AI ACCELERATOR ECOSYSTEM: AN OVERVIEW

DAVID BURNETTE, DIRECTOR OF ENGINEERING, MENTOR, A SIEMENS BUSINESS

Mentor®
A Siemens Business

WHITEPAPER

HIGH-LEVEL SYNTHESIS

www.mentor.com

One of the fastest growing areas of hardware and software design is Artificial Intelligence (AI)/Machine Learning (ML), fueled by the demand for more autonomous systems like self-driving vehicles and voice recognition for personal assistants. Many of these algorithms rely on convolutional neural networks (CNNs) to implement deep learning systems. While the concept of convolution is relatively straightforward, the application of CNNs to the ML domain has yielded dozens of different neural network approaches. While these networks can be executed in software on CPUs/GPUs, the power requirements for these solutions make them impractical for most inferencing applications, the majority of which involve portable, low-power devices. To improve the power/performance, hardware teams are forming to create ML hardware acceleration blocks. However, the process of taking any one of these compute-intensive networks into hardware, especially energy-efficient hardware, is a time consuming process if the team employs a traditional RTL design flow. Consider all of these interdependent activities using a traditional flow:

- Expressing the algorithm correctly in RTL.
- Choosing the optimal bit-widths for kernel weights and local storage to meet the memory budget.
- Designing the microarchitecture to have a low enough latency to be practical for the target application, while determining how the accelerator communicates across the system bus without killing the latency the team just fought for.
- Verifying the algorithm early on and throughout the implementation process, especially in the context of the entire system.
- Optimizing for power for mobile devices.
- Getting the product to market on time.

This domain is in desperate need of a productivity-boosting methodology shift away from an RTL flow.

ENTER CATAPULT HLS PLATFORM

Fifteen years ago, Mentor recognized that design and verification teams need to move up from RTL to the HLS level and developed the Catapult® HLS Platform. Companies worldwide trust the Catapult HLS Platform for designing and verifying ML accelerators and connecting them to systems. The platform provides a complete flow from C++ to optimized RTL (Figure 1).

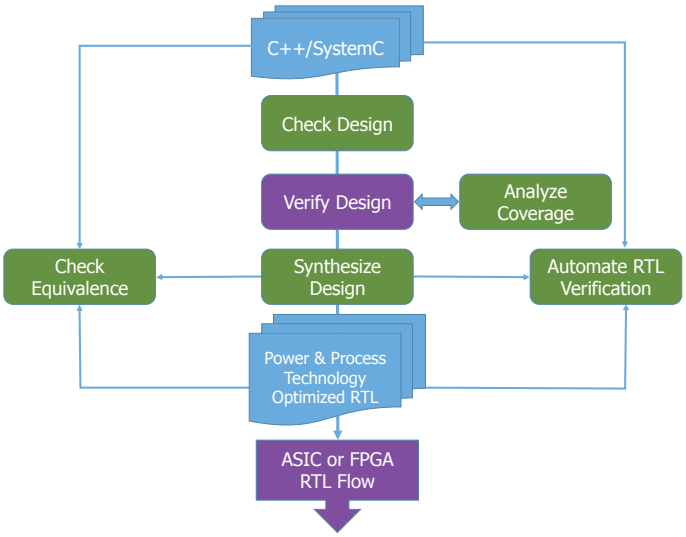

Figure 1: The Catapult HLS Platform

The Catapult HLS Platform provides a hardware design solution for algorithm designers that generates high-quality RTL from C++ and/or SystemC descriptions that target ASIC, FPGA, or eFPGA implementations. The platform delivers the ability to check the design for errors before synthesis, provides a seamless and reusable testing environment for functional verification and coverage analysis, and supports formal equivalence checking between the generated RTL and the original HLS source. This flow ensures fast design and verification and delivers power and technology optimized RTL ready for simulation and RTL synthesis.

- By employing these elements of the Catapult HLS Platform solution, teams take their products to market faster and at a lower cost because the solution:
- Enables late-stage changes. Easily change C++ algorithms at any time and regenerate RTL code or target a new technology.
- Supports hardware evaluation. Rapidly explore options for power, performance, and area without changing source code.
- Accelerates schedules. Reduce design and verification time from one year to a few months and add new features in days not weeks, all using C/C++ code that contains 5x fewer lines of code than RTL.

INTRODUCING THE AI ACCELERATOR ECOSYSTEM

Catapult HLS Platform provides a powerful tool flow for IC designers. But, Mentor has taken a big step farther and offers an AI accelerator ecosystem (Figure 2) that provides AI designers with an environment to jumpstart projects. Based on years of working with designers, Mentor deploys this ecosystem within the Catapult HLS Platform and enhances it with every release. In addition, Mentor offers many elements of this ecosystem free to any designer.

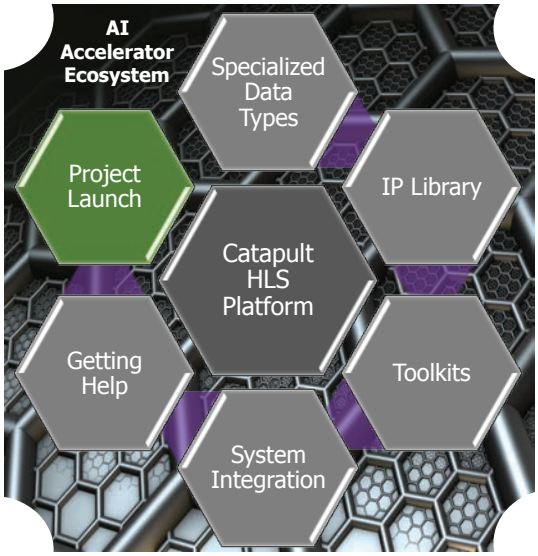

Figure 2: Catapult AI Accelerator Ecosystem

HLS LIBS SITE

Created by Mentor and facilitated by the Catapult team, HLS LIBS is a free and open-source set of libraries implemented in standard C++ for bit-accurate hardware design. HLS LIBS (Figure 3) is an open community for exchanging knowledge and IP that can be used to accelerate both research and design. The libraries are targeted to enable a faster path to hardware acceleration by providing easy-to-understand, high-quality, fundamental building blocks that can be synthesized into either FPGA, eFPGA, or ASIC technologies. Each library is fully documented and available as an open-source project on GitHub.

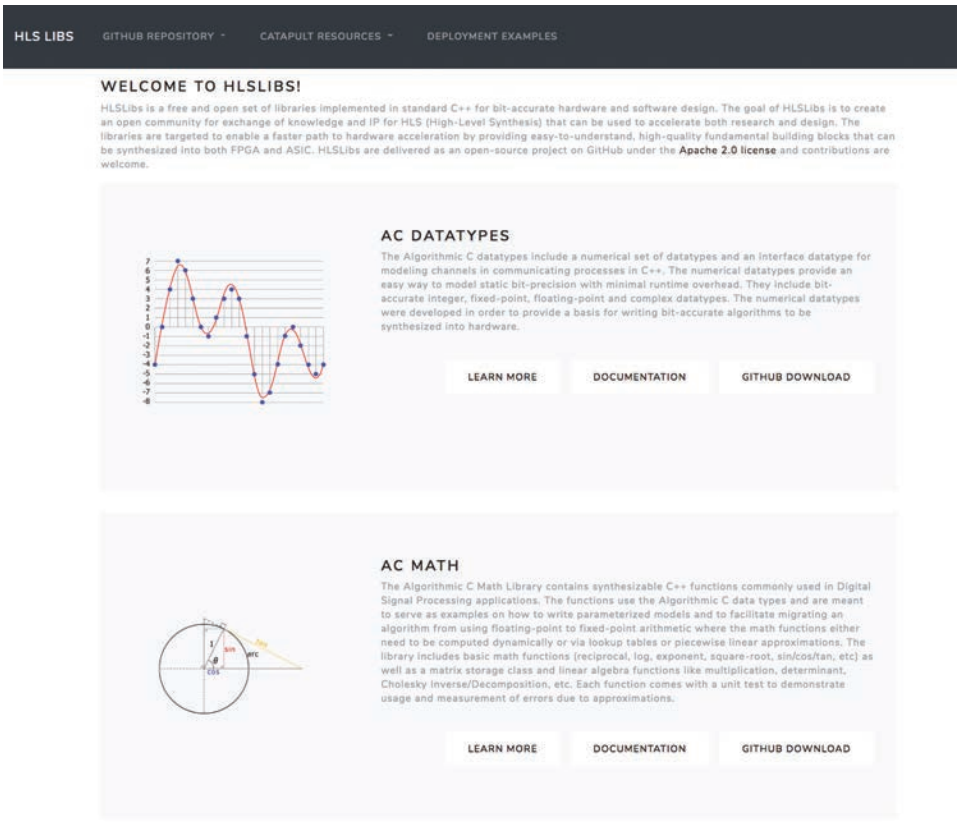

Figure 3: HLSLIBS provides free access to HLS IP

HLS LIBS provides a set of Algorithmic C (AC) libraries including specialized datatypes, basic math and linear algebra functions, digital signal processing, and image processing blocks.

AC DATATYPES

Catapult HLS provides bit-accurate, Algorithmic C datatypes (AC Datatypes) that are C++ classes that implement the datatype and provide overloaded operators for those datatypes. Using these datatypes for arbitrary length integer, fixed-point, floating-point, and complex numbers allows the code to synthesize into hardware with the same bit-for-bit behavior as the C++ model. These datatypes provide fast simulation, have well-defined simulation and synthesis semantics, and provide the foundation types for the IP libraries.

AC MATH LIBRARY

The Algorithmic C Math (AC Math) library defines synthesizable C++ functions for math operators typically found in the standard C++ math.h header as well as a C++ matrix class and linear algebra functions. All of the functions in AC Math are written using C++ template parameters that allow the designer to specify the numerical precision based on the target application. Many functions are implemented using different approximation strategies. For example, natural logarithm is provided in two forms – a piecewise-linear approximation and a cordic form. The former is smaller and faster when a small error in accuracy is acceptable, while the latter is slower but much more accurate. In all cases, the source can be customized to meet design goals. Each function/block comes with detailed design documentation and a C++ testbench that demonstrates the coverage/precision over selected ranges of bit-widths. Since the Catapult HLS Platform leverages C++ testbenches, it is easy to verify the RTL accuracy against the source design.

The categories of math functions in this library include:

- Piecewise linear functions - Absolute Value, Normalization, Reciprocal, Logarithm and Exponent (natural and base 2), Square Root, Inverse Square Root, and Sine/Cosine/Tangent (normal and inverse)
- Activation functions like Hyperbolic Tangent, Sigmoid, and Leaky ReLU
- Linear Algebra functions like matrix multiplication and Cholesky decomposition

Many of the functions also include a MATLAB® reference model useful for numerical analysis.

DSP LIBRARY

The Algorithmic C DSP (AC DSP) library defines synthesizable C++ functions that DSP designers typically require, such as filters and Fast Fourier Transforms (FFT). These functions employ C++ class-based design so that designers can easily instantiate multiple variations of objects to build up a complex DSP subsystem.

Just like the AC Math Library, the input and output arguments are parameterized so that arithmetic can be performed at the desired fixed-point precision to provide a high degree of flexibility when performing area and performance traded-offs for the synthesized hardware.

The DSP library includes:

- Filter functions like FIR, 1-D moving average, and poly-phase decimation
- Fast Fourier Transform functions like radix-22 single delay feedback, radix-2x dynamic in-place, and radix-2 in-place

IMAGE PROCESSING LIBRARY

The Algorithmic C Image Processing Library (AC IPL) starts with a definition of some common pixel format type definitions. These types utilize C++ template parameters to configure the color depth and format (RGB, YUV). The library then provides various function blocks useful for image processing including color conversion, BiCubic image scaling, boundary processing, and windowing classes for use in 2-D convolution.

TOOLKITS

The AI accelerator ecosystem provides toolkits that are real-world, tested examples of accelerator-based reference designs that teams can study, modify, and copy to jumpstart projects. These kits, shipped with Catapult, include configurable C++/SystemC IP source code, documentation, testbenches, and scripts to move the design through the HLS synthesis and verification flow. The toolkits demonstrate various approaches and coding techniques for experimenting with tradeoffs for performance (latency), frame rates, area, or power.

PIXEL-PIPE VIDEO PROCESSING TOOLKIT

The video processing toolkit demonstrates a live image processing application using a pixel-pipe accelerator (Figure 4). The accelerator block is implemented using C++ class hierarchy. The block down-scales the image, converts it from color to monochrome in order to perform edge detection, and then up-scales the image. A user-space application running on a CPU under Xilinx® PetaLinux allows software control to enable or bypass the edge detection block. The toolkit documentation shows how to integrate this block into a Xilinx board using Xilinx IP so that the team can demonstrate the system.

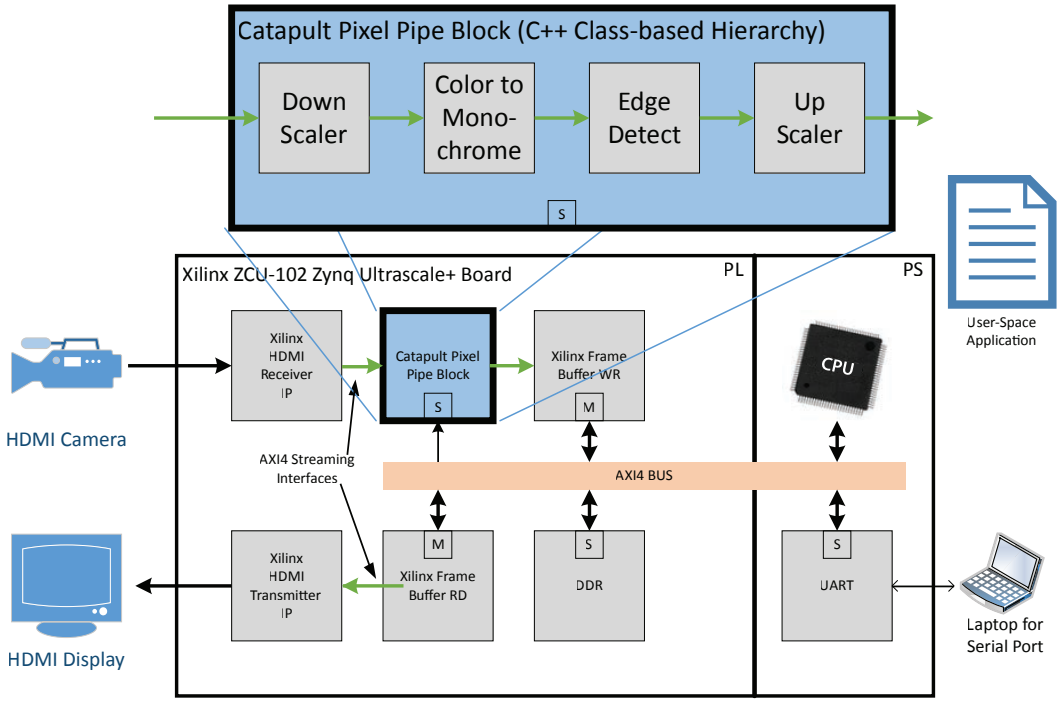

Figure 4: Pixel-pipe Video Processing Toolkit

2-D CONVOLUTION TOOLKIT

This toolkit demonstrates how to code an Eyeriss[1] processing element (PE) array in C++ that implements a 2-D convolution to perform image enhancement (sharpen, blur, and edge-detect). The processing element (Figure 5) can perform a 3x1 multiply accumulate (convolution).

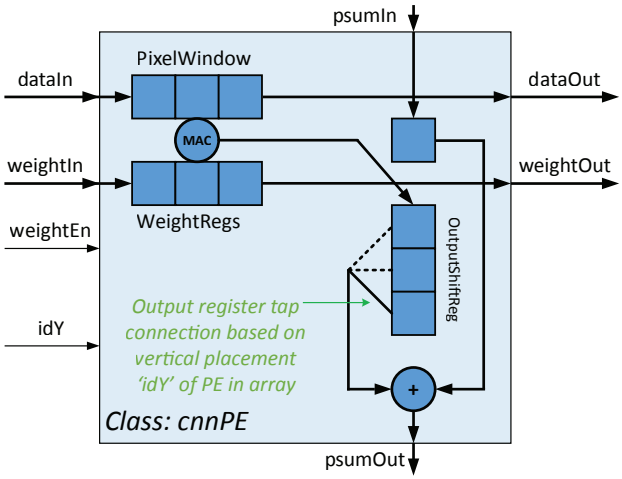

Figure 5: Eyeriss processing element

Stacking the processing elements vertically (Figure 6) yields a 3x3 convolution where the kernel weights determine the image enhancement performed.

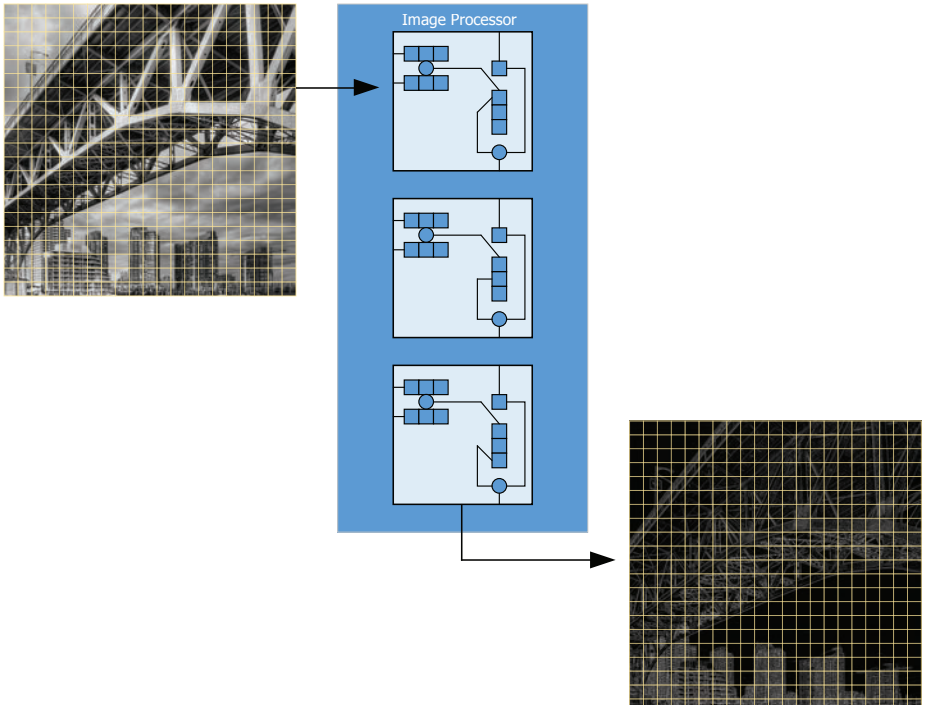

Figure 6: PE Array performing 2-D edge detect

TINYYOLO OBJECT CLASSIFICATION TOOLKIT

The object classification toolkit (Figure 7) demonstrates an object classification application using a convolution accelerator engine implemented with the PE array from the 2-D Eyeriss toolkit. It is based on the tinyYOLO ("You Only Look Once") neural network architecture and it includes a video preprocessing block to scale the image before executing object classification. The toolkit shows how to obtain high-speed data routing through AXI4 interconnect (reading kernel weight data from system memory) and demonstrates how to define a high-performance memory architecture. The toolkit provides an integration to TensorFlow for inference testing of the network layers implemented in C++. The documentation shows how to integrate this block into a Xilinx board using Xilinx IP so that the team can demonstrate the system.

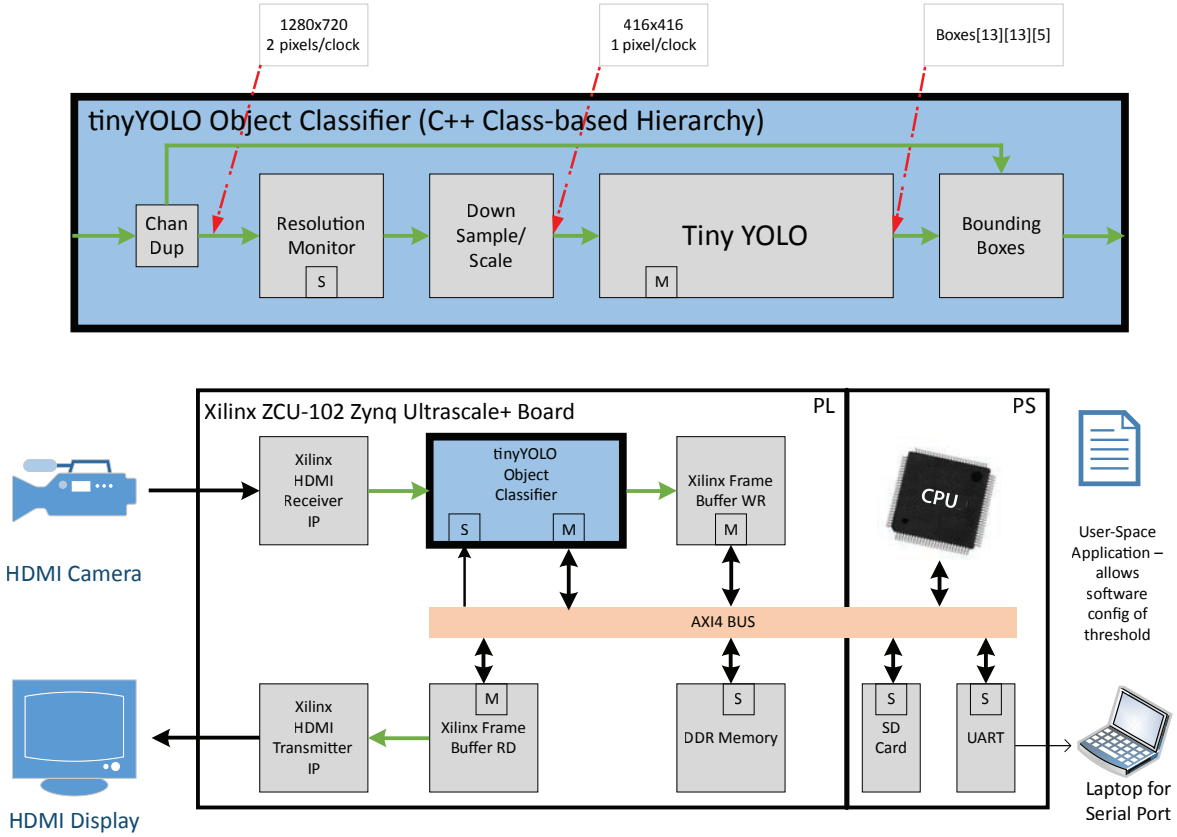

Figure 7: tinyYOLO Toolkit example – a system view

Figure 8 shows a detailed view of the interconnected PE array.

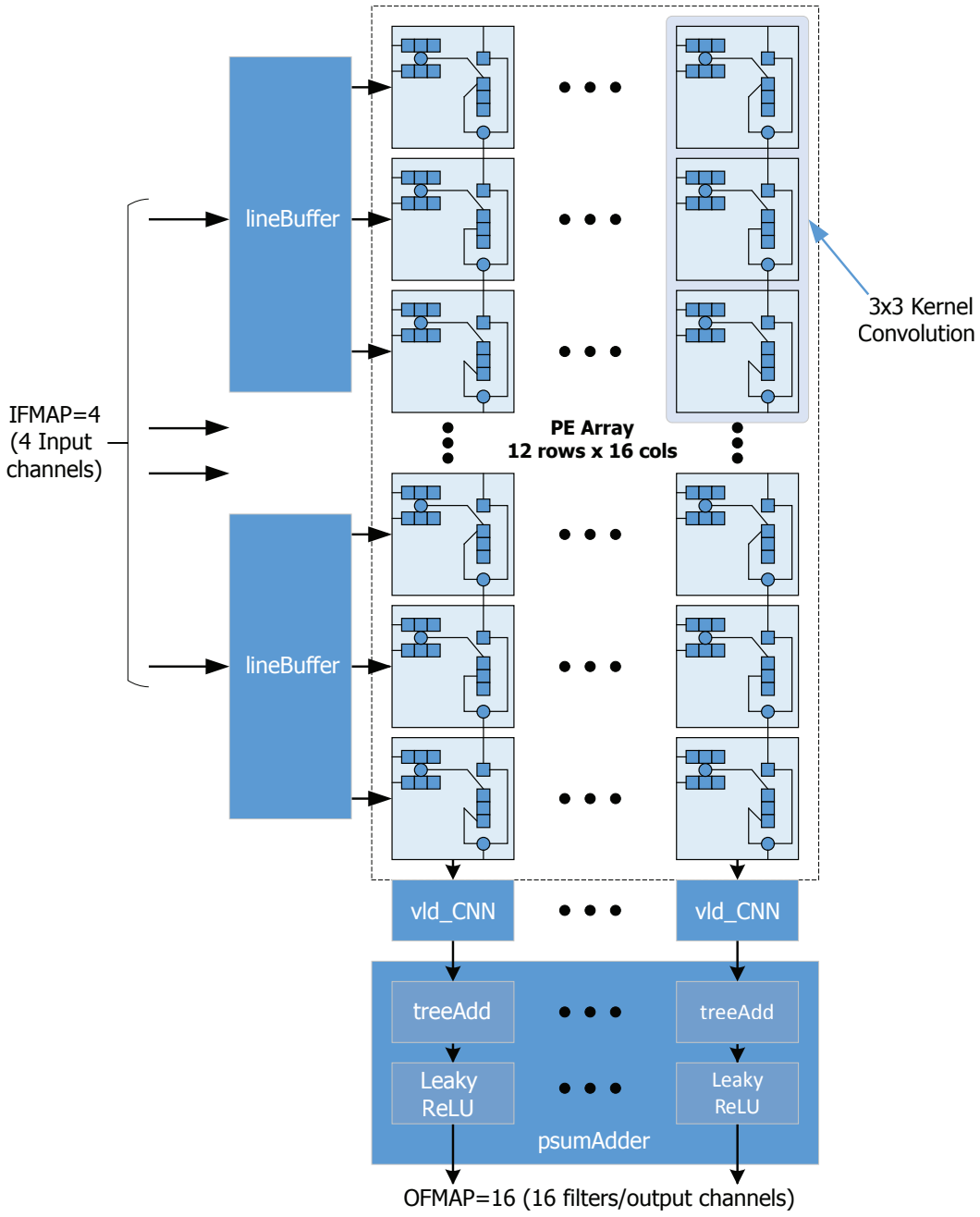

Figure 8: Expanded detail view of PE array and interconnect

SYSTEM INTEGRATION

An accelerator block does not live in isolation; it needs to be connected to a system. Catapult HLS offers Interface Synthesis to add a timed protocol to untimed C++ function interface variables. Designers simply need to set architectural constraints for the protocol in the Catapult GUI (Figure 9). The tool supports typical protocols such as AXI4 video stream, request/acknowledge handshaking, and memory interfaces. This allows designers to explore interface protocols without changing the C++ source.

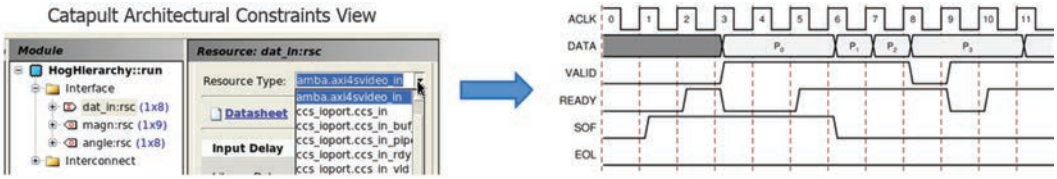

Figure 9: Selecting interface protocols

To support system integration, the AI accelerator ecosystem provides a set of fully-functional design examples.

AXI EXAMPLES

The AXI examples (Figure 10) show how to instantiate one or more accelerator components within an AXI SoC subsystem using the AXI interface IP that Catapult HLS generates. Master, slave, and streaming examples are available.

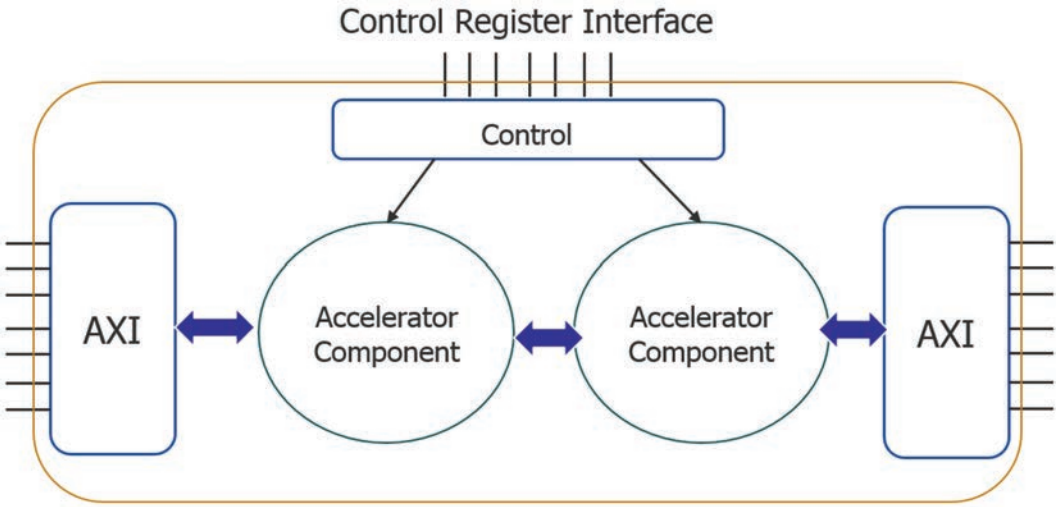

Figure 10: AXI examples

BASE PROCESSOR EXAMPLE

The base processor example (Figure 11) shows how to connect a ML accelerator into a complete processor-based system and it employs the AXI examples. The ML accelerator in this example employs a simple multiply/accumulate architecture with 2-D convolution and max pooling. Several 3rd-party processor IP models are supported and a software flow (with associated data) is included for bare metal programing.

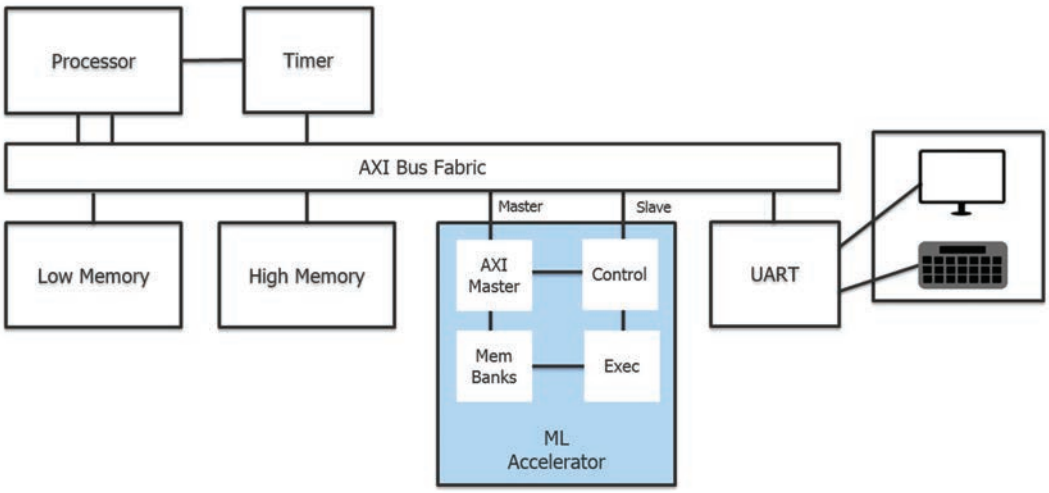

Figure 11: The Base Processor Platform example

GETTING HELP

Design teams might find that they need help with their projects and the ecosystem provides assistance in many ways:

- Online help and documentation: allows designers to instantly view information about using the tools.
- Self-paced tutorials: provide step-by-step lab workbooks and real design data to learn how to use the tools.
- On-demand training: delivers a structured, self-paced, online training experience to learn how to efficiently use the tools.
- Seminars: are offered live around the world and then captured on video for access online anytime. These seminars are updated regularly as the Catapult team works with designers to capture the very latest ML techniques.
- Onsite visits: by the Catapult application engineers and technical marketing engineers get teams up and running fast on the tool flow in the production environment.
- Consulting: provides onsite help for getting the tool flow integrated into the team's environment as well as help for designing C++ algorithms for optimal synthesis.
- Customer support: provides live access to award-wining engineers for help and the ability to tap into a wealth of online problem solving technology notes.

PROJECT LAUNCH

Combining the AI accelerator ecosystem with the Catapult HLS Platform design and verification solution allows teams to jumpstart and implement their AI projects quickly. The ecosystem grows with every release of the tools and as the elements of the ecosystem are proven with designers around the world, Mentor donates them to the HLS LIBS site. The ecosystem's wide range of offerings inspires a team at any level of experience to successfully launch their next AI IC project.

To learn more about the Catapult HLS Platform, visit its website.

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

Silicon Valley
Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
11F, No. 120, Section 2,
Gongdao 5th Road
HsinChu City 300,
Taiwan, ROC
Phone: 886.3.513.1000
Fax: 886.3.573.4734

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Trust Tower
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004

MGC 05-19 TECH18230-w