$\frac{\;}{16}$ $B = \frac{64000}{8} =$

۱) الف)

$N = 2$, $b = 4\ \text{word} \rightarrow 16\ \text{byte}$, $C = 64\ \text{KByte}$

$B = \frac{C}{b} = \frac{64000}{16} = \underline{4000}$, $S = \frac{B}{N} = \frac{4000}{2} = \underline{2000}$

$\log_2 2000 = 11 \longrightarrow$ 11 bits for set bits.

$2^4 = 16 \longrightarrow$ 4 bits for offset

4 بیت برای offset خواهیم داشت که 4 بیت کم ارزش هستند و 11 بیت برای set bits خواهیم داشت و 17 بیت باقی برای Tag است.

17	11	4
Tag	set	offset

0x03	0xb4	0x2b	0x02
00000011 ,	10110100 ,	00101011 ,	00000010
0xbf	0x58	0xbf	0x0e
10111111 ,	01011000 ,	10111110 ,	00001110
0xb5	0x2c	0xba	0xfd
10110101 ,	01011000 ,	10111010 ,	11111101

b4 03
2b

ب)

1	00...000	00...0000	0011	→ miss
2	00...000	00...1011	0100	→ miss
3	00...000	00...0010	0011	→ miss
4	00...000	00...0000	0010	→ hit
5	00...000	00...1011	1111	→ hit
6	00...000	00...0101	1100	→ miss
7	00...000	00...1011	1110	→ hit
8	00...000	00...0000	1110	→ hit
9	00...000	00...1011	0101	→ hit
10	00...000	00...0101	1000	→ hit
11	00...000	00...1011	1010	→ hit
12	00...000	00...1111	1101	→ miss

ج)

$$\text{Miss Rat} = \frac{۵}{۱۲} \times ۱۰۰ \simeq \boxed{۴۱,۶\%}$$

$$\text{Hit Rate} = \frac{۷}{۱۲} \times ۱۰۰ \simeq \boxed{۵۸,۴\%}$$

د) اگر N دو برابر شود آن گاه در هر set ۴ block خواهیم داشت و تعداد کل set ها برابر می شود با ۲.

$N=4$, $b=16^{(B)}$ و $C=64^{(KB)}$ و $B=4000$ و $S=1000$

حال اگر blocksize را دو برابر کنیم آن گاه خواهیم داشت

$N=2$, $b=32^{(B)}$ و $C=64^{(KB)}$ و $B=2000$ و $S=1000$

می دانیم که هر چه تعداد way ها بیشتر باشد conflict کمتری رخ می دهد پس بهتر است که تعداد N را دو برابر کنیم تا یک N-way associative داشته باشیم

و از آن جایی که در هر دو حالت مجموعاً دو set داریم منطقی است که حالتی را انتخاب کنیم که تعداد block های بیشتری داشته باشد.

از طرفی هم می دانیم که نرخ Hit با N رابطه مستقیم دارد و با مجذور B رابطه مستقیم دارد و طبعا داریم $2>\sqrt{2}$ پس N را دو برابر کنیم بهتر است.

ولی در این سوال و با این آدرس ها تغییری در نرخ Hit و یا نرخ Miss ایجاد نمی شود.

2) الف)

$C = 64KB$, $b = 32B \rightarrow B = \frac{C}{b} = \frac{64 \times 10^{10}}{2^5} =$

$= \frac{2^{11}}{2^5} = 2^{11}$! $\longrightarrow$ تعداد کل block ها

چون در حالت اول از Dirct Mapped استفاده می کنیم پس تعداد set ها نیز برابر با

2^{11} است و می دانیم که b = 8 word پس در هر مرحله 8 word از معوری وارد کش می شود

در کد از ما خواسته شده تا 10000 word که پشت هم هستن را در کش مقدار دهی کنیم

حال هنگام مقدار دهی word اول به miss می خوریم و یک word از کش پر می شود و بعد

7 word باقی مانده در کش قرار داده می شود پس تا 7 ارجاع بعد Hit می خوریم حال

8 word سمت اول (set 0) پر شده است به set بعدی (set 1) می رویم در این جا

دوباره برای مقدار دهی word اول یک miss می خوریم و 7 word بعدی در کش ذخیره

می شوند و 7 بار Hit می خوریم و به همین صورت تا آخر. حال مشکل

capacity miss را باید بررسی کنیم:

از آن جایی که 2^{11} set داریم که در هر set یک block با ظرفیت $8 = 2^3$ وجود

دارد پس در کل 2^{14} word در کش داریم که می تواند data در خود ذخیره کند و طبق

کد داده شده به ما، ما در کل به 10000 word برای ذخیره data در کش نیاز داریم

پس به مشکل capacity miss برنمی خوریم و به خاطر منحصر به فرد بودن حافظه ها هم

به conflict miss برنمی خوریم.

حال به بررسی حالت دوم می پردازیم که 2-way Associative است .

در این حالت هم مثل DM است با این تفاوت که چون N=2 است پس

در هر set ما دو block داریم که یعنی تعداد set ها نصف شده است

و در way0 و way1 دارد که اول way0 کامل پر می شود و بعد way1 پر می شود.

در fully Associative نیز ما یک set داریم با 2^{11} block با سایز $word 8 = 2^3$

پس یک set داریم با 2^{14} word و ما به 10000 word آن نیاز داریم پس هیچ

conflict و capacity miss نخواهیم داشت .

ب) الگوی Hit و miss مثل قسمت الف است که توضیح داده شد با این

تفاوت که ظرفیت از 64KB به 32KB تغییر کرده است پس خواهیم داشت

$$B = \frac{32 \times 10^{10}}{32} = 2^{10} \longrightarrow \text{تعداد کل block ها}$$

پس 2^{10} set وجود دارد پس در هر set یک block با سایز $word 8 = 2^3$ داریم که پس

در کل کش با 2^{13} word داریم که طبق کد داده شده به 10000 word آن نیاز داریم

پس در ابتداء اول word miss می خورد و ذخیره می شود و 7 word بعدی وارد کش

می شود پس تا 7 hit خواهیم داشت و به همین صورت آخر ولی در جایی از کش

باید به set0 برگردیم و خالی شود تا دوباره 8 word بعدی در آن ذخیره شود پس

conflict miss و capacity miss خواهیم داشت و به همین صورت برای set1

و الی آخر.