Verification Environment for ALTIROC3 ASIC of the ATLAS High Granularity Timing Detector

Simone Scarfì (CERN) on behalf of ATLAS HGTD group

TWEPP23, October 2023

simone.scarfi@cern.ch

High Granularity Timing Detector for HL-LHC

HL-LHC

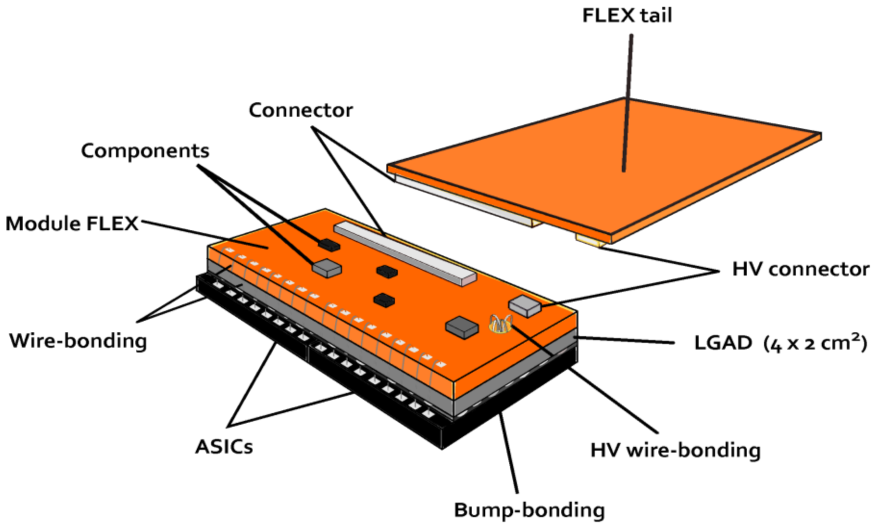
Integrated luminosity	$4000\ \mathrm{fb}^{-1}$
Pile-up	200

ATLAS experiment upgrades

HGTD

- Improve the performance of particle detection in the **forward region of the ATLAS detector**
- Provide **timing resolution of 30 ps** per minimum-ionizing particle to mitigate the effects of the pile-up in the forward region
- Provide a **luminosity** measurement

HGTD Module



https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/UPGRADE/CERN-LHCC-2020-007/

Posters at TWEPP 2023

Mingjie Zhai
HGTD PEB DC/DC Power Block in Low Temperature and Magnetic Field Operation
https://indico.cern.ch/event/1255624/contributions/5445331/

Zhenwu Ge
An FPGA-based Front-end Module Emulator for the High Granularity Timing Detector Readout Module
https://indico.cern.ch/event/1255624/contributions/5445302/

Jie Zhang
The Prototype Design of PEB - a Component of the HGTD In-detector Electronics for the ATLAS Phase-II Upgrade
https://indico.cern.ch/event/1255624/contributions/5445321/

Outline

ALTIROC3 **requirements** and **specifications**

Main **projects challenges** from a verification point of view:

- Source code management and IP-blocks versioning
- Complex clock architecture with several domains
- Accurate **analog IP-blocks** models using digital-on-top design methodology
- More than **1000 configurable registers**
- **SEE simulations**

Outline

ALTIROC3 **requirements** and **specifications**

Main **projects challenges** from a verification point of view:

- Source code management and IP-blocks versioning
- Complex clock architecture with several domains
- Accurate **analog IP-blocks** models using digital-on-top design methodology
- More than **1000 configurable registers**
- **SEE simulations**

Examples of **verification methodologies** adopted

Altiroc3 specifications

Chip area	*22.5 mm x 20 mm*
Number of pixels	*15 x 15 = 225*
Pixel size	*1.3 mm x 1.3 mm*
Technology	*130 nm technology*

Minimum detectable charge	*2 fC*
TOA measurement	*resolution 20 ps, range 2.5 ns*
TOT measurement	*resolution 40 ps, range 20 ns*
Luminosity	*resolution 100 ps, range 25 ns*

Triggered readout

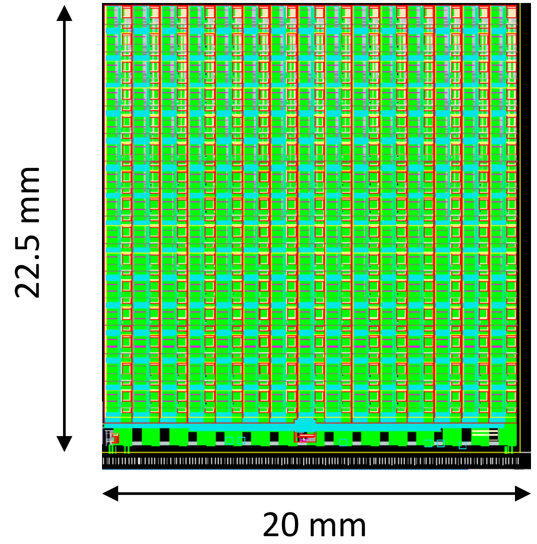
- *configurable transmission rate (320 Mbps, 640 Mbps, 1280 Mbps)*
- *configurable protocol (8b10b encoding, raw data)*

Continuous readout

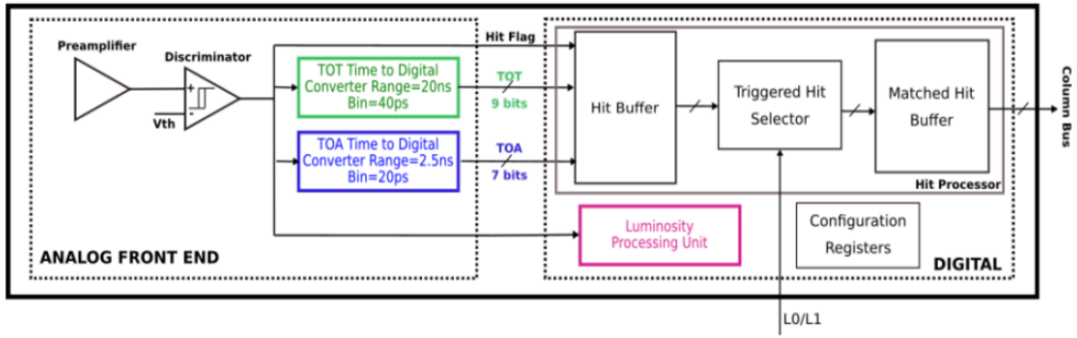
- *fixed transmission rate - 640 Mbps*
- *fixed protocol - 6b8b encoding*

Power consumption	*1.2 W (50% digital, 50% analog)*
Radiation tolerance	*TID 200 Mrad, SEE robustness*

Altiroc3 ASIC



Pixel block diagram



Source code management and IP-block versioning

Main requirements:

- Same tools in the four different institutions
- IP-blocks, RTL, scripts version control

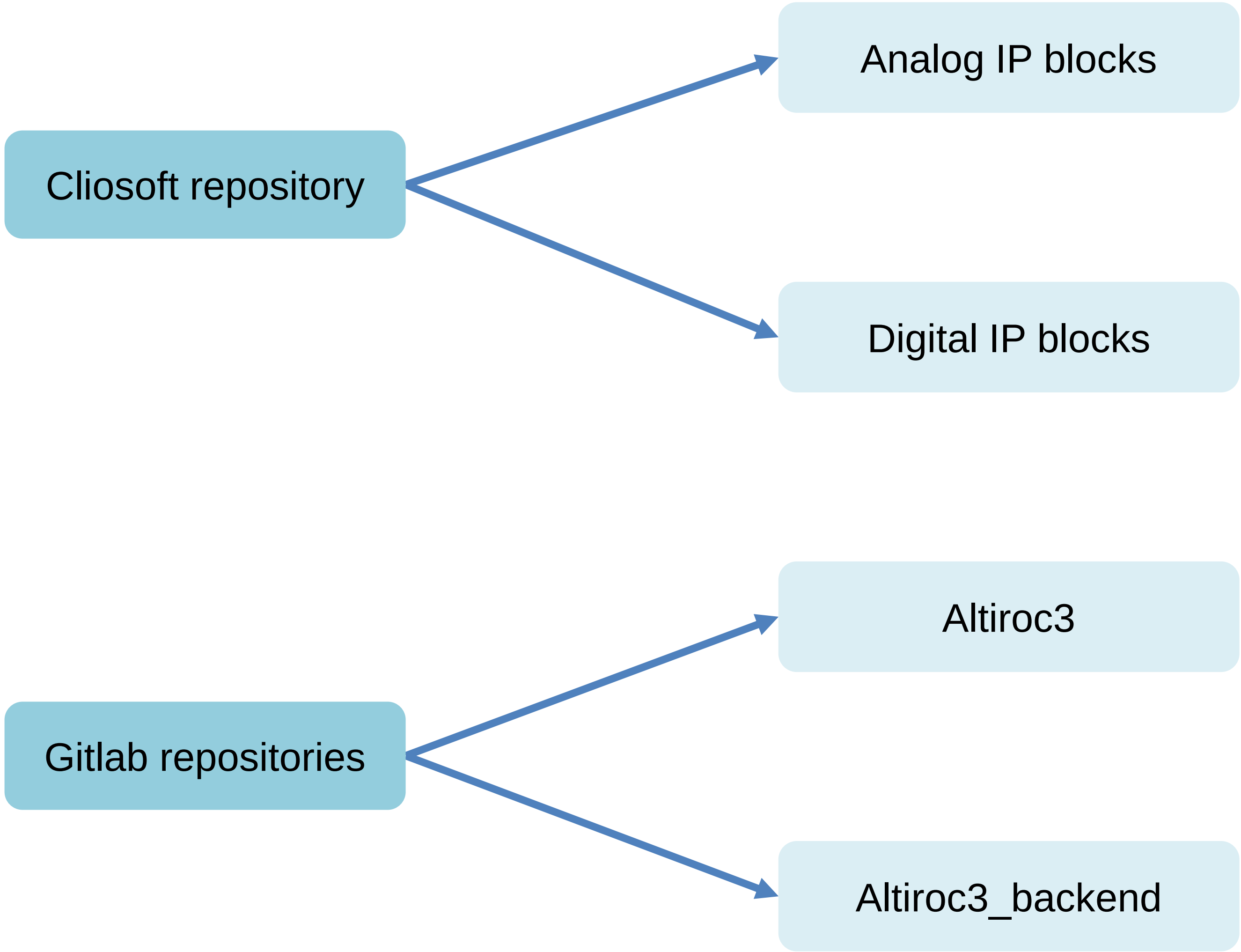
Each member of the team must be able to:

- Launch PNR flows
- Launch full chip simulations

Advantages:

- **Easier collaboration**
- **Reproducibility of results**

Multi-site project code organization



Cliosoft repository

Analog IP blocks

Digital IP blocks

Gitlab repositories

Altiroc3

Altiroc3_backend

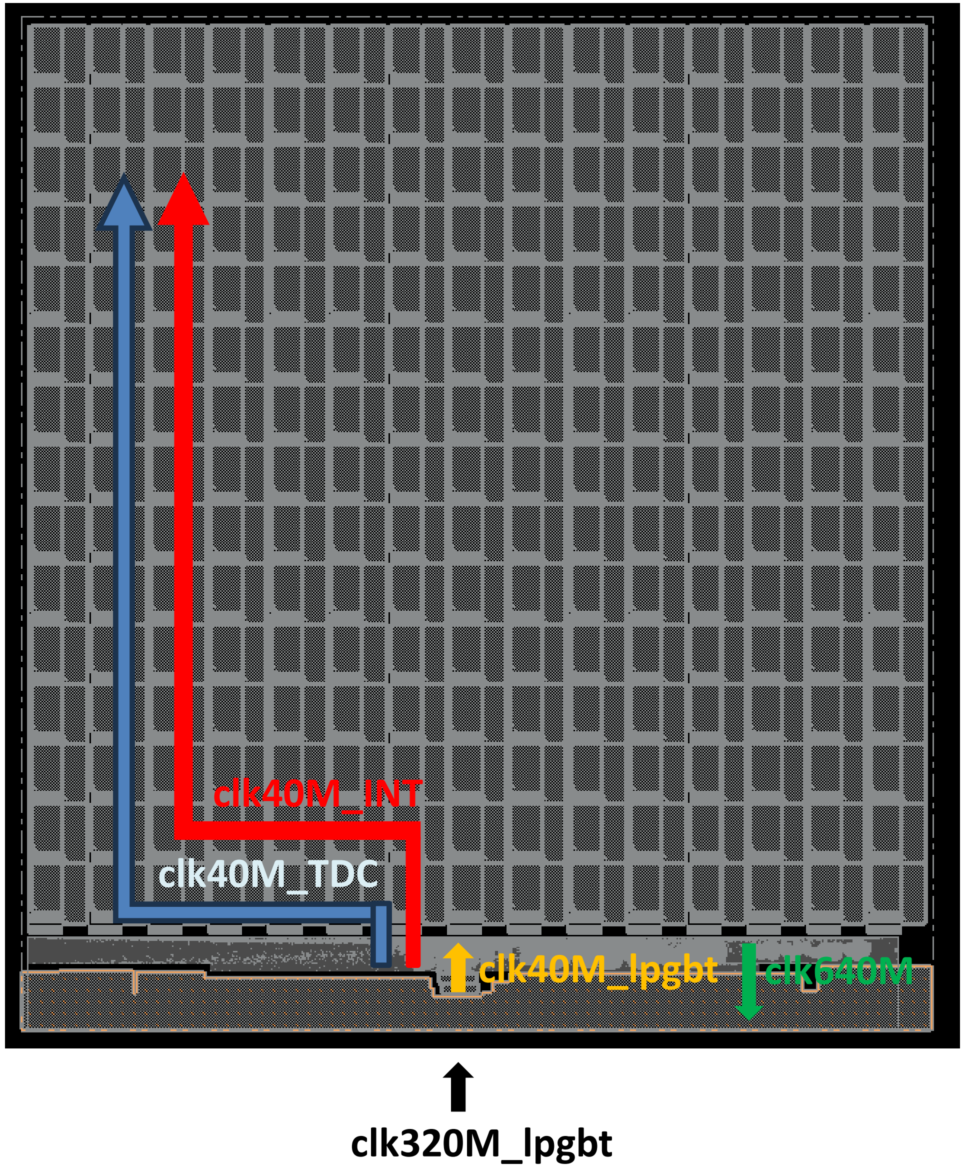
OA: virtuoso libraries
abstract: for digital top integration
liberty files: with extracted view in different corners used for digital implementation

OA: virtuoso library
liberty files: liberty files for implementation of the top level
physical: .def, .lef and .gds files
simulation: .v and .sdf files for simulation

RTL, **triplicated RTL**
UVM testbench

Synthesis scripts and **PNR** scripts

Altiroc3 simplified clock architecture



clk320M_lpgbt:

- Input clock from LpGBT at 320 MHz

clk40M_lpgbt:

- Derived from **clk320M_lpgbt**
- Used to send out data always in phase

clk40M_INT for digital processing logic (matrix and periphery):

- Derived from **clk320M_lpgbt**
- 40 MHz clock, fully triplicated
- Skew maximized to spread current peaks over a larger period, minimizing noise
- Phase set according to clk40M_TDC rising edge to avoid digital current spikes affecting TDC measurement (ensuring always a distance among the two)

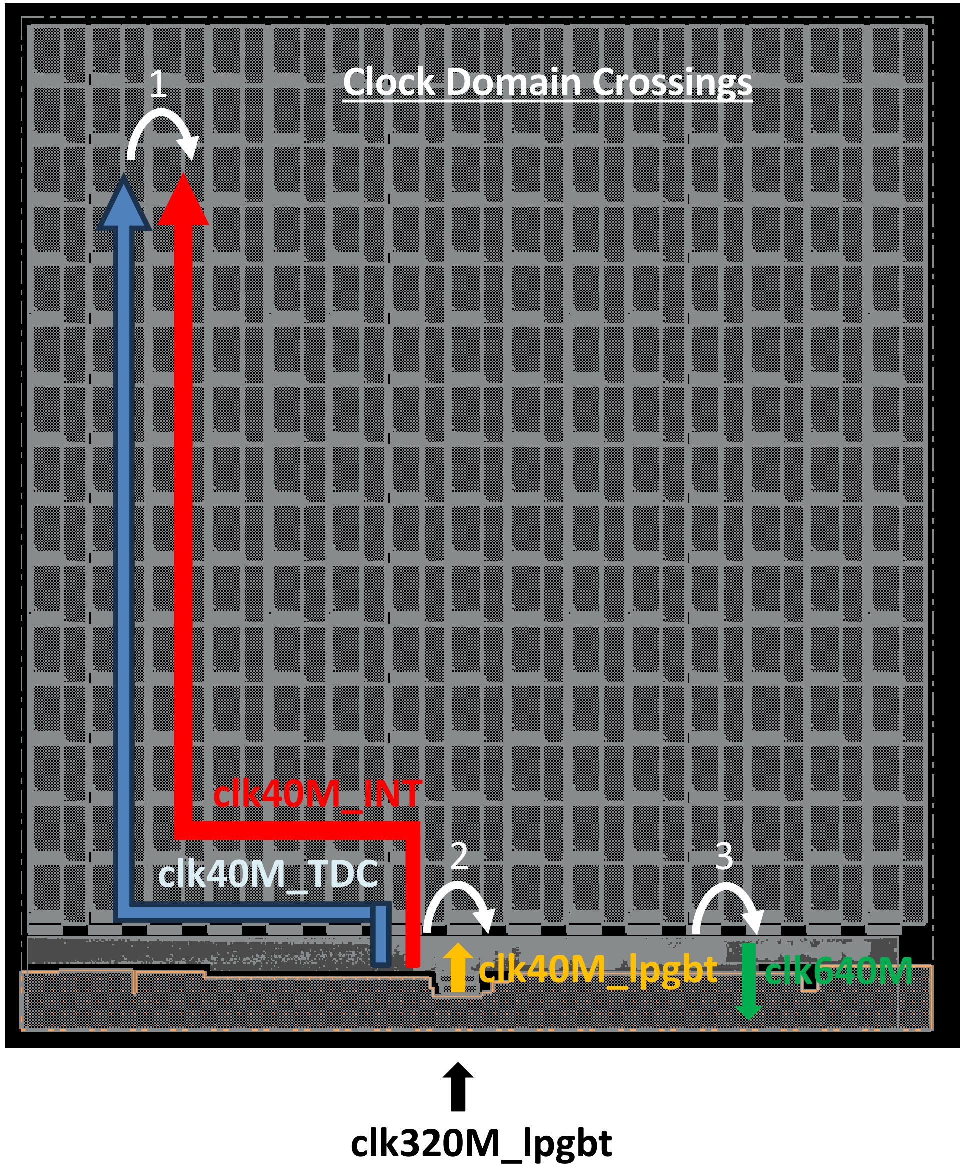
clk40M_TDC (stop condition for TDC):

- 40 MHz clock, not triplicated
- Requires to be aligned with the Bunch Crossing (BC) collision for a TOA measurement with a range of 2.5 ns
- Very precise skew over all matrix (+- 150 ps)
- PLL minimizes jitter
- Coarse shift by 1.56 ns (range 25 ns)
- Fine shift by 100 ps (range 1.56 ns) with Phase Shifter

clk640M:

- Derived by the PLL
- Needed for high-frequency output serializers

Altiroc3 Clock Domain Crossings (CDC)



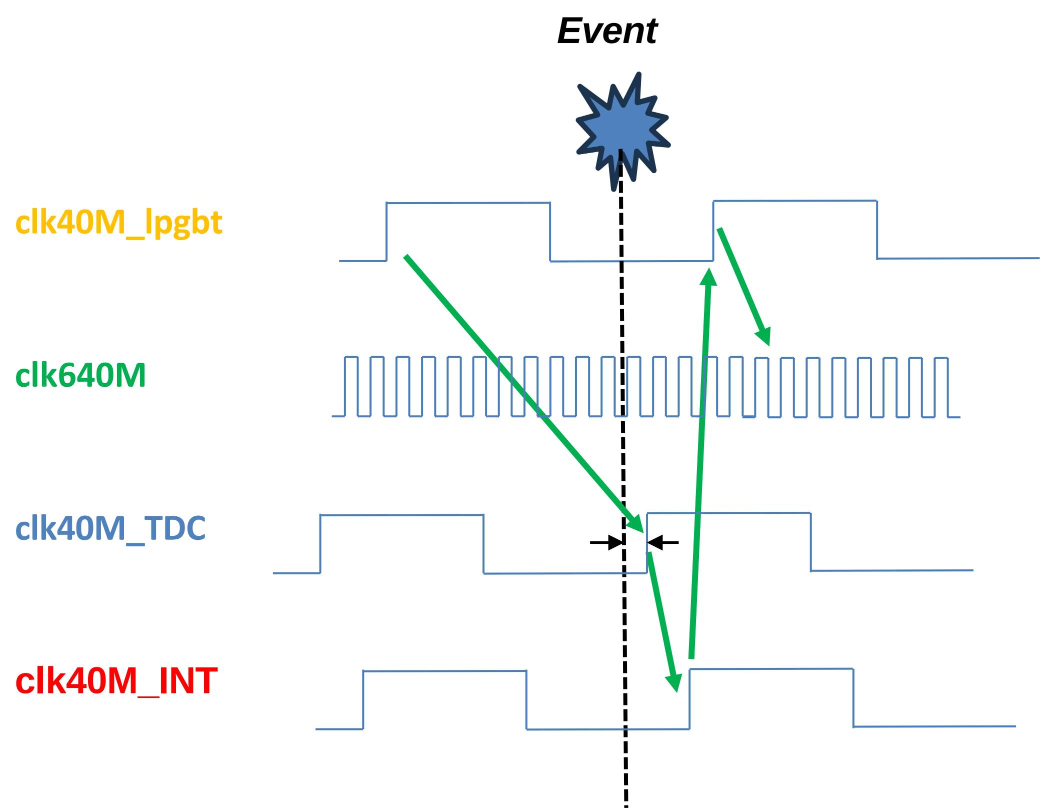
Clock Domain Crossings (CDC):

1. **clk40M_TDC - clk40M_INT** in each pixel:
   - Data from TDC are sampled with clk40M_TDC and are transmitted to clk40M_INT domain for processing
   - clk40M_TDC and clk40M_int accumulate different delay along the column/matrix, the CDC is checked via Static Timing Analysis (STA) and exact delay is known for all pixels in all corners

2. **clk40M_INT - clk40M_lpgbt** in periphery:
   - Data processed in the clk40M_INT domain must be aligned with clk40M_lpgbt domain for output transmission
   - Phase between them is known, custom CDC is used (different sampling edge positive/negative)

3. **clk40M_lpgbt - clk640M**
   - CDC through a PLL

Example of data processing



1. clk40M_lpgbt is in the same domain of **clk320M_lpgbt,** received by the ASIC
2. clk40M_TDC is set according to Time Of Flight and position of ASIC in the detector, to align within 2.5 ns window around event
3. clk40M_INT is set to ensure a certain distance from clk40M_TDC where digital noise on TDC is minimized
4. clk640M transmits out the data

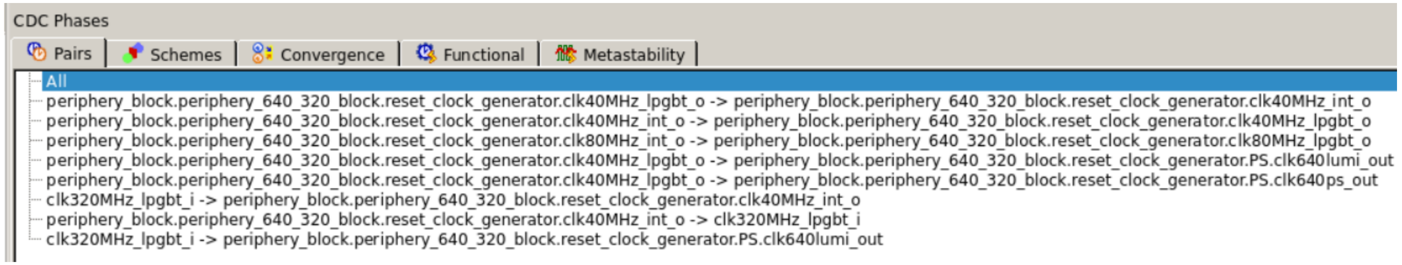
Verification of CDC with Formal Verification (JasperGold)

Efficiently **identify** all **CDC** at an **early stage** of the design (**RTL level**)

Debug **structural failures** (missing synchronizers, structural glitches, convergence/divergence)

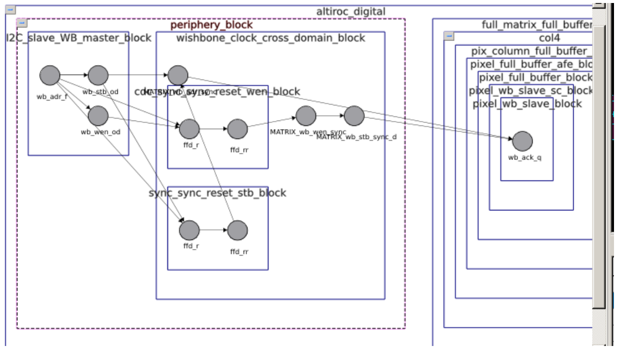
Generate **assertions** for running protocol checks during functional verification (e.g.: data stable for N clock cycles, etc)

Found Pairs

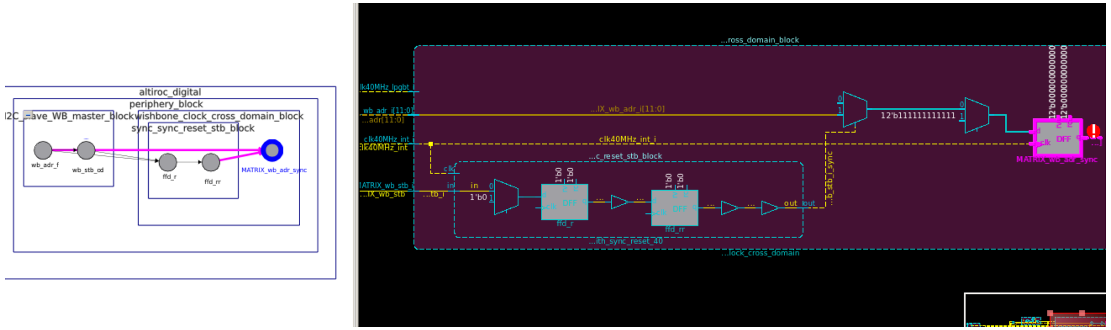


periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_lpgbt_o -> periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_int_o
periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_int_o -> periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_lpgbt_o
periphery_block.periphery_640_320_block.reset_clock_generator.clk80MHz_int_o -> periphery_block.periphery_640_320_block.reset_clock_generator.clk80MHz_lpgbt_o
periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_lpgbt_o -> periphery_block.periphery_640_320_block.reset_clock_generator.PS.clk640lumi_out
periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_lpgbt_o -> periphery_block.periphery_640_320_block.reset_clock_generator.PS.clk640ps_out
clk320MHz_lpgbt_i -> periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_int_o
periphery_block.periphery_640_320_block.reset_clock_generator.clk40MHz_int_o -> clk320MHz_lpgbt_i
clk320MHz_lpgbt_i -> periphery_block.periphery_640_320_block.reset_clock_generator.PS.clk640lumi_out

Scheme



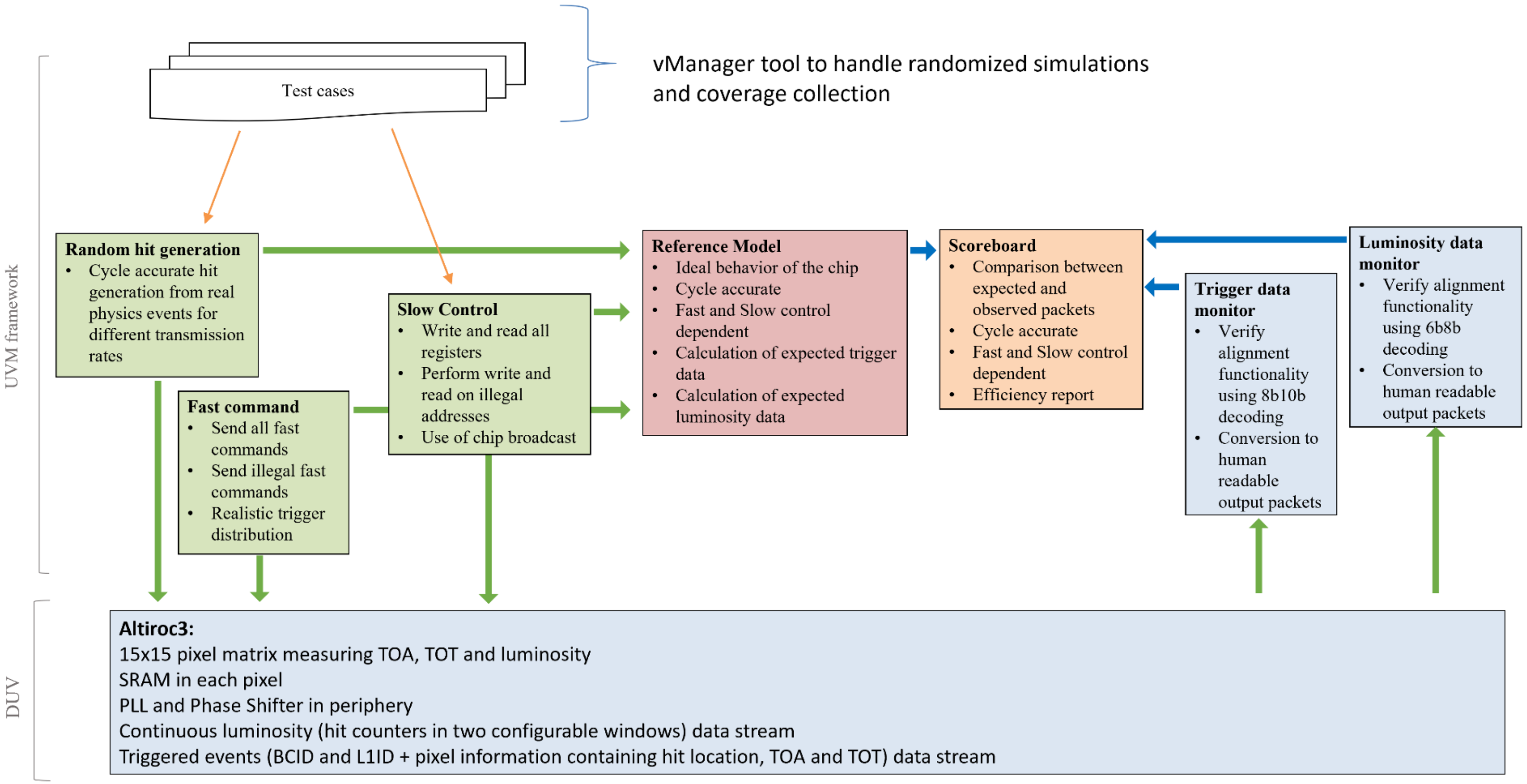
Analysis



Limitation: only checks for structural errors, we still require functional verification

Verification of CDC with Functional Verification

Only possible when the design is already at an **advanced stage (NETLIST level)**



Test cases

vManager tool to handle randomized simulations and coverage collection

UVM framework

Random hit generation
- Cycle accurate hit generation from real physics events for different transmission rates

Fast command
- Send all fast commands
- Send illegal fast commands
- Realistic trigger distribution

Slow Control
- Write and read all registers
- Perform write and read on illegal addresses
- Use of chip broadcast

Reference Model
- Ideal behavior of the chip
- Cycle accurate
- Fast and Slow control dependent
- Calculation of expected trigger data
- Calculation of expected luminosity data

Scoreboard
- Comparison between expected and observed packets
- Cycle accurate
- Fast and Slow control dependent
- Efficiency report

Trigger data monitor
- Verify alignment functionality using 8b10b decoding
- Conversion to human readable output packets

Luminosity data monitor
- Verify alignment functionality using 6b8b decoding
- Conversion to human readable output packets

DUV

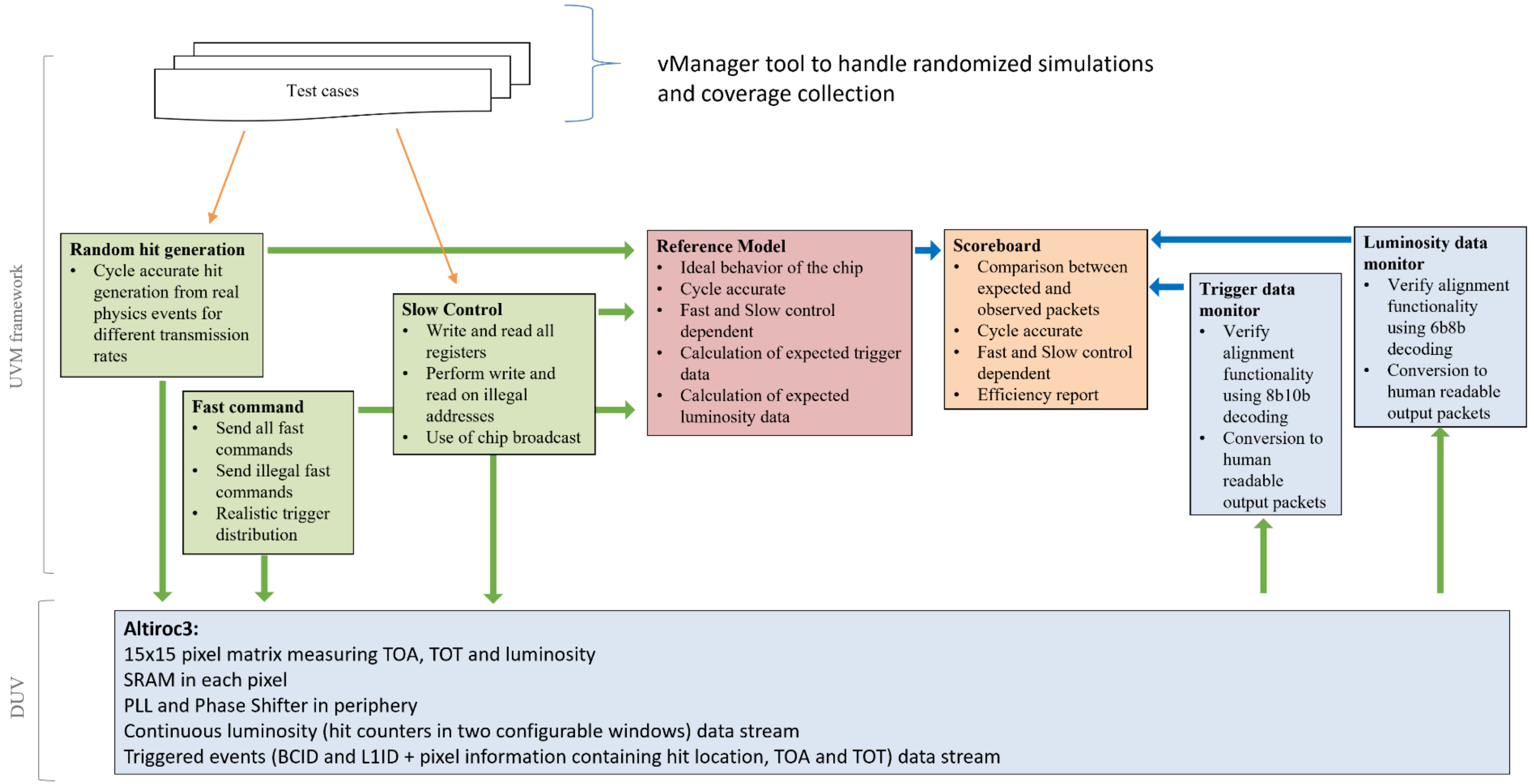
Altiroc3:
15x15 pixel matrix measuring TOA, TOT and luminosity
SRAM in each pixel
PLL and Phase Shifter in periphery
Continuous luminosity (hit counters in two configurable windows) data stream
Triggered events (BCID and L1ID + pixel information containing hit location, TOA and TOT) data stream

Ideally, one should develop the **UVM testbench** having in mind target application:

- **Stimuli**: Ability to **move hit generation**
- **Randomization**: Ability to configure internal **clock phases** in a ‘constrained randomized’ way
- **Correctness**: ASIC can proper sample hits and process data

Verification of CDC with Functional Verification

Only possible when the design is already at an **advanced stage (NETLIST level)**



Test cases

vManager tool to handle randomized simulations and coverage collection

UVM framework

Random hit generation
- Cycle accurate hit generation from real physics events for different transmission rates

Fast command
- Send all fast commands
- Send illegal fast commands
- Realistic trigger distribution

Slow Control
- Write and read all registers
- Perform write and read on illegal addresses
- Use of chip broadcast

Reference Model
- Ideal behavior of the chip
- Cycle accurate
- Fast and Slow control dependent
- Calculation of expected trigger data
- Calculation of expected luminosity data

Scoreboard
- Comparison between expected and observed packets
- Cycle accurate
- Fast and Slow control dependent
- Efficiency report

Trigger data monitor
- Verify alignment functionality using 8b10b decoding
- Conversion to human readable output packets

Luminosity data monitor
- Verify alignment functionality using 6b8b decoding
- Conversion to human readable output packets

DUV

Altiroc3:
15x15 pixel matrix measuring TOA, TOT and luminosity
SRAM in each pixel
PLL and Phase Shifter in periphery
Continuous luminosity (hit counters in two configurable windows) data stream
Triggered events (BCID and L1ID + pixel information containing hit location, TOA and TOT) data stream

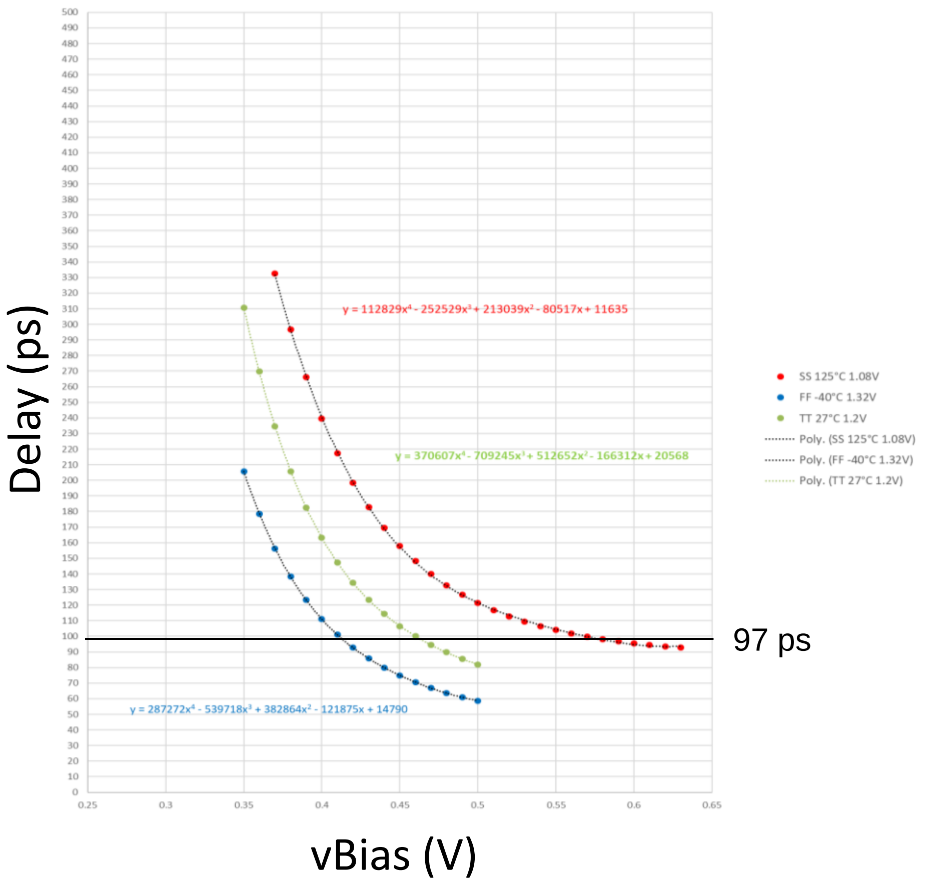
Requires accurate Verilog models:
- Analog Front-End
- PLL
- Phase Shifter

Ideally, one should develop the **UVM testbench** having in mind target application:
- **Stimuli**: Ability to **move hit generation**
- **Randomization**: Ability to configure internal **clock phases** in a ‘constrained randomized’ way
- **Correctness**: ASIC can proper sample hits and process data

Phase Shifter analog simulation

Analog simulations with extracted view

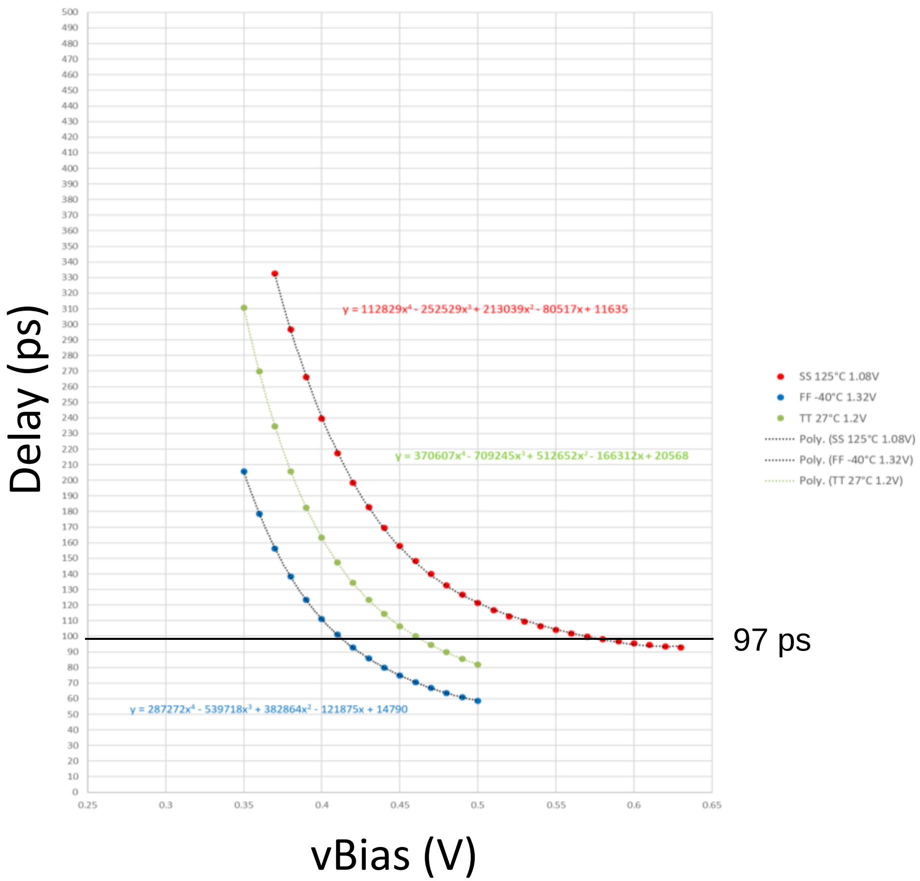
Delay of a single delay cell



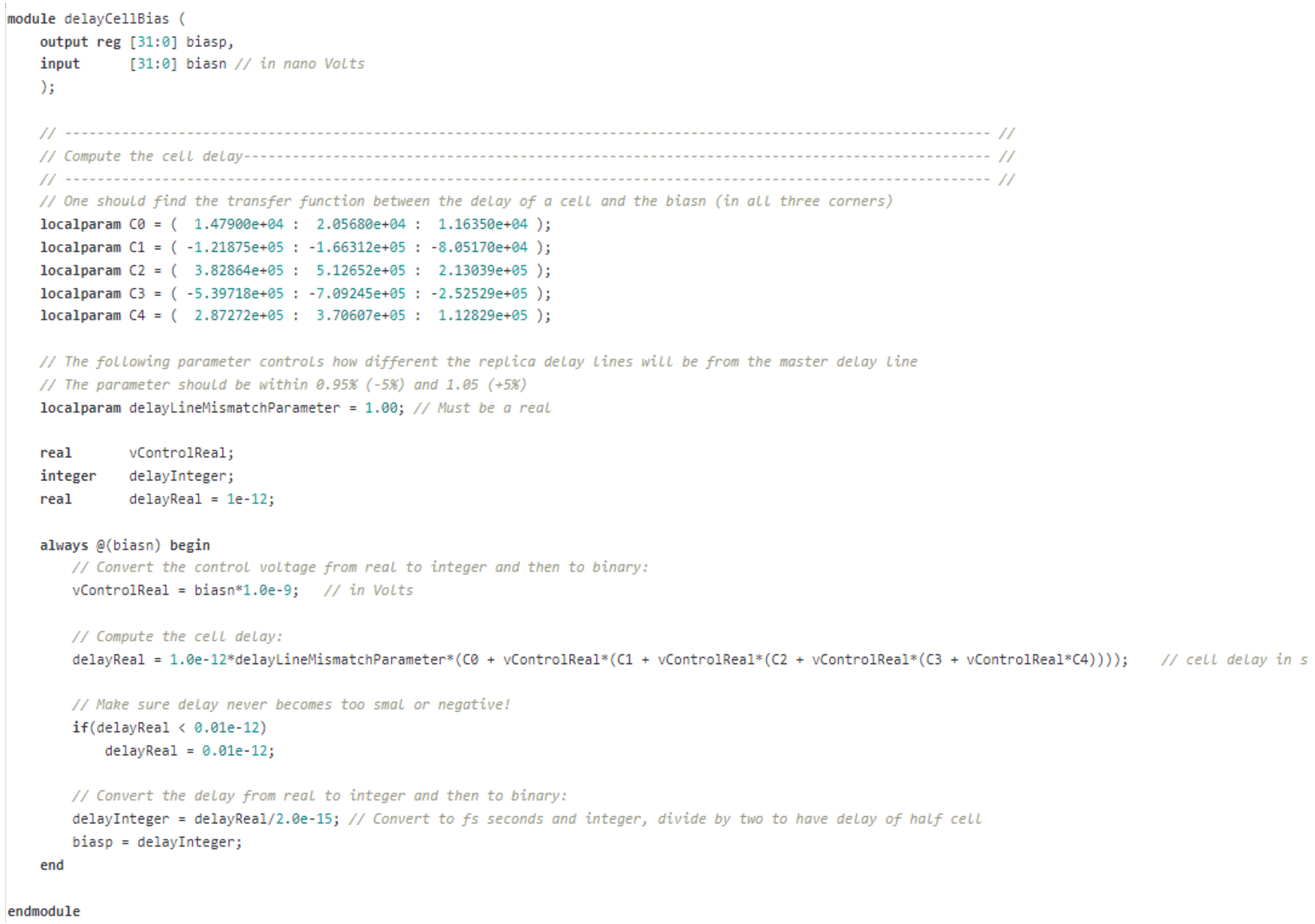
Phase Shifter Verilog model

Extraction of the model from Virtuoso and modifications of key blocks

Delay of a single delay cell



Example: Verilog model of a delay cell

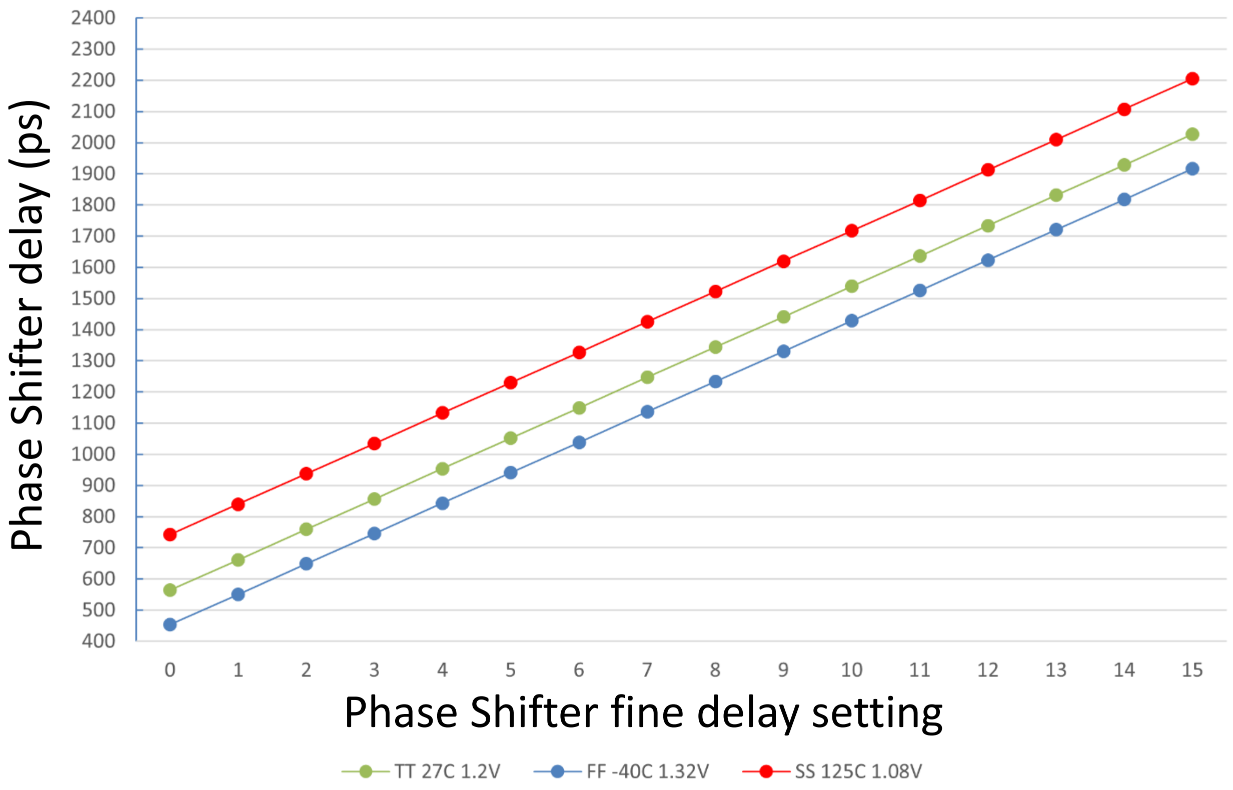


```
module delayCellBias (
    output reg [31:0] biasp,
    input      [31:0] biasn // in nano Volts
    );

    // ---------------------------------------------------------------------------------------------------------------- //
    // Compute the cell delay--------------------------------------------------------------------------------------------- //
    // ---------------------------------------------------------------------------------------------------------------- //
    // One should find the transfer function between the delay of a cell and the biasn (in all three corners)
    localparam C0 = (  1.47900e+04 :  2.05680e+04 :  1.16350e+04 );
    localparam C1 = ( -1.21875e+05 : -1.66312e+05 : -8.05170e+04 );
    localparam C2 = (  3.82864e+05 :  5.12652e+05 :  2.13039e+05 );
    localparam C3 = ( -5.39718e+05 : -7.09245e+05 : -2.52529e+05 );
    localparam C4 = (  2.87272e+05 :  3.70607e+05 :  1.12829e+05 );

    // The following parameter controls how different the replica delay lines will be from the master delay line
    // The parameter should be within 0.95% (-5%) and 1.05 (+5%)
    localparam delayLineMismatchParameter = 1.00; // Must be a real

    real       vControlReal;
    integer    delayInteger;
    real       delayReal = 1e-12;

    always @(biasn) begin
        // Convert the control voltage from real to integer and then to binary:
        vControlReal = biasn*1.0e-9;   // in Volts

        // Compute the cell delay:
        delayReal = 1.0e-12*delayLineMismatchParameter*(C0 + vControlReal*(C1 + vControlReal*(C2 + vControlReal*(C3 + vControlReal*C4))));    // cell delay in s

        // Make sure delay never becomes too smal or negative!
        if(delayReal < 0.01e-12)
            delayReal = 0.01e-12;

        // Convert the delay from real to integer and then to binary:
        delayInteger = delayReal/2.0e-15; // Convert to fs seconds and integer, divide by two to have delay of half cell
        biasp = delayInteger;
    end

endmodule
```

Phase Shifter Verilog model

Phase Shifter behavior fully modeled in all corners

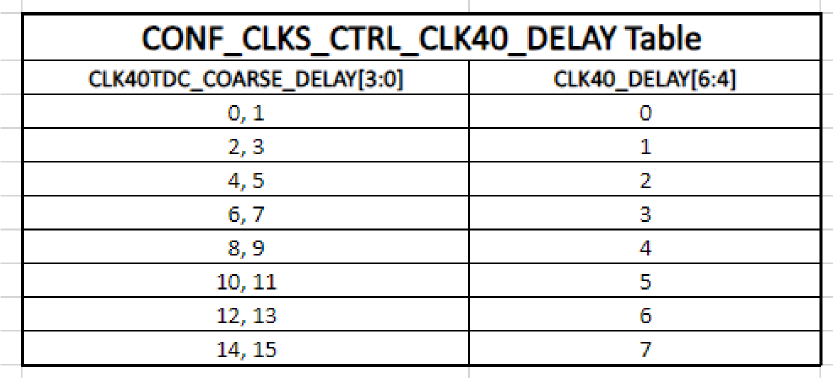
Phase Shifter delay (analog simulation = Verilog model)



UVM constrained randomization

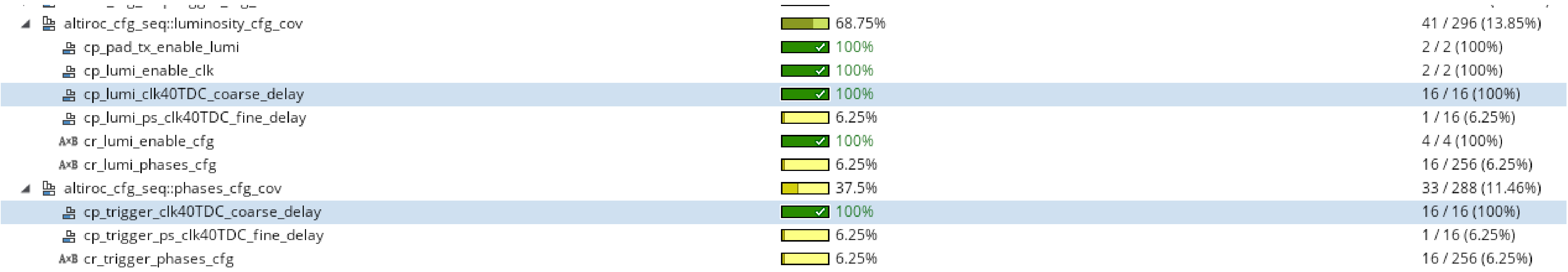
Constrained randomization:

- Moving the hit randomization with steps of 100 ps in a range of 25 ns (256 values)
- Constrain coarse delays and fine delays accordingly. For example, the phase for clk40M_TDC and clk40M_INT following the table:



CONF_CLKS_CTRL_CLK40_DELAY Table	
CLK40TDC_COARSE_DELAY[3:0]	CLK40_DELAY[6:4]
0, 1	0
2, 3	1
4, 5	2
6, 7	3
8, 9	4
10, 11	5
12, 13	6
14, 15	7

Metrics:



altiroc_cfg_seq::luminosity_cfg_cov	68.75%	41 / 296 (13.85%)
cp_pad_tx_enable_lumi	✓ 100%	2 / 2 (100%)
cp_lumi_enable_clk	✓ 100%	2 / 2 (100%)
cp_lumi_clk40TDC_coarse_delay	✓ 100%	16 / 16 (100%)
cp_lumi_ps_clk40TDC_fine_delay	6.25%	1 / 16 (6.25%)
cr_lumi_enable_cfg	✓ 100%	4 / 4 (100%)
cr_lumi_phases_cfg	6.25%	16 / 256 (6.25%)
altiroc_cfg_seq::phases_cfg_cov	37.5%	33 / 288 (11.46%)
cp_trigger_clk40TDC_coarse_delay	✓ 100%	16 / 16 (100%)
cp_trigger_ps_clk40TDC_fine_delay	6.25%	1 / 16 (6.25%)
cr_trigger_phases_cfg	6.25%	16 / 256 (6.25%)

Summary

Altiroc3 for ATLAS HGTD:

- 20 ps TOA meas. Resolution
- 40 ps TOT meas. resolution
- 100 ps luminosity meas. resolution

Main verification **challenges** and solutions:

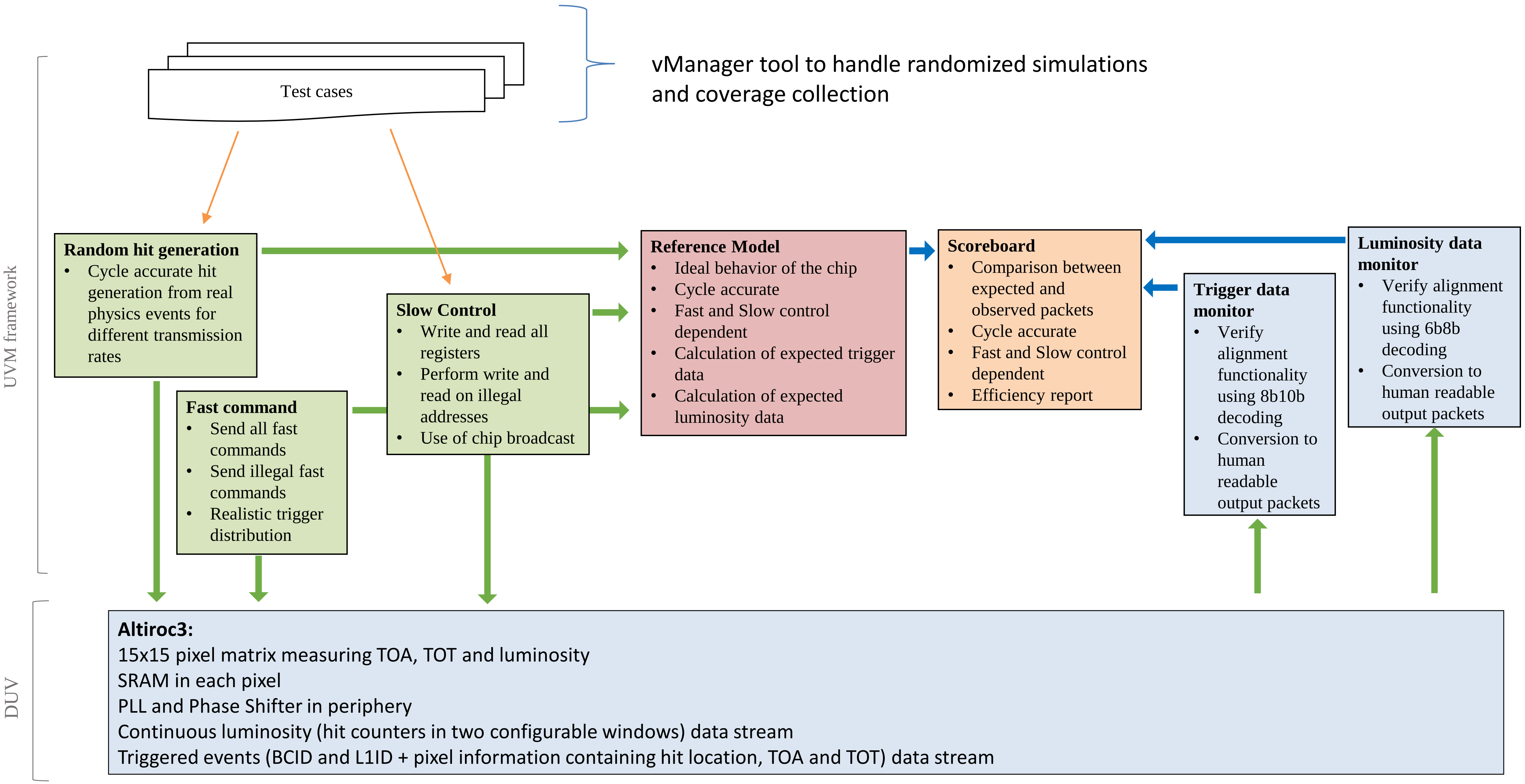
- **multi-site project** -> efficient source code management and version control
- **complex clock architecture** -> increased complexity of verification environment
- **critical analog macros** -> development of accurate Verilog models

Two **complementary** approaches to verify CDC:

- formal verification using JasperGold tool -> fast results, early stage (RTL)
- functional verification using UVM framework -> NETLIST level, check correctness

THANK YOU

UVM framework developed since September 2021



vManager tool to handle randomized simulations and coverage collection

Random hit generation

- Cycle accurate hit generation from real physics events for different transmission rates

Fast command

- Send all fast commands
- Send illegal fast commands
- Realistic trigger distribution

Slow Control

- Write and read all registers
- Perform write and read on illegal addresses
- Use of chip broadcast

Reference Model

- Ideal behavior of the chip
- Cycle accurate
- Fast and Slow control dependent
- Calculation of expected trigger data
- Calculation of expected luminosity data

Scoreboard

- Comparison between expected and observed packets
- Cycle accurate
- Fast and Slow control dependent
- Efficiency report

Trigger data monitor

- Verify alignment functionality using 8b10b decoding
- Conversion to human readable output packets

Luminosity data monitor

- Verify alignment functionality using 6b8b decoding
- Conversion to human readable output packets

Altiroc3:

15x15 pixel matrix measuring TOA, TOT and luminosity

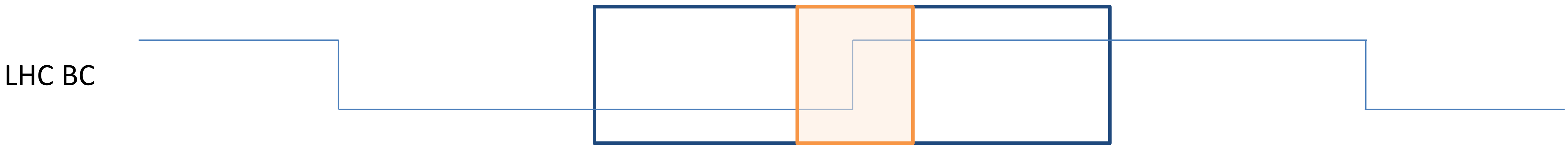
SRAM in each pixel

PLL and Phase Shifter in periphery

Continuous luminosity (hit counters in two configurable windows) data stream

Triggered events (BCID and L1ID + pixel information containing hit location, TOA and TOT) data stream

Hit generation



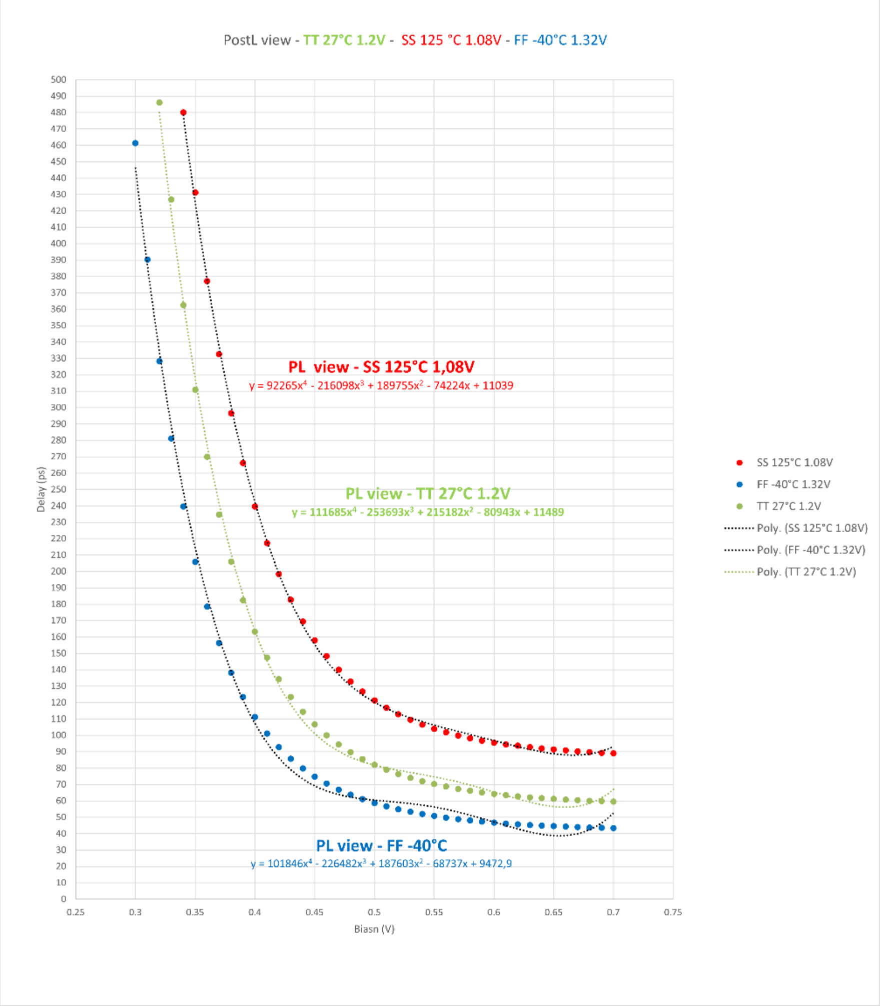
Hit injection over 12.5 ns:

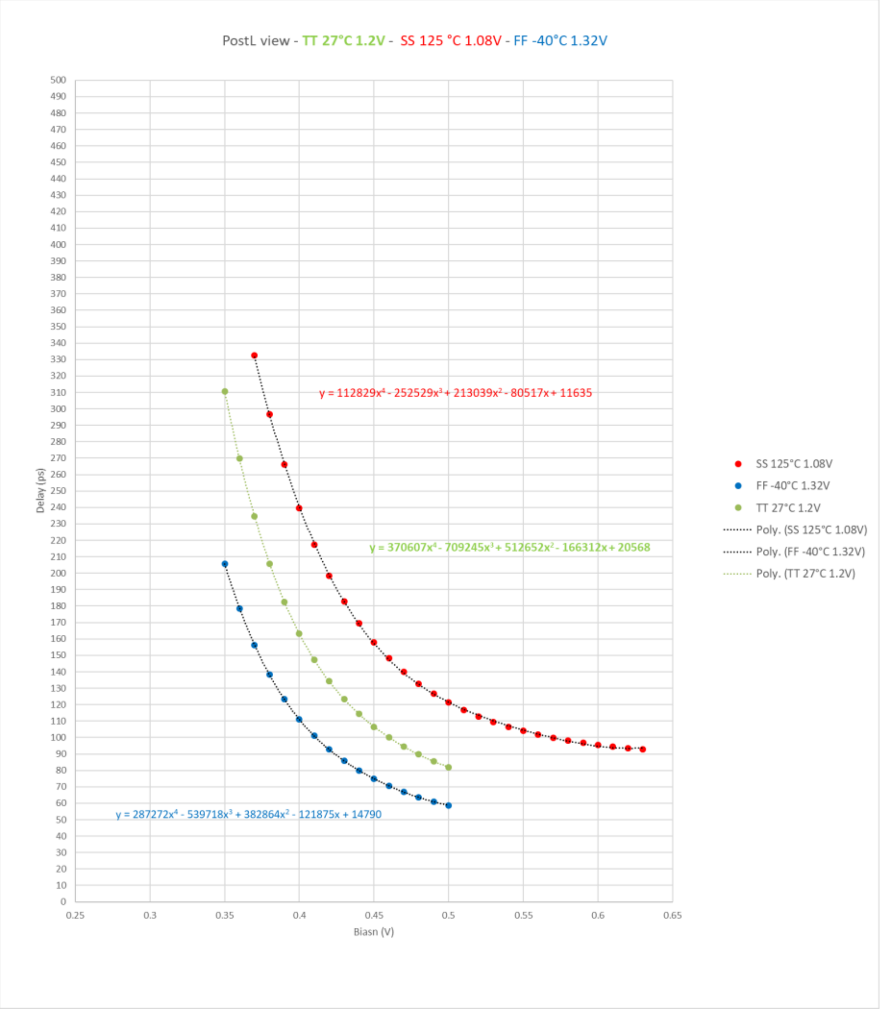
- Charges are injected around the BC posedge over 3.125 ns (number controlled with occupancy)
  - In reality the occupancy can go up to 35%: https://aleopold.web.cern.ch/aleopold/hgtd/hgtd_asic_simul_hitdistros_log/

- Few charges injected in the larger window of 12.5 ns to model afterglow effect (at maximum 5% of occupancy)

Phase Shifter Verilog model

Analog simulations to find relationship between voltage bias and delay in ps:

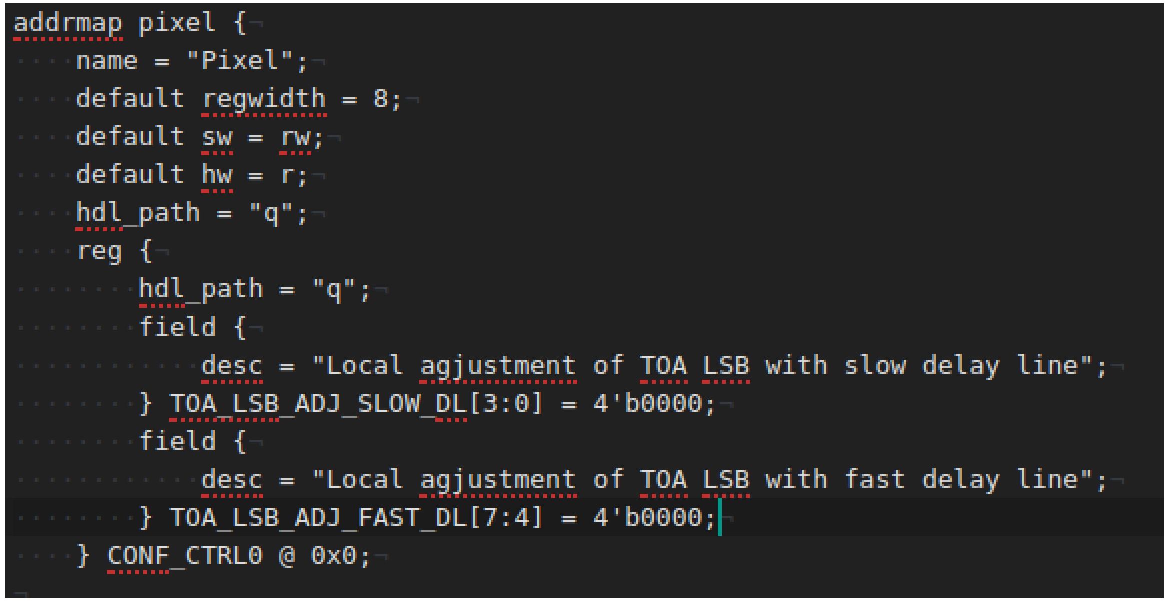




SystemRDL language

‘The ***SystemRDL*** *language, supported by the SPIRIT Consortium, was specifically designed to describe and implement a wide variety of* control status registers*. Using SystemRDL, developers can automatically* ***generate*** *and* ***synchronize*** *register views for specification, hardware design, software development, verification, and documentation.’* (Wikipedia)

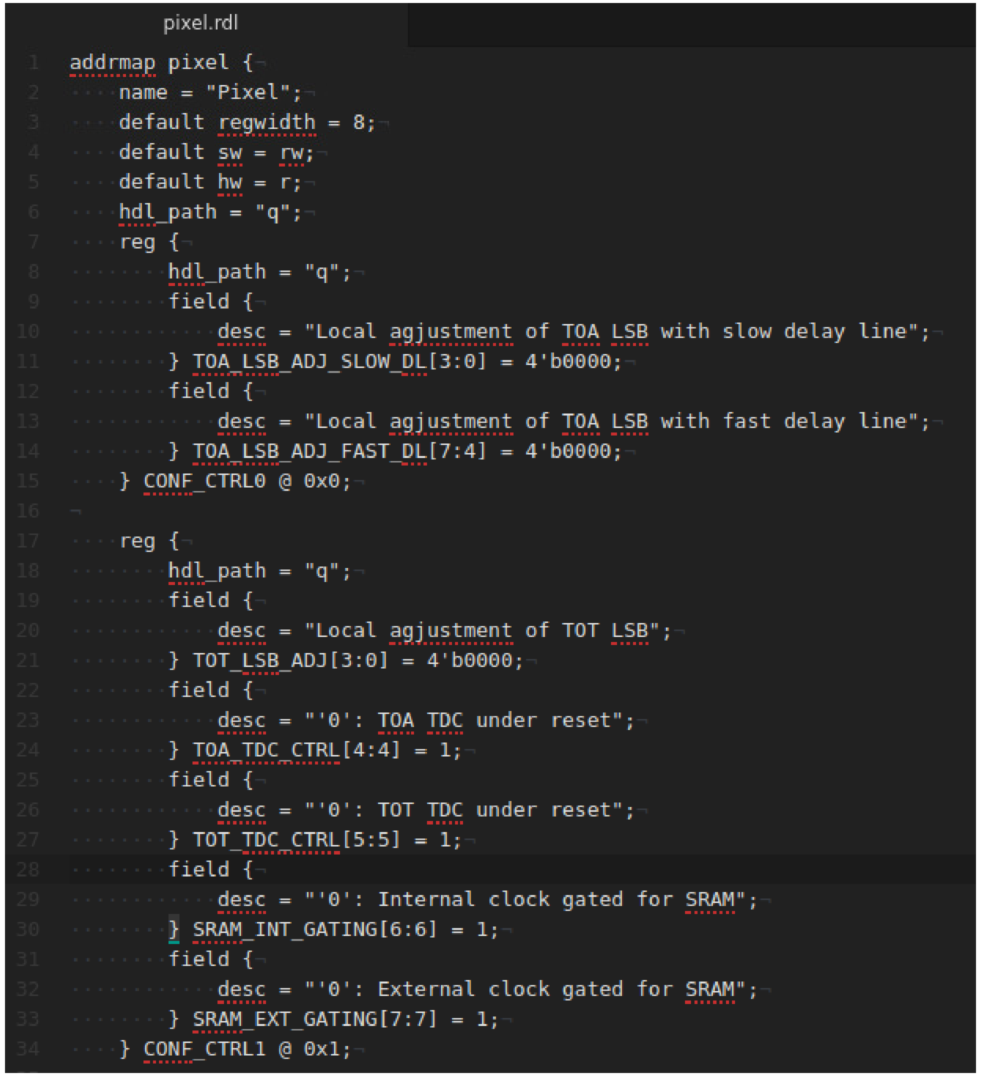
Example:



```
addrmap pixel {
    name = "Pixel";
    default regwidth = 8;
    default sw = rw;
    default hw = r;
    hdl_path = "q";
    reg {
        hdl_path = "q";
        field {
            desc = "Local agjustment of TOA LSB with slow delay line";
        } TOA_LSB_ADJ_SLOW_DL[3:0] = 4'b0000;
        field {
            desc = "Local agjustment of TOA LSB with fast delay line";
        } TOA_LSB_ADJ_FAST_DL[7:4] = 4'b0000;
    } CONF_CTRL0 @ 0x0;
```

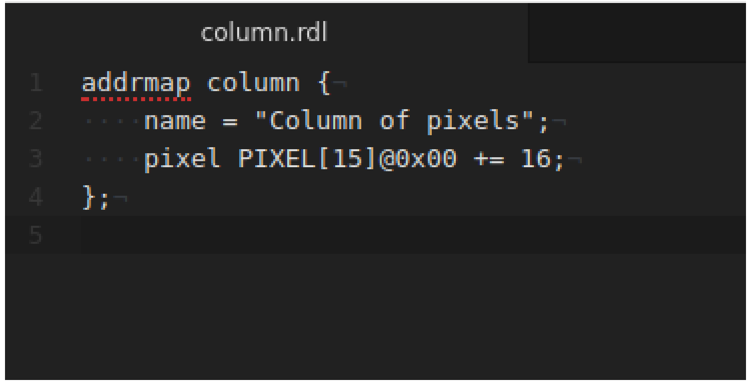
RDL files hierarchy

pixel.rdl



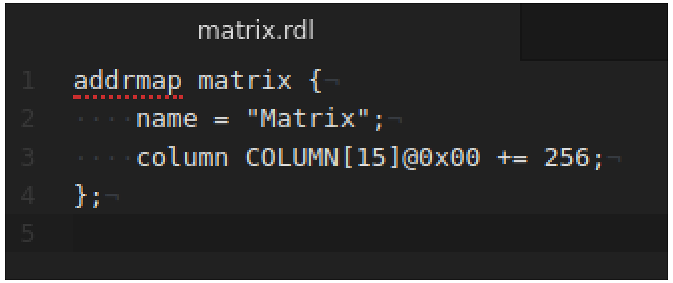
```
addrmap pixel {
    name = "Pixel";
    default regwidth = 8;
    default sw = rw;
    default hw = r;
    hdl_path = "q";
    reg {
        hdl_path = "q";
        field {
            desc = "Local agjustment of TOA LSB with slow delay line";
        } TOA_LSB_ADJ_SLOW_DL[3:0] = 4'b0000;
        field {
            desc = "Local agjustment of TOA LSB with fast delay line";
        } TOA_LSB_ADJ_FAST_DL[7:4] = 4'b0000;
    } CONF_CTRL0 @ 0x0;

    reg {
        hdl_path = "q";
        field {
            desc = "Local agjustment of TOT LSB";
        } TOT_LSB_ADJ[3:0] = 4'b0000;
        field {
            desc = "'0': TOA TDC under reset";
        } TOA_TDC_CTRL[4:4] = 1;
        field {
            desc = "'0': TOT TDC under reset";
        } TOT_TDC_CTRL[5:5] = 1;
        field {
            desc = "'0': Internal clock gated for SRAM";
        } SRAM_INT_GATING[6:6] = 1;
        field {
            desc = "'0': External clock gated for SRAM";
        } SRAM_EXT_GATING[7:7] = 1;
    } CONF_CTRL1 @ 0x1;
```

column.rdl



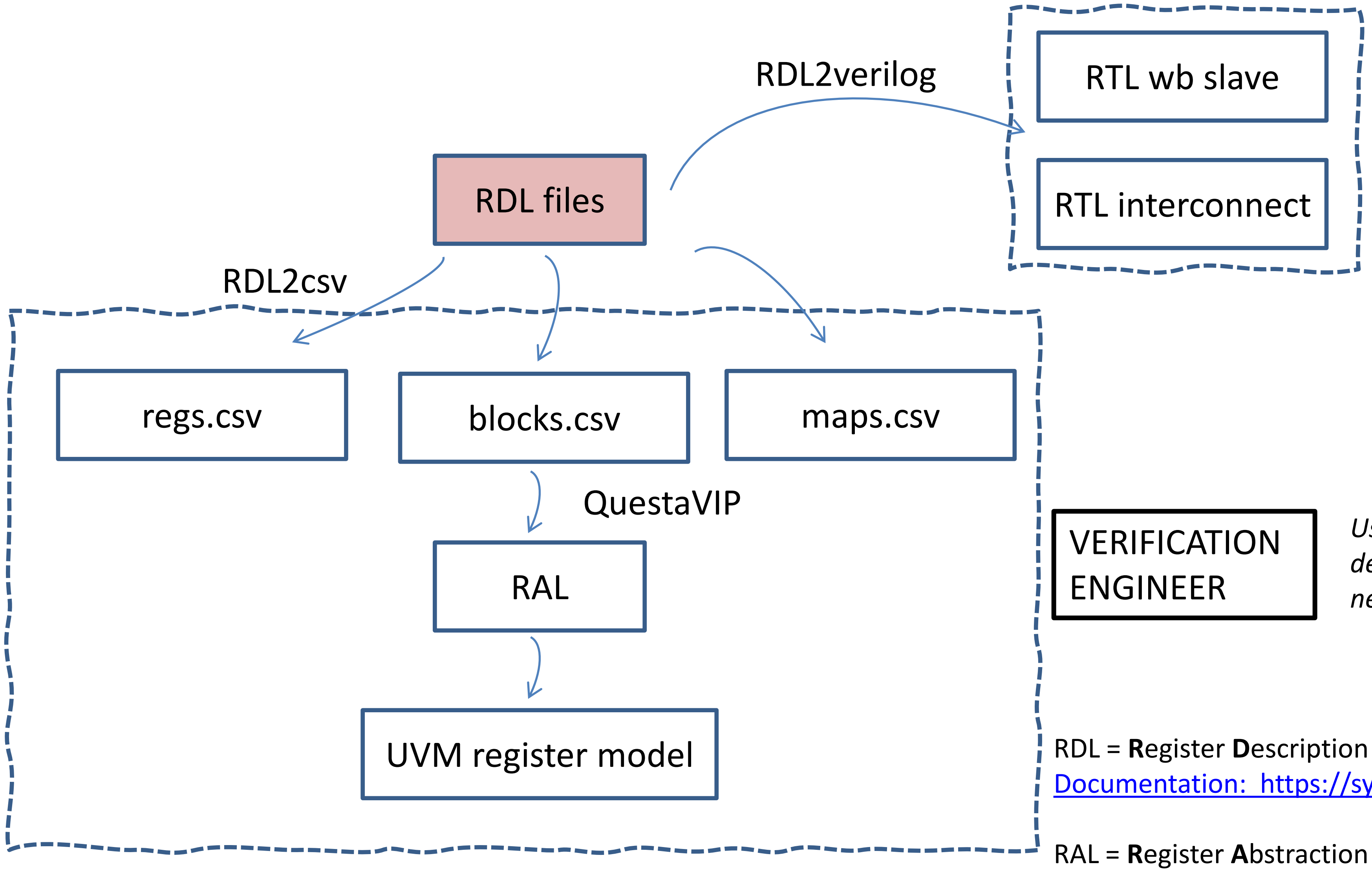
```
addrmap column {
    name = "Column of pixels";
    pixel PIXEL[15]@0x00 += 16;
};
```

matrix.rdl



```
addrmap matrix {
    name = "Matrix";
    column COLUMN[15]@0x00 += 256;
};
```

Flow for I2C registers



RTL DESIGNER

Modifies only the RDL files.
All the other files are automatically derived by the RDL

VERIFICATION ENGINEER

Uses the same RDL used by the designer to generate what is needed for verification

RDL = **R**egister **D**escription **L**anguage
Documentation: https://systemrdl-compiler.readthedocs.io/en/latest/

RAL = **R**egister **A**bstraction **L**ayer

RTL TMR: SEU injection results

SEU verification strategy:

- I2C test case
- Running mode test case

I2C test case:

- I2C operations never fail due to SEU and configuration is never lost in the ASIC

Running mode test case has four main categories:

- Triplicated registers -> No issues
- Untriplicated registers -> Reduced efficiency, but no synchronization issues
- Triplicated FIFOs -> No issues
- Untripliated FIFOs -> Reduced efficiency, but no synchronization issues

⬇

We can still have issues in PNR due to low distance among registers or simplification
SEU run only on digital part of the chip, not on analog blocks