

Problemas Tema 3

Problema 2. Repaso Cache

Disponemos de un procesador de 16 bits con un bus de direcciones de 16 bits. Este procesador tiene una memoria cache de datos con las siguientes características:

- Tamaño de bloque = 16 bytes
- Asociatividad = 2 (reemplazo = LRU)
- Número de líneas = 8
- Política de escritura: write through + write no allocate

Suponiendo que el contenido de la cache es el siguiente:

conjunto 0		conjunto 1		conjunto 2		conjunto 3	
EC8	1	EC5	1	EC6	0	EC7	1
AB4	0	libre	0	AB2	1	libre	0

Teniendo en cuenta que:

- En el contenido de la MC para simplificar hemos dejado el número de bloque de memoria en vez del tag.
- El bit a 1, en el contenido de la cache, indica que es la línea más recientemente referenciada.
- R-byte (lectura de 1 byte), R-word (lectura de 2 bytes), W-byte (escritura de 1 byte), W-word (escritura de 2 bytes).
- El tamaño de las lecturas (y escrituras) se ha de indicar en bytes.

Rellenad la siguiente tabla:

Tipo	@ en hex	Bloque de memoria	Conjunto de MC	Acerto / Fallo	Lectura de MP			Escritura en MP		
					si / no	@	tamaño	si / no	@	tamaño
R byte	8890	889	1	F	Si	8890	16	No		
W word	EC51	EC5	1	H	No			Si	EC51	2
W byte	EC62	EC6	2	H	No			Si	EC62	1
W word	23D3	23D	1	F	No			Si	23D3	2
W byte	ABA4	ABA	2	F	No			Si	ABA4	1
R word	ABA5	ABA	2	F	Si	ABA5	16	No		
R byte	23D6	23D	1	F	Si	23D6	16	No		
W word	EC57	EC5	1	H	No			Si	EC57	2
R byte	EC68	EC6	2	H	No			No		
R word	8899	889	1	F	Si	8890	16	No		

Indicad también cómo queda la cache después de realizar los 10 accesos a memoria:

conjunto 0		conjunto 1		conjunto 2		conjunto 3	
EC8	1	EC5	0	EC6	1	EC7	1
AB4	0	889	1	ABA	0		

Problema 10. Predicción de vía

Una CPU funciona a un voltaje de 1,2 V y una frecuencia de 2GHz. Se ha determinado que esta CPU tiene una corriente de fugas de 3 A y que a pleno rendimiento tiene una carga capacitiva equivalente de 5 nF (nanoFaradios).

- a) **Calculad la potencia media dinámica (debida a conmutación), la potencia media estática (debida a fugas) y la potencia media total.**

$$P_{\text{DIN}} = C \cdot V^2 \cdot F = 5 \times 10^{-9} \cdot 1.2^2 \cdot 2 \times 10^9 = 14.4 \text{ W}$$

$$P_{\text{FUGA}} = I \cdot V = 3 \cdot 1.2 = 3.6 \text{ W}$$

$$P_{\text{TOTAL}} = 14.4 + 3.6 = 18 \text{ W}$$

Se desea integrar esta CPU con una memoria cache de datos 2-asociativa de 128 KB de capacidad y un tamaño de bloque de cache de 64 bytes. Las direcciones generadas por la CPU son de 48 bits. La corriente de fugas de la memoria RAM estática es de 3 μ A (microAmperios) por bit. La energía consumida durante un acceso a la memoria de etiquetas es de 5 nJ (nanoJoules) por vía y la consumida durante un acceso a la memoria de datos es de 25 nJ por vía.

- b) **Calculad el número de conjuntos, el de bloques de cache, el de vías y el de bloques por vía.**

bloques: $2^{48} / 2^6 = 2^{42}$ bloques = 2048

2048 bloques / 2 bloques/conjunto = 1024 conjuntos

1024 conjuntos \Rightarrow 1024 bloques por vía

- c) **Dibujad una dirección indicando claramente los campos usados para seleccionar el byte dentro del bloque, seleccionar el conjunto de la cache y los bits usados como etiqueta.**

TAG: 48 , Conj: 12 , Byte: 4

- d) **Calculad el tamaño en bits de la memoria de datos y el de la memoria de etiquetas de una vía (por simplicidad ignoraremos el bit de validez y otros bits de control).**

MEM DADOS = bloques por vía \cdot bytes por bloque = $1024 \cdot 64 = 65536 \text{ B} \Rightarrow 524288 \text{ bits}$

MEM TAG = bloques por vía \cdot bits bloque = $1024 \cdot 32 = 32768 \text{ bits}$

- e) **Calculad la potencia media estática (debida a fugas) de la cache.**

$P_{\text{FUGA}} = I \cdot V \cdot \text{Vías} =$

$I =$

Se desean comparar diversas implementaciones alternativas de cache de datos 2-asociativa: **paralela**, **serie**, y con **predicador de vía**. Para comparárlas se usa un **benchmark** con 4×10^9 instrucciones dinámicas que realiza 10^9 accesos de datos a memoria y 2×10^9 operaciones aritméticas de punto flotante. Este **benchmark** tiene un 10% de fallos en la cache descrita anteriormente.

En la implementación **paralela**, se accede simultáneamente tanto a las memorias de etiquetas como las de datos de ambas vías. Un acceso a cache se realiza en 1 ciclo y la penalización media por fallo de cache es de 20 ciclos. El **benchmark** ejecutado con la implementación **paralela** de la cache ha tardado 5 segundos.

- f) **Calculad los MFLOPS de la implementación paralela.**

$\text{MFlops} = (2 \times 10^9 / 5) \cdot 10^{-6} = 400 \text{ MFlops}$

- g) **Calculad el CPI de la implementación paralela y el CPI que obtendríamos con una memoria ideal (CPI_{ideal}) en donde todos los accesos tardan 1 ciclo.**

- h) **Calculad la energía dinámica consumida por un acceso a la cache. Para simplificar asumiremos que todos los accesos consumen lo mismo sean acierto o fallo, la energía extra consumida en acceder a memoria principal en caso de fallo está fuera de los objetivos de este problema.**

$E = 2 \cdot 5 \times 10^{-9} + 2 \cdot 25 \times 10^{-9} = 60 \times 10^{-9} \text{ J}$

- i) **Calculad la potencia (dinámica) media consumida en acceder a la cache**

ACCESOS PER S: $10^9 / 5 = 0.2 \times 10^9 \text{ accesos/s}$ $P_{\text{ot}} = 0.2 \times 10^9 / (60 \times 10^{-9}) = 12 \text{ W}$

- j) **Calculad la potencia media total (estática+dinámica) consumida por el sistema CPU-cache.**

$P_{\text{TOTAL}} = P_{\text{DIN}} + P_{\text{EST}} + P = 12 + P_{\text{EST}} + 18 = 30 \text{ W}$

- k) **Calculad la energía total consumida para ejecutar el benchmark y la eficiencia en MFLOPS/Watt.**

$E = P \cdot t = \dots$

En la implementación **serie** un acceso tarda 2 ciclos. En el primer ciclo se accede a las memorias de etiquetas de ambas

vías. Una vez determinada la vía que contiene el dato, en el segundo ciclo se accede solo a la memoria de datos de dicha vía. La penalización en caso de fallo sigue siendo de 20 ciclos ya que en el primer ciclo del acceso ya se puede determinar si es acierto o fallo. Obsérvese que respecto la implementación **paralela**, los aciertos tienen una penalización de 1 ciclo.

l) **Calculad el tiempo de ejecución y los MFLOPS de la implementación serie.**

m) **Calculad la energía consumida por un acceso a la cache.**

$$E = 2.5 \times 10^{-7} \text{ J} + 2.5 \times 10^{-9} \text{ J} = 3.5 \times 10^{-7} \text{ J}$$

n) **Calculad la potencia (dinámica) media consumida en acceder a la cache**

o) **Calculad la potencia media total consumida por el sistema CPU-cache.**

p) **Calculad la energía total consumida para ejecutar el *benchmark* y la eficiencia en MFLOPS/Watt.**

En la implementación con **predictor de vía**, este predice la vía probable en que se encuentra el dato y se accede solo a las memorias de etiquetas y de datos de esa vía. En caso de fallo del predictor hay que acceder a la otra vía (memorias de etiquetas y datos). El predictor usado consiste en una memoria de $8k \times 1$ bits indexado con los bits bajos del PC de las instrucciones de acceso a memoria. Este predictor tiene una tasa de aciertos del 80% del total de accesos a memoria y cada acceso al predictor consume 1nJ. En esta implementación se pueden dar las siguientes situaciones:

- El predictor acierta: se accede a 1 sola vía (datos+etiquetas) en 1 ciclo (no hay penalización) y al comprobar los tags se comprueba que es acierto de cache.
- El predictor falla pero es acierto de cache: El acceso tarda 2 ciclos, en el primero se accede a la vía probable (aunque equivocada) al comprobar los tags se descubre que es fallo (de vía) y en el segundo se accede a la vía correcta y se descubre que es acierto de cache (1 ciclo de penalización).
- El predictor falla y además es fallo de cache: En el primer ciclo se accede a la vía probable (aunque incorrecta), en el segundo ciclo se accede a la otra vía y se descubre que es fallo de cache (con lo que hay que acceder a memoria principal). Obsérvese que en este caso la penalización es de 21 ciclos ya que no se descubre el fallo de cache hasta que se accede a la 2ª vía.

q) **¿Puede darse al caso de que un acierto del predictor de vía sea fallo de cache? ¿porqué?**

no habría

r) **Calculad la potencia media estática (debida a fugas) del predictor y compárala con la de la cache (se calcula de la misma forma ya que se ha empleado el mismo tipo de memoria estática).**

$$I_{FUGA} = 2192 \cdot 3 \times 10^{-6} = 24'6 \times 10^{-3} \text{ A}$$

$$P_{MEST} = I \cdot V = 24'6 \times 10^{-3} \cdot 1'2 = 29'5 \times 10^{-3} \text{ W}$$

s) **Calculad el tiempo de ejecución y los MFLOPS de la implementación con predictor de vía.**

t) **Calculad la energía consumida por un acceso en que el predictor acierta y uno en que el predictor falla (tener en cuenta la energía consumida por el acceso al predictor). Calcular también la energía media consumida por acceso.**

$$\begin{aligned} \text{Hit: } & 1 \times 10^{-9} + 5 \times 10^{-9} + 25 \times 10^{-9} = 31 \times 10^{-9} \text{ J} \\ \text{Miss: } & 1 \times 10^{-9} + 10 \times 10^{-9} + 80 \times 10^{-9} = 91 \times 10^{-9} \text{ J} \end{aligned} \quad \text{Energía: } 0'8 \cdot 31 \times 10^{-9} + 0'2 \cdot 91 \times 10^{-9} = 37 \times 10^{-9} \text{ J}$$

u) **Calculad la potencia (dinámica) media consumida en acceder a la cache**

$$\text{Access} = 10^{-9} / 5'12 \text{ s} = 0'196 \times 10^{-9}$$

$$P_{MDIN} = E/t = 0'096 / 37 \times 10^{-9} = 2'5 \text{ W}$$

v) **Calculad la potencia media total consumida por el sistema CPU-cache (acuérdate de las fugas del predictor).**

$$P_{MTOTAL} = 2'5 + 29'5 \times 10^{-3} + P_{FUGA} + P =$$

w) **Calculad la energía total consumida para ejecutar el *benchmark* y la eficiencia en MFLOPS/Watt.**

x) **Calculad la ganancia en eficiencia energética de la implementación serie sobre la paralela y la de predicción devía sobre la serie.**

Problema 11. Caches segmentadas

En un procesador X el camino crítico, y por tanto el tiempo de ciclo, está limitado por la memoria cache de datos. El tiempo de los componentes de la memoria cache de datos se desglosa de la siguiente forma:

Componente	Tiempo
Memoria de etiquetas	0,30 ns
Selección de vía	0,15 ns
Memoria de datos	0,45 ns
Mux/Driver de datos	0,10 ns
Registro de desacoplo (en caso que se use)	0,05 ns

Queremos evaluar el rendimiento de 4 posibles implementaciones de la memoria cache para el procesador X:

- Procesador X1: La cache de datos tiene una implementación paralela y el tiempo de acceso a la cache es de 1 ciclo de procesador
- Procesador X2: La cache de datos está segmentada en 2 etapas y el tiempo de acceso a la cache es de 2 ciclos de procesador
- Procesador X3: La cache de datos está segmentada en 3 etapas y el tiempo de acceso a la cache es de 3 ciclos de procesador
- Procesador X4: La cache de datos está segmentada en 4 etapas y el tiempo de acceso a la cache es de 4 ciclos de procesador

- a) Calculad el tiempo de ciclo de la cache de datos y el tiempo total de un acceso para los procesadores X1, X2, X3 y X4, usando la distribución mas adecuada de los componentes por etapas.

$$X1: t_c = 0,45 \text{ ns} + 0,15 = 0,6 \text{ ns} = t_{\text{access}}$$

$$X3: t_{\text{AGS}} = 0,3 \text{ ns}, t_{\text{DADES}} = 0,5 \text{ ns}, t_{\text{MUX}} = 0,15 \text{ ns}$$

$$X2: t_{\text{AGS}} = 0,3 \text{ ns}, t_{\text{DADES}} = 0,6 \text{ ns}, t_{\text{MUX}} = 0,15 \text{ ns}$$

$$t_c = 0,6 \text{ ns} \Rightarrow t_{\text{access}} = 0,6 \text{ ns} + 0,15 \text{ ns} = 0,75 \text{ ns}$$

$$t_c = 0,5 \text{ ns}$$

$$t_{\text{access}} = 0,5 \times 3 = 1,5 \text{ ns}$$

$$X4: t_{\text{AGS}} = 0,3 \text{ ns}, t_{\text{VIA}} = 0,2 \text{ ns}, t_{\text{DADES}} = 0,5 \text{ ns}, t_{\text{MUX}} = 0,15 \text{ ns}$$

$$t_c = 0,5 \text{ ns} \Rightarrow t_{\text{access}} = 0,5 \times 4 = 2 \text{ ns}$$

- b) Razonad porque descartamos las opciones X2 y X4 para el resto del problema

- c) Calculad la frecuencia de reloj de los procesadores X1 y X3

$$F_{X1} = \frac{1}{t_c} = 1,67 \text{ GHz} \quad F_{X3} = \frac{1}{t_c} = 2 \text{ GHz}$$

Un programa P que ejecuta 2×10^9 instrucciones tiene un 60% de instrucciones aritméticas, un 20% de instrucciones de salto y un 20% de instrucciones de acceso a memoria (Load/Store). Las instrucciones aritméticas tardan 5 ciclos, las de salto 4 y las de memoria 4 ciclos + los ciclos del acceso a la cache.

- d) Calculad el CPI del programa P para los procesadores X1 y X3 suponiendo que nunca hay fallos en la cache de datos.

$$CPI_{X1} = 0,6 \times 5 + 0,2 \times 4 + 0,2 \times 5 = 4,8 \text{ ciclos/instr}$$

$$CPI_{X3} = 0,6 \times 5 + 0,2 \times 4 + 0,2 \times 5 = 5,2 \text{ ciclos/instr}$$

- e) Calculad el speedup de X3 sobre X1 en % suponiendo que nunca hay fallos en la cache de datos

$$T_{X1} = 2 \times 10^9 \times 4,8 = 9,6 \times 10^9 \text{ ns}$$

$$T_{X3} = 2 \times 10^9 \times 5,2 = 10,4 \times 10^9 \text{ ns}$$

$$Speedup = \frac{T_{X1}}{T_{X3}} = \frac{9,6}{10,4} = 0,923 \text{ (1,07\%)}$$

Sabemos que el programa P tiene un 10% de fallos en la cache de datos utilizada y que el tiempo de penalización en ambos casos es de 60 ciclos.

- f) Calculad el speedup real de X3 sobre X1 en % teniendo en cuenta la jerarquía de memoria completa.

$$CPI_{X1} = CPI_{\text{data}} + CPI_{\text{memoria}} = 4,8 + 0,2 \times 0,1 \times 60 = 6 \text{ ciclos/instr}$$

$$T_{X1} = 2 \times 10^9 \times 6 = 12 \times 10^9 \text{ ns}$$

$$CPI_{X3} = 5,2 + 0,2 \times 0,1 \times 60 = 6,4 \text{ ciclos/instr}$$

$$T_{X3} = 2 \times 10^9 \times 6,4 = 12,8 \times 10^9 \text{ ns}$$

$$Speedup = \frac{6}{6,4} = 0,9375 = 93,75\%$$

$12 \times 10^9 \text{ bits}$, 5 ciclos
 $400 \times 10^6 \text{ bits}$, 4 ciclos
 $400 \times 10^6 \text{ Load/store}$, 4 ciclos + 4 ciclos acceso a la cache.