

A Journey to DSP + HWA Integration

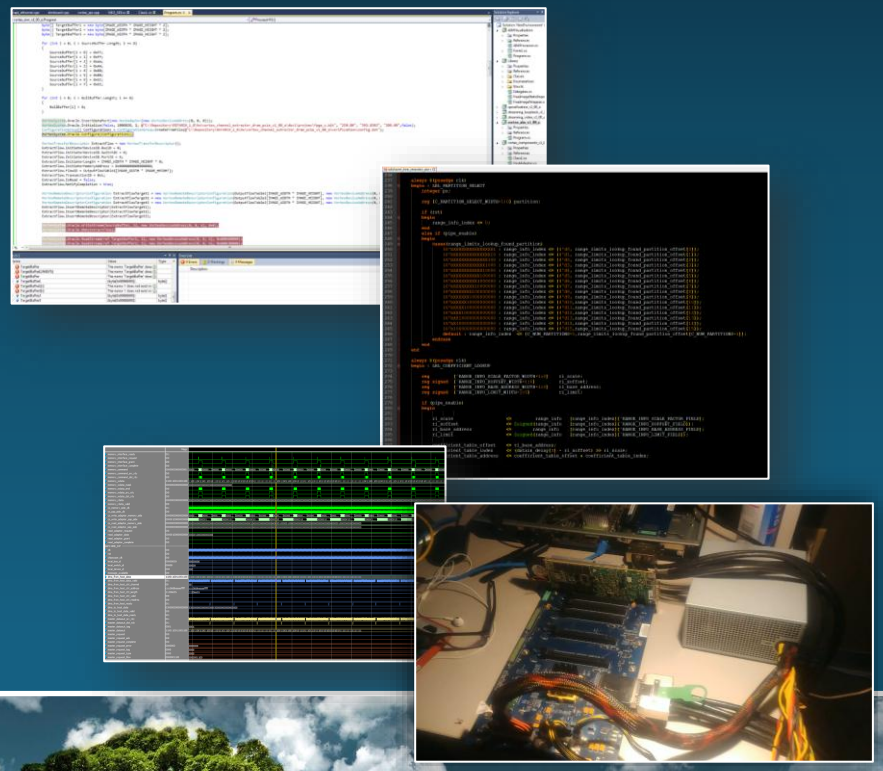
Presented by:
Dr. Kevin M. Irick, Founder & CEO
kirick@siliconscapes.net

Co-Authored by :
- Jimil Irick
- Kamil Amanowicz

SILICONscapes
bridging the gap between hardware and software

SiliconScapes

Hardware and software design consultancy with over 20 years of experience.

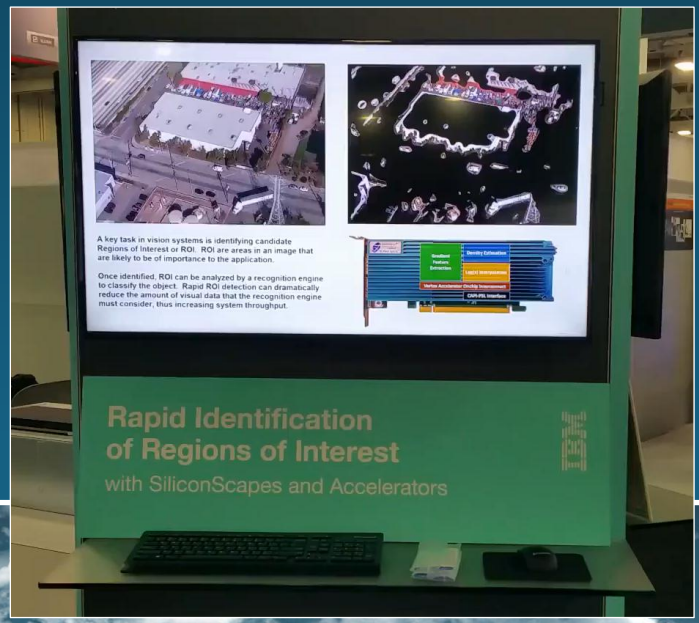


Who We Are

We are hardware and software engineers with the skill and commitment to keep our clients competitive. Whether creating custom hardware to accelerate algorithms or optimizing firmware to maximize system performance, our expertise and commitment to excellence is unmatched.

Our Expertise

- RTL Development
- Hardware Accelerator Design
- FPGA Prototyping
- Firmware Development
- Virtual Platform Development
- Silicon Emulation & Validation



www.siliconscapes.net/contact

SILICON scapes
bridging the gap between hardware and software

The Performance and Flexibility of Tensilica DSP

- ✓ DSP centric platform offers the flexibility of a software programmable system with microarchitecture and ISA features tuned for demanding signal processing applications
- ✓ Highly configurable base architecture allows for true bespoke processor design that aligns with system performance and footprint constraints
- ✓ Extensible ISA provides a path to domain specific optimization using the TIE design flow, significantly de-risking the silicon design process
- ✓ **Provides well defined, highly-configurable external interfaces that allow coupling of Hardware Accelerators (HWA) for extreme performance and performance/watt gains**

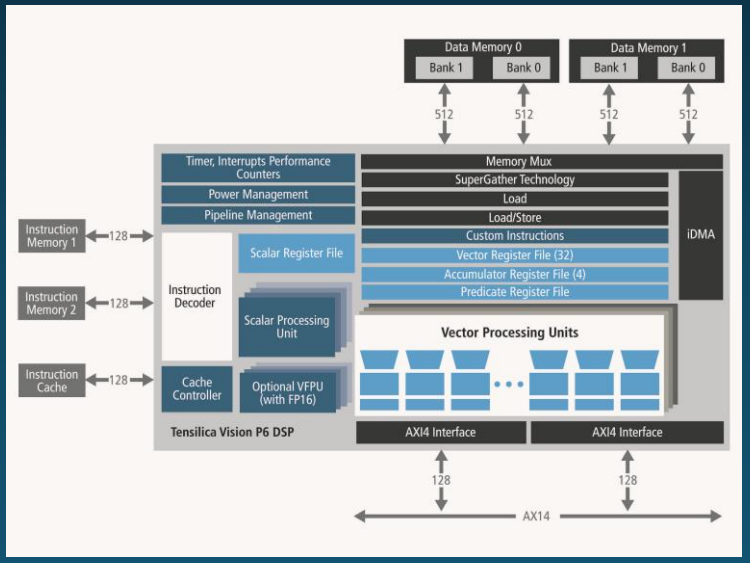
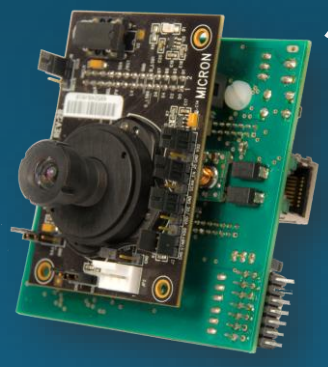
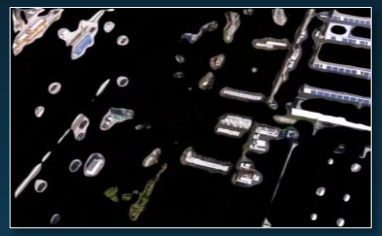


Image coutersy Cadence

Example – Smart Camera ICP (Image Content Processor)



Optical Image Saliency Accelerator



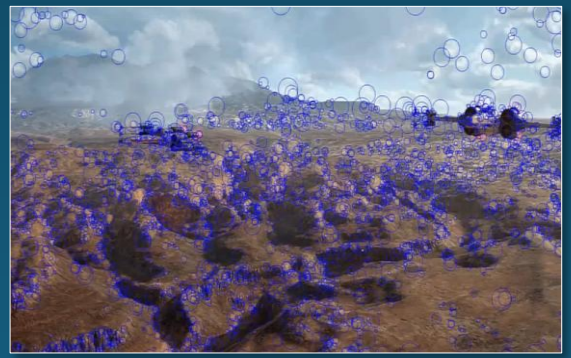
Identify regions of interest by generating an image “Saliency Map” by computing Log Likelihoods across a large number of learned featured maps. Firmware can *randomly* index this map to answer the query, “Is this pixel interesting”?

Pixel Tone Filtering Accelerator

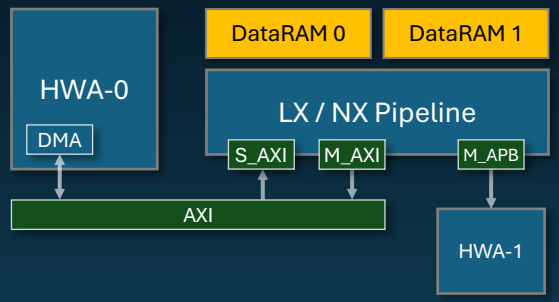


Identify pixels of interest by applying non-linear pixel tone classification. The ICP will process a frame buffer stored in Tightly Coupled Memory and generate a bitmap corresponding to pixels falling within a color range.

Interest Point Detection Accelerator

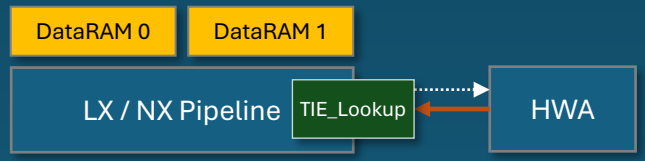


Detect scale and rotationally invariant pixel patches for image registration & stabilization, object matching and object tracking. The ICP can invoke this accelerator on frames that are buffered onchip or directly from the incoming image stream. Moreover this accelerator can be made available to software running on the main APU



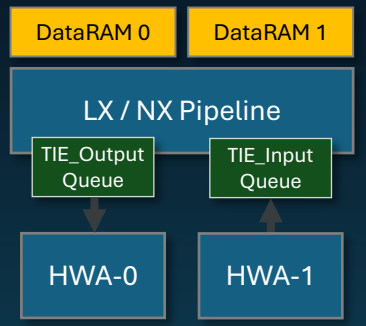
Shared Bus Attachments – provides a familiar and standard interface for accelerators that can be shared by multiple managing agents in the system. These accelerators work on large blocks of data and operate autonomously once configured and initiated. Tensilica processors feature all of the standard interfaces for shared bus accelerators, including AXI4, ACE, and APB.

- Macro Accelerators
- Coarse Shared Functional Units
- Legacy Accelerators



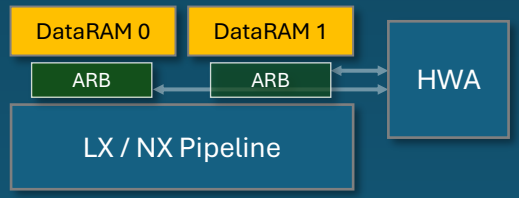
TIE Lookups - provide a fixed-latency, fully-pipelined lookup table style interface. Software issues lookup requests that appear at the input of the hardware accelerator. The accelerator performs the operation and returns the result after K latency cycles. The compiler is aware of the latency of each Lookup interface and is able to schedule for the return of the result.

- Latency tolerant memory attachments
- LUT based function approximation
- Histogram engines



TIE Queues - provide high-bandwidth data ingress and egress directly to/from the DSP pipeline from/to a hardware accelerator. Software interacts with TIE_Queue attached accelerators using simple push/pop operations. The accelerator and the DSP pipeline handle flow control through FIFO style handshake.

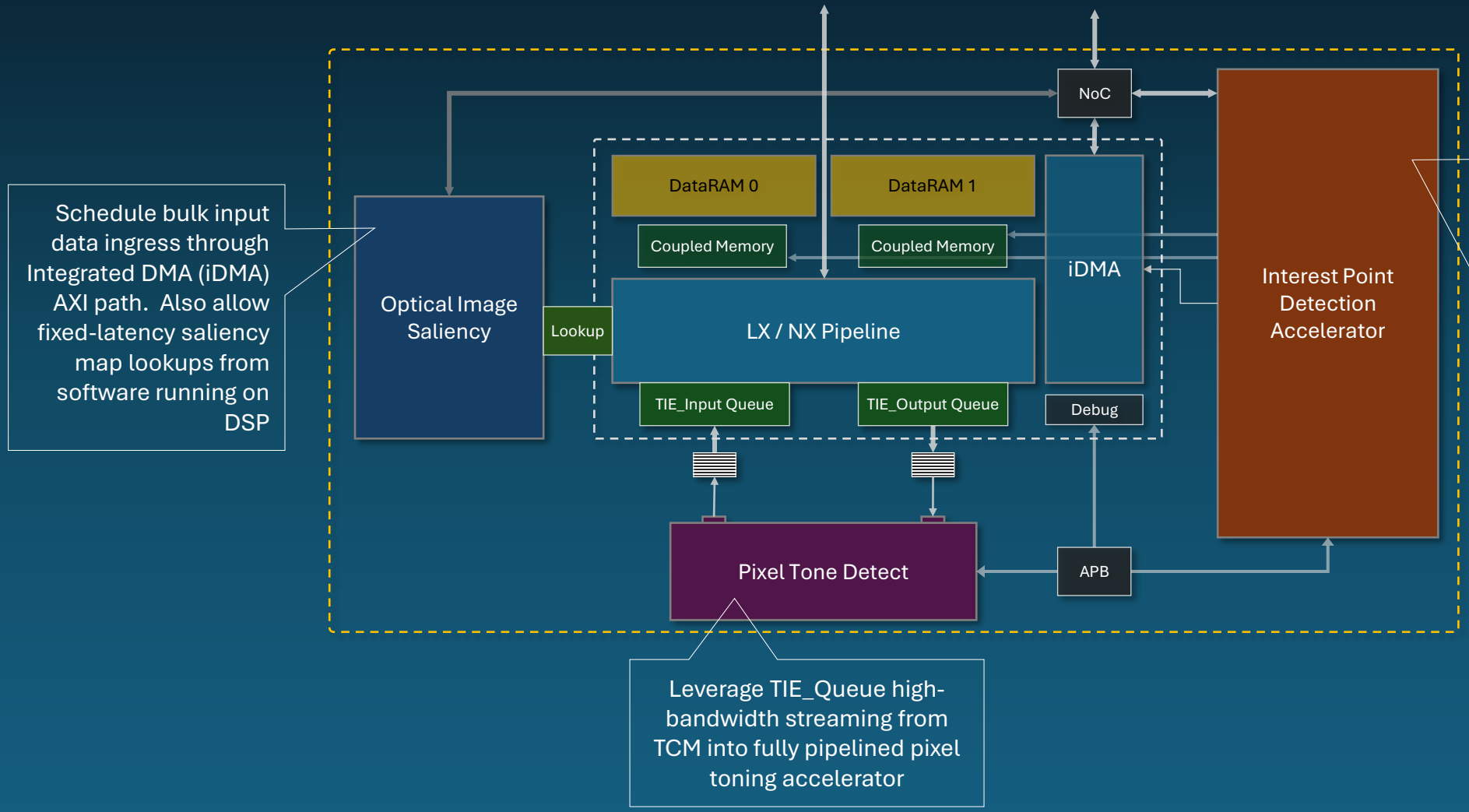
- Tightly coupled stream operators
- Command and work queues
- Wide vector coprocessors



Memory Coupled Attachments – provides a backdoor means for an accelerator to share access to the Tightly Coupled Memory (TCM) of a Tensilica core. Accelerators that operate autonomously on larger working sets can populate TCM in tandem with the normal operation of the core. The core can subsequently post-process the data from the fastest accessible memory.

- Tight hardware/software dataflows
- Accelerators that generate sparse results

Smart Camera ICP – Choosing Accelerator Interfaces



Schedule bulk input data ingress through Integrated DMA (iDMA) AXI path. Also allow fixed-latency saliency map lookups from software running on DSP

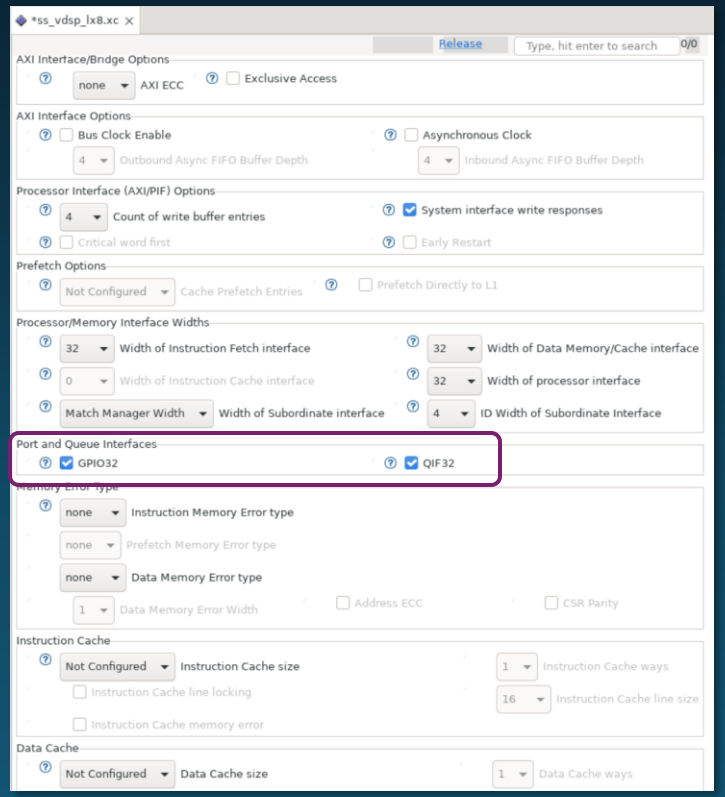
Leverage TIE_Queue high-bandwidth streaming from TCM into fully pipelined pixel toning accelerator

Software will configure this accelerator through a lightweight APB interface. The accelerator will initiate pre-programmed iDMA transactions via a TriggerIn interface.

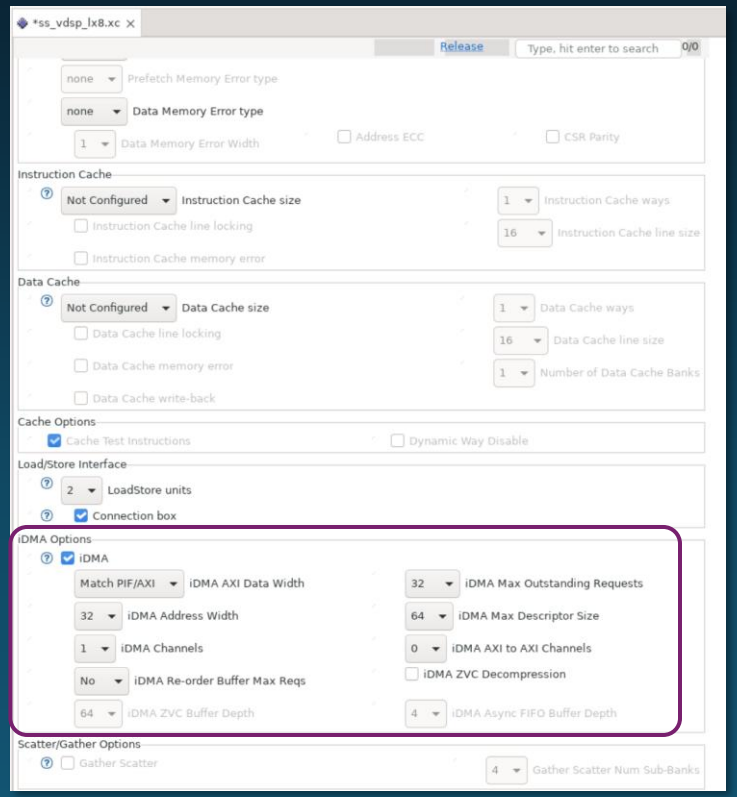
As this accelerator generates results it will write those results into the TCM via the backdoor "Coupled Memory" interface.

This "bus shared" accelerator will also be accessible by software running on the Application Processing Unit (APU).

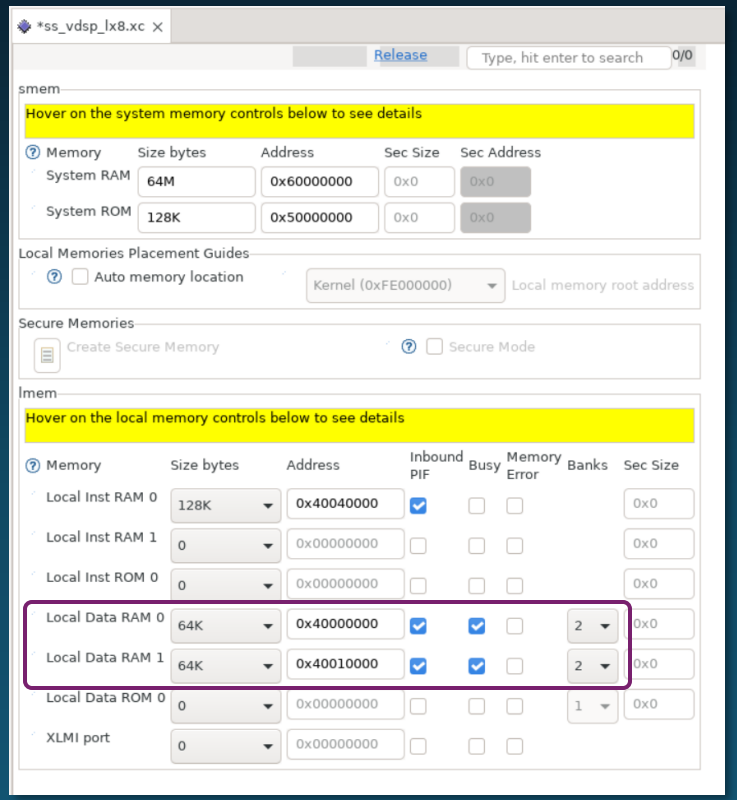
Smart Camera ICP – Configuring the Core



Selecting TIE_Queue interfaces. These are general purpose 32-bit interfaces and sufficient for our use case. Custom bit-width and deeper integration with custom ISA operations can be achieved by writing custom TIE.

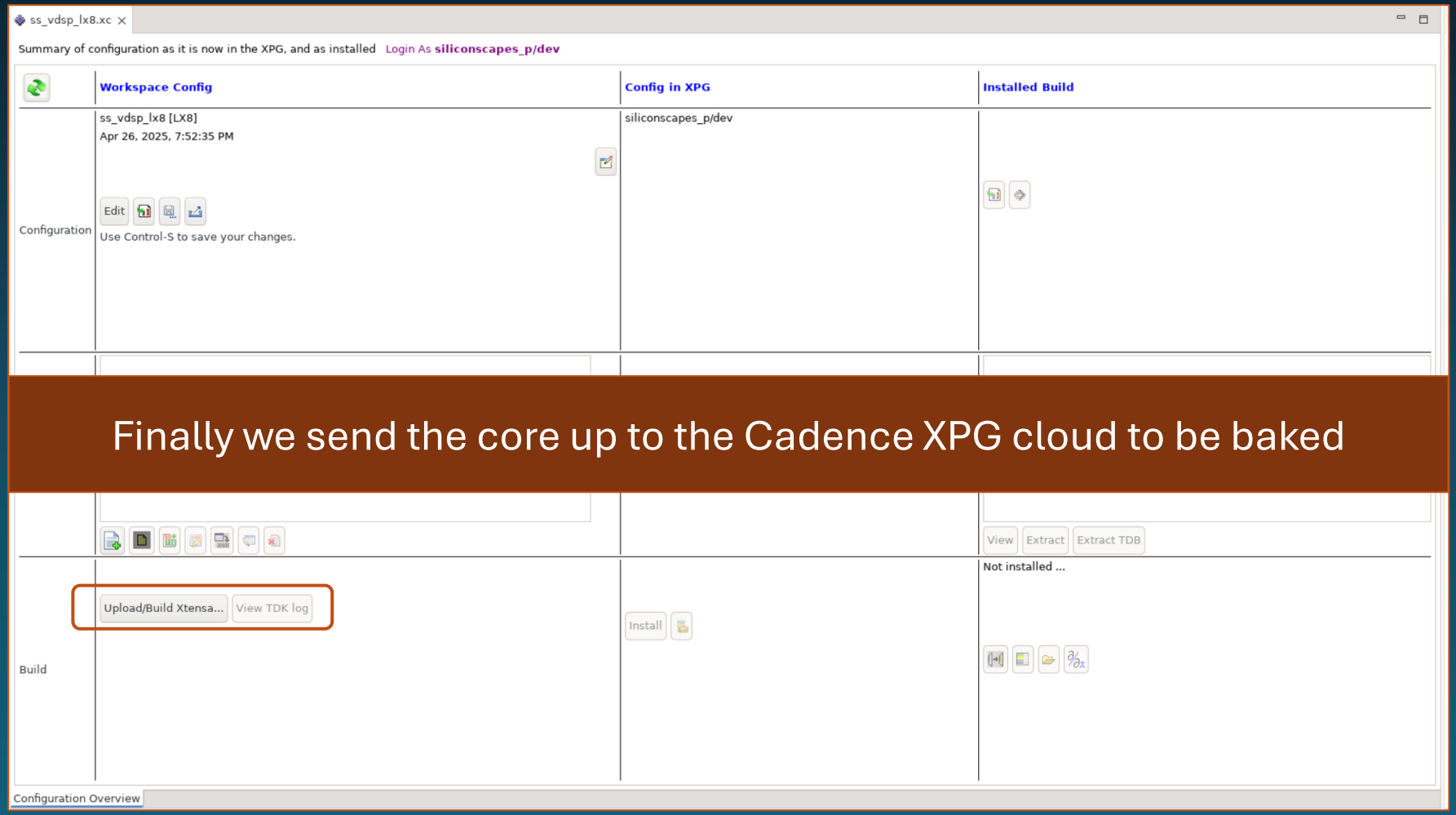


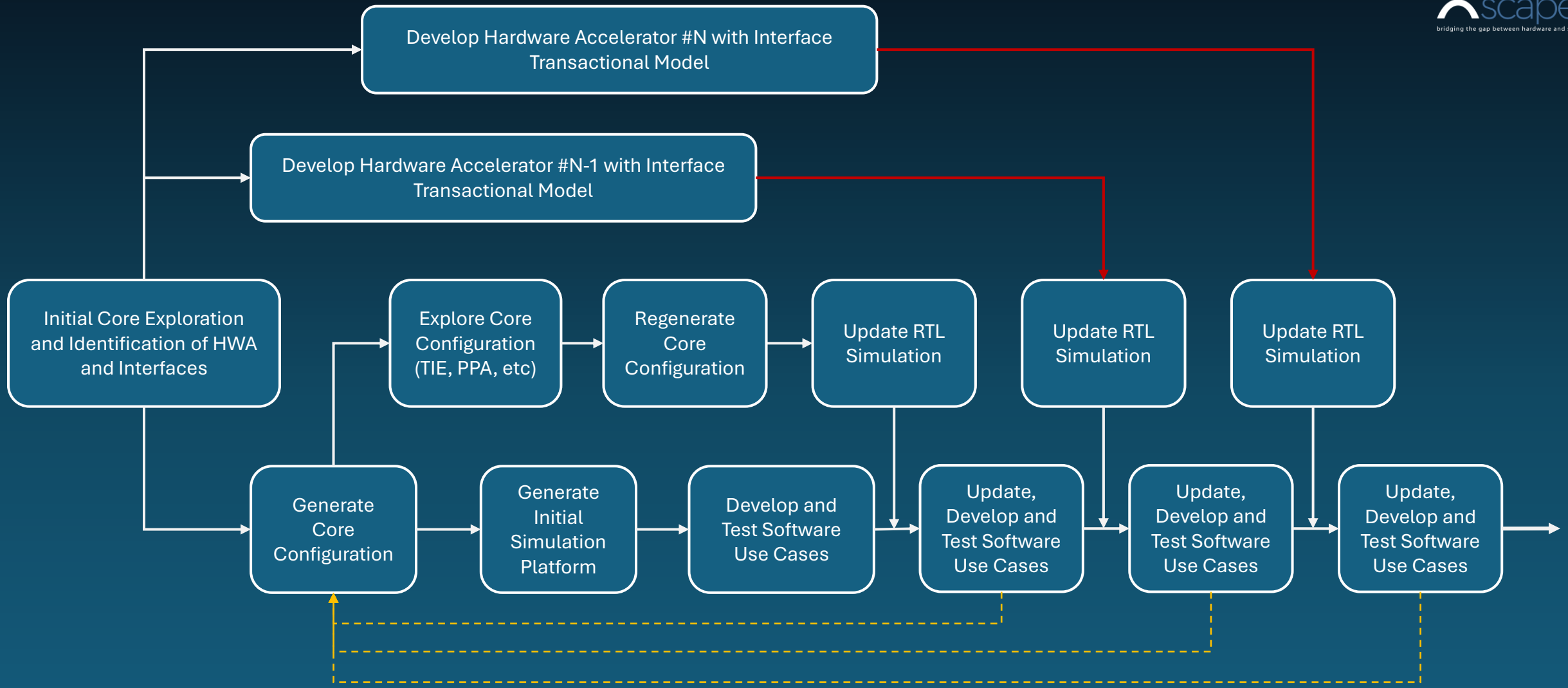
We configure iDMA to minimize complexity and duplication of Bus Managing functionality of our shared bus accelerators. iDMA is a feature rich component that can be scaled according to requirements and silicon budget.



The DataRAMs (TCM) are configured with a “Busy” interface that allow our accelerator to “steal” memory access from the core. We choose two DataRAMs, each with two banks to mitigate bank conflicts via static bank allocation.

Smart Camera ICP – Configuring the Core





IP integration needs to be nimble to changes triggered by exploration outcomes.

IP integration design needs to support parallel design flows with unaligned timelines.

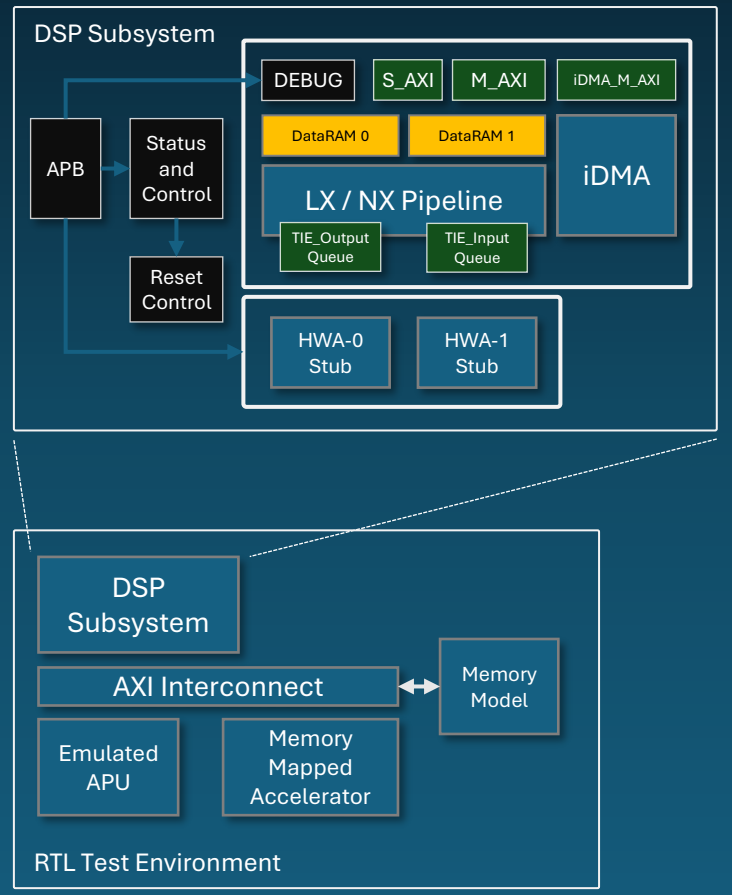
Hardware System Design needs to be "software aware" increasingly early in the design cycle

Smart Camera ICP – Integration Automation

SILICON scapes
bridging the gap between hardware and software

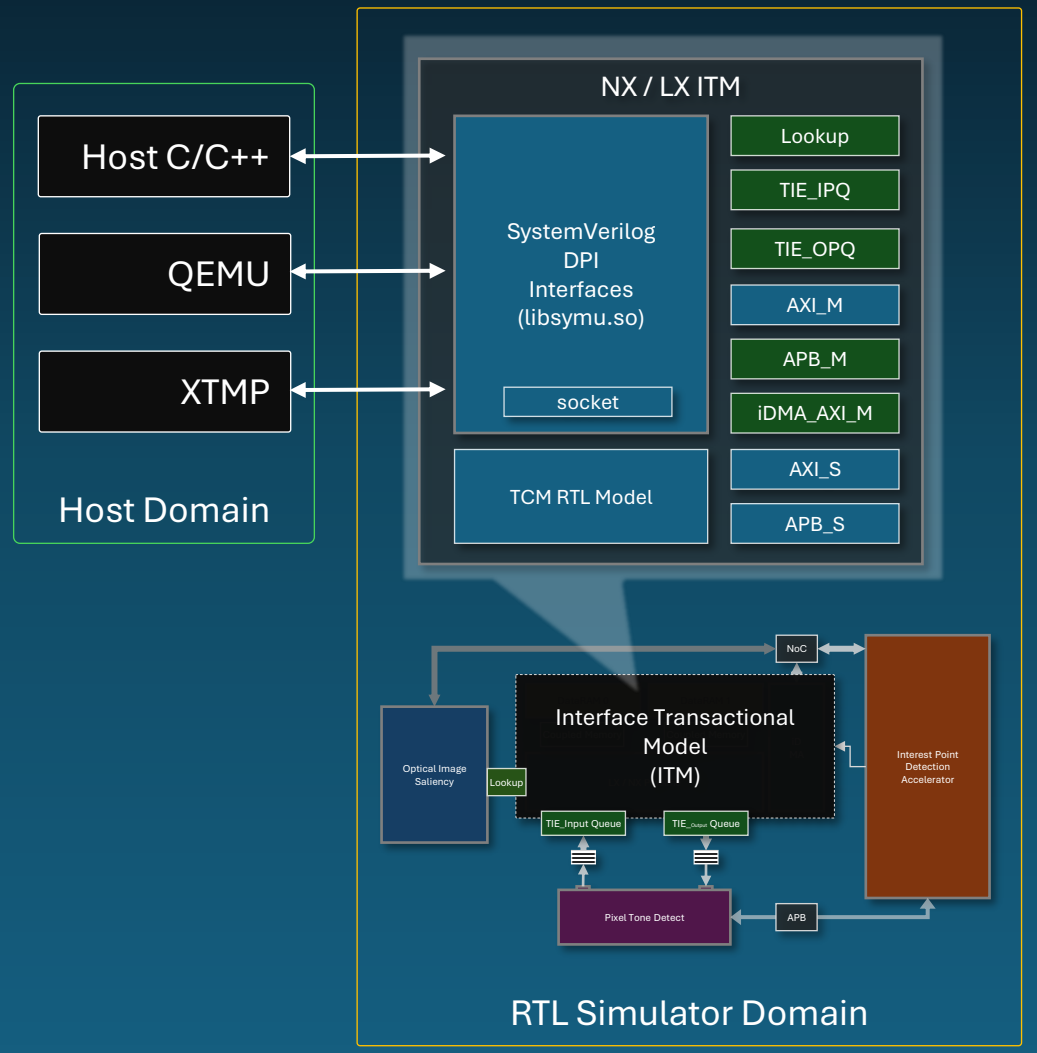
Core Config



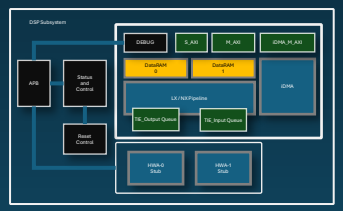


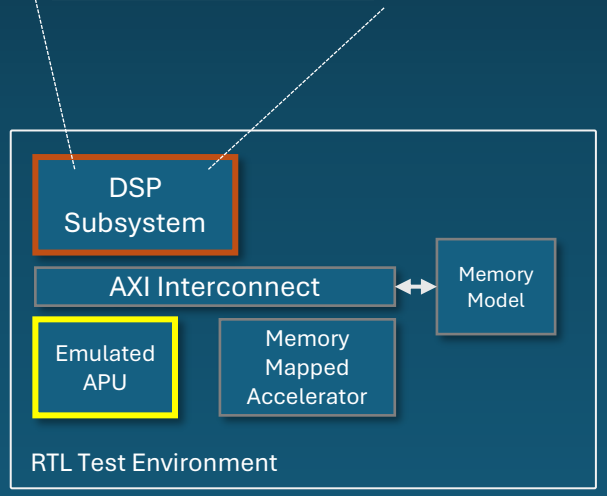
Hybrid RTL and Transactional Modeling

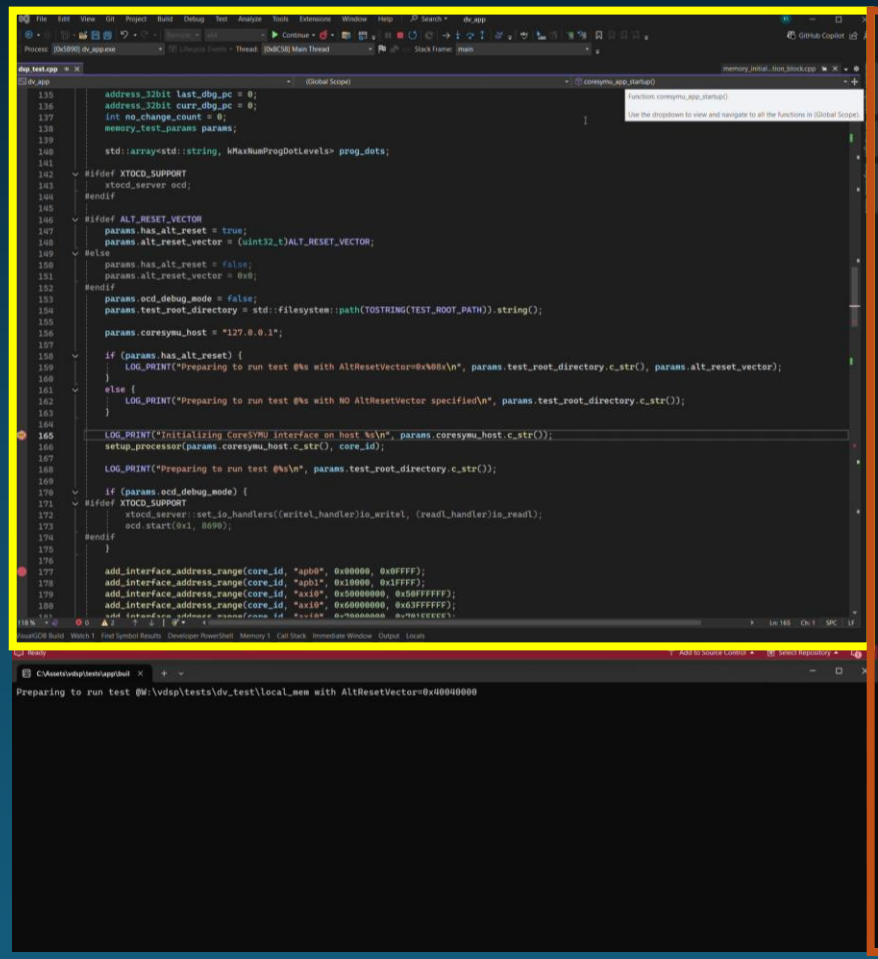
- RTL models when cycle-level behavior is critical
- Transactional BFMs when interface functional modeling is sufficient
  - Leverage SystemVerilog DPI to model functionality in C/C++
  - Leverage SystemVerilog DPI to bridge RTL simulation and C/C++ execution domains
- Offers an incremental approach to “building up” the system as accelerator RTL becomes available

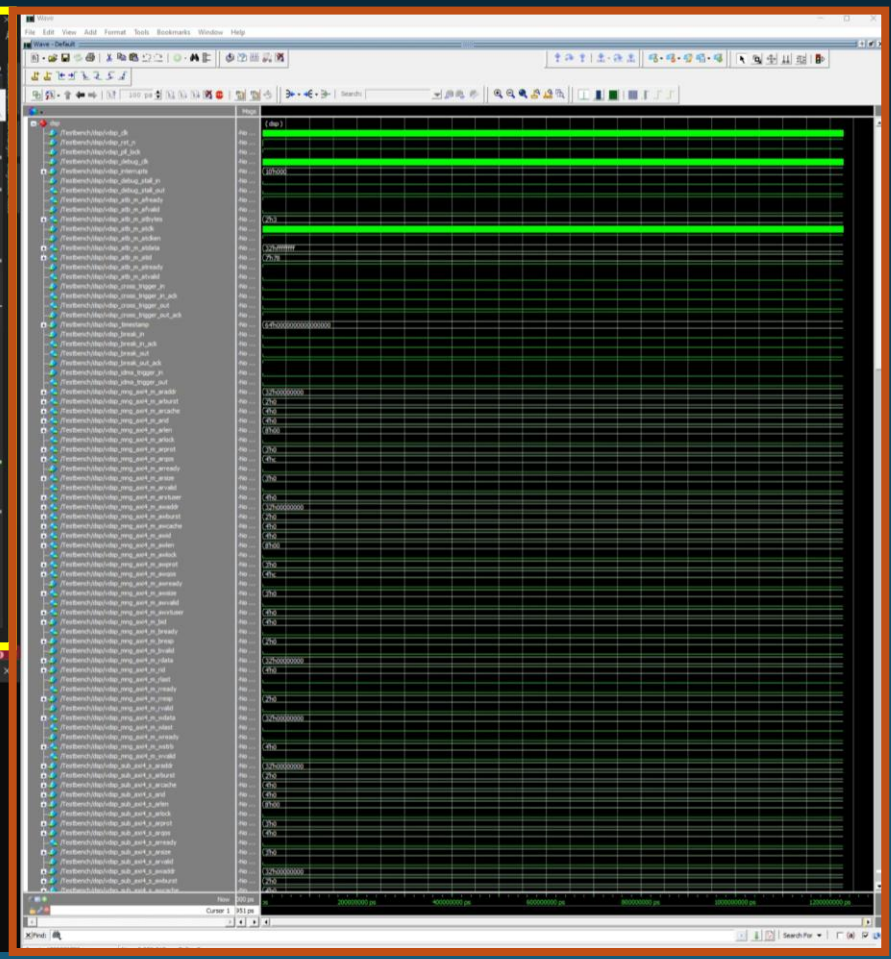


Smart Camera ICP – RTL Simulation



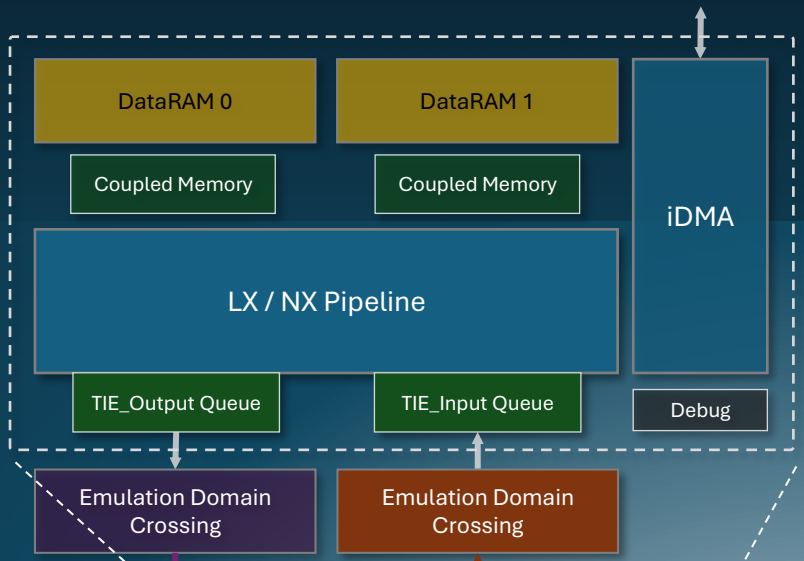


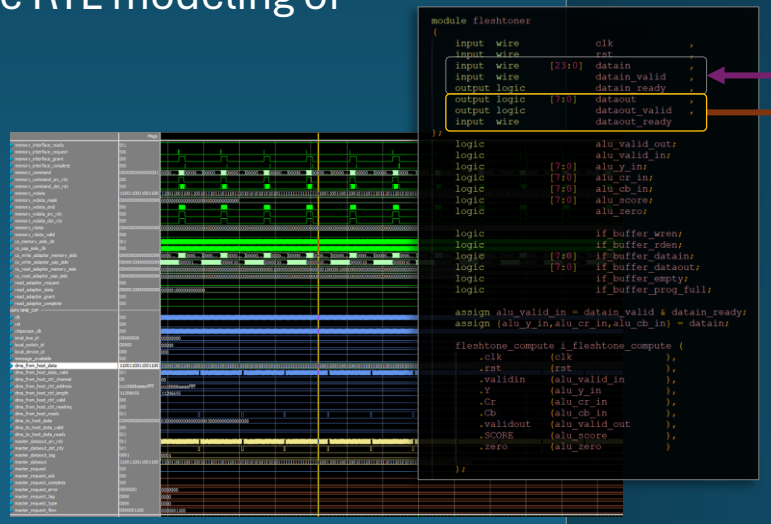


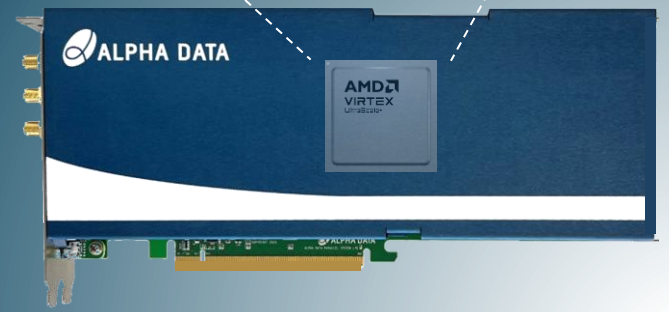


Hardware In the Loop + RTL Simulation

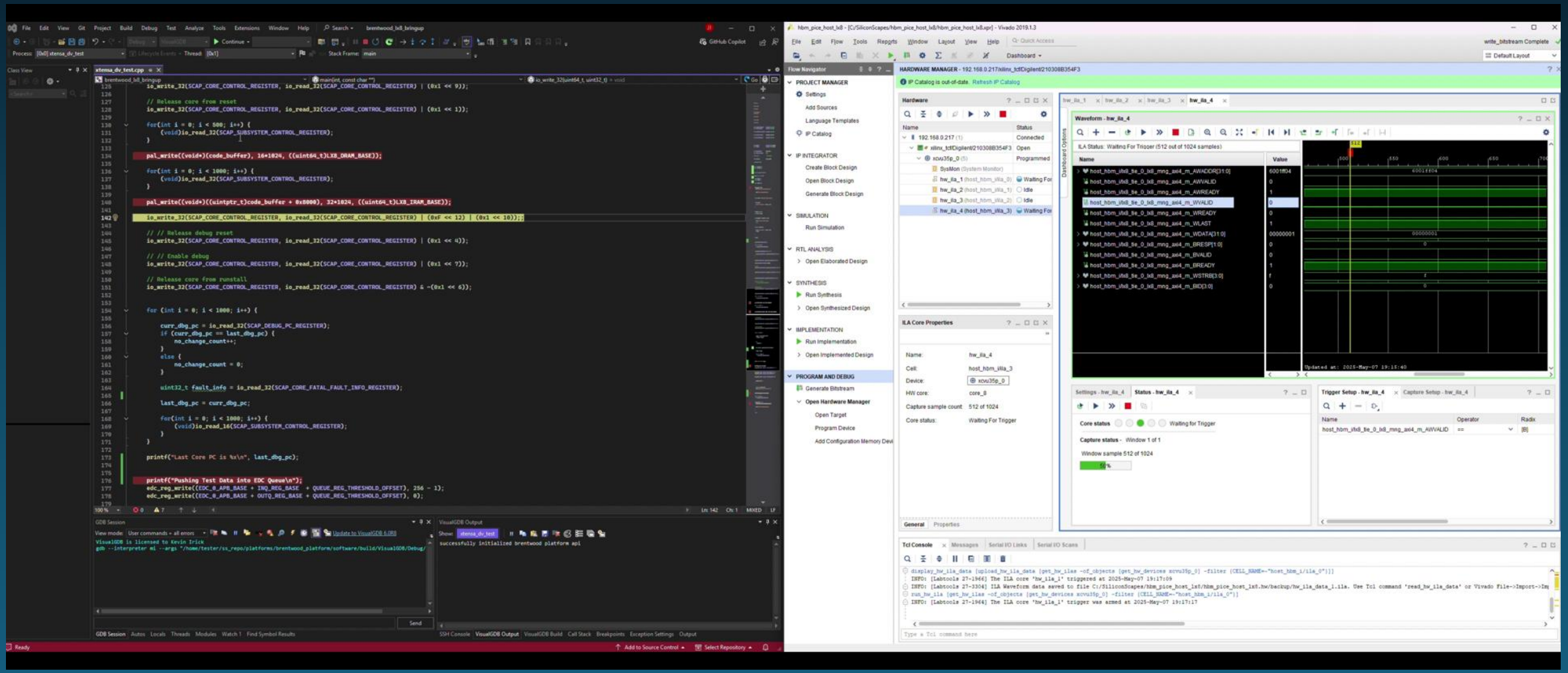
- Synthesized Tensilica core mapped to FPGA
- Emulation Domain Crossing (EDC) modules bridge TIE interfaces to CoreSYMU
  - Input EDC captures TIE Push operations and forwards data to CoreSYMU host domain
  - Processor is stalled as transactions are processed in cycle accurate RTL simulation
  - Results of cycle accurate execution are passed back to output EDC and presented to TIE Pop interface
- Offers “real-time” execution of DSP pipeline while allowing isolated cycle-accurate RTL modeling of accelerators







Smart Camera ICP – FPGA Prototyping



What we've learned along the way...

- ✓ ISA Extensibility is a powerful and distinguishing feature of Tensilica DSP
- ✓ Coupling DSP with custom hardware accelerators makes them even more powerful and application optimized
- ✓ There's an ideal interface for every accelerator architecture and use case. Choosing the right interfaces is crucial
- ✓ Use automation to integrate cores and leverage the rapid exploration that the Cadence XPG flow allows
- ✓ Use the right DSP modeling techniques for the various stages of accelerator development and integration
- ✓ Use FPGAs to prototype your DSP+HWA based system