MANCHESTER 1824
The University of Manchester

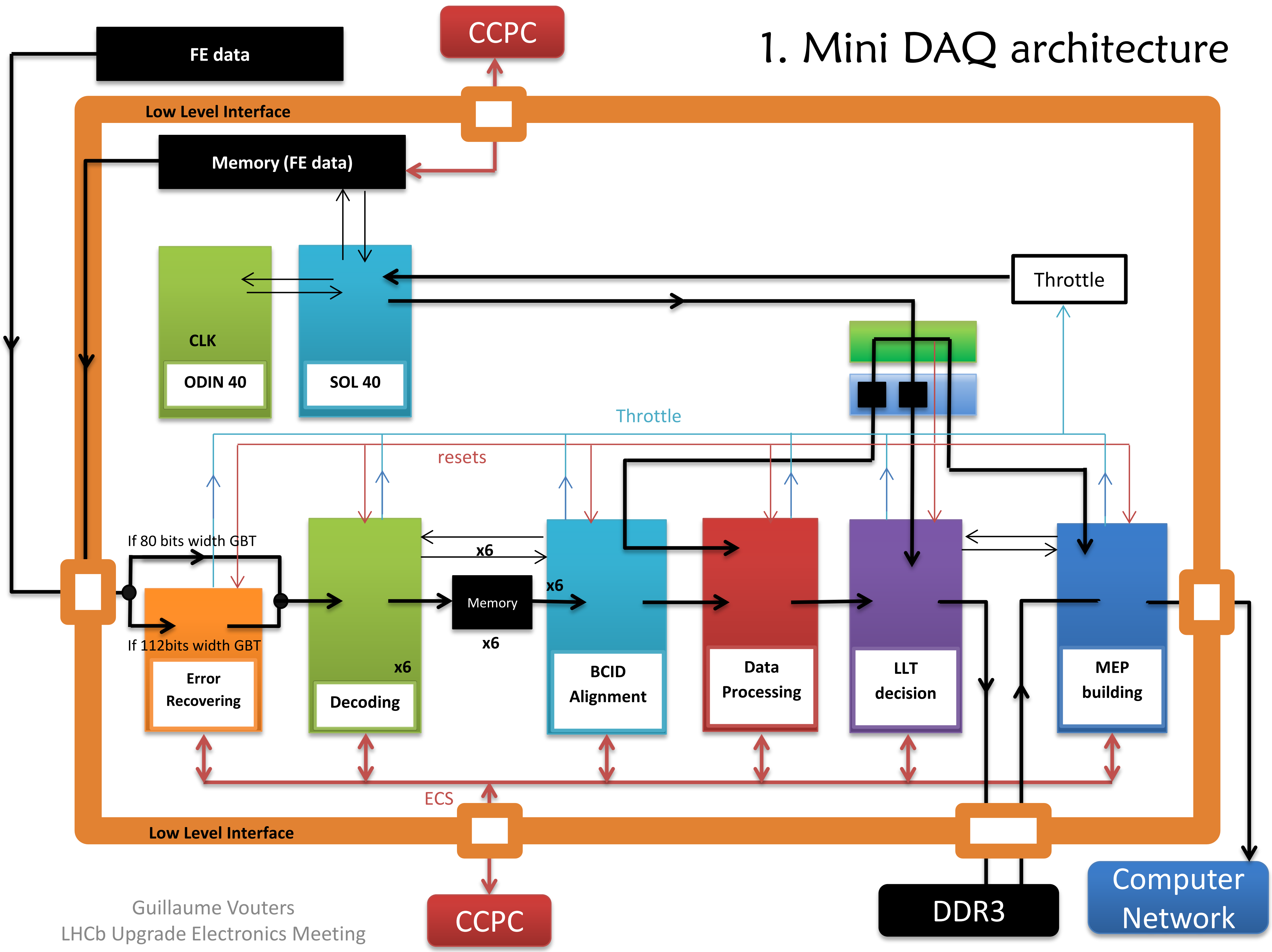
LLI and Decoder for VELO DAQ

Thanks to Jean-Pierre, Frederic, Guillaume, Federico, Tuomas and Martin for input!

Marco Gersabeck, Stefanie Reichert, Pablo Rodriguez Perez
University of Manchester

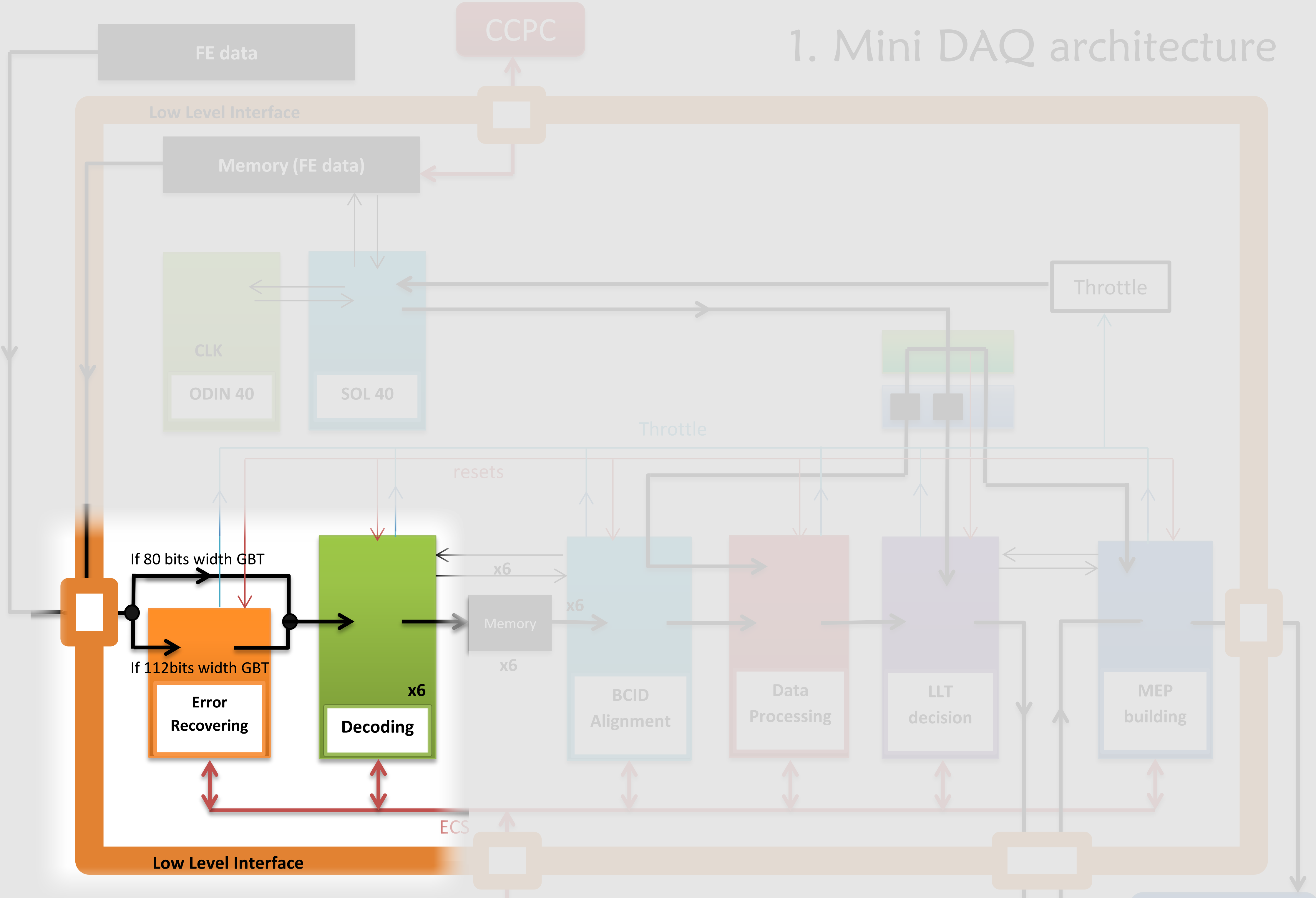
LHCb Upgrade Electronics Meeting
CERN - 12 December 2013

1. Mini DAQ architecture



Guillaume Vouters
LHCb Upgrade Electronics Meeting

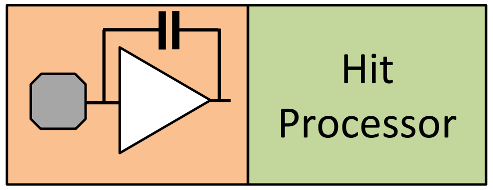
1. Mini DAQ architecture

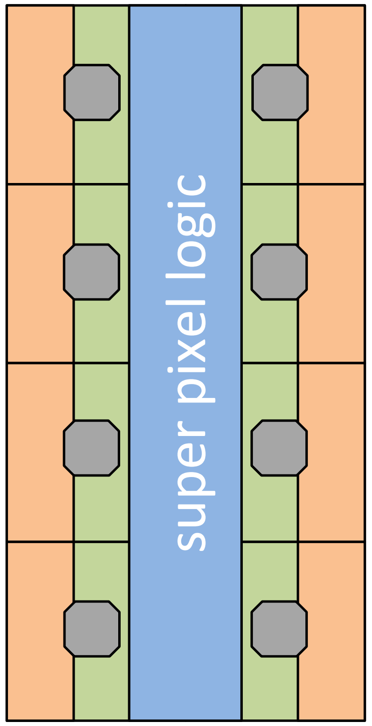


The University of Manchester

Super pixel packets

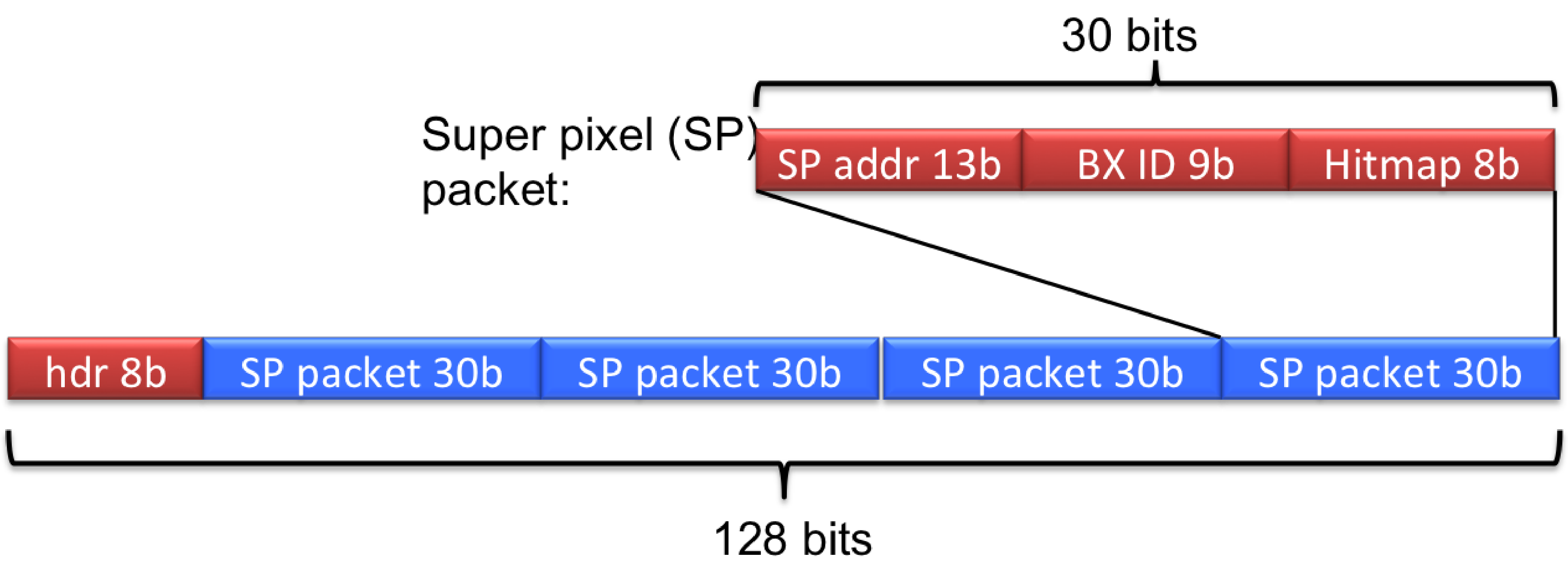
- Typical cluster size of 2 .. 4 pixels
- -> beneficial to combine 2x4 pixels in a so-called super-pixel
- Removes duplicate address and timestamp information compared to single pixel data packets
- Bandwidth gain of 30-40%
- Super-pixel (SP) has fixed boundaries
- Share logic in centre of SP
  - requires less area for routing





GWT protocol

- ~900 Mhits/s for hottest ASIC
- Packed into super-pixel packets: 2x4 pixels
- ~520 Mpackets/s, 30 bits each
- -> required effective bandwidth ~16 Gbit/s
- 4 SP packets in 128 bit frame
- 8 bit header: 4 bit fixed (0x5) and 4 parity bits
- SP packets are scrambled to reduce probability of long 0/1 sequences

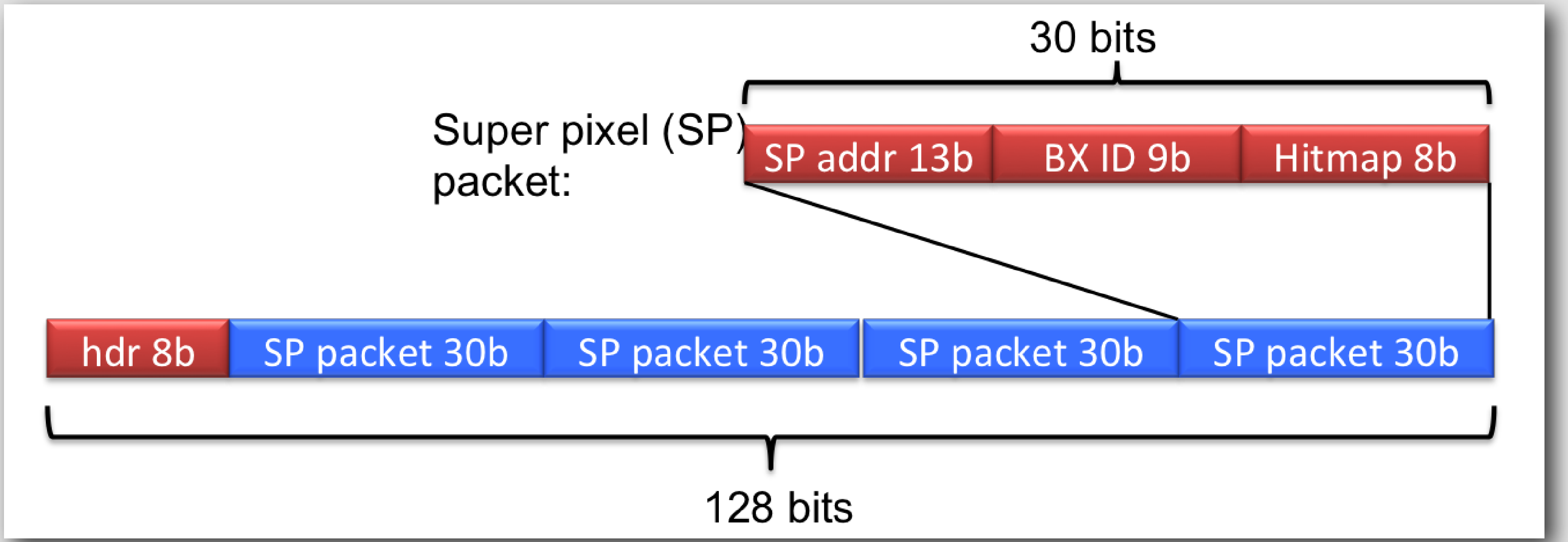


MANCHESTER 1824
The University of Manchester

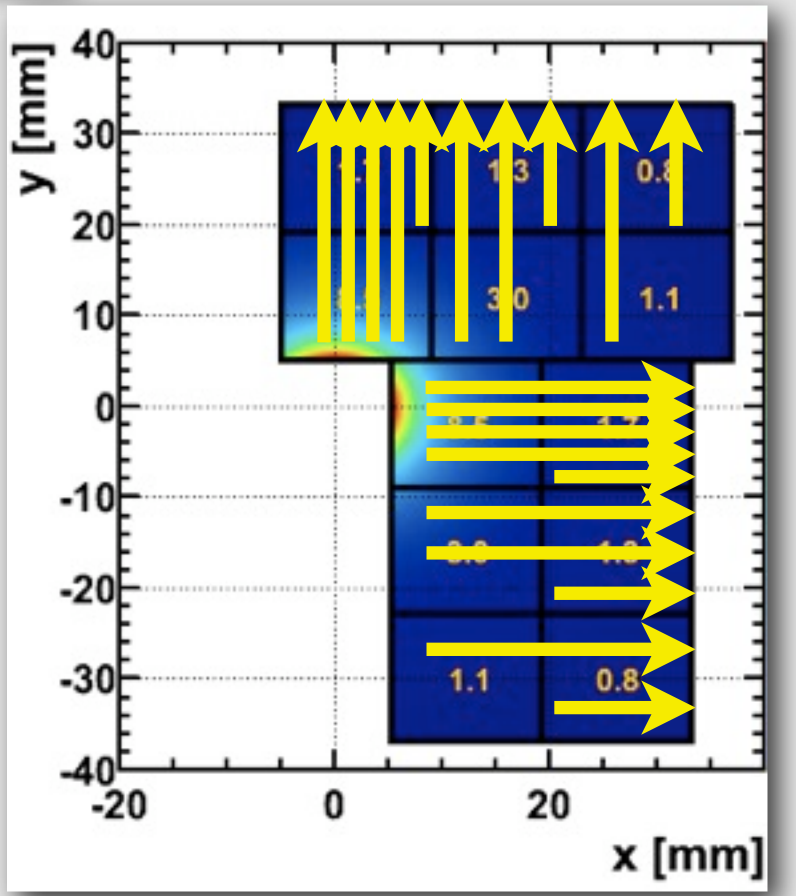
LLI for GWT

- Current implementation for GBT consumes 12% for 24 links
  - ➡ 4% for Reed-Solomon decoder
  - ➡ Assume 8% resource usage for GWT
- Scaling to 20 links: <7%
- 30-bit de-scrambling more expensive than 28-bit
  - ➡ Assume <10% resources
  - ➡ Can scramble with 2x15bit if needed
- VELO-specific implementation to be done
  - ➡ Will be based on Stratix V GBT code once available

Decoder for GWT



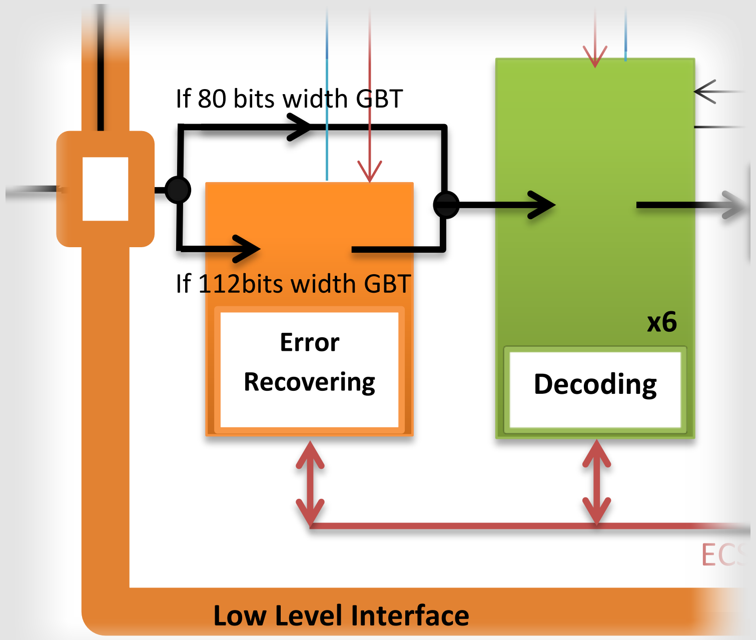
- GWT frame can contain different BCIDs
  - ➡ No need to extract BCID from header
- Just need to extract 4 30-bit SP packets (SPP)
  - ➡ 2^4 de-MUX < 50 ALMs
- Need to add 4-bit ASIC ID to each SPP
  - ➡ Constant ID per link = decoder
  - ➡ 34-bit packets sent to router
- Total: 20 x 50 x 2 (safety) = 2k ALMs ~ 1% FPGA



Total resources

- LLI: <10%
- Decoder: ~1%
- Assume for now <15% together
  - ➡ Possibly <10% feasible
- Concentrated on ALMs as memory usage is uncritical for these parts

Total resources



- LLI: <10%
- Decoder: ~1%
- Assume for now <15% together
  ➡ Possibly <10% feasible

- Concentrated on ALMs as memory usage is uncritical for these parts

Further steps

- Implement LLI + Decoder for GWT
- Work on extracting hits from full MC
  - ➡ Use as input for stress-testing firmware
- Work on full software emulation
  - ➡ Longer term