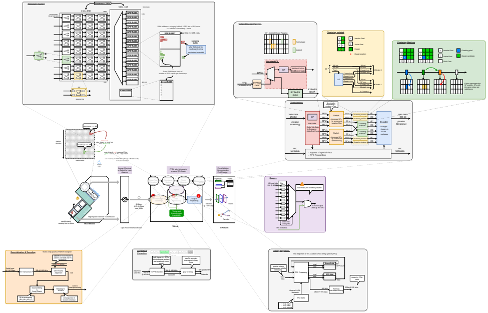
Readout firmware of the Vertex Locator

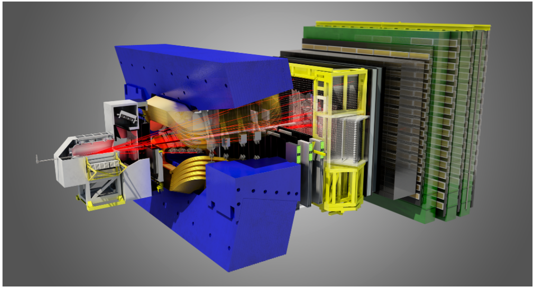
for LHCb Run 3 and beyond

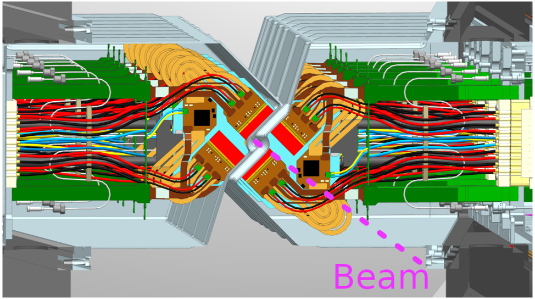
Karol Hennessy

on behalf of LHCb
University of Liverpool / CERN
October 14, 2020



LHCb VELO

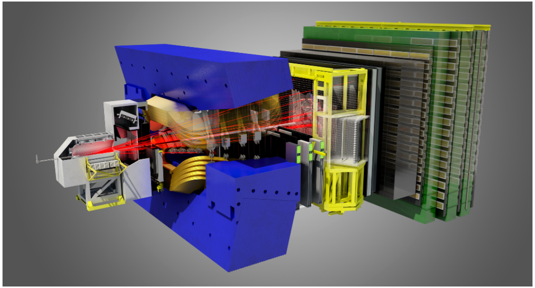


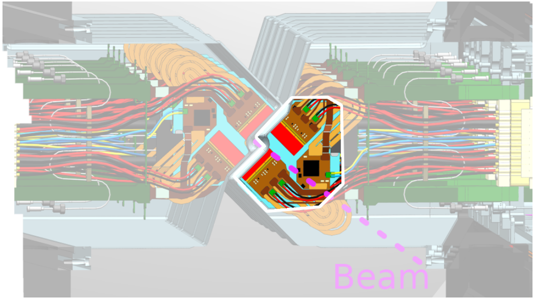


- LHCb - Flavour physics detector
- Excellent **vertexing** resolution and Particle ID
- LHCb has triggerless readout - full detector readout @ 40 MHz

- Vertex Locator (VELO)
- Silicon pixel modules around the LHC collision region
    - 50fb^{-1} integrated luminosity for LHC Runs 3 & 4
    - Very high radiation environment
    - In vacuum and under active cooling

LHCb VELO

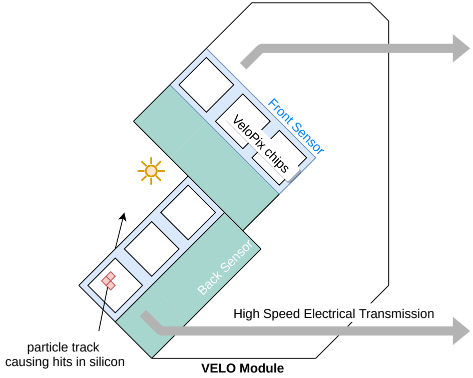




- LHCb - Flavour physics detector
- Excellent **vertexing** resolution and Particle ID
- LHCb has triggerless readout - full detector readout @ 40 MHz

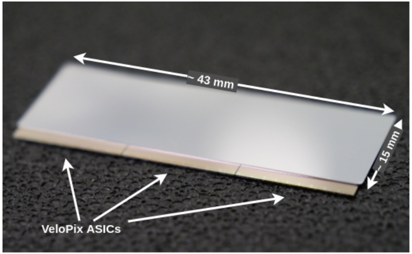
- Vertex Locator (VELO)
- Silicon pixel modules around the LHC collision region
  - 50fb^{-1} integrated luminosity for LHC Runs 3 & 4
  - Very high radiation environment
  - In vacuum and under active cooling

VELO Module

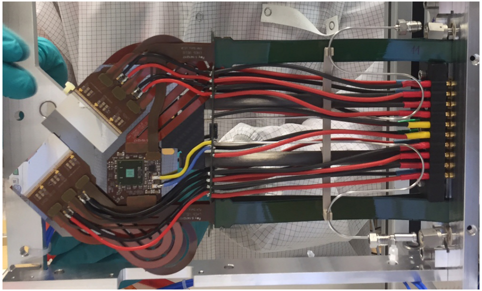


Front Sensor
VeloPix chips
Back Sensor
High Speed Electrical Transmission
particle track causing hits in silicon
VELO Module

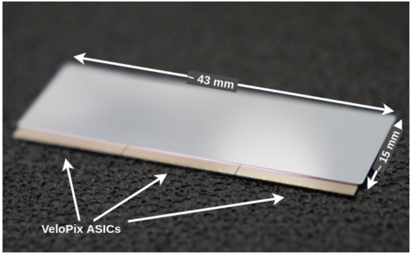
- Whole VELO = two halves of 26 modules
- Four sensors per module
    - 2 front
    - 2 back
- 3 VeloPix per sensor (i.e., 12 total)
- 20 high speed readout links
    - Chips closer to beam see more hits, and need more bandwidth



VELO Module

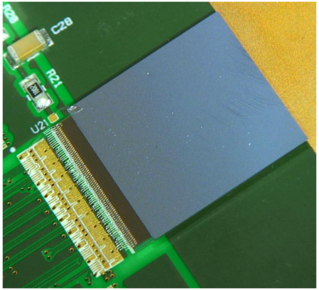


- Whole VELO = two halves of 26 modules
- Four sensors per module
  - 2 front
  - 2 back
- 3 VeloPix per sensor (i.e., 12 total)
- 20 high speed readout links
  - Chips closer to beam see more hits, and need more bandwidth

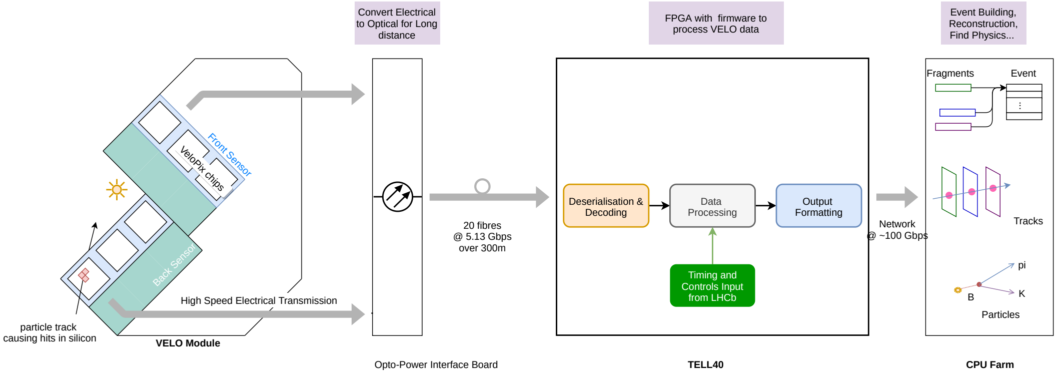


VeloPix ASIC

- Front-end ASIC driving the design of the VELO data acquisition system
- Part of the MediPix/TimePix family
- 130 nm CMOS technology
- 256×256 pixels of 55×55μm^2
- Clocked at 40 MHz
- Sends binary hit information (reducing bandwidth requirement)
  - full signal amplitude (ToT) available via slow readout for calibration



VELO Electronics and DAQ

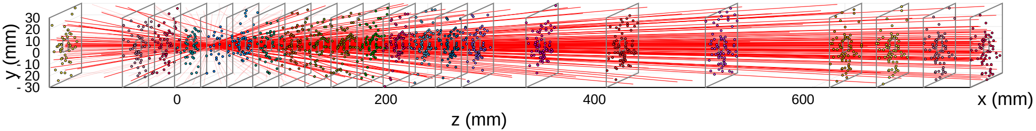


Convert Electrical to Optical for Long distance
FPGA with firmware to process VELO data
Event Building, Reconstruction, Find Physics...
Front Sensor
VeloPix chips
Back Sensor
particle track causing hits in silicon
VELO Module
High Speed Electrical Transmission
20 fibres @ 5.13 Gbps over 300m
Opto-Power Interface Board
Deserialisation & Decoding
Data Processing
Output Formatting
Timing and Controls Input from LHCb
TELL40
Network @ ~100 Gbps
Fragments
Event
Tracks
pi
K
B
Particles
CPU Farm

a slice of the VELO readout system

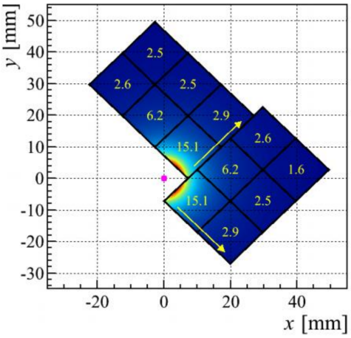
- See Flavio’s talk on Friday for a fuller description of the LHCb DAQ

Why an FPGA?



- *Lots of data!*
- VeloPix is optimised for high speed readout

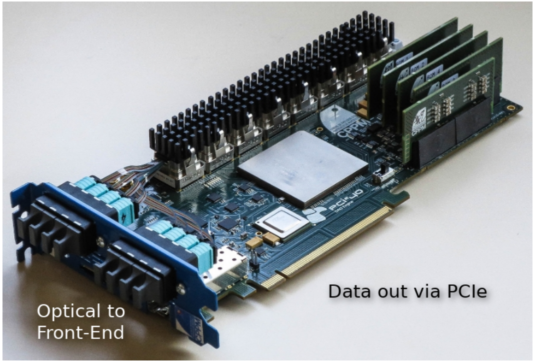
Peak hit rate	900 Mhits/s/ASIC
Max data rate	19.2 Gb/s
Total VELO	2.85 Tb/s



Data rate [Gbit/s] for hottest module.

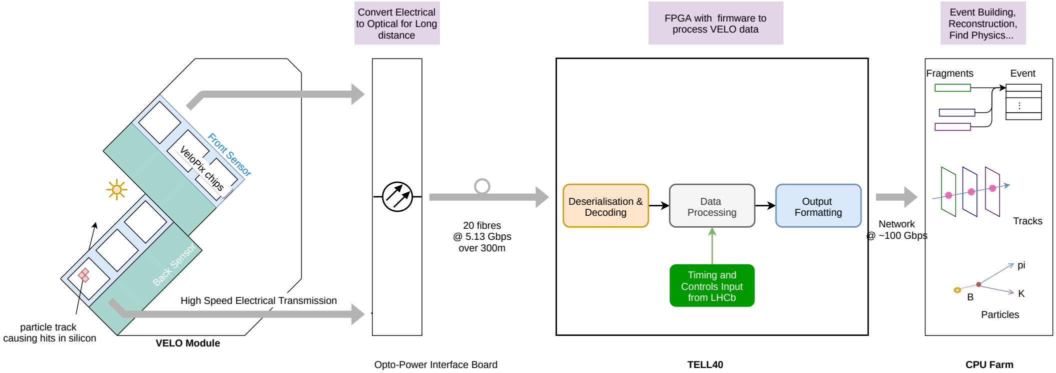
Readout Board - PCIe40/TELL40

- **Single control and readout board for the entire experiment**
- Can be used for Timing, Slow Control, DAQ or all
- Common hardware, shared firmware components
- PCIe Gen3 x16
- **Intel Arria10 FPGA** (10AX115S4F45E3SG)
- 1 TELL40 = 1 VELO module



- up to 48 bi-directional links @ $\sim$5 Gb/s
- Output bandwidth 100 Gb/s (measured).

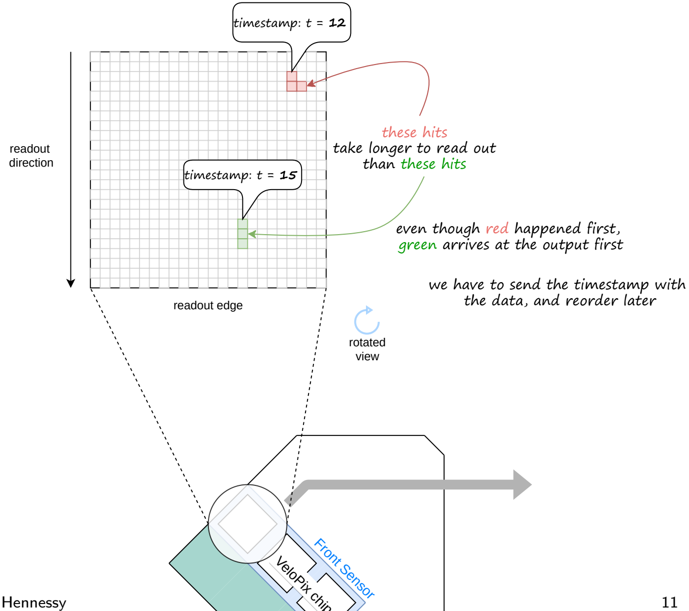
VELO Electronics and DAQ


Convert Electrical to Optical for Long distance
FPGA with firmware to process VELO data
Event Building, Reconstruction, Find Physics...
Front Sensor
VeloPix chips
Back Sensor
particle track causing hits in silicon
High Speed Electrical Transmission
VELO Module
20 fibres @ 5.13 Gbps over 300m
Opto-Power Interface Board
Deserialisation & Decoding
Data Processing
Output Formatting
Timing and Controls Input from LHCb
TELL40
Network @ ~100 Gbps
Fragments
Event
Tracks
pi
B
K
Particles
CPU Farm

So what does the VELO Firmware have to do?

What does VeloPix produce?

- **Time unordered data**



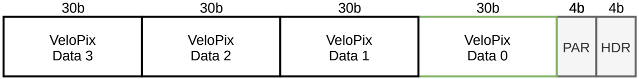
these hits take longer to read out than these hits

even though red happened first, green arrives at the output first

we have to send the timestamp with the data, and reorder later

What does VeloPix produce?

- Time unordered data
- **Custom transmission Protocol**



30b	30b	30b	30b	4b	4b
VeloPix Data 3	VeloPix Data 2	VeloPix Data 1	VeloPix Data 0	PAR	HDR

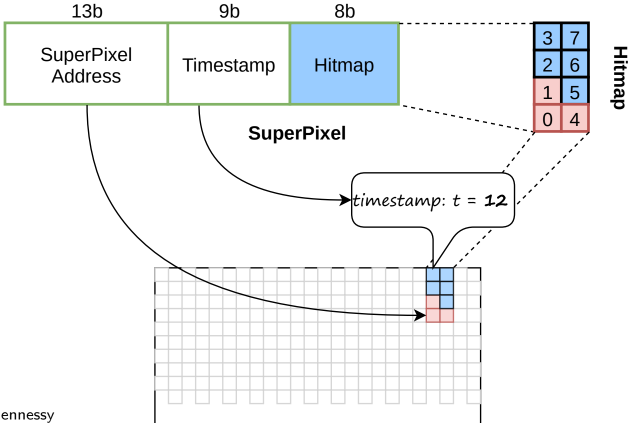
GWT Format

- Custom serializer - **Gigabit Wireline Transmitter (GWT)**
  - Chosen for low power - 60 mW
  - 5.12 Gb/s line rate (slightly higher than 4.8 Gb/s of G**B**T)
- GWT protocol
  - scrambled data (30 bit multiplicative)
  - parity check, no error recovery
  - low tolerance for header errors

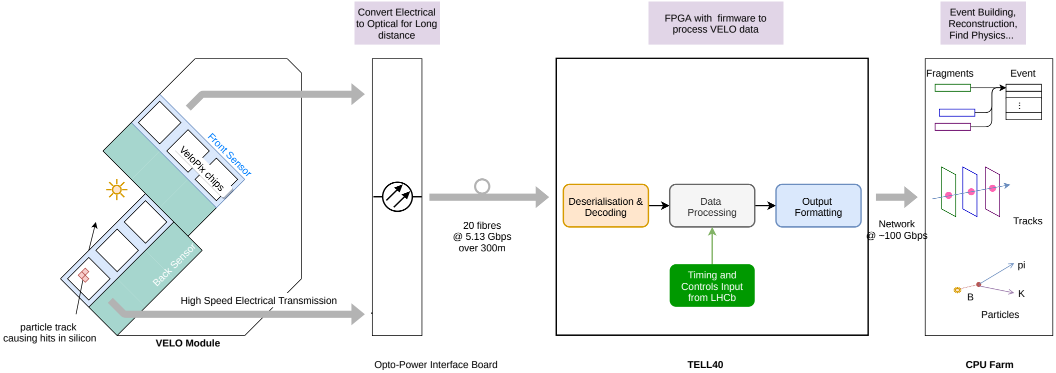
What does VeloPix produce?

- Time unordered data
- Custom transmission Protocol
- **SuperPixels**

- Pixel data is aggregated into groups of 2×4 called **SuperPixels**
  - 30% reduction in data size
- Timestamp stored in SuperPixel data packet



Readout Firmware



Convert Electrical to Optical for Long distance

FPGA with firmware to process VELO data

Event Building, Reconstruction, Find Physics...

Front Sensor

VeloPix chips

Back Sensor

particle track causing hits in silicon

High Speed Electrical Transmission

VELO Module

20 fibres
@ 5.13 Gbps
over 300m

Opto-Power Interface Board

Deserialisation & Decoding

Data Processing

Output Formatting

Timing and Controls Input from LHCb

TELL40

Network
@ ~100 Gbps

Fragments

Event

Tracks

pi

K

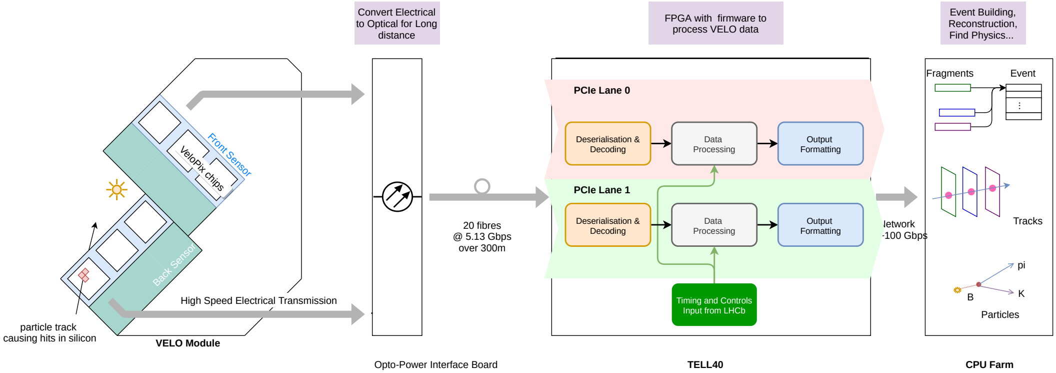
B

Particles

CPU Farm

What’s in the TELL40 Firmware?

Readout Firmware



Convert Electrical to Optical for Long distance

FPGA with firmware to process VELO data

Event Building, Reconstruction, Find Physics...

Front Sensor

VeloPix chips

Back Sensor

particle track causing hits in silicon

High Speed Electrical Transmission

VELO Module

20 fibres @ 5.13 Gbps over 300m

Opto-Power Interface Board

PCIe Lane 0

Deserialisation & Decoding

Data Processing

Output Formatting

PCIe Lane 1

Deserialisation & Decoding

Data Processing

Output Formatting

Timing and Controls Input from LHCb

TELL40

letwork
-100 Gbps

Fragments

Event

Tracks

pi

B

K

Particles

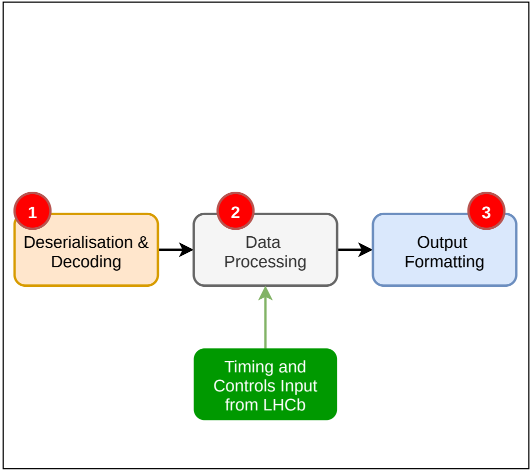
CPU Farm

- Actually, it's two parallel streams for PCIe bandwidth optimisation
- But it's simpler to describe just one

Handling VeloPix data

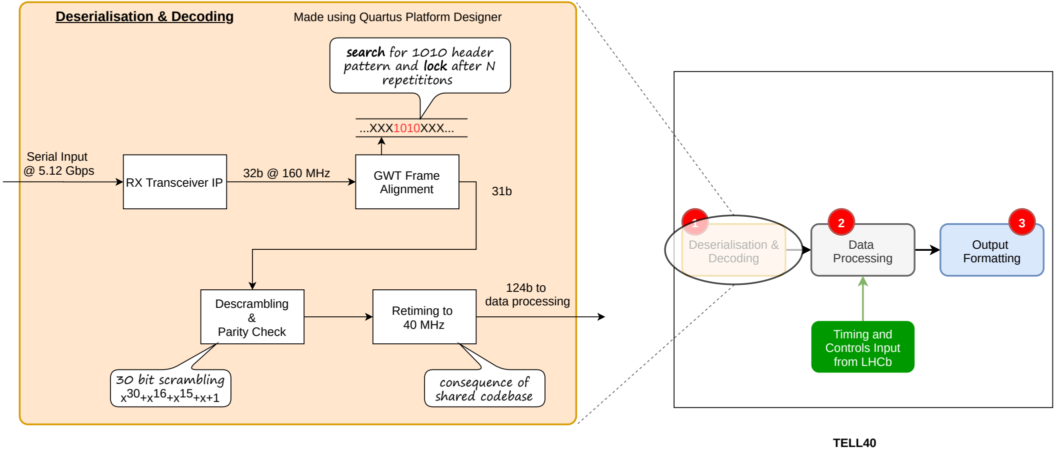
Going back to our list:

1. Custom transmission Protocol
2. Time unordered data SuperPixels
3. *is a generic component and won't be discussed here*


1
Deserialisation & Decoding
2
Data Processing
3
Output Formatting
Timing and Controls Input from LHCb

TELL40

Handling VeloPix data - Deserialisation & Decoding



Deserialisation & Decoding

Made using Quartus Platform Designer

search for 1010 header pattern and lock after N repetititons

...XXX1010XXX...

Serial Input @ 5.12 Gbps

RX Transceiver IP

32b @ 160 MHz

GWT Frame Alignment

31b

Descrambling & Parity Check

Retiming to 40 MHz

124b to data processing

30 bit scrambling $x^{30}+x^{16}+x^{15}+x+1$

consequence of shared codebase

1 Deserialisation & Decoding

2 Data Processing

3 Output Formatting

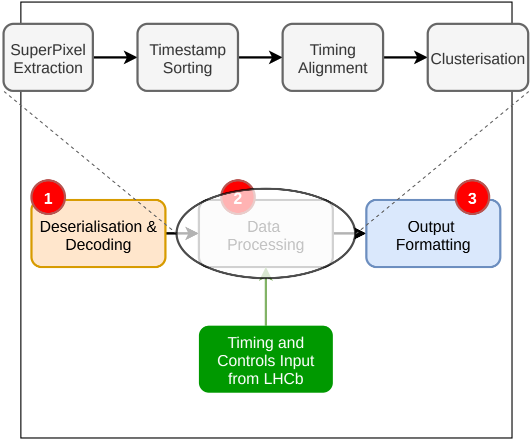
Timing and Controls Input from LHCb

TELL40

Handling VeloPix data

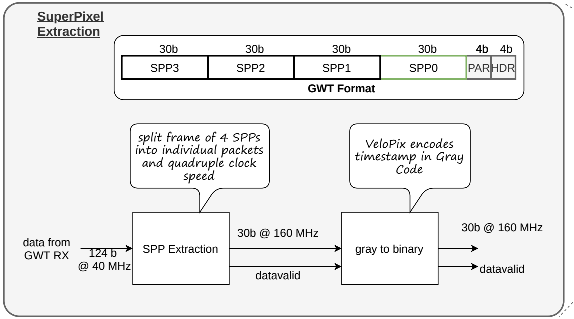
Going back to our list:

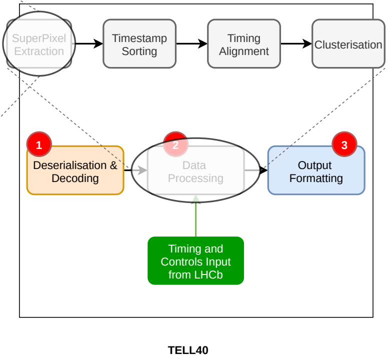
1. Custom transmission Protocol
2. Time unordered data SuperPixels
3. *is a generic component and won't be discussed here*



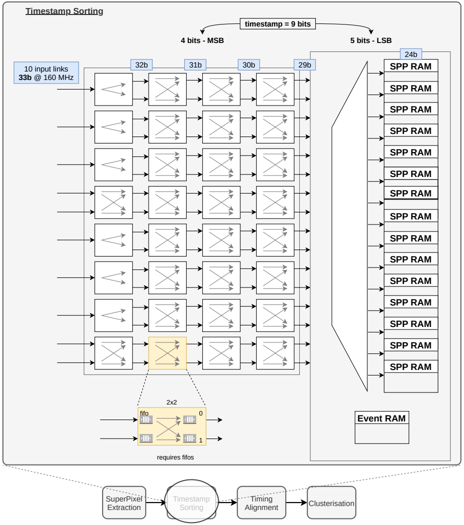
TELL40

Handling VeloPix data - SuperPixel Extraction



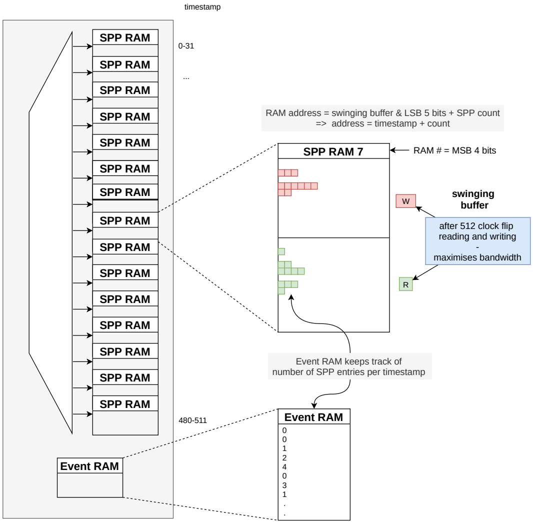


Handling VeloPix data - Time Reordering



- Timestamps are sorted 1 bit at a time in several layers
    - First column is MSB...
- Fifos are needed to avoid collisions
- Data are stored in RAMs at the end of the routing
- The whole reordering consumes a large amount of the FPGAs memory

Handling VeloPix data - Time Reordering



timestamp

0-31

...

480-511

SPP RAM

Event RAM

RAM address = swinging buffer & LSB 5 bits + SPP count
=> address = timestamp + count

SPP RAM 7

RAM # = MSB 4 bits

W

swinging buffer

after 512 clock flip reading and writing
-
maximises bandwidth

R

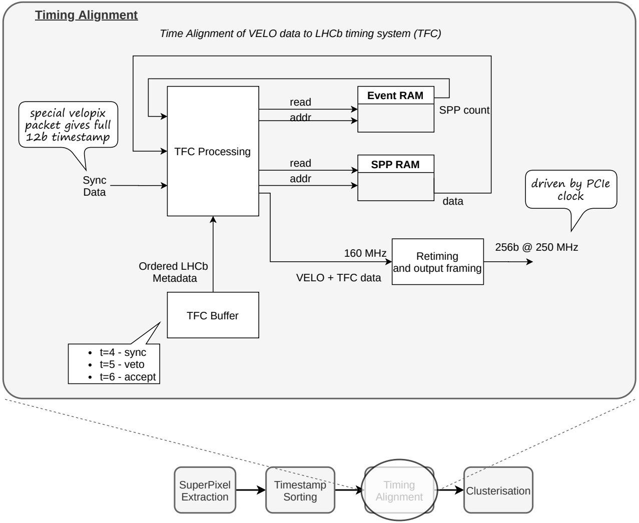
Event RAM keeps track of number of SPP entries per timestamp

Event RAM

0
0
1
2
4
0
3
1
.
.

- After the routing, the RAM address is equivalent to the timestamp
- SuperPixel Packets are stored for a maximum latency of 512 clock cycles
- A swinging buffer is used to maximise bandwidth

Handling VeloPix data - Time Alignment



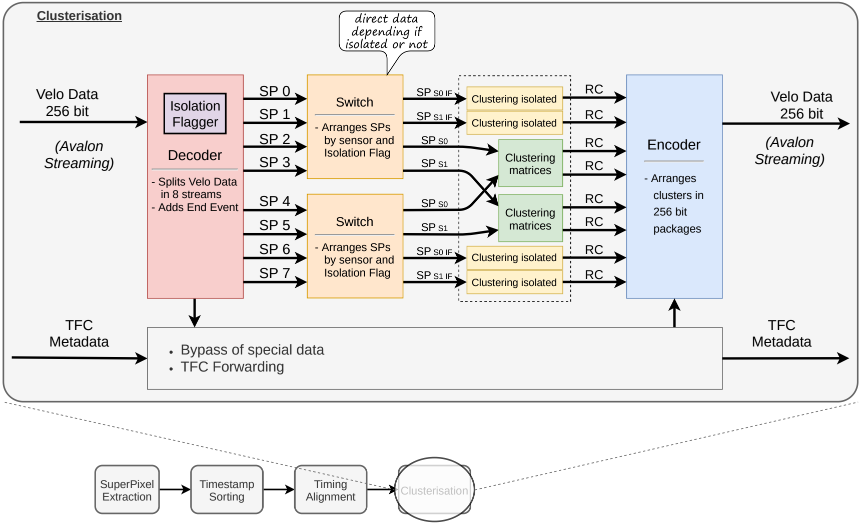
Timing Alignment

Time Alignment of VELO data to LHCb timing system (TFC)

- t=4 - sync
- t=5 - veto
- t=6 - accept

- After reordering, data must align to the rest of LHCb
- **Timing and Fast Control** (TFC) system provides LHCb timing metadata

Handling VeloPix data - Clusterisation



Clusterisation

direct data depending if isolated or not

Velo Data 256 bit

(Avalon Streaming)

Isolation Flagger

Decoder

- Splits Velo Data in 8 streams
- Adds End Event

SP 0

SP 1

SP 2

SP 3

SP 4

SP 5

SP 6

SP 7

Switch

- Arranges SPs by sensor and Isolation Flag

Switch

- Arranges SPs by sensor and Isolation Flag

SP S0 IF

SP S1 IF

SP S0

SP S1

SP S0

SP S1

SP S0 IF

SP S1 IF

Clustering isolated

Clustering isolated

Clustering matrices

Clustering matrices

Clustering isolated

Clustering isolated

RC

RC

RC

RC

RC

RC

RC

RC

Encoder

- Arranges clusters in 256 bit packages

Velo Data 256 bit

(Avalon Streaming)

TFC Metadata

- Bypass of special data
- TFC Forwarding

TFC Metadata

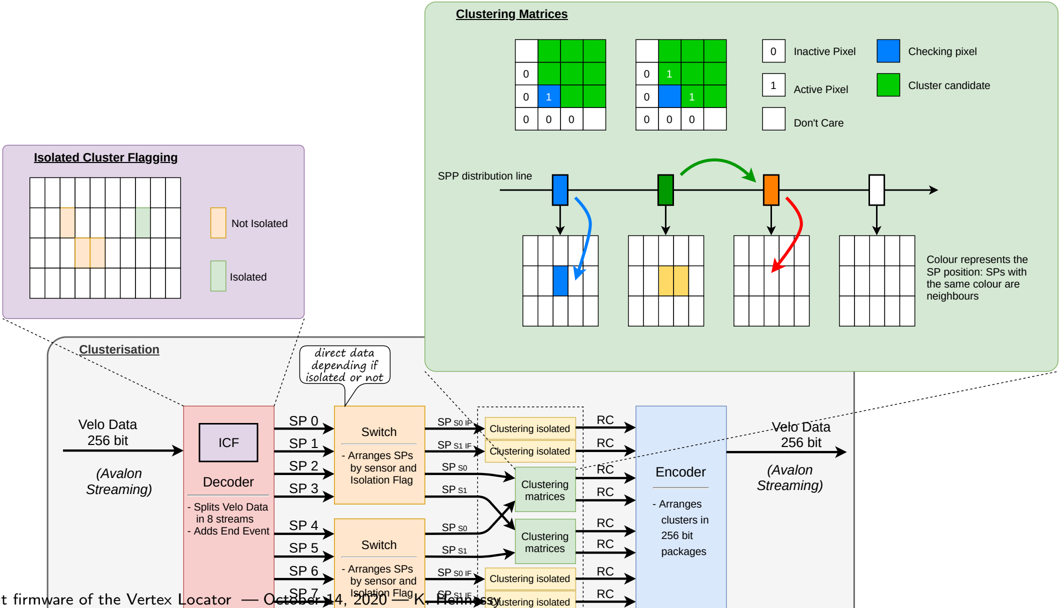
SuperPixel Extraction

Timestamp Sorting

Timing Alignment

Clusterisation

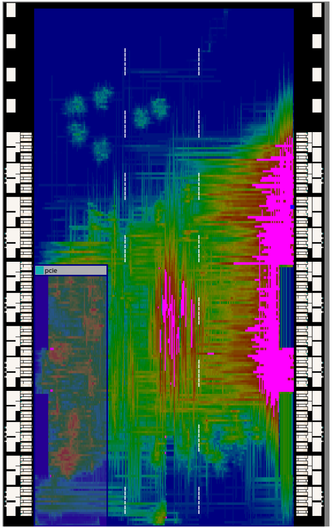
Handling VeloPix data - Clusterisation



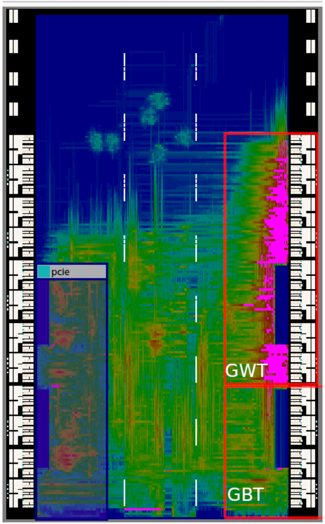
Challenges

Deserialisation & Decoding - Congestion

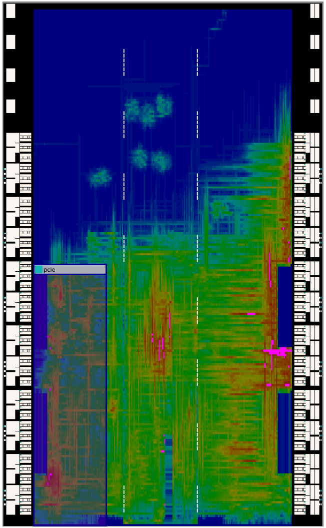
v1



v2 Improved frame aligner

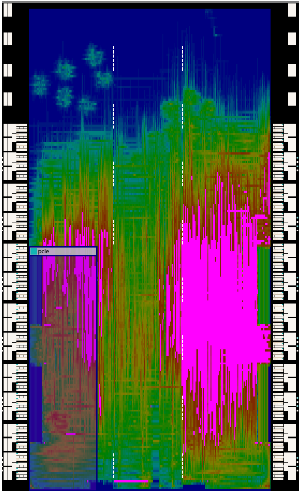


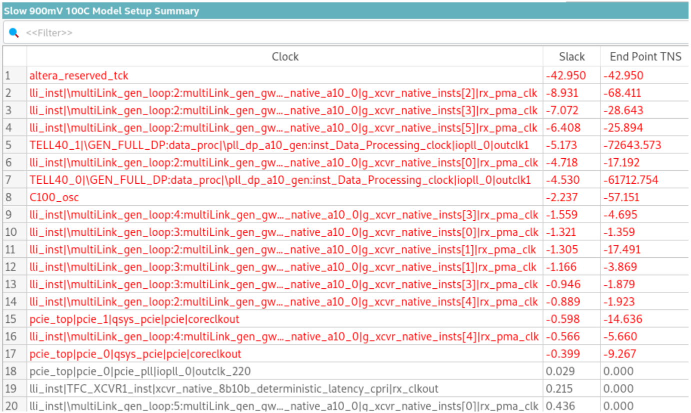
v3 Improved bit slipping



Full Data Processing

Now adding the full Time Reordering and Alignment





Slow 900mV 100C Model Setup Summary

<<Filter>>

	Clock	Slack	End Point TNS
1	altera_reserved_tck	-42.950	-42.950
2	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[2]|rx_pma_clk	-8.931	-68.411
3	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[3]|rx_pma_clk	-7.072	-28.643
4	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[5]|rx_pma_clk	-6.408	-25.894
5	TELL40_1|\GEN_FULL_DP:data_proc|\pll_dp_a10_gen:inst_Data_Processing_clock|iopll_0|outclk1	-5.173	-72643.573
6	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[0]|rx_pma_clk	-4.718	-17.192
7	TELL40_0|\GEN_FULL_DP:data_proc|\pll_dp_a10_gen:inst_Data_Processing_clock|iopll_0|outclk1	-4.530	-61712.754
8	C100_osc	-2.237	-57.151
9	lli_inst|\multiLink_gen_loop:4:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[3]|rx_pma_clk	-1.559	-4.695
10	lli_inst|\multiLink_gen_loop:3:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[0]|rx_pma_clk	-1.321	-1.359
11	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[1]|rx_pma_clk	-1.305	-17.491
12	lli_inst|\multiLink_gen_loop:3:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[1]|rx_pma_clk	-1.166	-3.869
13	lli_inst|\multiLink_gen_loop:3:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[3]|rx_pma_clk	-0.946	-1.879
14	lli_inst|\multiLink_gen_loop:2:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[4]|rx_pma_clk	-0.889	-1.923
15	pcie_top|pcie_1|qsys_pcie|pcie|coreclkout	-0.598	-14.636
16	lli_inst|\multiLink_gen_loop:4:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[4]|rx_pma_clk	-0.566	-5.660
17	pcie_top|pcie_0|qsys_pcie|pcie|coreclkout	-0.399	-9.267
18	pcie_top|pcie_0|pcie_pll|iopll_0|outclk_220	0.029	0.000
19	lli_inst|TFC_XCVR1_inst|xcvr_native_8b10b_deterministic_latency_cpri|rx_clkout	0.215	0.000
20	lli_inst|\multiLink_gen_loop:5:multiLink_gen_gw..._native_a10_0|g_xcvr_native_insts[0]|rx_pma_clk	0.436	0.000

Timing closure becomes tricky

Resource Estimate

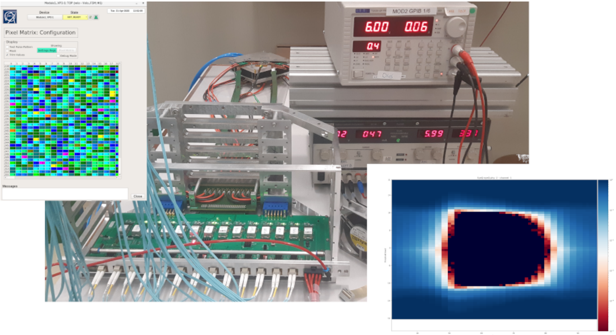
- Very preliminary estimate for now (sum of individual compilations - not the output of a complete build)

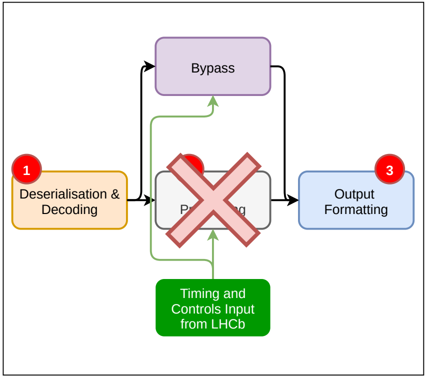
	Logic (ALMs)	M20K RAMs
Timestamp Reordering	39	72
Clustering	31	11
Total	70	83

- Looks like it will fit, but **congestion** and **timing closure** are the major challenges ahead

A working firmware...

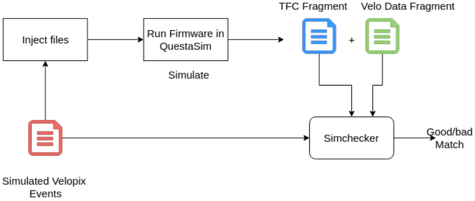
- Full data processing is not complete today
- A bypass is used for production and testing
  - Sorting/processing is done on CPU
  - Rate limited
- Can also be used to check data processing (send same data to both and check)

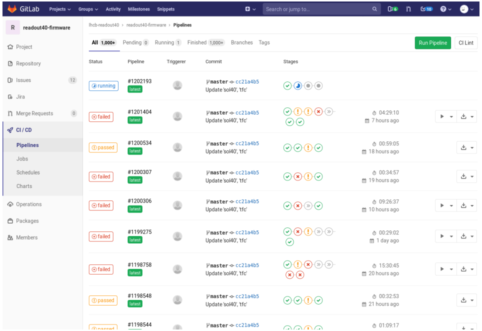



Bypass
1
Deserialisation & Decoding
Pr
g
3
Output Formatting
Timing and Controls Input from LHCb

Tools/Organisation

- Typical combination of Questasim and Quartus
- LHCb employs gitlab pipelines for checking new releases
  - Sim-checker injects files into firmware and verifies on output
  - Can add additional testbenches and cross-checks à la “nightlies”
  - Strict versioning and tracking
- VELO makes stable releases for production testing





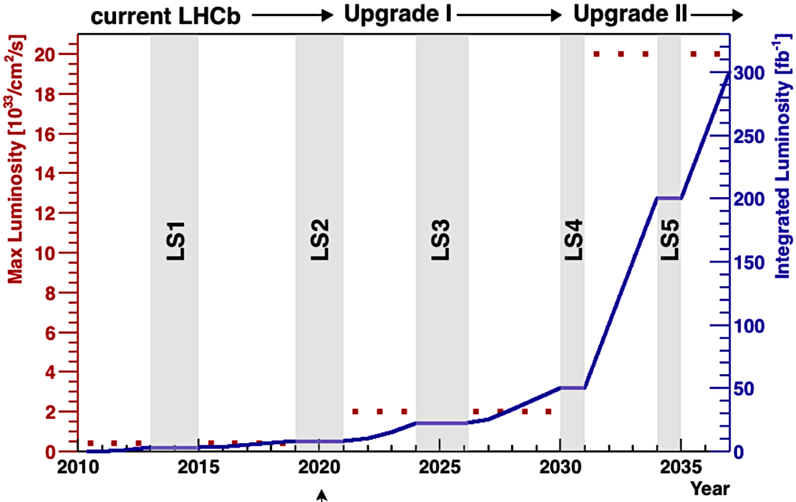
lhcb-readout40 › readout40-firmware › Pipelines

All 1,000+ | Pending 0 | Running 1 | Finished 1,000+ | Branches | Tags

Status	Pipeline	Triggerer	Commit	Stages	
running	#1202193 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ◔ ● ●	
failed	#1201404 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ! ! ✗ » ✓ ✓	04:29:10 7 hours ago
passed	#1200534 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ✓ ! ✓	00:59:05 18 hours ago
failed	#1200307 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ✗ ! ✓	00:34:57 19 hours ago
failed	#1200306 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ✗ » ✓	09:26:37 10 hours ago
failed	#1199275 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ✗ ! » » ✓	00:29:02 1 day ago
failed	#1198758 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ! ✗ » » ✗ ✗	15:30:45 20 hours ago
passed	#1198548 latest		master cc21a4b5 Update 'sol40', 'tfc'	✓ ✓ ! ✓	00:32:53 21 hours ago
passed	#1198544		master cc21a4b5	✓ ✓ ! ✓	01:09:17

…and beyond

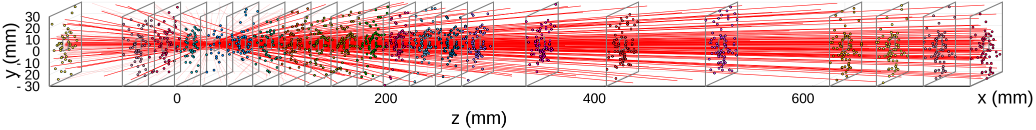
Timeline



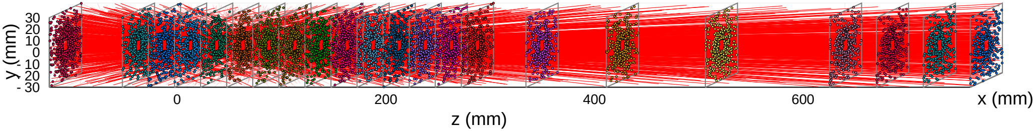
You are here

VELO U2

- HL-LHC (2028) will provide 7.5× luminosity
- Meaning 7.5× tracks/hits...
- Meaning we need a new VELO to go from this

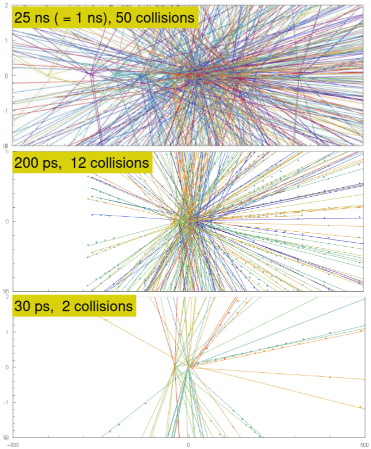


- to this



...and beyond

- Add extra timing precision (necessary for vertex/tracking)
- Bandwidth increase $O(10)$



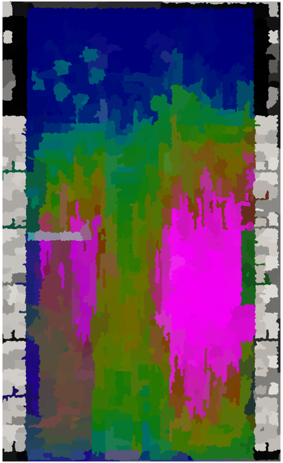
	Arria10	Agilex*	Factor Increase
Process	20 nm	10 nm	~ 2
Logic Elements (k)	1150	2692	2.3
M20k Memory (Mb)	53	259	4.9
DSP	1518	17056*	~ 16

Table: Comparison of FPGA resources for VELO U1b and a candidate for U2.

- Next gen FPGAs not quite scaling with the needs of the experiment!
- What can we do with all these DSPs?

Concluding Remarks

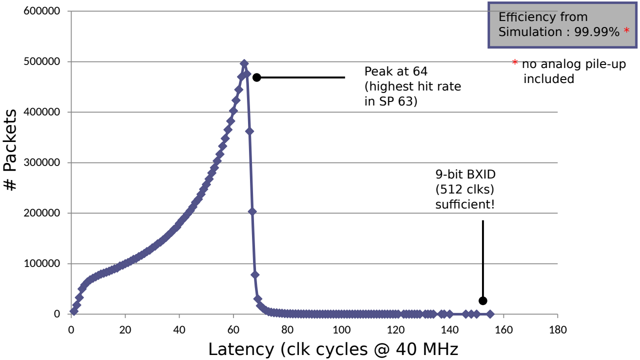
- LHCb VELO firmware on track to process VeloPix data
- Validating Time Reordering and Clustering
- Several challenges in terms of FPGA resources and timing closure
  - ○ Confident we can solve these
  - ○ We welcome any clever suggestions/tips
- Learning techniques to optimise the next generation of the experiment
- Need to adapt to the changing landscape of heterogenous computing



backup

BXID Router

- Time-ordering SuperPixel data
  - 9-bit **router** sorts data 1 bit at a time
  - Extensive simulation required - both to maximise speed (>160 MHz) and minimise FPGA resource usage
  - Latency limit < 512 clock cycles

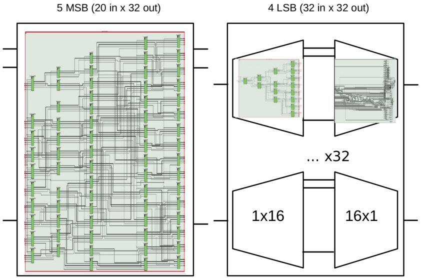


Efficiency from Simulation : 99.99% *

Peak at 64 (highest hit rate in SP 63)

* no analog pile-up included

9-bit BXID (512 clks) sufficient!

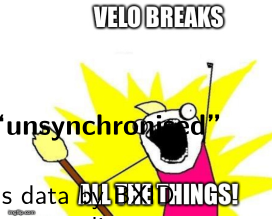


Special VeloPix

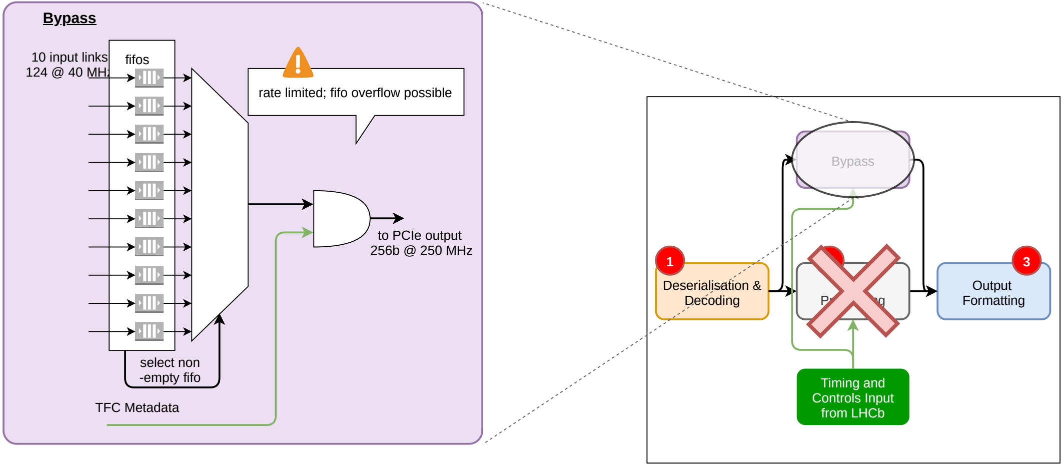
A lot of non-standard DAQ elements…

- VeloPix has **NO SCA**
  - SLVS communication component required
  - extra SOL40 firmware
  - extra SOL40 software
- VeloPix **does NOT use GBT** for DAQ
  - uses GWT
  - different frequency - 5.12 Gbps
  - special VELO LLI firmware component
  - special firmware decoding, clocking…
  - special VELO LLI software

- VeloPix sends data **“unsynchronised”**
  - Firmware re-aligns data by BXID
  - Cannot filter events pre-alignment
  - Special dataflow monitoring needed
- Big effort from Online, Annecy, Marseille to help integrate into the standard firmware and software. Must remain vigilent and ensure “special cases” are tested as standard.

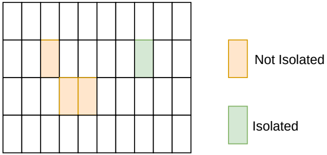


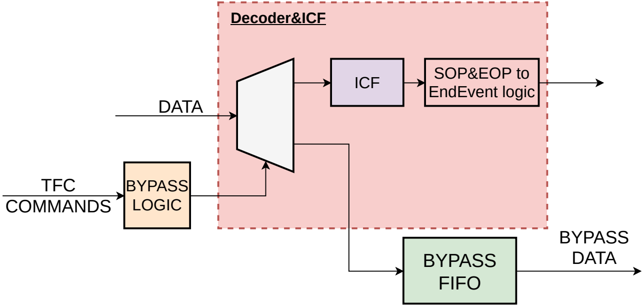
Bypass detail



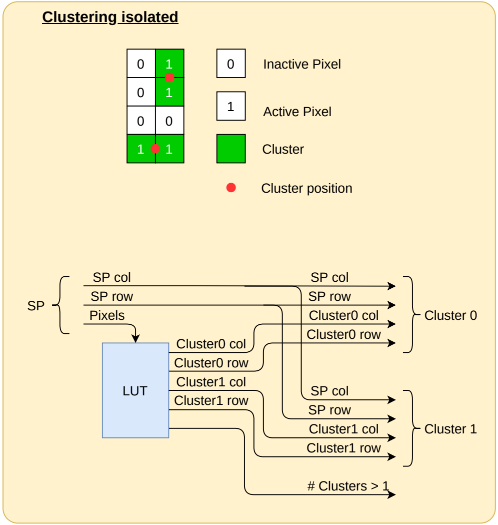
Isolated Cluster Flagging

ICF - Isolated Cluster Flagging





Isolated Clustering



Clustering isolated

0	1
0	1
0	0
1	1

0 Inactive Pixel

1 Active Pixel

🟩 Cluster

● Cluster position

SP: SP col, SP row, Pixels

LUT

Cluster0 col

Cluster0 row

Cluster1 col

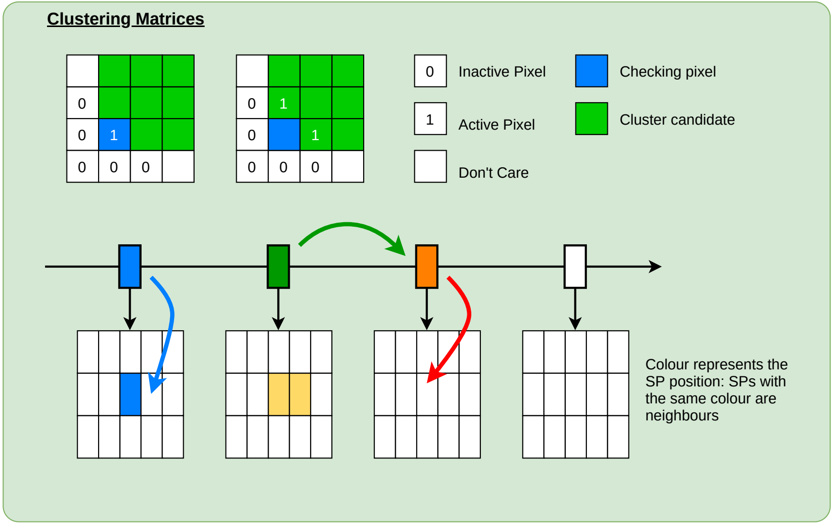
Cluster1 row

Cluster 0: SP col, SP row, Cluster0 col, Cluster0 row

Cluster 1: SP col, SP row, Cluster1 col, Cluster1 row

# Clusters > 1

Clustering Matrices



Clustering Matrices

- 0 Inactive Pixel
- 1 Active Pixel
- Don't Care
- 🟦 Checking pixel
- 🟩 Cluster candidate

Colour represents the SP position: SPs with the same colour are neighbours