

HPS DAQ updates

Sergey Boyarinov

JLAB

June 23, 2021

HPS DAQ & Trigger Requirements

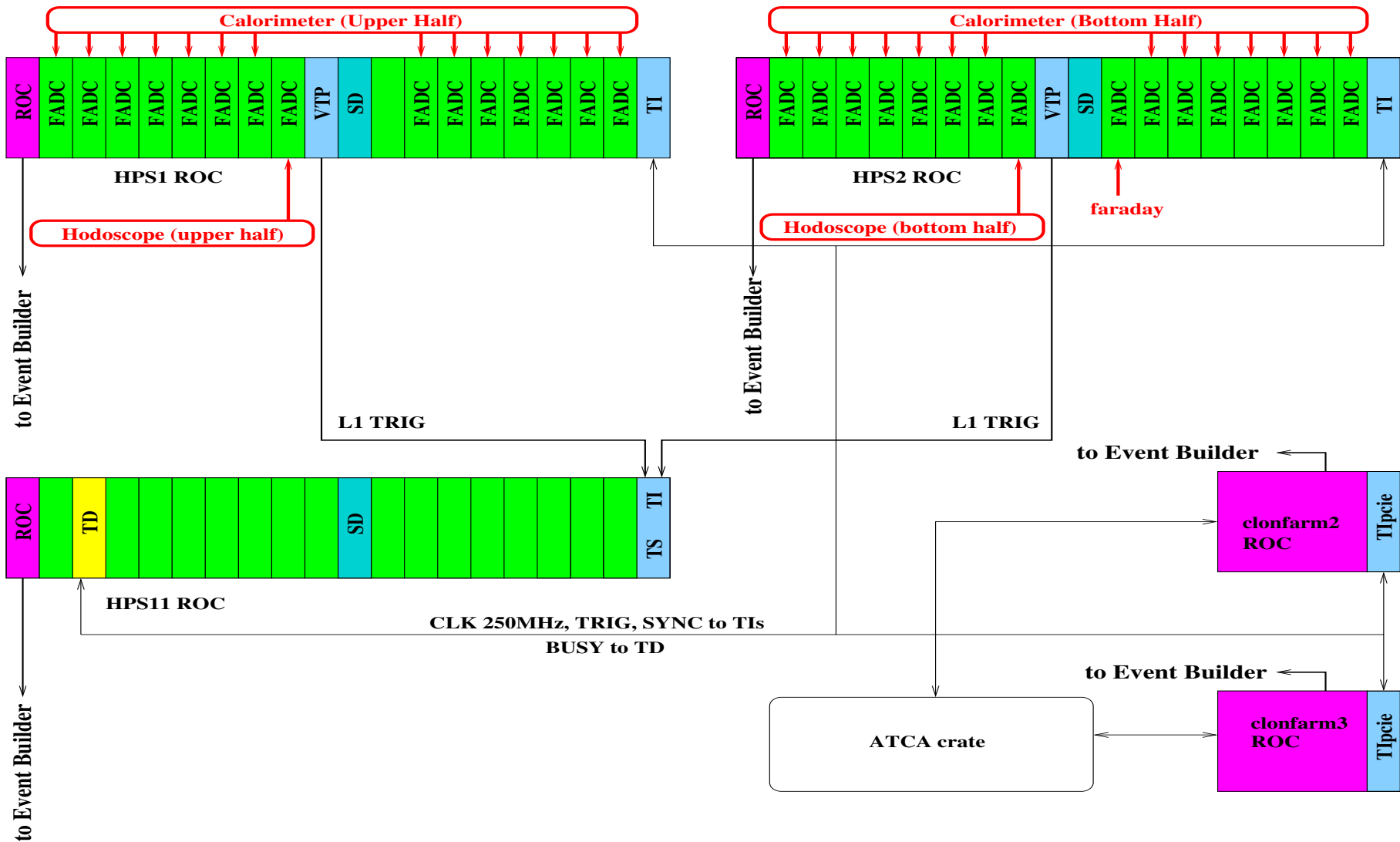
20kHz event rate

100MB/s data rate

>95% livetime

Achieved event rate 27kHz

HPS DAQ/Trigger Front-End Electronics



3 VXS crates, 2 servers, 1 ATCA crate, 7 Readout Controllers

HPS DAQ Status

- **Calorimeter Readout: 442 channels of 12bit 250MHz Flash ADCs**
- **Hodoscope Readout: 32 channels of 12bit 250MHz Flash ADCs**
- **CPU/VTP/TS/SD/TD trigger and signal distribution boards**
- **3 VXS crates**
- **2 servers (clonfarm2/3) with Tlpcie cards for SVT readout**
- **SVT readout (front-end and ATCA blades)**

Back-end computing and software is CLAS12 facility: network, computing (clondaq7), DAQ software, data monitoring, messaging system, realtime database

HPS DAQ Status

RunControl

File Preference

Run control Buttons

Control Transition

Cancel End Run

Reset Disconnect

Static parameters

Database	Session	Configuration	rcServer
hpsrun	clashps	KOD77_NOSV	clondaq7.jlab.org

Session status

Data file name

/data/stage_in/hpsnosvt

Config file name

lease/1.4.0/parms/trigger/HPS/Run2019/hps_v10_ram

Run status

Run number	Run status
10757	active
Start time	End time
Jun 23 11:34:24	
Limits	
Events	KBytes
0	0

Events/Sec

Run progress

Events this run

1709320 Read From: EB77

Rates	Events/S	Rate (KB/S)
Integrated	27569.6774	82268.3123
Differential	29640.0000	90613.2480

Help Statistics Options codaedit dbedit rocs

File Preference

rocs ocs1 ocs2 ocs3 ocs4 ocs5 ocs6 ocs7 ocs8 ocs9 ocs10

hps11 on hps11

```
net_thread: waiting= 34 sending= 0 microsec per event (nev=299)
proc_thread: waiting= 303 processing= 33 microsec per event (nev=30)
net_thread: waiting= 33 sending= 0 microsec per event (nev=300)
timer: 1 microsec (min= 1 max= 24 rms**2= 40)
proc_thread: waiting= 305 processing= 33 microsec per event (nev=30)
net_thread: waiting= 33 sending= 0 microsec per event (nev=301)
```

EB77 on clondaq7

```
<<< time2= 19 microsec (nevents2put=10) >>>
<<< time2= 19 microsec (nevents2put=100) >>>
<<< time2= 19 microsec (nevents2put=90) >>>
<<< time2= 18 microsec (nevents2put=100) >>>
<<< time2= 18 microsec (nevents2put=50) >>>
<<< time2= 19 microsec (nevents2put=10) >>>
```

hps1 on hps1

```
net_thread: waiting= 33 sending= 0 microsec per event (nev=296)
timer: 121 microsec (min= 117 max= 184 rms**2= 236112)
proc_thread: waiting= 286 processing= 50 microsec per event (nev=30)
net_thread: waiting= 33 sending= 0 microsec per event (nev=305)
proc_thread: waiting= 207 processing= 50 microsec per event (nev=29)
net_thread: waiting= 33 sending= 0 microsec per event (nev=296)
```

ET77 on clondaq7

```
UDP_standard_request >sta:ET77 active<
UDP_cancel: cancel >sta:ET77 paused<
codaUpdateStatus: updating request done
codaExecute done
CODAtcpServerWorkTask exit ?
CODAtcpServerWorkTask exit !
```

hps2 on hps2

```
net_thread: waiting= 33 sending= 0 microsec per event (nev=293)
timer: 135 microsec (min= 131 max= 207 rms**2= 292835)
proc_thread: waiting= 284 processing= 52 microsec per event (nev=30)
net_thread: waiting= 33 sending= 0 microsec per event (nev=300)
proc_thread: waiting= 286 processing= 52 microsec per event (nev=29)
net_thread: waiting= 33 sending= 0 microsec per event (nev=298)
```

title will be here

hps1vtp on hps1vtp

```
net_thread: waiting= 32 sending= 1 microsec per event (nev=297)
net_thread: waiting= 32 sending= 1 microsec per event (nev=299)
proc_thread: waiting= 216 processing= 120 microsec per event (nev=29)
timer: 12 microsec (min= 11 max= 309 rms**2= 2512)
proc_thread: waiting= 217 processing= 121 microsec per event (nev=29)
net_thread: waiting= 32 sending= 1 microsec per event (nev=298)
```

title will be here

hps2vtp on hps2vtp

```
net_thread: waiting= 32 sending= 1 microsec per event (nev=297)
timer: 12 microsec (min= 11 max= 299 rms**2= 2569)
proc_thread: waiting= 209 processing= 127 microsec per event (nev=29)
net_thread: waiting= 32 sending= 1 microsec per event (nev=299)
proc_thread: waiting= 211 processing= 126 microsec per event (nev=30)
net_thread: waiting= 32 sending= 1 microsec per event (nev=301)
```

title will be here

Development for upcoming run – SLAC setup

Two Tlpcie boards and one Tlpcie-new board were shipped to SLAC, along with VME TI board; VME crate with CPU was already at SLAC

Setup built at SLAC includes VME crate as master and two Linux servers with Tlpcie cards as slaves

CODA software was installed at SLAC with appropriate libraries, some modifications were made for Ubuntu (JLAB runs RHEL/CentOS, not Ubuntu) and for newer gcc

DAQ without SVT runs at 60kHz random pulser and 95% live, it indicates that all 3 TI cards communicates correctly

SVT integration has been started

Development for upcoming run – SLAC setup

The screenshot displays the RunControl software interface, which is used for managing and monitoring the SLAC (Stanford Linear Accelerator Center) setup. The interface is divided into several sections:

- Run control Buttons:** Includes buttons for Cancel, Reset, Disconnect, Configure, and Download.
- Static parameters:** Shows the current session configuration, including the Database (lpsrcm), Session (lpsrpd), Configuration (slac1_rsrv309), and rsServer (rsrv309).
- Session status:** Displays the Data file name and Config file name (rul/hps/server/cls12/parms/trigger/hps_v1_noThresholds.cnf).
- Run status:** Shows the Run number (30), Run status (configured), Start time, End time, Limits, Events (0), and KBytes (0).
- Run progress:** Displays the Events table with columns for Rates, Events/S, and Rate (KB/S). The integrated and differential rates are both 0.0.
- Log window:** Shows the execution logs for various components, including slac1, rsrv309, EB309, and ET309. The logs indicate the status of the components, such as "codeExecute done" and "codeExecute: alive" request.

Development for upcoming run – Tlpcie-new

Two Tlpcie-new boards were produced so far

Tlpcie-new was tested in servers at JLAB where old version failed, and it works, servers recognize it as standard PCI device

Library being developed, takes longer then expected, hope to get it ready soon

More boards are in production, it was delayed because of chips shortage problem, but recently all components arrived so production was resumed

SVT readout have to be (probably) adjusted using Tlpcie-new boards

Tipcie-new module

Better PCIe compatibility:

- Xilinx UltraScale+ FPGA: xcku3p
- PCIeexpress from gen1x1 (low power) to gen3x8 (high bandwidth)

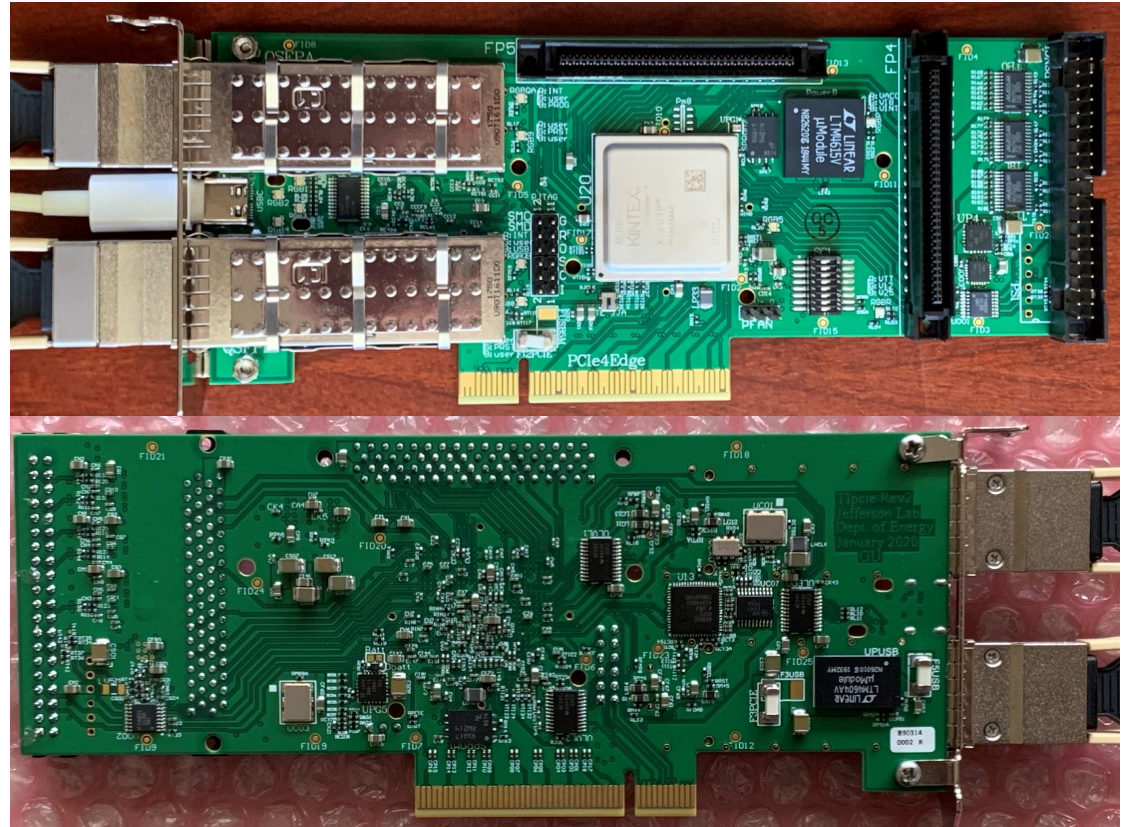
More like a VME TI:

- TI fiber#1, fiber#5
- 40-pin IO to the second front panel
8 outputs + 1 clock output;
(LVDS, 3 optional ECL)
10 inputs + 1 clock input; (any
diff. level)

More potentials:

- Another 64 LVDS connection to the
FPGA:
64-channel FPGA based TDC
- USBC connector on the front panel
+5V in for standalone operation

Two boards available, one in JLAB and
another in SLAC, library being
developed, more boards are in
production



DAQ Status

All hardware and software on JLAB side is ready to run any time

Integration with SVT DAQ has underway in test setup at SLAC using old Tlpcie modules

Tlpcie-new modules to be commissioned, it will provide enough spares for upcoming run