The ATLAS Liquid Argon Calorimeters Read Out Driver (ROD)

The TMS320C6414 DSP Mezzanine board

Abstract : This document describes the ATLAS Liquid Argon Calorimeters Read Out Driver mezzanine, designed around the 720 MHz TMS320C6414 DSP from Texas Instrument and the EP1C6 Cyclone FPGA from Altera. The document is organized as follows : after the required specifications, the chosen architecture of the board is described, as well as the functionality of the main components. The document ends with technical realization aspects and the tests performed.

Author : Julie PRAST
Web Page : http://wwwlapp.in2p3.fr/atlas/Electronique/RODs/index.html
File name : pu6414cyclone_v29.doc
Created : September 2003.
Last update : September 29, 2005
Version : 2.9

1 Table of contents

2 Introduction

The ROD Processing Unit (PU) of the ATLAS Liquiq Argon Calorimeter is a 120*85 mm mezzanine board, that can be plugged on a 9U VME ROD motherboard [5]. It calculates the precise energy deposited in each calorimeter cell and the timing of these signals from discrete time samples, as well as a quality factor for the pulse shape. It applies an optimal filtering algorithm in order to minimize the pileup and electronic noise. It also performs monitoring and formats the results for the next element in the electronic chain.
Each ROD PU is equipped with two DSPs, each one processing the 128 channels of one FEB in normal mode and 256 channels (2 FEBs) in staging mode.
The architecture of the daughterboards is based on programmable components (FPGAs) and Digital Signal Processors(DSP), precisely around the TMS320C6414, the last DSP generation from Texas Instrument.

3 Specifications

3.1 The ROD Processing Unit goals

3.1.1 Physics Calculation

In normal mode, a single ROD processing Unit receives data from 2 FEBs, that is (typically) five digitized samples from 256 calorimeter cells. The board is in charge of calculating the energy and the time relative to the peak of the signal for each channel, along with a pulse quality factor, which indicates how closely the samples follow the expected waveform.

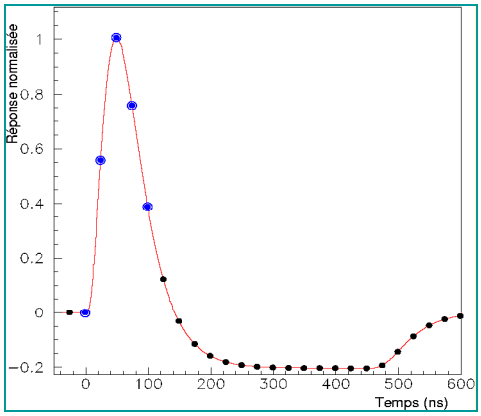

Figure 1 : The expected waveform at the input of the ROD modules

The algorithm implemented in the ROD to extract the energy and time for each channel, uses a technique called optimal filtering [1]. The idea is to estimate these quantities in an accurate and computationally efficient way, minimizing the background noise contributions. The energy (E) and time (T) are expressed as a weighted sum of the samples Si, as shown in the following formulas:

$$E = \sum a_i \,.(S_i - PED)$$

$$E.T = \sum b_i \,.(S_i - PED)$$

where i extends over all samples, PED is the pedestal value, and a_i and b_i are the optimal filtering weights.

The pulse quality factor is a normal chi squared calculation:

$$\chi^2 = \sum ((S_i - PED) - E.\, g_i)^2$$

where g_i is the expected normalized waveform for a given channel.

The error on the energy is amplitude independent, whereas the error on the time varies inversely with the amplitude. For this reason, it only makes sense to calculate T for those channels with E above some threshold value. For a given event, most of the cells have low energy, coming from background noise. There are few cells for which T, and χ^2 must be calculated. Simulations show that this fraction of high energy cells is less than 10 %.

As the raw data from the FEB is no longer available offline, the ROD module must perform monitoring of the calorimeter functioning by building histograms [2].

At LHC startup, non processed data, as well as processed data will be transferred to the Level 2.

During calibration runs [3], charges of various amplitudes are injected in the electronic chain. The ROD modules compute first and second moments and send data to a local processor, which will then calculate calibration constants (a_i, b_i in the formula) for each channel of the calorimeter.

3.1.2 Error Detections

The ROD processing unit aims at detecting errors in the incoming FEB data. Several points must be checked :

- Presence of the start of event
- Presence of the end of event.
- Parity for each word, except the start and end of event.
- Start of data alignment to check half-FEB (HFEB) synchronization. Functioning correctly, 2 HFEB of the same FEB, are synchronous. If they are not, this must be considered as an error.
- Identical gain for all samples of a given channel.
- Identical RADD, BCID and EVTID value within the 8 ADC of a same HFEB.
- Identical RADD, BCID and EVTID value between two HFEB.
- Value of the SCAC status.

The ROD processing unit must also check the synchronization between FEB data and TTC data, comparing the BCID and event ID values.

3.1.3 Requirements

The number of samples digitalized for each calorimeter cell is between 3 and 32. The most usual mode is 5 samples, which is a good compromise between the data flow at the FEB output and the precision obtained on the physics calculation.
FEB data is coded either on one ADC gain, which is the default mode, or on three gains in test mode.
The higher Level 1 trigger rate sustainable for events with 5 samples and 1 gain is 100 kHz. It corresponds to the maximum FEB throughput, when 5 samples of the calorimeter cells are digitalized. If the number of samples or gain increases, the average trigger rate seen at the input of the ROD will consequently decrease.

3.2 Interface with the Motherboard

3.2.1 Mechanical characteristics

The ROD PU is a 85*120 mm board that is plugged on the motherboard through 3 board-to-board connectors from AMP. The two “input” connectors are the 64 pins connectors AMP120-525-1 and the “output” connector is the 84 pins connector AMP120525-2.
Figure 2 shows the mezzanine board mechanical characteristics.

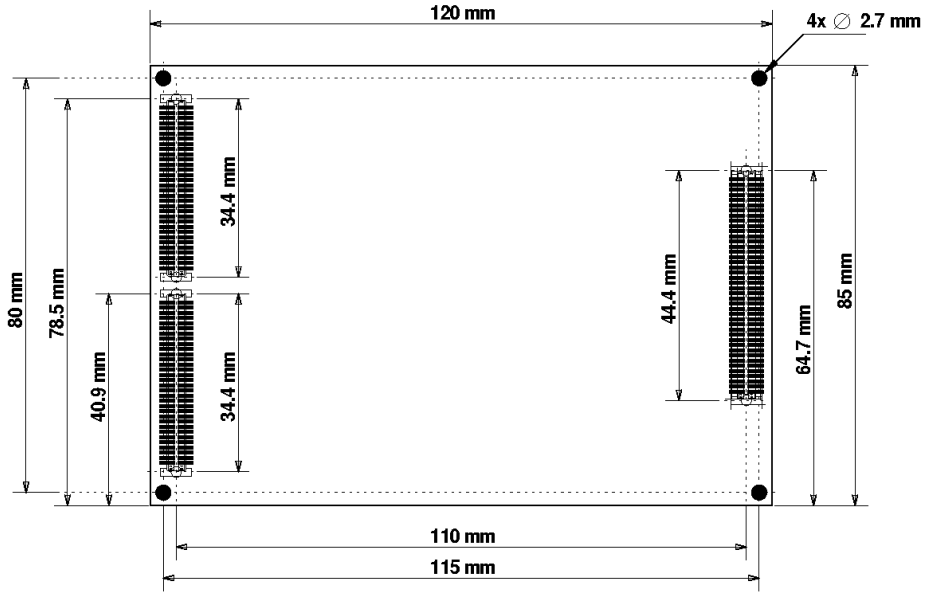

Figure 2 : The PU mechanical characteristics

To avoid mating problems, connector alignment constraints must be respected : The connector to connector spacing must be within a +/-.003" tolerance. The mated connector pairs also must be within +/-.003" of a theoretical centerline (they can be out of alignment by .006" max).

3.2.2 Power supply

The power supply of the mezzanine board is provided through the three connectors, via 31 ground pins and 34 3.3 V pins. The 3.3 V voltage is regulated from a 48V/3.3V 20A DC-DC regulator mounted on the motherboard. The maximum authorized current is 5 A for the whole PU.

3.2.3 FEB signals

A PU treats 2 FEB data in normal mode and 4 FEB data in staging mode. Data from each FEB (128 calorimeter cells, 16 ADC) arrives through a 17-bit wide bus. FEB data is synchronized on the motherboard, with a 80 MHz clock (TTCrx clock multiplied by 2). Each FEB is independent from one another, it means that the start of event of a FEB is not necessarily aligned with the start of event of another FEB.
Data arrives through a serial protocol, described in annex 1. The ADC readout event format (ie when parallelized) is described in annex 2.

3.2.4 TTC signals

TTC data arrives from the motherboard TTC FPGA on four signal lines, synchronized with a 40 MHz clock. Two signal lines encode BCID and event ID information, where as the two other lines encode the trigger type. The TTC communication protocol is described in the below figure:

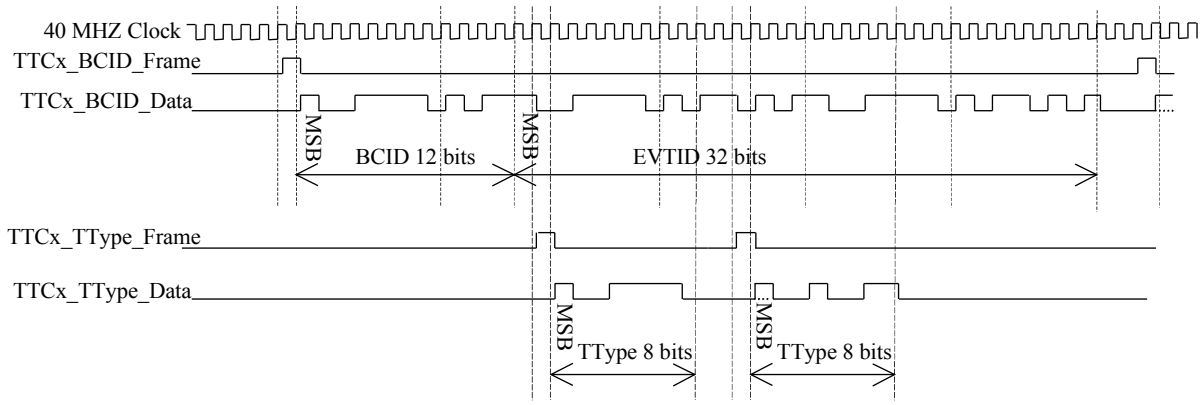

Figure 3 : TTC communication protocol

The protocol for BCID corresponds to a frame having 2 phases of different lengths, one for the BCID and the other for the Event ID each having one element. There is at least 2 clock cycles between each frame.
For the TType there is one frame with only one phase and one element. There are at least 2 clock cycles between each frame.

There is one trigger type, one BCID and event ID per FEB event. BCID and event ID are required to arrive before the complete FEB event in the DSP. The trigger type will arrive at least 4 us after the event ID and BCID.

3.2.5 VME interface

The VME interface is coded on six signal lines, synchronized by a 40 MHz clock and coming from the VME FPGA of the ROD motherboard. The protocol encoding the interface between the PU and the motherboard is described in reference [4].

3.2.6 Other signals

The interface between the motherboard and the PU also includes :

- Two 16-bit output FIFO data bus and the associated control signals.
- The JTAG signals
- Busy, interrupt and reset lines.

For more details about the interface with the motherboard, see reference [5].

3.3 The Staging mode

In normal mode, the ROD motherboard is equipped with 4 ROD Processing Units, each treating 2 FEB data. In some cases, the ROD motherboard can be equipped with only half of the PUs. This is called the staging mode, where the trigger rate is kept below 50 kHz. This is the reason why a data bus between staging FPGAs (32 bits at 80 MHz) has been introduced. Data from four G-link chips is routed through one staging FPGA to one PU board. Therefore, in staging mode the PU will process twice as many channels as in normal mode with all PUs. The below figure shows the ROD Motherboard architecture.

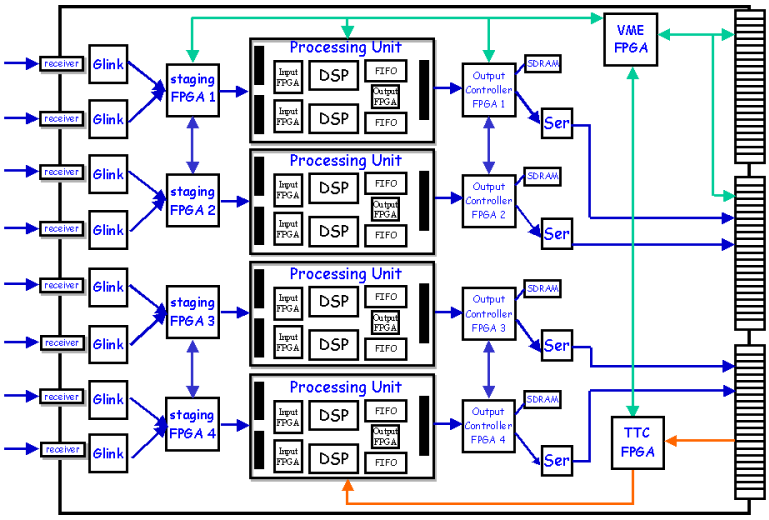

Figure 4 : The ROD motherboard architecture

4 The Mezzanine Architecture

4.1 The TMS320C6414 mezzanine architecture

Figure 5 shows the TMS320C6414 DSP mezzanine architecture :

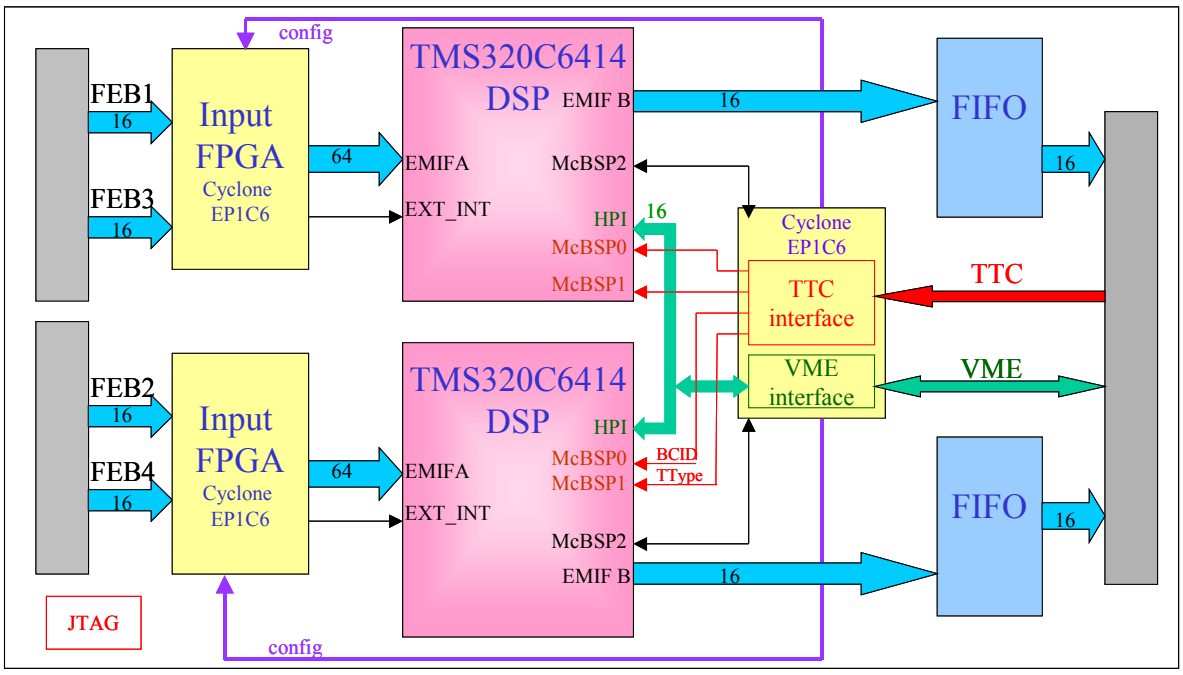

Figure 5 : TMS320C6414 DSP Prototype Mezzanine block diagram

The mezzanine is composed of two DSP blocks, able to treat each up to 128 calorimeter channels (1 FEB) in normal mode and 256 channels (2 FEB) in staging mode. Each DSP block is composed of an input FPGA (InFPGA), a TMS320C6414 DSP from Texas Instrument and an output FIFO. The mezzanine also contains an output FPGA (OutFPGA) used for the VME and TTC interface.
Input FEB data enters the InFPGA where it is formatted and checked as needed for the DSP algorithm. When an event is ready, an interrupt is sent to the DSP which launches a DMA to read the data on the 64-bit EMIFA bus. Once the DSP has finished processing an event, it writes the results in the output FIFO through the 16-bit EMIFB bus.
The TTC data is received in the OutFPGA and sent to each DSP via 2 serial ports (McBSP). A serial port is for the Trigger type and the other is for the BCID and EventID.

The OutFPGA allows the control of the board by VME, in particular :

- DSP boot written and histograms read through the 16-bit Host Port Interface (HPI) of the DSP.
- Full duplex serial port (McBSP2) with each DSP (run number written, DSP commands, status read)
- InFPGA configuration written (number of samples, number of gains, …) and status read through a serial line.
- InFPGA boot.

4.2 Boot of the PU

This is some elements that should be taken into account when booting the ROD system and in particular the PU:

1. When power turned on, a DSP reset is automatically provided by a microMonitor chip. This reset lasts 20 ms. It means that no access to the DSP must be performed during this time.
2. An EEPROM (EPC2) downloads the code in the OutFPGA.
3. The OutFPGA needs a reset before any VME access. This is performed through the ROD general reset. Then VME accesses are possible.

4. The InFPGA code must be downloaded from the VME, through the OutFPGA.
5. The DSP code must be downloaded from the VME, through the OutFPGA (through the HPI interface).
6. The InFPGA must be reset by the VME, through the configuration register of the OutFPGA. Then it can be configured (number of samples, gain, ...)
7. The Output FIFO must be reset by the DSP.
8. A TRSTn reset must also be performed to the DSP, so that they are properly initialized. While RESETn initializes the DSP core, TRSTn initializes the DSP's emulation logic. Both resets are required for proper operation and must be asserted upon power up. This can be performed from the motherboard, through the JTAG registers. A precise procedure must be followed. (see Jean Luc's note on this subject)

4.3 The input FPGA

The InFPGA parallelizes incoming FEB data, verifies their consistency (in particular potential corruption coming from radiation effects, like SEU), and formats the data as needed for the DSP algorithm. The InFPGA is a Cyclone EP1C6F256C8 from Altera.
The InFPGA has three main parts :

1. Parallelization and control of the incoming data.
2. Data organization in the dual port memory.
3. DSP interface.

Figure 6 shows the InFPGA architecture in staging mode, ie when treating 2 FEBs:

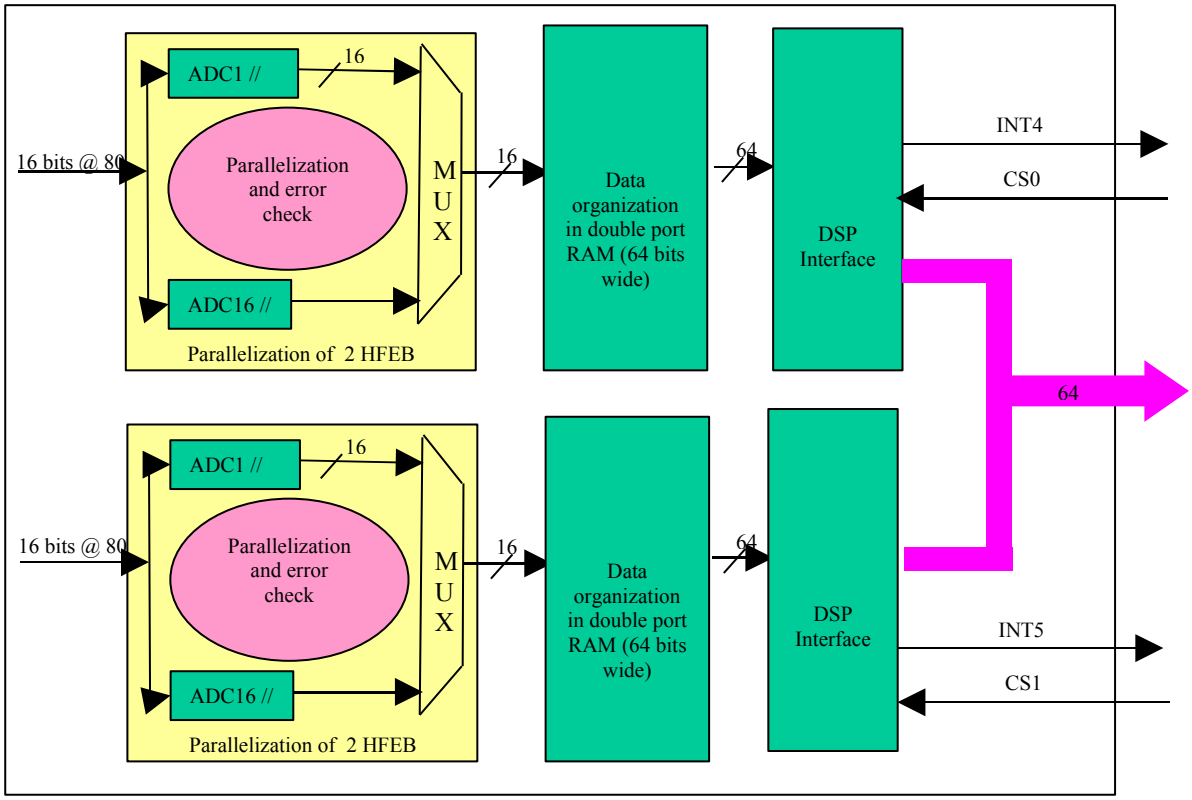

Figure 6 : Input FPGA architecture (staging mode)

4.3.1 Parallelization and control

The InFPGA receives input data from the mother board through two 80 MHz 16-bit bus. Each 16-bit bus corresponds to one FEB (16 ADC). In normal mode, the InFPGA treats one FEB, whereas in staging mode, the InFPGA treats two FEBs. FEBs are independent, and can be asynchronous, due to the length difference between optical fibers. That's why in staging mode, they are treated independently in the InFPGA. However their treatment is strictly similar.

The InFPGA first parallelizes and checks the input data :

- For each ADC, the InFPGA detects the start of event and parallelizes the data.
- Several checks are performed :

- Parity for each word, except the start and end of event.
- Start of data alignment to check half-FEB (HFEB) synchronization.
- Presence of the end of event (not yet implemented).
- Identical gain for all samples of a given channel.
- Identical RADD, BCID, EVT ID and SCAC status value within the 8 ADC of a same HFEB.
- Identical RADD, BCID and EVTID value between two HFEBs.
- Value of the SCAC status (should be 4801 or 805).
- No word with all zero or all ones bits during the event.
- Correct ADC Identifier (check if ADC0 ID is 0)
- Event start with flag bit = 0 ie means that channels 0-63 arrive first.
- Strict alternation of the flag bit during the event

- When en error is detected, the InFPGA fills the event status word. This word will be interpreted by the DSP and, if different to zero, the DSP will take the appropriate decision, as if asking for a FEB reset.
- The content of the event status word is described below:

17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Flag bit alternation	Flag bit start	FEB ID	ADC ID	Words at 0 or 1	SCAC 2	SCA C1	end	FE B_e vtid	FE B_ Rad d	FE B_ BCI D	parit y	gain	SC AC	radd	BCI D	evtid	ones

Table 1 : Content of the event status word

bit 0 ones : The beginning of the event is missing in one of the 16 groups of channels.
bit 1 evtID : EVTID or phase mismatch within half febs
bit 2 bcid : BCID mismatch within half febs
bit 3 radd : RADD mismatch within half febs
bit 4 scac : SCAC status mismatch within half febs
bit 5 gain : Gain mismatch (the gain bits are not preserved across the time samples, for some channels)
bit 6 parity: Parity errors (bit set for at least one parity error in the event)
bit 7 bcids : Comparison between BCID of channels 0-63 versus channels 64-128
bit 8 radds : Comparison between RADD of channels 0-63 versus channels 64-128
bit 9 evtids : Comparison between EVTID and phase of channels 0-63 versus channels 64-128
bit 10 end : The end of the event is missing in one of the 16 groups of channels (not yet implemented).
bit 11 scac 1 : Incorrect SCAC status in HFEB1 (should be 4801 or 805).
bit 12 scac 2 : Incorrect SCAC status in HFEB2 (should be 4801 or 805).
bit 13 words : A word with all zero or all ones bits was found during the event.
bit 14 adc id : ADC0 ID is not zero.
bit 15 feb id : 0 indicates FEB 1, whereas 1 indicated staged FEB.
bit 16 flag_bit_start ; event does not start with flag bit to 0.
Bit 17 flag_bit_alternation : error in the alternation of the flag bit during the event.
Bit [31..18] = 0

Notes concerning Glink error handling:

1. The InFPGA checks parity in every 16 bit words, except for the start of event. In case of at least one error in the event, bit 6 is set in the status word.
2. If the link loses frame lock, that is it is not linked at all, the LinkReadyn signal is high. In this case, the staging FPGA does not transfer data to the PU, because data are only ones, and the InFPGA would always detect start of event. This option can be removed if the bit 14 of the staging config 2 register is high.
3. The staging FPGA creates a signal called LinkLocked, which is set to zero as soon as LinkReadyn =1, and set to 1 if LinkReadyn=0 during more than 10 ms (LinkReadyn can oscillate if the Glink tries to regain frame lock). The signal LinkLocked is transmitted to the InFPGA. The AND of FEB LinkLocked and staging FEB LinkLocked is transmitted to the DSP through TInp1 pin, for control or histogramming (see table 11).
4. In case of very high error rate, the Glink control frame is corrupted and the LinkError signal is high. This signal is not transmitted to the PU. It is handled by the Staging FPGA, which increments a counter (+1), if there are more than 4 link errors during 400 ns. This counter can be read and reset from VME.

The Staging FPGA can also generate automatically Glink reset if the corresponding bit is set in the configuration register).

4.3.2 Data organization in the internal dual port memory

The EP1C6 Cyclone FPGA uses 32 kbits of internal memory per FEB. This memory is configured as a dual port memory, used to store data, before its transmission to the DSP. This memory is separated in two equal banks, one which is written, while the other is read by the DSP. The configuration in RAM type allows to write data in non consecutive addresses and thus organizes data to optimize the DSP algorithm execution. However if the event is bigger than 16 kbits (one memory bank), the event can not be completely reorganized.

The InFPGA allows three modes of FEB data organization. The first format can today only be applied for an event with 5 samples and one gain (most common case). It organizes data by channel, such as optimizing the DSP physics calculations (E,T,χ^2). The size of the internal memory of the InFPGA would allow format 1 organization up to 7 samples, but format 1 is today only implemented for 5 samples.

The second format can be applied whatever the number of samples and gain. In this format, FEB data is sent to the DSP almost in the same arrangement as it arrives on the PU. This format is mainly used for events with a high number of samples or for debugging purposes.

The third format is a transparent format. It allows event formatting whatever the number of samples and gains. This is the format, which is used for the BEC tests and for the Beam Test.

Today, each format (0,1,2) corresponds to a specific InFPGA code. Nevertheless, format 0 and format 2 InFPGA codes can be configured on line by VME, through the InFPGA configuration register, to work with any number of samples and gains.

4.3.2.1 FORMAT 1 : Data Organization for Nb_samples = 5 and Nb_gain= 1

The corresponding data format is presented below :

	bits [63..48]	bits[47..32]	bits [31..16]	bits[15..0]
0	Event status	Event status	EventID	BCID
1	ctrl1	ctrl2	ctrl3	Radd1
2	Radd2	Radd3	Radd4	Radd5
3	C0 Gain	C0 S1	C0 S2	C0 S3
4	C0 S4	C0 S5	C64 gain	C64 S1
5	C64 S2	C64 S3	C64 S4	C64 S5
6	C1 Gain	C1 S1	C1 S2	C1 S3
194	C127 S2	C127 S3	C127 S4	C127 S5

Table 2 : Dual port memory organization in format 1

In this format, 195 words per event and per FEB are sent to the DSP. The output data latency is 2 DSP cycles.
RADD, Ctrl1, ctrl2, ctrl3 words are kept for ADC 4.
For control 1, control 2, control 3 and RADD words, the parity is removed.
For each data word, the gain and parity information are removed.
All data words are shifted by 2 bits (stored on bits [13..2]) to accelerate the DSP algorithm.
The gain information (2 bits) is placed in bits 15 and 14.

4.3.2.2 FORMAT 2

Format 2 can be applied whatever the number of samples and gain in the event. Format 2 is presented below:

	bits [63..48]	bits[47..32]	bits [31..16]	bits[15..0]
0	0	0	EventID	BCID
1	ctrl1	ctrl2	Nb_gain	Nb_samples
2	RADD1ADC0	RADD1ADC8	RADD1ADC1	RADD1ADC9
3	RADD1ADC2	RADD1ADCA	RADD1ADC3	RADD1ADCB
4	RADD1ADC4	RADD1ADCC	RADD1ADC5	RADD1ADCD

5	RADD1ADC6	RADD1ADCE	RADD1ADC7	RADD1ADCF
6	S1 G1 ADC0 C0	S1 G1 ADC8 C0	S1 G1 ADC1 C0	S1 G1 ADC9 C0
	S1 G1 ADC2 C0			
	S1 G1 ADC6 C7	S1 G1 ADCE C7	S1 G1 ADC7 C7	S1 G1 ADCF C7
	S1 G2 ADC0 C0	S1 G2 ADC8 C0	S1 G2 ADC1 C0	S1 G2 ADC9 C0
	S1 G3 ADC6 C7	S1 G3 ADCE C7	S1 G3 ADC7 C7	S1 G3 ADCF C7
	RADD2ADC0	RADD2ADC8	RADD2ADC1	RADD2ADC9
	RADD2ADC2	RADD2ADCA	RADD2ADC3	RADD2ADCB
	RADD2ADC4	RADD2ADCC	RADD2ADC5	RADD2ADCD
	RADD2ADC6	RADD2ADCE	RADD2ADC7	RADD2ADCF
	S2 G1 ADC0 C0	S2 G1 ADC8 C0	S2 G1 ADC1 C0	S2 G1 ADC9 C0
	S2 G1 ADC2 C0			
	S2 G1 ADC6 C7	S2 G1 ADCE C7	S2 G1 ADC7 C7	S2 G1 ADCF C7
	S2 G2 ADC0 C0	S2 G2 ADC8 C0	S2 G2 ADC1 C0	S2 G2 ADC9 C0
	S2 G3 ADC6 C7	S2 G3 ADCE C7	S2 G3 ADC7 C7	S2 G3 ADCF C7
	RADD3ADC0	RADD3ADC8	RADD3ADC1	RADD3ADC9
	RADD3ADC2	RADD3ADCA	RADD3ADC3	RADD3ADCB
	RADD3ADC4	RADD3ADCC	RADD3ADC5	RADD3ADCD
	RADD3ADC6	RADD3ADCE	RADD3ADC7	RADD3ADCF
	SN G3 ADC6 C7	SN G3 ADCE C7	SN G3 ADC7 C7	SN G3 ADCF C7
	Status word	Status word	Ctrl3	0

Table 3 : Dual Port Memory Organization in format 2

Data is stored almost in the same order as it arrives from FEB. However :

- The event ID and BCID are extracted from control 1 and control 2 words to accelerate the synchronization task of the DSP.
- Ctrl1, ctrl2, ctrl3 words are kept for ADC 4.
- The number of samples and gain are specified.
- For each channel, the gain is only given for the first sample and masked on the other samples.
- A 32-bit status word is added at the end of the event.

The total number of words = 3+4*nb_samples + 32*nb_gain*nb_samples.
The below table gives the total number of words, function of the number of gains and samples. Results are given in 64-bit words and per FEB.

nb_samp/ nb_gain	1	3
3	111	303
5	183	503
7	255	703
16	579	1603
32	1155	3203

Table 4 : Number of words in the output event in format 2.

4.3.2.3 FORMAT 0 : Transparent mode

Format 0 can be applied whatever the number of samples and gain in the event.
Format 0 is the format that was used during tests at EMF, Combined Run and during the installation phase in USA15. It is implemented in the last InFPGA version (0355).
Format 0 is presented below:

	bits [63..48]	bits[47..32]	bits [31..16]	bits[15..0]
0	CTRL1 ADC0	CTRL1 ADC8	CTRL1 ADC1	CTRL1 ADC9
	CTRL1 ADC2	CTRL1 ADCA	CTRL1 ADC3	CTRL1 ADCB

	CTRL1 ADC4	CTRL1 ADCC	CTRL1 ADC5	CTRL1 ADCD
	CTRL1 ADC6	CTRL1 ADCE	CTRL1 ADC7	CTRL1 ADCF
1	CTRL2 ADC0	CTRL2 ADC8	CTRL2 ADC1	CTRL2 ADC9
	CTRL2 ADC2	CTRL2 ADCA	CTRL2 ADC3	CTRL2 ADCB
	CTRL2 ADC4	CTRL2 ADCC	CTRL2ADC5	CTRL2 ADCD
	CTRL2 ADC6	CTRL2 ADCE	CTRL2 ADC7	CTRL2 ADCF
2	RADD1 ADC0	RADD1 ADC8	RADD1 ADC1	RADD1 ADC9
3	RADD1 ADC2	RADD1 ADCA	RADD1 ADC3	RADD1 ADCB
4	RADD1 ADC4	RADD1 ADCC	RADD1 ADC5	RADD1 ADCD
5	RADD1 ADC6	RADD1 ADCE	RADD1 ADC7	RADD1 ADCF
6	S1 G1 ADC0 C0	S1 G1 ADC8 C0	S1 G1 ADC1 C0	S1 G1 ADC9 C0
	S1 G1 ADC2 C0			
	S1 G1 ADC6 C7	S1 G1 ADCE C7	S1 G1 ADC7 C7	S1 G1 ADCF C7
	S1 G2 ADC0 C0	S1 G2 ADC8 C0	S1 G2 ADC1 C0	S1 G2 ADC9 C0
	S1 G3 ADC6 C7	S1 G3 ADCE C7	S1 G3 ADC7 C7	S1 G3 ADCF C7
	RADD2ADC0	RADD2ADC8	RADD2ADC1	RADD2ADC9
	RADD2ADC2	RADD2ADCA	RADD2ADC3	RADD2ADCB
	RADD2ADC4	RADD2ADCC	RADD2ADC5	RADD2ADCD
	RADD2ADC6	RADD2ADCE	RADD2ADC7	RADD2ADCF
	S2 G1 ADC0 C0	S2 G1 ADC8 C0	S2 G1 ADC1 C0	S2 G1 ADC9 C0
	S2 G1 ADC2 C0			
	S2 G1 ADC6 C7	S2 G1 ADCE C7	S2 G1 ADC7 C7	S2 G1 ADCF C7
	S2 G2 ADC0 C0	S2 G2 ADC8 C0	S2 G2 ADC1 C0	S2 G2 ADC9 C0
	S2 G3 ADC6 C7	S2 G3 ADCE C7	S2 G3 ADC7 C7	S2 G3 ADCF C7
	RADD3ADC0	RADD3ADC8	RADD3ADC1	RADD3ADC9
	RADD3ADC2	RADD3ADCA	RADD3ADC3	RADD3ADCB
	RADD3ADC4	RADD3ADCC	RADD3ADC5	RADD3ADCD
	RADD3ADC6	RADD3ADCE	RADD3ADC7	RADD3ADCF
	SN G3 ADC6 C7	SN G3 ADCE C7	SN G3 ADC7 C7	SN G3 ADCF C7
	CTRL3ADC0	CTRL3ADC8	CTRL3ADC1	CTRL3ADC9
	CTRL3ADC2	CTRL3ADCA	CTRL3ADC3	CTRL3ADCB
	CTRL3ADC4	CTRL3ADCC	CTRL3ADC5	CTRL3ADCD
	CTRL3ADC6	CTRL3ADCE	CTRL3ADC7	CTRL3ADCF
	Status word	Status word	Nb_gain	Nb_samples

Data is stored in the same order as it arrives from FEB.

- The parity is kept for all words.
- The gain is kept for samples words.
- A 32-bit status word is added at the end of the event.
- The number of gains and samples are specified at the end of the event.

The total number of words = 1(status)+ 3*4(ctrl)+4*nb_samples (radd)+ 32*nb_gain*nb_samples.
The below table gives the total number of words, function of the number of gains and samples. Results are given in 64-bit words and per FEB.

nb_samp/ nb_gain	1	3
3	121	313
5	193	513
7	265	713
16	589	1613
32	1165	3213

Table 5 : Number of words in the output event in format 0.

4.3.3 DSP interface

When a complete FEB event is stored in the memory or the number of words written in the bank equals the chunk size defined in the InFPGA configuration register, the InFPGA sends an interrupt to the DSP. The DSP then launches a DMA to read the event. The reading is clocked by the DSP EMIFA clock, running at 120 MHz. From the reading side, the internal dual port memory of the input FPGA is seen by the DSP as a FIFO, implying that the data is read in consecutive addresses. There is one interrupt and DSP chip select per FEB.

4.3.4 LED and Test points

The InFPGA has 2 associated leds and 4 tests points.

Led1	DSP_GP0 pin
Led2	DSP_GP3 pin
TP1	FEB1 event is arriving
TP2	DSP read FEB1 data
TP3	BUSY signal
TP4	5 MHz clock for the watchdog input

Table 6: InFPGA LEDs and test points

4.3.5 Electrical characteristics

4.3.5.1 Compilation summary (2 FEB treatment)

	Used	**Percentage**
Logic elements	3224/5980	53 %
Pins	135/185	72 %
Memory	65k/92kbits	71 %

4.3.5.2 Maximum clock frequency

Clock Name	**Required Fmax**	**Actual simulated Fmax**
FEB clock	80 MHz	109 MHz
EMIFA DSP clk	120 MHz	148 MHz
TTC clock	40 MHz	138 MHz

4.4 The DSP

The DSP is the 720 MHz TMS320C6414GLZ, which is among the latest generation of the Texas Instrument DSP.
The DSP receives FEB data on its EMIFA memory bus, TTC information on serial ports McBSP0 and McBSP1 and transmits the output data through its EMIFB memory bus. The Host Port Interface (HPI) is used to boot the DSP and to read histograms.
The DSP is configured in Big Endian convention. It means that EMIFA, EMIF B and the HPI bus use this convention to receive or emit data.

Details about the DSP code are given in the following references :

- Input /output management and code structure: see reference [6] [7]
- Real time operating system : see reference [8]
- Physics code : see reference [9]
- Busy generation, event verification, event synchronization [10]
- DSP output format [15] [16].

The below paragraphs summarizes the DSP communications.

4.4.1 EMIFA

The External Memory Interface A (EMIFA) is directly connected to the InFPGA. The EMIFA is a 64-bit wide bus. The EMIFA clock AECLKOUT2 is generated internally and is configured to run at CPU/6 = 120 MHz. This EMIFA is used to read input FEB data in the InFPGA. As a dual port RAM is foreseen in the InFPGA for each FEB, one interrupt and one Chip Enable (CE) are foreseen per FEB.

CE	**DSP memory address**	**R/W**	**Connected to**	**Purpose**	**Mode**

CE0	8000 0000			Not defined	
CE1	9000 0000	R	InFPGA	FEB 2 DP-RAM (staging)	synchronous read interface.
CE2	A000 0000			Not defined	
CE3	B000 0000	R	InFPGA	FEB 1 DP-RAM	synchronous read interface.

Table 7 : EMIFA memory map

EMIFA latency must be configured to 3 cycles.

4.4.2 EMIFB

The External Memory Interface B (EMIFB) is directly connected to the output FIFO. The EMIFB is a 16-bit wide bus. The EMIFB clock BECLKOUT2 is generated internally and is configured to run at CPU/6 = 120 MHz. This EMIF is used to send output events to the FIFO.

CE	DSP memory address	R/W	Connected to	Purpose	Mode
CE0	6000 0000	W	Output FIFO	Output FIFO flags programming	synchronous with read enable mode (BAREn = 1)
	6010 0000	W		Output FIFO write	Synchronous (BAREn = 1)
	6010 0000	R		Output FIFO reset	synchronous (BAREn = 0)
CE1	6400 0000	W	Output FPGA	Not defined	
CE2	6800 0000		/		
CE3	6C00 0000	R	/		

Table 8 : EMIFB memory map

4.4.3 Host Port Interface (HPI)

The HPI is a 16-bit wide bus, connected between the DSP and the OutFPGA, through which the VME (host processor) can directly access the DSP's memory space. The VME has ease of access because it is the master of the interface. The host and DSP can exchange information via internal memory. The HPI is mainly used for the DSP boot, histogram reading and debugging purposes.

VMER/W	connected to	Purpose	Mode
W	output fpga	Boot	No particular configuration
R	Output fpga	Histograms, debug variables	No particular configuration

Table 9 : HPI description

Note : During DSP reset, the OutFPGA pulls down the HD5 signal to configure the HPI as a 16-bit words interface.

4.4.4 McBSP

The Multi channel Buffered Serial ports are full duplex communication DSP serial ports, with independent framing and clocking, for receiving and transmitting.
The TMS320C6414 has 3 McBSPs connected to the output FPGA. McBSP0 and McBSP1 are used for TTC data transfer (DSP receives only), while McBSP2 is bi-directional.

McBSP	R/W	Connected to	Purpose
McBSP0	R	output FPGA	BCID + Event ID
McBSP1	R	output FPGA	Trigger type transmission
McBSP2	R/W	OutputFPGA	DSP Commands write Interrupt identifier vector read…

Table 10 : Table 7 : McBSP description

4.4.5 Interrupts and general purposes Pins

The below table summarizes the DSP General Purposes (GP) configuration pins.

GP	Function	Connected to	Purpose
GP0	Output GP	InFPGA	Input FPGA led1 blinking
GP1/CLKOUT 4		NC	
GP2/CLKOUT	Clkout6	OutFPGA	Clkout6 check

6			
GP3	Output GP	InFPGA	Input FPGA led2 blinking
GP4/EXT_INT 4	Interrupt	InFPGA	Interrupt dedicated to FEB1 DMA
GP5/EXT_INT 5	Interrupt	InFPGA	Interrupt dedicated to FEB2 DMA
GP6/EXT_INT 6	input	OutFPGA	FIFO almost full
GP7/EXT_INT 7	Interrupt	OutFPGA	Fifo counter interrupt (every 256 read words)
GP8/CLKS2		NC	
GP9	Output GP	InFPGA	PU_IRQ
GP10	Output GP	InFPGA	BUSY
GP11	Output GP	OutFPGA	Testpoint
GP12	Output GP	OutFPGA	Testpoint
GP13	Output GP	OutFPGA	DSP fifo counter reset
GP14	Input GP	OutFPGA	Fifo empty
GP15	Output GP	NC	
TINP0	Input GP	InFPGA	Not used (clk manager)
TINP1	Input GP	InFPGA	FEB link locked and staging FEB link locked
TOUT0	Output GP	InFPGA	Not used
TOUT1	Output GP	InFPGA	Not used

Table 11 : interrupts and general purposes pins

4.5 The Output FPGA

The OutFPGA is a Cyclone EP1C6F256C8 from Altera. This FPGA has 2 main purposes: the TTC and VME interface.

4.5.1 The TTC interface

The OutFPGA buffers the TTC data coming from the motherboard TTC FPGA and going to both DSP, through 2 serial ports also called McBSPs. McBSP0 is reserved for the BCID and event ID, whereas the McBSP1 is reserved for the Trigger type. The TTC protocol is described §3.2.4.

4.5.2 The VME interface

The OutFPGA includes the VME interface of the mezzanine board. It decodes data coming from the motherboard and generates data according to the motherboard protocol. The communication protocol between the mezzanine and VME FPGA of the mother board is a home made protocol, in which the PU is always the slave. This protocol is described in reference [4].

The below table summarizes all the OutFPGA registers that can be accessed by VME (32-bit wide).

PU_add[4..0]	VMEadd	R/W	Function
0	+0	R/W	Test register
1	+4	R/W	control 1 register
2	+8	R/W	HPI register (DSP1)
3	+C	R	Status 1 register
4	+10	W	Broadcast HPI register (both DSP)
5	+14	W	serial data to McBSP2 (DSP1)
6	+18	R	Serial data from McBSP2 (DSP1)
7	+1C	W	InFPGA1 configuration register (16 LSB)
8	+20	W	InFPGA1 programmation register (8 MSB)
9	+24	R	InFPGA1 status register
10	+28	W	Broadcast InFPGA programmation register (both InFPGA)
11	+2C	R	OutFPGA version register
16	+40	R/W	Test register
17	+44	R/W	control 2 register
18	+48	R/W	HPI register (DSP2)
19	+4C	R	Status 2 register
20	+40	W	Broadcast HPI register (both DSP)
21	+54	W	serial data to McBSP2 (DSP2)
22	+58	R	Serial data from McBSP2 (DSP2)
23	+5C	W	InFPGA2 configuration register (16 LSB)
24	+60	W	InFPGA2 programmation register (8 MSB)
25	+64	R	InFPGA2 status register
26	+68	W	Broadcast InFPGA programmation register (both InFPGA)
27	+6C	R	OutFPGA version register

Table 12 : VME Memory Map

Note : All registers that must be accessed through VME FIFO mode (specific to Concurrent Technology VP110 CPU), are required to be placed in an address ending by 0 or 8. This concerns the DSPs and InFPGAs boot and the HPI read.

The below paragraphs detail the purpose of each register.

4.5.2.1 Control register

The control register is described in the below table:

Bit number	R/W	Bit wise operator	Functionality
0	R/W	0000 0001	DSP reset
1	R/W	0000 0002	Partial FIFO reset
2	R/W	0000 0004	Input FPGA reset
3	R/W	0000 0008	HPI reset
4	R/W	0000 0010	HPI burst
5	R/W	0000 0020	DSP launch

6	R/W	0000 0040	/Input fpga nconfig

Table 13 : Control Register

4.5.2.2 Status register

The status register is described in the below table:

Bit number	R/W	Bit wise operator	Functionality
[7:0]	R	0000 00FF	FIFO counter status [7..0]
8	R	0000 0100	output FIFO empty flag
9	R	0000 0200	output FIFO almost empty flag
10	R	0000 0400	output FIFO half full flag
11	R	0000 0800	output FIFO Almost full flag
12	R	0000 1000	output FIFO full flag
[14..13]		0000 6000	FIFO counter status [9..8]
[20..16]		001F 0000	Nb of words in McBSP2 FIFO
21		0020 0000	McBSP2 FIFO empty flag
22		0040 0000	McBSP2 FIFO almost full flag (> 24 words)
23		0080 0000	McBSP2 FIFO full flag (32 words)
24	R	0100 0000	GP11
25	R	0200 0000	GP12
26	R	0400 0000	GP13
27	R	0800 0000	output FIFO empty flag
28	R	1000 0000	HPI INT
29	R	2000 0000	HPI Ready
30	R	4000 0000	InFPGA nstatus
31	R	8000 0000	InFPGA confdone

Table 14 : Status register

4.5.2.3 HPI interface

Both DSPs can be accessed simultaneously (broadcast mode) or independently through the specific registers.
The way the HPI protocol is implemented in the output FPGA is described in [11].

4.5.2.4 McBSP2 interface

The OutFPGA has the possibility to exchange data with both DSPs through the McBSP2 serial port. This serial port is bi-directional. McBSP2 write access allows DSP commands sent, while McBSP2 read access allows for example interrupt identifier vector read.
Fot the read access side, a FIFO of 32 bits words of 32 bits, has been implemented. The status of this FIFO can be read in the status register.

4.5.2.5 Input FPGA programming

The OutFPGA allows the programming of the InFPGA from VME. The ability to re-program the InFPGA from VME, ie without use of the byte blaster, allows the configuration of the InFPGA at any time, with any code (eg. Format 1, format 2, …). The OutFPGA programs the InFPGAs through dedicated configuration signals (data0, dclk, nconfig, …) clocked at 5 MHz. Both InFPGA can be configured at the same time (broadcast mode).

4.5.2.6 Input FPGA configuration and status register

The OutFPGA can communicate with each InFPGA through a serial protocol. This allows the configuration of the InFPGA and the reading of the InFPGA status. The below tables describe the content of these registers :

15 8	7 6	5 0
Size of chunks -1	Number of gains	Number of samples

Table 15 : InFPGA configuration register

By default, at the InFPGA reset, the configuration register is configured with : FF45

31 16	15 8	7 6	5 0
InFPGA version	Size of chunks -1	Number of gains	Number of samples

Table 16 : InFPGA status register

4.5.3 Output FPGA LEDs and Test points

The OutFPGA has 2 associated leds and 4 tests points.

Led1	FIFO1 is empty
Led2	FIFO1 is almost full.
TP1	GP111
TP2	TTC_BCID_frame
TP3	GP112
TP4	PU_CTRL_ADD

Table 17 : OutFPGA LEDs and test points

4.5.4 Electrical characteristics

4.5.4.1 Compilation summary

	Used	**Pourcentage**
Logic elements	1430/5980	23 %
Pins	151/185	81 %
Memory	0/92kbits	0 %

4.5.4.2 Maximum clock frequency

Clock Name	**Required Fmax**	**Actual simulated Fmax**
PU clock	40 MHz	118 MHz
TTC clock	40 MHz	150 MHz

4.6 The output FIFO

The Output FIFO is an IDT72V253L7-5BC 3.3V high density super synchronous 18-bit bus FIFO. The FIFO is organized in 4 kilo words of 18 bits. It allows the storage of about 10 usual FEB events. (event output size ≈ half of the input event size = ½ [800*16 bits] = 400*16 bits). The DSP controls the status of the FIFO, through the Programmable Almost Full Flag. The FIFO possesses empty, full, Programmable Almost Empty (PAE) and Programmable Almost Full (PAF) flags. The PAE and PAF offset values are configurable from the DSP, using the EMIFB interface. During the initialization phase, the offset flag value is configured to 1024 words.

For most of the events, it allows the DSP to send an entire event at a time (one output DMA only) and so does not waste time.
For big output events (raw data, 32 samples, …), the DSP slices up the event. The DSP uses chained DMA, each DMA corresponding to a slice of event. To indicate the status of the FIFO, the Output Controller sends an interrupt to the DSP (through pins FIFO_end_event and ext_int7 of the OutFPGA)

4.7 Power supply and JTAG

4.7.1 Power-supply

The mezzanine board is supplied with an 3.3V unique voltage, coming from the mother board and made from the VME 48V through a DC-DC regulator. The power consumption is about 1.5A for the whole mezzanine.
The following voltages are derived from this 3.3V supply :

- the 1.5 V for the InFPGA and OutFPGA cores with two regulators TPS76815Q from Texas Instrument.
- the 1.4 V for the DSP core with an adjustable regulator, the TPS54610 from Texas Instrument.
- The 3.3V for the DSP input/output, which is set up after the DSP core voltage and with specific setup requirements.

4.7.2 JTAG chain

The JTAG chain includes in this order : DSP1, DSP2, InFPGA1, InFPGA2, OutFPGA, FIFO1, FIFO2, EPC2,.
The JTAG chain has 2 purposes on the board :

1. Boundary scan analyses: to detect faults coming from the production.
2. Altera components programming : FPGAs and EPC2 configuration.

The JTAG chain is controlled from the motherboard. For debugging purposes, the footprint of a 2*5 plug connector, compatible with the Byte Blaster Pad, exists on the PU. This connector is not populated in the experiment, because it is too high and would unable the next motherboard to be inserted in the neighbouring slot.

5 Technical Realizations

5.1 Picture of the PU

The below picture shows the mezzanine board.

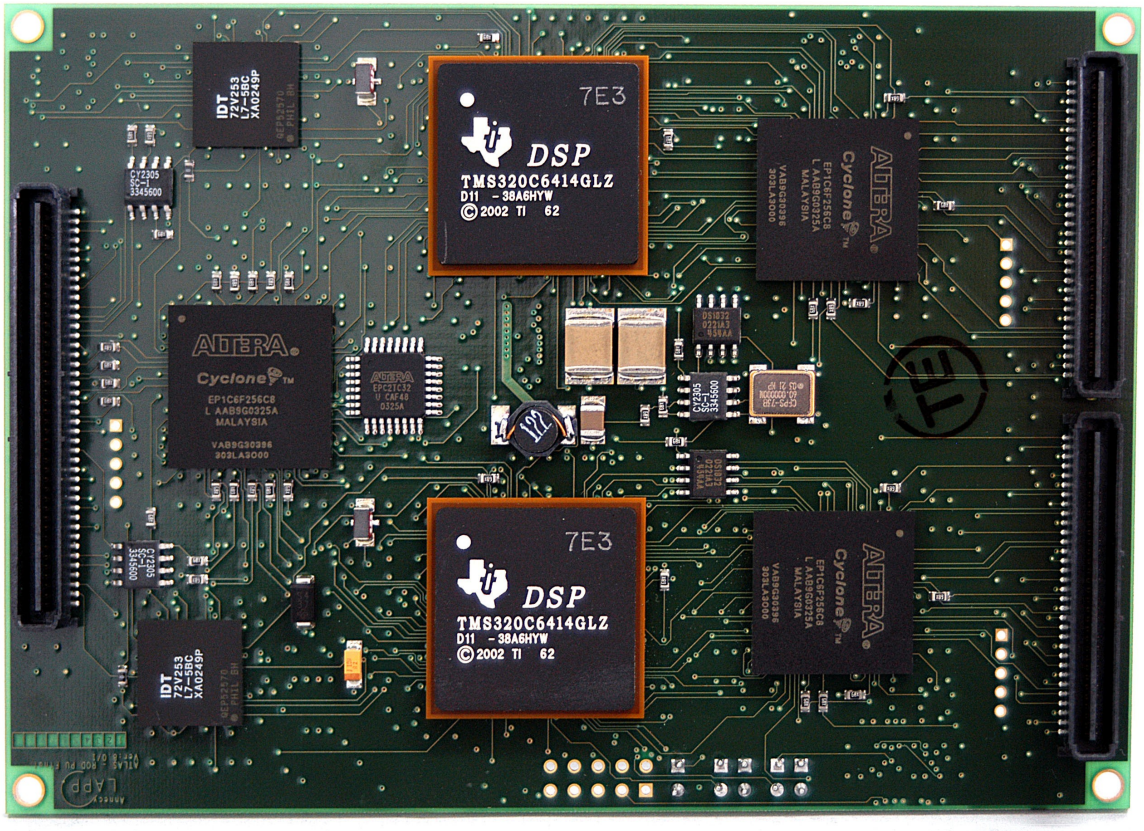

Figure 7 : Picture of the ROD PU

5.2 Board Routing

5.2.1 Technical characteristics

The mezzanine is a 120*85*1.4 mm board, with 10 layers. The board contains about 2600 etches and 2300 vias.
The vias are through hole vias and have the following characteristics :

- 0.5 mm diameter and 0.25 drill size under the DSP (0.8 mm BGA).
- 0.6 mm diameter and 0.3 drill size elsewhere.

The minimal line width is 120 um, as the minimal spacing.
The below figure summarizes the layout characteristics.

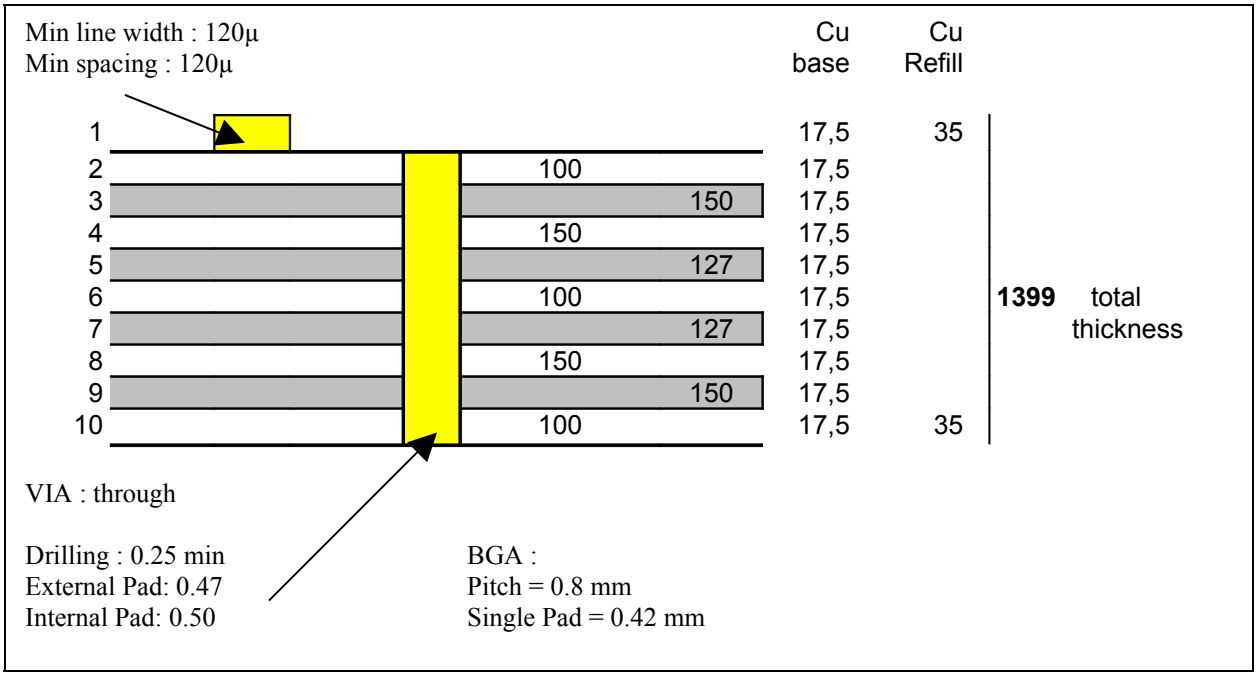

Figure 8 : Layout characteristics

5.2.2 Layout Cross Section

The mezzanine board has 10 layers: 6 signal layers and 4 power planes. The layers are organized so that the signal layers present almost the same impedance (around 50 ohm for a 0.120 mm wide net) to preserve signal integrity. The below table summarizes the layout cross section.

Material	type	Name	Thickness	Line width	Impedance
air	surface				
copper	conductive	Top	0,0525	0,12	57,8
FR4	dielectric	dielec1	0,1		
copper	plane	gnd1	0,0175		
FR4	dielectric	dielec2	0,15		
copper	conductive	signal1	0,0175	0,12	48,8
FR4	dielectric	dielec3	0,15		
copper	plane	VCC1 (3.3VDB)	0,0175		
FR4	dielectric	dielec4	0,127		
copper	conductive	signal2	0,0175	0,12	51,1
FR4	dielectric	dielec5	0,1		
copper	conductive	signal3	0,0175	0,12	51,1
FR4	dielectric	dielec6	0,127		
copper	plane	VCC2 (1.4V/1.8V/2.5V)	0,0175		
FR4	dielectric	dielec7	0,15		
copper	conductive	signal4	0,0175	0,12	48,8
FR4	dielectric	dielec8	0,15		
copper	plane	VCC3 (3.3 V)	0,0175		
FR4	dielectric	dielec9	0,1		
copper	conductive	bottom	0,0525	0,12	57,8
			1,399		

Table 18 : Mezzanine Layout Cross Section

5.2.3 Constraints on Routing

The mezzanine board layout has fast I/O pins with fall times as low as 1 ns to 3ns, that can contribute to noise generation, signal reflection, cross talk and ground bounce. That's why, we took special care :

- To filter noise coming from power distribution :
  - Use of decoupling capacitors
  - Use of power planes.
- To respect sensitive signals including clocks.
  - Point to point connection (use of zero delay PLL for clock when necessary)
  - Traces as straight as possible. The clock signals are routed first.

        - Number of vias minimized.
- To match impedance and use termination line.
        - Special layout cross section.
        - 50 ohm serial resistor at emitter side when necessary
- To minimize cross talk between parallel traces:
        - Spacing of 0.36 mm minimum at each side of clock signals.

5.3 Manufacturing

In fall 2003, 17 prototypes were produced in prevision of the back end crate test. The PCBs were produced by the firm Techci, based in Saint Genix sur Guiers (73), France, while the boards were cabled by the firm "Ardelec technologie", based in Saint Agreve (07), France. Both enterprises are ISO 9001:2000 certified.
16 boards were tested successfully. The 17 th, presents a problem with the EPC2 configuration (to be clarified).

5.3.1 Printed Circuit

The Printed Circuit Board (PCB) is made according to the below specified technical characteristics:

- Dielectric : Fr4
- Double face varnish.
- Chemical Ni Au finishing
- Layers impedance granted with 10% of margin.

After the PCB production, a complete (100 %) electrical test is performed on each PCB.

5.3.2 Assembling

The ROD PU has about 285 CMS components, cabled on both sides of the PCB, amongst which there are 7 BGA: two 532 pins 0.8 mm pad BGA, two 100 pins 1 mm pad BGA and three 256 pins 1 mm pad BGA.
Prior to assembling, the PCBs are stored under controlled temperature, to prevent PCB oxidization. The components placement is done automatically. The double refusion process is performed with nitrogen gas, to ensure a better quality of cabling between the PCB and components and a better board aspect.
After the assembling, the following checks are performed :

- Visual check
- BGA check using X rays.
- Interconnection test (JTAG) using CASCON tool.

5.4 Tests Performed

To validate the ROD PU, several tests were performed :

5.4.1 Interconnection test

The interconnection test is performed via the JTAG protocol, to prevent faults coming from the production. It allows to identify stuck at, short and open circuits and defaults. Boundary scan is the main test to validate the electrical functionality of the mezzanine and on a whole the ROD module. More details about these tools and the ROD test bench can be found in reference [14].

5.4.2 Functional tests

The following functional tests were performed on the PU :

- VME interface
- Boot of the PU.
- FEB data injection using the staging FPGA of the motherboard.
        - Event with 5 samples, 1 gain
        - Up to 100 kHz L1A frequency.
        - Normal mode (1 FEB/DSP) and staging mode (2 FEB per DSP)
- FEB data injection using the Injector board.
        - Event with any samples, any gain
        - Up to 100 kHz L1A frequency.
        - Normal mode (1 FEB/DSP) and staging mode (2 FEB per DSP)
- FEB data injection using FEB.
- Data formatting in the InFPGA, according to format 0, format 1 and format2.

- Use of different DSP codes :
    - Mode check and copy : input data is checked and transferred at the output
    - Physics mode : different physic algorithms are applied on the incoming data.
- Reception of TTC data in the DSP.
- Synchronization check between FEB and TTC data.
- McBSP2 use (in both directions)
- Communication between the InFPGA and OutFPGA.
- HPI read/write
- Output FIFO write and configuration from DSP.
- Output data read by the OC, sent to the SDRAM or SLINK and checked.
- Use of BUSY signal to handle Xon/Xoff protocol and stop data injection.

5.4.3 Long term tests

Injection with the staging FPGA, normal mode (1FEB/DSP) :

Always the same event, 5 samples, 1 gain
DSP in mode check and copy, data read by the SLINK.
Busy handled.

- Performed on 1 PU, during a night (15h), 3.4 giga events sent, L1A frequency = 75 kHz per burst, 63 kHz in average.
    - ➢ No error detected by the SLINK, no event lost.
- Performed on 4 PU, 6 millions of events sent, L1A frequency = 75 kHz per burst, 20 kHz in average.
    - ➢ No error detected by the DSP, no event lost in the SLINK

Injection with the staging FPGA, staging mode (2 FEB/DSP) :

- Performed on 2 PU, 6 millions of events sent, L1A frequency = 75 kHz per burst
    - ➢ No error detected by the DSP, no event lost in the SLINK

Injection with the Injector board with synchronized FEB and TTC data :

- 3 Billions of events transferred though each of the 4 Pus.
- L1A frequency = 100 kHz (busy handled)
- Injector was sending incremental data 5 samples events.
- No error detected on the Slink.

Injection with the Injector board, in staging mode:

- 800 millions of events transferred though each of the DSP.
- L1A frequency = 40 kHz (busy handled)
- Injector was sending incremental data 5 samples events.
- No error detected on the Slink.

5.4.4 Power consumption

In idle state, the power consumption of the PU was measured to 0.9 A, while in run state around 1.2 A.

5.5 ROD Processing Units Production Procedure

5.5.1 The Production Procedure

Within the context of the French public market, LAPP has launched a limited call for tender for the realization of the ROD Processing Units (estimated market over 130 000 euros).

The limited call for tender consists of :

1. A market survey published in the JOCE (Official Bulletin of the European community). This market survey briefly describes the market object and the technical criteria.
2. A first firm selection after a minimum legal delay of 37 days.
3. A call for tender sent to the selected firms, into which all the useful information for the production of the PU are provided.
4. A final choice of the firm responsible of the production, after a minimum legal delay of 40 days after the sent of the call for tender.

The call for tender of the ROD PU includes :

- The components purchase.
- The PCB manufacturing.
- The board assembly.
- Some tests.

5.5.2 Number of boards to be produced

384 (192*2) PU are needed for the staging mode and 768 (192*4) PU are needed for the normal mode.
10 % of spare boards and no additional components are planned.

LAPP has launched a limited call for tender for the realization of :

- of a definite batch of 426 cards in which a pilot production of validation of 16 cards.
- of a possible batch n°1 of 410 cards (for non staging mode)
- of two possible batches n°2 and n°3 of 20 cards each (for possible additional spares)
- of two possible batches n°4 and n°5 of 76 cards each (for Tiles)

5.5.3 Agenda

The PU production procedure has started.
The market survey was published at the end of January 2004.
Firms had to answer this market survey for March 10 th.
The call for tenders was sent to the selected firms at the end of April.
Firms have to answer for June 15 th.
We hope to place the order at the end of July, or on the latest at the beginning of September.
PU reception is planned 5 months maximum after the order (delay précised in the call for tender), ie at the end of 2004 or at the latest in January 05.
The installation will begin at CERN, in April 05.

5.5.4 Testing procedure

For the testing procedure, 2 identical test benches are foreseen, one at Geneva and the other at LAPP. Half of the boards (PU and Motherboards) will be tested in each lab.
More details concerning the testing procedure of the ROD PU are given in the “Production Tests of the ROD Boards (Motherboards and Processing Units)” document [17].

5.5.5 Additional Information

More information concerning the PU production procedure can be found in :

- The announce [18]
- The CCTP [19]
- The CCAP [20]
- The RDC [21].

All these documents were written in French for the market survey and call for tender.

6 REFERENCES

[1] "Study of energy reconstruction using optimal filtering with the LAr electromagnetic calorimeter". Isabelle Wingerter-Seez, 1995, ATL-LARG95-019

[2] Bill Cleland and Stefan Simion
"ROD PU acceptance criteria and parameters " April 24 th, 2002

[3] Pascal Perrodo, "Requirements to build the calibration system for the ATLAS LAr calorimeters and how to perform a full calibration sequence". August 2001
https://edms.cern.ch/cedar/plsql/doc.info?cookie=2363023&document_id=322832&version=1

[4] Annie Leger "VME protocol", October 2002
http://dpnc.unige.ch/LArgROD/VMEBusyFPGA.pdf

[5] University of Geneva, "The ROD motherboard for the ATLAS liquid argon calorimeter" December 2003
http://dpnc.unige.ch/LArgROD/ROD_MB.pdf

[6] Remi Lafaye and Nicolas Chevillot « Global View on Hardware and code structure. TestFinal v1.3" January 2004
http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-TESTDEMO-GEN-1.2.pdf

[7] Texas Instrument "TMS320C6414, Fixed Point processor" 2003
http://focus.ti.com/docs/prod/folders/print/tms320c6414.html

[8] Gelu Ionescu "RTX " September 2002
http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_RTX-TI-SRS1.pdf

[9] Remi Lafaye "The physics ROD 64x PU algorithm " January 2004
http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-DSP-PHYS.pdf

[10] Gelu Ionescu "Busy generation, event verification, event synchronization " September 2002
http://wwwlapp.in2p3.fr/informatique/Atlas_online/Documents/Dsp/ATL-ROD-RTX/AOS_ATL-SYNC-ANA.pdf

[11] Nicolas Chevillot "TMS320C6414 HPI interface" December 2001
https://edms.in2p3.fr/file/I-002610/1/HPI.pdf

[12] Stefan Simion "DSP to FIFO interface."
http://www.nevis.columbia.edu/~simion/rod_work/dsp_fifo.pdf

[13] Gelu Ionescu "FIFO simulation."
https://edms.in2p3.fr/file/I-002606/1/FIFO.pdf

[14] Guy Perrot "ROD Test Bench and integration tests" March 2002
http://wwwlapp.in2p3.fr/Electronique/Experiences/ATLAS-ELEC/Testbench/ROD_Testbench.pdf

[15] Remi Lafaye and Walter Lampl « Athena Larg event converter » January 2004

[16] Daniel La Marra, "The Output Controller FPGA of the Larg Rod board description". January 2004.
http://dpnc.unige.ch/LArgROD/OCLArgRODDescr.pdf

[17] "Production Tests of the ROD Boards (Motherboards and Processing Units)".

[18] The announce

[19] The CCTP

[20] The CCAP

[21] The RDC

7 ABBREVIATIONS

BCID	Bunch Crossing Identifier
DMA	Direct memory access
DSP	Digital Signal Processor
EMIF	External Memory InterFace of the DSP.
FEB	Front End Board
FIFO	First In First Out
HFEB	Half FEB
HPI	Host port Interface of the DSP
InFPGA	Input FPGA
McBSP	Multi channels Buffered Serial Port of the DSP
OC	Output Controller
OutFPGA	Output FPGA
RADD	Read address in the SCA (Switch Capacitors Table)
SEU	Single Event Upset

8 ANNEXES

8.1 Annex 1 : The serial FEB data protocol

Data from one FEB arrives on a 16-bit data bus. Each half FEB encodes 8 ADC data. At the output of the FEB, each 16-bit bus of the 8 ADC (0-7) of half FEB 1, is serialized on 2 bits and multiplexed with data from ADC 8-15 of half FEB 2.
The below figure shows this data treatment :

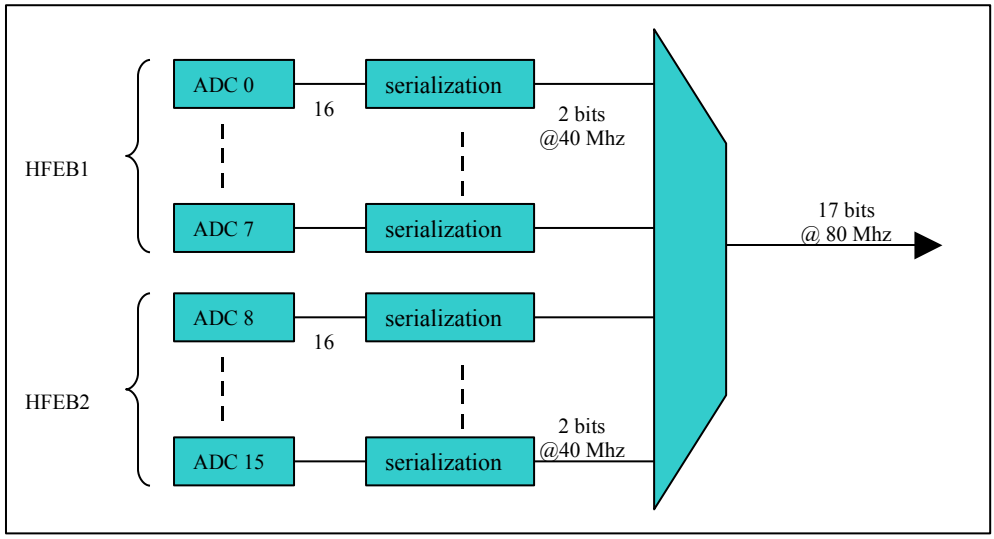

Figure 9 : Data treatment at the output of the FEB

At the output of the MUX, the method in which the 17-bit data bus is encoded is described below :

Data [0] and data[1] correspond to the 2 bits of ADC 0 and 8.
Data [2] and data[3] correspond to the 2 bits of ADC 1 and 9.
Data [14] and data[15] correspond to the 2 bits of ADC 7 and 15.

Most significant bit (MSB) arrives first (Big endian convention).
Data [16] indicates, the half FEB (HFEB1 ADC 0-7 or HFEB2 ADC 7-15) which is currently arriving.
Data [16] = 0 means channels 0-63 come out.
Data [16] = 1 means channels 64-127 come out.

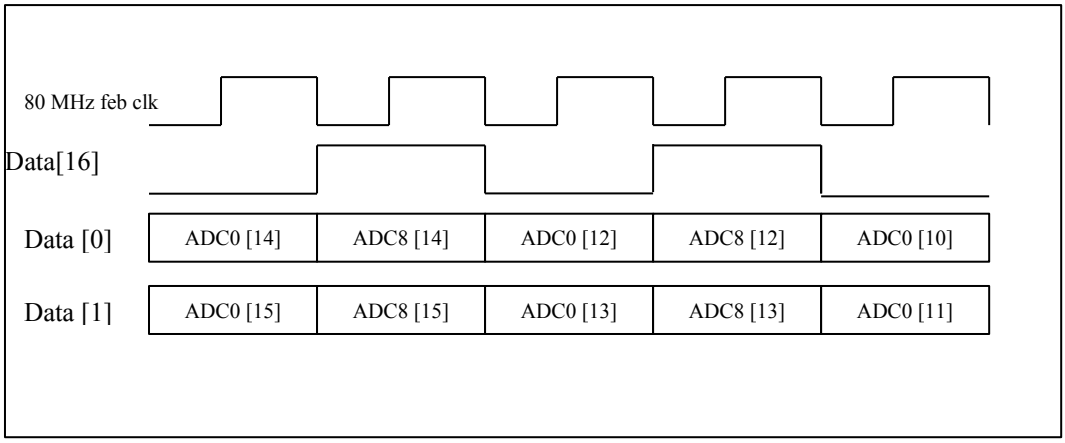

Figure 10 : Output MUX encoding

8.2 Annex 2 : ADC Read out input event format

	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
W1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
W2 : ctrl1	0	P	0	0	ADCID				PHASE			EVENTN				
W3: ctrl2	0	P	0	0	BCID											
W4 : radd	0	P	0	0	F	L	B	A	CELLN							
W5	0	P	Gain		ADC channel 0 sample 1											
W6	0	P	Gain		ADC channel 1 sample 1											
W7	0	P	Gain		ADC channel 2 sample 1											
W8	0	P	Gain		ADC channel 3 sample 1											
W9	0	P	Gain		ADC channel 4 sample 1											
W10	0	P	Gain		ADC channel 5 sample 1											
W11	0	P	Gain		ADC channel 6 sample 1											
W12	0	P	Gain		ADC channel 7 sample 1											
...	...															
...	...															
WEVT-10 : radd	0	P	0	0	F	L	B	A	CELLN							
WEVT-9	0	P	Gain		ADC channel 0 last sample											
...					...											
WEVT-2	0	P	Gain		ADC channel 7 last sample											
WEVT-1 : ctrl3	0	P	0	0	1	S	E	SCAC status								1
WEVT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- W1 : frame start tag

- W2 & W3 : event header
  ADCID : The ADC IDentifier.
  PHASE : Bits encoding the phase of the 5MHz rclk at the time of the trigger.
  EVENTN : Event number.
  BCID : Bunch Counter IDentifier.
  P : Parity

- W4 →WEVT-2 : Sample data
  * W4 : first sample header also called RADD
    F : First sample of event, sends event header, and performs gain selection algorithm (if in auto gain mode).
    L : Last sample of an event, sends event status, and delays further samples to make room for the event header of next event.
    B : Backporch bit. Does not raise the gain for any channel in auto-gain mode.
    A : Test mode bit.
    CELLN : The SCA cell (capacitor) number.

  * From W5 to W12 : first sample data
    Gain : - 00 test data.
    - 01 low gain.
    - 10 medium gain.
    - 11 high gain.
    ADC : ADC data
  * WEVT-10 : last sample header

  * From WEVT-9 to WEVT-2 : last sample data

- WEVT-1 : event trailer
  - B11 & B0 : forced high.
  - **S** : Single bit errors in EDC (Error Detection & Correction) unit.
  - **E** : Double bit errors in EDC unit.
  - **SCA C**ontroller status :
    - B8 : SEU in the latency's redundant MSB's.
    - B7 : done_fifo overflow.
    - B6 : free_fifo Underrun The SCA run out of free cells. Non valid cell numbers were inserted into the sequence.
    - B5 : Sequence error : The free_fifo emitted the same cell number twice in a row.
    - B4 : Single bit error : a cell address was found with a single bit error. This error was corrected.
    - B3 : Double bit error. A cell address was found with two bits in error. This bit will reoccur repeatedly, as the error will not be corrected.
    - B2 : BCID reset. The BCID counter was reset.
    - B1 : Init. Bit set in the first event after initialisation.

- WEVT : frame end tag

- For all words except the frame start/end tag
  - B14 : odd parity on bit 0-13, 15.
  - Note : the parity bit is set to 0 or 1 in order to always have an odd number of ones in the 16-bit word ; therefore it is impossible to have a "frame end tag" in an event.

8.3 Meaning of the PU's LED and test points