ATLAS L1 Calorimeter Trigger / DCS

- ATLAS L1 Calorimeter Trigger Overview
- Signals monitored by DCS
- Implementation, prototypes, and current activities
- Towards the real thing…

Birmingham – Heidelberg - Mainz – RAL- Queen Mary - Stockholm

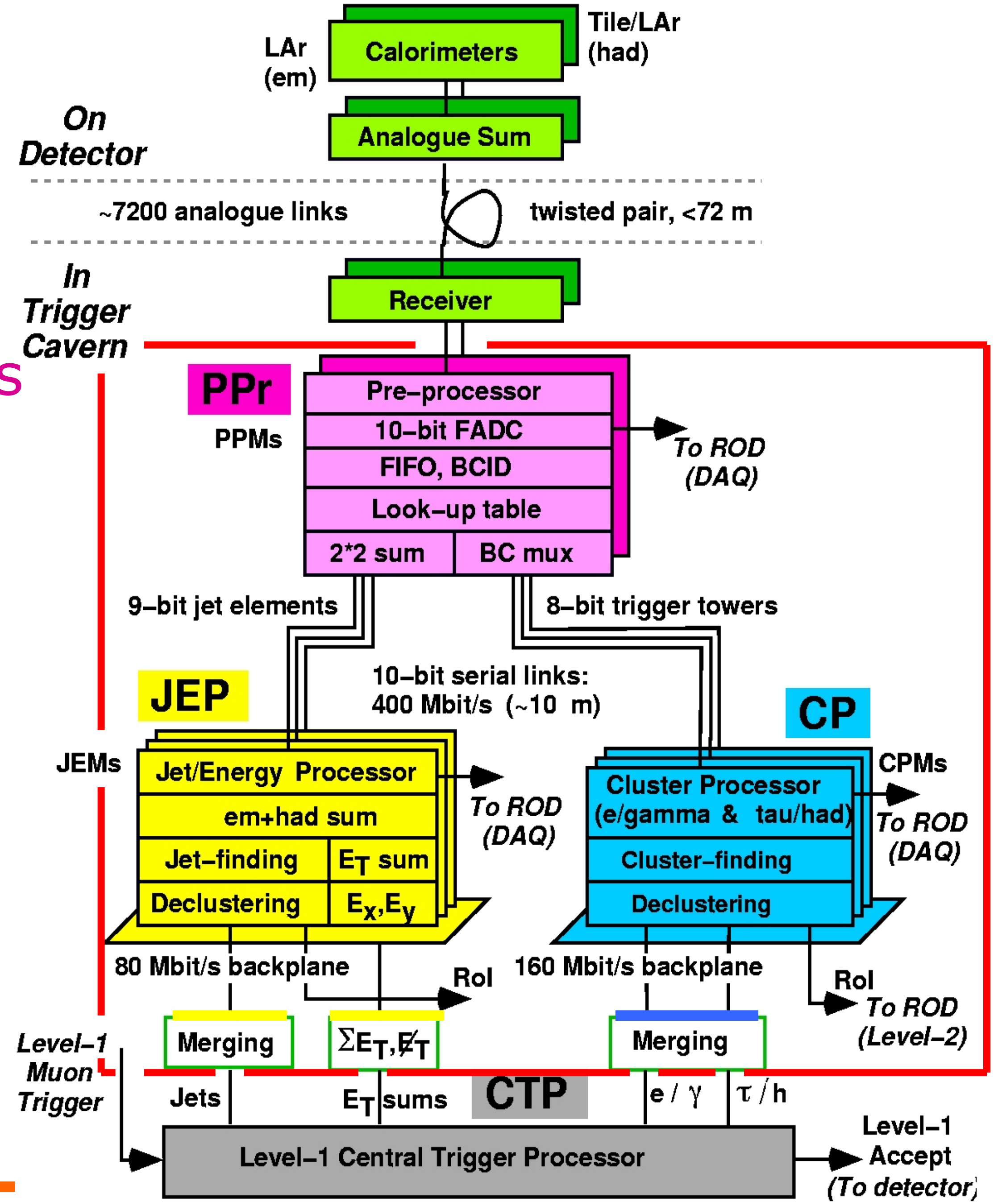
ATLAS L1 Calorimeter Trigger

Pre-Processor:
8 crates, ~128 processors

Jet/Energy Processor:
2 crates, 36 processors

Cluster Processor:
4 crates, 64 processors

All processors are VME-style modules, 9U×40cm
Off-Detector



Data and control channels into DCS

The L1 Calorimeter Trigger comprises a total of ~250 modules. Each of them needs to be monitored by DCS. Due to limited backplane connectivity only 2 pins per module are available for readout of environmental data. Therefore a local CAN node is required on each processor module.

- ~ 8 supply voltages/processor board, either low resolution ADC or digital OV/UV detection only
- ~ 8 temperatures/processor board, low resolution ADC (accurate to a couple of deg. Centigrade only)
- Standard crate monitoring / control (1 CAN node per crate: voltages, temperatures, PWR on/off)
- Sampling rate of ~1/minute might be sufficient

Total amount of data into DCS: < 250 x 16 = 4K words

Requirements

The processor boards are densely packed with components. All components have to adhere to the system-wide geographic address scheme

$\Rightarrow$

- Inexpensive CAN node with low-accuracy A/D converter
- Low consumption of board real estate
- CAN node ID definition via external GeoAdd lines encoded on the backplane
- Needs to fit into standard DCS environment
- No rad-hard components required

So as to reduce the number of CAN lines, crate-level CAN/CAN bridges are envisaged. A crate will look like a single CAN node from a SCADA point of view !

Implementation : (1) ELMB

- CERN standard, full software / firmware support
- Too large to fit on some of the processor modules
- CAN node IDs cannot presently be read in from external lines (GeoAdd) at PwrUp
- Only a tiny fraction of ELMB resources can be used

→ Expensive

Implementation : (2) Fujitsu MB90595

- Single chip CAN node
- Dual CAN version available (useful for CAN/CAN bridge)
- On-chip ADC
- Cheap
- Low real estate consumption
- Not firmware-compatible to ELMB → write our own

Implementation : (3) ???

Can we find a single-chip approach which is compatible with the ELMB in terms of firmware and software?

- Since it adheres to standards it could (?) be supported by CERN
- Small enough to fit on all of the processor modules
- CAN node IDs could easily be read in from external lines (GeoAdd) at PwrUp
- Inexpensive

Comparison

	ELMB	Fujitsu	??????????
Firm-/Software support	**Yes**	**No**	**?**
Single chip	**No**	**Yes**	**Yes**
Real estate	**High**	**OK**	**OK**
GeoAdd / node ID	**No**	**OK**	**OK**
Cost	**High**	**OK**	**OK**

Current Activities, Status

The cluster processor prototype will carry Fujitsu CAN

- A CAN/CAN bridge is being implemented on a „TCM“ module built at RAL
- Hardware exists
- Software being written at Queen Mary (QMUL) by Dave Mills, based on code kindly provided by NIKHEF

The jet/energy processor prototype will carry an ELMB

- A couple of ELMBs available for tests at Univ. Mainz (no further procurement of ELMBs in 2002).

The processor modules will go into combined tests in 2002

Towards the final setup in ATLAS

- Go into L1 calorimeter trigger joint tests (slice tests) in 2002 with 2 or 3 different DCS hardware setups
- Explore the use of PVSSII
- Based on slice test experience, decide on hardware to be used on the production modules. Find a common design for all L1 calorimeter trigger processor modules.

=> require either 0 or ~250 ELMBs on ATLAS