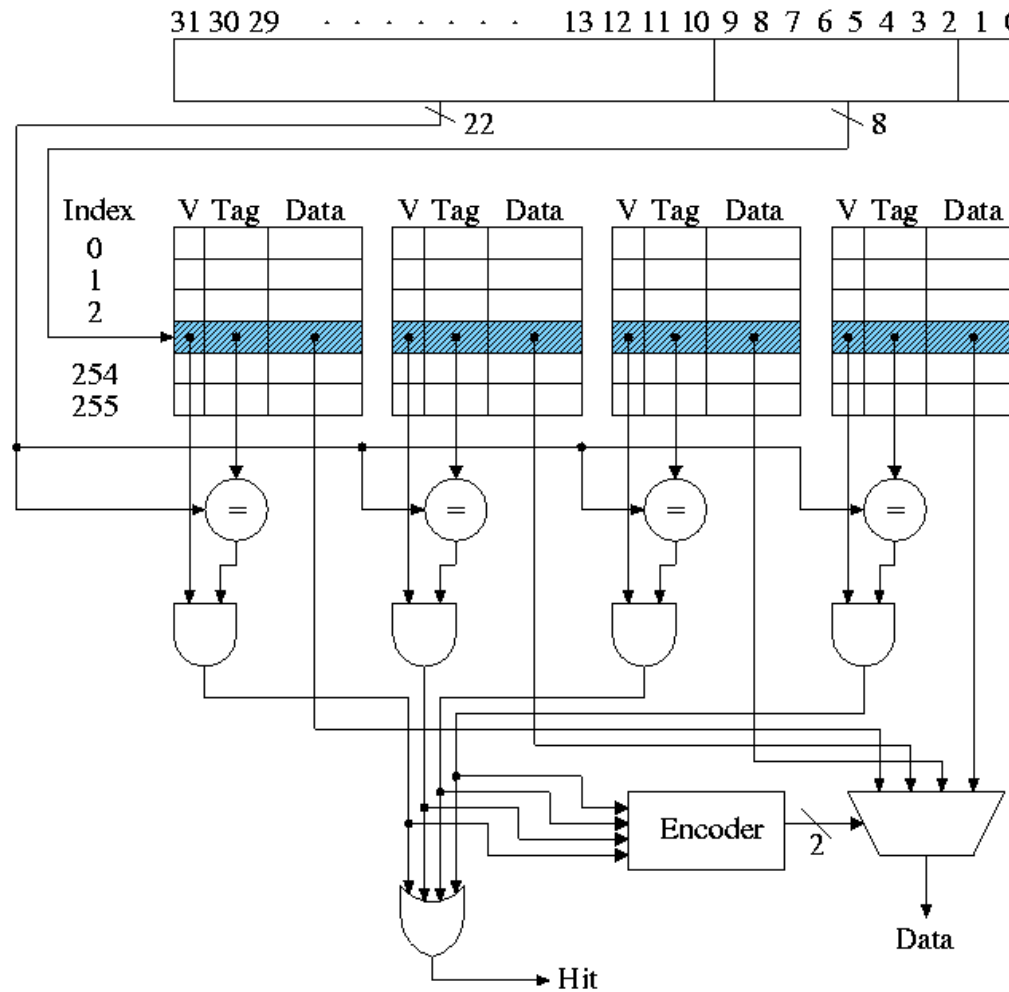
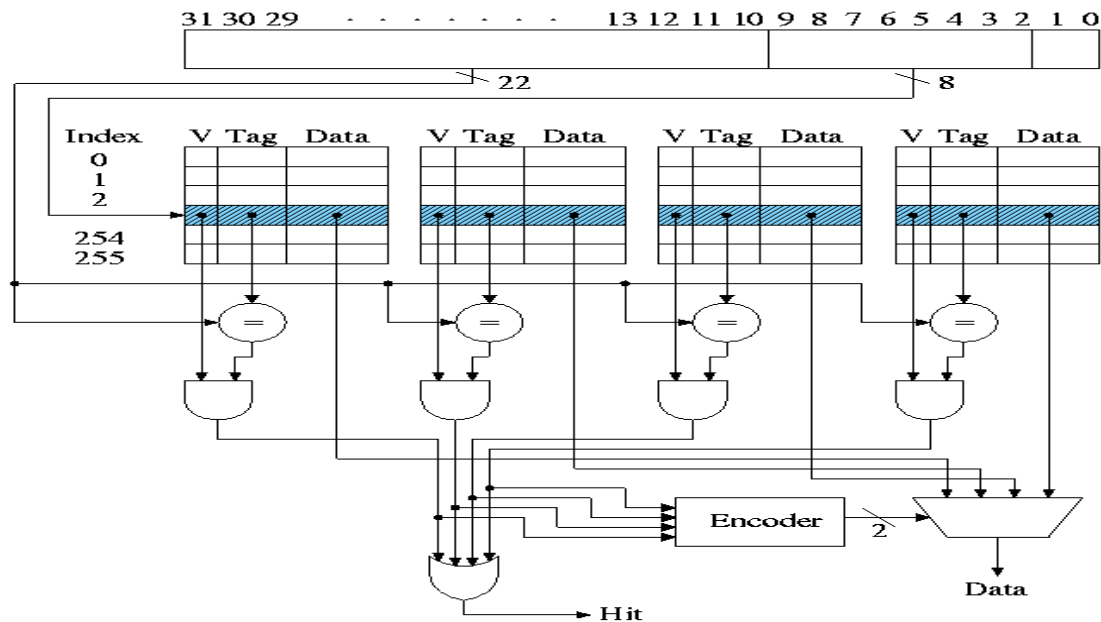


שאלות זיכרון מטמון Cache Memory • ארגון המחשב

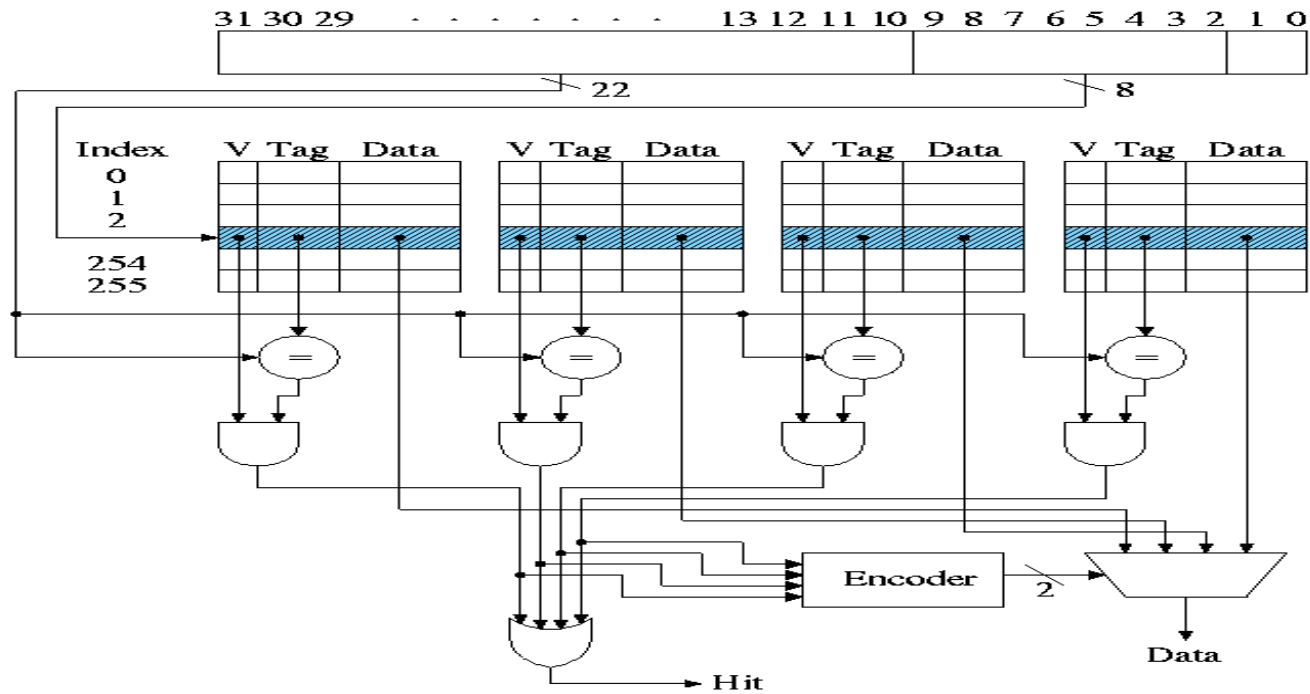
מרצה: ד"ר רם בוסני

שאלה 1: שאלת מבנה זיכרון מטמון



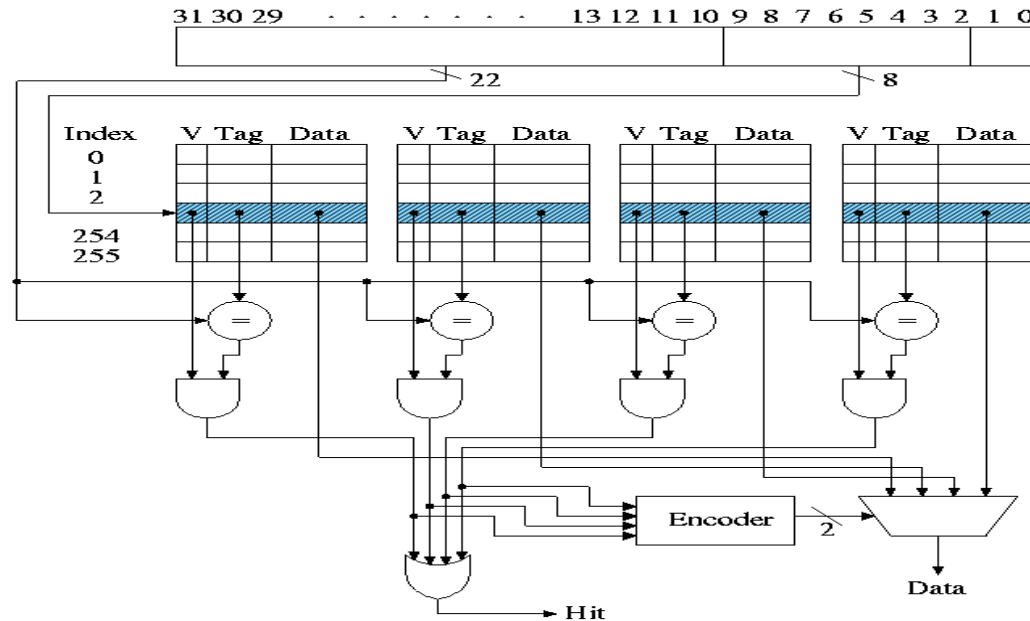


א. מהו סוג זיכרון זה ?
 זהו זיכרון 4-way set associative cache. גודל בלוק הינו מילה (המינימום).
 יש לנו אינדקס של 8 סיביות (256 שורות בזיכרון) ו tag של 22 סיביות (ע"מ
 להשלים לכתובת של 32 סיביות).



ב. מה כמות הנתונים שניתן לאפסן בזיכרון זה?

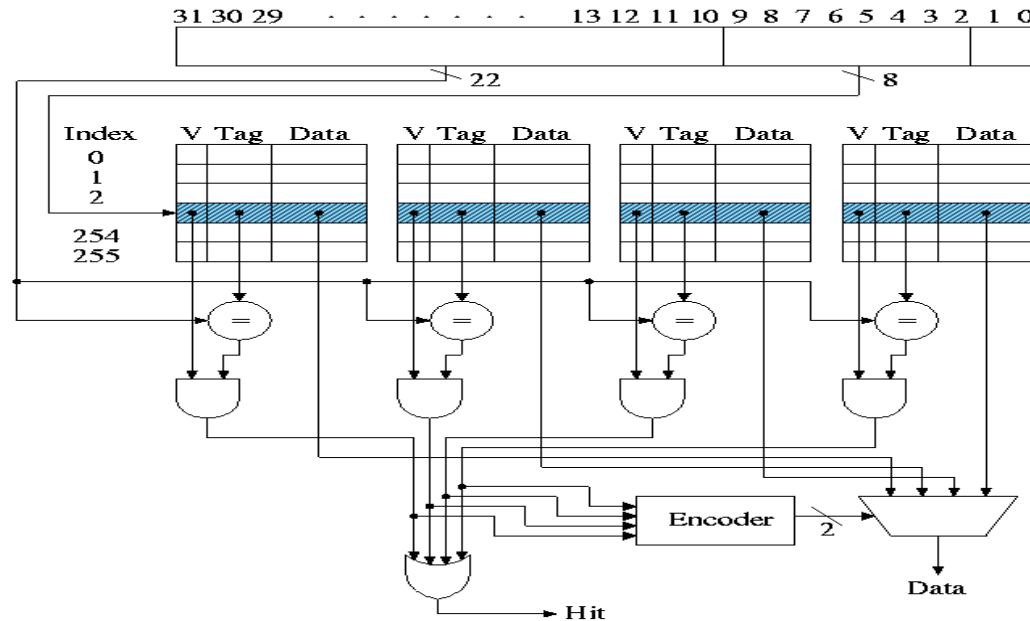
כמות הנתונים הינה
 $32,768[\text{bit}] = 4\text{KByte} = 32[\text{bit}] * 4(\text{set}) * 2^8(\text{שורות})$



ג. מה המספר הכולל של סיביות בזיכרון זה (כולל tag ו valid bit)?

המספר הכולל של סיביות הינו:

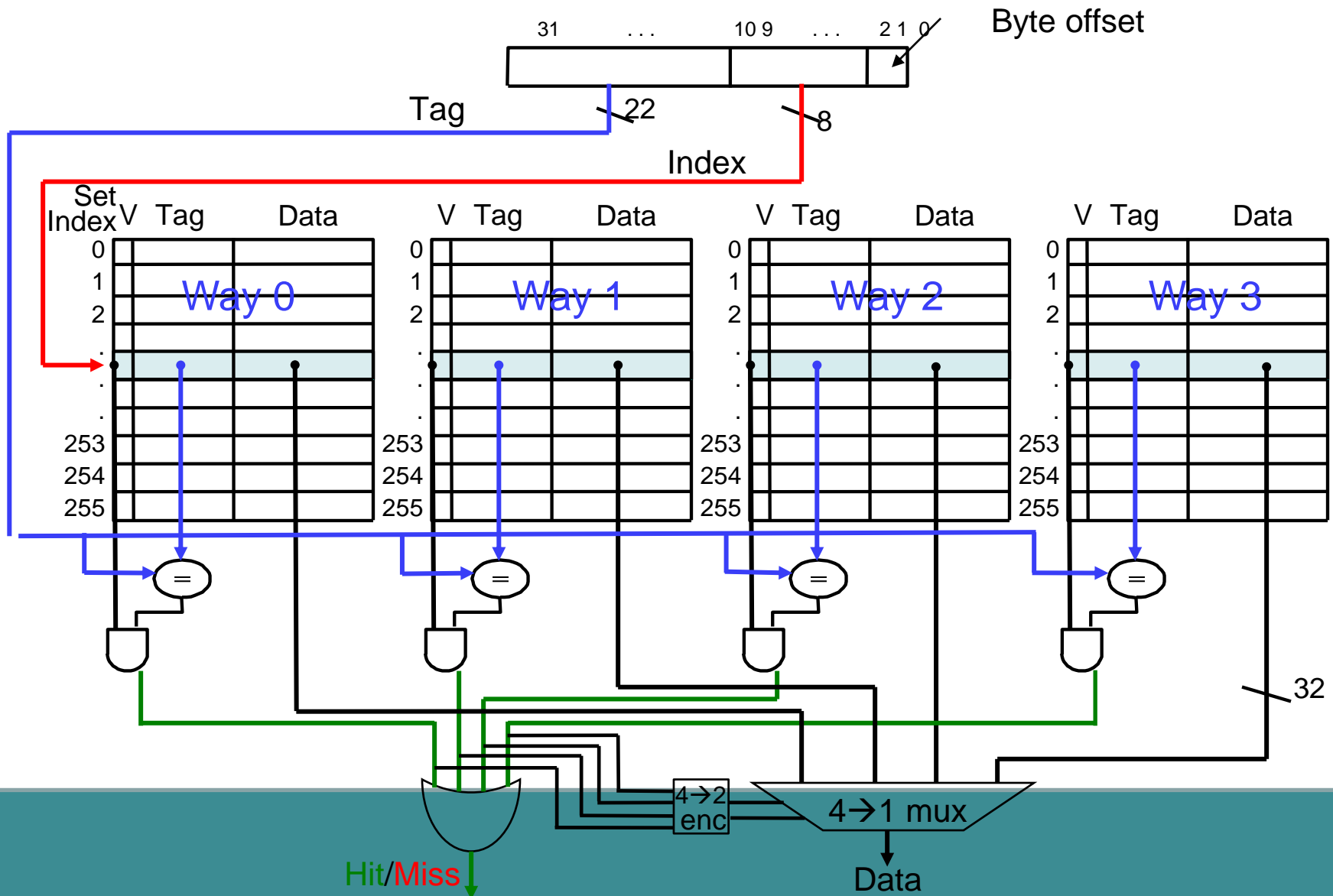
$$2^8 \text{ (שורות)} * 4 \text{ (set)} * (32 + 22 + 1) [\text{bit}] = 56,320 [\text{bit}]$$

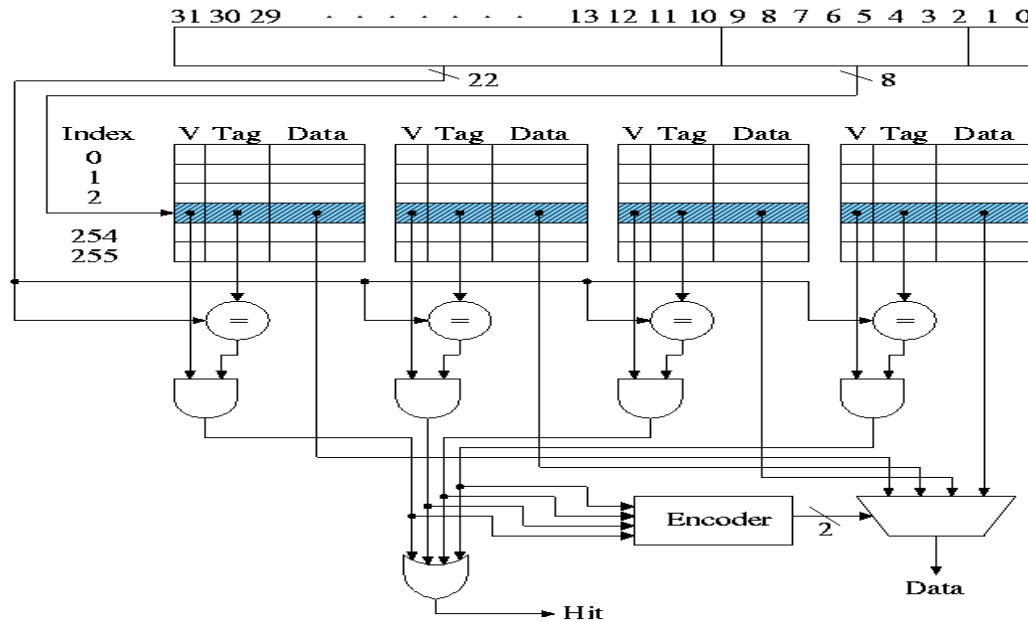


ד. יש לתאר את תהליך החיפוש בזיכרון זה ?
 כאשר מתקבלת כתובת אז: על סמך 8 סיביות האינדקס ניגשים לשורה המתאימה (מתוך 256)
 ומתבצעות 4 השוואות במקביל של שדה ה tag וסיבית ה valid נבדקת במקביל במידה
 ומתקיימים שני התנאים שער ה AND מוציא 1 וזה מועבר הלאה לשער ה or המוציא 1 ובכך
 מכריז על hit . במקביל הסיבית של האינדיקציה ל hit נכנסת גם למקודד (4 ל 2) המוציא כפלט
 את המיקום הפנימי בתוך ה set מתוך ה 4 בזיכרון בו התקיים hit. ובכך המרבב של 4 ל 1 בורר
 את הנתונים המתאימים ל hit .

Accessing Data in a Set Associative Cache

4 ways, 256 sets, 1 word/block •





ה. האם יכול להיווצר מצב בו שתי כניסות ל Encoder מקבלות 1 ?

אם כן מתי זה קורה ?

לא יכול להיות מצב של שני hit לאותה כתובת כלומר באותו ה set משום שבבניה נכונה של הזיכרון יש עדכון של tag אחד מתוך ה 4 ב set. כך שבניהול נכון של 4-way set associative cache המצב לא אמור לקרות.

הערה: אמנם ניתן להסתכל על 4-way set associative cache כמימוש של Direct Mapped 4 Cache במקביל אבל המימוש מסונכרן ויש קשר בתוך ה | set והבדיקות מבוצעות במקביל על כל ה set.

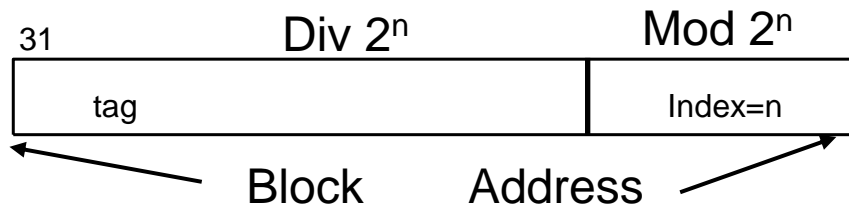
תהליך המיפוי מכתובת בזיכרון הראשי למציאת המיקום במטמון ע"פ נתוניו (m, n)

ניתן את המתכון לאופן הכללי מבחינת הסתכלות על השדות:

Tag, Index, Block offset, Byte offset

- קודם כל שימו לב שיש שאלות בתרגילים או במבחנים המתייחסות לכתובת בבתים במילים בהתאמה במידה והכתובת ניתנת כבר במילים פשוט נתעלם מה byte offset

דרך ראשונה להצגה היא ע"י רישום בבינארי של הכתובת הממופה מהזיכרון הראשי ובכך להגיע לחלוקה המתאימה לשדות. שלב ראשון:



דרך שניה:

"להעלים" את ההיסטים כלומר לחלק בגודל הבלוק מבלי להתייחס לשארית (הנמצאת בהיסטים) - החלוקה במילים או בבתים (כלומר 2^m או 2^{m+2}) לפי נתוני השאלה נותנת לנו מספר הנקרא Block Address (הערה לפעמים בנתוני שאלה של גודל בלוק שווה מילה ובכתובות המופיעות במילים למעשה נבצע חילוק ב 1 כלומר מראש התעלמנו מההיסטים וניתן לדלג על שלב זה)

- שלב שני:** כעת למעשה נשארו עם מספר הבלוק בזיכרון הראשי (Block address) כלומר עם סיביות האינדקס והTAG במיפוי.
- שלב שלישי:** ההפרדה בין האינדקס ל tag הינה באופן הבא נחלק את Block address במספר השורות \בלוקים (2^n) מטמון השארית תיתן את האינדקס והתוצאה תיתן את tag

שאלה 2

נתון מחשב עם זיכרון מטמון בו בוצעו גישות לזיכרון לרצף הכתובות הבאות, הסדר משמאל לימין : **7,13,37,27,33,11,7,18,34,23**. הניחו כי כתובות אלו מתייחסות לבתים והן נתונות בבסיס 10, כמו כן נתון שזיכרון המטמון ריק בהתחלה. במחשב זה נבחנו שני סוגי זיכרון מטמון:

מטמון במיפוי ישיר הבנוי מ-4 בלוקים של 8 בתים כל אחד.

מטמון במיפוי 2-Way Set associative המכיל 16 סטים בכל סט 2 בלוקים של 4 בתים כל אחד.

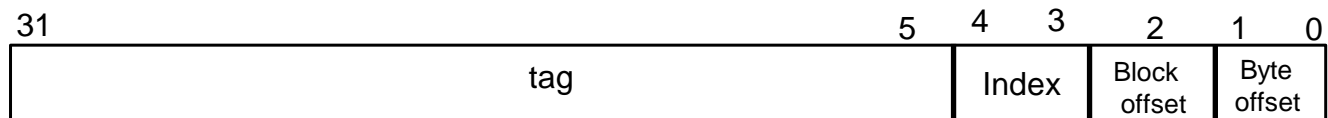
מדיניות החלפת בלוקים - LRU

מה מספר הפגיעות בכל אחד מזיכרונות המטמון ?

מטמון במיפוי ישיר הבנוי מ-4 בלוקים של 8 בתים כל אחד.

$n=2$

$m=1$

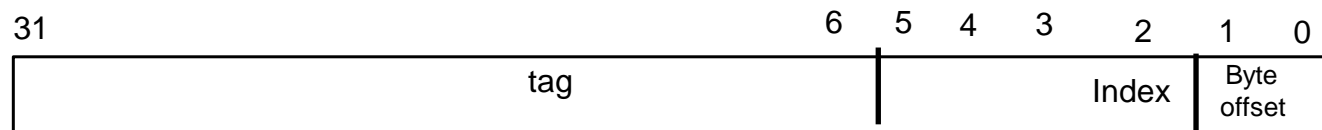


DMC($n=2$ $m=1$)

index	V	TAG	DATA	
			Word 0	word1
00	0			
01	0			
10	0			
11	0			

מטמון במיפוי 2-Way Set associative המכיל 16 סטים בכל סט 2 בלוקים של 4 בתים
כל אחד.

$n=4$
 $m=0$

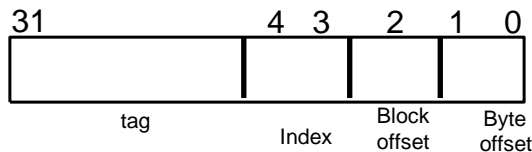


סידרת הגישות 7,13,37,27,33,11,7,18,34,23

DMC(n=2 m=1)

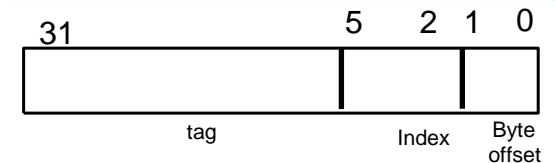
index	V	TAG	DATA	
			Word 0	word1
00	1	0	32 33 34 35	36 37 38 39
01	1	0	8 9 10 11	12 13 14 15
10	1	0	16 17 18 19	20 21 22 23
11	1	0	24 25 26 27	28 29 30 31

DMC		גישה	2-way
Miss-V	0111	7	Miss-V
Miss-V	1101	13	Miss-V
Miss-T	100101	37	Miss-V
Miss-V	11011	27	Miss-V
HIT	100001	33	Miss-V
HIT	1011	11	Miss-V
Miss-T	0111	7	HIT
Miss-V	10010	18	Miss-V
Miss-T	100010	34	HIT
HIT	10111	23	Miss-V



2-way (n=4 ,m=0, k=2)

Set	Way0	Way1					
	Index	V	TAG	DATA	V	TAG	DATA
0	0000	0			0		
1	0001	1	0	4 5 6 7	0		
2	0010	1	0	8 9 10 11	0		
3	0011	1	0	12 13 14 15	0		
4	0100	0	0	16 17 18 19	0		
5	0101	1	0	20 21 22 23	0		
6	0110	1	0	24 25 26 27	0		
7	0111	0			0		
8	1000	1	0	32 33 34 35	0		
9	1001	1	0	36 37 38 39	0		
10	1010	0			0		
11	1011	0			0		
12	1100	0			0		
13	1101	0			0		
14	1110	0			0		
15	1111	0			0		



שקף 13

נושא: שאלות זיכרון מטמון
Cache Memory

נתון זיכרון מטמון בגודל 16 מילים, אסוציאטיבי מלא ועם בלוקים בגודל 4 מילים. מתבצעת גישה לכתובות (של מילים) הבאות (משמאל לימין). הניחו שיטת החלפה LRU.

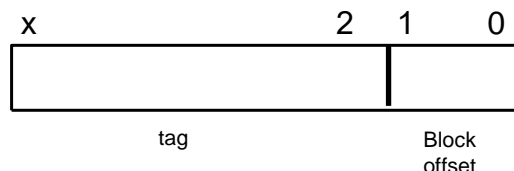
2,3,11,16,0,8,21,13

א. רשמו ליד כל כתובת אם היא פגיעה או החטאה
ב. הראו את התוכן הסופי של זיכרון המטמון.

m=2 n=0 k=4

כלומר המטמון הוא שורה (סט) אחת המורכבת מ 4 בלוקים (4-way)
כל בלוק גודלו 4 מילים סה"כ 16 מילים

אין צורך להציג במיפוי את שדה ה byte offset משום שהכתובות נתונות במילים ולכן המיפוי יהיה:



זיכרון מטמון בגודל 16 מילים, אסוציאטיבי מלא ועם בלוקים בגודל 4 מילים.
 למימוש ה LRU יש k אפשרויות כלומר 4! כלומר 24 צורות סידור שונות של ה way (בלוקים)

היות ויש רק שורה (set) אחת נסמן את ה ways (בלוקים) אחד מעל השני.
 במצב הראשוני של המטמון ריק נניח מצב איכלוס מלמעלה (Block0) עד למטה (Block3)
 -נסמן מצב זה בסדר LRU כ 1 2 3 4. (כאשר 4 מסמן את הבלוק להחלפה 3 את זה שאחריו וכן הלאה , בסדר יורד.
 מבחינת המיקום בסדר LRU צד שמאל מתייחס ל block0 וצד ימין ל Block3

סדר ה LRU		גישה במילים	תוצאה
4 3 2 1		לפני	
1 4 3 2	000010	2	Miss-V
1 4 3 2	000011	3	HIT
2 1 4 3	001011	11	Miss-V
3 2 1 4	010000	16	Miss-V
1 3 2 4	000000	0	HIT
2 1 3 4	001000	8	HIT
3 2 4 1	010101	21	Miss-V
4 3 1 2	001101	13	Miss-T

#block	Valid	Tag	Word0	Word1	Word2	Word3
0	1	0000	Mem[0)	Mem[1)	Mem[2)	Mem[3)
1	1	0010	Mem[8)	Mem[9)	Mem[10)	Mem[11)
2	1	0011	Mem[12)	Mem[13)	Mem[14)	Mem[15)
3	1	0101	Mem[20)	Mem[21)	Mem[22)	Mem[23)

$$n=0, m=2, k=4$$

