



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

بسمه تعالی

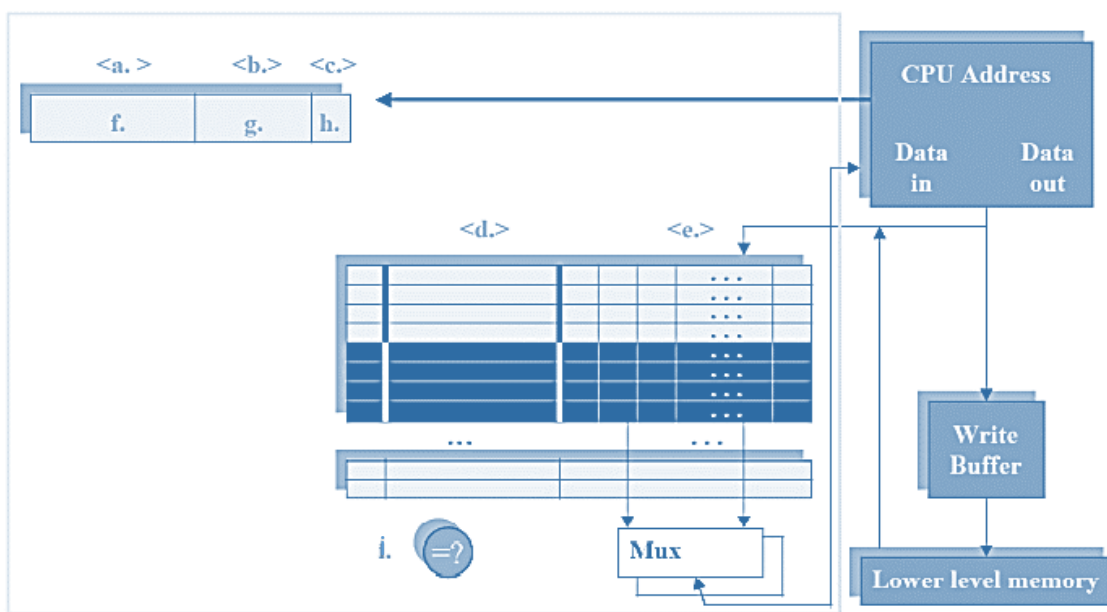
پاسخ تمرین سوم درس معماری کامپیوتر

نیم سال اول ۰۱-۰۰



دانشکده مهندسی کامپیوتر

۱. شکل زیر ساختار کلی یک حافظه‌ی نهان می‌باشد. با توجه به علامت‌های مشخص شده در شکل برای هر یک از علائم، پاسخ خواسته شده را بنویسید.
- هر کلمه ۳۲ بیت است.
 - هر بلاک حافظه‌ی نهان می‌تواند ۲۰۴۸ بیت داده ذخیره کند.
 - تعداد ۲۰۴۸ بلاک در حافظه‌ی نهان موجود است.
 - آدرس حافظه‌ی اصلی ۳۲ بیتی است که به هر کلمه اشاره می‌کند.



الف) هر کدام از a, b, c, d, e چه اندازه‌ای دارند؟

- قسمت c: هر کلمه ۲^۵ بیت است. هر بلاک ۲^{۱۱} بیت است. بنابراین هر بلاک ۲^۶ کلمه خواهد بود. در اینصورت آفست ۶ بیتی خواهد بود.
- قسمت b: تعداد ۲^{۱۱} بلاک در حافظه‌ی نهان وجود دارد در نتیجه ۱۱ بیت برای ایندکس نیاز داریم.
- قسمت a: باقی‌مانده‌ی بیت‌ها اندازه‌ی تگ می‌شود. از آنجا که طول آدرس حافظه‌ی اصلی ۳۲ بیتی است ۶ بیت آفست و ۱۱ بیت ایندکس را از آن کم می‌کنیم و تعداد بیت‌های تگ برابر خواهد بود با ۱۵.
- قسمت d: این بخش هم نشان‌دهنده‌ی تگ هر بلاک در حافظه‌ی نهان است پس اندازه‌ی آن مانند a برابر ۱۵ بیت خواهد بود.
- قسمت e: این قسمت نشان‌دهنده‌ی یک بلاک است. پس اندازه‌ی آن برابر تعداد بیت‌های هر بلاک یعنی ۲۰۴۸ بیت است.

ب) نام بخش‌هایی که h, g, f به آن اشاره می‌کنند چیست؟
قسمت f تگ، قسمت g ایندکس و قسمت h آفست است.

پ) سائز حافظه‌ی نهان (i) چقدر است؟

حافظه‌ی نهان دارای ۲۰۴۸ بلاک است که هر کدام ۲۰۴۸ بیتی یا ۲۵۶ بایتی هستند. در نتیجه

$$2048 \text{ blocks} \times 256 \frac{\text{bytes}}{\text{blocks}} = 2^{19} \text{bytes} = 0.5 \text{MB}$$

۲. برای نگهداری اطلاعات از دو حافظه‌ی اصلی و نهان استفاده می‌کنیم. هر کلمه‌ی حافظه‌ی اصلی ۱۶ بیت است و کل این حافظه‌ی اصلی ۵۱۲ بلاک است. هر بلاک نیز حجمی برابر با ۳۲ کلمه را دارد. حجم حافظه‌ی نهان نیز ۱۶ بلاک است و ساختار آن به صورت $2 - \text{way set associative}$ است. تعداد بیت‌های مورد نیاز برای موارد زیر را به دست آورید:

حافظه نهان ۱۶ بلاک و مجموع انجمنی دوتایی $\leftarrow 8 \text{ set}$ در این حافظه‌ی نهان داریم \leftarrow سه بیت آدرس برای $\text{set} \leftarrow 9-3=6$ بیت برای تگ نیاز است. در کل ۱۶ بلاک داریم پس ۱۶ تگ داریم که هریک ۶ بیت دارند.

همانطور که گفته شد برای مشخص کردن هر set به ۳ بیت احتیاج داریم.
هر بلاک ۳۲ کلمه دارد پس برای مشخص کردن هر کلمه به ۵ بیت نیاز داریم.

الف) برچسب (tag)

ب) مجموعه (set)

ج) کلمه (word)

امتیازی

۳. برنامه‌ای با هدف انجام محاسباتی در زمینه‌ی ماتریس‌ها نوشته شده‌است. در این برنامه، تابعی وجود دارد که تمام خانه‌های یک ماتریس دوبعدی را به اندازه‌ای دلخواه افزایش می‌دهد. مشخصات سیستمی که برنامه روی آن اجرا می‌شود به صورت زیر است:

- حافظه‌ی نهان ۳۲ کلمه‌ای با نگاشت مستقیم است. حافظه‌ی نهان دارای ۲ بلاک است.
- بلاک‌ها ۱۶ کلمه‌ای هستند.

- هر درایه از یک آرایه به اندازه‌ی ۴ کلمه حافظه اشغال می‌کند.

فرض کنید برنامه به ازای یک ماتریس دوبعدی ۲۰ در ۲۰ به هدف افزایش ۷ واحدی تمام درایه‌های آن اجرا شده است. نرخ موفقیت حافظه‌ی نهان را فقط برای تابع ذکر شده در دو حالت زیر محاسبه کنید.

الف) درایه‌های آرایه در این سیستم به صورت row major ذخیره می‌شود.

ب) درایه‌های آرایه در این سیستم به صورت column major ذخیره می‌شود.

راهنمایی:

نرخ موفقیت را به ازای قطعه کد زیر بررسی کنید.

```

for (i = 0; i < 20; i++)
    for (j = 0; j < 20; j++)
        A[i][j] = A[i][j] + 7;

```

فرض می‌کنیم آرایه‌ی دوبعدی ما به این صورت است.

A0,0	A0,1	A0,2	...	A0,19
A1,0	A1,1	A1,2	...	A1,19
A2,0	A2,1	A2,2	...	A2,19
...
A19,0	A19,1	A19,2	...	A19,19

ترتیب فراخوانی درایه‌ها طبق حلقه‌های `for` به ترتیب از چپ به راست به این شکل خواهد بود:

A0,0 – A0,0 – A0,1 – A0,1 – A0,2 – A0,2 – A0,3 – A0,3 – A0,19 – A0,19 – A1,0 – A1,0 – A1,1 – A1,1 – ... – A1,19 – A1,19 – ... – A19,19 – A19,19

زیرا هر خانه یک بار از حافظه خوانده می‌شود پس ابتدا در زمان `read` فراخوانی می‌شود. سپس در زمان `write` کردن مقدار جدید دوباره به آن درایه دسترسی پیدا می‌کنیم.

الف) Row Major

اگر بخواهیم خانه‌های این آرایه را به صورت `row major` پشت سر هم در نظر بگیریم، اعضای یک سطر پشت سر هم خواهند بود و سطرها نیز پشت هم قرار می‌گیرند. یعنی درایه‌های آرایه در حافظه به این شکل پشت هم قرار خواهند گرفت:

A0,0	A0,1	A0,2	A0,3	A0,4	A0,5	...	A0,19	A1,0	A1,1	A1,2	...	A1,19
------	------	------	------	------	------	-----	-------	------	------	------	-----	-------

A2,0	A2,1	...	A2,19	...	A19,0	A19,1	A19,2	...	A19,19
------	------	-----	-------	-----	-------	-------	-------	-----	--------

پس درایه‌های فراخوانی شده را متوجه شدیم. حال باید بلاک‌های فراخوانی شده را تشخیص دهیم. از آنجا که هر بلاک ۱۶ کلمه است و هر درایه هم ۴ کلمه است، نتیجه می‌گیریم که هر بلاک از ۴ درایه تشکیل شده است.

سطر صفر: B0, B0, B0, B0, B1, B1, B1, B1, ..., B4, B4, B4, B4:

سطر یک: B5, B5, B5, B5, B6, B6, B6, B6, ..., B9, B9, B9, B9:

...

سطر نوزده: B95, B95, B95, B95, B96, B96, B96, B96, ..., B99, B99, B99, B99:

بلاک در M.M	B0	B0	...	B0	B1	B1	...	B1	B2	...	B2	...	B99	B99	...	B99
----------------	----	----	-----	----	----	----	-----	----	----	-----	----	-----	-----	-----	-----	-----

بلاک در cache	C0	C0	...	C0	C1	C1	...	C1	C0	...	C0	...	C1	C1	...	C1
h/m	m	h	h	h	m	h	h	h	m	h	h	...	m	h	h	h

برای پیدا کردن نرخ برخورد کافیست بفهمیم که به ازای یک بار m چند بار h داریم. با خواندن اولین کلمه‌ی درایه‌ی $A0,0$ کل بلاک $B0$ وارد حافظه‌ی نهان خواهد شد. خود $A0,0$ دو بار فراخوانی می‌شود. در هر بار فراخوانی نیز به ۴ کلمه دسترسی پیدا می‌کنیم. بنابراین در طی کار با $A0,0$ هشت بار $B0$ فراخوانی می‌شود. درایه‌های $A0,1$ و $A0,2$ و $A0,3$ نیز هر کدام ۸ بار $B0$ را call می‌کنند. بنابراین کلاً $B0$ ، ۳۲ بار فراخوانی شد. سایر بلاک‌ها هم همینطور خواهند بود. پس

$$\text{Hit rate (row major)} = 31/32 \cong 96\%$$

الف) Column Major

این بار خانه‌های این آرایه به صورت column major پشت سر هم قرار گرفته‌اند؛ یعنی ترتیب قرار گرفتن درایه‌ها در حافظه به این شکل است:

A0,0	A1,0	A2,0	A3,0	A4,0	A5,0	...	A19,0	A0,1	A1,1	A2,1	...	A19,1
------	------	------	------	------	------	-----	-------	------	------	------	-----	-------

A0,2	A1,2	...	A19,2	...	A0,19	A1,19	A2,19	...	A19,19
------	------	-----	-------	-----	-------	-------	-------	-----	--------

در این صورت بلاک‌ها چنین خواهند بود:

سطر صفر: $B0, B5, B10, B15, B20, B25, \dots, B90, B95$

سطر یک: $B0, B5, B10, B15, B20, B25, \dots, B90, B95$

سطر دو: $B0, B5, B10, B15, B20, B25, \dots, B90, B95$

سطر سه: $B0, B5, B10, B15, B20, B25, \dots, B90, B95$

سطر چهار: $B1, B6, B11, B16, B21, B25, \dots, B91, B96$

سطر پنج: $B1, B6, B11, B16, B21, B25, \dots, B91, B96$

...

سطر نوزده: $B4, B9, B14, B19, B24, B29, \dots, B94, B99$

بلاک در M.M	B0	B0 ۷بار	B5	B5 ۷بار	B10	B10 ۷بار	B15	B15 ۷بار	B20	B20 ۷بار	B25	B25 ۷بار	...
بلاک در cache	C0	C0	C1	C1	C0	C0	C1	C1	C0	C0	C1	C1	...
h/m	m	h*7	m	h*7	m	h*7	m	h*7	m	h*7	m	h*7	...

همانطور که در جدول هم پیداست با فراخوانی درایه‌ی $A0,0$ ، ۸ بار بلاک $B0$ فراخوانی خواهد شد که یک بار آن miss می‌شود و ۷ بار دیگر hit خواهد بود. برای بلاک‌های بعدی هم روال به همین صورت خواهد بود. در این صورت

Hit rate (row major) = 7/8 \cong 87%

لطفا نکات زیر را در نظر بگیرید.

اشکالات خود را می‌توانید از طریق ایمیل autcafall2021@gmail.com بپرسید.
لینک کانال تلگرام درس <https://t.me/cafall2021> است. برای اطلاع از اخبار درس دنبال کنید.

موفق باشید