arm AI

Introducing a common operator set for Machine Learning accelerators – TOSA

Arm

Ronan Naughton, Eric Kunze

October 2022

arm AI

Welcome!

Tweet us: @ArmSoftwareDev -> #AIVTT

Check out our Arm Software Developers YouTube channel

Signup now for our next AI Virtual Tech Talk: www.arm.com/techtalks

Our upcoming Arm Tech Talks

Date	Title	Host
11th October	Introducing a common operator set for Machine Learning accelerators – TOSA	Arm
8th November	Energy-Efficient ML Vision application development with Ethos-U65 microNPU from NXP and Arcturus	NXP & Arcturus
22nd November	DavinSy Hands-on: Continuous learning beyond the edge	Bondzai
December 6th	Get started with AI at the Edge with a workshop from Arm	Arm

Visit: www.arm.com/techtalks

arm AI

Presenters

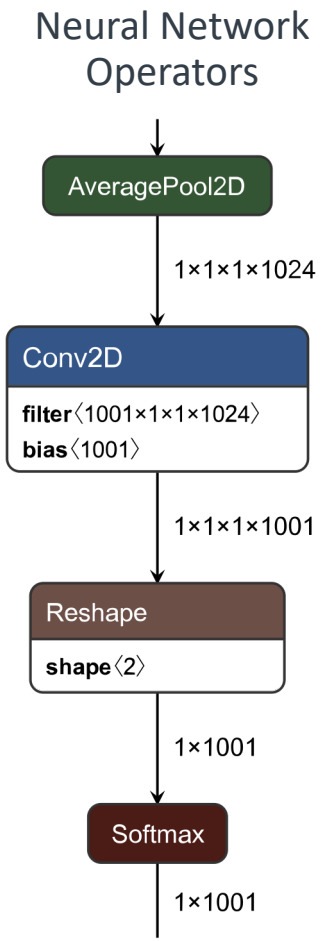
Ronan Naughton

Senior Product Manager

Eric Kunze

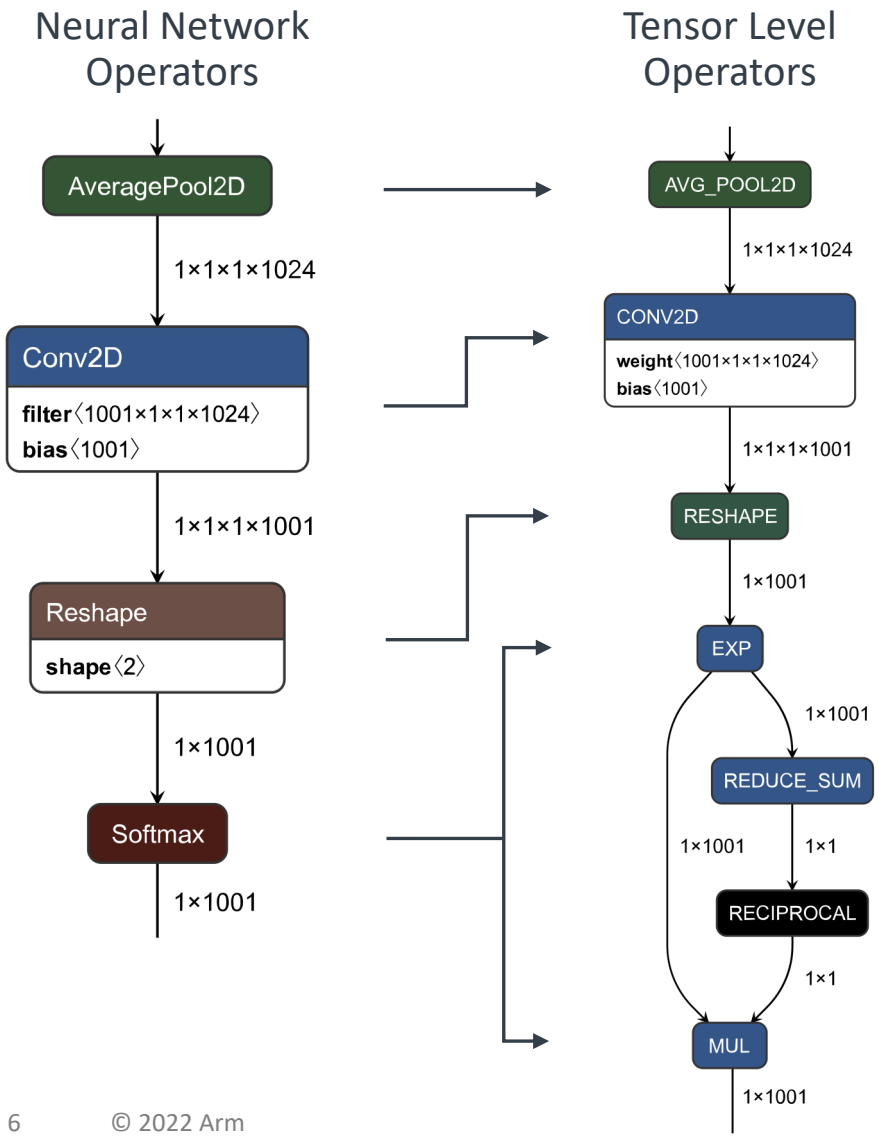
Distinguished Engineer

The Problem: Operator Proliferation in ML Frameworks



- There are many ML frameworks
  - TensorFlow, TensorFlow Lite, PyTorch, JAX
  - Each framework has many operators
  - TensorFlow Lite has >150, TensorFlow has >1000 etc.
- Operator proliferation and framework inconsistency is a persistent barrier to neural network deployment and portability

TOSA – What is it?



Tensor Operator Set Architecture (TOSA)

- A minimal and stable set of tensor-level operators to which most machine learning framework operators can be reduced
- Agnostic to any single high-level framework, compiler backend stack or particular target
- TOSA specification contains detailed functional and numerical descriptions which enables precise code construction for a diverse range of hardware – CPU, GPU & NPU

Benefits of TOSA

ML Standardization

- A stable ML specification to which Neural Network developers and SoC designers can adhere
- Complete code definition for each TOSA operator
- Controlled versioning of specification - currently v0.40
- Numerical precision is defined independent of framework behaviour

2.5.1.ADD

Elementwise addition of input1 and input2. Axis of size 1 will be broadcast, as necessary. Rank of input tensors must match.

Arguments:

Argument	Type	Name	Shape	Description
Input	in_t*	input1	shape1	Input tensor
Input	in_t*	input2	shape2	Input tensor with the same rank as input1
Output	in_t*	output	shape	Output tensor with broadcast shape if necessary

Operation Function:

```
for_each(index in shape) {
      index1 = apply_broadcast(shape, shape1, index);
      index2 = apply_broadcast(shape, shape2, index);
      in_t value1 = tensor_read<in_t>(input1, shape1, index1);
      in_t value2 = tensor_read<in_t>(input2, shape2, index2);
      in_t result = apply_add<in_t>(value1, value2);
      tensor_write<in_t>(output, shape, index, result);
```

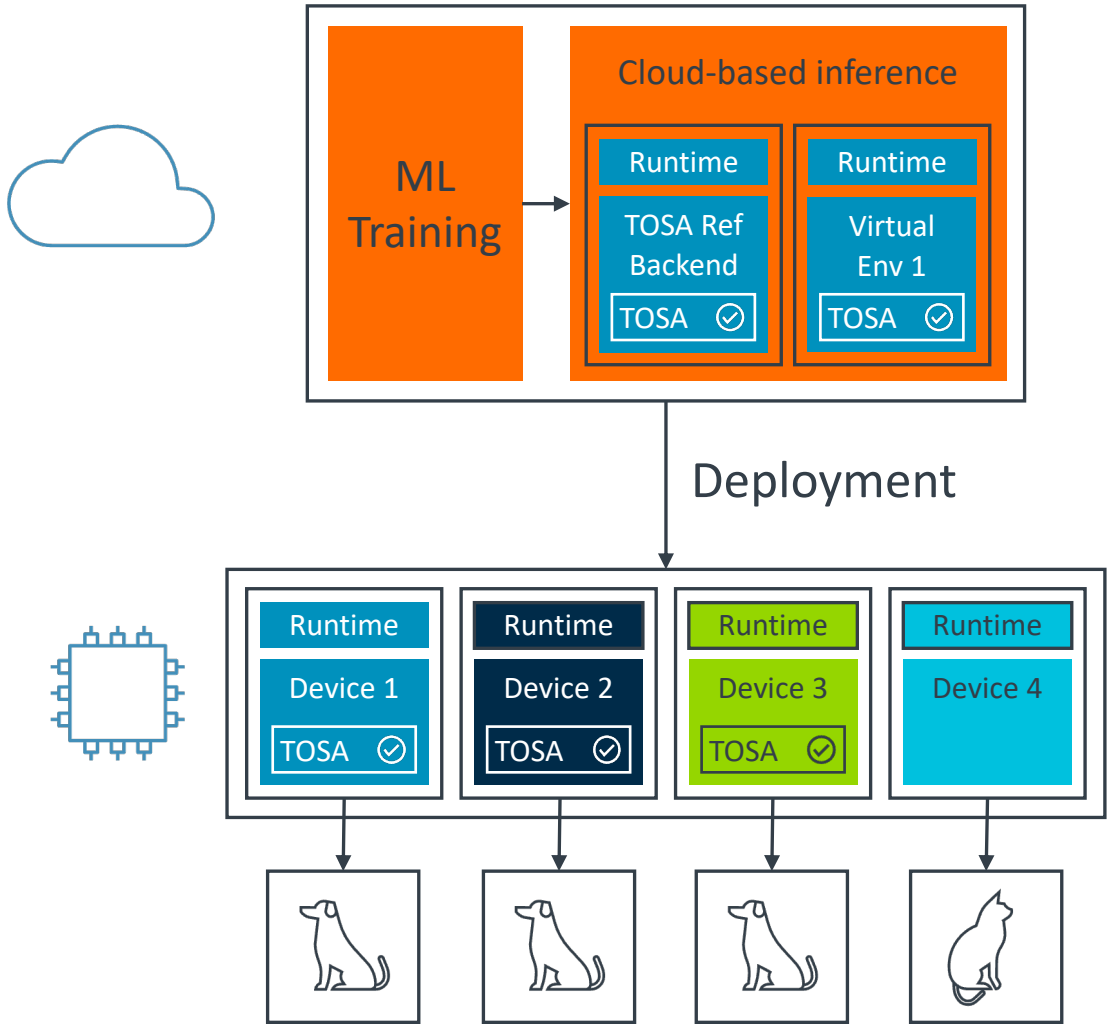
Supported Data Types:

Profile	Mode	in_t
Any	signed 32	int32_t
MI, MT	floating-point	float_t

Benefits of TOSA

Portability

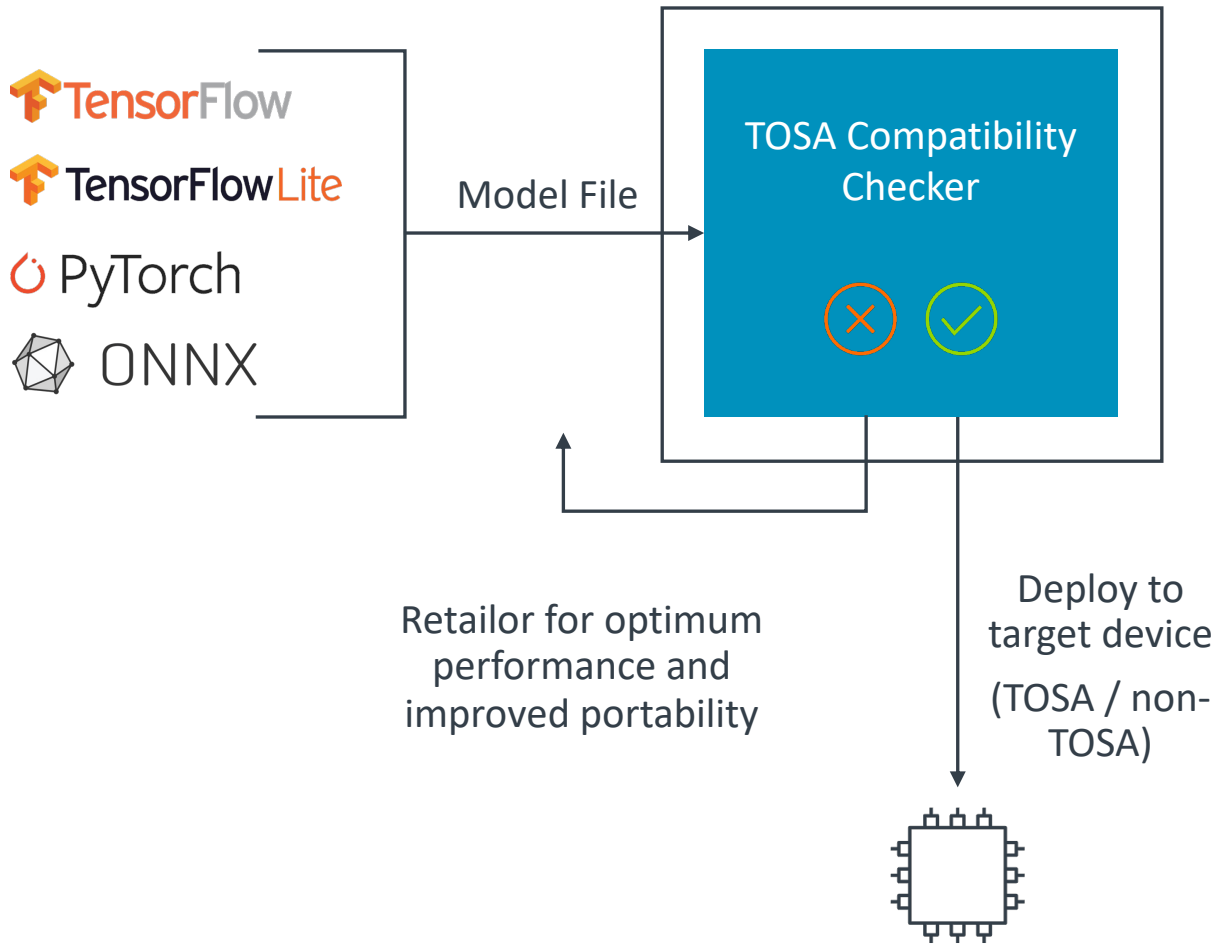
- TOSA compliant Neural Networks can run on any TOSA compliant HW whilst guaranteeing numerically consistent behaviour
- The specification defines precision for each operator
- Enables a dramatic reduction in development, test and certification time when deploying to multiple devices



Benefits of TOSA

Model Assurance

- Neural Networks can be assessed for TOSA compliance during development using TOSA Compatibility Checker
- Non TOSA compliant models can be re-tailored in advance of deployment to ensure optimum performance and portability



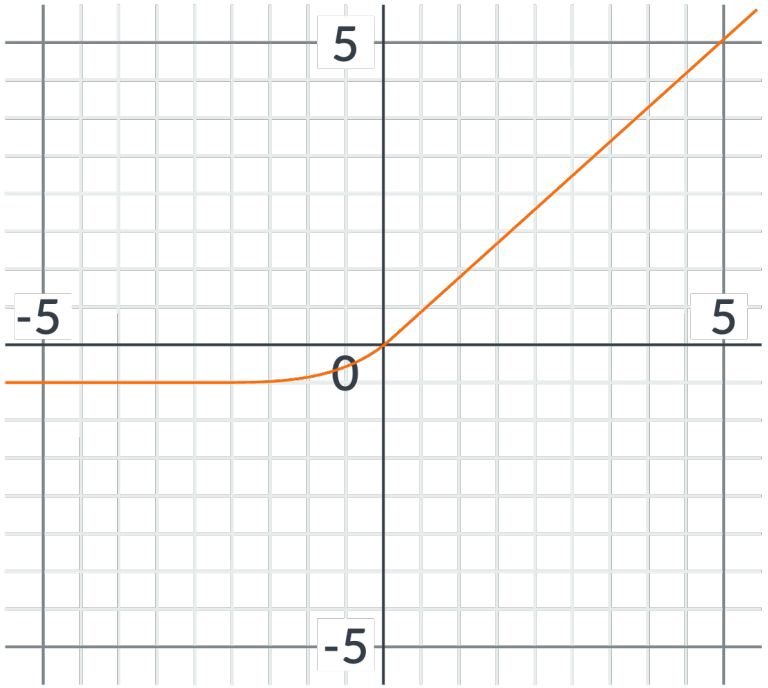
Benefits of TOSA

Future Proofing

- Future framework operators which can be reduced to TOSA operators can be supported in TOSA compliant hardware
  - → SoC Developers: Extending the lifetime of your hardware
  - → NN Developers: Exploit new operators immediately to already deployed devices

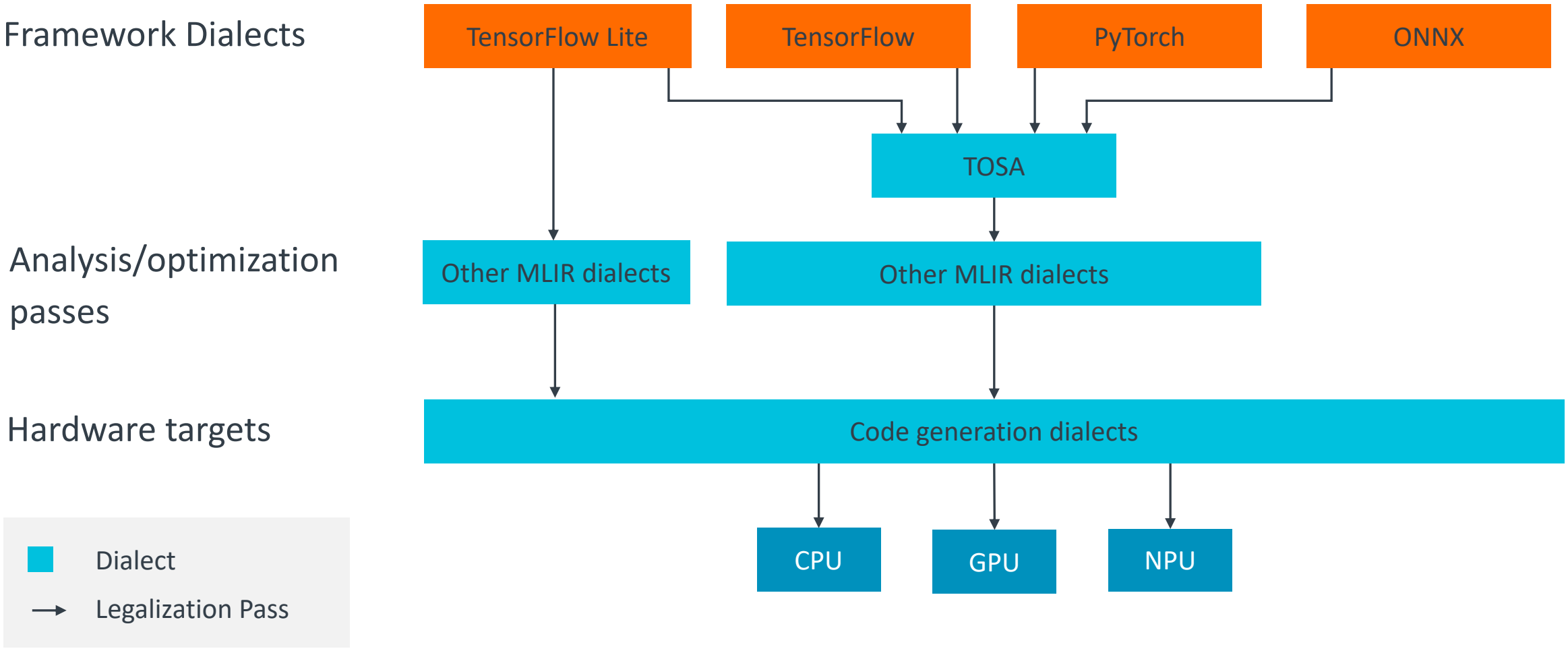
- Easy to construct a new operators using TOSA operators
  - E.g. ELU activation: elu(x) = if x>=0, exp(x)-1 otherwise
    - A = EXP(x)
    - B = SUB(A,1)
    - C = GREATER_EQUAL(X,0) // is X> 0
    - Output = SELECT(C, X, B) // return X or B based on >= results

ELU Activation



TOSA in MLIR

Example Compiler Flow (simplified)



Market Endorsements

Google

"We have found a canonical set of ops to be essential to taming the proliferation of framework ops and accelerators that need to be supported. TOSA provides a ***framework neutral, fully specified, industry standard*** *for frontends to target and backends to use as input for efficient codegen. We intend to target this for our compilation for mobile from TensorFlow and PyTorch".*

– Jacques Pienaar, IREE Compiler Team

AMD

"The rapidly evolving needs of Machine Learning compute has created many new opportunities for hardware acceleration. TOSA provides a ***small standard set of Machine Learning operators*** *capable of specifying a wide variety of ML models,* ***decoupling*** *hardware* ***details from the rapidly evolving needs of frontend frameworks****. In combination with compiler flows such as MLIR, TOSA serves as a valuable canonicalization point,* ***enabling a new generation of hardware innovation****".*

– Stephen Neuendorffer, AMD Fellow

HW vendor support: TOSA compatibility already stated by multiple NPU vendors

Summary of TOSA components

Specification Components

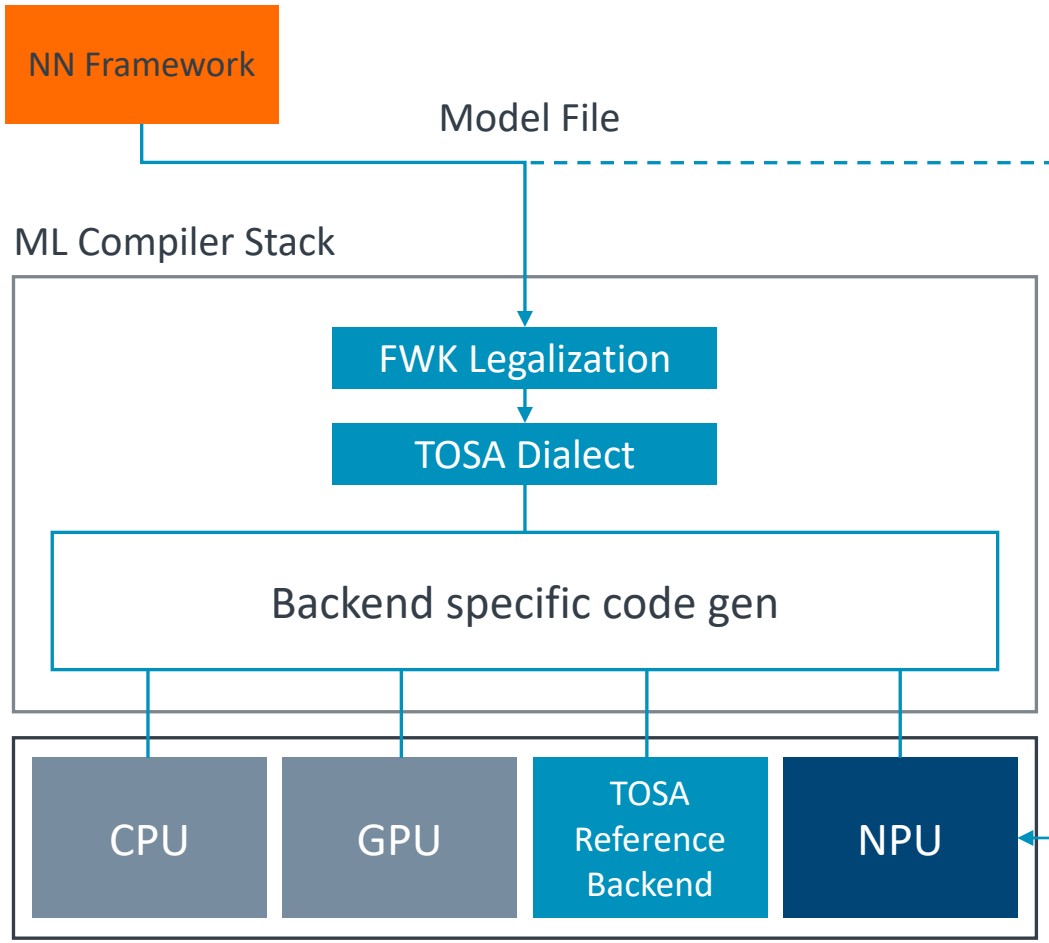
TOSA Specification:

- Arguments
- Code Construction
- Data Types

TOSA Profiles:

Profile	Name
Base Inference	TOSA-BI
Main Inference	TOSA-MI
Main Training	TOSA-MT

MLIR Components



Tooling Components

Using TOSA in Your Solution

TOSA will become a cornerstone of Machine Learning for future generations of Arm-based IP

- App Developers / Neural Network Developers:
  - Write your neural networks to be TOSA compliant ensuring portability to a diverse range of targets
- Silicon Designers:
  - Design your solution to be TOSA compliant ensuring compatibility with the fast moving frameworks
- Device Manufactures:
  - Encourage your SoC suppliers to implement TOSA compliance in their hardware to enable multi-target portability
- Compiler Engineers:
  - Add support for new operators to TOSA legalization in MLIR to progress industry standardization

https://www.mlplatform.org/tosa/

arm AI

AI Virtual Tech Talks Series

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה

arm
DEVSUMMIT
OCT 26–NOV 9

Build Your Future on arm

Join other future builders and get the insights and skills to develop and deploy on Arm-based hardware.

devsummit.arm.com

Register for Free Now!

arm AI

Thank you!

Tweet us: @ArmSoftwareDev -> #AIVTT

Check out our Arm Software Developers YouTube channel

Signup now for our next AI Virtual Tech Talk: www.arm.com/techtalks