

**Arquitectura e Sistema de Computadores I**  
Eng<sup>a</sup> Informática FCT/UNL – Exame (recurso) - 5/02/2007  
Duração: 2h30. Sem consulta. **Todas as respostas devem ser justificadas**

1.

- a) Indique a representação binária correspondente ao valor hexadecimal: A45C. Considerando esta como a representação de um inteiro sem sinal, indique também em base 10 o valor representado.
- b) Calcule a representação binária do número -3, numa arquitectura de 16 bits com representação de números negativos em complemento para 2.
- c) Calcule e escreva a representação binária para o número 129,125 na norma IEEE 754 de precisão simples (lembre-se que esta tem um bit para o sinal, 8 bits para o expoente que é representado em “excesso de 127” e 23 bits para a parte fraccionária da mantissa normalizada).

2. Considere uma arquitectura de computador suportando operações aritméticas sobre inteiros de 8bits e dispondo das *flags* Carry, Overflow, Signal e Zero tal como as apresentadas nas aulas. Indique o resultado em binário e os valores das *flags* para cada uma das seguintes operações:

- a) 1100 0001 + 0100 0001
- b) 1100 0001 - 0100 0001
- c) 0100 0001 - 1100 0001

3. Supondo que se altera o CPU MARIE para que este possa suportar a chamada de subrotinas, à semelhança dos Intel 80x86/Pentium. Pretende-se suportar as instruções (em *assembly*) **call X** e **ret** que devem alterar o PC de acordo com as instruções equivalentes do Intel.

- a) Querendo suportar chamadas sucessivas de subrotinas necessita de dispor de uma pilha (*stack*). Explique porquê e indique que registo(s) seria necessário acrescentar ao CPU MARIE para suportar esta pilha.
- b) Descreva o funcionamento das instruções indicadas, em termos de transferências entre registos, memória e acessos ao *bus*, usando a notação RTL (*Register Transfer Language*) usada nas aulas teóricas.

4. Considere a seguinte função C que incrementa a variável passada por referência, sendo que esta volta a zero quando o limite indicado for atingido (admita que o tipo `int` representa inteiros com sinal em 32 bits):

```
void prox( int *n, int limite )
{
    *n = *n + 1;
    if ( *n >= limite ) *n = 0;
}
```

- a) Reescreva esta função em *assembly* NASM/Intel (IA32), utilizando a passagem de argumentos pela pilha.
- b) Escreva em *assembly* o código que efectua a chamada desta função tal como no seguinte código C (sendo `x` uma variável inteira):

```
prox( &x, 100 );
```

5. Recorde os trabalhos sobre entradas / saídas (I/O).

- a) Indique, justificando, se são verdadeiras ou falsas cada uma das seguintes afirmações:
  - a1) Usando espera activa na recepção, pode-se perder os caracteres que cheguem enquanto não se chama a rotina para receber os caracteres.
  - a2) Numa solução usando interrupções, a rotina de atendimento às interrupções pode sempre ser interrompida por quaisquer outras interrupções que entretanto ocorram.
  - a3) Numa solução usando espera activa, devemos ter um menor uso do bus de sistema no acesso ao controlador do periférico, do que na solução baseada em interrupções.
  - a4) Qualquer rotina de atendimento de interrupções, tem de terminar ligando as interrupções no controlador de interrupções (PIC) usando a instrução *assembly* STI (Set Interrupt Flag).

b) Considere que pretende enviar caracteres para uma impressora via porta paralela de um computador compatível com o IBM/PC. Admita que existe no respectivo controlador um registo de estado e outro para o byte a enviar. No registo de estado o bit 0 indica que a impressora está ocupada e o bit 1 que esta está sem papel. O registo de dados deve ser escrito para enviar um caracter para a impressora. Escreva a implementação duma rotina que seja capaz de imprimir um caracter na impressora, por exemplo:

```
void printChar( char c );
```

Note que não pode enviar nada para a impressora enquanto esta não estiver desocupada e com papel. Represente os endereços de I/O dos registos de que necessita por constantes. Pode utilizar as extensões à linguagem C e funções auxiliares usadas nas aulas práticas.

6. Nas entradas e saídas de dados com os periféricos orientados ao bloco é mais conveniente o uso de DMA (Direct Memory Acces). Descreva em que consiste esta técnica e o porquê do seu uso nestes casos.

7. Admita uma arquitectura de computador com as seguintes características:

- registos de endereços e dados de 30 bits;
- espaço de endereçamento real máximo de 128 Mbytes (endereçamento ao byte);
  - a) Qual a memória máxima que pode ser fisicamente instalada nesta arquitectura (em Mbytes) e qual a dimensão (em bits) do bus de endereços para a suportar?
  - b) Qual o maior programa que pode ser executado nesta arquitectura (em Mbytes), admitindo o suporte de memória virtual ?

8. Admita uma arquitectura de um computador com as seguintes características:

- endereços virtuais de 30 bits, bus de endereços suportando 28 bits;
- uma memória instalada de 32Mbytes;
- cache de 128KBytes, operando sobre endereços reais, com blocos de 64bytes, associativa por grupos de 4 blocos cada grupo;
- MMU suportando paginação a pedido com páginas de 4Kbytes;

Admita ainda que um programa faz referência ao seguinte endereço:

00 0000 0000 0000 1000 0000 0000 1000

a) Indique como a MMU interpreta este endereço: os bits usados para indicar o número de página e os usados para indicar o deslocamento dentro da página. Indique a página virtual endereçada.

b) Suponha que a página física onde se encontra a página referida pelo endereço anterior é a número 1. Indique o endereço real de memória que vai ser referenciado.

c) Indique como o endereço da alínea anterior é interpretado pela cache, como é usado para procurar os bytes endereçados e como é decidido se se trata de um cache *hit* ou cache *miss*. (se não fez a alínea anterior use o seguinte endereço: 0000 0000 0000 1000 0000 0000 1000)

d) Admitindo uma política de *write-back*, explique o que acontece no caso de uma escrita (*store*) em que há um cache *miss*, desde o instante em que o *miss* é detectado até à conclusão da escrita

9. Considere um processador suportando o *pipelining* das varias fases de execução das instruções.

a) Nestes processadores, o *pipelining* pode ser atrasado (*stalled*) ou despejado (*flushed*) em algumas situações. Indique as principais e porque estas são um problema.

b) Explique uma técnica que seja usada para reduzir ou eliminar uma das situações antes indicada.