

APELLIDOS Y NOMBRE:

Solució

GRUPO:

PROBLEMA 1 (2 puntos) La ruta de datos R1 no segmentada monociclo tiene una frecuencia de funcionamiento de 100 MHz. La ruta R2 se obtiene segmentando la ruta R1 en cinco etapas (LI, DI, EX, M, ER), obteniendo los retardos $\tau_{LI} = \tau_{EX} = \tau_M = 5$ ns y $\tau_{DI} = \tau_{ER} = 3$ ns y con registros de segmentación de retardo $T_R = 1$ ns.

1. Calcule las productividades de las rutas R1 y R2 al ejecutar 1000 instrucciones, $X_{R1}(1000)$ y $X_{R2}(1000)$ (expreselas en millones de instrucciones por segundo). Suponga que la unidad de control de la ruta segmentada introduce un total de 200 ciclos de parada para resolver los posibles riesgos.

$$\text{Cicle de R1} = \frac{1}{100 \text{ MHz}} = \frac{1}{100 \times 10^6 \text{ s}^{-1}} = 10 \text{ ns (MONOCICLE)}$$

$$\text{cicle de R2} = \max\{\tau_i\} + t_R = 5 + 1 = 6 \text{ ns (SEGMENTADA)}$$

$$\text{Temps exec. R1} = 1000 \text{ inst} \times 10 \text{ ns} = 10 \mu\text{s}$$

$$\text{Temps exec. R2} = (1000 + 200 + 4) \cdot 6 \text{ ns} = 7,224 \mu\text{s}$$

Alternativa: calcular
 $t_{\text{exec}} = \text{CPI} \times I \times \text{cicle}$

$$X_{R1}(1000) = \frac{1000 \text{ inst}}{10 \mu\text{s}} = 100 \text{ MIPS}; \quad X_{R2}(1000) = \frac{1000 \text{ inst}}{7,224 \mu\text{s}} = 138 \text{ MIPS}$$

2. Determine la aceleración $S(1000)$ conseguida mediante la segmentación de la ruta monociclo en la ejecución de las 1000 instrucciones.

Dividim el temps d'execució de les dues rutes:

$$S(1000) = \frac{10 \mu\text{s}}{7,224 \mu\text{s}} = 1,38.$$

La ruta R2 és 1,38 vegades més ràpida que R1. El temps d'execució de R2 inclou els 200 cicles de parada i els 4 cicles inicials per a omplir la ruta segmentada.

3. En una situació ideal con infinitas instrucciones y sin ciclos de parada, calcule el valor máximo de la aceleración, $S(\infty)$, que obtendría la ruta segmentada respecto de la no segmentada.

En condicions ideals, R2 acaba d'executar una instrucció cada cicle de rellotge (no hi ha cap cicle de parada):

$$S(\infty) = \frac{\text{cicle de R1}}{\text{cicle de R2}} = \frac{10 \text{ ns}}{6 \text{ ns}} = 1,67.$$

Evidentment, tenim que $S(\infty) \geq S(1000)$

PROBLEMA 2 (3 puntos) Considere un procesador MIPS R2000 implementado con una ruta de datos MONOCICLO y un reloj con una frecuencia de 2 MHz que ejecuta el siguiente código. El programa hace la operación $v[i] = -2 * v[i]$ sobre un vector de enteros con signo. Nótese que el programa solamente contiene instrucciones máquina y que la instrucción lui es una instrucción aritmética.

```

                .data 0x10003000
vector:        .word -1, 3, 7, -5, 9, 2, -4, 8
                .globl __start
                .text 0x00400000
1)  __start:    lui  $t0, 0x1000
2)              ori  $t0, $t0, 0x3000
3)              addi $t1, $zero, 8
4)  bucle:      lw   $s0, 0($t0)
5)              sub  $s0, $zero, $s0
6)              add  $s0, $s0, $s0
7)              sw   $s0, 0($t0)
8)              addi $t0, $t0, 4
9)              addi $t1, $t1, -1
10)             bgtz $t1, bucle

```

1. ¿Cuál es el periodo de reloj?

$$\text{cicle} = \frac{1}{2\text{MHz}} = \frac{1}{2 \cdot 10^6 \text{s}^{-1}} = 500 \text{ ns}$$

2. ¿Cuántas instrucciones tiene el programa?

10 instrucciones máquina (de la 1 a la 10)

3. ¿Cuántos bytes ocupa el segmento de código del programa?

$10 \times 4 \text{ bytes} = 40 \text{ bytes}$ (cada instrucció ocupa 32 bits = 4 bytes)

4. Si el PC (contador de programa) contiene el valor 0x00400014, ¿a qué instrucción apunta?

add \$s0, \$s0, \$s0. $0x14 = 20_{10}$ (desplazament de la 5a instrucció en memòria)

5. ¿Cuánto tiempo tarda en ejecutarse una instrucción?

un cicle, això és, 500 ns (ruta monocicle!)

6. ¿Cuántas instrucciones se ejecutan?

$3 \text{ (fora del bucle)} + 8 \times 7 \text{ (dins del bucle)} = 59 \text{ instruccions}$

7. ¿Cuánto tiempo tarda el programa en ejecutarse?

$59 \times 500 \text{ ns} = 59 \text{ cicles} = 29,5 \mu\text{s}$

8. ¿Cuál es el valor del CPI obtenido por el programa?

$\text{CPI} = 1$ En una ruta monocicle cada instrucció s'executa en un cicle de rellotge.

PROBLEMA 3 (5 puntos) Considere un procesador MIPS R2000 implementado con una ruta de dades SEGMENTADA como la estudiada en clase y un reloj con una frecuencia de 10 MHz que ejecuta el mismo código que el del problema 2. La latencia de salto es de tres ciclos de reloj.

1. ¿Cuál es el periodo de reloj?

$$cicle = \frac{1}{10 \text{ MHz}} = \frac{1}{10 \cdot 10^6 \text{ s}^{-1}} = 100 \text{ ns}$$

2. ¿Cuánto tiempo tarda en ejecutarse una instrucción?

Una instrucció s'executa en $K \cdot C = 5 \cdot 100 \text{ ns} = 500 \text{ ns}$, això es, calen 5 cicles per a executar cada instrucció.

3. Indique en la siguiente tabla los conflictos de datos del programa. Use los números de línea para identificar las instrucciones que presentan conflicto.

Instrucciones	Registro
Entre 1 i 2 (lui i ori)	\$t0
Entre 2 i 4 (ori i lw)	\$t0
Entre 4 i 5 (lw i sub)	\$s0
Entre 5 i 6 (sub i add)	\$s0
Entre 6 i 7 (add i sw)	\$s0
Entre 9 i 10 (addi i bgtz)	\$t1

4. Indique dónde se producen los conflictos de control del programa.

S'originen per la instrucció bgtz. Aquest conflicte es resol amb una latència de 3 cicles.

5. Sin reordenar el código, inserte las instrucciones nop necesarias para resolver los conflictos de datos y de control del programa. ¿Cuánto tiempo tarda el programa en ejecutarse? ¿Cuál es el valor del CPI obtenido? Indique los detalles del cálculo.

1) lui
nop
nop
2) ori
3) addi
nop(*)
bucle: 4) lw
nop
nop
5) sub
nop
nop
6) add
nop
nop
7) sw
8) addi
9) addi
nop
nop
10) bgtz
nop
nop
nop

S'hi executen $6 + 8 \times 18 = 150$ instruccions
Tems d'execució = $150 + 4 = 154$ cicles
 $= 154 \times 100 \text{ ns} = 15,4 \mu\text{s}$

El CPI val 1 perquè tots els conflictes es resolen pel compilador amb instruccions nop i, per tant, no hi ha cap cicle de parada.

La instrucció nop(*) també es pot situar entre les instruccions 2) i 3).

Resolen el conflicte de control; la resta els conflictes de dades.

6. Sin reordenar el código, suponga ahora que se incorporan a la ruta de datos todos los cortocircuitos, tanto de la ALU como de memoria, necesarios para la anticipación de datos (MaEX, ERaEX y ERaM). Los conflictos de datos y de control no resueltos con cortocircuitos se resuelven mediante la inserción de instrucciones nop. Complete las tablas siguientes con la ejecución de la primera iteración del programa e indique, para cada conflicto, el cortocircuito que se aplica para resolverlo. ¿Cuál es ahora el valor del CPI?

Instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 lui	LI	DI	EX	M	ER										
2 ori		LI	DI	EX	M	ER									
3 addi			LI	DI	EX	M	ER								
4 lw				LI	DI	EX	M	ER							
nop					LI	DI	EX	M	ER						
5 sub						LI	DI	EX	M	ER					
6 add							LI	DI	EX	M	ER				
7 sw								LI	DI	EX	M	ER			
8 addi									LI	DI	EX	M	ER		
9 addi										LI	DI	EX	M	ER	
10 bgtz											LI	DI	EX	M	ER
nop												LI	DI	EX	M
nop													LI	DI	EX
nop														LI	DI
4 lw															LI
:															

Instrucciones	Cortocircuito aplicado	Ciclo
Entre 1 i 2 (lui i ori)	MaEX (S)	4
Entre 2 i 4 (ori i lw)	ERaEX (S)	6
Entre 4 i 5 (lw i sub)	ERaEX (I)	8
Entre 5 i 6 (sub i add)	MaEX (S i I)	9
Entre 6 i 7 (add i sw)	ERaM	11
Entre 9 i 10 (addi i bgtz)	MaEX (S)	13

← s'hi apliquen 2 cortocircuits alhora

El CPI val 1 perquè no hi ha cicles de parada. L'únic conflicte de dades no resol pels cortocircuits es resol amb una nop (càrrega retardada). El conflicte de control (latència de 3 cicles) també es resol amb tres nop.

7. Si en el caso anterior se hubiera podido ordenar el código, ¿qué instrucción se podría utilizar para llenar el hueco generado por la instrucción lw?

Segut a les dependències de dades entre les instruccions del bucle, el buit que genera lw només es pot omplir amb la instrucció addi \$t1, \$t1, -1