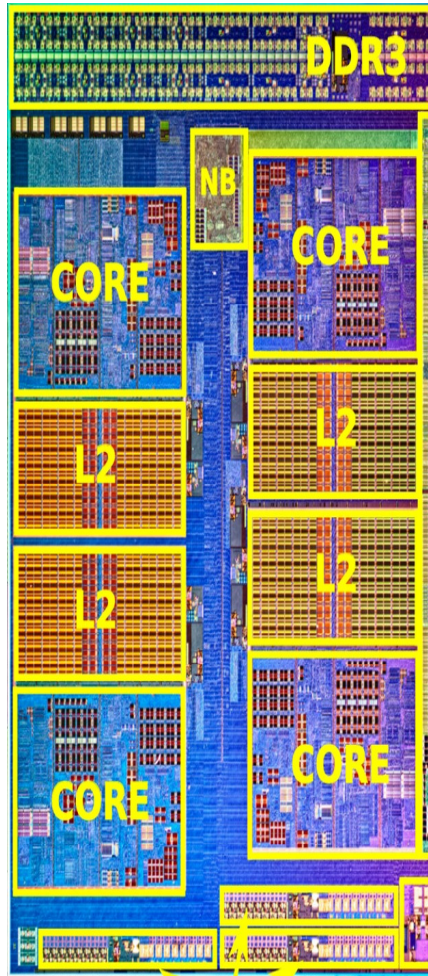


0273359 - Arquitetura e Organização de Computadores 1



Fonte: <http://www.techspot.com/article/904-history-of-the-personal-computer-part-5/>

Tools

Luciano de Oliveira Neris

luciano@dc.ufscar.br

Adaptado de slides do prof. Marcio Merino Fernandes
Figuras: David Patterson, John Hennessy
Arquitetura e Organização de Computadores – 4Ed, Elsevier, 2014

Departamento de Computação
Universidade Federal de São Carlos



Simulador MARS

- Simulador p/ instruções MIPS: MARS V4.5
 - <http://courses.missouristate.edu/KenVollmar/MARS/>
 - Arquivo zip contendo:
 - Simulador (precisa de Java p/ rodar)
 - Tutorial
 - Programas Exemplo
- *OBS: Similar ao simulador SPIM / QtSpim*

Diretivas do MIPS Assembler

- # Área de Dados do programa
- **.data**
- # Área de Código (instruções) do programa
- **.text**
- #Área de rótulos (labels) visíveis
- #globalmente (ou seja, outros módulos do programa).
- **.globl**

Programa Exemplo 1

```
##### Pgm p/ ler e somar 3 nros inteiros:
```

```
.data
prompt:      .asciiz      "Please enter three numbers: \n"
sum_msg:     .asciiz      "The sum is: "

.text
.globl main
main:
    la      $a0,prompt          # display prompt string
    li      $v0,4
    syscall

    li      $v0,5                # read 1st integer into $t0
    syscall
    move    $t0,$v0

    li      $v0,5                # read 2nd integer into $t1
    syscall
    move    $t1,$v0

    li      $v0,5                # read 3rd integer into $t2
    syscall
    move    $t2,$v0
```

Continua →

Programa Exemplo 1

```
addu  $t0,$t0,$t1          # accumulate the sum into $t0
addu  $t0,$t0,$t2

la    $a0,sum_msg          # write sum message
li    $v0,4
syscall

move  $a0,$t0              # output sum
li    $v0,1
syscall
```

Outras Ferramentas

- ProcSim MIPS v2.0

- ▣ <http://jamesgart.com/procsim/>

- DrMIPS

- ▣ <http://brunonova.github.io/drmips/>

- Hades

- ▣ <https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/download.html>