CS1216 - Monsoon 2022 - Homework 6

Jivansh Sharma
UG 24 1020211193

29/11/2022

Collaborators: None

3. Assume a 512 KB, 4-way set associative cache with a 64 byte block size (cache line size). How many sets does the cache have? How many bits are used for the offset, index, and tag, assuming that the CPU provides 32-bit addresses? How large is the tag array? Show your work.

Now, we have the following data to work with:

1. `Blocksize = 128 bytes`
2. `Cache size = 64 MB = 64 * 1024 * 1024 bytes = 67108864 bytes`
3. `Associativity = 32 Ways`
4. `Address size = 48 Bits`
5. `Let number of sets  = x`

Using the formula for Cache Size, we get:
`Cache Size` = `Number of Sets` x `Associativity` x `Block Size`

$$\begin{aligned} 67108864 &= x \times 32 \times 128 \\ x &= 67108864/(32 \times 128) \\ x &= 67108864/4096 \\ x &= 16384 \end{aligned} \tag{1}$$

Therefore the number of sets are `16,384`. Now, we have to find the number of bits used for the offset, index and tag.

$$\begin{aligned} \texttt{Offset} &= \log_2(128) \\ \texttt{Offset} &= 7Bits \\ \texttt{Index} &= \log_2(16384) \\ \texttt{Index} &= 14Bits \\ \texttt{Tag} &= 48 - 7 - 14 \\ \texttt{Tag} &= 27Bits \end{aligned} \tag{2}$$

Therefore, the number of bits used for the offset, index and tag are `7, 14, 27` respectively.
Calculating the size of the tag array:
`Tag Size` = `Number of Sets` x `Associativity` x `Tag Size`

$$\begin{aligned}
\texttt{Tag Size} &= 16384 \times 32 \times 27 \\
\texttt{Tag Size} &= 1,41,55,776\texttt{Bytes} \\
\texttt{Tag Size} &= 1,41,55,776/1024\texttt{KB} \\
\texttt{Tag Size} &= 13,824/1024\texttt{MB} \\
\texttt{Tag Size} &= 13.824\texttt{MB}
\end{aligned} \tag{3}$$

Therefore, the size of the tag array is `13.824 MB`.

4. A 64 MB L3 cache has a 128 byte line (block) size and is 32-way set-associative. How many sets does the cache have? How many bits are used for the offset, index, and tag, assuming that the CPU provides 48-bit addresses? How large is the tag array? Show your work.

Now, we have the following data to work with:

```
1. Blocksize = 128 bytes
2. Cache size = 64 MB = 64 * 1024 * 1024 bytes = 67108864 bytes
3. Associativity = 32 Ways
4. Address size = 48 Bits
5. Let number of sets  = x
```

Using the formula for Cache Size, we get:
`Cache Size` = `Number of Sets` x `Associativity` x `Block Size`

$$\begin{aligned} 67108864 &= x \times 32 \times 128 \\ x &= 67108864/(32 \times 128) \\ x &= 67108864/4096 \\ x &= 16384 \end{aligned} \tag{4}$$

Therefore the number of sets are `16,384`. Now, we have to find the number of bits used for the offset, index and tag.

$$\begin{aligned} \texttt{Offset} &= \log_2(128) \\ \texttt{Offset} &= 7Bits \\ \texttt{Index} &= \log_2(16384) \\ \texttt{Index} &= 14Bits \\ \texttt{Tag} &= 48 - 7 - 14 \\ \texttt{Tag} &= 27Bits \end{aligned} \tag{5}$$

Therefore, the number of bits used for the offset, index and tag are `7, 14, 27` respectively.
Calculating the size of the tag array:
`Tag Size` = `Number of Sets` x `Associativity` x `Tag Size`

$$\begin{aligned} \texttt{Tag Size} &= 16384 \times 32 \times 27 \\ \texttt{Tag Size} &= 1,41,55,776\texttt{Bytes} \\ \texttt{Tag Size} &= 1,41,55,776/1024\texttt{KB} \\ \texttt{Tag Size} &= 13,824/1024\texttt{MB} \\ \texttt{Tag Size} &= 13.824\texttt{MB} \end{aligned} \tag{6}$$

Therefore, the size of the tag array is `13.824 MB`.

5. For the following access pattern, (i) indicate if each access is a hit or miss. (ii) What is the hit rate? Assume that the cache has 2 sets and is 2-way set-associative. Assume that block A maps to set 0, B to set 1, C to set 0, D to set 1, E to set 0, F to set 1. Assume an LRU replacement policy.

Now, we have the following data to work with:

```
1. A, C, E Map to Set #0
2. B, D, F Map to Set #1
3. LRU Replacement Policy
4. 2 Way 2 Set Associative Cache
5. In each block, LRU is Rightmost and MRU is Leftmost
```

Serial Number	Address	Hit/Miss	Cache 0 State	Cache 1 State	Why?
1.	A	Miss	A		Compulsory Miss
2.	B	Miss	A	B	Compulsory Miss
3.	C	Miss	C, A	B	Compulsory Miss
4.	B	Hit	C, A	B	
5.	A	Hit	A, C	B	
6.	B	Hit	A, C	B	
7.	E	Miss	E, A	B	Compulsory Miss
8.	C	Miss	C, E	B	Conflict Miss
9.	A	Miss	A, C	B	Capacity Miss
10.	D	Miss	A, C	D, B	Compulsory Miss
11.	B	Hit	A, C	B, D	
12.	C	Hit	C, A	B, D	
13.	A	Hit	A, C	B, D	
14.	F	Miss	A, C	F, B	Compulsory Miss
15.	D	Miss	A, C	D, F	Capacity Miss
16.	B	Miss	A, C	B, D	Conflict Miss
17.	C	Hit	C, A	B, D	
18.	E	Miss	E, C	B, D	Conflict Miss
19.	A	Miss	A, E	B, D	Conflict Miss

Now, we know Hit Rate = (Total Hits/Total Requests) * 100
Using the formula for CPI, we get:
Hit Rate = (Total Hits/Total Requests) * 100

$$\texttt{Hit Rate} = 7/19 * 100 = 36.84\% \tag{7}$$

6. Consider a program that can execute with no stalls and a CPI of 1 if the underlying processor can somehow magically service every load instruction with a 1-cycle L1 cache hit. In practice, 10% of all load instructions suffer from an L1 cache miss. Every cache miss results in a 300-cycle stall while data is fetched from memory. What is the CPI for this program if 20% of the program's instructions are load instructions?

Now, we have the following data to work with:

```
1. Baseline CPI = 1
2. Miss Rate = 10\%
3. Miss Penalty = 300 cycles
4. Percentage of load instructions = 20\%
```

Using the formula for CPI, we get:
CPI = Baseline + (Miss Rate * Miss Penalty * `Percentage of load instructions`)

$$CPI = 1 + (0.1 * 300 * 0.2) = 1 + 6 = 7 \tag{8}$$

Therefore the CPI would be `7`.