Pix-ESL: a SystemC framework for architectural modelling of readout systems in HEP

[1]CERN, 1211 Geneva 23, Switzerland

Jashandeep Dhaliwal[1], Francesco Enrico Brambilla[1], Davide Ceresa[1], Stefano Esposito[1]

Contact info: **jashandeep.dhaliwal@cern.ch**

Abstract

The high cost of prototyping at advanced technology nodes, as well as the complexity of future detectors, necessitate the use of a system design technique widely used in industry: design space exploration through high-level architecture studies to establish precise and optimal requirements. This work presents Pix-ESL: a programmable SystemC framework for simulating the readout chain from the front-end chips to the detector back-end. The model is transaction accurate, comprises an event generator and connects with real-world physics events, and provides metrics such as readout efficiency, latency, and average queue occupancy. This contribution outlines the framework's structure as well as a case study based on Velopix2.

1. Objective

The development of future pixel-based detectors requires high-level modelling, from Front-End to Back-End, to:

- Perform **architectural studies** and **extract basic metrics**
- Provide a **reference model** for verification
- Support **RTL co-simulation**
- Perform **on-chip data processing**
- Estimate **power consumption** and **area cost**

2. Methodology

The proposed framework has the following characteristics:

- The design language is **SystemC**: a C++ library for architecture modelling standard
- The coding style is **approximately-timed** to allow architectural exploration: the components of a system adhere to a common time base and execute in synchronous
- The model is **clock-driven**
- Communication is based on an **always-pull configuration**
- Packet transfers are based on Transaction Level Modelling (**TLM2.0**) sockets
- The structure enables **modularity** and **code re-use**

Presented at the 7th Barcelona Techno Week Course on semiconductor radiation detectors July 3 – 7, 2023

3. Model description

The **stimuli** as input data can come from:

- **Event generator**: internally generated events for parametrized studies
- **External file**: external data deriving from physics-level detector simulations

The **transfer function** can convert an analog input into a digital packet, **model the front-end** and **perform the injection**.

The **SystemC model**, which **simulates the readout dataflow**, is based on two main components:

- **Layer**: contains the processing and readout modules. A basic module can store data and communicate with other modules as well as perform arbitration and routing
- **Network**: instantiates the connections between same-layer modules (intra-layer) and different-layer modules (inter-layer)

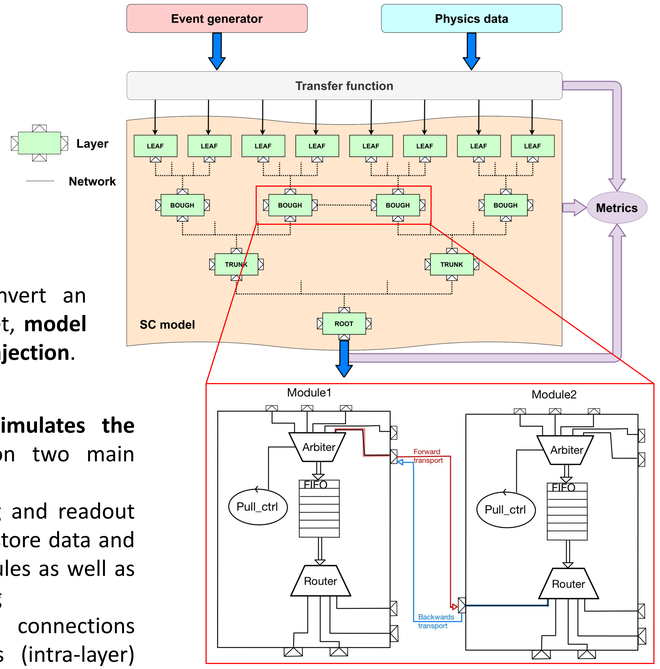

Figure 1: Scheme of the framework and SystemC model

The **metrics** collect information across the model to compute **readout efficiency**, **latency**, and **average queue occupancy**.

Packet transfer is based on **pull requests** in **TLM2.0** issued by an **initiator socket** of a module to a **target socket** of another module. Each module of a layer has both initiator and target sockets with specific processes.

5. Future outlooks

The **roadmap** plans to:

- **Generalize layers** and **network** generation functions
- Expand **TLM2.0** support
- Provide **libraries** for **transaction protocols**, **routers**, **arbiters** and **memory** elements
- Include **on-chip processing stages** to reduce the required bandwidth and off-chip processing
- Set up a **python-based tool-set for validation** and **analysis**
- Prototype and evaluate an **event-driven model**
- Add **hardware** and **power metrics**
- Expand the communication with an **always-push configuration**

4. Velopix2 case-study

Velopix2 is an LHCb VELO detector proposed for Phase-II Upgrade, to be installed during the LHC Long Shutdown 4; it has been chosen as the first case of study focused mainly on the **optimization of data-driven readout architecture**.

The architecture, shown in figure 2, is subdivided into 4 layers: **pixels**, **super-pixels (SP)**, **regions**, and **end-of-column (EoC) nodes**.

Velopix2 was parametrically studied on **four architectural configurations** with a **simulation time of 4000 BX cycles** (1000 warm-up + 2000 injection + 1000 cool-down).

	Config. 1	Config. 2	Config. 3	Config. 4
Readout eff.	91.43 %	99.88 %	99.98 %	99.98 %
Avg. latency	139 cy.	39 cy.	44 cy.	37 cy.
Max. latency	2965 cy.	2081 cy.	2079 cy.	1847 cy.

Table 2: Summary table of the four architectural configurations with physics data as the input source

In *config 1 (base configuration of table 1)*, the analysis revealed a **high occupancy of the region FIFO** which has been increased from 2 to 4 in *config 2 (config 1 + region FIFO depth of 4)*. Furthermore, *config 3 (config 2 + more priority of top regions)* changed the arbitration of the regions to **enhance the priority of the top regions to match the pixel occupancy distribution**. Finally, to **reduce area and power consumption**, *config 4 (config 3 + taller regions)* **reduces the number of regions per column** (from 16 to 8) which shows **lower average and maximum latencies without affecting the readout efficiency**.

The **development** of the **Pix-ESL framework** and the **high-level model of Velopix2** required **~2 months** of work and shows a **favourable runtime** compared to RTL simulation: the **architectural build** requires **~10 s** while the **simulation speed** is **~15 BX/s**.

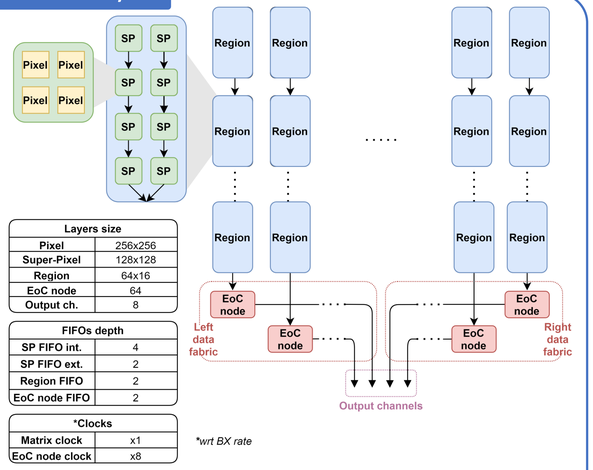

Table 1, Figure 2: Sizing and schematic of Velopix2 base architecture.

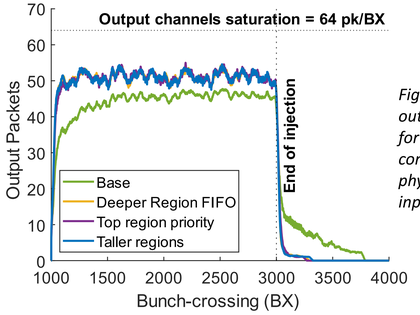

Figure 3: Number of output packets per BX for the four configurations, using physics data as the input source.

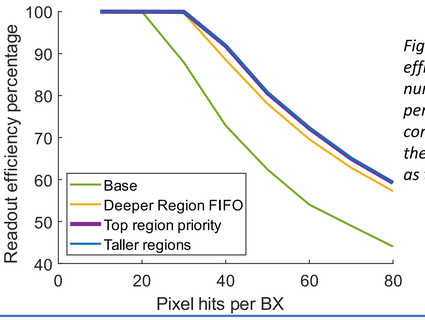

Figure 4: Readout efficiency VS input number of pixel hits per BX for the four configurations, using the Event Generator as the input source.