2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Fast, Flexible, Timing-accurate and Open-Source Performance Modeling Method for Compute Accelerators

Vishal Chovatiya, Infineon Technologies, Bangalore, India

Andrew Stevens, Infineon Technologies, Munich, Germany

Snehith Shenoy, Infineon Technologies, Bangalore, India

Infineon

accellera SYSTEMS INITIATIVE

Motivation – why?

- **Complex Accelerator Modelling**
  - Traditional discrete-event and TLM methods need detailed micro-architectural modeling

    High modelling complexity, **cost**, and **limited flexibility** for design space exploration

- **Existing A**rchitecture **D**escription **L**anguage(**ADL**) **Limitations**
  - Behavioral ADLs lack concurrency & dependency modeling

    Mixed ADLs **tightly couple functional and timing aspects**

  - Examples: nML, ISDL, LISA, EXPRESSION, MADL, Sparta

Motivation – how?

- **Flexible Framework**
  - Reduce dependency on detailed hardware models for faster exploration

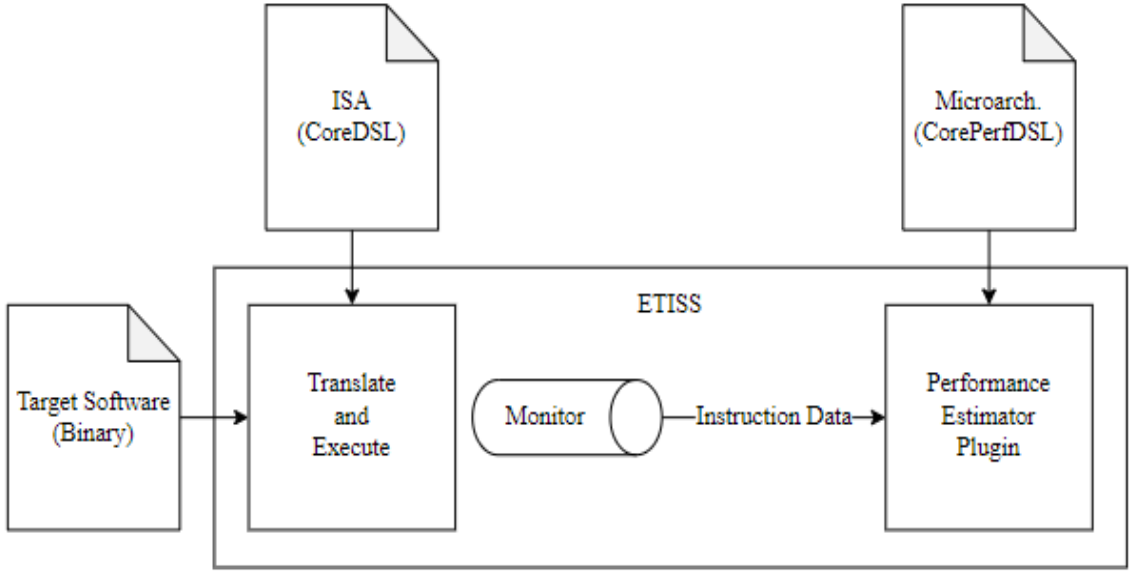
    Separate functional and timing concerns

    Abstract timing to balance accuracy and simulation speed

- **Enable Faster & Accurate Simulation**
  - Extending **CorePerfDSL**'s novel approach to accelerators
  - Faster and accurate simulation and architectural exploration for accelerators

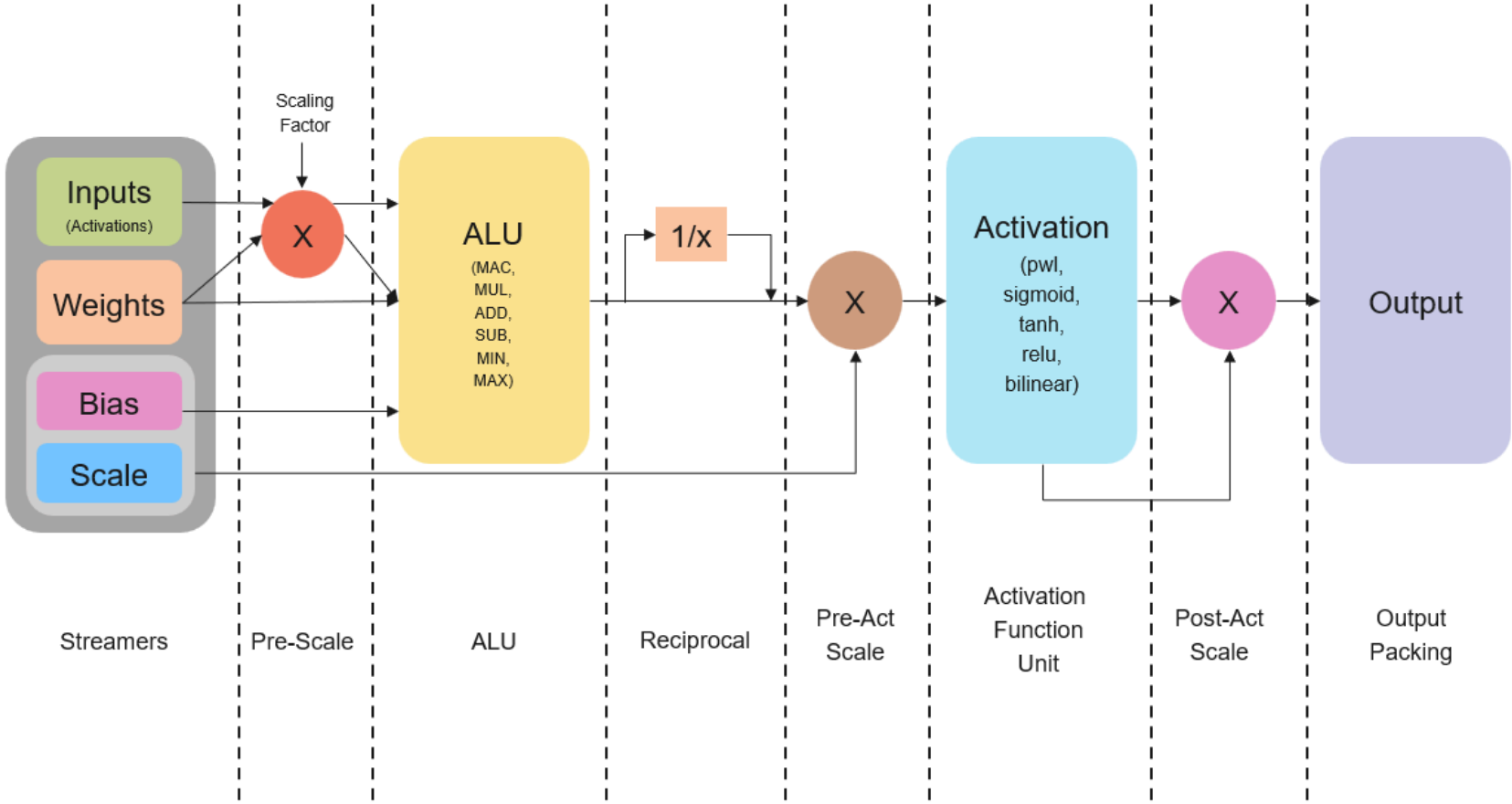
CorePerfDSL Method

- Dedicated language for modeling CPU pipeline timing
- **Opportunity to extend to accelerators like NPUs** for similar benefits

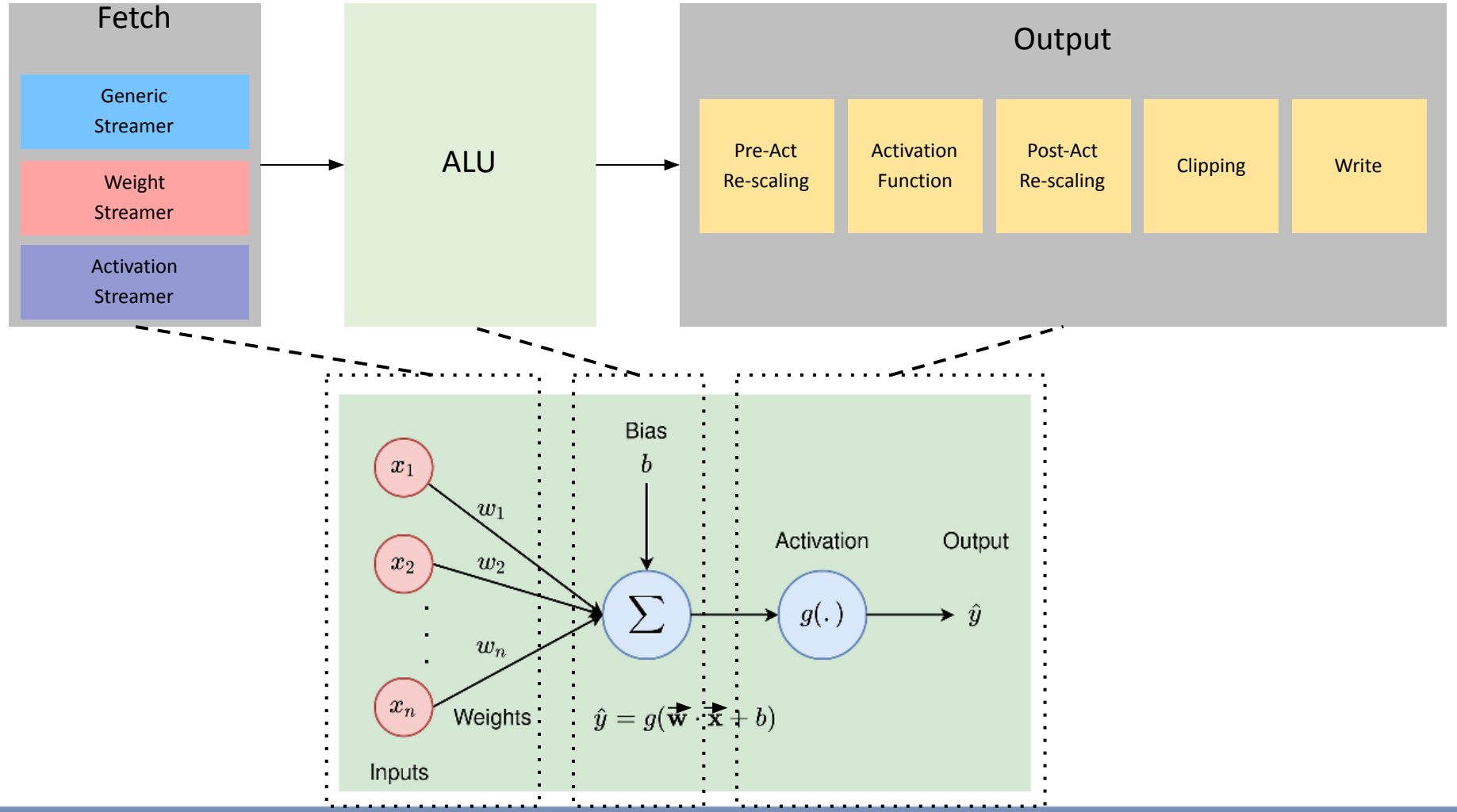


"CorePerfDSL: A Flexible Processor Description Language for Software Performance Simulation"
Conrad Foik, et al., FDL, 2022.

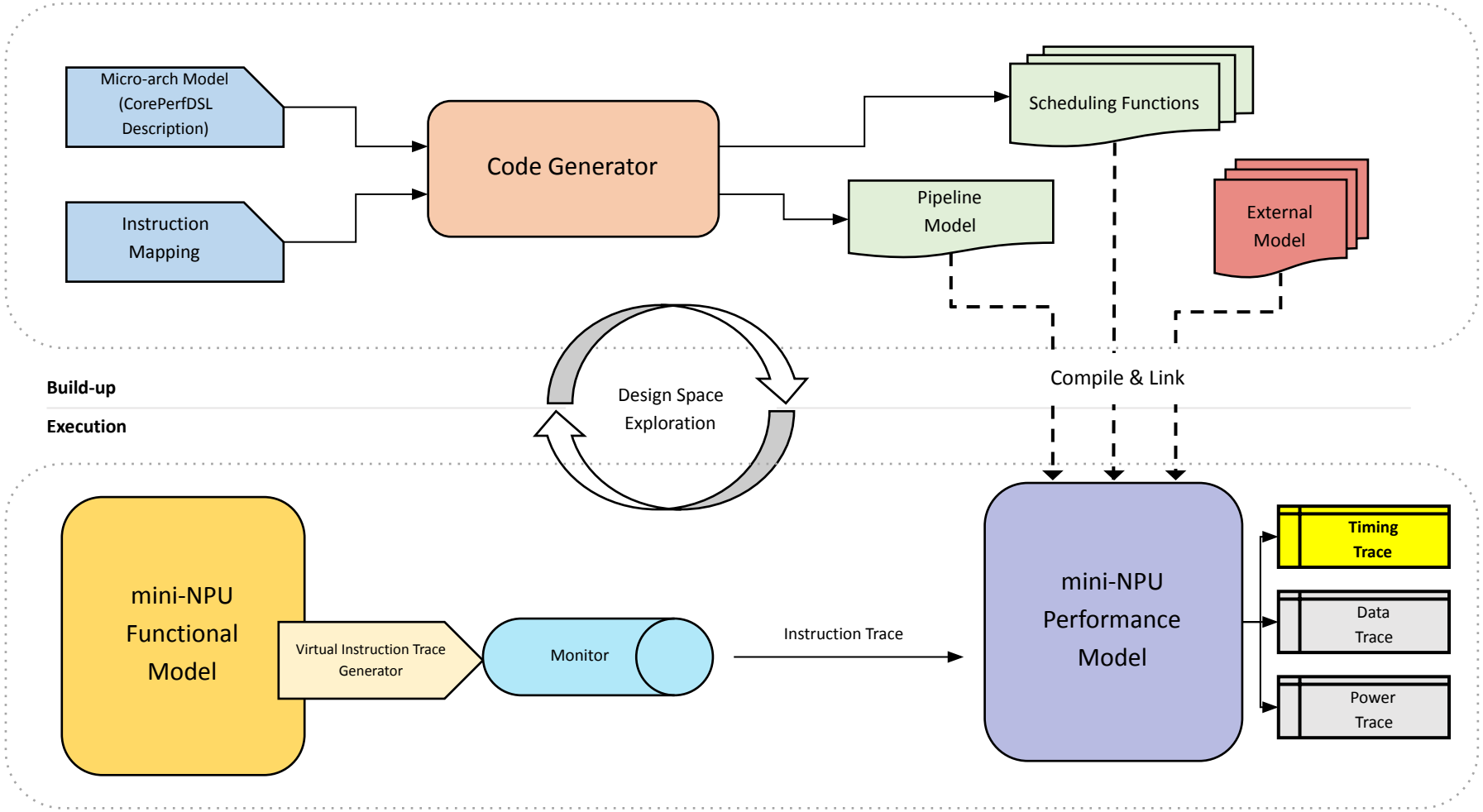
mini-NPU Physical Pipeline



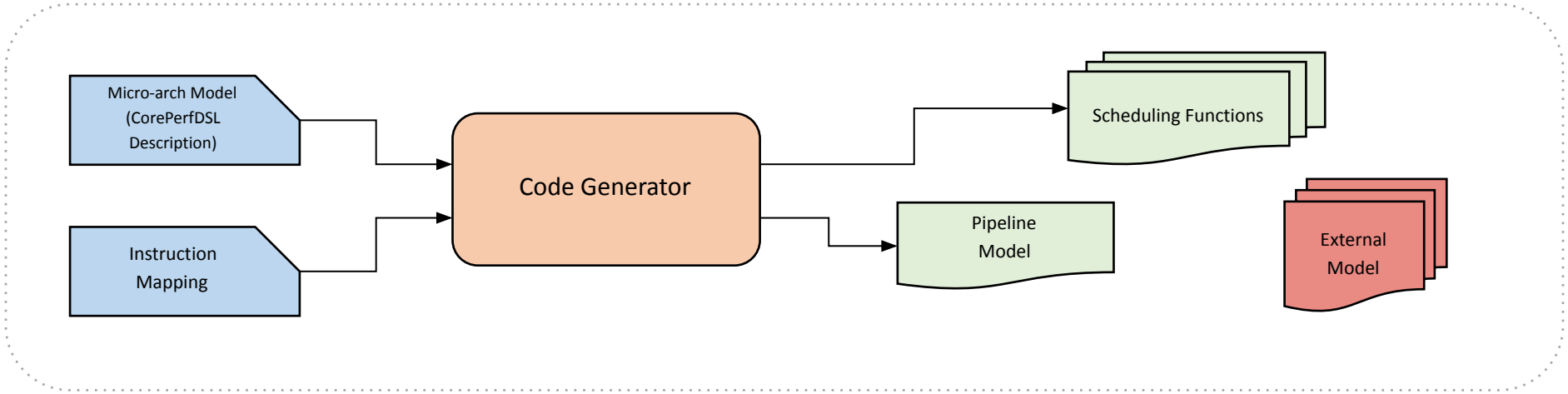
mini-NPU (Virtual) Instruction Pipeline



Performance Modelling Method (PMM)

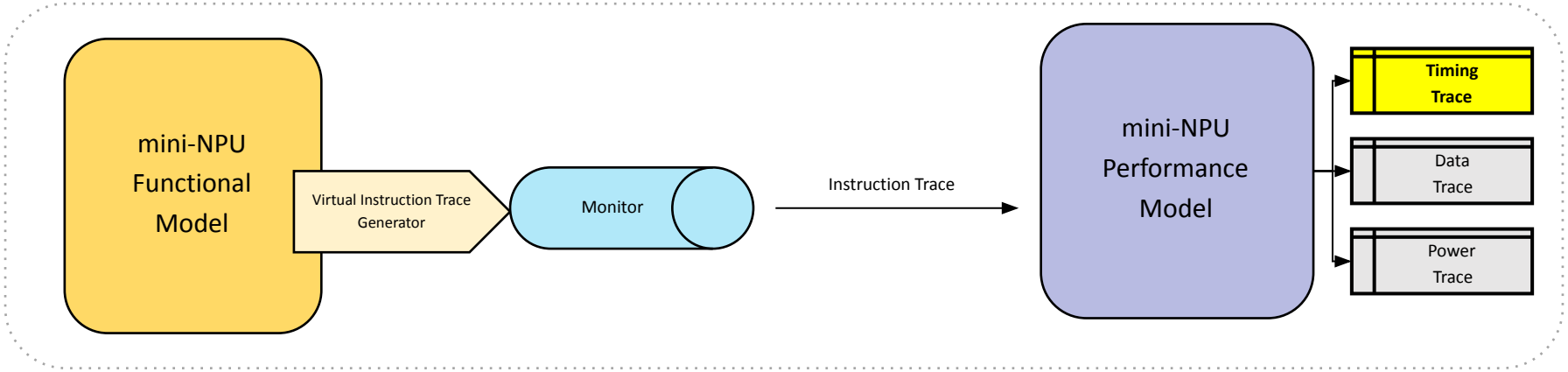


PMM: Build-up Phase



- **Micro-architectural Model**
  - Depicts architecture containing pipeline, stages and microaction(along with dependency) mapped to stages
- **Instruction Mapping**
  - Maps stages(and thereby respective microactions) to be activated as instruction passes through pipeline
- **Code Generator** that generates -
  - Scheduling Functions
    - Timing function that calculates and update the pipeline stages
  - Pipeline Models
    - Structural hierarchy that keeps track instruction timing
- **External Models**
  - Captures dynamic latency not directly represented in the CorePerfDSL

PMM: Execution Phase



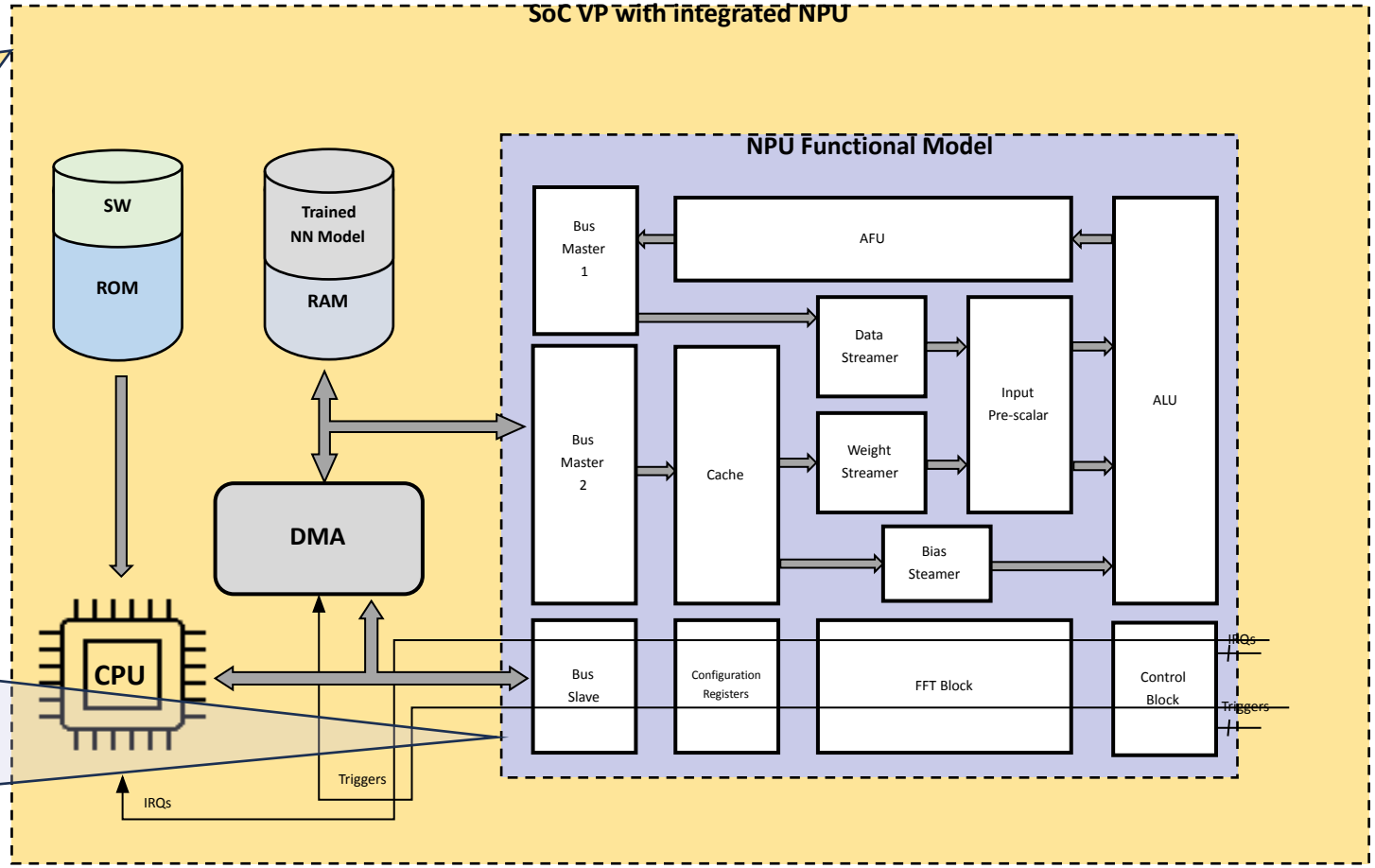
- **mini-NPU `Functional` Model**
  - Functionally accurate virtual model of hardware.
  - Includes a Virtual Instruction Trace Generator that creates a sequence of instructions that would be executed by the actual hardware when running target neural network workloads.
- **Monitor**
  - A pluggable hook to the functional model that captures and supplies instruction traces to the performance model.
- **mini-NPU `Performance` Model**
  - CorePerfDSL generated model that accepts instruction traces as input and generates comprehensive timing traces.

Target hardware: mini-NPU

- CPU, DMA, Memories
- RAM for trained model
- ROM for eSW
- **NPU accelerator functional model**

NPU functional model

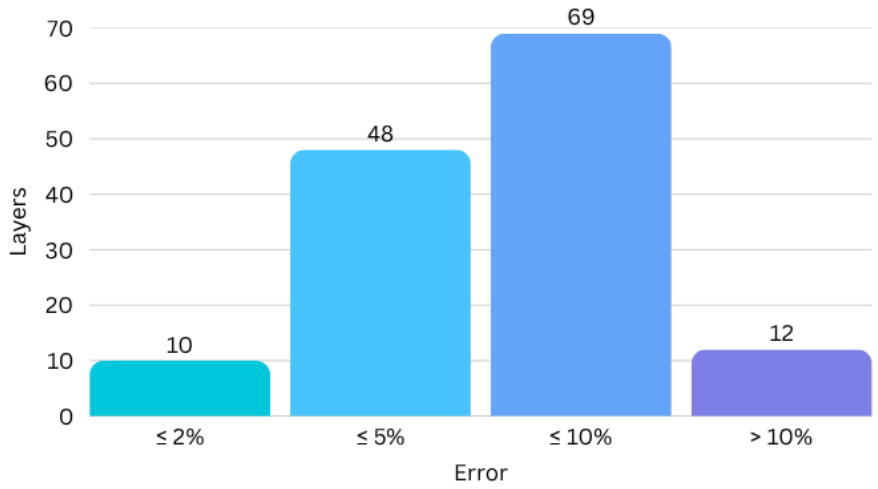
- Mimics HW without involving any timing detail or delay
- Used as peripheral in SoC to accelerate the AI/ML operations
- Helps in developing embedded AI toolchain early & efficiently



Results

Task	Dataset	Model	Mode	Quality	Latest Version Available
Keyword Spotting	Google Speech Commands	DS-CNN	Single-stream, Offline	90% (Top 1)	v1.1
Visual Wake Words	Visual Wake Words Dataset	MobileNetV1 0.25x	Single-stream	80% (Top 1)	v1.1
Image classification	CIFAR10	ResNet-8	Single-stream	85% (Top 1)	v1.1
Anomaly Detection	ToyADMOS	Deep AutoEncoder	Single-stream	0.85 (AUC)	v1.1

source: https://mlcommons.org/benchmarks/inference-



Operation Type	Layer Count	Mean Absolute Error (%)	Standard Deviation (%)	Max Error (%)
FullyConnected	13	2.44	2.25	8.74
Conv2D	28	6.67	4.93	20.68
DWConv2D	17	3.47	1.63	7.14
Add	3	14.23	0.03	14.26
AvgPool2D	3	5.52	5.42	13.17
Softmax Components	18	13.38	22.22	79.55

Benefits

- **HW ↔ SW co-design**
  - Enables early development, testing, and debugging of target SW
  - Hardware designers simultaneously validate & optimize their design based on software feedback

- **Architecture Exploration**
  - Code generator enables quick generation of different micro-architecture variants from CorePerfDSL description
  - Enables parameter based simulation sweeps

Benefits

- **Accurate Timing Model**
  - Separation of micro-architecture (through CorePerfDSL) model from functional model enables tuning timing detail independent of functionality

- **Faster Regressions**
  - Since functional and performance models are two different entities, performance model can be used adjunct to functional model
  - During regression run we can plug out the performance model to increase the simulation speed

Future Work

- **Utilization Graph**
  - Resource utilization graph generation
  - Identify bottleneck or under-utilized components/modules in design
- **Dynamic Power Estimation**
  - Power estimation using UPF gives relative and dynamic power estimates b/w different variants
  - Application based dynamic power estimation e.g. ResNet vs VGGNet
- **Crypto Accelerator**
  - Trying out method for crypto accelerators e.g. AES/DES accelerator

Questions?