

Arquitectura e Sistema de Computadores I

Eng^a Informática FCT/UNL – 1º teste - 21/10/2006

Sem consulta. Duração: 2h00

1. Considere o seguinte número binário (16 bits): 1000 0000 0000 0001

- Interprete este como a representação de um inteiro sem sinal e apresente esse número em hexadecimal e em decimal.
- Interprete agora como a representação em complemento para 2 de um inteiro com sinal e apresente esse número em decimal.
- Considere uma arquitectura de computador de 16 bits, com memória endereçada ao byte e organização *little-endian*. Supondo que este número era guardado no endereço 100, indique o conteúdo dos bytes nos endereços 100 e 101.

2. Considere uma arquitectura de computador de 8bits, suportando operações sobre inteiros 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) 0000 0010 + 0000 0011 | b) 0000 0010 - 0000 0011 |
| c) 1000 0000 + 1111 1111 | d) 1000 0000 - 1111 1111 |

3. Supondo que se pretende alterar o CPU MARIE para que este possa suportar uma pilha em memória à semelhança dos Intel 80x86/Pentium. Pretende-se suportar as operações de **push** e **pop** que devem empilhar e desempilhar respectivamente, o registo acumulador (AC).

- Indique, justificando, que registos acha necessário acrescentar ao CPU.
- Descreva o funcionamento das instruções push e pop, em termos de transferências entre registos, memória e acessos ao *bus*, usando a notação RTL (*Register Transfer Language*) usada nas aulas teóricas.

4. Apresente em *assembly* Intel (IA32), um ciclo que lhe permita copiar 1024 bytes de memória a partir do endereço dado pela etiqueta (*label*) A1 para a zona de memória com o início dado pela etiqueta A2.

5. Considere a seguinte função C, que devolve a soma de todos os inteiros entre 1 e n:

```
int somas( int n )
{
  int i, s = 0;
  for ( i=1; i<=n; i++ )
    s = s+i;
  return s;
}
```

- 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 = somas(4);
```

6. Na arquitectura Intel IA32, considere as instruções da forma ADD EAX, op, onde op pode ser um operando imediato, um registo do CPU, ou uma forma de endereçamento de memória. Indique, justificando, as diferenças entre estas instruções em termos de componentes da arquitectura envolvidos na sua execução e do tempo de execução.

7. Admita uma arquitectura do tipo Von Neumann com as seguintes características:

- registos PC (*program counter*) e SP (*stack pointer*) com 21 bits;
- registos de dados para operações da ALU de 16 bits.

- Qual deve ser a largura (número de linhas) dos *buses* de dados e de endereços? Justifique.
- Sabendo cada célula de memória contém 16 bits (2bytes) qual o limite máximo para a memória desta arquitectura (em KBytes ou em MBytes).
- Considerando que esta arquitectura suporta instruções com endereçamento directo da memória, discuta, justificando, o tamanho mínimo (em bits) da codificação de cada uma destas instruções em código máquina.