ARITH 2024

Novel Access Patterns Based on Overlapping Loading and Processing Times to Reduce Latency and Increase Throughput in Memory-based FFTs

31st IEEE International Symposium on Computer Arithmetic (ARITH 2024)
Málaga, Spain

10 June 2024

Zeynep Kaya and Mario Garrido

Bilecik Seyh Edebali University, Bilecik, Türkiye
Universidad Politécnica de Madrid, Madrid, Spain

Outline

1. Introduction
   - Fast Fourier Transform
   - Perfect Shuffle and Perfect Unshuffle Algorithms
   - Input and Output Orders
   - Latency and Throughput in Memory-Based FFTs
   - Conventional Memory-Based FFTs
2. Proposed Access Patterns
   - Case 1: Natural Input - Bit-Reversed Output
   - Case 2: Scrambled Input - Bit-Reversed Output
   - Case 3: Bit-Reversed Input - Natural Output
   - Case 4: Bit-Reversed Input - Scrambled Output
3. Discussion and Comparison
4. Conclusion

The discrete Fourier transform (DFT),

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, k = 0, 1, ..., N-1,$$

where $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$ are rotations in the complex plane called twiddle factors.

Complexity: N^2

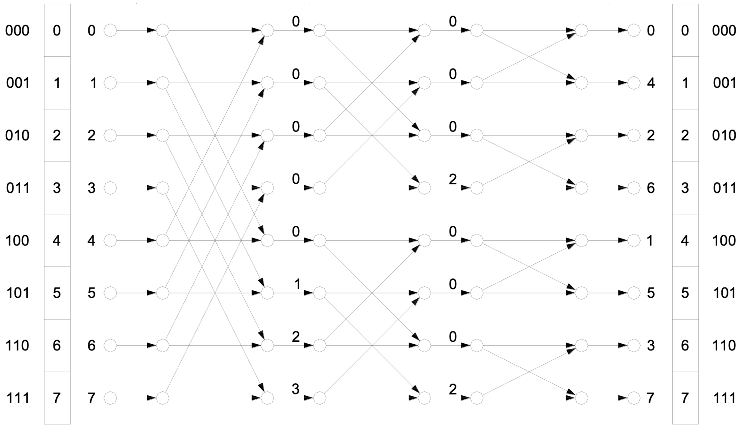
The radix-2 fast Fourier transform (FFT),

$$X[k] = \sum_{n=0}^{N/2-1} x[2n] W_{N/2}^{2nk} + \sum_{n=0}^{N/2-1} x[2n+1] W_{N/2}^{(2n+1)k}.$$

Complexity: $N \cdot \log_2 N$

ARITH 2024

N=8-point radix-2 decimation in frequency (DIF) FFT



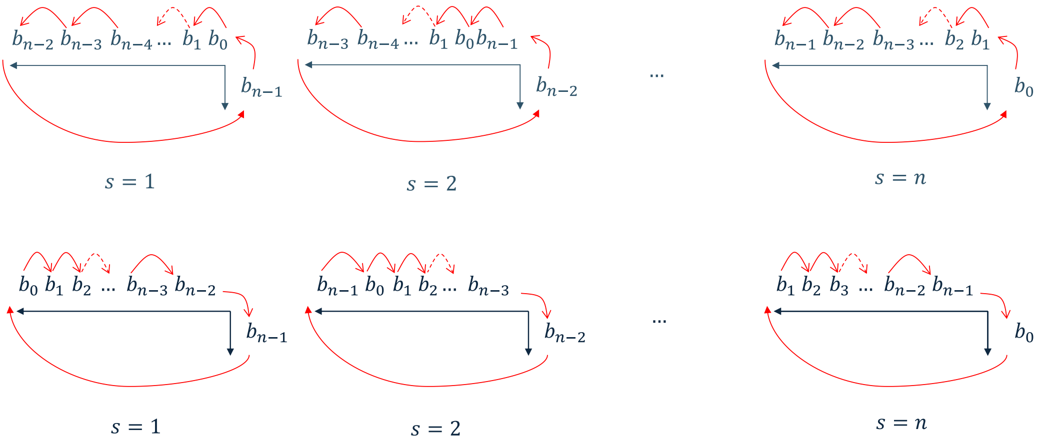
$N = 2^n$

$n =$ number of stages

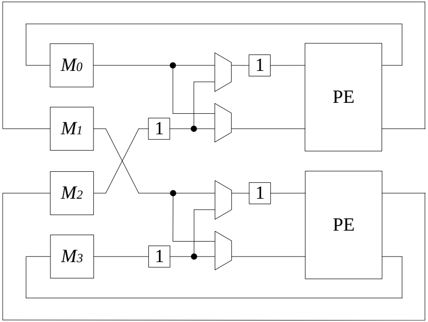
$I = b_{n-1}, \ldots, b_0$

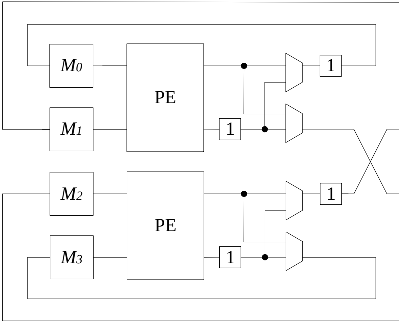
b_{n-s}

Perfect shuffle and perfect unshuffle



Architectures





$\sigma = \sigma_{sp} \circ \sigma_{pp} \circ \sigma_{ss}$

$I = b_{n-1}b_{n-2}\ldots b_p \mid b_{p-1}\ldots b_0$

σ_{ss}: $b_{n-2}\ldots b_p b_{n-1} \mid b_{p-1}\ldots b_0$

σ_{pp}: $b_{n-2}\ldots b_p\, b_{n-1} \mid b_{p-2}\ldots b_0 b_{p-1}$

σ_{sp}: $b_{n-2}\ldots b_p\, b_{p-1} \mid b_{p-2}\ldots b_0 b_{n-1}$

$\sigma = \sigma_{ss} \circ \sigma_{pp} \circ \sigma_{sp}$

$I = b_{n-1}b_{n-2}\ldots b_p \mid b_{p-1}\ldots b_0$

σ_{sp}: $b_{n-1}b_{n-2}\ldots b_0 \mid b_{p-1}\ldots b_1 b_p$

σ_{pp}: $b_{n-1}b_{n-2}\ldots b_0 \mid b_p b_{p-1}\ldots b_1$

σ_{ss}: $b_0 b_{n-1}\ldots b_{p+1} \mid b_p b_{p-1}\ldots b_1$

Input and output orders

Natural Input: $\mathcal{P} = b_{n-1} \dots b_p | b_{p-1} \dots b_0$

Bit-reversed Input: $\mathcal{P} = b_0 \dots b_{p-1} | b_p \dots b_{n-1}$

Natural Output: $\mathcal{P} = b_0 \dots b_{p-1} | b_p \dots b_{n-1}$

Bit-reversed Output: $\mathcal{P} = b_{n-1} \dots b_p | b_{p-1} \dots b_0$

Latency and throughput in memory-based FFTs

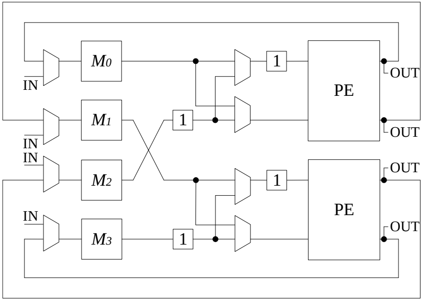
The Latency: the total clock cycles that are needed for the whole FFT calculation.

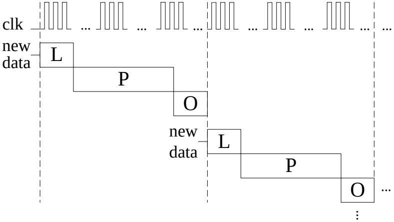
$$T_{LAT} = T_{LOAD} + T_{PROC} + T_{OUT}$$

The Throughput: the average number of samples processed per clock cycle.

$$\text{Th} = \frac{N}{\Delta T_{FFT}}$$

Conventional memory-based FFTs





$$T_{LOAD} = \frac{N}{P} \qquad T_{PROC} = \frac{N}{P} \cdot \log_r N \qquad T_{OUT} = \frac{N}{P}$$

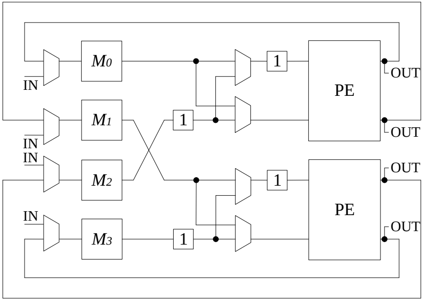
$$\boxed{\Delta T_{FFT} = T_{LAT} = \frac{N}{P} \cdot (2 + \log_r N)} \qquad \boxed{Th = \frac{P}{2+\log_r N}}$$

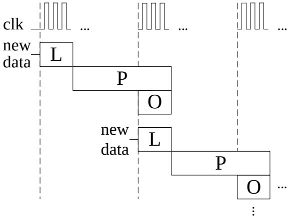
ARITH 2024

Outline

1. Introduction
   - Fast Fourier Transform
   - Perfect Shuffle and Perfect Unshuffle Algorithms
   - Input and Output Orders
   - Latency and Throughput in Memory-Based FFTs
   - Conventional Memory-Based FFTs
2. Proposed Access Patterns
   - Case 1: Natural Input - Bit-Reversed Output
   - Case 2: Scrambled Input - Bit-Reversed Output
   - Case 3: Bit-Reversed Input - Natural Output
   - Case 4: Bit-Reversed Input - Scrambled Output
3. Discussion and Comparison
4. Conclusion

Case 1: Natural input - Bit-reversed output



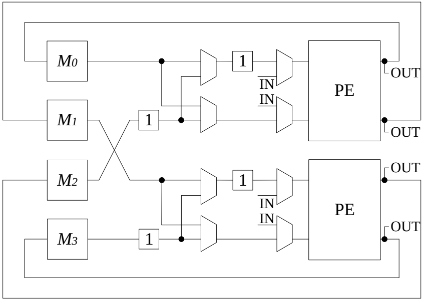


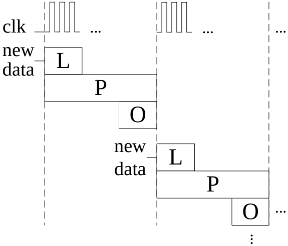
$$\Delta T_{FFT} = T_{LOAD} + T_{PROC} - T_{OUT} = \frac{N}{P} \cdot \log_2 N$$

$$T_{LAT} = T_{LOAD} + T_{PROC} = \frac{N}{P} \cdot (1 + \log_2 N)$$

$$Th = \frac{P}{\log_2 N}$$

Case 2: Scrambled input - Bit-reversed output



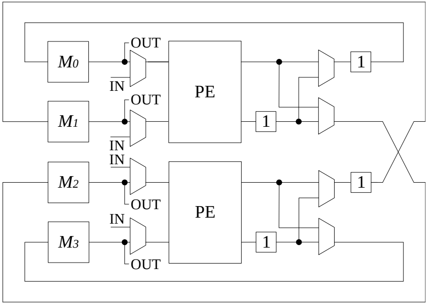


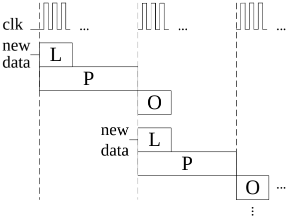
$$\Delta T_{FFT} = T_{PROC} = \frac{N}{P} \cdot \log_2 N$$

$$\boxed{T_{LAT} = T_{PROC} = \frac{N}{P} \cdot (\log_2 N)}$$

$$\boxed{Th = \frac{P}{\log_2 N}}$$

Case 3: Bit-reversed input - Natural output



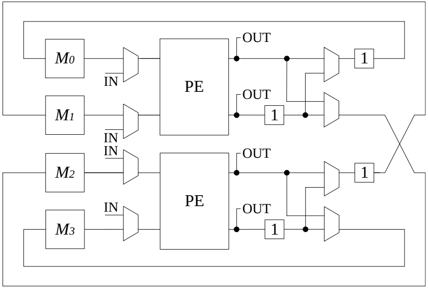


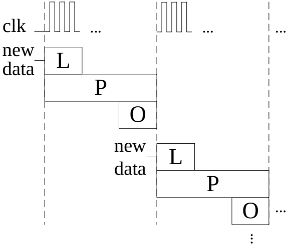
$$\Delta T_{FFT} = T_{PROC} = \frac{N}{P} \cdot \log_2 N$$

$$T_{LAT} = T_{PROC} + T_{OUT} = \frac{N}{P} \cdot (1 + \log_2 N)$$

$$Th = \frac{P}{\log_2 N}$$

Case 4: Bit-reversed input - Scrambled output





$$\Delta T_{FFT} = T_{PROC} = \frac{N}{P} \cdot \log_2 N$$

$$T_{LAT} = T_{PROC} = \frac{N}{P} \cdot (\log_2 N)$$

$$Th = \frac{P}{\log_2 N}$$

Outline

Comparison of all cases

Access Pattern	Order		Performance	
	Input	Output	Latency	Throughput
Conventional	Natural	Bit-reversed	$(2 + \log_2 N) \cdot N / P$	$P / (2 + \log_2 N)$
Case 1	Natural	Bit-reversed	$(1 + \log_2 N) \cdot N / P$	$P / (\log_2 N)$
Case 2	Scrambled	Bit-reversed	$(\log_2 N) \cdot N / P$	$P / (\log_2 N)$
Case 3	Bit-reversed	Natural	$(1 + \log_2 N) \cdot N / P$	$P / (\log_2 N)$
Case 4	Bit-reversed	Scrambled	$(\log_2 N) \cdot N / P$	$P / (\log_2 N)$

Outline

Conclusion

- The new access patterns have been proposed for the memory-based FFTs.
- The proposed architectures are based on the perfect shuffle and perfect unshuffle.
- All patterns allow for reducing latency and increasing throughput with respect to conventional approaches.
- Improvements are achieved without increasing the number of hardware resources of the architecture.

Thank you!