Thoughts on hardware simulation and test-vectors

- Motivation
- First attempts
  - Successes
  - Problems
- Current simulation
  - Architecture
  - Status
  - Plans

Motivation for simulation

- Online Monitoring
  - Read data from same event in several places
  - Simulate system response for input data
  - Check for differences
- Test Vector Generation
  - Complex system – many components
  - Difficult to maintain ad-hoc test vectors for each part of system, and know the exact response everywhere
  - Need to rationalise generation and vector storage

First attempts

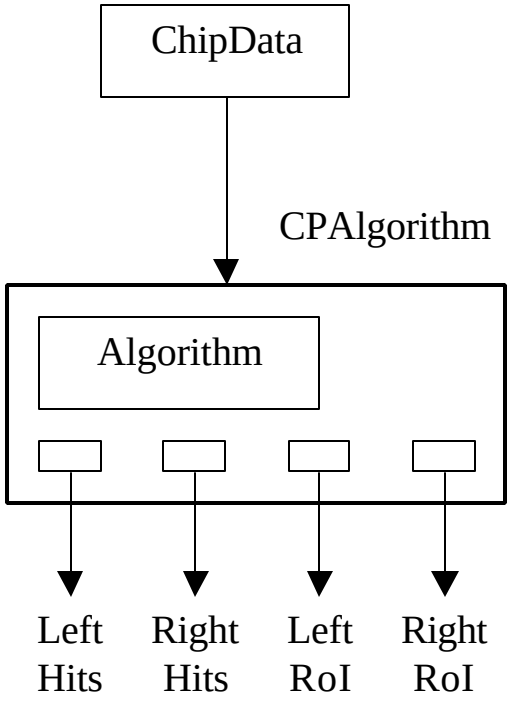
- Simple set of C++ classes used to generate test vectors for CP FPGA
  - Added BC-Mux simulation to already existing data
  - Concentrated on data i/o
  - Used simple pre-defined file formats for a single chip
- Successes
  - Detected mismatch in CP Chip Specifications
  - Data i/o class was genuinely re-usable!
- Problems
  - No easy way to scale to larger system
  - Algorithms did not fit well into architecture

Current Simulation Architecture

- Similar to a very much simplified VHDL
  - Basically two types of object
    - DataPort – cf VHDL ports – interface between processes
    - ProcessElement – cf VHDL entities – processes, algorithms
  - Hierarchy of elements possible
  - No detailed timing simulation (one step = one tick)
- File i/o now provided by specialised processes
- Currently configured by one procedure - creates modules and their connections

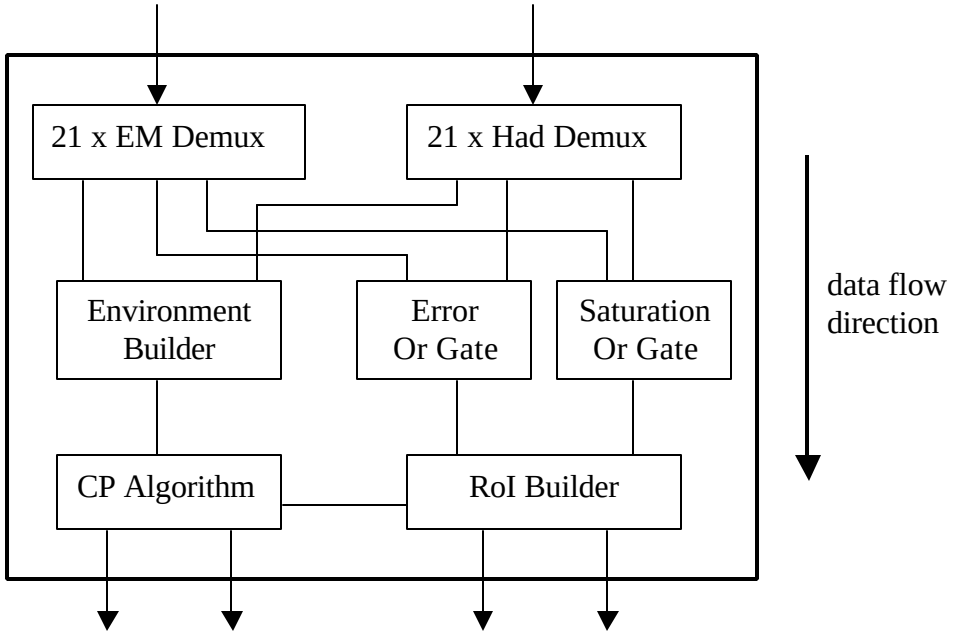
Example – CP Algorithm

- Input
  - ChipData – contains 5x7 EM+Hadronic cells
- Output
  - Left Hits
  - Right Hits
  - Left RoI
  - Right RoI
- Internals
  - CP algorithm



Example – CP Chip

- Can build individual blocks into a container object
- Complexity hidden in CPChip class internals
- Externally, appears exactly the same as a non-composite process



Status

- Have built an (almost) complete CPM
- Connected a few CPMs into backplane
- Tested with original test vectors, duplicated in eta/phi space
  - Difficult to be sure results are right
  - Little testing done on error conditions
- Working on integrating Bill's test vector generation techniques

Plans

- Write ROD simulation, test against Bill's work
- Write more complex test vector generators for detailed study of components
- Work on input/output data formats
- Re-write using ATLAS standards + offline code
- Many other questions to answer:
  - Think about need for further structure?
  - Integrate configuration database?
  - Integrate into online software?