



Universidade da Beira Interior

Microprocessadores

Problemas Teóricos

Problemas Práticos

1. Explique sumariamente, considerando o estudado:
 - (a) Qual a importância da dimensão da palavra na melhoria da eficiência de um μ Processador?
 - (b) Qual a melhoria introduzida com o uso de pipelines?
 - (c) Porque é que se usam memórias cache?
 - (d) Quais são as técnicas de escrita em memórias cache?
 - (e) Explique os principais modos de controlo de transferência de informação assíncrona?
 - (f) Quais as principais características dos μ Processador RISC?
 - (g) Como é que se processa o controlo da transferência do programa nas interrupções do μ Processador 8086?
2. Desde o Intel 4004 os microprocessadores evoluíram extraordinariamente. Quais os principais desenvolvimentos tecnológicos que permitiram a melhoria da sua eficiência? Comente em que medida é que cada um desses desenvolvimentos resulta numa melhoria de eficiência.
3. Qual a função do Descodificador de Instruções de um microprocessador?
4. Qual a vantagem da técnica de controlo da transferência de informação de "handshaking" sobre a de "strobing"?
5. Explique os principais modos de transferência de informação entre os microprocessadores e os sistemas de Entrada/Saída?
6. Explique o que é uma transferência DMA (Acesso Directo à Memória).
7. Quais são as vantagens que surgem com o uso de memórias cache? Explique o seu funcionamento.
8. Porque é que a implementação de sistemas com hierarquia de memória permite maior eficiência?
9. Explique o funcionamento da memória cache.
10. Comente em que medida a localização temporal e espacial permitem a eficiência de um sistema de memória hierarquizado.
11. Qual deve ser a dimensão de um bloco de memória de dados se a localização espacial não existe e a localização temporal é elevada? Comente a resposta.
12. Explique de forma genérica o funcionamento da memória virtual de um sistema de memória.
13. O objectivo de uma "pipeline" consiste em reduzir o tempo médio de execução para um período de clock. Quais as razões que levam a que isso não seja de facto assim?
14. Em que medida tecnologias como "superpipelining" e superescalaridade melhoram o desempenho de um μ Processador?
15. Quais as desvantagens e as vantagens do uso de pipelines longas (com muitos estágios)?
16. "Os saltos condicionais são a principal razão da perda de eficiência das pipelines e sistemas escalares". Comente esta afirmação, referindo quais os métodos que permitem diminuir essa perda de eficiência, e quais as outras causas que também originam perdas de eficiência.

17. Nos sistemas escalares existe multiplicação de unidades funcionais, como sejam ALUs, FPUs, etc. Será que se justifica ter cada uma das unidades funcionais por cada pipeline? Justifique.
18. Comente a afirmação: "A reordenação de instruções é uma forma de diminuir os efeitos negativos que reduzem a eficiência da pipeline."
19. Comente a afirmação: "Os microprocessadores CISC actuais incorporam muitas características dos microprocessadores RISC, aumentando a sua eficiência."
20. Porque é que os microprocessadores RISC têm uma arquitectura que facilitam a construção de pipelines escalares mais optimizadas?
21. Comente a afirmação: "Os microprocessadores CISC actuais incorporam muitas características dos microprocessadores RISC, aumentando a sua eficiência."
22. Comente a seguinte afirmação: "O espaço físico ocupado pela unidade de controlo dum microprocessador RISC no circuito integrado é muito mais pequeno que num CISC, permitindo que o espaço restante seja utilizado para melhorar a eficiência do microprocessador".
23. Explique quais as principais características de um microprocessador RISC? Quais são as possíveis vantagens que daí advêm?
24. Explique o funcionamento das interrupções no microprocessador 8086 da Intel.
25. Como é que se processa o controlo da transferência do programa nas interrupções do μ Processador 8086?

26. Na figura 1 é apresentado um circuito de memória.

- Classifique os circuitos relativamente ao tipo de memória. Qual a sua capacidade de armazenamento?
- Quais os endereços ocupados pelos circuitos de memória?

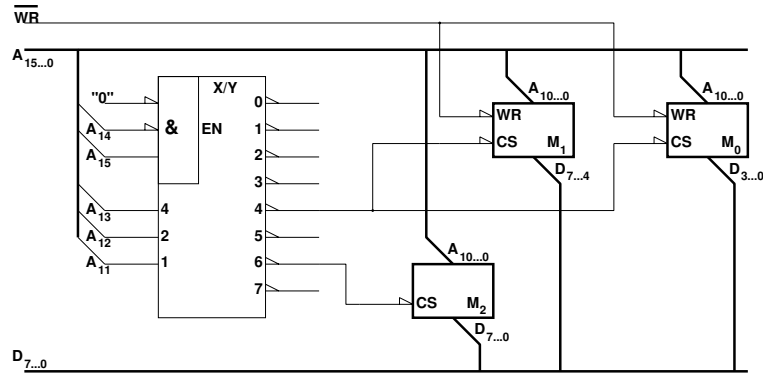


Figura 1

27. Na figura 2 está representado um circuito de memória incompleto.

- Classifique os circuitos relativamente ao tipo de memória. Qual a sua capacidade de armazenamento?
- Ligue as entradas e saídas do decodificador adequadamente, de forma a que as memórias fiquem respectivamente com os endereços: M_0, M_1 : 2000H-23FFH e M_2 : 3C00H-3FFFH

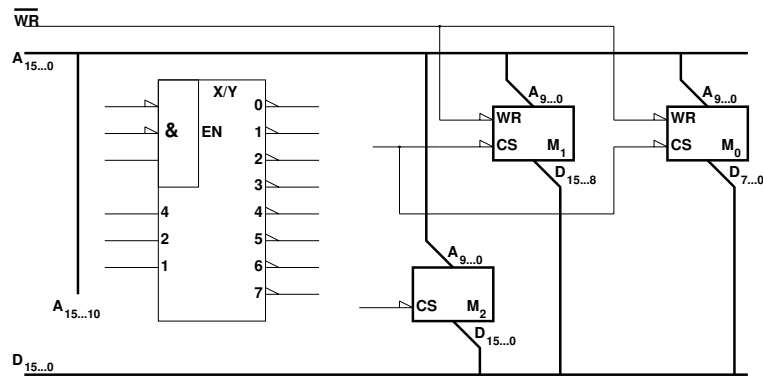


Figura 2

28. Considere o circuito da figura 3.

- Caracterize os circuitos de memória representados no circuito.
- Obtenha um mapa de endereços para este circuito de memória.

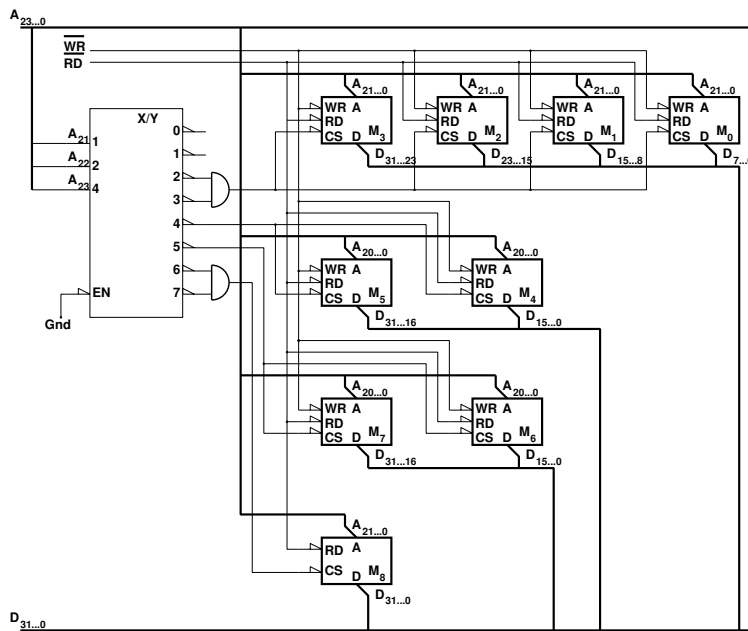


Figura 3

29. Num sistema com barramentos de dados e de endereços de 16 linhas, pretende-se criar uma zona de memória ROM e uma zona de memória RAM.

A zona de memória ROM tem 4K palavras e encontra-se nas posições correspondentes aos endereços mais baixos do mapa de memória do sistema. Para implementar esta zona estão disponíveis circuitos integrados ROM com uma organização interna de $2K \times 16$ com entrada de selecção activa no nível baixo (\overline{CS}).

A zona de memória RAM tem 2K palavras e encontra-se nas posições correspondentes aos endereços imediatamente acima do meio do mapa de memória do sistema. Para implementar esta zona estão disponíveis circuitos integrados RAM com uma organização interna de 1Kbyte com entrada de selecção activa no nível baixo (\overline{CS}).

Considere que as eventuais entradas de habilitação de saída e de escrita dos circuitos integrados de memória são activas no nível baixo.

- Quantos circuitos integrados ROM e RAM são necessários para implementar os blocos de memória acima referidos?
- Desenhe um diagrama lógico que represente a ligação dos circuitos integrados ROM e RAM aos barramentos de dados e de endereços, e às linhas de controlo (\overline{RD} , \overline{WR}) do sistema.
- Desenhe um circuito de descodificação para a selecção dos vários circuitos integrados de memória.

30. Considere um sistema baseado no microprocessador 8086 da Intel. Se $AX=AB45H$, $BX=0200H$, $DS=7000H$ e todos os outros registos forem $12ABH$, qual o valor dos barramentos quando se executa a instrução `MOV [BX][DI]+3000H, AX.` nas transição ascendentes **T1** e **T3** do ciclo de acesso à memória?

31. Considere um sistema baseado no μ Processador 8086 da Intel. Se $DI=10ACH$, $SI=89A4H$, $DS=CC00H$, $ES=8800H$ e todos os outros registos forem $F200H$, qual o valor dos barramentos quando se executa a instrução `CMP SW` na transição ascendente **T1** dos ciclos de acesso à memória?

32. Considere um sistema baseado no μ Processador 8086 da Intel. Se $DI=325AH$, $SI=34C7H$, $DS=7000H$, $ES=3000H$ e todos os outros registos forem $A4C5H$, qual o valor dos barramentos quando se executa a instrução `MOV SW` na transição ascendente **T1** dos ciclos de acesso à memória?

33. Considere um sistema baseado no microprocessador 8086 da Intel. Se $AX =03D5H$ e todos os outros registos forem $0100H$, qual o valor dos barramentos quando se executa a última instrução da sequência da figura ao lado, nas transições ascendentes **T1** e **T3** do ciclo de escrita do porto?

```
MOV [BX][DI],AX
MOV DX,[BX][SI]
OUT DX,AX
```

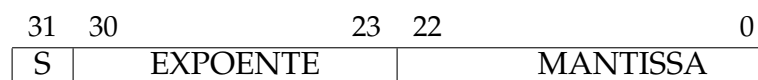
34. Escreva uma rotina na linguagem assembly do microprocessador da intel 8086 que implemente a função `strcmp()` do C. Esta função compara dois vectores de inteiros de oito bits que terminam com NULL (valor 0). Se os dois vectores forem iguais retorna 0. Caso contrário, retorna a diferença entre os dois primeiros bytes que são diferentes, considerando-os como inteiros sem sinal (notar que o resultado dessa diferença pode ser negativo).

Para implementar a rotina considere que os dois ponteiros para os dois vectores estão nos registos BX e BP , e que o resultado final deve ser devolvido no registo AX . Nenhum outro registo deve ser retornado alterado.

35. Escreva uma rotina na linguagem assembly do μ Processador da intel 8086 que implemente a função `strcasncmp(const char *s1, const char *s2)` do C. Esta função compara dois vectores de caracteres que terminam com NULL (valor 0) ignorando se as letras neles contidas são maiúsculas ou minúsculas. Se os dois vectores forem iguais retorna 0. Caso contrário, retorna a diferença entre os dois primeiros caracteres que são diferentes.

Para implementar a rotina considere que os dois ponteiros para os dois vectores estão nos registos BX e BP , e que o resultado final deve ser devolvido no registo AX . Nenhum outro registo deve ser retornado alterado.

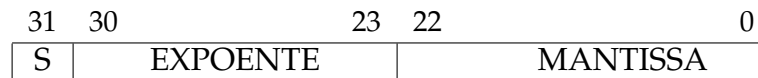
36. Desenvolva uma rotina em assembly do 8086 (ou alternativamente, do 386) que elabore a operação de multiplicação de um número real representados em vírgula flutuante ("floating point") de precisão simples segundo a norma do IEEE-754, versão 10 (ver figura seguinte) pelo número 2^n , sendo n um número inteiro maior que zero dado no registo AL .



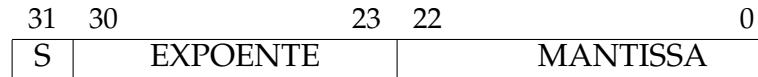
37. Explique como se processa o acesso à memória na operação de leitura de uma "Long Word" que começa no endereço $8FC35H$, num sistema baseado num μ Processador 386 da Intel e com uma memória organizada em "Long Words"(palavras de 32 bits). E se em vez da leitura duma "Long Word" for a escrita duma "Word" na mesma posição.

38. Considere que a partir da posição de memória apontada pelo registo BX se encontram CX palavras de 16 bits, com 4 números BCD cada. Pretende-se que escreva uma rotina na linguagem assembly do μ Processador Intel 8086 que coloque cada um destes números BCD em posições de memória consecutivas (com dimensão de oito bits) a começar na posição apontada pelo registo BP . A rotina não deve retornar nenhum registo alterado.

39. Desenvolva uma rotina em assembly do 8086 (ou alternativamente, do 386) que verifique se um número real representados em vírgula flutuante ("floating point") de precisão simples segundo a norma do IEEE-754, versão 10 (ver figura seguinte) é maior ou igual do que um número inteiro n dado no registo AL.



40. Desenvolva uma rotina em assembly do 8086 (ou alternativamente, do 386) que elabore a operação de multiplicação de dois números reais representados em vírgula flutuante ("floating point") segundo a norma do IEEE-754, versão 10 (ver figura seguinte). Para maior simplicidade, trunque a parte decimal do resultado, e faça eventuais multiplicações considerando apenas os 16 bits mais significativos do número.



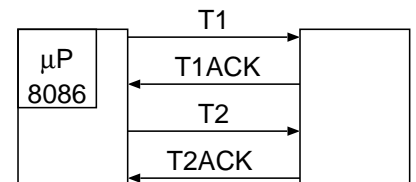
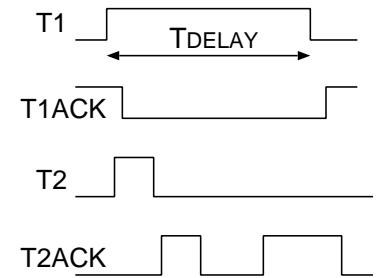
NOTA: Tenha em atenção o seguinte exemplo:

$$\left(-1.1110010101000110010010 \times 10^{00111011}\right) \times \left(+1.0010000000000000000000 \times 10^{00111000}\right) =$$

$$\left(-1.0001000011110111100010 \times 10^{01110100}\right)$$

41. Pretende-se projectar um sistema de supervisão de um equipamento industrial baseado no microprocessador da Intel 8086. Considere que tem a seu cargo o projecto de alguns dos módulos que vão permitir o sistema de testes. Sendo assim, projecte os seguintes módulos.

- (a) Projecte a rotina que faz 2 testes ao sistema. Para esse efeito usa-se o porto PTESTOUT com o formato ----- T2 T1 e o porto PTESTIN com o formato ----- T2ACK T1ACK. O teste 1 consiste em activar T1 durante TDELAY1 (considere uma rotina DELAY1 que permite fazer um atraso TDELAY1). O teste fica positivo se durante esse tempo o sinal T1ACK passar a nível baixo e se mantiver nesse nível (ver figura). Para implementar esse teste a entrada T1ACK também vai activar a interrupção INT 201 (use etiqueta RINT201). O teste 2 consiste em por T2 activo em nível alto passando depois a nível baixo. Isso deve desencadear uma sequência em T2ACK que assume o nível alto seguido do nível baixo seguido novamente de nível alto e nível baixo. Esta sequência pode ser controlada por "polling". Todavia a sequência de T2ACK tem um tempo máximo para ser cumprida. Caso não seja cumprida no tempo máximo, a interrupção INT 202 será activada (use etiqueta RINT202). Se ambos os testes falharem o AL deve ser 3 no retorno, será 2 se falhar o teste 2, e deve ser 1 se falhar o teste 1. Caso nenhum teste falhe AL deve ser 0. Nenhum outro registo pode ficar alterado por esta rotina.

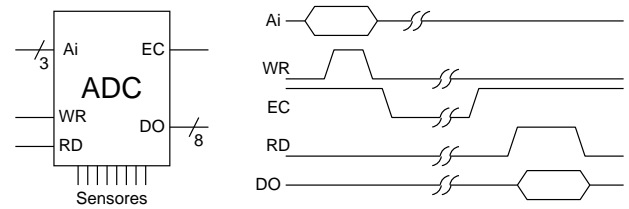


- (b) O resultado dos testes deve ser impresso numa impressora ASCII paralela (cada teste deve ser reportado numa linha). Explique como faria a implementação de um porto paralelo e escreva uma rotina que estabeleça o envio dos dados para a impressão.
- (c) Como se pode obter o endereço que corresponde a RINT201 e RINT202?

42. Pretende-se projectar um sistema de controlo para um edifício inteligente baseado no microprocessador da Intel 8086. Considere que tem a seu cargo o projecto de alguns dos módulos que vão permitir todo o controlo do sistema. Sendo assim, projecte os seguintes módulos.

- (a) O edifício tem oito sensores ligados a um ADC de oito entradas como o da figura. As linhas $A_2A_1A_0$ ligadas ao porto PSENS selecionam o sensor lido pelo ADC (0 a 7). O ADC é controlado pelo porto PCONTRADC=RD 0 0 0 0 0 WR, e é lido pelo porto PINADC.

Considerando que o ADC gera a interrupção INT 128 a que é atribuído um endereço com etiqueta ADCINTR, escreva a subrotina pretendida considerando que o sensor a ler é transportado no registo AL, e que o resultado da leitura do ADC é devolvido no mesmo registo. Todos os outros registos se devem manter inalterados.

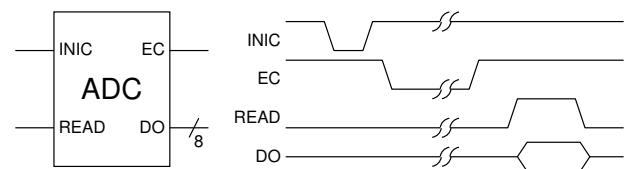


- (b) Considere um ADC genérico, cujos diagramas temporais estão representados na figura. A entrada no ADC, INIC, faz a recolha de uma amostra. A saída do ADC, EC, sinaliza o fim da conversão. A entrada do ADC, READ, permite a leitura do ADC. Considerando um porto de entrada PINADC onde está ligada a saída de dados do ADC, DO de 8 bits, um porto de saída POUTCNTR onde ligam os sinais de controlo

READ								INIC
------	--	--	--	--	--	--	--	------

, e um porto de saída POUTDAC onde se liga um DAC genérico, faça um programa para o microprocessador 8086 que controle e leia o ADC e envie o valor da operação $y[n] = x[n] + y[n-1] - \frac{1}{2}(x[n-1] + x[n-2])$ ao porto POUTDAC, para ser convertido pelo DAC.

- Considere que a amostra n do sinal digital de entrada, lido no ADC, é representado por $x[n]$, e a do sinal de saída, escrito no DAC, é representado por $y[n]$.
- Considere que tem disponível a interrupção NMI que quando activada chama uma rotina com etiqueta NMIROT.
- Considere que existe um circuito independente que gera o "Clock" do ADC.



- (c) Pretende-se controlar uma cancela de um parque de estacionamento. Esta cancela é movimentada por um motor passo a passo, e para funcionar, devem ser colocados a 1, e em sequência, os bits de um porto de saída POUTMTR

				P_3	P_2	P_1	P_0
--	--	--	--	-------	-------	-------	-------

. A sequência $P_3 \rightarrow \dots \rightarrow P_0 \rightarrow P_3 \dots$ abre a cancela e a sequência oposta fecha, devendo cada bit ser activado durante o tempo que dura a rotina DELAY. O controlo da cancela é feito através do porto de entrada PIN

				I_3	I_2	I_1	I_0
--	--	--	--	-------	-------	-------	-------

. A I_0 está ligado o comando da abertura da cancela, I_1 sinaliza que a cancela está completamente aberta, I_2 sinaliza que o veículo completou a passagem pela cancela e I_3 sinaliza que a cancela está completamente fechada.

- (d) O sistema de controlo tem que imprimir regularmente relatórios numa impressora paralela ligada a um porto PPIOUT. Como é sabido, o porto paralelo usa um protocolo de "handshaking", e para implementar o controlo está disponível um porto de Entrada/Saída PCNTRPPI em que o bit mais significativo é de entrada e o menos significativo é de saída. Implemente a subrotina Assembly que permita enviar o byte presente em AL para a impressora usando uma técnica de "polling" adequada. Todos os registos do processador devem manter-se inalterados.

- (e) O edifício tem 4 portas que perante a aproximação de uma pessoa abrem durante 2 segundos e depois fecham se não está alguém na proximidade. Considerando que os sensores de presença estão nos endereços de I/O 0F0H a 0F3H (se sinalizarem 1 existe alguém junto à porta, se sinalizarem 0 não há ninguém) escreva um programa que faça o controlo da abertura e fecho dessas portas de forma independente. Considere que para o efeito tem uma rotina de temporização de 0.4 segundos com etiqueta TMP04 e que o máximo erro admitido é de 0.5 segundos. Considere ainda que os sensores de presença activam a interrupção INT 129, a que corresponde uma rotina de interrupção com etiqueta PRTSNS. As portas são abertas e fechadas pelo controlo do porto PRTOPEN=0000P₃P₂P₁P₀, em que P_i abre ou fecha a porta i consoante seja 1 ou 0 respectivamente.

43. Usando o microprocessador 8086 projecte um sistema que controle dois semáforos com funcionamento independente, e em que um controle a passagem de veículos numa passagem de nível e outro controle a velocidade dos veículos. Assim, o sistema tem um sensor que sinaliza a chegada de um comboio com

nível lógico 1. A partir desse momento, um semáforo fica amarelo durante 2 segundos, e depois deve ficar vermelho até que um sensor fica activo, sinalizando o termo da passagem do comboio na passagem de nível.

Além deste sistema, existe um semáforo que controla eventuais excessos de velocidade. Quando um sensor de velocidade é activado, deve-se colocar o amarelo durante 1 segundo e passar a vermelho durante 3 segundos. O erro máximo admitido deve ser de 0.2 segundos.

Para o efeito, tem-se disponível:

- Rotina de temporização com etiqueta TMP200 que produz um atraso de 0.2 segundos.
- Porto de saída PSEMAF onde são ligados os semáforos $G_1 Y_1 R_1 G_0 Y_0 R_0$ com G_i, Y_i e R_i a controlarem as luzes dos semáforos verde, amarelo e vermelho repectivamente, que se acendem no nível lógico 1. O índice 0 refere-se ao semáforo da passagem de nível. O índice 1 refere-se ao semáforo do sistema de excesso de velocidade.
- Porto de entrada PSRS onde são ligados os sensores relativos aos dois semáforos, $ExV FPC IPC$, que respectivamente sinalizam excesso de velocidade, fim da passagem do comboio e início da passagem do comboio respectivamente.
- A interrupção NMI que chama uma rotina com etiqueta NMIROT.

Nota: Faça um fluxograma representativo do algoritmo seguido.

44. Considere um ADC genérico, cujos diagramas temporais estão representados na figura. A entrada no ADC RD faz a recolha de uma amostra. A saída do ADC EC sinaliza o fim da conversão. Considerando um porto $PINADC$ onde está ligada a saída de dados do ADC, DO de 8 bits, um porto $POUTCNTR$ onde liga o sinal de controlo RD , faça um programa para o microprocessador 8086 que controle e leia o ADC e coloque as leituras sequencialmente a partir da posição de memória com etiqueta $INDATA$.

Ao fim de 128 leituras, o programa deve recomeçar a escrever a partir da posição de memória com etiqueta $INDATA$. Considere que tem disponível a interrupção NMI que quando activada chama uma rotina com etiqueta $NMIROT$.

