


**شکل ۷-۱۱** ارسال متوالی ناهمگام (غیرهمزمان)

کاراکتر انتقال یافته طبق قانون انتقال مشخص می‌شود:

۱- وقتی کاراکتر در حال ارسال نیست خط در 1 منطقی نگهداشته می‌شود.

۲- آغاز انتقال کاراکتر از بیت شروع که همواره 0 است مشخص می‌گردد.

۳- بیت‌های کاراکتر همواره بدنبال بیت شروع ارسال می‌گردد.

۴- پس از ارسال آخرین بیت کاراکتر، بیت توقف، وقتی که خط حداقل به مدت زمان یک بیت در سطح 1 باقی بماند، مشخص می‌گردد.

با استفاده از این قوانین، گیرنده قادر به تشخیص بیت شروع پس از انتقال خط از 1 به 0 است. ساعتی در گیرنده، خط را در زمان‌های مناسب وارسی می‌کند. گیرنده سرعت انتقال بیت‌ها و تعداد بیت‌های کاراکترها مورد پذیرش را می‌داند. پس از انتقال بیت‌های کاراکتر، یک یا دو بیت توقف ارسال می‌گردد بیت‌های توقف همواره در وضعیت 1 منطقی هستند و انتهای کاراکتر را برای مشخص نمودن حالت انتظار یا بیکاری کادربندی می‌نمایند.

در انتهای کاراکتر خط بمدت زمان یک یا دو بیت در حالت 1 نگهداشته می‌شود بطوری که فرستنده یا گیرنده می‌توانند مجدداً با یکدیگر همزمان شوند. طول زمان بقاء سیستم در این حالت به مقدار زمانی که برای دستگاه‌ها لازم است بستگی دارد. برخی پایانه‌های الکترومکانیکی قدیمی دو بیت توقف بکار می‌برند ولی پایانه‌های جدیدتر از یک بیت توقف استفاده می‌کنند. خط، تا انتقال کاراکتر بعدی در حالت 1 باقی می‌ماند. زمان توقف تضمین می‌کند که کاراکتر جدید به مدت یک یا دو برابر مدت زمان یک بیت روی خط ظاهر نخواهد شد.

بعنوان مثال، انتقال سریال به یک پایانه کم سرعت را در نظر بگیرید که سرعت انتقال آن 10 کاراکتر در ثانیه است. هر کاراکتر انتقال یافته متشکل از یک بیت شروع، هشت بیت اطلاعات، و دو بیت توقف است که جمعاً 11 بیت می‌شود. 10 کاراکتر در ثانیه به معنی سرعت انتقال هر کاراکتر اطلاعات در 0.1 ثانیه است. چون 11 بیت برای انتقال وجود دارد، زمان انتقال 9.09 میلی ثانیه خواهد بود. سرعت ارسال[1] (باود) بعنوان سرعت انتقال اطلاعات تعریف شده و معادل با انتقال داده بر حسب تعداد بیت‌ها

---

1- Baud Rate

در ثانیه است. 10 کاراکتر در ثانیه با قالب 11 بیتی دارای سرعت ارسال 110 با ود است.

پایانه دارای صفحه کلید و چاپگر است. هر بار کلیدی فشار داده شود، پایانه 11 بیت را بطور سری درطول مسیر ارسال می‌دارد. برای هر چاپ یک کاراکتر در یک چاپگر، یک پیام 11 بیتی بر روی سیم دیگری باید دریافت شود. واسطه پایانه از یک فرستنده و یک گیرنده تشکیل شده است. فرستنده یک کاراکتر 8 بیتی را از کامپیوتر دریافت کرده و سپس یک پیام 11 بیتی سری را به خط چاپگر می‌فرستد. گیرنده یک پیام 11 بیتی را از خط صفحه کلید دریافت و کد یک کاراکتر 8 بیتی را به کامپیوتر ارسال می‌دارد. برای ایجاد واسطه بین کامپیوتر و پایانه‌های محاوره‌ای مشابه، مدارات مجتمع خاصی طراحی شده‌اند. چنین مدارهایی مدار واسطه ارتباطی غیرهمزمان[1] یا گیرنده فرستنده عام غیرهمزمان (UART)[2] خوانده می‌شود.

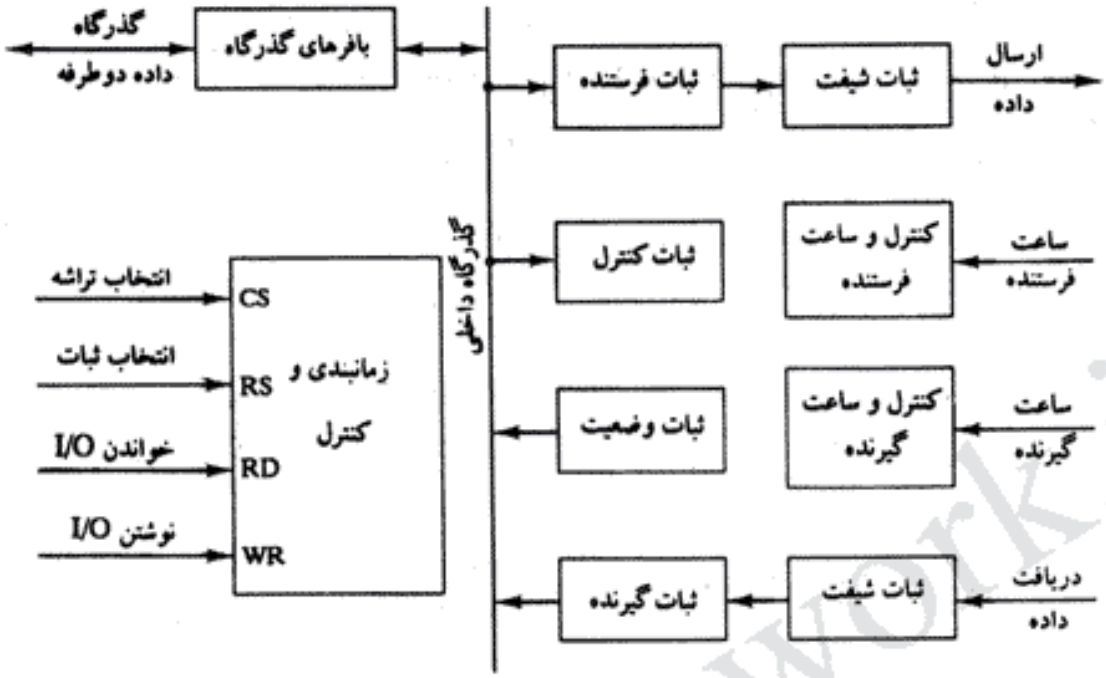
## واسطه ارتباطی غیرهمزمان[3]

بلاک دیاگرام مدار واسطه ارتباطی غیرهمزمان در شکل ۸-۱۱ نشان داده شده است. این واسطه هم بصورت فرستنده و هم گیرنده کار می‌کند. واسطه با استفاده از یک بایت کنترل که در ثبات کنترل بار می‌شود برای نوع خاصی از عمل انتقال راه‌اندازی اولیه می‌شود. ثبات فرستنده یک بایت داده را از طریق گذرگاه داده از CPU می‌پذیرد. این بایت به یک ثبات شیفت جهت ارسال سری منتقل می‌شود. بخش گیرنده، اطلاعات سری را با یک ثبات شیفت دیگر دریافت می‌کند و هنگامی که یک بایت کامل داده در این ثبات جای شد، آن را به ثبات گیرنده منتقل می‌کند. CPU می‌تواند ثبات گیرنده را برای خواندن بایت از طریق گذرگاه داده انتخاب کند. بیت‌های ثبات وضعیت برای پرچم‌های ورودی و خروجی و ضبط خطاهای معینی که ممکن است در حین ارسال رخ دهد بکار رود. CPU قادر است با خواندن ثبات وضعیت بیت‌های پرچم را وارسی کرده و معلوم نماید که خطایی رخ داده است یا خیر. ورودی انتخاب تراشه[4] (CS) و کنترل‌های خواندن و نوشتن با CPU در ارتباطند. ورودی انتخاب تراشه برای انتخاب مدار واسطه از طریق گذرگاه آدرس به کار می‌رود. ورودی انتخاب ثبات (RS) به کنترل‌های خواندن (RD) و نوشتن (WR) مربوط است. دو ثبات از نوع فقط نوشتن و دو ثبات دیگر از نوع فقط خواندن هستند. همانطور که در جدول همراه دیاگرام مشخص شده است، ثبات انتخابی تابعی از مقدار RS و وضعیت RD و WR است.

عملکرد واسطه ارتباطی غیرهمزمان با ارسال یک بایت به ثبات کنترل بوسیله CPU آغاز می‌شود. روند مقداردهی اولیه، با تعریف پارامترهای معینی از قبیل سرعت ارسال[5] مورد استفاده، تعداد بیت‌های توقف هرکاراکتر، آیا بیت توازن باید ایجاد و چک شود یا خیر، و تعداد بیت‌های هرکاراکتر، واسطه را در وضعیت کاری خاص قرار می‌دهد. دو بیت در ثبات وضعیت به عنوان پرچم بکارمی‌روند. یکی از این بیت‌ها برای

---

1- Asynchronous Communication Interface
2- Unirersal Asynchronous Reciever - Transmitter
3- Asynchronous
4- Chip Select
5- Baud rate




| CS | RS | عمل | ثباتی که انتخاب می‌شود |
|---|---|---|---|
| 0 | × | × | هیچ‌ا گذرگاه داده در امپدانس بالا |
| 1 | 0 | WR | ثبات فرستنده |
| 1 | 1 | WR | ثبات کنترل |
| 1 | 0 | RD | ثبات گیرنده |
| 1 | 1 | RD | ثبات وضعیت |

**شکل ۸-۱۱ بلاک دیاگرام یک نوع واسطه ارتباطات غیرهمزمان (غیرهمگام)**

نشان دادن خالی بودن ثبات فرستنده ثبات بودن و دیگری برای نشان دادن پربودن ثبات گیرنده بکارمی‌رود.

عملکرد بخش فرستنده واسطه بشرح زیر است. CPU ثبات وضعیت را می‌خواند و پرچم‌ها را برای خالی بودن ثبات فرستنده چک می‌کند. اگر این ثبات خالی باشد، CPU یک کاراکتر را به ثبات فرستنده ارسال می‌نماید و مدار واسطه پرچم را صفر می‌کند تا نشان دهد که ثبات پر است. اولین بیت ثبات فرستنده 0 می‌شود تا شروع را ایجاد کند. کاراکتر بطور موازی از ثبات فرستنده به ثبات شیفت منتقل می‌شود و تعداد مناسبی بیت‌های توقف که از پیش تعیین شده است به ثبات شیفت ملحق می‌گردد. سپس ثبات فرستنده بعنوان یک ثبات خالی تلقی می‌شود. حال کاراکتر می‌تواند بصورت هر بار یک بیت با جابجایی داده‌های موجود درثبات شیفت با سرعت انتقال مشخص شده ارسال شود. CPU می‌تواند پس از وارسی پرچم در ثبات وضعیت، کاراکتر دیگری را به ثبات فرستنده انتقال دهد. در این

حالت گوئیم مدار واسطه بصورت دوبله بافرشده[1] است. زیرا به محض اینکه کاراکتر قبلی شروع به ارسال شود کاراکتر جدید می‌تواند در ثبات فرستنده بار شود.

عملکرد بخش گیرنده مدار واسطه مشابه فوق است. وقتی که خط بیکار است ورودی دریافت داده در وضعیت 1 منطقی قرار می‌گیرد. واحد کنترل گیرنده بر خط دریافت داده نظارت می‌کند تا با مشاهده سیگنال 0 وقوع بیت شروع را شناسایی کند. پس از مشاهده بیت شروع، بیت‌های کاراکتر که متعاقباً فرا می‌رسند با سرعت ارسال از پیش تعیین شده وارد ثبات شیفت گیرنده می‌شوند. پس از دریافت بیت‌های داده، مدار واسطه، توازن و بیت‌های توقف را چک می‌کند. سپس بیت‌های کاراکتر بدون بیت‌های شروع و توقف بطور موازی از ثبات شیفت به ثبات گیرنده انتقال می‌یابند. پرچم مربوطه در ثبات وضعیت 1 می‌شود تا نشان دهنده پربودن ثبات گیرنده باشد. CPU ثبات وضعیت را خوانده و پرچم را وارسی می‌کند و در صورت 1 بودن آن، داده را از ثبات گیرنده می‌خواند.

در حین انتقال، مدار واسطه هر خطای ممکنی را چک نموده و بیت مربوطه را در ثبات وضعیت 1 می‌کند. در هر زمان که خطا رخ می‌دهد CPU می‌تواند ثبات وضعیت را وارسی کند. سه خطای ممکنی که مدار واسطه در طول ارسال وارسی می‌کند عبارتند از خطای توازن[2]، خطای کادربندی[3] و خطای تزاحم. خطای توازن هنگامی رخ می‌دهد که تعداد 1 در داده‌های رسیده دارای توازن صحیحی نیست. خطای کادربندی هنگامی بوقوع می‌پیوندد که تعداد بیت‌های توقف در انتهای کاراکتر رسیده صحیح نباشد. خطای تزاحم در صورتی رخ می‌دهد که قبل از خواندن کاراکتر از ثبات گیرنده بوسیله CPU، کاراکتر بعدی در ثبات شیفت قرار گیرد. این خطا موجب از دست رفتن کاراکتر ازرشته داده‌های دریافتی می‌شود.

## بافر اولین ورودی - اولین خروجی

بافر اولین ورودی - اولین خروجی (FIFO) واحد حافظه‌ای است که اطلاعات را به گونه‌ای ذخیره می‌کند که اولین داده وارد شده اولین داده‌ای خواهد بود که خارج شود. بافر FIFO پایانه‌های ورودی -خروجی جداگانه‌ای دارد. ویژگی مهم این نوع بافر اینست که می‌تواند ورود و خروج داده‌ها را با دو سرعت متفاوت انجام دهد و داده‌های خروجی همیشه بهمان ترتیبی که وارد می‌شوند خواهند بود. هر گاه FIFO بین دو واحد قرار گیرد، قادر خواهد بود داده‌ها را با سرعتی از واحد مبدأ دریافت و با سرعتی دیگر به واحد مقصد تحویل دهد. اگر واحد منبع کندتر از واحد مقصد باشد، می‌توان بافر را با سرعت کم در بافر پر کرد و بعداً با سرعت بیشتری آن را تخلیه نمود. اگر منبع سریعتر از مقصد باشد، FIFO برای مواقعی مفید خواهد بود که داده‌های ورودی بیکباره فرا برسند و میانگیر را پر کنند ولی بین هر دو بار دریافت داده زمان به حد کافی برای واحد مقصد وجود دارد تا همه یا قسمتی از اطلاعات را از بافر تخلیه کند. بنابراین بافر FIFO می‌تواند در برخی از کاربردها مفید باشد و آن هنگامی است که داده‌ها بطور غیرهمزمان انتقال یابند. این بافر داده‌ها را که به تدریج فرا می‌رسند انباشته کرده و آنها را هنگام لزوم با

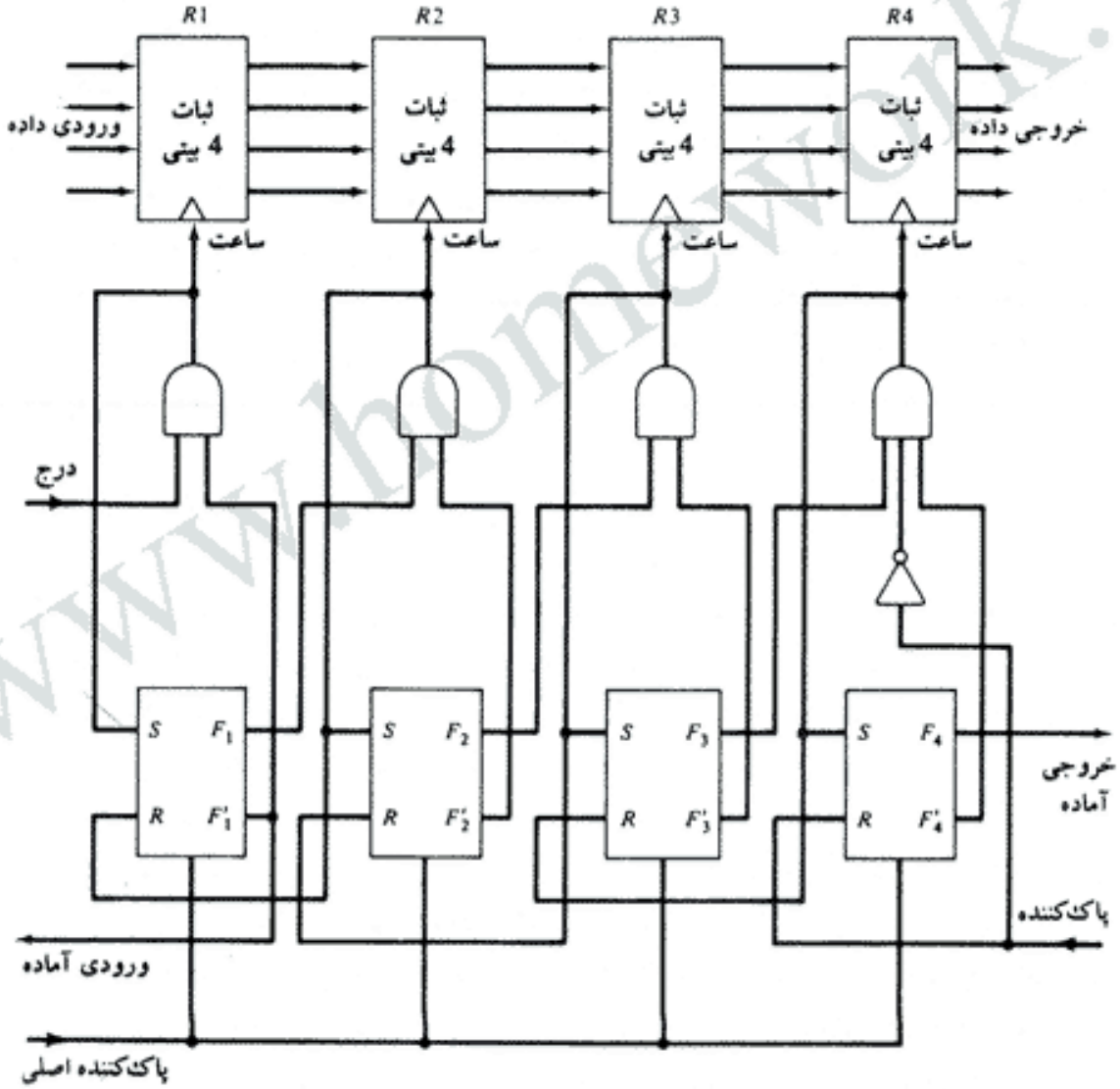
---

1- Double Buffered 2- Parity 3- Framing error

همان ترتیبی که رسیده‌اند تحویل می‌دهد.

دیاگرام منطقی یک بافر FIFO نمونه با ظرفیت 4×4 در شکل ۹-۱۱ نشان داده شده است. این بافر متشکل از چهار ثبات چهار بیت RI با I=1,2,3,4 و یک ثبات کنترل با فلیپ فلاپ‌های $F_i$، i=1,2,3,4 یعنی هر فلیپ فلاپ برای یک ثبات، می‌باشد. FIFO می‌تواند چهار کلمه چهار بیتی را ذخیره کند. تعداد بیت‌ها در هر کلمه می‌تواند با افزایش تعداد بیت‌ها در هر ثبات افزایش یابد و تعداد کلمات نیز با افزایش تعداد ثبات‌ها اضافه می‌گردد.

فلیپ‌فلاپ $F_i$ در ثبات کنترل که با 1 بار شده است نشان می‌دهد که یک کلمه چهار بیتی در ثبات




شکل ۹-۱۱ دیاگرام مدار بافر FIFO با اندازه 4 × 4

مربوطه‌اش RI ذخیره شده است. وجود 0 در $F_i$ بیانگر این حقیقت است که داده معتبری در ثبات مربوطه وجود ندارد. ثبات کنترل حرکت داده‌ها را در طول ثبات‌ها هدایت می‌نماید. هروقت بیت $F_i$ در ثبات کنترل 1 باشد ($F_i=1$) و بیت $F_{i+1}$ پاک شده باشد ($F'_{i+1}=1$) یک پالس ساعت تولید شده و موجب می‌شود که ثبات $R(I+1)$ داده را از ثبات RI بپذیرد. همان گذرپالس ساعت[1] $F_{i+1}$ را 1 و $F_i$ را 0 می‌کند. این وضعیت موجب می‌شود که پرچم کنترل یک مکان به راست همراه با داده حرکت کند. داده‌های موجود در ثبات‌ها تا جایی که فضای خالی در جلو آنها وجود داشته باشد به پائین FIFO و به سمت خروجی می‌روند. این حرکت موجی هنگامی که داده به ثباتی مانند RI برسد و فلیپ فلاپ بعدی آن یعنی $F_{i+1}$ مقدار 1 را داشته باشد، یا به آخرین ثبات $R_4$ برسد، متوقف می‌گردد. یک پاک‌کننده اصلی برای پاک کردن کلیه فلیپ فلاپ‌های ثبات کنترل بکار می‌رود.

داده‌ها بشرطی به داخل بافر وارد می‌شوند که سیگنال "ورودی آماده" فعال شده باشد. این هنگامی رخ می‌دهد که اولین فلیپ فلاپ کنترل، $F_1$، پاک شده باشد که بیانگر خالی بودن $R_1$ است . داده‌ها از طریق خطوط ورودی با فعال کردن ساعت در پایه درج[2] بار می‌شوند. همان پالس ساعت $F_1$ را 1 می‌نماید که این خود کنترل ورودی آماده[3] را غیرفعال نموده و این بنوبه خود بیانگر آنست که FIFO در حال حاضر مشغول و آماده پذیرش داده بیشتر نیست. فرآیند حرکت موجی بشرطی آغاز می‌شود که $R_2$ خالی باشد. داده از $R_1$ به $R_2$ منتقل و $F_1$ پاک می‌شود. این امر باعث فعال شدن ورودی آماده گشته و بیانگر آنست که ورودی‌ها برای کلمه داده دیگر آماده‌اند. اگر FIFO پرباشد، $F_1$ در حالت 1 باقی مانده ولی خط ورودی آماده در حالت 0 باقی می‌ماند. توجه کنید که دو خط کنترل ورودی آماده و درج یک جفت خط دست‌دهی را برای حالتی که مقصد آغازگر است تشکیل می‌دهد.

داده‌هایی که بداخل ثبات‌ها می‌افتند در سمت خروجی جمع می‌شوند. خط کنترل "خروجی آماده"[4] هنگامی که آخرین فلیپ فلاپ $F_4$ با 1 بار شود فعال می‌گردد که این مشخص کننده وجود داده معتبر در ثبات خروجی $R_4$ است. داده خروجی از $R_4$ بوسیله واحد مقصد پذیرفته شده که این سیگنال کنترل پاک کننده[5] را فعال می‌سازد. این بنوبه خود $F_4$ را پاک کرده و سبب غیرفعال شدن "خروجی آماده" می‌گردد. اگر FIFO خالی باشد، داده‌ای از $R_3$ وجود نداشته و $F_4$ در وضعیت 0 باقی می‌ماند. توجه کنید که دو خط کنترل "خروجی آماده" و "پاک کننده" تشکیل خطوط دست دهی برای منبع آغازگر را می‌دهند.

## ۴-۱۱ شیوه‌های انتقال

اطلاعات دودویی دریافتی از وسایل خارجی معمولاً برای پردازش بعدی در حافظه قرار می‌گیرند و اطلاعات منتقله از کامپیوتر مرکزی به وسیله خارجی در واقع از حافظه سرچشمه می‌گیرند. CPU صرفاً

---

1- Clock Transition
2- Insert
3- Input ready
4- Output ready
5- Delete

دستورالعمل‌های I/O را اجرا نموده و ممکن است داده را بطور موقت بپذیرد، ولی مبدا یا مقصد نهایی حافظه است. انتقال داده بین کامپیوتر مرکزی و وسایل I/O می‌تواند با شیوه‌های مختلفی انجام شود. برخی روشها از CPU بعنوان مسیر میانی استفاده می‌کند، بعضی مستقیماً داده را به و یا از حافظه منتقل می‌سازند. تبادل داده با وسایل جانبی به سه طریق زیر امکان پذیر است.

۱- I/O برنامه نویسی شده

۲- I/O بطریقه وقفه

۳- دستیابی مستقیم به حافظه (DMA)

عملیات I/O برنامه‌نویسی شده نتیجه دستورالعمل‌های I/O نوشته شده در برنامه کامپیوتر است. انتقال هر قلم داده از سوی دستورالعملی در برنامه آغاز می‌شود. معمولاً تبادل داده بین یک ثبات CPU و وسیله جانبی است. دستورات دیگری هم برای تبادل داده بین CPU و حافظه مورد نیازند. انتقال داده تحت کنترل برنامه مستلزم نظارت مداوم CPU بر دستگاه جانبی است. به محض آغاز انتقال، CPU باید بر مدار واسطه نظارت کند تا ببیند چه وقت می‌تواند دوباره انتقال را انجام دهد. در واقع این دیگر بعهده دستورات اجرا شده برنامه در CPU است تا دقیقاً بر هر اتفاقی که در واحد واسطه و وسیله I/O رخ می‌دهد نظارت کند.

در روش I/O برنامه نویسی شده، CPU در یک حلقه از برنامه باقی می‌ماند تا واحد I/O مشخص نماید که برای انتقال داده آماده است. این فرآیند موجب اتلاف قابل توجه وقت می‌شود زیرا پردازنده را بی‌آنکه لازم باشد مشغول نگه می‌دارد. برای احتراز از اتلاف وقت می‌توان از امکانات وقفه[1] و فرمانهای خاصی برای اطلاع به مدار واسطه استفاده کرده و از آن خواست تا بهنگام وجود داده از طرف وسیله جانبی یک سیگنال تقاضای وقفه[2] را صادر کند. در خلال این مدت مدار واسطه وسیله جانبی را زیرنظر دارد. وقتی مدار واسطه مشخص کند که وسیله جانبی آماده انتقال داده است، یک سیگنال تقاضای وقفه را به کامپیوتر صادر می‌کند. به محض تشخیص سیگنال وقفه خارجی، CPU موقتاً کار خود را متوقف کرده، به برنامه سرویس دهی برای پردازش انتقال I/O انشعاب نموده، و سپس به کاری که در اصل در حال انجام بود باز می‌گردد.

انتقال داده بروش I/O برنامه نویسی شده بین CPU و دستگاه جانبی انجام می‌شود. در دستیابی مستقیم به حافظه[3] (DMA)، مدار واسطه داده را از طریق گذرگاه حافظه تبادل می‌کند. CPU انتقال را با تهیه آدرس شروع برای واسطه و تعداد کلمات لازم برای انتقال آغاز نموده وسپس برای انجام سایر کارها پیشروی می‌نماید. وقتی که انتقال پایان یافت، DMA سیکل حافظه را از طریق گذرگاه حافظه تقاضا کند. وقتی که تقاضا بوسیله کنترل کننده حافظه پذیرفته شد، DMA داده‌ها را مستقیماً به حافظه می‌فرستد. CPU فقط عمل مراجعه به حافظه خود را به تأخیر می‌اندازد تا انتقال مستقیم I/O به حافظه

---

1- Interrupt 2- Interrupt request signal 3- Direct Memory Access

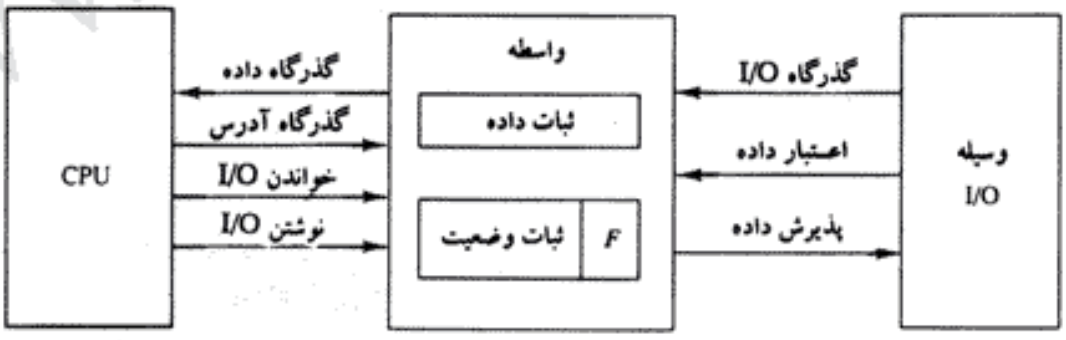
اجرا شود. چون معمولاً سرعت وسیله جانبی کم‌تر از سرعت پردازنده است، انتقال I/O - حافظه درمقایسه با مراجعات پردازنده به حافظه بندرت رخ می‌دهد. انتقال DMA در بخش ۶-۱۱ مفصل‌تر بحث شده است.

در بسیاری از کامپیوترها، مدار واسطه لازم برای ارجاع مستقیم به حافظه را در یک واحد که آن را پردازنده I/O (یا IOP)[1] می‌نامند قرار می‌دهد. IOP می‌تواند بسیاری از وسایل جانبی را از طریق یک DMA و امکان وقفه کنترل نماید. دریک چنین سیستمی، کامپیوتر به سه ماژول جداگانه تقسیم می‌شود: واحد حافظه، CPU ، و IOP. پردازشگر I/O در بخش ۷-۱۱ ارائه شده است.

## مثال I/O برنامه نویسی شده

در روش I/O برنامه‌نویسی شده، وسیله I/O دسترسی مستقیمی به حافظه ندارد. انتقال از یک وسیله I/O به حافظه مستلزم اجرای چندین دستور بوسیله CPU، از جمله یک دستور ورودی برای انتقال داده از وسیله به CPU و یک دستور ذخیره‌سازی برای انتقال داده از CPU به حافظه است. دستورالعمل‌های دیگری نیز ممکن است برای تحقیق اینکه وسیله جانبی داده را در اختیار گذاشته یا نه و شمارش تعداد کلمات انتقال یافته لازم باشد.

مثالی از یک وسیله I/O به CPU از طریق واسطه در شکل ۱۰-۱۱ نشان داده شده است. وسیله بایت‌های داده را یک به یک همانطور که در اختیار قرار می‌گیرند انتقال می‌دهد. وقتی که یک بایت در اختیار قرار گرفت، وسیله آن را روی گذرگاه I/O قرار می‌دهد و خط داده معتبر[2] را فعال می‌سازد. مدار واسطه بایت را پذیرفته ودر ثبات داده خود قرار می‌دهد و خط پذیرش داده[3] را فعال می‌نماید. مدار واسطه بیتی را در ثبات وضعیت 1 می‌نماید که ما آن را بیت F یا پرچم[4] می‌خوانیم. در این لحظه وسیله جانبی می‌تواند خط داده معتبر را غیرفعال نماید ولی بایت دیگری را ، تا غیرفعال شدن خط پذیرش داده




F: بیت پرچم

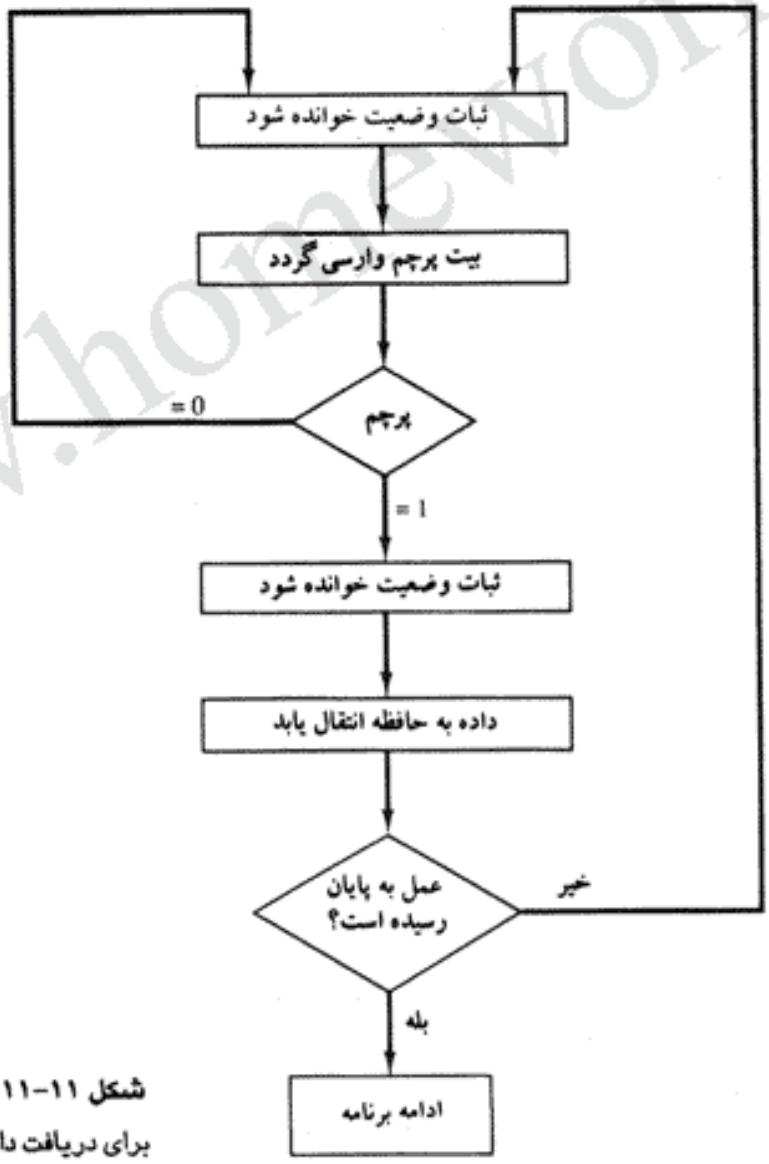
شکل ۱۰-۱۱ انتقال داده از وسیله I/O به CPU

1- I/O Processor 2- Data Valid 3- Data Accepted
4- Flag

بوسیله مدار واسطه، منتقل نخواهد کرد. این عملیات برطبق روند دست دهی مبتنی بر شکل ۵-۱۱ است.

برنامه‌ای برای کامپیوتر نوشته شده است تا پرچم را در ثبات وضعیت وارسی نموده و تعیین کند که آیا وسیله I/O بایتی در ثبات داده قرار داده است یا خیر. این عمل با خواندن ثبات وضعیت و قرار دادن آن در یک ثبات CPU و وارسی بیت پرچم صورت می‌گیرد. اگر پرچم برابر 1 باشد، CPU داده را از ثبات داده می‌خواند. سپس بیت پرچم با 0 شدن بوسیله CPU و یا مدار واسطه، بسته به اینکه مدار واسطه چگونه طراحی شده باشد، پاک می‌شود. به محض پاک شدن پرچم، مدارواسطه خط پذیرش داده را غیرفعال کرده و بنابراین وسیله جانبی می‌تواند بایت داده بعدی را منتقل سازد.

فلوچارت برنامه‌ای که باید برای CPU نوشته شود. در شکل ۱۱-۱۱ کشیده شده است. در این چارت فرض شده است که وسیله جانبی رشته‌ای ازبایت‌هایی که باید در حافظه ذخیره شوند را ارسال




**شکل ۱۱-۱۱ فلوچارت برنامه CPU برای دریافت داده‌های ورودی**

می‌نماید. انتقال هر بایت نیاز به سه دستورالعمل زیر دارد.

۱- خواندن ثبات وضعیت

۲- چک کردن بیت پرچم در ثبات وضعیت و انشعاب به مرحله 1 اگر این بیت 1 نباشد، و یا به مرحله 3 اگر بیت مذکور 1 باشد.

۳- خواندن ثبات داده.

هر بایت بداخل یک ثبات پردازنده خوانده شده و سپس با یک دستورالعمل ذخیره به حافظه منتقل می‌شود. یک کار متداول در برنامه‌نویسی I/O انتقال بلاکی از کلمات از وسیله I/O و ذخیره آنها در بافر حافظه است. برنامه‌ای که کاراکترهای ورودی را در یک بافر حافظه با استفاده از دستورات فصل ۶ تعریف می‌نماید در جدول ۲۱-۶ لیست شده است.

روش I/O برنامه نویسی شده خصوصاً در کامپیوترهای کوچک با سرعت کم یا در سیستم‌هایی مختص نظارت مداوم بریک وسیله مفید است. این نوع انتقال بدلیل تفاوت سرعت انتقال اطلاعات بین CPU و وسیله I/O روش کارآیی نیست. برای اثبات این عدم کارآیی، یک کامپیوتر نوعی را در نظر بگیرید که می‌تواند دو دستورالعملی که ثبات وضعیت را می‌خواند و پرچم را وارسی می‌کند در یک میکروثانیه اجرا نماید. فرض کنید که وسیله ورودی داده‌های خود را با سرعت 100 بایت در ثانیه منتقل نماید. این سرعت معادل یک بایت در هر 10 هزار میکروثانیه است. در نتیجه CPU بین هر دو بار انتقال 10 هزار بار پرچم را وارسی می‌کند. CPU هنگام وارسی پرچم بجای انجام یک کار پردازشی مفید دیگر، در واقع در حال اتلاف وقت است.

## I/O بکمک وقفه

راه دیگری علاوه بر نظارت مداوم CPU بر پرچم، ایجاد این امکان است که هر گاه واسطه آماده انتقال داده بود به کامپیوتر اطلاع دهد. این روش از انتقال از امکان وقفه استفاده می‌کند. CPU مادامی که مشغول اجرای یک برنامه است پرچم را وارسی نمی‌کند. با این وجود، هنگامی که پرچم 1 شده باشد، یک وقفه کامپیوتر را لحظه‌ای از اجرای برنامه جاری متوقف ساخته و 1 شدن پرچم را به آن اطلاع می‌دهد. CPU از آنچه در حال انجام آن است منصرف شده و به انتقال ورودی یا خروجی می‌پردازد. پس ازتکمیل انتقال، کامپیوتر به برنامه اصلی برمی‌گردد تا آنچه راکه قبل از وقفه انجام می‌داد ادامه دهد.

CPU در پاسخ به سیگنال وقفه، آدرس بازگشت را از شمارنده در پشته‌ای در حافظه ذخیره می‌کند و سپس کنترل به یک روال سرویس که انتقال I/O مطلوب را پردازش می‌کند، انشعاب می‌نماید.

طریقی که CPU برای انتخاب آدرس انشعاب روال سرویس برمی‌گزیند از یک نوع به نوعی دیگر فرق می‌کند. در اصل برای انجام این عمل دو راه وجود دارد. یک راه وقفه برداری[1] و دیگری وقفه

---

1- Vectored Interrupt

غیربرداری[1] نام دارد. در وقفه غیربرداری، آدرس انشعاب به مکان ثابتی از حافظه تخصیص می‌یابد. در وقفه بردرای منبعی که وقفه را ایجاد می‌کند اطلاعات انشعاب را نیز برای کامپیوتر فراهم می‌آورد. این اطلاعات بردار وقفه[2] نامیده می‌شود. در بعضی از کامپیوترها بردار وقفه آدرسی است که به مکانی از حافظه که روال سرویس I/O است اشاره می‌کند. سیستمی با وقفه برداری در بخش ۵-۱۱ آورده شده است.

### ملاحظات نرم افزاری

بحث قبلی به سخت افزار پایه مورد نیاز برای ارتباط وسایل I/O به کامپیوتر مربوط بود. کامپیوتر باید یک سری روال‌های نرم افزاری هم برای کنترل وسایل جانبی و انتقال داده‌ها بین پردازنده و وسایل جانبی داشته باشد. روال‌های I/O باید فرمان‌های کنترل را برای فعال کردن وسایل جانبی برای وارسی وضعیت و آگاهی از لحظه‌ای که آماده انتقال داده هستند صادر کنند. به محض آمادگی، اطلاعات قلم به قلم تا انتقال کامل آنها منتقل می‌شود. در بعضی موارد یک فرمان کنترلی صادر می‌شود تا عملی مثل توقف نوار یا چاپ کاراکترها انجام گیرد. غالباً همراه با انتقال، چک کردن خطاها و سایر مراحل مفید دیگر نیز انجام می‌شود. در انتقال‌های کنترل شده با وقفه، نرم افزار I/O باید فرمانهایی را به دستگاه جانبی ارسال کند تا هر وقت آماده بود وقفه را صادر نماید و هر وقت که وقفه صادر شده خود وقفه را سرویس دهد. در انتقال DMA، نرم افزار I/O باید کانال DMA را برای آغاز عمل آن راه اندازی کند.

کنترل نرم افزاری تجهیزات ورودی - خروجی کاری پیچیده است. باین دلیل روال‌های I/O برای دستگاه‌های جانبی استاندارد توسط سازنده بعنوان بخشی از سیستم کامپیوتری در اختیار گذاشته می‌شود. این روال ها معمولاً همراه با سیستم عرضه می‌شود. اکثر سیستم‌های عامل انواع متنوعی از برنامه‌های I/O برای پشتیبانی خط خاص دستگاه‌های جانبی برای کامپیوتر را به همراه دارند. روال‌های I/O معمولاً بصورت رویه‌های سیستم عامل در اختیار قرار دارند و کاربر بدون اینکه وارد برنامه‌های زبان ماشین شود با ارجاع به روال‌های مربوطه نوع انتقال مطلوب را مشخص می‌کند.

## ۵-۱۱ وقفه اولویت دار

انتقال داده بین CPU و وسیله I/O بوسیله CPU شروع می‌شود. با این وجود CPU نمی‌تواند تا آمادگی وسیله برای ارتباط، انتقال را شروع نماید. آمادگی وسیله جانبی توسط یک سیگنال وقفه معین می‌شود. CPU به درخواست وقفه با ذخیره کردن آدرس بازگشت از PC بداخل حافظه‌ای در پشته[3] پاسخ داده و سپس برنامه به روال سرویس که انتقال مورد نظر را انجام می‌دهد انشعاب می‌کند. همانطور که در بخش ۷-۸ بحث شد، بعضی از پردازنده‌ها PSW[4] جاری را به پشته منتقل می‌کنند و یک PSW جدیدی را نیز برای روال سرویس بار می‌نمایند. در اینجا ما از PSW چشم‌پوشی می‌کنیم تا بحث وقفه

---

1- Nonvectored Interrupt 2- Interrupt vector 3- Stack
4- Program Status Word (PSW)

I/O پیچیده نشود.

دریک کاربرد نوعی، تعدادی از وسایل I/O به کامپیوتر متصلند، و هر یک نیز قادر است یک تقاضای وقفه را ارسال دارد. اولین کار سیستم وقفه شناخت منبع وقفه است. همچنین امکان درخواست وقفه همزمان از طرف چندین وسیله نیز وجود دارد. دراین حالت سیستم باید تصمیم بگیرد که ابتدا به کدام وسیله سرویس دهد.

وقفه اولویت دار سیستمی است که اولویت را برای انواع منابع ایجاد می‌کند تا بدین ترتیب معین کند کدام یک از چند تقاضای همزمان ابتدا سرویس دهی شود. سیستم ممکن است معین کند که وقتی کامپیوتر مشغول سرویس دادن به وقفه است کدام وقفه دیگر می‌تواند کامپیوتر در حال اجرا را متوقف نماید. سطوح بالاتر وقفه معمولاً به تقاضاهایی اختصاص می‌یابد که اگر تأخیر یا وقفه‌ای در آنها ایجاد شود موجب عواقب وخیمی گردد. به وسایلی که دارای سرعت انتقال بالایی هستند، مانند دیسک‌های مغناطیسی، اولویت بالاتری داده می‌شود، و وسایل کند مانند صفحه کلید اولویت پائین‌تری را دارا می‌باشند. وقتی دو وسیله بطور همزمان کامپیوتر را متوقف کنند، کامپیوتر اول به اولویت بالاتر پاسخ می‌دهد.

ایجاد اولویت به وقفه‌های همزمان می‌تواند بوسیله نرم افزار یا سخت افزار انجام گیرد. برای شناخت منبعی با بالاترین اولویت بطریقه نرم افزاری، از رویه همه پرسی [1] استفاده می‌شود. در این روش، یک آدرس انشعاب مشترک برای تمام وقفه‌ها وجود دارد. برنامه‌ای که مراقبت از وقفه را بعهده دارد از این آدرس شروع شده و به همه منابع بطور متوالی مراجعه می‌کند. ترتیبی که این وقفه‌ها تست می‌شوند تعیین کننده اولویت هر وقفه است. منبعی که بالاترین اولویت را دارد اول تست می‌شود، و اگر سیگنال وقفه آن فعال باشد کنترل به روال سرویس این منبع انشعاب می‌کند. در غیراین صورت منبعی که اولویت پائینتر را دارد تست می‌شود و الی آخر. بنابراین روال سرویس دهی اولیه متشکل از برنامه‌ای است که منابع وقفه را بترتیب تست می‌کند و به یکی از چند روال سرویس موجود انشعاب می‌نماید. روال سرویس‌دهی که اجرا می‌شود متعلق به وسیله‌ای است که بالاترین اولویت را در بین همه وسیله‌هائی که به کامپیوتر وقفه ارسال داشته‌اند داراست. عیب روش نرم افزاری این است که اگر وقفه‌های بسیاری وجود داشته باشد، ممکن است زمان لازم برای مراجعه و همه پرسی از آنها از زمان سرویس‌دهی به وسیله I/O تجاوز کند. در این شرایط می‌توان از یک دستگاه وقفه اولویت دار سخت‌افزاری برای تسریع عمل استفاده کرد.

دستگاه وقفه اولویت‌دار سخت‌افزاری بصورت مدیر کل در سیستم وقفه عمل می‌کند. این دستگاه درخواست‌های وقفه را از چندین منبع دریافت و سپس تعیین می‌کند که کدام تقاضا دارای بالاترین اولویت است. آنگاه برمبنای آن یک درخواست وقفه را به کامپیوتر صادر می‌نماید. برای تسریع عمل، هر منبع وقفه بردار وقفه خاص خود را برای دستیابی مستقیم به روال سرویس مخصوص آن داراست. به

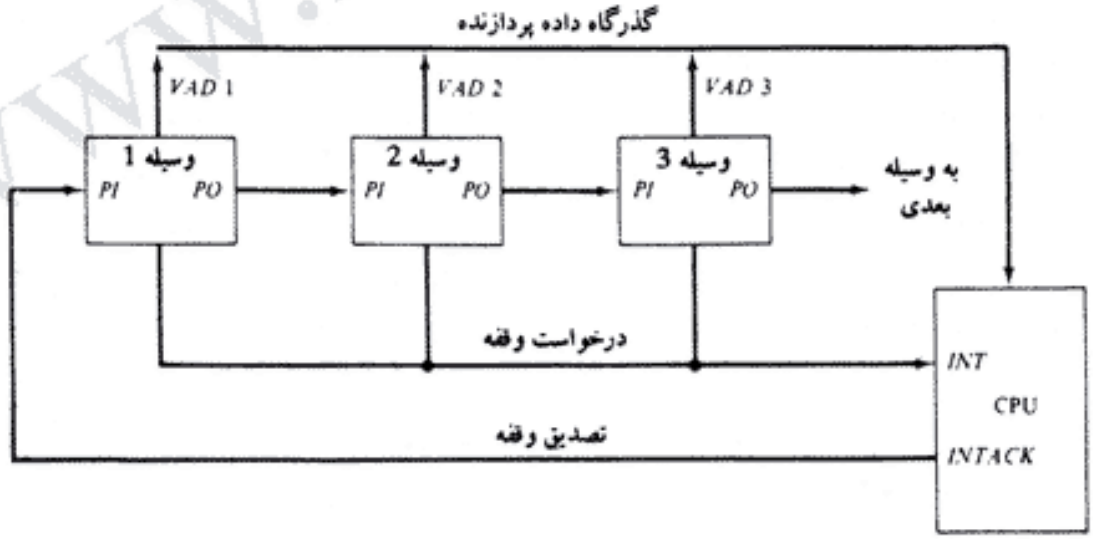
---

1- Polling Procedure

این ترتیب همه پرسی لازم نیست زیرا همه تصمیم‌ها توسط دستگاه وقفه اولویت دار سخت افزاری اتخاذ می‌گردد. عمل اولویت دهی سخت افزاری را می‌توان با اتصال سری یا موازی خطوط وقفه انجام داد. روش اتصال سری، اتصال زنجیره‌ای[1] نیز خوانده می‌شود.

## اولویت زنجیره‌ای

روش زنجیره‌ای برای اولویت دهی بر اتصال سری وسیله‌هایی که درخواست وقفه ارسال می‌دارند مبتنی است. وسیله‌ای که بالاترین اولویت را داراست در اولین مکان قرار داده می‌شود، و بدنبال آن وسیله‌های دارای اولویت پائین‌تر، تا وسیله‌ای که پائین‌ترین اولویت را دارد و در انتهای زنجیر قرار گرفته، آمده‌اند. این روش اتصال در شکل ۱۲-۱۱ برای سه وسیله و CPU نشان داده شده است. خط در خواست وقفه برای همه وسیله‌ها غیرمشترک بوده و یک اتصال منطقی سیمی[2] را تشکیل می‌دهد. اگر هر وسیله سیگنال وقفه خود را در سطح پائین ببرد، خط وقفه به سطح پائین رفته و ورودی وقفه را در CPU فعال می‌سازد. وقتی که هیچ وقفه‌ای رخ ندهد، خط وقفه در سطح بالا باقی مانده و هیچ وقفه‌ای بوسیله CPU تشخیص داده نمی‌شود. این وضعیت معادل OR با منطق منفی است. CPU با فعال کردن خط تصدیق[3] وقفه به تقاضای وقفه پاسخ می‌دهد. این سیگنال را وسیله شماره 1 در ورودی خود، PI، (ورودی اولویت) دریافت می‌کند. در صورتی که وسیله 1 درخواست وقفه نداده باشد سیگنال تصدیق از خروجی PO (خروجی اولویت) به وسیله بعدی منتقل می‌گردد. اگر وسیله 1 درخواست وقفه داده باشد، با قرار دادن مقدار 0 روی خروجی PO مانع از رسیدن سیگنال تصدیق به وسیله بعدی می‌شود.



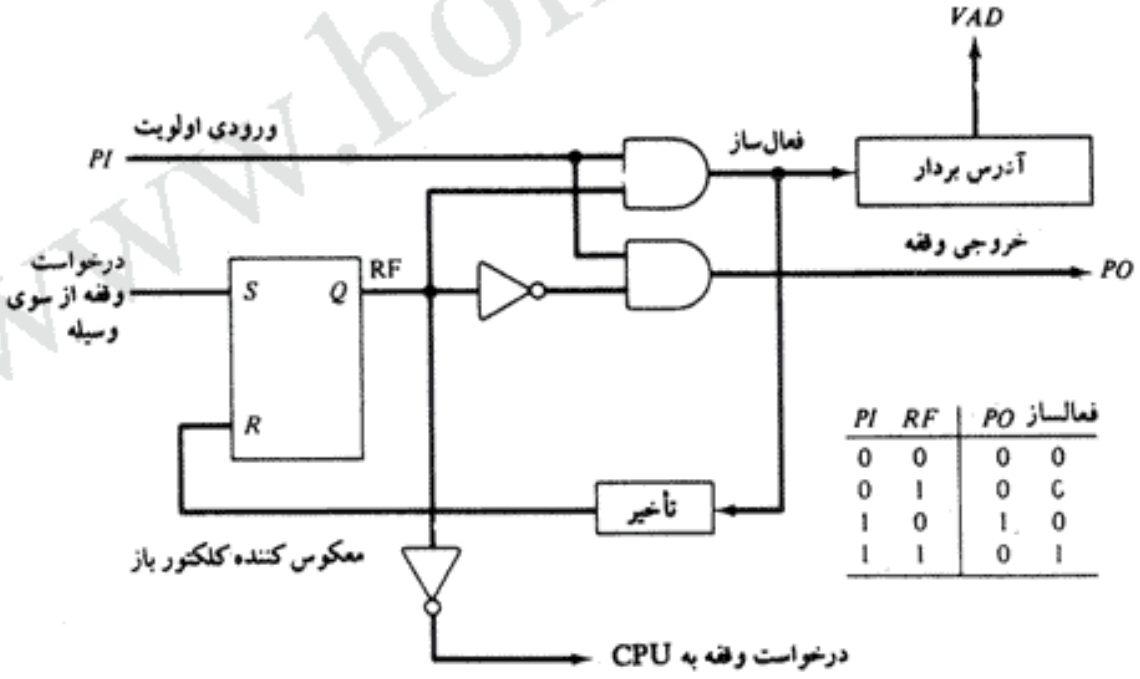

شکل ۱۲-۱۱ سیستم وقفه اولویت‌دار زنجیره‌ای

1- Daisy channing
2- Wired logic
3- Acknowledge

سپس این وسیله آدرس بردار وقفه[1] خود را (VAD) روی گذرگاه قرار می‌دهد تا CPU در سیکل وقفه از آن استفاده نماید.

وسیله‌ای که در ورودی PI دارای 0 باشد در خروجی PO هم 0 تولید می‌کند تا به وسیله‌ای که دارای اولویت وقفه پائین‌تر بعدی است اطلاع دهد که سیگنال تصدیق ممانعت شده است. وسیله‌ای که تقاضای وقفه نموده باشد و دارای 1 در ورودی PI باشد سیگنال تصدیق را با قرار دادن 0 در خروجی PO می‌بندد.اگر وسیله جانبی وقفه سرویس دهی شونده‌ای نداشته باشد، سیگنال تصدیق را با قرار دادن 1 در خروجی PO خود به وسیله بعدی می‌فرستد. بنابراین وسیله‌ای که دارای PI=1 و PO=0 باشد دارای بالاترین اولویت در تقاضای وقفه است، و این وسیله VAD خود را روی گذرگاه داده قرار می‌دهد. آرایش زنجیره‌ای بالاترین اولویت رابه وسیله‌ای که سیگنال تصدیق وقفه را از CPU دریافت کند تخصیص می‌دهد. هر چقدر وسیله از اولین مکان دورتر باشد دارای اولویت پائین‌تری است.

شکل ۱۳-۱۱ مدار منطقی درونی برای هر وسیله بهنگام اتصال به سیستم زنجیره‌ای را نشان می‌دهد. هنگامی که وسیله بخواهد CPU را متوقف کند فلیپ فلاپ RF خود را 1 می‌کند. خروجی فلیپ فلاپ RF از طریق یک معکوس کننده کلکتورباز[2]، یعنی مداری که منطق سیمی برای خط وقفه مشترک فراهم می‌سازد، می‌گذرد. اگر PI=0 باشد، هر دو خط PO و فعال ساز VAD، صرفنظر از مقدار RF ، برابر صفرند.اگر PI=1 باشد و RF=0 باشد، سپس RO=1 و بردار آدرس غیرفعال




شکل ۱۳-۱۱ یک طبقه از آرایش اولویت‌دهی زنجیره‌ای

1- vector
2- Open - collector

می‌گردد. این وضعیت سیگنال تصدیق را از طریق PO به وسیله بعدی منتقل می‌سازد. وسیله جانبی هنگامی فعال است که PI=1 و RF=1 باشد. این وضعیت مقدار 0 را در PO قرار داده و بردار آدرس رابرای گذرگاه داده فعال می‌سازد. فرض براین است که هر وسیله دارای بردار آدرس متمایز خاص خود است. مدتی پس از دریافت بردار آدرس توسط CPU، و اطمینان از انجام آن، فلیپ فلاپ RF، 0 می‌شود.

## وقفه اولویت دار موازی

روش وقفه اولویت دار موازی از ثباتی استفاده می‌کند که بیت‌های آن بطور جداگانه توسط سیگنال وقفه هر وسیله 1 می‌شود. اولویت براساس محل بیت‌ها در ثبات تعیین می‌گردد. علاوه بر ثبات وقفه، مدار ممکن است دارای یک ثبات پوشش[1] باشد که هدف از آن کنترل وضعیت هر تقاضای وقفه است. ثبات پوشش (ماسک) را می‌توان برای غیرفعال شدن وقفه‌های پائین رتبه‌تر، در حین سرویس دهی به وقفه‌هایی با اولویت بالاتر، برنامه‌دهی کرد. این ثبات همچنین می‌تواند این امکان را فراهم سازد که در حین سرویس دهی به وسیله‌ای با اولویت پائینتر ، وسیله‌ای با اولویت بالاتر بتواند به CPU وقفه دهد.

مدار منطقی اولویت برای یک سیستم با چهار منبع وقفه در شکل ۱۴-۱۱ نشان داده شده است. این مدار شامل یک ثبات وقفه است که تک تک بیت‌های آن تحت شرایط خارجی 1 و توسط دستورات برنامه 0 می‌شود. به یک دیسک مغناطیسی، چون یک وسیله سریع است، بالاترین اولویت داده می‌شود. چاپگر اولویت بعدی را داراست و بدنبال آن دستگاه کاراکتر خوان و صفحه کلید می‌آیند. ثبات پوشش دارای تعداد بیت‌های مساوی با ثبات وقفه است. توسط دستورات برنامه، می‌توان هر بیتی را در ثبات پوشش 0 یا 1 کرد. هر بیت وقفه و بیت پوشش متناظر آن به یک گیت AND برای تولید چهار ورودی انکدر اولویت، اعمال می‌شوند.بدین‌ترتیب، وقفه تنها در صورتی قابل تشخیص است که بیت پوشش متناظر آن بوسیله برنامه 1 شده باشد. انکدر اولویت دو بیت آدرس بردار را ایجاد می‌کند که به CPU منتقل می‌گردد.

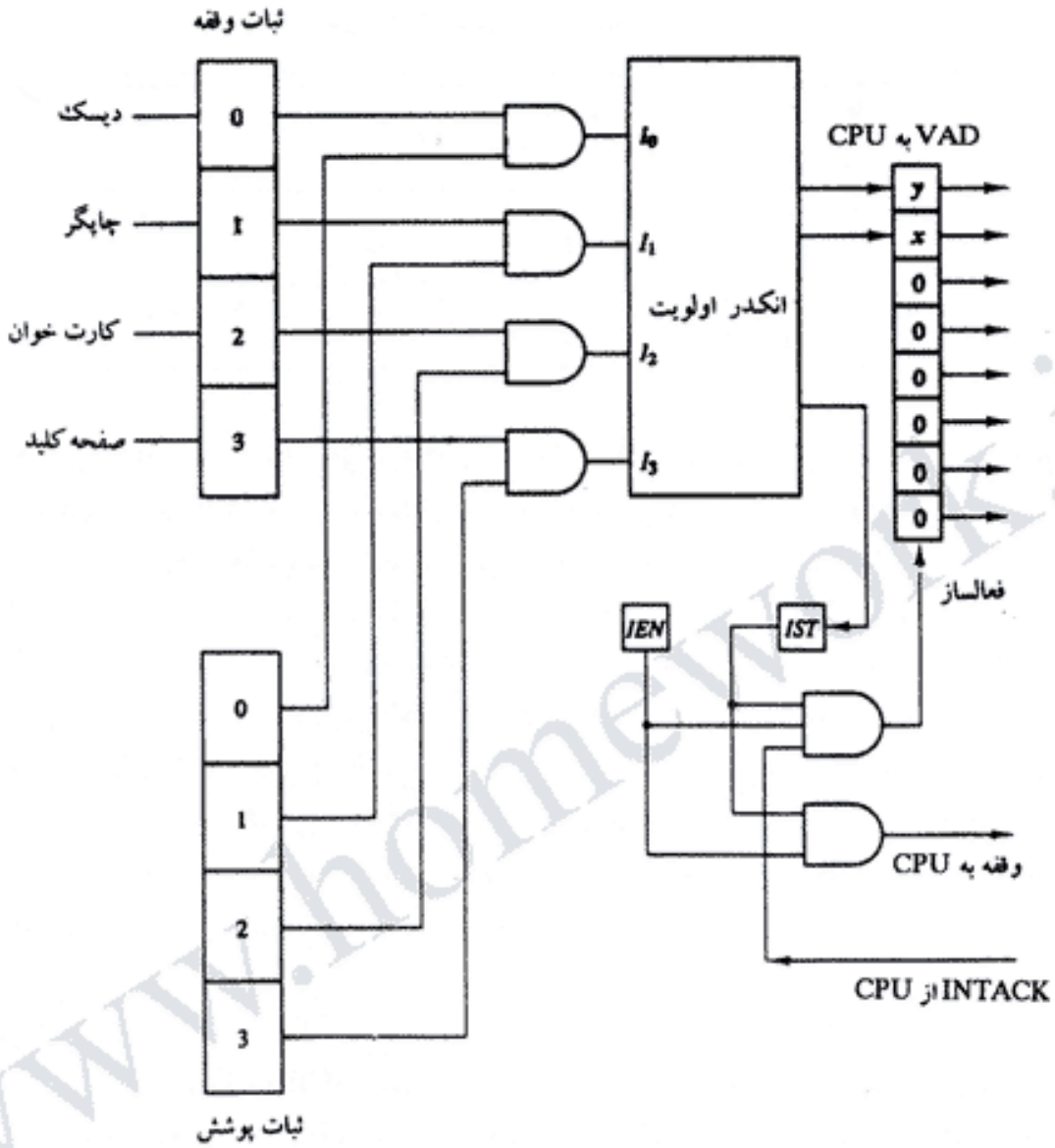
خروجی دیگر انکدر، فلیپ فلاپ وضعیت وقفه IST را، وقتی یک وقفه پوشش نیافته رخ دهد، 1 می‌کند. فلیپ فلاپ توانا ساز وقفه IEN می‌تواند بوسیله برنامه 1 یا 0 شود تا کنترل کلی را روی سیستم وقفه فراهم آورد. خروجی IST با AND شدن با IEN، سیگنال وقفه مشترکی را برای CPU فراهم می‌کند. سیگنال تصدیق وقفه INTACK از CPU، بافر گذرگاه در ثبات خروجی را فعال کرده و لذا یک آدرس بردار VAD روی گذرگاه داده قرار می‌گیرد. حال ما مدار انکدر اولویت را توضیح داده و سپس عکس‌العمل متقابل بین کنترل کننده وقفه و CPU را بحث خواهیم کرد.

## انکدر اولویت

انکدر اولویت مداری است که عمل اولویت دهی را تحقق می‌بخشد. منطق انکدر اولویت طوری

---

1- Mask Register




شکل ۱۴-۱۱ سخت‌افزار وقفه اولویت‌دار

است که اگر دو یا چند ورودی در یک لحظه وارد آن شوند، ورودی که بالاترین اولویت را دارا باشد تقدم خواهد داشت. جدول درستی انکدر اولویت چهار ورودی در جدول ۲-۱۱ ارائه شده است. x ها در جدول حالت بی‌اهمیت را تداعی می‌کنند. ورودی $I_0$ بالاترین اولویت را دارد، لذا صرف‌نظر از مقادیر دیگر ورودی‌ها، وقتی که این ورودی 1 باشد، خروجی تولید xy =00 می‌نماید. $I_1$ اولویت بعدی را دارد. اگر $I_1=1$ باشد، مشروط بر اینکه $I_0=0$ باشد، صرفنظر از مقادیر دو ورودی دیگر که اولویت پائینتر را دارند خروجی 01 خواهد بود.

**جدول ۲-۱۱** جدول ارزش کدگذار اولویت

| ورودی‌ها | | | | خروجیها | | | |
|---|---|---|---|---|---|---|---|
| $I_0$ | $I_1$ | $I_2$ | $I_3$ | $x$ | $y$ | $IST$ | توابع بولی |
| 1 | × | × | × | 0 | 0 | 1 | |
| 0 | 1 | × | × | 0 | 1 | 1 | $x = I'_0 I'_1$ |
| 0 | 0 | 1 | × | 1 | 0 | 1 | $y = I'_0 I'_1 + I'_0 I'_2$ |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | $(IST) = I_0 + I_1 + I_2 + I_3$ |
| 0 | 0 | 0 | 0 | × | × | 0 | |

خروجی برای $I_2$، هنگامی تولید می‌شود که ورودی‌های با اولویت بالاتر 0 باشند و به همین منوال اولویت‌های پائین‌تر. فلیپ فلاپ وضعیت وقفه IST فقط وقتی که یک یا چند ورودی برابر 1 باشند، 1 می‌شود. اگر تمام ورودی‌ها 0 باشند، IST پاک می‌شود و سایر خروجی‌های انکدر بکار نمی‌روند، لذا با x علامت زده شده‌اند. این بدان علت است که آدرس بردار وقتی که IST = 0 است به CPU منتقل نمی‌شود. توابع بول لیست شده در جدول منطق درونی انکدر را مشخص می‌کند. معمولاً یک کامپیوتر بیش از چهار منبع وقفه دارد. مثلاً یک انکدر با هشت ورودی، یک خروجی سه بیتی تولید می‌نماید.

خروجی انکدر اولویت، برای تشکیل بخشی از آدرس بردار هر منبع وقفه بکار می‌رود. به سایر بیت‌های بردار آدرس هر مقداری را می‌توان اختصاص داد. مثلاً آدرس بردار را می‌توان با الحاق شش صفر به خروجی‌های x و y انکدر تشکیل داد. با این انتخاب، شماره‌های دودویی 0، 1، 2، 3 به بردارهای وقفه مربوط به چهار وسیله I/O اختصاص می‌یابد.

## سیکل وقفه

فلیپ فلاپ تواناساز وقفه IEN که در شکل ۱۴-۱۱ نشان داده شده است می‌تواند بوسیله دستورالعمل‌های برنامه 1 یا 0 شود. وقتی که IEN پاک شود، تقاضای وقفه‌ای که از IST می‌رسد بوسیله CPU نادیده گرفته می‌شود. بیت IEN کنترل شده بوسیله برنامه به برنامه نویس اجازه می‌دهد تا درمورد استفاده یا عدم استفاده از امکانات وقفه تصمیم بگیرد. اگردستورالعملی برای پاک کردن IEN در برنامه گنجانیده شود، بدان معنی است که برنامه نویس مایل نیست برنامه‌اش متوقف شود. دستورالعملی برای 1 کردن IEN نشانه این است که در حین اجرای برنامه جاری از قابلیت وقفه استفاده خواهد شد. اکثر کامپیوترها دارای سخت افزار برای پاک کردن IEN، در مواقعی که وقفه بوسیله پردازنده تصدیق شوند، هستند.

در پایان هر سیکل دستورالعمل ، CPU بیت IEN وسیگنال وقفه را از IST وارسی می‌کند. اگر هر یک از آنها 0 باشد، کنترل با دستور بعدی به کار ادامه می‌دهد. اگر هر دو بیت IEN و IST برابر 1 باشند، CPU وارد سیکل وقفه می‌شود. در حین سیکل وقفه، CPU رشته ریزاعمال زیر را انجام می‌دهد.

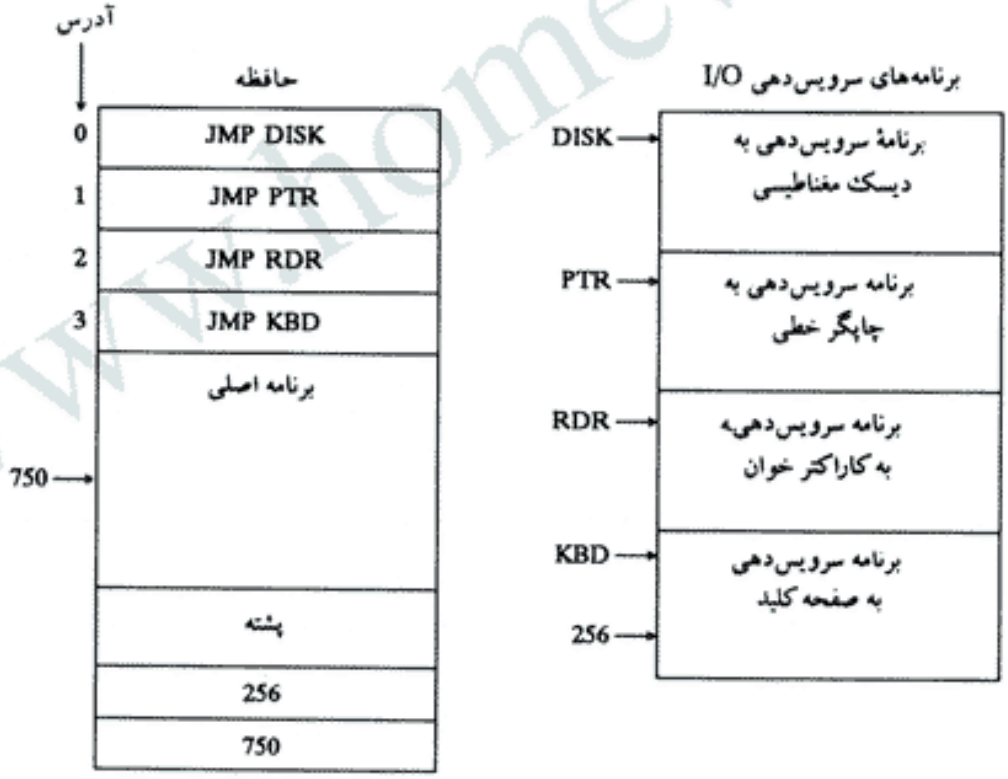
اشاره‌گر پشته را کاهش بده $SP \leftarrow SP-1$

| | |
|---|---|
| M [SP] ← PC | پوش کردن PC به داخل پشته |
| INTACK ← 1 | تصدیق وقفه را فعال کن |
| PC ← VAD | بردار آدرس را به PC منتقل کن |
| IEN ← 0 | وقفه بعدی را غیرفعال کن |
| | برو برای برداشت دستورالعمل بعدی |

CPU آدرس برگشت را از PC به پشته پوش می‌کند سپس با فعال کردن خط INTACK وقفه را فعال می‌کند. واحد اولویت وقفه با قرار دادن یک بردار وقفه منحصر بفرد در گذرگاه داده به وقفه پاسخ می‌دهد. CPU سپس با انتقال بردار آدرس به PC قبل از رفتن به فاز برداشت[1] بعدی IEN را پاک می‌کند. دستورالعملی که در حین فاز برداشت بعدی از حافظه خوانده می‌شود در بردار آدرس قرار خواهد داشت.

## روال‌های نرم افزاری

یک سیستم اولویت وقفه ترکیبی از تکنیک‌های سخت افزاری و نرم افزاری است. ماتاکنون جنبه‌های سخت افزاری اولویت وقفه را بحث کردیم. کامپیوتر باید برای سرویس دهی به تقاضاهای




**شکل ۱۵-۱۱** برنامه‌های ذخیره شده در حافظه برای سرویس‌دهی به وقفه‌ها

1- Fetch

وقفه و کنترل ثبات‌های سخت افزار وقفه روال‌های نرم افزاری نیز داشته باشد. شکل ۱۵-۱۱ برنامه‌هایی راکه باید برای اداره سیستم وقفه در حافظه قرار داشته باشد نشان می‌دهد. هر وسیله برنامه سرویس‌دهی خاص خود را دارد و از طریق یک دستورالعمل پرش (JMP)، که در آدرس بردار قرار دارد قابل دسترسی است. نام سمبلیک هر روال نشان دهنده آدرس شروع برنامه سرویس است. پشته نشان داده شده برای ذخیره کردن آدرس بازگشت پس از هر وقفه است.

برای روشن کردن مطلب با یک مثال فرض کنید که صفحه کلید بیت وقفه را مادامی که CPU در حال اجرای دستورالعملی در مکان 749 برنامه اصلی است، 1 کند. در پایان سیکل دستورالعمل، کامپیوتر به یک سیکل وقفه می‌رود. ابتدا آدرس بازگشت 750 را در پشته ذخیره کرده و سپس آدرس بردار 00000011 را ازگذرگاه پذیرفته و آنرا به PC انتقال می‌دهد. آنگاه دستورالعمل واقع درمکان 3 اجرا شده، و در نتیجه کنترل به روال KBD منتقل می‌شود. حال فرض کنید که دیسک بیت وقفه‌اش را، هنگامی که CPU در حال اجرای دستورالعمل واقع در مکان 255 در برنامه KBD است، 1 کند. آدرس 256 بداخل پشته پوش داده شده و کنترل به برنامه سرویس DISK منتقل می‌شود. آخرین دستور در هر برنامه، دستورالعمل بازگشت از وقفه است. وقتی که برنامه سرویس دیسک کامل شود، دستور بازگشت پشته را پاپ کرده و عدد 256 را در PC می‌گذارد. این عمل کنترل را به روال KBD برای ادامه روال سرویس صفحه کلید منتقل می‌سازد. در پایان برنامه KBD، آخرین دستورالعمل پشته را پاپ کرده و کنترل برنامه را به برنامه اصلی برمی‌گرداند. بنابراین، وسیله‌ای که دارای اولویت بالاتر باشد می‌تواند بوسیله‌ای که اولویت پائین‌تر را دارد وقفه دهد. فرض براین است که زمان صرف شده در سرویس به وسیله‌ای که وقفه اولویت بالا دارد نسبت به وسیله‌ای که اولویت پائین را داراست کوتاه‌تر باشد و بنابراین مشکل از دست دادن اطلاعات رخ نمی‌دهد.

## عملیات آغازین و پایانی

هر روال سرویس وقفه باید مجموعه‌ای از عملیات آغازین و پایانی را برای کنترل ثبات‌ها سیستم وقفه سخت افزاری داشته باشد. بخاطر بیاورید که فعال ساز وقفه IEN در پایان سیکل وقفه پاک می‌شود. این فلیپ فلاپ مجدداً باید برای فعال کردن تقاضاهای وقفه با اولویت بالاتر، 1 شود، ولی نه قبل از غیرفعال شدن وقفه‌های با اولویت پائین‌تر. رشته اعمال آغازین هر روال سرویس وقفه باید دارای دستورالعمل‌هایی برای کنترل سخت افزار وقفه به شکل زیر باشد.

۱- صفر کردن بیت‌های سطح پائینتر ثبات پوشش
۲- پاک کردن بیت وضعیت وقفه IST
۳- ذخیره کردن محتوای ثبات‌های پردازنده
۴- 1 کردن بیت فعال ساز وقفه IEN
۵- اجرای روال سرویس

بیت‌های سطح پائین‌تر ثبات پوشش (از جمله بیت منبعی که وقفه داده است) پاک شده تا مانع فعال شدن وقفه تحت این شرایط گردد. اگرچه منابع وقفه‌ای که اولویت پائینتر دارند به بیت‌های بالا رتبه ثبات پوشش تخصیص یافته‌اند، می‌توان اولویت را در صورت تمایل عوض کرد زیرا برنامه نویس می‌تواند هر آرایشی را برای ثبات پوشش اختیار کند. بیت وضعیت وقفه باید پاک شود تا بتواند هنگام رخداد وقفه با اولویت بالاتر، برابر با 1 شود. محتوای ثبات پردازنده ذخیره می‌شوند زیرا ممکن است بوسیله برنامه‌ای که به آن وقفه داده شده است پس از بازگشت کنترل به آن موردنیاز واقع شود. سپس فعال ساز وقفه IEN برابر با 1 شده و اجازه می‌دهد سایر وقفه‌ها (با اولویت بالاتر) رخ دهند و کامپیوتر تقاضاهای وقفه را سرویس می‌دهد.

درپایان هر روال سرویس دهی باید دستورالعمل‌هایی برای کنترل سخت افزار وقفه بشکل زیر موجود باشد.

۱- پاک کردن بیت فعال ساز وقفه IEN
۲- بازگرداندن محتوای ثبات‌های پردازنده
۳- پاک کردن بیت متعلق به منبعی که سرویس داده شده، در ثبات وقفه
۴- 1کردن بیت های با اولویت پائینتر در ثبات پوشش
۵- بازگرداندن آدرس بازگشت در PC، و 1کردن IEN

بیتی در ثبات وقفه که به منبع وقفه متعلق است باید 0 شود تا این منبع بتواند مجدداً وقفه را تقاضا کند. بیت‌هایی دارای اولویت پائین تر در ثبات پوشش (از جمله بیت مربوط به منبعی که به آن سرویس داده شد) 1 می‌شوند که بتوانند وقفه بدهند. بازگشت به برنامه متوقف شده با بازگرداندن آدرس برگشت به PC صورت می‌گیرد. توجه کنید که سخت افزار باید طوری طراحی شود که هیچ وقفه‌ای ضمن اجرای مراحل ۲ تا ۵ رخ ندهد؛ در غیراینصورت آدرس بازگشت ممکن است مفقود شده و اطلاعات در ثبات‌های پوشش و پردازنده ممکن است از دست برود. با این دلیل IEN ابتداً پاک شده و پس از انتقال آدرس بازگشت در ثبات PC دوباره 1 می‌شود.

عملیات آغازین و پایانی لیست شده فوق عملیات سربار[1] نامیده می‌شوند. این عملیات بخشی از برنامه اصلی سرویس نیستند ولی برای پردازش وقفه ضرورت دارند. همه علمیات سربار را می‌توان با نرم افزار پیاده‌سازی کرد. این عمل با گنجاندن دستورالعمل‌های مناسب در ابتدا و انتهای هر زیرروال انجام می‌شود. برخی از اعمال سربار بوسیله سخت افزار بطور اتوماتیک انجام می‌گردد. محتوای ثبات‌های پردازنده می‌تواند قبل از انشعاب به روال سرویس بداخل پشته پوش داده شود. سایر عملیات آغازین و پایانی را نیز می‌توان به عهده سخت افزار گذاشت. به این ترتیب می‌توان زمان بین دریافت یک وقفه و اجرای دستورالعمل‌های لازم برای سرویس دادن به منبع وقفه را کاهش داد.

---

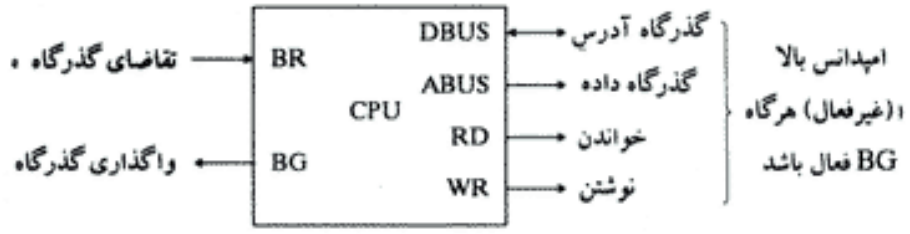
1- Overhead or Housekeeping Chores

## ۶-۱۱ دستیابی مستقیم به حافظه (DMA)[1]

انتقال داده بین یک وسیله ذخیره سازی سریع مانند دیسک مغناطیسی و حافظه اغلب بوسیله سرعت CPU محدود می‌شود. حذف CPU از مسیر و ایجاد امکان کنترل مستقیم گذرگاه حافظه توسط وسیله جانبی سرعت انتقال را بهبود می‌بخشد. این تکنیک انتقال دستیابی مستقیم به حافظه (DMA) نام دارد. در حین انتقال بروش DMA، CPU بیکار است و کنترلی بر گذرگاه‌های حافظه ندارد. یک کنترل کننده DMA به منظور اداره مستقیم امر انتقال و بین وسیله I/O و حافظه، کنترل گذرگاه‌ها را بدست می‌گیرد.

CPU را می‌توان بطرق مختلف در حالت بیکاری قرارداد. یک روش معمول که بطور گسترده‌ای در ریزپردازنده‌ها از آن استفاده می‌شود غیرفعال کردن گذرگاه‌ها بوسیله سیگنال‌های کنترل خاصی است. در شکل ۱۶-۱۱ دوسیگنال کنترل در CPU که امکان انتقال DMA را فراهم می‌کنند نشان داده شده است. ورودی تقاضای گذرگاه[2] (BR) بوسیله کنترل کننده DMA برای تقاضای وقفه از CPU بکار می‌رود تا کنترل گذرگاه‌ها را رها نماید. وقتی این ورودی فعال است، CPU اجرای دستورالعمل جاری را پایان داده و گذرگاه آدرس، گذرگاه داده، و خطوط خواندن و نوشتن را در حالت امپدانس بالا قرار می‌دهد. حالت امپدانس بالا مانند مدار باز عمل می‌کند، یعنی خروج قطع شده و معنی منطقی نخواهد داشت (۴-۳ را ببینید). CPU خروجی واگذاری گذرگاه[3] (BG) را فعال می‌کند تا به DMA اطلاع دهد که گذرگاه‌ها در حالت امپدانس - بالا هستند. اکنون DMAای که تقاضای گذرگاه را داده است می‌تواند کنترل گذرگاه‌ها را برای هدایت انتقال‌های حافظه بدون دخالت پردازنده به دست بگیرد. وقتی که DMA انتقال را به انجام رساند، خط تقاضای گذرگاه را غیرفعال می‌کند. CPU خط واگذاری گذرگاه را غیرفعال می‌کند، کنترل گذرگاه را بعهده می‌گیرد، و آنرا به عملکرد عادی خود باز می‌گرداند.

وقتی که DMA کنترل سیستم گذرگاه رابعهده می‌گیرد، مستقیماً با حافظه ارتباط برقرار می‌کند. انتقال می‌تواند به چندین طریق انجام شود. در انتقال یکباره[4] DMA، یک رشته بلاک متشکل از یک تعداد کلمات حافظه در حالی که DMA بر گذرگاه حافظه حاکم است، بطور پیوسته بیکباره منتقل می‌گردد. این شیوه انتقال برای وسایل سریع مانند دیسک‌های مغناطیسی ضروری است. در این سیستم‌ها




شکل ۱۶-۱۱ سیگنال‌های گذرگاه CPU برای انتقال DMA

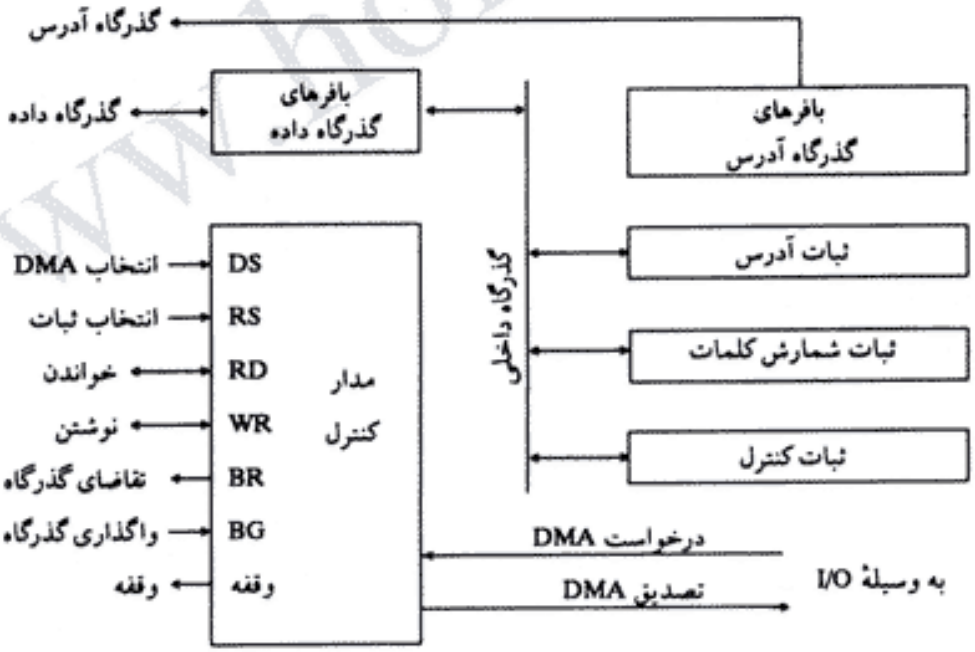
1- Direct Memory Access 2- Bus Request 3- Bus Grant
4- Burst Transfer

نمی‌توان تاانتقال کامل بلاک داده‌ها، انتقال را متوقف یا کند کرد. تکنیک دیگری که سرقت سیکل[1] نامیده می‌شود به کنترل کننده DMA اجازه می‌دهد که هر بار یک کلمه داده را انتقال دهد و پس از آن کنترل گذرگاه‌ها را به CPU بازگرداند. CPU صرفاً عملکرد خود را به اندازه یک سیکل حافظه به تأخیر می‌اندازد تا عمل انتقال حافظه I/O بطور مستقیم صورت گرفته و یک سیکل حافظه را برباید.

## 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 شمارش کلمه[2] و یک مجموعه خطوط آدرس هم مورد نیاز است. ثبات آدرس و خطوط آدرس برای ارتباط مستقیم با حافظه بکار می‌روند. ثبات شمارش کلمه تعداد کلماتی را که باید منتقل شود مشخص می‌کند. انتقال داده‌ها می‌تواند مستقیماً تحت کنترل DMA بین وسیله جانبی و حافظه صورت گیرد.

شکل ۱۷-۱۱ بلاک دیاگرام یک نمونه کنترل کننده DMA را نشان می‌دهد. کنترل کننده با CPU از طریق گذرگاه داده و خطوط کنترلی ارتباط برقرار می‌کند. ثبات‌های DMA بوسیله CPU از مسیر گذرگاه آدرس وبا فعال کردن ورودی‌های DS (انتخاب DMA)[3] و RS (انتخاب ثبات)[4] انتخاب می‌شوند. ورودی‌های RD (خواندن) و WR (نوشتن) دوجهته هستند. وقتی که BG (واگذاری گذرگاه) صفرباشد،




**شکل ۱۷-۱۱** بلاک دیاگرام کنترل کننده DMA

1- Cycle stealing
2- Word Count Register
3- DMA Select
4- Register select

CPU می‌تواند از طریق گذرگاه داده برای خواندن یا نوشتن در ثبات‌های DMA، با مشخص کردن آدرس و فعال کردن کنترل‌های RD و WR، مستقیماً با آنها ارتباط برقرار کند. DMA با وسایل خارجی از طریق خطوط تقاضا و تصدیق و استفاده از رویه از قبل تعین شده دست دهی ارتباط برقرار می‌کند.

کنترل‌کننده DMA سه ثبات دارد: ثبات آدرس، ثبات شمارش کلمه و ثبات کنترل. ثبات آدرس حاوی یک آدرس برای مشخص کردن مکان موردنظر در حافظه است. بیت‌های آدرس از طریق بافرها به گذرگاه آدرس وارد می‌شوند. ثبات آدرس پس از انتقال هر کلمه به حافظه یک واحد اضافه می‌شود. ثبات شمارش کلمه تعداد کلماتی که باید منتقل شوند را نگه می‌دارد. این ثبات پس از هر بار انتقال یک واحد کسر می‌شود. ثبات شمارش کلمه تعداد کلمات انتقالی را نگه می‌دارد. ثبات پس از هر انتقال یک واحد کم شده و در آغاز هم برای صفر بودن وارسی می‌شود. ثبات کنترل شیوه انتقال را معین می‌کند. تمام ثبات‌ها در DMA از دید CPU مانند ثبات‌های I/O هستند. بنابراین CPU از ثبات DMA تحت کنترل برنامه و از طریق گذرگاه، داده را خوانده و یا در آن می‌نویسد.

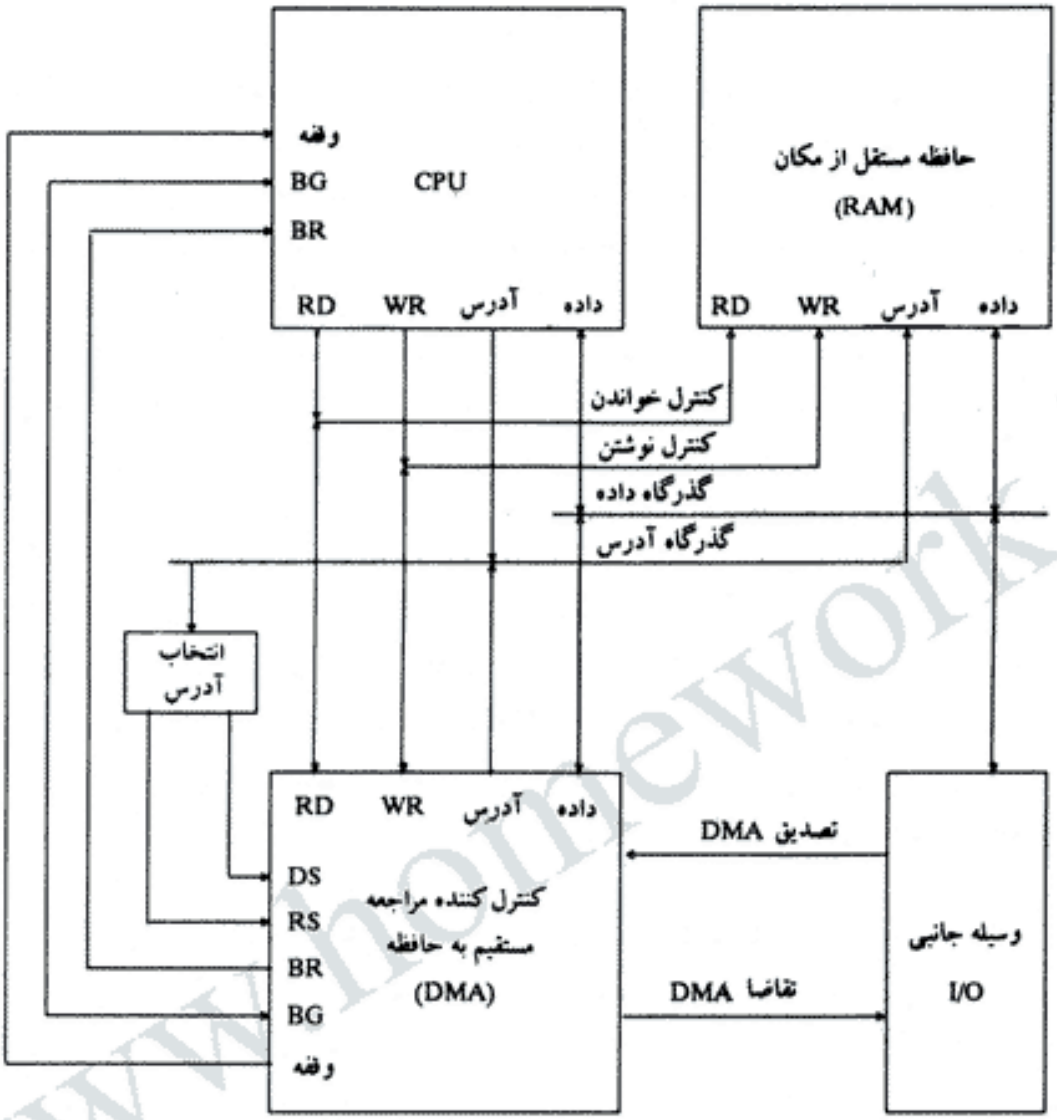
ابتدا DMA بوسیله CPU مقداردهی اولیه می‌شود. پس از آن، DMA شروع به کار کرده و انتقال داده بین حافظه و واحد جانبی را تا انتقال کامل بلاک انجام می‌دهد. روند مقداردهی اولیه اساساً برنامه‌ای است متشکل از دستورالعمل‌های I/O که شامل آدرس انتخاب ثبات‌های خاص DMA است. CPU با ارسال اطلاعات زیر از طریق گذرگاه داده DMA را مقداردهی اولیه می‌کند.

۱- آدرس شروع بلاکی از حافظه که داده‌ها در آن قرار دارد (برای خواندن) و یا داده‌ها باید در آن ذخیره شود (برای نوشتن)
۲- شمارش کلمات، یعنی تعداد کلمات بلاک حافظه
۳- کلمه کنترل برای مشخص کردن اطلاعات مربوط به شیوه انتقال مانند خواندن یا نوشتن
۴- کلمه کنترلی برای شروع انتقال DMA

آدرس شروع در ثبات آدرس ذخیره می‌شود. تعداد کلمه در ثبات شمارش و کلمه اطلاعات کنترلی در ثبات کنترل ذخیره می‌گردد. وقتی که DMA مقدار دهی اولیه شد، CPU از ارتباط با DMA دست می‌کشد مگر اینکه سیگنال وقفه‌ای دریافت کند و یا بخواهد تعداد کلمات انتقال یافته را وارسی نماید.

### انتقال DMA

موقعیت کنترل کننده DMA در میان سایر قسمت‌های کامپیوتر در شکل ۱۸-۱۱ نشان داده شده است. CPU با DMA از طریق گذرگاه‌های آدرس و داده، مانند سایر واسطه‌ها، ارتباط برقرار می‌کند. DMA آدرس خاص خود را دارد، که خطوط DS و RS را فعال می‌کند. CPU مقداردهی اولیه به DMA را از طریق گذرگاه داده انجام می‌دهد. وقتی که DMA فرمان شروع کنترل را دریافت می‌کند، انتقال وسیله جانبی و حافظه آغاز می‌شود.




**شکل ۱۸-۱۱** انتقال DMA در یک سیستم کامپیوتری

وقتی که وسیله جانبی درخواست DMA را ارسال می‌کند، کنترل کننده DMA خط BR را فعال می‌نماید، و به این طریق به CPU اطلاع می‌دهد که گذرگاه‌ها را رها کند. CPU با فعال کردن خط BG به DMA اطلاع می‌دهد که گذرگاه‌های آن غیر فعالند. سپس DMA مقدار فعلی ثبات آدرس خود را روی گذرگاه آدرس قرار داده، سیگنال RD یا WR را فعال می‌کند، و سیگنال تصدیق DMA را به وسیله جانبی می‌فرستد. توجه کنید که خطوط RD و WR در کنترل کننده DMA دوجهته هستند. جهت انتقال بستگی به وضعیت خط BG دارد. وقتی که BG=0 باشد، خطوط RD و WR برای DMA ورودی بوده و این امکان را برای CPU فراهم می‌کنند که با ثبات‌های DMA ارتباط برقرار کند. وقتی که BG=1 است، RD و WR خطوط خروجی از کنترل کننده DMA به RAM بوده و عمل خواندن یا نوشتن را مشخص می‌کنند.

وقتی که وسیله جانبی یک سیگنال تصدیق را از DMA دریافت کند، کلمه‌ای رادر گذرگاه داده قرار می‌دهد (برای نوشتن) یا کلمه را از گذرگاه داده دریافت می‌کند (برای خواندن). بنابراین DMA عمل خواندن یا نوشتن راکنترل می‌کند و آدرس حافظه را فراهم می‌کند. واحد جانبی سپس با حافظه از طریق گذرگاه داده برای انتقال مستقیم بین دو واحد ارتباط برقرار می‌نماید و این مدت CPU موقتاً غیرفعال است.

برای هر کلمه‌ای که انتقال یابد، DMA ثبات آدرس را افزایش داده و ثبات شمارش کلمه را کاهش می‌دهد. اگر شمارش کلمه به صفر نرسد، DMA خط تقاضای[1] واصله از وسیله جانبی را چک می‌کند. برای یک وسیله سریع، خط به محض خاتمه انتقال قبلی فعال خواهد شد. سپس دومین انتقال آغاز می‌شود، و روند تا انتقال کامل بلاک ادامه می‌یابد. اگر سرعت وسیله جانبی کندتر باشد، خط تقاضای DMA ممکن است قدری دیرتر برسد. در این صورت DMA خط تقاضای گذرگاه را غیرفعال می‌کند تا CPU بتواند به اجرای برنامه خود ادامه دهد. وقتی که دستگاه جانبی تقاضای انتقال کرد، DMA مجدداً گذرگاه‌ها را تقاضا می‌نماید.

اگر ثبات شمارش به صفر برسد، DMA از ارسال بیشتر انتقال خودداری می‌نماید و تقاضای گذرگاه خود را حذف می‌کند. همچنین CPU را از اتمام کار بوسیله وقفه مطلع می‌سازد. وقتی که CPU به وقفه پاسخ دهد، محتوای ثبات شمارش کلمه را می‌خواند. مقدار صفر در این ثبات بمعنی انتقال موفقیت آمیز کلیه کلمات است. CPU در هر لحظه قادر به خواندن این ثبات می‌باشد و بدینوسیله کلماتی راکه قبلاً منتقل شده‌اند وارسی می‌کند.

یک کنترل کننده DMA ممکن است بیش از یک کانال داشته باشد. در این حال، هر کانال دارای جفت سیگنال‌های کنترل تقاضا و تصدیق می‌باشد که بطور جداگانه به وسایل جانبی متصل است. هر کانال، همچنین دارای ثبات آدرس و ثبات شمارش کلمه خود در کنترل کننده DMA است. میان کانال‌ها می‌توان اولویتی هم برپاکرد بطوری که کانال با اولویت بالاتر قبل از کانال‌های با اولویت پائین‌تر سرویس داده شود.

انتقال DMA در بسیاری از کاربردها بسیار مفید است. این روش در انتقال سریع اطلاعات بین دیسک‌های مغناطیسی و حافظه بکار می‌رود. همچنین برای بهنگام کردن صفحه نمایش در پایانه محاوره‌ای نیز مفید است. معمولاً، تصویری از نمایش صفحه پایانه[2] در حافظه نگهداری می‌شود که قابل بهنگام شدن تحت کنترل برنامه است. محتوای حافظه می‌تواند بطور دوره‌ای با استفاده از انتقال DMA به صفحه ارسال شود.

## ۷-۱۱ پردازنده ورودی و خروجی IOP[3]

بعوض ارتباط هر مدار واسطه با CPU، یک کامپیوتر می‌تواند یک یا چند پردازنده خارجی داشته باشد که به هر یک از آنها کاری که مستقیماً ارتباط با وسایل I/O را فراهم می‌سازد تخصیص داد. یک

---

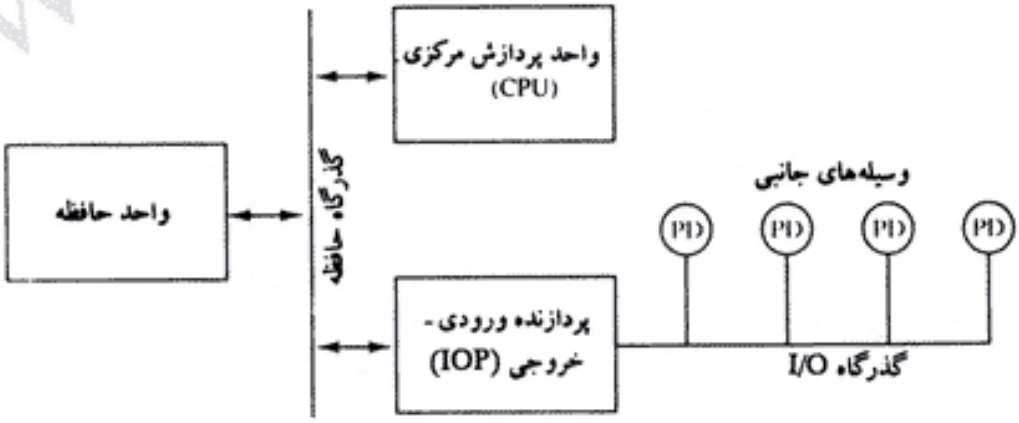
1- Request Line 2- Terminal 3- Input - Output Processor

پردازنده ورودی - خروجی (IOP) ممکن است بعنوان پردازنده‌ای با مراجعه مستقیم به حافظه طبقه‌بندی شود که با وسیله‌های I/O تبادل اطلاعات می‌کند. دراین پیکربندی، سیستم کامپیوتر می‌تواند به یک واحد حافظه و تعدادی پردازنده از CPU و یک یا چند IOP تقسیم شود. هر IOP از یک کار ورودی یا خروجی مراقبت می‌کند، و CPU را از علمیات سربار موجود در انتقال آزاد می‌سازد. یک پردازنده که با پایانه‌های دوردست از طریق تلفن و سایر وسایل مخابراتی بصورت سری تبادل اطلاعات می‌کند، پردازنده تبادل داده[1] (DCP) نامیده می‌شود.

IOP مشابه CPU است جز اینکه برای انجام جزئیات پردازش I/O طراحی شده است. برخلاف کنترل‌کننده DMA که باید کلاً بوسیله CPU تنظیم شود، IOP می‌تواند دستورالعمل‌های خود را برداشت و اجرا کند. دستورالعمل‌های IOP بخصوص برای انتقال IOP طراحی شده‌اند. بعلاوه IOP می‌تواند پردازش دیگری مانند محاسبات، منطق، انشعاب و ترجمه کد را نیز انجام دهد.

بلاک دیاگرام یک کامپیوتر با دو پردازنده در شکل ۱۹-۱۱ دیده می‌شود. واحد حافظه یک موقعیت مرکزی را اشغال کرده است و بصورت دستیابی مستقیم به حافظه می‌تواند با هر پردازنده ارتباط برقرار کند. CPU مسئول پردازش داده لازم در حل کارهای محاسباتی است. پردازنده IOP مسیری برای انتقال داده بین وسایل مختلف جانبی و واحد حافظه است. CPU معمولاً کار مقداردهی اولیه به برنامه I/O را برعهده دارد. از این پس IOP بطور مستقل از CPU کار کرده و انتقال داده را از وسیله خارجی به حافظه ادامه می‌دهد.

قالب‌های داده وسایل جانبی با قالب‌های داده حافظه و قالب داده CPU تفاوت دارد. IOP باید به کلمات داده دریافتی از منابع متعدد و مختلف ساختار بدهد. مثلاً، ممکن است لازم باشد تا چهار بایت از یک وسیله ورودی دریافت شده و قبل از انتقال به حافظه بصورت کلمه 32 بیتی درآید. در حالی که




**شکل ۱۹-۱۱** بلاک دیاگرام کامپیوتر با پردازنده I/O

1- Data Communication Prosessor

CPU برنامه خود را اجرا می‌کند داده‌ها در IOP با سرعت و ظرفیت بیت وسیله ورودی جمع می‌شوند. پس از اینکه داده‌ها در کلمه حافظه کنار هم قرار گرفتند[1]، مستقیماً از IOP با سرقت یک سیکل حافظه از CPU به حافظه منتقل می‌شوند. بطور مشابه یک کلمه حافظه انتقال یافته از حافظه به IOP از IOP به وسیله خروجی با سرعت و ظرفیت وسیله منتقل می‌گردد.

ارتباط بین IOP و وسایل متصله به آن مشابه روش کنترل برنامه در انتقال است. ارتباط با حافظه مشابه دستیابی مستقیم به حافظه است. روشی که بوسیله آن CPU و IOP با هم تبادل داده می‌کند بستگی به سطح پیچیدگی در سیستم دارد. در کامپیوترهای با مقیاس بزرگ (VLS)[2]، هر پردازنده از دیگری مستقل است و هر پردازشگر می‌تواند عملی را آغاز کند. در اکثر سیستم‌های کامپیوتری ، CPU حاکم و IOP یک پردازنده تابع است. آغاز کلیه عملیات به CPU واگذار شده است، ولی دستورالعمل‌های I/O در IOP اجرا می‌شود. دستورالعمل‌های CPU امکان شروع انتقال I/O و نیز امکان تست وضعیت لازم برای تصمیم‌گیری بر انواع فعالیت‌های I/O را فراهم می‌کنند. IOP به نوبه خود با استفاده از وقفه توجه CPU را جلب می‌کند. همچنین به تقاضای CPU با قراردادن کلمه وضعیت در مکان از پیش تعیین شده که بعداً توسط برنامه CPU وارسی می‌شود پاسخ می‌دهد. وقتی که یک عمل I/O لازم شود، CPU به پردازنده IOP اطلاع می‌دهد که درچه مکانی برنامه I/O را بیابد وسپس جزئیات انتقال را به IOP واگذار می‌کند.

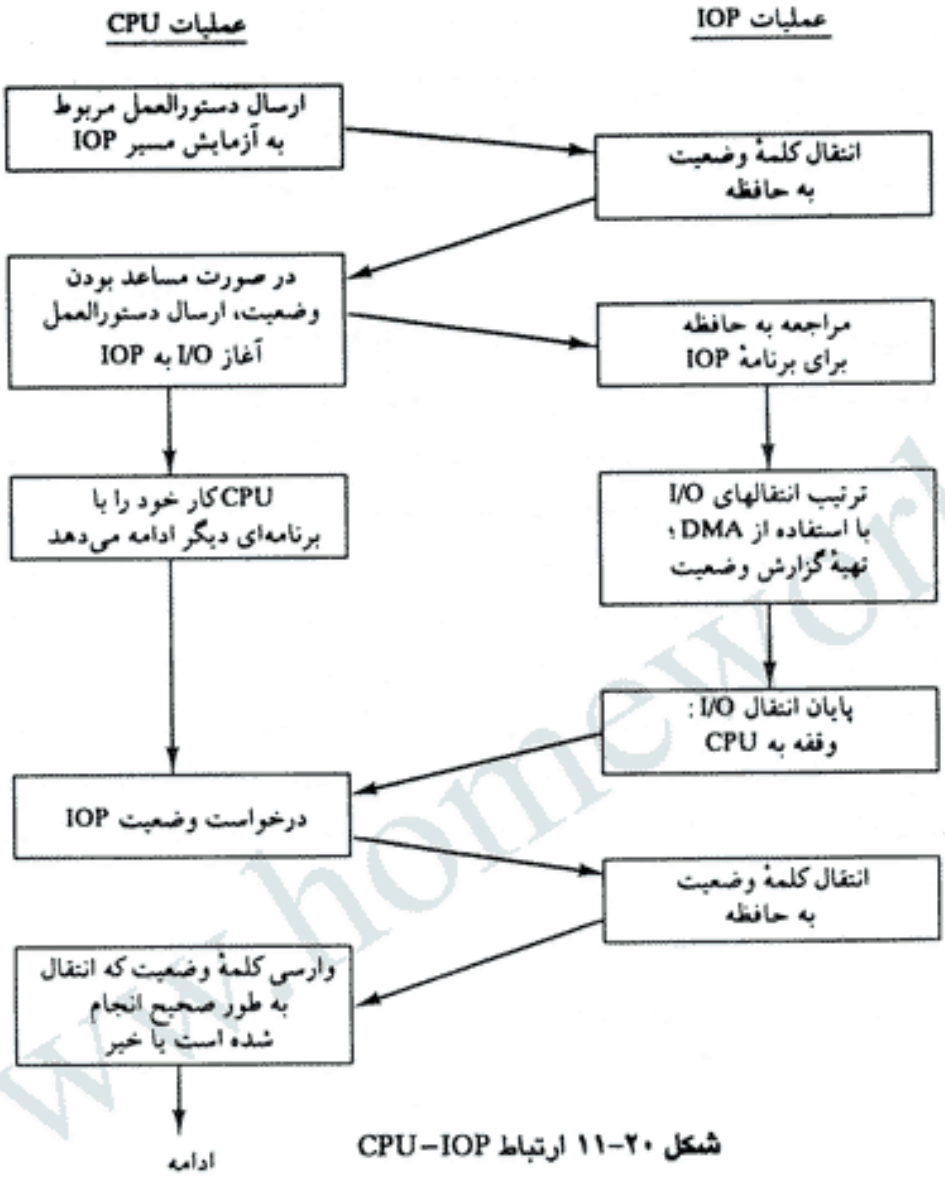
### ارتباط CPU - IOP

تبادل اطلاعات بین CPU و IOP ممکن است بسته به کامپیوتر خاص مورد استفاده، به شکل‌های مختلفی صورت گیرد. در بسیاری از حالات حافظه نقش مرکز پیامی را بازی می‌کند که هر پردازنده اطلاعات را برای دیگری در آن قرار می‌دهد. برای درک بهتر کار یک IOP ، ما روشی را که بوسیله آن CPU و IOP تبادل اطلاعات می‌کنند با مثالی تشریح خواهیم کرد. این مثال که بسیار ساده است از جزئیات بسیاری از اعمال صرفنظر می‌کند تا ایده‌ای کلی از مفاهیم اساسی را فراهم نماید.

رشته اعمال می‌تواند توسط فلوچارت شکل ۲۰-۱۱ نشان داده شود. CPU دستورالعملی را برای تست مسیر IOP ارسال می‌نماید. IOP با درج یک کلمه وضعیت در حافظه که CPU آن را وارسی می‌نماید، به آن پاسخ می‌دهد. بیت‌های کلمه وضعیت شرایط IOP و وسیله I/O را مانند، بار پیش از حد، وسیله با انتقالی دیگر مشغول است، یا وسیله برای انتقال I/O آماده است، مشخص می‌کند. CPU به یک کلمه وضعیت واقع در حافظه مراجعه می‌کند تا ببیند در مرحله بعدی باید چه کاری را انجام دهد. اگر همه چیز مرتب باشد، CPU دستورالعمل را برای شروع انتقال I/O ارسال می‌کند. آدرس حافظه‌ای که بوسیله این دستور دریافت می‌شود به IOP اطلاع می‌دهد که کجا برنامه خود را بیابد.
اکنون CPU می‌تواند با برنامه دیگری کار خود را دنبال کند در حالیکه IOP با برنامه I/O مشغول است.

---

1- Assemble
2- Very - Large - Scale Comrputers




شکل ۲۰-۱۱ ارتباط CPU-IOP

هر دو برنامه توسط انتقال DMA به حافظه مراجعه می‌کنند. وقتی که IOP اجرای برنامه خود را پایان می‌دهد، یک تقاضای وقفه به CPU ارسال می‌کند. CPU با صدور یک دستورالعمل جهت خواندن گزارش وضعیت واصله از IOP به وقفه پاسخ می‌دهد. کلمه وضعیت مشخص می‌کند که انتقال تکمیل شده است و یا اینکه خطایی در حین انتقال اتفاق افتاده است. CPU با بررسی بیت‌های کلمه وضعیت تشخیص می‌دهد که عملیات I/O بطور موفقیت آمیز و بدون خطا پایان یافته است.

IOP از کلیه انتقال داده‌ها بین چند واحد I/O و حافظه مراقبت می‌کند در حالیکه CPU برنامه دیگری را پردازش می‌نماید. IOP و CPU، هر دو برای استفاده از حافظه به آن مراجعه می‌کنند، بنابراین تعداد وسایلی که می‌توانند همزمان کار کنند بوسیله زمان دستیابی به حافظه محدود می‌شود. البته

امکان اشباع حافظه توسط وسایل جانبی I/O ضعیف است، زیرا سرعت وسایل خیلی کمتر از CPU است. با این وجود، برخی واحدهای بسیار سریع، مانند دیسک‌های مغناطیسی، می‌توانند تعداد قابل ملاحظه‌ای از سیکل موجود حافظه استفاده کنند. در این حالت سرعت CPU ممکن است افت کند زیرا در اغلب اوقات مجبور خواهد بود منتظر انجام انتقال‌های حافظه توسط IOP بماند.

## کانال I/O در IBM 370

پردازنده I/O در کامپیوتر IBM370، کانال[1] خوانده می‌شود. پیکربندی این کامپیوتر شامل تعدادی کانال است و هر کانال به یک یا چند وسیله I/O متصل است. سه نوع کانال وجود دارد: مولتی پلکسری، سلکتوری و بلاکی - مولتی پلکسری. کانال مولتی پلکسری را می‌توان به تعدادی وسایل با سرعت پائین یا متوسط وصل کرد و قادر است بطور همزمان با چند وسیله I/O کار کند. کانال سلکتور بصورتی طراحی شده که در هر زمان یک عمل I/O انجام دهد و معمولاً برای کنترل یک وسیله سریع بکار می‌رود. کانال بلاکی - مولتی پلکسری ترکیبی از ویژگی‌های هر دو کانال مولتی پلکسری و سلکتوری را داراست. این کانال با چندین وسیله سریع ارتباط برقرار می‌کند، ولی همه انتقال‌های I/O با همه بلاک داده صورت می‌گیرد، در صورتی که کانال مولتی پلکسر فقط هر لحظه یک بایت را منتقل می‌نماید.

CPU مستقیماً از طریق خطوط کنترل اختصاصی با کانال‌ها و از طریق نواحی رزرو شده ذخیره‌سازی در حافظه بطور غیرمستقیم تبادل اطلاعات می‌کند. شکل ۲۱-۱۱ قالب کلمات مربوط به عملکرد کانال را نشان می‌دهد.

| آدرس وسیله | آدرس کانال | کد عمل |
|---|---|---|

(الف) قالب دستورالعمل‌های I/O

| شمارش | وضعیت | آدرس | کلید |
|---|---|---|---|

(ب) قالب کلمه وضعیت کانال

| شمارش | پرچم‌ها | آدرس داده | کد فرمان |
|---|---|---|---|

(ج) قالب کلمه فرمان کانال

شکل ۲۱-۱۱ قالب‌های کلمات مربوط به I/O در IBM 370

1- Channel

قالب دستورالعمل I/O سه میدان دارد: کد عملیات، آدرس کانال، و آدرس وسیله. سیستم کامپیوتر ممکن است دارای تعدادی کانال باشد، که به هریک آدرسی اختصاص یافته است. بطور مشابه، هر کانال ممکن است به چند وسیله وصل شده باشد که به هر وسیله آدرسی اختصاص می‌یابد. کد عملیات یکی از هشت دستورالعمل I/O را مشخص می‌کند: شروع I/O ، شروع سریع I/O، آزمایش I/O ، پاک کردن I/O توقف I/O ، توقف وسیله، آزمایش کانال، و ذخیره شماره شناسایی کانال. کانالی که آدرس دهی شده به هریک از دستورالعمل‌های I/O پاسخ می‌دهد و آن را اجرا می‌کند. همچنین یکی از چهار کد وضعیتی ممکن را در یک ثبات پردازنده بنام PSW (کلمه وضعیت پردازنده) قرار می‌دهد. CPU می‌تواند با وارسی کد وضعیت موجود در PSW نتیجه عمل I/O را معین کند. معنای چهار کد وضعیتی برای هر دستورالعمل I/O متفاوت است. اما در حالت کلی مشخص می‌کنند که کانال مشغول است یاخیر، عملیاتی است یاخیر.

قالب کلمه وضعیت کانال در شکل ۲۱-۱۱ (ب) نشان داده شده است. این کلمه همیشه در مکان 64 حافظه ذخیره می‌شود. میدان کلید[1]، یک مکانیزم حفاظتی است که برای جلوگیری از مراجعه غیرمجاز از سوی یک کاربر به اطلاعات متعلق به کاربر دیگر یا سیستم عامل بکار می‌رود. میدان آدرس در کلمه وضعیت، آدرس آخرین کلمه استفاده شده بوسیله کانال را نشان می‌دهد. میدان شمارش، تعداد باقیمانده را پس از پایان انتقال نشان می‌دهد. اگر انتقال با موفقیت انجام شود میدان شمارش حاوی مقدار صفر خواهد بود. میدان وضعیت شرایط وسیله و کانال و هرگونه خطایی را که در حین انتقال رخ داده باشد نشان می‌دهد.

تفاوت بین دستورالعمل‌های شروع I/O و شروع سریع I/O این است که اجرای دومی به صرف زمان کمتری از سوی CPU نیاز دارد. وقتی که کانال یکی از این دو دستورالعمل را دریافت کند، به خانه حافظه 72 برای برداشت اولین کلمه فرمان کانال (CCW) مراجعه می‌کند. قالب کلمه فرمان کانال در شکل ۲۱-۱۱ (ج) نشان داده شده است. میدان آدرس داده اولین آدرس بافر حافظه را مشخص می‌کند و میدان شمارش تعداد بایت‌هایی که در انتقال شرکت دارند را معین می‌نماید. میدان فرمان مشخص کننده یک عمل I/O و بیت‌های پرچم اطلاعات اضافی دیگر را برای کانال فراهم می‌کنند. میدان فرمان متعلق به کد عملی است که یکی از شش نوع کار اصلی I/O زیر را مشخص می‌نماید.

۱- نوشتن - انتقال داده ازحافظه به وسیله I/O

۲- خواندن - انتقال داده از وسیله I/O به حافظه

۳- خواندن معکوس - خواندن نوار مغناطیسی بهنگام حرکت معکوس نوار

۴- کنترل - برای آغاز عملی که مستلزم انتقال داده نیست به کار می‌رود، مانند به عقب برگرداندن نوار، یا تغییر مکان مکانیزم دستیابی به دیسک[2]

۵- درک - به کانال اطلاع می‌دهد تا کلمه وضعیت کانال خود را به مکان 64 حافظه منتقل کند.

---

1- key 2- Channel Command word

۶- انتقال در کانال - در عوض دستورالعمل پرش بکار می‌رود. در اینجا میدان آدرس داده، آدرس کلمه فرمان بعدی را که کانال باید اجرا کند مشخص می‌نماید.

مثالی از برنامه کانال در جدول ۳-۱۱ نشان داده شده است. برنامه از سه کلمه فرمان تشکیل شده است. اولین کلمه موجب انتقال 60 بایت از حافظه با آدرس شروع 4000 به یک نوار مغناطیسی می‌گردد. دو کلمه فرمان بعدی عمل مشابهی را با بخش متفاوتی از حافظه و بایت شمارش انجام می‌دهند. شش پرچم در هر کلمه کنترل روابط دو جانبه‌ای را بین کلمات فرمان معین می‌کنند.

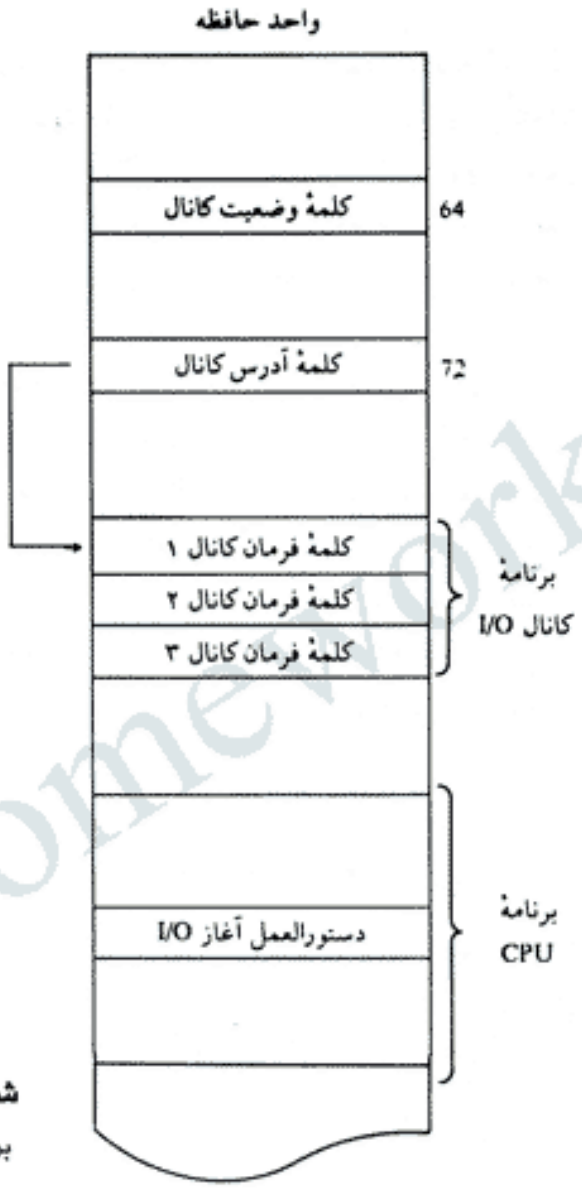
پرچم اول در اولین کلمه فرمان برابر 1 قرار داده می‌شود تا زنجیره‌ای شدن داده‌ها را مشخص کند. این امر منجر به ترکیب 60 بایت فرمان اول با 20 بایت فرمان بعدی آن و تشکیل یک رکورد 80 بایتی می‌شوند. این 80 بایت حتی اگر از دو بخش حافظه استفاده شده باشند بدون هر گونه جداسازی یا فاصله بر روی نوار نوشته می‌شوند. پرچم دوم در فرمان دوم 1 می‌گردد تا مشخص کننده زنجیره‌ای شدن فرمانها[1] باشد. این پرچم کانال را مطلع می‌سازد که کلمه فرمان بعدی از همان وسیله I/O، یعنی در اینجا نوار، استفاده خواهد شد. کانال به دستگاه نوار اطلاع می‌دهد که شروع به وارد کردن یک فاصله رکورد روی نوار کند و سپس به خواندن فرمان بعدی از حافظه بپردازد. سپس 40 بایت از سومین فرمان بصورت یک رکورد جداگانه بر روی نوارنوشته می‌شود. هر گاه همه پرچمها صفر باشند، نشان دهنده پایان عملیات I/O برای وسیله خاص مورد نظر است.

نقشه‌ای از حافظه که همه اطلاعات مربوط به پردازش I/O را نشان می‌دهد در شکل ۲۲-۱۱ رسم شده است. عملیات وقتی آغاز می‌شود که برنامه CPU با یک دستورالعمل شروع I/O مواجه شود. سپس IOP به مکان 72 حافظه مراجعه کرده و کلمه آدرس کانال را از آنجا برمی‌دارد. این کلمه حاوی آدرس شروع برنامه کانال I/O است. سپس کانال کار خود را با اجرای برنامه مشخص شده با کلمات فرمان کانال ادامه می‌دهد. کانال در حین انتقال یک کلمه وضعیت را تشکیل و آن را در مکان 64 ذخیره می‌کند. به محض وقوع وقفه CPU می‌تواند برای کلمه وضعیت به مکان 64 حافظه مراجعه نماید.

جدول ۳-۱۱ مثالی از برنامه کانال برای IBM 370

| شمارش | پرچم‌ها | آدرس | وقفه |
|---|---|---|---|
| 60 | 100000 | 4000 | نوشتن نوار |
| 20 | 010000 | 6000 | نوشتن نوار |
| 40 | 000000 | 3000 | نوشتن نوار |

1- Command chaning




**شکل ۲۲-۱۱** محل اطلاعات در حافظه برای عملیات I/O در IBM 370

## IOP مدل Intel 8089

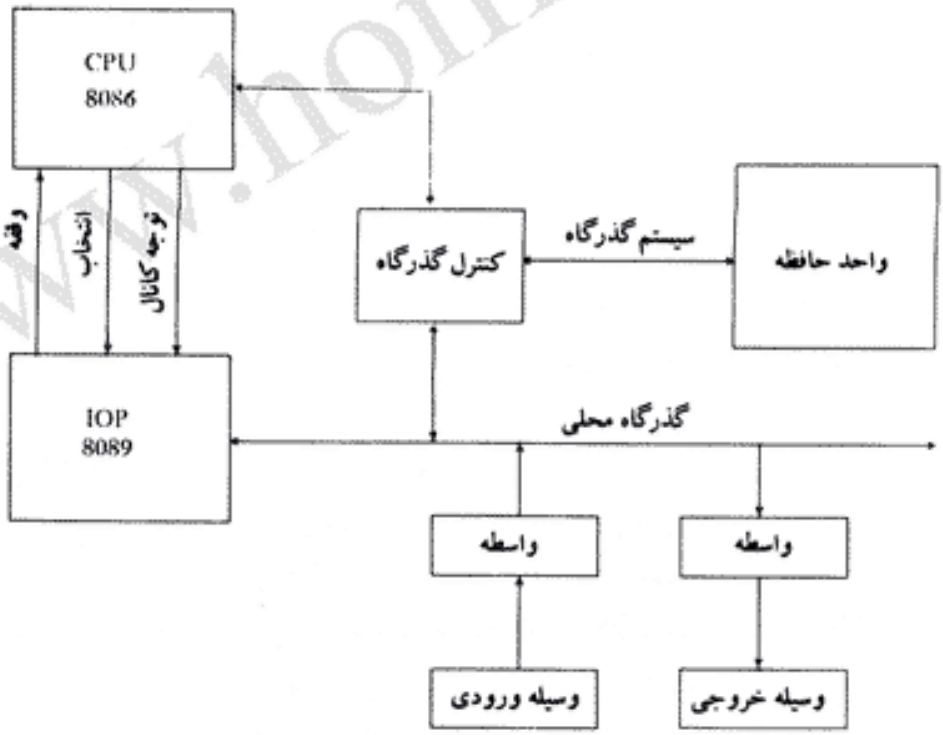
پردازنده I/O مدل 8089 در یک بسته مدار مجتمع 40 پایه‌ای قرار دارد. در داخل 8089 دو واحد مستقل به نام کانال وجود دارد. هر کانال مشخصات و ویژه‌گی‌های عمومی یک واحد پردازنده را همراه با امکان یک کنترل‌کننده با دستیابی مستقیم به حافظه[1] (DMAC) یکجا داراست. 8089 به صورتی طراحی شده است که در یک سیستم کامپیوتری، با ریزپردازنده Intel 8086 که نقش CPU را دارد، به عنوان IOP عمل کند. پردازنده 8086 با ایجاد یک پیام در حافظه که انجام کار موردنظر را معین می‌نماید، اقدام به یک عمل I/O می‌کند. پردازنده 8089 این پیام را از حافظه می‌خواند، عمل را انجام می‌دهد، و

1- Direct Memory Acces Controller

پس از اتمام کار به CPU اطلاع می‌دهد.

بر خلاف کانال IBM 370، که فقط دارای شش فرمان اصلی I/O است، پردازنده IOP 8089، پنجاه دستورالعمل اصلی دارد که می‌توانند روی بیت‌ها بطور جداگانه، روی بایت‌ها، یا روی کلمات 16 بیتی عمل کنند. IOP می‌تواند برنامه‌ها را مشابه با یک CPU اجرا نماید، جز اینکه مجموعه دستورالعمل‌ها خصوصاً برای انجام یک عمل ورود و خروج کارا انتخاب شده‌اند. این مجموعه دستورالعمل‌ها شامل دستورات عمومی انتقال داده، عملیات پایه حسابی و منطقی، اعمال انشعاب شرطی و غیرشرطی، و قابلیت فراخوانی و بازگشت از زیرروال است. این مجموعه، دستورالعمل‌های خاصی هم برای مقدار دهی اولیه برای انتقال DMA و صدور تقاضای وقفه برای CPU دارد. همچنین برای انتقال داده بین هر دو وسیله متصل به گذرگاه سیستم، مانند I/O به حافظه، حافظه به حافظه، یا I/O به I/O کارآیی خوبی دارد.

یک سیستم ریزکامپیوتری که از جفت مدار مجتمع 8086/8089 استفاده می‌کند در شکل ۲۳-۱۱ نشان داده شده است. 8086 مثل یک CPU و 8089 مانند یک IOP عمل می‌کنند.هر دو واحد از طریق سیستم گذرگاهی که بوسیله اینتل گذرگاه چندگانه[1] نامیده می‌شود، از حافظه مشترکی استفاده می‌کنند. IOP با استفاده از یک گذرگاه محلی با واحدهای واسطه که به وسایل I/O متصلند تبادل اطلاعات می‌کنند. CPU با فعال کردن خط توجه کانال[2] ارتباط برقرار می‌کند خط انتخاب که بوسیله CPU بکار




شکل ۲۳-۱۱ بلاک دیاگرام سیستم ریزکامپیوتر 8086/8089 اینتل

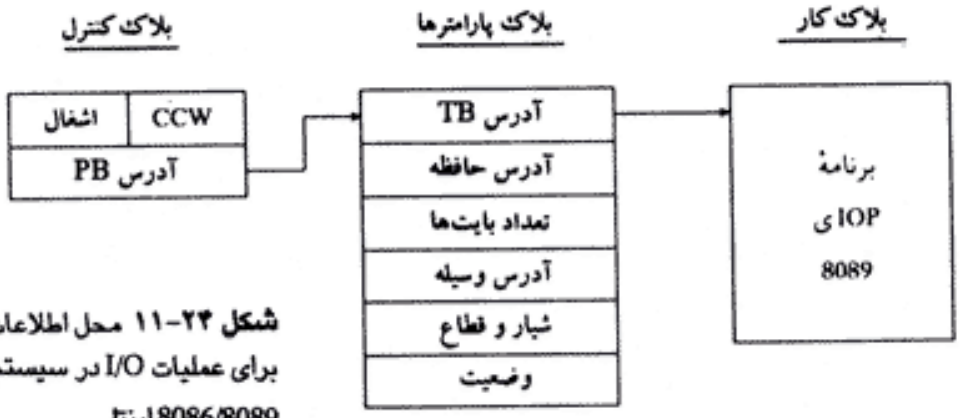
1- Multibus
2- Channel Attention

می‌رود برای انتخاب یکی از دو کانال در 8089 بکار می‌رود. IOP توجه CPU را با ارسال یک تقاضای وقفه جلب می‌کند.

CPU و IOP، با نوشتن پیام برای یکدیگر در سیستم حافظه، باهم ارتباط برقرار می‌کنند. CPU مکان پیام را آماده کرده و با فعال کردن خط توجه کانال IOP را آگاه می‌سازد. IOP پیام را خوانده، عملیات I/O درخواستی را انجام می‌دهد، وبرنامه کانال مناسب را اجرا می‌نماید. وقتی که کانال برنامه خود را بپایان رساند، یک تقاضا وقفه برای CPU ارسال می‌دارد.

این روش ارتباط، متشکل از برنامه‌های تقسیم شده‌ای است که بلاک[1] نامیده شده و طبق شکل ۲۴-۱۱ در حافظه ذخیره می‌شوند. هر بلاک حاوی اطلاعات کنترل و پارامترها و آدرس اشاره‌گر به بلاک بعدی است. آدرس بلاک کنترل در آغاز به هر کانال IOP داده می‌شود. پرچم اشغال بودن مشخص می‌کند که آیا IOP مشغول است یا آماده انجام یک عمل I/O جدید می‌باشد. CCW (کلمه فرمان کانال)[2] توسط CPU برای مشخص کردن نوع عملی که IOP باید انجام دهد معین می‌شود. CCW در 8089 مفهوم مشابهی با کلمه فرمان در کانال IBM ندارد. در اینجا CCW بیشتر شبیه یک دستورالعمل I/O است که یک عمل را برای IOP، مشخص می‌کند، بلاک پارامتر حاوی داده‌های متغیری است که برنامه IOP برای اجرای کار خود از آن استفاده می‌کند و بلاک کار[3] (تکلیف) حاوی برنامه واقعی است که باید در IOP اجرا شود.

CPU و IOP از طریق کنترل بلاک‌های پارامترها با یکدیگر کار می‌کنند. CPU پس از تست پرچم اشغال و اطمینان از در اختیار بودن IOP، استفاده از حافظه مشترک رابدست می‌گیرد. سپس اطلاعات لازم را دربلاک پارامترها پرمی‌کند و یک فرمان "شروع عمل" در CCW می‌نویسد. پس از برپایی بلاک تبادل اطلاعات[4]، CPU سیگنال "توجه کانال" را فعال می‌کند تا IOP را برای آغاز عمل I/O مطلع سازد. سپس CPU با برنامه دیگری بکار ادامه می‌دهد. IOP با قراردادن آدرس بلاک کنترل در شمارنده




**شکل ۲۴-۱۱ محل اطلاعات در حافظه برای عملیات I/O در سیستم کامپیوتری 8086/8089 اینتل**

1- Block
2- Channel Command Word
3- Task Block
4- Communication Block

برنامه‌اش، به سیگنال توجه کانال پاسخ می‌دهد. IOP به بلاک کنترل مراجعه کرده و پرچم اشغال را 1 می‌کند. پس از آن، عمل موجود در CCW را وارسی می‌نماید. سپس آدرس PB (بلاک پارامتر) و آدرس TB (بلاک تکلیف)[1] را به ثبات داخلی IOP منتقل می‌سازد. IOP با استفاده از اطلاعات موجود در بلاک پارامتر شروع به اجرای برنامه در بلاک تکلیف می‌کند. وارده‌های موجود در بلاک پارامترها به وسیله I/O بستگی دارد. پارامترهای لیست شده در شکل ۲۴-۱۱ برای انتقال داده به دیسک مغناطیسی و یا از آن، مناسبند. آدرس حافظه، آدرس شروع یک بافر حافظه است. شمارش بایت تعداد بایت‌های انتقال یافته را معین می‌نماید. آدرس وسیله، وسیله I/O خاصی که قرار است بکار گرفته شود را مشخص می‌کند. شماره‌های شیار[2] و قطاع داده را روی دیسک قرار می‌دهد. وقتی که عمل I/O پایان یابد، IOP بیت‌های وضعیت خود را در مکان کلمه وضعیت بلاک پارامتر قرار می‌دهد و به CPU سیگنال وقفه می‌فرستند. CPU به کلمه وضعیت رجوع کرده و انتقال موفقیت آمیز را وارسی می‌نماید.

## ۱۱-۸ تبادل اطلاعات سری

یک پردازنده تبادل داده‌ها یک پردازنده I/O است که داده را از پایانه‌های متعدد دور از هم که از طریق خطوط تلفن و دیگر خطوط ارتباطی به یکدیگر متصلند توزیع و یا جمع آوری می‌کند. این پردازنده نوعی پردازنده خاص I/O است که برای ارتباط مستقیم با شبکه‌های ارتباطی طراحی شده است. یک شبکه ارتباطی ممکن است از هر نوع وسیله متنوعی مانند چاپگر، وسایل نمایش محاوره‌ای، سنسورهای دیجیتال یا یک مرکز کامپیوتر دوردست متشکل باشد. با استفاده از یک پردازنده انتقال داده[3]، کامپیوتر می‌تواند بطور متناوب به بخشی از هر تقاضای شبکه سرویس بدهد و بنابراین رفتار ظاهری آن بصورتی باشد که به کاربران متعدد بطور همزمان سرویس می‌دهد. بااین ترتیب کامپیوتر قادر خواهد بود تا بطور موثری در محیط‌های تقسیم زمانی[4] بکار رود.

مهمترین اختلاف بین یک پردازنده I/O و یک پردازنده انتقال داده، نحوه ارتباط پردازنده با وسیله I/O است. یک پردازنده I/O از طریق یک گذرگاه مشترک I/O که از چندین خط داده و کنترل تشکیل شده است با وسایل I/O ارتباط برقرار می‌کند. تمام وسایل جانبی از گذرگاه بطور مشترک استفاده می‌کنند و از آن برای انتقال اطلاعات به و یا از پردازنده I/O استفاده می‌کنند. یک پردازنده ارتباطی (انتقال داده) از طریق یک جفت سیم باهر پایانه ارتباط برقرار می‌نماید. هر دو نوع اطلاعات داده و کنترل به فرم سری در آمده که این موجب سرعت انتقال بسیار کندتری می‌شود. وظیفه پردازنده انتقال داده، ارسال و جمع آوری اطلاعات دیجیتال به و یا از هر پایانه، تعیین اینکه اطلاعات از نوع داده است یا کنترلی و پاسخ به تمام تقاضاها طبق رویه‌های از پیش تعیین شده است. مسلماً پردازنده باید با CPU و حافظه هم بطور مشابه به هر پردازنده I/O ارتباط برقرار کند.

---

1- Task Block 2- Track

3- Data Communication Processor 4- Time - sharing

روش اتصال پایانه‌های دوردست به پردازنده انتقال داده از طریق خطوط تلفن یا سایر وسایل ارتباط خصوصی یا عمومی است. چون خطوط تلفن اصولاً برای تبادل صوت طراحی شده و کامپیوترها با سیگنال‌های دیجیتال با هم تبادل اطلاعات می‌کنند، نوعی تبدیل باید صورت گیرد. مبدّل‌ها، دستگاه داده[1]، کوپل‌کننده صوتی[2] و یا مودم[3] (مخفف مودولاتور و دمودولاتور) خوانده می‌شوند. مودم سیگنال دیجیتال را به علائم صوتی برای ارسال برروی خط تلفن تبدیل می‌کند. روش‌های مدولاسیون مورد استفاده و نیز درجه کیفیت رسانه‌های ارتباطی و سرعتهای انتقال در بین آنها متفاوت است. بسته به روش ارسال پایانه دوردست، خط ارتباطی ممکن است به یک واسطه همزمان یا غیرهمزمان متصل باشد. واسطه غیرهمزمان، داده‌های سری را با بیت‌های شروع و توقف در هر کاراکتر مطابق شکل ۷-۱۱ دریافت می‌کند. این نوع واسطه مشابه واحد واسطه ارتباطی غیرهمزمان در شکل ۸-۱۱ است.

انتقال همزمان (همگام) بیت شروع و توقف[4] را برای کادربندی کاراکترها استفاده نمی‌کند، و بنابراین از ارتباط جهت تبادل داده‌ها بصورت کاراکتری استفاده می‌شود. وسایلی که سرعت بالا دارند از انتقال همزمان برای رسیدن به این کارآیی استفاده می‌کنند. مودم‌هایی که در انتقال همزمان بکار می‌روند، دارای ساعت درونی بوده و این ساعت‌ها با فرکانس انتقال در خط ارتباط تنظیم می‌شوند. برای عملکرد صحیح مودم، لازم است ساعت در فرستنده و گیرنده در تمام مدت همگام باشند. با این وجود خط ارتباطی فقط حاوی بیت‌های داده است که از آنها اطلاعات ساعت باید استخراج شود. همگامی فرکانس توسط مودم گیرنده از گذارهای سیگنال که در داده‌های دریافتی رخ می‌دهد حاصل می‌گردد. هر شیفت فرکانسی که ممکن است بین ساعت‌های گیرنده و فرستنده رخ دهد بطور مداوم با نگهداشتن ساعت فرستنده در فرکانس رشته بیت‌های رسیده تنظیم می‌شود. مودم داده دریافتی را همراه با ساعت فرستنده به واحد واسطه می‌فرستد. واسطه یا پایانه در سمت فرستنده هم از اطلاعات ساعت مودم خود استفاده می‌کند. به این ترتیب در فرستنده و گیرنده سرعت بیت یکسانی برقرار می‌شود.

برخلاف ارسال غیرهمگام که در آن هر کاراکتر را می‌توان جداگانه همراه با بیت‌های شروع و توقف آنها ارسال کرد، انتقال همگام برای حفظ همگامی باید پیام پیوسته‌ای را ارسال نماید. پیام شامل گروهی از بیت‌های ارسال شده سری مانند یک بلاک از داده است. کل بلاک همراه با کاراکترهای کنترلی خاص در ابتدا و انتهای بلاک ارسال می‌شود. کاراکترهای کنترل در ابتدای بلاک اطلاعات لازم را برای تفکیک بیت‌های دریافتی به صورت کاراکترهای مستقل فراهم می‌کنند.

یکی از وظایف پردازنده انتقال داده وارسی خطاهای ارسال است. یک نوع خطا را با وارسی بیت توازن در هر کاراکتر رسیده می‌توان تشخیص داد. روش دیگر که در پایانه‌های غیرهمگام با اپراتوری انسانی بکار می‌رود انعکاس کاراکتر است. کاراکتری که از صفحه کلید به کامپیوتر ارسال می‌شود بوسیله پردازنده تشخیص داده شده و سپس به پایانه چاپگر منعکس می‌شود. اگر کاراکتر چاپ شده همان

---

1- Data set 2- Acoustion Coupler 3- Modem
4- Start–Stop

کاراکتری که از فشار کلید حاصل شده است نباشد، اپراتور درمی‌یابد که در حین ارسال خطایی رخ داده است.

در ارسال همگام، که در آن کل بلاک کاراکترها ارسال می‌شود، هر کاراکتر دارای یک بیت توازن برای وراسی توسط گیرنده است. پس از ارسال تمام بلاک، فرستنده یک کاراکتر اضافی که توازن را در طول پیام تشکیل می‌دهد ارسال می‌دارد. این کاراکتر وارسی تجمعی طولی [1] (LRC) نامیده شده و در واقع تجمع OR انحصاری تمام کاراکترهای ارسال شده است. ایستگاه گیرنده LRC را ضمن دریافت کاراکترها محاسبه می‌کند و آنرا با LRC ارسال شده مقایسه می‌نماید. برای پیام‌های بدون خطا، LRC محاسبه شده و دریافت شده با هم مساویند. اگر گیرنده در بلاک ارسال شده خطایی را بیابد، به فرستنده اطلاع می‌دهد تا یکبار دیگر همان اطلاعات را ارسال کند. روش بکار رفته دیگر برای وارسی خطاها در ارسال اطلاعات، وارسی تجمعی چرخشی [2] (CRC) است. این یک کد چند جمله‌ای حاصل از بیت‌های پیام است که با پاس دادن آنها از یک ثبات شیفت حاوی تعدادی گیت‌های OR انحصاری به فرستنده پسخورد (برگشت [3]) حاصل می‌شود. این نوع کد برای آشکار سازی خطاهای پیاپی در کانال ارتباطی مناسب است.

انتقال داده بین دو نقطه می‌تواند به سه طریق انجام شود. یک طرفه [4] نیمه دو طرفه و تمام دوطرفه. یک خط یک طرفه اطلاعات را فقط در یک جهت حمل می‌کند. این روش بندرت در خطوط ارتباطی بکار می‌رود زیرا گیرنده نمی‌تواند در صورت بروز خطا با فرستنده ارتباط برقرار کند. مثال‌هایی از این نوع ارتباط پیام‌های رادیوی و تلویزیونی است.

سیستم ارسال نیمه دوطرفه سیستمی است که قادر است اطلاعات را در هر دو جهت ارسال کند ولی در هر لحظه اطلاعات فقط در یک جهت منتقل می‌گردد. برای این روش یک جفت سیم لازم است. یک نوع متداول از این روش، استفاده از یک مودم بعنوان فرستنده و مودم دیگر بعنوان گیرنده است. وقتی که ارسال پیام در یک جهت پایان یابد، نقش مودم‌ها برای معکوس کردن جهت انتقال تعویض می‌گردد. زمان لازم برای تعویض خط سیستم نیمه دو طرفه [5] از یک جهت به جهت دیگر زمان تحویل خوانده می‌شود.

سیستم تمام دوطرفه قادر است که بطور همزمان داده را ارسال و دریافت کند. این عمل توسط خط چهار سیمی صورت می‌گیرد که در آن هر دو خط برای یکی از دو جهت در نظر گرفته شده است. راه دیگر، استفاده از یک مدار دو سیمه برای ارسال تمام دوطرفه است مشروط براین که طیف فرکانسی به دو بخش باند فرکانس غیرهمپوش [6] تقسیم شود تا در یک جفت سیم بتوان کانال‌های ارسال و دریافت جداگانه بوجود آورد.

خطوط ارتباطی، مودم‌ها و سایر تجهیزات بکار رفته در ارسال داده‌ها بین دو یا چند ایستگاه خطوط

---

1- Longitudinal Redundancy Check
2- Cyclic Redundancy Check
3- Feedback
4- simplex
5- Turnaround
6- Nonoverlap

ارتباط داده[1] نامیده می‌شود. ارسال منظم اطلاعات در یک خط ارتباط داده توسط یک قرارداد[2] انجام می‌شود. قرارداد کنترل یک خط ارتباط داده مجموعه‌ای از قوانین است که کامپیوترها و پایانه‌های متصل بهم برای اطمینان از یک انتقال صحیح اطلاعات از آن پیروی می‌کنند. هدف از قرارداد ارتباط داده برقراری و اتمام ارتباط بین دو ایستگاه، شناسایی فرستنده و گیرنده، اطمینان از ارسال صحیح و بدون خطای همه پیام‌ها، و انجام همه عملیات کنترلی لازم در انتقال رشته‌ای از داده‌هاست. قراردادها برحسب روش کادربندی پیام‌ها[3] به دو گروه عمده تقسیم می‌شوند. این دو گروه عبارتند از قرارداد مبتنی بر کاراکتر و قرارداد مبتنی بر بیت.

## قرارداد مبتنی بر کاراکتر

قرارداد مبتنی بر کاراکتر بر اساس کد دودویی مجموعه کاراکترها ایجاد شده است. متداول‌ترین کد مورد استفاده کد اسکی ASCII (کد استاندارد آمریکایی برای تبادل اطلاعات) است. این کد 7 بیتی است و بیت هشتم برای توازن بکار می‌رود. کد دارای 128 کاراکتر است که از 95 کد گرافیکی و 33 کاراکتر کنترلی تشکیل شده است. کاراکترهای گرافیکی شامل حروف بزرگ، کوچک، ده رقم اعداد و انواع سمبل‌های خاص است. لیستی از کاراکترهای ASCII در جدول ۱-۱۱ قابل ملاحظه است. کاراکترهای کنترل برای مسیردهی داده‌ها، آرایش متن با قالب مورد نظر، و صفحه‌آرایی[4] متن چاپی بکار می‌روند. کاراکترهایی که ارسال داده‌ها را کنترل می‌کنند، کاراکترهای کنترل ارتباطات[5] خوانده می‌شوند. این کاراکترها در جدول ۴-۱۱ لیست شده‌اند. هر کاراکتر 7 بیتی است و با یک سمبل سه حرفی معرفی شده است. نقش هر کاراکتر در کنترل ارسال داده بطور خلاصه در ستون نقش در جدول آمده است.

جدول ۴-۱۱ کاراکترهای کنترل ارتباطات اسکی

| کد | سمبل | عملکرد | معنی |
|---|---|---|---|
| 0010110 | SYN | در حالت بیکار، همگامی تولید می‌کند | بیکار و همگام |
| 0000001 | SOH | سرآیند بلاک پیام | آغاز سرآیند |
| 0000010 | STX | قبل از بلاک متن قرار گیرد | آغاز متن |
| 0000011 | ETX | در پایان بلاک متن قرار گیرد | پایان متن |
| 0000100 | EOT | عمل ارسال را پایان می‌دهد | پایان ارسال |
| 0000110 | ACK | تصدیق مثبت | تصدیق |
| 0010101 | NAK | تصدیق منفی | تصدیق منفی |
| 0000101 | ENQ | استعلام روشن بودن پایانه | استعلام |
| 0010111 | ETB | پایان بلاک داده | پایان بلاک ارسال |
| 0010000 | DLE | کاراکتر کنترلی خاص | کنترل رابط داده |

1- Data Link
2- Protocol
3- Message–Framing
4- Layout
5- Communication Control Character

کاراکتر SYN نقش همگام‌ساز را بین فرستنده و گیرنده بعهده دارد. وقتی که از کد 7 بیتی اسکی همراه با بیت توازن فرد در با ارزشترین مکان استفاده شود، کاراکتر SYN دارای کد هشت بیتی 00010110 خواهد بود و این خاصیت را دارد که با استفاده از شیفت حلقوی، فقط پس از هشت بار چرخش خود را تکرار می‌کند. وقتی که فرستنده ارسال کاراکتر 8 بیتی را آغاز می‌کند، ابتدا چند کاراکتر را فرستاده و سپس اقدام به فرستادن پیام واقعی می‌کند. رشته بیت‌های متوالی اولیه توسط گیرنده پذیرفته شده و به منظور یافتن کاراکتر SYN وارسی می‌شوند. به بیان دیگر، با هر پالس ساعت، گیرنده هشت بیت آخر دریافتی را وارسی می‌کند. اگر این هشت بیت با کاراکتر SYN مطابقت نکند، گیرنده بیت بعدی را می‌پذیرد، بیت بالا رتبه قبلی را حذف می‌نماید، و مجدداً هشت بیت آخر را برای یافتن کاراکتر SYN چک می‌کند. این عمل پس از هر پالس ساعت تکرار شده و بیت‌ها تا یافتن کاراکتر SYN دریافت می‌شوند. به محض تشخیص کاراکتر SYN، گیرنده کاراکتر راکادربندی می‌کند. از این پس گیرنده هر هشت بیت را شمارش کرده و آنرا بعنوان یک کاراکتر در نظر می‌گیرد. معمولاً گیرنده دو کاراکتر SYN متوالی را چک می‌کند تا هرگونه تردیدی در این مورد، که مبادا اولی در نتیجه سیگنال نویز روی خط باشد، را رفع کند. بعلاوه، وقتی که فرستنده بیکار است و هیچ پیام یا کاراکتری را ارسال نمی‌کند. رشته پیوسته‌ای از کاراکتر SYN را ارسال می‌نماید. گیرنده این کاراکترها را بعنوان شرایطی برای همگام سازی خطا می‌شناسد و به یک حالت بیکار و همگام می‌رود. در این حالت، هر دو واحد، همزمانی بیت و کاراکتر را حتی اگر اطلاعات معنی‌داری مخابره نشود حفظ می‌کنند.

پیام‌ها از طریق خط ارتباط داده همراه با قالبی متشکل از میدان سرآیند[1]، میدان متن، میدان وارسی خط ارسال می‌شوند. نمونه‌ای از قالب پیام برای قرارداد مبتنی بر کاراکتر در شکل ۲۵-۱۱ نشان داده شده است. دو کاراکتر SYN همگامی صحیح را در آغاز پیام تضمین می‌کنند. بدنبال کاراکترهای SYN، سرآیند قراردارد، که با کاراکتر SOH[2] شروع می‌شود (شروع سرآیند). سرآیند از اطلاعات آدرس و کنترلی تشکیل شده است. کاراکتر STX سرآیند را پایان داده و شروع متن را معین می‌نماید. بخش متن پیام از نظر طول متغیر بوده و می‌تواند حاوی هر کاراکتر اسکی بجز کاراکترهای کنترلی باشد. میدان متن با ETX پایان می‌پذیرد. آخرین میدان، یک کاراکتر وارسی بلاک[3] (BCC) است که برای وارسی خطا بکار می‌رود. این میدان معمولاً وارسی تجمعی طولی (LRC) یا وارسی تجمعی چرخشی (CRC) است. گیرنده پیام را دریافت کرده و BCC خود را محاسبه می‌کند. اگر BCC ارسالی با BCC محاسبه شود

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SYN | SYN | SOH | سرآیند | STX | متن | ETX | BCC |

**شکل ۲۵-۱۱ نوعی قالب پیام در قرارداد مبتنی بر کاراکتر**

1- Header field 2- Start of Header 3- Block Check Character

بوسیله گیرنده همخوانی نداشته باشد، گیرنده با ارسال یک کاراکتر تصدیق منفی[1] (NAK) واکنش نشان می‌دهد. پیام، سپس مجدداً ارسال و وارسی می‌شود. معمولاً قبل از اینکه خط اشکال‌دار فرض شود، ارسال مجدد پیام چند بار تکرار می‌شود. هنگامی که BCC ارسالی؛ محاسبه شده بوسیله گیرنده مطابقت نماید، پاسخ تصدیق مثبت بوده و کاراکتر ACK استفاده می‌شود.

## مثال ارسال داده

برای درک عملکرد یک پردازنده انتقال داده، اجازه بدهید روشی را که یک پایانه و پردازنده با هم ارتباط برقرار می‌کنند مثال بزنیم. تبادل اطلاعات بین واحد حافظه و CPU مشابه با هر پردازنده I/O است.

نمونه پیامی که ممکن است از یک پایانه به پردازنده ارسال شود در جدول ۵-۱۱ دیده می‌شود. نگاهی به این پیام مشخص می‌نماید که تعدادی کاراکترهای کنترلی برای شکل دادن به پیام بکار رفته است. هر کاراکتر، از جمله کاراکترهای کنترلی بصورت کد دودویی هشت بیتی شامل 7 بیت کد اسکی بعلاوه یک بیت توازن فرد در با ارزشترین مکان، بطور سری ارسال می‌شوند. برای همگامی فرستنده و گیرنده دو کاراکتر SYN بکار رفته است. سرآیند با کاراکتر SOH شروع می‌شود و به دو کاراکتر که آدرس پایانه را معین می‌کنند ختم می‌گردد. در این مثال خاص، آدرس $T_4$ است، ولی عموماً می‌تواند مجموعه‌ای از دو یا چند کاراکتر گرافیکی باشد. کاراکتر STX سرآیند را ختم کرده و شروع ارسال متن را معین می‌نماید. داده متن در اینجا "request of balance number 1234" است. تک تک کاراکترهای این پیام

**جدول ۵-۱۱** نمونه پیام ارسالی از یک پایانه به پردازنده

| کد | نماد | شرح |
|---|---|---|
| 0001 0110 | SYN | اولین کاراکتر Sync |
| 0001 0110 | SYN | دومین کاراکتر Sync |
| 0000 0001 | SOH | آغاز سرآیند |
| 0101 0100 | T | آدرس پایانه $T_4$ است |
| 0011 0100 | 4 | |
| 0000 0010 | STX | آغاز ارسال متن |
| 0101 0010 | | |
| 0100 0101 | request | متن ارسالی درخواست اعلام موجودی |
| . | balance | شماره حساب ۱۲۳۴ |
| . | of accoun | |
| . | No. 1234 | |
| 1011 0011 | | |
| 0011 0100 | | |
| 1000 0011 | ETX | پایان ارسال متن |
| 0111 0000 | LRC | کاراکتر توازن طولی |

1- Negative Acknowledge

در این جدول لیست نشده‌اند زیرا فضای بسیاری را اشغال می‌کنند. با این وجود، باید توجه داشت که هر کاراکتر در پیام دارای کد 8 بیتی است و هر بیت بطور سری ارسال می‌شود. کاراکتر کنترل EXT پایان کاراکترهای متن را مشخص می‌نماید. کاراکتر بعدی که بدنبال ETX آمده است یک کاراکتر وارسی طولی، LRC، است هر بیت در این کاراکتر بیت توازنی است که از محاسبه تمام بیت‌های واقع در یک ستون در بخش کد از جدول محاسبه می‌شود.

پردازنده انتقال داده، این پیام را دریافت کرده و به تحلیل آن می‌پردازد. پردازنده تشخیص می‌دهد که پایانه $T_4$ است و متن پیام را ذخیره می‌کند. در حین دریافت کاراکترها، پردازنده توازن را در هر کاراکتر وارسی کرده و توازن طولی را نیز محاسبه می‌کند. LRC محاسبه شده با LRC کاراکتر دریافتی مقایسه می‌شود. اگر هر دو با هم مطابقت داشته باشند، یک تصدیق مثبت (ACK) به پایانه بازگردانده می‌شود. اگر عدم تطابق وجود داشته باشد، یک تصدیق منفی به پایانه برگردانده می‌شود، که یک ارسال مجدد از همان بلاک را آغاز خواهد کرد. اگر پردازنده پیام را بدون خطا بیابد، آنرا به حافظه انتقال داده و CPU را وقفه می‌دهد. هنگامی که CPU وقفه را تصدیق نماید، پیام را تحلیل کرده و متن پیام را برای پاسخ به تقاضا آماده می‌کند. CPU یک دستورالعمل را برای پردازنده انتقال داده می‌فرستد تا پیام را به پایانه ارسال دارد.

نمونه پاسخ از پردازنده به پایانه در جدول ۶-۱۱ لیست شده است. پس از دو کاراکتر SYN، پردازنده پیام قبلی را با یک کاراکتر ACK تصدیق می‌کند. خط با کاراکتر SYN در حالت بیکاری قرار

**جدول ۶-۱۱** نمونه پیان ارسالی از پردازنده به پایانه

| کد | نماد | شرح |
|---|---|---|
| 0001 0110 | SYN | اولین کاراکتر Sync |
| 0001 0110 | SYN | دومین کاراکتر Sync |
| 1000 0110 | ACK | پردازنده دریافت پیام قبلی را تصدیق می‌کند |
| 0001 0110 | SYN | خط بیکار است |
| . | . | |
| . | . | |
| . | . | |
| 0001 0110 | SYN | خط بیکار است |
| 0000 0001 | SOH | آغاز سرآیند |
| 0101 0100 | T | آدرس پایانه $T_4$ است |
| 0011 0100 | 4 | |
| 0000 0010 | STX | آغاز ارسال متن |
| 1100 0010 | | |
| 1100 0001 | balance | متن ارسالی پاسخی از سوی کامپیوتر |
| . | is | است که موجودی حساب را می‌دهد |
| . | $100.00 | |
| . | | |
| . | | |
| 1011 0000 | | |
| 1000 0011 | ETX | پایان ارسال متن |
| 1101 0101 | LRC | کاراکتر توازن طولی |

می‌گیرد و منتظر دریافت پاسخ می‌ماند. پیام دریافتی از CPU توسط پردازنده با گنجاندن کاراکترهای کنترلی لازم در ابتدا و انتهای متن به قالب صحیحی درمی‌آید. پیام دارای سرآیند SOH و آدرس پایانه $T_4$ برای پایانه است. متن پیام پایانه را مطلع می‌سازد که بالانس 100$ است. یک کاراکتر LRC محاسبه شده و به پایانه ارسال می‌شود. اگر پایانه با کاراکتر NAK پاسخ دهد، پردازنده پیام را مجدداً ارسال می‌کند.

مادامی که پردازنده مراقب این پایانه است پردازش دیگر ترمینال‌ها را هم انجام می‌دهد. چون کاراکترها بصورت سری دریافت می‌شوند، جمع آوری 8 بیت از یک کاراکتر زمان معینی لازم دارد. در حین این جمع‌آوری پردازنده سایر خطوط ارتباطی را مولتی‌پلکس می‌نماید و هر یک را به نوبت سرویس می‌دهد. سرعت دورترین ترمینال‌ها در مقایسه با سرعت پردازنده فوق‌العاده کم است. این خصوصیت اجازه می‌دهد تا چند کاربر، مولتی‌پلکس شده و به کارآیی بهتری در سیستم اشتراک زمانی[1] دستیابی شود. این خصوصیت همچنین اجازه می‌دهد تا چندین کاربر بطور همزمان کار کنند و هر یک با سرعت واکنش معمولی انسان نمونه‌برداری شوند.

## شفافیت داده‌ها[2]

قرارداد مبتنی بر کاراکتر در واقع برای ارتباط با صفحه کلید، چاپگر و وسایل نمایشی که کاراکترهای الفبا عددی را بکار می‌برند تهیه شد. با توسعه زمینه ارتباطات، لازم شد تا اطلاعات دودویی که متن اسکی نبودند ارسال شوند. این نیاز، بعنوان مثال وقتی که دو کامپیوتر دور از هم برنامه‌ها و داده‌هایی را به یکدیگر با یک کانال ارتباطی ارسال کنند، وجود دارد. در قرارداد مبتنی بر کاراکتر، الگوی بیتی در متن پیام مشکل‌ساز می‌شود. این بدان علت است که هر الگوی 8 بیتی متعلق به کاراکتر کنترل انتقال ممکن است اشتباه تفسیر شود. بعنوان مثال، اگر داده دودویی در بخش متن پیام دارای الگوی هشت بیتی 10000011 باشد، گیرنده آنرا بعنوان یک کاراکتر EXT تفسیر کرده و فرض خواهد کرد که انتهای میدان متن فرارسیده است. وقتی که بخش متن پیام از نظر طول متغیر و حاوی بیتهایی باشد که باید بدون ارجاع به هر کد خاصی پذیرفته شوند گوئیم که حاوی داده‌های شفاف[3] است. این ویژگی مستلزم این است که مدار منطقی تشخیص کاراکتر گیرنده غیرفعال شود تا الگوی داده‌های موجود در میدان متن تصادفاً بعنوان اطلاعات کنترل ارتباطات تعبیر نگردد.

شفافیت داده در قرارداد مبتنی بر کاراکتر بوسیله گنجاندن کاراکتر چشم‌پوشی از ارتباط داده[4] DLE قبل از هر کاراکتر کنترل ارتباط حاصل می‌شود. بنابراین، سرآیند با دو کاراکتر DLE SOH شروع خواهد شد، و میدان متن هم با دو کاراکتر DLE ETX خاتمه می‌یابد. اگر الگوی بیتی DLE برابر 00010000 در بخش متن پیام ظاهر شود، فرستنده یک DLE دیگر بدنبال آن می‌گنجاند. گیرنده تمام DLE ها را حذف و الگویی هشت بیتی بعدی را وارسی می‌کند. اگر این الگو یک DLE دیگر باشد، گیرنده آن را

---

1- Time– Sharing 2- Data Transparency 3- Transparent data
4- Data link escape

بخشی از متن تصور کرده و به دریافت متن ادامه می‌دهد. در غیر اینصورت گیرنده الگوی هشت بیتی بعدی را یک کاراکتر کنترل ارتباطی تلقی می‌نماید.

کارایی دستیابی به شفافیت داده بوسیله کاراکتر DLE پایین و پیاده‌سازی آن هم تا حدی پیچیده است. بنابراین قراردادهای دیگری برای کارآتر کردن ارسال داده‌های شفاف تهیه شده است. یکی از این قراردادها که بوسیله کمپانی Digital Equipment بکار رفته است، بکارگیری میدان شمارش بایت است که تعداد بایت‌های پیامی را که بدنبال آن ارسال خواهد شد مشخص می‌کند. باین ترتیب گیرنده باید تعداد بایت‌های دریافتی را تا پایان میدان متن بشمرد. قراردادی که بیش از همه برای حل مشکل شفافیت داده و سایر مشکلات مربوط به قرارداد مبتنی بر کاراکتر بکاررفته است، قرارداد مبتنی بر بیت است.

## قرارداد مبتنی بر بیت[1]

قرارداد مبتنی بر بیت در میدان کنترل خود از کاراکترها استفاده نمی‌کند و مستقل از هر کد بخصوص است. این قرارداد امکان ارسال رشته‌ای از بیت‌های سری را با هر طول، بدون در نظر گرفتن مرزهایی برای کاراکترها فراهم می‌سازد. پیام‌ها با قالب خاصی بنام کادر سازماندهی می‌شوند. یک کادر علاوه بر میدان اطلاعات حاوی میدان‌های آدرس، کنترل وارسی خطاست. مرزهایی کادر از عدد هشت بیتی خاصی که پرچم نامیده می‌شود معین می‌گردد. مثالی‌هایی از قرارداد مبتنی بر بیت کنترل، ارتباط داده همگام[2]، SDLC، است که بوسیله IBM، قرارداد کنترل ارتباط داده سطح بالا[3] HDLC بکاررفته بوسیله سازمان بین‌المللی استاندارد و رویه کنترل پیشرفته مخابره داده، ADCCP، مورد استفاده انستیتو ملی استانداردهای آمریکا می‌باشد.

هر خط انتقال داده حداقل از دو ایستگاه که در انتقال شرکت دارند تشکیل شده است. ایستگاهی که مسئولیت ارتباط داده را بعهده دارد و زمانی را برای کنترل ارتباط صادر می‌کند ایستگاه اصلی[4] خوانده می‌شود. ایستگاه دیگر ایستگاه فرضی[5] یا ثانویه نامیده می‌شود. قراردادهای مبتنی بر بیت وجود یک ایستگاه اصلی و یک یا چند ایستگاه فرعی را فرض می‌کند. کلیه انتقال‌های داده از ایستگاه اصلی به ایستگاه فرعی، یا از یک ایستگاه فرعی به ایستگاه اصلی است.

قالب کادر برای قرارداد مبتنی بر بیت در شکل ۲۶-۱۱ نشان داده شده است. کادر با پرچم هشت بیتی 01111110 و بدنبال آن رشته آدرس و کنترل است. میدان اطلاعات محدودیتی در قالب ندارد و هر طولی می‌تواند داشته باشد. میدان وارسی کادر یک رشته CRC (وارسی چرخشی[6]) مورد استفاده در کشف خطا در ارسال است. پرچم انتهایی به ایستگاه گیرنده اطلاع می‌دهد که اطلاعات رسیده تا آن لحظه بیت‌های CRC است. کادر انتهایی ممکن است بوسیله کادری دیگر، پرچمی دیگر، یا یک

---

1- Bit-Oriented protocol
2- Synchronous Data link Control
3- High-Level Data Link Control
4- Primary station
5- Secondary station
6- Cyclic Redundancy Check

| پرچم 01111110 | وارسی کادر 16 بیتی | اطلاعات هر تعداد بیت | کنترل 8 بیت | آدرس 8 بیت | پرچم 01111110 |
|---|---|---|---|---|---|

**شکل ۲۶-۱۱** قالب کادر برای قرارداد مبتنی بر بیت

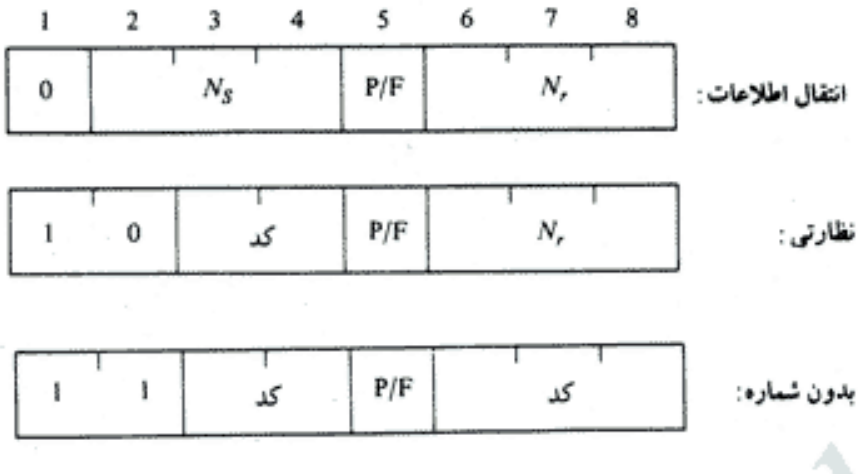
رشته‌های متوالی از 1 ها دنبال شود. وقتی دو کادر بدنبال هم بیایند پرچم فی‌مابین دو کادر، هم پرچم انتهایی کادر اول و هم پرچم شروع کادر بعدی است. اگر اطلاعاتی تبادل نشود، فرستنده یک سری پرچم را برای حفظ خط در وضعیت فعال خود ارسال می‌کند. با وقوع 15 بیت متوالی 1، یا بیشتر، خط فعال تصور می‌شود. کادرهایی که دارای پیام‌های کنترل معین نیستند بدون میدان اطلاعات ارسال می‌گردند. یک کادر باید حداقل 32 بیت بین دو پرچم برای میدان‌های آدرس، کنترل، و وارسی کادر داشته باشد. طول حداکثر به شرایط کانال ارتباطی و قابلیت‌های آن در ارسال پیام‌های طولانی بدون خطا وابسته است.

برای جلوگیری از وقوع یک پرچم در میانه یک کادر، در قرارداد مبتنی بر بیت از روشی بنام درج صفر[1] استفاده می‌شود. در این روش ایستگاه فرستنده پس از هر پنج 1 متوالی، یک صفر را ارسال می‌کند. گیرنده همواره هر صفری را که پس از پنج 1 آمده باشند حذف می‌کند. بنابراین الگوی بیتی 0111111 بصورت 01111101 ارسال گشته و بوسیله گیرنده پس از حذف 0 درج شده، بصورت اولیه‌اش ذخیره می‌گردد. در نتیجه به هیچ وجه الگوی 01111110 بین ابتدا و انتهای خط ارسال نمی‌گردد.

پس از پرچم، آدرس قراردارد، که بوسیله ایستگاه اصلی برای تشخیص آدرس ایستگاه ثانویه بکار می‌رود. وقتی که یک ایستگاه فرعی، یا ثانویه، کادری را ارسال می‌نماید، آدرس به کادر اصلی اطلاع می‌دهد که کدام ایستگاه فرعی کادر را ایجاد کرده است. یک میدان آدرس هشت بیتی می‌تواند تا 256 آدرس را مشخص کند. برخی قراردادهای مبتنی بر بیت امکان استفاده از میدان آدرس وسیعتری را در اختیار می‌گذارند. برای حصول به این امکان، کم‌ارزشترین بیت بایت آدرس فعلی، در صورت وجود بایت بعدی، صفر می‌گردد. تشخیص آخرین بایت آدرس با یافتن یک 1 در کم‌ارزشترین بیت آن آدرس امکان‌پذیر می‌باشد.

پس از میدان آدرس، میدان کنترل قراردارد. میدان کنترل می‌تواند در سه قالب مختلف مطابق شکل ۲۷-۱۱ وجود داشته باشد. قالب انتقال اطلاعات برای ارسال داده‌های معمولی بکار می‌رود. هر کادری که با این قالب ارسال گردد حاوی شمارش‌های ارسال و دریافت است. ایستگاهی که کادرهای متوالی را می‌فرستد هر کادر را شمارش و شماره‌گذاری می‌کند. این شمارش بوسیله ارسال شمار[2] $N_S$ شمرده می‌شود. ایستگاهی که کادرهای متوالی را دریافت می‌نماید، هر یک از کادرهای بدون خطای دریافتی را می‌شمارد. تعداد این کادرهای دریافتی را دریافت شمار[3] $N_R$ می‌شمارد. هر وقت کادری وارسی می‌شود و مشخص شود خطایی رخ نداده است، $N_R$ افزایش می‌یابد. گیرنده، کادرهای اطلاعاتی شماره‌دار

1- Zero Insertion 2- Send Count 3- Recieve Count




**شکل ۲۷-۱۱ قالب میدان کنترل در قرارداد مبتنی بر بیت**

پذیرفته شده را با برگرداندن شمار $N_r$ به ایستگاه فرستنده تأیید می‌کند.

بیت P/F توسط ایستگاه اصلی برای همه‌پرسی از یک ایستگاه برای تقاضای اقدام به ارسال به کار می‌رود. این بیت بوسیله ایستگاه فرعی برای شناسایی آخرین کادر ارسالی مورد استفاده قرار می‌گیرد. بنابراین میدان P/F وقتی که ایستگاه اصلی در حال ارسال باشد، P (بمعنی همه‌پرسی[1]) و وقتی که ایستگاه فرعی در حال ارسال باشد، F (بمعنی نهایی[2]) خوانده می‌شود. هر کادر ارسالی از ایستگاه اصلی به فرعی دارای بیت P برابر با 0 است. وقتی که ایستگاه اصلی بکار خود پایان دهد و آماده پاسخ ایستگاه فرعی گردد، P برابر 1 می‌گردد. سپس ایستگاه فرعی شروع به ارسال تعدادی کادر می‌کند که در آنها بیت F برابر 0 است. وقتی که ایستگاه دوم آخرین کادر را ارسال نمود، بیت F را 1 می‌کند. بنابراین، بیت P/F برای تعیین ختم ارسال داده بکار می‌رود.

قالب نظارتی میدان کنترل با 0 یا 1 بودن دو بیت اول مشخص می‌شود. دو بیت بعدی نوع فرمان را مشخص می‌نماید. بدنبال آن بیت P/F و کادر شمار رشته دریافتی[3] آمده است. کادرهای قالب نظارت حامل میدان اطلاعات نیستند. این کادرها برای کمک در انتقال اطلاعات استفاده می‌شوند، بدین ترتیب که آنها پذیرش کادرهای قبلی حاوی اطلاعات را تأیید می‌کنند، شرایط آمادگی یا مشغولیت را حمل می‌نمایند، و خطاهای شمارش کادر را گزارش می‌نمایند.

قالب بدون شماره با 11 بودن دو بیت اول مشخص می‌شود. پنج بیت موجود در این قالب می‌تواند تا 32 فرمان و پاسخ را معین نمایند. ایستگاه اصلی از میدان کنترل برای مشخص کردن یک فرمان برای

---

1- Poll
2- Final
3- Receive sequence frame Count

ایستگاه فرعی استفاده می‌کند. ایستگاه فرعی میدان را برای ارسال یک پاسخ به ایستگاه اصلی بکار می‌گیرد. کادرهای قالبی که شماره ندارند برای شروع عملیات ارتباط[1]، گزارش خطاهای رویه‌ای[2]، قرار دادن ایستگاه‌ها در حالت قطع، و سایر عملیات کنترلی ارتباط داده بکار می‌روند.

## مسائل

۱-۱۱ آدرس‌های اختصاص یافته به چهار ثبات واسطه I/O در شکل ۲-۱۱ برابر با معادل دودویی اعداد 12، 13، 14 و 15 است. مدار خارجی لازم که باید بین یک آدرس I/O هشت بیتی از CPU و در ورودی‌های CS، RS1 و RSO از مدار واسطه وصل شوند را نشان دهید.

۲-۱۱ شش واحد واسطه از نوعی که در شکل ۲-۱۱ نشان داده شده است به یک CPU که از یک آدرس I/O هشت بیتی استفاده می‌کند متصل‌اند. هر یک از شش ورودی انتخاب تراشه[3] CS به خط آدرس متفاوتی متصل است، بنابراین خط با ارزشتر آدرس به ورودی CS اولین واحد واسطه و ششمین خط آدرس به ششمین واحد واسطه متصل است. دو خط آدرس کم‌ارزشتر به RS1 و RSO از هر شش واحد واسطه متصل‌اند. آدرس هشت بیتی هر یک از ثبات‌های هر واسطه را معین کنید.

۳-۱۱ چهار وسیله جانبی که خروجی قابل درکی برای انسان داشته باشد نام ببرید.

۴-۱۱ نام کامل خود را به ASCII با استفاده از هشت بیت برای هر کاراکتر و یک 0 در سمت چپ‌ترین مکان بنویسید. بین قسمت‌های مختلف نام یک فاصله بگذارید و نیز در صورت استفاده از مخفف، پس از آن نقطه بگذارید.

۵-۱۱ اختلاف بین I/O مجزا با I/O نگاشت چیست؟ مزایا و معایب هر یک را بگویید.

۶-۱۱ مشخص کنید که کدام یک از موارد زیر یک فرمان کنترل، وضعیت یا انتقال داده می‌باشد.

الف) گذر از دستورالعمل بعدی اگر پرچم 1 باشد.

ب) جستجوی یک رکورد خاص بر روی یک دیسک مغناطیسی

ج) وارسی آماده بودن یا نبودن وسیله I/O

د) حرکت کاغذ چاپگر به ابتدای صفحه بعد

هـ) خواندن ثبات وضعیت واسطه

۷-۱۱ یک واحد واسطه تجاری برای خطوط دست‌دهی مربوط به انتقال داده‌ها از وسیله I/O به واحد واسطه از نام‌های مختلفی استفاده می‌کند. خط دست‌دهی ورودی واسطه STB (مخفف

---

1- Link Function
2- Reporting Procedural Errors
3- Chip Select

استروب یا فعال‌ساز[1]) و خط دست‌دهی خروجی واسطه IBF (مخفف پربودن بافر ورودی[2]) نام‌گذاری شده است. وجود یک سیگنال سطح بالا روی IBF نشان می‌دهد که داده توسط واسطه پذیرفته شده است. پس از سیگنال خواندن I/O از طرف CPU از طریق ثبات داده، IBF به سطح پایین می‌رود.

الف) یک بلاک دیاگرام که CPU، واسطه و وسیله I/O همراه با اتصال‌های لازم بین آنها را نشان دهد رسم کنید.

ب) یک دیاگرام زمانی برای انتقال دست‌دهی رسم کنید.

ج) برای انتقال از وسیله به واسطه و از واسطه به CPU یک چارت دنباله رخدادها[3] را رسم کنید.

۱۱-۸ یک CPU با ساعت 20MHz به یک واحد حافظه که زمان دسترسی آن 40ns است متصل است. دیاگرام‌های زمانی خواندن و نوشتن آنها را با استفاده از فعال‌کننده‌های READ و WRITE تهیه کنید. در دیاگرام زمانی آدرس را هم نشان دهید.

۱۱-۹ مدار واسطه ارتباطی غیرهمگام شکل ۱۱-۸ بین یک CPU و یک چاپگر متصل شده است. فلوچارتی رسم کنید که دنباله عملیات ارسال کاراکتر از CPU به چاپگر را نشان دهد.

۱۱-۱۰ حداقل شش وضعیت مختلف را برای 1 کردن بیت‌های ثبات وضعیت در یک واسطه ارتباطی غیرهمگام ارائه دهید.

۱۱-۱۱ اگر واسطه به پایانه‌ای متصل باشد که به یک بیت توقف نیاز دارد، چند بیت در ثبات شیفت فرستنده شکل ۱۱-۸ وجود دارد؟ با استفاده از کد اسکی توازن زوج بیت‌های ثبات شیفت را وقتی که حرف W ارسال شود مشخص کنید.

۱۱-۱۲ بر روی یک خط 1200-Baud با هر یک از روش‌های زیر چند کاراکتر در هر ثانیه ارسال می‌شود (کد کاراکترها را هشت بیتی فرض کنید).

الف) ارسال سری همگام (همزمان)

ب) ارسال سری غیرهمگام (با دو بیت توقف)

ج) ارسال سری غیرهمگام با یک بیت توقف

۱۱-۱۳ اطلاعاتی با سرعت m بایت در ثانیه به یک بافر FIFO وارد می‌شود. اطلاعات پاک شده در آن نیز n بایت در ثانیه است. حداکثر ظرفیت بافر ۱۲ بایت است.

الف) اگر m>n باشد چه مدت طول می‌کشد تا بافر خالی، پر شود.

ب) اگر m<n باشد چقدر طول می‌کشد تا بافر پر، خالی شود.

ج) اگر m=n باشد آیا بافر FIFO لازم است یا خیر؟

۱۱-۱۴ بیت‌های ثبات کنترل FIFO در شکل ۱۱-۹ عبارتند از $F_1F_2F_3F_4$ = 0011. رشته عملیات داخلی را هر گاه یک قلم از داده از FIFO حذف و سپس داده جدیدی وارد شود معین کنید.

---

1- Strobe
2- Input Buffer Full
3- Sequence-of events Chart

۱۱-۱۵ مقادیر "ورودی آماده" و "خروجی آماده" و بیت‌های کنترل $F_1$ تا $F_4$ در شکل ۹-۱۱ در وضعیت‌های زیر چیست؟
الف) وقتی بافر خالی است.
ب) وقتی بافر پر است.
ج) وقتی بافر حاوی دو قلم داده است.

۱۱-۱۶ یک بلاک دیاگرام مشابه شکل ۱۰-۱۱ را برای انتقال داده از CPU به یک واسطه و سپس به یک وسیله I/O رسم کنید. رویه‌ای برای 1کردن و 0کردن بیت پرچم ارائه دهید.

۱۱-۱۷ با استفاده از پیکربندی شکل مسئله ۱۶-۱۱ فلوچارتی (مشابه با شکل ۱۱-۱۱) برای برنامه CPU جهت خروجی داده بدست آورید.

۱۱-۱۸ مزیت اصلی استفاده از انتقال داده به کمک وقفه در برابر انتقال تحت کنترل برنامه بدون استفاده از وقفه چیست؟

۱۱-۱۹ در بیشتر کامپیوترها وقفه فقط پس از اجرای دستورالعمل جاری تشخیص داده می‌شود. امکان پذیرش وقفه را در هر زمانی در حین اجرای دستورالعمل بررسی کنید. در مورد مشکلاتی که ممکن است بروز کند بحث کنید.

۱۱-۲۰ در سیستم وقفه اولویت‌دار چرخشی شکل ۱۲-۱۱، هرگاه پس از درخواست وقفه وسیله 2 به CPU ولی قبل از تصدیق وقفه به وسیله CPU ، وسیله 1 درخواست وقفه کند چه اتفاقی می‌افتد.

۱۱-۲۱ کامپیوتری را بدون سخت‌افزار وقفه اولویت‌دار در نظر بگیرید. هر یک از چند منبع می‌توانند به کامپیوتر وقفه بدهند، در هر درخواست وقفه منجر به ذخیره آدرس برگشت و انشعاب به یک روال مشترک وقفه می‌گردد. توضیح دهید که چگونه می‌توان در برنامه سرویس‌دهی وقفه، اولویت برقرار کرد.

۱۱-۲۲ با استفاده از روش‌های طراحی مدارهای ترکیبی، عبارات بولی ارائه شده در جدول ۲-۱۱ را برای کد گذار اولویت بدست آورید. نمودار منطقی مدار را رسم کنید.

۱۱-۲۳ سخت‌افزار وقفه اولویت‌دار موازی را برای سیستمی با هشت منبع وقفه طراحی کنید.

۱۱-۲۴ جدول درستی انکدر 8 × 3 اولویت را بدست آورید. فرض کنید که سه خروجی xyz از انکدر اولویت برای تهیه یک آدرس‌بردار به شکل 101xyz00 بکار روند. هشت آدرس بردار را با شروع از آن که بالاترین اولویت را دارد بنویسید.

۱۱-۲۵ در شکل ۱۴-۱۱ چه کاری باید انجام داد تا چهار مقدار VAD برابر با 76، 77، 78، 79 گردند؟

۱۱-۲۶ برای وارسی زمانی که منبعی به کامپیوتر در حال سرویس‌دهی به وقفه قبلی بوسیله همان منبع، وقفه می‌دهد، چه برنامه‌ای لازم است.

۱۱-۲۷ چرا خطوط کنترل خواندن و نوشتن در کنترل کننده DMA دوطرفه است؟ تحت چه شرایطی و

به چه منظوری از آنها بعنوان ورودی استفاده می‌شود؟ تحت چه شرایطی و برای چه منظوری از آنها بعنوان خروجی استفاده می‌شود.

۱۱-۲۸ می‌خواهیم 256 کلمه را از یک دیسک مغناطیسی به بخشی از حافظه که از آدرس 1230 شروع می‌شود انتقال دهیم. انتقال با استفاده از DMA طبق شکل ۱۸-۱۱ صورت می‌گیرد.

۱۱-۲۹ یک کنترل کننده DMA کلمه‌های 16 بیتی را با استفاده از سرقت سیکل به حافظه منتقل می‌کند. این کلمه‌ها از وسیله‌ای دریافت می‌شود که کاراکترها را با سرعت 2400 کاراکتر در ثانیه ارسال می‌نماید. CPU دستورالعمل‌ها را با سرعت یک میلیون دستورالعمل در ثانیه برداشت و اجرا می‌کند. CPU در اثر انتقال به شیوه DMA تا چه حد کند می‌شود؟

۱۱-۳۰ چرا DMA بهنگام تقاضای انتقال به حافظه نسبت به CPU بهنگام تقاضای انتقال به حافظه اولویت دارد.

۱۱-۳۱ فلوچارتی مشابه ۱۱-۲۰ برای IBM 370 رسم کنید و ارتباط CPU با کانال انتقال I/O را توضیح دهید.

۱۱-۳۲ آدرس یک پایانه متصل به یک پردازنده انتقال داده از دو حرف الفبای انگلیسی یا یک حرف و بدنبال آن یکی از ده رقم تشکیل شده است. چند آدرس مختلف می‌توان ایجاد کرد.

۱۱-۳۳ رویه‌ای را برای خط انتقال و دنباله کاراکترها، به منظور ارتباط بین یک پردازنده انتقال داده و یک پایانه دوردست تنظیم کنید. پردازنده سئوال می‌کند که پایانه در حال کار است یا خیر. پایانه با بله و خیر به آن پاسخ می‌دهد. اگر پاسخ بلی باشد پردازنده بلاکی از متن برای آن ارسال می‌نماید.

۱۱-۳۴ یک ارتباط انتقال داده از قرارداد مبتنی بر کاراکتر با شفافیت داده‌ها با بکارگیری کاراکتر DLE استفاده می‌کند. پیام متنی که فرستنده بین STX و ETX می‌فرستد بصورت زیر است:

DLE STX DLE DLE ETX DLE DLE ETX DLE ETX

داده‌های شفاف متنی را بصورت دودویی بنویسید.

۱۱-۳۵ حداقل تعداد بیت‌هایی که یک کادر در قرارداد مبتنی بر بیت دارد چقدر است؟

۱۱-۳۶ نشان دهید که چگونه روش درج صفر در قرارداد مبتنی بر بیت، وقتی که یک صفر و بدنبال آن ده بیت معادل دودویی 1023 ارسال می‌شود عمل می‌کند.

# ۱۲ سازمان حافظه

۱۲-۱ سلسله مراتب حافظه
۱۲-۲ حافظه اصلی
۱۲-۳ حافظه کمکی
۱۲-۴ حافظه تداعیگر
۱۲-۵ حافظه کش
۱۲-۶ حافظه مجازی
۱۲-۷ سخت افزار مدیریت حافظه

## ۱۲-۱ سلسله مراتب حافظه

واحد حافظه در هر کامپیوتر دیجیتال یکی از بخش‌های اصلی است زیرا برای ذخیره برنامه‌ها و داده‌ها مورد نیاز می‌باشد. یک کامپیوتر بسیار کوچک با کاربرد محدود ممکن است بتواند کار معینی را بدون نیاز به افزایش ظرفیت ذخیره‌سازی انجام دهد. اما کامپیوترهای همه منظوره اگر علاوه بر ظرفیت حافظه اصلی، به امکانات ذخیره سازی اضافی مجهز باشند کارها را موثرتر اجرا خواهند نمود. در یک کامپیوتر معمولاً فضای واحد حافظه برای جای دادن همه برنامه‌های مورد استفاده کافی نیست. بعلاوه، اکثر کاربران کامپیوتر مقادیر قابل‌توجهی از نرم‌افزارهای پردازش داده را جمع‌آوری نموده و به این عمل همچنان ادامه می‌دهند. البته کلیه اطلاعات جمع‌آوری شده بطور همزمان مورد نیاز پردازنده نیست. بنابراین، مقرون به صرفه‌تر است از وسایل ذخیره‌سازی ارزان قیمت بعنوان پشتیبانی، اطلاعاتی که در حال حاضر مورد نیاز CPU نیستند، استفاده شود. واحد حافظه‌ای که مستقیماً با CPU ارتباط دارد حافظه اصلی[1] نامیده می‌شود. وسایلی که نقش پشتیبانی ذخیره‌سازی را بعهده دارند حافظه کمکی[2] خوانده می‌شوند. معمول‌ترین وسیله حافظه کمکی بکار رفته در سیستم‌های کامپیوتری دیسک‌ها و نوارهای مغناطیسی هستند. این وسایل برای ذخیره برنامه‌های سیستم، فایل‌های بزرگ داده، و سایر اطلاعات پشتیبانی مورد استفاده‌اند. در حافظه اصلی فقط برنامه‌ها و داده‌های مورد استفاده جاری پردازنده جای می‌گیرند. کلیه اطلاعات دیگر در حافظه کمکی ذخیره می‌شود و هنگام لزوم به حافظه اصلی منتقل می‌گردند.

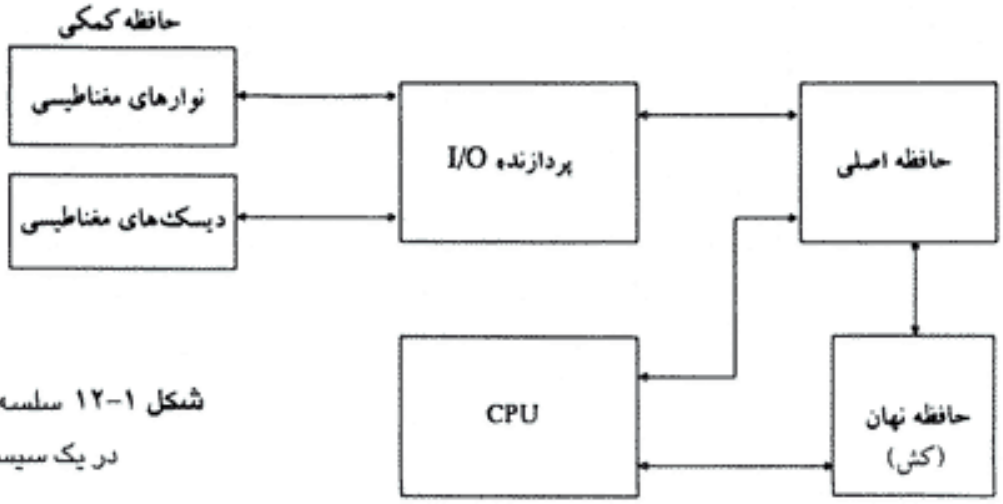
---

1- Main Memory 2- Auxiliary Memory

ظرفیت کل یک کامپیوتر را می‌توان بصورت یک سیستم سلسله مراتبی از اجزاء تصور کرد. این سیستم حافظه سلسله مراتبی متشکل است از کلیه وسایل ذخیره‌سازی در یک سیستم کامپیوتری، از حافظه کمکی کُند ولی با ظرفیت بالا گرفته تا حافظه اصلی نسبتاً سریع‌تر، و حتی حافظه کوچکتر ولی سریع‌تر کَش (نهان) که بوسیله مدارهای منطقی پردازشگر سریع قابل دسترسی هستند. شکل ۱-۱۲ اجزاء سلسله مراتبی حافظه‌ای نوعی را نشان می‌دهد. در پایین سیستم سلسله مراتبی نوارهای مغناطیسی نسبتاً کُند قرار دارد که برای ذخیره فایل‌های جابجایی بکار می‌رود. پس از آن دیسک‌های مغناطیسی واقعند که بعنوان ذخیره‌سازی پشتیبانی بکار می‌روند. حافظه اصلی مکان میانی را دارد. و قادر است مستقیماً با CPU و وسایل کمکی از طریق پردازنده I/O ارتباط داشته باشد. وقتی برنامه‌هایی که در حافظه اصلی قرار ندارند بوسیله CPU مورد نیاز باشند، از حافظه کمکی بداخل آورده می‌شوند. برنامه‌هایی که در حال حاضر در حافظه اصلی قرار دارند ولی مورد نیاز نیستند به حافظه کمکی منتقل می‌گردند تا فضای کافی برای برنامه‌ها و داده‌های جاری بکار رفته ایجاد شود.

نوعی حافظه خاص بسیار سریع که کَش نامیده می‌شود گاهی اوقات برای افزایش سرعت پردازش با در اختیارگذاشتن برنامه‌ها و داده‌های جاری برای CPU با سرعتی بالا، بکار می‌رود. حافظه کَش در کامپیوترها برای جبران اختلاف سرعت بین دستیابی حافظه اصلی و مدار پردازنده مورد استفاده قرار می‌گیرد. مدار CPU معمولاً سریعتر از زمان دستیابی (سرعت) حافظه اصلی است، لذا سرعت پردازش را عمدتاً سرعت حافظه اصلی محدود می‌کند. تکنیکی که برای جبران عدم تطابق سرعت‌های عملیاتی بکار می‌رود استفاده از یک حافظه کش کوچک و فوق‌العاده سریع بین CPU و حافظه اصلی است که زمان دستیابی آن نزدیک به پریود ساعت پردازنده می‌باشد. حافظه کش برای ذخیره قطعه‌هایی از برنامه‌ها که در CPU در دست اجرا هستند و داده‌های موقتی که در محاسبات جاری مکرراً بکار می‌روند مورد استفاده قرار می‌گیرد. با فراهم نمودن سریع برنامه‌ها و داده‌ها، امکان افزایش سرعت عملکرد وجود خواهد داشت.

مادامی که پردازنده I/O انتقال داده را بین حافظه کمکی و حافظه اصلی اداره می‌کند، سازمان کش به




**شکل ۱-۱۲** سلسله مراتب حافظه در یک سیستم کامپیوتری

انتقال اطلاعات بین حافظه اصلی و CPU می‌پردازد. بنابراین هر یک از آنها با سطح متفاوتی از سلسله مراتبی حافظه درگیر است. دلیل داشتن دو یا سه سطح در سلسله مراتب حافظه، مسائل اقتصادی است. با افزایش ظرفیت حافظه، ارزش ذخیره اطلاعات در هر بیت کاهش می‌یابد و زمان دستیابی حافظه طولانی‌تر می‌گردد. حافظه کمکی ظرفیت ذخیره‌سازی بالایی را داراست، نسبتاً ارزان است، ولی در مقایسه با حافظه اصلی سرعت دستیابی پایینی را دارد. حافظه کش خیلی کوچک است، نسبتاً گران است، و سرعت دستیابی بالایی دارد. بنابراین با افزایش دستیابی حافظه، ارزش نسبی آن اضافه می‌گردد. هدف نهایی از بکارگیری سلسله مراتب حافظه‌ای، فراهم آوردن بالاترین سرعت دستیابی متوسط ممکن با حداقل ارزش کل تمام حافظه سیستم است.

حافظه‌های کمکی و کش برای مقاصد مختلفی بکار می‌روند. کش آن بخش از برنامه و داده‌هایی را که بیشتر مورد استفاده‌اند ذخیره می‌کند، در حالی که حافظه کمکی بخش‌هایی راکه در حال حاضر مورد استفاده CPU نیست نگهمیدارد. بعلاوه CPU بر روی حافظه کش و حافظه اصلی دسترسی مستقیم دارد در صورتی که بر روی حافظه کمکی این چنین نیست. انتقال از حافظه کمکی به حافظه اصلی معمولاً بصورت دسترسی مستقیم به بلاک‌های داده است. نسبت زمان دستیابی بین کش و حافظه اصلی 1 به 7 است. مثلاً یک حافظه کش نوعی ممکن است زمان دستیابی 100ns را داشته باشد، در حالیکه زمان دستیابی حافظه اصلی 700ns است. زمان دستیابی متوسط حافظه کمکی معمولاً 1000 برابر حافظه اصلی است. سایز بلاک در حافظه کمکی بین 256 تا 2048 کلمه است، در حالی که سایز بلاک در حافظه کش از 1 تا 16 کلمه می‌باشد.

بسیاری از سیستم‌های عامل بصورتی طراحی می‌شوند تا CPU بتواند بطور همزمان تعدادی از برنامه‌های مستقل را اجرا نماید. این مفهوم که چندبرنامگی [1] خوانده می‌شود، به وجود دو یا چند برنامه در بخش‌های متفاوت سلسله مراتب حافظه در یک زمان اشاره دارد. بدین ترتیب می‌توان تمام بخش‌های کامپیوتر را باکار برروی چندین برنامه بطور سری، مشغول نگهداشت. مثلاً فرض کنید برنامه‌ای در CPU در دست اجراست و یک انتقال I/O لازم شود. CPU ، پردازنده I/O را برای شروع اجرای انتقال مقداردهی اولیه می‌نماید. این عمل به CPU اجازه می‌دهد تا برنامه دیگری را اجرا کند. در یک سیستم چندبرنامگی، وقتی یک برنامه برای انتقال ورودی - خروجی منتظر می‌ماند، برنامه دیگری آماده استفاده از CPU است.

در چندبرنامگی، نیاز به اجرای برنامه‌ها بصورت قطعه به قطعه، تغییر اندازه حافظه اصلی در حال استفاده توسط یک برنامه مفروض، و جابجایی برنامه‌ها به بخش‌های مختلف سلسله مراتب حافظه وجود دارد. برنامه‌های کامپیوتر گاهی طولانی‌تر از آن هستند که در کل فضای موجود در حافظه اصلی جابگیرند. بعلاوه یک سیستم کامپیوتر چندین برنامه را بکار می‌برد و همه برنامه‌ها نمی‌توانند در یک حافظه اصلی در تمام مدت قرار بگیرند. معمولاً یک برنامه با داده‌های مربوطه‌اش در حافظه کمکی قرار می‌گیرند. وقتی که برنامه یا بخشی از برنامه قرار است اجرا شود، ابتدا به حافظه اصلی انتقال می‌یابد تا بوسیله CPU اجرا شود. بنابراین هر کس می‌تواند تصور کند که حافظه کمکی حاوی همه اطلاعاتی است که در یک

---

1- Multiprogramming

سیستم ذخیره شده است. کار سیستم عامل این است که بخشی از این اطلاعات راکه در حال حاضر فعال است در حافظه اصلی نگهمیدارد. بخشی از سیستم کامپیوتر که بر جریان اطلاعات بین حافظه کمکی و حافظه اصلی نظارت می‌کند سیستم مدیریت حافظه[1] نامیده می‌شود. سخت‌افزار یک سیستم مدیریت حافظه در بخش ۷-۱۲ ارائه شده است.

## ۲-۱۲ حافظه اصلی

حافظه اصلی واحد ذخیره‌سازی مرکزی در یک سیستم کامپیوتری است. این واحد، حافظهٔ نسبتاً بزرگ و سریعی است که برای ذخیره برنامه‌ها و داده‌ها در حین عملکرد کامپیوتر مورد استفاده قرار می‌گیرد. تکنولوژی اصلی بکاررفته برای حافظه اصلی بر مدارهای مجتمع نیمه هادی مبتنی است. تراشه‌های RAM مدار مجتمع با دو شیوه عملکرد مختلف استاتیک[2] و دینامیک[3] در دسترس قرار دارند. RAM استاتیک اساساً از فلیپ‌فلاپها ساخته شده که اطلاعات دودویی را ذخیره می‌کنند. اطلاعات ذخیره شده مادامی که توان الکتریکی به واحد اعمال شود معتبر است. RAM دینامیکی اطلاعات دودویی را بصورت بارهای الکتریکی که به خازن‌ها اعمال می‌شوند، ذخیره می‌نماید. خازن‌ها در داخل تراشه با ترانزیستورهای MOS ایجاد می‌شوند. بارهای ذخیره شده در خازن‌ها به مرور تخلیه می‌شوند و لذا حافظه دینامیکی باید بصورت دوره‌ای بارگذاری شوند. تجدید بارگذاری، یا تازه‌سازی، با چرخش در بین کلمات در هر چند میلی‌ثانیه یکبار و احیای بار در حال تخلیه انجام می‌شود. در RAM دینامیکی مصرف توان الکتریکی کمتر و ظرفیت ذخیره‌سازی در یک حافظه بیشتر است. RAM استاتیکی از نظر کاربرد ساده‌تر بوده و سیکل خواندن و نوشتن در آن کوتاهتر است.

بیشتر حافظه اصلی در یک کامپیوتر همه منظوره از تراشه‌های مدار مجتمع RAM تشکیل می‌شود، اما بخشی از این حافظه ممکن است از تراشه ROM ساخته شود. در ابتدا RAM برای ارجاع به حافظه با دستیابی تصادفی[4] بکار رفت، اما اکنون از آن برای مشخص کردن حافظه خواندنی و نوشتنی[5] در مقابل حافظه فقط خواندنی[6] استفاده می‌شود، هرچند که ROM هم نوعی حافظه با دستیابی تصادفی است. RAM برای ذخیره کردن برنامه‌ها و داده‌هایی که قرار است تغییر یابند بکار می‌رود ولی ROM برای ذخیره برنامه‌هایی راکه بطور دائم در کامپیوتر جای داده می‌شوند و یا محتوای جداول ثابتی که پس از تکمیل ساخت کامپیوتر تغییر نمی‌کنند مورد استفاده قرار می‌گیرند.

در میان سایر امکانات در کامپیوتر، بخش ROM حافظه اصلی برای ذخیره برنامه آغازینی بنام بارکننده راه‌انداز[7] بکار می‌رود. بارکننده راه‌انداز برنامه‌ای است که کار آن آغاز کار نرم‌افزار کامپیوتر پس از روشن کردن دستگاه است. چون RAM فرار است، وقتی که برق دستگاه قطع می‌شود محتوای آن از بین

---

1- Memory Management System
2- Static
3- Dynamic
4- Random Access Memory
5- Read/Write Memory
6- Read Only Memory
7- Bootstrap Loader

می‌رود. محتوای ROM پس از قطع و وصل توان الکتریکی تغییری نمی‌کند. آغاز بکار کامپیوتر متشکل است از روشن شدن دستگاه و شروع اجرای یک برنامه اولیه. بنابراین وقتی که دستگاه روشن شد، سخت‌افزار کامپیوتر شمارنده برنامه را با اولین آدرس بارکننده راه‌انداز بار می‌کند. برنامه راه‌انداز بخشی از سیستم عامل را از دیسک به حافظه اصلی بار می‌کند و سپس کنترل به سیستم عامل انتقال می‌یابد، که کامپیوتر را برای استفاده عمومی آماده می‌نماید.

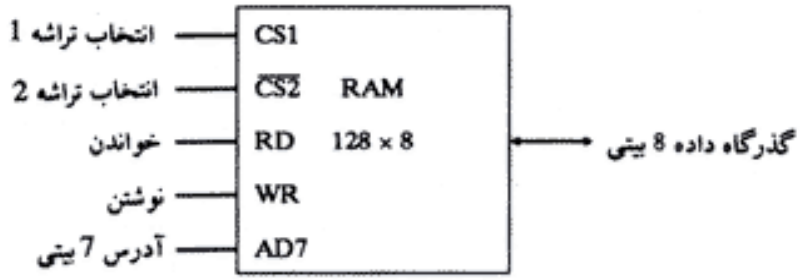
تراشه‌های RAM و ROM در اندازه‌های مختلف موجودند. اگر حافظه مورد نیاز برای کامپیوتر بزرگتر از ظرفیت یک تراشه باشد، لازم است تعدادی تراشه با هم ترکیب شوند تا اندازه حافظه لازم حاصل شود. برای نشان دادن اتصالات بین تراشه‌ها، ساخت یک تراشه 8 × 1024 را با تراشه‌های RAM با اندازه 8 × 128 و تراشه‌های ROM با اندازه 8 × 512 نشان می‌دهیم.

## تراشه‌های RAM و ROM

یک تراشه RAM اگر یک یا چند ورودی کنترل داشته باشد تا فقط در صورت نیاز این تراشه را انتخاب کنند به نحو مناسبتری قادر به ارتباط با CPU است. ویژه‌گی متداول دیگر، گذرگاه داده دوطرفه‌ای است که امکان انتقال داده از حافظه به CPU را در عمل خواندن، یا از CPU به حافظه را در حین عمل نوشتن فراهم می‌سازد. گذرگاه دو طرفه را می‌توان با بافرهای سه حالته ساخت. خروجی یک بافر سه حالته می‌تواند در یکی از سه حالت مختلف قرار گیرد: سیگنال معادل با 1 منطقی، سیگنال معادل با 0 منطقی، یا یک حالت امپدانس بالا. 1 و 0 منطقی سیگنال‌های معمولی منطقی هستند. حالت امپدانس بالا مانند یک مدار باز عمل می‌کند، و بدان معنی است که خروجی سیگنالی حمل نمی‌کند و معنای منطقی ندارد.

بلاک دیاگرام یک تراشه RAM در شکل ۲-۱۲ نشان داده شده است. ظرفیت حافظه 128 کلمه هشت بیتی است (یک بایت). این حافظه دارای 7 خط آدرس و یک گذرگاه داده هشت بیتی دو طرفه است. ورودی‌های خواندن و نوشتن عمل مورد نظر را مشخص می‌کنند و دو ورودی کنترل انتخاب تراشه (CS) برای فعال کردن تراشه بهنگام انتخاب توسط ریزپردازنده، بکار می‌روند. وجود بیش از یک ورودی کنترل برای انتخاب تراشه، عمل دیکد شدن آدرس را هنگامی که چندین تراشه در ریزپردازنده وجود داشته باشد تسهیل می‌نماید. ورودی‌های خواندن و نوشتن گاهی بصورت یک خط R/W در می‌آیند. هر وقت تراشه انتخاب شود، دو حالت دودویی این خط دو عمل خواندن یا نوشتن را مشخص می‌کند.

جدول کار لیست شده در شکل ۲-۱۲ (ب) طرز کار تراشه RAM را مشخص می‌نماید. واحد حافظه فقط هنگامی که CS1=1 و $\overline{CS2}$=0 باشد فعال می‌شود. خط بالای متغیر انتخاب دوم نشان می‌دهد که این ورودی وقتی که برابر 0 باشد فعال است. اگر ورودی‌های انتخاب تراشه فعال نباشند، و یا اگر فعال باشند ولی ورودی‌های خواندن یا نوشتن فعال نباشند، حافظه مسدود و گذرگاه داده آن در حالت




(الف) بلاک دیاگرام

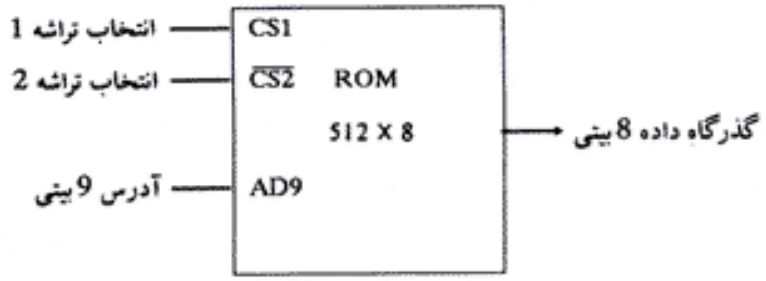
| CS1 | CS2 | RD | WR | عمل حافظه | وضعیت گذرگاه داده |
|---|---|---|---|---|---|
| 0 | 0 | × | × | مسدود | امپدانس بالا |
| 0 | 1 | × | × | مسدود | امپدانس بالا |
| 1 | 0 | 0 | 0 | مسدود | امپدانس بالا |
| 1 | 0 | 0 | 1 | نوشتن | ورودی داده به RAM |
| 1 | 0 | 1 | × | خواندن | خروجی داده به RAM |
| 1 | 1 | × | × | مسدود | امپدانس بالا |

(ب) جدول تابع

**شکل ۲-۱۲ تراشه‌ای از نوع RAM**

امپدانس بالا خواهد بود. اگر CS1=1 و $\overline{CS2}$=0 باشد، حافظه می‌تواند در شیوه عملکرد خواندن یا نوشتن قرار گیرد. اگر ورودی WR فعال شود، حافظه یک بایت را از گذرگاه داده در مکانی که خطوط آدرس تعیین می‌کند ذخیره می‌نماید. اگر ورودی RD فعال شود، محتوای بایت انتخاب شده روی گذرگاه داده قرار می‌گیرد. سیگنالهای RD و WR عملکرد حافظه و بافرهای گذرگاه مربوط به گذرگاه داده دو جهته را کنترل می‌نماید.

یک تراشه ROM هم سازمان خارجی مشابهی دارد. با این وجود، چون ROM تنها خوانده می‌شود، گذرگاه داده تنها در حالت خروجی قرار می‌گیرد. بلاک دیاگرام یک تراشه ROM در شکل ۳-۱۲ دیده می‌شود. در مقایسه دو تراشه RAM و ROM با سایز برابر، ROM می‌تواند تعداد بیت‌های بیشتری نسبت به RAM داشته باشد، زیرا سلولهای دودویی ROM فضای کمتری را نسبت به




**شکل ۳-۱۲ تراشه‌ای از نوع ROM**

RAM اشغال می‌کنند. به همین دلیل، دیاگرام ROM دارای 512 بایت و RAM دارای 128 بایت می‌باشد. نه خط آدرس تراشه ROM یکی از 512 بایت ذخیره شده در آن را مشخص می‌کند. دو ورودی انتخاب تراشه باید CS1=1 و $\overline{CS2}$=0 باشند تا واحد حافظه کار کند. در غیراینصورت، گذرگاه داده در حالت امپدانس بالاست. در ROM به کنترل خواندن یا نوشتن نیازی نیست زیرا این مدار فقط می‌تواند خوانده شود. بنابراین وقتی تراشه بوسیله دو ورودی انتخاب فعال شود بایت انتخابی بوسیله خطوط آدرس روی گذرگاه داده ظاهر می‌شود.

## نقشه آدرس‌های حافظه

طراح یک سیستم کامپیوتری باید مقدار حافظه لازم را برای کاربرد خاص محاسبه و به آن RAM یا ROM لازم را اختصاص دهد. سپس ارتباطات بین حافظه و پردازنده با دانستن اندازه حافظه و نوع تراشهای RAM یا ROM موجود تعیین می‌شود. آدرس دهی حافظه را می‌توان با استفاده از جدولی تنظیم کرد که آدرس حافظه اختصاص یافته به هر تراشه را مشخص می‌کند. این جدول که نقشه آدرس حافظه نامیده می‌شود، نمایش تصویری از فضای آدرس برای هر تراشه در سیستم است.

برای تشریح مطلب فوق با یک مثال فرض کنید که یک سیستم کامپیوتری به یک RAM با ظرفیت 512 بایت و یک ROM با ظرفیت 512 بایت نیاز دارد. تراشه های RAM و ROM بکار رفته در شکل ۲-۱۲ و۳-۱۲ نشان داده شده‌اند. نقشه آدرس‌های حافظه برای این پیکربندی در جدول ۱-۱۲ آمده است. ستون قطعات نشان می‌دهد که آیا ROM بکار رفته است یا RAM . ستون آدرس شانزده شانزدهی به هر تراشه محدوده‌ای از آدرس‌های معادل شانزده شانزدهی را مشخص می‌کند. خطوط گذرگاه آدرس در ستون سوم ارائه شده است. هر چند گذرگاه آدرس 16 خط دارد، جدول فقط 10 خط را نشان می‌دهد زیرا 6 خط دیگر در این مثال بکار نرفته است و فرض می‌شود صفر باشند. x های کوچک در زیر ستون گذرگاه آدرس خطوطی را مشخص می‌کنند که باید به ورودی‌های آدرس هر تراشه وصل شوند. تراشه‌های RAM دارای 128 بایت بوده وبه هفت خط آدرس نیاز دارند. تراشه ROM دارای 512 بایت بوده و به 9 خط آدرس نیاز دارد. x ها همیشه به خطوط پائین رتبه گذرگاه اختصاص می‌یابند: خطوط 1

**جدول ۱-۱۲** نقشه آدرس‌های حافظه برای ریزکامپیوتر

| قطعه | آدرس شانزده شانزدهی | گذرگاه آدرس 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RAM 1 | 0000-007F | 0 | 0 | 0 | x | x | x | x | x | x | x |
| RAM 2 | 0080-00FF | 0 | 0 | 1 | x | x | x | x | x | x | x |
| RAM 3 | 0100-017F | 0 | 1 | 0 | x | x | x | x | x | x | x |
| RAM 4 | 0180-01FF | 0 | 1 | 1 | x | x | x | x | x | x | x |
| ROM | 0200-03FF | 1 | x | x | x | x | x | x | x | x | x |

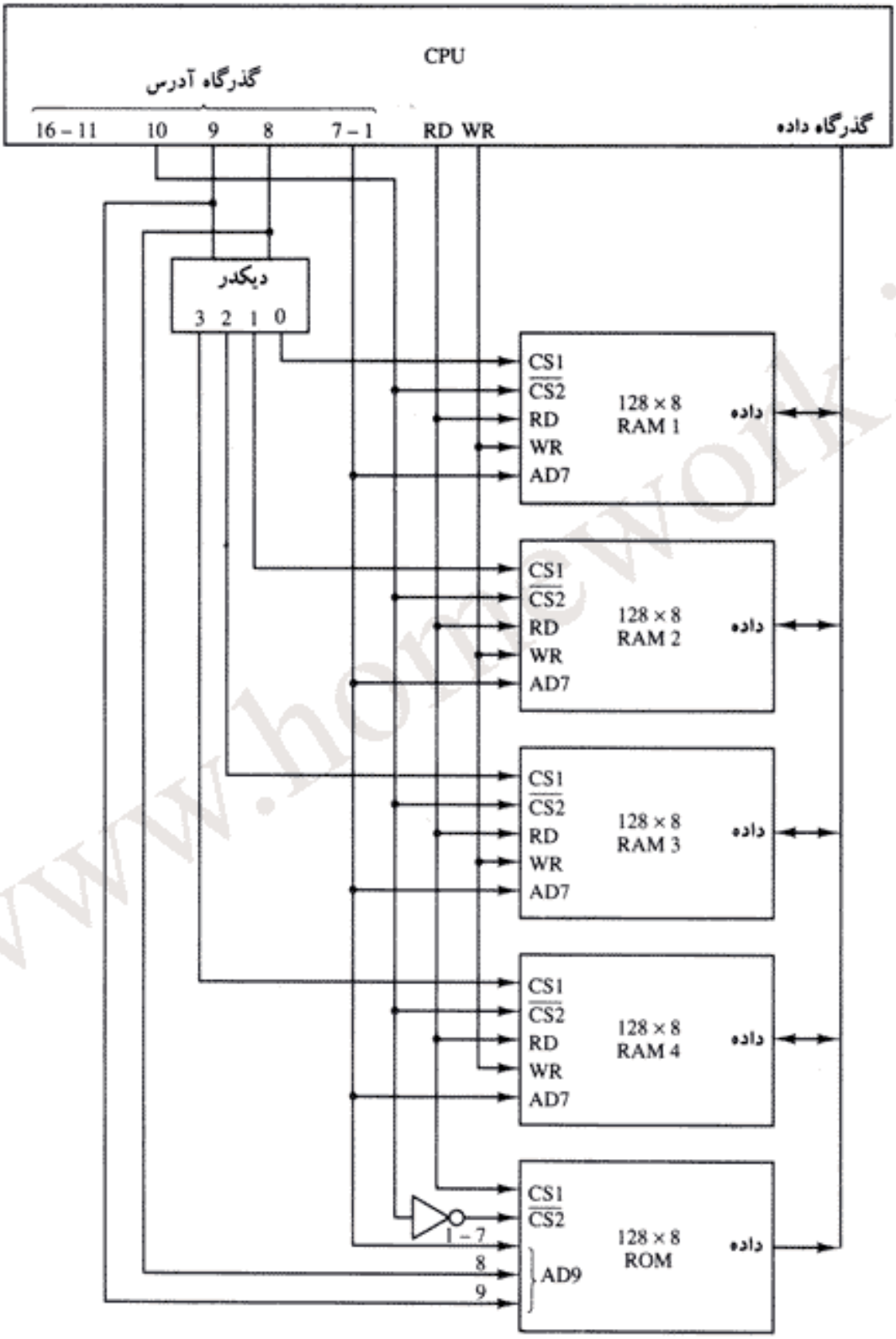
تا 7 به RAM و خطوط 1 الی 9 به ROM. حال لازم است تا چهار تراشه RAM را با تخصیص آدرس متفاوتی به هریک از هم متمایز کنیم. برای این مثال خاص، ما خطوط گذرگاه 8 و 9 را برای نمایش چهار ترکیب دودویی متمایز انتخاب می‌کنیم. دقت کنید هر جفت دیگری از خطوط گذرگاه که بکار نرفته باشند را می‌توان برای این منظور انتخاب کرد. جدول بخوبی نشان می‌دهد که 9 خط پائین رتبه آدرس فضای حافظه‌ای برابر با $2_q$=512 بایت را ایجاد می‌کند. تمایز بین آدرس‌های RAM و ROM به یک خط دیگر گذرگاه انجام می‌شود. در اینجا فقط خط 10 را برای این منظور انتخاب می‌کنیم. هنگامی که خط 10 صفر باشد، CPU یک تراشه RAM را انتخاب می‌کند. و اگر این خط 1 باشد، ROM انتخاب می‌شود.

آدرس معادل شانزده شانزدهی تراشه از اطلاعات زیر ستون گذرگاه آدرس فراهم می‌شود. خطوط گذرگاه آدرس به گروه‌های چهار بیتی تقسیم شده است بطوری هر گروه با یک رقم شانزده شانزدهی نمایش داده می‌شود. اولین رقم شانزده شانزدهی خطوط 13 تا 16 را نشان می‌دهد و همواره 0است. رقم شانزده شانزدهی بعدی خطوط 9 الی 12 را نشان می‌دهد ولی خطوط 11 و 12 همیشه صفرند. محدوده آدرس‌های شانزده شانزدهی برای هرقطعه از x های مربوط به آن تعیین می‌شود. این x ها یک عدد دودویی را از تماماً 0 تا تماماً 1 نمایش می‌دهند.

## اتصالات حافظه به CPU

تراشه‌های RAM و ROM از طریق گذرگاه‌های داده و آدرس به CPU متصل می‌شوند. خطوط پائین رتبه درگذرگاه آدرس بایت درون تراشه را انتخاب می‌کنند و سایر خطوط در گذرگاه آدرس تراشه را از طریق ورودی‌های انتخاب برمی‌گزینند. اتصال تراشه‌های حافظه به CPU در شکل ۴-۱۲ نشان داده شده است. این پیکربندی 512 بایت را برای RAM و 512 بایت را هم برای ROM فراهم می‌آورد. پیکربندی مذکور با توجه به جدول ۱-۱۲ پیاده سازی شده است. هر RAM هفت بیت پائین رتبه از گذرگاه آدرس را برای انتخاب یکی از 128 بایت ممکن دریافت می‌کند. ROM بخصوصی که انتخاب می‌شود با توجه به خطوط 8 و 9 گذرگاه است. این عمل از طریق یک دیکدر 2x4 که خروجی‌های آن به ورودی‌های CS1 در هر تراشه RAM متصل است صورت می‌گیرد. بنابراین وقتی که خطوط آدرس 8 و 9 برابر 00 شوند، اولین تراشه RAM انتخاب می‌شود. وقتی که 01 باشد، دومین تراشه انتخاب می‌گردد، و الی آخر. خروجی‌های RD و WR از ریزپردازنده به ورودی‌های همه تراشه‌های RAM وصل می‌شوند.

انتخاب بین RAM و ROM از طریق خط شماره 10 گذرگاه حاصل می‌گردد. RAM ها هنگامی انتخاب می‌شوند که بیت واقع بر روی این خط 0 باشد، و اگر مقدار بیت 1 شود ROM انتخاب خواهد شد. ورودی انتخاب تراشه دیگر در ROM به کنترل RD متصل است تا بهنگام خواندن موجب فعال شدن ROM گردد. خطوط گذرگاه آدرس 1 تا 9 بدون عبور از دیکدر مستقیماً به ورودی‌های آدرس ROM اعمال شده‌اند. این اتصالات آدرس‌ها 0 تا 511 را به RAM و 512 تا 1023 را به ROM اختصاص می‌دهند. گذرگاه داده ROM تنها امکان خروجی شدن را دارد، در صورتیکه گذرگاه داده متصل




شکل ۴-۱۲ اتصال حافظه به CPU

به RAM می‌تواند اطلاعات را در هر دوجهت هدایت کند.

مثالی که ارائه شد پیچیدگی اتصالات بین تراشه‌های حافظه و CPU را نمایان می‌سازد. هر چه تعداد تراشه‌های متصله بیشتر شود، دیکدرهای بیشتری بین حافظه‌ها مورد نیاز است طراح باید نقشه حافظه‌ای را تهیه کرده و آدرس‌ها را به تراشه تخصیص دهد تا با استفاده از آن اتصالات لازم معین شد.

## ۳-۱۲ حافظه کمکی

متداولترین دستگاه حافظه کمکی بکار رفته در سیستم‌های کامپیوتری دیسک‌ها و نوارهای مغناطیسی هستند. وسایل دیگری هم که نه کراراً ولی بکار می‌روند عبارتند از درام‌های مغناطیسی[1]، حافظه حباب مغناطیسی[2]، و دیسک‌های نوری می‌باشند. برای درک فیزیکی مکانیزم دستگاه‌های حافظه کمکی باید اطلاعاتی در مورد سیستم‌های مغناطیسی، الکترونیکی و الکترومکانیکی داشت. هر چند خواص فیزیکی این وسایل ذخیره‌سازی می‌تواند بسیار پیچیده باشد، ولی خصوصیات منطقی آنها قابل تشخیص و مقایسه است. مشخصه‌های مهم هر وسیله شیوه دستیابی، زمان دستیابی، سرعت انتقال، ظرفیت و قیمت آن است.

زمان متوسط لازم برای رسیدن به یک مکان ذخیره سازی در حافظه و دریافت محتوای آن، زمان دستیابی[3] (یا دسترسی) خوانده می‌شود. در وسایل الکترومکانیکی با قطعات متحرک مانند دیسک‌ها و نوارها، زمان دستیابی متشکل است از زمان جستجوی[4] لازم برای قراردادن هد خواندن - نوشتن بر روی یک مکان و زمان انتقال[5] لازم برای انتقال داده به و یا از وسیله. چون زمان جستجو معمولاً خیلی طولانی‌تر از زمان انتقال است، حافظه کمکی بصورت مجموعه‌ای از رکوردها یا بلاک‌ها سازمان یافته است. یک بلاک تعداد مشخصی از کاراکترها یا کلمات است. معمولاً خواندن یا نوشتن بر روی تمام رکورد انجام می‌شود. سرعت انتقال تعداد کلمات یا کاراکترهایی است که وسیله می‌تواند در هرثانیه، پس از قرارگرفتن در آغاز رکورد، انتقال دهد.

درام‌ها و دیسک‌های مغناطیسی از نظر عملکرد کاملاً مشابهند. هر دو از سطوح چرخان با سرعت بالا که از ماده مغناطیسی ضبط کننده پوشش یافته‌اند تشکیل شده‌اند. سطح چرخان درام، سیلندری شکل و دیسک سطح گرد مسطحی است. سطح ضبط کننده با سرعت یکنواختی چرخیده و در زمان دستیابی هرگز راه‌اندازی یا متوقف نمی‌شود. بیت‌ها بصورت نقطه‌های مغناطیسی با عبور سطح از زیر یک مکانیزم ساکن بنام هد نوشتن[6]، بر روی آن ضبط می‌شوند. بیت‌های ذخیره شده با تغییر میدان حاصل از نقطه‌های ثبت شده در روی سطح که با عبور از کنار هد خواندن[7] حاصل می‌شود آشکار می‌گردد. مقدار سطح موجود برای ضبط در یک دیسک بزرگتر از درامی است که از نظر ابعاد با آن یکسان

---

1- Magnetic Drum
2- Mangetic Bubble Memory
3- Access time
4- Seek time
5- Transfer time
6- Write Head
7- Read Head

باشد. بنابراین اطلاعات بیشتری را می‌توان بر روی یک دیسک هم اندازه با درام ذخیره کرده باین دلیل، دیسک‌ها در اکثر کامپیوترها جایگزین درام شده‌اند.

## دیسک‌های مغناطیسی

یک دیسک مغناطیسی صفحه دواری است که از فلز یا پلاستیک پوشش یافته با مواد مغناطیسی ساخته شده است اغلب هر دو طرف دیسک بکار رفته و چندین دیسک ممکن است روی یک محور نصب شده و روی هر سطح هم هد خواندن - نوشتن وجود داشته باشد. تمام دیسک‌ها با هم و با سرعت زیادی می‌چرخند و به منظور دستیابی راه اندازی یا متوقف نمی‌شوند. بیت‌ها در سطح مغناطیس شده بصورت نقاطی بر روی دوایر هم مرکزی بنام شیار[1] ذخیره می‌شوند. شیارها معمولاً به بخش‌هایی به نام قطاع[2] تقسیم می‌شوند. در بیشتر سیستم‌ها، حداقل اطلاعات که می‌تواند انتقال یابد یک قطاع است. تقسیم‌بندی یک دیسک به شیارها و قطاع‌ها در شکل ۵-۱۲ نشان داده شده است.

برخی از دستگاه‌ها از یک هد خواندن و نوشتن برای هر سطح دیسک استفاده می‌کنند. در این دستگاه‌ها قبل از خواندن و نوشتن، بیت های آدرس شیارها برای حرکت دادن هد خواندن یا نوشتن بر روی مکان موردنظر بکار می‌روند. در نوعی دیگر از سیستم‌های دیسک، هدهای خواندن - نوشتن جداگانه برای هر شیار در هر سطح تهیه شده است. سپس بیت‌های آدرس، شیار خاصی را بطور الکترونیکی از طریق یک دیکدر انتخاب می‌کنند. این نوع دستگاه‌ها گرانتر بوده و فقط بر روی سیستم‌های کامپیوتری بسیار بزرگ یافت می‌شوند.




**شکل ۵-۱۲** دیسک مغناطیسی

---

1- Track 2- Sector

در دیسک‌ها از شیارهای زمانبندی دائمی برای همگام کردن بیت‌ها و تشخیص قطاع‌ها استفاده می‌شود. یک سیستم دیسک با بیت‌های آدرسی که شماره دیسک، سطح دیسک، شماره قطاع، و شیار موردنظر در قطاع مربوطه را مشخص می‌کنند آدرس دهی می‌شود. پس از قرار گرفتن هد خواندن - نوشتن در شیار موردنظر، سیستم باید با چرخش دیسک تا رسیدن قطاع مشخص شده به زیر هد خواندن و نوشتن صبر کند. پس از رسیدن به ابتدای قطاع، انتقال اطلاعات با سرعت صورت می‌گیرد. دیسک ها ممکن است هدهای متعدد داشته باشند و بیت‌های چندین شیار بطور همزمان انتقال یابند.

در هر قطاع مفروض، شیاری که در نزدیکی لبه خارجی قرار دارد طولانی‌تر از شیاری است که در نزدیکی مرکز دیسک می‌باشد. اگر بیت‌ها با تراکم مساوی ذخیره شوند، برخی از شیارها دارای بیت‌های ضبط شده بیشتری خواهند بود. برای هم طول کردن همه شیارهای یک قطاع، برخی از دیسک‌ها از تراکم ضبط متغیر استفاده می‌کنند، بطوری که شیارهای نزدیکتر به مرکز نسبت به شیارهای نزدیک به لبه خارجی تراکم بیشتری دارند. باین ترتیب تعداد بیت‌های موجود بر روی همه شیارهای یک قطاع، معین یکسان خواهند بود.

دیسک‌هایی که بطور دائمی به مجموعه دستگاه متصل‌اند و آنها را هرکاربری نمی‌تواند از مجموعه خارج کند، دیسک سخت [1] می‌خوانند. دیسک خوانی که دیسک‌های قابل جابجایی داشته باشد دیسک‌نرم [2] خوانده می‌شود. دیسک‌هایی که در یک دیسک خوان نرم بکار می‌روند دیسک‌های قابل جابجایی کوچکی هستند که از پلاستیک ساخته شده و با ماده مغناطیسی ضبط کننده پوشش داده شده‌اند. دو سایز برای این دیسک‌ها متداول است، قطر 5.25 و قطر 3.5 اینچ. دیسک‌های 3.5 اینچی کوچکترند و می‌توانند داده‌های بیشتری را نسبت به دیسک‌های 5.25 اینچی ذخیره کنند. دیسک‌های نرم گسترده‌ای در کامپیوترهای شخصی بعنوان وسیله‌ای برای توزیع نرم افزار بین کاربران کامپیوتر بکار می‌روند.

## نوار مغناطیسی

حامل نوار مغناطیسی متشکل است از اجزاء الکتریکی، مکانیکی و الکترونیکی که قطعات و مکانیزم کنترل را برای دستگاه نوار مغناطیسی فراهم می‌نمایند. خود نوار باریکه ای از پلاستیک پوشیده از ماده مغناطیسی ضبط کننده است. بیت‌ها بشکل نقاط مغناطیسی روی نوار در طول چند شیار ضبط می‌شوند. معمولاً هفت یا نه رکورد بطور همزمان ضبط می‌شوند تا به همراه بیت توازن تشکیل یک رکورد را بدهند. هدهای خواندن و نوشتن هر کدام بر روی یک شیار قرار می‌گیرند بطوری که داده‌ها بصورت رشته ای از کاراکترها می‌توانند ضبط و یا خوانده شوند.

دستگاه‌های نوار مغناطیسی را می‌توان متوقف کرد، بسمت جلو یا عقب راه اندازی کرد، یا آن را به عقب برگرداند. با وجود این نمی‌توان آنها را از بین دو کاراکتر شروع و یا بین آنها متوقف کرد. بهمین دلیل، اطلاعات بصورت بلاک هایی که به آنها رکورد می گویند ضبط می‌شود. فواصلی از نوار ضبط

---

1- Hard Disk 2- Floppy Disk

نشده بین رکوردها گنجانیده می‌شود تا در آ نجا بتوان نوار را متوقف نمود. نوار هنگامی که روی یکی از فواصل قرار دارد شروع به حرکت می‌کند و تا زمانی که به رکورد بعدی برسد سرعت ثابت خود را بدست می‌آورد. هر رکورد واقع در روی نوار یک الگوی شناسایی بیتی در ابتدا و انتهای خود دارند. با خواندن الگوی بیتی واقع در ابتدای رکورد، کنترل نوار شماره رکورد را تشخیص می‌دهد. با خواندن الگوی بیتی واقع در انتها، واحد کنترل شروع یک فاصله را درک می‌کند. دستگاه نوار با مشخص کردن شماره رکورد و تعداد کاراکترهای رکورد آدرس دهی می‌شود. رکوردها ممکن است دارای طول ثبات یا متغیر باشند.

## ۱۲-۴ حافظه تداعیگر

بسیاری از کاربردهای داده‌پردازی به جستجو اقلامی که در یک جدول حافظه ذخیره شده‌اند نیاز است. یک برنامه اسمبلر جدول آدرس‌های سمبل را برای استخراج معادل دودویی جستجو می‌کند. ممکن است برای تعیین نام وضعیت حساب یکی از مشتریان بانک، شماره حساب وی در فایل جستجو شود، روش متداول در جستجوی جدول، ذخیره اقلام آن در جایی است که بتواند بصورت متوالی آدرس دهی شود. رویه جستجو سیاستی است برای انتخاب رشته‌ای از آدرسها، خواندن محتوای حافظه در هر آدرس، و مقایسه اطلاعات خوانده شده با داده تحت جستجو است تا اینکه انطباق رخ دهد. تعداد دفعات مراجعه با دسترسی به حافظه به مکان داده و کارآیی الگوریتم بستگی دارد. برای به حداقل رساندن تعداد مراجعات برای یافتن یک داده، الگوریتم‌های جستجوی متعددی ابداع شده است.

اگر بتوان داده‌های ذخیره شده در حافظه را به منظور دستیابی به آنها براساس محتوای خود داده و نه براساس یک آدرس شناسایی کرد، زمان لازم برای یافتن یک قلم داده تا حد قابل توجهی کاهش می‌یابد. واحد حافظه‌ای که دستیابی به آن براساس محتوا انجام می‌شود، حافظه تداعیگر[1] یا حافظه قابل آدرس دهی با محتوا (CAM) خوانده می‌شود دستیابی به این نوع حافظه بطور همزمان و موازی و براساس محتوای داده، بجای آدرس یا مکان خاص، انجام می‌شود. حافظه می‌تواند یک مکان خالی بدون استفاده را برای ذخیره کلمه مورد نظر بیابد. وقتی که قرار باشد کلمه‌ای از حافظه تداعیگر خوانده شود، محتوای کلمه، یا بخشی از کلمه مشخص می‌شود. حافظه تمام کلماتی راکه برای محتوای مشخص شده منطبق شوند برای خواندن علامت گذاری می‌کند.

حافظه تداعیگر بعلت سازمان داخلی‌اش خصوصاً برای جستجوهای موازی براساس داده‌ها مناسب است. بعلاوه جستجو می‌تواند بر روی کل کلمه و یا روی میدان خاصی در یک کلمه انجام شود. حافظه تداعیگر گرانتر از حافظه با دستیابی تصادفی است زیرا هر سلول آن علاوه برقابلیت ذخیره‌سازی باید دارای مدارهای منطقی برای انطباق محتوای آن با یک آرگومان خارجی نیزباشد. به همین دلیل، حافظه‌های تداعیگر در کاربردهایی که در آنها زمان جستجو بسیار اهمیت دارد و باید کوتاه باشد استفاده می‌شود.

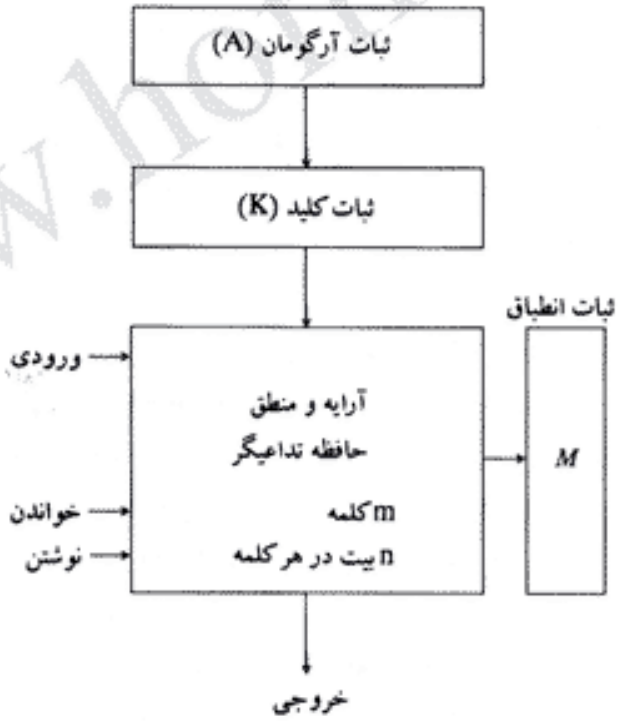
---

1- Associative Memory

## سازمان سخت افزاری

بلاک دیاگرام یک حافظه تداعیگر در شکل ۶-۱۲ داده شده است. این حافظه متشکل از یک آرایه حافظه و منطق لازم برای m کلمه n بیتی در هر کلمه است. ثبات آرگومان[1] و ثبات کلید[2] k هر یک n بیت دارند، یعنی یک بیت به ازای هر یک از بیت‌های کلمات. ثبات انطباق M، m بیتی است، یعنی یک بیت بازاء هر بیت از کلمه حافظه. هر کلمه در حافظه بطور موازی با محتوای ثبات آرگومان مقایسه می‌شود. کلماتی که با بیت‌های ثبات آرگومان تطبیق داشته باشند بیت متناظری را در ثبات انطباق 1 می‌کنند. پس از روند تطبیق، بیت‌هایی در ثبات انطباق که 1 شده‌اند نشان می‌دهند که کلمات متناظر آنها با کلمه مورد جستجو مطابقت دارند. عمل خواندن با مراجعهٔ متوالی به کلماتی از حافظه که بیت‌های متناظر آنها در ثبات انطباق 1 شده‌اند انجام می‌شود.

ثبات کلید پوششی را برای انتخاب یک میدان خاص یا کلید آرگومان فراهم می‌آورد. اگر ثبات کلید تماماً 1 باشد، کل آرگومان با هریک از کلمات حافظه مقایسه می‌شود. در غیراینصورت، فقط بیت‌هایی در آرگومان که محل متناظر آنها در ثبات کلید مقدار 1 دارد مقایسه می‌شوند. بنابراین ثبات کلید پوشش و یا قطعه اطلاعات شناسایی را برای ارجاع به حافظه فراهم می‌کند. برای تشریح مطلب توسط یک مثال عددی، فرض کنید که ثبات آرگومان A و ثبات کلید K دارای آرایش بیتی زیر، باشند. تنها سه بیت




شکل ۶-۱۲ بلاک دیاگرام حافظه تداعیگر

1- Argument Register 2- Switch Register

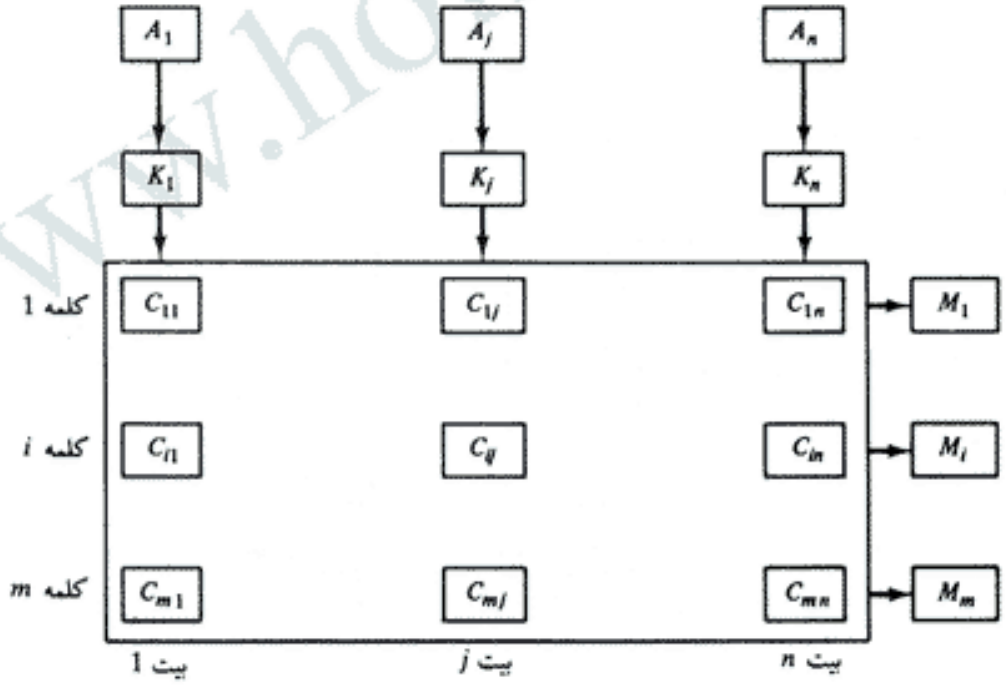
سمت چپ A با کلمات مقایسه زیرا K در این مکان‌ها دارای 1 است.

| | | | |
|---|---|---|---|
| A | 101 | 111100 | |
| K | 111 | 000000 | |
| کلمه 1 | 100 | 111100 | انطباق ندارد |
| کلمه 2 | 101 | 000001 | انطباق دارد |

کلمه 2 با میدان آرگومان پوشش نیافته تطبیق دارد زیرا سه بیت سمت چپ آرگومان و کلمه برابرند.

رابطه بین آرایه حافظه و ثبات‌های خارجی در حافظه تداعیگر در شکل ۷-۱۲ نشان داده شده است. سلول‌های آرایه با حرف C و دو زیر نویس مشخص شده‌اند. اولین زیرنویس شماره کلمه و دومین زیرنویس مکان بیت را در کلمه مشخص می‌کند. باین ترتیب سلول $C_{ij}$ مربوط به بیت j در کلمه i است. بیت $A_j$ در ثبات آرگومان با همه بیت‌های ستون j از آرایه مقایسه می‌شود بشرطی که $K_j=1$ باشد. این عمل برای همه ستون‌های $j=1,2, \ldots, n$ انجام می‌شود.

اگر بین همه بیت‌های پوشش[1] نیافته آرگومان و بیت‌های کلمه، تطابق وجود داشته باشد، بیت متناظر در ثبات انطباق یعنی $M_i$ برابر با 1 می‌شود. اگر یک یا چند بیت پوشش نیافته و کلمه مطابقت نداشته باشند $M_i$ برابر با 0 خواهد بود.




**شکل ۷-۱۲** حافظه تداعیگر به m کلمه و n بیت بازاء هر کلمه

1- Mask

سازمان درونی یک سلول نوعی $C_{ij}$ در شکل ۸-۱۲ نشان داده شده است. این سلول از یک المان ذخیره‌سازی فلیپ فلاپ $F_{ij}$ و مدارات لازم برای خواندن و نوشتن و انطباق سلول‌ها تشکیل شده است. بیت ورودی در عمل نوشتن به سلول ذخیره سازی انتقال می‌یابد. در حین عمل خواندن بیت ذخیره شده خوانده می‌شود. مدار منطقی محتوای سلول ذخیره سازی را با بیت متناظر پوشش نیافته آرگومان مقایسه کرده و یک خروجی را در اختیار مدار تصمیم‌گیری قرار می‌دهد تا بیت $M_i$ را 1 کند.

### مدار انطباق

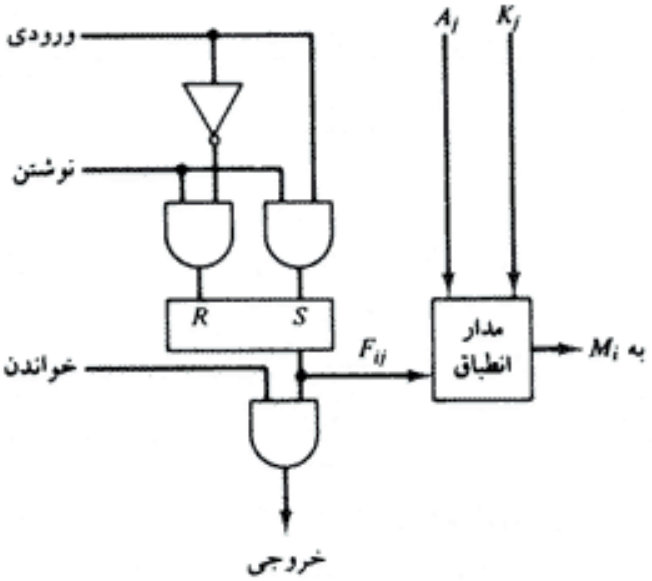
مدار منطقی برای هر کلمه را می‌توان از الگوریتم مقایسه دو عدد دودویی بدست آورد. ابتدا بیت‌های کلید را نادیده می‌گیریم و آرگومان A را با بیت های ذخیره شده در سلول‌های کلمه مقایسه می کنیم. کلمه i برابر با آرگومان در A است بشرطی که $A_j=F_{ij}$ بازاء $j=1,2,...n$ باشد. اگر هر دو بیت مساوی 1، یا 0 باشند، تساوی دو بیت بصورت تابع بول با عبارت زیر بیان می شود.

$$x_j = A_jF_{ij} + A'_jF'_{ij}$$

که در آن $x_j=1$ است اگر هر دو جفت بیت در مکان j مساوی باشد، در غیراینصورت $x_j=0$ خواهد بود. برای اینکه کلمه i برابر آرگومان A شود، تمام متغیرهای $x_j$ باید برابر 1 باشند. این وضعیت شرط 1 شدن بیت انطباق متناظر یعنی $M_i$ است، تابع بول برای این شرط عبارت است از

$$M_i = x_1x_2x_3...x_n$$

که بمعنی انجام عمل AND بر روی همه جفت بیت‌های انطباق یافته در یک کلمه است.




شکل ۸-۱۲ یک سلول از حافظه تداعیگر

اکنون بیت کلید $K_j$ را در مدار مقایسه می‌گنجانیم. اگر $K_j=0$ باشد، بیت های متناظر در $A_j$ و $F_j$ نیازی به مقایسه ندارند. فقط اگر $K_j=1$ باشد آنها باید مقایسه شوند. این شرط با OR کردن هر جمله بـا $K'_j$ حاصل می شود، بنابراین:

$$x_j + K'_j = \begin{cases} x_j & \text{اگر } K_j = 1 \\ 1 & \text{اگر } K_j = 0 \end{cases}$$

هر گاه $K_j=1$ باشد داریم $k'_j=0$ و $x_j+0=x_j$ هر وقت هم $K=0$ باشد، آنگاه $K'_j=1$ و $x_j+1=1$ می‌گردد. جمله $(x_j+K'_j)$ در وضعیت 1 خواهد بود اگر جفت بیت های آن مقایسه نشوند. این مطلب الزامی است زیرا هر جمله با همه جملات دیگر AND می‌شود و خروجی 1 اثری روی حاصل نخواهد داشت. فقط هنگامی که $K_j=1$ باشد مقایسه بیت‌ها تأثیر خواهد داشت.

اکنون می‌توان مدار انطباق برای کلمه i در یک حافظه تداعیگر را با تابع بولی زیر بیان کرد:

$$M_i = (x_1 + K'_1)(x_2 + K'_2)(x_3 + K'_3)\cdots(x_n + K'_n)$$

هر جمله در عبارت بشرطی 1 است که $K_j$ متناظر آن 0 باشد. اگر $K_j=1$ باشد، جمله بسته به مقدار $x_j$ برابر با 0 یا 1 خواهد بود. اگر همه جملات 1 باشد انطباق صورت گرفته و $M_j$ برابر 1 خواهد بود.

اگر تعریف اولیه $x_j$ را بجای آن قرار دهیم، تابع بول فوق بصورت زیر بیان می شود:

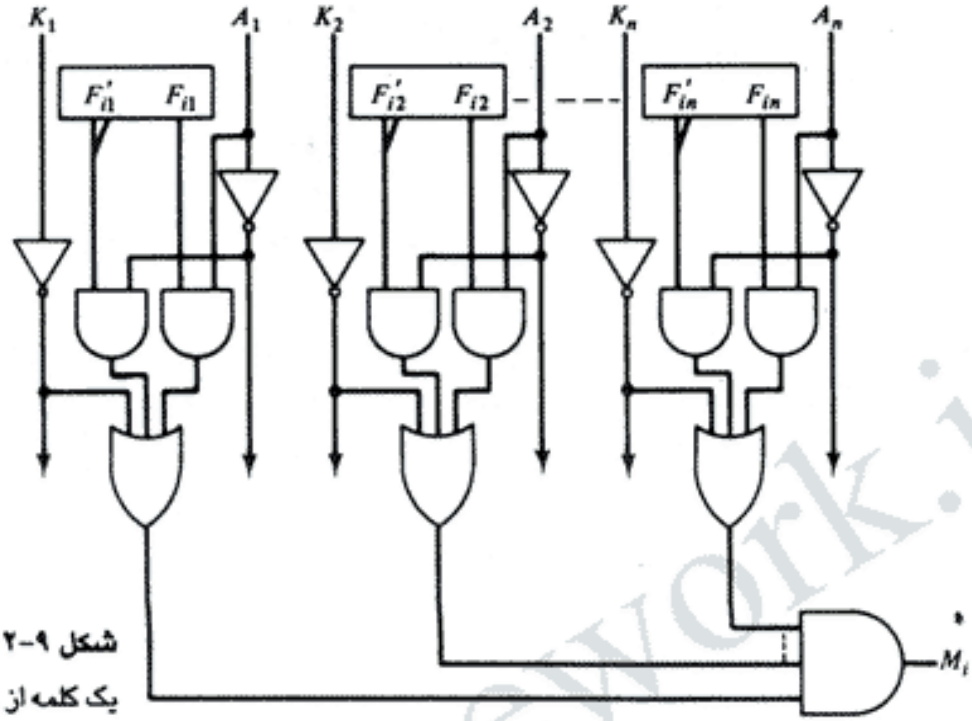
$$M_i = \prod_{j=1}^{n} (A_j F_{ij} + A'_j F'_{ij} + K'_j)$$

که در آن $\Pi$ سمبل ضرب و مشخص کننده AND تمام n جمله است. ما به m تابع این چنینی نیاز داریم که هر یک مربوط به یک کلمه m و ... و 3 و 2 و $i = 1$ است.

مدار انطباق برای هر کلمه در شکل ۹-۱۲ دیده می‌شود. هر سلول دو گیت AND و یک گیت OR نیاز دارد. معکوس کننده‌های $A_j$ و $K_j$ یک بار برای هر ستون لازم‌اند و برای همه بیت‌های ستون بکار می‌روند. خروجی تمام گیت‌های OR در سلول‌های یک کلمه بـه ورودی یک گـیت AND مشترک می‌روند سیگنال انطباق $M_i$ را ایجاد کنند. اگر انطباقی رخ دهد $M_i$ برابر 1 منطقی خواهد بود واگر انطباق رخ ندهد 0 است. توجه کنید که اگر ثبات کلید کلاً 0 باشد، خروجی $M_i$ صرف نظر از مقدار A یا کلمه 1 خواهد بود. در طول عملکرد عادی باید از چنین وضعی اجتناب شود.

## عمل خواندن

اگر بیش از یک کلمه با میدان آرگومان پوشش نیافته تطبیق نماید، بیت های متناظر با همه کلمات در ثبات انطباق مقدار 1 خواهند داشت. پس از آن لازم است تا بیت های ثبات انطباق یک به یک مرور




**شکل ۹-۱۲ مدار انطباق برای یک کلمه از حافظه تداعیگر**

شوند. کلمات انطباق یافته با اعمال سیگنال خواندن به ورودی خواندن هر کلمه‌ای که بیت $M_i$ متناظر آن 1 است بطور متوالی خوانده می‌شود.

در اغلب کاربردها، حافظه تداعیگر جدولی که در آن هیچ دو داده یکسانی تحت یک کلید ندارد را ذخیره می‌نماید. در این حالت، فقط یک کلمه ممکن است با میدان آرگومان پوشش نیافته مطابقت داشته باشد. با اتصال مستقیم $M_i$ به خط خواندن که در موقعیت همان کلمه (بجای ثبات M)، محتوای کلمه منطبق بطور اتوماتیک در خروجی ظاهر می‌شود و هیچ سیگنال فرمان خواندن بخصوصی لازم نیست. بعلاوه، اگر ما کلماتی راکه محتوای صفر دارند مستثنی کنیم، یک خروجی تمام صفر بیانگر عدم تطبیق بوده و داده مورد جستجو در حافظه موجود نیست.

## عمل نوشتن

یک حافظه تداعیگر باید قابلیت ذخیره سازی اطلاعات مورد جستجو را داشته باشد، نوشتن در یک حافظه تداعیگر می‌تواند، بسته به کاربرد اشکال مختلفی داشته باشد. اگر کل حافظه از قبل با اطلاعات جدیدی بار شده باشد، سپس نوشتن با آدرس دهی متوالی هر مکان میسر است. به این ترتیب مدار مورداستفاده به یک حافظه با دستیابی اتفاقی (RAM) برای نوشتن و به یک حافظه قابل آدرس‌دهی با محتوا برای خواندن تبدیل خواهد شد. در اینجا مزیت این است که آدرس برای ورودی مانند یک حافظه RAM دیکد می‌شود. در نتیجه بعوض داشتن m خط آدرس یعنی یک خط برای هر کلمه در حافظه.

تعداد خطوط آدرس بوسیله دیکدر به d خط کاهش می‌یابد که در آن $m=2^d$ است.

اگر که قرار باشد کلمه ناخواسته‌ای پاک شود و کلمات جدیدی یک به یک وارد شوند، به ثبات خاصی برای تمایز بین کلمات فعال و غیر فعال نیاز است. تعداد بیت های این ثبات که گاهی اوقات ثبات نشانه نامیده می‌شود، به تعداد کلمات در حافظه خواهد بود. به ازاء هر کلمه فعال ذخیره شده در حافظه، بیت متناظری در ثبات نشانه 1 می‌شود. با 0 کردن بیت نشانه کلمه متناظر آن در حافظه پاک می‌شود. برای ذخیره کلمات در حافظه، ثبات نشانه مرور می‌شود تا اینکه اولین 0 دیده شود. باین ترتیب اولین کلمه غیر فعال و نیز محل برای نوشتن یک کلمه جدید مشخص می‌شود. پس از ذخیره کلمه جدید در حافظه، این کلمه با 1 کردن بیت نشانه فعال می‌شود. اگر قرار باشد یک مکان خالی با تماماً 0 نشان داده شود می توان پس از حذف کلمه ناخواسته از حافظه، آن راپاک کرد. بعلاوه کلماتی که دارای بیت نشانه 0 هستند باید با آرگومان کلمه پوشش یابند (علاوه بر پوششی که بابیت‌های $K_j$ انجام می‌شود) تا فقط کلمات فعال مقایسه شوند.

## ۱۲-۵ حافظه کش

تحلیل تعداد زیادی از برنامه نشان داده است که ارجاعات به حافظه در هر فاصله زمانی معینی غالباً محدود به چند ناحیه موضعی در حافظه است. این پدیده، خاصیت محلیت ارجاع خوانده می‌شود. دلیل این خاصیت را می توان با بررسی این موضوع که یک برنامه کامپیوتر نوعاً بشکل مستقیم الخط پیش می رود و ضمن پیشروی با حلقه‌ها و زیر روال‌ها مکرراً برخورد می‌شود دریافت. وقتی که یک حلقه برنامه اجرا می‌شود، CPU متوالیاً به مجموعه‌ای از دستورات که حلقه را در حافظه تشکیل می‌دهند رجوع می نماید. هر بار یک زیر روال فراخوانی شود، مجموعه دستورالعمل های آن از حافظه برداشت می‌شود. بنابراین حلقه‌ها و زیر روال ها تمایل به محلی کردن ارجاع به حافظه برای برداشت دستورالعمل‌ها دارند. ارجاع داده به حافظه نیز، تا حدودی میل به محلی بودن دارد. رویه‌های جستجوی جدول مکرراً به بخشی از حافظه که جدول در آن است مراجعه می‌کنند. رویه‌های تکراری به مکان های حافظه مشترکی مراجعه می‌کنند و آرایه های اعداد به بخشی محلی از حافظه محدودند. نتیجه همه این ملاحظات خاصیت محلیت ارجاع است، که می‌گوید در فواصل زمانی کوتاه، آدرس‌های تولید شده توسط یک برنامه مکرراً به چند ناحیه محلی از حافظه اشاره می‌کنند، در حالی که بقیه برنامه بدفعات کمتری مورد دستیابی قرار می‌گیرند.

اگر بخش‌های فعال برنامه و داده‌ها در یک حافظه کوچک و سریع جای داده شوند، زمان دستیابی متوسط می‌تواند کاهش یافته و باین ترتیب زمان کل اجرای برنامه تقلیل می‌یابد. چنین حافظه سریع و کوچک، حافظه کش [1] خوانده می‌شود. این حافظه همان طور که در شکل ۱-۱۲ دیده شد بین CPU و حافظه اصلی قرار می‌گیرد. زمان دستیابی حافظه کش به نسبت ۵ تا ۱۰ برابر کمتر از زمان دستیابی حافظه اصلی

1- Cache

است. حافظه کش سریع ترین جزء در سلسله مراتب حافظه بوده و به حوالی سرعت CPU نزدیک است.

ایده اصلی سازمان کش این است که با نگهداری دستورالعمل‌ها و داده‌هایی که بیشتر دستیابی می‌شوند در یک حافظه کش اصلی سریع، زمان متوسط دستیابی حافظه به زمان دستیابی کش نزدیک شود. گرچه کش تنها بخش کوچکی از حافظه اصلی است، بدلیل خاصیت محلیّت ارجاع به حافظه ، بخش عمده اطلاعات یا در واقع نیازهای حافظه‌ای در حافظه سریع کش، یافت خواهد شد.

عملکرد اساسی سازمان حافظه بدین شرح است. وقتی که CPU نیاز به دستیابی به حافظه دارد، حافظه کش بررسی می‌شود. اگر کلمه موردنظر در کش یافت شود، از این حافظه سریع خوانده می‌شود. اگر کلمه‌ای که بوسیله CPU آدرس دهی داده شده در کش یافته نشود، حافظه اصلی برای خواندن کلمه آدرس دهی می‌شود. سپس بلاکی از کلمات از جمله کلمه‌ای که در آن لحظه دستیابی شد از حافظه اصلی به حافظه کش انتقال می‌یابد. اندازه بلاک می‌تواند از یک کلمه (کلمه‌ای که در لحظه اول دستیابی می‌شود) و 16 کلمه مجاور آن باشد. بدین ترتیب، مقداری داده به کش انتقال می‌یابد و لذا ارجاعات آینده به حافظه، کلمات مورد نیاز را در حافظه کش سریع بیابد.

عملکرد حافظه کش غالباً براساس کمیتی که نسبت برد[1] نامیده می‌شود سنجیده می‌گردد. هنگامی که CPU به حافظه مراجعه کند و کلمه موردنظر را در حافظه کش بیابد گفته می‌شود برد صورت گرفته است. اگر کلمه موردنظر در حافظه کش یافته نشود، مسلماً در حافظه اصلی است و باخت[2] صورت گرفته است. نسبت تعداد بردها تقسیم بر تعداد کل ارجاعات CPU به حافظه (برد بعلاوه باخت) نسبت برد است.

بهترین نسبت برد اندازه‌گیری شده تجربی، اجرای برنامه‌های نمونه در کامپیوتر و سنجش تعداد بردها و باخت‌ها در یک فاصله زمانی معین است. در این رابطه نسبت بردهای 0.9 و بالاتر گزارش شده است. این نسبت برد بالا اعتبار خاصیت محلیت ارجاع راتأیید می‌کند

زمان متوسط دستیابی به حافظه برای یک سیستم کامپیوتر با استفاده از حافظه کش می‌تواند بمقدار قابل ملاحظه‌ای اصلاح شود. اگر نسبت برد نسبتاً بالا باشد بطوری که در بیشتر اوقات CPU به جای حافظه اصلی به حافظه کش مراجعه کند، زمان میانگین دستیابی به زمان دستیابی حافظه سریع کش نزدیکتر خواهد بود. مثلاً کامپیوتری با زمان 100ns برای حافظه کش، و زمان دستیابی 1000ns برای حافظه اصلی، و نسبت برد 0.9 زمان میانگین دستیابی 200ns را خواهد داشت. این نتیجه یک اصلاح قابل توجه نسبت به کامپیوتر بدون حافظه کش است که دارای زمان دستیابی 1000ns می‌باشد.

مشخصه اصلی حافظه کش زمان دستیابی سریع آنست. بنابراین، وقتی جستجویی برای کلمه در کش صورت گیرد اتلاف زمانی کوتاه بوده یا اصلاً زمانی تلف نمی‌شود. انتقال داده از حافظه اصلی به حافظه کش فرآیند نگاشت[3] نام دارد. سه نوع رویه نگاشت هنگام بررسی سازمان حافظه کش عملاً مورد توجه است.

۱- نگاشت تداعیگر[4]

---

1- Hit ratio 2- Miss 3- Mapping

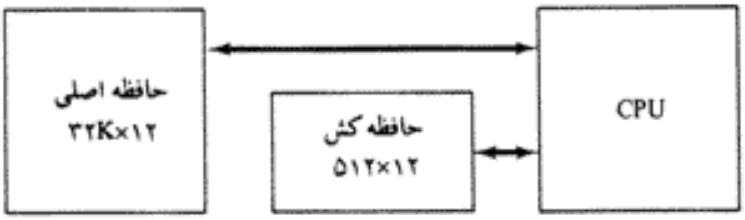
4- Assocative Mapping

۲- نگاشت مستقیم[1]
۳- نگاشت تداعیگر مجموعه‌ای[2]

برای کمک به درک بحث این سه رویه نگاشت ما از مثال خاصی برای سازمان حافظه که در شکل ۱۰-۱۲ نشان داده شده است استفاده می‌کنیم. حافظه اصلی می‌تواند 32k کلمه 12 بیتی را ذخیره کند. حافظه کش قادر است 512 عدد از این کلمات را در هر زمان ذخیره نماید. برای هر کلمه‌ای که در کش ذخیره شود، یک نسخه کپی در حافظه اصلی وجود دارد. CPU با هر دو حافظه ارتباط دارد. ابتدا آدرس 15 بیتی را به حافظه کش ارسال می‌نماید. اگر بردی صورت گیرد، CPU داده 12 بیتی را از کش می‌پذیرد. اگر باخت صورت گیرد، CPU کلمه را از حافظه اصلی خوانده و سپس آن کلمه به حافظه کش منتقل می‌شود.

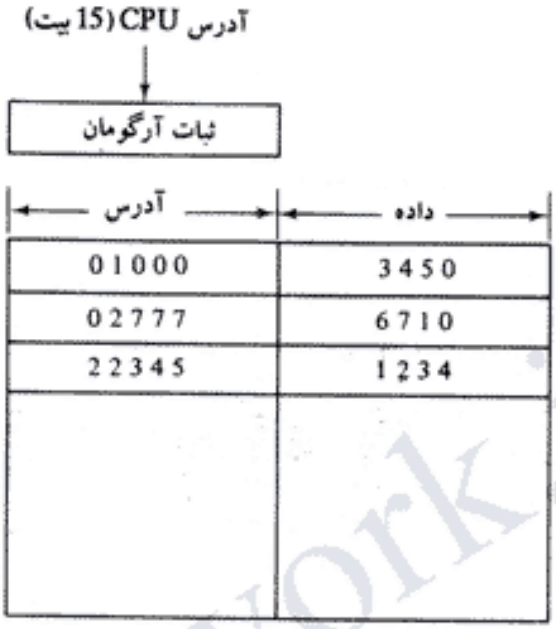
## نگاشت تداعیگر

در سریع‌ترین و انعطاف پذیرترین حافظه کش از یک حافظه تداعیگر استفاده می‌شود. این سازمان در شکل ۱۱-۱۳ نشان داده شده است. حافظه تداعیگر هم آدرس و هم محتوای (داده) کلمه حافظه را ذخیره می‌کند. این توانایی اجازه می‌دهد تا هر مکان در حافظه کش بتواند هر یک از کلمات حافظه اصلی را ذخیره کند. دیاگرام، سه کلمه را که در حال حاضر در حافظه کش ذخیره شده نشان می‌دهد. مقدار آدرس 15 بیتی بصورت یک عدد هشت هشتی پنج رقمی و کلمه 12 بیتی متناظر آن بصورت یک عدد هشت هشتی چهار رقمی نشان داده شده است. یک آدرس 15 بیتی بوسیله CPU در ثبات آرگومان قرار داده شده و حافظه تداعیگر برای تطبیق با آن آدرس جستجو می‌شود. اگر آدرس مورد نظر پیدا شود، داده 12 بیتی متناظر آن خوانده شده و به CPU ارسال می‌شود. اگر تطبیق صورت نگیرد، حافظه اصلی برای یافتن کلمه دستیابی می‌شود. سپس جفت کمیت آدرس - داده به حافظه کش تداعیگر منتقل می‌شود. اگر کش پر باشد، باید یک جفت آدرس - داده از آن خارج شود تا جا برای زوج موردنظر که در حافظه کش قرار ندارد باز شود. تصمیم در مورد زوجی که باید حذف شود براساس الگوریتمی است که طراح برای کش انتخاب می‌کند. یک رویه ساده این است که هر وقت که یک کلمه جدید از حافظه اصلی تقاضا شد




شکل ۱۰-۱۲ مثالی از حافظه کش

1- Direct Mapping 2- Set - Associative Mapping

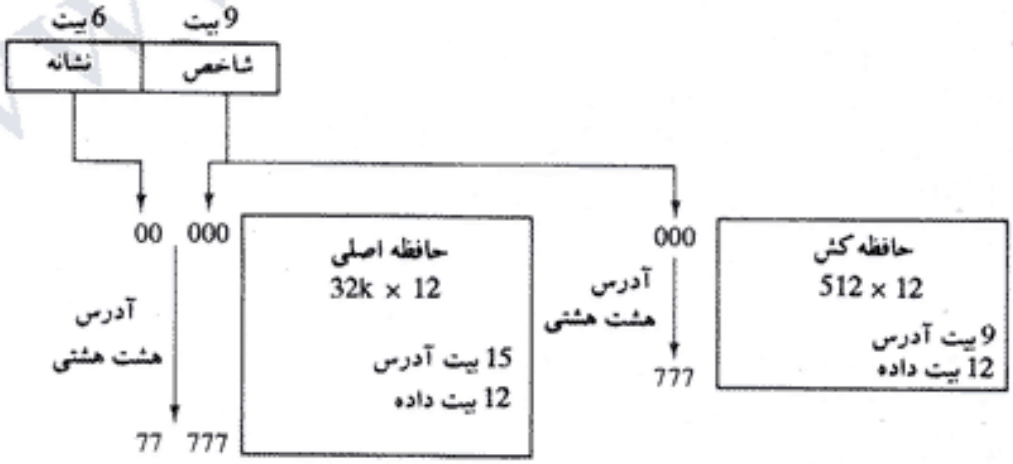



**شکل ۱۱-۱۲ حافظه کش با نگاشت تداعیگر**
**(همه اعداد در مبنای هشت هستند)**

سلول‌های موجود در حافظه کش بصورت چرخشی[1] جایگزین شوند. این رویه، سیاست اولین ورودی - اولین خروجی (FIFO) نام دارد.

### نگاشت مستقیم

حافظه‌های تداعیگر در مقایسه با حافظه های دستیابی تصادفی بدلیل مدار اضافی همراه هر سلول، گرانتر است. امکان استفاده از حافظه با دستیابی تصادفی برای حافظه کش در شکل ۱۲-۱۲ مورد بررسی

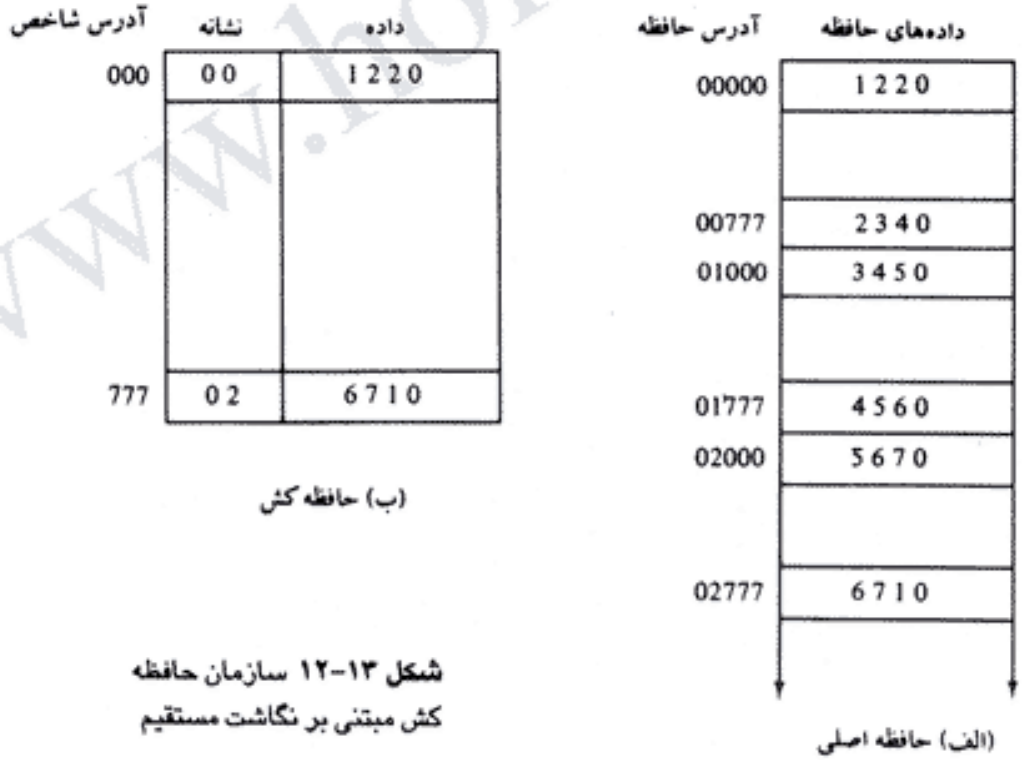



**شکل ۱۲-۱۲ روابط آدرس‌دهی بین حافظه‌های کش و اصلی**

1- Round - Robin order

قرار گرفته است. آدرس 15 بیتی CPU به دو میدان تقسیم شده است. نُه بیت کم ارزشتر میدان شاخص [1] و شش بیت باقیمانده میدان نشانه [2] را تشکیل می‌دهند. شکل نشان می‌دهد که حافظه اصلی نیاز به آدرسی دارد که هم شامل بیت‌های شاخص و هم شامل بیت های نشانه می‌باشد. تعداد بیت‌های میدان شاخص برابر با تعداد بیت‌های آدرس لازم برای دستیابی به حافظه کش است.

در حالت کلی، $2^k$ کلمه در حافظه کش و $2^n$ کلمه در حافظه اصلی وجود دارد. آدرس حافظه n بیتی به دو میدان تقسیم می‌شود: k بیت برای میدان شاخص و n-k بیت برای میدان نشانه. سازمان حافظه کش با نگاشت مستقیم، از آدرس n بیتی برای دستیابی به حافظه اصلی و k بیت شاخص برای دستیابی به حافظه کش استفاده می‌کند. سازمان داخلی کلمات در حافظه کش در شکل ۱۳-۱۲ (ب) نشان داده شده است. هر کلمه در کش متشکل از کلمه داده و نشانه مربوطه آنست. وقتی که کلمه جدیدی برای اولین بار به حافظه کش آورده شود، بیت‌های نشانه همراه با بیت‌های داده ذخیره می‌شوند. وقتی که CPU یک سیگنال تقاضای حافظه تولید می‌کند، میدان شاخص برای دستیابی به حافظه کش بکار می‌رود. میدان نشانه آدرس CPU، با نشانه موجود در کلمه خوانده شده از حافظه کش مقایسه می‌شود. اگر دو نشانه با هم تطبیق کنند یک برد صورت می‌گیرد و کلمه داده موردنظر در حافظه کش قرار دارد. اگر تطبیق صورت نگیرد، یک باخت وجود داشته و کلمه مطلوب از حافظه اصلی خوانده می‌شود. این کلمه سپس همراه با نشانه جدید در حافظه کش ذخیره شده و جایگزین مقدار قبلی می‌شود. عیب نگاشت مستقیم



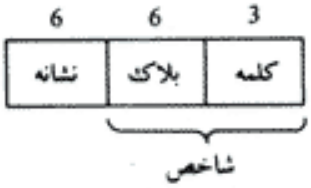

**شکل ۱۳-۱۲ سازمان حافظه کش مبتنی بر نگاشت مستقیم**

1- Index Field
2- Tag Field

این است که اگر دو یا چند کلمه که آدرسهایشان دارای شاخص یکسانی ولی نشانه متفاوت‌اند مرتباً دستیابی شوند نسبت برد بمیزان قابل توجهی افت نماید. با این وجود این امکان باین ترتیب به حداقل رسانده شده است که این چنین کلمات در محدوده آدرسهای نسبتاً دور از هم قرار دارند (در این مثال مضرب 512 مکان).

برای درک نحوه عملکرد سازمان نگاشت مستقیم مثال عددی شکل ۱۳-۱۲ را در نظر بگیرید. کلمه واقع در آدرس صفر اخیراً در حافظه کش ذخیره شده است (شاخص = 000، نشانه =00، داده =1220). فرض کنید که CPU اکنون بخواهد به کلمه‌ای واقع در آدرس 02000 دستیابی کند. آدرس شاخص 000 است، بنابراین این آدرس برای دستیابی به کش استفاده می‌شود. سپس دو نشانه مقایسه می‌شوند. نشانه کش 00 است ولی نشانه آدرس 02 است، که حالت تطابق را ایجاد نمی‌کند. بنابراین حافظه اصلی دستیابی شده و کلمه داده 5670 به CPU انتقال داده می‌شود. سپس کلمه کش در اندیس آدرس 000 با نشانه 02 و داده 5670 جایگزین می‌شود.

مثال نگاشت مستقیم که در بالا توصیف شد از بلاک‌هایی با اندازه یک کلمه استفاده می‌کند. سازمان مشابهی اما با استفاده از هشت کلمه در شکل ۱۴-۱۲ نشان داده شده است. میدان شاخص اکنون به دو بخش تقسیم شده است: میدان بلاک[1] و میدان کلمه[2]. در یک حافظه کش 512 کلمه‌ای، 64 بلاک 8 کلمه‌ای وجود دارد زیر 512=8×64 است. شماره بلاک با یک میدان 6 بیتی و محل کلمه در بلاک با یک میدان سه بیتی مشخص می‌شود. میدان نشانه که در حافظه کش ذخیره شده برای هر هشت کلمه یک بلاک یکسان است. هر بار یک باخت اتفاق بیفتد، تمام بلاک هشت کلمه‌ای باید از حافظه اصلی به حافظه کش انتقال یابد. هر چند این عمل وقتی اضافی را صرف می‌کند ولی نسبت برد بدلیل ماهیت ترتیبی برنامه‌های کامپیوتر با احتمال بسیار افزایش می‌یابد.




| شاخص | | نشانه | داده |
|---|---|---|---|
| بلاک 0 | 000 | 01 | 3450 |
| | 007 | 01 | 6578 |
| بلاک 1 | 010 | | |
| | 017 | | |
| | | | |
| بلاک 63 | 770 | 02 | |
| | 777 | 02 | 6710 |

**شکل ۱۴-۱۲** حافظه کش با نگاشت مستقیم و اندازه بلاک 8 کلمه

1- Block Field
2- Word Field

## نگاشت تداعیگر مجموعه‌ای

قبلاً اشاره شد که عیب نگاشت مستقیم این است که دو کلمه با شاخص یکسان در بیت آدرس ولی با مقادیر نشانه متفاوت نمی‌توانند بطور همزمان در حافظه کش قرار گیرند. نوع سومی از سازمان حافظه کش، که نسبت به نگاشت مستقیم از لحاظ اشکال فوق اصلاح شده است و نگاشت تداعیگر مجموعه‌ای خوانده می‌شود، می‌تواند دو یا چند کلمه با شاخص یکسان در آدرس را نگهداری کند. هر کلمه داده همراه با نشانه آن ذخیره می‌شود و تعداد اقلام نشانه - داده در هر کلمه از حافظه کش یک مجموعه نامیده می‌شود. مثالی از سازمان حافظه کش تداعیگر مجموعه‌ای برای مجموعه‌ای با اندازه 2 در شکل ۱۵-۱۲ نشان داده شده است. هر آدرس شاخص به دو کلمه داده و نشانه مربوطه‌اش اشاره می‌کند. هر نشانه به شش بیت نیاز دارد و هر کلمه داده 12 بیتی است. بنابراین طول کلمه 36=(12+6)2 بیت خواهد بود. هر آدرس شاخص 9 بیتی می‌تواند برای 512 کلمه جا مشخص کند. بنابراین اندازه حافظه کش 36×512 بیت است. این حافظه می‌تواند 1024 کلمه از حافظه اصلی را جای داد زیرا محتوای هر کلمه از حافظه کش دو کلمه داده است. بطور کلی، یک حافظه تداعیگر مجموعه‌ای با اندازه مجموعه k قادر است k کلمه از حافظه اصلی را در هر کلمه حافظه کش جای دهد.

اعداد هشت هشتی در شکل ۱۵-۱۲ براساس محتویات حافظه اصلی در شکل ۱۳-۱۲ (الف) لیست شده‌اند. کلمات در آدرس‌های 01000 و 02000 از حافظه اصلی در حافظه کش با آدرس شاخص 000 ذخیره شده‌اند . بطور مشابه، کلمات در آدرس‌های 02777 و 00777 در آدرس شاخص 777 در حافظه کش ذخیره شده‌اند. هنگامی که CPU یک تقاضای حافظه تولید می‌کند، مقدار شاخص آدرس برای دستیابی به کش مورد استفاده قرار می‌گیرد. سپس میدان نشانه آدرس CPU با هر دو نشانه در حافظه کش مقایسه می‌شود تا وقوع انطباق معین گردد. منطق مقایسه با جستجوی نشانه‌ها در مجموعه، به شیوه‌ای مشابه با جستجوی حافظه تداعیگر انجام می‌شود. نام "تداعیگر مجموعه‌ای" نیز از همینجا گرفته شده

| شاخص | نشانه | داده | نشانه | داده |
|---|---|---|---|---|
| 000 | 01 | 3450 | 02 | 5670 |
| | | | | |
| | | | | |
| 777 | 02 | 6710 | 00 | 2340 |

**شکل ۱۵-۱۲** حافظه کش با نگاشت تداعیگر در مجموعه دوتایی (اعداد در مبنای هشت هستند)

است. نسبت برد با افزایش اندازه مجموعه افزایش می‌یابد زیرا در اینصورت کلمات بیشتری با شاخص یکسان ولی نشانه‌های متفاوت می‌تواند در حافظه کش قرار داشته باشند. اما افزایش اندازه مجموعه تعداد بیت‌های کلمات کش را افزایش داده و مدارهای مقایسه پیچیده‌تری را نیاز خواهند داشت.

وقتی که باختی در حافظه کش تداعیگر مجموعه‌ای رخ دهد و مجموعه پر باشد، لازم است تا به جای یکی از اقلام نشانه - داده، مقدار جدیدی جایگزین شود. متداولترین الگوریتم جایگزینی بکار رفته عبارتند از: جایگزینی تصادفی، اولین ورودی - اولین خروجی (FIFO)، و قدیمترین مورد استفاده[1] (LRU). با سیاست جایگزینی تصادفی کنترل یک قلم نشانه- داده را برای جایگزین تصادفی برمی‌گزینند. رویه FIFO موردی که بیش از همه در مجموعه بوده است را انتخاب می‌کند. الگوریتم LRU اطلاعات Least Re...used بوسیله CPU را برمی‌گزینند. هر دو روش FIFO و LRU می‌توانند باافزایش چند بیت در هر کلمه کش پیاده سازی شوند.

## نوشتن در حافظه کش

یکی از جنبه‌های مهم سازمان کش به درخواست‌های نوشتن مربوط است. وقتی که CPU در حین عمل خواندن کلمه‌ای را در کش می‌یابد، حافظه اصلی در عمل انتقال دخالتی نمی‌کند. با این وجود اگر عمل یک نوشتن باشد، دو راه برای عملکرد سیستم وجود دارد.

ساده‌ترین و متداولترین رویه‌های مورد استفاده بهنگام کردن حافظه اصلی در قبال هر نوشتن حافظه است، که حافظه کش هم به موازات آن، اگر دارای کلمه موجود در آدرس مشخص شده باشد، بهنگام می‌گردد. این روش رویه کامل‌نویسی[2] خوانده می‌شود. مزیت این روش اینست که حافظه اصلی همواره داده یکسانی با حافظه کش دارد. این مشخصه در سیستم‌هایی که از روش انتقال با دستیابی مستقیم حافظه استفاده می‌کنند اهمیت دارد. روش مذکور اعتبار داده‌هایی راکه در تمام مدت در حافظه اصلی‌اند تضمین می‌کند، بنابراین وسیله I/O که از طریق DMA ارتباط دارد جدیدترین اطلاعات را دریافت خواهد کرد.

روش دوم، روش پس‌نویسی[3] پس نویسی خوانده می‌شود. در این روش فقط مکان کش در حین عمل نوشتن بهنگام می‌شود. سپس مکان مزبور با یک پرچم مشخص می‌شود تا بعداً که کلمه موجود در آن از حافظه کش خارج می‌شود بداخل حافظه اصلی کپی شود. دلیل بکارگیری روش پس نویسی این است که در زمان قرار گرفتن یک کلمه در حافظه کش، ممکن است چندین بار بهنگام شود؛ با این وجود مادامی که کلمه در کش قرار می‌گیرد، قدیمی بودن کپی واقع در حافظه اصلی مهم نیست، زیرا تقاضاهای کلمه از حافظه کش تأمین می‌شوند. فقط هنگامی که کلمه از حافظه کش خارج می‌شود لازم است نسخه صحیحی در حافظه اصلی نوشته شود. نتایج تحلیلی نشان می‌دهد که تعداد نوشتن‌ها در حافظه در یک برنامه، بین ۱۰ تا ۳۰ درصد کل ارجاعات به حافظه است.

---

1- Least Recently Used 2- Write Through 3- Write – Back

### مقدار دهی اولیه در حافظه کش

جنبه دیگری از سازمان کش که باید مورد توجه قرار گیرد مقداردهی اولیه آنست. حافظه کش هنگامی که کامپیوتر روشن می‌شود یا وقتی که حافظه اصلی با مجموعه کاملی از برنامه‌ها از حافظه کمکی بار می‌شود مقداردهی می‌گردد. پس از مقدار دهی اولیه حافظه کش خالی تصور خواهد شد، ولی در اصل حاوی مقداری داده‌های بی‌اعتبار است. مرسوم است تا به هر کلمه در کش یک بیت اعتبار[1] اختصاص یابد تا مشخص شود آیا کلمه حاوی داده معتبراست یا خیر.

حافظه کش با صفر کردن تمام بیت‌های معتبرش مقداردهی اولیه می‌شود. اولین باری که یک کلمه خاص از حافظه اصلی بار شود بیت اعتبار آن به مقدار 1 تغییر می‌یابد و تا مقداردهی اولیه بعدی همچنان برابر 1 باقی می‌ماند. معرفی بیت اعتبار بمعنی این است که آن کلمه با هیچ کلمه‌ای جایگزین نشود مگر اینکه بیت اعتبار 1 باشد و عدم تطابق نشانه‌ها رخ دهد. اگر بیت اعتبار 0 باشد، کلمه جدید بطور اتوماتیک جایگزین داده بی اعتبار می‌گردد. بنابراین شرط مقداردهی اولین این اثر را دارد که تا زمانی که حافظه نهان از داده‌های معتبر پر شود موجب باخت می‌شود.

## ۶-۱۲ حافظه مجازی

در یک سیستم سلسه مراتبی حافظه، برنامه‌ها و داده‌ها ابتدا در حافظه کمکی ذخیره می‌شوند. قسمت‌هایی از یک برنامه یا داده اگر قرار باشد مورد استفاده CPU واقع شوند به حافظه اصلی آورده می‌شوند. حافظه مجازی[2]، مفهومی مورد استفاده در برخی از سیستم‌های کامپیوتری بزرگ است که این امکان را به کاربر می‌دهد تا برنامه را بگونه‌ای بنویسید که گوئی فضای حافظه بزرگی باندازه تمام حافظه کمکی را در اختیار دارد. هر آدرسی که بوسیله CPU ارجاع شود از یک مکانیزم تبدیل آدرس عبور نموده و از یک آدرس مجازی به یک آدرس فیزیکی در حافظه اصلی بدل می‌شود. حافظه مجازی باین خاطر بکار می‌رود تا دربرنامه نویسی تصور را ایجاد کند که علیرغم داشتن یک حافظه نسبتاً کوچک، حافظه زیادی را در اختیار دارد. یک سیستم حافظه مجازی مکانیزمی برای ترجمه آدرس‌های تولید شده بوسیله برنامه، به آدرس‌های صحیح در حافظه اصلی را فراهم می‌کند. این عمل بصورت دینامیکی ضمن اجرای برنامه‌ها در CPU انجام می‌شود ترجمه یا نگاشت به صورت اتوماتیک بکمک جدول و بوسیله سخت افزار انجام می‌شود.

### فضای آدرس و فضای حافظه

آدرسی که بوسیله برنامه نویس مورد استفاده قرار می‌گیرد و آدرس مجازی[3] و مجموعه این آدرسها فضای آدرس[4] خوانده می‌شود. یک آدرس در حافظه اصلی، مکان[5] یا آدرس فیزیکی[6] نامیده می‌شود.

---

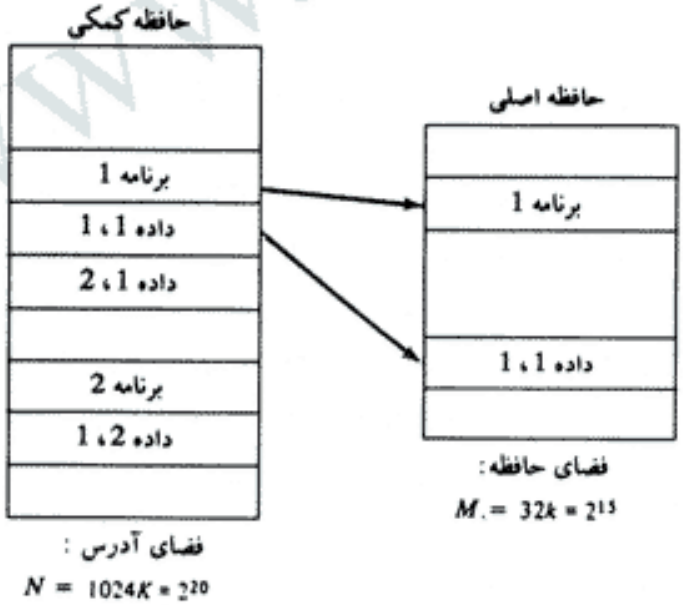
1- Valid Bit
2- Virtual Memory
3- Virtual Address
4- Address Space
5- Location
6- Physical Address

مجموعه چنین مکان‌هایی فضای حافظه[1] خوانده می‌شود. بنابراین فضای آدرس مجموعه‌ای از آدرس‌های تولید شده بوسیله برنامه‌ها می‌باشند، که به دستورالعمل‌ها و داده‌ها اشاره می‌کنند؛ و فضای حافظه هم متشکل از مکان‌های واقعی حافظه اصلی می‌باشند، که مستقیماً برای پردازش قابل آدرسی‌دهی هستند. در اکثر کامپیوترها فضاهای آدرس و حافظه یکی هستند. در کامپیوترهایی که حافظه مجازی دارند فضای آدرس بزرگتر از فضای حافظه است.

بعنوان تشریحی بر مطلب فوق، یک کامپیوتر با حافظه اصلی 32k کلمه (K=1024) را در نظر بگیرید. برای مشخص کردن آدرس فیزیکی در حافظه به پانزده بیت نیاز است چون $32K = 2^{15}$ است. فرض کنید که کامپیوتر دارای حافظه‌ای مجازی برای ذخیره $1024K = 2^{20}$ کلمه باشد. بنابراین حافظه کمکی دارای ظرفیت ذخیره سازی اطلاعات معادل با 32 برابر حافظه اصلی است. هرگاه فضای آدرس را با N و فضای حافظه را با M نشان دهیم، برای مثال فوق $N = 1024K$ و $M = 32K$ خواهد بود.

در سیستم‌های کامپیوتر چند برنامه‌ای با برنامه‌ها و داده‌ها براساس تقاضای CPU از حافظه اصلی به حافظه کمکی و یا از آن به حافظه اصلی انتقال می‌یابد. فرض کنید که برنامه 1 در حال حاضر در CPU در حال اجراست. برنامه 1 و بخشی از داده مربوط به آن از حافظه کمکی به حافظه اصلی طبق شکل ۱۶-۱۲ انتقال یابند. بخشهای مختلف برنامه‌ها و داده‌ها ضرورتاً لازم نیست در مکان‌های متوالی قرار داشته باشند، زیرا اطلاعات بطور مداوم به داخل و خارج حافظه انتقال می‌یابد و نیز ممکن است فضاهای خالی در مکان‌هایی پراکنده حافظه موجود باشند.

در یک سیستم با حافظه مجازی به برنامه نویس چنین گفته می‌شود که آنها تمام فضای آدرس را در اختیار دارند. بعلاوه، میدان آدرس کد دستورالعمل دارای تعداد بیت کافی برای مشخص کردن تمام




**شکل ۱۶-۱۲** رابطه بین فضای آدرس و حافظه در یک سیستم حافظه مجازی

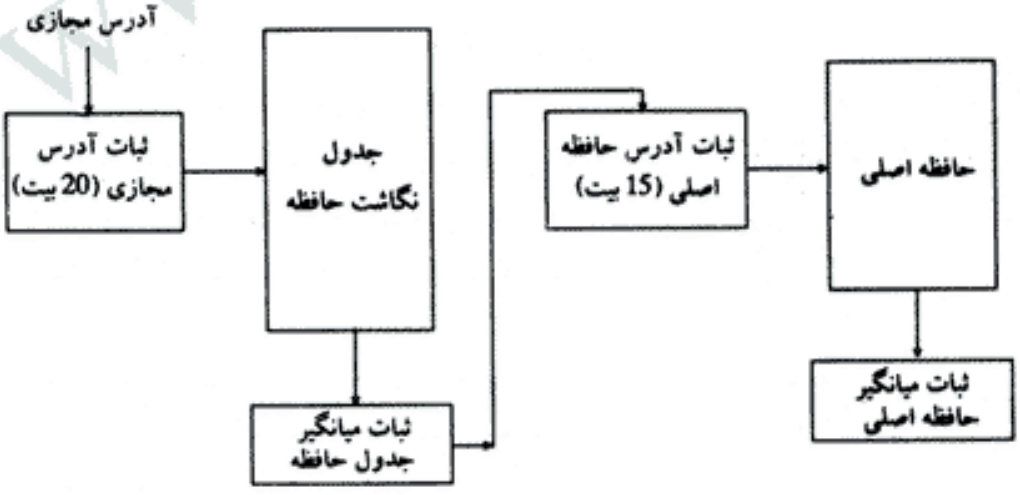
1- Memory Space

حافظه‌های مجازی است. در مثال ما، میدان آدرس یک کد دستورالعمل، از 20 بیت تشکیل شده است ولی آدرس‌های حافظه فیزیکی باید بوسیله 15 بیت معین شوند. بنابراین CPU دستورالعمل‌ها و داده‌ها را با آدرس 20 بیتی فرا می‌خواند، ولی اطلاعات در این آدرس بایستی از حافظه فیزیکی برداشت شود زیرا دستیابی به حافظه مجازی برای تک تک کلمات بیش از حد وقت گیر خواهد بود. (به یاد داشته باشید که برای کارآیی عمل انتقال، حافظه کمکی یک رکورد کامل را به حافظه اصلی منتقل می‌کند. سپس همانطور که در شکل ۱۷-۱۲ دیده می‌شود، جدولی برای نگاشت آدرس مجازی 20 بیتی به آدرس فیزیکی 15 بیتی لازم خواهد بود. این تبدیل آدرس یک عمل دینامیکی است، یعنی هر آدرس به محض ارجاع CPU به یک کلمه تبدیل می‌شود.

جدول تبدیل[1] می‌تواند در یک حافظه جداگانه طبق شکل ۱۷-۱۲ و یا در حافظه اصلی باشد. در حالت اول، یک واحد حافظه اضافی و نیز زمان دستیابی به حافظه اضافی لازم است. در حالت دوم، جدول حافظه اصلی را اشغال می‌کند و دو بار دستیابی به حافظه لازم می‌باشد ضمن اینکه برنامه با نصف سرعت عادی اجرا می‌شود. روش سوم استفاده از حافظه تداعیگر است که در ذیل تشریح می‌شود.

## نگاشت آدرس با استفاده از صفحات

پیاده‌سازی جدول تبدیل یا نگاشت آدرس بشرطی ساده می‌شود که اطلاعات در فضای آدرس و فضای حافظه هر یک به گروه‌هایی با اندازه ثابت تقسیم شوند. حافظه فیزیکی به گروه‌هایی با اندازه مساوی که بلاک خوانده می‌شود و هر یک می‌تواند از 64 تا 4096 کلمه داشته باشد شکسته می‌شود. اصطلاح صفحه[2] اشاره به گروه‌هایی از فضای حافظه هم اندازه دارد. مثلاً، اگر یک صفحه یا بلاک از 1K




شکل ۱۷-۱۲ جدول حافظه برای نگاشت آدرس‌های مجازی

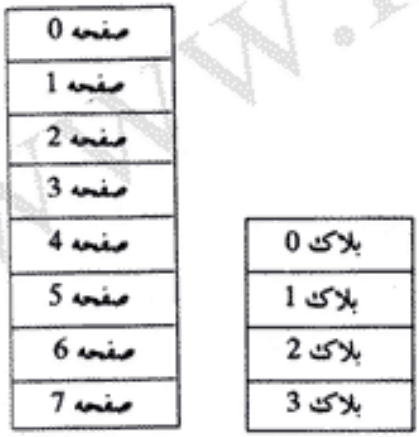
1- Mapping Table 2- Page

کلمه تشکیل شده باشد، سپس با استفاده از مثال قبل، فضای آدرس به 1024 صفحه و حافظه اصلی به 32 بلاک تقسیم می‌شود. هر چند یک صفحه و بلاک هر دو به گروهی از کلمات 1K تقسیم شده‌اند، یک صفحه به سازمان فضای آدرس و یک بلاک به سازمان فضای حافظه اشاره دارد. برنامه‌ها نیز منقسم شده به صفحات تصور می‌شوند. انتقال بخش‌هایی از برنامه‌ها از حافظه کمکی به حافظه اصلی بصورت رکوردهایی با اندازه مساوی با یک صفحه صورت می‌گیرد. گاهی از اصطلاح "کادر صفحه"[1] برای اشاره به بلاک استفاده می‌شود.

یک کامپیوتر با فضای آدرس 8K و فضای حافظه 4K را در نظر بگیرید. اگر ما هر یک را به گروه‌های 1K کلمه تقسیم کنیم هشت صفحه و چهار بلاک مطابق شکل ۱۸-۱۲ خواهیم داشت. در هر لحظه، تا چهار صفحه از فضای آدرس می‌تواند در هر یک از چهار بلاک از حافظه اصلی قرار گیرد.

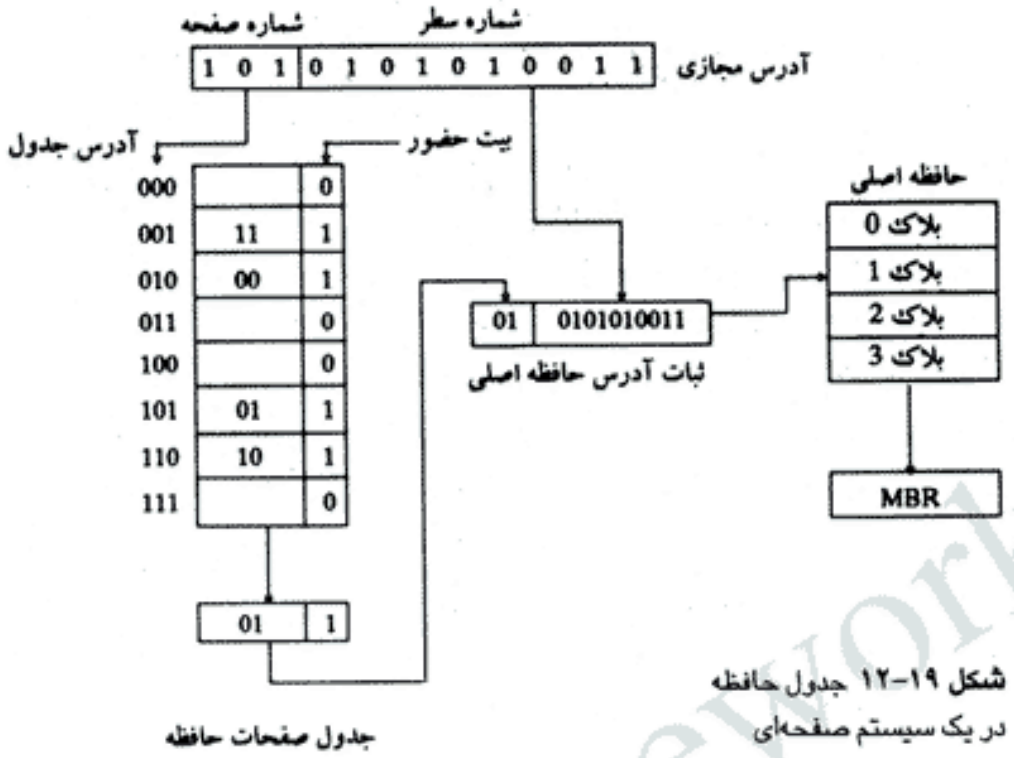
نگاشت از فضای آدرس به فضای حافظه می‌تواند ساده شود بشرطی که هر آدرس مجازی با دو عدد نشان داده شود: یک آدرس شماره صفحه و یک خط در صفحه. در کامپیوتری با $2^p$ کلمه در هر صفحه، p برای مشخص کردن آدرس خط و بقیه بیت‌های مرتبه بالاتر از حافظه مجازی برای مشخص کردن شماره صفحه مورد استفاده قرار می‌گیرد. در مثال شکل ۱۸-۱۲، آدرس مجازی دارای 13 بیت است. چون هر صفحه از $2^{10}=1024$ کلمه تشکیل شده است، سه بیت با ارزشتر آدرس مجازی یکی از هشت صفحه را مشخص خواهد کرد و 10 بیت کم ارزشتر آدرس سطر را در صفحه فراهم می‌کند. توجه کنید که آدرس سطر در فضای آدرس و فضای حافظه یکسان است؛ تنها تبدیل لازم از شماره‌صفحه به شماره‌بلاک می‌باشد.

سازمان جدول نگاشت یا تبدیل حافظه در یک سیستم صفحه‌بندی شده در شکل ۱۹-۱۲ نشان داده‌شده است. جدول صفحات حافظه از هشت کلمه تشکیل می‌شود، یعنی یک کلمه بازاء هر صفحه آدرس جدول صفحات شماره صفحه را نشان می‌دهد و محتوای کلمه شماره بلاکی را که صفحه مورد نظر در حافظه اصلی در آن قرار دارد مشخص می‌نماید. جدول نشان می‌دهد که صفحات 1، 2، 5 و 6 هم اکنون در حافظه اصلی بترتیب بلوک‌های 3، 0، 1 و 2 قرار دارند. یک بیت حضور در هر خانه مشخص می‌کند که صفحه از حافظه کمکی به حافظه اصلی منتقل شده است یا خیر. مقدار 0 در هر بیت حضور نشان می‌دهد که این صفحه در حافظه اصلی موجود نیست. CPU به هر کلمه واقع در حافظه با یک آدرس مجازی 13 بیتی رجوع می‌کند. سه بیت پرارزشتر آدرس مجازی شماره صفحه و نیز آدرسی را برای جدول صفحات حافظه مشخص می‌نماید. محتوای کلمه در جدول صفحات حافظه در آدرس شماره صفحه خوانده شده و در ثبات میانگیر جدول حافظه قرار داده می‌شود. اگر بیت حضور مقدار 1 را داشته




**شکل ۱۸-۱۲** تقسیم فضای آدرس و فضای حافظه به دسته‌های 1K کلمه‌ای

1- Page Frame




**شکل ۱۹-۱۲** جدول حافظه در یک سیستم صفحه‌ای

باشد شماره بلاکی که باین ترتیب خوانده‌می‌شود به دو بیت پرارزشتر ثبات آدرس حافظه اصلی منتقل می‌گردد. شماره سطر از آدرس مجازی به 10 بیت کم ارزشتر ثبات آدرس منتقل می‌شود. یک سیگنال خواندن که به حافظه اصلی متصل است محتوای کلمه را به ثبات میانگیر حافظه منتقل و در آنجا آماده استفاده CPU خواهد بود. اگر بیت حضور در کلمه خوانده شده از جدول صفحه 0 باشد، بمعنی این است که محتوای کلمه‌ای که بوسیله آدرس مجازی به آن اشاره شده است در حافظه اصلی نیست. سپس قبل از ادامه محاسبات، بافراخوانی سیستم عامل صفحه مطلوب از حافظه کمکی برداشت و در حافظه اصلی قرار داده می‌شود.

## جدول صفحات از حافظه تداعیگر

یک جدول صفحات حافظه با دستیابی تصادفی کارآیی لازم را از نظر بهره‌گیری دارا نیست. در مثال شکل ۱۹-۱۲ دیدیم که حداقل به هشت کلمه حافظه نیاز است، که هر یک به یک صفحه تعلق داشتند، ولی همواره چهار کلمه علامت خالی خواهند داشت زیرا حافظه اصلی گنجایش بیش از چهار بلاک را ندارد. بطور کلی، سیستمی با n صفحه و m بلاک یک جدول حافظه - صفحه n مکانی را لازم دارد که در آن حداکثر m بلاک حاوی شماره بلاک و بقیه خالی خواهد بود. بعنوان یک مثال عددی یک فضای آدرس 1024K کلمه در یک فضای حافظه 32K کلمه را در نظر بگیرید. اگرهر صفحه یا بلاک 1K کلمه داشته باشد، تعداد صفحات 1024 و تعداد بلاک‌ها 32 خواهد بود.ظرفیت جدول حافظه - صفحه باید 1024 کلمه و فقط 32 مکان دارای بیت حضور 1 می‌باشند. در هر لحظه حداقل 992 مکان خالی و بلااستفاده است.

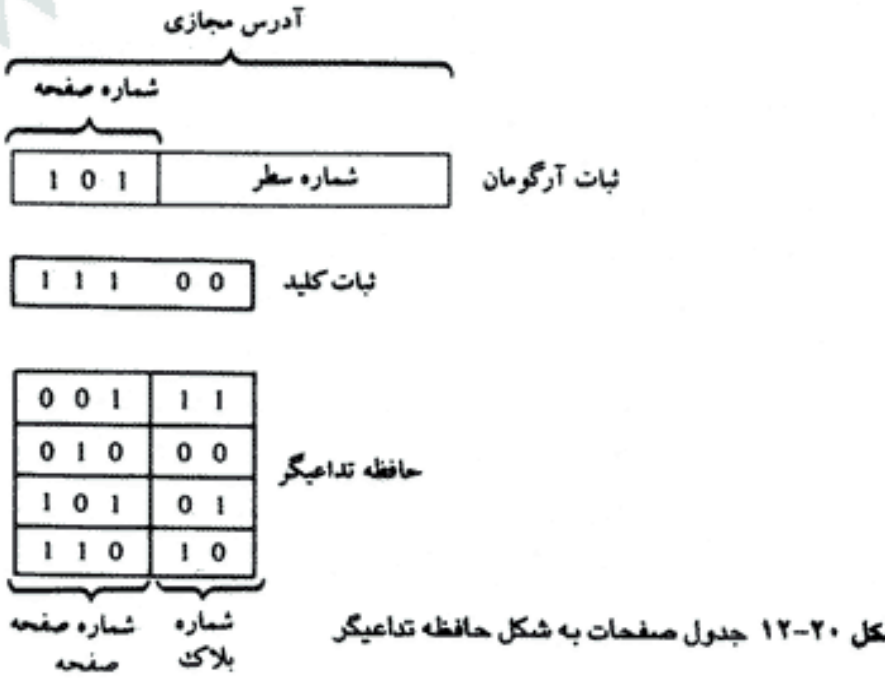
راه کارآتری برای سازمان دهی جدول صفحات این است که آن را با تعدادی از کلمات برابر با بلاک‌ها در حافظه اصلی بسازیم. بدین ترتیب اندازه حافظه کاهش یافته و هر مکان بطور کامل مورد استفاده قرار

می‌گیرد. این روش را می‌توان بکمک حافظه تداعیگر به انجام رساند که هر کلمه در حافظه حاوی یک شماره صفحه به همراه شماره بلاک متناظر آن باشد. میدان صفحه در هر کلمه با شماره صفحه در آدرس مجازی مقایسه می‌شود. اگر انطباق رخ دهد کلمه از حافظه خوانده می‌شود و شماره بلاک آن استخراج می‌گردد.

مجدداً حالت هشت صفحه و چهار بلاک را در نظر بگیرید. همانطورکه در شکل ۱۹-۱۲ ملاحظه شد به جای جدول حافظه صفحه از نوع با دستیابی تصادفی از آرایه حافظه تداعیگر استفاده می‌کنیم که هر یک از وارده‌های حافظه تداعیگر از دو میدان ساخته شده است (شکل ۲۰-۱۲). سه بیت اول میدانی را برای ذخیره شماره صفحه مشخص می‌کند. دو بیت آخر میدانی را برای ذخیره شماره بلاک معین می‌نماید. آدرس مجازی در ثبات آرگومان قرار گرفته است. بیت‌های متعلق به شماره صفحه در ثبات آرگومان با تمام شماره صفحات در میدان صفحه حافظه تداعیگر مقایسه می‌شوند. اگر شماره صفحه پیدا شود، کلمه 5 بیتی از حافظه خوانده می‌شود. شماره بلاک متناظر، که در همان کلمه است، به ثبات آدرس حافظه اصلی منتقل می‌گردد. اگر تطبیق صورت نگیرد، برای آوردن صفحه از حافظه کمکی یک فراخوانی سیستم عامل صورت می‌گیرد.

## جایگزینی صفحات

یک سیستم حافظه مجازی ترکیبی از تکنیک‌های سخت‌افزاری و نرم‌افزاری است. سیستم مدیریت حافظه نرم‌افزاری تمام عملیات نرم‌افزاری را برای بهره‌برداری موثر از فضای حافظه انجام می‌دهد. این سیستم باید در این موارد تصمیم بگیرید: (۱) کدام صفحه از حافظه اصلی باید برای ایجاد فضای لازم برای یک صفحه جدید حذف شود، (۲) چه زمانی باید یک صفحه جدید از حافظه کمکی به حافظه اصلی منتقل شود، و (۳) صفحه مورد نظر باید در چه محلی ازحافظه اصلی قرار داده شود. مکانیزم نگاشت سخت افزاری و نرم افزار مدیریت حافظه همراه با هم معماری حافظه مجازی را تشکیل می‌دهند.




شکل ۲۰-۱۲ جدول صفحات به شکل حافظه تداعیگر

وقتی که برنامه‌ای شروع به اجرا می‌کند، یک یا چند صفحه به حافظه اصلی انتقال یافته و جدول صفحات مقدار می‌گیرد تا محل آنها را نشان دهد. برنامه از حافظه اصلی اجرا می‌شود تا اینکه اقدام به‌ارجاع به صفحه‌ای کند که هنوز در حافظه کمکی قرار دارد. این وضعیت فقدان صفحه[1] نامیده می‌شود. هنگامی که فقدان صفحه رخ دهد اجرای برنامه جاری متوقف می‌شود تا اینکه صفحه لازم به حافظه اصلی منتقل گردد. چون بارکردن یک صفحه از حافظه کمکی به حافظه اصلی اصولاً یک عمل I/O است، سیستم عامل این کار را به پردازنده I/O واگذار می‌کند. در این حال، کنترل به برنامه‌بعدی در حافظه که منتظر پردازش در CPU است منتقل می‌شود. پس از اختصاص بلاک حافظه و اتمام انتقال، برنامه قبلی می‌تواند عملکرد خود را از سربگیرد.

هرگاه، در یک سیستم حافظه فقدان صفحه رخ دهد، بدان معنی است که صفحه ارجاع شده بوسیله CPU درحافظه اصلی نیست. سپس یک صفحه جدید از حافظه کمکی به حافظه اصلی انتقال می‌یابد. اگر حافظه اصلی پرباشد، لازم است یک صفحه از بلاک حافظه حذف گردد تا فضای کافی برای صفحه جدید ایجاد گردد. سیاست انتخاب صفحات برای حذف توسط الگوریتم جایگزینی بکار رفته معین می‌شود. هدف از سیاست جایگزینی تلاش برای حذف صفحه است که کمترین احتمال ارجاع مجدد در آینده نزدیک را دارد.

دو الگوریتم جایگزینی متداول‌تر مورد استفاده عبارتند از اولین ورودی - اولین خروجی (FIFO) و قدیمی‌ترین مورد استفاده (LRU). الگوریتم FIFO صفحه‌ای را که بیشتر از همه در حافظه وجود داشته است برای جایگزینی انتخاب می‌کند. هر بار که صفحه‌ای به حافظه بار شود، شماره شناسایی آن در یک پشته FIFO پوش می‌شود. هر وقت در حافظه هیچ بلاک خالی وجود نداشته باشد FIFO پر است. وقتی که قرار باشد صفحه جدیدی بار شود، قدیمی‌ترین صفحه وارده حذف می‌شود. تعیین صفحه‌ای که قرار است حذف شود آسان است زیرا شماره شناسایی آن در بالای پشته FIFO قرار دارد. سیاست جایگزینی FIFO این مزیت را دارد که‌از نظر پیاده‌سازی ساده است. عیب این روش این است که در شرایط معینی تعداد ورود و خروج صفحات از حافظه اصلی خیلی زیاد است.

سیاست LRU از نظر پیاده سازی مشکل‌تر است ولی بلحاظ مناسب‌تر بودن قدیمی‌ترین صفحه مورد استفاده بعنوان کاندید حذف، نسبت به قدیمی‌ترین صفحه بار شده FIFO، جذاب‌تر می‌باشد. الگوریتم LRU را می‌توان با اختصاص یک شمارنده برای هر صفحه‌ای که در حافظه اصلی قرار دارد پیاده سازی کرد. هر وقت صفحه‌ای مورد ارجاع قرار گیرد، شمارنده مربوطه‌اش صفر می‌شود در فواصل زمانی ثابت شمارنده‌های مربوط به تمام صفحات موجود در حافظه افزایش می‌یابند. قدیمی‌ترین صفحه مورد استفاده صفحه‌ای است که شمارنده‌اش بیشترین مقدار را دارد. این شمارنده‌ها اغلب ثبات‌های سن[2] خوانده می‌شوند، زیرا شمارش، سن آنها را معین می‌کند، یعنی صفحه متناظر با آنها چه مدت قبل مورد ارجاع قرار گرفته است.

---

1- Page Fault 2- Aging Register

## ۱۲-۷ سخت افزار مدیریت حافظه

در یک محیط چند برنامه‌ای که برنامه‌های زیادی در حافظه مقیم است، گاهی لازم می‌شود تا برنامه‌ها و داده‌ها در حافظه به این سو و آن سو جابجا شوند، تا مقدار حافظه مورد استفاده یک برنامه معین تغییر نماید و از برنامه های دیگر توسط یک برنامه جلوگیری گردد. مقتضیات ناشی از چند برنامگی در رابطه با حافظه کامپیوتر نیاز به یک سیستم مدیریت را بوجود آورده است. سیستم مدیریت حافظه مجموعه‌ای از رویه‌های سخت‌افزاری و نرم‌افزاری برای مدیریت انواع برنامه‌های مقیم در حافظه است. نرم‌افزار مدیریت حافظه بخشی از سیستم عامل کلی موجود در اکثر کامپیوترهاست. در اینجا ما به واحد سخت‌افزاری مربوط به سیستم مدیریت حافظه می‌پردازیم.

اجزای اصلی یک واحد مدیریت حافظه عبارتند از:

۱- امکاناتی برای جابجایی دینامیک در حافظه که ارجاع‌های منطقی حافظه را به آدرسهای فیزیکی حافظه تبدیل کند.

۲- امکان برای استفاده مشترک کاربران مختلف از برنامه‌های ذخیره شده در حافظه

۳- حفاظت اطلاعات در مقابل دستیابی غیرمجاز بین کاربران و جلوگیری از تغییر توابع سیستم عامل توسط کاربران

سخت افزار جابجایی دینامیک در حافظه یک روند تبدیل مشابه به سیستم صفحه‌بندی شرح داده شده در بخش ۶-۱۲ است. ثابت بودن اندازه صفحه مورد استفاده در سیستم حافظه مجازی موجب اشکالات معینی در رابطه با اندازه برنامه و ساختار منطقی برنامه‌ها می‌شود.

بهتر است تا برنامه‌ها را به بخش‌های منطقی بنام قطعه[1] تقسیم می‌کرد. یک قطعه مجموعه‌ای از دستورات منطقی مرتبط با هم یا عناصر تحت یک نام خاص است. قطعات ممکن است بوسیله برنامه نویس یا سیستم عامل تولید شود. مثال‌هایی از قطعات عبارتند از، زیرروال‌ها، آرایه‌ای از داده‌ها، جدول سمبل‌ها، یا برنامه یک کاربر.

استفاده مشترک از برنامه‌ها بخش لاینفکی از یک سیستم چند برنامه‌ای است. مثلاً چندین کاربر که مایلند برنامه‌های فرترن خود را ترجمه[2] کنند، باید بتوانند از یک نسخه واحد از برنامه مترجم استفاده کنند نه اینکه هر کدام نسخه جداگانه‌ای از آن را در حافظه داشته باشند. در یک سیستم چند برنامه‌ای، همه کاربران از دیگر برنامه‌های سیستم موجود در حافظه نیز بطور مشترک استفاده می‌کنند و لازم نیست نسخه‌های متعددی از آنها ایجاد شود.

سومین مسئله در سیستم چند برنامه‌ای حفاظت از یک برنامه از تداخل دیگر برنامه‌هاست. مثالی از اینگونه دخالت‌های ناخواسته اینست که یک کاربر بطور غیرمجاز برنامه یک کاربر دیگر را کپی کند. جنبه دیگری از حفاظت، جلوگیری از انجام عملیات سیستم عامل و لذا ایجاد وقفه در ترتیب منظم عملیات

---

1- Segment 2- Compile

در یک سیستم کامپیوتری توسط کاربران عادی است. برنامه‌های معینی که جنبه سری دارند باید از دسترسی پرسنل غیرمجاز حفاظت شوند تا از سوء استفاده در فعالیت‌های محرمانه یک سازمان جلوگیری شود.

آدرس تولید شده بوسیله یک برنامه قطعه‌ای آدرس منطقی خوانده می‌شود. این آدرس مشابه آدرس مجازی است جزاینکه فضای آدرس منطقی در عوض تقسیم به صفحات با طول ثابت، به قطعه‌هایی با طول متغیر تقسیم می‌گردد. در اینجا نیز همانند سیستم حافظه مجازی، ممکن است آدرس منطقی بزرگتر از آدرس حافظه فیزیکی باشد. هر قطعه علاوه بر اطلاعات مربوط به تغییر مکان، اطلاعات حفاظتی را نیز همراه خود دارد. برنامه‌های مشترک، در یک قطعه منحصر بفرد در فضای آدرس منطقی هر کاربر قرار داده می‌شود، تا بتوان از یک نسخه فیزیکی واحد بطور مشترک استفاده کرد. وظیفه واحد مدیریت حافظه تبدیل آدرس‌های منطقی به آدرس های فیزیکی مانند تبدیل در حافظه مجازی است.

## نگاشت قطعه - صفحه‌ای

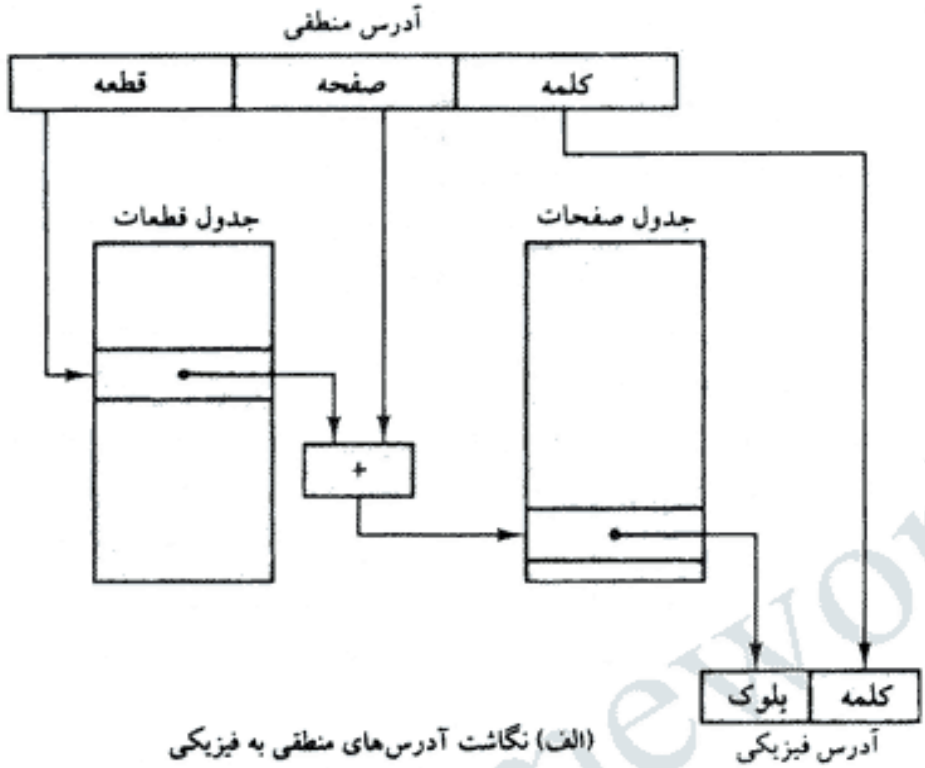
قبلاً گفته شده که خاصیت فضای منطقی این است که از قطعات با طول متغیر استفاده می‌کند. طول هر قطعه می‌تواند براساس نیازهای برنامه جاری افزایش یا کاهش یابد. یکی از راه‌های مشخص کردن طول هر قطعه، متناظر ساختن آن با تعدادی از صفحات هم‌اندازه است. برای آگاهی از چگونگی انجام آن، آدرس منطقی شکل ۲۱-۱۲ را در نظر بگیرید. آدرس منطقی به سه میدان تقسیم شده است. میدان قطعه یک شماره قطعه را مشخص می‌کند. میدان صفحه، آن را در داخل قطعه و میدان کلمه نیزکلمه را در صفحه مشخص می‌کند. یک میدان صفحه با k بیت قادر است تا $2^k$ صفحه را معین کند. یک شماره قطعه می‌تواند، فقط یک صفحه یا تا $2^k$ صفحه متناظر باشد. بنابراین طول یک قطعه برطبق تعداد صفحاتی که به آن اختصاص داده می‌شود متغیر است.

تبدیل آدرس منطقی به آدرس فیزیکی بوسیله دو جدول انجام می‌شود، شکل ۲۱-۱۲ (الف). شماره قطعه آدرس منطقی، آدرس جدول قطعه را مشخص می‌کند. واردها در جدول قطعه، اشاره‌گر[1] آدرس برای پایه[2] (شروع) جدول صفحه می‌باشد. پایه جدول صفحه به شماره صفحه که در آدرس منطقی داده می‌شود اضافه می‌شود. مجموع دو کمیت اشاره‌گر آدرسی را برای یک وارده در جدول صفحه تولید می‌کند. مقداری که از جدول صفحات استخراج می‌شود شماره بلاک حافظه فیزیکی را مشخص می‌نماید. باکنار هم قراردادن میدان بلاک و میدان کلمه، آدرس تبدیل شده فیزیکی نهایی بدست می‌آید.

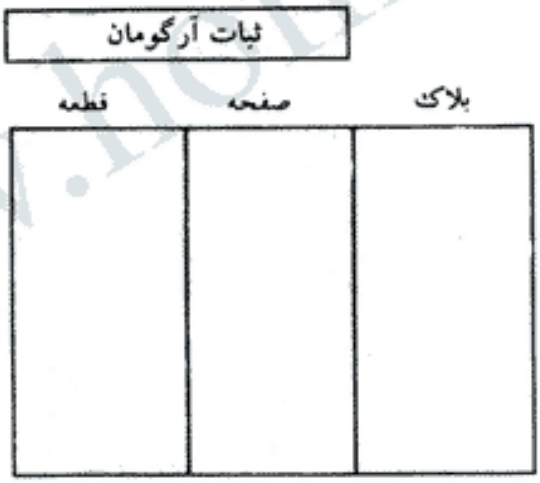
دو جدول تبدیل ممکن است در دو حافظه جدا از هم کوچک یا در حافظه اصلی ذخیره شود. در هر صورت، یک ارجاع حافظه بوسیله CPU سه دستیابی به حافظه را نیاز دارد: یکی به جدول قطعات، یکی به جدول صفحه و بار سوم به حافظه اصلی. این امر سیستم را در مقایسه با سیستم معمولی که فقط یک ارجاع به حافظه را نیاز دارد بسیار کند می‌نماید. برای اجتناب از این کاهش سرعت، یک حافظه تداعیگر سریع ، برای نگهداری آخرین وارده‌های ارجاع شده جدول بکار می‌رود. (این نوع حافظه گاهی بافر دم دستی ترجمه یا TLB خوانده می‌شود.) اولین باری که یک بلاک ارجاع داده شود، مقدار آن همراه با

---

1- Pointer 2- Base




(الف) نگاشت آدرس‌های منطقی به فیزیکی




(ب) بافر دم‌دستی ترجمه (TLB) در حافظه تداعیگر

**شکل ۲۱-۱۲** نگاشت در واحد مدیریت حافظه قطعه- صفحه‌ای

شماره‌های قطعه و صفحه طبق شکل ۲۱-۱۲ (ب) وارد یک حافظه تداعیگر می‌شود. بنابراین روند تبدیل ابتدا با جستجوی تداعیگر با شماره‌های قطعه و صفحه آزمایش می‌شود. اگر نتیجه موفقیت آمیز باشد، تأخیر تبدیل تنها متعلق به حافظه تداعیگر است. اگر انطباقی رخ ندهد، جدول تبدیل شکل ۲۱-۱۲ (الف) که آهسته تر است بکار رفته و نتیجه تبدیل بداخل حافظه تداعیگر برای کاربرهای آینده انتقال می‌یابند.

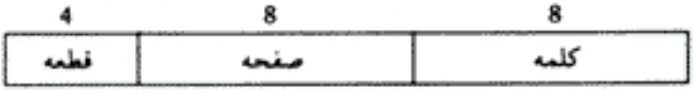
## مثال عددی

یک مثال عددی می‌تواند عملکرد یک واحد مدیریت حافظه را روشن‌تر کند. آدرس منطقی 20 بیتی شکل ۲۲-۱۲ (الف) را در نظر بگیرید. شماره قطعه 4 بیتی یکی از شانزده قطعه موجود را مشخص می‌کند. شماره صفحه 8 بیتی می‌تواند تا 256 صفحه را مشخص می‌نماید، و میدان کلمه هشت بیتی دلالت بر وجود صفحات با ظرفیت 256 کلمه دارد. این پیکربندی اجازه می‌دهد که هر قطعه هر تعداد صفحه تا 256 صفحه داشته باشد. کوچکترین قطعه ممکن یک صفحه یا 256 کلمه خواهد داشت. بزرگترین قطعه 256 صفحه و در کل 64K = 256×256 کلمه خواهد داشت.

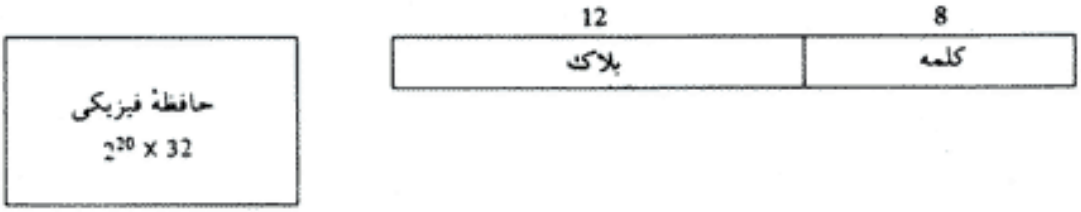
حافظه فیزیکی نشان داده شده در شکل ۲۲-۱۲ (ب) از $2^{20}$ کلمه 32 بیتی تشکیل شده‌است. آدرس 20 بیتی به دو میدان تقسیم شده است: یک شماره بلاک 12 بیتی و یک شماره کلمه 8 بیتی . بنابراین، آدرس فیزیکی به 4096 بلاک 256 بیتی تقسیم شده است. یک صفحه در آدرس منطقی دارای بلاک متناظری در حافظه فیزیکی می‌باشد. توجه کنید که آدرس‌های منطقی و فیزیکی 20 بیتی هستند. در غیاب یک واحد مدیریت حافظه، آدرس 20 بیتی از CPU می‌تواند مستقیماً برای دستیابی به حافظه بکار رود.

برنامه‌ای را در نظر بگیرید که برای بار شدن در حافظه به 5 صفحه نیاز دارد. همانطور که در شکل ۲۳-۱۲ (الف) دیده می‌شود سیستم عامل می‌تواند به این برنامه قطعه 6 و صفحات 0 تا 4 را اختصاص دهد. محدوده کل آدرس منطقی برای برنامه از شانزده شانزدهی 60000 تا 604FF است. وقتی که برنامه در حافظه فیزیکی بار شود در میان پنج بلاک حافظه فیزیکی که توسط سیستم عامل خالی تشخیص داده شود توزیع می‌گردد. سپس واسطه‌های بین هر بلاک حافظه و شماره صفحه منطقی طبق شکل ۲۳-۱۲ (ب) وارد جدول می‌شوند. اطلاعات حاصل از این جدول مطابق شکل ۲۴-۱۲ (الف) وارد جداول قطعات و صفحات می‌گردد.

اکنون آدرس منطقی خاص نشان داده شده در شکل ۲۴-۱۲ را در نظر بگیرید. آدرس 20 بیتی

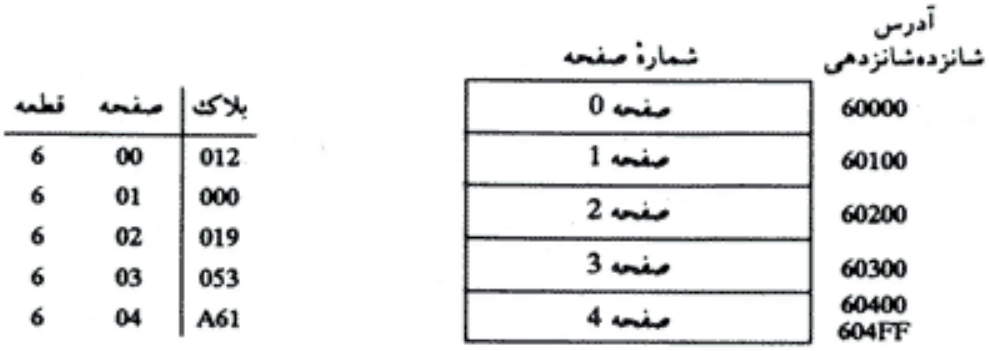



(الف) قالب آدرس منطقی: 16 قطعه 256 صفحه‌ای که در هر صفحه 256 کلمه قرار دارد




(ب) قالب آدرس فیزیکی: 4096 بلاک 256 کلمه‌ای که هر کلمه 32 بیت دارد

**شکل ۲۲-۱۲** مثالی از آدرس‌های منطقی و فیزیکی




(الف) تخصیص آدرس‌های حافظه منطقی (ب) بلاک‌های حافظه متناظر با قطعه - صفحه‌ها

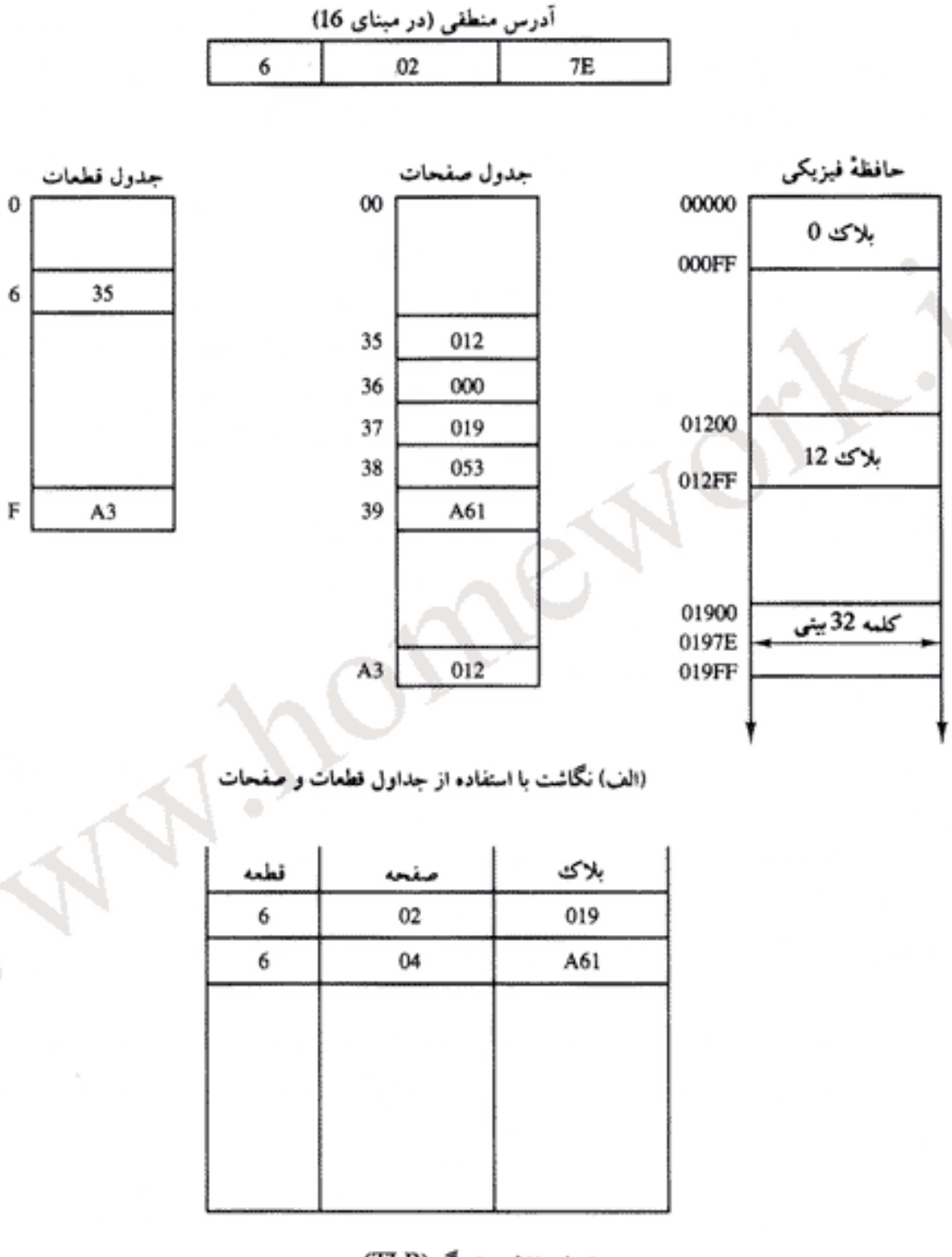
**شکل ۲۳-۱۲ مثالی از تخصیص آدرسهای حافظه منطقی و فیزیکی**

بصورت یک عدد پنج رقمی شانزده شانزدهی نشان داده شده است. این آدرس به شماره کلمه 7E از صفحه 2 در قطعه 6 اشاره می‌کند. شروع قطعه 6 در جدول صفحه در آدرس 35 است. قطعه 6 پنج صفحه دارد، که در جدول صفحات در آدرسهای 35 تا 39 نشان داده شده‌اند. صفحه 2 از قطعه 6 در آدرس 37=2+35 قرار دارد. ازجدول صفحات دیده می‌شود که بلاک حافظه فیزیکی در 019 است. کلمه 7E در بلاک 19 آدرس فیزیکی 20 بیتی 0197E را بدست می‌دهد. توجه کنید که صفحه 0 از قطعه 6 به بلاک 12 و صفحه 1 به بلاک 0 نگاشت می‌شود. حافظه تداعیگر شکل ۲۴-۱۲(ب) نشان می‌دهد که صفحات 2 و 4 از قطعه 6 قبلاً مورد ارجاع قرار گرفته‌اند و بنابراین شماره بلاک متناظر آنها در حافظه تداعیگر ذخیره شده است.

با توجه به مثال فوق آشکار است که سیستم مدیریت حافظه قادر است هر تعداد صفحات را به هر قطعه اختصاص دهد. هر صفحه منطقی را می‌توان به هریک از بلاک‌های حافظه فیزیکی نگاشت کرد. صفحات را می‌توان بسته به نیاز فضای لازم برای حافظه، به بلاک‌های مختلفی در حافظه جابجا کرد. تنها بهنگام‌سازی موردنیاز تغییر شماره بلاک در جدول صفحه است. صفحات می‌توانند بدون تأثیر بر یکدیگر کوچک یا بزرگ شوند. اگر قرار باشد برنامه مشترکی بوسیله چند کاربر مورد استفاده قرار گیرد قطعات مختلف می‌توانند از یک بلاک استفاده کنند. مثلاً به بلاک شماره 12 در حافظه فیزیکی آدرس منطقی دومی نیز می‌توان از F0000 تا F00FF اختصاص داد. این محدوده شماره قطعه 15 و شماره صفحه 0 را داراست، و همانطور که در شکل ۲۴-۱۲ (الف) دیده می‌شود به بلاک 12 نگاشت می‌شود.

### حفاظت حافظه

حفاظت حافظه می‌تواند از طریق آدرس فیزیکی یا آدرس منطقی انجام شود. حفاظت حافظه از طریق آدرس فیزیکی بدین ترتیب صورت می‌گیرد که به هر بلاک در حافظه تعدادی بیت حفاظت که معین‌کننده نوع دستیابی به بلاک مربوطه‌اش می‌باشد اختصاص می‌یابد. هر بار صفحه‌ای از یک بلاک به بلاک دیگر جابجا شود لازم است بیت حفاظت بهنگام شود. اعمال حفاظت درفضای آدرس منطقی بهتر از فضای




(الف) نگاشت با استفاده از جداول قطعات و صفحات

| قطعه | صفحه | بلاک |
|---|---|---|
| 6 | 02 | 019 |
| 6 | 04 | A61 |
| | | |

(ب) حافظه تداعیگر (TLB)

**شکل ۲۴-۱۲** مثال نگاشت حافظه منطقی به فیزیکی (همه عددها در مبنای شانزده)

آدرس فیزیکی است. این کار را می‌توان با گنجاندن اطلاعات حفاظتی در جدول قطعات یا ثبات قطعه سخت‌افزار مدیریت حافظه انجام داد.

محتوای هر وارده در جدول قطعه یا یک ثبات قطعه توصیفگر[1] خوانده می‌شود. یک توصیفگر نوعی علاوه بر میدان آدرس پایه دارای یک یا دو میدان اضافی برای اهداف حفاظتی است. نوعی قالب‌برای توصیفگر قطعه در شکل ۲۵-۱۲ نشان داده شده است. آدرس پایه آدرس شروع جدول صفحه را در سازمان قطعه - صفحه و یا در آدرس پایه بلاک در سازمان ثبات قطعه فراهم می‌سازد. این آدرس در تبدیل یک آدرس منطقی به فیزیکی بسیار موثر است.

میدان طول[2]، اندازه قطعه را با مشخص کردن حداکثر تعداد صفحات قابل اختصاص به قطعه معین می‌کند. اگر شماره صفحه خارج از مرز قطعه باشد. تخطی از اندازه صورت گرفته است. به این ترتیب یک برنامه معین و داده‌های آن نمی‌توانند به حافظه‌ای که سیستم عامل به آن اختصاص نداده است دست یابد.

میدان حفاظت در توصیفگر قطعه، حقوق دستیابی مجاز به قطعه موردنظر را مشخص می‌کند. در سازمان قطعه - صفحه، هر یک از وارده‌ها ممکن است میدان حفاظت خود را برای توصیف حقوق دستیابی هر صفحه داشته باشد. اطلاعات حفاظت توسط برنامه کنترل اصلی سیستم عامل در توصیفگر قرار می‌گیرد. بعضی از حقوق دستیابی مورد نظر مقیم در حافظه برای حفاظت برنامه‌ها عبارتند از:

۱- حقوق کامل خواندن و نوشتن
۲- فقط خواندن (حفاظت در برابر نوشتن)
۳- فقط اجرا (حفاظت از برنامه ها)
۴- فقط سیستم (حفاظت از سیستم عامل)

حقوق کامل خواندن و نوشتن هنگامی که یک برنامه در حال اجرای دستورالعل های خودش باشد به آن داده می شود. حفاظت در برابر نوشتن برای استفاده مشترک از برنامه های سیستم از قبیل برنامه‌های ابزاری[3] و روال های کتابخانه‌ای[4] دیگر مفید است. این برنامه های سیستم در ناحیه‌ای از حافظه ذخیره می‌شوند تا بوسیله کار بران متعدد مورد استفاده مشترک قرار گیرند. این برنامه‌ها را سایر برنامه‌ها می‌توانند بخوانند، اما نوشتن در آنها مجاز نیست. این خصوصیت آنها را از هر گونه تغییر توسط برنامه‌های دیگر حفاظت می‌کند.

شرط فقط - اجرا برنامه‌ها را از کپی شدن محافظت می‌نماید. این شرط قطعه را محدود می‌کند تا فقط در حین فاز برداشت مورد ارجاع قرار گیرد و نه در حین فاز اجرا. بنابراین به کاربران اجازه می‌دهد

| حفاظت | طول | آدرس مبنا |
|---|---|---|

شکل ۲۵-۱۲ قالب نوعی توصیف‌گر قطعه

1- Descriptor
2- Access Leights
3- Utilitys Programs
4- Libray Routine

تادستورالعمل‌های برنامه در قطعه را اجرا کند اما از خواندن دستورالعمل‌ها بعنوان داده به قصد کپی کردن آنها ممانعت می‌کند.

در هر لحظه از زمان بخش‌هایی از سیستم عامل در حافظه واقع است. این برنامه‌های سیستم با قرار نگرفتن در دسترس کاربران غیرمجاز محافظت شود. شرط حفاظت سیستم عامل برای تمام برنامه‌های سیستم عامل در توصیفگر قرار می‌گیرد تا مانع دستیابی کاربران موقت به قطعات سیستم عامل گردد.

## مسائل

۱-۱۲ الف) چند تراشه RAM، 8×128 برای تهیه یک حافظه با ظرفیت 2048 لازم است؟

ب) چند خط گذرگاه آدرس باید بکاربرد تا 2048 بایت از حافظه دستیابی شود؟ چند خط از این خطوط برای همه تراشه‌ها مشترک است.

پ) چند خط برای انتخاب تراشه باید دیکدر شود؟ اندازه دیکدر را مشخص کنید.

۲-۱۲ کامپیوتری از یک RAM با ظرفیت 1×1024 استفاده می‌کند.

الف) چند تراشه لازم است، و چگونه خطوط آنها باید متصل شود تا 1024 بایت حافظه حاصل شود؟

ب)چند تراشه برای تهیه 16K بایت حافظه لازم است؟ شفاهاً توضیح دهید که چگونه تراشه‌ها باید به گذرگاه حافظه وصل شوند.

۳-۱۲ یک تراشه ROM با 8×1024 بیت دارای چهار ورودی انتخاب بوده و با منبع تغذیه 5 ولتی کار می‌کند. برای بسته IC چند پایه لازم است. بلاک دیاگرام را رسم کرده و پایانه‌های ورودی و خروجی را در ROM نام گذاری کنید.

۴-۱۲ سیستم حافظه شکل ۴-۱۲ را به 4096 بایت RAM و 4096 بایت ROM گسترش دهید. نقشه حافظه - آدرس را رسم کنید. مشخص کنید که اندازه دیکدر مورد استفاده باید چقدر باشد.

۵-۱۲ کامپیوتری از تراشه‌های RAM با 8×256 و ROM با 8×1024 استفاده می‌کند. سیستم کامپیوتر به 2K بایت از RAM، 4K بایت از ROM، و چهار واحد واسطه، که هر یک چهار ثبات دارد، نیاز دارد. از پیکربندی I/O نگاشت حافظه استفاده شده است. به دو بیت باارزشتر گذرگاه آدرس 00 برای RAM، 01 برای ROM و 10 برای ثبات‌های واسطه اختصاص داده می‌شود.

الف) چند تراشه RAM و ROM لازم است

ب) نقشه حافظه - آدرس را برای سیستم رسم کنید.

پ) محدوده آدرس را برحسب شانزده شانزدهی برای RAM، ROM و مدار واسطه معین کنید.

۶-۱۲ یک کامپیوتر دارای گذرگاه آدرس 16 بیتی است. 15 خط اول آدرس‌ها برای انتخاب یک بانک 32K بایتی حافظه استفاده شده است. بیت با ارزشتر آدرس برای انتخاب یک ثبات که محتوای گذرگاه داده

را دریافت می‌کند بکار رفته است. توضیح دهید چگونه می‌توان از این پیکربندی استفاده کرده و ظرفیت حافظه را به هشت بانک 32K بایتی افزایش داد تا یک بانک 256K بایتی حاصل شود.

۱۲-۷ یک سیستم دیسک مغناطیسی دارای پارامترهای زیر است،

$T_s$ = زمان متوسط لازم برای قرار گرفتن هد مغناطیسی روی یک شیار

R = سرعت چرخش دیسک برحسب دور برثانیه

$N_t$ = تعداد بیت‌ها در شیار

$N_s$ = تعداد بیت‌ها در قطاع

زمان متوسط $T_a$ که برای خواندن یک قطاع لازم است چقدر می‌باشد.

۱۲-۸ سرعت انتقال یک نوار مغناطیسی هشت شیاره که سرعتش 120 اینچ در ثانیه و تراکم آن 1600 بیت در اینچ است چقدر باشد.

۱۲-۹ تابع متمم منطق انطباق یک کلمه را در حافظه تداعیگر بدست آورید. به بیان دیگر، نشان دهید که $M'_i$ مجموع توابع OR انحصاری است. دیاگرام منطقی را برای $M'_i$ رسم نموده و با یک معکوس‌کننده آنرا پایان دهید تا $M_i$ بدست آید.

۱۲-۱۰ تابع بول را برای منطق انطباق یک کلمه را با یک حافظه تداعیگر با در نظر گرفتن یک بیت نشانه که مشخص کند کلمه فعال است یا غیرفعال ، بدست آورید.

۱۲-۱۱ چه مدار منطقی اضافی لازم است تا نتیجهٔ عدم تطابق را برای یک کلمه در حافظه تداعیگر وقتی که همه بیت‌های کلید صفر است مشخص کند.

۱۲-۱۲ الف) دیاگرام منطقی تمام سلول‌های یک کلمه در حافظه تداعیگر را رسم کنید. منطق خواندن و نوشتن شکل ۸-۱۲ و منطق انطباق شکل ۹-۱۲ را نیز اضافه کنید.

ب) دیاگرام منطقی تمام سلول‌های عمودی یک ستون (ستون J) را در حافظه تداعیگر رسم کنید. یک خط خروجی مشترک را برای همه بیت‌هایی که در یک ستون قرار دارند اضافه کنید.

ج) با استفاده از دیاگرام‌های (الف) و (ب) نشان دهید که اگر خروجی $M_i$ به خط خواندن همان کلمه وصل شود، کلمه منطبق به خارج خوانده خواهد شد، مشروط بر اینکه فقط یک کلمه با آرگومان پوشانده شده مطابقت داشته باشد.

۱۲-۱۳ با استفاده از بلاک دیاگرام شرح دهید چگونه می‌توان کلمات منطبق متعدد را از حافظه تداعیگر به خارج خواند.

۱۲-۱۴ منطق یک سلول و نیزیک کلمه کامل را برای یک حافظه تداعیگر، همراه با نشانگری برای مواقعی که آرگومان پوشش نیافته بزرگتر از (اما نه مساوی با) کلمه موجود در حافظه تداعیگر باشد، بدست آورید.

۱۲-۱۵ یک حافظه کش تداعیگر دوتایی از بلاک‌های چهار کلمه‌ای استفاده می‌کند. حافظه کش می‌تواند مجموعاً 2048 کلمه از حافظه اصلی را در خود جای دهد. اندازه حافظه اصلی 128x32 است.

الف) همه اطلاعات لازم برای ساختن این حافظه کش را فرموله کنید.

ب) اندازه حافظه کش چقدر است.

۱۲-۱۶ زمان دستیابی یک حافظه کش 100ns و حافظه اصلی 1000ns است. پیش بینی می‌شود 80 درصد تقاضاهای حافظه برای خواندن و 20 درصد بقیه برای نوشتن می‌باشند. نسبت برد برای دستیابی‌های خواندن فقط 0.9 است . رویه کامل نویسی استفاده می‌شود.

الف) زمان دستیابی متوسط سیستم فقط با درنظر گرفتن سیکل خواندن چقدر است.

ب) زمان دستیابی متوسط سیستم برای تقاضای خواندن و نوشتن چقدر است.

ج) نسبت برد با در نظر گرفتن سیکل‌های نوشتن چقدر است.

۱۲-۱۷ یک مجموعه حافظه کش تداعیگر چهارتایی در هر مجموعه چهار کلمه دارد. یک رویه جایگزینی مبتنی بر الگوریتم بر قدیمی‌ترین مورد استفاده (LRU) باشمارنده‌های دوبیتی متناظر با هر یک از کلمات مجموعه پیاده‌سازی می‌شود. بنابراین مقداری در محدوده 0 تا 3 برای هر کلمه ثبت می‌شود. وقتی بردی رخ دهد، شمارنده متناظر با کلمه مورد ارجاع 0 می‌گردد. شمارنده‌هایی که مقادیر قبلی آنها کمتر از مقدار شمارنده مورد ارجاع است 1 واحد افزایش می‌یابند و بقیه تغییر نمی‌کنند. اگر باخت رخ دهد، کلمه‌ای که مقدار شمارنده آن 3 است حذف می‌شود، کلمه جدید در محل آن قرار داده می‌شود، و شمارنده آن 0 می‌گردد. سه شمارنده دیگر 1 واحد افزایش می‌یابند. نشان دهید که این رویه برای دنباله ارجاعات زیر درست عمل می‌کنند:A، B، C، D، B، E، D، A، C، E، C، E (با A، B، C و D بعنوان مقادیر اولیه شروع کنید که در آنها A قدیمی‌ترین مورد استفاده است).

۱۲-۱۸ یک کامپیوتر دیجیتال دارای واحد حافظه 64K×16 و یک حافظه کش 1K کلمه‌ای است. کش از نگاشت مستقیم استفاده می‌کند و سایز بلاک چهار کلمه است.

الف) در میدان‌های نشانه، شاخص (اندیس)، بلاک، و کلمه قالب آدرس چند بیت وجود دارد؟

ب) در هر کلمه کش چند بیت وجود دارد، و آنها چگونه به توابع مختلف کاری تقسیم شده‌اند.

ج) حافظه کش چند بلاک را در خود جای می‌دهد.

۱۲-۱۹ یک فضای آدرس با 24 بیت و فضای حافظه متناظر آن با 16 بیت مشخص می‌شود

الف) چند کلمه در فضای آدرس وجود دارد. ب) چند کلمه در فضای حافظه وجود دارد.

ج) اگر یک صفحه از 2K کلمه ساخته شده باشد، چند صفحه و بلاک در سیستم وجود دارد.

۱۲-۲۰ یک حافظه مجازی دارای یک صفحه 1K کلمه‌ای است. در این حافظه هشت صفحه و چهار بلاک وجود دارد. جدول صفحات حافظه تداعیگر حاوی وارده‌های زیر است:

| صفحه | بلاک |
|---|---|
| 0 | 3 |
| 1 | 1 |
| 4 | 2 |
| 6 | 0 |

لیستی از آدرس‌های مجازی (به دهدهی) تهیه کنید که در صورت استفاده CPU از آن موجب فقدان صفحه شوند.

۱۲-۲۱ یک سیستم حافظه مجازی دارای فضای آدرس 8K کلمه یک فضای حافظه 4K کلمه، و صفحات و بلاک‌های 1K کلمه است (شکل ۱۸-۱۲). تغییرات ارجاع صفحه زیر در طول یک فاصله زمانی معین رخ می‌دهد. (فقط تغییرات صفحات نوشته شده است. اگر یک صفحه مجدداً مورد ارجاع قرار گرفته باشد دوبار نوشته نشده است.)

4 2 0 1 2 6 1 4 0 1 0 2 3 5 7

چهار صفحه مقیم در حافظه اصلی را پس از هر تغییر ارجاع صفحات تعیین کنید بشرطی که الگوریتم جایگزینی مورد استفاده (الف) FIFO، (ب) LRU باشد.

۱۲-۲۲ دو آدرس منطقی را از شکل ۱۲-۲۲ (الف) که سبب دستیابی به حافظه فیزیکی در آدرس فیزیکی شانزده شانزدهی 012AF می‌گردد معین کنید.

۱۲-۲۳ فضای آدرس منطقی در یک سیستم کامپیوتری متشکل از 128 قطعه است. هر قطعه می‌تواند تا 32 صفحه 4K کلمه‌ای داشته باشد. حافظه فیزیکی از 4K بلاک 4K کلمه در هر کدام است. قالب آدرس‌های فیزیکی و منطقی را تنظیم کنید.

۱۲-۲۴ معادل دودویی آدرس منطقی تنظیم شده در مسئله ۱۲-۲۳ را برای قطعه 36 و کلمه 2000 در صفحه 15 بنویسید.

# ۱۳ چند پردازنده‌ها

۱۳-۱ مشخصات چند پردازنده‌ها
۱۳-۲ ساختارهای اتصالات متقابل
۱۳-۳ داوری بین پردازنده‌ها
۱۳-۴ ارتباط و همگامی بین پردازنده‌ها
۱۳-۵ همبستگی حافظه کش

## ۱۳-۱ مشخصات چند پردازنده‌ها

یک سیستم چند پردازنده‌ای از اتصال دو یا چند CPU به همراه حافظه و تجهیزات ورودی - خروجی تشکیل می‌شود. اصطلاح "پردازنده" در چند پردازنده[1] می‌تواند بمعنی یک واحد پردازش و یا یک پردازنده ورودی - خروجی (IOP) باشد. با این وجود، سیستمی با یک CPU و چند IOP در تعریف سیستم چند پردازنده نمی‌گنجد مگر اینکه IOP دارای تسهیلات محاسباتی قابل مقایسه با CPU باشد. با توجه به متداولترین تعریف، مفهوم یک سیستم چند پردازنده دلالت بر وجود چند CPU دارد، گرچه معمولاً یک یا چند IOP هم ممکن است موجود باشد. همان طورکه در بخش ۱-۹ ذکر شد، چند پردازنده‌ها بعنوان سیستم‌هایی با رشته‌های چند دستوری[2] و رشته‌های چند داده‌ای[3] دسته‌بندی می‌شوند.

بین سیستم‌های چند پردازنده و چند کامپیوتری شباهت‌هایی وجود دارد زیرا هر دو آنها عملیات همزمان را پشتیبانی می‌کنند. معهذا اختلاف مهمی بین سیستمی با چند کامپیوتر و سیستمی با چند پردازنده وجود دارد. کامپیوترها با اتصالات بین خود بهم متصل شده و یک شبکه کامپیوتری را تشکیل می‌دهند. شبکه از چندین کامپیوتر مستقل تشکیل می‌شود که می‌توانند با هم ارتباط داشته یا نداشته باشند. یک سیستم چند پردازنده توسط یک سیستم عامل کنترل می‌شود تا اتصالات بین پردازشگرها برقرار شده و در نتیجه تمام اجزاء سیستم در حل یک مسئله همکاری نمایند.

گرچه بعضی کامپیوترهای بزرگ دو یا چند CPU را در سیستم کلی خود دارا هستند، انگیزه عمده

---

1- Multiprocessor 2- Multiple Instruction 3- Multiple Data

برای سیستم‌های چند پردازنده، ظهور ریزپردازنده بوده است. این واقعیت که ریز پردازنده فضای فیزیکی کوچکی را اشغال کرده و بسیار ارزان است، اتصال تعداد زیادی از ریز پردازنده ها و تشکیل یک سیستم مرکب را امکان پذیر می‌سازد. تکنولوژی مدارهای مجتمع با مقیاس بزرگ قیمت اجزاء کامپیوتر را به سطح پایینی برده است که مفهوم استفاده از چند پردازنده برای دستیابی به نیازمندی‌های عملکردی یک سیستم، تبدیل به امکان طراحی جذابی شده است.

چند پردازشی قابلیت اطمینان سیستم را طوری بالا می‌برد که اشکال یا خطایی در یک بخش اثر محدودی بر بقیه سیستم دارد. اگر اشکالی موجب از کار افتادگی یک پردازنده شود، می‌توان پردازنده دیگری را برای انجام اعمال پردازنده غیرفعال اختصاص داد. سیستم در کل، می‌تواند بطور صحیح، با احتمالاً مقداری افت بازدهی، بکار خود ادامه دهد.

مزیت حاصل از یک سازمان چند پردازنده، ارتقای عملکرد سیستم است. عملکرد بالای سیستم ناشی از حقیقت است که محاسبات می‌تواند بطور موازی به یکی از دو روش زیر انجام شود.

۱- چند عمل[1] مستقل می‌توانند بطور موازی انجام شوند.

۲- یک کار مستقل می‌تواند به چند کار کوچکتر[2] موازی تقسیم شود.

یک کار[3] کلی را می‌توان به تعدادی کار کوچکتر (یا تکلیف) تقسیم کرد بطوری که هرپردازنده بتواند مستقلاً یکی از آنها را انجام دهد. کارهای کوچک سیستم را می‌توان به پردازنده‌های کاربرد ویژه‌ای[4] محول کرد که طراحی آنها برای اجرای انواع معینی از پردازش بهینه‌سازی شده است. به عنوان مثال می‌توان سیستمی را در نظر گرفت که در آن یک پردازنده محاسبات مربوط به کنترل یک فرآیند صنعتی را انجام می‌دهد در حالی که پردازنده دیگری پارامترهای مختلف از قبیل دما، سرعت جریان سیال را نظارت و کنترل می‌کند. مثال دیگر کامپیوتری است که در آن یک پردازنده محاسبات ریاضی ممیز - شناور سرعت‌های بالا را انجام می‌دهد و کامپیوتر دیگری بر کارهای کوچک یک داده‌پردازی نظارت می‌کند.

چند پردازنده‌ای می‌تواند با تفکیک یک برنامه به چند کار کوچک قابل اجرا بصورت موازی، عملکرد را بهبود ببخشد. این امر به یکی از دو روش زیر امکان‌پذیر است. کاربر می‌تواند بطور صحیح تصمیم بگیرد تا کارهای کوچک معینی از برنامه بطور موازی اجرا شوند. این کار باید قبل از بار کردن برنامه با مشخص کردن قطعه‌های قابل اجرای موازی انجام شود. اکثر سازندگان چند پردازنده‌ها سیستم عاملی با ساخت‌هایی در زبانهای برنامه‌نویسی که برای پردازش موازی مناسبند را در اختیار می‌گذارند. روش کاراتر دیگر عبارتست از تهیه مترجمی[5] با نرم‌افزار چند پردازنده‌ای که می‌تواند بطور اتوماتیک امکان پردازش موازی را در برنامه کاربر کشف کند. مترجم برنامه را برای یافتن وابستگی داده‌ها وارسی می‌کند. اگر بخشی از برنامه به داده‌های تولید شده در بخشی دیگر وابستگی داشته باشد، بخش

---

1- Job
2- Task
3- Function
4- Special Purpose CPU
5- Compiler

تولیدکننده داده باید زودتر اجرا شود. مع الوصف دو بخش از برنامه که از داده‌های تولید شده یکدیگر استفاده نمی‌کنند می‌توانند همزمان اجرا گردند. مترجم موازی کننده تمام برنامه را به منظور یافتن هر گونه وابستگی‌های احتمالی بین داده‌ها چک می‌کند. سپس بخش‌هایی که هیچگونه وابستگی نداشته باشند برای اجرای همزمان روی پردازنده‌های مختلف بررسی می‌شوند.

چند پردازنده‌ها با توجه به نحوه سازماندهی حافظه خود طبقه‌بندی می‌شوند. سیستم چند پردازنده‌ای با حافظه اشتراکی،[1] در طبقه چند پردازنده‌های دارای حافظه اشتراکی یا چند پردازنده‌های با کوپل محکم[2] قرار می‌گیرد. این امر مانع نمی‌شود تا هر پردازنده حافظه محلی خاص برای خود داشته باشد. در واقع، اغلب چند پردازنده‌های با کوپل محکم یک حافظه کش را برای هر CPU در اختیار می‌گذارند. بعلاوه، یک حافظه مشترک کلی وجود دارد که همه CPUهابه آن دسترسی دارند. بنابراین اطلاعات می‌تواند با قرار گرفتن در حافظه مشترک کلی بین CPUها مشترک باشد.

مدل دیگری برای چند پردازنده ها، سیستم دارای حافظه توزیع شده[3] یا کوپل سست[4] است. در یک سیستم با کوپل سست، هر عنصر پردازنده دارای حافظه محلی مربوط به خود است. پردازنده‌ها بوسیله یک طرح سوئیچینگ که برای راه‌دهی اطلاعات از یک پردازنده به پردازنده دیگر از طریق یک طرح پیام‌رسانی طراحی شده اند، بهم متصل می‌شوند. پردازنده‌ها برنامه و داده‌ها را به شکل بسته به پردازنده‌های دیگر می‌فرستند. هر بسته از یک آدرس، محتوای داده و برخی کدهای کشف خطا تشکیل شده است. بسته‌ها بر اساس سیستم ارتباطی بکار رفته، به آدرس یک پردازنده خاص ارسال می‌شوند یا بوسیله اولین پردازنده آزاد دریافت می‌گردند. سیستم‌های با کوپل سست، وقتی عمل و عکس‌العمل متقابل بین کارهای کوچک حداقل است کاراتر می‌باشد، در صورتی که سیستم‌های با کوپل محکم می‌توانند درجه بالاتری از اینگونه اعمال و عکس‌العمل‌ها را بین تکالیف تحمل کنند.

## ۱۳-۲ ساختارهای اتصالات متقابل

اجزایی که یک سیستم چند پردازنده‌ای را تشکیل می‌دهند عبارتند از CPU ها، IOP ها که به وسایل ورودی-خروجی متصلند، و یک واحد حافظه که ممکن است به چند ماژول جداگانه تقسیم شده باشد. اتصالات فی‌مابین این اجزاء بسته به تعداد مسیرهای انتقال موجود بین پردازنده‌ها و حافظه در یک سیستم حافظه مشترک یا میان عناصر پردازش در سیستم‌های با کوپل سست، می‌توانند آرایش‌های مختلف فیزیکی داشته باشد. برای ایجاد شبکه فی‌مابین آنها چند شکل فیزیکی وجود دارد. بعضی از این انواع در این بخش ارائه شده است.

۱- گذرگاه اشتراکی زمان مشترک

---

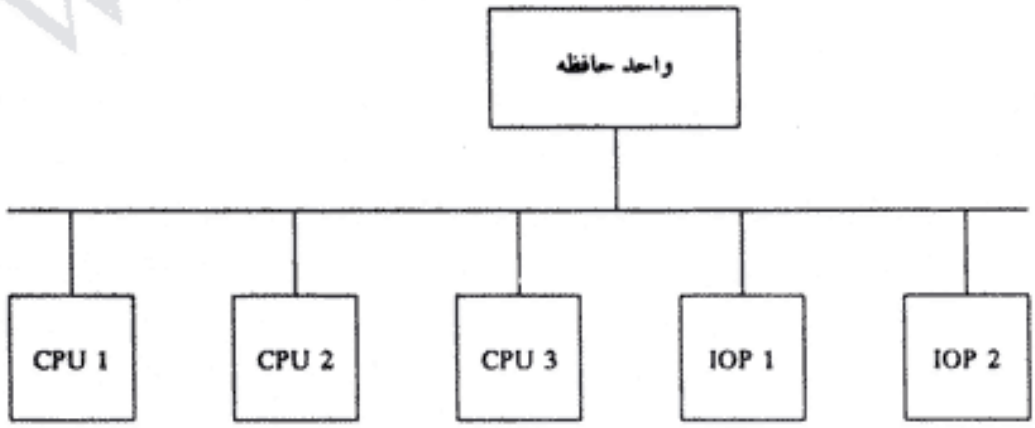
1- Shared Memory
2- Tightly Coupled Multiprocessor
3- Distributed - Memory System
4- Loosely Coupled System

۲- حافظه چند پورتی[1]
۳- سوییچ های تقاطعی
۴- شبکه سوئیچینگ چند طبقه
۵- سیستم فوق مکعبی

### گذرگاه مشترک با تسهیم زمانی

یک چند پردازنده گذرگاه مشترک متشکل است از تعدادی پردازنده که از طریق مسیر های مشترک به یک واحد حافظه متصل شده‌اند. یک گذرگاه مشترک با تسهیم زمانی[2] برای پنج پردازنده در شکل ۱-۱۳ نشان داده است. در هر لحظه از زمان تنها یک پردازنده می‌تواند با حافظه یا پردازنده دیگر ارتباط برقرار کند. عملیات انتقال توسط پردازنده‌ای انجام می‌شود که از طریق یک مسیر مشترک به یک واحد حافظه متصل است. هر پردازنده دیگری که بخواهد یک انتقال را آغاز کند باید ابتدا وضعیت گذرگاه را از نظر آزادبودن آزمایش کند، و پس از در اختیار بودن گذرگاه پردازنده می‌تواند واحد مقصد را برای آغاز انتقال آدرس دهی نماید. برای آگاه کردن واحد مقصد از نوع عمل، دستوری به آن صادر می‌شود. واحد گیرنده آدرس آن را در گذرگاه تشخیص داده و به سیگنال‌های کنترل که از فرستنده پس از آغاز انتقال می‌رسند پاسخ می‌دهد. سیستم بعلت اشتراک گذرگاه ممکن است دچار مشکلاتی در انتقال گردد زیرا یک گذرگاه بوسیله همه پردازنده‌ها به اشتراک گرفته شده است. این مشکلات باید با استفاده از یک کنترل کننده گذرگاه که واحدهای درخواست‌کننده را اولویت‌بندی می‌کند حل گردد.

یک سیستمی با یک گذرگاه مشترک به یک انتقال در هر لحظه محدود است. این بدان معنی است که وقتی یک پردازنده با حافظه تبادل اطلاعات می‌کند، کلیه پردازنده‌های دیگر یا مشغول کار داخلی می‌باشند یا باید بانتظار استفاده از گذرگاه بیکار بمانند. در نتیجه، سرعت انتقال کل در سیستم توسط سرعت یک گذرگاه




شکل ۱-۱۳ سازمان گذرگاه اشتراکی زمان مشترک

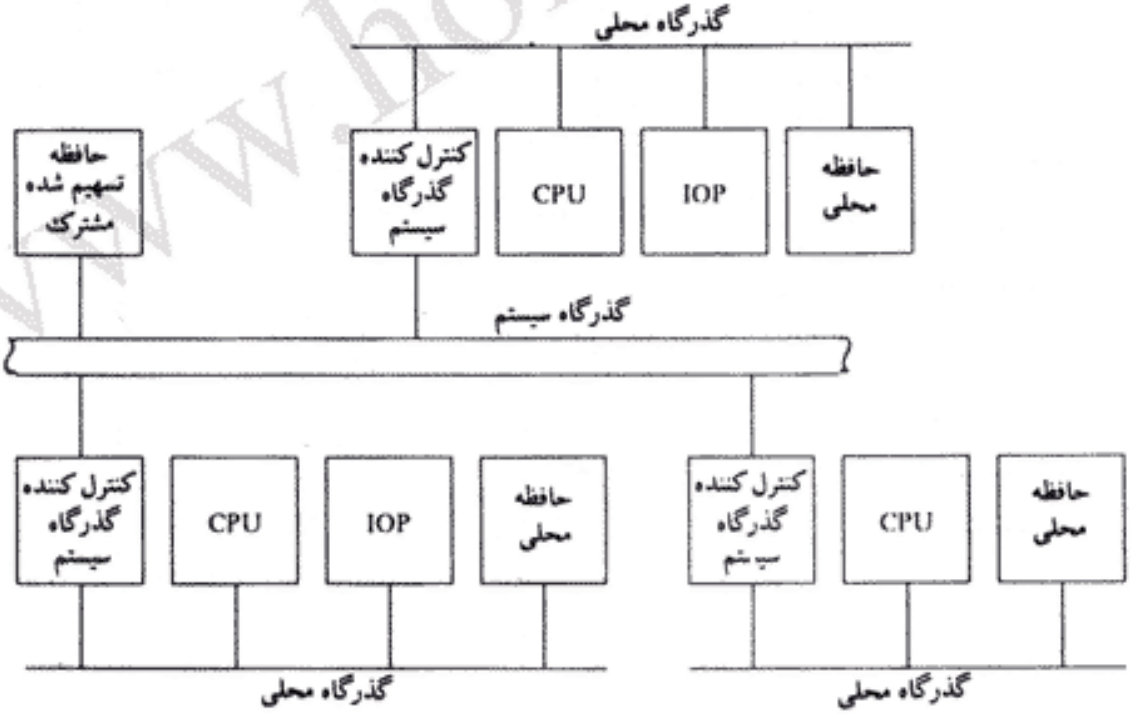
1- Multiport
2- Time Shared Common Bus

محدود می‌شود. پردازنده‌ها در سیستم، اغلب از طریقه بکارگیری دو یا چند گذرگاه برای انتقال‌های همزمان متعدد مشغول نگهداشته می‌شوند. با این وجود این روش هزینه و پیچیدگی سیستم را افزایش می‌دهد.

پیاده‌سازی مقرون به صرفه‌تری برای یک ساختار دو گذرگاهی در شکل ۲-۱۳ نشان داده شده است. در این شکل تعداد گذرگاه‌های محلی وجود دارد که هر کدام به حافظه محلی خود و یک یا دو پردازنده وصل شده‌اند. هر گذرگاه محلی ممکن است به یک CPU، یا IOP، یا هر ترکیبی از پردازنده‌ها متصل گردد. یک سیستم کنترل‌کننده گذرگاه، هر گذرگاه محلی را به سیستم گذرگاه مشترک ارتباط می‌دهد. وسایل I/O متصل به IOP محلی و حافظه محلی در دسترس پردازنده‌های محلی هستند. حافظه متصل به سیستم گذرگاه مشترک بصورت اشتراکی در اختیار همه پردازنده‌ها قرار دارد. اگر یک IOP مستقیماً به گذرگاه سیستم وصل شده باشد، وسایل I/O متصل به آن قابل دسترسی برای پردازنده‌ها خواهند بود. در هر لحظه از زمان تنها یک پردازنده می‌تواند با حافظه اشتراکی یا هر منبع مشترک دیگر از طریق گذرگاه سیستم ارتباط برقرار نماید. سایر پردازنده‌ها با حافظه محلی خود و نیز وسایل I/O مشغول خواهند بود. بخشی از حافظه محلی ممکن است بصورت حافظه کَش متصل به CPU طراحی گردد (بخش ۶-۱۲). بدین ترتیب زمان متوسط دستیابی حافظه محلی به سمت سیکل زمانی CPU متصل به آن سوق داده می‌شود.

## حافظه چند پورتی

یک سیستم حافظه چند پورتی از گذرگاه‌های تفکیک شده بین هر ماژول حافظه[1] و هر CPU استفاده

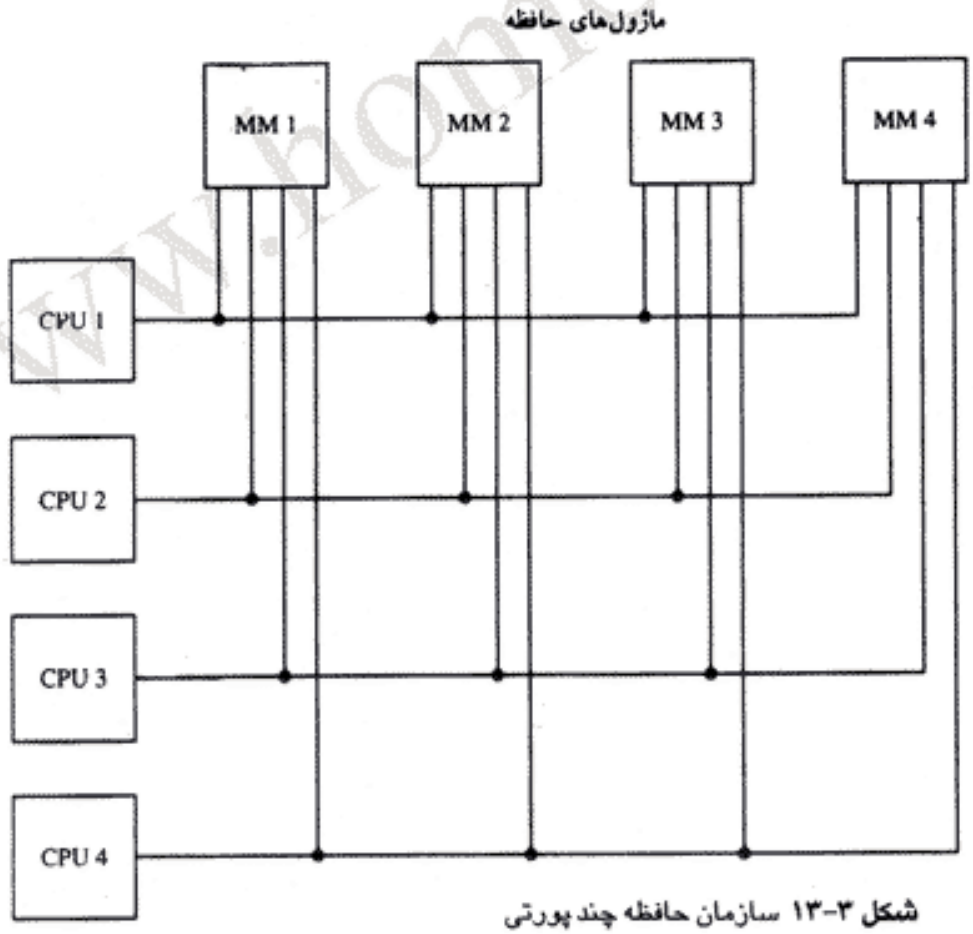



شکل ۲-۱۳ ساختار گذرگاه سیستم برای چند پردازنده‌ها

1- Memory Modules

می‌کند. این آرایش برای چهار CPU و چهار ماژول حافظه (MMS) در شکل ۳-۱۳ نشان داده شده است. گذرگاه هر پردازنده به هر ماژول حافظه متصل شده است. یک گذرگاه پردازنده از خطوط لازم آدرس، داده و کنترل تشکیل شده است تا با حافظه ارتباط برقرار کند. در این آرایش گوئیم ماژول حافظه چهار پورت دارد و هر پورت یکی از گذرگاه‌ها را سرویس می‌دهد. ماژول حافظه باید دارای مدارهای منطقی کنترل داخلی برای تعیین پورتی که باید به حافظه در هر لحظه دستیابی نماید باشد. پیچیدگی دستیابی به حافظه با اختصاص اولویت‌های ثابت به هر پورت حافظه حل می‌شود. اولویت در دستیابی به حافظه متناظر با هر پردازنده را می‌توان با استفاده از انتخاب محل فیزیکی پورتی که گذرگاه آن در هر ماژول اشغال می‌کند تعیین کرد. بااین ترتیب CPU1 بر CPU2 اولویت خواهد داشت، و CPU2 بر CPU3 اولویت دارد، و نهایتاً CPU4 پائین‌ترین اولویت را دارد.

مزیت سازمان حافظه چند پورتی عبارتست از سرعت انتقال بالا که بعلت مسیرهای چندگانه بین پردازنده و حافظه امکان پذیر است. عیب این سازمان این است که نیاز به مدار منطقی کنترل حافظه گرانقیمتی دارد و مقدار زیادی کابل و رابط لازم دارد. در نتیجه ساختار اتصالات فی‌مابین معمولاً برای سیستم‌هایی مناسب است که تعداد کمی پردازنده داشته باشد.

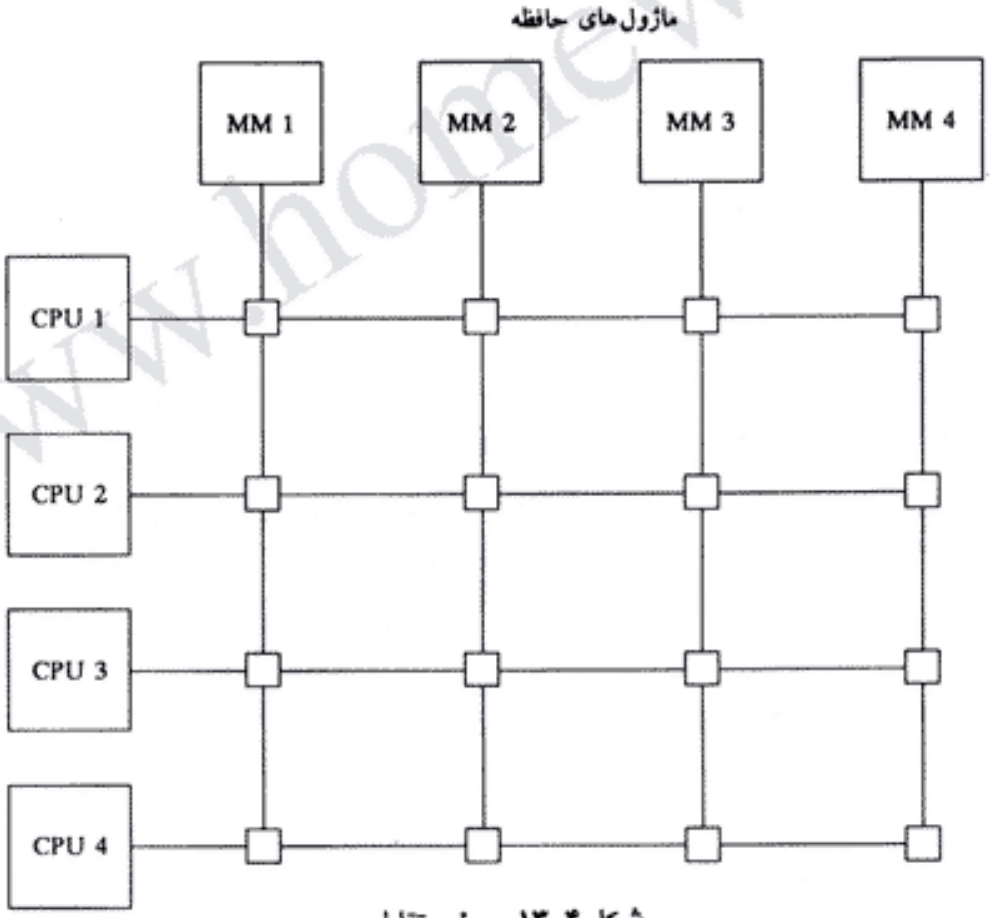



شکل ۳-۱۳ سازمان حافظه چند پورتی

## سوئیچ تقاطعی

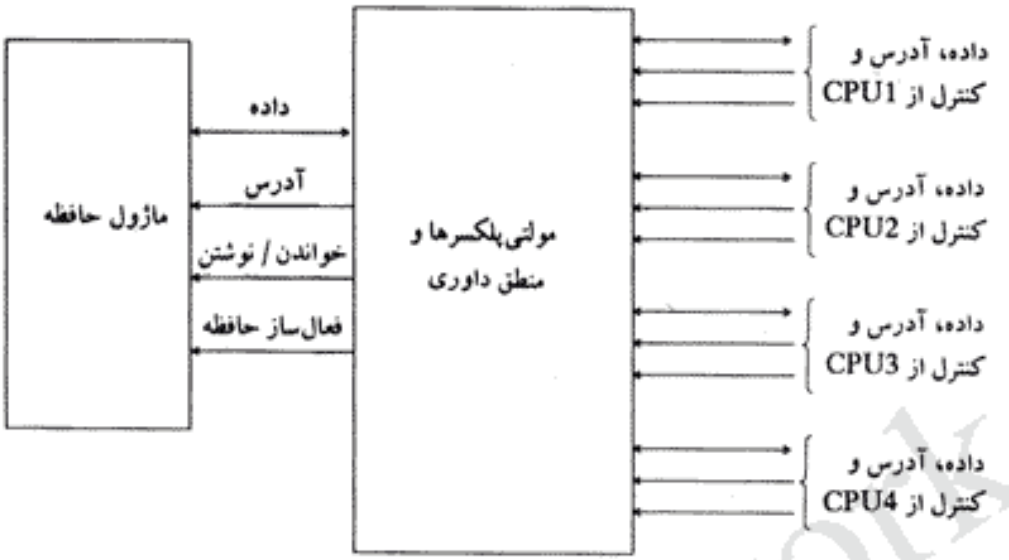
سازمان یک سوئیچ تقاطعی از تعدادی نقطه تقاطعی تشکیل می‌شود که در محل تقاطع گذرگاه‌های بین پردازنده‌ها و مسیرهای ماژول حافظه قرار داده می‌شود. شکل ۴-۱۳ اتصالات بین چهار CPU و چهار ماژول حافظه را با سوئیچ تقاطعی نشان می‌دهد. مربع کوچک در هر نقطه تقاطع یک کلید است که مسیر را از پردازنده به ماژول حافظه معین می‌کند. هر نقطه سوئیچ دارای مدار منطقی کنترل برای ایجاد مسیر بین پردازنده و حافظه است. این سوئیچ آدرس روی گذرگاه را برای تعیین آدرس یک ماژول خاص بررسی می‌کند. همچنین مشکل درخواست‌های متعدد برای دستیابی به یک ماژول آدرس‌دهی شده را بر مبنای اولویت قبلی حل می‌کند.

شکل ۵-۱۳ طرح عملیاتی یک سوئیچ تقاطعی متصل به یک ماژول حافظه را نشان می‌دهد. این مدار از مولتی پلکسرهایی تشکیل می‌شود که داده‌ها، آدرس، و کنترل را از یک CPU برای ارتباط با ماژول حافظه انتخاب می‌کنند. سطوح اولیه توسط منطق داوری[1] برقرار می‌شود تا هر وقت دو




شکل ۴-۱۳ سوئیچ تقاطعی

1- Arbitration Logic




**شکل ۵-۱۳** بلاک دیاگرام سوئیچ تقاطعی

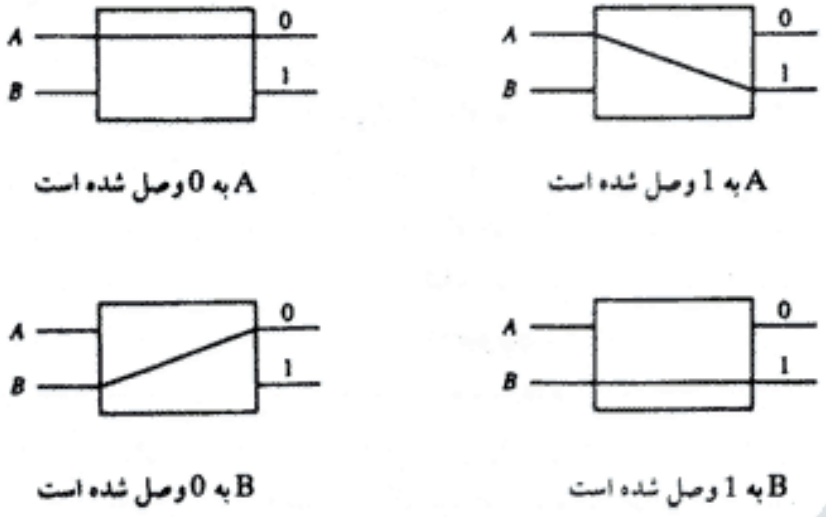
یا چند CPU اقدام به مراجعه به یک حافظه کنند یکی از آنها انتخاب شود. مولتی پلکسرها با کددودویی که توسط یک گذرگاه اولویت در داخل منطق داوری ایجاد می‌شود کنترل می‌گردند.

سازمان سوئیچ تقاطعی انتقال‌های همزمان از همه ماژول‌های حافظه را پشتیبانی می‌کند زیرا برای هر ماژول یک مسیر جداگانه موجود است. با این وجود سخت افزار لازم برای پیاده‌سازی سوئیچ می‌تواند بسیار بزرگ و پیچیده باشد.

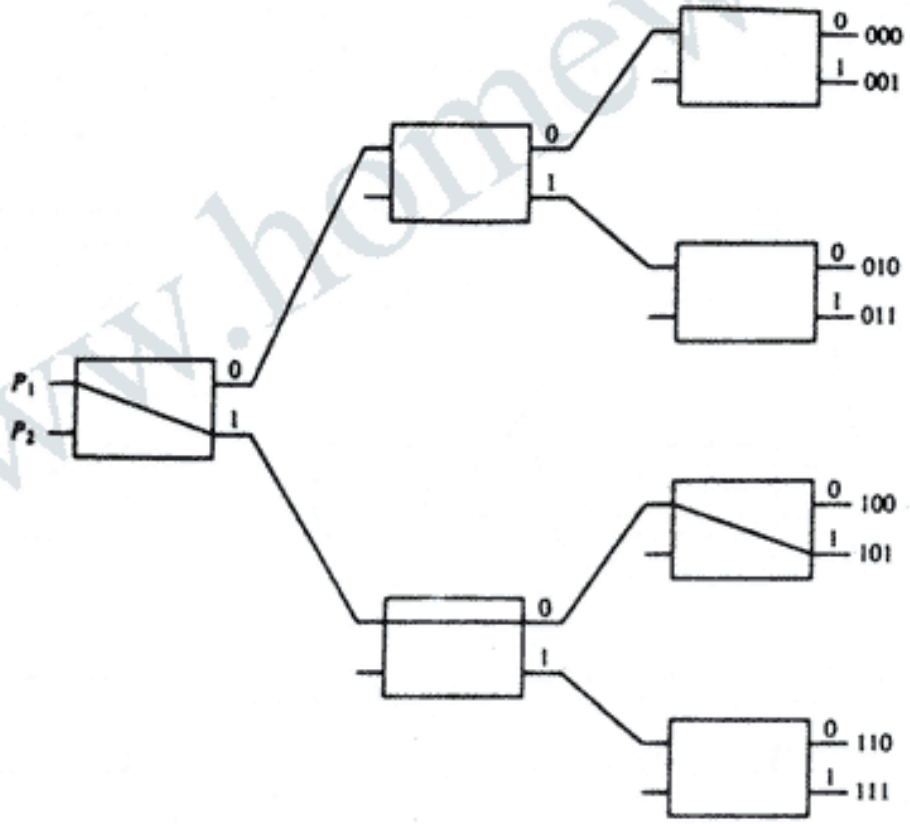
## شبکه سوئیچینگ چند طبقه

جزء اصلی یک شبکه چند طبقه، یک سوئیچ تبادلی دو ورودی - دو خروجی است. همانطور که در شکل ۶-۱۳ دیده می‌شود، سوئیچ 2×2 دارای دو ورودی A و B، و دو خروجی 0 و 1 است. در رابطه با هر سوئیچ سیگنال‌های کنترلی (دراینجا نشان داده نشده‌اند) که این ارتباط بین پایانه‌های ورودی - و خروجی را برقرار کنند وجود دارند. سوئیچ قابلیت اتصال A را به هر یک از خروجی‌ها دارد. پایانه B در سوئیچ نقش مشابهی دارد. همچنین سوئیچ داوری بین درخواست‌های مشکل‌ساز را دارد. اگر ورودی‌های A و B، هر دو یک پایانه خروجی را تقاضا کنند، تنها یکی از آنها متصل شده و دیگری بلوکه می‌شود.

با استفاده از سوئیچ 2×2 بعنوان بلاک ساختمانی، می‌توان یک شبکه چند طبقه برای کنترل ارتباط بین چند منبع و مقصد ساخت. برای ملاحظه نحوه انجام این کار، درخت دودویی شکل ۷-۱۳ را در نظر بگیرید. دو پردازنده $P_1$ و $P_2$ از طریق سوئیچ‌ها به هشت ماژول حافظه که بصورت 000 تا 111 شماره‌گذاری شده‌اند متصل شده‌اند. هر مسیر یک منبع به یک مقصد از بیت‌های شماره مقصد تعیین می‌شود. اولین بیت شماره مقصد خروجی سوئیچ را در اولین طبقه تعیین می‌کند. بیت دوم خروجی




شکل ۶-۱۳ عملکرد یک سوئیچ متقابل 2×2



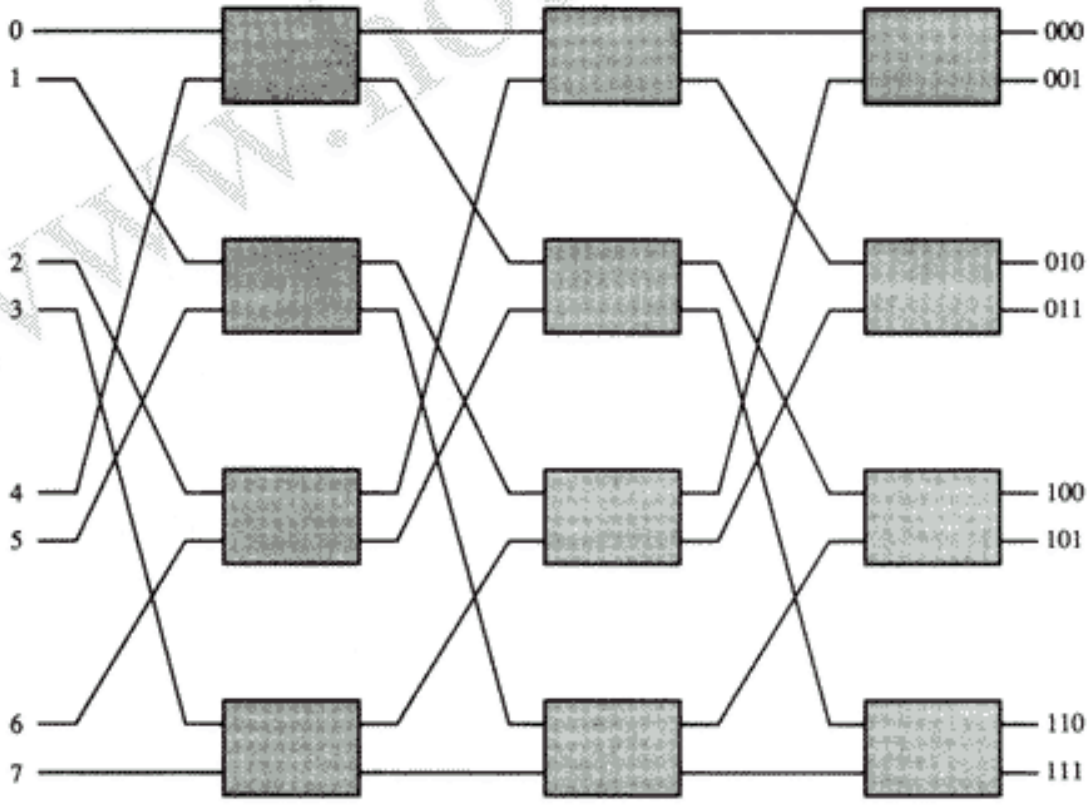

شکل ۷-۱۳ درخت دودویی با سوئیچهای 2×2

سوئیچ طبقه دوم را مشخص می‌سازد، و سومین بیت خروجی سوئیچ را در طبقه سوم معین می‌نماید. مثلاً، برای اتصال $P_1$ به حافظه 101، لازم است مسیری از $P_1$ به خروجی 1 در سطح اول سوئیچ، به

خروجی 0 در سطح دوم سوئیچ، و به خروجی 1 در سطح سوم برقرار شود. واضح است که هر یک از دو ورودی $P_1$ یا $P_2$ قابل اتصال به هریک از هشت حافظه است. با این وجود الگوهای تقاضای معینی برای ارتباط نمی‌تواند بطور همزمان ایجاد شود. مثلاً، اگر $P_1$ به یکی از مقصدهای 000 تا 011 وصل شود، $P_2$ فقط به یکی از مقصدهای 100 تا 111 وصل می‌گردد.

توپولوژی‌های متعددی برای شبکه‌های سوئیچینگ چند طبقه جهت کنترل ارتباط پردازنده - حافظ در سیستم چند پردازنده با کوپل محکم یا کنترل ارتباط بین عناصر پردازنده در سیستم باکوپل سست پیشنهاد شده است . نمونه‌ای از این توپولوژی‌ها شبکه سوئیچینگ شکل ۸-۱۳ می‌باشد. در این آرایش، دقیقاً یک مسیر از هر منبع به هر مقصد بخصوص وجود دارد. با این وجود برخی الگوهای مورد نظر را نمی‌توان بطور همزمان برقرار کرد. مثلاً هر دو منبع نمی‌توانند به مقصدهای 000 و 001 وصل شوند.

یک درخواست بخصوص در شبکه سوئیچینگ توسط مبداء آغاز می‌شود، و ضمن آن یک الگوی ۳ بیتی نماینده شماره مقصد را ارسال می‌کند. با حرکت الگوی دودویی در طول شبکه، هر یک از سطوح بیت متفاوتی را بررسی می‌کنند تا نحوه تنظیم سوئیچ 2x2 تعیین شود. سطح 1 پرارزش‌ترین بیت را بررسی می‌کند، سطح 2 بیت وسط، و سطح 3 کم ارزشترین بیت را چک می‌نماید. وقتی که تقاضا به هر یک از ورودی‌های سوئیچ $2 \times 2$ می‌رسد، اگر بیت مشخص شده 0 باشد به خروجی بالایی و اگر 1 باشد به خروجی پائینی هدایت می‌شود.



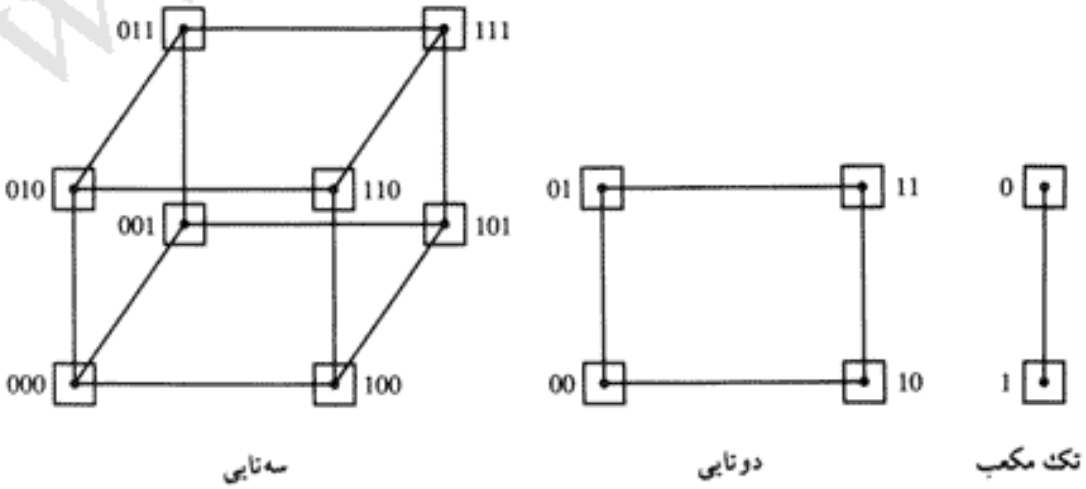

شکل ۸-۱۳ شبکه سوئیچینگ امگای $8 \times 8$

در یک سیستم چند پردازنده با کوپل محکم، منبع یک پردازنده و مقصد ماژول حافظه است. اولین مرور در شبکه مسیر را برقرار می‌کند. مرورهای بعدی برای انتقال آدرس به حافظه و سپس انتقال داده‌ها در یکی از دو جهت، بسته به اینکه درخواست خواندن یا نوشتن باشد بکار می‌روند. در سیستم چند پردازنده با کوپل سست، هر دو عنصر منبع و مقصد عناصر پردازنده هستند. پس از برقراری مسیر، پردازنده منبع پیام را به پردازنده مقصد انتقال می‌دهد.

## اتصال فوق مکعبی [1]

ساختار چند پردازنده فوق مکعبی یا چند پردازنده n بعدی، یک سیستم با کوپل سست متشکل از $2^n$ پردازنده است که بشکل یک مکعب دودویی n بعدی بهم متصل شده‌اند. هر پردازنده یک گره را برای مکعب تشکیل می‌دهد. گرچه معمولاً گفته می‌شود که هر گره دارای یک پردازنده است ولی در واقع نه فقط یک CPU، بلکه یک حافظه محلی و یک واسطه I/O نیز بهمراه آن است. هر پردازنده ارتباط مستقیمی با پردازنده‌های مجاور خود دارد. این مسیرها اضلاع مکعب را تشکیل می‌دهند. $2^n$ آدرس دودویی متمایز وجود دارد که می‌توانند به پردازنده‌ها اختصاص یابند. هر آدرس پردازنده با آدرس هر یک از n پردازنده مجاور آن دقیقاً در یک بیت اختلاف دارند.

شکل ۹-۱۳ ساختار یک فوق مکعب را برای 3و 2و 1 =n نشان می‌دهد. در ساختار مکعب یک تایی داریم n=1 و $2^n$=2 این ساختار حاوی دو پردازنده متصل بهم توسط یک مسیر است. ساختار مکعب دوتایی دارای n=2 و $2^n$=4 است. این ساختار دارای چهار گره می‌باشد که بشکل مستطیل بهم متصلند. یک ساختار مکعب سه تایی دارای هشت گره متصل بهم می‌باشد و تشکیل مکعب را می‌دهد. یک ساختار مکعب




**شکل ۹-۱۳** ساختارهای فوق مکعبی برای n = 1, 2, 3

1- Hypercube Interconnection

n تایی دارای $2^n$ گره بوده و هر پردازنده در یک گره قرار گرفته است. به هر گره یک آدرس دودویی آنچنان اختصاص یافته است که آدرس دو گره مجاور دقیقاً در یک بیت اختلاف دارند. مثلاً سه گره همسایه با آدرس 100 در یک ساختار مکعب سه تایی عبارتند از 000، 110 و 101. هر یک از این اعداد دودویی با آدرس 100 تنها در یک بیت اختلاف دارند.

راه‌دهی پیام‌ها در ساختار مکعب n تایی ممکن است از یک تا n مسیر متصل، از هر گره منبع به گره مقصد باشد. مثلاً در ساختار مکعب سه تایی، گره 000 می‌تواند مستقیماً با گره 001 ارتباط برقرار کند. برای ارتباط با 011 باید دست کم از دو ضلع عبور کند (از 000 به 001 به 011، یا از 000 به 010، به 011). برای ارتباط گره 000 با گره 111 باید دست کم از سه ضلع عبور کرد. یکی از روش های مسیریابی محاسبه OR انحصاری آدرس گره مبداء با آدرس گره مقصد است. در عدد دودویی حاصل، بیت های متناظر با محورهایی که روی آنها دو گره با هم متفاوتند مقدار 1 خواهند داشت. سپس پیام در طول این محورها ارسال می‌شود. مثلاً در ساختار مکعب سه تایی، برای ارسال پیامی از 010 به 001، OR انحصاری دو آدرس محاسبه می‌شود و نتیجه 011 را می‌دهد. پیام می‌تواند در طول محور دوم به 000 و سپس در طول محور سوم به 001 ارسال شود.

مثالی از معماری فوق مکعبی، مجموعه کامپیوتری iPSC اینتل است. این سیستم از 128 (n=7) ریز کامپیوتر، که از طریق کانال‌های ارتباطی بهم متصل اند، تشکیل شده است. هر گره از یک CPU، یک پردازنده ممیز شناور، حافظه محلی، و واحدهای واسطه ارتباطات سری تشکیل می‌شود. هر یک از گره‌ها بر اساس برنامه های مقیم بطور مستقل بر روی داده های ذخیره شده در حافظه محلی عمل می‌کند. داده‌ها و برنامه های هر گره از طریق یک سیستم پیام رسانی از گره‌های دیگر یا از مدیر مکعب تامین می‌شود. برنامه‌های کاربردی روی مدیر مکعب تهیه، ترجمه و سپس در گره‌های مختلف بار می‌شوند. محاسبات در سیستم توزیع و بطور موازی اجرا می‌گردد.

## ۱۳-۳ داوری بین پردازنده‌ها[1]

سیستم های کامپیوتری برای سهولت انتقال اطلاعات بین اجزاء مختلف دارای تعدادی گذرگاه در سطوح مختلف هستند. CPU حاوی گذرگاه‌های داخلی برای انتقال اطلاعات بین ثبات های پردازنده و ALU است. گذرگاه حافظه از خط انتقال داده، آدرس و اطلاعات خواندن و نوشتن تشکیل شده است. یک گذرگاه I/O برای انتقال اطلاعات به وسایل ورودی و خروجی و بالعکس استفاده می‌شود. گذرگاهی که تعداد قابل توجهی از اجزاء مانند CPU ها و IOP را در یک سیستم چند پردازنده‌ای بهم وصل می‌کنند گذرگاه سیستم[2] نامیده می‌شوند. مدارهای فیزیکی یک گذرگاه سیستم در تعدادی صفحات مدار چاپی[3] مشابه قرار داده می‌شود. هر یک از این صفحات سیستم به یک ماژول خاص تعلق

---

1- Interprocessor Arbitration
2- System bus
3- Printed Circuit Board

دارد. صفحات از طریق کانکتورهایی بطور موازی با هم نصب می‌شوند. هر پایه از هر کانکتور [1] یک مدار با سیم به پایه‌های مربوط به خود بر روی سایر بردها وصل می‌شود. بنابراین هر مدار را می‌توان در یک شکاف [2] در صفحه نگهدارنده‌ای [3] که گذرگاه سیستم را تشکیل می‌دهد قرار داد.

پردازنده‌ها در یک سیستم چند پردازنده‌ای نیاز به حافظه مشترک یا سایر منابع مشترک از طریق گذرگاه سیستم دارند. اگر هیچیک از پردازنده‌ها از گذرگاه سیستم استفاده نکنند، تقاضای دستیابی یک پردازنده ممکن است بلافاصله پاسخ داده شود. با این وجود، در صورت استفاده گذرگاه سیستم بوسیله پردازنده دیگر، پردازنده متقاضی باید صبر کند. بعلاوه، دیگر پردازنده‌ها هم ممکن است بطور همزمان گذرگاه سیستم را تقاضا کنند. برای حل این رقابت چند جانبه بوسیله منابع نوعی داوری باید صورت گیرد. منطق داوری بخشی از کنترل کننده گذرگاه سیستم است که بین گذرگاه سیستم، طبق شکل ۲-۱۳، قرار دارد.

### گذرگاه سیستم

یک گذرگاه سیستم از تقریباً 100 خط سیگنال تشکیل می‌شود. این خطوط به سه گروه عملیاتی تقسیم می‌شوند: خطوط داده، آدرس و کنترل. بعلاوه خطوط توزیع نیروی الکتریکی نیز برای تأمین توان موردنیاز قطعات لازم است. مثلاً سیستم چند گذرگاهی 796 استاندارد IEEE دارای 16 خط داده، 24 خط آدرس، 26 خط کنترل و 20 خط توان و کلاً 86 خط می‌باشد.

خطوط داده مسیری برای انتقال داده بین پردازنده و حافظه مشترک فراهم می‌نمایند. تعداد خطوط داده معمولاً مضربی از 8 و عمدتاً 16 و یا 32 می‌باشد. خطوط آدرس برای شناسایی آدرس یک حافظه یا هر منبع یا مقصد دیگر مانند پورت‌های ورودی یا خروجی می‌باشد. تعداد خطوط آدرس حداکثر ممکن ظرفیت حافظه را در سیستم معین می‌کند. مثلاً یک آدرس 24 خطی می‌تواند تا به $2^{24}$ (16 مگا) کلمه حافظه دستیابی نماید. خطوط داده و آدرس به بافرهای سه حالته منتهی می‌شود (شکل ۵-۴). بافرهای آدرس یک جهته بوده و از پردازنده بسمت حافظه می‌باشند. خطوط داده دو جهته بوده (شکل ۳-۱۲) و انتقال داده را در هر دو جهت ممکن می‌سازند.

انتقال داده‌ها در گذرگاه سیستم می‌تواند همگام و یاغیرهمگام باشد. در یک گذرگاه همگام، هر قلم داده در طول یک بازه زمانی که در ابتدا برای هر دو واحد منبع و مقصد مشخص است انتقال می‌یابد. عمل همگامی با بکار انداختن هر دو واحد توسط یک منبع ساعت مشترک صورت می‌گیرد.

روش دیگر این است که ساعتهای جداگانه‌ای با فرکانس تقریباً یکسان در هر واحد داشته باشیم. سیگنال‌های همگام سازی بطور متناوب ارسال می‌شوند تا همه ساعت‌های سیستم همگام با یکدیگر باقی بمانند. در یک سیستم غیرهمگام هر قلم داده‌ای که انتقال می‌یابد با سیگنال کنترل دست دهی (شکل ۹-۱۱) برای نشان دادن زمان انتقال داده‌ها از مبدأ و دریافت آنها توسط مقصد همراه است.

خطوط کنترل سیگنال‌هایی را برای کنترل انتقال اطلاعات بین واحدها فراهم می‌کنند. سیگنال‌های

---

1- Connector 2- Slot 3- Backplane

زمانبندی، اعتبار اطلاعات داده و آدرس را مشخص می‌کنند. سیگنال‌های فرمان، عملیاتی را که باید انجام شود مشخص می‌کنند. خطوط کنترل شامل سیگنال‌های انتقال از قبیل خواندن و نوشتن حافظه، تصدیق یک انتقال، درخواست‌های وقفه، و سیگنال‌های کنترل گذرگاه از قبیل درخواست گذرگاه و واگذاری گذرگاه و سیگنال‌های مربوط به رویه‌های داوری هستند.

در جدول ۱-۱۳، 86 خطی که در گذرگاه چندتایی 796 استاندارد IEEE وجود دارد فهرست شده است. این سیستم شامل 16 خط داده و 24 خط آدرس است. همه سیگنال‌های این گذرگاه در سطح پائین خود فعال می‌شوند. از جمله سیگنال‌های کنترل انتقال داده عبارتند از خواندن و نوشتن حافظه و خواندن و نوشتن I/O. در نتیجه خطوط آدرس می‌توانند فضاهای حافظه و I/O را جداگانه آدرس‌دهی کنند. وقتی که انتقال به پایان رسید، حافظه یا I/O با سیگنال تصدیق انتقال پاسخ می‌دهد. هر پردازنده متصل به چند پورت تا هشت خروجی درخواست وقفه و یک خط ورودی تصدیق وقفه دارد. این خروجی‌ها معمولاً به یک کنترل کننده وقفه اولویت دار مشابه باکنترل کننده‌ای که در بخش ۲۱-۱۱

جدول ۱-۱۳ سیگنال‌های چند گذرگاه 796 استاندارد IEEE

| نام سیگنال | |
|---|---|
| داده آدرس | |
| خطوط داده (16 خط) | DATA0-DATA15 |
| خطوط آدرس (24 خط) | ADRS0-ADRS23 |
| انتقال داده | |
| خواندن حافظه | MRDC |
| نوشتن حافظه | MWTC |
| خواندن I/O | IORC |
| نوشتن I/O | IOWC |
| تصدیق انتقال | TACK |
| کنترل وقفه | |
| درخواست وقفه (8 خط) | INT0-INT7 |
| تصدیق وقفه | INTA |
| سیگنال‌های کنترل متفرقه | |
| ساعت اصلی | CCLK |
| مقداردهی اولیه سیستم | INIT |
| فعال‌ساز بایت بالا | BHEN |
| بازدارنده حافظه (2 خط) | INH1-INH2 |
| قفل گذرگاه | LOCK |
| داوری گذرگاه | |
| درخواست گذرگاه | BREQ |
| درخواست گذرگاه مشترک | CBRQ |
| اشغال | BUSY |
| ساعت گذرگاه | BCLK |
| ورودی اولویت گذرگاه | BPRN |
| خروجی اولویت گذرگاه | BPRO |
| منبع تغذیه و زمین (20 خط) | |

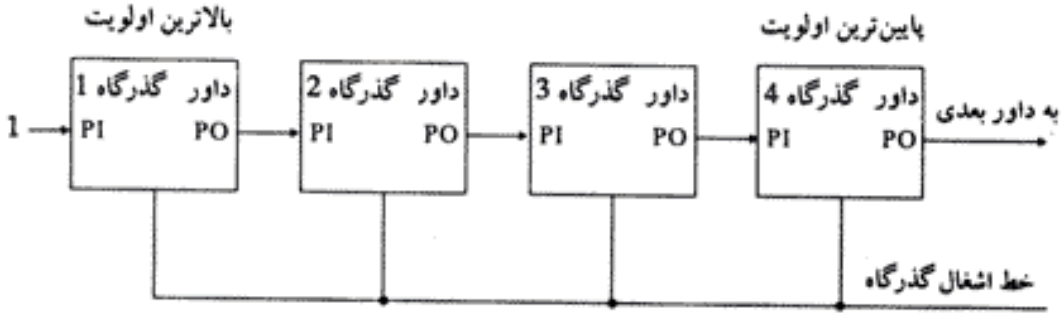
شرح داده شد اعمال می‌شوند. سیگنال‌های کنترل متفرقه قابلیت زمانبندی و مقداردهی اولیه را در اختیار می‌گذارند. خصوصاً سیگنال قفل کردن گذرگاه برای کاربردهای مربوط به چند پردازنده‌ها ضروری است. این سیگنال که توسط پردازنده فعال می‌شود برای جلوگیری از در اختیار قرار گرفتن گذرگاه توسط پردازنده‌های دیگر در حال اجرای دستورالعمل‌های تست و نشاندن به کار می‌رود. این دستورالعمل‌ها برای همگام سازی صحیح پردازنده لازم‌اند (بخش ۴-۱۳).

شش سیگنال داوری گذرگاه برای داوری بین پردازنده بکار می‌روند. این سیگنال‌ها را بعداً پس از بحثی درباره رویه‌های داوری سری و موازی شرح می‌دهیم.

## رویه داوری سری

رویه‌های داوری به تمام تقاضاهای پردازنده‌ها براساس اولویت‌های مقرر شده سرویس می‌دهند. تکنیک سخت افزاری اولویت‌بندی درخواست‌های گذرگاه را با استفاده از اتصال سری یا موازی واحدهای درخواست کننده کنترل گذرگاه سیستم می‌توان برقرار کرد. تکنیک اولویت دهی سری از یک اتصال زنجیره‌دوار [1] بیان شده در بخش ۵-۱۱ حاصل می‌گردد. اولویت به پردازنده‌های متصل به گذرگاه سیستم براساس موقعیت آنها در طول خط کنترل اولویت حاصل می‌شود. نزدیکترین وسیله به خط اولویت دارای بالاترین اولویت است. وقتی که وسیله‌های متعددی بطور همزمان استفاده از گذرگاه را درخواست کنند، به وسیله‌ای که بالاترین اولویت را دارد امکان دستیابی داده می‌شود.

شکل ۱۰-۱۳ اتصل زنجیره دوار چهار داور را نشان می‌دهد. فرض بر این است که هر پردازنده مدار داور گذرگاه خاص خود را با خطوط ورودی و خروجی اولویت داراست. خروجی اولویت PO از هر داور به ورودی اولویت مدار داور اولویت طبقه پائین‌تر متصل است. PI مربوط به واحد دارای اولویت بالاتر در مقدار منطقی 1 نگهداشته می‌شود. اگر واحدی که بالاترین اولویت را در سیستم دارد گذرگاه سیستم را تقاضا کند همیشه موفق به دستیابی به آن خواهد بود. اگر ورودی یک داورخاص PI برابر 1 باشد و پردازنده مربوطه‌اش تقاضای کنترل گذرگاه را نکند در اینصورت خروجی PO آن برابر 1 است. به این




**شکل ۱۰-۱۳** داوری سری (زنجیره‌ای)

1- daisy chain

ترتیب است که اولویت به واحد بعدی در زنجیره منتقل می‌شود. اگر پردازنده درخواست کنترل گذرگاه کند و داور مربوطه‌اش ورودی PI متناظرش را برابر 1 بیابد خروجی PO را 0 می‌کند. داور الویت پائین‌تر یک 0 در PI دریافت کرده و یک 0 در PO تولید می‌کند. بنابراین به پردازنده‌ای که دارای PI=1 و PO=0 باشد کنترل گذرگاه داده می‌شود.

ممکن است وقتی که پردازنده‌ای در میانه یک عمل با گذرگاه است، پردازنده‌ای با الویت بالاتر گذرگاه را درخواست کند. در این حال پردازنده با اولویت پائین‌تر باید قبل از رها کردن کنترل گذرگاه، عمل خود را روی آن به پایان برساند. خط اشغال گذرگاه[1] که در شکل ۱۰-۱۳ نشان داده شده است، مکانیزمی را برای کنترل بترتیب انتقال فراهم می‌سازد. خط اشغال از مدارهای کلکتور باز در هر مدار گرفته شده و اتصال OR سیمی[2] را فراهم می‌کند. وقتی که یک داور کنترل گذرگاهی را دریافت می‌کند (زیرا در آن PI=1 و PO=0 است) خط اشغال را بررسی می‌کند. اگر خط غیرفعال باشد، بدان معنی است که پردازنده دیگری در حال استفاده از گذرگاه است. داور خط اشغال را فعال کرده و پردازنده مربوطه آن، اختیار گذرگاه را بدست می‌گیرد. با این وجود اگر داور گذرگاه را اشغال بیابد مفهوم این است که پردازنده دیگری درحال استفاده از گذرگاه است. داور به بررسی خود از خط اشغال ادامه می‌دهد در حالی که پردازنده با اولویت پائین‌تر که کنترل گذرگاه را از دست داده است عمل خود را به پایان می‌رساند. وقتی که خط اشغال گذرگاه به حالت غیرفعال خود باز گردد، داور با اولویت بالاتر خط اشغال را فعال می‌نماید و از آن به بعد پردازنده مربوطه‌اش قادر خواهد بود تا انتقال خود را روی گذرگاه انجام دهد.

## منطق داوری موازی

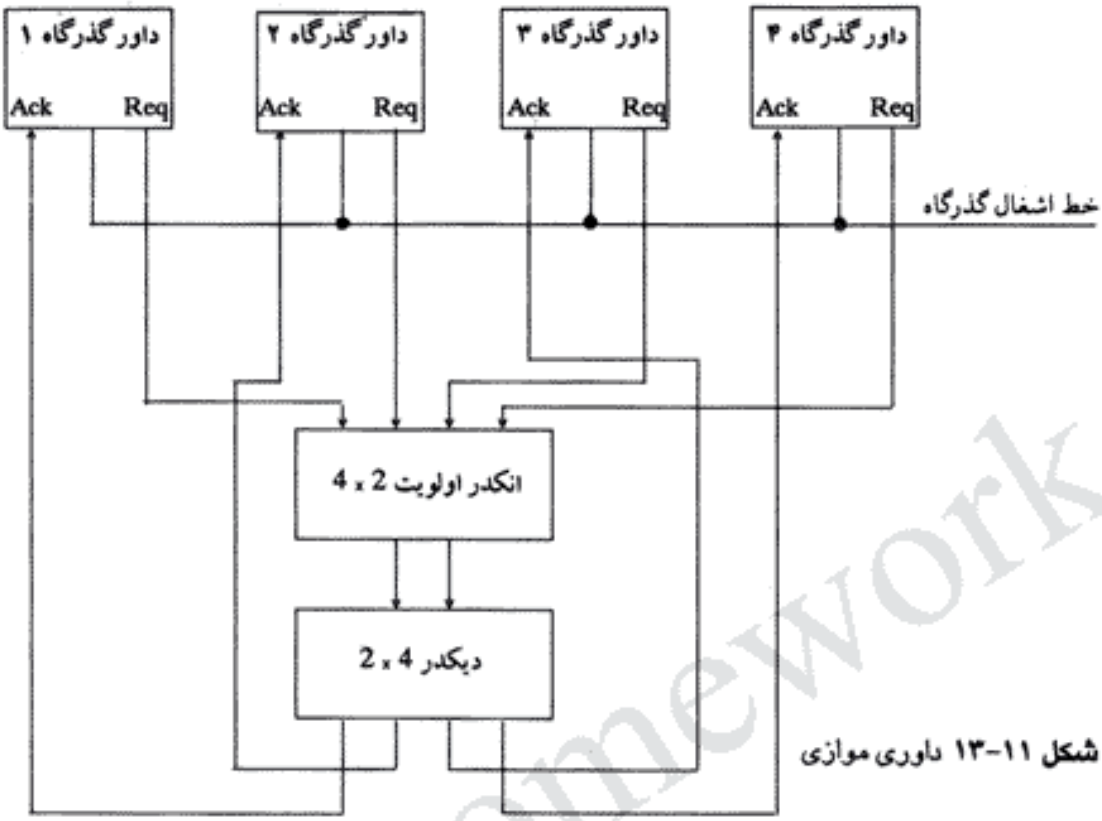
در تکنیک داوری موازی گذرگاه، از یک انکدر اولویت خارجی و یک دیکدر که طبق شکل ۱۱-۱۳ نشان داده شده است استفاده می‌شود. در روش موازی، هر داور گذرگاه یک خط خروجی تقاضای گذرگاه و یک خط ورودی تصدیق گذرگاه دارد. هر داور وقتی که پردازنده دستیابی به گذرگاه سیستم را درخواست کند، خط تقاضا را فعال می‌نماید. پردازنده بشرطی کنترل گذرگاه را بدست می‌گیرد که خط ورودی تصدیق فعال شده باشد، خط اشغال گذرگاه همچون روش زنجیره‌ای، کنترل بترتیب انتقال را انجام می‌دهد.

شکل ۱۱-۱۳ خطوط تقاضا را که از چهار داور به یک انکدر اولویت می‌روند نشان می‌دهد. خروجی انکدر یک کد دو بیتی را که نمایش دهنده واحدی است که بالاترین اولویت را در میان متقاضیان گذرگاه دارد تولید می‌کند. جدول درستی انکدر اولویت را می‌توان در جدول ۲-۱۱ (بخش ۵-۱۱) یافت. کد دوبیتی از خروجی دیکدر، 2x4 که خط تصدیق مشخصی را فعال می‌نمایند، دستیابی به گذرگاه را برای واحدی که بالاترین اولویت را دارد امکان پذیر می‌سازد.

اکنون ما عملکرد و سیگنال‌های داوری اولویت لیست شده در جدول ۱-۱۳ را توضیح می‌دهیم.

---

1- Bus Busy Line 2- Wired OR




**شکل ۱۱-۱۳** داوری موازی

ورودی الویت گذرگاه[1] BPRN و خروجی اولویت گذرگاه[2] BPRO برای اتصال زنجیره‌ای مدارهای داوری گذرگاه بکار رفته‌اند. سیگنال اشغال گذرگاه BUSY یک خروجی کلکتور باز است که برای اعلام مشغول بودن گذرگاه به همه داورها به کار می‌رود. درخواست گذرگاه مشترک CBRQ نیز یک خروجی کلکتور باز است که برای اعلام این موضوع به داور بکار می‌رود که داورهای دیگری با اولویت پائین‌تر در حال درخواست استفاده از گذرگاه سیستم هستند یا خیر. سیگنال‌های مورد استفاده برای پیاده‌سازی رویه داوری موازی، سیگنال درخواست گذرگاه BREQ و ورودی اولویت BPRN متناظر با سیگنال‌های درخواست و تصدیق شکل ۱۱-۱۳ هستند. ساعت گذرگاه BCLK برای همگام کردن همه همزمانی همه انتقال‌های گذرگاه بکار می‌روند.

## الگوریتم‌های داوری دینامیک

دو روش داوری که در فوق توصیف شد از الگوریتم اولویت استاتیک استفاده می‌نماید زیرا اولویت هر وسیله توسط ارتباطش با گذرگاه تثبیت می‌گردد. درمقابل، یک الگوریتم داوری دینامیک به سیستم

---

1- Bus Priority - in  2- Bus Priority - out

توانایی تغییر اولویت وسایل را در حین عمل فراهم می‌کند. اکنون درباره چند رویه داوری که از الگوریتم‌های داوری دینامیک استفاده می‌کنند بحث خواهیم کرد.

الگوریتم برش زمانی[1]، برشی با طول ثابت از زمان گذرگاه را بترتیب و بصورت چرخشی[2] به هر پردازنده اختصاص می‌دهد. سرویس ارائه شده به هر سیستم با این روش مستقل از مکان آن در طول گذرگاه است. به هیچ وسیله خاصی ارجحیتی داده نمی‌شود چون به هریک مقدار زمان مساوی برای ارتباط با گذرگاه داده می‌شود.

در یک سیستم گذرگاه که از همه پرسی[3] استفاده می‌کند، بجای سیگنال واگذاری گذرگاه مجموعه‌ای از خطوط را بعنوان خطوط همه پرسی داریم که به همه واحدها متصلند. این خطوط بوسیله کنترل کننده گذرگاه برای تعیین یک آدرس برای هر وسیله متصل به گذرگاه استفاده می‌شوند. کنترل کننده گذرگاه رشته آدرس‌ها را بصورت از قبل تعیین شده‌ای مرور می‌کند. اگر پردازنده‌ای که درخواست دستیابی نموده آدرس خود را تشخیص دهد، خط اشغال گذرگاه را فعال می‌کند و سپس به گذرگاه دست می‌یابد. پس از چند سیکل گذرگاه، فرآیند همه پرسی با انتخاب یک پردازنده دیگر ادامه می‌یابد. دنباله همه پرسی معمولاً قابل برنامه ریزی است و لذا انتخاب اولویت را می‌توان تحت کنترل برنامه تغییر داد.

الگوریتم قدیمی‌ترین مورد استفاده (LRU) بالاترین اولویت را به وسیله درخواست‌کننده می‌دهد که مدت عدم استفاده آن از گذرگاه بیش از بقیه باشد. اولویت‌ها پس از چند سیکل ساعت طبق الگوریتم LRU تنظیم می‌شود. با این رویه به هیچ پردازنده‌ای نسبت به دیگری ارجحیت داده نمی‌شود زیرا اولویت‌ها بصورت دینامیکی تغییر می‌کنند تا هر وسیله فرصت دستیابی به گذرگاه را داشته باشد.

در روش اولین سرویس- اولین درخواست[4]، تقاضاها بترتیبی که دریافت می‌شوند سرویس‌دهی می‌گردند. برای پیاده کردن این الگوریتم، کنترل‌کننده گذرگاه، صفی را براساس زمان دریافت تقاضاهای گذرگاه مرتب شده ایجاد می‌کند. هر پردازنده باید برای رسیدن نوبت استفاده از گذرگاه براساس اولین ورودی- اولین خروجی (FIFO)، صبر کند.

روشی زنجیره‌ای چرخان[5] نوع تعمیم یافته دینامیکی از الگوریتم زنجیره‌ای است. دراین روش کنترل کننده مرکزی گذرگاه موجود نیست، و خط اولویت از خروجی اولویت[6] آخرین وسیله به ورودی اولویت[7] اولین وسیله بازگشته و یک حلقه بسته را تشکیل می‌دهد. این نحوه اتصال مشابه به شکل ۱۰-۱۳ است جز اینکه خروجی PO داور 4 به ورودی PI داور 1 متصل شده است. هر وسیله‌ای که به گذرگاه دست یابد بعنوان کنترل‌کننده داور بعدی نیز عمل می‌کند. اولویت هر داور برای هر سیکل گذرگاه بر اساس مکانش در طول خط اولویت گذرگاه و با شروع از داوری که پردازنده آن در حال حاضر کنترل گذرگاه را در دست دارد معین می‌شود. وقتی که داوری گذرگاه را رها می‌کند، کمترین اولویت را خواهد داشت.

---

1- Time Slite
2- Round - Robin Fashhion
3- Polling
4- First Come - First Serve
5- Rotating Daisy - Chain
6- Priority - Out
7- Priority - in

## ۱۳-۴ ارتباط و همگامی بین پردازنده‌ها

در یک سیستم چند پردازنده‌ای باید پردازنده‌های متفاوتی با امکان ارتباط با یکدیگر فراهم شوند. مسیر ارتباط می‌تواند از طریق کانال های مشترک ورودی - خروجی ایجاد شود. در یک سیستم چند پردازنده با حافظه اشتراکی، متداولترین رویه، کنارگذاشتن بخشی از حافظه است که برای همه پردازنده‌ها قابل دسترسی است. هدف اصلی از بکارگیری این حافظه مشترک ایفای نقش یک مرکز پیام مشابه با یک صندوق پستی است که هر پردازنده می‌تواند پیامی برای سایر پردازنده‌ها باقی گذارده و پیام‌های موجود در آن را بردارد.

پردازنده فرستنده یک درخواست، پیام، یا یک رویه را ساخته و آن را درصندوق پستی حافظه قرار می‌دهد. بیت وضعیت واقع در حافظه مشترک معمولاً برای مشترک کردن حالت صندوق پستی، اینکه حاوی اطلاعات معنی دار است یا خیر، و اینکه پیام برای کدام پردازنده قرار داده شده است استفاده می‌شود. پردازنده گیرنده می‌تواند بصورت دوره‌ای صندوق پستی را برای یافتن پیام‌های معتبر وارسی کند. زمان پاسخ این رویه می‌تواند وقت گیر باشد، چون یک پردازنده فقط با همه پرسی پیام‌ها می‌تواند تقاضایی را شناسایی کند. رویه کارآکتر این است که پردازنده فرستنده با کمک سیگنال وقفه پردازنده گیرنده را آگاه کند. این عمل را می‌توان توسط یک وقفه بین پردازنده‌ای که بوسیله نرم‌افزار و با استفاده از دستورالعملی در برنامه پردازنده‌ها تولید می‌شود، انجام داد و وقتی اجرا شود وضعیتی مشابه با وقفه خارجی در پردازنده دوم بوجود می‌آورد. باین ترتیب پردازنده متوقف شده آگاه می‌شود که پیام جدیدی توسط پردازنده وقفه دهنده در صندوق پستی قرار داده شده است.

یک سیستم چند پردازنده علاوه بر حافظه مشترک ممکن است منابع مشترک دیگری هم داشته باشد. مثلاً یک دیسک مغناطیسی متصل به یک IOP ممکن است در اختیار همه CPUها قرار گیرد. با این ترتیب امکان استفاده مشترک از برنامه‌های سیستم ذخیره شده بر روی دیسک بوجود می‌آید. مسیر ارتباطی بین دو CPU ممکن است از طریق رابطی بین دو IOP متناظر با دو CPU ایجاد شده باشد. این نوع ارتباط اجازه می‌دهد تا هر CPU دیگری را بعنوان یک وسیله I/O در نظر گرفته و لذا پیام‌ها قابل انتقال از طریق مسیر I/O خواهد بود.

برای جلوگیری از ایجاد مزاحمت ناشی از استفاده منبع مشترک به وسیله چندین پردازنده، باید اولویتی در تخصیص منابع به پردازنده‌ها وجود داشته باشد. این وظیفه به سیستم عامل محول شده است. در طراحی سیستم عامل برای چند پردازنده‌ها، از سه سازمان استفاده شده است. آرایش حاکم و تابع[1]، سیستم عامل جداگانه و سیستم عامل توزیع شده.

در شیوه حاکم و تابع، همیشه یک پردازنده، که حاکم نامیده می‌شود، عملیات سیستم عامل را اجرا می‌کند. پردازنده‌های دیگر بمانند یک تابع عمل کرده و عملیات سیستم عامل را اجرا نمی‌کنند. اگر پردازنده تابع به سرویس سیستم عامل نیاز داشته باشد باید آن را با وقفه دادن به حاکم تقاضا کند و تا

---

1- Master Slave Mode

وقفه برنامه جاری منتظر بماند.

در سازمان مبتنی بر سیستم عامل جداگانه، هر پردازنده می‌تواند روالهایی از سیستم عامل راکه به آنها نیاز دارد اجرا کند. این سازمان برای سیستم‌های با کوپل سست که هر پردازنده ممکن است نسخه کامل از سیستم عامل خود را داشته باشد بیشتر مناسب است.

در سازمان مبتنی بر سیستم عامل توزیع شده، روال‌های سیستم عامل بین پردازنده های موجود توزیع شده‌اند. با این وجود ، هر تابع خاص از سیستم عامل در هر زمان فقط به یک پردازنده اختصاص داده می شود. این نوع سازمان دهی ، سیستم عامل شناور [1] خوانده می شود زیرا روال ها از یک پردازنده به پردازنده دیگر شناورند و ممکن است اجرای روال ها در زمان های مختلف به عهده پردازنده‌های مختلف باشد.

در یک سیستم چند پردازنده با کوپل سست حافظه بین پردازنده توزیع شده و حافظه مشترکی برای تبادل اطلاعات وجود ندارد، تبادل ارتباطات بین پردازنده‌هااز طریق کانال‌های I/O صورت می‌گیرد. این تبادل اطلاعات بوسیله یک پردازنده آغاز می‌شود و طی آن یک رویه واقع در حافظه، پردازنده‌ای که مایل به این ارتباط است را فرا می‌خواند. وقتی که پردازشگر فرستنده و پردازشگر گیرنده یکدیگر را بعنوان منبع و مقصد صدا کنند، یک کانال ارتباط برقرار می‌شود. سپس پیامی با سرآیند و داده‌های مختلف مورد استفاده برای ارتباط بین گره‌ها ارسال می‌شود. ممکن است مسیرهای متعددی برای ارسال پیام بین دو گره وجود داشته باشد. سیستم عامل موجود در هر گره حاوی اطلاعات مسیریابی که مسیرهای مختلف دیگری که می‌تواند برای ارسال پیام به سایر گره‌ها بکار رود را مشخص می‌کند. کارآرایی تبادل اطلاعات یک شبکه بین پردازنده‌ای به قرارداد مسیریابی ارتباطی، سرعت پردازنده، سرعت رابط داده‌ها، و توپولوژی شبکه بستگی دارد.

## همگامی بین پردازنده‌ها

مجموعه دستورالعمل‌های یک چند پردازنده حاوی دستورالعمل اصلی است که برای پیاده‌سازی ارتباط و همگامی بین پردازنده‌های همکار مورد استفاده قرار می‌گیرد. منظور از ارتباط، تبادل داده‌ها بین پردازنده‌هاست. مثلاً، پارامترهایی که به یک رویه در پردازنده دیگری انتقال می‌یابد یک تبادل بین پردازنده‌ای را می‌طلبد. همگامی به حالت خاصی که در آن داده‌های مورد استفاده در ارتباط بین پردازنده‌ها اطلاعات کنترلی هستند اتلاق می‌شود. همگامی برای ایجاد توالی صحیح در پردازش‌ها و نیز تضمین جلوگیری از مراجعه همزمان به داده‌های مشترک قابل نوشتن، مورد نیاز است.

سیستم‌های چند پردازنده‌ای معمولاً شامل مکانیزم‌های متعددی برای همگام‌سازی منابع هستند. عملیات پیش پا افتاده مستقیماً بوسیله سخت‌افزار پیاده‌سازی می‌شوند. این عملیات اساس مکانیزم‌هایی است که جدایی متقابل را برای مکانیزم‌های پیچیده‌تری که توسط نرم‌افزار پیاده‌سازی

---

1–Floating Operating System

می‌شوند ایجاد می‌کنند. یکی از متداولترین روش‌ها استفاده از سمافور دودویی [1] است.

### جداسازی متقابل با استفاده از سمافور

برای اینکه یک سیستم چند پردازنده درست کار کند باید مکانیزمی را فراهم سازد تا دستیابی متوالی و منظم به حافظه اشتراکی و دیگر منابع مشترک تضمین شود. این امر برای حفاظت داده‌ها از تغییر همزمان توسط دو یا چند پردازنده لازم است. این مکانیزم جداسازی متقابل [2] نامیده می‌شود. جداسازی متقابل باید در سیستم‌های چند پردازنده‌ای برای فعال کردن یک پردازنده فراهم شود تا وقتی که آن پردازنده با یکی از منابع مشترک در یک بخش بحرانی مشغول انجام کار است مانع دستیابی دیگر پردازنده‌ها به آن منبع گردد. بخش بحرانی، رشته‌ای از دستورات برنامه است که اگر اجرای آن شروع شود نباید قبل از اتمام اجرای آن هیچ پردازنده دیگری به آن منبع مشترک دست یابد.

یک متغیر دودویی بنام سمافور اغلب برای مشخص کردن اینکه پردازنده‌ای یک بخش بحرانی را اجرا می‌کند یا نه بکار می‌رود. سمافور یک پرچم کنترل شده با نرم افزار است که در مکانی از حافظه که قابل دسترسی بوسیله تمام پردازنده‌ها است قرار داده می‌شود. اگر سمافور 1 باشد، مفهوم این است که پردازنده‌ای یک بخش بحرانی را اجرا می‌کند و لذا حافظه مشترک برای پردازنده‌های دیگر مهیا نیست. وقتی که سمافور 0 باشد، حافظه مشترک برای هر پردازنده متقاضی در دسترس است. پردازنده‌هایی که از یک قطعه حافظه بطور مشترک استفاده می‌کنند بنابه قرارداد توافق دارند که از آن قطعه استفاده نکنند مگر اینکه سمافور برابر 0 باشد، که نشان دهنده این است که حافظه قابل دسترسی است. همچنین، پردازنده‌ها توافق دارند تا بهنگام اجرای یک بخش بحرانی سمافور را 1 و در پایان آن را 0 کنند.

تست و 1 کردن سمافور خود یک عمل بحرانی است و باید به صورت یک عمل منفرد غیر قابل تقسیم انجام شود. در غیر اینصورت، دو یا چند پردازنده ممکن است بطور همزمان سمافور را تست کرده و هر یک جداگانه آن را 1 کنند و اجازه یابند تا باتفاق وارد بخش بحرانی گردند. این عمل سبب می‌شود تا اجرای همزمان بخش بحرانی بوجود آید که می‌تواند منجر به مقداردهی غلط پارامترهای کنترلی و از دست رفتن اطلاعات اساسی گردد.

سمافور را می‌توان با استفاده از دستورالعمل تست و یک مکانیزم قفل سخت‌افزاری مقداردهی اولیه نمود. قفل سخت‌افزاری یک سیگنال تولید شده بوسیله پردازنده است که از بکارگیری گذرگاه توسط پردازنده‌های دیگر تا زمانی که این سیگنال فعال است ممانعت می‌کند. دستورالعمل تست و یک کننده سمافور را 1 می‌کند و مکانیزم قفل را در حین اجرای دستورالعمل فعال می‌سازد. این عمل سبب می‌شود تا سایر پردازنده‌ها در فاصله زمانی اجرای عمل تست و 1 کردن سمافور قادر به تغییر آن نباشند. فرض کنید که سمافور بیت کم ارزشتر مکانی از یک کلمه حافظه است که آدرس آن با SEM نشان داده شده باشد. اجازه بدهید سمبل TSL بمعنی عمل "تست و 1 شدن در حین قفل شدن" باشد.

---

-1- Binary Semaphore 2-Mutual Exclusion

دستورالعمل

TSL SEM

در دو سیکل حافظه بدون تداخل اجرا می‌شود (اولین سیکل برای خواندن و دوم برای نوشتن)

| | |
|---|---|
| تست سمافور | R ← M[SEM] |
| 1کردن سمافور | M[SEM] ← 1 |

سمافور با انتقال مقدارش به ثبات R، در پردازنده تست شده و سپس مقدار 1 را می‌گیرد. مقدار واقع در R کار بعدی را معین می‌سازد. اگر پردازنده R=1 را بیابد، خواهد دانست که R سمافور از قبل 1 بوده است. (دقت کنید که دوباره 1شدن R مقدار سمافور را عوض نمی‌کند). این بدان معنی است که پردازنده دیگری مشغول اجرای بخش بحرانی است، بنابراین پردازنده‌ای که سمافور را چک کرده است به حافظه مشترک دست نمی‌یابد. اگر R=0 باشد، یعنی حافظه مشترک مهیا است (یا منبع مشترکی که سمافور نماینده آن است). سمافور برابر 1 می‌شود تا از دستیابی پردازنده‌های دیگر به حافظه جلوگیری شود. اکنون پردازنده می‌تواند بخش بحرانی را اجرا کند. آخرین دستورالعمل برنامه باید خانه SEM را 0کند تا منبع مشترک قابل دسترسی پردازنده‌های دیگر گردد.

توجه کنید که در حین اجرای دستورالعمل تست و 1کردن، سیگنال قفل باید فعال باشد. وقتی که سمافور 1 شد دیگر لازم نیست قفل فعال باقی بماند. باین ترتیب مکانیزم قفل از دستیابی پردازنده‌های دیگر به حافظه در حالی که سمافور در حال 1 شدن است جلوگیری می‌کند. خود سمافور، وقتی که 1 شود، از دستیابی پردازنده‌های دیگر به حافظه مشترک در حالی که پردازنده‌ای مشغول اجرای یک بخش بحرانی است جلوگیری می‌کند.

## ۱۳-۵ همبستگی حافظه کش

عملکرد حافظه کش در بخش ۶-۱۲ تشریح شد. مزیت عمده حافظه کش قابلیت کاهش میانگین زمان دستیابی در تک پردازنده‌هاست. هر گاه پردازنده‌ای در عمل خواندن، کلمه مورد نظری را در کش بیابد، حافظه اصلی در عمل انتقال دخالت نمی‌کند. اگر عمل نوشتن باشد، دو رویه متداول برای بهنگام کردن حافظه وجود دارد. در سیاست کامل نویسی، هر دو بخش حافظه کش و حافظه اصلی با هر عمل نوشتن بهنگام می‌شود. در سیاست پس نویسی، فقط حافظه کش بهنگام می‌شود و خانه موردنظر علامت‌گذاری می‌شود تا بعداً در حافظه اصلی کپی شود.

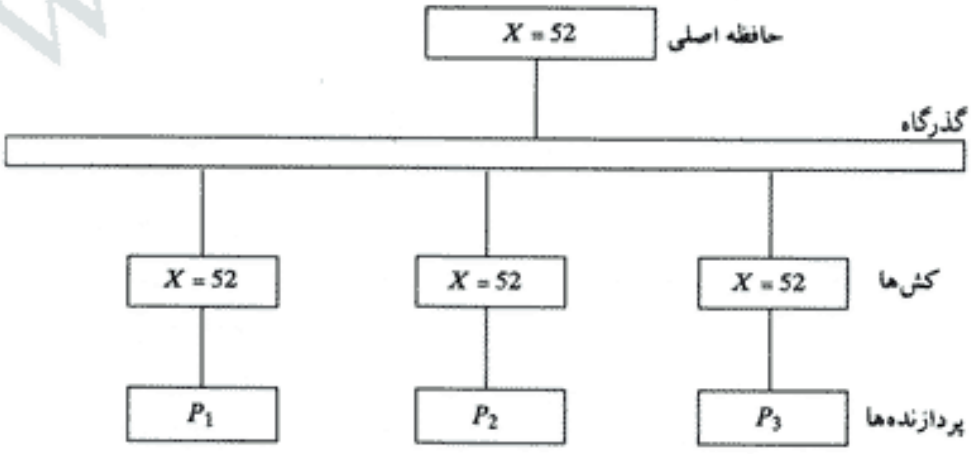
در یک سیستم چند پردازنده با حافظه مشترک، همه پردازنده‌ها از حافظه مشترک استفاده می‌کنند. بعلاوه هر پردازنده ممکن است یک حافظه محلی هم داشته باشد، که یک بخش یا تمام آن می‌تواند کش باشد. دلیل داشتن اجباری حافظه کش جداگانه برای هر پردازنده، کاهش میانگین زمان دستیابی در آن

است. البته اطلاعات یکسانی ممکن است در نسخه‌های متعدد در برخی حافظه کش و حافظه اصلی قرار داده شود. برای اطمینان از توانایی در اجرای صحیح عملیات حافظه، نسخه‌های متعدد باید یکسان نگهداشته شوند. این نیاز، مسئله همبستگی حافظه کش[1] را بوجود می‌آورد. یک سیستم حافظه همبسته است اگر مقداری که یک دستورالعمل باردهی برمی‌گرداند همیشه مقداری باشد که با آخرین دستور ذخیره با همان آدرس داده شده است. بدون راه حل مناسبی برای مسئله همبستگی در حافظه کَش نمی‌توان از این نوع حافظه در چند پردازنده‌ها مبتنی بر گذرگاه با دو یا چند پردازنده استفاده کرد.

## شرایط عدم بستگی

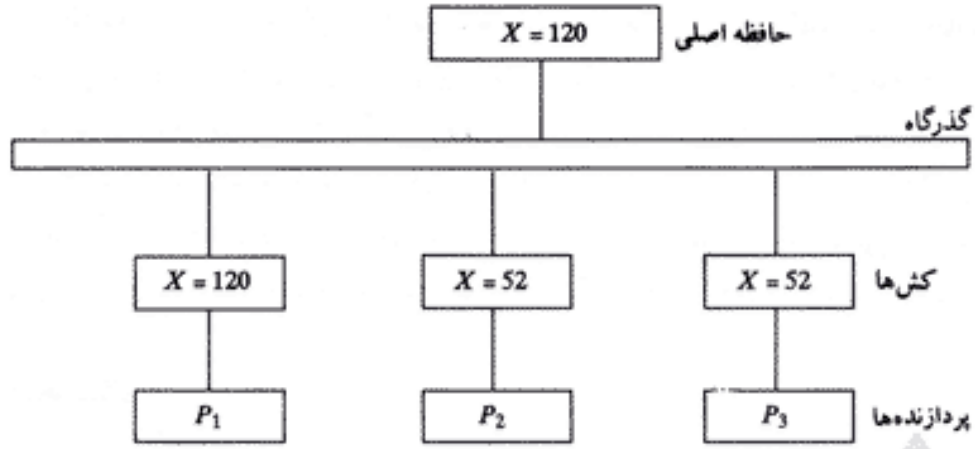
مشکلات همبستگی حافظه کش در چند پردازنده‌هایی که بدلیل نیاز به داده‌های مشترک قابل نوشتن دارای حافظه‌های اختصاص می‌باشند بوجود می‌آید. داده‌های فقط خواندنی را می‌توان با اطمینان و بدون مکانیزم‌های ایجاد کننده همبستگی حافظه کَش تکثیر کرد. برای روشن شدن مسئله، آرایش سه پردازنده با حافظه‌هایی کش اختصاصی در شکل ۱۲-۱۳ را در نظر بگیرید. گاهی در حین کار سیستم، عنصری مثل X از حافظه اصلی در سه پردازنده $P_1$، $P_2$، $P_3$ بار می‌شود. در نتیجه، این عنصر در حافظه‌های اختصاصی سه پردازنده نیز کپی می‌شود. برای سادگی، ما فرض خواهیم کرد که X حاوی مقدار 52 باشد، بار شدن X در سه پردازنده منجر به نسخه‌های یکسان در حافظه‌های کش و حافظه اصلی می‌گردد.

اگر یکی از پردازنده عمل ذخیره در X را انجام دهد، نسخه‌های X در حافظه‌های کش تغییر می‌کنند. بارکردن بوسیله پردازنده‌های دیگر آخرین مقدار را برنمی‌گرداند. بسته به سیاست بهنگام‌سازی حافظه در سیستم حافظه کش، ممکن است محتوای حافظه اصلی هم با حافظه کش یکی نباشد. این مطلب در شکل ۱۳-۱۳ نشان داده شده است. یک عمل ذخیره در X با مقدار 120 در حافظه کش پردازنده $P_1$ در

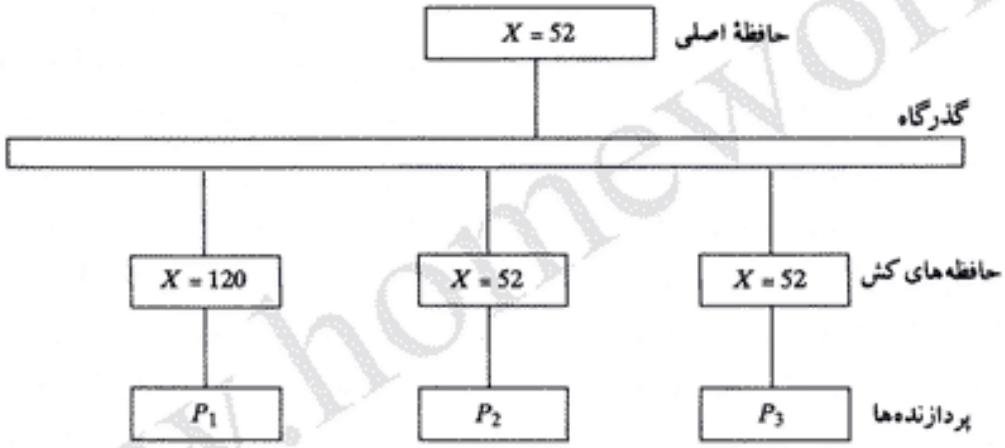



**شکل ۱۲-۱۳** آرایش کش پس از بار کردن در X

1- Cache Coherence




(الف) با استفاده از سیاست کامل‌نویسی




(ب) با استفاده از سیاست پس‌نویسی

**شکل ۱۳-۱۳** آرایش کش پس از ذخیره کردن در X بوسیله پردازنده $P_1$

سیاست کامل نویسی، حافظه را به مقدار جدید بهنگام می‌کند. سیاست کامل نویسی هماهنگی بین حافظه اصلی و حافظه کش مبداء تغییر را حفظ می‌کند، اما دو حافظه کش دیگر ناهماهنگ خواهند بود زیرا هنوز حاوی مقدار قبلی هستند. در سیاست پس نویسی، حافظه اصلی در هنگام ذخیره بهنگام نمی‌شود. نسخه‌های موجود در دو حافظه کش دیگر و حافظه اصلی متفاوت خواهند بود. در نهایت حافظه اصلی وقتی که داده‌های تغییر داده شده موجود در حافظه کش در آن کپی شوند بهنگام می‌شود.

یک پیکربندی دیگر که ممکن است موجب مشکلات مربوط به ناهماهنگی شود عمل دستیابی مستقیم به حافظه DMA همراه با یک IOP متصل به گذرگاه سیستم است. در حالت ورودی، DMA ممکن است خانه‌هایی از حافظه اصلی راکه در حافظه کش نیز مقیم‌اند تغییر دهد بدون آنکه حافظه کش را بهنگام نماید. در حین عمل خروجی DMA، در صورت استفاده از سیاست پس نویسی، ممکن است مکان‌هایی از حافظه اصلی قبل از اینکه روی حافظه کش بهنگام گردند، خوانده شوند. با شرکت دادن

IOP در حل مشکل همبستگی حافظه کش که در سیستم بکار گرفته می‌شود می‌توان برنا همبستگی مبتنی بر I/O غلبه کرد.

### راه حل‌هایی برای حل مسئله همبستگی حافظه کش

روش‌های متعددی برای حل مسئله همبستگی حافظه کش در چند پردازنده‌های با حافظه مشترک پیشنهاد شده است. ما در اینجا برخی از این روش‌ها را بطور مختصر بحث می‌کنیم. برای بحث‌های مشروح‌تر به مراجع 3 و 10 مراجعه نمایید.

یک روش ساده عبارتست از عدم اجازه به وجود حافظه‌های کش اختصاصی برای هر پردازنده و داشتن یک حافظه کش مشترک همراه با حافظه اصلی است . هر دستیابی داده به حافظه کش اشتراکی صورت می‌گیرد. این روش اصل نزدیک بودن CPU را به حافظه کش نقض می‌کند و میانگین زمان دستیابی را افزایش می‌دهد. در واقع این روش مشکل را با اجتناب از آن حل می‌نماید.

از دیدگاه ملاحظات مربوط به عملکرد، بهتر است که یک حافظه کش در هر پردازنده وجود داشته باشد. یکی از روشهای بکار رفته فقط اجازه ذخیره داده‌های غیرمشترک و فقط خواندنی را در حافظه‌های کش می‌دهد. چنین داده‌هائی قابل کش شدن[1] خوانده می‌شوند. داده‌های قابل نوشتن مشترک قابلیت ذخیره شدن در کش را ندارند.[2] کامپایلر باید داده‌ها را از نظر قابل ذخیره شدن در کش و نشدن عنوان‌گذاری کند، و سخت افزار سیستم طوری عمل خواهد کرد که فقط داده‌های قابل ذخیره در کش ذخیره شوند. داده‌هایی که در کش ذخیره نمی‌شوند در حافظه اصلی باقی می‌مانند. این روش نوع داده‌های ذخیره شده در حافظه‌ای کش را محدود کرده و یک سربار اضافی را که ممکن است موجب افت عملکرد گردد ایجاد می‌نماید.

روشی که اجازه وجود داده‌های قابل نوشتن در حداقل یک حافظه کش را می‌دهد، روشی است که از یک جدول سراسری متمرکز در برنامه کامپایلر خود استفاده می‌کند. وضعیت بلاک‌های حافظه در جدول سراسری مرکزی ذخیره می‌شوند. هربلاک به عنوان فقط خواندنی (RO) یا خواندنی - نوشتنی (RW) مشخص می‌گردد.

تمام حافظه‌های کش می‌توانند یک کپی از بلاک‌ها راکه به عنوان RO مشخص شده‌اند داشته باشند. بنابراین اگر داده‌ها در حافظه کشی که حاوی بلاک RW است بهنگام شود، دیگر حافظه‌های کش بعلت نداشتن کپی‌هایی از این بلاک تحت تأثیر قرار نمی‌گیرند. مسئله همبستگی حافظه کش می‌تواند با استفاده از ترکیبی از نرم‌افزار، و سخت افزار یا با استفاده از روشهای صرفاً سخت افزاری حل شود. دو روشی که قبلاً ذکر شد از رویه‌های مبتنی بر نرم افزار استفاده می‌نمایند که قابلیت نشانه‌گذاری برای جلوگیری از قرارگرفتن داده‌های قابل نوشتن در حافظه کش دارند. در راه حل‌های سخت افزاری، کنترل کننده کش بطور خاصی طراحی می‌شود تا بتواند بر همه درخواست‌های گذرگاه از سوی CPU ها و IOP ها

---

1- Cachable 2- Nonchachable

نظارت داشته باشد. همه حافظه کش متصل به گذرگاه بطور مستمر بر شبکه‌برای یافتن یک عمل احتمالی نوشتن نظارت می‌کنند. این حافظه‌های کش باید در صورت کشف انطباق، بسته به روش مورد استفاده کپی‌های موجود خود را بهنگام و یاغیر معتبر نمایند. کنترل کننده گذرگاهی که بر این عمل نظارت می‌کند کنترل کننده مداخله گر [1] نامیده می‌شود. این کنترل کننده اساساً یک واحد سخت افزاری است که برای برقراری یک مکانیزم نظارت بر گذرگاه در همه حافظه‌های نهان متصل به گذرگاه طراحی می‌شود.

روش‌های مختلفی برای حل مسئله همبستگی حافظه نهان با استفاده از قرارداد کنترل کننده مداخله گر حافظه کش پیشنهاد شده است ساده‌ترین متد استفاده از سیاست کامل نویسی و استفاده از رویه زیر است. تمام کنترل‌کننده‌های مداخله گر برای یافتن عملیات ذخیره حافظه بر گذرگاه نظارت می‌کنند. هر گاه کلمه‌ای با نوشته شدن در یک حافظه کش بهنگام شود، مکان متناظر آن در حافظه اصلی نیز بهنگام می‌شود. کنترل کننده‌های مداخله گر محلی مربوط به کلیه حافظه‌های کش دیگر، حافظه خود را وارسی می‌کنند تا مشخص کنند کپی کلمه‌ای راکه روی آن نوشته‌اند دارند یاخیر. اگر کپی در حافظه کش دیگر وجود داشته باشد، آن مکان بعنوان نامعتبر علامت گذاری می‌شود. چون همه حافظه کش همه عمل‌های نوشتن گذرگاه را زیر نظر دارند، هرگاه کلمه‌ای نوشته شود، اثر خالص این عمل بهنگام شدن آن در حافظه کش و حافظه اصلی و حذف آن از همه حافظه‌های کش دیگر خواهد بود. اگر در آینده، پردازنده‌ای به داده نامعتبر واقع در حافظه کش خود مراجعه نماید، پاسخی که دریافت می‌کند باخت در حافظه کش خواهد بود، و شکل بهنگام آن داده از حافظه اصلی آورده می‌شود. باین ترتیب از ایجاد کپی‌های متفاوت با کپی‌های جدید داده جلوگیری می‌گردد.

## مسائل

۱۳-۱ اختلاف بین چند پردازنده با کوپل محکم و چند پردازنده باکوپل سست را از نقطه نظر سازمان سخت‌افزار و تکنیک‌های برنامه‌نویسی بحث کنید.

۱۳-۲ هدف از کنترل کننده گذرگاه سیستم در شکل ۱۳-۲ چیست؟ توضیح دهید که چگونه می‌توان سیستم را به صورتی طراحی کرد که ارجاع به حافظه محلی و ارجاع به حافظه مشترک را از هم تشخیص دهد.

۱۳-۳ در یک شبکه سوئیچ تقاطعی که p پردازنده را به m ماژول حافظه متصل می‌کند چند نقطه سوئیچ وجود دارد.

۱۳-۴ شبکه سوئیچینگ امگای 8x8 در شکل ۱۳-۸ سه طبقه و چهار سوئیچ در هر طبقه و جمعاً 12 سوئیچ دارد. برای یک شبکه سوئیچینگ nxn، چند طبقه و چند سوئیچ در هر طبقه لازم است.

۱۳-۵ فرض کنید که در شبکه سوئیچینگ امگای شکل ۱۳-۸ سیم بین سوئیچ واقع در ردیف اول و ستون دوم و سوئیچ واقع در ردیف دوم ستون سوم قطع شود. چه مسیرهایی مسدود می‌شود؟

---

1- Snoopy Cache Controller

۱۳-۶ نموداری را برای یک شبکه سوئیچینگ امگای 4x4 بسازید. تنظیم‌های سوئیچ‌ها را برای اتصال ورودی 3 به خروجی 1 نشان دهید.

۱۳-۷ برای طراحی یک شبکه ارتباطات چند طبقه از سه نوع سوئیچ استفاده شده است: سوئیچ تبادلی[1] با دو ورودی و دو خروجی طبق شکل ۶-۱۳، یک سوئیچ داوری با دو ورودی و یک خروجی، و سوئیچ توزیع با یک ورودی و دو خروجی

الف) نحوه عملکرد سوئیچ‌های داوری و توزیع را نشان دهید.

ب) با استفاده از سوئیچ‌های داوری و تبادلی، یک شبکه 8x4 با یک مسیر منحصر بفرد بین هر مبدأ و هر مقصد بسازید.

ج) با استفاده از سوئیچ‌های توزیع و تبادلی، یک شبکه 4x8 با یک مسیر منحصر بفرد بین هر مبدأ و هر مقصد بسازید.

۱۳-۸ دیاگرامی رسم کنید که ساختار یک شبکه فوق مکعبی چهار بعدی را نشان دهد. همه مسیرهای موجود از گره 7 به گره 9 را که از حداقل تعداد گره‌های میانی استفاده می‌کنند مشخص کنید.

۱۳-۹ یک نمودار منطقی با استفاده از گیت و فلیپ فلاپ رسم کنید که مدار یک طبقه داور گذرگاه رادر طرح داوری زنجیره‌ای شکل ۱۰-۱۳ نشان دهد.

۱۳-۱۰ مدار تحت کنترل مدار داوری موازی در شکل ۱۱-۱۳ در ابتدا بکار است. سپس وسیله‌های 2 و 3 بطور همزمان گذرگاه را درخواست می‌کنند. مفادیر دودویی ورودی و خروجی را در انکدر و دیکدر مشخص نمایند و مشخص کنید به کدام داور گذرگاه سیگنال تصدیق داده می‌شود.

۱۳-۱۱ نشان دهید که چگونه مدار داوری شکل ۱۰-۱۳ را می‌توان برای فراهم نمودن یک رویه داوری زنجیره‌ای بسته اصلاح کرد. توضیح دهید که وقتی خط گذرگاه غیرفعال است چگونه اولویت تعیین می‌شود.

۱۳-۱۲ یک توپولوژی گذرگاه که در آن دوپردازنده از طریق یک بافر در حافظه مشترک با هم تبادل اطلاعات می‌کنند را در نظر بگیرید. اگر یکی از پردازنده‌ها بخواهد با پردازنده دیگری ارتباط برقرار کند، اطلاعات را در بافر حافظه قرار می‌دهد و پرچمی را 1 می‌کند. پردازنده دیگر بطور متناوب پرچم‌ها را وارسی می‌کند تا معلوم کند که اطلاعاتی برای دریافت دارد یا خیر. برای اطمینان از همگامی صحیح و به حداقل رساندن زمان بین ارسال و دریافت اطلاعات چه کاری می‌توان انجام داد؟

۱۳-۱۳ اصطلاحات زیر را در رابطه با چند پردازنده شرح دهید. (الف) جداسازی متقابل، (ب) بخش بحرانی، (ج) قفل سخت افزاری، (د) سمافور، (ه) دستورالعمل تست و 1 نمودن.

۱۳-۱۴ همبستگی حافظه کش چیست، و چرا در سیستم‌های چند پردازنده با حافظه مشترک مهم است؟ چگونه می‌توان این مسئله را با یک کنترل کننده مداخله‌گر حافظه نهان حل کرد؟

---

1- Interchange Switch