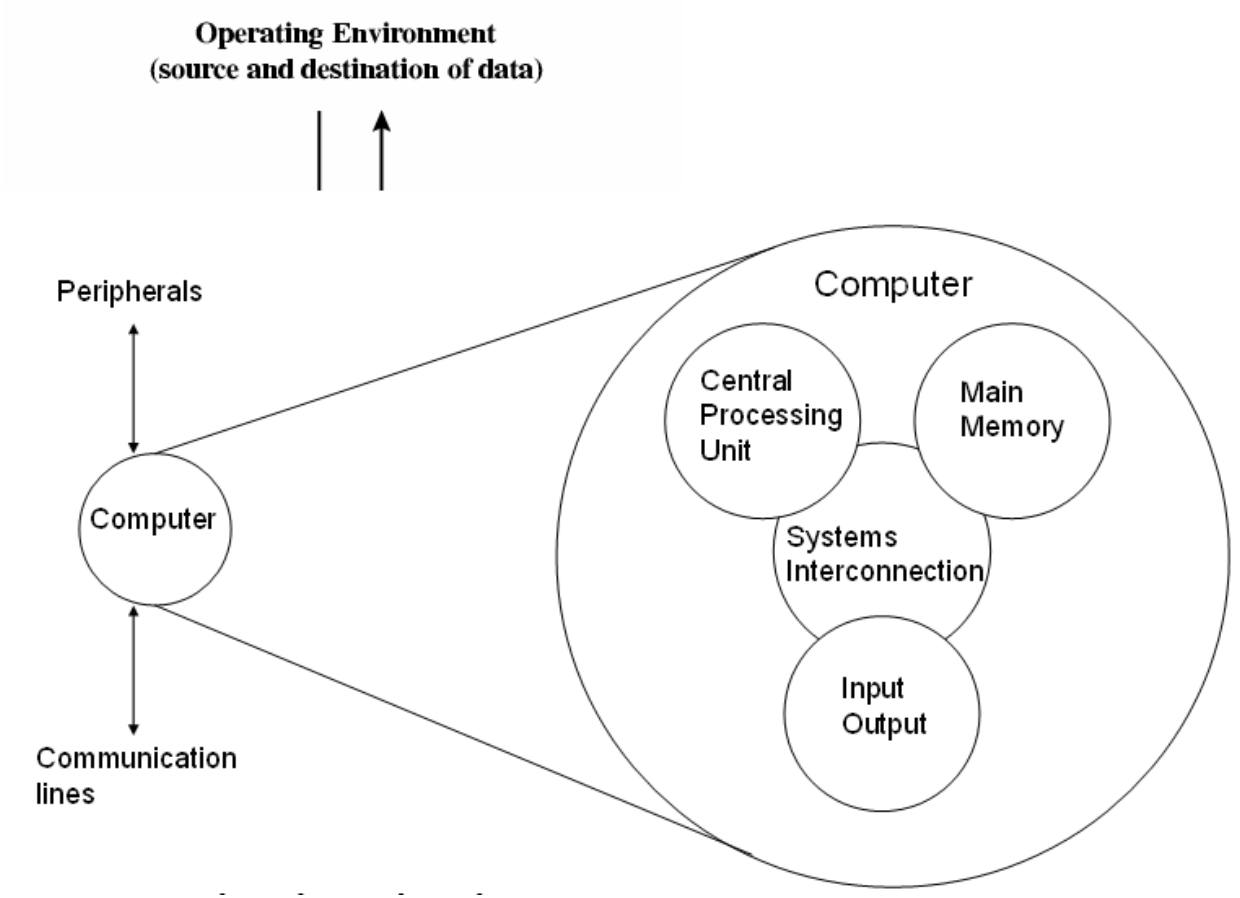
ESTRUTURA CPU

TeSP de Aplicações Móveis

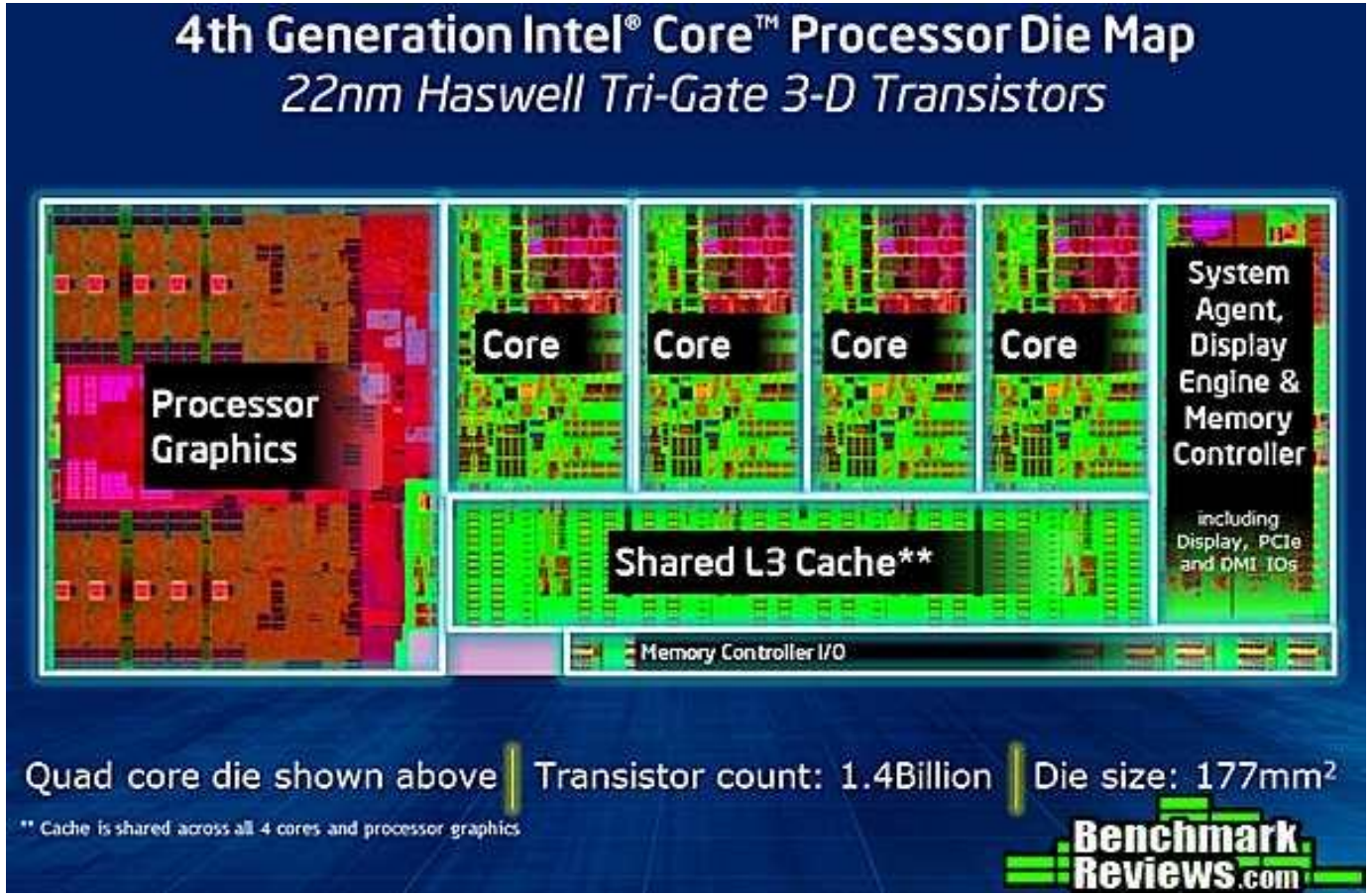
André Martins Pereira

IPCA
INSTITUTO POLITÉCNICO
DO CÁVADO E DO AVE
ESCOLA SUPERIOR DE TECNOLOGIA

ESTRUTURA DE UM PROCESSADOR



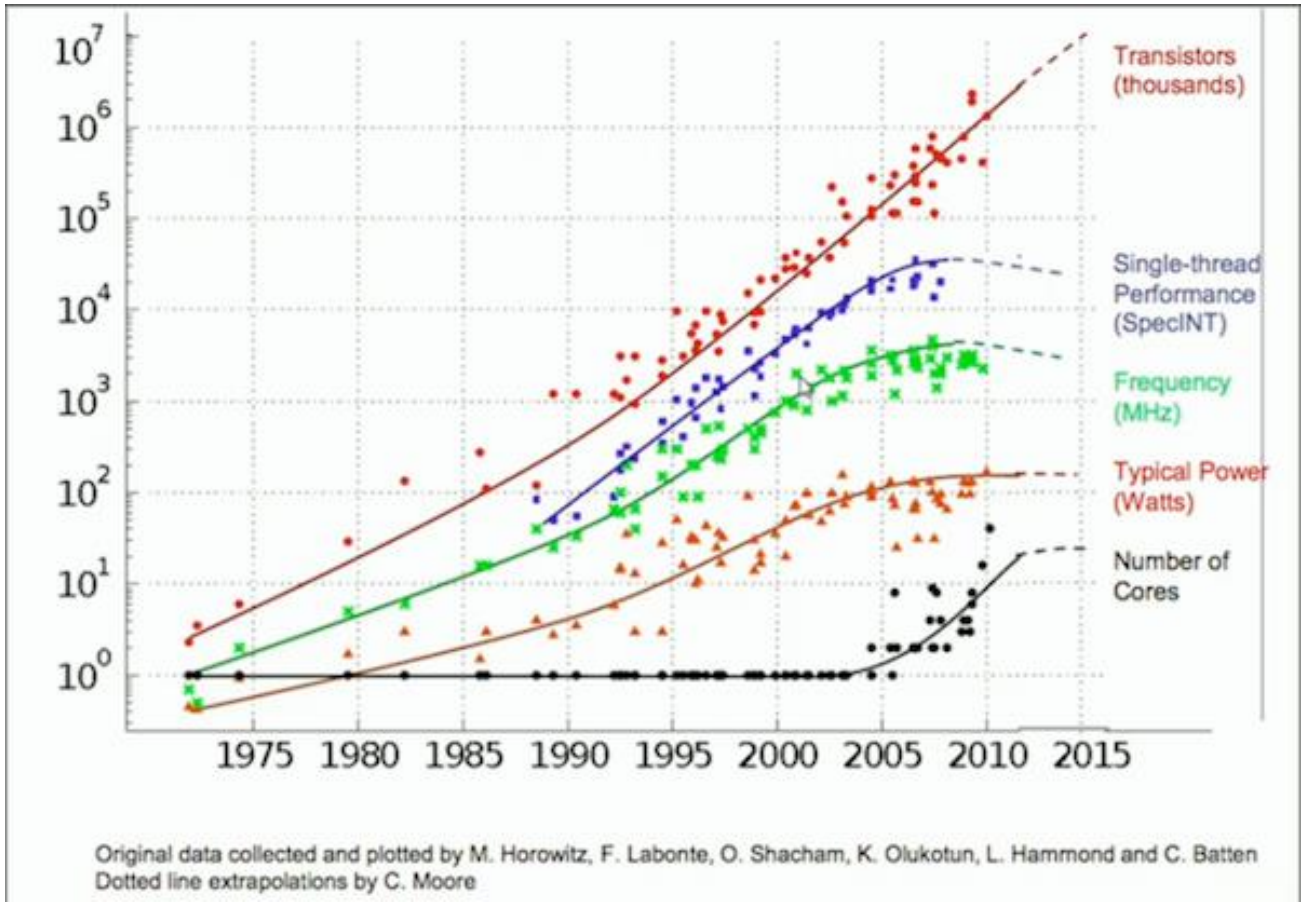
ESTRUTURA DE UM PROCESSADOR



CLOCK DE UM PROCESSADOR

- Frequência de clock
  - Quantidade de ciclos que são processados por segundo
  - Associado à quantidade de mudanças de corrente suportadas pelos transístores
  - Uma instrução pode demorar um ou mais ciclos de clock
- Tempo de execução de um programa
  - T = #instruções * ciclos_por_instrução / frequência_clock

A FREQUÊNCIA AUMENTA PARA SEMPRE?



ESTRUTURA DE UM PROCESSADOR

- Computação
  - Lê comando da memória (*fetch*)
  - Interpreta o comando (*decode*)
  - Executa o comando (*execute*)
- Localização
  - Lê de uma posição específica (*instruction pointer*)
  - Para um registo (*instruction register*)

ESTRUTURA DE UM PROCESSADOR

- Tipos de instruções
  - Operações aritméticas
  - Cópia de dados
  - Decisão de qual o local da próxima instrução

INSTRUCTION SET ARCHITECTURE

- O que são instruções?
  - Comando específico e bem definido num ISA
  - Podem usar argumentos
  - Vários tipos de instruções
    - Operações lógicas/artiméticas
    - Operações de cópia de dados
    - Operações de salto condicional

```
int sum(int x, int y) {
   int t = x+y;
   return t;
}
```

```
_sum:
   pushl %ebp
   movl %esp,%ebp
   movl 12(%ebp),%eax
   addl 8(%ebp),%eax
   movl %ebp,%esp
   popl %ebp
   ret
```

INSTRUCTION SET ARCHITECTURE

- Operações lógicas/aritméticas
  - Recebem um número de argumentos (2 em IA-32)
  - Operandos podem ser escalares ou estruturados
    - Escalares são guardados em registo
    - Estruturados são guardados em memória

CISC VS RISC

CISC

- Ênfase no hardware
- Instruções complexas de vários clocks
- Operações memória/memória
- Loads/stores incluídos nas instruções
- Transístores usados para guardar instruções complexas

RISC

- Ênfase no software
- Instruções simples e single-clock
- Operações registo/registo
- Loads/stores são instruções independentes
- Transístores usados para guardar registos

CISC VS RISC

CISC

MULT end1, end2

RISC

LOAD A, end1
LOAD B, end2
PROD A, B
STORE end1, A

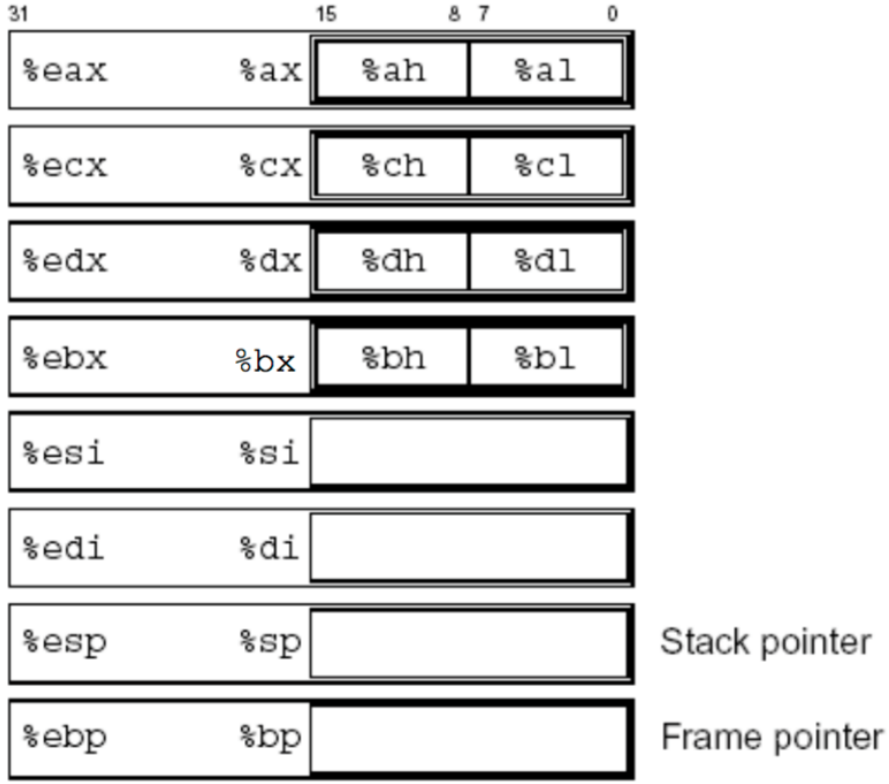
INSTRUCTION SET ARCHITECTURE

- Modos de aceder a operandos em arquitecturas RISC (IA-32)
  - Operandos sempre em registo para operações lógicas/aritméticas
  - Carregar/guardar dados usam 1 ou 2 modos de endereço de memória

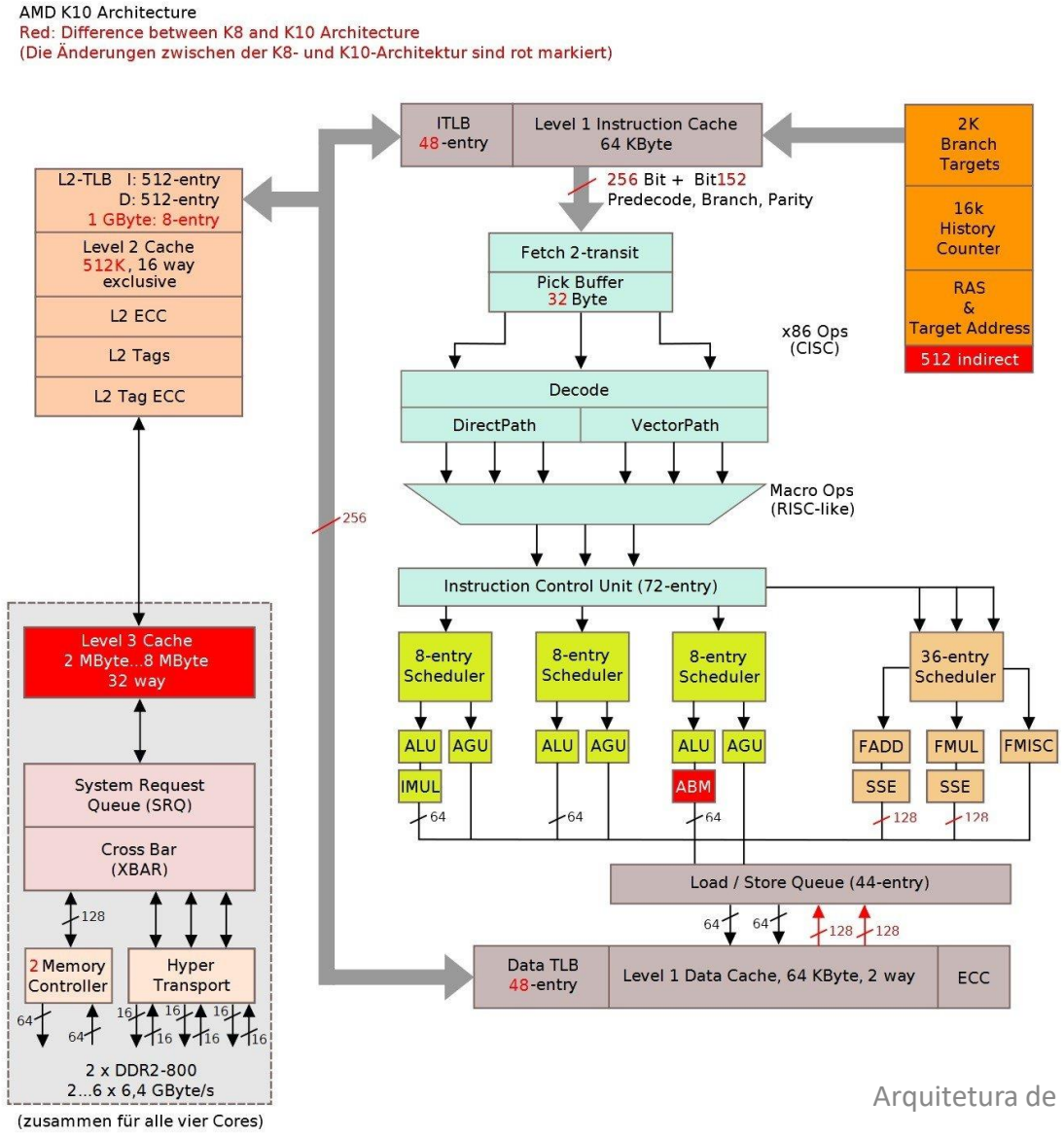
Type	Form	Operand value	Name
Immediate	$\$Imm$	Imm	Immediate
Register	$\mathbf{E}_a$	$\mathrm{R}[\mathbf{E}_a]$	Register
Memory	Imm	$\mathrm{M}[Imm]$	Absolute
Memory	$(\mathbf{E}_a)$	$\mathrm{M}[\mathrm{R}[\mathbf{E}_a]]$	Indirect
Memory	$Imm\,(\mathbf{E}_b)$	$\mathrm{M}[Imm + \mathrm{R}[\mathbf{E}_b]]$	Base + displacement
Memory	$(\mathbf{E}_b, \mathbf{E}_i)$	$\mathrm{M}[\mathrm{R}[\mathbf{E}_b] + \mathrm{R}[\mathbf{E}_i]]$	Indexed
Memory	$Imm\,(\mathbf{E}_b, \mathbf{E}_i)$	$\mathrm{M}[Imm + \mathrm{R}[\mathbf{E}_b] + \mathrm{R}[\mathbf{E}_i]]$	Indexed
Memory	$(, \mathbf{E}_i, s)$	$\mathrm{M}[\mathrm{R}[\mathbf{E}_i] \cdot s]$	Scaled indexed
Memory	$Imm\,(, \mathbf{E}_i, s)$	$\mathrm{M}[Imm + \mathrm{R}[\mathbf{E}_i] \cdot s]$	Scaled Indexed
Memory	$(\mathbf{E}_b, \mathbf{E}_i, s)$	$\mathrm{M}[\mathrm{R}[\mathbf{E}_b] + \mathrm{R}[\mathbf{E}_i] \cdot s]$	Scaled indexed
Memory	$Imm\,(\mathbf{E}_b, \mathbf{E}_i, s)$	$\mathrm{M}[Imm + \mathrm{R}[\mathbf{E}_b] + \mathrm{R}[\mathbf{E}_i] \cdot s]$	Scaled indexed

INSTRUCTION SET ARCHITECTURE

- Organização de registos em RISC
  - 32 registos genéricos
  - No IA-32:

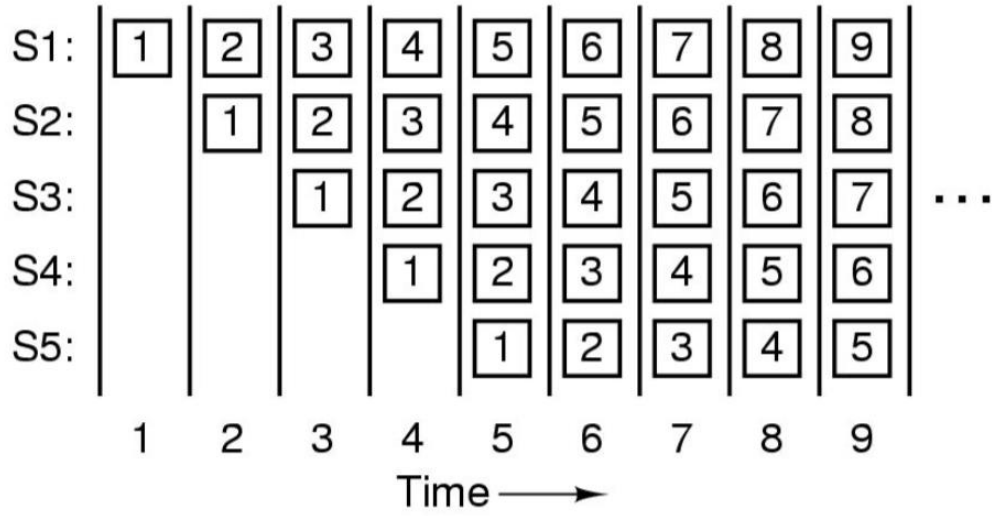


EXECUÇÃO DE INSTRUÇÕES



PIPELINE DE EXECUÇÃO

(a)



HISTÓRIA DOS PROCESSADORES

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+paginada, capaz de correr Unix*			
486	1989	1.9M	
– *integração num único chip: 386, co-proc 387, até 16kB cache L1*			
– *poucas alterações na arquitetura interna do processador*			

AJProença, Sistemas de Computação, UMinho, 2014/15

HISTÓRIA DOS PROCESSADORES

Pentium 1993 3.1M *(= **P5** , aka i586)*
- *arquitectura superescalar, 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 vetores de 64-bits, tipo `int` de 1, 2, ou 4 bytes*

Pentium II 1997 7.5M *(= Pro + MMX)*

Pentium III 1999 8.2M
- *"Streaming SIMD Ext", SSE: vetores 128-bits, `int`/`fp` 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 `fp` de 8-bytes*

Pentium M 2003 77M *(= **P-M**)*
- *arquitetura mais próxima do Pentium III (eficiência energética)*

AJProença, Sistemas de Computação, UMinho, 2014/15

E NOS SMARTPHONES?