



به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخ نامه بنویسید.
- ۲- کل پاسخ تمرینات را در قالب یک فایل pdf با شماره دانشجویی خود نام گذاری کرده در سامانه CW بارگذاری کنید.
- ۳- این تمرین ۶۰ نمره دارد که معادل ۰,۶ نمره از نمره کلی درس است.
- ۴- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر کل نمره این تمرین را از دست خواهند داد.

۱- (۵ نمره) در یک پردازنده هر بار مراجعه به حافظه اصلی ۷۲ نانوثانیه طول می کشد. جدول زیر دو پیکربندی متفاوت از حافظه های نهان را نشان می دهد. زمان متوسط دسترسی به حافظه را در دو حالت زیر حساب کنید.

	hit rate	hit time
a	80%	0.6 ns
b	50%	4 ns

- الف- از یک سطح حافظه نهان با پیکربندی a استفاده می کنیم.
- ب- از دو سطح حافظه نهان استفاده می کنیم، پیکربندی a برای لایه اول (نزدیک تر به پردازنده) و پیکربندی b برای لایه دوم.

پاسخ:

$$AMAT_1 = 0.6 + 0.2 \times 72 = 15 \text{ ns}$$

$$AMAT_2 = 0.6 + 0.2 \times (4 + 0.5 \times 72) = 8.6 \text{ ns}$$

۲- (۱۵ نمره) توضیح دهید هر کدام از موارد زیر چه تاثیری در زمان برخورد (hit time)، نرخ فقدان (miss rate) و جریمه فقدان (miss penalty) خواهد داشت؟ (افزایش، کاهش یا بدون تغییر) پاسخ خود را توضیح دهید.

- الف- دو برابر کردن Associativity با ثابت نگه داشتن کل ظرفیت حافظه نهان و اندازه هر بلوک (خط)
- ب- نصف کردن اندازه هر بلوک (خط) با ثابت نگه داشتن Associativity و تعداد مجموعه ها
- ج- دو برابر کردن تعداد مجموعه ها با ثابت نگه داشتن کل ظرفیت حافظه نهان و اندازه هر بلوک (خط)

پاسخ:

الف- چون کل ظرفیت و اندازه هر بلوک ثابت است، تعداد مجموعه های حافظه نصف می شود. زمان برخورد زیاد می شود، چون هم برچسب (tag) بزرگتر می شود و هم بعد از پیدا کردن مجموعه مورد نظر باید یک مقایسه انجام دهد تا داده را پیدا کند. نرخ فقدان احتمالاً کمتر می شود چون تعداد فقدان های ناشی از تناقض (Conflict) را کم می کند. جریمه فقدان تغییر نمی کند.

ب- اگر اندازه بلوک نصف شود اما دو مورد دیگر ثابت بماند، یعنی اندازه حافظه نهان هم نصف شده است. زمان برخورد احتمالاً کم می شود، چون اندازه حافظه نصف شده و پیدا کردن یک خط نیاز به رمزگشایی کمتری دارد. البته tag هم یک بیت زیادتر می شود و تا اندازه ای زمان برخورد را زیاد می کند، اما نه خیلی. نرخ فقدان زیاد می شود باز چون اندازه حافظه نصف شده است، تعداد فقدان های ناشی از تناقض زیاد می شود و در عین حال چون اندازه بلوک هم نصف شده، فقدان های اجباری (Compulsary) هم بیشتر می شوند. جریمه فقدان کمتر می شود چون در هر بار ارجاع به حافظه تعداد بیت کمتری قرار است به حافظه نهان منتقل شود.

ج- وقتی تعداد مجموعه‌ها بیشتر شود، اما ظرفیت حافظه نهان و اندازه هر بلوک ثابت باشد، باید associativity را کم کرد. بنابراین زمان برخورد احتمالاً کمتر می‌شود چون بعد از پیدا کردن یک خط نیاز به مقایسه کمتری هست. از طرفی رمزگشایی بیشتری برای رسیدن به آن خط هست و tag کوچکتر می‌شود که این دو مورد اخیر تقریباً اثر هم را خنثی می‌کنند. نرخ فقدان افزایش می‌یابد چون associativity کم شده است و جریمه فقدان تغییر نمی‌کند.

۳- (۲۰ نمره) در طراحی یک حافظه نهان با نگاشت مستقیم (direct-mapped) و آدرس‌های ۳۲ بیتی، بیت‌های ۳۱ تا ۱۰ برای برچسب (tag)، ۹ تا ۵ برای اندیس (index) و ۴ تا ۰ برای آفست (offset) استفاده شده‌اند. با توجه به این ساختار به سوالات زیر پاسخ دهید:

- الف- اندازه هر بلوک (خط) چقدر است؟ (بر حسب کلمات ۳۲ بیتی)
- ب- تعداد درایه‌های حافظه نهان (cache entries) چند است؟
- ج- چه درصدی از کل بیت‌های حافظه نهان برای ذخیره داده استفاده شده است؟
- فرض کنید از زمانی که حافظه نهان کاملاً خالی است آدرس‌های زیر به ترتیب از راست به چپ، درخواست شده‌اند. (هر عدد آدرسِ بایتِ مورد درخواست است).

۰، ۴، ۱۶، ۱۳۲، ۲۳۲، ۱۶۰، ۱۰۲۴، ۳۰، ۱۴۰، ۳۱۰۰، ۱۸۰، ۲۱۸۰

به سوالات زیر پاسخ دهید:

- د- در هر مورد برچسب، اندیس، آفست و وقوع برخورد یا فقدان (miss/hit) را مشخص کنید.
- ه- نرخ برخورد چقدر است؟

پاسخ:

- الف- 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 است.
- ج- در هر درایه ۲۲ بیت برای برچسب و یک بیت برای اعتبار داده (valid bit) استفاده شده و در مقابل 8×32 بیت داده داریم. بنابراین درصدی که برای ذخیره داده استفاده شده این طور محاسبه می‌شود:

$$\frac{8 \times 32}{8 \times 32 + 23} \times 100 = \%92$$

- د- طبق جدول، نرخ برخورد ۴ از ۱۲ یعنی یک‌سوم 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

۴- (۲۰ نمره) در این سوال می‌خواهیم میزان SRAM مصرفی توسط حافظه نهان را محاسبه کنیم. فرض کنید کلمات داده و آدرس هر دو ۶۴ بیتی و حافظه‌های نهان بایت‌شمار (Byte-Addressable) هستند.

الف- تعداد بیت‌های مورد نیاز برای یک حافظه نهان ۳۲ کیلوبایتی و بلوک‌های دو کلمه‌ای با نگاشت مستقیم چقدر است؟

ب- تعداد بیت‌های مورد نیاز برای یک حافظه نهان ۶۴ کیلوبایتی و بلوک‌های ۱۶ کلمه‌ای با نگاشت مستقیم چقدر است؟ این مقدار چقدر بزرگتر از حافظه نهان بند الف است؟

ج- توضیح دهید چگونه حافظه نهان ۶۴ کیلوبایتی بند ب می‌تواند با وجود اندازه بزرگتری که نسبت به حافظه نهان ۳۲ کیلوبایتی بند الف دارد، کارایی بدتری داشته باشد.

د- توالی از درخواست حافظه تولید کنید که روی یک حافظه نهان ۳۲ کیلوبایتی انجمنی دو انتخابی (2-way set associative) به نسبت حافظه نهان بند الف نرخ فقدان (miss rate) کمتری داشته باشد.

پاسخ:

الف- حافظه نهان ۳۲ کیلوبایت و هر بلوک ۱۲۸ بیت (معادل ۱۶ بایت) است، پس تعداد بلوک‌ها عبارت است از:

$$noOfBlocks = \frac{32 \times 1024}{2 \times (64/8)} = 2048 \text{ blocks}$$

چون هر بلوک ۱۶ بایت است، پس آفست ۴ بیت دارد.

چون تعداد بلوک‌ها ۲۰۴۸ تا است، اندیس ۱۱ تایی است.

چون هر آدرس ۶۴ بیت دارد، پس برچسب ۴۹ بیت دارد.

$$noOfBitsPerBlock = (1 + 49) + (2 \times 64) = 178 \text{ bits}$$

$$TotalBits = 178 \times 2048 = \mathbf{364,544 \text{ bits}}$$

ب-

$$noOfBlocks = \frac{64 \times 1024}{16 \times (64/8)} = 512 \text{ blocks}$$

چون هر بلوک ۱۲۸ بایت است، پس آفست ۷ بیت دارد.

چون تعداد بلوک‌ها ۵۱۲ تا است، اندیس ۹ تایی است.

چون هر آدرس ۶۴ بیت دارد، پس برچسب ۴۸ بیت دارد.

$$noOfBitsPerBlock = (1 + 48) + (128 \times 8) = 1073 \text{ bits}$$

$$TotalBits = 1073 \times 512 = \mathbf{549,376 \text{ bits}}$$

تعداد بیت موردنیاز این ساختار ۱۸۴۸۳۲ بیت بیشتر از بیت موردنیاز ساختار الف است.

ج- حافظه بند ب می‌تواند نسبت به حافظه بند الف کارایی کمتری داشته باشد، چون اندازه بلوک‌های آن بزرگتر است و همین جریمه فقدان را بیشتر می‌کند. از طرفی ممکن است هر بار داده‌هایی وارد حافظه نهان شوند که لزوماً بعداً مورد استفاده قرار نمی‌گیرند.

د- هر توالی که به صورت متناوب نیاز به دسترسی به آدرس‌هایی داشته باشد که اندیس یکسان و برچسب متفاوت دارند و به همین دلیل در ساختار الف مرتب جایگزین یکدیگر می‌شوند. همین الگوی دسترسی در حافظه انجمنی دو انتخابی مشکلی ایجاد نمی‌کند.