

ÍNDICE GERAL

AGRADECIMENTOS IX

PREFÁCIO XI

ÍNDICE DAS SIMULAÇÕES XXV

1 - INTRODUÇÃO AO MUNDO DOS COMPUTADORES 1

1.1	O computador como ferramenta.....	2
1.2	A importância dos computadores	4
1.3	Processamento da informação.....	5
1.4	Estrutura básica de um computador	6
1.5	O mundo com apenas dois símbolos	9
1.6	Interação pessoa-computador	11
1.7	A gestão de um computador	14
1.8	A evolução dos computadores	16
1.9	Perspectivas de evolução futura	24
1.10	Conclusões.....	26

2 - O MUNDO BINÁRIO 29

2.1	Circuitos electrónicos analógicos	30
2.2	Circuitos electrónicos digitais.....	31
2.2.1	Funcionamento básico	31
2.2.2	Diagramas temporais	33
2.2.3	Portas lógicas.....	34
2.3	Álgebra de Boole.....	38
2.4	Funções lógicas	39
2.5	Circuitos combinatórios.....	46
2.5.1	Síntese de circuitos combinatórios	46
2.5.2	<i>Multiplexers</i>	48
2.5.3	Descodificadores	51
2.5.4	ROMs	54

2.6	Circuitos Sequenciais	57
2.6.1	Elementos bi-estáveis	57
2.6.1.1	Trinco SR	57
2.6.1.2	Trinco D	58
2.6.1.3	Báscula D	59
2.6.2	Registos	63
2.6.3	Portas lógicas de três estados (<i>tristate</i>)	65
2.6.4	Banco de registos	66
2.6.5	Contadores	68
2.6.6	Registos de deslocamento	73
2.6.7	Máquinas de estados	74
2.6.7.1	Modelo das máquinas de estados	74
2.6.7.2	Diagramas de estados	76
2.6.7.3	Máquinas de estados sintetizadas	83
2.6.7.4	Máquinas de estados microprogramadas	83
2.7	Representação de números	87
2.7.1	Números em base 10 (decimais) e 2 (binários)	87
2.7.2	Números em base 16 (hexadecimais)	89
2.7.3	Potências de 2	91
2.7.4	Quantos <i>bits</i> para representar um número?	93
2.7.5	Representação de números negativos	94
2.7.6	Representação de números em complemento para 2	95
2.7.7	Extensão do número de <i>bits</i> de um número	97
2.8	Operações aritméticas	99
2.8.1	Soma de dois números binários	99
2.8.2	Subtracção de dois números binários	100
2.8.3	Excesso	101
2.8.4	Multiplicação de dois números binários	103
2.8.5	Divisão de dois números binários	105
2.9	Conclusões	109
2.10	Exercícios	110

3 - O MEU PRIMEIRO COMPUTADOR 113

3.1	Componentes básicos de um computador	114
3.2	RAM – a memória para guardar informação	115
3.3	O processador (PEPE-8)	119
3.3.1	Unidade de dados	120
3.3.1.1	Registo na unidade de dados	120
3.3.1.2	Unidade aritmética e lógica (ALU)	123
3.3.1.3	Funcionamento da unidade de dados	128
3.3.2	Unidade de controlo	130
3.3.2.1	Sinais de controlo	130
3.3.2.2	Contador de Programa (PC)	131
3.3.2.3	Um programa simples	133
3.3.2.4	Constantes no programa	135

3.3.2.5	Salto no programa	140
3.3.2.6	Funcionamento detalhado do programa	142
3.3.3	O processador (PEPE-8) e as memórias	145
3.3.3.1	Processador (PEPE-8)	145
3.3.3.2	Memória de dados	146
3.3.3.3	Memória de instruções	147
3.4	Programação em baixo nível de um computador	149
3.4.1	Instruções em vez de sinais de controlo	149
3.4.2	Linguagem <i>assembly</i>	152
3.4.3	Implementação das instruções	155
3.4.4	Programação em linguagem <i>assembly</i>	160
3.4.5	Programação do PEPE-8 em <i>assembly</i> : contagem de <i>bits</i>	163
3.5	Periféricos	165
3.5.1	Estrutura do <i>hardware</i>	165
3.5.2	Programação com periféricos	173
3.5.2.1	Uso de periféricos de saída	174
3.5.2.2	Uso de periféricos de entrada	176
3.6	Soluções específicas ou genéricas?	178
3.7	Conclusões	179
3.8	Exercícios	180

4 - ARQUITECTURA BÁSICA DE UM PROCESSADOR 183

4.1	Banco de registos	184
4.2	Endereços de dados e de instruções	190
4.2.1	Memórias de dados e de instruções separadas: <i>caches</i>	190
4.2.2	Espaço de endereçamento e mapa de endereços	192
4.3	Impacte da largura das instruções	194
4.4	Endereçamento de <i>byte</i> e de palavra	196
4.5	Codificação das instruções	200
4.6	Registos	203
4.7	<i>Bits</i> de estado	205
4.8	Conjunto de instruções	210
4.9	Instruções de salto	213
4.10	Instruções de transferência de dados	216
4.10.1	Combinações de operandos	216
4.10.2	Transferências entre registos	218
4.10.3	Transferência de uma constante para um registo	219
4.10.4	Transferências entre um registo e a memória	223
4.10.4.1	Endereços constantes e em registos	223
4.10.4.2	Modos de acesso à memória em 16 <i>bits</i>	224

4.10.4.3	Acesso à memória em 16 <i>bits</i> com índice variável	225
4.10.4.4	Acesso à memória em 16 <i>bits</i> sem índice	226
4.10.4.5	Acesso à memória em 16 <i>bits</i> com índice constante	227
4.10.4.6	Instruções de acesso à memória em 16 <i>bits</i>	229
4.10.4.7	Acesso à memória em 8 <i>bits</i>	230
4.10.4.8	Acesso à memória em 8 <i>bits</i> e 16 <i>bits</i>	235
4.10.5	Transferências para memória de uma constante ou memória	237
4.11	Instruções aritméticas	237
4.11.1	Instruções aritméticas mais simples	238
4.11.1.1	Soma e excesso: série de Fibonacci	239
4.11.1.2	Soma e transporte: números grandes	240
4.11.2	Multiplicação e divisão	242
4.12	Instruções Lógicas	246
4.12.1	Funcionalidade das instruções lógicas	246
4.12.2	Expressões booleanas	249
4.12.3	Instruções de manipulação de um só <i>bit</i>	250
4.12.4	Operações lógicas com máscaras	254
4.12.4.1	Funcionamento das máscaras	254
4.12.4.2	Máscaras AND	255
4.12.4.3	Máscaras OR	257
4.12.4.4	Máscaras XOR	259
4.13	Instruções de deslocamento	261
4.13.1	Instruções de deslocamento linear	262
4.13.2	Instruções de deslocamento circular (rotações)	266
4.14	Modos de endereçamento	268
4.15	Conclusões	271
4.16	Exercícios	272

5 - PROGRAMAÇÃO DE UM COMPUTADOR 275

5.1	Um problema simples	276
5.1.1	Modo de actuação de um ser humano	276
5.1.2	Modo de actuação de um computador	276
5.2	Modelação do problema com fluxogramas	277
5.3	Programação em alto nível	280
5.4	Mapeamento da programação de alto nível em linguagem <i>assembly</i>	283
5.5	Dados, declarações e directivas	285
5.5.1	Constantes simbólicas e a directiva EQU	285
5.5.2	Variáveis	286
5.5.2.1	Tipos das variáveis	286
5.5.2.2	Acesso a variáveis de tipos de dados estruturados	287
5.5.2.3	Directivas WORD, TABLE e STRING	288
5.5.3	A directiva PLACE	291
5.5.4	Apontadores	294

5.6	Instruções	298
5.6.1	Atribuição e expressões	300
5.6.2	Decisão	301
5.6.2.1	Decisão simples	301
5.6.2.2	Decisão múltipla	302
5.6.3	Iteração	304
5.7	Rotinas.....	306
5.7.1	Estruturação do código	306
5.7.1.1	Funções nas linguagens de alto nível	306
5.7.1.2	Rotinas em linguagem <i>assembly</i>	310
5.7.1.3	Variante com apontadores.....	314
5.7.2	Mecanismo de chamada e retorno	316
5.7.2.1	Endereço de retorno	316
5.7.2.2	Chamada de rotinas com endereço de retorno em registo.....	317
5.7.2.3	Chamada de rotinas com endereço de retorno na memória (pilha).....	321
5.7.2.4	Qual dos mecanismos de chamada de rotinas se deve usar?	331
5.7.2.5	Variantes do funcionamento da pilha.....	333
5.7.3	Outras utilizações da pilha em rotinas	334
5.7.3.1	Guarda de registos nas rotinas	334
5.7.3.2	Variáveis locais.....	342
5.7.3.3	Passagem de parâmetros e do resultado	344
5.7.3.4	Contextos de chamada das rotinas	351
5.7.3.5	Recursividade.....	356
5.8	Gestão dos dados	360
5.8.1	Quando os registos não chegam	360
5.8.2	Cálculo de expressões.....	361
5.8.3	Execução de instruções imbricadas	362
5.8.4	Tabelas	363
5.8.4.1	Tabelas de uma só dimensão.....	363
5.8.4.2	Tabelas multidimensionais.....	368
5.8.4.3	Tabelas de apontadores.....	374
5.8.5	Estruturas de dados dinâmicas (montão)	378
5.8.6	Listas ligadas	382
5.9	Desenvolvimento de programas	389
5.9.1	Ciclo de desenvolvimento	389
5.9.2	Programação em alto nível ou em linguagem <i>assembly</i> ?	394
5.9.3	Desenvolvimento em linguagem <i>assembly</i>	394
5.9.4	Ambientes de desenvolvimento.....	396
5.9.4.1	Computador alvo e hospedeiro	396
5.9.4.2	Sistemas embebidos	401
5.10	Conclusões.....	404
5.11	Exercícios	405

6 - O COMPUTADOR COMPLETO 409

6.1	Interligação dos componentes de um computador	410
6.1.1	Barramentos.....	410
6.1.2	Operações de leitura e escrita	413
6.1.3	Descodificação de endereços (de palavra).....	416
6.1.3.1	Seleção de dispositivo a aceder.....	416
6.1.3.2	Implementação do mapa de endereços.....	417
6.1.3.3	Descodificação parcial dos endereços.....	422
6.1.3.4	Descodificação de mapas de endereços irregulares	425
6.1.3.5	Descodificação de endereços programável	427
6.1.4	Descodificação de endereços (de <i>byte</i>).....	430
6.1.5	Impacte do endereçamento de <i>byte</i>	435
6.1.5.1	Organização da memória em <i>bytes</i>	435
6.1.5.2	Endereçamento <i>little -endian</i> e <i>big -endian</i>	438
6.1.5.3	Alinhamento dos acessos	443
6.1.6	Ciclos de acesso à memória/periféricos.....	446
6.1.6.1	Ligação ao barramento de dados.....	446
6.1.6.2	Ciclos de leitura e escrita	448
6.1.6.3	Temporizações no acesso aos dispositivos.....	451
6.1.6.4	Acesso a dispositivos lentos.....	456
6.2	Excepções	459
6.2.1	Princípios básicos	459
6.2.2	Interrupções	462
6.2.2.1	Pinos de interrupção.....	462
6.2.2.2	Controlo do atendimento de interrupções	464
6.2.2.3	Comportamento das interrupções.....	466
6.2.2.4	Mecanismo básico de atendimento de interrupções	468
6.2.2.5	Programação com interrupções	469
6.2.2.6	Controlador de interrupções.....	476
6.2.3	Outras excepções.....	478
6.2.3.1	Invocação explícita e retorno de uma excepção	478
6.2.3.2	Excepções predefinidas.....	480
6.3	Tipos de periféricos	482
6.3.1	O que é um periférico?	482
6.3.2	Periféricos de memória de massa	483
6.3.3	Periféricos gráficos	486
6.3.4	Periféricos de comunicação	487
6.3.4.1	Princípios básicos.....	487
6.3.4.2	Comunicação paralela.....	490
6.3.4.3	Comunicação série.....	492
6.4	Arquitectura do sistema de periféricos.....	506
6.4.1	Barramentos hierárquicos	506
6.4.2	Modos de transferência de dados.....	507
6.4.2.1	Transferência por teste (<i>polling</i>)	507
6.4.2.2	Transferência por interrupções.....	509

6.4.2.3	Transferência por acesso directo à memória (DMA)	511
6.4.2.4	Transferência por processador de entradas/saídas	515
6.5	Exemplos de computadores completos	516
6.5.1	Classes de computadores	516
6.5.2	O PC	518
6.5.2.1	Arquitectura original	518
6.5.2.2	Evolução nos processadores	520
6.5.2.3	Evolução nas memórias	527
6.5.2.4	Evolução nos periféricos	530
6.5.2.5	Evolução nos barramentos	531
6.5.3	O microcontrolador	534
6.5.3.1	Características básicas	534
6.5.3.2	CREPE: um microcontrolador baseado no PEPE	538
6.6	Avaliação de desempenho dos computadores	541
6.6.1	O que é o desempenho	541
6.6.2	Programas de avaliação (<i>benchmarks</i>)	543
6.6.3	A lei de Amdahl	545
6.6.4	Avaliação do desempenho do processador	547
6.6.5	Avaliação do desempenho da memória	548
6.6.6	O impacte do compilador	551
6.6.7	A filosofia RISC	552
6.6.8	Avaliação do desempenho dos periféricos	555
6.7	Conclusões	560
6.8	Exercícios	560
7 - O PROCESSADOR EM DETALHE		565
7.1	Diagrama de blocos geral	566
7.2	Núcleo do processador	568
7.2.1	Caminho de dados	568
7.2.1.1	Funcionamento geral	568
7.2.1.2	Banco de registos	572
7.2.1.3	Gerador de constantes	574
7.2.1.4	Unidade aritmética e lógica (ALU)	575
7.2.2	Unidade de excepções	579
7.2.3	Unidade de controlo	580
7.2.4	Microprogramação	585
7.2.4.1	Circuito simples microprogramado	585
7.2.4.2	Microprogramação no PEPE	588
7.3	Processamento em estágios	595
7.3.1	Princípios de funcionamento	595
7.3.2	Cadeias de estágios	599
7.3.3	Implementação das cadeias de estágios	603
7.3.3.1	Cadeia de estágios de instruções	603
7.3.3.2	Cadeia de estágios de microinstruções	608
7.3.4	Excepções com processamento em estágios	610

7.3.5	Dependências de dados.....	613
7.3.6	Dependências de controlo.....	618
7.4	Interface de memória	621
7.5	Caches.....	622
7.5.1	Princípios de funcionamento das <i>caches</i>	622
7.5.2	Organização das <i>caches</i>	625
7.5.2.1	Princípios da organização	625
7.5.2.2	Mapeamento directo.....	627
7.5.2.3	Mapeamento associativo.....	630
7.5.2.4	Mapeamento associativo por conjuntos	633
7.5.3	Políticas de substituição de blocos	635
7.5.4	Políticas de escrita nas <i>caches</i>	636
7.5.5	Evolução do subsistema de <i>caches</i>	638
7.5.6	Casos em que não se quer <i>cache</i>	640
7.5.7	<i>Caches</i> no PEPE.....	642
7.6	Memória virtual.....	643
7.6.1	Hierarquia de memórias	643
7.6.2	Princípios de funcionamento da memória virtual	644
7.6.3	Tradução de endereços virtuais para físicos	647
7.6.4	Gestão das páginas	651
7.6.5	A TLB e o seu papel na tradução de endereços.....	654
7.6.6	Integração da memória virtual e das <i>caches</i>	657
7.6.7	Memória virtual no PEPE.....	660
7.7	Suporte para Processos.....	664
7.7.1	Modelos de programação e de execução	664
7.7.2	Multiprogramação	665
7.7.3	Interação entre processos.....	671
7.7.3.1	Sincronização de baixo nível	671
7.7.3.2	Sincronização com semáforos.....	675
7.7.3.3	Comunicação	678
7.7.4	Programação cooperativa	679
7.7.5	Protecção	682
7.7.6	Gestores de periféricos	685
7.8	Conclusões.....	687
7.9	Exercícios	689

APÊNDICE A - MANUAL DE PROGRAMADOR DO PEPE 695

A.1	Pinos do módulo PEPE	695
A.2	Registos.....	696
A.2.1	Registos principais	696
A.2.2	Registos auxiliares.....	697
A.2.2.1	Configuração do núcleo	698
A.2.2.2	Configuração das <i>caches</i>	699
A.2.2.3	Configuração da memória virtual	700

A.3	Excepções	701
A.4	Conjunto de instruções	701
A.5	Programação do PEPE	708

APÊNDICE B - MANUAL DE PROGRAMADOR DO CREPE 711

B.1	Pinos do módulo CREPE	711
B.2	Registos auxiliares	711
B.3	Funcionamento dos Periféricos	714
B.3.1	Portos de entrada/saída	714
B.3.2	Temporizadores	715
B.3.3	UARTs	716
B.3.4	Informação sobre o estado dos periféricos	718
B.4	Excepções	718
B.5	Exemplo de utilização	720

APÊNDICE C - INTRODUÇÃO AO SIMULADOR (SIMAC) 725

C.1	Desenho de circuitos	725
C.2	Simulação de circuitos	728

APÊNDICE D - COMPUTAÇÃO EM VÍRGULA FLUTUANTE 735

D.1	Representação em vírgula flutuante	735
D.2	A norma IEEE 754	737
D.3	Operações aritméticas em vírgula flutuante	740

APÊNDICE E - CODIFICAÇÃO DE CARACTERES EM ASCII 743

BIBLIOGRAFIA 745

ÍNDICE REMISSIVO 747