فصل هفتم

ورودی / خروجی

۷-۱ **دستگاه‌های خارجی**
صفحه‌کلید / مانیتور — دیسک گردان

۷-۲ **مدول‌های I/O**
وظایف مدول — ساختار مدول I/O

۷-۳ **I/O برنامه‌نویسی شده**
مروری بر I/O برنامه‌نویسی شده
فرامین I/O — دستورات I/O

۷-۴ **I/O تحت وقفه**
پردازش وقفه — نکات طراحی
کنترلگر وقفه‌ی Intel 82C59A
واسط جانبی برنامه‌پذیر Intel 82C55A (PPI)

۷-۵ **دستیابی مستقیم به حافظه**
نقاط ضعف I/O برنامه‌نویسی شده و تحت وقفه
دستیابی مستقیم به حافظه (DMA)
کنترلگر 8237 A DMA اینتل

۷-۶ **کانال‌های I/O و پردازنده‌ها**
توسعه‌ی عملکرد I/O
مشخصه‌های کانال I/O

۷-۷ **واسط خارجی: FIREWIRE و INFINIBAND**
انواع واسطها
آرایش‌های نقطه به نقطه و چندنقطه‌ای
Thunderbolt
InfiniBand

۷-۸ **ساختار Enterprise 196 I/O از IBM**

۷-۹ **خواندنی‌های پیشنهادی**

۷-۱۰ **واژه‌های کلیدی، پرسش‌های مرور درس، و مسائل**

اهداف آموختنی

پس از مطالعه‌ی این فصل شما قادر خواهید بود:

- مدول‌های I/O را به عنوان بخشی از سازمان کامپیوتر توضیح دهید.
- اختلاف بین I/O برنامه‌نویسی شده و I/O تحت وقفه را بدانید و ارزش نسبی آن‌ها را بحث نمایید.
- مروری بر عملکرد دستیابی مستقیم به حافظه را ارائه نمایید.
- کار و استفاده از کانال‌های I/O را توضیح دهید.
- مروری بر Thunderbolt را ارائه کنید.
- مروری بر InfiniBand داشته باشید.

ابزار طراحی سیستم I/O

علاوه بر پردازنده و مجموعه‌ای از مدول‌های حافظه، سومین عنصر کلیدی یک سیستم کامپیوتری مجموعه‌ای از مدول‌های I/O است. هر مدول به گذرگاه سیستم یا سوئیچ مرکزی وصل می‌گردد و یک یا چند وسیله‌ی خارجی را کنترل می‌نماید. یک مدول I/O صرفاً مجموعه‌ای از اتصالات مکانیکی برای ارتباط دستگاه به گذرگاه سیستم نیست بلکه مدول I/O مقداری هوشمندی دارد، یعنی حاوی مداری برای اجرای اعمال تبادلی و مخابره بین وسیله‌ی جانبی و گذرگاه است.

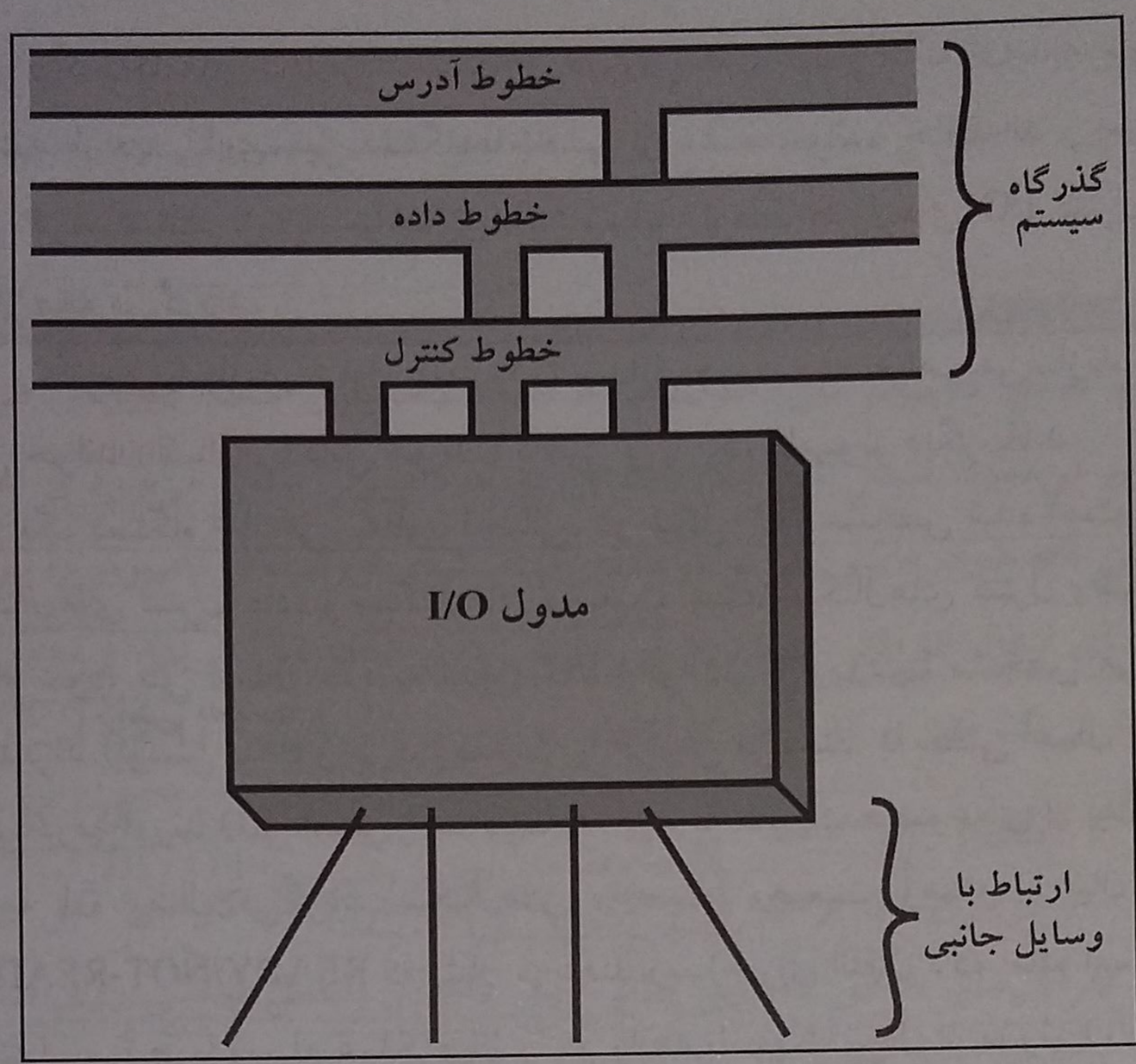
خواننده ممکن‌است این سؤال را مطرح کند که چرا وسایل جانبی مستقیماً به گذرگاه سیستم وصل نمی‌شوند. دلایل به شرح زیر می‌باشند:

- انواع مختلف و متعددی از وسایل جانبی با عملکردهای متفاوتی وجود دارند، لذا لحاظ مدارهای متعدد منطقی لازم در یک پردازنده برای کنترل انواع دستگاه‌ها غیرعملی است.
- سرعت انتقال داده از وسایل جانبی اغلب خیلی کُندتر از حافظه یا پردازنده است. بنابراین استفاده از یک گذرگاه سیستم سریع برای تبادل داده‌ی مستقیم با وسایل جانبی غیرعملی است.
- از طرف دیگر، سرعت انتقال داده‌ی بعضی از وسایل جانبی سریع‌تر از حافظه یا پردازنده است. مجدداً اگر مدیریت صحیحی اجرا نشود عدم تطابق منجر به پایین‌آمدن راندمان می‌گردد. باز هم اگر این عدم تطابق به درستی مدیریت نشود، عدم بازدهی مطلوب را به دنبال خواهد داشت.
- وسایل جانبی اغلب فرمت داده‌ی متفاوت و طول کلمه‌ی مختلفی را با کامپیوتری که به آن وصلند اختیار می‌کنند.

بنابراین یک مدول I/O مورد نیاز است. این مدول دو وظیفه‌ی عمده دارد (شکل ۱-۷):

- اتصال به پردازنده و حافظه از طریق گذرگاه سیستم یا سوئیچ مرکزی.
- اتصال به یک یا چند دستگاه جانبی با اتصالات داده‌ی سازگار.

ما این فصل را با بحثی کوتاه از وسایل خارجی آغاز می‌کنیم و سپس مروری بر ساختار و کار یک مدول I/O خواهیم داشت. آن‌گاه به روش‌های مختلف اجرای کار I/O در کنار پردازنده و حافظه



شکل ۱-۷ مدل عمومی یک مدول I/O.

می‌پردازیم. بالاخره واسط داخلی و سرانجام اتصال I/O خارجی (بیرونی) بین مدول I/O و دنیای خارج را ملاحظه خواهیم کرد.

۱-۷ دستگاه‌های خارجی

اعمال I/O از طریق وسایل طبقه‌بندی‌شده‌ی خارجی که نقش تبادل داده بین کامپیوتر و محیط خارج را به عهده دارند، صورت می‌گیرد. یک وسیله‌ی خارجی به وسیله‌ی یک مدول I/O به کامپیوتر وصل می‌شود (شکل ۱-۷). اتصال به منظور تبادل کنترل، وضعیت و داده بین مدول I/O و وسیله‌ی خارجی انجام می‌گیرد. وسیله‌ی خارجی متصل به مدول I/O اغلب به *وسیله‌ی جانبی* معروف است.

می‌توانیم وسایل جانبی را به سه دسته تقسیم کنیم:

- **قابل خواندن توسط انسان**[1]: مناسب برای تبادل با کاربر کامپیوتر.
- **قابل خواندن توسط ماشین**[2]: مناسب برای تبادل با تجهیزات.
- **مخابره**: برای مبادله با وسایل دوردست.

مثال‌هایی از وسایل قابل خواندن انسان عبارتنداز: پایانه‌های ویدیویی (VDT) و چاپگرها. مثال‌هایی از وسایل قابل خواندن ماشین عبارتنداز: دیسک مغناطیسی، سیستم‌های نوار، سنسورها و محرک‌ها[3]، همانند آن‌چه در کاربردهای رباتیک استفاده می‌شوند. توجه کنید که ما سیستم‌های دیسک و نوار را در

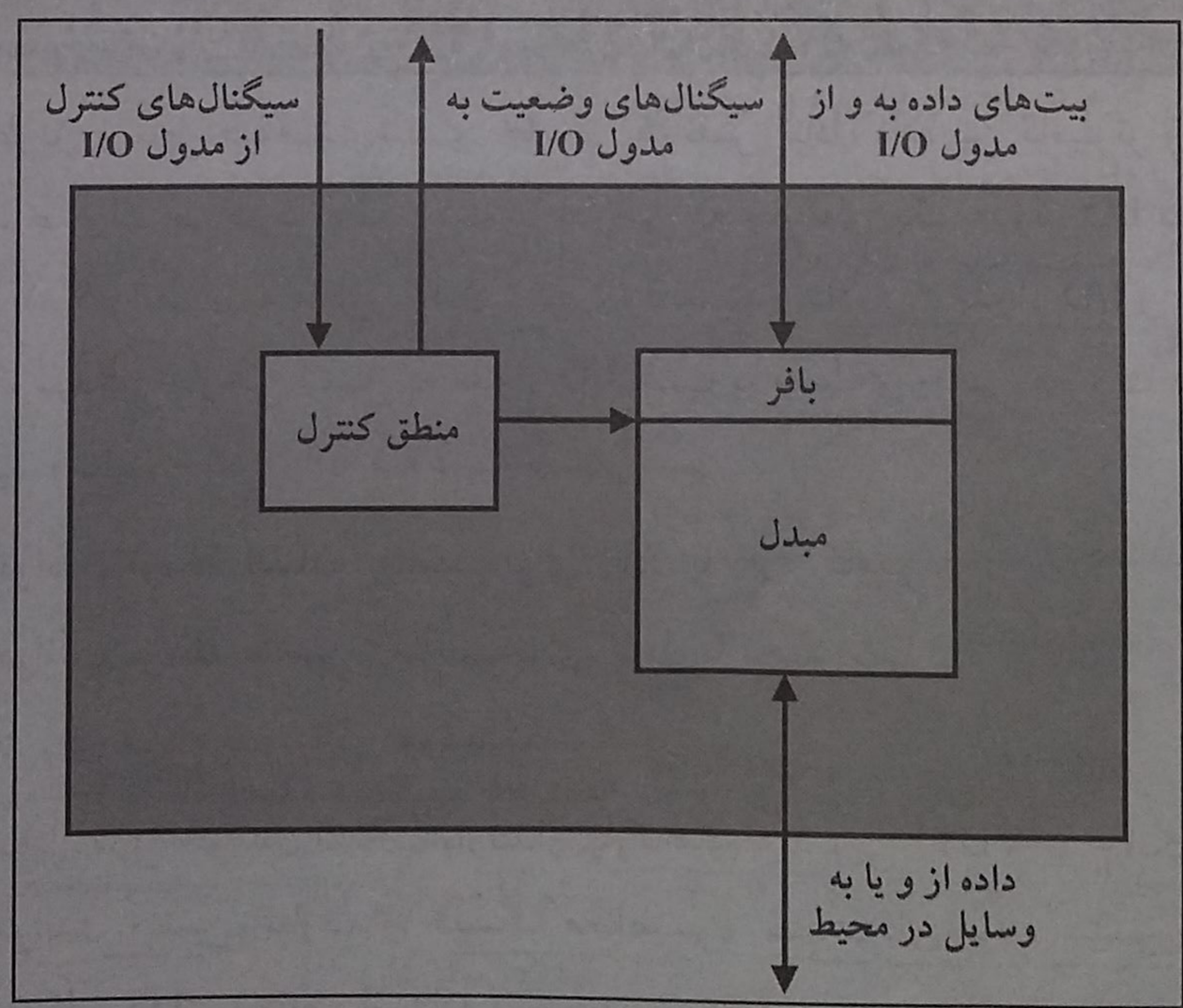
1. Human readable 2. Machine readable 3. Actuator

این فصل به عنوان دستگاه‌های I/O می‌شناسیم، در صورتی که در فصل ۶ به آن‌ها به عنوان دستگاه‌های حافظه‌ای نگاه شد. از دید کاری، این دستگاه‌ها بخشی از سلسله مراتب حافظه‌اند و استفاده از آن‌ها به خوبی در فصل ۶ بحث شد. از دید ساختاری، این وسایل توسط مدول‌های I/O کنترل می‌شوند و لذا در این فصل ملاحظه می‌گردند.

وسایل مخابره‌ای اجازه‌ی تبادل داده را با وسایل دوردست[1] فراهم می‌سازند، که ممکن‌است یک وسیله قابل خواندن انسان، یا قابل خواندن ماشین و یا یک کامپیوتر دیگر باشد.

ساختار یک دستگاه خارجی، به‌طور اجمالی، در شکل ۲-۷ مشخص شده است. اتصال به مدول I/O به فرم *سیگنال‌های کنترل*، داده و سیگنال‌های وضعیت است. سیگنال‌های کنترل وظیفه‌ی وسیله را که قرار است انجام شود، مثل ارسال داده به مدول I/O (خواندن یا ورودی) مشخص می‌کنند، داده را از مدول I/O می‌پذیرند (نوشتن یا خروجی)، وضعیت را گزارش می‌دهند، یا بعضی اعمال کنترل خاص یک وسیله را به اجرا درمی‌آورند (مثل استقرار هد دیسک). *داده* به صورت مجموعه‌ای از بیت‌ها از مدول I/O دریافت و یا به آن ارسال می‌گردد. *سیگنال‌های وضعیت*، وضعیت وسیله را بیان می‌کنند. مثال‌ها عبارت‌انداز READY/NOT-READY که نشان می‌دهند وسیله برای انتقال داده آماده است یا خیر.

مدار کنترل مرتبط با وسیله عملکرد آن را در رابطه با جهت تبادل از مدول I/O کنترل می‌نماید. مبدلی[2] داده را از فرم الکتریکی به فرم‌های دیگر انرژی در حین خروجی و از دیگر فرم‌ها به نوع الکتریکی در حین ورود تبدیل می‌کند. معمولاً بافری برای نگهداری داده‌ی مورد انتقال بین مدول I/O و محیط خارج، در مبدل وجود دارد. اندازه‌ی رایج بین 8 الی 16 بیت است.



شکل ۲-۷ نمودار بلوکی یک دستگاه خارجی.

1. Remote device 2. Transducer

واسط بین مدول I/O و وسیله‌ی خارجی در بخش ۷-۷ مورد بررسی قرار گرفته است. بررسی واسط بین وسیله‌ی خارجی و محیط، خارج از دیدگاه این کتاب است ولی چند مثال ساده در این‌جا ارائه شده است.

صفحه‌کلید/مانیتور

رایج‌ترین مفهوم تعامل کامپیوتر/کاربر مجموعه صفحه‌کلید/مانیتور است. کاربر از طریق صفحه‌کلید ورودی را فراهم می‌سازد. سپس این اطلاعات وارده به کامپیوتر منتقل می‌شود و ممکن‌است روی مانیتور نیز نمایش داده شود. مانیتور می‌تواند داده‌ی فراهم‌شده به وسیله‌ی کامپیوتر را نیز نمایش دهد.

واحد مبنای انتقال کاراکتر[1] است. به هر کاراکتر کدی تخصیص می‌یابد و معمولاً 7 الی 8 بیت طول دارد. رایج‌ترین کد متن به کار رفته کد الفبای عددی مرجع جهانی[2] (IRA) است. هر کاراکتر در این کد با یک کد دودویی منحصربه‌فرد 7 بیتی نمایش داده می‌شود. بنابراین 128 کاراکتر متفاوت را می‌توان نشان داد. کاراکترها دو نوعند: نوع قابل چاپ و کنترل. کاراکترهای قابل چاپ عبارتنداز: الفبایی، اعداد و کاراکترهای خاص که قابل چاپ روی کاغذ و یا قابل نمایش روی صفحه نمایش‌اند. بعضی از علائم یا کاراکترهای کنترل در چاپ یا نمایش کاراکترها نقش دارند، که از آن جمله علامت بازگشت نورد است. دیگر کاراکترهای کنترلی در روال‌های مخابره مورد استفاده قرار می‌گیرند. پیوست مربوطه ملاحظه شود.

برای ورودی صفحه‌کلید، وقتی که کاربر کلیدی را می‌فشارد، سیگنالی الکترونیکی تولید می‌شود که به وسیله‌ی مبدل داخل صفحه‌کلید تفسیر شده و به الگوی بیتی متناظر با کد IRA ترجمه یا تبدیل می‌گردد. سپس این الگوی بیتی به مدول I/O انتقال می‌یابد. در کامپیوتر متن می‌تواند با همان کد ذخیره شود. در خروجی کاراکترهای کد IRA از مدول I/O به یک وسیله‌ی خارجی انتقال می‌یابند. مبدل داخل دستگاه خارجی، این کدها را تفسیر کرده سیگنال‌های الکترونیکی لازم را به دستگاه خارجی برای نمایش کاراکتر مربوطه یا انجام عمل کنترل تقاضا شده ارسال می‌دارد.

دیسک گردان

دیسک‌گردان یا گرداننده‌ی دیسک شامل مجموعه‌ای الکترونیکی برای تبادل داده، کنترل و سیگنال‌های وضعیت با یک مدول I/O و الکترونیکی برای کنترل راهکار خواندن و نوشتن است. در دیسک با هد ثابت، مبدل قادر است الگوهای مغناطیسی حاصل از سطح دیسک متحرک را به بیت‌هایی در بافر دستگاه تبدیل کند (شکل ۲-۷). دیسکی با هد متحرک هم باید بتواند بازوی دیسک را در راستای شعاع در روی صفحه‌ی دیسک داخل و خارج کند.

1. Character 2. International References Alphabet

۷-۲ مدول‌های I/O

وظایف مدول

مهم‌ترین وظایف یا نیازهای یک مدول I/O به صورت زیر دسته‌بندی می‌شود:

- کنترل و زمان‌بندی
- مخابره و تبادل با پردازنده
- مخابره یا تبادل با دستگاه
- بافرکردن داده
- تشخیص خطا

در هر پریودی از زمان پردازنده ممکن‌است با یک یا چند وسیله، تبادل اطلاعات پیش‌بینی نشده‌ای را برحسب نیاز I/O برنامه به عمل آورد. منابع اطلاعات داخلی مثل حافظه‌ی اصلی و گذرگاه سیستم، بین تعدادی از این‌گونه اعمال از جمله I/O داده‌ی مشترک می‌باشند. بنابراین وظیفه‌ی I/O شامل **کنترل و زمان‌بندی** تقاضاها است، تا جریان ترافیک بین منابع داخلی و وسایل بیرونی هماهنگ شوند. مثلاً کنترل انتقال داده از یک وسیله‌ی خارجی به پردازنده ممکن‌است مراحل متوالی زیر را شامل گردد:

۱. پردازنده از مدول I/O می‌خواهد تا وضعیت دستگاه متصل را چک کند.
۲. مدول I/O وضعیت دستگاه را باز می‌گرداند.
۳. اگر دستگاه در حال کار و آماده برای انتقال باشد، پردازنده تقاضای انتقال داده را خواهد کرد و این کار با یک فرمان به مدول I/O صورت می‌گیرد.
۴. مدول I/O یک واحد داده را (8 یا 16 بیت) از وسیله‌ی خارجی دریافت می‌نماید.
۵. داده از مدول I/O به پردازنده منتقل می‌گردد.

اگر سیستم از یک گذرگاه استفاده نماید، آن‌گاه هر واکنش بین پردازنده و مدول I/O از یک یا چند حکمیّت داده استفاده خواهد کرد.

سناریوی ساده‌شده‌ی قبلی همچنین بیان می‌دارد که مدول I/O باید با پردازنده و وسیله‌ی خارجی **تبادل اطلاعات** کند. تبادل داده‌ی پردازنده شامل موارد زیر است:

- **دیکد فرمان**[1]: مدول I/O فرامین را از پردازنده می‌پذیرد. معمولاً این فرامین به صورت سیگنال‌های کنترل روی گذرگاه کنترل ارسال می‌شود. مثلاً یک مدول I/O برای یک گرداننده‌ی دیسک فرامین زیر را می‌پذیرد: READ SECTOR، WRITE SECTOR ، SEEK (جستجو) شماره‌ی شیار و SCAN (پویش) ID رکورد. دو فرمان آخر دارای پارامتری هستند که به گذرگاه داده ارسال می‌گردد.

1. Command decoding

- **داده:** داده بین پردازنده و مدول I/O از طریق گذرگاه داده تبادل می‌شود.
- **گزارش وضعیت:** چون وسایل جانبی کُند هستند، لازم است وضعیت I/O را بدانیم، مثلاً اگر از یک مدول I/O تقاضای ارسال داده به پردازنده بشود (عمل خواندن)، ممکن‌است آماده‌ی انجام آن نباشد زیرا هنوز در حال کار بر روی فرمان قبلی است. این وضع می‌تواند با سیگنال وضعیتی گزارش شود. سیگنال‌های وضعیت رایج عبارتنداز BUSY و READY . همچنین سیگنال‌های دیگری برای گزارش خطاها وجود دارند.
- **شناسایی آدرس:** درست مثل حافظه که هر کلمه آدرس خاص خود را دارد، وسیله‌ی I/O نیز دارای آدرس است. بنابراین یک مدول I/O باید یک آدرس منحصربه‌فرد را برای هر وسیله‌ی جانبی تحت کنترل شناسایی کند.

از طرف دیگر، مدول I/O باید قادر به انجام **تبادل داده با دستگاه** باشد. این تبادل شامل فرامین، اطلاعات وضعیت و داده است (شکل ۲-۷).

سرعت بالا — تنظیم سرعت با فرد — RAM ⟵ I/O ⟵ وسیله‌ی جانبی (پریفرال)

یکی از وظایف اساسی مدول I/O ، **بافرکردن داده** است. نیاز به انجام این وظیفه با توجه به شکل ۱۱-۲ کاملاً آشکار است. درحالی‌که سرعت انتقال به داخل و خارج از حافظه یا پردازنده کاملاً بالا است، سرعت برای وسایل جانبی خیلی پایین‌تر بوده و محدوده‌ی گسترده‌ای را شامل می‌شود. داده‌ی واصله از حافظه‌ی اصلی با سرعت زیادی به مدول I/O ارسال می‌گردد. داده در مدول I/O بافر شده و سپس به وسیله‌ی جانبی بر اساس سرعت داده‌اش ارسال می‌گردد. در سمت دیگر، باز هم داده بافر می‌شود تا حافظه را در سرعت کند گرفتار ننماید. بنابراین مدول I/O باید بتواند با هر دو سرعت وسیله‌ی جانبی و حافظه کار کند. به طور مشابه، اگر وسیله‌ی جانبی با سرعت بالاتر از دستیابی به حافظه کار کند، مدول I/O عمل بافرکردن را انجام می‌دهد.

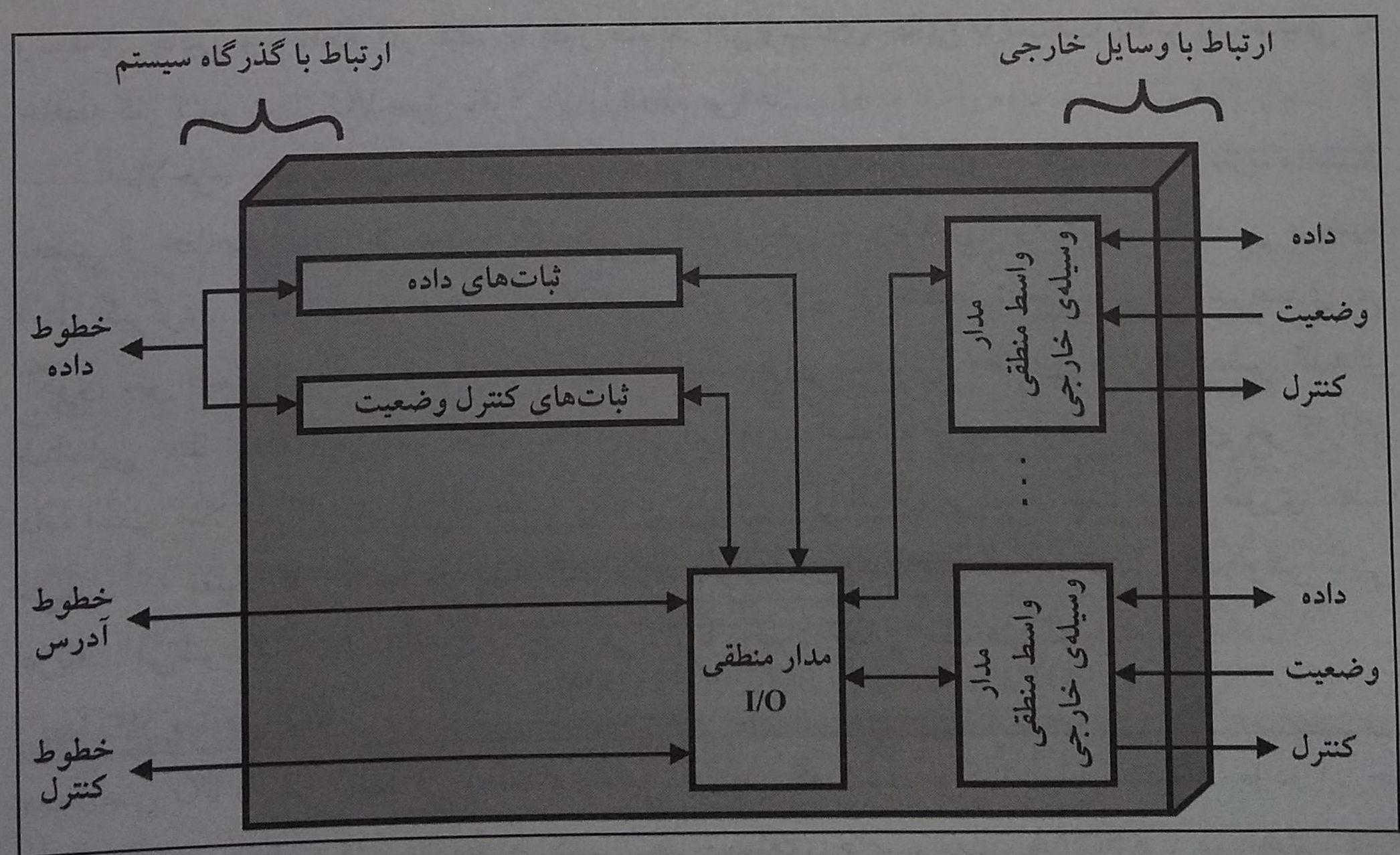
بالاخره، یک مدول I/O مسئول تشخیص خطا و متعاقباً گزارش خطاها به پردازنده است. بعضی از خطاهای ناشی از عیوب مکانیکی و الکترونیکی به وسیله‌ی خود دستگاه گزارش می‌شود (مثل گیرکردن کاغذ، شیار خراب دیسک). دسته‌ی دیگری از خطاها شامل تغییرات غیرعمدی در الگوی بیتی ضمن ارسال از وسیله به مدول I/O است. برای شناسایی خطاهای انتقال از بعضی کدهای شناسایی خطا استفاده می‌شود. مثال ساده‌ای در این مورد استفاده از بیت توازن در روی هر کاراکتر داده است. مثلاً کد کاراکتر IRA ، هفت بیت از یک بایت را اشغال می‌نماید. بیت هشتم طوری تنظیم می‌شود که تعداد کل یک‌ها در یک بایت زوج (توازن زوج) یا فرد (توازن فرد) باشد. وقتی بایتی دریافت می‌شود، مدول I/O توازن را چک می‌کند تا رخداد خطا را مشخص نماید.

ساختار مدول I/O

مدول‌های I/O از نظر پیچیدگی و تعداد وسایل خارجی که کنترل می‌کنند بسیار متنوعند. ما در این‌جا سعی می‌کنیم مطالب کلی را مطرح کنیم (یک وسیله‌ی خاص، یعنی 82C55A ، در بخش ۴-۷

توصیف شده است). شکل ۳-۷ نمودار بلوکی کلی یک مدول I/O را نشان می‌دهد. مدول از طریق یک مجموعه از خطوط سیگنال به بقیه‌ی کامپیوتر وصل است (مثلاً از طریق گذرگاه سیستم). داده‌های وارده یا خارج‌شده از مدول در یک یا چند ثبات داده بافر می‌شوند. همچنین ممکن‌است یک یا چند ثبات وضعیت هم وجود داشته باشند تا اطلاعات فعلی را فراهم نمایند. یک ثبات وضعیت ممکن‌است به عنوان ثبات کنترل هم کار کند و اطلاعات کنترل را از پردازنده بپذیرد. مدار منطقی موجود در مدول با پردازنده از طریق مجموعه‌ای از خطوط کنترل تعامل می‌کند. پردازنده از خطوط کنترل برای صدور فرامین به مدول I/O استفاده می‌کند (مثلاً سیگنال‌های حکمیّت و وضعیت). مدول باید بتواند آدرس وسیله‌ی تحت کنترل خود را تولید و یا شناسایی کند. هر مدول I/O دارای آدرس منحصربه‌فردی است، یا اگر بیش از یک وسیله‌ی خارجی را کنترل می‌کند باید یک مجموعه آدرس‌های منحصربه‌فردی را داشته باشد. بالاخره مدول I/O حاوی مدار خاصی برای اتصال به هر یک از وسایل تحت کنترل خود است.

کار یک مدول I/O طوری است که پردازنده گستره‌ی وسیعی از وسایل را به طریقی ساده نظاره کند. البته طیفی از توانمندی‌ها را ممکن است فراهم کند. مدول I/O ممکن است جزئیات زمان‌بندی، فرمت‌ها و الکترومکانیک یک وسیله‌ی خارجی را چنان از دید پردازنده مخفی نگه دارد تا پردازنده بتواند با فرامین ساده‌ی خواندن و نوشتن عمل کند و احتمالاً فرامین باز و بستن فایل‌ها را اجرا نماید. در ساده‌ترین فرم خود، مدول I/O ممکن‌است بسیاری از کارهای کنترل یک دستگاه (مثل برگرداندن یک نوار) را به پردازنده واگذار کند.



شکل ۳-۷ نمودار بلوکی یک مدول I/O.

یک مدول I/O که بیشتر مسئولیت بار پردازشی را بر عهده می‌گیرد و ارتباط سطح بالایی را با پردازنده دارد معمولاً *کانال I/O*[1] یا *پردازنده‌ی I/O*[2] نامیده می‌شود. مدول I/Oای که خیلی ساده باشد و کنترل زیادی درخواست کند، *کنترلگر I/O*[3] یا *کنترلگر وسیله* خوانده می‌شود. کنترلگرهای I/O معمولاً در میکروکامپیوترها مشاهده می‌شوند در حالی که کانال‌های I/O در کامپیوترهای بزرگ به کار می‌روند.

در ادامه، ما از کلمه‌ی *مدول I/O* استفاده می‌کنیم تا اشتباهی رخ ندهد و هر وقت هم لازم شد جملات خاص به کار خواهند رفت.

۷-۳ I/O برنامه‌نویسی شده

برای اعمال I/O سه تکنیک امکان‌پذیر است. با I/O برنامه‌نویسی‌شده، داده بین پردازنده و مدول I/O تبادل می‌شود. پردازنده برنامه‌ای را اجرا می‌نماید که به آن کنترل مستقیم عمل I/O را محول می‌کند، و از جمله آن‌ها حس وضعیت وسیله، ارسال یک فرمان خواندن و نوشتن و انتقال داده است. وقتی که پردازنده فرمانی به مدول I/O ارسال می‌کند، باید منتظر بماند تا عمل I/O پایان یابد. اگر پردازنده سریع‌تر از مدول I/O باشد، این فرمان تلف خواهد شد. در I/O کنترل‌شده با وقفه، پردازنده یک فرمان I/O صادر می‌کند و اجرای دیگر دستورات را ادامه می‌دهد، وقتی مدول I/O کار خود را تمام کند پردازنده به وسیله‌ی آن مدول متوقف می‌شود. به کمک I/O برنامه‌نویسی‌شده و I/O وقفه، پردازنده مسئول استخراج داده از حافظه‌ی اصلی برای خروجی و ذخیره‌ی داده در حافظه‌ی اصلی برای ورودی است. روش دیگر دستیابی مستقیم به حافظه[4] (DMA) است. در این حالت، مدول I/O و حافظه‌ی اصلی داده را مستقیماً مبادله می‌کنند و این کار بدون دخالت پردازنده صورت می‌گیرد.

جدول ۱-۷ رابطه‌ی بین این سه تکنیک را نشان می‌دهد. در این بخش ما به بررسی I/O برنامه‌نویسی‌شده می‌پردازیم، I/O وقفه و DMA به‌ترتیب در دو بخش بعد تشریح شده‌اند.

مروری بر I/O برنامه‌نویسی‌شده

وقتی‌که پردازنده حین اجرای برنامه با دستوری مربوط به I/O مواجه می‌شود، آن دستور را با صدور فرمانی به مدول I/O مربوطه، اجرا می‌کند. با I/O برنامه‌نویسی‌شده، مدول I/O عمل درخواستی را انجام داده و سپس بیت‌های مربوطه را در ثبات وضعیت I/O (شکل ۳-۷) تنظیم می‌کند. مدول I/O هیچ واکنش دیگری برای اطلاع پردازنده از خود نشان نمی‌دهد. به خصوص پردازنده را وقفه نمی‌دهد. بنابراین این به عهده‌ی پردازنده است که متناوباً وضعیت مدول I/O را چک کند تا تکمیل عمل را بیابد.

برای تشریح بیشتر تکنیک I/O برنامه‌نویسی‌شده ما ابتدا آن را از دید فرامین صادره‌ی I/O به وسیله‌ی پردازنده به مدول I/O می‌نگریم، و سپس از دید دستورات اجرا شده‌ی I/O به وسیله‌ی پردازنده به آن نگاه خواهیم کرد.

---

1. I/O Channel 2. I/O Processor 3. I/O Controller 4. Direct Memory Access

پردازش توسط ... درخواست کنترل زیاد I/O <=> Main Memory

جدول ۱-۷ تکنیک‌های I/O.

	بدون وقفه	با استفاده از وقفه
انتقال از I/O به حافظه از طریق پردازنده	I/O برنامه‌نویسی شده	I/O تحت وقفه
انتقال مستقیم I/O به حافظه		دستیابی مستقیم به حافظه (DMA)

I/O Command

فرامین I/O

برای اجرای یک دستور مرتبط با I/O ، پردازنده آدرسی برای مشخص‌کردن مدول I/O خاص و وسیله‌ی خارجی، و یک فرمان I/O ، صادر می‌کند. چهار نوع فرمان I/O وجود دارد که مدول I/O ، هنگام احضار به وسیله‌ی پردازنده، ممکن‌است دریافت کند:

- **کنترل:** این فرمان برای فعال‌کردن یک وسیله‌ی جانبی به کار می‌رود و به آن می‌گوید که چه کاری انجام دهد. مثلاً ممکن است به یک واحد نوار مغناطیسی فرمان برگرداندن نوار یا پیشروی به جلو به اندازه‌ی یک رکورد داده شود. این فرامین برای انواع خاصی از وسایل جانبی فراهم می‌شوند.
- **تست:** برای تست انواع شرایط وضعیت متعلق به یک مدول I/O و وسایل جانبی به کار می‌رود. پردازنده می‌خواهد بداند که آیا وسیله‌ی جانبی موردنظر روشن و آماده برای استفاده است. همچنین می‌خواهد بداند که آیا آخرین عمل I/O تکمیل شده است یا نه و این که خطایی رخ داده است یا خیر.
- **خواندن:** موجب می‌شود تا یک مدول I/O یک قلم داده را از وسیله‌ی جانبی دریافت کرده و آن را در یک بافر داخلی قرار دهد (در شکل ۳-۷ به عنوان ثبات داده معرفی شده است). آن‌گاه با تقاضای پردازنده، داده از مدول I/O روی گذرگاه داده قرار می‌گیرد.
- **نوشتن:** موجب می‌شود تا مدول I/O داده (بایت یا کلمه) را دریافت نموده و آن را به وسیله‌ی جانبی بفرستد.

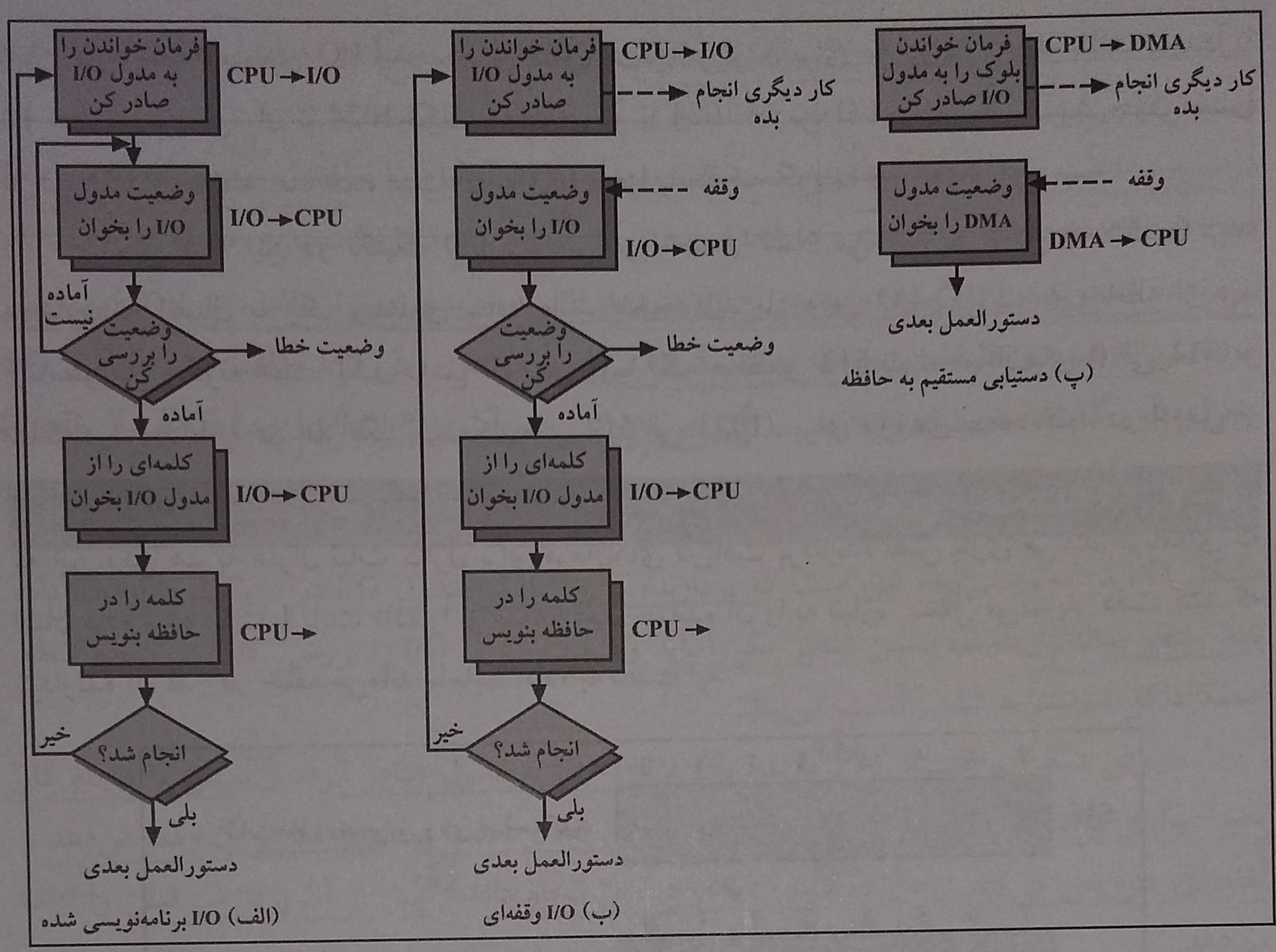
شکل ۴-۷(الف) مثالی از کاربرد I/O برنامه‌نویسی‌شده برای خواندن بلوکی از داده از وسیله‌ی جانبی به حافظه را نشان می‌دهد (مثلاً رکوردی از یک نوار). هر بار یک کلمه داده (مثلاً 16 بیت) خوانده می‌شود. برای هر کلمه‌ی خوانده شده، پردازنده باید در سیکل چک‌کردن وضعیت باقی بماند تا مشخص شود که کلمه در ثبات داده‌ی مدول I/O موجود است. این فلوچارت نقص این تکنیک را نشان می‌دهد: این روش بسیار فرآیند وقت تلف‌کن بوده و پردازنده را بدون دلیل مشغول نگه می‌دارد.

I/O Instruction

دستورات I/O

با I/O برنامه‌نویسی‌شده بین دستورات مربوط به I/O که به وسیله‌ی پردازنده از حافظه برداشته می‌شود و فرامین I/O صادره به وسیله‌ی پردازنده به مدول I/O برای اجرای دستورات، ارتباط نزدیکی وجود دارد. یعنی دستورات درواقع به فرامین I/O تبدیل می‌شوند و اغلب بین آن‌ها رابطه‌ی ساده‌ی یک‌به‌یک وجود دارد. فرم دستورات به روش آدرس‌دهی وسایل خارجی بستگی دارد.

* I/O برنامه نویسی شده دستورات مربوطه را به وسیله ی پردازنده از حافظه برمی دارد ⇐ دستورات به فرامین I/O تبدیل می شوند.



شکل ۴-۷ سه تکنیک برای ورود یک بلوک از داده.

معمولاً وسایل I/O متعددی از طریق مدول I/O به سیستم متصل می‌شوند. به هر وسیله یک شناسه‌ی منحصربه‌فرد یا آدرسی تعلق می‌گیرد. وقتی که پردازنده یک فرمان I/O صادر می‌کند، فرمان حاوی آدرس وسیله‌ی موردنظر است. بنابراین هر مدول باید خطوط آدرس را تفسیر کند تا مشخص شود که فرمان برای اوست.

وقتی که پردازنده، حافظه‌ی اصلی و I/O گذرگاه مشترکی را شریک‌اند، دو مد یا روش آدرس‌دهی وجود دارد: نگاشت به حافظه[1] و مجزا[2]. با I/O نگاشت به حافظه تنها یک آدرس برای مکان حافظه و وسایل I/O وجود دارد. پردازنده به ثبات‌های وضعیت و داده‌ی I/O به چشم مکان‌های حافظه می‌نگرد و دستورات ماشین یکسانی برای دستیابی به هر دو وسیله‌ی حافظه و I/O به کار می‌برد تا به حافظه و وسایل I/O دست یابد. بنابراین، به عنوان مثال، با 10 خط آدرس، مجموعاً $2^{10} = 1024$ مکان حافظه و آدرس I/O با هر ترکیبی قابل پشتیبانی است. ← نمی‌فهمم!!!

با I/O نگاشت به حافظه، تنها یک خط نوشتن و یک خط خواندن روی گذرگاه مورد نیاز است. به روشی دیگر، گذرگاه ممکن است با خواندن و نوشتن حافظه به علاوه خطوط فرمان ورودی و خروجی تجهیز گردد. در این صورت خط فرمان مشخص می‌کند که آدرس به مکانی از حافظه

خطوط فرمان ورودی و خروجی + خواندن و نوشتن حافظه

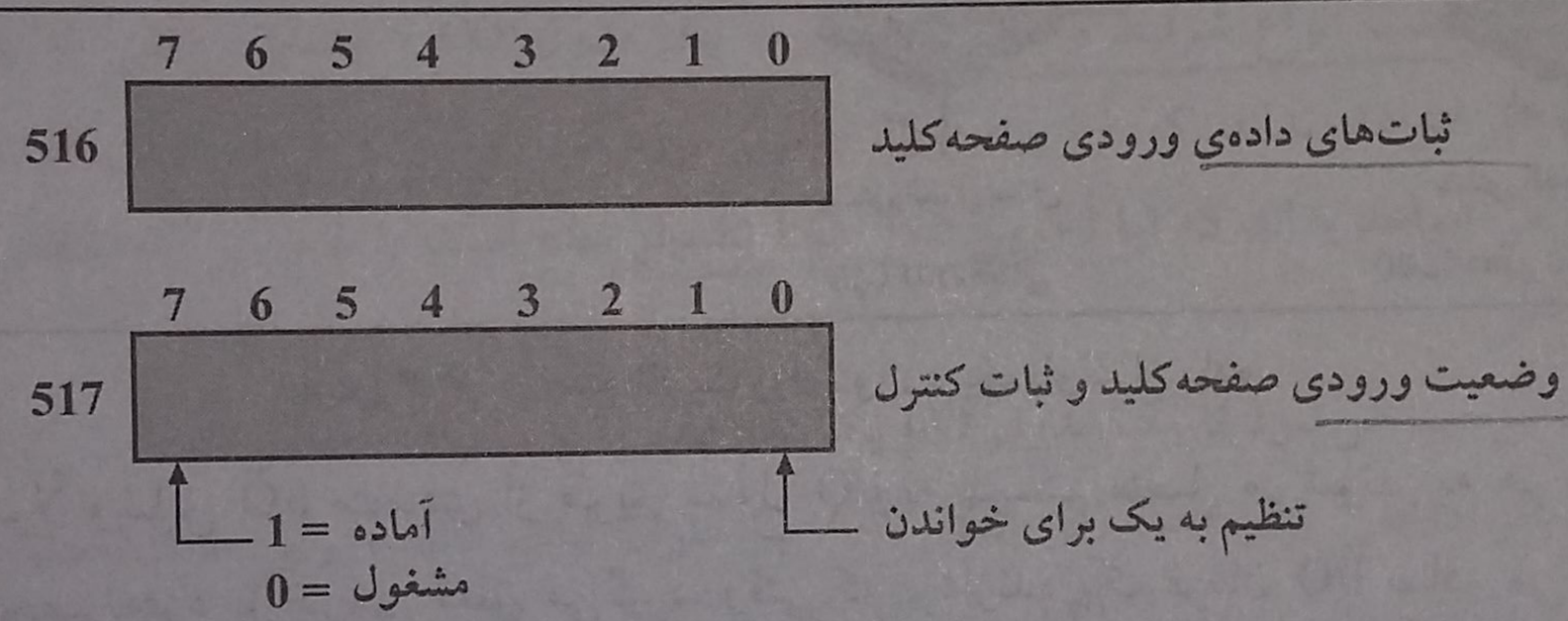
گذرگاه:

آدرس حافظه است یا I/O

1. Memory mapped 2. Isolated

اشاره دارد یا به وسیله‌ی I/O [محدوده‌ی کامل آدرس می‌تواند برای هر دو در اختیار باشد. مجدداً با 10 خط آدرس می‌توان تا 1024 مکان حافظه و نیز تا 1024 آدرس I/O را پشتیبانی نمود. چون فضای آدرس I/O از حافظه جدا شده است آن را **I/O مجزا** یا **منفک**[1] گویند.]

شکل ۵-۷ این دو تکنیک I/O برنامه‌نویسی شده را نشان می‌دهد. در شکل ۵-۷(الف) دیده می‌شود که اتصال به یک وسیله‌ی ساده مانند صفحه‌کلید با روش I/O نگاشت به حافظه از دید برنامه‌نویس چگونه است. یک آدرس 10 بیتی را با یک حافظه‌ی 512 بایت (مکان‌های 0 الی 511) و 512 آدرس I/O را در نظر بگیرید (مکان‌های 512 الی 1023). برای ورودی صفحه‌کلید دو آدرس از پایانه‌ی خاصی در نظر گرفته شده است. آدرس 516 به ثبات داده و 517 به ثبات وضعیت اشاره دارند، که این یکی هم به عنوان ثبات کنترل برای فرمان‌های دریافت پردازنده نقش بازی می‌کند. برنامه‌ای که نشان داده شده است 1 بایت داده را از صفحه‌کلید خوانده آن را به انباره[2] منتقل می‌نماید. دقت کنید که پردازنده آن قدر در حلقه می‌ماند تا بایت داده به دست آید.



آدرس	دستورالعمل	عملوند	توضیح
200	Load AC	"1"	انباره را باز کن
	Store AC	517	خواندن کلید را شروع کن
202	Load AC	571	بایت وضعیت را بردار
	Branch if Sign = 0	202	در حلقه بمان تا تکمیل
	Load AC	516	بایت داده را باز کن

(الف) I/O نگاشت به حافظه

آدرس	دستورالعمل	عملوند	توضیح
200	Load I/O	5	خواندن کلید را شروع کن
201	Test I/O	5	از نظر کامل بودن تست کن
	Branch Not Ready	201	در حلقه بمان تا تکمیل
	In	5	بایت داده را باز کن

(ب) I/O منفک

شکل ۵-۷ نگاشت به حافظه و I/O مجزا.

1. Isolated I/O 2. Accumulator

Memory Mapped
دستورات قابل استفاده و برنامه‌نویسی موثرتر: م
بخشی از حافظه فضای حافظه اشغال: ع

با I/O مجزا (شکل ۵-۷(ب))، پورت‌های I/O فقط به وسیله‌ی فرامین خاص I/O قابل دسترس‌اند و این فرامین خطوط فرمان I/O را روی گذرگاه فعال می‌کنند.

در اغلب پردازنده‌ها، مجموعه دستورات بسیاری برای ارجاع به حافظه وجود دارد. اگر از I/O مجزا استفاده شود، تنها چند دستور در اختیار است. بنابراین مزیت I/O نگاشت به حافظه این است که این تعداد دستورات قابل استفاده بوده و اجازه می‌دهد تا برنامه‌نویسی موثرتر باشد. عیب آن این است که بخش ارزشمندی از فضای حافظه به کار گرفته می‌شود. هر دو روش I/O نگاشت به حافظه و I/O مجزا به‌طور گسترده مورد استفاده‌اند.

مثلاً I/O برنامه‌ریزی شده ← ① انتظار برای دریافت یا ارسال داده ② بررسی مکرر وضعیت I/O

۷-۴ I/O تحت وقفه

مشکل I/O برنامه‌نویسی‌شده این است که پردازنده برای آمادگی مدول I/O در دریافت یا ارسال داده باید منتظر بماند. پردازنده ضمن انتظار مکرراً باید وضعیت مدول I/O را بررسی کند در نتیجه سطح عملکرد کل سیستم به شدت آسیب می‌بیند.

روش دیگر این است که پردازنده یک فرمان I/O به مدول صادر کرده و سپس به انجام کار مفید دیگری بپردازد. آن‌گاه مدول I/O وقتی که آماده‌ی تبادل داده با پردازنده شد به آن وقفه می‌دهد و تقاضای سرویس می‌کند. پردازنده سپس انتقال داده را انجام داده و پس از آن پردازش قبلی را ادامه می‌دهد.

ابتدا اجازه بدهید ببینیم که از دید مدول I/O این کار چگونه صورت می‌گیرد. برای ورودی، مدول I/O یک فرمان READ از پردازنده دریافت می‌کند. آن‌گاه I/O برای خواندن داده از یک وسیله‌ی جانبی پیش می‌رود. پس از استقرار داده در ثبات داده‌ی مدول، وقفه‌ای به وسیله‌ی مدول از طریق یک خط کنترل به پردازنده ارسال می‌گردد. سپس مدول منتظر می‌ماند تا داده از طرف پردازنده تقاضا گردد. وقتی که تقاضا اعلام شد، مدول داده‌اش را روی گذرگاه داده قرار می‌دهد و آن‌گاه منتظر I/O دیگری باقی می‌ماند.

خیلی جالبه

عمل ورودی از دید پردازنده به قرار زیر است. پردازنده یک فرمان READ صادر می‌کند، و سپس برای انجام کار دیگری مشغول می‌شود (مثلاً پردازنده ممکن‌است روی چند برنامه‌ی دیگر به طور همزمان عمل کند). در انتهای هر سیکل دستور، پردازنده رخداد وقفه را چک می‌کند (شکل ۳-۹). پس از وقوع وقفه از طرف مدول I/O ، پردازنده وضعیت برنامه‌ی فعلی (مانند شمارنده‌ی برنامه و ثبات‌های پردازنده) را ذخیره می‌کند و وقفه را پردازش می‌نماید. در این حال، پردازنده کلمه‌ی داده را از مدول I/O خوانده آن را در حافظه ذخیره می‌کند. سپس کمیت‌هایی را که قبلاً برنامه با آن کار می‌کرد از محل ذخیره شده بازگرداند و اجرای برنامه ادامه می‌یابد.

شکل ۴-۷(ب) استفاده از I/O تحت وقفه را برای خواندن بلوکی از داده نشان می‌دهد. این شکل را با شکل ۴-۷(الف) مقایسه کنید. I/O وقفه خیلی موثرتر از I/O برنامه‌نویسی شده است زیرا

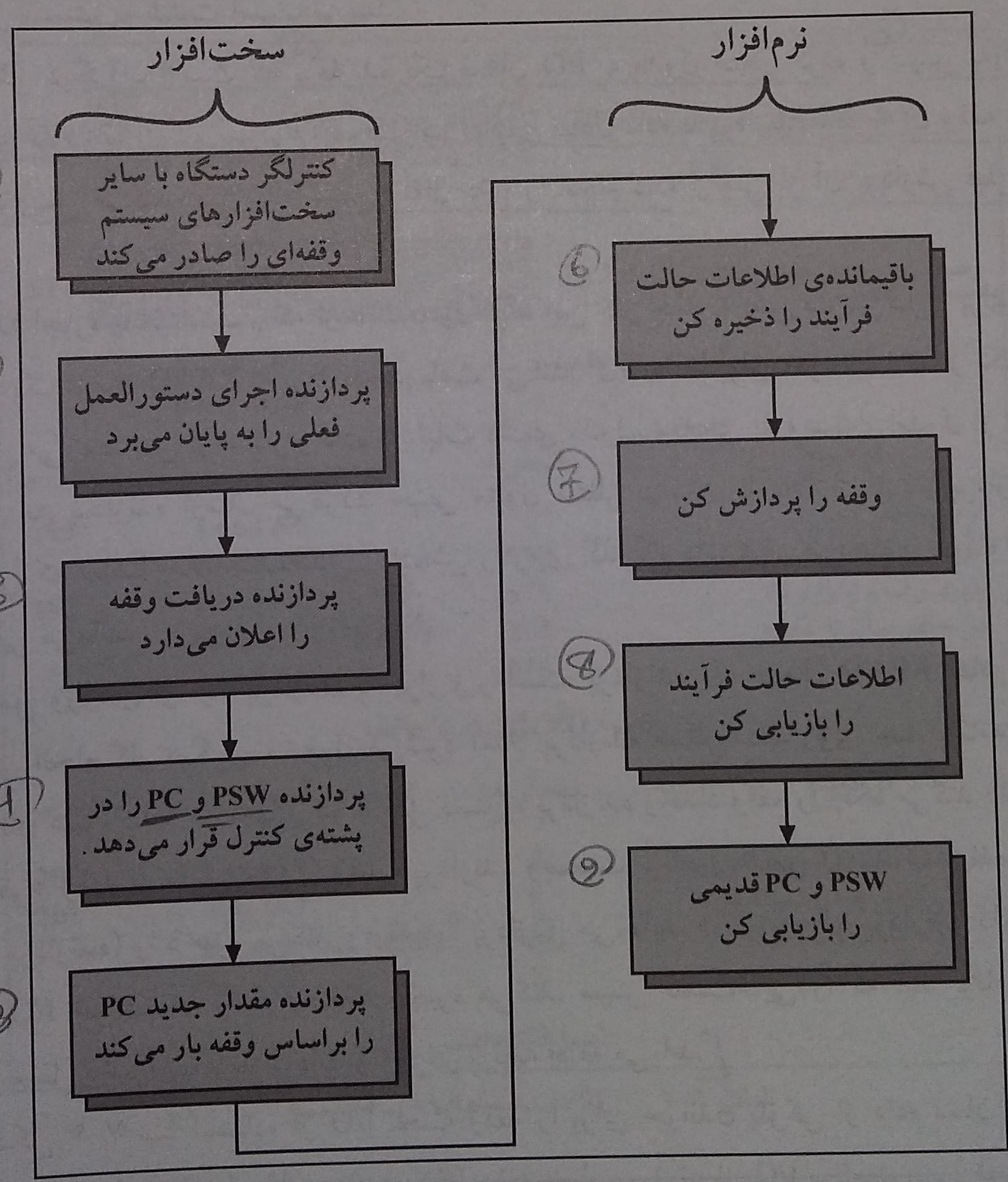
I/O وقفه موثرتر از I/O برنامه‌نویسی شده ← به دلیل حذف انتظار

انتظار را حذف کرده است. با این وجود I/O تحت وقفه باز هم زمان زیادی را از پردازنده می‌گیرد، زیرا هر کلمه‌ای از داده که از حافظه به مدول I/O و بالعکس می‌رود، باید از پردازنده عبور کند.

پردازش وقفه

بگذارید نقش پردازنده را در I/O تحت وقفه بیشتر بررسی کنیم. وقوع یک وقفه، پدیده‌های زیادی را هم در سخت‌افزار و هم در نرم‌افزار موجب می‌شود (شکل ۶-۷). وقتی‌که یک وسیله‌ی I/O یک عمل I/O را کامل می‌کند پدیده‌های سخت‌افزاری زیر به وقوع می‌پیوندد:

۱. وسیله یک سیگنال وقفه به پردازنده می‌فرستد.
۲. پردازنده اجرای دستور فعلی را قبل از واکنش به وقفه، پایان می‌دهد (شکل ۹-۳).
۳. پردازنده وقوع وقفه را تست می‌کند و در صورت وجود، سیگنال تصدیق را به دستگاه متقاضی وقفه ارسال می‌کند. تصدیق موجب می‌شود تا دستگاه، سیگنال وقفه را رها کند.



شکل ۶-۷ پردازش یک وقفه‌ی ساده.

۴. اکنون پردازنده آماده‌ی تحویل کنترل به روال وقفه است. برای شروع لازم است اطلاعات لازم برای ادامه‌ی برنامه‌ی فعلی از نقطه‌ی وقفه، ذخیره گردد. حداقل اطلاعات عبارتست از **(الف)** وضعیت پردازنده، که در ثبات کلمه‌ی وضعیت[1] (PSW) قرار دارد و **(ب)** مکان دستور بعدی مورد اجرا، که در شمارنده‌ی برنامه می‌باشد. این اطلاعات در پشته‌ی کنترل سیستم درج می‌گردد.

۵. اینک پردازنده شمارنده‌ی برنامه را با مکان ورودی برنامه‌ی اداره‌ی وقفه که به این وقفه سرویس می‌دهد، بار می‌کند. بسته به معماری کامپیوتر و طراحی سیستم‌عامل، ممکن‌است تنها یک برنامه برای هر نوع وقفه، یا یک برنامه برای هر دستگاه و هر نوع وقفه وجود داشته باشد. اگر بیش از یک برنامه اداره‌ی وقفه موجود باشد، پردازنده باید مشخص کند که کدام یک احضار گردد. این اطلاعات ممکن است در سیگنال وقفه‌ی اصلی وجود داشته باشد، و یا پردازنده ممکن است تقاضایی به وسیله ارسال کند تا پاسخی حاوی اطلاعات لازم را بفرستد.

پس از بارشدن شمارنده‌ی برنامه، پردازنده به سیکل دستور بعدی رفته و یک دستور برداشت می‌نماید. چون برداشت دستور به وسیله‌ی محتوای شمارنده‌ی برنامه معین می‌گردد، نتیجه این‌است‌که کنترل به برنامه‌ی مدیریت وقفه[2] محول می‌گردد. اجرای این برنامه اعمال زیر را موجب می‌شود:

۶. در این‌جا، شمارنده‌ی برنامه و PSW متعلق به برنامه‌ی وقفه دیده در پشته‌ی سیستم ذخیره می‌گردد. با این وجود، اطلاعات دیگری هم وجود دارد که به عنوان بخشی از "حالت" اجرای برنامه است. به خصوص، محتوای ثبات‌های پردازنده باید ذخیره شود، چون ممکن‌است به وسیله‌ی برنامه اداره وقفه استفاده شوند. به این ترتیب همه‌ی این مقادیر به علاوه اطلاعات حالت دیگر باید ذخیره شوند. معمولاً برنامه اداره وقفه با ذخیره کردن محتویات ثبات‌ها در پشته شروع می‌شود. شکل ۷-۷(الف) مثال ساده‌ای را نشان می‌دهد. در این مثال یک برنامه کاربر پس از اجرای دستور مکان N وقفه دیده است. محتوای همه‌ی ثبات‌ها به علاوه آدرس دستور بعدی (N+1) در داخل پشته درج شده است. اشاره‌گر پشته برای اشاره به بالای جدید پشته اصلاح می‌گردد و شمارنده‌ی برنامه اصلاح می‌شود تا به آغاز برنامه‌ی سرویس وقفه اشاره کند.

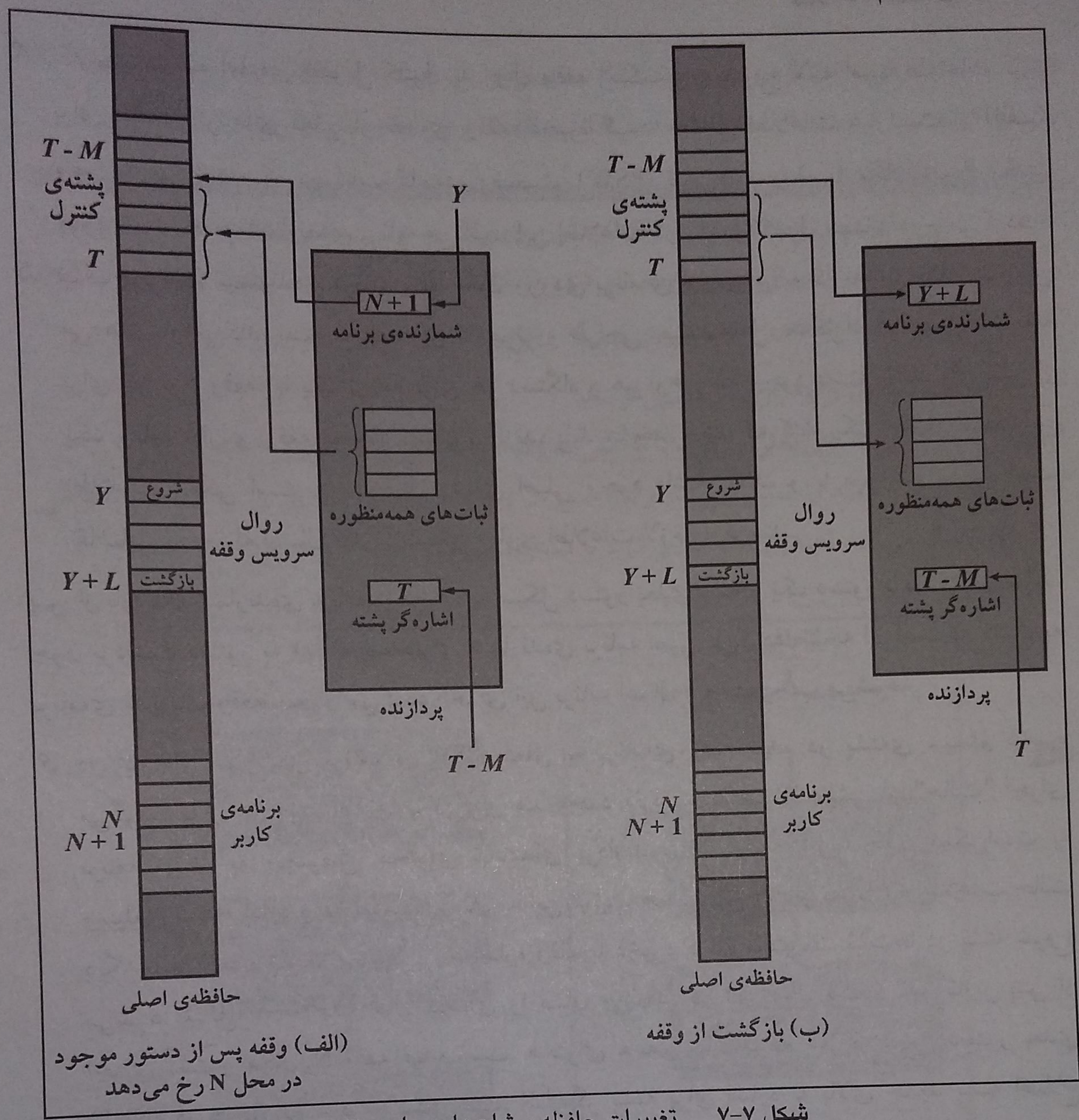
۷. اداره‌کننده‌ی وقفه سپس وقفه را پردازش می‌کند. این پردازش شامل بررسی اطلاعات وضعیت مرتبط با عمل I/O یا دیگر عوامل ایجاد وقفه می‌باشد. همچنین ارسال فرامین اضافی یا تصدیق‌ها را به وسیله‌ی I/O شامل است.

۸. پس از تکمیل پردازش وقفه، مقادیر ثبات‌های ذخیره شده از پشته بازیابی شده و در ثبات‌ها قرار داده می‌شوند (شکل ۷-۷(ب)).

۹. آخرین عمل، بازگرداندن PSW و مقادیر شمارنده‌ی برنامه از پشته می‌باشد. در نتیجه دستور مورد اجرای بعدی از برنامه وقفه یافته‌ی قبلی می‌باشد.

---

1. Program Status Word 2. Interrupt-handler program



شکل ۷-۷ تغییرات حافظه و ثبات‌ها به علت وقفه.

دقت کنید که ذخیره‌ی همه‌ی اطلاعات وضعیت در مورد برنامه‌ی وقفه دیده جهت ادامه‌ی کار اهمیت دارد. دلیل آن این‌است‌که وقفه یک روال فراخوانده‌شده از داخل برنامه نیست. بلکه، وقفه می‌تواند در هر زمان رخ دهد و لذا در هر نقطه‌ای از اجرای برنامه‌ی کاربر ممکن است به وقوع بپیوندد. به علاوه همان‌طور که در فصل بعد خواهیم دید، دو برنامه ممکن است بخش مشترکی نداشته باشند و یا متعلق به دو کاربر متفاوت باشند.

نکات طراحی

دو نکته در ارتباط با پیاده‌سازی I/O وقفه‌ای وجود دارد. اول، چون اغلب تعداد مدول‌های I/O چندگانه موجود است، پردازنده چگونه می‌فهمد که کدام وقفه صادر شده است؟ دوم، اگر چند وقفه همزمان رخ دهد پردازنده چگونه تصمیم می‌گیرد که کدام یک را پردازش کند؟

ابتدا به شناسایی دستگاه یا وسیله می‌پردازیم. چهار تکنیک رایج زیر به کار می‌رود:

- خطوط وقفه‌ی چندگانه[1] ← سرراست‌ترین (ولی ناممکن)
- سرکشی نرم‌افزاری[2] ← اتلاف کننده‌ترین (به لحاظ زمان)
- زنجیره‌ای[3] (سرکشی سخت‌افزاری، برداری‌شده) ← وقفه برداری (عبور از میان تک تک I/O ها)
- حکمیّت گذرگاه[4] (برداری‌شده) ← یک ماژول ابتدا کنترل گذرگاه را به دست می‌گیرد

سرراست‌ترین روش در رابطه با موضوع، **خطوط وقفه‌ی چندگانه** بین پردازنده و مدول‌های I/O است. با این وجود تخصیص بیش از چند خط گذرگاه یا پایه‌ی پردازنده به خطوط وقفه غیرعملی است. در نتیجه حتی اگر خطوط چندتایی هم به کار رود، به هر خط چند مدول I/O متصل می‌گردد. بنابراین یکی از سه تکنیک دیگر باید روی هر خط به کار گرفته شود. در نتیجه حتی اگر خطوط چندتایی هم به کار رود، به هر خط چند مدول I/O متصل می‌گردد.

روش دوم، **سرکشی نرم‌افزاری** است. وقتی‌که پردازنده وقفه‌ای را تشخیص دهد، به یک روال سرویس وقفه انشعاب می‌کند که کار آن سرکشی به هر مدول I/O برای تعیین مدول وقفه‌دهنده است. سرکشی می‌تواند به فرم یک خط فرمان جداگانه (مثل TEST I/O) باشد. در این حالت، پردازنده TEST I/O را می‌افرازد و آدرس یک مدول I/O خاص را روی خطوط آدرس قرار می‌دهد. اگر واحد I/O مذکور وقفه را فعال کرده باشد، پاسخ مثبت است. به طریقی دیگر، هر مدول I/O می‌تواند دارای ثبات وضعیت آدرس‌پذیر باشد. آن‌گاه پردازنده ثبات وضعیت هر مدول I/O را برای شناسایی مدول وقفه‌دهنده می‌خواند. به محض شناسایی مدول مزبور، پردازنده به روال سرویس دستگاه خاص آن انشعاب می‌نماید.

عیب سرکشی نرم‌افزاری این‌است‌که اتلاف‌کننده‌ی زمان است. تکنیک کارا استفاده از **روش زنجیره‌ای** است، که در واقع نوعی سرکشی سخت‌افزاری است. مثالی از آرایش زنجیره‌ای در شکل ۳۰-۳ ملاحظه شد. در وقفه‌ها، همه‌ی مدول‌ها از یک خط وقفه‌ی مشترک استفاده می‌کنند. خط تصدیق وقفه زنجیروار از میان مدول‌ها می‌گذرد. وقتی‌که پردازنده وقفه‌ای را حس می‌کند، یک سیگنال تصدیق وقفه را ارسال می‌کند. این سیگنال از میان یک سری مدول‌های I/O منتشر می‌شود تا به مدول متقاضی برسد. مدول متقاضی با قراردادن یک کلمه روی خطوط داده، واکنش لازم را نشان می‌دهد. این کلمه داده را *بردار* می‌گویند که ممکن‌است آدرس مدول I/O یا شناسه‌های دیگری از آن باشد. در هر حال، پردازنده از بردار به عنوان اشاره‌گر به روال سرویس دستگاه مذکور استفاده می‌نماید. این کار مانع از این می‌شود که ابتدا روال عمومی سرویس وقفه اجرا گردد. این تکنیک را *وقفه‌ی برداری شده*[5] می‌نامند.

1. Multiple interrupt lines 2. Software polling 3. Display chain 4. Bus arbitration
5. Vector interrupted

روشی دیگر در استفاده از وقفه‌های برداری حکمیّت گذرگاه است. در **حکمیّت گذرگاه**، یک مدول I/O ابتدا کنترل گذرگاه را قبل از افراشتن خط تقاضای وقفه به دست می‌گیرد. بنابراین در هر لحظه فقط یک مدول می‌تواند خط را بیافرازد. وقتی‌که پردازنده وقفه را شناسایی کند، با خط تصدیق وقفه پاسخ می‌دهد. سپس مدول متقاضی بردارش را روی خطوط داده می‌گذارد.

تکنیک‌های فوق‌الذکر برای شناسایی مدول I/O متقاضی به کار می‌رود. آن‌ها می‌توانند وقتی بیش از یک دستگاه متقاضی سرویس وقفه است، اولویت‌هایی را معین کنند. در حالت چند خطی، پردازنده فقط خطی را برمی‌گزیند که دارای بالاترین اولویت باشد. با سرکشی نرم‌افزاری، ترتیب سرکشی اولویت آن‌ها را مشخص می‌کند. به‌طور مشابه، مرتبه‌ی مدول در یک روش زنجیره‌ای اولویت را مشخص می‌نماید. بالاخره حکمیّت گذرگاه هم می‌تواند طرح اولویت را به کار گیرد که در بخش ۴-۳ ملاحظه شد.

اکنون به دو مثال از ساختارهای وقفه می‌پردازیم.

کنترلگر وقفه‌ی Intel 82C59A

پردازنده‌ی 80386 دارای یک خط تقاضای وقفه[1] (INTR) و یک خط تصدیق وقفه[2] (INTA) است. برای این‌که 80386 بتواند چند دستگاه را با ساختار اولویت سرویس دهد، معمولاً با یک حَکَم یا داور خارجی یعنی 82C59A همراه است. وسایل خارجی به 82C59A وصل می‌شوند، که این یکی به نوبه‌ی خود به 80386 متصل می‌گردد.

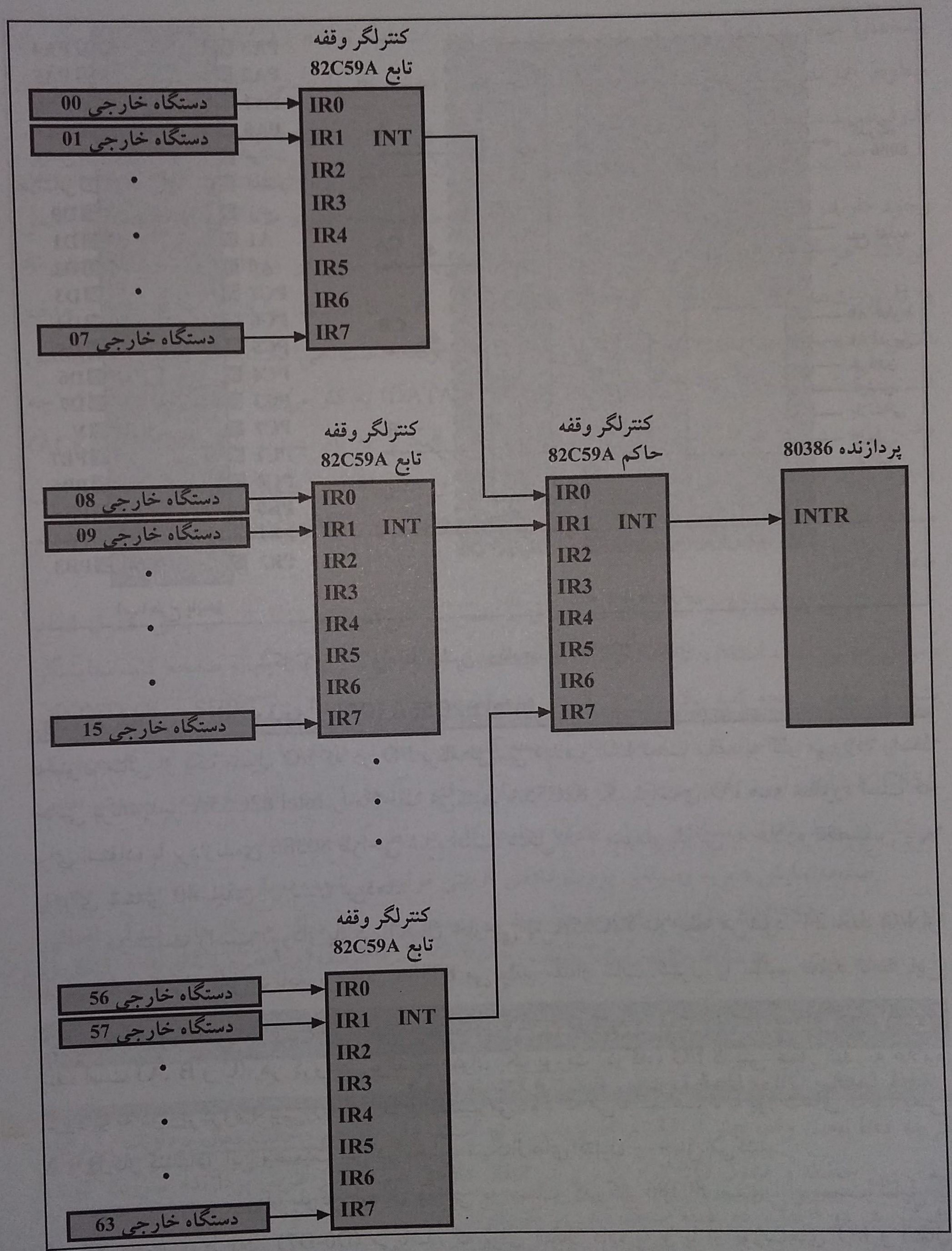
شکل ۸-۷ اتصال 82C59A را به مدول‌های I/O چندگانه برای 80386 نشان می‌دهد. یک 82C59A می‌تواند تا به 8 مدول وصل شود. اگر بیش از 8 مدول لازم باشد، آرایش متوالی را می‌توان به کار برد تا با 64 مدول ارتباط برقرار شود.

تنها مسئولیت 82C59A مدیریت وقفه است. این وسیله تقاضاهای وقفه را از مدول‌های متصل دریافت می‌کند و اولویت آن‌ها را مشخص کرده و سپس پردازنده را با افراشتن خط INTR مطلع می‌سازد. پردازنده تقاضا را از طریق خط INTA تصدیق می‌کند. این موجب می‌شود تا 82C59A اطلاعات برداری مربوطه را روی گذرگاه داده قرار دهد. آن‌گاه پردازنده می‌تواند برای پردازش وقفه پیش رفته و مستقیماً با مدول I/O برای خواندن یا نوشتن داده تعامل نماید.

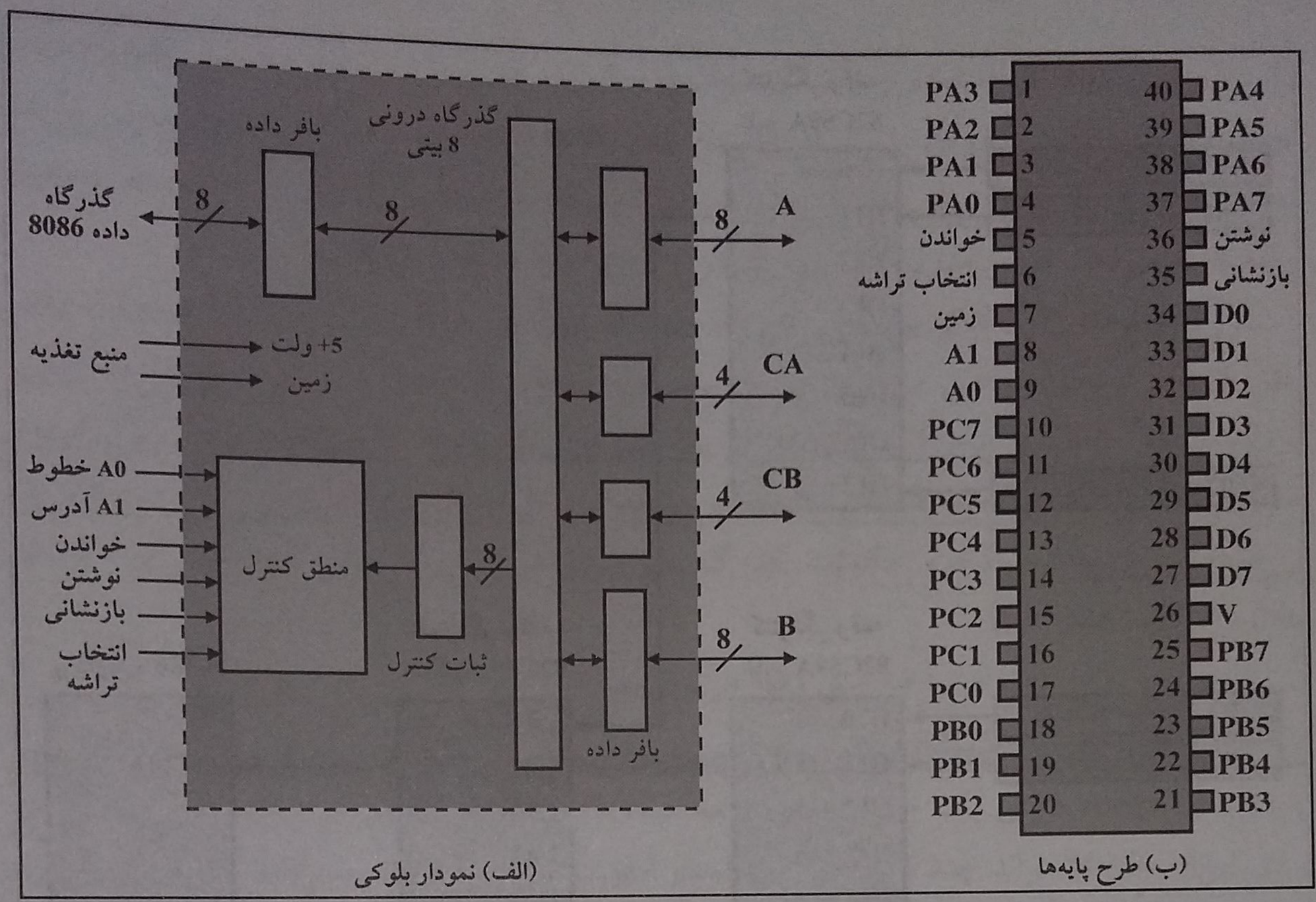
82C59A قابل برنامه‌ریزی است. 80386 با تنظیم کلمه‌ی کنترل در 82C59A طرح اولویت را مشخص می‌کند. مدل‌های وقفه‌ی زیر امکان‌پذیرند:

- **تودرتوی کامل**[3]: تقاضاهای وقفه به‌ترتیب اولویت از 0(IR0) تا 7(IR7) مرتب می‌شوند.
- **چرخشی**[4]: در بعضی از کاربردها تعدادی از وسایل وقفه دارای اولویت یکسانی هستند. در این مُد یک وسیله پس از سرویس دارای پایین‌ترین اولویت در گروه می‌شود.
- **پوشش خاص**[5]: این عمل موجب می‌شود تا از وقفه‌ی بعضی از وسایل جلوگیری گردد.

1. Interrupt request 2. Interrupt acknow 3. Fully nested 4. Rotating 5. Special mask



شکل ۸-۷ استفاده از کنترلگر 82C59A.



شکل ۹-۷ واسط جانبی برنامه‌پذیر Intel 82C55A .

واسط جانبی برنامه‌پذیر[1] Intel 82C55A (PPI)

به‌عنوان مثالی از یک مدول I/O که در I/O برنامه‌نویسی شده و I/O تحت وقفه به کار می‌رود، واسط جانبی برنامه‌پذیر Intel 82C55A را ملاحظه می‌کنیم. 82C55A یک تراشه‌ی I/O همه منظوره است که برای استفاده با پردازنده‌ی 80386 طراحی شده است. شکل ۹-۷ نمودار بلوکی به علاوه تخصیص پایه را برای بسته‌ی 40 پایه‌ی آن نشان می‌دهد.

در سمت راست نمودار بلوکی اتصال خارجی به 82C55A ملاحظه می‌شود. 24 خط I/O از طریق ثبات کنترل قابل برنامه‌ریزی‌اند. 80386 می‌تواند مقدار ثبات کنترل را تنظیم نماید تا به این ترتیب مدهای کاری متفاوت و آرایش‌های مختلف تعریف شود. 24 خط به سه گروه 8 بیتی تقسیم شده است (A، B و C). هر گروه می‌تواند به عنوان یک پورت (درگاه) I/O 8 بیتی عمل کند. به علاوه گروه C به دو زیرگروه 4 بیتی (CA و CB) تقسیم می‌شود که می‌توانند همراه با پورت‌های I/O ، یعنی A و B کار کنند. در این وضعیت این دو بخش، سیگنال‌های کنترل را حمل می‌کنند.

سمت چپ نمودار بلوکی اتصال داخلی به سمت گذرگاه 80386 است. این سمت شامل 8 گذرگاه داده‌ی دوطرفه‌ی (D0-D7) می‌باشد، که برای انتقال داده به و یا از پورت‌های I/O و انتقال اطلاعات کنترلی به ثبات کنترل به کار می‌رود. دو خط آدرس یکی از سه پورت I/O یا ثبات کنترل را

1. Programmable Peripheral Interface

مشخص می‌کنند. انتقال هنگامی رخ می‌دهد که خط انتخاب تراشه (CHIP SELECT) با یکی از خطوط خواندن (READ) یا نوشتن (WRITE) فعال شود. خط بازنشانی (RESET) برای بازآغازی مدول مورد استفاده است.

ثبات کنترل به وسیله‌ی پردازنده بار می‌شود تا مد عمل را کنترل نماید و اگر سیگنال‌هایی وجود دارند، آن‌ها را تعریف کند. در مد صفر، سه گروه 8 بیتی به عنوان سه گذرگاه I/O 8 بیتی عمل می‌کنند. هر گروه می‌تواند به عنوان ورودی یا خروجی تعریف شود. در غیر این صورت گروه‌های A و B پورت‌های I/O و خطوط گروه C خطوط کنترل A و B هستند. سیگنال‌های کنترل دو هدف اصلی را سرویس می‌دهند: یعنی دست‌دهی[1] و تقاضای وقفه. دست‌دهی یک راهکار ساده‌ی زمان‌بندی است. یک خط کنترل به نام آمادگی داده (DATA READY) به کار می‌رود تا مشخص شود که چه وقت داده روی خطوط داده‌ی I/O موجود است. خط دیگری به نام تصدیق (ACKNOWLEDGE) وجود دارد و مشخص می‌کند که داده‌ی خوانده شده و خطوط داده احتمالاً آزاد است. خط دیگری ممکن است با تقاضای وقفه‌ی (INTERRUPT REQUEST) نامگذاری شده و به گذرگاه سیستم متصل گردد.

چون 82C55A از طریق ثبات کنترل برنامه‌ریزی می‌شود، می‌توان از آن برای کنترل انواع وسایل جانبی ساده استفاده کرد. شکل ۱۰-۷ استفاده از آن را در کنترل یک پایانه صفحه کلید/نمایشگر نشان می‌دهد. صفحه کلید یک ورودی 8 بیتی را فراهم می‌کند. دو کلید یعنی SHIFT و CONTROL، مفهوم خاصی در اجرای برنامه‌ی سرویس صفحه کلید در پردازنده دارند. با این وجود این تفسیر برای 82C55A شفاف است که به راحتی 8 بیت داده را پذیرفته و آن‌ها را روی گذرگاه داده‌ی سیستم ظاهر می‌کند. دو خط کنترل دست‌دهی برای کار با صفحه کلید در نظر گرفته شده‌اند.

صفحه‌نمایش هم به وسیله‌ی پورت داده‌ی 8 بیتی مرتبط می‌شود. دوباره دو بیت مفهوم خاصی دارند و برای 82C55A شفافند. علاوه بر دو خط دست‌دهی، دو خط دیگر هم کارهای کنترلی اضافی را بر عهده دارند.

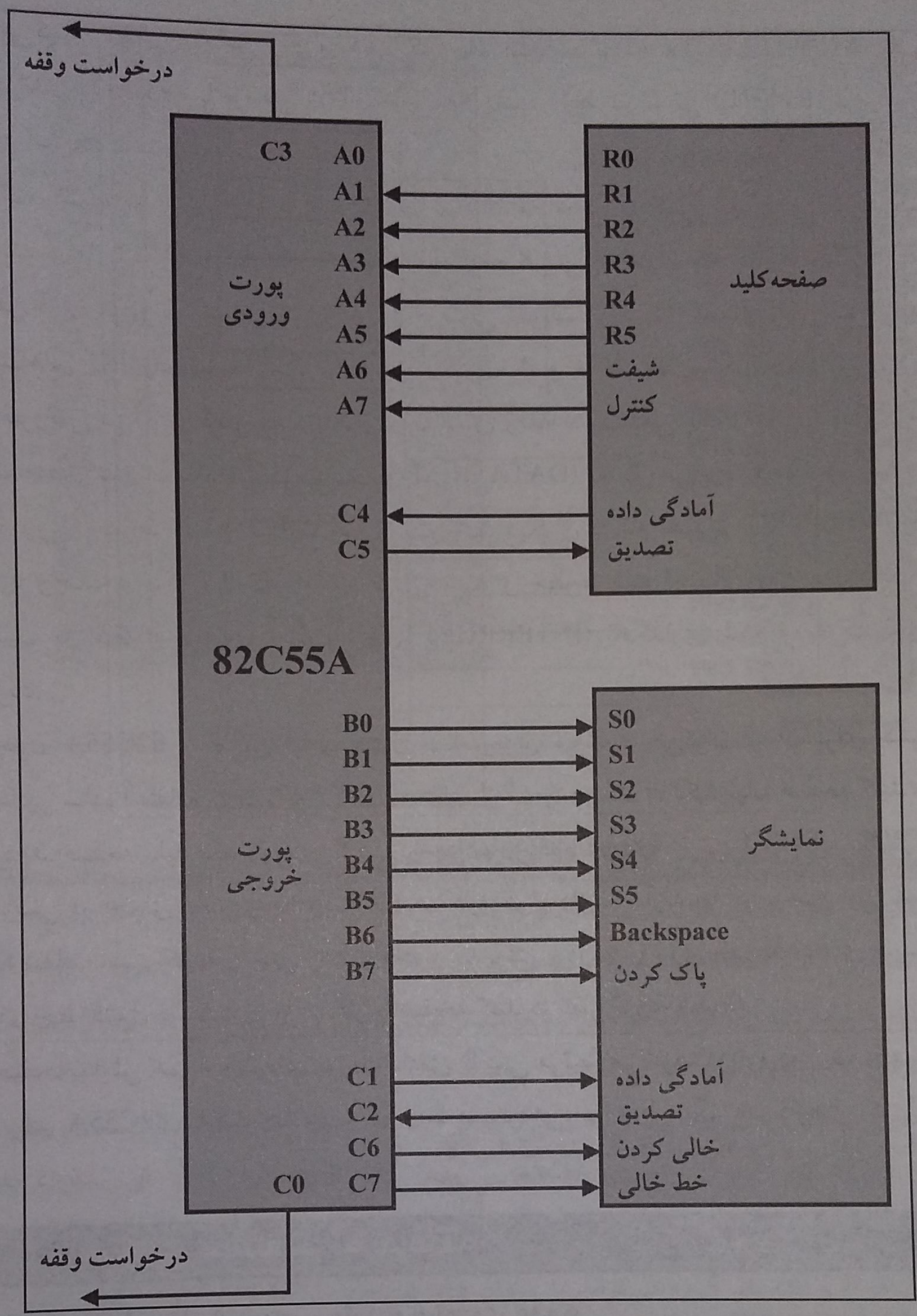
۷-۵ دستیابی مستقیم به حافظه

نقاط ضعف I/O برنامه نویسی شده و تحت وقفه

گرچه I/O تحت وقفه بهتر از I/O برنامه‌نویسی شده است، ولی باز هم دخالت فعال پردازنده را برای انتقال بین حافظه و مدول I/O لازم دارد و هر انتقال داده‌ای هم باید از طریق پردازنده صورت گیرد. بنابراین هر دو آن‌ها از دو نقطه ضعف ذاتی برخوردارند.

۱. سرعت انتقال I/O با سرعت تست و سرویس پردازنده محدود است.

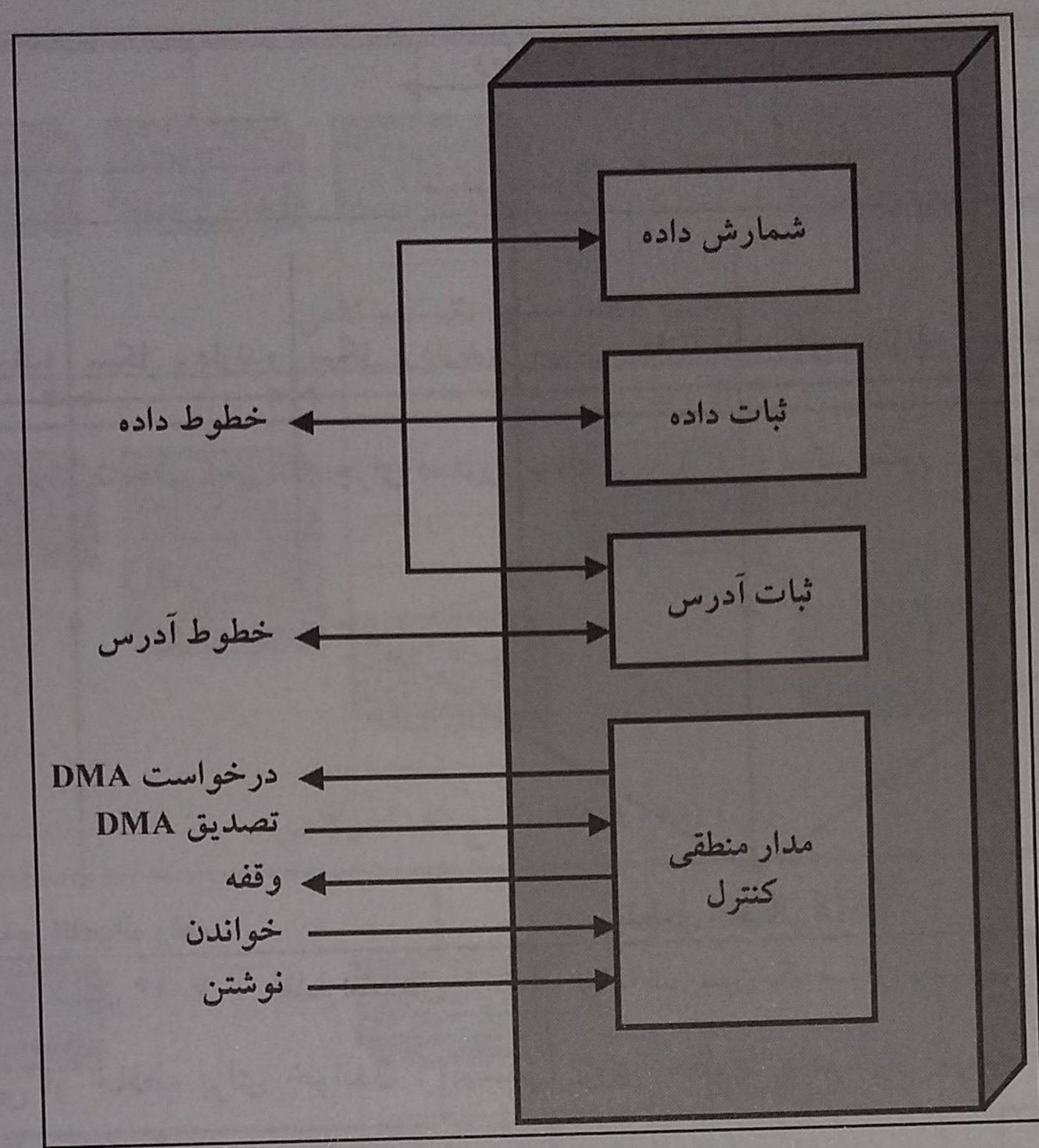
1. Handshaking



شکل ۱۰-۷ واسط صفحه‌کلید/نمایشگر به 82C55A.

۲. پردازنده به مدیریت انتقال I/O گره می‌خورد و تعدادی از دستورات برای هر انتقال I/O باید اجرا شود (مثل شکل ۵-۷).

نوعی مصالحه بین این دو ضعف وجود دارد. انتقال بلوکی از داده را در نظر بگیرید. با به کارگیری یک I/O برنامه‌ریزی شده‌ی ساده، اگر پردازنده کار دیگری انجام ندهد، با انجام I/O قادر خواهد بود تا داده را با سرعت زیادی جابه‌جا کند. I/O وقفه‌پذیر در ازای کمی هزینه از سرعت انتقال I/O، پردازنده را رها می‌کند. با این وجود هر دو روش تأثیر معکوسی بر روی فعالیت پردازنده و سرعت انتقال I/O دارند.



شکل ۱۱-۷ نمودار بلوکی DMA نمونه.

وقتی که حجم زیادی از داده انتقال می‌یابد، تکنیک کاراتری لازم است؛ یعنی دستیابی مستقیم به حافظه (DMA).

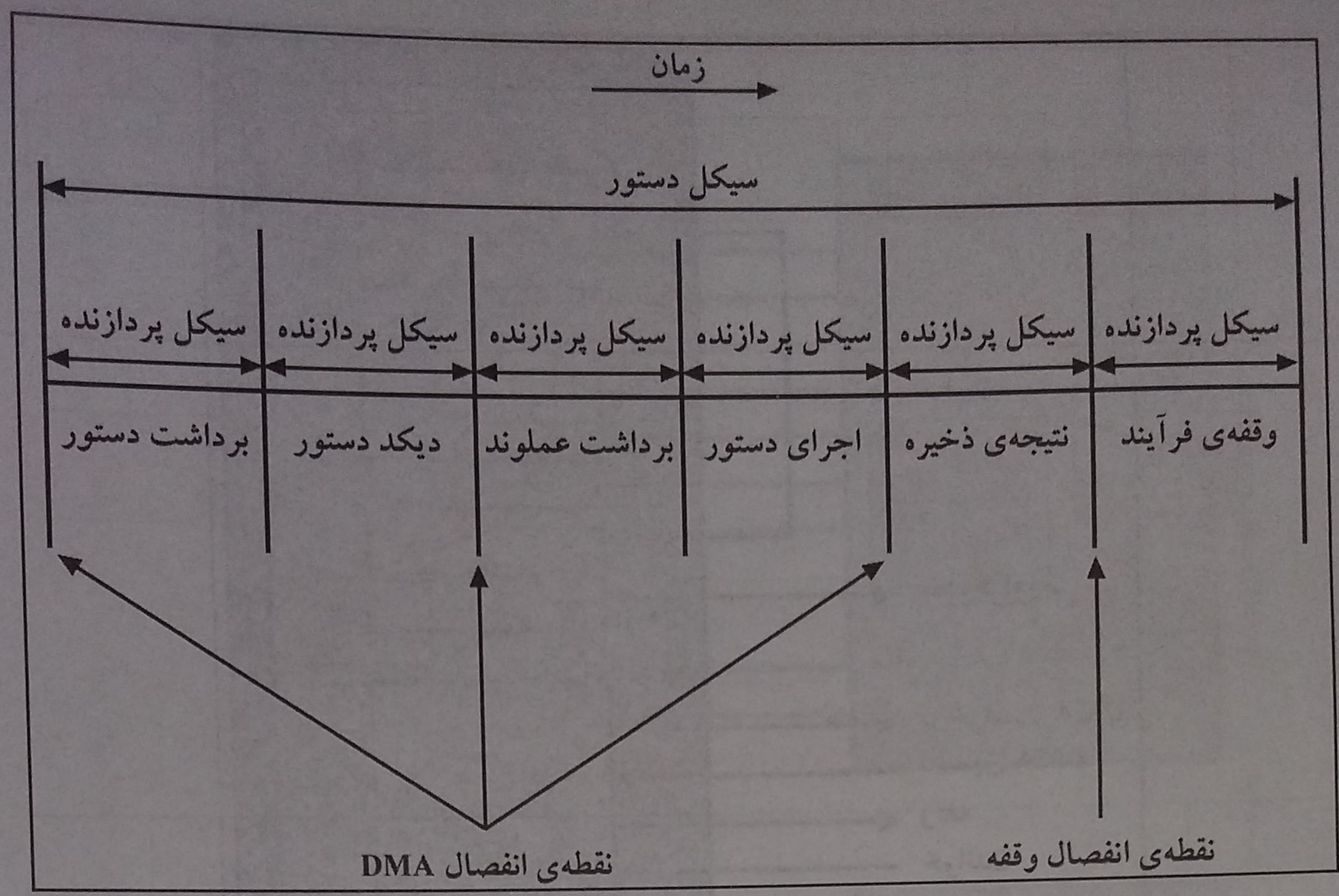
طرز کار DMA

DMA یک مدول اضافی دیگر بر روی گذرگاه سیستم است. مدول DMA (شکل ۱۱-۷) قادر است کار پردازنده را تقلید کند، به علاوه کنترل سیستم را از پردازنده بگیرد. مدول این کار را به خاطر انتقال داده به و از حافظه از طریق گذرگاه سیستم انجام می‌دهد. به این خاطر مدول DMA باید گذرگاه را فقط وقتی پردازنده آن را لازم ندارد، به کار بگیرد. تکنیک اخیر رایج‌تر بوده و به آن سرقت سیکل[1] می‌گویند، زیرا مدول DMA در واقع سیکل را می‌دزدد.

وقتی‌که پردازنده می‌خواهد بلوکی از داده را بخواند یا بنویسد، یک فرمان به مدول DMA صادر می‌کند، و این کار را با ارسال اطلاعات زیر به مدول DMA انجام می‌دهد:

- خواه خواندن یا نوشتن تقاضا شود، از خط کنترل خواندن یا نوشتن بین پردازنده و مدول DMA استفاده می‌گردد.
- آدرس وسیله‌ی I/O درگیر روی گذرگاه داده قرار داده می‌شود.

1. Cycle stealing



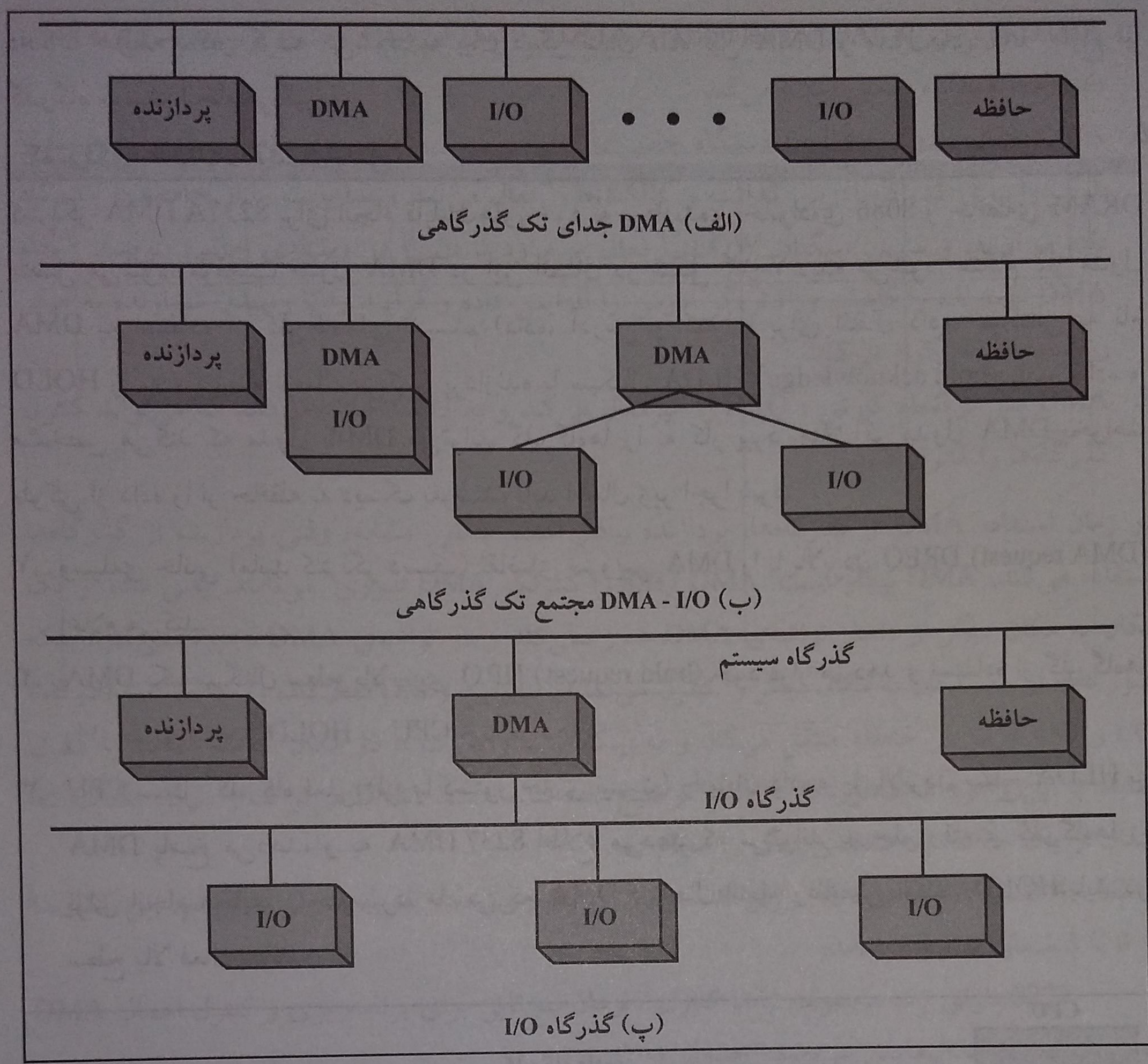
شکل ۱۲-۷ نقاط انفصال DMA و وقفه در حین یک سیکل دستور.

- آدرس مکانی از حافظه برای خواندن و نوشتن، روی خطوط داده قرار گرفته و در ثبات آدرس مدول DMA قرار می‌گیرد.
- تعدادی از کلمات خواندنی یا نوشتنی، مجدداً از طریق خطوط داده‌ی مبادله شده و در ثبات شمارش داده ذخیره می‌گردند.

آن‌گاه پردازنده به انجام کارهای دیگر مشغول می‌گردد. او این عمل I/O را به مدول DMA واگذار کرده است. مدول DMA هر بار کل بلوک داده را منتقل کرده و بدون عبور از پردازنده آن را مستقیماً به و یا از حافظه منتقل می‌سازد. پس از تکمیل انتقال، مدول DMA سیگنال وقفه‌ای به پردازنده می‌فرستد بنابراین پردازنده فقط در ابتدا و انتهای انتقال درگیر می‌شود (شکل ۴-۷(پ)).

شکل ۱۲-۷ نشان می‌دهد که در سیکل دستورالعمل چه موقع پردازنده به حالت تعلیق در می‌آید. در هر حال پردازنده درست قبل از لزوم استفاده از گذرگاه به تعلیق در می‌آید. آن‌گاه مدول DMA یک کلمه‌ی انتقال داده و کنترل را به پردازنده باز می‌گرداند. توجه کنید این عمل، یک وقفه نیست، پردازنده هیچ متنی را ذخیره نمی‌کند و یا کار دیگری را انجام نمی‌دهد. بلکه برای یک سیکل گذرگاه درنگ می‌کند. در مجموع این کار موجب کُندشدن کار پردازنده می‌شود. در هر صورت، برای انتقال I/O چند کلمه‌ای DMA بسیار کاراتر از I/O تحت وقفه یا I/O برنامه‌نویسی شده است.

راهکار DMA به طرق مختلف قابل پیاده‌سازی است. بعضی از این روش‌ها در شکل ۱۳-۷ مشاهده می‌شود. در مثال اول، همه‌ی مدول‌ها از یک گذرگاه سیستم مشترک استفاده می‌کنند. مدول DMA که در نقش یک پردازنده‌ی جانشین عمل کرده، از I/O برنامه‌نویسی شده برای تبادل داده بین



شکل ۱۳-۷ آرایش دیگری از DMA.

حافظه و مدول I/O از طریق مدول DMA استفاده می‌کند. این آرایش، ضمن این‌که ممکن‌است ارزان باشد، آشکارا کارآمد نیست. همچنین I/O برنامه‌نویسی شده کنترل شده با پردازنده، در هر انتقال کلمه، دو سیکل گذرگاه را مصرف می‌کند.

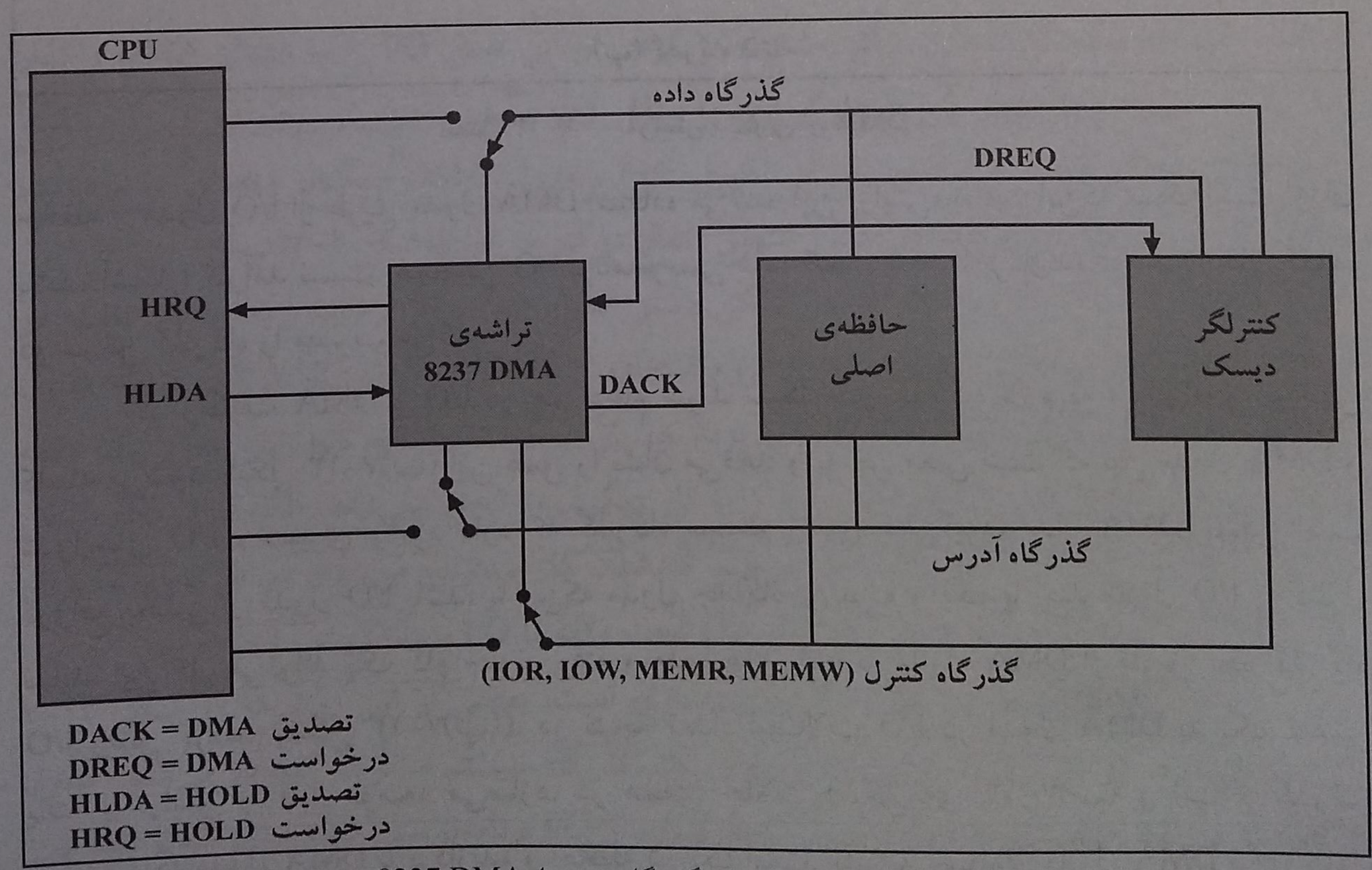
اگر وظایف DMA و I/O در هم ادغام شوند سیکل‌های گذرگاه لازم به میزان قابل توجهی کاسته می‌شود. شکل ۱۳-۷(ب) این عمل را نشان می‌دهد و به این معنی است که بین مدول DMA و مدول‌های I/O ، مسیری وجود دارد که گذرگاه سیستم را دربرنمی‌گیرد، مدار DMA ممکن است درواقع بخشی از مدول I/O باشد، یا این‌که مدول جداگانه‌ای بوده و یک یا چند مدول I/O را کنترل نماید. این کار می‌تواند یک گام جلوتر رفته و مدول‌های I/O به مدول DMA از طریق یک گذرگاه I/O وصل گردند (شکل ۱۳-۷(پ)). در نتیجه تعداد اتصالات I/O در مدول DMA به یک کاهش یافته و آرایش را قابل توسعه می‌سازد. در همه‌ی حالات (شکل‌های ۱۳-۷(ب) و (پ))، گذرگاه سیستمی که مدول DMA با پردازنده و حافظه شریک است، تنها به وسیله‌ی مدول DMA برای تبادل

داده با حافظه به‌کار گرفته می‌شود. به بیان دیگر تبادل داده بین DMA و مدول‌های I/O خارج از گذرگاه سیستم انجام می‌گیرد.

کنترلگر 8237A DMA اینتل

کنترلگر 8237A DMA برای انجام قابلیت‌های خود به پردازنده‌ی خانواده‌ی 8086 و حافظه‌ی DRAM متصل می‌شود. موقعیت مدول DMA در این اتصال در شکل ۱۴-۷ دیده می‌شود. هنگام نیاز مدول DMA به استفاده از گذرگاه‌های سیستم (داده، آدرس و کنترل) برای انتقال داده، سیگنالی به نام HOLD را به پردازنده ارسال می‌کند. پردازنده با سیگنال HLDA (hold acknowledge) پاسخ داده و مشخص می‌کند که مدول DMA می‌تواند گذرگاه‌ها را به کار ببرد. مثلاً اگر مدول DMA بخواهد بلوکی از داده را از حافظه به دیسک بفرستد، باید اعمال زیر اجرا شود:

۱. وسیله‌ی جانبی (مانند کنترلگر دیسک) تقاضای سرویس DMA را با بالابردن DREQ (DMA request) اعلام می‌کند.
۲. DMA یک سیگنال سطح بالا روی HRQ (hold request) خود قرار می‌دهد و استفاده از گذرگاه‌ها را از طریق پایه‌ی HOLD به CPU خبر می‌دهد.
۳. CPU سیکل گذرگاه فعلی (لزوماً دستور جاری نیست) را پایان داده و با بالابردن سطح HLDA به DMA پاسخ می‌دهد، و به 8237 DMA اطلاع می‌دهد که می‌تواند به جلو رفته و گذرگاه‌ها را برای انجام وظایفش جلو ببرد. مادامی که DMA در حال انجام وظایفش است، HOLD باید در سطح بالا فعال بماند.



شکل ۱۴-۷ استفاده از گذرگاه توسط 8237 DMA.

۴. DMA، خروجی DACK (DMA Acknowledge) را فعال می‌کند و به این ترتیب، شروع به انتقال داده را به دستگاه جانبی اعلام می‌کند.
۵. DMA انتقال داده از حافظه به دستگاه جانبی را آغاز می‌کند. به این ترتیب که آدرس اولین بایت از بلوک را روی گذرگاه آدرس گذاشته و MEMR را فعال می‌کند و لذا بایت را از حافظه به گذرگاه داده می‌آورد، و سپس سیگنال LOW را فعال می‌سازد تا آن را در وسیله‌ی جانبی بنویسد. آنگاه DMA شمارنده را کاهش و اشاره‌گر آدرس را افزایش داده و فرآیند را تا رسیدن شمارنده به صفر و اتمام وظیفه تکرار می‌کند.
۶. DMA پس از اتمام کارش ، HRQ را غیرفعال می‌کند و به CPU اطلاع می‌دهد که می‌تواند کنترل گذرگاه‌ها را کنترل نماید.

در زمان استفاده DMA از گذرگاه‌ها، پردازنده بیکار است. به‌طور مشابه، وقتی پردازنده از گذرگاه‌ها استفاده می‌کند، DMA بیکار است. 8237 DMA را کنترلگر DMA عبوری[1] می‌نامند. یعنی داده از یک مکان به مکان دیگر از داخل تراشه‌ی DMA عبور نمی‌کند و در تراشه‌ی DMA ذخیره نمی‌شود بلکه عبور داده از یک مکان به مکان دیگر را کنترل می‌نماید. بنابراین DMA فقط داده را بین پورت (درگاه) I/O و یک آدرس از حافظه منتقل می‌کند و نه اینکه بین دو پورت یا دو مکان حافظه انتقال را کنترل نماید. با این وجود همان‌طور که در ادامه توضیح داده شده است، DMA می‌تواند از طریق یک ثبات، انتقال از یک حافظه به حافظه‌ی دیگر را انجام دهد.

8237 ، 4 کانال DMA دارد که می‌توانند مستقلاً برنامه‌ریزی‌شده و هر لحظه فعال گردند. این کانال‌ها از 0 تا 3 شماره‌گذاری شده‌اند.

8237 دارای یک مجموعه ثبات کنترل / فرمان پنج‌تایی برای برنامه‌ریزی و کنترل اعمال DMA روی هر یک از کانال‌هایش می‌باشد (جدول ۲-۷).

- **فرمان**[2]: پردازنده این ثبات را برای کنترل عملکرد DMA بار می‌نماید. D0 یک انتقال حافظه به حافظه را فعال می‌کند که طی آن کانال 0 برای انتقال یک بایت به یک ثبات موقت 8237 به کار می‌رود و کانال 1 برای انتقال بایت از ثبات به حافظه مورد استفاده قرار می‌گیرد. وقتی کنترل حافظه به حافظه فعال شد، D1 می‌تواند برای غیرفعال‌کردن افزایش/کاهش روی کانال 0 به کار رود. به این ترتیب یک مقدار ثابت در یک بلوک از حافظه‌ی قابل نوشتن D2 ، DMA را فعال یا غیرفعال می‌کند.
- **وضعیت**: پردازنده این ثبات را برای تعیین وضعیت DMA می‌خواند. بیت‌های D0-D3 به کار می‌روند تا مشخص شود آیا کانال‌های 3 - 0 به انتهای شمارش TC[3] خود رسیده‌اند یا خیر. بیت‌های D4 تا D7 به وسیله‌ی پردازنده به این خاطر به کار می‌روند تا معین شود آیا کانالی یک تقاضای DMA دارد یا نه.

1. Fly-by 2. Command 3. Terminal Count

جدول ۲-۷ ثبات‌های 8237A اینتل.

بیت	فرمان	وضعیت	مد	ماسک تک	تمام ماسک
D0	حافظه به حافظه E/D	کانال 0 به انتهای شمارش رسیده TC	انتخاب کانال	بیت ماسک انتخاب کانال	بیت ماسک کانال 0، پاک / نشاندن
D1	نگهداری آدرس کانال 0 ، E/D	کانال 1 به انتهای شمارش رسیده TC			بیت ماسک کانال 1، پاک / نشاندن
D2	کنترلگر E/D	کانال 2 به انتهای شمارش رسیده TC	انتقال تأیید/نوشتن/خواندن	بیت ماسک پاک/ نشاندن	بیت ماسک کانال 2، پاک / نشاندن
D3	زمان‌بندی نرمال/فشرده	کانال 3 به انتهای شمارش رسیده TC		به کار نرفته	بیت ماسک کانال 3، پاک / نشاندن
D4	اولویت ثابت/چرخشی	تقاضای کانال 0	بازآغازی خودکار E/D		به کار نرفته
D5	انتخاب نوشتن بعدی/ توسعه‌یافته	تقاضای کانال 0	انتخاب افزایش / کاهش آدرس		
D6	حس DREQ فعال بالا/پایین	تقاضای کانال 0	انتخاب مد متوالی/ بلوکی/تک/ تقاضا		
D7	حس DACK بالا/پایین	تقاضای کانال 0			

E/D = فعال / غیرفعال TC = شمارش پایانه

- **مد:** پردازنده برای تعیین نوع عمل DMA این ثبات را 1 می‌کند. بیت‌های D0 و D1 برای انتخاب یک کانال مورد استفاده‌اند. بیت‌های دیگر انواع مدهای عملیاتی را برای کانال انتخابی، انتخاب می‌نمایند. بیت‌های D2 و D3 مشخص می‌کنند که آیا انتقال از یک وسیله‌ی I/O به حافظه است (نوشتن) یا از حافظه به I/O (خواندن)، یا یک عمل تأیید است. اگر D4 برابر 1 شود، آنگاه ثبات آدرس حافظه و ثبات شمارنده در انتهای انتقال داده با مقادیر اولیه‌ی خود بارگذاری مجدد می‌شوند. بیت‌های D6 و D7 طریقه‌ی کاربری 8237 را تعیین می‌کنند. در مد تکی[1]، یک بایت تک از داده انتقال می‌یابد. مدهای بلوک[2] و تقاضا[3] برای انتقال بلوکی به کار می‌روند، و در آن مد تقاضا اجازه‌ی زودهنگام را برای پایان دادن به انتقال می‌دهد. مد متوالی، امکان متوالی‌شدن چند 8237 را برای گسترش تعداد کانال‌ها به بیش از 4 ، فراهم می‌نماید.
- **پوشش تک:** پردازنده این ثبات را تنظیم می‌کند. بیت‌های D0 و D1 کانال انتخاب می‌شوند. بیت D2 بیت ماسک آن کانال (پوشش) را پاک (0) یا برقرار (1) می‌نماید. از طریق این ثبات، ورودی DREQ یک کانال خاص، ماسک (غیرفعال) یا غیرماسک (فعال) می‌شود. ضمن این‌که ثبات فرمان می‌تواند برای غیرفعال‌کردن تمام تراشه‌ی DMA به کار برود، ثبات ماسک تک به برنامه‌نویس اجازه می‌دهد تا یک کانال خاص را فعال یا غیرفعال کند.

1. Single mode 2. Block mode 3. Demand mode

- **تمام ماسک:** این ثبات مشابه با ثبات ماسک تک است با این تفاوت که همه‌ی 4 کانال با یک عمل نوشتن ماسک یا غیرماسک گردد.

علاوه‌بر امکانات فوق، 8237A دارای هشت ثبات داده است: یعنی یک ثبات آدرس حافظه و یک ثبات شمارش برای هر کانال. پردازنده این ثبات‌ها را برای مشخص‌کردن مکان سایز حافظه‌ی اصلی که توسط انتقال‌ها تحت تأثیر واقع می‌شوند، تنظیم می‌نماید.

۷-۶ کانال‌های I/O و پردازنده‌ها

تکامل عملکرد I/O

با تکامل سیستم‌های کامپیوتری، پیچیدگی و کارایی هر عنصر رو به افزایش گذاشته است. این پیشرفت هیچ‌کجا واضح‌تر از بخش I/O نیست. ما قبلاً بخشی از این تکامل را دیده‌ایم. گام‌های تکامل را می‌توان مطابق زیر خلاصه کرد:

۱. CPU مستقیماً یک وسیله‌ی جانبی را کنترل می‌کند. این نوع عمل در وسایل کنترل شده با ریزپردازنده‌ها دیده می‌شود.

۲. یک کنترلگر یا یک مدول I/O اضافه می‌شود. CPU از I/O برنامه‌نویسی‌شده بدون وقفه استفاده می‌کند. در این مقطع CPU از اتصال با وسایل خارجی که تا حدی پیچیده‌اند مبرّا و منفک است.

۳. آرایش مانند گام دوم است، ولی اکنون وقفه نیز به کار گرفته می‌شود. لازم نیست که CPU برای انجام عمل I/O منتظر مانده و وقت صرف کند و لذا بهره‌وری افزایش می‌یابد.

۴. به مدول I/O از طریق مدول DMA دستیابی مستقیم به حافظه فراهم می‌شود. اکنون مدول مذکور می‌تواند یک بلوک داده را به و یا از حافظه بدون دخالت CPU به جز در آغاز و پایان کار، انتقال دهد.

۵. مدول I/O با مجموعه دستورات خاصی که برای I/O در نظر گرفته شده تکمیل شده است تا حدی که خود نوعی پردازنده شده است. CPU، پردازنده‌ی I/O را برای اجرای برنامه‌ی I/O در حافظه هدایت می‌کند. پردازنده‌ی I/O بدون دخالت CPU این دستورات را برداشت کرده و اجرا می‌نماید. این موجب می‌گردد تا CPU یک رشته اعمال I/O را به آن اختصاص دهد و سپس فقط وقتی تمام رشته اجرا شد و پایان یافت، وقفه داده می‌شود.

۶. مدول I/O حافظه‌ی محلی خاص خود را دارد و در واقع خود یک کامپیوتر است. با این معماری مجموعه‌ی زیادی از وسایل I/O قابل کنترل خواهند بود و در آن درگیری CPU نیز حداقل است. کاربرد رایج این معماری عبارتست از کنترل تبادل داده با پایانه‌های محاوره‌ای. پردازنده‌ی I/O بخش عمده‌ی کار در کنترل پایانه را بر عهده می‌گیرد.

با توجه به این مسیر تکامل مشاهده می‌شود که رفته‌رفته عمل I/O بدون دخالت CPU انجام گردیده است. CPU به طور فزاینده‌ای وظایف مربوط به I/O را رها کرده و عملکردش اصلاح می‌گردد.

در دو گام آخر (۵ تا ۶)، بیشترین تغییر با معرفی مفهوم مدول I/O که قادر به اجرای برنامه است رخ می‌دهد. در گام ۵، مدول I/O اغلب به *کانال I/O* معروف است. در گام ۶ اغلب *پردازنده‌ی I/O* به کار می‌رود. در هر صورت هر دو نام در مواردی به هر دو وضع اتلاق می‌گردند. در ادامه ما از جمله *کانال I/O* استفاده خواهیم کرد.

مشخصه‌های کانال I/O

کانال I/O تکمیل مفهوم DMA است. یک کانال I/O قادر است دستورالعمل‌ها را اجرا کند و به این ترتیب به آن کنترل کامل عملکرد I/O داده می‌شود. در یک سیستم کامپیوتر با چنین وسایلی، CPU دستورات I/O را اجرا نمی‌کند. این دستورات در حافظه‌ی اصلی ذخیره می‌شود تا به وسیله‌ی پردازنده‌ی خاصی در کانال I/O اجرا گردند. بنابراین CPU به کانال I/O فرمان می‌دهد تا برنامه‌ای در حافظه را اجرا کند و به این ترتیب انتقال I/O آغاز می‌شود. نقش برنامه تعیین دستگاه یا دستگاه‌ها و محل یا محل‌های حافظه برای ذخیره‌سازی، اولویت و اقدامات لازم برای تشخیص خطاهای ممکن است. کانال I/O این دستورات را دنبال کرده و انتقال داده را انجام می‌دهد.

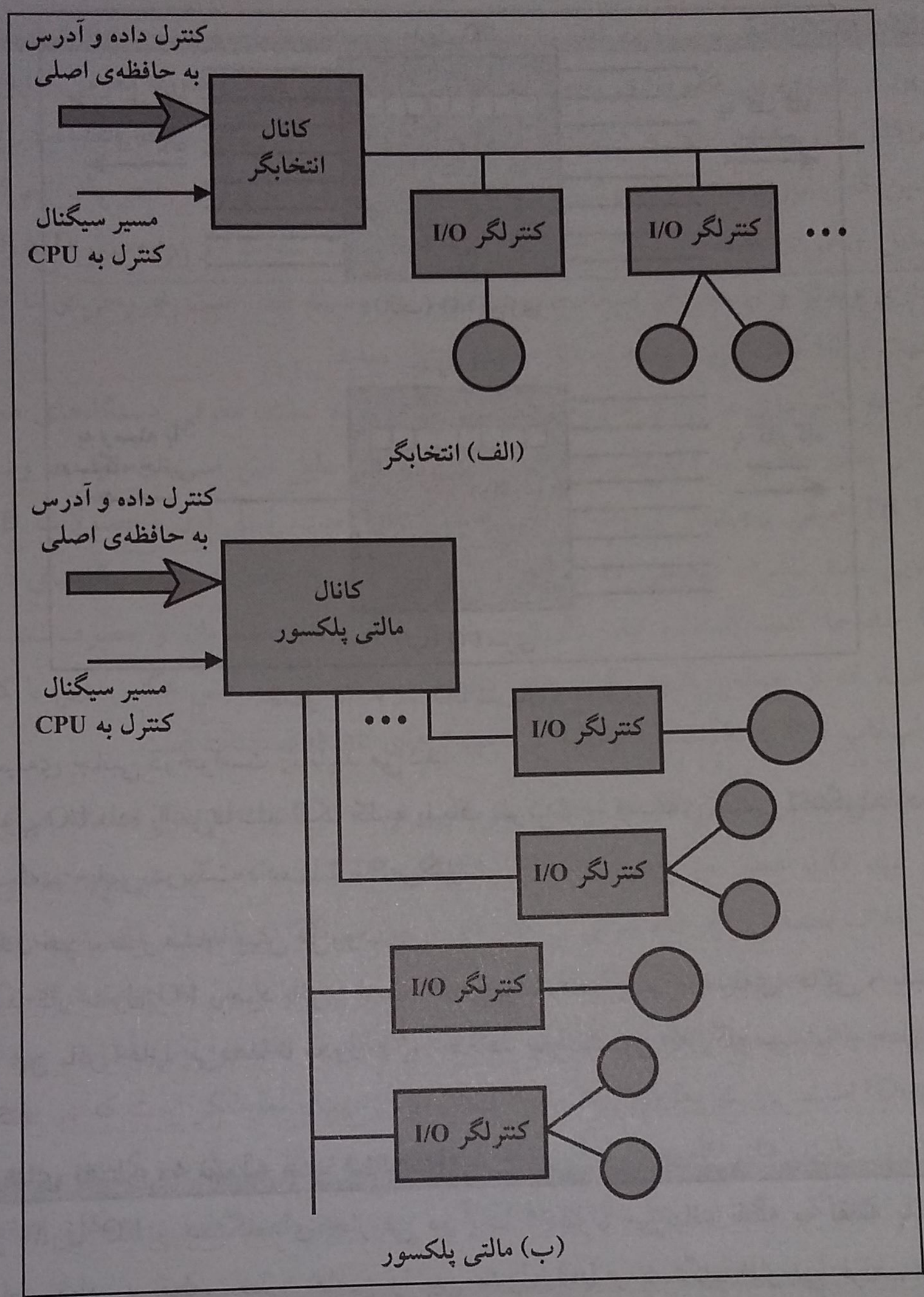
دو نوع کانال I/O رایج است (شکل ۱۵-۷). *کانال انتخابگر*[1] وسایل سریع چندگانه را کنترل می‌کند و هر بار داده یکی از آن‌ها را انتقال می‌دهد. بنابراین کانال I/O یکی از دستگاه‌ها را انتخاب نموده و انتقال داده را انجام می‌دهد. هر دستگاه یا مجموعه‌ی کوچکی از دستگاه‌ها به وسیله‌ی یک کنترلگر یا مدول I/O رسیدگی می‌شود که بسیار شبیه مدول‌های I/O مورد بحث است. بنابراین کانال I/O به جای CPU در کنترل این کنترلگرهای I/O سرویس‌دهی می‌نماید. یک *کانال مالتی‌پلکسور*[2] قادر است I/O چند دستگاهی را به طور همزمان رسیدگی کند. در دستگاه‌های کم سرعت، یک مالتی‌پلکسور بایتی کاراکترها را پذیرفته یا با حداکثر سرعت به دستگاه‌های چندگانه ارسال می‌دارد. مثلاً، رشته‌ی کاراکترهای حاصل از سه دستگاه با سرعت متفاوت با رشته‌های خاص $A_1 A_2 A_3 A_4 \ldots$ ، $B_1 B_2 B_3 B_4 \ldots$ و $C_1 C_2 C_3 C_4 \ldots$ ممکن است به صورت $A_1 B_1 C_1 A_2 C_2 A_3 B_2 C_3 A_4$ و غیره باشد. در دستگاه‌های پرسرعت، مالتی‌پلکسور بلوکی، بلوک‌های داده را از چندین دستگاه برمی‌چیند.

۷-۷ واسط خارجی THUNDERBOLT و INFINIBAND

انواع واسط‌ها

ارتباط به یک دستگاه جانبی از یک مدول I/O باید با توجه به طبیعت و عملکرد وسیله‌ی جانبی سازگار باشد. مشخصه‌ی مهمی از مدار واسط این‌است‌که آیا نوع آن سریال است یا موازی (شکل ۱۶-۷). در ارتباط با **واسط موازی**، چندین خط رابط بین مدول I/O و وسیله‌ی جانبی وجود داشته و چندین بیت به‌طور همزمان انتقال می‌یابد و این انتقال درست مثل کلمه‌ای است که به طور همزمان از گذرگاه

1. Selector channel 2. Multiplexor

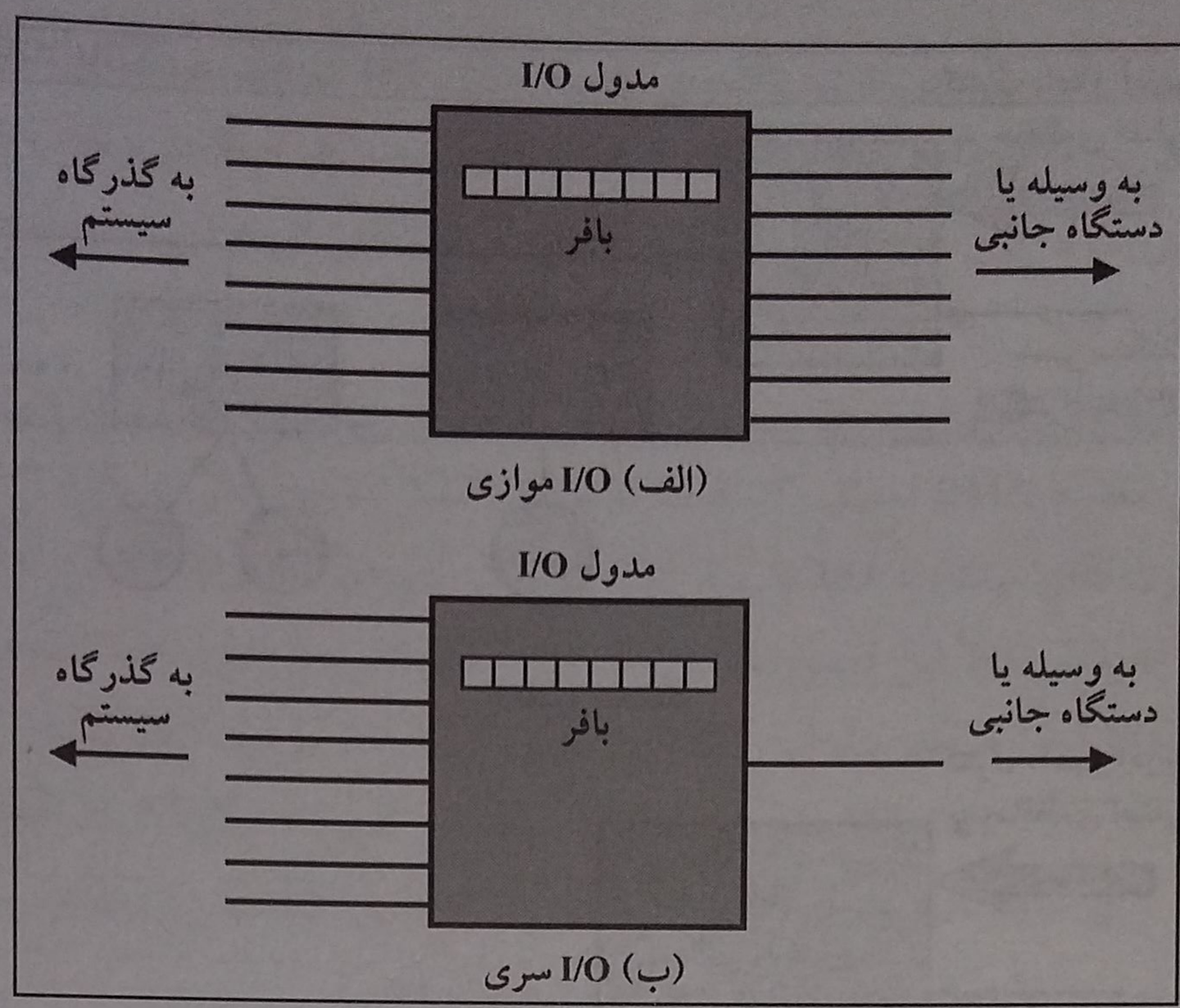


شکل ۱۵-۷ معماری کانال I/O.

داده انتقال می‌یافت. در ارتباط با **واسط سریال** فقط یک خط برای انتقال داده وجود دارد و هر بار یک بیت منتقل می‌شود. رابط یا واسط موازی از آغاز برای وسایل جانبی پرسرعت مثل نوار و دیسک به‌کار رفته است درحالی‌که واسط سریال در وسایلی چون چاپگرها و پایانه‌ها به کار گرفته شده است. با معرفی واسط‌های پر سرعت نسل جدید، واسط‌های موازی کمتر مورد استفاده قرار می‌گیرند.

در هر حال مدول I/O باید با وسیله‌ی جانبی وارد محاوره گردد. به‌طورکلی محاوره برای عمل نوشتن چنین است:

۱. مدول I/O سیگنال کنترلی ارسال و اجازه‌ی ارسال داده را درخواست می‌نماید.



شکل ۱۶-۷ I/O سریال و موازی.

۲. وسیله‌ی جانبی درخواست را تأیید می‌کند.

۳. مدول I/O داده را می‌فرستد (یک کلمه یا یک بلوک و به وسیله‌ی جانبی بستگی دارد).

۴. وسیله‌ی جانبی دریافت داده را تأیید می‌کند.

عمل خواندن هم به‌طور مشابه پیش می‌رود.

کلید کار مدول I/O وجود بافری است که می‌تواند داده را بین وسیله‌ی جانبی و بقیه‌ی سیستم پاس دهد. این بافر اجازه می‌دهد تا مدول I/O اختلاف سرعت بین گذرگاه سیستم و خطوط خارجی را جبران کند.

آرایش‌های نقطه به نقطه و چندنقطه‌ای

اتصال بین مدول I/O و دستگاه‌های خارجی در یک کامپیوتر می‌تواند نقطه به نقطه یا چندنقطه‌ای باشد. ارتباط نقطه به نقطه، خطی خاص را بین مدول I/O و دستگاه خارجی فراهم می‌سازد. در سیستم‌های کوچک مانند PCها و ایستگاه‌های کاری، ارتباط نوع نقطه به نقطه شامل صفحه‌کلید، چاپگر و مودم خارجی است. مثال نمونه‌ای از چنین واسطی عبارتست از مشخصات EIA-232 (برای توضیح بیشتر [STAL04] ملاحظه شود).

نوع مهم‌تر، اتصال چند نقطه‌ی خارجی است، که برای پشتیبانی از دستگاه‌های ذخیره‌ساز حجیم خارجی به کار گرفته می‌شود (مثل راه‌اندازهای دیسک و نوار) و دستگاه‌های چندرسانه‌ای (CD-ROM ، ویدیو و وسایل صوتی). این واسطهای چندنقطه‌ای در واقع گذرگاه‌های خارجی‌اند، و منطقی مشابه با گذرگاه‌های فصل ۳ ارائه می‌نمایند. در این بخش ما به دو مثال کلیدی نگاه می‌کنیم که عبارتنداز Thunderbolt و InfiniBand .

Thunderbolt (TB)

جدیدترین و سریع‌ترین تکنولوژی اتصال که برای کاربردهای همه منظوره معرفی شده است، TB می‌باشد. TB به وسیله‌ی اینتل و همکاری آپل ساخته شد. TB می‌تواند کارهایی را مدیریت کند که قبلاً چندین کابل نیاز داشتند. تکنولوژی، اتصال داده، ویدیو، صوت و برق را در یک اتصال پرسرعت برای وسایل جانبی‌ای مانند درایوهای هارد (گرداننده‌ی دیسک سخت) و آرایه‌های RAID، جعبه‌های نسخه‌برداری ویدئو و واسط‌های شبکه جای داده‌اند. این وسیله قادر است خروجی‌ای تا 10 Gbps را در هر جهت و 10 وات انرژی را به وسایل جانبی منتقل نماید.

گرچه تکنولوژی و ویژگی‌های مربوطه‌اش پایدار شده است، معرفی دستگاه‌های مجهز به TB در بازار در حال حاضر به آهستگی در حال تکامل است. دلیل این است که واسط وسایل جانبی سازگار با TB خیلی پیچیده‌تر از یک دستگاه USB ساده است. نسل اول محصولات TB در بازار حرفه‌ای‌هایی مانند ناشران شخصی که می‌خواهند حجم بالایی از داده را بین دستگاه‌های ذخیره‌ساز و لپ‌تاپ‌ها جابه‌جا کنند، استفاده شد. باارزان‌تر شدن تکنولوژی، مشتریان و مصرف‌کنندگان زیادی خواهد یافت، که از جمله‌ی آن‌ها بک‌آپ‌های داده‌ی پرسرعت و ویرایش عکس‌هایی با کیفیت. TB استاندارد لپ‌تاپ Apple's MacBook Pro و کامپیوترهای iMac دسکتاپ است.

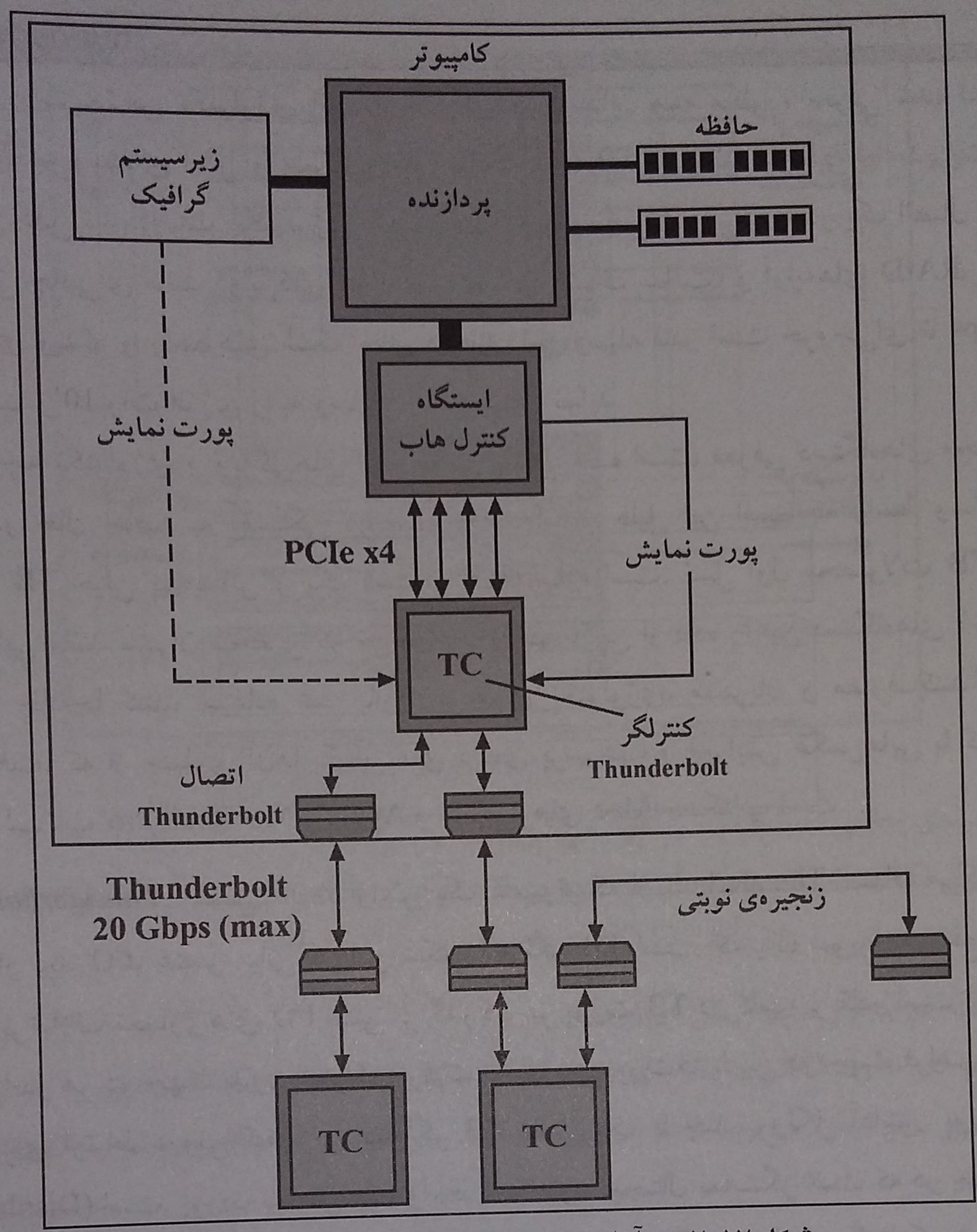
آرایش Thunderbolt شکل ۱۷-۷ آرایش یک کامپیوتر که از Thunderbolt استفاده می‌کند را نشان می‌دهد. از دید I/O، عنصر میانی در این شکل **کنترلگر TB** است، که یک سوییچ متقاطع[1] با کارایی بالاست. برخلاف معماری‌های I/O مبتنی بر گذرگاه هر پورت TB در کامپیوتر قادر است انتقال داده‌ی سریعی را در هر دو جهت بدون اشتراک ظرفیت انتقال بین پورت‌ها یا بین دو جهت فراهم کند.

برای ارتباط درونی کامپیوتر، کنترلگر TB شامل یک یا چند پروتکل آداپتور **پورت نمایش** (Display Port) است. پورت نمایش یک واسط استاندارد دیجیتال نمایشگر است که در حال حاضر به طور گسترده در مانیتورهای کامپیوتر، نمایشگرهای لپ‌تاپ و دیگر واسط‌های گرافیک و ویدئو مورد استفاده است. کنترلگر یک **سوییچ PCI Express** با چهار **پورت آداپتور پروتکل PCI Express** می‌باشد که در ارتباطات داخلی به کار می‌رود.

کنترلگر TB دستیابی به وسایل بیرونی از طریق یک یا چند **کانکتور TB** را فراهم می‌نماید، و هر کانکتور می‌تواند تا 10 Gbps را در هر جهت ارسال کند. همان کانکتور می‌تواند برای کابل‌های الکتریکی و نوری استفاده شود. کابل الکتریکی می‌تواند 3 متر طول داشته باشد، در حالی که کابل نوری می‌تواند تا 10 متر باشد.

کاربر می‌تواند وسایل جانبی با کارایی بالا را از طریق یک کابل به PC وصل نماید، که در آن یکی پس از دیگری و تا 7 دستگاه متصل می‌گردد، که 1 یا 2 از آن می‌تواند از نوع نمایشگرهای

1. Crossbar

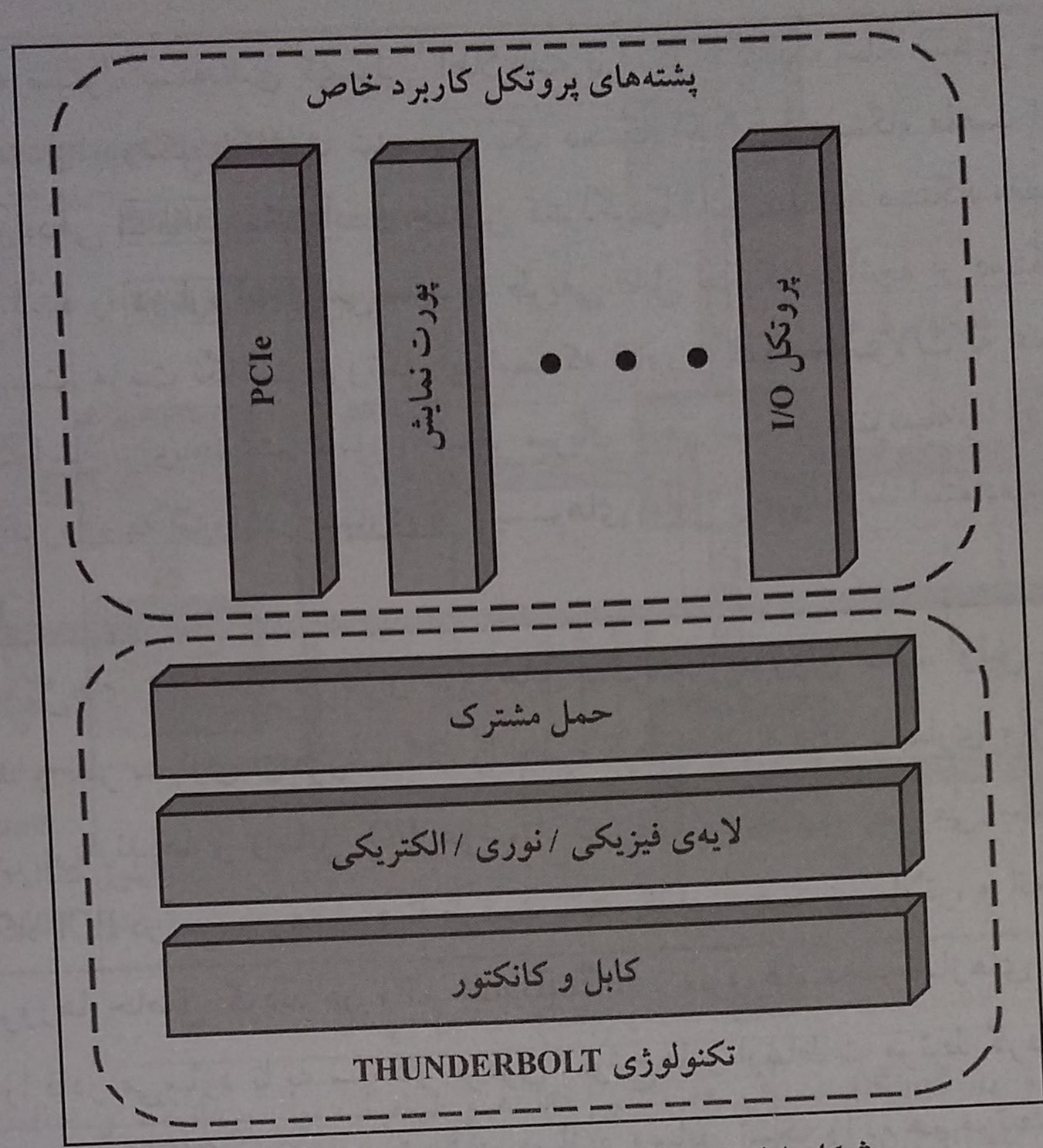


شکل ۱۷-۷ آرایش کامپیوتر نمونه با Thunderbolt .

پورت نمایش (Display Port) باشد (بسته به آرایش کنترلگر در PC میزبان) چون تکنولوژی TB دو کانال تمام - باند را حمل می‌کند، کاربر می‌تواند پهنای باند زیاد را نه تنها در دستگاه متصل اول بلکه در دیگر دستگاه‌ها تا انتها را نیز حمل نماید.

معماری پروتکل TB شکل ۱۸-۷ معماری پروتکل TB را شرح می‌دهد. **لایه‌ی کابل و کانکتور** دستیابی محیط انتقال را فراهم می‌سازد. این لایه صفات فیزیکی و الکتریکی پورت کانکتور را مشخص می‌سازد.

لایه‌ی فیزیکی پروتکل TB مسئول نگهداری پیوند از جمله تشخیص hot-Plug و انکد کردن داده برای تهیه‌ی انتقال داده بسیار مؤثر است. لایه‌ی فیزیکی برای معرفی سرآمد بسیار کم و دوپلکس کامل 10 Gbps از ظرفیت قابل استفاده برای لایه‌ی بالاتر است.



شکل ۱۸-۷ لایه‌های پروتکل Thunderbolt .

لایه‌ی حمل مشترک کلید کار TB است و همان چیزی است که آن را به عنوان یک تکنولوژی I/O جانبی سرعت بالا مطرح کرده است. بعضی از ویژگی‌ها عبارتست از:

- معماری تعویض، سوییچینگ، توان پایین، کارایی بالا.
- فرمت بسته‌ی سرآمد پایین، کارایی بالا با پشتیبانی سرویس کیفیت (QoS) انعطاف‌پذیر، که امکان مالتی‌پلکس کردن صورت وضعیت PCI Express یک‌جا را فراهم می‌نماید و مخابره‌ی پورت نمایش در همان پیوند صورت می‌گیرد. لایه‌ی حمل قابلیت تخصیص پهنای باند را با استفاده از اولویت و مکانیزم رزرواسیون دارد.
- استفاده از بسته‌هایی با سایز کوچک برای دستیابی به تأخیر کوچک.
- استفاده از کنترل جریان اعتبار ـ محور برای داشتن بافر با سایز کوچک.
- معماری متفاوتی که توپولوژی‌های انعطاف‌پذیر را پشتیبانی می‌کند (ستاره، درخت، زنجیری و غیره) و ارتباط همتا به همتا را فراهم می‌کند.
- پروتکل هم‌گامی در زمان جدید که اجازه می‌دهد محصولات TB در فضایی وصل شوند تا زمان‌شان را در محدوده‌ی 8 ns از یکدیگر هم‌گام نمایند.

لایه‌ی کاربرد حاوی پروتکل‌های I/Oای است که در لایه‌ی حمل نگاشت شده‌اند. در ابتدا TB پشتیبانی کاملی برای PCIe و پورت نمایش فراهم می‌سازد. این کار به وسیله‌ی یک آداپتور پروتکل

انجام می‌شود، که مسئول بسته‌بندی کپسولی[1] اطلاعات پروتکل نگاشت شده (تبدیل شده) در لایه‌ی حمل است. بسته‌های پروتکل نگاشت شده بین یک دستگاه مبداء و دستگاه مقصد است که ممکن است از مسیری راه‌دهی شده و ممکن است چندین کنترلگر را قطع کند. در دستگاه مقصد، یک آداپتور پروتکل نگاشت شده را دوباره ایجاد می‌نماید به طریقی قابل تفکیک با آنچه از دستگاه مبداء رسیده قابل تشخیص نیست. مزیت نگاشت پروتکل این‌است‌که فناوری TB محصولات به عنوان یک دستگاه PCIe یا پورت نمایش برای سیستم عامل کامپیوتر میزبان ظاهر شوند، بدینوسیله ما را قادر می‌سازد از گردانندگان استانداردی که امروز در بسیاری از سیستم‌های عامل وجود دارند استفاده نماییم.

InfiniBand

InfiniBand ویژگی‌های I/O اخیر در بازار سِرورهای پیشرفته‌ی امروزی است. اولین بار در اوایل سال ۲۰۰۱ اعلام شد و نظر بسیاری از فروشندگان را جلب کرد. استاندارد آن معماری ویژگی‌هایی را برای جریان داده بین پردازنده‌ها و وسایل I/O هوشمند تعریف می‌کند. به نظر می‌رسد که InfiniBand جایگزین گذرگاه PCI در سرور رها شود تا ظرفیت بستر، قابلیت گسترش بالاتر، و انعطاف پیشرفته‌تری در طراحی سرورها حاصل گردد. در واقع InfiniBand ، سرورها، ذخیره‌سازهای راه دور، و دیگر وسایل شبکه را قادر می‌سازد تا به ساختار مرکزی سوئیچ‌ها و ارتباطات مرتبط گردند. معماری مبتنی بر سوئیچ می‌تواند تا 64000 سرور، سیستم ذخیره‌ساز و وسایل شبکه را به هم مرتبط سازد.

معماری InfiniBand (IB)

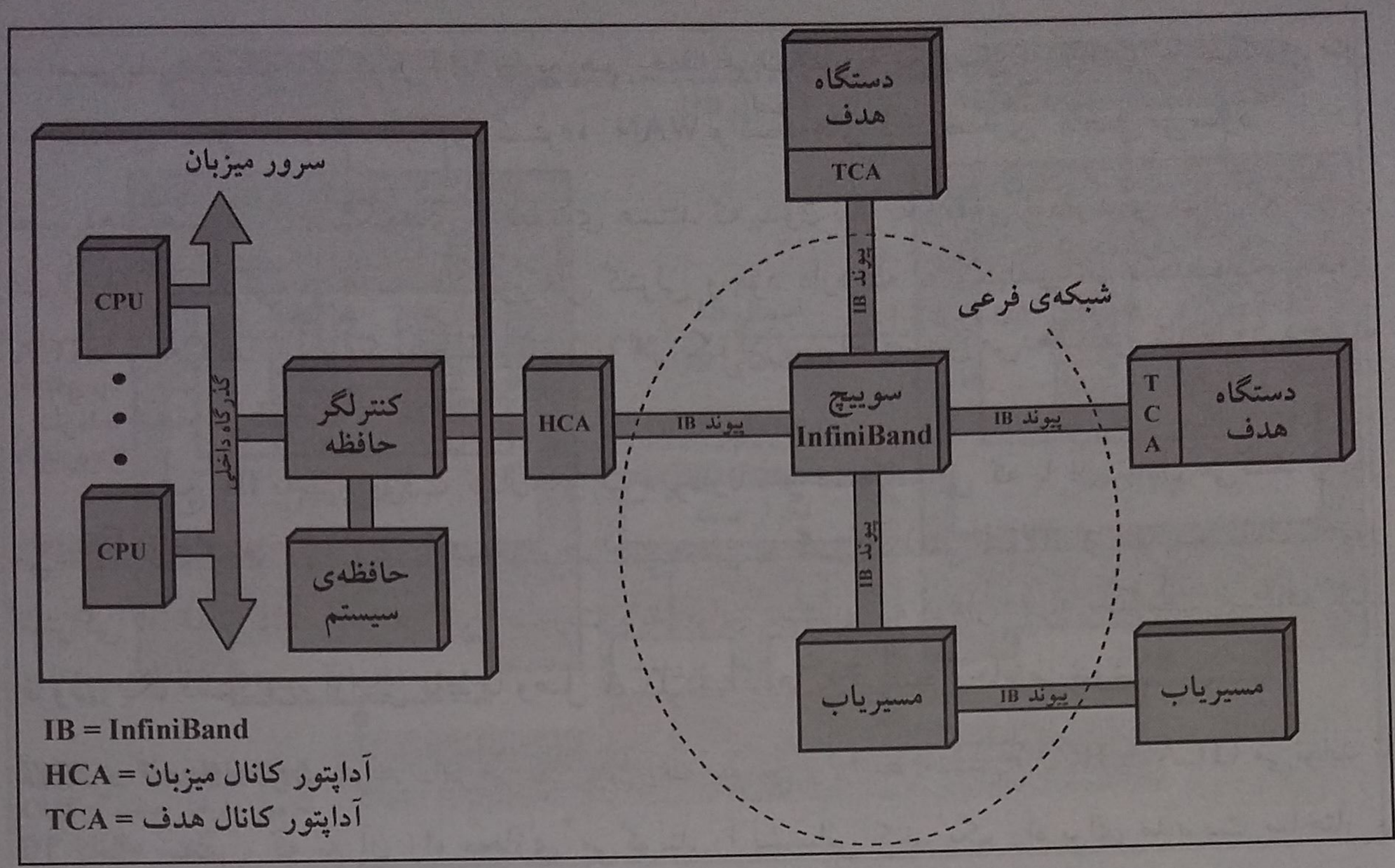
گرچه PCI روش درون ارتباطی مطمئنی است و در ایجاد سرعت بیشتر، تا 4 Gbps در حال پیشروی است، ولی معماری آن در مقایسه با IB محدود است. با IB لازم نیست که در شاسی سرور سخت‌افزار واسط I/O اصلی را داشته باشیم. با IB ذخیره‌سازهای راه دور، شبکه و اتصالات بین سرورها با اتصال همه‌ی وسایل به ساختار مرکزی سوئیچ‌ها و ارتباط‌ها ایجاد می‌گردد. حذف I/O از شاسی سرور امکان افزایش بیشتر چگالی سرور را فراهم کرده و مرکز داده انعطاف‌پذیرتر و مقیاس‌پذیرتری فراهم می‌گردد. به هر تعداد گره‌ی مستقل که نیاز باشد قابل گسترش است.

برخلاف PCI ، که فاصله از بُرد اصلی CPU را به سانتی‌متر اندازه می‌گیرد، طراحی کانال IB وسایل I/O را قادر می‌سازد که با کابل مسی به فاصله‌ی 17 متر و با فیبر نوری چند مُد[2] تا 300 متر و با فیبر نوری تک مد[3] تا 10 کیلومتر قرار بگیرند. سرعت انتقال بالغ بر 30 Gbps می‌تواند باشد.

شکل ۱۹-۷ معماری IB را تشریح می‌کند. عناصر مهم عبارتنداز:

- **تطبیق‌دهنده‌ی کانال میزبان[4] (HCA):** در عوض چند شکاف PCI ، یک سرور تنها به یک واسط به HCA نیاز دارد تا سرور را به سوئیچ IB متصل سازد. HCA به عنوان کنترلگر حافظه به سرور

1. Encapsulation 2. Multimode 3. Single mode 4. Host Channel Adapter



شکل ۱۹-۷ ساختار سوئیچ IB.

وصل می‌شود و به گذرگاه سیستم دسترسی داشته و ترافیک بین پردازنده و حافظه و HCA و حافظه را کنترل می‌نماید. HCA از دستیابی مستقیم به حافظه برای خواندن و نوشتن حافظه استفاده می‌کند.

- **تطبیق‌دهنده‌ی کانال هدف[1] (TCA):** TCA برای اتصال یک وسیله‌ی ذخیره‌ساز سیستم، مسیریاب[2]، راه‌یاب‌ها یا دیگر وسایل جانبی به سوئیچ IB به کار می‌رود.
- **سوییچ IB:** یک سوییچ اتصال فیزیکی نقطه به نقطه را برای تعدادی دستگاه فراهم می‌کند و مجموعه‌ای از سوییچ‌ها یک رابط را به دیگری وصل می‌نماید. سرورها و دستگاه‌های دیگر از طریق تطبیق‌دهنده‌های‌شان و از مسیر سوییچ تبادل داده می‌کنند. هوشمندی سوییچ، اتصالات را بدون وقفه به عمل سرور به وجود می‌آورد.
- **پیوندها:** رابط بین یک سوییچ و یک تطبیق‌دهنده‌ی کانال یا بین دو سوییچ است.
- **شبکه‌ی فرعی:** یک شبکه‌ی فرعی یا جزیی متشکل از یک یا چند سوییچ مرتبط با هم به علاوه‌ی رابط‌هایی است که دیگر دستگاه‌ها را به آن سوییچ‌ها وصل می‌کند. شکل ۱۹-۷ یک شبکه‌ی فرعی را با یک سوییچ نشان می‌دهد ولی شبکه‌های فرعی پیچیده‌تری برای ارتباط تعداد بیشتری از دستگاه‌ها مورد نیاز است. شبکه‌ی فرعی اجازه می‌دهد تا مدیر انتقال‌های دوردست و متوسط در داخل شبکه‌ی فرعی را محدود کند.

---

1. Target Channel Adapter 2. Router

- **مسیریاب:** شبکه‌های فرعی IB را به هم وصل می‌نماید، یا یک سوییچ IB را به شبکه‌ای مثل شبکه‌ی محلی[1] LAN ، شبکه‌ی گسترده[2] WAN و شبکه‌های ذخیره‌سازی[3] متصل می‌سازد.

تطبیق‌دهنده‌های کانال دستگاه‌های هوشمندی هستند که بدون نیاز به وقفه‌ی پردازنده‌ی سرور به همه‌ی اعمال I/O رسیدگی می‌کنند. مثلاً پروتکل کنترلی وجود دارد که به وسیله‌ی آن یک سوییچ همه‌ی TCAها و FCAها را در یک محیط یافته و به هر یک آدرسی اختصاص می‌دهد. این کار بدون دخالت پردازنده انجام می‌شود.

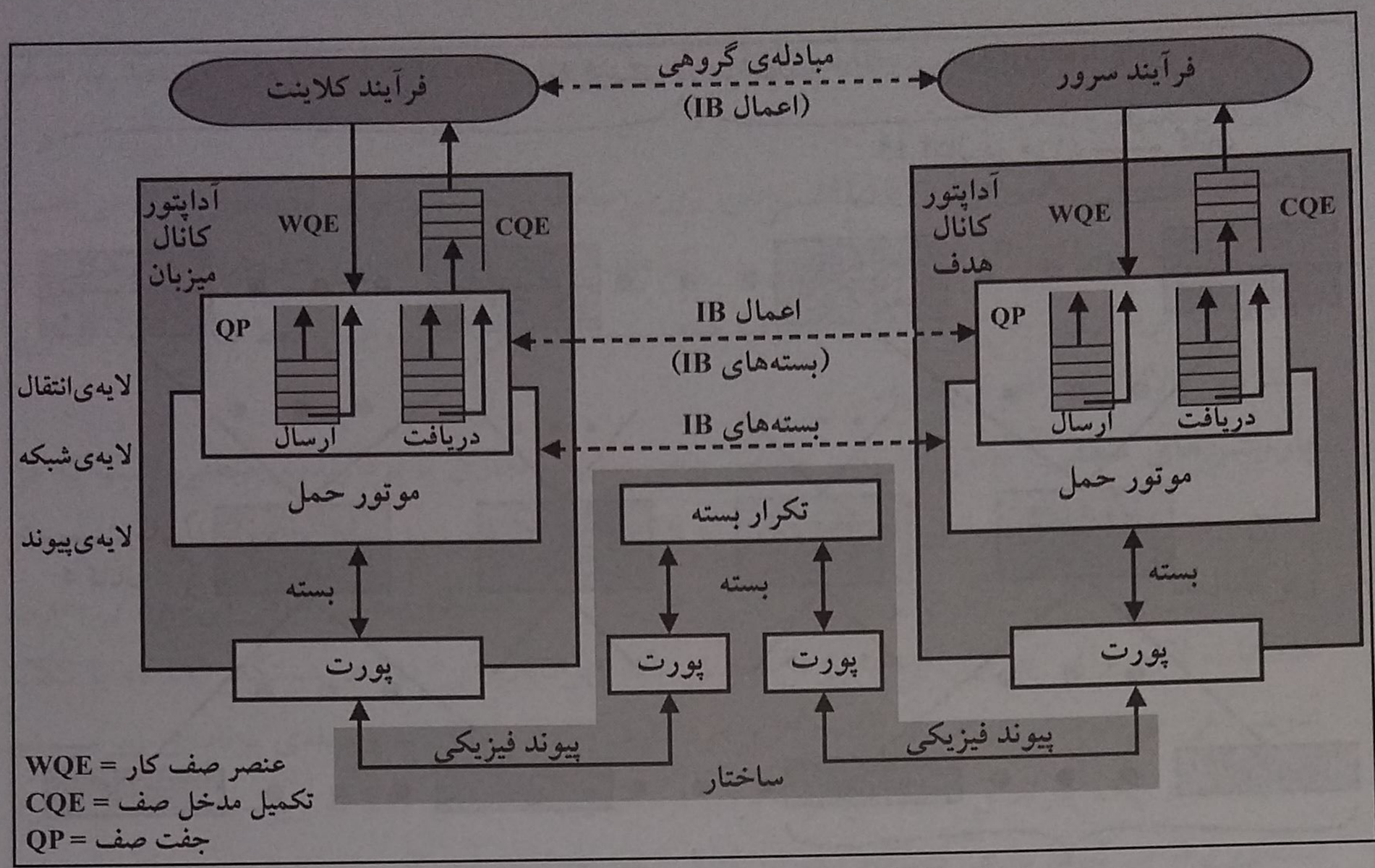
سوئیچ IB به‌طور موقت کانال‌های بین پردازنده و دستگاه‌هایی که با آن تبادل می‌کنند را باز می‌نماید. دستگاه‌ها لازم نیست همچون طراحی مبتنی بر گذرگاه (مثل PCI)، از ظرفیت کانال به‌طور اشتراکی استفاده کنند زیرا در این صورت باید برای دستیابی به پردازنده از حکمیّت استفاده کند. افزودن یک دستگاه به آرایش فقط با وصل TCA دستگاه‌ها به سوییچ انجام می‌شود.

عملکرد InfiniBand هر ارتباط فیزیکی بین یک سوییچ و واسط متصل (HCA یا TCA) می‌تواند تا 16 کانال منطقی، که به آن **راه مجازی**[4] می‌گویند را پشتیبانی کند. یک راه برای مدیریت ساختار و دیگر راه‌ها برای انتقال داده به کار می‌روند. داده به صورت بسته‌های متوالی ارسال می‌گردد و هر بسته حاوی بخشی از داده‌ی مورد انتقال، به علاوه آدرس‌دهی و کنترل اطلاعات است. پس مجموعه‌ای از پروتکل‌های تبادل داده برای مدیریت انتقال داده به کار گرفته می‌شود. یک راه مجازی به طور موقت به انتقال داده از یک گره‌ی انتهایی به دیگری از طریق ساختار IB، اختصاص می‌یابد. سوییچ IB ترافیک را از یک راه رسیده به راه رونده نگاشت می‌کند تا داده را بین نقاط انتهایی موردنظر راهدهی کند.

شکل ۲۰-۷ ساختار منطقی به کار رفته برای پشتیبانی تبادل در IB را نشان می‌دهد. باید در نظر داشت که بعضی از دستگاه‌ها می‌توانند داده را سریع‌تر از بافرهای موقت ارسال داشته و صفی از داده در مدخل ورودی یا در خروجی ایجاد کنند. صف‌ها می‌توانند در تطبیق‌دهنده‌ی کانال یا در حافظه‌ی دستگاه‌های متصل قرار گیرند. برای هر راه مجازی یک جفت صف جداگانه به کار می‌رود. کامپیوتر میزبان از این صف‌ها به طریق زیر استفاده می‌کند. میزان سیگنالی را به نام صف کار[5] (WQE) روی هر یک از دو صف ارسال[6] و دریافت[7] قرار می‌دهد. مهم‌ترین WQEها عبارتنداز اعمال RECEIVE و SEND . در یک عمل SEND ، WQE بلوکی از داده را در فضای حافظه‌ی دستگاه برای سخت‌افزار مشخص می‌کند تا به مقصد بفرستد. در RECEIVE ، WQE به سخت‌افزار می‌گوید که داده‌ی رسیده از یک دستگاه دیگر را وقتی مصرف‌کننده یک عمل SEND اجرا می‌کند، در کجا بنشاند. تطبیق‌دهنده‌ی کانال هر WQE ارسالی را در یک ترتیب اولویت‌دار پردازش کرده و یک اطلاعات تکمیل صف[8] (CQE) برای مشخص‌کردن حالت تکمیل تولید می‌کند.

---

1. Local Area Network 2. Wide Area Network 3. storage area network 4. virtual lane
5. Work Queue Entry 6. Send 7. Receive 8. Completion Queue Entry



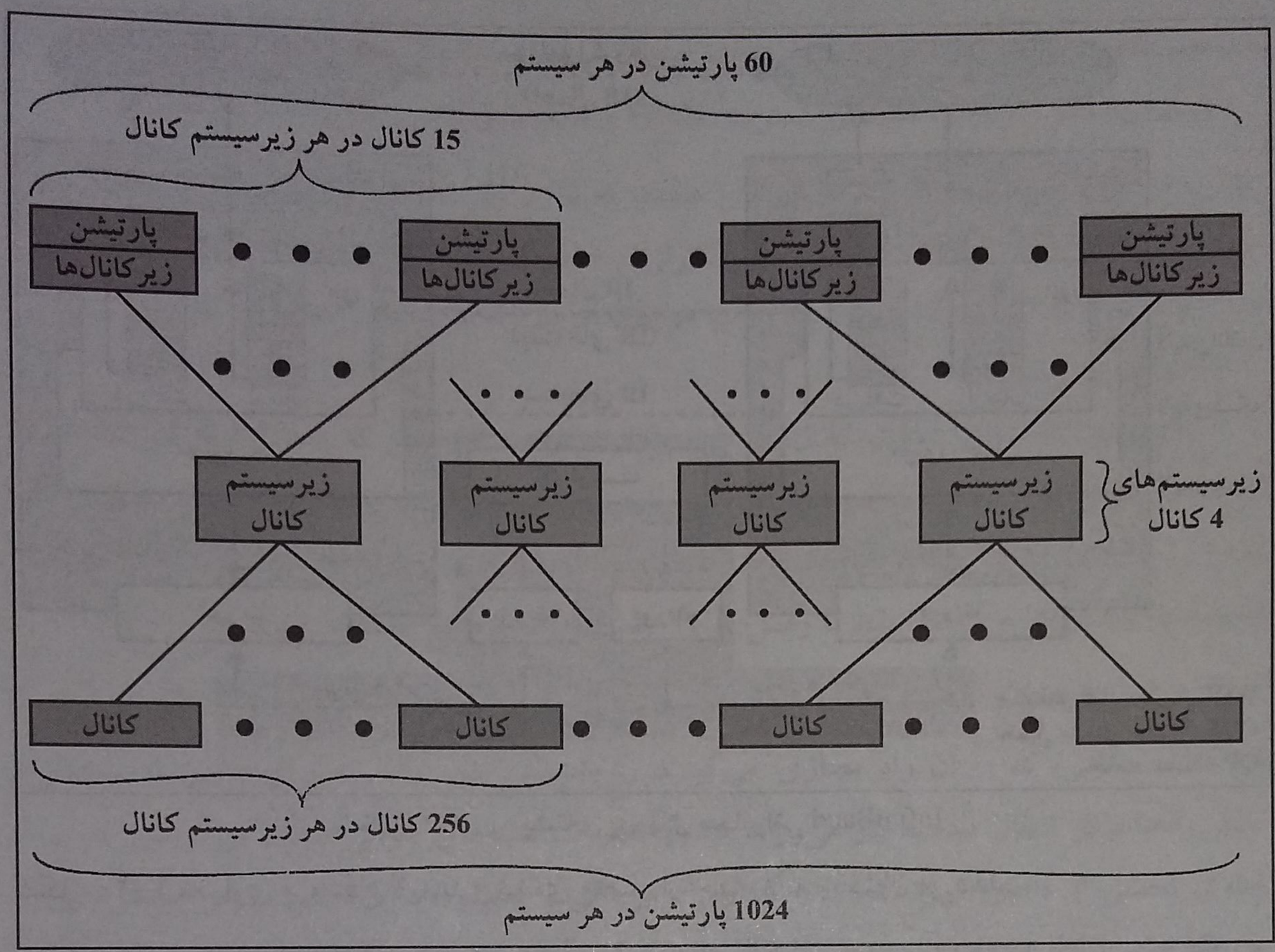
شکل ۲۰-۷ پشته‌ی پروتکل مخابراتی InfiniBand .

شکل ۲۰-۷ معماری پروتکل لایه‌بندی‌شده‌ی متصل از چهار لایه را نشان می‌دهد:

- **فیزیکی:** مشخصات لایه‌ی فیزیکی سه سرعت پیوند (1X ، 4X و 12X) را تعریف می‌کند که به‌ترتیب 10.25 و 30 Gbps را برای انتقال ایجاد می‌نمایند (جدول ۳-۷). لایه‌ی فیزیکی، محیط فیزیکی را هم تعریف می‌کند که از آن جمله مس و فیبر نوری است.
- **پیوند:** این لایه ساختار بسته‌ی پایه‌ی به کار رفته برای تبادل داده، از جمله طرح آدرس‌دهی یک آدرس پیوند منحصربه‌فرد برای هر دستگاه در یک شبکه‌ی فرعی را تعریف می‌کند. این سطح شامل مداری منطقی برای تنظیم راه‌های مجازی و سوییچینگ داده از طریق سوییچ‌ها از منبع به مقصد در یک شبکه‌ی فرعی است. ساختار بسته شامل یک کد تشخیص خطا برای ایجاد قابلیت اطمینان است.
- **شبکه:** لایه‌ی شبکه بسته‌های بین شبکه‌های فرعی IB مختلف را راهدهی می‌کند.
- **حمل:** لایه‌ی حمل راهکار اطمینان، برای انتقال انتها ـ به ـ انتهای بسته‌ها در طول یک یا چند شبکه‌ی فرعی را فراهم می‌سازد.

جدول ۳-۷ پیوندهای InfiniBand و توان عملیاتی داده‌ها.

پیوند	سرعت سیگنال (یک جهته)	ظرفیت قابل استفاده (80% میزان سیگنال)	توان عملیاتی داده‌ی مؤثر (ارسال + دریافت)
1 - wide	2.5 Gbps	2 Gbps (250 Mbps)	(250 + 250) Mbps
4 - wide	10 Gbps	8 Gbps (1 Gbps)	(1 + 1) Gbps
12 - wide	30 Gbps	24 Gbps (3 Gbps)	(3 + 3) Gbps



شکل ۲۱-۷ ساختار زیرسیستم کانال IBM z196 I/O .

۸-۷ ساختار IBM zENTERPRISE 196 I/O

zEnterprise 196 آخرین محصول مین‌فریم کامپیوتری است که در سال ۲۰۱۰ معرفی شد. سیستم براساس تراشه‌ی z196 طراحی شده است، که یک تراشه با 4 هسته‌ی 5.2 GHz می‌باشد. ساختار z196 می‌تواند حداکثر 24 تراشه‌ی پردازنده و جمعاً 96 هسته داشته باشد. در این بخش به ساختار I/O از zEnterprise 196 نگاه خواهیم کرد.

ساختار کانال

zEnterprise 196 (EP) یک زیرسیستم I/O اختصاصی دارد که همه‌ی اعمال I/O را اداره می‌نماید که کاملاً آن را بدون بارگذاری و تحمیل حافظه بر پردازنده‌های اصلی می‌کند. شکل ۲۱-۷ ساختار منطقی زیرسیستم I/O را نشان می‌دهد. از 96 هسته‌ی پردازنده، چهار عدد به I/O می‌تواند اختصاص یابد، که چهار **زیرسیستم کانال (CSS)** ایجاد خواهد شد. هر CSS از اجزای زیر ساخته شده است:

- **پردازنده‌ی کمکی سیستم[1] (SAP):** SAP یک پردازنده‌ی هسته است که برای عمل I/O ساخته شده است. نقش آن بی‌بار کردن اعمال I/O و مدیریت کانال‌ها و صف‌های اعمال I/O است. این

1. System assist processor

پردازنده، پردازنده‌های همه‌ی وظایف I/O دیگر را آزاد می‌کند، و به آن‌ها اجازه می‌دهد به مدار کاربرد تخصیص یابد.

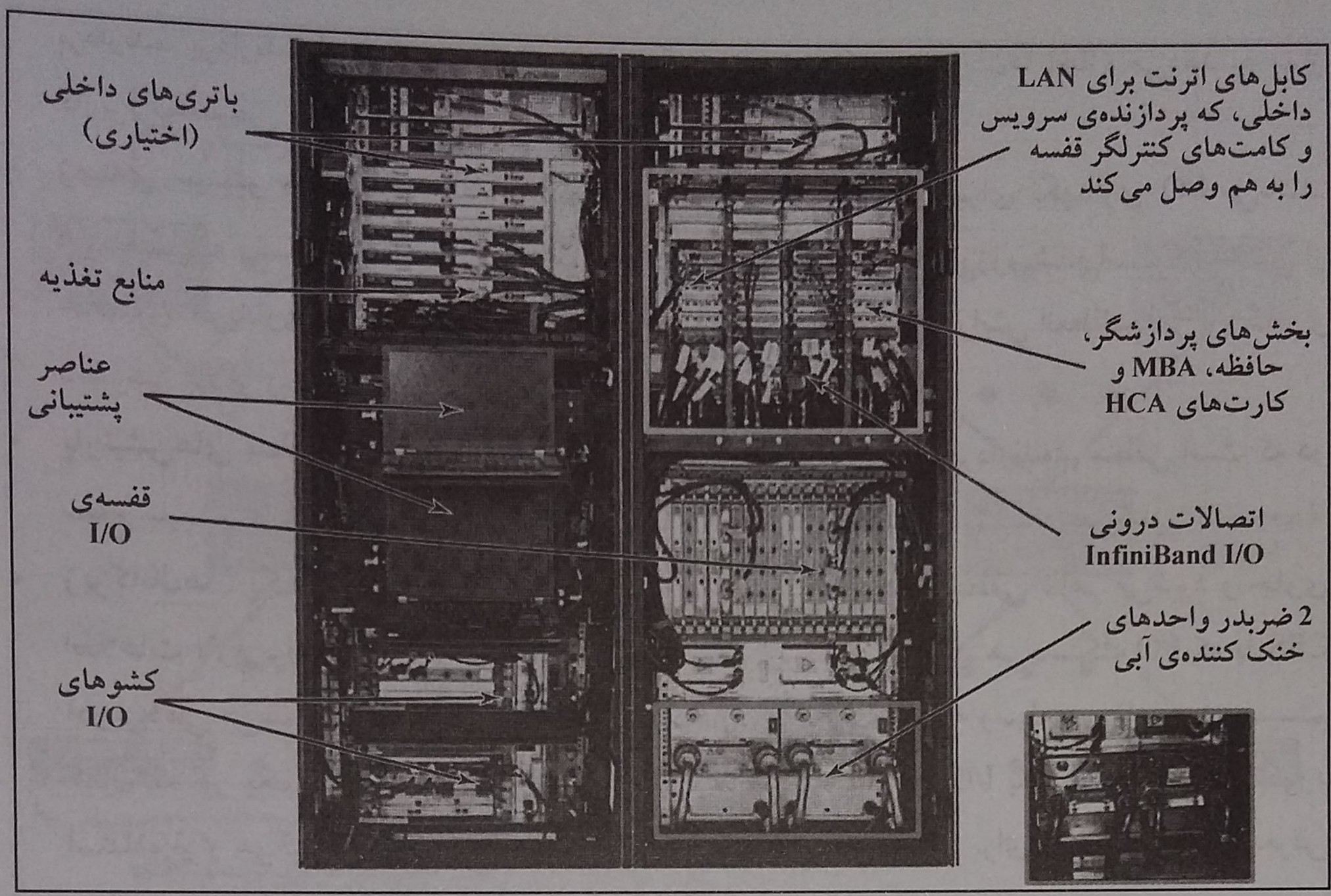
- **زمینه‌ی سیستم سخت‌افزار[1] (HAS):** بخش رزرو حافظه‌ی سیستم برای نگهداری I/O می‌باشد. این بخش به وسیله‌ی SAPها به کار می‌رود. یک مقدار ثابت 16 GB رزرو شده است که بخشی از حافظه‌ی خریداری شده به وسیله‌ی مصرف کننده نیست. این بخش آرایش انعطاف‌پذیر بزرگ‌تر و دسترسی بالاتر را فراهم می‌نماید.
- **پارتیشن‌های منطقی[2]:** بخش پارتیشن منطقی نوعی ماشین مجازی یا پردازنده‌ی منطقی است که در سطح سیستم عامل تعریف می‌شود. هر ESS تا 16 پارتیشن منطقی را پشتیبانی می‌کند.
- **زیرکانال‌ها[3]:** یک زیرکانال برای یک برنامه به صورت یک دستگاه منطقی ظاهر می‌شود و حاوی اطلاعات لازم برای اجرای یک عمل I/O است. یک زیرکانال در ازای هر دستگاه I/O که با CSS آدرس‌دهی می‌شود، برای هر دستگاه I/O وجود دارد. یک زیرکانال به وسیله‌ی برنامه‌ی زیرسیستم کانال که در یک پارتیشن اجرا می‌شود، برای ارسال یک تقاضای I/O به زیرسیستم کانال مورد استفاده قرار می‌گیرد. به هر دستگاه یک زیرکانال تخصیص می‌یابد که برای پارتیشن منطقی معرفی می‌گردد. در ازای هر CSS تا 196 K زیرکانال قابل تعریف است.
- **مسیر کانال[4]:** یک مسیر کانال یک تک واسط بین یک زیرسیستم کانال و یک یا چند واحد کنترل است، که از طریق کانال برقرار می‌شود. فرامین و داده‌ها به مسیر کانال فرستاده می‌شوند تا درخواست‌های I/O را اجرا نمایند. هر CSS می‌تواند تا 256 مسیر کانال داشته باشد.
- **کانال:** کانال‌ها پردازنده‌های کوچکی هستند که با واحدهای کنترل ارتباط برقرار می‌کنند. آن‌ها انتقال داده را بین حافظه و دستگاه‌های خارجی مدیریت می‌نمایند.

این ساختار پیچیده مین‌فریم را قادر می‌سازد تا مقدار زیادی دستگاه‌های I/O و ارتباطات مخابره را مدیریت کند. همه‌ی فرآیندهای I/O از کاربر و پردازنده‌های سرور جدا می‌شوند، و کارایی را بالا می‌برد. پردازنده‌های زیرسیستم کانال تا حدی از نظر پیکربندی کلی‌اند، و این اجازه می‌دهد تا گستره‌ی وسیعی از وظایف I/O را اداره کرده و با نیازها پیش برود. پردازنده‌های کانال به طور خاصی برای واحدهای کنترل I/O که به آن‌ها وصلند، برنامه‌نویسی می‌شوند.

سازمان سیستم I/O

برای توضیح سازمان I/O لازم است که ابتدا به طور خلاصه ساختار فیزیکی zEnterprise 196 را توضیح دهیم. شکل ۲۲-۷ نمای جلوی ماشین را که از نوع آب خنک‌کن است، نشان می‌دهد (نوع هوا خنک‌کن هم وجود دارد)، سیستم دارای مشخصه‌های زیر است:

1. Hardware system area 2. Logical partitions 3. Subchannel 4. Channel path



شکل ۲۲-۷ نمای جلو ـ فریم‌های IBM z196 I/O .

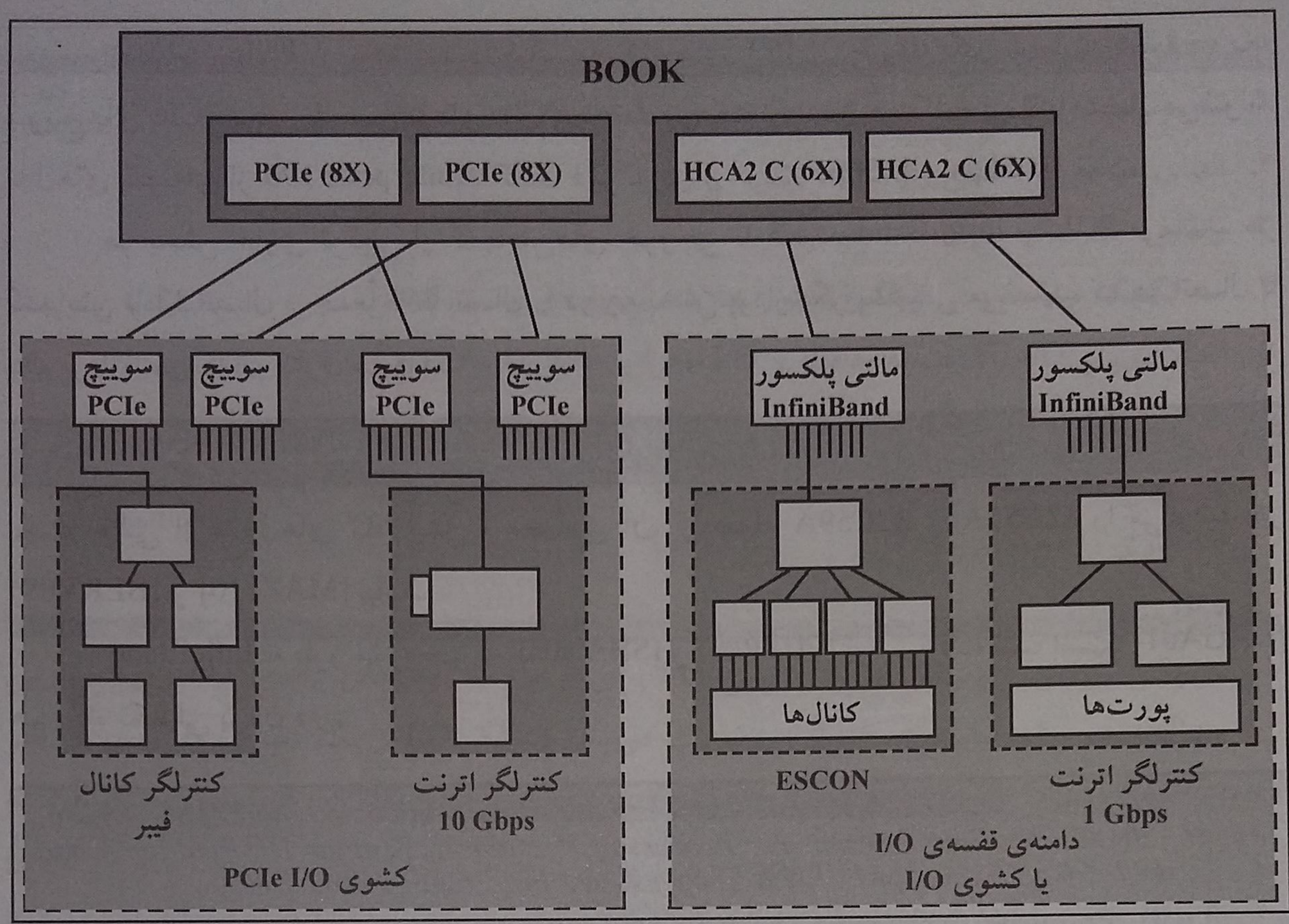
- وزن: 2185 kg (4817 lbs)
- پهنا: 1.534 m (5 ft)
- عمق: 1.375 m (4.5 ft)
- ارتفاع: 2.012 m (6.6 ft)

دقیقاً لپ‌تاپ نیست.

سیستم از دو محفظه‌ی بزرگ به نام فریم تشکیل شده است، که اجزای مختلف zEnterprise 196 را در خود جای می‌دهد. فریم سمت راست A شامل دو قفس بزرگ، به‌علاوه‌ی فضایی برای کابل‌ها و اجزای دیگر است. قفس بالا، قفس پردازنده است و چهار شیار برای جای دادن چهار بخش پردازشگر است که به طور کامل به هم وصلند. هر بخش پردازشگر حاوی یک مدول چند تراشه‌ای (MCM)، کارت‌های حافظه و اتصالات قفس است. هر MCM بوردی است که شش تراشه‌ی چند هسته‌ای و دو تراشه‌ی کنترل ذخیره‌سازی دارد.

قفس پایین در فریم A یک قفس I/O است که حاوی سخت‌افزار I/O از جمله مالتی‌پلکسورها و کانال‌هاست. قفس I/O یک واحد ثابت است که برای سفارشات مشتری در کارخانه نصب شده است.

فریم سمت چپ Z حاوی باتری‌ها و منابع تغذیه و فضایی برای یک یا چند بخش پشتیبانی است، که به وسیله‌ی مدیر سیستم برای مدیریت مجموعه به کار می‌رود. فریم Z حاوی شیارهایی برای دو کشوی I/O اضافی هم هست. یک کشوی I/O مشابه با اجزای یک قفس I/O است. اختلاف این است که کشوها کوچکترند و به راحتی در محل مشتری برای تأمین نیازها بیرون آورده شده و داخل برده می‌شوند.



شکل ۲۳-۷ ساختار سیستم IBM z196 I/O .

با این زمینه‌ی فکری اکنون یک آرایش از سیستم zEnterprise 196 I/O را در شکل ۲۳-۷ ملاحظه نمایید. بخش پردازشگر z 196 دو زیرساخت I/O (یعنی داخلی به فریم A و Z): یعنی InfiniBand برای قفس‌های I/O، کشوهای I/O و PCI Express (PCIe) برای کشوهای I/O . این کنترلگرهای کانال را **گنجایش خروجی**[1] می‌نامند.

اتصالات InfiniBand از بخش پردازشگر به قفس‌های I/O و کشوهای I/O از طریق گنجایش خروجی آداپتور (تطبیق‌دهنده) میزبان (HCA) است، که دارای پیوندهایی به مالتی‌پلکسورها در قفس یا کشو می‌باشد. مالتی‌پلکسورهای InfiniBand برای ارتباط سرورهای داخلی، تجهیزات زیرساخت مخابرات، ذخیره‌سازی و سیستم‌های اختصاصی است. علاوه بر استفاده از InfiniBand برای ارتباط درونی سیستم‌هایی که همگی InfiniBand را به کار می‌برند، مالتی‌پلکسور دیگر تکنولوژی‌های I/O را پشتیبانی می‌نماید. ESCON اتصال به دیسک‌ها، نوارها و چاپگر که از تکنولوژی فیبر نوری استفاده می‌نماید. اتصالات اترنت، اتصالات 1 Gbps و 10 Gbps برای انواع دستگاه‌هایی که این تکنولوژی شبکه‌ی محلی را پشتیبانی می‌کنند فراهم می‌سازد. یک کاربرد قابل ذکر ساخت یک سرور بزرگ برای سرورهای کشویی متصل به هم با دیگر مین‌فریم است.

1. Fanout

اتصالات PCIe از بخش پردازشگر به کشوهای I/O از طریق یک گنجایش خروجی به سوییچ‌های PCIe برقرارند. سوییچ‌های PCIe به تعدادی کنترلگرهای دستگاه‌های I/O متصل می‌شوند. مثال‌های نمونه‌ای از zEnterprise 196، کانال فیبر نوری و اترنت 1 Gbps و 10 Gbps است.

هر بخش حاوی ترکیبی از گنجایش‌های خروجی تا 8 عدد InfiniBand و PCIe می‌باشد. هر گنجایش تا 32 اتصال و جمعاً 256 اتصال را در هر بخش پردازشگر پشتیبانی می‌نماید، که هر اتصال با یک پردازنده‌ی کانال کنترل می‌گردد.

۷-۹ خواندنی‌های پیشنهادی

بحث جالبی از مدول‌های I/O اینتل و معماری آن، از جمله B2C59A و 82C55A را می‌توانید در [BERY09] و [MAZ110] بیابید.

InfiniBand به طور مشروح‌تر در [SHAN03] و [FUTR01] پوشش داده شده است. [KAGA01] هم نگاه تکمیلی بر آن دارد.

BREY09 Brey, B. *The Intel Microprocessors: 8086 / 8066, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4 and Core2 with 64-bit Extensions*. Upper Saddle River, NJ: Prentice Hall, 2009.

FUTR01 Futral, W. *InfiniBand Architecture: Development and Deployment*. Hillsboro, OR: Intel Press, 2001.

KAGA01 Kagan, "InfiniBand: Thinking Outside the Box Design." *Communications System Design*, September 2001. (www.csdmag.com)

MAZ110 Mazidi, M., and Mazidi, J.; and Causey, D. *The x86 PC: Assembly Language, Design and Interfacing*. Upper Saddle River, NJ: Prentice Hall, 2010.

SHAN03 Shanley, T. *InfinBand Network Architecture*. Reading, MA: Addison Wesley, 2003.

۷-۱۰ واژه‌های کلیدی، پرسش‌های مرور درس و مسائل

واژه‌های کلیدی

cycle stealing	سرقت سیکل	isolated I/O	I/O منفک
direct memory access (DMA)	دستیابی مستقیم به حافظه	memory – mapped I/O	I/O نگاشت به حافظه
InfiniBand	اینفینی باند	multiplexor channel	کانال مالتی‌پلکسور
interrupt	وقفه	parallel I/O	I/O موازی
interrupt – driven I/O	وقفه – گرداننده‌ی I/O	peripheral device	دستگاه جانبی
I/O channel	کانال I/O	programmed I/O	I/O برنامه‌نویسی شده
I/O command	فرمان I/O	selector channel	کانال انتخابگر
I/O module	مدول I/O	serial I/O	I/O سریال
I/O processor	پردازنده‌ی I/O	Thunderbolt	تاندربالت

پرسش‌های مرور درس

۱. سه دسته‌بندی گسترده از دستگاه‌های خارجی یا جانبی را لیست کنید.
۲. الفبای مرجع بین‌المللی چیست؟
۳. وظایف مهم یک مدول I/O چیست؟
۴. سه تکنیک برنامه‌نویسی I/O را به‌طور خلاصه تعریف و لیست کنید.
۵. اختلاف بین I/O نگاشت به حافظه و I/O مجزا یا منفک را بیان کنید؟
۶. وقتی‌که یک وقفه دستگاه رخ می‌دهد، پردازنده چگونه دستگاه صادرکننده را می‌شناسد.
۷. چه موقع یک مدول DMA کنترل یک گذرگاه را به عهده می‌گیرد و هنگام حفظ کنترل پردازنده چه کاری را انجام می‌دهد؟

مسائل

۱. در یک نمونه‌ی ریزپردازنده یک آدرس I/O برای ارجاع به ثبات‌های داده‌ی I/O و یک آدرس جداگانه هم برای ثبات‌های کنترل و وضعیت در یک کنترلگر I/O از یک دستگاه مفروض در نظر گرفته شده است. این ثبات‌ها را **پورت** یا **درگاه** می‌نامند. در اینتل 8088 ، دو فرمت دستور I/O به کار رفته است. در یکی از آن‌ها، یک کد عمل I/O موجب می‌شود یک عمل I/O صورت گیرد؛ به دنبال آن یک آدرس پورت 8 بیت قرار دارد. انواع دیگری از کدهای I/O آدرس I/O با ثبات 16 بیتی DX دارند. در هر مد آدرس‌دهی I/O در 8088 ، چند پورت وجود دارد.

۲. فرمت دستور مشابهی در ریزپردازنده‌ی Z8000 به کار می‌رود. در این حالت، یک قابلیت آدرس‌دهی مستقیم موجود است که در آن یک آدرس پورت 16 بیتی بخشی از دستور می‌باشد، و نیز یک قابلیت آدرس‌دهی پورت غیرمستقیم که در آن دستور به یکی از ثبات‌های 16 بیتی ارجاع دارد که حاوی آدرس پورت است. Z8000 در هر مد آدرس‌دهی I/O چند پورت را می‌تواند آدرس‌دهی کند؟

۳. Z8000 یک قابلیت انتقال بلوکی I/O نیز هست، که برخلاف DMA ، تحت کنترل مستقیم پردازنده می‌باشد. دستورات انتقال بلوکی یک ثبات آدرس پورت (Rp)، یک ثبات شمارش (Rc) و یک ثبات مقصد (Rd) را مشخص می‌کنند. Rd حاوی آدرس حافظه‌ی اصلی است که در آن اولین بایت خوانده شده از پورت ورودی در آن ذخیره می‌گردد. Rc یکی از این ثبات‌های همه‌منظوره‌ی 16 بیتی است. بلوک داده‌ی انتقالی چقدر بزرگ است؟

۴. پردازنده‌ای را در نظر بگیرید که دارای دستور انتقال I/O مشابه با Z8000 است. پس از اولین اجرا، این دستور به پنج سیکل ساعت برای اجرای مجدد نیاز دارد. بنابراین، اگر یک دستور I/O غیربلوکی به کار رود، جمعاً 20 سیکل ساعت برای برداشت و اجرا مورد نیاز است. برای دستور بلوکی I/O جهت انتقال یک بلوک 128 بایتی افزایش سرعت را محاسبه کنید.

۵. یک سیستم مبتنی بر ریزپردازنده‌ای 8 بیتی بوده و دارای دو دستگاه I/O است. کنترلگرهای I/O در این سیستم از ثبات‌های کنترل و وضعیت جدا از هم استفاده می‌کنند. هر دو دستگاه براساس مدیریت هر بار یک بایت کار می‌نمایند. دستگاه اول دو خط وضعیت و سه خط کنترل و دستگاه دوم سه خط وضعیت و چهار خط کنترل دارد.

الف. چند ثبات مدل کنترل I/O ، 8 بیت برای خواندن وضعیت و کنترل هر دستگاه لازم است؟

ب. در کل چند ثبات مدول کنترل که اولین دستگاه فقط خروجی باشد لازم است؟

پ. چند آدرس جدا برای کنترل دو دستگاه لازم است؟

۶. برای I/O برنامه‌نویسی‌شده، شکل ۵-۷ نشان می‌دهد که پردازنده در یک حلقه‌ی انتظار برای تست وضعیت یک دستگاه I/O باقی می‌ماند. برای افزایش بازدهی، نرم‌افزار I/O طوری نوشته می‌شود که پردازنده وضعیت دستگاه را به طور پریودیک بخواند. اگر دستگاه آماده نبود پردازنده بتواند به وظیفه‌ی دیگری پرش کند و پس از مدتی دوباره برای چک‌کردن وضعیت باز گردد.

الف. فرض کنید که طرح فوق برای خروج داده‌ای است که هر بار کاراکتری را به چاپگری می‌فرستد که این یکی در هر ثانیه 10 کاراکتر را (CPS) چاپ می‌نماید. اگر وضعیتش را هر 200 ms اسکن کند چه رخ خواهد داد؟

ب. سپس صفحه‌کلیدی را با یک بافر کاراکتر در نظر بگیرید. کاراکترها به طور متوسط سرعت ورود 10 CPS دارند. با این وجود فاصله‌ی زمانی بین دو فشردگی کلید می‌تواند تا حداقل 60 ms باشد. در چه فرکانسی صفحه‌کلید به وسیله‌ی برنامه‌ی I/O باید اسکن شود؟

۷. ریزپردازنده‌ای وضعیت خروجی یک دستگاه I/O را در هر 20 ms یک بار اسکن می‌کند. این کار به کمک تایمری انجام می‌شود که هر 20 ms یک بار پردازنده را خبر کند. واسط دستگاه شامل دو پورت است: یکی برای وضعیت و دیگری برای خروجی داده. با فرض سرعت ساعت 8 MHz ، چقدر طول می‌کشد تا دستگاه اسکن شده و سرویس شود؟ تمام سیکل‌های ساعت دستورات مورد بحث 12 سیکل ساعت است.

۸. در بخش ۳-۷ یک مزیت و یک عیب I/O نگاشت به حافظه را در مقایسه با I/O منفک لیست کردیم. دو مزیت و دو عیب دیگر را لیست کنید.

۹. یک سیستم خاص توسط یک اپراتور از طریق فرامین وارد شده از صفحه‌کلید کنترل می‌شود. تعداد متوسط فرامین وارده در فاصله‌ی زمانی 8 ساعت، 60 است.

الف. فرض کنید که پردازنده صفحه‌کلید را هر 100 ms یک بار اسکن کند. در طول 8 ساعت صفحه‌کلید چند بار چک خواهد شد؟

ب. اگر از I/O وقفه‌پذیر استفاده می‌شد تعداد ویزیت‌های پردازنده چقدر بود؟

۱۰. سیستمی را در نظر بگیرید که از I/O وقفه‌پذیر برای انتقال داده از دستگاه خاصی استفاده می‌کند. براساس پیوسته بودن عمل، متوسط 8 KB/s فرض می‌شود.

الف. فرض کنید که پردازش وقفه حدود 100 ms زمان بخواهد (یعنی زمان پرش به روال سرویس وقفه (ISR)، اجرای آن و بازگشت به برنامه‌ی اصلی). مشخص کنید که چه کسری از وقت پردازنده برای این I/O صرف می‌شود، با این فرض که وقفه در ازای هر بایت رخ دهد.

ب. اکنون فرض کنید که دستگاه دو بافر 16 بایت دارد و هرگاه یکی از بافرها پر شود، وقفه رخ می‌دهد. طبیعتاً، پردازش وقفه طولانی‌تر خواهد بود زیرا ISR باید 16 بایت را انتقال دهد. ضمن اجرای ISR ، پردازنده برای انتقال هر بایت 8 ms وقت صرف می‌کند. معین کنید که چه کسری از وقت پردازنده برای این کار توسط دستگاه I/O صرف می‌گردد.

پ. اکنون فرض کنید که پردازنده با یک دستور I/O بلوکی انتقال مانند Z8000 مجهز باشد. این قابلیت اجازه می‌دهد تا ISR مربوطه هر بایت از بلوک را فقط در 2 ms طی کند. بگویید در این حالت چه کسری از وقت پردازنده توسط I/O صرف خواهد شد.

۱۱. تقریباً در همه‌ی سیستم‌هایی که مدول‌های DMA دارند، اولویت DMA نسبت به CPU در دستیابی به حافظه بالاتر است. چرا؟

۱۲. یک مدول DMA ارسال کاراکترها به حافظه را با سرقت سیکل و از دستگاه با سرعت 9600 bps انجام می‌دهد. پردازنده دستورات را با سرعت 1 میلیون بر ثانیه برداشت می‌کند (1 MIPS). به دلیل فعالیت DMA پردازنده چقدر آهسته می‌شود.

۱۳. سیستمی را در نظر بگیرید که در آن زمان انتقال داده از طریق گذرگاه 500 ns باشد. انتقال کنترل گذرگاه به هر یک از جهات، یعنی از پردازنده به دستگاه I/O و بالعکس، 250 ns است. یکی از وسایل دارای سرعت انتقال داده‌ی 50 KB/s بوده و از DMA استفاده می‌کند. داده هر بار یک بار انتقال می‌یابد.

الف. اگر ما از یک DMA برای مد یکجا[1] استفاده کنیم، واسط DMA قبل از شروع انتقال بلوک گذرگاه را به اختیار گرفته و کنترل گذرگاه را تا انتقال کل بلوک حفظ می‌نماید. اگر بلوک 128 بایت باشد، وسیله تا چه زمانی به گذرگاه متصل است.

ب. برای مد سرقت سیکل (چرخه) عمل را تکرار نمایید.

۱۴. بررسی نمودار زمان‌بندی 8237 A نشان می‌دهد که از لحظه شروع انتقال بلوک هر سیکل DMA ، سه سیکل ساعت زمان طول می‌کشد. 8237 A در سیکل DMA یک بایت از اطلاعات را بین حافظه و وسیله‌ی I/O انتقال می‌دهد.

الف. فرض کنید 8237 A با سرعت 5 MHz ساعت باشد، انتقال یک بایت چقدر طول می‌کشد؟

---

1. Burst

ب. حداکثر سرعت انتقال داده‌ی قابل دسترسی چقدر است؟

پ. فرض کنید که حافظه به قدر کافی سریع نیست و باید دو حالت انتظار را در ازای هر سیکل DMA وارد کنیم. سرعت واقعی انتقال چقدر است؟

۱۵. اگر در مسئله‌ی سیستم قبل، یک سیکل حافظه 750 ns طول بکشد، سرعت ساعت گذرگاه را تا چه حد بدون تأثیر بر سرعت انتقال داده می‌توان تقلیل داد؟

۱۶. یک کنترلگر DMA چهار خط مخابره از نوع فقط دریافتی را سرویس می‌دهد (یکی در ازای هر کانال DMA) و سرعت 64 Kbps برای هر یک از آن‌ها دارد.

الف. آیا کنترلگر را در مد یکجا و سرعت چرخه به کار می‌برید؟

ب. برای سرویس کانال‌های DMA چه طرح اولویتی پیشنهاد می‌کنید؟

۱۷. یک کامپیوتر 32 بیت دارای دو کانال سلکتور (انتخابگر) و یک کانال مالتی‌پلکسور است. هر کانال انتخابگر دو دیسک مغناطیسی و دو نوار مغناطیسی را پشتیبانی می‌نماید. کانال مالتی‌پلکسور دو خط چاپگر، دو کارت‌خوان و پایانه VDT دارد که به آن متصل است. سرعت‌های انتقال زیر را ملاحظه نمایید:

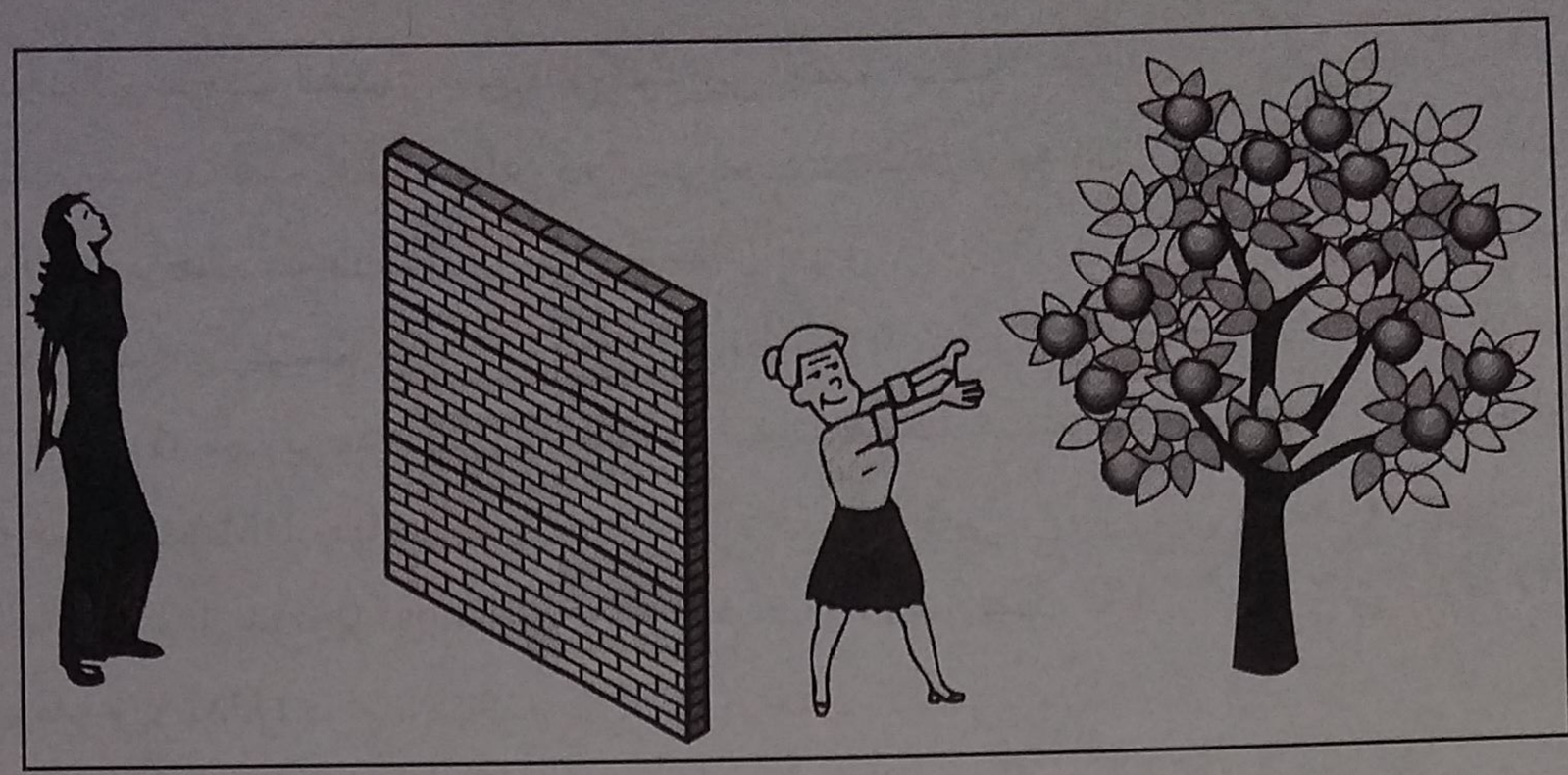
راه‌انداز دیسک	800 Kbytes/s
راه‌انداز نوار مغناطیسی	200 Kbytes/s
چاپگر خطی	6.6 Kbytes/s
کارت‌خوان	1.2 Kbytes/s
VDT	1 Kbytes/s

حداکثر سرعت انتقال I/O در این سیستم را تخمین بزنید.

۱۸. یک کامپیوتر متشکل از یک پردازنده و یک دستگاه I/O ، D از طریق گذرگاه مشترک با عرض گذرگاه یک کلمه به حافظه‌ی اصلی M وصل است. پردازنده می‌تواند تا 10^6 دستور را در ثانیه انجام دهد. به طور متوسط یک دستور پنج سیکل ماشین لازم دارد که سه سیکل آن از گذرگاه حافظه استفاده می‌نماید. یک عمل خواندن یا نوشتن یک سیکل ماشین را به کار می‌برد. فرض کنید که پردازنده به طور پیوسته برنامه‌های زمینه‌ای را که 95% دستورات را تشکیل می‌دهد و فاقد دستورات I/O است اجرا نماید و نیز فرض کنید یک سیکل پردازنده برابر یک سیکل گذرگاه باشد. اکنون تصور کنید که دستگاه I/O برای انتقال بلوک‌های بزرگی از داده بین M و D به کار رود.

الف. اگر I/O برنامه‌نویسی شده به کار رود و هر انتقال I/O یک کلمه‌ای از پردازنده اجرای دو دستور را تقاضا نماید حداکثر سرعت انتقال داده‌ی I/O برحسب کلمه بر ثانیه ممکن را از D تخمین بزنید.

ب. اگر DMA را به کار ببریم، تخمین را مجدداً مشخص کنید.



شکل ۲۴-۷ مسئله‌ی سیب.

۱۹. یک منبع داده کاراکترهای IRA هفت بیتی را تولید می‌نماید که به هر یک از آن‌ها یک بیت توازن اضافه شده است. عبارتی برای میزان داده‌ی مؤثر حداکثر (میزان بیت داده IRA) در یک خط Rbps برای موارد زیر مشخص کنید.

الف. انتقال غیرهمگام، بایت توقف 1.5 واحدی.

ب. انتقال همگام بیتی با قابی متشکل از 48 بیت کنترل و اطلاعات 128 بیتی.

پ. مثل (ب) با میدان اطلاعات 1024 بیتی.

ت. همگام کاراکتری، با کاراکترهای کنترل 9 بیت در هر قاب و 16 کاراکتر اطلاعاتی.

ث. مثل (ت) با میدان اطلاعات 128 .

۲۰. مسئله‌ی زیر بر شرح راهکار I/O در [ECKE90] مبتنی است (شکل ۲۴-۷). دو دختربچه در هر یک از دو سمت یک دیوار بلند بازی می‌کنند. یکی از بچه‌ها، به‌نام سرویس‌دهنده‌ی سیب، یک درخت زیبای سیب با سیب‌های خوش‌طعم دارد که در سمت خودش رشد کرده است. او از این‌که سیب را به دختر دیگر به هنگام لزوم می‌دهد خوشحال است. دختر دیگر به نام خورنده‌ی سیب دوست دارد سیب بخورد ولی چیزی ندارد. در واقع نامبرده باید با سرعت ثابت بیشترین سیب را بخورد (یک سیب در روز او را از دکتر بی‌نیاز می‌نماید). اگر او سیب را سریع‌تر از حد بخورد مریض می‌شود. اگر کندتر بخورد دچار عارضه می‌گردد. هیچ یک از دو بچه نباید صحبت کنند تا سیب از سرویس‌دهنده‌ی سیب به خورنده‌ی سیب با سرعت صحیح برود.

الف. فرض کنید یک ساعت اخطار دهنده روی دیوار باشد و قابلیت تنظیم چندگانه باشد. از ساعت چگونه می‌توان برای حل مسئله استفاده کرد؟ نمودار زمان‌بندی را برای تشریح مسئله رسم کنید.

ب. اکنون فرض کنید که هیچ ساعت اخطاری وجود ندارد. در عوض خورنده‌ی سیب دارای پرچمی است که به هنگام نیاز، آن را تکان می‌دهد. راه حل جدیدی را پیشنهاد کنید. آیا داشتن

پرچم برای سرویس‌دهنده‌ی سیب کمکی به موضوع می‌کند؟ اگر مثبت است، آن را در این حل دخالت دهید. عواقب و معایب این روش را مورد بحث قرار دهید.

پ. اکنون پرچم را کنار بگذارید و یک رشته‌ی طویل را در نظر بگیرید. راه‌حلی پیشنهاد کنید که با رشته از (ب) بهتر باشد.

۲۱. فرض کنید که دو ریزپردازنده‌ی 8 بیت و 16 بیت به یک سیستم گذرگاه وصل شوند. جزئیات زیر مفروض است:

۱. هر دو پردازنده ویژگی انتقال هر نوع داده را دارند: یعنی I/O برنامه‌نویسی شده، I/O تحت وقفه و DMA .

۲. همه‌ی ریزپردازنده‌ها گذرگاه آدرس 16 بیتی دارند.

۳. دو بورد حافظه، هر یک با ظرفیت 64 K بایت به گذرگاه وصل است. طراح می‌خواهد یک حافظه‌ی مشترک را با حداکثر ظرفیت به کار ببرد.

۴. گذرگاه سیستم حداکثر چهار وقفه و یک خط DMA را پشتیبانی می‌کند. هر گونه فرض لازم دیگر را می‌توانید در نظر بگیرید.

الف. ویژگی‌های گذرگاه سیستم را برحسب تعداد و نوع خطا بیان کنید.

ب. یک پروتکل ممکن را برای تبادل داده روی گذرگاه توصیف کنید، یعنی خواندن/نوشتن وقفه و DMA .

پ. بگویید چگونه دستگاه‌های قبل به گذرگاه سیستم وصل می‌شوند.