Future of prototype modules & firmware development strategy for the FC7-testBorard

2016-10-10
Phase 2 Outer Tracker System Test & DAQ
Tracker week
stefano mersi

New hardware coming soon

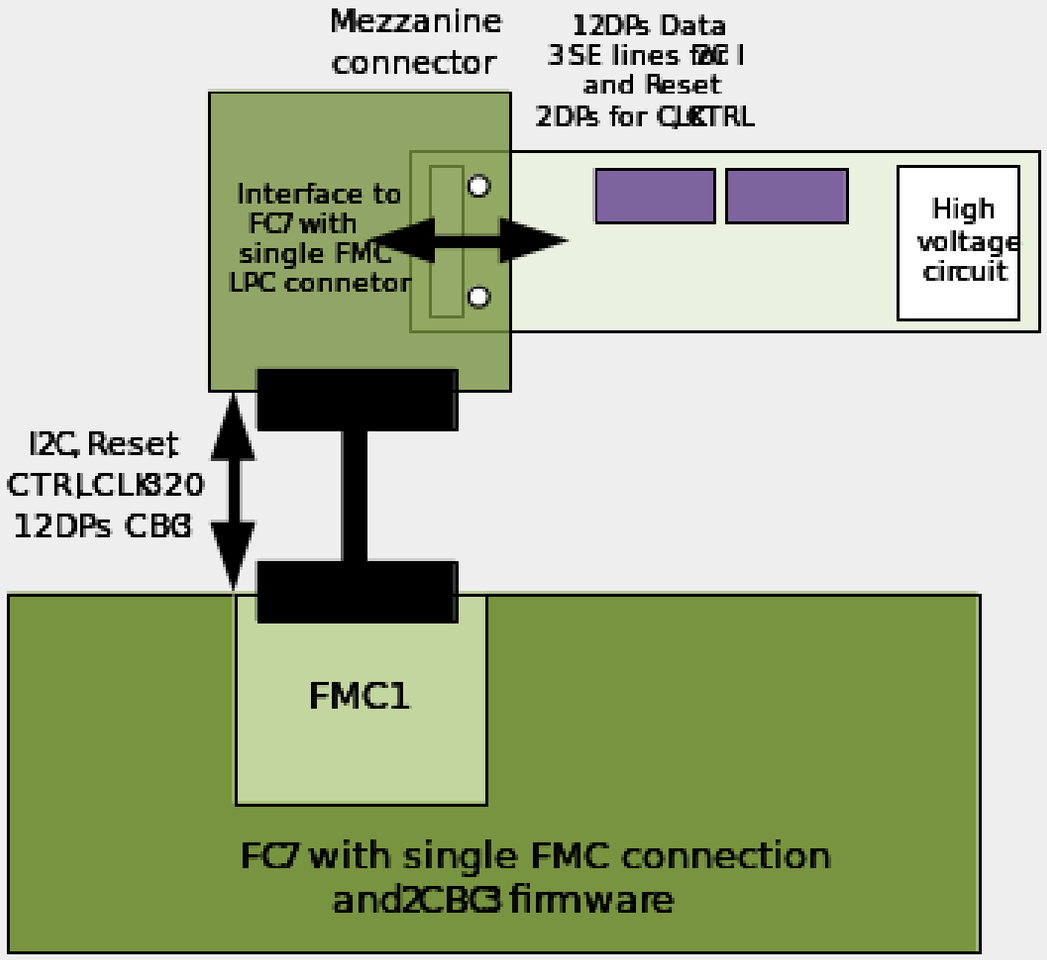
- CBC3 ASIC available (most probably) in November
- ASICs testing by Mark Raymond
- Functional tests require hybrid+DAQ
- CIC will arrive later (CIC-less system might be needed)
- Hybrid and architecture proposals (M. Kovacs) in the next slides

M. Kovacs https://indico.cern.ch/event/568162/

2×CBC3 flex hybrid

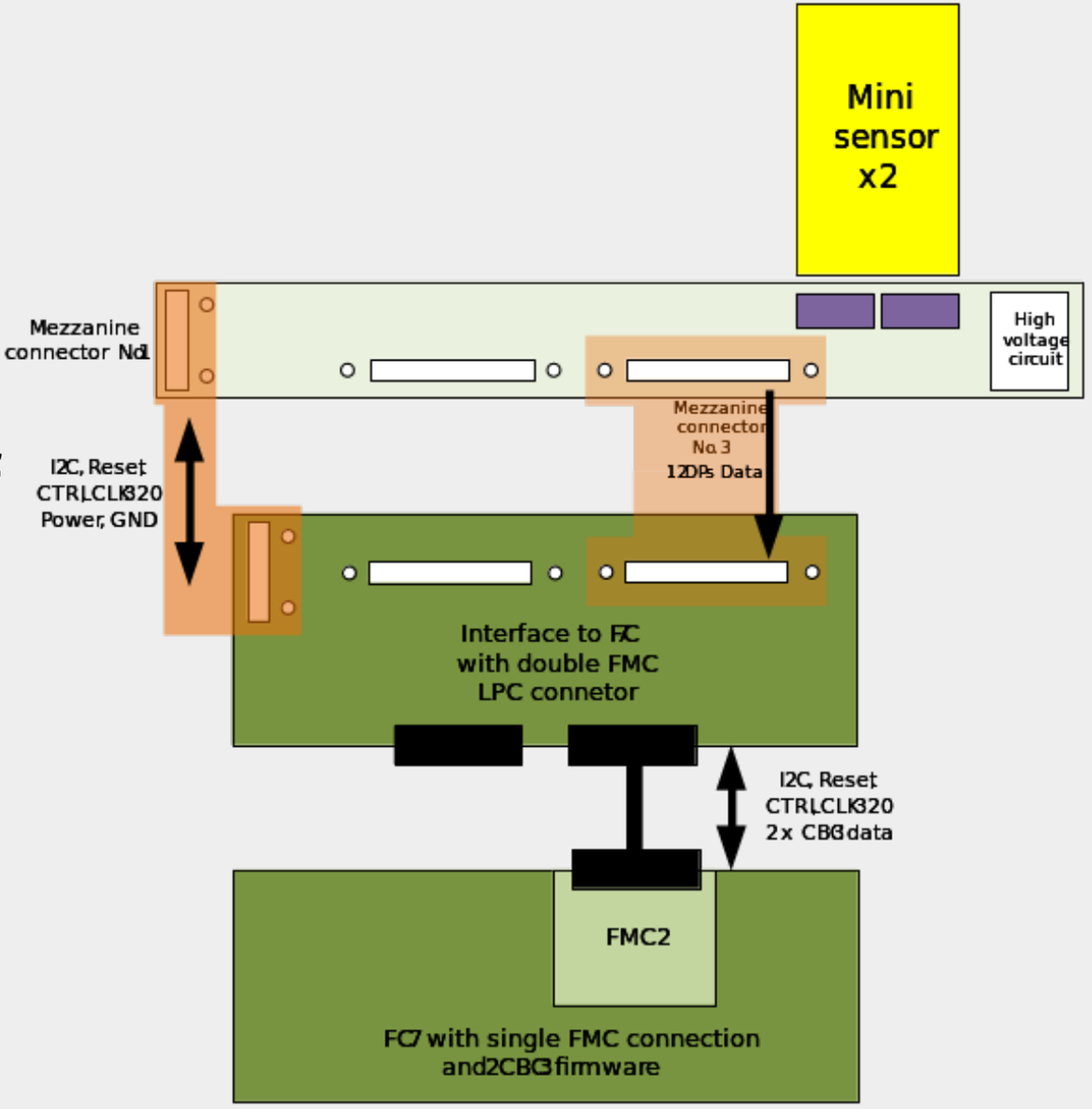
Reduced size flex hybrid, populated with 2×CBC3 chips

- Mini modules with mini sensors
- Not compatible with CIC: readout connector + interface board
  - 12 differential pairs for the L1 data and trigger from the 2×CBC3
  - 1 differential pair for the clock
  - 1 differential pair for the control signal
  - 3 single-ended lines for the I2C and Reset
  - 25 pins for GND and power connection (4 might be used for 40MHZ reference CLK test)
- Interface board:
  - SLVS → LVDS and LVDS → SLVS converters
  - A single VHDCI connector to connect to one **Low Pin Count (LPC) FMC connector** (proposed: Novastack HDP 35)



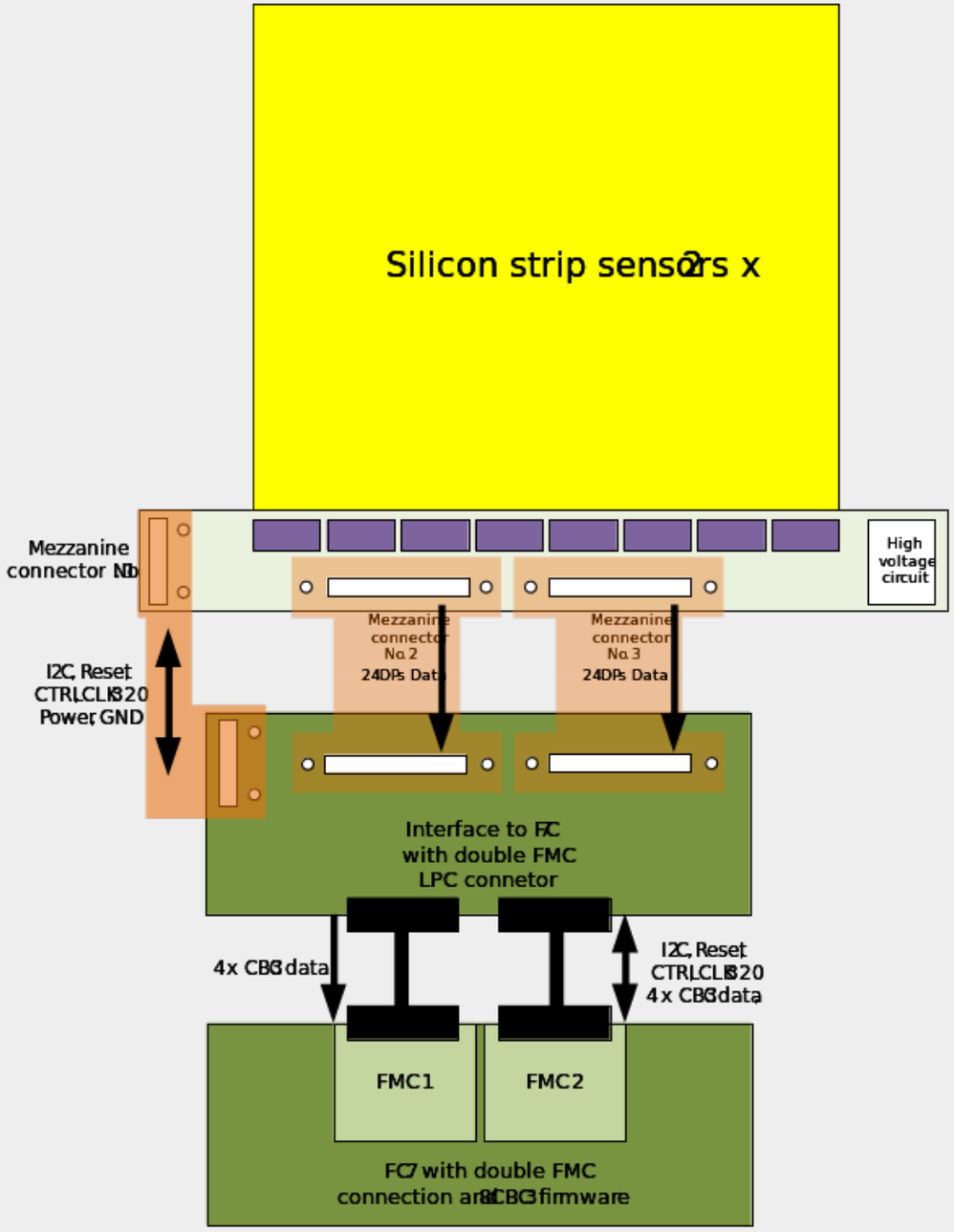
8×CBC3 flex hybrid

- Mini modules with mini sensors
- Full size module
- Possible integration with a real service hybrid
- Two connectors possible in place of the CIC
- Two possible configurations:
  - 2×CBC3 and one connector only
  - 8×CBC3 and two connectors
- The FC7 has 68 user defined differential IO pairs that can be used up to 400 Mbps



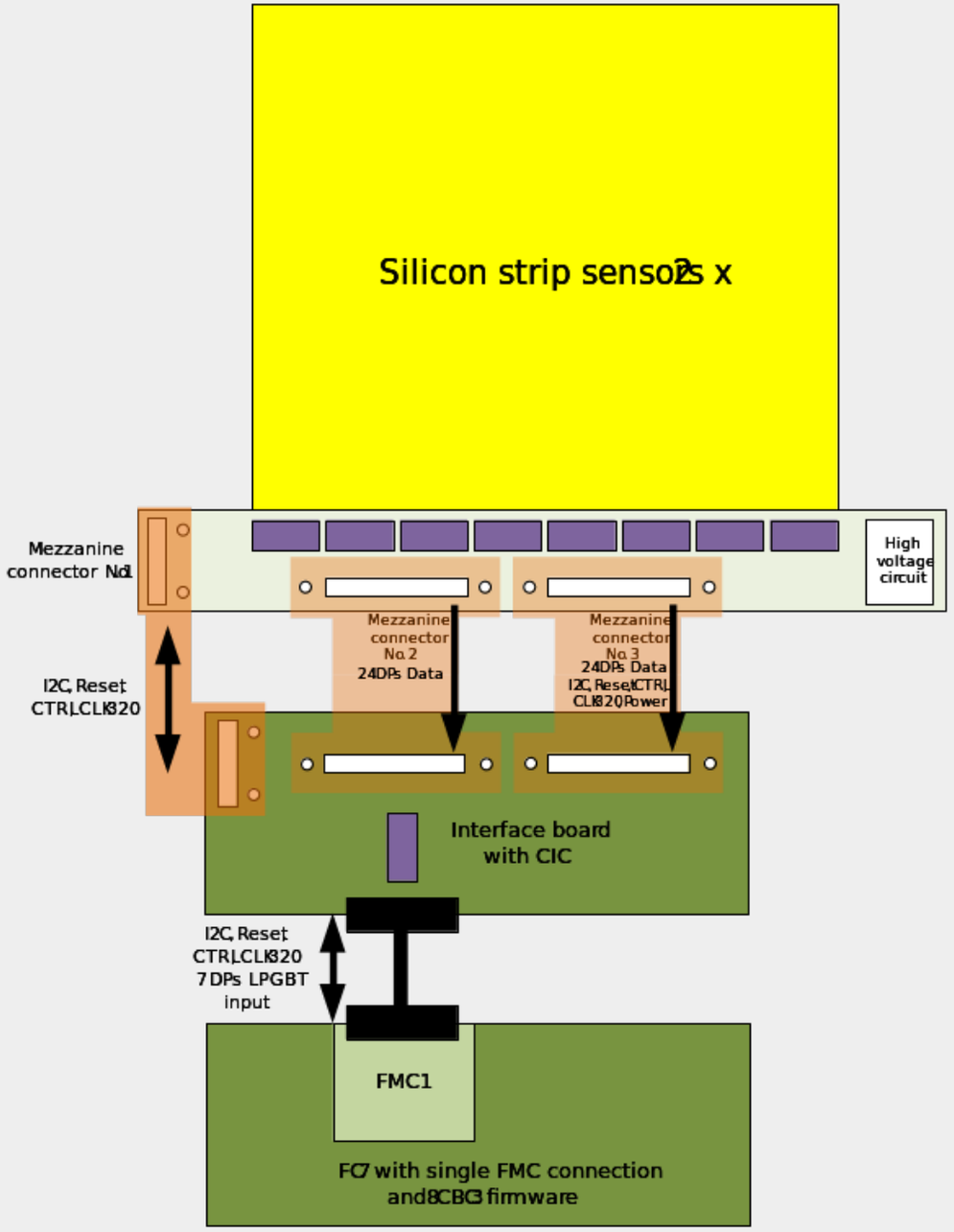
8×CBC3 flex hybrid

- Mini modules with mini sensors
- Full size module
- Possible integration with a real service hybrid
- Two connectors possible in place of the CIC
- Two possible readout configurations:
  - 2×CBC3 and one connector only
  - 8×CBC3 and two connectors
- Two possible readout schemes
  - Without CIC
  - With CIC (or CIC emulator)
- FC7 has 68 user defined differential IO pairs that can be used up to 400 Mbps



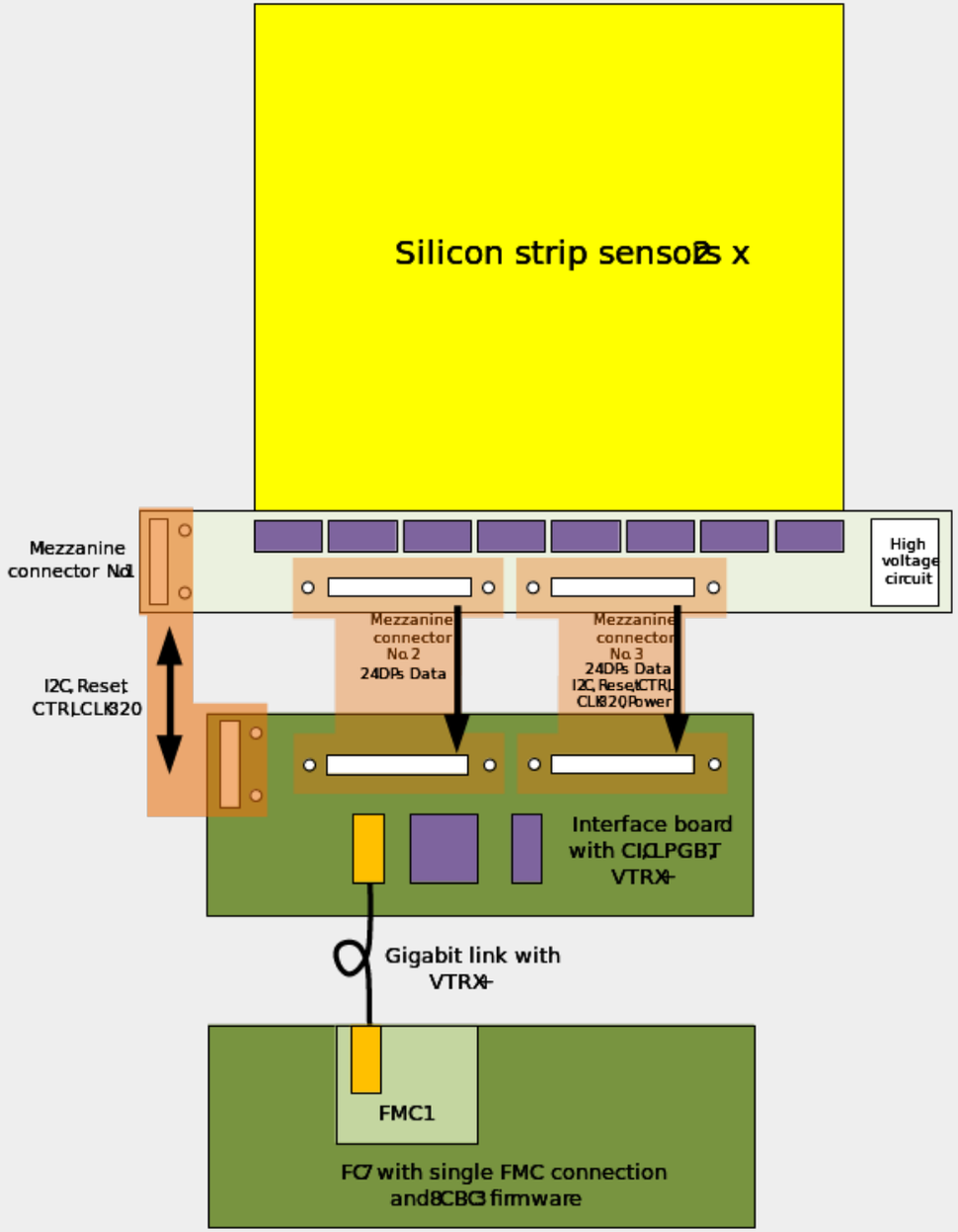
8×CBC3 flex hybrid

- Mini modules with mini sensors
- Full size module
- Possible integration with a real service hybrid
- Two connectors possible in place of the CIC
- Two possible readout configurations:
  - 2×CBC3 and one connector only
  - 8×CBC3 and two connectors
- Two possible readout schemes
  - Without CIC
  - With CIC (or CIC emulator)
- FC7 has 68 user defined differential IO pairs that can be used up to 400 Mbps



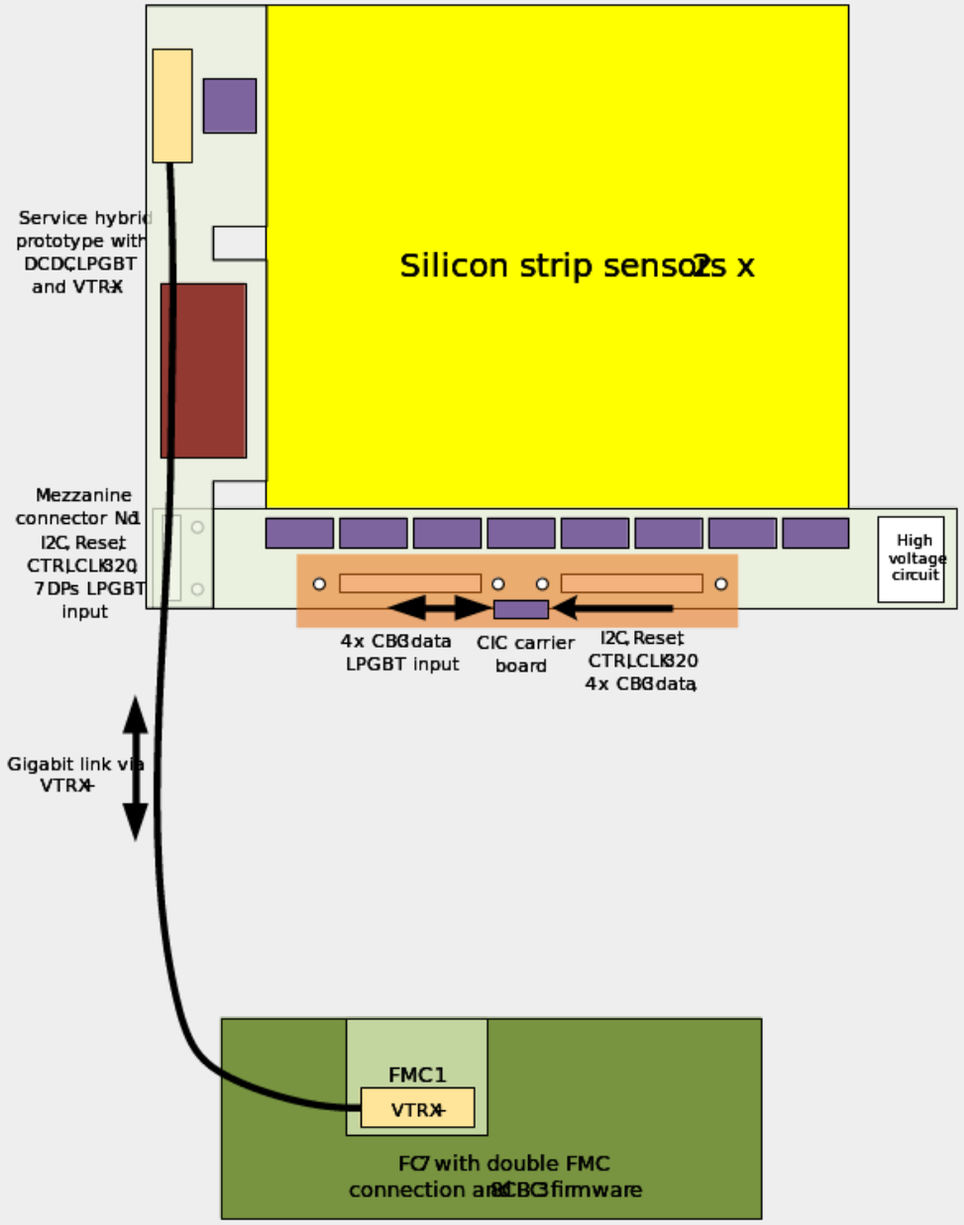
8×CBC3 flex hybrid

- Possible to add a GBTx ↔ GBT-FPGA link via the mezzanine



8×CBC3 flex + service

- Seamless transition when a complete service hybrid is available with
  - Prototype service hybrid can instrument a GBTx
  - Final object: lpGBT



Final choice?

from M. Kovacs's proposal:

> *Based on the estimated development time and cost, I think the 2×CBC3flex option is not worth the effort that it needs. The 8×CBC2flex system is much more flexible and it can be used for several other applications than testing the CBC3.*

- Proposal was discussed in Electronics meeting and is likely to undergo some modifications, especially if the CIC will turn out to be available earlier than currently expected
- A similar approach could be used to test the variants of the PS module prototypes and test structures
  - Less defined at the moment

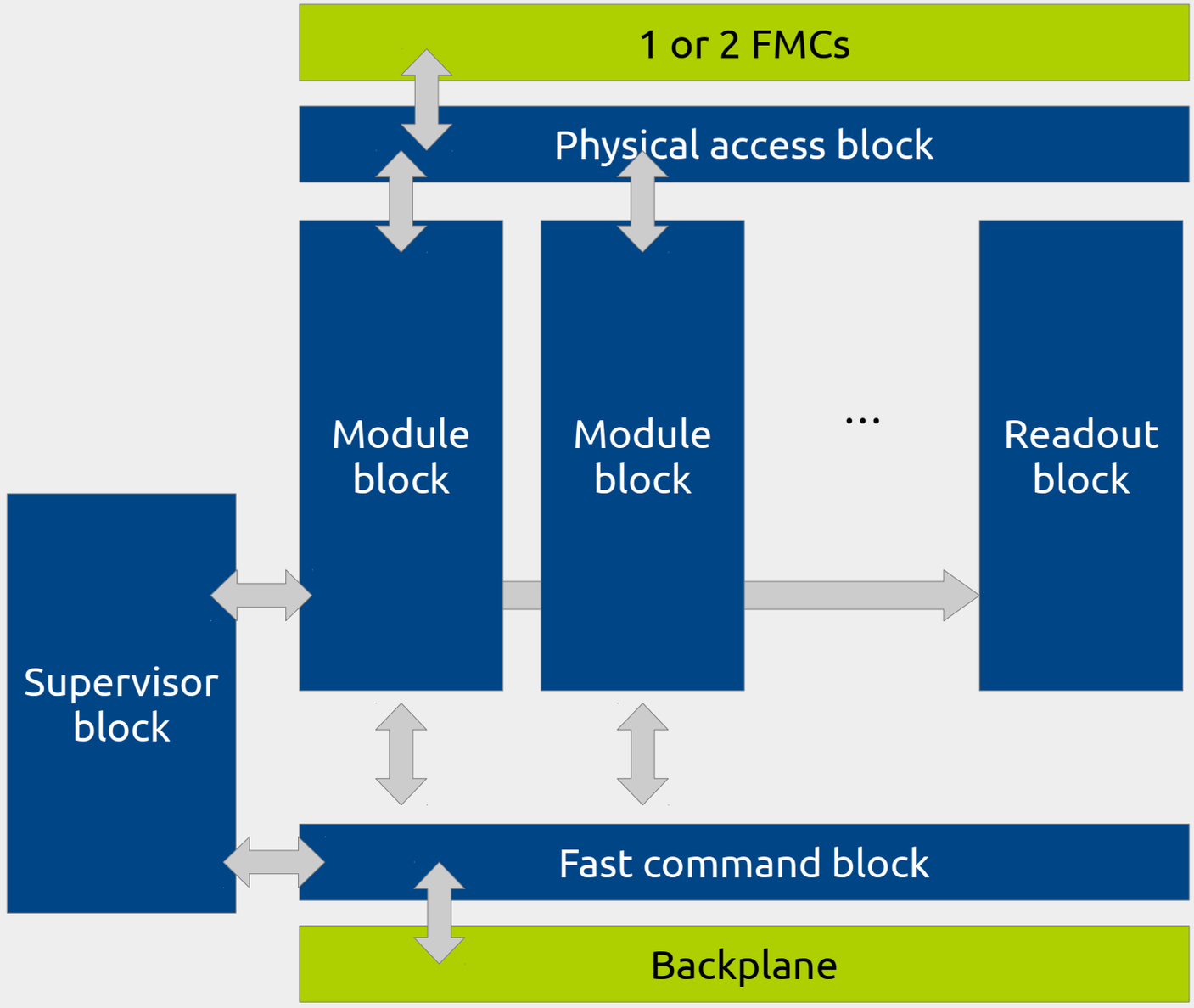
Front-end variants: conclusions

- The set of variants will not be settled for a while
- A number of test module variants should be expected
  - With/without CIC
  - With/without (lp)GBT
- If adequate DAQ system available, it can be used to address a large set of measurements/tests with these cards
  - Component testing
  - System integration
  - Beam tests
- A modular approach in the DAQ firmware design seams favoured here

Firmware design proposal

- architecture design: start with an agreement on blocks
  - well defined functions (see next slides)
  - well defined interfaces (to be done soon by a small WG)
  - switch between compatible blocks via compilation (if code is well structured it should be easy to recompile a FW with a given set of blocks – see later for an example)
- all included in the same repository
  - Centrally managed
  - Possibly gitHub (or other version control system) to ease code review
    - Clear integration work-flow
    - pull → merge → test → push

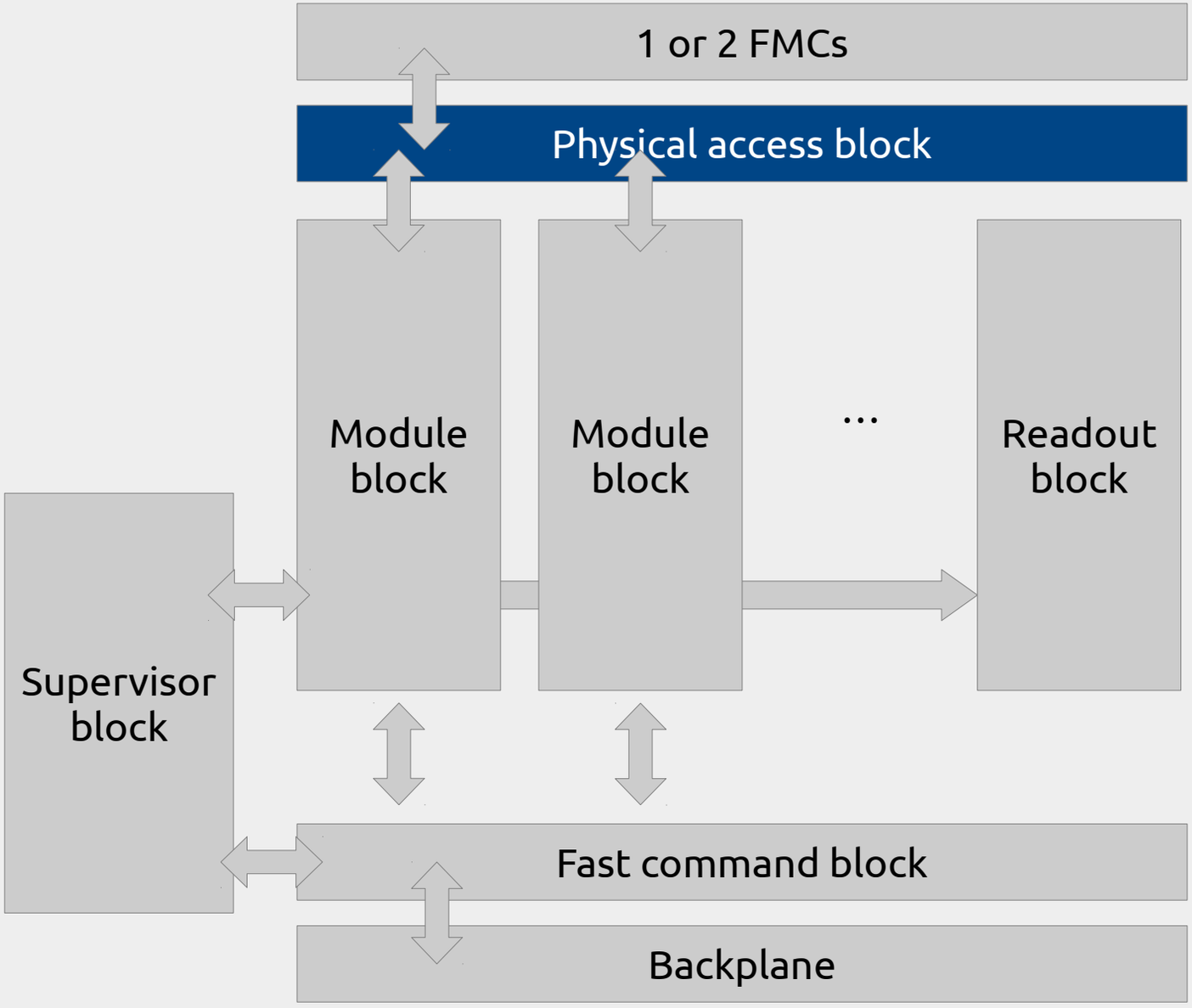
Firmware design proposal



Top level

- Blocks to be implemented in the user code
- Relies on the standard FC7 system firmware
  - IPbus
  - Memory access
  - …

Firmware design proposal

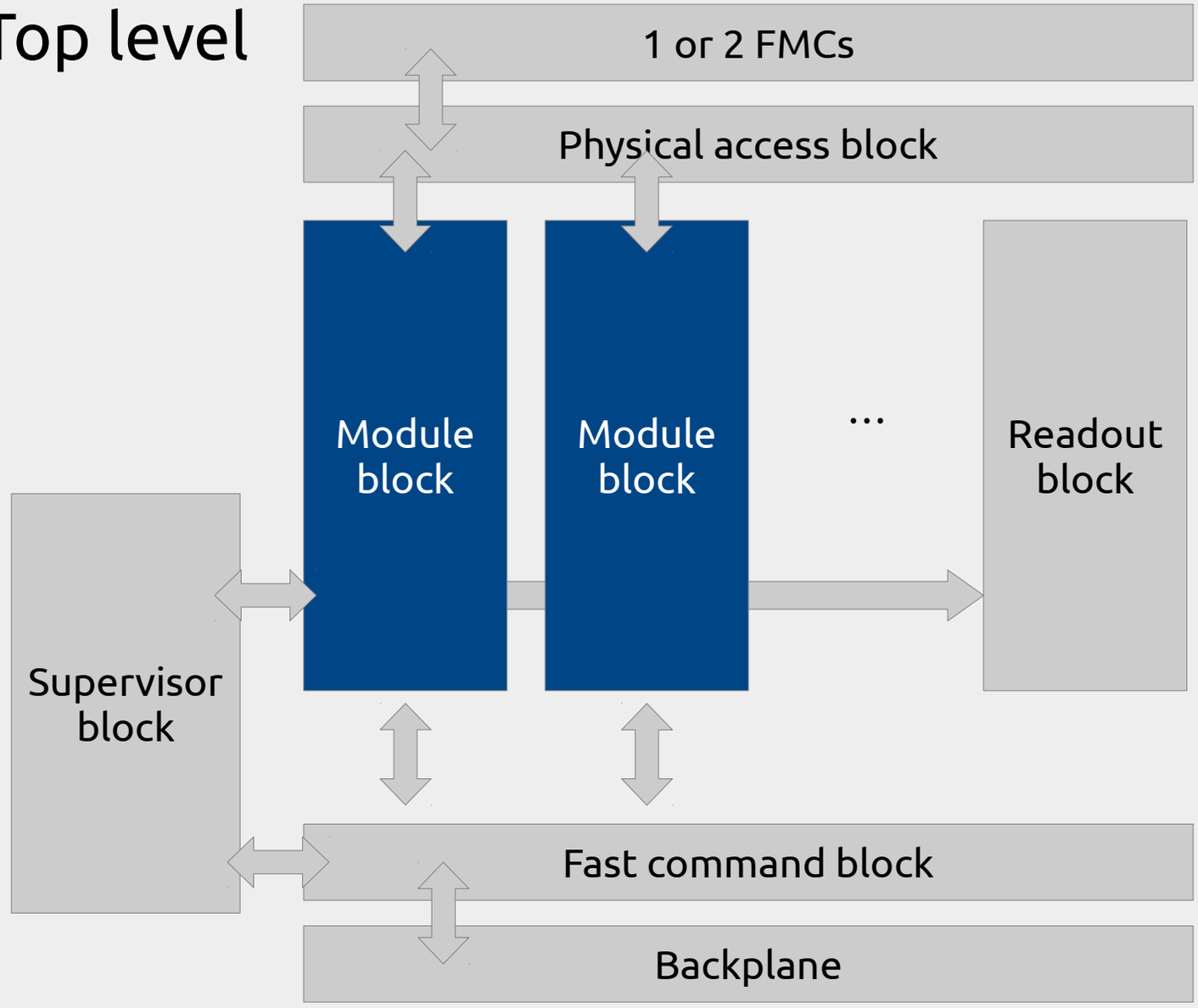


Physical access block

- Two working modes:
  - Optical: wrapping around N×GBT-FPGA blocks → SFP+ FMC
  - Electrical → interface FMC (must implement I²C master)
- Presents the same interface to the module block
  - Module does not know in which mode the system is working
  - The only difference is the number of available links
- Detects FE power-on and sends command to module

Firmware design proposal

Top level



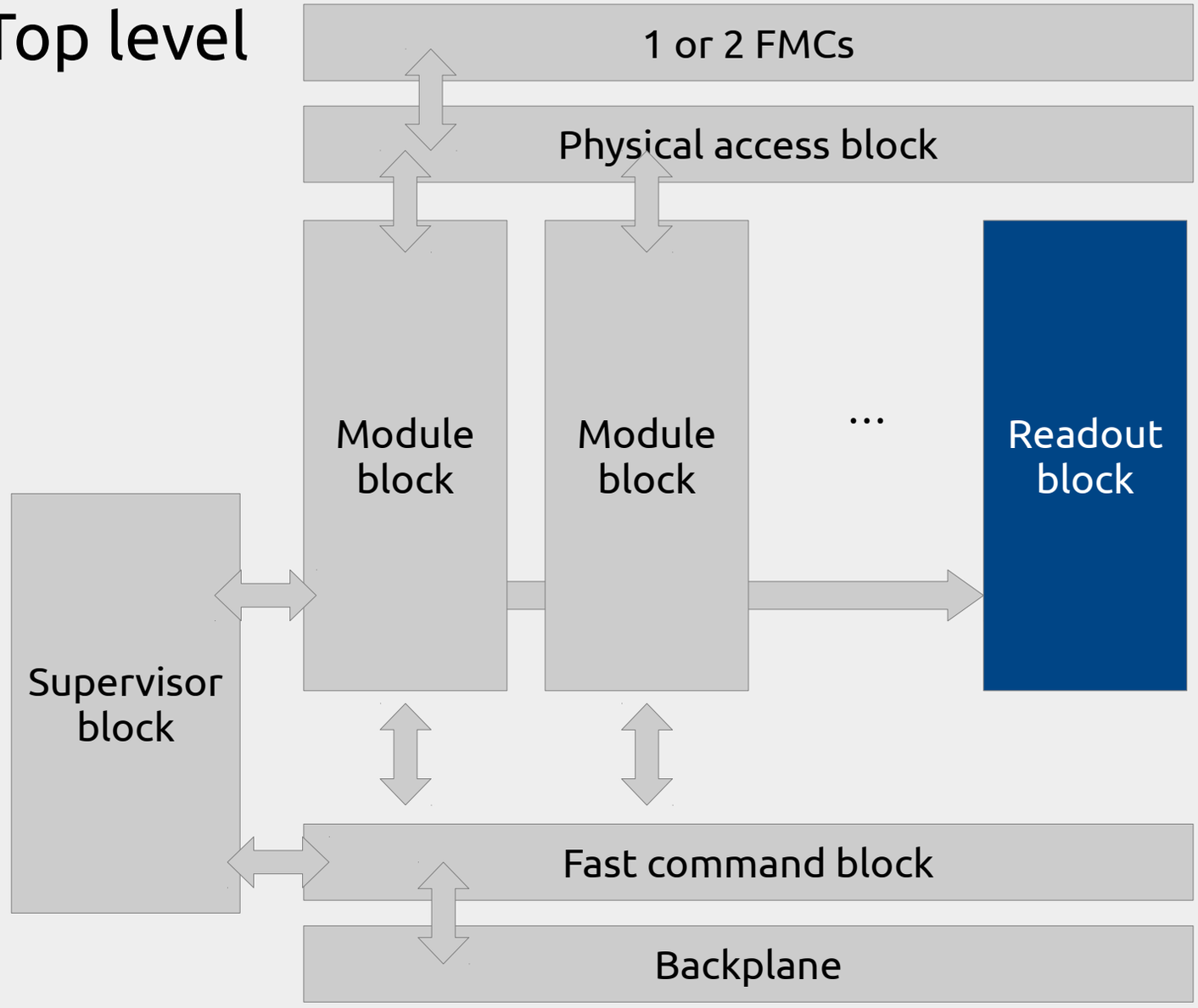
Module block

- Transmits slow and fast commands
- Receives and unpacks data
- Syncs trigger and stub data
- Can forward stub detection to fast command block
- Sends data to the readout block
- Can pre-process hits data for calibration (hit or stub counting)

(more details further on)

Firmware design proposal

Top level

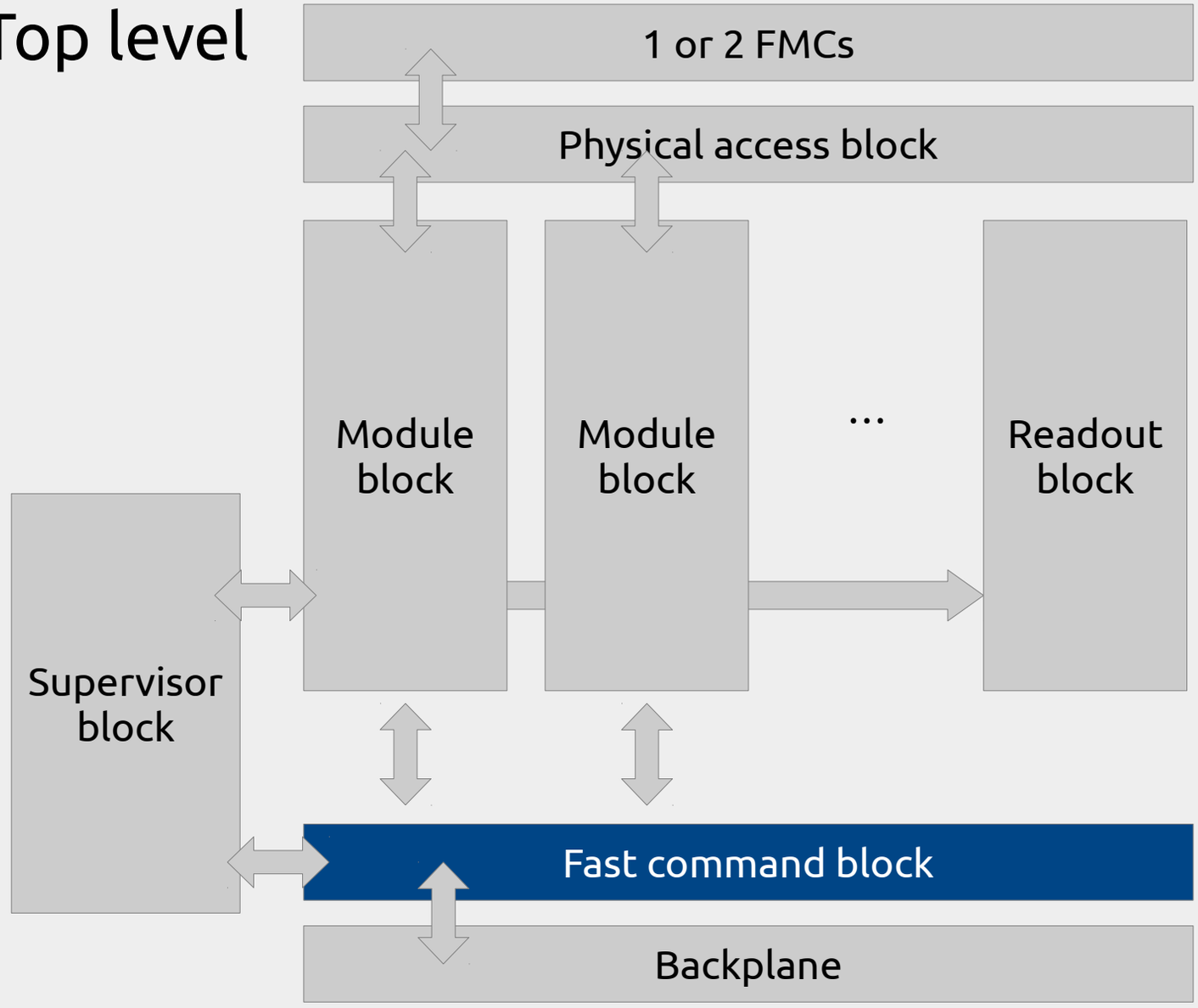


Readout block

- **Data mode**:
  - Manages memory readout and back-pressure
  - Optionally could stream data to SFP+ connectors on the FMC (iif optical readout is present, one connector can be dedicated to fast streams)
  - Data formatting for the DAQ, according to payload specs
  - Performs post-scaling (to exercise fast trigger rates with limited bandwidth)
- **Count mode**:
  - Used for commissioning: holds hit/cluster/stub counts *per strip*
  - Takes care to add counters properly – as many counters as conditions (e.g. thresholds)

Firmware design proposal

Top level

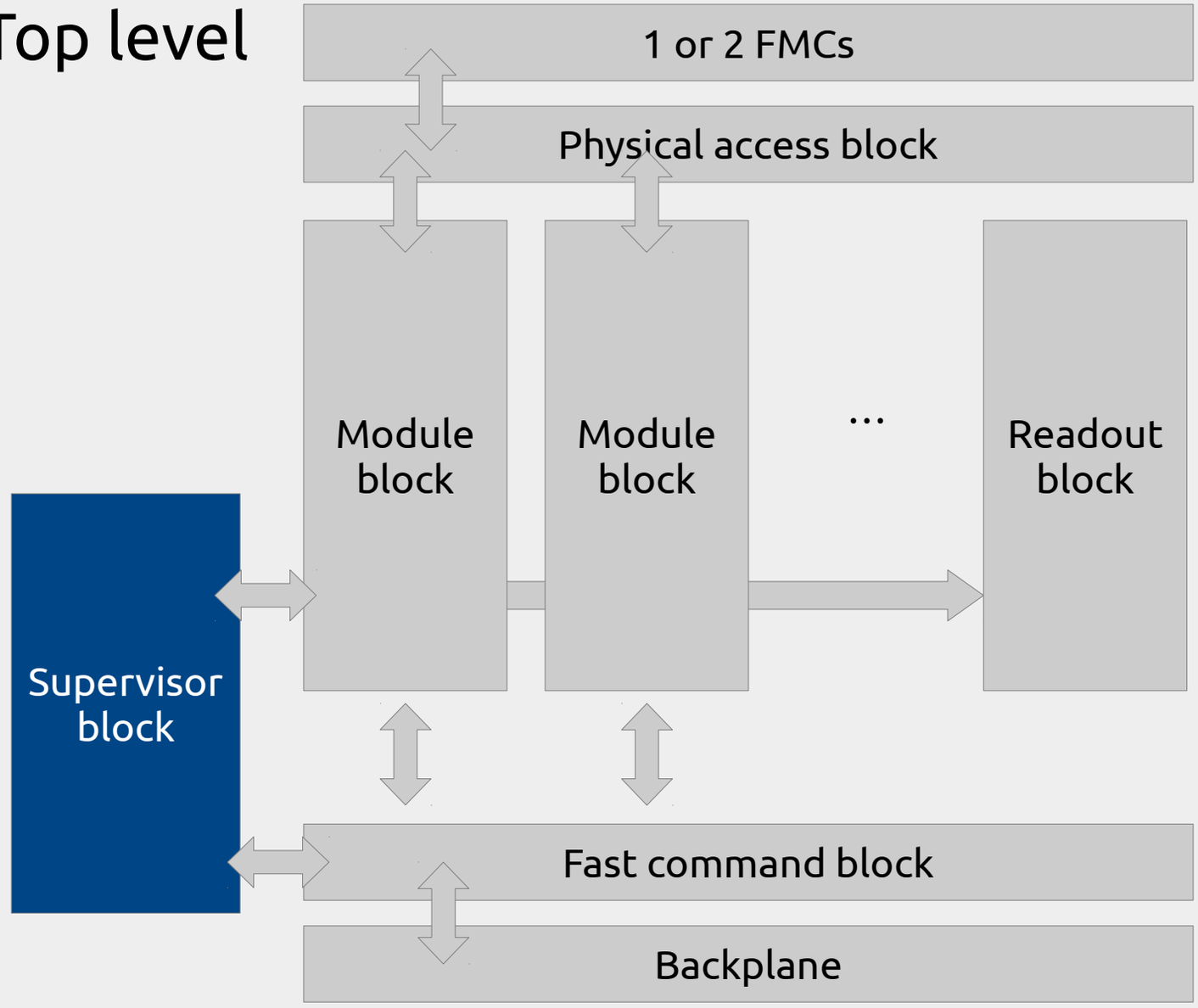


Fast command block

- Recovers clock and trigger from backplane
- Can generate clock and periodic triggers
- Holds a trigger counter: it can be programmed to accept next N triggers and then hold the next incoming triggers
- Can issue a trigger upon reception of a stub from modules

Firmware design proposal

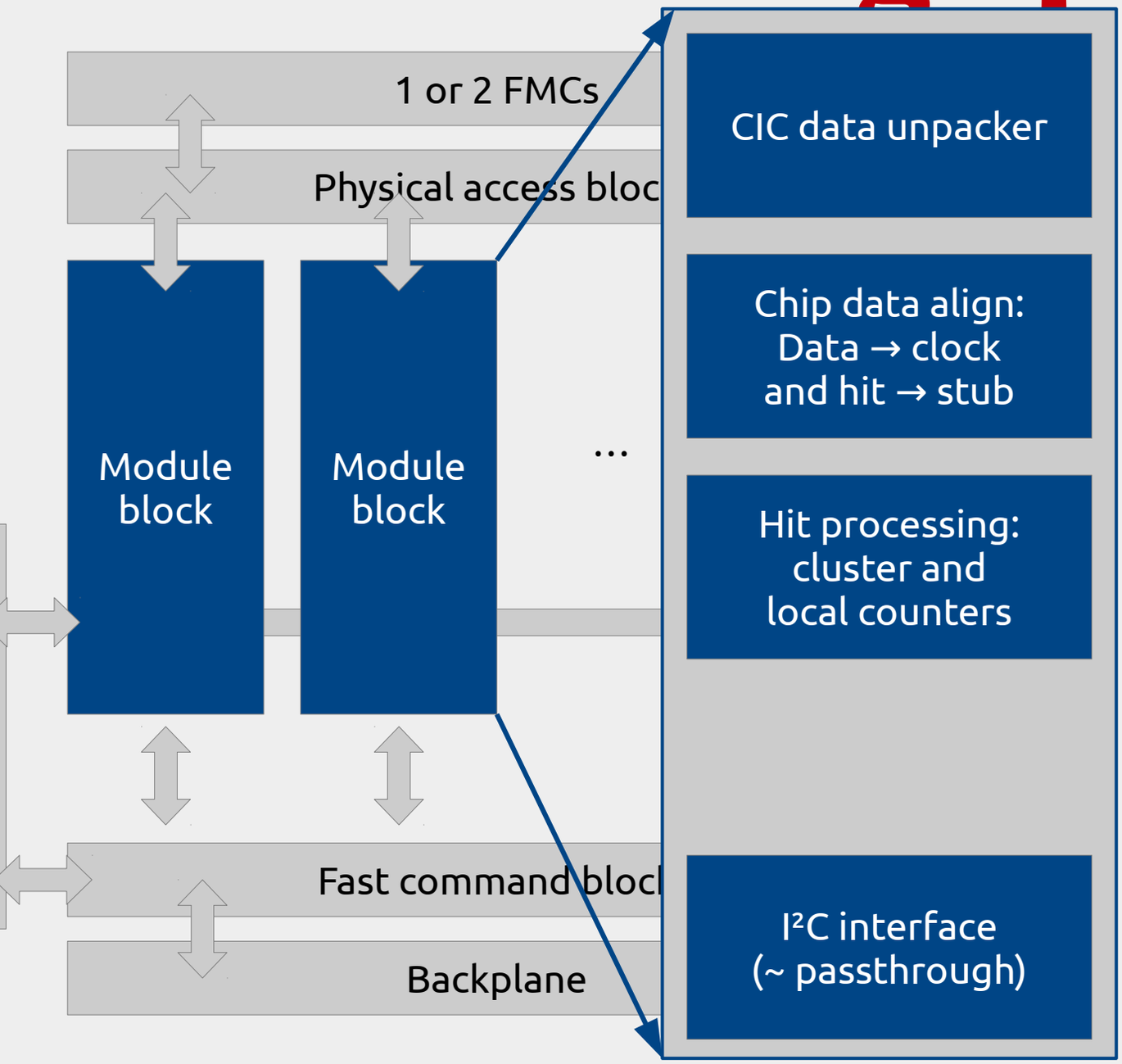
Top level



Supervisor block

- Holds a local stack of operations to be executed sequentially:
  - I²C transactions for F.E. (via module)
  - Generate (or accept next) N triggers
  - Push hit counters to appropriate memory location
- Significant reduction of IPBus transactions needed for a calibration operation (à la pixel DTB)

Firmware design proposal



Module block (detailed)

- **CIC unpacker**: (if needed) splits stub & hits streams, otherwise ~ passthrough
- **Chip data align**
- **Hit processing**:
  - **Hit count** can be activated to count hits (or clusters or stub) per strip locally
  - Always sends stub to trigger with fixed latency
  - When requested, queues data to readout block

Development proposal

- If no objection here this architecture will be circulated in the DAQ working group
- Groups/individuals should make their interest known
  (dedicated e-group should be formed)
- During the next ~2-3 weeks:
  - collect feedback on the architecture
  - form a small working group to study initial definition of interfaces between blocks
- Before end of November:
  - Draft specification of all blocks
  - Call for a round-table meeting to review the specs

- Open questions:
  - Details of interfaces (see above)
  - Blocks for PS modules (as much as possible compatible with 2S architecture)