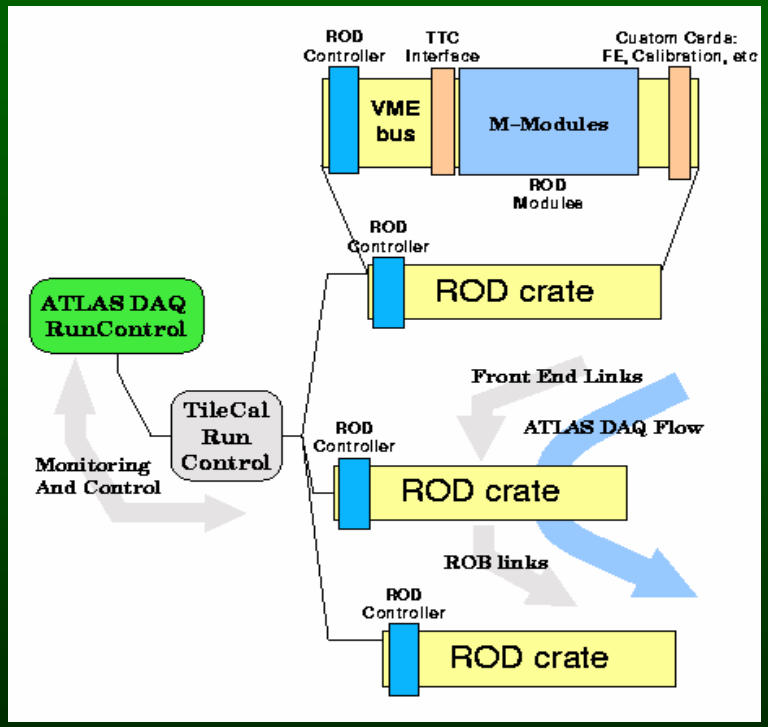
On the developments of the Read Out Driver for the ATLAS Tile Calorimeter

Enrique Sanchis, Vicente González,
Jose Torres, Jose Castelo
IFIC - Universidad de Valencia
Tile Calorimeter – ROD
Stockholm, 12 Sept. 2001

ROD CRATE

- Based on ROD Crate
  - ***ROD Controller***
  - ***TTC interface*** *(TBM): Trigger and Busy Module*
  - *Links to*
    - **TileCal Run Control**
    - **ATLAS DAQ Run Control**



ROD Controller (I)

- Tasks
  - *Monitoring*
    - Drives monitoring data flow
    - Uses VME to exchange control and monitoring flow
    - Spy 1-5% of data flow and 100% of special trigger (calibration)
    - Local DAQ in absence of ROB
    - Minimum Bias And Pile- Up Noise must be reported to provide feedback to Optimal Filtering
    - Statistics related with ROD activity

ROD Controller (II)

- Control
  - *Shared function between CPU ROD Crate Controller and each ROD board*
  - *Functions*
    - Software downloading from a ATLAS General DataBase through VME (DSP code, Calibration coefficients, Optimal filter parameters, etc …)
    - On Physics or Calibration triggers different programs VME transferred on-line to DSPs

TTC Interface

- Implemented by the Trigger and Busy Module (VMEbus module)
  - *Receives the TTC info, makes a O/E conversion, and sends a LVDS signal through P3 to all TTCrx chips per ROD*
  - *Sends the Busy Signal generated by an "OR" logic function of all modules of the ROD crate: issue a per crate ROD_BUSY signal*

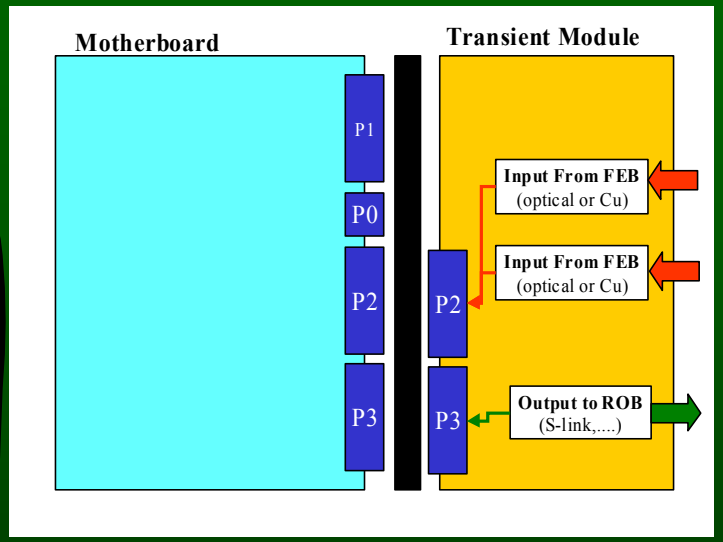
THE ROD Module (I)

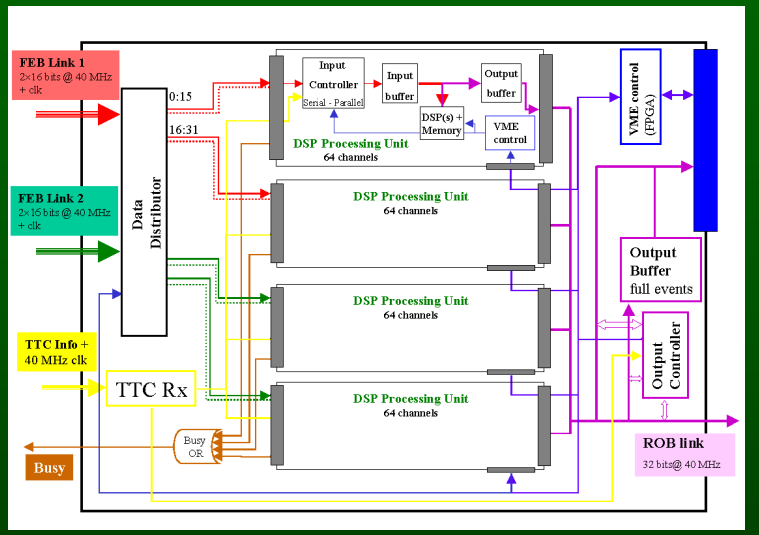
- **Functions – Requirements**:
  - *General functions as described in the TDAQ/FE Requirement Document*
  - *The **ROD** board receives the data from the **FEB** which after some processing are send to the **ROB***
  - ***Buffering of the FEB data:***
    - work with the maximum **LVL1 trigger rate (100 KHz)** without introducing extra dead time
  - ***Calculate Energy and Time for each cell***
    - use **optimal filtering** algorithms
    - evaluate a **quality flag** for the pulse shape ($\chi 2$)
    - For most of the channels the **raw data** are rejected
  - ***Data monitoring** during physics runs*
  - *Do the **first pass in the analysis of the calibration data***
    - the complete analysis will be done in the local CPU of the ROD crate

THE ROD Module (II)

- **More requirements**:
  - *Send the **raw data** information*
    - in the case of interesting events, or
    - for large energy depositions in a cell, or
    - for debugging purposes
  - *Apply a **corrections to the Energy or Time** estimators*
    - for example:
      - *for the non-linearities in the shaper or in the ADC*
  - *Monitor the **Minimum Bias** and **pile-up noise***
  - *Keep the possibility for **special runs at reduced rate***

THE ROD Module (III): LiArg ROD and Tical ROD baseline differences





	ROD LiARG	ROD TiCal Baseline
Input links (32 bits @ 40 MHz)	**2**	**4**
Number of channels per board	**256**	**154** (2•46b+2•31eb)
Number of DSP Processing Units	**4**	4
Number of channels / DSP PU	**64**	**46b** or **31eb**
Output links (800 Mb/s)	**1**	**1** (expected 1,14Gbits/sec)

THE ROD Module (IV): Adapting LiArg ROD demonstrator board to Tiles

- Using the **same design reports**:
  - ***Low development time***. *Software implementation in a well tested HW.*
  - ***Low cost Motherboards***. *If we order a production in conjunction with LiArg the total cost will be decreased because of a high number of units that LiArg needs.*
  - ***A Synergy solution***: *Collaboration with LiArg people.*

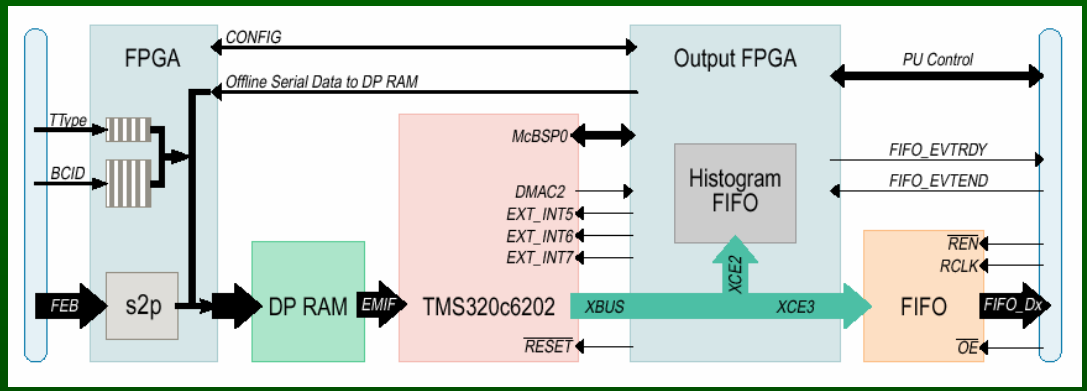
- We need to **reconsider**:
  - *Data **input/output format** and rates*
    - **LAr**: 2 inputs/1 output
    - **TileCal**: 4 inputs/1 output
    - MODIFY THE TRANSITION MODULE: **TM4Plus1** under construction
  - *Processing power*
    - Maybe it **is not necessary as much power as Lar (TileCal has less channels/board)** => software redesigning of RODPU adapted to TileCal needs.

The ROD Demonstrator Board

- **ROD Board Characteristics**:
  - ***9U VME64x Slave***
  - ***Four Processing Units (PU)***
    - treats **46(B)/31(EB) channels**
    - has one C6202 DSPs, Fifos, FPGAs, and dual port memory.
    - large buffering needed to absorb the processing time fluctuations of the events
  - ***16 MB output buffer for event monitoring and spying***
    - possible to store complete events (test beam) fully controlled by VME

DSP Processing Units	**4**
Mezzanine board dimensions	**85 × 185 mm**
Number of channels in the board	**256**
Number of channels per PU	**64**
Inputs	**4**
FEB links	**2** (32-bits @ 40 MHz + clock)
TTC input	**CLCK, BCRST, BCID, EVID, EVCRST, Ttype, Reset…**
Outputs	**1**
ROB links	**1.28 Gbit/s** (32-bits @ 40 MHz)
VME bus	Full readout and board control (**VME64x slave**)
BUSY signal	**Asynchronous**

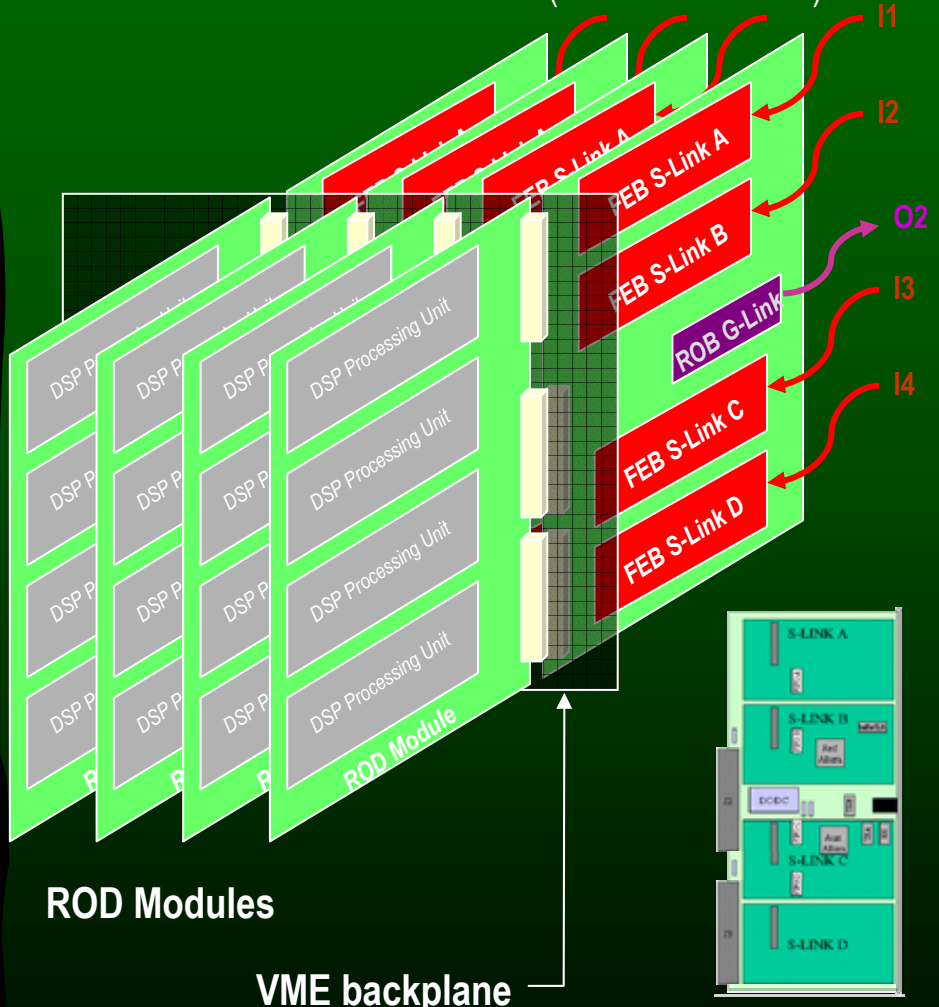
The Processing Unit



- **DSP**: **250 MHz TMS320C6202** DSP with **256K** Bytes of internal program memory and **128K** Bytes of data memory.
- **Input FPGA**: receive the **FEB data** and **TTC information**. Does **data format consistency** checking, and some data **rearrangement**.
- **Dual Port Memory**: The **events are then buffered this memory**, available to the DSP as read-only external asynchronous RAM. Also serves for **initializing the DSP internal program and data memory**, **at reset**.
- **Output FPGA**: After event processing, the **DSP writes the output data** in the final format, to the **output FIFO**, ready to be read by the **OC** of **ROD motherboard**. The output FPGA is responsible for **booting** the PUboard by **configuring** the input FPGA, then loading the DSP code. It also provides a **second FIFO (for histogramming purposes)** which can be written to by the DSP, and read from by the ROD.
- **Upgrade path:**
  - *Replace the **C6202** by the **C6203** running at **300 MHz***
  - *Upgrade to **faster dual- port memory** and **output FIFO**.*

The Transition Module TM4Plus1 (I)

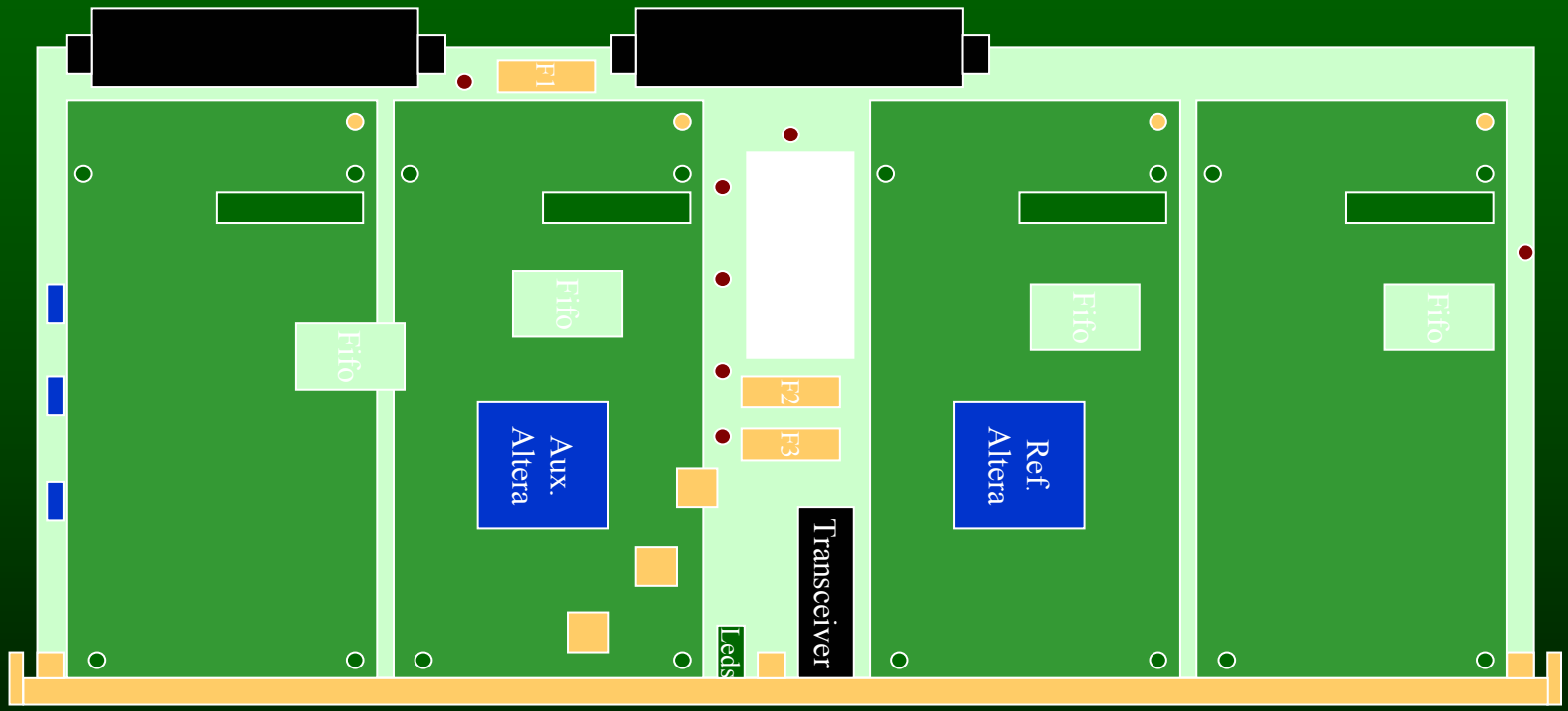
Transition Modules (Back of VME crate)



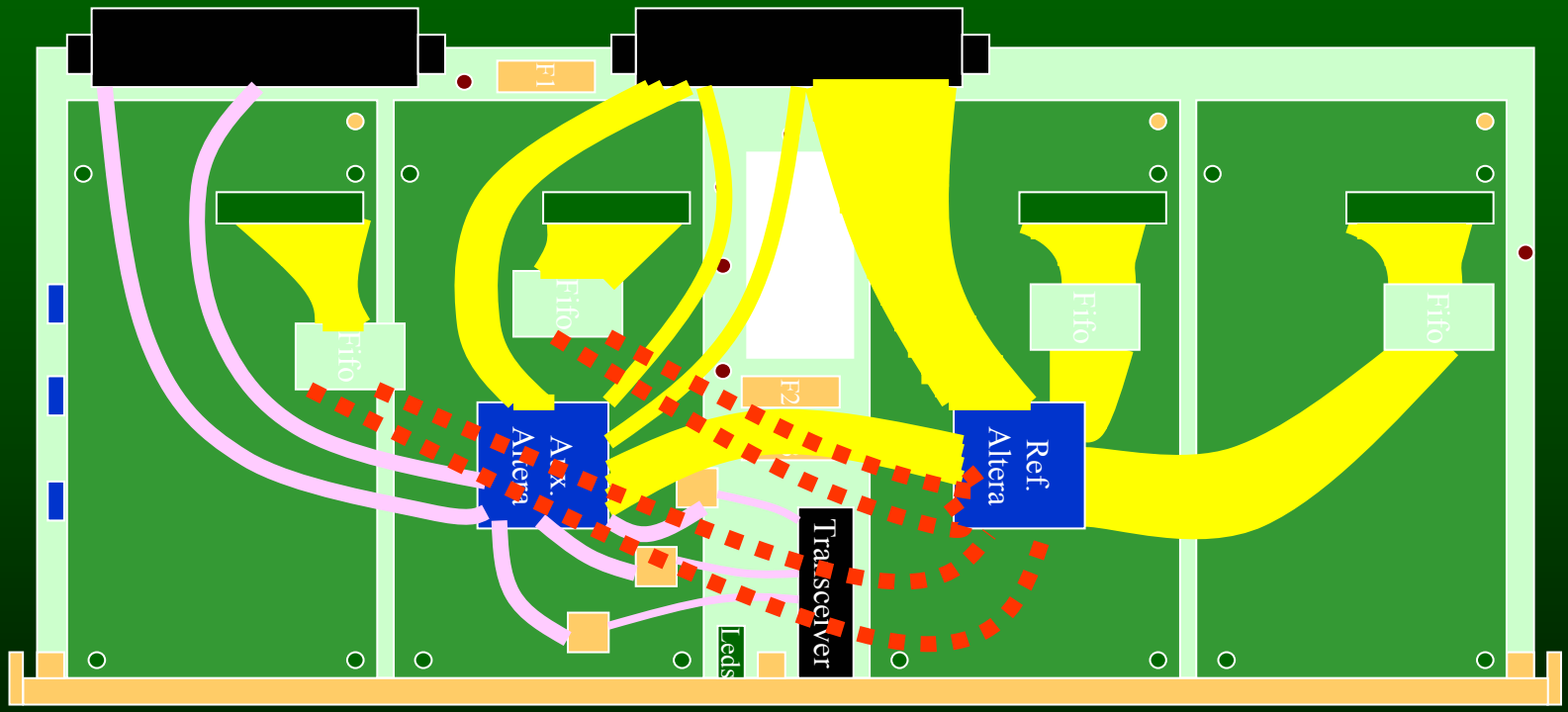
Transition Module Characteristics:

- **Four** FEB plug-in inputs **S-link (LDC)**: PMC for testing different input tecnologies
  - *4×32bits @ 40 MHz*
  - *~200Mb/sec @ 100 KHz L1 rate*
- **One** Integrated (due to space) **SLINK** output **LSC**
  - *32 bits @ 40 MHz*
- **4 KWord** buffer for each input (**FIFOs**)
- **ALTERA logic**: **Programmable**
  - ***Reformatting Altera:data multiplexing*** *and* ***S-LINK control.*** *Reformats and merges incoming data (4→2)*
  - ***Auxiliary Altera: Integrated ODIN output code*** *(not designed yet). There are free logic cells for use in conjunction with Altera Ref.*

The Transition Module TM4Plus1 (II)

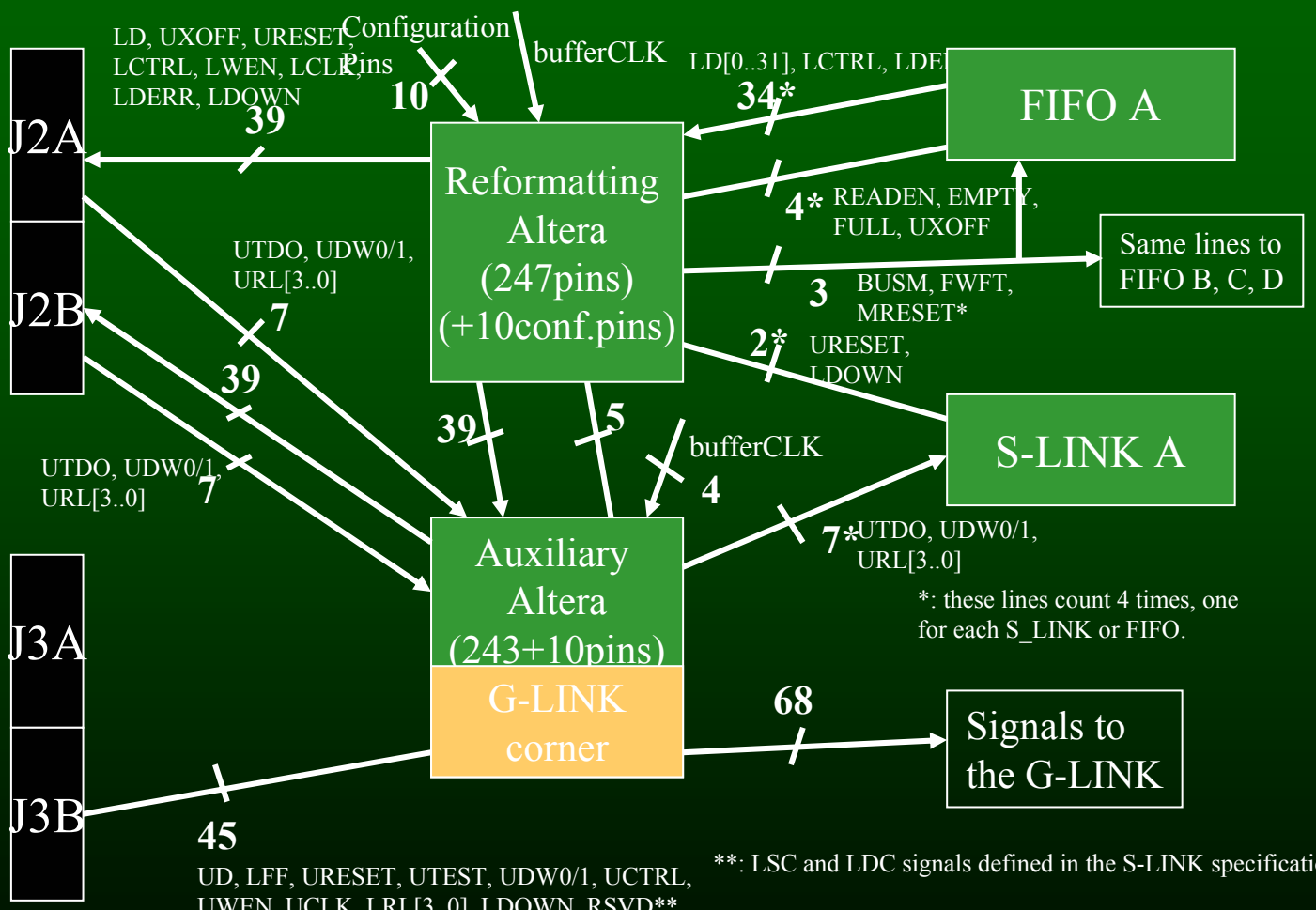


The Transition Module TM4Plus1 (II)



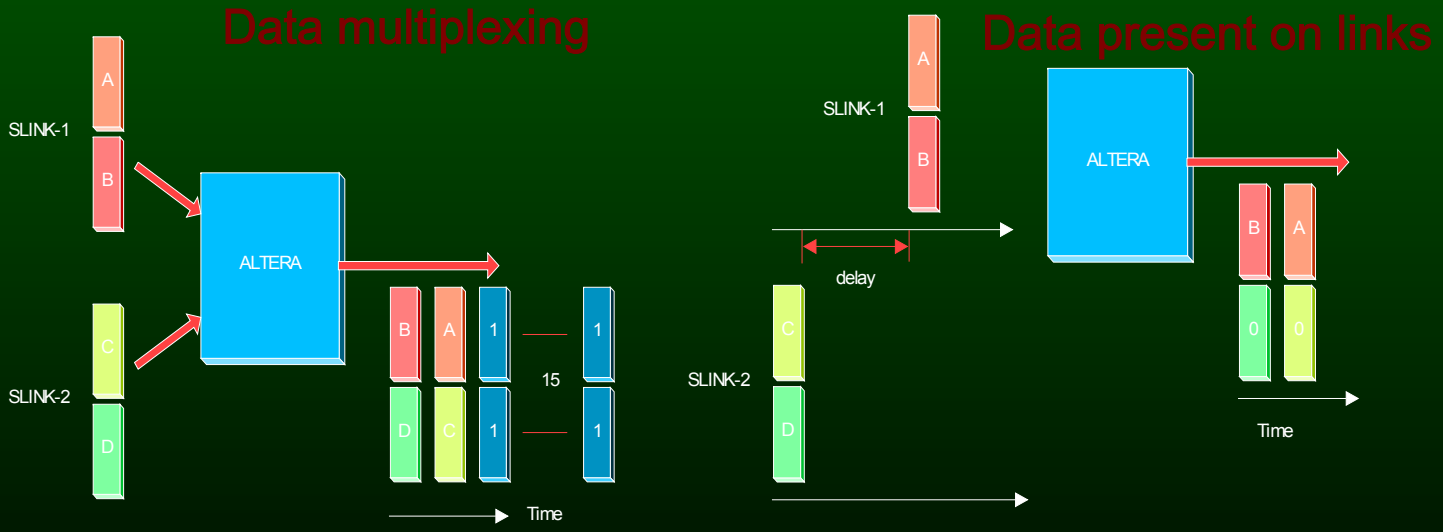
The Transition Module TM4Plus1 (III)

Signal mapping

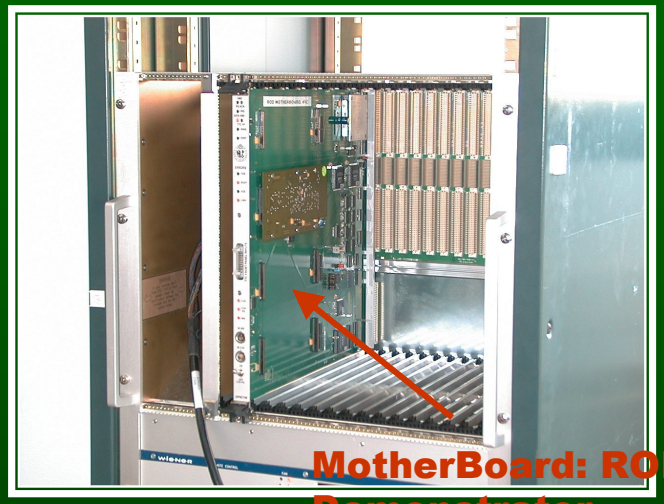


The Transition Module TM4Plus1 (IV)

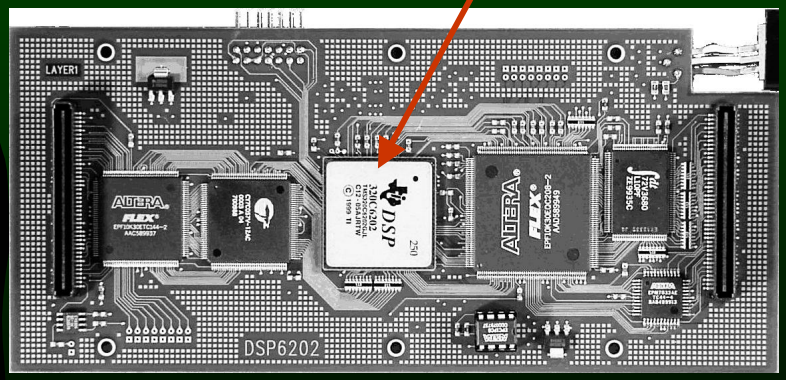
- Altera logic: SLINK Control
  - ***LAr** provides a **VALID** signal to **PU**. Relies on **zero data value** meaning **no data***
  - *Options*
    - **Emulate this functioning**
      - *Control done at the transition module. Mark different events in a header*
    - **Input SLINK control signals** (LDOWN, LCTRL, ...) **into the ALTERA** in the 9U module to generate **VALID** => different signal for each PU.

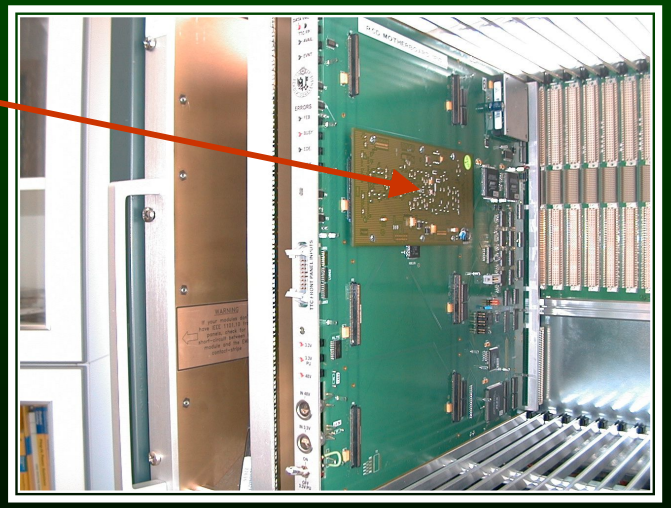


Some Photos: ROD Laboratory at Valencia



DSP Processing Unit



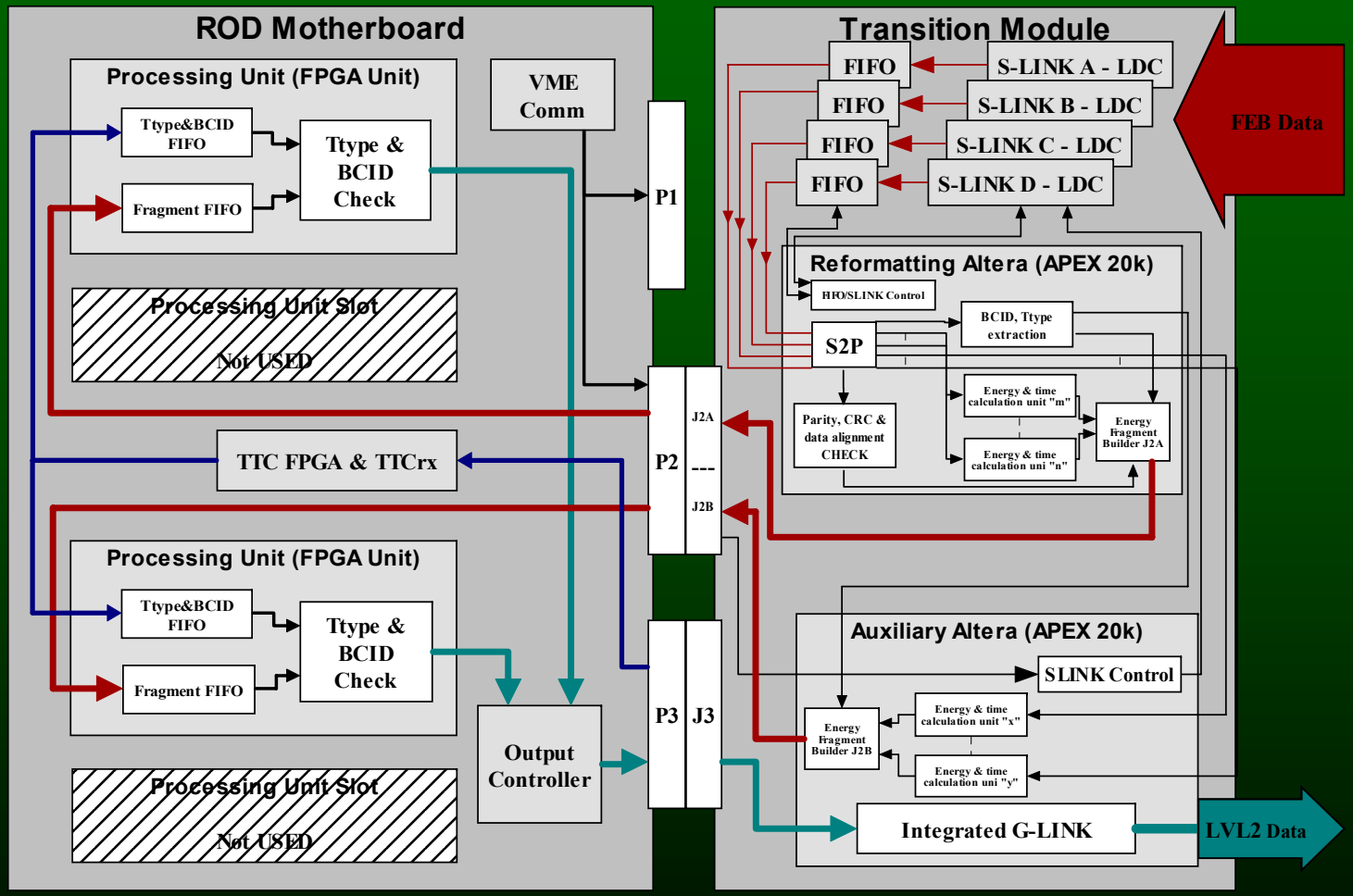


LEB 2001, Seventh LHC Electronics Workshop, Stockholm, Sweden

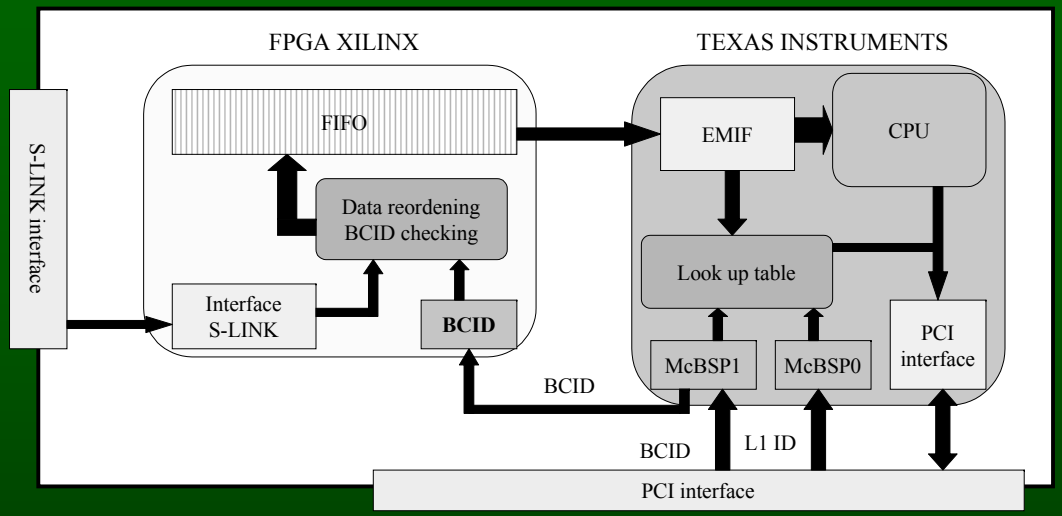
Actual Developments: FPGA Processing Unit (I)

- Motivations:
  - *Build a simple FPGA processing unit and do a preprocessing of the data at transition module level.*
- This technology will provide:
  - ***Parallelism**: We could implement "n" units processing "m" channels each. A CPU(DSP) solution could only do 8 parallel instructions/cycle if it's optimized (DSP62xx – 8 ALUs). The **efficiency** of the FPGA will be increased with the number of parallel units*
  - ***Lower Cost**: only an FPGA with a FIFO is needed (no high cost DSP and dual-port memory)*

Actual Developments: FPGA Processing Unit (II)



Actual developments:
PMC DSP unit for testing algorithms



- DSP TMS320C6205: 64kbyte program memory, PCI master/slave interface revision 2.2, two serial channels to load TTC data (BCID, EventID, Ttype).
- Xilinx FPGA X2CS100 Implements:
  - *S-Link interface for the input data (FEB)*
  - *Data reordering*
  - *BCID check*
  - *Communication with EMIF*

Work plans

- **Hardware**:
  - *Test the **ROD Demonstrator Motherboard** and **PUs (finished)***
  - *Test the Transition Module **TM4Plus1 (not yet finished)***
  - *General purpose **DSP** based PMC for real time data processing in HEP experiments. **(prototyping)***
  - *Design of a custom **FPGA processing unit (starting)***

- **Software**: **Adapt** the ROD Demonstrator Board to tilecal's needs.
  - ***Transition module (FPGAs)**: we can do **flexible electronic circuits** into these devices to be compliant with with Tilecal **(working)***
  - ***DSPs**: build the tilecal's **optimal filtering algorithms (studying coefficients for the best resolution and SNR)** and program the **FLEX Altera input FPGA. (working)***
  - ***CPU (ROD Controller):***
    - **Adapting the LiARG ROD software libraries** to our System Controller **BIT3** (custom bit3 linux driver **IFIC1.1**). ***(done)***
    - **TileCal ROD Integration with DAQ-1:** Integration of **Local ROD VME software, online software**, and **ROS** Dataflow. ***(starting soon)***