

# Arquitectura de Computadores 2

Aula 1 – Introdução

Pedro Miguel Lavrador

## Avaliação

- $NF = 0,5 * NT + 0,5 * NP$
- Nota Mínima 8,0
  - $NP = 30\% * TP1 + 50\% * TP2 + 20\% * AC$
- Data dos testes:
  - Teste Prático 1                      3 e 8 de Abril
  - Teste Prático 2                      22, 27 e 26 de Maio  
   3 de Junho

## Avaliação TE

- Os alunos com o estatuto de trabalhador-estudante deverão, obrigatoriamente, estar inscritos numa turma prática.
- Os alunos realizam os testes práticos na sua turma nas datas anunciadas.
- No caso dos alunos TE, que não frequentem pelo menos 80% das aulas, a nota final da componente prática será obtida do seguinte modo:
  - $NP = (TP1 \cdot 0.3 + TP2 \cdot 0.50) / 0.80$

11-02-2014

PML - AC2 - 2014

3

## Horas de Contacto

- Atendimento para todas as turmas práticas
  - 4<sup>a</sup>s-feiras, das 15:00 às 16:00, no IT
- Não haverá atendimento na véspera dos testes práticos.

11-02-2014

PML - AC2 - 2014

4

## Material a usar nas aulas práticas

- No início do semestre será atribuído a cada grupo, mediante a entrega de uma caução, um conjunto de material que inclui:
  - uma placa de desenvolvimento DETPIC32,
  - um cabo USB
  - componentes electrónicos diversos.
- Todo o material deverá ser devolvido no final do semestre, sendo a caução restituída se o mesmo se encontrar nas condições em que foi entregue.

11-02-2014

PML - AC2 - 2014

5

## Requisição de Material

- 1) O valor da caução é 40€.
- 2) A devolução da caução é feita, após a devolução da placa, por transferência bancária, pelos serviços administrativos centrais da UA.
- 3) Com a caução os alunos entregam uma requisição com os dados pessoais, que está disponível na portaria do DETI.
- 4) De entre os dados que têm que preencher está o NIB da sua conta bancária. É obrigatória a entrega do talão multibanco com o NIB.
- 5) No momento do pagamento é entregue aos alunos um recibo provisório do pagamento. Os recibos definitivos são posteriormente emitidos. Será publicada no site a data a partir da qual os alunos podem levantar os recibos definitivos.
- 6) Após o pagamento, os alunos levantam a placa DETPIC32 no armazém de componentes.
- 7) No final do semestre, a devolução da placa é também feita no armazém de componentes. Após a entrega da placa inicia-se o processo para devolução da caução.
- A data-limite para a devolução da placa é, tal como está no guião, a data do exame de recurso. A entrega tardia das placas tem como consequência provável a devolução das cauções apenas em Setembro...

11-02-2014

PML - AC2 - 2014

6

## Regime de Faltas

- Limite máximo de 3 Faltas
- A justificação de faltas deve ser entregue na secretaria do DETI.
- **Só serão consideradas as justificações que dêem entrada na secretaria até 5 dias úteis após o fim do período que, justificadamente, deu origem à falta. Justificações entregues fora deste prazo serão automaticamente descartadas.**

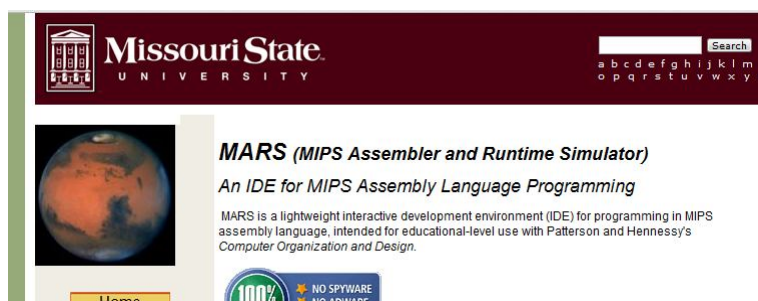
11-02-2014

PML - AC2 - 2014

7

## Aula 1

- Em AC1 usámos o Assembly do MIPS...
- ... No ambiente de simulação MARS



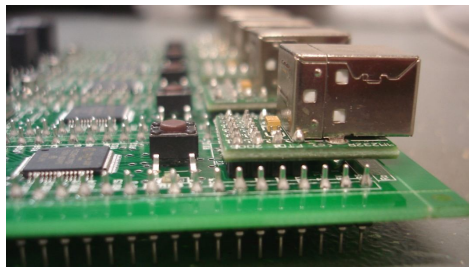
11-02-2014

PML - AC2 - 2014

8

## Aula 1

- Agora, o ambiente de simulação tornou-se real... ganhou portos de entrada e saída, e alguns periféricos...

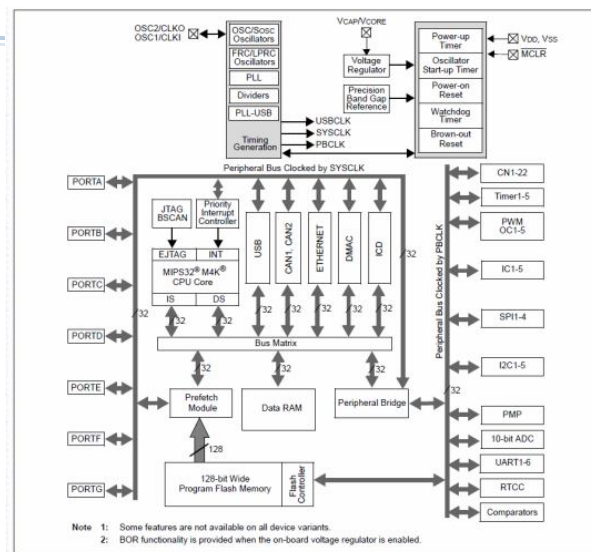


11-02-2014

PML - AC2 - 2014

9

## Aula 1 – O PIC32



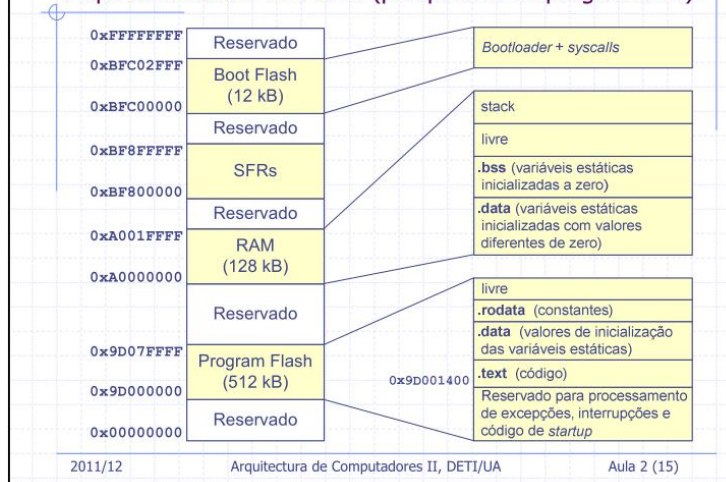
11-02-2014

PML - AC2 - 2014

10

## Aula 1 – Memória do PIC32

Mapa de memória do PIC32 (perspectiva do programador)



11-02-2014

PML - AC2 - 2014

11

## Aula 1 - Objetivos

- Conhecer o processo de criação de um programa escrito em *assembly* para correr na placa DETPIC32: compilação, transferência e execução.
- Utilizar os *system calls* disponibilizados na placa DETPIC32.
- Rever os conceitos associados à manipulação de *arrays* de caracteres.

11-02-2014

PML - AC2 - 2014

12

## Aula 1 – Ferramentas

- Gvim Editor do ficheiro .s
- pcompile Compilador de Assembly (ou de C) para .hex
- hex2asm permite ver o código máquina que foi criado
- ldpic32 transfere o código máquina para a placa
- pterm monitor da porta série

11-02-2014

PML - AC2 - 2014

13

## System Calls

Tabela IV: *System Calls* do DETPIC32

Protótipo equivalent em C	\$v0	Parâmetros de entrada	Retorno
<b>char</b> inkey( <b>void</b> )	1		\$v0
<b>char</b> getChar( <b>void</b> )	2		\$v0
<b>void</b> putChar( <b>char</b> ch)	3	\$a0 = character	
<b>unsigned int</b> readInt( <b>unsigned int</b> base)	4	\$a0 = base	\$v0
<b>int</b> readInt10( <b>void</b> )	5		\$v0
<b>void</b> printInt( <b>unsigned int</b> val, <b>unsigned int</b> base)	6	\$a0 = val, \$a1 = base	
<b>void</b> printInt10( <b>int</b> val)	7	\$a0	
<b>void</b> printStr( <b>char</b> *str)	8	\$a0 = str	
<b>void</b> readStr( <b>char</b> *buffer, <b>unsigned int</b> nc)	9	\$a0 = buffer, \$a1 = nc	
<b>void</b> exit( <b>int</b> code)	10	\$a0 = exit code	
<b>unsigned int</b> readCoreTimer( <b>void</b> )	11		\$v0
<b>void</b> resetCoreTimer( <b>void</b> )	12		

printInt(), "base": 16 lsbits – [2.. 16] / 16 msbits – número de caracteres com que o resultado é apresentado (o valor por omissão é 0, i.e. sem formatação)

11-02-2014

PML - AC2 - 2014

14

## Arquitectura de Computadores 2

### Breve Resumo da Convenção de Uso de Registos do MIPS

## Convenção de Uso de Registos no MIPS

- $\$s0..\$s7$  Geral (**não** podem ser alterados pelas funções)
- $\$t0..\$t9$  Geral (podem ser alterados pelas funções)
- $\$a0..\$a3$  Passagem de argumentos para funções
- $\$v0..\$v1$  Retorno de funções
- $\$sp$  Stack Pointer
- $\$ra$  Endereço de retorno das funções



## Codificação de Subrotinas: SR Chamada

---

- No início:
  - Salva na *stack* os registos **\$s0 a \$s7** que pretende utilizar.
  - Salva o registo **\$ra** no caso de a rotina também ser chamadora.
- No fim:
  - Coloca o valor de retorno em **\$v0** (excepto se for tipo **void**).
  - Restaura os registos **\$s0 a \$s7** que salvaguardou no início.
  - Restaura o registo **\$ra** (no caso de ter sido salvaguardado no início).
  - Retorna, executando a instrução **"jr \$ra"**.

11-02-2014

PML - AC2 - 2014

17

## Codificação de Subrotinas: SR Chamadora

---

- Antes de chamar:
  - Passa os parâmetros; os 4 primeiros são passados nos registos **\$a0...\$a3** e os restantes na *stack*.
  - Executa a instrução **"jal"**.
- Após regresso:
  - Usa o valor de retorno que está em **\$v0**.

11-02-2014

PML - AC2 - 2014

18

## Codificação de subrotinas

---

- A rotina chamadora **não pode assumir** em caso algum que qualquer dos **registos temporários** (\$a0...\$a3, \$t0...\$t9, \$v0 e \$v1) **são preservados** pela rotina chamada.
- A codificação da rotina "main" **está sujeita às mesmas regras** que se aplicam às restantes rotinas.