به نام خدا

دانشکده مهندسی کامپیوتر

درس معماری کامپیوتر
نیم‌سال دوم ۰۴-۰۳
استاد: دکتر اسدی

پاسخنامه تمرین سری هشتم

---

۱. (آ)

- Direct Mapped

تعداد بلوک‌های حافظه نهان به صورت زیر محاسبه می‌شود:

$$\text{Number of cache blocks} = \frac{\text{cache size}}{\text{block size}} = \frac{64 \times 1024}{64} = 1024 = 2^{10}$$

اندازه هر بخش آدرس به صورت زیر محاسبه می‌شود:

$$\text{block bits} = log_2(64) = 6 \text{ bits}$$

$$\text{index bits} = log_2(1024) = 10 \text{ bits}$$

$$\text{tag bits} = 24 - 10 - 6 = 8$$

- بیت‌های ۰ تا ۵: بیت‌های offset
- بیت‌های ۶ تا ۱۵: بیت‌های index
- بیت‌های ۱۶ تا ۲۳: بیت‌های tag

- 2-way set associative

تعداد مجموعه‌های حافظه نهان به صورت زیر محاسبه می‌شود:

$$\text{Number of sets} = \frac{1024}{2} = 512 = 2^9$$

اندازه هر بخش آدرس به صورت زیر محاسبه می‌شود:

$$\text{block bits} = log_2(64) = 6 \text{ bits}$$

$$\text{index bits} = log_2(512) = 9 \text{ bits}$$

$$\text{tag bits} = 24 - 9 - 6 = 9$$

- بیت‌های ۰ تا ۵: بیت‌های offset
- بیت‌های ۶ تا ۱۴: بیت‌های index
- بیت‌های ۱۵ تا ۲۳: بیت‌های tag

(ب)

$$1.5 + 0.08 \times (20 + 0.02 \times 100) = 3.26$$

$$(0.6 \times 1.5) + (0.4 \times 3.26) = 2.204$$

۲. (آ) ۵ بیت برای آفست داریم. یعنی در هر بلوک ۳۲ بایت داریم که معادل ۸ کلمه ۴ بایتی است.

(ب) ۵ بیت برای اندیس داریم. پس تعداد درایه‌های حافظه ۳۲ تا است.

(ج) در هر درایه ۲۲ بیت برای برچسب و یک بیت برای اعتبار داده استفاده شده و در مقابل ۸ ضربدر ۳۲ بیت داده داریم. پس درصدی که برای ذخیره داده استفاده شده حدودا ۹۲ درصد است.

(د) نرخ برخورد طبق جدول یک سوم است.

A	A/32	offset (A%32)	index (A/32)%32	tag (A/32)/32	hit/ miss	addresses in cache after access to A
0	0	0	0	0	miss	0-31
4	0	4	0	0	hit	0-31
16	0	16	0	0	hit	0-31
132	4	4	4	0	miss	0-31, 128-159
232	7	8	7	0	miss	0-31, 128-159, 224-255
160	5	0	5	0	miss	0-31, 128-159, 160-191, 224-255
1024	32	0	0	1	miss	1024-1056, 128-159, 160-191, 224-255
30	0	30	0	0	miss	0-31, 128-159, 160-191, 224-255
140	4	12	4	0	hit	0-31, 128-159, 160-191, 224-255
3100	96	28	0	3	miss	3072-3103, 128-159, 160-191, 224-255
180	5	20	5	0	hit	3072-3103, 128-159, 160-191, 224-255
2180	68	4	4	2	miss	3072-3103, 2176-2207, 160-191, 224-255

۳. هر بلوک ۱۶ کلمه ۲ بیتی دارد، لذا آفست هر بلوک برابر است با:

$$offset = log_2(16 * 2)$$

کش ما ۸ ست دارد لذا ایندکس کش $log_2(8) = 3$ بیت خواهد داشت.

سایر بیت‌های آدرس نیز تگ ما خواهد بود. در ادامه، تگ و ایندکس هرکدام از دسترسی‌ها به حافظه را محاسبه می‌کنیم و تعداد هیت و میس را محاسبه می‌کنیم. تنها نکته حائز اهمیت این است که به صورت یکی در میان، یک درخواست به حافظه برای خواندن اندیس و بار دیگر برای خواندن مقدار داخل آن اندیس خواهیم داشت.

آدرس خوانده شده از حافظه	تگ	اندیس کش	وضعیت برخورد
۰۰۰۰	۰۰	۰۰۰	Miss
۴A۳۲	۴A	۰۰۱	Miss
۰۰۰۲	۰۰	۰۰۰	Hit
B۸۳۰	B۸	۰۰۱	Miss
۰۰۰۴	۰۰	۰۰۰	Hit
B۸۳۲	B۸	۰۰۱	Hit
۰۰۰۶	۰۰	۰۰۰	Hit
B۸۳۴	B۸	۰۰۱	Hit
۰۰۰۸	۰۰	۰۰۰	Hit
B۸۲۲	B۸	۰۰۱	Hit
۰۰۰A	۰۰	۰۰۰	Hit
۴A۳۲	۴A	۰۰۱	Miss
۰۰۰C	۰۰	۰۰۰	Hit
۳۱۱A	۳۱	۰۰۰	Miss
۰۰۰E	۰۰	۰۰۰	Miss
۳۱۴۲	۳۱	۰۱۰	Miss

نرخ برخورد: $\frac{9}{16}$

۴. ۱. ۶ دسترسی به دستورالعمل + ۲ دسترسی به داده = ۸ دسترسی
۲. ۳ دسترسی به دستورالعمل + ۲ دسترسی به داده = ۵ دسترسی
۳. با فرض اینکه رشته مبدأ شامل ۳ کاراکتر و یک مقدار null در انتها است، تعداد تکرارها ۴ بار خواهد بود:

$$(3 \times 8) + 5 = 29$$

۴.
- **اولین تکرار:**
  - ۶ خطا برای دستورالعمل‌ها
  - ۱ خطا برای داده (بارگذاری مقدار از مبدأ)
  - از آنجایی که کش no write allocate است، مقدار مستقیماً در حافظه نوشته می‌شود و در کش ذخیره نمی‌شود.
  - load، اولین دستورالعمل را از کش خارج می‌کند.
  - store نیز به‌عنوان یک خطا در نظر گرفته می‌شود.

  مجموع: ۸ خطا
- **دومین تکرار:**
  - خطا در اولین دستورالعمل
  - برخورد در پنج دستورالعمل بعدی
  - خطا در load داده (چون load دستورالعمل قبلی، آن را از کش خارج کرده است)
  - خطا در نوشتن مقدار

  مجموع: ۳ خطا
- **تکرارهای سوم و چهارم:** مانند تکرار دوم، هر کدام ۳ خطا دارند.

$$8 + 3 + 3 + 3 = 17$$

$$\frac{17}{29} = 0.59$$

۵.
- هیچ تداخلی بین اولین دستورالعمل و آدرس رشته‌ی مبدأ وجود ندارد.
- با این حال، عملیات store همچنان به‌عنوان خطا در نظر گرفته می‌شود.

$$8 + 1 + 1 + 1 = 11$$

$$\frac{11}{29} = 0.38$$

۶.
- **اولین تکرار:**
  - ۶ خطا برای دستورالعمل‌ها
  - ۱ خطا برای load داده
  - داده‌ی ذخیره‌شده در کش نوشته می‌شود اما به حافظه اصلی ارسال نمی‌شود

  مجموع: ۸ خطا (نیاز به load بلوک داده‌ی ذخیره‌شده)
- **دومین تکرار:**

- بسته به سیاست جایگزینی، می‌تواند ۱، ۲ و یا ۳ خطا داشته باشد.

- **تکرارهای سوم و چهارم:**

- بسته به سیاست جایگزینی و حالت حافظه نهان در حلقه قبلی، می‌تواند ۱، ۲ و یا ۳ خطا داشته باشد.

به طور کلی به جواب‌هایی که هرکدام از حالات بالا را محاسبه کرده باشند، نمره تعلق گرفته است.

۵. (آ)
- index ۱۰ بیتی است پس یعنی درمجموع $2^{10} = 1024$ مجموعه وجود دارد و چون حافظه نهان ۲ راهه است، پس در مجموع تا بلاک دارد که یعنی در مجموع $1024 \times 2 = 2048$ بلاک وجود دارد.
- چون که اندازه بلاک ها ۳۲ بایت است، درنتیجه نیاز به یک offset به اندازه ۵ بیت داریم.
- برای بیت های tag یعنی بیت هایی که نه offset هستند و نه index پس $32 - 5 - 10 = 17$ بیت می باشند.

(ب) پاسخ این بخش در جدول زیر:

Address	Hit/Miss
110001	Miss
100111	Miss
001111	Miss
001100	Hit
010001	Miss
110010	Hit
100101	Hit
001110	Hit
100001	Miss
110001	Hit

(ب) قسمت پاسخ ۱ : Table

(ج) در ماشین خط لوله‌ای ایده‌آل با حافظه نهان کامل و بدون خطا، CPI برابر ۱ خواهد بود. برای محاسبه‌ی CPI واقعی، باید تعداد چرخه‌های توقف حافظه به ازای هر دستور را اضافه کنیم.

$$\text{CPI} = (\text{main CPI}) + (\text{load frac} + \text{store frac}) \times \text{miss rate} \times \text{miss penalty}$$

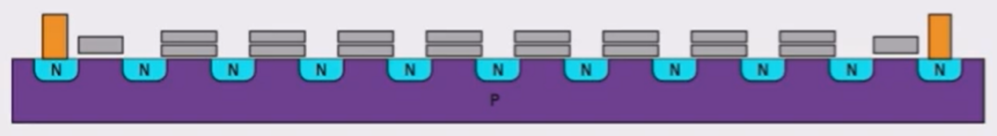
$$\text{CPI} = (1) + (0.25 + 0.15) \times 0.02 \times 100 = 1.8 cycles$$

۶. ابتدا به NAND Flash و NOR Flash می‌پردازیم. در این دو حافظه، از ترانزیستورهای دروازه شناور استفاده می‌شود. ساختار این ترانزیستورها به گونه‌ای است که می‌توان مرز[۱] تنظیم شده برای آن‌ها را تغییر داد. از این ویژگی به این صورت استفاده می‌شود که، یک ولتاژ تست که معمولا برابر میانگین ولتاژ بیشینه و کمینه است در نظر گرفته می‌شود. برای وقت‌هایی که می‌خواهیم ترانزیستور مقدار ۱ را در خود نگه دارد، مرز آن را به گونه‌ای تنظیم می‌کنیم که با اعمال این ولتاژ، وصل شود و در زمان‌های دیگر، برعکس این کار انجام می‌شود. حال با دانستن این ویژگی به توضیح ویژگی‌های منحصر به فرد این دو حافظه می‌پردازیم.

- **NAND Flash**

در این ساختار، تمامی گیت‌ها به صورت سریالی به هم متصل بوده و ساختار آن شباهت زیادی به ساختار الکتریکی گیت نند دارد. برای بررسی مقدار موجود در یکی از خانه‌های این نوع حافظه، به خانه‌هایی که نمی‌خواهیم بررسی کنیم بیشینه ولتاژ را اعمال کرده تا وصل باشند. برای خانه مورد بررسی، ولتاژ بررسی را اعمال می‌کنیم و به این صورت مقدار نهایی این بلاک نات مقدار موجود در خانه مورد نظر ما خواهد بود.

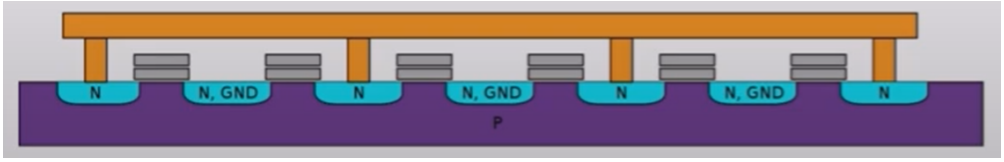
این حافظه به دلیل اینکه ترانزیستورها را به صورت خطی به یکدیگر متصل کرده، و هیچ بخش اضافی‌ای ندارد، عملا **متراکم‌ترین** ساختار ممکن را داراست.



شکل ۱: NAND Flash

- **NOR Flash**

در این ساختار، هر کدام از ترانزیستورها از زمین به خروجی متصل هستند. برای بررسی هرخانه، صرفا کافیست ولتاژ کمینه برای ترانزیستورهای دیگر اعمال شده و ولتاژ بررسی به ترانزیستور مدنظر اعمال شود تا مقدار نات آن در خروجی دیده شود.



شکل ۲: NOR Flash

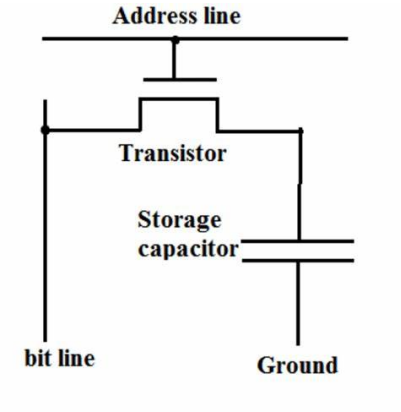
- **DRAM**

ساختار این حافظه به ازای هر بیت داده یک ترانزیستور و یک خازن دارد و معمولا به فرم مجموعه مستطیلی از داده ذخیره می‌شود. در خانه‌هایی که مقدار ۱ ذخیره شده، ترانزیستور شارژ و در سایر خانه‌ها ترانزیستور دشارژ است. برای خواندن بلوکی از داده، صرفا کافی است به خط مربوط به آن بلوک شارژ اعمال کنیم. با این کار، در هر بیت خروجی اگر خازن موجود در آن بیت در بلاک فعال شارژ داشته باشد، شاهد اختلاف ولتاژ با ولتاژ پایه هستیم و در سایر نقاط ولتاژ تفاوتی ندارد.

قابلیت دسترسی تصادفی و همچنین ساختار سلولی ساده، به همراه این امکان که چندین سلول به صورت همزمان خوانده شوند باعث شده است DRAM **سریع‌ترین** حافظه از بین این سه حافظه به شمار رود.

---

[۱] threshold


Address line
Transistor
Storage
capacitor
bit line
Ground

شکل ۳: DRAM