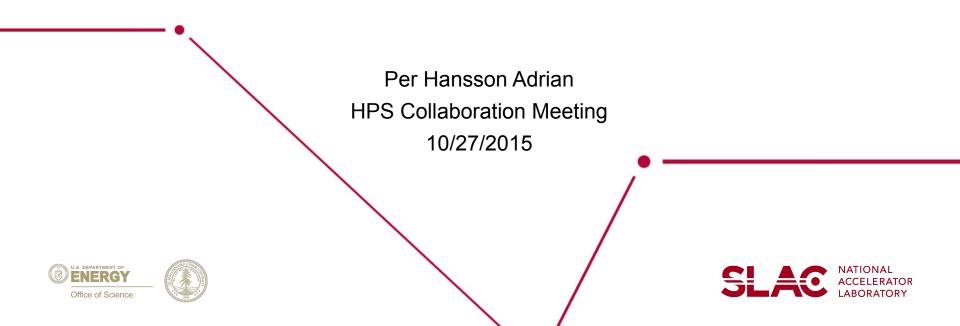
SVT DAQ



Overview

SLAC

Trigger rate improvements Optimized data format Shorter APV25 shaping time Single event upset monitor Data integrity

Plans

Deadtime Improvements: Simulation

Data taking efficiency Live time fraction Engineering run in green We are running 0.9 with yellow settings 0.8 0.7 0.6 0.5 TI rule=none TI rule= TI-1=1.4 TI-4=83.0 0.4 TI rule= TI-1=1.4 TI-4=88.4 TI rule= TI-1=1.4 TI-5=110.0 0.3 TI rule= TI-1=1.4 TI-4=30.8 ×10³ 0.2^L0 30 50 80 90 10 20 40 60 70 100 Average trigger rate (Hz)

NOTE: the average rate here is input or "ungated" rate

SLAC

Deadtime Improvements: DAQ setup

Trigger using special ecal configuration

- HV OFF
- Two crystals with low thresholds (all other super large)
- Singles-0 and singles-1 active with different thresholds (to allow "fine" tuning of rates)
- Event size with ECal only is ~ 1kB per event

SVT is configured normally

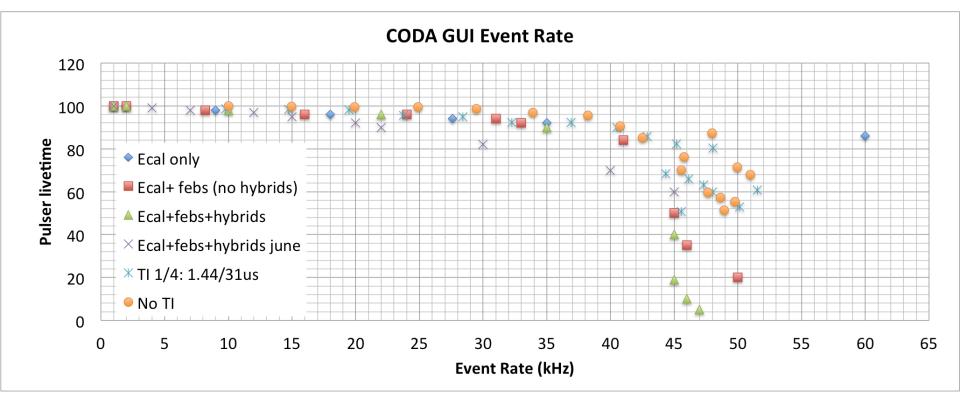
- Chiller at +5C: hybrids at ~8C while running)
- Bias at 180V
- All FEBs and hybrids included (if not otherwise specified)
- New thresholds derived and applied at this temperature
- Event size with SVT was ~4kB per event

SVT asserts busy based on internal APV25 pipeline status

Good news is that we never crashed with the SVT in (run for hours with lots of high rates) => Busy is working.

SL AC

Deadtime Improvements: GUI rates



"No TI" and "TI 1/4" are simulated rates, stat. uncertainty obvious at high rates Live time seems turns down at the same place (except June tests)

SLAO

SLAC

Big improvement

- DAQ live time with this setup is about 92% at 35kHz. Drops to about 45% at 45kHz
- Up to 35kHz within few percent from simulation. At 45kHz we are around 40% lower than simulation. Not clear why.

Issues

- Hard to get a good measurement at 40-50kHz because rates as measured fluctuates wildly
- One idea is that rate measurements are not integrating long enough (we are close to the limit)
- We also had an issue correlating the CODA GUI and EPICS variable rates (we sum all gated rates)

Next steps

- Hold off more tests until DAQ has been upgraded (new OS and network)
- Figure out any last (significant) difference w.r.t. simulation

SVT Data Format SLAC Data from a single RCE (14 in the event) MULTISAMPLE DATA **BANK-of-BANKs** SVT_DATA_HEADER SVT EVENT HEADER MULTISAMPLE TAIL **MULTISAMPLES** UINT32 DATA TI_EVENT_HEADER Each trigger: TI DATA For each APV ch. passing thresholds: **MULTISAMPLE DATA: 6 samples** . . . from an APV ch. **UINT32 DATA** MULTISAMPLE TAIL: Type of SVT DATA EVENT **MULTISAMPLES** HEADER multisample, Ch ID, error bits SVT DATA TAIL SVT DATA For every APV in that RCE MULTISAMPLE DATA: error/sync and de-multiplexing data MULTISAMPLE_TAIL: Type of multisample, Ch ID, error bits Call this "multisample header" • "All headers" mode : this is engineering run 2014 "First header" mode removes all but one multisample headers per RCE 7 "No header" mode removes all but one multisample headers

SVT Data Format

ank content="bank" data	a_type="0xe" tag	="51" padding="(0" num="64" len	igth="72" ndata="70">
<uint32 data_type="</td"><td></td><td></td><td></td><td>"6" ndata="4"></td></uint32>				"6" ndata="4">
0x20010003	0×40	Øxcde82c9	0×1	
<uint32 data_type="</td"><td></td><td></td><td></td><td></td></uint32>				
0x100003f	0x96f594f7	0x9aff98fd	0x9ef99cfb	0x42000800
0x36f534f7	0x3aff38fd	0x3ef93cfb	0×48800800	0x76f574f7
0x7aff78fd	0x7ef97cfb	0x4d800800	0x56f554f7	0x5aff58fd
0x5ef95cfb	0×49000800	0x96f594f7	0x9aff98fd	0x9ef99cfb
0x4e000800	0x16f514f7	0x1aff18fd	0x1ef91cfb	0×40000800
0x16f514f7	0x1aff18fd	0x1ef91cfb	0x4c000800	0x36f534f7
0x3aff38fd	0x3ef93cfb	0×40800800	0x76f574f7	0x7aff78fd
0x7ef97cfb	0×49800800	0x56f554f7	0x5aff58fd	0x5ef95cfb
0×41000800	0x96f594f7	0x9aff98fd	0x9ef99cfb	0x4a000800
0x36f534f7	0x3aff38fd	0x3ef93cfb	0x4c800800	0x16f514f7
0x1aff18fd	0x1ef91cfb	0×48000800	0x56f554f7	0x5aff58fd
0x5ef95cfb	0x4d000800	0x76f574f7	0x7aff78fd	0x7ef97cfb
0x41800800	0xf			
	data type=" 0xe " [·]	tag="52" padding	7="0" num="64"	length="92" ndata="90"
<uint32 data_type="</td"><td></td><td>" padding="0" ni</td><td>um="52" length=</td><td></td></uint32>		" padding="0" ni	um="52" length=	
0×20010003	"0x1" tag="57610 0x40			
0x20010003 	0×40	" padding="0" ni 0xcde82c9	um="52" length= 0x1	"6" ndata="4">
0x20010003 <uint32 data_type="</td"><td>0x40 "0x1" tag="3" pa</td><td>" padding="0" ni Øxcde82c9 dding="0" num="5</td><td>um="52" length= 0x1 52" length="84"</td><td>"6" ndata="4"> ' ndata="82"></td></uint32>	0x40 "0x1" tag="3" pa	" padding="0" ni Øxcde82c9 dding="0" num="5	um="52" length= 0x1 52" length="84"	"6" ndata="4"> ' ndata="82">
0x20010003 <uint32 data_type="<br">0x100003f</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7	" padding="0" ni 0xcde82c9 dding="0" num="5 0x5aff58fd	um="52" length= 0x1 52" length="84" 0x5ef95cfb	"6" ndata="4"> ndata="82"> 0x45000801
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd	" padding="0" ni 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb	um="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701	"6" ndata="4"> ndata="82"> 0x45000801 0x76f574f7
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd 0x7ef97cfb	" padding="0" nu 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb 0x4d800701	um="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7	"6" ["] ndata="4"> 'ndata="82"> 0x45000801 0x76f574f7 0x7aff78fd
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301	" padding="0" nu 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb 0x40800701 0x56f554f7	<pre>im="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd</pre>	"6" ndata="4"> ndata="82"> 0x45000801 0x76f574f7 0x7aff78fd 0x5ef95cfb
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x49000701</uint32>	0x40 "0x1" tag="3" par 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301 0x96f594f7	" padding="0" nu 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb 0x4d800701 0x56f554f7 0x9aff98fd	<pre>im="52" length=</pre>	"6" ndata="4"> ndata="82"> 0x45000801 0x76f574f7 0x7aff78fd 0x5ef95cfb 0x4e000701
0x20010003 uint32 data_type= 0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x49000701 0x96f594f7	0x40 "0x1" tag="3" par 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301 0x96f594f7 0x9aff98fd	" padding="0" ni 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb 0x4d800701 0x56f554f7 0x9aff98fd 0x9ef99cfb	<pre>im="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301</pre>	"6" ndata="4"> ndata="82"> 0x45000801 0x76f574f7 0x5ef95cfb 0x4e000701 0x76f574f7
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x49000701 0x96f594f7 0x7aff78fd</uint32>	0x40 "0x1" tag="3" pai 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301 0x96f594f7 0x9aff98fd 0x7ef97cfb	<pre>" padding="0" ni 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3ef93cfb 0x4d80701 0x56f554f7 0x9aff98fd 0x9ef99cfb 0x45800801</pre>	<pre>m="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7</pre>	"6" ndata="4"> 0x45000801 0x76f574f7 0x7aff78fd 0x5ef95cfb 0x7ef0701 0x76f574f7 0x7aff18fd
0x20010003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x49000701 0x96f594f7 0x7aff78fd 0x1ef91cfb</uint32>	0x40 "0x1" tag="3" par 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301 0x96f594f7 0x9aff98fd 0x7ef97cfb 0x4c000701	padding="0" ni 0xcde82c9 0x5aff58fd 0x3ef93cfb 0x4d800701 0x56f554f7 0x9aff98fd 0x9ef99cfb 0x4580801 0x96f594f7	<pre>m="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7 0x9aff98fd</pre>	"6" ndata="4"> 0x45000801 0x76f574f7 0x5ef95cfb 0x4600701 0x76f574f7 0x46000701 0x76f574f7 0x1aff18fd 0x9ef99cfb
0x200706003 <uint32 0x100003f 0x36f534f7 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd</uint32 	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x4d800301 0x96f594f7 0x3aff38fd 0x7ef97cfb 0x7ef97cfb 0x4c000701 0x76f574f7	<pre>" padding="0" ni 0xcde82c9 0xsde82c9 0xsaff58fd 0x3af93cfb 0x4d800701 0x56f554f7 0x9aff98fd 0x9aff99cfb 0x45800801 0x96f594f7 0x7aff78fd</pre>	<pre>m="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7 0x9aff98fd 0x7ef97cfb</pre>	"6" ndata="4"> ndata="82"> 0x45000001 0x76f574f7 0x5ef95cfb 0x4e000701 0x16f574f7 0x3ef99cfb 0x3ef99cfb 0x4909cfb 0x4980701
0x20070003 <uint32 data_type="<br">0x100003f 0x30f534f7 0x7aff78fd 0x7ef97cfb 0x49000701 0x96f594f7 0x7aff78fd 0x1ef91cfb 0x46500801 0x16f514f7</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x40800301 0x96f594f7 0x9aff98fd 0x7ef97cfb 0x4c000701 0x76f574f7 0x3aff18fd	<pre>" padding="0" ni 0xcde82c9 dding="0" num="! 0x5aff58fd 0x3ef93cfb 0x5aff554f7 0x9aff98fd 0x9ef99cfb 0x4580801 0x96f594f7 0x7aff78fd 0x1ef91cfb</pre>	<pre>m="52" length= 0x1 52" length="84" 0x5ef95cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7 0x9aff98fd 0x7ef97cfb 0x4c000301</pre>	"6" ndata="4"> ndata="82"> 0x35008001 0x76f5747 0x7aff78fd 0x5ef95cfb 0x4e000701 0x1aff18fd 0x9ef99cfb 0x9699cfb 0x49800701 0x16f151477
	0x40 "0x1" tag="3" pa 0x56755477 0x3aff38fd 0x7e197cfb 0x40800301 0x96f594f7 0x3aff98fd 0x7e197cfb 0x4c000701 0x76f574f7 0x1aff18fd 0x1aff18fd 0x1aff18fd	<pre>"padding="0" ni 0xcde82c9 dding="0" num="1" 0x5aff58fd 0x3af93cfb 0x4680701 0x56f554f7 0x9aff98fd 0x9af99cfb 0x9aff98fd 0x9aff98ff 0x7aff78fd 0x1ef91cfb 0x44008001</pre>	<pre>imm="52" length= 0x1 0x2 length="84" 0x5ef50cfb 0x48800701 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7 0x9aff98fd 0x7ef97cfb 0x4c000301 0x96f594f7</pre>	"6" ndata="4"> ndata="82"> 0x45000001 0x76157477 0x7aff78fd 0x5ef95cfb 0x4e000701 0x761574f7 0x3ef99cfb 0x49509cfb 0x49509cfb 0x49508701 0x46f514f7 0x3ef198fd
0x20070003 <uint32 data_type="<br">0x100003f 0x30f534f7 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x1aff178fd 0x45000801 0x1aff18fd 0x1aff18fd</uint32>	0x40 "0x1" tag="3" pa 0x56f554f7 0x3aff38fd 0x7ef97cfb 0x40800301 0x96f594f7 0x3aff38fd 0x7ef97cfb 0x4c000701 0x76f574f7 0x1aff18fd 0x1ef91cfb 0x4c000701	<pre>"padding="0" ni 0xcde82c9 dding="0" num="5 0x5aff58fd 0x3af93cfb 0x4d800701 0x9aff98fd 0x9af99cfb 0x456980801 0x96f594f7 0x7aff78fd 0x1af91cfb 0x3aff534f7</pre>	<pre>imm="52" length= 0x1 0x5ef95cfb 0x76f574f7 0x5aff58fd 0x9ef99cfb 0x4e000301 0x16f514f7 0x9aff98fd 0x7ef97cfb 0x4c000301 0x96f594f7 0x3aff38fd</pre>	"6" ndata="4"> ndata="82"> 0x45008081 0x76f5747 0x7aff78fd 0x5ef95cfb 0x4e000701 0x1aff18fd 0x9ef99cfb 0x96f99cfb 0x49600701 0x3aff98fd 0x3ef93cfb
	0x40 "0x1" tag="3" pa 0x56755477 0x3aff38fd 0x7e197cfb 0x4d800301 0x96759477 0x3aff38fd 0x7e197cfb 0x4c000701 0x76157477 0x1aff18fd 0x1e191cfb 0x4a000701 0x36753477	** padding=*0** nd 0xcde82c9 ddng=*0** num=*1 dx3aff58fd 0x3aff38fd 0x3aff38fd 0x3aff38fd 0x9aff38fd 0x9aff38fd 0x45800801 0x7aff78fd 0x1af91cfb 0x44000001 0x3aff334f7	im="52" length= 8x1 %x5ef95cfb %x5ef95cfb %x7ef574f7 %x5aff58fd %x9aff98fd %x4e000301 %x9aff98fd %x4e000301 %x9aff98fd %x4e000301 %x4600301 %x45f54f77 %x3aff38fd %x3ef39cfb	"6" ndata="4"> ndata="82"> 0x45000001 0x76f574f7 0x7aff78fd 0x5ef95cfb 0x4c000701 0x1aff18fd 0x969907b 0x43080701 0x1aff18fd 0x95080701 0x3aff93fd 0x3af93cfb 0x4c808301
0x20070003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7af778fd 0x7aff78fd 0x7aff78fd 0x16f514f7 0x1aff18fd 0x16f514f7 0x1aff18fd 0x16f514f7</uint32>	0x40 "0x1" tag="3" pa 0x56f5547 0x3aff38fd 0x7ef97cfb 0x4d800301 0x36f594f7 0x3aff38fd 0x7ef97cfb 0x4c000701 0x76f574f7 0x1aff18fd 0x4ef91cfb 0x4a000701 0x36f534f7 0x3aff18fd	"padding="0" ni 0xcde82c9 dding="0" num="3 0x5af158fd 0x5af158fd 0x5af158fd 0x5af198fd 0x5af198fd 0x5af198fd 0x5af198fd 0x5af158ff 0x1af191cfb 0x3af138fd 0x3af138fd 0x3af138fd 0x1af131cfb	<pre>im="52" length=</pre>	"6" ndata="4"> ndata="82"> 0x450080801 0x76f5747 0x7aff78fd 0x5ef95cfb 0x4e000701 0x1aff18fd 0x9ef99cfb 0x49608701 0x3aff98fd 0x3aff98fd 0x3aff98fd 0x3aff93cfb 0x46808301 0x56f55477
0x20070003 <uint32 data_type= 0x100003f 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x7aff78fd 0x49000701 0x7aff78fd 0x1ef91cfb 0x460080801 0x16f514f7 0x1aff18fd 0x645097cfb 0x1aff18fd 0x5aff58fd 0x16f514f7 0x5aff58fd</uint32 	0x40 0x1" tag="3" pa 0x56755477 0x3aff38fd 0x7e197cfb 0x40800301 0x96759477 0x9aff98fd 0x7e197cfb 0x4c000701 0x76757477 0x1aff18fd 0x4a000701 0x3aff38f477 0x1aff18fd 0x5e195cfb	** padding=*** mi %xcde82c9 %x3cf387f58f6 %x3cf387f58f6 %x3cf387d7 %x5af58f7 %x5af58f7 %x5af58f7 %x45806801 %x45806801 %x7af78f6 %x45806801 %x7af78f7 %x44000801 %x3af738f7 %x3af738f7 %x3af738f7 %x460080701	im="52" length="84" 0x1 52" length="84" 0x548960701 0x56757477 0x56757477 0x56757477 0x56757477 0x96757477 0x96757477 0x96759477 0x36759477 0x36753477 0x36753477 0x36753477	"6" ndata="4"> ndata="82"> 0x35000001 0x76f754f7 0x7aff78fd 0x5ef95cfb 0x4c000701 0x1aff18fd 0x9699cfb 0x43000701 0x1aff18fd 0x3ef93cfb 0x3aff35fb 0x3ef93cfb 0x3cf55faf7 0x5aff58fd
0x20070003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x45000001 0x450008001 0x16f514f7 0x1aff18fd 0x4c0000701 0x1aff18ff 0x4c56f514f7</uint32>		"padding="0" ni 0xcde82c9 dding="0" num="3 0x5af158fd 0x5af158fd 0x5af158fd 0x5af198fd 0x5af198fd 0x5af198fd 0x5af198fd 0x5af158ff 0x1af191cfb 0x3af138fd 0x3af138fd 0x3af138fd 0x1af131cfb	<pre>im="52" length=</pre>	"6" ndata="4"> ndata="82"> 0x450080801 0x76f5747 0x7aff78fd 0x5ef95cfb 0x4e000701 0x1aff18fd 0x9ef99cfb 0x49608701 0x3aff98fd 0x3aff98fd 0x3aff98fd 0x3aff93cfb 0x46808301 0x56f55477
0x20070003 <uint32 data_type="<br">0x1000037 0x36f534f7 0x7aff78fd 0x7ef97cfb 0x7aff78fd 0x49000701 0x7aff78fd 0x1ef91cfb 0x460080801 0x1aff18fd 0x5ef99cfb 0x1aff18fd 0x5aff58fd 0x5aff58fd 0x5aff58fd 0x5aff58fd 0x5aff58fd 0x5aff58fd 0x5aff58fd</uint32>	0x40 0x1" tag="3" pa 0x56755477 0x3aff38fd 0x7e197cfb 0x40800301 0x96759477 0x9aff98fd 0x7e197cfb 0x4c000701 0x76757477 0x1aff18fd 0x4a000701 0x3aff38f477 0x1aff18fd 0x5e195cfb	** padding=*** mi %xcde82c9 %x3cf387f58f6 %x3cf387f58f6 %x3cf387d7 %x5af58f7 %x5af58f7 %x5af58f7 %x45806801 %x45806801 %x7af78f6 %x45806801 %x7af78f7 %x44000801 %x3af738f7 %x3af738f7 %x3af738f7 %x460080701	im="52" length= 0x1 0x1 0x52" length="64" 0x58795cfb 0x587657477 0x587657477 0x587657477 0x587757477 0x96757477 0x96757477 0x96759477 0x38773876 0x76797cfb 0x76797cfb 0x76797cfb 0x36753477 0x38733cfb 0x38733cfb 0x586755477	"6" ndata="4"> ndata="82"> 0x35000001 0x76f754f7 0x7aff78fd 0x5ef95cfb 0x4c000701 0x1aff18fd 0x9699cfb 0x43000701 0x1aff18fd 0x3ef93cfb 0x3aff35fb 0x3ef93cfb 0x3cf55faf7 0x5aff58fd
0x20070003 <uint32 data_type="<br">0x100003f 0x36f534f7 0x7aff78fd 0x7aff78fd 0x7aff78fd 0x45000001 0x450008001 0x16f514f7 0x1aff18fd 0x4c0000701 0x1aff18ff 0x4c56f514f7</uint32>		** padding=*** mi %xcde82c9 %x3cf387f58f6 %x3cf387f58f6 %x3cf387d7 %x5af58f7 %x5af58f7 %x5af58f7 %x45806801 %x45806801 %x7af78f6 %x45806801 %x7af78f7 %x44000801 %x3af738f7 %x3af738f7 %x3af738f7 %x460080701	im="52" length= 0x1 0x1 0x52" length="64" 0x58795cfb 0x587657477 0x587657477 0x587657477 0x587757477 0x96757477 0x96757477 0x96759477 0x38773876 0x76797cfb 0x76797cfb 0x76797cfb 0x36753477 0x38733cfb 0x38733cfb 0x586755477	"6" ndata="4"> ndata="82"> 0x35000001 0x76f754f7 0x7aff78fd 0x5ef95cfb 0x4c000701 0x1aff18fd 0x9699cfb 0x43000701 0x1aff18fd 0x3ef93cfb 0x3aff35fb 0x3ef93cfb 0x3cf55faf7 0x5aff58fd



SVT overhead reduction ~ 0.075

- Engineering run had up to 80 overhead words from APV25 chips per ROC
- New format has 4

Large fraction of SVT data was overhead

- Needed for data integrity checks: implemented those in firmware now
- We have typically more APV25 chips than hits in the SVT
- With ECal in special mode, and NO beam, we see a 86% reduction in SVT bank size and ~70% reduction in HPS event size.

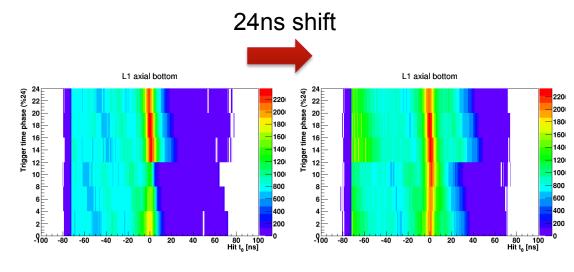
Trigger Delay and Faster Shaping Time

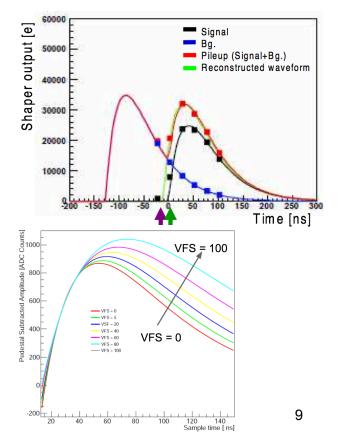
Increase adjustment of sampling point: DONE – not tested

- Added firmware option to delay triggers being sent to Front End Board in steps of 8 ns.
- Allows for more optimal timing-in of SVT to improve pile-up rejection: we are picky

Reduce APV25 shaping time to 35ns

- Less sensitive to pile-up
- Hopefully slightly better time resolution
- Ongoing work





SLAC

Monitor the FPGA for errors

- Xilinx FPGA in FEB has ability to continually monitor the configuration for errors caused by Single Event Upsets (SEUs), typically caused by neutron radiation
- Not enabled during the Engineering run

Implementation

- Option 1: stream asynchronous data to mounted filesystem continuously (time stamped)
- Implement registers for monitoring and alarms (EPICS and ALH)
- DAQ will lock up in case of errors
- Ongoing work

SLA0

SLAC

SVT DAQ Errors Latency setting

Burst mode noise

Analysis is ongoing

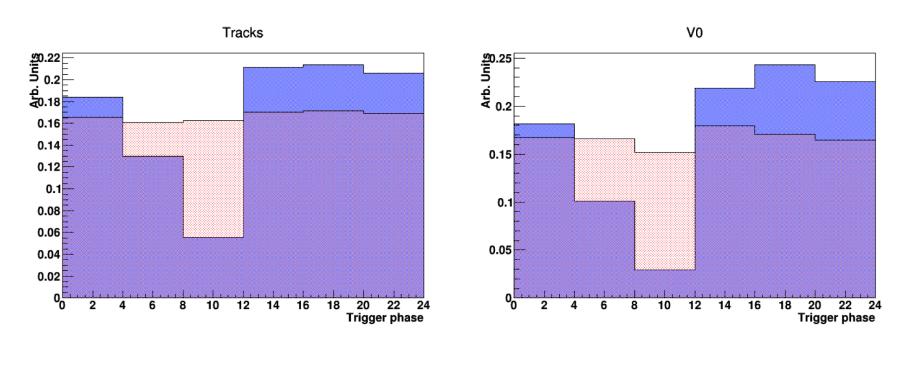
- Initially looked at subset of errors in 10% of the data
 - Saw very few (~500) data link error issues
 - More importantly saw ~2% of the events affected by "SyncError"; this error latches on (and was responsible for missing hybrids that some shifters found)
- Work is ongoing to look at all runs and all errors; will give an update soon

Physics analysis

 There is now (~pass3+) an event flag (Icio &DST) that tells you if there was any issues related to the SVT DAQ

Efficiency loss due to latency setting

- 1.7G 0.5 mm events before fix, 2.3G events after
- Lost 25% of vertex candidates in affected runs: 10% loss overall (would want to simulate this behavior in MC)
- Can just drop events in the 2 (of 6) affected trigger phases: lose 33% of events in affected runs, 15% overall



E

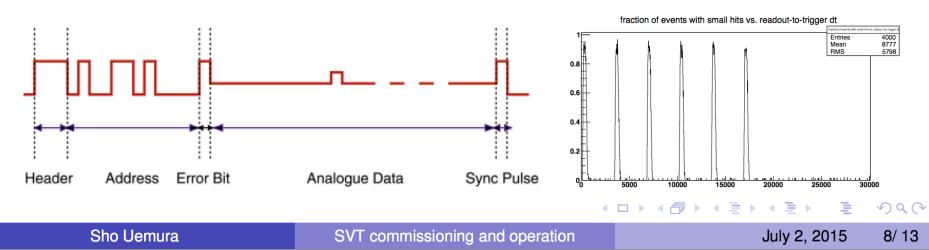
 $\mathcal{A} \mathcal{A} \mathcal{A}$

< ロ > < 同 > < 三 > < 三 >

SVT commissioning and operation

Burst-mode noise

- APV25 readout chip puts analog and digital signals on the same output line: 128 clocks of analog data, 12 clocks of digital data
- Preamp crosstalk from large currents/voltage swings: events coinciding with digital output will have extra noise
- "Burst mode" (allowing a trigger while the APV25 is still sending data) increases noise in the SVT (only in a subset of events)
 - Noisy events can be identified by looking for noise hits in the SVT (normally 0-2; noisy events have up to 100), and are correlated with time between triggers
 - From TI timestamps, we can identify all events where hits overlapped with digital output
 - Noisy events are 4-6% of data: just discard these



SLAC

Currently operating with SVT at -20C

Plan is to continue with this for a few more weeks

- Adjust APV25 shaping time
- Tune ADC sampling points across the system
- Investigate hybrid with handful of APV25 channels showing abnormal behavior

Misc.

- Need to replace broken RTD
- Need to add new flow meter (at JLab now)
- We plan not to test the spare flange due to unnecessary risks (unless analysis show link errors more prevalent)
- GUI updates

Summary

Large improvement in rates using TI busy

Expect reduction of SVT event size with about 80%

Ongoing work this fall

- Tune SVT operating points
- Implement SEU software and test
- More rate tests with updated DAQ
- Spares…

Documentation

- Data format note
- DAQ notes/papers

The Data Format for the HPS Engineering Run 2014

Per Hansson Adrian¹, Benjamin Reese¹, Ryan Herbst¹, Sergey Boiarinov², Nathan Baltzell² ¹SLAC National Accelerator Laboratory, Menio Park, CA, USA ²Thomas Jefferson National Accelerator Facility, Newport News, VA, USA

1 Overview

Description of the FADC?

The SVT DAQ consists of multiple data processing daughter boards (DPMs) with two processing nodes (RCEs) per DPM. Each RCE is accessing data from between two and four hybrids that each carry five APV25 readout ASICs. Each RCE runs a CODA Readout Controller (ROC) to transfer data to the CODA event builder.

2 The ECal Data Format

ECal data format description.

3 The SVT Data Format

The SVT data from each of Readout Controllers (ROCs) is stored in a SVT evio bank of type "bank-of-banks" as described in Tab. 1. During normal data taking there are 14 SVT data banks. The SVT evio bank contains information distributed by the master trigger interface (TI) board and the SVT DAQ itself. These are described in detail in Sec. 3.1 and 3.2, respectively.

3.1 SVT TI Data

Each of the SVT RCEs (one per ROC) obtain TI information from the TI interface. This information is attached to each of the events. Table 2 and Tab. 3 describes the TI bank header and the TI data, respectively.

 Table 1: SVT EVIO bank-of-banks description.

 Content
 Type
 Description

 TI Data
 UINT32 bank
 TI information

 SVT Data
 UINT32 bank
 SVT data

SLAC