ATLAS Level-1 Calorimeter Trigger
Technical Information Note TIN 4/2005
Version 0.1, 8-Sep-2005
EDMS Document number

Cable Database

Draft 0.1

C. N. P. Gee

Summary

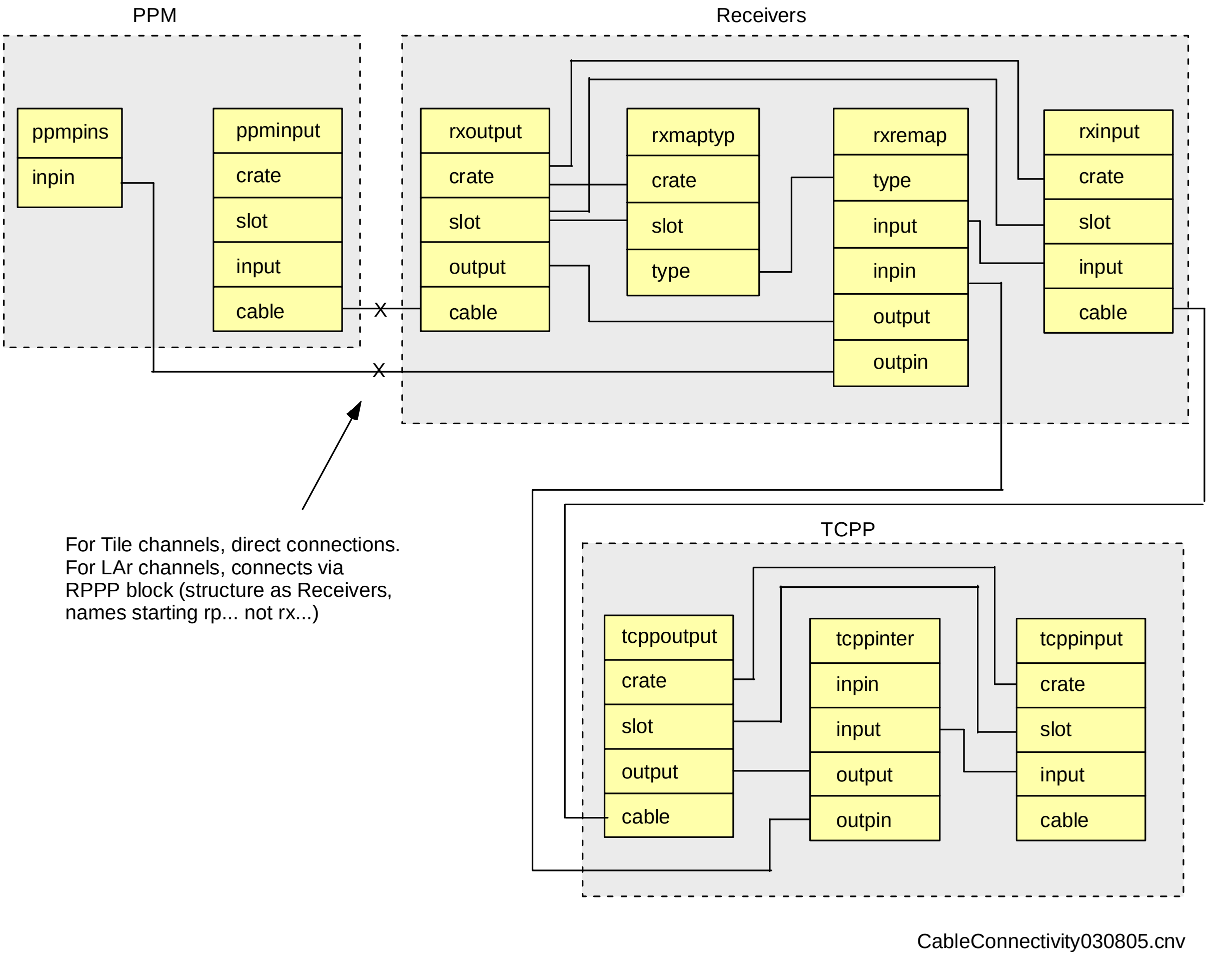
This short node describes the current status of work on the cabling database.

1 Overview

This note briefly describes a database containing the connectivity between the calorimeters, patch panels, receivers and Preprocessors in the Level-1 Calorimeter Trigger. The intention is that it should be possible to find out which calorimeter tower is connected to a PPM input, which receiver gain has to be adjusted, and so on. A key point is that the database should allow the frequent changes of connectivity expected during trigger installation.

2 Implementation

The database comprises the nine tables illustrated below plus four more for the RPPPs. The only component not yet incorporated is the Summing Patch Panels.



For Tile channels, direct connections.
For LAr channels, connects via
RPPP block (structure as Receivers,
names starting rp... not rx...)

3 Data Insertion

All the data in the Cable Mapping document has been converted to PERL scripts. These generate data files of comma-separated values (.csv files) which can be read into MYSQL or ORACLE.

For MYSQL, a set of SQL macros reads the csv files into temporary tables which are then manipulated to produce the normalised structure above. A single shell script can be used to perform the entire loading operation.

For ORACLE, the procedure is more messy, as three stages are needed: first, the temporary tables are create with SQLPLUS, then the ORACLE data loader is used to import the csv files, then a second set of SQLPLUS macros performs the extraction and normalisation. This is not yet far advanced. A single shell script can probably do the complete job.

4 Queries

A few specimen queries have been written. An example is reproduced below which extracts a summary of all the connections for the Tilecal. This summary can trivially be written into a single table or a flat file. The query runs in about a second, but takes about six minutes if the tables aren’t indexed.

```
select   ppminput.crate as pcrt
       , ppminput.slot as pslt
       , ppminput.input as pconn
       , ppminput.cable as pcbl
       , ppmpins.inpin as pin

-- receiver outputs

       , rxoutput.crate as rocrt
       , rxoutput.slot as roslt
       , rxoutput.output as "ro/p"
--       ,  rxoutput.cable as rocbl

-- remap board details

       , rxmaptyp.type as rmtyp
       , rxremap.inconn as rmicon
       , rxremap.outpin as rmopin

-- receiver inputs

       , rxinput.input as 'rii/p'
       , rxinput.cable as rxicbl

-- tcpp outputs

       , tcppoutput.crate as tocrt
       , tcppoutput.slot as toslt
       , tcppoutput.output as 'too/p'

-- tcpp internal cable merging

       , tcppinter.outpin txopin
       , tcppinter.input as txin
       , tcppinter.inpin as txipin

-- tcpp inputs

       , tcppinput.input as "tii/p"
       , tcppinput.cable as tpcbl

       from
           ppmpins, ppminput , rxoutput , rxmaptyp, rxremap, rxinput
         , tcppoutput
         , tcppinter
         , tcppinput
       where
         ppminput.crate=6 and
         ppminput.slot=1
         and ppminput.input=1
--       and ppmpins.inpin=1

-- connect ppm input cable

       and ppminput.cable=rxoutput.cable

-- choose receiver remap type

       and rxmaptyp.crate=rxoutput.crate
       and rxmaptyp.slot=rxoutput.slot

-- connect remap outputs

       and rxremap.type=rxmaptyp.type
       and rxremap.outconn = rxoutput.output
       and rxremap.outpin = ppmpins.inpin

-- connect receiver inputs to remap inputs

       and rxinput.crate=rxoutput.crate
       and rxinput.slot=rxoutput.slot
       and rxinput.input=rxremap.inconn

-- connect tcpp output cable to receiver input cable

       and tcppoutput.cable = rxinput.cable

-- connect tcpp internals

       and tcppinter.outpin=rxremap.inpin
       and tcppinter.output=tcppoutput.output

-- connect tcpp inputs

       and tcppinput.crate=tcppoutput.crate
       and tcppinput.slot=tcppoutput.slot
       and tcppinput.input=tcppinter.input

       order by
       ppminput.crate, ppminput.slot, ppminput.input,rxremap.outpin
```

5 Use for Installation

As cables will be moved very often, a single shell script can probably be used to update the database, rerun the queries, and generate required output (COOL and/or flat files).