**CMSSW Integration**

The 'top' module in firmware is the module sp. The following is the dependency graph:

* module sp
  + module prim\_conv\_sector
    - module prim\_conv
    - module prim\_conv11
  + module zones
  + module extend\_sector
    - module extender
  + module ph\_pattern\_sector
    - module ph\_pattern
  + module sort\_sector
    - module zone\_best3
      * module zone\_best (in sort\_zone.sv)
  + module coord\_delay
  + module match\_ph\_segments
    - module find\_segment
  + module deltas\_sector
    - module deltas
      * module best\_delta
  + module best\_tracks
  + module single
  + module ptlut\_address

In the emulator, the following processor classes perform the jobs of the firmware modules:

* class SectorProcessor
  + class PrimitiveSelection
    - N/A (the job is done by input links/cables)
    - Preserve Primitive selection in source code
  + class PrimitiveConversion
    - module prim\_conv\_sector
    - **EMTFHit& conv\_hit** holds the value of LCTs
      * **Wg = conv\_hit.Wire()**
      * **Hstrip = conv\_hit.Strip()**
      * **Quality = conv\_hit.Quality()**
      * **cpat = conv\_hit.Pattern()**
  + class PatternRecognition
    - modules zones, extend\_sector, ph\_pattern\_sector, sort\_sector
  + class PrimitiveMatching
    - module match\_ph\_segments
  + class AngleCalculation
    - module deltas\_sector
  + class BestTrackSelection
    - module best\_tracks
  + classes PtAssignment, PtAssignmentEngine
    - module ptlut\_address

Guidelines

* The header files are located in the folder called “interfaces”
* The source files are located in src
* SectorProcessorHLS is the wrapper function for the HLS core files
* An object of type SectorProcessorHLS is defined from SectorProcessor.hh file
* The function calls for HLS EMTF logic can be found in SectorProcessor.cc
* The inputs for HLS EMTF logic i.e LCTs are decoded from the class TriggerPrimitive defined in L1Trigger/L1TMuon/interface/MuonTriggerPrimitive.h
* LUT is filled from an object of type SectorProcessorLUT defined in Trackfinder.hh
* Use the command scramv1 ProjectRename when you change project directories
  + <https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideScram#CmsswSCRAMBuildFlags>
* The HLS source code, WITHOUT ANY CHANGES can be used for:
  + Synthesis (Vivado HLS tool defines **\_\_SYNTHESIS\_\_** macro which is used to ignore the simulation section)
  + C simulation (macro called **MY\_CONTROL\_MACRO** is used to ignore synthesis section)
  + Simulation in CMSSW by defining a macro called CMSSW\_MACRO. This is defined by adding the line **<flags CPPDEFINES="CMSSW\_MACRO="/>** in /afs/cern.ch/user/n/npratapg/public/CMSSW\_9\_0\_0\_pre2/config/BuildFile.xml.