Online SW for Phase 1

Murrough Landon
27 January 2015

- Requirements
- Proposals
- Timescales
- Effort?

Requirements (1)

- What we had for the original system
  - Low level VME model and debugging tool: HDMC
  - Database interface (OKS, menu, COOL)
  - Hardware configuration for TDAQ ("module services")
  - Online simulation (test vectors, spy memories, readout)
  - Calibration and test programs
  - Low level monitoring and diagnostic tools
    - Rodmon, event dumps, mapping tool, rate metering, etc
- Generic tests of full or partial system
  - Setup OKS database
  - Generate test vectors
  - Run hardware and online simulation
  - Compare results from spy memories and/or ROD readout
  - Works for any subset of the full system

Requirements (2)

- Probably need much the same for phase 1 system
  - Easiest to continue in same L1Calo style
    - Common framework for all FEXes
- IPbus debugging:
  - Low level interface: uhal library from CMS
    - Now included in TDAQ release (from tdaq-05-05-00)
  - GUI: I recently made an HDMC-like prototype
- Database interface
  - Present module developers with one DB interface collecting information from OKS, trigger menu and COOL
  - Worked OK so far - but mainly useful as a common interface between hardware configuration and online simulation

Requirements (2)

- Hardware configuration
  - Already extended VME module services to "module control" package for L1Topo
  - Propose to continue this style for FEX run controllers
- Online simulation
  - Original system: online simulation totally separate from offline
  - L1Topo: algorithms in Athena-free DetCommon packages
    - Then packages in online simulation framework to provide line to online test vectors, database interface and readout
  - What approach should we take for the FEXes?
    - Simple(ish) algorithms: just implement in online simulation classes
    - Or follow L1Topo module and use DetCommon
  - And how do we run tests with DPS+FEX+L1Topo?

Requirements (3)

- Calibration: simpler in future?
  - Currently mostly for PPM
    - Will move to DPS (somebody elses problem!)
  - Plus CPM/JEM timing setup
    - Optical links automatically align themselves via K characters?
    - So just need to set BC delays between inputs?
- Test, monitoring and diagnostic tools
  - More complex mappings
  - More channels for display tools
  - Etc?

Miscellaneous: RC States

- Run control states: original L1Calo scheme
  - For historical reasons we use substeps of RC Configure step (and almost ignore Connect):
    - Configure (l1caloLoad substep):
      - Initialise FPGAs and clocks on each module: output links should be stable
    - Configure (l1caloConfigure substep):
      - Try to connect input links, load calibration, menu, test vectors etc
    - Connect: barely used (if at all)
  - CMX issue
    - Output link clock only established after input links are connected
    - L1Topo will need to use the Connect step to connect its inputs
    - Not suitable for a long pipeline of optical links: extra substep per stage!
  - Would like FEX firmware to be able to configure output links without needing input links connected
  - Is this feasible?

Miscellaneous: Naming!

- Software naming conventions
  - Typically (following original ATLAS SW rules) we have:
    - packageNames
    - ClassNames
    - methodNames
    - variableNames
    - and DISLike lots of CAPITALLetters run together
  - So things like eFEXClass would be disfavoured
  - Propose EfexClass, efexPackage etc
    - Unless there are loud and persistent howls of outrage

Timescales

- 2015: Link speed tests
  - Probably mostly carried out with firmware tools?
  - But useful to configure/view module via IPbus GUI?
  - Use GUI to develop robust configuration sequences?
- Later?
  - Implement HW configuration packages to run under TDAQ
    - Using procedures derived from GUI based tests and simple scripts?
  - Get simulation and test vector comparison going
  - Extend other existing tools, write new ones as needed...

Effort

- Review who is available for online SW
  - Generally need one person per hardware module based at the institute developing the hardware or firmware or where prototype testing will be carried out
  - Additional person (per FEX?) taking care of simulation?
  - One person dealing with common issues (database, etc)
  - Plus people for monitoring, other tools, etc?