Distilling Bit-level Sparsity Parallelism for General Purpose Deep Learning Acceleration

路航[1], 常亮[2], 李成龙[2], 竹子轩[2], 鲁圣健[1], 刘艳欢[1] , 张明喆[3]

[1]State Key Laboratory of Computer Architecture, Institute of Computing Technology, CAS, Beijing, China

[2]University of Electronic Science and Technology of China, Chengdu, China

[3]State Key Laboratory of Information Security, Institute of Information, CAS, Beijing, China

计算机体系结构国家重点实验室
State Key Laboratory of Computer Architecture, ICT, CAS

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

电子科技大学

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING, CAS

Executive Summary

The contribution of this work

1. Propose a novel philosophy of leveraging bit-level sparsity – **bit interleaving**
2. Propose a specialized general-purpose accelerator – **bitlet**, to mine the maximum potential of bit interleaving

Benefits:

① General purpose	Leveraging the sparsity for accelerating both training and inference
② Multi-precision support	Floating point fp32/16, fixed point from 1b~24b
③ High performance and efficiency	Up to 15× and 81× speedup over GPUs

OUTLINE

Motivation & Background

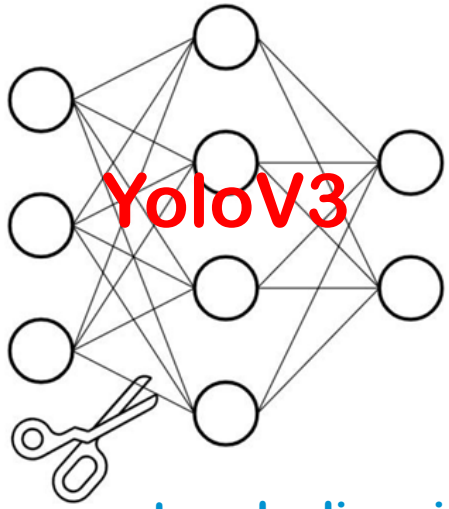
Methodology

Evaluation

Recap

Motivation & Background

- The benefits of general purpose accelerators:
  - to fit for various DL tasks
  - that can be applied in both **training** and **inference**
- How to boost the performance - **leveraging the sparsity of the operands**
  - Software-based pruning
  - Quantization-aware training



network slimming (ICCV'17)

Weakness

2~3 days sparse training ☹

Time-consuming!

3~4 days retraining / fine-tuning ☹

Quantization

+1/-1 weights, Courbariaux 2016
Bineray, Courbariaux 2015

INQ, Zhou 2017

Ternary, Li 2016

.....

Weakness

Requires tedious re-training ☹
Suffering from accuracy loss ☹
Not precision tunable ☹

Motivation & Background

- The headroom of value-level sparsity is very limited.
- However, bit-level sparsity is inherently fertile.

Model	Weight Sparity	Bit Sparity
DenseNet121	4.84%	48.64%
ResNet50	0.33%	48.64%
ResNet152	0.75%	48.64%
ResNext50_32x4d	0.37%	48.64%
ResNext101_32x8d	3.43%	48.65%
InceptionV3	0.05%	48.64%
MNASNet0.5	0.00%	48.60%
MNASNet1.0	8.07%	48.98%
MobileNetV2	0.01%	48.67%
ShuffleNetV2_x0_5	0.00%	48.36%
ShuffleNetV2_x1_0	1.53%	48.63%
SqueezeNet1_0	0.05%	48.64%
SqueezeNet1_1	0.02%	48.64%

Very limited headroom

Weight sparsity : the values below 10^{-5} over the total parameter size

Significantly abundant

Bit sparsity : total bit 0s over the total “bit count” of the mantissas

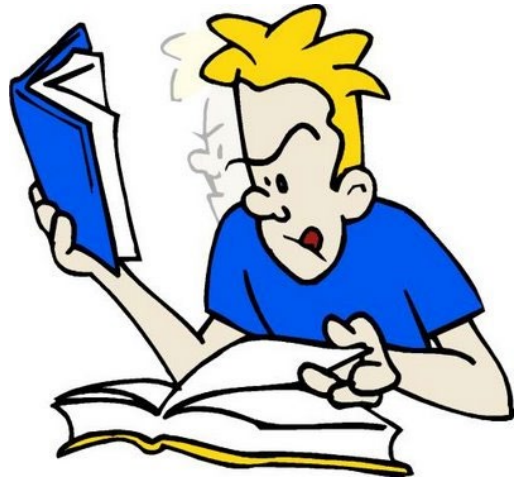
Motivation & Background

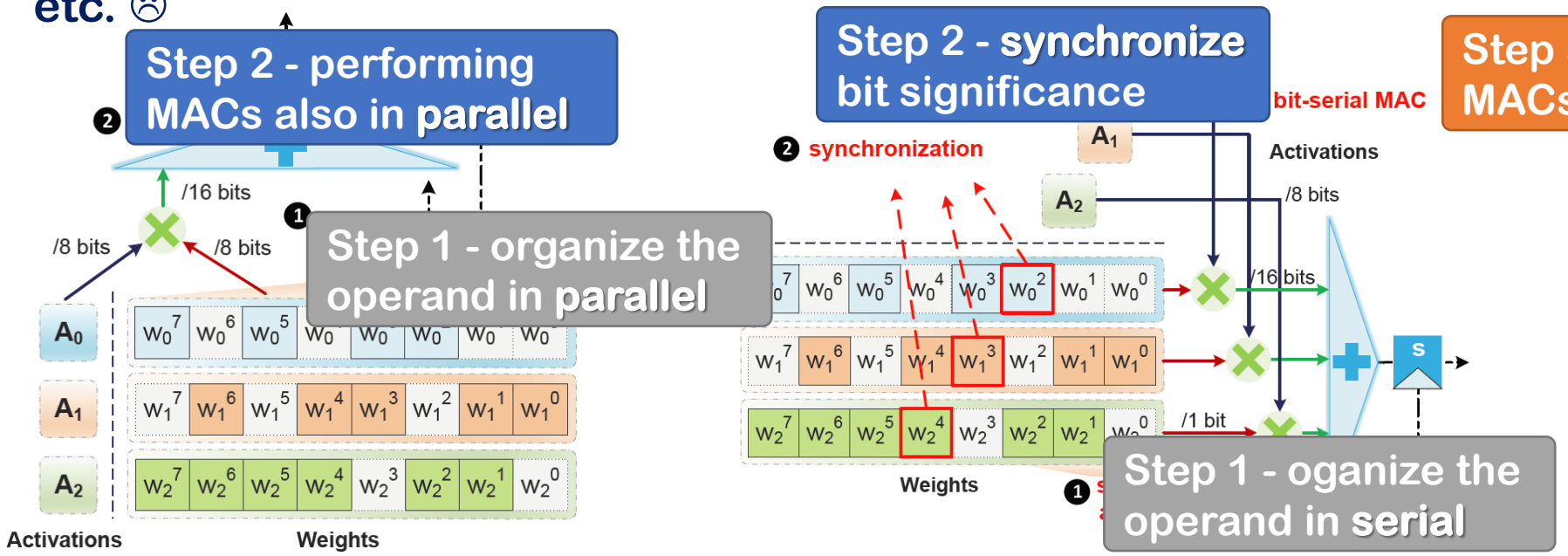
Any state-of-the-art solutions?

Phil.	Design	Sparsity Exploited	Precisions	Training Support
Bit parallel	Eyeriss, DaDianNao	N/A	16b	No
	Cambricon-S, EIE	A- / W- value	16b	No
	SCNN	A- & W- value	16b	No
Bit serial	UNPU, Stripes	N/A	1~16b	No
	Bit Fusion	N/A	2,4,8,16b	No
	Pragmatic	A- / W- bit	1~16b	No
	Bit Tactical	A- bit & W-value	1~16b	No
	Laconic	A- & W- bit	1~16b	No
Bit interleaving	Bitlet (Ours)	W- bit & W-value (or A- bit & A-value)	fp32/16, 1~24b	Yes

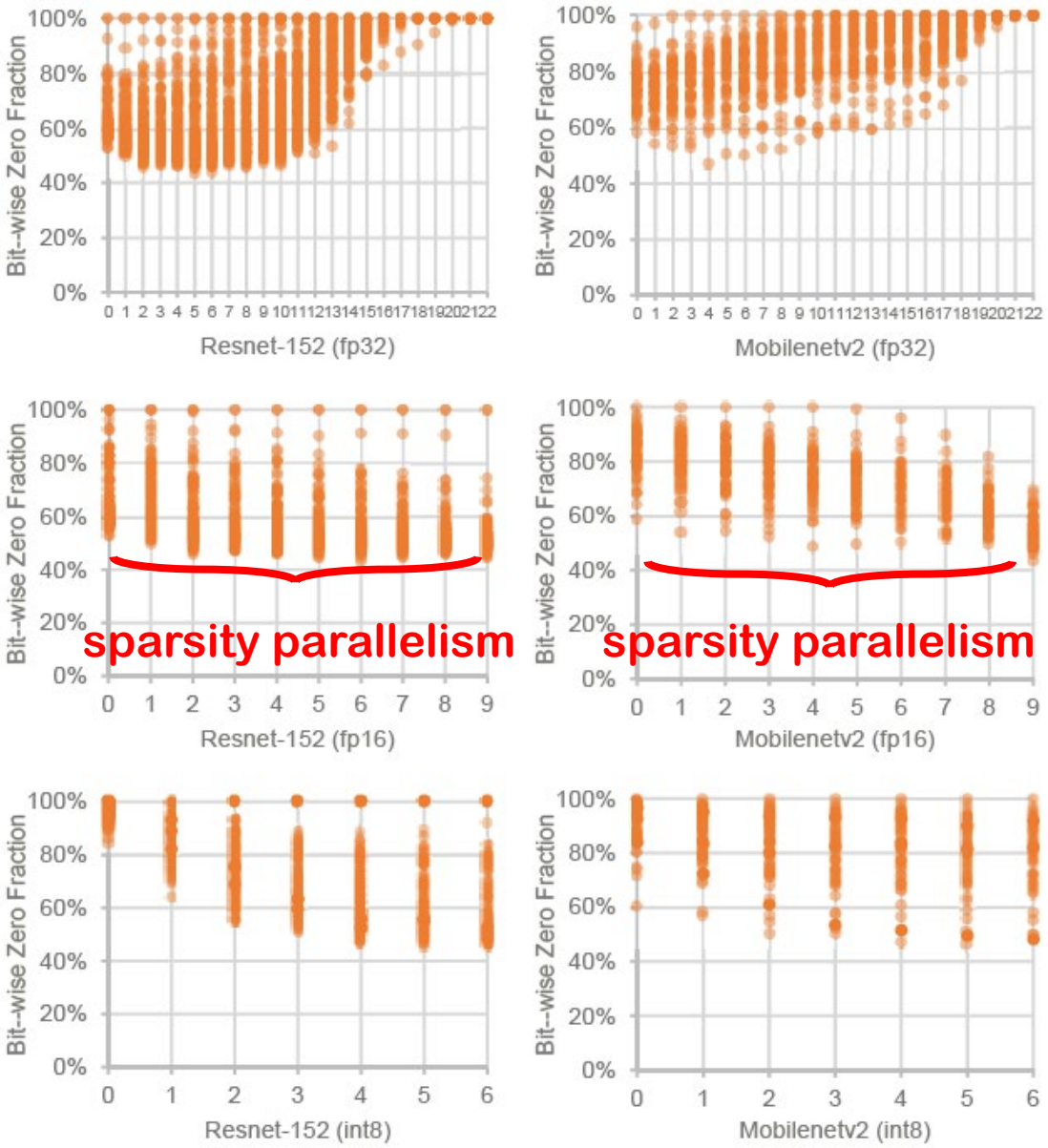
Motivation & Background

- **The weaknesses of such design philosophy:**
  - **None of them are general-purpose!**
    - Only use in inference ☹
    - Only support fixed-point arithmetic ☹
  - **Sub-optimal sparsity utilization**
    - Bit-parallel cannot leverage bit-level sparsity ☹
    - Bit-serial must incur synchronization, ie. Booth coding, look ahead synchronization etc. ☹





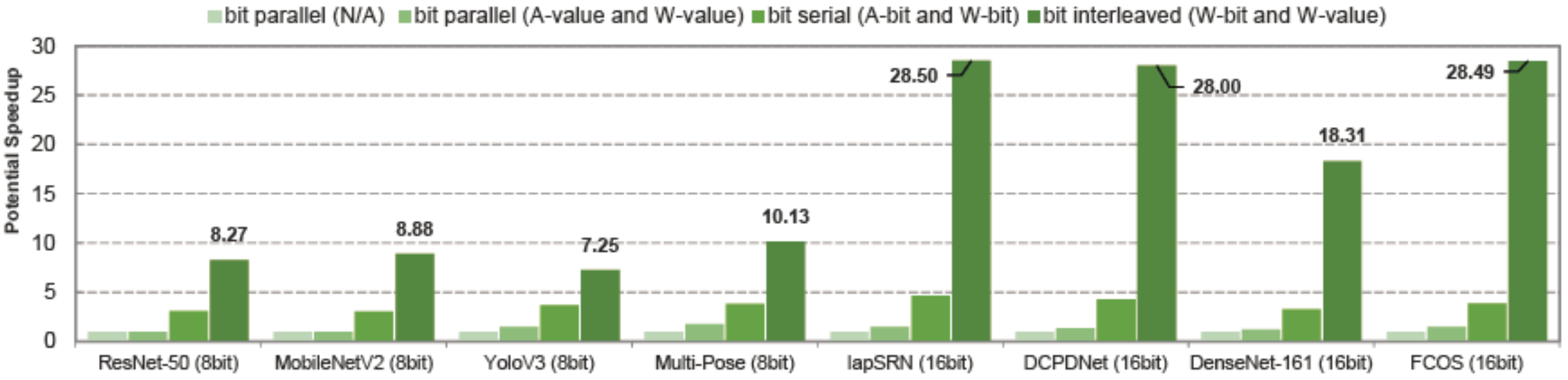
Motivation & Background – key observation #1



- High sparsity percentage at each bit significance
- The sparsity is nearly uniform in terms of :
  - Different precisions, including floating point, fixed point and integer
  - Different bit significances (for floating point, we focus on the mantissa)

Motivation & Background -- key observation – #2

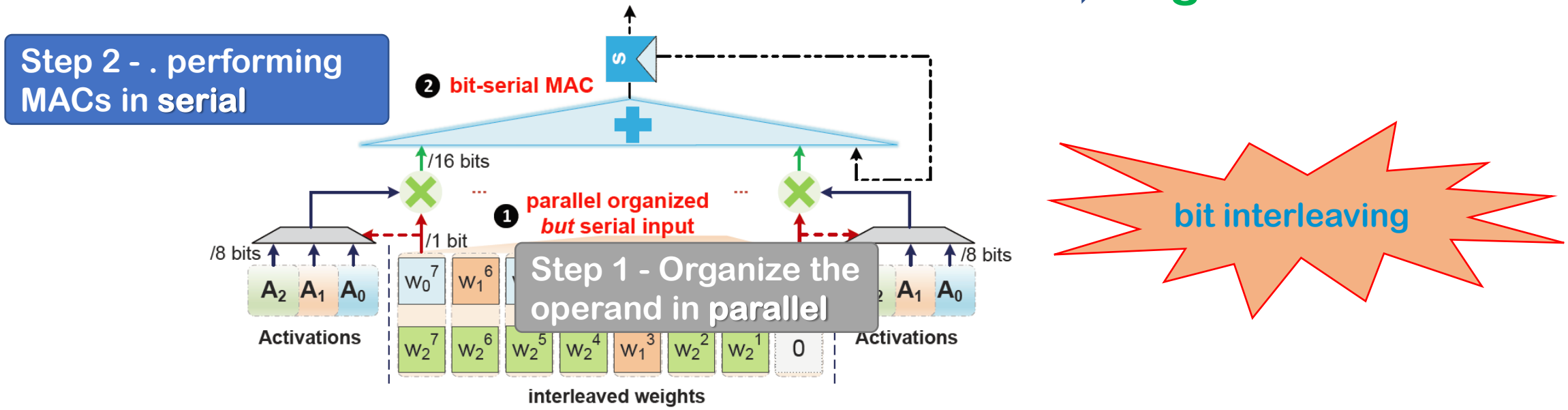
- Leveraging such parallelism is **beneficial**
  - **No synchronization** is ever required.
  - Bit-level arithmetic could be issued **independently**.



Motivation & Background

Our solution – bit interleaving

- Can we obtain the benefits of them both? → High TOPs/W ☺
  - Efficient utilization of the sparsity
  - Avoid the complex synchronization procedures
- Most importantly, design a general-purpose accelerator that can leverage the bit-level sparsity → High TFLOPS/W ☺



OUTLINE

- Motivation & Background
- 

Methodology
- Evaluation
- Recap

Methodology – theorem

Computing pattern analysis

Floating-point MAC decomposition:

Floating point MAC could be transformed to pure shift and add

How we embark

$$\sum_{i=0}^{N-1} A_i \times W_i = \sum_{i=0}^{N-1} (-1)^{S_{W_i}} A_i \times M_{W_i} \times 2^{E_{W_i}}$$

Sign

Exponent

mantissa

Detailed deduction is in the paper

$$\sum_{i=0}^{N-1} \sum_{b=E_i-E_{max}}^{E_i-E_{max}-23} \left[(-1)^{S_{W_i}} \oplus S_{A_i} \cdot \left(M_{A_i} \times M_{W_i}^{b} \right) \right] \times 2^{E_{max}+b}$$

Exponent matching

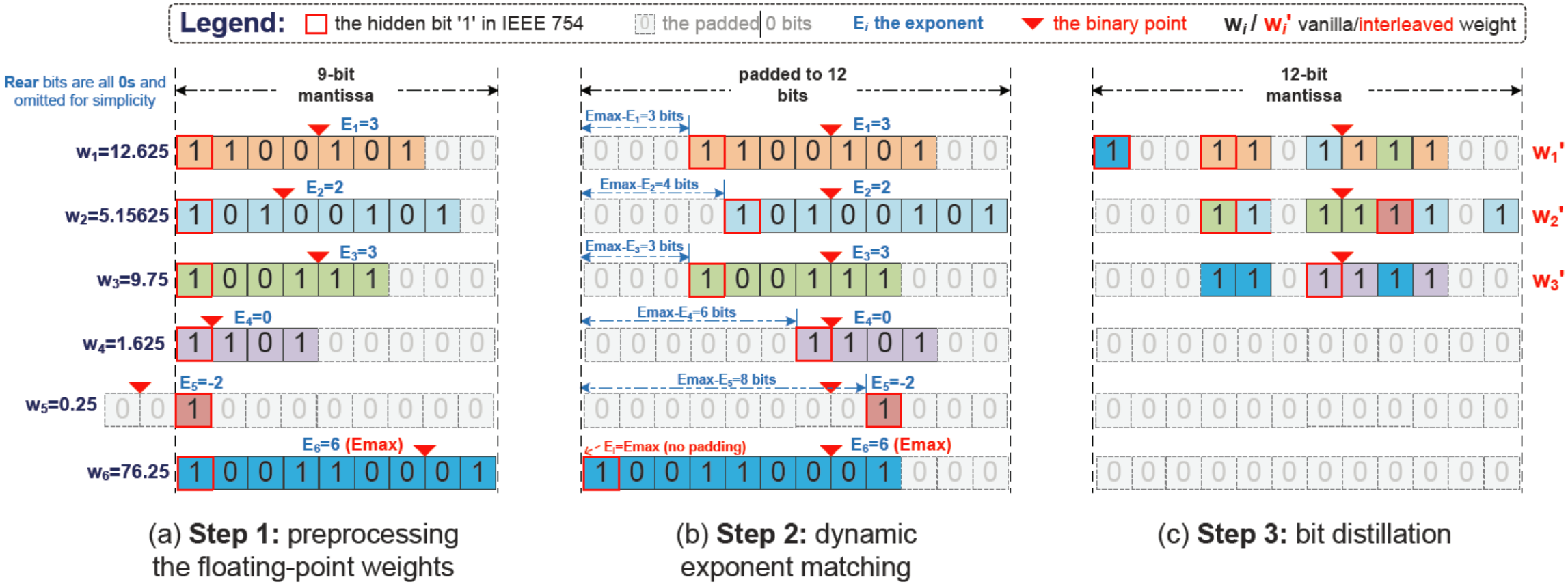
Bit-level arithmetic

Bit significance

Final sign

Maximum exponent

Methodology – bit interleaving



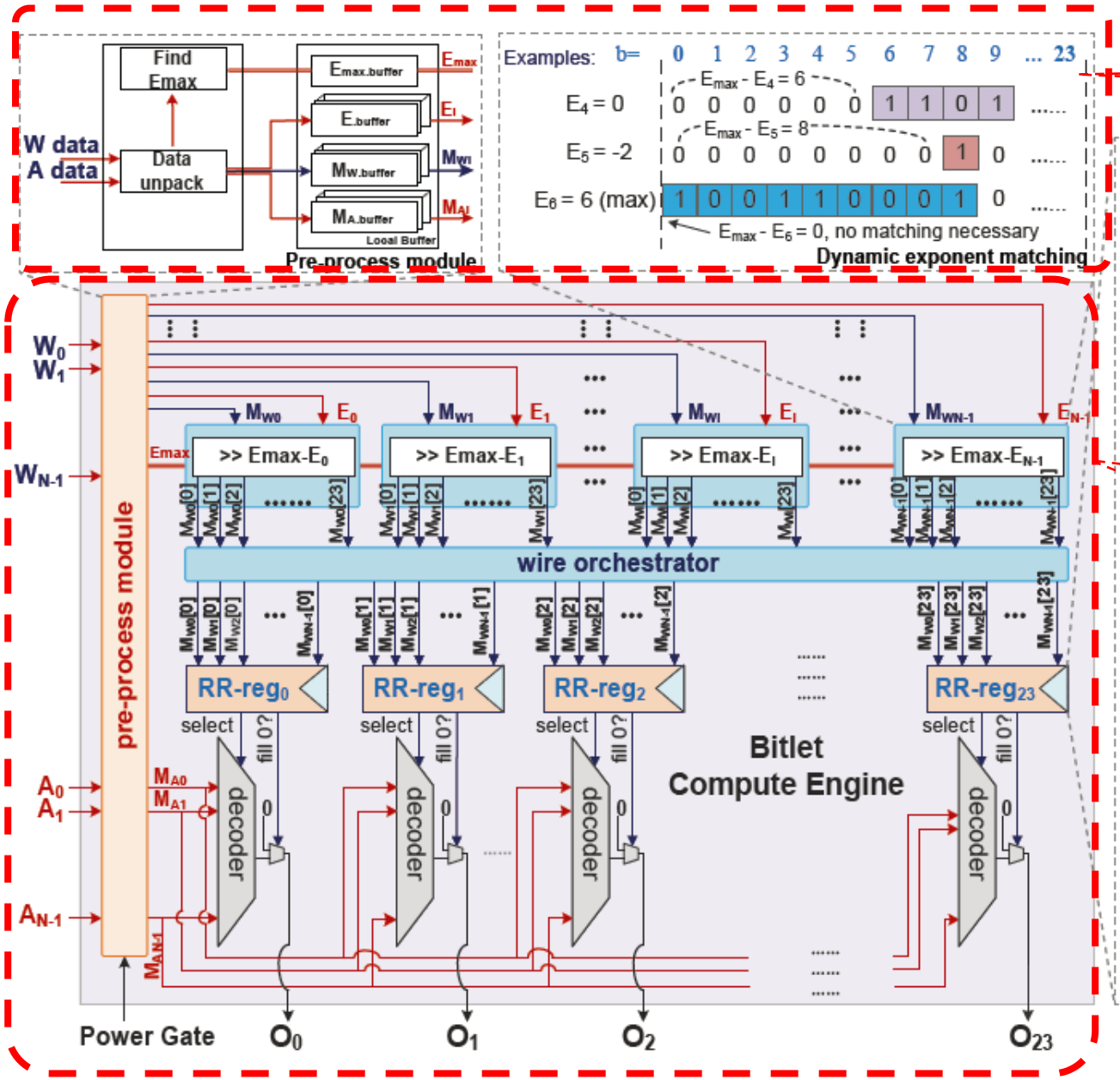
(a) **Step 1:** preprocessing the floating-point weights

(b) **Step 2:** dynamic exponent matching

(c) **Step 3:** bit distillation

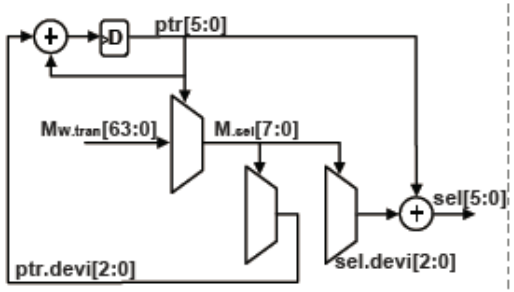
- **Operations:**
  - **Dynamic exponent matching – matching each exponent to Emax**
  - **Bit distillation – leveraging the sparsity!**

Methodology – bitlet accelerator



High level example

Diagram of RR-reg:

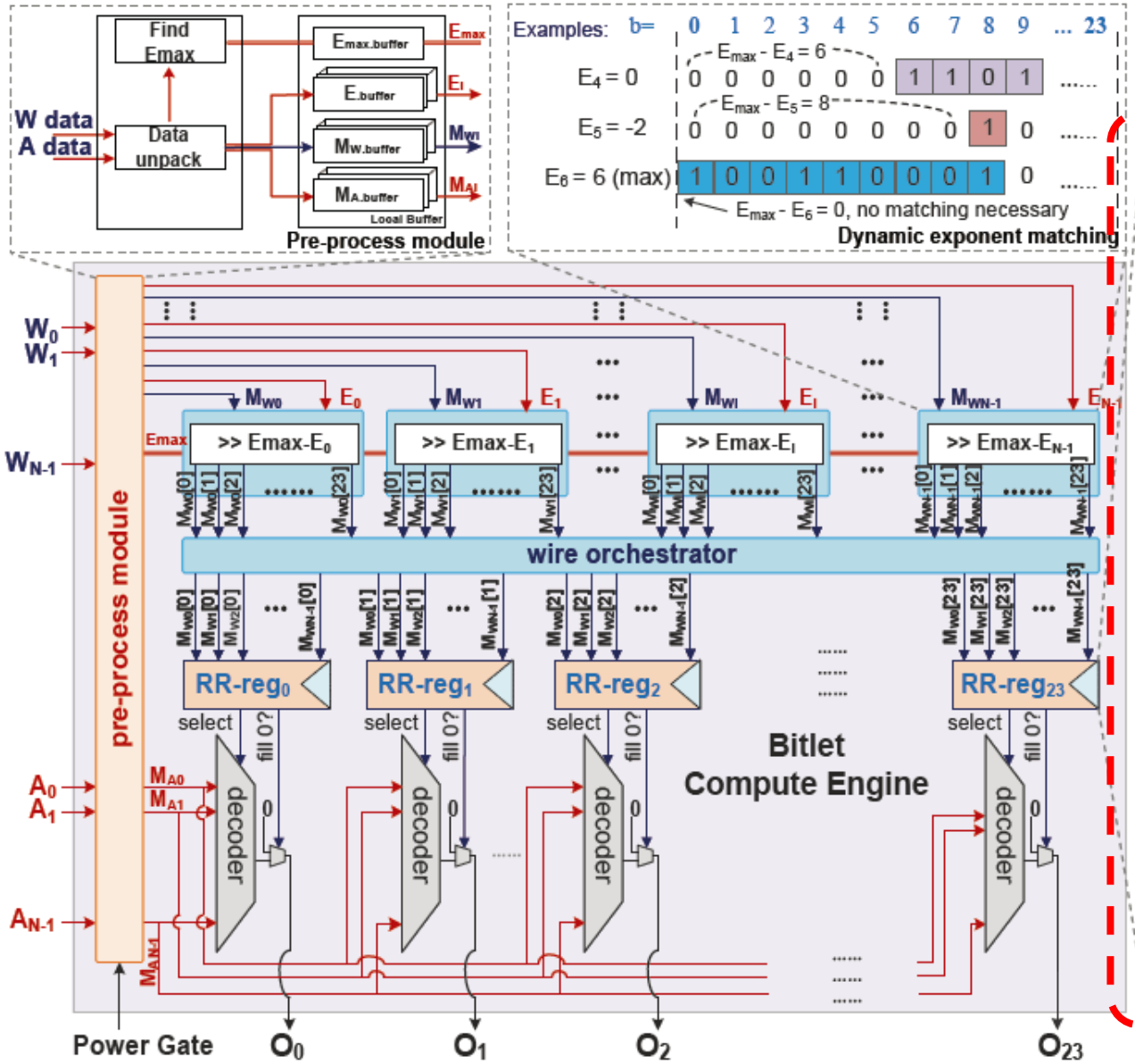


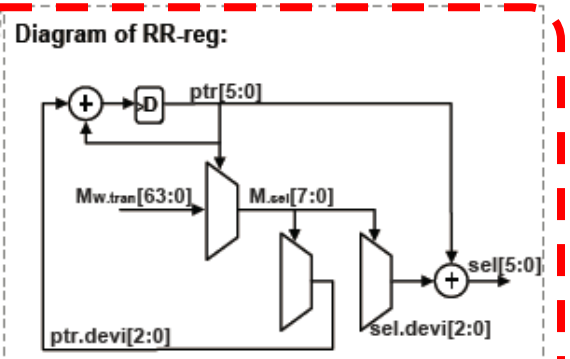
Logic in RR-reg:

```
i = 0, selector;
while(i<23):
    if  Mi[b] == 1
        selector.append[i]
if selector is empty:
    fill_0 = 1
else:
    item = selector.popfirst()
    select = item
    selector.delete[item]
    return select
```

Dynamic exponent matching: Emax is the maximum exponent, i.e. E6 in the above high level example

Methodology – bitlet accelerator



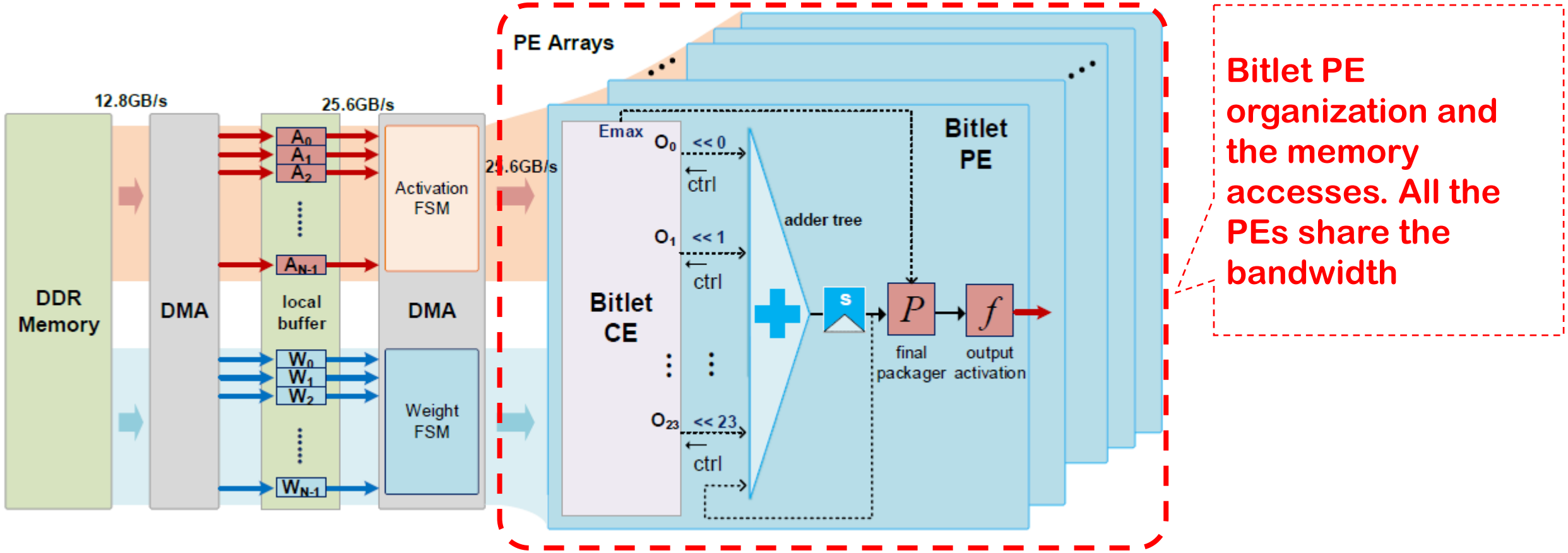


Logic in RR-reg:

```
i = 0, selector;
while(i<23):
    if  Mi[b] == 1
        selector.append[i]
if selector is empty:
    fill_0 = 1
else:
    item = selector.popfirst()
    select = item
    selector.delete[item]
    return select
```

Essential bit distillation: RR-reg is responsible for this operation, but the logic is very simple!

Methodology – bitlet accelerator



Bitlet PE organization and the memory accesses. All the PEs share the bandwidth

- Overall accelerator design :
  - Each bitlet PE is composed of one bitlet CE and subsequent adder tree for accumulation.
  - DDR3 is used on our FPGA Virtex-7 SoC platform.

OUTLINE

- Motivation & Background
- Methodology
- 

Evaluation
- Recap

Evaluation – Deep learning applications

We use various deep learning application datasets:

Models	Type	Precision	Domain	Dataset	GFLOPS	Weights	W-bit Sparsity (%)
ResNet-50[18]	2D Convolution	8 bit	Image Classification	ILSVRC'12[3]	8.21	25.56M	70.15 (fixed point)
MobileNetV2[35]	2D Convolution	8 bit	Image Classification	ILSVRC'12[3]	0.615	3.49M	76.85 (fixed point)
YoloV3[34]	2D Convolution	8 bit	Object Detection	CoCo[1]	25.42	61.95M	77.78 (fixed point)
Multi-Pose[24]	2D Convolution	8 bit	Pose Estimation	CoCo[1]	97.55	59.59M	66.33 (fixed point)
lapSRN[25]	2D De-Convolution	16 bit	Image Super Resolution	SET14[4]	736.73	0.87M	74.31 (fixed point)
DCPDNet[45]	Encoder -Decoder	16 bit	Deraining /Dehazing	NYU-Depth[38]	254.37	66.9M	75.00 (fixed point)
DenseNet-161[20]	2D Convolution	16 bit	Image Classification	ILSVRC'12[3]	15.56	28.68M	68.92 (fixed point)
FCOS[39]	Feature Pyramid	16 bit	Object Detection	CoCo[1]	80.14	32.02M	70.83 (fixed point)
CartoonGAN[11]	GAN	float 32	Style Transfer	flickr[2]	108.98	11.69M	48.49 (floating point)
Transformer[41]	Seq2Seq	float 32	Word Embedding	wmt'14[6]	10.6	176M	45.75 (floating point)
C3D[40]	3D Convolution	float 32	Video Understanding	UCF101[5]	38.57	78.41M	45.83 (floating point)
D3DNet[44]	3D Deformable	float 32	Video Super Resolution	Vimeo-90k[42]	408.82	2.58M	47.69 (floating point)

In order to prove the general-purpose feature of bitlet, we use 12 domain-specific DL tasks with 8 of them quantized to 16bit and int8.

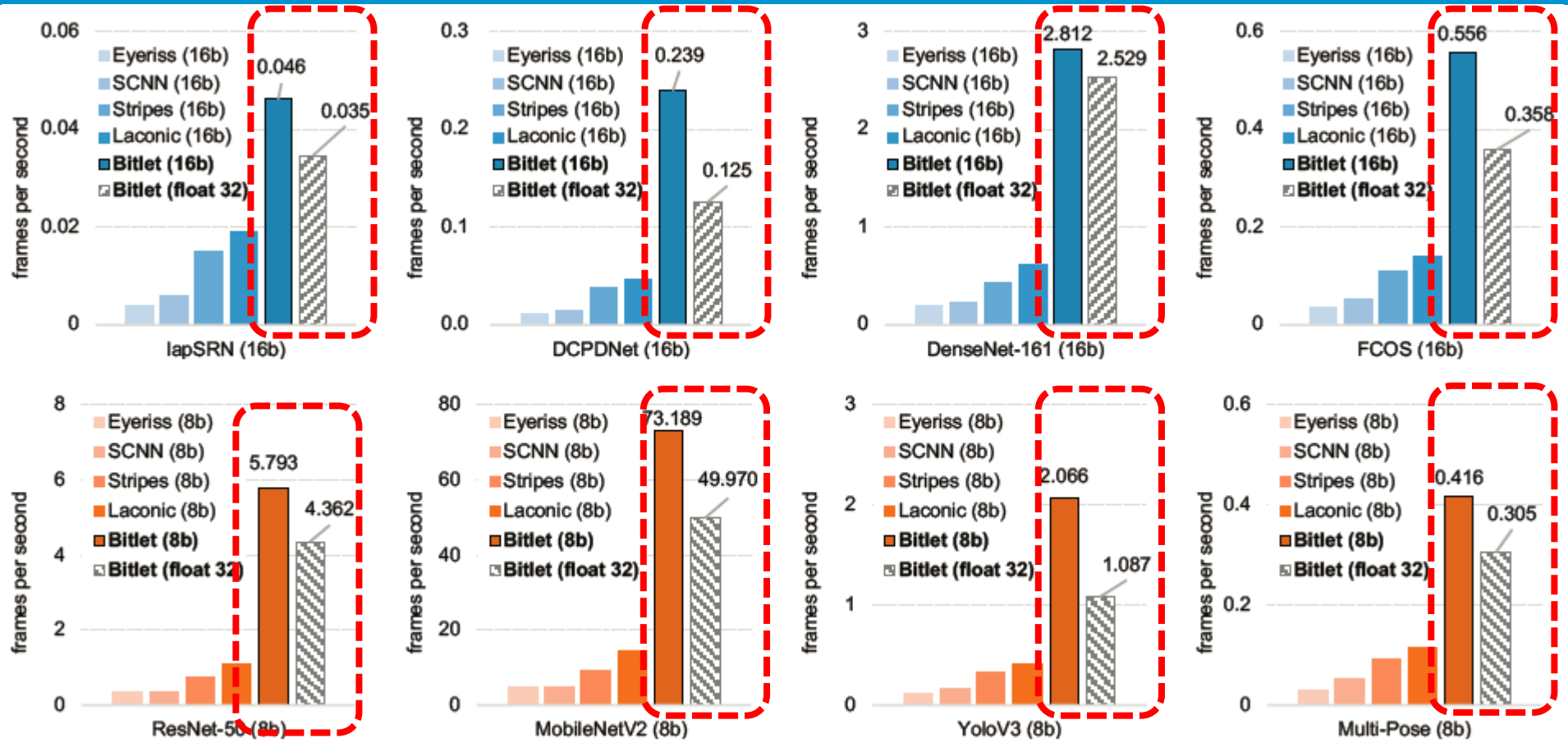
Evaluation – Specification

Overall accelerator Specs :

	Accelerator ASICs						GPUs		
Chip	Eyeriss [14]	SCNN [32]	Stripes [22]	Laconic [36]	**Bitlet (Ours)**		Titan V	Titan Xp	Tegra X2
PEs/Cores	168	64	4096	192	**32**		5120	3840	256
Precision	16b	16b	1~16b	1~16b	**fp32/16, 1~24b**		fp32/16, 8b	fp32, 8b	fp32/16
Technology	65nm TSMC	16nm TSMC	65nm TSMC	65nm TSMC	**28nm TSMC**	**65nm TSMC**	12nm TSMC	16nm TSMC	16nm TSMC
Freq. (MHz)	250	1000	980	1000	**1000**		1455	1582	854
PEAK Performance (GOPs)	23.1	2000	–	–	**204.8 (fp32) 372.35 (16b) 744.7 (8b)**		14900 (fp32) 29800 (fp16)	12150 (fp32)	750.1(fp32) 1330 (fp16)
Power	278mW	–	–	–	**570mW(fp32) 432mW(16b) 366mW(8b)**	**1829mW(fp32) 1390mW(16b) 1199mW(8b)**	250W	250W	15W
PEAK Power Efficiency (GOPs/W)	83.09	–	–	441 (16b) 805 (8b)	**359.15 (fp32) 667.97(16b) 1335.93 (8b)**	**111.97 (fp32) 267.87 (16b) 621.10 (8b)**	59.6(fp32) 119.2(fp16)	48.6 (fp32)	50.0 (fp32) 88.7(fp16)
Area (mm^2)	12.25	7.9	122.1	1.59	**1.54**	**5.80**	–	–	–

Compared with SOTA accelerators, bitlet supports floating-point arithmetic with higher efficiency (GOPs/W)

Evaluation – Speedup

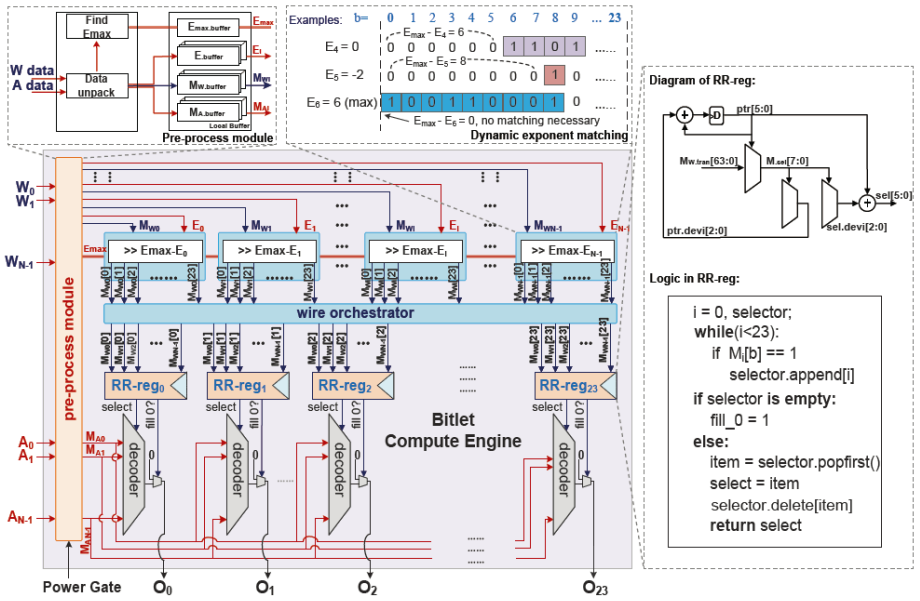


- We compare bitlet with fixed-point accelerators:
  - An interesting phenomenon is that bitlet-fp32 behaves even better than 16/8b fixed point accelerators!

Evaluation – Hardware Ablation Study

Instance	bare-m	*bitlet-fp32*			
Parameter	$N = 1$	$N = 32$		$N = 64$	
Ablation	w/o M w/o D	w/ M w/o D	w/ M w/ D	w/ M w/o D	w/ M w/ D
CartoonGAN	0.012	0.209	0.269	0.244	0.352
Transformer	0.126	2.152	2.879	2.509	3.763
C3D	0.035	0.590	0.771	0.670	0.976
D3DNet	0.003	0.056	0.068	0.065	0.084
Instance	bare-m	***Bitlet-16b***			
Parameter	$N = 1$	$N = 32$		$N = 64$	
Ablation	w/o M w/o D	w/o M w/o D	w/o M w/ D	w/o M w/o D	w/o M w/ D
lapSRN	0.004	0.033	0.040	0.038	0.046
DCPDNet	0.011	0.096	0.184	0.109	0.239
DenseNet-161	0.187	1.567	2.260	1.779	2.812
FCOS	0.036	0.304	0.455	0.345	0.556
Instance	bare-m	***Bitlet-8b***			
Parameter	$N = 1$	$N = 32$		$N = 64$	
Ablation	w/o M w/o D	w/o M w/o D	w/o M w/ D	w/o M w/o D	w/o M w/ D
ResNet-50	0.354	2.970	4.598	3.371	5.793
MobileNetV2	4.730	39.644	60.001	45.001	73.189
YoloV3	0.114	0.959	1.640	1.089	2.066
Multi-Pose	0.030	0.250	0.346	0.284	0.416

- hardware components' contribution to the performance
  - w/ or w/o M: with or without exponent Matching
  - w/ or w/o D: with or without bit Distilation



Evaluation – Area at different tech. nodes

Item	*Bitlet*(float 32) Area (mm^2)	*Bitlet*(float 32) Power (mW)	*Bitlet*(16b) Power (mW)	*Bitlet*(8b) Power (mW)
Preprocess-ing Module	1.916 (33%)	1208.5 (66.1%)	1000.8 (71.9%)	1000.8 (83.5%)
Wire Orch. & Decoder	2.327 (40.1%)	164.1 (8.1%)	111.4 (8.0%)	55.7 (4.6%)
RR-Reg & Check Win.	0.7 (12.0%)	112.1 (7.2%)	74.8 (5.3%)	37.4 (2.9%)
Adder Tree	0.7 (12.0%)	293.3 (16.0%)	195.5 (14.1%)	97.8 (9.8%)
PostProcess-ing Module	0.2 (2.9%)	48.5 (2.7%)	7.4 (0.5%)	7.4 (0.6%)
Total	**5.8**	**1829.6**	**1390.0**	**1199.1**

TSMC 65nm

Item	*Bitlet*(float 32) Area (mm^2)	*Bitlet*(float 32) Power (mW)	*Bitlet*(16b) Power (mW)	*Bitlet*(8b) Power (mW)
Preprocess-ing Module	0.553 (35.8%)	356.8 (62.6%)	296.598 (68.6%)	296.598 (81.2%)
Wire Orch. & Decoder	0.570 (35.9%)	63.857 (11.2%)	40.28 (9.73%)	20.651 (5.54%)
RR-Reg & Check Win.	0.131 (9.6%)	27.37 (4.8%)	20.28 (4.56%)	10.128 (2.88%)
Adder Tree	0.244 (15.8%)	107.424 (18.8%)	71.616 (16.6%)	35.808 (9.80%)
PostProcess-ing Module	0.044 (2.9%)	14.88 (2.6%)	2.304 (0.53%)	2.304 (0.63%)
Total	**1.542**	**570.15**	**432.08**	**365.49**

TSMC 28nm

OUTLINE

- Motivation & Background
- Methodology
- Evaluation
- **Recap**

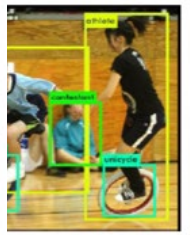
Recap

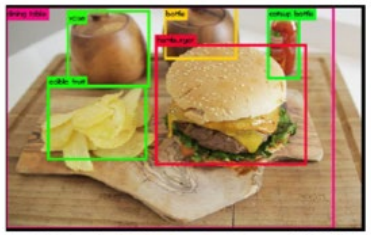
The contribution of this work

1. Propose a novel philosophy of leveraging bit-level sparsity – bit interleaving
2. Propose a specialized general-purpose accelerator – bitlet, to mine the maximum potential of bit interleaving

① General purpose	Leveraging the sparsity for accelerating both training and inference
② Multi-precision support	Floating point fp32/16, fixed point from 1b~24b
③ High performance and efficiency	Up to 15× and 81× speedup over GPUs

Designed for General-purpose Deep Learning Applications, and what's more?





Redmon, J. & Farhadi, A., YOLO9000: Better, Faster, Stronger. CVPR 2017.

Thanks for listening!
Q & A

路航[1], 常亮[2], 李成龙[2], 竹子轩[2], 鲁圣健[1], 刘艳欢[1] , 张明喆[3]

[1]State Key Laboratory of Computer Architecture, Institute of Computing Technology, CAS, Beijing, China

[2]University of Electronic Science and Technology of China, Chengdu, China

[3]State Key Laboratory of Information Security, Institute of Information, CAS, Beijing, China

计算机体系结构国家重点实验室
State Key Laboratory of Computer Architecture, ICT, CAS

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

电子科技大学

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING, CAS

Backup Slides

计算机体系结构国家重点实验室
State Key Laboratory of Computer Architecture, ICT, CAS

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

电子科技大学

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING, CAS

Evaluation – Efficiency

- We compare bitlet with GPUs for both training and inference:
  - Due to the general-purpose characteristic, bitlet could also accelerate the forward propagation in training.

GPU Baselines	Models (**Train** / **Inference** efficiency in GOPs/W)			
	Cartoon -GAN	Trans -former	C3D	D3DNet
Titan V	0.27 / 3.19	4.09 / 21.80	1.82 / 4.32	0.06 / 4.67
Titan Xp	0.20 / 2.36	3.03 / 16.13	1.35 / 3.19	0.04 / 3.46
Jetson TX2	-- / 0.20	-- / 1.39	-- / 0.27	-- / 0.30
***Bitlet* (float 32)**	9.40 / 67.29	7.36 / 69.97	8.59 / 66.04	4.87 / 60.24

Evaluation – Efficiency

- For the accelerators, we only evaluate the fixed point quantized tasks

Accelerators Baselines	**Inference** efficiency is in GOPs/W			
	lapSRN	DCPDNet	DenseNet -161	FCOS
Eyeriss (16b)	9.92	9.42	9.79	9.71
SCNN (16b)	24.29	20.96	19.49	23.77
Stripes (16b)	25.06	21.34	15.03	19.63
Laconic (16b)	46.04	39.32	31.27	36.64
***Bitlet* (16b)**	168.75	302.71	217.87	221.87
***Bitlet* (float 32)**	44.64	55.82	69.03	50.33

Accelerators Baselines	**Inference** efficiency is in GOPs/W			
	ResNet-50	Mobile -NetV2	YoloV3	Multi -Pose
Eyeriss (8b)	9.79	9.80	9.76	9.85
SCNN (8b)	15.97	15.98	23.88	27.87
Stripes (8b)	14.32	13.03	18.91	20.57
Laconic (8b)	29.60	29.00	35.29	36.58
***Bitlet* (8b)**	236.82	224.13	261.50	202.07
***Bitlet* (float 32)**	62.83	53.92	48.46	52.24

Evaluation – Sensitivity

- Two design parameters are evaluated
  - Number of simultaneous input of the bitlet compute engine -- N
  - PE numbers