

Data output to the LVL2 Supervisor

Data from the output buffer after RoI coordinates conversion are sent to the LVL2 Supervisor via S-Link.

Synchronisation

The RoI Builder receives from the TTC system EventIDs for accepted by LVL1 events (using individual TTC receiver chip or taking this information from LVL1 timing module, to be defined). These EventIDs are used by control part of the RoI Builder for checking the LVL1 data from CP crates and the CTP, (which should include the EventID embedded in the message format).

Control interface

RoI Builder has an VME interface to load control parameters and to read status information.

Implementation scenario

Before building actual hardware a functional model of the RoI Builder may be created and tested using hardware description language (Verilog or VHDL).

This model may be checked using event data sets consistent with those used for LVL1 and LVL2 simulation (to be defined). This allows to verify the correctness of the algorithm implementation.

Several possible partition schemes of the functional model to the physical components may be worked out. Individual components of the RoI Builder may be implemented in FPGAs using synthesis and timing verification tools. In an (unlikely) case the timing requirements (clock frequency) of the FPGA implementation is not satisfactory a migration to a custom ASIC implementation is a possible solution.

Attention should be made to the possible re-use of individual components and parts of the electron/photon RoI Builder for other RoI Builders (hadron, jets and muons) as well as for the CTP and missing Et LVL1/LVL2 interfaces. The possibility of merging several individual designs should not be forgotten also.

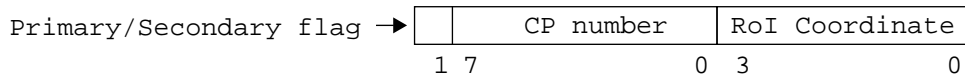
A complete RoI Builder demonstrator board should be built in parallel with development of the test environment, which includes:

- hardware infrastructure similar to one used in LVL1,
- test and debugging software
- possibly additional hardware module(s) - to be defined

References

- [1] I.P.Brawn et al., The level-1 calorimeter trigger system for ATLAS, ATLAS note DAQ-030, 17 January 1995.
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note30/ATLASDAQ30.ps>
- [2] A. Watson, RoI Information from the ATLAS L1 e/γ Trigger, ATLAS note DAQ-073, 1 August 1997.
http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note73/DAQ_No_073.ps.Z

The (N) bits from the CTP decision word for this event are converted into 8-bit word by LUT. The bits of this word are logically ANDed with 8-bit HITS of RoI word and then individual bits are logically ORed. The result of this operation is a 13-bit word.



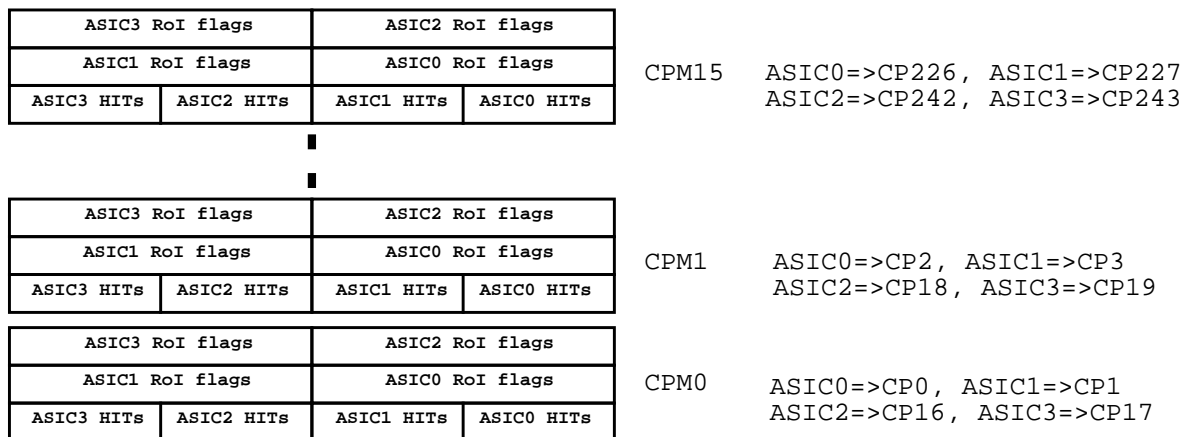
The contention of the LUT are loaded via a control interface. The size of the LUT may be quite small, e.g. 8 inputs x 8 outputs (256 8-bit words) - but the number of input bits from the CTP decision word and the contention of the LUT (conversion rules) still has to be defined.

Data input from LVL1 electron/photon trigger and CTP

The physical implementation of the data link between LVL1 CP crates and the LVL1 CTP from one side and the RoI Builder from the other side is a subject of a LVL1/LVL2 agreement and will be defined later. This includes data format, transfer protocol, physical media, etc.,

Data sorting

Depending of the mapping of CPM crates to the trigger towers map and crate read-out sequence incoming 32-bit data stream from every the CP crate should be sorted and placed in the 16 16-bits RoI Flags Buffers and in the 16 8-bits Trigger Hits Buffers in suitable order. For example for the crate mapping on page 3 (on the left, this is a preferable mapping) LVL1 data may arrive in the following sequence:



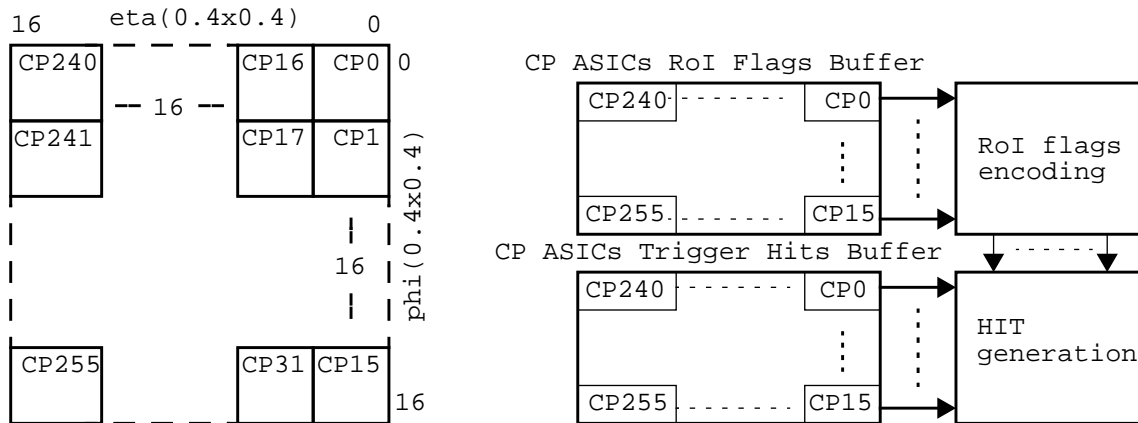
On the right is a possible correspondence between CP ASIC number in each particular CPM and the global numbering scheme for CP ASICs which shows that data sorting has to be done. As far as the actual mapping is not defined this part of the RoI Builder can't be designed.

RoI coordinates conversion

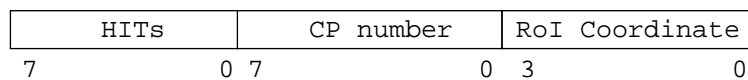
As described before an RoI coordinate conversion may be necessary from LVL1 trigger presentation to the coordinate format suitable for LVL2. This is done by LUT (12 inputs x 12 outputs, 4K 12-bit words), contention is calculated once for the chosen CPMs to trigger towers mapping and is stored in EPROM.

This processor needs for one processing step RoI flags and Trigger Hits from 16 CP ASICs (circle in phi) plus Trigger Hits from the next 16 CP ASICs (next circle in phi). The processing step is completed when all RoI flags are encoded. This step is repeated 16 times (maximum, depending on actual RoI generation area). With serial RoI flags encoding the raw processing time may be estimated as $6.4 \mu\text{s}$ (16×16 steps @ 40 MHz clock) plus some overhead. This is less than $10 \mu\text{s}$ (average L1A inter-arrival time) and may be significantly reduced using priority RoI flags encoding or higher clock frequency (may be questionable).

The input data should be located in the input buffer in the following order to allow efficient algorithm execution (using the following global numbering scheme for CP ASICs):



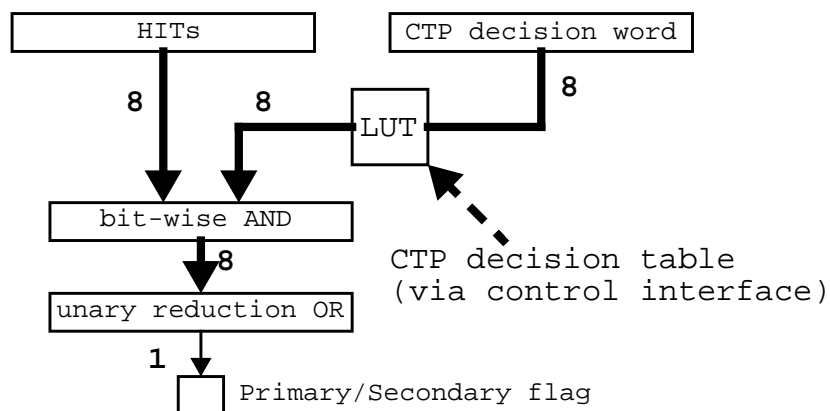
The 16 outputs from the matching processor provide the data for the next processing steps (classifying and RoI coordinate conversion) in the following form:



This data may be read in sequence to provide one (or two) output data stream(s) for classifying and conversion processor(s).

Classifying processor

A classifying processor works on the data after the matching processor as described before.



Software implementation

Raw LVL1 data received from the electron/photon trigger and the CTP are stored in input FIFO buffers. Further data sorting, matching, classifying and conversion may be performed by commercial programmable processor(s). Data to be transmitted to the LVL2 Supervisor (to the Input Router) are stored in output FIFO buffer.

Algorithms may be tested on commercial CPU boards but actual implementation may be done using single ship microprocessor without operating system. The software implementation of the RoI Builder may be a subject of separate note (or will be added to this note later).

Hardware implementation

The LVL1 RoI data processing algorithm may be equally implemented in hardware - the more so as the nature of the RoI data allows parallel data processing on several levels:

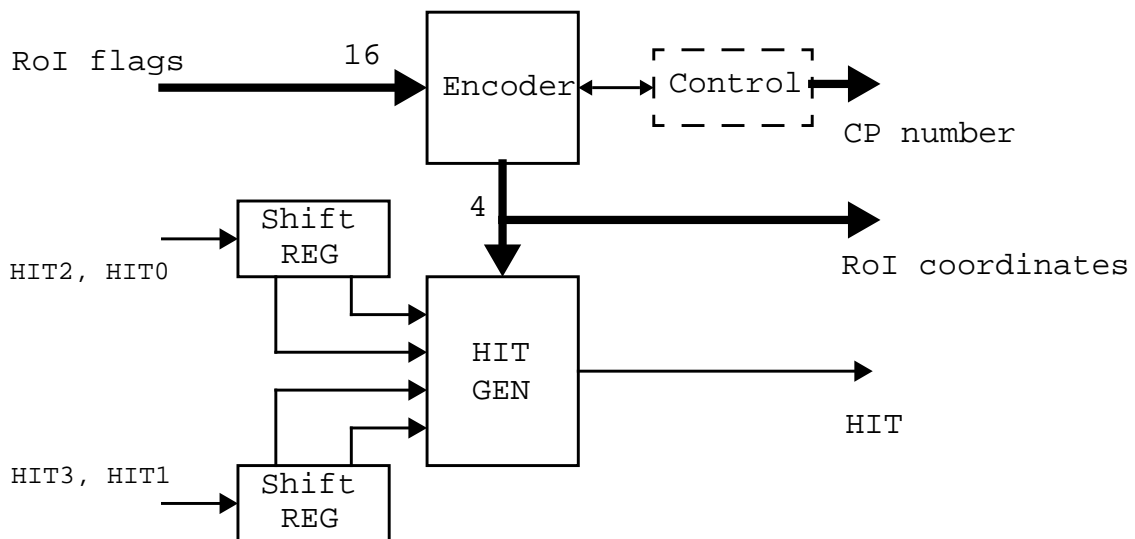
- Each one out of 8 individual 1-bit 16x16 Trigger Hits maps may be processed in parallel.
- Several 2x2 Trigger Hit windows may be processed in parallel.

Usage of non zero-suppressed LVL1 data of fixed size and sequential LVL1 data read-out and storage in input buffers favours pipeline processing, shifting input LVL1 data through a set of hardware (elementary) processors.

4. Possible hardware implementation

Matching processor

The matching algorithm as it is previously described may be implemented in hardware (as shown below) for 2x2 window of one Trigger Hits map (one thresholds set):

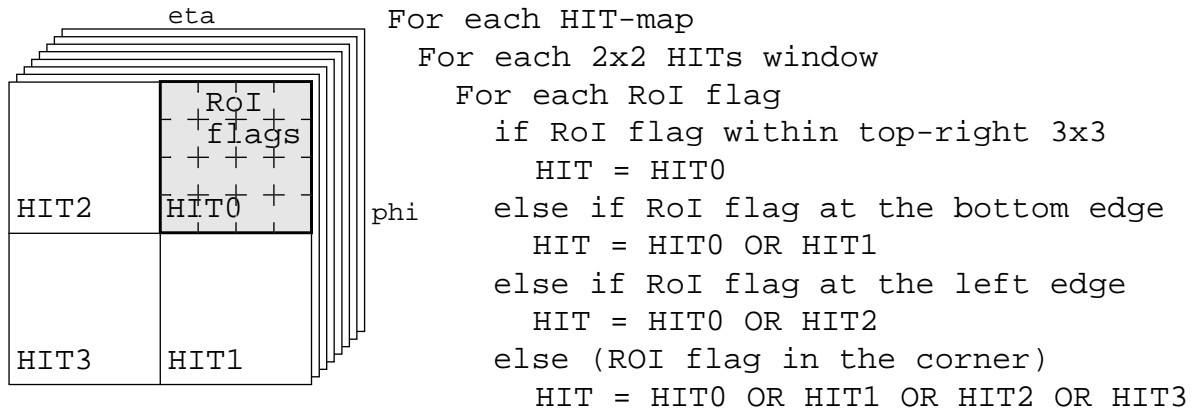


The Trigger Hits from 4 CP ASICs are loaded into shift registers and are assigned by HIT generator to encoded RoI flags. Encoding may be priority (variable number of steps) or serial (fixed number of steps, 16 in total). An output is a list of RoI coordinates with HIT flags.

The shift registers and HIT generator may be replicated for all 8 Trigger Hit maps using the same encoder. This 8-bit elementary processor may be then repeated for 16 overlapping 2x2 Trigger Hit windows in phi forming complete circle (with 16 individual encoders).

RoI information processing algorithm

The first step in LVL1 electron/photon trigger data processing is **matching** 1-bit 64x64 RoI flags map with 8-bit 16x16 Trigger Hits map (which may be considered as eight independent 1-bit 16x16 maps processed in parallel). For RoI flags (encoded) from each CP ASIC one has to look at the Trigger Hits in the 2x2 window (as shown below) in each of the 8 Trigger Hit maps and assigning a HIT flag to the RoI flag according to the algorithm below:



The result of the first step is the RoI flag (encoded) accompanied by 8-bit HIT word.

The second step is **classifying** RoIs to primary and secondary using information from the CTP. To perform this operation a part of CTP decision word, relevant to the electron/photon trigger, should be converted (by mean of Look-Up Table, LUT) to 8-bit word. Bits of this word correspond to 8 threshold sets defined for CP ASIC.

A contention of the Look-Up Table (conversion rules for the CTP decision word) is defined by the CTP decision table. It is generated and reloaded when the CTP decision table is updated.

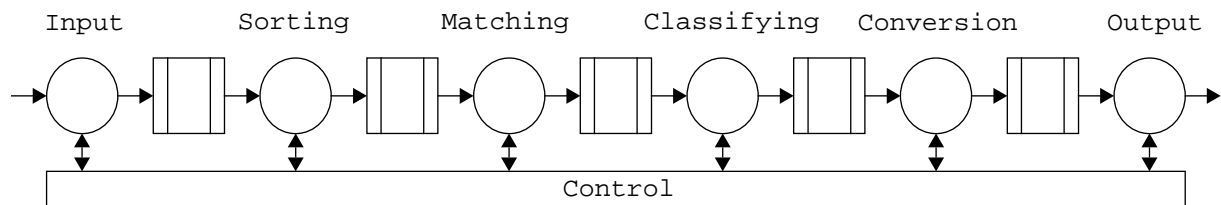
The algorithm of the primary/secondary flag generation consists of:

- bit-wise AND operation between RoI flag 8-bit HIT word and 8-bit LUT word,
- logical OR of individual bits in resulting word from the previous operation.

The result of the second step is the RoI flag accompanied by 1-bit primary/secondary flag.

3. RoI Builder implementation

The RoI Builder may be decomposed to a number of operations (or processes) to be performed on LVL1 raw data in sequence as shown below:

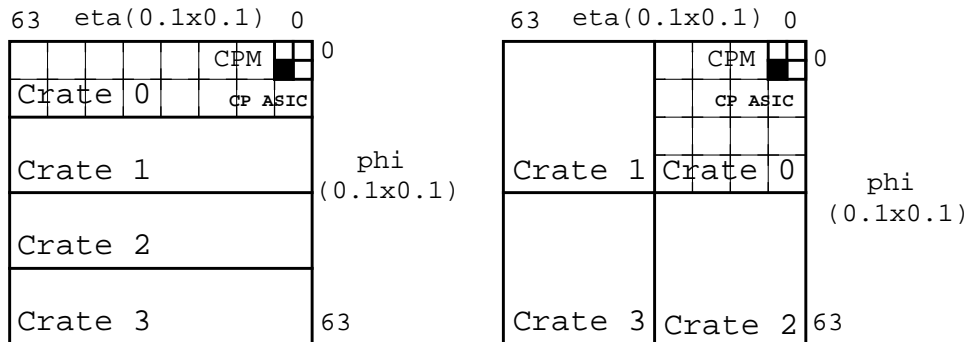


Each process works on the data stored in the input buffer (for this process) and sends a result to the output buffer. Processing time should be less than 10 μ s (average L1A inter-arrival time).

The whole information from 256 CP ASICs may be represented as:

- for the Trigger Hits data (256 8-bit words from CP ASICs) - one 8-bit 16x16 map or eight 1-bit 16x16 maps
- for the RoI flags data (256 16-bit words from CP ASICs) - one 1-bit 64x64 map

Mapping of CPM crates to the trigger towers map (64x64 in eta/phi with 0.1x0.1 granularity) is not yet finally defined (see possible examples).



Depending on the CPMs/trigger towers mapping the Trigger Hits and RoI flags data arrives in the RoI Builder in different order and may need **sorting** to suit the hardware/software algorithm requirements (see later).

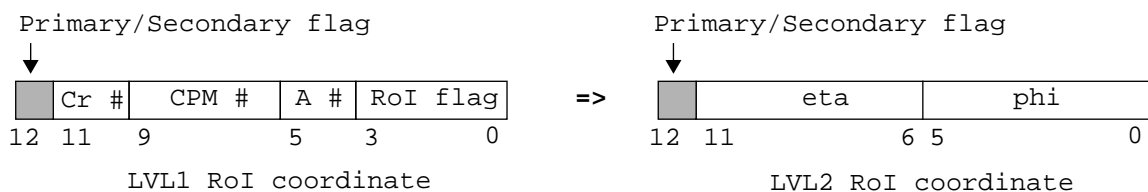
RoI information presentation for LVL2

LVL2 needs from LVL1 an information about RoI coordinates with trigger tower granularity and primary/secondary flag to distinguish RoIs contributed to the LVL1 decision and not.

Processing of the raw LVL1 data includes:

- **matching** Trigger Hit and RoI flag maps in order to assign to the each RoI flag a HIT flag for every possible thresholds set passed
- **classifying** RoIs to primary and secondary using information from the CTP and attaching primary/secondary flag to the RoI coordinate

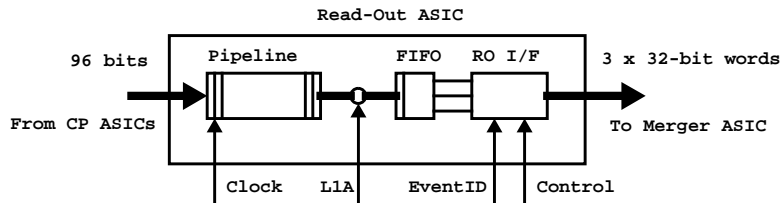
RoI coordinates may need some **conversion** from LVL1 representation - example shows a possible RoI coordinate representation by LVL1 with already encoded RoI flag position inside CP ASIC (left) and coordinate format suitable for LVL2 (right):



CTP information

The CTP information which is necessary for RoI flags classification (primary/secondary) is a part of CTP decision word relevant to the electron/photon trigger (to be defined) for this accepted event and a CTP decision table (this information may be communicated to the RoI Builder via slow control system and updated when needed).

These bits have to be stored in pipelines, similar to ones in the front-end electronics, until a decision is made by the LVL1 Central Trigger Processor (CTP). Trigger Hits and RoI flags data for accepted events (L1A, 10 μ s average inter-arrival time) are transferred then to the electron/ photon RoI Builder.



The transfer could be done without zero suppression packing 96 bits in three 32-bit words (as shown below)

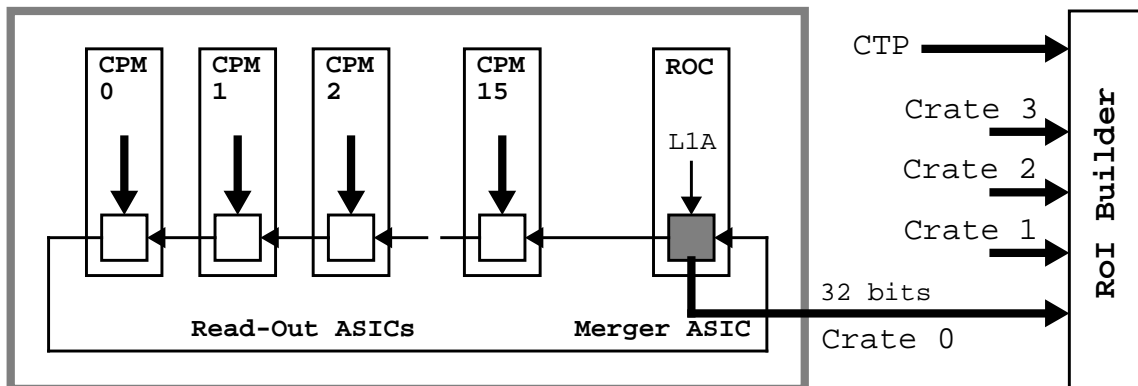
31	24	23	16	15	8	7	0
ASIC3 HITS		ASIC2 HITS		ASIC1 HITS		ASIC0 HITS	
ASIC1 RoI flags				ASIC0 RoI flags			
ASIC3 RoI flags				ASIC2 RoI flags			

or using RoI flags position encoding. Non zero-suppressed information might be suitable for a hardware implementation of the processing algorithm and encoded information - for a software implementation. Read-out of the Trigger Hits and RoI flags data is performed via Read-Out ASIC in CPM controlled by Merger ASIC in Read-Out Controller.

Cluster Processor Crate and electron/photon trigger system layout

The LVL1 calorimeter trigger should cover detector area with $|\eta| \leq 3.2$ and $0 < \phi < 2\pi$ - 64x64 trigger towers map in η/ϕ with 0.1x0.1 granularity (but the electron/photon trigger generates RoI flags in a precision tracking region with $|\eta| \leq 2.5$). An array of 16x16 CP ASICs (each processes 4x4 trigger towers in η/ϕ) cover the entire area. With 4 CP ASICs per CPM (8x8 area in η/ϕ) and 16 CPM per crate whole system may fit in 4 crates.

Upon receiving L1A signal from the Central Trigger Processor Merger ASIC in the crate Read-Out Controller (ROC) starts sequential transfer of the Trigger Hits and RoI flags data from Read-Out ASICs to the RoI Builder. Four crates are read-out in parallel (see figure below). Each crate sends to the RoI Builder a fixed length message containing 50 32-bit words - Header, EventID plus Trigger Hits and RoI flags data (3 words * 16) from CPMs. The transfer time may be estimated as 1.25 μ s for 40 MHz read-out clock and 32-bit parallel transmission.



Electron/photon RoI Builder Demonstrator(s)

<http://www.pa.msu.edu/hep/atlas/rbd.html>

1. Introduction

The ATLAS LVL1 trigger system provides a Region of Interest (RoI) information for the LVL2 trigger system. The RoI Builder is a part of the LVL1 trigger system which assembles and organizes the information from different parts of the LVL1 trigger and sent it to the LVL2 Supervisor via number of S-Links.

The electron/photon RoI Builder receives a raw RoI information generated in the electron/photon part of the LVL1 calorimeter trigger [1] and after processing sends it to the LVL2 Supervisor. The electron/photon RoI information generation and processing algorithm is described in the ATLAS note DAQ-073 [2] which is an essential reading beforehand.

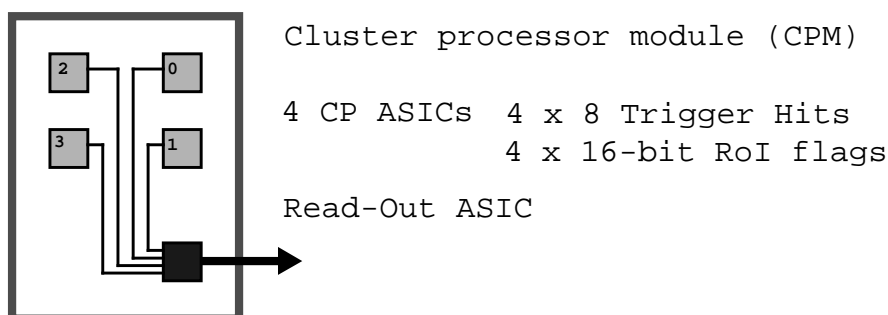
This note describes a possible hardware (and software?) implementation of the electron/photon RoI Builder demonstrator.

2. LVL1 electron/photon Trigger and RoI information

Cluster Processor ASIC and Cluster Processor Module

The LVL1 electron/photon trigger and RoI generation algorithm is based upon 4x4 array (trigger window) of 0.1x0.1 trigger towers slide by 1-tower step in either eta or phi and results in 1 bit trigger decision and 1 bit RoI decision with 0.1x0.1 granularity [2].

16 overlapping trigger windows (4x4 area in eta/phi) are processed within a single Cluster Processor (CP) ASIC. A Cluster Processor Module (CPM) contains 4 such ASICs and processes 64 trigger windows (8x8 area in eta/phi).



Each CP ASIC generates one Trigger Hit (an logical OR of 16 internal trigger decisions) per programmable threshold sets (cluster, electromagnetic and hadronic isolation, 8 threshold sets in total) with 0.4x0.4 granularity and 4x4 RoI flags bit-map with 0.1x0.1 granularity (16 bits in total). Four CP ASICs in CPM generate 96 bits in total (for every bunch crossing).