Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Flexible trigger menu implementation on the Global Trigger for the CMS Level-1 trigger upgrade

To cite this article: Takashi MATSUSHITA and for the CMS Collaboration 2017 *J. Phys.: Conf. Ser.* **898** 032033

Flexible trigger menu implementation on the Global Trigger for the CMS Level-1 trigger upgrade

Takashi MATSUSHITA for the CMS Collaboration

Institute of High Energy Physics, Austrian Academy of Sciences,
Nikolsdorfer Gasse 18, 1050 Wien, Austria

E-mail: `takashi.matsushita@cern.ch`

Abstract. The CMS experiment at the Large Hadron Collider (LHC) has continued to explore physics at the high-energy frontier in 2016. The integrated luminosity delivered by the LHC in 2016 was 41 fb^{-1} with a peak luminosity of 1.5×10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and peak mean pile-up of about 50, all exceeding the initial estimations for 2016. The CMS experiment has upgraded its hardware-based Level-1 trigger system to maintain its performance for new physics searches and precision measurements at high luminosities. The Global Trigger is the final step of the CMS Level-1 trigger and implements a trigger menu, a set of selection requirements applied to the final list of objects from calorimeter and muon triggers, for reducing the 40 MHz collision rate to 100 kHz. The Global Trigger has been upgraded with state-of-the-art FPGA processors on Advanced Mezzanine Cards with optical links running at 10 GHz in a MicroTCA crate. The powerful processing resources of the upgraded system enable implementation of more algorithms at a time than previously possible, allowing CMS to be more flexible in how it handles the available trigger bandwidth. Algorithms for a trigger menu, including topological requirements on multi-objects, can be realised in the Global Trigger using the newly developed trigger menu specification grammar. Analysis-like trigger algorithms can be represented in an intuitive manner and the algorithms are translated to corresponding VHDL code blocks to build a firmware. The grammar can be extended in future as the needs arise. The experience of implementing trigger menus on the upgraded Global Trigger system will be presented.

1. Introduction

In 2016 the CMS experiment at the Large Hadron Collider (LHC) [1] has continued to explore physics at the high-energy frontier. The integrated luminosity delivered by the LHC in 2016 was 41 fb^{-1} with a peak luminosity of 1.5×10^{34} cm^{-2} s^{-1} and peak mean pile-up of about 50, all exceeding the initial estimations for 2016. The CMS Level-1 trigger system has been upgraded to improve its performance for selecting interesting physics events and to operate within the predefined data-acquisition rate in this challenging environment [2].

The Global Trigger is the final step of the CMS Level-1 trigger and implements a trigger menu, a set of selection requirements applied to the final list of objects from the calorimeter and muon trigger sub-systems, for the CMS physics programme [3]. The conditions for trigger object selection, with possible topological requirements on multi-object triggers, are combined by logical operators (`AND`,`OR`,`NOT`) to form the algorithms. The most basic algorithm consists of an E_T or p_T threshold applied to a single object. External signals, which are binary input signals coming from sub-detectors such as the Beam Pickup Timing for the eXperiment (BPTX), can

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 032033 doi:10.1088/1742-6596/898/3/032033

also be used in the algorithms. Figure 1 shows a schematic view of the CMS Level-1 trigger system.

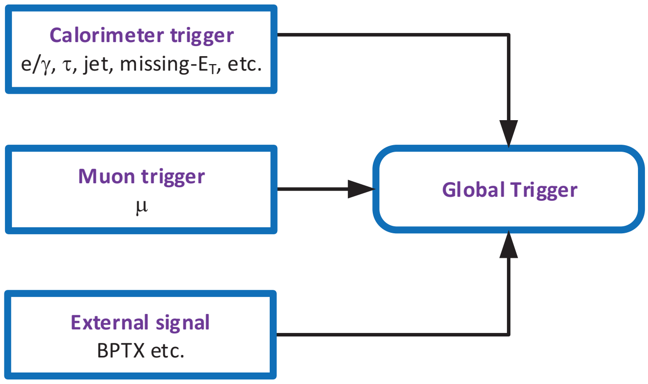

Figure 1. Schematic view of the CMS Level-1 trigger system. Calorimeter and muon trigger objects as well as external signals for each event are fed into the Global Trigger. Based on the Level-1 trigger menu implemented on the Global Trigger, decisions are made if an event is to be processed further or not.

The Global Trigger has been re-implemented on modern FPGAs on an Advanced Mezzanine Card in a MicroTCA crate (uGT). The MP7 (Master Processor board, Xilinx Virtex-7) developed under the leadership of Imperial College [4] has been used for the upgraded Global Trigger. The processing resources of the upgraded system with optical links running at 10 Gbit/s enable implementation of more algorithms at a time than previously possible, allowing CMS to be more flexible in how it handles the trigger bandwidth. With this powerful processing unit, more sophisticated quantities, such as the invariant mass of a pair of objects, can be computed. Different objects, e.g. muons with jets, can be matched with higher resolution and efficiency thanks to the upgrade of calorimeter and muon trigger systems.

Algorithms for a trigger menu, including topological requirements on multi-objects, can be realised on the uGT using the newly developed trigger menu specification grammar [5]. Analysis-like trigger algorithms can be represented in an intuitive manner, and the algorithms can be translated to corresponding VHDL code blocks to build a firmware. The grammar can be extended in future as needs arise. The experience of implementing trigger menus on the uGT system will be presented.

2. uGT Trigger Menu Library

In order to handle the increased complexity of the trigger menu for the upgraded system, the uGT Trigger Menu (UTM) software library has been developed. The library consists of several key components:

- uGT trigger menu specification grammar
- XML schema
- XML data binding for C++
- C++ classes for the Level-1 menu
- Python bindings.

Each of the components are briefly explained in the following sections.

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 032033 doi:10.1088/1742-6596/898/3/032033

2.1. Level-1 menu grammar

A set of rules to describe trigger algorithms for uGT, the Level-1 menu grammar, has been introduced. By using the grammar the data structure of the Level-1 trigger menu becomes flexible for a future evolution of the system. In addition, thanks to the abstract nature of the grammar, algorithm expressions are decoupled from the underlying hardware implementation. An application programme is used to translate algorithms expressed in the Level-1 menu grammar to the concrete VHDL code blocks. A trigger algorithm is comprised of a combination of “object”, “cut”, “function”, “external signal” and “logical operators”.

An object is a physics object reconstructed by the calorimeter and the muon trigger systems. At the beginning of the 2016 data taking period, μ (`MU`), e/γ (`EG`), jet (`JET`), τ (`TAU`), total transverse energy (`ETT`), total hadronic transverse energy (`HTT`), missing transverse energy (`ETM`) and missing hadronic transverse energy (`HTM`) were defined in the grammar. In the current firmware implementation of uGT, objects are available within $\pm$ 2 bunch crossings with respect to the colliding bunch. Objects in the grammar are abstract by themselves. The abstract object expression becomes a concrete one by specifying the relative bunch crossing ID with respect to the colliding bunch and the E_T/p_T threshold of the object. An example is `MU20` and `MU20+1`, which correspond to a muon with $p_T \geq 20$ GeV in the colliding bunch and in the subsequent bunch crossing, respectively.

Each object has additional attributes such as η (`ETA`), ϕ (`PHI`), charge (`CHG`), isolation (`ISO`) and quality (`QLTY`). For each attribute, a cut can be specified for filtering the objects. The objects and their associated cuts available in the grammar are summarised in table 1. The general form of algorithm expression with an object and cuts is object[$\mathtt{cut}_1$, $\mathtt{cut}_2$, ..., $\mathtt{cut}_n$]. An example is `EG20[EG-ETA_CEN,EG-ISO_1]` which corresponds to an e/γ with $E_T \geq 20$ GeV in the specified η range with specified isolation criteria.

The external signals available in the uGT are handled as a variant of an object except for threshold attributes and cuts.

Table 1. List of objects and their associated cuts defined in the grammar.

Object	Available cuts
MU	ETA, PHI, CHG, QLTY
EG, TAU	ETA, PHI, ISO
JET	ETA, PHI
ETM/HTM	PHI
ETT/HTT	not applicable

Topological selections as well as requirements on sophisticated quantities computed on uGT, such as the invariant mass of a pair of objects, can be described by using functions of the Level-1 menu grammar. The general form of the algorithm expression with a function is func{$\mathtt{object}_1$, $\mathtt{object}_2$, ..., $\mathtt{object}_m$}[$\mathtt{cut}_1$, $\mathtt{cut}_2$, ..., $\mathtt{cut}_n$]. Selection criteria on correlations of two objects can be described by using the “`dist`” function with metrics specified in cuts for the function. For the `dist` function, $\Delta\phi$ (`DPHI`), $\Delta\eta$ (`DETA`) and ΔR (`DR`) can be specified in the cuts for the function. An example is `dist{MU10,MU0}[DETA_MAX_1p8]` which means $\Delta\eta$ between two muons should be in the specified range. The invariant mass of a pair of objects is expressed with the “`mass`” function with the mass range specified in cuts (`MASS`). An example is `mass{JET30,JET30}[MASS_MIN_580]` which means that the invariant mass of two jets should be in the specified range. A multi-object trigger of the same object type also uses a function, “`comb`”, in which up to four objects can be specified. Charge correlations of muon objects can be specified in cuts (`CHGCOR`). An example is `comb{EG14,EG10,EG8}` which requires

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 032033 doi:10.1088/1742-6596/898/3/032033

a combination of three unique e/γ objects in an event. The functions and their associated cuts available in the grammar are summarised in table 2. The functions cannot be nested in the Level-1 menu grammar. However more complex functions can be introduced, e.g. to perform selection with invariant mass after removing overlapped objects by their distances.

Table 2. List of functions and their associated cuts defined in the grammar.

Functions	Available cuts
`comb`	`CHGCOR`
`dist`	`DETA`, `DPHI`, `DR`
`mass`	`MASS`

Figure 2 shows the input trigger objects of the uGT within ± 2 bunch crossings with respect to the colliding one and example algorithms expressed with the Level-1 menu grammar in fixed-width fonts. Example algorithms include simple object selections, topological selections with two objects, invariant mass selections with two objects and an example with logical operators as well as with external signals in the preceding and the following bunch crossings.

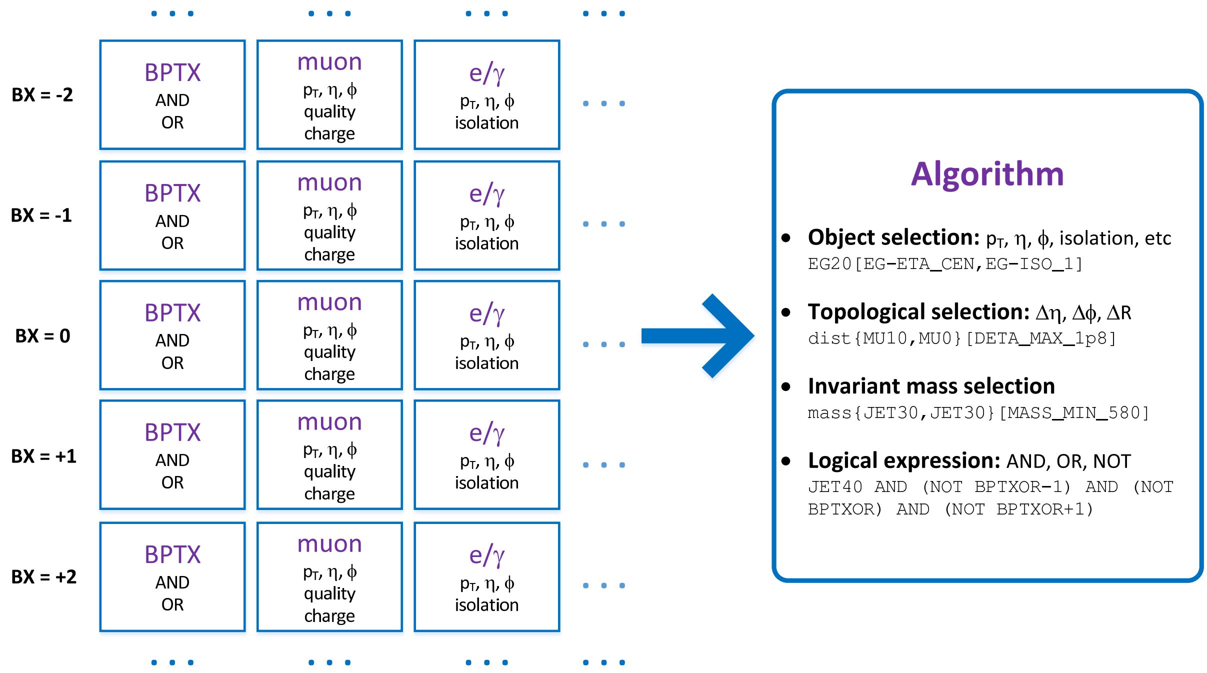

Figure 2. Input trigger objects of the uGT and example algorithms on uGT expressed with the Level-1 menu grammar in fixed-width fonts. The expressions, "`BPTXOR-1`", "`BPTXOR`" and "`BPTX+1`", mean the signal in previous, current and next bunch crossings, respectively.

2.2. XML schema

An XML schema for the Level-1 menu has been developed. The schema is based on the entity-relationship model of elements required for the menu specification so that the mapping of XML data to the relational database becomes straightforward. The top level element is "menu", which includes its name as a unique identifier. An algorithm element includes its name for unique identification in a menu and the algorithm expressed in the Level-1 menu grammar. The menu and algorithms are associated with links so that a single menu can contain multiple algorithms. An object element includes the type of the object, E_T/p_T threshold of the object and the relative bunch crossing offset with respect to the colliding bunch. A cut element includes its name for unique identification in a menu, type of the cut, valid range of the object's attribute or quantities computed by functions, and text string as data for the cut on object classification such as isolation and quality. The objects and cuts used in the algorithm expression are associated with links between them. Figure 3 shows the simplified view of the data structure for the Level-1 menu grammar.

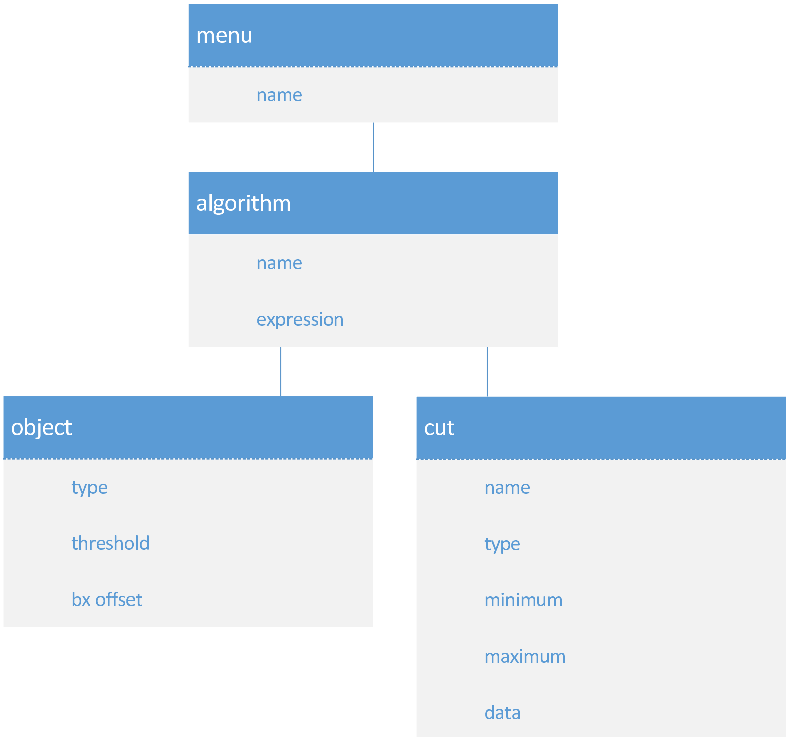

Figure 3. Simplified view of the data structure for the Level-1 trigger menu.

2.3. XML data binding for C++
The XML data binding for C++ is generated with CodeSynthesis XSD [6]. The C++/Parser mapping is used for in-memory representations of XML data with XML Schema validation. The C++/Tree mapping is used for serialising in-memory data back to XML data. An intermediate table-like data structure is defined for a possible interface with a relational database as well as a bridge between the C++/Parser mapping and the C++/Tree mapping.

2.4. C++ classes for the Level-1 menu
The C++ classes for representing the Level-1 menu have been defined so that the in-memory representations of XML data can be converted for persistence in the conditions database. By using the C++ classes, the uGT emulator software in the CMSSW framework can handle the Level-1 menu stored in an XML file or in the conditions database in the same way. With the help of class methods, all the algorithms in a menu as well as all the objects and cuts used for an algorithm can be accessed to instantiate selection logic as in the corresponding firmware. The

Level-1 menu XML files used for data taking are automatically converted to conditions database entries for offline use. A Level-1 menu in the conditions database is retrieved as the C++ classes representing the Level-1 menu thus the methods provided by the UTM library can be used in offline analysis. Manual conversion of XML files to conditions database entries can be performed allowing for fixed-point emulation of the Global Trigger in simulated data.

2.5. Python bindings
The Simplified Wrapper and Interface Generator (SWIG) [7] is used to generate a Python [8] interface. Python is used for developing utility programmes such as a graphical user interface of the XML file and the application to generate VHDL code blocks from the algorithm specification in the menu XML file.

3. Evolution of the Level-1 menu during the 2016 data taking period
The UTM library [9] was used in the online and offline software environments. In the online software environment, the library was used to configure the hardware system from the XML menu stored in the configuration database. In the offline software environment, both the XML file and condition database were used to configure the uGT emulator software for the development of the Level-1 menu and for the validation of the menu. Because of the different software environments in online and offline domains, commonly available libraries in both domains, Boost [10] and Xerces-C [11], were used to develop the UTM library. Figure 4 shows the overview of the UTM library and its use.

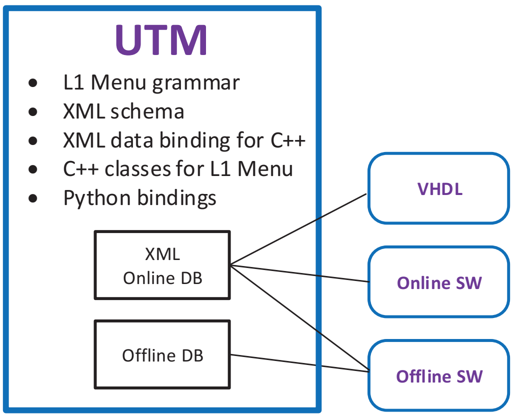

Figure 4. Overview of uGT Trigger Menu library and its use.

In order to cope with changing LHC conditions, 13 and 3 revisions of the Level-1 menu were used online during the 2016 data taking period for *pp* collisions and *p*Pb collisions, respectively. In 2016, up to 267 algorithms were defined in the Level-1 menu which were implemented in 2 uGT boards. The current system allows up to 512 algorithms which can be implemented in maximum of 6 uGT boards. New calorimeter trigger objects were introduced in the middle of the 2016 data taking period for minimum bias event selections. The UTM library was updated for handling them in the Level-1 menu. Care had to be taken whenever the new types of trigger objects or new functions were introduced in the Level-1 menu so that consistent versions of the UTM library were available both in the online and offline software environments.

Thanks to the introduction of the Level-1 menu grammar, it was possible to introduce the new types of trigger objects in the Level-1 trigger menu by updating the Level-1 menu grammar and its corresponding parsers. There was no need to change the XML schema and the C++

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 032033 doi:10.1088/1742-6596/898/3/032033

classes for persistence in the conditions database as it was intended in the original design of the UTM library.

The Python bindings of the UTM library can be used to prepare tools useful for the study of the Level-1 trigger in offline software environments. Examples of such tools are available on gitlab [12] which makes it possible to create the Level-1 menu XML file from the algorithms implemented in C++. This will help streamline Level-1 trigger menu studies and the XML menu preparation in the future.

4. Summary

The Level-1 Trigger system of the CMS experiment has been upgraded for the 2016 data taking period. The Global Trigger was re-implemented on modern FPGAs on an Advanced Mezzanine Card in a MicroTCA crate. In order to handle the increased complexity of the trigger menu for the upgraded system as well as for hiding the underlying hardware implementation to make the data structure for a trigger menu flexible for a future evolution of the trigger system, the uGT Trigger Menu (UTM) software library was introduced in online and offline software environments. In 2016 the Level-1 trigger menu was updated 16 times including the introduction of new calorimeter trigger objects to cope with changing LHC conditions, proving the flexibility of the Level-1 menu implementation enabled by the UTM library. The Python bindings of the UTM library can be used for rapidly implementing various tools needed for the Level-1 trigger menu development.

Acknowledgments

We acknowledge the enduring support for the construction and operation of the LHC and the CMS detector provided by the Bundesministerium für Wissenschaft, Forschung und Wirtschaft (Austria).

References

[1] CMS Collaboration 2008 *JINST* **3** S08004
[2] CMS Collaboration 2012 *CMS technical design report for the level-1 trigger upgrade* CMS-TDR-012, CERN-LHCC-2013-011
[3] M Jeitler *et al* 2007 *JINST* **2** P01006
[4] The Imperial Master Processor, Virtex-7 (MP7) – `http://www.hep.ph.ic.ac.uk/mp7/`
[5] T Matsushita *et al* 2015 *J Phys Conf Ser.* **664** 082031
[6] CodeSynthesis XSD – `http://www.codesynthesis.com/products/xsd/`
[7] David M. Beazley 1996 *Proceedings of the 4th conference on USENIX Tcl/Tk Workshop* p15
[8] Python scripting language – `https://www.python.org`
[9] uGT Trigger Menu software library – `https://gitlab.cern.ch/cms-l1t-utm/utm`
[10] Boost C++ library – `http://www.boost.org/`
[11] Xerces XML parser – `http://xerces.apache.org/xerces-c/`
[12] the uGT Trigger Menu tools – `https://gitlab.cern.ch/cms-l1t-utm/scripts`