L1 cache 注重 減少 hit time

L2 cache 注重 減少 miss rate (防止太大的 miss penalty).

重點七：使用多層快取來減少失誤處罰時間

- 在多數情況下快取都與微處理器放在同一個晶片上。為了縮小處理器與DRAM的存取時間之差距，許多微處理器支援了一層額外的快取。第二層快取可以放在同一個晶片上、或是在晶片外面以一組SRAM構成。
- 當主要的快取發生失誤時，第二層快取就會被存取。如果第二層快取有包含所要的資料，那麼第一層快取的失誤處罰時間將會等於第二層快取的存取時間，而這時間遠比主記憶體的存取時間來的短。
- 如果主要與次要快取都沒包含所要的資料，就必須存取主記憶體而造成較大的失誤處罰時間。因此，在兩層快取的設計上，次要快取必須足夠大才能避免花更長時間的記憶體存取，也才能減少主要快取的失誤處罰時間。
- 設計主要快取與次要快取的考量是相當不同的。兩層快取結構可讓(主要快取注重在減小命中時間來縮短時脈週期)，並讓(次要快取注重在失誤率，以降低存取記憶體花太長時間的影響)
- 因為次要快取的存在，有效的降低了主要快取的失誤處罰時間，這樣可以允許主要快取較小，有較高的失誤率。對次要快取來說，因為主要快取的存在，所以存取時間相對的變的不重要，因為次要快取的存取時間只影響到主要快取的失誤處罰時間，並不會影響到主要快取的命中時間。

多層快取的效能

練習

[101東華資工]

Suppose we have a processor with a base CPI of 1.0, assuming all references hit in the primary cache, and a clock rate of 4 GHz. Assume a

= 0.25 ns

main memory access time of 100 ns, including all the miss handling. Suppose the *miss rate per instruction* at the primary cache is 2%. How much faster will the processor be if we add a secondary cache that has a 5 ns access time for either a hit or a miss and is large enough to reduce the miss rate to main memory to 0.5%.

Answer

到主記憶體的處罰時間為 100/0.25 = 400 clock cycles

對只有一層快取處理器的有效 CPI 為 $CPI_1 = 1 + 2\% \times \frac{100}{0.25} = 9.$

Total CPI = Base CPI + Memory-stall cycles per instruction

= 1.0 + Memory-stall cycles per instruction

= 1.0 + 2% × 400 = 9.0

在有兩層快取的狀況下，在主要快取發生失誤時，資料若可以在次要快取中找到則第二層快取失誤的處罰時間為 $CPI_2 = 1 + 2\% \cdot \frac{5}{0.25} + 0.5\% \cdot \frac{100}{0.25}$

5/0.25 = 20 clock cycles $= 3.4$

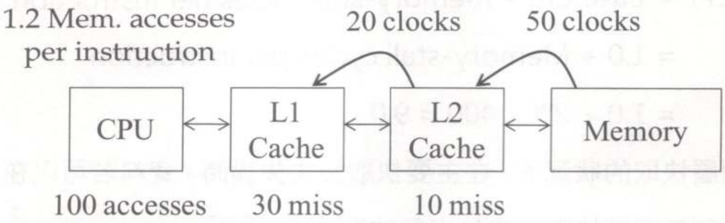
如果此失誤最後需要到主記憶體，那麼總共的處罰時間將會等於次要快取與主記憶體的存取時間的總和。因此，在兩層的快取中，total CPI 為每一層快取造成暫停的總週期數與 base CPI 的總和： $\frac{9}{3.4}$ #

Total CPI = 1 + Primary stalls per instruction + Secondary stalls per instruction = 1 + 2% × 20 + 0.5% × 400 = 1 + 0.4 + 2.0 = 3.4

因此，有包含次要快取的處理器會快 9.0/3.4 = 2.6

另解：我們可用存取次要快取命中的((2% − 0.5%) × 20 = 0.3)與到主記憶體存取的暫停週期數(0.5% × (20 + 400) = 2.1)加在一起來算暫停週期數。所以總和為 1.0 + 0.3 + 2.1 一樣等於 3.4。

- 上一題的範例是假設合併式的快取(combined cache)，且使用miss rate per instruction來計算各層快取的暫停時間，以下說明各層快取miss rate per instruction的定義。
- 如下圖所示，假設CPU的100次存取其中30次在L1快取找不到、10次在L2快取找不到，而L1快取的失誤處罰時間，也就是從L2搬一個區塊至L1快取所需時間為20 clocks，L2快取的失誤處罰時間，也就是從記憶體搬一個區塊至L2快取所需時間為50 clocks，另外假設一個指令平均存取記憶體1.2次。



Total stall cycles = L1 stall cycles + L2 stall cycles = L1 misses × L1 miss penalty + L2 misses × L2 miss penalty = 30 × 20 + 10 × 50

Stall cycle per access = Total stall cycles/number of CPU access

$$= \underset{\substack{\uparrow \\ \text{L1 miss rate}}}{\left(\frac{30}{100}\right)} \times 20 + \underset{\substack{\uparrow \\ \text{L2 miss rate}}}{\left(\frac{10}{100}\right)} \times 50$$

Stall cycle per instruction

= Memory access per instr. × Stall cycle per access

$$= \underset{\substack{\uparrow \\ \text{L1 miss rate} \\ \text{per instruction}}}{\left(1.2 \times \frac{30}{100}\right)} \times 20 + \underset{\substack{\uparrow \\ \text{L2 miss rate} \\ \text{per instruction}}}{\left(1.2 \times \frac{10}{100}\right)} \times 50$$

練習

Consider a processor with the following parameters:

	Base CPI, no memory stalls	Processor Speed	Main memory access time	1st-level cache miss rate per instruction	Second-level cache, direct-mapped speed	Global miss rate with 2nd-level cache, direct-mapped	2nd-level cache, 8-way set associative speed	Global miss rate with 2nd-level cache, 8-way set associative
a.	2.0	3GHz	125ns	5%	15 cycles	3.0%	25 cycles	1.8%
b.	2.0	1GHz	100ns	4%	10 cycles	4.0%	20 cycles	1.6%

(1) Calculate the CPI for the processor in the table using: ① only a first-level cache, ② a second-level direct-mapped cache, and ③ a second-level eight-way set-associative cache.

(2) It is possible to have an even greater cache hierarchy than two levels. Given the processor above with a second-level, direct-mapped cache, a designer wants to add a third-level cache that takes 50 cycles to access and will reduce the global miss rate to 1.3%. Would this provide better performance? In general, what are the advantages and disadvantages of adding a third-level cache?

better performance.

complex cache coherng

More expansive

(3) In older processors such as the Intel Pentium or Alpha 21264, the second level of cache was external (located on a different chip) from the main processor and the first-level cache. While this allowed for large second-level caches, the latency to access the cache was much

higher, and the bandwidth was typically lower because the second-level cache ran at a lower frequency. Assume a 512 KB off-chip second-level cache has a global miss rate of 4%. If each additional 512 KB of cache lowered global miss rates by 0.7%, and the cache had a total access time of 50 cycles, how big would the cache have to be to match the performance of the second-level direct-mapped cache listed in the table? Of the eight-way set-associative cache?

Answer

(1)

a.	Memory miss cycles: 125 ns × 3G = 375 ① Total CPI: 2.0 + 375 × 5% = 20.75 ✓ ② Total CPI: 2.0 + 15 × 5% + 375 × 3% = 14 ✓ ③ Total CPI: 2.0 + 25 × 5% + 375 × 1.8% = 10 ✓
b.	Memory miss cycles: 100 clock cycles ① Total CPI: 2.0 + 100 × 0.04 = 6.0 ② Total CPI: 2.0 + 100 × 0.04 + 10 × 0.04 = 6.4 ③ Total CPI: 2.0 + 100 × 0.016 + 20 × 0.04 = 4.4

(2)

a.	Total CPI: 2.0 + 15 × 5% + 50 × 3% + 375 × 1.3% = 9.125 ✓ This would provide better performance, but may complicate the design of the processor. This could lead to: more complex cache coherency, increased cycle time, larger and more expensive chips.
b.	Total CPI: 2.0 + 100 × 0.013 + 10 × 0.04 + 50 × 0.04 = 5.7 This would provide better performance, but may complicate the design of the processor. This could lead to: more complex cache coherency, increased cycle time, larger and more expensive chips.

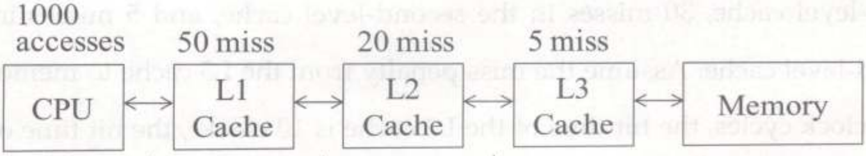
(3)

a.	Total CPI: 2.0 + 50 × 5% + 375 × (4% – 0.7% × n) n = 2 → 1.5 MB L2 cache to match direct-map n = 4 → 2.5 MB L2 cache to match 8-way
b.	Total CPI: 2.0 + 50 × 0.04 + 100 × (0.04 – 0.007 × n) n = 2 → 1.5 MB L2 cache o match direct-map n = 5 → 3 MB L2 cache to match 8-way

註(3)a：Let 2.0 + 50 × 5% + 375 × (4% – 0.7% × n) = 14 → n = 2.1
Let 2.0 + 50 × 5% + 375 × (4% – 0.7% × n) = 10 → n = 3.6

區域失誤率與全域失誤率

- **Global miss rate (GMR):** The fraction of references that miss in all levels of a multilevel cache. $\frac{\text{該level miss次數}}{\text{全部 access 次數}}$
- **Local miss rate (LMR):** The fraction of references to one level of a cache that miss; used in multilevel hierarchies. $\frac{\text{該 level miss 次數}}{\text{該 level access 次數}}$
- 以下例舉Global及Local miss rate的計算及轉換的公式。



GMR	50/1000	20/1000	5/1000
LMR	50/1000	20/50	5/20

L1 GMR = L1 LMR
L2 GMR = L1 LMR × L2 LMR
L3 GMR = L1 LMR × L2 LMR × L3 LMR

$$GMR = \frac{\text{該level miss次數}}{\text{全部 access 次數}} \qquad LMR = \frac{\text{該level miss 次數}}{\text{該level access 次數}}$$