



Profesores: Néstor González – Leo Medina
 Fecha: 11 de septiembre de 2018
 Duración: 90 minutos

Pregunta 1. Procesador Monociclo (2 pts.)

En el diseño de procesadores, para considerar una posible mejora en el camino de datos del procesador, la decisión muchas veces depende del compromiso entre costo y desempeño. Considere el procesador monociclo de la Figura 1 y las siguientes latencias, en ps, y costos, en \$, de los bloques que se indican:

	I-Mem	Add	Mux	ALU	Regs	D-Mem	Control
Latencia (ps)	400	100	30	120	200	350	100
Costo (\$)	1000	30	10	100	200	2000	500

Considere la adición de un multiplicador a la ALU. Esta potencial mejora agregará 300 ps a la latencia de la ALU, e incrementará su costo en \$600. Agregar el multiplicador significará una reducción del 25% del número total de instrucciones, pues ya no será necesario emular la instrucción MUL.

- A) (0,5 pts.) Indique qué función cumple cada uno de los dos bloques sumadores etiquetados como “Add” en el diagrama de la Figura 1. ¿Participan en el camino de datos crítico que determina la tasa de reloj de este procesador?

El sumador de la izquierda incrementa el contador de programa, PC, en 4 para obtener la instrucción siguiente, y el de la derecha, le suma al PC el offset de un salto. Estos sumadores no contribuyen al camino de datos crítico, pues sus latencias son menores que aquellas de Regs y ALU, y no interfieren en los cálculos aritméticos o de direcciones de las instrucciones que toman mas tiempo.

- B) (0,5 pts.) ¿Cuál es la tasa de reloj de este procesador monociclo con y sin esta mejora?

El camino crítico está dado por una instrucción de acceso a memoria: I-Mem + Regs (toma mas tiempo que Control) + Mux (seleccionar entrada a ALU) + ALU + D-Mem + Mux (seleccionar valor de memoria para ser escrito en registros). Entonces, sin mejora se tiene una latencia total de 1130 ps, y con la mejora, 1430 ps (como la ALU está en el camino crítico, simplemente se suman los 300 ps adicionales de latencia de la ALU en esta condición).

- C) (0,5 pts.) ¿Cuál es la aceleración lograda al incorporar esta mejora?

Con la mejora, el reloj es mas lento, pero se requieren 25% menos ciclos en total para el programa. La aceleración es $(1/0.75) * (1130/1430) = 1,054$. Es decir, el procesador con la mejora sería 5,4% mas rápido.

- D) (0,5 pts.) Compare la razón costo/desempeño con y sin esta mejora, y concluya sobre la conveniencia de introducir esta mejora.

Del diagrama del procesador monociclo, y considerando todos los elementos (no solo aquellos del camino crítico), tenemos un costo total de \$3890 sin la mejora, y de \$4490 con la mejora. El costo relativo es entonces 1,15, y la razón costo/desempeño, $1,15/1,054 = 1,09$. Es decir, conviene introducir la mejora pues pagamos mas, pero obtenemos un desempeño comparativamente mejor por el costo adicional.

**Pregunta 2. Pipeline (2 pts.)**

Considere el siguiente bucle en MIPS para un procesador con un *pipeline* de 5 etapas:

```

loop:    lw     r1, 0(r1)
         and    r3, r1, r2
         lw     r1, 0(r1)
         lw     r1, 0(r1)
         beq    r1, r0, loop
  
```

Asuma que se tiene predicción de saltos perfecta, es decir, no hay esperas por *hazards* de control, y que el *pipeline* dispone de *forwarding* completo. Además, suponga que se ejecutan muchas iteraciones de este bucle antes de que el bucle termine.

- A) (0,5 pts.) Indique qué tipo de instrucción (R, I o J) es cada una de las instrucciones del bucle.
lw→I, *and*→R, *beq*→I
- B) (0,5 pts.) Muestre un diagrama de ejecución del *pipeline* para la tercera iteración de este bucle, desde el ciclo en el cual se puede hacer *fetch* de la primera instrucción de esa iteración, hasta (pero no incluyendo) el ciclo en el cual se puede hacer *fetch* de la primera instrucción de la iteración siguiente. Muestre todas las instrucciones que están en el *pipeline* durante estos ciclos (no solo aquellos desde la tercera iteración).

LW R1,0(R1)	WB
LW R1,0(R1)	EX MEM WB
BEQ R1,R0,Loop	ID *** EX MEM WB
LW R1,0(R1)	IF *** ID EX MEM WB
AND R1,R1,R2	IF ID *** EX MEM WB
LW R1,0(R1)	IF *** ID EX MEM
LW R1,0(R1)	IF ID ***
BEQ R1,R0,Loop	IF ***

- C) (0,5 pts.) ¿En qué porcentaje de los ciclos se tiene que las cinco etapas del *pipeline* están haciendo trabajo útil? (por ejemplo, la instrucción “*add r3, r1, r2*” no realiza trabajo útil durante la etapa MEM).
 En el diagrama anterior, las etapas marcadas en celeste no realizan trabajo útil (notar que *beq* está haciendo trabajo útil en MEM pues está determinando el valor correcto de la siguiente instrucción en esa etapa). En todos los ciclos hay o bien etapas en espera o sin realizar trabajo útil y, por lo tanto, en el 0% de los ciclos se tienen todas las etapas del *pipeline* haciendo trabajo útil.
- D) (0,5 pts.) Determine el rendimiento medido en CPI para estas condiciones.
 Las 5 instrucciones del bucle son ejecutadas en 8 ciclos de reloj. Entonces, $CPI = 8/5$.

**Pregunta 3. Caché (2 pts.)**

Considere direcciones de memoria de 32 bits, y asuma un largo de palabra de 4 bytes en todos los casos. Se tienen tres configuraciones de caché distintas, con sus respectivos tamaños de datos indicados entre paréntesis, y para las cuales los bits de la dirección de memoria son usados para “tag”, “index” y “offset” como se indica.

Caché X (4 MiB):

Tag	Offset
31-10	9-0

Caché Y (2 MiB):

Tag	Index	Offset
31-21	20-7	6-0

Caché Z (1 MiB):

Tag	Index	Offset
31-18	17-2	1-0

- A) (0,5 pts.) Para los datos entregados, ¿cuál es el tamaño máximo de memoria principal?
Dirección de memoria de 32 bits \rightarrow tamaño máximo $= 4 \cdot 2^{30}$ bytes, o 4 GiB.
- B) (0,5 pts.) Para cada caché, indique el tamaño del bloque en palabras y el número total de bloques.
Caché X: tamaño bloque $= 2^{10}$ bytes $= 2^8$ palabras. Nro bloques $= 4 \cdot 2^{20}$ bytes $/ 2^{10} = 2^{12}$ bloques.
Caché Y: tamaño bloque $= 2^7$ bytes $= 2^5$ palabras. Nro bloques $= 2 \cdot 2^{20}$ bytes $/ 2^7 = 2^{14}$ bloques.
Caché Z: tamaño bloque $= 2^2$ bytes $= 1$ palabra. Nro bloques $= 2^{20}$ bytes $/ 2^2 = 2^{18}$ bloques.
- C) (0,5 pts.) Indique el tipo de mapeo de caché en cada caso (si hay alguna caché de conjunto asociativo, especifique el número de vías).
Caché X no tiene index \rightarrow full asociativa.
Caché Y tiene 2^{14} líneas (bits index) e igual número de bloques \rightarrow mapeo directo.
Caché Z tiene 2^{16} líneas (bits index) y 2^{18} bloques \rightarrow mapeo conjunto asociativo de 4 vías.
- D) (0,5 pts.) Suponga que inicialmente cada caché está vacía. Para la siguiente secuencia de accesos a memoria principal expresadas en *byte*, calcule la tasa de desaciertos (“miss rate”) de cada caché, donde N es el número de bloques de esa caché y m es el número de bits de offset usados para esa caché.

$0 \cdot 2^m$	$1 \cdot 2^m$	$2 \cdot 2^m$...	$(N-1) \cdot 2^m$	$0 \cdot 2^{m+1}$	$1 \cdot 2^{m+1}$	$2 \cdot 2^{m+1}$...	$(N-1) \cdot 2^{m+1}$
---------------	---------------	---------------	-----	-------------------	-------------------	-------------------	-------------------	-----	-----------------------

NOTA: “tag” = etiqueta; “offset” se refiere al *offset* de palabra y byte dentro del bloque.

La primera mitad de esta secuencia de accesos toma el primer byte de cada uno de los N bloques, y la segunda mitad, el segundo byte de cada uno de los N bloques. Entonces, independientemente del tipo de mapeo, la primera mitad de la secuencia resulta sólo en *misses* y no se producen reemplazos, y la segunda mitad sólo en *hits*. En consecuencia, en todos los casos la tasa de desaciertos es 0,5.