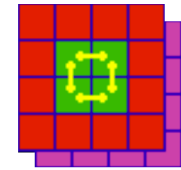




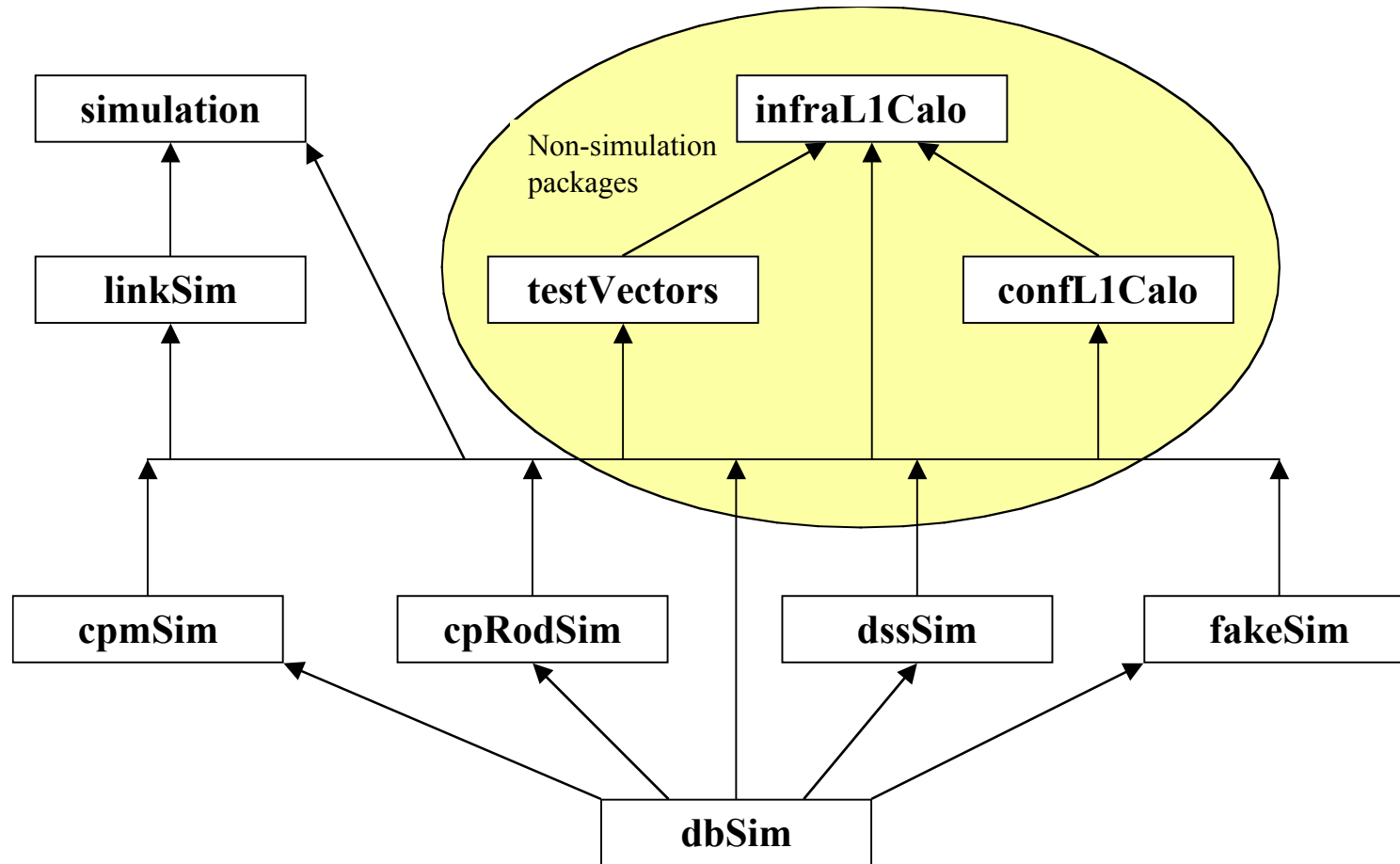
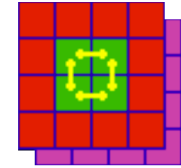
Simulation and test vectors



- Last reported at Heidelberg meeting
- As you might hope, lots of progress since then!
- Major areas of improvement:
 - Code split into several packages and put under CMT
 - Development of common L1Calo classes
 - L1A, BCnum and EventId integration
 - Addition of DSS simulation
 - Additions to CpRod and CPM simulations
 - Integration with database for:
 - Module creation and connection
 - Module settings
 - Common test-vector reading scheme
 - Test-vector generation and simulation scheme with run control
 - Finally, integrated test with DSS/ROD test system

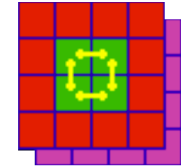


CMT package structure





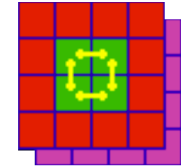
Common L1Calo classes



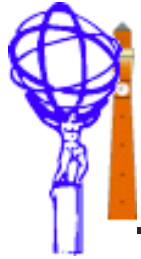
- simulation package provides a flexible generalised framework
- linkSim adds L1Calo specific stuff
 - Base classes for modules, crates
 - L1CaloSimModule, L1CaloSimCrate
 - Classes for objects shared between modules
 - cable connections - LvdsCable, GlinkStream
 - crate backplanes - CpBackPlane
 - Other shared implementation
 - TTC information and connections



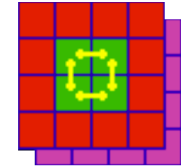
TTC information



- Needed way to distribute TTC information in the simulation
 - Addition of TtcInfo class along with TtcInfoReader
 - All L1CaloSimModules have access to this information
 - Some (most) need for data stream
 - Provides:
 - Trigger (L1A)
 - Bunch-crossing number
 - Event Id
- Also need way to generate the information
 - Currently a zeroth order scheme implemented
 - Will be done in hardware by DSS with special load
 - Still need to simulate this DSS mode in simulation



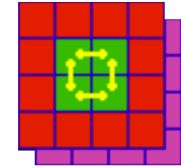
DSS Simulation



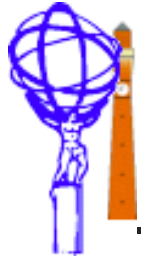
- Partial implementation provided
 - Glink outputs
 - Slink input
 - ie all that is needed for the DSS/ROD test
 - Will need extension as tests proceed
- Why is it needed?
 - Common test-vector interface for hardware and simulation
 - Test-vector input done in hardware via DSS playback memory load, so copy this in software.
 - Mechanism is also needed for modules with playback memory
 - Aside: led to inclusion of new general playback memory class in the basic simulation library



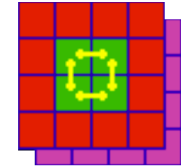
Additions to CpRod and CPM Models



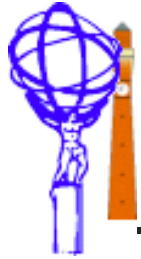
- Use common Glink class
- CPM:
 - Add playback memory functionality
 - Add Scan-path configuration
 - Integrate with L1A scheme
- CpRod:
 - Add more configuration options
 - Cope with 'dead' channels
 - Proper module/channel number setting
 - Better output file information
 - Integrate with L1A scheme



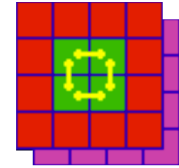
Integration with database



- New package dbSim
 - Code for running simulation from database
 - Code for generating test vectors
- Simulation run
 - Completely general
 - Modules and connections generated automatically from database hardware configuration
 - Module setting also read from database
 - Test-vectors loaded from database according to newly defined scheme
 - Important that it is general as should also cope with real data
- Generation run
 - Less general
 - Need to write a new class for a new test setup
 - Class should check database settings and warn if inconsistencies
 - Currently can generate all test vectors that I know about
 - CPM crate tests, Bill's test vectors, Bruce's test vectors



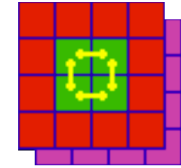
Test vector reading scheme



- New package testVectors
 - Used by simulation and moduleServices
 - Common interface for loading playback memories
 - Agreed over several meetings with Bruce and Murrough
 - At present just reads files generated by standard mechanism
- Future Direction
 - Needs to cope with more than single shot run
 - Possible test vector generation on the fly?



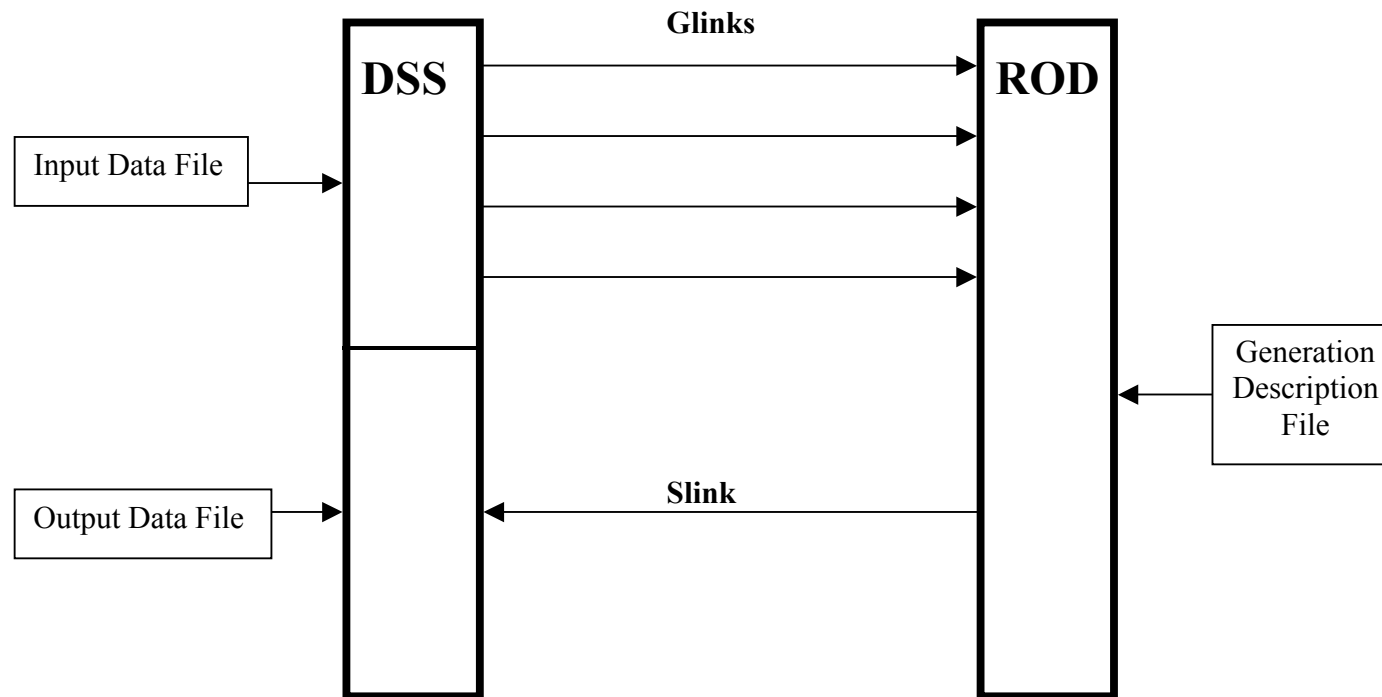
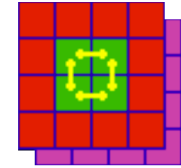
Run control integration



- Simple interface to dbSim
 - Test vector generation:
 - Create DbGeneration - dg(database)
 - Run it - dg.run()
 - Delete it - delete dg
 - Hardware simulation:
 - Create DbSimulation - ds(database)
 - Run it - ds.run()
 - Delete it - delete ds
- Murrough wrote a simple run controller to do this
 - Note - has to be run before data loading into modules
- It appears to work

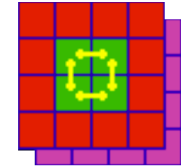


Integration Tests: Setup





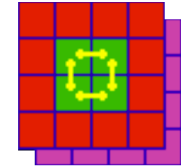
Integration test - results



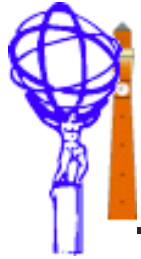
- Attempt to run all this together
 - 28th August RAL
 - Unfortunately there were (known) hardware problems
 - However, integrated test was consistent with old software behaviour
 - In fact it worked slightly better!
 - Database allows easy disconnection of bad channel
 - Simulation can cope with 3 channels, old software can't
- Conclusion
 - The software works, as far as we can tell
 - Needs more careful checking with working hardware



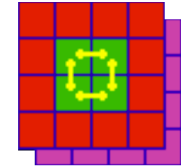
Plans



- Immediate
 - Check DSS/ROD more thoroughly
 - Test new software with all old test vectors..
 - ..plus a few new ones I'd like to try
 - Finish work on test vector generator for CPM scan-path mode
 - Integrate cpmServices with database, run control
 - Try simple CPM tests from Run Control
- Longer term
 - CPM improvements - deal with >1 slice readout
 - ROD improvements
 - More algorithms (CMM, JEM)
 - Cope with more than one data type per module
 - TTC integration - generate DSS contents and L1A test patterns



Other news



- From hardware phone conference
 - Paul is working on PPM simulation
 - Seems to be quite happy
 - Sam's students are still working on JEM
 - Good news: they are starting to use my framework