

هم طراحی سخت افزار نرم افزار

جلسه بیست و هشتم: زیرساخت های ارتباطی

ارائه دهنده: آتنا عبدی

a_abdi@kntu.ac.ir

مباحث این جلسه



- برقراری ارتباط بین اجزای سخت‌افزاری و نرم‌افزاری
 - زیرساخت برقراری ارتباط بین اجزا
 - ارتباط از طریق باس
 - ارتباط نقطه به نقطه

برقراری ارتباط بین سخت افزار و نرم افزار

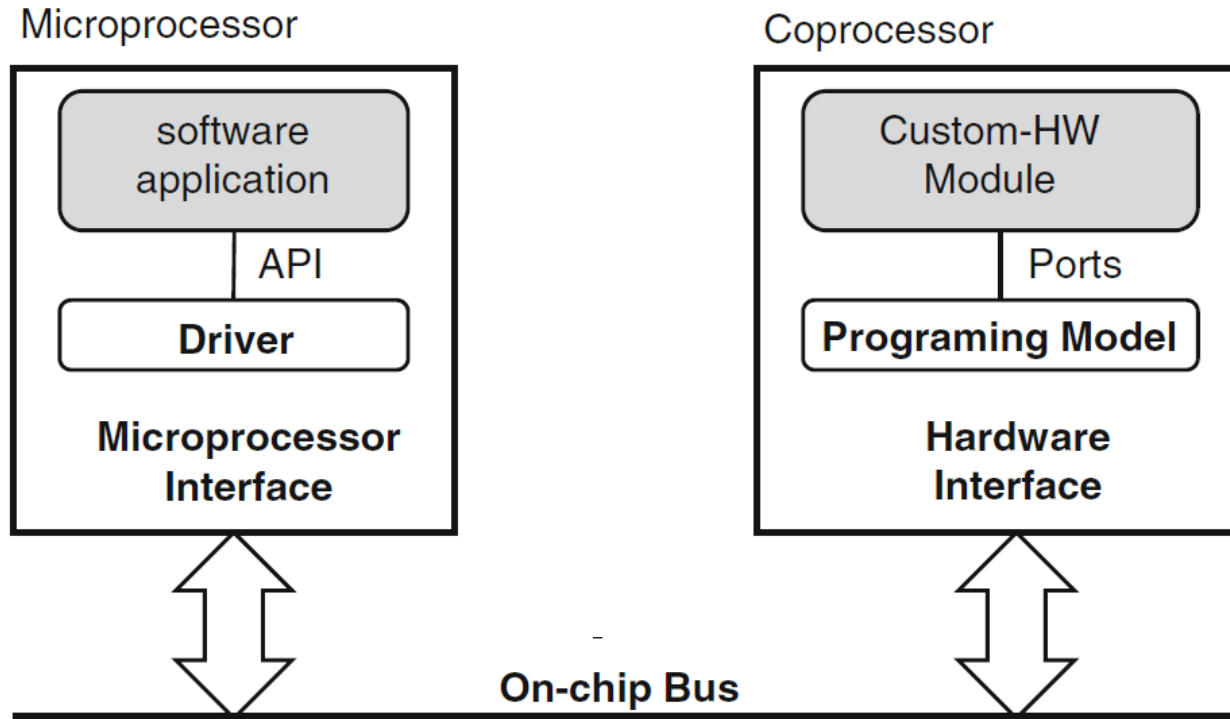


- بمنظور برقراری ارتباط، زیرساخت های گوناگونی مطرح می شوند:

- باس

- اتصال نقطه به نقطه

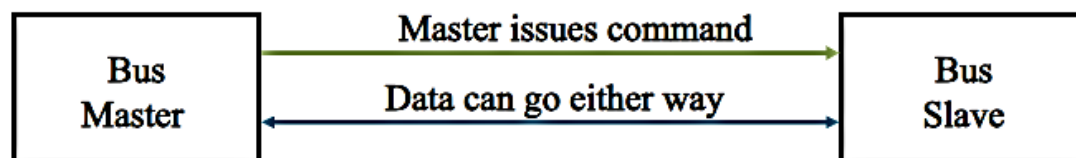
- شبکه های میان ارتباطی



تراکنش ارتباطی روی باس



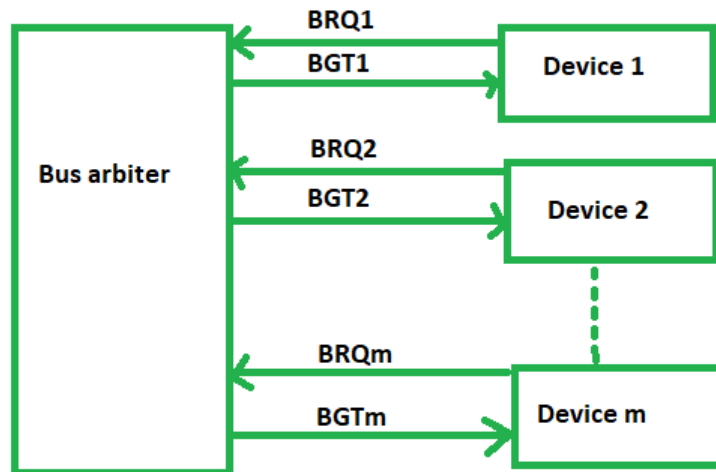
- هر تراکنش: دنباله‌ای از پیام‌ها که دنبال هم می‌آیند
- دو عملیات در هر تراکنش در نظر گرفته می‌شود:
 - شروع درخواست: Master
 - پاسخ‌دهی و انتقال داده: Slave
- مثال: درخواست خواندن از حافظه و دنبال آن پاسخ شامل داده موردنظر
- در نظر گرفتن یک master ساده‌ترین راهکار است ولی سربار زیادی دارد



Bus Arbitration



- اشتراک گذاری عادلانه باس بین منابع پردازشی مختلف
- فرایند جمع آوری درخواست masterها و اعطای دسترسی به باس به یکی از آنها
- انتخاب براساس دآوری بین درخواست دهنده ها و تعیین اولویت
- ایجاد تعادل بین اولویت و عدالت



Bus Arbitration (ادامه)



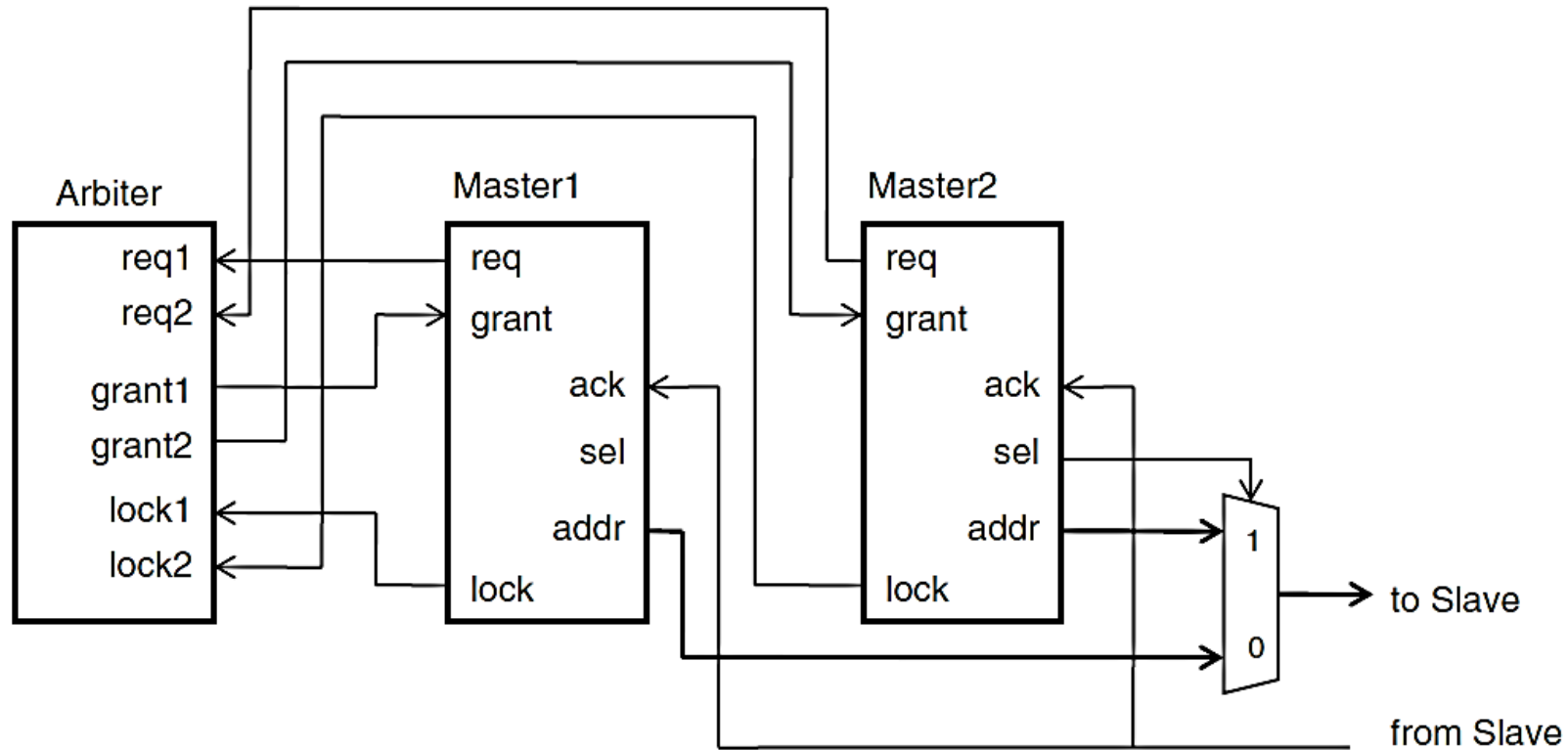
- مکانیزم داوری با هدف ایجاد دسترسی انحصاری به باس
 - به اندازه مشخص و ثابت
 - به اندازه نیاز سیستم و متغیر
- رعایت عدالت در داوری و تعریف عدالت بصورت
 - مساوات در سرویس دهی
 - سرویس دهی وزن دار
 - سرویس دهی بصورت مکانیزم FIFO

مکانیزم‌های Bus Arbitration



- صف کشیدن (Queuing)
- مرتب‌سازی درخواست‌ها به صورت FIFO و ارسال grant به ترتیب ورود به سیستم
- اولویت ثابت (Fixed Priority)
- لحاظ کردن اولویت‌های ثابت و از قبل مشخص شده که ممکن است منجر به گرسنگی شود
- Round Robin
- تخصیص کمترین اولویت به تجهیزاتی که به تازگی باس را در اختیار داشته است
- TDMA
- تخصیص ایستای هر slot زمانی به یک master

Bus Arbitration

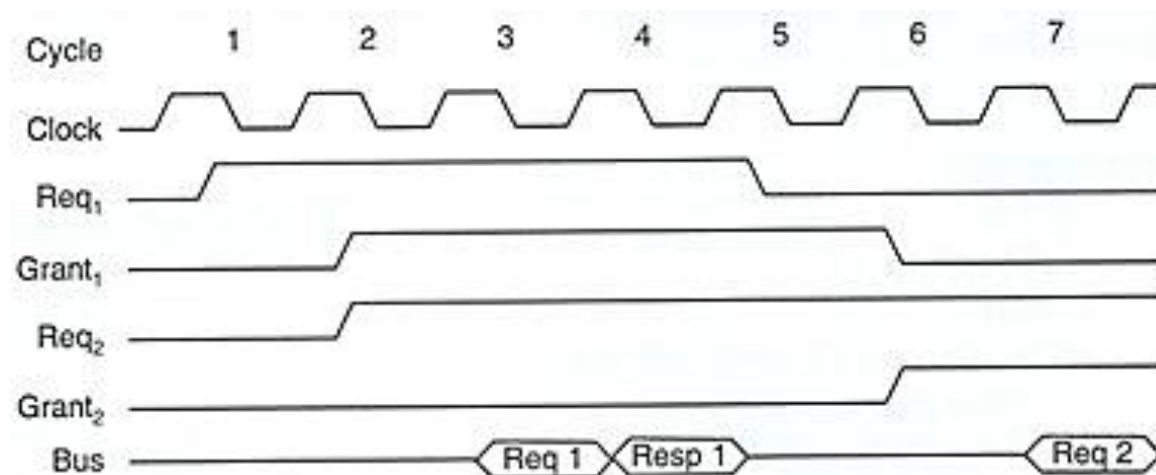


چالش‌های باس



- باس یک منبع مشترک است و در هر لحظه توسط یک ماژول قابل استفاده است
- تاخیر زیاد انتقال اطلاعات

- دنباله پیام‌های مبادله‌شونده در شکل‌گیری یک تراکنش ارتباطی روی باس
- قفل شدن باس تا پایان مبادله و طولانی شدن انتظار منابع و کاهش کارایی



پیاده‌سازی خطلوله در باس



- فرایند دسترسی به باس، ترتیبی و متشکل از چندین مرحله می‌باشد
- دخیل نبودن باس در تمامی مراحل ایجاد دسترسی و انتقال و امکان بکارگیری روش خطلوله
- موازی‌سازی بخش‌هایی که اجرای وابسته ندارند
- در هر لحظه باس در اختیار یک ماژول است

Write Access

| | AR | ARB | AG | RQ | ACK |
|-------------|----|-----|----|----|-----|
| Arb request | | | | | |
| Arbiter | | | | | |
| Arb grant | | | | | |
| Bus | | | | | |

Read Access

| | AR | ARB | AG | RQ | P | RPLY |
|-------------|----|-----|----|----|---|------|
| Arb request | | | | | | |
| Arbiter | | | | | | |
| Arb grant | | | | | | |
| Bus | | | | | | |

پیاده‌سازی خطلوله در باس (ادامه)



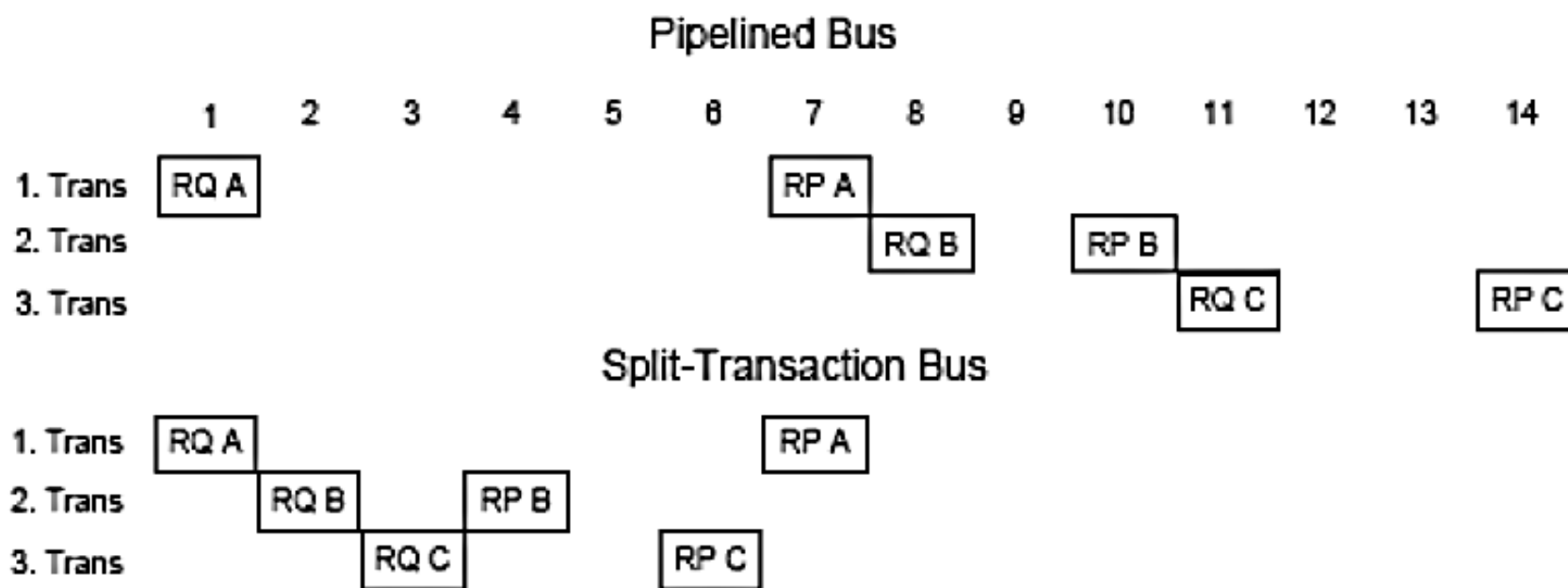
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------|----|-----|-----|-----|-------|-------|-----|-------|-----|-------|-----|----|-------|-------|----|
| 1. Read | AR | ARB | AG | RQ | P | RPLY | | | | | | | | | |
| 2. Write | | AR | ARB | AG | Stall | Stall | RQ | ACK | | | | | | | |
| 3. Write | | | AR | ARB | Stall | Stall | AG | Stall | RQ | ACK | | | | | |
| 4. Read | | | | AR | Stall | Stall | ARB | Stall | AG | Stall | RQ | P | RPLY | RQ | |
| 5. Read | | | | | | | AR | Stall | ARB | Stall | AG | RQ | P | RPLY | |
| 6. Read | | | | | | | | | AR | Stall | ARB | AG | Stall | Stall | RQ |
| Bus busy | | | | | | | | | | | | | | | |

مکانیزم تقسیم تراکنش



- تراکنش‌های باس به دو بخش تقسیم می‌شوند (Split-Transaction):
 - درخواست و پاسخ
 - گاهی انجام عملیات پس از درخواست، زمانی طول می‌کشد
 - در این مدت زمان master آزاد باشد تا عملیات دیگری شروع شود
 - افزایش موازی‌سازی
 - افزایش سربار سخت‌افزاری به سیستم

مکانیزم‌های افزایش سرعت باس



انتقال Burst mode

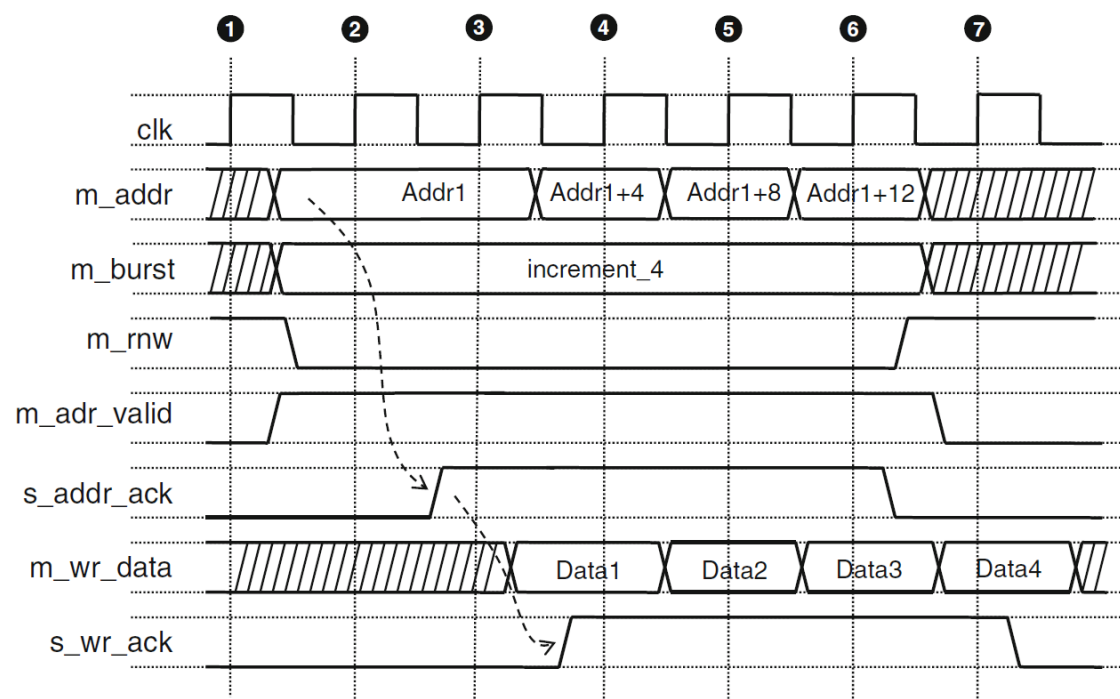


- روش دیگر بمنظور افزایش کارایی باس می باشد

- انتقال اطلاعات مشابه در آدرس های نزدیک بصورت همزمان

- مشابه آنچه در حافظه نهان رخ می دهد

- در نظر گرفتن طول ثابت و طول متغیر برای انتقال داده



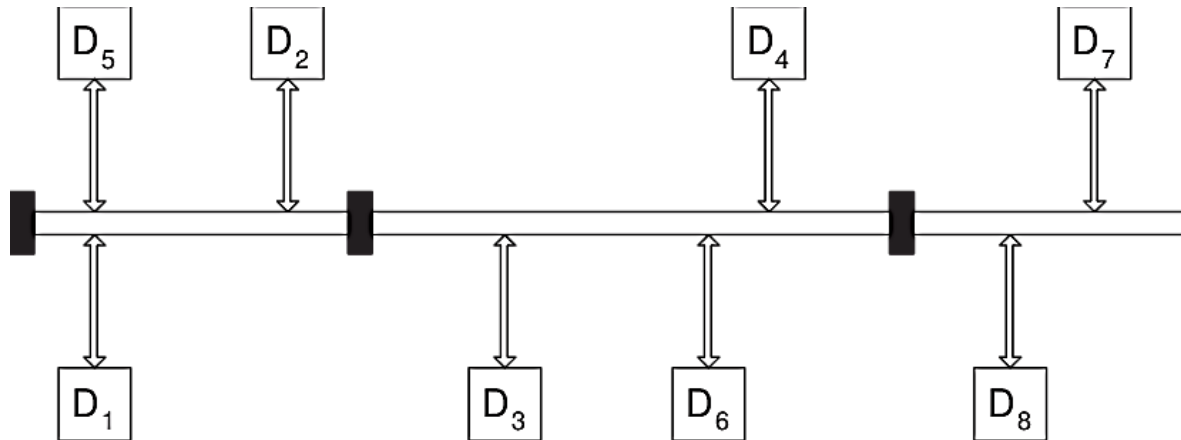
چالش‌های باس (ادامه)



- چالش دیگر باس مسئله مقیاس‌پذیری است

- چالش‌های کارایی و توان مصرفی با افزایش تعداد تجهیزات روی باس شدیدتر می‌شوند

- راهکار بخش‌بندی کردن باس، تا حدی مشکل را حل می‌کند





باس بخش‌بندی شده

- هر بخش از باس master و slave دارد
- اتصال بخش‌ها توسط bridge که می‌تواند ورودی یا خروجی باشد
- مشخص کردن بخش کنترل کننده باس در هر لحظه توسط arbiter

