

Probl2

# ILP

Arquitectura e Ingeniería de  
Computadores

Valentin Puente



10

## 1 SEGMENTACIÓN

Suponer que el porcentaje de saltos condicionales de un código es 15% y los incondicionales de 1%. El 60% de los condicionales son tomados. Estamos trabajando con un pipeline de 4 etapas (suponer el cálculo de la dirección y acceso a memoria en los load/stores se realiza en EX) en el que los saltos incondicionales son resueltos en el segundo ciclo y los condicionales al final de 3er ciclo. Suponiendo que la primera etapa del pipeline siempre puede ser hecha (independientemente de dónde va el salto condicional). ¿Cuánto más rápido puede ir el pipeline sin ningún riesgo de control?

## 2 RIESGOS DE DATOS Y PLANIFICACIÓN ESTÁTICA

Imagina que eres el encargado de diseñar una nueva arquitectura de procesador y que estas intentando descubrir cuál es el mejor modo de maximizar la tasa de utilización de las unidades funcionales. La latencia de cada una de las unidades que puedes emplear es:

Memoria LD	7
Memoria ST	3
Enteros ADD/SUB	1
Branches	3
ADDF	3
MULTF	6

```

loop: LD      F2, 0(R1)
      LD      F4, 0(R2)
      MULF    F2, F0, F6      # F6 is dest reg.
      ADDF    F6, F4, F6      # F6 is dest reg.
      LD      F4, 100(R3)
      ADDF    F6, F4, F2      # F2 is dest reg.
      SD      F2, 20000(R1)
      ADD     R1, #8, R1      # R1 is dest reg.
      ADD     R2, #8, R2      # R2 is dest reg.
      BLT    R1, R4, loop
  
```

- i. Suponiendo que no disponemos de *forwarding*, determinar cuál el número de ciclos requerido para su ejecución. Suponer que no se puede iniciar la ejecución de una nueva instrucción hasta que la anterior no ha finalizado (unidades no segmentadas). Ignorar los efectos en el Fetch y decode, asumiendo que en la entrada en ejecución siempre hay una instrucción disponible. Solo puede ser lanzada a ejecución una instrucción por ciclo. Emplear la representación usada en ejecución fuera de orden para hacer la solución: eje vertical del diagrama instrucciones, eje horizontal fase de la instrucción, dentro ciclo en que se inicia la fase.

- ii. Desarrollar una vez el lazo y planificar el código para optimizar su ejecución. ¿Cuál es el Speedup logrado si el lazo tiene 10000 iteraciones?

### 3 EJECUCIÓN FUERA DE ORDEN I

Construir y completar las siguientes tablas:

1. ROB
2. Estaciones de reserva
3. Map Table
4. Free List
5. CDB

Para un procesador R10K ejecutando el siguiente código al final de la segunda iteración:

```
loop:  ldf X(r1),f1
        ldf X(r2),f2
        mulf f0,f1,f2
        stf f0,X(r1)
        addi r1,4,r1
        addi r2,4,r2
        subi r3,1,r3
        BNQZ r3,loop
```

Suposiciones a tener en cuenta:

1. El procesador puede hacer *issue* de una sola instrucción por ciclo
2. El procesador puede hacer *retire* de una sola instrucción por ciclo
3. Las instrucciones de punto flotante y acceso a memoria requieren 2 ciclos para ejecutarse
4. Las operaciones en enteros requieren 1 ciclo
5. La predicción de saltos es perfecta y el ciclo anterior se repite dos veces
6. Solo hay disponibles 12 registros físicos y pueden ser usados para guardar valores en FP o INT
7. El ROB tiene 8 entradas y hay 5 RS

En el ciclo 3 las tablas tienen el siguiente contenido:

```

ROB
ht   #   Inst           T   Told S   X   C   R
h1   ldf X(r1),f1     PR#7 PR#2 c2   c3   c5
t2   ldf X(r2),f2     PR#8 PR#3 c3

```

```

Reservation Stations
#FU  Busy op   T   T1   T2
1ALU  N
2L/S1 Y   ldf  PR#7 -   PR#4+
3L/S2 N
4FP1  N
5FP2  N

```

```

Map Table
f0    PR#1+
f1    PR#7
f2    PR#8
r1    PR#4+
r2    PR#5+
r3    PR#6+

```

```

Free List
PR#9, PR#10, PR#11, PR#12

```

## 4 EJECUCIÓN FUERA DE ORDEN II

Repetir el problema anterior suponiendo que la unidad de punto flotante requiere 5 ciclos, el load requiere 3 ciclos y el store 2 y que el código a ejecutar es:

```

loop:  ldf X(r1),f1
        mulf f1,f3, f0
        ldf X(r2),f1
        addf f1, f0, f2
        stf f2,X(r1)
        addi r1,4,r1
        addi r2,4,r2
        slt r1,r3,r4
        BNQZ r4,loop

```

En el ciclo 2 las tablas tienen el siguiente contenido:

ROB								
ht	#	Inst	T	Told	S	X	C	R
h1		ldf X(r1),f1	PR#9	PR#2	c2			
t2		mulf f1,f3,f0	PR#10	PR#1				
3								

Reservation Stations					
#FU	Busy	op	T	T1	T2
1ALU	N				
2L/S1	Y	ldf	PR#9	-	PR#5+
3L/S2	N				
4FP1	Y	mulf	PR#10	PR#9	PR#4+
5FP2	N				

Map Table	
f0	PR#10
f1	PR#9
f2	PR#3+
f3	PR#4+
r1	PR#5+
r2	PR#6+
r3	PR#7+
r4	PR#8+

Free List
PR#11, PR#12, PR#13, PR#14, PR#15, PR#16