DAQ1000 DDL Protocol

May 18, 2005
Version 0.2
Tonko Ljubicic

This document provides the protocol on the DDL link between the TPC RDO and the TPC Sector Broker.

The protocol adheres to CERN's DDL general protocol mechanism and defines DAQ1000 specific meanings. For the names and acronyms of the DDL commands refer to the CERN-DDL manuals.

General Introduction

The SIU daughtercard used on the RDO enables a bidirectional but half-duplex communication with the D-RORC card in the Sector Broker PC. The half-duplex nature makes it slightly more complex since the "direction" of the link (RDO can send data vs. Sector Broker can send data) is an important factor during the communication.

The link direction is determined by the SIU and depends on the previous command received by the SIU. Note that the RDO *can not* change this direction by itself! For this reason the protocol is initiated by the PC master while the RDO slave only responds.

The power-up/reset condition of the SIU is to point the link towards the RDO (i.e. a write from the SB is allowed but not a write from the RDO to the PC). This is also the standard direction while the system is idle (i.e. not taking data). The opposite is true during the data taking mode when the link points to the SB.

The following DDL commands are supported, divided into two groups, ones that expect a response/return value from the RDO and the ones that don't. The "response" group turns the direction of the link to point toward the SB until the FESTW command is received by the D-RORC at which point the link is pointed back to the RDO.

No response ("write only")

FECTRL
STBWR **(framed by EOBTR)**
EOBTR **(DDL-specific use only)**

Response (“read/write”)

FESTRD	**(FESTW is the only response)**
STBRD	**(data followed by FESTWb)**
RDYRX	**(data followed by FESTWb)**

The response of the RDO comes in either the form of un-interpreted data or the FESTW command. The FESTW comes in two flavors: with and without the “End Of Block” marker differentiated by a single bit and, symbolically, by the letter “b” added to the command name.

All of these commands are 32 bit wide where most of the bits are used to encode the command itself as well as other DDL command features. All, however, leave 19 bits free which can be used to encode additional, STAR specific, meanings.

In the following, the commands are described by the name, mnemonic, DDL command, 19 bit value as well as the required state of the RDO (User or Factory) FPGA. Any value marked with “x” is currently reserved.

Simple Commands (small payload)

Write Only

RECONFIG **FECTLR** **0x0nnnp** **WO** **U/F**

Forces the reconfiguration of the RDO FPGA(s). “p” is the intended page of a possible multi-configuration configuration device. 0000 is the default, non-modifiable, so-called “factory” configuration, while any other number (1-15, typically 1) is the so-called “user” configuration.
“nnn” chooses which FPGA in case of multiples.

REBOOT **FECTLR** **0x11xxx** **WO** **U/F**

Reboots the NIOS CPU only.

CLEAR **FECTLR** **0x12xxx** **WO** **U/F**

Clears misc. counters (i.e. event/trigger related)

Read and Write

SET_ID	**FESTRD**	**0x0ssrr**	**WR**	**U/F**
reply	*FESTW*	*0xfssrr*		

Sets the ID of the RDO to ID "ss" (the sector number, 1-24) and sub-id "rr" (i.e. the RDO number 1-6).

GET_ID	**FESTRD**	**0x10000**	**RO**	**U/F**
reply	*FESTW*	*0xfssrr*		

Reads the Id/subid from the RDO. "ss" is the Id, "rr" is the subId. "f" indicates the factory configuration is set, user otherwise.

SEND_RCV_ C	**FESTRD**	**0x2V0cc**
reply	*FESTW*	*0x2V0cc*

Sends/Receives a single byte character "cc" which is valid only if V=1. Used to implement a console-like serial device.

Bulk Read & Write

Each bulk write or read uses STBWR for the write and STBWR followed by STBRD for a read. A STBWR before a STBRD is necessary to pass the start address and the length for the following write or any other longer set of parameters.

A STBWR "packet" contains an 8 byte header prepended to the data:

Word 0:	**number of bytes of the payload**	
Word 1:	**address**	
Word 2…:	**payload**	**(or none in case of a read)**

The command parameter codes the required read/write width in its last nibble (W):
0 for 32 bits,
1 for 8 bits &
2 for 16 bits.
The first nibble (T) gives the type of the transaction. Second nibble (S) is the subtype. Any reserved bits may code additional information. Thus:

WRITE(generic)	**STBWR**	**0xTSx0W**	
Ends with	*EOBTR*	*0xTSx0W*	
READ(generic)	**STBWR**	**0xTSx1W**	*passes the bytes & address*
Ends with	*EOBTR*	*0xTSx1W*	
	STBRD	**0xTSxxW**	*initiates the read*
Reply	*FESTWb*	*0x1TSxE*	*where E=0 for OK, otherwise error*

Type	Content
0	Generic
1	Altera EPC Flash (or other parallel FLASH device)
2	JTAG
3	ALTRO (or any other FEE)
4	Run Control
5	Logging/monitoring
6,7	RESERVED

Generic Read/Write

Read/Write bytes from/to address

Flash

Read/Write FLASH memory

JTAG

S=0 Read/Write chain/device/content
S=1 Scan JTAG chain(s)

FEE Specific

S=0 Read/Write Pedestals
S=1 Read/Write Bad pads

Run Control

Read/Write CONFIG

Logging/Monitoring Commands

S=0 Read Log
S=1 Read Monitoring
S=2 Read Debugging

Start/Stop Run

The following commands have meaning both for the STAR Run Control as well as the DDL protocol:

START_RUN	**RDYRX**	**0x00000**	**WO**
CONT_RUN	**RDYRX**	**0x00001**	**WO**
STOP_RUN	**EOBTR**	**0x00000**	**WO**
PAUSE_RUN	**EOBTR**	**0x00001**	**WO**

RDO Event Termination

Each event sent from the RDO is framed (terminated) by a FESTWb command.
The parameters of the FESTWb reply determine the type of event. The last nibble (S) is the status return with S=0 meaning OK.

EOE_PHYS	**FESTWb**	**0x0xxxS**

The event marks a “Physics” i.e. frontend readout event.

Apart from a frontend-related event the RDO may send any other event type interspersed with the frontend events. These are the same as replies of the STBRD command in previous paragraphs, i.e.:

EOE_LOG	**FESTWb**	**0x150xS**
EOE_MON	**FESTWb**	**0x151xS**
EOE_STRING	**FESTWb**	**0x152xS**
Etc.		