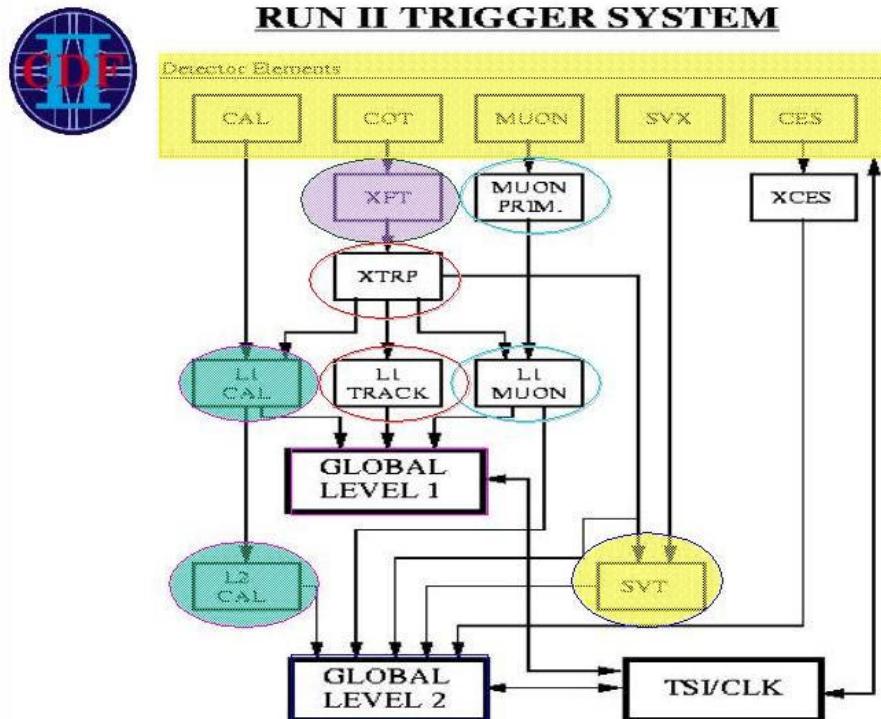


TRGSim++ status

TRGSim++ is a set of (C++) packages emulating the various trigger levels decision steps (CDF trigger is fully digital)

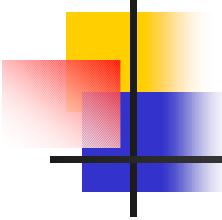


Trigger decision steps: A_C++ modules, organized in packages:

CalTrigger -> CalTriggerExe
MuonTrigger -> MuonTriggerExe
XFTSim -> XFTTest (tbin)
svtsim -> svtsimtest (tbin)
XTRPSim -> XTRPSimExe(tbin)
L2GlobalTrigger ->L2Sim
L1GlobalTrigger ->FredSim
TriggerMods -> TRGSim++
TriggerObjects -> trigger banks

TRGSim++ modules run off detector raw data and produce emulated trigger data identical to real hardware data.

It also runs off COTQ and CalData. No SIXQ



Documentation and examples

<http://ncdf70.fnal.gov:8001/trgsim/trgsim.html>

General Information;

Information on the code : what it does and how to run (link
to most recent tcl's).

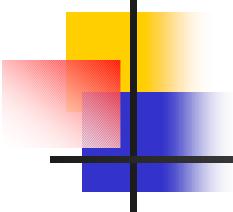
List of changes in: <http://ncdf70.fnal.gov:8001/trgsim/TAGS.html>

To run TRGSim++ ALWAYS go and look at the following tcl:

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++.tcl

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++_MC.tcl

This are the only tcl we constantly update for users reference

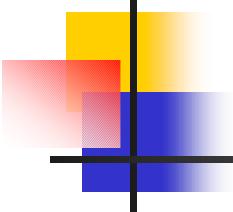


Status: CalTrigger

- Trigger Tower energies (TC2D – first word)
- L1 DIRAC Triggers (TC1D)
 - http://ncdf70.fnal.gov:8001/trgsim/dirac_trigger.html
- L2 clustering and Iso sums (TC2D – second word)
- database access for trigger definition - real data
 - TriggerDB DOWNLOADS Table
 - possibility to run on simulated run with conditions from real run: in talk-to

```
use_software_CAL_banks set t  
use_xtrp set t  
use_hardware_xtrd set f  
use_hardware_L1 set f  
run_Number set 138233
```





CalTrigger input/output

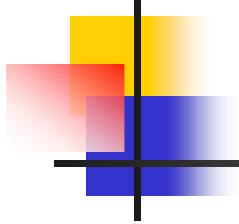
CalTrigger reads in input the calorimeter D-banks: [CEMD](#), [CHAD](#), [PEMD](#), [PHAD](#), [WHAD](#). It alternatively reads CalData and puffs it into D-banks, via a call to CalorimetryModule.

It also gets the [track input from XTRD_StorableBank](#).

There are 2 options to get the track input controlled by the parameter "standalone" in the talk-to: from real data (standalone = 1) or from a simulated XTRD (standalone = 0).

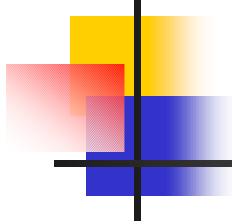
CalTrigger produces in output the following banks:

- [TC2D](#) with the list of trigger tower energies and clustering summary word for each tower (description "Simulated Trigger Bank").
- [TC1D](#) with the results of the DIRAC triggers: description "Simulated Trigger Bank".
- [TL1D](#) with the PreFred bits for CalTrigger and SumEt: description "Simulated Trigger Bank".
- [TL2D](#) where the only block filled is the clustering block: description "Calorimeter TL2D Bank".



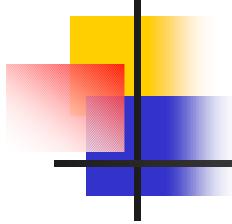
Status XFTSim

- XFT tracks :
 - XFLD and XFFD diagnostic banks
 - XTRD with tracks parameter
- DB access implemented
 - real run conditions applied to simulated run:
- For the Road and Mask files one can set the file types. So one can mimic exactly what was run. This is not automated so one has to do it by hand.
- Deadwires mapping from db not there yet.

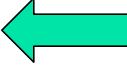


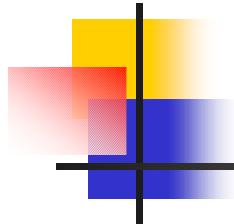
XFT input/output

- Input:
 - COTD hit information (void XFTSim::getInputData())
 - COTQ if COTD is not found in data stream
- Output:
 - XFFD - XFT Finder bank
 - XFLD - XFT Linker bank



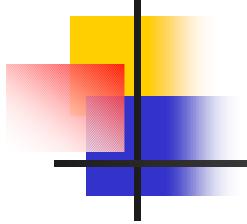
Status : XTRPSim

- XTRP tracks:
 - XTRD
- database access for trigger definition - real data
 - TriggerDB DOWNLOADS Table : XFT_PT
 - possibility to run on simulated run with conditions from real run: in talk-to
 - standalone set f
 - run_Number set 138233 
- Also it possible to set the muon pt thresholds by hand



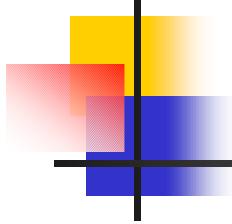
XTRPSim

- Input : XFLD (real or simulated controlled via talk-to parameter “*standalone*”)
- output XTRD bank:
 - track data: 2 blocks:
 - first block: 12 cards (2 wedges per card)
 - Track data:
 - ϕ
 - p_T
 - Isolation bit
 - short track bit
 - XTRP/Calorimeter bus
 - 2nd block: track trigger bits word



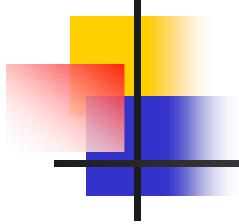
Status: MuonTrigger

- Muon primitives:
 - TCMD
 - Xtrapolation with XTRP done
 - database access not implemented
 - stubs thresholds hardwired - stable so far
- Talk-to added to simulate only selected detector:
 - Everything is ON by default



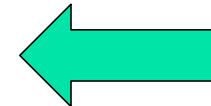
MuonTrigger input/output

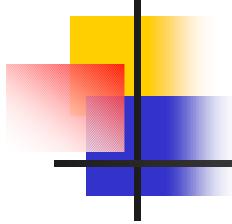
- Input:
 - CMUD
 - CMXD
 - CMPD
 - XTRPMatchBoxData
- Output:
 - TCMD



Status: svtsim

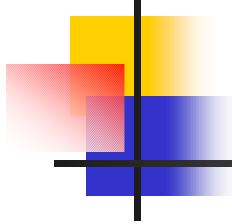
- SVT tracks:
 - SVTD
- database access implemented
 - Possibility to emulate the conditions of a given run module talk svtsim
 - DBRunNumber set 138233
 - writeBank set true
 - useDB set true
 - exit





Status: svtsim

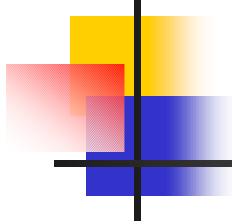
- SVT tracks:
 - SVTD
- database access implemented
- Input SIXD and XFLD



Status: L1GlobalTrigger

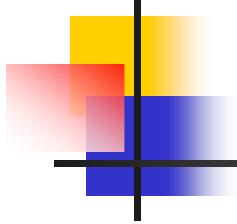
- FredSim: L1 bits
 - TFRD
- Input : TL1D preFred bits
- db access implemented
- database access for trigger definition - real data
 - TriggerDB DOWNLOADS Table : bits mapping
 - possibility to run on simulated run with conditions from real run: in talk-to

```
use_simTL1D set t  
run_Number set 138233
```



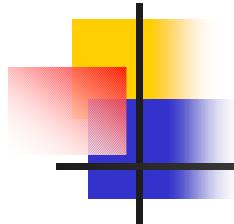
Status: L2GlobalTrigger

- L2Sim: L2 bits + L2 quantities
 - TL2D simulated fills only:
 - cluster block -P6
 - Iso sums block –P5
 - SVT block – P4
 - RECES block –P7
 - no db access
 - Work is in progress to port the online L2code into L2GlobalTrigger ~ end of August
 - Db access will be implemented
 - SVT+lepton trigger code –L. Vacavant, a. Cerri



Trigger banks

- TriggerObjects
 - trigger banks as in [cdf_note4152](#)
 - simple accessors (get_word (int, int, int))
 - named accessors (depending on the bank)
 - look at the code browser
 - examples: TriggerObjects/src/Writer.cc,
TriggerObjects/src/Reader.cc, TriggerObjects/ntuple
 - Most of the trigger banks are in evtNtuple:
 - trigger towers energies, MET and SumEt
 - XTRP tracks
 - SVT track
 - L1 trigger bits (prescaled and uprescaled)
- <http://ncdf70.fnal.gov:8001/talks/eN/eN.html>



TRGSim++

- Built as part of all the integration releases and frozen releases.
- Built nightly as part of development

To run TRGSim++ ALWAYS use the following tcl:

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++.tcl

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++_MC.tcl

Try to use the latest integration release, if you can, otherwise check:

<http://ncdf70.fnal.gov:8001/trigsim/TAGS.html>