भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Indian Institute Of Technology Hyderabad

CS2323

Computer Architecture

Submitted To:
Sparsh Mittal
Asst. Professor
Computer Science Department

Submitted By :
Vijay Tadikamalla
Cs17BTECH11040

This document is generated by LaTeX

1. Consider an L2 cache which gets 50000000 accesses per second. It is designed with a memory which has leakage power of 0.07W and dynamic energy of 0.8nJ/access. If an application executes for one second, how much percentage of total cache energy comes from leakage energy?

Solution:

Given:
L2 cache which gets 50000000 accesses per second
Leakage power = 0.07w
Dynamic energy = 0.8nJ/access
Time = 1sec

$$\begin{aligned} Total\ Dynamic\ energy &= Dynamic\ energy\ per\ access \times no.\ of\ access\ per\ second \times time \\ &= 0.8 \times 10^{-9} \times 50000000 \times 1 \\ &= 0.04J \end{aligned}$$

$$Total\ energy\ =\ Dynamic\ energy + leakage\ energy = 0.04 + 0.07 = 0.11J$$

$\therefore$ Percentage of total cache energy that comes from leakage energy

$$= \frac{0.07}{0.11} \times 100\% = 63.63\%$$

2. Find out the reach of this DTLB:

PageSize	Entries	Associativity
4KB	64	4-way
2MB	32	8-way
1GB	8	fully-associative

Solution:

TLB Coverage (or reach or mapping size)= Sum of memory mapped by all TLB entries

$$= \sum (Pagesize) \times (Entries)$$

$$= 64 \times 4KB + 32 \times 2MB + 8 \times 1GB$$

$$= 8GB\ 64MB\ 256KB$$

3. Assume that a processor uses 8-bit address space. Assume that the address pattern that accesses the cache is:

Sequence1: 0, 63, 1, 62, 2, 61, 3, 60, 4, 59, 5, 58, 6, 57, 7, 56, 8, 55, 9, 54, 10, 53, 11, 52

Assume two different caches use the following two different address subdivision methods (figure is not drawn to scale).

Address subdivision method used by cache 1 :

Tag	Set Index	Offset

Address subdivision method used by cache 2 :

Offset	Set Index	Tag

Both the caches are direct-mapped, with a block size of 4 and have 8 sets each. In other words, the architectures are identical, except that they use different subdivision methods.
(a) Compute the tag and set for each address for both subdivision methods (hint: you can write a small C program to do that). You need not show this in your submission. For each address, show whether it leads to a hit or a miss and finally, what is the hit ratio (hits/accesses) for each cache?
(b) Repeat (a) but with the following sequence:

Sequence2: 0, 64, 128, 192, 1, 65, 129, 193, 11, 75, 139, 203, 9, 137, 201, 73

Solution:
Given:
Sequence1: 0, 63, 1, 62, 2, 61, 3, 60, 4, 59, 5, 58, 6, 57, 7, 56, 8, 55, 9, 54, 10, 53, 11, 52
Sequence2: 0, 64, 128, 192, 1, 65, 129, 193, 11, 75, 139, 203, 9, 137, 201, 73
Hit Ratio(cache 1, sequence 1)=$\frac{18}{24}$=0.75

Numbers(Seq1)	Tag	Set-Index	Hit Or Miss(Cache1)
0	000	000	Miss
63	001	111	Miss
1	000	000	Hit
62	001	111	Hit
2	000	000	Hit
61	001	111	Hit
3	000	000	Hit
60	001	111	Hit
4	000	001	Miss
59	001	110	Miss
5	000	001	Hit
58	001	110	Hit
6	000	001	Hit
57	001	110	Hit
7	000	001	Hit
56	001	110	Hit
8	000	010	Miss
55	001	101	Miss
9	000	010	Hit
54	001	101	Hit
10	000	010	Hit
53	001	101	Hit
11	000	010	Hit
52	001	101	Hit

Number(seq1)	Tag	Set-Index	Hit Or Miss(cache2)
0	000	000	Miss
63	111	111	Miss
1	001	000	Miss
62	110	111	Miss
2	010	000	Miss
61	101	111	Miss
3	011	000	Miss
60	100	111	Miss
4	100	000	Miss
59	011	111	Miss
5	101	000	Miss
58	010	111	Miss
6	110	000	Miss
57	001	111	Miss
7	111	000	Miss
56	000	111	Miss
8	000	001	Miss
55	111	110	Miss
9	001	001	Miss
54	110	110	Miss
10	010	001	Miss
53	101	110	Miss
11	011	001	Miss
52	100	110	Miss

Hit Ratio(cache 2, sequence 1)=$\frac{0}{24}$=0

Number(seq2)	Tag	Set-Index	Hit Or Miss(cache1)
0	000	000	Miss
64	010	000	Miss
128	100	000	Miss
192	110	000	Miss
1	000	000	Miss
65	010	000	Miss
129	100	000	Miss
193	110	000	Miss
11	000	010	Miss
75	010	010	Miss
139	100	010	Miss
203	110	010	Miss
9	000	010	Miss
137	100	010	Miss
201	110	010	Miss
73	010	010	Miss

Hit Ratio(cache 1, sequence 2)=$\frac{0}{16}$=0

Number(seq2)	Tag	Set-Index	Hit Or Miss(cache2)
0	000	000	Miss
64	000	000	Hit
128	000	000	Hit
192	000	000	Hit
1	001	000	Miss
65	001	000	Hit
129	001	000	Hit
193	001	000	Hit
11	011	001	Miss
75	011	001	Hit
139	011	001	Hit
203	011	001	Hit
9	001	001	Miss
137	001	001	Hit
201	001	001	Hit
73	001	001	Hit

Hit Ratio(cache 2, sequence 2)=$\frac{12}{16}$=0.75

4. Consider two processors (P1 and P2) which run the same instruction set architecture (ISA). The frequency of P1 and P2 are 2.2GHz and 1.6GHz, respectively. In this ISA, there are four classes of instructions A, B, C, and D. The CPI of each of these classes are given in the following table.

	A	B	C	D
P1	1	2	3	4
P2	2	2	2	2

There is a program which has $10\hat{6}$ instructions divided into classes as follows: 20% class A, 25% class B, 45% class C, and 10% class D. Which processor is faster for this program?

Solution:
Given:
Frequency of P1 = 2.2GHZ
Frequency of P2 = 1.6GHZ

$$CPU\ time = \frac{\sum_{i=1}^{n} IC_i \times CPI_i}{Frequency}$$

$$CPU\ time\ for\ P1 = \frac{10^6 \times (0.2 \times 1 + 0.25 \times 2 + 0.45 \times 3 + 0.1 \times 4)}{2.2 \times 10^9} = 1.1136 \times 10^{-3} s$$

$$CPU\ time\ for\ P2 = \frac{10^6 \times (0.2 \times 2 + 0.25 \times 2 + 0.45 \times 2 + 0.1 \times 2)}{1.6 \times 10^9} = 1.25 \times 10^{-3} s$$

Since, CPU time of P1 $<$ CPU time of P2 $\implies$ Processor P1 is faster than processor P2

5. Assume that a system has 4 processors (P=4). Assume that directory- based coherence protocol is used. Show the state of (P+1) bit directory for a cache block after each of these operations to that block.
i. P0 has read miss

ii. P1 has write miss
iii. P3 has write miss
iv. P2 has read miss
v. P2 has write miss
vi. P0 has read miss

Solution: States of (P+1)bit directory for a cache block after each instruction are shown below:

Sr. No	P_0	P_1	P_2	P_3	Exclusive Bit
i	1	0	0	0	0
ii	0	1	0	0	1
iii	0	0	0	1	1
iv	0	0	1	1	0
v	0	0	1	0	1
vi	1	0	1	0	0

6. Two applications are running on a processor which has shared L2 cache.
For application1: L2 cache misses with 2 and 6 ways (of last level cache) are 1000 and 400, respectively.
For application2: L2 cache misses with 2 and 6 ways (of last level cache) are 2000 and 1800, respectively
Assume that in between 2 and 6 ways, number of misses scale linearly (i.e., use linear interpolation).
Assume the cache has 8 ways, then which application should get how many ways for improving performance. An application needs to get at least two ways.

Solution:
Let application 1 has p-way cache and application 2 has q-way cache
Given: p+q=8
$p >= 2$ and $q >= 2$
L2 misses for application 1 with x-way cache =1000 - $\frac{(1000-400)}{4}$×(p-2) = 1300 - 150p
Similarly L2 misses for application2 = 2100-50q
Total misses = 1300-150p+2100-50q = 3400-150p-50(8-p) = 3000 -100p
No.of misses will be minimum if p will be maximum.
$\therefore$ p=6
So, application 1 should have 6 way cache and application 2 should have 2 way cache.

7. Three applications P, Q, R have a transactions rate of 34 per minute, 58 per minute and 81 per minute respectively. They run one after another. If each of them make 500 transactions, find the correct average value of transactions per minute. Also write which mean would you use to get the average.

Solution:
Given:
Transaction rate of A = 34
Transaction rate of B =58
Transaction rate of C =81
We can use Harmonic mean to calculate the average transactions per minute because no. of transactions are same.

$$Average\ transactions\ per\ minute = \frac{3}{\frac{1}{34}+\frac{1}{58}+\frac{1}{81}} = 50.84$$

Proof:

$$Average\ transactions\ per\ minute = \frac{Total\ no.\ of\ transactions}{total\ time}$$

$$Total\ no.\ of\ transactions = 500 \times 3 = 1500$$

$$Total\ time = 500/34 + 500/58 + 500/81$$

$$\therefore Average\ transactions\ per\ minute = \frac{1500}{\frac{500}{34} + \frac{500}{58} + \frac{500}{81}} = \frac{3}{\frac{1}{34} + \frac{1}{58} + \frac{1}{81}} = H.M$$

Hence Proved.

8. An application spends 25% of time in initialization, 37% of time in vision-processing function and remaining time in signal-processing function. In system0, all the tasks are run on a single-core CPU. However, system1 has an signal-processing accelerator and a vision-processing accelerator which give a speedup of 10X and 6X, respectively over the single-core CPU execution. Find the speedup of system1 over system0 assuming that both the accelerators are used on system1.

Solution:
Given:

Process	Time %
Initialization	25
Vision Processing	37
Singal Processing	38

$$Amdahl's\ law: \quad Overall\ speedup = \frac{1}{(1-f) + \frac{f}{s}}$$

where F be the fraction where enhancement is applied and s is speed up factor.

$$Speedup\ of\ System1\ over\ System0 = \frac{1}{0.25 + 0.37/6 + 0.38/10} = 2.859$$

9. Consider a processor that runs at 3 GHz and 1 Volt. The processor is capable of executing safely at voltages between 0.8 V to 1.2 V. Voltage and frequency follow a linear relationship (i.e., if voltage doubles, frequency doubles as well). When running a given CPU-bound program, the processor consumes 110 W, of which 30 W is leakage. The program takes 30 seconds to execute. Compute the following values (and also show at what frequency/voltage they are obtained):
(i)The smallest time it takes to execute the program
(ii) The lowest power to execute the program
(iii) The lowest energy to execute the program

Solution:
Given:
f= 3GHz and voltage= 1V
Leakage power= 30W and dynamic power=80W

a) Time $\propto \frac{1}{frequency}$

So, Minimum time $\implies$ Maximum frequency
$\therefore$ Max frequency= 1.2f (Max voltage=1.2V and Voltage and frequency follow a linear relationship)
$\therefore$ Min Time=$\frac{T}{1.2}$= $\frac{30}{1.2}$= 25sec

b)Power= Dynamic Power+ Leakage power
Dynamic power $\propto$ activity$\times$capacitance$\times voltage^2\times$frequency
Leakage power $\propto$ voltage
For minimum power, voltage should be minimum i.e. voltage=0.8V
Dynamic power at v=0.8V is $80\times(0.8)^3$=40.96W
Leakage power at v=0.8V is $30\times(0.8)$= 24W
$\therefore$ Min Power= 40.96+24=64.96W

c)Energy= Power$\times$time=(Dynamic Power+ Leakage power)$\times$time
$\implies$ Energy $\propto voltage^2$
So, Energy is minimum when voltage is minimum(v=0.8V).
Energy = Power$\times$time= $64.96\times\frac{30}{0.8}$=2436J

10. A processor uses 48 bit virtual address. It has 2GB physical memory and the memory addresses are defined at the level of each byte. Page size is 2KB. A processor has only one level of TLB which has 32 entries. Find out the size of TLB (excluding valid bits, etc).

Solution:
Given:
A processor that uses 48 bit virtual address and has 2GB physical memory
Total no. of entries= 32
So, no.of address = 2^{31}
Size of one page entry is =48-11+31-11 = 57 bits
Size of TLB = No.of entries $\times$ size = 57×32 = 1824 bits = 228 bytes

11. Consider a TLB which has 4 ports. The processor has a frame size of 1KB. In a given cycle, the addresses coming to the four ports of TLB are: 0x4795BA21, 0x4795BB21, 0x5795BA21 and 0x4785BA21. Find out how many unique accesses can be sent to TLB if it uses intra-cycle compaction technique to save energy.

Solution: In intra-cycle compaction technique, we take the unique addresses from the cycle and send them to the TLB.

1KB=2^{10} so there is 10 bit offset and hence the last 10 bits will be for offset.So, the remaining bits should be same if they give the same access. So,if 32-10=22 bits are same for two accesses, then there will be only one access.

Now binary representation of A=1010

Binary representation of B=1011

So, the first 22 bits of 0x4795BA21 and 0x4795BB21 are same.

Hence there will be only one access for 0x4795BA21 and 0x4795BB21. So, total number of unique accesses sent to TLB are 3 as the other addresses don't have the first 22 bits common with other addresses.

12. We have a small 2-entry, 2-way cache and the block size is 1B. Consider an access stream with addresses A, B, C, D, A, B, C, D, A, B, C, D.
Show whether each of the access is a hit or a miss with (a) LRU (least recently used) replacement policy (1 mark) and (b) MRU (most recently used) replacement policy. (you don't need to show the state of the cache. Just show the hit/miss decision for each access and total number of misses).

Solution:

Access Input	Least Recently Used Policy	Most Recently Used Policy
A	Miss	Miss
B	Miss	Miss
C	Miss	Miss
D	Miss	Miss
A	Miss	Hit
B	Miss	Miss
C	Miss	Miss
D	Miss	Hit
A	Miss	Miss
B	Miss	Miss
C	Miss	Hit
D	Miss	Miss
Total Miss	12	9