Acos Workshop XI.

Malchenko Oleg, 193

№ 1.

Assume, the miss rate of an instruction cache is 2% and the miss rate of the DATA cache is 4%. If a processor has a CPI of 2 without any memory stalls, and the miss penalty is 100 cycles for all misses. Determine how much faster a processor would run with a perfect cache that never missed. Assume that the frequency of all loads and stores is 36%.

$I_{miss} = 2\%$.

$D_{miss} = 4\%$.

Penalty = 100 cycles

CPI = 2. (no memory stalls)

Freq. = 36 %.

Solution:

Let's calculate the total number of penalty cycle per instruction in each cache type:

Miss cycles in Data & Instruction cache:

- MCIC = 0.02·100 = 2 (cycles)
- MCDC = 0.04·100·0.36 = 1.44 (cycles) DATA CACHE
- ACPI = CPI + MCIC + MCDC = 2+2+1.44 = 5.44, that is we

may conclude that the perfect CPU would be: $n = \frac{ACPI}{CPI} = \frac{5.44}{2} = 2.72$ times faster than nonideal CPU.

Answer: 2.72 times faster.

№ 2

Find the AMAT for a processor with 1 n.sec. clock time, a miss penalty of 20 clock cycles, a miss rate of 0.05 misses per instruction, and a cache access time (including hit detection) of 1 clock cycle. Assume that the read and write miss penalties are the same, and ignore other write stalls.

$I_{miss} = 5\%$

Penalty = 20 cycles.

CPU [1ns]

A\HDT = 1 clock cycle

Solution:

$$AMAT = t_{cache} + MR_{cache}[t_{MM} + MR_M(t_{VM})].$$

$$AMAT = 1 + 0.05[20+0] = 1+1 = 2\,ns.$$

Answer: AMAT = 2 ns.

№ 3.

Use the system with the access times of $\overset{t_c}{1}$, $\overset{t_{MM}}{10}$, and $\overset{t_{VM}}{100}$ cycles for the L1 and L2 cache, and main memory, respectively. Assume that the L1 and L2 caches have miss rates of 5%, and 20% respectively. Specifically, of the 5% of accesses that miss the L1 cache, 20% of those also miss the L2 cache. What is the AMAT?

Solution:

- We know that: $AMAT = t_{cache} + MR_{cache}[t_{MM} + MR_M(t_{VM})]$.

In our case: $t_{cache} = 1$ cycles, $t_{MM} = 10$ cycles, $t_{VM} = 100$ cycles. $MR_{cache} = 0.05$, $MR_M = 0.2$, thus:

- $AMAT = 1 + 0.05[10 + 0.2 \cdot 100] = 1 + 0.05[30] = 1 + 1.5 = 2.5$ cycles

$\boxed{\text{Answer: AMAT} = 2.5 \text{ cycles}}$

№ 4. Assuming a cache of 4096 blocks, a four-word block size, and a 64bit address, find the total number of sets and the total number of tag bits for caches that are direct-mapped, 2-way and 4-way set associative, and fully associative.

Solution:

I. 1-way set associative (direct-mapped) case:

Table:

	tag	DATA
0	#...	{...}
1	#...	{...}
⋮		⋮
4095	#...	{...}

set

set of sets

Let's calulate the size of tag bits section of 64-bit address for each block:

- As we have direct-mapped cache, then the total number of sets is equal to: $|\#SETS| = 4096$, as each set contains only 1 entry. ($|SETENT| = 1$), $|\#SETS| \cdot |SETENT| = |CACHE| = 4096$.
- Now, let's consider the address' bit collection:

$$i: \underbrace{b_2^{(63)} \cdots\cdots b_2^{(12)}}_{tagbits_i}\underbrace{b_2^{(11)} \cdots\cdots b_2^{(4)}}_{index_i}\underbrace{b_2^{(3)} b_2^{(2)} b_2^{(1)} b_2^{(0)}}_{offset_i}$$

→ * $b_2^{(k)}: k \in \mathbb{N}[0;63] \wedge b_2^{(k)} \in \{0,1\} \forall k$.

* 1) $|offset_i|$: we know that every word is 4-bit long & every block consists of 4 words, hence, each block has $4 \times 4 = 16 = 2^{4}$ bits. That is, in the address each block has 4-bits representative offset in address.

2) $|index_i| = \log_2(|\#SETS|)$, since each blocking set unit has to have its unique index, which is in range: $idx_2 \in [0; |\#SETS|-1]_2$, thus, in our case: $|index_i| = \log_2(4096) = \log_2(2^{12}) = 12$. Hence, in the address we have 12-bits representative for each set.

3) Now, calculating $|tagbits_i|$ is relatively easy: for each $i \in [0; |\#SETS|-1]$ $|tagbits_i| = 64 - |offset_i| - |index_i|$ That is: $|tagbits_i| = 64 - (12 + 4) = 48$.

4) Now, we can calculate the total number of tagbits in the given cache is just:

$|TAG_{CACHE}| = 48 \cdot |\#SETS| \cdot |SETENT| = 48 \cdot 4096 = 196608$.

$\boxed{\text{Answer: for direct-mapped: } |\#SETS| = 4096, |TAG_{CACHE}| = 196608.}$

II. 2-way set associative:

- $|\#SETS| = 2048$
- $|SETENT| = 2$.

blocks

	tag	DATA	tag	DATA
0	#...	{...}	#...	{...}
⋮	#...	{...}	#...	{...}
2047	#...	{...}	#...	{...}

1) $|offset_i| = \log_2(2^{2+2}) = 4$.

2) $|index_i| = \log_2|\#SETS| = \log_2(2^{11}) = 11$, thus:

3) $|tagbits_i| = 64 - 11 - 4 = 49$. 4) $|TAG_{CACHE}| = 49 \cdot 2048 \cdot 2 =$

$= 100352 \cdot 2 = 200704.$

Answer: for two-way set associative cache: $|\#SETS| = 2048, |TAG_{CACHE}| = 200704.$

III. 4-way set associative:

	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0	#...	{...}	...	...	...	...	...	...	...	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1023	#...	{...}	...	...	...	...	...	...	...	...

- $|\#SETS| = 1024.$
- $|SETENT| = 4.$

1) $|offset_i| = 4.$
2) $|index_i| = \log_2(2^{10}) = 10$, thus:
3) $|tagbits_i| = 64 - 4 - 10 = 50$, that is:
4) $|TAG_{CACHE}| = 50 \cdot |CACHE| = 204800.$

Answer: $|\#SETS| = 1024, |TAG_{CACHE}| = 204800.$

IV. 4096-way set associative.

	TAG	DATA	TAG	DATA	....	TAG	DATA
0	#...	{...}	#...	{...}	....	#...	{...}

1st block — 2nd block — 4096th block

- $|\#SETS| = 1.$
- $|SETENT| = 4096.$

1) $|offset_i| = 4$
2) $|index_i| = \log_2(2^1) = 1.$
3) $|tagbits_i| = 64 - 5 = 59$, thus: $|TAG_{CACHE}| = 59 \cdot 4096 = 241664.$

Answer: $|\#SETS| = 1, |TAG_{CACHE}| = 241664.$