

זיכרון המטמון בנוי מבלוקים של סיביות, המטמון מאורגן לפי בלוקים של סיביות. כאשר מידע חדש מגיע למטמון, הוא מגיע בתוך בלוק מהזיכרון. בזיכרון המטמון יש רק כמות מוגבלת של מקומות לבלוקים, ולכן נשאלת באיזה אופן יסודרו הבלוקים, ומהו המקום המיטבי לשים בו בלוק חדש שהגיע? יש מספר שיטות מיפוי שונות של בלוקים למקומות מתאימים בזיכרון המטמון, נפרט אותם.

מיפוי ישיר:

בשיטה זו כל כתובת בזיכרון ממופה ישירות למיקום יחידה אפשרי בזיכרון המטמון. נזכיר שהכתובות בזיכרון, הן בבתים. גם בכל בלוק יש כמות מסוימת של בתים. נסביר תחילה על כתובת הבלוק:

- משמעות המושג "כתובת הבלוק" של כתובת מסוימת - היא לקבל את הכתובת ביחידות מידה של בלוקים במקום ביחידות מידה של בתים. כדי לעבור מכתובת בבתים לכתובת הבלוק, נצטרך לחלק את הכתובת (בבתים) בכמות הבתים שיש בכל בלוק.
- הסיבה שנרצה לדבר על כתובת הבלוק היא - שהמטמון מורכב מבלוקים ולכן כתובת מסוימת תזוהה, באמצעות הבלוק שהיא שייכת אליו.
- לדוגמה אם הכתובת בבתים היא $0x1234567$ ובכל בלוק יש 4 מילים (כלומר 16 בתים), כדי לחשב את כתובת הבלוק נמיר קודם לייצוג בינארי. הכתובת בבתים היא $0001001000110100010101100111_b$, המשמעות של לחלק ב-16 ($2^4 = 16$) היא למחוק את 4 הסיביות הימניות, כלומר נקבל שכתובת הבלוק היא $000100100011010001010110_b$, נמיר חזרה להקסא ונקבל שכתובת הבלוק היא $0x123456$.
- כעת נוכל להסביר שזיכרון המטמון במיפוי ישיר הוא מעין טבלה ארוכה כך שבכל שורה נכנס רק בלוק אחד.
- כדי לדעת לאיזה שורה ימופה בלוק מסוים מהזיכרון נחשב: **כתובת הבלוק מודולו כמות הבלוקים במטמון**.

נניח שבמטמון שלנו יש 4 בלוקים (כלומר מספר השורה יכול להיות בין 0 ל 3 בגלל המודולו), במקרה נקבל עבור כתובת הבלוק שחישבנו קודם $0x123456 \% 4 = 000100100011010001010110_b \% 4 = 10_b = 2$

כלומר כתובת זו תמופה לשורה מספר 2 במטמון (כאשר מתחילים לספור מ-0) כלומר בשורה השלישית.

כל שורה במיפוי הישיר, מייצגת בלוק, ויש לה 3 שדות: שדה ה data שמכיל את תוכן הבלוק בזיכרון, גודל שדה זה כגודל הבלוק (בדוגמה שלנו 16 בתים).

כל כתובת ממופה לפי הסיביות הנמוכות שלה, ולכן כדי להבדיל בין כתובות שונות שיש להן את אותן סיביות נמוכות (כמו 10 בדוגמה לעיל)

נשתמש בשדה ה tag, שדה זה יכיל את כל הסיביות הגבוהות של הכתובת, כלומר 0001001000110100010101

ושדה ה Valid הוא סיבית אחת שכאשר הערך שלה הוא 0, אפשר לדעת שהבלוק עדיין ריק ממידע.

Valid	Tag	Data	מספר שורה
			00_b
			01_b
			10_b
			11_b

מיפוי קבוצתי K-אסוציאטיבי

- ☐ **זיכרון קבוצתי K-אסוציאטיבי (K-way set-associative cache)**
 בשיטה זו - בזיכרון המטמון יש מספר קבוע של כניסות שבהן אפשר לשים כל בלוק - K. כל בלוק מהזיכרון ממופה חד ערכית לקבוצה יחידה (set) של כניסות (ways) בזיכרון המטמון. כל קבוצה כזו מסומנת באמצעות שדה האינדקס. הבלוק יכול להיכנס לכל אחת מהכניסות שלה. בשיטה זו מגיעים לאינדקס הקבוצה המתאימה, ובתוכה מושוות כל התגיות לתגית המבוקשת.
- ☐ באיור ניתן לראות המחשה של זיכרון 5 אסוציאטיבי עם 4 קבוצות. סיביות ה Valid הושמטו לצורך נוחות.

- באיור ניתן לראות המחשה של זיכרון 5 אסוציאטיבי עם 4 קבוצות. סיביות ה Valid הושמטו לצורך נוחות.

[illegible]

מיפוי קבוצתי K-אסוציאטיבי

זיכרון קבוצתי K-אסוציאטיבי (K-way set-associative cache)

□ נסמן:

❖ כמות הבתים בזיכרון הראשי: 2^u (כתובת היא בבתיים)

❖ כמות הבתים של ה data במטמון: 2^c

❖ כמות הבתים בכל בלוק: 2^b

○ כמות המילים בכל בלוק: $2^m = 2^{b-2}$

❖ כמות סיביות ה tag לכל בלוק: $\text{tag} = u - c + \lg(k)$

□ נקבל:

❖ מספר הבלוקים במטמון: 2^{c-b}

❖ מספר הסטים במטמון: $\frac{2^{c-b}}{k}$ סטים. (בכל כניסה k בלוקים)

נחלק כל כתובת בבינארית בזיכרון בבתיים לפי השדות הבאים:

Tag	Set/Index(n)	Block offset/Word offset	כמות הבתים בכל מילה
סיביות ה tag בשורה	מספר הקבוצה (סט) במטמון	מספר מילה בבלוק	מספר בית במילה
$u - c + \lg(k)$	$c - b - \lg(k)$	$b - 2 = m$	2

u

מספר סיביות tag:

שווה לכמות הבלוקים כפול כמות
סיביות ה tag בבלוק: $2^{c-b} * \text{tag}$

מספר סיביות הכולל במטמון כולל

valid bit, כולל שדה tag:

כמות המידע בסיביות בתוספת
סיביות ה tag וה valid.

$$2^{c+3} + 2^{c-b}(\text{tag} + 1)$$

הערה: מיפוי ישיר הוא

1 אסוציאטיבי ($k=1$, $\lg(k)=0$)

דוגמה : מיפוי 4-אסוציאטיבי

□ נתון זיכרון מטמון 4-אסוציאטיבי עם הנתונים הבאים:

□ Byte offset = 2 , Block offset - 4 , Index = 14 , Tag = 12

עלינו למצוא את הכמות הכוללת של סיביות בזיכרון מטמון זה כולל את ה Valid , tag ו-data.

□ פיתרון : נעבוד על פי הסימון שהוצג ונמלא את הטבלה :

Tag	Set/Index(n)	Block offset/Word offset	כמות הבתים בכל מילה
סיביות ה tag בשורה	מספר הקבוצה (סט) במטמון	מספר מילה בבלוק	מספר בית במילה
$u - c + \lg(k)$	$c - b - \lg(k)$	$b - 2 = m$	2
12	14	4	2
u			

מהסימונים בטבלה ומנתוני השאלה נוכל להסיק ש :

$u = 12 + 14 + 4 + 2 = 32$, $k=4$, $\lg k = 2$, $b = m + 2 = 6$, $c = 14 + b + \lg k = 22$, $\text{tag} = 12$.
 כמו כן כדי למנוע סרבול בנוסחאות נשים לב שגם $\text{tag} + 1 = 13$, $c - b = 16$, $c + 3 = 25$.

אם כן לפני הנוסחה השנייה נקבל שמספר הסיביות הכולל בזיכרון המטמון הוא :

$$2^{25} + 2^{16} \cdot 13 = 34,406,400$$

מספר סיביות tag :

שווה לכמות הבלוקים כפול כמות
 סיביות ה tag בבלוק : $2^{c-b} \cdot \text{tag}$

מספר סיביות הכולל במטמון כולל

valid bit , כולל שדה tag :

כמות המידע בסיביות בתוספת
 סיביות ה tag וה valid .

$$2^{c+3} + 2^{c-b}(\text{tag} + 1)$$

הערה : מיפוי ישיר הוא

1 אסוציאטיבי ($k=1$, $\lg(k)=0$)

מיפוי אסוציאטיבי מלא

כל בלוק במטמון יכול להיכנס לכל כניסה של הזיכרון. אם יש n בלוקים במטמון, אז בלוק חדש יכול להשתבץ בכל אחד מהם, בשונה מהמיפוי הישיר שבו לכל בלוק היה רק מקום אופציונלי אחד. הדבר דומה לשורה ארוכה של בלוקים, כאשר נרצה לבדוק האם בלוק מסוים נמצא במטמון, עלינו לבדוק את כל n האפשרויות.

❑ נסמן:

❖ כמות הבתים בזיכרון הראשי: 2^u (כתובת היא בבתים)

❖ כמות הבתים של ה data במטמון: 2^c

❖ כמות הבתים בכל בלוק: 2^b

○ כמות המילים בכל בלוק: $2^m = 2^{b-2}$

❖ כמות סיביות ה tag לכל בלוק: $tag = u - b$

❖ מספר הבלוקים במטמון: 2^{c-b} .

מספר סיביות tag:
שווה לכמות הבלוקים כפול כמות
סיביות ה tag בבלוק: $2^{c-b} * tag$

מספר סיביות הכולל במטמון כולל
valid bit, כולל שדה tag:
כמות המידע בסיביות בתוספת
סיביות ה tag וה valid.
 $2^{c+3} + 2^{c-b}(tag + 1)$

Tag	Block offset/Word offset	כמות הבתים בכל מילה
סיביות ה tag בשורה	מספר מילה בבלוק	מספר בית במילה
$u-b$	$b-2 = m$	2
u		

מיפוי אסוציאטיבי מלא

פה ניתן לראות המחשה של זיכרון מטמון במיפוי אסוציאטיבי מלא
כאשר יש 4 בלוקים במטמון.

[illegible]