

Introdução aos Sistemas Embarcados

Sergio Cavalcante

svc@cin.ufpe.br: Usem assunto com [ese]

<http://www.cin.ufpe.br/~svc/ese>

98835.0950

Programa

- Introdução aos sistemas embarcados

- O que são?
- Arquitetura básica de um S.E.
- Tecnologias empregadas
- Metodologias de projeto
- Arquiteturas Padrão
- Sistemas operacionais de tempo real

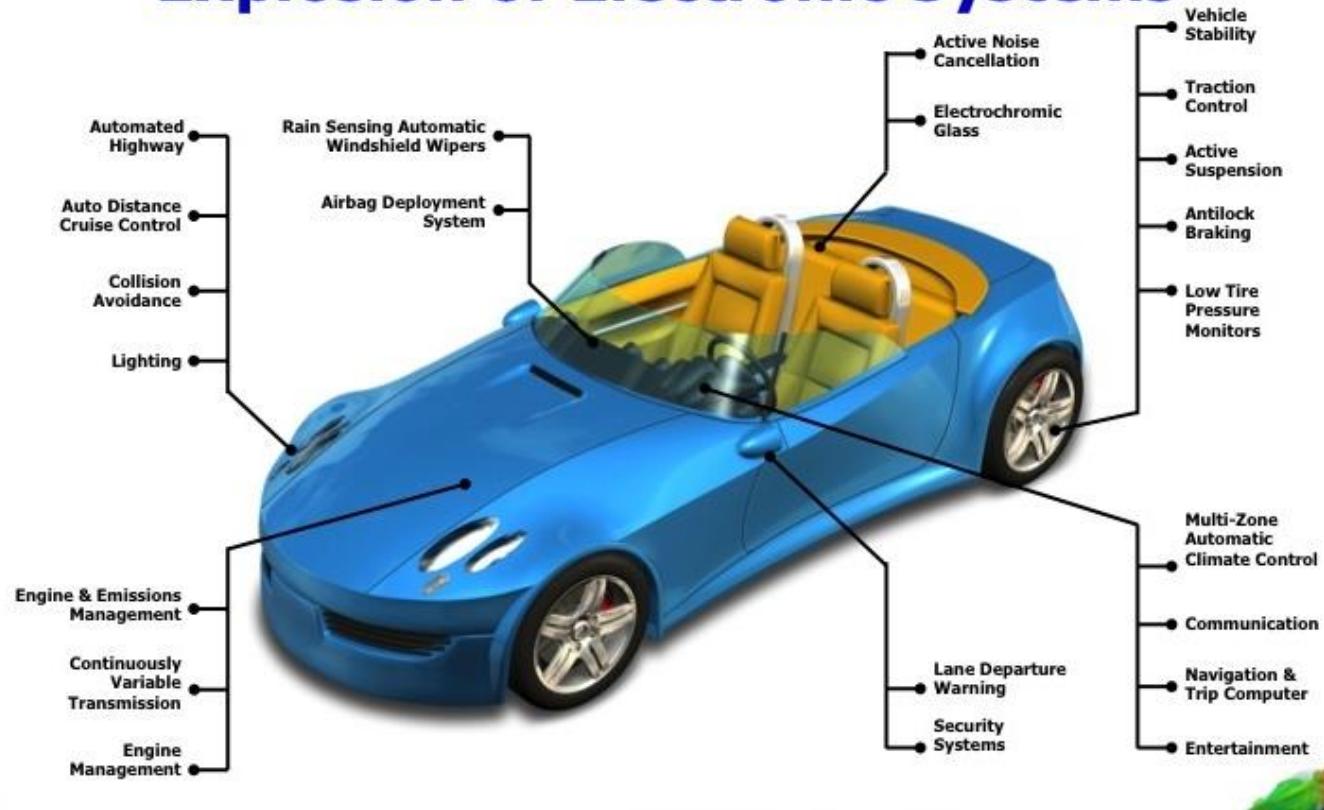
Intro. a Sistemas Embarcados

- Visão geral
- Projeto e Arquitetura
 - Visão geral
 - Áreas de aplicação
 - Características
 - Arquitetura básica
 - Mercado
 - Projeto e Arquitetura

Sistemas Embarcados

- São sistemas computacionais que estão inseridos em máquinas ou em sistemas maiores
- Embutidos em outros equipamentos:
 - Forno de micro-ondas, ar condicionados, carros (controle de transmissão, injeção e ignição eletrônica, suspensão ativa, freio ABS), controle remoto, smart TV, câmeras digitais, vídeo games, calculadoras, máquinas de lavar, sistemas de iluminação, etc.
- Encontrado em quase todas as aplicações que necessitam de algum tipo de **controle** ou que necessitem de **processamento digital de sinais**.

Explosion of Electronic Systems



Carros modernos
podem ter até 150
ECUs

<https://www.embitel.com/blog/embedded-blog/automotive-control-units-development-innovations-mechanical-to-electronics>

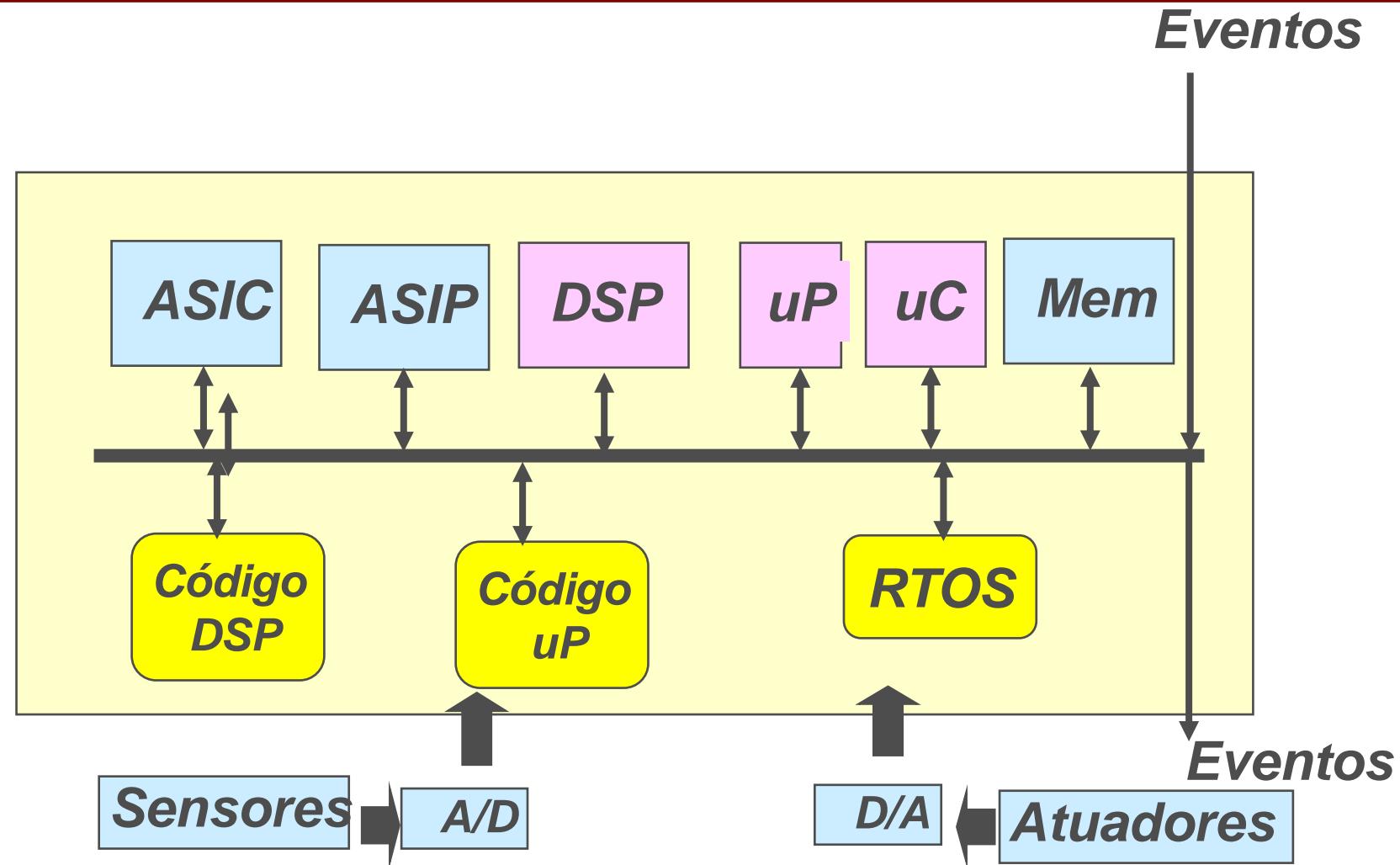
Características

- Funcionalidade única, executada repetidamente
- Entrada/Saída intensivo
- Executa tarefas em paralelo
- Restrições de projeto mais rígidas:
 - Custo, tamanho, peso, desempenho, potência dissipada, etc.
- Tempo real:
 - O tempo para fornecer resultados é determinado pelo tempo que o ambiente pode esperar.
 - Sistemas em que têm aspectos temporais na especificação.
- Sistemas reativos
 - Reagem continuamente a estímulos externos

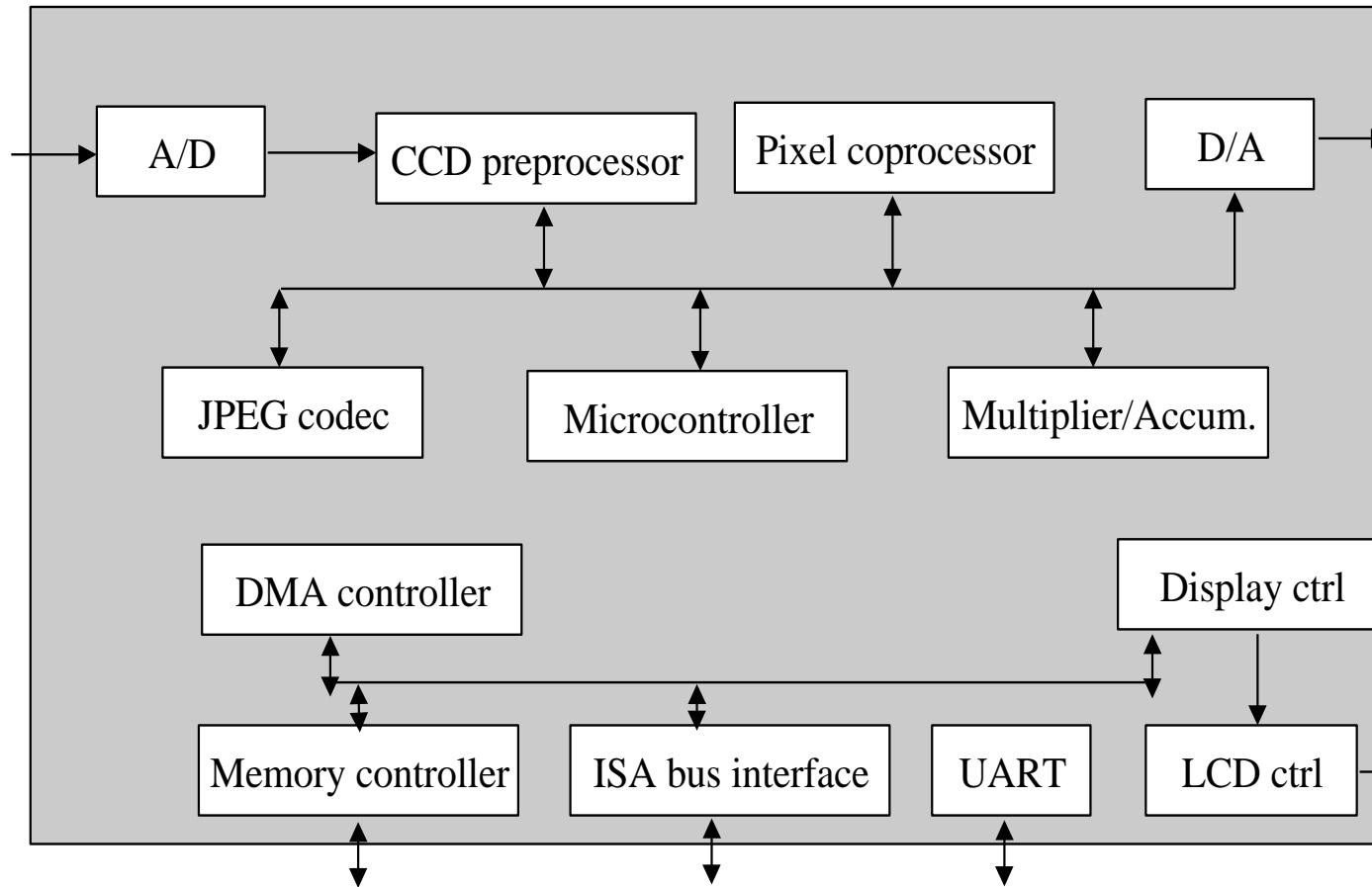
Arquitetura Básica

- Arquitetura de Hardware
 - Forte comunicação com o ambiente
 - Forte restrição de recursos, tamanho, potência, peso...
- Arquitetura de Software
 - Tratamento rápido de interrupção
 - Sistemas operacionais de tempo real
 - Softwares eficientes em tamanho e desempenho

Arquitetura de Hardware



Exemplo: Câmera Digital



Exame 1

- O que são Sistemas Embarcados?
- O que é um Sistema de Tempo Real?
- O que são Sistemas Reativos?
- Você diria que todo S.E. é também um Sistema de Tempo Real?
E é um Sistema Reativo?
- É fundamental a execução de tarefas em paralelo em um S.E.? Porquê?
- Como você compara o mercado de S.E. em relação ao de desktops?

Intro. a Sistemas Embarcados

- Visão • Visão geral
 - Áre . Projeto e Arquitetura
 - Car - Metodologia de projeto
 - Arc - Hardware
 - Mel - Software
- Proje

Metodologia de Projeto

- Principais diferenças entre o projeto de S.E. e projeto de aplicações para desktops
 - Requisitos não-funcionais são fundamentais: preço, tamanho, peso, potência,...
 - Flexibilidade: plataforma não definida, vários tipos de S.O., controle total da máquina.
 - Grande preocupação com previsibilidade no uso de recursos
 - Sistema muito restrito: eficiência no uso de recursos é fundamental

Exemplo

- Projeto de um controle remoto de televisão (infra-vermelho)
 - Controle simples com 3 botões:
 - Liga/desliga
 - Seleção de canais
 - Opera com bateria
 - Deve ser leve
 - Controla a televisão por infravermelho

Exemplo: Restrições

- Protocolo de comunicação com a televisão
 - Proprietário
 - Uso de ROM fornecida pelo cliente
- Satisfazer as especificações temporais fornecidas pelo cliente:
 - Código de assinatura
 - Comando

Exemplo: Requisitos do cliente

- Funcionamento a Bateria:
 - 2 baterias AAA
 - Duração: 10.000 pressões nos botões
- Características do produto final:
 - Peso < 100 gramas
 - Dimensão: 10cm X 5cm X 1.5 cm
 - Material: plástico de alta densidade
 - Botão liga/desliga: vermelho e circular
 - Botão canais: preto e quadrado
 - Deve ser robusto o suficiente para cair de 1,5 metros sem danificar

Exemplo: Requisitos do cliente

- Características do produto final:
 - Sinais de infravermelho transmitidos conforme especificação do cliente
 - Controle deve funcionar a 10 metros da TV quando posicionado até 45 graus da TV e 20 graus do sensor
 - O sinal de infravermelho deve ser transmitido até 20 msec após botão pressionado

Exemplo: Definição do Hw e Sw

- Processador:
 - Microcontrolador de 8 bits
- Sistema Operacional:
 - Não há necessidade
- Linguagem de programação:
 - C
- Bibliotecas de software:
 - Nenhuma
- Componentes de Hardware:
 - Botões
 - LED infravermelho

Exame 2

- O que diferencia o projeto de S.E. do projeto de aplicativos para computadores?
- Mencione pelo menos 5 restrições de projeto comuns em S.E.?
- Quais as vantagens e desvantagens de usar Software e/ou Hardware para implementar um S.E.?

Intro. a Sistemas Embarcados

- Visão • Visão geral
- Projeto • Projeto e Arquitetura
 - Me
 - Har
 - Sof
- Projeto e Arquitetura
 - Metodologia de projeto
 - Hardware
 - Processadores
 - Memória
 - Periféricos
 - Software

Dispositivos Processadores

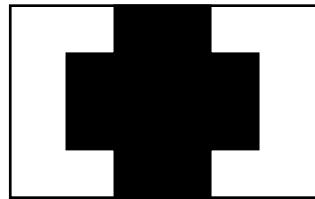
- Transformam, movem dados, tomam decisões e/ou executam ações.
- Não precisam ser programáveis
- Tipos:
 - Processadores de uso geral
 - Processadores de Aplicação Específica
 - Processadores de propósito único

Tecnologia

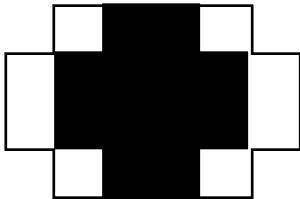
Processadores variam na adequação ao problema



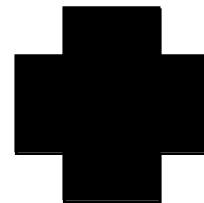
Funcionalidade
Desejada



Processador
de uso geral



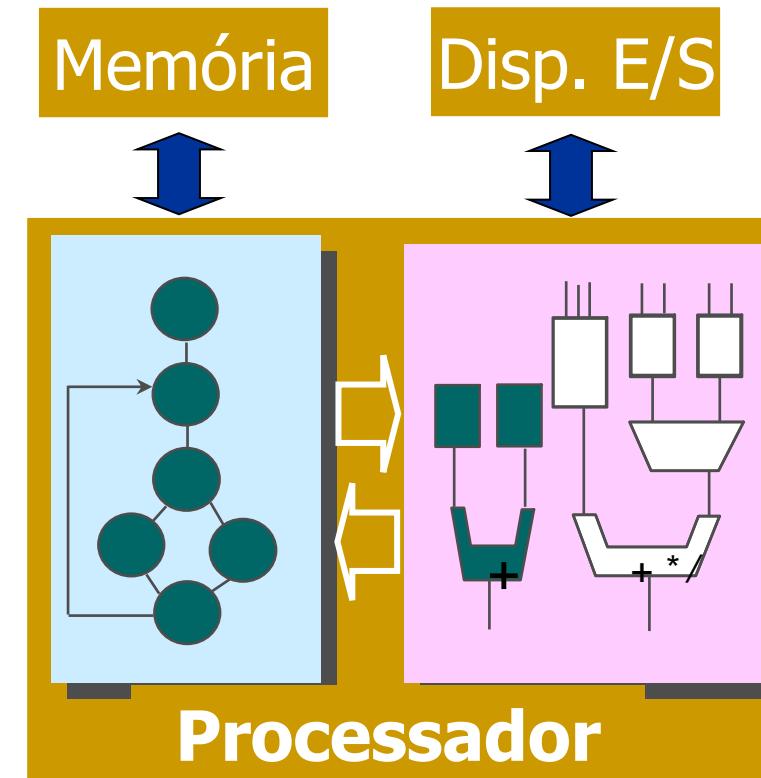
Processador de
aplicação
específica



Processador
de propósito
único

Processadores de uso geral

- Programados via software
- Memória para dados e programa
- Vantagens
 - Pequenos *time-to-market* e custo não recorrente
 - Alta flexibilidade
- Ex: PowerPC, Pentium, Z80



SE::P & A::Hw::Processadores de Uso Geral

Processadores p/ desktops

- Facilidade para desenvolvimento de software
- Necessário adicionar muitos dispositivos auxiliares
- Conjunto de instruções não ideal
- Alto consumo de energia

SE::P & A::Hw::Processadores de Uso Geral **Processadores** **Embarcados**

- Proc. de uso geral adaptados para sistemas embarcados:
 - Dispositivos internos
 - Menor potência
 - Facilidade para desenvolver software

Produto	Clock (MHz)	No. I/O	Portas Seriais	Timers/Contad.	Canais DMA	WDT	Controle Interrupção	Refresh DRAM
80386DX	16,20,25,33	0	Não	0	0	Não	Não	Não
80386EX	25	24	3	3	2	Sim	Sim (8259A)	Sim

Proc. de Aplicação Específica (ASIPs)

- Processador programável otimizado para uma classe de problema
- Características
 - Memória interna
 - Unidade de Execução otimizada
 - Periféricos especiais internos
- Vantagem
 - Bom compromisso entre flexibilidade, velocidade, tamanho e potência
- Ex: Microcontroladores (ex.Nitron, 8051)
DSPs (Digital Signal Processors)

Proc. de Aplicação Específica (ASIPs)

- Projeto de ASIPs:
 - O ASIP e seu compilador são projetados em paralelo (Hw-Sw co-design)
 - Melhor escolha de implementação de instruções (em hardware ou em software)

Microcontroladores

- Não há consenso sobre a diferença de microcontroladores e processadores embarcados
- Nossa definição:
 - microcontroladores não são derivados de famílias de processadores de propósito geral e têm, normalmente, um poder de processamento menor.
- Para diversificar as opções, lançam-se famílias de microcontroladores

Família Nitron

- Lançada pela Motorola Semicondutores (Brasil) em setembro/2002
- Vendeu mais de 1 milhão de unidades até dez/2002
- Mais de 40 milhões encomendados
- Prêmio "Product of the Year" como melhor lançamento do ano de 2002 (revista Electronic Products)
- Preço de \$0.70 por unidade

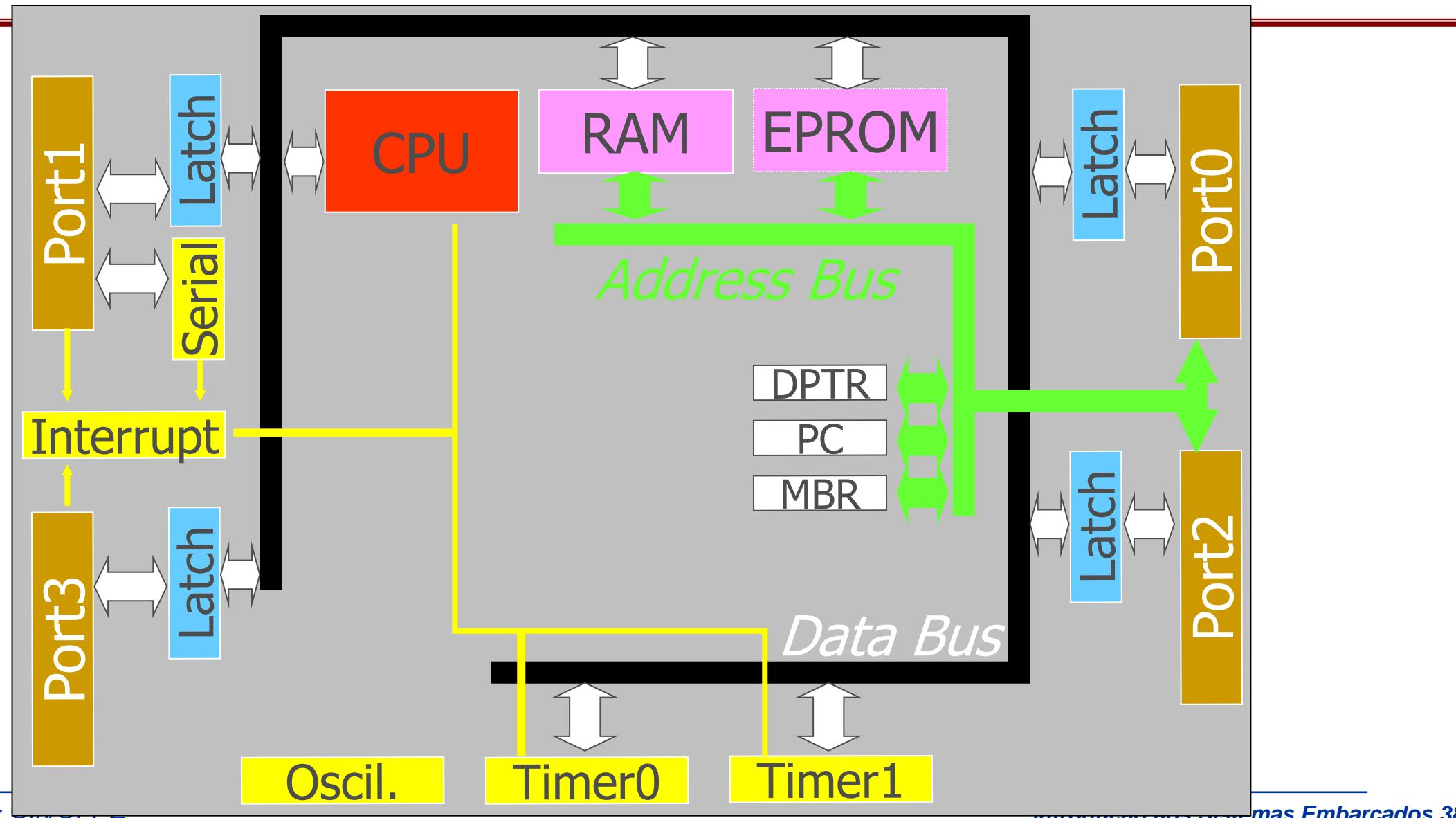
Família Ntron

- Baseado na família HC, com 8 ou 16 pinos
- Memória Flash (1,5 K a 4 Kbytes) reprogramável na aplicação
- Timer de dois canais de 16 bits, com comparação e PWM
- Conversor A/D de 8 bits e quatro canais

Família 8051

- Fabricada por várias empresas como Philips, Atmel, Dallas Semiconductors, Intel
- Preços baixos
- Muitas opções

SE::P & A::Hw::Processadores::ASIPs::Microcontroladores
Família 8051



Outras Famílias

- ARM da Intel, PIC da Microchip, Série HC da Motorola e Transputers da SGS-Thomson.
- Escolha da família:
 - fatores técnicos: velocidade, potência, tamanho, periféricos
 - ambientes de desenvolvimento existentes
 - conhecimento prévio do time de desenvolvimento
 - facilidade de compra, número de fornecedores,etc.

Seleção de Processadores

Processor	Clock speed	Periph.	Bus Width	MIPS	Power	Trans.	Price
General Purpose Processors							
Intel PIII	1GHz	2x16 K L1, 256K L2, MMX	32	~900	97W	~7M	\$900
IBM PowerPC 750X	550 MHz	2x32 K L1, 256K L2	32/64	~1300	5W	~7M	\$900
MIPS R5000	250 MHz	2x32 K 2 way set assoc.	32/64	NA	NA	3.6M	NA
StrongARM SA-110	233 MHz	None	32	268	1W	2.1M	NA
Microcontroller							
Intel 8051	12 MHz	4K ROM, 128 RAM, 32 I/O, Timer, UART	8	~1	~0.2W	~10K	\$7
Motorola 68HC811	3 MHz	4K ROM, 192 RAM, 32 I/O, Timer, WDT, SPI	8	~.5	~0.1W	~10K	\$5
Digital Signal Processors							
TI C5416	160 MHz	128K, SRAM, 3 T1 Ports, DMA, 13 ADC, 9 DAC	16/32	~600	NA	NA	\$34
Lucent DSP32C	80 MHz	16K Inst., 2K Data, Serial Ports, DMA	32	40	NA	NA	\$75

Sources: Intel, Motorola, MIPS, ARM, TI, and IBM Website/Datasheet; Embedded Systems Programming, Nov. 1998

SE::P & A::Hw::Processadores

Proc. de Propósito Único

- Circuito digital projetado para executar um único algoritmo
- Características
 - Contém apenas o necessário ao algoritmo
 - Não tem memória de programa
- Vantagens
 - Projeto sob encomenda pode obter o melhor do tamanho, potência, velocidade, mas perde em flexibilidade
- Ex: co-processadores e periféricos

Application-Specific Integrated Circuit-ASIC

- Preço elevado inicial
- Perda de flexibilidade com relação a mudanças
- Uso depende dos requisitos de performance, consumo, tamanho, preço, etc.

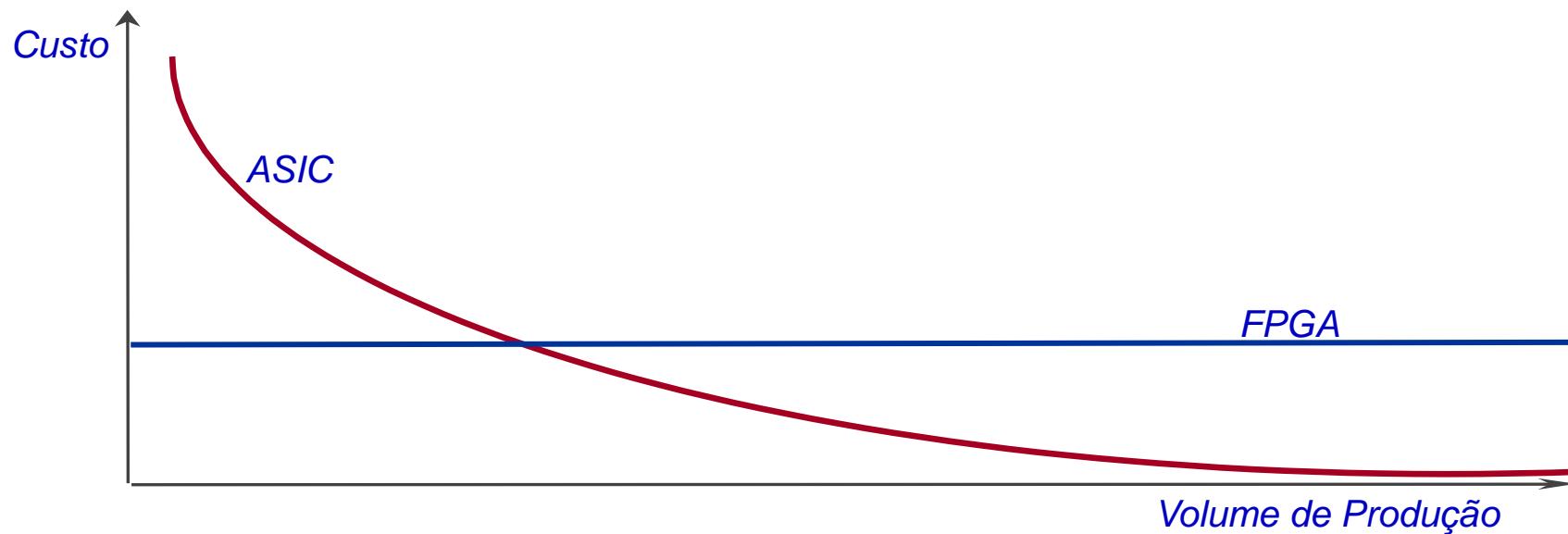
SE::P & A::Hw::Processadores de Propósito Único

Field-Programable Gate Array - FPGA

- Dispositivos de hardware programáveis
- Reconfigurável on-line (tipo RAM)
- Tipos de programação:
 - tipo PROM, programáveis uma única vez
 - tipo EPROM, re-programáveis em laboratório
 - tipo RAM, re-programáveis durante a execução.

FPGA: estrutura geral

- Custo por unidade maior que o custo unitário de um *ASIC*
- Não tem custo relacionado à fabricação da primeira unidade



System-On-A-Chip

- Sistema completo implementado em um único chip
- Uso de núcleos de processadores (cores)
- Baixo custo de fabricação em série
- Alta qualidade
- Diminuição de defeitos de montagem e fabricação em geral
- Baixa potência consumida
- Pequeno tamanho
- Alta velocidade

Memória

- Tipos básicos
 - Memória volátil
 - Perde informações quando não tem energia
 - Usada para armazenar dados
 - Memória não-volátil
 - Mantém informações quando não tem energia
 - Usada para armazenar programas e dados permanentes (configuração)

Nomenclatura Básica

Memória volátil

- RAM = Random Access Memory
- SRAM = Static RAM
- DRAM = Dynamic RAM
- VRAM - Video RAM
- WRAM - Windows RAM

Memória não-volátil

- ROM = Read Only Memory
- PROM ou OTP = Programmable ROM ou One-Time Programmable
- EPROM = Erasable PROM
- EEPROM = Electrically Erasable PROM
(apagamento byte a byte)
- Flash EPROM = Fast erasable EPROM
(apagamento por bloco)

RAM Dinâmica vs. Estática

- DRAM (Dynamic Random Access Memory)
 - Grande capacidade de integração (baixo custo por bit)
 - Perda de informação após algum tempo: Necessidade de refreshing
- SRAM (Static Random Access Memory)
 - Pequeno tempo de acesso
 - Não existe necessidade de refreshing
 - Alto custo por bit (alta integração)

Características Básicas

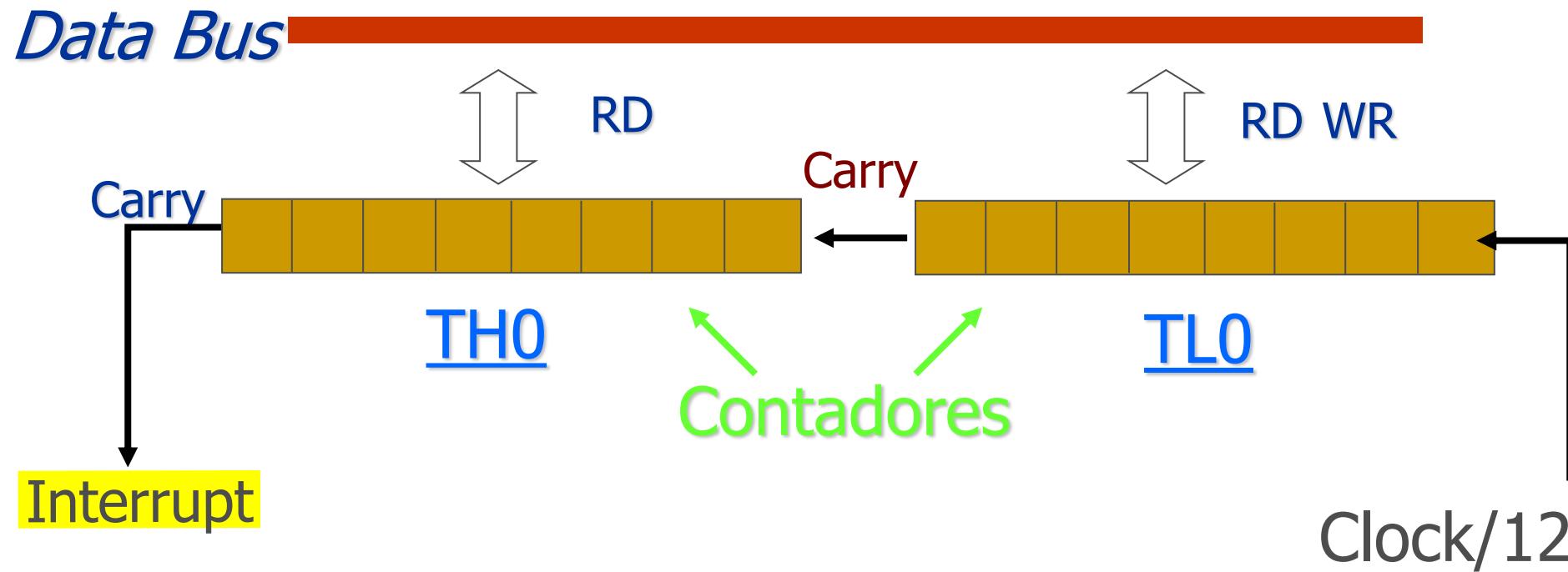
Tipo de Memória	Categoria	Apagamento	Escrita	Volatile
Random-Access Mem. (RAM)	Read-Write	Elétrico byte a byte	Elétrica	Volátil
Read-Only Mem. (ROM)	Read-only	Impossível	Máscara	
Programmable ROM (PROM)				
Erasable PROM (EPROM)		Ultra-violeta	Elétrica	não-volátil
Electrically EPROM (EEPROM)	Read-mostly	Elétrico byte a byte		
Flash EEPROM		Elétrico por bloco		

Periféricos

- Teclados e Visores de cristal líquido
- Temporizadores e contadores
- Interfaces Seriais
- Transdutores
- Conversores Digital-Analógicos e Analógico-Digitais
- Modulação de largura de pulso - PWM
- Motores de passo

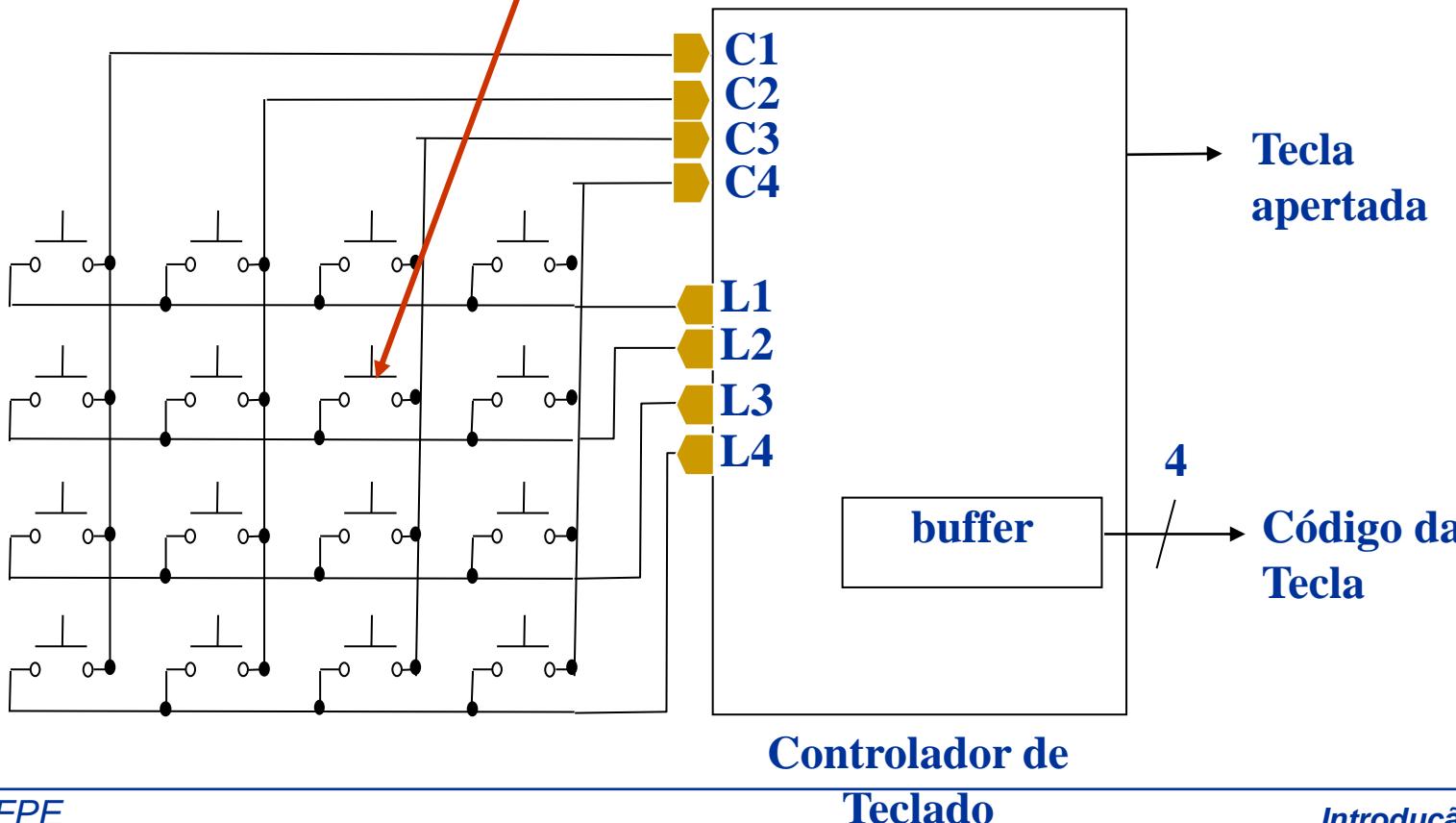
Timer(temporizador)/Contador

- Dispositivo que gera interrupções em intervalos regulares de tempo.
Ex: Timer do 8051



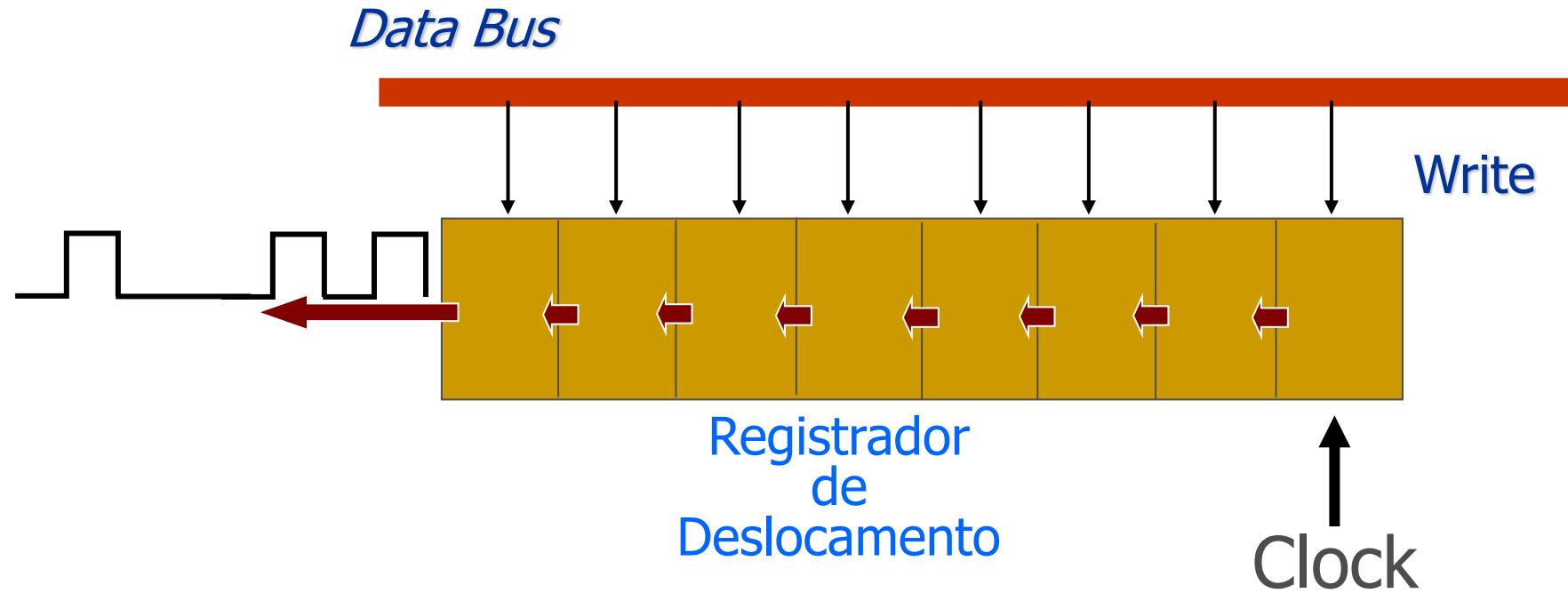
Teclado

- Para saber se esta tecla está apertada, o controlador do teclado põe nível 0 em L2, mantém nível 1 em L1, L3 e L4 e testa o valor de C3



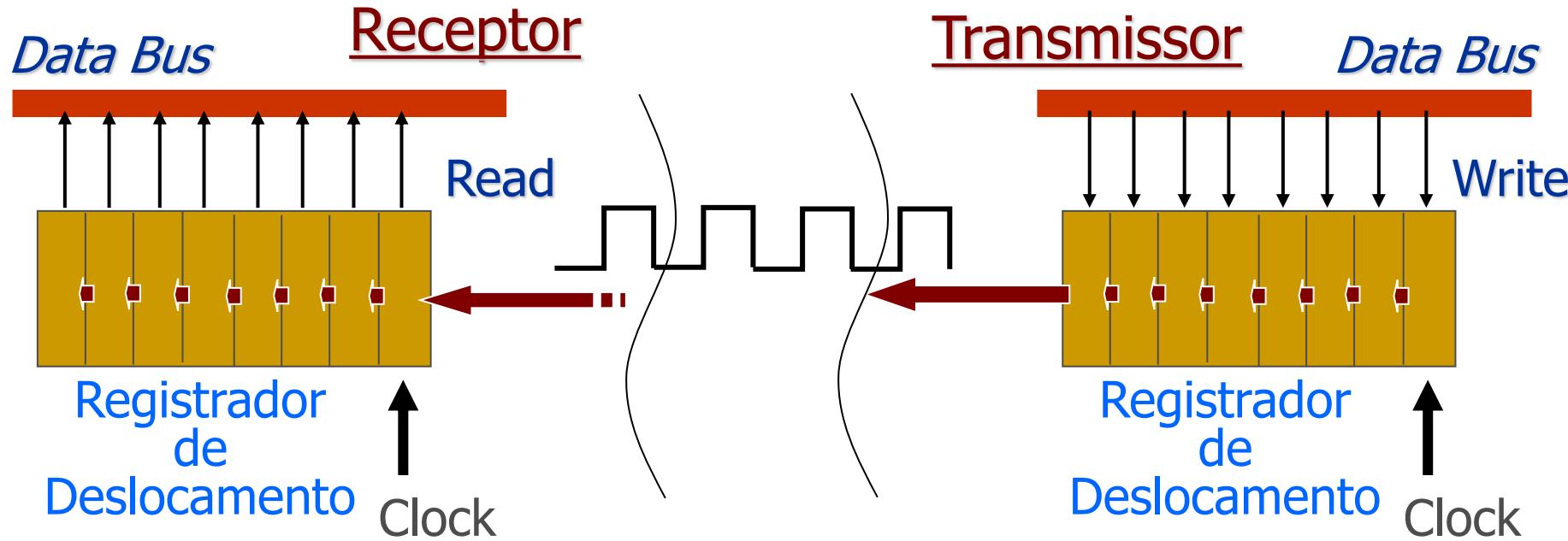
Interfaces Seriais

- Comunicação em que cada bit é enviado em sequência por um único fio



Interfaces Seriais

- Exemplo de Comunicação



RS232

- RS232: Recommended Standard 232 da EIA
 - Comunicações ponto a ponto
 - Usado por modems, mouses e algumas impressoras
 - Baixa imunidade a ruído
 - Mencionado no padrão:
 - limite de transmissão 20kbps
 - distância máxima entre dispositivos: 15 metros
 - Na prática:
 - pode-se transmitir até cerca de 200kbps
 - atinge 100m.

RS422 e RS485

- Adequadas para condições adversas de operação
- comunicação por pares de fio trançados
- velocidades superiores a 100Mbps
- distâncias de vários quilômetros
- conexão de vários dispositivos na mesma linha

Universal Serial Bus - USB

- Taxas de até 12Mbps
- Até 127 periféricos por linha
- Conexão com sistema ligado (*hot plugging*)
- Reconhecimento automático do dispositivo (*Plug-and-Play*)
- Microcontroladores dotados de interfaces USB

Infravermelho

- IrDA (Infrared Data Association - 1993)
 - Vários padrões de comunicação
 - IrDA-Data, IrDA-Control, e o Air
 - IrDA-Data
 - ponto-a-ponto
 - cone de incidência estreito (30°)
 - distância máxima de 1 metro
 - velocidades de 9600 bps a 16 Mbps.
 - base instalada com alta taxa de crescimento.

Bluetooth

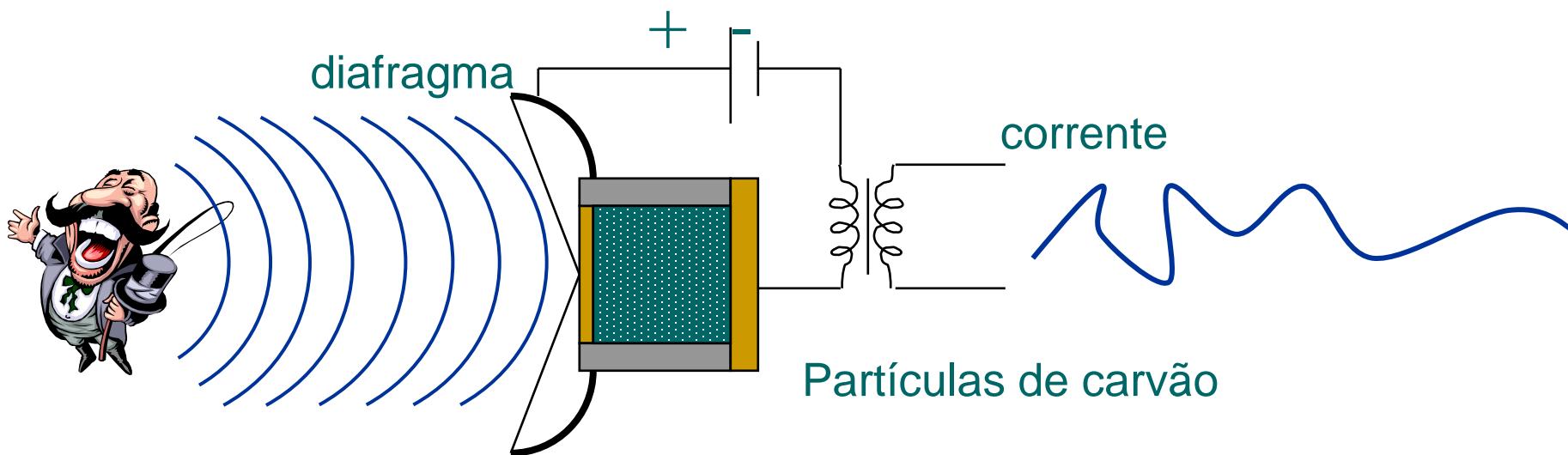
- ondas de rádio
- comunicação de voz e dados
- pequenas distâncias (10cm a 10m)
- ponto-a-multiponto (*broadcast*)
- faixa de frequência usada é de 2.4 GHz
- suporta rede de até 8 dispositivos
- fácil integração com protocolo TCP/IP.
- Várias companhias adotando como Motorola, Ericsson, Nokia, IBM, Toshiba, Intel, entre outras.

Visor de Cristal Líquido - LCD

- vários modelos
 - alfanuméricos, que só apresentam letras, números e caracteres de pontuação
 - gráficos que permitem o acesso a cada ponto independentemente
- vários tamanhos
- com ou sem iluminação (*backlight*)
- interface, em geral, paralela.
- Ponto importante da interface com o usuário

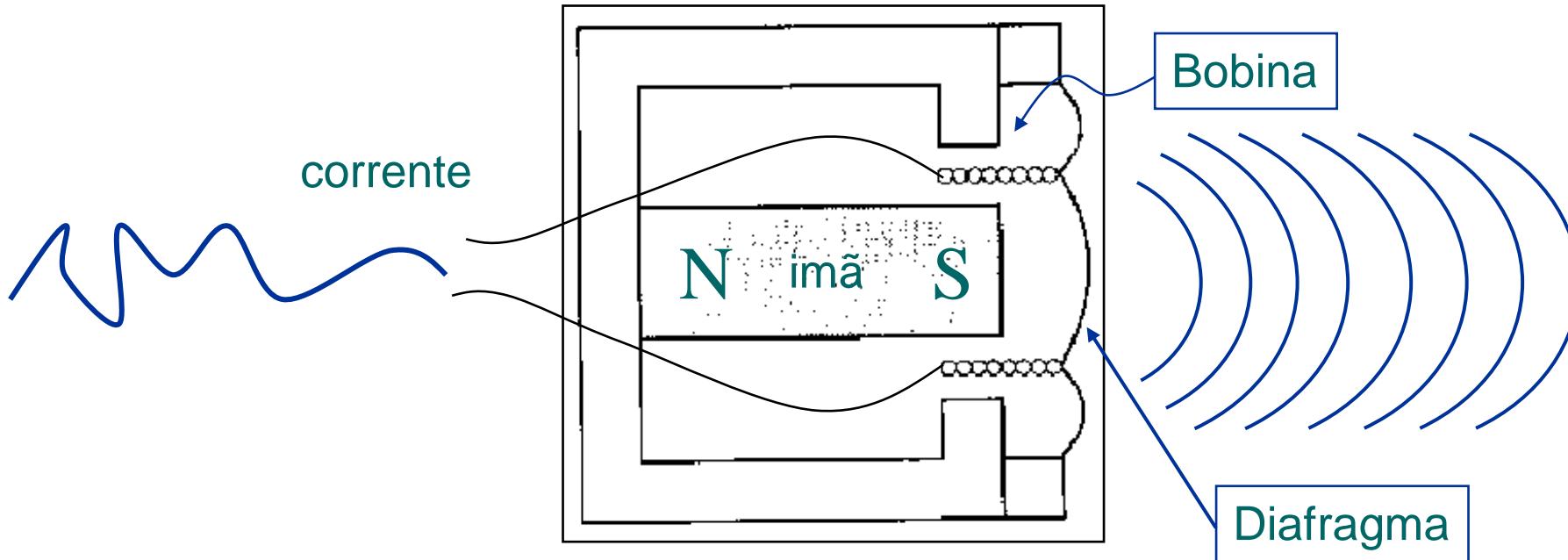
Transdutores

- Capazes de converter tipos de energia
- Exemplo: microfones de carvão
 - A pressão do ar (onda sonora) desloca o diafragma
 - que muda a densidade de partículas de carvão
 - variando a resistência elétrica
 - que produz uma onda elétrica similar à sonora



Transdutores

- Alto-falante



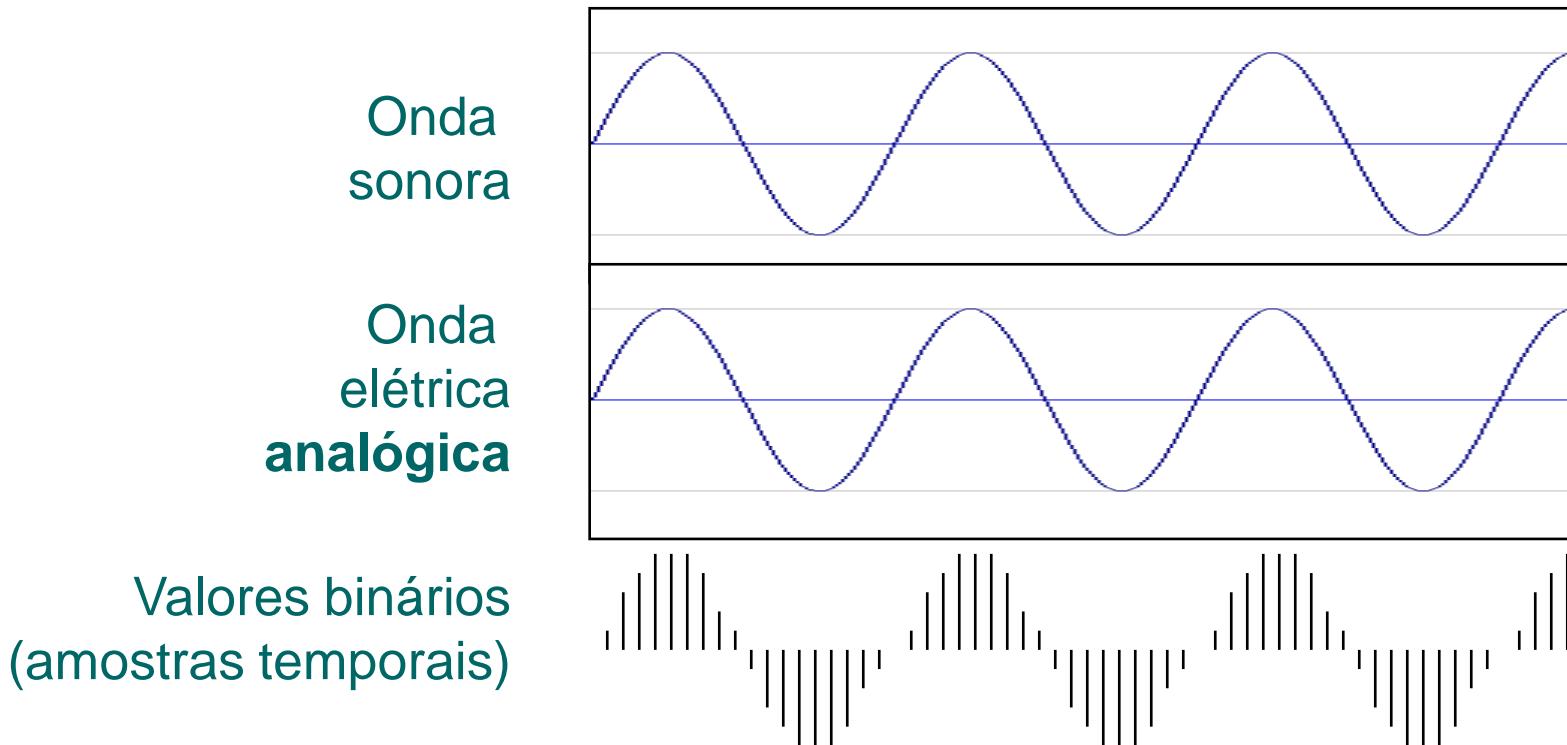
- A corrente elétrica variável é aplicada à bobina
 - que induz um campo magnético
 - que movimenta o diafragma
 - que desloca o ar formando uma onda sonora

Transdutores

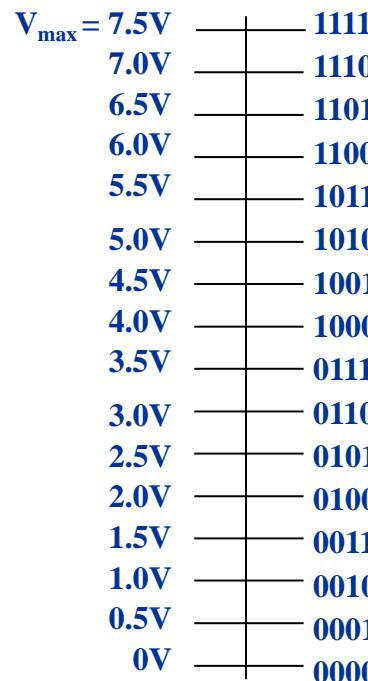
- Materiais especiais
 - piezzo-elétricos, são capazes de se expandir ou contrair de acordo com a voltagem (e vice-versa)
 - sensores de pressão, peso, e em microfones e tweeters
 - termopares, geram voltagens proporcionais à sua temperatura
 - células fotoelétricas, geram voltagens ou modificam a resistência proporcionalmente à iluminação sobre elas

Conversores Analógico/Digitais

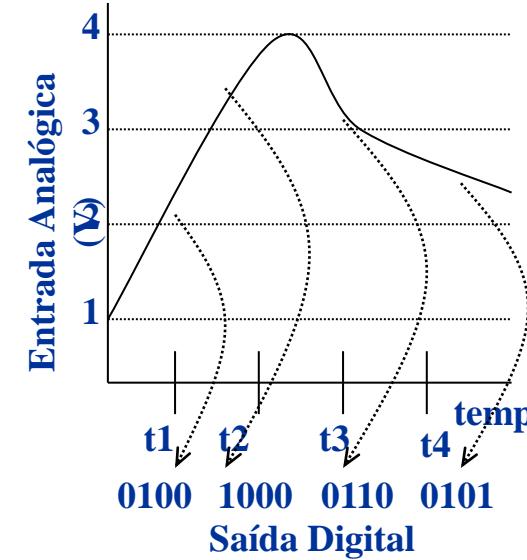
- Conversão de valores analógicos (contínuos no tempo e na amplitude) em valores digitais (números binários)



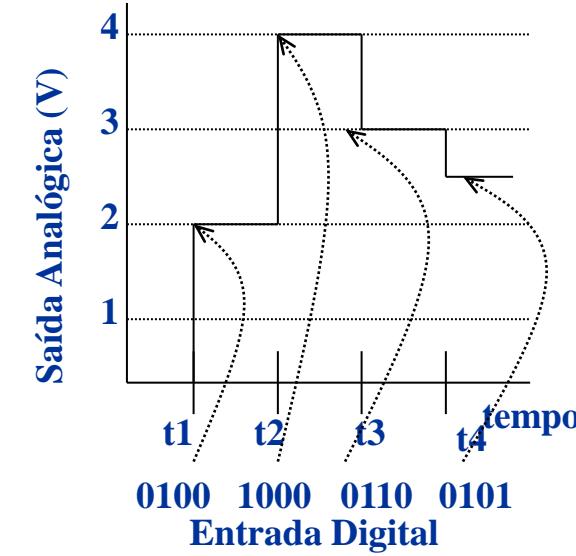
Conversores Analógico/Digitais



Relação de Valores



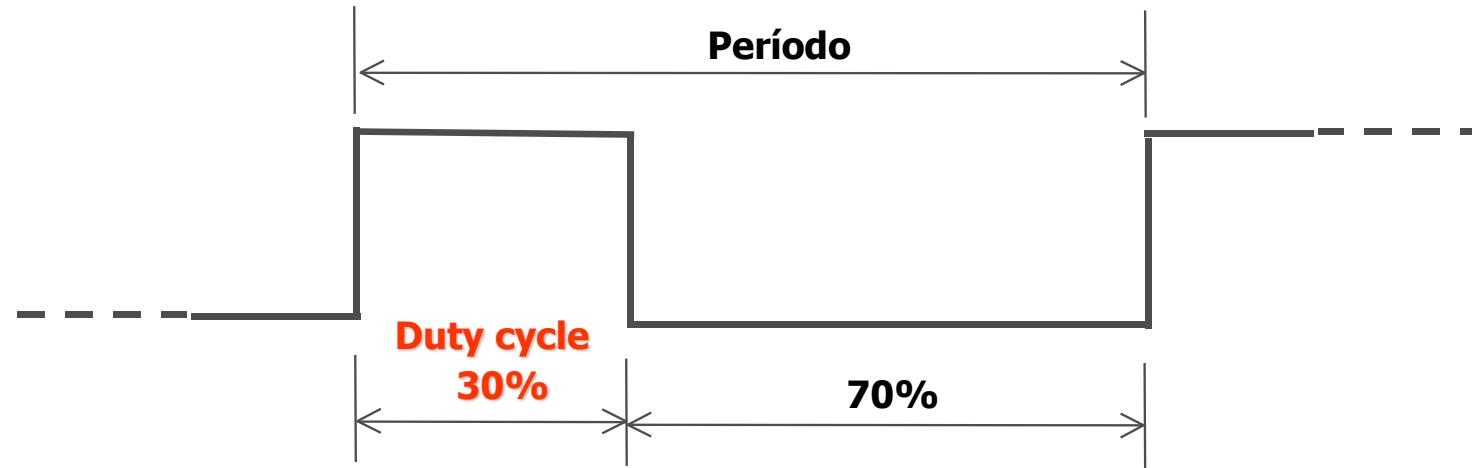
Conversão
Analógico \Rightarrow Digital



Conversão
Digital \Rightarrow Analógico

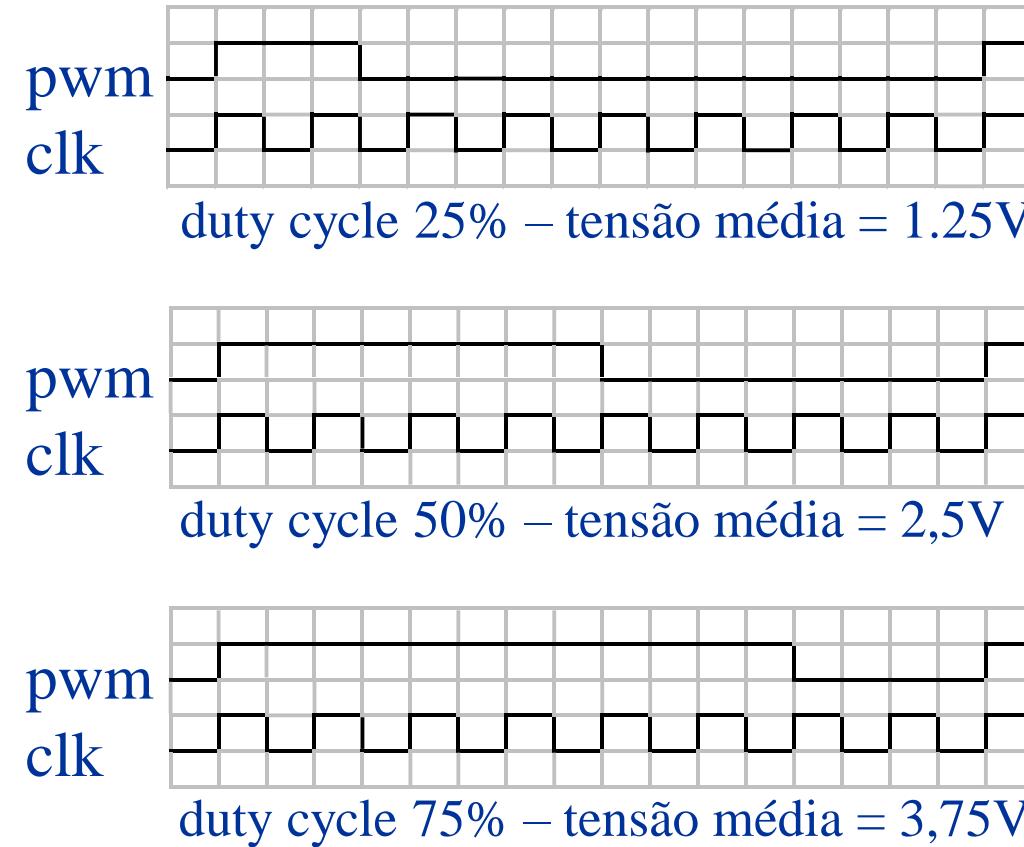
Modulação de largura de pulso (PWM)

- Onda quadrada
 - definição do período
 - definição do *duty cycle* (percentual do tempo em que o sinal fica em 1)



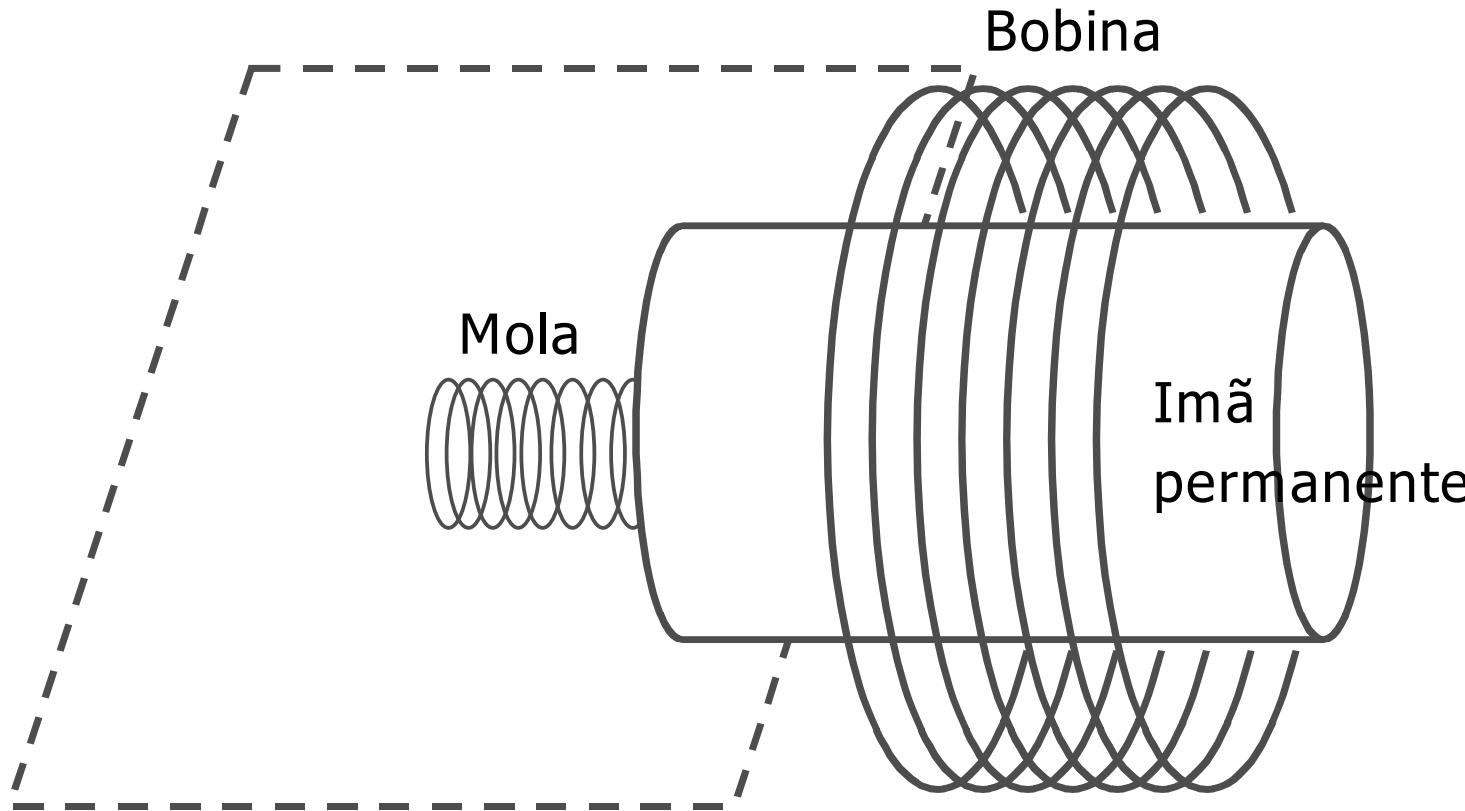
Modulação de largura de pulso (PWM)

- Gera pulsos com "duty cycle" definido
- Controla a tensão média de um dispositivo externo (ex: motor, lâmpada)
 - Mais simples que conversor DC-DC ou A/D



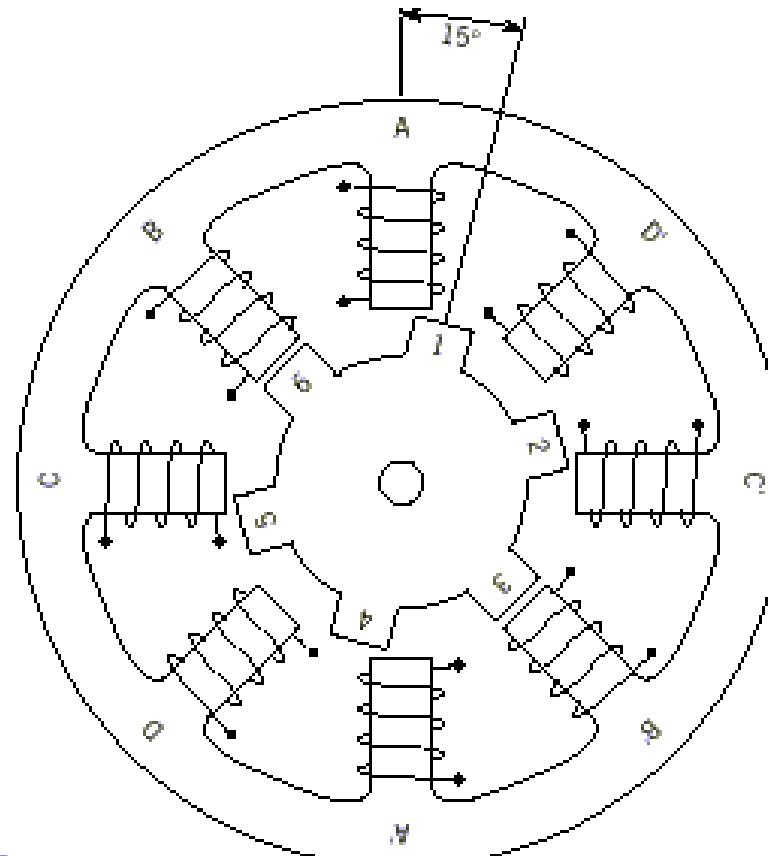
Modulação de largura de pulso (PWM)

- Controle de motores de corrente contínua e de solenóides



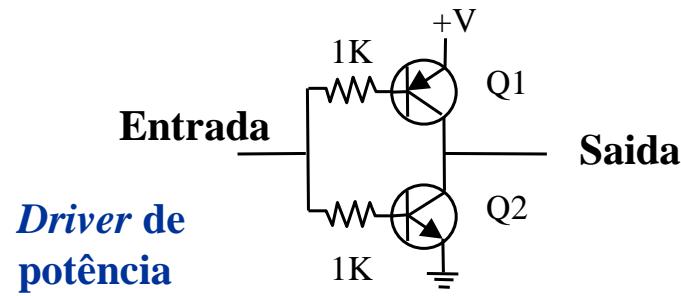
Motores de Passo

- Tipos de motores que giram “passo a passo”, girando alguns graus por vez

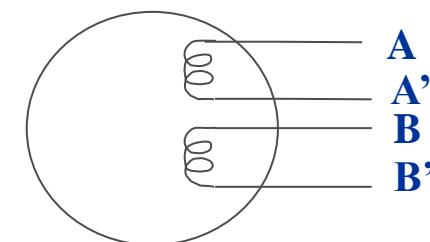


Motores de Passo

- Para rodar o motor é necessário seguir uma seqüência de acionamento das bobinas
- Existem controladores para facilitar esta tarefa
- Requer “driver” de potência



Sequencia	A	B	A'	B'
1	+	+	-	-
2	-	+	+	-
3	-	-	+	+
4	+	-	-	+
5	+	+	-	-



Intro. a Sistemas Embarcados

- Visão geral
- Projeto e Arquitetura
 - Hardware
 - Software
 - Linguagens
 - Sistemas operacionais de tempo real

Linguagens Empregadas

- Segundo uma pesquisa recente da www.8052.com
 - 49% assembly
 - 33% C
 - 5% Basic
 - 3% Pascal
 - 9% outras linguagens
- A previsibilidade no uso de recursos e tempo de execução é fundamental para sistemas críticos

Linguagens orientadas a objetos

- Linguagens orientadas a objeto
 - ciclo de desenvolvimento mais rápido
 - melhores métodos de estruturação e modularização
 - reutilização de objetos
 - criação e destruição dinâmica de objetos, garbage collectors criam problemas para previsibilidade temporal e de alocação de memória
 - Java, não há um modelo definido para o escalonamento das threads. Inviabiliza tempo real

Linguagens imperativas

- C e Assembly
 - Excelente controle do hardware e do tempo de execução
 - Alta performance
 - Dificuldades no desenvolvimento
 - Assembly:
 - Difícil modularização e reutilização

Linguagens especiais

- Ada
 - Projetada pelo Department of Defense dos EUA
- Esterel
 - Voltada para sistemas reativos
 - Baseada em eventos e ações
 - Semelhante às linguagens visuais (ex. Visual Basic)

Sistemas de Tempo Real

- Sistemas computacionais de tempo real:
 - Tem aspectos temporais em sua especificação
 - Submetidos a requisitos de natureza temporal
 - Resultados devem estar corretos lógica e temporalmente
“Fazer o trabalho usando o tempo disponível”
 - Requisitos definidos pelo ambiente físico
- Aspectos temporais
 - Não estão limitados a uma questão de desempenho
 - Estão diretamente associados com a funcionalidade



STR e o Ambiente

- Fluxo de controle definido pelo ambiente
 - Tarefas ativadas por estímulos do ambiente
 - STR geralmente não tem como limitar estes estímulos
- Tempos de resposta ao ambiente definem deadlines
- Dados com prazos de validade
 - Dados desatualizados podem conduzir a respostas incorretas

Modelagem das Tarefas

- Deadline – Tempo máximo para término de uma tarefa
- Release-time – Tempo mínimo para início de uma tarefa
- Tempo de execução: considera o pior caso
 - WCET – Worst-Case Execution Time

Criticidade

Sistema de tempo real crítico (Hard real-time system)

- Todas as tarefas têm Hard Deadline
 - Perda do deadline pode ter consequências catastróficas
- É necessário garantir requisitos temporais ainda durante o projeto
- Exemplo: usina nuclear, indústria petroquímica, mísseis

Sistema de tempo real não crítico (Soft real-time system)

- O requisito temporal descreve apenas o comportamento desejado
- Perda do deadline não tem consequências catastróficas
- Existe interesse em terminar a tarefa mesmo com atraso
- Exemplo: início de gravação de vídeo-cassete

Criticidade

Deadline Firm

- Perda do deadline não tem consequências catastróficas
- Não existe valor em terminar a tarefa após o deadline
- Exemplo: ler o valor da temperatura

Modelagem das Tarefas

- Periodicidade
 - Tarefas Aperiódicas
 - São disparadas em intervalos imprevisíveis de tempo
 - Não garantem escalonabilidade
 - Tarefas Esporádicas
 - É conhecido o intervalo mínimo entre execuções (*inter-arrival time*)
 - Tarefas Periódicas
 - Devem ser executadas em intervalos regulares de tempo

Modelos de Implementação

- Time-driven systems
 - Sistemas em que a execução é regida pelo tempo
 - Não existe outra fonte de eventos além do timer interno
 - A ordem de execução é toda definida a priori
- Event-driven systems
 - A execução é regida por eventos externos e eventos do timer
 - A garantia em projeto se dá pelo uso de prioridades fixas para as tarefas

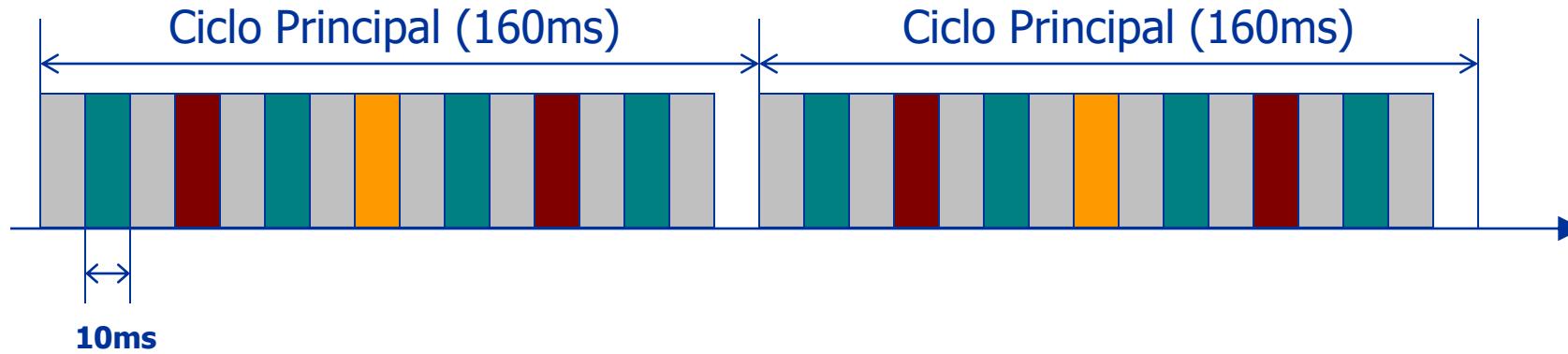
Time-Driven Systems

Executor Cíclico

- Tarefas são arranjadas numa lista que define a ordem e tempo de execução de cada uma
- Cada tarefa é colocada em execução em momentos controlados por um temporizador (timer)
- A lista é executada repetidamente, caracterizando ciclos de execução
- Não é usada concorrência: tarefas grandes devem ser quebradas manualmente se necessário.

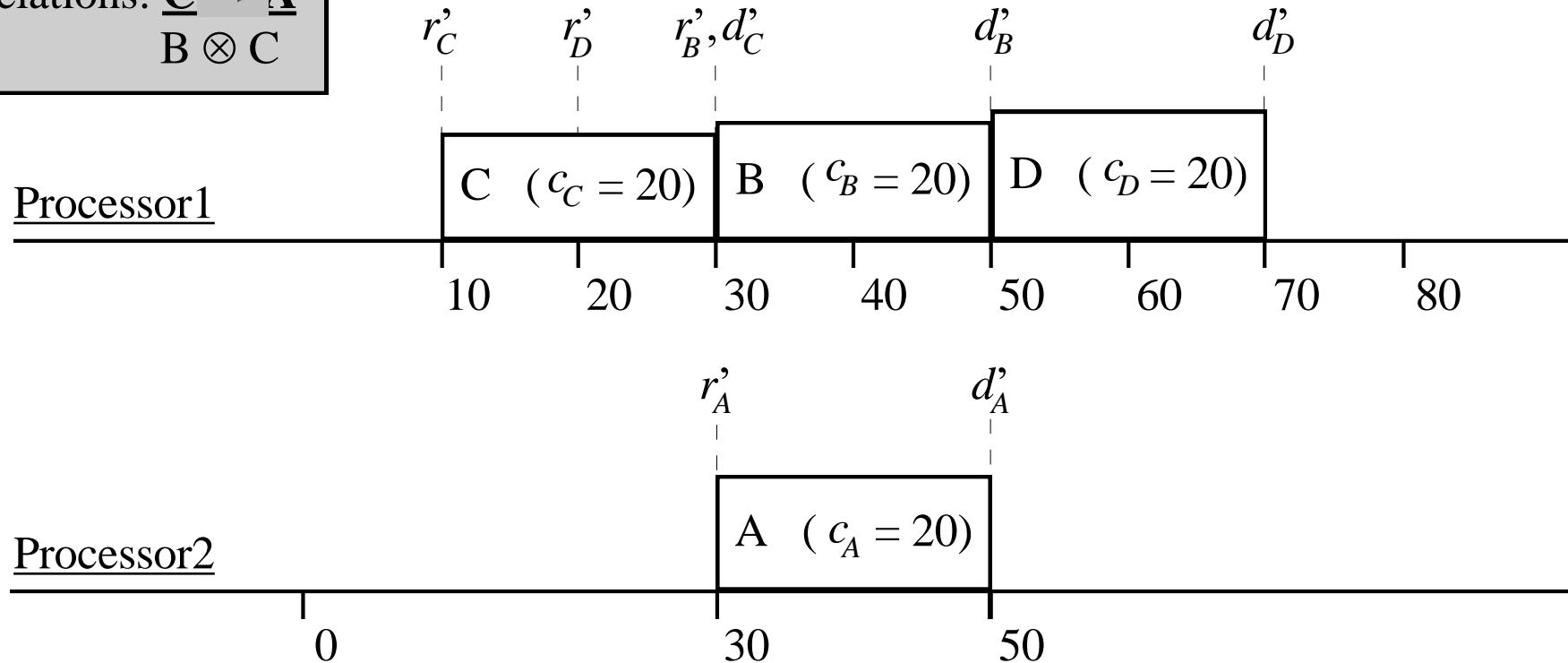
Time-Driven Systems: Exemplo 1

- Considere 4 funções com as seguintes características:
 - Função 1: ciclo de 50Hz (20ms)
 - Função 2: ciclo de 25 Hz (40ms)
 - Função 3: ciclo de 12,5 Hz (80ms)
 - Função 4: ciclo de 6,25Hz (160ms)



Time-Driven Systems: Exemplo 2

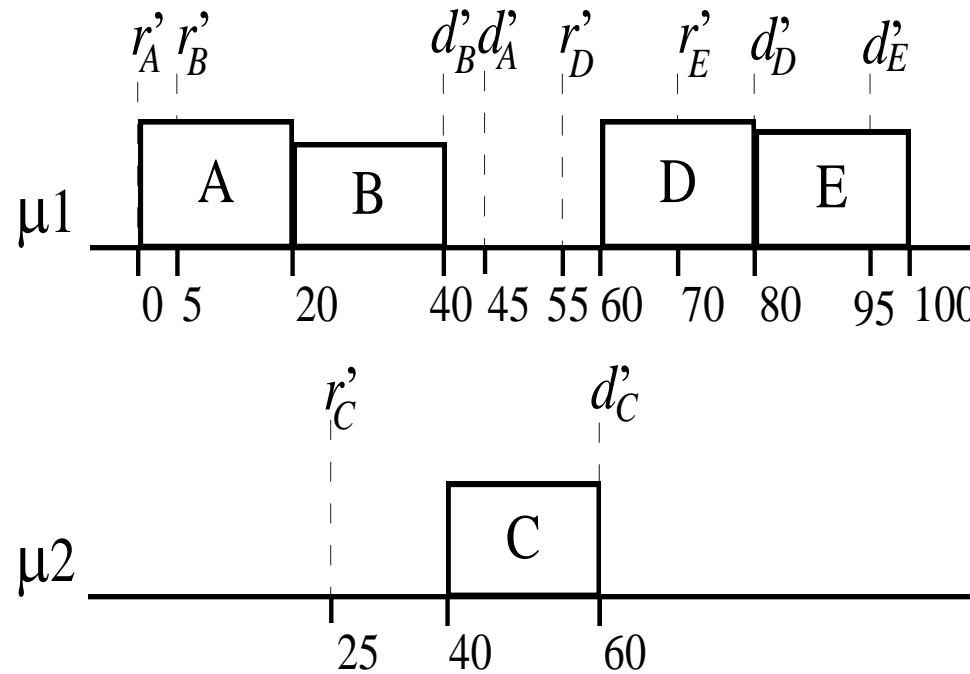
Relations: $\underline{C \rightarrow A}$
 $B \otimes C$



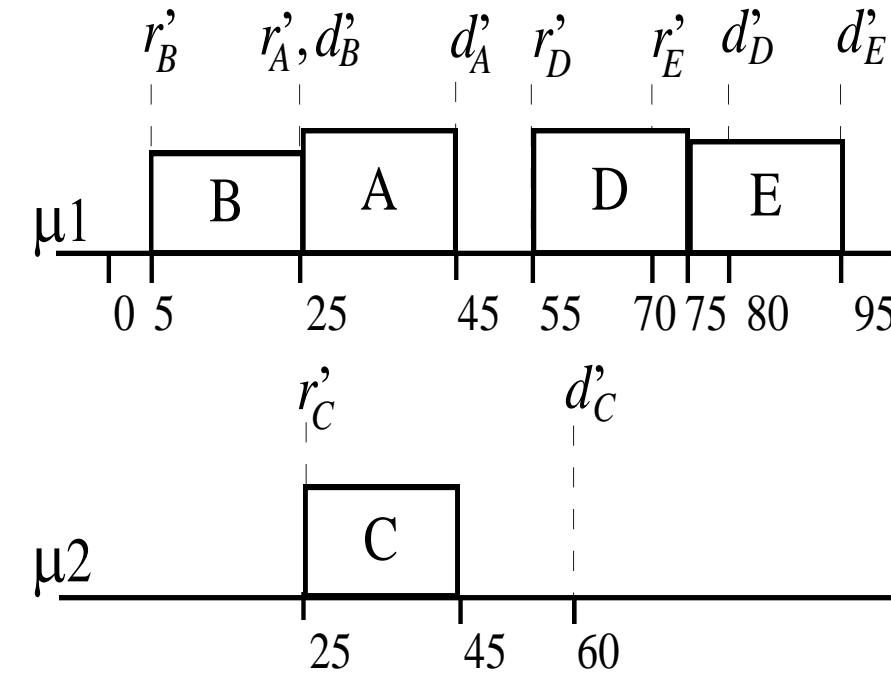
SE::P & A::Sw::STR::Escalonamento::Garantia em Projeto

Time-Driven Systems: Exemplo 3

Relations: $A \otimes B$,
 $B \rightarrow C$, $C \rightarrow D$

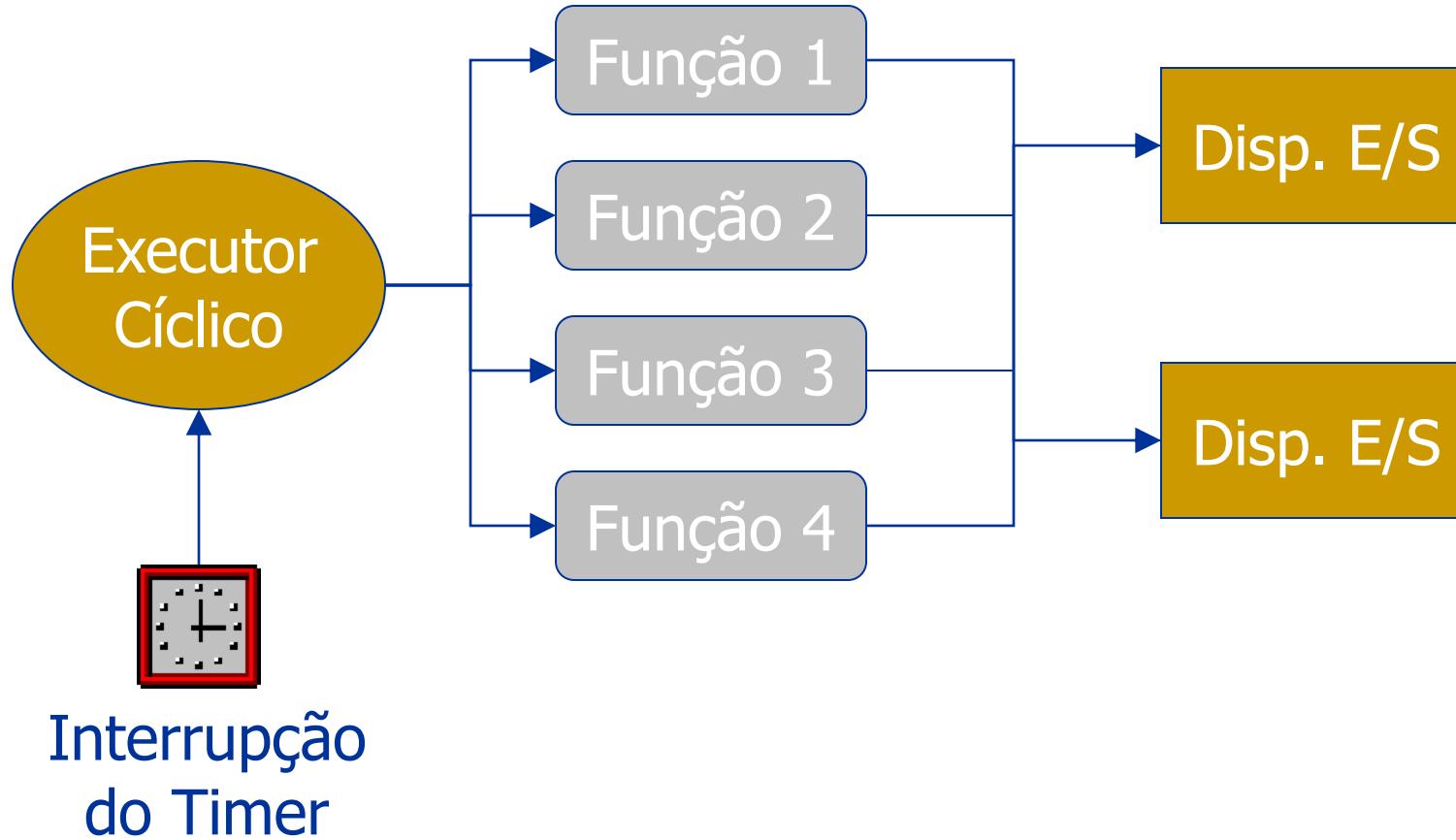


Relations: $\underline{B \rightarrow A}$,
 $B \rightarrow C$, $C \rightarrow D$



Time-Driven Systems: Executor Cíclico

- Arquitetura de Software



Event-Driven Systems

Sistemas de Prioridade Fixa

- Tarefas têm prioridades definidas em tempo de projeto
- Tarefas de maior prioridade preemptam as de menor prioridade
- Prioridades são definidas com base nas restrições temporais
- A concorrência requer mecanismos para evitar deadlock e inversão de prioridade (exemplo: semáforos, herança de prioridade)
- Para garantir respostas previsíveis, tarefas aperiódicas não são tratadas

Event-Driven Systems

Sistemas de Prioridade Fixa

- *Escalonamento Rate-Monotonic (RMS)*
 - Escalonamento preemptivo de prioridade fixa
 - Tarefas mais freqüentes têm maior prioridade
 - Os deadlines devem ser iguais aos períodos
 - Tarefas não têm nenhuma relação entre si (precedência, exclusão, etc)
 - É ótimo, ou seja, nenhum outro método é melhor que este com estas condições

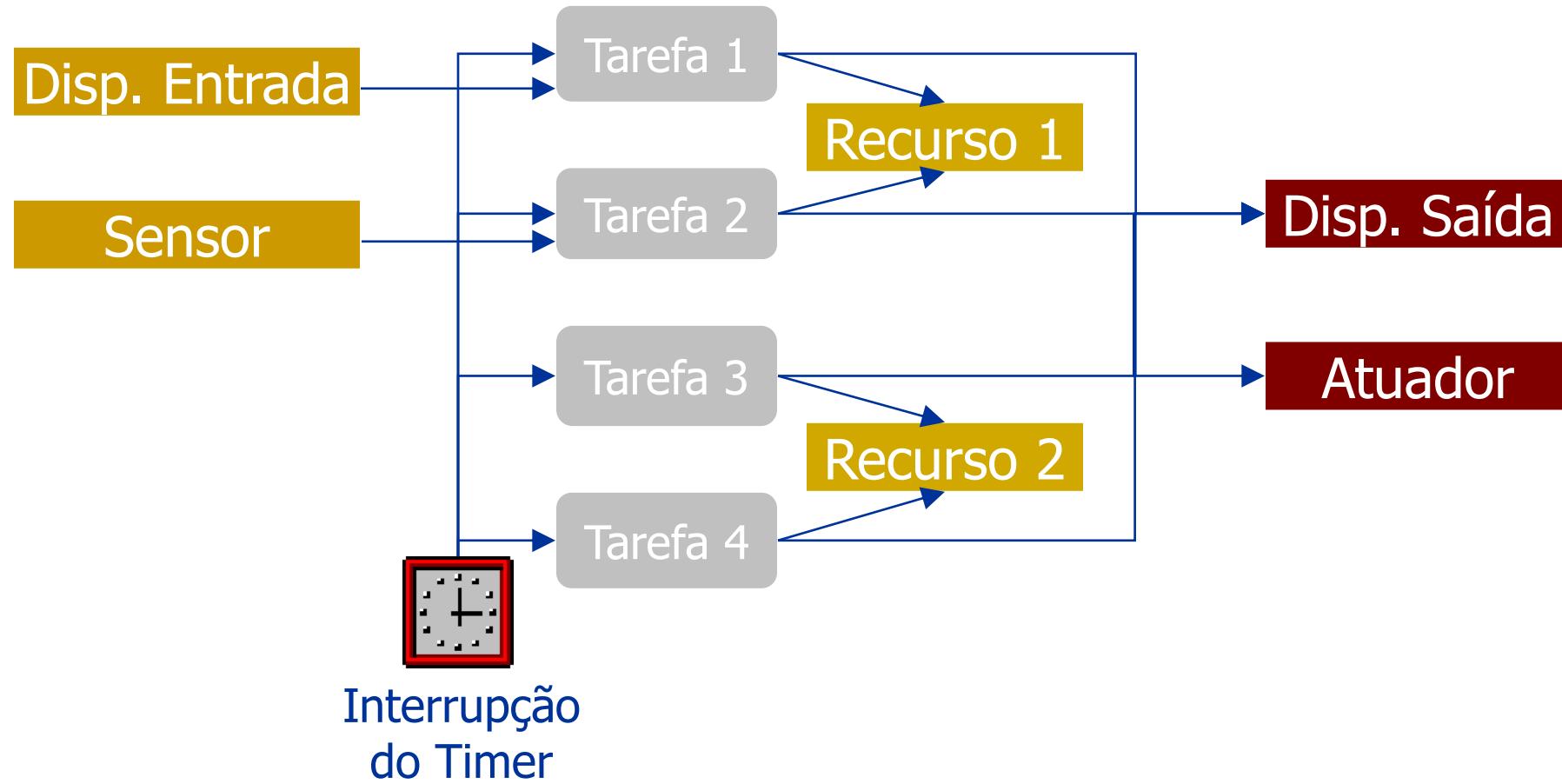
Event-Driven Systems

Sistemas de Prioridade Fixa

- *Deadline-Monotonic Scheduling:*
 - Semelhante ao RMS, mas quanto menor o deadline da tarefa maior sua prioridade
 - Os deadlines são fixos e relativos aos começos dos períodos

Event-Driven Systems

Arquitetura de Software



Event-Driven Systems

Análise de Escalonabilidade

Escalonamento Rate-Monotonic (RMS)

- Notação:
 - Processo: p_i Período: T_i Tempo de Computação: C_i Deadline: D_i
- Em um conjunto de n processos periódicos independentes escalonados pelo RMS sempre obedecerá o seu deadline se (condição suficiente mas não necessária):

$$\frac{C_1}{T_1} + \dots + \frac{C_n}{T_n} \leq U(n) = n \left(2^{\frac{1}{n}} - 1 \right)$$

onde, $U(n)$ é o limite de utilização para n processos.

$$\underline{n \rightarrow \infty \Rightarrow U(n) \rightarrow 69\%}$$

Sistemas Operacionais de Tempo Real

- Escalonamento preemptivo
- Escalona processos com base em prioridades (não é justo, pode provocar *starvation*)
- Todas as chamadas ao S.O. tem tempo máximo de execução definido e otimizado
- A mudança de contexto tem tempo limitado, conhecido (*fixed overhead*) e otimizado
- Tratamento de inversão de prioridade

Referências

- **Livro de Sistemas de Tempo Real**
Jean- Marie Farines, Joni da Silva Fraga, Rômulo Silva de Oliveira. Escola de Computação'2000 - IME- USP
<http://www.lcni.ufsc.br/gtr/livro/principal.htm>
- **IEEE Computer Society, Technical Committee on Real- Time Systems (IEEE- CS TC- RTS)**
<http://www.cs.bu.edu/pub/ieee-rts>
- The Concise Handbook Of Real-Time Systems.
TimeSys Corporation, Versão 1.1, 2000.
<http://www.timesys.com>