

Table of Contents

Principal Reviewer.....	1
Code Location.....	2
Comments.....	3
Proposed Event Classes.....	3
Notes for Developers.....	4
Review comments.....	5
Multiple Return Statements.....	5
.If statements.....	5
.Messaging.....	6

Principal Reviewer

Chris Jones

Code Location

Package	Project	Functionality	Responsible
Det/STDet	LHCb	The detector element package	Jianchun Wang
Event/DigiEvent	LHCb	Event model classes related to Digitization	Jianchun Wang
Event/MCEvent	LHCb	LHCb Event model. MC truth classes	Jianchun Wang
Kernel/LHCbKernel	LHCb	LHCb specific Kernel classes	Jianchun Wang
ST/STKernel	LHCb	LHCb ST specific Kernel classes	Jianchun Wang
ST/STDAQ	Lbcom	The raw bank encoder/decoder	Jianchun Wang
ST/STMonitors	Lbcom	Monitoring NZS and ZS data	Jianchun Wang
ST/STDigiAlgorithms	Boole	Algorithms for the ST digitization	Jianchun Wang

Comments

- Average radiation length of $4.83\% X_0$ in region $2 < \eta < 5$
- Light tight TT box made of 400 μ m thick (Kevlar + graphite paper + epoxy)
- Kapton (thickness = 100 μ m) surrounding the beam pipe to replace the current jackets
- Smaller beam pipe hole @ 28mm (to check the maximum coverage gain)
- New TT detector hierarchy and addressing (more readout sectors to address)
- Sensors are placed at front & back of module alternatively to provide sufficient overlaps
- Sensors with semi-circular cuts surrounding the beam pipe for maximum inner acceptance
- Sensor positions are optimized to have overlap (minimal) & full coverage
- Finer pitch in the bending direction of the central region

The UT related code can be found at <svn+ssh://svn.cern.ch/repos/lhcb/UTReview>

The `test/CommandToRun` file lists the commands to setup the environment.

The `test/Modification.list` includes the files that were added or modified, as well as a few comments.

Proposed Event Classes

Class	Class ID	Description
aClass	anID	aDescription

Notes for Developers

Please make the reviewer aware of the code location and also provide them with an explanation of how the code works and fits into the LHCb software framew

Review comments

Multiple Return Statements

Avoid multiple return statements where possible. Particularly in small inline methods. The inline directive is only ever a suggestion to the compiler, and it is free to ignore it. Having a method with multiple return statements is essentially a guarantee it will be ignored. The ? operator is useful to help write efficient code with a single return statement, so should be used (Yes, ignore code convention R60). e.g. replace

```
if (isUT()) return (unsigned int)((m_channelID & sector1Mask) >>> sector1Bits);
else       return (unsigned int)((m_channelID & sectorMask) >>> sectorBits);
```

with

```
return ( isUT() ?
        (unsigned int)((m_channelID & sector1Mask) >>> sector1Bits) :
        (unsigned int)((m_channelID & sectorMask) >>> sectorBits) );
```

If statements

Support for UT added in several places by adding a lot of if statements.

```
template <class PBASE>
inline std::string ST::CommonBase<PBASE>::uniqueLayer(const LHCB::STChannelID& chan) const{
    if ( m_detType == "TT" )
        return LHCB::TTNames().UniqueLayerToString(chan) ;
    else if ( m_detType == "IT" )
        return LHCB::ITNames().UniqueLayerToString(chan) ;
    else
        return LHCB::UTNames().UniqueLayerToString(chan) ;
}
```

Make sure this sort of pattern doesn't become too widespread. If it does, consider some other method to specialise for UT, TT and IT (inheritance, templation for example). Above also uses a slow string comparison, which should be avoided.

Hardcoded maps ? Not specific to UT, also there for TT and IT. Shouldn't these really be in the database ?

```
const STBoardMapping::Map& STBoardMapping::ITNumberToSourceIDMap() {
    static Map s_map;
    if (s_map.empty()){
        s_map
            = boost::assign::map_list_of(1, 0)(2, 1)(3, 2)(4, 3)(5, 4)(6, 5)(7, 6)
            (8, 7)(9, 8)(10, 9)(11, 10)(12, 11)(13, 12)(14, 13)(15, 32)(16, 33)
            (17, 34)(18, 35)(19, 36)(20, 37)(21, 38)(22, 39)(23, 40)(24, 41)(25, 42)
            (26, 43)(27, 44)(28, 45)(29, 64)(30, 65)(31, 66)(32, 67)(33, 68)(34, 69)
            (35, 70)(36, 71)(37, 72)(38, 73)(39, 74)(40, 75)(41, 76)(42, 77);
    }
    return s_map;
}
```

Messaging

Don't use `printf` or similar methods. Use Gaudi messaging where possible (`info()` etc.) or failing that `std::cout`.

UpgradeSoftwareReviewTT < LHCb < TWiki

RecSummary does not need explicit fields for both the number of hits in UT and TT, as only one of these can be active at once ! The current TT field should be used for both (and if required renamed).

This topic: LHCb > UpgradeSoftwareReviewTT

Topic revision: r9 - 2012-08-21 - ChristopherRJones



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback