

# هم طراحی سخت افزار نرم افزار

جلسه بیست و پنجم: ارتباط بین سخت افزار و نرم افزار

ارائه دهنده: آتنا عبدی

[a\\_abdi@kntu.ac.ir](mailto:a_abdi@kntu.ac.ir)

# مباحث این جلسه



- برقراری ارتباط بین اجزای سخت‌افزاری و نرم‌افزاری
  - شیوه‌های برقراری ارتباط
    - حافظه اشتراکی
    - انتقال پیغام
  - واسطه‌های سخت‌افزار و نرم‌افزار
    - ارتباط از طریق باس

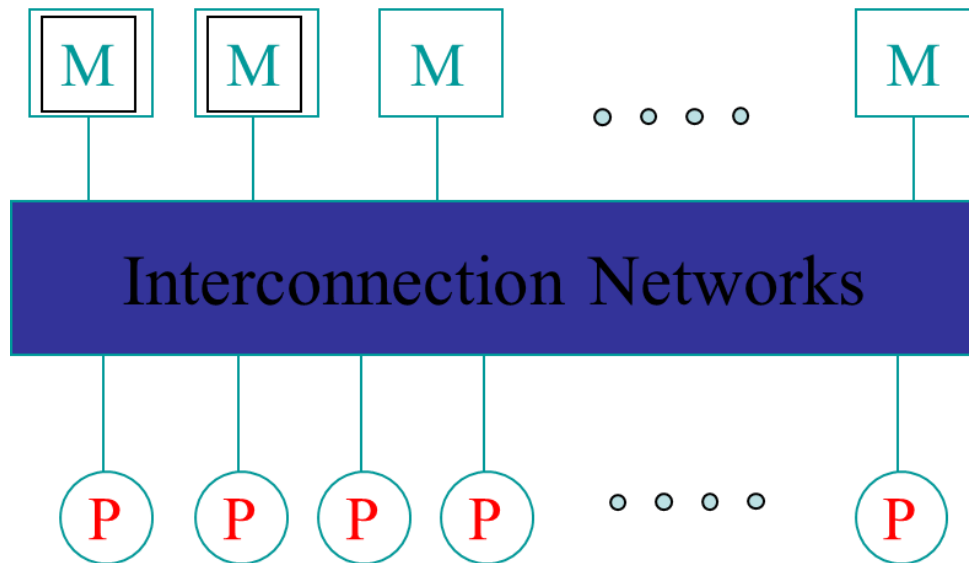
# برقراری ارتباط بین سخت‌افزار و نرم‌افزار



- پس از فرایند سنتز توأم و شکل‌گیری تخصیص و زمان‌بندی وظایف به ایجاد ارتباط می‌رسیم
- تقسیم بحث در این زمینه به دو حیطه

- روش‌های انتقال اطلاعات

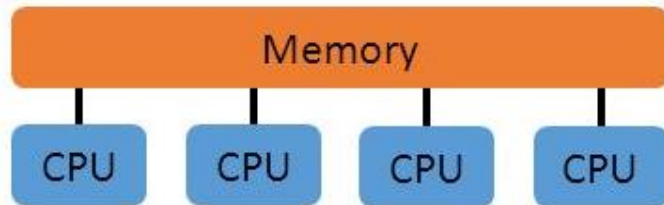
- زیرساخت‌های برقراری ارتباط



# برقراری ارتباط بین سخت افزار و نرم افزار



- پروتکل های انتقال داده

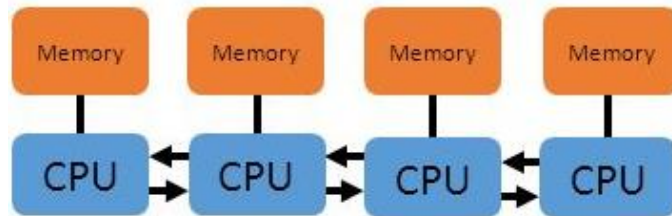


- حافظه مشترک (Shared Memory)

- برقراری ارتباط از طریق خواندن و نوشتن روی فضای حافظه مشترک

- Load/Store Data

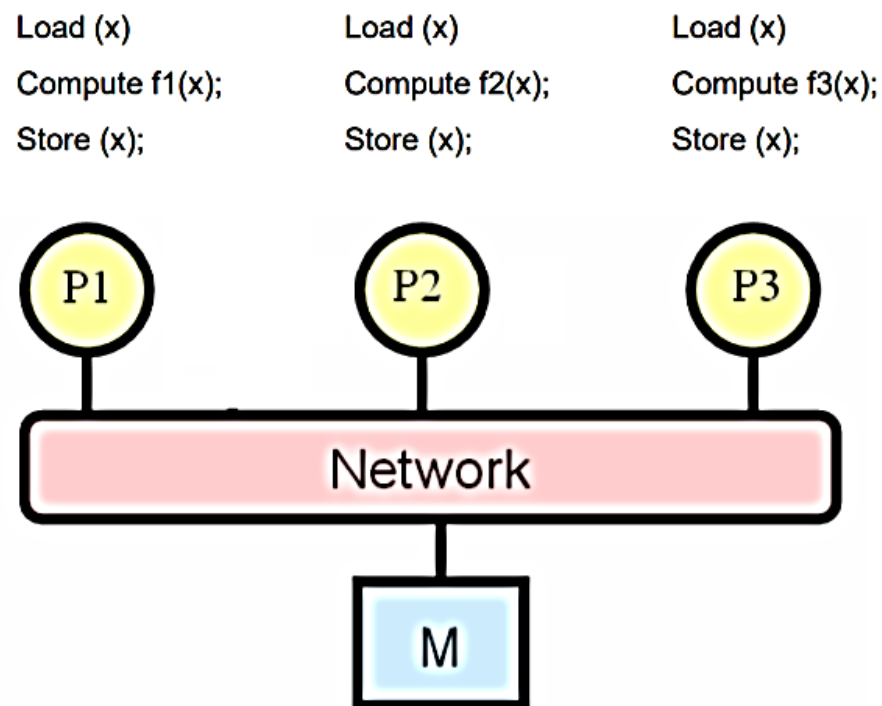
- انتقال پیغام (Message Passing)



- برقراری ارتباط از طریق ارسال و دریافت پیغام

- تخصیص شناسه و بافر مجزا به هر واحد برای ارسال و دریافت

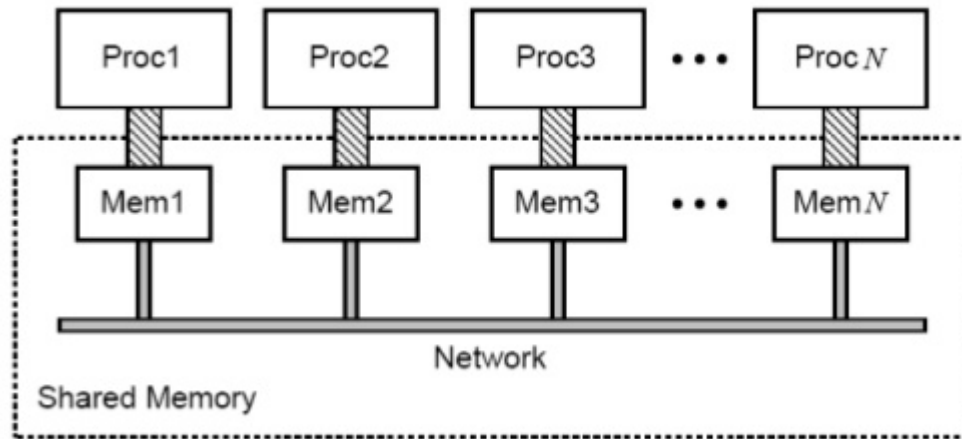
# برقراری ارتباط توسط حافظه مشترک





# برقراری ارتباط توسط حافظه مشترک

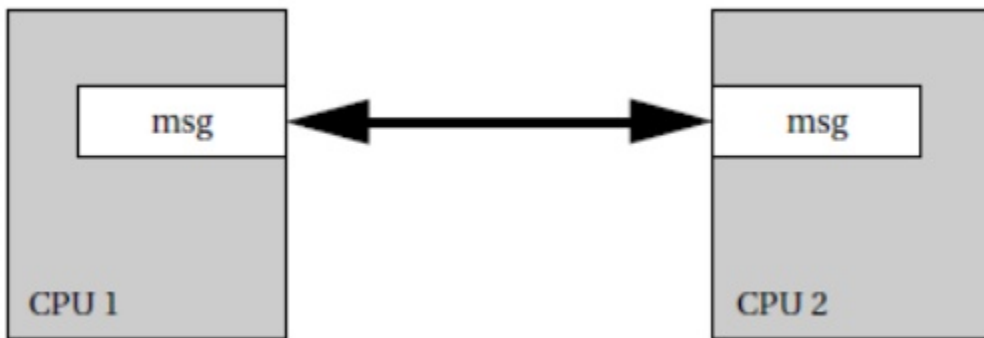
- در حالت قبل، حافظه گلوگاه سیستم است
- راهکار: حافظه مشترک توزیع شده
- امکان دسترسی همه ماژول ها به فضاهای حافظه با ارسال درخواست فراهم است
- این فرایند از دید برنامه نویس مخفی است



# برقراری ارتباط توسط انتقال پیغام



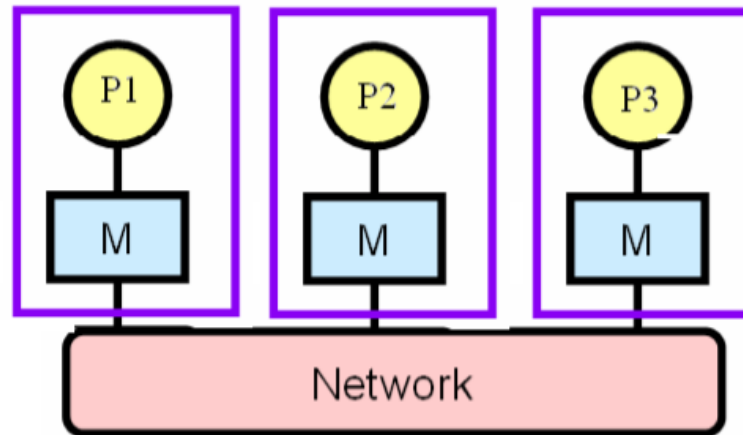
- تعریف فضای آدرس مجزا برای هر پردازنده
- تخصیص حافظه مجزا به هر المان پردازشی
- انتقال داده در قالب پیغام با عملیات send و receive



# برقراری ارتباط توسط انتقال پیغام



Load (x) // from an input or local memory	Wait for (x); //(P2 may Store (x) in local memory)	Wait for (x); //(P3 may Store (x) in local memory)
Compute f1(x);	Compute f2(x);	Compute f3(x);
send (x) to p2;	send (x) to p3;	send (x) to output;







# مقایسه پروتکل‌های برقراری ارتباط

- روش حافظه مشترک:
  - سریع‌تر است و حداکثر سرعت ارتباط را در عین سادگی پیاده‌سازی فراهم می‌کند
  - مشکل سنکرون‌سازی و حفاظت از حافظه دارد
- روش انتقال پیغام:
  - برای سیستم‌های بزرگ و توزیع‌شده مناسب‌تر است
  - سرعت انتقال کمتری دارد

# اتصال سخت افزار و نرم افزار



- بخش واسط که ارتباط سخت افزار و نرم افزار را ایجاد می کند

- Communication Infrastructure

- بمنظور برقراری ارتباط، مکانیزم های گوناگونی مطرح می شوند:

- باس

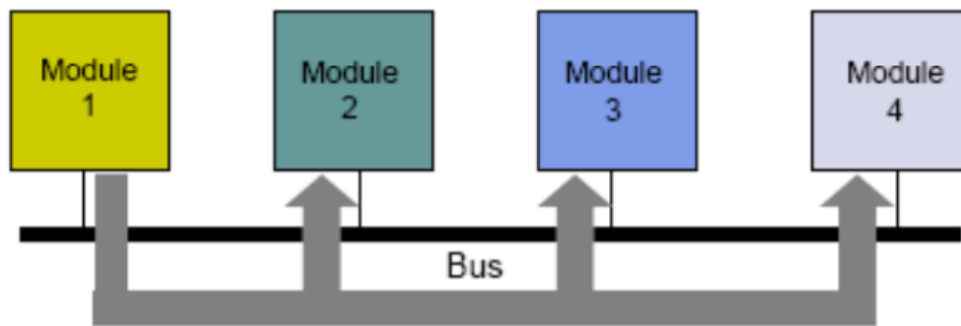
- اتصال نقطه به نقطه

- شبکه های میان ارتباطی

# باس



- ساده‌ترین و رایج‌ترین مکانیزم برقراری ارتباط بین تجهیزات
- اشتراک‌گذاری یک کانال بین چندین ماژول
- ارسال داده به‌صورت همه‌پخشی
- انتخاب مقصد توسط شناسه یا سیگنال‌های کنترلی
- مدیریت برقراری ارتباط در باس
- شکل‌دهی ارتباط توسط واحد کنترلر

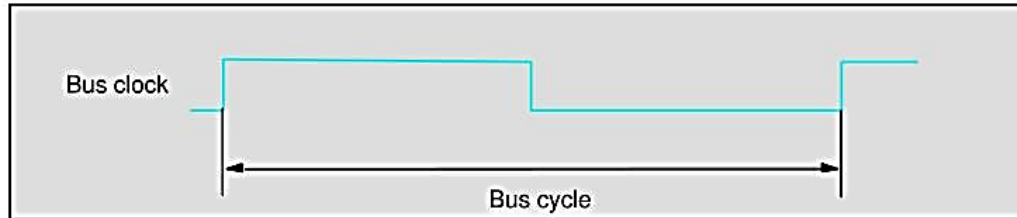




# زمان‌بندی ارسال در باس

- باس سنکرون:

- کنترل و هماهنگی ارتباطات توسط سیگنال کلاک
- مکانیزم مشخص ارتباطی که کلاک در آن دخیل است



- مزیت: سرعت بالا و منطق بسیار ساده

- عیب:

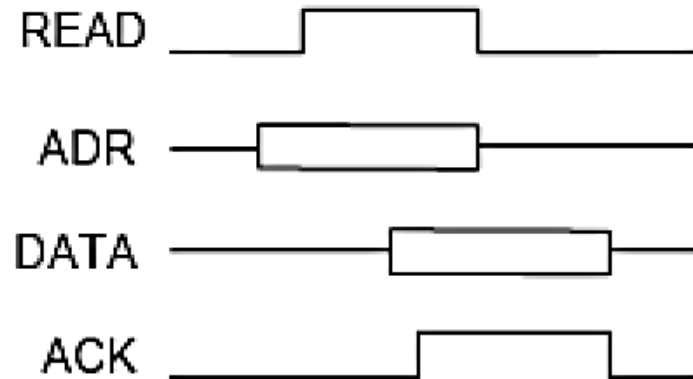
- همه تجهیزات می‌بایست با یک سرعت هماهنگ باشند
- ارتباط به دلیل ناهماهنگی‌های کلاک نمی‌تواند طولانی شود

# زمان‌بندی ارسال در باس (ادامه)



- باس آسنکرون:

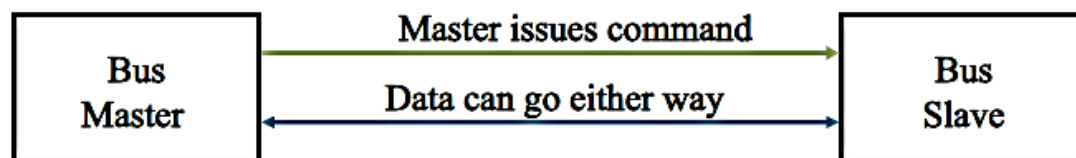
- کلاک در سیستم وجود ندارد
- مقیاس‌پذیری بالایی دارد و تجهیزات زیادی را پشتیبانی می‌کند
- مشکلات ناهماهنگی کلاک وجود ندارد
- هماهنگی توسط مکانیزم‌های دسته‌دهی



# تراکنش ارتباطی روی باس



- هر تراکنش: دنباله‌ای از پیام‌ها که دنبال هم می‌آیند
- دو عملیات در هر تراکنش در نظر گرفته می‌شود:
  - شروع درخواست: Master
  - پاسخ‌دهی و انتقال داده: Slave
- مثال: درخواست خواندن از حافظه و دنبال آن پاسخ شامل داده موردنظر
- در نظر گرفتن یک master ساده‌ترین راهکار است ولی سربار زیادی دارد



# Bus Arbitration



- اشتراک‌گذاری عادلانه باس بین منابع پردازشی مختلف
- فرایند جمع‌آوری درخواست masterها و اعطای دسترسی به باس به یکی از آنها
- انتخاب براساس دآوری بین درخواست‌دهنده‌ها و تعیین اولویت
- ایجاد تعادل بین اولویت و عدالت

