KU-The Future

FINEA: An Efficient Neural Network Accelerator Exploiting Factorized Input Features

Yujin Kim*, Chanhun Jeong*,
Yunho Oh*, Myung Kuk Yoon†, Gunjae Koo*

*Korea University

†Ewha Womans University

KOREA UNIVERSITY

EWHA WOMANS UNIVERSITY

The Limits of AI Accelerators

1. Large model size: growing compute and memory demands
2. Underutilized hardware: low efficiency due to irregular workloads

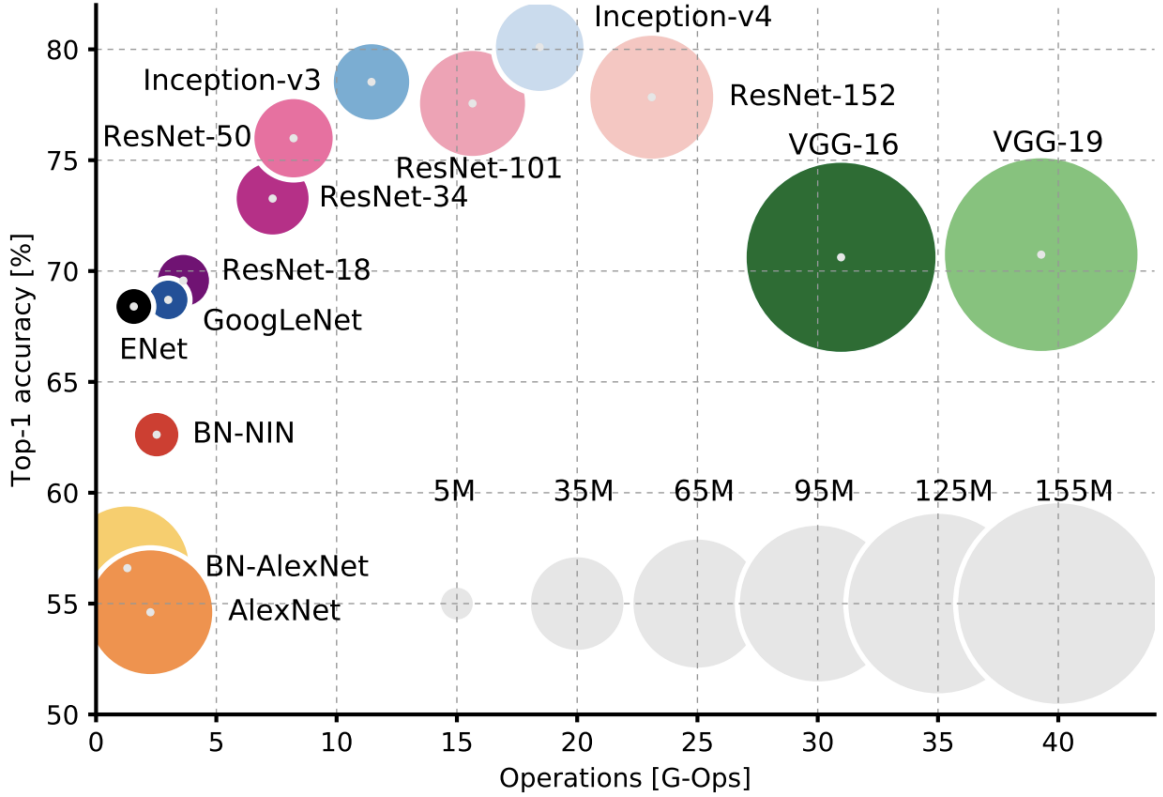


*Figure adapted from: Google Cloud Team, *"Google Trillium: Our Next-Generation TPU,"* Google Cloud Blog, May 2024.

Quantization for DNN models

As DNNs scale up, quantization has become essential for inference

Quantization: FP are mapped into discrete INT ranges

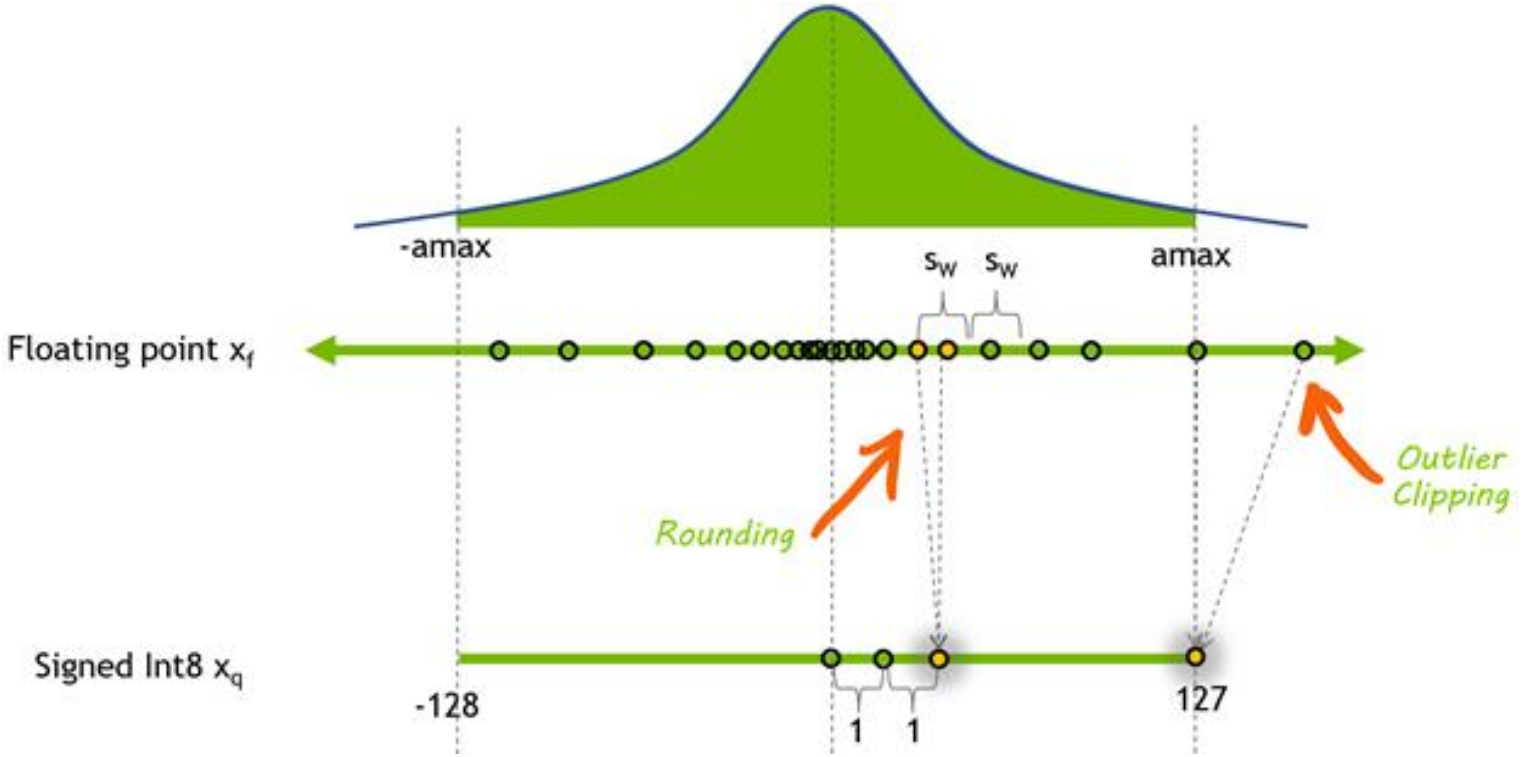


*Figure adapted from: M. Grootendorst, "A Visual Guide to Quantization," Exploring Language Models, July 22, 2024. © 2024 Maarten Grootendorst

Underutilization in Systolic Arrays

Poor utilization in systolic arrays: 21.7%

- ***Dimension mismatch***
- ***Interconnect bottleneck***
- ***Limited parallelism***

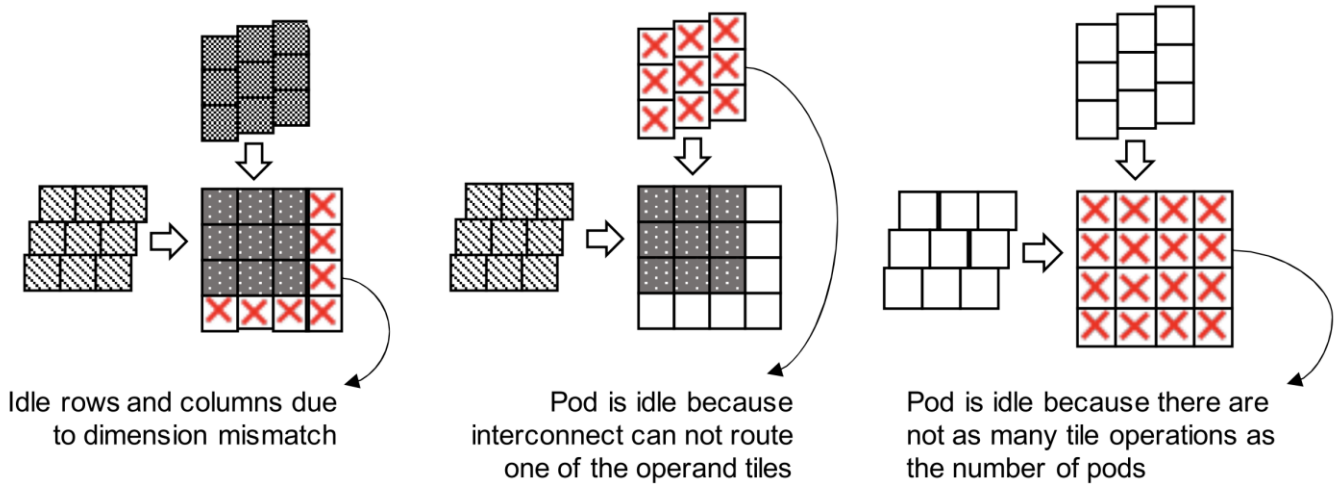


*Figure adapted from: Yüzügüler, A. C., Sönmez, C., Drumond, M., Oh, Y., et al., "Scale-out Systolic Arrays," *ACM TACO*, vol. 20, no. 2, 2023, DOI:10.1145/3572917

Weight Duplication

High weight duplication in quantized DNNs: 70.4%

- ***Quantization narrows value range,***

Input factorization based on weight duplication

A	B	B
A	E	F
A	E	I

$$\frac{\sum_{w \in W}(c(w)-1)}{n} = \frac{(3-1)+(2-1)+(2-1)+(1-1)+(1-1)}{9} = 44.4\%$$

n : The number of parameters in a convolution filter

$(c(w)-1)$: A count of the duplicated weight values

Factorization in a Convolution Filter

Input feature map 3 x 3

A	B	C
D	E	F
G	H	I

Filter 2 x 2

1	2
1	0

Original dot-product

$$= 1 \times A + 2 \times B + 1 \times D + 0 \times E$$

Factorization in a Convolution Filter

Input feature map 3 x 3

A	B	C
D	E	F
G	H	I

Filter 2 x 2

1	2
1	0

Original dot-product

$$= 1 \times A + 2 \times B + 1 \times D + 0 \times E$$

Factored dot-product

$$= 1 \times (A + D) + 2 \times B$$

Reduces redundant multiplications

Decreases computation energy and inference latency

Improves hardware utilization efficiency

Factorization in a Convolution Filter

Input feature map 3 x 3

A	B	C
D	E	F
G	H	I

Filter 2 x 2

1	2
1	0

Original dot-product

$$= 1 \times A + 2 \times B + 1 \times D + 0 \times E$$

Factored dot-product

$$= 1 \times (A + D) + 2 \times B$$

Identify duplicated weights and their positions for factored dot-products

Weight Tables

✓ **Build Factored /Unfactored weight tables in preprocessing**

- **F-WTB holds 4 indices and U-WTB holds 1**

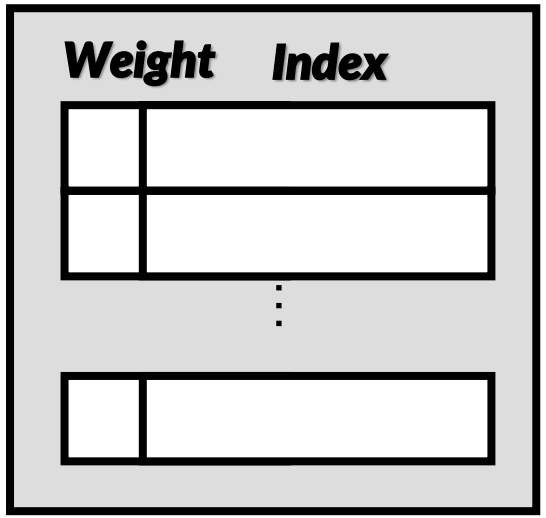
Input feature map 4 x 4

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

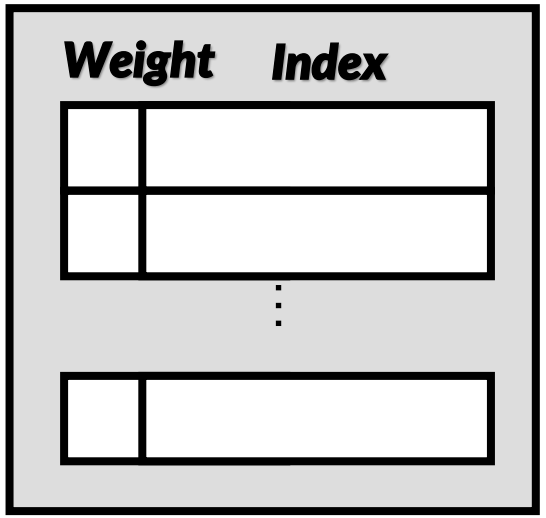
Filter 3 x 3

1	2	2
3	2	0
0	2	3

Factored weight table



Unfactored weight table



Weight Tables

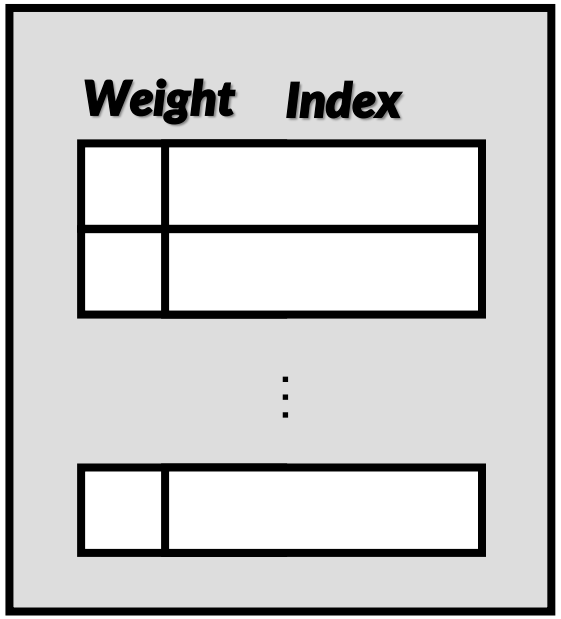
✓ Th_{fact} **(Factorization threshold) is determined by the ratio of multiplier counts between factored and unfactored PEs.**

Filter 3 x 3

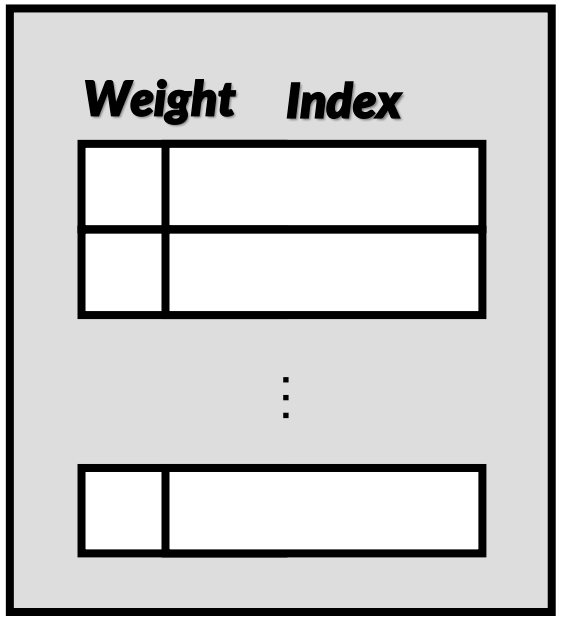
1	2	2
3	2	0
0	2	3

Filter weight	Index
1	0
2	1, 2, 4, 7
3	3, 8

Factored weight table



Unfactored weight table

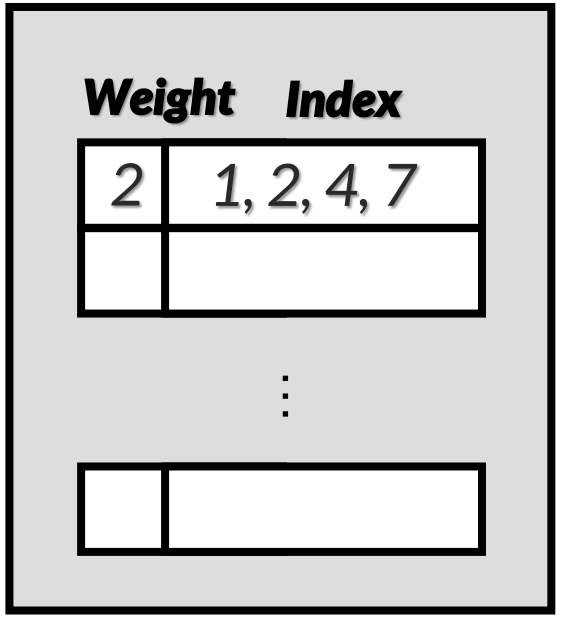


Weight Tables

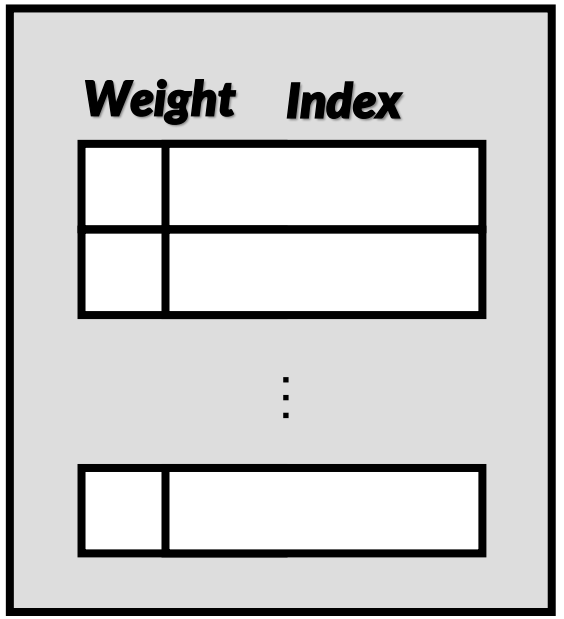
✓ ***Weights with high duplication are assigned to the factored weight table.***

Filter weight	Index
1	0
2	1, 2, 4, 7
3	3, 8

Factored weight table



Unfactored weight table



Weight Tables

- ***Weights are assigned to the factored and unfactored tables considering their processing time.***

Filter weight	Index
1	0
3	3, 8

Factored weight table

PSID 0

Weight	Index
2	1, 2, 4, 7
⋮	

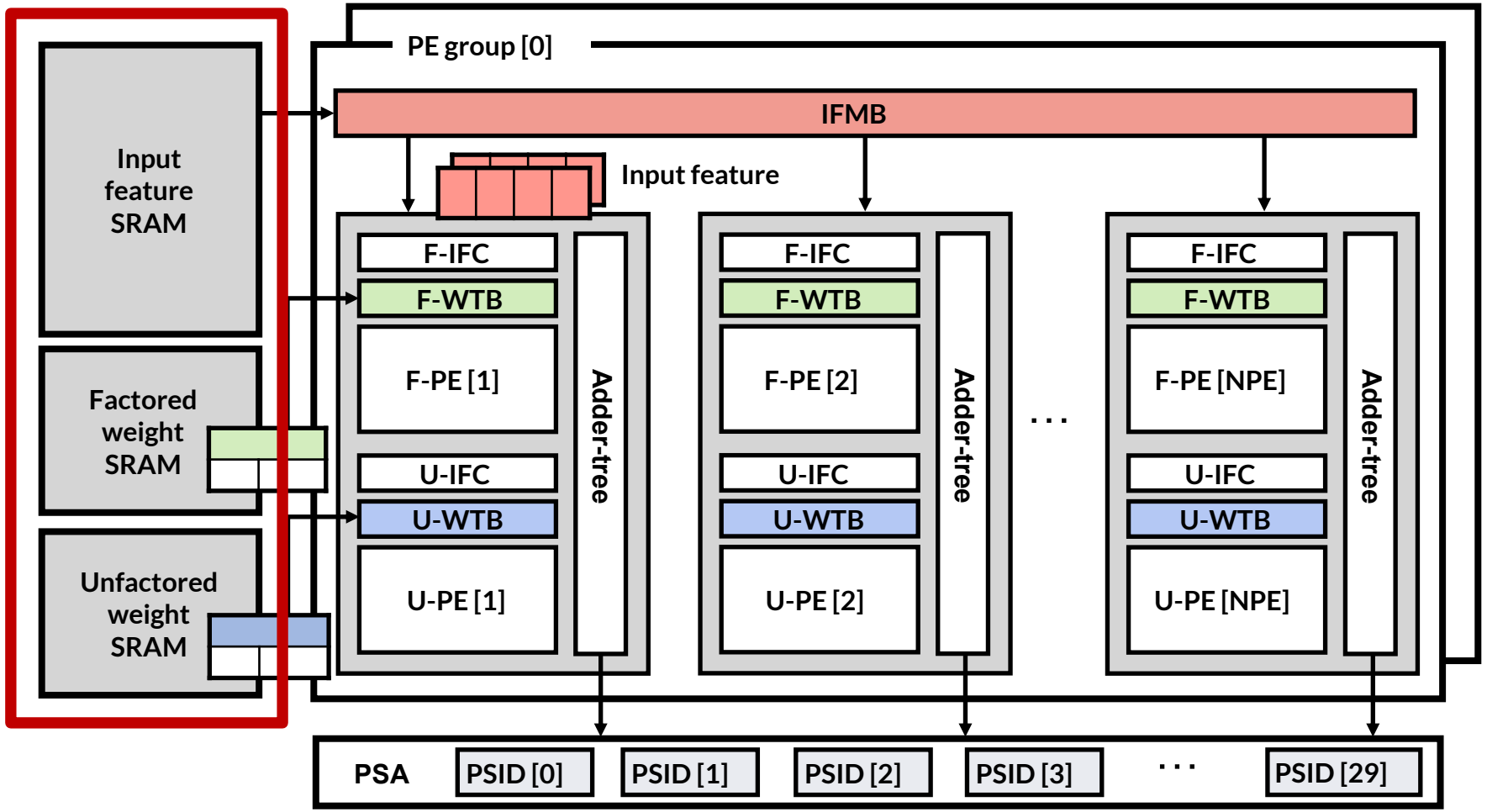
Unfactored weight table

PSID 0

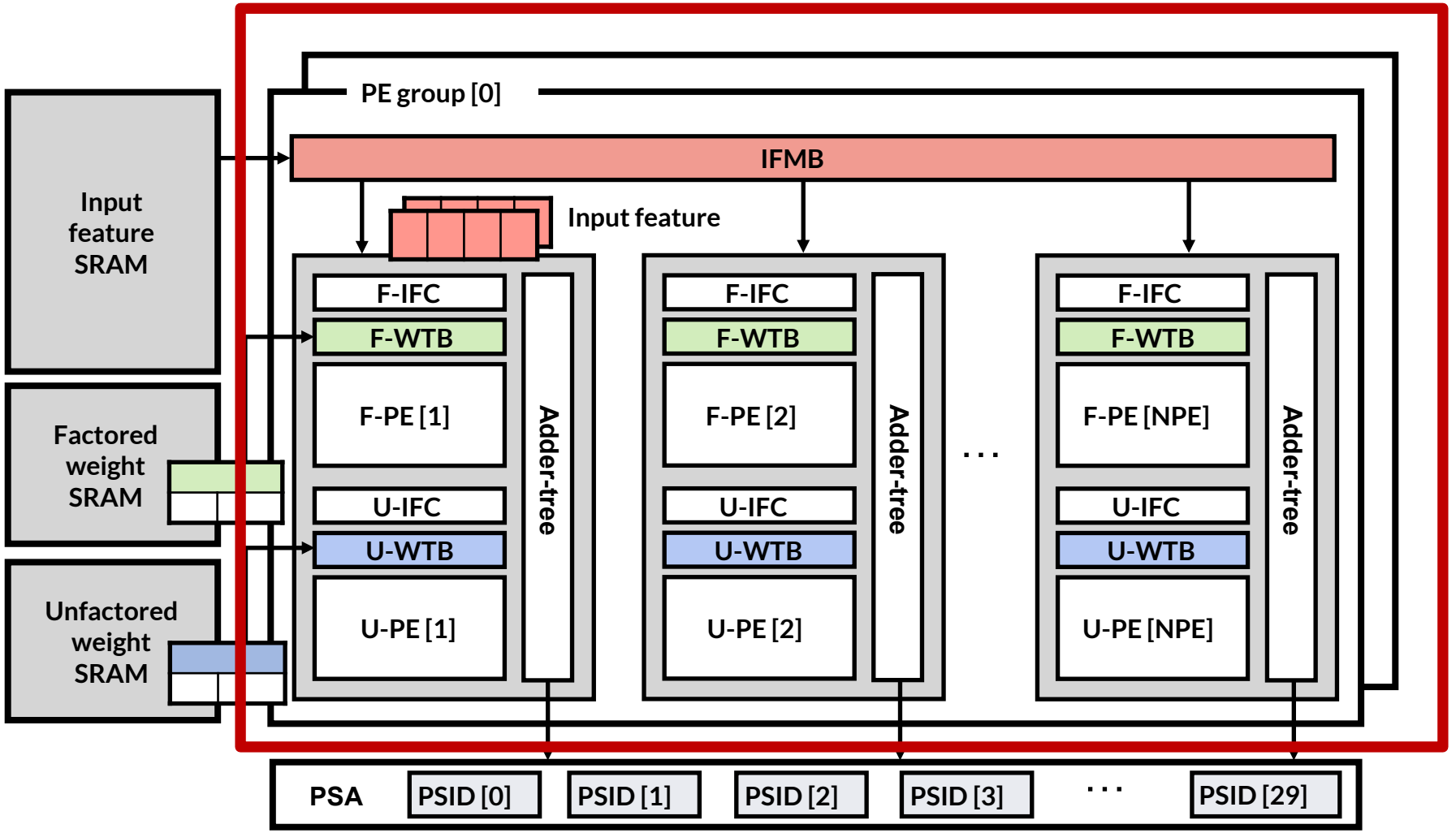
Weight	Index
3	3, 8
1	0
⋮	

PSID is filter number for partial-sum accumulation in PSA

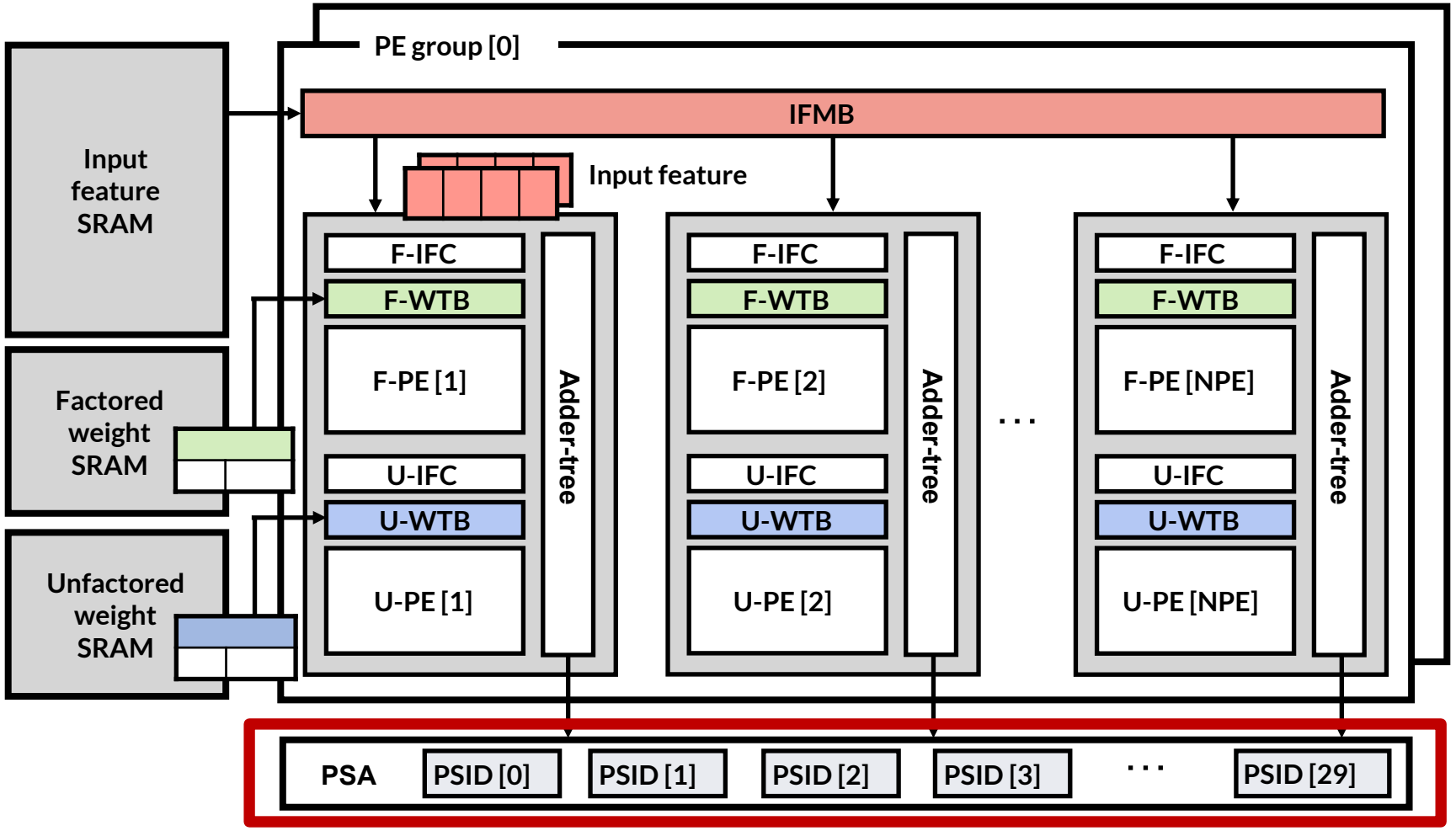
FINEA: Factorized Input-based Neural Engine Architecture



FINEA: Factorized Input-based Neural Engine Architecture



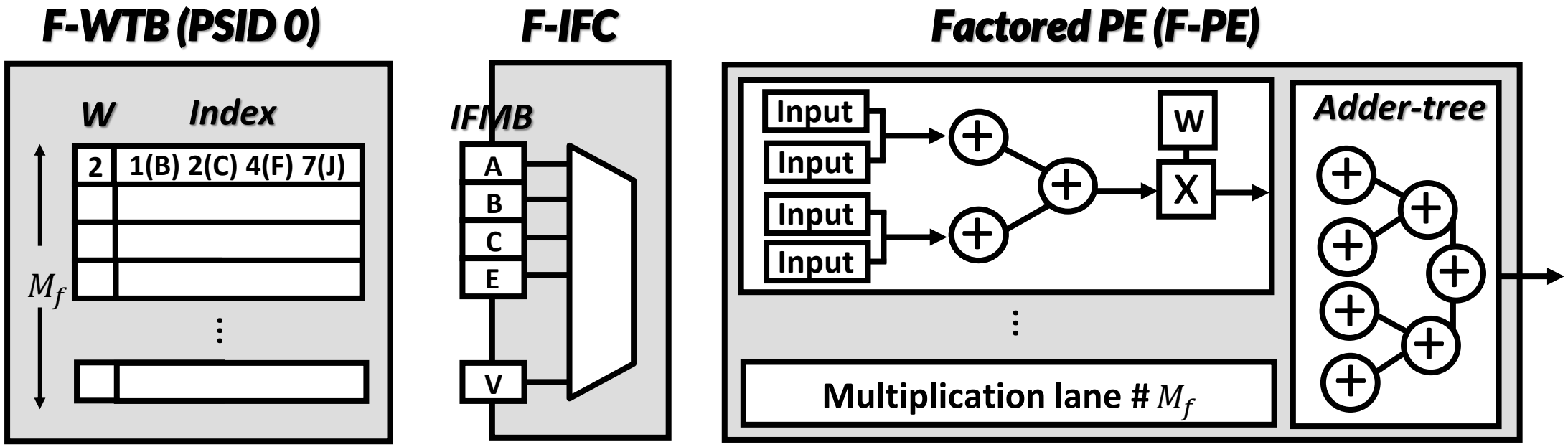
FINEA: Factorized Input-based Neural Engine Architecture



Factored dot-product

✓ **F-PE structure overview**

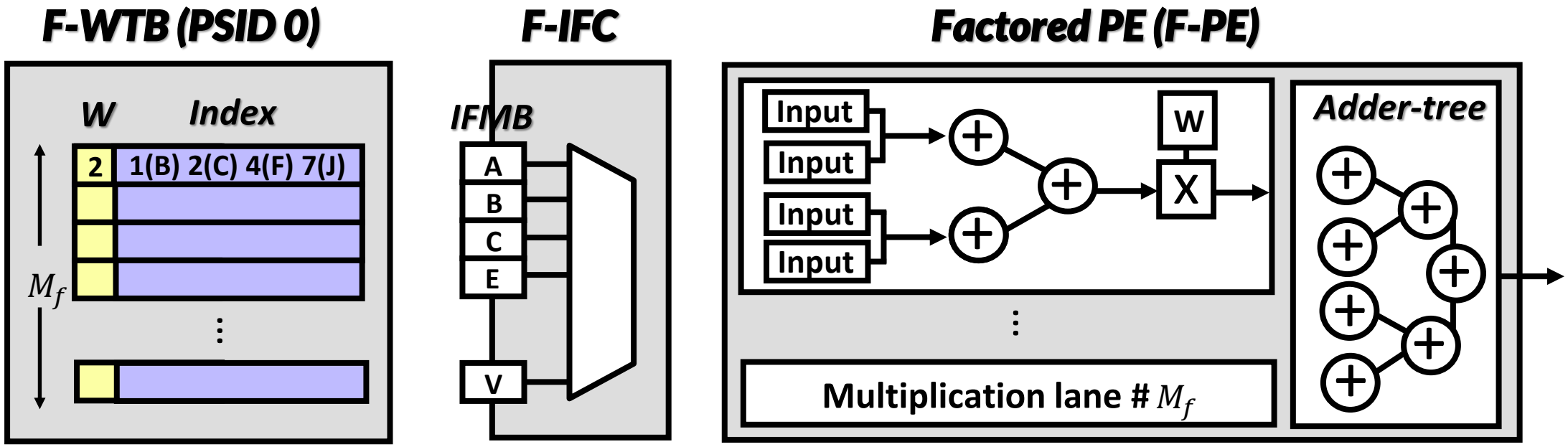
- **The Factored Processing Engine consists of F-WTB, F-IFC, and F-PE**



Factored dot-product

✓ **F-WTB (Factored Weight Table Buffer)**

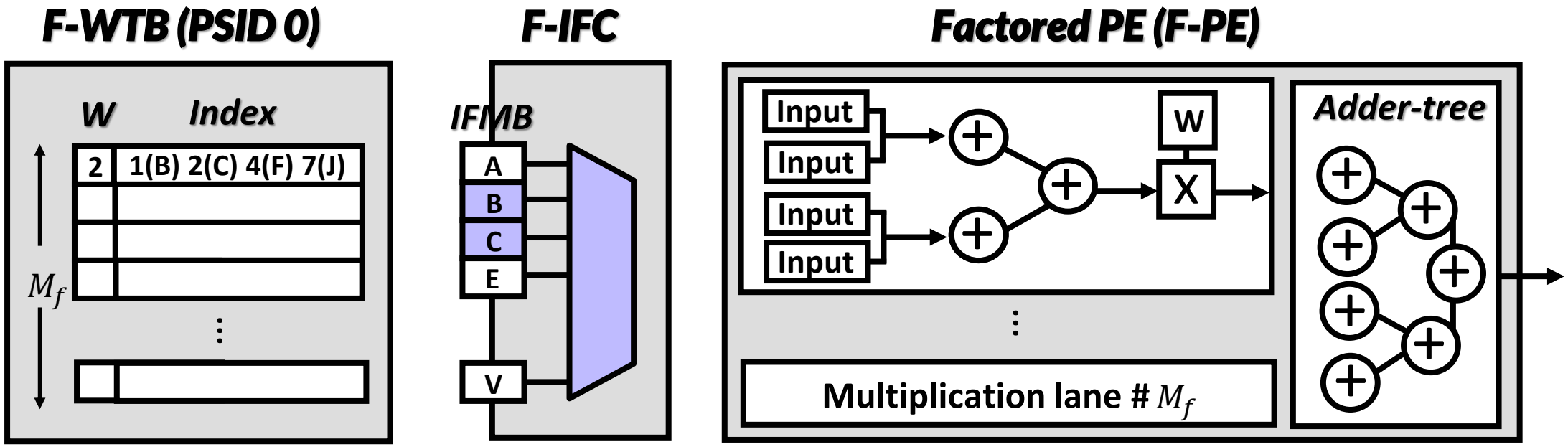
- **Stores the factored weight table.**



Factored dot-product

✓ **F-IFC (Factored-Input Feature Collector)**

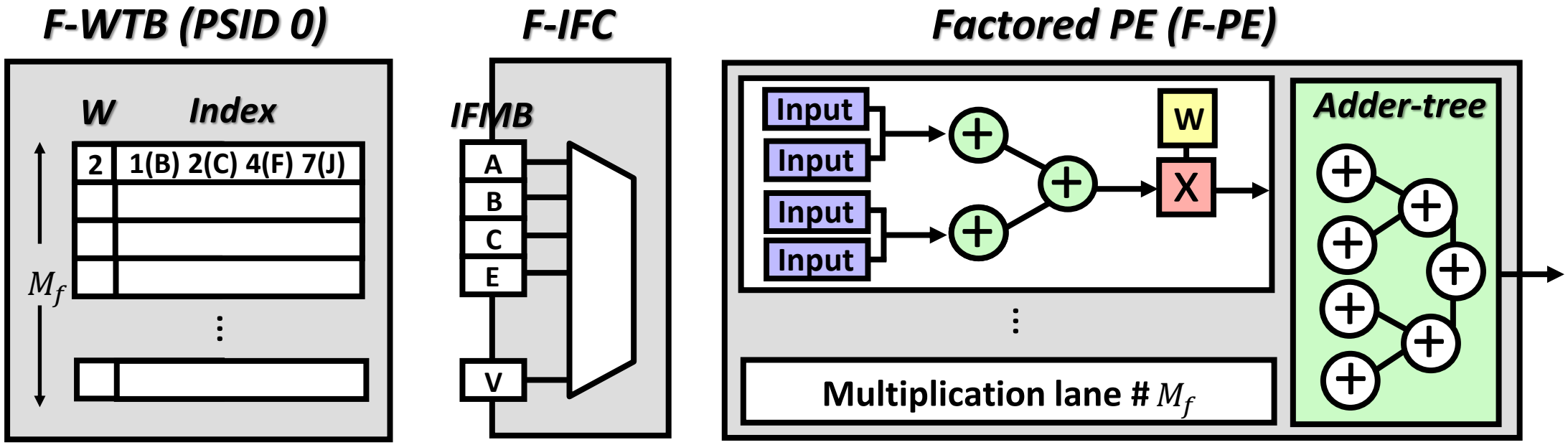
- Collects input values from the IFMB using the indices of the weight table buffer



Factored dot-product

✓ **F-PE (Factored-Processing Engine)**

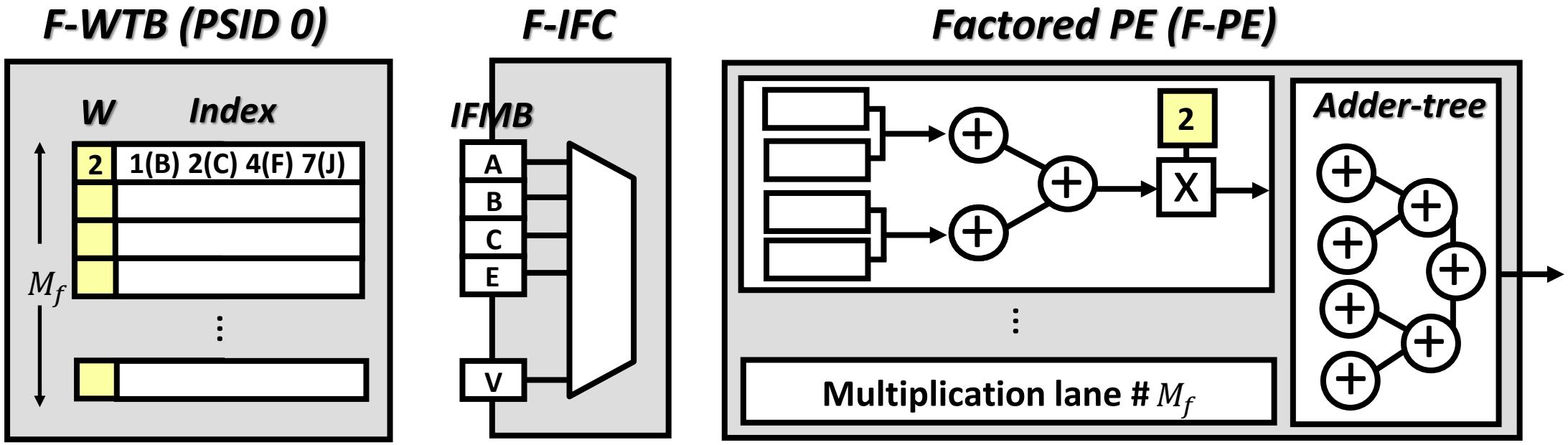
- An engine that performs factorized dot-product operations.



Factored dot-product

✓ **Weight loading**

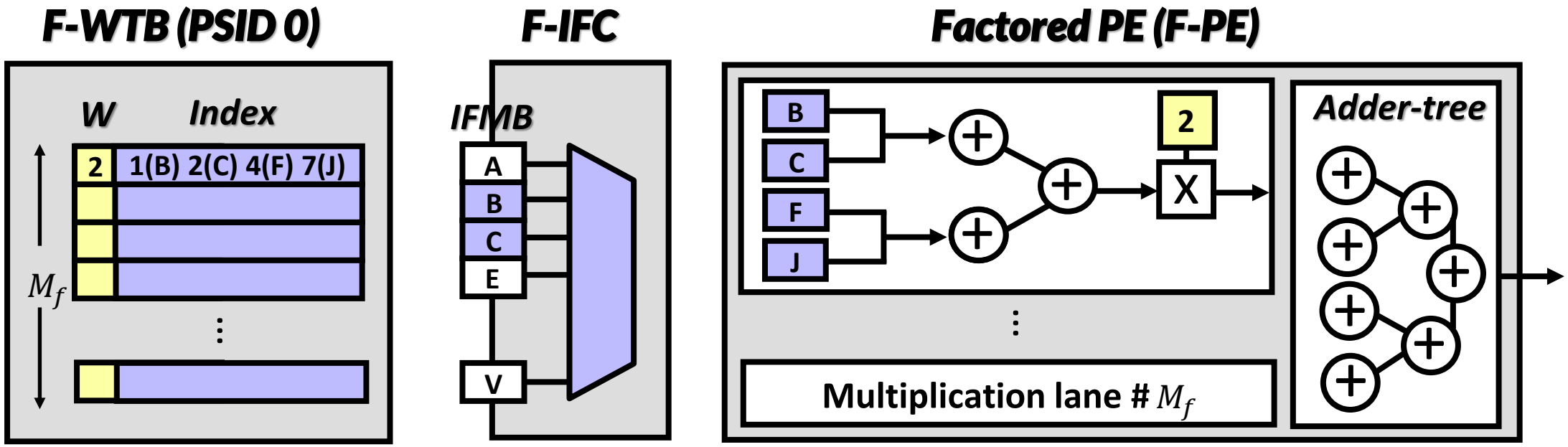
- Weights stored in the F-WTB are allocated to the F-PE for factored computation.



Factored dot-product

✓ ***Indexed input fetching***

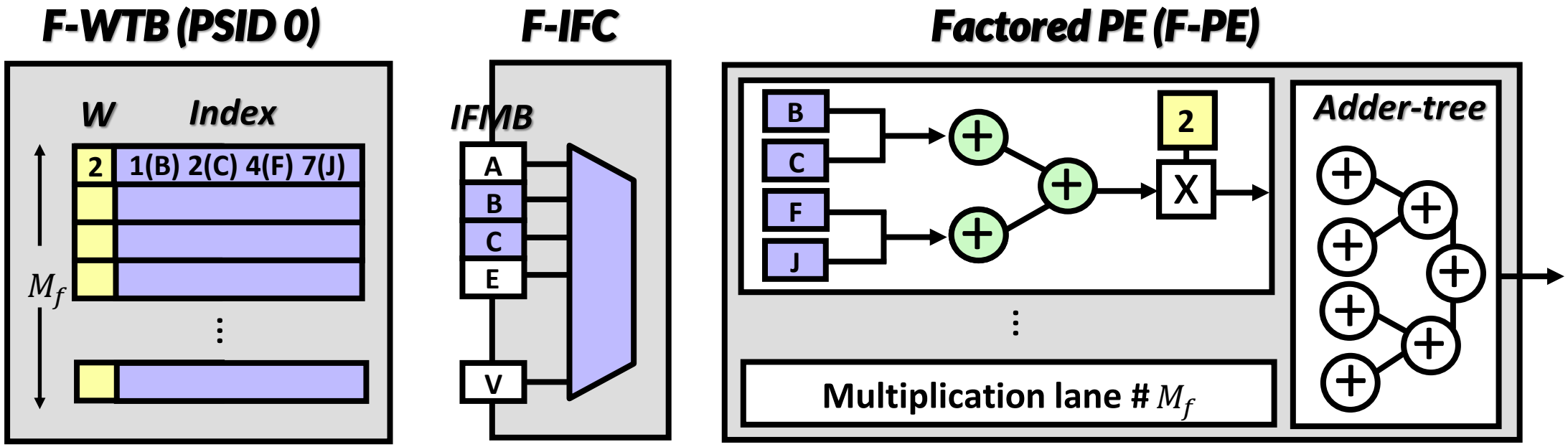
- ***F-IFC fetches input features from IFMB based on the indexes in F-WTB.***



Factored dot-product

✓ ***Input factorization***

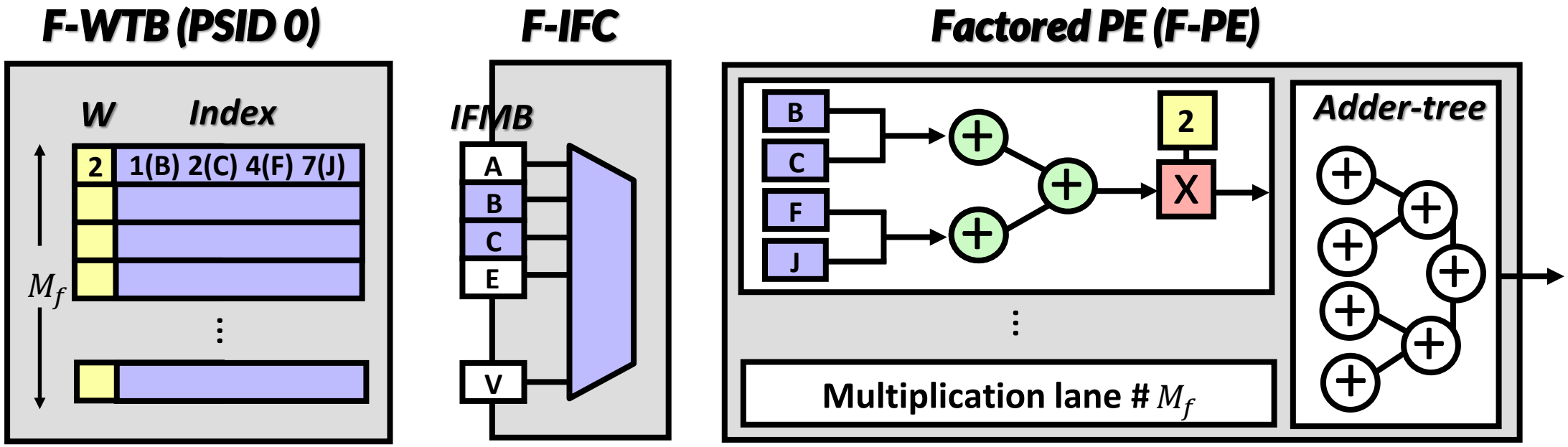
- ***Fetched input features are summed through the adder-tree for input factorization.***



Factored dot-product

✓ **Weighted multiplication**

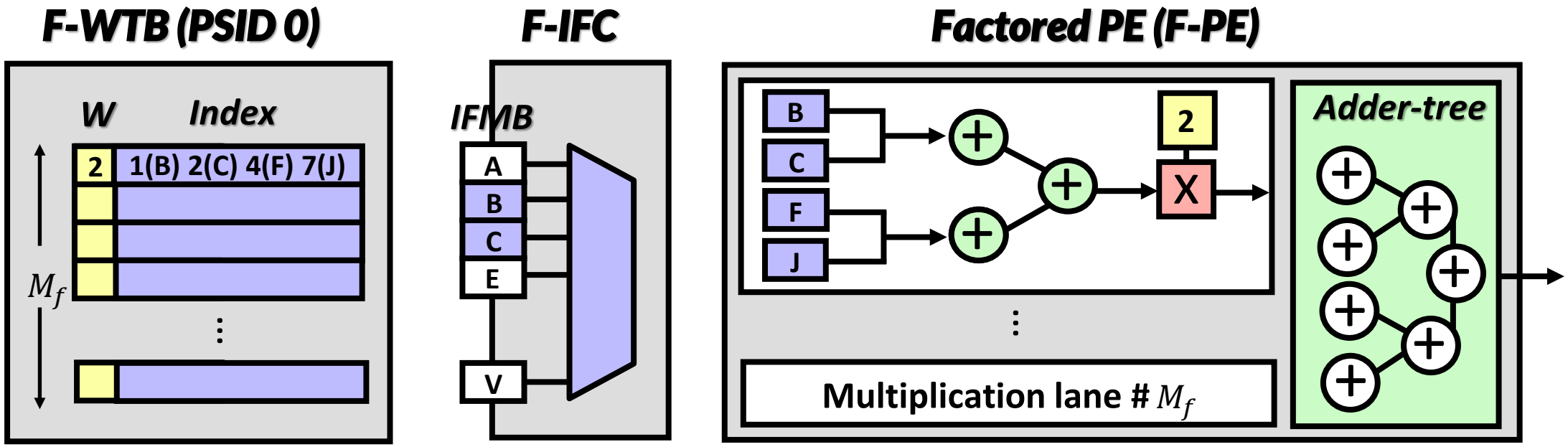
- The factorized input is multiplied by its corresponding unique weight.



Factored dot-product

✓ **Result accumulation**

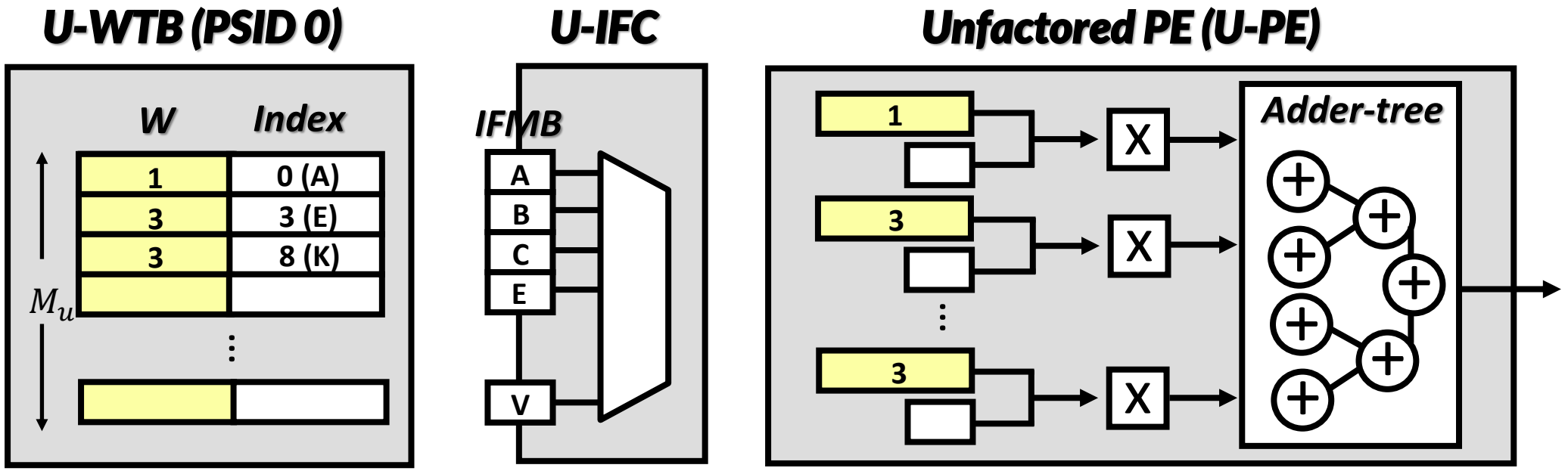
- Partial products from all multiplication lanes are accumulated through the adder-tree.



Unfactored dot-product

✓ **Weight loading**

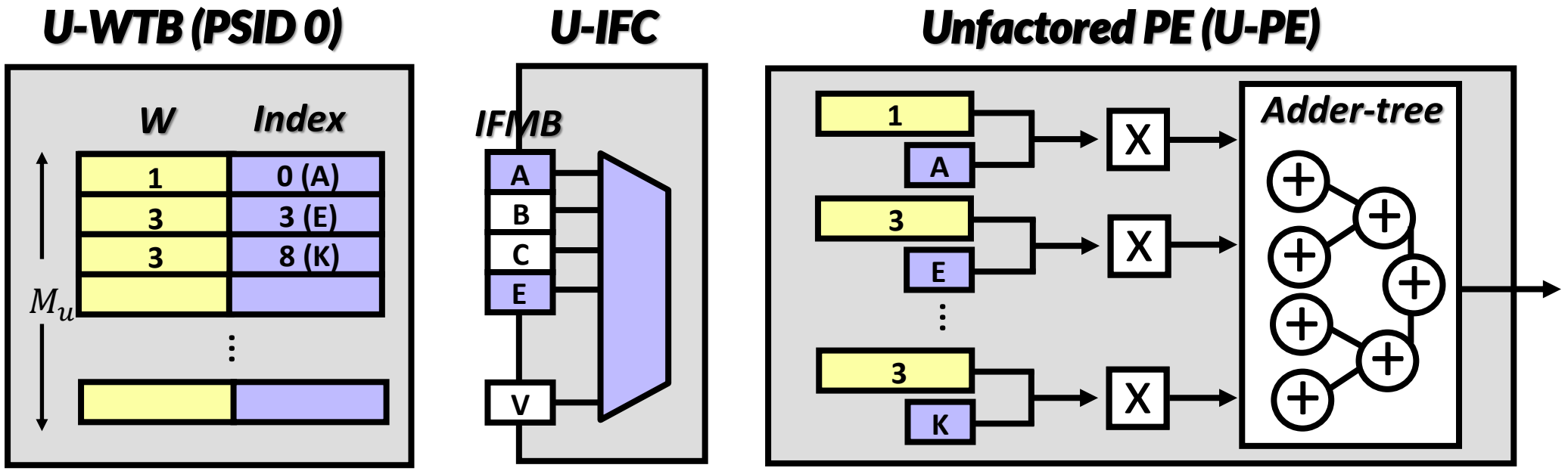
- Weights stored in the U-WTB are mapped to the corresponding multiplier in the U-PE.



Unfactored dot-product

✓ ***Indexed input fetching***

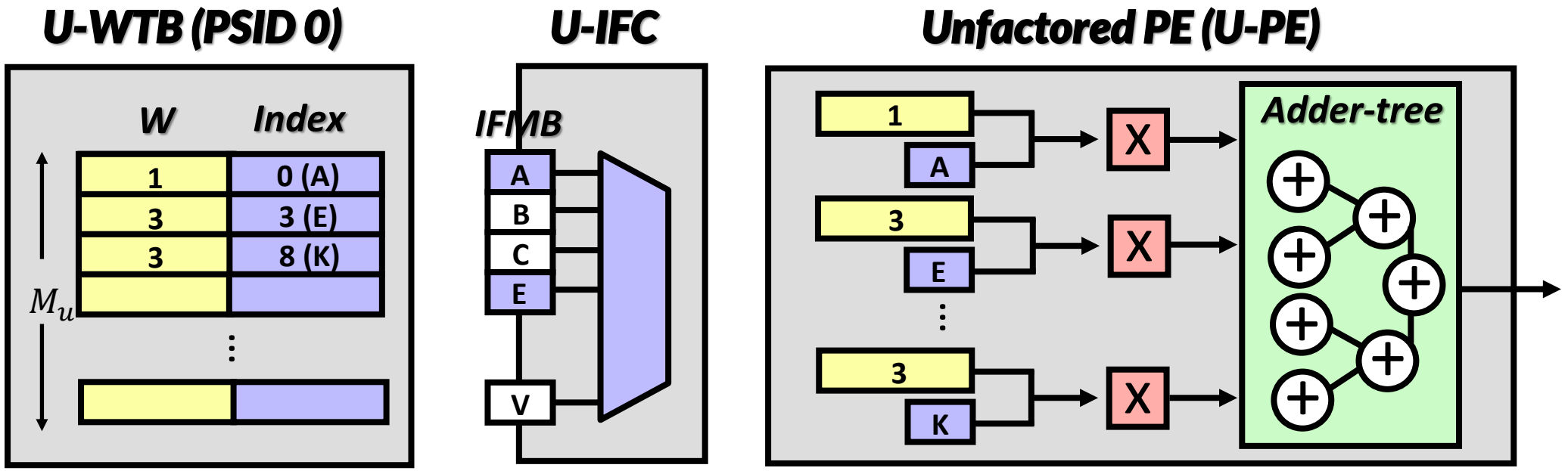
- ***U-IFC fetches input features from IFMB using the indexes in the U-WTB entries.***



Unfactored dot-product

✓ **Multiplication & Accumulation**

- Each multiplier multiplies its input and weight.
- Partial products from all multiplier are accumulated



Evaluation

Simulation and Implementation

- Implemented cycle-accurate simulator and Verilog HDL model of FINEA.

- Synthesized at 1GHz using 7nm standard-cell library to estimate power and energy.

TABLE I: Default FINEA configuration

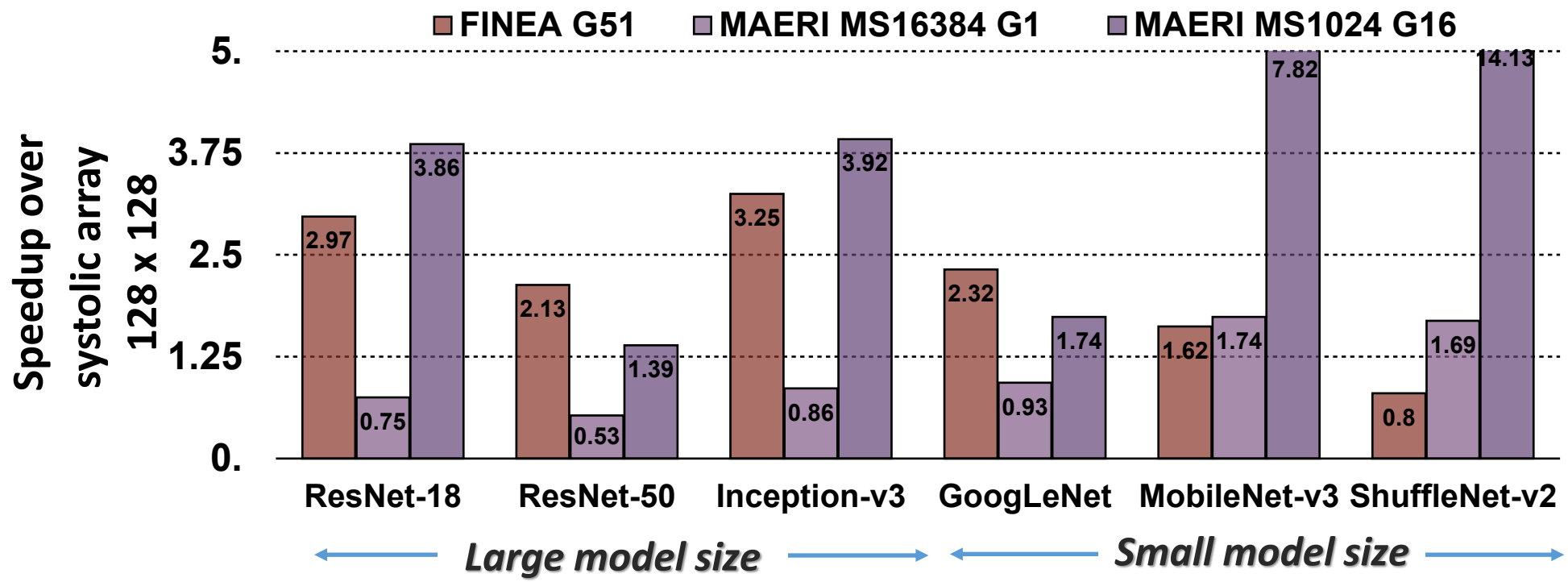
	Small	Medium	Large
Factored input slots per lane (N_a)	4	4	4
No. of multiplier lanes in F-PE (M_f)	8	8	8
No. of multiplier lanes in U-PE (M_u)	32	32	32
No. of PEs per PE group (N_{PE})	8	8	8
No. of PE groups (G)†	3	12	51
Input feature SRAM (KB)	5.5	5.5	5.5
Factored weight SRAM (KB)	27.5	27.5	27.5
Unfactored weight SRAM (KB)	11	11	11

† G is chosen so that G×$(M_f + M_u)$×N_{PE} matches the total multiplier count of each systolic array.

Performance: Speedup over Baseline

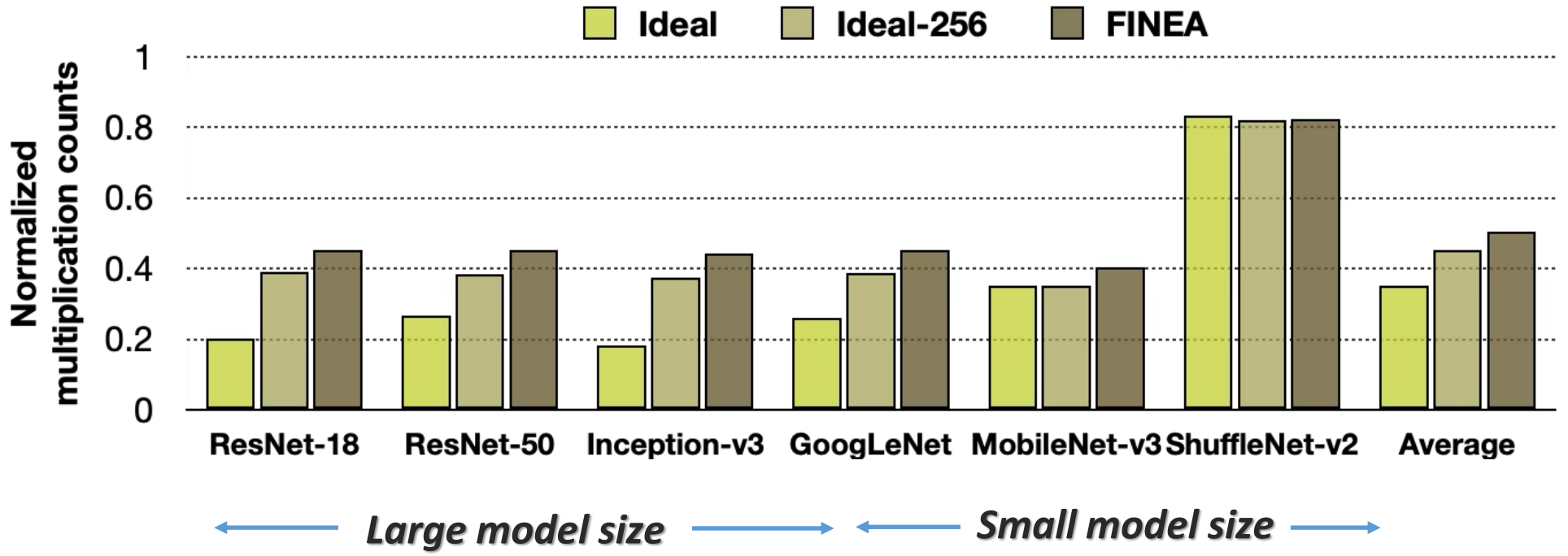
✓ ***FINEA achieves up to 2.18× speedup over systolic arrays***

- ***Higher speedups for large DNNs with more weight redundancy.***



Computation Reduction

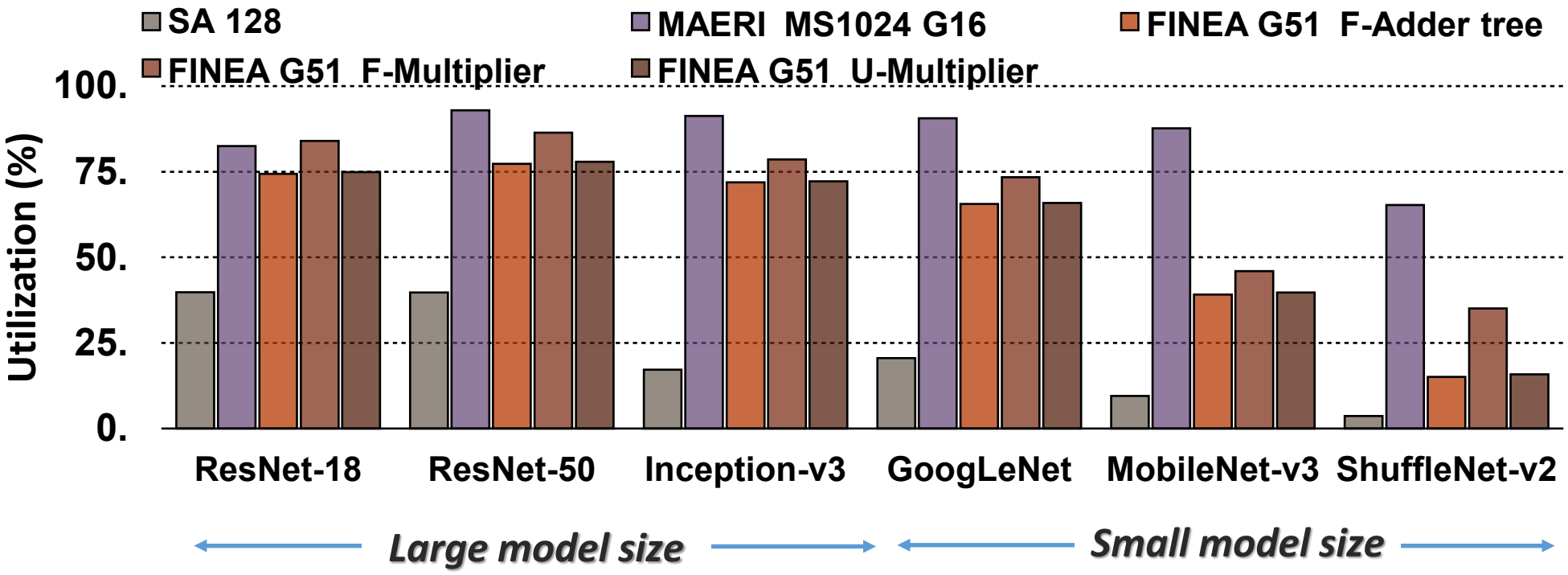
✓ ***FINEA reduces multiplication counts: 51%***



Additional Evaluations in Paper

Utilization of compute units

- F-Adder tree shows 60% average utilization, reaching up to 72% for large DNNs.



Conclusion

- ***Exploits redundant weights within convolution filters for computation reduction.***

- ***Introduces factored (F-PE) and unfactored (U-PE) processing engines with shared weight tables.***
- ***Achieves 51% reduction in multiplications and x2.97 performance improvement over systolic arrays.***

Thank you

Questions?

✉ yujin_kim@korea.ac.kr 🗀 http://csarch.korea.ac.kr