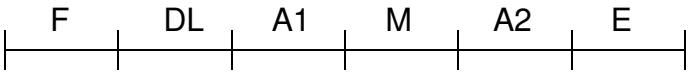
Examen Final Segmentació i Paral.lelisme

Departament d'Arquitectura de Computadors
25 de gener del 2000

Problema 1. (1er parcial)

Tenemos un procesador segmentado lineal con 6 etapas.



F:	fetch, actualiza PC
DL:	decodifica, analiza riesgos, lee registros fuente
A1:	calcula dirección de dato
M:	accede a memoria de datos
A2:	efectúa operación aritmético-lógica, pone (cond) si es OPRM
E:	escribe registro destino al final del ciclo

Las instrucciones a ejecutar son de los cuatro tipos siguientes:

```
OPRM   Ri ← Ri op Mem[Rj + Rk]; cond ← según resultado
LOAD   Ri ← Mem[Rj + Rk]
STORE  Mem[Rj + Rk] ← Ri
BRANCH if (cond) then PC ← PC + Mem[Rj + Rk]
```

Se pide:

a) Diseñar el camino de datos de este procesador para que sea capaz de ejecutar cualquiera de estas instrucciones sin ningún riesgo estructural, usando sucesivamente las 6 etapas.

b) Sabemos que este camino de datos tiene que incluir algunos cortocircuitos, porque no se produce ningún ciclo de bloqueo cuando ejecuta los 4 fragmentos de código siguientes:

```
LOAD   R3 ← Mem[R4 + R5]
LOAD   R4 ← Mem[R5 + R6]
STORE  Mem[R3 + R6] ← R5
```

CODIGO I

```
OPRM   R3 ← R3 op Mem[R4 + R5]
LOAD   R4 ← Mem[R5 + R6]
OPRM   R5 ← R5 op Mem[R6 + R7]
BRANCH if (cond) then PC ← PC + Mem[R3 + R7]
```

CODIGO II

```
LOAD   R3 ← Mem[R4 + R5]
STORE  Mem[R5 + R6] ← R3
OPRM   R6 ← R6 op Mem[R5 +
R7]
```

CODIGO III

```
LOAD   R3 ← Mem[R4 + R5]
LOAD   R4 ← Mem[R6 + R5]
OPRM   R6 ← R6 op Mem[R7 + R5]
STORE  Mem[R3 + R7] ← R5
```

CODIGO IV

Se pide mostrar en cuatro cronogramas distintos la ejecución de cada uno de estos fragmentos de código, para deducir e identificar los cortocircuitos necesarios.

Problema 2. (2on parcial)

Sigui el seguent programa:

```
for ( i = 1; i <= N ; i++) {
   if (A[i] ≠ 0) then X[j] = Y[i] / A[i];
}
```

i la seva traducció a llenguatge màquina per un processador escalar:

CODI ESCALAR

```
        mov $1, N
        mov $2, @A
        mov $3, @X
        mov $4, @Y
bucle:  ld F1, 0($2)
        beqf F1, fin
        ld F2, 0($3)
        divf F3, F2, F1
        st F3, 0($4)
fin:    add $2, $2, 8
        add $3, $3, 8
        add $4, $4, 8
        dec $1
        bne $1, bucle
```

Considereu que l'arquitectura superescalar inclou:

- Processador superescalar capaç d'iniciar l'execució de W instruccions per cicle.
- Unitats funcionals: una unitat funcional entera (latencia 1 cicle), una unitat funcional de punt flotant (latencia 10 cicles, segmentada) i dos unitats d'accés a memòria (latencia 1 cicle).
- El processador incorpora predicció de salts perfecte (sempre encerta el resultat del salt).
- La memoria cache es ideal i te una longitud de linia de 4 paraules (4 instruccions). Considereu que la primera instruccio del programa esta alineada al comencament d'una linia de cache.

Es demana: Calculeu el temps mitg d'execució del programa que executa N=64 iteracions en l'arquitectura superescalar per amplades d'execuci'o W=2 i W=4. Considereu que la probabilitat de saltar a "beqf F1, fin" es p.