12-3) الف) کامپیوتر آدرس 32 بیتی و حافظه‌ی نهان با نگاشت مستقیم دارد.
از آنجایی که هر خط آدرس 32 بیت است، 5 بیت برای آدرس دهی هر بایت استفاده می‌شود.
از آنجایی که حافظه نهان 1K بایت حافظه دارد داریم:

تعداد خطوط حافظه نهان ← $\frac{1024}{32} = 32 = 2^5$

⇒ حافظه نهان به 5 بیت برای مشخص کردن همه مجموعه نیاز دارد. 5 بیت ایندکس دارد

ب) 22 بیت برای تگ آدرس نیاز است. ← $Tag = 32 - 5 - 5 = 22$

ج) کل حافظه‌ی cache ← $32 \times (32 \times 8 + 22 +) = 8928$

12-4) الف) حافظه cache = $1MB = 1024 \times 1024 = 2^{20}$
چون آدرس دهی 32 بیتی است پس 32 بیت برای آدرس نیاز است.
سایز هر بلوک = تعداد کلمات در هر خط × تعداد بیت هر کلمه = $4 \times 4 = 16$ بایت
offset field = $lg(\text{block size}) = lg(16) = 4$ bit ← 4 بیت برای مشخص کردن بیت بلوک نیاز است
چون cache، از نوع نگاشت انجمنی مجموعه‌ای است و 2-way است پس هر مجموعه 2 بلوک دارد.
سایز هر مجموعه ← $2 \times (\text{block size}) = 32$ bits

تعداد مجموعه‌ها = $\frac{\text{ظرفیت cache}}{\text{سایز مجموعه}} = \frac{2^{20}}{32} = 2^{15}$ ⇒ 15 بیت برای مشخص کردن index ها نیاز است

www.nahal-alma.com
Scanned with CamScanner

تعداد بیت‌های Tag = تعداد بیت‌های آدرس − تعداد بیت‌های index − تعداد بیت‌های offset

⇒ تعداد بیت‌های Tag = 32 − 15 − 4 = 13 bits

index field = 15 bits

Tag field = 13 bits

ب) آدرس های داده شده در مبنای 16 را به باینری تبدیل کرده و با اضافه کردن 0 به سمت چپ آن ها به 32 بیت تبدیل کنید.

چون آدرس 13 بیت برای tag، 15 بیت برای index، 4 بیت برای offset دارد آدرس را بر حسب این تعداد بیت‌ها جدا کرده و index را جدا کرده و به مبنای 16 تبدیل کنید.

Physical Ad	Binary code	Index	Index Hexadecimal
00F8C00F	00000000111110001100000000001111	00011000000000	C00
9214AC89	00100100101011001010110010001001	10010101100 1000	4AC8
7142CF0F	00011100010100001010011110000111	01011001111 0000	2CF0
2BD4CF0C	00001010111110101001100111100001100	10011001111 0000	4CF0
F83ACF04	00111110000011101011001111000001000	01011001111 0000	2CF0

www.nahal-alma.com

Scanned with CamScanner

ج) از جدول صفحه‌ی قبل می‌توان نتیجه گرفت که داده‌های ورودی سوال را نمی‌توان همزمان در cache نگهداری کرد چون index هر آدرس 7142CF0F و F83ACF09 یکسان است و برابر 2CF0 است.

پس زمانی که داده مربوط به آدرس F83ACF09 وارد cache شود، آدرس 7142CF0F از cache پاک می‌شود.

5-12) الف) مزایای استفاده از حافظه نهان جداگانه برای داده‌ها و دستورالعمل‌ها:

1) برنامه اطلاعات آن‌ها را ویرایش نمی‌کند.
2) حافظه نهان اطلاعات ممکن است به‌عنوان دستگاه RO هندل شود.
3) حافظه نهان اطلاعات می‌تواند مستقیماً یک داده را از درون cache پاک کند و آن را درون RAM بنویسد.
4) از پیچیدگی و تنش بین بلوک‌های دستورالعمل و داده‌ها جلوگیری می‌شود.

معایب استفاده از حافظه نهان جداگانه برای داده‌ها و دستورالعمل‌ها:

1) نوشتن برنامه‌های self-modifying سخت می‌کند.
2) وقتی یک برنامه اطلاعات خود را تولید می‌کند، این اطلاعات، این اطلاعات، اطلاعات خام در نظر گرفته می‌شوند و در cache اشتباه ذخیره می‌شوند.

مزایای استفاده از cache یکسان:

1) کامپیوتر عمومی‌تر خواهد بود چون در هر دو برنامه‌های دستورالعمل و داده کار کرد خواهد بود.
2) برنامه‌های self-modifying راحت‌تر اجرا می‌شوند.
3) کارایی بیشتر ایجاد می‌کند
4) نسبت برد (hit ratio) بالاتری دارد

www.nahal-alma.com

Scanned with CamScanner

- مزایای استفاده از یک cache برای دستورالعمل ها و داده ها:

1) نسبت برد بالاتر همان است که گفته شده باشد.
2) استفاده از unified cache باعث می شود که یک دسترسی در لحظه داشته باشیم در حالی که استفاده از دو cache جداگانه در لحظه 2 دسترسی همزمان را پشتیبانی می کند.

ب) - مزایای استفاده از write-back cache:

1) تعداد دسترسی به حافظه اصلی را کاهش می دهد و دسترسی مستقیم به حافظه، پردازنده خروجی ورودی یا واحد پردازش مرکزی دیگر را در سیستم امکان پذیرتر می کند.

- معایب استفاده از write-back cache:

1) ورودی های اصلی حافظه متناظر با کلماتی که در حافظه نهان نوشته شده اند معتبر نیستند و این می تواند از نظر پردازنده ورودی / خروجی یا پردازنده های دیگر در سیستم مشکل ایجاد کند.
2) هر زمان که خطای خواندن رخ write-back cache به زمان بیشتری نیاز خواهد داشت.
3) زمانی اجرا می شود که dirty bit برابر با 1 باشد.

- مزایای استفاده از write-through cache:

1) سطح حافظه اساسی منسجم است که سیستم های چند پردازنده را ساده می کند.
2) بلوک فقط در سطح پایین اصلاح می شود نه در حافظه نهان
3) خطر از دست رفتن داده ها را به حداقل می رساند.

- معایب استفاده از write-through cache:

1) هر زمان که پهنای باند از اهمیت بیشتری برخوردار باشد از آن استفاده نمی شود.
2) هر زمان که حجم داده زیاد باشد این روش هیچ مزیتی ندارد.
3) کارایی روش write-back را ندارد.

www.nahal-alma.com

Scanned with CamScanner

(12-11

$T_e = T_c + (1-h) T_M$

T_c: زمان دسترسی به cache — T_M: زمان دسترسی به RAM — T_e: متوسط زمان دسترسی به حافظه

الف)

$T_e = 4 + (1-0.91) \times 40 = 7.6\ ns$

ب)

$T_e = 4 + (1-0.82) \times 40 = 11.2\ ns$

ج)

$T_e = 4 + (1-0.96) \times 40 = 5.6\ ns$

(10-9 واکشی دستورالعمل: stage 1 ، اجرا: stage 2 ، پس‌نویسی: stage 3

stage 1 تأخیر: $(0.5 + 0.1)\ ns = 0.6\ ns$

stage 2 تأخیر: $(0.1 + 0.7)\ ns = 0.8\ ns$

stage 3 تأخیر: $(0.1 + 0.1 + 0.5)\ ns = 0.7\ ns$

$\Rightarrow max = 0.8$ ← بیشترین تأخیر = کمترین زمان کلاک

الف) بیشترین فرکانس کلاک = 1 / کمترین زمان کلاک $= \frac{1}{0.8} = 1.25\ GHz$

ب) زمان تأخیر = $(0.8 \times 3) = 2.4\ ns$

ج) سرعت اجرای دستورالعمل در هر ثانیه = 1.25 billion ← دستورالعمل در هر ثانیه

www.nahal-alma.com

Scanned with CamScanner