A 4.69mW LLM Processor with Binary/Ternary Weights for Billion-Parameter Llama Model

Sangyeob Kim[1)], Jungwan Lee[2)], Byeongju Kim[2)] and Hoi-Jun Yoo[2)]

1) Yonsei University, Seoul, South Korea

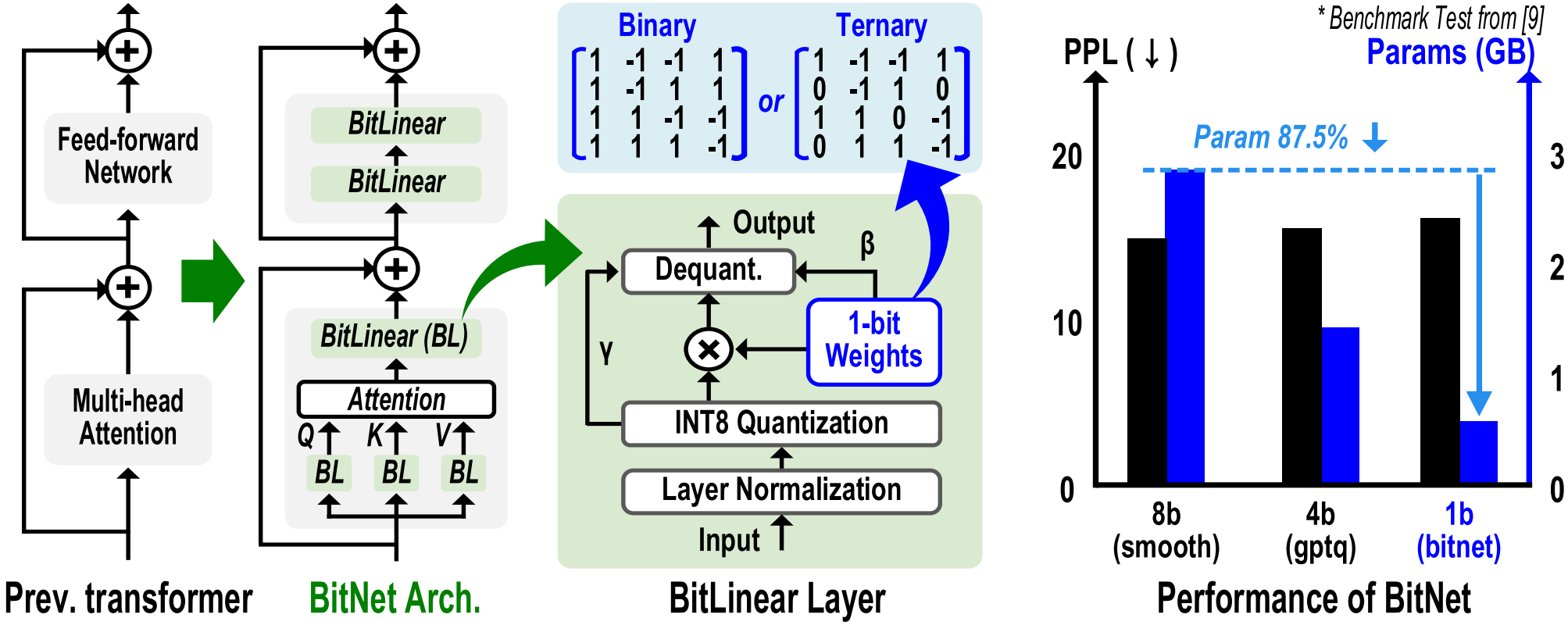
2) KAIST, Daejeon, South Korea

Lowest Energy per Parameter LLM Processor with BitNet

- Key Feature:
  - **I. Output Reuse Scheme for High Weight Sparsity**
    - ➔ **Achieving 62.3% Binary/Ternary Weight Sparsity**
  - **II. Sparsity-aware Lookup-Table for High Energy Efficiency**
    - ➔ **Increasing System Energy Efficiency by 36%**
  - **III. Index Vector Reordering for Reduced Power Consumption**
    - ➔ **Reducing System Power Consumption by 21.3%**

BitNet: State-of-the-art QAT-based LLM

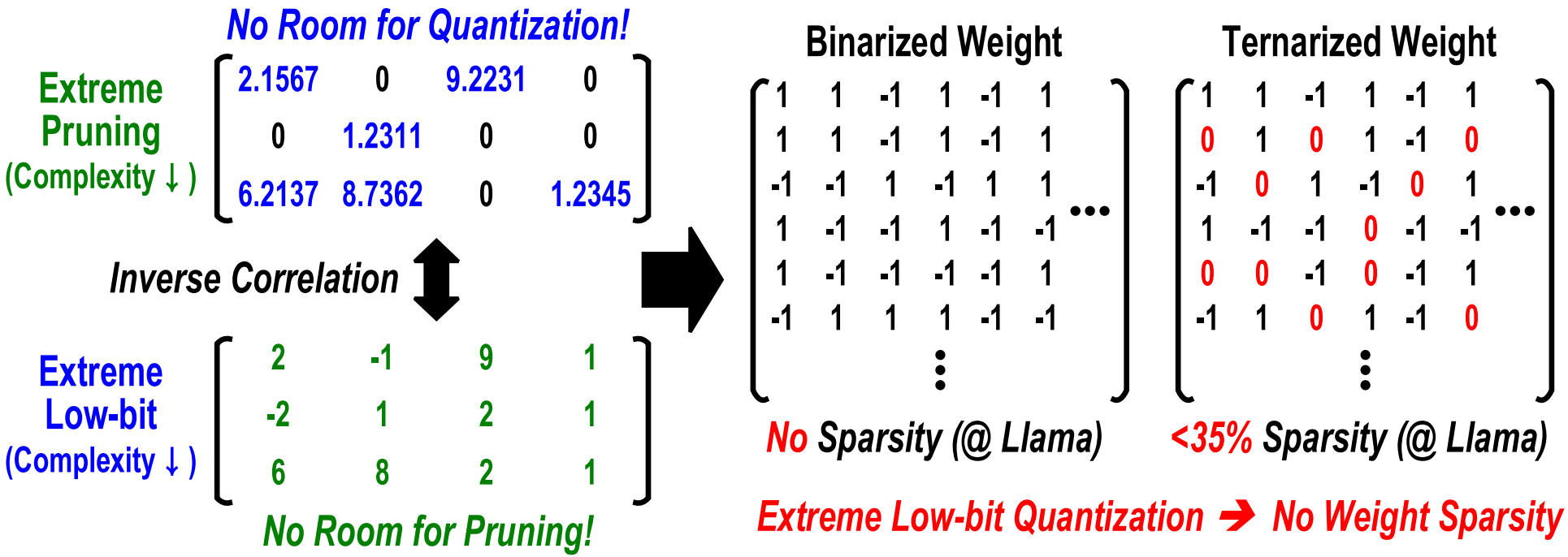
- Binary/Ternary Weight with BitLinear Layer for Attention/FFN
- Param. Size 87.5% Compared to 8b Quant. w/ Same Accuracy



Challenge of Accelerator for BitNet

- There is Inverse Correlation between Pruning & Quantization
- Binary/Ternary (Extreme Low-bit) Quantization ➔ Only 0~35% Sparsity ☹

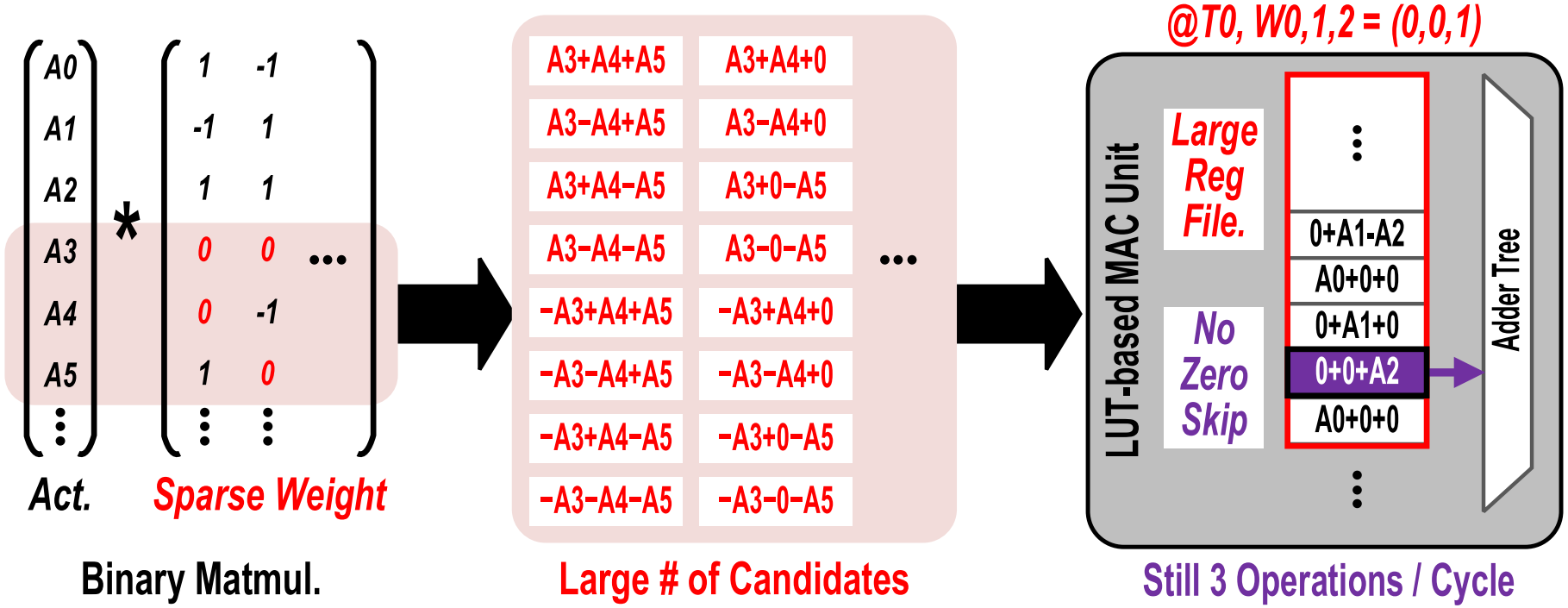
➔ **Feature 1 was Proposed to Increase Sparsity of BitNet**



Challenge of Accelerator for BitNet

- Sparsity of LUT-based MAC ↑ (Feature 1) ➔ Many Candidates ➔ LUT Size ↑
- Previous LUT-based MAC: No Zero Skipping ➔ No Speed Up @ High Sparsity

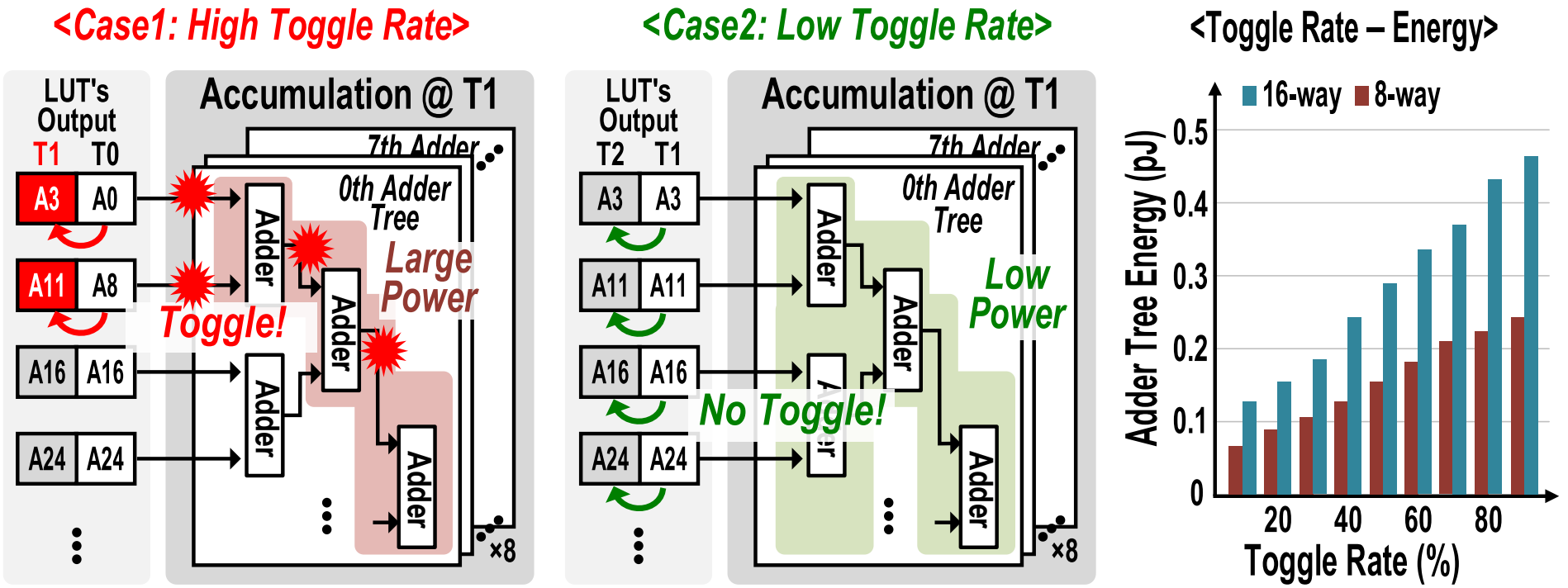
➔ **Feature 2 was Proposed to Increase Processing Speed and Area Efficiency**



Challenge of Accelerator for BitNet

- Binary Matmul. ➔ Adder Tree Power Consumption is Dominant (~50%)
- Reducing Data Toggle Rate ➔ Dynamic Power Consumption

➔ **Feature 3 was Proposed to Decrease Dynamic Power of Adder Tree**

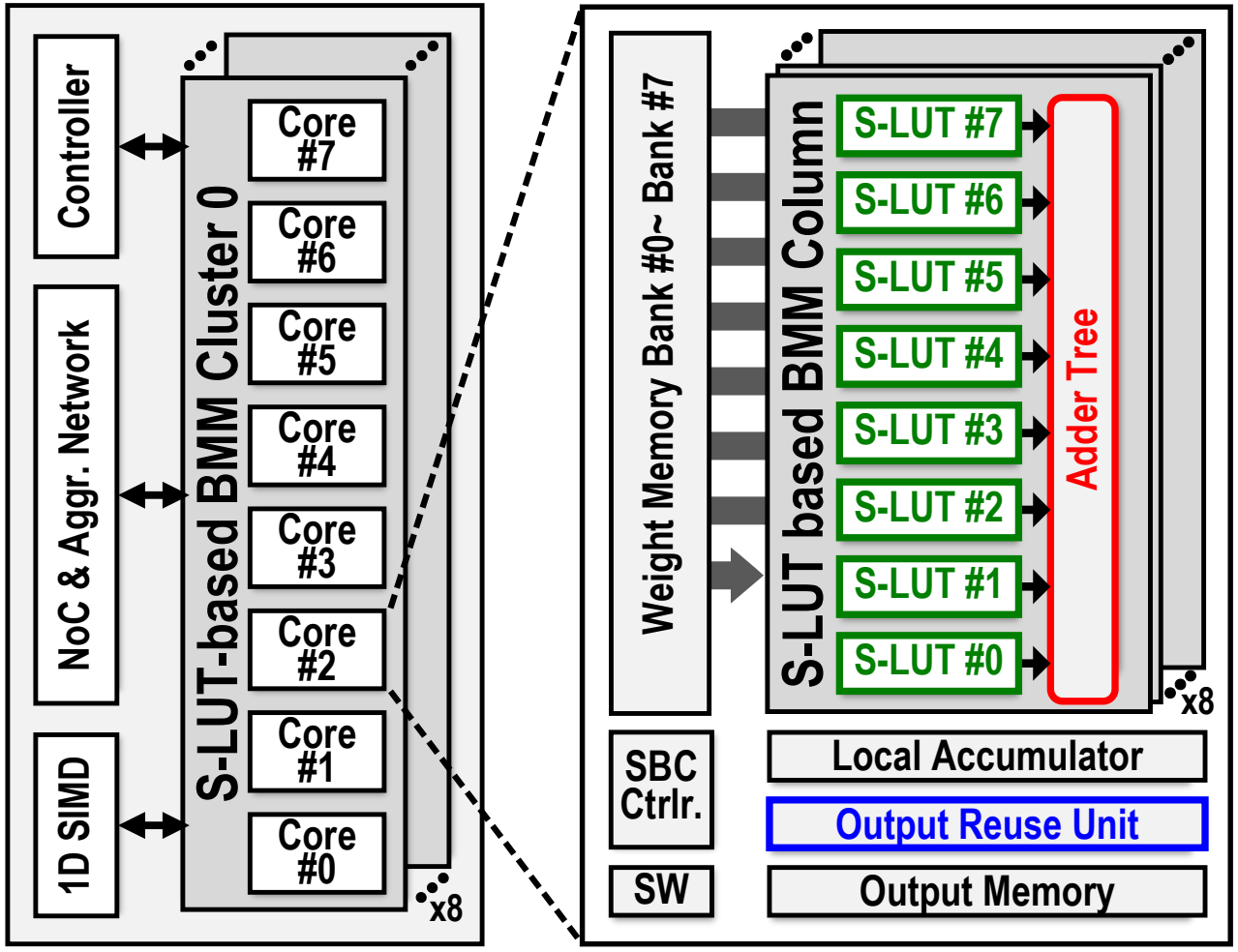


Overall Architecture of Slim-Llama

1. Output Reuse Scheme
- Increasing Sparsity

2. Sparsity-aware LUT
- Increasing Throughput

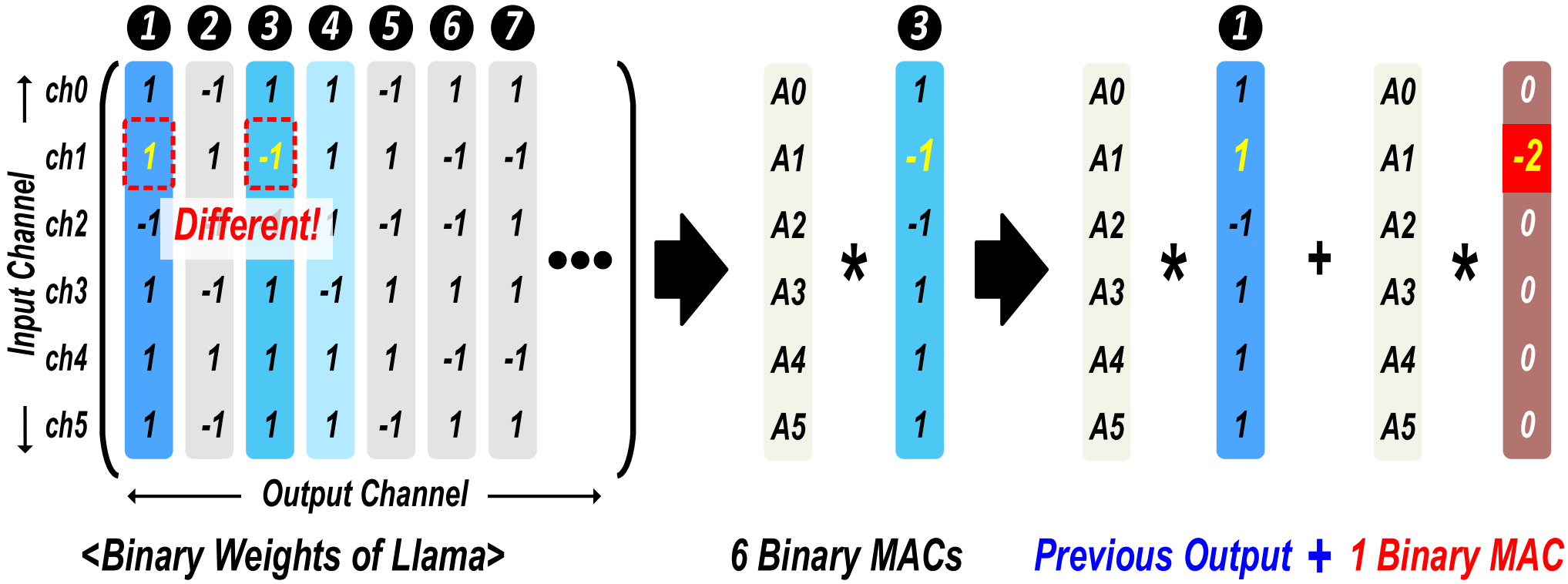
3. Index Vector Reordering
- Reducing Power Consumption



Feature1: Sparsity Generation

- Only Recalculation b/w Different Weight & Activation is Required

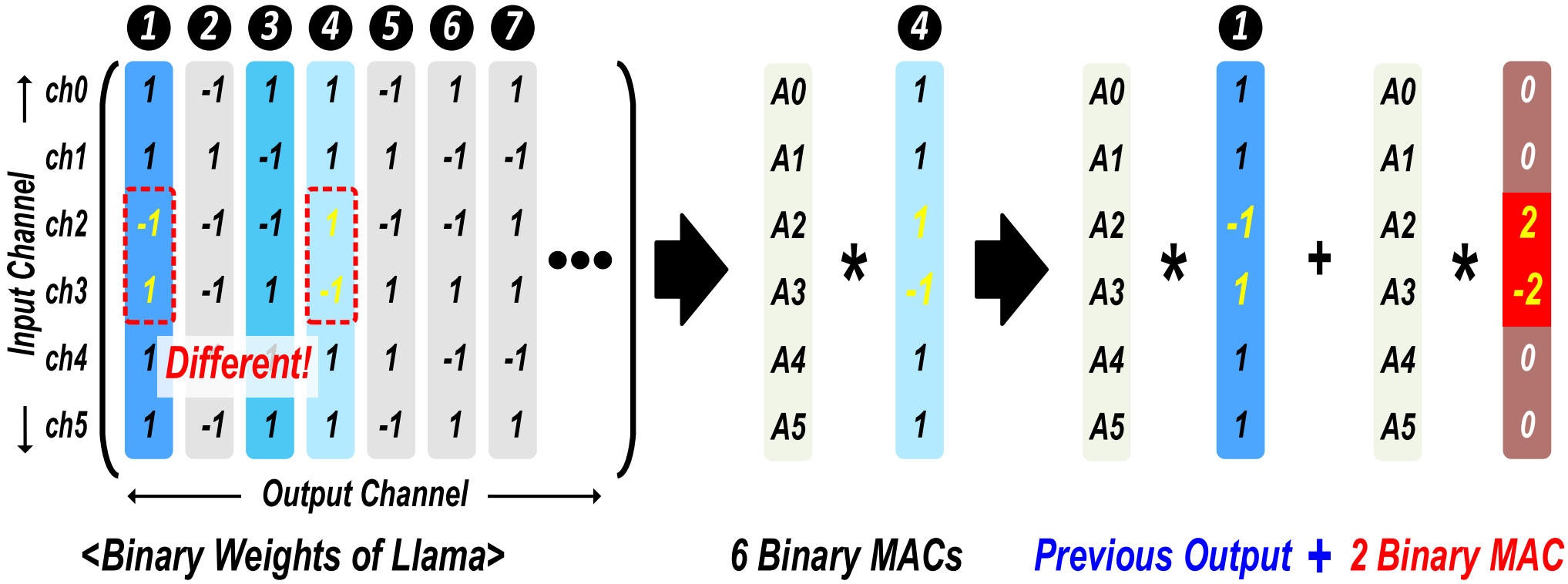
➔ By Reusing Previous Output, 1 Operation is Required for 3rd Vector



Feature1: Sparsity Generation

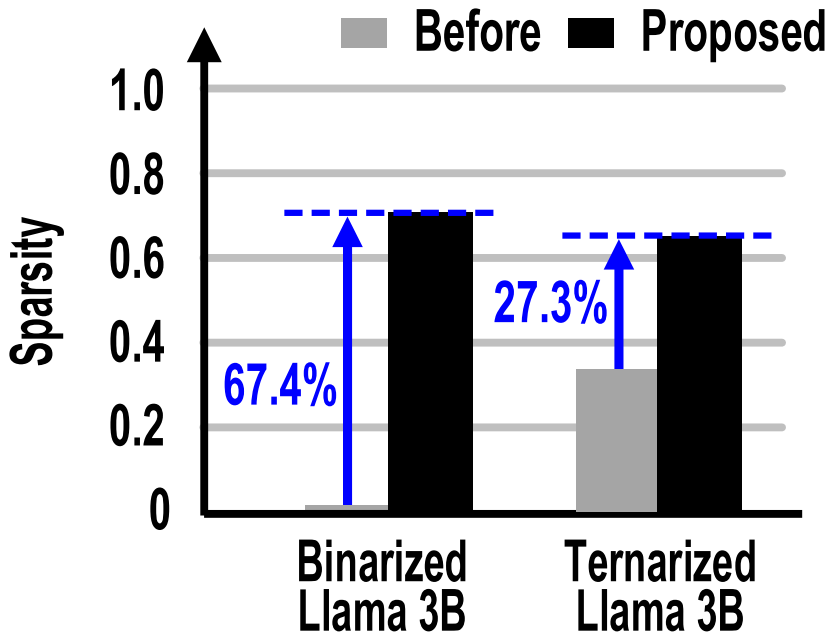
- Only Recalculation b/w Different Weights & Activations are Required

➔ By Reusing Previous Output, 2 Operations are Required for 4th Vector

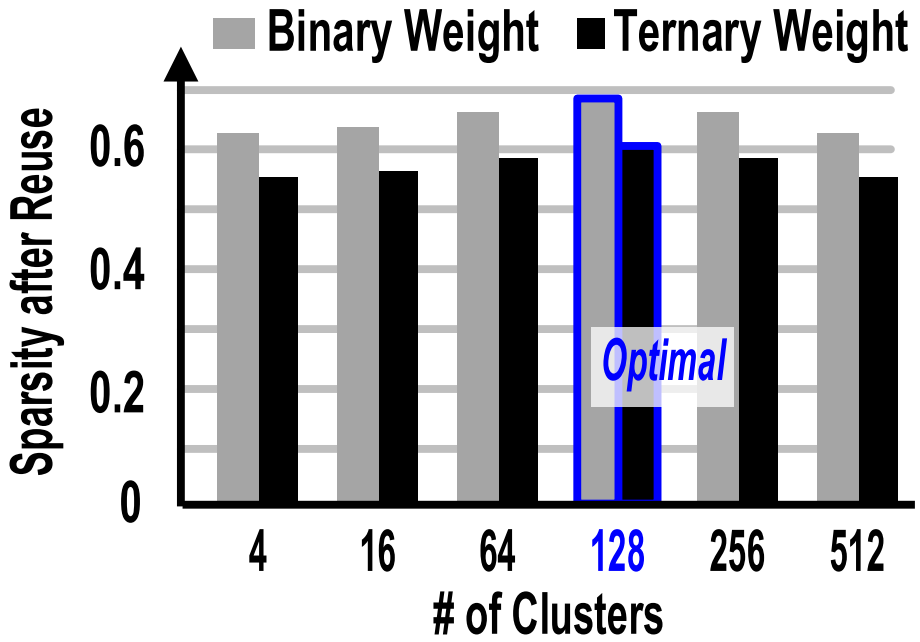


Feature1: Sparsity Generation

- 62.3~67.4% Sparsity is Achieved by Output Reuse for Llama 3B
- 128 is Optimal # of Clusters to Achieve High Sparsity for Llama



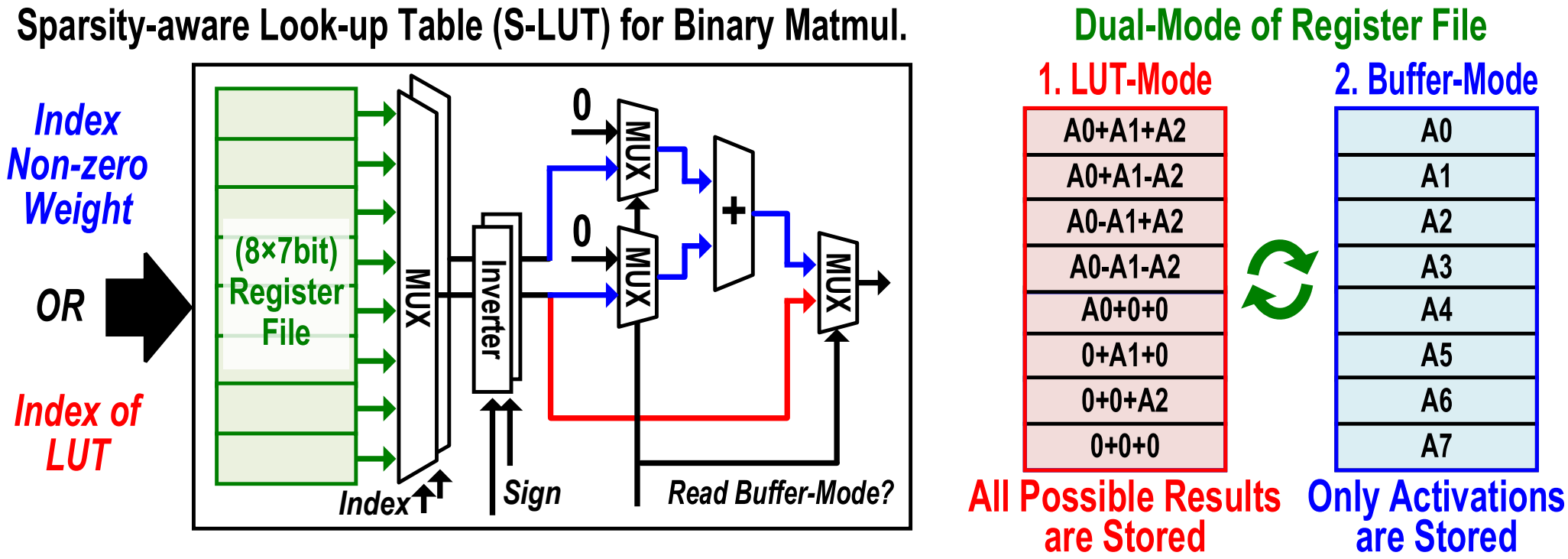
<Weight Sparsity due to Output Reuse>



<# of Clusters – Weight Sparsity>

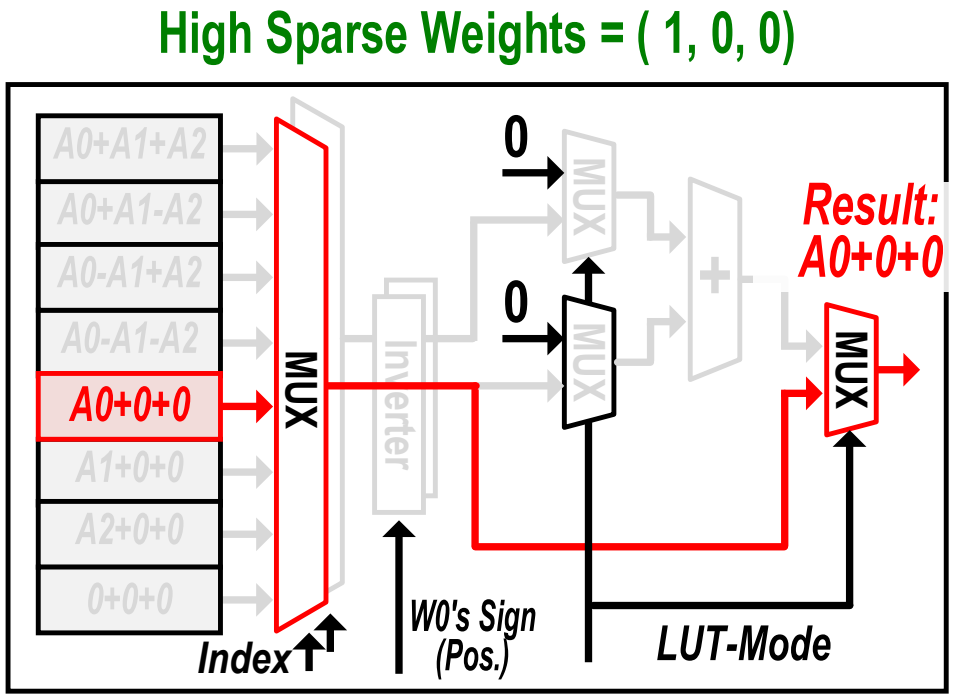
Feature2: Sparsity-aware LUT

- LUT-Mode: S-LUT Outputs the Result of 3 Binary MACs
- Buffer-Mode: S-LUT Outputs 1 Non-Sparse Binary MAC Result

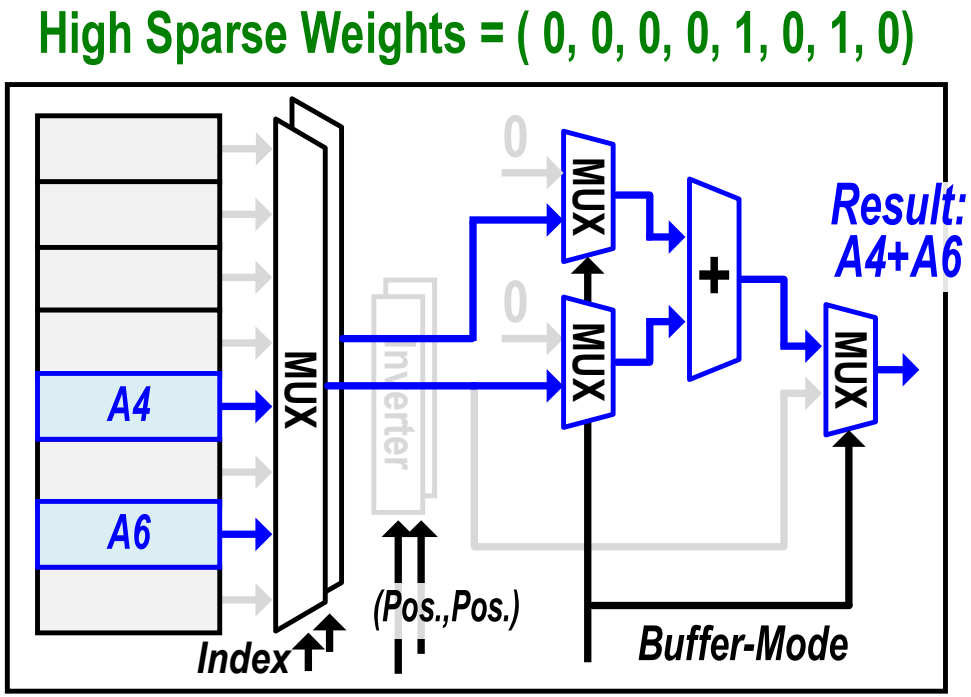


Feature2: Sparsity-aware LUT

- Buffer-Mode with Zero Skipping Performs >3 Binary MACs
- Sparsity Can Improve Throughput in Buffer-Mode



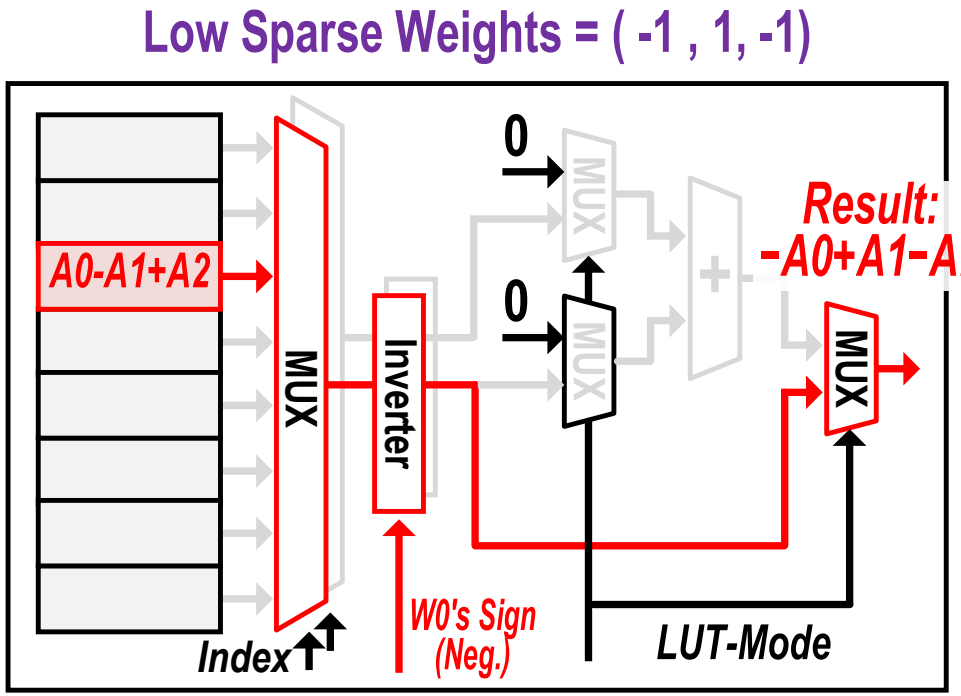
LUT → 3 Operations / Cycle



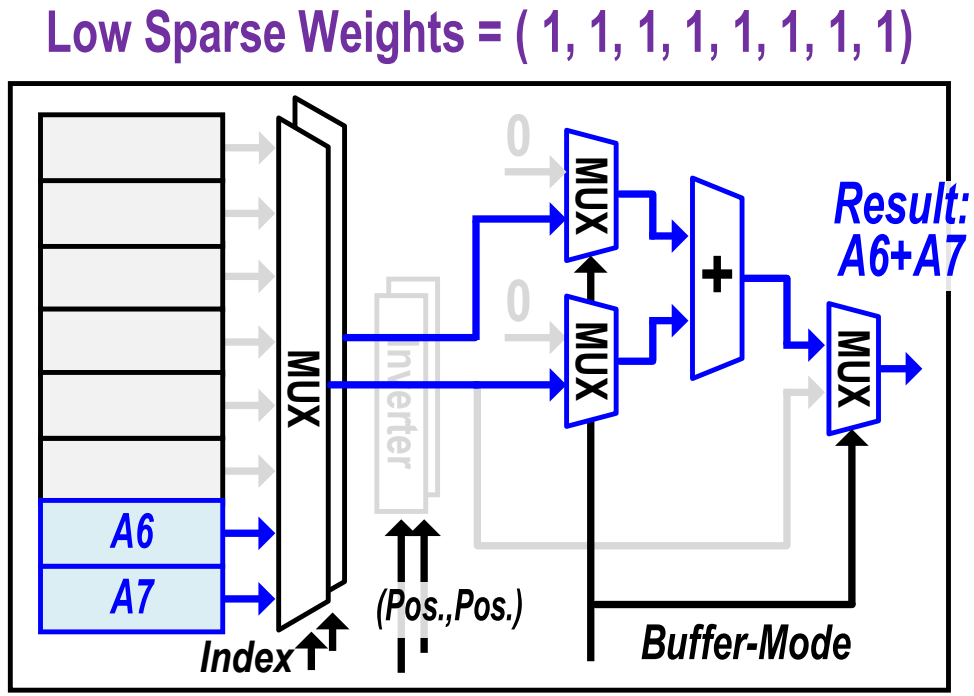
Buf. → 8 Operations / Cycle @ 75% Sparsity

Feature2: Sparsity-aware LUT

- LUT-Mode Always Performs 3 MACs Regardless of Sparsity
- Higher Throughput Can be Achieved than Buffer-Mode



LUT → 3 Operations / Cycle @ 0% Sparsity

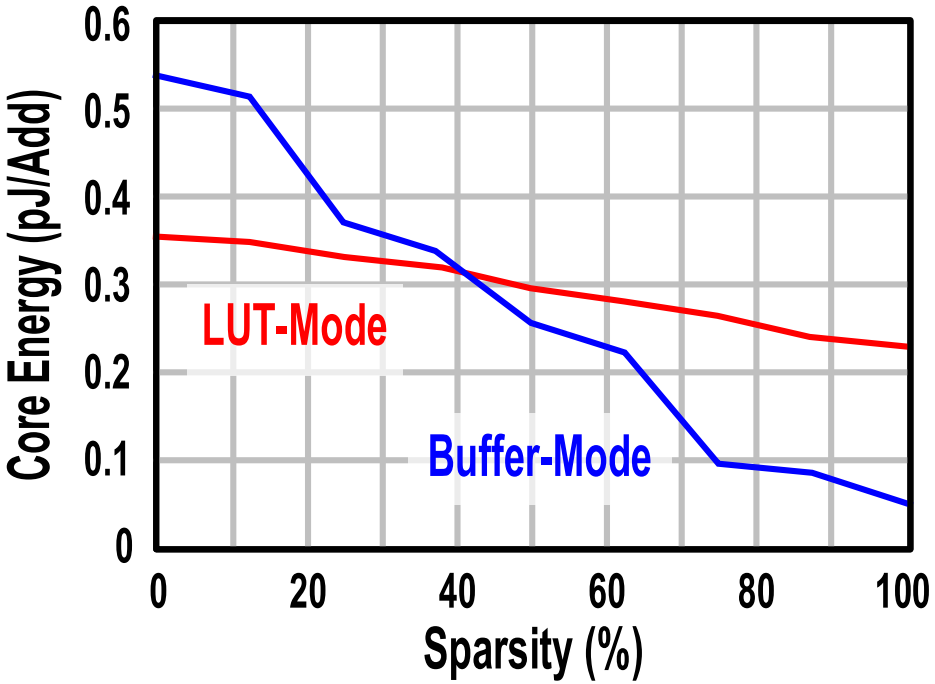


Buf. → 2 Operations / Cycle @ 0% Sparsity

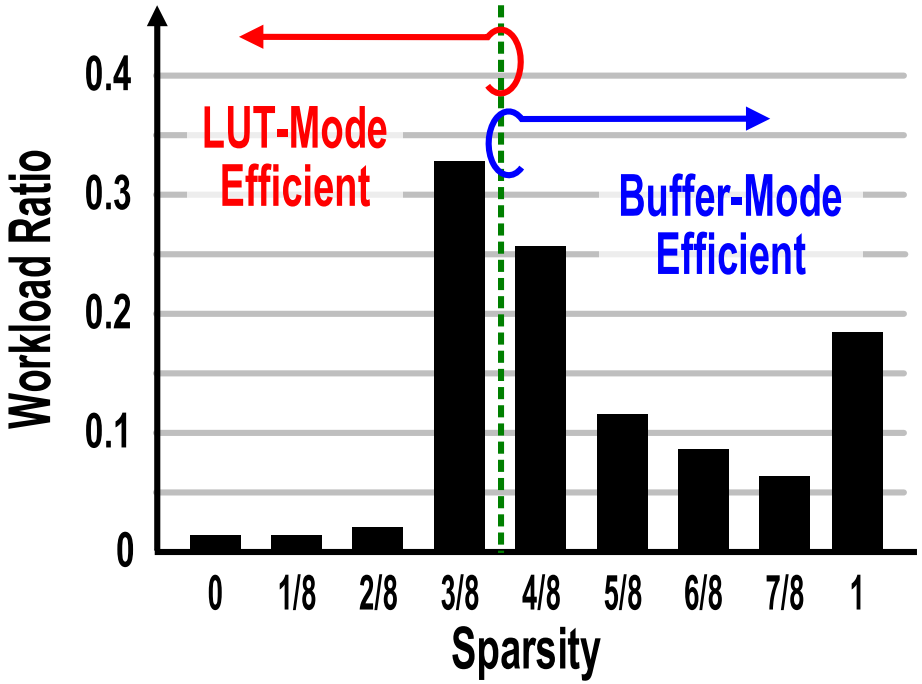
Feature2: Sparsity-aware LUT

- Sparsity > 38% ➔ Buffer Mode, Sparsity < 38% ➔ LUT-Mode
- @ Llama, 65.4%:34.6% = Buf:LUT ➔ Energy Efficiency 36%

<Sparsity-Energy Trade-off>

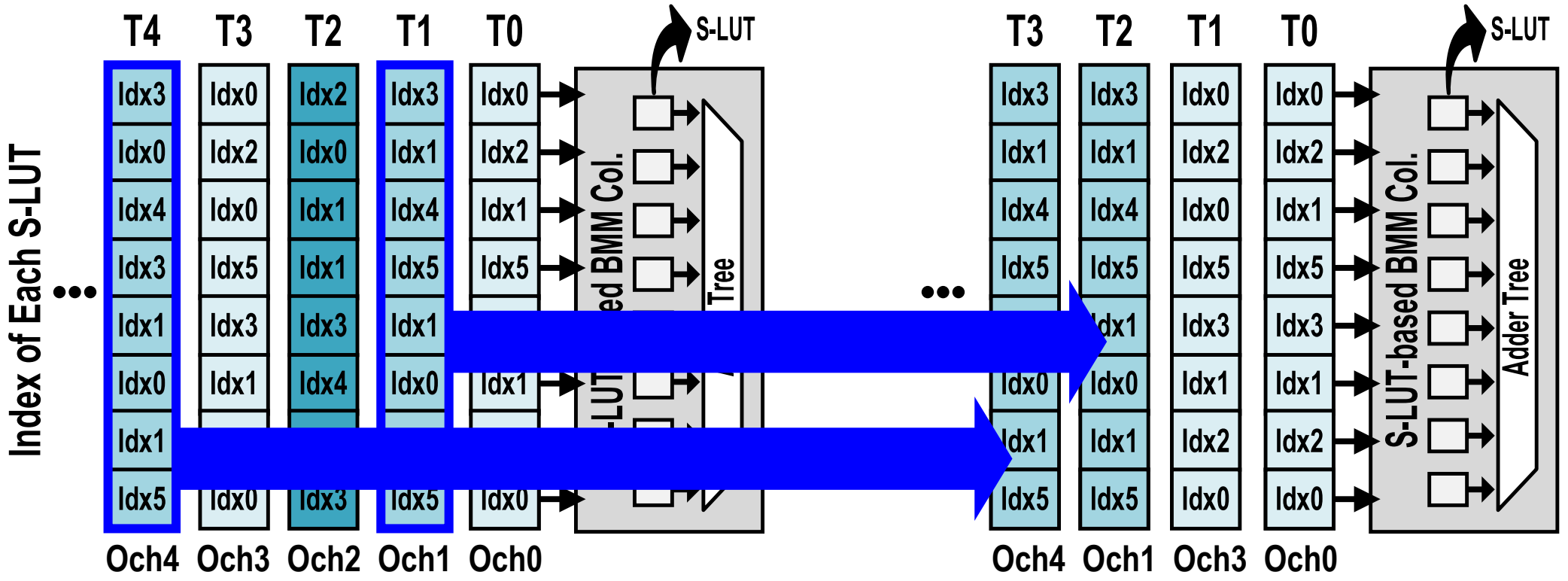


<Workload Ratio of Each Sparsity>



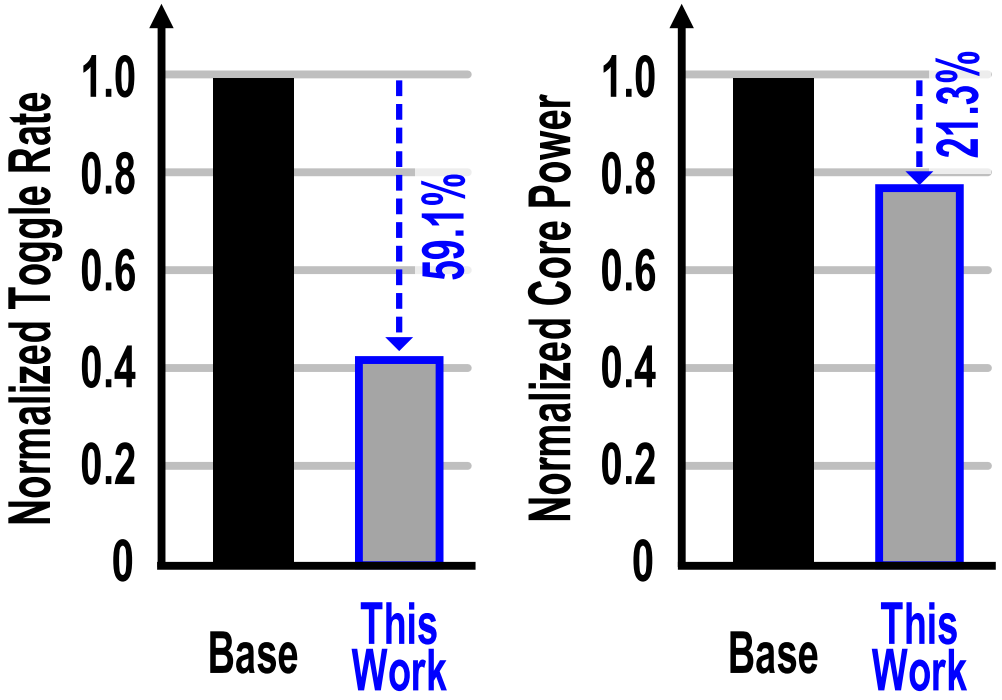
Feature3: Index Vector Reordering

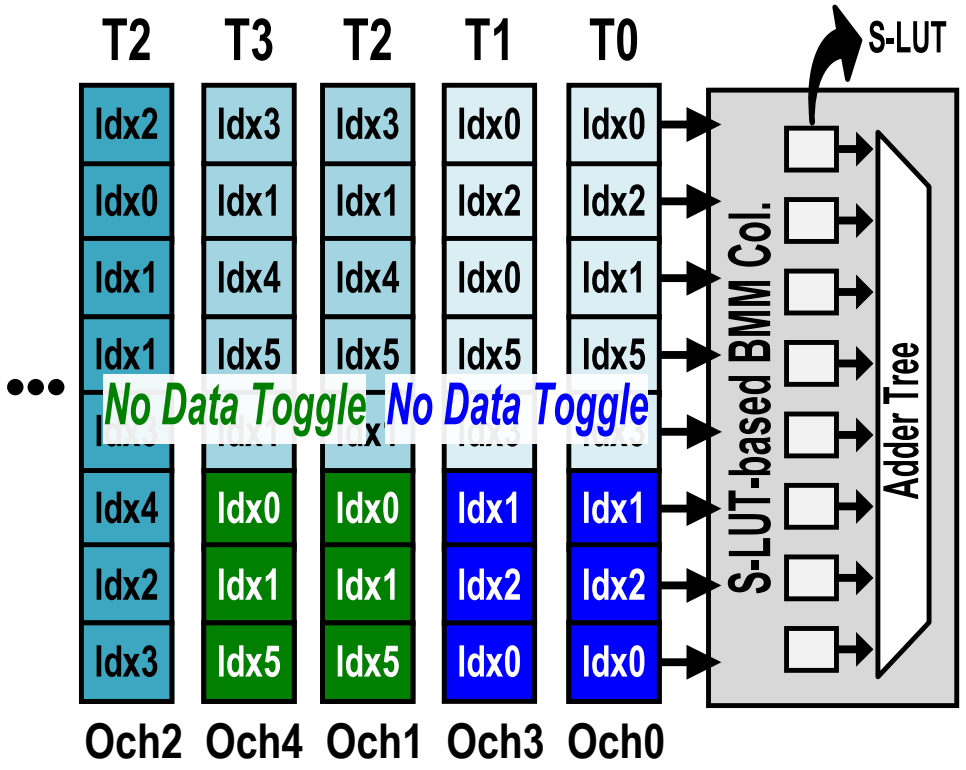
- Vectors with Similar S-LUT Idx Patterns are Reordered in Order
- Och4 Vector is Reordered to be Processed after Och1



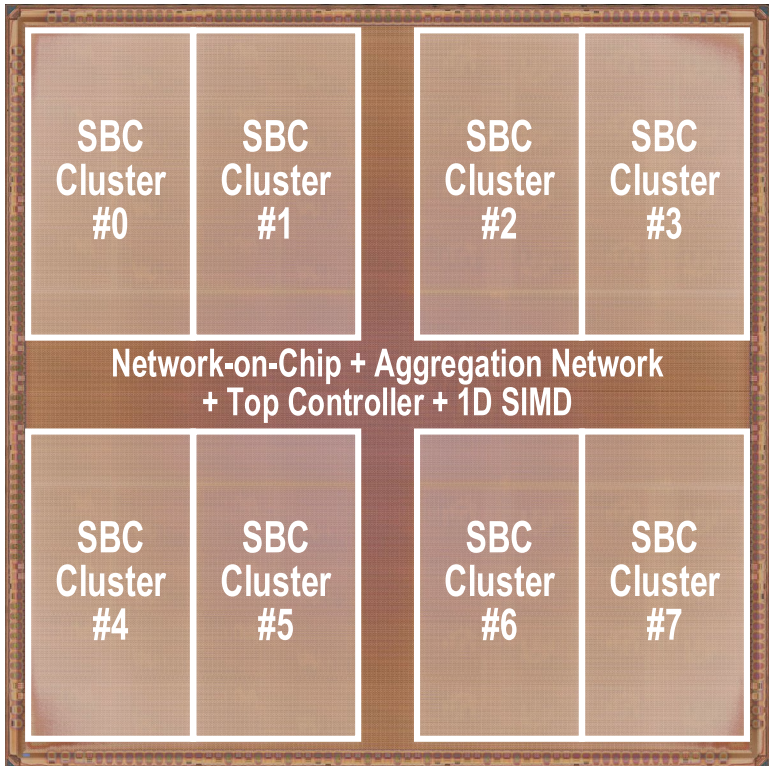
Feature3: Index Vector Reordering

- Adder's Input Toggle rate is Reduced by 59.1%
- Slim-Llama's Power Consumption is Reduced by 21.3%





Chip Photograph and Summary



1) Llama benchmark test @ 50 MHz, 0.65V
2) Bit means precision of weight & All model's activation is INT8
3) Dataset consists of Pile, Common Crawl snapshots, RealNews, CC-Stories
4) EMA is included (with DDR3 interface)
5) Normalized to 1024 Tokens
6) @ 200 MHz, 1.0 V

	Specifications		
Technology	Samsung 28nm 1P8M CMOS		
Die Area	20.25 mm^2		
Voltage	0.58 – 1.0 V		
Frequency	25 – 200 MHz		
SRAM	500 KB		
Data Precision	A: INT4/8/16, W: INT1-16 or Ternary		
System Power (mW)	4.69 (@ 25MHz, 0.58V)		
	82.07 (@ 200MHz, 1.0V)		
Peak Performance	4.92 – 13.1 TOPS (@ 200MHz, 1.0V)		
Benchmark Energy Efficiency[1] (TOPS/W)	255.9 (A: 4b, W: 1.58b), 128.0 (A: 8b, W: 1.58b), 64.1 (A: 16b, W: 1.58b)		
External Bandwidth	1.6 GB/s (@ 200MHz)		
Model Type[2]	Llama 1bit	Llama 1.58bit	Llama 8bit
Dataset	English Corpus[3]	WikiText2	WikiText2
Parameters	3 Billion	3 Billion	3 Billion
Perplexity (↓)	17.07	10.3	10.0
System Energy Consumption[1],[4],[5]	28.3 (uJ/Token)	41.3 (uJ/Token)	226.7 (uJ/Token)
Latency[4],[5],[6]	489 ms	635 ms	3915 ms

Acknowledgement

- This research was supported by the Yonsei University Research Fund of 2025-22-01 (Future-Leading Research Initiative, Contribution Rate: 50%) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2025-02218704 , Development of Chiplet-based DNN-SNN Hybrid Computing Processor for Large-Language-Models, Contribution Rate: 50%).