

The Logical Layer

Data on the Rambus Channel moves in blocks. The size of these block-oriented transfers can be optimized to match the needs of each particular system. To implement this, the Rambus Channel defines a protocol with the following three types of packets:

- Request;
- Acknowledge;
- Data.

The combination of a request, an acknowledge, and a data packet constitute a transaction. The following types of transactions are defined:

- Read memory space;
- Write memory space;
- Read register space;
- Write register space;
- Broadcast write register space.

To ensure proper synchronization of all devices connected to the Rambus Channel, request, acknowledge and data packets all begin during even intervals (falling clock edge).

Request Packet

A request is issued by the master. Each request is, as depicted in Figure 7, six intervals long. There are ten bits per interval. A request contains:

- A start bit (Start);
- An opcode (Op[3:0]);
- An address (Adr[35:0]);
- A count (Count[7:0]).

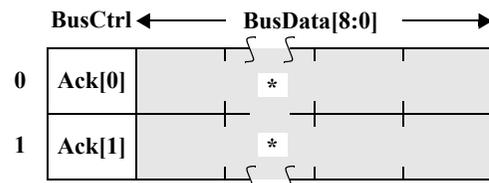
The opcode specifies the type of data transfer to take place. Read and write operations are defined for both

memory and register spaces. Each slave device contains configuration registers (in the register space) in addition to memory space.

The 36-bit address specifies the first byte that is transferred. From 1 to 256 bytes can be moved with a single transaction, as specified by the 8-bit count field.

Acknowledge Packet

Upon receipt of a request, the addressed slave responds with an acknowledge. As shown in Figure 8, an acknowledge is sent across the BusCtrl wire to the master. This may be concurrent with a data packet.



* May contain a portion of a Data Packet

Figure 8: Acknowledge Packet Format

Coding of the acknowledge packet is given in Table 1.

Table 1: Acknowledge Packet Encoding

Ack[1:0]	Definition
00	Addressed slave does not exist
01	Okay; slave will respond to request
10	Nack; slave busy, try request later
11	Reserved

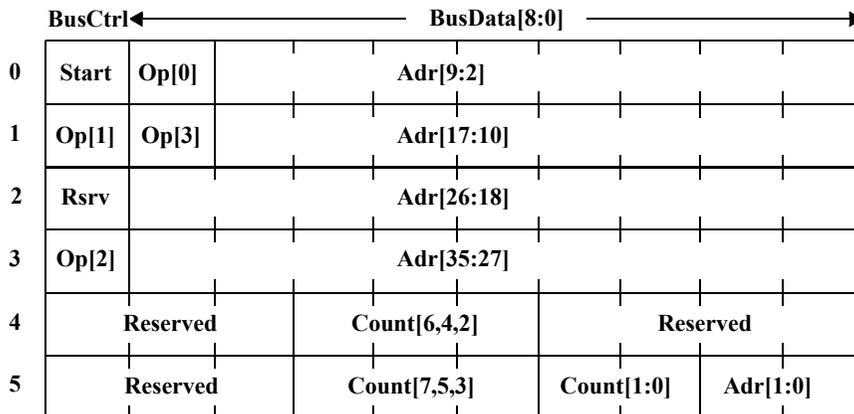


Figure 7: Request Packet Format



With the exception of broadcast writes, a transaction addresses only a single slave. Thus, slaves never arbitrate for use of the Rambus Channel.

Data Packet

Data packets contain 1 to 256 9-bit data bytes. Figure 9 depicts the format of a data packet.

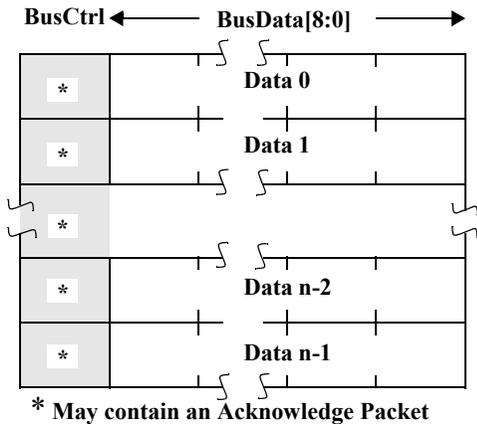


Figure 9: Data Packet Format

Read Transaction

The format of a read transaction is shown in Figure 10. After a request packet is issued, an acknowledge packet is returned a time AckDelay later. If the acknowledge is Okay, the read data packet is returned a time ReadDelay

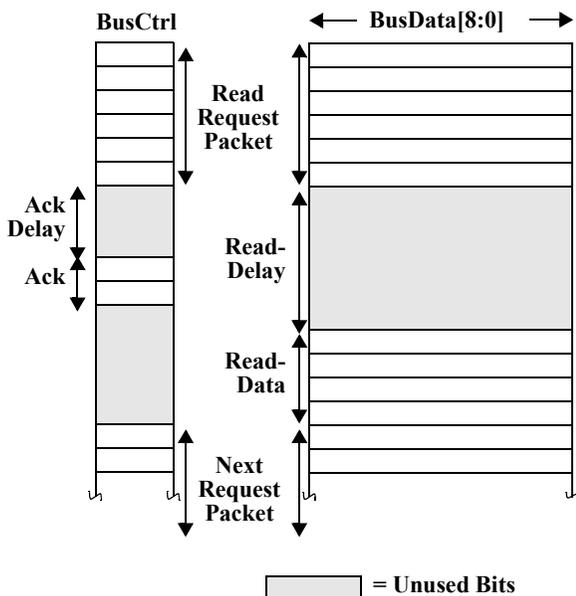


Figure 10: Read Transaction Format

after the request packet. Both of these delay values are programmed into the configuration registers of all devices during system initialization.

Write Transaction

A write transaction is shown in Figure 11. As in the read transaction, the acknowledge packet is returned a time AckDelay after the request packet. If the acknowledge is Okay, the write data packet is transferred a time WriteDelay after the request packet.

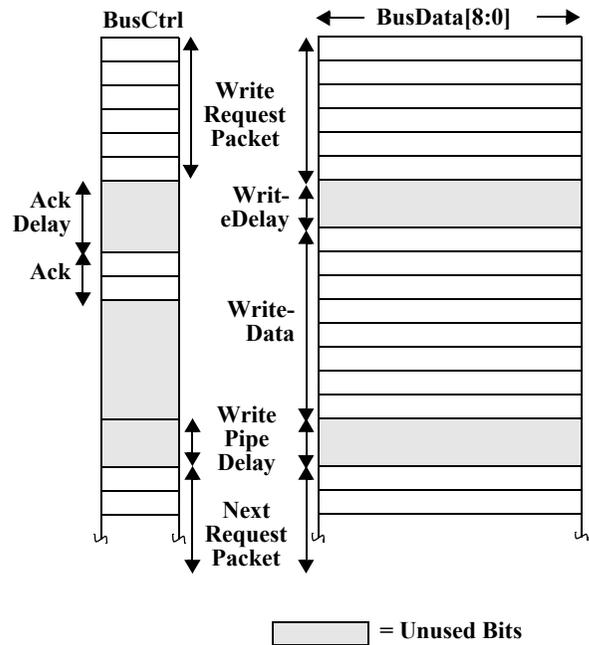


Figure 11: Write Transaction Format

Depending on the address and count values, a small delay may be required between the data packet of the current transaction and the request packet of the next transaction for write transactions. This delay is the WritePipeDelay. It allows the Rambus device time to finish the previous operation before beginning the next.