



ARQUITECTURA DE COMPUTADORAS

Trabajo Práctico N° 8

Explotando el Paralelismo a Nivel de Instrucciones y Multiple Issue Processors ¹

Primer Cuatrimestre de 2018

Ejercicios

1. Explique las diferencias entre una **dependencia verdadera** y una **dependencia de nombre**. ¿Cuál es su relación con los denominados *hazards*?
2. Discuta la veracidad de la siguiente afirmación: *"Mientras más unidades de ejecución se agreguen al procesador mejor, la ganancia en performance es lineal."*
3. ¿Cuál es el objetivo de la técnica conocida como *register renaming*? Analice las ventajas y desventajas de contar con un esquema de renaming dinámico por hardware o contar con un número mayor de registros a nivel de ISA.
4. Considerando el siguiente fragmento de código:

I_1	LD	R7, (R8)
I_2	MUL	R1, R7, R2
I_3	SUB	R7, R4, R5
I_4	ADD	R9, R7, R8
I_5	LD	R8, (R12)
I_6	DIV	R10, R8, R10

- a) Identifique todas las dependencias.
 - b) Muestre el renombre dinámico realizado por el hardware asumiendo que posee los suficientes registros de arquitectura.
5. Dado el siguiente fragmento de código:

I_1	LD	R1, (R2)
I_2	ADD	R3, R4, R1
I_3	SUB	R4, R5, R6
I_4	MUL	R7, R4, R8
I_5	ADD	R8, R9, R10
I_6	SUB	R11, R8, R12
I_7	DIV	R12, R13, R14
I_8	ST	(R15), R12

¹Fecha sugerida de finalización de este trabajo práctico: 5 de junio de 2018.

Determinar cuántos registros en hardware son necesarios para eliminar todas las dependencias de nombre y mostrar el mapeo dinámico realizado por el hardware.

6. Realice una comparación entre un esquema de planificación estático (*static scheduling*) y un esquema de planificación dinámico (*dynamic scheduling*). Analice las ventajas y desventajas de cada enfoque.

Inst. produce resultado	Inst. usa resultado	Latencia (ciclos)
FP ALU OP	FP ALU OP	3
FP ALU OP	ST DOUBLE	2
LD DOUBLE	FP ALU OP	1
LD DOUBLE	ST DOUBLE	0
INT ALU OP	*	1

Cuadro 1: Latencias entre operaciones

7. Dado el siguiente fragmento de código: Desarrolle los siguientes incisos teniendo en cuenta

```

I1 : Loop: LD      F0, 0 (R1)
I2 :      ADDD    F4, F0, F2
I3 :      SD      F4, 0 (R1)
I4 :      DADDUI  R1, R1, #-8
I5 :      BNE     R1, R2, Loop

```

las latencias mostradas en el Cuadro 1.

- Explique que tarea realiza. ¿Podría asociarlo a alguna estructura de control de un lenguaje de alto nivel?
 - Identifique las dependencias involucradas y muestre en que ciclo se realiza el **issue** de cada instrucción.
 - Utilice la técnica **pipeline scheduling** para planificar las instrucciones. ¿El código resultante presenta alguna mejora en cuanto a tiempo de ejecución?
8. Considerando el código del ejercicio 7 y las latencias reflejadas en el cuadro 1, muestre cómo quedaría el código inicial aplicando la técnica de loop unrolling para obtener cuatro copias del cuerpo del ciclo. Elimine cálculos redundantes y no reutilice registros. Asuma que el número de operaciones es múltiplo de 4 y que la arquitectura posee la su ciente cantidad de registros.

9. Dada la siguiente secuencia de instrucciones:

Resolver los siguientes incisos asumiendo que la arquitectura cuenta con un procesador superescalar con **ejecución en orden**, el cual cuenta con dos unidades de ejecución, es decir que se pueden despachar dos instrucciones (de cualquier tipo) por ciclo ($w=2$). Asumir que las instrucciones de acceso a memoria poseen una latencia de dos ciclos, mientras que las restantes poseen una latencia de un ciclo.

I_1 :	LD	R1, (R2)
I_2 :	SUB	R4, R5, R6
I_3 :	ADD	R3, R1, R7
I_4 :	MUL	R8, R3, R3
I_5 :	ST	(R11), R4
I_6 :	ST	(R12), R8
I_7 :	ADD	R15, R14, R13
I_8 :	SUB	R10, R15, R10
I_9 :	ST	(R9), R10

- a) Esquematizar en un *diagrama de Gantt* el solapamiento entre instrucciones, indicando claramente en qué ciclo de reloj se despacha cada instrucción y por qué.
 - b) Indique el tiempo de despacho y el tiempo de ejecución de la secuencia de instrucciones.
10. Repetir el Ejercicio 9, esta vez considerando que la arquitectura posee dos unidades funcionales para el cálculo de direcciones efectivas de 1 ciclo, una unidad funcional de suma de 1 ciclo y una unidad multiplicadora de 3 ciclos. A su vez, suponer que la arquitectura posee *forwarding* y hacer explícitos los *bypasses* necesarios.
 11. En un procesador Superescalar (Dinamic Issue) con ejecución fuera de orden, se realiza el fetch de un conjunto w de instrucciones y se las despacha (issue) en un determinado ciclo. Los conflictos a nivel de recursos de datos no frenarán el pipeline, solo a las instrucciones con dependencias.
Rehacer el ejercicio 10 para las mismas condiciones pero con ejecución fuera de orden.
 12. Un procesador VLIW define cinco slots por instrucción y está conformado por dos unidades funcionales para el cálculo de operaciones enteras de dos ciclos, una unidad funcional para el cálculo de operaciones de punto flotante de dos ciclos y dos unidades para acceso a memoria de tres ciclos. En este contexto, mostrar el formato de la instrucción y determinar cómo el compilador planificaría el siguiente código:

I_1 :	ADDI	R1, R2, R3	I_{10} :	LD	F21, (R23)
I_2 :	MULI	R4, R5, R1	I_{11} :	LD	R24, (R25)
I_3 :	SUBD	F1, F3, F4	I_{12} :	ADDD	F17, F18, F19
I_4 :	LD	R2, (R21)	I_{13} :	SUBD	F15, F21, F23
I_5 :	SUBI	R8, R4, #8	I_{14} :	ADDI	R12, R21, R3
I_6 :	LD	F11, (R4)	I_{15} :	MULI	R21, R24, R28
I_7 :	ADDI	R9, R15, R16	I_{16} :	MULD	F23, F24, F28
I_8 :	SUBI	R15, R16, R17	I_{17} :	SUBD	F18, F4, F21
I_9 :	LD	R21, (R22)			

Referencias

- [Bae10] BAER, J. L. *Microprocessor Architecture*. Cambridge University Press, 2010.
- [HP12] HENNESSY, J., AND PATTERSON, D. *Computer Architecture*, fifth ed. Morgan Kaufmann, 2012.