Acceleration of a Compute-Intensive Algorithm for Power Electronic Converter Control Using Versal AI Engines

Fernando Flores, Óscar López Sánchez, Luis Jacobo Álvarez Ruiz de Ojeda,

María Dolores Valdés Peña, José Manuel Villapún Sánchez

XXXIX Conference on Design of Circuits and Integrated Systems

13-15 November, 2024, Catania, Italy

This work was supported in part by MCIN/AEI/10.13039/501100011033 and FEDER, UE under Project PID2021-124136OB-I00.

Outline

Introduction

1

Background and motivation

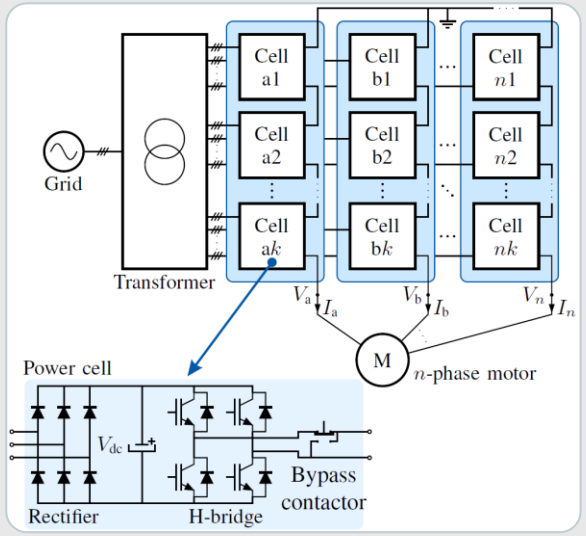
Modern power control applications require more compute-intensive algorithms.

Wide band-gap power semiconductors with higher switching frequencies.

More demanding timing requirements for power electronic converter control.

Use case demo

Cascaded H-bridge converter

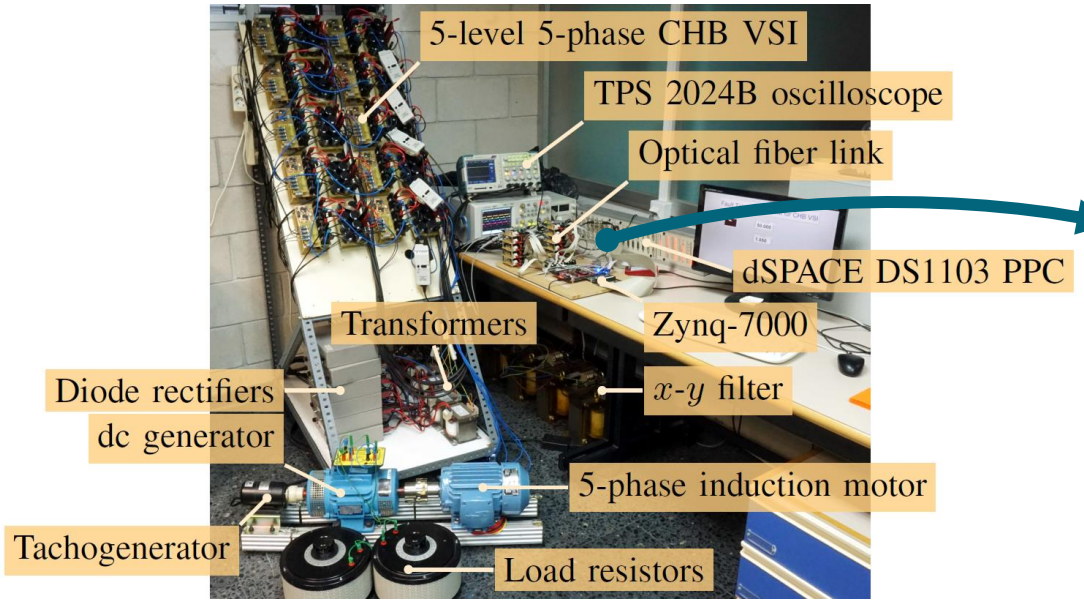


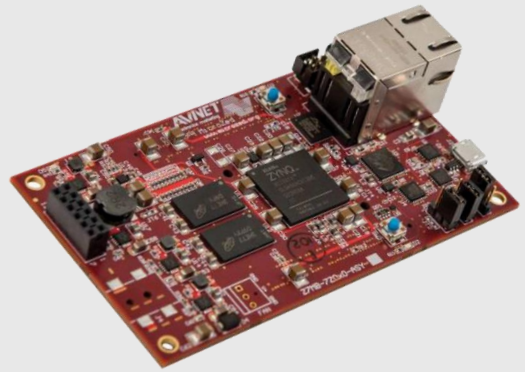
Fault in a power cell.

Power cell bypassed.

Need for post-fault strategy.

Minimum Infinity-Norm (MIN) post-fault strategy tested in Zynq-7000.

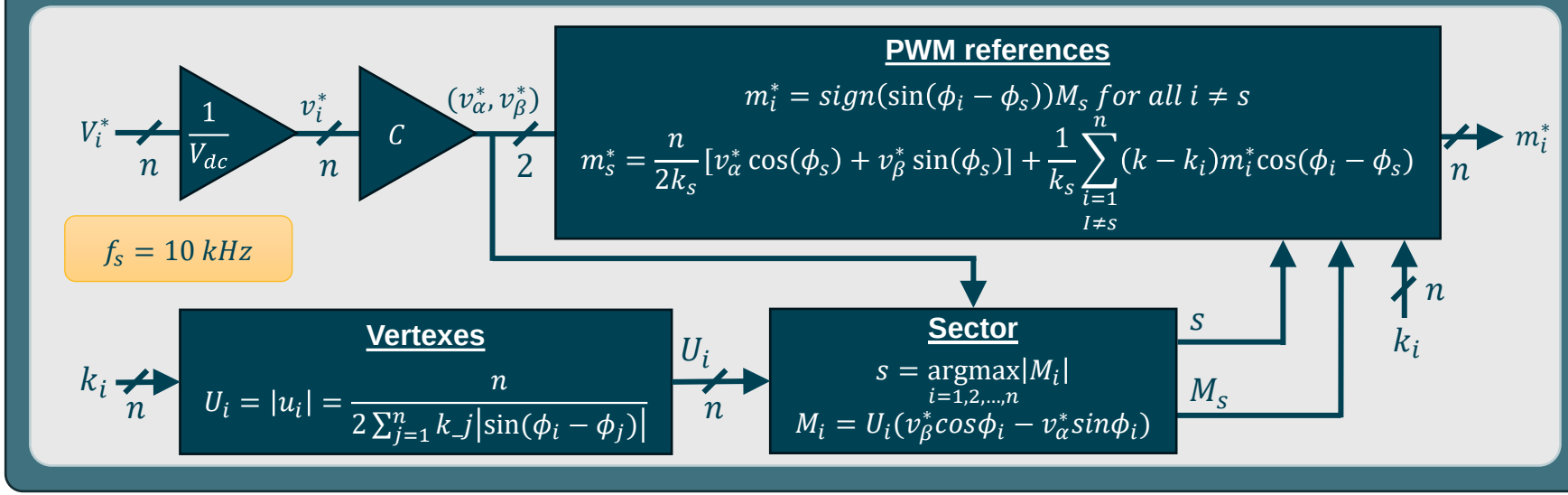


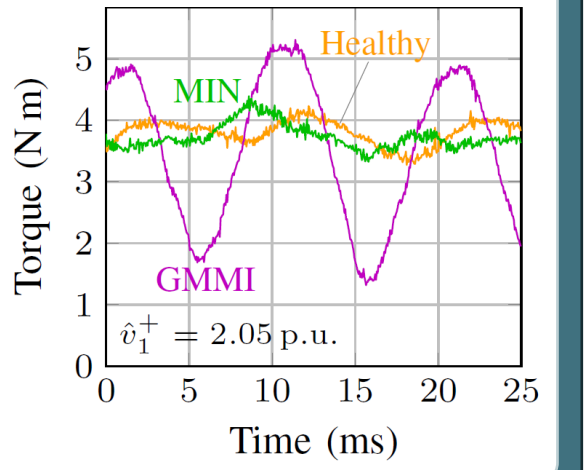


MicroZed board with Zynq-7000 SoC.

Minimum Infinity-Norm (MIN) base algorithm

Reduces torque ripple during power cell faults while maximizes utilization of the DC-link resources.

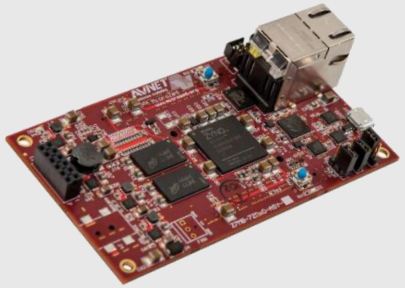




Algorithm implementation in two embedded architectures

TE0950 board with Versal SoC.

VS



MicroZed board with Zynq-7000 SoC.

Objectives

Comparison of Versal ACAP and Zynq-7000 architectures for the implementation of a power electronic converter control algorithm.

Feasibility analysis of adapting a power electronic converter control algorithm to the Versal AI Engines architecture.

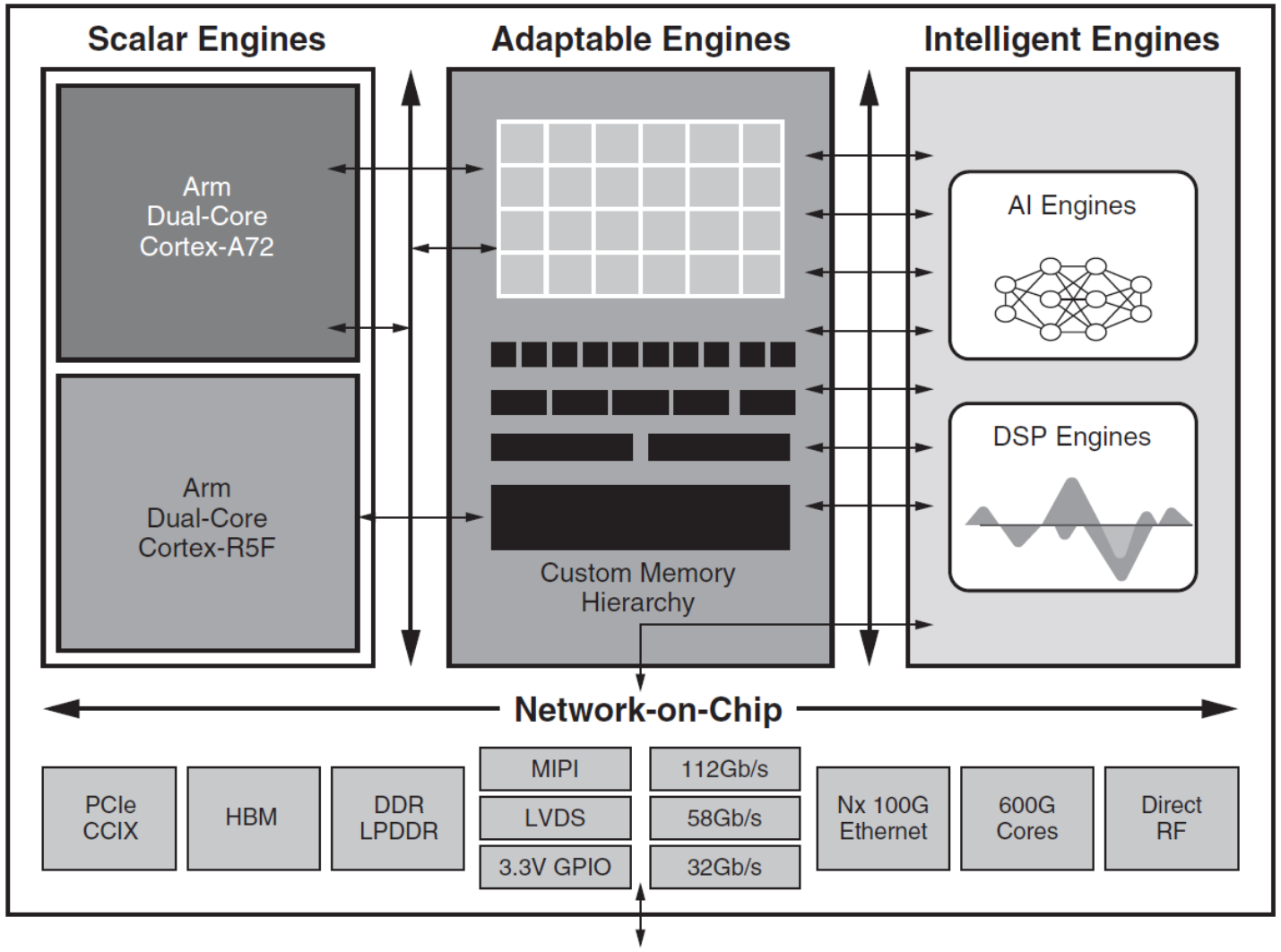
Acceleration of the Minimum Infinity-Norm algorithm using Versal AI Engines as a demostrator.

indra

Versal ACAP Architecture

2

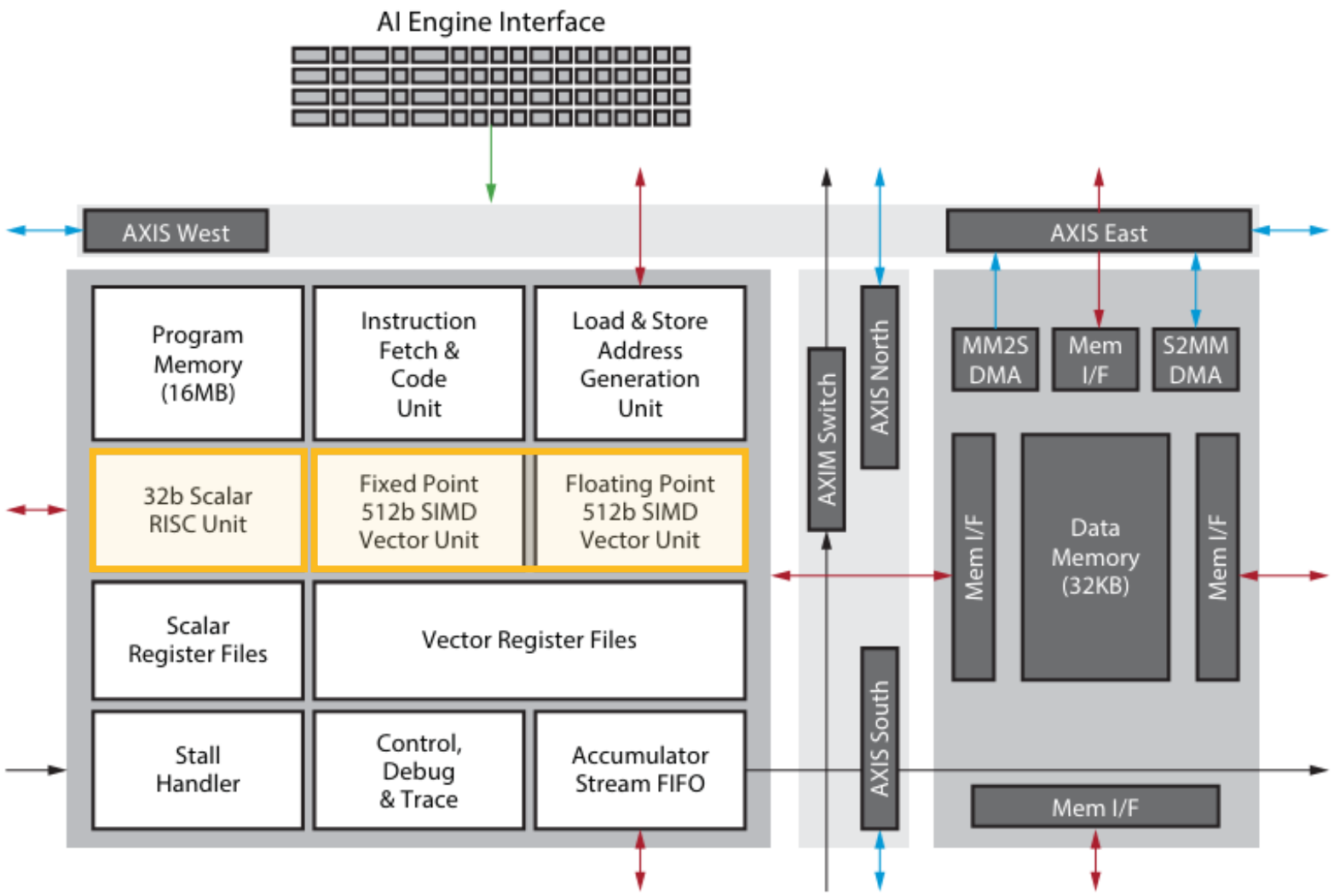
Versal ACAP architecture



New key features

- 7 nm fabric logic.
- DSP58s: Located in the PL.
- AI Engines.
- Hardened Network on Chip (NoC).

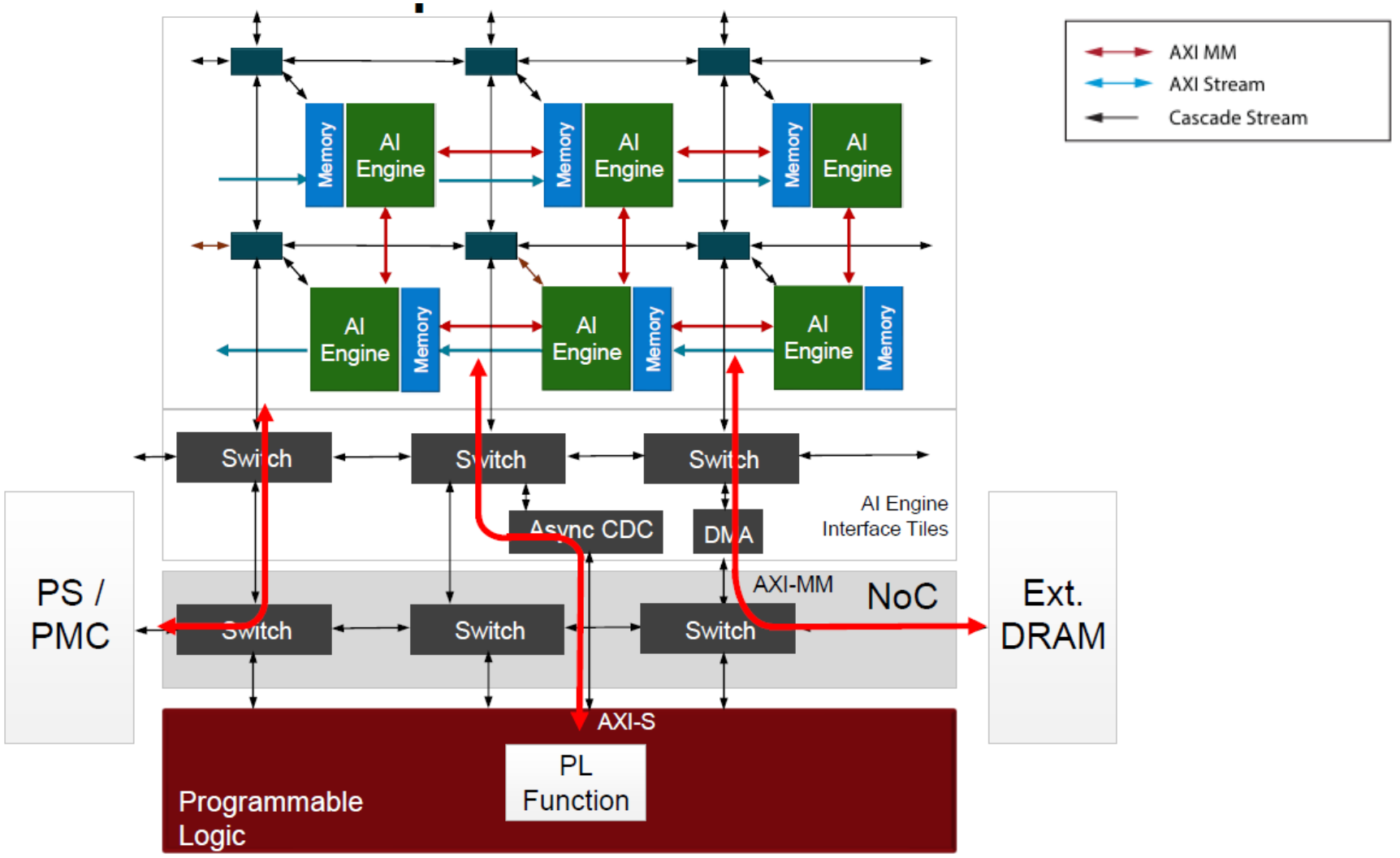
Versal AI Engine architecture



New key features

- Scalar and vectorial processing units.
- 16 KB program memory and 32 KB local data memory.
- Working frequency can be either 1 or 1.25 GHz.
- Vector unit optimized for INT16/CINT16 data types.

Versal AI Engines array connectivity



Adaptation and implementation of the MIN algorithm

3

MIN algorithm implementation overview

Processing System (PS)

- Data types quantization.

Programable Logic (PL)

- Data types quantization.
- Algorithm logic adaptation.

Versal AI Engine scalar unit

- Data types quantization.
- Data memory optimization.
- Program memory optimization.

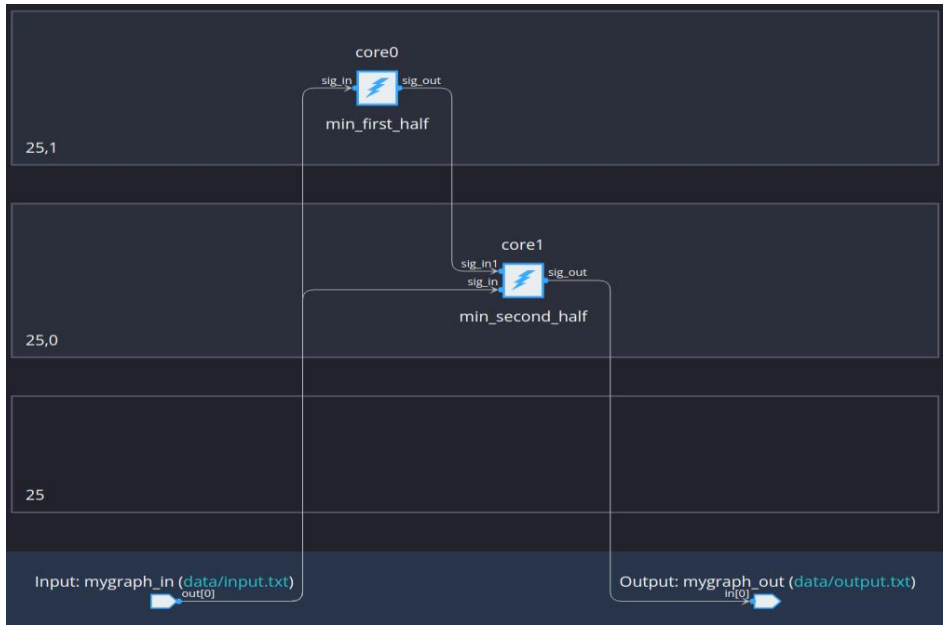
Versal AI Engine vector unit

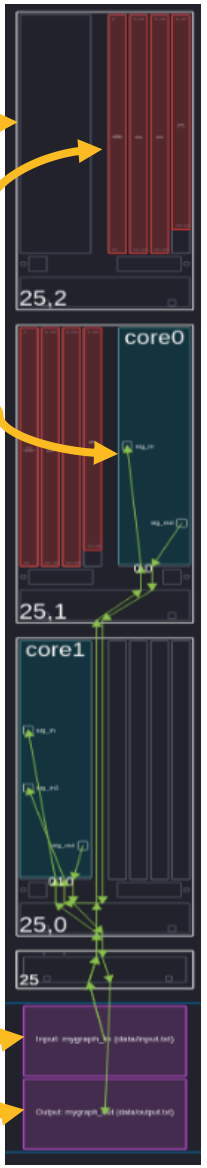
- Data types quantization.
- Data memory optimization.
- Program memory optimization.
- Adaptation to intrinsics.

MIN algorithm implementation overview

Versal multiple AI Engines: scalar and vector units

- Data types quantization.
- Data memory optimization.
- Program memory optimization.
- (Adaptation to intrinsics).
- MIN Algorithm partitioning.





1. AI Engine Tile
2. Tile Memory
3. Processing kernel
4. PL input interface
5. PL output interface

Experimental Results and Discussion

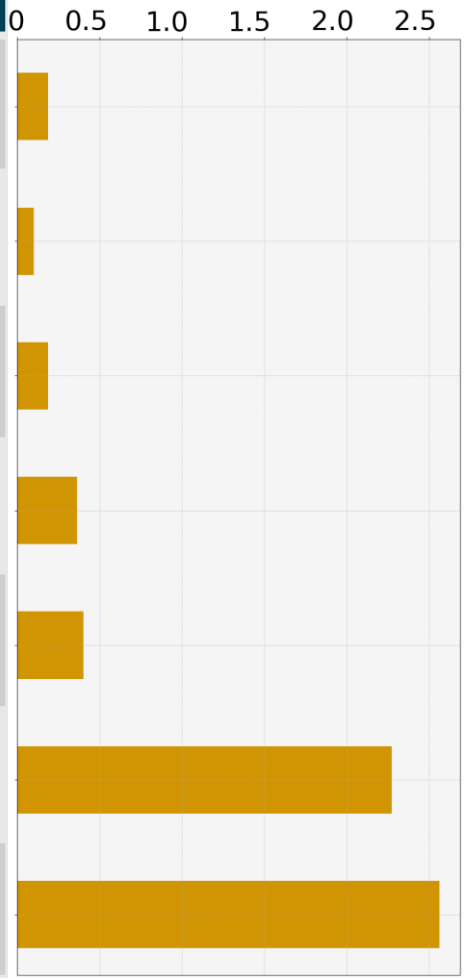
4

Performance review

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)

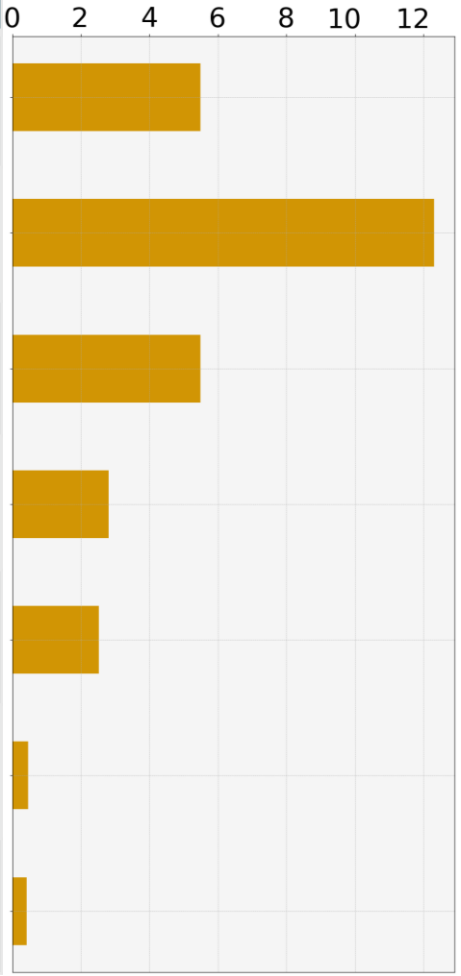
Performance review: Throughput

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)



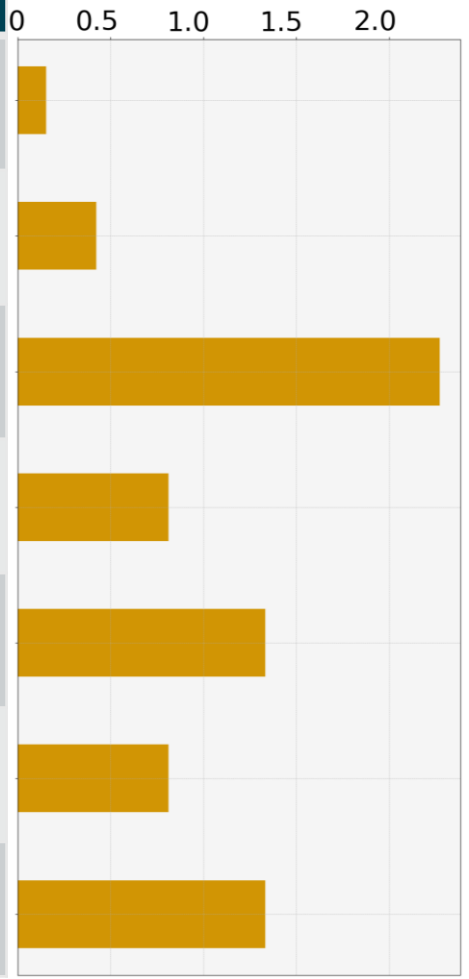
Performance review: Latency

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)



Performance review: Power

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)



Performance review: Hardware cost

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)

Performance review: Use case highlights

Test case	Throughput	Latency	Power	Hardware cost
Zynq-7000 PL	0.18 MSps	5.47 µs	0.15 W	5,809 LUTs, 336 LUTRAMs, 584 FFs, 4 DSPs
Zynq-7000 PS	0.10 MSps	12.3 µs	0.42 W	-
Versal PL	0.18 MSps	5.47 µs	2.27 W	6,112 LUTs, 336 LUTRAMs, 587 FFs, 4 DSPs
1 AI Engine scalar unit	0.36 MSps	2.79 µs	0.81 W	1 AI Engine
2 AI Engine scalar units	0.40 MSps	2.51 µs	1.33 W	3 AI Engine (2 for compute)
1 AI Engine vector unit	2.27 MSps	0.44 µs	0.81 W	1 AI Engine
2 AI Engine vector units	2.56 MSps	0.39 µs	1.33 W	3 AI Engine (2 for compute)

Conclusions and Future Work

5

Conclusions and future work

- Versal AI Engines are suitable for implementing power electronic converter control algorithms.
- Versal AI Engines outperform Zynq-7000 in terms of latency and throughput for the implementation of the Minimum Infinity-Norm algorithm.
- Versal PL and AI Engines rise power consumption for these kind of algorithms.
- Applicability to high-speed electrical machines with severe timing constraints and to predictive control algorithms for converters.

Acceleration of a Compute-Intensive Algorithm for Power Electronic Converter Control Using Versal AI Engines

Fernando Flores, Óscar López Sánchez, Luis Jacobo Álvarez Ruiz de Ojeda,

María Dolores Valdés Peña, José Manuel Villapún Sánchez

XXXIX Conference on Design of Circuits and Integrated Systems

13-15 November, 2024, Catania, Italy

This work was supported in part by MCIN/AEI/10.13039/501100011033 and FEDER, UE under Project PID2021-124136OB-I00.