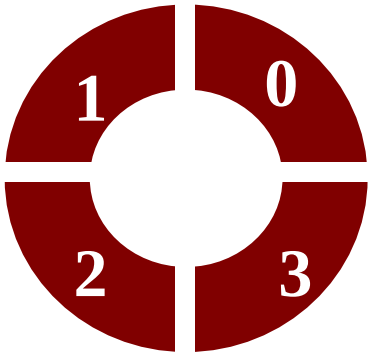
JEP /JEM status

- Overview
- Interfaces
- JEM0 implementation details
- FCAL handling
- Latency / energy sum merging
- Milestones

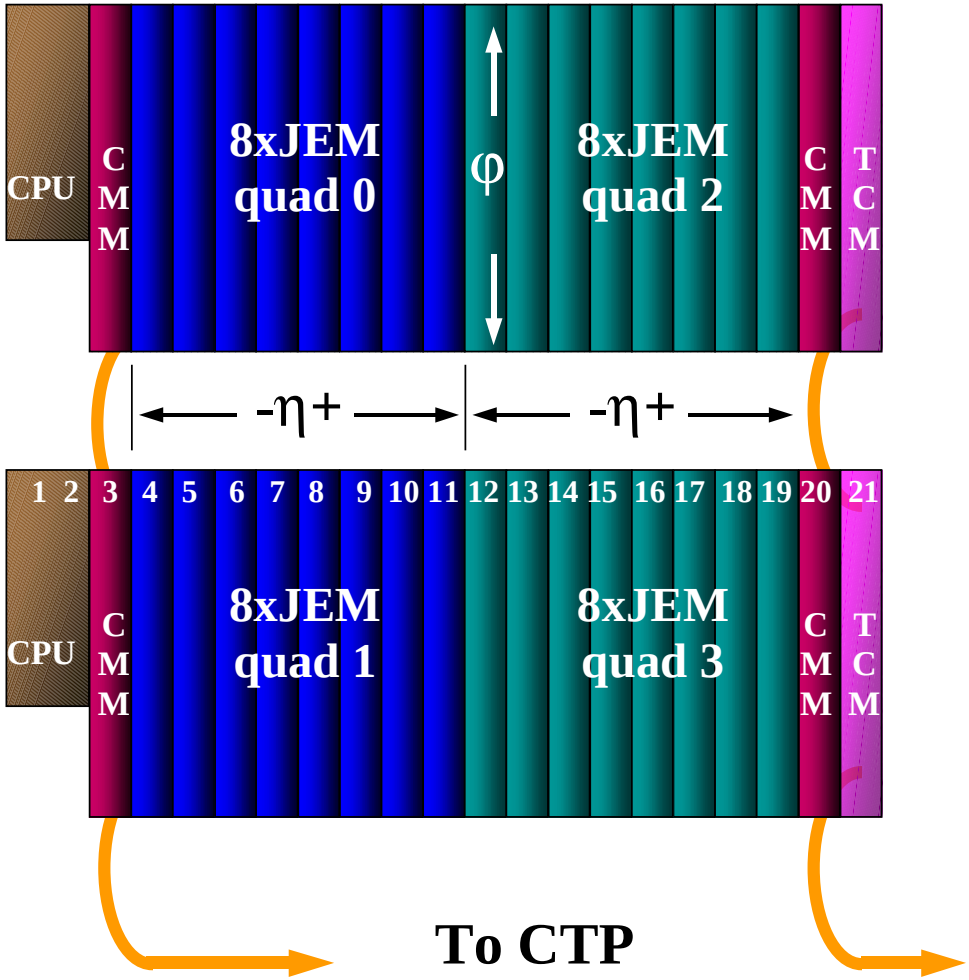
JE system : 2xJEP



2 JEP crates

16 JEMs, 2CMMs per JEP

top level merger processes data from local backplane and from one cable port only



JEP functionality (real-time data path)

JEM :

- receive 10-bit parity protected data from PPr, 88 channels / JEM
- synchronise data to bunch clock
- calculate jet tower sum $E_{em}+E_{had}$ and apply thresholds
- find jets in overlapping windows and count them (8 thresholds)
- apply high threshold to towers and determine scalar transverse energy sum
- calculate vectors (E_x,E_y) and determine vector sum
- format jet count and energy sums (8bit encoded) to parity protected words
- forward two 25-bit words to one CMM each via backplane

CMM (**crate** level merging):

- receive 25-bit parity-protected data from 16 JEMs
- count jets and report jet count to top level jet merger, or
- calculate scalar and vector sums and report to top level sum merger

CMM (**top** level merging):

- receive jet count from local module and from slave crate, calculate total sum and report to CTP (8x3bit)
- receive scalar and vector sums from local module and from slave
- calculate scalar sum, compare to ~8 thresholds, report to CTP
- calculate vector sum, calculate norm, threshold (8) and report to CTP

JEM functionality
(DAQ/RoI/timing/diagnostics/control)

DAQ/RoI :
- pipeline and derandomise LVL1-accepted input data (88 bit) and send to DAQ
- pipeline and derandomise jet count and energy sums (50 / 96 bit) --> DAQ
- derandomise and send jet coordinates (RoI, 13 bit) to LVL2 trigger
- send bunch crossing number to DAQ and LVL2

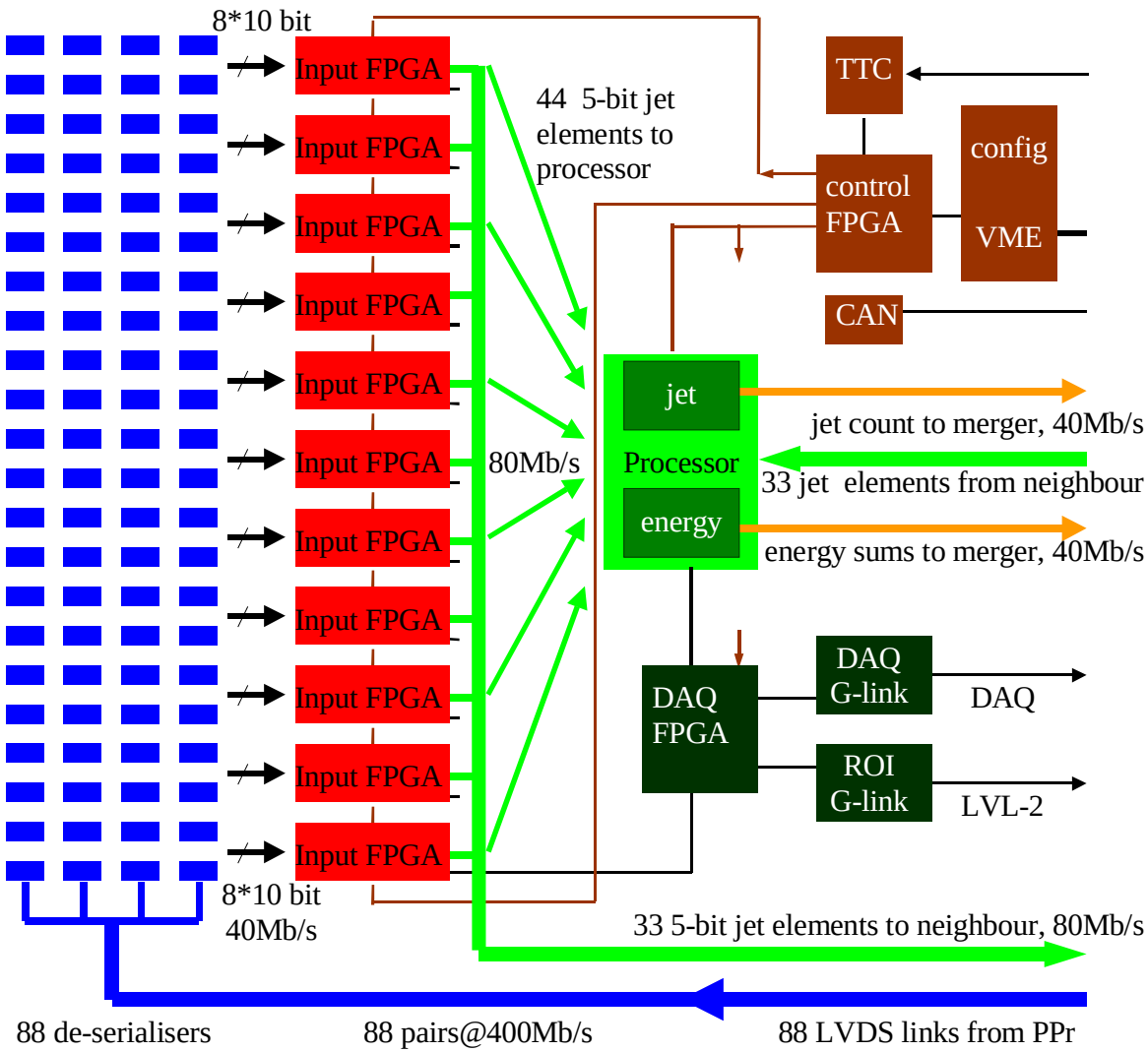
Timing :
- receive timing data from TTC system via on-board TTCrx:
  - o raw and deskewed clocks
  - o event parameters (bunch count)
  - o broadcast signals
- distribute clock and data to processors

Diagnostics/Control:
- provide playback and spy memory to be cycled upon broadcast command
- provide VME access to the processors
- provide CAN access to DCS (CANopen protocol)
- provide JTAG access for board test and FPGA configuration
- provide on-board FPGA configuration storage

JEM block diagram

New !

Jet and energy processor operate both on 0.2x 0.2 jet elements now so as to handle FCAL signals correctly. => they may go into a common FPGA !

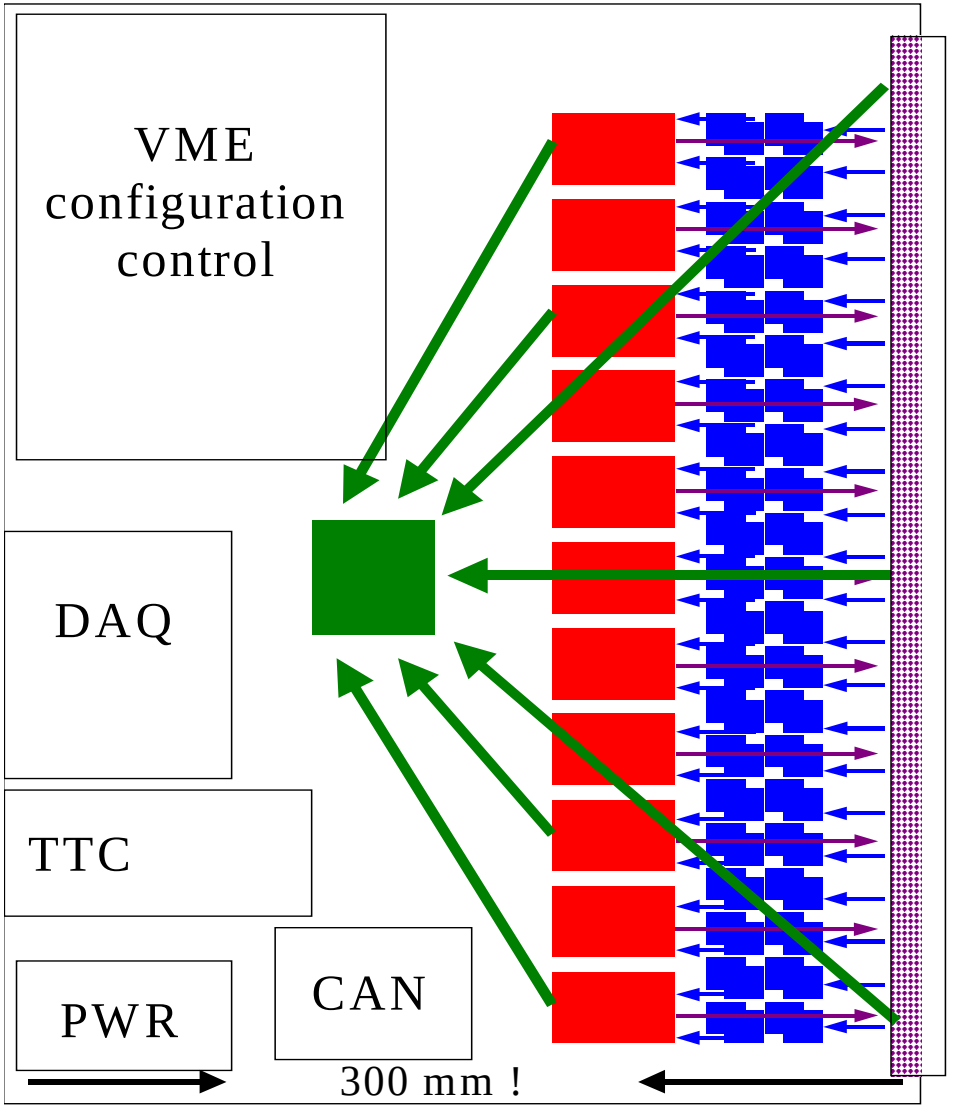


JEM-x floorplan

JEM0 PCB is probably 300mm deep

Mechanically extended to 400mm

Cable connections to front panel.



JEM interfaces

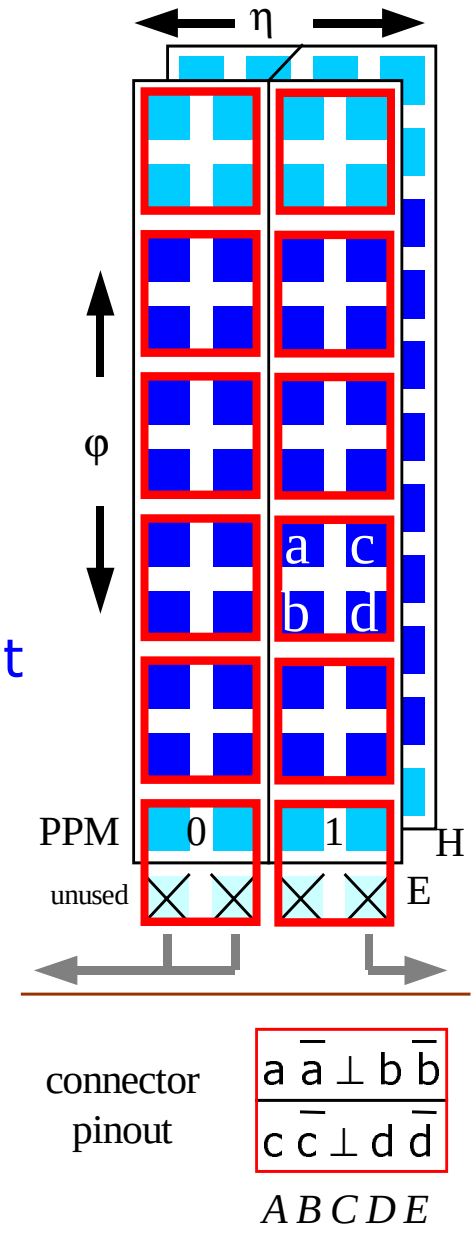
Inputs – on parallel pair assemblies via backplane:

- Each JEM is fed by two electromegnetic and two hadronic PPMs + fanout
- 4 signals per cable, arranged 2×2 in φ×η (barrel)
- 8 cables from PPMs in same quadrant
- 4 fanout cables from neighbouring quads, 2 fanout cables carry 2 signals only

Backplane – see processor backplane specs

DAQ/LVL2:

- Two serial data links (G-link) via front panel connectors (SMC ?)



JEM prototype implementation details

Cable:	mostly AMP 4-pair, 2×2 (φ×η) FCAL !
Deserialiser:	DS92LV1224 (40-66MHz) jitter tolerance and low skew
Processors:	11 input processors (XC2S), main processor (XCV600E +)
Backplane:	FIO/merger signals 2.5V CMOS, 60 Ω ? , **source** terminated
Timing :	socket for tilecal TTC daughter board + TTCrx footprint
DAQ/ROI:	similar to CP
DCS :	socket for ELMB (monitor < 8 voltages, < 8 temperatures)
VME :	5V LSTTL VME-- on backplane, unidirectional ring on-JEM
Power:	step-down regulator for +2.5V, linear regulator +1.8V

FCAL JEMs

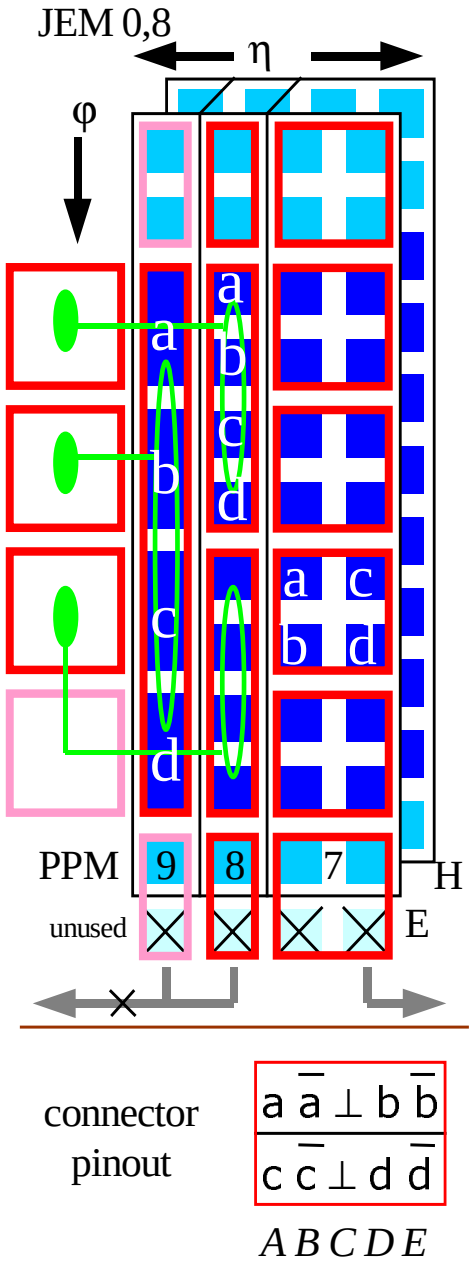
FCAL/endcap JEMs (0,7,8,15) are fed by 3 PPMs (7,8,9)

FCAL channels (PPM_9) have twice the width in φ

=> map 1 cable signal to 2 jet bins

- Populate only one input connector ▯ in FCAL η-bin
- Identify FCAL channel by geographic address
- Divide FCAL signal by 2 (input processor)
- Copy signal to neighbouring φ-bin (main processor)
- Re-map ▯ signal channels to ▭ geometry. Latency !

A small number of 2-pair cables is required to fan out endcap (and FCAL?) signals



Latency (E_{tmiss})

cable from PPM to JEM (7.5 m) 12m	1.5	BC	+1
G-link reception LVDS	2	BC	
initial summing e/h (sync !)	2.5	BC	
threshold, mux, chip2chip, demux	0	BC	+2.5
pre-summation (η)	0.5	BC	
multiplication to Ex and Ey	1	BC	-.5 ?
board summation of Ex and Ey	1.5	BC	-.5 ?
transmission to SMM in crate	1	BC	
crate summation of Ex and Ey	2	BC	
transmission to SMM 2	1.5	BC	
final summation and threshold	3.5	BC	
level shift and transmit to CTP incl. Cable	1.5	BC	
Missing-Et and sum-Et total	18.5	BC	**=> 21+ ?**

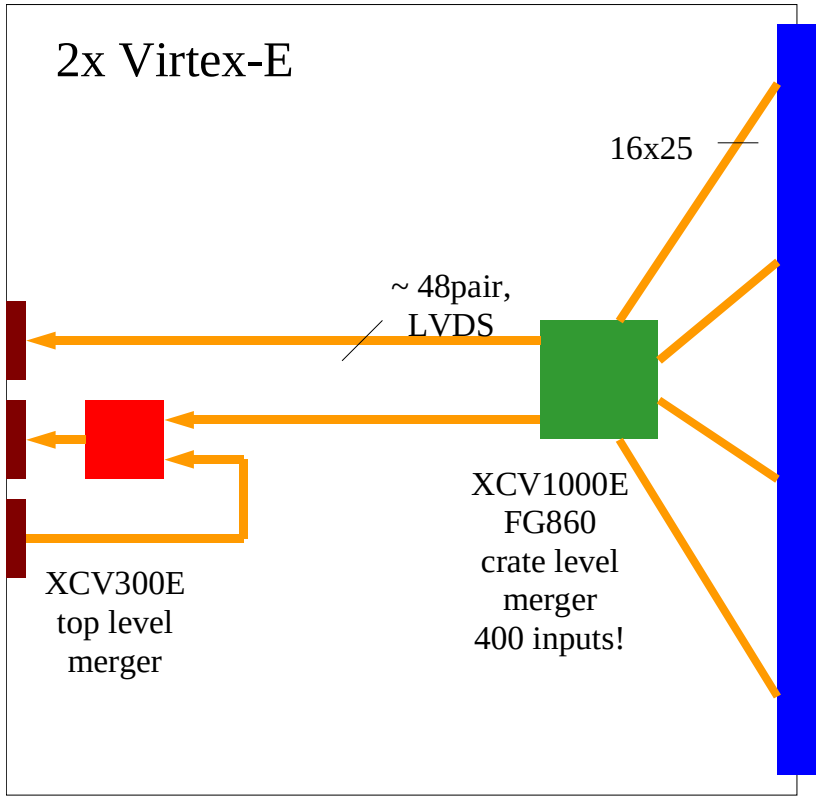
Merger architecture

Minimise merger latency:

Optimise sum merger algorithm **and**

- Run both crate and top level merger on a **single** FPGA each
- No external level translators, use Virtex-E LVDS I/O
- Allow for wide cable ports to the top level merger to avoid encoding and wide parity encoding trees

⇒Almost empty CMM board with one high pin-count FPGA (entirely pin bound)



Energy sum merger algorithm

Crate level merging :

- receive 16 25-bit words from backplane, parity check
- decode each word to 3x12-bit linear (requires **48** 4k memory blocks !)
- 4-deep adder tree for E_t, E_x, E_y
- send E_t, E_x, E_y (~16bit each, incl. parity) to system level merger

System level merging :

- receive 3 words from local board, 3 words from cable port
- parity check, sum E_t, E_x, E_y @ ~12 bit range
- threshold E_t, ~4 bits to CTP
- calculate sqrt (E_x^2+E_y^2) and threshold, ~4bits (?) to CTP
    - o single LUT (E_x, E_y) → encoded trigger bits → decode to 8 bit ?
    - o 2047 GeV range, 4 GeV resolution (from URD) accuracy 1.6% (6 bit)
    - o → LUT size 8 kbyte (or 32 kbyte at 0.8% accuracy, *latency!*)
    - o or 4095 GeV, 1 GeV, 1.6% → 14 kbyte (=XCV200E)

- need to calculate and download LUT contents via VME for each new set of thresholds!
- Block memory resources required in **real-time data path**, → DAQ, RoI ?

Outlook

JEM0 specifications :	under way
Review :	2000
Prototype design:	2000

The module being built is not a proper module 0. It will not be built from CORE money.

It is meant to be functionally equivalent to the final JEM and will be built to JEM0 specifications as far as possible.