

## 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*)

## Execução de programas num computador (2)

<pre>int x = x+y;</pre>
<pre>addl 8(%ebp),%eax</pre>
<p>Idêntico à expressão</p> <p><math>x = x + y</math></p>
<pre>0x401046: 03 45 08</pre>

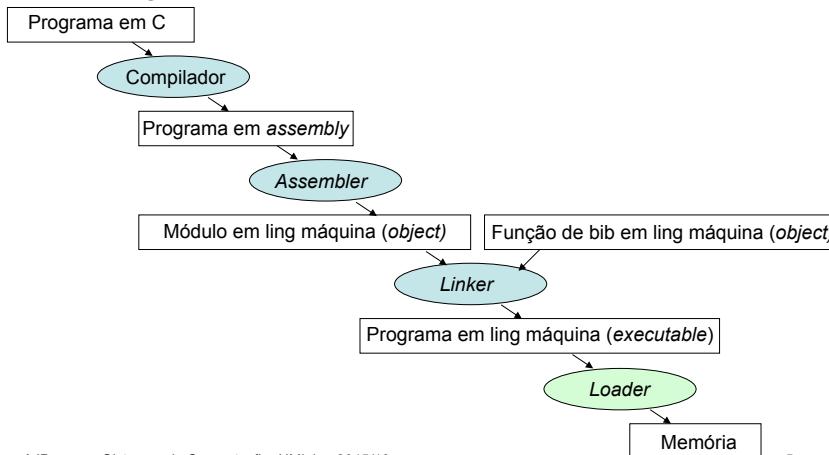
## Execução de programas num computador (3)

<h3>Mecanismos de conversão</h3> <p>(para comandos do CPU):</p> <ul style="list-style-type: none"><li>– compilador<ul style="list-style-type: none"><li>• <u>traduz</u> um programa de um nível de abstração para outro inferior (converte um ficheiro de texto noutro de texto); por ex., de C para <b>assembly</b></li><li>• normalmente inclui mais que um passo de conversão, até chegar à linguagem máquina</li></ul></li><li>– <b>assembler</b> (“montador”)<ul style="list-style-type: none"><li>• “<u>monta</u>” os comandos / instruções em texto, para binário (<i>object</i>), de acordo com as regras do fabricante do CPU</li></ul></li><li>– interpretador<ul style="list-style-type: none"><li>• <u>analisa</u>, uma a uma, as instruções de um programa em HLL, e:<ul style="list-style-type: none"><li>» <u>gera código</u> em linguagem máquina para essa instrução, e</li><li>» <u>executa</u> esse código (<u>nota</u>: não guarda o código gerado)</li></ul></li></ul></li></ul>
---

## Execução de programas num computador (4)



### De um programa em HLL até à sua execução:



5

## 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`

AJProenca, Sistemas de Computação, UMinho, 2015/16

6

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



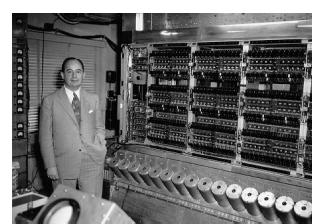
### ENIAC (1ª geração, 1945)

- 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)

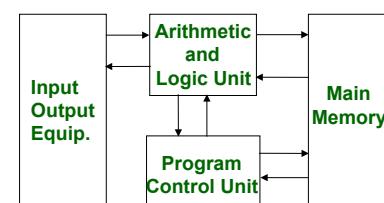


### Von Neumann introduz conceito de stored-program :

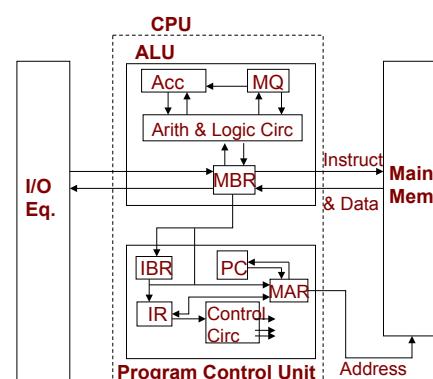
- dados e instruções em binário, e armazenados numa memória
- memória acedida pelo endereço da informação
- execução de instruções de modo sequencial (dai o *Program Counter*, PC), interpretadas pela unid. controlo
- constrói novo computador, IAS



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



Estrutura expandida do IAS



AJProenca, Sistemas de Computação, UMinho, 2015/16

7

AJProenca, Sistemas de Computação, UMinho, 2015/16

8

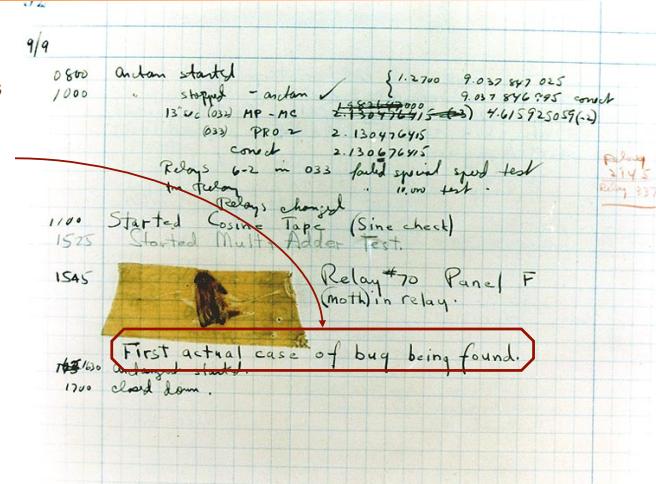
## Relato do primeiro bug num computador

## Modelo de arquitetura de um computador elementar



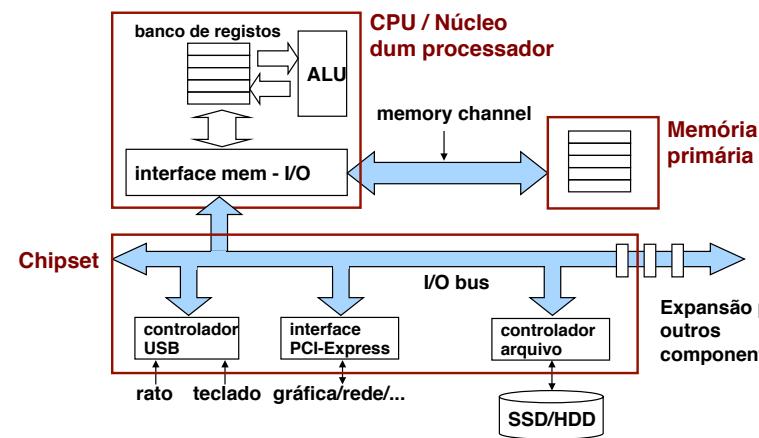
**IBM, Mark II**

(09-Set-1947):  
- entrada que os operadores de serviço fizeram no logbook...



AJProenca, Sistemas de Computação, UMinho, 2015/16

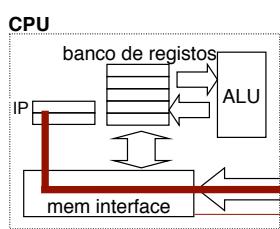
9



10

## 1. Leitura da instrução na memória (1)

Ex.: `movl Mem_Loc,%eax`



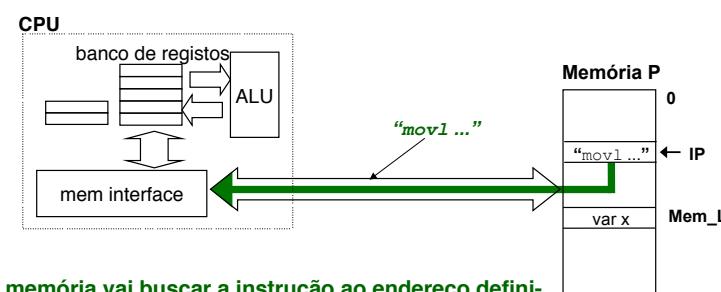
O CPU coloca no **address bus** o valor em IP (endereço p/ próxima instrução), e ativa o sinal de controlo RD

AJProenca, Sistemas de Computação, UMinho, 2015/16

11

## 1. Leitura da instrução na memória (2)

Ex.: `movl Mem_Loc,%eax`



A memória vai buscar a instrução ao endereço definido por IP e coloca-a no **data bus** p/ ser lida pelo CPU

AJProenca, Sistemas de Computação, UMinho, 2015/16

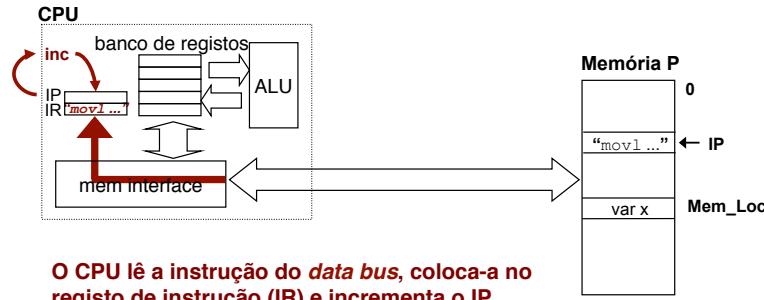
12

### 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

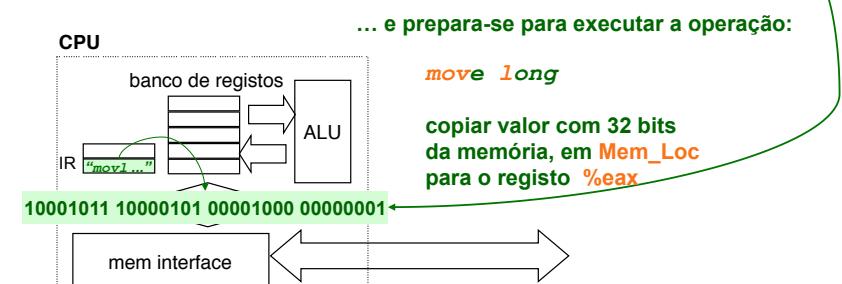


### Exemplo de execução de uma instrução em linguagem máquina (4)

Ex.: `movl Mem_Loc, %eax`

#### 2. Descodificação da instrução

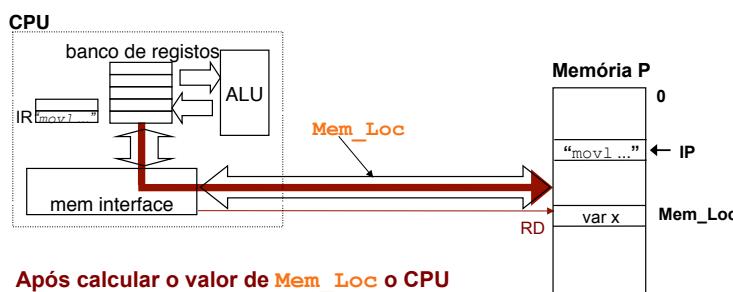
A unidade de controlo do CPU descodifica a instrução...



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

Ex.: `movl Mem_Loc, %eax`

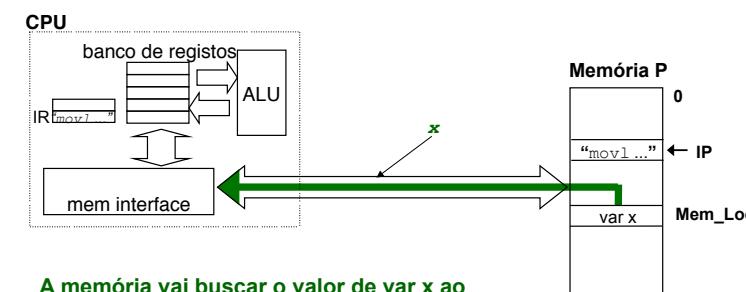
#### 3. Execução da operação (1)



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

Ex.: `movl Mem_Loc, %eax`

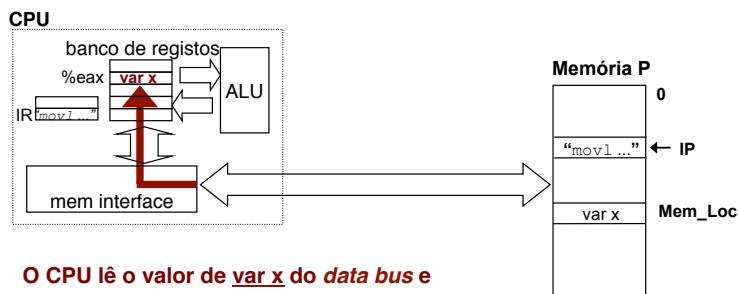
#### 3. Execução da operação (2)



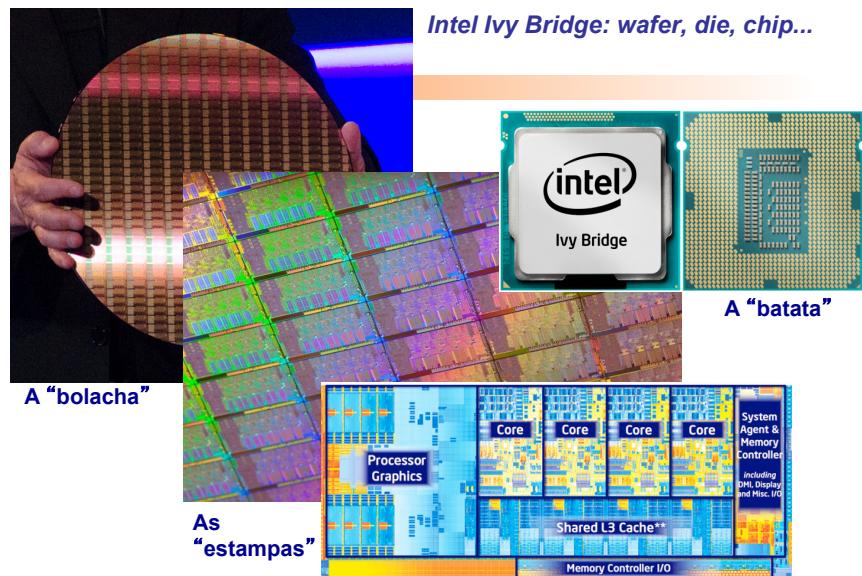
## Exemplo de execução de uma instrução em linguagem máquina (7)

Ex.: `movl Mem_Loc, %eax`

### 3. Execução da operação (3)



O CPU lê o valor de `var x` do *data bus* e coloca-o no registo `%eax`



**Intel Many Integrated Core (MIC):  
the Xeon Phi with 60 cores**

