

Low-Power Time-Domain Hardware Accelerator for Edge Computing

Jie Lou Advisor: Tobias Gemmeke

Email: lou@ids.rwth-aachen.de Homepage: <https://loujie1994.github.io/>

Chair of Integrated Digital Systems and Circuit Design (IDS), RWTH Aachen University, Aachen, Germany

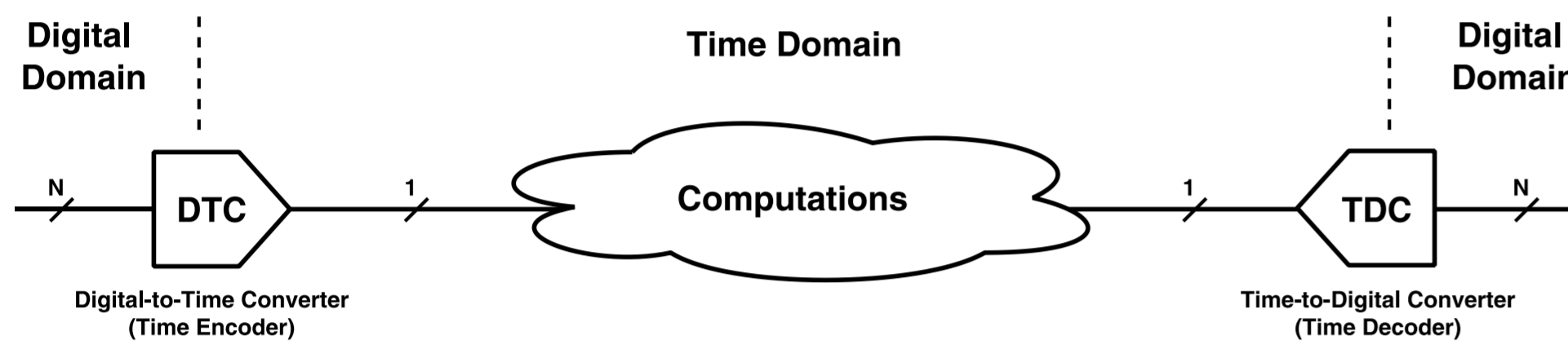


Efficient computing is crucial for energy-limited edge devices. With the growing use of artificial neural networks (ANNs), reducing energy consumption is a key challenge in edge computing. Time-domain (TD) computing has attracted attention for its inherent analog signal processing properties and compatibility with digital circuits. So, it remained unclear which scenarios are best suited for computation in the time domain. This thesis focuses on developing hardware accelerators for time-domain computing, analyzing suitable application domains, and identifying the principles and constraints that should be considered during ASIC implementation.

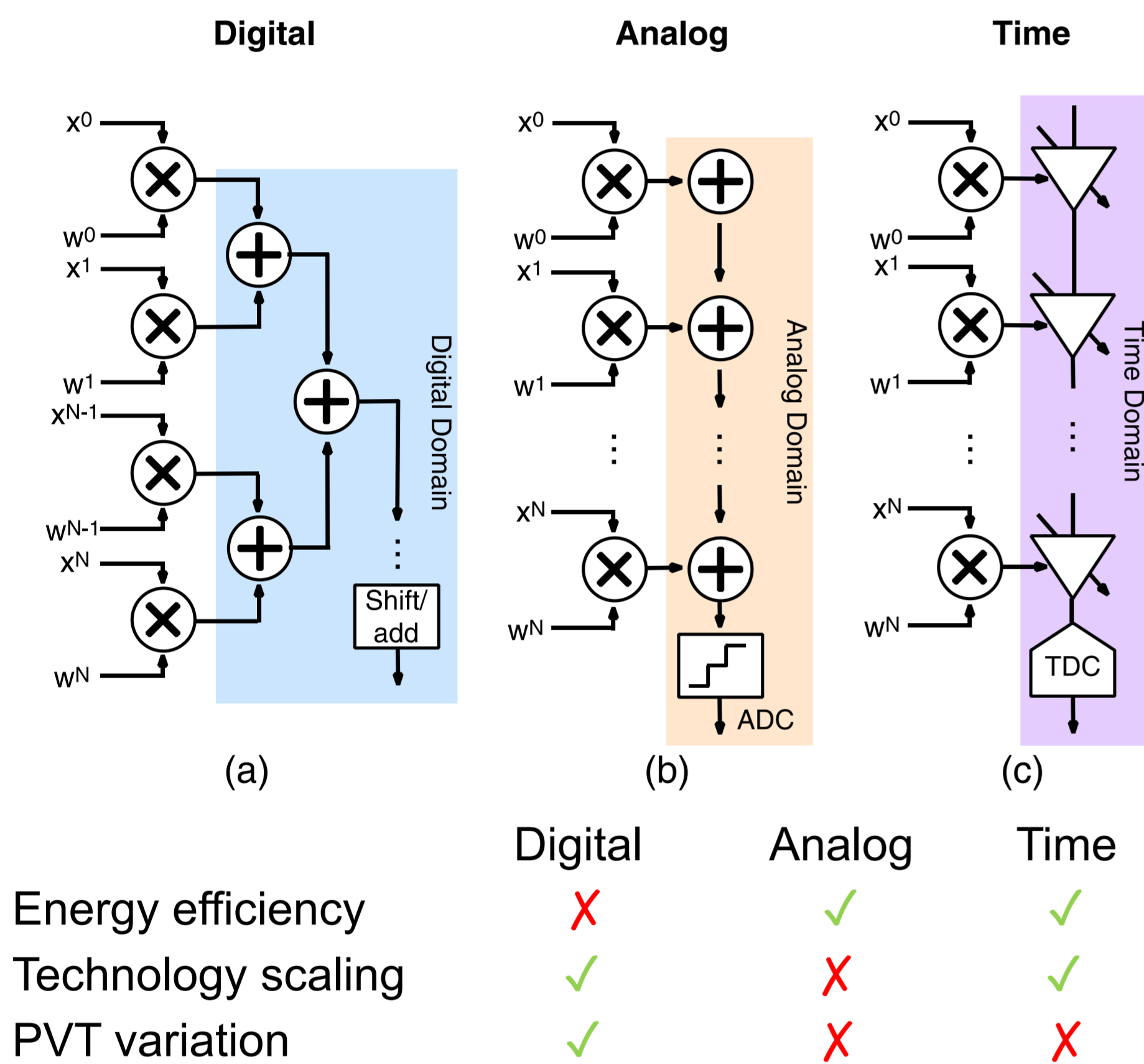
Introduction

Time-domain (TD) computing

- Encode numeric values as relative arrival time of signal edges
- Utilize the circuit's inherent delay characteristics for energy-efficient operation



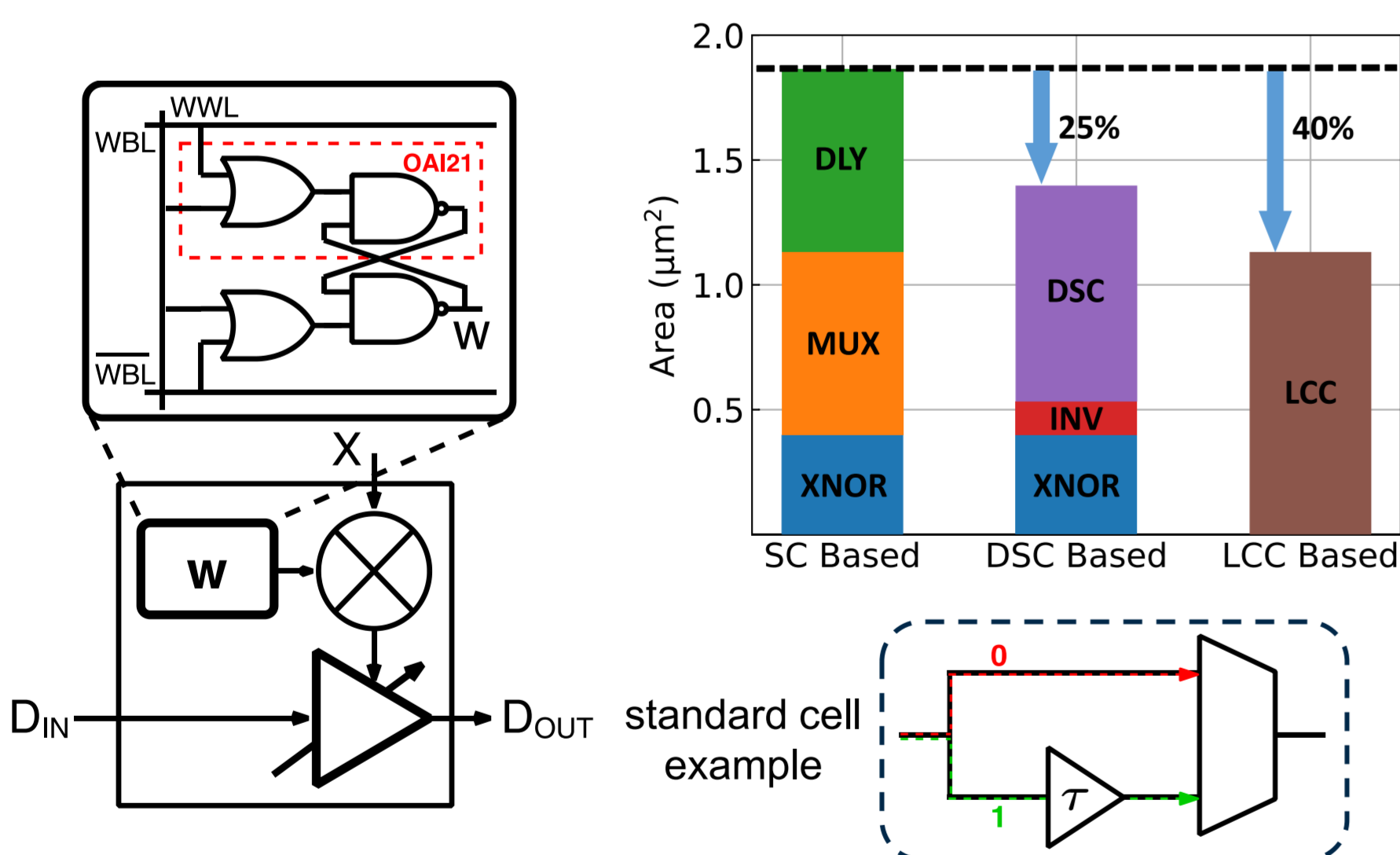
Compute domain comparison



Time Domain Basic Cells

Multiple cell types for BNN & CNN

- SR-latch based memory
- Standard and custom based cells with adjustable delay
- Operates on both rising and falling transitions
- Trade-off among area, energy and accuracy



Assisted by standard cell memory compiler

- Input: text based description of a standard cell array
- Output: Verilog netlist, placement file, routing script
- Traditional synthesis and place & route (P&R) for peripheral logic

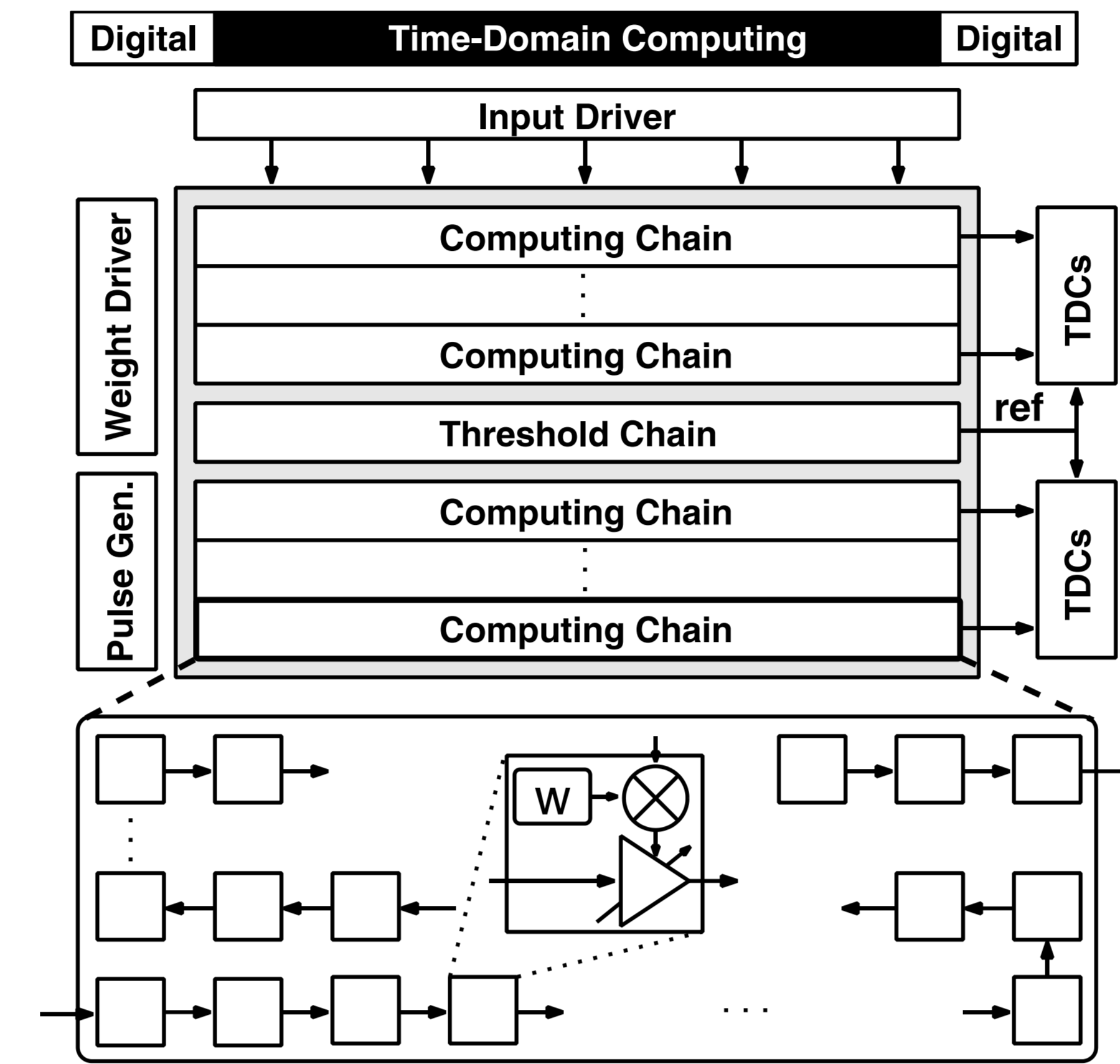
```

"name": "BasicCell",
"cell": ["OAI", "OAI", "AOI", "DLY", "MUX"],
"conn": [
[[0, "Z"], [1, "A"]],
[[1, "Z"], [0, "A"]],
[[1, "Z"], [2, "B"]],
[[2, "Z"], [4, "S"]],
[[3, "Z"], [4, "D1"]]
],
"ports": [
{"name": "WBL", "properties": ..., "conn": [[0, "B1"]]},
{"name": "WBLN", "properties": ..., "conn": [[1, "B1"]]},
{"name": "WBL", "properties": ..., "conn": [[0, "B2"]]},
{"name": "XN", "properties": ..., "conn": [[2, "A"]]},
{"name": "D_IN", "properties": ..., "conn": [[3, "A"], [4, "D1"]]},
{"name": "D_OUT", "properties": ..., "conn": [[4, "Z"]]}
],
"flip": [1],
"wiring": [...]
    
```

Time Domain Architecture

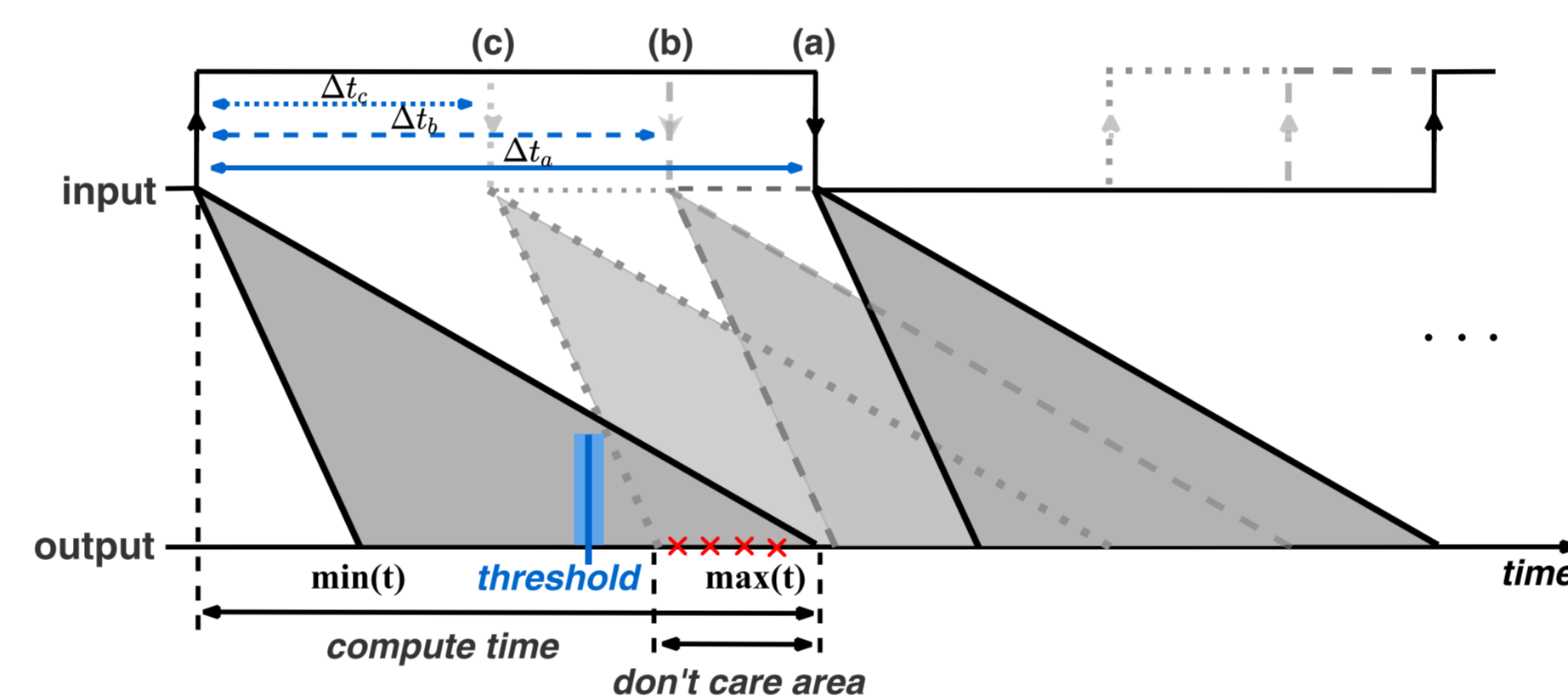
Three tapeouts on 22nm FDSOI technology

- Binary Neural Networks (BNNs)
- Convolutional Neural Networks (CNNs)
- Multiple computing chains and single threshold chain
- Snake-like arrangement to optimize chain aspect ratio



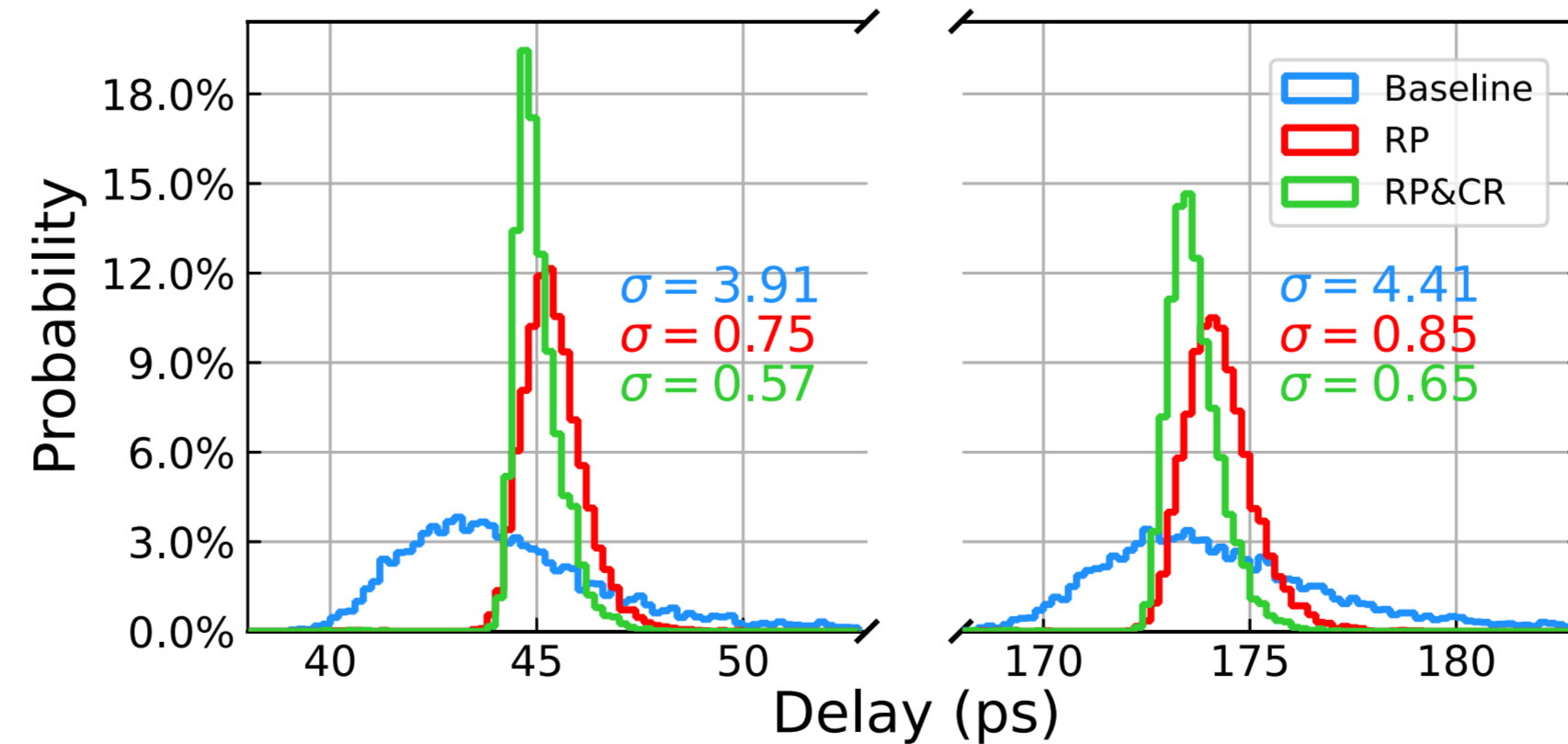
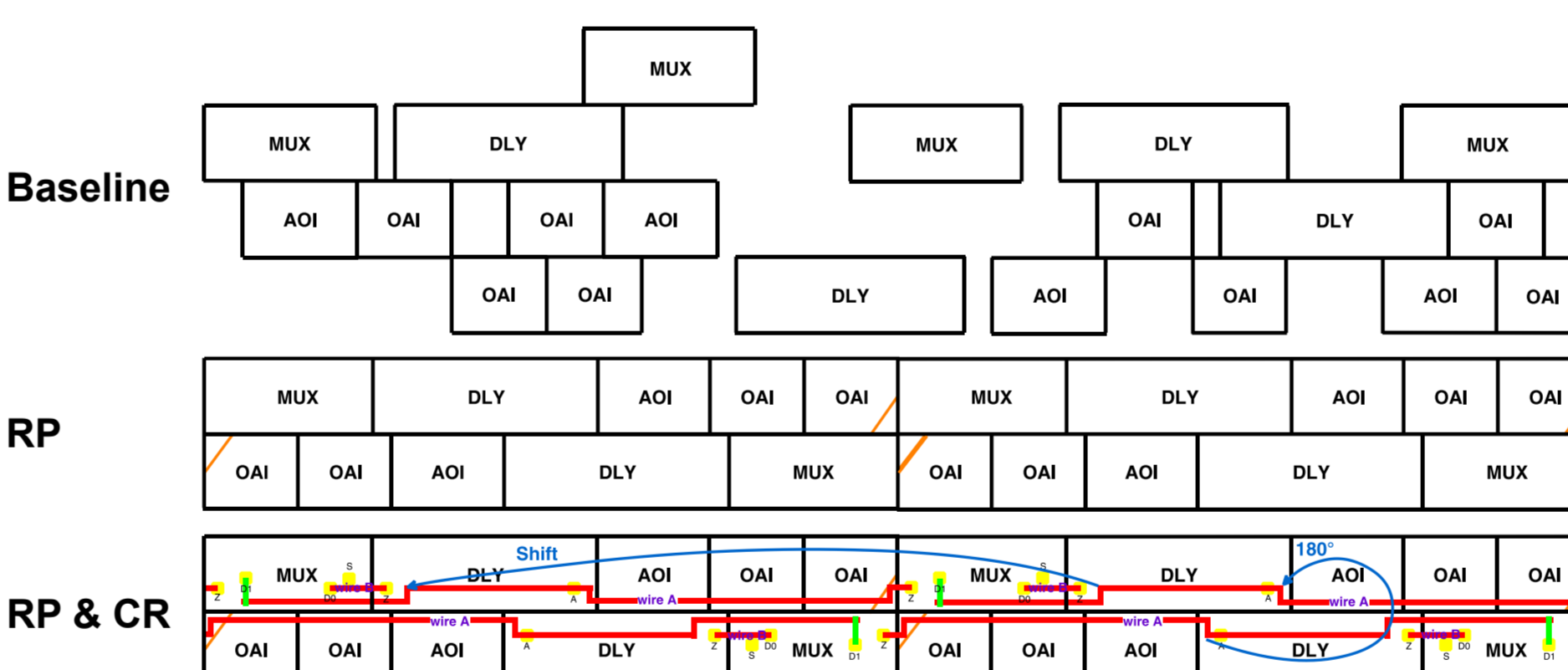
Wave-pipelining technique

- Throughput is limited by the total delay of the computing chain → improved by wave-pipelining technique
- Reduce leakage power and increase energy efficiency



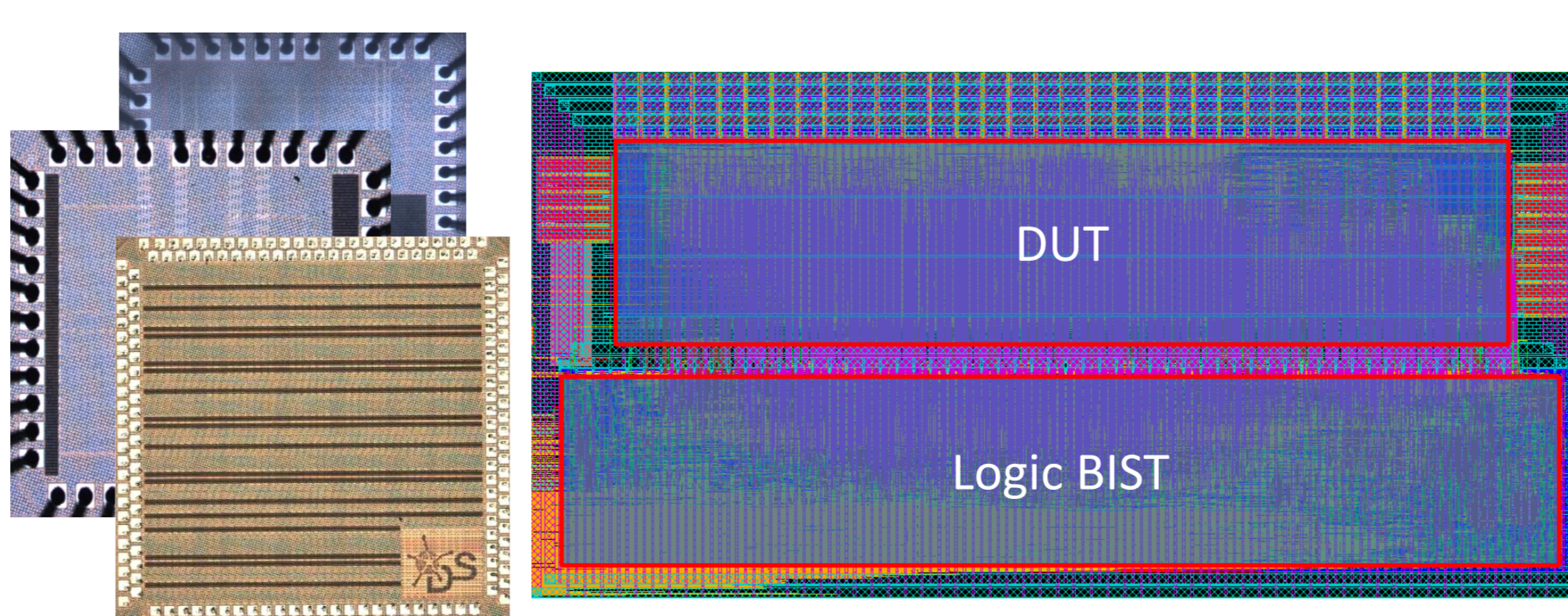
Regular placement (RP) & custom routing (CR)

- Timing uncertainty sets a lower bound on delay per unit
- RP: specify data path information for cell placement
- CR: creation of wire segments with specified coordinates, layers and directions
- Supported by standard cell memory compiler



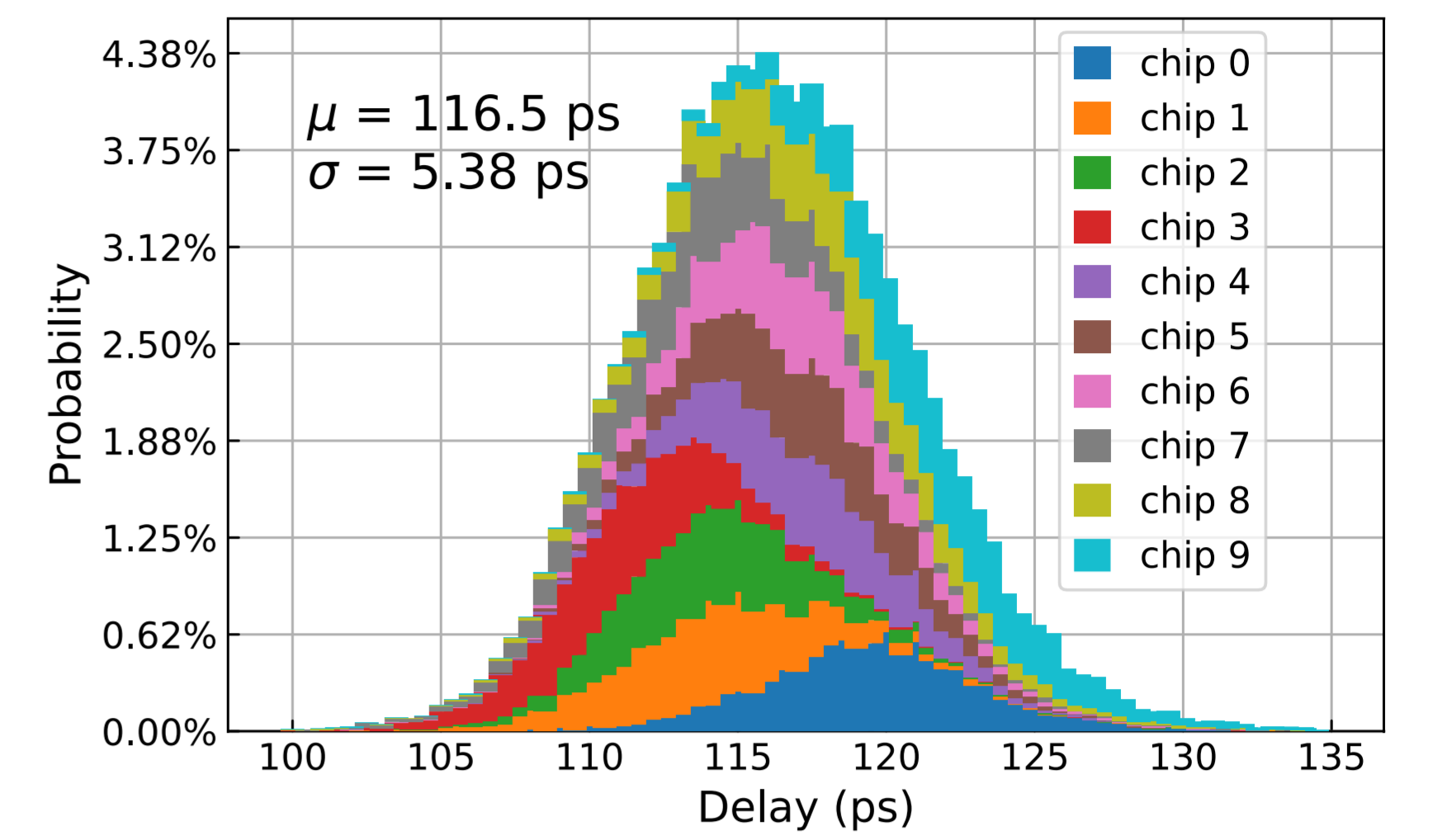
*Extracted from standard delay format (SDF) file after layout

Die Micrographs and Layout View

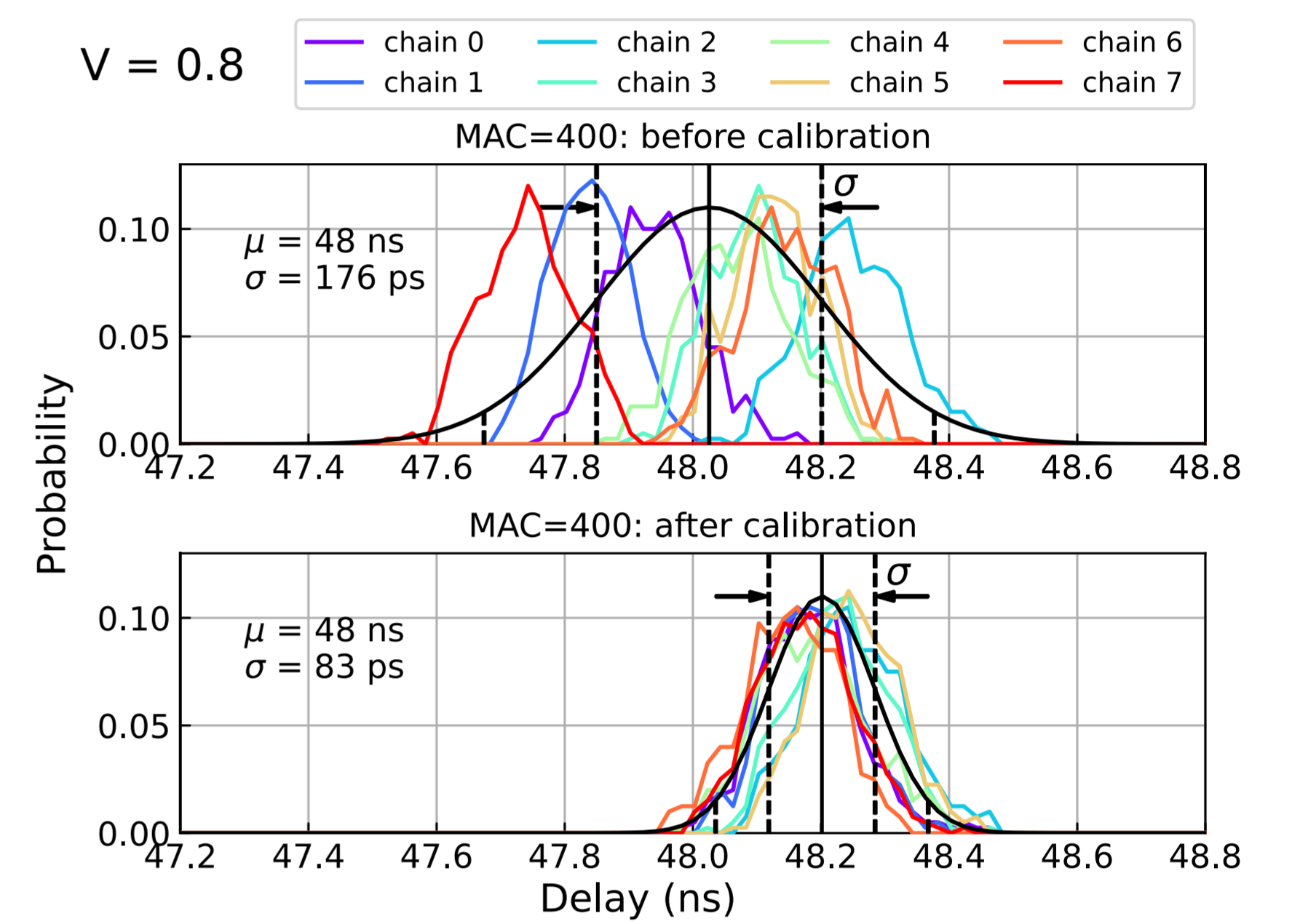


Measurement Results

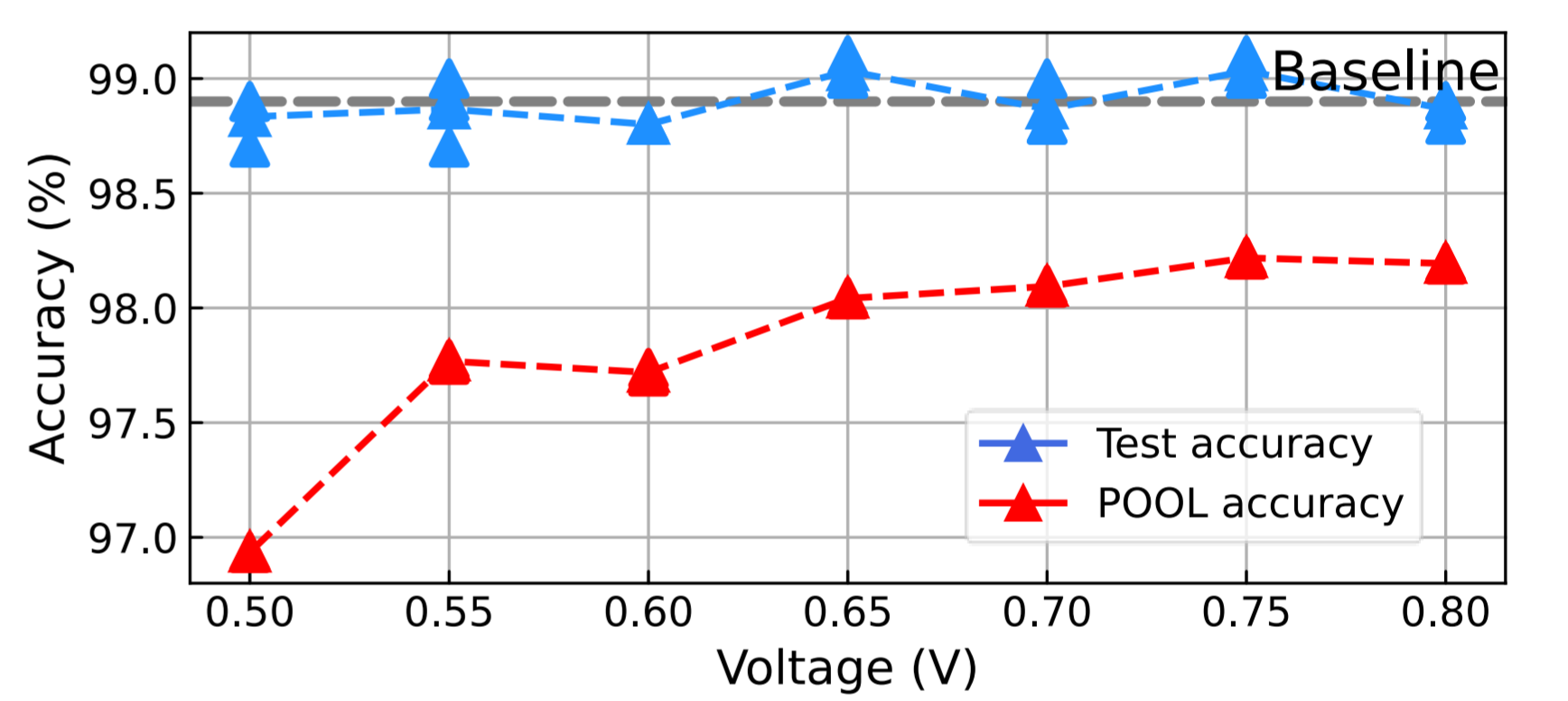
Measured cell delay across 10 dies



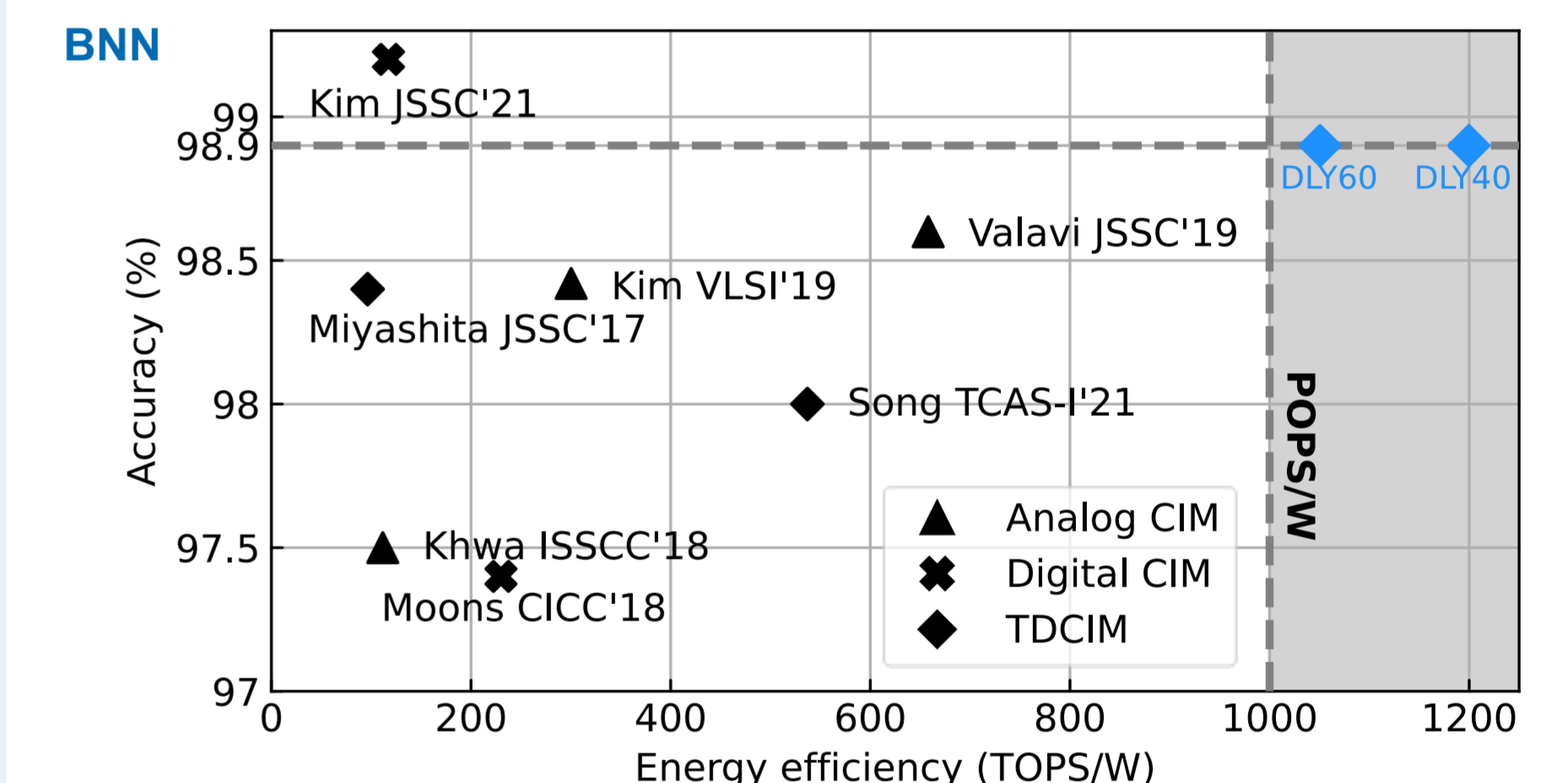
Measured computing delay



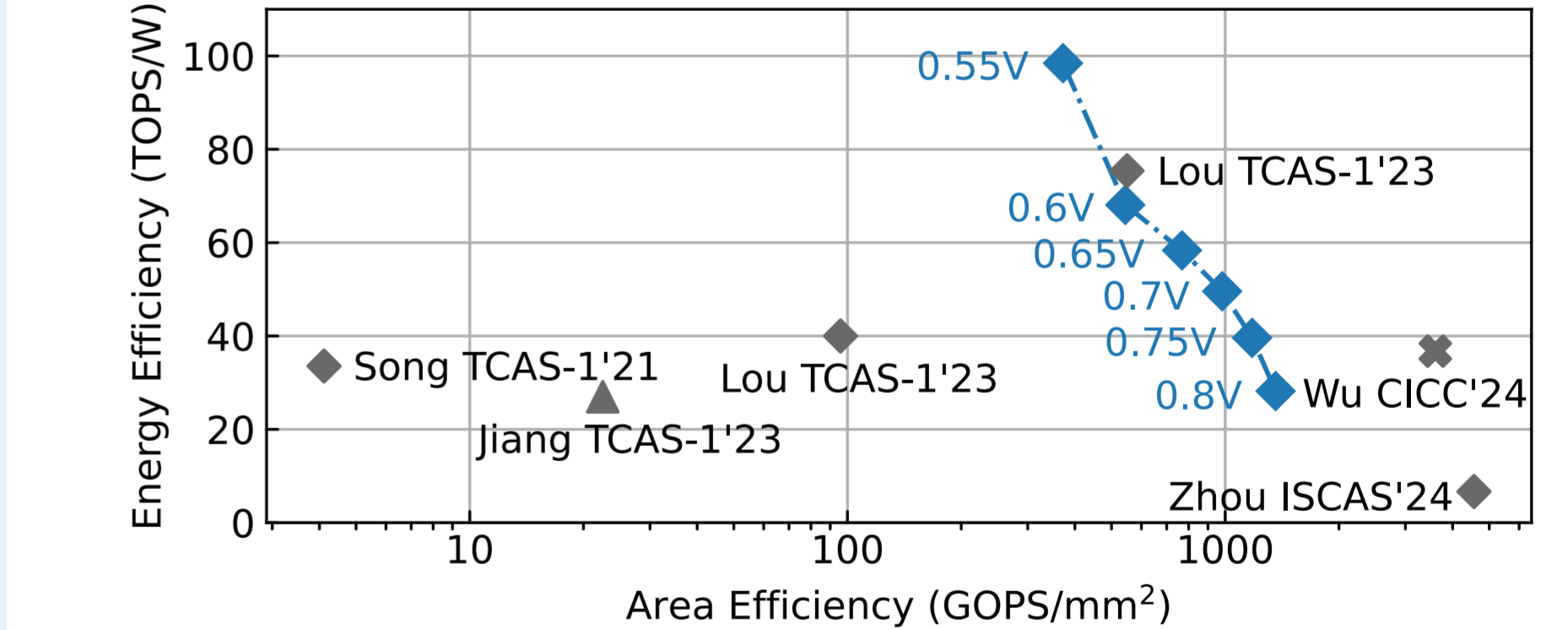
Accuracy



Comparison with SotA



CNN (4b)



Selected Publications

- [1] J. Lou, C. Lanus, F. Freye, T. Stadtmann, and T. Gemmeke, "All-Digital Time-Domain Compute-in-Memory Engine for Binary Neural Networks With 1.05 POPS/W Energy Efficiency", IEEE European Solid State Circuits Conference (ESSCIRC), 2022.
- [2] J. Lou, F. Freye, C. Lanus, and T. Gemmeke, "Scalable Time-Domain Compute-in-Memory BNN Engine with 2.06 POPS/W Energy Efficiency for Edge-AI Devices", Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI), 2023.
- [3] J. Lou, F. Freye, C. Lanus, and T. Gemmeke, "An Energy Efficient All-Digital Time-Domain Compute-in-Memory Macro Optimized for Binary Neural Networks", IEEE Transactions on Circuits and Systems I: Regular Papers (TCAS-I), vol. 71, no. 1, pp. 287-298, 2024.
- [4] J. Lou, F. Freye, C. Lanus, and T. Gemmeke, "An All-Digital Time-Domain Compute-in-Memory Engine for Convolutional Neural Networks in 22nm", IEEE International Symposium on Circuits and Systems (ISCAS), 2025.
- [5] F. Freye, J. Lou, C. Bengel, S. Menzel, S. Wiefels, and T. Gemmeke, "Memristive devices for time domain compute-in-memory", IEEE Journal on Exploratory Solid-State Computational Devices and Circuits (JXCDC), vol. 8, no. 2, pp. 119-127, 2022.
- [6] C. Lanus, J. Lou, J. Loh, and T. Gemmeke, "Automatic generation of structured macros using standard cells - application to CIM", IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), 2023.
- [7] F. Freye, J. Lou, C. Lanus, and T. Gemmeke, "Merits of time-domain computing for VMM - a quantitative comparison", International Symposium on Quality Electronic Design (ISQED), 2024.