

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
1.º Teste – 2006/2007, 1.º semestre lectivo

1. (2 valores)

Relativamente à arquitectura PC, descreva as principais características das gerações PC-XT e PC-AT.

2. (3 valores)

Em todas as gerações de PC posteriores ao PC-XT, existe *hardware* para adaptar o barramento de 8 *bit*, de periféricos e memórias, à largura de dados do processador utilizado.

a) [1] Porquê?

b) [2] Que implicações resultariam se não existisse?

3. (2 valores)

Um determinado sistema funciona com dois módulos SIMM de 30 pinos mas não funciona com apenas um. Que característica do processador justifica este comportamento?

4. (2 valores)

Comente a seguinte afirmação: “Na evolução da arquitectura PC, com o i80386 torna-se evidente a necessidade de *cache*.”.

5. (5 valores)

Relativamente às DRAMs, descreva:

a) [2] Os diferentes tipo de *refresh*.

b) [2] A importância do *refresh Cas-Before-Ras* para o desenho do controlador das DRAMs.

c) [1] A característica que possibilita o refrescamento em simultâneo de vários bancos de memória.

6. (6 valores)

Considere uma *cache* do tipo *8-way set associative* com 1024k *byte* de RAM de dados em linhas de 32 *byte*, integrada num sistema com endereçamento a 32 *bit*.

a) [2] Determine, em *bits*, a dimensão dos campos OFFSET, INDEX (SET) e TAG.

b) [2] Descreva uma possível política de substituição que seja eficiente.

c) [2] A taxa de ocupação de uma *cache* com *4-way set associative*, com a mesma dimensão de memória de dados, é menor que a de uma *8-way set associative*. Indique um cenário que confirme esta afirmação.

2006.10.26

Mário Simões, Pedro Sampaio, João Trindade

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
2.º Teste – 2006/2007, 1.º semestre lectivo

1. [3 valores]

No bus PCI, o final de uma transacção pode ser identificado pela ocorrência simultânea de IRDY#=1 e TRDY#=1?

2. [3 valores]

A associação da mesma linha de interrupção do PIC (Programmable Interrupt Controller) a vários dispositivos obriga à execução de múltiplos handlers (rotinas de tratamento de interrupção) sempre que um desses dispositivos gerar uma interrupção. Por outro lado, a norma PCI obriga a que todos os dispositivos que possam gerar interrupções utilizem (pelo menos) a linha INTA#.

Comente a afirmação: “O atendimento de uma interrupção INTA# obriga, tipicamente, à execução de uma cadeia de handlers muito mais longa do que a de uma interrupção INTD#.”

3. [3 valores]

Caracterize os endpoints que cada um dos seguintes dispositivos USB deve apresentar:

- a) [1] Leitor de códigos de barras
- b) [1] Receptor de rádio
- c) [1] Drive de disquetes

4. [4 valores]

Num bus USB, o host e os hubs cooperam de modo a garantir que apenas um dispositivo (em configuração) responde ao endereço 0.

- a) [2] Explique a importância deste requisito e descreva a forma de o assegurar.
- b) [2] Seria viável realizar alterações para que pudessem existir dois dispositivos em configuração no mesmo bus (podendo reservar outro endereço além do 0, se necessário)? Se não, porquê? Se sim, descreva as alterações.

5. [4 valores]

Escreva, em assembly IA-32, o código da função

`unsigned int strspn(const char * str, const char * strCharSet);`

que retorna o índice do primeiro carácter de str que não ocorre em strCharSet.

Exemplo: `strspn("acabada", "abc")` deve retornar 5.

6. [3 valores]

Escreva, em assembly IA-32, o código da função

`const char * longest_match(const char * vstr[], unsigned int nstrs, const char * strCharSet);`

que recebe, em vstr, um array de nstrs strings e retorna a que começar com o maior número de caracteres presentes em strCharSet.

Exemplo: sendo `const char * palavras[] = { "acabei", "abacate", "abadessa" }`, a execução de `longest_match(palavras, 3, "abc")` deve retornar o endereço de "abacate".

Tire partido da função desenvolvida em 5.

2006.12.09

Mário Simões, Pedro Sampaio, João Trindade

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 1.^a Época – 2006/2007, 1.º semestre lectivo

Esta prova é constituída por duas partes e destina-se à repetição de um dos testes parciais ou à realização de teste total. Quem realizar repetição de um teste responde somente a uma das partes e termina após 1 hora e 30 minutos. Quem realizar teste total termina após 3 horas; a classificação é calculada pela média aritmética das duas partes.

Parte I (ou repetição do 1º teste)

1. [4 valores]

Relativamente à evolução da arquitectura PC, explique como é assegurado o suporte de

- a) [2] Placas de expansão desenhadas para PC-XT (8088) no PC-AT (286);
- b) [2] Placas de expansão desenhadas para PC-AT (286) em sistemas com *bus* PCI.

2. [3 valores]

As células de memória das RAMs estáticas também são usadas, com modificações, nas RAMs dinâmicas. Para quê? Tendo em conta que leituras e escritas são feitas sobre o mesmo tipo de célula, por que motivo é mais lento o acesso às RAMs dinâmicas?

3. [5 valores]

Nas memórias DRAM, os sinais de endereço são multiplexados – através dos mesmos pinos é passado o endereço de linha, registado pelo sinal RAS, e o endereço de coluna, registado pelo sinal CAS.

- a) [1] Qual é a importância de o endereçamento ser multiplexado?
- b) [2] Em alguma circunstância ocorre activação de RAS ou de CAS sem registar endereço? Justifique.
- c) [2] Em alguma circunstância há *bits* de endereço de coluna registados pelo sinal RAS ou *bits* de endereço de linha registados pelo sinal CAS? Justifique.

4. [8 valores]

No "*Intel® 64 and IA-32 Architectures Optimization Reference Manual*" (Nov. 06, pp. 3-61) afirma-se que:

"On Intel Core 2 Duo, Intel Core Duo, Intel Core Solo, and Pentium M processors, there will be an excess of first-level cache misses for more than 8 simultaneous references to addresses that are apart by 4-KByte modulus."

Tradução livre: Nos processadores Intel Core 2 Duo, Intel Core Duo, Intel Core Solo e Pentium M, a utilização simultânea de mais de 8 localizações afastadas entre si de múltiplos de 4 *Kbytes* resultará num excesso de ocorrências de *cache-misses* no primeiro nível de *cache*.

Sabendo que a dimensão de cada linha de *cache* é de 64 *bytes*:

- a) [2] Qual o tipo de organização da *cache*? Justifique.
- b) [3] Qual a capacidade (em dados) da *cache*? Apresente os cálculos.
- c) [3] Considerando um espaço de endereçamento a 32 *bits*, qual a capacidade da RAM de *tags*? Apresente os cálculos.

Parte II (ou repetição do 2º teste)

5. [4 valores]

Relativamente ao *bus* PCI,

a) [2] Descreva os significados dos sinais C/BE#[3:0].

b) [2] Na leitura de diagramas temporais, é possível distinguir se uma transacção é de leitura ou escrita sem observar os sinais C/BE#[3:0]?

6. [4 valores]

A norma PCI especifica um *bus master/slave* que suporta a coexistência de vários *masters*. A norma USB especifica um *bus master/slave* com apenas um *master*.

a) [2] No caso do PCI, como é gerido o acesso ao *bus* quando vários dispositivos têm dados para transferir?

b) [2] No caso do USB, como é gerido o acesso ao *bus* quando vários dispositivos têm dados para transferir?

7. [3 valores]

Relativamente ao USB, explique se é possível um dispositivo ter a garantia de que os dados que quer enviar são recebidos sem erros.

8. [9 valores]

a) [5] Implemente, em *assembly* IA-32, a seguinte função:

```
char *struprnt(char *str, unsigned int *pcnt);
```

que converte para maiúsculas os caracteres da *string* *str*, deixando na localização apontada por *pcnt* o número de caracteres convertidos. A função retorna o endereço da sequência convertida.

Para a conversão de cada carácter, utilize a função *toupper*, que recebe o valor de um carácter *e*, se este for uma letra minúscula, retorna o valor da maiúscula correspondente; nas outras situações retorna o valor recebido, sem modificação.

```
int toupper(int c);
```

Se precisar de verificar se um carácter corresponde a uma letra minúscula, utilize a função *islower*, que recebe o valor de um carácter, retornando um valor diferente de zero se este corresponder a uma minúscula.

```
int islower(int c);
```

b) [4] Considere as seguintes definições:

```
struct StrToConvert { unsigned int nconverted; char *str; };  
unsigned int ConvertAllStrs(struct StrToConvert toConv[],  
                           unsigned int nconvs,  
                           char * (*strconvf)(char *, unsigned int *));
```

Implemente, em *assembly* IA-32, a função *ConvertAllStrs*, que aplica a função de conversão apontada por *strconvf* às *strings* apontadas pelos campos *str* de cada um dos *nconvs* elementos do *array* *toConv*, retornando o número total de caracteres convertidos.

A função de conversão apontada por *strconvf* recebe, como primeiro argumento o endereço da *string* a converter *e*, como segundo argumento, o ponteiro para a localização onde deixar o número de caracteres convertidos. Retorna o endereço da sequência convertida.

2007.01.15

Mário Simões, Pedro Sampaio, João Trindade

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 2.^a Época – 2006/2007, 1.º semestre lectivo

1. (2 valores)

Comente a afirmação, relativa à evolução da arquitectura PC: “Enquanto nas gerações XT e AT-286 é viável expandir a DRAM do sistema usando placas de memória colocadas no *bus* de expansão, nas versões posteriores, com o processador 386 ou sucessores, só é interessante o uso de DRAM ligada à *motherboard*”.

2. (3 valores)

Na geração de SIMM de 8 bits de dados, utilizavam-se DRAMs organizadas com dados a 1 bit. Na geração de DIMM de 64 bits de dados, utilizam-se vulgarmente DRAMs organizadas com dados a 8 bits. Apresente as razões que motivam estas opções, tendo em conta:

- i) Espaço físico ocupado;
- ii) Complexidade da lógica de controlo;
- iii) Eficiência dos acessos.

Pode ilustrar as razões com exemplos de dimensões de módulos de memória.

3. (2 valores)

Para a concepção de uma *cache*, considerando uma determinada dimensão de RAM de dados, apresente as vantagens e inconvenientes de optar por mapeamento directo ou *2-way set associative*.

4. (2 valores)

Numa *cache* do tipo *4-way set associative*, cada valor de SET (INDEX) selecciona um campo de TAG ou quatro campos de TAG em simultâneo? Justifique.

5. (3 valores)

Considere um sistema com *bus* PCI a 32 bit com dois *masters*, designados por A e B, com um esquema de prioridades fixas em que A é o mais prioritário.

O *master* A pretende fazer uma transacção:

i) Escrita de 5 *bytes* com os valores 0x06, 0x43, 0x75, 0x5f e 0xff da memória com o endereço inicial 0x10da0c0a; Admita que o *target* introduz um estado de espera em cada fase de dados.

O *master* B pretende fazer duas transacções:

ii) Leitura de 4 *bytes* com os valores 0xa6, 0xc3, 0x47 e 0x3c da memória com o endereço inicial 0x00800104; Admita que o *master* introduz um estado de espera;

iii) Escrita de 2 *bytes* com o valor 0x2d e 0x04, na memória com o endereço inicial 0x0080100c.

Supondo que o *master* A pede acesso ao *bus* após o *master* B tomar posse para realizar a primeira transacção, desenhe os diagramas temporais destas transacções. Represente a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal.

6. (1 valor)

Comparando o funcionamento dos dispositivos USB *low-speed* e *full-speed*, observa-se que a comunicação é codificada com polaridade eléctrica inversa. Descreva a razão e as consequências desta característica.

7. (2 valores)

Da especificação USB justifique:

- a) [1] obrigatoriedade de existir, nos dispositivos, o *endpoint* 0 de entrada e saída.
- b) [1] A necessidade de definir dois tipos de canais (*stream pipes* e *message pipes*) para as comunicações entre o *host* e os *devices*.

8. (5 valores)

Escreva, em *assembly* IA-32, o código da função

```
char * getStringAddress(int count, const char * match, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de ponteiros para *strings*, com tipo `char *`, e compara cada uma das *strings* com a apontada por `match`, terminando quando tiver encontrado `count` *strings* idênticas a `match` e retornando o endereço da última delas. Se não houver *strings* idênticas em número suficiente, retorna NULL. A lista de parâmetros variável é terminada por um ponteiro com o valor NULL.

Para fazer as comparações, deve utilizar a função de biblioteca

```
int strcmp( const char * str1, const char * str2 );
```

que retorna zero se as *strings* apontadas por `str1` e `str2` forem idênticas.

2007.02.01

Mário Simões, Pedro Sampaio, João Trindade

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Exame de época especial, Março de 2007

1. (1,5 valores)

As versões da arquitectura PC posteriores ao modelo XT incluem circuitos de conversão para dispor de um bus de 8 bits a partir da largura de dados do processador.

a) (0,5) Explique a importância desta conversão.

b) (1) Indique o número de registos *latch* de 8 bits pertencentes a esta conversão, para os casos de processadores com dados a 16 e a 32 bits. Descreva as circunstâncias em que estes registos são utilizados.

2. (1 valor)

Em relação ao *Programmable Interrupt Controller* (PIC) 8259, explique para que serve, porque razão foi incluído no IBM-PC e por que motivo foi adicionado um segundo PIC no PC-AT.

3. (3,5 valores)

Para o projecto de um sistema embebido baseado no processador 486, pretende-se construir o módulo de memória principal com 64 Mbytes.

a) (1) Desenhe o módulo de memória utilizando circuitos integrados DRAM de 16 M x 1. Quantos circuitos integrados são necessários e quantos pinos têm? Quantos ciclos são necessários para o seu refrescamento completo?

b) (1) Desenhe o módulo de memória utilizando circuitos integrados DRAM com a mesma capacidade em bits mas com organização a 4 bits de dados. Quantos circuitos integrados são necessários e quantos pinos têm? Quantos ciclos são necessários para o seu refrescamento completo?

c) (1) Desenhe o módulo de memória utilizando circuitos integrados SRAM com organização a 8 bits de dados, admitindo que estes circuitos têm a mesma área de silício que os de DRAM anteriores e que uma célula de memória SRAM ocupa o quádruplo da área de uma célula de DRAM. Quantos circuitos integrados são necessários e quantos pinos têm?

d) (0,5) Indique, justificando, qual das opções anteriores é preferível.

4. (3,5 valores)

Considere um processador com *cache* interna L1 e L2. Admita que a *cache* L2 é 4-way *set associative*, com 512 Kbyte de dados em linhas de 32 bytes.

a) (1) Qual é a dimensão, em bits, da memória de *Tag*? Apresente os cálculos.

b) (1) Que vantagens e inconvenientes teria a modificação para uma organização 8-way?

c) (1) Que vantagens e inconvenientes teria a modificação da dimensão da linha para 64 bytes?

d) (0,5) Admitindo a possibilidade de dispor de maior área de circuito integrado para memória de *cache*, explique os critérios a seguir para utilizar essa memória em *Data* ou em *Tag*.

5. (2,5 valores)

Considere um *bus* PCI a 32 *bits* com dois *masters*, designados por A e B, com um esquema de prioridade rotativa, sendo o *master* A o primeiro a obter o bus no conjunto de transacções seguinte:

i) O *master* A pretende realizar duas transacções de leitura de memória de quatro *bytes*, a primeira com os valores 0x12, 0x34, 0x56 e 0x78, do endereço 0x00010002 e a segunda com os valores 0x11, 0x22, 0x33 e 0x44, do endereço 0x00018004.

ii) O *master* B pretende realizar duas transacções de escrita em memória, a primeira de um byte com o valor 0x5a no endereço 0x0008f001 e a segunda de oito bytes com os valores 0x32, 0xf4, 0xff, 0xfd, 0x51, 0xde, 0x78 e 0xcc, no endereço 0x0008f002. Admita que o *target* introduz um estado de espera em cada fase de dados.

Desenhe os diagramas temporais para estas transacções. Represente a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal ou binário.

6. (1 valor)

Relativamente ao mecanismo de transferência directa de dados entre periféricos e a memória,

a) (0,5) Apresente a motivação para a existência deste mecanismo.

b) (0,5) Um periférico localizado numa placa de expansão PCI pode utilizar transferências deste tipo? Se não, porquê? Se sim, que características deve ter essa placa?

7. (3 valores)

Relativamente aos *hubs* USB, descreva, justificando,

a) (0,5) Qual é a sua importância no sucesso desta interface;

b) (1) Qual é a limitação no número de *hubs* ligados em cadeia e qual é a razão dessa limitação;

c) (1) Que funções desempenham no processo de ligação de *devices*;

d) (0,5) Em que circunstâncias necessitam de alimentação externa.

8. (4 valores)

Considere as seguintes definições:

```
typedef struct {  
    void * context;  
    int (*func)(void * ctx, int n, ...);  
} Function;
```

```
int TestAll(Function * f[], int nfuncs, int nwords, ...);
```

Implemente, em *assembly* IA-32, a função `TestAll`, que, para cada um dos `nfuncs` elementos de `f`, chama a função indicada pelo respectivo campo `func`, passando como argumentos o valor do campo `context`, o valor de `nwords` e as `nwords` *words* recebidas por `TestAll` na lista de argumentos variável. A função `TestAll` retorna a soma dos valores retornados pelas várias chamadas.

2007.03.30

Mário Simões, João Trindade

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
1.º Teste – 2006/2007, 2.º semestre lectivo

1. [5 valores]

Nos processadores intel com endereçamento e dados a 32 bit, há 30 linhas de endereço (A2 a A31) e 4 sinais *byte enable* (BE0# a BE3#).

- a) [1] Descreva o significado dos sinais BE0# a BE3#.
- b) [1] O processador pode aceder a palavras com 1, 2 ou 4 bytes, em qualquer endereço, desdobrando automaticamente o acesso se necessário. Indique quais as configurações possíveis dos sinais BE0# a BE3#.
- c) [2] Escreva a tabela de verdade para determinar os valores de A0 e A1 a partir dos sinais BE0# a BE3#.
- d) [1] A arquitectura PC inclui circuitos de interface para ligação de periféricos e memórias a 8 bits. No caso de processadores com dados a 32 bits, indique o número de registos *latch* de 8 bits pertencentes a esta interface e descreva as circunstâncias em que estes registos são utilizados.

2. [6 valores]

Relativamente à ligação de DRAMs em *interleaving*, descreva

- a) [2] O seu princípio de funcionamento;
- b) [2] A motivação para o uso deste tipo de organização;
- c) [2] As implicações que tem na escolha dos módulos de DRAM a utilizar. Exemplifique para o caso de instalar 512 Mbyte de DRAM num sistema com bus de dados a 64 bits.

3. [2 valores]

Na evolução das DRAMs, o aumento de dimensão do *chip*, em *bits*, implica o aumento do tempo de retenção da informação em cada célula, na ausência de refrescamento. Descreva os motivos desta regra.

4. [4 valores]

Nos processadores Intel baseados na microarquitECTURA Core, há uma versão em que a cache de nível 2 é associativa de 8 vias com 2 Mbyte dados em linhas de 64 byte.

- a) [2] Considerando que o endereçamento destes processadores é a 36 bit, determine o número de linhas e os bits de endereço que definem os campos BYTE OFFSET, SET (INDEX) e TAG.
- b) [1] Noutra versão, a RAM de dados da *cache* tem 4 Mbyte. Indique as opções possíveis relativamente aos parâmetros da cache a alterar para ter esta dimensão.
- c) [1] Apresente as razões que no seu entender motivaram o fabricante a optar por duplicar o número de vias.

5. [3 valores]

Os processadores Intel baseados na microarquitECTURA Core, incluem no mesmo *chip* duas *caches* de primeiro nível (L1), de código e dados, e uma *cache* de segundo nível (L2), genérica. A comunicação entre a *cache* L1 de dados e a *cache* L2 é realizada através de um *bus* com 256 bit de dados.

- a) [1] Qual é a motivação?
- b) [2] Porque não se utiliza a mesma dimensão de *bus* entre a *cache* L1 de dados e o *core* de CPU nem entre a *cache* L2 e o exterior?

2007.05.03
Mário Simões, Pedro Sampaio

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
2.º Teste – 2006/2007, 2.º semestre lectivo

1. [2 valores]

Relativamente à arbitragem do bus PCI,

- a) [1] Por que razão, para um *master* iniciar uma transacção, não basta ter o sinal GNT# mas é também necessário observar os sinais FRAME# e IRDY# inactivos?
- b) [1] Se o sistema fosse concebido de modo que bastasse GNT# activo para o *master* iniciar a transacção, que implicações teria essa opção?

2. [1 valor]

Um *master* PCI que usa os comandos *Memory Read Line*, *Memory Read Multiple* e *Memory Write and Invalidate* tem que disponibilizar, num registo de configuração, a dimensão da linha de *cache*. Porquê?

3. [2 valores]

Explique a motivação para o PCI suportar a inserção de estados de espera, o significado dos sinais IRDY# e TRDY# e a condição para que se conclua uma fase de dados.

4. [5 valores]

Considere um bus PCI a 32 *bit* com dois *masters*, designados por A e B, com um esquema de prioridades fixas, sendo o *master* A mais prioritário:

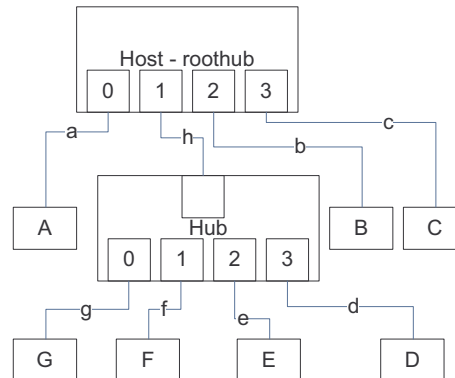
- i) O *master* A pretende realizar a escrita de um bloco de dados com 3 *bytes* com os valores 0x11, 0x66, 0xff localizado no endereço de memória 0x002343a1. Admita que o *target* introduz um estado de espera em cada fase de dados.
- ii) O *master* B pretende realizar uma leitura de dois blocos de dados. Um com 1 *byte* com o valor 11h localizado no endereço de memória 0xa0000bdc; o outro com 6 *bytes* com os valores 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, localizado no endereço de memória 0x08acc2dc.

Considerando que o *master* A manifesta a intenção de realizar a sua transacção quando o *master* B já está a realizar a primeira transacção, desenhe os diagramas temporais para estas transacções. Represente a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal ou binário.

5. [4 valores]

Considere as ligações USB representadas na figura. Admita que o *Host* atribui endereços a partir de 1 e que pesquisa os portos *downstream* dos *hubs* pela ordem crescente do seu número.

- a) [2] Suponha que os cabos foram ligados pela ordem a, b, c, d, e, f, g, h. Indique o endereço atribuído a cada um dos dispositivos.
- b) [2] O *hub* respeita a norma USB 2.0 e os restantes dispositivos pertencem às classes de rapidez: A, G e D, *Low-speed*; B e F, *Full-speed*; C e E, *High-speed*. Indique, justificando, a classe de rapidez utilizada em cada uma das ligações. Considerando que os *hubs* difundem as tramas, recebidas do *host*, para todos os portos *downstream*, excepto para os de classe de rapidez inferior, indique as ligações em que são difundidas as tramas destinadas a cada um dos dispositivos.



6. [3 valores]

Considere um equipamento multifunções, de interface USB, com *scanner*, impressora, botões de controlo e leitor de cartões de memória. Quando este é o único periférico ligado, o *host* identifica a presença de vários dispositivos independentes.

- a) [1] O que significa isso, relativamente à construção deste equipamento?
- b) [1] Se estiver ligado através de uma cadeia de *hubs*, qual é o número máximo de *hubs* entre o equipamento e o *host*. Justifique.
- c) [1] Caracterize os *endpoints* pertencentes a cada uma das funcionalidades apresentadas. Justifique as características que considerar.

7. [1 valores]

Sabendo que os componentes dos periféricos USB funcionam, na maioria dos casos, a 3,3 V, indique a razão por que a alimentação fornecida pelas fichas do *host* e dos *hubs* é a 5 V.

8. [2 valores]

A comunicação de dados entre o *host* e um periférico USB envolve sequências IN-DATA-ACK ou OUT-DATA-ACK. Em função da dimensão, os dados podem ser fraccionados em blocos, dando origem a várias sequências destas.

- a) [1] A dimensão destes blocos é estabelecida por uma característica do *host* ou do *device*? Que característica é essa?
- b) [1] As comunicações do *host* com outros periféricos podem ocorrer em qualquer instante (interrompendo eventualmente uma sequência, IN-DATA-ACK ou OUT-DATA-ACK), entre sequências ou somente após a totalidade das sequências? Justifique.

2007.06.09

Mário Simões, Pedro Sampaio

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 1.ª Época – 2006/2007, 2.º semestre lectivo

Esta prova é constituída por duas partes e destina-se à repetição de um dos testes parciais ou à realização de teste total. Quem realizar repetição de um teste responde somente a uma das partes e termina após 1 hora e 30 minutos. Quem realizar teste total termina após 3 horas; a classificação é calculada pela média aritmética das duas partes.

Parte I (ou repetição do 1.º teste)

1. [3 valores]

Na evolução do PC-AT com o processador 286 para o PC-AT com o processador 386, o barramento de dados passou de 16 para 32 bits, tendo no entanto sido garantida a compatibilidade de *software* no acesso aos periféricos. Enumere as vantagens e desvantagens desta solução.

2. [3 valores]

Os ciclos de leitura e de escrita da DRAM incluem sempre uma fase de refrescamento. No caso da escrita, tendo em conta que o valor refrescado na célula acedida é esmagado pelo valor escrito, por que motivo é realizado o refrescamento?

3. [8 valores]

Admita a definição dos campos do endereço, para acesso a uma *cache* L2, como é mostrado na figura

A35	A18	A17	A6	A5	A0
TAG		SET (INDEX)		BYTE OFFSET	

- a) [2] Para a *cache* com estas definições, qual é a dimensão total mínima da memória de dados? Caracterize este sistema de *cache*, relativamente a número de vias, dimensão de cada via e dimensão da linha.
- b) [2] Mantendo os campos idênticos, duplicou-se o número de vias. Caracterize o novo sistema de *cache*.
- c) [2] Enumere as vantagens e desvantagens da solução apresentada na alínea anterior.
- d) [2] Indique a relação que existe entre: dimensão da TAG e a dimensão da memória de dados; dimensão do SET (INDEX) e a dimensão da memória de dados.

4. [6 valores]

Considerando que existem no mercado circuitos integrados de RAM estática com dimensões relativamente grandes, é possível conceber um computador cuja memória principal seja RAM estática em vez de RAM dinâmica.

- a) [2] Que alterações resultariam dessa opção no comportamento da memória principal?
- b) [2] Quais são as implicações dessa opção em relação à utilidade da *cache* existente nos processadores actuais?
- b) [2] Porque não se encontram no mercado computadores construídos dessa forma?

Parte II (ou repetição do 2.º teste)

5. [2 valores]

Na arbitragem entre dois *masters*, o árbitro pode desactivar um GNT# e activar o outro no mesmo ciclo de clock ou em ciclos consecutivos

- a) [1] Em que situações ocorre cada um dos dois casos?
- b) [1] Que motivo justifica a existência destas duas possibilidades?

6. [2 valores]

O barramento PCI sendo um barramento *master/slave*, suporta o conceito de *multi-master*. Que importância tem esta característica, quando a arquitectura que suporta o barramento inclui, tipicamente, um único processador?

7. [6 valores]

Relativamente à situação designada por *turn-around* no bus PCI,

- a) [2] Descreva o que significa, porque ocorre e que consequências tem na evolução temporal dos sinais.
- b) [2] Enumere as situações em que ocorre. Justifique.
- c) [2] A frequência de *turn-around* nos sinais AD é inferior, igual ou superior à dos sinais C/BE#? Justifique.

8. [4 valores]

Relativamente ao USB, indique:

- a) [1] Enumere os tipos de transacções existentes;
- b) [2] Das transacções enumeradas na resposta à alínea a), qual ou quais as que têm detecção de erro? Porquê?
- c) [1] Das transacções enumeradas na resposta à alínea a), indique qual ou quais as que não são iniciadas pelo *host*. Justifique.

9. [4 valores]

Sabendo que existem duas linhas, D+ e D-, para a comunicação de dados no barramento USB,

- a) [2] Justifique o porquê da existência destas duas linhas.
- b) [2] Explique, resumidamente, como é possível a comunicação *upstream* e *downstream*.

10. [2 valores]

Comente a seguinte afirmação:

“Os dispositivos USB quando se ligam têm sempre o endereço zero. No entanto, mesmo que dois dispositivos sejam ligados em simultâneo nunca existe conflito.”

2007.06.22

Mário Simões, Pedro Sampaio

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 2.^a Época – 2006/2007, 2.º semestre lectivo

1. (x valores)

No contexto das optimizações realizadas no funcionamento das DRAMs, designadas por *page mode*, *hyper page mode* (EDO) e *interleaving*, indique para cada caso, justificando, se é aplicável a frase “... é mais rápido porque faz o acesso enquanto está a decorrer o acesso anterior.”.

2. (x valores)

Relativamente às DRAMs, o aumento da dimensão em bits tem como contrapartida o aumento da frequência de refrescamento:

- a) [x] Este aumento de frequência é motivado por quê? De que forma se relaciona com o aumento da dimensão?
- b) [x] Com o objectivo de minimizar o aumento da frequência de refrescamento, aumenta-se a capacidade dos condensadores que retêm a informação nas células de memória dinâmica. Porquê? Que consequências (vantagens e desvantagens) resultam desta solução.

3. (x valores)

Admita a existência de um sistema de cache L2 2-way set associative, com as seguintes características:

TAG: 15 bits

INDEX(SET): 12 bits

BYTE OFFSET: 5 bits

- a) [x] Determine a dimensão da memória de dados.
- b) [x] Determine a capacidade (física) de endereçamento do sistema.
- c) [x] Mantendo os campos TAG, INDEX, BYTE OFFSET, retirou-se uma way (via). Determine a nova dimensão da memória de dados.
- d) [x] Mantendo a memória de dados com a mesma dimensão determinada na alínea a), retirou-se uma way (via). Determine, em bits, a dimensão dos campos TAG, INDEX e BYTE OFFSET.
- e) [x] Indique, justificando, relativamente aos três sistemas de cache, o inicial, o determinado na alínea c) e o determinado na alínea d), por qual optava.

4. (x valores)

Considere um sistema com *bus* PCI a 32 bit com três *masters*, designados por A, B e C, com um esquema de prioridades fixas em que *master* A é o mais prioritário e o *master* C o menos prioritário.

O *master* A pretende fazer uma transacção:

i) Escrita de 2 *bytes* com os valores 0x33 e 0xef da memória com o endereço inicial 0x0000ffff; Admita que o *target* introduz um estado de espera em cada fase de dados.

O *master* B pretende fazer uma transacção:

ii) Leitura de 4 *bytes* com os valores 0x46, 0xa3, 0xf7 e 0xcc da memória com o endereço inicial 0x0900010c; Admita que o *master* introduz um estado de espera;

O *master* C pretende fazer uma transacção:

iii) Escrita de 3 *bytes* com o valor 0x1d, 0x32 e 0x40, na memória com o endereço inicial 0x01111111.

Supondo que os *masters* A e C pedem acesso ao *bus* após o *master* B tomar posse do mesmo para realizar a transacção, desenhe os diagramas temporais das transacções apresentadas. Represente a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal.

5. (x valores)

- a) [x] Indique os pacotes que constituem as diferentes transferências no barramento USB.
- b) [x] Relativamente aos pacotes indicados na alínea anterior, indique, justificando, quais os que são “formatados” pelo *host*.
- c) [x] Dos pacotes que não são formatados pelo *host*, indique, justificando, qual o elemento que é responsável por estes.

6. (x valores)

Da especificação USB, justifique:

- a) [x] A obrigatoriedade de os dispositivos quando se ligam terem o endereço 0 e suportarem o *endpoint* 0 de entrada e saída.
- b) [x] A impossibilidade de existir comunicação *full-duplex* entre o *host* e os *devices*.
- c) [x] A obrigatoriedade de um dispositivo USB 2.0 *High-speed* implementar a velocidade *Full-Speed*.

7. (x valores)

Comente a seguinte afirmação: “Um *hub* USB é visto pelo *host* como um *device*.”

2007.07.06

Mário Simões, Pedro Sampaio

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação escrita de época especial, semestre lectivo 2006/07-2

1. (2 valores)

Os computadores PC de todas as gerações têm internamente um bus com dados a 8 bits, mesmo que não disponham de fichas para inserção de placas, como é o caso das gerações actualmente no mercado.

- a) [1] Para que serve esse bus interno e quais são as razões para que ele continue a existir?
- b) [0,5] Como é suportado este bus, no caso de máquinas cujo processador tem dados a 32 ou 64 bits?
- c) [0,5] Quais são as implicações, para o funcionamento deste bus, da enorme diferença de frequência de relógio entre os processadores dos PC mais antigos e dos actuais?

2. (2,5 valores)

Relativamente à organização das memórias DRAM, descreva:

- a) [0,5] A motivação para se usar uma estrutura em matriz;
- b) [1] Os nomes, significado e funcionalidade dos pinos típicos de um circuito integrado, agrupando-os em endereço, controlo, dados e alimentação;
- c) [1] A importância dos modos de acesso por página e a sua relação com a escolha dos bits de endereço, de maior ou menor peso, para a selecção de linha ou coluna.

3. (1,5 valores)

Relativamente ao mecanismo de *precharge* da DRAM, explique:

- a) [0,5] A razão da sua existência;
- b) [0,5] O seu funcionamento;
- c) [0,5] A sua relação com os tempos de acesso.

4. (1 valor)

No contexto dos sistemas de cache *n-way set associative*, indique, justificando, se é verdadeira ou falsa a afirmação: “para carregar um novo conteúdo, pode ocorrer a necessidade de substituir uma linha preenchida, ainda que outras linhas estejam vazias”

5. (3 valores)

Considerando uma cache de nível 2 com 256 kbyte de dados em linhas de 64 bytes, com organização *4-way set-associative*, num computador com endereçamento a 32 bits, determine:

- a) [1] Os bits de endereço que formam os campos *byte-offset*, *set* e *tag*;
- b) [1] A dimensão total, em bits, da memória de gestão que inclui os campos de *Tag*, *Valid* e *Modified* (admitindo que a política de escrita é *write-back*);
- c) [1] As alterações da memória de gestão se a organização fosse alterada para 8 vias, mantendo a dimensão total da memória de dados e o comprimento das linhas.

6. (2 valores)

Considere um *bus* PCI a 32 *bits* com dois *masters*, designados por A e B, com um esquema de prioridade fixa, sendo o *master* A o mais prioritário.

- i) O *master* A pretende realizar uma transacção de leitura de memória de oito *bytes* com os valores 0x12, 0x34, 0x56, 0x78, 0x11, 0x22, 0x33 e 0x44, do endereço 0x00010002. Admita que o *target* introduz dois estados de espera em cada fase de dados.
- ii) O *master* B pretende realizar uma transacção de escrita em memória, de um *byte* com o valor 0x5a no endereço 0x0008f001.

Supondo que os dois *masters* activam REQ# ao mesmo tempo, desenhe os diagramas temporais para estas transacções. Represente a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal ou binário.

7. (2 valores)

Relativamente ao sistema de interrupções do *bus* PCI,

- a) [1] Descreva a motivação para as linhas de interrupção terem sensibilidade a nível e *active-low*;
- b) [1] Considere um *bus* PCI e com quatro fichas que dispõe de quatro linhas de interrupção. Desenhe as ligações das linhas de interrupção às fichas, de modo que: se em todas as fichas forem inseridas placas e as placas usarem o mesmo número de linhas, as interrupções sejam uniformemente distribuídas pelas quatro linhas do sistema.

8. (3 valores)

Relativamente ao funcionamento do USB, descreva:

- a) [1] Os tipos de *endpoint* existentes e a sua relação com as características do periférico;
- b) [1] A importância do armazenamento realizado pelos buffers de *endpoint*;
- c) [0,5] A relação entre as características do buffer de *endpoint* e a dimensão das tramas utilizadas na comunicação;
- d) [0,5] A estratégia seguida pelo processo de enumeração para determinar a dimensão do buffer de *endpoint* de controlo do periférico.

9. (3 valores)

Admita que dispõe de um computador com três portas USB, a que pretende ligar cinco periféricos apresentando, entre outras, as seguintes características: P1 – *low power, low speed*; P2 – *low power, full speed*; P3 – *low power, high speed*; P4 – *high power, full speed*; P5 – *high power, high speed*.

Considere o uso de um Hub 2.0, com a possibilidade de usar alimentação externa ou não.

- a) [1] Desenhe a ligação dos periféricos ao *host*, utilizando o *hub*; justifique a escolha dos periféricos que liga através do *hub*;
- b) [1] Identifique a classe de rapidez utilizada em cada troço do esquema desenhado;
- c) [1] Indique o endereço atribuído a cada um dos dispositivos, supondo que estavam todos ligados quando o sistema iniciou a actividade.

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
1.º Teste – 2007/2008, 1.º semestre lectivo

1. [2 valores]

Relativamente à concepção da arquitectura PC, descreva as vantagens e inconvenientes de especificar o funcionamento do *bus* de expansão com as mesmas características que os sinais de endereço, dados e controlo suportados pelo módulo de processador.

2. [6 valores]

Tendo por objectivo a construção do módulo de memória para um sistema com dados a 64 bit, admita que dispõe de *chips* de DRAM com as seguintes capacidades e organizações:

256 Mbit – 64M x 4, 32M x 8 ou 16M x 16;

512 Mbit – 128M x 4, 64M x 8 ou 32M x 16;

1 Gbit – 256M x 4 ou 128M x 8;

2 Gbit – 512M x 4.

Escolha os *chips* e desenhe os diagramas de blocos do módulo de DRAM com a capacidade de 1 Gbyte, sem detecção de erros, de modo a:

- a) [3] Minimizar o número de circuitos integrados;
- b) [3] Permitir a optimização dos tempos de acesso através da técnica de *interleaving*, minimizando o número de circuitos integrados.

3. [3 valores]

Comente a afirmação: “O tipo de ciclo de refrescamento *hidden-refresh* só é útil se o processador tiver uma duração do ciclo de leitura superior à duração mínima do ciclo de acesso da DRAM.”

4. [7 valores]

Considere um sistema com um processador “Pentium M” com endereçamento a 36 bit, dispondo de *cache* do tipo 8-way *set associative* com 2 Mbyte de RAM de dados em linhas de 64 byte.

- a) [2] Determine, em *bits*, a dimensão dos campos OFFSET, SET e TAG.
- b) [3] Sabendo que a implementação do sistema de *cache* se torna mais complexa com o aumento das *ways*, justifique o motivo de o fabricante optar por uma *cache* com a configuração indicada, em vez de uma do tipo *direct map* com 2 Mbyte de RAM de dados em linhas de 64 byte.
- d) [2] Num cenário de aumento da dimensão da *cache* do sistema, para 4 Mbyte de RAM de dados, indique justificando, a organização que adoptaria.

5. [2 valores]

Na evolução dos sistemas de *cache* tem aumentado significativamente a dimensão da memória de dados, o que permitiria o aumento do número de linhas por *way*, do número de *ways* ou da dimensão de cada linha.

No entanto, constata-se que a dimensão da linha não aumenta. Porquê?

2007.11.13

Mário Simões, Pedro Sampaio, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
2.º Teste – 2007/2008, 1.º semestre lectivo

1. (5 valores)

Escreva, em *assembly* IA-32, o código da função

```
char *strNesimaStr(const char *str1, const char *str2,  
                  int nesima);
```

que retorna o apontador para a n-ésima ocorrência do conteúdo da string *str2* na *string* *str1*. Se não existir o número de ocorrências pretendido, retorna o valor NULL.

Para localizar as ocorrências da *string* deve utilizar as funções de biblioteca

```
char * strstr(const char *cs, const char *ct);
```

que retorna o apontador para primeira ocorrência da string *ct* na string *cs*, ou NULL se não existir;

```
int strlen(const char * cs);
```

que retorna a dimensão da string *cs*.

2. (5 valores)

Dada a estrutura

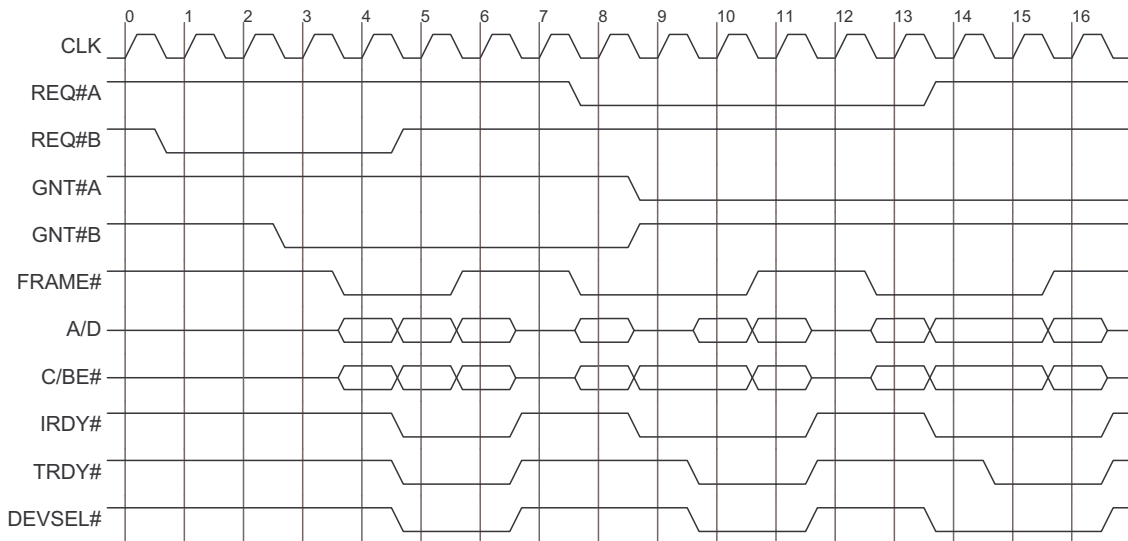
```
struct FuncDsc {  
    void (*func)(char, int);  
    int i;  
    char c;  
};
```

Escreva, em *assembly* IA-32, o código da função

```
void FuncCaller(struct FuncDsc *f);
```

que recebe um apontador para a estrutura *FuncDsc* e realiza a chamada à função *func*, com os parâmetros *i* e *c*.

3. (5 valores)



Relativamente à figura, indique:

- [2] O número e tipo (leitura ou escrita) de transacções representadas, explicitando o *master* que realiza cada uma delas.
- [2] O número do(s) ciclo(s) de relógio em que ocorre *turn-around* no barramento A/D, descrevendo a sua necessidade.
- [1] Se existem estados de espera representados. Em caso afirmativo, indique o número do(s) ciclo(s) de relógio em que ocorre(m) e o(s) dispositivo(s) que o(s) introduz(em).

4. (1 valor)

Relativamente ao barramento PCI, comente a seguinte afirmação: “Um *master*, só pode assumir que o barramento está livre quando o sinal de FRAME# e o sinal de IRDY# estiverem ambos com o valor lógico 1.”

5. (4 valores)

O barramento PCI suporta a existência de vários *masters*.

- [2] Descreva sucintamente os sinais necessários para gestão dos acessos ao bus pelos vários *masters*, referindo o significado, bem como a respectiva forma de actuação e avaliação.
- [2] Discuta o problema da monopolização do bus por parte de um *master* e descreva os critérios que o árbitro pode utilizar para a gestão das prioridades.

2008.01.11

Mário Simões, Pedro Sampaio, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 1.^a Época – 2007/2008, 1.º semestre lectivo

Esta prova é constituída por duas partes e destina-se à repetição de um dos testes parciais ou à realização de teste total. A repetição de um teste parcial, identificada por Parte I ou Parte II, tem a duração de 1 hora e 30 minutos. O teste total tem a duração de 3 horas, sendo a classificação calculada pela média aritmética das duas partes.

Parte I (ou repetição do 1.º teste)

1. [1 valor]

Na arquitectura actual do PC, a flash-ROM está ligada num barramento com dados a 8 bits. Porquê?

2. [4 valores]

A diminuição do número de transístores utilizados na implementação de células de memória DRAM relativamente às células de memória SRAM, obrigou à utilização de técnicas específicas para garantir a retenção de dados nas células e evitar que a leitura desses dados seja destrutiva.

Descreva, de forma sucinta, as técnicas adoptadas.

3. [5 valores]

O tempo de acesso de uma célula de memória dinâmica pode dividir-se em duas partes, uma inicial de preparação da leitura (*pre-charge* e fixar o endereço); a segunda, a transferência dos dados da célula de memória. Sabendo que a segunda parte está comprometida com a tecnologia disponível, verifica-se que é possível, sem alterações de tecnologia, diminuir o tempo médio de acesso. Descreva as formas de conseguir isto.

4. [4 valores]

Comente a seguinte afirmação: “A utilização da política de actualização *write-through* em *caches* permite garantir a consistência de dados no sistema de uma forma simples. Contudo, esta simplicidade traduz-se numa relativa degradação no desempenho do sistema.”

5. [6 valores]

Considere, num sistema com endereçamento a 32-bits, uma *cache* L2 do tipo associativo a 4-vias com 2 Mbyte de dados em linhas de 32 bytes.

a) [2] Determine o número de bits no barramento de endereço usados para INDEX (SET) e OFFSET.

b) [2] Determine a capacidade, em bits, da memória de TAG.

c) [2] Indique, justificando, a melhor configuração para a *cache* L1 deste sistema de forma a maximizar a sua eficiência:

i. uma *cache* associativa de 2-vias com 8 Kbyte de dados em linhas de 32 bytes;

ii. uma *cache* associativa de 4-vias com 8 Kbyte de dados em linhas de 16 bytes;

Parte II (ou repetição do 2.º teste)

6. [3 valores]

Dada a estrutura

```
struct UserDsc { long id; short number; char status; void * class; };
```

Escreva, em *assembly* IA-32, o código da função

```
void backup(struct UserDsc *src, struct UserDsc *dst, int nelem);
```

que copia o conteúdo do *array* apontado por *src*, com *nelem* elementos, para o *array* apontado por *dst*. A função devolve o número de *bytes* transferidos.

7. [6 valores]

Escreva, em *assembly* IA-32, o código da função

```
int getStrAdr(char *dst[], int max, char * sample, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de endereços de *string*, terminado por NULL, e retorna o número de *strings* com início idêntico a *sample*; preenche o *array* *dst* com os endereços dessas *strings*, limitando pelo valor de *max* o número de elementos preenchidos.

Para identificar as *strings* deve utilizar as funções de biblioteca

```
int strncmp(const char *str1, const char *str2, int count);
```

que compara os primeiros *count* caracteres de *str1* e *str2*, retornando 0 se forem iguais;

```
int strlen(const char * str);
```

que retorna a dimensão da *string* *str*.

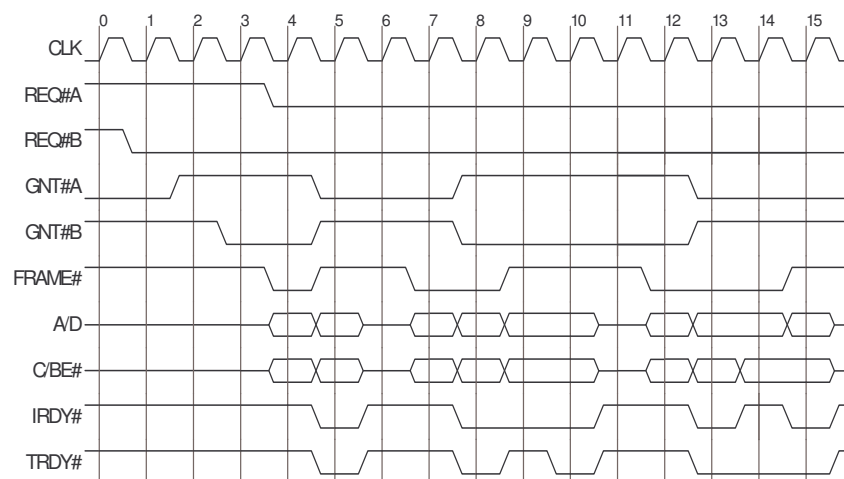
8. [4 valores]

A concepção do bus PCI permite que as linhas de interrupção provenientes de diferentes placas sejam ligadas à mesma entrada do sistema de interrupções.

a) [2] Apresente as limitações que a arquitectura do PC apresentava para que tivesse sido tomada esta opção.

b) [2] Comente as consequências de ligar muitas linhas a uma única entrada e apresente a motivação para que as placas, conforme o número de interrupções, devam usar as linhas: A; A e B; A, B e C; A, B, C e D.

9. [7 valores]



Pela análise do diagrama temporal, relativo a um bus PCI,

a) [2] Indique, justificando, os números dos ciclos de relógio em que ocorre estado de espera e qual dos elementos envolvidos o introduz;

b) [2] Comparando a segunda com a terceira transacções, descreva a diferença de comportamento do sinal *FRAME#* e indique a razão dessa diferença;c) [2] Qual é o motivo para, no início, o sinal *GNT#A* estar activo, estando *REQ#A* inactivo?

d) [1] Identifique a política de prioridades seguida pelo árbitro do bus. Justifique.

2008.01.24

Mário Simões, Pedro Sampaio, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 2.ª Época – 2007/2008, 1.º semestre lectivo

1. [1 valor]

Tendo o barramento EISA algumas características do semelhantes às do barramento PCI, nomeadamente transacções em *burst* e dados a 32 ou 64 bits, indique o que torna o barramento PCI, em média, mais rápido que o EISA.

2. [3 valores]

Em memórias DRAM, é necessário um tempo de guarda entre acessos, para leitura ou escrita, a duas posições de memória que estejam em linhas com endereços diferentes.

- a) [1] Descreva os motivos para este requisito;
- b) [1] Indique as consequências que ele tem no desempenho dos sistemas;
- c) [1] Apresente uma técnica utilizada para reduzir essas consequências.

3. [1 valores]

Comente a seguinte afirmação: “As SRAMs têm uma interface mais simples e tempos de acesso consideravelmente menores que as DRAMs. No entanto, na arquitectura do PC continua a utilizar-se DRAMs para a implementação da memória principal.”

4. [4 valores]

Considere um processador dispondo de *cache on-chip* com os seguintes blocos: L1 de código; L1 de dados; L2 genérica.

- a) [1] Que vantagens apresentam as caches L1 especializadas, face à possibilidade de ter uma *cache* L1 genérica, com o somatório da dimensão daquelas?
- b) [1] Descreva a motivação para os conteúdos das *caches* L1 serem um subconjunto dos conteúdos da *cache* L2.
- c) [2] Apresente as vantagens e inconvenientes de ter um número elevado de bits de dados nas ligações entre: O *core* de CPU e as *caches* L1; As *caches* L1 e a *cache* L2; A *cache* L2 e o exterior.

Indique, justificando, as dimensões dos *buses* de dados que considera convenientes para um processador que tem: dimensão máxima de registo de 128 bits; *caches* L1 e L2 com linhas de 32 bytes; bus externo de 64 bits.

5. [1 valor]

Considere as duas versões seguintes de um troço de código em linguagem C

Versão A

```
int i, j, s=0;
for(i=0; i<MAX_L; ++i)
    for(j=0; j<MAX_C; ++j)
        s += array[j][i];
```

Versão B

```
int i, j, s=0;
for(j=0; j<MAX_C; ++j)
    for(i=0; i<MAX_L; ++i)
        s += array[j][i];
```

Admitindo que o código é executado num computador com cache, indique, justificando, se alguma das versões é preferível.

6. [4 valores]

Escreva, em *assembly* IA-32, o código da função

```
int copyAbove(int *dst[], unsigned int limit, int n, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de n valores inteiros, copia para o *array* *dst* os que tiverem valor absoluto superior a *limit* e retorna o número de elementos copiados.

Para determinar o valor absoluto dos parâmetros deve utilizar a função de biblioteca

```
int abs(int v);
```

que retorna o valor absoluto do parâmetro *v*.

7. [1 valor]

Considere que foi executado o assembler *as*, tendo como *input* as linhas seguintes:

```
.intel_syntax noprefix
mov [ebx], 0x35
mov eax, [ecx]
mov [edi], [esi]
```

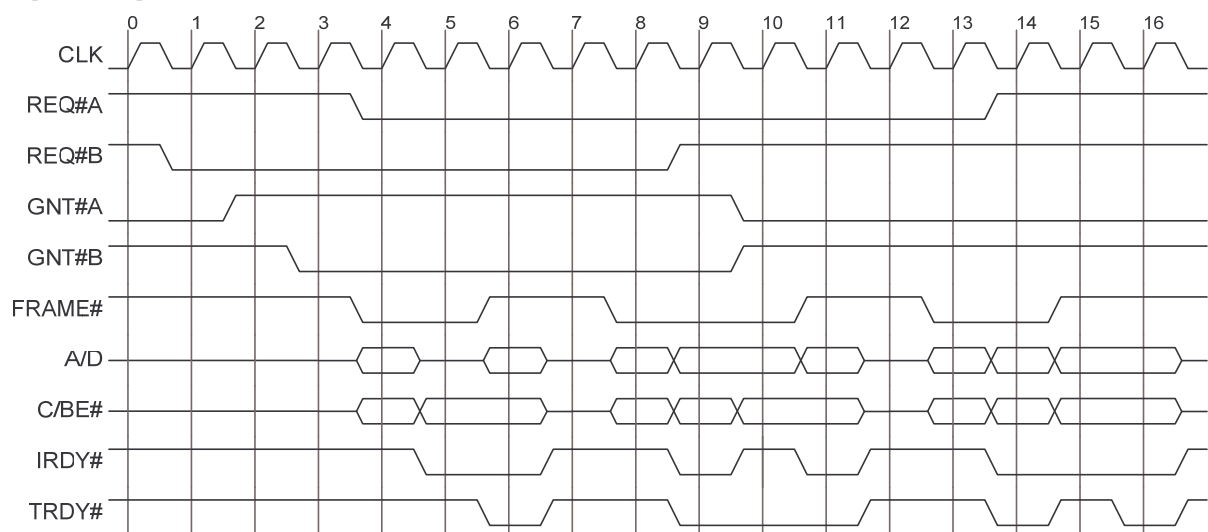
e produziu estas mensagens:

```
2: Error: ambiguous operand size for 'mov'
```

```
4: Error: too many memory references for 'mov'
```

Explique o significado, a causa e como se pode resolver cada um dos erros indicados.

8. [5 valores]



Pela análise do diagrama temporal, relativo a um bus PCI,

- [1] Indique número e tipo (leitura ou escrita) das transacções representadas, explicitando o *master* que realiza cada uma delas.
- [1] Indique, justificando, os números dos ciclos de relógio em que ocorre estado de espera e qual dos elementos envolvidos o introduz;
- [1] Comparando a segunda com a terceira transacções, descreva a diferença de comportamento dos sinais *FRAME#*, *TRDY#* e *IRDY#* e indique a razão dessa diferença;
- [1] Justifique o facto de o sinal *GNT#A*, no início do diagrama, estar activo, estando *REQ#A* inactivo?
- [1] Indique número do(s) ciclo(s) de relógio em que ocorre *turn-around* no barramento *A/D*, descrevendo a sua necessidade.

2008.02.14

Mário Simões, Pedro Sampaio, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, Época Especial – 2007/2008, 1.º semestre lectivo

1. [x valor]

Usualmente, num sistema projectado de raiz, se o processador tem barramento de dados a 16 bits, os periféricos de 8 bits são localizados somente em endereços pares; se o bus for de 32 bits, são localizados em endereços múltiplos de 4.

- a) Descreva os motivos para esta forma de organização;
- b) Justifique o facto de ela não ser seguida nas diversas evoluções da arquitectura PC e indique as consequências desta opção.

2. [x valores]

Relativamente ao mecanismo de *precharge* da DRAM,

- a) Descreva a sua necessidade e como funciona;
- b) Indique, justificando, as suas implicações no tempo de acesso e no intervalo entre acessos consecutivos.
- c) Indique uma técnica utilizada para minorar este problema, justificando a sua viabilidade.

3. [x valores]

Considere um chip de DRAM com 16 Mbit, organizado em 16 M x 1, com matriz quadrada, tempo máximo entre refrescamentos da mesma célula de 64 ms e tempo de acesso para refrescamento 128 ns.

- a) Determine o número de acessos necessários para assegurar o refrescamento completo e a percentagem de tempo que essa actividade consome no acesso à memória;
- b) Determine os mesmos parâmetros para o caso de uma DRAM, igualmente de 16 Mbit, mas organizada em 16 M x 4, mantendo matrizes quadradas e as especificações de tempo.

4. [x valores]

5. [x valores]

6. [x valores]

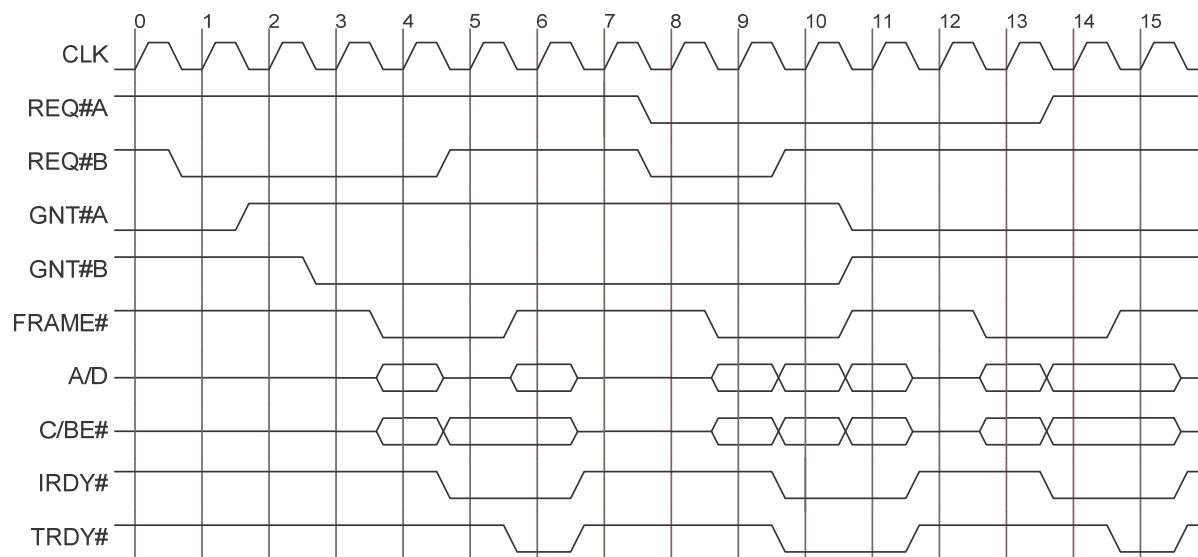
Escreva, em *assembly* da arquitectura IA-32, a função

```
void copyPlus(int dst[], ...);
```

que preenche o array `dst` com os elementos, do tipo `int`, positivos da lista de parâmetros variável que termina com o valor -1.

7. [x valores]

8. [x valores]



Pela análise do diagrama temporal, relativo a um bus PCI,

- [1] Indique o(s) número(s) e tipo (leitura ou escrita) das transacções representadas, explicitando o *master* que realiza cada uma delas.
- [1] Indique a prioridade dos *masters*.
- [1] Indique, justificando, os números dos ciclos de relógio em que ocorre estado de espera e qual dos elementos envolvidos o introduz;
- [1] Indique, justificando, porque é que a segunda transacção só se inicia no ciclo de relógio 9;
- [1] Justifique o facto de o arbitro, no início do diagrama, só activar o sinal GNT#B no ciclo de relógio 3.
- [1] Indique o número do(s) ciclo(s) de relógio em que ocorre *turn-around* no barramento A/D, descrevendo a sua necessidade.

2008.04.11

Mário Simões, Pedro Sampaio, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
1.º Teste – 2007/2008, 2.º semestre lectivo

1. [2 valores]

O chipset “Mobile Intel 965 Express” está preparado para controlar DRAM com a dimensão máxima de 64 Gbyte e suporta a definição de áreas reservadas correspondentes aos primeiros: 1 Mbyte; 16Mbyte; 4Gbyte.

Explique a motivação para a definição destas áreas e justifique as dimensões indicadas.

2. [6 valores]

Relativamente à tecnologia de RAM dinâmica,

- a) [2] Apresente as razões que motivam o funcionamento multiplexado dos pinos de endereço.
- b) [2] Descreva, justificando, os critérios para definir quais os bits de endereço do sistema que devem ser usados para endereço de linha e de coluna.
- c) [2] Justifique a organização das células de memória em matriz quadrada ou rectangular com mais colunas que linhas.

3. [2 valores]

Comente a afirmação:

“Relativamente ao projecto de um circuito integrado de DRAM, com uma determinada capacidade em bits, o aumento do número de pinos de dados torna o refrescamento mais eficiente.”

4. [6 valores]

Considere um sistema com um processador “Pentium M” com endereçamento a 36 bit, dispondo de caches L1 com 32 kbyte para código e 32 kbyte para dados, ambas do tipo 4-way *set associative*, e de cache L2, unificada, do tipo 8-way *set associative* com 1 Mbyte em linhas de 64 byte.

- a) [2] Indique, justificando, a dimensão das linhas das caches L1.
- b) [2] Determine, em bits, a capacidade da memória de TAG da cache de dados L1.
- c) [2] Sabendo-se que um maior grau de associatividade permite obter *hit-rates* mais elevados, justifique a organização adoptada para a cache de dados L1 comparativamente com a organização da cache L2.

5. [4 valores]

Considerando um controlador de cache com funcionamento em modo *write-back* e uma política de substituição *random*, descreva as operações desencadeadas por um *miss* de escrita, admitindo os seguintes cenários para a linha a substituir: i) está inválida; ii) está válida não modificada; iii) está válida e modificada.

2008.04.22
Mário Simões, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
2.º Teste – 2007/2008, 2.º semestre lectivo

1. (2 valores)

Considerando as definições

```
typedef struct {int a, int b} sX;  
typedef struct {int a, int b, int c} sY;  
  
sX fX(int x);  
sY fY(int y);
```

Indique, justificando, quantos parâmetros recebe em *stack* cada uma destas funções.

2. (3 valores)

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
int charinrange(char c, char first, char last);
```

que devolve 1 se valor de *c* está compreendido no intervalo de *first* a *last*, inclusive; caso contrário, devolve 0.

3. (6 valores)

Considerando a definição

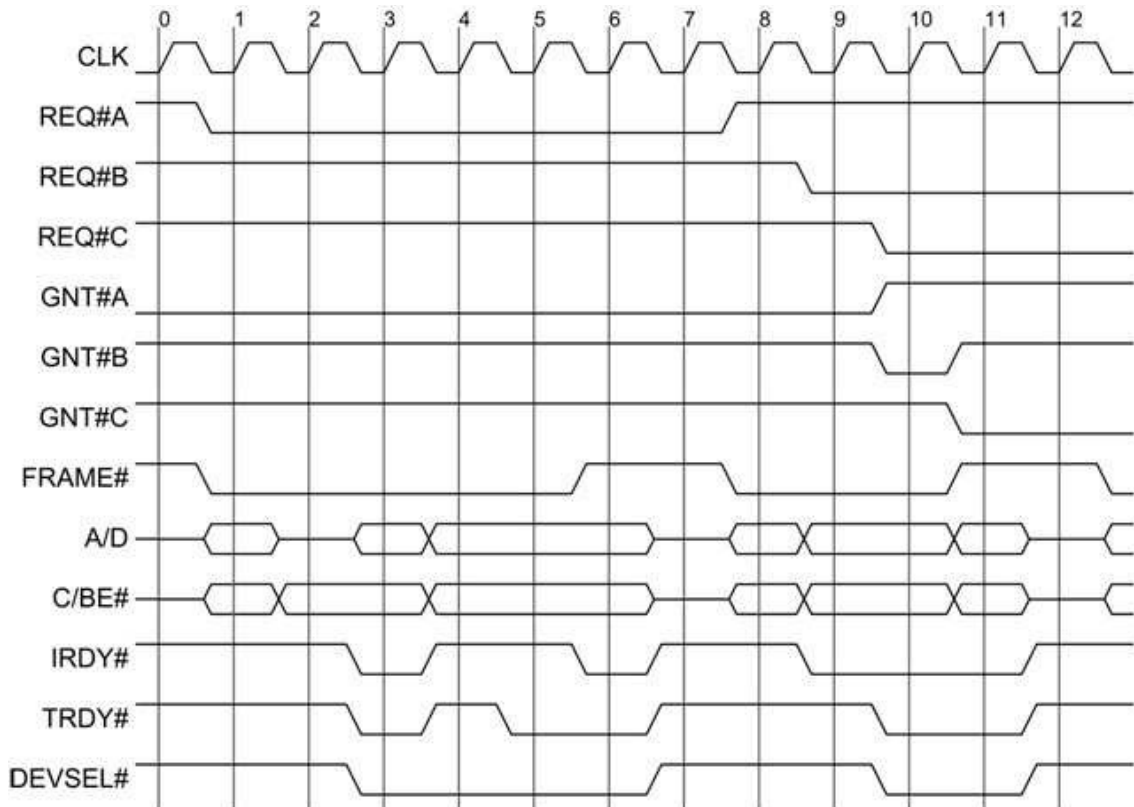
```
typedef struct {int lower, int upper} charcount;
```

escreva, em *assembly* da arquitectura IA-32, o código da função

```
charcount charstat(...);
```

que recebe na lista de parâmetros variável, terminada por NULL, os endereços de várias *strings* e devolve, nos campos *lower* e *upper*, respectivamente, o número total de minúsculas e maiúsculas existentes nessas *strings*. Para determinar a gama a que pertence cada carácter, deve utilizar a função *charinrange*.

4. (7 valores)



Relativamente à figura, indique, justificando,

- a) [1] O número e tipo (leitura ou escrita) de transacções representadas, explicitando o *master* que realiza cada uma delas.
- b) [2] O motivo de o sinal REQ#A ser activado no início do diagrama, estando o sinal GNT#A activo.
- c) [1] O número de estados de espera representados, explicitando o número do(s) ciclo(s) de relógio em que ocorre(m) e o(s) dispositivo(s) que os introduz(em).
- d) [2] O número do(s) ciclo(s) de relógio em que ocorre *turn-around* no barramento A/D, descrevendo a sua necessidade.
- e) [1] O *master* que realiza a transacção iniciada no ciclo de relógio 12.

5. (2 valores)

Comente a seguinte afirmação: “No barramento PCI, a activação do sinal GNT# a um *master* não garante que este dispositivo inicie imediatamente uma transacção.”

2008.06.03

Mário Simões, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 1.^a Época – 2007/2008, 2.º semestre lectivo

Esta prova é constituída por duas partes e destina-se à repetição de um dos testes parciais ou à realização de teste total. A repetição de um teste parcial, identificada por Parte I ou Parte II, tem a duração de 1 hora e 30 minutos. O teste total tem a duração de 3 horas, sendo a classificação calculada pela média aritmética das duas partes.

Parte I (ou repetição do 1.º teste)

1. [4 valores]

Na evolução da arquitectura do PC até à gama de processadores Pentium o acesso à DRAM passou progressivamente de 8 para 64 bits.

- a) [2] Indique as vantagens e inconvenientes desta situação.
- b) [2] Explique como é que os processadores Pentium tiram partido de um barramento de dados a 64 bits, sabendo que os seus registos e unidades aritméticas e lógicas são a 32 bits.

2. [7 valores]

Nas várias gerações da arquitectura do PC, a memória principal do sistema tem sido sempre implementada com DRAMs.

- a) [2] Sabendo que a SRAM proporciona melhores desempenhos, justifique a opção pela DRAM.
- b) [2] Indique os principais inconvenientes desta opção, fundamentando os seus motivos, e apresente soluções para minorar os seus efeitos.
- c) [3] Comente a afirmação: “Se nas DRAM o sistema de endereçamento não fosse multiplexado seria possível reduzir significativamente o tempo de acesso a estas memórias.”

3. [9 valores]

Considere um sistema com endereçamento a 32 bits e apenas um nível de cache, do tipo *4-way set associative*, com 8 kB em linhas de 16 byte e uma política de escrita do tipo *write-through*.

- a) [2] Determine, em bits, a dimensão dos campos OFFSET, SET e TAG.
- b) [2] Determine, em bits, a capacidade da memória de TAG.
- c) [2] Indique as principais vantagens e inconvenientes da política de escrita adoptada.
- d) [3] Com vista a aumentar o desempenho do sistema pretende-se substituir esta cache. Discuta as vantagens e desvantagens das seguintes soluções:
 - Cache do tipo *direct-mapped* com capacidade 16 kB;
 - Cache do tipo *8-way set associative* com capacidade 8 kB.

Parte II (ou repetição do 2.º teste)

4. [2 valores]

Considere que foi executado o assembler *as*, tendo como *input* as linhas seguintes:

```
.intel_syntax noprefix  
add eax, byte ptr [ebx]  
add ebx, 1
```

e produziu esta mensagem:

```
2: Warning: using `%al' instead of `%eax' due to `b' suffix
```

Explique o significado, a causa e como se pode resolver o problema indicado, considerando que se pretende efectivamente adicionar um byte ao registo EAX.

5. [3 valores]

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
int accumulate(short val, int sum);
```

que devolve a soma de *val* com *sum*.

6. [7 valores]

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
int zeroMedian(short a[], int n);
```

que determina a média aritmética dos *n* elementos do array e subtrai esse valor a cada um dos elementos, de modo que, no final, a média dos elementos seja 0. A função retorna a média aritmética calculada.

Como auxiliar para o cálculo da média, deve utilizar a função *accumulate*.

7. [6 valores]

Considere que num barramento PCI estão ligados dois *masters* A e B, em que A é mais prioritário que B, e um *target* C, na gama de endereços 0x00070000 a 0x0007ffff que necessita sempre de dois ciclos de relógio para completar uma fase de dados. Desenhe o diagrama temporal referente às seguintes transacções:

- o *master* A pretende escrever a palavra 0x000AC22007200802 no endereço 0x00078100;
- o *master* B pretende ler uma palavra de 16 bits, que tem o valor 0x55aa, do endereço 0x0007ffc0;
- a necessidade de fazer estas transacções ocorre em simultâneo para os dois *masters*, num ciclo de relógio em que não está nenhuma transacção a decorrer, havendo *parking* sobre o *master* B.

8. [2 valores]

Comente a afirmação: “A introdução do barramento PCI na arquitectura do PC permitiu aumentar significativamente as taxas de transferência de dados com os periféricos, garantindo contudo o acesso aos periféricos mais antigos.”

2008.07.09

Mário Simões, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, 2.^a Época – 2007/2008, 2.º semestre lectivo

1. [1 valor]

Nos processadores Intel com endereçamento e dados a 32 bit, há 30 linhas de endereço (A2 a A31) e 4 sinais *byte enable* (BE0# a BE3#), estando os dados organizados em grupos: D0 a D7; D8 a D15; D16 a D23; D24 a D31.

- a) [0,5] Descreva o motivo de agrupar os dados desta forma e a relação de cada um dos grupos com os sinais BE0# a BE3#.
- b) [0,5] Considere que o processador faz um conjunto de acessos a palavras com um *byte* apenas. Indique, justificando, qual é o agrupamento do *bus* de dados usado quando o endereço acedido termina em 0, 1, 2 ou 3.

2. [3 valores]

Um fabricante de DRAMs disponibiliza circuitos integrados da geração DDR SDRAM, com a capacidade de 512 Mbit, em três modelos: (A) 32 M x 16; (B) 64 M x 8; (C) 128 M x 4. Todos eles são organizados internamente em 4 bancos e necessitam de 8192 ciclos para realizar o refrescamento completo.

- a) [1] Determine o número total de matrizes de células DRAM existentes em cada um dos modelos. Apresente todos os cálculos.
- b) [1] Determine a geometria das matrizes e o número total de *sense-amplifiers* de cada um dos modelos. Apresente todos os cálculos.
- c) [1] Considerando que se pretende construir módulos DIMM com dados a 64 bit, apresente, justificando, a motivação para escolher cada um destes modelos.

3. [1 valor]

Comente a afirmação: “Na maioria dos chipsets para PC, o controlador de DRAM permite melhores tempos de acesso se os módulos de memória usados formarem pares, com a mesma dimensão”.

4. [4 valores]

Considere um processador com endereçamento a 36 bit, dispondo de cache interna com dois níveis. A cache L2 é unificada, do tipo 8-way *set associative*, com 1 Mbyte de dados em linhas de 64 bytes, com políticas de escrita *write-back* e de substituição *random*.

- a) [1] Determine a dimensão total, em bits, da memória necessária para esta cache, incluindo dados, *tags*, bits V (*valid*) e M (*modified*). Apresente todos os cálculos, devidamente identificados.
- b) [1] Determine, em percentagem, o acréscimo de memória necessário para duplicar o número de vias. Apresente todos os cálculos, devidamente identificados.
- c) [1] Enumere, justificando, os critérios a considerar para definir o número de vias, no projecto da cache, e indique a sua relevância relativa.
- d) [1] Discuta as vantagens e inconvenientes de usar a política de substituição indicada.

5. [1 valor]

Justifique o facto de, nas caches do tipo *n-way set associative*, os bits de endereço usados para *tag* serem os de maior peso.

6. [1,5 valores]

Relativamente à preservação de registos nas chamadas a funções:

- a) [0,5] Indique, justificando, os registos cuja preservação é indispensável;
- b) [0,5] Indique os registos, além dos anteriores, que são preservados nas convenções adoptadas pelas ferramentas GNU, descrevendo as vantagens de fazer essa preservação;
- c) [0,5] Justifique o facto de os restantes registos não serem preservados.

7. [2 valores]

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
int limitedCopy(char * d, char * s, int dSize);
```

que copia a *string* *s* para o endereço *d*, limitando a *dSize* o número de caracteres copiados, incluindo o terminador de *string*. Devolve 1 se copiou a totalidade da *string* ou 0, caso contrário.

8. [3 valores]

Escreva, em *assembly* da arquitectura IA-32, o código da função

```
int limitedConcat(char * dst, int dstSize, ...);
```

que recebe na lista de parâmetros variável, terminada pelo *pointer* NULL, um conjunto de *strings* e copia o seu conteúdo, pela ordem em que estão na lista, para o endereço *dst*, limitando a *dstSize* o número de caracteres copiados, incluindo o terminador de *string*. Devolve 1 se copiou a totalidade das *strings* ou 0, caso contrário.

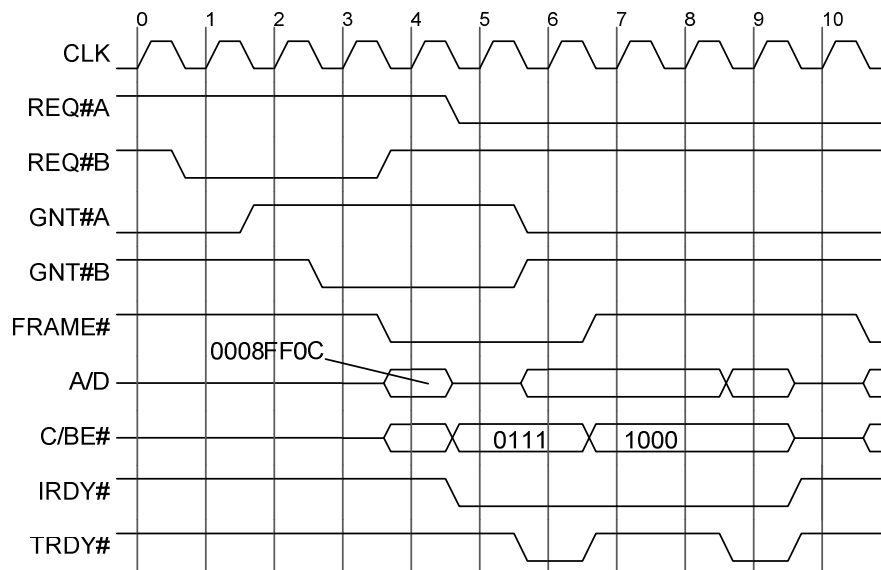
Para executar o preenchimento de *dst* deve usar a função `limitedCopy()`, definida na pergunta anterior.

9. [1,5 valores]

Relativamente ao controlo das transacções no bus PCI,

- [0,5] Justifique o facto de FRAME# ser desactivado na última fase de dados e não no final da transacção.
- [0,5] Porque não basta a um *master* receber GNT# para iniciar uma transacção, sendo necessário detectar também FRAME# e IRDY# inactivos?
- [0,5] Descreva as circunstâncias em que FRAME# permanece activo durante a última fase de dados. Ilustre a descrição, desenhando o diagrama temporal de uma transacção com duas fases de dados em que ocorra este caso.

10. [2 valores]



Pela análise do diagrama temporal, relativo a um bus PCI,

- [0,5] Indique, justificando, se está representada alguma situação de *parking*;
- [0,5] Descreva e justifique a diferença de comportamento, na mudança de estado dos sinais GNT#, entre as duas situações representadas nos ciclos 1 e 5;
- [0,5] Indique o(s) número do(s) ciclo(s) de relógio em que ocorre transferência de dados;
- [0,5] Tendo em conta o endereço, indicado em hexadecimal, e os valores de BE#, em binário, determine a dimensão e o endereço da palavra transferida.

2008.07.22

Mário Simões, Tiago Dias

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
Licenciatura em Engenharia Informática e de Computadores
ARQUITECTURA DE COMPUTADORES II
Avaliação Escrita, Época Especial – 2007/2008, 2.º semestre lectivo

1. [1 valor]

Na evolução da arquitectura PC, o módulo de ROM foi implementado: com um circuito de EPROM, na geração XT; com dois circuitos de EPROM, na geração AT; com um circuito de EPROM, em gerações posteriores ao AT; com um circuito de FLASH-ROM, nas gerações mais recentes.

- a) [0,5] Apresente os motivos que levaram os fabricantes a realizar o módulo ROM com as características indicadas, nas gerações referidas.
- b) [0,5] Indique os casos em que estas características implicam alguns inconvenientes. Descreva esses inconvenientes e a forma como são resolvidos.

2. [2 valores]

Relativamente aos circuitos de memória DRAM,

- a) [1] Indique as vantagens e inconvenientes de disporem de um contador interno de endereços para refrescamento e descreva a forma de o utilizar.
- b) [1] Justifique o facto de serem mais lentos que a SRAM, tanto no tempo necessário para ler ou escrever como no intervalo mínimo entre acessos consecutivos.

3. [3 valores]

A DRAM síncrona (SDRAM, DDR, etc) é concebida para ser acedida em *burst*.

- a) [1] Descreva o que significa “ser acedida em *burst*”.
- b) [1] Apresente a relação desta característica com a organização das células de memória em matriz.
- c) [1] Apresente a relação desta característica com a existência de cache.

4. [3 valores]

Considere, associado a um processador com endereço a 36 bits, um sistema de *cache* do tipo 8-way *set associative* cuja RAM de dados tem a dimensão total de 1 Mbyte, em linhas de 32 bytes.

- a) [1] Determine quais os bits de endereço utilizados para *byte offset*, *set (index)* e *tag*. Justifique a posição relativa de cada um destes grupos de bits.
- b) [1] Determine a quantidade total, em bits, de memória para armazenamento de *tags*.
- c) [1] Descreva as vantagens e inconvenientes de modificar esta *cache*, indicando justificadamente, quais dos valores calculados nas alíneas anteriores são modificados, relativamente a duas opções:
 - i) reduzir o número de vias para 4;
 - ii) aumentar a dimensão da linha para 64 bytes.

5. [1 valor]

Considere duas alternativas para a realização de um *line fill*, motivado por *miss* na leitura:

- i) Ler a DRAM por ordem de endereço crescente;
- ii) Ler da DRAM a palavra que causou *miss* e, só depois, as partes inicial e final da linha.

Apresente as vantagens e inconvenientes destas duas possibilidades e indique, justificando, qual considera preferível.

6. [3 valores]

Escreva, em *assembly* da arquitectura IA-32, a função

```
int proper(char * s);
```

que modifica a *string* recebida por parâmetro, convertendo para maiúscula o primeiro carácter de cada palavra e para minúscula os restantes. Considere as palavras separadas pelo carácter espaço. A função devolve o número de palavras existentes. Para fazer as conversões deve utilizar as funções de biblioteca

```
int toupper(int c); // converte para maiúscula  
int tolower(int c); // converte para minúscula
```

7. [2 valores]

Escreva, em *assembly* da arquitectura IA-32, a função

```
int properList(int count, ...);
```

que recebe, na lista de parâmetros variável, um conjunto de *count* endereços de *string* e executa sobre cada uma delas a função *proper* definida na questão anterior. A função *properList* devolve o número total de palavras existentes nas *strings* processadas.

8. [5 valores]

Considere um bus PCI a 32 bit com dois *masters*, designados por A e B, com um esquema de prioridades fixas onde A é o mais prioritário:

- i) O *master* A pretende realizar duas transacções: a leitura de 2 bytes com o valor 0x75 e 0x45 com o endereço inicial 0xcab2438e; a escrita de um bloco de dados 4 bytes com os valores 0x4e, 0x44, 0xf3 e 0x3e no endereço 0x243fcafc. Admita que o *master* introduz um estado de espera na primeira transacção.
- ii) O *master* B pretende realizar a leitura de 3 bytes de memória com os valores 0x4a, 0xdf e 0xe0 com o endereço inicial 0x00fdaede. Admita que o *target* introduz um estado de espera;

Desenhe os diagramas temporais para estas transacções, representando a evolução de todos os sinais envolvidos (protocolo de arbitragem e de transferência). Os sinais A/D[31:0] e C/BE#[3:0] devem ser representados por valores em hexadecimal.

2008.10.23

Mário Simões, Pedro Sampaio