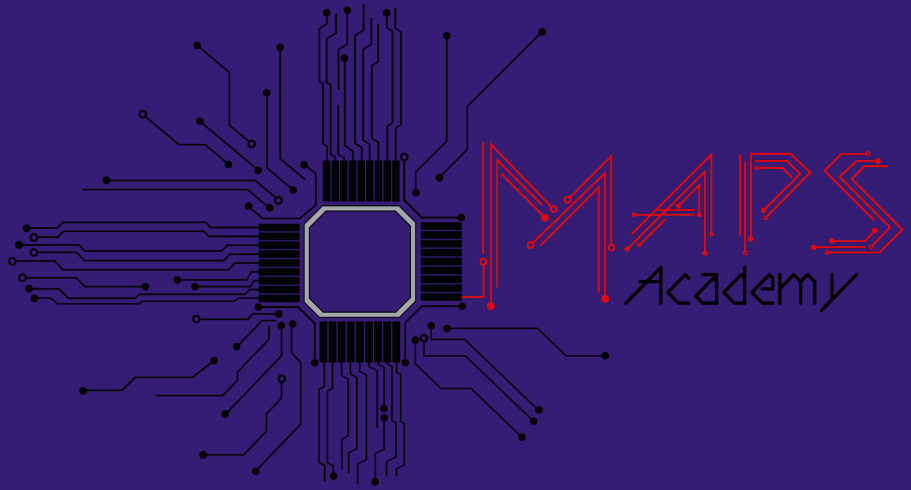
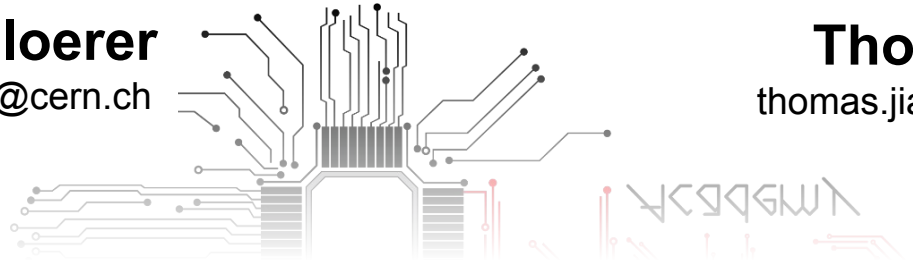


Group B's PixESL Overview

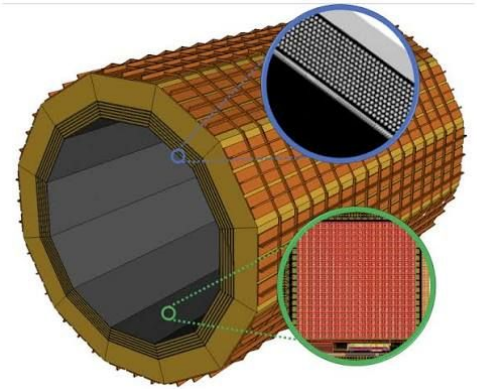
Eduardo Ploerer
eduardo.ploerer@cern.ch

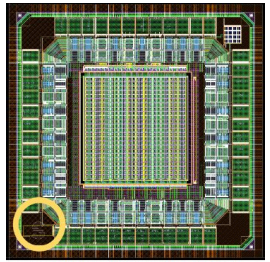
Thomas Wu
thomas.jiayu.wu@cern.ch



Motivation

- ➔ Modern MAPS detectors contain millions of pixels, and they are complete electronic systems.
- ➔ Even if the sensor works perfectly, the detector performance also depends on the readout electronics.
- ➔ PixESL is an Electronic System-Level (ESL) framework built on SystemC that lets us model these electronics before hardware is fabricated.





Monolithic Active Pixel Sensors. Source: https://indico.fnal.gov/event/22303/contributions/245974/attachments/157387/205863/MAPS.pdf

How PixESL Fits into MAPS

PixESL allows us to model the complete electronic readout chain, from an individual pixel to an entire detector network.

➔ **Front-End**

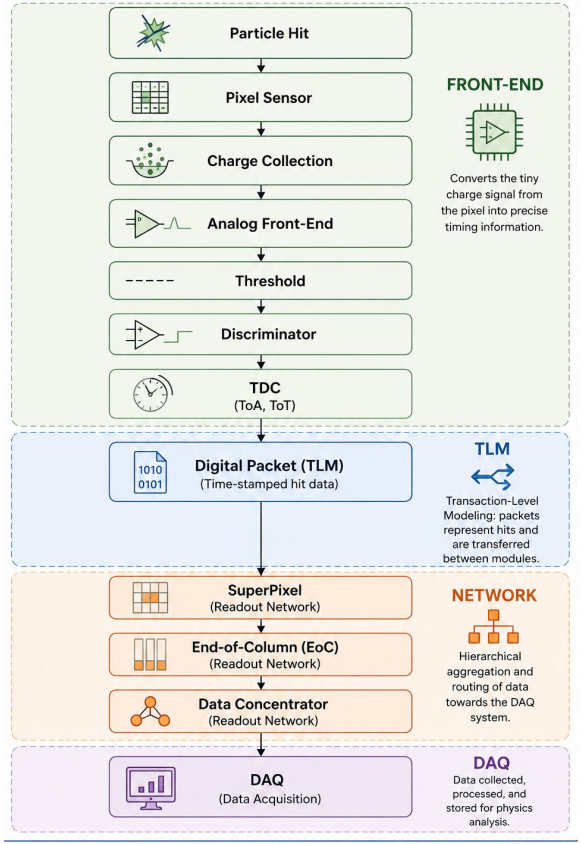
- The electronics inside each individual pixel that convert collected charge into digital information.

➔ **TLM (Transaction-Level Modeling)**

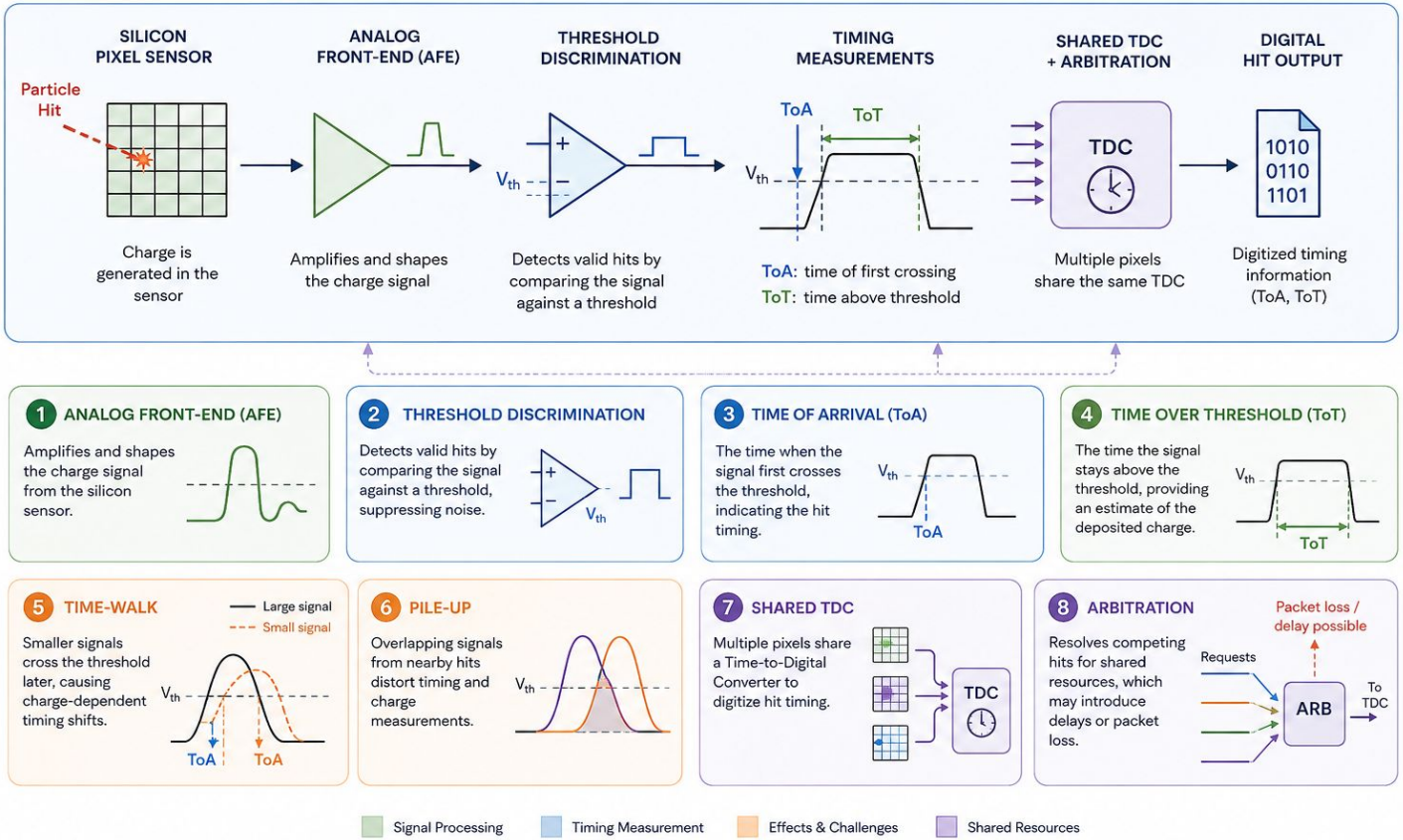
- The communication of digital hit data between detector modules using high-level SystemC transactions instead of individual wires or signals.

➔ **Network**

- The complete detector readout architecture, built by connecting many Front-End and TLM modules into a scalable detector model.

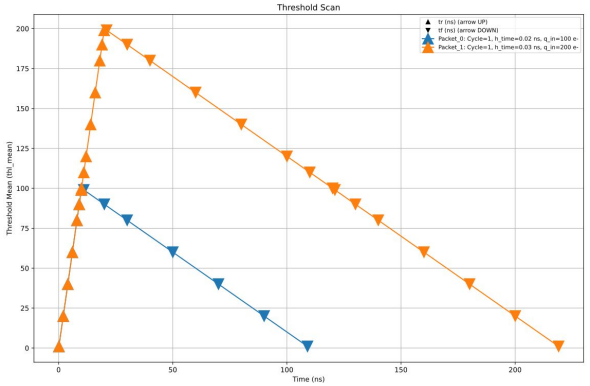


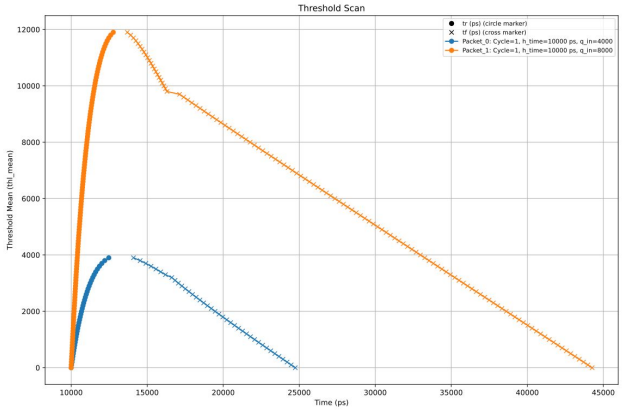
Front-End


SILICON PIXEL SENSOR
ANALOG FRONT-END (AFE)
THRESHOLD DISCRIMINATION
TIMING MEASUREMENTS
SHARED TDC + ARBITRATION
DIGITAL HIT OUTPUT
Particle Hit
Charge is generated in the sensor
Amplifies and shapes the charge signal
Detects valid hits by comparing the signal against a threshold
ToA
ToT
Vth
ToA: time of first crossing
ToT: time above threshold
TDC
Multiple pixels share the same TDC
1010
0110
1101
Digitized timing information (ToA, ToT)
1 ANALOG FRONT-END (AFE)
Amplifies and shapes the charge signal from the silicon sensor.
2 THRESHOLD DISCRIMINATION
Detects valid hits by comparing the signal against a threshold, suppressing noise.
3 TIME OF ARRIVAL (ToA)
The time when the signal first crosses the threshold, indicating the hit timing.
4 TIME OVER THRESHOLD (ToT)
The time the signal stays above the threshold, providing an estimate of the deposited charge.
5 TIME-WALK
Large signal
Small signal
Smaller signals cross the threshold later, causing charge-dependent timing shifts.
6 PILE-UP
Overlapping signals from nearby hits distort timing and charge measurements.
7 SHARED TDC
Multiple pixels share a Time-to-Digital Converter to digitize hit timing.
8 ARBITRATION
Packet loss / delay possible
Requests
ARB
To TDC
Resolves competing hits for shared resources, which may introduce delays or packet loss.
Signal Processing
Timing Measurement
Effects & Challenges
Shared Resources

Front-End - Threshold Scan

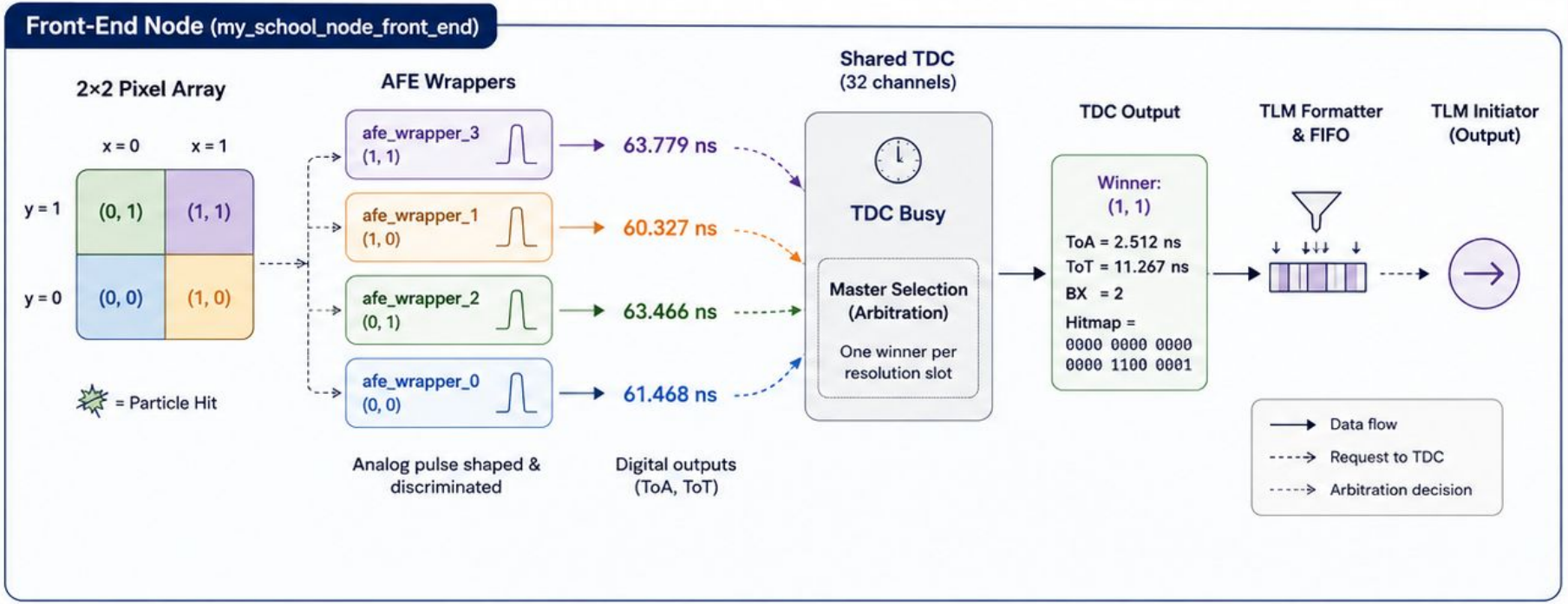
- ➔ Implemented a simple triangular analog pulse model and discriminator
  - ◆ Time of Arrival (ToA) from the rising-edge threshold crossing (tr)
  - ◆ Time over Threshold (ToT) from the duration above threshold (tf-tr)
- ➔ Signals with larger amplitude stay above the threshold longer, producing a larger ToT.
- ➔ The Leading edge provides timing information, while ToT provides information related to signal size (charge).
- ➔ Real detector electronics do not produce ideal triangle pulses. When two hits arrive close together, their analog signals overlap (pile-up).
  - ◆ Changes the measured timing and Time-over-Threshold (ToT).





Front-End - Digitizing Hits

From result.log:



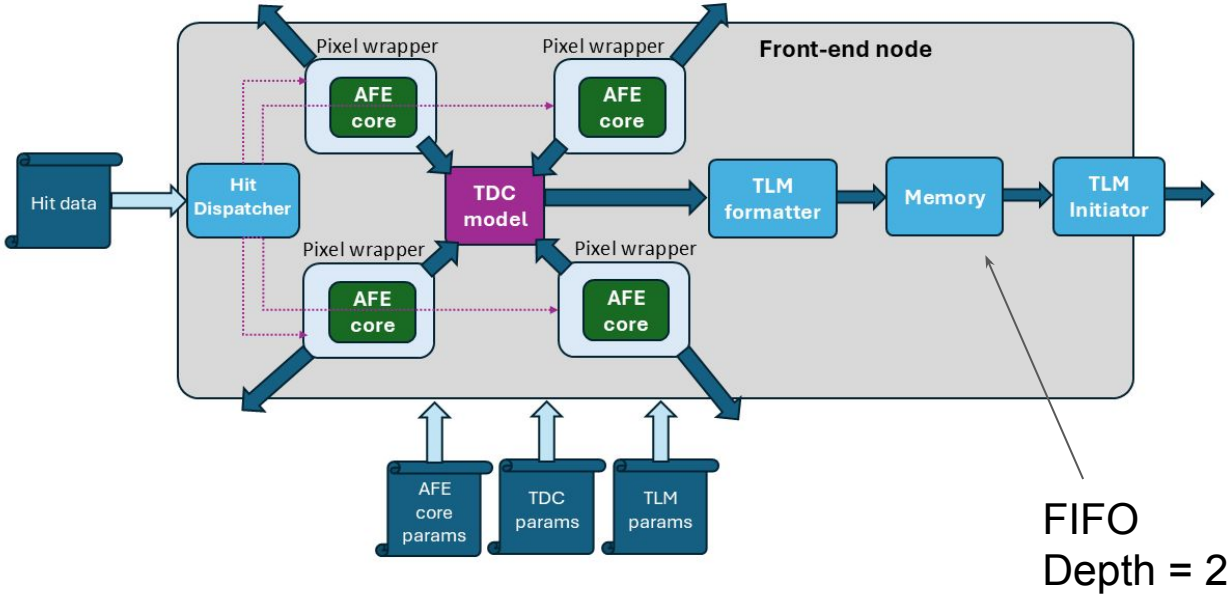
TLM - TDC Output & TLM Packet

TDC Output (tdc_output.csv) → **TLM Formatter** → **TLM Packet (tlm_formatter_memory_in.csv)**

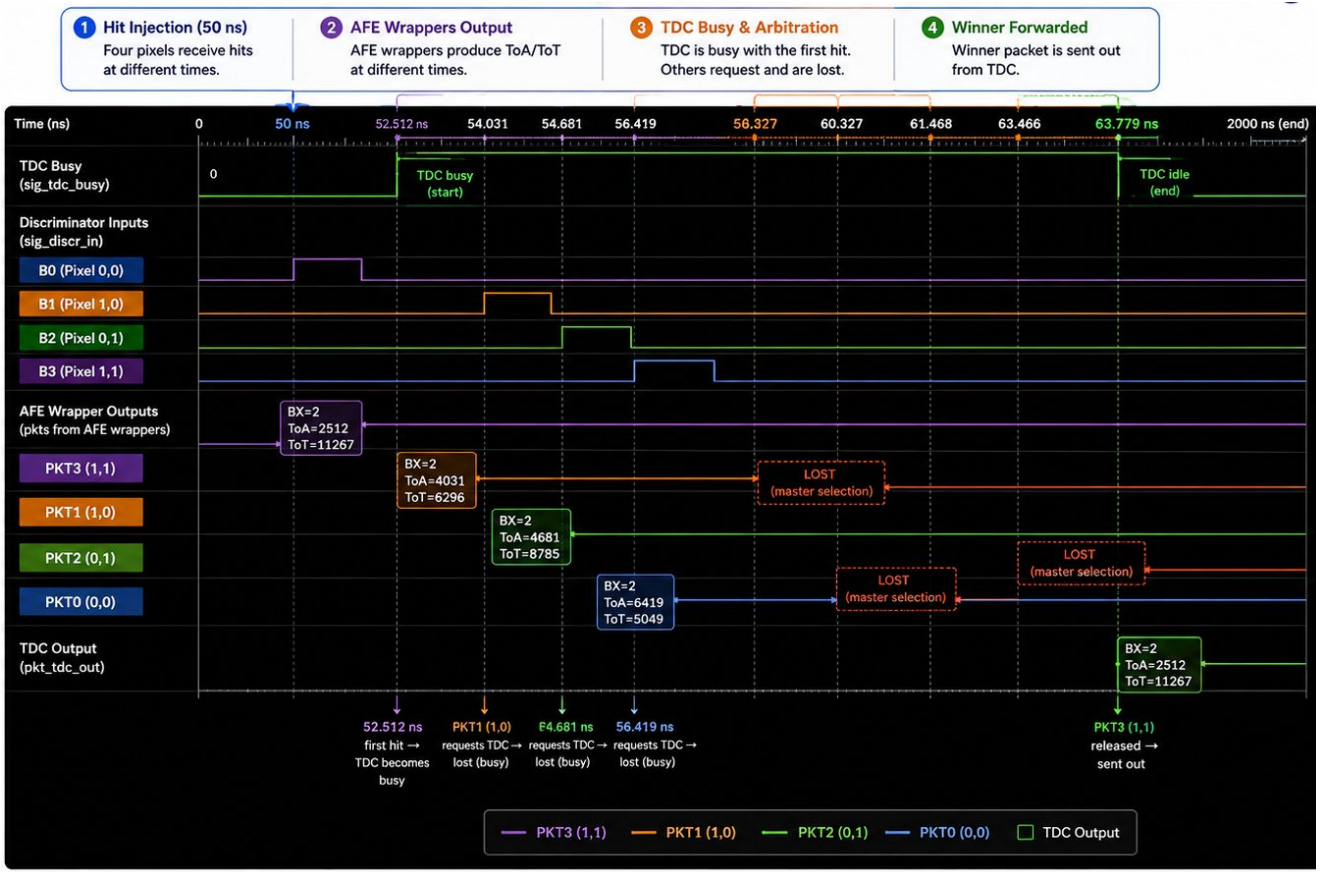
Variable	Value	Meaning
BX	2	Event belongs to bunch crossing 2
COL	1	Pixel column
ROW	1	Pixel row
TOA	2512	Measured arrival time (ps)
TOT	11267	Time over threshold or pulse width (ps)
HITMAP	0000000000000000 11000001	Pixels active in this event

Variable	Value
BX	2
COL	1
ROW	1
LOG_TIME	75000

TLM Architecture

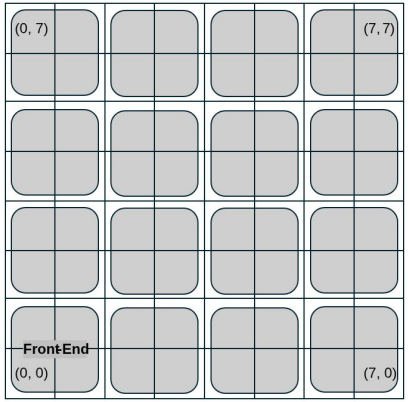


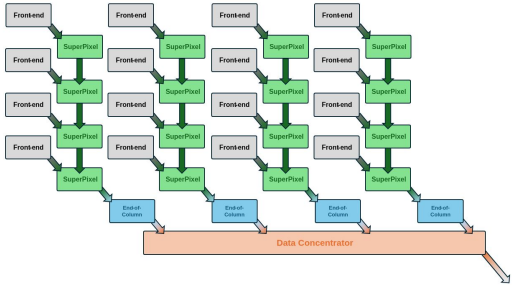
TLM Architecture



Network

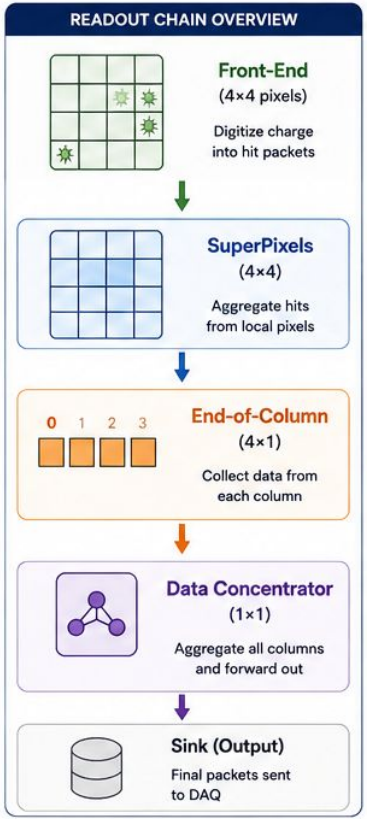
- **Pixel columns**
  - Organize pixels into vertical groups that share local readout resources.
- **Pixel matrices**
  - Combine many pixel columns into a complete two-dimensional sensor array.
- **SuperPixels**
  - Group neighboring pixels into a single readout unit to reduce data traffic and simplify routing.
- **End-of-Column (EoC) blocks**
  - Collect and buffer data from each pixel column before forwarding it to higher-level readout.
- **Data Concentrators**
  - Aggregate data from multiple EoC blocks and transmit it to the detector readout or DAQ system.



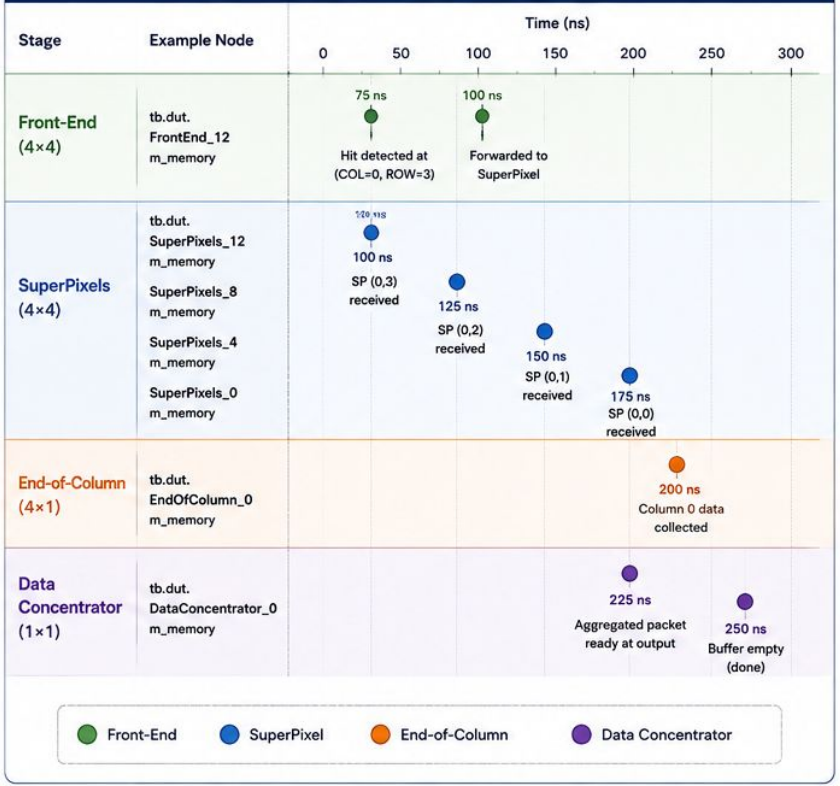


Network - Pixel Matrix

READOUT CHAIN OVERVIEW



PACKET PROPAGATION TIMELINE (From occupancy.csv)



1) FRONT-END → SUPERPIXEL INPUT

Front-End → SuperPixel input

File: packets_SuperPixels_IN_from_children.csv

BX	COL	ROW	LOG_TIME	LOG_COL	LOG_ROW
2	1	7	100000	0	3

At 100000 ps (100 ns), the hit (COL=1, ROW=7) is received by SuperPixel (LOG_COL=0, LOG_ROW=3).

2) SUPERPIXEL → END-OF-COLUMN OUTPUT

SuperPixel → End-of-Column output

File: packets_output_EndOfColumn.csv

BX	COL	ROW	LOG_TIME	LOG_COL	LOG_ROW
2	1	7	200000	0	0

At 200000 ps (200 ns), the same hit is output by End-of-Column (column 0).

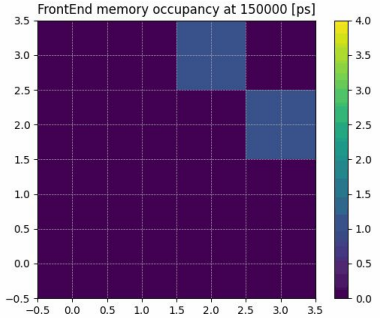
PIPELINE SUMMARY (Propagation Delay)

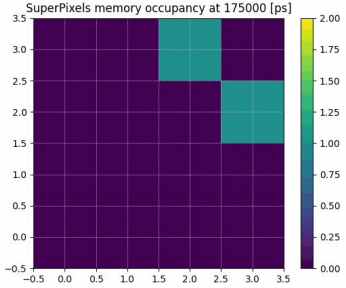
Stage Transition	Time (ns)	Delay from Previous (ns)
Front-End (hit detected)	75	–
Front-End → SuperPixel (in)	100	+25
SuperPixel (progressive)	125 / 150 / 175	+25 each
SuperPixel → End-of-Column	200	+25
End-of-Column → Data Concentrator	225	+25
Data Concentrator → Output (done)	250	+25

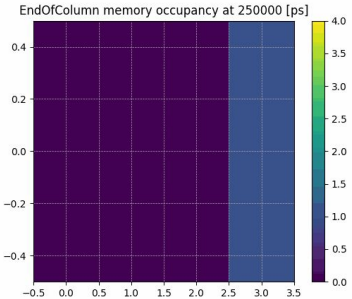
Total propagation time (Front-End hit → Final output): **250 ns**

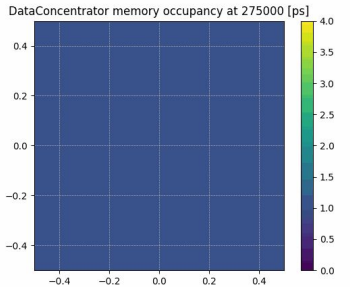
Network - Performance Analysis

- **Dark/black area**
  - FIFO occupancy is near zero (no backlog)
- **Green regions**
  - Packets are temporarily stored in the FIFO (memory usage increases)
- **More intense/large green areas**
  - Higher occupancy and possible congestion
- **DataConcentrator**
  - Shows periodic occupancy peak caused by packet bursts
  - Combines all four EndOfColumns into a single output path



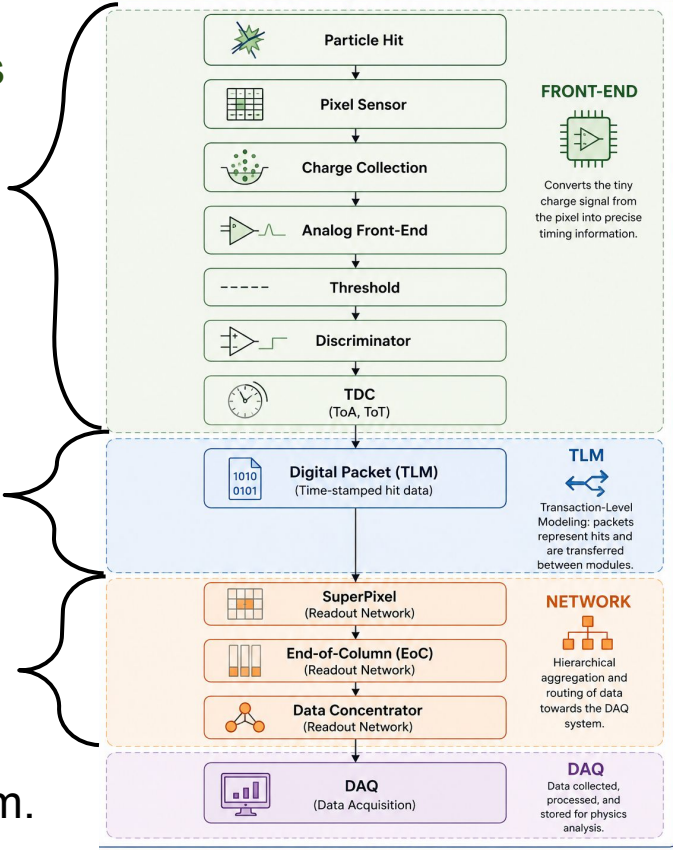






Putting It All Together

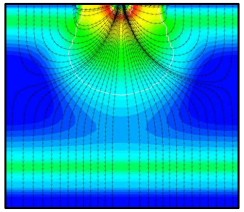
- ➔ The Front-End exercises modeled the first part of this chain.
- ➔ The TLM exercises modeled the communication.
- ➔ The Network exercises modeled the entire detector architecture.
- ➔ Together, they form a complete MAPS readout system.



A More Detailed Analog Front End?

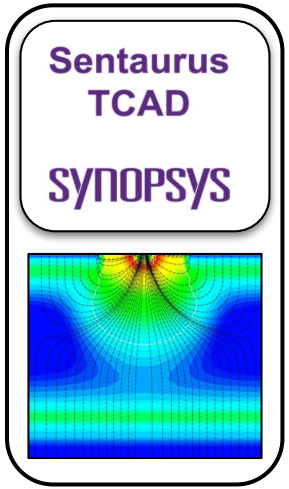
Sentaurus TCAD

SYNOPSYS



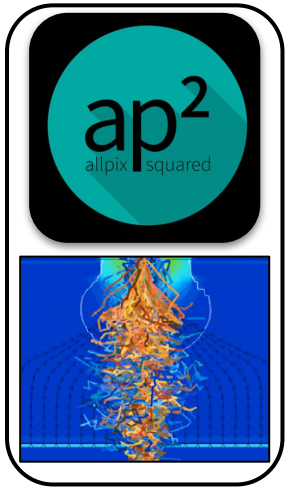
[D.Dannheim]

A More Detailed Analog Front End?



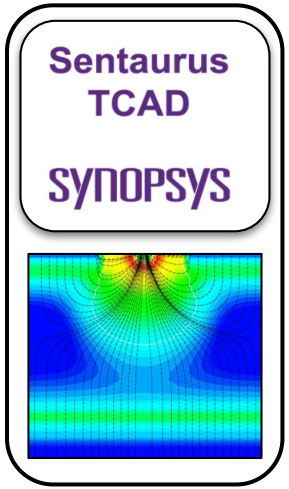
[D.Dannheim]

E-field



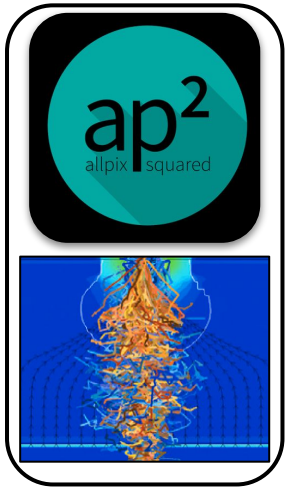
[L. Mendes]

A More Detailed Analog Front End?



[D.Dannheim]

E-field



[L. Mendes]

Digi hits

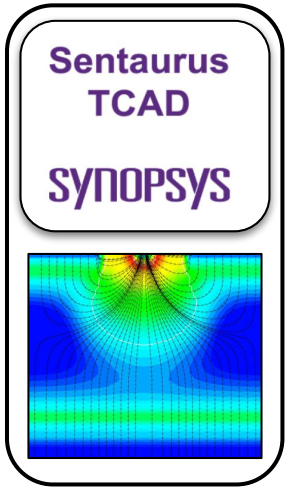
PixESL

Pixel detectors virtual **E**lectronic **S**ystem-**L**evel prototyping framework

➔ Saw simplified model of **analog response** to input **charge**

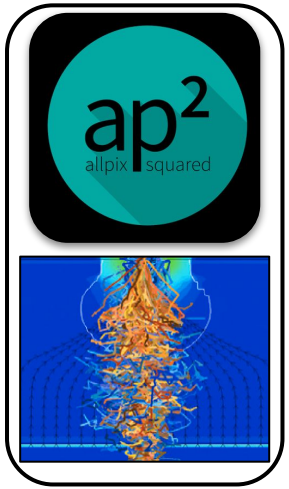
- Triangular waveform

A More Detailed Analog Front End?



[D.Dannheim]

E-field



[L. Mendes]

Digi hits

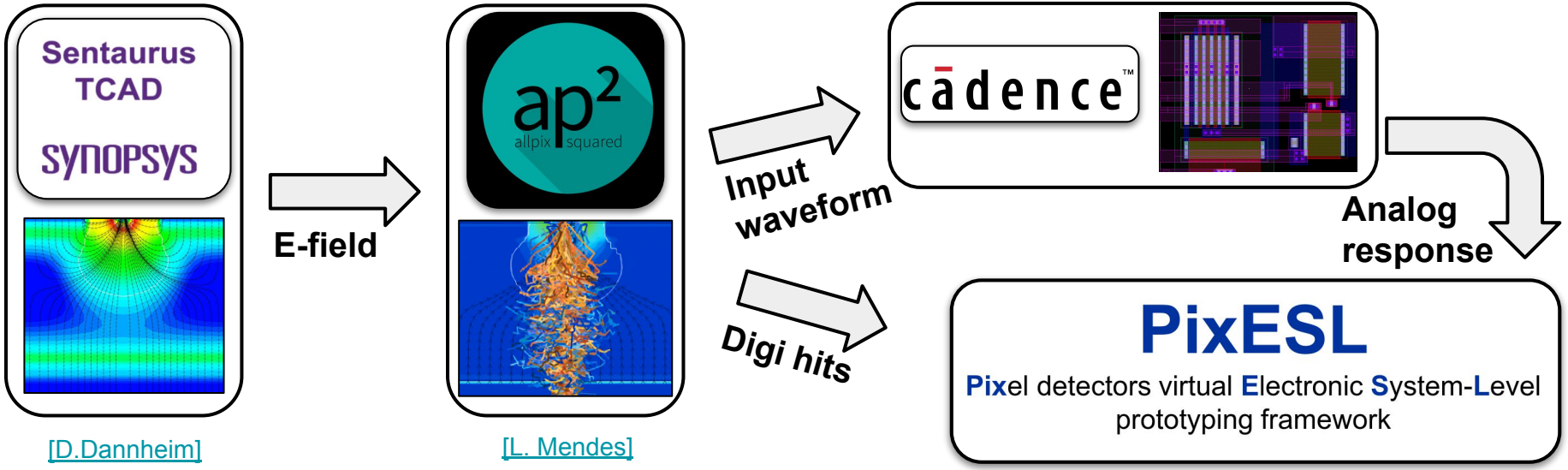
PixESL

Pixel detectors virtual **E**lectronic **S**ystem-**L**evel prototyping framework

➔ Saw simplified model of **analog response** to input **charge**

- Triangular waveform
- Can we do **better**?

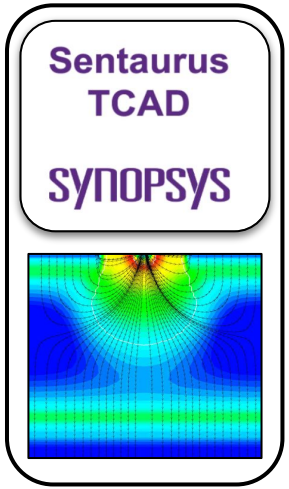
A More Detailed Analog Front End?



➔ Saw simplified model of **analog response** to input **charge**

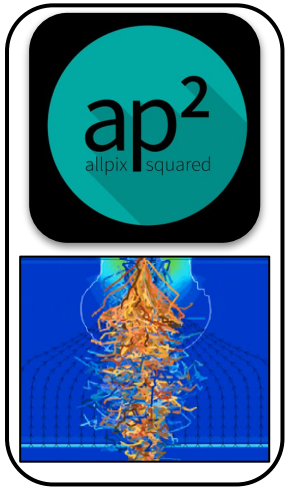
- Triangular waveform
- Can we do **better**?

A More Detailed Analog Front End?



[D.Dannheim]

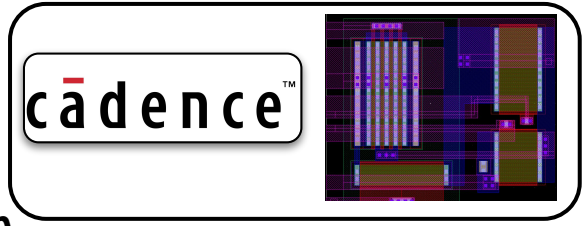
E-field



[L. Mendes]

Input waveform

Digi hits



Analog response

PixESL

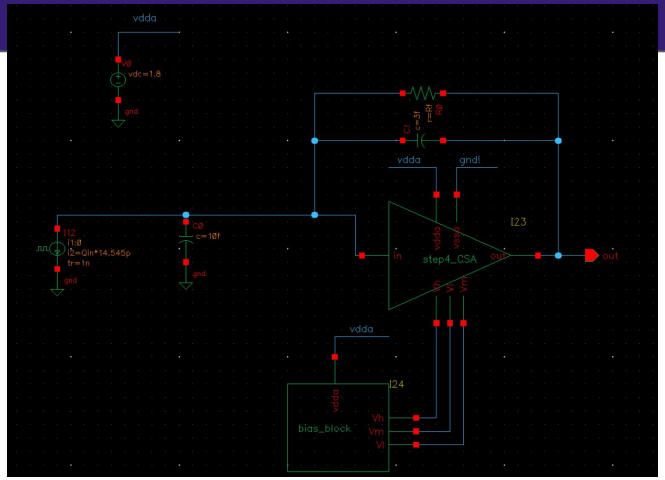
Pixel detectors virtual **E**lectronic **S**ystem-**L**evel prototyping framework

➔ Saw simplified model of **analog response** to input **charge**

- Triangular waveform
- Can we do **better**?

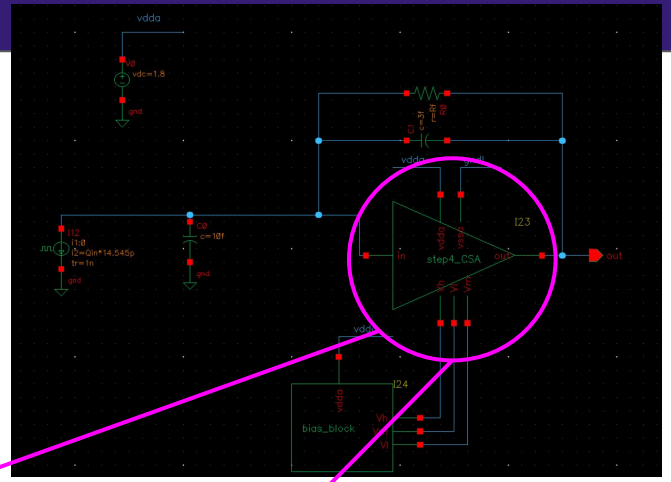
Reminder: CSA in Cadence

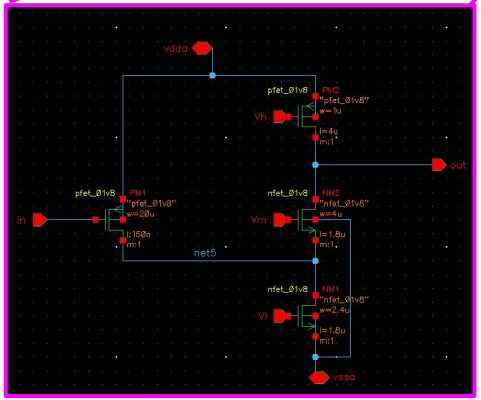
- ➔ Minimal **4-T design** in SkyWater 130nm CMOS process
  - ◆ Integrate current on feedback cap + reset with feedback resistor
  - ◆ Layout done by Matthew [see talk yesterday] in **Cadence Virtuoso**
- ➔ Observe response at **output** node to **input current** pulses



Reminder: CSA in Cadence

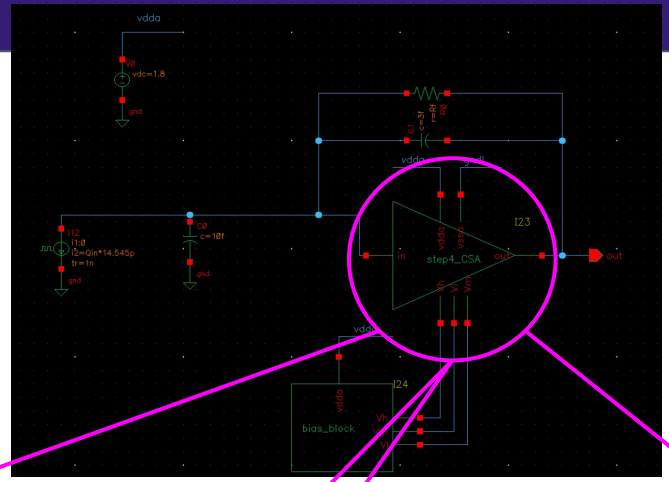
- ➔ Minimal **4-T design** in SkyWater 130nm CMOS process
  - ◆ Integrate current on feedback cap + reset with feedback resistor
  - ◆ Layout done by Matthew [see talk yesterday] in **Cadence Virtuoso**
- ➔ Observe response at **output** node to **input current** pulses

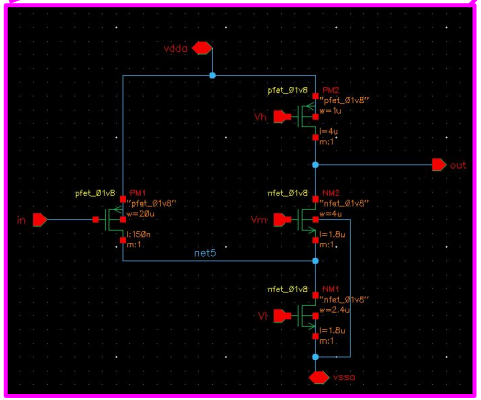


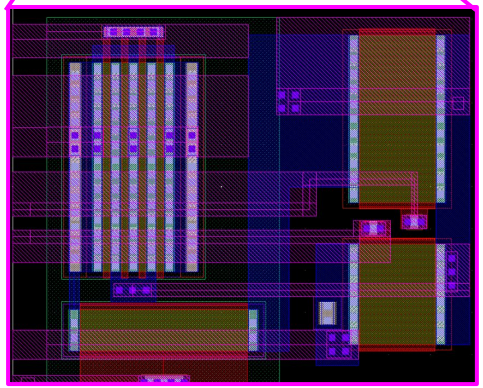


Reminder: CSA in Cadence

- ➔ Minimal **4-T design** in SkyWater 130nm CMOS process
  - ◆ Integrate current on feedback cap + reset with feedback resistor
  - ◆ Layout done by Matthew [see talk yesterday] in **Cadence Virtuoso**
- ➔ Observe response at **output** node to **input current** pulses

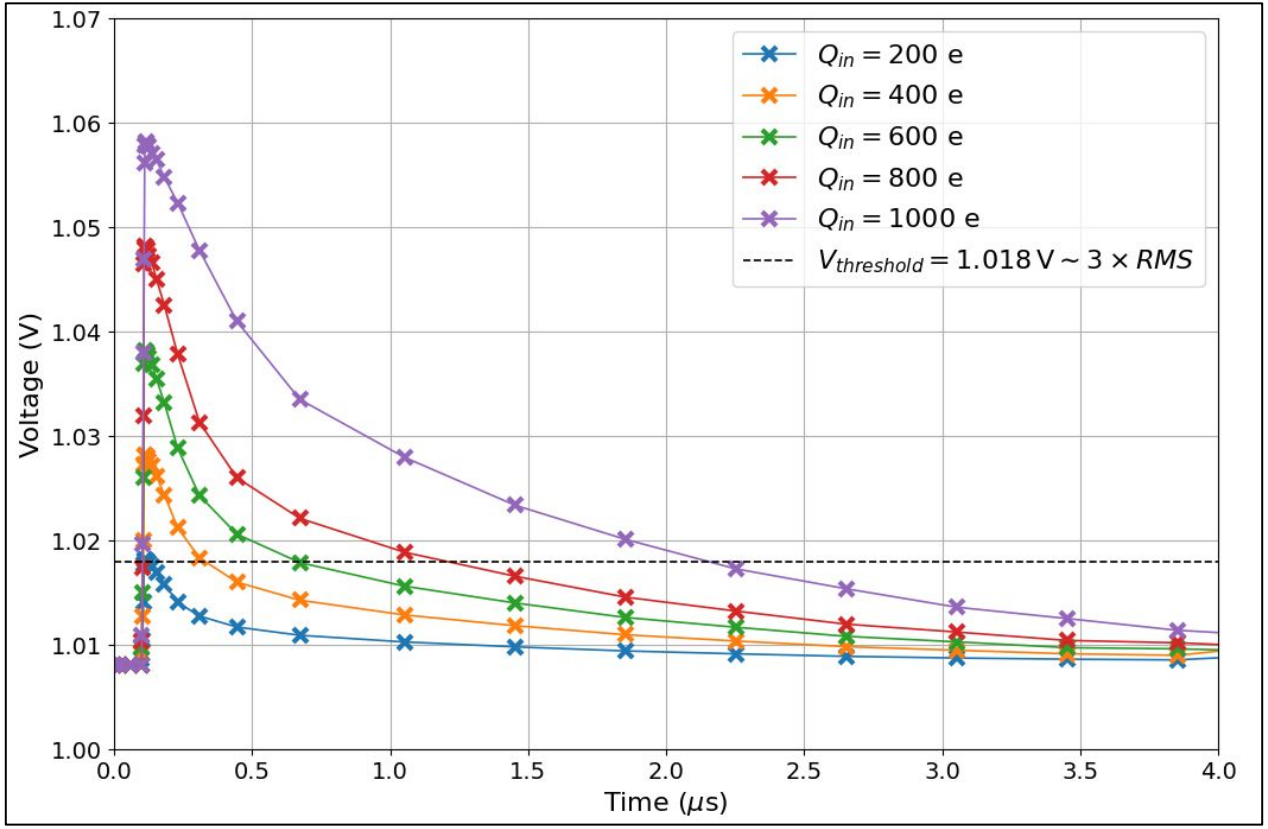






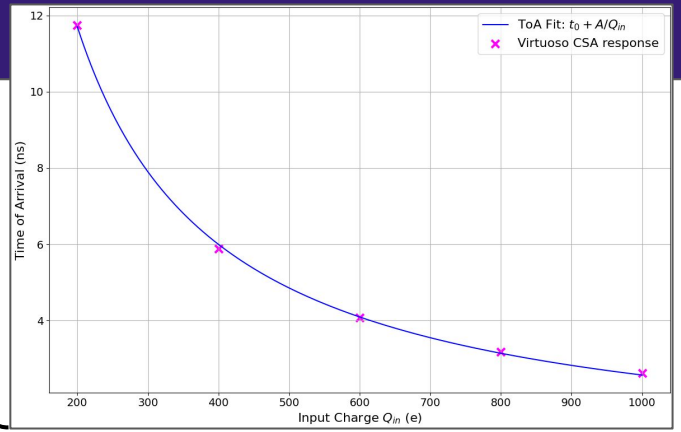
Transient Voltage Response

- ➔ Observe **voltage at output** node following charge injections
  - ◆ [200, 1000] e
- ➔ Set threshold to **~3x RMS** noise at input
- ➔ Derive **ToA**, **ToT**, **ToE** for all waveforms



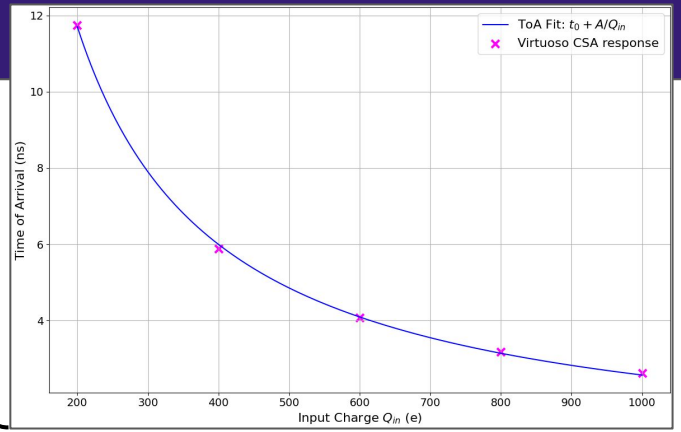
ToA/ToT vs Q

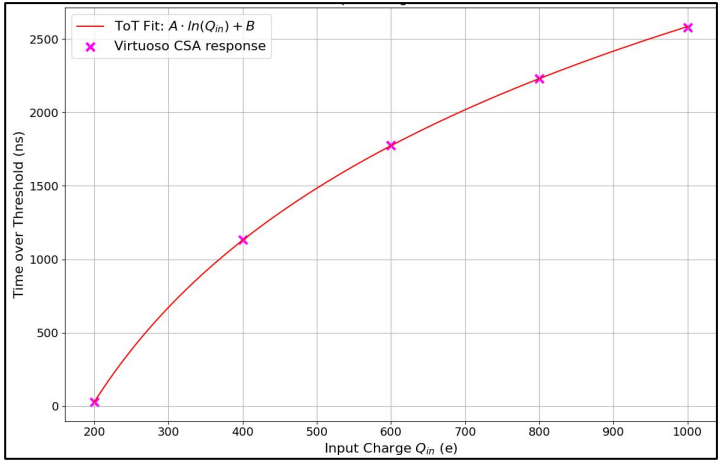
- ➔ After **fixing threshold** scan input charge to extract
  - ◆ ToA
  - ◆ ToT
- ➔ Assume rise time ~ constant
  - ◆ $$\Delta t_{ToA} \sim \frac{V_{th}}{C_f \cdot Q/t_r} \sim \frac{1}{Q}$$



ToA/ToT vs Q

- After **fixing threshold** scan input charge to extract
  - ToA
  - ToT
- Assume rise time ~ constant
  - $\Delta t_{ToA} \sim \frac{V_{th}}{C_f \cdot Q/t_r} \sim \frac{1}{Q}$
- Voltage decays exponentially as cap discharges
  - $V(t) \sim V_0 \cdot e^{\frac{-t}{\tau}}$
  - $\Delta t_{ToT} \sim \tau \cdot \ln(Q) + C$





Implementation in PixESL

➔ Implement ToA/ToT fits as analytical model in **Analog Front-End core**

$$\mathrm{ToA} = f(\mathrm{Q}_{in})$$
$$\mathrm{ToT} = g(\mathrm{Q}_{in})$$

Implementation in PixESL

➔ Implement ToA/ToT fits as analytical model in **Analog Front-End core**

$$\mathrm{ToA} = f(\mathrm{Q}_{in})$$
$$\mathrm{ToT} = g(\mathrm{Q}_{in})$$

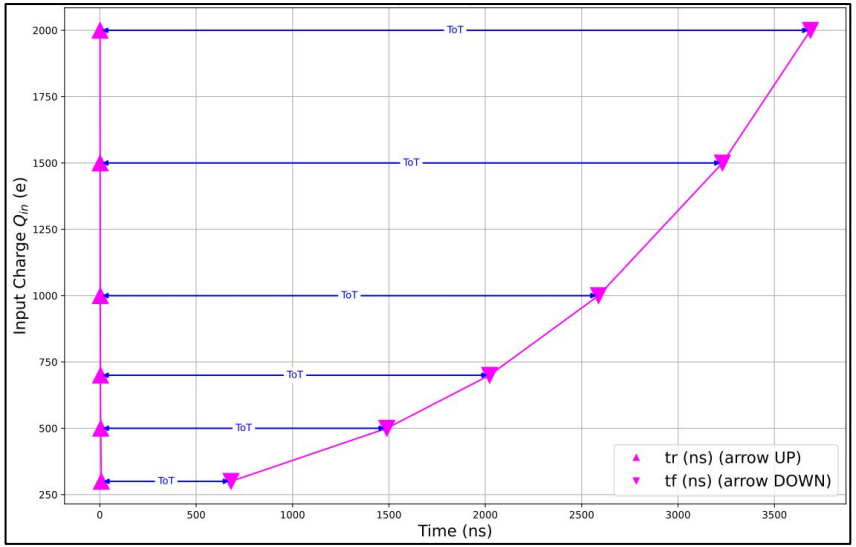
Analog Front-End core [C++]
analytical model of the analog circuits

Implementation in PixESL

- ➔ Implement ToA/ToT fits as analytical model in **Analog Front-End core**
- ➔ Scan ToA and ToE for different input charges
  - ◆ [300, 2000] e
  - ◆ Observe more-realistic, non-linear response

$$\mathrm{ToA} = f(\mathrm{Q}_{in})$$
$$\mathrm{ToT} = g(\mathrm{Q}_{in})$$

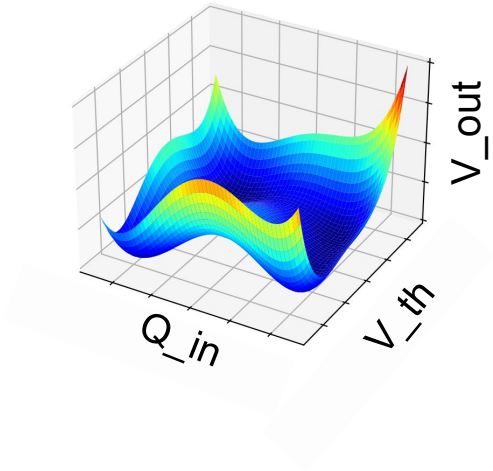
Analog Front-End core [C++]
analytical model of the analog circuits



Next?

- ➔ Pass model to further PixESL modules including **TDC** and **network**
- ➔ Implement **2-d plane** (Q_in, V_threshold) -> V_out directly?
  - ◆ Digitization could be performed by AFE core
- ➔ And finally…

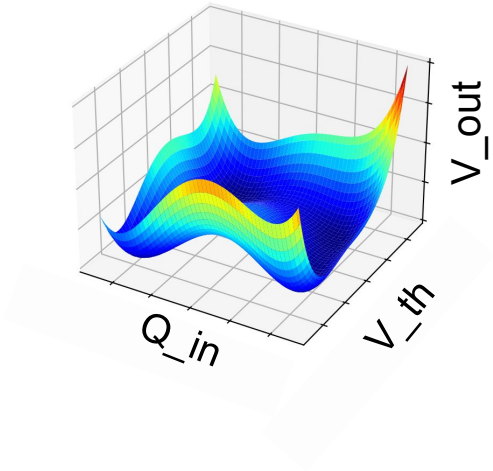
$$\mathrm{ToA} = f(\mathrm{Q}_{in})$$
$$\mathrm{ToT} = g(\mathrm{Q}_{in})$$



Next?

- ➔ Pass model to further PixESL modules including **TDC** and **network**
- ➔ Implement **2-d plane** (Q_in, V_threshold) -> V_out directly?
  - ◆ Digitization could be performed by AFE core
- ➔ And finally…

$$\mathrm{ToA} = f(\mathrm{Q}_{in})$$
$$\mathrm{ToT} = g(\mathrm{Q}_{in})$$



Thanks to the organizers and facilitators for a wonderful school!

Backup