C.E.R.N., EP Division, AID Group

RMKI, RFFO Division, Detector Building Group

ALICE DETECTOR DATA LINK

ALICE-DDL

Front-end Electronics Emulator Interface Card

User Requirement Document and Technical Specification

Issue: Draft
Revision: 3.1

Reference: ALICE Detector Data Link FEMU-IC
Created: 04/05/98
Last modified: 23/08/06

Edited by: Csaba Soós

1 Document Information

1.1 Abstract

This document specifies the user requirements and the technical specification of the Front-end Electronics Emulator Interface Card.

1.2 Document Status Sheet

1. Document Title: ALICE/98-XX			
2. Document Reference Number: ALICE Detector Data Link FEMU-IC			
3. Issue	**4. Revision**	**5. Data**	**6. Reason for change**
Draft	1	5 April, 1998	Original document
Draft	2	6 May, 1998	Changes are based on further requirements
Draft	3	12 August, 2003	Update features
Draft	3.1	23 August, 2006	Update Table 8.

Table 1 Document Status Sheet

1.3 Document Change Record

Document Change Record		**DCR No.**	
		Date	
		Originator	
		Approved by	
1. Document Title			
2. Document Reference Number		ALICE Detector Data Link FEMU-IC	
3. Document Issue/Revision Number		Draft/**3.1**	
4. Page	**5. Paragraph**	**6. Reason for Change**	

Table 2 Document Change Record (of changes made since issue ...)

1.4 Table of Contents

1 DOCUMENT INFORMATION 3

1.1 ABSTRACT 3
1.2 DOCUMENT STATUS SHEET 3
1.3 DOCUMENT CHANGE RECORD 3
1.4 TABLE OF CONTENTS 4

2 INTRODUCTION 5

2.1 PURPOSE OF THE DOCUMENT 5
2.2 SCOPE OF THE DOCUMENT 5
2.3 ACRONYMS 5
2.4 ABBREVIATIONS 5
2.5 REFERENCES 6
2.6 OVERVIEW OF THE DOCUMENT 6

3 FRONT-END ELECTRONICS EMULATOR 7

3.1 GENERAL REQUIREMENTS 7
3.2 ARCHITECTURE 8
3.3 SPECIFIC REQUIREMENTS 9
3.3.1 FEE-SIU interface protocol emulation 9
3.3.2 DDL data communication monitor 10
3.3.3 Power supplies 10
3.3.4 Front-end clock (foCLK) 10

4 FRONT-END ELECTRONICS EMULATOR INTERFACE CARD 11

4.1 INTERFACES 11
4.1.1 The FEE-SIU interface 11
4.1.2 Interface to the HP16555A LA plug-in 11
4.1.3 Control interface 11
4.1.4 Clock in 12
4.1.5 Trigger in 12
4.2 FUNCTIONAL DESCRIPTION 13
4.2.1 Event data generation 13
4.2.2 Event data block length 15
4.2.3 Event data triggering 17
4.2.4 Flow control 18
4.2.5 Transfer enable 19
4.2.6 Internal memory 20
4.3 PHYSICAL ARRANGEMENT 21
4.4 NOTES: 22

2 Introduction

2.1 Purpose of the Document

This document has been written for the Front-end Electronics Emulator designers providing detailed information for the realization of the Front-end Electronics Emulator Interface Card.

2.2 Scope of the document

All FEEs of the ALICE sub-detectors will be connected to the DAQ through the DDL. Any FEE shall be designed at a later phase of the DDL project. Therefore, emulating FEE operating conditions is essential for testing the DDL.

The FEMU consist of two parts: the FEIC and the Logic Analyzer. The Logic Analyzer is connected to the FEE-SIU interface, monitoring signal flow.

2.3 Acronyms

DAQ	Data Acquisition System
DDL	ALICE Detector Data Link
FEE	Front-end Electronics
FEMU	Front-end Electronics Emulator
FEIC	Front-end Electronics Emulator Interface Card
FECMD	Front-end Command Word
FESTW	Front-end Status Word
TID	Transaction ID
HWG	Hardware Guide for the Front-end Designers
ICD	Interface Control Document
LA	Logic Analyzer
RORC	Read-out Receiver Card
SIU	DDL Source Interface Unit
TAP	Test Access Port

2.4 Abbreviations

EOBTR	Front-end Command:	Close any Block Transmission Transactions
EOB	Front-end Status Bit:	End of Data Block
FECMD	Front-end Command	
FECTRL#address	Front-end Command:	Front-end Control
FESTRD#address	Front-end Command:	Front-end Status Read-out
FESTW	Front-end Status Word	
STBRD#address	Front-end Command:	Open a Data Block Read Transaction
STBWR#address	Front-end command:	Open a Data Block Write Transaction
RDYRX	Front-end Command:	Open an Event Data Transmission Transaction

2.5 References

[1] Interface Control Document, ALICE/96-43, Internal Note/DAQ, 12 December 1996

[2] Physical and Signaling Interface Specification for the Fibre Channel Implementation of the DDL, ALICE/97-04, Internal Note/DAQ, 14 February 1997

[3] Hardware Guide for the Front-end Designers, ALICE/98-21, Internal Note/DAQ, 19 May 1998

2.6 Overview of the Document

The first chapter of this document contains:

1. the abstract
2. the document status sheet
3. the document change record
4. the table of contents

The second chapter of this document contains:

1. the purpose of the document
2. the scope
3. the acronyms
4. the abbreviations
5. the references
6. an overview of the document

The third chapter of this document describes:

1. the general requirements of the FEMU
2. the FEMU architecture
3. the specific requirements of the FEMU

The fifth chapter of this document describes:

1. the function description of the FEIC

3 Front-end Electronics Emulator

3.1 General Requirements

The Front End Electronic Emulator shall provide real time emulation of the ALICE FEEs, according to the DDL protocol [1] and timing [3]. It shall execute the following transactions:

- front-end control;
- front-end status read-out;
- event data transmission;
- data block downloading;
- data block read

The FEIC is one of the components of the FEMU. It shall provide the following main functions:

- generates test data blocks with different lengths and patterns and transfers them to the SIU with flow control;
- transfers downloaded data from the internal memory to the SIU with flow control;
- transfers data blocks from the SIU to the general purpose internal memory;
- receives FECMDs from the SIU and sends FESTWs back, if it is requested;
- connect FEE-SIU all the interface signals to the logic analyzer (LA);
- applies flow control for the SIU for the interruption of the incoming data block transfer;
- generates trigger for starting the data block transfers from the FEMU to the SIU either internally or from external sources;
- generates FEE system clock for the SIU either internally or from external sources;
- supplies power for the SIU

3.2 Architecture

The FEMU consists of the following building blocks (see Figure 1):

- Front-end Electronic Emulator Interface Card (FEIC);
- Logic Analyzer (LA);

The connection between the FEIC and the HP16555A LA plug-in is described in chapter 4.1.2.

The following external components also can be connected to the FEMU:

- Trigger source;
- External clock source;
- Power supply;

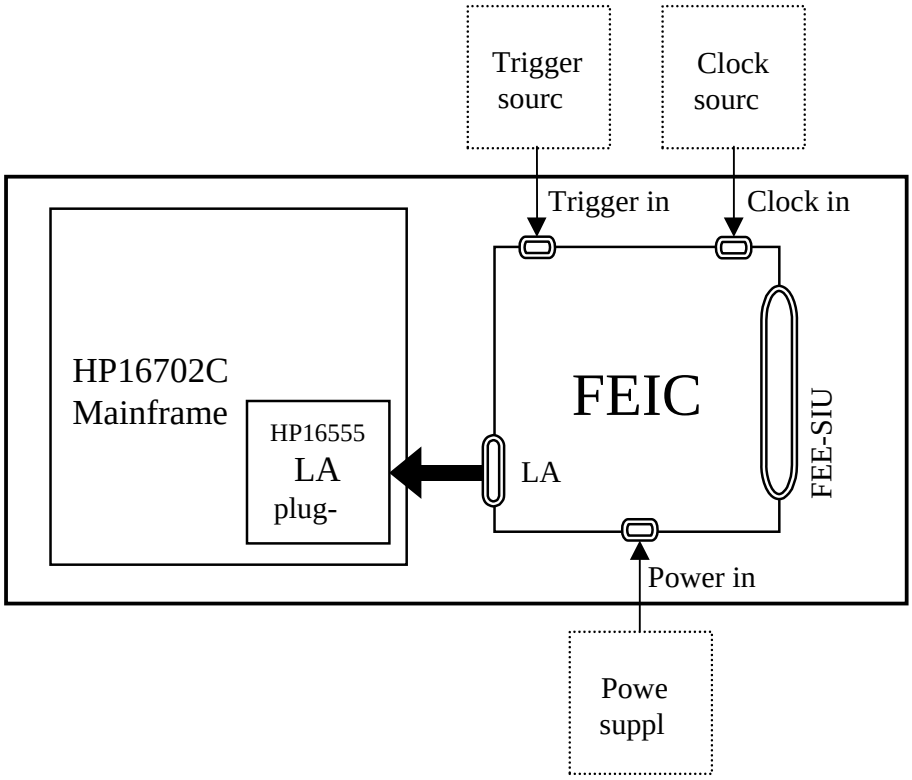

Figure 1 FEMU Architecture

3.3 Specific Requirements

3.3.1 FEE-SIU interface protocol emulation

Test data block generation and transfer

1. The user shall be able to select between the test data block generation and transfer. This selection can be done either by setting switches located on the FEIC or externally by sending a FECMD through the DDL.
2. The FEIC shall generate test data blocks for the SIU with different length. The length can be set either by switches located on the FEIC or externally by sending a FECMD through the DDL. The user shall be able to set the block length within the range of 16 words - 256k words (64 byte – 1M byte) with the following steps: 16, 32, 64, 128, 256, 512, 1k, 2k, 4k, 8k, 16k, 32k, 64k, 128k, 256k.
3. The FEIC shall generate test data blocks with the following different patterns:
   - alternating pattern 0000⇔1111
   - flying ones 0001⇒0010⇒0100⇒1000
   - flying zeros 1110⇒1101⇒1011⇒0111
   - 19 bit binary up counter 0000⇒0001⇒0010⇒0011
   - 19 bit binary down counter 1111⇒1110⇒1101⇒1100

Data transfer from the internal memory

The FEIC shall transfer data words from the internal memory to the SIU by using the data block read transaction. The contents of the data memory can be downloaded previously by using the data block write transaction.

Status generation and transfer

The FEIC shall be able to receive FECMDs from the SIU, which are transferred from the RORC to control the working of the FEIC. If it is necessary the FEIC shall send a FESTW back.

Event data block triggering

The FEIC shall start the event data block transfer to the SIU, when it receives a trigger. The triggering mode can be selected either by switches located on the FEIC or externally by sending a FECMD through the DDL.

Internal event trigger shall be generated:

- with gap of 16 foCLK cycle between the events;
- with gap of 128 foCLK cycle between the events;
- periodically after each 10 milliseconds;
- periodically after each 100 milliseconds;

The FEIC must provide external event data trigger input (see Figure 1) for external trigger circuit asynchronous to the FEIC.

Event data triggering can be initiated manually by push button.

Flow control generation

The FEIC shall be able periodically interrupt the data block transfer from the SIU to the FEIC for one foCLK period by activating the foBSY_N signal after each:

- single received word;
- 128 received words;

- 16k received words

The periodic interrupt period can be selected by switches located on the FEIC or externally by sending a FECMD through the DDL.

End of Data Block identification

The FEIC shall be able to generate a FESTW(EOB=1) at the end of each test data block transferred from the FEIC to the SIU. The data block transfer is used in the event data transmission, as well as in the data block read transaction [1].

3.3.2 DDL data communication monitor

The FEMU shall be able to monitor the running transactions on the FEE-SIU interface. For this purpose the FEIC shall connect all the interface signals to the LA port.

3.3.3 Power supplies

The FEIC shall provide +3.3V and +5V power supplies for the SIU from external sources.

3.3.4 Front-end clock (foCLK)

The FEIC shall provide front-end clock signal (foCLK) for the SIU. This clock signal can be sourced either from an external, or an internal clock source.

4 Front-end Electronics Emulator Interface Card

4.1 Interfaces

The FEIC is one of the components of the FEMU (see Figure 1). The FEIC shall fully be compatible with the FEE-SIU interface protocol, as described in [1]. The FEIC shall provide interfaces between the SIU and the HP16555A LA plug-in.

4.1.1 The FEE-SIU interface

The FEMU shall completely emulate the working of a FEE. For this purpose the FEIC must fulfill the requirements of the FEE-SIU interface [1].

The FEMU shall execute transactions, shown in the general requirements. The details of these transactions are discussed in [1].

4.1.2 Interface to the HP16555A LA plug-in

The FEIC shall connect all the lines of the FEE-SIU interface to the LA for monitoring the transactions. The FEIC has a HP 16555A LA compatible interface, the LA port. The allocation of the signals on each POD can be seen in the Table 3.

Labels	POD CLOCK	POD 3	POD 2	POD 1
Byte 0				"--------********"
Byte 1				"********--------"
Byte 2			"--------********"	
Byte 3			"********--------"	
foCLK	"---*"			
fbCTRL#		"---------------*"		
fbTEN#		"--------------*-"		
fiLF#		"-------------*--"		
fiDIR		"------------*---"		
foBSY#		"-----------*----"		
fiBEN#		"----------*-----"		

Table 3 Logic analyzer line assignments

As shown in Table 3, the 32-bit wide data bus is assigned to the POD 1 and POD 2 of the LA, and the control signals are connected to the POD3. The foCLK signal is used, as a clock signal to the LA, and must be connected to one line of the POD CLOCK of the LA.

4.1.3 Control interface

Switches

The FEMU can be controlled manually by using the following setting switch arrays (see the Physical Arrangement section later in this document):

- Internal pattern generator ***PS3-PS0***;
- Event data block length ***BL3-BL0***;
- Flow control mode ***FC1-FC0***;
- Transfer enable mode ***TE1-TE0***;
- Event data trigger mode ***DT2-DT0***;

The meaning of these switch arrays are discussed later in this document.

The FEMU shall always provide status information about selected mode. The status can be read out from the dedicated status registers by sending a FESTRD#address command to the

FEMU [1].

LEDs

The FEMU shall display status information of different working conditions with the following LEDs:

- Power ***LED0*** (colour);
- Internal pattern generator status ***PSLED3-PSLED0*** (colour);
- Event data block length status ***BLLED3-BLLED0*** (colour);
- Flow control mode status ***FCLED1-FCLED0*** (colour);
- Transfer enable mode status ***TELED1-TELED0*** (colour);
- Event data trigger mode status ***DTLED2-DTLED0*** (colour);

The ***LED0 power-on LED*** shall be switched on, when the FEIC is powered by the external power supply.

The status LEDs shall display the status of the FEMU, whether it is controlled manually, or via front-end control transaction.

4.1.4 Clock in

The FEIC shall operate from external clock source. This external clock source is used for test purpose only. The clock signal must fulfill the following requirements:

- 3.3V TTL level;
- maximum clock frequency: 40MHz;
- clock symmetry (duty cycle) : 40 – 60%;
- rise time: 1ns;
- fall time: 1ns;
- input impedance: 50Ω;

4.1.5 Trigger in

The FEIC shall start the event data transmission transaction when trigger occurs. The trigger shall be generated internally, or from an external trigger source. For this purpose the FEIC has a Trigger in port, which is capable of receiving asynchronous trigger signal. The signal of the external trigger source must fulfill the following requirements:

- 3.3V TTL level, active low;
- minimum pulse width must be at least two foCLK period;
- input impedance: 50Ω;

4.2 Functional description

4.2.1 Event data generation

The FEIC shall transmit event data blocks from the internal function generator engine. Different data block lengths and pattern types of the internal pattern generator can be set manually or by the RORC using a FECTRL command.

Internal pattern generator setting switches

Internal pattern generator functions can be set by switches PS3-PS0 (see Table 4). Actual operating mode is displayed by LED string PSLED3-PSLED0.

PS3	PS2	PS1	PS0	Pattern type / function
0	0	0	0	Internal pattern generator is programmed by the RORC.
0	0	0	1	Reserved
0	0	1	0	Internal pattern generator: alternating 32-bit patterns, i.e. 00000000 ⇔ FFFFFFFF
0	0	1	1	Internal pattern generator: flying ‘0’s, i.e. FFFFFFFE⇒FFFFFFFD⇒FFFFFFFB⇒ etc
0	1	0	0	Internal pattern generator: flying ‘1’s, i.e. 00000001⇒00000002⇒00000004⇒etc
0	1	0	1	Internal pattern generator: 19 bit incrementing counter starting from 0
0	1	1	0	Internal pattern generator: 19 bit decrementing counter starting from 0
0	1	1	1	Reserved
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

Table 4 Internal pattern generator setting switches

Setting internal pattern generator status by a FECMD

Internal pattern generator can be programmed using a front-end command, when PS3-PS0 switches are all set to ‘0’. In this case, the pattern generator status will be displayed by PSLED3-PSLED0.

In order to setup the internal pattern generator mode from the software, a FECTRL command must be sent from the RORC to the FEIC. The parameter field of the command should contain the pattern generator mode bits (D15-D12) and the address of the register (D25-D20) as shown in Table 5. Pattern generator mode bits are identical to PS3-PS0 settings as shown in Table 4.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000000	res.	res.	res.	res.	PS3	PS2	PS1	PS0	TID	0xC4

Table 5 FECTRL command for setting internal pattern generator

Reading out internal pattern generator status by a FESTW

For reading out internal pattern generator mode a FESTRD#0x0 command must be sent from the RORC to the FEIC. The status word contains the pattern generator mode bits in position D15-D12 as shown in Table 6.

D31	D30-D16	D15	D14	D13	D12	D11-D8	D7-D0
Err./OK	-	PS3	PS2	PS1	PS0	TID	0x44

Table 6 FESTW containing internal pattern generator setting

4.2.2 Event data block length

The FEIC shall transfer event data word blocks to the SIU according to the FEE-SIU data transfer protocol described in [1]. Data block length can vary from 16 words to 256k words.

Event data block length setting switches

Event data block length can be set by using BL3-BL0 switches (see Table 7). Please note that Table 7 contains values that are beyond the range, which can be set using the switches. In order to set block length values bigger than 256k words, the soft setting mode will have to be used (see the next paragraph). Actual event data block length is displayed by LED string BLLED4-BLLED1.

BL4*	BL3	BL2	BL1	BL0	Event data block length
0	0	0	0	0	Event data block length is set by the RORC
0	0	0	0	1	16 words
0	0	0	1	0	32 words
0	0	0	1	1	64 words
0	0	1	0	0	128 words
0	0	1	0	1	256 words
0	0	1	1	0	512 words
0	0	1	1	1	1k words
0	1	0	0	0	2k words
0	1	0	0	1	4k words
0	1	0	1	0	8k words
0	1	0	1	1	16k words
0	1	1	0	0	32k words
0	1	1	0	1	64k words
0	1	1	1	0	128k words
0	1	1	1	1	256k words
1	0	0	0	0	512k words
1	0	0	0	1	1M words
1	0	0	1	0	2M words
1	0	0	1	1	4M words
1	0	1	0	0	8M words
1	0	1	0	1	16M words
1	0	1	1	0	32M words
1	0	1	1	1	64M words
1	1	0	0	0	128Mwords
1	1	0	0	1	256M words

Table 7 Event data block length setting switches

Setting event data block length by a FECMD

Event data block length can be programmed using a command, when BL3-BL0 switches are all set to '0'. In this case, the actual event data block length setting will be displayed by BLLED3-BLLED0.

In order to setup the event data block length from the software, a FECTRL command must be sent from the RORC to the FEIC. The parameter field of the command should contain the event data block length bits (D19-D12) and the address of the register (D25-D20) as shown in Table 8. Block length bits are identical to BL4-BL0 settings as shown in Table 7.

* BL4 cannot be set using switches; use a FECMD instead to set this bit.

When using the soft setting mode, the user can choose between fixed and random sized block generation. If bit D19 is set, the FEIC will produce data blocks of different sizes from a range between 1 and a maximum value, which is defined by BL4-BL0.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000001	RND	res.	res.	BL4	BL3	BL2	BL1	BL0	TID	0xC4

Table 8 FECTRL command for setting event data block length

The pseudo random generator can optionally be initialized to a given number between 0 and 255 (S7-S0) using a command as shown in Table 9.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000101	S7	S6	S5	S4	S3	S2	S1	S0	TID	0xC4

Table 9 FECTRL command to initialize the pseudo random generator

Reading out event data block length by a FESTW

For reading out internal pattern generator mode a FESTRD#0x100 command must be sent from the RORC to the FEIC. The status word contains event data block length bits in position D15-D12 as shown in Table 10.

D31	D30-D17	D16	D15	D14	D13	D12	D11-D8	D7-D0
Err./OK	-	RND	BL3	BL2	BL1	BL0	TID	0x44

Table 10 FESTW containing event data block length setting

4.2.3 Event data triggering

Event data blocks can be generated periodically or triggered by external source (push button or event data trigger input). Trigger mode can be set by either Data Trigger switches, DT1-DT0 or by the RORC using a FECTRL command.

Event data trigger mode switches

Event data trigger mode can be set by switches DT2-DT0 (see Table 11). Actual event data trigger mode is displayed by LED string DTLED2-DTLED0.

DT2	**DT1**	**DT0**	**Event Data Trigger Mode**
0	0	0	Event Data Trigger Mode is set by the RORC
0	0	1	Event Data Triggering by external pushbutton
0	1	0	Event Data Triggering by Trigger Input
0	1	1	Event Data Triggering with 16 CLK gap between blocks
1	0	0	Event Data Triggering with 128 CLK gap between blocks
1	0	1	Event Data Triggering in every 10 millisecond
1	1	0	Event Data Triggering in every 100 millisecond
1	1	1	Reserved

Table 11 Event data trigger mode switches

Setup event data trigger mode by a FECMD

Event data trigger mode can be programmed using a command, when DT2-DT0 switches are all set to '0'. In this case, the actual event data trigger mode setting will be displayed by DTLED2-DTLED0.

In order to setup the event data trigger mode from the software, a FECTRL command must be sent from the RORC to the FEIC. The parameter field of the command should contain the event data trigger bits (D14-D12) and the address of the register (D25-D20) as shown in Table 12. Event data trigger mode bits are identical to DT2-DT0 settings as shown in Table 11.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000010	res.	res.	res.	res.	res.	DT2	DT1	DT0	TID	0xC4

Table 12 FECTRL command for setting event data trigger mode

Readout of event data trigger mode by a FESTW

For reading out FEMU event data trigger mode a FESTRD#0x200 command must be sent from the RORC to the FEIC. The status word contains event data trigger mode bits in position D14-D12 as shown in Table 13.

D31	D30-D15	D14	D13	D12	D11-D8	D7-D0
Err./OK	-	DT2	DT1	DT0	TID	0x44

Table 13 FESTW containing event data trigger mode bits

4.2.4 Flow control

Flow control settings allow emulating FEEs with slow internal memory. The FEMU shall set the foBSY# flow control signal to active low periodically, signaling that FEE is not able to receive more data words from the SIU. After 4 foCLK cycles, foBSY# is reset to inactive high.

Flow control mode switches

Flow control mode can be set by using FC1-FC0 switches (see Table 14). Actual flow control mode is displayed by LED string FCLED1-FCLED0.

FC1	**FC0**	**Flow Control Mode**
0	0	Flow Control Mode is set by the RORC
0	1	foBSY# activated after each received word
1	0	foBSY# activated after each 128 received words
1	1	foBSY# activated after each 16k words received

Table 14 Flow control mode switches

Setting flow control mode by a FECMD

Flow control mode can be programmed using a command, when FC1-FC0 switches are all set to '0'. In this case, the actual flow control mode setting will be displayed by FCLED1-FCLED0.

In order to setup the flow control mode from the software, a FECTRL command must be sent from the RORC to the FEIC. The parameter field of the command should contain the flow control mode bits (D13-D12) and the address of the register (D25-D20) as shown in Table 15. Flow control mode bits are identical to FC1-FC0 settings as shown in Table 14.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000011	res.	res.	res.	res.	res.	res.	FC1	FC0	TID	0xC4

Table 15 FECTRL command for setting flow control mode

Reading out flow control mode by a FESTW

For reading out the FEMU flow control mode a FESTRD#0x300 command must be sent from the RORC to the FEIC. The status word contains the flow control mode bits in position D13-D12 as shown in Table 16.

D31	D30-D14	D13	D12	D11-D8	D7-D0
Err./OK	-	FC1	FC0	TID	0x44

Table 16 FESTW containing flow control mode bits

4.2.5 Transfer enable

Transfer enable (fbTEN#) line is controlled by either the SIU or the FEE to enable data transfer on rising edge of foCLK. Event data block transfer from the FEE to the SIU can temporary be interrupted by the FEE setting fbTEN# active low, when the FEE is not ready for data transfer to the SIU. Periodic event data transfer interrupts can be emulated by the FEMU. According to Transfer Enable Mode switches (TE1-TE0) fbTEN# can be set active low after each, after each 128 or after each 16k words transferred event data words. fbTEN# is reset inactive high after 4 foCLK periods.

Transfer enable mode switches

Transfer enable mode can be set by using TE1-TE0 switches (see Table 17). Actual transfer enable mode is displayed by LED string TELED1-TELED0.

TE1	**TE0**	**Transfer Enable Mode**
0	0	Transfer Enable Mode is set by the RORC
0	1	fbTEN# deactivated after each received word
1	0	fbTEN# deactivated after each 128 received words
1	1	fbTEN# deactivated after each 16k words received

Table 17 Transfer enable mode switches

Setting transfer enable mode by a FECMD

Transfer enable mode can be programmed using a command, when TE1-TE0 transfer enable mode switches are all set to '0'. In this case, the actual transfer enable mode setting will be displayed by TELED1-TELED0.

In order to setup the transfer enable mode from the software, a FECTRL command must be sent from the RORC to the FEIC. The parameter field of the command should contain the transfer enable mode bits (D13-D12) and the address of the register (D25-D20) as shown in Table 18. Transfer enable mode bits are identical to TE1-TE0 settings as shown in Table 17.

D30-D26	D25-D20	D19-D12								D11-D8	D3-D0
-	000100	res.	res.	res.	res.	res.	res.	TE1	TE0	TID	0xC4

Table 18 FECTRL command for setting transfer enable mode.

Reading out transfer enable mode by a FESTW

For reading out FEMU transfer enable mode a FESTRD#400 command must be sent from the RORC to the FEIC. The status word contains the transfer enable mode bits in position D20-D19 as shown in Table 19.

D31	D30-D14	D13	D12	D11-D8	D7-D0
Err./OK	-	TE1	TE0	TID	0x44

Table 19 FESTW containing transfer enable mode bits

4.2.6 Internal memory

The FEIC shall emulate FEEs with internal memory. For this purpose the FEIC has 256 word, i.e. 1kbyte data memory. The internal memory space is organized as a memory bank and starts at the 0x0 address.

Data block write

The data words can be downloaded into the internal memory by using the data block write transaction [1]. For starting the data block download a STBWR#0 command must be sent to the FEMU. If the number of the downloaded data words is more than the size of the internal memory, the FEIC shall store only the first 256 words of the transmitted data.

Data block read

The data words, downloaded previously, can be read out by using the data block read transaction [1]. To initiate the data block read transaction a STBRD#0 command must be sent to the FEMU. The FEMU shall send a FESTW(EOB) status word back after the last data word transmitted.

4.3 Physical arrangement

4.4 Notes: