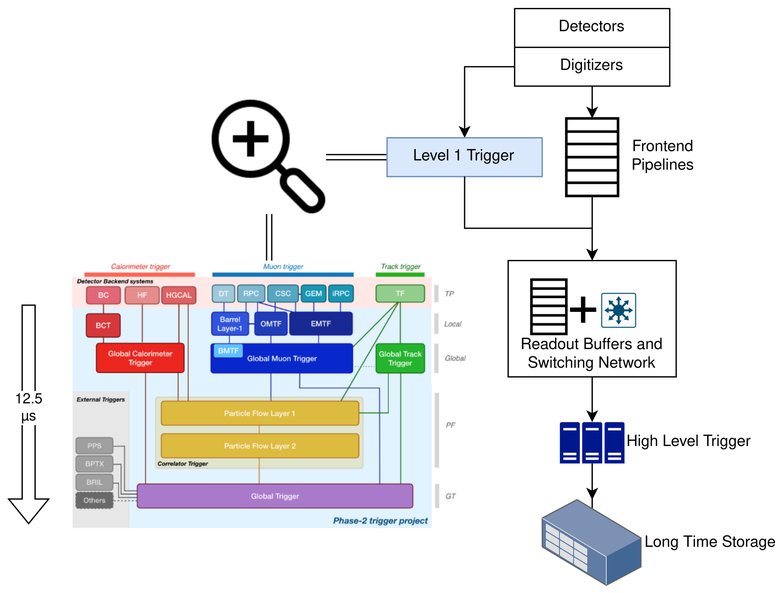
The Level-1 Global Trigger for Phase-2: Algorithms, configuration and integration in the CMS offline framework

Gabriele Bortolato[1,2], Benjamin Huber[1,3], Elias Leutgeb[1,3,✉], Dinyar Rabady[1], Hannes Sakulin[1] on behalf of the CMS collaboration

1)CERN 2)Università degli Studi di Padova 3)Technische Universität Wien

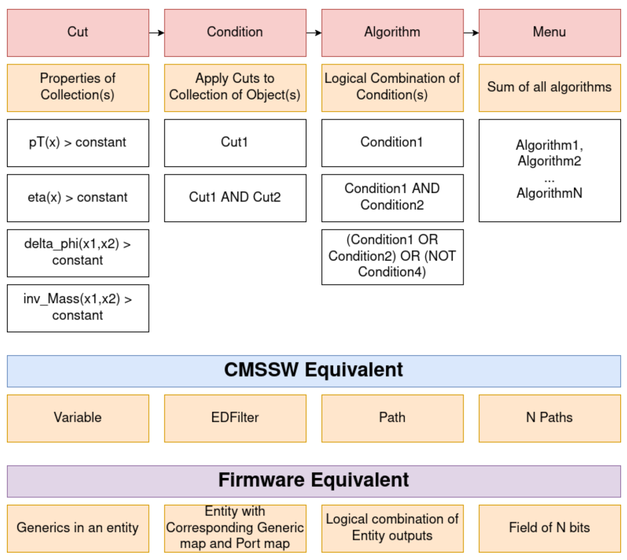
✉ elias.leutgeb@cern.ch

Overview



The Level 1 Trigger (L1T) in the CMS experiment is a pipeline of hardware (FPGA based) triggers, which receive condensed data from the CMS detector with a fixed latency of $\approx 12.5\mu s$. It decides whether an event should be read out or discarded.

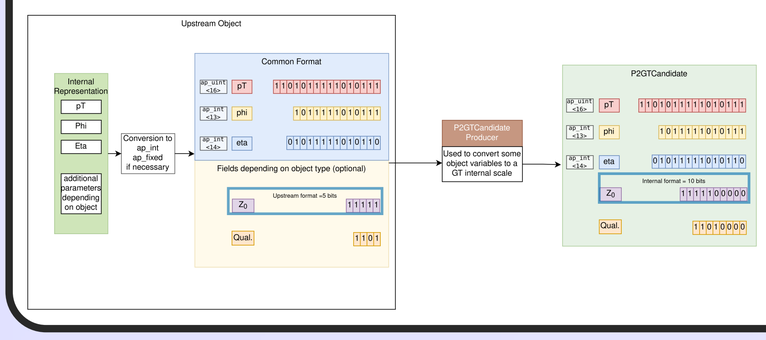
The final stage of the L1T, the Global Trigger (GT), receives data from all regional triggers as well as the Correlator Trigger. It outputs the final trigger decision *L1 accept* to the Timing Control and Distribution System. This decision triggers the readout of the event, which is then further processed by the High Level Trigger (HLT). The GT consists of up to 13 custom boards (12 algorithm, 1 Final Or), each equipped with a Xilinx Ultrascale+ FPGA. The Global Trigger is planned to evaluate up to 1000 configurable cut based and neural net algorithms, the so called *Menu*, in a timing budget of $1\mu s$. To evaluate and test the functionality of the GT algorithms, a bitwise compatible emulator in the CMS software framework (CMSSW) has been developed. CMSSW provides a modular framework consisting of Event Data (ED) producers (add data to the event file), ED filters (read the event data and stop/continue processing the path) and ED analyzers (read event data, used to study event properties). CMSSW is used for data analysis, emulation of the firmware, as well as by the HLT during data taking for event reconstruction. CMSSW is written in C++ and configured in Python.



In addition to the emulator, a program has been developed which translates the Python *Menu* into VHDL. This translation can be directly used to build the GT firmware. As the GT hardware consists of up to 12 algorithm boards, the VHDL writer also takes care of distributing these algorithms over the boards. Here, physical limitations of the FPGAs are taken into account, like the availability of special logic blocks such as digital signal processors (DSPs) or Block RAMs, as well as limitation in routing resources.

P2GT candidate producer

The P2GT (Phase 2 Global Trigger) Candidate Producer (ED producer) takes in collections from the upstream objects and generates generic GT objects. It also translates not common parameters to common scales used in the internal GT format.



Scales

The scales for the GT can be configured in a Python configuration file, which is then read by the CMSSW emulator and exposed to the VHDL writer via pybinds11.

Conditions

The conditions are written in a generic way and can be applied to any particle-like GT object. The possible cuts include simple cuts such as pT, η or ϕ and correlational cuts such as ΔR or invariant mass. There are currently four generic condition types.

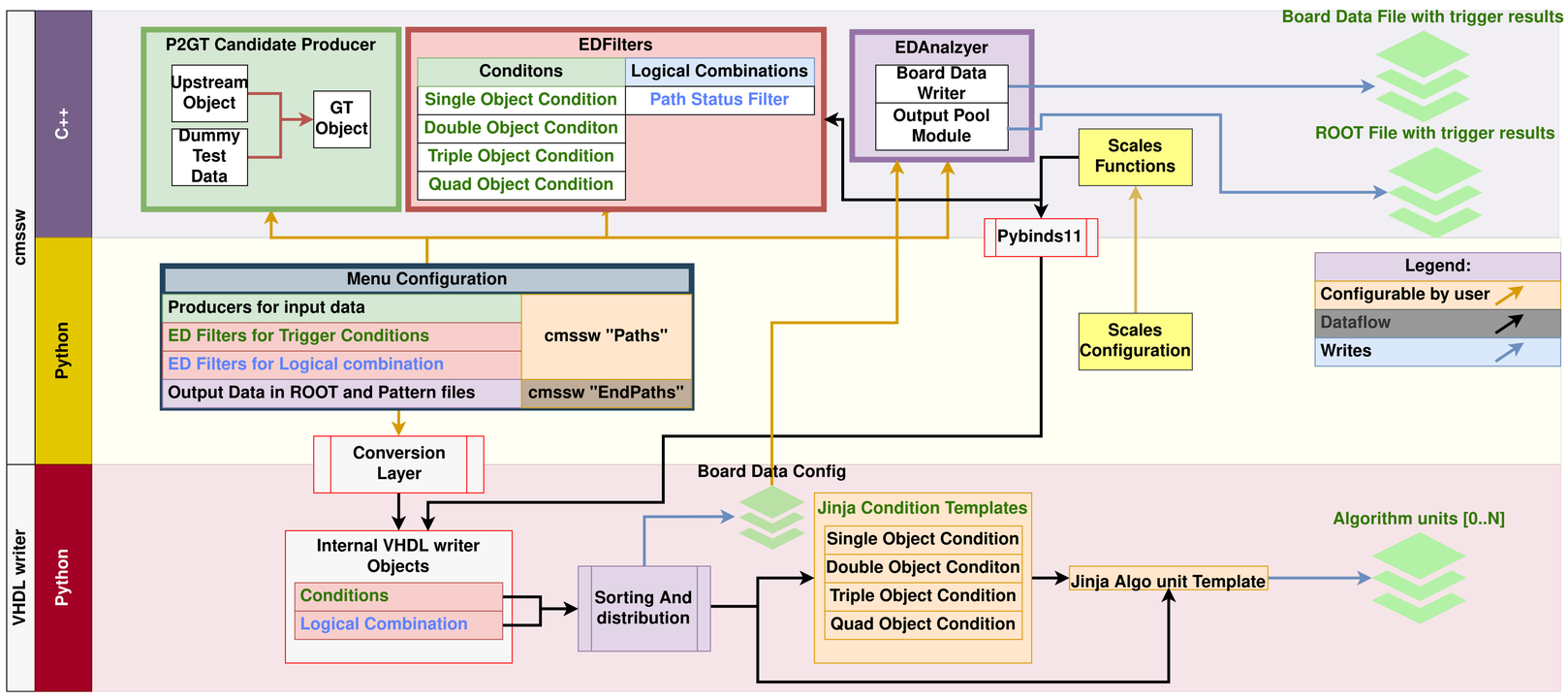
Logical combinations

As the CMSSW framework does not support the logical *or* operator in its path statements, the GT emulator makes use of the *Path status filter*, where arbitrary logical combinations of GT conditions can be applied.

Analyzers

The results of the GT emulator can be stored in root files. Additionally, the results can be output in a format used for firmware testing.

Structure of the GT emulator and VHDL writer



Internal VHDL writer objects

The VHDL writer reads the CMSSW menu configuration and translates the CMSSW modules into internal objects with additional information (concerning the size of the condition on the FPGA). It differentiates between four types of *conditions* and *logical combinations* as the latter are applied to the results of conditions.

Sorting and Distribution

To achieve an even distribution and to meet requirements set by the FPGA, the menu has to be distributed over the FPGAs and their super logic regions (SLRs). This is dependent on the properties of the algorithms, as, for example, correlational cuts require DSPs which are a limited resource on the FPGA. To facilitate routing, the VHDL writer can also try to minimize the number of input collections required on each board.

Templating

After sorting, the VHDL writer creates the VHDL translation of the menu, which is a set of algorithm unit files (one per SLR). This is done by predefined jinja templates which are then integrated in the GT firmware code.