A 65nm In-Memory Computing Macro with Enhanced Channel-Wise Adder Tree Architecture

Advisor : 張孟凡 教授　　Group : A525　　Group Member : 林柮杉、陳科錡

Abstract

Static RAM (SRAM) offers fast access, low power, and CMOS compatibility, making it ideal for computing-in-memory (CIM) architectures that reduce data transfer bottlenecks in Von Neumann systems.

In this project, we implement a 65nm SRAM-based CIM macro using the High-bit Full-precision Multiply Cell (HFMC) and Double-Bit 6T SRAM (DBcell) to support MAC operations. Weights are accessed via dual wordlines and inputs via LBL/LBLB. To enhance accumulation efficiency, we optimize the Channel-wise Adder Tree (CAT) beneath the HFMC array, achieving improved compute performance and a better figure of merit (FoM).

Circuit Architecture

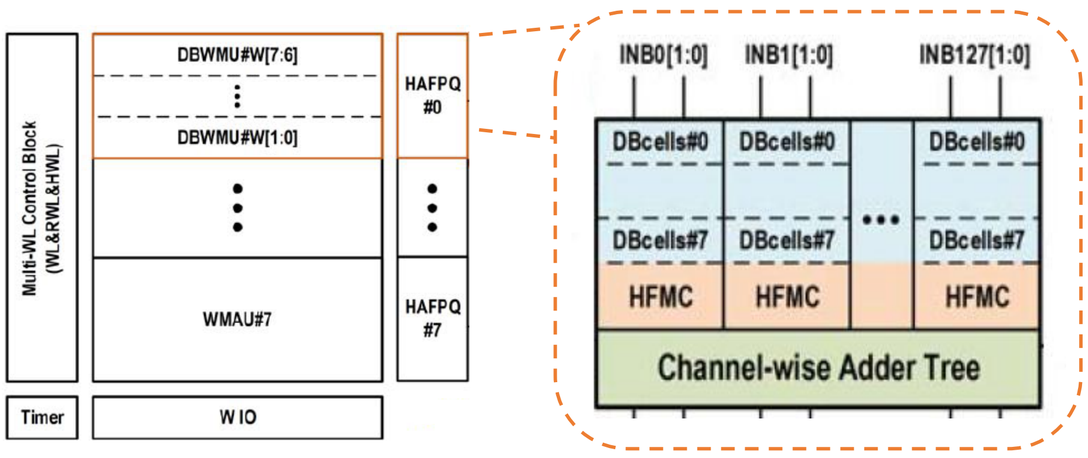


Fig. 1 CIM Macro Structure [1]

Double-Bit Cell (DBcell) :

The DBcell consists of two 6T-SRAM bitcells with a split wordline structure controlled by WL, RWL, and HWL. In **Memory Mode**, HWL is high, enabling data write through GBL–LBL; in **Computation Mode**, HWL is low to isolate GBL and LBL, avoiding interference.
By controlling RWL_{2N} and WL_{2N+1}, the two stored weights (WB_{2M} and WB_{2M+1}) can be selectively read via LBL/LBLB for 2-bit weight access in HFMC operations.

HFMC (High-bit Full-precision Multiply Cell) :

The HFMC (High-bit Full-precision Multiply Cell), located beneath the DBcell, performs two 2-bit input × 1-bit weight multiplications and outputs two partial products (PP), which are sent to the adder tree for accumulation. Implemented fully in digital logic, HFMC operates by:

- Converting inputs into LBL/LBLB voltage signals
- Reading 2-bit weights from the DBcell as control signals
- Executing logic operations (e.g., OR/NOR) to generate PPs

Each HFMC supports two parallel multiplication paths, increasing overall data throughput.

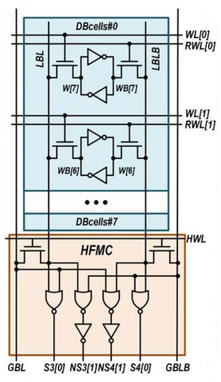


Fig. 2 Computation Logic Schematic [1]

$\overline{Weight}$	$\overline{Input}$		$Input \times Weight$
LBL / LBLB	IN[1:0]	GBLB & GBL	Out0[1:0] / Out1[1:0]
0	11	00	11
	10	01	10
	01	10	01
	00	11	00
1	X	X	00

Table. 1 Truth table of HFMC [1]

Comparison of Adder

1. **28T Full Adder** uses standard CMOS logic with strong drive strength, full voltage swing, and high stability, ideal for critical paths. Its drawbacks are large area and high power consumption.
2. **14T Full Adder**, based on GDI, reduces transistor count for lower area and power. However, it suffers from voltage drop and weak drive, making it suitable for non-critical stages.
3. **PTL Full Adder** uses transmission gates and optimized logic paths to reduce delay and capacitance. Though efficient, it may have incomplete voltage swing, requiring careful buffering.

Interleaved Hybrid Design

To balance power, delay, and signal integrity in full adder design, we propose an interleaved hybrid adder tree combining 14T and 28T full adders. While GDI-based 14T adders are compact and energy-efficient, they suffer from limited signal swing and weak drive strength. On the other hand, traditional 28T CMOS adders provide strong driving capability but consume more area and power. Our design strategically interleaves 14T and 28T adders across logic levels (Fig. 7). By inserting 28T adders at critical stages, we compensate for signal degradation in 14T paths, effectively breaking long degradation chains and enhancing timing stability and signal correctness.

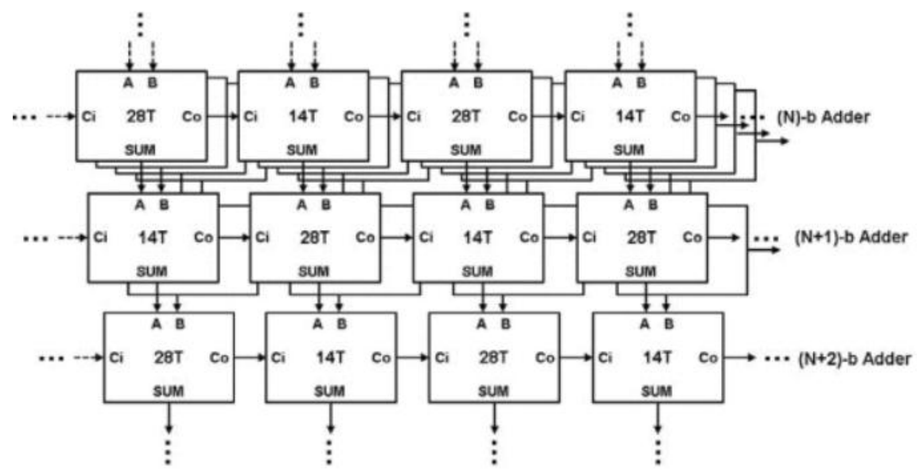


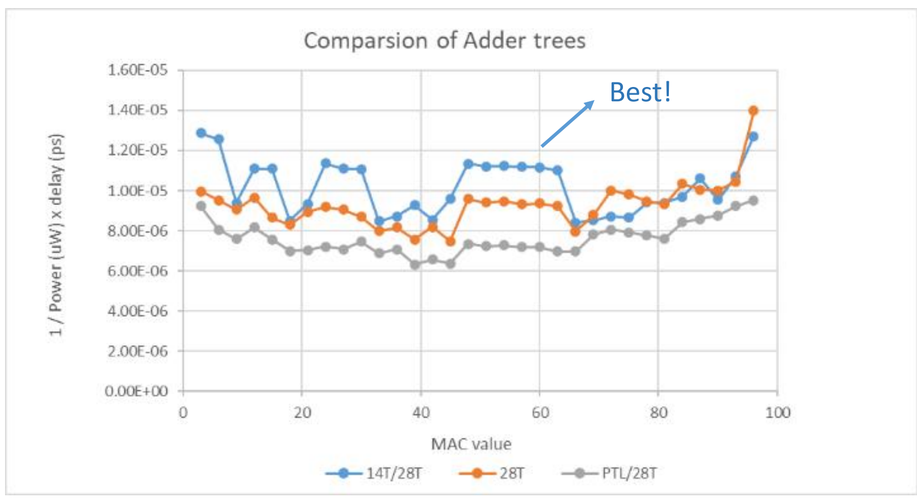
Fig. 3 Interleaved Hybrid Adder Tree [2]

Compared to pure 14T or 28T adder trees, the interleaved structure offers better signal swing and logic stability than pure 14T.Lower area and energy cost than pure 28T.Balanced trade-off among power, delay, and area.

This design is a practical solution for energy-efficient, high-throughput CIM architectures.

Result & Conclusion

To overcome the limited drive strength and swing degradation of GDI-based 14T adders. We implemented an SRAM-CIM macro featuring an interleaved hybrid adder tree, where 14T GDI and 28T CMOS full adders are alternately arranged to balance energy efficiency and signal integrity. Optimization structure with following result:



Simulation results show that :

1. 28T Adder Tree: Shows stable FOM performance across MAC values, with moderate power and delay. However, it never reaches optimal performance in any region, reflecting its limited energy efficiency despite its signal stability.
2. 14T/28T Hybrid Tree: Achieves the highest FOM in several MAC regions, especially when MAC values range from 40 to 70. The interleaving helps compensate for the 14T's lower signal swing with 28T's strong output, improving overall reliability and energy efficiency.
3. PTL/28T Hybrid Tree: Suffers from weak PTL drive strength, leading to signal degradation and increased delay and power in some cases. The high sensitivity to layout and loading makes it less suitable for large-scale use without careful design consideration.

Overall, the interleaved 14T/28T adder tree achieves an optimal trade-off among power, delay, and area, making it well-suited for energy-efficient, multi-channel integer neural network accelerators.

Reference :

[1] A. Guo et al., "A 28nm 64-kb 31.6-TFLOPS/W Digital-Domain Floating-Point-Computing-Unit and Double-Bit 6T-SRAM Computing-in-Memory Macro for Floating-Point CNNs," 2023 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2023

[2] Y. -D. Chih et al., "16.4 An 89TOPS/W and 16.3TOPS/mm2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," 2021 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 2021

[3] Yin, N., Pan, W., Yu, Y., Tang, C., & Yu, Z. (2023). Low-Power Pass-Transistor Logic-Based Full Adder and 8-Bit Multiplier. Electronics, 12(15), 3209.