

## مقدمه

درس معماری کامپیوتر یکی از درس‌هایی است که داوطلبان معمولا با آن مشکل دارند. دلیل این امر در این است که معمولا داوطلب به این درس مانند درس مدار منطقی می‌نگرد و آنرا در قالب یک درس که باید در آن با استفاده از روش‌ها و راه حل‌های مشخص یکسری مسائل را حل کند، می‌نگرد. در صورتی که در کنکور معمولا سوالات این درس یک پیش‌زمینه مفهومی از موضوعات را نیاز دارند. یعنی شما علاوه بر تسلط بر فرمول‌ها و راه حل‌های استاندارد، باید به نحوی در صورت مسئله تغییراتی ایجاد کنید تا آماده استفاده از فرمول‌ها و روابط معمول شوید. نکته مهم دیگری که اغلب داوطلبان با آن مشکل دارند، اینست که معمولا این درس را ادامه درس مدار منطقی دانسته و مطالب درس مدار منطقی را پیش نیاز این مطالب می‌دانند. در صورتی که اینچنین نیست. این مسئله برای دانشجویان کارشناسی صحت دارد، زیرا باید با کلیات مدارها و انواع ابزارهای در دسترس آشنایی جزئی داشته باشند ولی در کنکور، که معمولا داوطلب با هر دو درس آشنا شده است، دیگر این مسئله اهمیت آنچنانی ندارد. در این درس پس از آشنایی با مفاهیم کلی، به بررسی واحد کنترل پرداخته‌ایم، پس از آن در مورد پردازش خط لوله‌ای و روشهای محاسباتی پردازنده نکاتی را ارائه نموده‌ایم. در نهایت دو بخش حافظه و ورودی/خروجی‌ها که از دیگر مباحث موجود این درس می‌باشد را بیان کرده‌ایم.

**برای انسانهای بزرگ بن‌بستی وجود ندارد. زیرا بر این باورند که: یا راهی خواهیم یافت و یا راهی خواهیم ساخت.**

با آرزوی موفقیت و پیروزی

محمد آیینی

در جدول ذیل دروس به سرفصلهای مهم آن طبقه بندی شده و مشخص شده است که در هر سال از هر مبحث چند تست سوال شده است و دانشجوی محترم می تواند زمان باقیمانده تا کنکور را با توجه به اهمیت مباحث مدیریت نماید.

رشته: مهندسی کامپیوتر      درس: معماری کامپیوتر								
ردیف	مبحث	۱۳۸۵	۱۳۸۶	۱۳۸۷	۱۳۸۸	۱۳۸۹	مجموع ۵ سال	نسبت از کل
		تعداد تست	تعداد تست	تعداد تست	تعداد تست	تعداد تست		
1	روند توسعه معماری کامپیوتر	1	0	0	0	1	2	8%
2	سازمان و طراحی کامپیوتر پایه	0	1	0	0	3	4	15%
3	روشهای مختلف نمایش اعداد	1	1	1	1	0	4	15%
4	مدارهای ریاضی و محاسباتی	1	1	0	1	0	3	12%
5	پایپ لاین و اندازه گیری کارایی و تسریع	1	0	1	1	1	4	15%
6	طراحی واحد کنترل	1	1	0	1	0	3	12%
7	حافظه	0	2	1	2	1	6	23%
8	دستگاههای ورودی خروجی	0	0	0	0	0	0	0%
جمع		5	6	3	6	6	26	100%

# فصل اول

## روند توسعه معماری کامپیوتر

### نکات اولیه در معماری کامپیوتر

✓ کلیه اطلاعات ورودی از طریق بردهای کامپیوتر به درون کامپیوتر انتقال می‌یابد، که این اطلاعات به سیگنال‌های اطلاعاتی که بر حسب صفر و یک می‌باشند تبدیل می‌گردد.

✓ سیگنال‌های اطلاعاتی با پشتیبانی Clock بوجود می‌آیند.

✓ برای نگهداری اطلاعات حافظه Ram از تعدادی سلول استفاده می‌نماید که هر کدام از این سلولها یک فلیپ فلاپ می‌باشد.

✓ در هر فلیپ فلاپ فقط مقادیر صفر و یک می‌تواند قرار گرفته باشد و واحد نگهداری اطلاعات در هر سلول یک بیت است.

✓ اطلاعات وارد شده از طریق حافظه Ram وارد Cache می‌گردد و برای پردازش به Cpu انتقال داده می‌شود، Cpu نیز پس از پردازش، آن را به Ram باز می‌گرداند تا به دستگاههای خروجی منتقل گردد.

BUS: کلیه مسیرهای ارتباطی میان قطعات داخلی کامپیوتر را Bus گویند که دارای مسیرهای دوطرفه می‌باشند.

Cache: حافظه‌ای است با سرعت و قیمت بالا که سرعت سیستم را افزایش می‌دهد، Cache حالت انتظار در Cpu را کاهش داده و به همین شیوه باعث بالا رفتن سرعت اجرای دستورات می‌گردد.

MIPS: معمول‌ترین معیار اندازه گیری کارایی یک CPU می‌باشد.

یادداشت:

.....

.....

.....

.....

## زبان انتقال ثبات (RTL)

یک سیستم دیجیتال مجموعه‌ای از Module های سخت افزاری متصل به هم است که کار خاصی را در زمینه پردازش انجام می‌دهند.

عملیاتی که روی داده‌های یک ثبات صورت می‌گیرد ریز عمل (Micro Operation) نامیده می‌شود.

ساختار سخت افزار داخلی یک کامپیوتر با مشخص شدن موارد زیر تعیین می‌شود:

- ۱- مجموعه ثبات‌های کامپیوتر و وظایف آنها
- ۲- رشته ریز عمل‌های انجام شده روی داده‌های ذخیره شده در ثبات‌ها صورت می‌گیرد.
- ۳- واحد کنترلی که موجب آغاز رشته ریز عمل‌ها می‌شود.

## انتقال ثبات

ثبات‌های کامپیوتر با حروف بزرگ الفبای انگلیسی برای نشان دادن کار ثبات مشخص می‌گردند، کلاً ثباتی که آدرس را برای یک واحد حافظه در نظر می‌گیرد، ثبات آدرس حافظه (Memory Address Register) نامیده می‌شود.

- ✓ برای طراحی Bus برای n رجیستر k بیتی با مالتی پلکسر نیاز به k مالتی پلکسر n به 1 است، و با بافر سه حالت نیاز به یک رمز گشا و  $kn$  با فرسه حالت است و اندازه رمزگشا باید  $\log_2 n \times n$  می‌باشد.
- ✓ اتصال خروجی Tri State Buffer ها مانند OR عمل می‌کند. و اگر دو خروجی همزمان 1 نشوند حالت  $\times$  پدید می‌آید.
- ✓ ریز عملیات‌ها شامل عملیات‌های انتقال، حسابی، منطقی و شیفت هستند.
- ✓ در a shl اگر بیت علامت تغییر یابد، خطای سر ریز (Overflow) رخ داده است که آنرا با علامت V نشان می‌دهیم یعنی  $V = R_{n-1} \oplus R_{n-2}$  البته لازم به ذکر است که سر ریز زمانی رخ می‌دهد که دو بیت سمت چپ مانند یکدیگر نباشند.

## سازمان و معماری کامپیوتر

معماری کامپیوتر به صفاتی از سیستم اشاره می‌کند که قابل رؤیت به وسیله برنامه ساز می‌باشد. به عبارت دیگر، صفاتی که بر اجرای منطقی برنامه اثر مستقیم دارد.

بعنوان مثال می‌توان به مجموعه دستورالعمل‌ها، ثبات‌ها، تعداد بیت‌های اختصاص یافته به نمایش داده‌ها و ... اشاره نمود. سازمان کامپیوتر به واحدهای عملیاتی و اتصال بین آنها اشاره می‌نماید که در واقع مشخصات معماری را تحقق می‌بخشد. از جمله صفات در ارتباط با سازمان کامپیوتر، جزئیات سخت افزاری است مانند: سیگنال کنترلی، میانجی (interface) بین کامپیوتر و ... .

## ثبات‌های کامپیوتر

دستورالعمل‌های کامپیوتر معمولاً در مکان‌هایی از حافظه به طور متوالی ذخیره شده و هر یک به نوبت اجرا می‌گردند. اعمال موجود در سازمان کامپیوتر که توسط واحد کنترل، کنترل و اجرا می‌گردند برای دستکاری داده‌ها و پردازش آنها نیازمند تعدادی ثبات می‌باشند که در زیر نام این ثبات‌ها و وظایفشان مشخص می‌گردد.

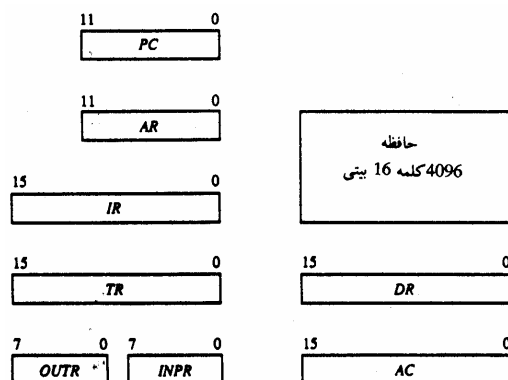
یادداشت:

.....

.....

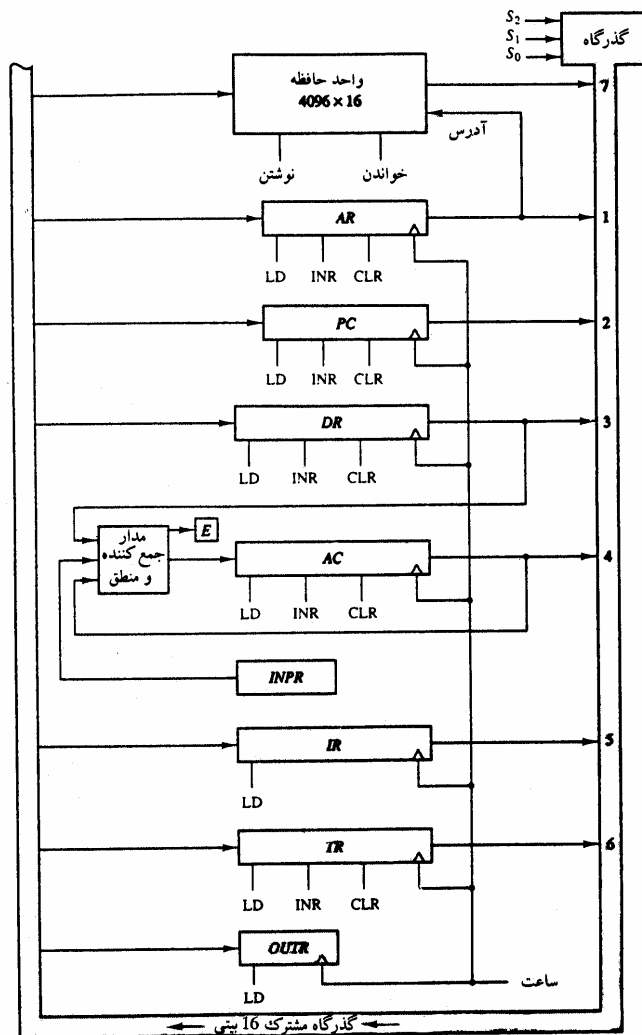
.....

.....



### سیستم گذرگاه مشترک

در اینجا ما یک کامپیوتر را معرفی می‌نماییم که دارای هشت ثبات، یک واحد حافظه و یک واحد کنترل می‌باشد، برای انتقال اطلاعات بین یک ثبات با ثبات دیگر یا حافظه و یا ... نیاز به یک Bus مشترک (گذرگاه مشترک) و مسیر ارتباطی می‌باشد.



یادداشت:

.....

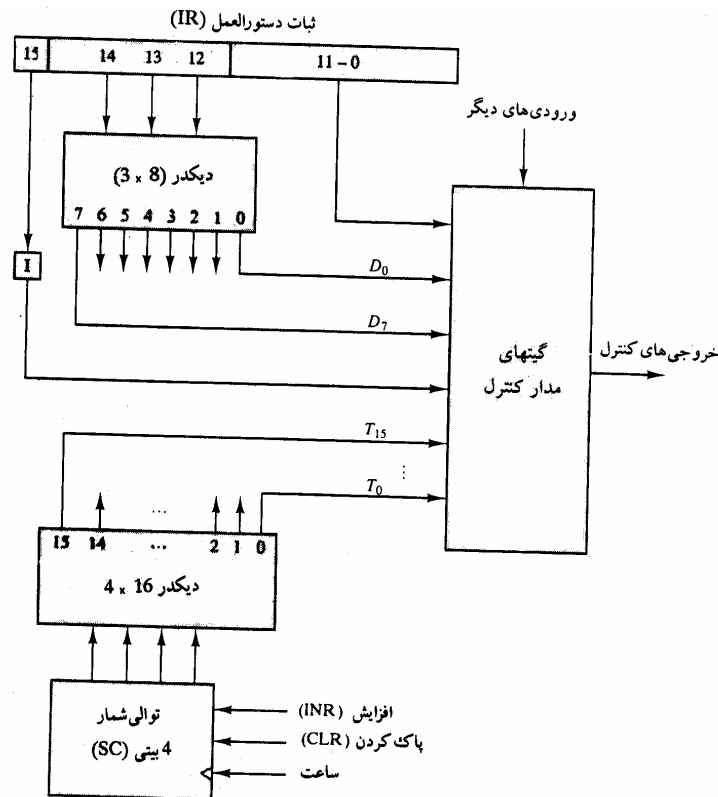
.....

.....

.....

دو نوع ساختار کنترلی عمده وجود دارند:

- ۱- کنترل سخت افزاری (Hardwared)
- ۲- کنترل ریز برنامه نویسی شده (Micro Programmed)



## سیکل دستورالعمل

برنامه‌ای که در واحد حافظه کامپیوتر ذخیره شده از دنباله‌ای از دستورالعمل‌ها تشکیل گردیده و در سیکل‌های اجرایی دستورالعمل اجرا می‌شود. در کامپیوتر پایه هر سیکل دستورالعمل از فازهای زیر تشکیل شده است:

- ۱- برداشت یک دستور از حافظه
- ۲- رمزگشایی دستور
- ۳- یافتن آدرس موثر
- ۴- اجرای دستورالعمل

## ریز عملیات برداشت و رمزگشایی:

در آغاز شمارنده برنامه، Pc با آدرس اولین دستورالعمل برنامه بارگذاری می‌گردد و شمارنده Sc از بین رفته و سیگنال  $T_0$  را تولید می‌کند. سپس با هر پالس ساعت Sc یک واحد افزایش یافته تا سیگنالهای  $T_0$ ،  $T_1$ ،  $T_2$  و ... تولید گردند.

یادداشت:

.....

.....

.....

.....

$$T_0: AR \leftarrow PC$$

$$T_1: IR \leftarrow M[AR], PC \leftarrow PC + 1$$

$$T_2: D_0, \dots, D_7 \leftarrow \text{Deode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$$

نکته این که در کلاک‌های  $T_0, T_1, T_2$  برداشت و رمزگشایی دستور صورت می‌گیرد، در نتیجه عملیات مورد انجام  $T_3$  توسط روابط زیر مشخص می‌گردد:

$$D_7' I' T_3 = \text{nothing}$$

اگر دستور حافظه‌ای مستقیم باشد ( $I=0$ ) کاری انجام نمی‌پذیرد.

$$D_7' I T_3: AR \leftarrow M[AR]$$

اگر دستور حافظه‌ای غیر مستقیم باشد.

$$D_7' I' T_3: \text{Execute a register Instruction}$$

اگر دستور ثباتی باشد اجرا می‌شود.

$$D_7 I T_3: \text{Execute an I/O Instruction}$$

اگر دستور I/O باشد اجرا می‌شود.

### ریز عملیات مربوط به دستورات حافظه‌ای:

AND:  $D_0 T_4: DR \leftarrow M[AR]$

$$D_0 T_5: AC \leftarrow AC \wedge DR$$

ADD:  $D_1 T_4: DR \leftarrow m[AR]$

$$D_1 T_5: AC \leftarrow AC + DR, E \leftarrow \text{Cout}, SC \leftarrow 0$$

LDA:  $D_2 T_4: DR \leftarrow M[AR]$

$$D_2 T_5: AC \leftarrow DR, SC \leftarrow 0$$

STA  $D_3 T_4: M[AR] \leftarrow AC, SC \leftarrow 0$

BUN  $D_4 T_4: PC \leftarrow AR, SC \leftarrow 0$

BSA:  $D_5 T_4: M[AR] \leftarrow PC, AR \leftarrow AR + 1$

$$D_5 T_5: PC \leftarrow AR, SC \leftarrow 0$$

### طراحی کامپیوتر پایه:

کامپیوتر پایه دارای بخش‌های سخت‌افزاری زیر است:

۱- یک واحد حافظه با ۴۰۹۶ کلمه ۱۶ بیتی

۲- ۸ ثبات  $INPR - OUTFR - TR - IR - AC - DR - PC - AR$  و یک شمارنده SC

۳- هفت فلیپ فلاپ  $FGO, FGI, IEN, R, E, S, I$

۴- دو دیکدر: یک دیکدر عملیاتی  $3 \times 8$  و یک دیکدر زمانی  $4 \times 16$

۵- یک گذرگاه مشترک ۱۶ بیتی

۶- کنترل منطقی گیتی

۷- جمع کننده و مدار منطقی متصل به ورودی AC

### یادداشت:

.....

.....

.....

.....

### گیت‌های مدار کنترل:

- خروجی‌های مدار کنترل منطقی عبارتند از:
- ۱- سیگنال‌های کنترل ورودی 9 عدد ثبات
  - ۲- سیگنال‌های کنترل ورودی‌های خواندن و نوشتن
  - ۳- سیگنال‌هایی برای فعال نمودن، پاک کردن و مکمل سازی فیلپ فلاپها
  - ۴- سیگنال‌های مربوط به  $S_2, S_1, S_0$  برای انتخاب ثبات برای گذرگاه
  - ۵- سیگنال‌های کنترل مدار منطقی و جمع کننده AC

یادداشت:

.....

.....

.....

.....

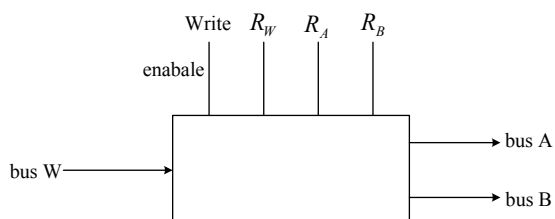


## تست‌ها

۱- در یک ماشین 32 بیتی باینری دستور  $rd \leftarrow rs + rt$  add U rd, rs, rt به صورت زیر تعبیر می‌شود: (سال ۸۵)

31	26	21	16	11	6	0
op	rs	rt	rd	shamt	funct	

اگر این ماشین دارای رجیستر فایل مقابل باشد تعداد بیت‌های  $R_W$ ,  $R_A$ ,  $R_B$  و تعداد رجیسترها به ترتیب چند است؟



(۱) 32, 5, 32, 5

(۲) 16, 6, 16, 6

(۳) 32, 6, 16, 5

(۴) 16, 5, 32, 6

۲- کدام عبارت صحیح است؟ (سال ۸۷)

(۱) برای پیاده سازی گذرگاه مشترک با وجود 8 ثبات 16 بیتی، جهت پیاده سازی با روش بافر سه حالت به 8 بافر 16 بیتی و یک رمز گشایی  $3 \times 8$  نیاز است و جهت پیاده سازی با MUX، به  $16 \times 1$  MUX نیاز است.

(۲) برای پیاده سازی گذرگاه مشترک با وجود 8 ثبات 16 بیتی، جهت پیاده سازی با روش بافر سه حالت به 8 بافر 16 بیتی و یک رمز گشایی  $4 \times 16$  نیاز است و جهت پیاده سازی با MUX، به  $16 \times 1$  MUX نیاز است.

(۳) برای پیاده سازی گذرگاه مشترک با وجود 8 ثبات 16 بیتی، جهت پیاده سازی با روش بافر سه حالت به 16 بافر 8 بیتی و یک رمز گشایی  $3 \times 8$  نیاز است و جهت پیاده سازی با MUX، به  $16 \times 1$  MUX نیاز است.

(۴) برای پیاده سازی گذرگاه مشترک با وجود 8 ثبات 16 بیتی، جهت پیاده سازی با روش بافر سه حالت به 16 بافر 8 بیتی و یک رمز گشایی  $4 \times 16$  نیاز است و جهت پیاده سازی با MUX، به  $16 \times 1$  MUX نیاز است.

۳- اتصال 14 ثبات 8 بیتی (که دارای پایه Load هستند) به یک باس داده چه سخت افزاری نیاز دارد تا بتوان محتویات هر یک از ثبات‌ها را در هر ثبات دیگر قرار داد؟

(سال ۸۵)

(۲) 14 عدد  $8 \times 1$  MUX و یک عدد دکودر  $4 \times 16$

(۱) 8 عدد دکودر  $4 \times 16$  و یک عدد  $8 \times 1$  MUX

(۴) 8 عدد  $16 \times 1$  MUX و یک عدد دکودر  $4 \times 16$

(۳) 8 عدد دکودر  $3 \times 8$  و 14 عدد  $16 \times 1$  MUX

یادداشت:

.....

.....

.....

.....

## پاسخ‌ها

۱ - گزینه ۱ درست است.

نکته : اگر طول فیلد رجیستر  $n$  بیت باشد، تعداد رجیسترها  $2^n$  تا است و طول خطوط کنترل برابر  $n$  است. با توجه به نکته‌ی بالا تعداد رجیسترها  $2^5 = 32$  تا است، و طول خطوط کنترل  $RA$  و  $RW$  برابر ۵ است. طول  $busW$  برابر طول  $bas$  ماشین است، یعنی ۳۲ بیت.

۲ - گزینه ۳ درست است.

۳ - گزینه ۴ درست است.

نکته : تعداد مالتی پلکسرها برابر تعداد بیت‌های رجیسترهای  $bas$  و تعداد ورودی‌های مالتی پلکسر برابر تعداد ثبات هاست، به شرطی که توانی از ۲ باشند.

در نتیجه تعداد ۸ مالتی پلکسر با ۱۶ ورودی (از آن جا که ۱۴ توان ۲ نیست) نیاز است.

یادداشت:

.....

.....

.....

.....

## فصل دوم

### کنترل ریزبرنامه نویسی شده

#### واحد کنترل :

تولید سیگنالهای کنترل ثابتها، باس مشترک و ALU جهت تولید دنباله‌ای از عملیات به منظور اجرای دستورالعمل بر عهده واحد کنترل می‌باشد. متغیرهای کنترلی که بوسیله واحد کنترل تولید می‌شوند دنباله‌ای از 1,0 ها هستند که آن‌ها را کلمه کنترلی می‌نامند.

#### روش‌های پیاده‌سازی واحد کنترل

۱. سخت افزاری (Hardware) :
- سیگنالهای کنترلی توسط سخت‌افزارهای حاصل از مدارات منطقی، مانند گیت‌های ترکیبی تولید می‌گردد.

۲. ریزبرنامه‌نویس (Microprogram) :

- مجموعه سیگنالهای کنترلی که شامل یک کلمه کنترلی می‌باشد.
- مجموعه کلمات کنترلی و موارد اضافی
- مجموعه ریزدستورالعملها در یک حافظه کنترلی
- عملیات محاسبه آدرس مؤثر و اجرای آن

#### سازمان کنترل ریزبرنامه نویسی شده:

۱. حافظه کنترل شامل روالهای ریزبرنامه
۲. ثبات آدرس حافظه کنترل که آدرس ریزدستورالعمل بعدی در آن وجود دارد
۳. ثبات داده حافظه کنترل که شامل ریزدستورالعمل جاری می‌باشد
۴. تولید کننده آدرس بعدی

#### یادداشت:

.....

.....

.....

.....

نکته اینکه به تبدیل کد عملیاتی یک دستور به یک آدرس، در حافظه کنترل، عمل نگاشت (Mapping) می‌گویند.

آدرس ریز دستور بعدی در حافظه کنترل به صورت‌های زیر محاسبه می‌گردد:

۱- جمع CAR با یک

۲- پرش غیر شرطی یا شرطی

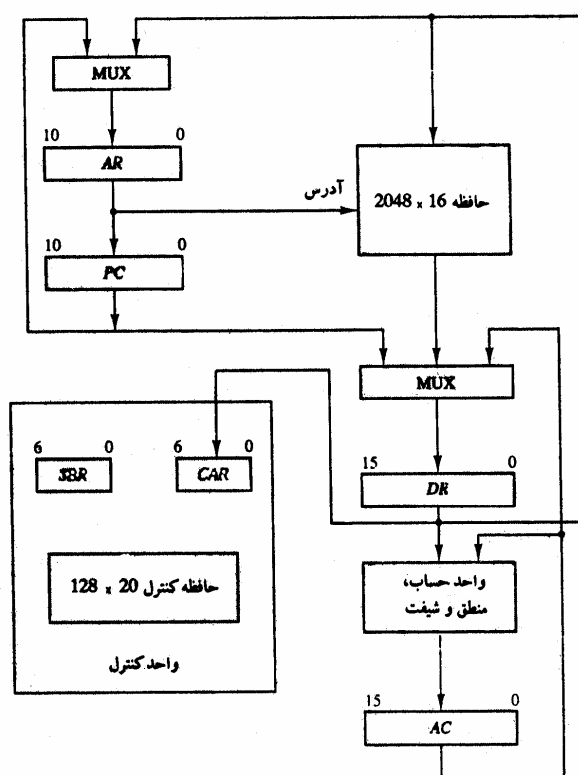
۳- فراخوانی و بازگشت از Sub Routine های داخل حافظه کنترل

## آرایش کامپیوتر:

کامپیوتر مورد نظر دارای دو واحد حافظه می‌باشد که مورد اول حافظه اصلی برای ذخیره دستورات و Data و دومی یک حافظه کنترل برای ذخیره ریز برنامه‌ها می‌باشد. چهار ثابت به واحد پردازنده (AC-DR-AR-PC) و دو ثابت به واحد کنترل (SBR, CAR) مربوط‌اند.

باید دقت شود که جابجایی اطلاعات در این کامپیوتر نمی‌تواند از طریق باس مشترک صورت گیرد، و دیگر اینکه ورودی و خروجی و وقفه نیز ندارند و حافظه کمتری مورد نیاز است.

نکته دیگر این که AC و DR، ۱۶ بیتی می‌باشند و AR و PC، ۱۱ بیتی هستند، و حافظه کنترل دارای ۱۲۸ ریز دستور ۲۰ بیتی می‌باشد. در نتیجه SBR و CAR هر کدام ۷ بیتی می‌باشند.



یادداشت:

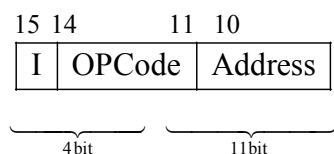
.....

.....

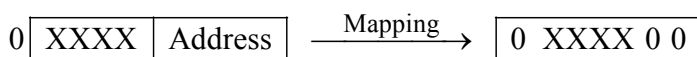
.....

.....

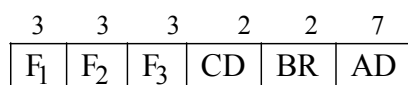
برای کامپیوتر ساده سه نوع دستورالعمل در نظر گرفته شده است، فیلد آدرس که 11 بیتی است، فیلد opcode، 4 بیتی و فیلد مد آدرس که یک بیتی می‌باشد.



نگاشت از یک دستور به آدرس زیر دستور در حافظه کنترل به صورت زیر انجام می‌پذیرد:



هر دستور شامل 4 ریز دستور می‌باشد. با توجه به اینکه که کد عملیاتی 4 بیتی می‌باشد در کل می‌توان 16 دستور برای این ماشین تعریف نمود زمانی که هر دستور دارای 4 ریز دستور باشد مجموعاً  $4 \times 16 = 64$  ریز دستور مورد نیاز است.



هر فیلد ریز عمل 3 بیتی است که می‌تواند 7 ریز عمل را مشخص کند (None=000) و جمعاً 21 ریز عمل را می‌توان معرفی نمود، هر ریز دستور حداکثر می‌تواند 3 ریز عمل را اجرا کند. اگر تعداد کمتری ریز عمل مورد نیاز بود یک یا بیشتر از فیلدهای ریز عمل دارای کد 000 می‌بودند. باید توجه داشت که ریز عمل‌ها با یکدیگر تناقض نداشته باشند یعنی بتوانند با هم اجرا شوند.

## Fetch Routine

ریز دستورات مورد نیاز برای این روال عبارتند از :

$AR \leftarrow PC$

$DR \leftarrow M[AR]$  ,  $PC \leftarrow PC+1$

$AR \leftarrow DR(0-10)$  ,  $CAR(2-5) \leftarrow DR(11-14)$  ,  $CAR(0,1,6) \leftarrow 0$

Fetch Routine به سه ریزدستورالعمل نیاز دارد، که در آدرس ۶۴، ۶۵ و ۶۶ حافظه کنترل قرار داده می‌شوند. داریم:

ORG 64

FETCH:	PCTAR	U	JMP	NEXT
	READ INCPC	U	JMP	NEXT
	DRTAR	U	JMP	

یادداشت:

.....

.....

.....

.....

جدول سمبول‌ها و کدهای باینری برای فیلدهای ریز دستور

F1	Micro opration	Symbol	
000	None	NOP	
001	$AC \leftarrow AC + DR$	ADD	
010	$AC \leftarrow 0$	CLRAC	
011	$AC \leftarrow AC + 1$	INCAC	
100	$AC \leftarrow DR$	DRTAC	
101	$AR \leftarrow DR(0-10)$	DRTAR	
110	$AR \leftarrow PC$	PCTAR	
111	$M[AR] \leftarrow DR$	WRITE	
F1	Micro opration	Symbol	
000	None	NOP	
001	$AC \leftarrow AC - DR$	SUB	
010	$AC \leftarrow AC \vee DR$	OR	
011	$AC \leftarrow AC \wedge DR$	AND	
100	$DR \leftarrow M[AR]$	READ	
101	$DR \leftarrow AC$	ACTDR	
110	$DR \leftarrow DR + 1$	INCDR	
111	$DR(0-10) \leftarrow PC$	PCTDR	
F1	Micro opration	Symbol	
000	None	NOP	
001	$AC \leftarrow AC \oplus DR$	XOR	
010	$AC \leftarrow \overline{AC}$	COM	
011	$AC \leftarrow shl\ AC$	SHL	
100	$AC \leftarrow shr\ AC$	SHR	
101	$PC \leftarrow PC + 1$	INCPC	
110	$PC \leftarrow AR$	ARTPC	
111	Reserved		
CD	Condition	Symbol	Comments
00	Always=1	U	انشعاب غیرشرطی
01	DR(15)	I	بیت آدرس غیرمستقیم
10	AC(15)	S	بیت علامت AC
11	AC=0	Z	مقدار صفر در AC

یادداشت:

.....

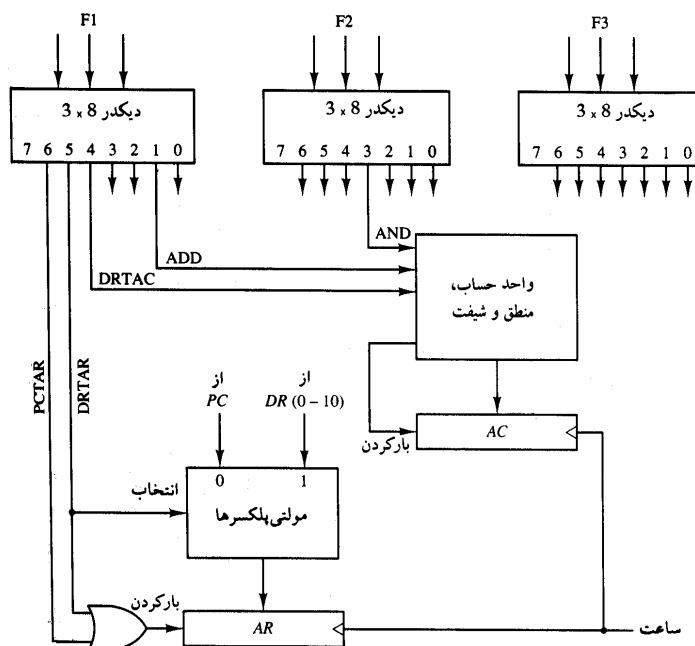
.....

.....

.....

BR	Symbol	Function
00	JMP	CAR $\leftarrow$ AD if condition=1 CAR $\leftarrow$ CAR+1 if conditon=0
01	CALL	CAR $\leftarrow$ AD, SBR $\leftarrow$ CAR+1 if condition=1 CAR $\leftarrow$ CAR+1 if condition=0
10	RET	CAR $\leftarrow$ SBR (Return from subrountione)
11	MAP	CAR(2-5) $\leftarrow$ DR(11-14), CAR(0,1,6) $\leftarrow$ 0

در هر روتین باید ریز دستوراتی داشت که بتوان آدرس موثر را محاسبه کرد. مد آدرس دهی غیر مستقیم در تمام دستورات مراجعه به حافظه وجود دارد. اگر ریز دستورات مربوط به یافتن آدرس موثر را به عنوان یک زیر روال در حافظه کنترل قرار دهیم، در کلمات حافظه کنترل صرفه جویی کرده‌ایم.



### یادداشت:

.....

.....

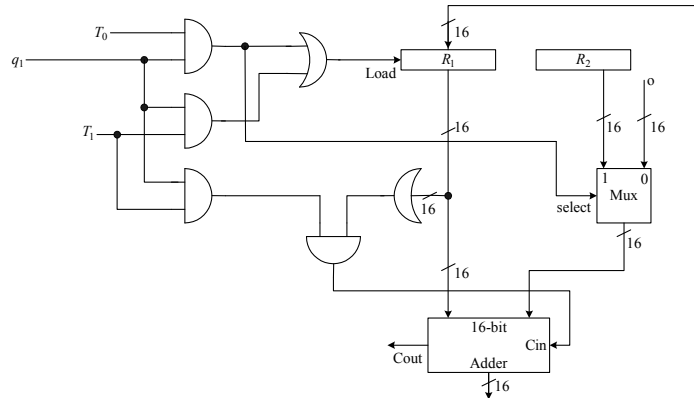
.....

.....

## تست‌ها

(سال ۸۵)

۱- شکل زیر پیاده‌سازی کدام دو میکروآپ پیایی است که رخ می‌دهند؟



$$T_0q_1: \text{if}(R_1=0) \text{ then } (R_1 \leftarrow R_1 + R_2) \quad (1)$$

$$T_1q_1: R_1 \leftarrow R_1 + 1$$

$$T_0q_1: R_1 \leftarrow R_1 + R_2$$

$$T_1q_1: \text{if}(R_1 \neq 0) \text{ then } (R_1 \leftarrow R_1 + 1) \text{ else } (R_1 \leftarrow R_1 + R_2) \quad (2)$$

$$T_0q_1: R_1 \leftarrow R_1 + R_2$$

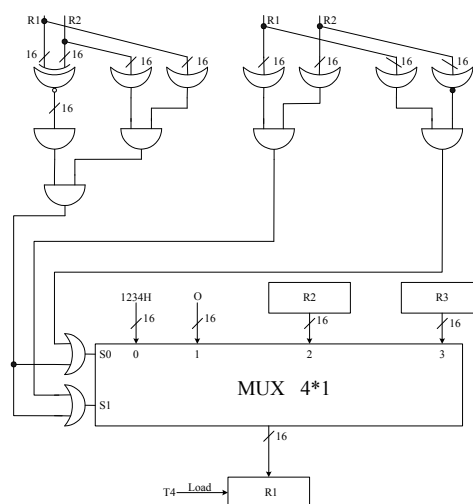
$$T_1q_1: \text{if}(R_1 \neq 0) \text{ then } R_1 \leftarrow R_1 + 1 \quad (3)$$

$$T_0q_1: \text{if}(R_1 \neq 0) \text{ then } (R_1 \leftarrow R_1 + R_2) \text{ else } (R_1 \leftarrow 0) \quad (4)$$

$$T_1q_1: R_1 \leftarrow R_1 + 1$$

۲- با فرض این که رجیسترهای  $R_1, R_2$  و  $R_3$  16 بیتی هستند سخت‌افزار مقابل کدام میکروآپ را پیاده‌سازی می‌کند؟

(سال ۸۶)



یادداشت:

.....

.....

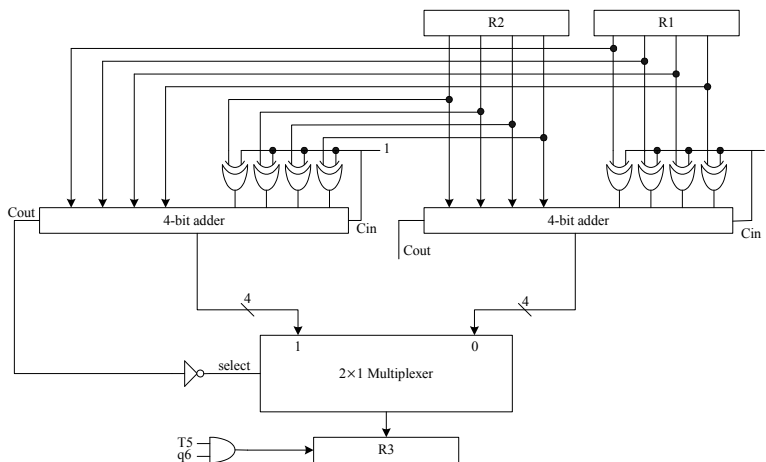
.....

.....



- T4: if  $(R_1 = R_2 \neq 1)$  THEN  $(R_1 \leftarrow R_3)$  ELSE IF  $(R_1 = 0) \wedge (R_2 \neq 1)$  THEN  $(R_1 \leftarrow R_3)$  ELSE IF  $((R_1 \neq 1) \wedge (R_2 = 0))$  THEN  $(R_1 \leftarrow 0)$  ELSE  $(R_1 \leftarrow 1234_H)$  (۱)
- T4: if  $(R_1 = R_2 \neq 0)$  THEN  $(R_3 \leftarrow R_1)$  ELSE IF  $((R_1 = 1) \wedge (R_2 \neq 0))$  THEN  $(R_1 \leftarrow R_2)$  ELSE IF  $((R_1 \neq 0) \wedge (R_2 = 0))$  THEN  $(R_1 \leftarrow 0)$  ELSE  $(R_1 \leftarrow 1234_H)$  (۲)
- T4: if  $(R_1 = R_2 \neq 1)$  THEN  $(R_2 \leftarrow R_3)$  ELSE IF  $(R_1 = 1) \wedge (R_2 \neq 0)$  THEN  $(R_1 \leftarrow R_2)$  ELSE IF  $((R_1 \neq 0) \wedge (R_2 = 0))$  THEN  $(R_1 \leftarrow 0)$  ELSE  $(R_1 \leftarrow 1234_H)$  (۳)
- T4: if  $(R_1 = R_2 \neq 0)$  THEN  $(R_1 \leftarrow R_3)$  ELSE IF  $((R_1 = 0) \wedge (R_2 \neq 0))$  THEN  $(R_1 \leftarrow R_2)$  ELSE IF  $((R_1 \neq 0) \wedge (R_2 = 0))$  THEN  $(R_1 \leftarrow 0)$  ELSE  $(R_1 \leftarrow 1234_H)$  (۴)

۳- محتویات ثبات‌های 4 بیتی  $R_1$ ،  $R_2$  اعداد بدون علامت است. وظیفه سخت‌افزار را کدام میکرو آپ بیان می‌کند؟ (سال ۸۷)



- T5 q6: if  $(R1 < R2)$  then  $R3 \leftarrow R1 - R2$  else  $R3 \leftarrow R2 - R1$  (۱)
- T5 q6: if  $(R1 > R2)$  then  $R3 \leftarrow R2 - R1$  else  $R3 \leftarrow R1 - R2$  (۲)
- T5 q6: if  $(R1 < R2)$  then  $R3 \leftarrow R2 - R1$  else  $R3 \leftarrow R1 - R2$  (۳)
- T5 q6: if  $(R1 > R2)$  then  $R3 \leftarrow R1 - R2$  else  $R3 \leftarrow R2 - R1$  (۴)

۴- یک مکانیزم سخت افزاری که دارای دو ثبات  $A$  و  $B$  و چهار فلیپ فلاپ  $e$ ،  $s$ ،  $p$ ،  $q$  است توسط ریز دستورات زیر توصیف

(سال ۸۵)

شده است. کدام گزینه در مورد عملکرد این مکانیزم صحیح است؟

- $s: s \leftarrow 0, e \leftarrow 0, p \leftarrow 1, q \leftarrow 0$   
 $p: p \leftarrow 0, \text{IF } (A < B) \text{ then } (e \leftarrow 1) \text{ else } (q \leftarrow 1)$   
 $q: A \leftarrow A - B, p \leftarrow 1, q \leftarrow 0$

- (۱) این مکانیزم حاصل تفریق  $A-B$  را محاسبه و در  $A$  قرار می‌دهد.  
 (۲) این مکانیزم باقیمانده تقسیم صحیح  $A$  بر  $B$  را محاسبه و در  $A$  قرار می‌دهد.  
 (۳) از نظر سخت افزاری پیاده سازی مکانیزم توصیف شده به خاطر تناقض در عملیات غیر ممکن است.  
 (۴) از نظر سخت افزاری پیاده سازی مکانیزم توصیف شده به خاطر عدم وجود ترتیب زمانی عملیات غیر ممکن است.

یادداشت:

.....  
 .....  
 .....  
 .....

## پاسخ‌ها

۱ - گزینه ۳ درست است.

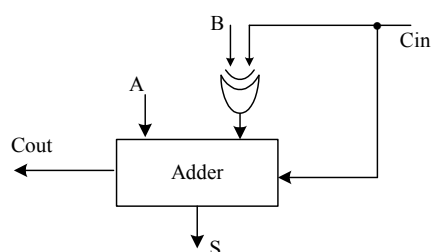
اگر  $T_0q_1$  برابر یک باشد، MUX ثبات  $R_2$  را انتخاب می‌کند و جمع‌کننده  $R_2 + R_1$  را حساب می‌کند و در  $R_1$  قرار می‌دهد. اگر  $T_1q_1$  یک باشد و اگر  $R_1 \neq 0$  در این صورت جمع‌کننده  $R_1 + 0 + 1$  را در  $R_1$  قرار می‌دهد.

۲ - گزینه ۴ درست است.

اگر  $R_1 = R_2$  بوده، به شرطی که مخالف صفر باشند سپس  $S_1S_0 = 11$  شده و  $R_3$  به  $R_1$  بار می‌شود. اگر  $R_1 \neq R_2$  باشد و  $R_1 = 0$  و  $R_2 \neq 0$  آن‌گاه  $S_1S_0 = 10$  شده و  $R_2$  به  $R_1$  بار می‌شود که گزینه ۴ است.

۳ - گزینه ۱ درست است.

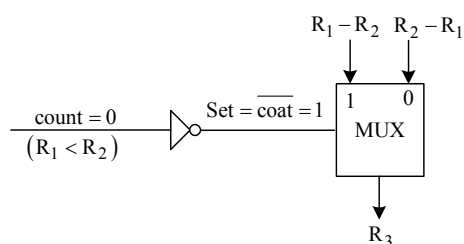
نکته :



```

if Cin = 0 then S := A + B;
if Cin = 1 then S := A + B̄ + 1 = A + 2'sB
                = A - B;
if Count = 1 then A >= B;
if Count = 0 then A < B;
    
```

با توجه به نکته بالا، مدار را ساده می‌کنیم:



حال به سادگی می‌توان فهمید که گزینه ۱ درست است.

۴ - گزینه ۲ درست است.

الگوریتم سؤال، B را n بار از A کم می‌کند تا A از B کوچکتر شود. n برابر تقسیم صحیح A بر B است. پس گزینه ۲ درست است.

یادداشت:

.....

.....

.....

.....

## فصل سوم

### پردازش خط لوله‌ای

پردازش موازی، روشی را برای اجرای همزمان چند دستورالعمل بوجود می‌آورد. پردازش موازی به زمان تاخیر یک کار کمکی نمی‌کند، بلکه توان عملیاتی کل بار کاری را بهبود می‌بخشد و باعث افزایش سرعت محاسبات می‌گردد.

تسریع (Speed up) نسبت سرعت‌های دو محاسبه از یک مسئله را شامل می‌گردد و موارد زیر را در بر دارد:

- طول نامتوازن مراحل خط لوله باعث کاهش تسریع می‌شود.
- زمان مصرفی برای پر شدن و خالی شدن خط لوله باعث کاهش تسریع می‌شود.

$$\text{Speed Up} = \frac{\text{زمان اجرا روی یک پردازنده}}{\text{زمان اجرا روی } P \text{ پردازنده}} = \frac{nt_n}{(k+n-1)t_p}$$

بازدهی پایپلاین (Efficiency):  $E = \frac{S}{K}$  که  $S$  تسریع به دست آمده و  $K$  طول پایپلاین است.

✓ اگر تمام طبقات دارای تأخیر انتشار مساوی باشند بهترین کارایی به دست می‌آید.

طبقه بندی پایپلاین‌ها:

۱- Arithmetic Pipeline

۲- Instruction Pipeline

۳- Processor Pipeline

یادداشت:

.....

.....

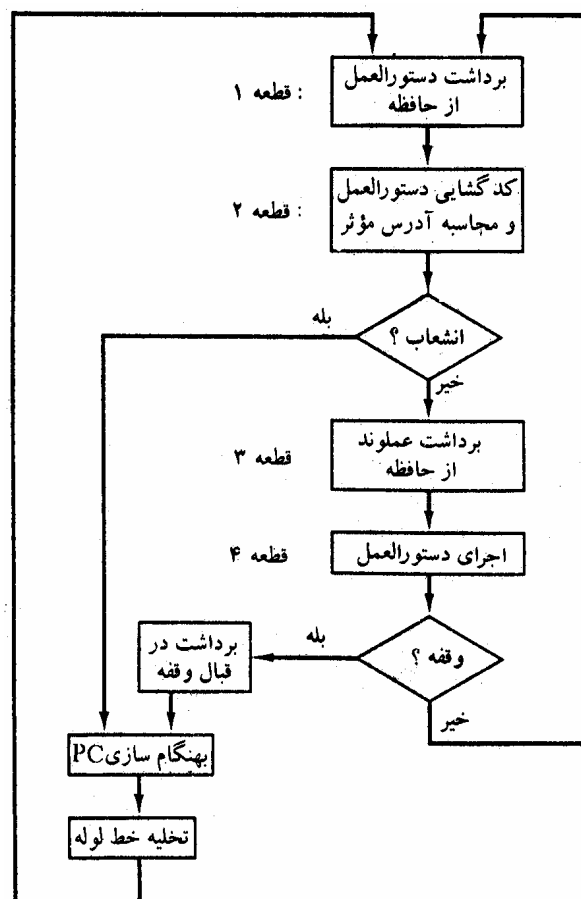
.....

.....

**پایپلاین محاسباتی:** خط لوله محاسباتی در کامپیوتر های بسیار سریع موجود می باشد و برای پیاده سازی عملیات ممیز شناور، ضرب اعداد با ممیز ثابت و سایر اعمال محاسباتی که در مسائل علمی یافت می گردند استفاده می شود.

**پایپلاین دستورالعمل:** دارای مراحل مختلفی از قبیل:

۱. واکشی دستور از حافظه
۲. رمزگشایی دستورالعمل
۳. محاسبه آدرس مؤثر
۴. واکشی عملوندها از حافظه
۵. اجرای دستورالعمل
۶. ذخیره نمودن نتیجه در مکان مناسبی از حافظه



**نکته:** بطور کلی سه مشکل اساسی موجود است که باعث انحراف پایپلاین از حالت عادی آن می گردد:

**۱- منابع مشترک:** زمانی که دوقطعه به طور همزمان نیاز به حافظه داشته باشند یکی از آنها باید دست از کار بکشد. برای رفع مشکل از حافظه های داده و دستورات جداگانه استفاده می نمائیم.

**یادداشت:**

.....

.....

.....

.....

۲- وابستگی داده‌ای: زمانی رخ می‌دهد که دستوری نیازمند نتیجه دستور قبلی باشد. بنابراین زمانی مورد پردازش قرار می‌گیرد که دستور قبلی کامل گردد.

۳- مشکلات انشعابی: این مشکلات بوسیله دستورالعمل انشعاب و سایر دستوراتی که محتویات PC را تغییر می‌دهند تولید می‌گردد.

### فرمول آمدال:

اگر  $f$  درصد کد سریال باشد خواهیم داشت:

$$S = \frac{1}{f + \frac{1-f}{p}} \rightarrow S_{\max} = \frac{1}{f} \quad (p \rightarrow \infty)$$

### محاسبه تأخیر پرش در پایپلاین:

اگر  $n$  کار داشته باشیم و عمق پایپلاین  $k$  باشد، در حالت بدون دستور پرش زمان کل  $(n + (K-1))T$  می‌شود. اما اگر  $j$  دستور پرش داشته باشیم زمان اجرا عبارتست از:

$$(n - j) * T + j * K * T$$

یادداشت:

.....

.....

.....

.....

## تست‌ها

۱- برای اجرا دستورات در یک پردازنده باید چهار کار متوالی انجام شود که مدت زمان هر کار به ترتیب 4ns، 4ns، 9ns و 3ns می‌باشد. اگر پردازنده دیگری طراحی کنیم که همین کارها را به صورت خط لوله (pipeline) انجام دهد و تاخیر ثبات خط لوله را 1ns فرض کنیم. افزایش سرعتی که پردازنده جدید را در اجراء 7 دستور متوالی غیروابسته تولید می‌کند چقدر است؟ (سال ۸۵)

(۱) 1.4 (۲) 2 (۳) 1.7 (۴) 4

۲- پردازنده‌ای با ساختار خط لوله (pipeline) دارای چهار مرحله‌ی زیر است:  
WB (ذخیره حاصل دستور) → EX (اجرای دستورات) → ID (دیکود دستورات) → IF (واکشی دستورات)  
فرض کنید با هر واکشی می‌توان 3 دستور را وارد پردازنده کرد ولی به دلیل نبود واحدهای موازی کافی، در مراحل بعدی دستورات فقط دو دستور از سه دستور واکشی شده همزمان قابل اجرا است. حال اجرای 33 دستور در این پردازنده چند پالس ساعت طول می‌کشد؟ (سال ۸۷)

(۱) 20 (۲) 25 (۳) 35 (۴) 36

۳- چهار قطعه برنامه مقابل را در نظر بگیرید:

A	B	C	D
load $R_1, (R_2 + 100)$	load $R_1, (R_2 + 100)$	load $R_1, (R_2 + 100)$	load $R_1, (R_2 + 100)$
Add $R_4, R_1$	Add $R_4, R_1$	Add $R_4, R_1$	Add $R_4, R_1$
Sub $R_5, R_1$	Sub $R_4, R_5$	Sub $R_4, R_6$	Sub $R_2, R_3$
Add $R_6, R_1$	Add $R_6, R_1$	Add $R_4, R_2$	Add $R_2, R_5$

کدام برنامه در صورت اجرا روی یک پردازنده با قابلیت اجرای موازی دستورات مستقل کارآیی بیشتری دارد؟ (فرض کنید  $OP\ R_1, R_2 \equiv R_1 \leftarrow (R_1) OP (R_2)$ ، برای مثال:  $Add\ R_1, R_2 \equiv R_1 \leftarrow (R_1) + (R_2)$ ). (سال ۸۸)

(۱) A (۲) B (۳) C (۴) D

۴- در یک پردازنده لوله‌ای (pipeline) از یک خط لوله 11 سطحی برای واکشی و اجرای دستورات استفاده می‌شود. اگر 10% دستورات برنامه پرش باشند حداکثر و حداقل تسریع قابل اتصال توسط این پردازنده نسبت به پردازنده مشابه غیر لوله‌ای چقدر خواهد بود؟ (فرض کنید مشکلات وابستگی داده‌ای و دسترسی به حافظه برای اجرای دستورات وجود ندارد). (سال ۸۵)

(۱) حداکثر 10 و حداقل 8.5 (۲) حداکثر 11 و حداقل 5.5 (۳) حداکثر 10 و حداقل 8.1 (۴) حداکثر 11 و حداقل 9.9

۵- در یک پردازنده خط لوله (pipe line) با 5 (stage)، تأخیر مراحل به ترتیب برابر است با: 2، 1، 4 و 3 نانو ثانیه. بیشترین میزان تسریعی که از این پایپ لاین می‌توان گرفت چقدر است؟ (سال ۸۸)

(۱) 2 (۲) 3 (۳) 4 (۴) 5

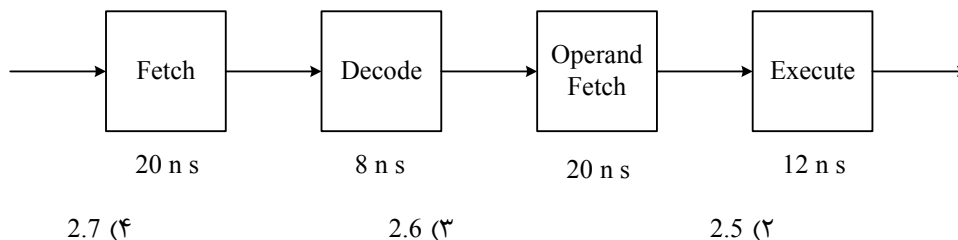
یادداشت:

.....  
 .....  
 .....  
 .....

۶- در یک سیستم خط لوله (pipe line) که از هشت مرحله (stage) تشکیل شده است، برای پردازش 100 ورودی حداقل چند کلاک نیاز است؟ (سال ۸۸)

- (۱) 106 (۲) 107 (۳) 108 (۴) 109

۷- به فرض داشتن یک خط لوله 4 سطحی برای اجرای دستورات در یک پردازنده، اگر در یک برنامه به طور متوسط در هر 10 دستور یک پرش وجود داشته باشد و به احتمال  $\frac{1}{2}$  پرش انجام شود. حداکثر تسریع به دست آمده برای اجرای این برنامه نسبت به زمانی که پردازنده لوله‌ای نیست در مدت طولانی اجرای برنامه چقدر خواهد بود؟ (سال ۸۶)



۸- در یک کامپیوتری از معماری خط لوله‌ای استفاده می‌شود که 8 بخش است و هر بخش در یک پالس ساعت عملیات خود را انجام می‌دهد. در این کامپیوتر با فرض آن که خط لوله همیشه پر خواهد بود، میزان افزایش سرعت (speed up) نسبت به کامپیوتر بدون خط لوله، برای برنامه‌ای متشکل از 1000 دستورالعمل که هر دستور 8 پالس ساعت و هر پالس ساعت 1 نانو ثانیه طول می‌کشد چقدر خواهد بود؟ (سال ۸۷)

- (۱) 8 (۲)  $\frac{8000}{1007}$  (۳)  $\frac{8000}{1008}$  (۴) برای دستورات محدود تفاوتی ندارد.

یادداشت:

.....

.....

.....

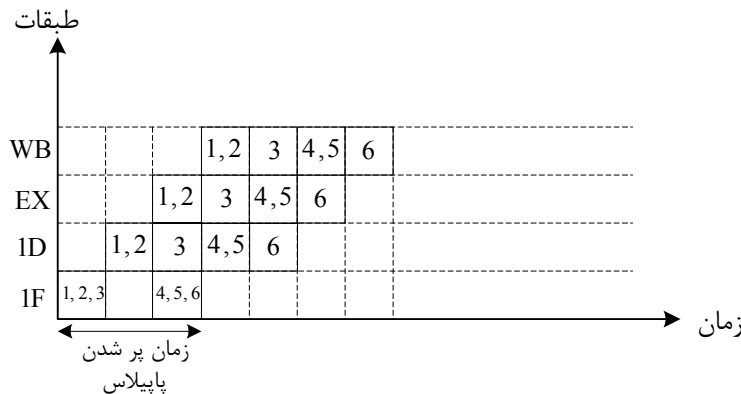
.....

## پاسخ‌ها

۱ - گزینه ۱ درست است.

$$\begin{aligned} \Rightarrow \text{تسریع} &= \frac{140}{100} = 1.4 \\ \text{بدون پایپلین} &= 7 \times (4 + 4 + 9 + 3) = 140 \text{ ns} \\ \text{با پایپلین} &= 4 \times 10 + (7 - 1) \times 10 = 100 \text{ ns} \end{aligned}$$

۲ - گزینه ۲ درست است.



3 کلاک طول می‌کشد تا پایپ لاین پر شود.

سپس، بعد از هر 2 کلاک، 3 دستور اجرا می‌شود. بنابراین برای اجرای 33 دستور:

$$3 + \left( \frac{33}{3} \times 2 \right) = 3 + 22 = 25$$

25 کلاک (پالس ساعت) نیاز است.

۳ - گزینه ۱ درست است.

در بین چهار برنامه، برنامه A وابستگی بین دستورات ندارد بجز وابستگی به دستور اول که در همه 4 برنامه مشترک است. بنابراین برای اجرای موازی کارایی بیشتری دارد.

۴ - گزینه ۲ درست است.

$$\begin{aligned} T_{\text{pipe}} &= (n + k - 1) \times T = 11T + [(0.1 \times 11 + 0.9)n - 1] \times T \\ &= 2nT + 10T = T(2n + 10) \end{aligned}$$

$$T_{\text{unpipe}} = 11nT \quad \text{تسریع} = \frac{T_{\text{unpipe}}}{T_{\text{pipe}}} = \frac{11n}{2n + 10} \xrightarrow{n \rightarrow \infty} \text{تسریع} = 5.5$$

و در بهترین حالت  $K = 11 = \text{تسریع می‌باشد}$ .

۵ - گزینه ۲ درست است.

$$\text{تأخیر در پردازنده بدون پایپ لاین} = \text{تسریع} = \frac{2 + 1 + 2 + 4 + 3}{\text{Max}(2, 1, 2, 4, 3)} = \frac{12}{4} = 3$$

یادداشت:

.....

.....

.....

.....



۶ - گزینه ۲ درست است.

اگر در هر کلاک یک مرحله از خط لوله پر شود پس از ۸ مرحله خط لوله پر شده است و از آنجا به بعد در هر کلاک یک دستور اجرا می‌شود یعنی می‌توان گفت به تعداد  $100+8-1$  کلاک یعنی ۱۰۷ کلاک نیاز داریم.

۷ - گزینه ۳ درست است.

طبق سؤال به طور متوسط هر ۲۰ دستور یک بار پرش داریم.

$$\text{یعنی: } \frac{20 \times (20 + 8 + 20 + 12)}{(4 + 19) \times 20} = \frac{60}{23} = 2.6$$

۸ - گزینه ۲ درست است.

$$\text{Speed up} = \frac{n \cdot k \cdot t_p}{[k + (n - 1)]t_p} = \frac{8 \times 1000}{8 + (1000 - 1)} = \frac{8000}{1007}$$

یادداشت:

.....

.....

.....

.....

## فصل چهارم

### روش‌های محاسباتی پردازنده

#### ضرب اعداد:

ضرب دو عدد ممیز ثابت دودویی توسط عمل شیفت و جمع انجام پذیر است بدین ترتیب که بیت‌های ضرب‌کننده بطور متوالی با شروع از کم ارزشترین بیت موجود مشاهده گردند و اگر بیت ضرب‌کننده یک باشد ضرب‌شونده در پائین کپی گردد و در غیر اینصورت صفرها در پائین کپی گردند. اعدادی که در سطرهای متوالی کپی می گردند نسبت به سطر قبل یک بیت به چپ شیفت داده می‌شوند. در نهایت اعداد با هم جمع شده و نتیجه بدست آمده جواب ضرب می‌باشد.

#### ضرب‌کننده آرایه‌ای:

تست یک به یک بیت‌های ضرب‌کننده و بدست آوردن حاصل ضرب عملیات ترتیبی می‌باشد، ضرب دو عدد دودویی بوسیله یک ریز عمل و مدار ترتیبی که بیت‌های ضرب را ایجاد می‌نماید شکل می‌یابد. این روش یک روش با سرعت بالا برای ضرب دو عدد می‌باشد. در نتیجه یک ضرب‌کننده آرایه‌ای نیاز به تعداد زیادی گیت دارد. نمونه‌ای از ضرب آرایه ۴ بیت در ۳ بیت در شکل زیر نمایش داده شده است.

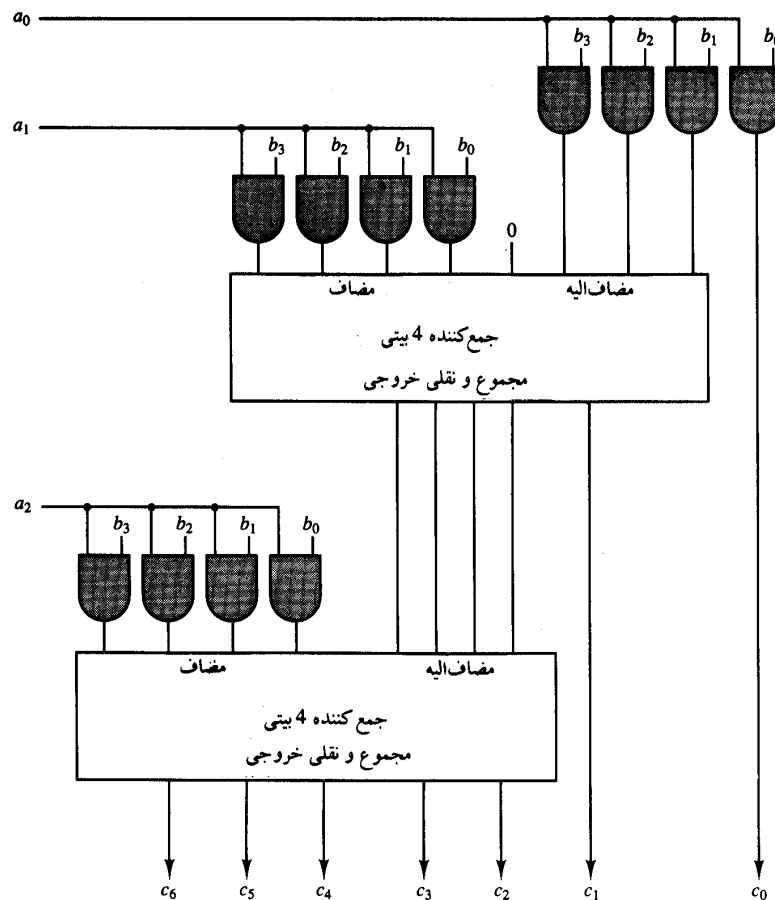
#### یادداشت:

.....

.....

.....

.....



### الگوریتم ضرب بوث:

این الگوریتم به مانند تمام الگوریتم های ضرب به بررسی بیت های ضرب کننده و شیفت می پردازد و مطابق با قوانین زیر عمل می نماید.

- ✓ اگر به اولین یک کم ارزش در رشته یک، در ضرب کننده برسیم ضرب شونده از حاصل ضرب جزئی کم می شود.
- ✓ اگر به اولین صفر (اگر قبل از آن یک باشد) در رشته صفر، در ضرب کننده برسیم ضرب شونده با حاصل ضرب جزئی جمع می شود.
- ✓ اگر بیت ضرب کننده با بیت یکسان باشد حاصل ضرب جزئی تغییر نمی یابد.

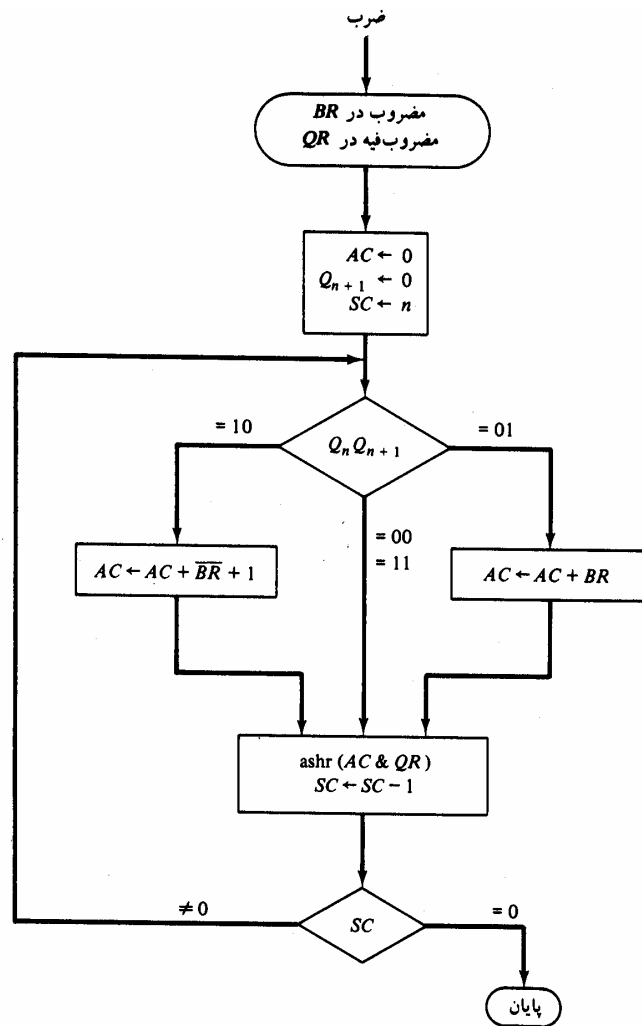
یادداشت:

.....

.....

.....

.....



## الگوریتم تقسیم

### ۱- الگوریتم مقایسه‌ای (Comparision Method)

- ✓ برای ذخیره کردن مقسوم باید 2-Reg در نظر بگیریم.
- ✓ با XOR نمودن Sign بیت‌های مقسوم و مقسوم علیه می‌توان Sign bit خارج قسمت را تولید نمود.
- ✓ بیت‌های مقسوم علیه را با بیت‌های MSB مقسوم مقایسه می‌کنیم.
- اگر مقسوم بزرگتر بود  $1 \rightarrow q$  و مقسوم علیه را از مقسوم و بعداً از باقیمانده جزئی تفریق می‌کنیم.
- اگر مقسوم کوچکتر بود  $0 \rightarrow q$  و صفحه را از مقسوم و بعداً از باقیمانده جزئی تفریق می‌کنیم.

### ۲- روش بازیافتی (Restoring Method)

یادداشت:

.....

.....

.....

.....

## ۳- روش غیربازیافتنی (Non Restoring Method)

فرمت یک عدد ممیز شناور بدین صورت است :

S	e	m
sign	exponent	mantissa

عدد مورد نظر از رابطه  $(-1)^s * 0.m * 2^e$  محاسبه می‌گردد. بیت‌های بیشتر برای مانتیس (m) دقت را افزایش می‌دهند (مانتیس عددی کسری یا صحیح باشد). و بیت‌های بیشتر برای نما (e) بازه را افزایش می‌دهند. اگر با ارزش‌ترین رقم m صفر نباشد آنگاه یک عدد ممیز شناور را نرمال شده نامند. در ضمن نمایش ممیز شناور محدوده اعداد قابل نمایش را افزایش می‌دهد. نکته دیگر اینکه مقدار بایاس یا افزونه عددی به هر نما افزوده می‌شود تا تمام نماها مثبت شوند.

## جمع و تفریق:

۱- تست صفر بودن هر کدام از عددها

۲- هم ردیف نمودن مانتیس‌ها

۳- جمع یا کم کردن مانتیس‌ها

۴- نرمال نمودن نتیجه

در این باره باید به نکات زیر نیز توجه نمود :

- ✓ ممکن است پس از عمل جمع over flow رخ دهد که در این صورت m را یک بیت به راست شیفت می‌دهیم و یکی به توان اضافه می‌کنیم.
- ✓ ممکن است پس از عمل تفریق under flow رخ دهد که در این صورت m را یک بیت به چپ شیفت می‌دهیم و یکی از توان کم می‌کنیم.
- ✓ هنگام جمع و تفریق عدد دارای توان کمتر را انتخاب کرده و m آن را به راست شیفت می‌دهیم و توان را اضافه می‌کنیم تا نماها برابر گردد.

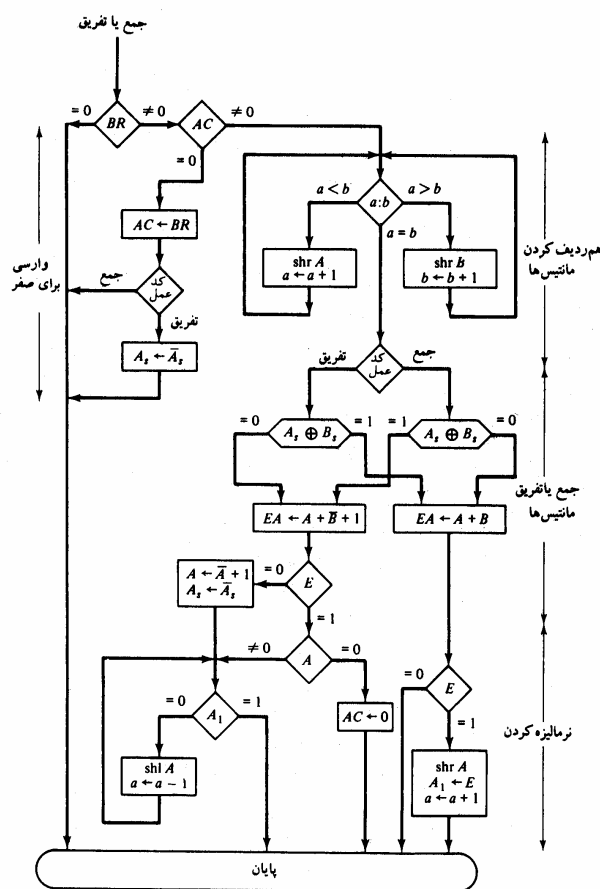
یادداشت:

.....

.....

.....

.....



### ضرب:

ضرب دو عدد ممیز شناور با ضرب مانتیس‌ها انجام می‌گردد و نماها با هم جمع می‌شوند. در اینجا لزومی به مقایسه نماها و تنظیم مانتیس نیست. ضرب مانتیس‌ها مشابه ضرب دو عدد ثابت در سیستم علامت مقدار انجام می‌گردد. فقط یک اختلاف وجود دارد و آن اینست که پس از ضرب، نیمه با اولویت بالاتر ذخیره می‌شود و از نیمه پایینی صرف نظر خواهد شد.

### مراحل ضرب:

- ۱- بررسی صفر بودن
- ۲- جمع نمودن نماها
- ۳- ضرب مانتیس‌ها
- ۴- نرمال کردن حاصل ضرب

### یادداشت:

.....

.....

.....

.....

## تقسیم ممیز شناور:

در تقسیم، مانیتیس‌ها تقسیم می‌گردند و نماها از هم کم خواهند شد. تقسیم مانیتیس‌ها مشابه حالت ممیز ثابت انجام می‌پذیرد. با این تفاوت که در این حالت مقسوم یک کلمه است که در AC قرار داده می‌شود. پس از اینکه عمل تقسیم انجام پذیرفت باید مقدار بایاس را به نمای حاصل اضافه کرد، زیرا مقدار بایاس هنگام کم کردن نماها حذف خواهد شد.

## مراحل تقسیم:

۱. تست وجود صفر

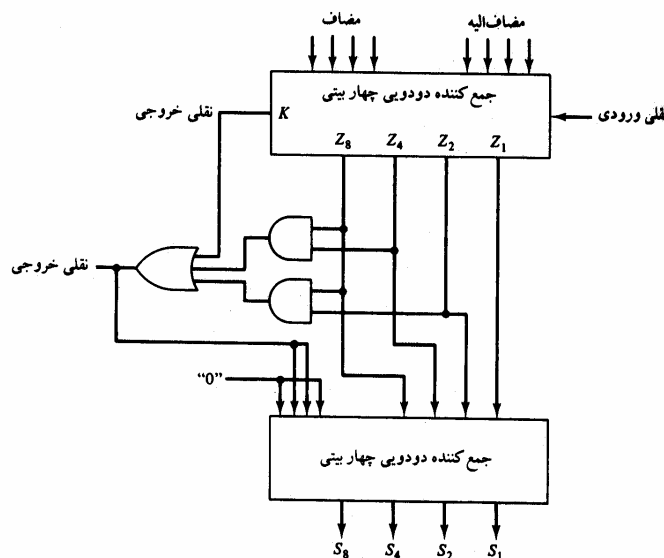
۲. تعیین علامت و دادن مقدار اولیه به ثباتها

۳. تنظیم نمودن مقسوم

۴. کم کردن نمادها

۵. تقسیم نمودن مانیتیس‌ها

**جمع کننده BCD:** جمع دو رقم دهدهی به شکل BCD حداکثر ۱۸ می‌باشد که با احتساب رقم نقلی حداکثر ۱۹ می‌شود. بدلیل اینکه هر رقم ورودی از ۹ بیشتر نمی‌باشد و حاصل جمع خروجی به صورت  $9+9+1=19$  خواهد بود. حال اگر از جمع کننده دودویی استفاده نمائیم و حاصل آن بیش از ۹ باشد باید آن را اصلاح نمائیم. برای اصلاح آن باید حاصل را با ۶ جمع کرد.



**تفریق BCD:** برای تفریق دو رقم در کد BCD بهترین روش اینست که رقم اول با مکمل ۱۰ رقم دوم جمع گردد. برای این حالت باید ابتدا مکمل ۹ عدد دوم یعنی B را مجاسه نمائیم و پس از آن ۱ را به آن اضافه کنیم. بدین منظور دو راه تصحیح برای پیدا کردن مکمل ۹ وجود دارد که به صورت زیر می‌باشد:

۱- عدد دودویی ۱۰۱۰ را به هر رقم ممتهم می‌افزاییم و از رقم نقلی بعد از هر جمع صرف نظر می‌نمائیم.

۲- عدد دودویی ۰۱۱۰ را به B می‌افزاییم و سپس حاصل را is complement می‌کنیم.

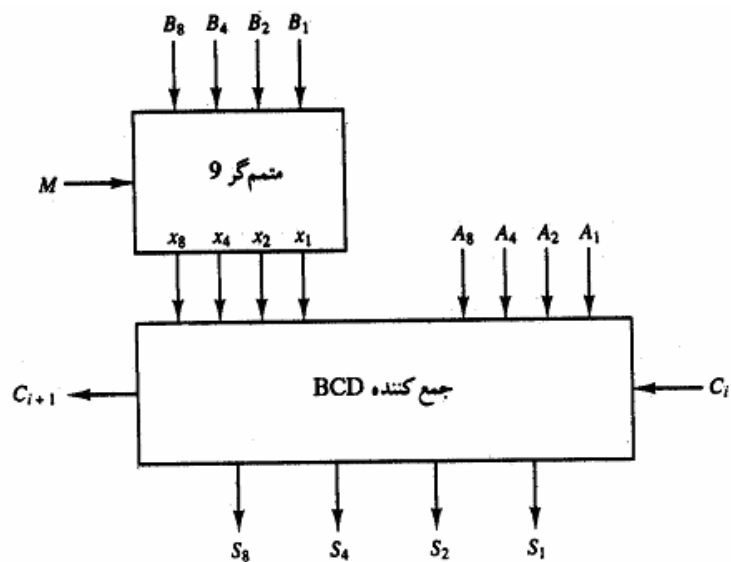
## یادداشت:

.....

.....

.....

.....



$A > B$	$A = 5$		
0101	$B = 3$		
<u>1101</u>		$C = 1$	
0010		$S \oplus V = 0$	
$\boxed{C=1}$	$\boxed{S=0}$	$\boxed{V=0}$	
<hr/>			
		$C = 1$	
$A = B$	$A = 5$		
	$B = 5$	$S \oplus V = 0$	
0101			
<u>1011</u>			
0000			
$\boxed{C=1}$	$\boxed{S=0}$	$\boxed{V=0}$	
<hr/>			
$A < B$	$A = 3$	$C = 0$	
	$B = 5$		
0011		$S \oplus V = 1$	
<u>1011</u>			
1110			
$\boxed{C=0}$	$\boxed{S=1}$	$\boxed{V=0}$	

$$\Rightarrow C = \overline{S \oplus V}$$

یادداشت:

.....

.....

.....

.....



برای انتخاب کوچکترین عدد مثبت مانتیس باید به صورت 0,10000 باشد، یعنی مانتیس باید نرمال باشد در غیر این صورت باید مانتیس را یک یا چند بیت به چپ شیفت دهیم از Exponent کم کنیم که با توجه به انتخاب کوچکترین مقدار exp مجاز به انجام این کار نمی‌باشیم.

### نکات تکمیلی

✓ اجرای این 3 دستور محتوای A و B را با هم عوض می‌کند :

$$\begin{aligned} \text{XOR } A, B \\ \text{XOR } B, A \\ \text{XOR } A, B \end{aligned} \begin{cases} A \geq B \text{ (S} \oplus \text{V)} = 0 \\ A < B \text{ (S} \oplus \text{V)} = 1 \end{cases}$$

✓ در تفریق A-B در سیستم مکمل 2، در صورتی  $CF = 1$  می‌شود که  $A \geq B$  باشد.

✓ در روش Booth تعداد Shiftها برابر تعداد بیت‌های ضرب کننده می‌باشد تعداد جمع و تفریق نیز برابر تعداد تغییر بیت در آن می‌باشد.

✓ اگر ضرب کننده  $Q_R$  باشد قرار می‌دهیم  $Q_{n+1} = 0$  حال در این  $x+1$  رقم از چپ به راست حرکت می‌کنیم. هر تغییر از 1 به 0 نمایانگر Sub، هر تغییر از 0 به 1 نمایانگر Add می‌باشد و

$$t_{\text{Sub}} = t_{\text{Add}} + t_{\text{Complement}}$$

✓ Divide By Zero باعث Set شدن VF (پرچم سرریز) می‌شود.

✓ در اعمال چهارعمل اصلی روی مانتیس اعداد ممیز شناور، فقط در عمل ضرب است که سرریز رخ نمی‌دهد.

✓ وقتی می‌خواهیم دو عدد را در سیستم 2's Complement ضرب کنیم، اگر ضرب کننده منفی بود آن را مثبت کرده و ضرب را با گسترش علامت انجام می‌دهیم و حاصل ضرب را در صورت لزوم تغییر علامت می‌دهیم.

✓ در ضرب به روش آرایه‌ای اگر مضروب k بیتی و ضرب کننده j بیتی باشد  $\begin{pmatrix} k \\ j \end{pmatrix}$ ، چون j سطح داریم پس :

$$\text{Delay} += j * \{\text{Max}(\text{carry}, \text{sum})\}$$

در اولین سطح a.b داریم پس  $\text{Delay} += \text{Delay And}$ ، در آخرین سطح یک جمع کننده موازی  $K-1$  بیتی داریم که از یک H.A، K-2 تا F.A تشکیل شده. پس جمع اخیر عبارتست از:

$$\text{Delay} = T_{\text{Gate}} + (K-2) * T_{\text{carry}} + j * \text{Max}\{T_{\text{carry}}, T_{\text{sum}}\}$$

حل: در هر سه روش عمل شیفت n تایی وجود دارد که یکسان است بنابراین در مقایسه شرکت نمی‌دهیم:

زمان ( $\mu s$ )	عمل انجام شده	روش
$2 * n + n / 2 = 5n / 2$	n مقایسه و $n/2$ تفریق	مقایسه
$1 * n + 1 * n / 2 = 3n / 2$	n تفریق و $n/2$ جمع	ترمیم
$n / 2 * 1 + n / 2 * 1 = n$	مقایسه به کمک تفریق پس $n/2$ تفریق و $n/2$ جمع	غیر ترمیم

✓ در جمع کننده BCD، تأخیر جمع کننده (sum) برابر  $16t$  و تأخیر رقم نقلی  $10t$  است و اگر n تا جمع کننده BCD با هم سری شوند تأخیر نهایی برابر است با:

$$(n-1)10t + 16t$$

یادداشت:

.....

.....

.....

.....

## تست‌ها

۱- در یک سیستم اعداد ممیز شناور 16 بیتی هستند. مقدار عددی یک عدد ممیز شناور با نمایش بیتی  $b_{15} \cdots b_1 b_0$  برابر است با: (سال ۸۶)

$$\left(b_{15} - \frac{1}{4}\right) \times M \times 2^s$$

$$S = \sum_{i=10}^{14} b_i (-2)^{i-10}, \quad M = \sum_{i=0}^9 b_i \left(-\frac{1}{2}\right)^{10-i}$$

کوچکترین و بزرگترین عدد قابل نمایش در این سیستم به ترتیب برابرند با:

$$(1) \quad \left(2^{18} + 2^{16} + 2^{14} + 2^{12} + 2^{10}\right), \quad -\left(2^{17} + 2^{15} + 2^{13} + 2^{11}\right)$$

$$(2) \quad \left(2^{18} + 2^{16} + 2^{14} + 2^{12} + 2^{10}\right), \quad -3\left(2^{18} + 2^{16} + 2^{14} + 2^{12} + 2^{10}\right)$$

$$(3) \quad 3\left(2^{17} + 2^{15} + 2^{13} + 2^{11}\right), \quad -\left(2^{17} + 2^{15} + 2^{13} + 2^{11}\right)$$

$$(4) \quad 3\left(2^{17} + 2^{15} + 2^{13} + 2^{11}\right), \quad -3\left(2^{18} + 2^{16} + 2^{14} + 2^{12} + 2^{10}\right)$$

۲- در یک کامپیوتر اعداد ممیز شناور به صورت 32 بیتی ذخیره می‌شوند. مقدار عددی رشته 32 بیتی  $b_{31}b_{30} \dots b_1b_0$  برابر است با:

$$2^{E-64} \times (2 \times b_{31} - 1) \times \sum_{i=0}^{23} (\bar{b}_i \times 2^{i-12})$$

$$E = \sum_{i=24}^{30} (2^{i-24} \times b_i)$$

مقدار عددی کوچکترین عدد مثبت و کوچکترین عدد منفی قابل نمایش چقدر است؟ (سال ۸۸)

$$(1) \quad -2^{51}, 2^{-64} \quad (2) \quad -2^{75}, 2^{-76} \quad (3) \quad 2^{51} - 2^{75}, 2^{-76} \quad (4) \quad -2^{63}, 2^{-64}$$

۳- در یک سیستم نمایش اعداد ممیز شناور با قالب زیر، نمایش عدد  $7\frac{1}{7}$  چگونه خواهد بود؟ S بیت علامت، M مانتیس و E توان

عدد را نشان می‌دهد. فرض کنید مانتیس عدد به صورت نرمال  $1 \leq M < 2$  است و عدد یک برای مانتیس تلویحی است و نمایش

داده نمی‌شود و توان عدد به صورت Biased - 127 نمایش داده می‌شود. (سال ۸۶)

1	8	23
S	E	M

40 E49249 (۴)

40F24924 (۳)

00F24924 (۲)

00E49249 (۱)

یادداشت:

.....

.....

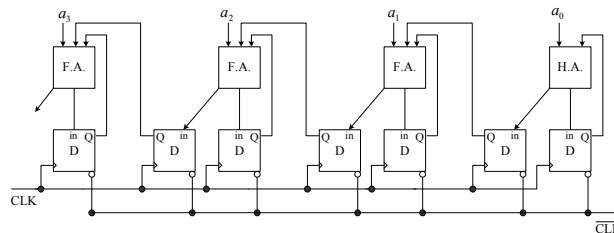
.....

.....

۴ - دستورات یک کامپیوتر دو آدرس یک بایتی و سه بایتی هستند. ماشین دارای 4 ثبات همه منظوره است و شیوه نشاندگی حافظه مستقیم (direct) است. تعداد دستورات سه بایتی 32 عدد است. طول ثبات MAR و تعداد دستورات یک بایتی به ترتیب می‌توانند برابر کدام باشند؟ (سال ۸۶)

- (۱) 16 یا MAR=9 و 16 = تعداد دستورات یک بایتی  
(۲) 16 یا MAR=9 و 8 = تعداد دستورات یک بایتی  
(۳) 17 یا MAR=8 و 8 = تعداد دستورات یک بایتی  
(۴) 17 یا MAR=8 و 16 = تعداد دستورات یک بایتی

۵ - مدار زیر مفروض است:



۶ - کدام گزینه صحیح است؟ (سال ۸۵)

- (۱) از این مدار می‌توان بر جمع  $n$  عدد 4 بیتی در  $n$  کلاک استفاده کرد.  
(۲) از این مدار می‌توان برای جمع  $n$  عدد 4 بیتی در  $2n$  کلاک استفاده کرد.  
(۳) از این مدار می‌توان برای جمع  $n$  عدد 4 بیتی در  $n+3$  کلاک استفاده کرد.  
(۴) از این مدار می‌توان برای جمع  $n$  عدد 4 بیتی در  $2n+3$  استفاده کرد.

۷ - در طراحی بخش محاسبات کامپیوتر پایه 32 بیتی از الگوریتم ضرب بوث (Booth) استفاده شده است. در این الگوریتم حداکثر تعداد جمع و حداکثر تعداد تفریق در هنگام ضرب دو عدد چقدر خواهد بود؟ (سال ۸۷)

- (۱) حداکثر تعداد جمع = 16  
(۲) حداکثر تعداد جمع = 15  
(۳) حداکثر تعداد جمع = 31  
(۴) حداکثر تعداد جمع = 32  
حداکثر تعداد تفریق = 16  
حداکثر تعداد تفریق = 16  
حداکثر تعداد تفریق = 31  
حداکثر تعداد تفریق = 32

۸ - در ضرب  $n \times n$  bits به روش Booth حداکثر تعداد عمل جمع یا تفریق برابر کدام است؟ (سال ۸۸)

- (۱)  $n$   
(۲)  $n-1$   
(۳)  $\left\lceil \frac{n}{2} \right\rceil$   
(۴)  $\left\lfloor \frac{n}{2} \right\rfloor$

یادداشت:

.....

.....

.....

.....

## پاسخ‌ها

۱ - گزینه ۴ درست است.

$b_x$  ها هر کدام می‌توانند مثبت یا منفی باشند.

$$M = b_0 \left(-\frac{1}{2}\right)^{10} + b_1 \left(-\frac{1}{2}\right)^9 + b_2 \left(-\frac{1}{2}\right)^8 + \dots + b_9 \left(-\frac{1}{2}\right)^1$$

اگر بخواهیم  $|M|$  بزرگ شود یا باید  $b_i$  های فرد را یک و  $b_i$  زوج را صفر و یا برعکس انتخاب کنیم که مشخص است اگر فردها یک باشند بزرگتر می‌شود:

$$M = -\frac{1}{2} - \frac{1}{2^3} - \frac{1}{2^5} - \frac{1}{2^7} - \frac{1}{2^9}$$

و  $b_{15}$  را برابر یک در نظر می‌گیریم

$$S = b_{10}(-2)^0 + b_{11}(-2)^1 + b_{12}(-2)^2 + b_{13}(-2)^3 + b_{14}(-2)^4 = b_{10} - 2b_{11} + 4b_{12} - 8b_{13} + 16b_{14}$$

که برای این که عدد نهایی تا حد امکان بزرگ شود باید  $b_i$  ها با اندیس فرد صفر باشند:

$$= \frac{3}{4} \times 2^{21} (-1) \left( \frac{1}{2} + \frac{1}{2^3} + \dots + \frac{1}{2^9} \right) = -3 \times 2^{19}$$

و برای بزرگترین عدد مثبت کافی است  $M$  را با ضرایب با اندیس زوج بنویسیم.

۲ - گزینه ۴ درست است.

هر کلاک پردازنده باید حداقل 10ns باشد تا عمل جمع یا شیفت در یک کلاک انجام شود. مضروب فیه عدد 1011 می‌باشد که به تعداد بیت‌های آن عمل شیفت و به تعداد بیت‌های 1 آن به عمل جمع نیاز داریم. پس در کل به 70ns زمان نیاز داریم.

۳ - گزینه ۴ درست است.

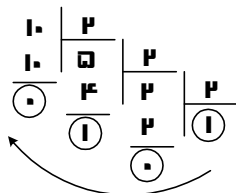
عدد به صورت مقابل می‌شود (بدون بایاس):  $(-1)^s \times o.m \times 2^E$

چون عدد مثبت است، بیت علامت صفر است.

عدد  $\frac{1}{7}$  به باینری تبدیل می‌شود:

**نکته:** برای تبدیل یک عدد به باینری، آن را متوالیا تقسیم بر ۲ می‌کنیم و باقیمانده‌ها را به ترتیب معکوس می‌نویسیم.

**مثال:** عدد 10 در مبنای ۲ می‌شود:



**یادداشت:**

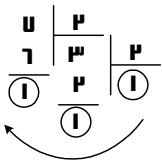
.....

.....

.....

.....

قسمت اعشار عدد را متوالیا ضرب در دو می‌کنیم، ببینیم چه زمان به یک (عدد صحیح) می‌رسیم. ابتدا ۷ را به باینری تبدیل می‌کنیم:



حال  $\frac{1}{7}$  را باینری می‌کنیم:

$$\frac{1}{7} \times 2 = \frac{2}{7} < 1 \Rightarrow 0$$

$$\frac{2}{7} \times 2 = \frac{4}{7} < 1 \Rightarrow 0$$

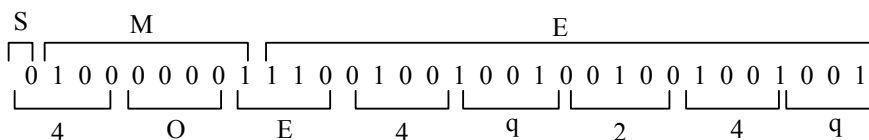
$$\frac{4}{7} \times 2 = \frac{8}{7} > 1 \Rightarrow 1 \quad \frac{8}{7} - 1 = \frac{1}{7} \text{ می‌ماند}$$

$$7 \frac{1}{7} = (111.\overline{001})_2 \quad \text{پس:}$$

اگر آن را نرمال کنیم، می‌شود  $1/11001 \times 2^2$

پس توان بایاس 127 می‌شود  $2+127=129$  و  $129 = (1000000)_2$  یعنی  $M.E=10000001$  هم می‌شود  $11001$  که 001 چندین بار تکرار می‌شود تا 23 بیت پر شود.

نتیجه می‌شود:



\* هر ۴ بیت در مبنای ۲ را با هم انتخاب می‌کنیم تا مبنای ۱۶ (هگزا دسیمال) آن به دست آید.

۴ - گزینه ۲ درست است.

ارتباط حافظه با محیط خارج به کمک دو سیگنال Read و Write و دو ثبات MAR و MBR است. اندازه‌ی MBR یا Memory Buffer

Register برابر «تعداد کلمات حافظه»  $\log_2$  است و اندازه MAR یا Memory Address Register برابر تعداد بیت‌های شامل در هر کلمه است.

۵ - گزینه ۳ درست است.

شکل سؤال، اعداد چهار بیتی را به صورت پایپلین جمع می‌کند. طبق قانون پایپلین، زمان برابر  $4+(n-1)$  کلاک زمان صرف می‌شود. یعنی گزینه ۳ درست است.

یادداشت:

.....

.....

.....

.....

۶ - گزینه ۱ درست است.

در الگوریتم Booth هرگاه 10 تفریق می‌کنیم و هرگاه 01 ببینیم جمع می‌کنیم. پس اگر عدد ۳۲ بیتی به ترتیب  $\underbrace{101010\dots10}_{32}$  باشد،

16 تا جمع و 16 تا تفریق خواهیم داشت.

۷ - گزینه ۱ درست است.

در صورتی که به صورت  $\underbrace{1010\dots101}_n$  باشد به ازای هر 10 یک عمل تفریق و به ازای هر 01 یک عمل جمع صورت می‌پذیرد. در ضمن

یک صفر یک در ابتدای رشته مذکور برای محاسبه اضافه می‌کنیم یعنی به تعداد n عدد جمع یا تفریق نیاز داریم. بنابراین گزینه ۱ درست است.

یادداشت:

.....

.....

.....

.....

## فصل پنجم

### سازمان ورودی و خروجی (Input/Output)

**واسط ورودی-خروجی:** واسطه‌ای است که روش تبادل اطلاعات و انتقال داده‌ها بین حافظه و دستگاه‌های ورودی خروجی را فراهم می‌نماید.

**گذرگاه I/O :** برای انتقال اطلاعات بین پردازنده و وسایل جانبی مورد استفاده قرار می‌گیرد، گذرگاه I/O از خطوط داده، آدرس و خطوط کنترل بوجود آمده است. به عبارت دیگر این مجموعه خطوط در کنار یکدیگر می‌توانند گذرگاه I/O تشکیل دهند. همزمان با وجود آدرس روی خطوط آدرس، پردازنده یک کد عملیات را روی خطوط کنترل قرار می‌دهد. و واسط این کدها را گرفته و به آنها پاسخ می‌دهد. این کدها فرمان I/O نامیده می‌شوند. انواع این فرمان‌ها عبارتند از:

- فرمان کنترلی : برای فعال کردن دستگاه جانبی و اطلاع دادن به آن که چه کاری را باید انجام دهد مورد استفاده قرار می‌گیرد
- فرمان وضعیت : برای بررسی حالت‌های مختلف دستگاه جانبی، یا یک واسط مورد استفاده قرار می‌گیرد.
- فرمان داده خروجی : سبب می‌شود تا مدار واسط آنچه را که روی I/O Bus است به یکی از ثبات‌های خود انتقال دهد.
- فرمان داده ورودی : یک فرمان ورود داده عکس خروجی داده می‌باشد و باعث می‌شود که واسط داده‌ای را که در گذشته از دستگاه جانبی دریافت کرده با این توضیح که در ثبات آن موجود بوده است، بر روی خطوط داده از I/O Bus قرار دهد.

برای ارتباط گذرگاه I/O و گذرگاه حافظه سه روش موجود می‌باشد :

۱- از دو گذرگاه مجزا یکی برای I/O و دیگری برای حافظه استفاده شود.

۲- برای هر دو بخش حافظه و I/O از یک گذرگاه مشترک استفاده گردد و هر یک دارای خطوط کنترل مجزا باشند.

۳- برای I/O و حافظه از یک گذرگاه مشترک استفاده می‌نمائیم.

**یادداشت:**

.....

.....

.....

.....

## شیوه‌های انتقال داده

### ۱. Programmed I/O :

در این روش تمام عملیات I/O لازم برای انتقال اطلاعات کاملاً توسط CPU کنترل می‌گردد و دستگاه I/O ارتباط مستقیم با حافظه نخواهد داشت و برای انتقال داده از I/O به حافظه نیاز به اجرای یک یا چند دستور CPU می‌باشد. CPU از طریق واسط ورودی - خروجی به I/O متصل گردیده و داده‌ها را از واسط گرفته و می‌نویسد.

### ۲. Interrupt Initiated I/O :

در این روش I/O دسترسی مستقیم به حافظه ندارد و برای انتقال از I/O به حافظه CPU باید چندین دستور، از جمله یک دستور ورودی برای انتقال داده از I/O به CPU و یک دستور ذخیره سازی برای انتقال داده از CPU به حافظه را اجرا نماید.

بدست آوردن آدرس I/O Service Routine به دو صورت امکان پذیر است:

الف) Non Vectored Interrupt : آدرس پرش هنگام وقوع وقفه در حافظه مکان ثابتی را دارا است.

ب) Vectored Interrupt : منبعی که وقفه را می‌دهد، اطلاعات مرتبط با محل پرش را تعیین می‌کند که این اطلاعات باعث ایجاد Vectored Interrupt می‌گردد.

## آشنایی با Interruptها (وقفه‌ها)

وقفه برنامه‌ای است که از چند دستورالعمل تشکیل شده است که یک هدف را دنبال می‌نماید. در واقع وقفه‌ها برنامه‌هایی حاضر و آماده می‌باشند که باید یک عمل خاصی را اجرا نمایند.

دو نوع وقفه وجود دارد:

۱. وقفه‌های سخت افزاری، برنامه‌هایی هستند که توسط Cpu اجرا می‌گردند و اجرای آنها به دریافت سیگنال بوسیله Cpu بستگی دارد.

۲. وقفه‌های نرم افزاری

نکته اینکه پالس‌هایی که برای انجام وقفه‌ها می‌باشند از مسیر IRQ به CPU انتقال می‌یابند.

**تذکر:** هر گاه وقفه‌ای را از CPU تقاضا نمائیم اجرای برنامه‌های خود را متوقف نموده و دستورات درون وقفه را تا رسیدن به دستور IRET اجرا می‌نماید و پس از آن به درون دستورات در حال اجرا باز می‌گردد.

### اولویت‌بندی وقفه: سه روش دارد که عبارتند از:

(a) **Polling :** این روش نرم افزاری می‌باشد و با آمدن هر وقفه به محل ثابتی در حافظه رفته و به ترتیب اولویت منابع وقفه را مورد بررسی قرار می‌دهد و با شناخت اینکه کدام مورد باعث وقفه گردیده است برنامه سرویس مرتبط با آن را اجرا می‌نماید.

(b) **Daisy chain :** این روش سخت افزاری می‌باشد و در آن دستگاه‌هایی که امکان تقاضای وقفه دارند به طور متوالی به یکدیگر متصل می‌شوند. به گونه‌ای که بر حسب اولویت‌ها چیدمان می‌گردند و در نتیجه اولویت‌های بالاتر در ابتدا و اولویت‌های پایین‌تر در انتها قرار خواهند گرفت.

**یادداشت:**

.....

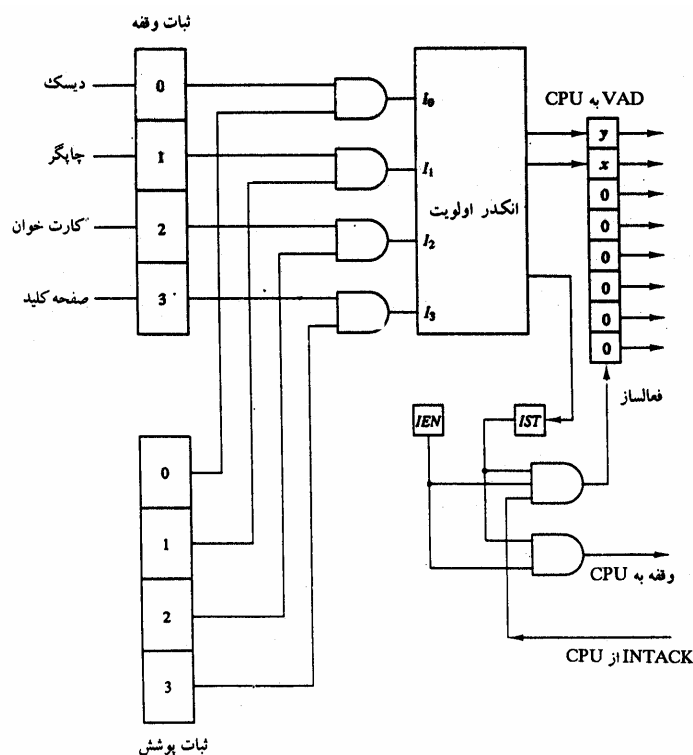
.....

.....

.....



**Parallel (C):** این روش از ثباتی استفاده می‌نماید که بیت‌های آن بطور جداگانه توسط سیگنال وقفه هر وسیله صفر می‌گردد و اولویت آنها با توجه به محل قرار گرفتن بیت‌ها در ثبات تعیین می‌شود.



### ۳. Direct Memory Access :

نیاز به قسمتی در سیستم احساس می‌شود که فارغ از CPU ارتباط حافظه و I/O را فراهم آورد. این تکنیک DMA نامیده می‌شود و قسمت کنترل کننده DMA controller نام دارد.

### کنترل کننده DMA:

کنترل کننده DMA برای ارتباط با CPU و وسیله جانبی به مدارات معمول واسطه نیاز دارد. بعلاوه یک ثبات آدرس و یک ثبات Cycle stealing و یک مجموعه خطوط آدرس نیز مورد استفاده می‌باشد. ثبات آدرس و خطوط آدرس برای ارتباط مستقیم با حافظه مورد استفاده قرار می‌گیرند. ثبات شمارش کلمه نیز تعداد کلماتی را که باید انتقال یابند را مشخص می‌نماید. انتقال داده‌ها به طور مستقیم تحت نظارت DMA میان وسیله جانبی و حافظه صورت می‌پذیرد.

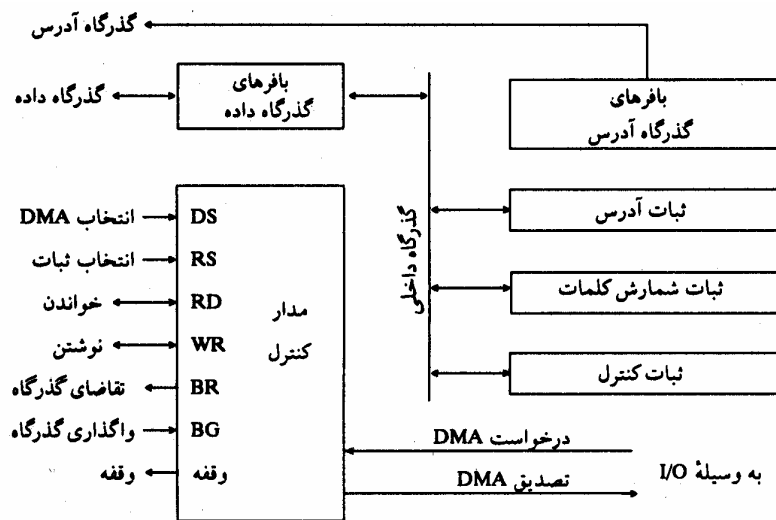
یادداشت:

.....

.....

.....

.....



انتقال DMA علاوه بر انتقال سریع اطلاعات بین دیسک‌ها و حافظه، بهنگام‌کردن تصویر در پایانه محاوره‌ای با استفاده از حافظه تصویری است. محتوای حافظه به طور دوره‌ای به صفحه نمایش ارسال می‌شود و برای تغییر تصویر، حافظه تصویر تغییر می‌یابد.

یادداشت:

.....

.....

.....

.....

## تست‌ها

۱ - کدام گزینه در مورد روش‌های I/O صحیح نیست؟ (سال ۸۵)

- ۱) روش Programmed I/O پردازنده اصلی را درگیر عملیات I/O می‌کند.
- ۲) روش Interrupted I/O پردازنده اصلی را درگیر عملیات I/O می‌کند.
- ۳) روش DMA I/O پردازنده اصلی را درگیر عملیات I/O می‌کند (در حین I/O).
- ۴) روش I/O با استفاده از هم پردازنده خاص I/O امکان نوشتن برنامه‌های مختلط از دستورات پردازنده اصلی و هم پردازنده I/O را می‌دهد.

۲ - کدام یک از عبارات زیر صحیح می‌باشد؟ (سال ۷۸)

- ۱) یک سیستم کامپیوتر با توانایی وقفه از نوع فقط non- vectored نمی‌تواند توانایی DMA داشته باشد.
  - ۲) یک سیستم کامپیوتر با توانایی وقفه از نوع فقط vectored مجبور است که در (Interrupt service ISR routine) خود قبل از شروع به ارائه سرویس، دستگاه (device) اینترپت‌دهنده را به کمک Io polling تشخیص بدهد.
- الف) گزینه ۱      ب) گزینه ۲      ج) هر دو      د) هیچکدام

## پاسخ‌ها

۱ - گزینه ۳ درست است.

DMA بر خلاف دو روش دیگر، CPU را درگیر عملیات I/O نمی‌کند، پس گزینه ۳ جواب سؤال می‌باشد.

۲ - گزینه ۴ درست است.

## یادداشت:

.....

.....

.....

.....

## فصل نهم

### سازمان حافظه (Memory)

#### سلسله مراتب حافظه:

سلسله مراتب حافظه به خاطر سرعت بخشیدن دسترسی به حافظه با حداقل هزینه بوجود آمده است و عبارت است از :

۱. Cache Memory

۲. Main Memory

۳. Auxiliary Memory

#### حافظه اصلی:

واحد ذخیره سازی مرکزی در یک سیستم کامپیوتری می باشد و نسبتاً سریع هستند که دو نوع هستند :

۱. RAM : این حافظه هم امکان Read دارد و هم امکان Write و به دو صورت Static و dynamic می باشد.

۲. ROM : این حافظه دارای دسترسی تصادفی می باشد و فقط امکان Read را داراست.

#### اتصال حافظه به CPU:

۱. تراشه های RAM و ROM از طریق گذرگاه داده و آدرس به CPU متصل می شوند.

۲. بیت های کم ارزش تر در خطوط آدرس، بیت های درون یک تراشه حافظه و بیت های با ارزش، بیشتر یک تراشه از چندین تراشه حافظه را انتخاب می کنند.

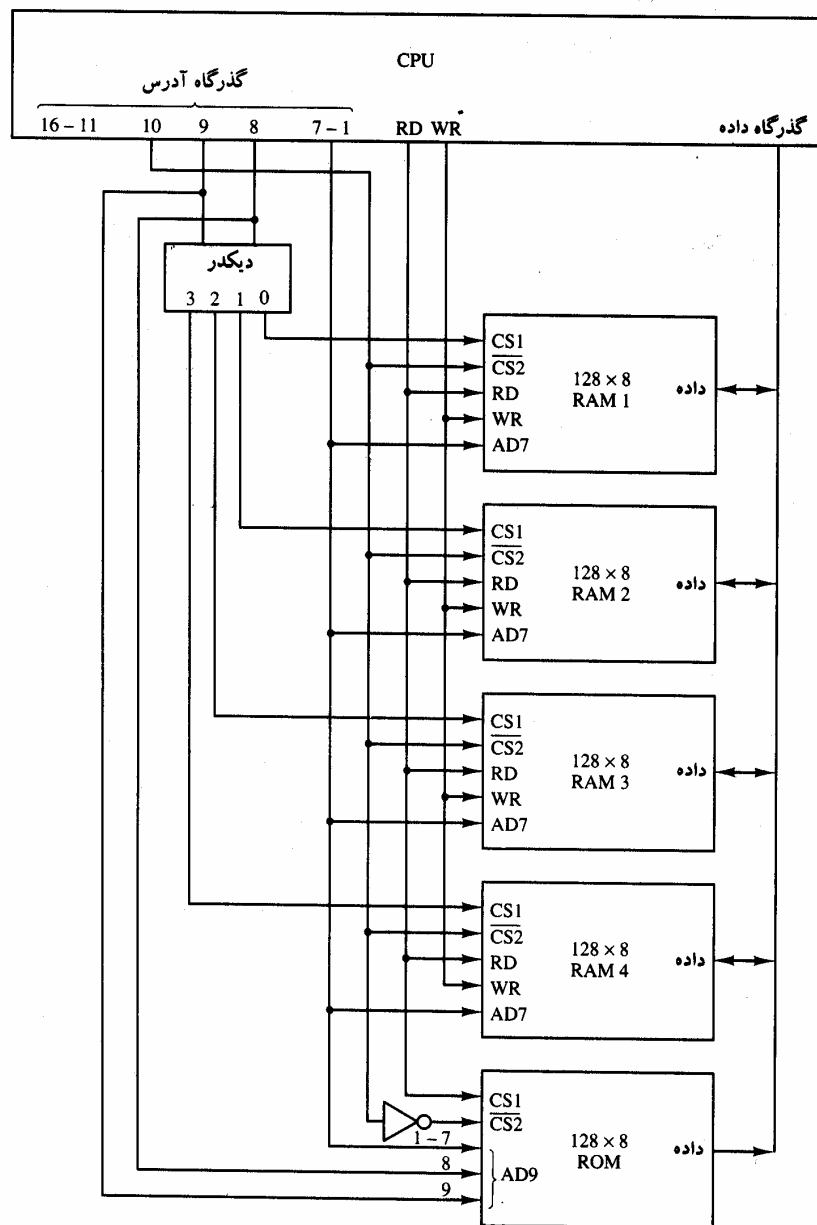
#### یادداشت:

.....

.....

.....

.....



### حافظه‌های جانبی:

۱. سازماندهی اطلاعات بر روی نوارهای مغناطیسی
۲. سازماندهی دیسک‌های سخت افزاری

یادداشت:

.....

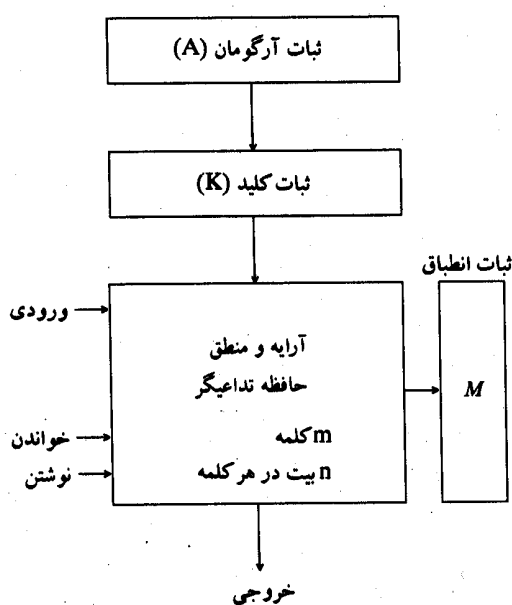
.....

.....

.....

### حافظه انجمني:

- ✓ دسترسی به این گونه حافظه ها بر اساس محتوای خود داده صورت می گیرد نه بر اساس آدرس آنها.
- ✓ این گونه حافظه ها به حافظه های قابل آدرس دهی با محتوا (CAM) هم نامیده می شوند.
- ✓ سازماندهی سخت افزاری



### عملکرد حافظه انجمني:

- هر کلمه در ثبات CAM به طور موازی با محتویات ثبات A (Argument Register) مقایسه می شود.
- اگر در CAM,  $\text{word}[i]=A$ , آنگاه  $M[i]=1$ .
- تمام کلماتی که در CAM مقدار  $M[i]=1$  آن به صورت ترتیبی از CAM خوانده می شود.
- از ثبات K برای ماسک کردن یک فیلد و یا کلید در ثبات A استفاده می شود.

### امکانات یک حافظه انجمني:

- ۱- خواندن یک کلمه از حافظه
- ۲- نوشتن یک کلمه در حافظه
- ۳- پیدا نمودن کلمه ای که دارای داده ای خاص است.
- ۴- نوشتن در کلمه ای که دارای داده ای خاص است.

### یادداشت:

.....

.....

.....

.....

## حافظه نهان:

- طبق اصل مراجعات محلی به حافظه دسترسی به حافظه در بازه‌های زمانی معین متمایل به سلول‌های مجاور حافظه است. این نواحی حافظه را به زیر مجموعه‌ای تقسیم می‌کند که در طول زمان این زیر مجموعه‌ها با زیر مجموعه‌های جدید معاوضه می‌گردند.
- محلی بودن مراجعات از دو جنبه قابل بررسی می‌باشد:
  ۱. محلی بودن موقت: طبق این اصل اطلاعاتی که در آینده نزدیک مورد استفاده قرار خواهند گرفت به احتمال زیاد هم اکنون مورد استفاده واقع شده‌اند.
  ۲. محلی بودن فیزیکی: طبق این اصل اگر یک کلمه مورد دستیابی قرار گیرد، کلمه‌های مجاور با این کلمه در آینده نزدیک مورد دستیابی قرار خواهند گرفت.
- حافظه‌های نهان، حافظه‌های سریع و کم حجمی می‌باشند که وظیفه آنها ذخیره اطلاعاتی است که ممکن است به زودی مورد استفاده واقع شوند.

## عملکرد حافظه نهان:

کلیه دسترسی‌ها به حافظه در ابتدا به طرف حافظه نهان ارسال می‌گردد، اگر کلمه مورد نظر در حافظه نهان یافت شود حافظه مورد دستیابی قرار می‌گیرد، در غیر اینصورت بلوکی که کلمه مورد نظر در آن قرار دارد از حافظه اصلی جایگزین یک بلوک در حافظه نهان می‌گردد. به عبارت دیگر عملکرد حافظه نهان توسط معیار hit مشخص می‌گردد. اگر هنگام رجوع به حافظه نهان کلمه مورد نظر یافت شود hit انجام پذیرفته و در غیر این صورت Miss انجام می‌پذیرد. بنابراین خواهیم داشت:

$$t_a = t_c + (1 - h) t_m$$

$$t_a = t_c + (1 - h_1) t_{m_1} + (1 - h_1)(1 - h_2) t_{m_2}$$

که در آنها hit درصدی از کل دسترسی‌های به حافظه است که از طریق حافظه نهان برآورده می‌شود،  $T_a$  زمان موثر دسترسی به یک کلمه از حافظه در سیستمی که از حافظه نهان استفاده می‌کند می‌باشد و  $T_c$  زمان دسترسی به حافظه نهان است و  $T_m$  نیز زمان دسترسی به حافظه اصلی است.

## نگاشت:

انتقال داده‌ها از حافظه اصلی به حافظه نهان را نگاشت گویند که سه نوع می‌باشد:

۱. نگاشت انجمنی:

✓ هر بلوک از حافظه نهان می‌تواند هر بلوک از حافظه اصلی را ذخیره کند به عبارت دیگر دارای قابلیت انعطاف می‌باشد.

✓ جدول نگاشت در حافظه انجمنی به راحتی پیاده سازی می‌شود در نتیجه سریع و گران قیمت می‌باشد.

✓ جدول نگاشت، آدرس کلمه حافظه و محتویات موجود در آن را ذخیره می‌نماید.

## یادداشت:

.....

.....

.....

.....

۲. نگاشت مستقیم :

✓ هر بلوک حافظه تنها در یک مکان از حافظه نهان می تواند ذخیره گردد.

✓ یک آدرس n-بیتی حافظه اصلی دارای دو قسمت به صورت زیر می باشد.

▪ k بیت برای Index.

▪ n-k بیت برای Tag.

✓ از آدرس k بیتی برای دسترسی به حافظه نهان و از آدرس n بیتی برای دسترسی به حافظه اصلی می توان استفاده نمود.

۳. نگاشت انجمنی مجموعه ای :

✓ هر کلمه ای که در حافظه اصلی وجود دارد دارای مجموعه ای از مکانها در حافظه نهان می باشد که می تواند در آنها قرار گیرد.

### نوشتن در حافظه نهان:

• اگر کلمه ای که می نویسیم درون حافظه نهان وجود داشت Hit آن کلمه را به صورت موازی در حافظه نهان و حافظه اصلی می نویسیم.

• اگر کلمه ای که می نویسیم درون حافظه نهان وجود نداشت Miss آن کلمه را فقط در حافظه نهان می نویسیم.

• اگر در هنگام خواندن یک Miss اتفاق افتاد بلوکی که درون cache قرار ندارد دوباره درون cache نوشته می شود. نوشتن در حافظه نهان به دو صورت زیر می باشد :

۱. از طریق through-write که در این روش محتویات Ram و Cache همیشه برابر می باشد.

۲. از طریق back-write که در این روش تا زمانی که کلمه در حافظه نهان وجود داشته باشد در حافظه اصلی درج نمی گردد و زمانی که کلمه از حافظه نهان حذف گردید حافظه اصلی به روز می گردد.

### فرمول های زمان متوسط دسترسی:

$$۱) t_a = t_c + (1-h)t_m$$

$$۲) t_a = h.t_c + (1-h)t_m$$

$$۳) t_a = t_{ci} + ht_{cr} + (1-h)t_m$$

اگر نوشتن مد نظر باشد و روش writ-through را مورد استفاده قرار دهیم داریم:

$$۴) t_a = t_c + (1-h)t_m + \omega(t_m - t_c)$$

$$\omega = \frac{\text{نوشتن ها}}{\text{کل مراجعات}}$$

اگر سیاست نوشتن مد نظر باشد و روش write-back مورد استفاده قرار دهیم خواهیم داشت:

$$۵) t_a = t_c + (1-h)*t_m * 2$$

در سیاست write-back اگر احتمال تغییر داده در حافظه نهان وجود داشته باشد داریم:

$$۶) t_a = t_c + (1-h)t_m + \omega_b(1-h)t_m$$

یادداشت:

.....  
 .....  
 .....  
 .....



## تست‌ها

۱ - زمان دستیابی به یک بایت از حافظه اصلی 12.5 نانوثانیه است. زمان دستیابی به یک حافظه نهان (cache) برابر با یک نانو ثانیه می‌باشد. حال اگر از بلوک‌های 8 بایتی در cache استفاده شود، درصد موفقیت (hit) حافظه cache از چه مقدار باید حداقل بیشتر باشد تا استفاده از cache مقرون به صرفه باشد. (سال ۸۵)

(۱) 49.5% (۲) 50.5% (۳) 86.5% (۴) 88.5%

۲ - حافظه اصلی دارای حجم  $2^{20}$  بایت است. حافظه cache از نوع نگاشت مستقیم دارای  $2^{14}$  بایت می‌باشد. بلوک‌های حافظه 16 بایت است. خواندن اطلاعات از حافظه نهان یک نانوثانیه طول می‌کشد و خواندن هر بلوک از حافظه اصلی به چهل نانوثانیه نیاز دارد. رشته آدرس‌های زیر توسط پردازنده خوانده می‌شود و ابتدا حافظه نهان تهی است.

$01000_H, 01001_H, 01002_H, 59000_H, 5900F_H, 01003_H, 56780_H, 5678F_H, 56790_H, 56791_H$

متوسط زمان دستیابی به رشته آدرس‌های فوق چند نانو ثانیه است؟ (سال ۸۷)

(۱) 1 (۲) 11 (۳) 21 (۴) 40

۳ - خطوط آدرس یک پردازنده  $A_0, A_1, A_2, \dots, A_{15}$  هستند و از بلوک‌های حافظه به حجم 8k برای ساخت حافظه اصلی پردازنده استفاده کرده‌ایم. به فرض دسترسی به خانه‌های متوالی حافظه با فاصله آدرسی 3 ( $\text{stride} = 3$ ) و برگ برگ‌سازی حافظه با خطوط  $A_0, A_1, A_2, A_3$ ، حداکثر نسبت پهنای باند دسترسی در این سازمان حافظه به سازمان حافظه برگ برگ شده به صورت High-order چقدر است؟ (سال ۸۸)

(۱) 5.33 (۲) 5.5 (۳) 6 (۴) 8

۴ - کدام یک از گزینه‌های زیر در مورد روش‌های نگاشت در حافظه نهان صحیح نیست؟ (سال ۸۵)

- (۱) اگر اندازه set برابر یک بلوک باشد، روش نگاشت Associative - Set با روش Mapping - Direct معادل است.  
 (۲) اگر اندازه set برابر تعداد بلوک‌های حافظه نهان باشد، روش نگاشت Associative - Fully با روش Associative - Set معادل است.  
 (۳) در روش Mapping - Direct سرعت تشخیص حضور بلوک در حافظه نهان بیشتر از سرعت تشخیص در روش نگاشت Associative - Fully است.  
 (۴) در روش Associative - Fully سرعت تشخیص حضور بلوک در حافظه نهان از سرعت تشخیص در روش Associative - Set بیشتر است اما قیمت آن نیز گران تر است.

۵ - یک حافظه cache دستور با گنجایش 256 بایت را در نظر بگیرید که هر بلوک آن حاوی یک کلمه 32 بیتی است. فرض کنید یک حلقه حاوی 65 دستور متوالی برای 3 بار اجرا می‌شود. با فرض این که حافظه cache در شروع کار خالی باشد، Hit Rate در دو حالت زیر چقدر است؟ (هر دستور 32 بیت است) (سال ۸۶)

I - اگر cache از Direct Mapping استفاده کند.

II - اگر cache به صورت Fully Associative با سیاست جایگزینی LRU باشد.

(۱) I و 0 (II)  $\frac{42}{65}$  (۲) I  $\frac{42}{65}$ ، II 0 (۳) I و  $\frac{42}{65}$  (II)  $\frac{42}{65}$  (۴) I 0، II 0

یادداشت:

.....  
 .....  
 .....  
 .....

۶- در یک حافظه نهان (Cache Memory) که نرخ دسترسی موفق در آن برابر  $h$  است، چنانچه اندازه بلوک در آن  $B_c$  و اندازه بلوک در حافظه اصلی  $B_m$  بایت باشد و نرخ انتقال اطلاعات از حافظه نهان به پردازنده و از حافظه اصلی به حافظه نهان / پردازنده به ترتیب  $T_c\_transfer$  و  $T_m\_transfer$  باشد و همچنین زمان اولین دسترسی به حافظه نهان و اصلی به ترتیب  $T_c\_access$  و  $T_m\_access$  باشد، زمان متوسط انتقال  $B_c$  بایت اطلاعات درخواستی از سوی پردازنده چقدر خواهد بود؟ (سال ۸۵)

$$(1) h.(T_c\_access+B_c.T_c\_transfer)+(1-h).(T_c\_access+T_m\_access+B_m.T_m\_transfer)$$

$$(2) h.(T_c\_access+B_c.T_c\_transfer)+(1-h).(T_c\_access+T_m\_access+(B_c+B_m).T_c\_transfer)$$

$$(3) h.(T_c\_access+B_c.T_c\_transfer)+(1-h).(T_m\_access+B_c.T_c\_transfer+T_c\_access+B_m.T_m\_transfer)$$

$$(4) h(T_c\_access+B_c.T_c\_transfer)+(1-h).(T_c\_access+T_m\_access+B_m.T_m\_transfer+B_m/B_c.T_c\_transfer)$$

۷- کدام عبارت در مورد سلسله مراتب حافظه صحیح است؟ (سال ۸۵)

(۱) برای افزایش فضای آدرس دهی پردازنده از این روش استفاده می‌شود.

(۲) برای کم کردن فاصله‌ای که بین سرعت پردازنده و سرعت حافظه وجود دارد از این روش استفاده می‌شود.

(۳) وقتی باس داده ۸ بیتی است برای دستیابی به کلماتی که بزرگتر از یک بایت است از این روش استفاده می‌شود.

(۴) روش سلسله مراتب حافظه دیگر منسوخ شده است و هم اکنون از حافظه نهان (cache) استفاده می‌شود.

۸- یک سیستم حافظه با دو سطح cache را در نظر بگیرید. اطلاعات مورد نیاز را در جدول زیر پیدا می‌کنید. متوسط زمان دسترسی به این سیستم حافظه چقدر است؟ (سال ۸۶)

$L_1$  Hit Time = 1cycle

$L_2$  Miss Rate = 1%

$L_2$  Hit Time = 4 cycle

$L_2$  Miss Rate = 20%

$L_2$  Miss Penalty = 150cycle

(۴) هیچ کدام

(۳) 2.61

(۲) 2.45

(۱) 1.34

یادداشت:

.....  
 .....  
 .....  
 .....

## پاسخ‌ها

۱ - گزینه ۴ درست است.

برای مقرون به صرفه بودن باید زمان دسترسی به حافظه با کش کمتر از زمانی که کش وجود ندارد باشد. بدون کش فقط کلمه موردنظر خوانده می‌شود. در حالی که با کش ابتدا به کش رجوع می‌شود و در صورت عدم وجود یک بلاک حافظه که ۸ بایت است با زمان  $12.5 \times 8$  خوانده می‌شود.

پس:

زمان با وجود کش  $>$  زمان بدون وجود کش

$$12.5 > 1 + (1-h)(8 \times 12.5)$$

$$\rightarrow h > 1 - (11.5/100) \rightarrow h > 88.5\%$$

۲ - گزینه ۳ درست است.

آدرس هر کلمه ۲۰ بیتی که ۱۴ تای آن index و ۶ تای دیگر tag می‌باشد چون بلوک‌ها ۱۶ بیتی هستند. پس ۴ بیت برای کلمه‌های هر بلوک استفاده می‌شود. اگر بلوک دو آدرس یکسان بود ولی tag آن‌ها تفاوت داشت آن بلوک جایگزین بلوک قبلی می‌شود چون نگاشت مستقیم است.

6          14

tag	index
-----	-------

6          10          4

tag	block	word
-----	-------	------

tag	block	word	
000000	0100000000	0000	miss
000000	0100000000	0001	hit
000000	0100000000	0010	hit
010110	0100000000	0000	miss
010110	0100000000	1111	hit
000000	0100000000	0011	miss
010101	1001111000	0000	miss
010101	1001111000	1111	hit
010101	1001111001	0000	miss

۵ تا miss وجود دارد برای هر ارجاع، یک بار به cache مراجعه می‌شود و برای هر miss یک بار به حافظه مراجعه می‌شود

$$t = 5 \times 40 + 10 = 210$$

$$\text{متوسط زمان} = \frac{210}{10} = 21$$

یادداشت:

.....

.....

.....

.....

۳ - گزینه ۲ درست است.

$$\frac{2^{24}}{2^{13}} = 2^{11} = 2048 \text{ تعداد صفحه در حافظه مجازی:}$$

$$\frac{2^{19}}{2^{13}} = 2^6 = 64 \text{ تعداد بلوک در حافظه اصلی}$$

۴ - گزینه ۴ درست است.

۵ - گزینه ۲ درست است.

$$32 \text{ بیت یعنی } 4 \text{ بایت. پس تعداد بلوک‌های حافظه نهان برابر است با } \frac{256}{4} = 64$$

در روش Fully Associative به روش LRU، پس از آن جا که ۶۵ دستور و ۶۴ بلوک داریم، همواره miss رخ می‌دهد. یعنی پس از ۶۴ تا miss که cache پر شد، دستور ۶۵ ام به جای دستور اول می‌نشیند، سپس دستور اول می‌خواهد اجرا شود که چون در cache نیست، در cache به جای دستور دوم می‌نشیند و...

در روش direct mapping، پس از miss‌های دفعه‌ی اول، در دفعات بعد فقط ۲ تا miss داریم؛ وقتی دستور اول جایگزین دستور ۶۵ م می‌شود، و وقتی دستور ۶۵ م جایگزین دستور اول می‌شود پس:

$$\text{hit ratio} = \frac{0 + (65 - 2) + (65 - 2)}{3 \times 65} = \frac{42}{65}$$

۶ - گزینه ۱ درست است.

اگر hit صورت گیرد، به حافظه نهان دسترسی حاصل می‌شود و BC بایت به نرخ انتقال Tc\_transfer انتقال می‌یابد، که می‌شود Tc-access + BC. Tc\_transfer.

اگر miss رخ دهد، ابتدا یک دسترسی به حافظه‌ی نهان رخ داده، سپس یک دسترسی به حافظه‌ی اصلی، سپس از روی حافظه اصلی Bm بایت به نرخ انتقال Tm\_transfer انتقال می‌یابد، که می‌شود Tm\_transfer + Bm.Tm\_transfer + Tm\_access + Tc - Access.

۷ - گزینه ۲ درست است.

حافظه ایده‌آل حافظه‌ای است که گنجایش زیاد، قیمت ارزان و سرعت زیاد داشته باشد. از آن جا که این هر سه با هم در یک حافظه جمع نمی‌شود، از حافظه‌ی سلسله مراتبی استفاده می‌شود. مثلاً سرعت رجیسترهای CPU زیاد است. سرعت حافظه‌ای اصلی خیلی کمتر است. پس از حافظه‌ی نهان (cache) بین این دو استفاده می‌شود.

۸ - گزینه ۱ درست است.

$$\text{زمان متوسط دسترسی} = L1 \text{ hit} + L1 \text{ Miss}(L2 \text{ hit} + L2 \text{ Miss} \times \text{penalty})$$

$$= 1 + \frac{1}{100} \left( 4 + \frac{20}{100} \times 150 \right) = 1.34$$

یادداشت:

.....  
 .....  
 .....  
 .....