



**Universidade Federal de Uberlândia
Engenharia Eletrônica e de Telecomunicações**

– Microprocessadores –

Cap. 2 – Microprocessador, operação de hardware e computação

Prof. Alan Petrónio Pinheiro

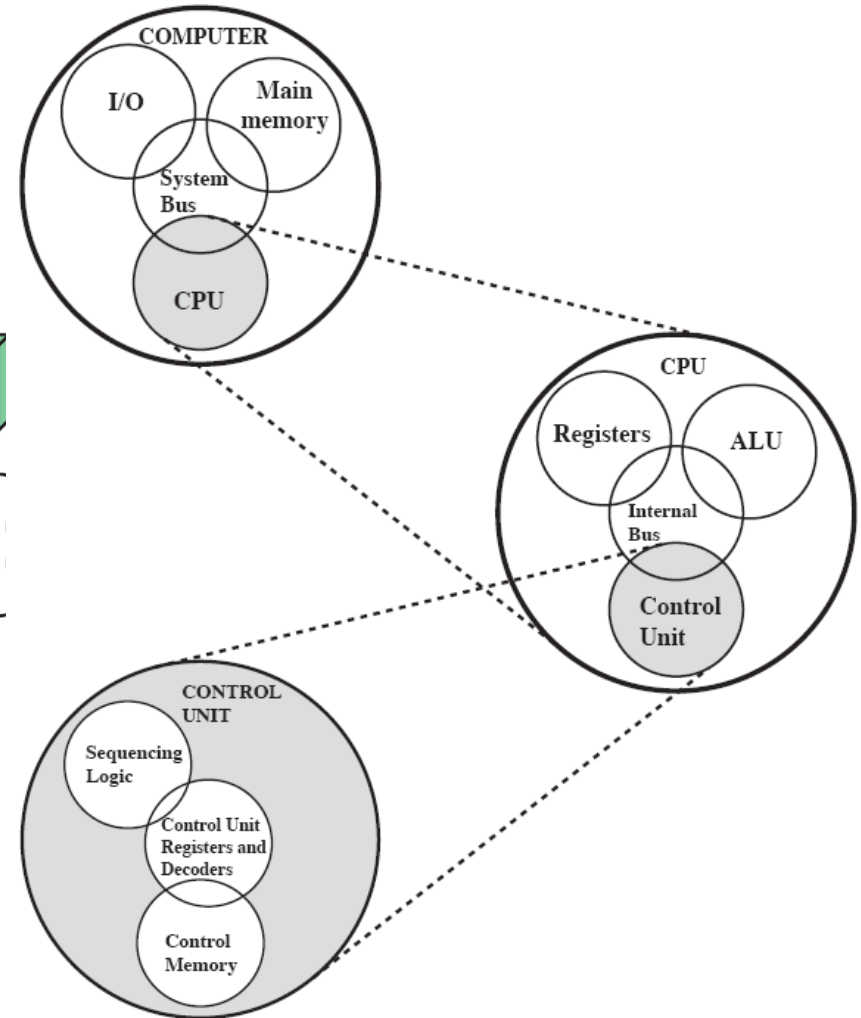
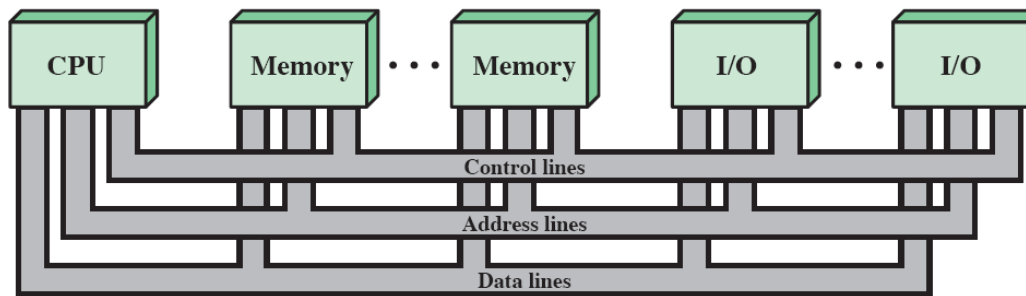
Sumário

- 2.1 Arquiteturas microprocessadas
- 2.2. Diagrama de blocos (arquitetura, registradores, ALU, etc.)
- 2.3. Barramentos
- 2.4. Busca e execução de instruções
- 2.5. Estudo de um sistema mínimo realizável: o SAP
- 2.6. Tópicos adicionais: DMA, cache, multiprocessamento ...
- 2.7 . Arquiteturas microprocessadas modernas de 32 bits
- 2.8 Apresentação de microcontroladores comerciais

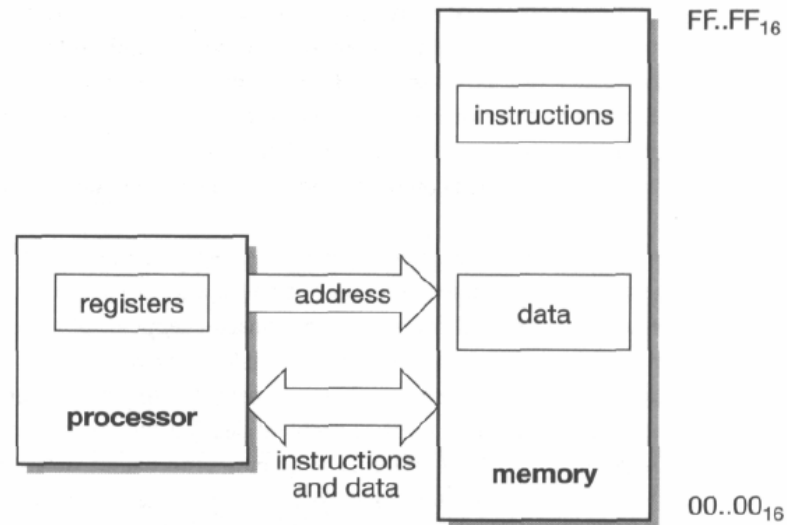


2.1 – Arquiteturas microprocessadas

- Microcontrolador x microprocessador

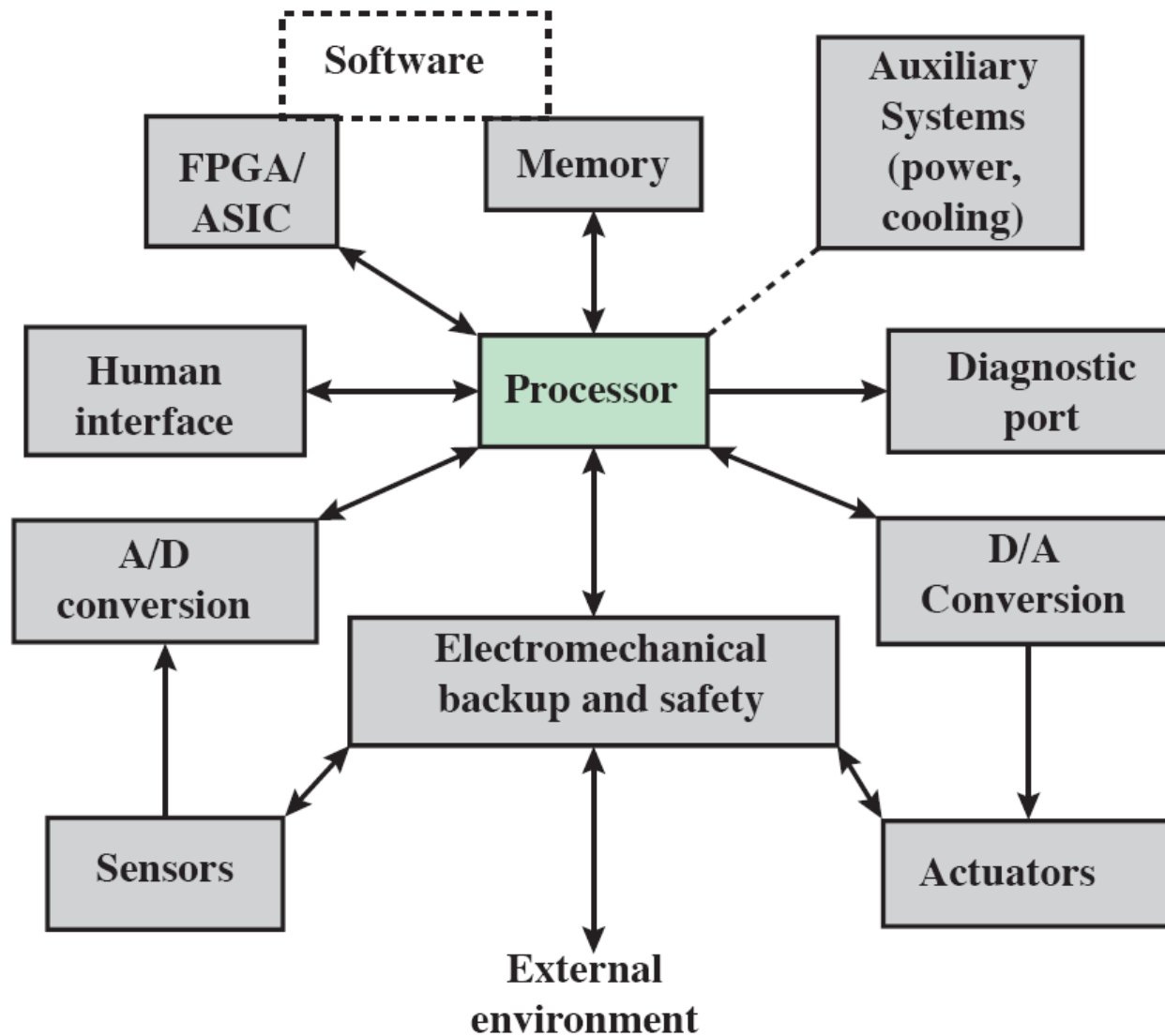


- (micro)processador: CI que executa instruções em memória

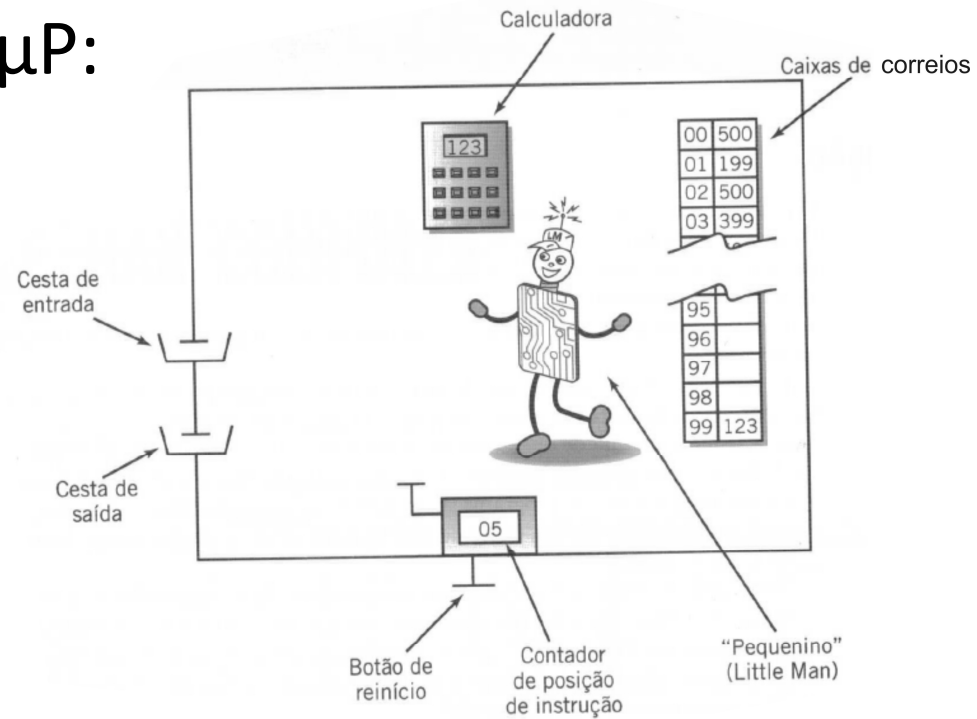


- Princípio do “programa armazenado em memória”
 - Instrução e dados na mesma memória
 - Flexibilidade e universalidade
- Avanços
 - Eletrônica: transistores efeito-campo e tecnologia VLSI
 - Arquitetura/organização microprocessadores
 - Pipeline; cache; busca antecipada, RISC/CISC, etc ...

- Microcontrolador:



- A analogia do μC / μP :



Op code



3 | 25

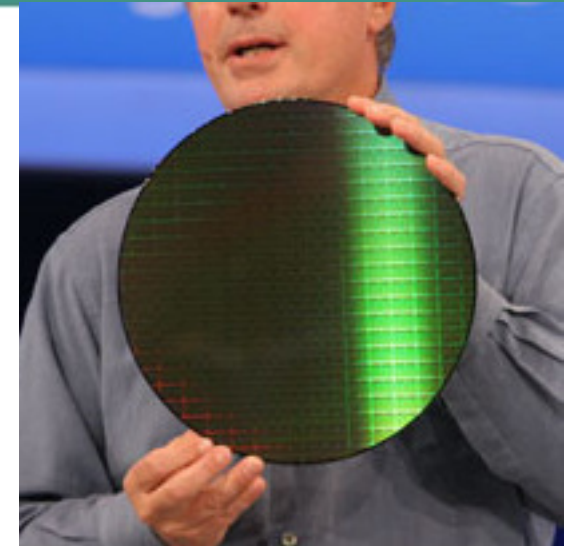


endereço

– Instruções:

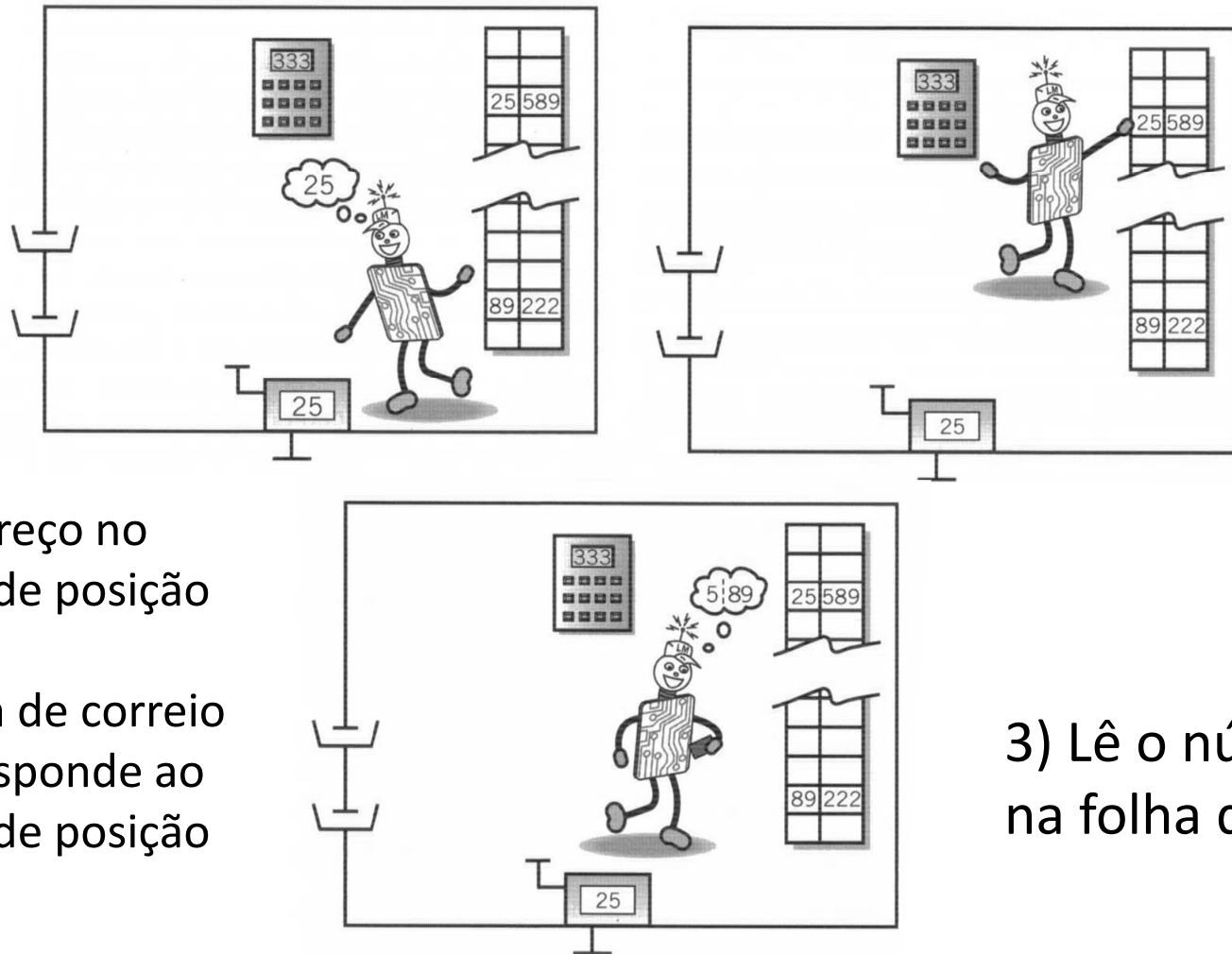
- LOAD (op code 5)
- STORE (op code 3)
- ADD (op code 1)
- SUBTRACT (op code 2)

- INPUT (op code 9, end 01)
 - OUTPUT (op code 9, end 01)
 - HALT (op code 0)
 - SUBTRACT (op code 2)
- Tipos de instruções:
- (i) transferência e dados;
 - (ii) aritmética simples;
 - (iii) entrada e saída;
 - (iv) controle de máquina.
- Exemplo de um programa:
- 901
 - 399
 - 901
 - 199
 - 902
 - 000

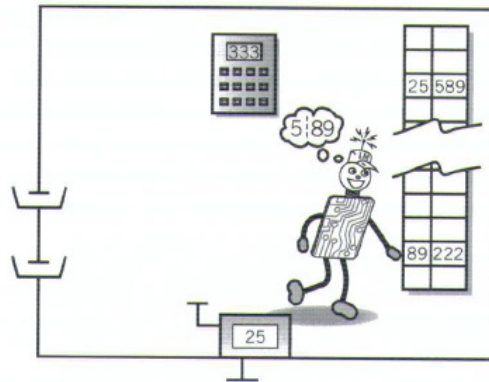


– Ciclo de instrução

- Busca (fetch):

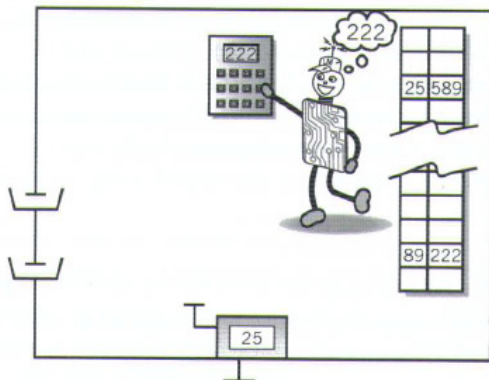


– Ciclo de execução:

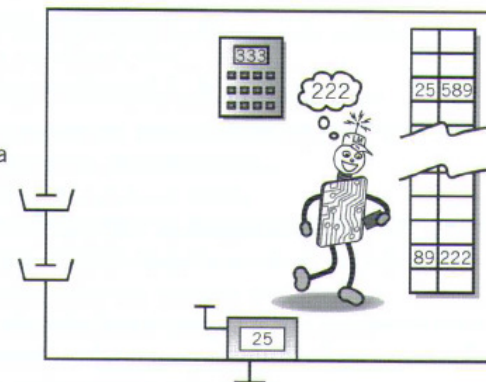


(1) O Pequenino vai para o endereço da caixa de correio especificado na instrução que buscou anteriormente

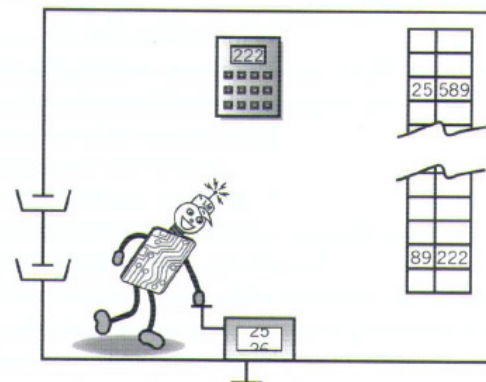
(2) . . . ele lê o número naquela caixa de correio (ele se lembra de devolvê-lo, para o caso de precisar dele novamente)



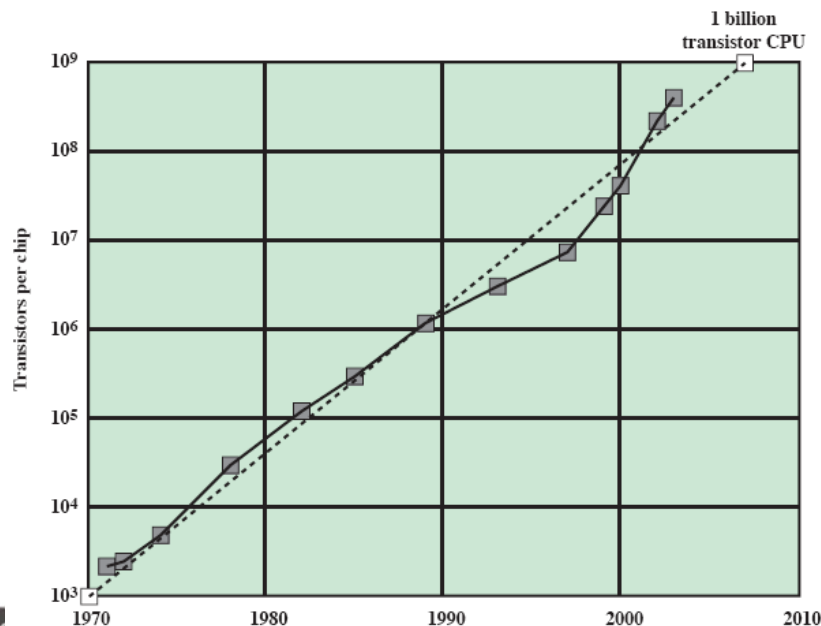
(3) . . . ele se dirige à calculadora e digita o número nela



(4) . . . por fim, ele se encaminha ao contador de posição e clica nele, o que deixa pronto para buscar a próxima instrução



- Conceitos importantes: arquitetura de von Neumann
 - Conceito de programa armazenado
 - Memória endereçada linearmente
 - Memória endereçada pelo número da posição independente de dados
- “Lei” de Moore
 - Desde 1970: dobra quantidade transistor a cada 18 meses

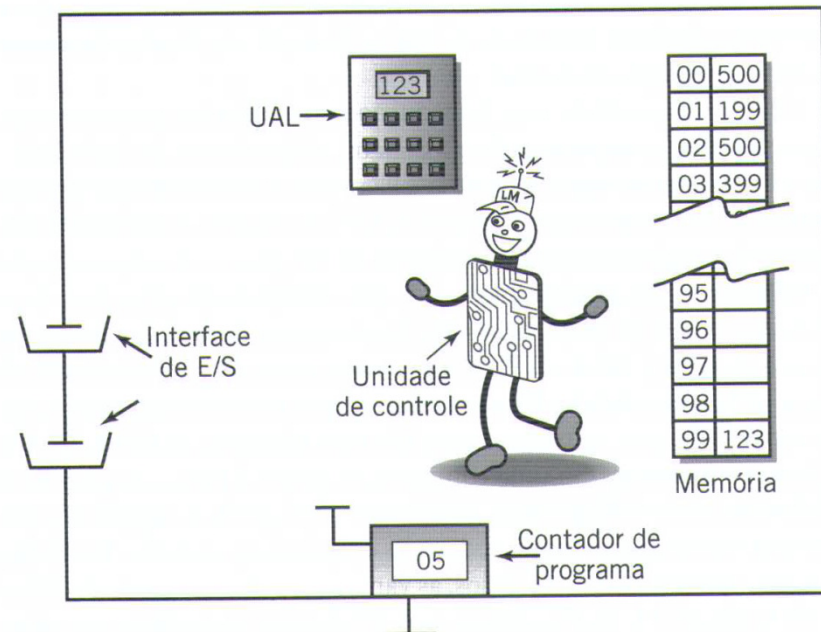
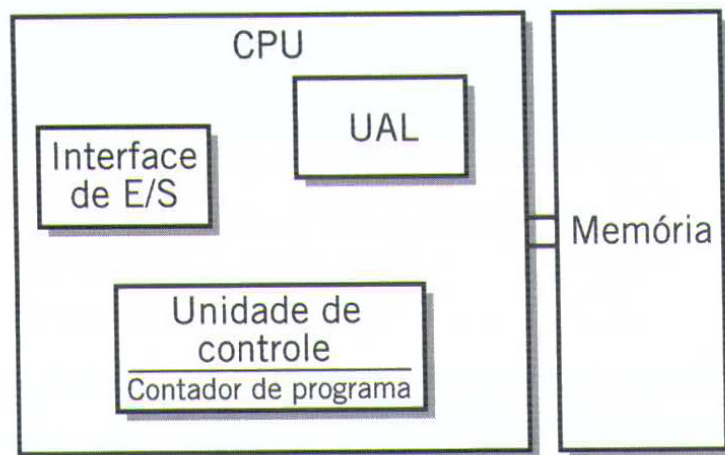


Evolução processadores Intel (arquitetura x86): 4004, 8080, 8086, 80286, 80386, 80486, Pentium (Pro), 'Xeon', Pentium II, Pentium III, Pentium 4, Core, Core 2 (Quad), i3/5/7,...



2.2 Diagrama de blocos: CPU e registradores

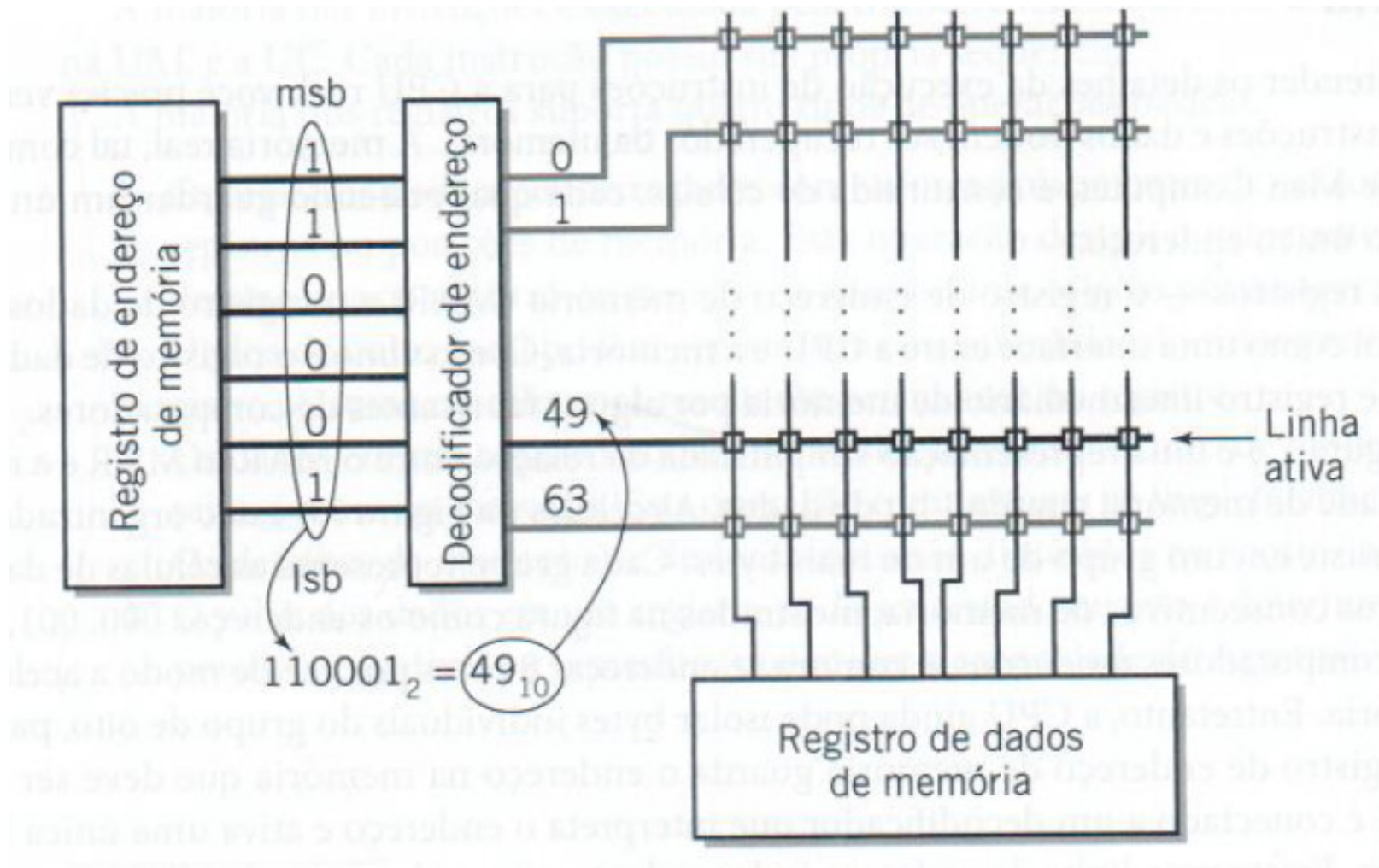
- Unidade central de processamento (CPU)
 - UAL + unidade de controle
 - Contador de programa = ponteiro de instrução
- Unidade de interface de barramento
 - Unidade de gerenciamento de memória (busca dados e instruções na memória)
 - Interface E/S



- Registradores
 - Finalidade única
 - Manipulados diretamente pela unidade controle durante execução de uma instrução
 - Não endereçáveis como a memória de programa
- Alguns registradores importantes:
 - Acumulador – A
 - “calculadora”
 - Registradores de uso geral - GR
 - Contador de programa - PC
 - Registro de instrução – IR
 - guarda instrução corrente
 - Registro de endereço de memória – MAR
 - guarda o endereço de posição de uma memória
 - Registro de dados de memória – MDR
 - guarda o dado que está sendo recuperado ou enviado à memória

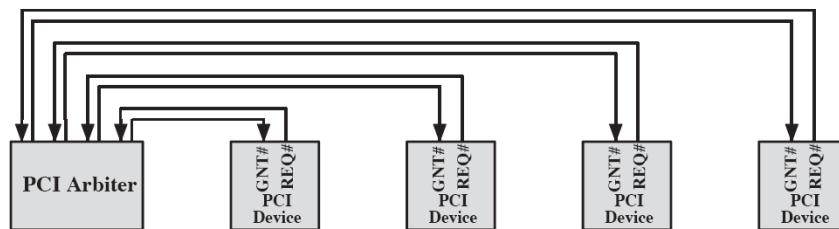
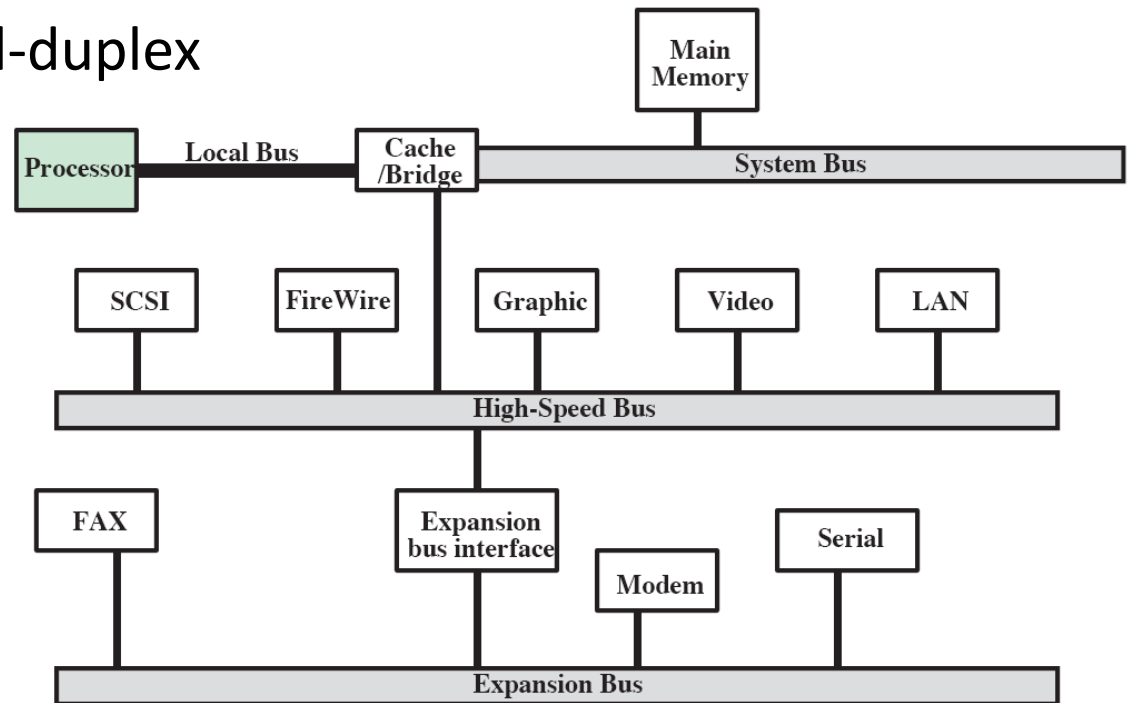


- Operação da memória:

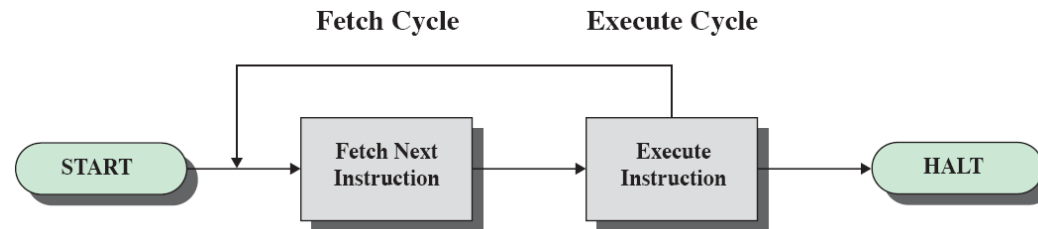


2.3 Barramentos

- Seriais x Paralelos
- Simplex x half-duplex x full-duplex
- Ponto a ponto x broadcast
- Hierarquia:
- Arbitração:



2.4 – Busca e execução de instruções



- Ciclo de busca
 - Etapa 1: $PC \rightarrow MAR$
 - Etapa 2: $MDR \rightarrow IR$
- Ciclo de execução
 - Depende da instrução
 - Exemplo: LOAD
 - Etapa 3: $IR [\text{endereço}] \rightarrow MAR$
 - Etapa 4: $MDR \rightarrow A$
 - Etapa 5: $PC + 1 \rightarrow PC$



- Outros exemplos:

- STORE

- $PC \rightarrow MAR$
 - $MDR \rightarrow IR$
 - $IR [\text{endereco}] \rightarrow MAR$
 - $A \rightarrow MDR$
 - $PC + 1 \rightarrow PC$

- ADD

- $PC \rightarrow MAR$
 - $MDR \rightarrow IR$
 - $IR [\text{endereco}] \rightarrow MAR$
 - $A + MDR \rightarrow A$
 - $PC + 1 \rightarrow PC$



• Instruções:

Tipo de instrução	Opcode	Representação simbólica	Descrição
Transferência de dados	00001010	LOAD MQ	Transfere o conteúdo de MQ para AC
	00001001	LOAD MQ,M(X)	Transfere o conteúdo do local de memória X para MQ
	00100001	STOR M(X)	Transfere o conteúdo de AC para o local de memória X
	00000001	LOAD M(X)	Transfere M(X) para o AC
	00000010	LOAD – M(X)	Transfere – M(X) para o AC
	00000011	LOAD M(X)	Transfere o valor absoluto de M(X) para o AC
	00000100	LOAD – M(X)	Transfere – M(X) para o acumulador
Desvio incondicional	00001101	JUMP M(X,0:19)	Apanha a próxima instrução da metade esquerda de M(X)
	00001110	JUMP M(X,20:39)	Apanha a próxima instrução da metade direita de M(X)
Desvio condicional	00001111	JUMP+ M(X,0:19)	Se o número no AC for não negativo, apanha a próxima instrução da metade esquerda de M(X)
	00010000	JUMP+ M(X,20:39)	Se o número no AC for não negativo, apanha a próxima instrução da metade direita de M(X)
Aritmética	00000101	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000111	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000110	SUB M(X)	Subtrai M(X) de AC; coloca o resultado em AC
	00001000	SUB M(X)	Subtrai M(X) de AC; coloca o resto em AC
	00001011	MUL M(X)	Multiplica M(X) por MQ; coloca os bits mais significativos do resultado em AC; coloca bits menos significativos em MQ
	00001100	DIV M(X)	Divide AC por M(X); coloca o quociente em MQ e o resto em AC
	00010100	LSH	Multiplica o AC por 2; ou seja, desloca à esquerda uma posição de bit
	00010101	RSH	Divide o AC por 2; ou seja, desloca uma posição à direita
Modificação de endereço	00010010	STOR M(X,8:19)	Substitui campo de endereço da esquerda em M(X) por 12 bits mais à direita de AC
	00010011	STOR M(X,28:39)	Substitui campo de endereço da direita em M(X) por 12 bits mais à direita de AC



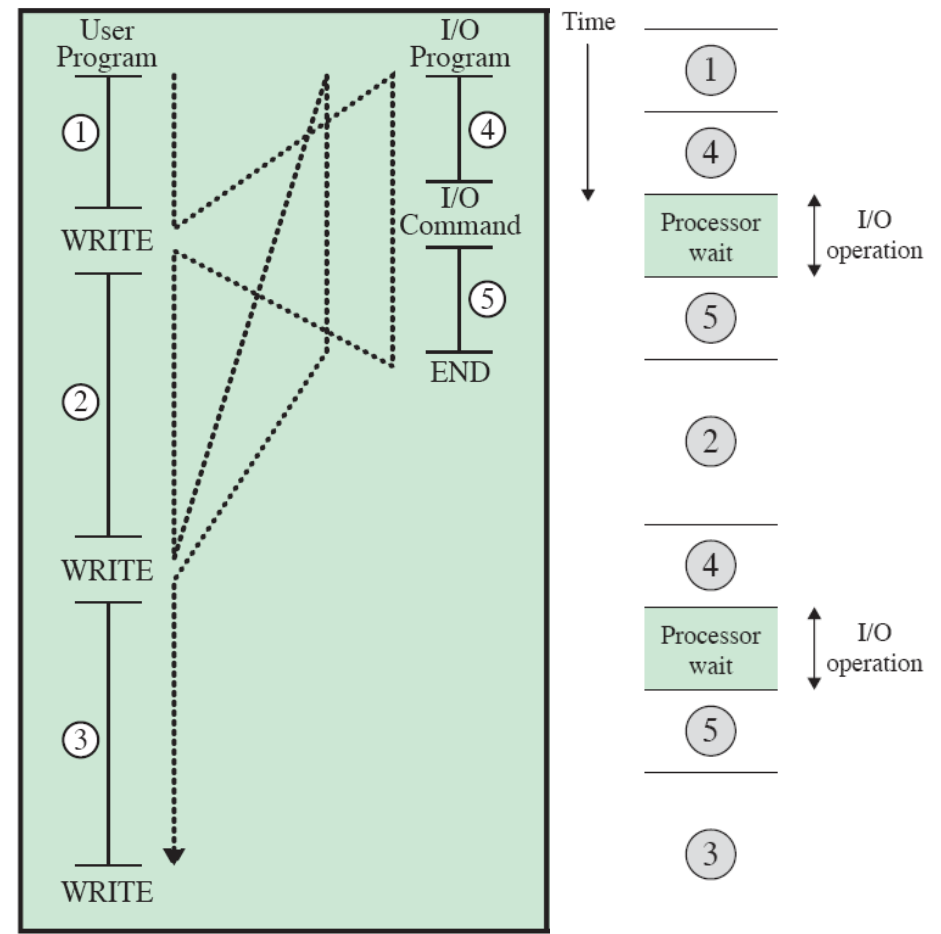
2.5 – Estudo de um sistema mínimo realizável: o SAP

- Material disponível em:
<http://www.alan.eng.br/arquivos/uP1.pdf>



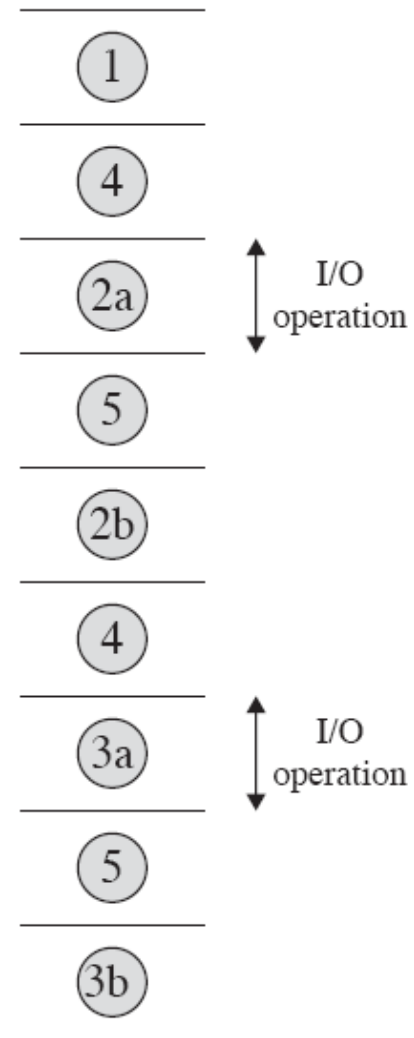
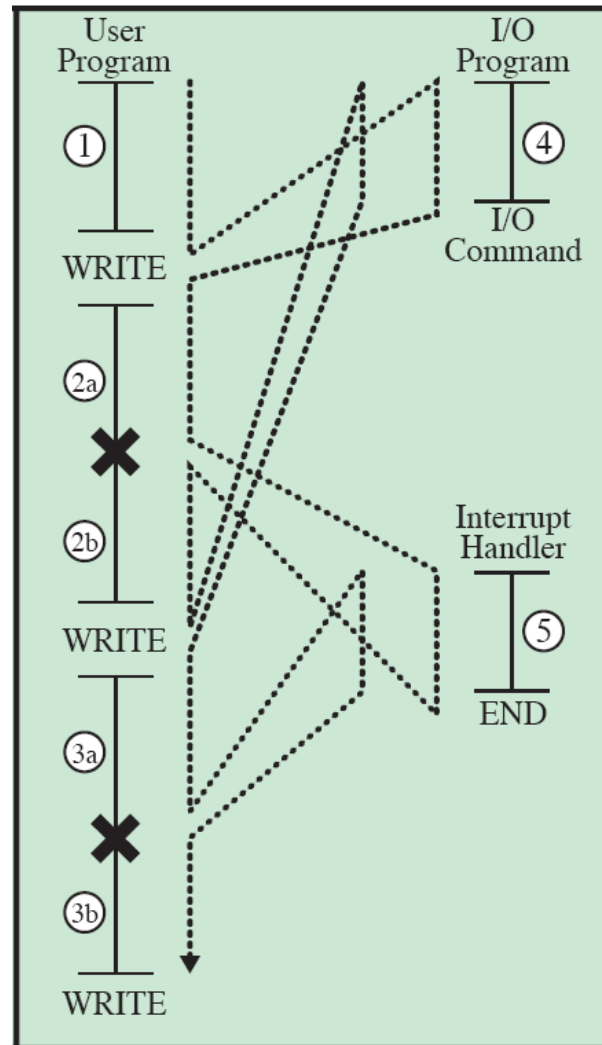
2.6 – Tópicos adicionais: DMA, cache, multiprocessamento, ...

- Interrupções:
 - Classes: software; timer; I/O; falha hardware
 - *Cenário 1: sem interrupções !*

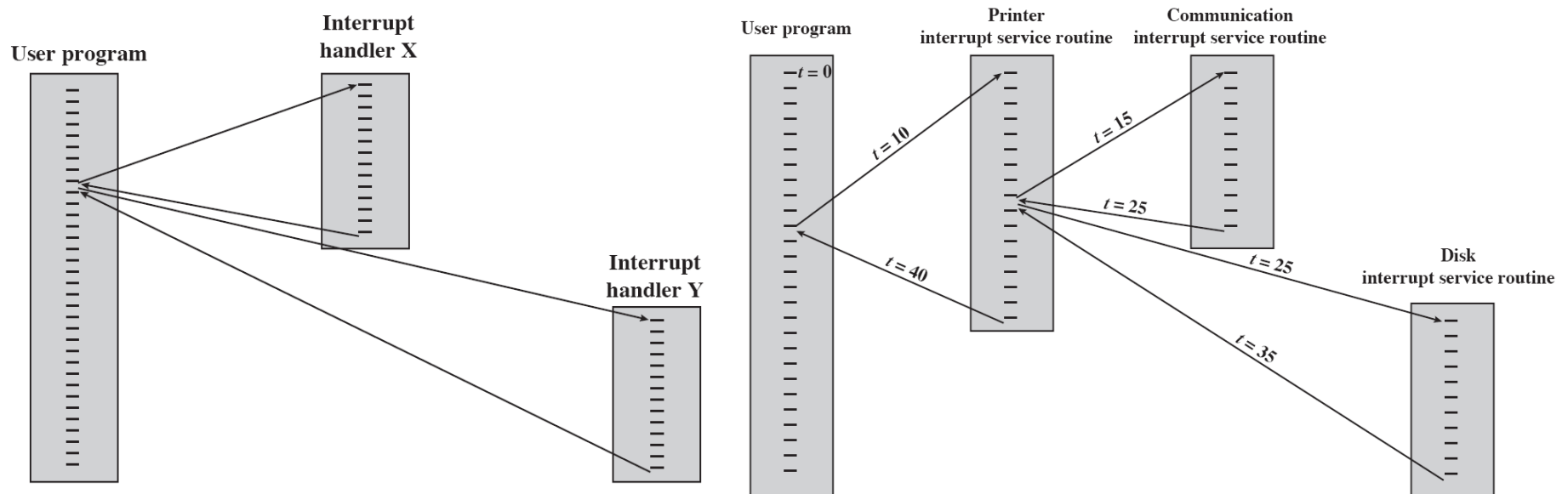


– Cenário 2: com interrupções

- Requisição de interrupção

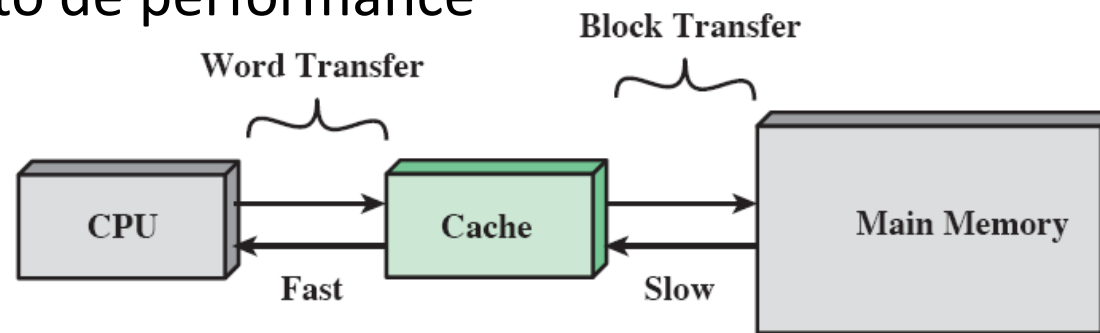


- Interrupções múltiplas
 - Desabilitar vetor de interrupções (sequencial)
 - Prioridade nas interrupções (paralela)

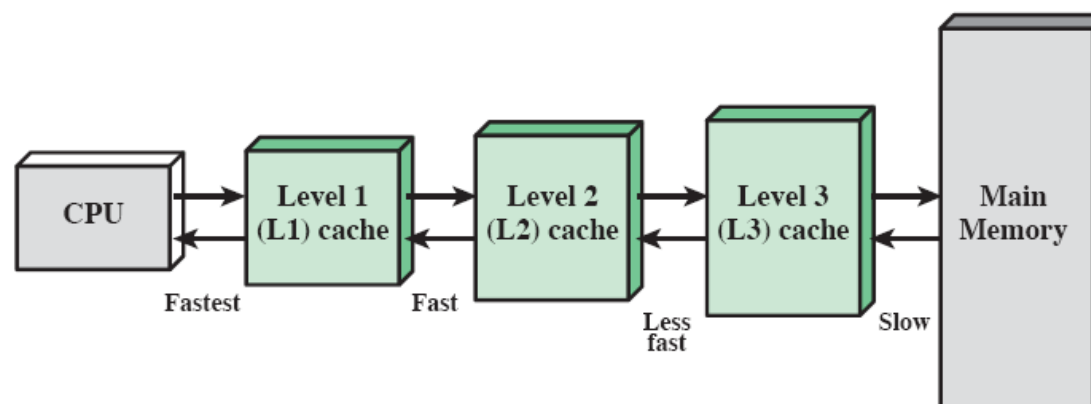


- Cache

- Não é visível
- Caráter “auxiliar”
- Aumento de performance

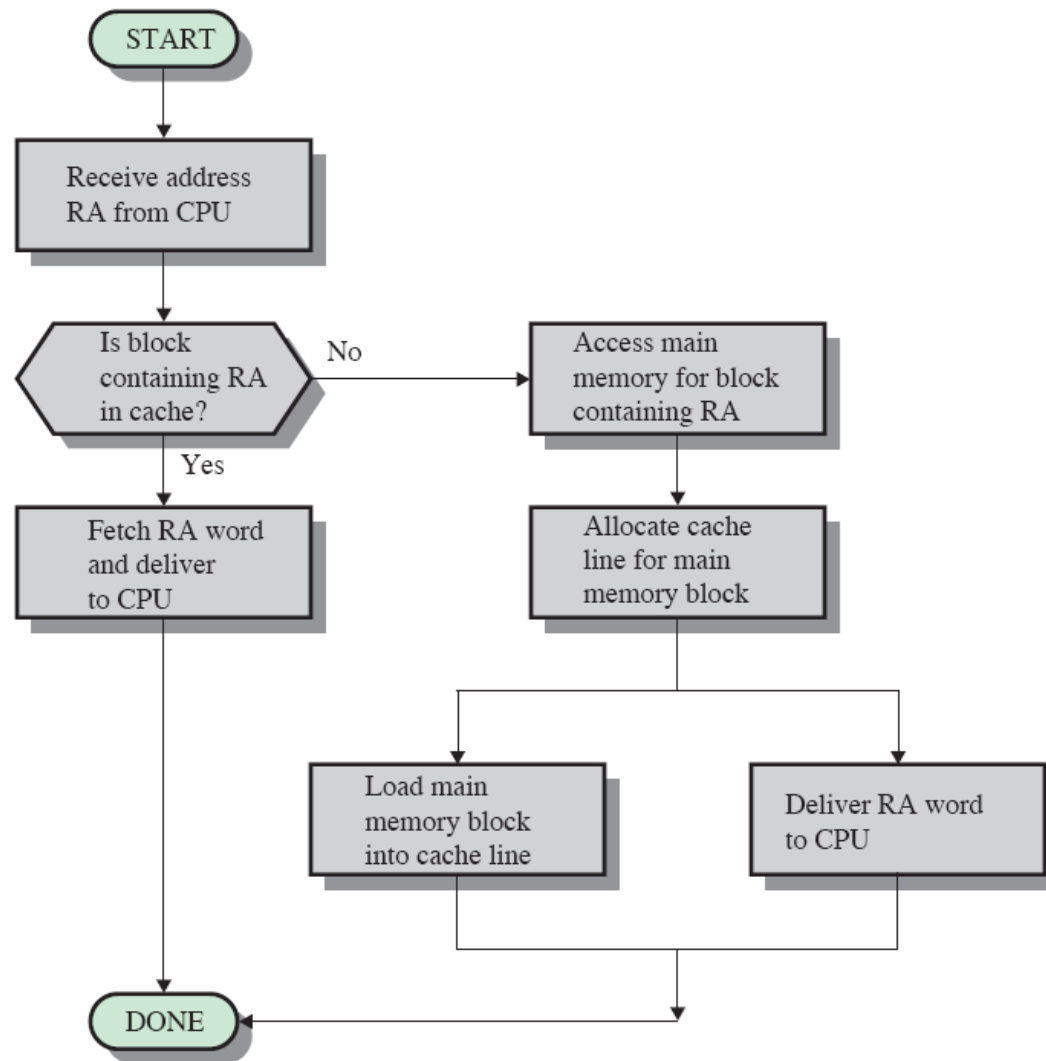


(a) Single cache

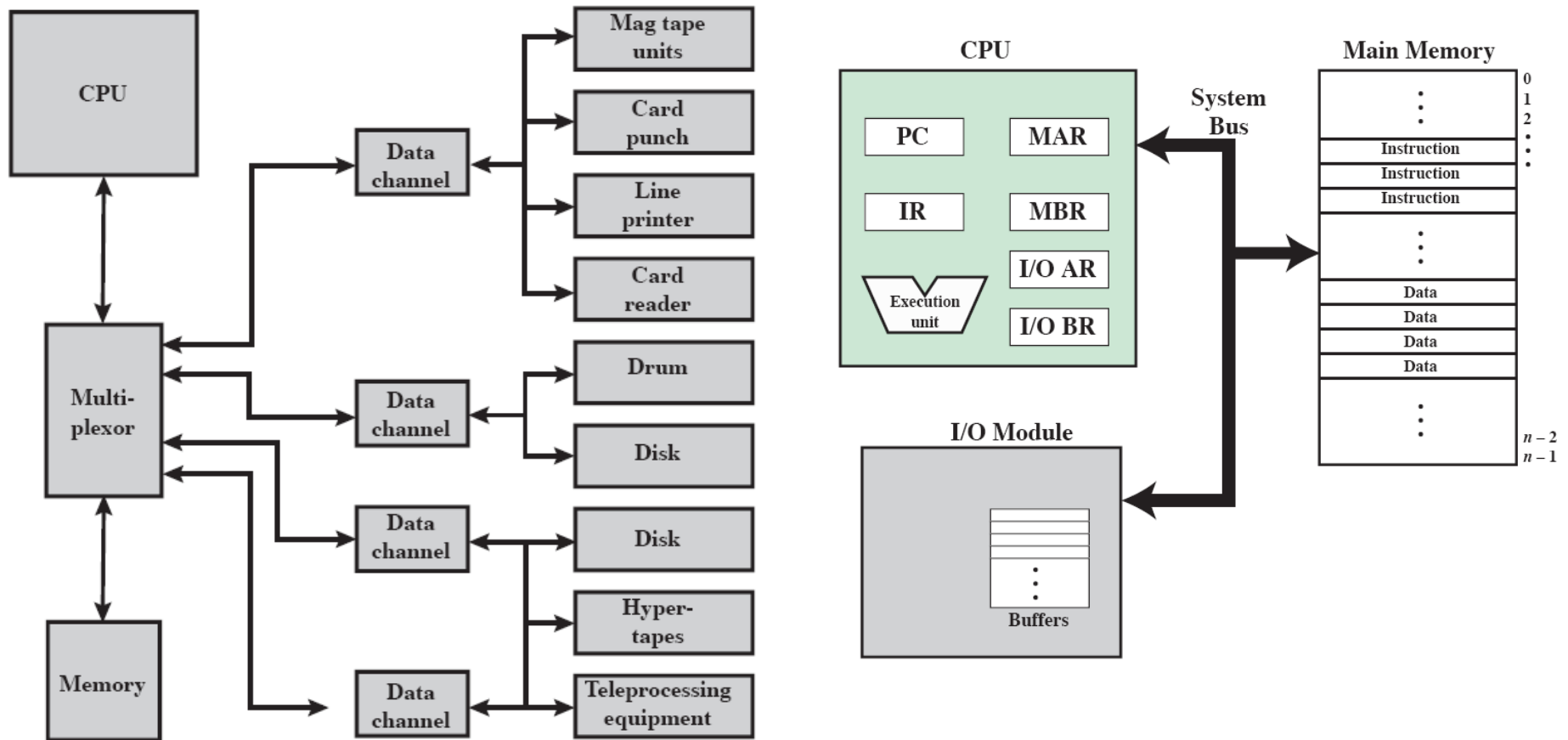


(b) Three-level cache organization

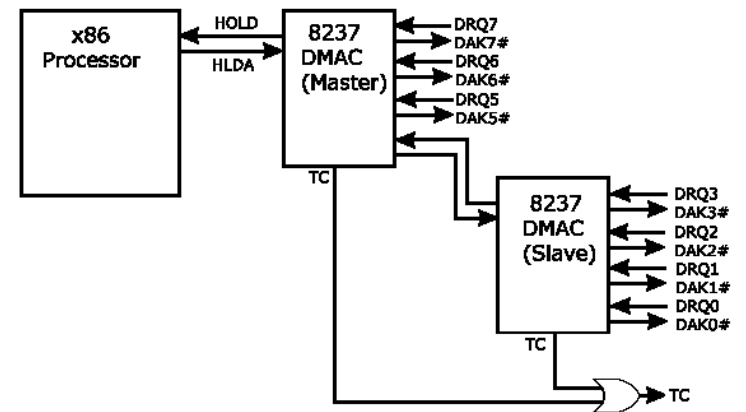
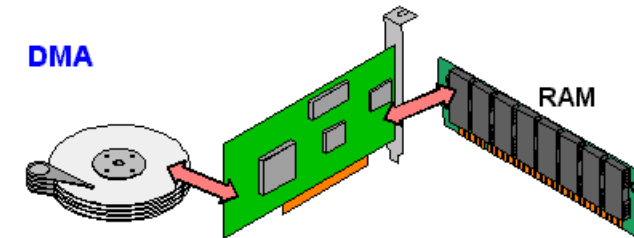
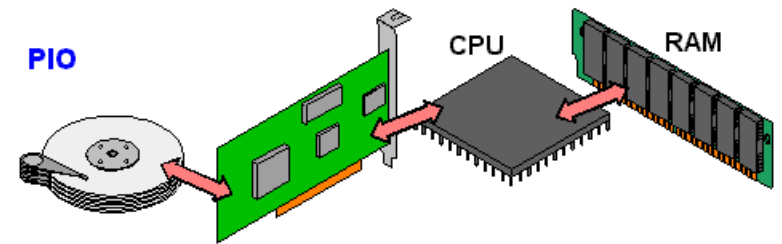
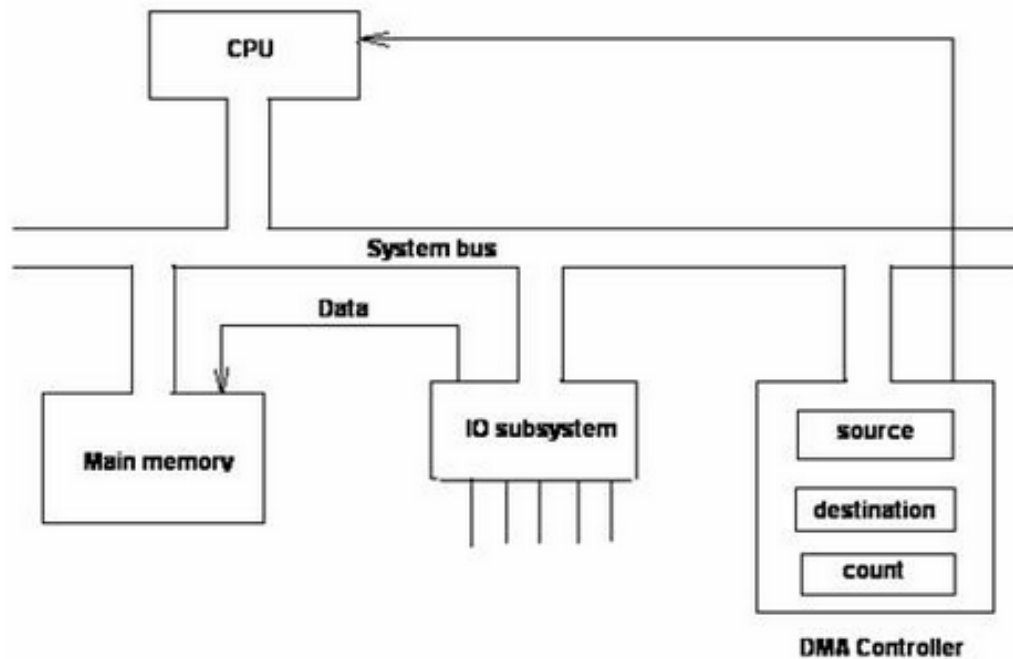
- Cache
 - “Blocos de palavras”



- Endereçamento memória e I/O:

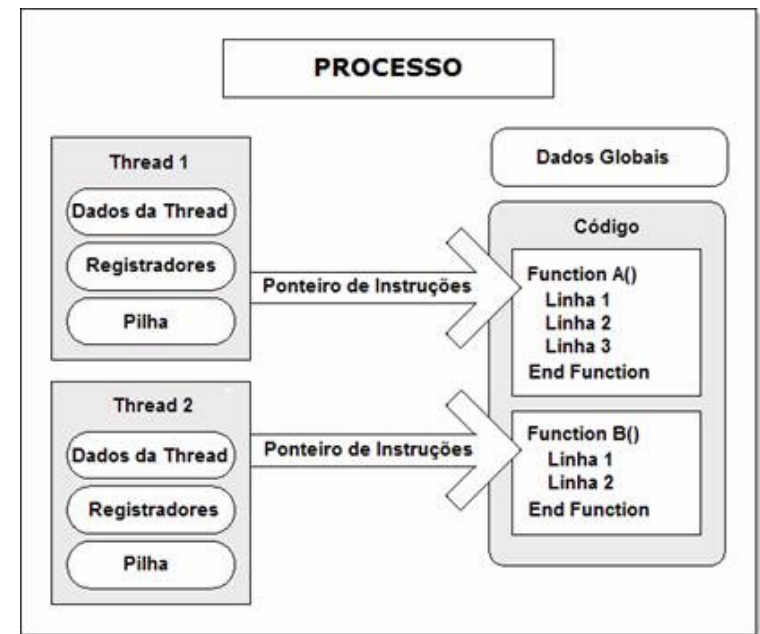
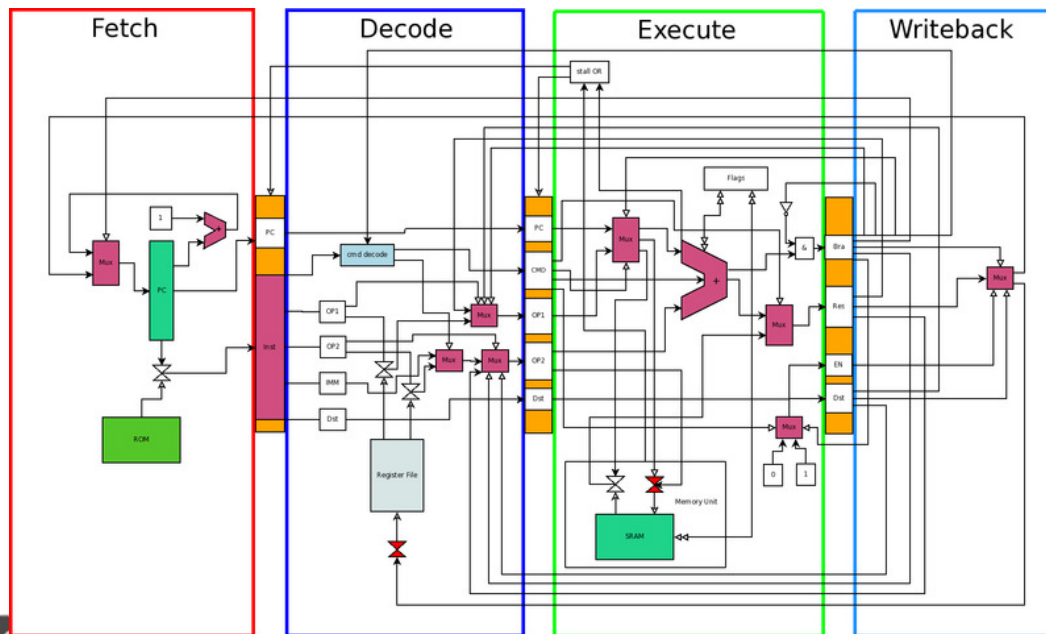
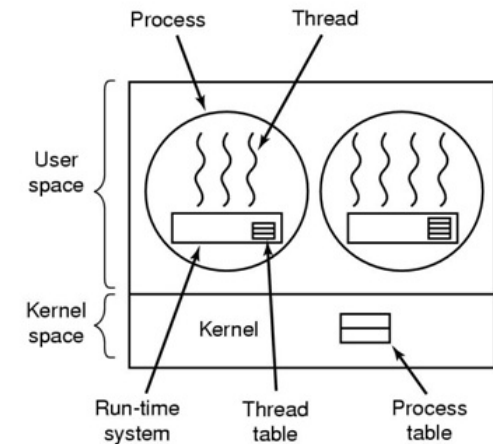


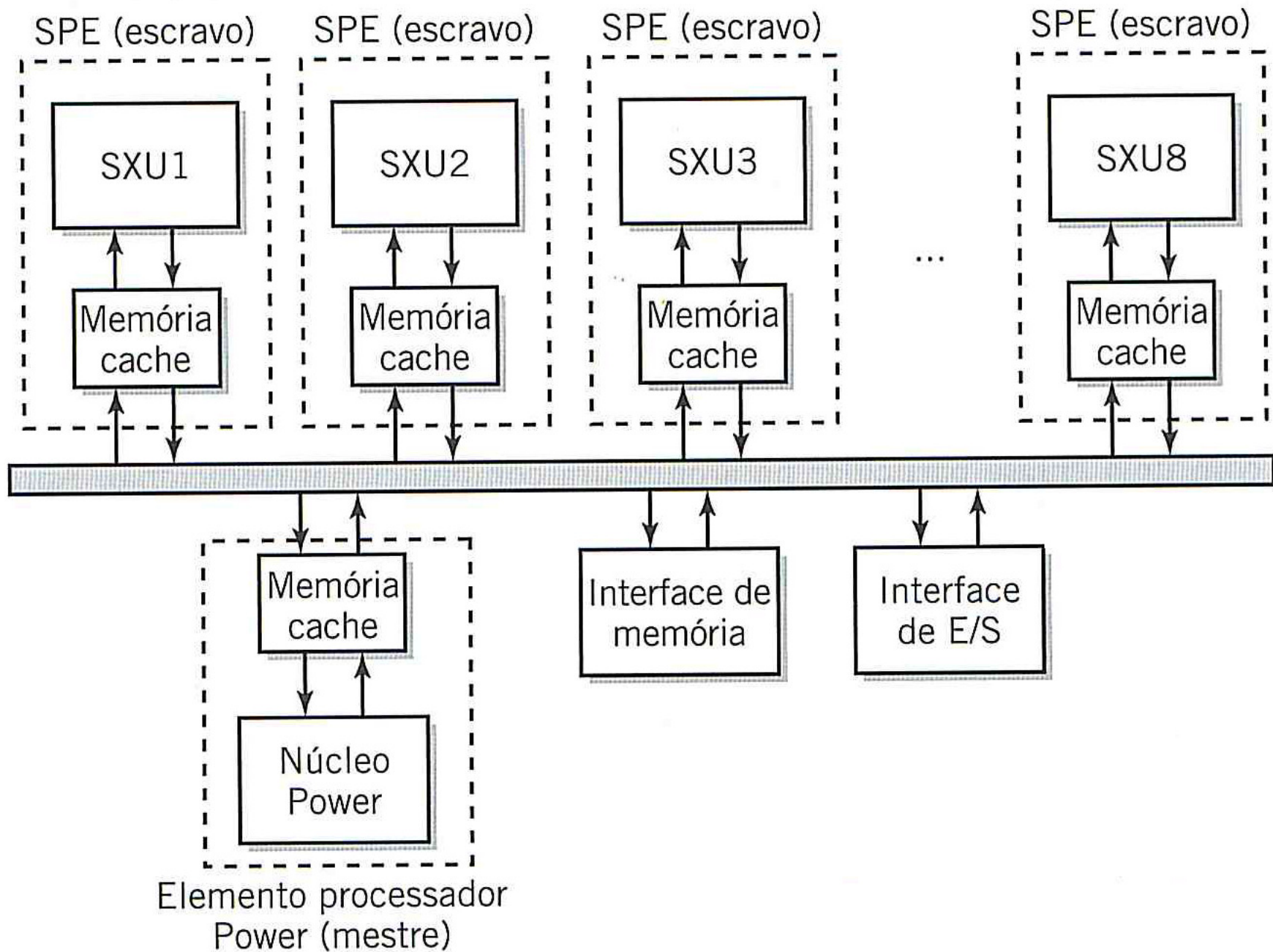
- DMA
 - Acesso direto à memória



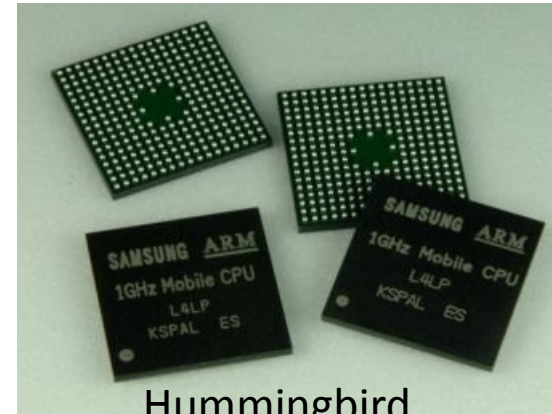
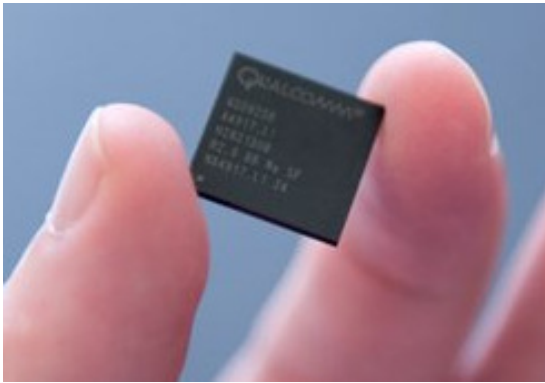
MULTIPROCESSAMENTO

- Multinúcleo x Pipeline
- SO designa tarefas (threads e escalonamento)
- MIPS
- Arquitetura mestre-escravo
 - mestre controla recursos escalonamento
- Arquitetura SMP (multiprocessamento simétrico)
 - Recursos iguais onde cada CPU escolana sua própria tarefa





2.7 – Arquiteturas microprocessadas modernas de 32 bits



Hummingbird



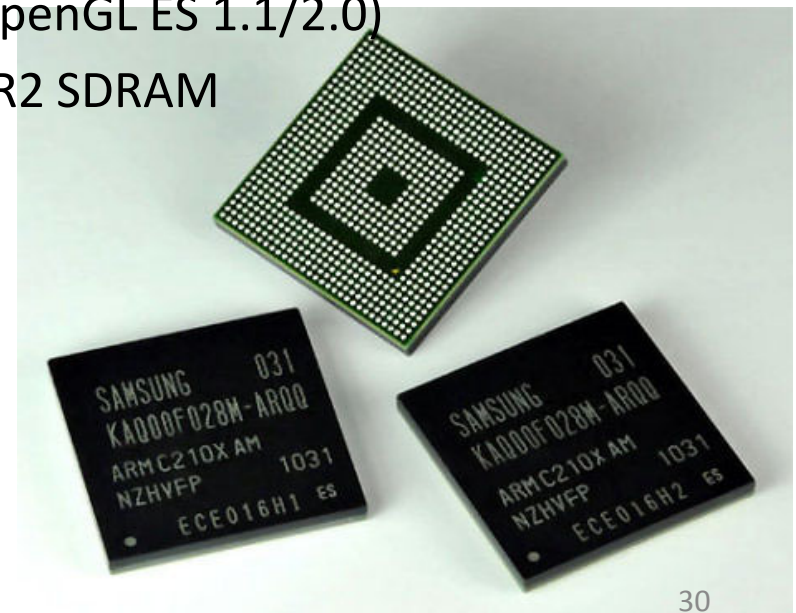
•Apple A5:

- CPU Structure: RISC
- Semiconductor Technology: CMOS
- Clock: 1,2GHz
- Primary CPU: Dual ARM Cortex-A9 Superscalar
- L1 Instruction Cache: 32 KB/Core
- L1 Data Cache: 32 KB/Core
- L2 Cache: 1,024 KB/Core
- Word Length: 32 bit
- Data Bus: 64/32-bit
- GPU: PowerVR SGX543MP2
- RAM Interface: Dual-channel 266 MHz DDR2 SDRAM
- GPS Module Type: Embedded GPS module



- **Hummingbird:**

- Semiconductor Technology: CMOS 45 nm
- Clock: 1,0GHz
- Primary CPU: ARM Cortex-A8 based CPU core (Exynos)
- L1 Instruction Cache: 32 KB/Core
- L1 Data Cache: 32 KB/Core
- L2 Cache: 1,024 KB/Core
- Word Length: 32 bit
- Data Bus: 32-bit
- GPU: PowerVR SGX 540 GPU (supported OpenGL ES 1.1/2.0)
- RAM Interface: Dual-channel 266 MHz DDR2 SDRAM



• **OMAP5** (Open Multimedia Applications Platform Processors)

- CPU dual ARM A15 (e outros 2 núcleos ARM cortex M4)
- CPU: Up to 1.5-1.7 GHz
- Tecnologia de 28nm
- Grupos de instrução: ARMv7
- Cache L1: 32 + 32KB
- Cache L2: 4MB
- Cache L3: não tem
- GPU: PowerVR SGX544MP2
- Usuários: BlackBerry PlayBook, Panasonic, LG, Motorola, PandaBoard, Samsung (Galaxy S II Galaxy Tab 2), TianyeIT, AT&T, Amazon Kindle, Archos, Barnes and Noble Nook Tablet, Archos, Google Glass.



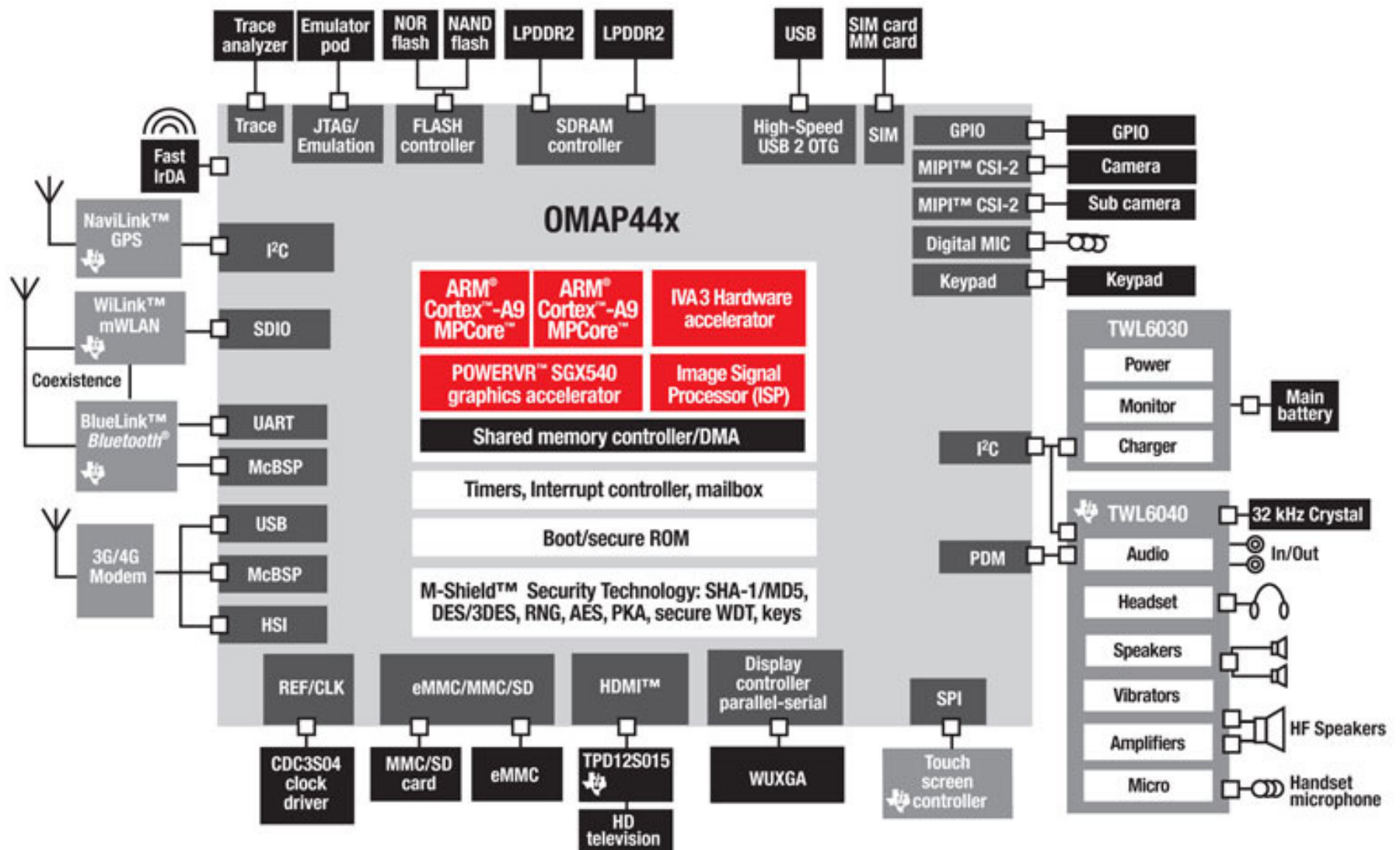
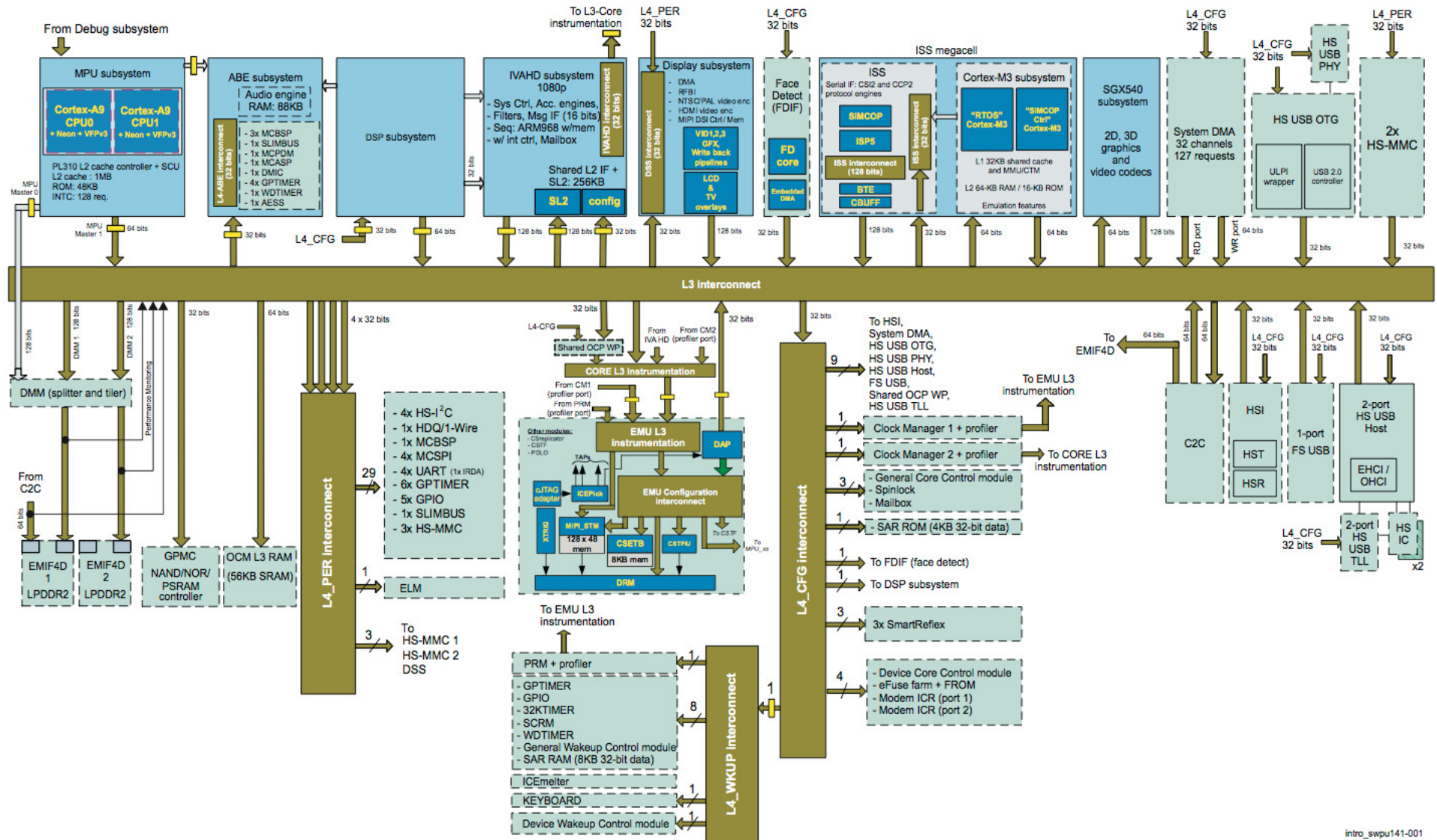


Figure 1-2. OMAP4430 Block Diagram



intro_swpu141-001

Altera

- NIOS II (32 bits)

Analog Devices

- Blackfin (16-32bits com DSP, baixo consumo, DMA)
- ADSP-21xx (DSP 32 bits)

Atmel

- AVR (8-32 btis, vários tipos de núcleo, sistemas embarcados, AVR32 com Java Machine)

Cypress

- PSoC - programmable sustem on CHIP (sistema embarcado programável, itens analógicos e PLD)

Dallas semiconductor

- 8051 família

Freescale/Morotola

- 68HC05 a 68HC11 (8 btis)
- 68HC12 a 68HC16 com o DSP56800 (16 bits)
- Kinetis, MCore e MPC fampilia (32 bits)



Intel

- MCS48 e MCS151 família (8btis, 8051 instruction set)
- MCS251 e MCS296 (16 a 32 bits)

Microchip

- PIC10(12 bits instruções), PIC12(12 bits instruções), PIC16(14 bits instruções), PIC18 (16 bits instruções) - Todos de 8btis verdadeiros
- PIC24 (16 bits de barramento de dados porém 24 bits de palavras de instruções)
- dsPIC (baseado no PIC24 com funções DSP)
- PIC32MX (32bits de barramento de dados com arquitetura superescalar)

NXP

- LPC700 a LPC900 (baseados 8051)
- LPC2100 a LPC2400 (núcleos ARM7 32bits)
- LPC2900 a LPC3200 (núcleos ARM9 32bits)
- LPC1100, LPC1200 e LPC800 (núcleos ARM CortexM0 32bits)
- LPC4000 a LPC4300(núcleo ARM cortex-M4 32 btis)

ST

- ST6, ST7, STM8 (8 btis)
- ST20, ARM7TDMI (ARM7), STR9 (ARM9), STM32F0 (ARM Cortex-M0), STM32 F1-F2 (ARM Cortex-M3)



Texas

- TMS370 (8 bits)
- MSP430 (16 bits)
- C2000 (núcleo C28x até 300MHz de 32 bits)
- Tiva e Hercules (ARM até 220MHz)
- Stellaris (ARM Cortex-M3)
- Sitara (ARM A8, 1GHz, 32bits, DDR3, 32+32KB cache)
- OMAP (ARM até 1,5GHz)
- TMS320 (DSP)

• Exemplo:

Processador	ARM-A15
Max. clock rate	1.0 GHz a 2.5 GHz
Min. feature size	32 nm a 28 nm
Instruções	ARMv7
Cores	1–4 per cluster
L1 cache	64 KB (32 KB I-cache, 32 KB D-cache) <i>per core</i>
L2 cache	4 MB <i>per cluster</i>
L3 cache	nenhuma

