Universidade do Minho — 8 de Junho de 2001

Engª de Sistemas e Informática — 1ª Chamada

Arquitectura de Computadores II — Duração: 2 horas

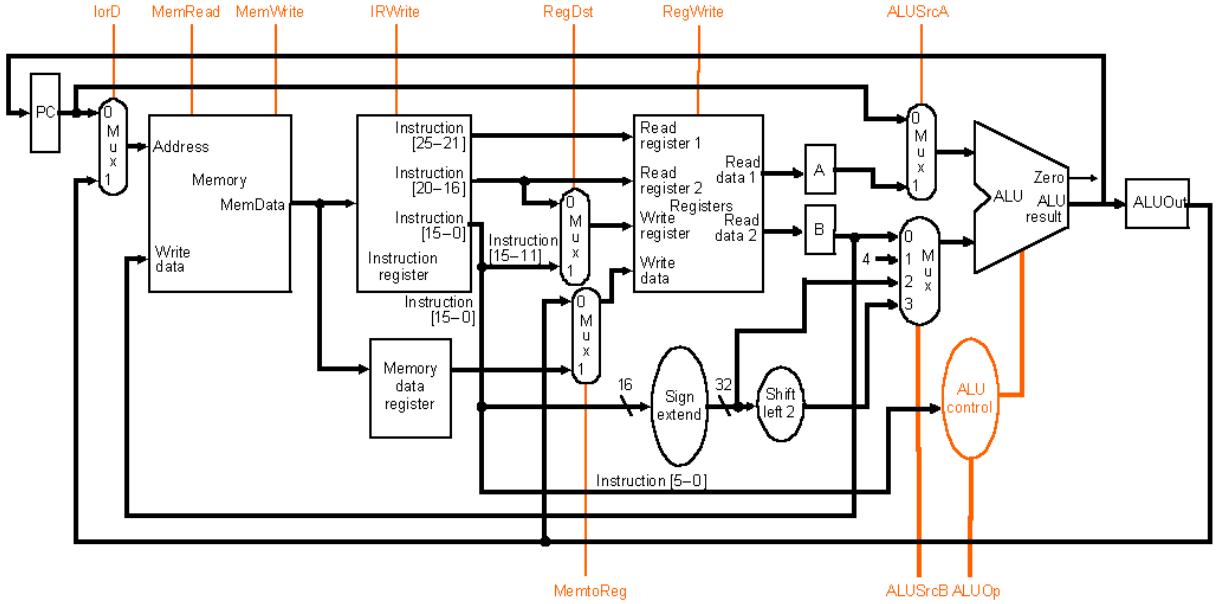
---

1. Discuta as vantagens e as desvantagens de uma arquitectura que executa as instruções num só ciclo máquina, relativamente a uma arquitectura que executa cada instrução em vários ciclos máquina.

2. Considere uma arquitectura do MIPS com suporte à execução de instruções em vários ciclos máquina, apresentada na figura.

2.1 Descreva as várias fases de execução da instrução *lw $t1, 0x200($t0)*. Indique, para cada fase, o valor dos sinais de controlo.

2.2 Indique as alterações que acha necessárias para que esta arquitectura possa executar uma instrução que efectue a adição de três registos (i.e., add $rd, $rs1, $rs2, $rs3).



3. Numa arquitectura de computadores actual, existe uma hierarquia de barramentos entre os vários componentes do sistema. Explique qual a finalidade de uma hierarquia de barramentos e discuta as suas principais vantagens e desvantagens.

4. O seguinte fragmento de código, desenvolvido para o PG/01, efectua uma digitalização de um som, sendo este armazenado no vector *vect*. Altere o programa por forma a que o som digitalizado seja reproduzido (através do conversor D/A), ao mesmo tempo que é digitalizado. Considere que a porta de dados do conversor D/A está no endereço 0x300 e que é necessário efectuar uma sondagem no bit 0 do registo 0x308 (deve estar a um), para assegurar que o conversor D/A se encontra disponível.

```
int vect[10000];
int i=0;

for(i=0; i<10000; i++) {
      outp(0x304,1);          // inicia a conversão A/D
      while ( inp(0x300)&1 ) ;
      vect[i] = inp(0x304);   // lê o valor
}
```

5. Em algumas arquitecturas existe uma instrução que efectua um *load* e automaticamente incrementa um registo. Por exemplo, *lw $t0, 0x100($t1)+*, carrega para o registo $t0 o valor de mem[$t1+0x100] e incrementa o registo $t1. Discuta o impacto que este tipo de instruções tem no mecanismo de execução em *pipeline*. Discuta também no caso da arquitectura ser super-escalar.

6. Considere o seguinte fragmento de programa em *assembly* do MIPS:

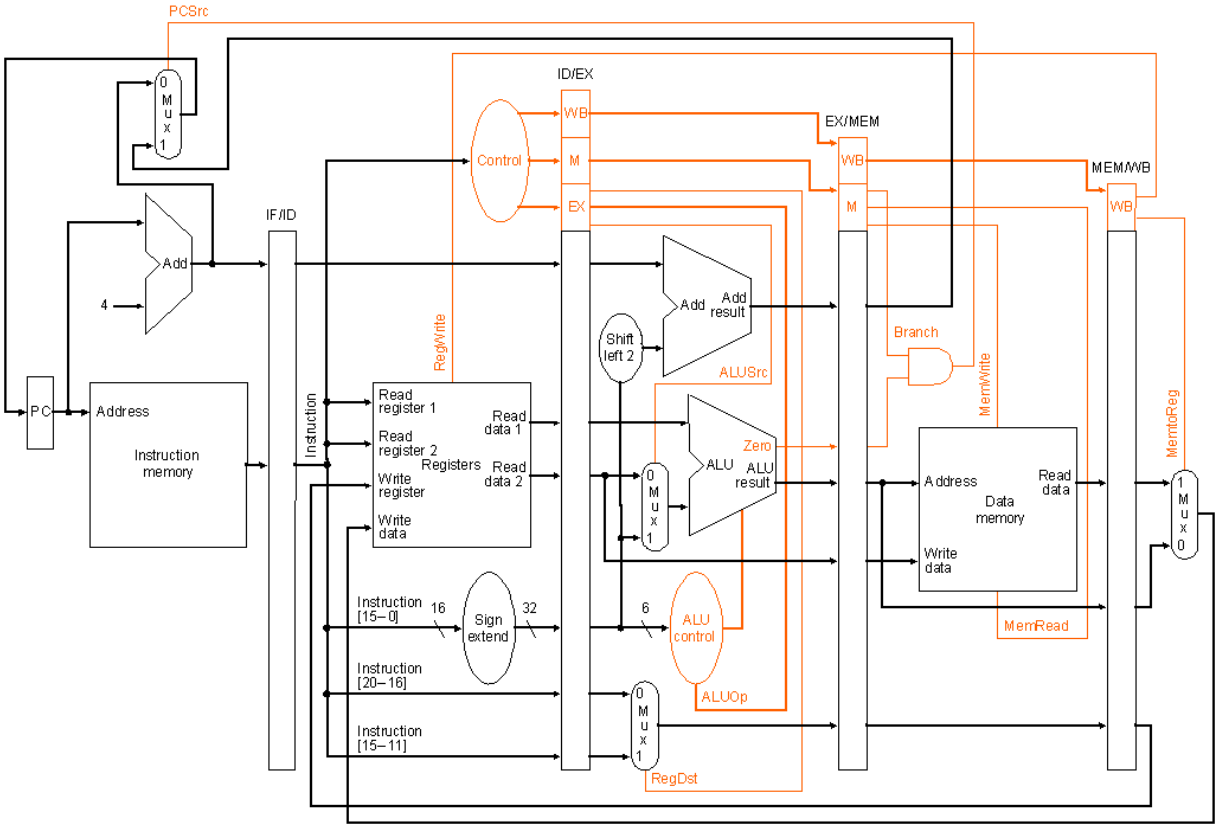
```
        mov $s0, 0x200
        mov $s1, 0
ld:     lw $t0, 0x300 ($s0)
        add $s1, $s1, $t0
        add $s0, $s0, -4
        bne $s0, $0, ld
```

Considere que toda a informação necessária para a execução se encontra em *cache*, que não existe encaminhamento de dados e que a fase de *write back* é realizada na primeira metade do ciclo, sendo a leitura dos registos efectuada na segunda metade.

6.1 Calcule o número de ciclos por instrução (CPI) obtido na execução deste programa. Apresente todos os cálculos que efectuar.

6.2 Calcule o CPI obtido quando é introduzida uma unidade de encaminhamento de dados.

6.3 Apresente o conteúdo do registo ID/Ex no início do 5º ciclo de execução, considerando que o programa é carregado para o endereço 0x60000 e que existe uma unidade de encaminhamento de dados.



7. Num sistema multiprocessador com memória partilhada (i.e., centralizada), indique porque é necessário um protocolo de coesão de *caches*, especificando qual princípio do seu funcionamento.