

UNIVERSIDAD DE
MURCIA



MEMORIA PRÁCTICA 2

Arquitectura y Organización de Computadores

Por Eduardo González Sevilla y
José Antonio Mazón San Bartolomé

Curso 2018/2019
Grupo 2.1

ÍNDICE

ÍNDICE	1
BLOQUE I.....	2
1.....	2
2.....	2
3.....	2
BLOQUE II	3
4.....	3
5.....	3
6.....	4
BLOQUE III	4
7.....	4
8.....	5
9.....	5
10.....	5
BLOQUE III	6
11.....	6
12.....	6

BLOQUE I

1.

$$IPC = \frac{\text{Intrucciones confirmadas}}{\text{Tiempo de ejecución}}$$

$$IPC = \frac{18908478}{234340259}$$

$$IPC = 0,0807$$

2.

$$TFE = \frac{\text{Intrucciones emitidas} - \text{Intrucciones confirmadas}}{\text{Intrucciones emitidas}}$$

$$TFE = \frac{19619566 - 18908478}{19619566}$$

$$TFE = 0,0362$$

No coincide el número de instrucciones emitidas con el de confirmadas porque se producen fallos al predecir saltos e instrucciones, con lo que se emiten ciertas instrucciones que, al detectarse una excepción, deben ser canceladas.

3.

BLOQUE II

4.

En una caché de **escritura directa** (*write through*) las operaciones de escritura actualizan tanto la memoria caché como la memoria principal inmediatamente, mientras que en una caché de **postescritura** (*write back*) un bloque escrito en la caché solo se actualiza en la memoria principal cuando es desalojado de la caché.

Dicho lo cual, observamos que con una caché L1 de *postescritura* en el simulador obtenemos un tiempo de ejecución de $2,34394 \cdot 10^8$ ciclos y una tasa de aciertos del 93%, con 9.693.354 aciertos y 718.914 fallos, mientras que en la de escritura directa el tiempo de ejecución es de $2,43586 \cdot 10^8$ ciclos y tenemos que la tasa de aciertos es de un 83%, presentando 8.605.337 aciertos y 1.780.696 fallos.

En conclusión, ejecutando el simulador con una caché L1 obtenemos una disminución del 10% de la tasa de aciertos y un tiempo de ejecución un 3,92% mayor, con lo que **no es rentable en términos de eficiencia esta modificación**.

5.

- **L2 16 KB**: Tiempo de ejecución de $2,34394 \cdot 10^8$ ciclos. 255.537 aciertos y 463.380 fallos (**tasa de acierto L2 del 36 %**).

- **L2 32 KB**: Tiempo de ejecución de $1,68027 \cdot 10^8$ ciclos. 352.053 aciertos y 349.425 fallos (**tasa de acierto L2 del 50 %**).

- **L2 64 KB**: Tiempo de ejecución de $2,34956 \cdot 10^7$ ciclos. 654.216 aciertos y 9.315 fallos (**tasa de acierto L2 del 99 %**)

A la vista de los resultados, es evidente que a medida que aumentamos el tamaño de la L2 el tiempo de ejecución se reduce, así como aumenta la tasa de acierto. La explicación es sencilla: **con una L2 más grande cubrimos de forma más eficiente los fallos de la L1**. Al aprovechar la localidad espacial de los bloques de memoria y tener espacio suficiente en la L2 para no tener que estar reemplazando bloques continuamente, cuando la L1 falle es muy probable que la L2 contenga el bloque de memoria al que se quiere acceder, con lo que el tiempo de acceso al bloque se reduce, y, en consecuencia, el tiempo de ejecución también.

6.

- **L1 4 KB**: Tiempo de ejecución de $2,34956 \cdot 10^7$ ciclos. 9.500.085 aciertos y 663.530 fallos (**tasa de acierto L1 del 93 %**).
- **L1 8 KB**: Tiempo de ejecución de $2,29818 \cdot 10^7$ ciclos. 9.584.840 aciertos y 579.280 fallos (**tasa de acierto L1 del 94 %**).
- **L1 16 KB**: Tiempo de ejecución de $2,37462 \cdot 10^7$ ciclos. 9.733.626 aciertos y 430.663 fallos (**tasa de acierto L1 del 96 %**)

Como ocurría con la L2, al aumentar el tamaño de la caché L1 obtenemos una mejora en el tiempo de ejecución, pues, lógicamente, **la caché tiene más espacio para almacenar bloques que probablemente vayan a ser accedidos**.

En concreto, la L1 de 8KB reduce el tiempo de ejecución en un 2,23% y aumenta en un 1% la tasa de acierto respecto a la L1 de 4KB.

Además, la L1 de 16 KB aumenta en un 2% la tasa de acierto respecto a la L1 de 8KB, aunque también aumenta su tiempo de ejecución en un 3,326%, con lo que, en términos de rendimiento, **la L1 de 8KB es la que mejores resultados ofrece en este caso**.

BLOQUE III

7.

- **ROB 16 entradas**: Tiempo de ejecución de 22.928.503 ciclos. 0,8246 instrucciones por ciclo (IPC) y TFE de 0,0041.
- **ROB 32 entradas**: Tiempo de ejecución de 19.903.158 ciclos. 0,95 instrucciones por ciclo (IPC) y TFE de 0,0094. Observamos un **speedup del 15,2%** respecto al de ROB de 16 entradas, aunque su TFE es un 129,26% peor.
- **ROB 64 entradas**: Tiempo de ejecución de 16.362.349 ciclos. 1,1556 instrucciones por ciclo (IPC) y TFE de 0,01803. Observamos un **speedup del 40,14%** respecto al de ROB de 16 entradas, aunque su TFE es un 339,75% peor.

Según los datos obtenidos, **podemos concluir que la mejor opción es la de un ROB de 64 entradas**.

8.

- **2 instrucciones por ciclo**: Tiempo de ejecución de 16.362.349 ciclos. 1,1556 instrucciones por ciclo (IPC) y TFE de 0,01803.

- **4 instrucciones por ciclo**: Tiempo de ejecución de 15.060.398 ciclos. 1,2555 instrucciones por ciclo (IPC) y TFE de 0,06093. Observamos un ***speedup del 8,644%*** respecto al de 2 instrucciones por ciclo, aunque su TFE es un 237,807% peor.

- **8 instrucciones por ciclo**: Tiempo de ejecución de 16.227.814 ciclos. 1,1556 instrucciones por ciclo (IPC) y TFE de 0,15822. Observamos un ***speedup del 0,829%*** respecto al de 2 instrucciones por ciclo, aunque su TFE es un 777,148% peor.

A la vista de los resultados, **concluimos que la mejor opción es la de 4 instrucciones por ciclo.**

9.

- **2 ALUs**: Tiempo de ejecución de 14.969.389 ciclos. 1,263142 instrucciones por ciclo (IPC) y TFE de 0,060663. Observamos un ***speedup del 0,607%*** respecto al mejor del ejercicio anterior, y un TFE un 12,25% mejor.

- **2 FPU's**: Tiempo de ejecución de 15.058.904 ciclos. 1,2555 instrucciones por ciclo (IPC) y TFE de 0,06066. Observamos un ***speedup del 0,0099%*** respecto al de 2 instrucciones por ciclo, y un TFE un 0,449% mejor.

- **2 ADDRs**: Tiempo de ejecución de 13.762.782 ciclos. 1,37388 instrucciones por ciclo (IPC) y TFE de 0,061902. Observamos un ***speedup del 9,428%*** respecto al de 2 instrucciones por ciclo, aunque su TFE es un 1,578% peor.

Así, **concluimos que la mejor opción es la de duplicar el número de unidades de generación de direcciones.**

10.

Con un **predictor 2bitagree de 64 contadores** obtenemos un tiempo de ejecución de 13.635.393 ciclos.

En cuanto al IPC, tenemos 1,38672 instrucciones por ciclo.

Además, observamos que su TFE es de 0,020564.

Por tanto, obtenemos un ***speedup del 0,934%%*** respecto al mejor del ejercicio anterior, y un TFE un 201,01% mejor.

BLOQUE III

11.

Configuracion Inicial

PC = 0,080688133

recio 0\$

4.

Precio = 1000\$

IPC = 0,07764261

5.

l2size 16	----->	l2size 32	----->	l2size 64
Precio 0 \$		Precio 3k€		Precio 6k€
IPC = 0,080688133		0,112568205		0,806597122

6.

l1size 4	----->	l1size 8	----->	l1size 16
Precio 6k€		Precio 9k€		Precio 15k€
IPC = 0,806597122		0,824671284		0,080688133

7.

Active List 16	----->	Active List 32	----->	Active List 64
Precio 9k€		Precio 11k€		Precio 13k€
IPC = 0,824671284		0,95		1,1556

8.

In Ciclo 2	----->	In Ciclo 4	----->	In Ciclo 8
Precio 13k€		Precio 14k€		Precio 17k€
IPC = 1,155609014		1,255509848		1,16518947

9.

2 ALUs	----->	2 FPUs	----->	UGD
Precio 15k€		Precio 15k€		Precio 15k€
PC = 1,263142938		1,255634407		1,373884873

10.

64 Contadores

Precio 16k€

IPC = 1,386720427

12.

Si nos basamos en las configuraciones de los ejercicios anteriores la que nos da mas IPC es :

numnodes 1

bpbtype static

numalus 1

numfpus 1
numaddrs 1
linesize 32
l1type WB
l1size 8
l1assoc 2
l1taglatency 3
l2size 64
l2assoc 2
l2taglatency 9
l2datalatency 2
memorylatency 300
STOPCONFIG 1

OPTIONS="-z\${CONFIG_FILE}" # fichero de configuración

OPTIONS="\${OPTIONS} -K" # activa cargas especulativas

OPTIONS="\${OPTIONS} -p" # activa prefetch controlado por hardware

OPTIONS="\${OPTIONS} -n" # activa la simulación de accesos a cachés privadas

OPTIONS="\${OPTIONS} -i2" # número de instrucciones en la etapa de fetch por ciclo

OPTIONS="\${OPTIONS} -a32" # tamaño de la lista de instrucciones activas (ROB)

Con un precio de 11k€ y 0,95 de IPC.

El IPC original era de 0,080688133 por lo que la mejora es del 1036,183395291% Una mejora bastante importante aunque si la ponemos en contraste con los 12000€ de aumento tenemos que el 1% de mejora nos sale a 11,85€