

۵

وجود یک سیستم (system bus) مشترک بین تمامی مولفه‌ها که در سیستم کامپیوتری یا در مورد گذرگاه

استفاده قرار می‌دهند، چالش‌های مهمی چون موارد زیر را در بر می‌گیرد:

* هماهنگی دستگاه‌های مختلف برای در اختیار گرفتن گذرگاه برای ترافیک

تلفات ناشی شده و مسبب ایجاد تاخیر و کاهش بهره‌وری سیستم

* انتظار برای در اختیار گرفتن گذرگاه،

* تفاوت نرخ نیاز به سرسی گذرگاه برای دستگاه‌های مختلف و همچنین

تفاوت سرعت کلاک (clock speed) دستگاه‌های مختلف.

۱۳

در ادامه با یک بررسی جنبه بلندی:

* وقفه‌ها (interrupts)،

* ساختار ذخیره‌سازی (storage structure)،

* ساختار I/O (I/O structure)،

به شرح نحوه عملکرد یک سیستم کامپیوتری پردازش

۱۴

وقفه‌ها - یک مثال

روی یک سیستم کامپیوتری، برنامه‌ار را در نظر بگیرید که در حال اجراست و نیازمند به یک عمل I/O

برای انجامی کند. برای این کار فرآیند زیر دنبال می‌شود:

۱! برای شروع عمل I/O، Device Driver (DD) راجع به‌های مناسب در

Device Controller (DC) را بر حسب نیاز مندی پیش‌اکمده، مقداردهی می‌کند.

۲ در مرحله بعد، DC اقدام به بررسی دستورهای رجیسترها برای آگاه شدن از محل مورد تقاضای نماید.
این عمل به عنوان مثال می تواند خواندن یک کاراکتر از صفحه کلید باشد.

۳ در ادامه، DC شروع به انتقال داده از دستگاه به بافر محلی (Local Buffer) خود می کند.

۴ هنگامی که انتقال داده پایان یافت، DC این انجام را به DD اطلاع می دهد.

۵ در پایان، DD ممکن است مثلاً از درخواست I/O به سبب خواندن از صفحه کلید بود،
داده پردازش شده را برگرداند. لازم به ذکر است که DD ممکن است با وضعیت های
مختلف روبرو شود: * اشغال دستگاه I/O مورد نظر، * نوسان موفقیت آمیز
موارد مشابه نیز عواضه شود.

نکته: مرحله ۴ فراگرفته شود، یعنی آگاه کردن DD توسط DC، از طریق وقفه
(Interrupt) انجام می شود.

۱۵) وقفه ها - نمای کلی

* وقفه ها بخش کلیدی تقابل سیستم عامل و سخت افزار را تشکیل می دهند.

* دستگاه ها (Devices) در هر زمان با ارسال سیگنال به CPU از طریق دستگاه
سیستم می توانند وقفه را راه اندازی کنند.

* نمای کلی از شروع تا اتمام یک وقفه:

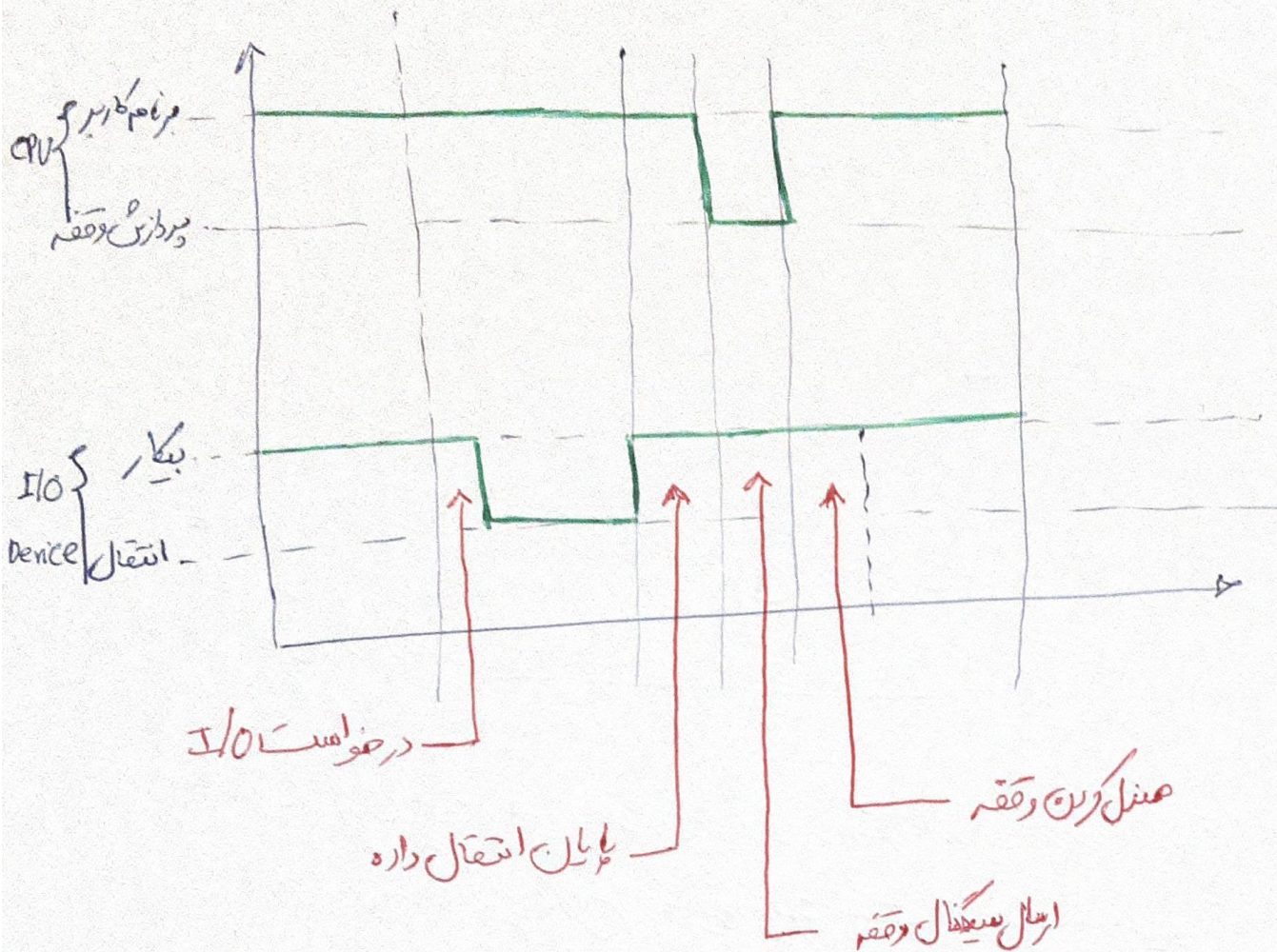
۱. ~~دستگاه مورد نیاز از CPU درخواست می کند~~ ارسال وقفه از Device به CPU.

۲. توقف اجرای برنامه (بگذار از دستور العمل جاری CPU) و اجرای روتین مربوط به سرویس
وقفه.

۳. اجرای روتین وقفه درخواستی،

۴. پس از اتمام روتین وقفه می بایست محاسبات قطع شده از سر گرفته شود.

شکل زیر چند زمانی و وقفه برابر یک برنامه در حال اجرا را نشان می دهد. (شکل ۱.۳ منصفه ۸ کتاب)



۱۶) چالش هایی که در ارسال سیگنال وقفه و منتظر کردن آن وجود دارد:

* هر مداری که کامپیوتری (سولف مارک) مکانیزم وقفه محفوظ به خود دارد و بالطبع هر وقفه ایجاد شده، هدف خاصی را دنبال می کند. پس هر وقفه صرفاً یک سیگنال ساده و فاقد اطلاعات نیست.

* فرکانس تولید وقفه ها در یک سیستم کامپیوتری بالاست و به همین دلیل برای عملکرد صحیح یک سیستم کامپیوتری نیاز است که آن ها خیلی سریع منتظر شوند.

* پیدا کردن یک روتین (routine) برای منتظر کردن هر وقفه Interrupt handler routine

* پیدا کردن یک ساختار برای نگهداری روتین مربوط به هر وقفه Interrupt vector ← دلیلی است

* باید مکانیزمی داشت باشیم که اگر برنامه در حال اجرای وقفه راه شد، پس از اتمام وقفه بتوان ادامه کار آن برنامه را از سر گرفت.

(۱۷) مکانیزم باید وقفه که مشترک بین تمامی سیستم های کامپیوتری است، به صورت زیر عمل می کند:

۱! سخت افزار CPU یک ^{wire}سیم به نام خط درخواست وقفه (Interrupt-request line) دارد. بعد از اتمام هر دستور العمل آن را می کشد.

۲! اگر CPU تشخیص دهد که وقفه صورت گرفته است، شماره وقفه را خوانده و رویت عمل کننده وقفه (Interrupt-handler routine) را اجرای کند.

۳! روال عمل کننده وقفه یا توصیف شماره وقفه، رویت مربوط به آن را اجرای کند. لازم به ذکر است که این روال وظایفی نظیر: ثبت وضعیت ها، دلیل وقفه، بررسی پردازش های فنرویی، بازبینی وضعیت، بازگشت از وقفه به اجرای برنامه متوقف شده، بسیاری دیگر موارد مشابه را بر عهده دارد.

(۱۸) شکل زیر نحوه I/O مبتنی بر وقفه را بر پایه مطالب ذکر شده قبلی به تصویر می کشد. ^{شکل ۱۰.۴ صفحه ۱۰}

