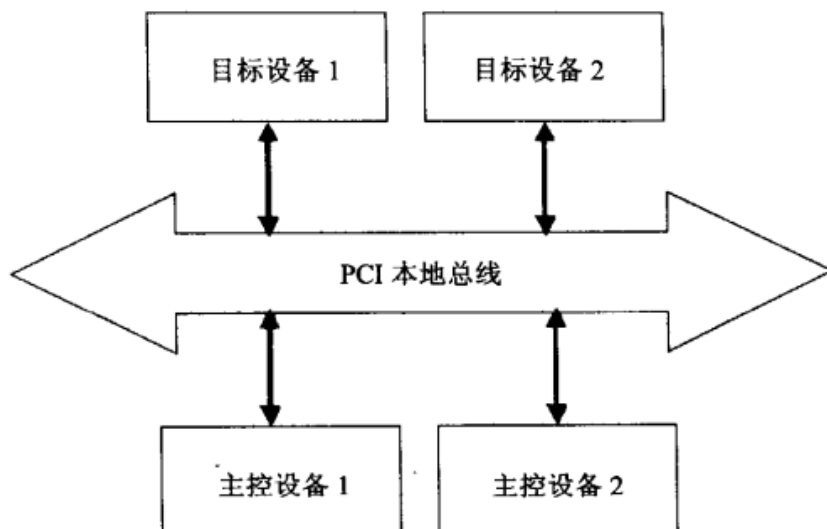


## PCI 系统



- PCI 总线会挂载很多设备。每个设备都可以成为主机（master）或从机（slave）。
- 必须通过一个仲裁过程来确保在任何时刻只有一个主机控制总线，这个仲裁过程由 PCI 总线上的仲裁器负责协调。
- 只有成为主机的设备才能发起事务。此时，其它设备自动成为从机，并根据主机的请求执行操作。
  - PCI 事务
    - 读事务：主机向从机发起请求，读从机中的数据
    - 写事务：主机向从机发起请求，向从机中写数据

## PCI 设备的端口列表

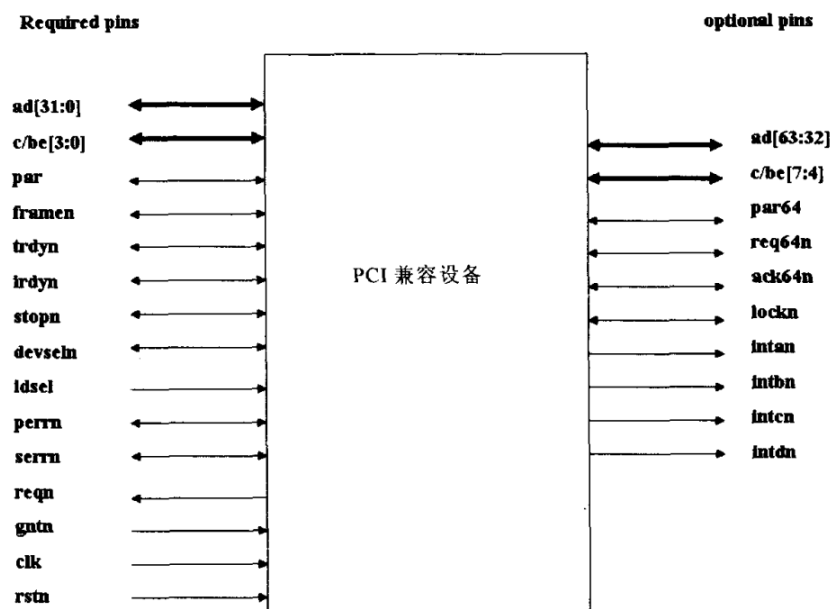


图 6-1 PCI 兼容设备

1. ad[31:0]

- 命名规则：ad = address + data
- 这是 32 位宽的地址/数据总线，用于传输地址和数据。在 PCI 总线的操作中，这些引脚在不同的时钟周期中可以承载地址或数据，这是通过总线的控制信号来确定的。这种复用（地址和数据使用相同的引脚）可以减少所需的引脚数量，降低成本和复杂度。

2. c/be[3:0]

- 命名规则：c/be = command / byte enable
- 当 ad[31:0] 总线上是地址时，该信号包含下表中的一种指令，用来指示从机进行何种操作（读/写）。

表 6-1 PCI 总线指令	
C/BE[3:0]	命令类型
0000	中断确认(Interrupt Acknowledge)
0001	特殊周期(Special Cycle)
0010	I/O 读
0011	I/O 写
0100	保留
0101	保留
0110	存储读(Memory Read)
0111	存储写(Memory Write)
1000	保留
1001	保留
1010	配置读(Configuration Read)
1011	配置写(Configuration Write)
1100	多次存储器读(Memory Read Multiple)
1101	双重地址周期(Dual Address cycle)
1110	存储读线
1111	存储写、无效(Memory Write and Invalidate)

- 当 ad[31:0] 总线上是数据时，该信号用于指示当前 ad[31:0] 总线上的哪些字节是有效的。

3. par

- 命名规则：parity
- 在 ad[31:0], c/be[3:0], par 中应该一共有偶数个 1 出现。
- 该校验位的值被驱动的时刻是在 ad[31:0] 被驱动的一个时钟周期之后。

4. framen

- 命名规则：frame + n, n 表示低位有效
- 该信号是一个控制信号，用于指示一个 PCI 事务的开始和结束。
- 当 PCI 主机开始一个事务时，它会将该信号拉低；在完成事务的最后一个数据阶段后，等待下一个时钟周期结束时，PCI 主机将该信号拉高，以此结束事务。
- 在一次事务期间，如果该信号被断言，那么其他设备知道总线正忙，不会尝试开始新的事务。当该信号被释放，总线上的设备知道当前的事务已经结束，总线已经空闲，因此其他设备可以请求使用总线。

5.1 trdyn

- 命名规则：target + ready + n, n 表示低位有效
- 该信号如果拉低，则代表从机准备好向主机传输数据或接受主机的数据。
- 如果从机还没准备好，该信号会保持在高电平。

## 5.2. irdyn

- 命名规则: initiator + ready + n, n 表示低位有效
- 该信号如果拉低, 则代表主机准备好向从机发送数据或者接收从机的数据。
- 如果主机还没准备好, 该信号会保持在高电平。

## \*7. stopn

- 命名规则: stop + ready + n, n 表示低位有效
- 该信号如果拉低, 则代表从机无法继续接受或发送数据。
- 主机则会在该信号拉低后, 在当前或下一个数据阶段结束事务, 并释放总线。

## 8.1. devseln

- 命名规则: device + select + n, n 表示低位有效
- 当主机通过 ad[31:0] 发送地址的时候, 所有的从机都会检查这个地址, 以确定是否是它们应该响应的。符合条件的从机将该信号拉低, 表示它已经被选择并准备好响应事务。

## \*8.2. idsel

- 命名规则: initiator + select
- 当主机通过 ad[31:0] 发送地址的时候, 所有的从机都会检查这个地址, 以确定是否是它们应该响应的。如何决定哪一个从机响应呢?
- 每个从机通过该信号与 ad[31:0] 上的一个特定地址线相连。如果这个特定的地址线被激活, 那么对应的从机就会"认为"自己被选中, 进而响应该地址。

## \*9.1. perrn

- 命名规则: parity + error + n, n 表示低位有效
- 该信号用于报告数据传输中的奇偶校验错误。如果在数据阶段检测到奇偶校验错误, 那么产生错误的设备会拉低该信号来通知其他设备发生了错误。

## \*9.2. seern

- 命名规则: system + error + n, n 表示低位有效
- 该信号用于报告严重错误, 产生错误的设备会拉低这个信号来指示系统发生严重错误。

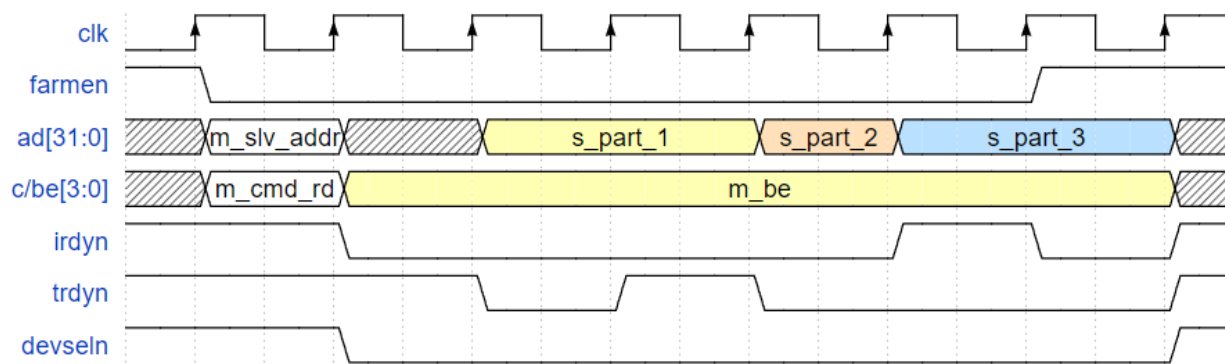
## \*10.1. reqn

- 命名规则: request + n, n 表示低位有效
- 该信号由希望成为主机的 PCI 设备用来请求总线控制权。当设备需要发送或接收数据时, 它会通过拉低该信号来发出请求。

## \*10.2. gntn

- 命名规则: grant + n, n 表示低位有效
- 该信号由 PCI 总线的仲裁器用来响应 reqn 请求, 并赋予设备总线控制权。当仲裁器准备好授予请求设备控制总线时, 它会拉低相应设备的 gntn 信号

## PCI 读事务



注：1.  $m\_slv\_addr$  表示主机写入总线的从机地址； $s\_part1,2,3$  表示从机写入总线的分块数据  
 $m\_cmd\_rd$  表示主机写入总线的读命令， $m\_be$  表示主机写入总线的字节允许标志  
2. 分别假设了两种情况，一种是从机忙碌一周，第二种是主机忙碌一周

1. 读事务的流程：master 发起读事务，选择一个 slave，slave 将数据返回 master。

2. 具体时序：

- 寻址阶段

- 第一个上升沿

- 拉低 farnen 信号，指示事务的开始；
    - 对 ad[31:0] 写入特定从机的地址；
    - 对 c/be[3:0] 写入读命令，指示当前事务为读事务；

- 转换阶段

- 第二个上升沿

- 所有 slave 通过对 ad[31:0] 解码来识别自己，一旦识别自己，则拉低 devseln 信号；
    - 由于已经选择到了特定的 slave，因此 master 停止驱动 ad[31:0]；
    - 同时，master 将拉低 irdyn 信号表示它已准备好进行事务；
    - 脱离地址阶段，因此 master 对 c/be[3:0] 写入字节允许标志；

- 数据阶段

- 第三个上升沿

- 从机将请求的数据放置在 ad[31:0] 总线上；
    - 并将 trdyn 信号拉低来表示数据已发送；

- 第四个上升沿

- irdyn 和 trdyn 均为低，主机从 ad[31:0] 总线读取数据；
    - 假设情况 1：将 trdyn 信号拉高，即假设从机需要一个周期才能将新的数据部分放置在 ad[31:0] 总线上，此时 ad[31:0] 总线上保留上部分的数据；

- 第五个上升沿

- 从机将请求的数据放置在 ad[31:0] 总线上；
    - 并将 trdyn 信号拉低来表示数据已发送；

- 第六个上升沿

- **irdyn** 和 **trdyn** 均为低，主机从 **ad[31:0]** 总线读取数据；
- 假设情况 2：将 **irdyn** 信号拉高，即假设主机在读完第二部分数据后进入忙碌状态，持续一个周期；
- 注：此时从机没有忙碌 (**trdyn** 信号是低位的)，表示从机已经将新的数据放置在了 **ad[31:0]** 总线上。
- 第七个上升沿
  - 检测到数据的最后部分，因此主机拉高 **framen** 信号，指示事务即将结束；
  - 将 **irdyn** 信号拉低来表示主机结束了忙碌状态，可以读取数据；
- 第八个上升沿
  - **irdyn** 和 **trdyn** 均为低，主机从 **ad[31:0]** 总线读取数据；
  - 主机释放总线