Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Low Complexity Multiply-Accumulate Units for Convolutional Neural Networks with Weight-Sharing

PASM

James Garland, David Gregg

SFI Project 12/IA/1381

Date 23 Jan 2019

Research Challenge

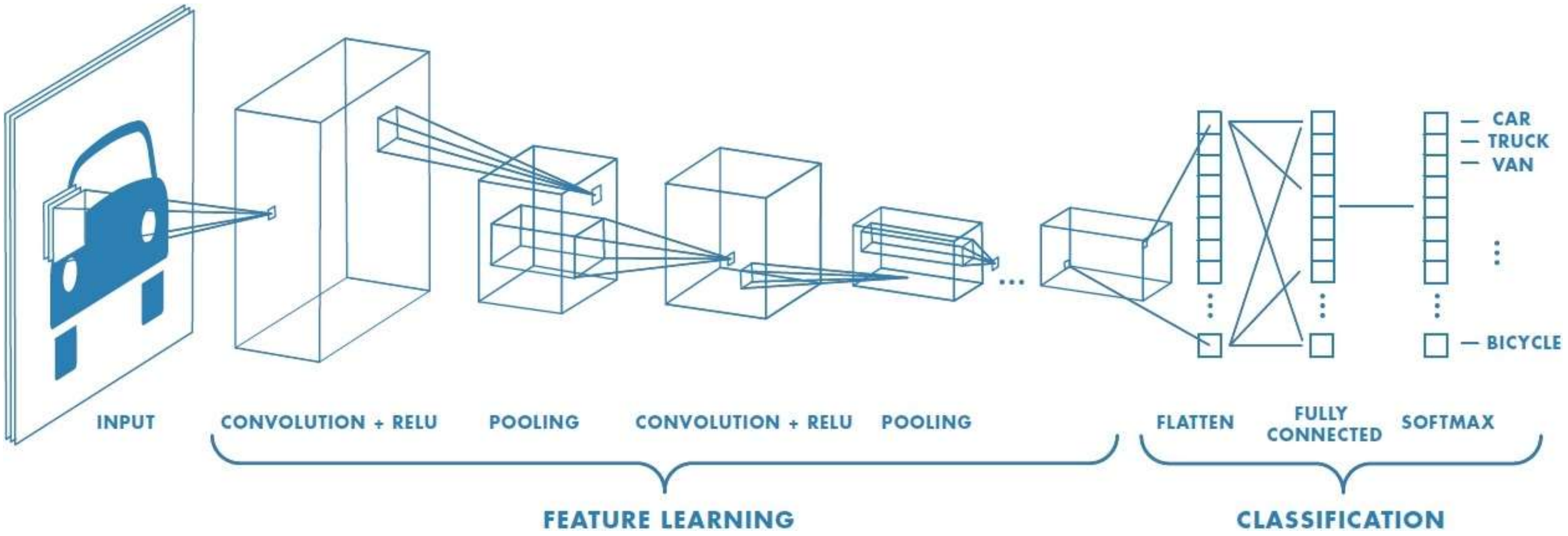
"By the year 2600, the world's population would be standing shoulder to shoulder, and the electricity consumption would make the Earth glow red-hot." [1]

- We need to start now to prevent a toasty warm environment!

- Artificial Intelligence (AI) & machine learning (ML) getting more ubiquitous.
- They consume more and more power in data centres.
- How can we stop this increasing power consumption trend whilst getting ML into off-line embedded devices?

[1] Hawking, Tencent WE Summit, 2017.

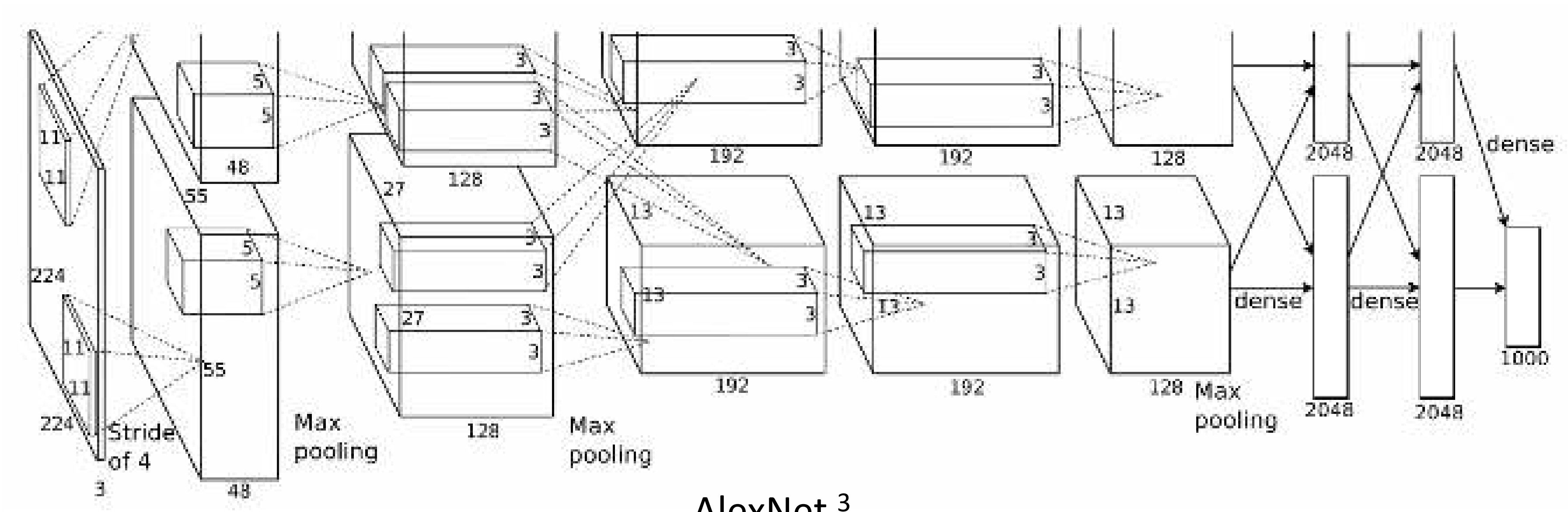
Quick Intro to CNNs



Convolutional neural network (CNN) architecture [2]

[2] mathworks.com

The One That Started It All! (AlexNet)

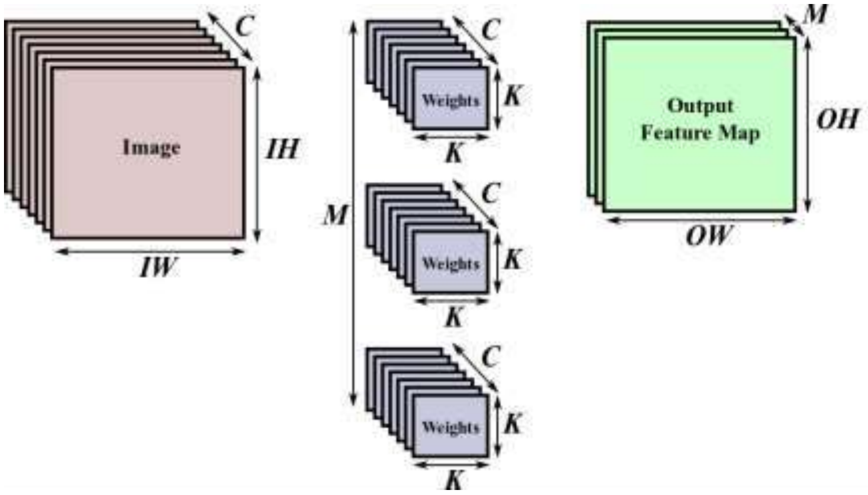


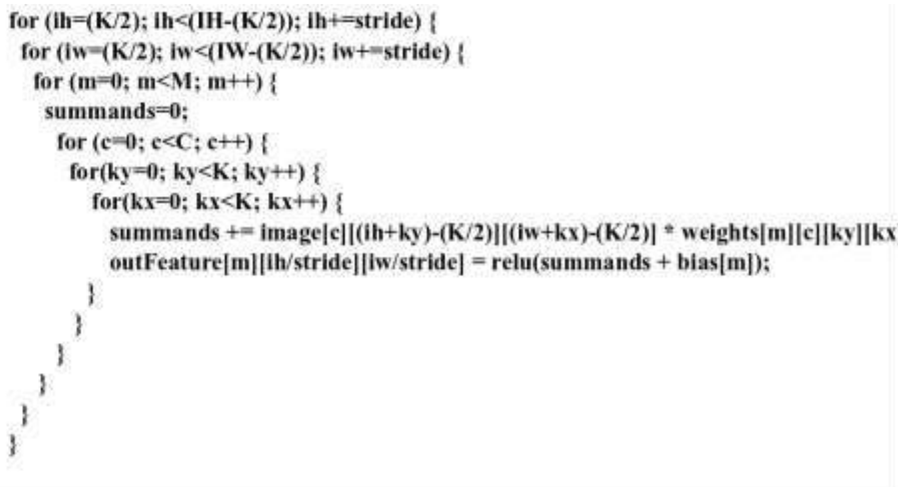
AlexNet [3]

- **CNNs have 100,000's or more multiply-accumulates, e.g. AlexNet [3]**
  - However, LeNet was the pioneer for OCR [4]
- **90% of time in computation is spent in the convolution layer [5]**

[3] Krizhevsky et al. 2012.
[4] LeCun et al. 1998.
[5] Farabet et al. 2010.

Convolution Layer





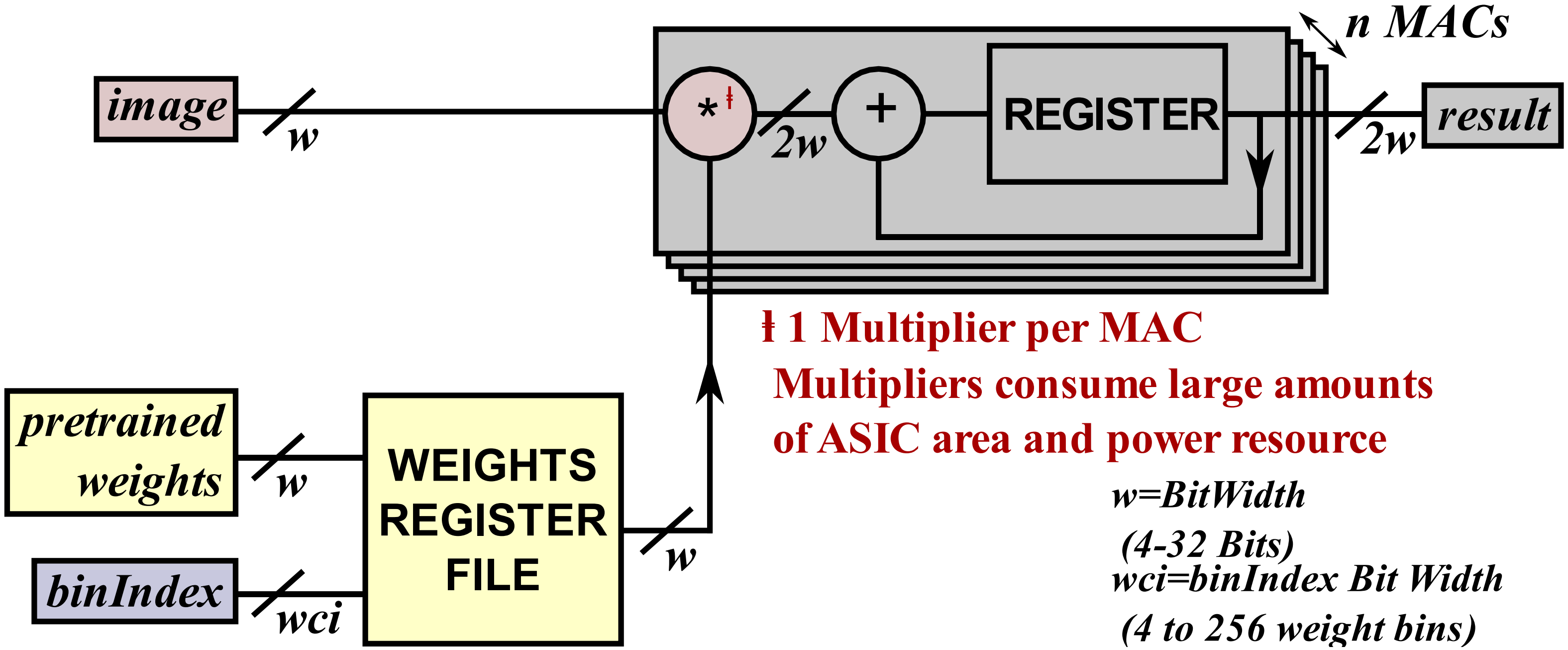
```
for (ih=(K/2); ih<(IH-(K/2)); ih+=stride) {
 for (iw=(K/2); iw<(IW-(K/2)); iw+=stride) {
  for (m=0; m<M; m++) {
   summands=0;
    for (c=0; c<C; c++) {
     for(ky=0; ky<K; ky++) {
       for(kx=0; kx<K; kx++) {
         summands += image[c][(ih+ky)-(K/2)][(iw+kx)-(K/2)] * weights[m][c][ky][kx]
         outFeature[m][ih/stride][iw/stride] = relu(summands + bias[m]);
       }
      }
     }
    }
  }
}
```

- To reduce computation time, systolic array loops are unrolled
- CNN Challenges:
  - A lot of data movement required due to megabytes of weight data
  - Hardware convolution accelerators could have as many multipliers as multiply-accumulate (MAC) operations
  - Hardware multipliers are large and power hungry. [6]

[6] Sabeetha et al. 2015.

Weight Shared CNN Accelerator

- Reduce the weight data movement
- Pre-trained weights pruned and quantised to 16-256 shared values [7].
- Pre-trained **weight** values are stored in a weights register file.
- Values indexed, retrieved, multiplied by corresponding **image** value.



ǂ 1 Multiplier per MAC
Multipliers consume large amounts of ASIC area and power resource

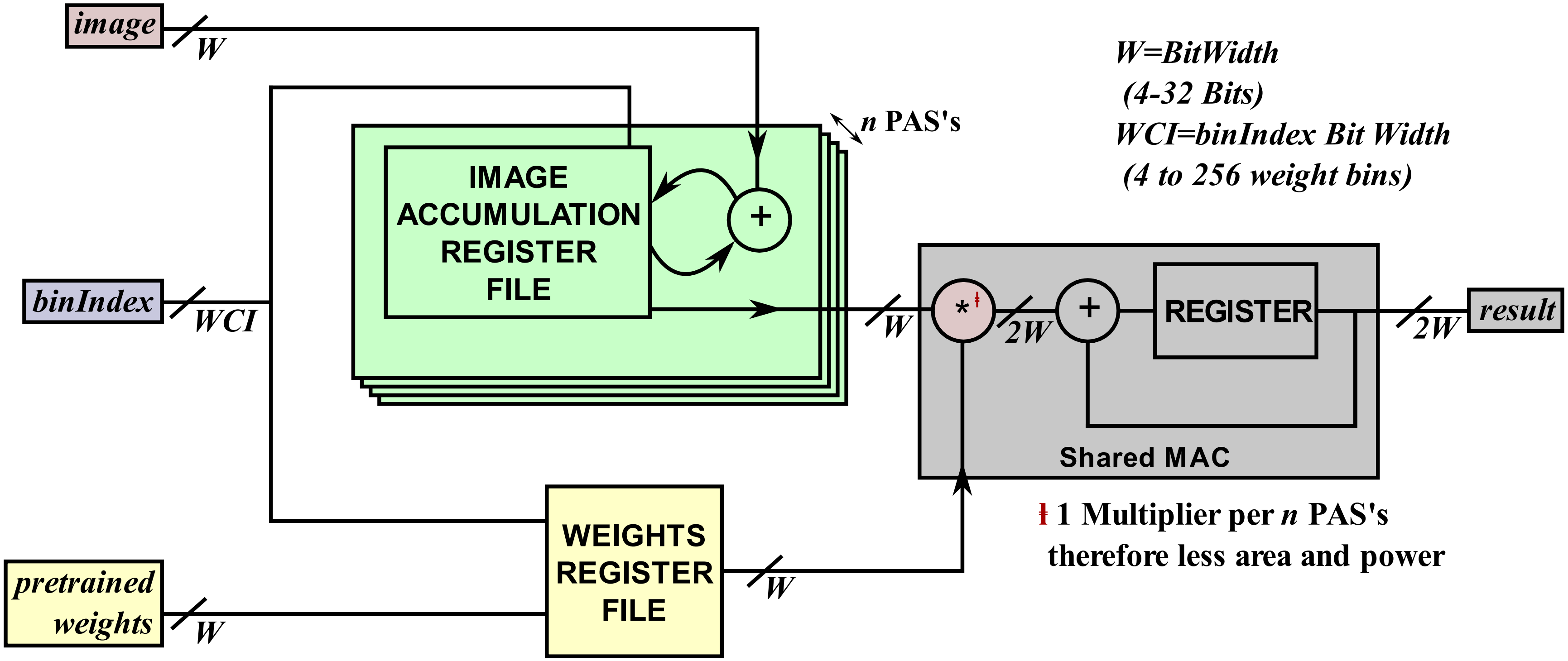
w=BitWidth (4-32 Bits)
wci=binIndex Bit Width (4 to 256 weight bins)

[7] Han et al. 2016; 2015.

We Propose PASM

Multiple-PAS-Shared-MAC (parallel accumulate shared MAC (PASM))

- Multiple parallel accumulate and store (PAS) units followed by **one** shared MAC.
- PASs accumulate *w* bit **image** into *b* = 2^{wci} bins register file
- Post-pass MAC multiplies weights with binned image values



W=BitWidth (4-32 Bits)
WCI=binIndex Bit Width (4 to 256 weight bins)

‡ 1 Multiplier per *n* PAS's therefore less area and power

For Reference: A Weight-Shared MAC

binIndex	0	1	2	3	0
image	26.7	3.4	4.8	17.7	6.1

binIndex	0	1	2	3
pretrained weights	1.7	0.4	1.3	2.0

*

+=

result 98.8

For Reference: A Weight-Shared MAC

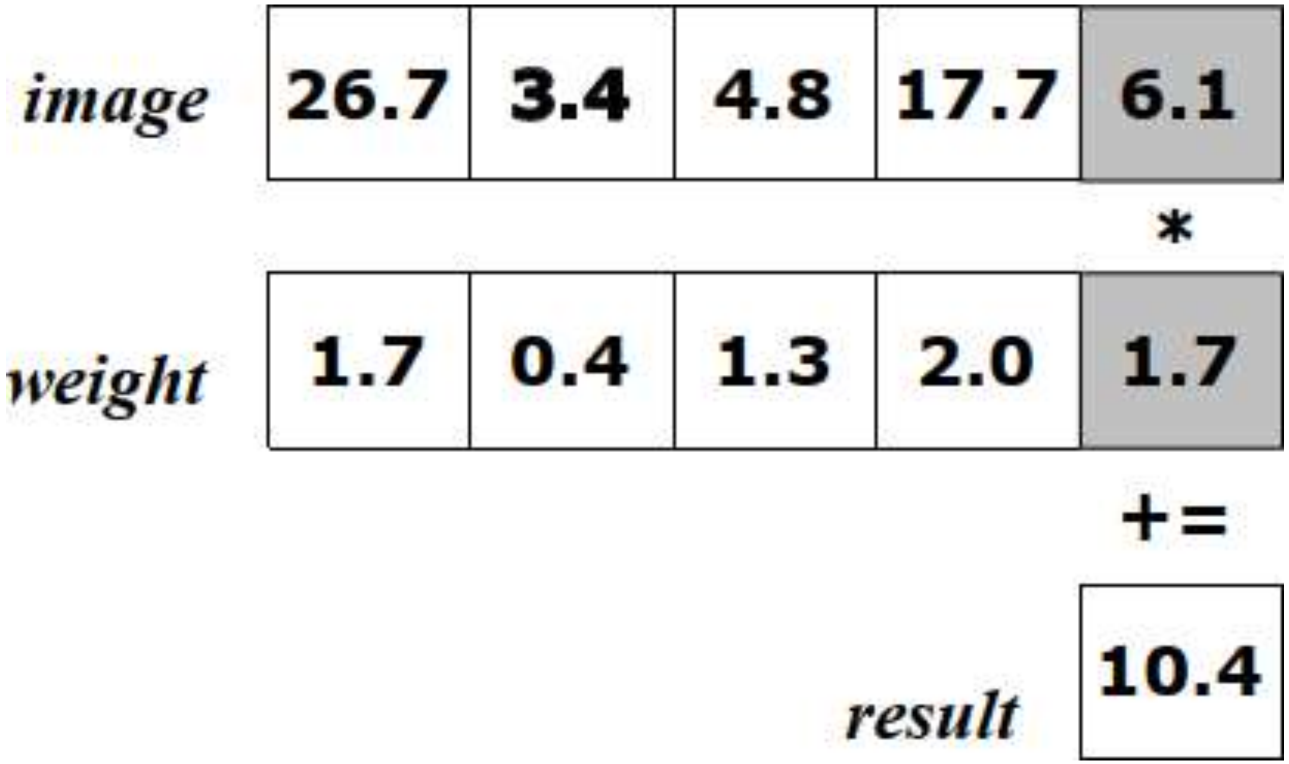


image	26.7	3.4	4.8	17.7	6.1
					*
weight	1.7	0.4	1.3	2.0	1.7
					+=
				result	10.4

For Reference: A Weight-Shared MAC

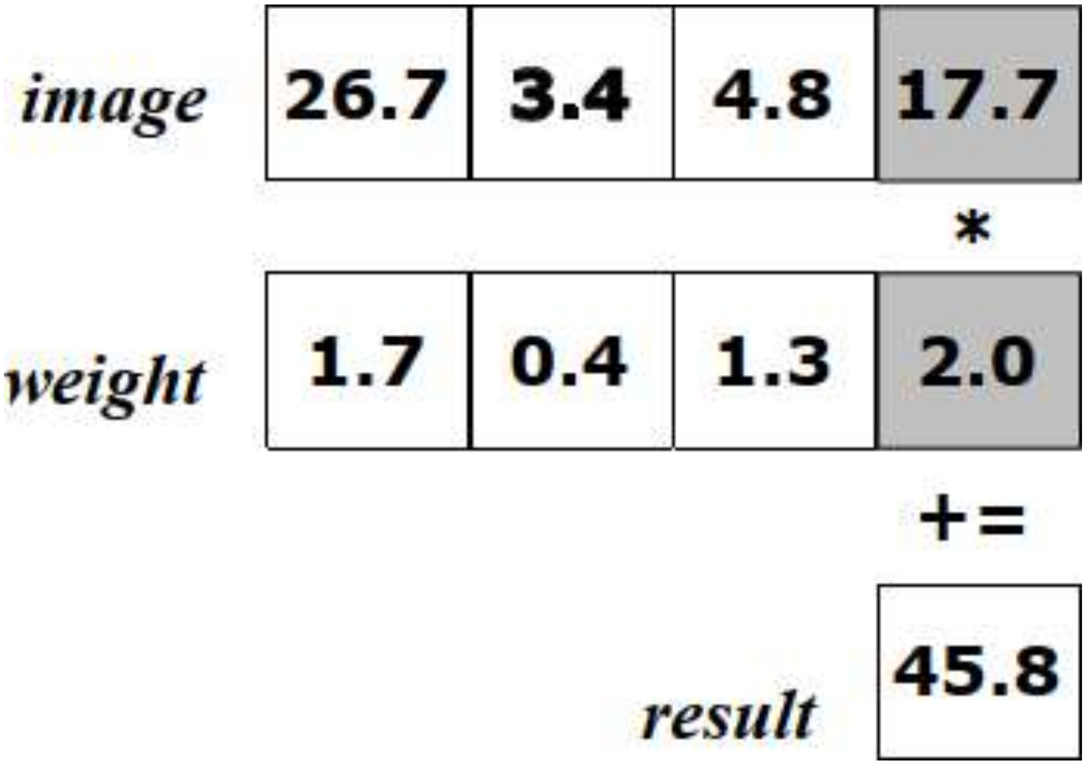


image	26.7	3.4	4.8	17.7
				*
weight	1.7	0.4	1.3	2.0
				+=
			result	45.8

For Reference: A Weight-Shared MAC

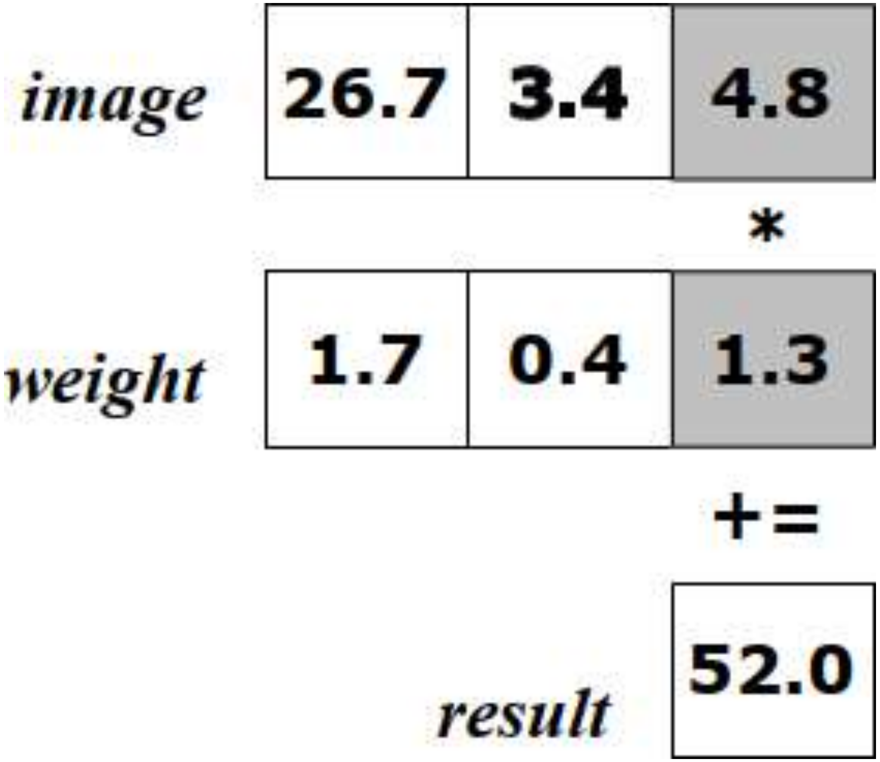
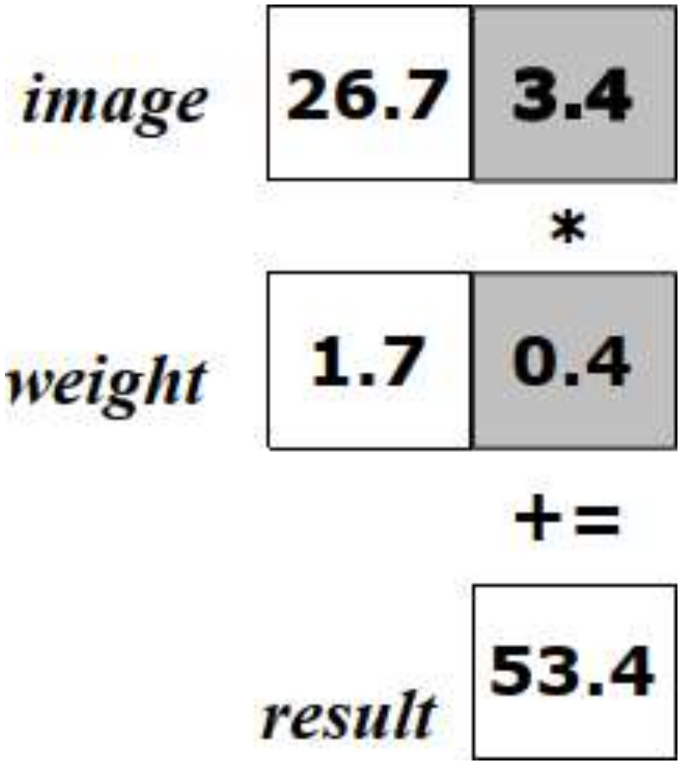


image	26.7	3.4	4.8
			*
weight	1.7	0.4	1.3
			+=
		result	52.0

For Reference: A Weight-Shared MAC



For Reference: A Weight-Shared MAC

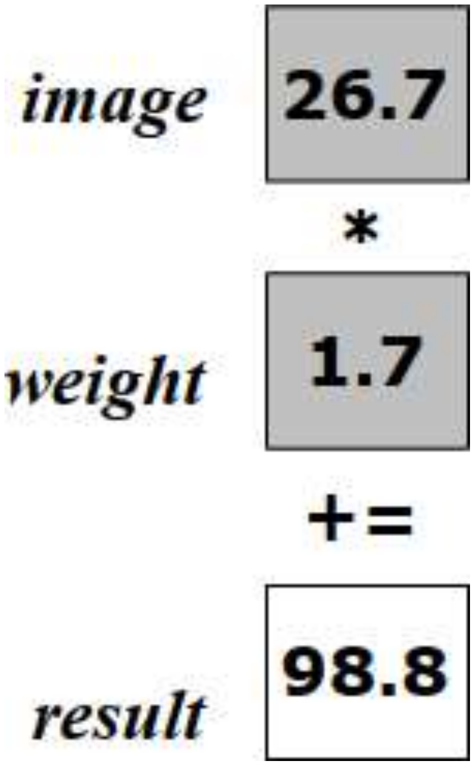


image 26.7

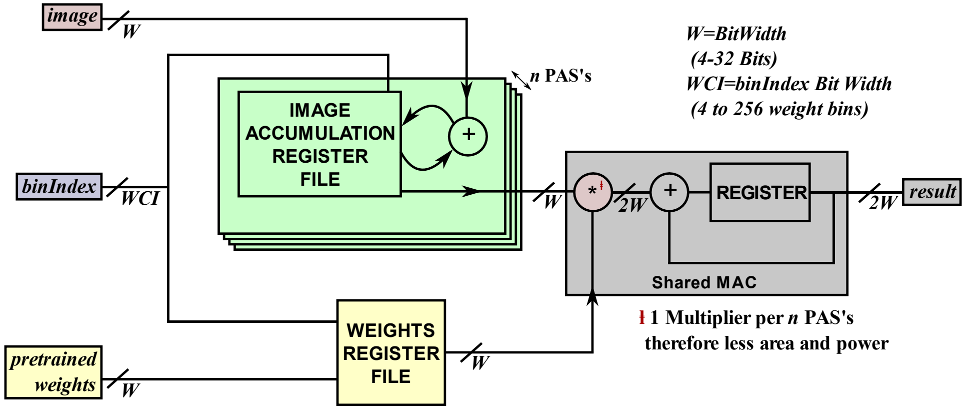
*

weight 1.7

+=

result 98.8

PASM In Operation



W=BitWidth
(4-32 Bits)
WCI=binIndex Bit Width
(4 to 256 weight bins)

† 1 Multiplier per *n* PAS's
therefore less area and power

binIndex	0	1	2	3	0
image	26.7	3.4	4.8	17.7	6.1

+=

binIndex	0	1	2	3
image bins	32.8	3.4	4.8	17.7

PAS Phase

binIndex	0	1	2	3
pretrained weights	1.7	0.4	1.3	2.0

*****

binIndex	0	1	2	3
image bins	32.8	3.4	4.8	17.7

+=

result	98.8

PASM Phase

PASM In Operation – PAS Phase

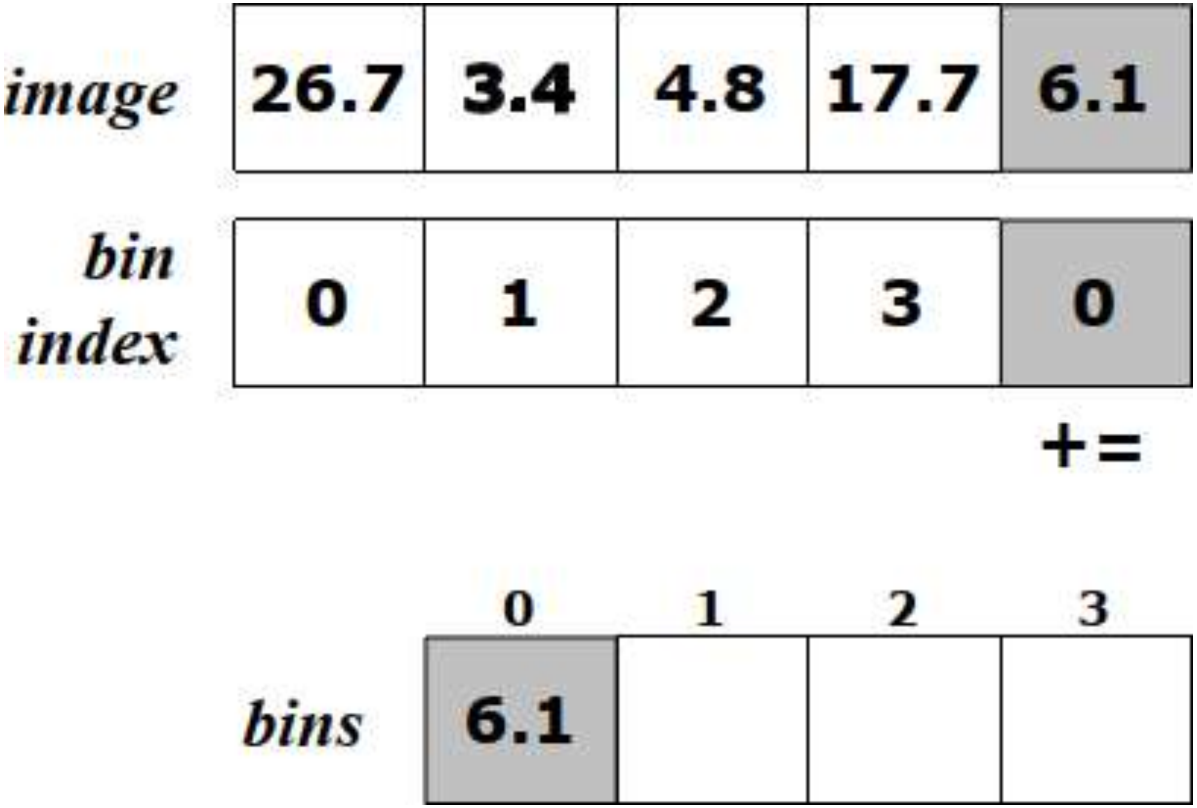


image	26.7	3.4	4.8	17.7	6.1
bin index	0	1	2	3	0

+=

	0	1	2	3
bins	6.1			

PASM In Operation – PAS Phase

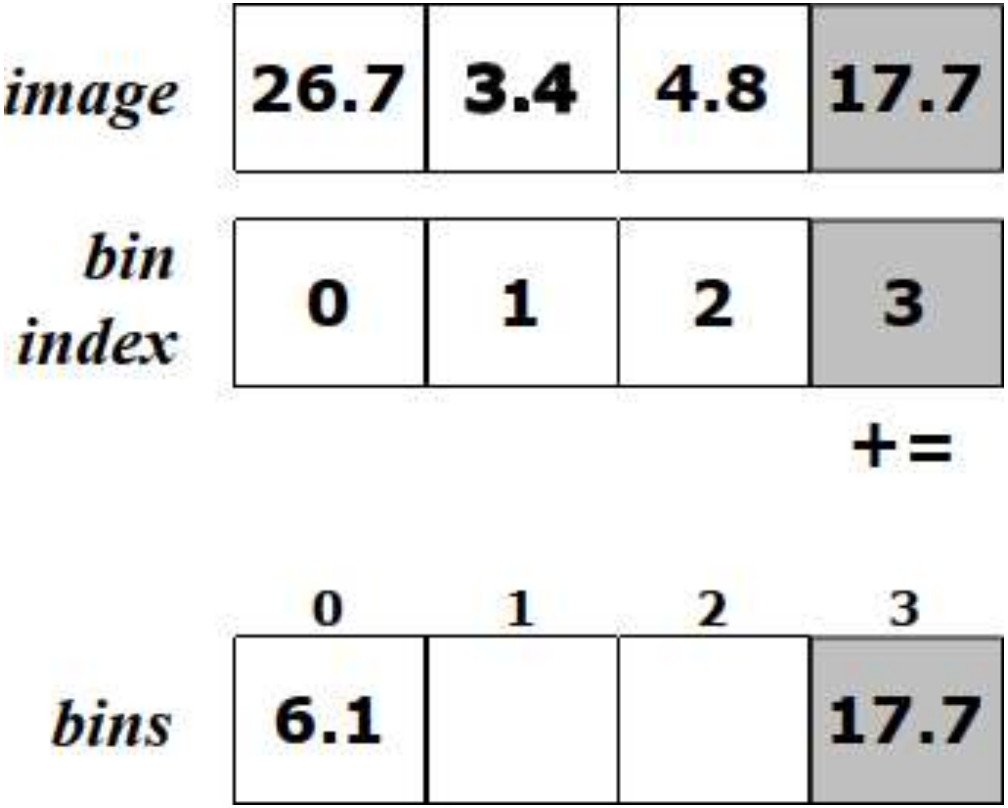


image	26.7	3.4	4.8	17.7
bin index	0	1	2	3
				+=

	0	1	2	3
bins	6.1			17.7

PASM In Operation – PAS Phase

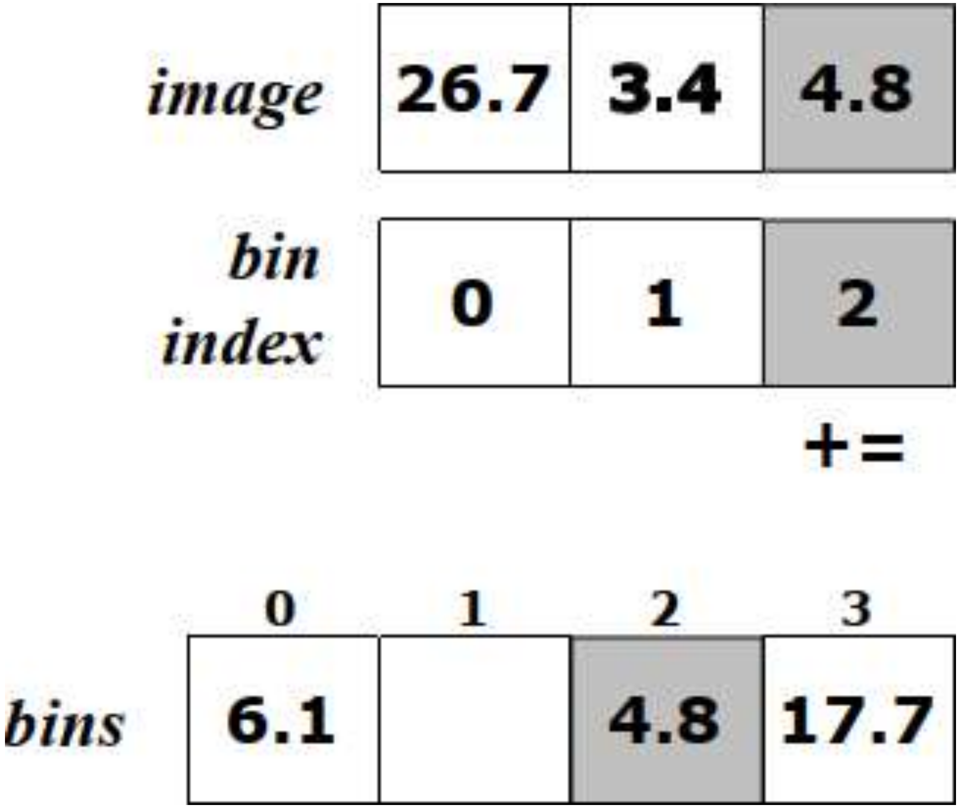


image	26.7	3.4	4.8
bin index	0	1	2

+=

	0	1	2	3
bins	6.1		4.8	17.7

PASM In Operation – PAS Phase

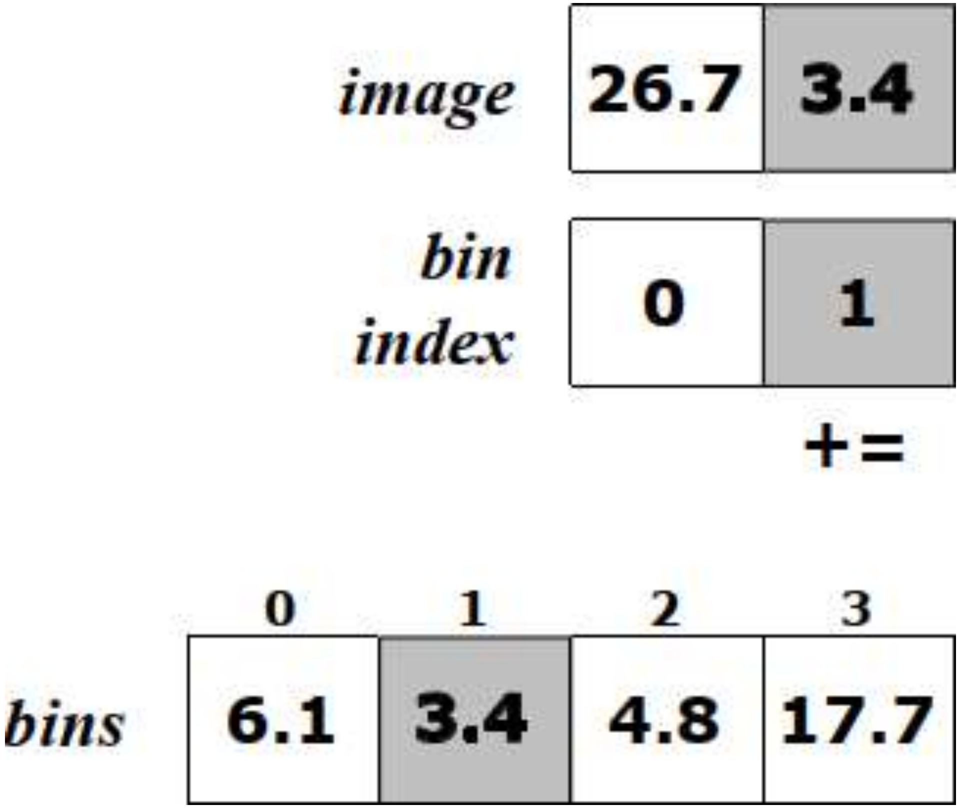


image	26.7	3.4
bin index	0	1
		+=

	0	1	2	3
bins	6.1	3.4	4.8	17.7

PASM In Operation – PAS Phase

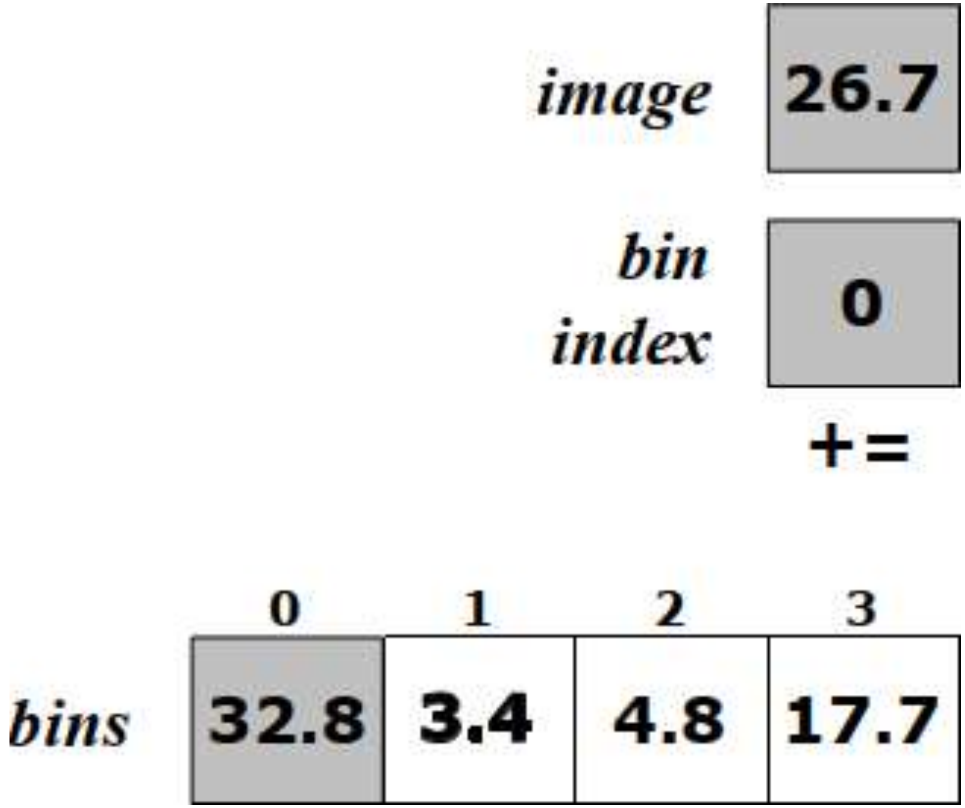


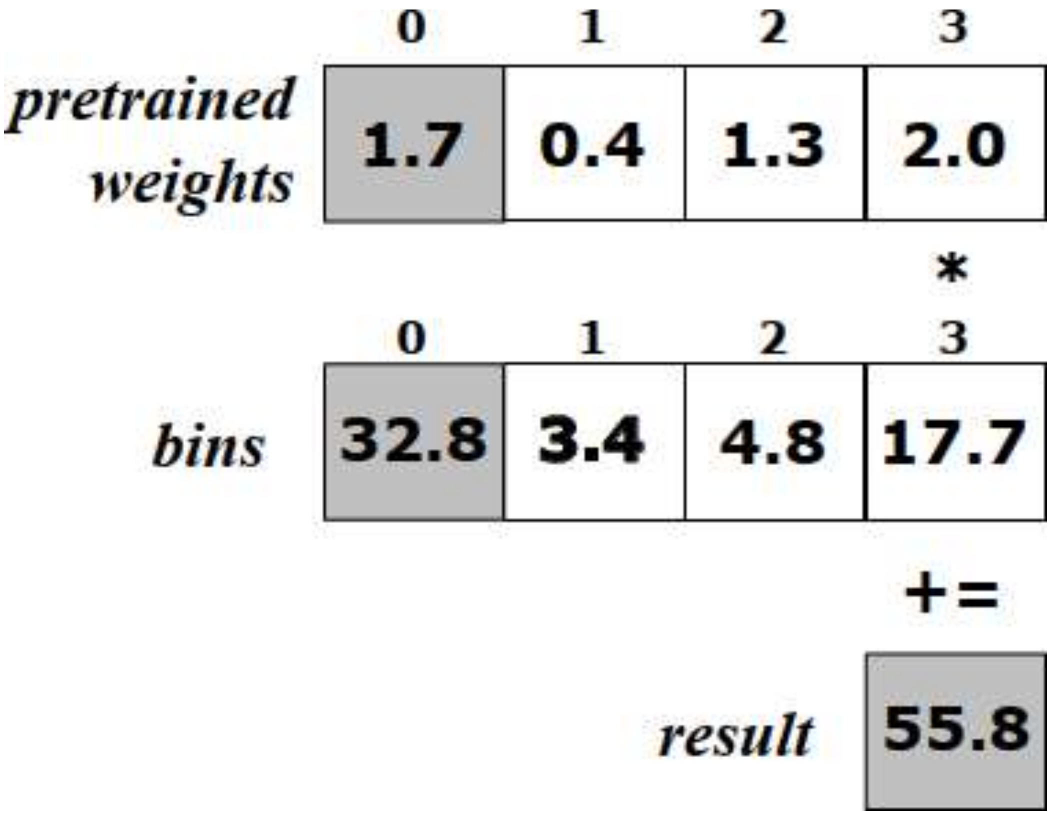
image 26.7

bin index 0

+=

	0	1	2	3
bins	32.8	3.4	4.8	17.7

PASM In Operation – PASM Phase



	0	1	2	3
pretrained weights	1.7	0.4	1.3	2.0

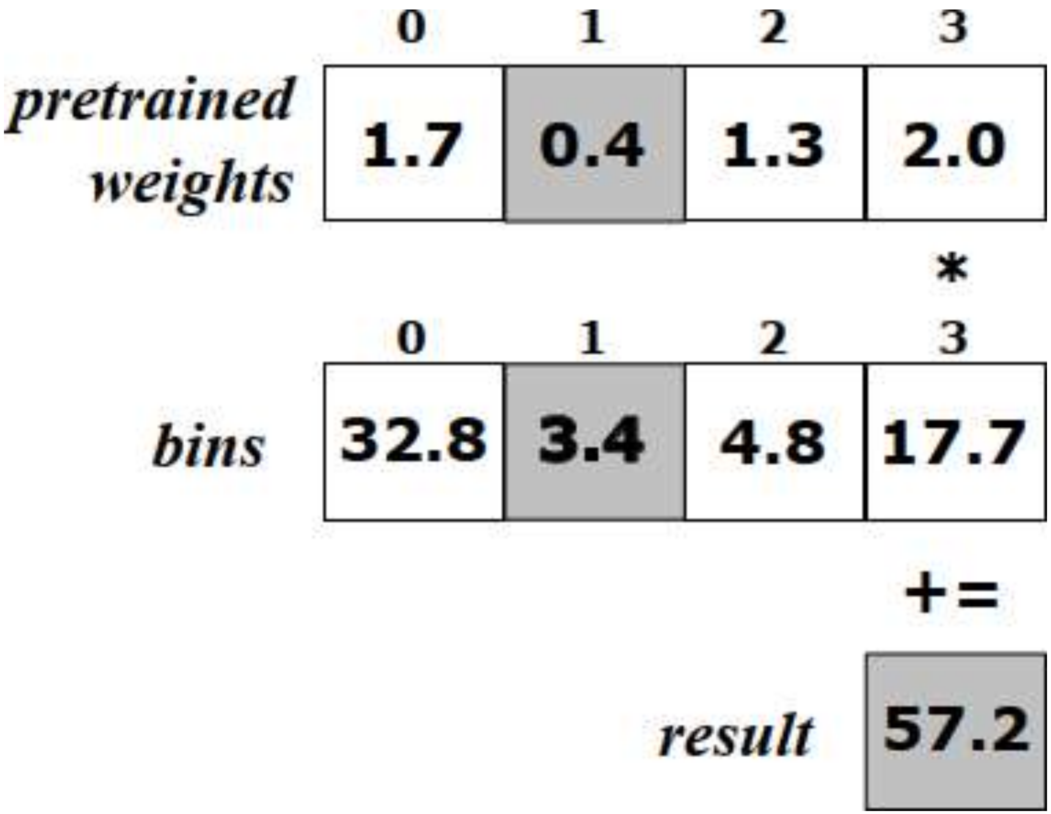
*

	0	1	2	3
bins	32.8	3.4	4.8	17.7

+=

result 55.8

PASM In Operation – PASM Phase



	0	1	2	3
pretrained weights	1.7	0.4	1.3	2.0

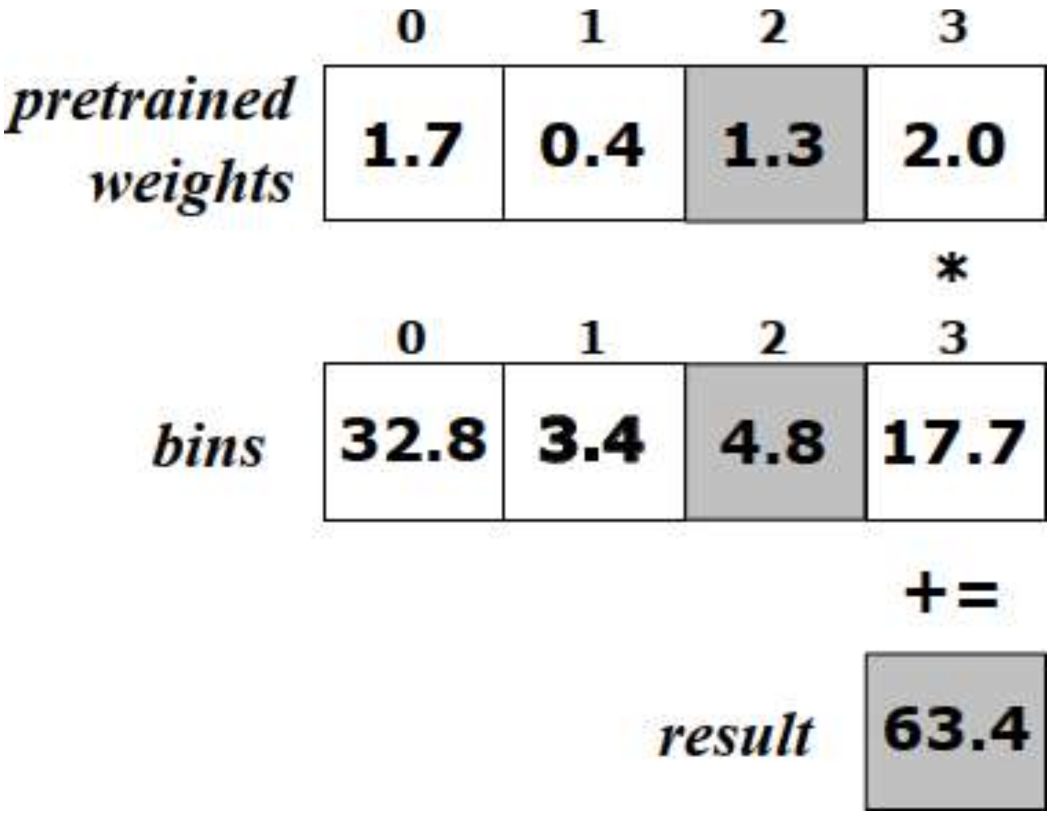
*

	0	1	2	3
bins	32.8	3.4	4.8	17.7

+=

result 57.2

PASM In Operation – PASM Phase



	0	1	2	3
pretrained weights	1.7	0.4	1.3	2.0

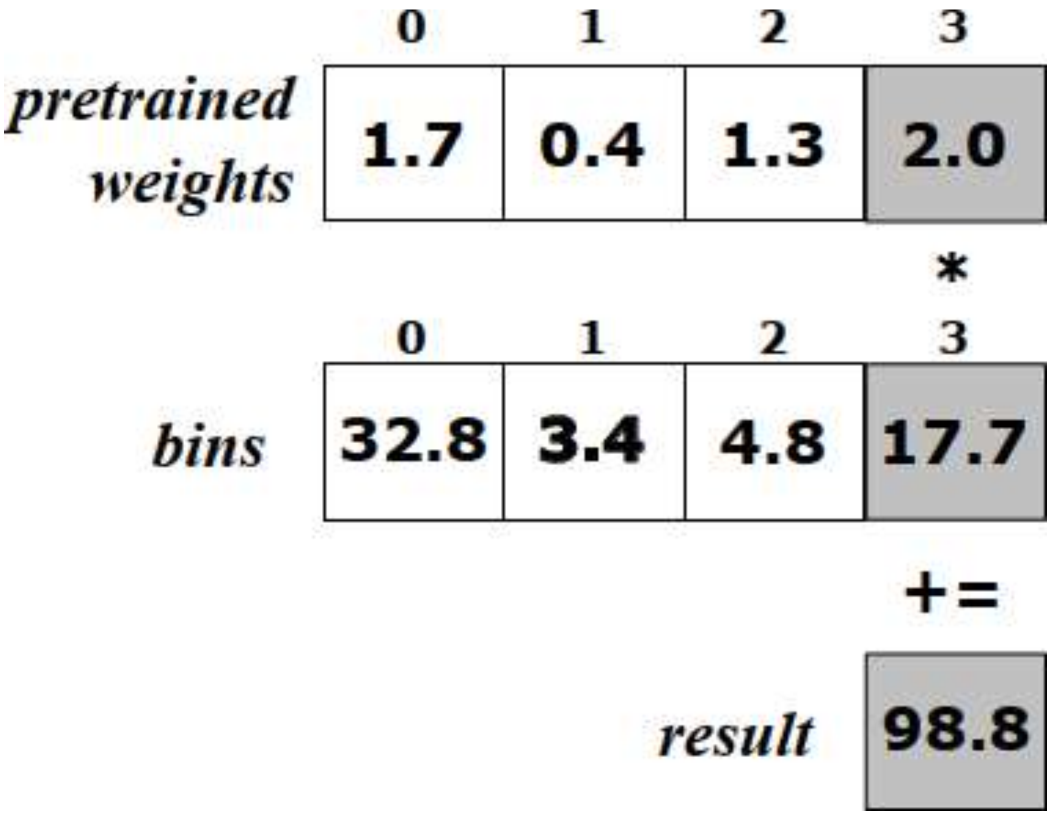
*

	0	1	2	3
bins	32.8	3.4	4.8	17.7

+=

result 63.4

PASM In Operation – PASM Phase



pretrained weights

0	1	2	3
1.7	0.4	1.3	2.0

*

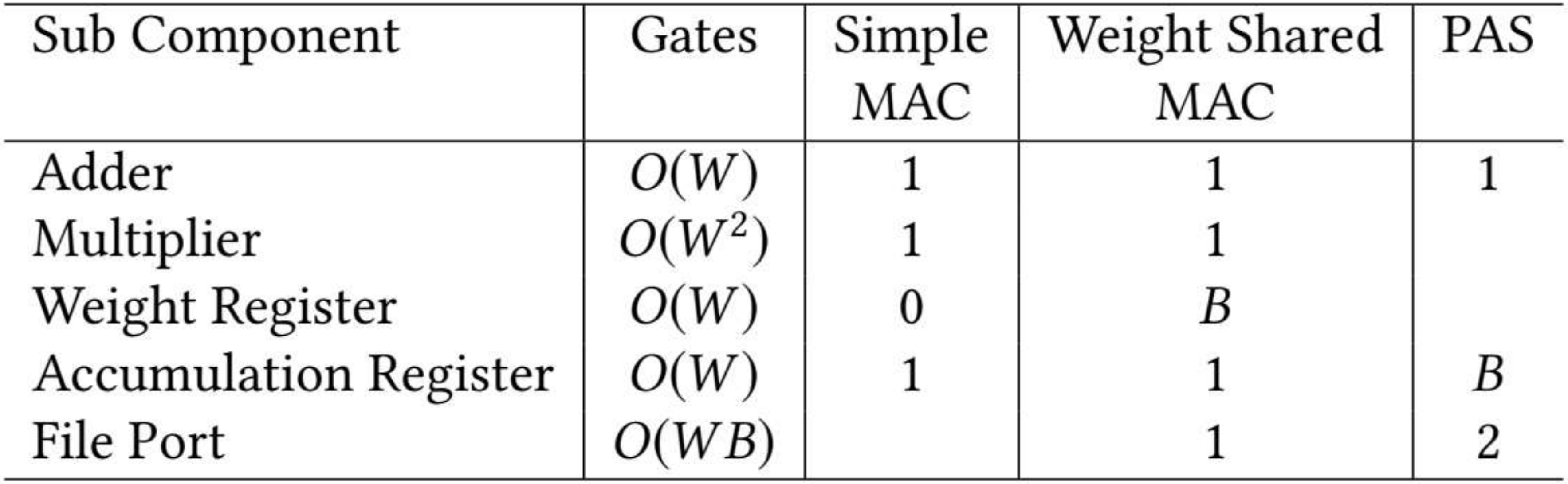
bins

0	1	2	3
32.8	3.4	4.8	17.7

+=

result 98.8

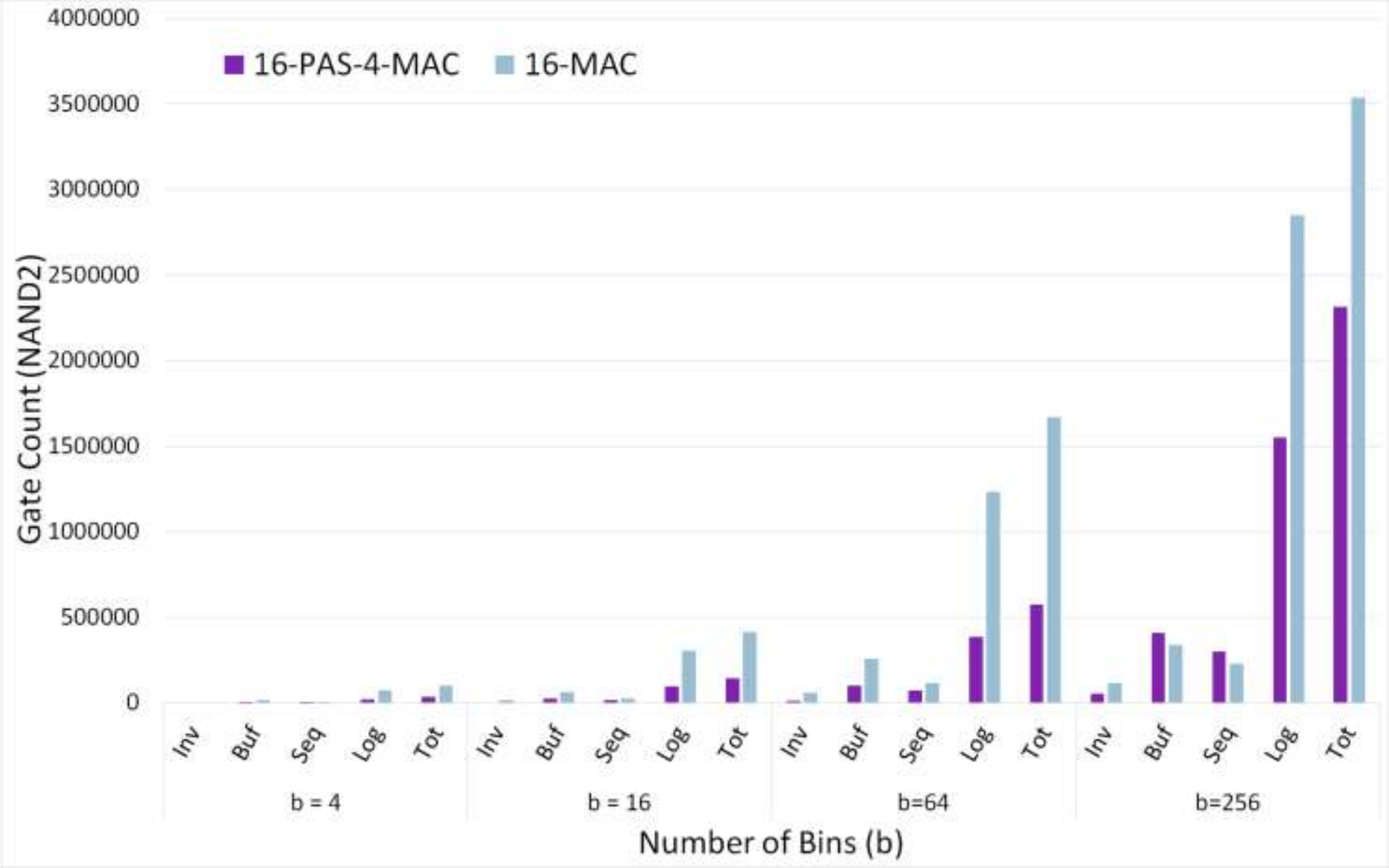
Complexity of the PAS



Sub Component	Gates	Simple MAC	Weight Shared MAC	PAS
Adder	$O(W)$	1	1	1
Multiplier	$O(W^2)$	1	1	
Weight Register	$O(W)$	0	B	
Accumulation Register	$O(W)$	1	1	B
File Port	$O(WB)$		1	2

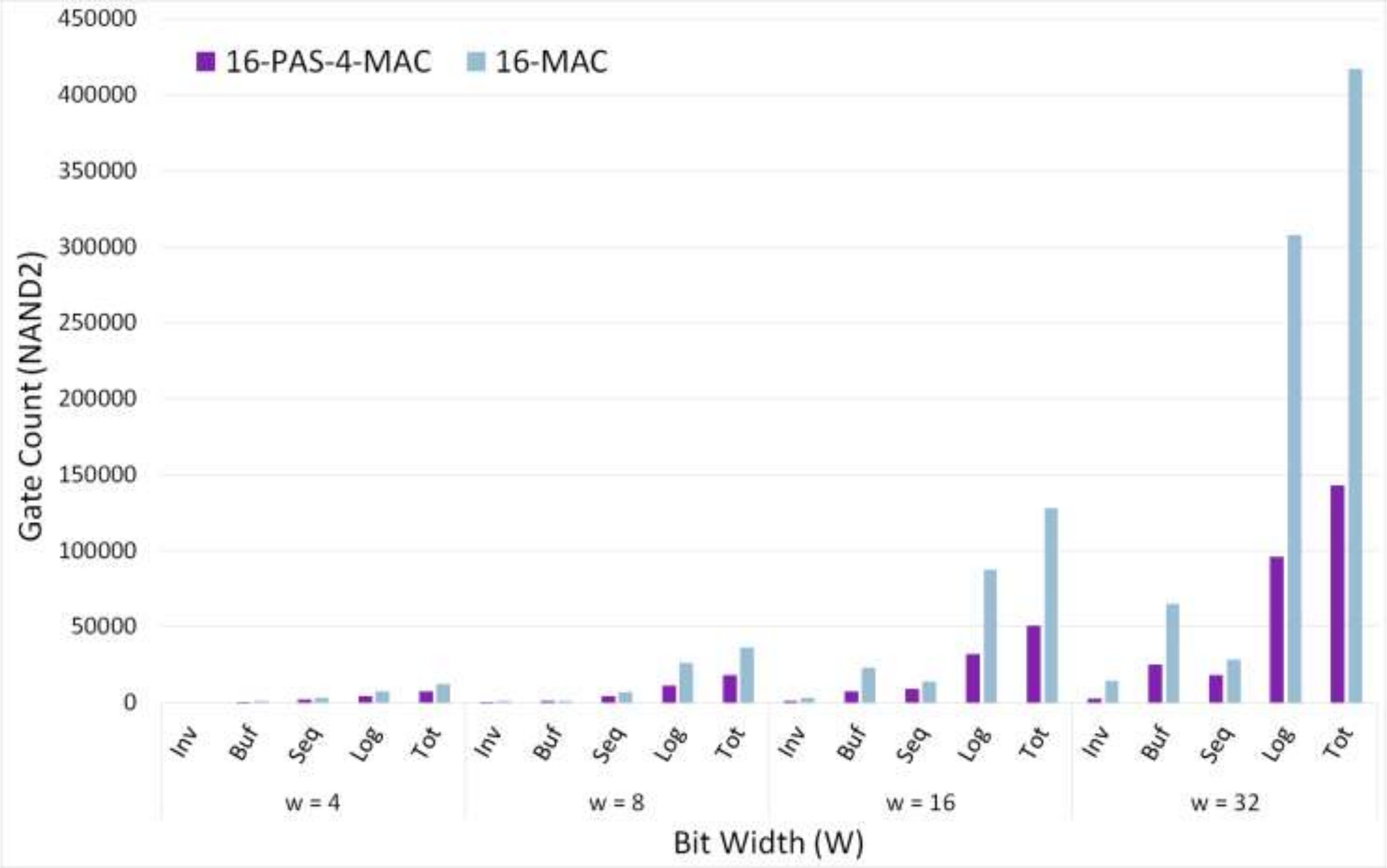
PASM - Gate Count Results

- Utilization results show more **66**% efficiency increase in NAND2 gate count for PASM - **lower is better.**



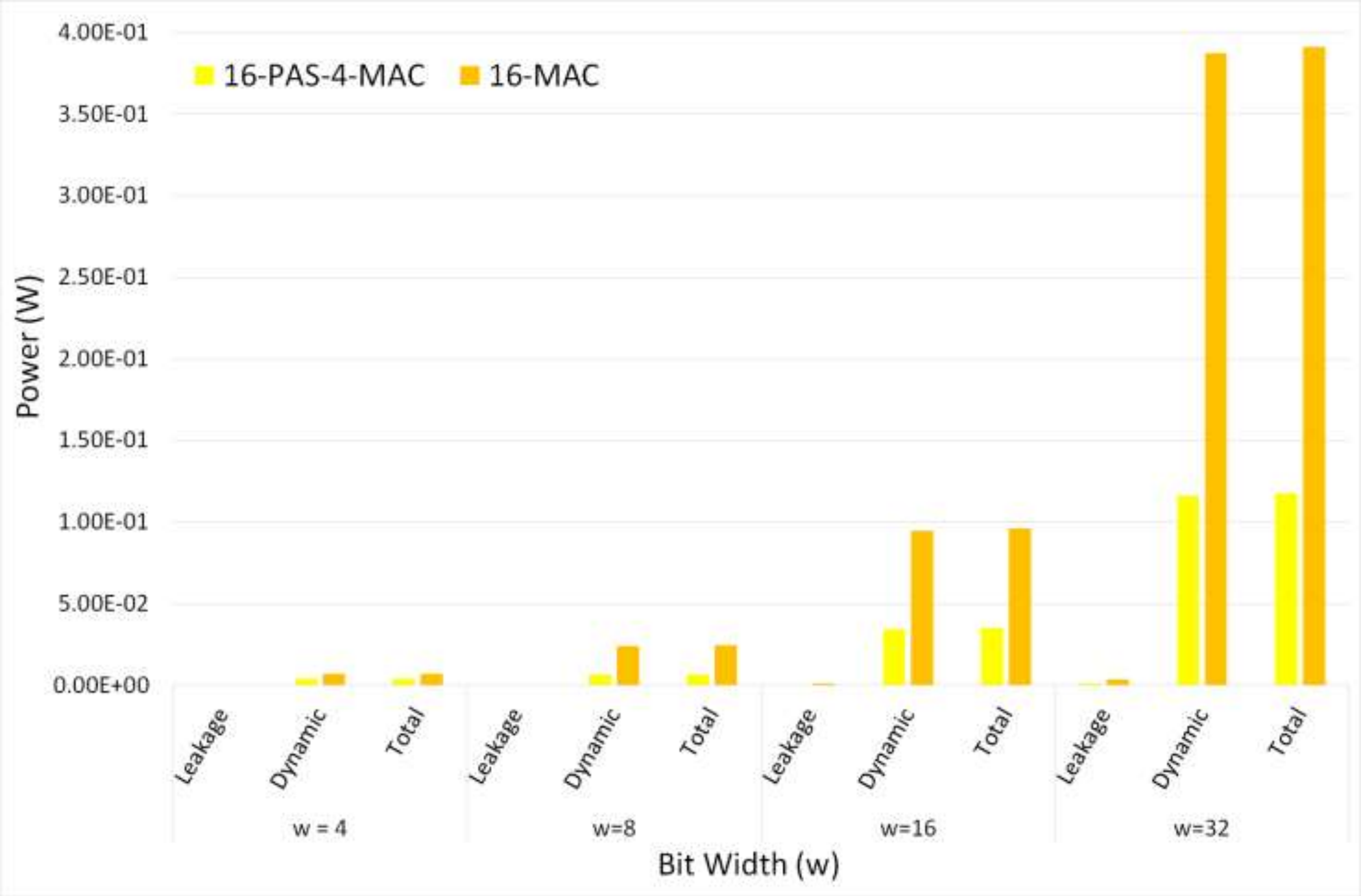
PASM - Gate Count Results

- Utilization results show more 66% efficiency increase in NAND2 gate count for PASM - **lower is better.**



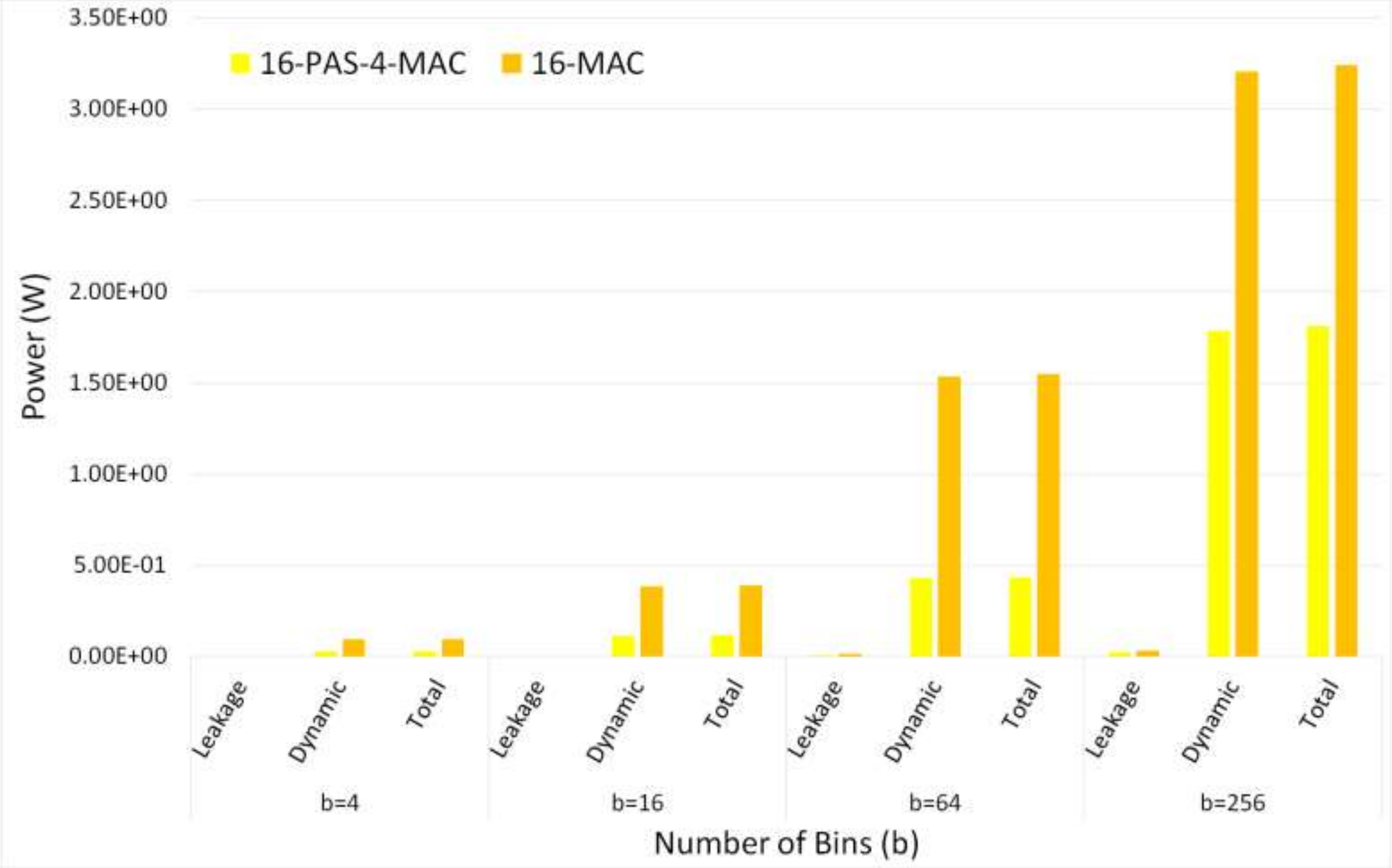
PASM - Power Consumption Results

- Power results show **70%** lower total power consumption for PASM - **lower is better.**



PASM - Power Consumption Results

- Power results show **70%** lower total power consumption for PASM - **lower is better.**



Kernel Idea Published by IEEE CAL

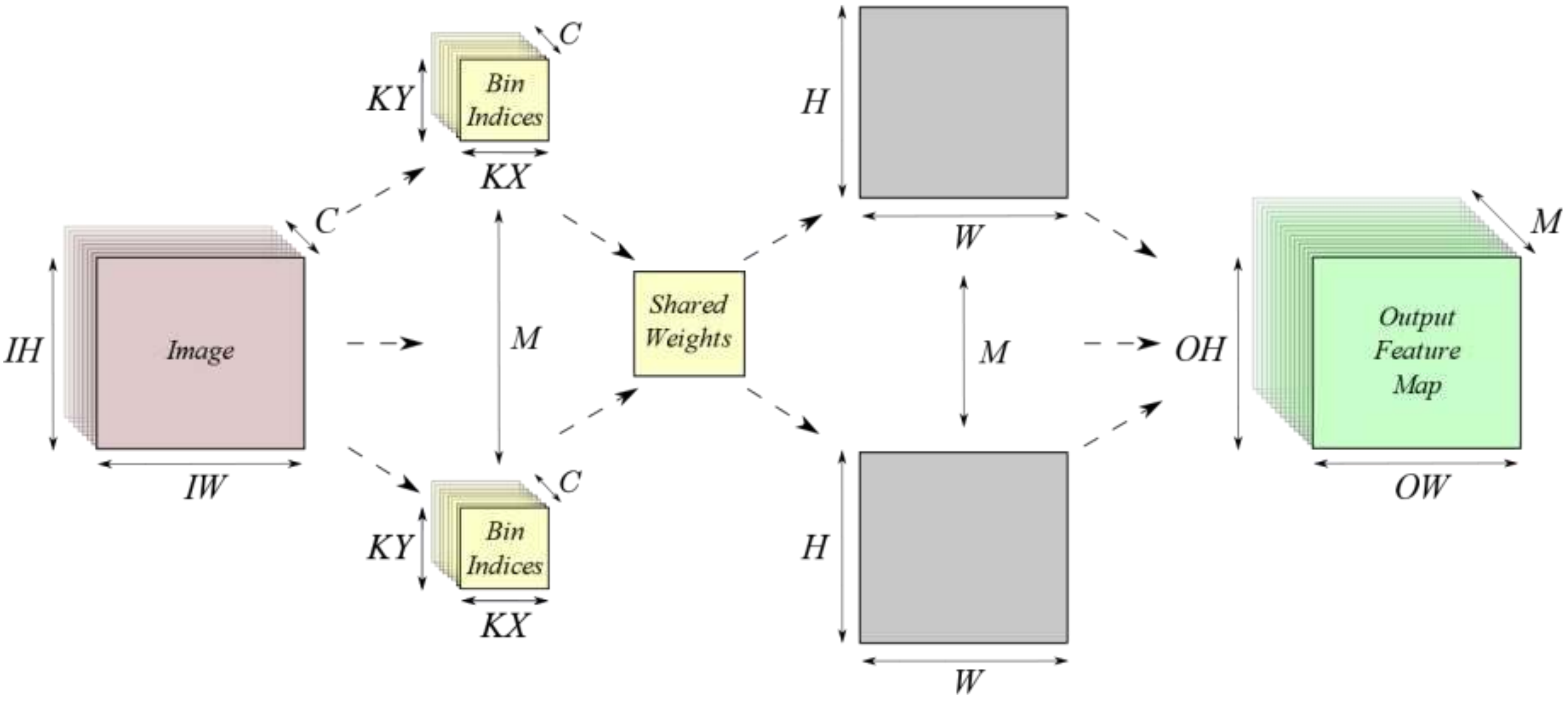
- Short 4 page paper published in IEEE Computer Architecture Letters [8].
- DOI: 10.1109/LCA.2017.2656880
- Cited three times (so far!)

[8] Garland et al. 2017.

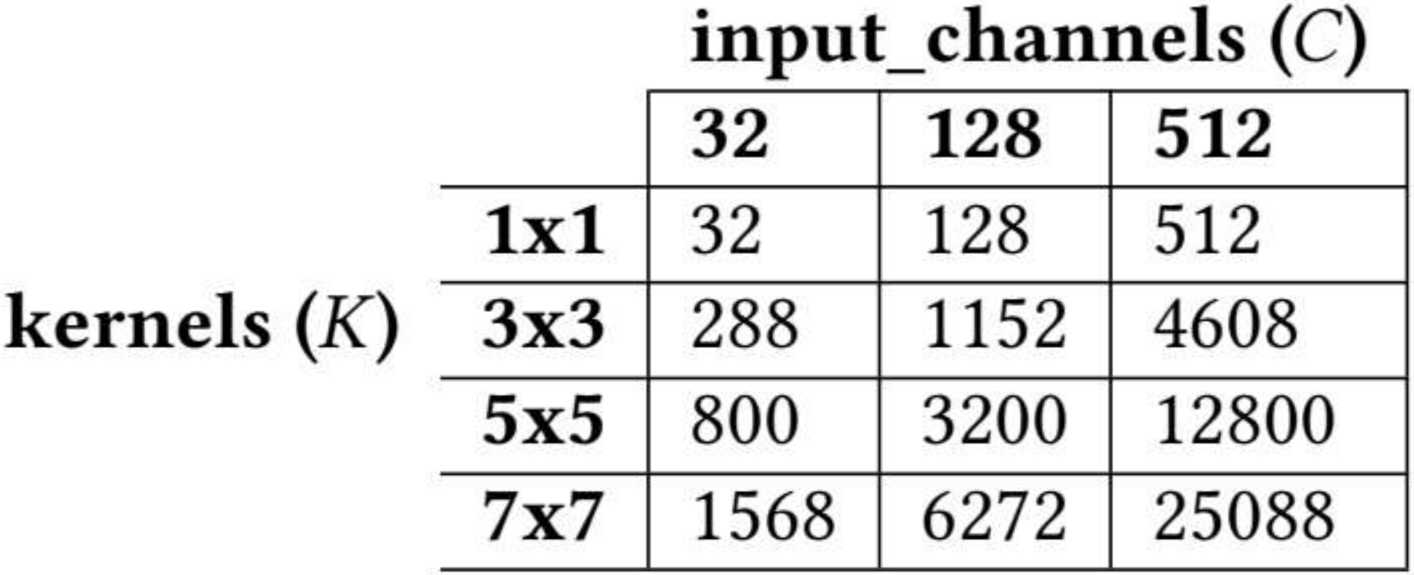
Extended Research

- Designed three CNN accelerators
  - Standard convolution accelerator (no weight sharing).
  - Weight shared convolution accelerator
  - Weight shared convolution accelerator implemented with PASM.
- Designed in System C rather than Verilog
- Optimised / implemented in field programmable gate array (FPGA) and application specific integrated circuit (ASIC)
- Compared timing, latency, power and gate count of the three designs in FPGA and ASIC

For Reference: Weight-Shared CNN

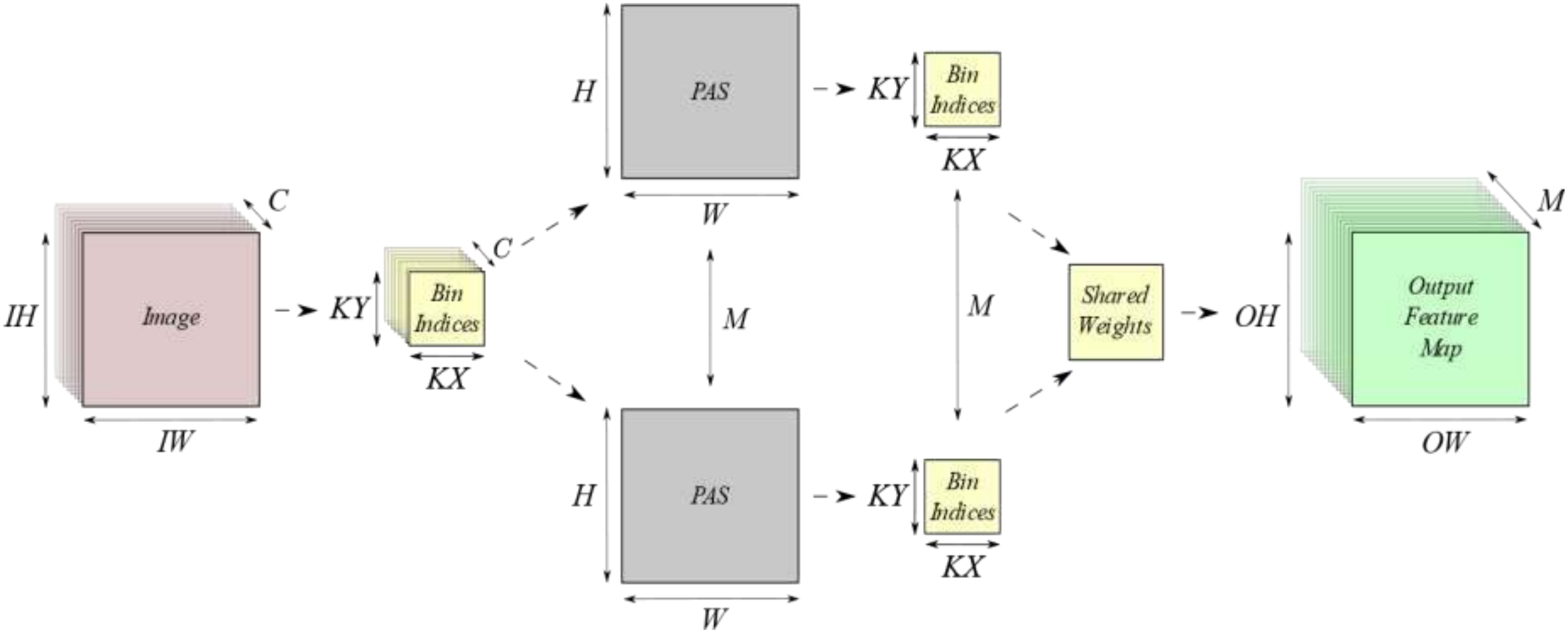


Typical Numbers of MAC Operations

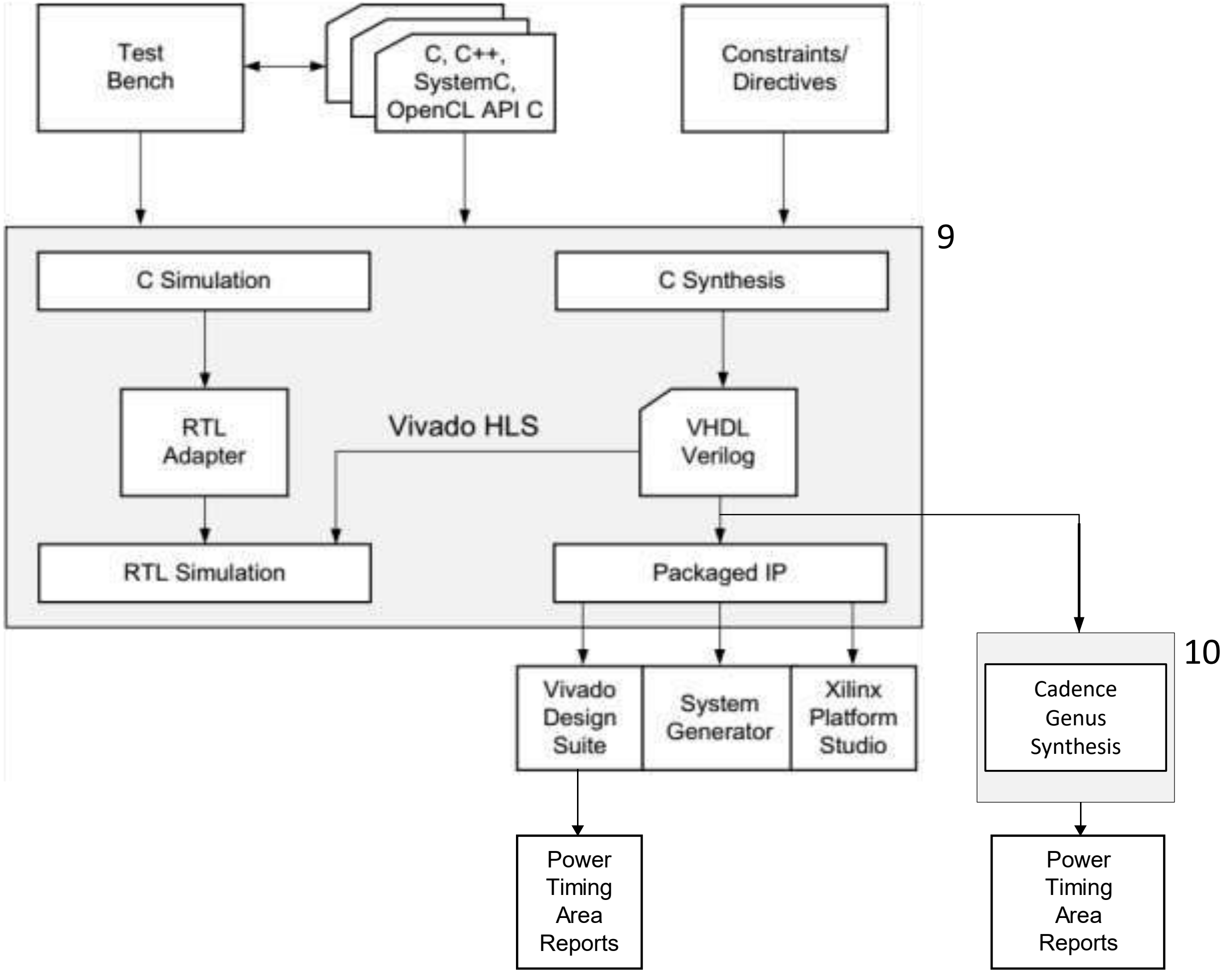


		input_channels (C)		
		32	**128**	**512**
kernels (K)	**1x1**	32	128	512
	3x3	288	1152	4608
	5x5	800	3200	12800
	7x7	1568	6272	25088

Weight-Shared Convolution with PASM



Development Flow - FPGA and ASIC



[9] Xilinx User Guide 902 Vivado High Level Synthesis.

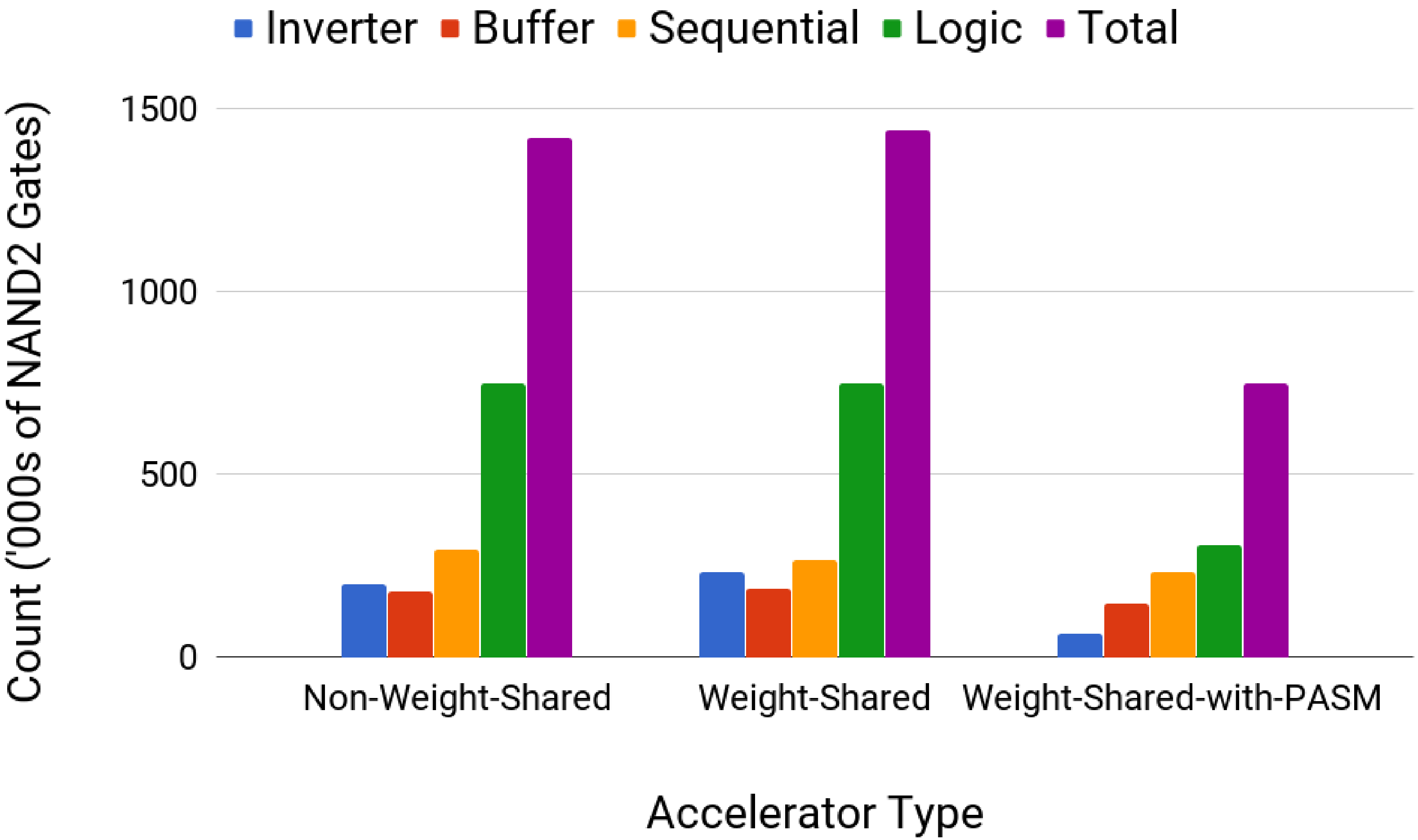
[10] Cadence Genus User Guide.

PASM in CNN Convolution Layer

ASIC Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8% increase in latency
48% less total area
53% less total power

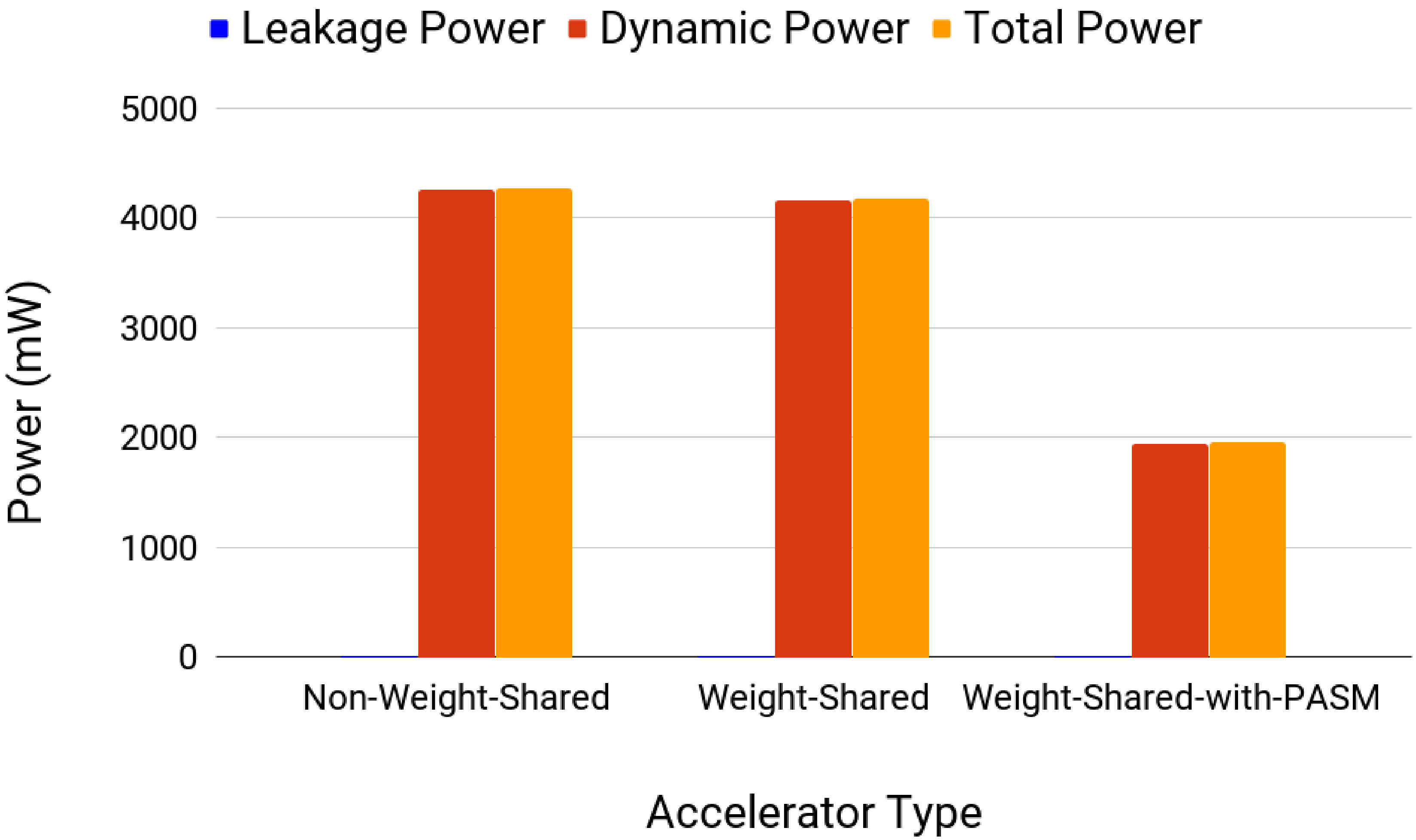


PASM in CNN Convolution Layer

ASIC Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8% increase in latency
48% less total area
53% less total power

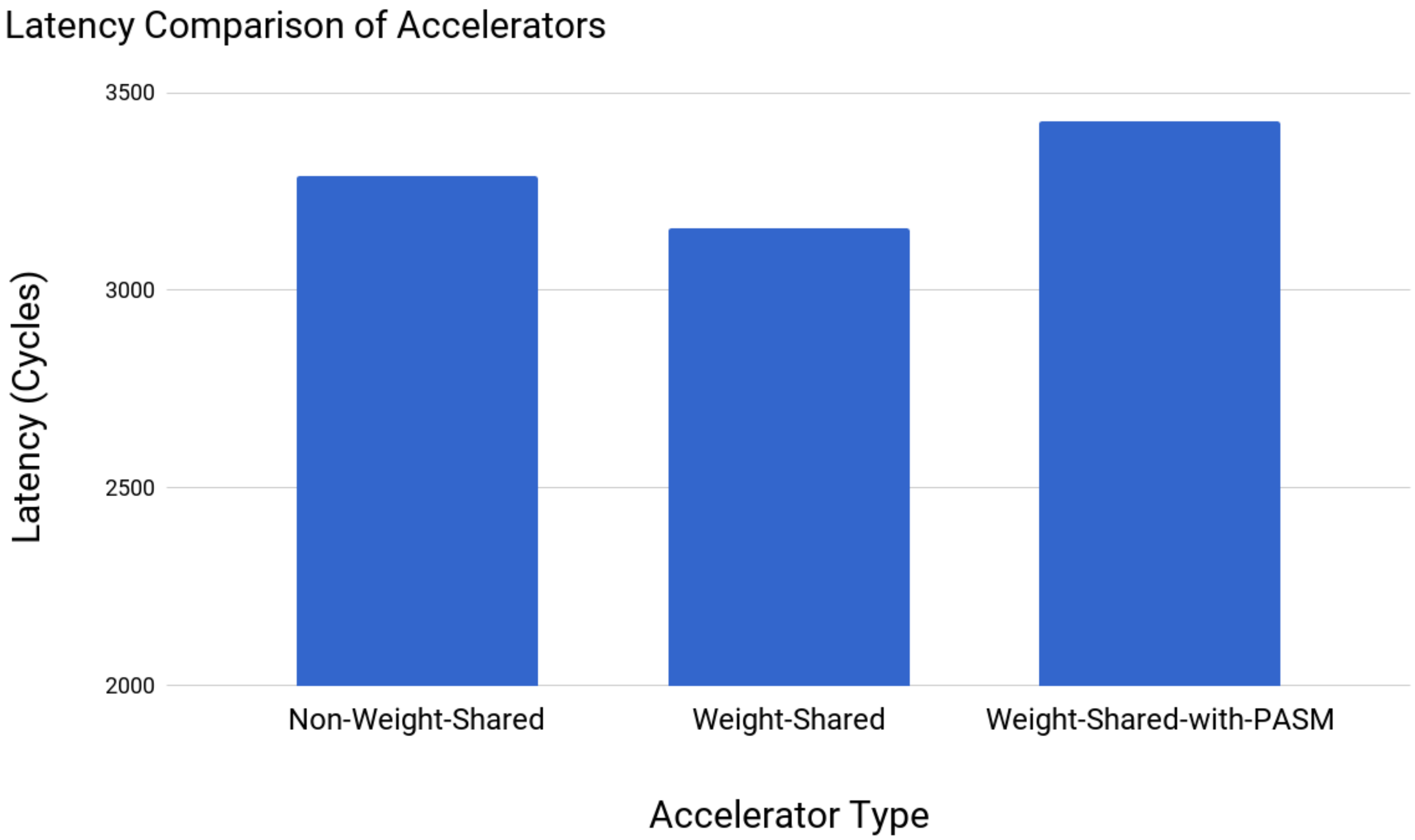


PASM in CNN Convolution Layer

ASIC Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8% increase in latency
48% less total area
53% less total power

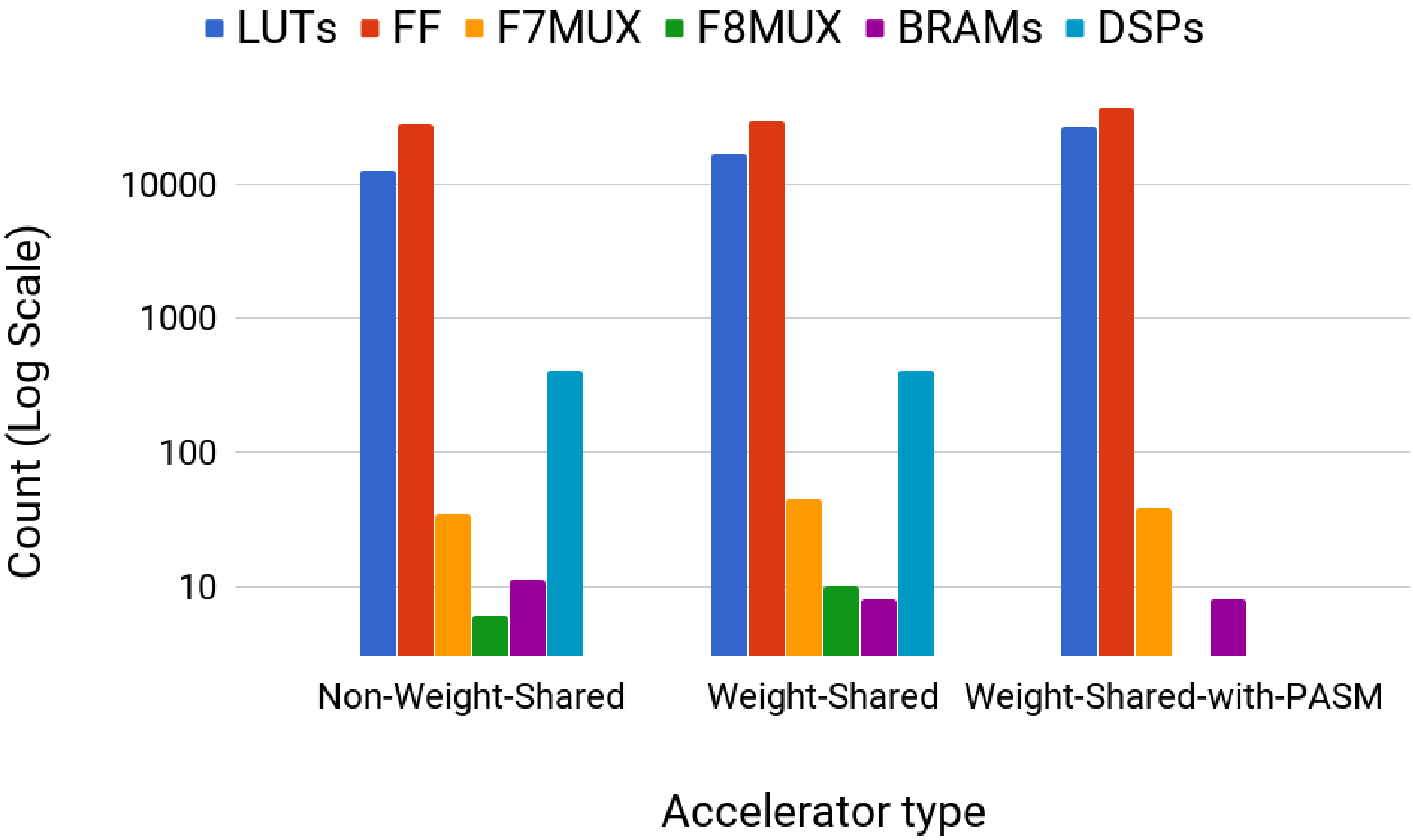


PASM in CNN Convolution Layer

FPGA Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8.5% increase in latency
99% fewer DSPs
28% fewer BRAMs
80% power saving

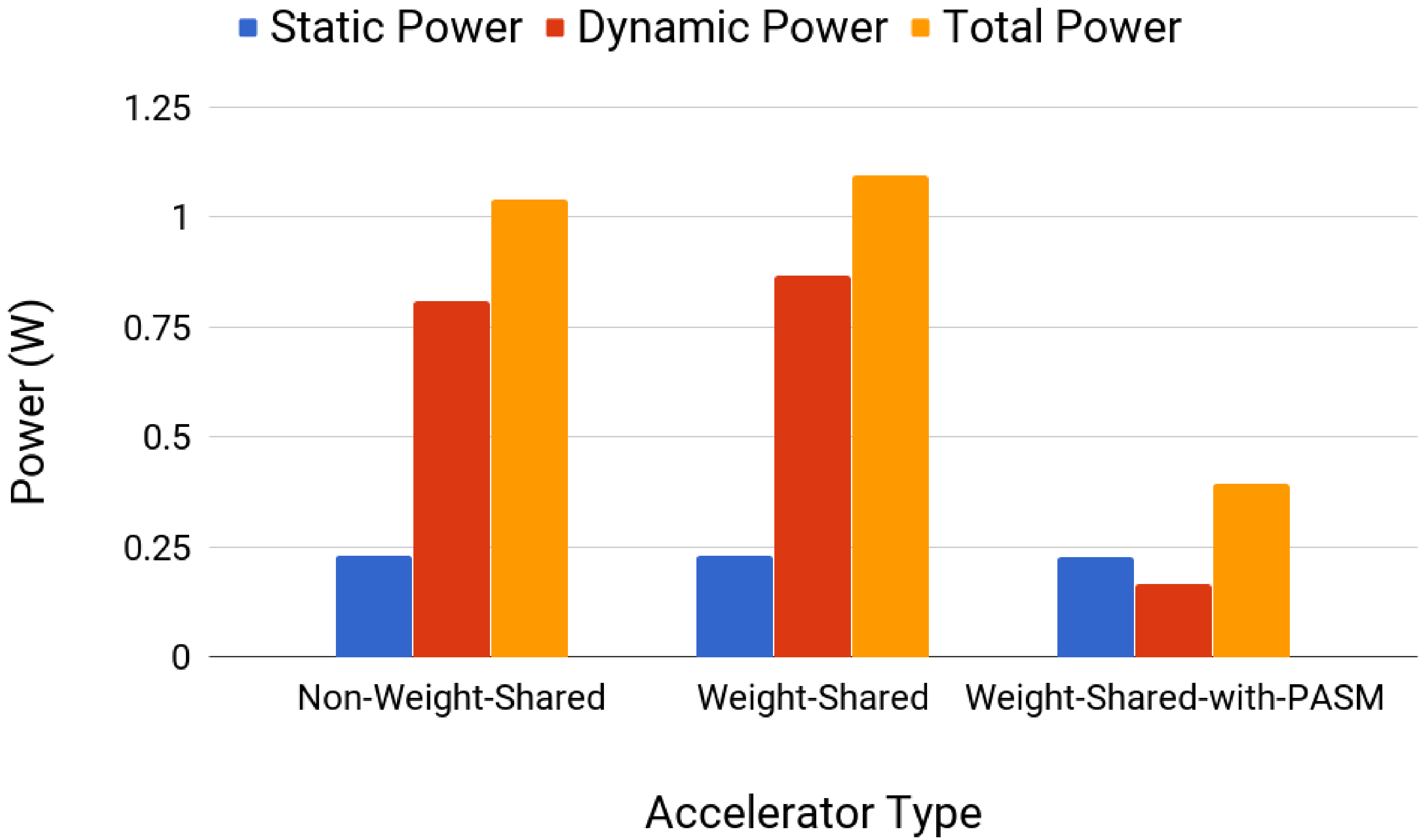


PASM in CNN Convolution Layer

FPGA Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8.5% increase in latency
99% fewer DSPs
28% fewer BRAMs
80% power saving

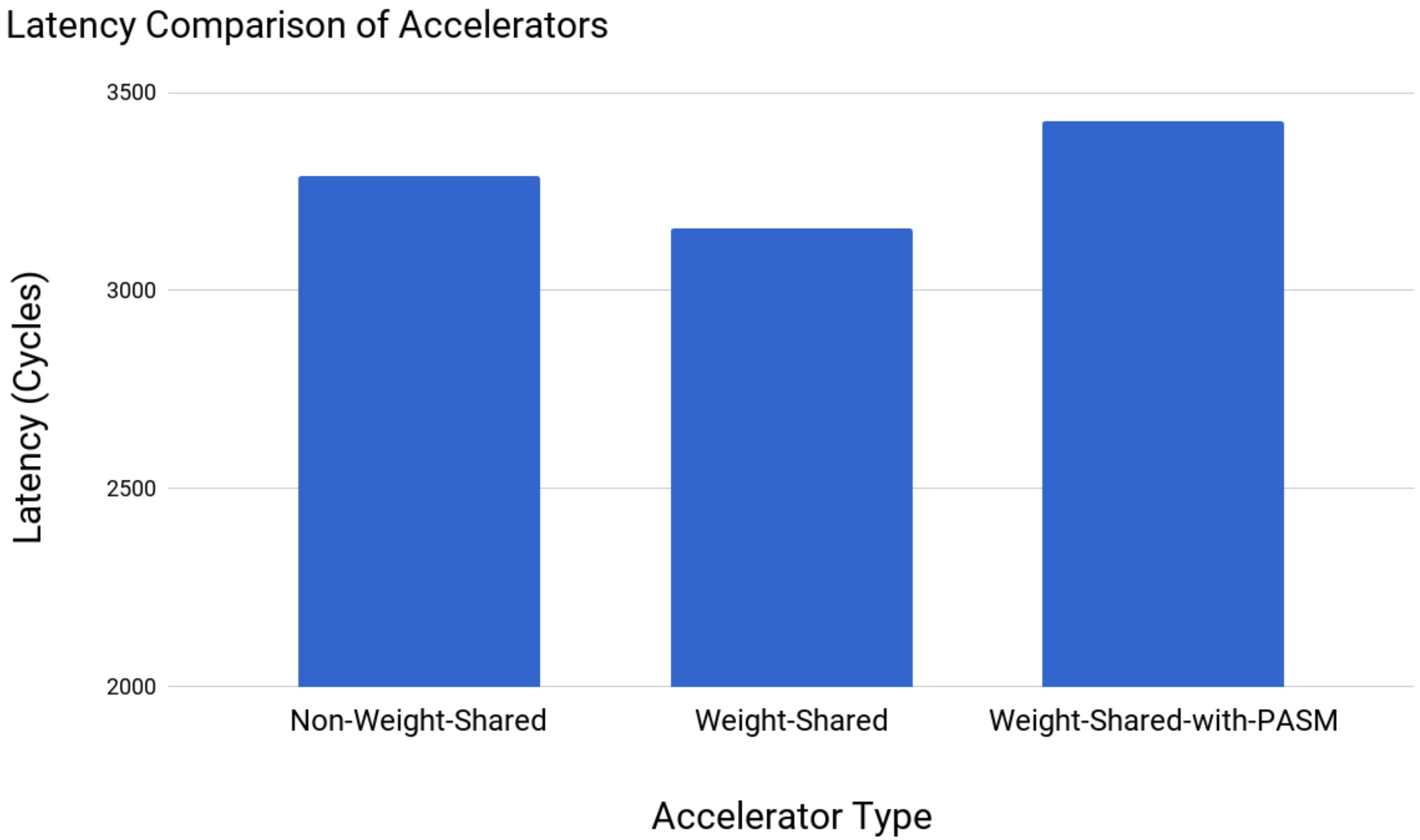


PASM in CNN Convolution Layer

FPGA Results

- 4 bin - 32 bit values, IMG=5 × 5, K=3 × 3, C=15, M=2

8.5% increase in latency
99% fewer DSPs
28% fewer BRAMs
80% power saving



Extended Idea Published by ACM TACO

- 25 page paper published in ACM TACO [11].
- DOI: 10.1145/3233300
- Cited once (so far!)

[11] Garland et al. 2018.

To Sum Up

- There's a great need to reduce power and resources in a CNN.
- This will aid power consumption in data centres, allow implementation in low power embedded devices and save the environment.
- We change the programming model of CNN by rearchitecting the MAC.
- These are optimised / implemented in FPGA and ASIC.
  - **8.5%** increase in latency for PASM
  - ASIC: **48%** less total area; **53%** less total power
  - FPGA: **99%** fewer DSPs; **28%** fewer BRAMs; **80%** less total power
- We show timing, power and ASIC gate count and FPGA resources of the three designs are reduced with only a slight increase in latency.

[12] The Irish News, 2018.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You

James Garland https://www.scss.tcd.ie/~jgarland/
David Gregg https://www.scss.tcd.ie/David.Gregg/