TriggerPrimitive Generation: Progress Update

Alejandro Oranday

June 12, 2024

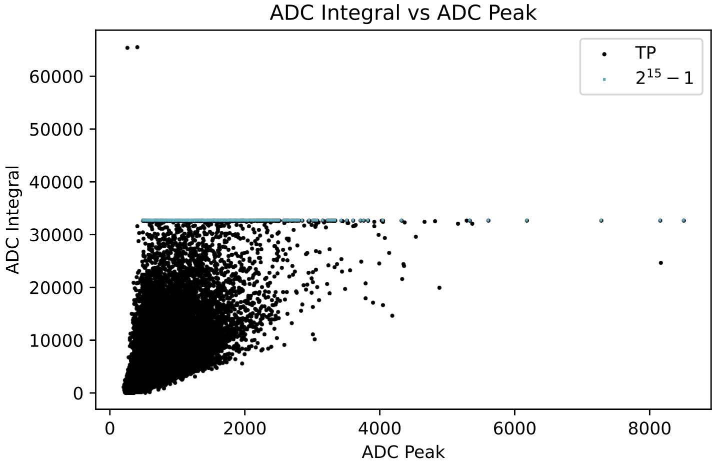
What’s been happening?

- NP04 starts running.
  - ⇒ RS ADC data parameters redefined:
    - ► ADC Integral, ADC Peak, and Peak Time.
  - ⇒ Configure RunningSum (RS) algorithms by plane.
  - ⇒ (***WIP***) TPHandler by plane.
- Documentation:
  - ► Frame expansion.
  - ► (***WIP***) TPG Algorithms.
- AVX2 vs Naive:
  - ► Consistent!
  - ⇒ Usable naive implementation in emulation & simulation.

RS ADC Redefinition

Previously struggled with negative values and centering to 0 ADC.

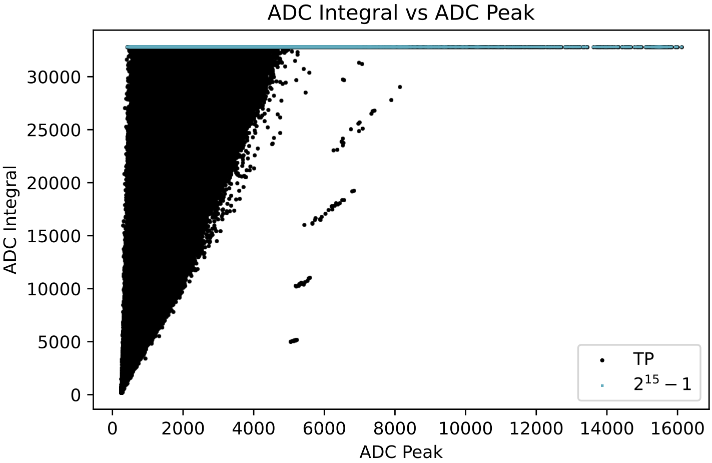
Old SRS



RS ADC Redefinition

No longer struggles!

New SRS



Plane Refactoring

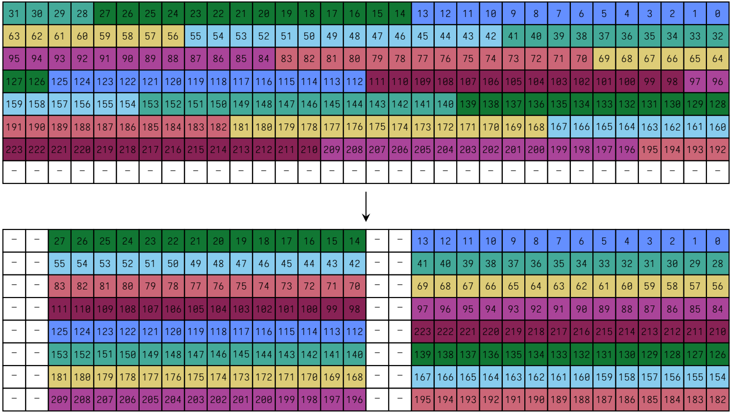
TPG by plane is a clear refactor point:

- Thresholding, RS configuring, and TPHandler identities are all by plane.
- Code duplication happened in a rush.
- Prompts a new configuration schema (*OKS*).

⇒ *Targets v5*.

Documentation

Diagrams of frame expansion: 14-bit to 16-bit.



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

↓

–	–	27	26	25	24	23	22	21	20	19	18	17	16	15	14	–	–	13	12	11	10	9	8	7	6	5	4	3	2	1	0
–	–	55	54	53	52	51	50	49	48	47	46	45	44	43	42	–	–	41	40	39	38	37	36	35	34	33	32	31	30	29	28
–	–	83	82	81	80	79	78	77	76	75	74	73	72	71	70	–	–	69	68	67	66	65	64	63	62	61	60	59	58	57	56
–	–	111	110	109	108	107	106	105	104	103	102	101	100	99	98	–	–	97	96	95	94	93	92	91	90	89	88	87	86	85	84
–	–	125	124	123	122	121	120	119	118	117	116	115	114	113	112	–	–	223	222	221	220	219	218	217	216	215	214	213	212	211	210
–	–	153	152	151	150	149	148	147	146	145	144	143	142	141	140	–	–	139	138	137	136	135	134	133	132	131	130	129	128	127	126
–	–	181	180	179	178	177	176	175	174	173	172	171	170	169	168	–	–	167	166	165	164	163	162	161	160	159	158	157	156	155	154
–	–	209	208	207	206	205	204	203	202	201	200	199	198	197	196	–	–	195	194	193	192	191	190	189	188	187	186	185	184	183	182

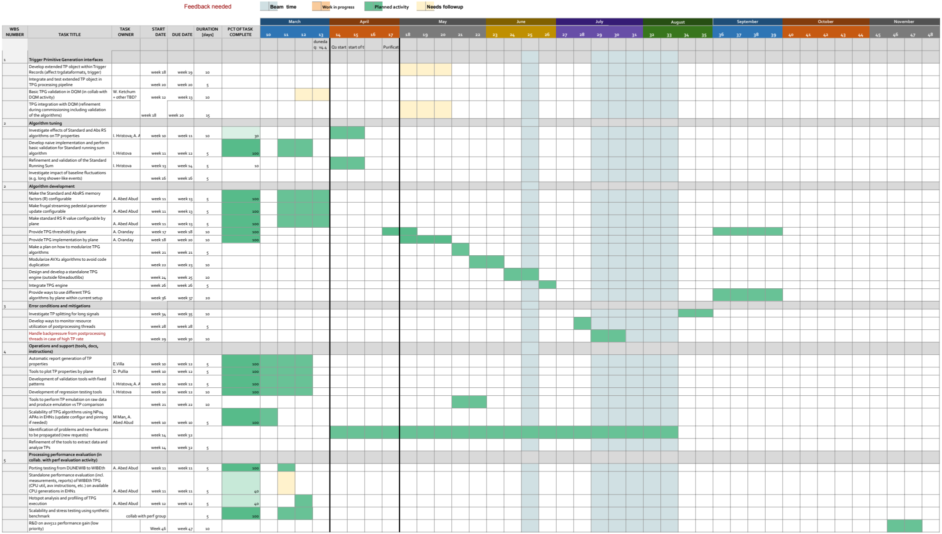
Documentation

Continue documentation for TPG algorithms.

- No complex register manipulation; less diagrams.
- Some AVX2 rule-bending; more math.

```
// Perform the division of __m256i with a const int
inline __m256i _mm256_div_epi16(const __m256i va, const int b)
{
    __m256i vb = _mm256_set1_epi16(32768 / b);
    return _mm256_mulhrs_epi16(va, vb);
}
```

TPG Gantt



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION [days]	PCT OF TASK COMPLETE
1	**Trigger Primitive Generation interfaces**					
	Develop extended TP object within Trigger Records (affect trgdataformats, trigger)		week 18	week 19	10	
	Integrate and test extended TP object in TPG processing pipeline		week 20	week 20	5	
	Basic TPG validation in DQM (in collab with DQM activity)	W. Ketchum + other TBD?	week 12	week 13	10	
	TPG integration with DQM (refinement during commissioning including validation of the algorithms)		week 18	week 20	15	
2	**Algorithm tuning**					
	Investigate effects of Standard and Abs RS algorithms on TP properties	I. Hristova; A. A	week 10	week 11	10	30
	Develop naive implementation and perform basic validation for Standard running sum algorithm	I. Hristova	week 11	week 12	5	100
	Refinement and validation of the Standard Running Sum	I. Hristova	week 13	week 14	5	10
	Investigate impact of baseline fluctuations (e.g. long shower-like events)		week 16	week 16	5	
2	**Algorithm development**					
	Make the Standard and AbsRS memory factors (R) configurable	A. Abed Abud	week 11	week 13	5	100
	Make frugal streaming pedestal parameter update configurable	A. Abed Abud	week 11	week 13	5	100
	Make standard RS R value configurable by plane	A. Abed Abud	week 11	week 13	5	100
	Provide TPG threshold by plane	A. Oranday	week 17	week 18	10	100
	Provide TPG implementation by plane	A. Oranday	week 18	week 20	10	100
	Make a plan on how to modularize TPG algorithms		week 21	week 21	5	
	Modularize AVX2 algorithms to avoid code duplication		week 22	week 23	10	
	Design and develop a standalone TPG engine (outside fdreadoutlibs)		week 24	week 25	10	
	Integrate TPG engine		week 26	week 26	5	
	Provide ways to use different TPG algorithms by plane within current setup		week 36	week 37	20	
3	**Error conditions and mitigations**					
	Investigate TP splitting for long signals		week 34	week 35	10	
	Develop ways to monitor resource utilization of postprocessing threads		week 28	week 28	5	
	Handle backpressure from postprocessing threads in case of high TP rate		week 29	week 30	10	
4	**Operations and support (tools, docs, instructions)**					
	Automatic report generation of TP properties	E.Villa	week 10	week 12	5	100
	Tools to plot TP properties by plane	D. Pullia	week 10	week 12	5	100
	Development of validation tools with fixed patterns	I. Hristova; A. A	week 10	week 12	5	100
	Development of regression testing tools	I. Hristova	week 10	week 12	10	100
	Tools to perform TP emulation on raw data and produce emulation vs TP comparison		week 21	week 22	10	
	Scalability of TPG algorithms using NP04 APAs in EHN1 (update configur and pinning if needed)	M Man, A. Abed Abud	week 10	week 10	5	100
	Identification of problems and new features to be propagated (new requests)		week 14	week 32		
	Refinement of the tools to extract data and analyze TPs		week 14	week 32	5	
5	**Processing performance evaluation (in collab. with perf evaluation activity)**					
	Porting testing from DUNEWIB to WIBEth	A. Abed Abud	week 11	week 11	5	100
	Standalone performance evaluation (incl. measurements, reports) of WIBEth TPG (CPU util, avx instructions, etc.) on available CPU generations in EHN1	A. Abed Abud	week 11	week 11	5	40
	Hotspot analysis and profiling of TPG execution	A. Abed Abud	week 12	week 12	5	40
	Scalability and stress testing using synthetic benchmark	collab with perf group			5	100
	R&D on avx512 performance gain (low priority)		Week 46	week 47	10	

Points to Ponder

- RS:
  - AbsRS is overflow prone.
    - Add an overflow guard after division? For reference, $(|\texttt{sample}| * \texttt{scale})/10$.
    - Apply the same guard to SRS?
  - Does SRS become the standard instead of ST?
- `uint16_t` → `uint32_t`?