

Arquitectura e Sistema de Computadores I

Eng^a Informática FCT/UNL – Exame (época normal) - 12/01/2007

Duração: 2h30. Sem consulta. **Todas as respostas devem ser justificadas**

1. Considere o seguinte representação binária (arquitecturada 32 bits):

1100 0000 0100 0000 0000 0000 0000 0000

- Interprete os 2 bytes (16bits) mais significativos como a representação de um inteiro sem sinal e apresente esse número em hexadecimal e em decimal.
- Interprete agora os 2 bytes (16bits) mais significativos como a representação, em complemento para 2, de um inteiro com sinal e apresente esse número em decimal.
- Interprete agora os 32 bits como a representação de um número real no formato de precisão simples da norma IEEE 754. Apresente o número representado na base 10. Lembre-se que na norma IEEE 754 a precisão simples corresponde a 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:

- 1100 0000 + 0100 0000
- 1100 0000 - 0100 0000
- 0100 0000 - 1100 0000

3. Supondo que se altera o CPU MARIE para que este possa suportar endereçamento baseado, à semelhança dos Intel 80x86/Pentium. Pretende-se suportar as instruções (em *assembly*) **loadB X** e **storeB X** que devem efectuar as transferências entre o registo acumulador (AC) e a memória, sendo o endereço efectivo dado por *X* mais o conteúdo dum registo base (chamado RB). 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 (admita que o tipo `int` representa inteiros com sinal em 32 bits):

```
int resto( int a, int b )
{
    while ( a >= b )
        a = a - b;
    return a;
}
```

- Reescreva esta função em *assembly* NASM/Intel (IA32), utilizando a passagem de argumentos pela pilha e o retorno no registo `eax`.
- Escreva em *assembly* o código que efectua a chamada desta função tal como no seguinte código C:

```
x = resto(5, 3);
```

5. Recorde os trabalhos sobre a porta série. Admita uma situação em que pretende implementar um programa capaz de, ao mesmo tempo, receber e enviar caracteres por essa porta.

a) Indique, justificando, se são verdadeiras ou falsas cada uma das seguintes afirmações:

- Usando espera activa, quando se pretende enviar um caracter, não se pode receber os caracteres que entretanto cheguem.
- Numa solução usando interrupções, se a rotina de atendimento à interrupção terminar sem dar o comando “end of interrupt” ao controlador de interrupções (PIC), nunca mais ocorrem interrupções da porta série.
- Se este programa usar interrupções, poderemos ter um maior uso do bus de sistema do que na solução baseada em espera activa, pois a rotina, de cada vez que for activada, vai transferir um byte (de ou para a porta série) via esse bus.
- Para que a rotina de atendimento da porta série seja chamada, temos que no inicio do programa, colocar o endereço dessa rotina numa qualquer posição livre do vector de interrupções e, depois, programar o CPU para que este chame essa rotina de cada vez que ocorrer uma interrupção da porta série.

b) Admita que todo o sistema está iniciado para gerar uma interrupção de cada vez que chegue um carácter e de cada vez que pode enviar o próximo (só existe uma interrupção para as duas situações). Escreva uma implementação da rotina de atendimento de interrupções que permita receber ou enviar um carácter via porta série, consoante o motivo da interrupção. Recorde a interface da UART em que num registo de estado o bit 0 indica se chegou um novo carácter e o bit 5 se pode enviar um novo carácter. Represente os endereços de I/O dos registos de que necessite por constantes, indicando a função de cada um. Considere que existem dois *buffers*, um para os caracteres recebidos e outro com os caracteres a enviar. Pode utilizar as extensões à linguagem C e funções auxiliares usadas nas aulas práticas.

6. Explique as principais vantagens dos controladores dos periféricos gerarem interrupções, face ao uso de ciclos da espera activa no programa, para poder operar com esses periféricos.

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

- registos de endereços e dados de 32 bits;
- bus de acesso à memória com 30 bits para endereços e 32 bits para dados;
 - a) Qual o espaço de endereçamento real e qual o de endereçamento virtual em termos do número de bits?
 - b) Qual a memória máxima que pode ser fisicamente instalada nesta arquitectura (em Mbytes)?
 - c) 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 como a descrita na pergunta anterior e ainda com as seguintes características:

- uma memória instalada de 16Mbytes, endereçada ao byte;
- cache de 512KBytes, operando sobre endereços reais e associativa pura com blocos de 64bytes;
- MMU suportando paginação a pedido com páginas de 8Kbytes;

Admita ainda que um programa faz referência ao seguinte endereço:

0000 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.
- b) Descreva como a MMU usa o TLB e a tabela de páginas para detectar se um endereço está em memória e como obtém o endereço real.
- c) 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 e a que bloco de memória tal corresponde.
- d) Indique como este endereço 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*.
- e) Admitindo uma política de *write-back*, explique o que acontece no caso de uma leitura (*load*) em que há um cache *miss* (desde o instante em que o *miss* é detectado até à conclusão da leitura).

9. Considere os processadores Intel da linha Pentium.

- a) Nestes processadores, a memória cache de nível L1 encontra-se dividida em duas caches distintas: uma para dados e outra para instruções. Explique como pode o controlo das caches no CPU distinguir entre os acessos a memória para dados e acessos a memória para instruções, para poder manter esta distinção entre caches.
- b) Nestes CPUs a cache L2 é comum a dados e instruções e aparece em diferentes tamanhos. Discuta as vantagens e desvantagens de aumentar o tamanho dessa cache L2.