# Annex ### Mapping SPI onto GBB

The definition of mapping SPI messages into GBB messages is out of scope of this standard, since it is device dependent and should be implemented in control registers bits.

This annex specifies the usage of 1722 GBB when mapping SPI.

The SPI connection can have parallel configuration (using multiple “not Chip Select” (CSn) to select the active device) or daisy chain configuration (single CSn applies to all devices in chain). Here we only address single SPI device connection.

Every SPI CSn active period (CSn equals12) is called as one SPI transaction.

Most of SPI devices treat SPI transactions as sequences of 8 bit Bus Cycles (instructions and data are all 8 bits base (in byte unit)). Here we only define 8 bit wideSPI Bus Cycles.

## SPI Bus Definitions

An example of a SPI write transaction is shown in Figure 1. For SPI, the Bus Cycle is instruction, determining if the entire transaction is a write (w) or read (r) transaction. Then, followed by an optional 2nd Bus Cycle address, followed by an arbitrary number of data Bus Cycles. Some devices may have instruction and address in 1st byte only. Some other devices may have more than 3 address bytes. This definition allows for a flexible number of instruction and/or address Bus Cycles.

A typical SPI read transaction with one Bus Cycle instruction and one Bus Cycle Address as an example is shown in Figure 2.

The instruction in SPI can be only read or write. There is no mix of read and write in one SPI transaction. The data write can be 1 byte to many bytes, the data read can also be 1 byte to many bytes depends on type of instruction.

Figure 1. SPI Bus Write Transaction

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bus Cycle |  |  |  |  | **0** | **1** | **2** | **3** | **...** | **n+1** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CSn |  | Passive | Active | Passive |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SDI |  |  |  |  | Instruction (w) | Address | Data 0 | Data 1 | ... | Data n | Idle |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SDO |  | Hi-Z |

Figure 2. SPI Bus Read Transaction

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bus Cycle |  |  |  |  | **0** | **1** | **2** | **3** | **...** | **n+1** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CSn |  | Passive | Active | Passive |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SDI |  |  |  |  | Instruction (r) | Address | Hi-Z |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SDO |  |  |  |  |  |  |  |  |  |  |  |  | Data 0 | Data 1 | ... | Data n | Idle |

A SPI read or write transaction consists of all the data transfer required to read or write out a block of data from/to the addressed SPI slave device. A SPI read or write transaction has CSn active (12 ) for one time only.

The SPI mapping definition only contains read data observed from MISO, no SPI bus specific error definition exists. Some SPI devices may have error status codes indicated inside their response messages, which are handled transparently.

The message in section 1.2 below refers to “Byte Message Info” of ACF header.

## Use of GBB message

The ACF\_BYTE\_BUS and ACF\_BYTE\_BUS\_BRIEF message will be simply referred to as message in the following .

A SPI write transaction or read transaction may or may not fit into one AVTPDU depending on the maximum frame size permitted in the system.

A SPI write transaction consists of a first message which includes as many byte\_msg\_payload quadlets required to describe one or more Bus Cycles with a write instruction, one or more SPI Bus Cycles for address and one or more SPI Bus Cycles of data as defined in clause 1.1. Typically, a short write transaction can be expressed by a single message, but a long write transaction may be split across multiple 1722 AVTPDUs. For multiple 1722 AVTPDU segment, the host need control the packets flow speed not to overflow 1722 slave SPI write speed.

A SPI read transaction consists of one AVTPDU header with three or more ACF messages. The 1st message contains as many byte\_msg\_payload quadlets as required to describe the SPI Bus Cycles for instruction write and address. The second message contains the information to specify how many bytes to read and no byte\_msg\_payload. The third message will then contain the respectively read response, indicated by the rsp (response) field set to one (12), as well as many byte\_msg\_payload quadlets as required to describe the read SPI Bus Cycles.

For read transactions exceeding the system’s maximum frame size, the transaction may be split-up into multiple read requests below the system’s frame size limitation.

## Examples

A generic ACF GBB packet. For acf\_msg\_type = 0x0E, message\_timestamp will be omitted.



### Short 1722 SPI write Transaction

The example below is the packets sent for Figure 1 with n=6.

Request Packet



### Long 1722 SPI Write Transaction

The example is the packets sent for Figure 1 with n=2500.

Request Packets



For a data overflow example.

As above long write’s segment\_num == 0x003 hit overflow, the segment will be discarded by 1722 target and the target will response with below 1722 frame. When host receives it, it needs resend the same segment.

Response packet



### Short 1722 Read Transaction

The example is the packets sent for Figure 2 with n=6.

Request Packet



Response Packet



### Long SPI Read Transaction

The example is the packets sent for Figure 2 with n=2099.

Request Packet



Response Packets

