Packed SIMD Vectorization of the DRAGON2-CB

Riadh Ben Abdelhamid[1], Yoshiki Yamaguchi[2]

[1] Graduate School of Science and Technology, University of Tsukuba, Japan
[2] Faculty of Engineering, Information and Systems, University of Tsukuba, Japan

December 20th, 2022

Outline

Outline

Accuracy-flexible computing using FPGAs

Benefits of using FPGAs

- Re-Programmable on the hardware level and praised for their power-efficiency.
- Abundant hardware resources that can implement custom computing architectures.
- Energy-efficient Accuracy-flexible computation.

Packed SIMD (Single Instruction Multiple Data) to perform accuracy-flexible computation

- Can achieve energy-efficiency by reducing computing precision.
- Split N-bit FPU (Floating-Point Unit) into multiple M-bit Unit.
- Programmable instruction-set FPGA overlay can benefit from such an approach.

Outline

Floating-Point Representation

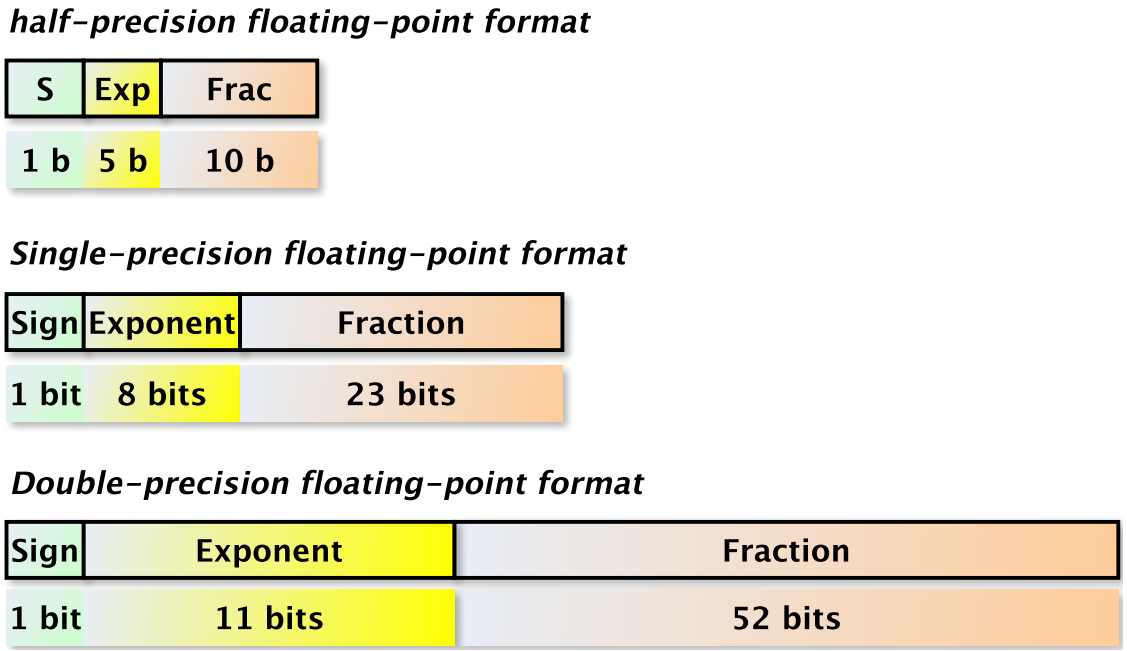


Figure – IEEE 754 standard half, single and double-precision floating-point formats.

Using fields to compute a Floating-point value

- $NormalFloat = (-1)^{S} \times (1.F)_2 \times 2^{E\text{-}B}$
- $SubNormalFloat = (-1)^{S} \times (0.F)_2 \times 2^{1\text{-}B}$

State-of-the-art Packed SIMD

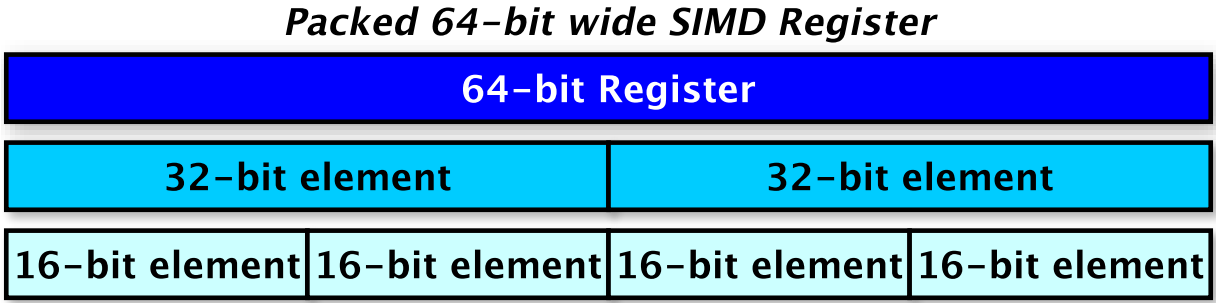


Figure – Packing of 32-bit wide and 16-bit wide data elements into a packed SIMD 64-bit wide register.

Example packed SIMD from the industry

- Intel Advanced Vector Extensions (AVX) [8,9,10].
  (16 registers, each can hold 4 FP64 double-precision or 8 FP32 single-precision.)
- ARM Cortex-A15 can support up to 16 concurrent operations using 128-bit vector registers (ARM NEON technology)[11].
- Visual Instruction Set (VIS) [12] can hold several 8, 16, or 32-bit integer values while reusing existing SPARC V9 CPU 64-bit floating-point registers.

Outline

The DRAGON2 many-core overlay

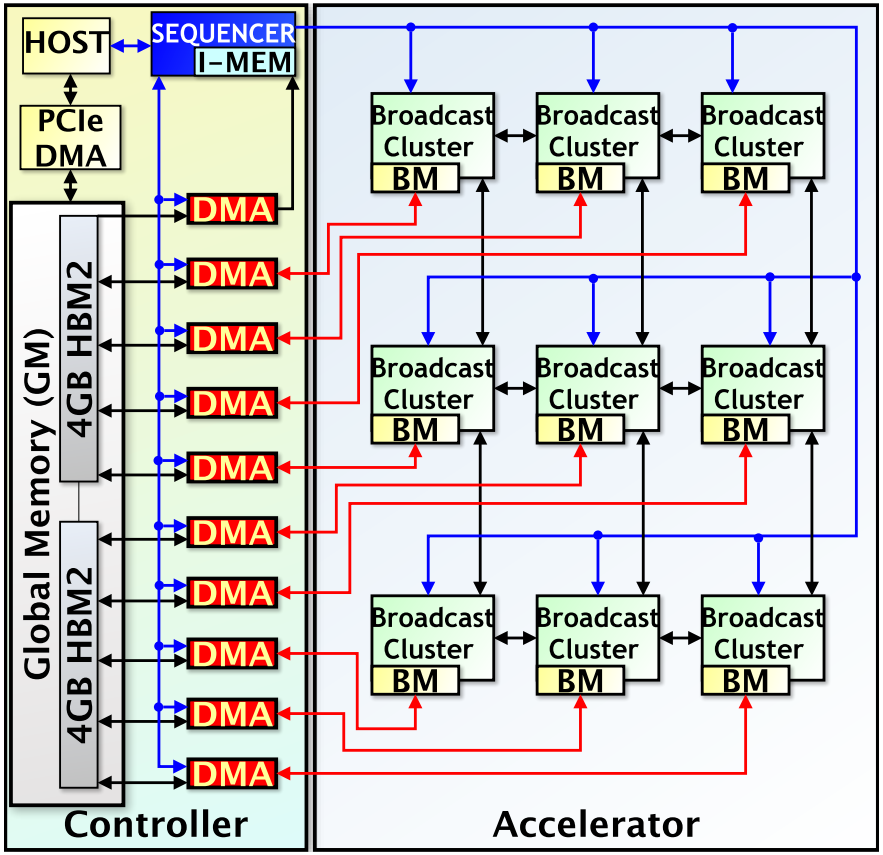


Figure – Overview of the DRAGON, DRAGON2 and DRAGON2-CB many-core overlay architecture [*, **].

configuration

- Accelerator (contains multiple interconnected clusters of 4x4 PEs, each.)
- Controller (contains control logic and issues instructions)

[*] Riadh Ben Abdelhamid et al. 2021. A Highly-Efficient and Tightly-Connected Many-Core Overlay Architecture. IEEE Access 9 (2021), 65277–65292.
[**] Riadh Ben Abdelhamid, Yoshiki Yamaguchi, and Taisuke Boku. 2022. A scalable many-core overlay architecture on an HBM2-enabled multi-die FPGA. ACM Trans. Reconfigurable Technol. Syst. Just Accepted (July 2022). https ://doi.org/10.1145/3547657

FPU-only DRAGON2-CB-FP64 Processing Element

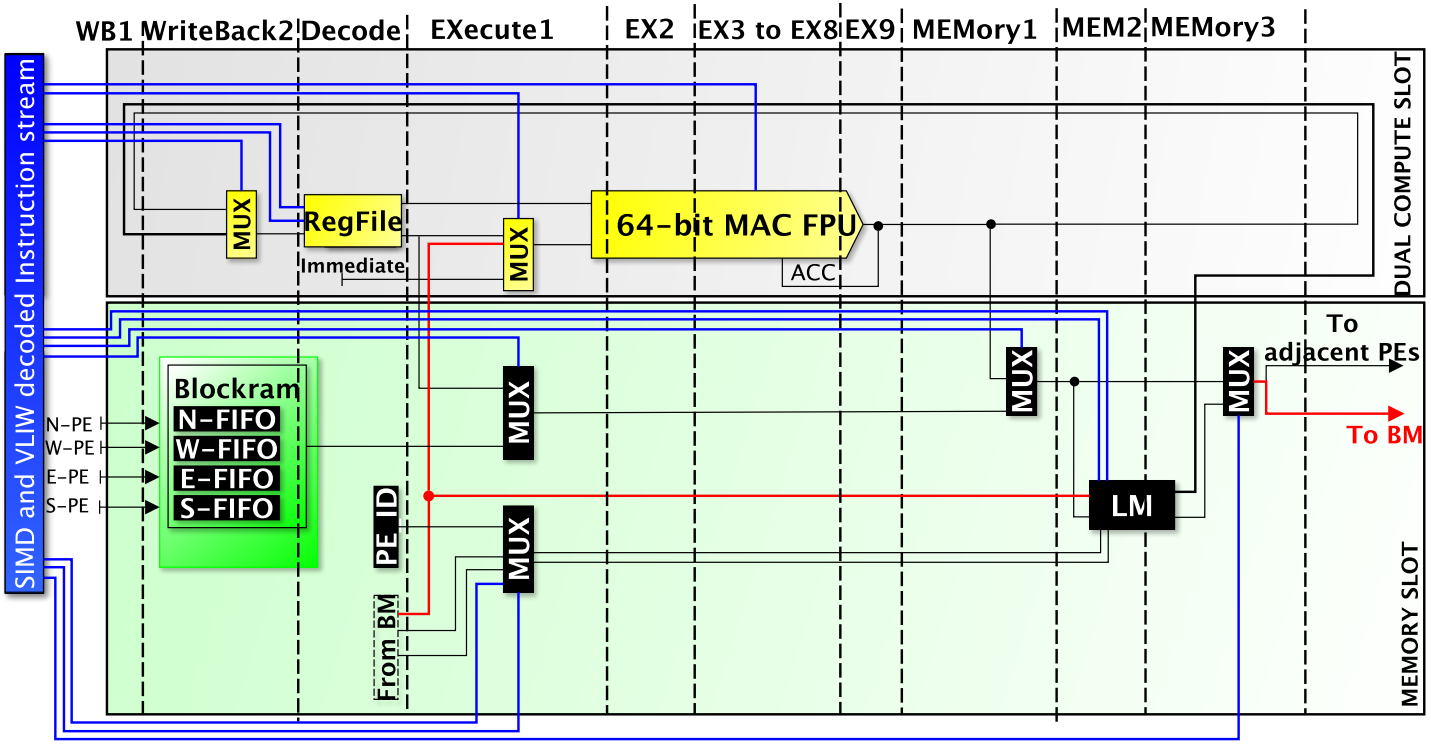


Figure – Architecture of the Processing Element. The ALU is removed to simplify the study.

Vectorization approach (FP32 Single-Precision)

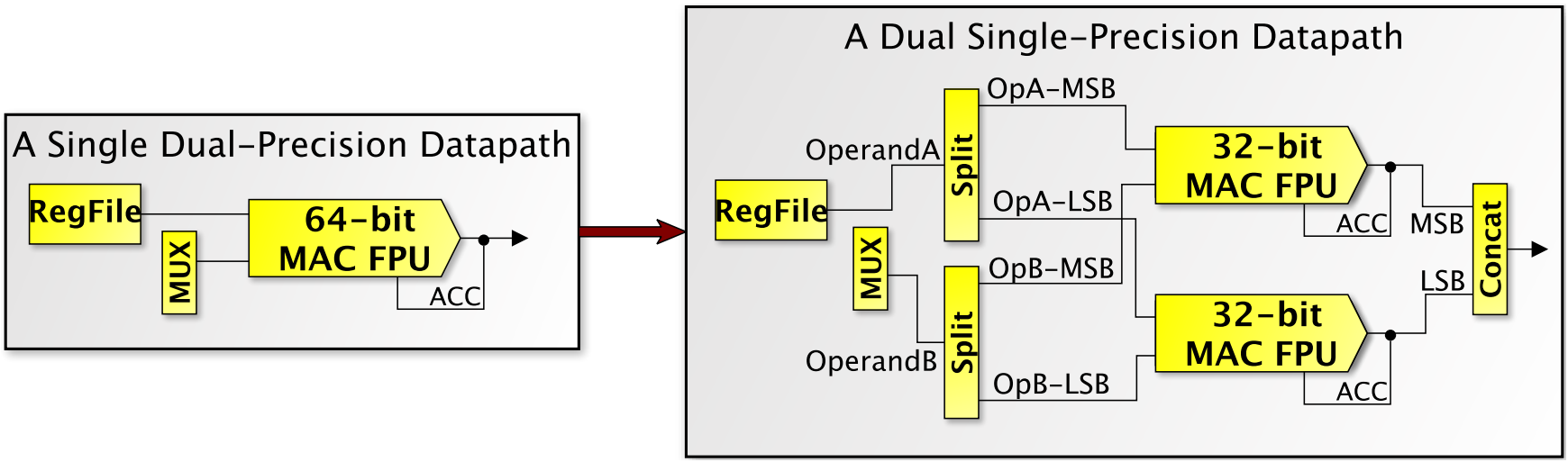


Figure – Single dual-precision datapath as compared to a dual single-precision datapath.

The way it works

- Re-use the same 64-bit registers from RF.
- Programmer manages packing two 32-bit words inside each 64-bit data at the program level.
- Implement two 32-bit FPUs instead of a single 64-bit FPU.
- Split the original operands inputs in hardware implementation to forward MSB parts of each input to the upper 32-bit FPU and LSB parts to the lower 32-bit FPU.
- Concat the outputs of both 32-bit FPUs back into a single 64-bit data.

Vectorization approach (FP16 Half-Precision)

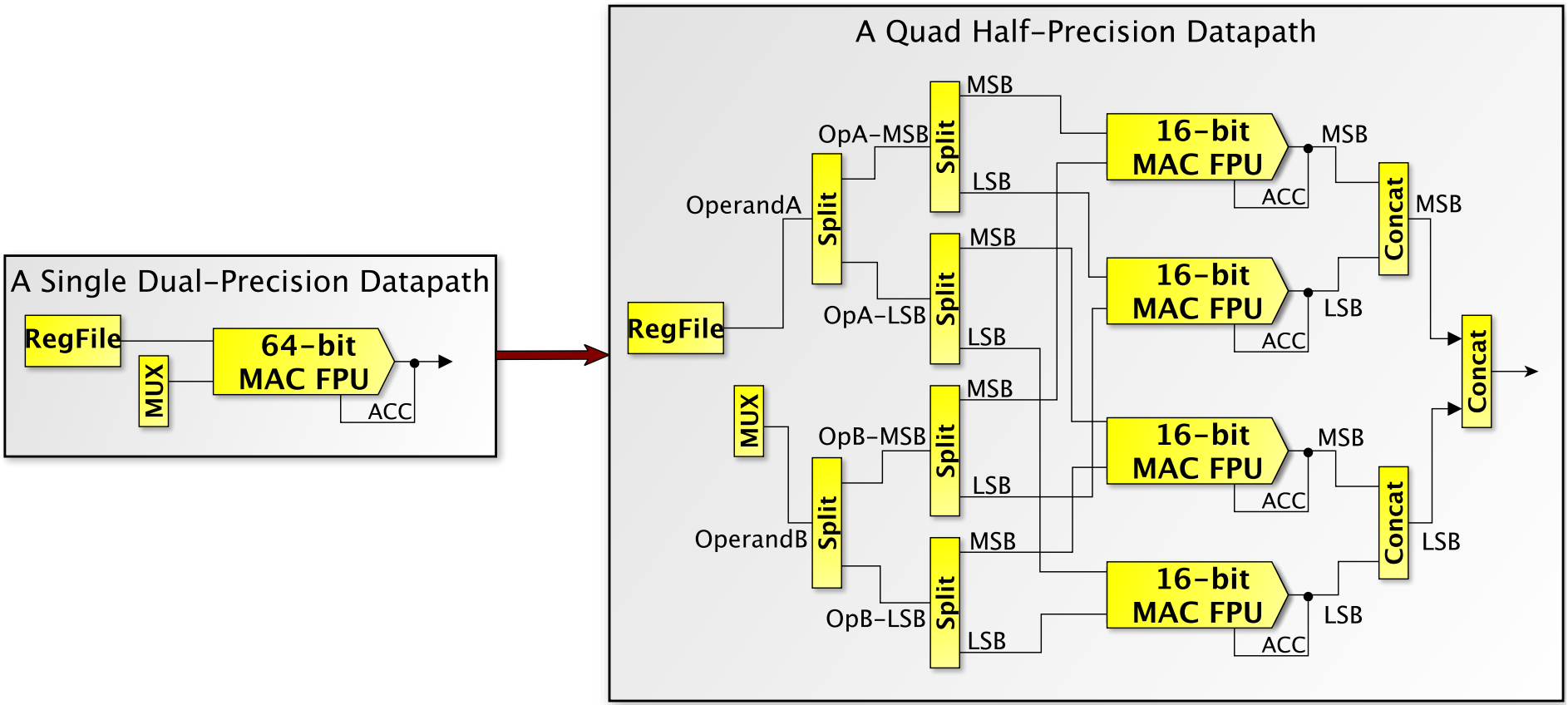


Figure – Single dual-precision datapath as compared to a quad half-precision datapath.

Outline

Evaluation : Area comparison (PE-level break-down)

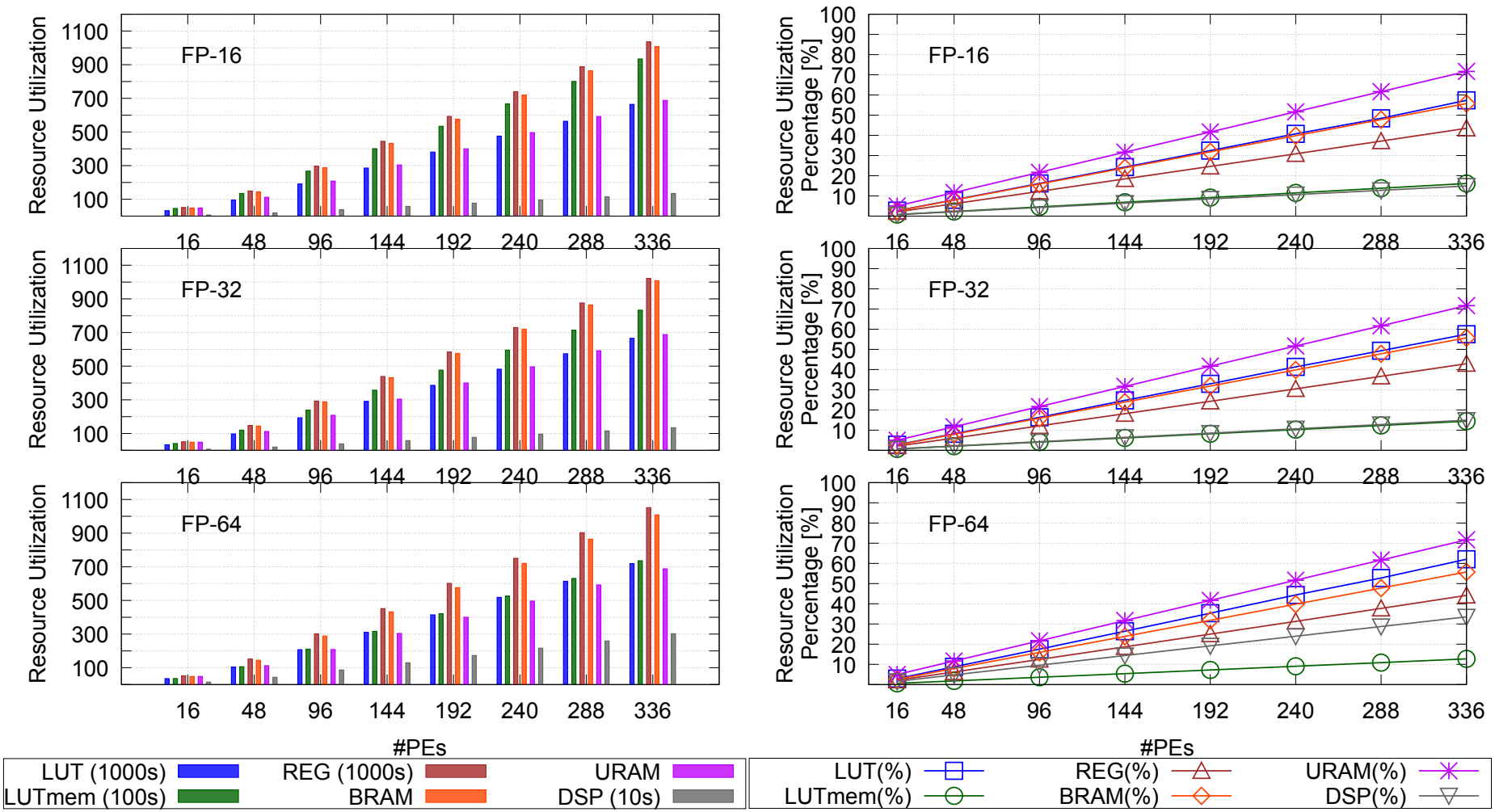
Table – Resource utilization comparison between half-, single- and double-precision MAC FPU as well as the Processing Element and its slots.

	LUT			LUTmem			REG			DSP		
Module	**FP-16**	**FP-32**	**FP-64**	**FP-16**	**FP-32**	**FP-64**	**FP-16**	**FP-32**	**FP-64**	**FP-16**	**FP-32**	**FP-64**
FPAdder	173	423	920	2	2	2	173	343	689	0	0	0
FPMult	49	60	215	29	49	75	72	130	349	1	2	9
MAC FPU	260	549	1265	35	55	81	281	541	1171	1	2	9
DCS	1136	1192	1361	182	152	123	1524	1482	1571	4	4	9
MS	333	333	334	92	92	92	859	859	859	0	0	0
PE	1469	1525	1695*	274	244	215	2383	2341	2430	4	4	9

* There is a typo in the paper where the value appears as 11695, this value is corrected here and should be 1695, that is the sum of LUTs in both the DCS (Dual Compute Slot) and the MS (Memory Slot).

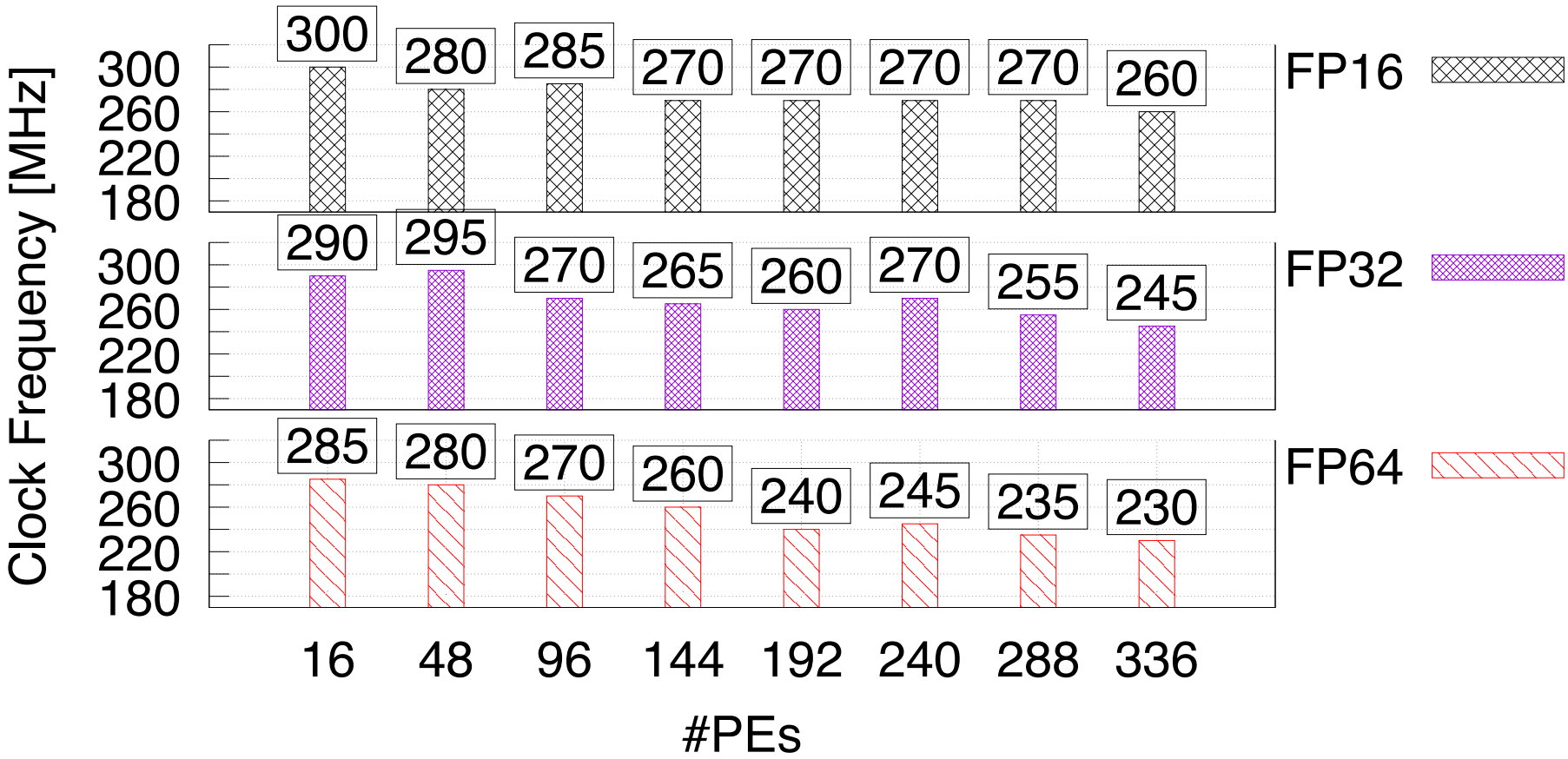
- FP64 consumes more DSPs due to the large significand multiplication (53x53).
- Comparable utilization for other kinds of hardware resources.

Evaluation : Area scalability (4xFP16 vs 2xFP32 vs 1xFP64)



- On-chip memory resources (URAM and BRAM unchanged) ✓.
- Varying precision maintains the same good scalability. ✓.

Evaluation : Clock speed (4xFP16 vs 2xFP32 vs 1xFP64)



The clock speed in FP64 implementations degrades faster with the increase of #PE (More DSPs complicates routing).

Reduced precisions achieve the best clock speed outcomes ✓.

Environment setup and Evaluation (Stencil benchmarks)

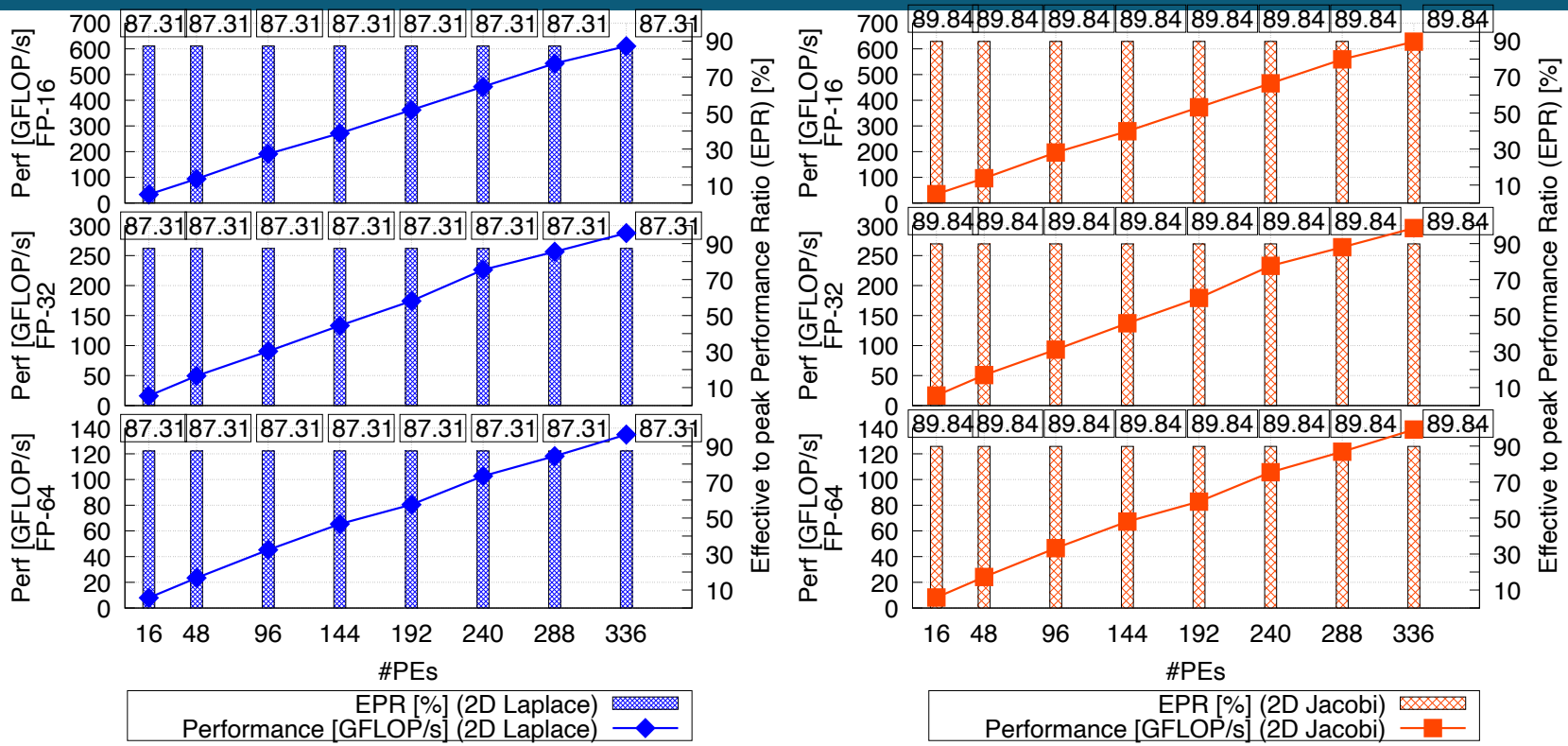
Table – Environment setup used in the experiments.

CPU (FPGA host)	Intel Core i9-9900K CPU 3.60GHz (8 cores), 64GB DDR4 RAM
Operating system (FPGA host)	Ubuntu 18.04.1 LTS
Accelerator	Alveo U280 Data Center Accelerator Card
FPGA Compiler	Xilinx Vitis 2020.2 (64-bit)

Table – Benchmarks used in the experimental evaluation.

Benchmark	Equation
2D Laplace	$U_{i,j}^{t+1} = (1/4) \times (U_{i-1,j}^{t} + U_{i+1,j}^{t} + U_{i,j-1}^{t} + U_{i,j+1}^{t})$
2D Jacobi	$U_{i,j}^{t+1} = c_{WEST}.U_{i-1,j}^{t} + c_{EAST}.U_{i+1,j}^{t} + c_{CENTER}.U_{i,j}^{t} + c_{NORTH}.U_{i,j-1}^{t} + c_{SOUTH}.U_{i,j+1}^{t}$

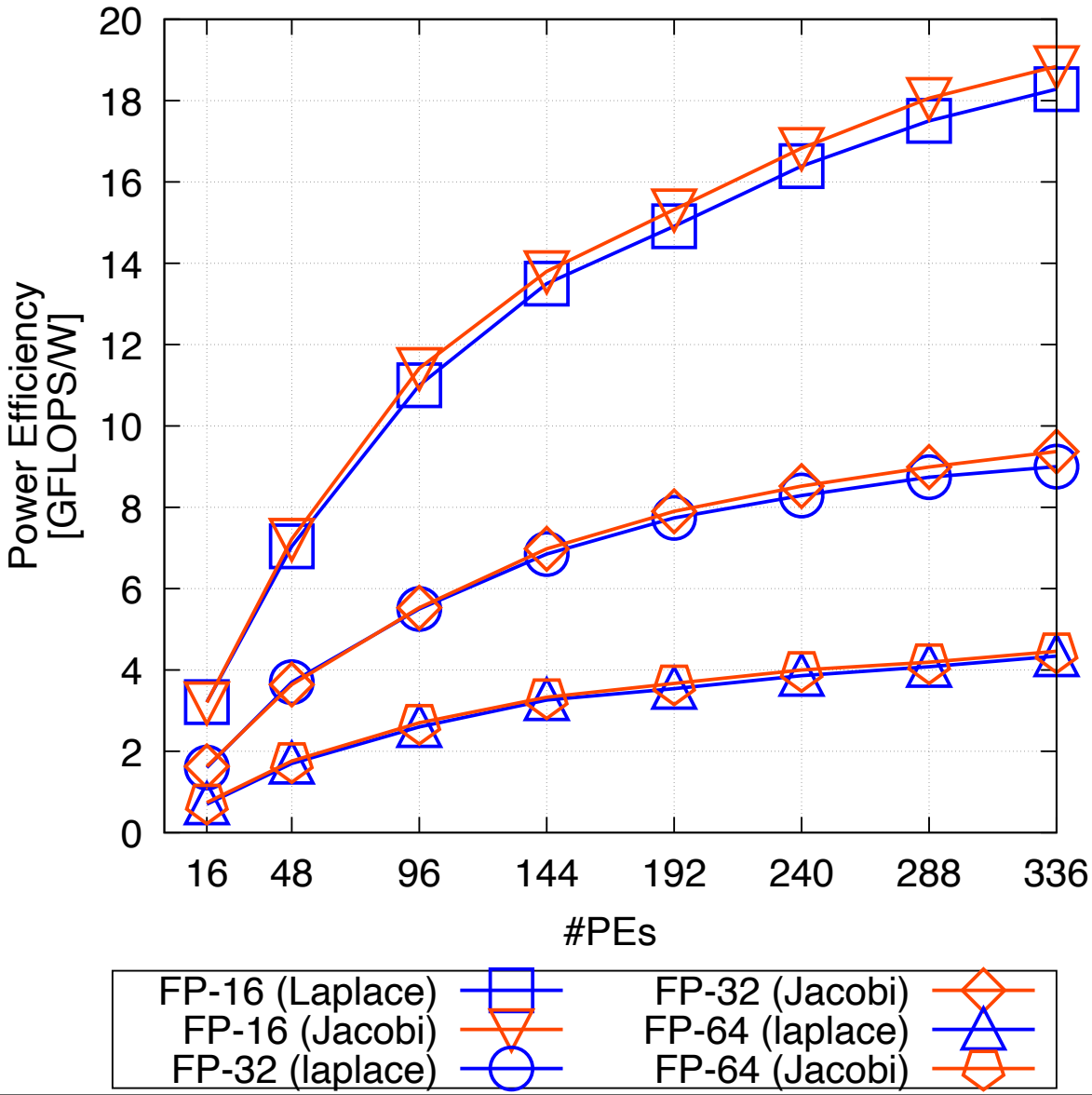
Evaluation : Sustained Performance and EPR (4xFP16 vs 2xFP32 vs 1xFP64)



- **EPR** : Effective-to-peak Performance Ratio.
- **EPR** = SP (Sustained Performance) / TPP (Theoretical Peak Performance)
- **SP** = #OPs / (Execution Time)
- **TPP** = 2 x #PEs x Freq

✓Constant EPR and Good Performance scalability.

Evaluation : Power-efficiency (4xFP16 vs 2xFP32 vs 1xFP64)



Precision is halved
$\Longrightarrow$
Power-efficiency approximately doubles.

(Higher precision + increased #PEs)
$\Longrightarrow$ More clock frequency degradation.
$\Longrightarrow$ More power-efficiency degradation.

Outline

Summary : Best case Performance, Power Efficiency and EPR

	Ref	This work (DRAGON2-CB-FP)			[6] (DRAGON2-CB)	
	Module	Four FP-16 FPU*	Two FP-32 FPU*	One FP-64 FPU*	Two 32-bit ALU+FPU*	One 64-bit ALU+FPU
	#PEs	336	336	336	288	288
	Fmax [MHz]	260	245	230	275	270
2D Laplace	Perf. [GFLOPS]	610.22	287.5	134.95	276.61	135.79
2D Laplace	P.Eff [GFLOPS/W]	18.28	9	4.34	8.77	4.2
2D Laplace	EPR[%]	87.31	87.31	87.31	87.31	87.31
2D Jacobi	Perf. [GFLOPS]	627.91	295.84	138.86	284.62	139.72
2D Jacobi	P.Eff [GFLOPS/W]	18.84	9.37	4.46	8.82	4.33
2D Jacobi	EPR[%]	89.84	89.84	89.84	89.84	89.84

*FPU is automatically generated through a custom Chisel-based design.

Conclusion

Packed SIMD Vectorization of the DRAGON2-CB

- Simple approach for Packed SIMD : Overlay architecture and most modules unchanged except for the Dual Compute Slot module.
- Reduced precisions achieve the best clock speed outcomes, performance and energy efficiency.
- Good Scalability is maintained across all kinds of precisions.
- EPR is independent of the used precision and depends only on architectural aspects and software implementation of the benchmarks.

Future work

Packed SIMD on DRAGON2-CB achieves better performance and better energy efficiency by reducing precision which is enough for example for ML (Machine Learning) applications that will be investigated in future work.