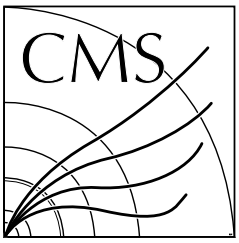


Global Trigger Logic - Description for emulator

Herbert Bergauer, Babak Rahbaran, Johannes Wittmann
Email: herbert.bergauer@oeaw.ac.at

Institute of High Energy Physics (HEPHY)

http://www.hephy.at
http://globaltrigger.hephy.at

September 17, 2020

Revision History

Doc Rev	Description of Change	Revision Date
5.0	Major changes in description of GTL firmware, because of new correlation cuts calculation structure (outside of conditions).	2020/09/17
4.5	Updated text in section "Correlation conditions" 2.4.9.	2020/09/16
4.4	Fixed typo (unconstrained pt)	2020/09/09
4.3	Inserted text for new muon structure in sections 2.2.2 and 2.4.8.1.	2020/08/04
4.2	Additional text in section 2.4.9.1.3.	2020/05/25
4.1	Updated glossary.	2020/04/21
4.0	Removed listings (not usefull for emulator designers).	2020/04/17
3.9	Inserted text in section 2.4.3.2.2 for Calorimeter Overlap Remover conditions and 2.4.9.1.3 for Calo Calo Overlap Remover Correlation conditions.	2020/04/16
3.8	Updated text in sections 2.4.3, 2.4.8 and 2.4.9 for changes which have been done for GTL VHDL version 1.8.0 (module names without version number, "five eta cuts").	2019/08/13
3.7	Inserted "Asymmetry" and "Centrality" of "Energy sums" (GTL VHDL version 1.6.0). Therefore updated sections 2.1, 2.2.1, 2.4.4 added section "Centrality condition" 2.4.7	2018/08/13
3.6	Updated text in section "Global Trigger Logic" (2) according to firmware version v1.5.0 of gtl_module.vhd	2018/02/21
3.5	Inserted text for "Minimum bias trigger conditions" (2.4.5) and "Towercount condition" (2.4.6). Updated glossary.	2016/11/28
3.4	Updated text in section "Muon Muon Correlation condition module" (**??**).	2016/01/15
3.3	Removed "Double objects requirements condition with spatial correlation", because not used anymore in the future, replaced by Correlation conditions (see sections 2.4.3.2 and 2.4.8.2.	2016/01/08
3.2	Minor changes in text and updated Figure 2.	2016/01/08
3.1	Changed colour in Figure 4 and updated text for correlation conditions (see section 2.4.9.	2016/01/07
3.0	Updated Figures 2 and text **??**.	2015/12/21
2.9	Inserted drawing of VHDL structure of cuts for correlation conditions (see Figure 5).	2015/11/18

Doc Rev	Description of Change	Revision Date
2.8	Updated muon η ranges (Table 10) and inserted correlation conditions (see section 2.4.9). Created scheme for conversion of calorimeter η and φ to muon scale for calo-muon-correlation conditions (see Figure 3).	2015/11/17
2.7	Added text in sections (2.4.3.2.4) and (2.4.8.2.2).	2015/10/08
2.6	Updated Tables 9 and 16. Remaned section "Calorimeter conditions module - version 2" to "Calorimeter conditions module - version 3" (see 2.4.3.2), section "Muon conditions module" to "Muon conditions module - version 2" and section "Muon comparators module" to "Muon comparators module - version 2" (see 2.4.8.2)	2015/10/02
2.5	Updated text and tables of η ranges for Calorimeter objects (see 2.4.3.1).	2015/09/22
2.4	Corrected calculation of muon η step width (see 2.4.8.1).	2015/09/10
2.3	Edited text in Tables 9, 15 and 16.	2015/08/28
2.2	Updated definition of η ranges for Calorimeter objects and Muon objects (2.4.3.1 and 2.4.8.1).	2015/08/20
2.1	Added section Calo Muon Correlation condition (**??**)	2015/08/19
2.0	Added section "Correlation conditions" (2.4.9).	2015/06/19
1.9	Added tables for calorimeter isolation-bits and for muon quality- and isolation-bits definition (8, 12 and 14). Edited section glossary (3) and acronyms.	2015/05/07
1.8	Added text for "Energy sum conditions" (2.4.4) and updated chapters for "Calorimeter conditions" for version 2. Inserted isolation bits for electron/γ and tau objects (2.4.3.1).	2015/05/06
1.7	Minor changes in sections "Muon data" (2.4.8.1)	2014/11/06
1.6	Minor changes in sections "Calorimeter conditions definition" (2.4.3.2) and "Muon conditions definition" (2.4.8.2)	2014/07/01
1.5	Minor changes	2014/06/12
1.4	Fixed bug in Figure 4	2014/04/30
1.3	Added section "Muon conditions" (2.4.8)	2014/04/22
1.2	Changed Figure 4 and minor changes in text for anti-clockwise behaviour in φ	2014/04/04
1.1	Changed text in section 2.4.3.2.	2014/02/11
1.0	Document created. Description of Calorimeter conditions.	2013/10/15

Contents

List of Figures 5

List of Tables 6

1 Package: lhc_data_pkg 7

2 Global Trigger Logic 8

2.1 μGTL Interface . . . 8

2.2 Definition of optical interfaces . . . 9

2.2.1 Calo-Layer2 optical interfaces . . . 9

2.2.2 Global Muon Trigger optical interfaces . . . 10

2.3 Implementation in firmware . . . 11

2.3.1 Top-of-hierarchy module . . . 11

2.3.2 Package module . . . 12

2.4 μGTL structure . . . 13

2.4.1 Data ±2bx . . . 13

2.4.2 Calculation of correlation cuts . . . 14

2.4.2.1 Calculation of differences in η and φ . . . 14

2.4.2.2 Calculation of $\cosh(\Delta\eta)$ and $\cos(\Delta\varphi)$ . . . 15

2.4.2.3 ΔR calculation . . . 16

2.4.2.4 Invariant mass calculation . . . 16

2.4.2.5 Transverse mass calculation . . . 16

2.4.2.6 Two-body pt calculation . . . 16

2.4.2.6.1 Invariant mass divided by ΔR calculation . . . 17

2.4.2.7 Invariant mass calculation for three objects . . . 17

2.4.2.8 Muon charge correlation module . . . 18

2.4.3 Calorimeter conditions . . . 20

2.4.3.1 Calorimeter data . . . 20

2.4.3.2 Calorimeter conditions definition . . . 23

2.4.3.2.1 Calorimeter conditions module . . . 24

2.4.3.2.2 Calorimeter Overlap Remover conditions module . . . 24

2.4.3.2.3 Calorimeter conditions module - template for VHDL-Producer . . . 24
2.4.3.2.4 Calorimeter comparators module . . . 24
2.4.4 Energy sum quantities conditions . . . 27
2.4.4.1 Energy sum quantities data . . . 27
2.4.4.2 Energy sum quantities conditions module (including Asymmetry conditions) . . . 29
2.4.4.3 Energy sum quantities conditions module - template for VHDL-Producer . . . 29
2.4.5 Minimum bias trigger conditions . . . 29
2.4.6 Towercount condition . . . 29
2.4.7 Centrality condition . . . 30
2.4.8 Muon conditions . . . 30
2.4.8.1 Muon data . . . 30
2.4.8.2 Muon conditions definition . . . 33
2.4.8.2.1 Muon conditions module . . . 34
2.4.8.2.2 Muon comparators module . . . 34
2.4.9 Correlation conditions . . . 36
2.4.9.1 Correlation condition types . . . 36
2.4.9.1.1 Overview of possible correlation cuts in conditions . 38
2.4.9.1.2 Correlation condition types . . . 40
2.4.9.1.3 Calo Calo Correlation condition module with Calo Overlap Remover . . . 40
2.4.9.1.4 Calo Calo Correlation condition module for Invariant Mass with Three Objects . . . 40
2.4.9.1.5 Muon Muon Correlation condition module for Invariant Mass with Three Objects . . . 41
2.4.10 External Conditions . . . 41
2.4.11 Algorithms logic . . . 41

3 Glossary **42**

List of Figures

1 μGTL firmware 8
2 Scheme of μGTL pipeline structure 13
3 Conversion of calorimeter η and φ to muon scales 15
4 Setting the limits for "window"-comparators for φ 26
5 VHDL structure of cuts for correlation conditions 37

List of Tables

2 Muon charge correlation - Double Muon 18
3 Muon charge correlation - Triple Muon 18
4 Muon charge correlation - Quad Muon 19
5 η scale of electron/γ and tau 21
6 η scale of jet 22
7 φ scale of calorimeter objects 22
8 Definition of e/γ and tau isolation bits 22
9 LUT contents for isolation comparison of electron/γ and tau objects 25
10 η scale of muon objects 31
11 φ scale of muon objects 31
12 Definition of muon quality bits 32
13 Definition of muon isolation bits 32
14 Definition of muon impact parameter bits 32
15 LUT contents for quality comparison of muon objects 35
16 LUT contents for isolation comparison of muon objects 35

1 Package: lhc_data_pkg

The VHDL record `lhc_data_t` (shown in Listing 1) is used as a container for all object streams processed by the system. It is declared in the VHDL package `lhc_data_pkg`. For debugging and simulation purposes a second package (`lhc_data_debug_util_pkg`) is created which contains functions to convert the `lhc_data_t` to a hexadecimal string representation and vice versa. The testbench of the design uses these functions to load the contents of the SIM memory from a file.

Listing 1: lhc_data_t record specification

```
type lhc_data_t is record
        muon : muon_array_t;
        eg : eg_array_t;
        tau : tau_array_t;
        jet : jet_array_t;
        ett : std_logic_vector(ETT_DATA_WIDTH-1 downto 0);
        ht : std_logic_vector(HT_DATA_WIDTH-1 downto 0);
        etm : std_logic_vector(ETM_DATA_WIDTH-1 downto 0);
        htm : std_logic_vector(HTM_DATA_WIDTH-1 downto 0);
        etmhf : std_logic_vector(ETMHF_DATA_WIDTH-1 downto 0);
        htmhf : std_logic_vector(HTMHF_DATA_WIDTH-1 downto 0);
        link_11_fr_0_data : std_logic_vector(LINK_11_FR_0_WIDTH-1 downto
            0);
        link_11_fr_1_data : std_logic_vector(LINK_11_FR_1_WIDTH-1 downto
            0);
        link_11_fr_2_data : std_logic_vector(LINK_11_FR_2_WIDTH-1 downto
            0);
        link_11_fr_3_data : std_logic_vector(LINK_11_FR_3_WIDTH-1 downto
            0);
        link_11_fr_4_data : std_logic_vector(LINK_11_FR_4_WIDTH-1 downto
            0);
        link_11_fr_5_data : std_logic_vector(LINK_11_FR_5_WIDTH-1 downto
            0);
        external_conditions : std_logic_vector(
            EXTERNAL_CONDITIONS_DATA_WIDTH-1 downto 0);
end record;
```

2 Global Trigger Logic

Remark:
This description is for version 1.11.0 of Global Trigger Logic.

The Global Trigger Logic (μGTL) firmware contains conditions and Algorithms for trigger decision.

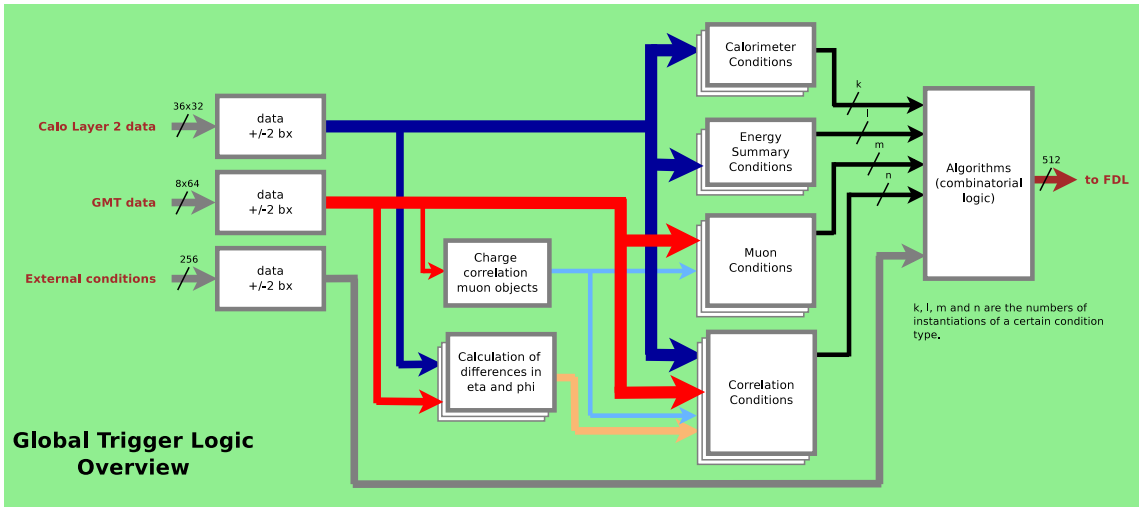


Figure 1: μGTL firmware

2.1 μGTL Interface

Inputs:

- Calo-Layer2 data
  - Electron/γ objects
  - Jet objects
  - Tau objects
  - Energy summary information: Total Et (ET), total Et from ECAL only (ETTEM), total calibrated Et in jets (HT), missing Et (ET_{miss}), missing Et including HF (ET^{HF}_{miss}), missing Ht objects (HT_{miss}), missing Ht including HF (HT^{HF}_{miss}) and "Asymmetry" information (ASYMET, ASYMHT, ASYMETHF, ASYMHTHF)
  - Minimum bias HF bits (included in energy summary information data structure)
  - Towercount bits (number of firing HCAL towers, included in energy summary information data structure)
  - "Centrality" bits
- Global Muon Trigger data

- External conditions
- LHC-clock

Outputs:

- Algorithms

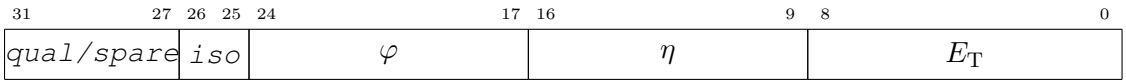
2.2 Definition of optical interfaces

Remark: all definitions for scales in the following chapters are from a CMS Detector Note: "Scales for inputs to μGT" (see actual version in https://raw.githubusercontent.com/cms-l1-globaltrigger/mp7_ugt_legacy/master/doc/scales_inputs_2_ugt/pdf/scales_inputs_2_ugt.pdf).

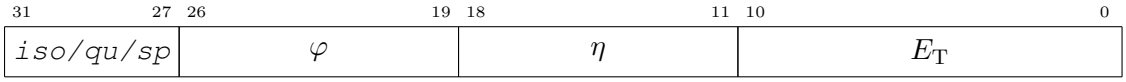
2.2.1 Calo-Layer2 optical interfaces

The configuration of optical connections from Calo-Layer2 to μGT is shown in Table **??**.

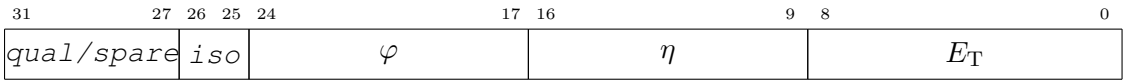
The data structure of an electron/γ object (bits 27..31 are not defined yet, reserved for quality, ...):



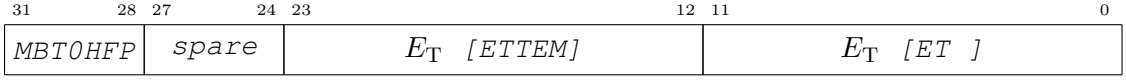
The data structure of a jet object (bits 27..31 are not defined yet, reserved for quality, ...):



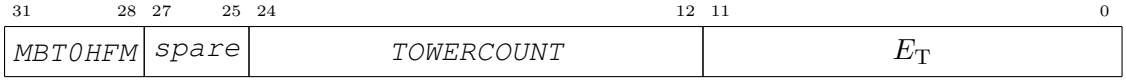
The data structure of a tau object (bits 27..31 are not defined yet, reserved for quality, ...):



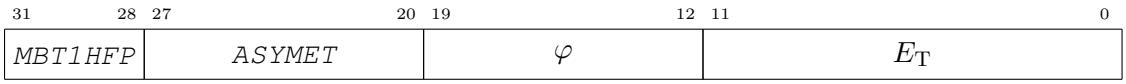
The data structure of "total Et" (ET) quantity [including "total Et from ECAL only" (ETTEM) and "minimum bias HF+ threshold 0" bits]:



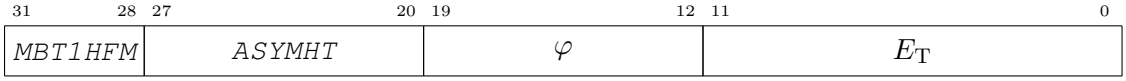
The data structure of "total calibrated Et in jets" (HT) quantity [including "towercount" and "minimum bias HF- threshold 0" bits]:



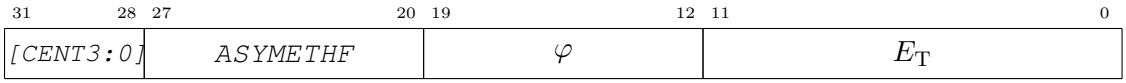
The data structure of "missing Et" (ET_{miss}) quantity [including "Asymmetry" ASYMET and "minimum bias HF+ threshold 1" bits]:



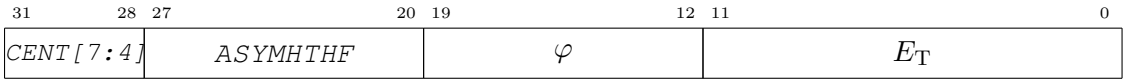
The data structure of "missing Ht" (HT_{miss}) quantity [including "Asymmetry" ASYMHT and "minimum bias HF- threshold 1" bits]:



The data structure of "missing Et including HF" (ET^{HF}_{miss}) quantity [including "Asymmetry" ASYMETHF and "Centrality" bits (3:0)]:

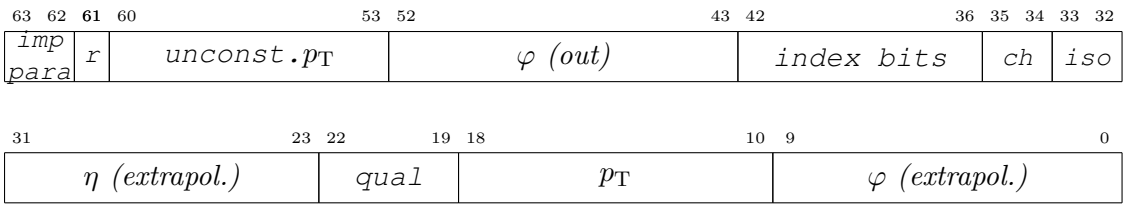


The data structure of "missing Ht including HF" (HT^{HF}_{miss}) quantity [including "Asymmetry" ASYMHTHF and "Centrality" bits (7:4)]:



2.2.2 Global Muon Trigger optical interfaces

The data structure of a muon object (64 bits - bit 34 = charge sign, bit 35 = charge valid, bit 61 is a spare bit, bit 63..62 = impact parameter):



2.3 Implementation in firmware

The firmware of μGTL consists of two main parts:

- first part which is depending on L1Menus and is created by VHDL Producer
  - top-of-hierarchy file (`gtl_module.vhd`), which contains:
    - pipeline for ±2bx data
    - instantiations of calculators for differences in η and φ, for ΔR, for all mass types, for two-body pt
    - instantiations of charge correlation logic of muons
    - instantiations of conditions
    - Algorithms logic for 512 Algorithms
  - as well as a package file (`gtl_pkg.vhd`) for declarations.
- second part
  - which is a set of VHDL-files for all the modules instantiated in top-of-hierarchy and the modules in the hierarchy. These files, called the "fixed part", are not influenced by VHDL Producer.

Actually 6 AMC board are used to contain 512 Algorithms. Therefore the 512 Algorithms are **partitioned by VHDL Producer**. The VHDL Producer for every Trigger Menu creates VHDL snippets files (`algo_index.vhd`, `gtl_module_instances.vhd`, `gtl_module_signals.vhd`, `ugt_constants.vhd`), these snippets are inserted into templates for `gtl_module.vhd` and `gtl_pkg.vhd` during simulation and synthesis.

The latency of μGTL is fixed to 5 bunch-crossings, 2 bunch-crossings for the pipeline of ±2bx data (for data with +2bx and +1bx), 2 bunch-crossings for conditions (fixed), also for the conditions requested in the future, 1 bunch-crossing for the logic of Algorithms (See Figure 2).

2.3.1 Top-of-hierarchy module

The structure of top-of-hierarchy module in this firmware version is different to the former. Calculation of correlation cuts is done outside of conditions. So, calculation of correlation cuts of a certain objects type bunch-crossing combination is instantiated only once in a module. This gives the advantage of using less resources for calculations.

The top-of-hierarchy module (`gtl_module.vhd`) contains

- pipeline for ±2bx data
- instantiations of charge correlation logic of muons (generated by VHDL Producer)
- instantiations of calculators for differences in η and φ, for ΔR, for all mass types, for two-body pt (generated by VHDL Producer)

- instantiations of conditions (generated by VHDL Producer)
- boolean logic for Algorithms (generated by VHDL Producer)

Listing 2 contains the entity-declaration of the top-of-hierarchy file (`gtl_module.vhd`).

Listing 2: Entity declaration of `gtl_module.vhd`

```
entity gtl_module is
    port(
        lhc_clk : in std_logic;
        eg_data : in calo_objects_array(0 to NR_EG_OBJECTS-1);
        jet_data : in calo_objects_array(0 to NR_JET_OBJECTS-1);
        tau_data : in calo_objects_array(0 to NR_TAU_OBJECTS-1);
        ett_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        ht_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        etm_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        htm_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
--
    ********************************************************************************

-- HB 2016-04-18: updates for "min bias trigger" objects (quantities) for Low-
    pileup-run May 2016
        mbt1hfp_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        mbt1hfm_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        mbt0hfp_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        mbt0hfm_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
-- HB 2016-06-07: inserted new esums quantities (ETTEM and ETMHF).
        ettem_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        etmhf_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
-- HB 2016-09-16: inserted HTMHF and TOWERCNT
        htmhf_data : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        towercount_data : in std_logic_vector(MAX_TOWERCOUNT_BITS-1 downto 0);
--
    ********************************************************************************

        muon_data : in muon_objects_array(0 to NR_MUON_OBJECTS-1);
        external_conditions : in std_logic_vector(NR_EXTERNAL_CONDITIONS-1 downto
            0);
        algo_o : out std_logic_vector(NR_ALGOS-1 downto 0));
end gtl_module;
```

2.3.2 Package module

All the declarations for arrays ('type'), parameters ('constant') and look-up-tables ('constant') used in modules are available in `gtl_pkg.vhd` package-file.

2.4 μGTL structure

2.4.1 Data ±2bx

The μGTL input data flow through a register pipeline of four stages. With those data it is possible to have conditions with objects from different bunch-crossings (within ±2 bunch-crossings), e.g. for Correlation conditions.
See Figure 2 for a scheme of μGTL pipeline structure. The data "data_p_1bx" and "data_p_2bx" occur 1 respectively two bunch-crossings after data for a certain bunch-crossing, therefore we got two bunch-crossings of latency from those data. The data "data_m_1bx" and "data_m_2bx" have no influence on latency, because coming before data for a certain bunch-crossing.

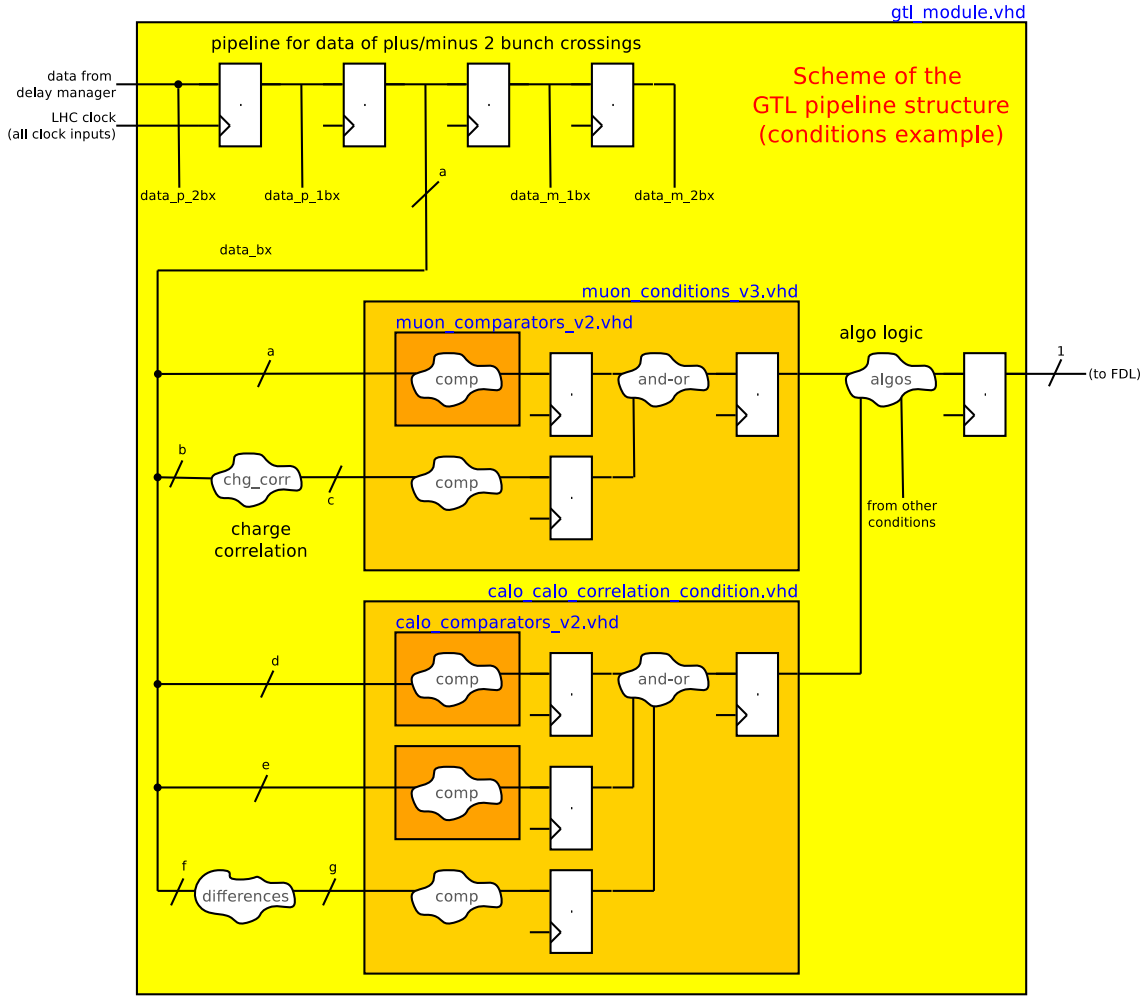


Figure 2: Scheme of μGTL pipeline structure

2.4.2 Calculation of correlation cuts

Calculation of correlation cuts is done outside of conditions in this version of Global Trigger firmware. Therefore a new set of VHDL modules is used.

2.4.2.1 Calculation of differences in η and φ

Some conditions types namely correlation conditions use differences in η and φ to make the decision. Therefore these differences are calculated out of these conditions, because the differences can be used several times in different condition types. The differences in η and φ are calculated in bins. These differences in bins are converted to numbers (by LUTs), which represents values of differences (multiples of units in η and φ). Differences in φ are provided by module `sub_phi_integer_obj_vs_obj.vhd`, which instantiates the module `sub_unsigned_phi.vhd` as many times as the numbers of both objects determine.
In the module `sub_unsigned_phi.vhd` a calculation of a difference of two objects is done, both objects must have the same resolution, namely the higher one. The result is the absolute value of the difference. There are two differences in φ, one "clockwise" and one "anti-clockwise". For the final result the smaller difference is taken.
Differences in η are provided by module `sub_eta_integer_obj_vs_obj.vhd`, which instantiates the module `sub_signed_eta.vhd` as many times as the numbers of both objects determine.
In the module `sub_signed_eta.vhd` a calculation of a difference of two objects is done with a signed subtraction, because of the Two's Complement notation of η values. The result is the absolute value of the difference.

Because of the different scales of calorimeter and muon objects in η and φ, there are LUTs for conversion the calorimeter bins to muon bins (in `gtl_pkg.vhd`: e.g. EG_ETA_CONV_2_MUON_ETA_LUTand EG_PHI_CONV_2_MUON_PHI_LUT).

Remark:
The center value of bins are used as reference value for conversion. The content of EG_ETA_CONV_2_MUON_ETA_LUTis calculated with formular:
"converted-calo-eta[bin] = calo-eta[bin] × 4 + 2",
of EG_PHI_CONV_2_MUON_PHI_LUTwith formular:
"converted-calo-phi[bin] = calo-phi[bin] × 4 + 2".
The conversion calculations are preliminary, others may be proposed.

Definitions of scales (see Tables 6, 7, 10 and 11):

- Calorimeter objects:
  - η bin width = $\frac{0.087}{2}$ (bin 0 from 0.0 to $\frac{0.087}{2}$)
  - ϕ bin width = $\frac{2\pi}{144}$ (bin 0 from 0.0 to $\frac{2\pi}{144}$)
- Muon objects:
  - η bin width = $\frac{0.087}{8}$ (bin 0 from $0.5\times\frac{-0.087}{8}$ to $0.5\times\frac{+0.087}{8}$)

- ϕ bin width = $\frac{2\pi}{576}$ (bin 0 from 0.0 to $\frac{2\pi}{576}$)

- ET_{miss}, ET^{HF}_{miss} or HT_{miss}:
  - ϕ bin width = $\frac{2\pi}{144}$ (bin 0 from 0.0 to $\frac{2\pi}{144}$)

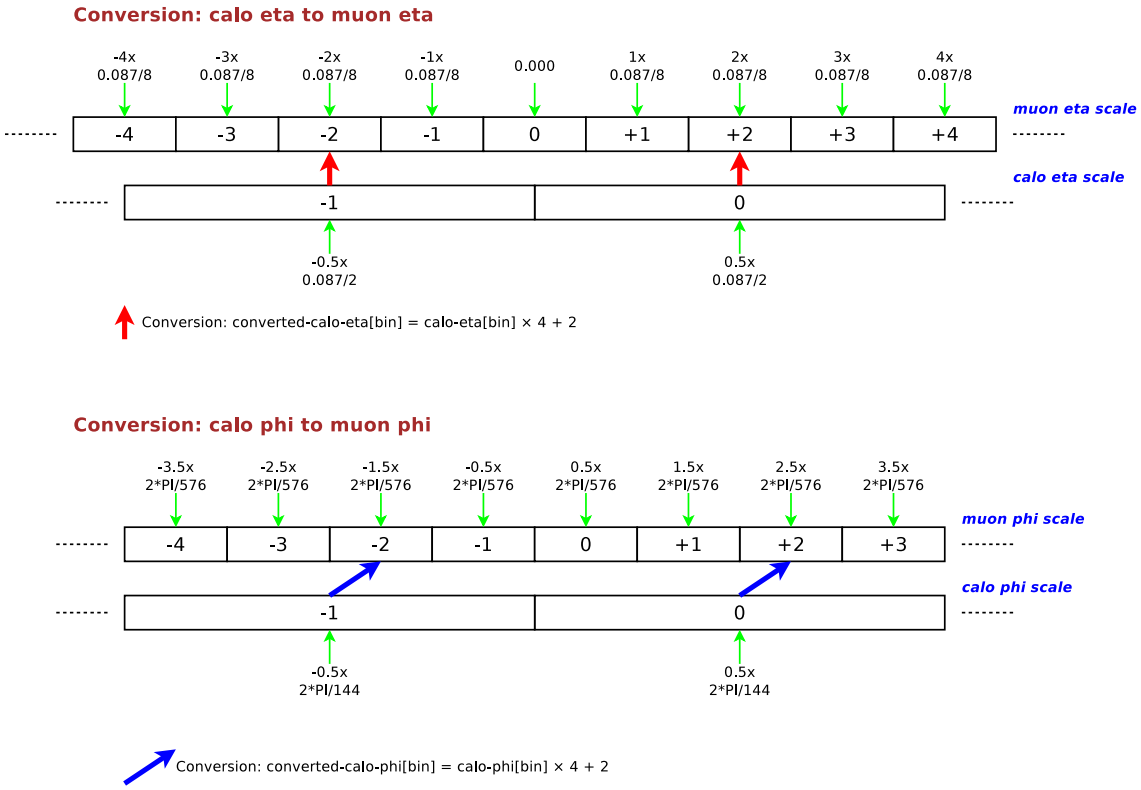


Figure 3: Conversion of calorimeter η and φ to muon scales

2.4.2.2 Calculation of $\cosh(\Delta\eta)$ and $\cos(\Delta\varphi)$

Some conditions types namely conditions with mass cuts use $\cosh(\Delta\eta)$ and $\cos(\Delta\varphi)$ to calculate mass.
The contents of the LUTs for $\cosh(\Delta\eta)$ (obj_type_1_obj_type_2_COSH_DETA_LUT, ...) and $\cos(\Delta\varphi)$ (obj_type_1_obj_type_2_COS_DPHI_LUT, ...) for invariant mass (formular see 2.4.2.4) are created by calculating hyperbolic cosine and cosine. Calculation for calo and esums objects is done by rounding-up at the 3rd position after decimal point, and multiplying by 1000 to get integer values.[1].
Calculation for conditions with muon objects is done by rounding-up at the 4th position after decimal point, and multiplying by 10000 to get integer values.[2]

[1]Definition of "`constant CALO_INV_MASS_COSH_COS_PRECISION...`" in file `gtl_pkg.vhd`. Value 1000 from $10^{\text{CALO_INV_MASS_COSH_COS_PRECISION}}$

[2]Definition of "`constant MUON_INV_MASS_COSH_COS_PRECISION`" and other constants (e.g. "`constant CALO_MUON_INV_MASS_COSH_COS_PRECISION`" in file `gtl_pkg.vhd`. Value 10000 from $10^{\text{MUON_INV_MASS_COSH_COS_PRECISION}}$.

2.4.2.3 ΔR calculation

The calculation of ΔR of two objects is done with formula in module `delta_r_calc.vhd`:

$$\Delta R = \sqrt{(\eta_1 - \eta_2)^2 + (\varphi_1 - \varphi_2)^2}.$$

In the TME there are two thresholds for ΔR: "greater/equal lower limit" and "less/equal upper limit", given in floating point notation with one position after decimal point. The comparison in VHDL is done with ΔR^2 (no square root in VHDL), thresholds for ΔR^2 are provided by VHDL-Producer.

2.4.2.4 Invariant mass calculation

The calculation of *invariant mass of two objects* is done with formula in module `mass_invariant_calc.vhd`:

$$\text{M}=\sqrt{2pt_1pt_2(\cosh(\eta_1 - \eta_2) - \cos(\varphi_1 - \varphi_2))}.$$

In the TME there are two thresholds for M: "greater/equal lower limit" and "less/equal upper limit", given in GeV (floating point notation) with one position after decimal point in even numbers.[3] The comparison in VHDL is done with $\frac{M^2}{2}$ (no square root in VHDL), thresholds for $\frac{M^2}{2}$ are provided by VHDL-Producer.

2.4.2.5 Transverse mass calculation

The calculation of *transverse mass of two objects* is done with formula in module `mass_transverse_calc.vhd`:

$$\text{M}=\sqrt{2pt_1pt_2(1 - \cos(\varphi_1 - \varphi_2))}.$$

In the TME there are two thresholds for M: "greater/equal lower limit" and "less/equal upper limit", given in GeV (floating point notation) with one position after decimal point in even numbers.
The comparison in VHDL is done with $\frac{M^2}{2}$ (no square root in VHDL), thresholds for $\frac{M^2}{2}$ are provided by VHDL-Producer.

2.4.2.6 Two-body pt calculation

The calculation of *two-body pt* is done with formula in module `tbpt_calc.vhd`:

[3] even numbers to get a precision of one position after decimal point after dividion by 2, because VHDL-Producer calculates thresholds for $\frac{M^2}{2}$, which includes a division by 2.

$$\text{pt}=\sqrt{pt_1^2 + pt_2^2 + 2pt_1pt_2(\cos(\varphi_1)\cos(\varphi_2) + \sin(\varphi_1)\sin(\varphi_2))}$$

In the TME there is one threshold for pt, given in GeV (floating point notation) with one position after decimal point. The comparison in VHDL is done with pt^2 (no square root in VHDL), threshold for pt^2 is provided by VHDL-Producer.
The contents of the LUTs for $\cos(\varphi)$ (xxxx_COS_PHI_LUT) and $\sin(\varphi)$ (xxxx_SIN_PHI_-LUT) for two-body pt (formular see 2.4.2.6) are created by calculating cosine and sine, rounding-up at the 3rd position after decimal point and multiplying by 1000 to get integer values.

2.4.2.6.1 Invariant mass divided by ΔR calculation

The formulas for *invariant mass divided by ΔR of two objects* are:

$$\text{M}=\sqrt{2pt_1pt_2(\cosh(\eta_1 - \eta_2) - \cos(\varphi_1 - \varphi_2))}.$$

$$\Delta R = \sqrt{(\eta_1 - \eta_2)^2 + (\varphi_1 - \varphi_2)^2}.$$

In the TME there is one threshold for M/ΔR, given in GeV (floating point notation) with one position after decimal point. The calculation of *invariant mass divided by ΔR of two objects* is done in an own module outside of the condition (`mass_div_dr_calculator.vhd`) with $\frac{M^2}{2}\times(1/\Delta R^2)$ (no square root in VHDL). This module is instantiated once for every object type bunch-crossing combination.
The values of $1/\Delta R^2$ in VHDL are given by LUTs stored in BRAMs (as ROMs). The addresses of the BRAMs are given by $\Delta\eta$ and $\Delta\varphi$.
The calculation of *invariant mass divided by ΔR of two objects* is done in modules `mass_-invariant_calc.vhd` and `mass_invariant_div_dr_calc.vhd` (for LUTs and comparison).

2.4.2.7 Invariant mass calculation for three objects

The calculation of *invariant mass calculation for three objects* is done by calculating the invariant mass for all two object combinations and make a summary of three invariant mass of two object combinations.
In the TME there are two thresholds for M: "greater/equal lower limit" and "less/equal upper limit", given in GeV (floating point notation) with one position after decimal point in even numbers.
The calculation of *invariant mass calculation for three objects* is done in modules `mass_-invariant_calc.vhd` and `calo_calo_mass_3_obj_condition.vhd` or `muon_muon_-mass_3_obj_condition.vhd` (for summary of mass and comparison).

2.4.2.8 Muon charge correlation module

For definition of muon charge, see 2.4.8.
In the muon charge correlation module (`muon_charge_correlations.vhd`), the charge correlations are made for different muon conditions-types. The charges of objects (number of objects depends on muon condition type) are compared to get "like sign charge" ("LS") or "opposite sign charge" ("0S"), "LS" means that the charges (charge sign)of objects are the same, "0S" means that at least one object has different charge than the others. This information is used in all instatiated muon conditions.
There is no charge correlation for single type conditions.
In all cases the "charge valid" bit of the objects must be set.
In TME one can select "LS", "0S" or ignore for charge correlation in muon conditions.

Table 2: Muon charge correlation - Double Muon

x x	I ignore (charge x = +, -, I)
+ +	LS both positive muons
− −	LS both negative muons
I I	LS both muons with the same sign, positive or negative
+ −	OS two muons of opposite sign
− +	OS idem
I I	OS idem

Table 3: Muon charge correlation - Triple Muon

x x x	I ignore (charge x = +, -, I)
+ + +	LS three muons of positive charge
− − −	LS three muons of negative charge
I I I	LS three muons of the same sign (positive or negative)
+ + −	OS a pair plus a positive muon
+ − −	OS a pair plus a negative muon
+ − I	OS a pair plus a negative or positive muon

Table 4: Muon charge correlation - Quad Muon

x x x x	I ignore (charge x = +, -, I)
+ + + +	LS four muons of positive charge
− − − −	LS four muons of negative charge
I I I I	LS four muons of the same sign (positive or negative)
+ + + −	OS a pair plus two positive muons
+ + − −	OS two pairs
+ − − −	OS a pair plus two negative muons
+ − I I	OS a pair plus two negative or positive muons

2.4.3 Calorimeter conditions

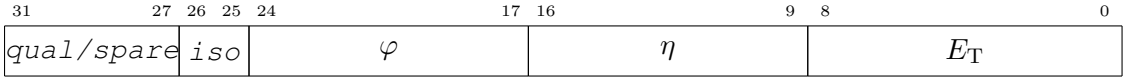
2.4.3.1 Calorimeter data

The calorimeter trigger processing identifies **electron/γ, jet and tau** objects and **energy sum quantities**.

electron/γ:
Twelve objects are passed to the μGT for each event.
For each selected object, the Calo-Layer2 sends parameters for E_{T} and for position and quality information - encoded in 32 bits:

- 9 bits E_{T}, range = 0..255 GeV (HW index = 0..0x1FF), step = 0.5, the highest bin will mark an overflow (HW index 0x1FF): meaning has to be defined
- 8 (7+1 sign) bits pseudo-rapidity (η) position, range = -3.0 to 3.0, step = 0.087/2, linear scale, 138 bins (HW index = 0xBC..0x44)
- 8 bits azimuth angle (φ) position, range = 2π, step $\approx 2\pi/144$ ($\triangleq$2.5°), 144 bins (HW index = 0..0x8F), HW index starting at 0° (anti-clockwise)
- 2 bits isolation (meaning not defined yet!)
- 5 bits quality and spare (not defined yet!)

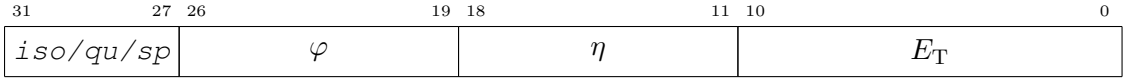
The data structure of an electron/γ object (bits 27..31 are not defined yet, reserved for quality, ...):



jet:
Twelve objects are passed to the μGT for each event.
For each selected object, the Calo-Layer2 sends parameters: E_{T}, for position and quality information - encoded in 32 bits:

- 11 bits E_{T}, range = 0..1023 GeV (HW index = 0..0x7FF), step = 0.5, the highest bin will mark an overflow (HW index 0x7FF): meaning has to be defined
- 8 (7+1 sign) bits pseudo-rapidity (η) position, range = -5.0 to 5.0, step = 0.087/2, linear scale, 230 bins (HW index = 0x8E..0x72)
- 8 bits azimuth angle (φ) position, range = 2π, step $\approx 2\pi/144$ ($\triangleq$2.5°), 144 bins (HW index = 0..0x8F), HW index starting at 0° (anti-clockwise)
- 5 bits quality and spare (not defined yet!)

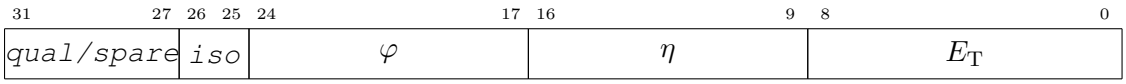
The data structure of a jet object (bits 27..31 are not defined yet, reserved for quality, ...):



tau:
Twelve objects are passed to the μGT for each event.
For each selected object, the Calo-Layer2 sends parameters for E_T and for position and quality information - encoded in 32 bits:

- 9 bits E_T, range = 0..255 GeV (HW index = 0..0x1FF), step = 0.5, the highest bin will mark an overflow (HW index 0x1FF): meaning has to be defined
- 8 (7+1 sign) bits pseudo-rapidity (η) position, range = -3.0 to 3.0, step = 0.087/2, linear scale, 138 bins (HW index = 0xBC..0x44)
- 8 bits azimuth angle (φ) position, range = 2π, step $\approx 2\pi/144$ ($\hat{=}$2.5°), 144 bins (HW index = 0..0x8F), HW index starting at 0° (anti-clockwise)
- 2 bits isolation (meaning not defined yet!)
- 5 bits quality and spare (not defined yet!)

The data structure of a tau object (bits 27..31 are not defined yet, reserved for quality, ...):



The representation of the 8 bits (called "hardware index [HW index]") in η is expected as Two's Complement notation as shown in Table 6.

Table 5: η scale of electron/γ and tau

HW index	η **range**	η **bin**
0x44	68∗0.087/2 to 69∗0.087/2	68
...	...	...
0x01	0.087/2 to 2∗0.087/2	1
0x00	0 to 0.087/2	0
0xFF	0 to -0.087/2	-1
0xFE	-0.087/2 to -2∗0.087/2	-2
...	...	...
0xBC	-68∗0.087/2 to -69∗0.087/2	-69

The representation of the 8 bits in φ is expected as shown in Table 7.

The representation of the 2 bits for isolation (e/γ and tau) is expected as shown in Table 8.

Table 6: η scale of jet

HW index	η **range**	η **bin**
0x72	114∗0.087/2 to 115∗0.087/2	114
...	...	...
0x01	0.087/2 to 2∗0.087/2	1
0x00	0 to 0.087/2	0
0xFF	0 to -0.087/2	-1
0xFE	-0.087/2 to -2∗0.087/2	-2
...	...	...
0x8E	-114∗0.087/2 to -115∗0.087/2	-115

Table 7: φ scale of calorimeter objects

HW index	φ range	φ range [degrees]	φ bin
0x00	0 to $2\pi/144$	0 to 2.5	0
0x01	$2\pi/144$ to $2*2\pi/144$	2.5 to 5.0	1
...	...	...	...
0x8F	$143*2\pi/144$ to 2π	357.5 to 360	143

Table 8: Definition of e/γ and tau isolation bits

bits [26..25]	definition
00	not isolated
01	isolated
10	TBD
11	TBD

2.4.3.2 Calorimeter conditions definition

A condition consists of calorimeter objects as input data and a set of requirements, which contain the requirements to be complied.

The requirement for calorimeter conditions contains:
one threshold for E_{T}, ranges for η, φ LUTs for isolation and differences in η and φ. In addition the selection of the "relative bx" of objects is done in the requirement.
The condition is complied, if every comparison between object parameters and requirements is valid for the following equation:

- E_{T} greater-equal (or equal) threshold
- η in range
- φ in range
- isolation as requested (for electron/γ and tau)

Additional comparisons for "quality information" could be part of the equation - but not defined yet.

There are different types of calorimeter conditions implemented, depending of how many objects have to comply the requirements.

- "Quad objects requirements condition": this condition type consists of requirements for 4 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 4 different objects, each of which fulfills at least one of the requirements.
- "Triple objects requirements condition": this condition type consists of requirements for 3 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 3 different objects, each of which fulfills at least one of the requirements.
- "Double objects requirements condition": this condition type consists of requirements for 2 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 2 different objects, each of which fulfills at least one of the requirements.[4]
- "Single object requirement condition": this condition type consists of one requirement for one trigger object of a given object type. To fulfill this condition, there must exist at least one object which fulfills the requirement.

The selection of the mode of E_{T}-comparator (greater/equal or equal), the E_{T}-threshold-value, ranges for η and φ set with thresholds, LUTs for isolation and ranges for differences in η and φ set with thresholds are fixed values given by VHDL Producer for every Trigger Menu. The objects have to be of same type and same bunch-crossing.

[4] "Double objects requirements condition with spatial correlation" not used anymore, replaced by Correlation conditions

2.4.3.2.1 Calorimeter conditions module

The module for conditions with calorimeter objects (`calo_conditions.vhd`) instantiates the calorimeter comparators module (`calo_comparators.vhd`) as many times as the numbers of objects and requirements determine. Depending on the condition-type, different and-or-structures of object vs. requirement are selected. For comparison in η, φ and the differences in η and φ, "window"-comparators are used.
In the calorimeter conditions module a cut for two-body pt calculation can be selected (see 2.4.2.6). Therefore a threshold value for two-body pt is required.

2.4.3.2.2 Calorimeter Overlap Remover conditions module

The Calorimeter Overlap Remover conditions consits of a Calorimeter condition (2.4.3.2.1) and a single condition for a different calo object type. One or more correlation cut(s) ($\Delta\eta$, $\Delta\varphi$ and ΔR - 2.4.9) for overlap removal is required between different calo object types. Overlap Remover conditions `calo_conditions_orm.vhd` are implemented only for calo object types.

2.4.3.2.3 Calorimeter conditions module - template for VHDL-Producer

See in Chapter **??** and in Listing **??** for a VHDL-template for VHDL-Producer of instantiating a calorimeter condition (`calo_conditions.vhd`).

2.4.3.2.4 Calorimeter comparators module

A comparator between the energy (E_{T}) and a threshold (et_threshold) and a comparison in η with five "window"-comparators and φ with two "window"-comparators is done in this basic module. The values for E_{T} threshold, the 'mode-selection' for the E_{T} comparator and the "limits" of the "window"-comparators is given in the generic interface list of the module. Additionally the data-structure of input data (data_i in port interface list) is provided as a record in this list. The output signal of the module is in high state, if all comparisons are true.
The comparison in η is done with five "window"-comparators, so one gets max. five ranges for η. The η value (HW index) has a Two's Complement notation, the comparisons is done signed. Number of windows is given for η.
The comparison in φ is done with two "window"-comparators, so one gets two ranges for φ. The comparisons is done unsigned. There are two flags, one for "full-range" and one for "ignore-second-window" for the selection of the ranges.
There are two cases how the limits of one "window"-comparator could be set (see also Figure 4 and Listing 3):

- Upper limit is less than lower limit => φ range between the limits, including the φ bin with value = 0 (HW index).

- Upper limit is greater/equal than lower limit => φ range between the limits, not including the φ bin with value = 0 (HW index).

The comparison of isolation (for electron/γ and tau) is done with LUTs.

Listing 3: VHDL code of "window"-comparator in φ

```
phi_comp_w1 <= '1' when phi_w1_upper_limit < phi_w1_lower_limit and
                (phi <= phi_w1_upper_limit or phi >= phi_w1_lower_limit) else
               '1' when phi_w1_upper_limit >= phi_w1_lower_limit and
                (phi <= phi_w1_upper_limit and phi >= phi_w1_lower_limit)
               else '0';
```

The values of η and φ have to be inside of only one of the required ranges ("or").

The comparison of isolation (for electron/γ and tau) is done with LUTs (see Table 9). Only the least significant 4 bits of LUT are used, because currently 2 isolation bits are defined.

Table 9: LUT contents for isolation comparison of electron/γ and tau objects

LUT content (16 bits)	isolation bits [26..25]	trigger
X"0"	xx	no trigger
X"1"	00	trigger on isolation bits = 00
X"2"	01	trigger on isolation bits = 01
X"3"	00 or 01	trigger on isolation bits = 00 or 01
X"4"	10	trigger on isolation bits = 10
X"5"	00 or 10	trigger on isolation bits = 00 or 10
X"6"	01 or 10	trigger on isolation bits = 01 or 10
X"7"	00 or 01 or 10	trigger on isolation bits = 00 or 01 or 10
X"8"	11	trigger on isolation bits = 11
X"9"	00 or 11	trigger on isolation bits = 00 or 11
X"A"	01 or 11	trigger on isolation bits = 01 or 11
X"B"	00 or 01 or 11	trigger on isolation bits = 00 or 01 or 11
X"C"	10 or 11	trigger on isolation bits = 10 or 11
X"D"	00 or 10 or 11	trigger on isolation bits = 00 or 10 or 11
X"E"	01 or 10 or 11	trigger on isolation bits = 01 or 10 or 11
X"F"	00 or 01 or 10 or 11	trigger on isolation bits = 00 or 01 or 10 or 11 (= "ignore" isolation)

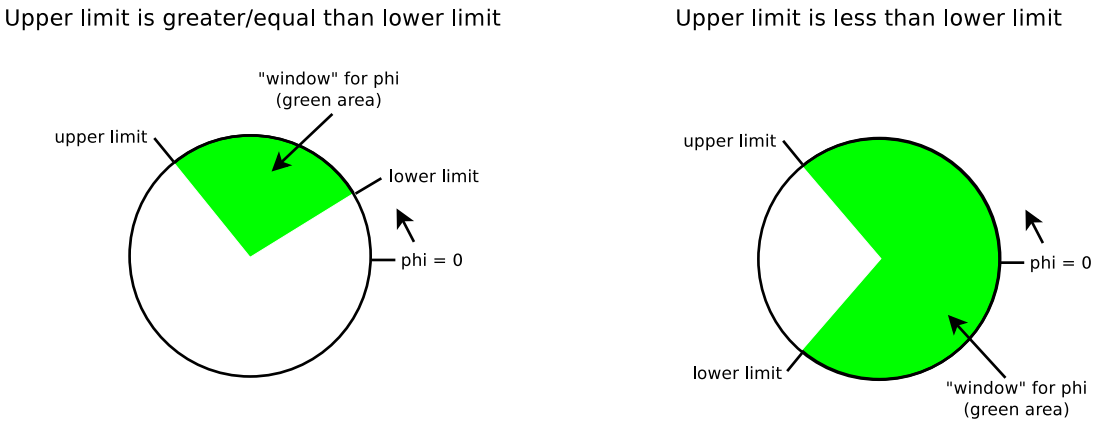


Figure 4: Setting the limits for "window"-comparators for φ

2.4.4 Energy sum quantities conditions

energy sum quantities:
Consists of following quantities (naming convention see 3):

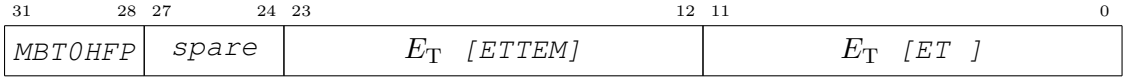
- **ET**
- **HT**
- ET_{miss}
- HT_{miss}
- **ETTEM**
- $\mathbf{ET}^{HF}_{miss}$
- $\mathbf{HT}^{HF}_{miss}$
- **ASYMET**
- **ASYMHT**
- **ASYMETHF**
- **ASYMHTHF**
- **CENT0**
- ..
- **CENT7**

2.4.4.1 Energy sum quantities data

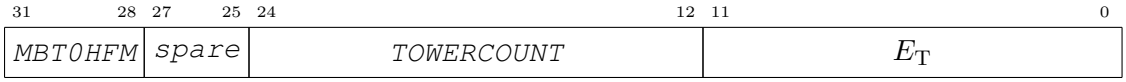
Calo-Layer2 sends 6 frames (each 32 bits) with Energy sum quantities containing the following information:

- E_{T}, 12 bits, range = 0..2047 GeV (HW index = 0..0xFFF), step = 0.5, the highest bin will mark an overflow (HW index 0xFFF): meaning has to be defined
- azimuth angle (φ) position, 8 bits, range = 2π, step $\approx 2\pi/144$ ($\hat{=}2.5°$), 144 bins (HW index = 0..0x8F), HW index starting at 0° (anti-clockwise)
- "Towercount", 13 bits, range = 0..8191
- "Minimum bias", 4 bits, range = 0..15
- "Asymmetry", 8 bits, range = 0..255 (used 0..100)
- "Centrality", 8 bits, used as signals

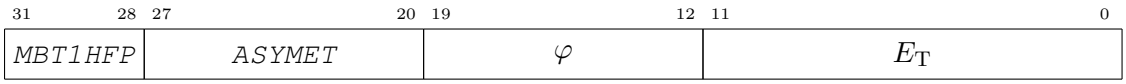
Frame0: The data structure of "total Et" (ET) quantity [including "total Et from ECAL only" (ETTEM) and "minimum bias HF+ threshold 0" bits]:



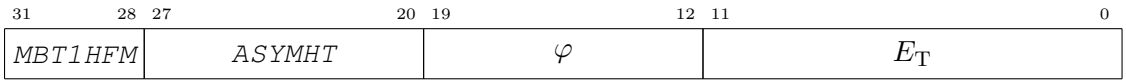
Frame1: The data structure of "total calibrated Et in jets" (HT) quantity [including "towercount" and "minimum bias HF- threshold 0" bits]:



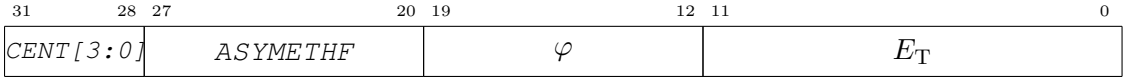
Frame2: The data structure of "missing Et" (ET_{miss}) quantity [including "Asymmetry" ASYMET and "minimum bias HF+ threshold 1" bits]:



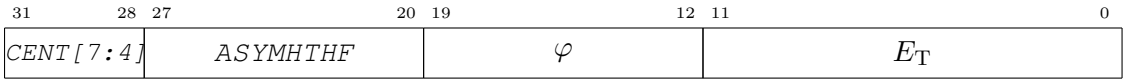
Frame3: The data structure of "missing Ht" (HT_{miss}) quantity [including "Asymmetry" ASYMHT and "minimum bias HF- threshold 1" bits]:



Frame4: The data structure of "missing Et including HF" (ET^{HF}_{miss}) quantity [including "Asymmetry" ASYMETHF and "Centrality" bits (3:0)]:



Frame5: The data structure of "missing Ht including HF" (HT^{HF}_{miss}) quantity [including "Asymmetry" ASYMHTHF and "Centrality" bits (7:4)]:



2.4.4.2 Energy sum quantities conditions module (including Asymmetry conditions)

A comparator between E_T and a threshold (et_threshold) and, depending on object type, a comparison in φ with two "window"-comparators is done in this module. The value for E_T threshold, the 'mode-selection' for the E_T comparator and the limits for the "window"-comparators are given in the generic interface list of the module. The selection whether a comparison in φ is part of the condition is done with the value of the generic parameter 'obj_type' ('ETM_TYPE', 'ETMHF_TYPE', 'HTM_TYPE' and 'HTMHF_TYPE' force a comparison). The comparison in φ is done in the same way as for calorimeter conditions (see 2.4.3.2.4). Additionally the data-structure of input data (data_i in port interface list) is provided as a record in this list. The output signal of the module is in high state, if all comparisons are true.
Data for Asymmetry trigger are received on 4 frames on bits 27..20 (8 bits). For every type a comparision with an 8-bit threshold (greater-equal [or equal]) is done. Asymmetry data are interpreted as counts.

2.4.4.3 Energy sum quantities conditions module - template for VHDL-Producer

A VHDL-template for VHDL-Producer of instantiating `esums_conditions.vhd` is given below (see Listing **??**).

2.4.5 Minimum bias trigger conditions

Data for Minimum bias trigger are received on the 4 MSBs of 4 frames used for Energy sum quantities (see 2.4.4).

- MBT0HFP: "minimum bias HF+ threshold 0" bits
- MBT0HFM: "minimum bias HF- threshold 0" bits
- MBT1HFP: "minimum bias HF+ threshold 1" bits
- MBT1HFM: "minimum bias HF- threshold 1" bits

In the min_bias_hf_conditions.vhd module there is a comparision with a 4-bit threshold (greater-equal [or equal]).

2.4.6 Towercount condition

Data for Towercount trigger (number of firing HCAL towers) are received on frame HT (see 2.4.4) on bits 24..12 (13 bits) of HT data structure. In the towercount_condition.vhd module there is a comparision with a 13-bit threshold (greater-equal [or equal]).

2.4.7 Centrality condition

Centrality bits used as a signals for triggers (similar to external signals).

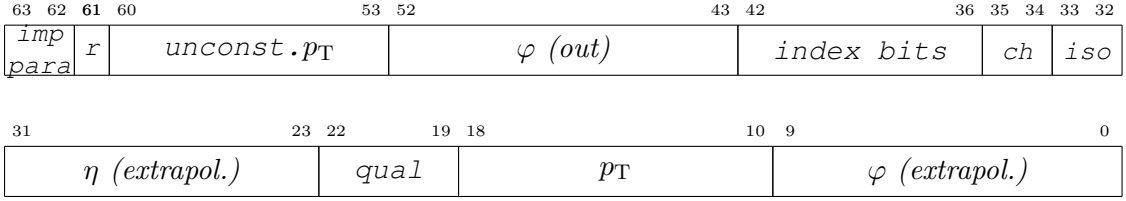
2.4.8 Muon conditions

2.4.8.1 Muon data

Eight Muon objects are provided by Global Muon Trigger. One Muon object has a 64 bits data structure with parameters for p_T, for position, charge, quality and isolation information:

- 10 bits azimuth angle (φ) position, range = 2π, step $\approx 2\pi/576$ ($\hat{=}$0.625°), 576 bins (HW index = 0..0x23F), HW index starting at 0° (anti-clockwise)
- 9 bits p_T, range = 0..255 GeV (HW index = 0..0x1FF), step = 0.5, the highest bin will mark an overflow (HW index 0x1FF): meaning has to be defined
- 4 bits quality, 16 types for quality (meaning not defined yet!)
- 9 (8+1 sign) bits pseudo-rapidity (η) position, range = -2.45 to 2.45, step = 0.087/8, linear scale, 452 bins (-225..225, HW index = 0x11F..0x0E1)
- 2 bits isolation, 4 types for isolation (meaning not defined yet!)
- 1 bit charge sign, charge sign = '0' means "positive" charge, charge sign = '1' means "negative" charge
- 1 bit charge valid (='1' means "valid")
- 7 index bits
- 10 bits azimuth angle (φ) position, raw data
- 8 bits unconstrained p_T, range = 0..255 GeV (HW index = 0..0xFF), step = 1.0, the highest bin will mark an overflow (HW index 0xFF)
- 1 spare bit
- 2 bits impact parameter

The data structure of a muon object (64 bits - bit 34 = charge sign, bit 35 = charge valid, bit 61 is a spare bit, bit 63..62 = impact parameter):



The representation of the 9 bits (called "hardware index [HW index]") in η is expected as Two's Complement notation as shown in Table 10.
The central value of the bin 0 (-0.010875/2 to +0.010875/2) = 0.0, the left edge of the bins will range from $-255 \times 0.010875 - 0.010875/2 = -2.7785625$ to $+255 \times 0.010875 - 0.010875/2 = 2.7676875$. The central value of the bins will range between ± 2.773125 . The physical η range of the muon detectors is about ± 2.45, so that not all possible η bins will be used.

Table 10: η scale of muon objects

HW index	η range	η bin
0x0E1	224.5∗0.087/8 to 225.5∗0.087/8	225
0x0E0	223.5∗0.087/8 to 224.5∗0.087/8	224
...	...	...
0x001	0.5∗0.087/8 to 1.5∗0.087/8	1
0x000	0.5∗-0.087/8 to 0.5∗0.087/8	0
0x1FF	0.5∗-0.087/8 to 1.5∗-0.087/8	-1
0x1FE	1.5∗-0.087/8 to -2.5∗0.087/8	-2
...	...	...
0x11F	-224.5∗0.087/8 to -225.5∗0.087/8	-225

The representation of the 10 bits in φ is expected as shown in Table 11.

Table 11: φ scale of muon objects

HW index	φ range	φ range [degrees]	φ bin
0x000	0 to $2\pi/576$	0 to 0.625	0
0x001	$2\pi/576$ to $2*2\pi/576$	0.625 to 1.250	1
...	...	...	...
0x23F	$575*2\pi/576$ to 2π	359.375 to 360	575

The representation of the 4 bits for quality is expected as shown in Table 12.

The representation of the 2 bits for isolation is expected as shown in Table 14.

The representation of the 2 bits for impact parameter is expected as shown in Table 14.

Table 12: Definition of muon quality bits

bits [22..19]	definition
0000	quality "level 0"
0001	quality "level 1"
0010	quality "level 2"
0011	quality "level 3"
0100	quality "level 4"
0101	quality "level 5"
0110	quality "level 6"
0111	quality "level 7"
1000	quality "level 8"
1001	quality "level 9"
1010	quality "level 10"
1011	quality "level 11"
1100	quality "level 12"
1101	quality "level 13"
1110	quality "level 14"
1111	quality "level 15"

Table 13: Definition of muon isolation bits

bits [33..32]	definition
00	not isolated
01	isolated
10	TBD
11	TBD

Table 14: Definition of muon impact parameter bits

bits [63..62]	definition
00	TBD
01	TBD
10	TBD
11	TBD

2.4.8.2 Muon conditions definition

A condition consists of input-data and a set of requirements, which contain the requirements to be complied.

The requirement for muon conditions contains:
a threshold for p_{T}, a threshold for unconstrained p_{T}, ranges for η and φ, a LUT for quality, a LUT for isolation, a requsted charge. The condition is complied, if every comparison between object parameters and requirements is valid for the following equation:

- p_{T} greater-equal (or equal) threshold
- unconstrained p_{T} greater-equal (or equal) threshold
- η in range
- φ in range
- requested charge
- quality LUT
- iso LUT

There are different types of calorimeter conditions implemented, depending of how many objects have to comply the requirements.

- "Quad objects requirements condition": this condition type consists of requirements for 4 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 4 different objects, each of which fulfills at least one of the requirements.
- "Triple objects requirements condition": this condition type consists of requirements for 3 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 3 different objects, each of which fulfills at least one of the requirements.
- "Double objects requirements condition": this condition type consists of requirements for 2 different trigger objects of the same object type. For each object the requirements can be different. To fulfill this condition, there must exist at least one set of 2 different objects, each of which fulfills at least one of the requirements.[5]
- "Single object requirement condition": this condition type consists of one requirement for one trigger object of a given object type. To fulfill this condition, there must exist at least one object which fulfills the requirement.

In addition requested charge correlation must be matched (except for "Single object requirement condition", there is no charge correlation). The calculation of charge correlations is done in an own module in the top-of-hierarchy module (`gtl_module.vhd`).

[5] "Double objects requirements condition with spatial correlation" not used anymore, replaced by Correlation conditions

2.4.8.2.1 Muon conditions module

A module for conditions with muon objects (`muon_conditions.vhd`) instantiates the muon comparators module (`muon_comparators.vhd`) as many times as the numbers of objects and requirements determine. Depending on the condition-type different and-or-structures of object vs. requirement are selected.

2.4.8.2.2 Muon comparators module

A comparator between p_T and a threshold (pt_threshold), a comparator between unconstrained p_T and a threshold (upt_threshold), a comparison in η with five "window"-comparators and φ with two "window"-comparators, a comparison of quality with LUT, a comparison of isolation with LUT and a comparison of the requested charge is done in this basic module. The values for p_T threshold, unconstrained p_T threshold, the 'mode-selection' for the p_T comparator, the "limits" of the "window"-comparators, the quality LUTs, the isolation LUTs and the requested charge is given in the generic interface list of the module. Additionally the data-structure of input data (data_i in port interface list) is provided as a record in this list. The output signal of the module is in high state, if all comparisons are true.
The comparison in η is done with five "window"-comparators, so one gets max. five ranges for η. The η value (HW index) has a Two's Complement notation, the comparisons is done signed. Number of windows is given for η.
The comparison in φ is done with two "window"-comparators, so one gets two ranges for φ. The comparisons is done unsigned. There are two flags, one for "full-range" and one for "ignore-second-window" for the selection of the ranges.
There are two cases how the limits of one "window"-comparator could be set (see also Figure 4 and Listing 3):

- Upper limit is less than lower limit => φ range between the limits, including the φ bin with value = 0 (HW index).
- Upper limit is greater/equal than lower limit => φ range between the limits, not including the φ bin with value = 0 (HW index).

The values of η and φ have to be inside of only one of the two required ranges ("or").

Charge valid and charge sign bits must be equal to the requested charge.
The comparison of quality is done with LUT. To ignore quality comparison, all bits in the LUT have to be '1'.

The comparison of isolation is done with LUT. To ignore isolation comparison, all bits in the LUT have to be '1' (see Table 16).

Table 15: LUT contents for quality comparison of muon objects

LUT content (16 bits)	quality bits [22..19]	trigger
X"0000"	xxxx	no trigger
X"0001"	0000	trigger on quality "level 0"
X"0002"	0001	trigger on quality "level 1"
X"0003"	0001 or 0000	trigger on quality "level 1" or "level 0"
X"0004"	0010	trigger on quality "level 2"
...	...	...
X"8000"	1111	trigger on quality "level 15"
X"C000"	1111 or 1110	trigger on quality "level 15" or "level 14"
...	...	...
X"FFFF"	xx	trigger on all quality "levels" (= "ignore")

Table 16: LUT contents for isolation comparison of muon objects

LUT content (4 bits)	isolation bits [33..32]	trigger
X"0"	xx	no trigger
X"1"	00	trigger on isolation bits = 00
X"2"	01	trigger on isolation bits = 01
X"3"	00 or 01	trigger on isolation bits = 00 or 01
X"4"	10	trigger on isolation bits = 10
X"5"	00 or 10	trigger on isolation bits = 00 or 10
X"6"	01 or 10	trigger on isolation bits = 01 or 10
X"7"	00 or 01 or 10	trigger on isolation bits = 00 or 01 or 10
X"8"	11	trigger on isolation bits = 11
X"9"	00 or 11	trigger on isolation bits = 00 or 11
X"A"	01 or 11	trigger on isolation bits = 01 or 11
X"B"	00 or 01 or 11	trigger on isolation bits = 00 or 01 or 11
X"C"	10 or 11	trigger on isolation bits = 10 or 11
X"D"	00 or 10 or 11	trigger on isolation bits = 00 or 10 or 11
X"E"	01 or 10 or 11	trigger on isolation bits = 01 or 10 or 11
X"F"	00 or 01 or 10 or 11	trigger on isolation bits = 00 or 01 or 10 or 11 (= "ignore" isolation)

2.4.9 Correlation conditions

The correlation conditions contain a combination of two "Single object requirement conditions" of two object types or one "Double objects requirement condition" of objects of the same type. In addition with "object requirements" there are cuts for $\Delta\eta$, $\Delta\varphi$, ΔR, mass and "two-body pt".

The following cuts can be used:

- Cut for $\Delta\eta$ (DETA).
- Cut for $\Delta\varphi$ (DPHI).
- Cut for ΔR (DR).
- Cuts for mass (MASS) of following mass types:
  - Cut for Invariant mass.
  - Cut for Invariant mass with unconstrained pt (only for muons).
  - Cut for Invariant mass divided by ΔR.
  - Cut for Transverse mass.
- Cut for Two-body pt.

There is one correlation condition type for a mass cut with three objects:

- Cut for invariant mass for three objects (MASS).

2.4.9.1 Correlation condition types

As described in section Correlation conditions (2.4.9), correlations of two object types are available. Following possible correlations (objects 1-objects 2) are available, all implementated with `correlation_conditions.vhd`:

- Correlation condition with calorimeter objects
  : electron/γ-electron/γ, electron/γ-jet, electron/γ-tau, jet-jet, jet-tau and tau-tau are possible.)
- Correlation condition for mass divided by ΔR with calorimeter objects
  (: electron/γ-electron/γ, electron/γ-jet, electron/γ-tau, jet-jet, jet-tau and tau-tau are possible.) item Correlation condition with calorimeter objects and muons objects
  : electron/γ-muon, jet-muon and tau-muon are possible.)
- Correlation condition with muon objects

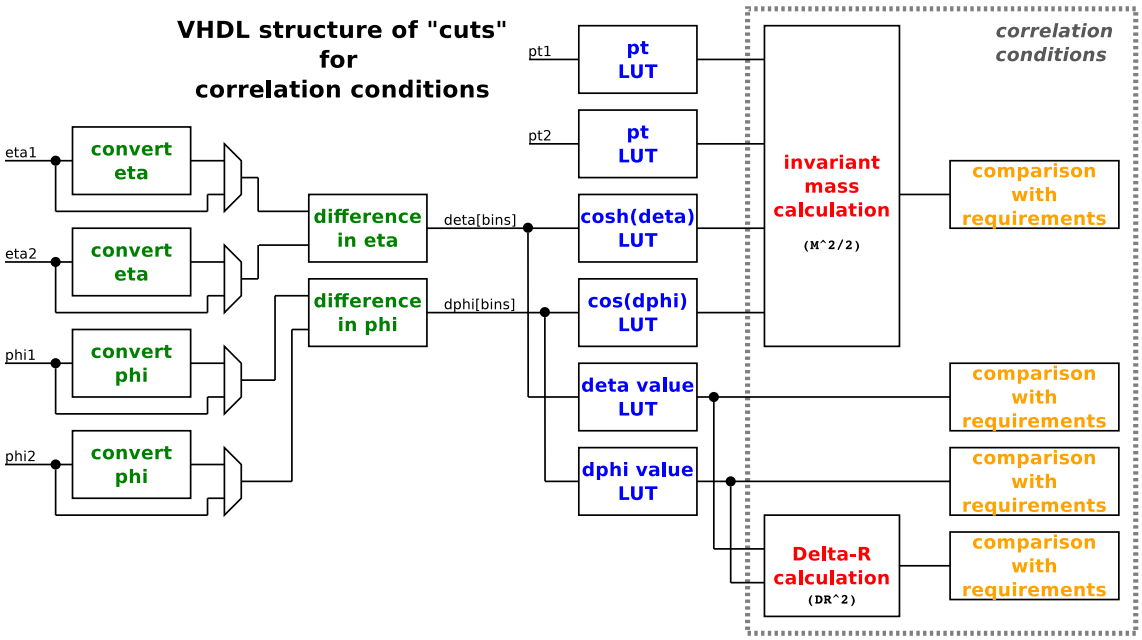


Figure 5: VHDL structure of cuts for correlation conditions

- Correlation condition with calorimeter objects and energy sum quantities (ET_{miss}, ET^{HF}_{miss} and HT_{miss} only)
  : electron/γ-etm, jet-etm, tau-etm, electron/γ-htm, jet-htm, tau-htm, electron/γ-etmhf, jet-etmhf and tau-etmhf are possible.)
- Correlation condition with muon objects
- Correlation condition for mass divided by ΔR with muon objects
- Correlation condition with muon objects and energy sum quantities (ET_{miss}, ET^{HF}_{miss} and HT_{miss} only)
  : muon-etm, muon-etmhf and muon-htm are possible.)

There is one modules for calorimeter objects correlation condition with calorimeter object overlap removal:

- Correlation condition for calorimeter objects with calorimeter object overlap removal (`calo_calo_calo_correlation_orm_condition.vhd`

There are two modules for mass with three objects:

- Correlation condition for mass with three objects with calorimeter objects (same type, same bunch-crossing)
  (`calo_calo_mass_3_obj_condition.vhd`
- Correlation condition for mass with three objects with muon objects
  (`muon_muon_mass_3_obj_condition.vhd`)

2.4.9.1.1 Overview of possible correlation cuts in conditions

The following list gives an overview of possible correlation cuts in conditions:

- Calo conditions (calo_conditions.vhd):
  - two-body pt (for double condition)
- Calo conditions overlap removal (calo_conditions_orm.vhd):
  - $\Delta\eta$ overlap removal
  - $\Delta\varphi$ overlap removal
  - ΔR overlap removal
  - two-body pt (for double condition)
- Muon conditions (muon_conditions.vhd):
  - charge correlation
  - two-body pt (for double condition)
- Calo calo correlation condition with calo overlap removal (calo_calo_calo_correlation_orm_condition.vhd):
  - $\Delta\eta$ overlap removal
  - $\Delta\varphi$ overlap removal
  - ΔR overlap removal
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt
- Calo calo correlation condition (correlation_conditions.vhd):
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt
- Calo calo correlation condition for invariant mass divided by ΔR (correlation_conditions.vhd):
  - invariant mass divided by ΔR
- Calo calo correlation condition mass with three objects (calo_calo_mass_3_obj_condition.vhd):

  - invariant mass with three objects

- Calo muon correlation condition (correlation_conditions.vhd):
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt
- Calo esums correlation condition (correlation_conditions.vhd):
  - $\Delta\varphi$
  - transverse mass
  - two-body pt
- Muon muon correlation condition (correlation_conditions.vhd):
  - charge correlation
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass or invariant mass unconstraint pt
  - two-body pt
- Muon muon correlation condition for invariant mass divided by ΔR (correlation_conditions.vhd):
  - charge correlation
  - invariant mass divided by ΔR
- Muon muon correlation condition mass with three objects (muon_muon_mass_3_obj_condition.vhd):
  - charge correlation
  - invariant mass with three objects
- Muon esums correlation condition (correlation_conditions.vhd):
  - $\Delta\varphi$
  - transverse mass
  - two-body pt

2.4.9.1.2 Correlation condition types

The VHDL module `correlation_conditions.vhd` is instantiated for all correlation conditions with two objects. The correlation conditions module contains two "Single object requirement conditions" for different object types or same type with data from different bunch-crossings as one possible mode and a "Double objects requirement condition" objects of same type and same bunch-crossing as a second mode (selection is done by a parameter in the generic list of `correlation_conditions.vhd` named "same_bx").
In addition there are "Cuts" for differences in η (DETA) and φ (DPHI), a calculation of ΔR (DR), a calculation of invariant mass (MASS) or invariant mass with unconstrained pt (MASS) or invariant mass divided by ΔR (MASS) or transverse mass (MASS) and a calculation of two-body pt, see Figure 5.
The calculation of all these cuts is done outdside of correlation conditions for all needed object type bunch-crossing combinations. Results of these calculations are inputs to the instantiated correlation conditions module depending which cuts are used in a certain condition. The comparison with limits or thresholds is part of this module.

2.4.9.1.3 Calo Calo Correlation condition module with Calo Overlap Remover

The Calo Calo Overlap Remover Correlation conditions consits of two modes. One with a Calo Calo Correlation condition with "Double objects requirement condition" for calo objects of same type and same bunch-crossing (**??**) and a single condition for a different calo object type (can have different bunch-crossing too). There has to be at least one correlation cut for the objects of "Double objects requirement condition" and a correlation cut for overlap removal between objects (one or more cut(s) of $\Delta\eta$, $\Delta\varphi$ and ΔR) of different object types ("2plus1"). A second mode ("1plus1") with a Calo Calo Correlation condition with a single condition and a different calo object type (can have different bunch-crossing too) also with a single condition. There has to be at least one correlation cut for the different objects (e.g. invariant mass) and a correlation cut for overlap removal between the objects (one or more cut(s) of $\Delta\eta$, $\Delta\varphi$ and ΔR).
Overlap Remover Correlation conditions `calo_calo_calo_correlation_orm_condition.vhd` are implemented only for calo object types.

2.4.9.1.4 Calo Calo Correlation condition module for Invariant Mass with Three Objects

The calo correlation condition module for invariant mass with three objects (`calo_calo_mass_3_obj_condition.vhd`) contains a "Triple objects requirement condition" for calo objects of same type and same bunch-crossing.
In addition a "Cut" for calculation of *invariant mass with three objects* (see 2.4.2.7) is mandatory.
No other correlation cuts available in this condition type.

2.4.9.1.5 Muon Muon Correlation condition module for Invariant Mass with Three Objects

The muon correlation condition module for invariant mass with three objects (`muon_muon_mass_3_obj_condition.vhd`) contains a "Triple objects requirement condition" for objects of same type and same bunch-crossing.
In addition a "Cut" for calculation of *invariant mass with three objects* (see 2.4.2.7) is mandatory.
No other correlation cuts, except "charge correlation", available in this condition type.

2.4.10 External Conditions

Maximal 256 External Conditions are possible in Global Trigger. They are provided as inputs in the Algorithms logic of μGTL. External Conditions will include the "Technical Trigger" of the legacy system.

2.4.11 Algorithms logic

The outputs of all the instantiated conditions are combined in the Algorithms logic with boolean algebra given by TME for every single Algorithm. These Algorithms are registered and provided as inputs for Final Decision Logic.

3 Glossary

electron/γ = electron/gamma objects over Calo-Layer2 (VHDL: eg)

jet = jet objects over Calo-Layer2 (VHDL: jet)

tau = tau objects over Calo-Layer2 (VHDL: tau)

muon = muon objects over μGMT (VHDL: muon)

ET = Scalar sum of transverse energy components over Calo-Layer2 (VHDL: ett)

ETTEM = Scalar sum of transverse energy components from ECAL only over Calo-Layer2 (VHDL: ettem)

MBTxHFy = Minimum bias HF bits (VHDL: MBT0HFP, MBT0HFM, MBT1HFP, MBT1HFM)

HT = Magnitude of the vectorial sum of transverse energy of jets (hadronic) over Calo-Layer2 (VHDL: htt)

TOWERCOUNT = tower counts (VHDL: towercount)

ET_{miss} = 2-vector sum of transverse energy over Calo-Layer2 (VHDL: etm)

HT_{miss} = Missing Total transverse energy of jets over Calo-Layer2 (VHDL: htm)

$\mathbf{ET}^{HF}_{miss}$ = 2-vector sum of transverse energy including HF over Calo-Layer2 (VHDL: etmhf)

$\mathbf{HT}^{HF}_{miss}$ = Missing Total transverse energy of jets including HF over Calo-Layer2 (VHDL: htmhf)

ASYMET = Asymmetry of ET over Calo-Layer2 (VHDL: asymet)

ASYMHT = Asymmetry of HT over Calo-Layer2 (VHDL: asymht)

ASYMETHF = Asymmetry of ET including HF over Calo-Layer2 (VHDL: asymethf)

ASYMHTHF = Asymmetry of HT including HF over Calo-Layer2 (VHDL: asymhthf)

CENTx = Centrality bits [7:0] over Calo-Layer2 (VHDL: cent7, cent6, ...)

p_{T} = transverse momentum of muon objects(VHDL: pt)

E_{T} = energy of calorimeter objects (VHDL: et)

η = pseudo-rapidity position (VHDL: eta)

φ = azimuth angle position (VHDL: phi)

isolation = isolation information (VHDL: iso)

quality = quality information (VHDL: qual)

Acronyms

DAQ Data Acquisition

DM Delay Manager Module

FDL Final Decision Logic Module

GTL Global Trigger Logic Module

ROP Readout-Process Module

TCM Timing Counter Manager Module

TCS Trigger Control System

GCT Calorimeter Trigger Layer-2

GMT Global Muon Trigger

GT Global Trigger