

معماری کامپیوتر

جلسه بیست و هشتم: واحد ورودی-خروجی ۲

ورودی و خروجی



- اتصال پردازنده، حافظه و تجهیزات I/O از طریق باس
- سه شیوه اتصال متفاوت و انتخاب حالت باس داده و آدرس مشترک و کنترل مجزا
- انتقال انبوه بین I/O و حافظه و روش DMA
- شیوه‌های شروع انتقال اطلاعات روی باس
- نحوه انتقال اطلاعات بین پردازنده و تجهیزات I/O
 - روش مبتنی بر سرکشی
 - روش مبتنی بر وقفه



شیوه‌های انتقال اطلاعات

- انتقال اطلاعات بین دو بخش در باس با چه پروتکلی انجام گیرد:

۱- انتقال همگام (سنکرون)

- در این روش به‌ازای هر کلاک اطلاعات انتقال می‌یابند
- این روش سیم‌کشی زیاد برای کلاک نیاز دارد
- در صورتی مناسب است که سرعت طرفین ارتباط مشابه باشد (تنظیم فرکانس کلاک)
- قبلاً گفتیم که تجهیزات I/O کند هستند و منجر به کاهش کارایی ارتباط و پردازنده می‌شوند

۲- انتقال ناهمگام (آسنکرون)

- روش رایج‌تر در انتقال اطلاعات

انتقال ناهمگام اطلاعات



- نیازی به کلاک یکپارچه نیست و به محض فراهم شدن داده، انتقال انجام می گیرد
- **انتقال داده سریال:** انتقال بیت به بیت داده به صورت سری و متوالی (تمرکز روی این روش)
- **انتقال داده موازی:** به تعداد بیت ها سیم کشی داریم تا همزمان انتقال انجام گیرد
- هماهنگی ارسال در این روش چگونه انجام می گیرد؟



- روش Strobe
- روش دست دهی (Hand Shaking)



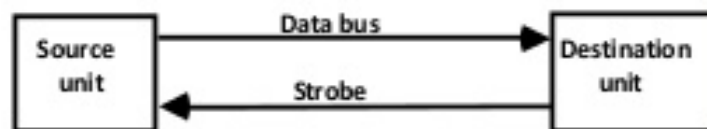
انتقال ناهمگام اطلاعات – هماهنگی ارسال

- نیاز به اضافه کردن سیگنال‌هایی است که مشخص کند چه زمانی آمادگی ارسال وجود دارد
- روش سیگنال Strobe: معتبر بودن داده برای انتقال را با سیگنال تک‌بیتی مشخص می‌کند
 - به ابتکار فرستنده:
 - زمانی که داده تکمیل شد، فرستنده سیگنال strobe را فعال می‌کند تا گیرنده داده را بردارد
 - کنترل strobe در دست فرستنده است و برای زمان مشخصی فعال می‌ماند تا گیرنده از اعتبار داده مطلع شود
 - به ابتکار گیرنده:
 - با نیاز گیرنده به داده سیگنال strobe فعال می‌شود و پس از استفاده از داده این سیگنال غیرفعال می‌گردد

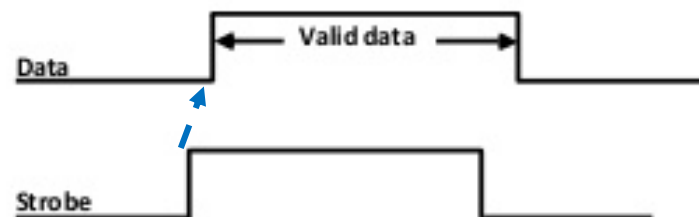
هماهنگی ارسال - سیگنال Strobe



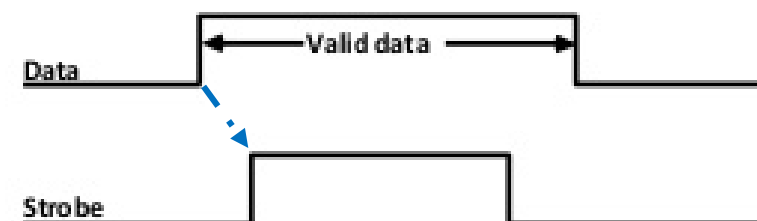
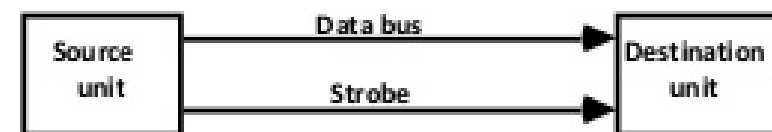
به ابتکار گیرنده



Timing Diagram



به ابتکار فرستنده



هماهنگی ارسال - سیگنال Strobe



- ارتباط از نوع سری است و در هر بار فعال شدن strobe یک بیت منتقل می‌شود
- مزیت روش:
- یک سیم برای هماهنگی ارتباط لازم است و سربار سخت‌افزاری کمی دارد و سریع است
- عیب روش:
- روش یک‌طرفه است، فرستنده نمی‌داند که گیرنده متوجه معتبر بودن داده شده و آن را برداشته یا خیر
- گیرنده نمی‌داند که فرستنده درخواست را گرفته و داده روی باس معتبر است یا خیر

انتقال ناهمگام اطلاعات – هماهنگی ارسال

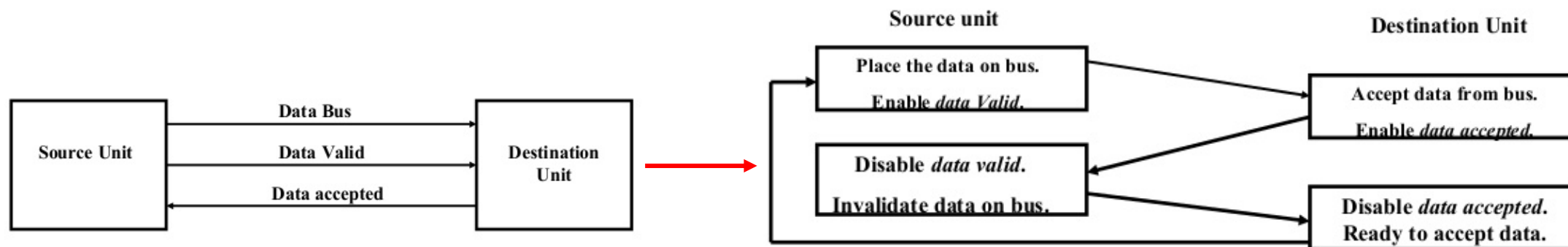


- روش دست‌دهی (Hand Shaking)
- در این حالت دو سیگنال کنترلی تعریف می‌شود (valid , accept/request)
- ارتباط رفت و برگشتی فرستنده و گیرنده ایجاد شود
- به ابتکار فرستنده
- سیگنال valid را فرستنده و accept را گیرنده فعال می‌کند
- به ابتکار گیرنده
- سیگنال request را گیرنده و valid را فرستنده فعال می‌کند

هماهنگی ارسال - مکانیزم دست‌دهی



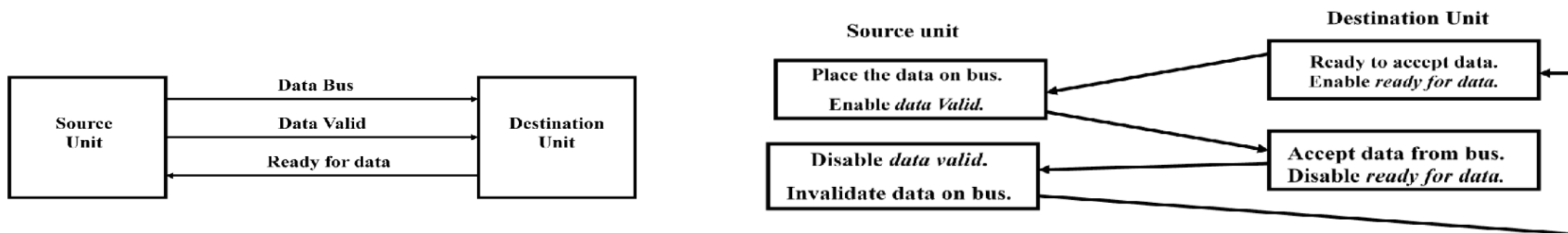
- دست‌دهی به ابتکار فرستنده
- با آماده شده داده، سیگنال valid فعال می‌شود
- زمانی که گیرنده داده را دریافت کرد، accept فعال می‌شود
- با غیرفعال شدن valid، سیگنال accept هم غیرفعال می‌شود.



هماهنگی ارسال - مکانیزم دست‌دهی



- دست‌دهی به ابتکار گیرنده
- با آمدن درخواست از گیرنده (فعال شدن سیگنال request) فرستنده داده را روی باس می‌گذارد
- پس از گذاشتن داده روی باس، فرستنده سیگنال valid را فعال می‌کند
- داده زمانی از روی باس برداشته می‌شود که گیرنده request را غیرفعال کند



برقراری ارتباط بین پردازنده و I/O



- می‌دانیم که سرعت تجهیزات I/O نسبت به پردازنده خیلی کم است
- برای برقراری ارتباط بهتر است پردازنده منتظر نباشد و I/O در زمان نیاز درخواست دهد
- زمان نیاز: شروع یک عملیات یا اطلاع دادن برای پایان یک عملیات
- دو روش وجود دارد:
 - روش مبتنی بر سرکشی (Polling)
 - مرتباً I/O چک می‌شود که آیا خروجی آماده است و درخواستی دارد یا نه که تاخیر زیاد و سخت‌افزار ساده دارد
 - روش مبتنی بر وقفه (Interrupt)
 - زمانی که نیاز بود، پایه interrupt پردازنده توسط تجهیز I/O فعال می‌شود

برقراری ارتباط بین پردازنده و I/O - سرکشی



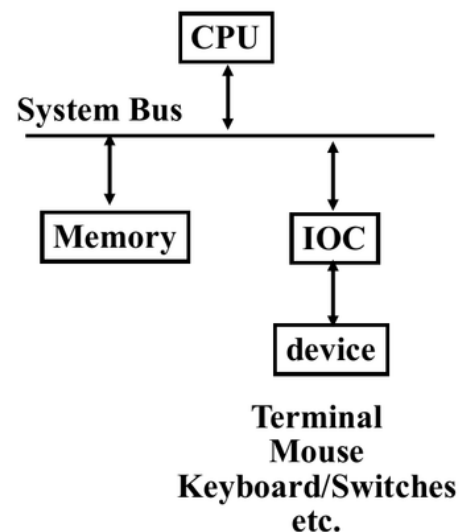
- روش مبتنی بر سرکشی (Polling):

- یک flag داریم که هرگاه توسط دستگاه ورودی/خروجی فعال شد یعنی نیاز به پردازنده است

- لازم است پردازنده به طور متناوب این flag را چک کند

- سربار کارایی زیادی دارد

- سخت افزار ساده دارد





برقراری ارتباط بین پردازنده و I/O- وقفه

- روش مبتنی بر وقفه (Interrupt):
 - برای درخواست یا اعلام اتمام عملیات پایه interrupt پردازنده فعال می شود
 - دو نوع وقفه داریم:
 - خارجی (External): سیم کشی مستقل سخت افزاری دارند (مدنظر ماست و اولویت بالاتر دارند)
 - داخلی (Internal): دو نوع است:
 - از داخل پردازنده و پس از عملیات نادرست مانند تقسیم بر صفر فراخوانی می شود (Exception)
 - توسط کاربر در زبان اسمبلی و با دستور INT انجام می شود و برای برقراری ارتباط هم استفاده می شود
 - مانند subroutine هاست ولی اولویت بالاتری دارد (مانند نوشتن زمان روی نمایشگر)



برقراری ارتباط بین پردازنده و I/O- وقفه

• روش مبتنی بر وقفه (Interrupt):

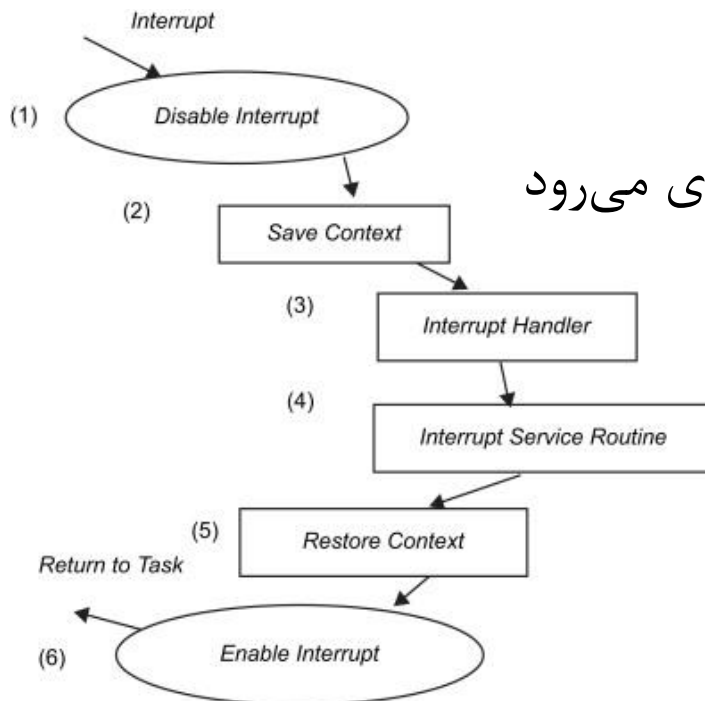
• برای درخواست یا اعلام اتمام عملیات پایه interrupt پردازنده فعال می شود

• پردازنده پس از اتمام دستور جاری، به وقفه رسیدگی می کند

• در اثر وقفه، عملیات جاری پردازنده متوقف شده و کنترل برنامه به مکان دیگری می رود

• Interrupt handler/Interrupt subroutine (ISR)

• عملیات مدنظر وقفه اجرا شده و کنترل برنامه به حالت قبل برمی گردد



برقراری ارتباط بین پردازنده و I/O- وقفه



- در حالت سخت‌افزاری، چند پایه وقفه روی پردازنده داشته باشیم؟
- به‌ازای هر تجهیز ورودی/خروجی یک پایه داشته باشیم
- هزینه و سیم‌کشی زیاد و مدار پیچیده‌تر
- یک پایه وقفه داشته باشیم و دسترسی به آن را مدیریت کنیم
- روش Daisy Chain
- روش Parallel Interrupt

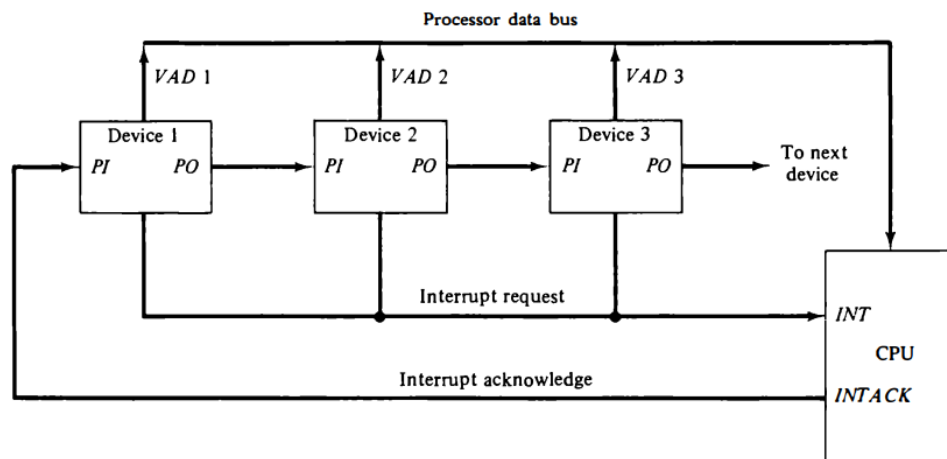


برقراری ارتباط بین پردازنده و I/O- وقفه

• روش مدیریت وقفه Daisy Chain

- زمانی که پایه وقفه پردازنده فعال شد، سیگنال Ack به اولین I/O با اولویت بالاتر می‌فرستد
- اگر این دستگاه درخواستی داشته باشد ارسال می‌کند در غیر این صورت سیگنال Ack را به بعدی می‌دهد
- در این حالت ممکن است برخی I/O ها نوبتشان رعایت نشود

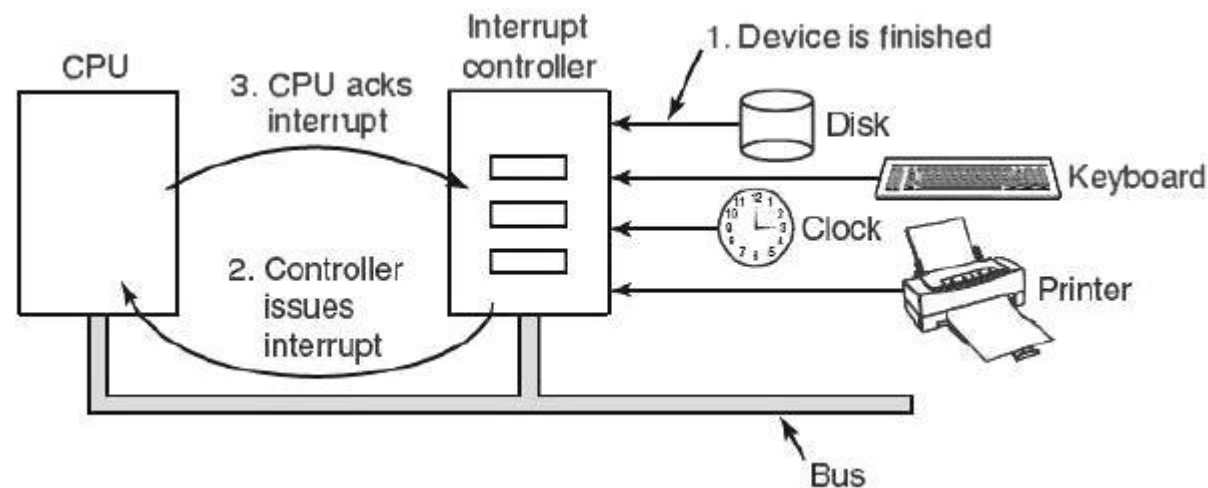
- دستگاه با اولویت بیشتر نوبت دیگری را بگیرد



برقراری ارتباط بین پردازنده و I/O- وقفه



- روش مدیریت وقفه موازی
- واحد PIC (Programmable Interrupt Control) مدیریت وقفه‌ها از I/O به پردازنده را برعهده دارد
- در صورت فعال شدن Ack این واحد براساس اولویت انتخاب می‌کند کدام وقفه به پردازنده برود





مدیریت وقفه پس از قبولی توسط پردازنده

- بعد از آنکه وقفه آمد و پذیرفته شد، نوع آن از کجا مشخص می‌شود؟
 - Vector:
- در کلاک بعدی که وقفه آمد، مقدار یا شماره وقفه روی باس داده فرستاده می‌شود
 - Non-Vector:
- شماره وقفه در مکانی از حافظه که از پیش توافق شده وجود دارد و پردازنده می‌بایست به آن مراجعه کند



برقراری ارتباط بین پردازنده و I/O

- مثال: فرض کنید پردازنده با فرکانس 1GHz داریم که می‌بایست ۱۰۰۰ بایت داده را از ورودی بخواند. سرعت ورودی، یک بایت در هر ۰.۰۲ میلی‌ثانیه است. اگر پردازش داده و ذخیره آن در بافر ۱۰۰۰ سیکل طول بکشد، به سوالات زیر پاسخ دهید:
 - الف) اگر پردازنده با روش سرکشی متوجه حضور داده شود و تکرار سرکشی ۶۰ سیکل طول بکشد، و زمان سرکشی ۶ سیکل باشد زمان کل عملیات چند سیکل می‌شود؟
 - ب) اگر بجای سرکشی از وقفه استفاده کنیم و اجرای وقفه ۲۰۰ سیکل طول بکشد، زمان کل عملیات چند سیکل می‌شود؟

برقراری ارتباط بین پردازنده و I/O



• حل:

(الف)

$0.02 \text{ ms} * 1\text{GHz} - 1000 = 19000$ number of cycles for data transfer

$19000 / 60 = 316.66$ number of polls

$317 * 6 = 1902$ number of polling cycles

$1902 + 1000 = 2902$ number of cycles for each byte (polling + buffering)

$2902 * 1000 = 2902000$ cycle for whole data (1000 bytes)

(ب) با ورود هر بایت، ۱۲۰۰ سیکل صرف پردازش داده (بافرینگ و وقفه) می شود

$1200 * 1000 = 1200000$ cycle for whole data