# Appendix A – Slow Control Packet Structure

The G-2 Interface module uses a simple packet structure for the slow control communication path between the Interface module and a host computer. The host computer communicates to the Interface module over TCP/IP. The structure shown below is the data payload sent by both the host and the G-2 Interface module. The orange region is a fixed size header that is included in all transactions. The green region is a variable size data portion that is only included when necessary based on the Command word in the header. There is a 256 word maximum for the data portion of a slow control packet (equal to the constant MAX\_CTRL\_DATA). Each header field and data word is a 32-bit integer.

|  |  |  |
| --- | --- | --- |
| WORD | FIELD | DESCRIPTION |
| 0 | LENGTH | Length(31:0) |
| 1 | ADDR | Address(31:0) |
| 2 | CMD | Command(31:0) |
| 3 | SIZE | Size(31:0) |
| 4 | STAT | Status(31:0) |
| 5 | DATA0 | Data Word 0 (31:0) |
| … | … | … |
| n+5 | DATAn | Data Word n (31:0) |
| n < 256 | | |

The slow control protocol is a master/slave interface controlled by the host computer. The host sends a command to the G-2 Interface and waits for a response. The G-2 Interface never initiates a slow control transaction but it always sends something in response to every command.

When the G-2 Interface module receives a packet, it inspects the header for a valid combination of Address, Command, and Size. (The host always sends a Status word of zero.) If the header is invalid, the G-2 Interface module will respond with a packet consisting of a copy of the received header with the Status word changed to one of the error values defined later in this document. The G-2 Interface module does not attempt any processing of the packet if the header is invalid.

If the G-2 Interface module receives a valid header, it performs the requested operation and returns a response packet. The response contains a copy of received header along with an appended data portion, if necessary.

## A.1 Slow Control Addresses

The following table summarizes the legal ranges of addresses that the G-2 Interface will respond to the command. All addresses point to 32-bit values within the G-2 Interface. Therefore, the lower 2 bits of the address must always be zero to avoid alignment issues. The G-2 Interface treats unaligned addresses as errors.

|  |  |  |
| --- | --- | --- |
| Address Range | Device | Valid Commands |
| 0x00000000 - 0x0FFFFFFC | ARM Processor | 0 – 7 |
| 0x10000000 - 0x10FFFFFC | Reserved | N/A |
| 0x20000000 - 0x20FFFFFC | Configuration Flash Memory | 0, 1, 4, 8 – 12 |
| 0x30000000 - 0x30FFFFFC | G-2 Interface Firmware Flash Memory | 0, 1, 4, 8 – 12 |
| 0x40000000 - 0x7FFFFFFC | Zynq FPGA | 0 – 7 |
| 0x80000000 - 0xBFFFFFFC | Reserved | N/A |

## A.2 Slow Control Commands

The following table summarizes the legal combinations of commands and packet formats for the G-2 Interface. For each command, it shows the number of header and data words sent between the host computer and G-2 Interface. All slow control transactions use 32-bit data words.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Host Packet | | | Interface Module Response | | CMD Value | ADDR Alignment |
| CMD Field | Header  Words | Data  Words | Header  Words | Data  Words |
| cmdRead | 5 | 0 | 5 | 1 | 0 | 4 Byte |
| cmdReadMask | 5 | 1 | 5 | 1 | 1 | 4 Byte |
| cmdWrite | 5 | 1 | 5 | 0 | 2 | 4 Byte |
| cmdWriteMask | 5 | 2 | 5 | 0 | 3 | 4 Byte |
| cmdArrayRead | 5 | 0 | 5 | SIZE | 4 | 4 Byte |
| cmdArrayWrite | 5 | SIZE | 5 | 0 | 5 | 4 Byte |
| cmdFifoRead | 5 | 0 | 5 | SIZE | 6 | 4 Byte |
| cmdFifoWrite | 5 | SIZE | 5 | 0 | 7 | 4 Byte |
| cmdNVWrite | 5 | 1 | 5 | 0 | 8 | 1 Byte |
| cmdNVArrayWrite | 5 | SIZE\* | 5 | 0 | 9 | 1 Byte |
| cmdNVEraseSector | 5 | 0 | 5 | 0 | 10 | 4 KByte |
| cmdNVEraseBlock | 5 | 0 | 5 | 0 | 11 | 64 KByte |
| cmdNVEraseChip | 5 | 0 | 5 | 0 | 12 | 16 MByte |
| \*For NVArrayWrite SIZE cannot be more than 64. For all others may be up to 256. | | | | | | |

The individual commands are described next:

* cmdRead - This operation reads a single 32-bit value from the address in the ADDR field. The SIZE field should always equal 1.
* cmdReadMask - This operation performs a mask read of a single 32-bit value from the address in the ADDR field. Only those bits that are set in the MASK are returned. All other bits will read zero. The SIZE field should always equal 1. The MASK is sent as DATA0.
* cmdWrite - This operation writes a single 32-bit value to the address in the ADDR field. The SIZE field should always equal 1.
* cmdWriteMask - This operation modifies the 32-bit value at the address in the ADDR field using a MASK and VALUE. Only those bits that are set in the MASK are modified to match the corresponding bits in the VALUE. All other bits are not modified. The SIZE field should always equal 1. The MASK is sent as DATA0. The VALUE is sent as DATA1.
* cmdArrayRead - This operation reads a series of 32-bit values starting from the address in the ADDR field. The SIZE field must be less than or equal to 256.
* cmdArrayWrite - This operation writes a series of 32-bit values starting at the address in the ADDR field. The SIZE field must be less than or equal to 256.
* cmdFifoRead - This operation repeatedly reads a series of 32-bit values from the address in the ADDR field. ADDR must correspond to a FIFO access register or the return packet will simply contain SIZE copies of the same value. The SIZE field must be less than or equal to 256.
* cmdFifoWrite - This operation repeatedly writes a series of 32-bit values to the address in the ADDR field. ADDR must correspond to a FIFO access register or the SSP's behavior will be undefined. The address will eventually acquire the final data value in the packet. The SIZE field must be less than or equal to 256.
* cmdNVWrite - This operation writes a single 32-bit value to the flash memory address in the ADDR field. The SIZE field should always equal 1. Only use for testing.
* cmdNVArrayWrite - This operation writes a series of 32-bit values starting at the flash memory address in the ADDR field. The SIZE field must be less than or equal to 64.
* cmdNVEraseSector - This erases 4,096 bytes of flash memory starting at the address in the ADDR field. The address must fall on a 4 KB boundary.
* cmdNVEraseBlock - This erases 65,536 bytes of flash memory starting at the address in the ADDR field. The address must fall on a 64 KB boundary.
* cmdNVEraseChip - This erases an entire region, 16,777,216 bytes. The provided address must be either 0x10000000, 0x20000000, or 0x30000000.