
 Universidade do Minho	Arquitectura de Computadores II L.E.S.I. – 3º ano	
	Módulo nº 7 Escalonamento de Instruções	

A bibliografia relevante para este módulo é a secção 5.7 do livro “Computer Systems: a Programmer’s Perspective”, de Randal E. Bryant e David O’Hallaron.

Exercícios

1 – Considere um processador baseado na arquitectura IA32 que inclui 4 unidades funcionais, com as latências e *issue time* apresentadas:

- **Integer/branch** – realiza todas as operações com inteiros e também processa os saltos condicionais; Latência 1 ciclo; *Issue time* 1 ciclo;
- **Integer** – realiza todas as operações com inteiros; Latência 1 ciclo; *Issue time* 1 ciclo;
- **Load** - realiza todas as leituras de memória; Latência 3 ciclos; *Issue time* 1 ciclo;
- **Store** – realiza todas as escritas em memória; Latência 3 ciclos; *Issue time* 1 ciclo.

O programa em C aqui apresentado:

```
int i, a[1000];
for (i=0 ; i< 1000 ; i++) a[i] += 20;
```

foi compilado tendo resultado no seguinte código *assembly* para o corpo do ciclo:

```
.L3: addl 20, (%ebx, %eax, 4) # a[i] += 20
     incl %eax             # i++
     cmpl 1000, %eax      # Códigos de condição <= i-1000
     jle .L3              # salta se cc S=1
```

Este código é convertido pela unidade de controlo de instruções nas seguintes operações elementares. (NOTA: *n* representa o número da iteração)

```
.L3: load (%ebx, %eax.n, 4)    -> t1.n
     addl 20, t1.n            -> t1.(n+1)
     store (%ebx, %eax.n, 4)
     incl %eax.n             -> %eax.(n+1)
     cmpl 1000, %eax.(n+1)  -> cc.n
     jle cc.n               .L3
```

1.1 – Apresente o diagrama de ocupação das tabelas funcionais.

1.2 – Suponha que o compilador realiza *loop unrolling*, resultando nas seguintes operações elementares:

```
.L3: load (%ebx, %eax.n, 4)    -> t1.n
      addl 20, t1.n           -> t1.(n+1)
      store (%ebx, %eax.n, 4)
      load 4 (%ebx, %eax.n, 4) -> t2.n
      addl 20, t2.n           -> t2.(n+1)
      store 4 (%ebx, %eax.n, 4)
      addl 2, %eax.n          -> %eax.(n+1)
      cmpl 1000, %eax.(n+1) -> cc.n
      jnl cc.n               .L3
```

Apresente o diagrama de ocupação das tabelas funcionais.

2 – Considere um processador baseado na arquitectura IA32 cujas unidades funcionais têm as seguintes características:

Operação	Duração em ciclos	Ciclos entre operações
Operações Inteiras Elementares e Salto	1	1
Operações Inteiras Elementares	1	1
Ler/Escrever na Memória (<i>cache</i>)	2	1
Multiplicação Inteira	2	1

Abaixo, em linguagem simbólica, um excerto do resultado da compilação de um programa em C:

```
.L3: imull (%eax, %ebx, 4), %ecx
      incl %ebx
      cmpl %esi, %ebx
      jnl .L3
```

2.1 – Apresente as operações elementares geradas por um hipotético decodificador de instruções para posterior execução pela unidade de controlo do processador.

2.2 – Apresente o diagrama de ocupação das tabelas funcionais.

2.3 – Para tirar partido das reais capacidades das unidades funcionais é possível combinar: **1)** o paralelismo de duas vias com **2)** o desenrolar de laço em duas vias. Isto é, fazer ao mesmo tempo, o processamento **paralelo** de **dois pares** de elementos de dados, por iteração.

2.3.1 - Apresente a sequência de operações elementares a realizar pelo processador.

2.3.2 - Apresente o diagrama de ocupação das tabelas funcionais.