سازمان سما

وابسته به دانشگاه آزاد اسلامی

دانشگاه سما واحد حاجی آباد

حل مسئله
معماری کامپیوتر

منبع : معماری کامپیوتر-منوچهر بابایی

WWW.HREZAPOUR.IR

حمیدرضا رضاپور

معماری کامپیوتر

درس دهم: سازمان حافظه

(سلسله مراتب حافظه)

- حافظه نهان(Cache)
- حافظه اصلی(Main Memory)
- حافظه کمکی(Auxiliary Memory)

حافظه اصلی برای ذخیره داده ها و برنامه ها در حین عملکرد کامپیوتر مورد استفاده قرار می گیرد

این نوع حافظه دو نوع است:

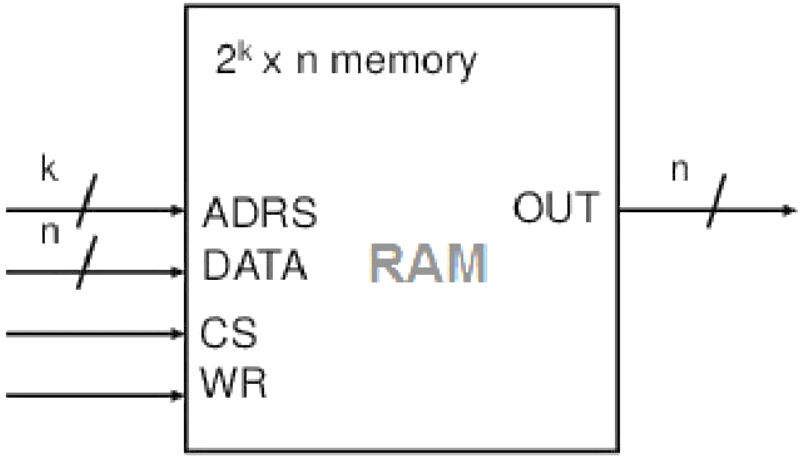
۱-حافظه با دسترسی تصادفی(RAM)

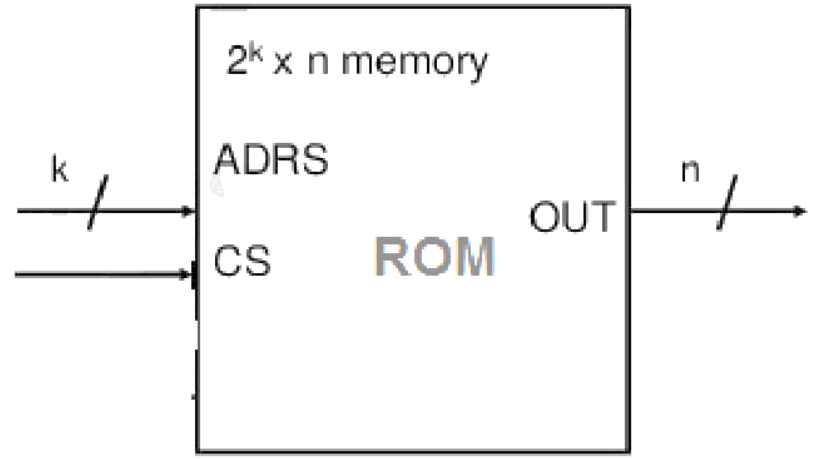
۲-حافظه فقط خواندنی(ROM)

حافظهRAM از نقطه نظر نگه داری اطلاعات دو نوع است:

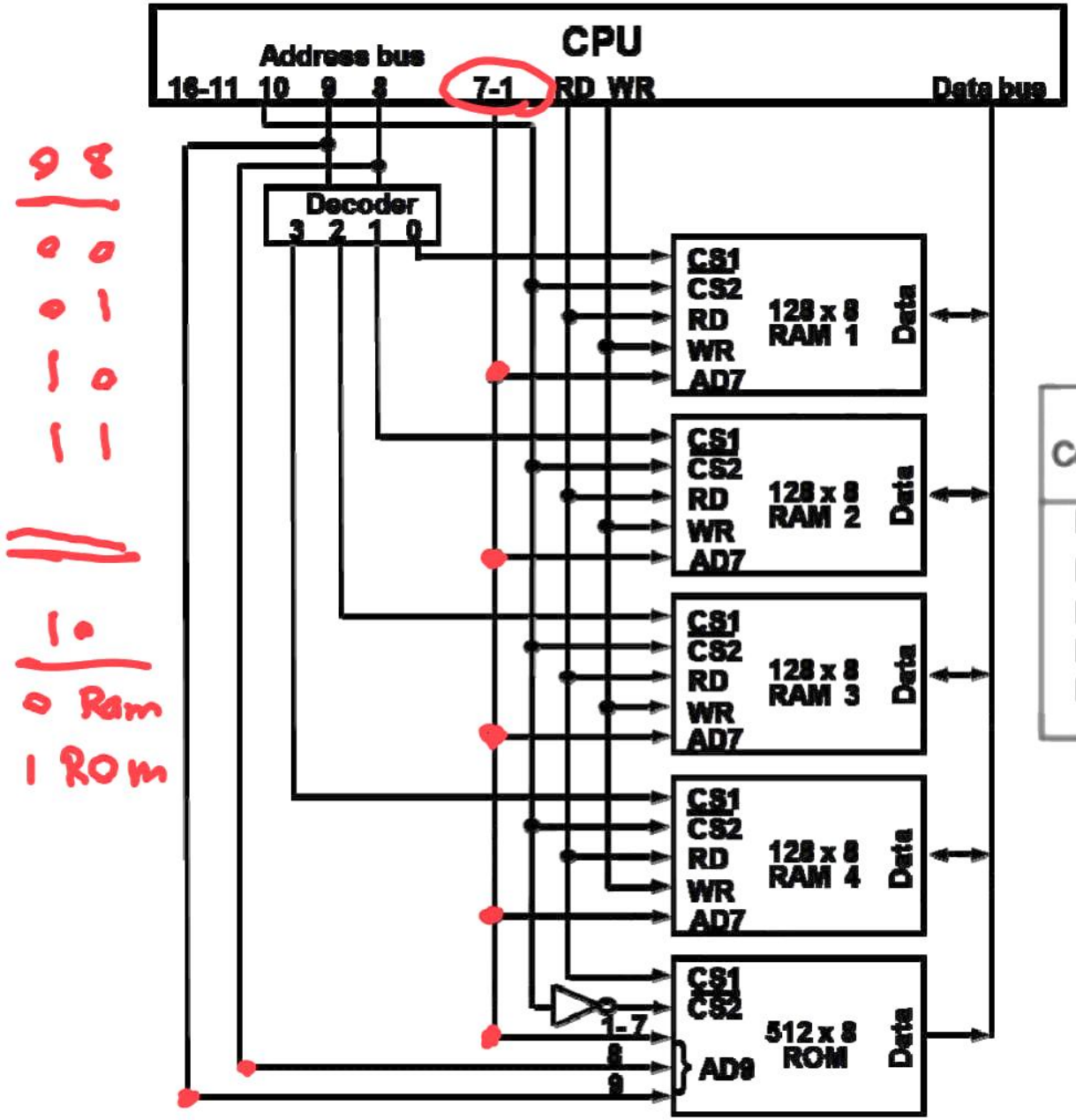
۱-ایستا(Static) :تا هنگامیکه کامپیوتر روشن باشد اطلاعات آن از بین نمی رود

۲-پویا(Dynamic) :به مرور زمان اطلاعات آن از بین می رود.لذا بصورت دوره ای تازه سازی شود.





CS	WR	Memory operation
0	x	None
1	0	Read selected word
1	1	Write selected word



$128 = 2^7$

$512 = 2^9$

0000 0001

Component	Hexa address	Address bus									
		10	9	8	7	6	5	4	3	2	1
RAM 1	0000 - 007F	0	0	0	x	x	x	x	x	x	x
RAM 2	0080 - 00FF	0	0	1	x	x	x	x	x	x	x
RAM 3	0100 - 017F	0	1	0	x	x	x	x	x	x	x
RAM 4	0180 - 01FF	0	1	1	x	x	x	x	x	x	x
ROM	0200 - 03FF	1	x	x	x	x	x	x	x	x	x

0010

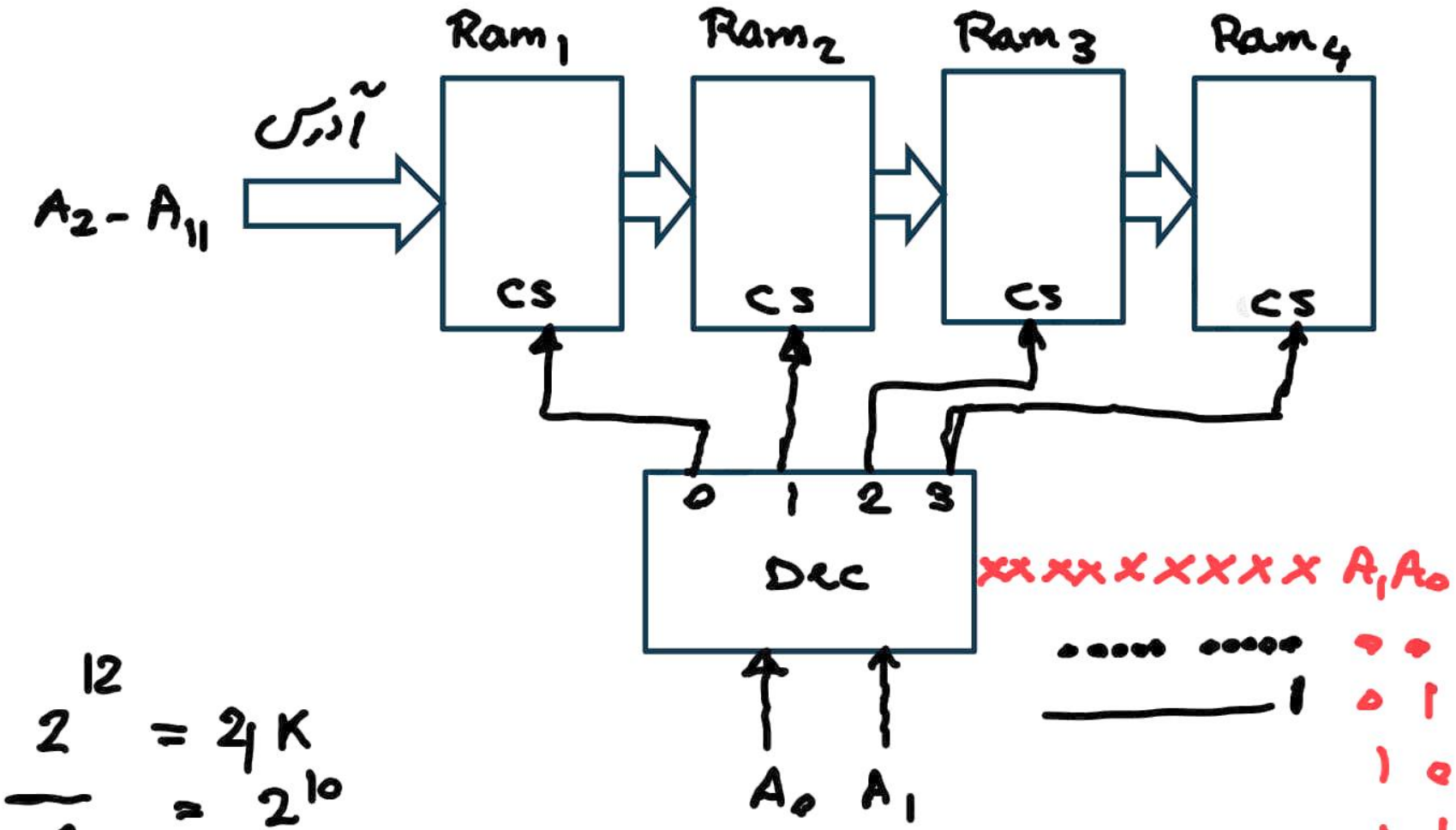
(برگ،برگ کردن حافظه)
Memory Inter Leaving

برای جبران کندی حافظه،(زمان دستیابی به آدرس های مختلف)از روش برگ،برگ کردن حافظه استفاده می شود.یعنی با قرار گیری یک ادرس خاص روی گذرگاه بسته نوع مکانیسم،دسترسی به آدرس های مختلف امکانپذیر می شود.دو روش برای اینکار وجود دارد:

۱-High-Order: بیت های بالای گذرگاه آدرس بعنوان انتخاب تراشه های برگ انتخاب می شوند.آدرس های متوالی در یک برگ قرار می گیرند.

۲-Low-Order: بیت های پایین گذرگاه آدرس بعنوان انتخاب تراشه های برگ انتخاب می شوند.آدرس های متوالی در برگ های مختلف قرار می گیرند.

روش Low-order



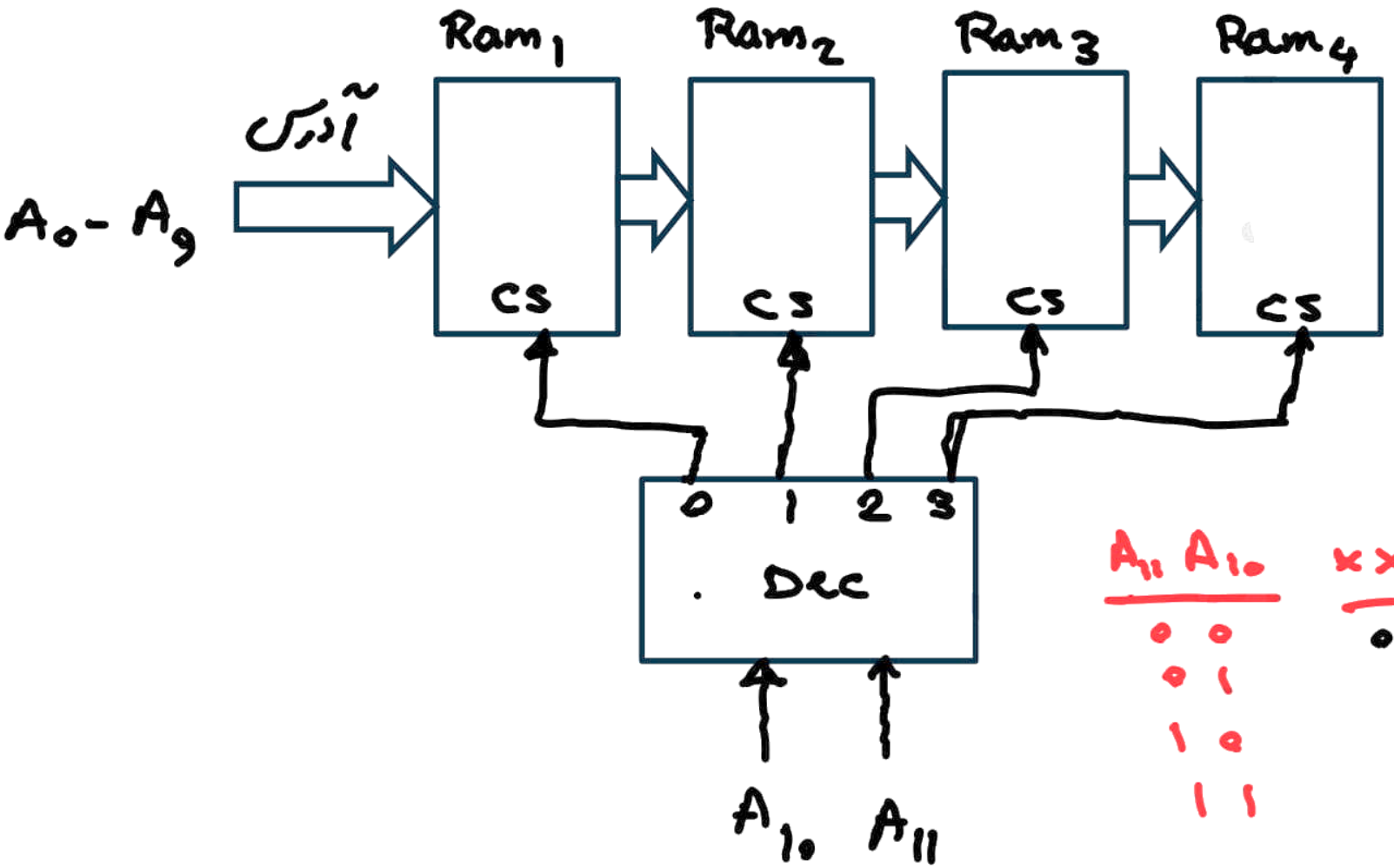
$$\frac{2^{12}}{2^{2}} = 4K = 2^{10}$$

xxxxxxxxxx A_1A_0

0 0	$\xrightarrow{1}$ 0, 4, 8
0 1	$\xrightarrow{2}$ 1, 5, 9
1 0	$\xrightarrow{3}$ 2, 6, 10
1 1	$\xrightarrow{4}$ 3, 7, 11

$i + 4j$

روش High-order



A_{11}	A_{10}
0	0
0	1
1	0
1	1

xxxxxxxxxx

$\xrightarrow{1}$ 0

$\xrightarrow{2}$ 1024

$\xrightarrow{3}$ 2048

$\xrightarrow{4}$ 2048 + 1024 = 3072

مثال (۱): میخواهیم سرعت دسترسی به آدرس های متوالی حافظه با فاصله آدرسی 2 را افزایش بدهیم.اگر حجم حافظه ۲۰۴۸ واحد آدرس پذیر باشد.اگر از ۸ برگ برای اینکاراستفاده کنیم کدام روش برای انتخاب تراشه های برگ مناسبتر است؟

الف-High-Order ب-Low-Order ج-بیت های A1 تا A3

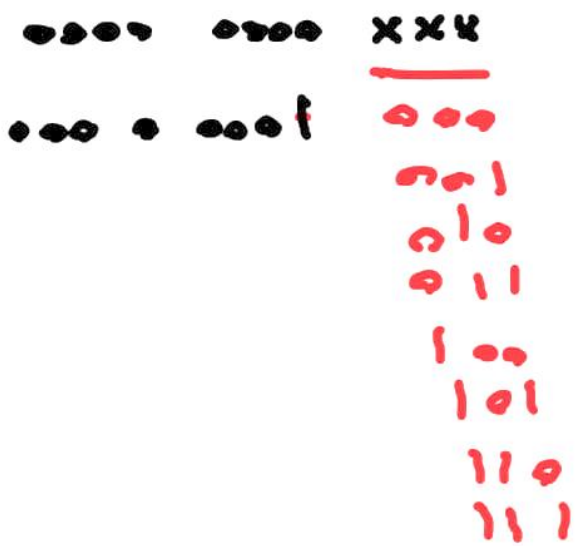
0 - 2 - 4 - 6 - 8 ...

$$\frac{2048}{8} = \frac{2^{11}}{2^{3}} = 2^{8}$$

T 0 - 256 - 512 - 768 - 1024 - 1280 - 1536 - 1792

1 - 257 - 513 ....

T 2 - 258 ....



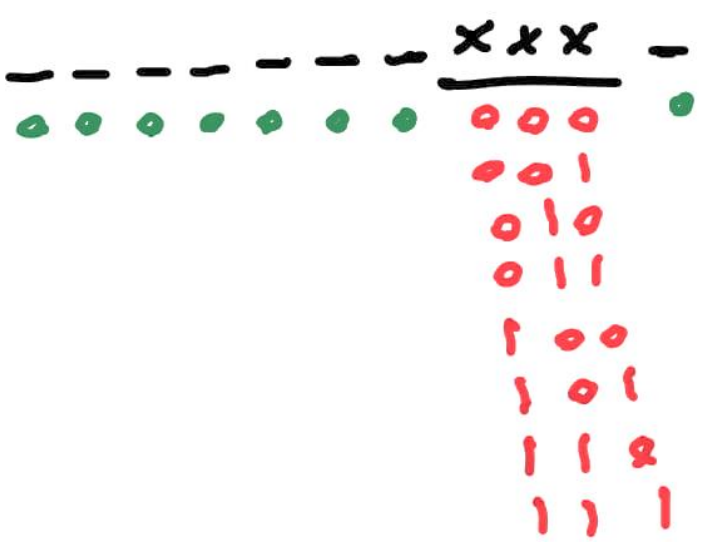
T: 0, 1, 2, 3, 4, 5, 6, 7

T: 8, 9, 10, 11, 12, 13, 14, 15

16, 17, 18, 19, 20, 21, 22, 23

T: تأخیر دسترسی به حافظه

$\rightarrow \frac{T}{4}$



T: 0 - 2 - 4 - 6 - 8 - 10 - 12 - 14

16 - 18 - 20 - 22 - 24 - 26 - 28 - 30

$\rightarrow \frac{T}{8}$ (بهترین راهکار)

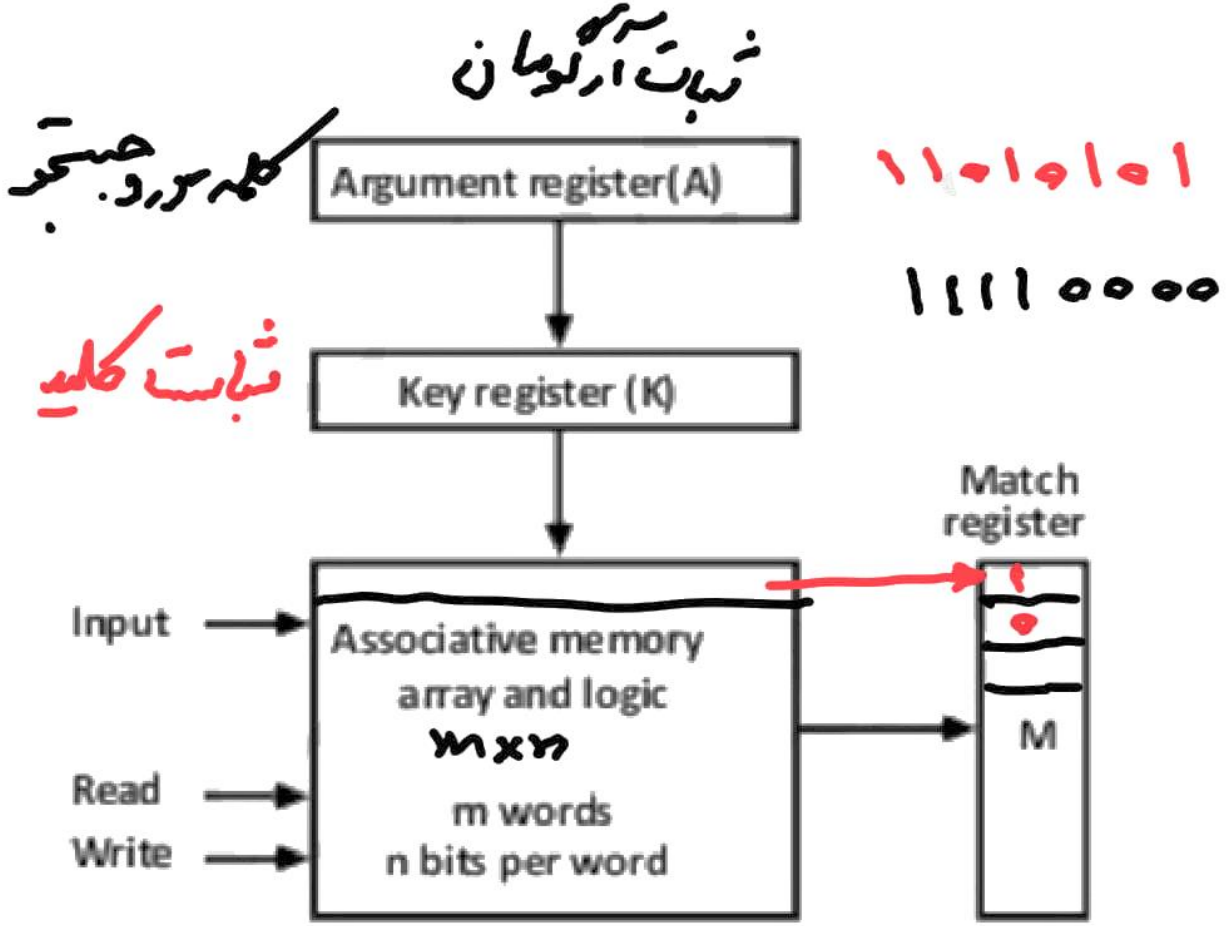
مثال (۲): فرض کنید که یک ارایه ۸*۸ در حافظه به ترتیب سطر بار شده است،کدامیک ازسازمان های حافظه سریعترین دسترسی به سومین ستون آرایه را ممکن می سازد؟حجم حافظه 2^{16} کلمه است که متشکل از ۸ پیمانه حافظه 2^{13} کلمه ای است)

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
		18					
		26					
		34					
		42					
		50					
		58					

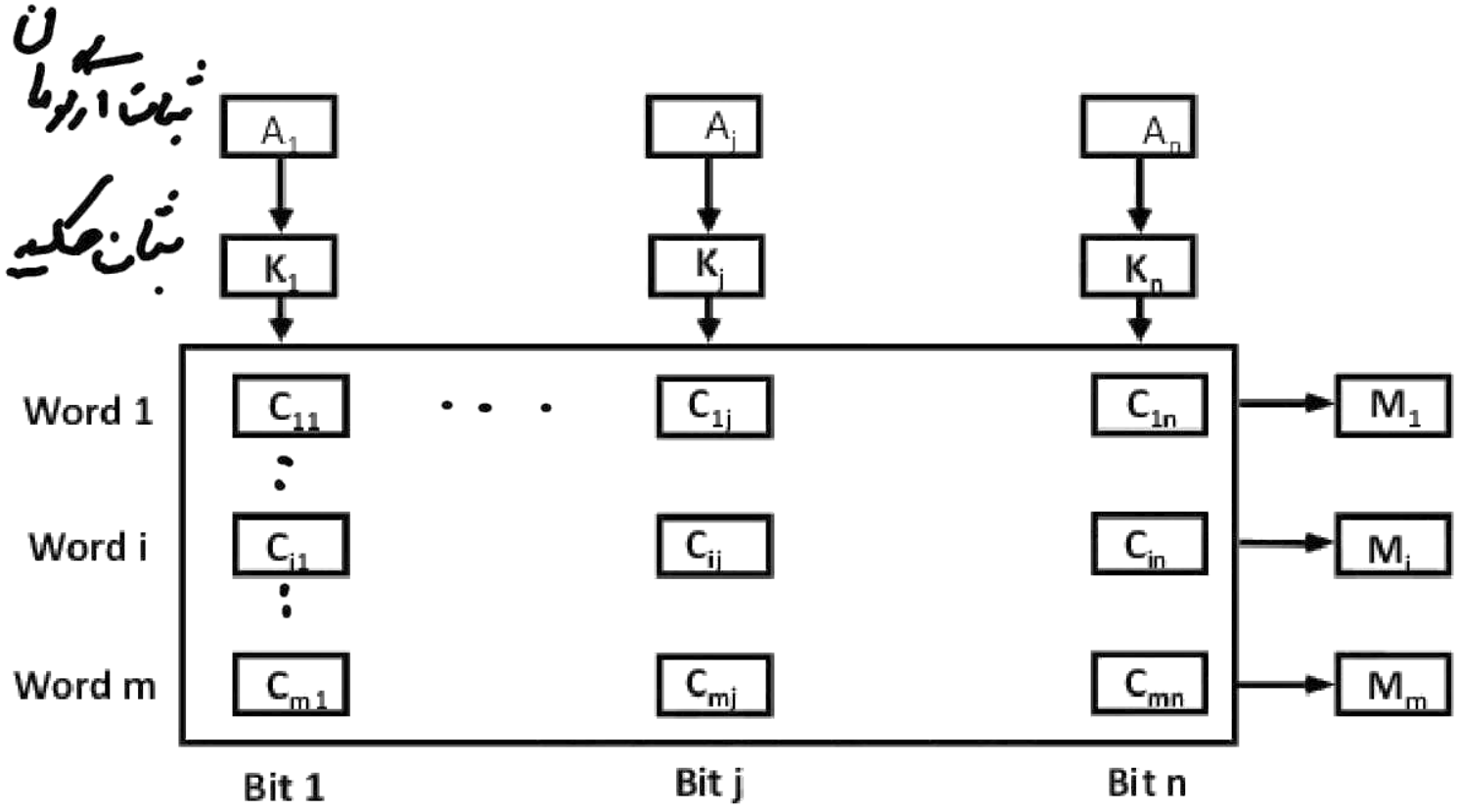
(حافظه تداعیگر)

Associative Memory

CAM



(تطبیق کلمه در حافظه)



A = 101100

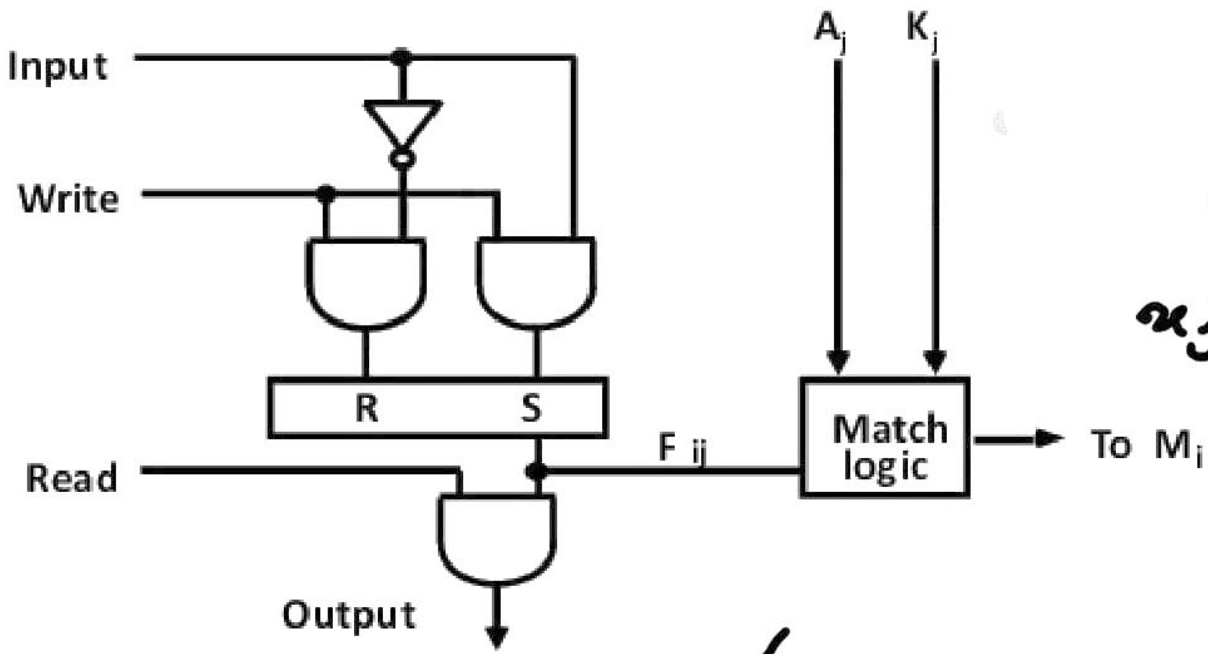
K = 011001

001001 $M_1 = 0$

001010 $M_2 = 1$

100110 $M_3 = 0$

(ساختمان داخلی هر سلول از کلمه حافظه)



انطباق

$A_j = F_{ij}$

$x_j = (A_j \oplus F_{ij})'$

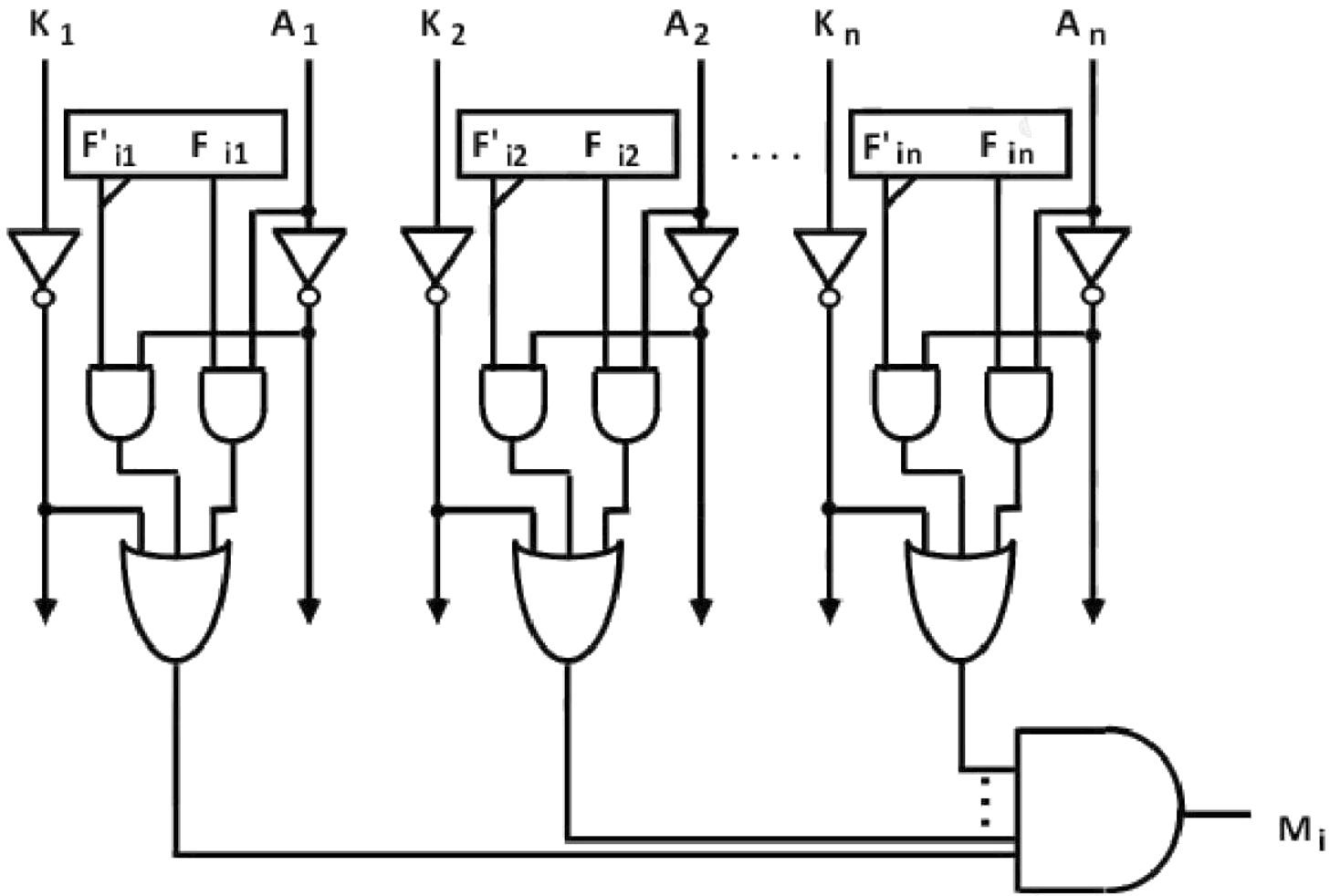
$\rightarrow x_j = A_j F_{ij} + A_j' F_{ij}'$ $\xrightarrow{\text{انطباق کلی}}$ $x_j = 1 \ , \ j = 1:n$

$M_i = 1 \longrightarrow M_i = x_1 x_2 x_3 \cdots x_n$

$$\begin{cases} 1 & K_j = 0 \\ x_j & K_j = 1 \end{cases} \rightarrow x_j + K'_j \qquad \text{تا}$$

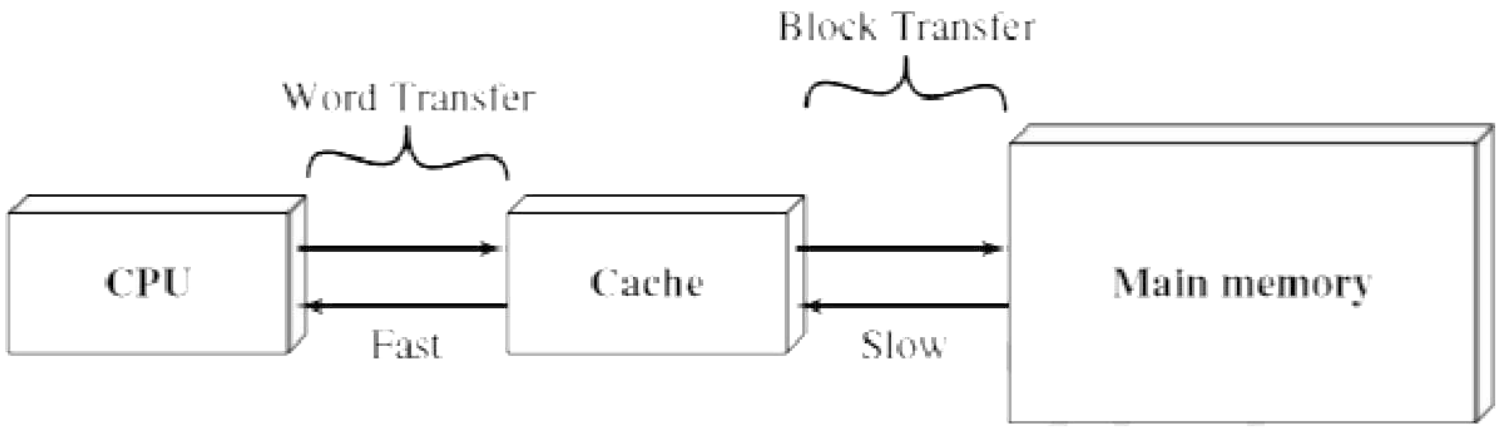
$$M_i = \bigwedge_{j=1}^{n} (x_j + K'_j) \rightarrow M_i = \bigwedge_{j=1}^{n} (A_j K_{ij} + A'_j K'_{ij} + K'_j)$$

(مدار انطباق برای تست یک کلمه)

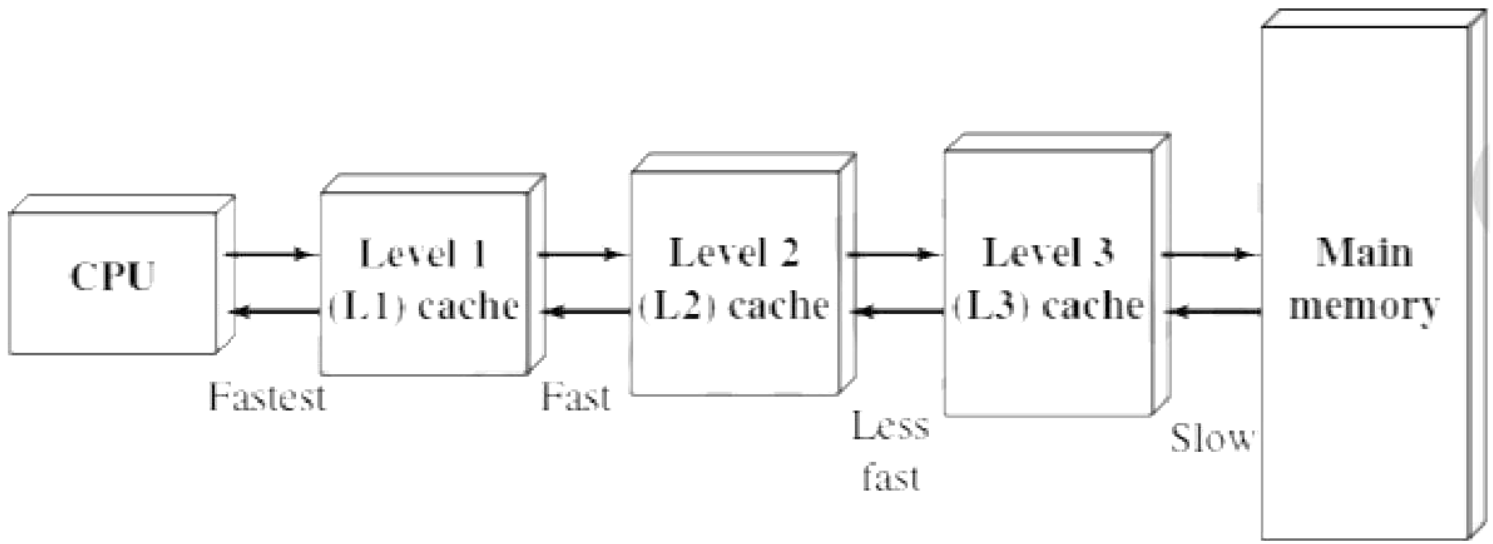


(حافظه کش)

- برای داشتن یک حافظه سریع،که بتواند خلا عدم هماهنگی بین سرعت پردازنده و حافظه اصلی را حل کند از کش(حافظه نهان) استفاده می شود.که بین پردازنده و حافظه اصلی قرار می گیرد.
- هنگامی پردازنده،یک آدرس روی گذرگاه قرار می دهد ابتدا به حافظه کش مراجعه می کند و در صورت نبود داده،به حافظه اصلی مراجعه می کند.



(a) Single cache



(b) Three-level cache organization

(محلی بودن مراجعات)
Locality Reference

- محلی بودن مراجعات یعنی اینکه هنگام اجرای یک برنامه ،آدرس های تولید شده مکررا به یک یا چند ناحیه محلی از مراجعه می کنند.بعنوان مثال خواندن داده های یک آرایه که بصورت پشت سر هم در حافظه ذخیره شده اند.
- لذا با پیش بینی این موضوع میتوان آن بخش از برنامه را تحت یک بلوک به حافظه نهان منتقل کرد تا در مراجعات بعدی سرعت دسترسی بیشتر باشد.

(انواع محلیت های ارجاعی)

- محلیت زمانی(Temporal locality):اگر به یک مکان از حافظه مراجعه شود امکان دارد به زودی دوباره به آن رجوع شود.
- محلیت مکانی(Spatial locality): اگر به یک مکان از حافظه رجوع شود به احتمال زیاد به زودی به مکان های مجاور آن رجوع می شود.

- برخورد(Hit): هرگاه پردازنده به حافظه کش مراجعه کند و کلمه مورد نظر در کش باشد برخورد نرخ می دهد.
- فقدان-باخت(Miss): اگر کلمه موردنظر پردازنده در کش حاضر نباشد میگوییم فقدان رخ داده است
- نرخ برخورد(Hit Ratio) : تعداد برخوردها در مراجعات تقسیم بر کل مراجعات به کش

$$h = \frac{\text{تعداد برخوردها}}{\text{تعداد کل مراجعات}}$$

(متوسط زمان دسترسی به سیستم حافظه
هنگام خواندن)

- اگر دسترسی پردازنده به حافظه ها بصورت سریال باشد:

$$t_a = ht_{Cache} + (1-h)(t_{Cache} + t_{Main}) = t_{Cache} + (1-h)t_{Main}$$

- اگر دسترسی بصورت سریال نباشد:

$$t_a = ht_{Cache} + (1-h)t_{Main}$$

- هنگام انتقال یک داده از حافظه اصلی به حافظه کش باید یک فرآیند برای پیدا نمودن مکان کلمه در حافظه کش داشته باشیم،به این فرآیند نگاشت(Mapping) گفته می شود.

سه روش نگاشت برحسب سازمان حافظه کش وجود دارد:

۱-نگاشت تداعیگر(انجمنی) (Associative Mapping)

۲-نگاشت مستقیم(Direct Mapping)

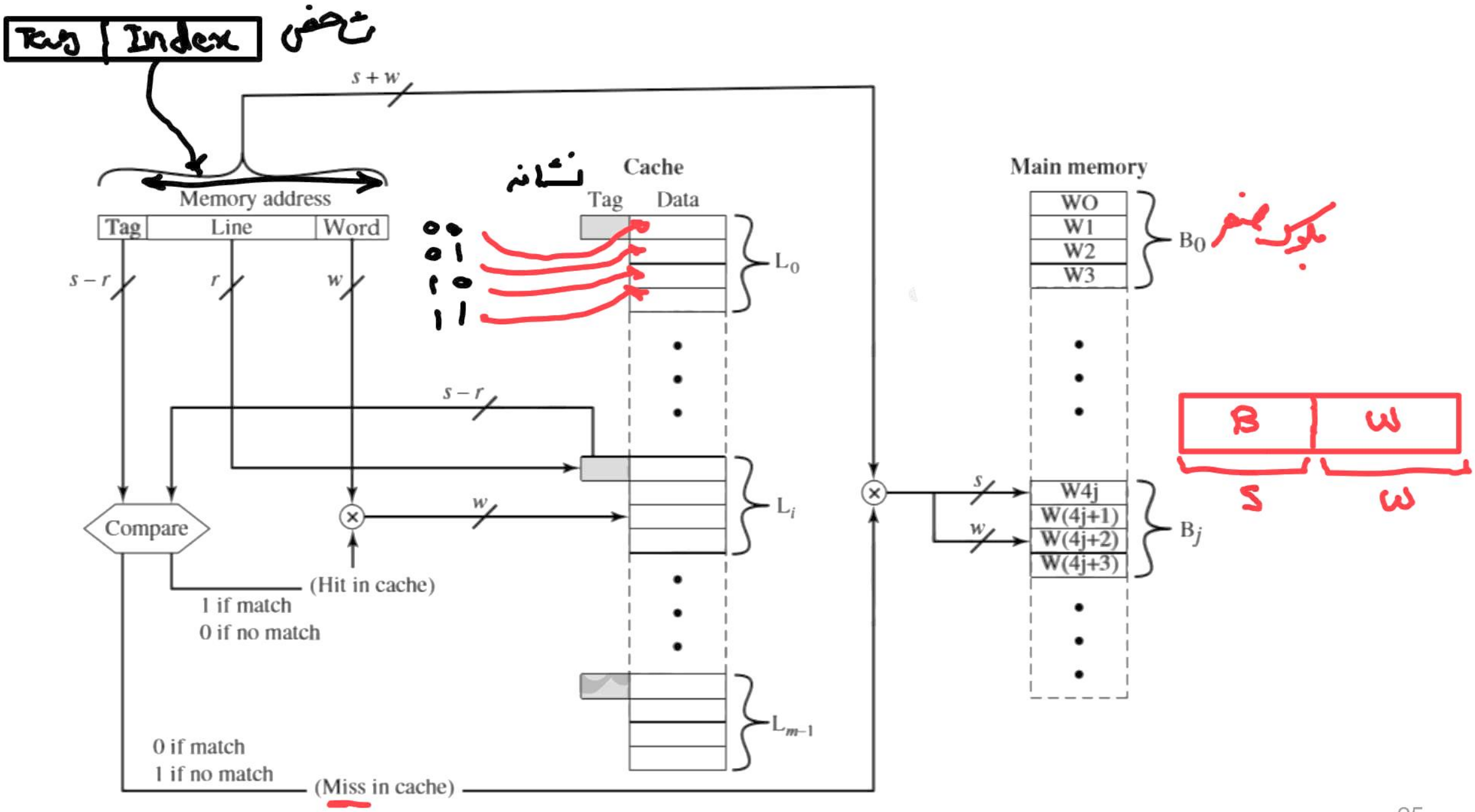
۳-نگاشت تداعیگر مجموعه ای(Set Associative Mapping)

نگاشت مستقیم(Direct Mapping)

- ساده ترین روش نگاشت می باشد.در این روش هر بلوک از حافظه اصلی فقط و فقط به یک مکان مشخص از حافظه کش نگاشت می شود.

(تعداد بلوکهای کش) modulo (آدرس بلوک در حافظه اصلی)=آدرس مربوط به بلوک در حافظه کش

Cache line	Main memory blocks assigned
0	$0, m, 2m, \ldots, 2^s - m$
1	$1, m + 1, 2m + 1, \ldots, 2^s - m + 1$
⋮	⋮
$m - 1$	$m - 1, 2m - 1, 3m - 1, \ldots, 2^s - 1$


Tag | Index
شخص
s + w
Memory address
Tag
Line
Word
s − r
r
w
نشانه
Cache
Tag
Data
L0
Li
Lm−1
Compare
(Hit in cache)
1 if match
0 if no match
0 if match
1 if no match
(Miss in cache)
Main memory
WO
W1
W2
W3
B0
بلوک صفر
W4j
W(4j+1)
W(4j+2)
W(4j+3)
Bj
s
w
B
w
s
w

(بعنوان مثال)

در یک حافظه کش با ۳۲ بلوک و ۲۵۶ کلمه،اگر حجم حافظه اصلی ۲۰۴۸ کلمه باشد آنگاه تعداد بیت های Index،Tag word و Block(Line) بصورت زیر است:

	Block	Word

Tag Index

$2048 = 2 \times 1024 = 2^{11}$

$256 = 2^{8}$

$32 = 2^{5}$

Tag+Index=11
Index=8
Tag=3
Block=5
Word=3

مثال (۳): اگر یک حافظه کش دارای ۶۴ بلوک و اندازه هر بلوک ۳۲ بایت باشدانگاه شماره بلوک مربوط به بایت ۷۰۰ و آدرس ۷۰۰ را بیابید(روش نگاشت مستقیم)

$$\left\lfloor \frac{700}{32} \right\rfloor = 21$$

$$21 \% 64 = 22$$

$$\begin{cases} 700 \% 256 = 188 \\ 188 \% 64 = 60 \end{cases} \quad , \quad 700 \% 64 = 60$$

مثال (۴):یک حافظه اصلی دارای ۲۵۶ کیلو کلمه و یک حافظه کش به بزرگی ۴ بلوک ۴ کلمه ای موجود است،اگر از روش نگاشت مستقیم(Direct Mapping) استفاده شود آنگاه با فرض خالی بودن کش،نرخ برخورد(Hit Rate) در انتهای صدور آدرس های زیر از طرف پردازنده را بیابید.

$AA, 101, A8, 103, B0 \quad Hex$

گنجایش کش = 16 کلمه → Index = 4 bit

حافظه اصلی = 256×2^{10} → آدرس حافظه = 18 bit

14	2	2
Tag	block	word

AA : A 10 10 → $B_2 W_2$: A8, A9, AA, AB : miss

101 : 10 0001 → $B_0 W_1$: 100, 101, 102, 103 : miss X

A8 : A 1000 → $B_2 W_0$: hit

103 : 10 0011 → $B_0 W_3$: Hit

B0 : B 0000 → $B_0 W_0$: B0, B1, B2, B3 : miss

$$\text{Hit Rate} = \frac{\text{تعداد برخورد}}{\text{تعداد کل مراجعات}} = \frac{2}{5} = 0/.40$$

مثال (۵): درصورتی که دسترسی به حافظه نهان(Cache) از طریق نگاشت مستقیم به روش مقابل باشد،مشخص کنید که حجم حافظه اصلی،حافظه نهان و محل استقرار آدرس $(000B\,3A\,4F)_H$ را بیابید.

21	7	4
Tag	Block	Word

Index = Block + word

حجم حافظه اصلی : $2^{(21+7+4)} = 2^{32} = 2^2 \times 2^{30} = 4\,GB$

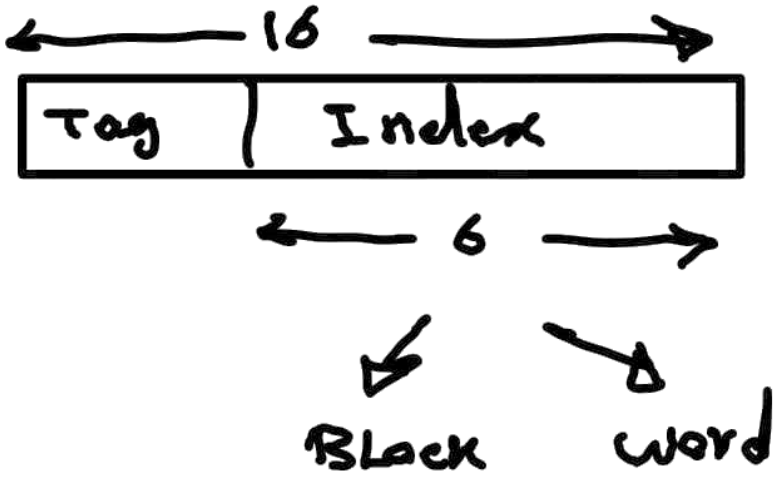
حجم کش : $2^7 \times 2^4 = 2^{11} = 2\,KB$

000B3 $\underbrace{1010\ 0100}_{B}\ \underbrace{1111}_{w}$: B_{36} w_{15}

مثال (۶): یک کامپیوتر دارای واحد حافظه با ابعاد ۶۴ کیلوکلمه که یک کیلو برابر بلوک های کش بلوک دارد.در روش نگاشت مستقیم تعداد بیت های فیلدهای **Tag** و **Index** را بیابید.

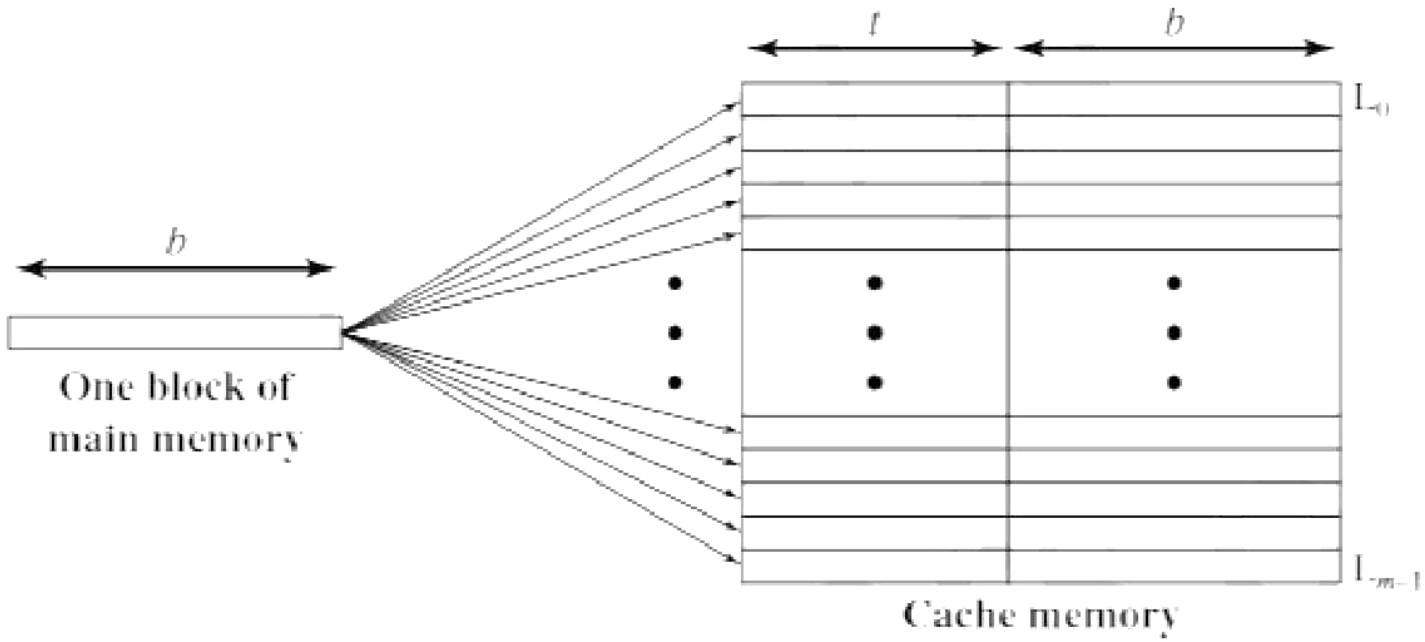
حجم حافظه اصلی : $2^6 \times 1024 = 2^{16}$

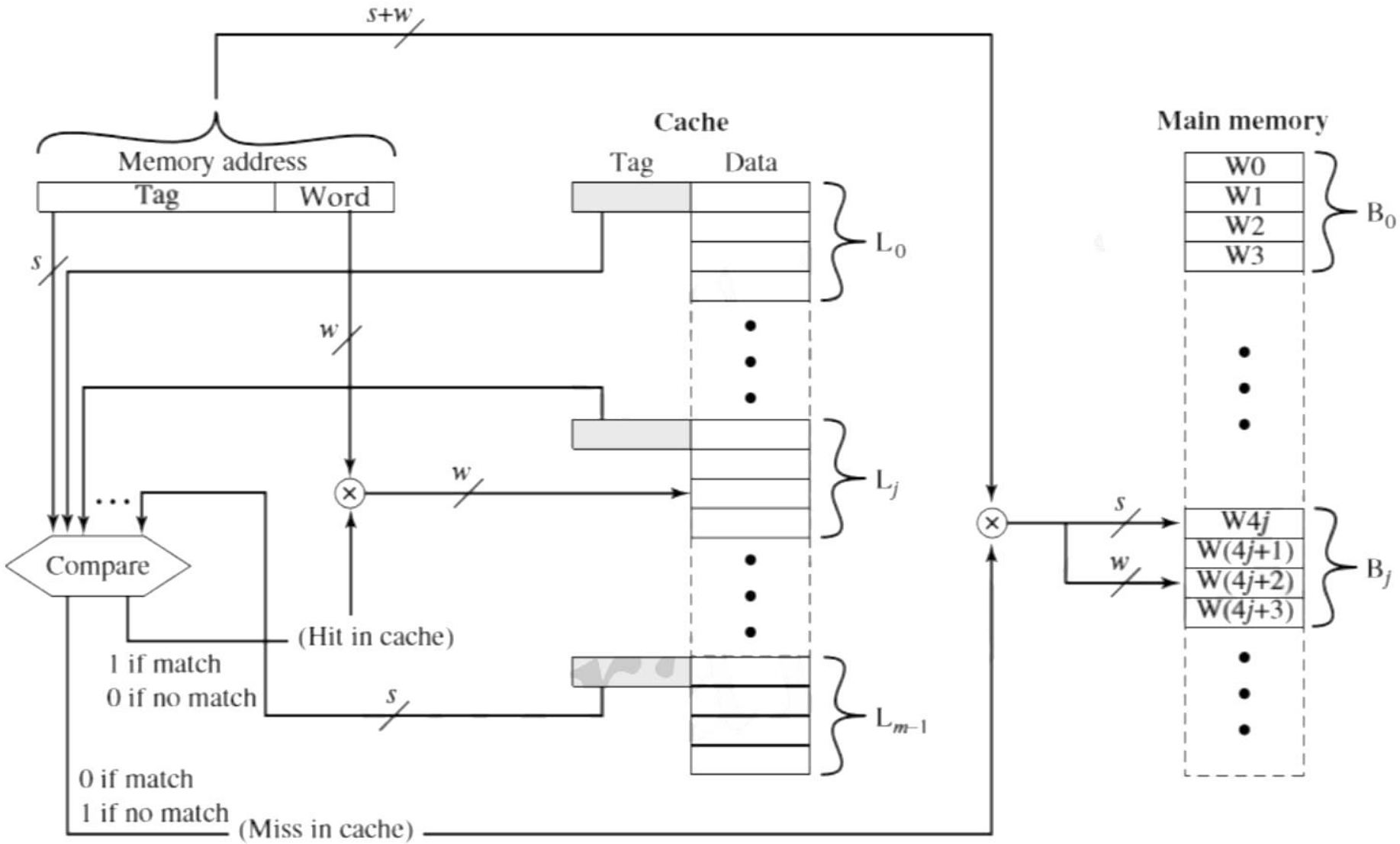
حجم حافظه کش : $2^6 = 64 \rightarrow$ Index = 6 bit



$\rightarrow$ Tag = 10 bit

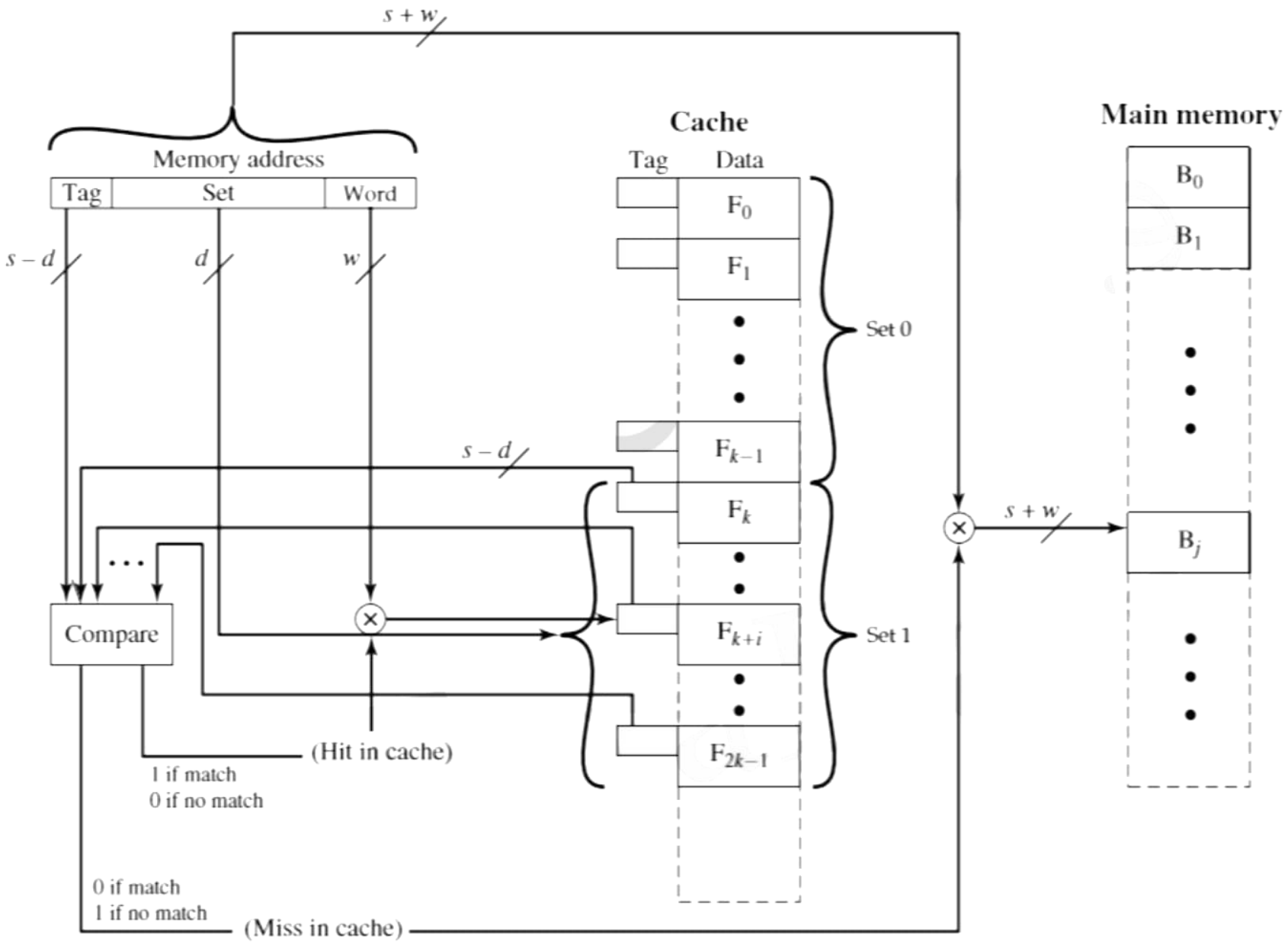
نگاشت تداعیگر(انجمنی) (Associative Mapping)




s+w
Memory address
Tag
Word
s
w
Cache
Tag
Data
L0
Lj
Lm−1
Main memory
W0
W1
W2
W3
B0
W4j
W(4j+1)
W(4j+2)
W(4j+3)
Bj
Compare
(Hit in cache)
1 if match
0 if no match
0 if match
1 if no match
(Miss in cache)

نگاشت تداعیگر مجموعه ای(Set Associative Mapping)

- این نگاشت ترکیبی از نگاشت های انجمنی و مستقیم است.در واقع کش به مجموعه هایی مشتمل بر چندین بلوک که Tag خاص خود را دارند تقسیم می شود.وتمامی بلوک های یک مجموعه دارای آدرس یکسانی هستند.وهربلوک در حافظه اصلی به یک مجموعه یکتا در حافظه کش نگاشته می شود.و بلوک را میتوان در هر عنصر از مجموعه مورد نظرش قرار داد.


s + w
Memory address
Tag
Set
Word
s – d
d
w
Cache
Tag
Data
F0
F1
Fk–1
Fk
Fk+i
F2k–1
Set 0
Set 1
s – d
Compare
(Hit in cache)
1 if match
0 if no match
0 if match
1 if no match
(Miss in cache)
s + w
Main memory
B0
B1
Bj

(بعنوان مثال)

یک حافظه اصلی با ۲۰۴۸ کلمه داریم.وحافظه کش 2-way set associative داریم که دارای ۶۴ واحد آدرس پذیر است.تعداد بیت های Word،Set، Tag بصورت زیر است:

Tag	Set (6)	Word (1)

$64 = 2^6$

$2048 = 2^{11}$

Tag+Set+Word=11

Set=6

Word=1

Tag=11-6-1=4

مثال (۷):یک حافظه نهان با نگاشت مجموعه انجمنی ۱۶ راهه و قالب آدرس ارسالی از CPU به شکل زیر موجود است.تعداد مجموعه ها و بلوک های موجود در حافظه نهان و نیز حجم حافظه نهان در صورتیکه هر کلمه۸بیت باشد،را بیابید.

31 – 12	11 – 4	3 – 0
Tag	Set	Word

$set = 8 \rightarrow 2^{set} = 256$ مجموعه

$256 \times 16 = 2^8 \times 2^4 = 2^{12}$ تعداد بلوک ها

$Tag = 20$

$Word = 4 \rightarrow 2^4 = 16$ کلمه در هر بلوک

Tag	Data	...

$16 \times B$

داده های هر بلوک $= 20 + 16 \times 8$

set داده های هر $= 16(20 + 16 \times 8)$

cache کل داده $= 256\left[16(20 + 16 \times 8)\right]$

مثال (۸):در صورت اجرای قطعه کد زیر در یک کامپیوتر با کش ۱۲۸ بایتی ۴ راهه،اگر هربلوک این کش ۴ کلمه ۸ بیتی باشدبا فرض خالی بودن کش . زمان دسترسی ۹۹نانو ثانیه برای حافظه اصلی و زمان دسترسی ۱۰ نانو ثانیه برای کش،زمان موثر دسترسی به آدرس ها را بیابید.

هر داده Type ۱ بایت حافظه اشغال می کند و الگوریتم جایگزینی FIFO است.همچنین آرایه بصورت Row-major ذخیره شده است.

```
A:array[0..2,0..2]  of Type
For I=0 to 2
  For J=0 to 2
    A[J,I]=0
```

هربلوک : 4 Byte

set هر : 16 Byte

$$\text{set تعداد} = \frac{128}{16} = \frac{2^7}{2^4} = 2^3$$

$\rightarrow$ set = 3

word = 2

0	1	2
3	4	5
6	7	8

— 00006 : S_0, W_0 : 0,1,2,3 : miss
00011 : S_0, W_3 : Hit
00110 : S_1, W_2 : 4,5,6,7 : miss
00001 : S_0, W_1 : hit
00100 : S_1, W_0 : Hit
00111 : S_1, W_3 : Hit
00010 : S_0, W_2 : Hit
00101 : S_1, W_1 : Hit
01000 : S_2, W_0 : miss

$$\text{Hit Rate} = \frac{6}{9} = \frac{2}{3}$$

$$t_a = t_c + (1-h)t_m = 10 + \frac{1}{3}(99) = 43\,ns$$

مثال (۹): سه نوع پیکربندی برای حافظه نهان ۶۴ کیلوبایتی با مشخصات زیر طراحی شده است. این حافظه نهان قرار است که به یک پردازنده ۸ بیتی با گذرگاه آدرس ۳۲ بیتی متصل شود.تعداد کل بیت های Tag استفاده شده در سه نوع حافظه نهان را بیابید.

Type (C)	Type(B)	Type(A)
Fully associative	8-way set associative	Direct-mapped
16-byte block size	32-byte block size	64-byte block size
Hit ratio:99.8%	Hit Ratio:99.5%	Hit Ratio:98%

Type (A) - نگاشت مستقیم :

Tag	Block	word

تعداد بلوک : $\frac{64 \times 1024}{64} = 1024 \longrightarrow$ BLOCK = 10 bit

8 bit ⟶ 1 byte ⟶ در هر بلوک 64 کلمه قرار می‌گیرد.

word = 6

Tag = 32 − 6 − 10 = 16 ⟹ $2^{10} \times 16 = 2^{14}$ bit

Taype (B) : 8 set

Tag	Set	word

تعداد بلوک‌ها : $\frac{64 \times 2^{10}}{32} = 2^{11}$

تعداد مجموعه‌ها : $\frac{2^{11}}{2^{3}} = 2^{8}$ ⟶ set = 8 ⟶ Tag = 32 − 13 = 19

تعداد کلمات در هر بلوک : $\frac{32 \text{ byte}}{1 \text{ byte}} = 32$ ⟶ word = 5

Tag کلی ⟹ $2^{11} \times 19$

Tag	ward

تعداد بلوک‌ها : $\frac{64 \times 2^{10}}{16} = \frac{2^{16}}{2^4} = 2^{12}$

کلمه : $\frac{16\,byte}{1\,byte} = 2^4 \longrightarrow$ ward $= 4$

و Tag ها $\Longrightarrow 28 \times 2^{12}$

$Tag = 32 - 4 = 28$

مثال (۱۰): تعداد کلمات حافظه اصلی در یک سیستم به اندازه دو کیلو برابر حافظه نهان انجمنی k-way set است.اگر تعداد مجموعه های حافظه نهان ۱۲۸ باشد.در صورتیکه تعداد بیت های فیلد Tag برابر ۱۵ باشد،مقدار k رابیابید.

$128 = 2^7 \longrightarrow Set = 7$

اندازه حافظه اصلی = $2 \times 2^{10} \times$ اندازه حافظه کش

اندازه حافظه کش = تعداد مجموعه ها × تعداد بلوک ها × تعداد کلمات در هر بلوک

$r = \log K \rightarrow K = 2^r$

$\Longrightarrow$ حافظه اصلی $= 2^{11} \times 2^7 \times \underbrace{K}_{2^r} \times 2^w$

حافظه اصلی $= 2^{18+r+w}$

Tag	Set	Word

$$18 + r + \cancel{w} = \text{Tag} + \underset{8}{\text{set}} + \cancel{\text{word}}$$

$$\Longrightarrow 11 + r = \text{Tag}$$

$$\text{Tag} = 15 \longrightarrow 11 + r = 15 \longrightarrow r = 4 \;,\; K = 2^4 = 16$$

(روش های جایگزینی در حافظه کش)

سه روش برای جایگزینی بلوک در کش وجود دارد:

۱- LRU(Least Recently Used)

۲- FIFO(First In First Out)

۳- LFU(Least Frequently Used)

(روش های نوشتن در حافظه کش)

دو روش برای نوشتن در حافظه کش وجود دارد:

۱-رویه کامل نویسی(Write-through)

۲-روش پس نویسی (write-back)

رویه کامل نویسی(Write-through)

- در این روش هر نوشتن در کش در حافظه اصلی نیز اعمال می شود.
- در این حالت داده موجود در حافظه اصلی همواره معتبر است.
- عمده ترین مشکل این روش ایجاد ترافیک بر روی حافظه اصلی می باشد.

روش پس نویسی (write-back)

- در این روش تا زمانیکه داده در کش باشد، نوشتن ها فقط درکش اتفاق می افتد
- برای هر بلاک در کش یک بیت تحت **dirty bit** یا **used bit** وجود دارد.
- هنگام جایگزینی یک بلوک در کش با توجه به **dirty bit** درصورتیکه تغییر کرده باشد در حافظه اصلی نوشته می شود.
- مشکل این روش نامعتبر بودن بخشی از حافظه اصلی است که در کش تغییر کرده است.

حافظه مجازی (Virtual Memory)

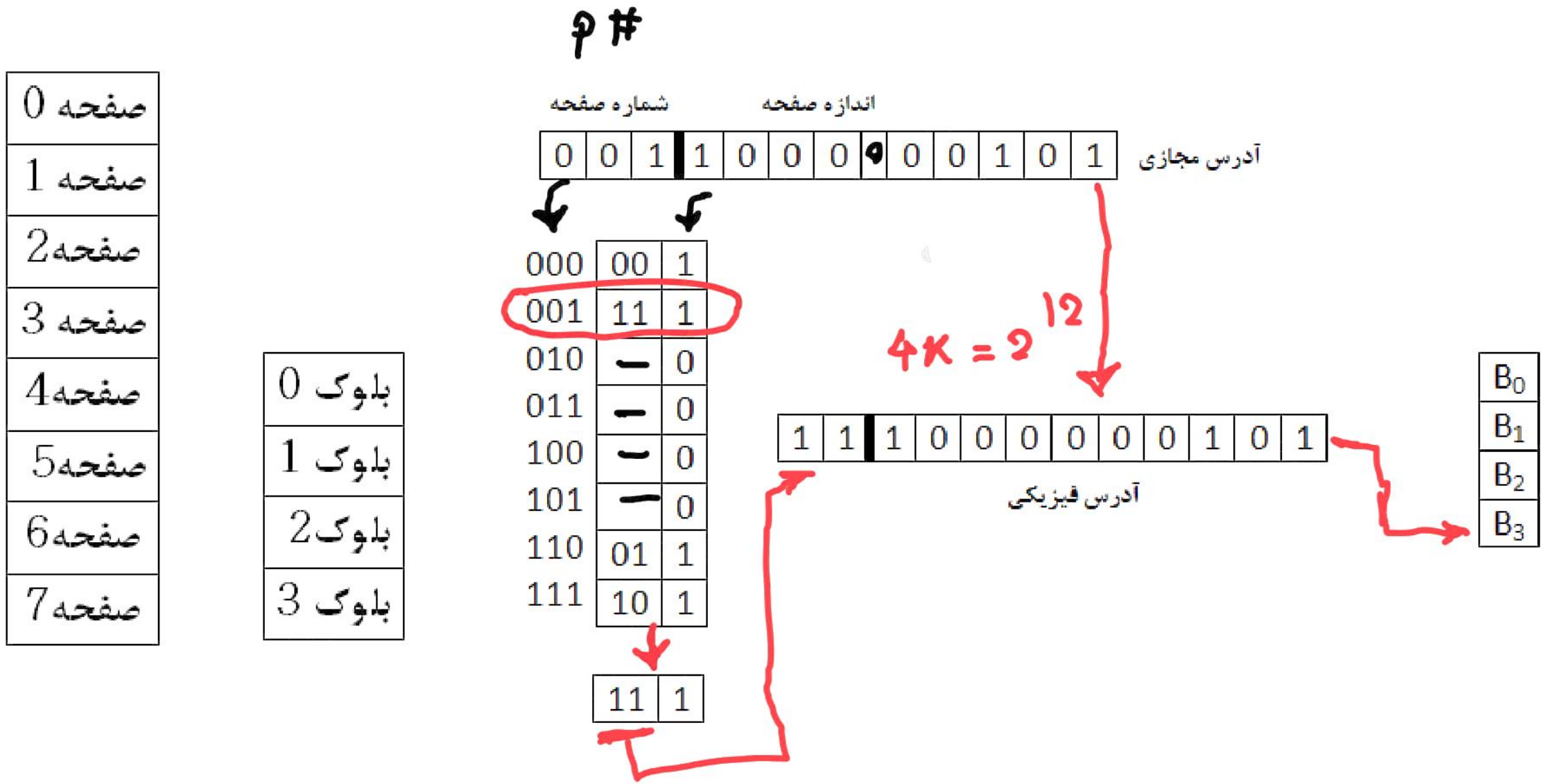
- هنگامیکه برنامه اجرایی حجم بیشتری نسبت به گنجایش حافظه اصلی داشته باشد آنگاه جای کافی برای بارگذاری تمامی برنامه وجود ندارد.لذا باید برنامه به بخش هایی تقسیم می شود تا بتواند اجرا شود برای این کار از تکنیک حافظه مجازی استفاده می شود.
- مهم ترین انگیزه برای ایجاد حافظه مجازی ،رفع محدودیت گنجایش حافظه اصلی و به اشتراک گذاری حافظه اصلی بین برنامه ها می باشد

آدرس مجازی(منطقی): آدرسی که توسط برنامه نویس مورداستفاده قرار می گیرد.

آدرس فیزیکی :آدرس در حافظه اصلی(آدرس واقعی) را آدرس فیزیکی گویند.

صفحه(Page): در مکانیسم صفحه بندی،حافظه مجازی مربوط به هر برنامه به تکه های کوچک با اندازه ثابت و یکسان تقسیم می شوند که به هر تکه،صفحه(Page) گویند.

بلوک(قاب صفحه): حافظه اصلی به تکه هایی برابر با اندازه صفحه تقسیم می شود که به هر تکه بلوک(قاب صفحه)می گویند.


p #
شماره صفحه
اندازه صفحه
0 0 1 1 0 0 0 0 0 0 1 0 1
آدرس مجازی
000 00 1
001 11 1
010 — 0
011 — 0
100 — 0
101 — 0
110 01 1
111 10 1
11 1
4K = 2^12
1 1 1 0 0 0 0 0 0 1 0 1
آدرس فیزیکی
B0
B1
B2
B3
صفحه 0
صفحه 1
صفحه 2
صفحه 3
صفحه 4
صفحه 5
صفحه 6
صفحه 7
بلوک 0
بلوک 1
بلوک 2
بلوک 3

استفاده از حافظه انجمنی

P#

0	0	1	1	0	0	0		0	0	1	0	1

11100

P#	B#
000	00
001	11
110	01
111	10

001	11

آدرس فیزیکی

1	1	1	0	0	0	0	0	0	1	0	1

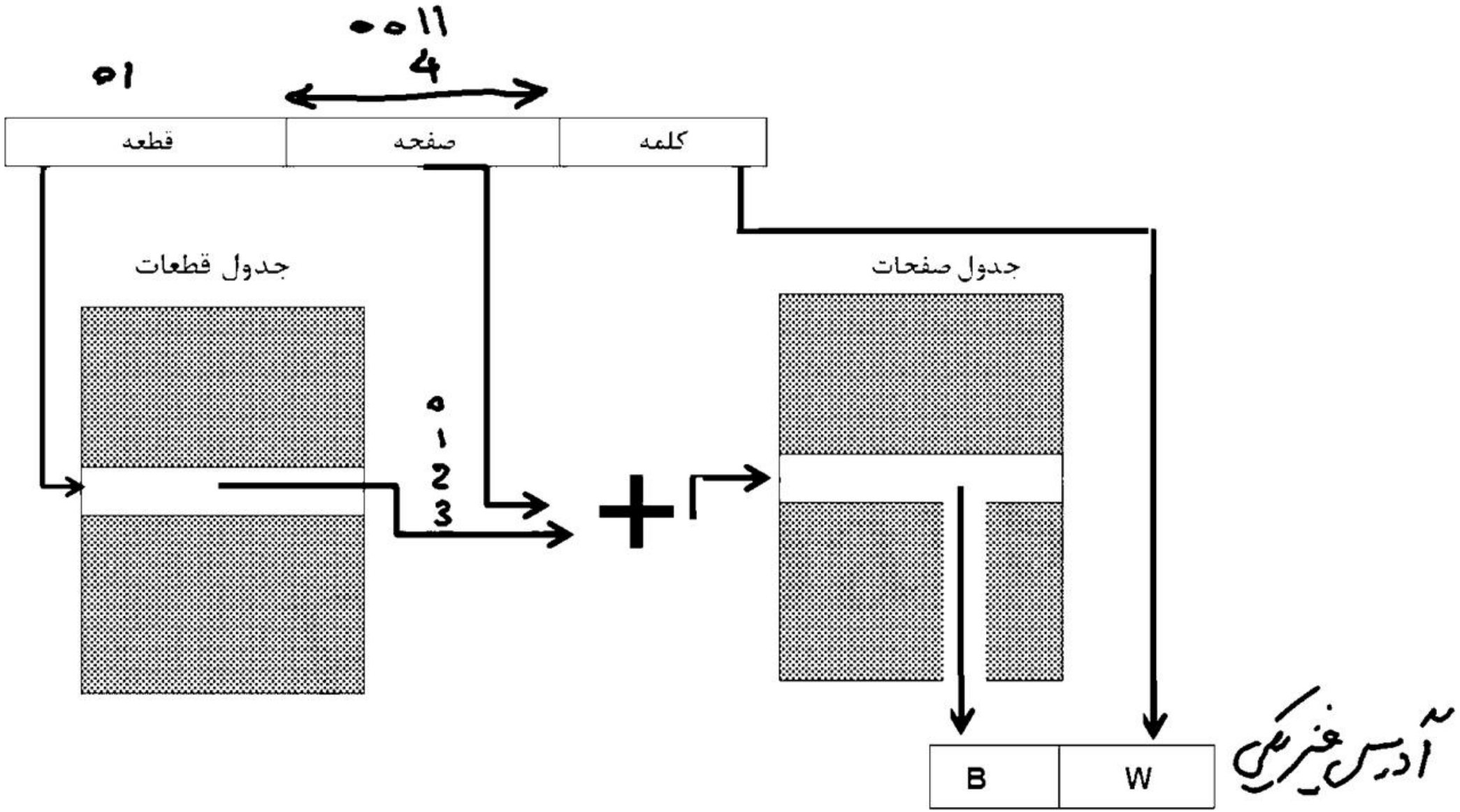
مثال (۱۱): در یک سیستم حافظه مجازی،فضای آدرس دهی ۲۴ بیت و فضای حافظه در دسترس توسط ۱۹ بیت مشخص می شود در این سیستم هر صفحه ۸کیلوکلمه است.تعداد صفحه و تعداد بلوک و اندازه هرصفحه چندبیت است؟

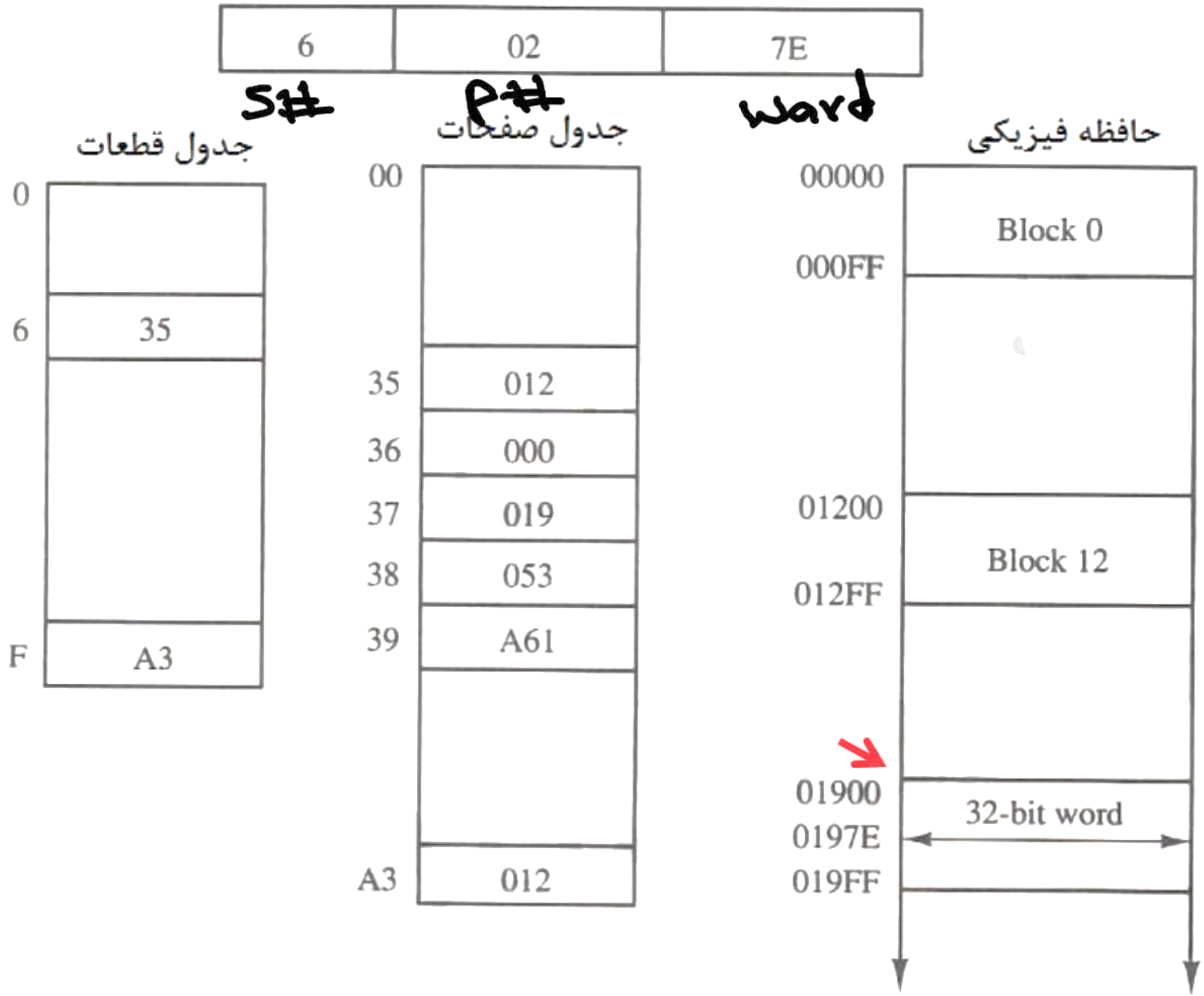
تعداد صفحه : $\frac{2^{24}}{2^3 \times 2^{10}} = 2^{11} = 2K$ page

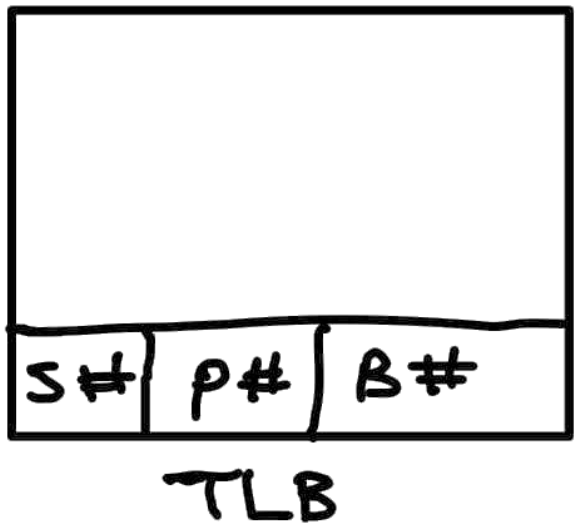
تعداد بلوک : $\frac{2^{19}}{2^{13}} = 2^6 = 64$ BLOCK

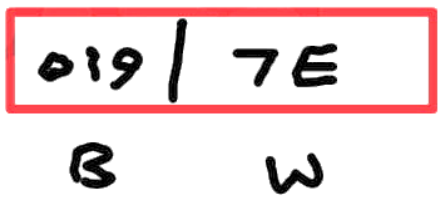
اندازه جدول صفحه Ram : $(6+1) \times 2^{11} = 7 \times 2^{11}$

اندازه جدول صفحه (انجمنی) : $(6+11) \times 2^6 = 17 \times 2^6$




6
02
7E
S#
P#
word
جدول قطعات
0
6
35
F
A3
جدول صفحات
00
35
012
36
000
37
019
38
053
39
A61
A3
012
حافظه فیزیکی
00000
Block 0
000FF
01200
Block 12
012FF
01900
32-bit word
0197E
019FF


S#
P#
B#
TLB


019
7E
B
W

مثال (۱۲): یک سیستم کامپیوتری دارای حافظه مجازی به صورت قطعه-صفحه می باشد.اگر تعداد قطعات ۱۶ و حافظه اصلی یک کیلو بلاک ۵۱۲کلمه ای داشته باشد.درصورتیکه حافظه مجازی دارای ۱۰۲۴ کیلو کلمه باشد.درصورت استفاده از حافظه انجمنی برای ترجمه آدرس و استفاده از یک بیت اعتبار،تعداد بیت های هر سطر از این حافظه انجمنی را محاسبه کنید.

$16 = 2^4 \longrightarrow S\# = 4$

$1024 = 2^{10} \longrightarrow B\# = 10$

S#	P#	word

آدرس حافظه مجازی : 20

$\longrightarrow S\# + P\# + W = 20$

$\Rightarrow P\# + W = 16 \Longrightarrow P\# = 7$

$1024 \times 1024 = 2^{10} \times 2^{10} = 2^{20}$

$512 = 2^9 \longrightarrow W = 9$

V	S#	P#	B#
1	4	7	10

$\rightarrow 10+7+4+1=22$