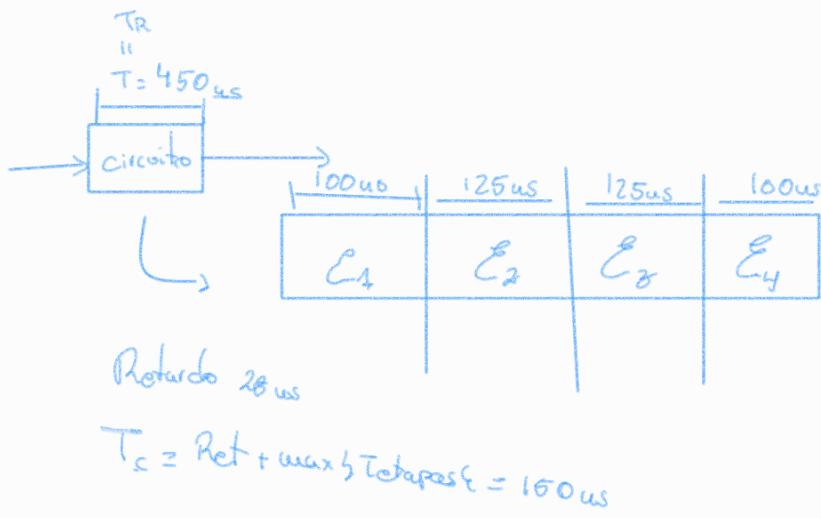


**Ejercicio 2.** Un circuito que implementaba una operación en un tiempo de  $T_{op} = 450$  ns se ha segmentado mediante un cauce lineal con cuatro etapas de duración  $T_1 = 100$  ns,  $T_2 = 125$  ns,  $T_3 = 125$  ns y  $T_4 = 100$  ns respectivamente, separadas por un registro de acople que introduce un retardo de 25 ns. **(a)** ¿Cuál es la máxima ganancia de velocidad posible? ¿Cuál es la productividad máxima del cauce? **(b)** ¿A partir de qué número de operaciones ejecutadas se consigue una productividad igual al 90% de la productividad máxima?



$$a) S(N) = \frac{T^b(N)}{T^s(N)} =$$

$$= \frac{N \cdot T_R}{T_{LI} + (N-1) \cdot T_c}$$

↓

$N^{\text{etapas}} \cdot T_c$

$$T_c = \text{Ret} + \max \{ T_{\text{etapas}} \} = 150 \text{ ns}$$

$$S(N \gg) = \frac{T_R}{T_c} =$$

$$= \lim_{N \rightarrow \infty} S(N)$$

**Ejercicio 1.** En el código de prueba (*benchmark*) que ejecuta un procesador no segmentado que funciona a 300 MHz, hay un 20% de instrucciones LOAD que necesitan 4 ciclos, un 10% de instrucciones STORE que necesitan 3 ciclos, un 25% de instrucciones con enteros que necesitan 6 ciclos, un 15% de instrucciones con operandos en coma flotante que necesitan 8 ciclos por instrucción, y un 30% de instrucciones de salto que necesitan 3 ciclos. **(a)** ¿Cuál es la ganancia que se puede obtener por reducción a 3 ciclos de las instrucciones con enteros? y **(b)** ¿cuál es la ganancia que se puede obtener por reducción a 3 ciclos de las instrucciones en coma flotante?

### Datos

•) No segmentado

•)  $f_p = 300 \text{ MHz}$

•)	0'20I	LOAD	4 ciclos (C0)
	0'10I	STORE	3 ciclos (ST)
	0'25I	ints	6 ciclos (FX)
	0'15I	float	8 ciclos (FP)
	0'30I	salto	3 ciclos (BR)

b) FP → 3 ciclos. Realmente

•)	0'2	LOAD	4 ciclos (C0)
	0'1	STORE	3 ciclos (ST)
	0'25	ints	6 ciclos (FX)
	0'15	float	3 ciclos (FP)
	0'3	salto	3 ciclos (BR)

a)  $f_x \rightarrow 3$  ciclos. Realmente

•)	0'2	LOAD	4 ciclos (C0)	$S(N) = \frac{T^b}{T_p} ;$
	0'1	STORE	3 ciclos (ST)	
	0'25	ints	6 ciclos (FX)	
	0'15	float	8 ciclos (FP)	
	0'3	salto	3 ciclos (BR)	

$$T_b = N_I \left[ \underbrace{0'2 \cdot 4c + 0'1c + 0'25 \cdot 6c}_{\text{No. instr}} + 0'15 \cdot 8c + 0'3 \cdot 3c \right] \cdot T_c$$

$$T_b = N_I \cdot (0'2 + 0'15) \cdot T_c =$$

$$= N_I \cdot 4/7 T_c$$

Hacerlo de forma análoga.

$$T_c = \frac{1}{f}$$

$$\begin{aligned} T_p &= N_I \left[ 0'2 + 0'15 + 0'3 \right] \cdot T_c = \\ &= N_I \cdot 3'95 T_c \\ \frac{T_p}{T_b} &= \frac{N_I \cdot 3'95 \cdot T_c}{N_I \cdot 4/7 T_c} = 1'18997 \end{aligned}$$

En conclusión, la ganancia depende del nº de veces que se usa la mejoría

**Ejercicio 3.** En un procesador sin segmentación de cauce, determine cuál de estas dos alternativas para realizar un salto condicional es mejor:

- ALT1: Una instrucción COMPARE actualiza un código de condición y es seguida por una instrucción BRANCH que comprueba esa condición. Se usan dos instrucciones.
- ALT2: Una sola instrucción incluye la funcionalidad de las instrucciones COMPARE y BRANCH. Se usa una única instrucción.

Víe su código, no sé cual, es el de  
un caso particular.

Hay que tener en cuenta que hay un 20% de instrucciones BRANCH para ALT1 en el conjunto de programas de prueba; que las instrucciones BRANCH en ALT1 y COMPARE+BRANCH en ALT2 necesitan 4 ciclos mientras que todas las demás necesitan sólo 3; y que el ciclo de reloj de la ALT1 es un 25% menor que el de la ALT2, dado que en éste caso la mayor funcionalidad de la instrucción COMPARE+BRANCH ocasiona una mayor complejidad en el procesador.

Datos

$I_i^1$	$T_{ops,1}$	$NI_i^1$	$I_i^2$	$T_{ops,2}$	$NI_i^2$
BR	4c	0'2 NI <sup>1</sup>	CMP BR	4c	0'2 NI <sup>2</sup>
CMP	3c	0'2 NI <sup>1</sup>	Resto	3c	0'6 NI <sup>2</sup>
Resto	3c	0'6 NI <sup>1</sup>			0'8 NI <sup>2</sup>

$$T_c = 0'75 T_0$$

$$T_{cpu}^1 = 0'2 \cdot [0'2 \cdot 4c + 0'2 \cdot 3c + 0'6 \cdot 3c] \cdot T_0$$

$$T_{cpu}^2 = 0'8 NI^2 \cdot [0'2 \cdot 4c + 0'6 \cdot 3c] \cdot 0'75 T_0$$

$$\Delta = \frac{T_{cpu}^2}{T_{cpu}^1} = \frac{0'2 \cdot 4 + 0'8 \cdot 3}{0'8 \cdot 0'75 + 0'2 \cdot 1 + 0'6 \cdot 0'3} = 2'045$$

Por tanto  $T_{cpu}^2$  es mejor que  $T_{cpu}^1$

**Ejercicio 5.** Considere un procesador no segmentado con una arquitectura de tipo LOAD/STORE en la que las operaciones sólo utilizan como operandos registros de la CPU. Para un conjunto de programas representativos de su actividad se tiene que el 43% de las instrucciones son operaciones con la ALU (3 CPI), el 21% LOADs (4 CPI), el 12% STOREs (4 CPI) y el 24% BRANCHs (4 CPI).

Se ha podido comprobar que un 25% de las operaciones con la ALU utilizan operandos en registros que no se vuelven a utilizar. Compruebe si mejorarían las prestaciones si, para sustituir ese 25% de operaciones, se añaden instrucciones con un dato en un registro y otro en memoria. Tengan en cuenta en la comprobación

que para estas nuevas instrucciones el valor de CPI es 4 y que añadirlas ocasiona un incremento de un ciclo en el CPI de los BRANCHs, pero no afectan al ciclo de reloj.

Datos

Instrucciones

$I_i^1$	$GPI^1$	$NI^1$	$I_i^2$	$CPI^2$	$NI^2$
ALU	3c	0'43 NI <sup>1</sup>		3c	0'75 0'43 NI <sup>1</sup> = 0'3225 NI <sup>1</sup>
LD	4c	0'21 NI <sup>1</sup>	ALU <sub>reg</sub>	4c	0'25 0'43 NI <sup>1</sup> = 0'1075 NI <sup>1</sup>
ST	4c	0'12 NI <sup>1</sup>	LO	4c	0'21 NI <sup>1</sup> - 0'1075 NI <sup>1</sup> = 0'1025 NI <sup>1</sup>
BR	4c	0'24 NI <sup>1</sup>	ST	4c	0'12 NI
			BR	5c	$\frac{0'24 NI}{0'8925} \cdot NI^2 = NI^2$

$$T_{CPU}^1 = N^3 \cdot [0'45 \cdot 3 + 0'21 \cdot 4 + 0'12 \cdot 4 + 0'24 \cdot 4] \cdot T_c$$

$$T_{CPU}^2 = N^2 \cdot [0'3225 \cdot 3 + 0'1035 \cdot 4 + 0'1025 \cdot 4 + 0'12 \cdot 4 + 0'24 \cdot 5] \cdot T_c.$$

$$S(n) = \frac{T_{CPU}^1}{T_{CPU}^2} = \frac{3'57}{3'4876} > 1$$

Luego  $T_{CPU}^2$  es menor luego la segunda alternativa es mejor.

**Ejercicio 9.** Suponga que, en el código siguiente, a[] es un array de números de 32 bits en coma flotante y b un número de 32 bits en coma flotante y que debería ejecutarse en menos de 0,5 segundos para  $N=10^9$ :

```
for (i=0; i<N; i++) a[i+2]=(a[i+2]+a[i+1]+a[i])*b;
```

(a) ¿Cuántos GFLOPS se necesitan para poder ejecutar el código en menos de 0,5 segundos?

(b) Suponiendo que este código en ensamblador tiene  $7N$  instrucciones y que se ha ejecutado en un procesador de 32 bits a 2 GHz. ¿Cuál es el número medio de instrucciones que el procesador tiene que poder completar por ciclo para poder ejecutar el código en menos de 0,5 segundos?

(c) Estimando que el programa pasa el 75% de su tiempo de ejecución realizando operaciones en coma flotante, ¿cuánto disminuiría como mucho el tiempo de ejecución si se redujesen un 75% los tiempos de las unidades de coma flotante?

Datos

Vectores y matrices de 32 bits

$T < 0.5$  s para  $N = 10^9$

$$a) GFlops = \frac{NFP}{T_{CPU} \cdot 10^9} = \frac{3 \cdot 10^9}{T_{CPU} \cdot 10^9} = \frac{3}{T_{CPU}}$$

$$T_{CPU} < 0.5 \Rightarrow \frac{3}{GFlops} < 0.5 \Leftrightarrow 3 < 0.5 \cdot GFlops \\ \frac{3}{0.5} < GFlops$$

Por tanto al menos  $6 \text{ GFlops}$ .

b) Cada iteración 7 instr.

Procesador 32 bits 2 GHz

$T < 0.5$  s

$$IPC = \frac{1}{ops}$$

$$T_{CPU} = NI \cdot CPS \cdot T_c = \frac{N^3}{IPC} \cdot f = \frac{7 \cdot 10^9}{2 \cdot 10^9} =$$

$$= \frac{7}{2 \cdot 10^9}$$

$$\frac{7}{2 \cdot 10^9} < 0.5$$

Por tanto necesita hacer como mínimo 7 instrucciones.

c)

$$\frac{T_b - T_p}{T_b} \approx 100$$

$$T_p = 0'25 T_b + 0'75 T_b = 0'4375 T_b$$

$$\left(1 - \frac{T_p}{T_b}\right) \cdot 100 = \left(1 - \frac{0'4375 T_b}{T_b}\right) \cdot 100 = 56.25\%$$

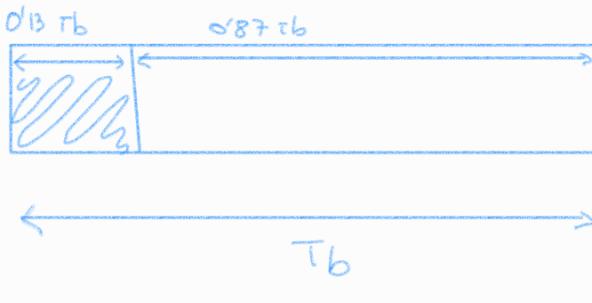
**Ejercicio 8.** Suponga que en los programas que constituyen la carga de trabajo habitual de un procesador las instrucciones de coma flotante consumen un promedio del 13% del tiempo del procesador.

(a) Ha aparecido en el mercado una nueva versión del procesador en la que la única mejora con respecto a la versión anterior es una nueva unidad de coma flotante que permite reducir el tiempo de las instrucciones de coma flotante a tres cuartas partes del tiempo que consumían antes. ¿Cuál es la máxima ganancia de velocidad que puede esperarse en los programas que constituyen la carga de trabajo si se utiliza la nueva versión del procesador?

(b) ¿Cuál es la máxima ganancia de velocidad con respecto a la versión inicial del procesador que, en promedio, puede esperarse en los programas debido a mejoras en la velocidad de las operaciones en coma flotante?

(c) ¿Cuál debería ser el porcentaje de tiempo de cálculo con datos en coma flotante en los programas para esperar una ganancia máxima de 4 en lugar de la obtenida en el apartado (b)?

(d) ¿Cuánto debería reducirse el tiempo de las operaciones en coma flotante con respecto a la situación inicial para que la ganancia máxima sea 2 suponiendo que en la versión inicial el porcentaje de tiempo de cálculo con coma flotante es el obtenido en (c)?



a)

$$S = \frac{T_b}{T_p} = \frac{T_b}{0'87T_b + 0'13T_b \cdot 0'875} = 10'336$$

$$\begin{aligned} p &= \frac{4}{3} \\ f &= 0'87 \end{aligned}$$

$$b) S = \frac{T_b}{0'87 T_b + 0} = 1'149$$

$$c) \frac{\frac{T_b}{p \cdot f T_b + (1-f) T_b}}{p} = 4 \quad \frac{p}{p f + (1-f)} = 4 \quad \frac{1}{f} = 4 \Rightarrow f = 25\% \rightarrow 0'75 = \frac{1-f}{p}$$

$\circlearrowleft$  Es despreciable

$$d) p? \text{ Despejar de } \frac{1-f}{p} = 0'75$$

**Ejercicio 4.** ¿Qué ocurriría en el problema anterior si el ciclo de reloj fuese únicamente un 10% mayor para la ALT2?

Datos

$$T_{CPU1} = 3'2 \text{ ns}^{-1} T_c^{-1}$$

$$S = \frac{T_{CPU1}}{T_{CPU2}} = \frac{3'2}{2'611} > 1$$

$$T_{CPU2} = 2'6 \text{ ns}^{-1} \cdot 1'1 T_c^{-1}$$

$T_{CPU2}$  sigue siendo mayor que  $T_{CPU1}$ ; por tanto no hay gran

cambio, esasí, la proporción es muy cercana a 1.

**Ejercicio 6.** Se ha diseñado un compilador para la máquina LOAD/STORE del problema anterior. Ese compilador puede reducir en un 50% el número de operaciones con la ALU, pero no reduce el número de LOADs, STOREs, y BRANCHs. Suponiendo que la frecuencia de reloj es de 50 Mhz. ¿Cuál es el número de MIPS y el tiempo de ejecución que se consigue con el código optimizado? Compárelos con los correspondientes del código no optimizado.

Datos

Instrucciones

$I_o^1$	CPI <sup>1</sup>	NI <sup>1</sup>
ALU	3c	$0'43 NI^1$
LD	4c	$0'21 NI^1$
ST	4c	$0'12 NI^1$
BR	4c	$0'24 NI^1$



$I_o^2$	CPI <sup>2</sup>	NI <sup>2</sup>
ALU	3c	$0'43 NI^1 \cdot 0'5$
LD	4c	$0'21 NI^1$
ST	4c	$0'12 NI^1$
BR	4c	$0'24 NI^1$

$$F = 50 \text{ MHz}$$

$$\begin{aligned} T_{CPU}^1 &= NI \cdot CPI \cdot T_c = \frac{NI \cdot CPI}{F} \\ &= T_c \cdot NI \cdot [0'21 \cdot 3 + 0'21 \cdot 4 + 0'12 \cdot 4 + 0'24 \cdot 4] \\ &= 2'91 NI^2 T_c \end{aligned}$$

$$MIPS = \frac{50}{2'91} = 14'006 \text{ MIPS}$$

$$T_{CPU}^1 = 3'57 NI^2 T_c$$

$$S = \frac{2'91}{3'57} < 1 \quad \text{es decir hay mejora reflejada en los tiempos}$$

$$MIPS = \frac{NI}{T_{CPU} \cdot 10^6} = \frac{F}{CPI \cdot 10^6}$$

$$\begin{aligned} MIPs^2 &= \frac{50}{[0'21 \cdot 3 + 0'21 \cdot 4 + 0'12 \cdot 4 + 0'24 \cdot 4]} \\ &= 17'18 \text{ MIPS} \end{aligned}$$

**Ejercicio 7.** En un programa que se ejecutan en un procesador no segmentado que funciona a 100 MHz, hay un 20% de instrucciones LOAD que necesitan 4 ciclos, un 15% de instrucciones STORE que necesitan 3 ciclos, un 40% de instrucciones con operaciones en la ALU que necesitan 6 ciclos, y un 25% de instrucciones de salto que necesitan 3 ciclos. (a) Si en las instrucciones que usan la ALU el tiempo en la ALU supone 4 ciclos, determine cuál es la máxima ganancia que se puede obtener si se mejora el diseño de la ALU de forma que se reduce su tiempo de ejecución a la mitad de ciclos. (b) Con qué porcentaje de instrucciones con operaciones en la ALU se podría haber obtenido en los cálculos del apartado (a) una ganancia mayor que 2? Razona tu respuesta.

Datos

$$F = 100 \text{ MHz} \Rightarrow T_c = \frac{1}{F} = 10^{-9} \text{ s}$$

$I_o^1$	CPI <sup>1</sup>	NI <sup>1</sup>
LD	4c	$0'2 NI^1$
ST	3c	$0'15 NI^1$
ALU	6c	$0'4 NI^1$
BR	3c	$0'25 NI^1$

$I_o^2$	CPI <sup>2</sup>	NI <sup>2</sup>
LD	4c	$0'2 NI^1$
ST	3c	$0'15 NI^1$
ALU	4c	$0'4 NI^1$
BR	3c	$0'25 NI^1$

Tiempo ejecución  $T_c$

$$S = \frac{T_{CPU}^2}{T_{CPU}^1} < 1 \Rightarrow \text{mejoría}$$

$$2 \leq \frac{4'4}{0'2 \cdot 4 + 0'15 \cdot 3 + x \cdot 4 + 0'25 \cdot 3} \Rightarrow$$

$$T_{CPU}^2 = NI^2 \left[ 0'2 \cdot 4 + 0'15 \cdot 3 + 0'4 \cdot 6 + 0'25 \cdot 3 \right] T_c = 4'4 \cdot T_c \cdot NI^2$$

$$T_{CPU}^2 = 3'6 NI^2 \cdot T_c$$

$$\Leftrightarrow 4 \leq \frac{4'4}{4x} \Leftrightarrow 1 \leq \frac{11}{x}$$

$$x \leq 11$$

Es decir un 110% de mej. de diseño

Si pusiera 6 ciclos:

$$4 \leq \frac{414}{6x}$$

$$24x \leq 414$$
$$x \leq \frac{414}{24} = 17\frac{1}{2}$$

Es decir, con un porcentaje de 18,3% teniendo 6 ciclos por construcción de lau.