



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
6. Evolução da tecnologia e da eficiência



Níveis de abstração:

- nível das linguagens HLL (*High Level Languages*): as linguagens convencionais de programação (puro texto)
 - » imperativas e OO (Basic, Fortran, C/C++, Java, ...)
 - » funcionais (Lisp, Haskell, ...)
 - » lógicas (Prolog, ...)
- nível da linguagem *assembly* (de “montagem”): linguagem intermédia (comandos do CPU em formato texto)
- nível da linguagem máquina: a linguagem de comandos, específica p/ cada CPU ou família de CPU's (em binário puro)
 - » arquiteturas CISC (*Complex Instruction Set Computers*)
 - » arquiteturas RISC (*Reduced Instruction Set Computers*)



```
int x = x+y;
```

- Código C
 - somar 2 inteiros (c/ sinal)

```
addl 8(%ebp), %eax
```

Idêntico à expressão

```
x = x + y
```

- *Assembly* (da GNU p/ IA-32)
 - somar 2 inteiros de 4 *bytes*
 - operandos “*long*” em GCC
 - a mesma instrução, c/ ou s/ sinal
 - operandos:
 - x: em registo %eax
 - y: na memória M[%ebp+8]

```
0x401046: 03 45 08
```

- Código *object* em IA-32
 - instrução com 3 *bytes*
 - na memória a partir do endereço 0x401046

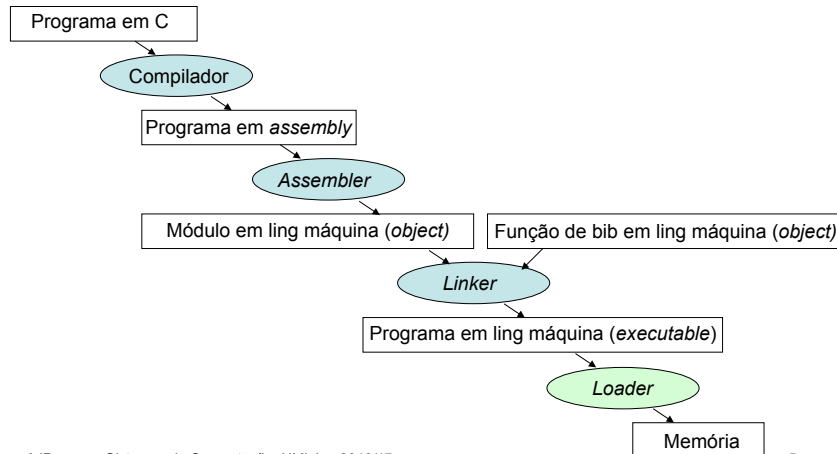


Mecanismos de conversão (para comandos do CPU):

- compilador
 - traduz um programa de um nível de abstração para outro inferior (converte um ficheiro de texto noutra de texto); por ex., de C para *assembly*
 - normalmente inclui mais que um passo de conversão, até chegar à linguagem máquina
- *assembler* (“montador”)
 - “monta” os comandos / instruções em texto, para binário (*object*), de acordo com as regras do fabricante do CPU
- interpretador
 - analisa, uma a uma, as instruções de um programa em HLL, e:
 - » gera código em linguagem máquina para essa instrução, e
 - » executa esse código (nota: não guarda o código gerado)

Execução de programas num computador (4)

De um programa em HLL até à sua execução:



Execução de instruções (em linguagem máquina) num CPU

Ciclo de execução de instruções:

- Leitura de uma instrução da memória
... e incremento do IP
- Descodificação da instrução
- Execução da operação
 - cálculo da localização do(s) operando(s),
e ir buscá-lo(s), se necessário
 - execução da ação especificada
 - guardar resultado, se necessário

Modelo de computação de von Neumann (1945)

Análise de um exemplo: `movl Mem_Loc, %eax`

Modelo de computação de von Neumann, 1945/46 (1)

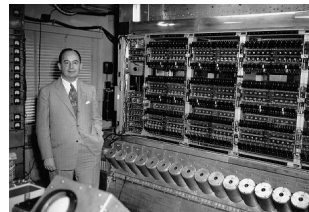
ENIAC (1ª geração, 1945)

- objetivo: cálculo tabelas de artilharia (mas 1º teste foi p/ bomba H)
- máquina **decimal** (base 10)
- 17.468 válvulas, 27 toneladas
- programação: manual, alterando as conexões (cablagem)

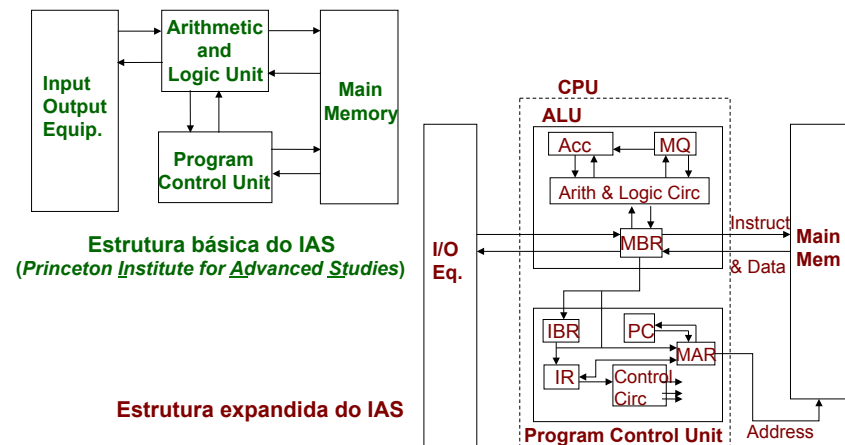


Von Neumann introduz conceito de **stored-program**:

- dados e instruções em **binário**, e armazenados numa memória
- memória acedida pelo endereço da informação
- execução de instruções de modo sequencial (daí o *Program Counter*, PC), interpretadas pela unid. controlo
- constrói novo computador, IAS



Modelo de computação de von Neumann, 1945/46 (2)



IBM, Mark II
(09-Set-1947):

- entrada que os operadores de serviço fizeram no logbook...

Handwritten logbook entry (IBM Mark II, 09-Set-1947):

9/9

0800 Anton started
1000 stopped - anton ✓

13:00 (032) HP - MC ~~2.13047695~~ 2.13047695 (2) 4.61592505(-2)
033 PRO 2 2.13047695
convd 2.13047695

Relays 6-2 in 033 failed speed test
in relay 10,00 test.

Relays changed

1100 Started Cosine Taps (Sine check)
1525 Started Multi Adder Test.

1545

Relay #70 Panel F (Moth) in relay.

First actual case of bug being found.

1700/1800 and signal stacks.
1700 closed down.

9

O diagrama ilustra a arquitetura de um sistema de computadores, destacando a conexão entre o processador, a memória e o chipset.

- CPU / Núcleo dum processador:** Contém o **banco de registos** (registro bank) e a **ALU** (Arithmetic Logic Unit).
- Memória primária:** Representada por um bloco com linhas horizontais, conectada ao processador via **memory channel**.
- Chipset:** Um bloco central que atua como um hub de comunicação. Ele contém:
 - interface mem - I/O:** Conecta o processador ao chipset.
 - controlador USB:** Gerencia a comunicação com dispositivos USB (como **rato** e **teclado**).
 - interface PCI-Express:** Gerencia a comunicação com dispositivos de alta velocidade (como **gráfica/rede/...**).
 - controlador arquivo:** Gerencia a comunicação com dispositivos de armazenamento (como **SSD/HDD**).
- I/O bus:** Uma linha de comunicação horizontal que conecta o chipset aos dispositivos de entrada/saída.
- Expansão para outros componentes:** Indica a possibilidade de adicionar mais dispositivos ao sistema através do I/O bus.

10

Ex.: `movl Mem Loc,%eax`

O diagrama ilustra a interação entre a CPU e a Memória P. A CPU, representada por um retângulo tracejado, contém um banco de registros, uma ALU e uma interface de memória. A Memória P, representada por um retângulo sólido, contém uma tabela de endereços com valores como "movl ..." e "var x". Uma seta vermelha indica o fluxo de dados da Memória P para a CPU, rotulado com "(IP)" e "RD".

11

```
EX: movl Mem Loc,%eax
```

Diagram illustrating the memory access phase of instruction execution:

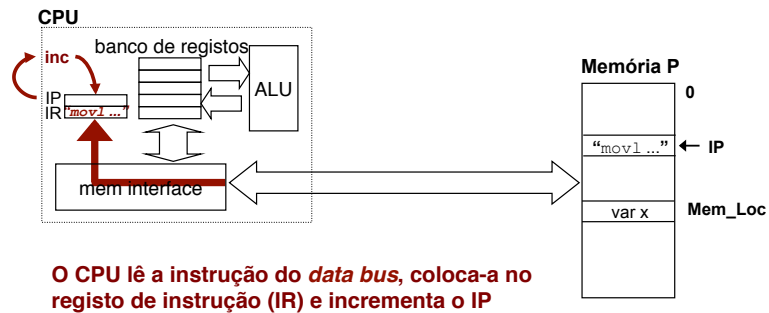
- CPU Side:**
 - banco de registros** (Register Bank): Connected to the **ALU** and the **mem interface**.
 - ALU** (Arithmetic Logic Unit): Receives data from the register bank and performs operations.
 - mem interface** (Memory Interface): Connected to the **Memória P** and the **mem interface**.
- Memória P (Physical Memory):**
 - Address **0**: Contains the instruction **"movl ..."**.
 - Address **IP** (Instruction Pointer): Points to the instruction **"movl ..."**.
 - Address **var x**: Contains the variable **x**.
 - Mem_Loc** (Memory Location): Label for the memory stack.
- Instruction Flow:** A green arrow points from the **Memória P** to the **mem interface**, labeled **"movl ..."** and **IP**, indicating the instruction being fetched.

12

Exemplo de execução de uma
instrução em linguagem máquina (3)

Ex.: `movl Mem_Loc,%eax`

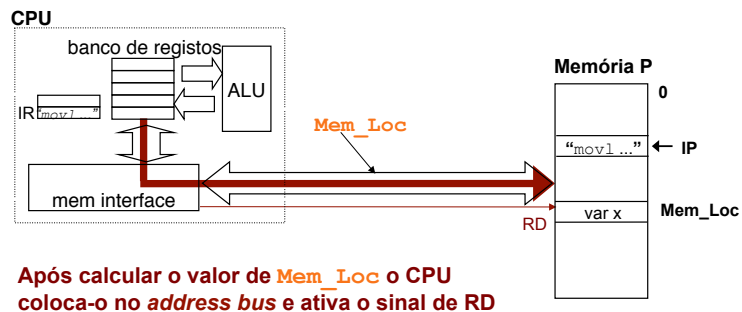
1. Leitura da instrução na memória (3)
... e incremento do IP



Exemplo de execução de uma
instrução em linguagem máquina (5)

Ex.: `movl Mem_Loc,%eax`

3. Execução da operação (1)

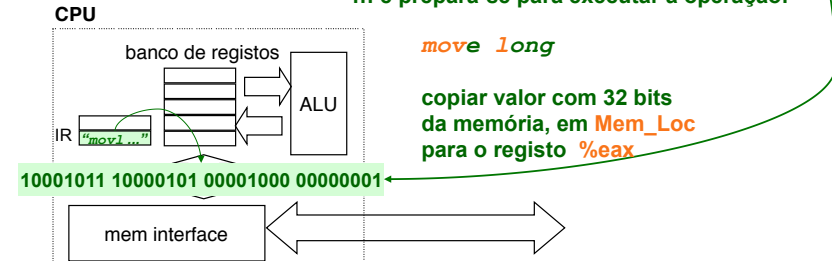


Exemplo de execução de uma
instrução em linguagem máquina (4)

Ex.: `movl Mem_Loc,%eax`

2. Descodificação da instrução

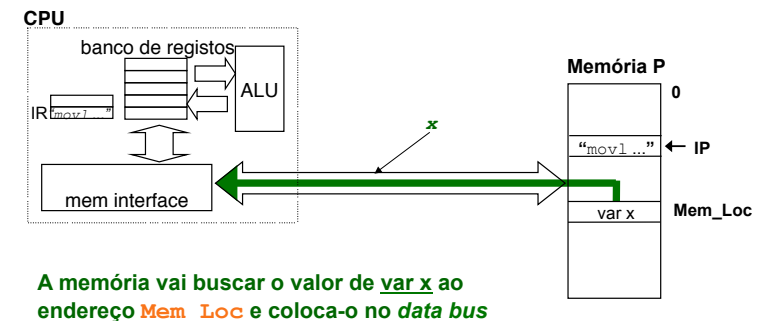
A unidade de controlo do CPU descodifica a instrução...
... e prepara-se para executar a operação:



Exemplo de execução de uma
instrução em linguagem máquina (6)

Ex.: `movl Mem_Loc,%eax`

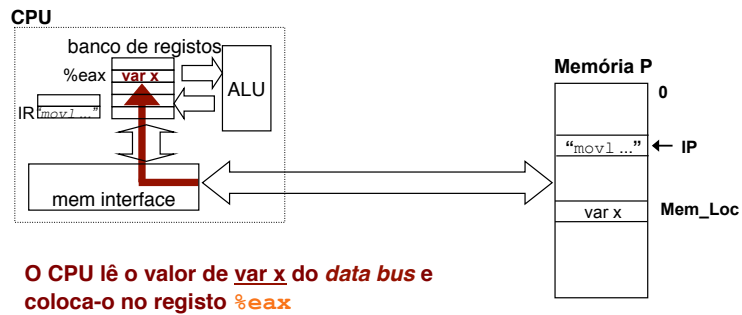
3. Execução da operação (2)



Exemplo de execução de uma
instrução em linguagem máquina (7)

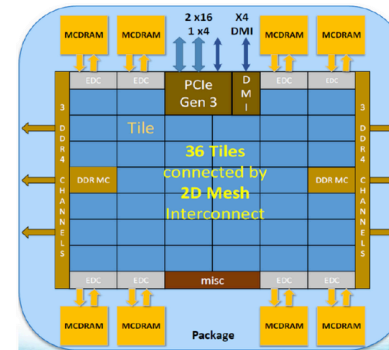
Ex.: `movl Mem_Loc,%eax`

3. Execução da operação (3)



Intel Many Integrated Core (MIC):
the Xeon Phi KNL (up to 72 cores)

KNL Overview



TILE	2 VPU	CHA	2 VPU
	Core	1MB L2	Core

