CS18BTECH11039_CA_HW1

RAJ PATIL

August 20, 2019

This document is generated by LaTeX

Abstract

Contains the answers and comments to questions provided in homework 1 of CS2323.

Contents

1 Question 1

time of execution (toe) = 10^{-6} s
Total accesses = 10^6

1.1 L1 Cache

$$\begin{aligned}
hit\ rate &= 95\% \\
no.\ of\ hits\ for\ L1\ cache &= 0.95 * 10^6 \\
energy\ per\ hit &= 0.217 * 10^{-9} J/hit \\
dynamic\ energy\ used(DE) &= no.\ of\ hits * energy\ per\ hit = 2.0615 * 10^{-4} J \\
leakage &= 0.2W \\
static\ energy\ used(SE) &= leakage * toe = 2 * 10^{-7} J \\
total\ energy\ used(TE) &= DE + SE = 2.0635 * 10^{-4} J
\end{aligned}$$

$$\%\ \mathbf{DE} = \frac{DE * 100}{TE} = \mathbf{99.90}\%$$

1.2 L2 Cache

$$\begin{aligned}
hits\ for\ L2\ cache &= 0.05 * 10^6 \\
energy\ per\ hit &= 1.47 * 10^{-9} J/hit \\
dynamic\ energy\ used(DE) &= no.\ of\ hits * energy\ per\ hit = 7.35 * 10^{-5} J \\
leakage &= 6.9W \\
static\ energy\ used(SE) &= leakage * toe = 6.9 * 10^{-6} J \\
total\ energy\ used(TE) &= DE + SE = 8.04 * 10^{-5} J
\end{aligned}$$

$$\%\ \mathbf{DE} = \frac{DE * 100}{TE} = \mathbf{91.42}\%$$

1.3 Conclusion

L2 Cache has lower Dynamic Energy usage for a program: both absolutely and relatively
and the opposite holds for Static Energy.

2 Question 2

On increasing associativity but keeping the cache size same, we only decrease the number of conflict misses and the capacity and compulsory misses still occur. Also, the number of conflict misses for a well-designed cache program are less due to advantage taken from spatial locality of data which results in different set-index mappings, hence we experience only a marginal increase in the hit-rate upon increasing the associativity of the cache.

3 Question 3

I've coded the solution in an ipython notebook because of the ease of exporting/importing csv files and including the code in LaTeX

3.1 Explanation

As the architechture is byte addresable, we have 8 bits for the address out of which : the offset, set index and tag bits are as follows:-

$$\begin{aligned}
\#\ of\ offset\ bits\ &= \log_2(block\ size) = 2\\
\#\ of\ set\ index\ bits &= \log_2(\#\ of\ sets) = 3\\
\#\ of\ tag\ bits &= 8 - \#\ of\ offset\ bits\ - \#\ of\ set\ index\ bits\\
&= 3
\end{aligned}$$

given a byte address where the bits are addressed from left to right: as 0 1 2 3 4 5 6 7

Cache 1	**Cache 2**
tag bits : 0 1 2	tag bits : 5 6 7
set index bits : 3 4 5	set index bits : 2 3 4
offset bits : 6 7	offset bits : 0 1

Including the code for sequence 1: cache 1 only; For the rest will be including just the exported csv files

3.2 Sequence 1: Cache 1 (Q3 a)

```
import numpy as np
import pandas as pd
sequence1 = list(pd.read_csv(”sequence1.csv”))
sequence1_bin = []
sequence1_tag = []
sequence1_set_i = []
for i in sequence1:
    sequence1_bin.append(format(int(i),”08b”))
    sequence1_tag.append(sequence1_bin[-1][0:3])
    sequence1_set_i.append(sequence1_bin[-1][3:6])
sequence1 = np.asarray(sequence1).reshape((24,1))
sequence1_bin = np.asarray(sequence1_bin).reshape((24,1))
sequence1_tag = np.asarray(sequence1_tag).reshape((24,1))
sequence1_set_i = np.asarray(sequence1_set_i).reshape((24,1))
cache1 = [-1]*8
hit_miss_log = []
cache_state = []
hit_ctr=0
for i in range(0,24):
    set_i = int(sequence1_set_i[i][0],2)
    tag = int(sequence1_tag[i][0],2)
    if (cache1[set_i]==tag) :
        hit_miss_log.append('hit')
        hit_ctr+=1
    else :
        hit_miss_log.append('miss')
        cache1[set_i] = tag
    cache_state.append(np.asarray(cache1))
cache_state = np.asarray(cache_state)
hit_miss_log = np.asarray(hit_miss_log).reshape((24,1))
sequence1_df = np.concatenate((sequence1,sequence1_bin,sequence1_tag,sequence1_set_i,
                               hit_miss_log,cache_state),axis=1)
indexlist1 = [”decimal”,”binary”,”tag”,”set_index”,”hit/miss”]
indexlist2 = [(”set_”+str(i)) for i in range(0,8)]
indexes = indexlist1+indexlist2
sequence1_df = pd.DataFrame(sequence1_df,columns=indexes,index=range(1,25))
sequence1_df.to_csv(r”./CSVs/S1C1.csv”)
```

	decimal	binary	tag	set index	hit/miss	set 0	set 1	set 2	set 3	set 4	set 5	set 6	set 7
1	0	00000000	000	000	miss	0	-1	-1	-1	-1	-1	-1	-1
2	63	00111111	001	111	miss	0	-1	-1	-1	-1	-1	-1	1
3	1	00000001	000	000	hit	0	-1	-1	-1	-1	-1	-1	1
4	62	00111110	001	111	hit	0	-1	-1	-1	-1	-1	-1	1
5	2	00000010	000	000	hit	0	-1	-1	-1	-1	-1	-1	1
6	61	00111101	001	111	hit	0	-1	-1	-1	-1	-1	-1	1
7	3	00000011	000	000	hit	0	-1	-1	-1	-1	-1	-1	1
8	60	00111100	001	111	hit	0	-1	-1	-1	-1	-1	-1	1
9	4	00000100	000	001	miss	0	0	-1	-1	-1	-1	-1	1
10	59	00111011	001	110	miss	0	0	-1	-1	-1	-1	1	1
11	5	00000101	000	001	hit	0	0	-1	-1	-1	-1	1	1
12	58	00111010	001	110	hit	0	0	-1	-1	-1	-1	1	1
13	6	00000110	000	001	hit	0	0	-1	-1	-1	-1	1	1
14	57	00111001	001	110	hit	0	0	-1	-1	-1	-1	1	1
15	7	00000111	000	001	hit	0	0	-1	-1	-1	-1	1	1
16	56	00111000	001	110	hit	0	0	-1	-1	-1	-1	1	1
17	8	00001000	000	010	miss	0	0	0	-1	-1	-1	1	1
18	55	00110111	001	101	miss	0	0	0	-1	-1	1	1	1
19	9	00001001	000	010	hit	0	0	0	-1	-1	1	1	1
20	54	00110110	001	101	hit	0	0	0	-1	-1	1	1	1
21	10	00001010	000	010	hit	0	0	0	-1	-1	1	1	1
22	53	00110101	001	101	hit	0	0	0	-1	-1	1	1	1
23	11	00001011	000	010	hit	0	0	0	-1	-1	1	1	1
24	52	00110100	001	101	hit	0	0	0	-1	-1	1	1	1

$$\text{Hit\%} = 75\%$$

3.3 Sequence 1: Cache 2 (Q3 a)

	decimal	binary	tag	set index	hit/miss	set 0	set 1	set 2	set 3	set 4	set 5	set 6	set 7
1	0	00000000	000	000	miss	0	-1	-1	-1	-1	-1	-1	-1
2	63	00111111	111	111	miss	0	-1	-1	-1	-1	-1	-1	7
3	1	00000001	001	000	miss	1	-1	-1	-1	-1	-1	-1	7
4	62	00111110	110	111	miss	1	-1	-1	-1	-1	-1	-1	6
5	2	00000010	010	000	miss	2	-1	-1	-1	-1	-1	-1	6
6	61	00111101	101	111	miss	2	-1	-1	-1	-1	-1	-1	5
7	3	00000011	011	000	miss	3	-1	-1	-1	-1	-1	-1	5
8	60	00111100	100	111	miss	3	-1	-1	-1	-1	-1	-1	4
9	4	00000100	100	000	miss	4	-1	-1	-1	-1	-1	-1	4
10	59	00111011	011	111	miss	4	-1	-1	-1	-1	-1	-1	3
11	5	00000101	101	000	miss	5	-1	-1	-1	-1	-1	-1	3
12	58	00111010	010	111	miss	5	-1	-1	-1	-1	-1	-1	2
13	6	00000110	110	000	miss	6	-1	-1	-1	-1	-1	-1	2
14	57	00111001	001	111	miss	6	-1	-1	-1	-1	-1	-1	1
15	7	00000111	111	000	miss	7	-1	-1	-1	-1	-1	-1	1
16	56	00111000	000	111	miss	7	-1	-1	-1	-1	-1	-1	0
17	8	00001000	000	001	miss	7	0	-1	-1	-1	-1	-1	0
18	55	00110111	111	110	miss	7	0	-1	-1	-1	-1	7	0
19	9	00001001	001	001	miss	7	1	-1	-1	-1	-1	7	0
20	54	00110110	110	110	miss	7	1	-1	-1	-1	-1	6	0
21	10	00001010	010	001	miss	7	2	-1	-1	-1	-1	6	0
22	53	00110101	101	110	miss	7	2	-1	-1	-1	-1	5	0
23	11	00001011	011	001	miss	7	3	-1	-1	-1	-1	5	0
24	52	00110100	100	110	miss	7	3	-1	-1	-1	-1	4	0

$$\text{Hit\%} = 0\%$$

3.4 Sequence 2: Cache 1 (Q3 b)

	decimal	binary	tag	set index	hit/miss	set 0	set 1	set 2	set 3	set 4	set 5	set 6	set 7
1	0	00000000	000	000	miss	0	-1	-1	-1	-1	-1	-1	-1
2	64	01000000	010	000	miss	2	-1	-1	-1	-1	-1	-1	-1
3	128	10000000	100	000	miss	4	-1	-1	-1	-1	-1	-1	-1
4	192	11000000	110	000	miss	6	-1	-1	-1	-1	-1	-1	-1
5	1	00000001	000	000	miss	0	-1	-1	-1	-1	-1	-1	-1
6	65	01000001	010	000	miss	2	-1	-1	-1	-1	-1	-1	-1
7	129	10000001	100	000	miss	4	-1	-1	-1	-1	-1	-1	-1
8	193	11000001	110	000	miss	6	-1	-1	-1	-1	-1	-1	-1
9	11	00001011	000	010	miss	6	-1	0	-1	-1	-1	-1	-1
10	75	01001011	010	010	miss	6	-1	2	-1	-1	-1	-1	-1
11	139	10001011	100	010	miss	6	-1	4	-1	-1	-1	-1	-1
12	203	11001011	110	010	miss	6	-1	6	-1	-1	-1	-1	-1
13	9	00001001	000	010	miss	6	-1	0	-1	-1	-1	-1	-1
14	137	10001001	100	010	miss	6	-1	4	-1	-1	-1	-1	-1
15	201	11001001	110	010	miss	6	-1	6	-1	-1	-1	-1	-1
16	73	01001001	010	010	miss	6	-1	2	-1	-1	-1	-1	-1

Hit% = 0%

3.5 Sequence 2: Cache 2 (Q3 b)

	decimal	binary	tag	set index	hit/miss	set 0	set 1	set 2	set 3	set 4	set 5	set 6	set 7
1	0	00000000	000	000	miss	0	-1	-1	-1	-1	-1	-1	-1
2	64	01000000	000	000	hit	0	-1	-1	-1	-1	-1	-1	-1
3	128	10000000	000	000	hit	0	-1	-1	-1	-1	-1	-1	-1
4	192	11000000	000	000	hit	0	-1	-1	-1	-1	-1	-1	-1
5	1	00000001	001	000	miss	1	-1	-1	-1	-1	-1	-1	-1
6	65	01000001	001	000	hit	1	-1	-1	-1	-1	-1	-1	-1
7	129	10000001	001	000	hit	1	-1	-1	-1	-1	-1	-1	-1
8	193	11000001	001	000	hit	1	-1	-1	-1	-1	-1	-1	-1
9	11	00001011	011	001	miss	1	3	-1	-1	-1	-1	-1	-1
10	75	01001011	011	001	hit	1	3	-1	-1	-1	-1	-1	-1
11	139	10001011	011	001	hit	1	3	-1	-1	-1	-1	-1	-1
12	203	11001011	011	001	hit	1	3	-1	-1	-1	-1	-1	-1
13	9	00001001	001	001	miss	1	1	-1	-1	-1	-1	-1	-1
14	137	10001001	001	001	hit	1	1	-1	-1	-1	-1	-1	-1
15	201	11001001	001	001	hit	1	1	-1	-1	-1	-1	-1	-1
16	73	01001001	001	001	hit	1	1	-1	-1	-1	-1	-1	-1

Hit% = 75%

3.6 Tabulation Explanation

The first 6 columns have intuitive headings
The last 8 columns represent the current state of the corresponding cache after the present access request has been processed.
-1 indicates that that set hasn't been used yet and valid bit for it is still 0
the rest of the non-negative numbers are the tags stored by that set

4 Question 4

$$\begin{aligned} \#\ of\ A\ (\#A) &= 0.3 * 10^6 \\ \#\ of\ B\ (\#B) &= 0.2 * 10^6 \\ \#\ of\ C\ (\#C) &= 0.35 * 10^6 \\ \#\ of\ D\ (\#D) &= 0.15 * 10^6 \end{aligned}$$

4.1 Processor P1

$$\begin{aligned} frequency\ of\ P1 &= 2.2\ GHz \\ CPI\ for\ A &= 2 \\ CPI\ for\ B &= 2 \\ CPI\ for\ C &= 4 \\ CPI\ for\ D &= 4 \end{aligned}$$

$$\begin{aligned} Total\ Cycles &= 2 * (\#A) + 2 * (\#B) \\ &\quad + 4 * (\#C) + 4 * (\#D) \\ &= 3 * 10^6\ cycles \\ Execution\ Time &= \frac{3 * 10^6\ sec}{2.2 * 10^9} \\ &= 1.36 * 10^{-3}\ seconds \end{aligned}$$

4.2 Processor P2

$$\begin{aligned} frequency\ of\ P2 &= 1.6\ GHz \\ CPI\ for\ A &= 2 \\ CPI\ for\ B &= 1 \\ CPI\ for\ C &= 2 \\ CPI\ for\ D &= 3 \end{aligned}$$

$$\begin{aligned} Total\ Cycles &= 2 * (\#A) + 1 * (\#B) \\ &\quad + 2 * (\#C) + 3 * (\#D) \\ &= 1.95 * 10^6\ cycles \\ Execution\ Time &= \frac{1.95 * 10^6\ sec}{1.6 * 10^9} \\ &= 1.22 * 10^{-3}\ seconds \end{aligned}$$

4.3 Conclusion

Processor P2 is faster than Processor P1 for this program

5 Question 5

Directory based coherence

Sr no.	Event	P_0	P_1	P_2	P_3	Exclusive bit
0	initial state	0	0	0	0	0
1	P_1 :- read miss	0	1	0	0	0
2	P_2:- write miss	0	0	1	0	1
3	P_0 :- write miss	1	0	0	0	1
4	P_3 :- read miss	1	0	0	1	0
5	P_3 :- write miss	0	0	0	1	1
6	P_2 :- read miss	0	0	1	1	0

the exlusive bit is only updated when a write miss occurs (0 $\longrightarrow$ 1) when the exlusive bit is 0
or
a read miss occurs when the exlusive bit is already 1(1 $\longrightarrow$ 0).

6 Question 6

let application 1 be assigned x ways and application 2 has 8-x ways

6.1 Misses for app 1

let M_1 be the corresponding misses
Then, we have

$$\frac{4000 - M_1}{2 - x} = \frac{4000 - 3600}{2 - 6} \ simplifying\ we\ have\ M_1 = 4200 - 100x$$

6.2 Misses for app 2

let M_2 be the correspoding misses
Then, we have

$$\frac{2040 - M_2}{2 - (8 - x)} = \frac{2040 - 1600}{2 - 6} \ simplifying\ we\ have\ M_2 = 1380 + 110x$$

6.3 Total misses

$$Total\ misses = M_1 + M_2 = 5580 + 10x$$

Hence min misses occur at min value of x = 2
Hence **application 1 gets 2 ways** & **application 2 gets 6 ways** for minimising the total number of misses

7 Question 7

7.1 A

Transaction Rates in tpm (transactions per minute)

$$P = 44\ tpm$$
$$Q = 77\ tpm$$
$$R = 91\ tpm$$

They run after one another and each of them (P,Q and R) have the same number of transactions :- 600
Hence, as the numerator of the quantity being calculated is same(average rate) for the separate instances, we will use Harmonic Mean

$$Average\ Rate\ = \frac{3\ tpm}{\frac{1}{44} + \frac{1}{77} + \frac{1}{91}} = 64.24\ tpm$$

verifying using the basic method

$$total\ transactions = 1800$$
$$total\ time = (\frac{600}{44} + \frac{600}{77} + \frac{600}{91})\ minutes$$
$$Average\ Rate\ = \frac{1800}{600 * (\frac{1}{77} + \frac{1}{91} + \frac{1}{44})}\ tpm = 64.24\ tpm$$

Hence, Harmonic mean is the correct kind of mean for this entity

7.2 B

Quantity being considered :- Cycles per Instructions (CPI)

Weighted AM

$$\begin{aligned} weights &= \frac{70}{240}, \frac{80}{240}, \frac{90}{240} \\ mean &= \frac{45 * 70}{70 * 240} + \frac{35 * 80}{80 * 240} + \frac{40 * 90}{90 * 240} \\ &= \frac{120}{240} = 0.5\ CPI \end{aligned}$$

Weighted HM

$$\begin{aligned} weights &= \frac{45}{120}, \frac{35}{120}, \frac{40}{120} \\ mean &= \frac{1}{\frac{45*70}{45*120} + \frac{35*80}{35*120} + \frac{40*90}{40*120}} \\ &= \frac{120}{240} = 0.5\ CPI \end{aligned}$$

8 Question 8

Assume system 0 takes 100 units of total time(100 u)

8.1 System 0

$$\begin{aligned} initialization &= 29\ u \\ vision\ processing &= 39\ u \\ signal\ processing\ function &= 32\ u \\ total &= 100\ u \end{aligned}$$

8.2 System 1

$$\begin{aligned} initialization &= 29\ u \\ vision\ processing &= \frac{39}{7}\ u = 5.57\ u \\ signal\ processing\ function &= \frac{32}{12}\ u = 2.66\ u \\ total &= 37.23\ u \end{aligned}$$

$$speedup\ of\ System\ 1\ w.r.t\ System\ 0 = \frac{100}{37.23} = 2.685$$

9 Question 9

$$\begin{aligned} base\ Voltage\ (BV) &= 1V \\ base\ Frequency\ (BF) &= 3GHZ \\ 0.8\ V \leq allowable\ Voltage\ (AV) &= k * F \leq 1.2\ V \\ Static\ Power\ (SP) &= k_1 * I * AV \\ &= 40\ W(base) \\ Dynamic\ Power\ (DP) &= k_2 * C * AV^2 * F \\ &= 110\ W(base) \\ Execution\ Time(ET) &= 40\ secs \\ Total\ Cycles(TC) &= ET * BF = 120 * 10^9\ cycles \end{aligned}$$

9.1 i)

For smallest time, maximising Frequency which occurs at max Voltage = 1.2 V,
Hence new Frequency = 1.2*3 GHZ = 3.6GHZ
Hence, minimum

$$ET = TC/F = \frac{120 * 10^9}{3.6 * 10^9}\ seconds = 33.33\ seconds$$

Occurs at V = 1.2 V and F = 3.6 GHZ

9.2 ii)

Lowest power occurs at least allowable Voltage = 0.8V,
Hence F = 0.8*3GHz = 2.4GHZ

$$SP = 0.8 * SP_{base} = 0.8 * 40\ W = 32\ W$$

$$DP = 0.8^3 * DP_{base} = 0.8^3 * 110\ W = 56.32\ W$$

$$Total\ Power = DP\ +\ SP = 88.32\ W$$

occurs at V = 0.8 V and F = 2.4 GHZ

9.3 iii)

for minimum Energy Consumption, the ET also matters hence the Frequency also affects the expression

Let the new V = K*BV = K*1V = KV where K ranges from 0.8 to 1.2 Hence, the new frequency = K*BF = K*3GHZ = 3K GHz

$$SP = SP_{base} * K$$

$$DP = DP_{base} * K^3$$

$$ET = \frac{TC}{F} = \frac{ET_{base}}{K}$$

$$Energy\ Consumption = (SP + DP) * ET = (40 * K + 110 * K^3) * (\frac{40}{K})\ J = (40 + 110 * K^2) * (40)\ J$$

this is a basic Quadratic dependence with its minimum occuring for K=0.8 in our allowable domain of K

$$\therefore Energy_{min} = (40 + 110 * 0.8^2) * 40\ J = \mathbf{4416\ J}$$

occurs at V = 0.8 V and F = 2.4 GHZ

10 Question 10

10.1 (A)

a) Replacement Policy:- LRU

for an LRU(Least Recently Used) replacement Policy

The cache performs poorly and the first 4 will be compulsory misses whereas the later ones would be capacity(no conflict misses here as we are dealing with a fully associative cache architecture) misses, Hence we will have 12 misses
Request access :- PQRSPQRSPQRS
Hence, Miss/Hit Log (M:- Miss || H:- Hit) (Left to right) :-

$$M \mid M \mid M \mid M \mid M \mid M \mid M \mid M \mid M \mid M \mid M \mid M$$

$$\frac{Misses}{Total\ accesses} = \frac{12}{12}$$

b) Replacement Policy:-MRU

Here the most recently used tag will be chosen for eviction when we need a replacement

The cache performs better in this case with an MRU with the first 4 being compulsory misses followed by 2 capacity misses later. The total number of misses will be 6
Request access :- PQRSPQRSPQRS
The Miss/Hit Log (M:- Miss || H:- Hit) (Left to right) :-

$$M \mid M \mid M \mid M \mid H \mid H \mid M \mid H \mid H \mid M \mid H \mid H$$

$$\frac{Misses}{Total\ accesses} = \frac{6}{12}$$

10.2 (B)

			Cache State				
Sr no.	**Request access**	**Hit/Miss**	**way 1**	**way 2**	**way 3**	**Current LRU Block**	**Victim Cache**
1	P	Miss	P	-	-	P	-
2	Q	Miss	P	Q	-	P	-
3	R	Miss	P	Q	R	P	-
4	S	Miss	S	Q	R	Q	P
5	P	Hit	S	P	R	R	Q
6	Q	Hit	S	P	Q	S	R
7	R	Hit	R	P	Q	P	S
8	S	Hit	R	S	Q	Q	P
9	P	Hit	R	S	P	R	Q
10	Q	Hit	Q	S	P	S	R
11	R	Hit	Q	R	P	P	S
12	S	Hit	Q	R	S	Q	P

following a least recently used policy here; victim cache improves the performance of the cache here mainly because the total number of blocks accessed by the programs are $\leq$ cache size hence there won't be any misses after the first 4 compulsory misses.

To get a better idea of the performance of the victim cache, one should run an application with the number of different blocks it requires being greater than the total cache size($L_1 + victim$).

11 Question 11

11.1 Given info

$$\begin{aligned}
Processor\ Frequency &= 2\ GHz \\
Base\ CPI &= 3 \\
L_1\ I - Cache\ miss\ rate &= 1\% \\
L_1\ D - Cache\ miss\ rate &= 3\% \\
L_2\ I - Cache\ miss\ rate(local) &= 0\% \\
L_2\ D - Cache\ miss\ rate(local) &= 4\% \\
main\ memory\ latency &= \frac{60ns}{\frac{1}{2GHz}} = 120\ cycles \\
Unified\ L_2\ latency &= \frac{6ns}{\frac{1}{2GHz}} = 12\ cycles \\
\%\ Loads/Stores &= 30\ \%
\end{aligned}$$

11.2 Case 1

$$\begin{aligned}
D - cache\ miss\ overhead &= 0.3 * 120 * 0.03 = 1.08 \\
I - cache\ miss\ overhead &= 120 * 0.01 = 1.2 \\
effective\ CPI &= 3 + 1.08 + 1.2 = \mathbf{5.28\ cycles}
\end{aligned}$$

11.3 Case 2

$$\begin{aligned}
D - cache\ miss\ overhead &= 0.3 * (0.03 * 12 + 0.03 * 0.04 * 120) = 0.1512 \\
I - cache\ miss\ overhead &= 12 * 0.01 = 0.12 \\
effective\ CPI &= 3 + 0.1512 + 0.12 = \mathbf{3.2712\ cycles}
\end{aligned}$$