# DDR的数传输控制

## 一，总体介绍

DDR的数据的读写是通过axi总线进行数据传输。AXI（Advanced eXtensible Interface）是一种总线协议，该协议是ARM公司提出的AMBA（Advanced Microcontroller Bus Architecture）3.0协议中最重要的部分，是一种面向高性能、高带宽、低延迟的片内总线。它的地址/控制和数据相位是分离的，支持不对齐的数据传输，同时在突发传输中，只需要首地址，同时分离的读写数据通道。

AXI4和AXI4-Lite包含5个不同的通道：

1. 读地址通道；
2. 写地址通道；
3. 读数据通道；
4. 写数据通道；
5. 读响应通道；

利用网络上开源的aq\_axi\_master模块<https://github.com/aquaxis/IPCORE/tree/master/aq_axi_vdma> 进行axi总线的数据传输控制。该模块AXI的读写信号生成模块，将axi传输协议封装在模块中，并留出数据读写控制接口，用于简化axi数据传输操作。

用户控制端口分为读控制通道和写控制通道。下面分别做介绍。

## 二，写控制通道。

用户控制写通道的端口信号如下组成：

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 方向 | 描述 |
| **WR\_START** | 1 | input | 写开始信号。该信号有效后，启动一次突发传输。 |
| **WR\_ADRS** | 32 | input | 突发传输的起始地址。 |
| **WR\_LEN** | 32 | input | 一次突发传输的数据长度，以字节为单位。 |
| **WR\_READY** | 1 | output | 写状态。当axi处于空闲时，该信号为高。其余状态下，该信号为低。 |
| **WR\_FIFO\_RE** | 1 | output | 读fifo（数据）使能信号。该信号有效时，表示需要将有效数据送到写数据端口。 |
| **WR\_FIFO\_EMPTY** | 1 | input | fifo空标志位。 |
| **WR\_FIFO\_AEMPTY** | 1 | input | fifo几乎空标志位。 |
| **WR\_FIFO\_DATA** | 64 | input | 写数据端口。要传输的数据通过该端口输入。 |
| **WR\_DONE** | 1 | output | 写突发传输完成的标志位。一个时钟高电平。 |

【注】

1，一般数据是从先缓存在fifo中，再从fifo中读出，通过axi总线再进行一次突发传输。

1. WR\_FIFO\_RE是fifo的写读能信号，当axi总线正在传输数据时，该信号有效，通知数据提供模块发出新的数据。由于数据的输出一般都有一个时钟的延迟，故该信号要在有效数据发送前提前一个时钟有效，保证有效数据正好对齐。
2. 传输地址和突发长度是以字节为单位的，当数据位宽为64bit时，一次传输的数据为8字节，地址增量为8。一次突发长度为256时，突发地址的增量为 256\*8 =2048。
3. 在进行一次数据突发传输时，将**WR\_START**信号置高的同时，将突发首地址和突发长度放置到**WR\_ADRS**，**WR\_LEN**端口，然后等待**WR\_FIFO\_RE**有效时，将要突发传输的数据依次放置于**WR\_FIFO\_DATA**端口，直到**WR\_DONE信号为高，标志一次**突发传输完成。进行下一次的突发传输时，重复以上步骤即可。

5，**WR\_START** 信号可以保持多个时钟有效，但必须在RD\_DONE信号为高后将该信号置低，否则会自动触发下一次的突发传输。

6，空闲状态下，**WR\_START**信号要置为低**， WR\_ADRS**，**WR\_LEN**端口的数据在突发传输开始时更新，在其他时间可保持不变。内部模块会在空闲状态且 **WR\_START** 为高时接收寄存这些数据，启动一次突发传输。

7，在**WR\_READY** 信号为高或者接收到**WR\_DONE信号为高**后，传输进入空闲状态，此时再启动一次突发传输。或在非空闲态就启动开始信号，将地址和传输长度放置在总线上，等到进入空闲态时，会自动进入下一次突发传输。

## 读控制通道。

用户控制写通道的端口信号如下组成：

|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位宽 | 方向 | 描述 |
| RD\_START | 1 | input | 读开始信号。用于启动一次读突发传输。 |
| RD\_ADRS | 32 | input | 读突发首地址。 |
| RD\_LEN | 32 | input | 一次突发传输的数据长度，以字节为单位。 |
| RD\_READY | 1 | output | 读状态。当axi处于空闲时，该信号为高，其余状态下，该信号为低。 |
| RD\_FIFO\_WE | 1 | output | 写fifo使能信号。当该信号有效时，表示读数据端口输出有有效的数据。可用该信号作为fifo的写使能信号，将数据写入到fifo中。 |
| RD\_FIFO\_FULL | 1 | input | fifo满标志位。 |
| RD\_FIFO\_AFULL | 1 | input | fifo将满标志位。 |
| RD\_FIFO\_DATA | 64 | output | 读数据端口。axi总线读出的数据通过该端口输出。 |
| RD\_DONE | 1 | output | 读突发传输完成的标志位。一个时钟高电平。 |

【注】

1. 一般数据读出后先缓存到fifo中，等fifo中的数据达到一定数量时，再由其他模块读取。

2，RD\_FIFO\_WE 信号是输出数据的有效信号，当该信号为高时，表示读数据端口有有效的数据输出。可将该信号作为fifo的写使能信号。该信号与输出数据保持对齐。

3，传输地址和突发长度是以字节为单位的，当数据位宽为64bit时，一次传输的数据为8字节，地址增量为8。一次突发长度为256时，突发地址的增量为 256\*8 =2048。

4，在进行一次读突发数据传输时，将**RD\_START**信号置高的同时，将突发首地址和突发长度放置到**RD\_ADRS**，**RD\_LEN**端口，然后等待**RD\_FIFO\_RE**有效时，有效的读数据从**RD\_FIFO\_DATA**端口输出，直到**RD\_DONE信号为高，标志一次**突发传输完成。进行下一次的突发传输时，重复以上步骤即可。

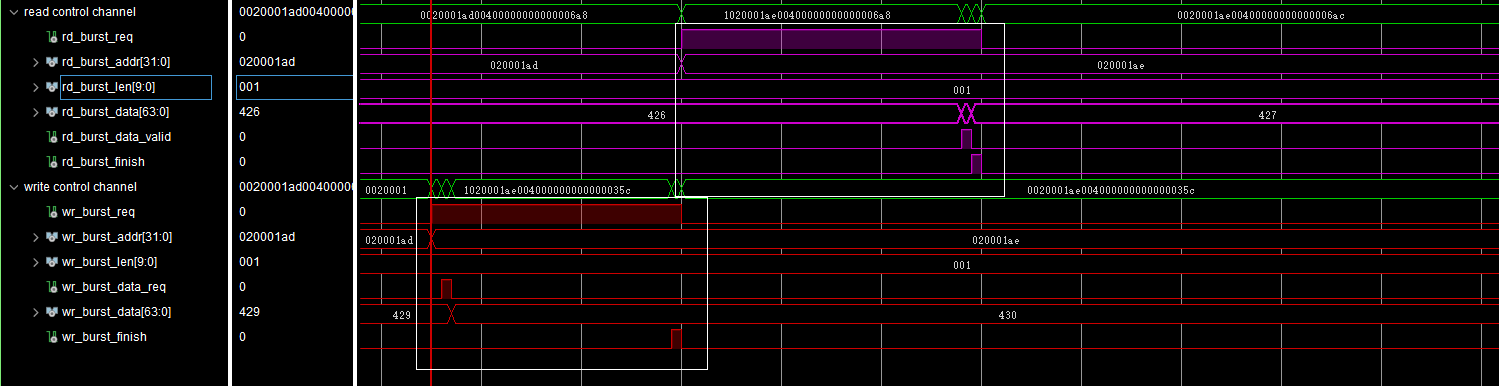
1. **RD\_START** 信号可以保持多个时钟有效，但必须在RD\_DONE信号为高后将该信号置低，否则会自动触发下一次的突发传输。
2. 空闲状态下，**RD\_START**信号要置为低**， RD\_ADRS**，**RD\_LEN**端口的数据在 突发传输开始时更新，在其他时间可保持不变。内部模块会在 空闲状态且 **RD\_START** 为高时接收寄存这些数据，启动一次突发传输。

7，在**RD\_READY** 信号为高 或者接收到 **RD\_DONE信号为高**后，传输进入空闲状态，此时再启动一次突发传输。或在非空闲态就启动开始信号，将地址和传输长度放置在总线上，等到进入空闲态时，会自动进入下一次突发传输。

## 数据传输设计的一些细节。

1. 突发长度最小可以设置为1.
2. RD\_START，WR\_START信号在启动一次数据传输时可以一直为高，但要在RD\_DONE，WR\_DONE为高时将该信号置低，否则会自动触发下一次传输，造成数据传输错误。
3. 读出的数据和RD\_FIFO\_WE 标志位对齐。写入的数据为 WR\_FIFO\_RE 信号有效之后传输进写数据端口的数据。

如下图，对应的是突发长度为1的写+读数据验证，先向设定的地址写入长度为1的数据，然后再读出该地址的数据。比较写入的数据和读出的数据是否一致。



如图，写入的数据为 430，读出的数据也为430.

