Benchmarking Floating Point Performance of Massively Parallel Dataflow Overlays on AMD Versal Compute Primitives

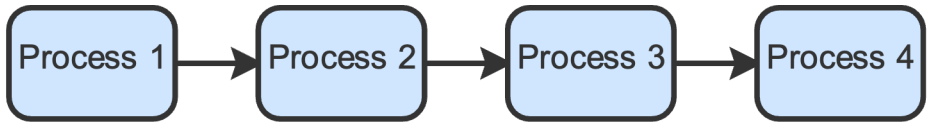
Mohamed Bouaziz, Suhaib A. Fahmy

Accelerated Connected Computing Lab (**ACCL**),

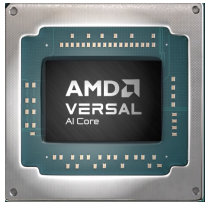
King Abdullah University of Science and Technology (**KAUST**), Saudi Arabia

Floating-Point Numbers in Dataflow

- Floating-point numbers are widely used in many HPC (and AI) applications.
- General-purpose architectures come with optimised parallel FP units, such as AVX and SVE extensions.
- GPU architectures, such as Nvidia Ampere, come with FP tensor units.
- Dataflow-based processing does not fit these general-purpose architectures.



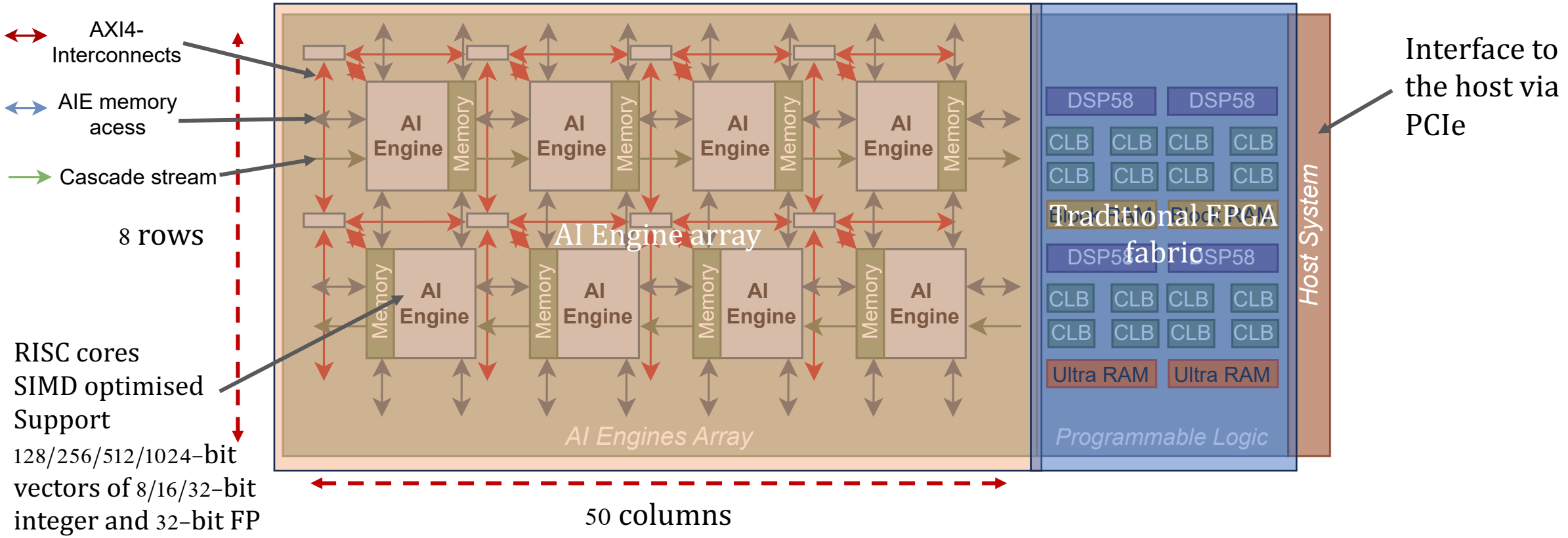
- CGRAs/RDAs are well-suited for running dataflows in pipelines.
- AMD Versal SoC allows the implementation of dataflow overlays through different levels of reconfigurability.



AMD Versal SoC[1]

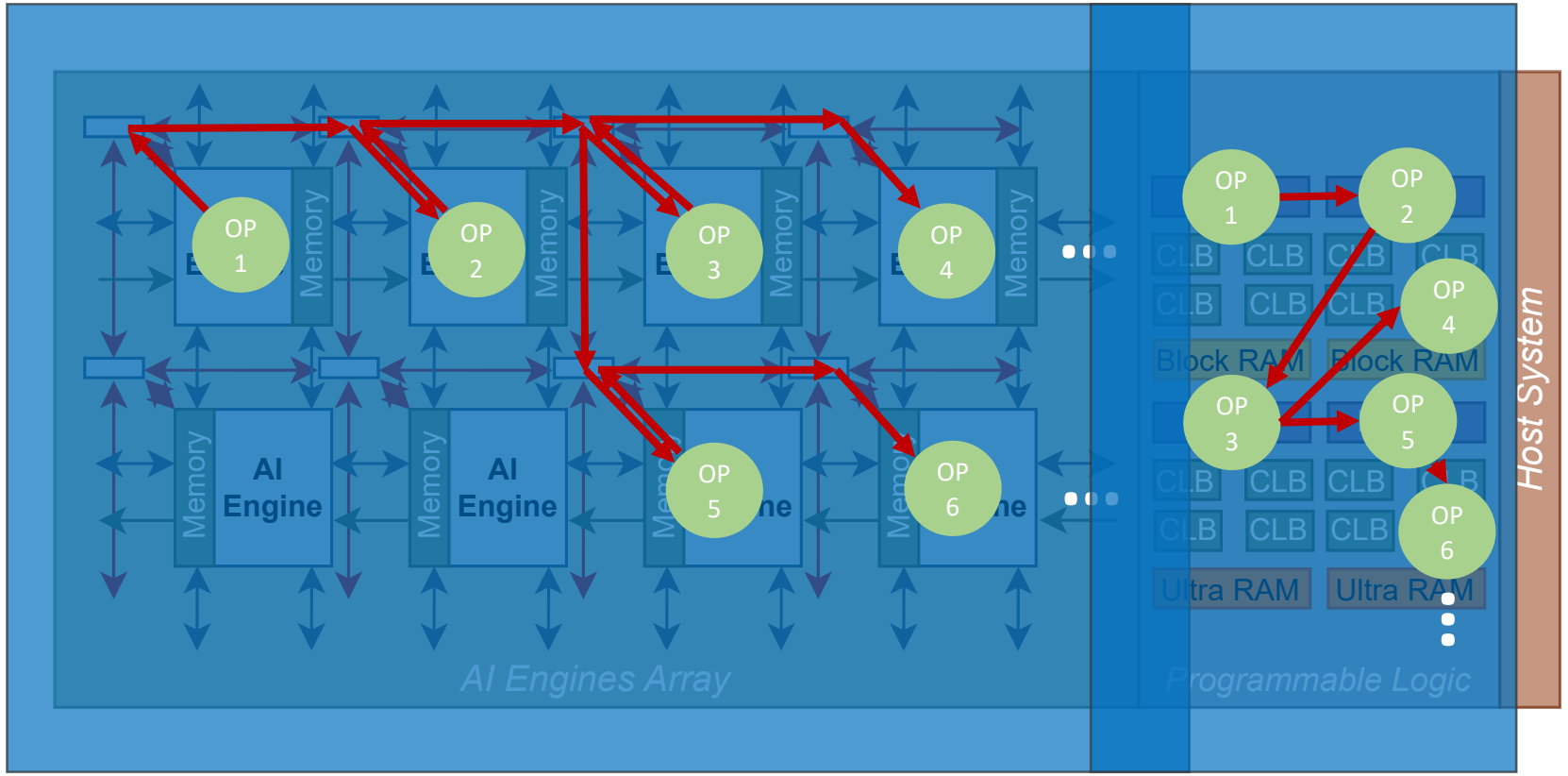
1: https://www.amd.com/en/products/adaptive-socs-and-fpgas/technologies/ai-engine.html

AMD Versal SoC architecture



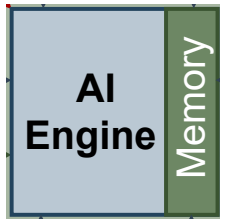
- RISC cores
- SIMD optimised
- Support 128/256/512/1024-bit vectors of 8/16/32-bit integer and 32-bit FP elements.

Dataflow Overlay



AMD Versal's Floating–Point Primitives

AI Engine array side

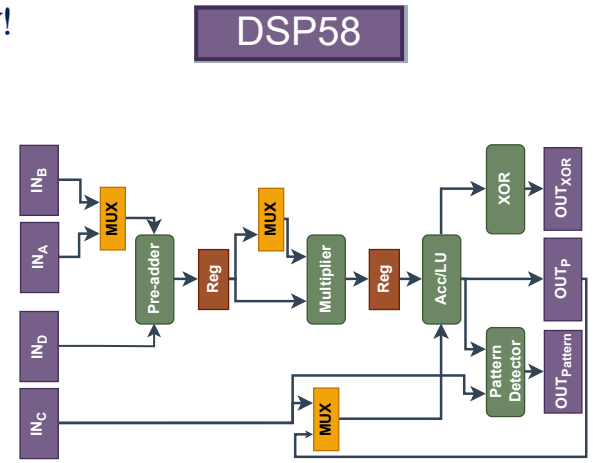


- Both support FP operations natively!
- Older reconfigurable architectures, such as UltraScale+, don't!

```
void aie_vmul_stream(input_window<float> *in0,
                     input_window<float> *in1,
                     output_windw<float> *out) {
  aie::vector<float, 8> a = window_readincr_v<8>(in0);
  aie::vector<float, 8> b = window_readincr_v<8>(in1);
  aie::vector<float, 8> res = aie::mul(a, b);
  window_writeincr<8>(out, res);
}
```

- Runs based on a program
- 8 SPFP/cycle at 1GHz (x400 cores)
- Pipelined by construction

Programmable logic side



- Runs based on a configuration
- 1 SPFP/cycle at variable freq. (x1968 blocks)
- Pipeline stages can be configured

Main Contributions

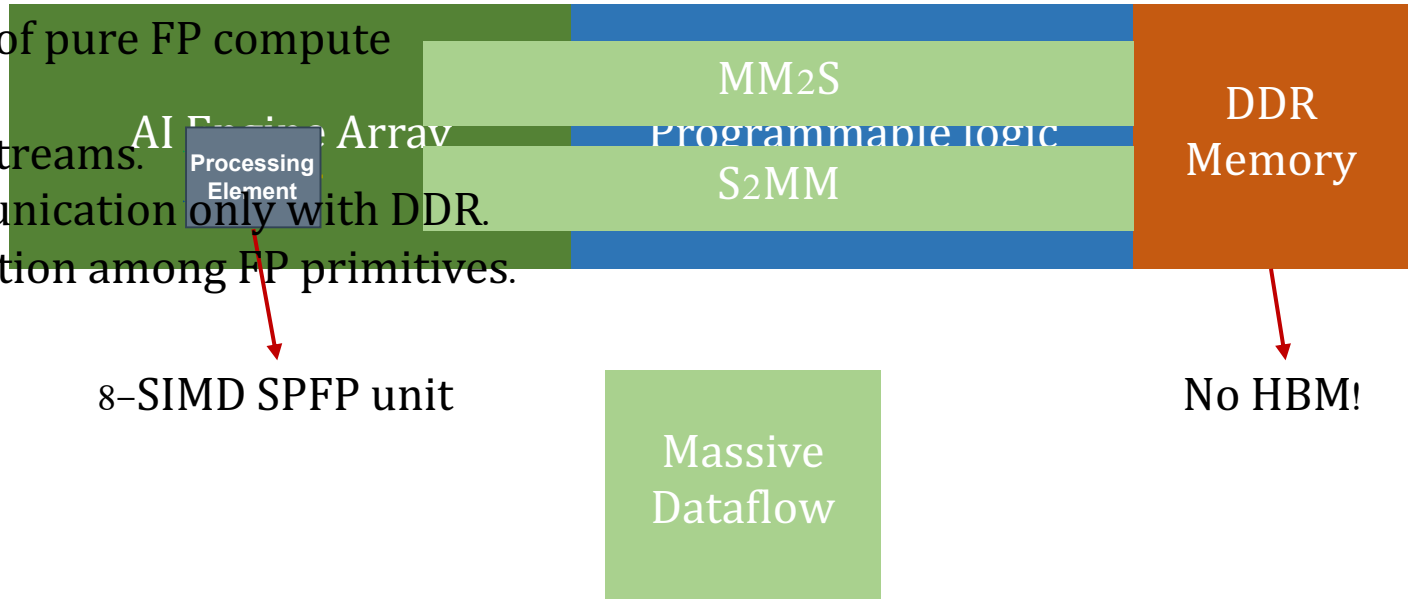
For a massive dataflow of parallel FP ops, what does performance look like? since:

- DSP frequency varies with dataflow size.
- DSP frequency varies with pipeline stages.
- Versal DSP58 blocks natively support FP ops while UltraScale + DSP48E2 don't.
- Power consumption varies.

Implementation Requirements

For the extraction of pure FP compute performance:

- Feed-forward streams.
- Off-chip communication only with DDR.
- No synchronisation among FP primitives.

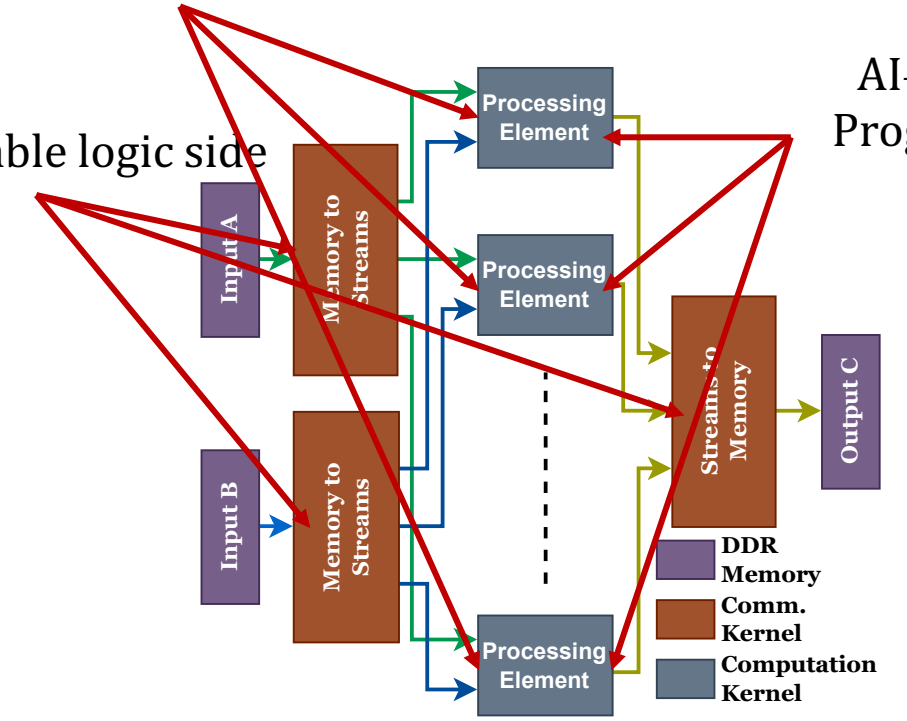


Proposed Architectural Model

8–SIMD element–wise vector multiplication

Programmable logic side

AI–Engine–Array / Programmable–logic side

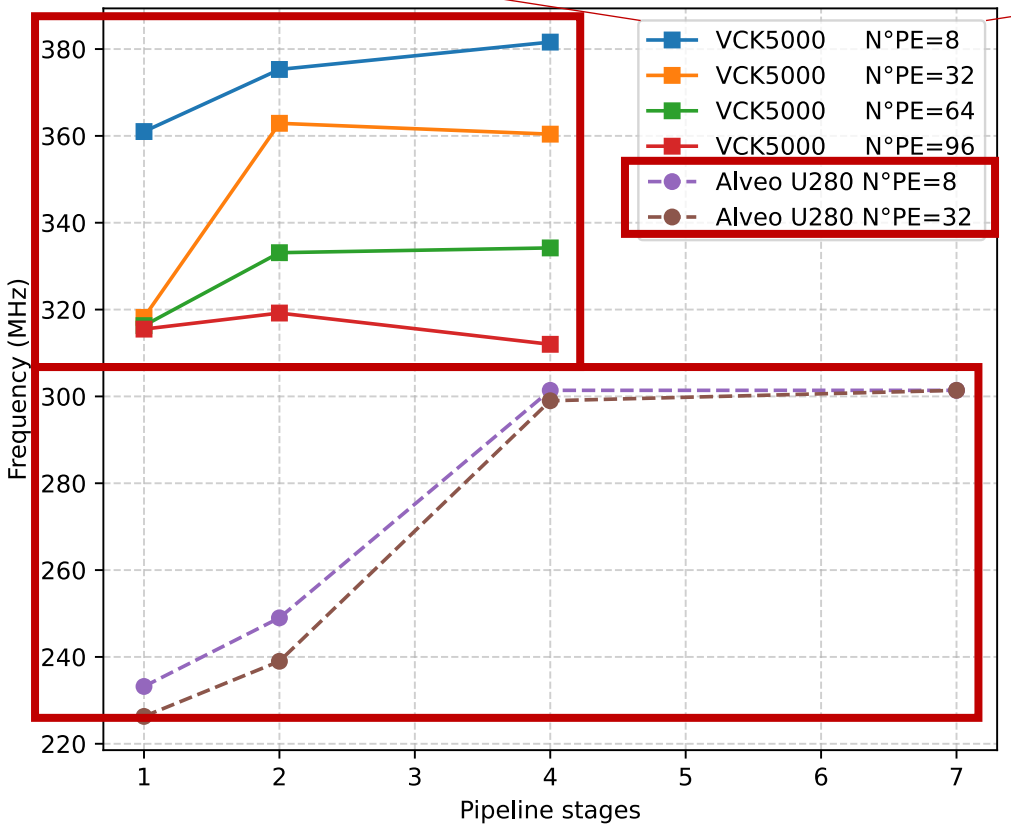


This is also implemented on the older UltraScale+ DSP48E2 (no native support for FP) for pipelining effect comparison!

Pipelining Effect

- Frequency somewhat scales with more pipeline stages on Versal DSP58
- The pipelining effect starts to vanish with more PEs.
- Frequency scales with up to 4 pipeline stages on UltraScale+ DSP48E2
- The scaling plateaus with more pipeline stages.

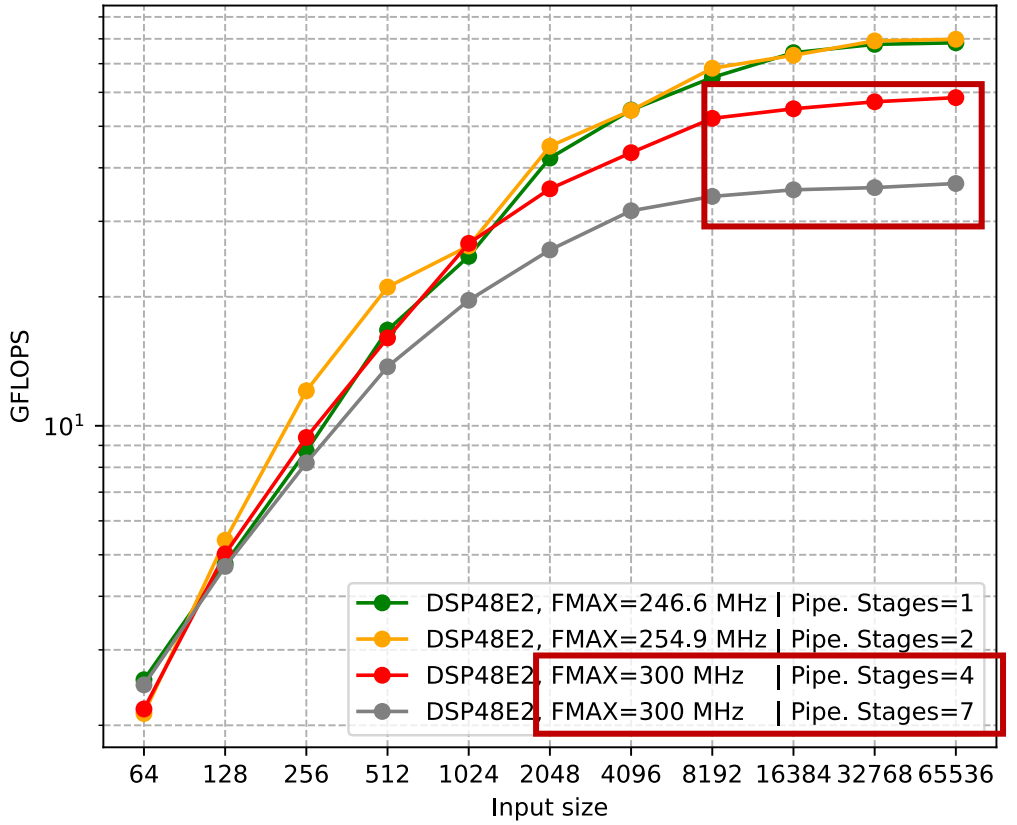
VCK5000 => Versal DSP58
Alveo U280 => UltraScale+ DSP48E2



UltraScale+ fails to implement more than 32 PEs due to high congestion.

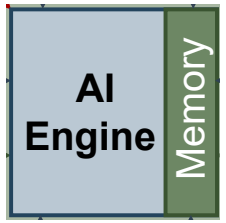
Pipelining Effect, contd...

- To isolate the effect of pipelining from the congestion issue on the UltraScale+, we pipeline a single PE.
- This allows measuring the impact of hardware overhead on throughput.
- The frequency plateaus at 4 stages.
- The throughput decreases with more data being moved around the extra hardware that comes with pipelining.



Higher Density Dataflow Performance

What about the AI Engine and > 96 DSP-PEs performance?

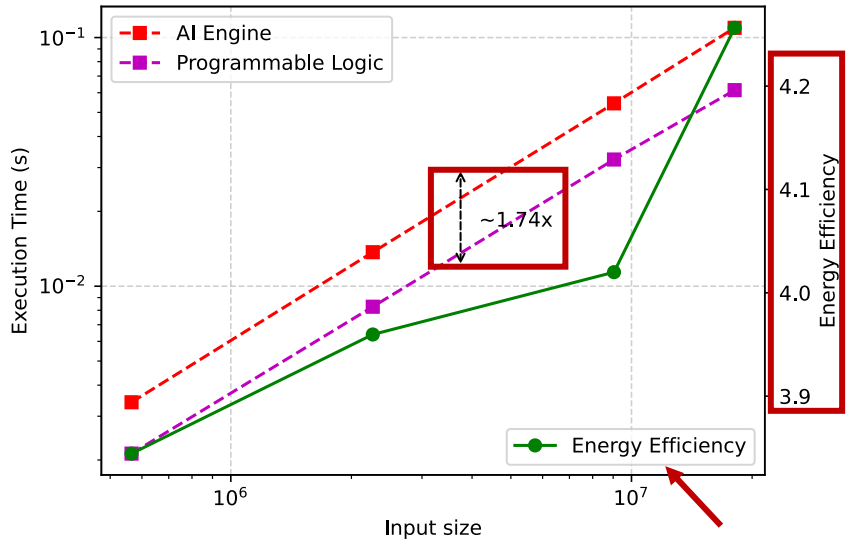


x384 (96%)

Maximum achievable resource usage without congestion

DSP58

184 x 8-SIMD PE = 1472 DSP58 (74%)



$$\frac{\textit{Energy consumed by AI Engines}}{\textit{Energy consumed by the DSP58s}}$$

- *For such a feed-forward application, the DSP58-based design is 1.74x faster than the AIE-based design.*
- *It is also more energy efficient with the DSP58-based design.*

BUT THAT'S ONLY WITH FEED-FORWARD APPLICATION (MatMul in CHARM, FPGA'23)

Conclusion

- Dataflow overlays can be mapped at different levels on reconfigurable hardware architectures.
- AMD Versal SoC allows mapping dataflow overlays on traditional FPGA fabric as well as on coarse – grained AI Engines array.
- Different primitives enable FP ops. efficiently on AMD Versal.
- For feed–forward applications, DSP58 can achieve higher throughput and lower energy consumption, although at a lower frequency.
- More compute patterns will be further evaluated to assess the achievable performance of FP ops. with more real–world constraints.

Thanks!

Benchmarking Floating Point Performance of Massively Parallel Dataflow Overlays on AMD Versal Compute Primitives

Github Repository

Reach out via email

Connect on LinkedIn