

# EA075

## Introdução



Faculdade de Engenharia Elétrica e de Computação (FEEC)  
Universidade Estadual de Campinas (UNICAMP)

Prof. Rafael Ferrari

(Documento baseado nas notas de aula do Prof. Levy Boccato)

# Sistemas Embarcados

- **Sistemas embarcados:** são sistemas de computação inseridos (embutidos) em dispositivos elétricos, eletrônicos e/ou mecânicos de maior porte.
- Possível definição: qualquer sistema computacional que não seja um *desktop*.
- Bilhões de unidades são produzidas anualmente, bem mais que o número de desktops fabricados.
- Exemplos: impressoras, modems, celulares, câmeras digitais, veículos automotivos, caixas eletrônicas, televisões, etc.

# Sistemas Embarcados

- **Características distintivas:**

- Executam uma única função repetidamente (em contraste com um *desktop*).
- Estão sujeitos a restrições de projeto bastante rigorosas com relação a algumas métricas, como tamanho, potência consumida, tempo de resposta, custo, etc.
- Devem continuamente reagir a mudanças no ambiente e computar certos resultados em tempo real, com atraso mínimo.



# Métricas de Projeto

- **Métrica de projeto:** característica ou atributo desejável e mensurável de uma determinada implementação de sistema.
- **Desafio:** construir uma implementação que atenda à funcionalidade desejada e, ao mesmo tempo, que otimize várias métricas de projeto.

# Métricas de Projeto

- Métricas importantes:
  - Custo NRE (*Non-Recurring Engineering cost*): custo de projeto do sistema. Uma vez que o sistema esteja projetado, qualquer número de unidades pode ser fabricado sem incorrer em custo adicional (por isso o termo *não-recorrente*).
  - Custo por unidade: custo de fabricação de cada cópia do sistema, excluindo o NRE.
  - Tamanho: espaço físico exigido pelo sistema (e.g., medido em bytes no caso de *software* ou n<sup>o</sup> de transistores para *hardware*).
  - Desempenho: tempo de execução do sistema.
  - Potência: quantidade de potência consumida pelo sistema.

# Métricas de Projeto

- Métricas importantes:
  - Flexibilidade: habilidade de mudar a funcionalidade do sistema sem que isto demande um alto custo NRE. (*software*, por exemplo, é bastante flexível).
  - Tempo de prototipagem: tempo necessário para a construção de uma versão do sistema que funcione.
  - Tempo para o mercado: tempo necessário para desenvolver o sistema até o ponto que possa ser lançado e vendido aos consumidores (inclui os tempos de projeto, fabricação e testes).
  - Sustentabilidade (Manutenção): capacidade / facilidade de modificar o sistema após seu lançamento inicial, especialmente por projetistas que não participaram do desenvolvimento original.

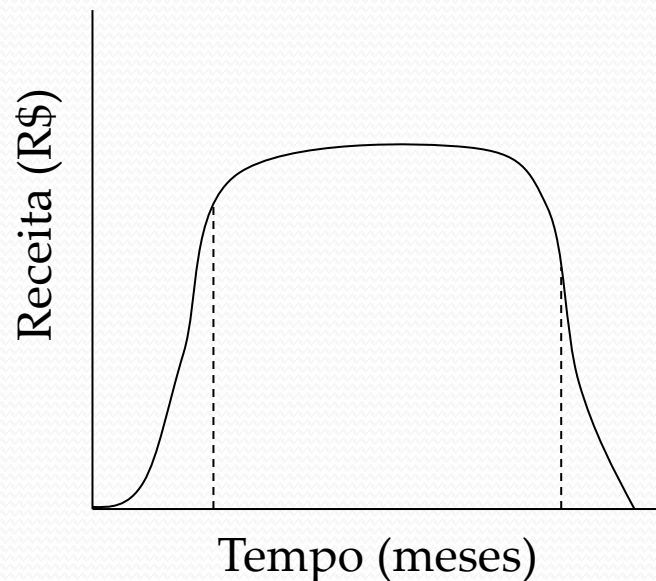
# Métricas de Projeto

- **Importante:** tipicamente, estas métricas competem entre si no sentido que melhorar o sistema em termos de uma métrica usualmente leva a uma degradação com respeito a outra métrica.
- Exemplo: Aumentar a performance pode gerar um aumento na potência consumida pelo sistema.
- Para ser bem-sucedido, um projetista precisa estar familiarizado com uma variedade de tecnologias de *software* e *hardware* a fim de encontrar a melhor implementação da funcionalidade desejada, respeitando o conjunto de restrições imposto ao projeto. Quanto maior seu repertório, melhor.



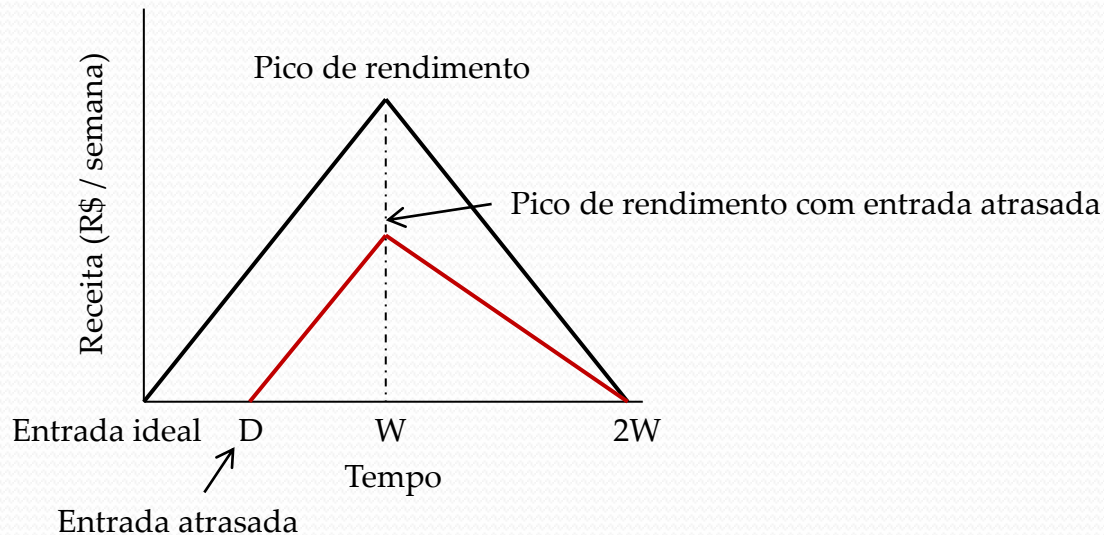
# Janela de Mercado

- Existe um período mais propício para o lançamento do produto, durante o qual espera-se alcançar as maiores vendas.
- Eventuais atrasos no início das vendas podem significar uma perda elevada na receita total.



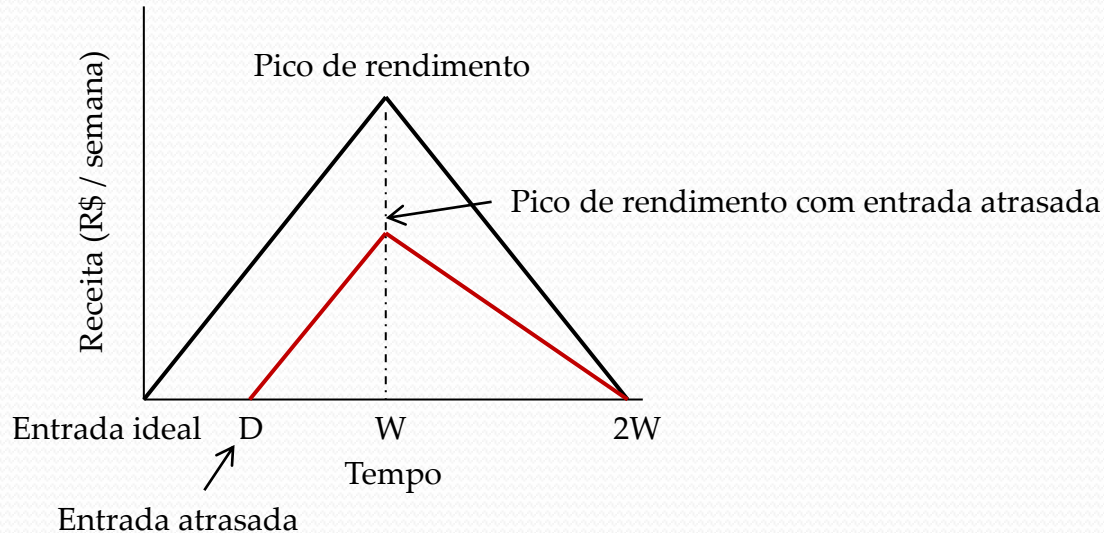
# Janela de Mercado

- Modelo simples



- Qual a perda percentual de receita por causa do atraso no lançamento do produto?
  - Área de cada curva define o rendimento total alcançado com as vendas do produto no tempo em que esteve disponível.
  - A diferença entre as áreas das curvas corresponde à perda de receita.

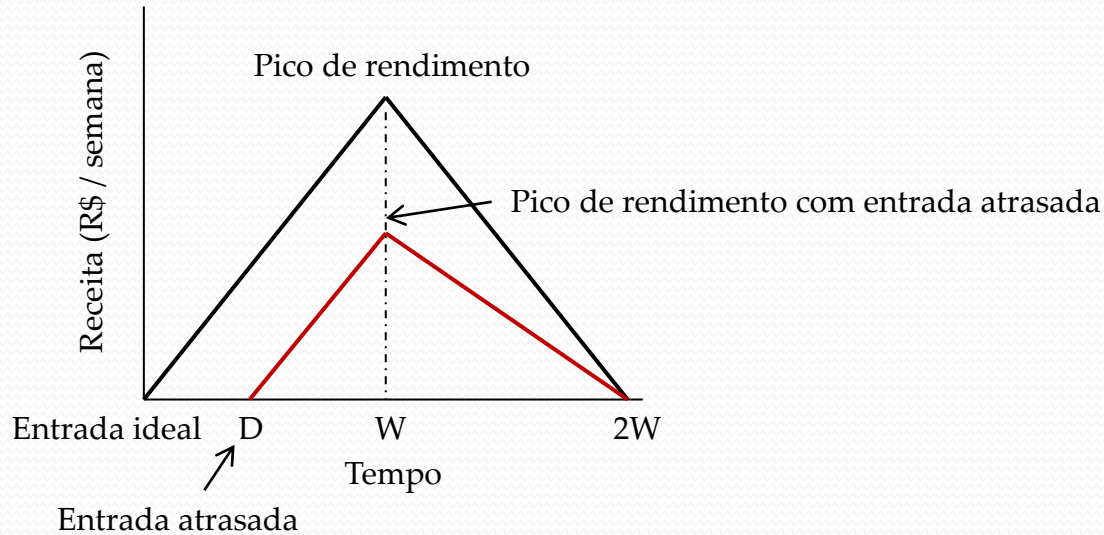
# Janela de Mercado



Considerando um ângulo de subida de  $45^\circ$ ,

- Área (entrada ideal) =  $(1/2) * (2W) * W = W^2$
- Área (entrada atrasada) =  $(1/2) * (2W-D) * (W-D)$
- Perda percentual =  $[W^2 - (1/2) * (2W^2 - 3WD + D^2)] / W^2 \quad (\times 100\%)$   
 $= D * (3W - D) / 2W^2 \quad (\times 100\%)$

# Janela de Mercado



**Exemplo:** tempo de mercado ( $2W$ ) = 52 semanas

atraso ( $D$ ) = 4 semanas

$$\text{perda} = 4 \cdot (3 \cdot 26 - 4) / 2 \cdot 26^2 = 22\%$$

Se o atraso for de 10 semanas,

$$\text{perda} = 10 \cdot (3 \cdot 26 - 10) / 2 \cdot 26^2 = 50\%$$

# Custo NRE e Custo por Produto

- Para uma dada implementação, são conhecidos o custo NRE ( $v_{\text{NRE}}$ ) e o custo por unidade ( $v_u$ ). Logo,

$$\text{Custo total} = v_{\text{NRE}} + v_u \times \text{número de unidades}$$

- Cada produto fabricado, portanto, custou:

$$\begin{aligned} \text{Custo por produto} &= \text{Custo total} / \text{número de unidades} \\ &= v_{\text{NRE}} / \text{número de unidades} + v_u \end{aligned}$$

# Custo NRE e Custo por Produto

- Exemplo:  $v_{\text{NRE}} = \$ 2000$  e  $v_u = \$ 100$

Supondo que 20 unidades foram produzidas,

$$\text{Custo total} = \$ 2000 + \$ 100 * 20 = \$ 4000$$

$$\text{Custo por produto} = \underbrace{\$ 2000 / 20} + \$ 100 = \$ 200$$

O custo NRE traz um custo adicional por produto de \$ 100.

Supondo que 2000 unidades foram produzidas,

$$\text{Custo por produto} = \underbrace{\$ 2000 / 2000} + \$ 100 = \$ 101$$

O custo NRE traz um custo adicional por produto de \$ 1.

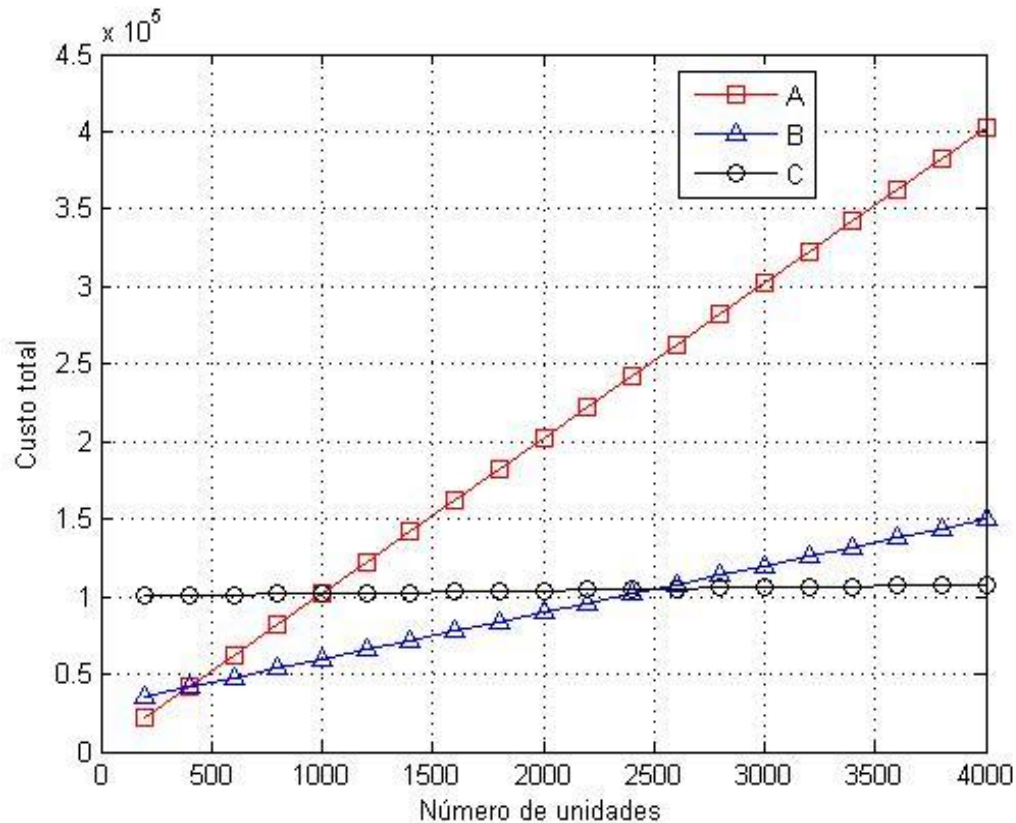


**É possível amortizar o custo NRE se o produto for fabricado em larga escala.**

# Custo NRE e Custo por Produto

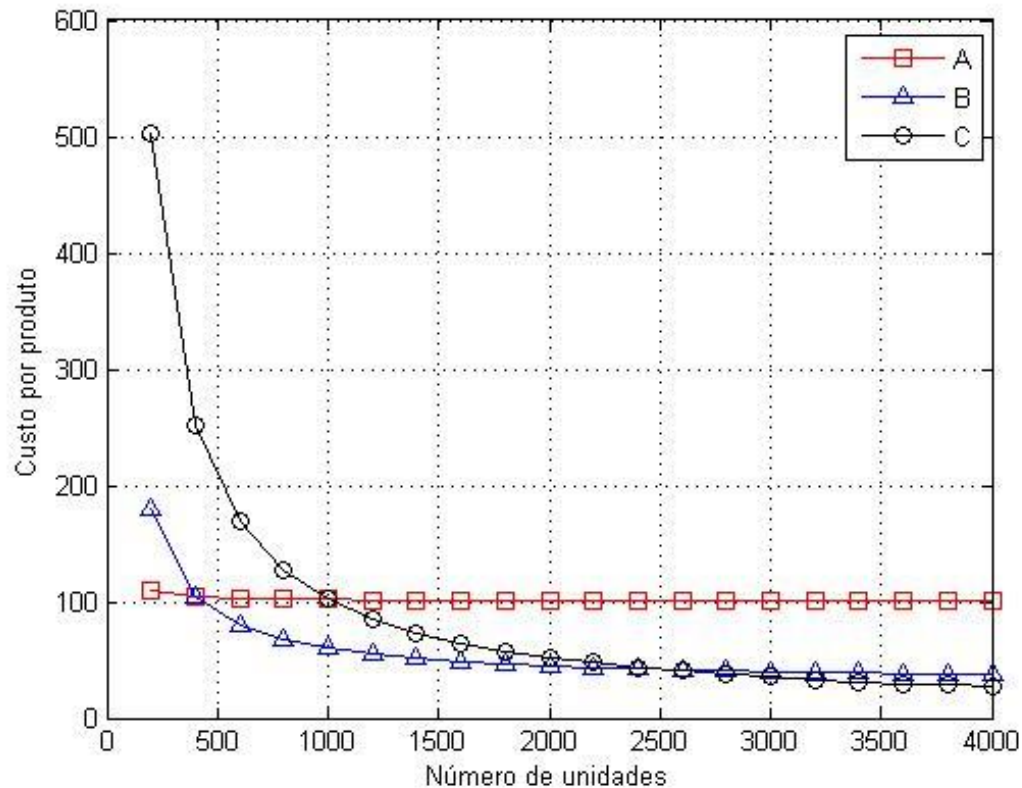
- Suponha que existam três tecnologias para a implementação de um produto:
  - Tecnologia A:  $v_{\text{NRE}} = \$ 2.000$  e  $v_u = \$ 100$
  - Tecnologia B:  $v_{\text{NRE}} = \$ 30.000$  e  $v_u = \$ 30$
  - Tecnologia C:  $v_{\text{NRE}} = \$ 100.000$  e  $v_u = \$ 2$
- Quando é vantajoso utilizar cada uma destas tecnologias?

# Custo NRE e Custo por Produto





# Custo NRE e Custo por Produto



# Desempenho

- Métrica que informa quanto tempo o sistema precisa para executar a tarefa desejada.
- Algumas medidas, como a frequência do relógio interno (*clock*) ou o número de instruções por segundo que o processador consegue executar têm influência no desempenho mas não são, necessariamente, medidas de desempenho.
- Exemplo: câmera digital – mais importante do que conhecer estas medidas, o parâmetro crítico é a velocidade de processamento de uma imagem.

# Desempenho

- Duas medidas importantes:
  - **Latência** (ou tempo de resposta) = tempo entre o início e a conclusão da tarefa. (e.g., uma câmera processa uma imagem em 0,25 segundos).
  - **Throughput** (taxa, vazão de produção) = número de tarefas que podem ser processadas em uma unidade de tempo. (e.g., uma câmera processa 4 imagens por segundo).

**OBS:** É possível aumentar o *throughput* utilizando paralelismo.

- Aceleração (*speedup*) = desempenho de *B* / desempenho de *A*

# Tecnologias

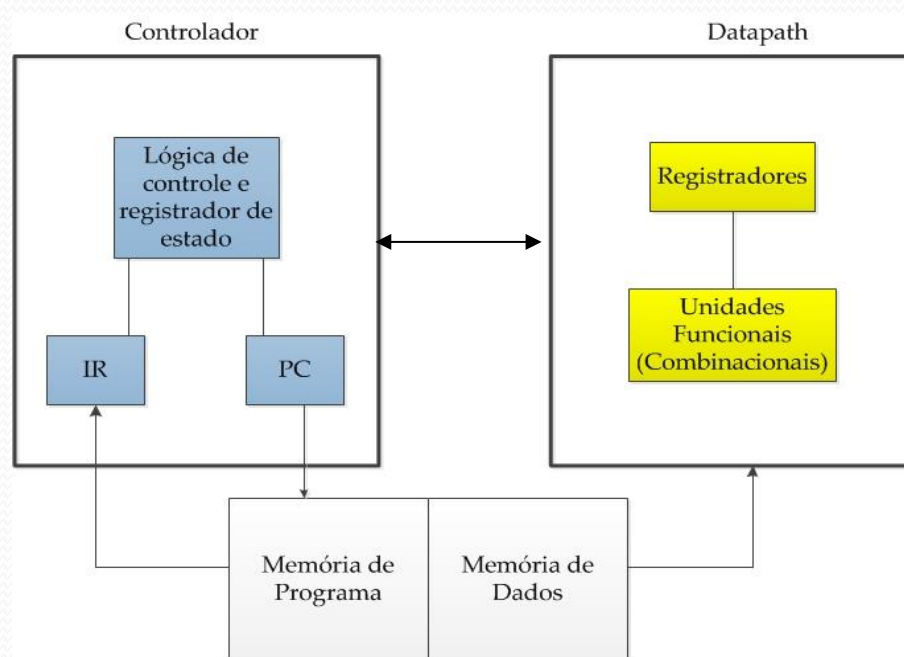
- Tecnologia = meio (forma) de realizar uma tarefa, especialmente usando técnicas de processamento, métodos ou conhecimentos técnicos.
- Três tipos:
  - Tecnologia de processador;
  - Tecnologia de circuito integrado (IC);
  - Tecnologia de projeto.

# Tecnologia de Processador

- Refere-se à organização da máquina de computação empregada para implementar a funcionalidade desejada do sistema.
- **Processador:** circuito digital projetado para realizar tarefas de computação.
- Dois elementos básicos:
  - **Datapath:** responsável por armazenar e manipular dados.
  - **Controlador (Unidade de controle):** responsável por sequenciar/comandar coerentemente a execução das operações.

# Tecnologia de Processador

- Um projetista de sistemas embarcados dispõe de três opções básicas de processadores:
  - **Processadores de propósito geral (genéricos):** a tarefa que irá realizar depende da sequência de comandos armazenada em forma de código em sua memória.

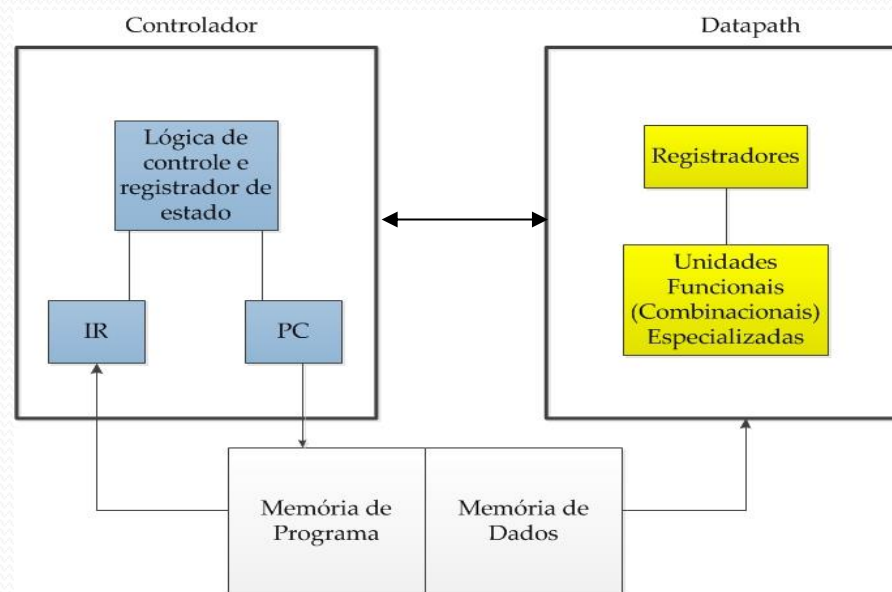


# Tecnologia de Processador

- **Processadores de propósito geral (genéricos)**
  - Dispositivos programáveis utilizados em uma variedade de aplicações.
  - **Benefícios:**
    - Baixo custo NRE (apenas a preparação do programa);
    - Tempo para o mercado é pequeno.
    - Alta flexibilidade (trocar funcionalidade = trocar programa).
    - Baixo custo por produto para pequenas quantidades.
  - **Cuidados:**
    - Tamanho e consumo de potência podem ser elevados por conta da presença de *hardware* desnecessário.
    - Desempenho pode ser lento em algumas aplicações.
  - Abordagem baseada em *software*.

# Tecnologia de Processador

- Um projetista de sistemas embarcados dispõe de três opções básicas de processadores:
  - **Processadores especializados (ASIPs – Application-Specific Processors):** possuem um conjunto de instruções/unidades funcionais específico para uma determinada classe de aplicações.





# Tecnologia de Processador

- Processadores especializados (de aplicação específica)
  - Processadores programáveis otimizados para uma classe de aplicações particular tendo em comum algumas características.
  - **Benefícios:**
    - Boa flexibilidade;
    - Desempenho, consumo de potência e tamanho podem ser satisfatórios.
  - **Cuidados:**
    - Verificar a disponibilidade de compiladores e ferramentas de *debug* para uso de um determinado ASIP.

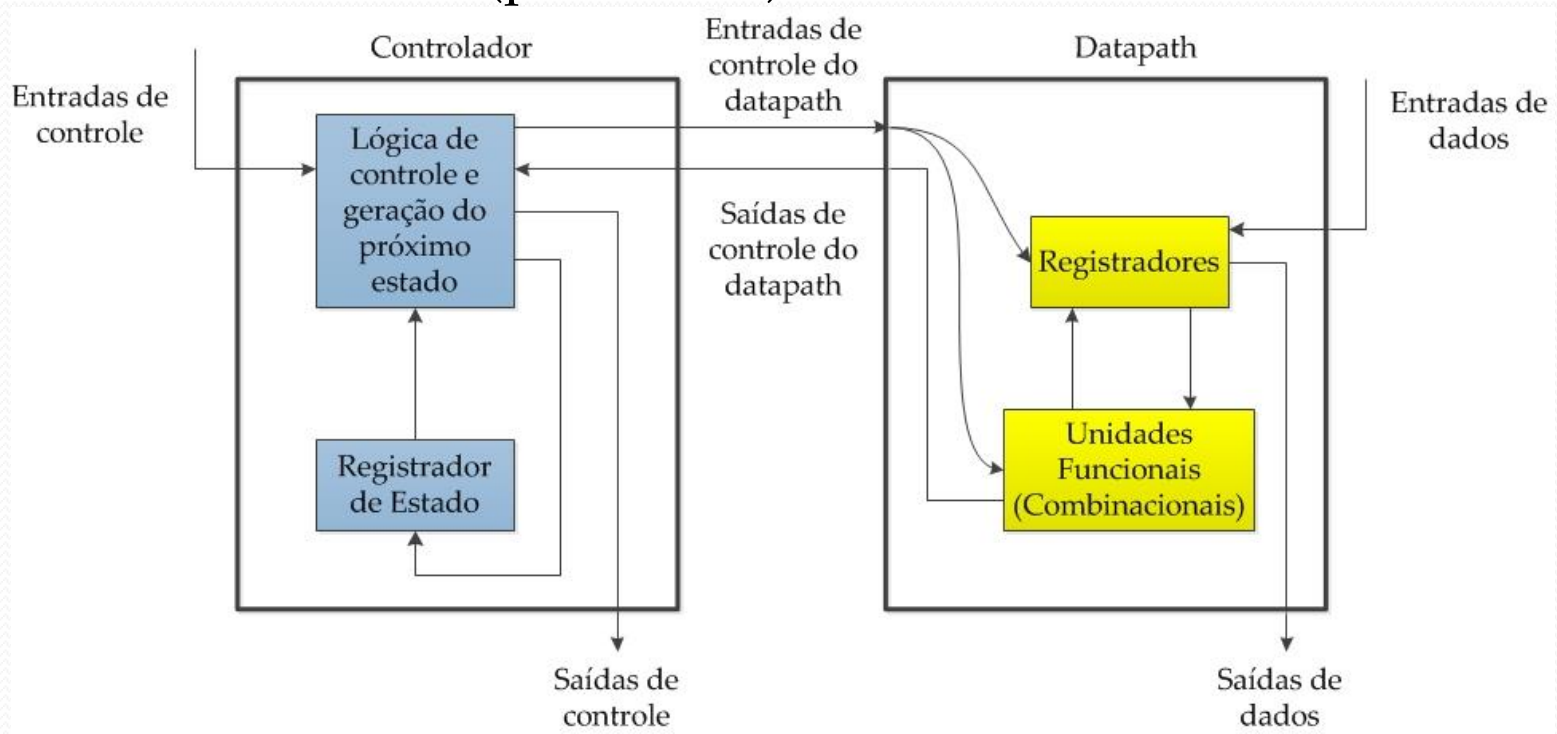
# Tecnologia de Processador

- Exemplos

- **Microcontroladores:** processadores otimizados para tarefas de controle embarcado, as quais envolvem o monitoramento e ajuste de sinais de controle de um bit.
  - *Datapath* mais simples que se sobressai na realização de operações sobre bits e leitura/escrita de bits externos.
- **DSPs:** processadores projetados para realizar operações sobre sinais digitais – otimizados para a execução de operações matemáticas de forma intensa.
  - Conta com *hardware* que permite o acesso a posições de memória sequenciais em paralelo com outras operações para acelerar a execução das operações.

# Tecnologia de Processador

- Um projetista de sistemas embarcados dispõe de três opções básicas de processadores:
  - **Processadores dedicados:**
    - ✓ Personalizados
    - ✓ Padronizados (periféricos): timer, UART, PWM, A/D, etc.



# Tecnologia de Processador

- **Processadores dedicados**

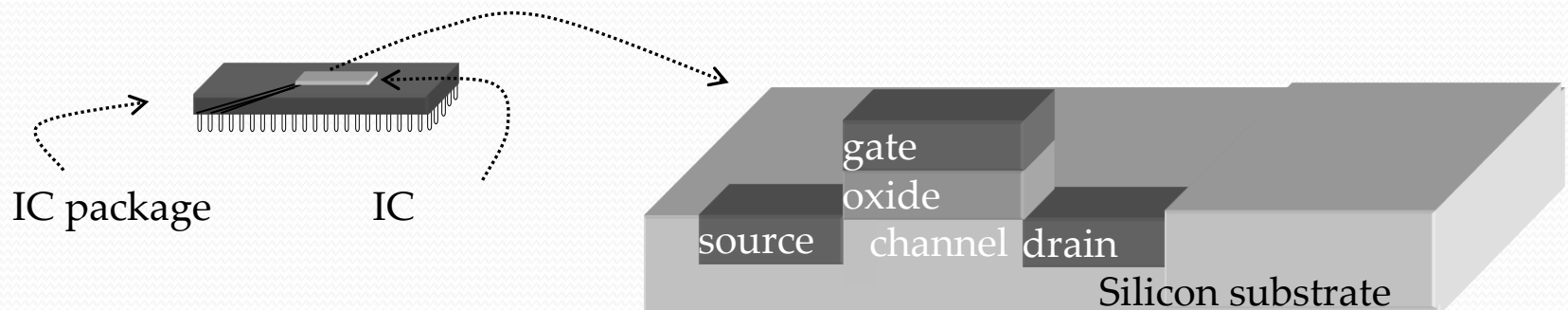
- Circuitos digitais projetados com a finalidade de realizar uma única tarefa. Não possuem memória de programa, pois as “instruções” que irá realizar já estão implementadas no próprio circuito.
- **Benefícios:**
  - Melhor desempenho: mais rápidos;
  - Melhor ocupação de área no circuito integrado (menor tamanho);
  - Menor consumo de potência;
  - Custo por produto pode ser pequeno para grandes quantidades.
- **Cuidados:**
  - Alto custo de projeto (NRE);
  - Tempo para lançamento do dispositivo no mercado será maior;
  - Baixa flexibilidade;
  - Custo por produto pode ser elevado para produção em baixa escala.
- Abordagem baseada em *hardware*.

# Tecnologia de Circuito Integrado

- Envolve a maneira como mapeamos uma implementação digital (em nível de transferência de registradores ou portas lógicas) em um circuito integrado (CI).
- CI: dispositivo semicondutor formado por um conjunto de transistores conectados e por outros componentes.
- As diferentes tecnologias de CI diferem no quão personalizado é o CI para uma tarefa em particular.

# Tecnologia de Circuito Integrado

- Transistor MOSFET básico:
  - Composto por três camadas:
    - A fonte e o dreno estão no substrato de silício;
    - Há uma camada de isolamento (dióxido de silício);
    - O gate, constituído de polisilício, está acima da camada de isolamento.
    - A tensão aplicada no gate determina se uma corrente de elétrons flui da fonte para o dreno.



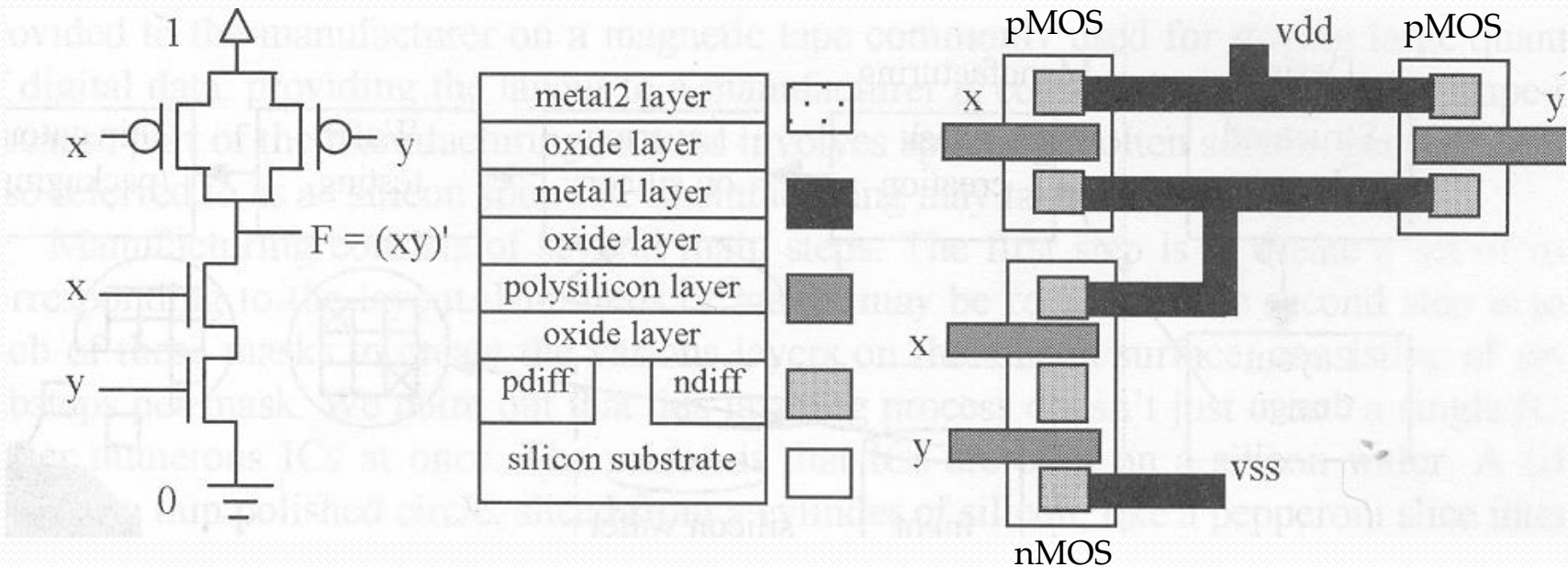
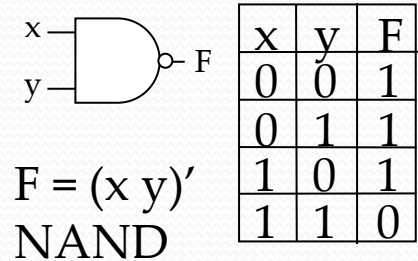
# Tecnologia de Circuito Integrado

- Exemplo: porta NAND

- Transistor nMOS operando como chave:

- “0” no gate: chave aberta;
    - “1” no gate: chave fechada;

- Transistor pMOS: lógica complementar.



# Tecnologia de Circuito Integrado

- A produção de um CI consiste na construção de todas as camadas de metal / silício que formam os transistores e as conexões entre eles.
- As tecnologias de CI se distinguem pelo número de camadas que o projetista precisa projetar para a implementação de um processador.
- **Importante:** pode-se usar qualquer tecnologia de CI para implementar um processador (seja ele de propósito geral, dedicado ou um ASIP).

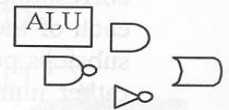


# Tecnologia de Circuito Integrado

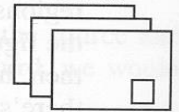
- **CI totalmente personalizado (VLSI, *very large scale integrated circuit*):** o projetista precisa criar o layout completo do processador.
- *Layout*: especifica o posicionamento de cada transistor e de cada conexão entre os transistores no CI desejado (como se fosse um mapa).
- A partir do *layout*, um conjunto de máscaras é construído, o qual define todas as camadas que precisam ser criadas no CI.

## START

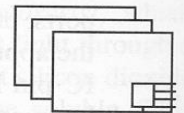
Designers create layouts for basic components.



Designers place the components, resulting in masks.



Designers provide the connections among components, which are translated to masks.



The masks are sent to the fabrication plant to produce ICs.



ICs are now ready to be tested/used.



# Tecnologia de Circuito Integrado

- **CI totalmente personalizado (VLSI, *very large scale integrated circuit*):**
  - **Benefícios:**
    - Excelente desempenho;
    - Tamanho e consumo de potência reduzidos.
  - **Desvantagens:**
    - Alto custo NRE;
    - Tempo para o mercado é bastante longo em relação às demais tecnologias (a fabricação do circuito pode levar vários meses).
  - Esta abordagem é usada predominantemente para aplicações de desempenho extremamente crítico ou para sistemas produzidos em grandes quantidades (para amortizar o NRE).

# Tecnologia de Circuito Integrado

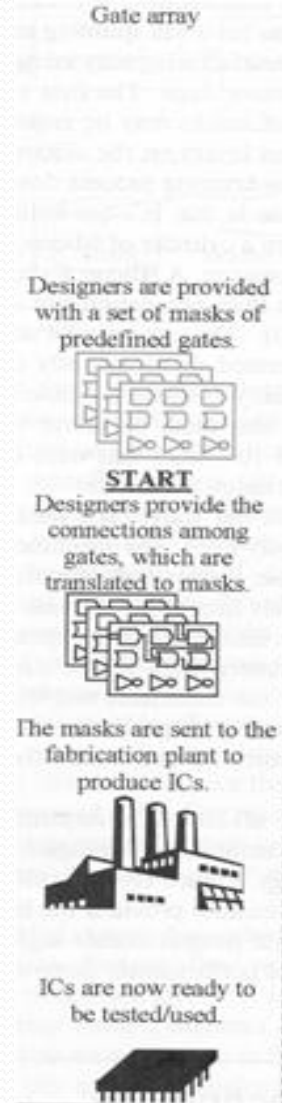
- **Semi-custom:**

- Também chamados de ASICs (*application specific integrated circuits*).
- As camadas inferiores do CI estão total ou parcialmente construídas, deixando o acabamento das camadas superiores (*layout*) a cargo do projetista do sistema embarcado.
- Dois tipos:
  - Gate-array
  - Standard Cell

# Tecnologia de Circuito Integrado

- Gate-array:

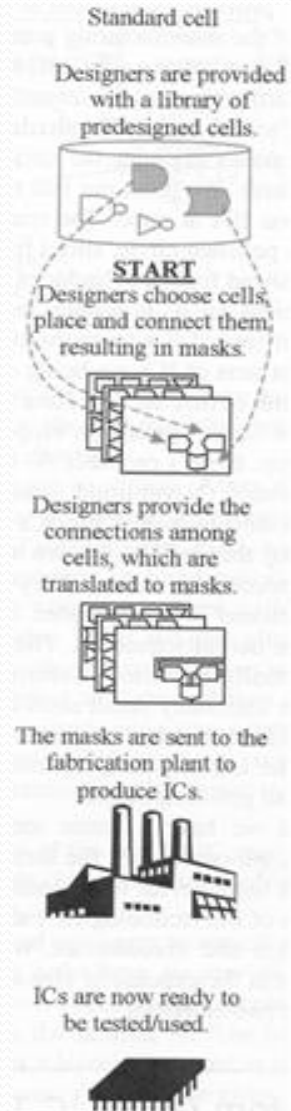
- Todas as portas lógicas do CI já estão posicionadas, deixando ao projetista a tarefa de conectar as portas (roteamento) de tal forma a implementar o circuito digital desejado.



# Tecnologia de Circuito Integrado

- **Standard Cell:**

- Funções lógicas comuns (células) estão disponíveis.
- O projetista deve: (1) decidir quais células vai utilizar, (2) de que modo estas células serão posicionadas e (3) como conectá-las.



# Tecnologia de Circuito Integrado

- **Semi-Custom:**

- É a opção mais popular no momento.
- Bom desempenho, tamanho e consumo de potência satisfatórios
- Menor custo NRE se comparado à VLSI.
- Ainda requerem de semanas a meses para serem fabricados.

# Tecnologia de Circuito Integrado

- **Dispositivo lógico programável (PLD):**
  - Projetistas adquirem um CI com todas as camadas prontas.
  - A “programação” envolve a criação ou destruição de conexões entre fios que ligam as portas
  - Field-Programmable Gate Array (FPGA) é uma opção muito popular.
  - **Benefícios:** baixo custo NRE (bastante usado para prototipagem), rápida confecção do IC.
  - **Desvantagens:** tamanho, consumo de potência e custo por produto elevados, além de desempenho mais lento.

# Tecnologia de Projeto

- Procedimento pelo qual convertemos o conceito da funcionalidade desejada para o sistema em uma implementação.
- Processo de projeto de um sistema (*top-down*):
  - Especificação do sistema em linguagem natural ou em linguagem executável de alto nível (e.g. C).
  - Especificação comportamental: porções da especificação são distribuídas para vários processadores dedicados ou de propósito geral.
  - Especificação no nível de transferência entre registradores (RT): nível Assembly ou de conexões entre registradores + máquinas de estados.
  - Especificação lógica: equações Booleanas.
  - Implementação: código de máquina para processadores de propósito geral e lista de ligações do circuito lógico para processadores dedicados.

Nível decrescente de abstração





# Tecnologia de Projeto

- **Compilação / Síntese:** permite ao projetista especificar a funcionalidade em uma forma abstrata (de alto nível) e gera detalhes de implementação de níveis mais baixos.
  - **Sistema:** especificação abstrata do sistema → conjunto de programas para processadores gerais ou dedicados.
  - **Comportamental:** programa sequencial → FSM (*Finite State Machine*) e RT (*Register Transfer*) ou programa sequencial → código assembly (compilador).
  - **RT:** FSM e RT → *datapath* de componentes RT e equações Booleanas do controlador.
  - **Lógico:** expressões Booleanas → circuito lógico (lista de ligações entre portas lógicas).

# Tecnologia de Projeto

- **Bibliotecas / Propriedade Intelectual:** envolve o reuso de implementações pré-existentes. Utilizar bibliotecas pode melhorar a produtividade se o tempo necessário para encontrar, adquirir, integrar e testar um item de biblioteca for inferior ao de projetar aquele item.
  - **Sistema:** sistemas completos que implementam uma determinada funcionalidade.
  - **Comportamental:** componentes comumente usados: interfaces de barramento, controladores de *display*, controladores de motor de passo, processadores de propósito geral (cores).
  - **RT:** registradores, multiplexadores, decoders, ALUs e unidades funcionais.
  - **Lógico:** *layouts* para portas lógicas e células lógicas.

# Tecnologia de Projeto

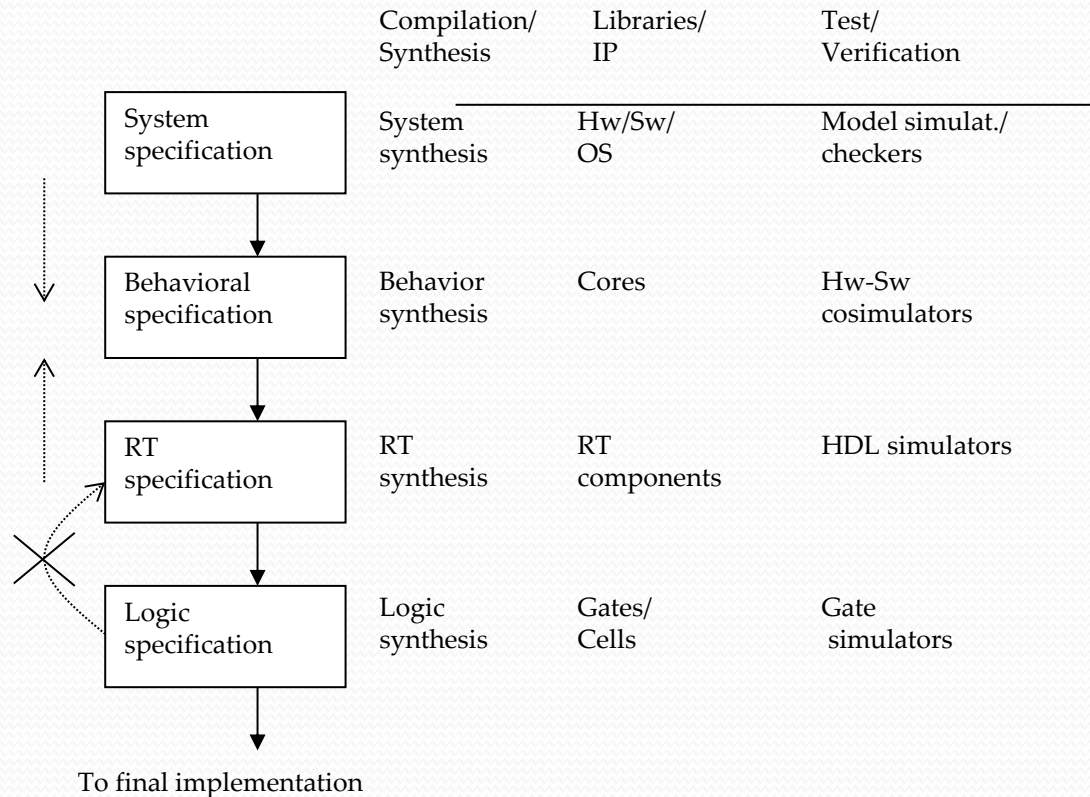
- **Teste / Verificação:** envolve garantir que a funcionalidade está correta – evita tarefas de debugagem em níveis mais baixos de abstração e, conseqüentemente, a necessidade de refazer etapas em níveis mais altos.
  - **Sistema:** simulação usando um modelo computacional independente da tecnologia do processador.
  - **Comportamental:** simuladores HDL (*Hardware Description Level*) e simuladores de processadores de propósito geral são empregados para verificação combinada de hardware e software.
  - **RT:** simuladores HDL executam descrições de nível RT e fornecem formas de onda de saída mediante estímulos de entrada.
  - **Lógico:** simuladores de circuitos lógicos geram formas de onda de saída a partir de sinais de entrada. Simuladores de processadores de propósito geral executam código de máquina.

# Tecnologia de Projeto

*Compilation/Synthesis:* Automates exploration and insertion of implementation details for lower level.

*Libraries/IP:* Incorporates pre-designed implementation from lower abstraction level into higher level.

*Test/Verification:* Ensures correct functionality at each level, thus reducing costly iterations between levels.

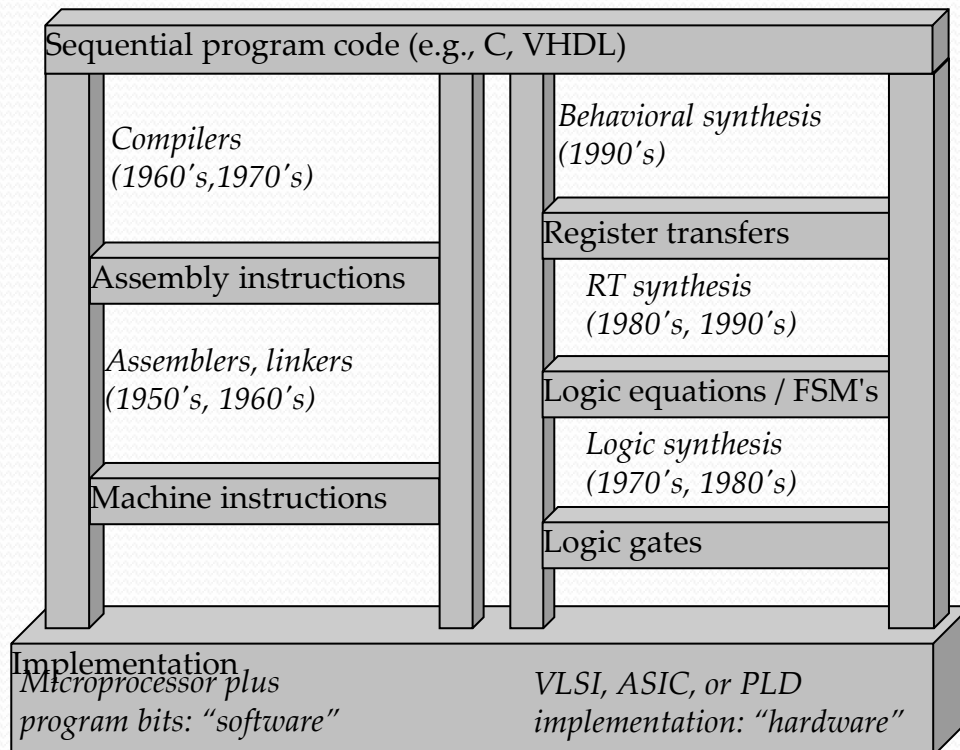


# Co-projeto de Software e Hardware

- Desafio do projeto de sistemas embarcados: otimização de métricas que competem entre si.
- O projetista precisa estar familiarizado com várias tecnologias de projeto.
- Divisão histórica: especialistas em *software* (processadores de propósito geral) e especialistas em *hardware* (processadores dedicados).
- Divisão do sistema entre subsistemas de *hardware* e de *software* nos estágios iniciais do projeto: ineficiência na otimização das métricas de projeto.

# Co-projeto de Software e Hardware

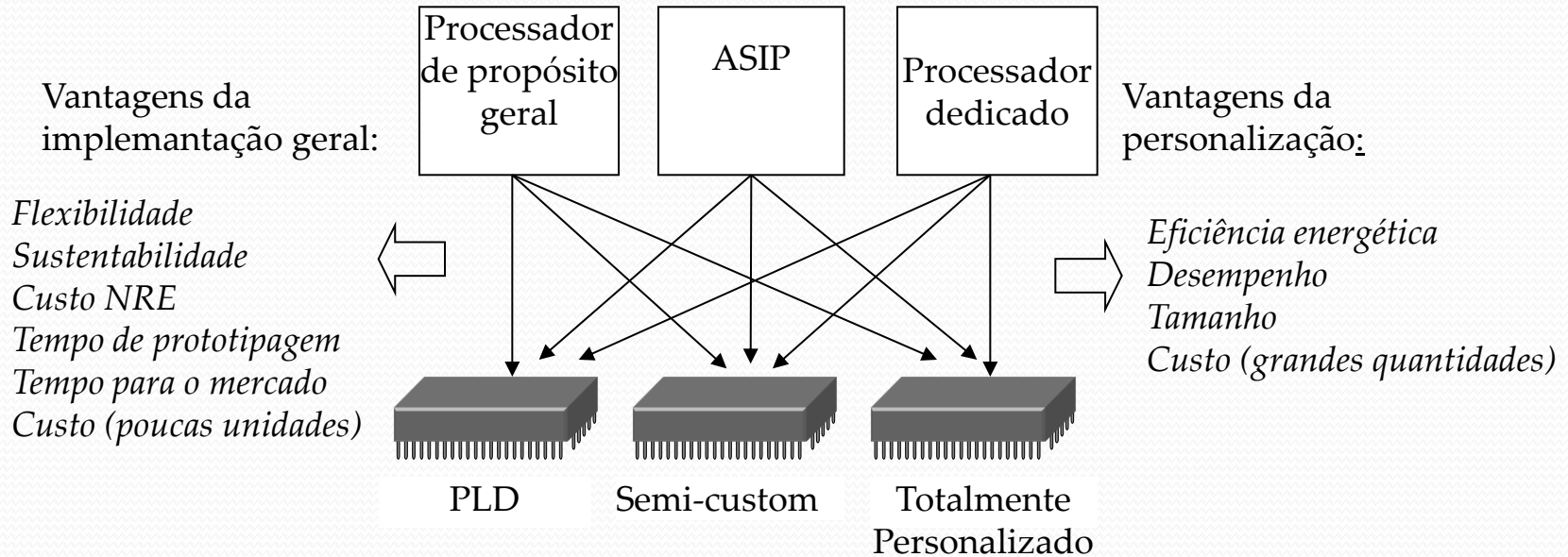
- A disponibilidade de ferramentas de síntese nos níveis comportamentais e RT possibilitou uma visão unificada das abordagens de projeto baseadas em *software* e *hardware*.



# Co-projeto de Software e Hardware

- **Ponto fundamental:** A escolha entre *hardware* e *software* para a implementação de uma função específica é simplesmente um compromisso (*trade-off*) entre várias métricas de projeto, como desempenho, potência, custo NRE e, especialmente, flexibilidade. Não há uma diferença fundamental entre o que *software* e *hardware* podem implementar.
- Co-projeto de *Hardware* e *Software*: um campo que prega uma visão unificada de *hardware* e de *software* e desenvolve ferramentas de síntese e simulação que permitem o co-desenvolvimento de sistemas usando *software* e *hardware* simultaneamente.

# Independência entre tecnologias





# Créditos

- Algumas figuras foram extraídas de F. Vahid e T. Givargis, *“Embedded System Design: A Unified Hardware/Software Introduction”*, John Wiley & Sons, 2001.

<http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471386782.html?filter=TEXTBOOK>