Global Trigger Logic - Description for emulator

Herbert Bergauer, Babak Rahbaran, Johannes Wittmann
Institute of High Energy Physics (HEPHY)
http://www.hephy.at
http://globaltrigger.hephy.at

November 16, 2021

Revision History

Doc Rev	Description of Change	Revision Date
5.0	Inserted "Calculation of look-up-tables (LUTs) for correlation cuts" (1.4.7).	2021/11/10
4.9	Updated (and renamed) description of "Invariant mass over delta R calculation" (see 1.4.6).	2021/09/14
4.8	Fixed typo in section "Invariant mass calculation for three objects" 1.4.6.	2020/12/03
4.7	Inserted links to VHDL modules.	2020/09/18
4.6	Updated text in section "Correlation conditions" 1.4.13. Description is for v1.10.0 of Global Trigger Logic.	2020/09/17
4.5	Inserted description of "Invariant mass divided by delta R calculation" (see 1.4.6).	2020/09/10
4.4	Fixed typo (unconstrained pt).	2020/09/09
4.3	Inserted text for new muon structure in sections 1.2.2 and 1.4.4.	2020/08/04
4.2	Additional text in section calo calo overlap remover condition module.	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 Calorimeter Overlap Remover conditions and Calo Calo Overlap Remover Correlation conditions.	2020/04/16
3.8	Updated text in sections Calorimeter conditions, Muon conditions and Correlation conditions 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).	2018/08/13
3.6	Updated text in section "Global Trigger Logic" (1) according to firmware version v1.5.0 of gtl_module.vhd	2018/02/21
3.5	Inserted text for "Minimum bias trigger conditions" (1.4.10) and "Towercount condition" (1.4.11). 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.	2016/01/08
3.2	Minor changes in text and updated Figure 3.	2016/01/08
3.1	Changed colour in Figure 4 and updated text for correlation conditions (see section 1.4.13.	2016/01/07

Doc Rev	Description of Change	Revision Date
3.0	Updated Figures 3 and text calo calo correlation condition module.	2015/12/21
2.9	Inserted drawing of VHDL structure of cuts for correlation conditions (see Figure 5).	2015/11/18
2.8	Updated muon η ranges (Table 4) and inserted correlation conditions. Created scheme for conversion of calorimeter η and φ to muon scale for calo muon correlation conditions.	2015/11/17
2.7	Added text in sections Calo comparator module and Muon comparator module.	2015/10/08
2.6	Remaned section "Calorimeter conditions module - version 2" to "Calorimeter conditions module - version 3", section "Muon conditions module" to "Muon conditions module - version 2" and section "Muon comparators module" to "Muon comparators module - version 2".	2015/10/02
2.5	Updated text and tables of η ranges for Calorimeter objects (see 1.4.2).	2015/09/22
2.4	Corrected calculation of muon η step width (see 1.4.4).	2015/09/10
2.3	Edited text in Table 11.	2015/08/28
2.2	Updated definition of η ranges for Calorimeter objects and Muon objects (1.4.2 and 1.4.4).	2015/08/20
2.1	Added section Calo Muon Correlation condition.	2015/08/19
2.0	Added section "Correlation conditions" (1.4.13).	2015/06/19
1.9	Added tables for calorimeter isolation-bits and for muon quality- and isolation-bits definition (3, 6 and 8). Edited section glossary (2) and acronyms.	2015/05/07
1.8	Added text for "Energy sum conditions" (1.4.9) and updated chapters for "Calorimeter conditions" for version 2. Inserted isolation bits for electron/γ and tau objects (1.4.2).	2015/05/06
1.7	Minor changes in sections "Muon data" (1.4.4).	2014/11/06
1.6	Minor changes in sections "Calorimeter conditions definition" and "Muon conditions definition".	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".	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 Calo conditions definition.	2014/02/11
1.0	Document created. Description of Calorimeter conditions.	2013/10/15

Contents

List of Figures

List of Tables

1 Global Trigger Logic

SECTION STILL UNDER CONSTRUCTION

Remark:
This description is for version 1.16.0 of Global Trigger Logic.

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

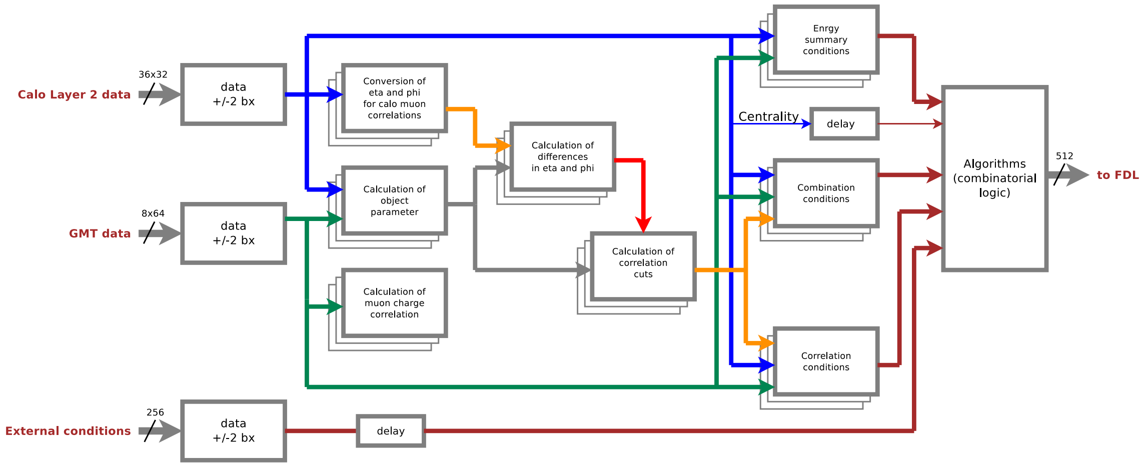

Figure 1: μGTL firmware

1.1 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})

    * "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

Outputs:

- Algorithms

1.2 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).

1.2.1 Calo-Layer2 optical interfaces

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

31 ... 27	26 25	24 ... 17	16 ... 9	8 ... 0
qual/spare	iso	φ	η	E_{T}

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

31 ... 27	26 ... 19	18 ... 11	10 ... 0
iso/qu/sp	φ	η	E_{T}

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

31 ... 27	26 25	24 ... 17	16 ... 9	8 ... 0
qual/spare	iso	φ	η	E_{T}

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

31 28	27 24	23 12	11 0
MBT0HFP	spare	E_{T} [ETTEM]	E_{T} [ET]

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

31 28	27 25	24 12	11 0
MBT0HFM	spare	TOWERCOUNT	E_{T}

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

31 28	27 20	19 12	11 0
MBT1HFP	ASYMET	φ	E_{T}

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

31 28	27 20	19 12	11 0
MBT1HFM	ASYMHT	φ	E_{T}

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

31 28	27 20	19 12	11 0
[CENT3:0]	ASYMETHF	φ	E_{T}

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

31 28	27 20	19 12	11 0
CENT[7:4]	ASYMHTHF	φ	E_{T}

1.2.2 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):

63 62	61	60 53	52 43	42 36	35 34	33 32
imp para	r	unconst.p_{T}	φ (out)	index bits	ch	iso

31 23	22 19	18 10	9 0
η (extrapol.)	qual	p_{T}	φ (extrapol.)

1.3 Implementation in firmware

The firmware of μGTL consists of two main parts:

- A top-of-hierarchy file (`gtl_module.vhd`), which contains the pipeline for $\pm$2bx data, the instantiations of calculators for differences in η and φ, the instantiations of conditions, the instantiations of charge correlation logic of muons and the Algorithms logic for 512 Algorithms, as well as a package file (`gtl_pkg.vhd`) for declarations. 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` (`gtl_module_tpl.vhd`) and `gtl_pkg.vhd` (`gtl_pkg_tpl.vhd`) during simulation and synthesis.
- A set of VHDL-files exists 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.

The latency of μGTL is fixed to 5 bunch-crossings, 2 bunch-crossings for the pipeline of $\pm$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 3).

1.3.1 Top-of-hierarchy module

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

- the pipeline for $\pm$2bx data
- the instantiations of charge correlation logic of muons (generated by VHDL Producer)
- the instantiations of calculators for differences in η and φ (generated by VHDL Producer)
- the instantiations of conditions (generated by VHDL Producer)
- a boolean logic for Algorithms (generated by VHDL Producer)

Listing 1 contains the entity-declaration of the top-of-hierarchy file (`gtl_module.vhd`).

Listing 1: 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;
```

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

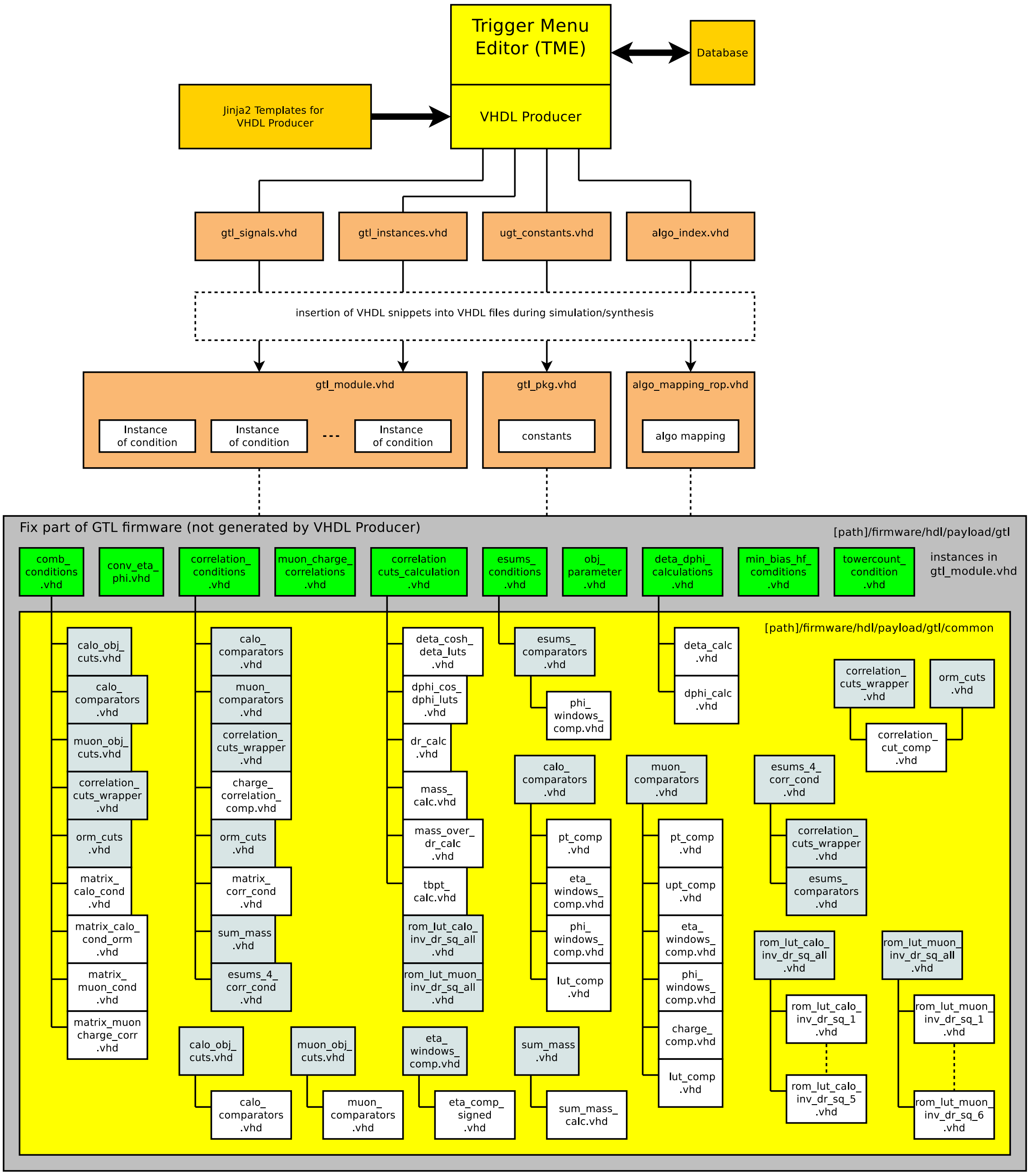

Figure 2: VHDL file generation by VHDL Producer

1.4 μGTL structure

1.4.1 Data $\pm$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 $\pm$2 bunch-crossings), e.g. for Correlation conditions.
See Figure 3 for a scheme of μGTL pipeline structure. The data "data__p__1bx" and "data__p__-2bx" occur 1 respectively 2 bunch-crossings after data for a certain bunch-crossing, therefore we got 2 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.

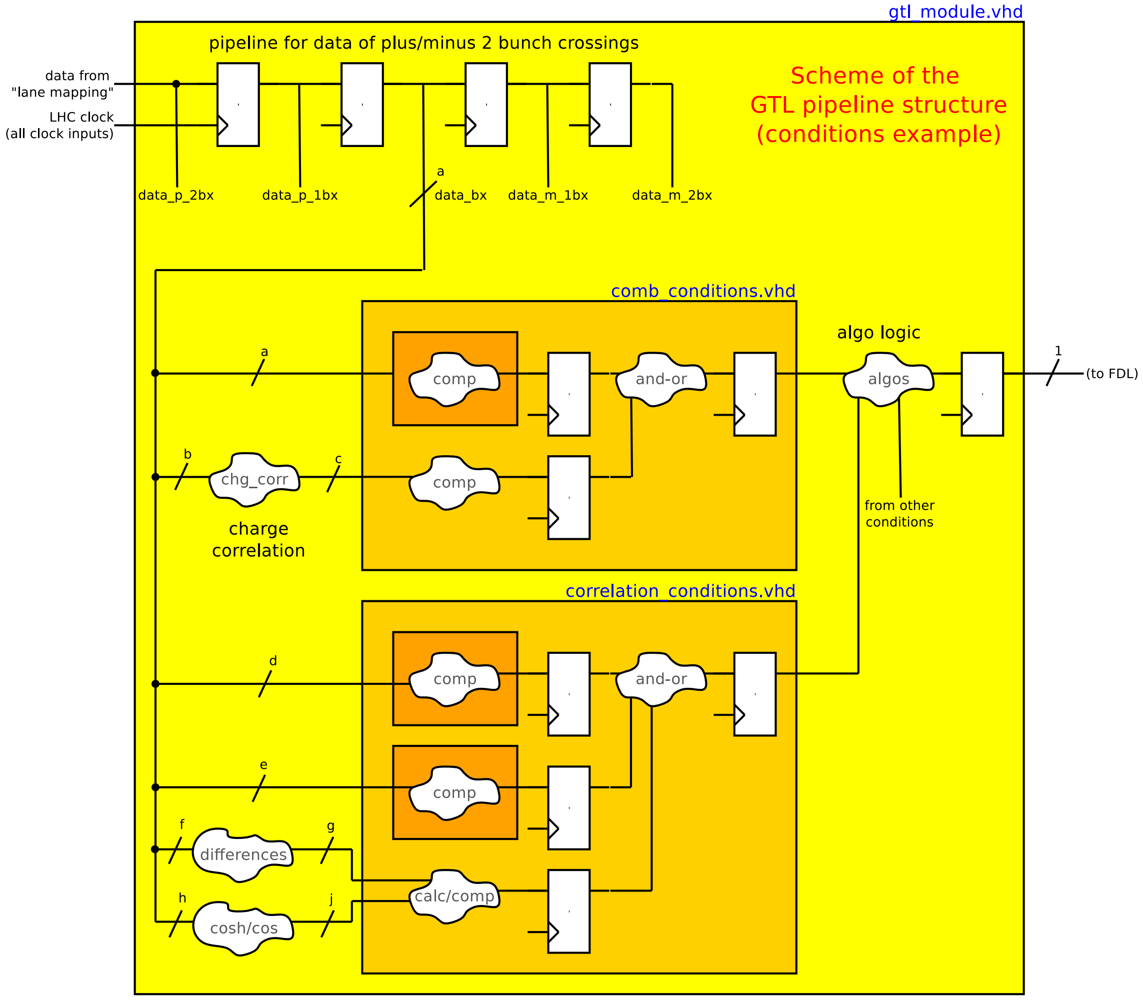

Figure 3: Scheme of μGTL pipeline structure

1.4.2 Definitions of 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 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
- 8 (7+1 sign) bits pseudo-rapidity (η) position, range = -5.0 to 5.0, 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
- 5 bits spare

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

31 … 27	26 … 25	24 … 17	16 … 9	8 … 0
spare	iso	φ	η	p_{T}

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 p_{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 $\approx 2\pi/144$ ($\hat{=}2.5°$), 144 bins (HW index = 0..0x8F), HW index starting at 0° (anti-clockwise)
- 5 bitsspare

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

31 … 27	26 … 19	18 … 11	10 … 0
spare	φ	η	p_{T}

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

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 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
- 8 (7+1 sign) bits pseudo-rapidity (η) position, range = -5.0 to 5.0, 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
- 5 bits spare

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

31 – 27	26 – 25	24 – 17	16 – 9	8 – 0
spare	iso	φ	η	E_{T}

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

The representation of the 8 bits in φ is expected as shown in Table 2.

The representation of the 2 bits for isolation (e/γ and tau) is expected as shown in Table 3.

Table 2: 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 3: Definition of e/γ and tau isolation bits

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

1.4.3 Definitions of Energy sum quantities data

energy sum quantities:
Consists of following quantities (naming convention see 2):

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

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]:

31 28	27 24	23 12	11 0
MBT0HFP	spare	E_{T} [ETTEM]	E_{T} [ET]

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

31 28	27 25	24 12	11 0
MBT0HFM	spare	TOWERCOUNT	E_{T}

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

31 28	27 20	19 12	11 0
MBT1HFP	ASYMET	φ	E_{T}

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

31 28	27 20	19 12	11 0
MBT1HFM	ASYMHT	φ	E_{T}

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

31 ... 28	27 ... 20	19 ... 12	11 ... 0
CENT[3:0]	*ASYMETHF*	φ	E_{T}

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

31 ... 28	27 ... 20	19 ... 12	11 ... 0
CENT[7:4]	*ASYMHTHF*	φ	E_{T}

1.4.4 Definitions of 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 unconstrained p_{T}, for impact parameter, 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):

Table 4: 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

Table 5: 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

63 62	61	60 53	52 43	42 36	35 34	33 32
`imp para`	r	`unconst.`p_{T}	φ *(out)*	`index bits`	`ch`	`iso`

31 23	22 19	18 10	9 0
η *(extrapol.)*	`qual`	p_{T}	φ *(extrapol.)*

The representation of the 9 bits (called "hardware index [HW index]") in η is expected as Two's Complement notation as shown in Table 4.
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.

The representation of the 10 bits in φ is expected as shown in Table 5.

The representation of the 4 bits for quality is expected as shown in Table 6.

The representation of the 2 bits for isolation is expected as shown in Table 8.

Table 6: Definition of muon quality bits

bits [22..19]	definition
0000	quality "level 0"
0001	quality "level 1"
...	...
1110	quality "level 14"
1111	quality "level 15"

Table 7: Definition of muon isolation bits

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

The representation of the 2 bits for impact parameter is expected as shown in Table 8.

Table 8: Definition of muon impact parameter bits

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

1.4.5 Calculation of object cuts

List of object cuts:

- p_{T}
- η
- φ
- isolation
- requested charge
- quality
- unconstrained p_{T}
- impact parameter

1.4.5.1 Object cuts

A comparator between the energy (p_{T}) and a threshold (pt_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 p_{T} threshold, the 'mode-selection' for the p_{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. Number of windows is given for φ.
There are two cases how the limits of one "window"-comparator could be set (see also Figure 4):

- 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).

```
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").

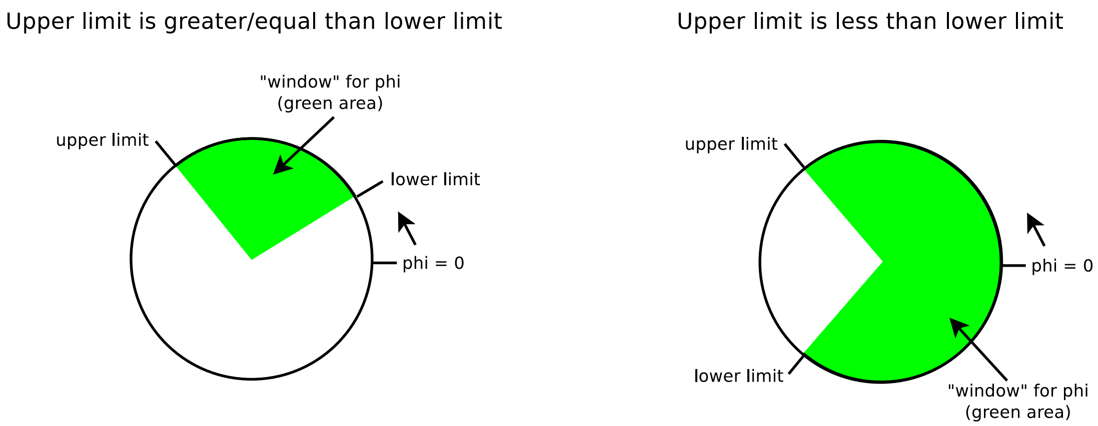

Figure 4: Setting the limits for "window"-comparators for φ

The comparison of isolation (for electron/γ, tau and muon) is done with a LUT (Table 9). [To ignore quality comparison, all bits in the LUT have to be '1'.]

Table 9: LUT contents for isolation comparison

LUT content (4 bits)	isolation (2 bits)	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)

The comparison of impact parameter is done with LUT (Table 10). [To ignore quality comparison, all bits in the LUT have to be ’1’.]

The comparison of quality is done with LUT (Table 11). [To ignore quality comparison, all bits in the LUT have to be ’1’.]

Charge valid and charge sign bits must be equal to the requested charge.

Table 10: LUT contents for impact parameter comparison

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

Table 11: LUT contents for quality comparison of muon objects

LUT content (16 bits)	quality bits (4 bits)	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")

1.4.6 Calculation of correlation cuts

The following cuts are used for two objects correlations:

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

There is one mass cut for correlations with three objects:

- Invariant mass for three objects (MASS).

The generation of look-up-tables (LUTs) for calculations of correlation cuts is described in chapter "Calculation of look-up-tables (LUTs) for correlation cuts" (see 1.4.7).

Calculation of $\Delta\eta$

The calculation of $\Delta\eta$ of two objects is done with formula:

$\Delta\eta$=abs(η1-η2)

where η1 and η2 are represented in signed hardware indices.

Calculation of $\Delta\varphi$

The calculation of $\Delta\varphi$ of two objects is done with formula:

$\Delta\varphi$=abs(φ1-φ2) with ("φ full bin range"-$\Delta\varphi$) when ($\Delta\varphi >$ "φ half bin range").

where φ1 and φ2 are represented in unsigned hardware indices.

ΔR calculation

The calculation of ΔR of two objects is done with formula:

ΔR=$\sqrt{(\eta 1 - \eta 2)^2 + (\varphi 1 - \varphi 2)^2}$.

The calculation of ΔR^2 in VHDL (no square root in VHDL) is done by adding the square of $\Delta\eta$ and $\Delta\varphi$ LUT values.

Invariant mass calculation

The calculation of *invariant mass of two objects* is done with formula:

M=$\sqrt{2pt_1pt_2(\cosh(\eta1-\eta2)-\cos(\varphi1-\varphi2))}$.

The calculation of $\frac{M^2}{2}$ in VHDL (no square root in VHDL) is done by multiplying LUT values of pt1, pt2 and the difference of cosh($\Delta\eta$) and cos($\Delta\varphi$).

Transverse mass calculation

The calculation of *transverse mass of two objects* is done with formula:

M=$\sqrt{2pt_1pt_2(1-\cos(\varphi1-\varphi2))}$.

Calculation similar to "Invariant mass calculation".

Invariant mass over ΔR calculation

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

M=$\sqrt{2pt_1pt_2(\cosh(\eta1-\eta2)-\cos(\varphi1-\varphi2))}$.

ΔR=$\sqrt{(\eta1-\eta2)^2+(\varphi1-\varphi2)^2}$.

The calculation of *invariant mass over ΔR of two objects* is done with $\frac{M^2}{2}\times(1/\Delta R^2)$ (no square root in VHDL).
A direct calculation of $1/\Delta R^2$ is not possible in firmware (VHDL code), therefore the implementation of the calculation is done by LUTs. In the hardware the values of these LUTs are stored in "large" ROMs, which was realized using the Block RAMs (BRAMs) of the Virtex chip.
Due the limited number of available BRAMs there are some restrictions for creating algorithms with *invariant mass over ΔR*:

- Objects must have the same type (e.g.: "muon muon", "eg eg", ...)
- Objects must be of same bx
- Resolution of $\Delta\eta$ and $\Delta\varphi$:
  - Full resolution for calos (max. deta bins=230, max. dphi bins=72)
  - Half resolution only for muons (max. deta bins=226, max. dphi bins=144)
- If $1/\Delta R^2$=0 ($\Delta\eta$=0 and $\Delta\varphi$=0) then correlation cut *invariant mass over ΔR* is true
- The values of LUTs are only valid for current definitions and restrictions. Every change might cause a recalculation of the values and a regeneration of IPs (representing LUTs in BRAMs) in Vivado (firmware generation tool)

The values of LUTs in firmware are listed in coe files of ROMs (created by same scripts mentioned above), currently 5 ROMs for "calo calo" and 6 ROMs for "muon muon" (see `lut_calo_inv_dr_sq_rom1.coe`, etc. and `lut_muon_inv_dr_sq_rom1.coe`, etc.). The addresses of the BRAMs are given by $\Delta\eta$ and $\Delta\varphi$. All ROMs for calos have 4096 addresses, for muons 8192 addresses. The data width of ROMs is different depending on the highest LUT value in ROM. Because of these different data widths, the partitioning of several ROMs was done to save BRAM resources. Currently 873 BRAMs (36kb) are available per Virtex chip. Following numbers of BRAMs (36kb) are needed for:

- "calo calo": 660
- "muon muon": 672

Currently one calculation of *invariant mass over* ΔR of "calo calo" or "muon muon" is possible in one Virtex chip, but one can have some algorithms containing *invariant mass over* ΔR with different thresholds, but with same objects and same bx.

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 take the sum of the three invariant masses of the two-object combinations.

Two-body pt calculation

The calculation of *two-body pt* is done with formula:

pt=$\sqrt{pt_1^2 + pt_2^2 + 2pt_1 pt_2(\cos(\varphi 1)\cos(\varphi 2) + \sin(\varphi 1)\sin(\varphi 2))}$

The calculation of pt^2 in VHDL (no square root in VHDL) using LUTs for pt1, pt2, cos(φ) and sin(φ).

Muon charge correlation

For definition of muon charge, see 1.4.4.
In the muon charge correlation module (`muon_charge_correlations.vhd`), the charge correlations are made for different muon conditions-types. The module is instantiated in the top-of-hierarchy module (`gtl_module.vhd`) and not inside of a muon conditions module. 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 12: 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 13: 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 14: 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

1.4.7 Calculation of look-up-tables (LUTs) for correlation cuts

LUTs are defined as a VHDL "constant" in `gtl_luts_pkg.vhd` (VHDL package file). The values of precision and step size are given by "scale_set" in XML file of a L1 menu.

Overview of precision types for correlation cuts (an example for e.g. e.g. correlation):

- *EG-EG-Delta* relevant for DeltaEta and DeltaPhi LUTs
- *EG-EG-MassPt* relevant for pt and unconstrained pt LUTs (used in mass and two-body pt calculations)
- *EG-EG-Math* relevant for cos(DeltaPhi) and cosh(DeltaEta) LUTs (used in mass calculations)
- *EG-EG-InverseDeltaRMath* relevant for 1/DeltaR LUTs (used in mass over deltaR calculations)
- *EG-EG-TwoBodyPtMath* relevant for cos(Phi) and sin(Phi) LUTs (used in two-body pt calculations)
- *EG-EG-DeltaOverlapRemoval* is obsolete, used EG-EG-Delta (same scales for η and φ)
- *EG-EG-Mass* currently not used
- *EG-EG-TwoBodyPt* is obsolete, used EG-EG-MassPt

Overview of precision names (example for "MassPt"):
EG-EG-MassPt
EG-JET-MassPt
EG-TAU-MassPt
JET-JET-MassPt
JET-TAU-MassPt
EG-ETM-MassPt
JET-ETM-MassPt
TAU-ETM-MassPt
EG-HTM-MassPt
JET-HTM-MassPt
TAU-HTM-MassPt
EG-ETMHF-MassPt
JET-ETMHF-MassPt
TAU-ETMHF-MassPt
EG-MU-MassPt
JET-MU-MassPt
TAU-MU-MassPt
MU-MU-MassPt
MU-ETM-MassPt
MU-HTM-MassPt
MU-ETMHF-MassPt

LUTs for p_{T} and unconstrained p_{T} used in mass and two-body pt calculations

The values of p_{T} or unconstrained p_{T} LUT are calculated by building the half difference of maximum and minimum value of a bin, adding minimum value, rounding at precision position after decimal point and multiplying with $10^{\text{precision}}$ to get integer values.
The address input of the LUT for p_{T} or unconstrained p_{T} is the value of hardware index of p_{T} or unconstrained p_{T}.

The precision values in XML file are given by (an example for e.g. e.g. correlation):
<scale>
<object>PRECISION</object>
<type>EG-EG-MassPt</type>
...
<n_bits>1</n_bits>
</scale>

VHDL names of p_{T} and unconstrained p_{T} LUTs:
EG_PT_LUT (used also for tau)
JET_PT_LUT
ETM_PT_LUT (used also for HT_{miss} and ET^{HF}_{miss})
MU_PT_LUT
MU_UPT_LUT

LUTs for $\Delta\eta$

The values of the LUT are calculated by multiplying $\Delta\eta$ in hardware indices with η step size, rounding at precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.
The address of the LUT is the value of $\Delta\eta$ in hardware indices.

The precision value in XML file is given by (an example for e.g. e.g. correlation):
<scale>
<object>PRECISION</object>
<type>EG-EG-Delta</type>
...
<n_bits>3</n_bits>
</scale>
where <n_bits> is the precision value and <type> represents a precision name.

The η (=$\Delta\eta$) step size in XML file is given by (an example for e.g.):
<scale>
<object>EG</object>
<type>ETA</type>
...
<step>+4.3499999999999997E-02</step>
...

</scale>

VHDL names of $\Delta\eta$ LUTs:
CALO_CALO_DIFF_ETA_LUT
CALO_MU_DIFF_ETA_LUT
MU_MU_DIFF_ETA_LUT

LUTs for $\Delta\varphi$

The values of the LUT are calculated by multiplying $\Delta\varphi$ in hardware indices with φ step size, rounding at precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.
The address of the LUT is the value of $\Delta\varphi$ in hardware indices.

The precision values of $\Delta\varphi$ are identical with $\Delta\eta$.

The φ (=$\Delta\varphi$) step size in XML file is given by (an example for e.g.):
<object>EG</object>
<type>PHI</type>
...
<step>+4.3633231299858237E-02</step>
...
</scale>

VHDL names of $\Delta\varphi$ LUTs:
CALO_CALO_DIFF_PHI_LUT
CALO_MU_DIFF_PHI_LUT
MU_MU_DIFF_PHI_LUT

LUTs for $\cosh(\Delta\eta)$ used in mass calculations

The values in the LUT are calculated by multiplying $\Delta\eta$ in hardware indices with η step size, calculating cosine hyperbolic, rounding at "Math" precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.
The address of the LUT for $\cosh(\Delta\eta)$ is the value of $\Delta\eta$ in hardware indices.
For calo muon correlations one has to use the muon step size.

The precision values in XML file are given by (an example for e.g. e.g. correlation):
<scale>
<object>PRECISION</object>
<type>EG-EG-Math</type>
...
<n_bits>3</n_bits>
</scale>
used for $\cosh(\Delta\eta)$ and $\cos(\Delta\varphi)$.

VHDL names of $\cosh(\Delta\eta)$ LUTs:
CALO_CALO_COSH_DETA_LUT
CALO_MUON_COSH_DETA_LUT
MU_MU_COSH_DETA_LUT

LUTs for $\cos(\Delta\varphi)$ used in mass calculations

The values in the LUT are calculated by multiplying $\Delta\varphi$ in hardware indices with φ step size, calculating cosine, rounding at "Math" precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.
The address of the LUT for $\cos(\Delta\varphi)$ is the value of $\Delta\varphi$ in hardware indices. For calo muon correlations one has to use the muon step size.

VHDL names of $\cos(\Delta\varphi)$ LUTs:
CALO_CALO_COS_DPHI_LUT
CALO_MUON_COS_DPHI_LUT
MU_MU_COS_DPHI_LUT

LUTs for $1/\Delta R^2$ used in mass over deltaR calculations

The calculation of $1/\Delta R^2$ is done by multiplying $\Delta\eta$ in hardware indices with η step size, making the square, doing the same for $\Delta\varphi$, adding the squares, inverting the sum, rounding at "InverseDeltaRMath" precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values. The address of the two-dimensional LUT for $1/\Delta R^2$ consists of values of $\Delta\eta$ and $\Delta\varphi$ in hardware indices.

The precision values in XML file are given by (an example for e.g. e.g. correlation):
<scale>
<object>PRECISION</object>
<type>EG-EG-InverseDeltaRMath</type>
...
<n_bits>5</n_bits>
</scale>

Precision names for "InverseDeltaRMath":
EG-EG-InverseDeltaRMath
JET-JET-InverseDeltaRMath
TAU-TAU-InverseDeltaRMath
MU-MU-InverseDeltaRMath

LUTs for $\cos(\varphi)$ used in two-body pt calculations

The values in the LUT are calculated by building the half difference of maximum and minimum value of a φ bin, adding minimum value, calculating cosine, rounding at "TwoBodyPtMath" precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.

The precision values in XML file are given by (an example for e.g. e.g. correlation):
<scale>
<object>PRECISION</object>
<type>EG-EG-TwoBodyPtMath</type>
...
<n_bits>3</n_bits>
</scale>
used for $\cos(\varphi)$ and $\sin(\varphi)$.

VHDL names of $\cos(\varphi)$ LUTs:
CALO_COS_PHI_LUT
MUON_COS_PHI_LUT

LUTs for $\sin(\varphi)$ used in two-body pt cuts

The values in the LUT are calculated by building the half difference of maximum and minimum value of a φ bin, adding minimum value, calculating sine, rounding at "TwoBodyPtMath" precision position after decimal point and multipling the result with $10^{\text{precision}}$ to get integer values.

VHDL names of $\sin(\varphi)$ LUTs:
CALO_SIN_PHI_LUT
MUON_SIN_PHI_LUT

1.4.8 Combination conditions

1.4.8.1 Combination conditions definition

A condition consists of input data and a set of requirements, which contain the requirements to be complied. The requirements are called "object cuts".

The requirement list contains:
thresholds for p_{T}, ranges for η and φ, LUTs for isolation, LUTs for quality, requsted charges, thresholds for unconstrained p_{T}, a LUT for impact parameter. The condition is complied, if every comparison between object parameters and requirements is valid for the following object cuts (only for requested cuts):

For Calorimeter input data:

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

For Muon input data:

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

There are different types of 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.[1]
- "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 values of the requirements are given by VHDL Producer for every Trigger Menu.
The input data objects have to be of same type and same bunch-crossing.

With "Double objects requirements condition" a correlation cut of "two-body pt" can be required (calorimeter and muon objects).
Additionally charge correlation cuts with "Double objects requirements condition", "Triple objects requirements condition" and "Quad objects requirements condition" of muon objects can be required.

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

Table 15: Explanation of Listing 2

Item	Explanation
`et_ge_mode`	'mode-selection' for the E_{T} comparator. Valid strings are 'true' and 'false' (type is boolean), 'true' means comparator works on greater/equal, 'false' means equal (for tests only)
`obj_type`	valid strings are 'ETT_TYPE', 'HTT_TYPE', 'ETM_TYPE', 'HTM_-TYPE' and 'ETMHF_TYPE'.
`et_threshold`	threshold value for comparison in E_{T}. The size of the std_logic_vector depends on the number of E_{T} bits.
`phi_full_range`	boolean to set full range of φ.
`phi_w1_upper_limits`	"upper limit" of "window"-comparator 1 for φ.
`phi_w1_lower_limits`	"lower limit" of "window"-comparator 1 for φ.
`phi_w2_ignore`	boolean to ignore "window"-comparator 2 for φ.
`phi_w2_upper_limits`	"upper limit" of "window"-comparator 2 for φ.
`phi_w2_lower_limits`	"lower limit" of "window"-comparator 2 for φ.
`clk`	clock input (LHC clock).
`data_i`	input data, structure defined in `obj_type`.
`condition_o`	output of condition (routed to Algorithms logic, see 1.4.15).

1.4.9 Energy sum quantities conditions

1.4.9.1 Energy sum quantities conditions module (including Asymmetry conditions)

For the entity-declaration of `esums_conditions.vhd`, see Listing 2.

Listing 2: Entity declaration of `esums_conditions.vhd`

```
entity esums_conditions is
    generic     (
        et_ge_mode : boolean;
        obj_type : natural := ETT_TYPE; -- ett=0, ht=1, etm=2, htm=3
        et_threshold: std_logic_vector(MAX_ESUMS_TEMPLATES_BITS-1 downto 0);
        phi_full_range : boolean;
        phi_w1_upper_limit: std_logic_vector(MAX_ESUMS_TEMPLATES_BITS-1 downto 0)
            ;
        phi_w1_lower_limit: std_logic_vector(MAX_ESUMS_TEMPLATES_BITS-1 downto 0)
            ;
        phi_w2_ignore : boolean;
        phi_w2_upper_limit: std_logic_vector(MAX_ESUMS_TEMPLATES_BITS-1 downto 0)
            ;
        phi_w2_lower_limit: std_logic_vector(MAX_ESUMS_TEMPLATES_BITS-1 downto 0)
   );
    port(
        clk : in std_logic;
        data_i : in std_logic_vector(MAX_ESUMS_BITS-1 downto 0);
        condition_o : out std_logic
    );
end esums_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. 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.

1.4.10 Minimum bias trigger conditions

Data for Minimum bias trigger are received on the 4 MSBs of 4 frames used for Energy sum quantities (see 1.4.9).

- 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 minimum bias trigger conditions module there is a comparision with a 4-bit threshold (greater-equal [or equal]).

1.4.11 Towercount condition

Data for Towercount trigger (number of firing HCAL towers) are received on frame HT (see 1.4.9) on bits 24..12 (13 bits) of HT data structure.
In towercount condition module there is a comparision with a 13-bit threshold (greater-equal [or equal]).

1.4.12 Centrality condition

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

1.4.13 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 cuts there are correlation cuts for $\Delta\eta$, $\Delta\varphi$, ΔR, mass, mass divided by ΔR and "two-body pt".
The correlation condition of "Invariant mass for three objects" contains one "Triple objects requirement condition" of objects of the same type with one object cut for mass.

List of correlation cuts in 1.4.6.

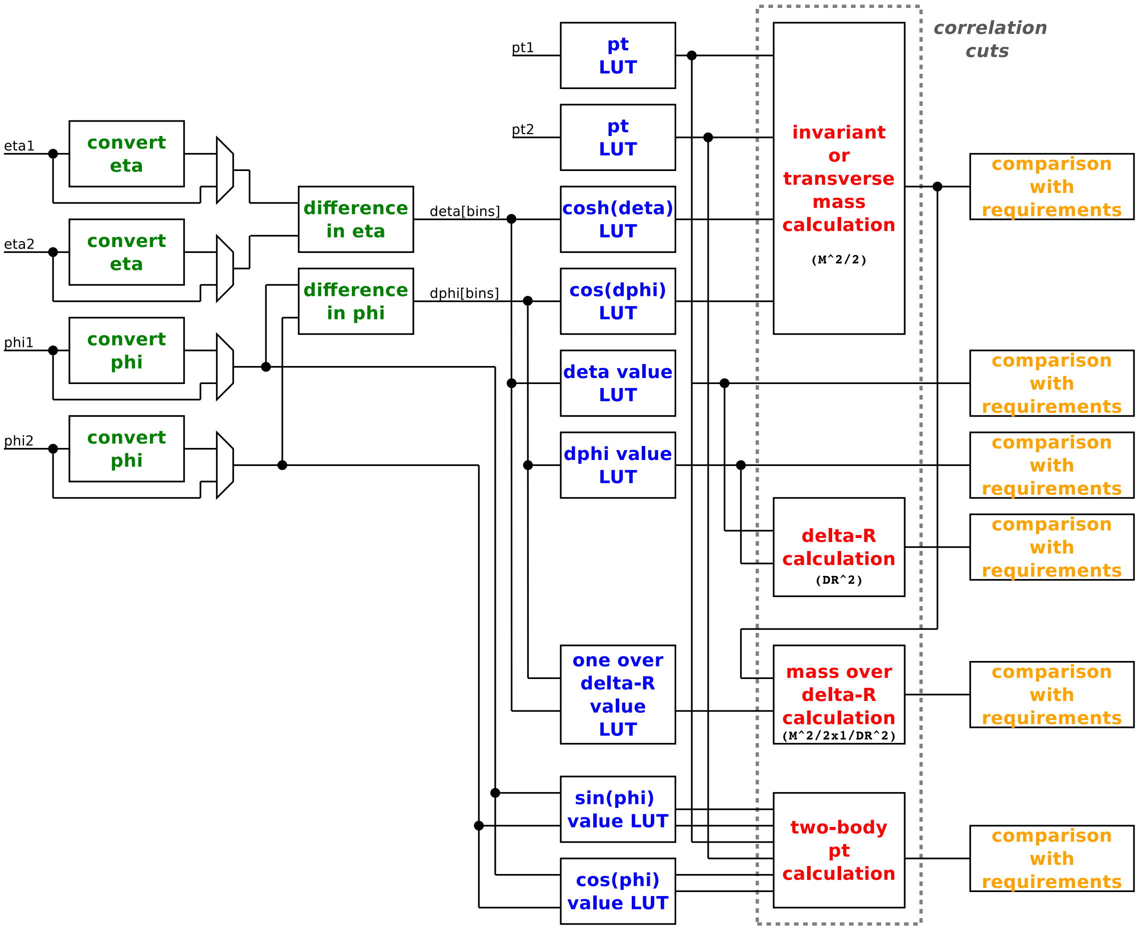

Figure 5: VHDL structure of cuts for correlation conditions

Overview of correlation cuts in conditions

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

- Calo conditions:
  - two-body pt (for double condition)
- Calo conditions overlap removal:

  - $\Delta\eta$ overlap removal
  - $\Delta\varphi$ overlap removal
  - ΔR overlap removal
  - two-body pt (for double condition)

- Muon conditions:
  - charge correlation
  - two-body pt (for double condition)

- Calo calo correlation condition with calo overlap removal:
  - $\Delta\eta$ overlap removal
  - $\Delta\varphi$ overlap removal
  - ΔR overlap removal
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt

- Calo calo correlation condition:
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt

- Calo calo correlation condition for invariant mass divided by ΔR:
  - invariant mass divided by ΔR

- Calo calo correlation condition mass with three objects:
  - invariant mass with three objects

- Calo muon correlation condition:
  - $\Delta\eta$
  - $\Delta\varphi$
  - ΔR
  - invariant mass
  - two-body pt

- Calo esums correlation condition:

  - $\Delta\varphi$
  - transverse mass
  - two-body pt

- Muon muon correlation condition:
  - 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:
  - charge correlation
  - invariant mass divided by ΔR
- Muon muon correlation condition mass with three objects:
  - charge correlation
  - invariant mass with three objects
- Muon esums correlation condition:
  - $\Delta\varphi$
  - transverse mass
  - two-body pt

1.4.13.1 Correlation condition module

As described in section Correlation conditions (1.4.13), correlations of two object types are available. Therefore several correlations (objects 1-objects 2) are possible:

- Correlation condition with calorimeter objects
  electron/γ-electron/γ, electron/γ-jet, electron/γ-tau, jet-jet, jet-tau and tau-tau.
- 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.
- Correlation condition with calorimeter objects and muons objects
  electron/γ-muon, jet-muon and tau-muon.
- Correlation condition 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.

There are two correlations for mass with three objects:

- Correlation condition for mass with three objects with calorimeter objects (same type, same bunch-crossing)

- Correlation condition for mass with three objects with muon objects

1.4.14 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.

1.4.15 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.

2 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