

Estrutura do tema ISC

1. Representação de informação num computador
2. Organização e estrutura interna dum computador
3. Execução de programas num computador
4. O processador e a memória num computador
5. Da comunicação de dados às redes

Componentes (físicos) a analisar:

- o processador (info adicional):
 - o nível ISA (*Instruction Set Architecture*): tipos/formatos de instruções, acesso a operandos, ...
 - CISC versus RISC
 - paralelismo no processador: *pipeline*, super-escalaridade, ...
 - paralelismo fora do processador: *on-chip* e *off-chip*
- a hierarquia de memória:
 - *cache*, memória virtual, ...
- periféricos:
 - interfaces humano-computador (HCI)
 - arquivo de informação
 - comunicações (no tema 5...)

O processador: análise do nível ISA (*Instruction Set Architecture*)

Análise do nível ISA (*Instruction Set Architecture*) (1)

Tópicos a analisar

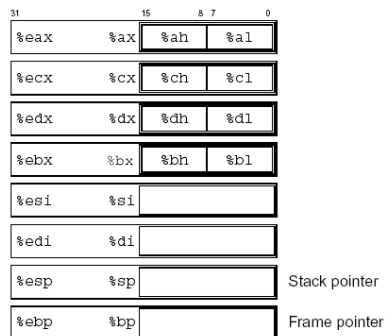
- operações num processador
- registos visíveis aos programador
- modos de acesso aos operandos
- tipos de instruções presentes num CPU
- formatos de instruções em linguagem máquina
- instruções de *input/output*
- ordenação de *bytes*

Operações num processador

- nº de operandos em cada instrução
 - 3-operandos (RISC, ...)
 - 2-operandos (IA32, ...)
 - 1-operando (microcontroladores, ...)
 - 0-operandos (*stack-machine*, ...)
- localização dos operandos
 - variáveis escalares (registos...)
 - variáveis estruturadas (memória...)

Registos visíveis aos programador (inteiros)

- em arquitecturas RISC (32 registos genéricos...)
- no IA32



Tipos de instruções presentes num CPU

- transferência de informação
 - de/para registos/memória, ...
- operações aritméticas e lógicas
 - soma, subtração, multipl, div, ...
 - AND, OR, NOT, XOR, comparação, ...
 - deslocamento de bits, ...
- controlo do fluxo de execução
 - para apoio a estruturas de controlo
 - para apoio à invocação de procedimentos
- outras...

Modos de acesso aos operandos

- em arquitecturas RISC
 - em operações aritméticas/lógicas, sempre em registo
 - em load/store usando 1 ou 2 modos de endereço à memória
- no IA32

Type	Form	Operand value	Name
Immediate	$\$Imm$	Imm	Immediate
Register	E_a	$R[E_a]$	Register
Memory	Imm	$M[Imm]$	Absolute
Memory	(E_b)	$M[R[E_b]]$	Indirect
Memory	$Imm(E_b)$	$M[Imm + R[E_b]]$	Base + displacement
Memory	(E_b, E_i)	$M[R[E_b] + R[E_i]]$	Indexed
Memory	$Imm(E_b, E_i)$	$M[Imm + R[E_b] + R[E_i]]$	Indexed
Memory	$(, E_i, s)$	$M[R[E_i] \cdot s]$	Scaled indexed
Memory	$Imm(, E_i, s)$	$M[Imm + R[E_i] \cdot s]$	Scaled Indexed
Memory	(E_b, E_i, s)	$M[R[E_b] + R[E_i] \cdot s]$	Scaled indexed
Memory	$Imm(E_b, E_i, s)$	$M[Imm + R[E_b] + R[E_i] \cdot s]$	Scaled indexed

Ex: instruções de transferência de info no IA32

mov	S, D	$D \leftarrow S$	Move (byte,word,long_word)
movzbl	S, D	$D \leftarrow \text{ZeroExtend}(S)$	Move Zero-Extended Byte
movsbl	S, D	$D \leftarrow \text{SignExtend}(S)$	Move Sign-Extended Byte
push	S	$\%esp \leftarrow \%esp - 4; \text{Mem}[\%esp] \leftarrow S$	Push
pop	D	$D \leftarrow \text{Mem}[\%esp]; \%esp \leftarrow \%esp + 4$	Pop
lea	S, D	$D \leftarrow \&S$	Load Effective Address

D – destino [Reg | Mem] **S** – fonte [Imm | Reg | Mem]
D e **S** não podem ser ambos operandos em memória no IA32

Ex: instruções aritméticas/lógicas no IA32

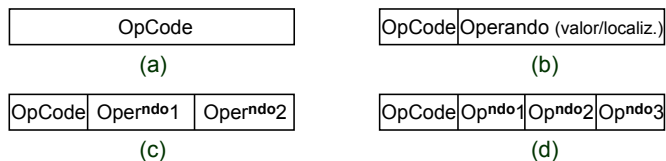
inc	D	$D \leftarrow D + 1$	Increment
dec	D	$D \leftarrow D - 1$	Decrement
neg	D	$D \leftarrow -D$	Negate
not	D	$D \leftarrow \sim D$	Complement
add	S, D	$D \leftarrow D + S$	Add
sub	S, D	$D \leftarrow D - S$	Subtract
imul	S, D	$D \leftarrow D * S$	32 bit Multiply
and	S, D	$D \leftarrow D \& S$	And
or	S, D	$D \leftarrow D S$	Or
xor	S, D	$D \leftarrow D \wedge S$	Exclusive-Or
shl	k, D	$D \leftarrow D \ll k$	Left Shift
sar	k, D	$D \leftarrow D \gg k$	Arithmetic Right Shift
shr	k, D	$D \leftarrow D \gg k$	Logical Right Shift

Ex: instruções de controlo de fluxo no IA32

jmp	Label	$\%eip \leftarrow \text{Label}$	Unconditional jump
je	Label		Jump if Zero/Equal
js	Label		Jump if Negative
jg	Label		Jump if Greater (signed >)
jge	Label		Jump if Greater or equal (signed >=)
ja	Label		Jump if Above (unsigned >)
call	Label	pushl $\%eip$; $\%eip \leftarrow \text{Label}$	Procedure call
ret		popl $\%eip$	Procedure return

Formatos de instruções em linguagem máquina

– campos numa instrução

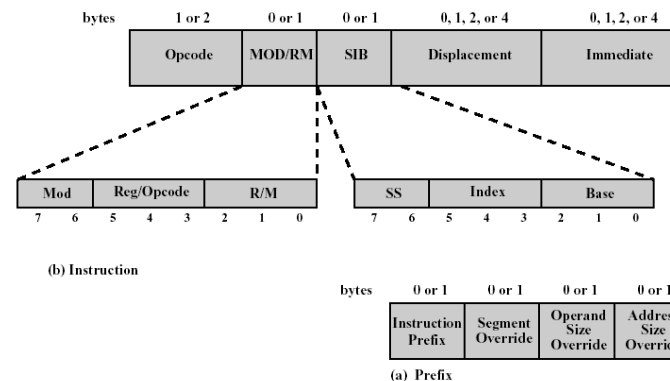


– comprimento das instruções

- variável (prós e contras; IA32...)
- fixo (prós e contras; RISC...)

– exemplos de formatos de instruções

Formatos de instruções no Pentium



Formatos de instruções no MIPS (RISC)



Desempenho nos Sistemas de Computação: oportunidades para otimizar

Otimização do desempenho

- com introdução de **paralelismo**
 - ao nível do processo (sistemas concorrentes/paralelos/distribuídos)
 - só fio de execução (*multi-threading/ -core / -chip ...*)
 - processo (memória partilhada/distribuída)
 - ao nível da instrução (*Instruction Level Parallelism*)
 - só nos dados (processamento vectorial)
 - paralelismo desfasado (*pipeline*)
 - paralelismo "real" (VLIW, super-escalar)
 - no acesso à memória
 - paralelismo desfasado (*interleaving*)
 - paralelismo "real" (maior largura do *bus*)
- com **hierarquia de memória**
 - *cache ...*

Instruções de *input/output*

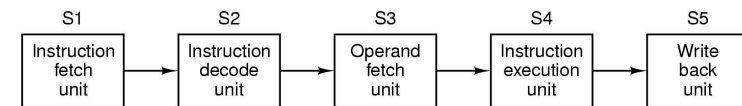
- finalidade
 - escrita de comandos
 - leitura de estado
 - escrita/leitura de dados
- específicas (requer sinais de controlo no *bus...*) ; ou
- idênticas ao acesso à memória
 - » *memory mapped I/O*

Ordenação de bytes

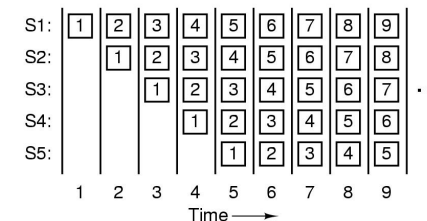
- *big-endian*
- *little-endian*

Paralelismo no processador Exemplo 1

Exemplo de *pipeline*

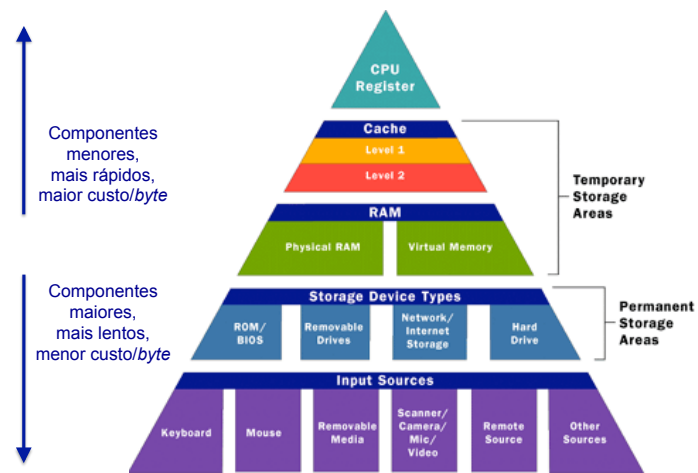


(a)



(b)

Hierarquia de memória



AJProença, Sistemas de Computação, UMinho, 2010/11

17

© HowStuffWorks

Evolução do Intel x86 : pré-Pentium (visão do programador)

Nome	Data	Nº transístores
8086	1978	29K
		– processador 16-bits (registos + ALU); base do IBM PC & DOS – espaço de endereçamento limitado a 1MB (DOS apenas vê 640K)
80286	1982	134K
		– endereço 24-bits e protected-mode; base do IBM PC-AT & Windows
386	1985	275K
		→ primeiro IA-32 !! – estendido para 32-bits: registos + op. inteiros + endereçamento – memória segmentada+paginaada, capaz de correr Unix
486	1989	1.9M
		– integração num único chip: 386, co-proc 387, até 16kB cache L1 – poucas alterações na arquitectura interna do processador

AJProença, Sistemas de Computação, UMinho, 2010/11

18

Evolução do IA-32: família Pentium (visão do programador)

Pentium	1993	3.1M	(= P5)
			– arquitectura superscalar, com 2 pipelines de inteiros (de 5 níveis)
PentiumPro	1995	5.5M	(= P6 , aka i686)
			– out-of-order execution, 14 níveis pipeline, 3-issue superscalar – endereço 36-bits, cache L2 on-package
Pentium/MMX	1997	4.5M	
			– SIMD: opera com vectores de 64-bits, tipo <i>int</i> de 1, 2, ou 4 bytes
Pentium II	1997	7.5M	(= Pro + MMX)
Pentium III	1999	8.2M	
			– “Streaming SIMD Ext”, SSE: vectores 128-bits, <i>int</i> / <i>fp</i> 1/2/4 bytes
Pentium 4	2000	42M	(= NetBurst , aka i786)
			– trace cache, pipeline muito longo (20 ou 31), suporta multi-threading – SSE2: mais instruções e com dados <i>fp</i> de 8-bytes
Pentium M	2003	77M	(= P-M)
			– arquitectura mais próxima do Pentium III (eficiência energética)

AJProença, Sistemas de Computação, UMinho, 2010/11

19

Evolução do IA-32 para Intel 64 (visão do programador)

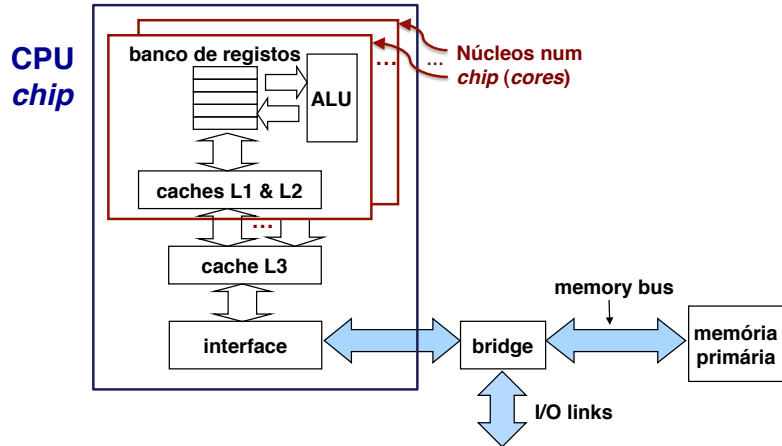
- IA-32 ou x86 *open architecture* cresce para 64-bits
 - HP e Intel propõem arquitectura incompatível, IA-64 (Itanium CPU)
 - AMD anuncia em 1999 extensão do x86: x86-64
 - Intel segue AMD: IA-32e (Fev-04) EM64T (Mar-04), ou Intel-64 (2006)
 - AMD64 e Intel-64 diferentes; compiladores usam sub-set comum
- arquitectura **Core** surge em 2006 (151M transístores)
 - desenvolvida pela mesma equipa que o P-M (Israel)
 - 14 níveis de pipeline (como P6), mas 4-issue superscalar
 - 2 níveis de cache on-chip
 - multi-core on-chip (mesmo o Solo!) e virtualização por h/w
 - suporta fusão de instruções RISC (μ -ops na terminologia Intel)
 - arquitectura Core 2 é integralmente 64-bit (Intel 64)

AJProença, Sistemas de Computação, UMinho, 2010/11

20

A cache em arquitecturas multicore

As arquitecturas multicore mais recentes:



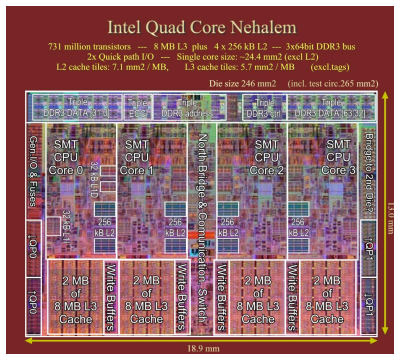
AJProença, Sistemas de Computação, UMinho, 2010/11

21

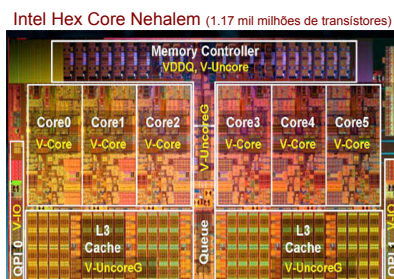
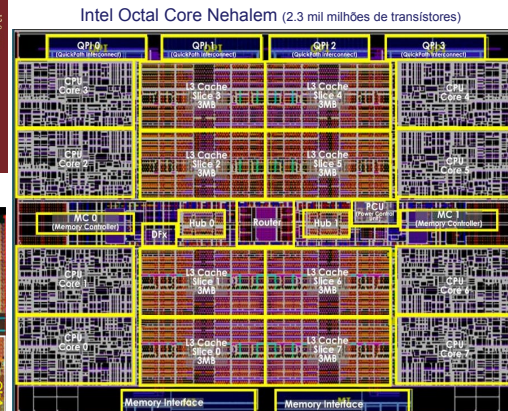
- arquitectura **Nehalem** anunciada em 2008 (731M transístores)
 - inspirada no NetBurst (com multi-threading e maiores clock rates)
 - 2 a 8 cores por chip, com cache L3 on-chip
 - com conexão ponto-a-ponto inter-CPUs
 - integra controlador de memória numa arquitectura NUMA
- arquitectura **Sandy Bridge** anunciada em 2010 (>1000M transístores)
 - evolui para arquitectura híbrida multicore, integrando o processador gráfico
 - interface com o processador gráfico através da cache L3
 - processamento vectorial estendido para 256-bits
 - integração no chip do interface PCIe 16x

AJProença, Sistemas de Computação, UMinho, 2010/11

22



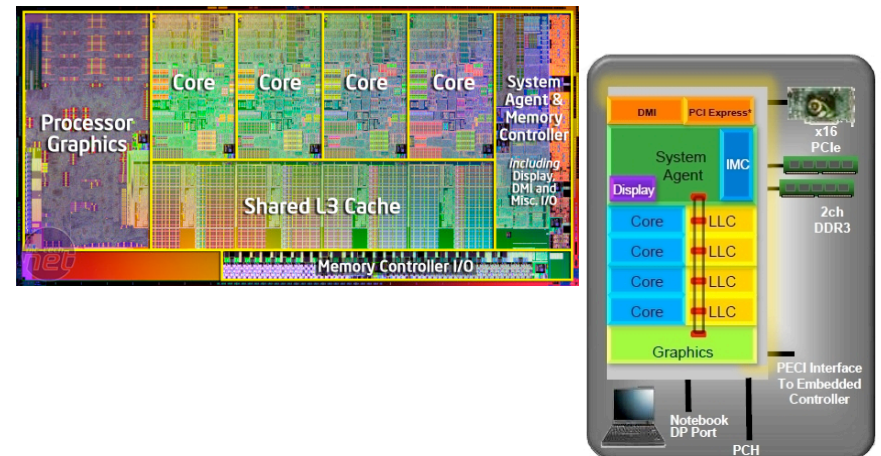
Gerações de Pentium recentes



AJProença, Sistemas de Computação, UMinho, 2010/11

23

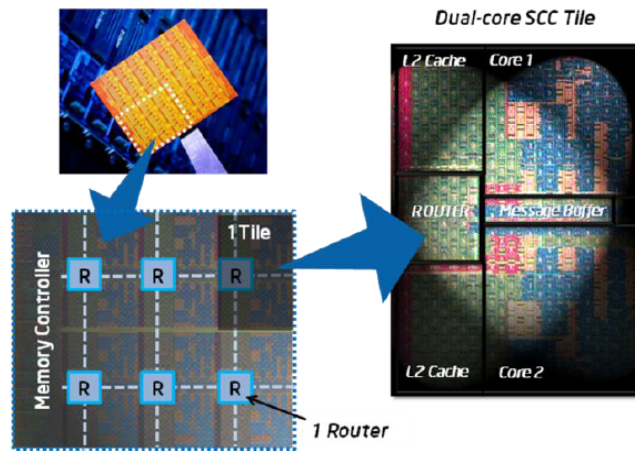
Lançamento da Intel em 2010:
Sandy Bridge



AJProença, Sistemas de Computação, UMinho, 2010/11

24

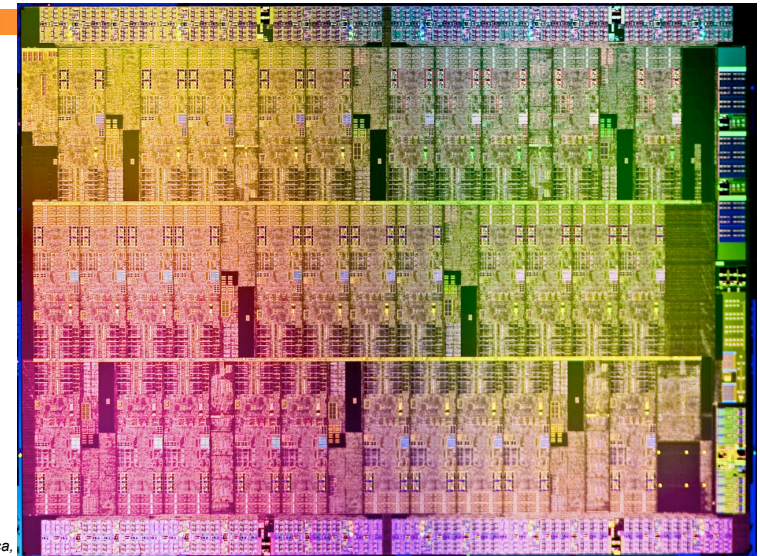
Protótipos da Intel em 2010/11:
Single-chip Cloud Computer



AJProença, Sistemas de Computação, UMinho, 2010/11

25

Protótipos da Intel em 2010/11:
Knights Corner com 50 cores (Larrabee)



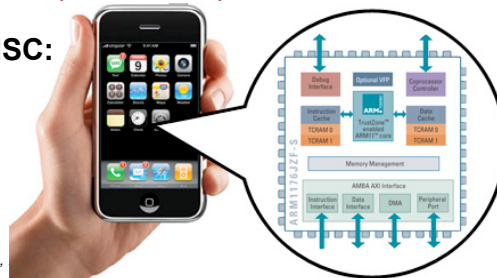
AJProença,

CISC versus RISC

Caracterização das arquitecturas RISC

- conjunto reduzido e simples de instruções
- formatos simples de instruções
- operandos sempre em registos
- modos simples de endereçamento à memória
- uma operação elementar por ciclo máquina

Exemplo de um chip RISC:
ARM



AJProença, Sistemas de Computação, UMinho,