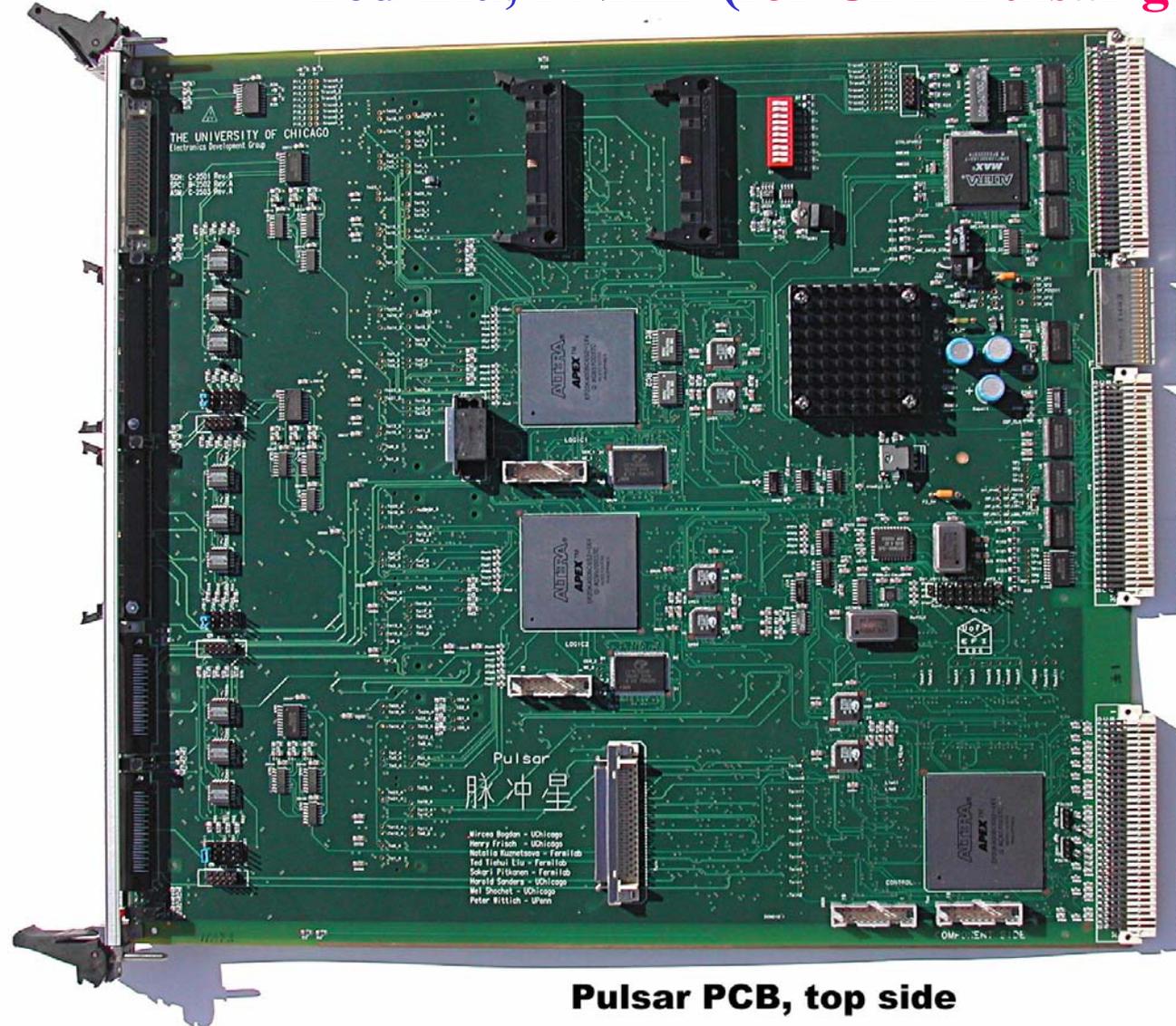


Using Pulsar as an upgrade for L2 decision crate

Ted Liu, FNAL (for CDF Pulsar group)



Pulsar PCB, top side

For more information about Pulsar board:

<http://hep.uchicago.edu/~thliu/projects/Pulsar/>

Back to Basic:

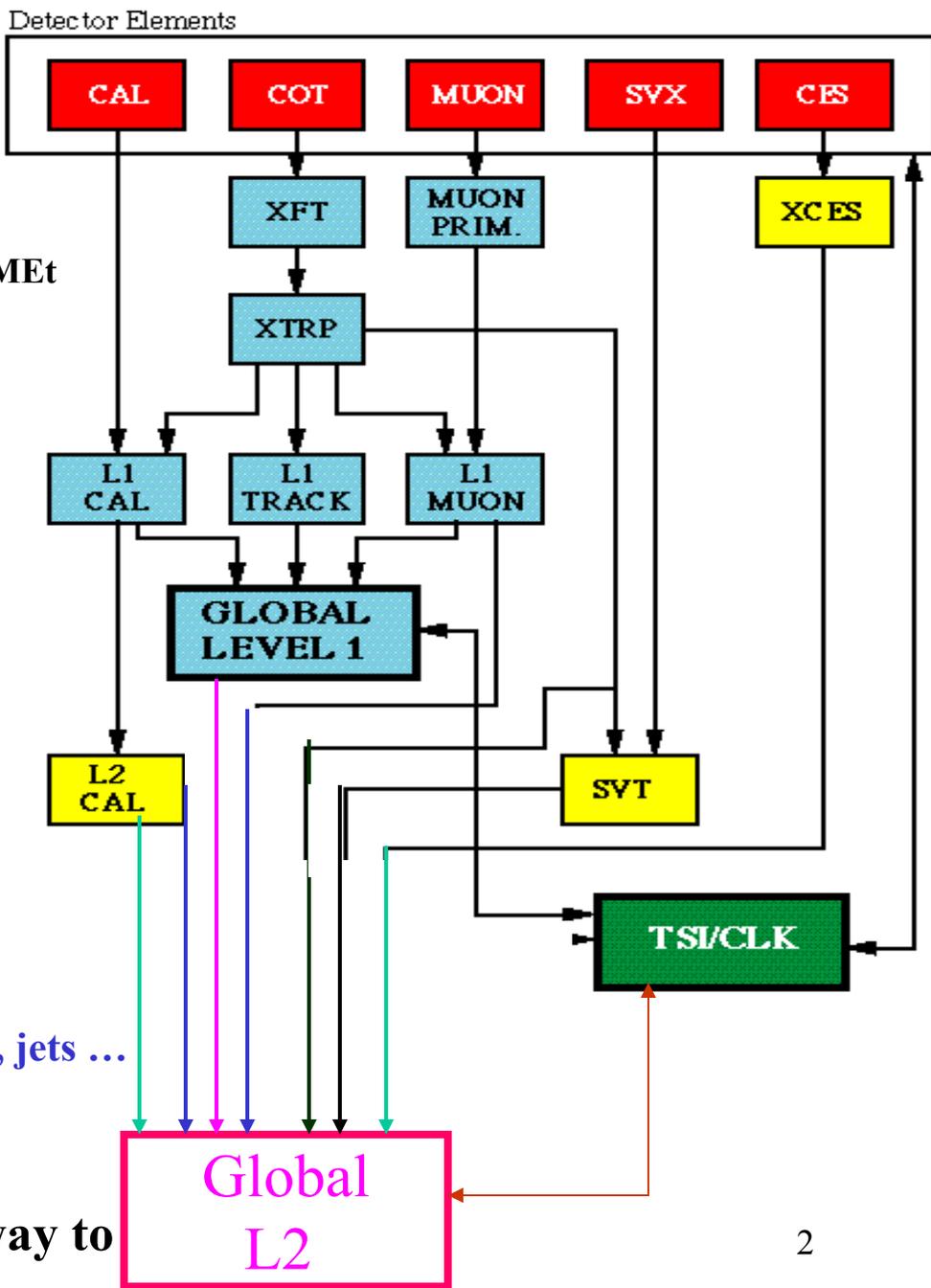
What does Global L2 really do?



make decisions based on
objects already found upstream

	L1	trk	svt	clist	Iso	reces	mu	SumEt,MEt
Tracks	●	●	●					
Jets	●	●	●	●				
electron	●	●	●	●	●	●		
photon	●			●	●	●		
muon	●	●	●				●	
Taus	●	●			●			
Met	●							●
SumEt	●							●
...								

physics object examples



- Combines/matches trigger objects into e, muon, jets ...
- Count physics objects above thresholds, or,
- Cut on kinematics quantities

Technical requirement: need a **FAST** way to collect many data inputs...

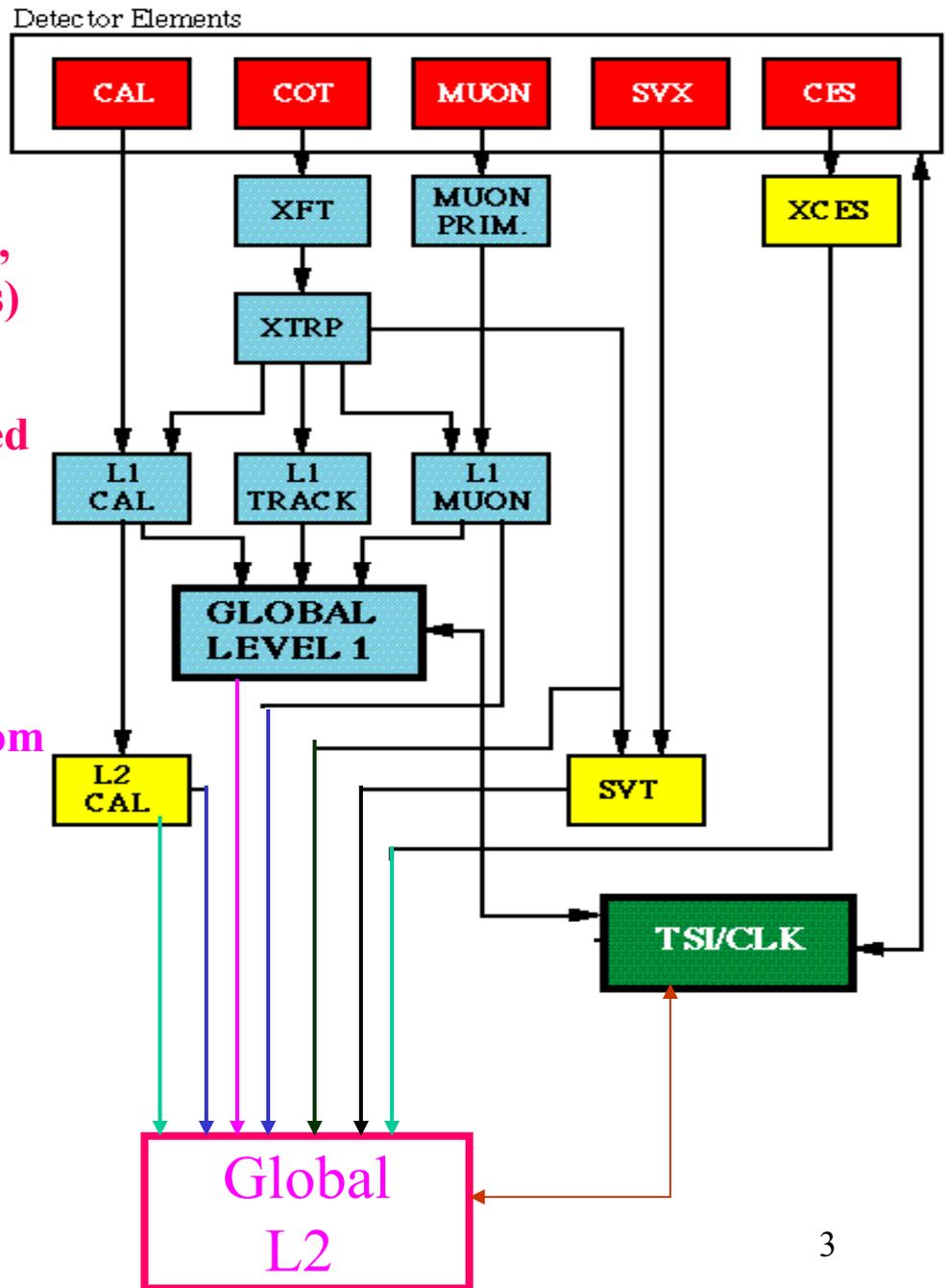
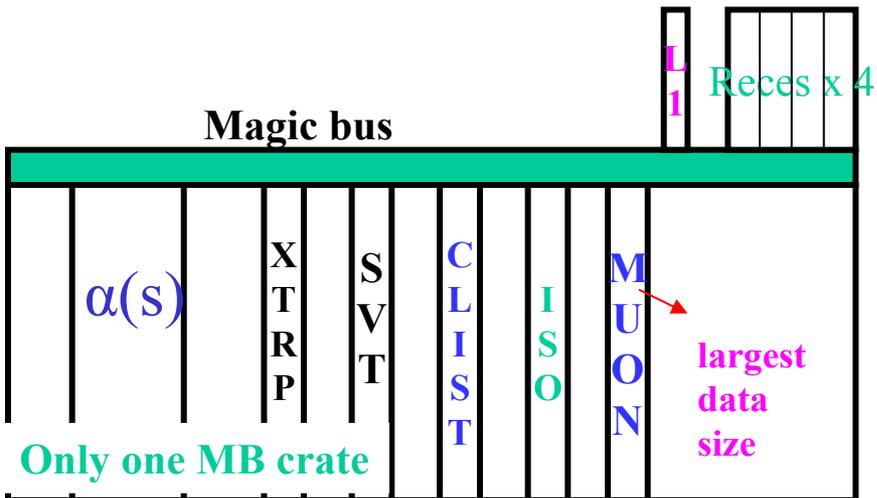
Current L2 Global Crate:

Technical requirement: need a **FAST** way to collect many **data inputs...**

→ With the technology available back then, had to design custom backplane (magicbus) and processor ...

→ data is either pushed(via DMA) or pulled (via Programed IO) to ease the bandwidth demands

→ In addition, one has to deal with the fact that each data input was implemented in a different way ... → 6 different type of custom interface boards



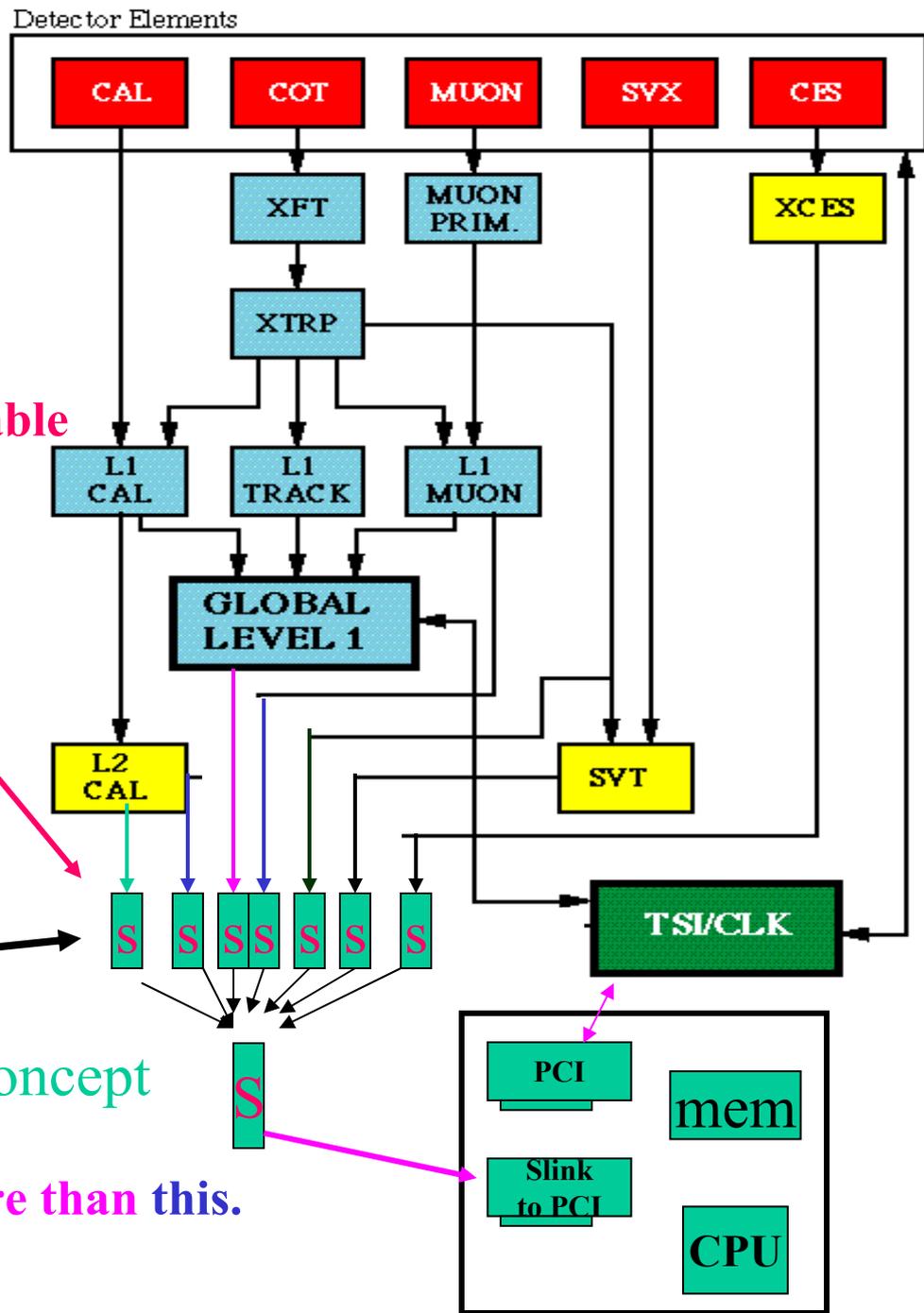
Basic idea on possible new approach using modern technology:

- **Convert/merge different data inputs into a standard link**
- **interface with commodity PC(s)**
- **high bandwidth, commercially available link to PCI interface cards:**

CERN S-LINK is designed just for this purpose for LHC and other experiments.

Real question: can we design an **universal** interface board (and the rest are all commercial products) ?

Pulsar is designed to be able to do **more than this**.



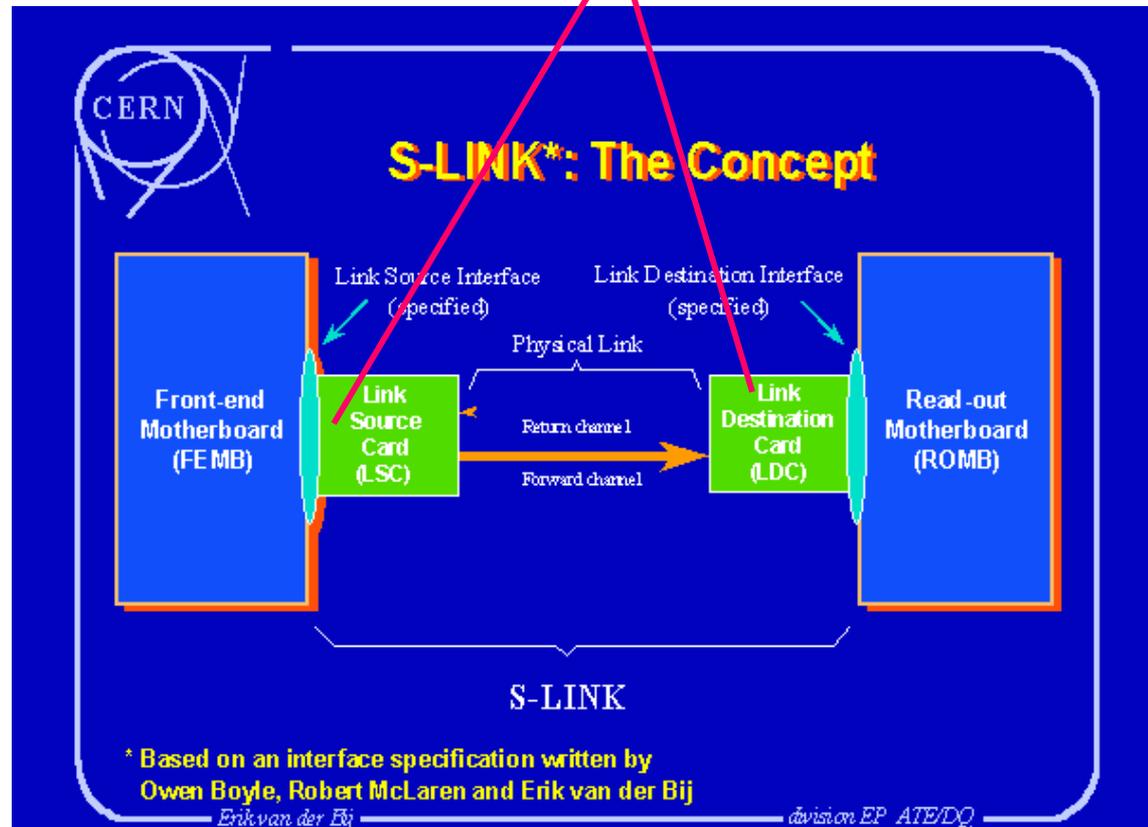
Concept

SLINK format example:

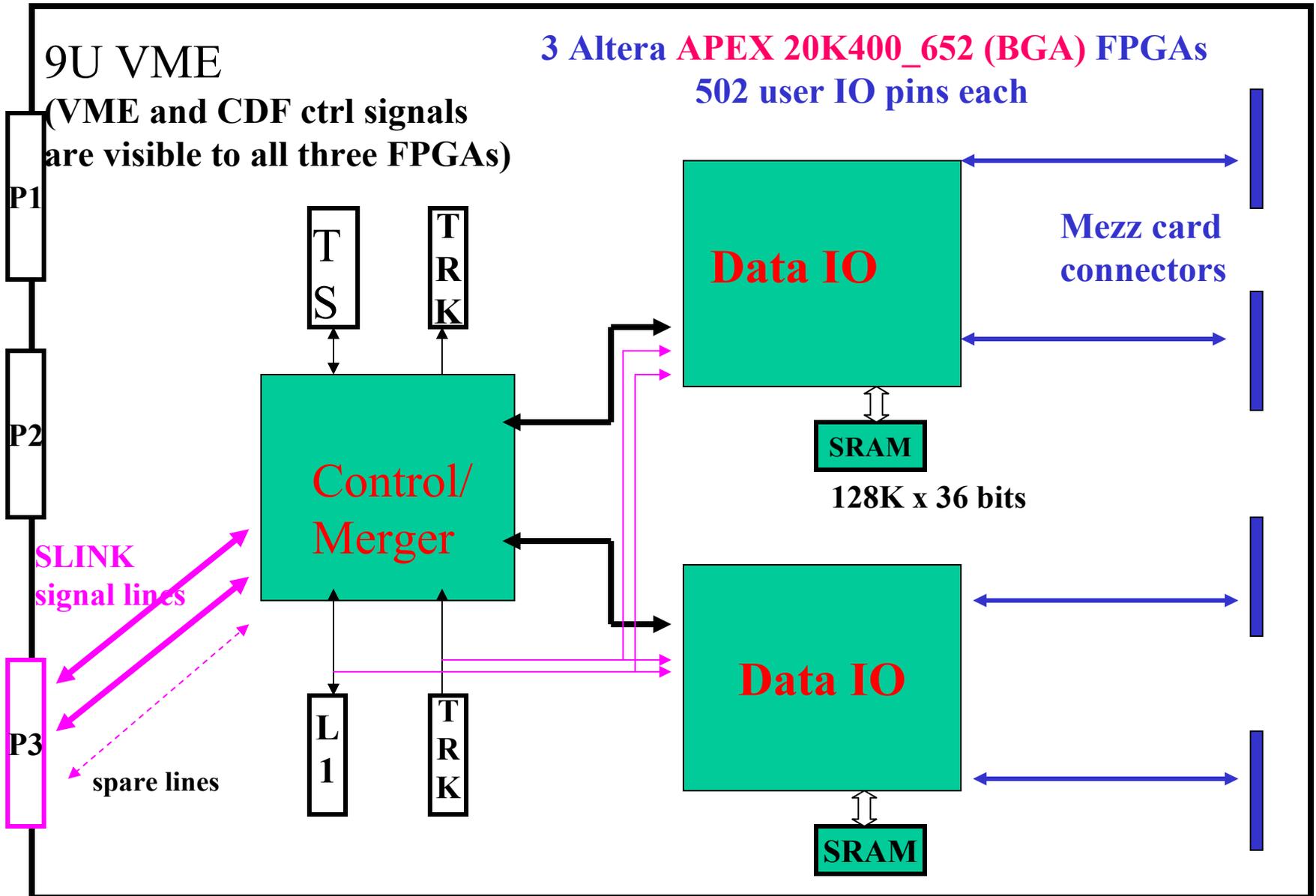
ATLAS SLINK data format

Beginning of Block control word
Start of Header Marker
Header Size
Format Version No.
Source Identifier
Level 1 ID
Bunch Crossing ID
Level 1 Trigger Type
Detector Event Type
Data or Status elements
Status or Data elements
Number of status elements
Number of data elements
Data/Status First Flag
End of Block control word

SLINK interface mezzanine card



Pulsar design



3 APEX20K400 FPGAs on board = 3 Million system gates/80KB RAM per board ⁶
2 128K x 36 pipelined SRAMs with No Bus Latency: 1 MB SRAM (~5ns access time)

Pulsar Design methodology

A major fraction of the design effort was dedicated to core firmware development and extensive design verifications by using state-of-the-art CAD tools:

- Leonardo Spectrum for VHDL synthesis;
- Quartus II for place and route, and FPGA level simulation;
→ developed core firmware (VHDL) first to guide and optimize the board design
- QuickSim for board and multi-board level simulation;
→ used to carefully verify board schematics
- Interconnect Synthesis tool for trace and cross talk analysis;
- IS_MultiBoard tool for signal integrity checks between motherboard and mezzanine cards;
→ used to carefully verify board layout

All details can be found at:

<http://hep.uchicago.edu/~thliu/projects/Pulsar/>

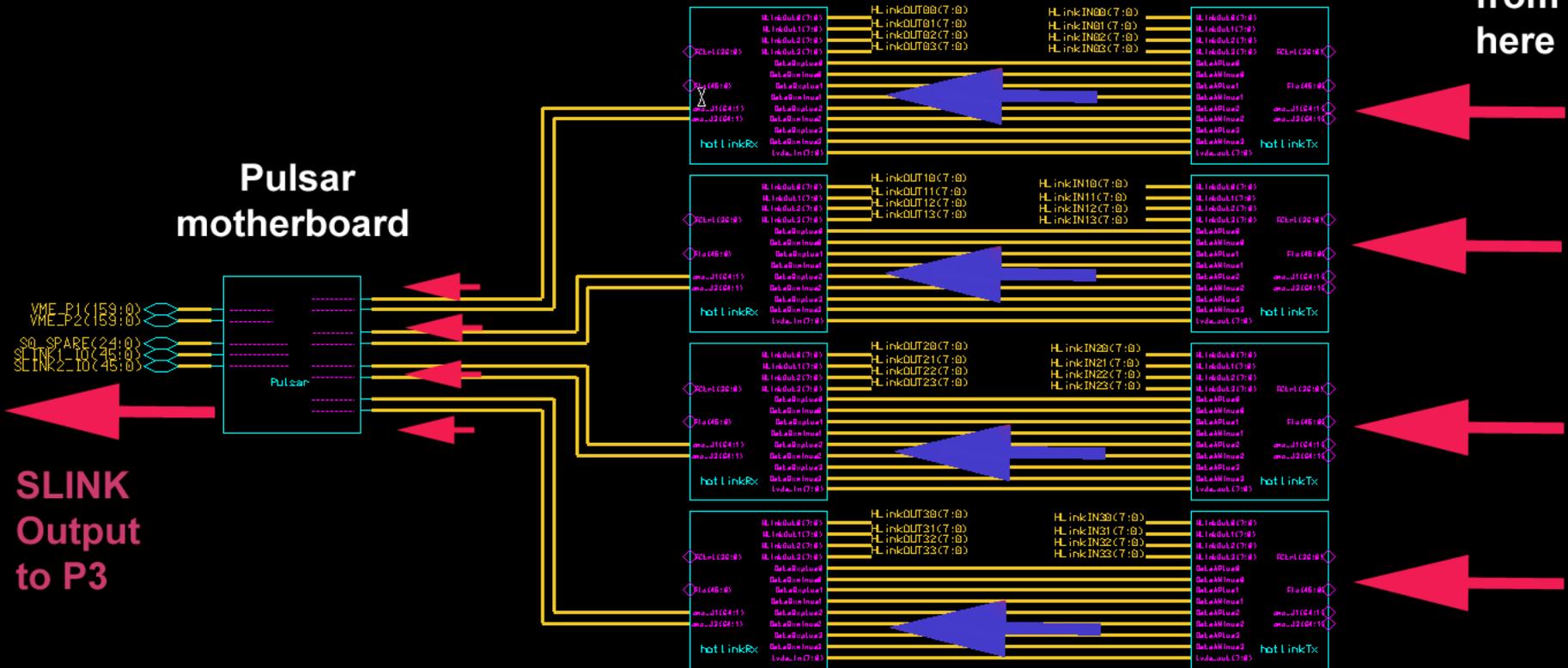
Performed multi-board simulation to verify the design

multi-board simulation

inputs
forced
from
here

Rx mezzanine

Tx mezzanine

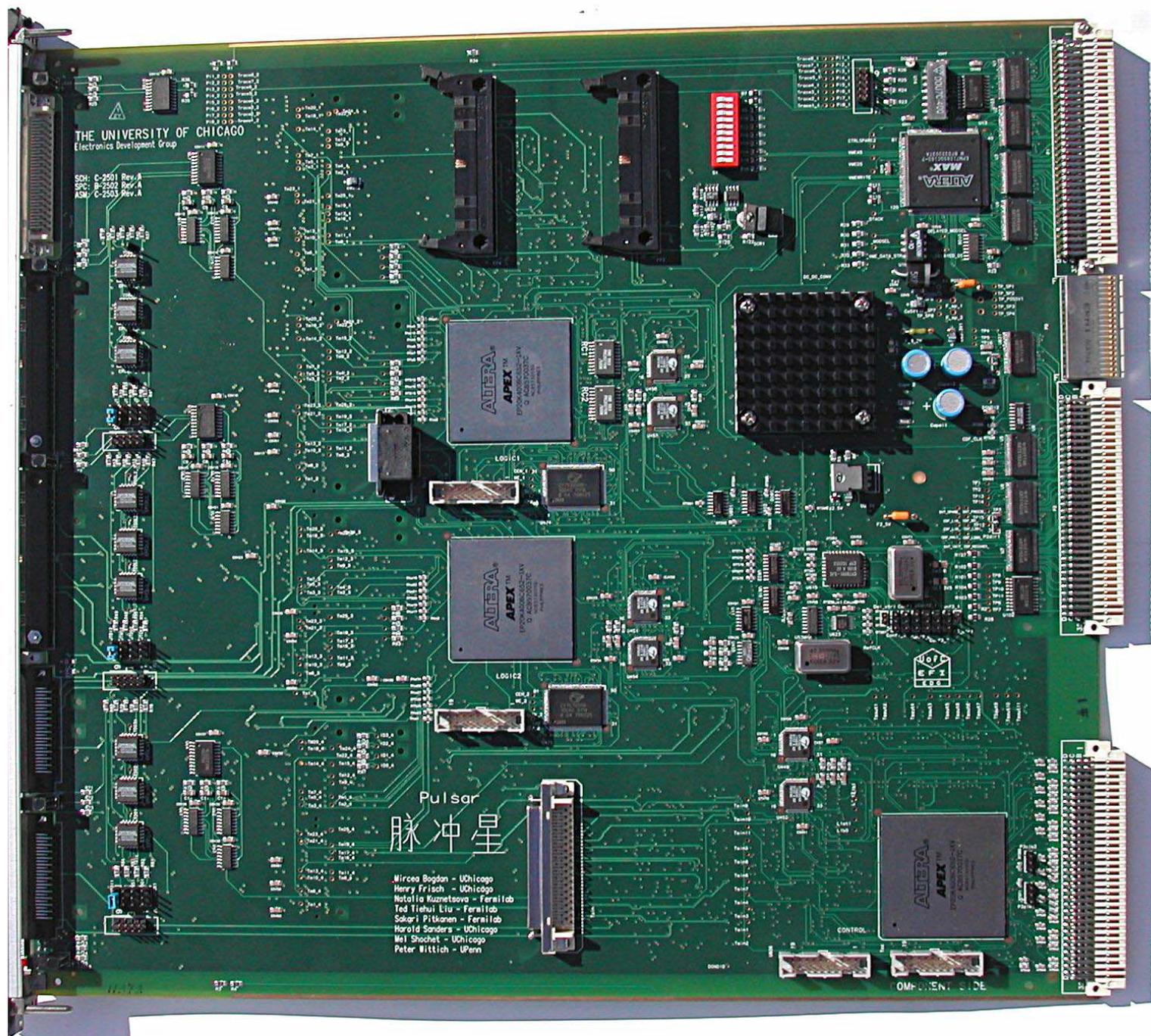


Pulsar and his eight daughters

Board
design work
started
Nov. 2001.

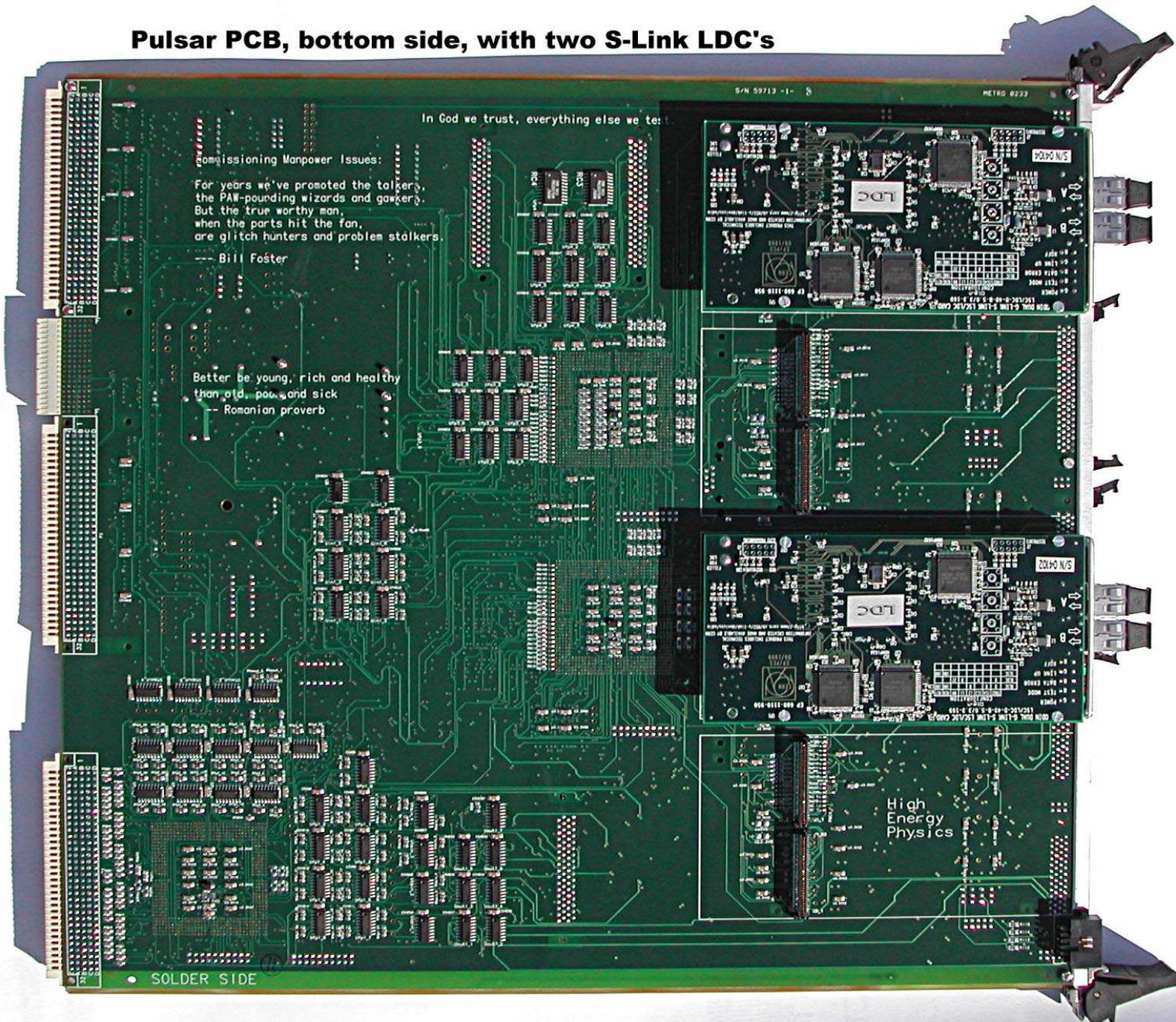
2 Pulsar
prototypes
received:
**Sept 19th,
2002.**

Initial VME
access to
all three
FPGAs
are
successful:
**Sept. 21th,
2002**



Pulsar PCB, top side

Pulsar PCB, bottom side, with two S-Link LDC's



Hotlink mezzanine card prototypes have been tested in July, 2002.

(for muon and cluster data paths)

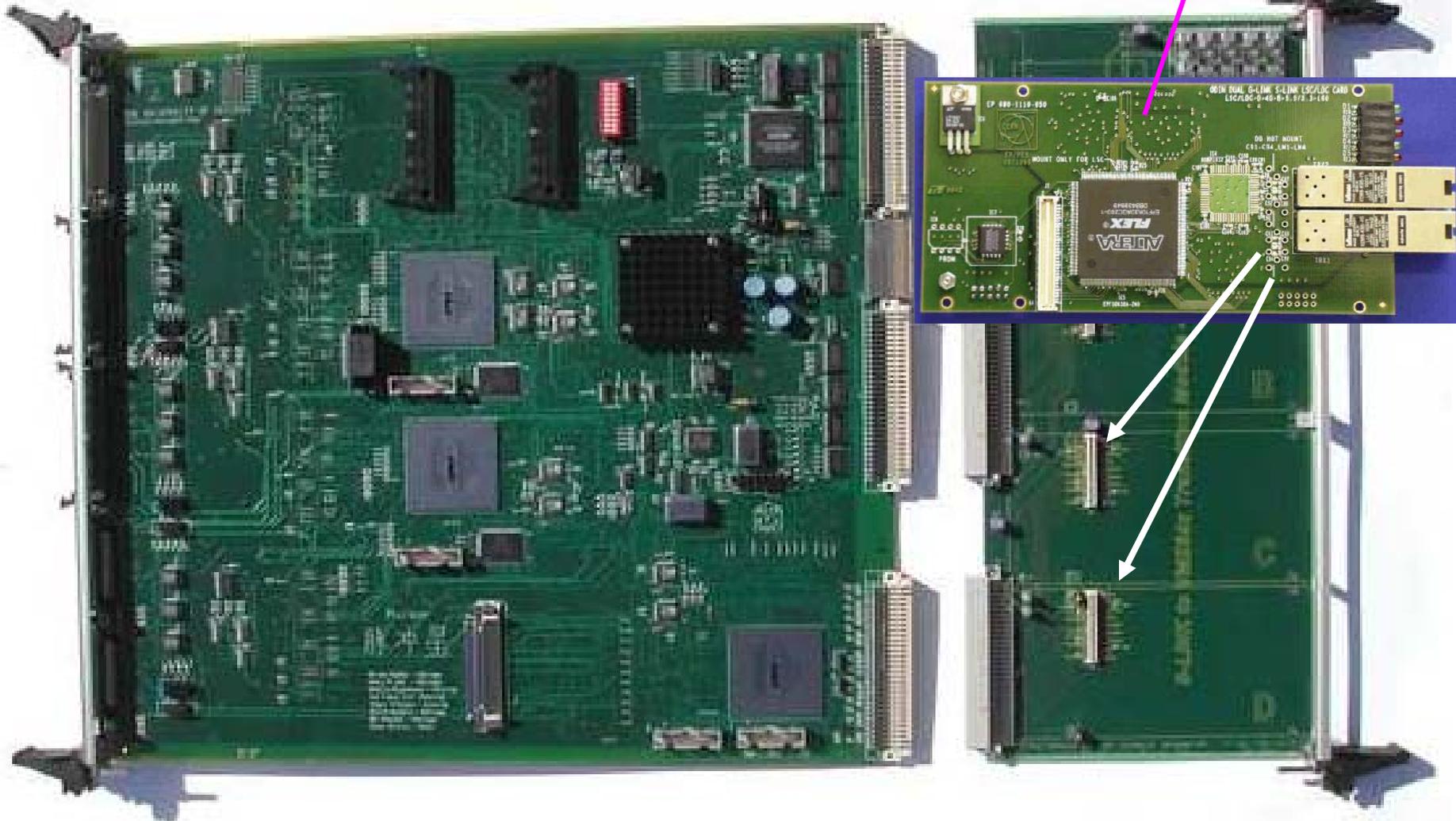
Hotlink Tx mezzanine card prototype



Hotlink Rx mezzanine card prototype

Pulsar with AUX card

SLINK
Mezzanine card

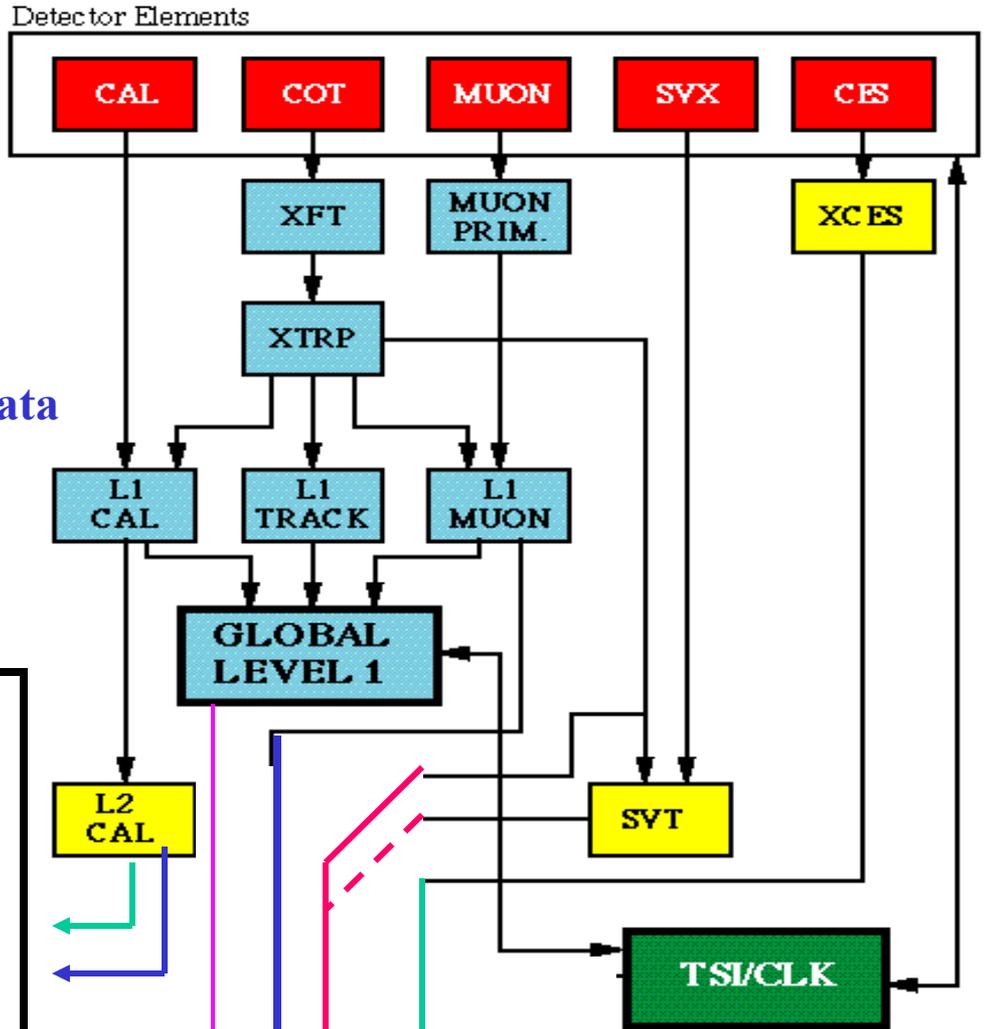
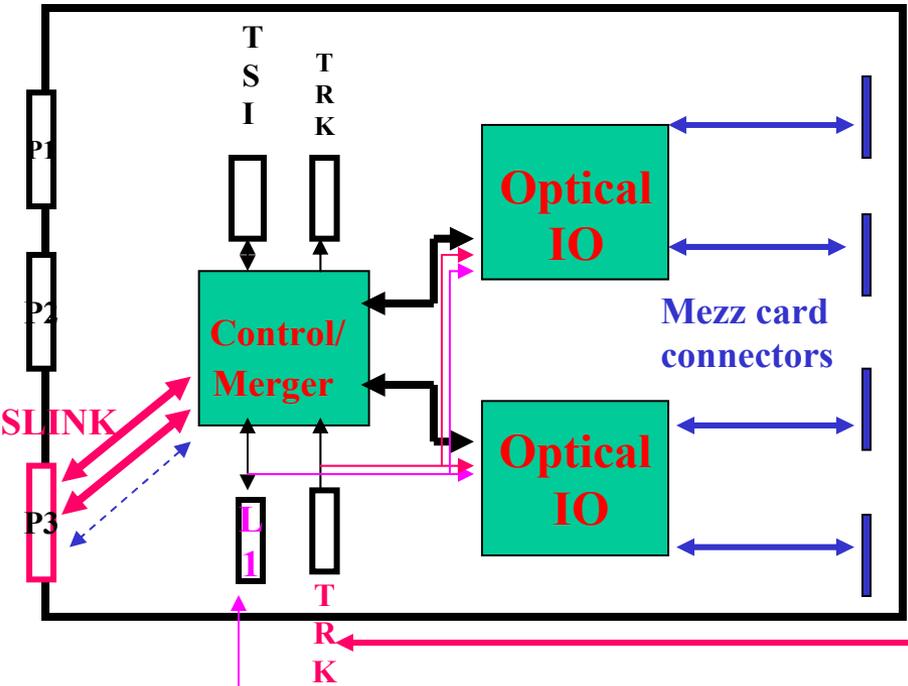


Pulsar approach:

Goal: only build **one type** of custom interface board and the rest are all commercial products.

Use mezzanine cards to take care optical data paths (Cluster, **Isolation**, Muon and **Reces**)

Pulsar design



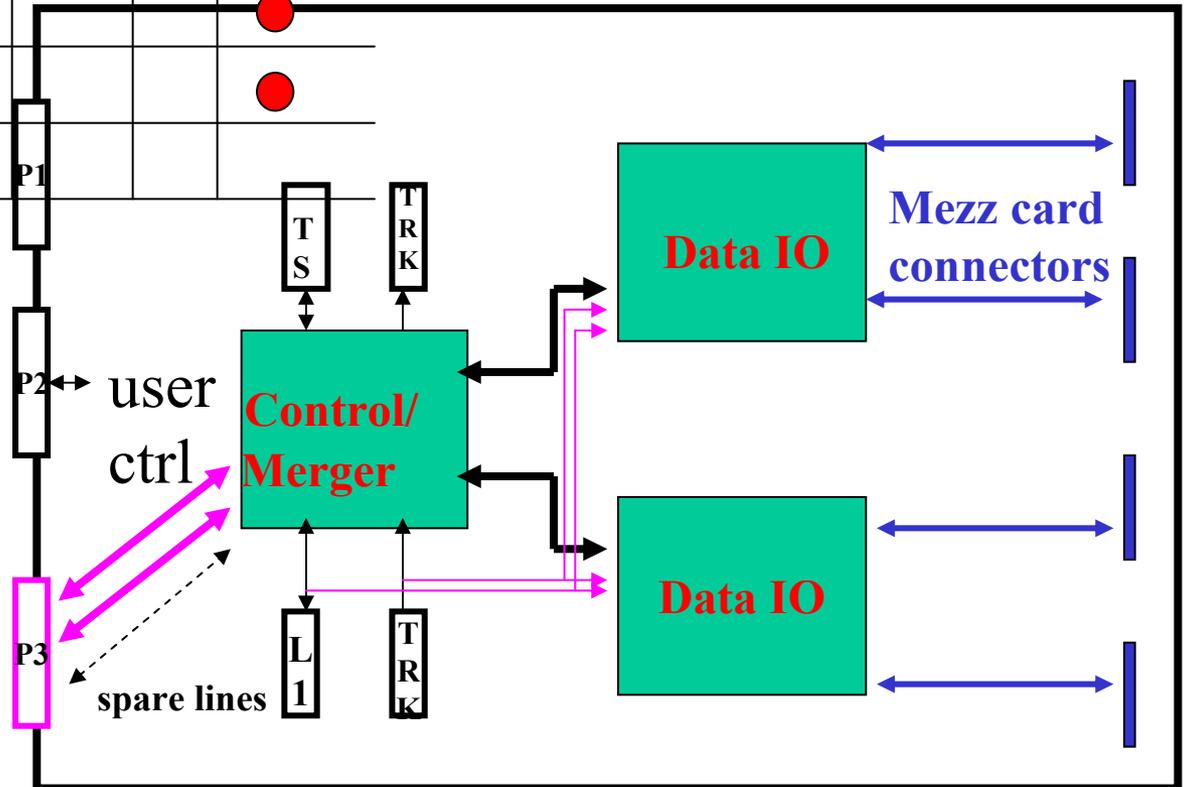
Extra features:
L1 trigger bits and **trk/svt input** are visible to all three FPGAs for flexibility...

	L1	trk	svt	clist	Iso	reces	mu	SumEt,MEt
Tracks	●	●	●					
Jets	●	●	●	●				
electron	●	●	●	●	●	●		
photon	●			●	●	●		
muon	●	●	●				●	
Taus	●	●			●			
Met	●							●
SumEt	●							●
...								

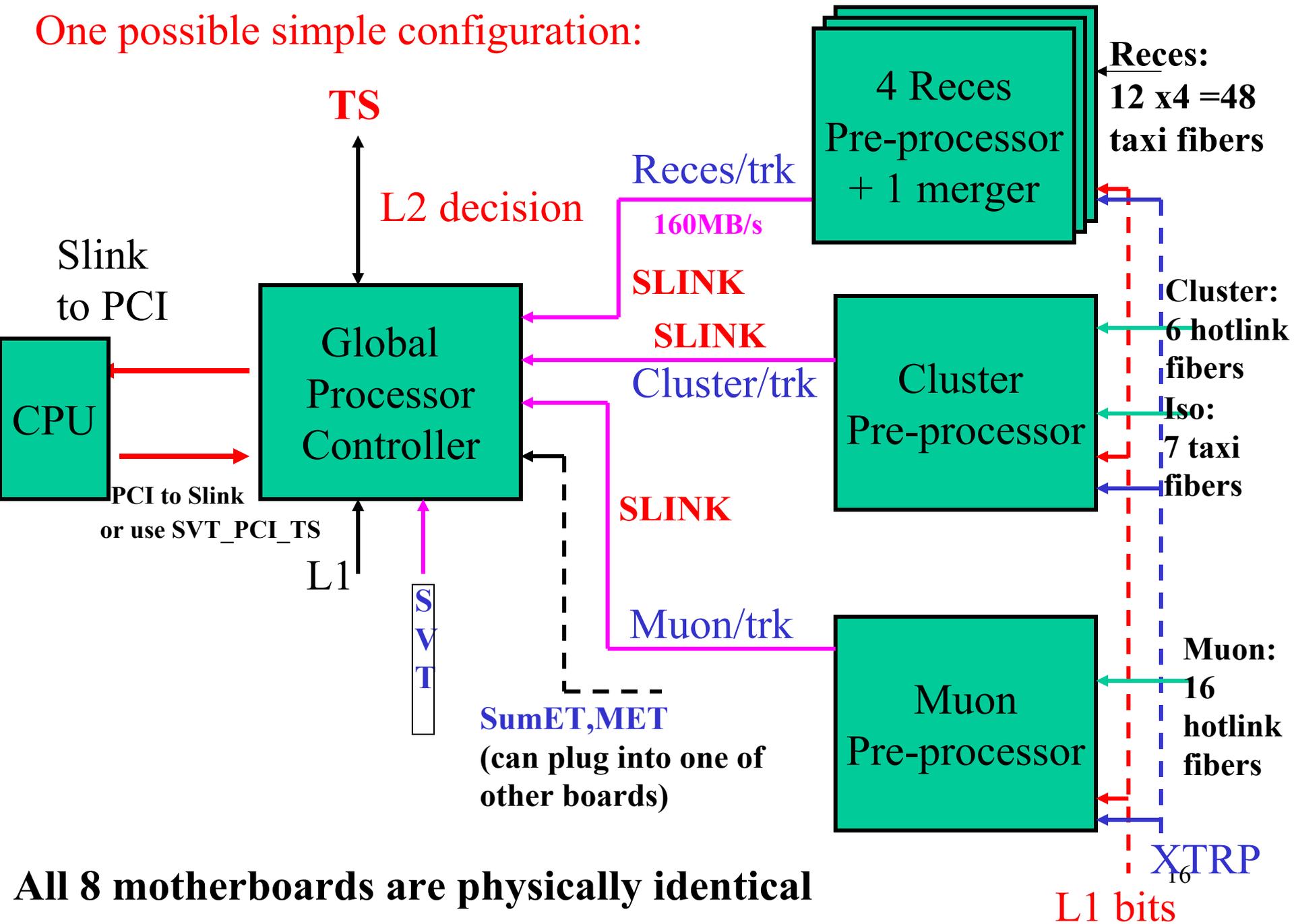
What does Level 2 really do?

- Combine/matches trigger objects into e,muon ...
- Count objects above thresholds, or,
- Cut on kinematics quantities

Most trigger objects need L1 and track/svt trigger information, this is reflected in Pulsar design: SLINK I/O for flexibility



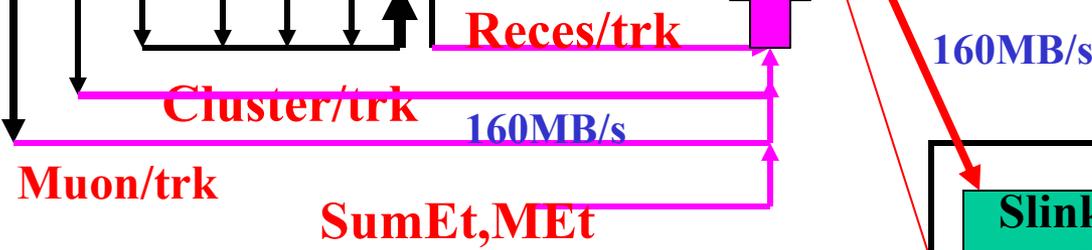
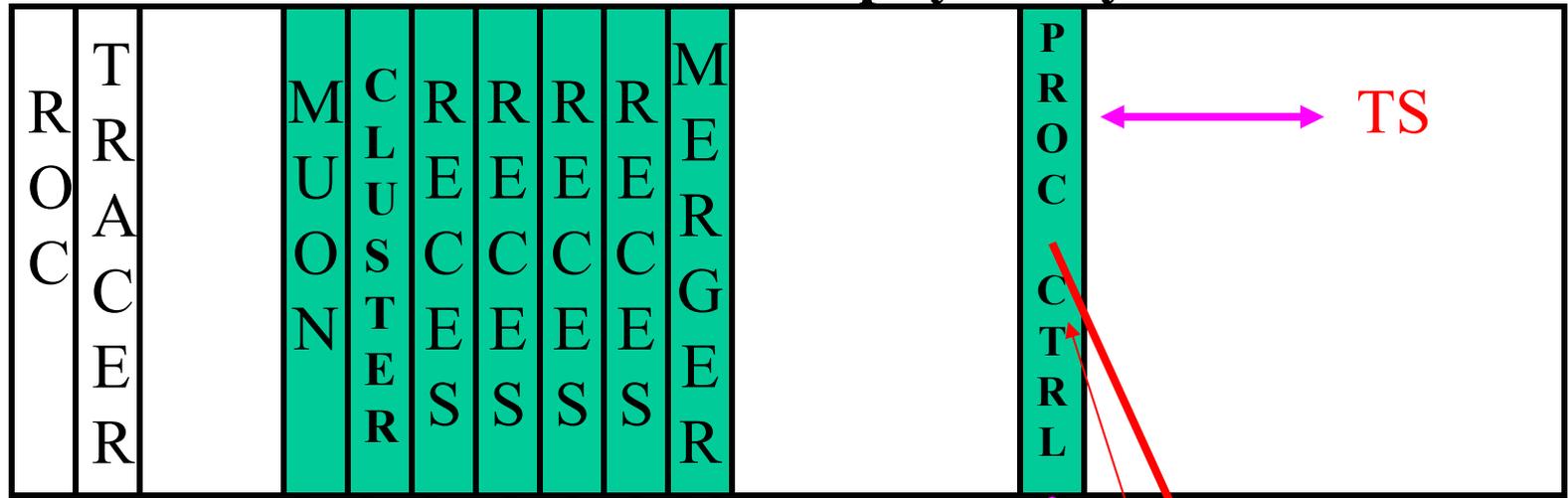
One possible simple configuration:



All 8 motherboards are physically identical

All 8 motherboards are physically identical

Possible
New L2
Decision
Crate



Each Pulsar board take two slots (due to mezzanine cards)
Total: 8 pulsar boards = 16 slots

Baseline design: use pre-processor Pulsars to simply suppress/organize data, use Processor Controller Pulsar to simply pass data to CPU via Slink to PCI and also handshake with TS. All trigger algorithm will be handled by CPU.

GHz PC or
VME processor

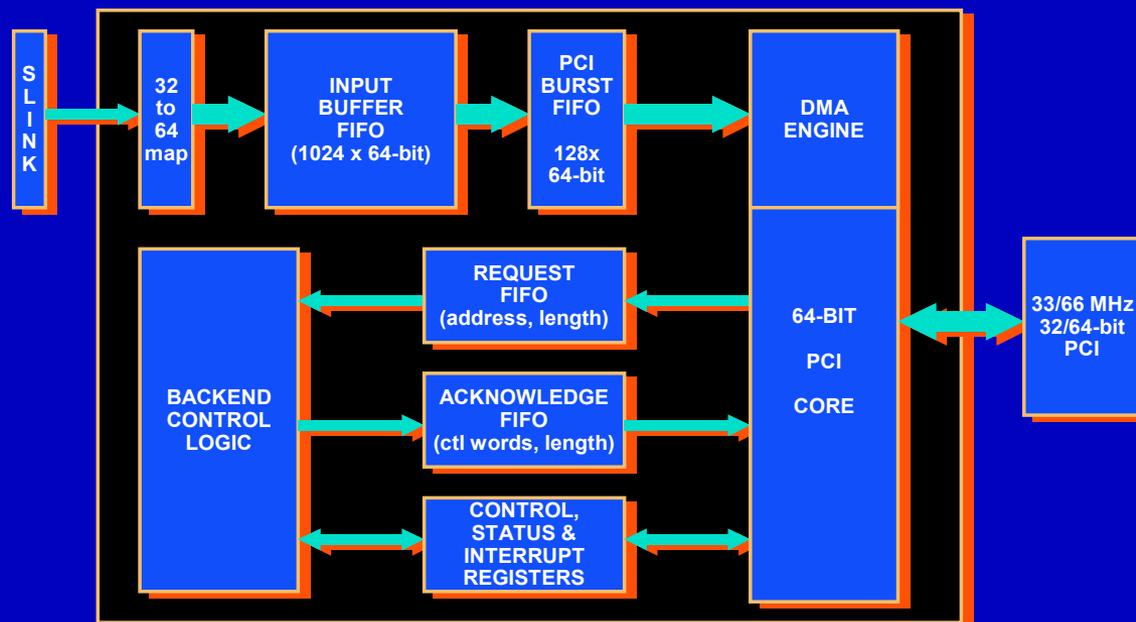
New 32-bit SLINK to 64 bit PCI interface card: S32PCI64



- highly autonomous data reception
- 32-bit SLINK, 64-bit PCI bus
- 33MHz and 66 MHz PCI clock speed
- up to **520MByte/s** raw bandwidth



S-LINK to PCI-64 interface



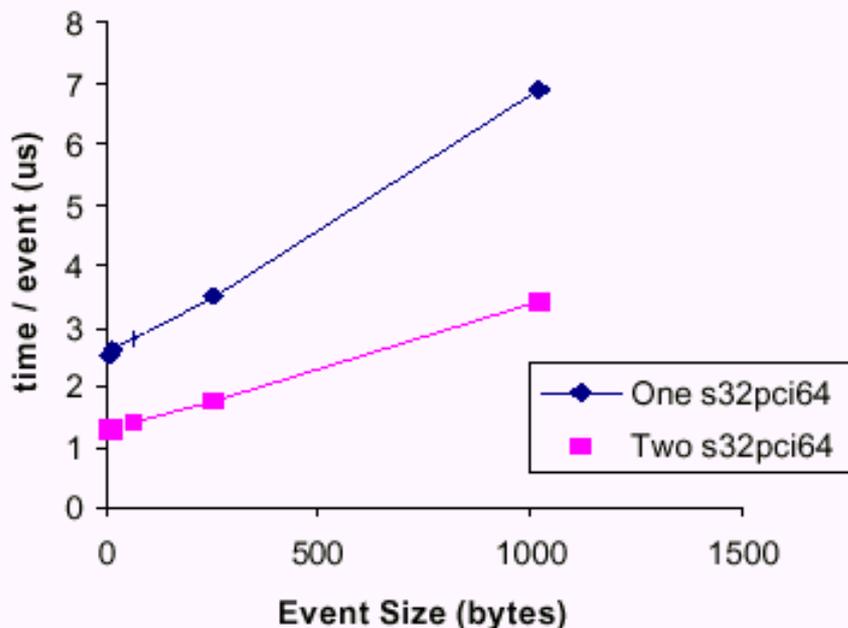
High-speed follow up of the Simple SLINK to PCI interface card

New 32-bit SLINK to 64 bit PCI interface card: S32PCI64

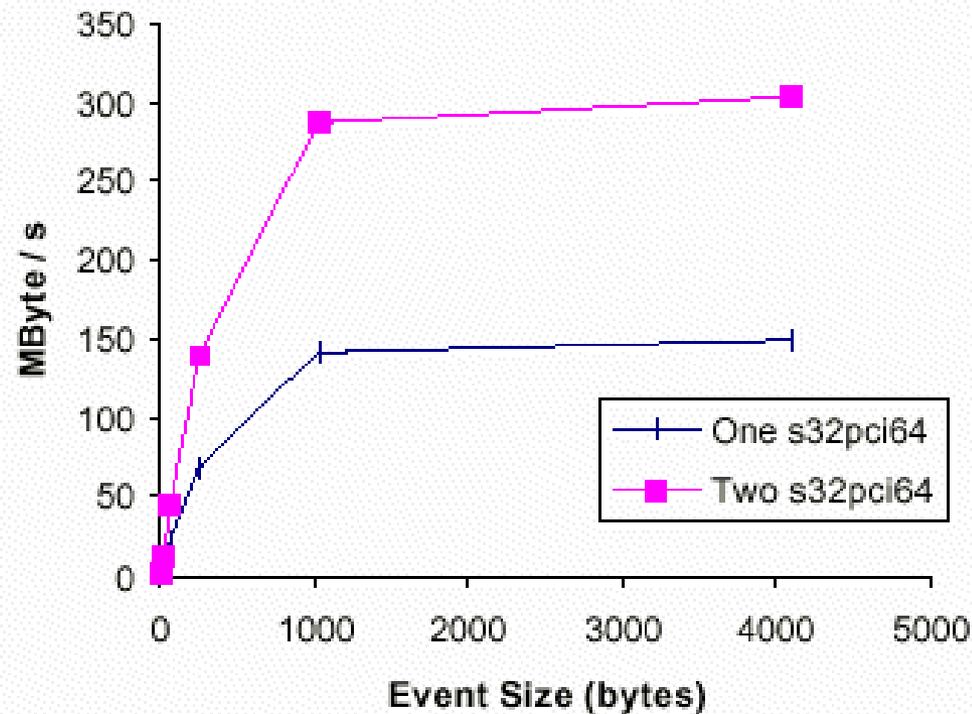


- highly autonomous data reception
- 32-bit SLINK, 64-bit PCI bus
- 33MHz and 66 MHz PCI clock speed
- up to **520MByte/s** raw bandwidth

S32PCI64 overheads



S32PCI64 throughput

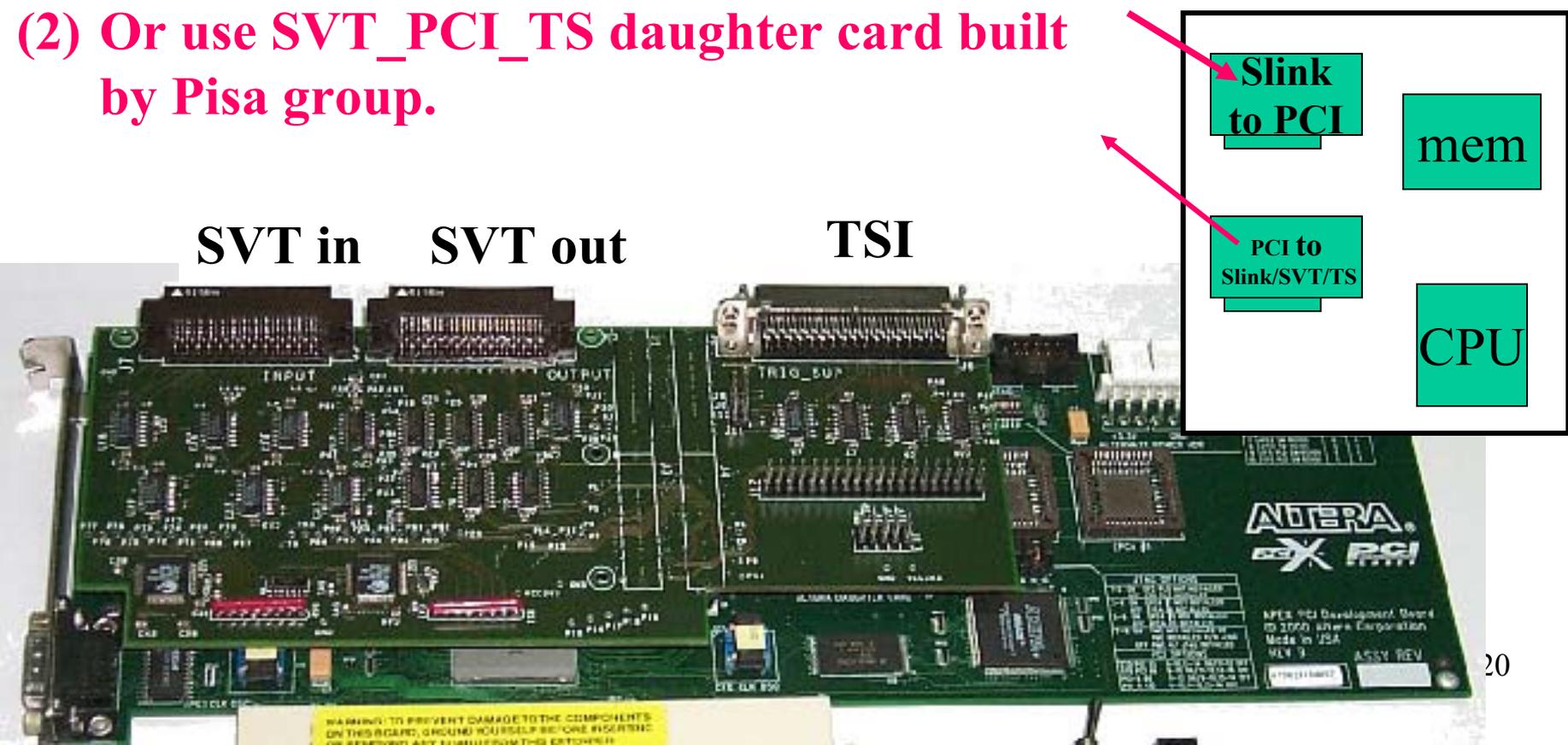


<http://hsi.web.cern.ch/HSI/s-link/devices/s32pci64/>

L2 decision from CPU: how to handshake with Trigger Supervisor?

There could be a few ways to achieve this:

- (1) could use PCI to SLINK card: send a SLINK message back to Pulsar Processor Controller, then Pulsar handshakes with TS;
- (2) Or use SVT_PCI_TS daughter card built by Pisa group.



PCI daughter card: SVT-PCI-TS which can be plugged into Altera PCI board

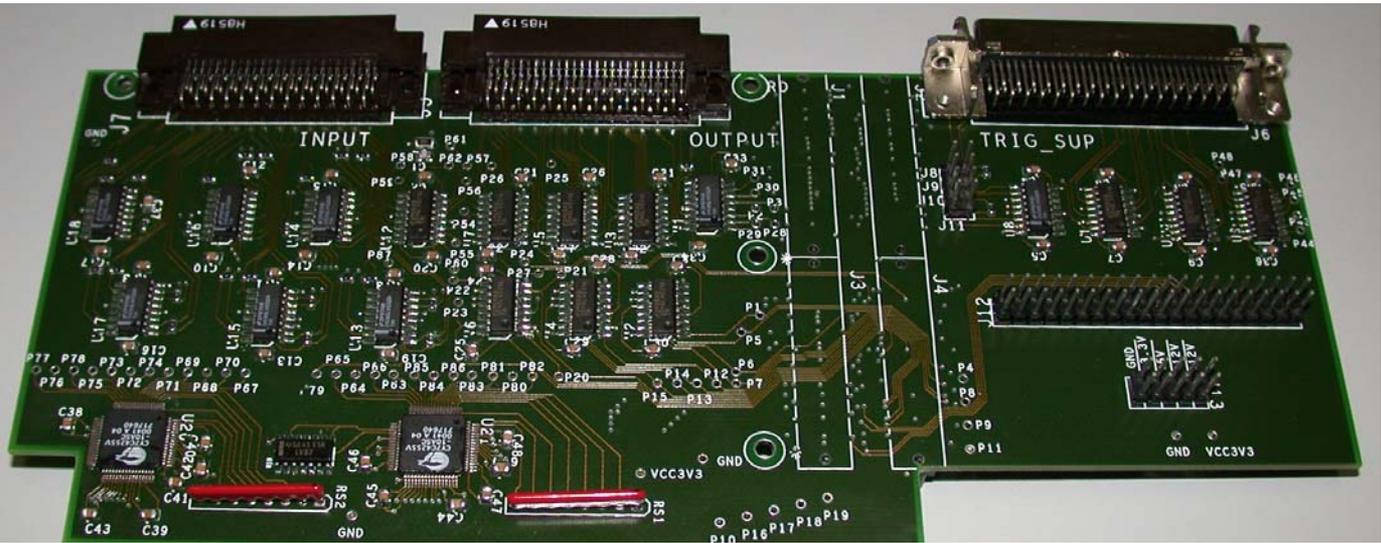
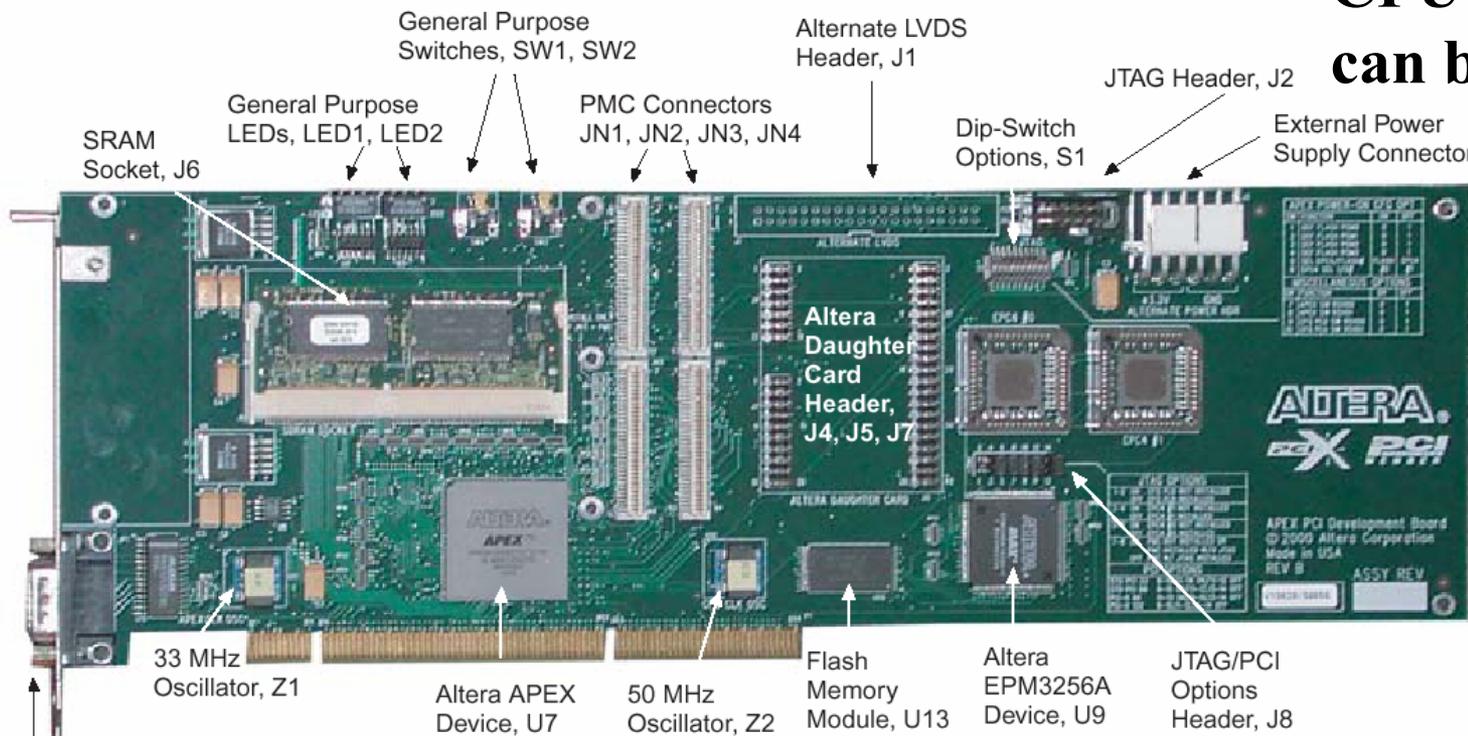


Figure 1. APEX PCI Development Board

The time to send a decision from CPU to Pulsar/TS can be short:

~ 1us measured value





Baseline Design (with the simple configuration) could be:

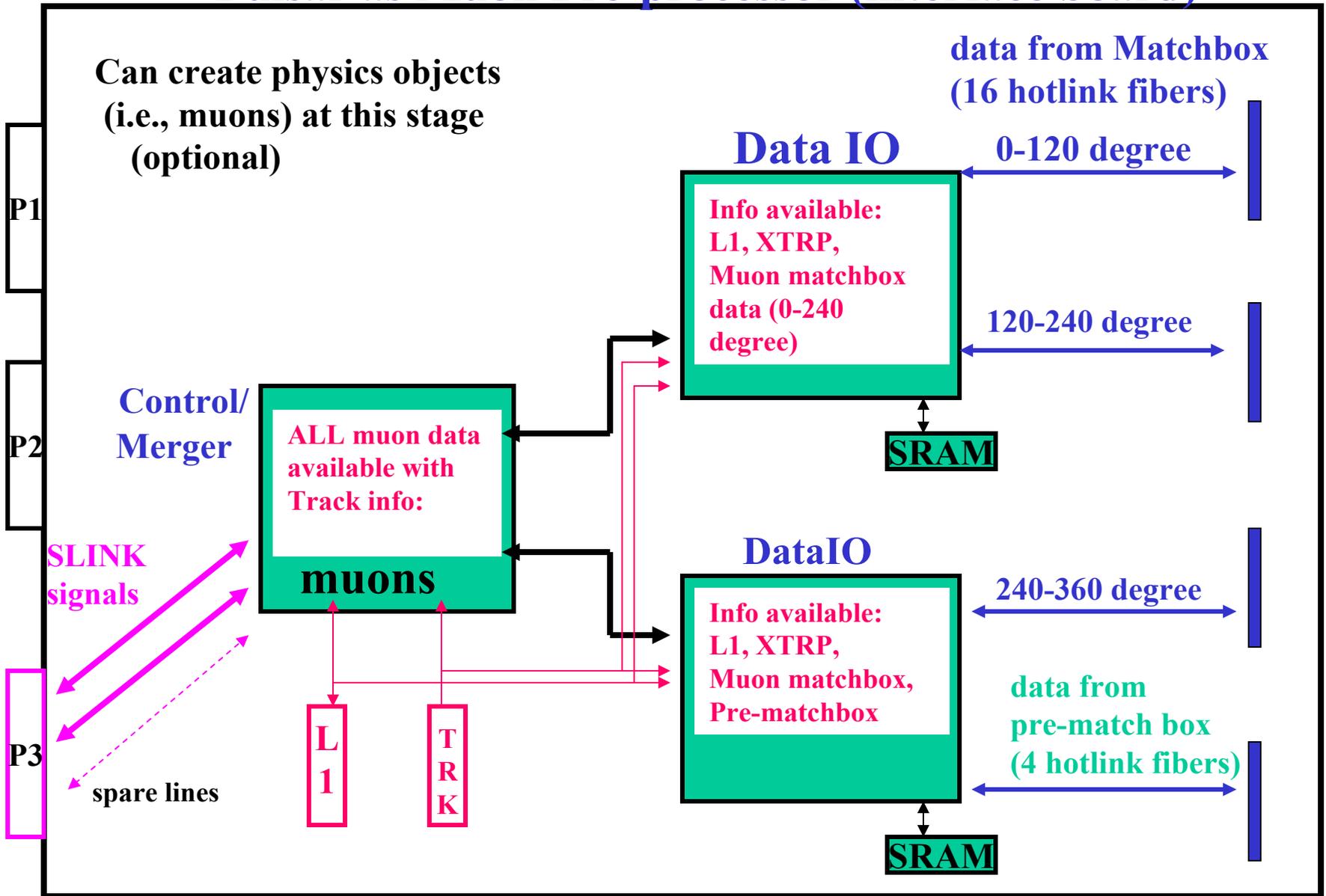
- **PreProcessor/Interface versions** of the board gather data from each subsystem and package the data. This can include sparsification based on L1 trigger bits, tracking, or the data itself.
- **Processor controller/Merger version** of the board merges data from interface boards and packages the data for transfer to a CPU. The data can be further sparsified at this stage. This board also provide the interface between L2 and the Trigger Supervisor.
- Both types of PreProcessor board can be used to readout data

The default operation would have the final decisions made in code in the CPU.

**Pulsar is designed to be able to do more if necessary.
Designed to be quite flexible...**

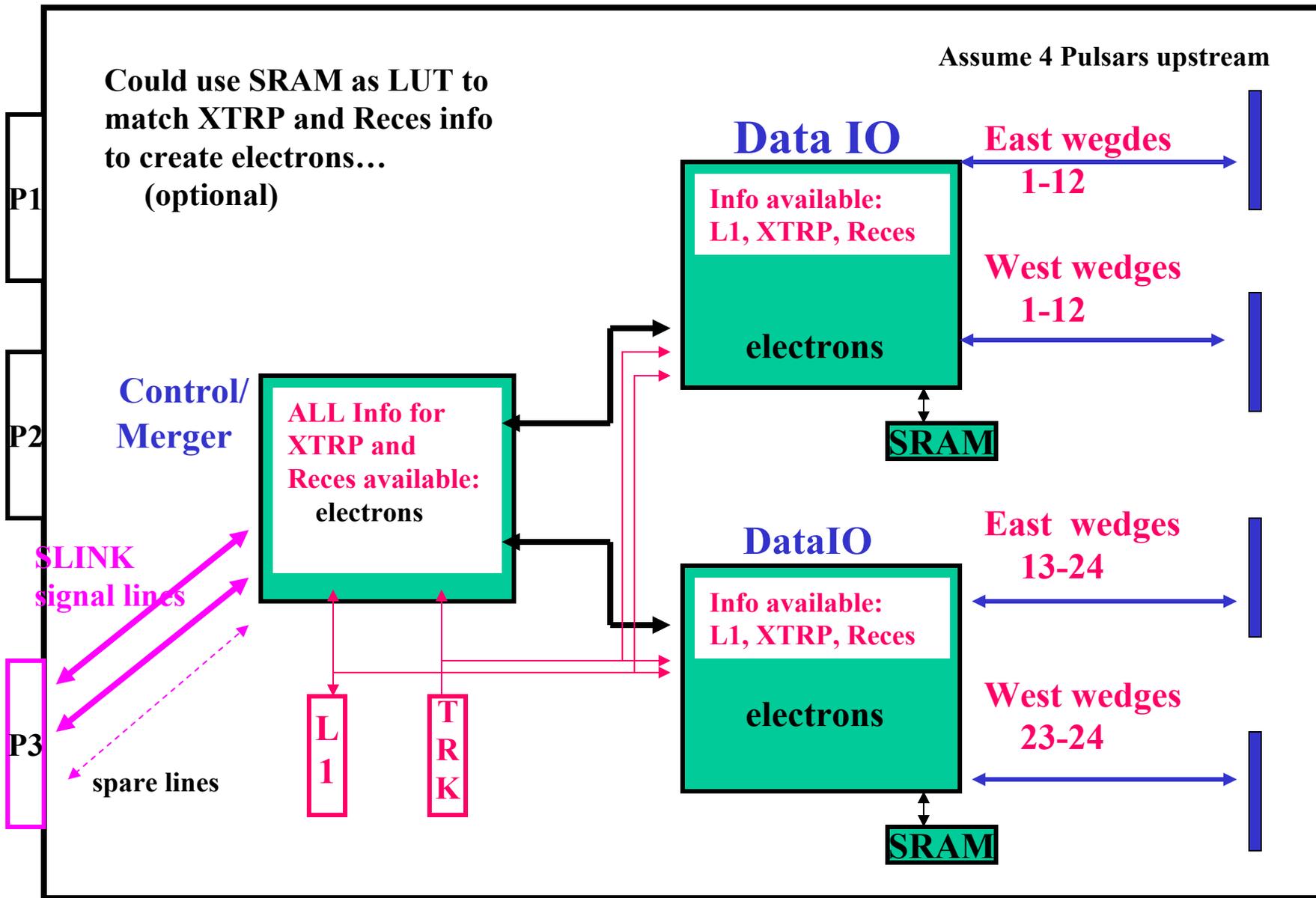
board level and system level Pulsar flexibility

Pulsar as Muon Pre-processor (interface board)



Board level flexibility

Pulsar as Reces data PreProcessor (merger)

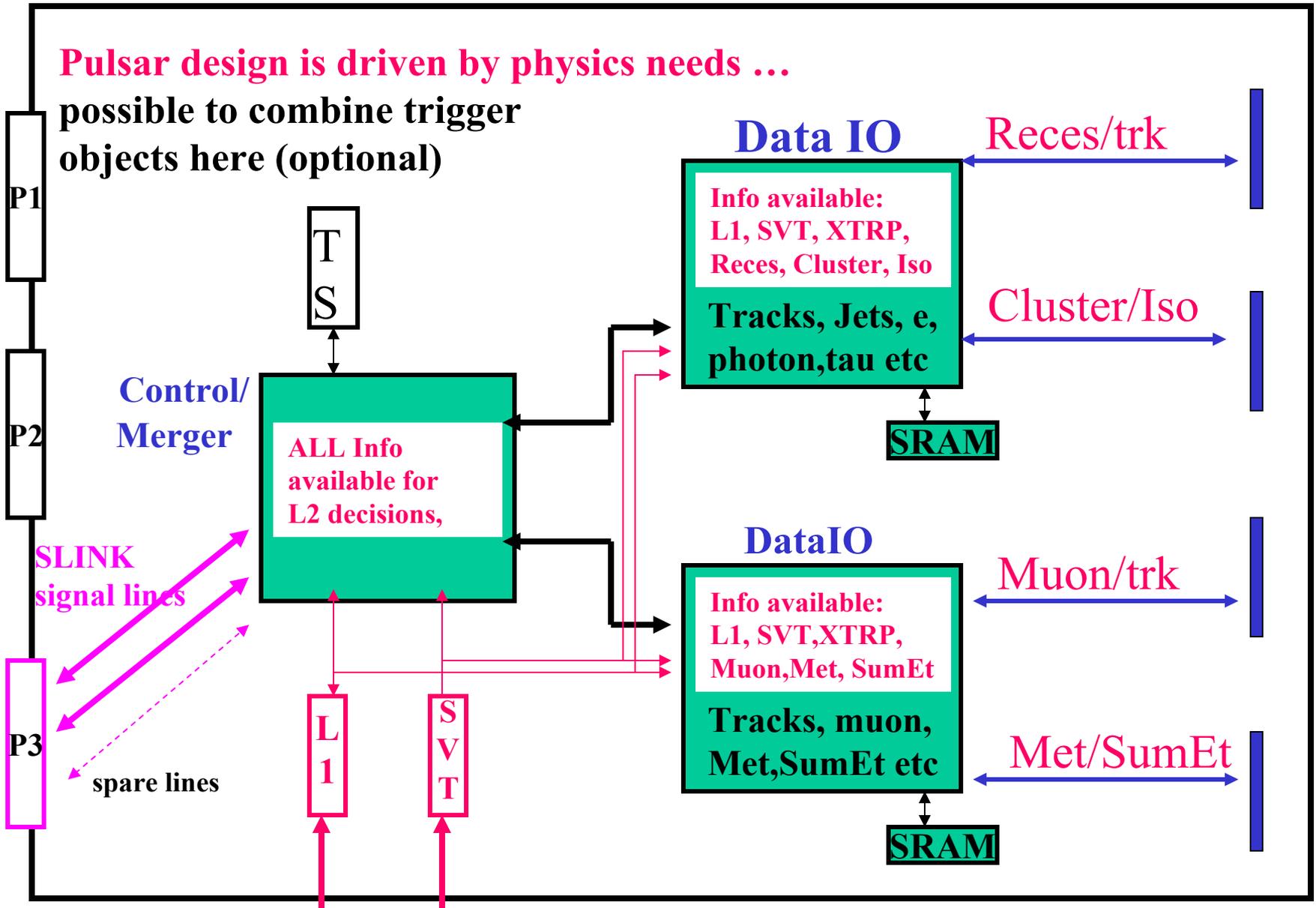


Board level flexibility

Pulsar as Processor Controller

Pulsar design is driven by physics needs ...

possible to combine trigger objects here (optional)



Board Level Flexibility

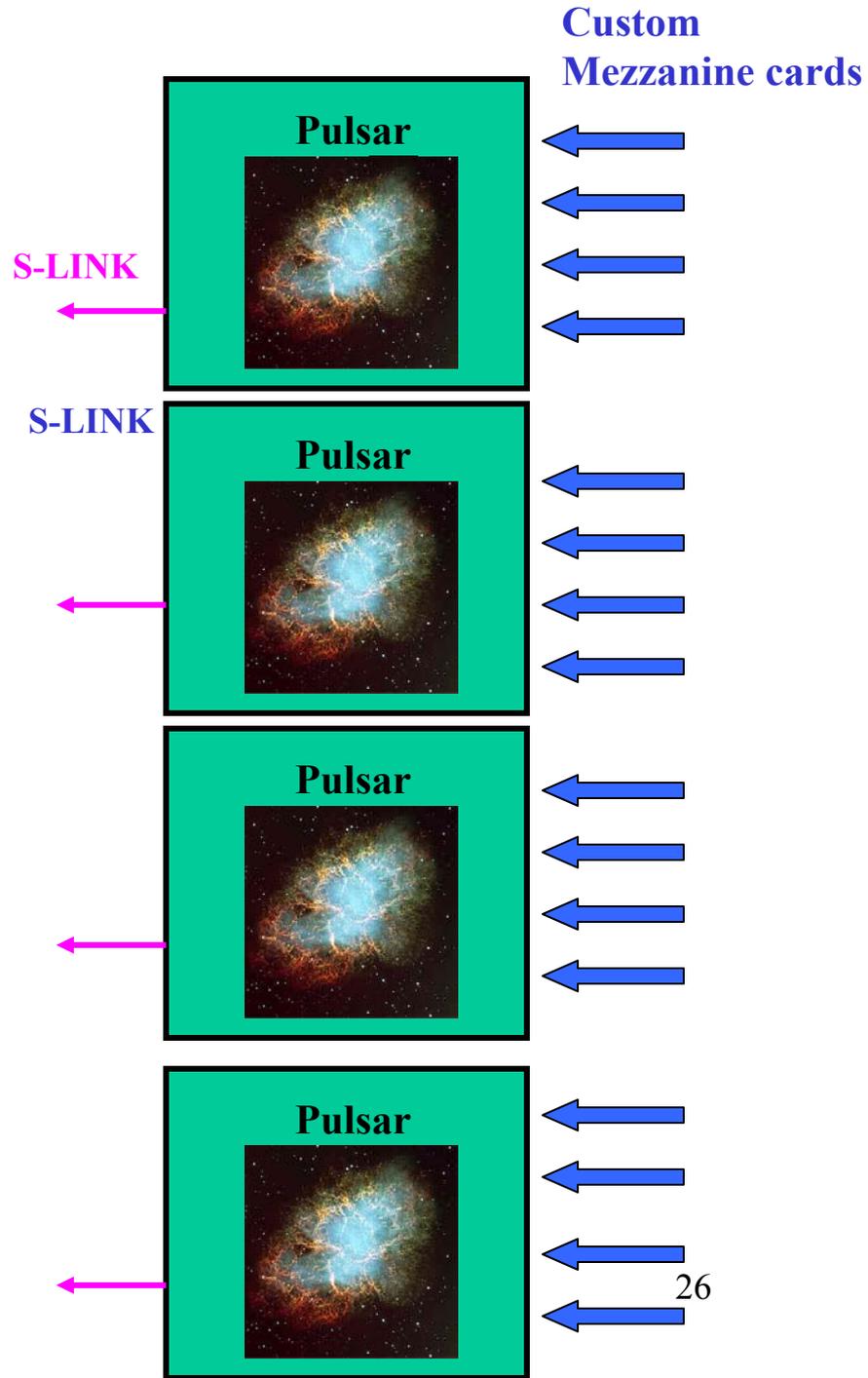
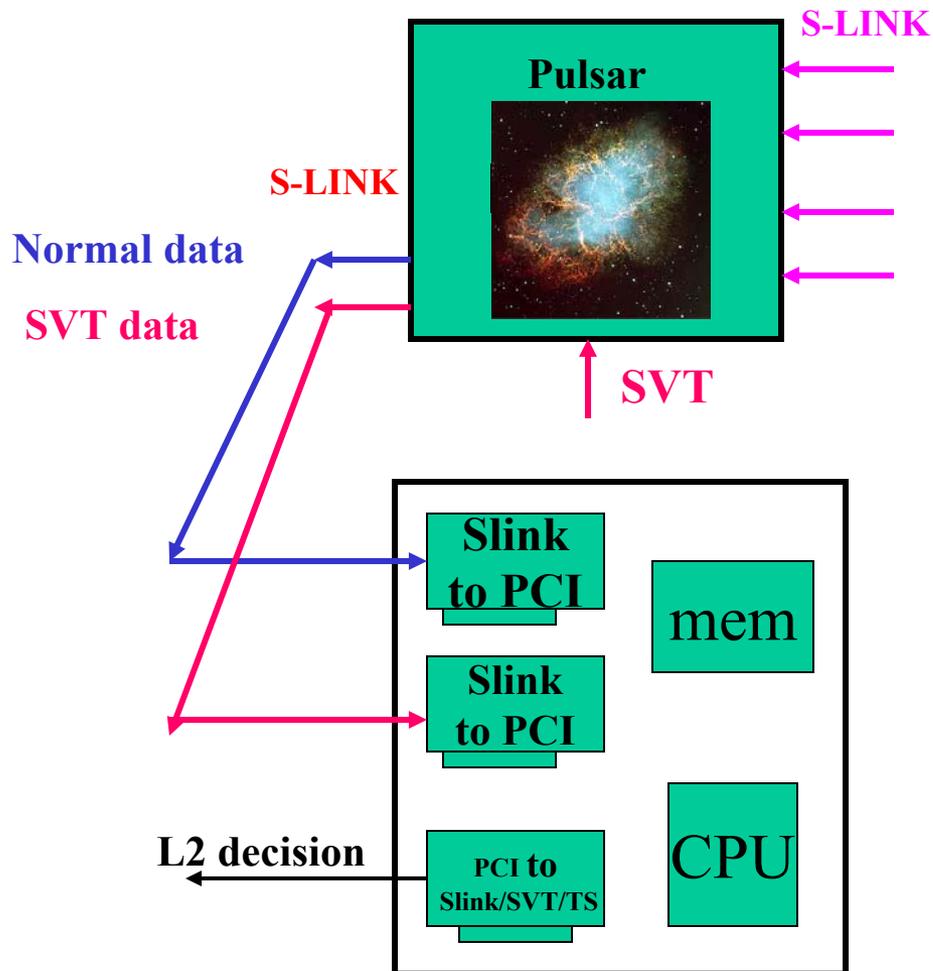
One possible configuration

one PC use two PCI slots to receive SLINK data:

One receives normal trigger data,

the other is dedicated for SVT data (arrives late).

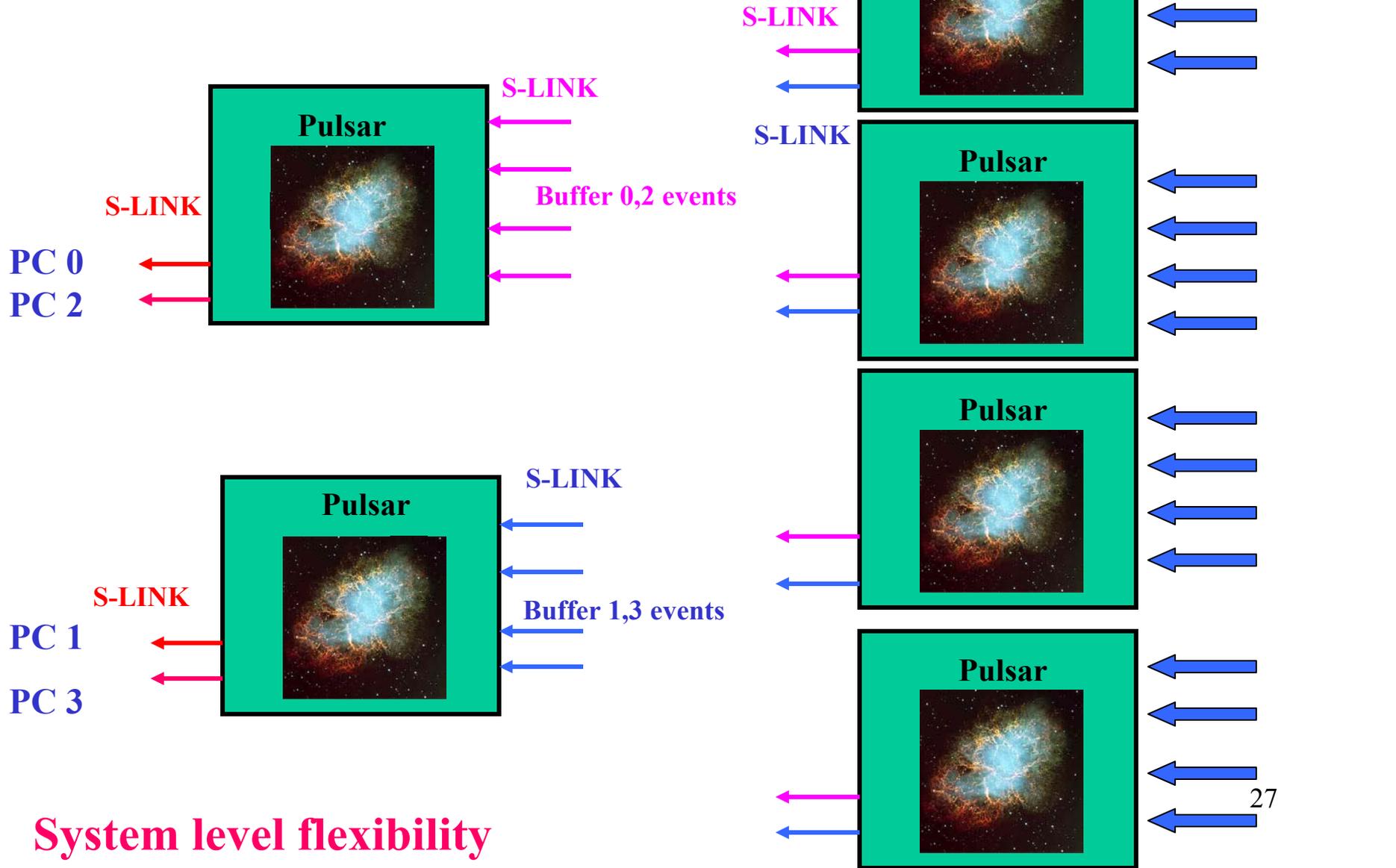
CPU can start working on the early arrival data first



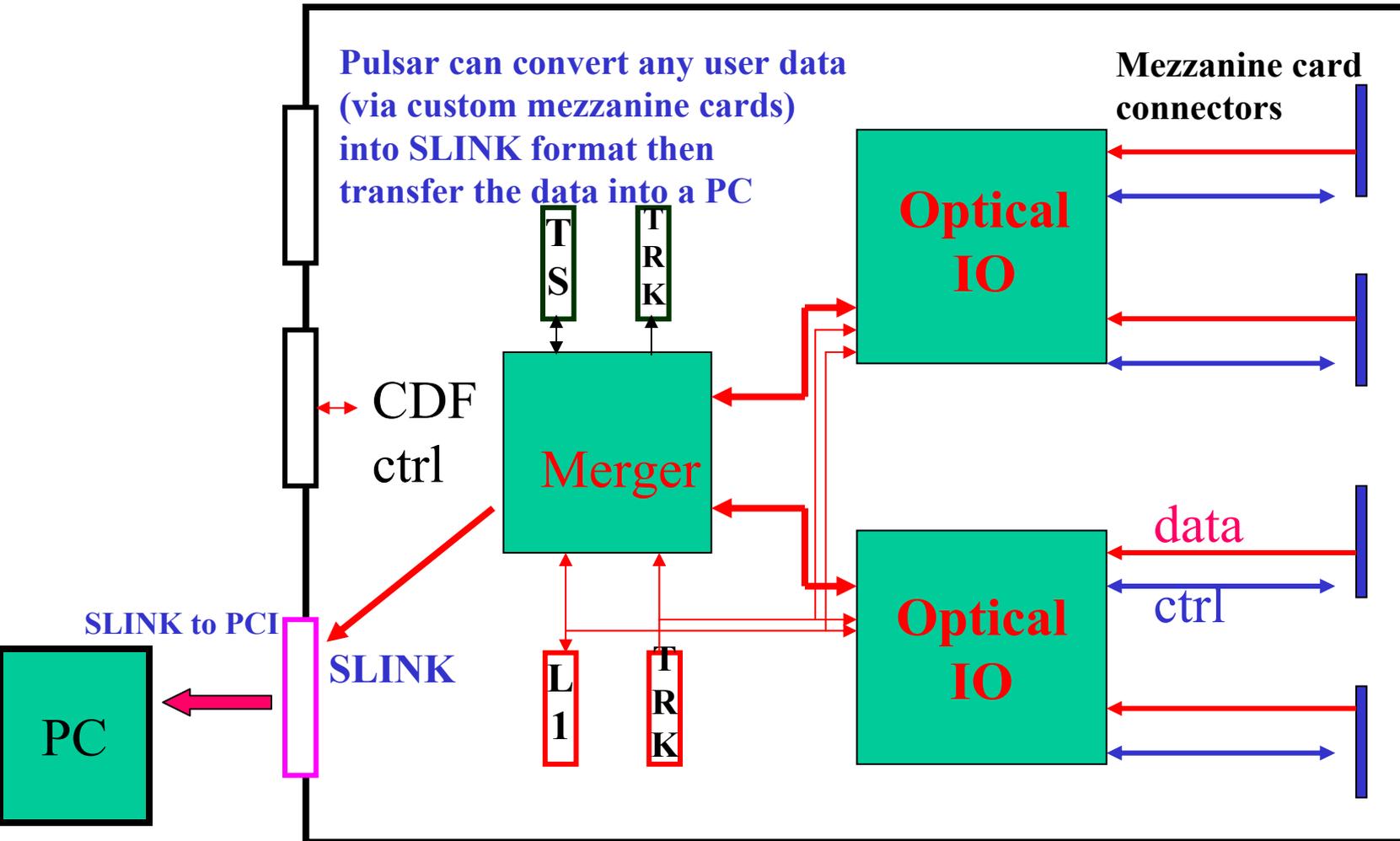
System level flexibility

Another possible configuration using four PCs:

Each PC(i) gets its own event(say buffer i), each works on a different event. Say PC 0 is working on buffer 0, PC 2 is working on buffer 1 etc. **This way one “long tail” event would not prevent other three events to be processed.**

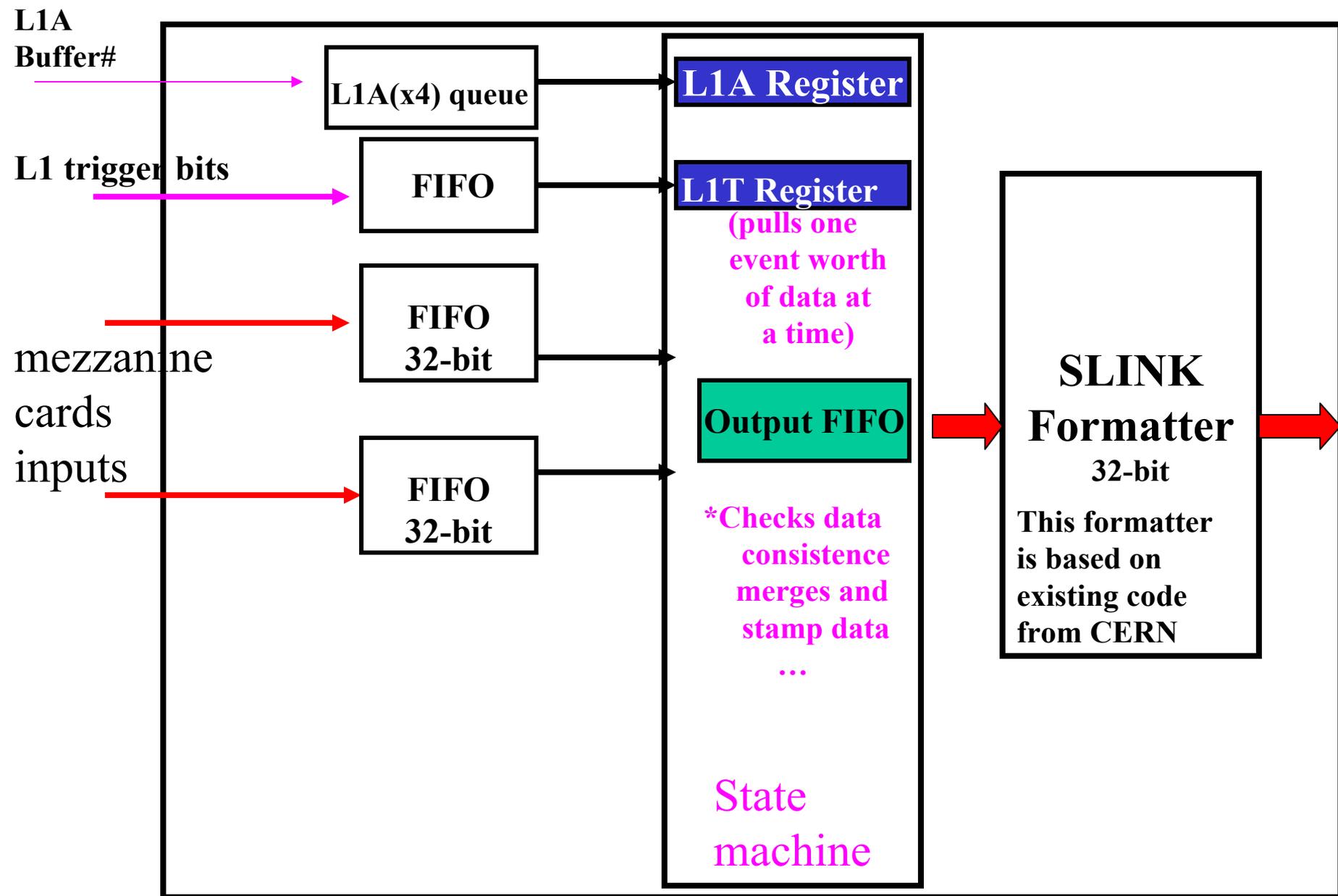


Current firmware status: we have written firmware to record data into a PC for the teststand. Work applicable to upgrade needs.



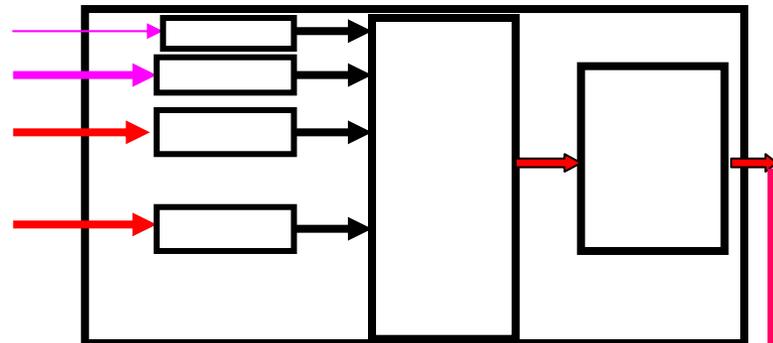
Pulsar in (**general purpose**) recorder mode (directly into a PC)

Firmware for Pulsar in recorder mode(into a PC):

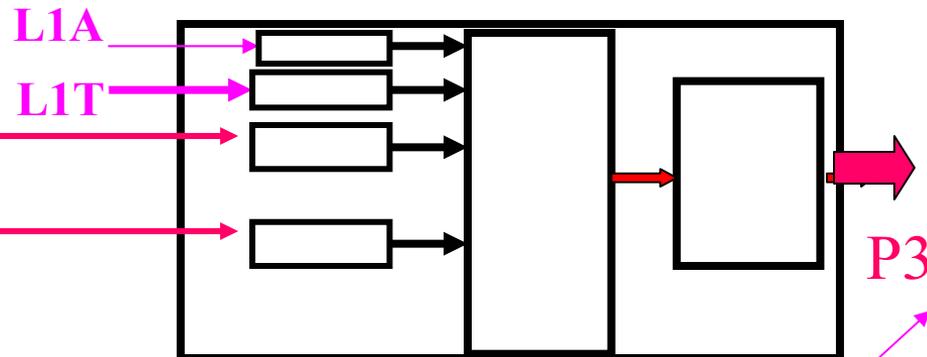


Firmware design is similar in all three FPGA's on all versions of the Pulsar

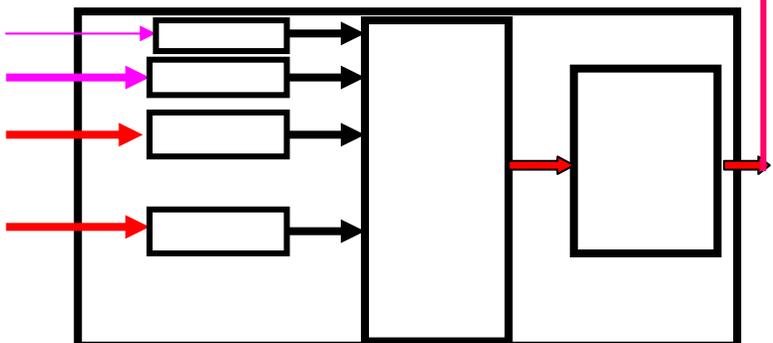
DataIO FPGA



Control/merger FPGA



DataIO FPGA



L1A
L1T

P3

Data package (per L1A)
in SLINK format sent ...

In Summary...

- **Pulsar design is powerful, modular and universal**
 - **Distributed processing motivated by physics**
 - **Sufficient safety margin in raw bandwidth and flexibility at both board- and system-level to handle unexpected Run II B challenges**
 - **Can be used as a test-stand as well as an upgrade path**
 - **Suitable to develop and tune an upgrade in stand-alone mode**
 - **Reduces impact on running experiment during commissioning phase**
 - **Pulsar is useful for Run II A maintenance and *any L2 upgrade***
 - **Built-in maintenance capability**
- **System relies heavily on commercial resources**
 - **Only one custom board**
 - **Firmware design is similar in all FPGA's on all incarnations of the Pulsar**
 - **Easier long-term maintenance**
 - **The rest is commercially available**
 - **Easily upgradeable CPU, PCI boards ...**
 - **Widely used link – well documented LHC standard**
 - **Knowledge transferable to *and from* LHC community**