



SLHC CMS Firmware

SLHC CMS Firmware Organization, Validation, and Commissioning

**M. Schulte,
University of Wisconsin**



Current Status of CMS Firmware

- **Different subsystems often have much different methods for**
 - Storing and versioning firmware
 - Developing and maintaining firmware
 - FPGA and CPLDs utilized
 - Tools employed
 - Languages used
 - Downloading, testing and verifying firmware
 - Documenting firmware
- **These differences make it difficult to**
 - Share firmware design and testing resources
 - Enable others to help maintain and modify firmware
 - Ensure the full system will work correctly



Firmware Methodologies

- **We should investigate and adapt consistent methodologies for developing, validating, sharing, and commissioning CMS firmware**
- **Next are some initial ideas to motivate further discussions**



Firmware Sharing

- **There is a large amount of firmware, testing infrastructure, and expertise that can be shared by CMS subsystems**
 - **RocketIO, FPGA control, sorters, adders, etc.**
 - **Developed firmware should go through a “review” process and then be made available to other subsystems**
 - **To facilitate this, consistent design, testing, and documentation policies should be adopted.**
 - **There should also be regular virtual meetings of firmware developers to exchange ideas**



Firmware Binary File Storage

- 1) Develop a database at CERN that stores all the binary files downloaded onto the FPGAs at P5.**
 - Should contain a “snapshot’ of all the firmware used for each running period marked with a timestamp
 - Enable people working remotely to request to submit firmware to this database
 - Someone local to CERN should have control over firmware submissions
 - Preliminary work on a database depository tool for CMS is being conducted



Firmware Download

2) Have a safe and consistent methodology for downloading firmware for P5.

- Utilize the database from (1) to select the binaries to download
- Provide read-back validation of downloaded binaries
- Include a CRC check to validate the integrity and version number of each downloaded binary
- Each firmware file should have its name, version number, and creation date written into the firmware file contents to detect and correct incorrect files
- May be able to leverage practices already in place by the DAQ and HCAL subsystems



Firmware Maintenance

- 3) The database from (1) should contain information to understand, modify, rebuild and test the firmware for each running period.**
- **Name, location, and version number of all software packages used to build and simulate the firmware**
 - **Compiler options used to build the firmware – when feasible default options should be used**
 - **Links to the firmware's source code, testing infrastructure, support software, and documentation.**
 - **Links to the entire “project” directory used to develop the firmware.**



Version Control

- 4) CMS should have global Subversion (SVN) repository that is used by all of the subsystems**
- **Should store firmware source code, testing infrastructure, support software, and documentation**
 - **Should be linked to using the database from (1)**
 - **Should be easily accessible to groups outside of CERN**
 - **Work on the SVN firmware repository is being performed by Christoph Schwick**
 - **This will greatly facilitate sharing of firmware and firmware reviews**



Firmware Coordination

5) Appoint a CMS firmware coordinator who

- Helps ensure that policies are in place and utilized to develop, maintain, test, and document CMS firmware
- Promotes collaboration and coordination between firmware developers, subsystem managers, and others

6) Establish a small CMS firmware committee that sets the policies mentioned in (5)

- Determines policies in consultation with the ESSC, key people from CMS subsystems, and experts in related areas (e.g., databases, testing, and version control)
- Could include the firmware coordinator, the electronics coordinator, and others with firmware expertise



Firmware Reviews and Testing

- 7) The firmware coordinator should lead a detailed review of CMS firmware**
 - Each subsystem should collect the necessary data and present it for review
 - Potentially have cross subsystem reviews
 - The online database and SVN repository should greatly facilitate the review

- 8) Establish thorough methodologies for testing the firmware before and after it is downloaded at P5.**
 - May be able to leverage firmware testing techniques to be provide by the trigger supervisor
 - Existing techniques for testing firmware should be thoroughly reviewed for all subsystems



Continued Maintenance

9) Establish policies for handling obsolescence of FPGAs, CPLDs and their tools

- Collect detailed list of FPGAs, CPLDs, and tools for each subsystem
- May be useful to stockpile older FPGAs/CPLDs
- Individual subsystems or CERN may need to maintain older versions of tools/computers to rebuild firmware
 - Investigate use of virtual machines to help with this

10) Ensure that multiple people can maintain and enhance each firmware design.

- Short term – have at least two developers familiar with each major piece of firmware
- Long term – establish consistent methodologies to develop, maintain, test, and document firmware



Unified Firmware Framework

- **To ensure the long-term viability of CMS firmware over the life of the LHC and its upgrades:**
 - It will be necessary to establish a unified framework for firmware development that all CMS subsystems follow
 - This unified framework will require a significant departure from current CMS firmware practices
 - Such a framework should greatly reduce the costs and risks associated with maintaining and updating CMS firmware
- **The following are recommendations for this unified framework**



Unified Firmware Framework

- 1) All subsystems should use the database and Subversion repositories described previously**
 - Ensure the integrity of the downloaded firmware, allow better tracking of firmware versions, and facilitate long-term firmware collaboration and maintenance
 - Proper use of these systems should be monitored by the CMS firmware coordinator and firmware committee

- 2) We should try to limit the number of different types of FPGAs, CPLDs, and tools utilized by firmware developers**
 - Facilitate better collaboration and design reuse, and ease the burden of maintaining tools
 - Remote developers should be provided easy access to any tools selected



Unified Firmware Framework

- 3) CMS should establish stronger relationships with FPGA vendors. Potential benefits include**
 - Access to roadmaps for FPGA architectures and tools
 - Additional FPGA expertise
 - Help resolving issues related to licensing and obsolete FPGAs and tools
- 4) Establish consistent methodologies to develop, download, test, and document firmware**
 - Established by the firmware coordinator and firmware committee with input from firmware developers and subsystem managers
 - Used by all subsystems



Unified Firmware Framework

- 5) Develop methodologies to facilitate upgrading FPGAs and their tools**
 - Limit use of schematic-based design and platform-specific IP
 - Limit the number of non-default compiler options used for synthesis
 - Develop modular designs that clearly separate computation from data transfers and I/O
 - Develop testing infrastructure that helps ensure firmware continues to work with new devices or tools
- 6) Develop/utilize tools that automate certain tasks including testing, version tracking, firmware verification, and some documentation**



Recommended Next Steps

- **Further discussion and refinement of recommendations**
- **Continue important work on firmware database and SVN repository**
- **Select a firmware coordinator and firmware committee**
- **Established policies to develop, maintain, test, and document CMS firmware**
- **Perform a detailed review of CMS firmware**
- **Begin work on the unified firmware framework**



Questions or Comments?

Questions or Comments?