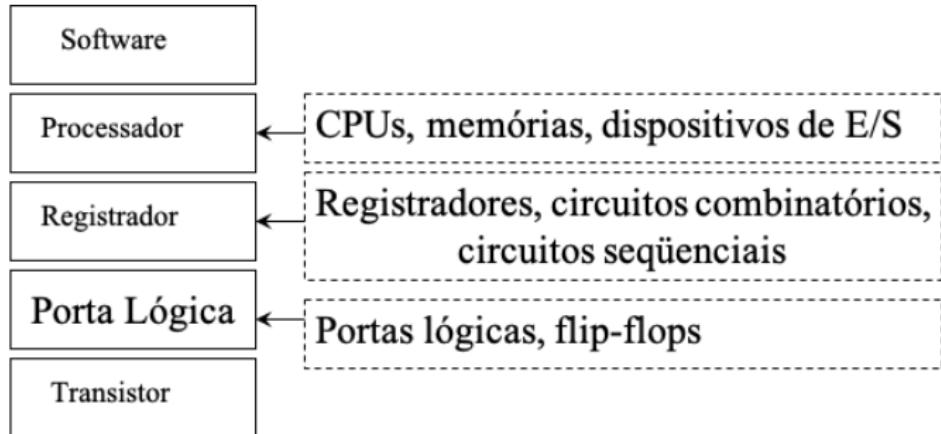


Fonte: [https://github.com/fujitsu/A64FX/blob/master/doc/A64FX\\_Microarchitecture\\_Manual\\_en\\_1.2.pdf](https://github.com/fujitsu/A64FX/blob/master/doc/A64FX_Microarchitecture_Manual_en_1.2.pdf)

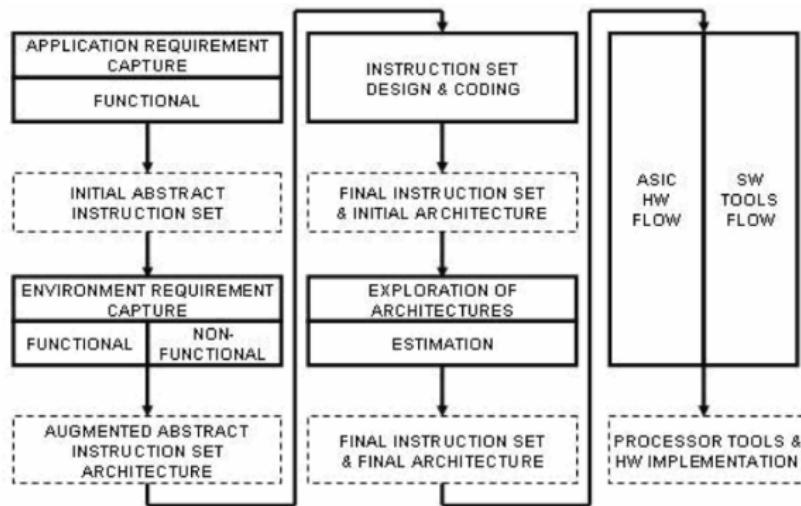
# *Design* para circuitos complexos

- Os métodos “simples”, como Mapa de Karnaugh ou Máquina de Estados, não são escaláveis para circuitos complexos;
- Entram em cena as idéias de níveis de abstração e de *design* estruturado;
- Forte uso de linguagens de descrição de hardware (HDL - *Hardware Description Language*) e bibliotecas IPs (IP - *Intellectual Property*);
- Ferramentas computacionais de suporte às etapas do *design* moderno: descrição, compilação, simulação, verificação, roteamento, etc.

# Níveis de Abstração

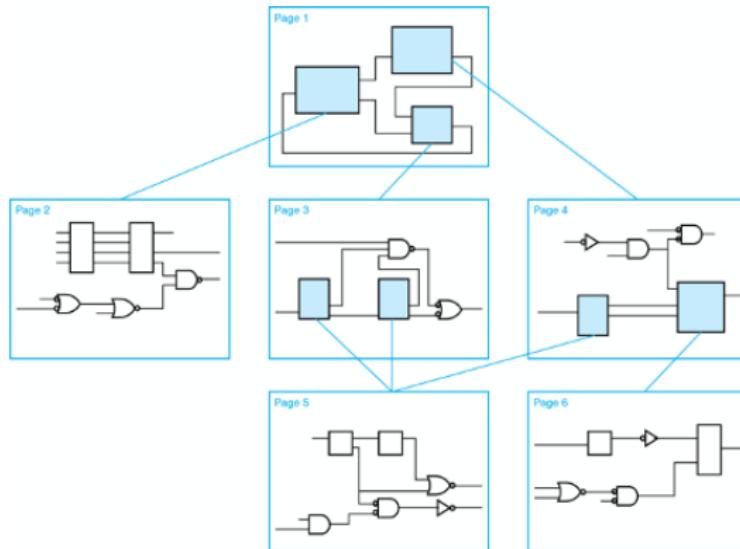


# Fluxo de *design* de um processador



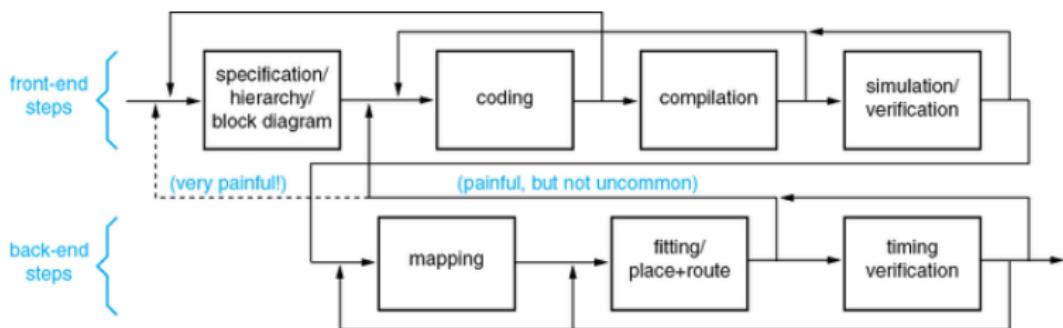
Fonte: [5]

# Modelo Estruturado



Fonte: [7]

# Design baseado em HDL



Fonte: [7]

# O que estudamos em Sistemas Digitais I? 1/3

- Sistemas de numeração: representação binária e hexadecimal;
- Códigos: BCD; ASCII, Gray Code, Hamming Codes;
- Aritmética: adição, subtração, multiplicação e divisão; sem e com sinal (Complemento de Base);

# O que estudamos em Sistemas Digitais I? 2/3

- Álgebra de Chaveamento;
- Circuitos CMOS e portas lógicas;
- *Design* de circuitos combinatórios; Mapa de Karnaugh;
- Blocos combinatórios: codificadores, multiplexadores, comparadores e somadores;

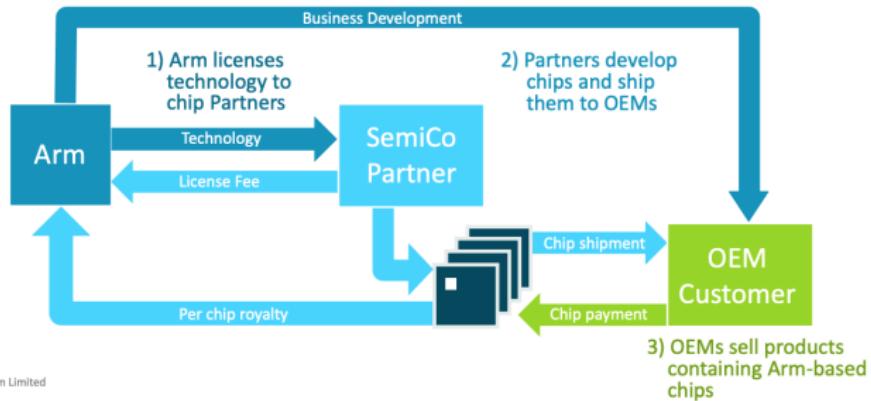
# O que estudamos em Sistemas Digitais I? 3/3

- Biestáveis, Latches e Flip-Flops;
- Máquina de Estados Finito;
- *Design* de circuitos sequenciais: modelos de Moore e de Mealy, diagramas ASM;
- Blocos sequenciais: contadores, registradores, registradores de deslocamento;
- Memórias e Lógica Programável: ROM, RAM, PLD e FPGA;
- Linguagem de Hardware: VHDL e testbenches.

# O que estudaremos em Sistemas Digitais II?

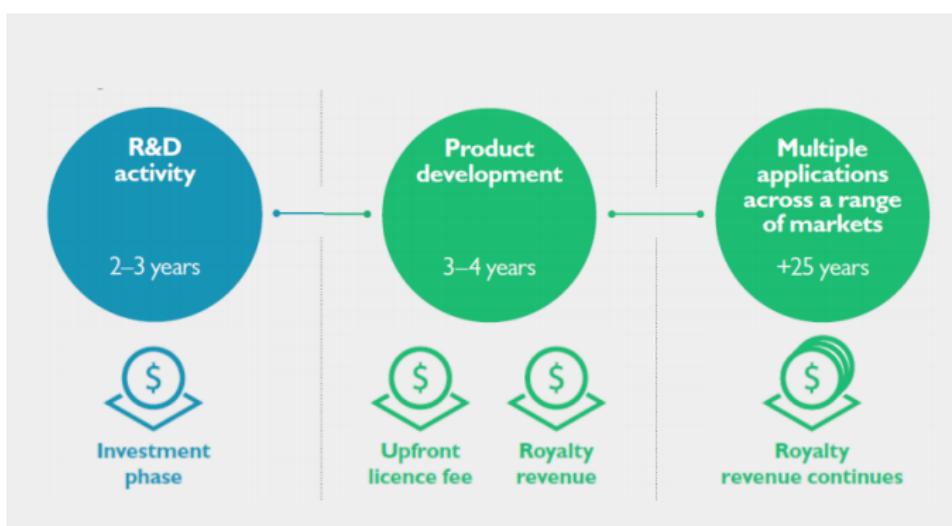
- Recapitação: testbenches, diagramas ASM;
- Memórias e endereçamento;
- Arquiteturas de computadores; ARM v8 *Instruction Set*;
- Aritmética: inteira e ponto flutuante;
- Processador monociclo: PoliLegv8;
- Barramentos e interfaces assíncronas;
- Modelagem e síntese de sistemas;
- Síntese e verificação de hardware;
- Exemplos de aplicações.

# Por que ARM?



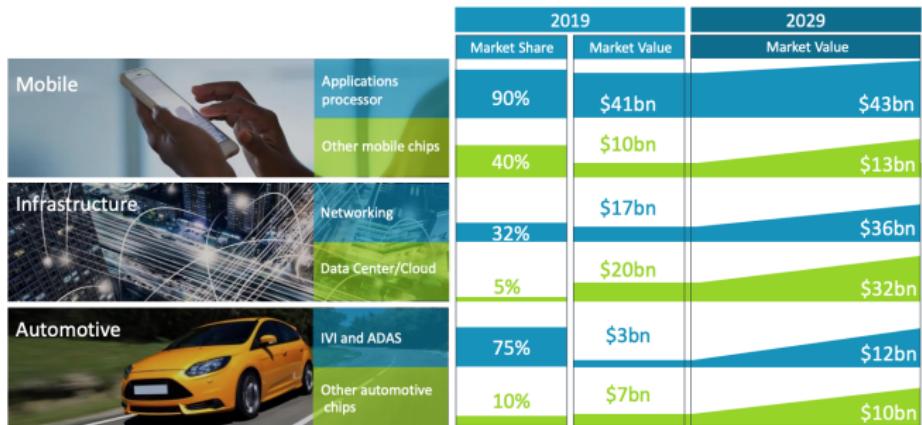
Fonte: [https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides\\_q4fy2019\\_01\\_en.pdf](https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides_q4fy2019_01_en.pdf)

# Ciclo do Modelo de Negócio



Fonte: [https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides\\_q4fy2019\\_01\\_en.pdf](https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides_q4fy2019_01_en.pdf)

# Onde é usado?



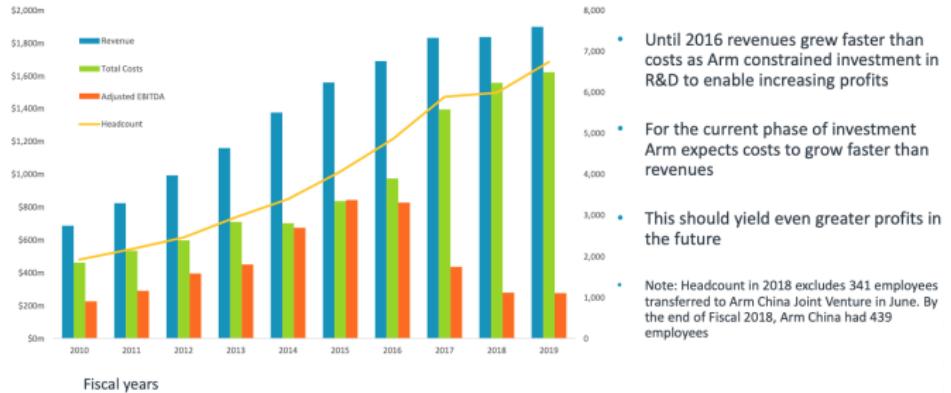
Fonte: [https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides\\_q4fy2019\\_01\\_en.pdf](https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides_q4fy2019_01_en.pdf)

# Quantidade de processadores entregues



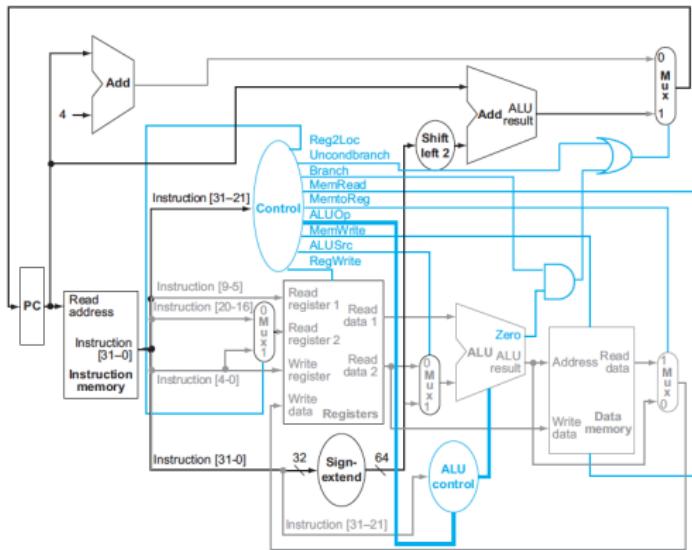
Fonte: [https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides\\_q4fy2019\\_01\\_en.pdf](https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides_q4fy2019_01_en.pdf)

# Receita, investimento e lucro



Fonte: [https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides\\_q4fy2019\\_01\\_en.pdf](https://group.softbank/system/files/pdf/ir/presentations/2019/arm-roadshow-slides_q4fy2019_01_en.pdf)

# Diagrama do PoliLegv8



Fonte: [6]

# 8 Grandes Idéias em Arquitetura de Computadores

- 1 Projete pensando na Lei de Moore;
- 2 Use a abstração para simplificar o *design*;
- 3 Torne rápido o caso mais frequente;
- 4 Melhoria do desempenho por paralelismo;
- 5 Melhoria do desempenho por *pipelining*;
- 6 Melhoria do desempenho por predição;
- 7 Hierarquia de memórias;
- 8 Confiabilidade por redundância.

# Objetivos de Aprendizagem

- Entender os requisitos do Sistema Digital a partir do problema a ser resolvido;
- Elaborar o projeto do Sistema Digital;
- Sintetizar o projeto em VHDL;
- Conhecer a arquitetura de um microprocessador;
- Conhecer a modelagem e síntese de um Sistema Digital.

# Método de estudo da disciplina

- Assista às aulas e estude de maneira incremental;
- Leia o material textual: 2 livros e os textos sugeridos para cada tópico;
- Recorra às aulas gravadas e slides se precisar;
- Faça as listas de exercícios;
- Experimente o *design* usando o simulador GHDL;
- Participe do Plantão de Dúvidas e do Fórum de Discussões;
- Não deixe para fazer os projetos na última hora!

# Método de Avaliação

- Nota Final =  $\frac{\sum_{i=1}^3 N_i}{3}$ , onde
  - $N_1$  = média aritmética das Provas de Acompanhamento (PA);
  - $N_2$  = média ponderada dos Projetos submetidos ao Juiz (PR). Cada projeto terá um peso diferente;
  - $N_3$  = nota da Prova Final (PF).
- As PAs e a PF serão assíncronas, feitas no e-Disciplinas e com duração definida por prova;
- Não haverá prova substitutiva;
- A frequência será computada através da realização de atividades pós-aula no e-Disciplinas. O prazo para realizar a atividade pós-aula será de 48 horas. A frequência mínima para aprovação é de 70%.

# Exercício

Vamos ao VHDL Judge!

# Obrigado!



Universidade de São Paulo



DEPARTAMENTO DE ENGENHARIA DE  
COMPUTAÇÃO E SISTEMAS DIGITAIS

**PCS**



# Referências I

-  *Fujitsu Processor A64FX Specifications.*
-  W. Aspray.  
The intel 4004 microprocessor: what constituted invention?  
*Annals of the History of Computing, IEEE*, 19(3):4–15, Jul-Sep 1997.
-  F. Faggin.  
The making of the first microprocessor.  
*Solid-State Circuits Magazine, IEEE*, 1(1):8–21, Winter 2009.

## Referências II

-  Gordon E. Moore.  
*Cramming More Components onto Integrated Circuits*,  
page 56–59.  
Morgan Kaufmann Publishers Inc., San Francisco, CA,  
USA, 2000.
-  J. Nurmi.  
*Processor Design: System-On-Chip Computing for ASICs  
and FPGAs*.  
Springer Netherlands, 2007.
-  D.A. Patterson and J.L. Hennessy.  
*Computer Organization and Design ARM Edition: The  
Hardware Software Interface*.  
The Morgan Kaufmann Series in Computer Architecture  
and Design. Elsevier Science, 2016.

## Referências III

-  John Wakerly.  
*Digital Design: Principles and Practices (4th Edition).*  
Prentice-Hall, Inc., USA, 2005.