Proposed Changes to HDMC

Murrough Landon – 28 June 2001

http://www.hep.ph.qmw.ac.uk/~landon/talks

Overview

- Introduction
- Packages and libraries
- Parts and the PartManager
- Modules and composite components
- Registers
- VME access
- GUI developments
- New parts
- Integration with Online software
- Parts file syntax

HDMC Changes (1)

Introduction

- Around and after the time of the last meeting Bruce and I (mainly) had some discussions about changes to HDMC required to allow its use in other parts of our software, ie to provide a “hardware access library” functionality to “DAQ world” packages.
- I wrote up my views, with comments from Bruce, into a document which is available at
  `http://hepwww.ph.qmw.ac.uk/l1calo/doc/pdf/HdmcChanges.pdf`
- Oliver knows about it...
- ...but no chance yet for response and discussion

Packages and libraries

- Presently HDMC is a single package.
- Change to separate general hardware access library from GUI application
- NB future use of new ATLAS software release tool (CMT) and possible pressure for common CVS repository.

HDMC Changes (2)

Parts and the PartManager

- Presently Parts are only created by the GUI or reading a Parts file
- Need to be able to create them by other software
- Would like to allow Parts which themselves create their own known set of component Parts
- Possibility to #include Parts file within other Parts files (saving any changes back to the relevant files)
- Requires changes to Part and PartManager classes

Modules and Composite Components

- HDMC Module class is just a generic container
- Duplicated Modules can diverge in their contents
- Would like to treat a Module (subclasses) as a single entity, so that changes propogate to all instances of it
- Also useful to have a SubModule class which can be part of a Module, but also host Registers etc.
  NB this has now been implemented.

HDMC Changes (3)

Registers

- This is a very complex class (particularly the GUI) with many supporting classes
- Warning: difficult to understand and maintain and thus might act as a brake on other developments

VME Access

- At present only A24 access is implemented (I think?)
- Not clear if present Address classes are sufficient to extend this to A32.
- Some redesign of Bus and Address classes may be desirable
- Presently offsets are variously expressed in bytes, words or longwords.
- Less error prone if everything is in bytes – but repercussions on existing parts files

HDMC Changes (4)

GUI Improvements

- Some bugs and deficiencies in the ModuleView reported in the document are now fixed
- Could still do with extending ModuleView to cover components other than Registers
- What to do about Module subclasses?

New Parts

- Document not really concerned with new Parts, but...
- Need extension to existing FPGA classes
- Parts for composite (firmware) components, eg Serialiser FPGA etc
- Playback memories: subclassing of Memory classes to add specific decoding?
- FIFO: ie memory accessed by pointer and reset registers (mainly a GUI issue)
- TTCrx: description of its internal registers and access via our (common?) programming model scheme
- etc
- Also with associated GUIs in many cases...

HDMC Changes (5)

Integration with Online Software

- Above mostly assumes standalone operation or use as hardware access layer for other L1Calo software
- Do we need to integrate HDMC itself with (some components) of the Online software?
- Maybe if we change the Parts and Configuration file syntax, we should use Online database package?

Parts File Syntax

- HDMC reads two files: parts file and configuration file
- Same basic syntax, but handled differently internally
- Configuration file is edited by hand and defines the Register formats for use by the GUI
- Parts files are maintained automatically by HDMC itself via the GUI
- Long term intention to replace private syntax by XML – but never implemented
- Could change to use Online database package: this already will contain top level hardware configuration information such as crates and modules.
- Change could involve a lot of work... but could do parts files and configuration file separately