

Avalon MM-Master

1. Avalon-MM Master

[Signal Types](#)

[Classical Transfer Scenarios](#)

[Classical One](#)

[Master Interface Burst Transfer](#)

[Burst Write](#)

[Burst Read](#)

FROM: <https://blog.csdn.net/chen495277820/article/details/79068877>

1. Avalon-MM Master

Signal Types

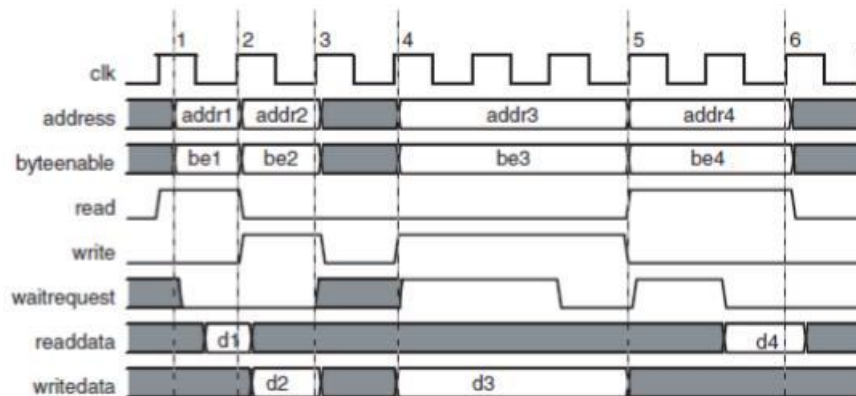
Some Signals that I have used in the MINI-PROJECT of CS-473 is shown in the Table

信号类型	宽度	方向	描述
<code>read</code> <code>read_n</code>	1	输出	读请求信号。如果该信号存在，那么需要 <code>readdata</code>
<code>write</code> <code>write_n</code>	1	输出	写请求信号。如果该信号存在。那么需要 <code>writedata</code>
<code>address</code>	1-32	输出	指定byte地址。
<code>readdata</code>	8,16,32,64,128,256,512,1024	输入	读过程读出的数据
<code>writedata</code>	8,16,32,64,128,256,512,1024	输出	写过程从总线上传过来的数据，数据的宽度需要和 <code>readdata</code> 的宽度相同。
<code>byteenable</code> <code>byteenable_n</code>	1,2,4,8,16,32,64,128	输出	读写过程中字节使能信号。在读过程中指定那些字节的信号能够被读取，在写过程中指定哪些字节能够被写入。
<code>waitrequest</code> <code>waitrequest_n</code>	1	输入	使主接口等待总线完成工作。在等待过程中保持控制信号不变
<code>readdatavalid</code> <code>readdatavalid_n</code>	1	输入	用于流水式读取过程。表示在信号线上是否出现了需要的信号
<code>burstcount</code>	1-32	输出	用于表示突发传输的次数
<code>resetrequest</code> <code>resetrequest_n</code>	1	输出	复位请求

Classical Transfer Scenarios

Classical One

- 传输在第一个时钟上升沿开始，在第一个周期 `address`，`byteenable` 和 `read` 或者 `write` 信号有。
- 如果 `waitrequest` 信号被置位，那么主接口需要等待传输，并保持信号不变。传输在 `waitrequest` 信号失效后的第一个时钟上升沿结束



1. 第一个时钟上升沿，主设备提供`address`，`byteenable`信号，置位`read`，开始一次读传输。
2. 在第二个时钟上升沿锁存`readdata`。并可以开始其他传输过程。图中`read`信号清零，置位`write`，`writedata`，`address`信号，开始一个写传输。
3. 第三个时钟上升沿结束写传输，清除所有信号，随时可以开始新的传输过程。
4. 开始新的写传输过程，用`waitrequest`信号延时。

Master Interface Burst Transfer

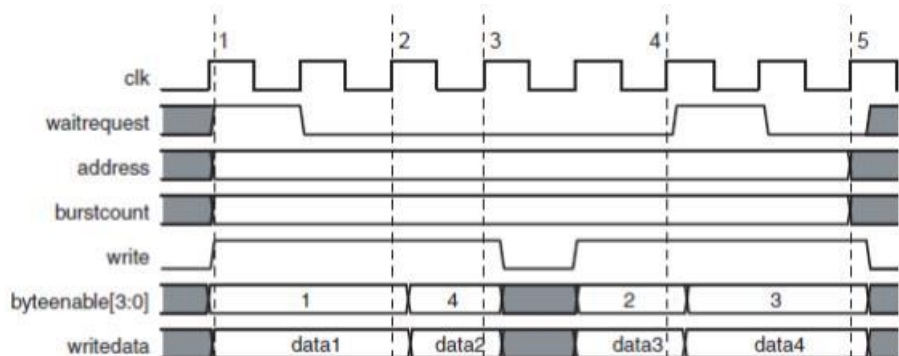
- `burstcount` 信号表示了突发传输的长度。
- 主接口只给一次地址，其余地址由从接口进行推算。
- 突发传输只有完成了规定次数的传输才能够停止

Burst Write

主接口需要给 `address`，`writedata`，`write`，`byteenable` 和 `burstcount` 信号来开始突发写传输。`writedata` 信号在在第一个时钟上升沿被获取。在整个传输过程中需要保持 `address` 和 `burstcount` 信号不变

- 可以通过 `write` 信号来暂停读传输。
- 当 `waitrequest` 信号被置位时，主接口需要保持 `address`，`writedata`，`write` 和 `byteenable` 不变

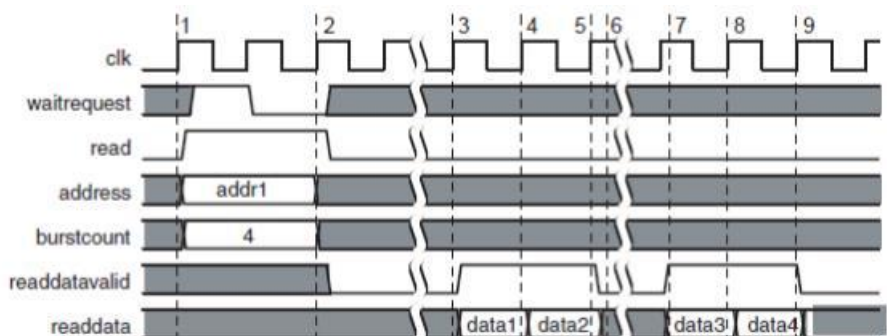
Burst Read



1. 第一个时钟上升沿，主设备提供address, data, byteenable, burstcount 置位write, 开始写传输。整个过程中address和burstcount不变。
2. 在waitrequest为低后的第一个时钟上升沿写入数据。
3. 4. 可以用write, waitrequest信号暂停传输过程。两者的暂停有区别。
5. 传输达到burstcount规定数目后结束burst传输过程。

主接口突发读传输是流传输的一种形式。主接口需要给 `address`, `read` 和 `burstcount` 信号来开始突发读传输。当 `waitrequest` 信号无效时，地址读写结束。传输过程中，`readdatavalid` 信号可以暂停传输。主接口没法停止突发读传输过程。

- 在 `readdatavalid` 置位时，主接口需要读取 `readdata`
- 在读写地址时，`byteenable` 信号需保持不变



1. 第一个时钟上升沿，主设备提供address, burstcount, 置位read, 开始读传输。
2. 在waitrequest清零后的第一个时钟上升沿无效read, address, burstcount。
4. 5. 8. 9. `readdatavalid`置位的时钟上升沿锁存readdata。直到读到的数据个数和burstcount相同后结束burst传输。