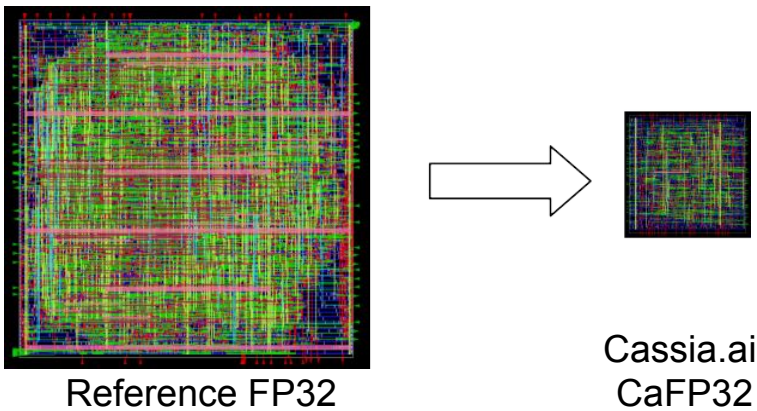
Better Power, Performance and Area with Approximate Arithmetic for ML

Approximate arithmetic from Cassia.ai increases performance and reduces power and area with negligible loss of accuracy

Cassia.ai synthesizable IP provides substantial improvement in AI/ML Power, Performance and Area (PPA) with <<1% loss in model-level accuracy; it works with any process node and is compatible with most computational simplifications used in AI

Cassia.ai functions include Multiply, Divide, Log, Exponent, Activation (GeLu, Sigmoid)

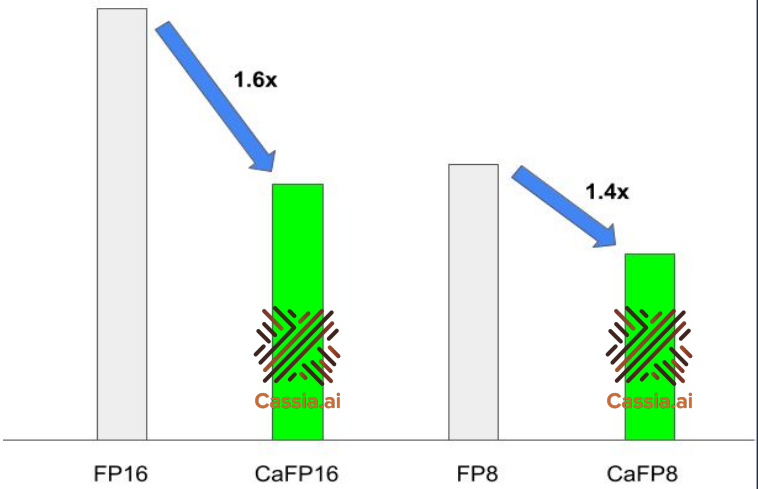
Cassia.ai IP supports many quantization formats including FP8, BF16, FP16 and more



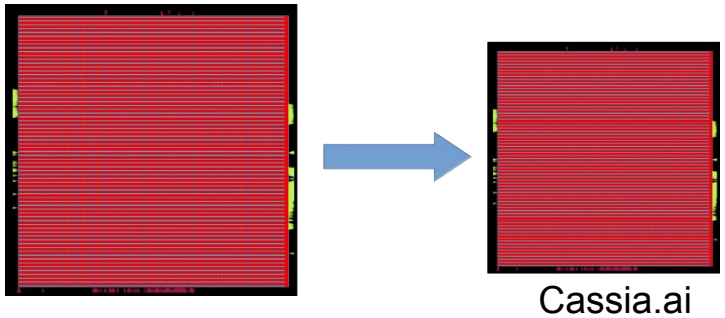
Reference FP32

Cassia.ai CaFP32

Layout comparison of standard FP32 multiplier with Cassia.ai CaFP32 multiplier. We realize 6.85x improvement in TOPs/W and 18.6x improvement in TOPs/um2.



Size comparison of an 8x8 systolic array synthesized to different multiplier data types.



Reference FP16

Cassia.ai CaFP16

Fully placed 8x8 systolic arrays

Please visit www.cassia.ai or email info@cassia.ai to learn more

Cassia.ai

Better Power, Performance and Area with Approximate Arithmetic for ML

	Reference FP8	Cassia.ai CaFP8	Cassia.ai Improvement	Reference FP16	Cassia.ai CaFP16	Cassia.ai Improvement
Latency (ps)	Information available under NDA		1.2x	Information available under NDA		1.2x
Power (W)			2.2x			2.1x
Area (um2)			1.2x			1.5x
TOPs/W			2.6x			2.5x

Comparison of latency, power, area, and efficiency for FP16 vs CaFP16 placed layouts

Model	Dataset	Float32 Accuracy	Cassia.ai CaFloat32 Accuracy
FC+Sigmoid	MNIST	87.50%	87.40%
LeNet	MNIST	98.60%	98.50%
MobileNet V1	MNIST	98.46%	98.19%
ResNet18	ImageNet	69.76% / 89.08%	69.22% / 88.79%
ResNet50	ImageNet	76.13% / 92.86%	75.22% / 92.56%

Popular models and their errors with Cassia.ai technology
Accuracy can be recovered by minor re-training with Cassia.ai arithmetic

Keep your Architecture with Cassia.ai

With Cassia.ai approximate arithmetic, your architecture does not change and you keep the standard integer and floating-point data formats. The only changes are to the core arithmetic operations: **multiplication**, division, exponentiation, and logarithms
Some key benefits:

- No need to transform the memory layout of your tensors
- No need to increase the number of scatter-gather operations to and from memory
- No special preparation of data either on-chip or off-chip
- No constraints on tensor sizing or formatting (e.g. dimensions or channels)
- No special DMA requirements

Please visit www.cassia.ai or email info@cassia.ai to learn more