

Departamento de Electrónica, Telecomunicações e Informática

Introdução à Arquitetura de Computadores

LEI, 1º Ano, 2º semestre

Dossiê Pedagógico

Ano letivo 2017/18
(Fevereiro/2018)

Índice

1. Enquadramento da disciplina na Licenciatura em Engenharia Informática do DETI	2
2. Objectivos da disciplina	2
3. Conhecimentos prévios	2
4. Bibliografia Básica	2
5. Programa	3
I – Introdução: sistemas de computação de uso geral. Microprocessadores.....	3
II – Representação da informação e operações básicas	3
III – Circuitos Lógicos	3
IV – O Modelo de Von Neumann	3
V – A arquitetura MIPS.....	3
VI – A linguagem assembly e o <i>assembler</i>	4
VII – Organização interna do processador	4
VIII – Comunicação com o exterior: entrada e saída de dados.....	4
6. Avaliação	5
Regras gerais	5
Avaliação na época de recurso.....	5
Trabalhadores estudantes	5
7. Calendário previsto das aulas teóricas.....	6
8. Programa da componente laboratorial.....	9
9. Regras gerais de funcionamento da disciplina	10
Regime de faltas.....	10
E-mails	10
Esclarecimento de dúvidas. Funcionamento das OT	10
Ilícitos	10
10. Equipa docente.....	10

1. Enquadramento da disciplina na Licenciatura em Engenharia Informática do DETI

A disciplina de Introdução à Arquitetura de Computadores é uma disciplina da Licenciatura em Engenharia Informática (LEI). A disciplina é lecionada no 2º semestre do 1º ano, com uma escolaridade semanal de 2h de aulas teórico-práticas (TP) e 2h de aulas práticas (P). O volume de trabalho exigido pela disciplina é de 6 ECTS, correspondendo a 162 (6x27h) horas de trabalho no total do semestre.

Considerando que 10% do esforço será dedicado à preparação para exame, e atendendo a que o semestre tem 15 semanas de aulas, resulta que o número médio de horas por semana dedicados à disciplina é de 11h. Assim, para além das 4h de aulas, espera-se que os alunos dediquem semanalmente 7h horas de estudo adicional à disciplina.

A disciplina de *Introdução à Arquitetura de Computadores* (IAC) é a primeira do elenco de disciplinas obrigatórias da área científica de Arquitectura dos Sistemas Computacionais. A organização dos computadores digitais numa perspetiva funcional, através da descrição do reportório de instruções e da programação em *assembly*, e estrutural (descrição da estrutura interna básica dos processadores), é do que se ocupa IAC. Os conhecimentos aqui adquiridos fornecem a base necessária para o estudo de Sistemas Operativos no semestre seguinte.

2. Objetivos da disciplina

- Conhecer as formas de representação da informação nos computadores digitais, com relevo para a representação da informação numérica e as operações aritméticas básicas.
- Conhecer as operações lógicas e as componentes eletrónicas que as realizam.
- Compreender o funcionamento dos sistemas com memória e o funcionamento dos principais dispositivos de armazenamento de informação.
- Compreender a organização interna dos computadores digitais.
- Compreender os mecanismos de comunicação do computador com o exterior.
- Adquirir familiaridade com a arquitectura de processadores através da programação em *assembly*.

3. Conhecimentos prévios

IAC pressupõe que os alunos nela inscritos têm já um conhecimento dos princípios básicos de Programação adquiridos na disciplina Fundamentos de Programação do 1º semestre e consolidados em Programação Orientada a Objetos que decorre em simultâneo com IAC.

4. Bibliografia Básica

- D.M.Harris, S.L.Harris, *Digital Design and Computer Architecture*, 2nd. Edition, Morgan Kaufmann, 2013.
- D.A.Patterson, J.Hennessy, *Computer Organization and Design – the hardware/software interface*, Elsevier, 2009.
- Y.N. Patt, S.J.Patel, *Introduction to Computing Systems – from bits & gates to C & beyond*, 2nd edition, McGraw_Hill Education, Indian Edition. 2005.

5. Programa

I – Introdução: sistemas de computação de uso geral. Microprocessadores.

II – Representação da informação e operações básicas

- Representação da informação: o bit como unidade de informação. Tipos de dados inteiros – inteiros positivos e inteiros com sinal; representações em sinal e módulo, complemento para um e complemento para dois: gama de representação. Conversão decimal-binário e binário-decimal. Notação hexadecimal.
- Aritmética binária: adição e subtração.
- Representação de informação não numérica. O código ASCII.
- Representação em vírgula flutuante. Breve referência ao Standard IEEE 754.
- Operações lógicas: AND, OR, NOT, Exclusive OR.
- Álgebra de Boole. Axiomas e teoremas. Formas canónicas. Simplificação de funções booleanas.

III – Circuitos Lógicos

- Portas lógicas: NOT, OR, NOR, AND, NAND.
- Blocos combinatórios básicos: decodificadores, multiplexers, *full-adder*.
- Dispositivos básicos de armazenamento de informação: R-S *latch*, Flip-Flop D. Registos.
- Memória: espaço de endereçamento. Células de memória: tecnologias SRAM e DRAM.

IV – O Modelo de VonNeumann

- As componentes básicas: Memória, Unidade de Processamento Central (CPU), Entrada e Saída de dados, Unidade de Controlo.
- Estrutura de um processador MIPS. Ciclo básico de processamento das instruções.
- Organização da memória central dos sistemas de computação. Ciclos básicos de acesso à memória.
- Traços gerais da evolução da arquitetura dos sistemas de computação e das tecnologias de implementação.
- Os diferentes níveis de utilização dos sistemas de computação: aplicação, linguagem de programação de alto nível, *assembly*, serviços do sistema de operação.
- Função e estrutura. Arquitetura como visão funcional básica comum a uma "família" de processadores. Modelo de programação e repertório de instruções.

V – A arquitetura MIPS

- Modelo de programação. Repertório de instruções; as principais classes de instruções: processamento (aritméticas e lógicas), transferência da informação e controlo do fluxo de execução.
- Instruções aritméticas e lógicas da arquitetura MIPS. Códigos de instrução.
- Instruções de acesso à memória e modos de endereçamento.
- Tradução das estruturas de decisão *if* e *if_then_else* para *assembly*: instruções de salto condicional e incondicional.
- Tradução das estruturas de controlo de fluxo: *for*, *while* e *do...while*. Acesso sequencial a elementos de *arrays*.
- Procedimentos de invocação e retorno. Passagem de parâmetros. Utilização da stack. Convenções de salvaguarda do conteúdo dos registos.

VI – A linguagem assembly e o assembler

- A linguagem assembly.
 - Instruções
 - Diretivas
 - Exemplos
- O processo de tradução dos programas *assembly* em binário executável.

VII – Organização interna do processador

- Unidades operativas e unidade de controlo; construção de um *datapath* genérico para uma arquitectura tipo MIPS;
- Análise dos blocos constituintes necessários à execução de cada tipo de instruções básicas: Tipo R; *load* e *store*; Salto condicional; Montagem de um *datapath* completo para execução de instruções no modo *single-cycle*.
- Desenho da unidade de controlo da ALU; A unidade de controlo principal; Exemplos de funcionamento do *datapath* incluindo a unidade de controlo;
- Limitações das arquiteturas *single-cycle*; Versão de referência de uma arquitetura *multicycle*; Exemplos do processamento das instruções numa arquitetura *multicycle*;
- Unidade de controlo para *datapath multicycle*; Diagrama de estados da unidade de controlo.
- Sinais de controlo e valores do *datapath multicycle*; Exemplo com execução sequencial de algumas instruções.

VIII – Comunicação com o exterior: entrada e saída de dados

- Entradas e saídas mapeadas em memória *versus* instruções específicas para entrada e saída de dados.
- O sistema de interrupções. Programação com interrupções. Entradas/Saídas com interrupção.

6. Avaliação

Regras gerais

A nota final obtém-se da média ponderada das componentes de avaliação prática e teórica com os pesos de 40% e 60%, respetivamente.

A avaliação teórica consta de dois momentos de avaliação:

- Um teste escrito presencial (T1) com um peso de 45% na nota final, a realizar na aula teórica de 11.4.2018
- Um teste escrito presencial (T2) com um peso de 55% na nota final a realizar na época de exames.

$$NT = \text{Nota_T1} \times 0.45 + \text{Nota_T2} \times 0.55$$

De acordo com o ponto 2 do artigo 30º do Regulamento de estudos da UA, publicado no Diário da República, 2ª série, em 8 de setembro de 2016, os alunos que desejem submeter-se a exame final à componente teórica devem manifestar essa intenção ao docente responsável até final da segunda semana de aulas.

A avaliação prática consta de três componentes:

- Dois testes escritos presenciais a realizar nas aulas práticas dos dias 22 e 23 de março e 7 e 8 de junho com os pesos de 40% e 50% respetivamente.
- Uma componente de Avaliação Continua relativa à preparação e participação nas aulas com um peso de 10%.

$$NP = \text{Nota_TP1} \times 0.40 + \text{Nota_TP2} \times 0.50 + \text{Nota_AC} \times 0.10$$

- **Nota mínima**

Em cada uma das duas componentes de avaliação é necessário obter pelo menos 7,0 valores para obter aprovação à disciplina.

Avaliação na época de recurso

O exame da componente teórica na época de recurso incide sobre toda a matéria lecionada no âmbito da disciplina constituindo a classificação nele obtida a nota final dessa componente.

A nota obtida na componente prática é mantida para a época de recurso sempre que seja igual ou superior à nota mínima exigida (7,0). Sempre que houver lugar à realização de um exame prático de recurso a nota prática anterior é anulada. A nota obtida na componente prática na época de recurso não é mantida para o ano letivo seguinte.

Trabalhadores estudantes

Os alunos com o estatuto de trabalhador-estudante que pretendam usufruir do mesmo modelo de avaliação dos estudantes em regime ordinário deverão participar em, pelo menos, 80% das aulas práticas.

7. Calendário previsto das aulas teóricas

Aula	Data	Tema
1	14/Fev	I - Introdução: Apresentação dos objetivos e programa da disciplina e do seu enquadramento curricular. Bibliografia. Avaliação. II - Representação da informação nos computadores. O sistema binário. Representação de inteiros.
2	16/Fev	Representação em sinal e módulo e em complemento para 2.
3	21/Fev	Adição e subtração binárias. Conversão de decimal para binário e de binário para decimal. Representação hexadecimal.
4	23/Fev	Representação de valores reais. Vírgula fixa e vírgula flutuante. O <i>standard</i> IEEE de vírgula flutuante. Adição e multiplicação em vírgula flutuante.
5	28/Fev	Lógica binária. As operações lógicas básicas.
6	02/Mar	Álgebra de Boole. Axiomas e teoremas. Formas canónicas.
7	07/Mar	Minimização de funções booleanas.
8	09/Mar	<i>Gates</i> e circuitos combinatórios. Os blocos combinatórios básicos: codificadores e decodificadores, <i>multiplexers</i> ; somadores.
9	14/Mar	Circuitos sequenciais. Latch R-S e <i>flip-flops</i> D.
10	16/Mar	Registos. Máquinas e diagramas de estado.
11	21/Mar	Memória: espaço de endereçamento. Endereçagem das células de memória. Tecnologias de memória: SRAM e DRAM.
12	23/Mar	III - O Modelo de VonNeumann As componentes básicas: Memória, Unidade de Processamento, Entrada e Saída de dados, Unidade de Controlo.
13	04/Abr	Estrutura de um processador MIPS. Ciclo básico de processamento das instruções. Organização da memória central dos sistemas de computação. Ciclos básicos de acesso à memória. Os diferentes níveis de utilização dos sistemas de computação: aplicação, linguagem de programação de alto nível, <i>Assembly</i> , serviços do sistema de operação. Função e estrutura. Arquitetura como visão funcional básica comum de uma "família" de processadores. Modelo de programação e repertório de instruções.
14	06/Abr	IV- A arquitectura MIPS: Aspectos chave da arquitectura do MIPS; Instruções e classes de instruções; Princípios básicos de projecto de uma arquitectura; Instruções aritméticas;
15	11/Abr	Teste 1.
16	13/Abr	Codificação de Instruções: Tipo-R, Tipo-I e Tipo-J Programa em Memória: Execução; Interpretação do código máquina Mais Instruções: Lógicas: <i>and</i> , <i>or</i> , <i>xor</i> e <i>nor</i> ; De deslocamento (<i>shift</i>): lógico e aritmético;
17	18/Abr	Multiplicação/Divisão de Inteiros. 'Saltos' condicionais e incondicionais.

		<p>Controlo de fluxo de execução em <i>Assembly</i>:</p> <p>Fluxo condicional: <i>if, if/else</i>;</p> <p>Ciclos de repetição: <i>while</i> e <i>for</i>.</p>
18	20/Abr	<p><i>Arrays</i> - Acesso a elementos</p> <p>Uso de índices; <i>Array</i> de inteiros</p> <p>Código ASCII; <i>Array</i> de bytes</p> <p>Funções</p> <p>Invocação e retorno (<i>Caller/Callee</i>)</p> <p>Convenções: Passagem de argumentos e retorno</p>
19	27/Abr	<p>Uso do <i>Stack</i></p> <p>Salvaguarda de registos (MIPS)</p> <p>Recursividade*</p> <p>Modos de Endereçamento (<i>Addressing</i>)</p> <p>Tipo-R: Só-Registos</p> <p>Tipo-I: Imediato, Endereço-Base, PC-Relativo</p> <p>Tipo-J: Pseudo-Direto</p>
20	2/Mai	<p>Mais <i>Assembly</i></p> <p>Ponteiro: definição e propriedades</p> <p>Uso de <i>arrays</i> com ponteiros</p> <p>Índices versus ponteiros; Exemplos.</p> <p>Instruções <i>signed/unsigned</i>; Resumo.</p>
21	4/Mai	<p><i>Assembling & Loading</i></p> <p>Fases de tradução dum Programa</p> <p>Mapa de Memória</p> <p>Segmentos de Texto e de Dados</p> <p>O Assembler: Diretivas; Pseudo Instruções.</p> <p>O Código Executável (<i>Linking</i> e <i>Loading</i>)</p>
22	9/Mai	<p>MicroArquitetura MIPS <i>Single-cycle</i></p> <p>Tipos de Implementações</p> <p>Arquitetura dum CPU MIPS</p> <p><i>Datapath</i> e Controlo</p> <p>Fases de projeto dum CPU</p> <p><i>Datapath</i></p> <p>Subconjunto de Instruções</p> <p>Execução de Instruções: <i>Load, Store, Tipo-R</i> e <i>Branch</i></p>
23	11/Mai	<p>Unidade de Controlo</p> <p>Decodificador da ALU; Exemplo de ALU Principal</p> <p>Decodificador Principal</p> <p>Exercício: Execução da instrução <i>or</i></p> <p>Instruções adicionais: <i>addi</i> e <i>j(ump)</i></p>
24	23/Mai	Resolução de problemas sobre uArquiteturas <i>Single-cycle</i> .

25	25/Mai	<p>MicroArquitetura MIPS <i>Multicycle</i></p> <p><i>Performance Single-cycle</i>: Caminho Crítico; Tempo de Execução</p> <p>Arquitetura <i>Multicycle</i></p> <p>Limitações do <i>datapath Single-cycle</i>; <i>Multicycle versus Single-cycle</i>.</p> <p><i>Datapath</i></p> <p>Elementos de Estado: Memória Única</p> <p>Execução de instruções: lw, sw, tipo-R e beq</p>
26	30/Mai	<p>Unidade de Controlo: Máquina de Estados (FSM)</p> <p>Mais Instruções: <i>addi</i> e <i>j</i></p> <p>Performance <i>multicycle</i>: caminho crítico; tempo de execução</p>
27	1/Jun	Resolução de problemas sobre uArquiteturas <i>Multicycle</i> .
28	6/Jun	<p>Comunicação com o Exterior (I/O)</p> <p>Introdução</p> <p>Acesso a dispositivos de I/O:</p> <p>I/O Memory-Mapped;</p> <p>Hardware e Software (Asm).</p> <p>Dispositivos de I/O <i>Embedded</i></p> <p>uControlador PIC32 (MIPS);</p> <p>I/O Digital: <i>Switches</i> e LEDs;</p> <p>I/O Série : SPI e UART;</p>
29	8/Jun	<p>Timers; Interrupções;</p> <p>I/O analógico: ADC e DAC;</p> <p>Outros periféricos: LCD e <i>Bluetooth</i></p>

8. Programa da componente laboratorial

Aula	Data	Descrição
1	15/Fev 16/Fev	Representação Informação. Operações Aritméticas.
2	22/Fev 23/Fev	Representação Informação. Operações Aritméticas. Representação de números negativos.
3	1/Mar 2/Mar	Álgebra de Boole. Operações lógicas. Simplificação Algébrica.
4	8/Mar 9/Mar	Blocos básicos combinatórios. Introdução ao software <i>DesignWorks</i> .
5	15/Mar 16/Mar	Circuitos sequenciais, registos e memórias.
6	22/Mar 23/Mar	Teste Prático 1.
7	5/Abr 6/Abr	Introdução ao <i>assembly</i> do MIPS. Simulador do MIPS: MARS.
8	12/Abr 13/Abr	Codificação de estruturas de controlo de fluxo.
9	19/Abr 20/Abr	Codificação de estruturas de controlo de fluxo e processamento de <i>arrays</i> em <i>assembly</i> .
10	26/Abr 27/Abr	Processamento de <i>arrays</i> em <i>assembly</i> .
11	3/Mai 4/Mai	Invocação e retorno de funções em <i>assembly</i> .
12	10/Mai 11/Mai	Invocação e retorno de funções em <i>assembly</i> .
13	24/Mai 25/Mai	Simulação do <i>datapath single-cycle</i> do MIPS.
14	- 1/Jun	Revisões.
15	7/Jun 8/Jun	Teste Prático 2.

9. Regras gerais de funcionamento da disciplina

Regime de faltas

- Todos os estudantes que, não usufruindo do estatuto de trabalhador-estudante no corrente ano letivo, faltarem injustificadamente a mais de 20% das aulas práticas ou a mais de 30% das aulas teórico-práticas reprovam automaticamente à disciplina ficando impedidos de se apresentar a qualquer prova da mesma durante o corrente ano letivo.
- A justificação de faltas deve ser entregue na secretaria do DETI, no prazo máximo de 10 dias de acordo com o regulamento de Estudos, a qual remeterá cópia para o docente coordenador da disciplina.
- Consideram-se faltas justificadas as indicadas no Artigo 23º do Regulamento de Estudos.

E-mails

- Os e-mails sobre questões relativas à componente prática da disciplina devem ser obrigatoriamente dirigidas ao docente responsável pela turma:
 - Pedro Lavrador (plavrador@ua.pt)
 - António Cruz (cruz@ua.pt)
- Os e-mails sobre questões relativas à componente teórico-prática da disciplina ou sobre questões relacionadas com o funcionamento da disciplina devem ser obrigatoriamente dirigidas ao responsável da disciplina: Pedro Lavrador (plavrador@ua.pt)
- Só será dada resposta a e-mails que, no corpo da mensagem, indiquem explicitamente o nome e o número mecanográfico do remetente.

Esclarecimento de dúvidas. Funcionamento das OT

- Para o esclarecimento de dúvidas, durante o período letivo, deverá ser utilizado o período de Orientação Tutorial (OT) para o efeito descrito no horário da disciplina.
- O dia imediatamente anterior aos momentos de avaliação não poderá ser usado para esclarecimento de dúvidas.

Ilícitos

A cópia, no todo ou em parte, de qualquer material entregue para avaliação é considerada fraude. Sem prejuízo de outras medidas, nomeadamente as previstas nos nº 11 e 12 do art. 30º do Regulamento de Estudos da Universidade de Aveiro, a deteção dessa prática implica a atribuição de nota 0 (zero) ao elemento de avaliação em causa.

10. Equipa docente

Pedro Lavrador (plavrador@ua.pt), leciona TP1, P1 e P4.

António Cruz (cruz@ua.pt), leciona TP1, P2 e P3.