

Status of the Trigger Performance during Spring 2017: Part I

A. Somov, Jefferson Lab
March 29, 2017

Analysis Overview

- Study hardware performance in spring 2016
 - Trigger emulation
 - Yield of mesons (ρ , π , ω) for different trigger types
 - Trigger efficiency
- Simulation of L1 trigger

Main Triggers in Spring 2017

Bit 1	$25 E_{\text{FCAL}} + E_{\text{BCAL}} > 45000$ $(E_{\text{FCAL}} + 0.5 E_{\text{BCAL}} > 0.5)$	47 kHz
Bit 3	$E_{\text{BCAL}} > 54000$	13 kHz
Bit 4	PS	2.24 kHz
Bit 8	$(E_{\text{FCAL}} + E_{\text{BCAL}}) \& \text{ST}$	21 kHz
Bit 9	TAGH & ST	339 kHz (prescaling 65)

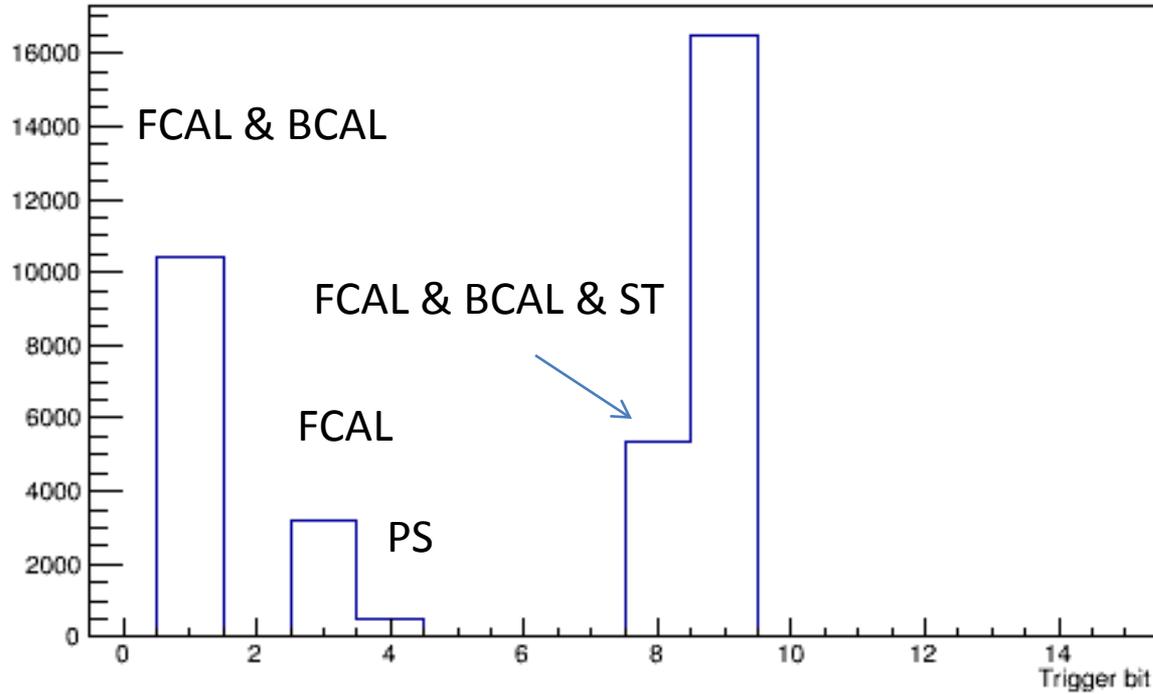
**225 nA, $3 \cdot 10^{-4} X_0$ Al, 5 mm collimator, 75 μm Be,
DAQ rate: 60 kHz, Live time 95 %**

More stringent threshold on BCAL energy

- 30 % smaller rate compared to runs in Spring 2016

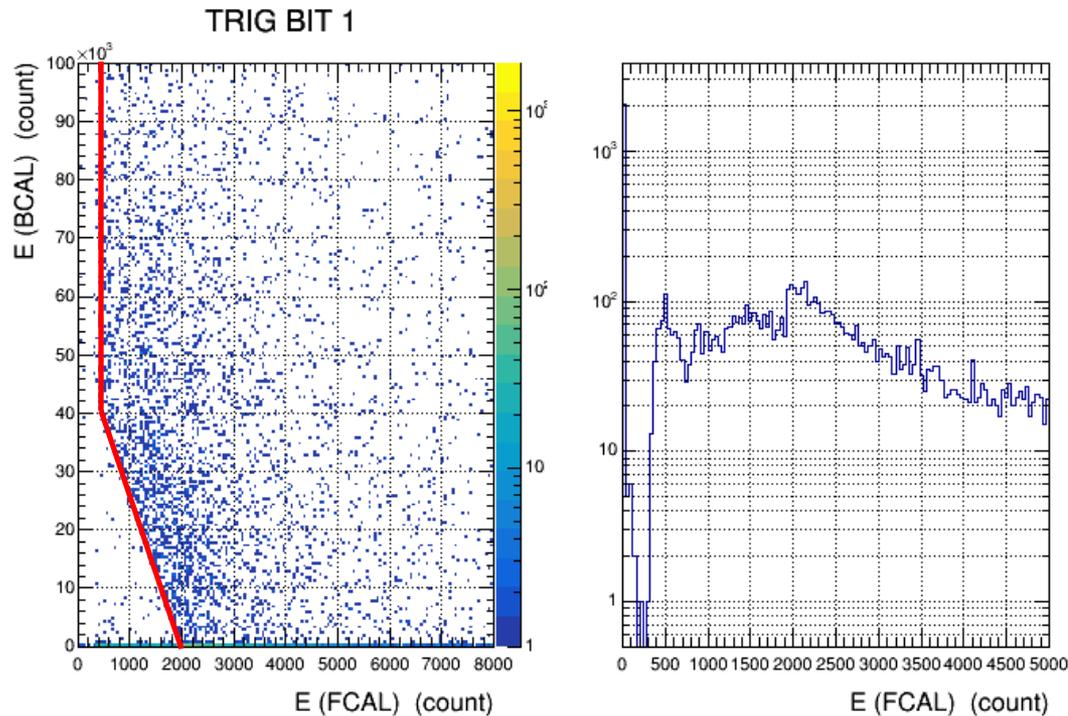
FCAL & BCAL

TAGH & ST



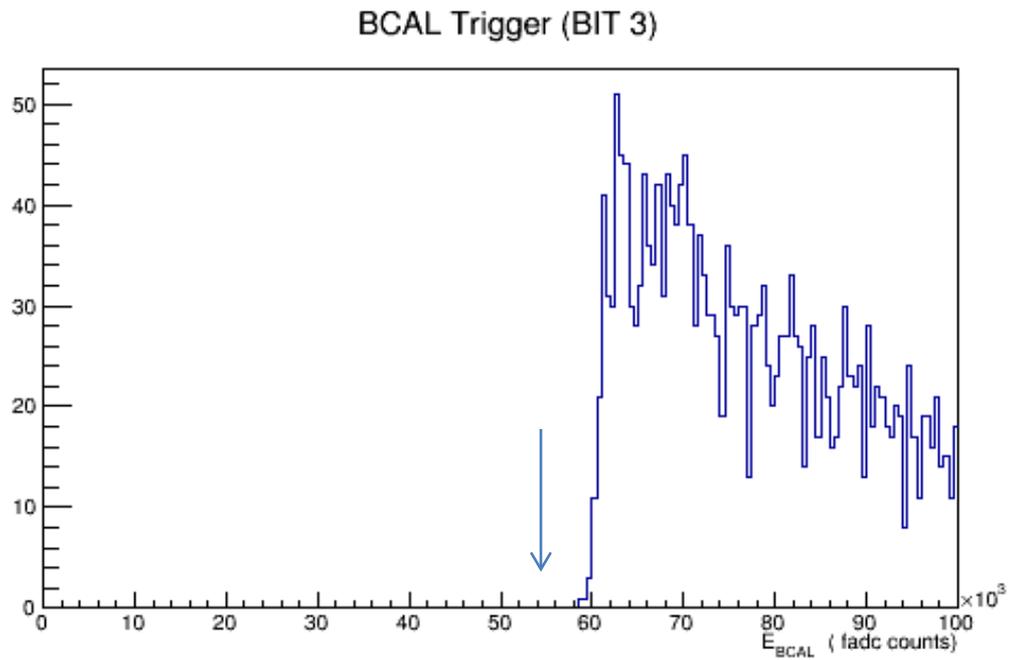
FCAL & BCAL

Check trigger performance using recorded data (reproduce thresholds)

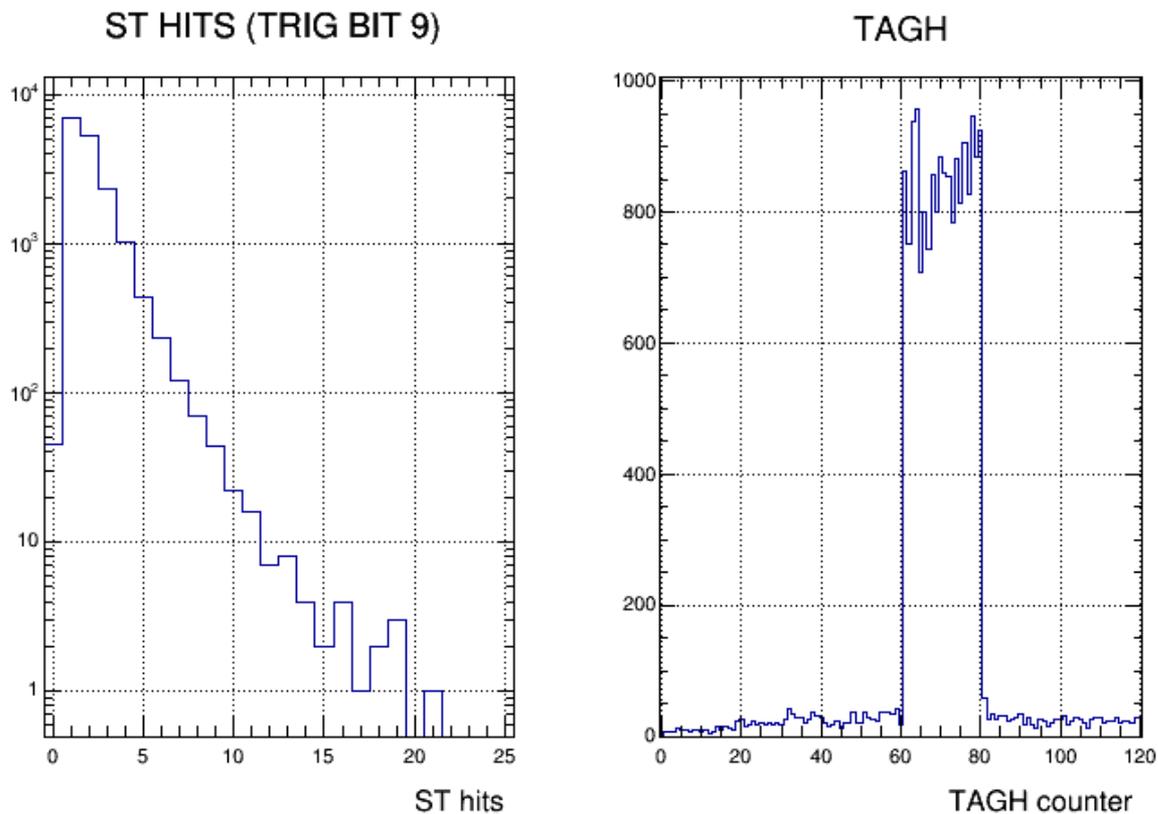


Relatively good agreement (missing hits in FCALHit when fadc timing algorithm failed)
Use raw data verify hardware performance

BCAL Trigger



TAGH & ST Trigger



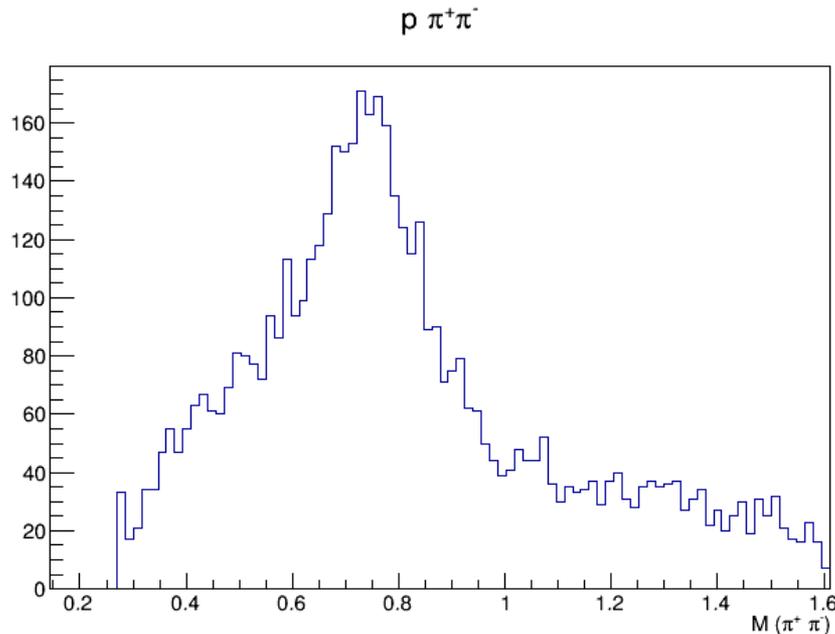
20 TAGH counters in coincidence with ST
- large accidental rate at high-lumi

Yield of ρ Mesons for Various Trigger Types

- **Event selection**

- one π^- candidate and two charged tracks in the event
- one proton candidate based on dE/dx
- extrapolate tracks to the FCAL or BCAL, require cluster matching
- require no energy deposition in the FCAL / BCAL (except from three tracks)

Efficiency for Events with 3 tracks ($\pi\pi\rho$)



Efficiency > 95 %
(based on small data sample)

fraction of rho candidates
for TAGH & ST trigger:
- 10 % small lumi
- 0.5 % at high lumi
(225 nA, presc 65)
(17 cand per 1 M triggers)

Comparison of FCAL & BCAL & ST and FCAL & BCAL:
- relative efficiency 98 %

Simulation of L1 trigger

- **Interface with the RCDB (GTP, FADC settings, masked channels) and CCDB (energy calibration, peak to integral, etc.)**
 - add table for masked channels (Dmitry)
- **Currently testing**