



Design-Space-Exploration of 22nm FD-SOI SoC for Convolutional Neural Network Computation

Tensilica Day 2018

Nicolai Behmann, M.Sc.

Prof. Dr.-Ing. Holger Blume

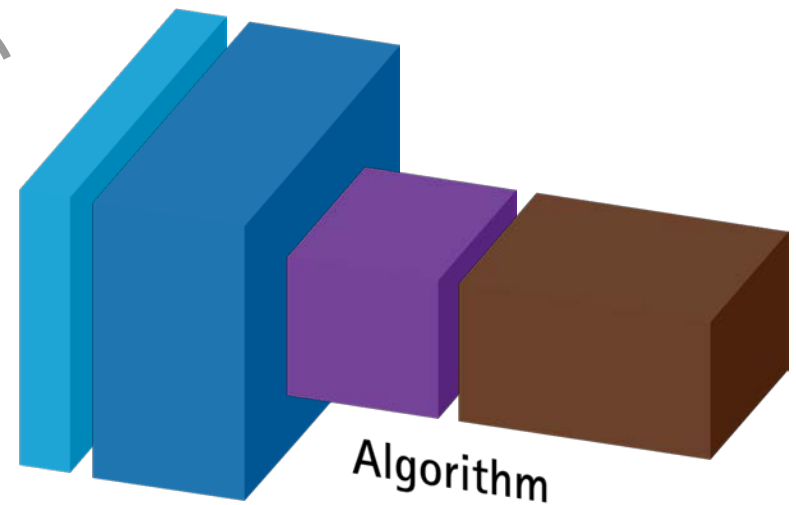
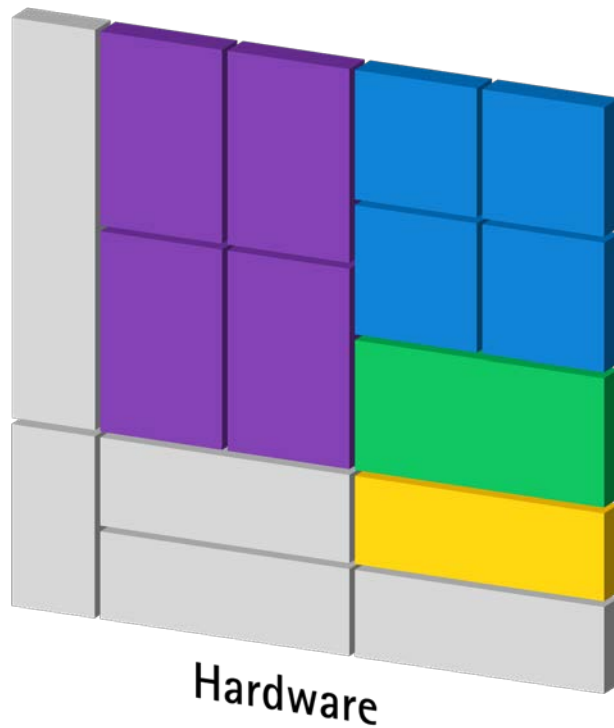


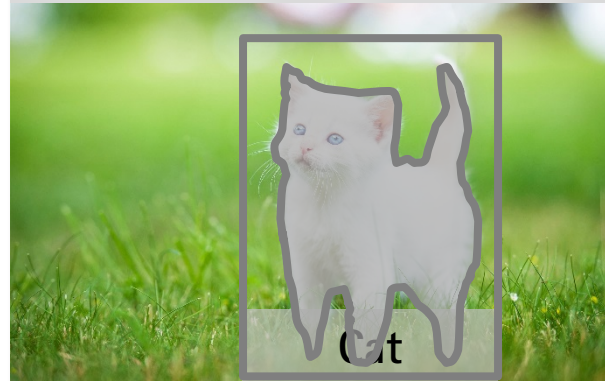
Image Classification



Cat

A. Krizhevsky et. al., *ImageNet Classification with Deep Convolutional Neural Networks*

Instance Segmentation



Cat

K. He et. al., *Mask R-CNN*

Convolutional
Neural Networks

Object Detection



Cat

J. Redmon et. al., *You Only Look Once: Unified, Real-Time Object Detection*

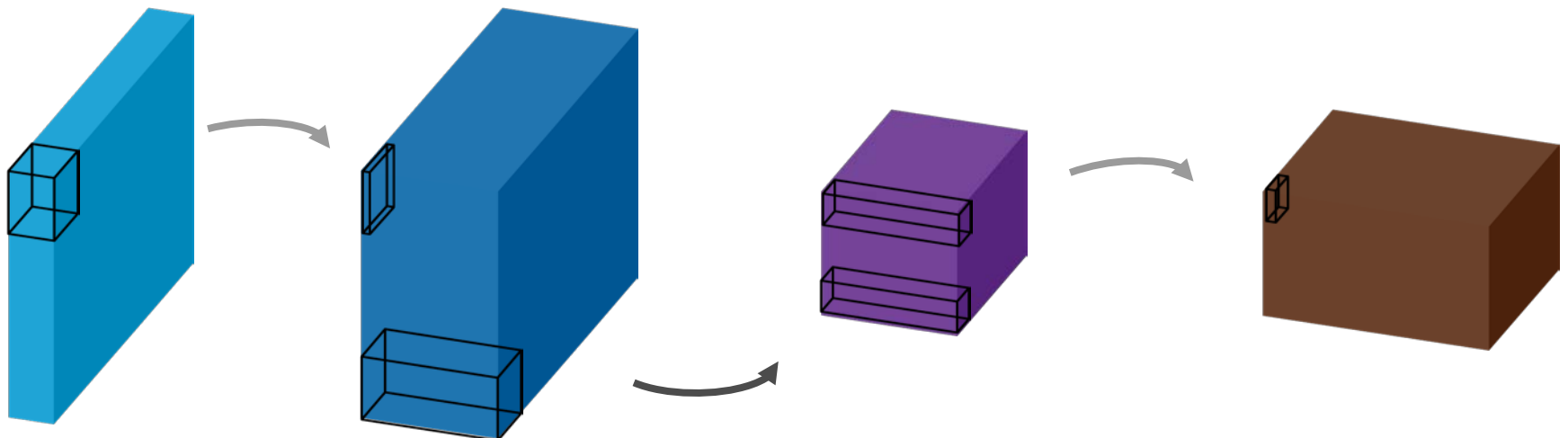
Local Matching




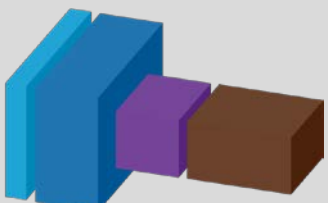

W. Luo et. al., *Efficient deep learning for stereo matching.*

Convolutional Neural Network

- Similar CNN architecture with common layers for all tasks
- Convolutional Layer: local feature extraction (usually approx 90% of computation time)
 - 3D-Convolution of input feature map volume with pretrained filter kernels
 - Subsequent pixel-wise transformation with non-linear activation function
- Max Pooling Layer: non-linear information reduction
 - Maximum/average value from local neighborhood



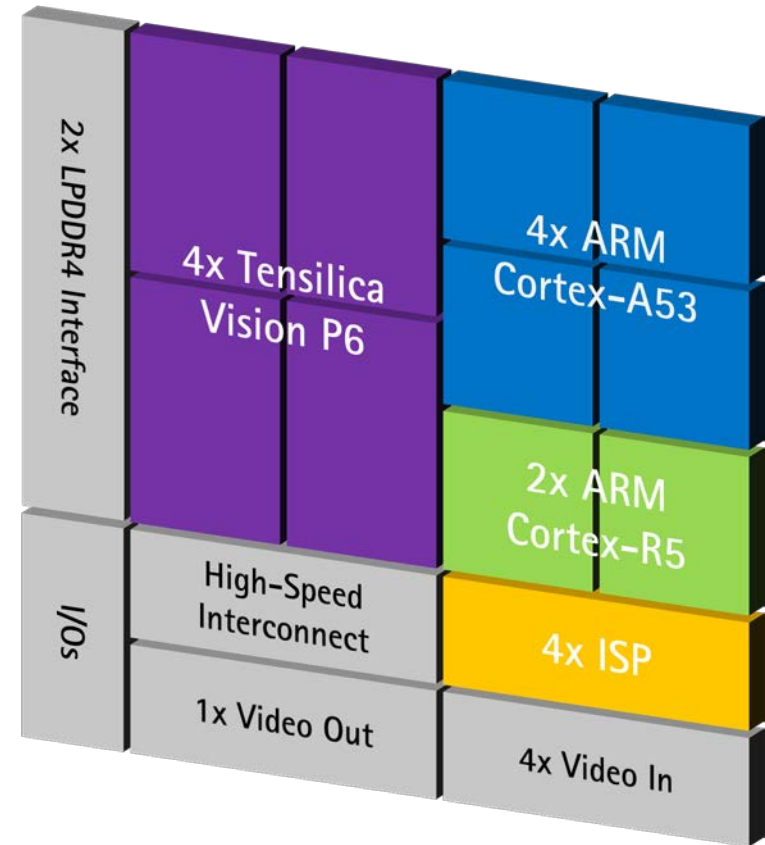
Challenges for Embedded Convolutional Neural Networks

Hardware	CNN-Architecture	Requirements
		
<ul style="list-style-type: none"> • Limited computational capacity • Limited memory bandwidth • Expensive chip area 	<ul style="list-style-type: none"> • Tradeoff between detection accuracy and execution time • Programming flexibility for updates or future extensions 	<ul style="list-style-type: none"> • Real-Time requirements • Accuracy requirements • Power requirements
<p style="text-align: center;">✓</p>	<p style="text-align: center;">✓</p>	<p style="text-align: center;">Estimate performance, accuracy, power</p>
<p style="text-align: center;">Choose Hardware according to Algorithm & Requirements</p>	<p style="text-align: center;">✓</p>	<p style="text-align: center;">✓</p>
<p style="text-align: center;">✓</p>	<p style="text-align: center;">Design CNN-Architecture to requirements and hardware</p>	<p style="text-align: center;">✓</p>

Design Space Exploration

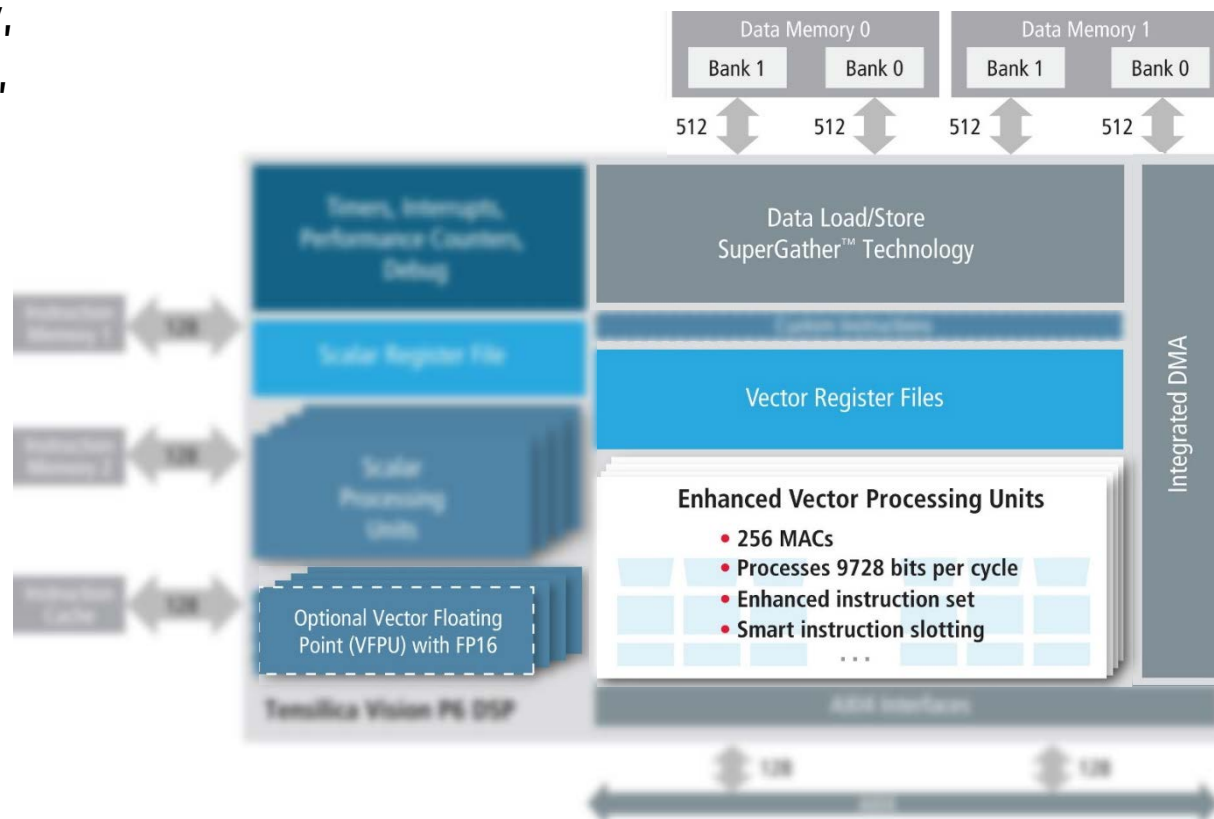
Dream Chip *Software Defined Image Processing (SDIP)* SoC

- 4x *ARM Cortex-A53*
 - ARM-v8A general purpose processor
 - 128-bit SIMD (NEON)
- 4x *Tensilica Vision P6*
 - 32-bit RISC based
 - 512-bit SIMD accelerator
- *ARM Cortex-R5* Lock Step (ADAS safety)
- Rich video support
 - 4x video in, 1x video out
 - Integrated ISP pipeline
- 2x LPDDR4 memory
- 22nm FD-SOI Globalfoundries process
 - comparable backend optimization for each IP



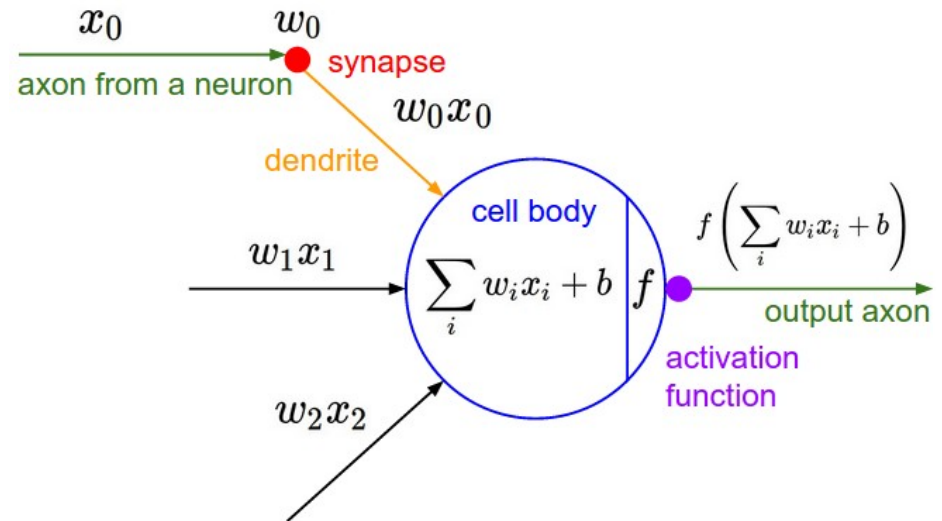
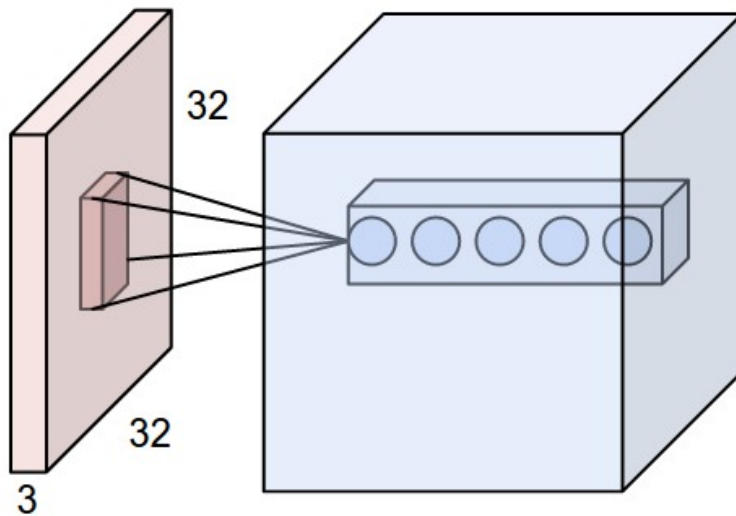
Tensilica Vision P6 architecture

- 1000/1500/640/512 pixels
- 60/90/320/640 memory (60/90/320/640) (60/90/320/640)
- 16x320 transfers / cycle
- 8x320 parallelization, 4-5 issues slot VLIW
- 90/16/2x250KB system memory, local system memory, local system memory
- 2 loads or 1 load + 1 store per cycle



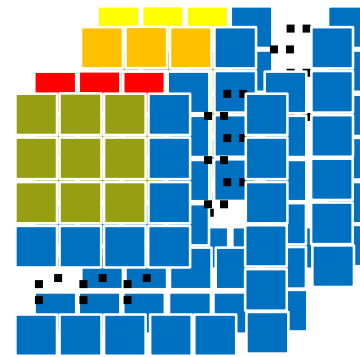
Convolution Layer Implementation

- 3D kernel slices through 3D tile window of input volume
- each output neuron represents the sum of weighted input volume (+ bias)
- Non-linear activation function following accumulation, e.g. rectified linear unit (ReLU)

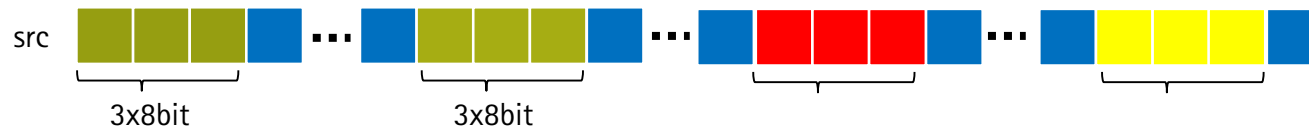


Conv Vectorization / Data Organisation

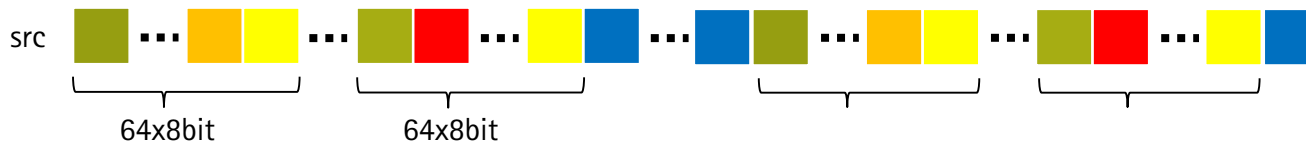
- Src: uint8 - 55x55x64, Kernel: 3x3



- width-height-depth (WHD)



- depth-width-height (DWH)

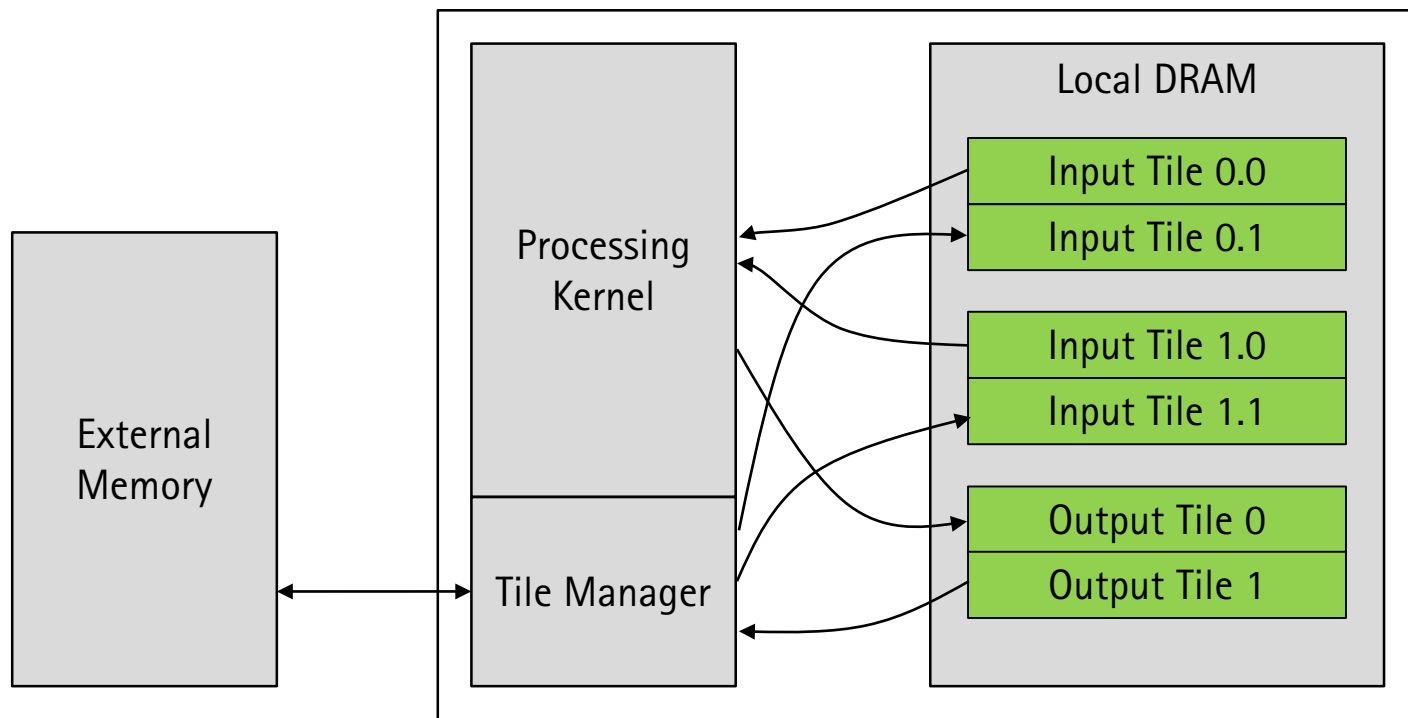


- Vectorization accross depth (usually multiple of 32 layers)

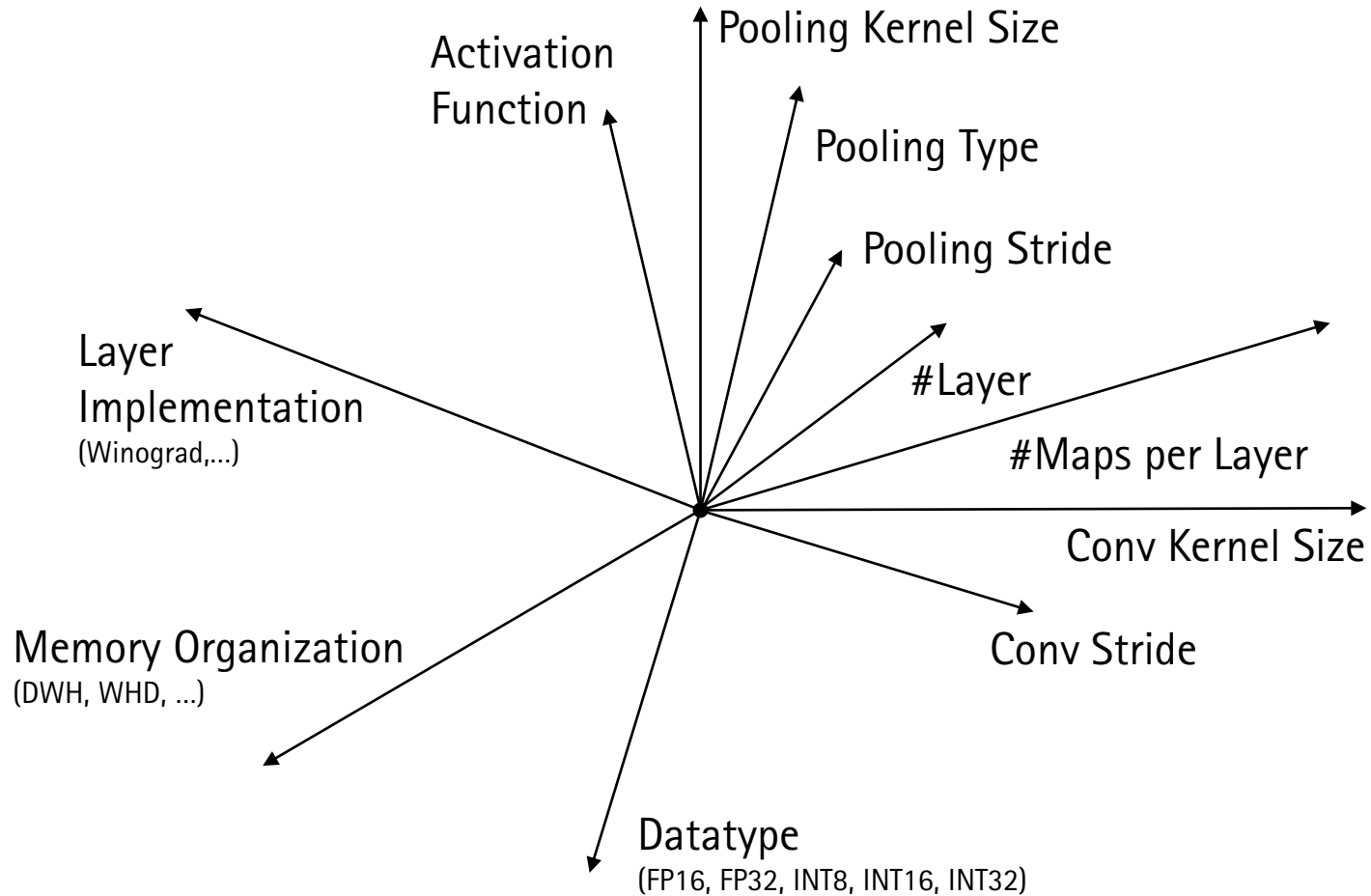


Tensilica Vision – Tile Manager

- Local DRAM holds image / feature map & weight tiles
- DSP runs processing Kernels on 3D-Tiles
- Double Buffer is used to allow processor and DMA work in parallel
- Tile Manager keeps track of buffers in external und local DRAM and schedules DMA

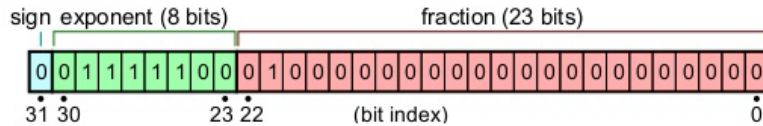


Design Space for Convolutional Neural Networks



Design Space Modeling

Floating vs. Fixed Point Datatype



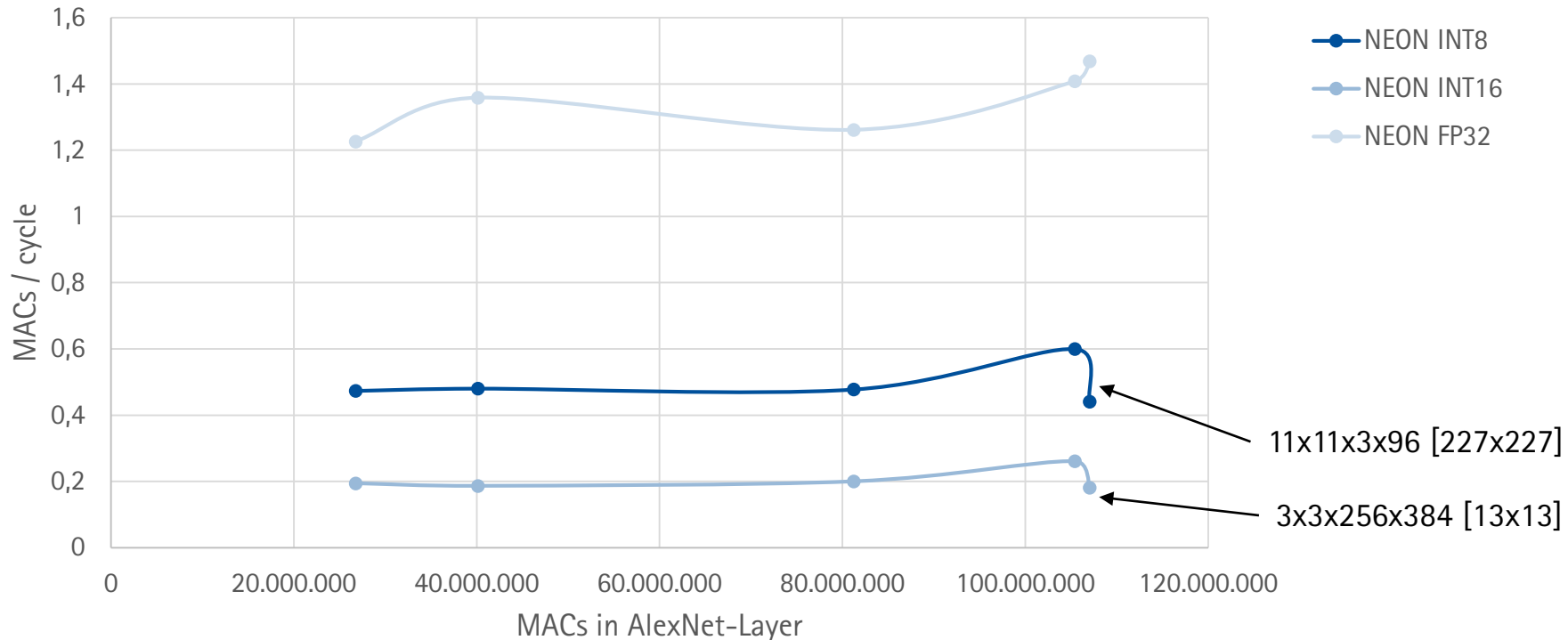
Floating Point representation

- 16, 32 or 64 bit standardized
- dynamic number range:
 - Small accuracy (big number)
 - High accuracy (small numbers)
- Needs additional FPU
- MAC in one cycle

Fixed Point representation

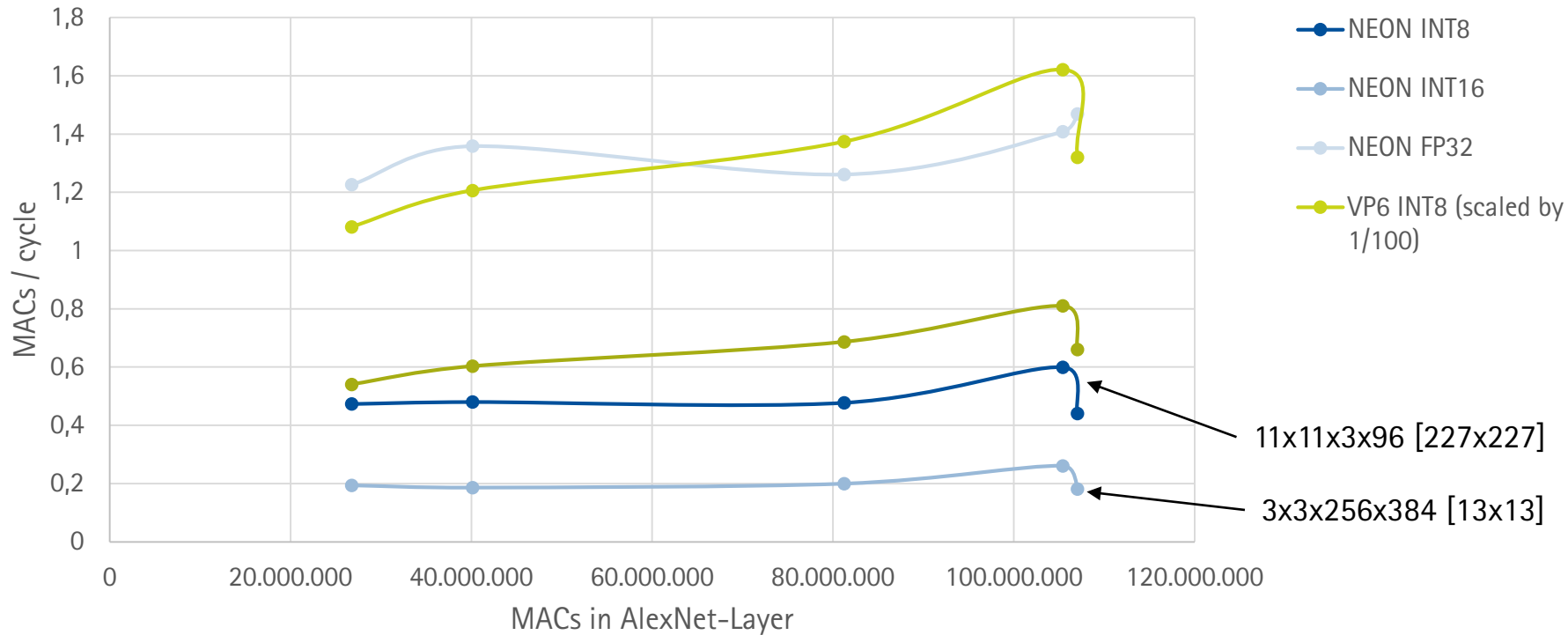
- 8, 16, 32 or 64 bit usually available
- Fixed, limited range (eventually overflows):
 - Small accuracy (high dynamic range)
 - High accuracy (small dynamic range)
- No additional HW necessary
- MAC in two cycles (MAC + shift back)
- Reduced memory bandwidth

Design Space Exploration: Datatype Dependency (NEON)



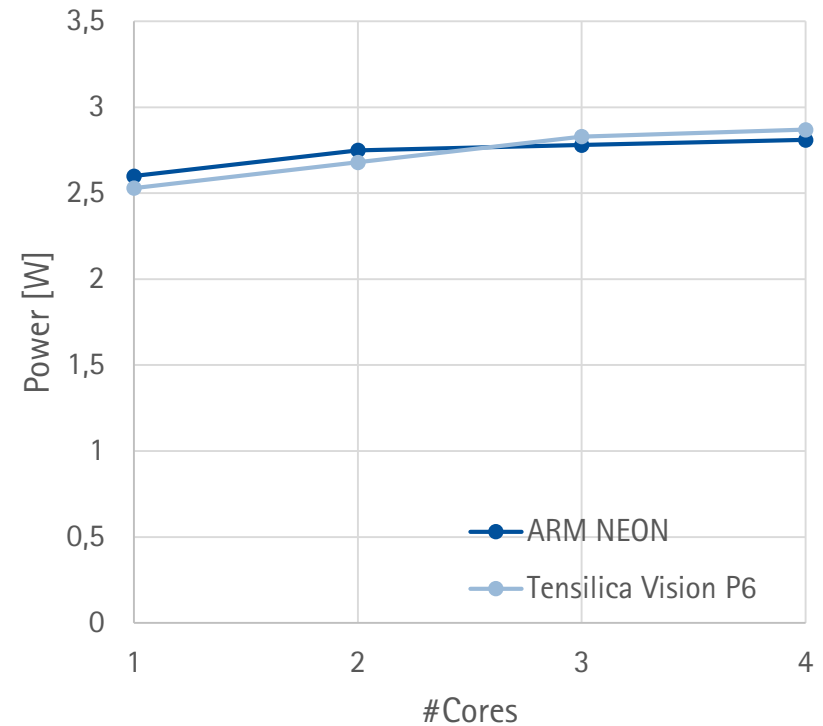
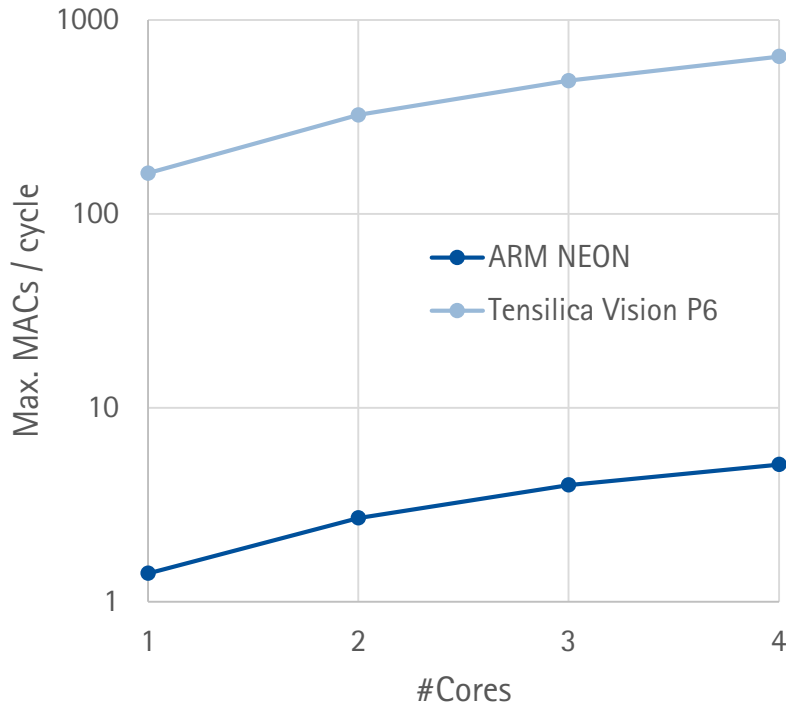
- Floating point superior in performance on ARM-NEON: 6x more MACs / cycle (vs INT16)
- Theoretic NEON MACs/cycle performance: 128-bit SIMD engine
 - 4 MACs / cycle (FP32),
 - 4 MACs / cycle (INT16), 8 MACs / cycle (INT8), due to additional shift
- ARM Compute Library, Tensilica XiLibCNN, ARM Cortex-A53 @ 500 MHz

Design Space Exploration: Datatype Dependency



- Tensilica Vision P6 benefits from smaller datatypes: reduced memory bandwidth
- VLIW shadows scalar, shifting and write back instructions
- Software cache (local memory) highly useful for CNN acceleration (fixed access pattern)
- Theoretic maximum of 256 MACs / cycle (8-bit)
- ARM Compute Library, Tensilica XiLibCNN, ARM Cortex-A53 @ 500 MHz

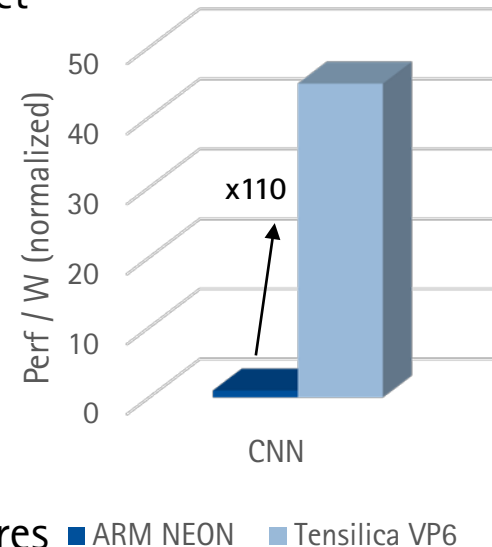
Design Space Exploration: 22nm SoC



- Tensilica Vision P6 offers ~115x more MACs per Cycle
- MAC utilization of Tensilica Vision P6 approx. 50 %, 35 % for NEON
- Power Consumption of Vision P6 and NEON engine comparable (heavy load)

Conclusion – Design–Space–Exploration of 22nm FD–SOI SoC for Convolutonal Neural Network Computation

- Models from Design Space Exploration accelerate product development
 - Estimate performance for HW & algorithm,
 - Choose HW configuration based on requirements,
 - Tune algorithm to fit in HW, given real time req.
- Comparison & evaluation of NEON and VP6 SIMD
 - VP6 offers 115x more MACs per cycle
 - Power efficiency of VP6 is 110x higher than NEON
 - Different datatypes suitable for different architectures
- Future Work: Fill Design Space Models
 - Increase clock frequency, forward body biasing
 - More CNN Layer configurations



Tensilica Vision P6 ideally suitable for CNN acceleration, as well as Computer Vision

No free lunch

