Lecture

15

ECE 216B

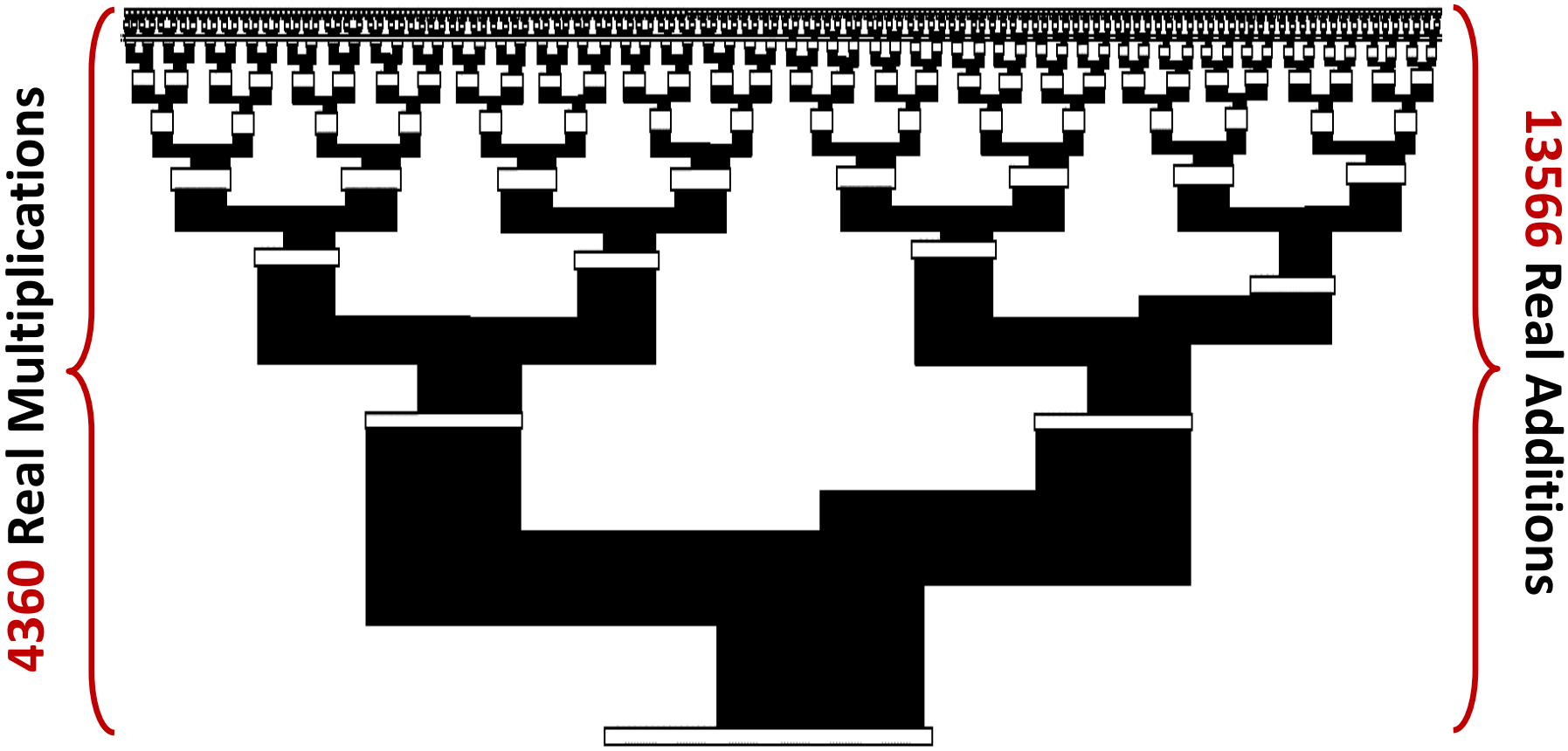
# FFT Architecture Optimization

Prof. Dejan Marković

ee216b@gmail.com

# 512-pt FFT: Synthesis-Time Bottleneck

- **Direct-mapped architecture took 8 hours to synthesize, retiming unfinished after 2.5 days**
  - Difficult to explore the design space using synthesis



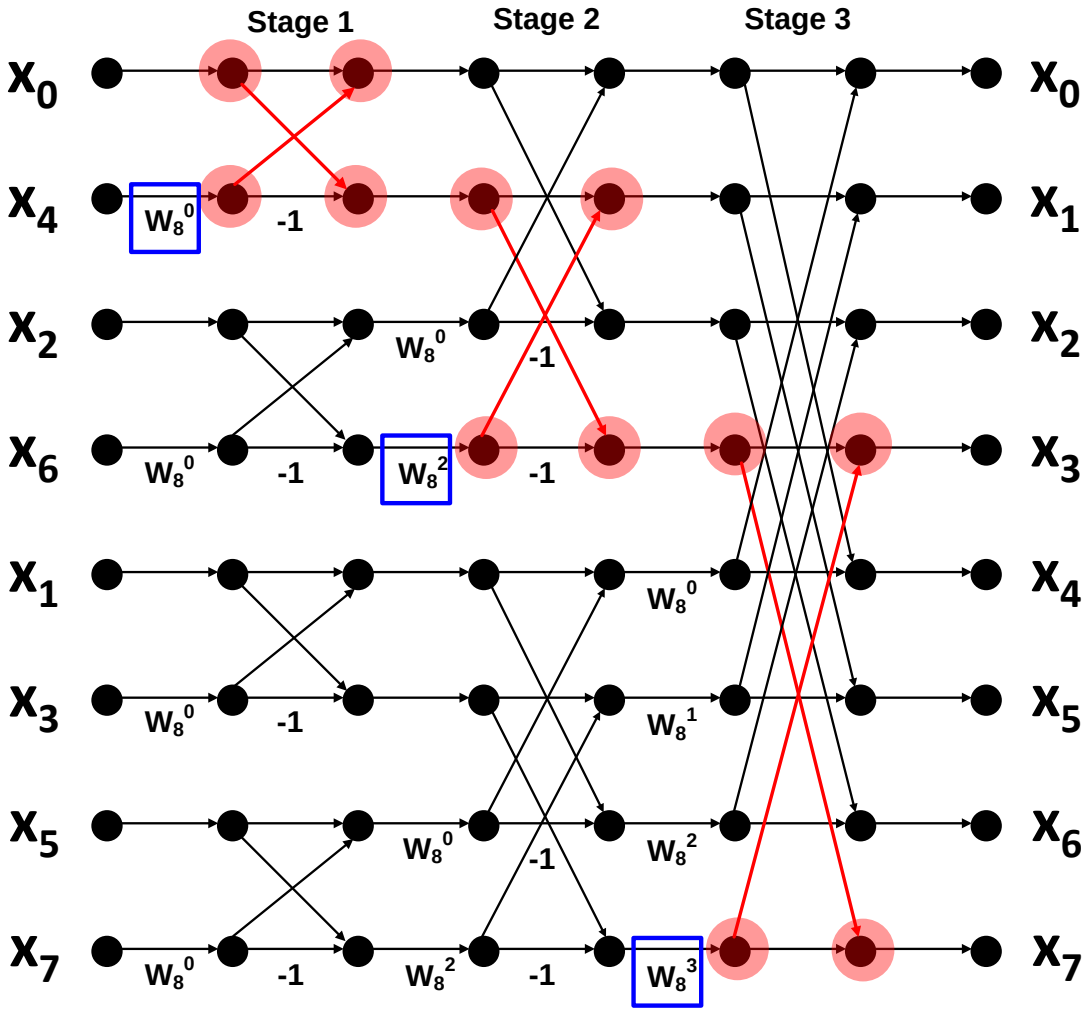
# Agenda

- **FFT hardware estimation**
  - Structured model
- **Basic FFT architectures**
  - Time-mux, direct-mapped
  - SDF, MDC
- **Architecture optimization**
  - Fixed FFT size
  - Flexible FFT size
  - CORDIC example

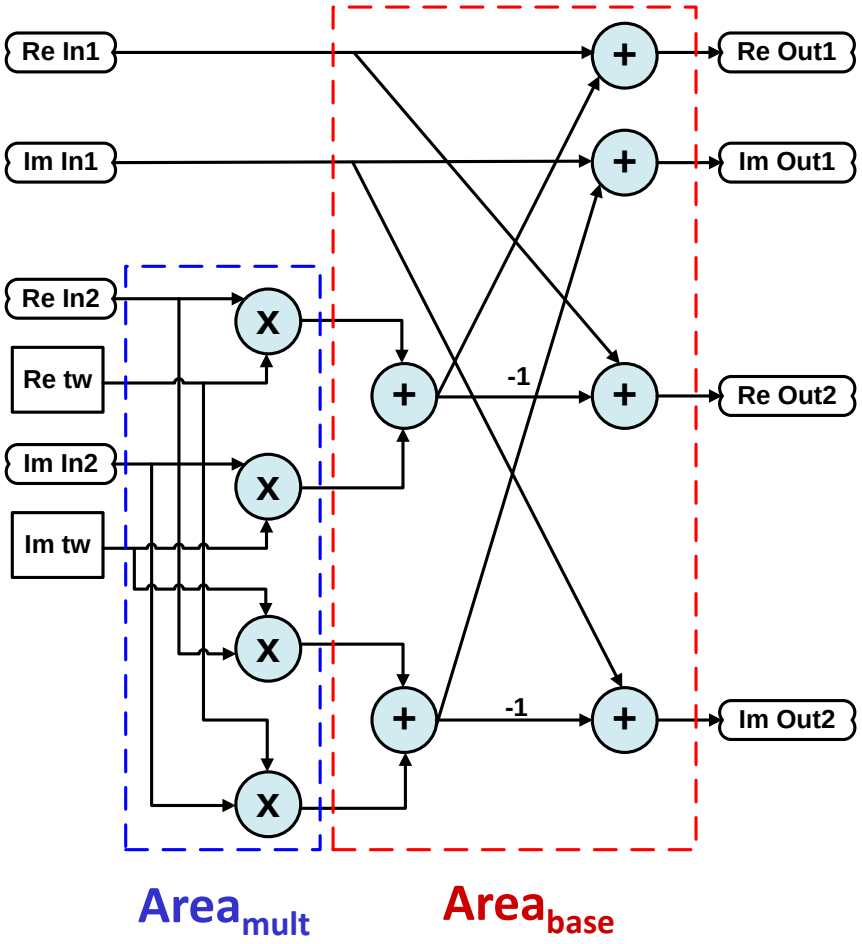
# FFT Model for
# Hardware Estimation

# Exploiting Hierarchy

**Treat each stage as a group of butterfly-2 structures, with varying twiddle factors**

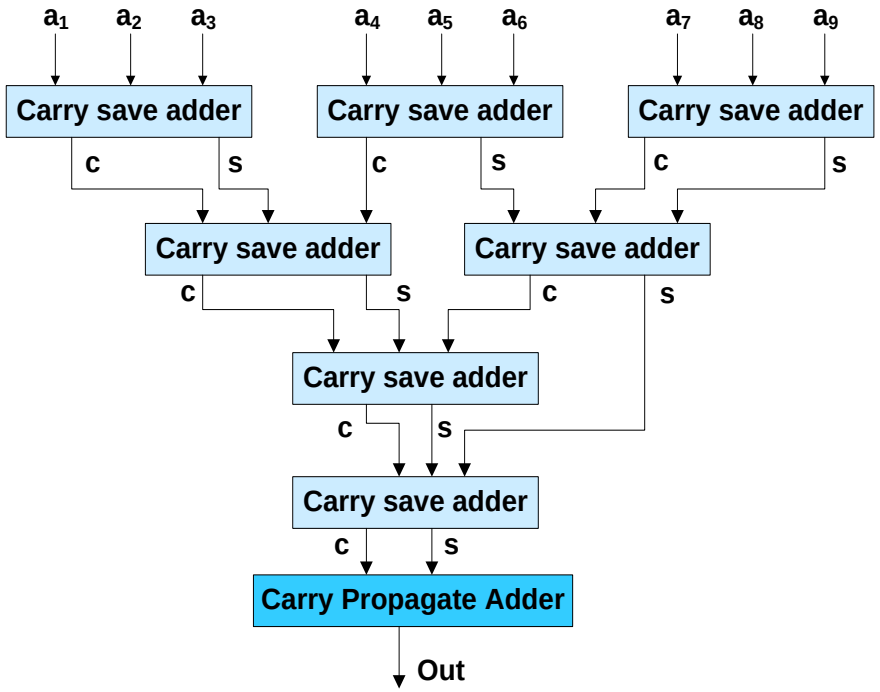


# Modeling the Radix-2 Butterfly: Area



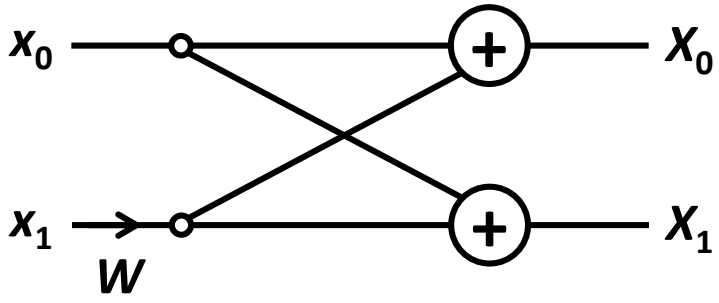

## Multiplication with CSA

Shifted version of the input



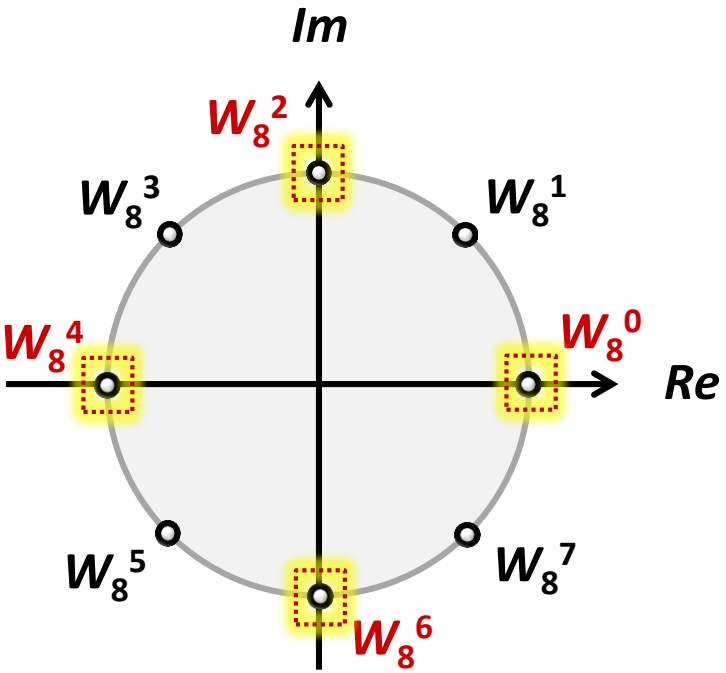

$$Area_{total} = Area_{base} + Area_{mult}$$

# Further Simplifications




$$W = e^{-j\frac{2\pi nk}{N}}$$

$n = 1, k = 1$
in this example

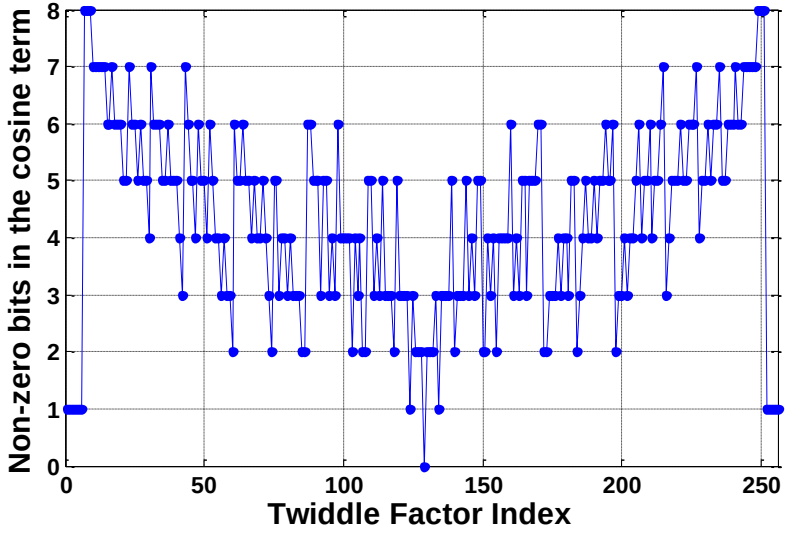


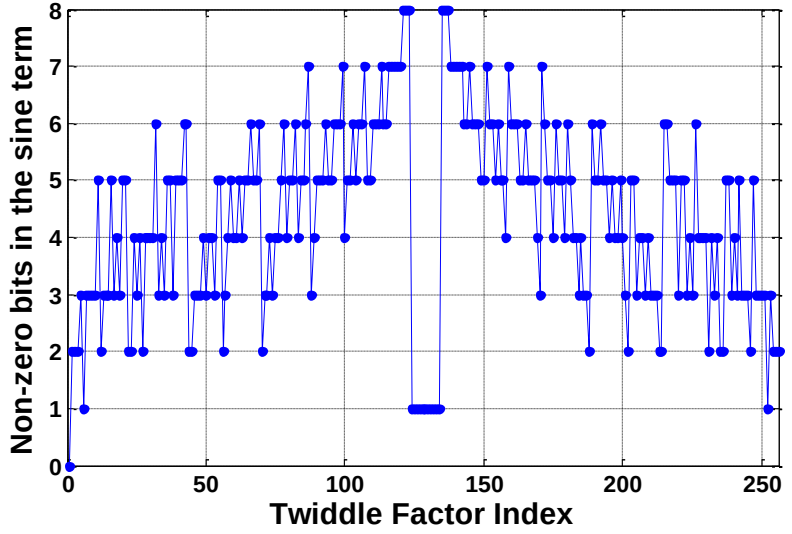

**Example: $N = 8$**

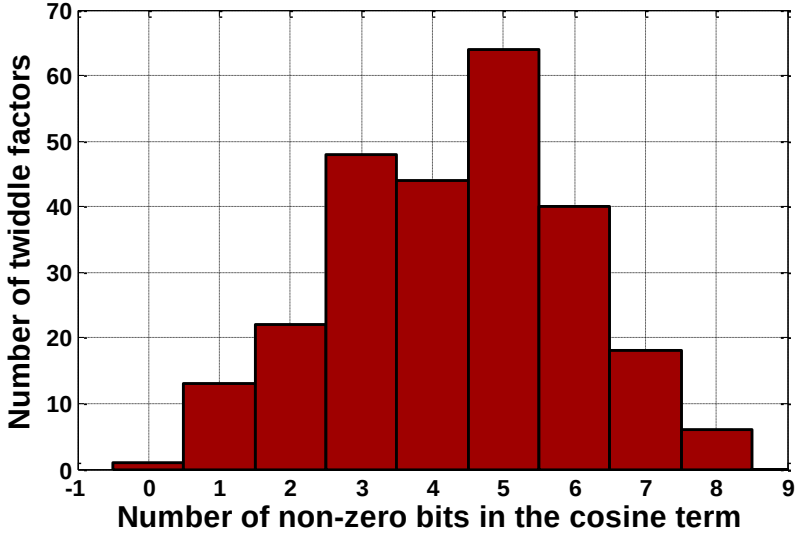
- **Considerably simpler**
  - Sign inversion
  - Swap Re/Im (or both)

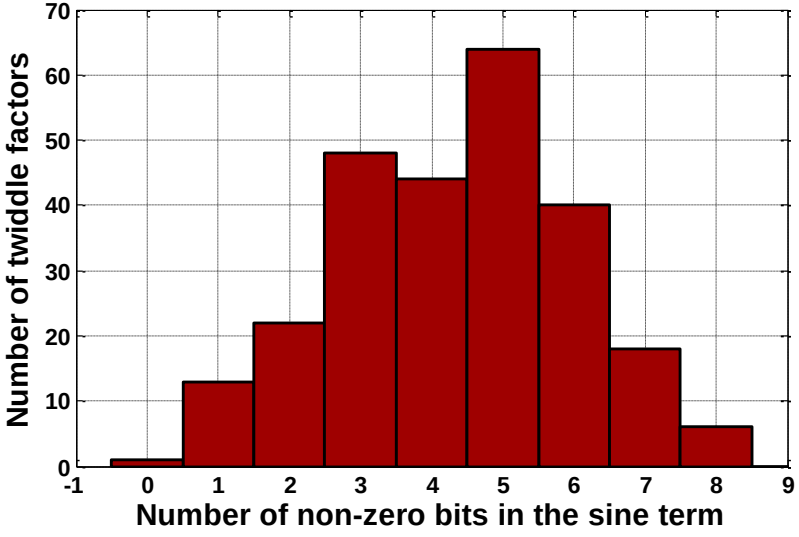
# Twiddle-Factor Area Estimation

## Non-zero bits 0-8 for the sine and cosine terms | observe symmetry

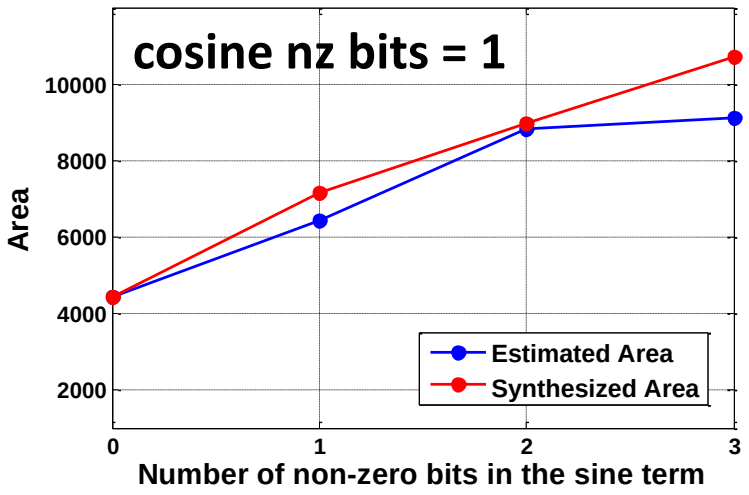


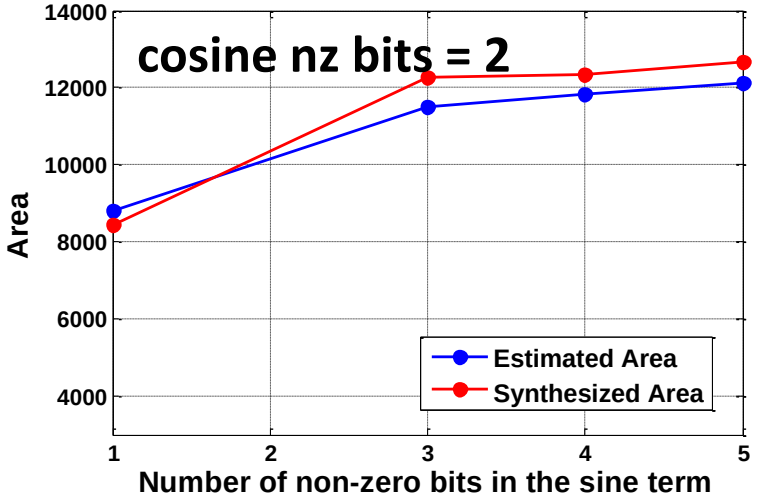



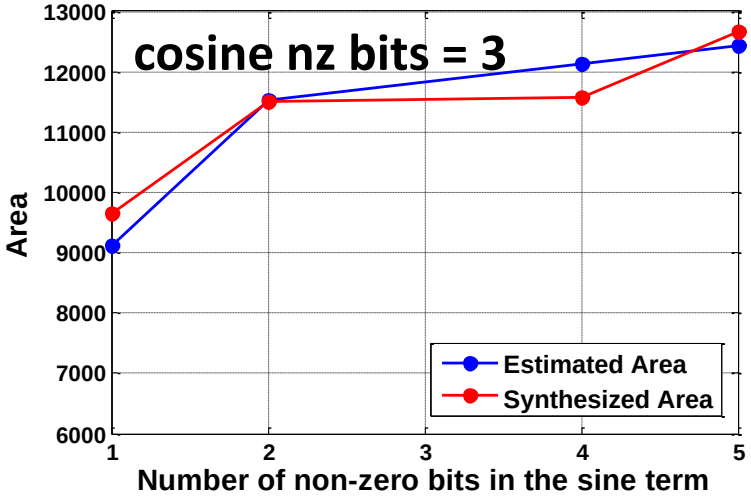



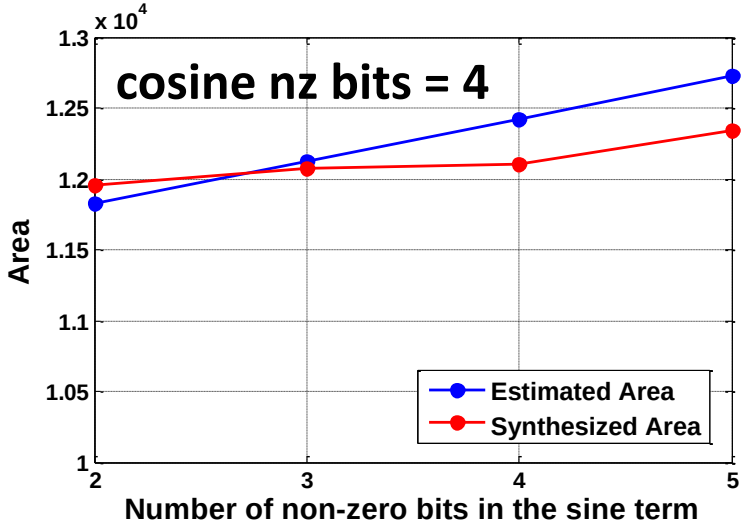



# Area Estimates from Synthesis



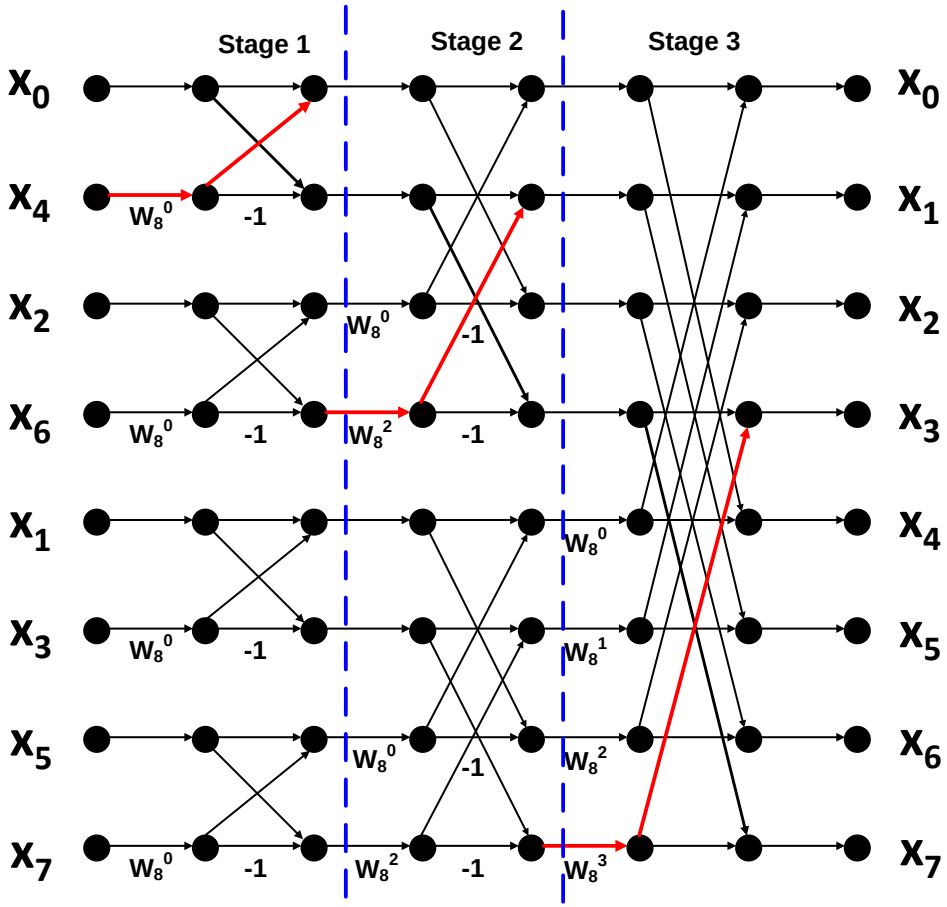




**Average error in area estimate: ~ 5%**

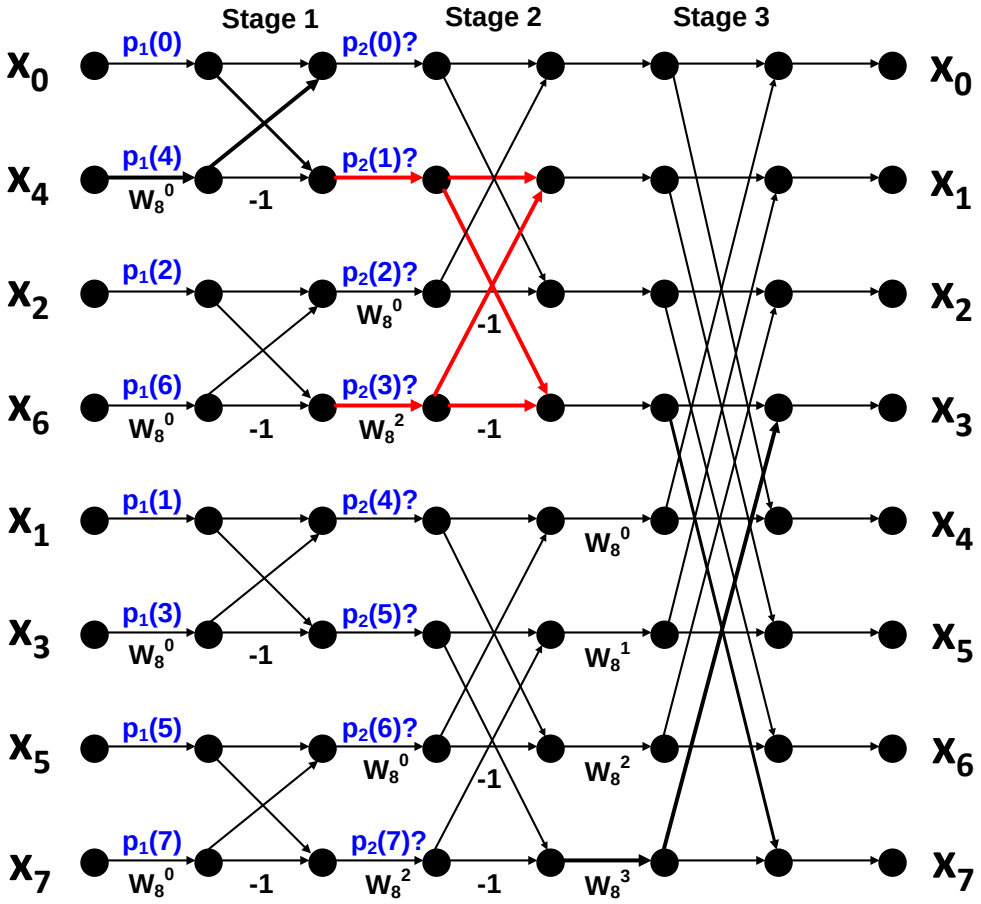
# FFT Timing Estimation




- **Critical path in each stage: delay of most complex butterfly**
  - Varied between 1.6 ns to 2 ns per stage (90-nm CMOS)
- **Inserting pipeline registers between stages reduces the delay**

# FFT Energy Estimation

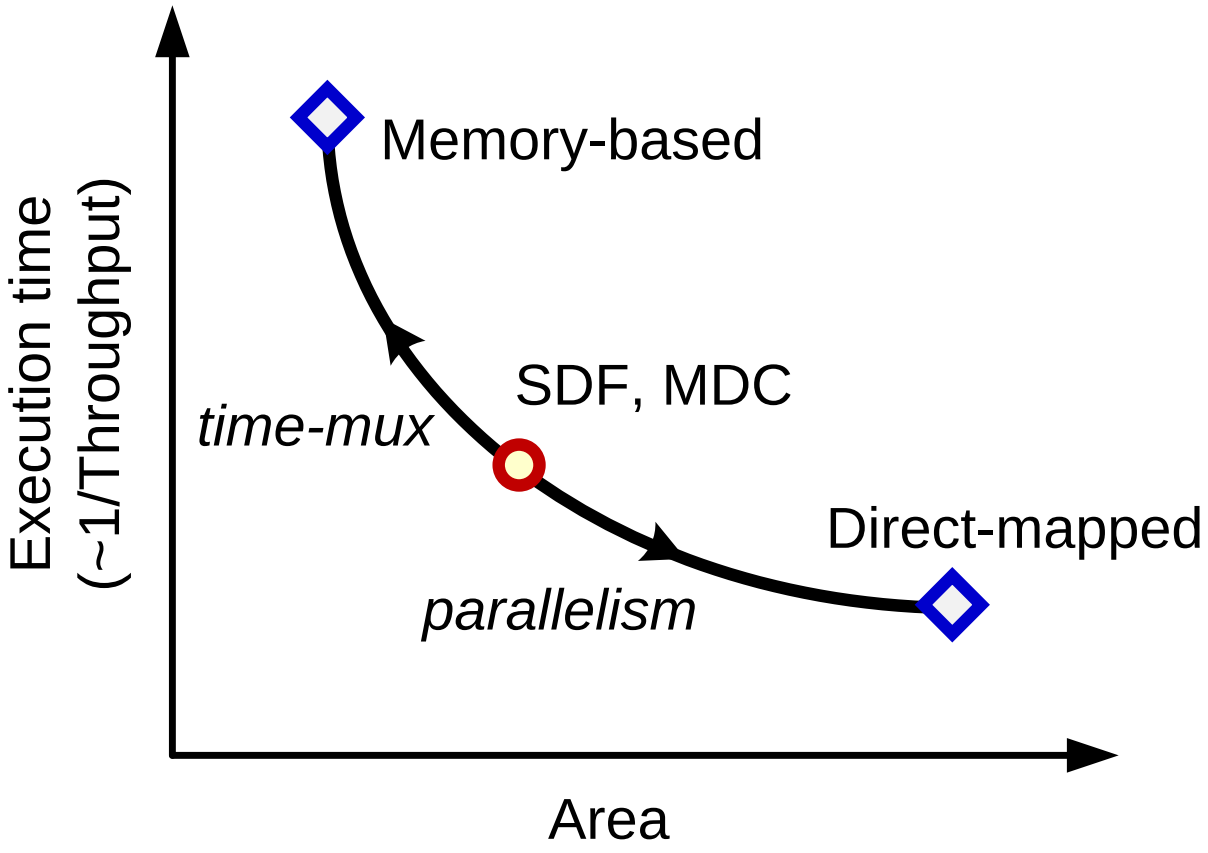
- Time-consuming switch-level (MATLAB) or gate-level (RC) sims
- Derive analytical expressions for each butterfly as a function of input switching activity and nonzero bits in the twiddle factor



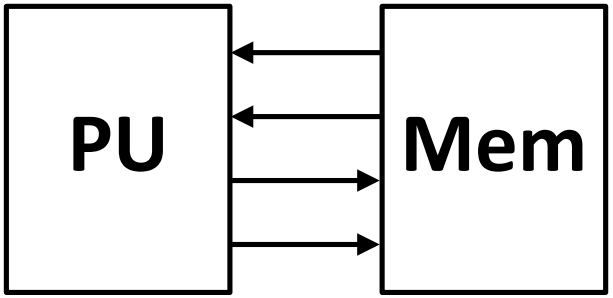
# FFT Architectures:

## A Space-Time Tradeoff

# FFT Architectures

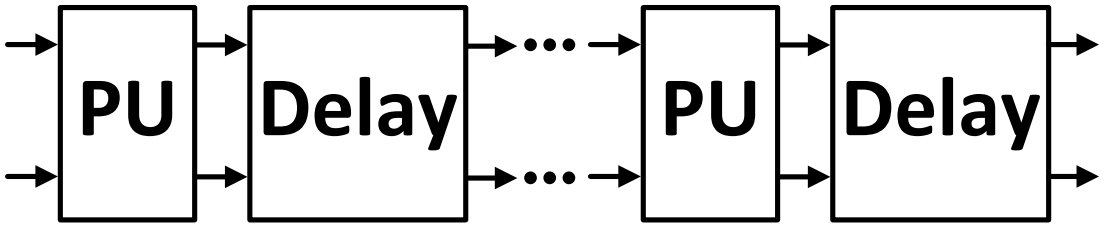


# Basic Architectures (1/2)




## Memory-based

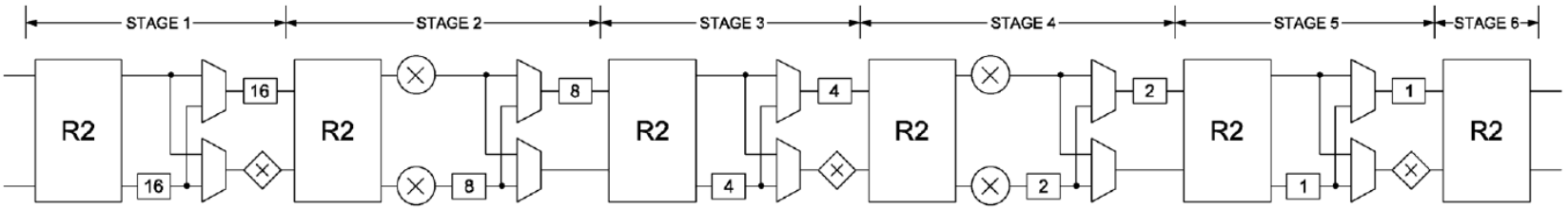
- One PU
- Large Mem



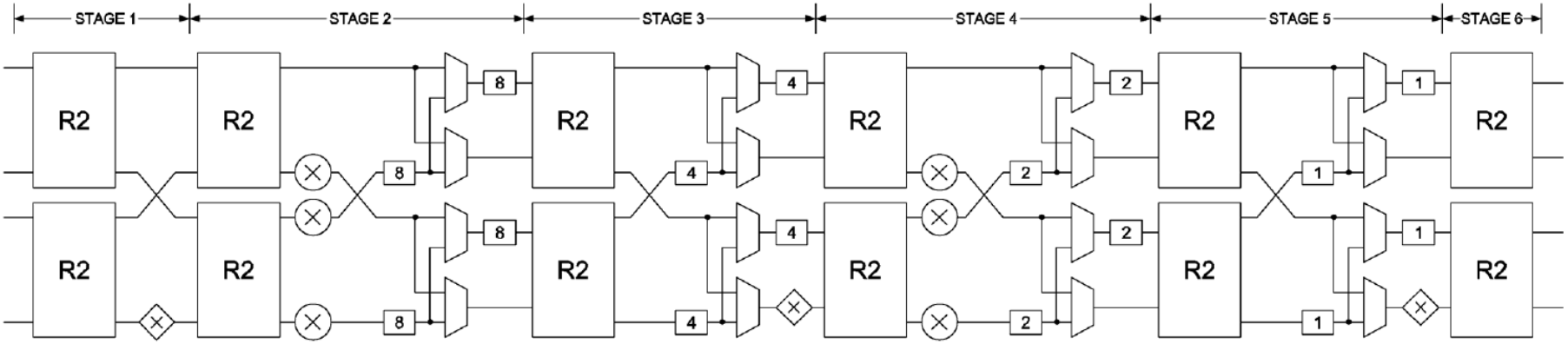

## Multipath Delay Commutator

- 1/r utilization
- Feed-forward

# Parallel MDC FFT



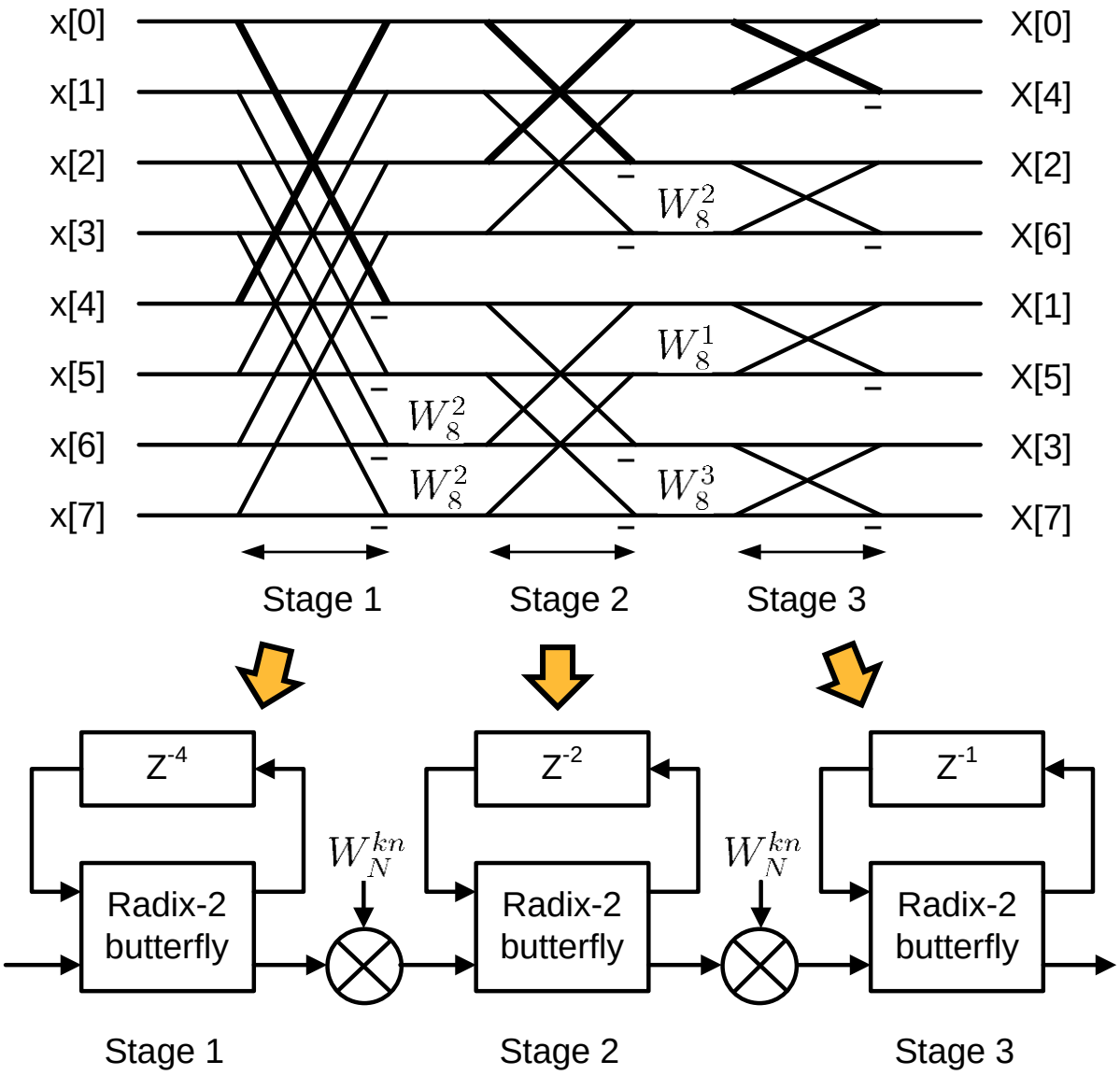

**2-way parallel radix-$2^2$ MDC FFT**



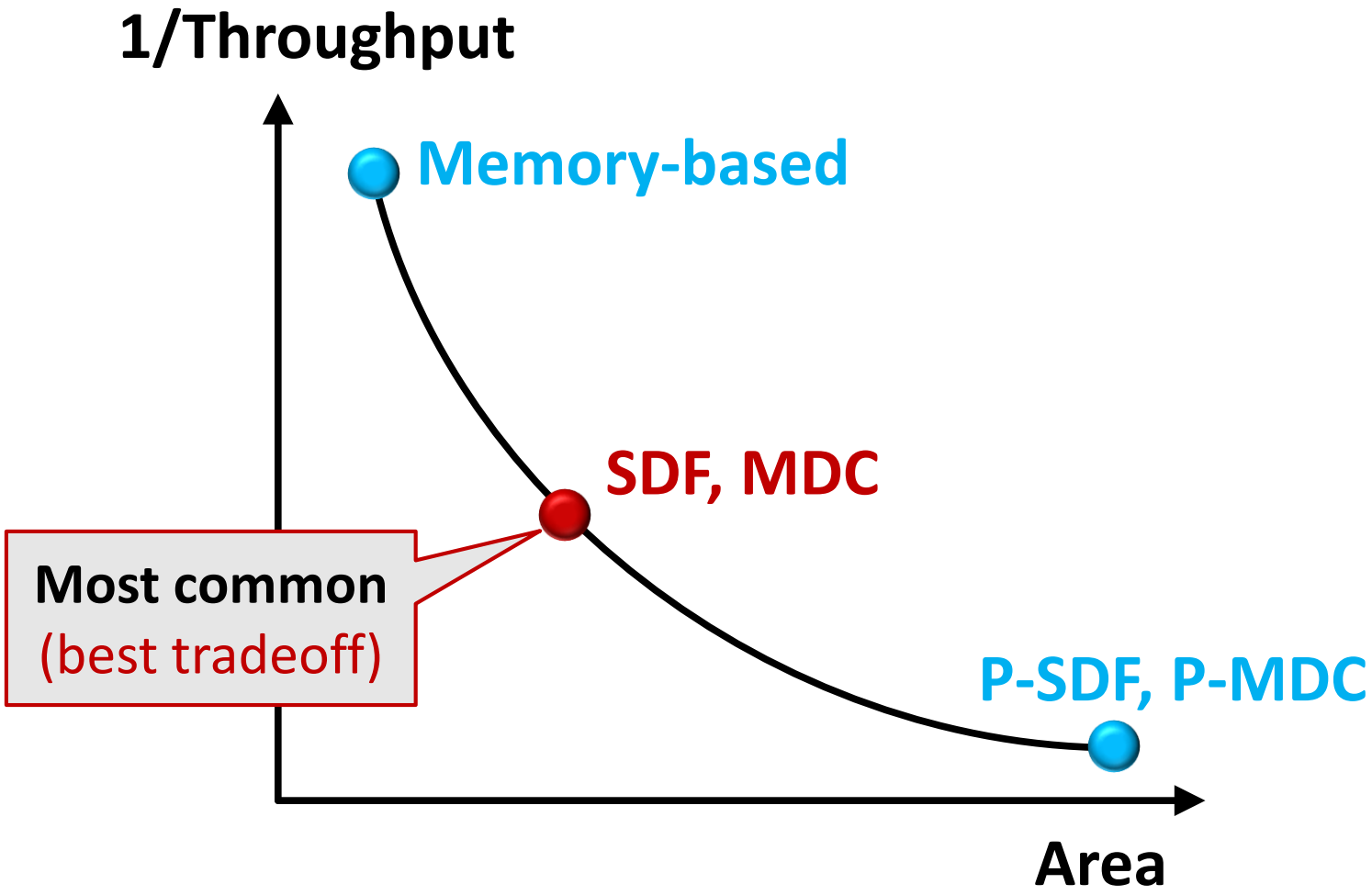

**4-way parallel radix-$2^2$ MDC FFT**

**Also allows mix-radix implementation**

# Single-Path Delay Feedback (SDF) FFT



# FFT Design Space



# SDF-based FFT:

## Architecture Optimization

# FFT Optimization

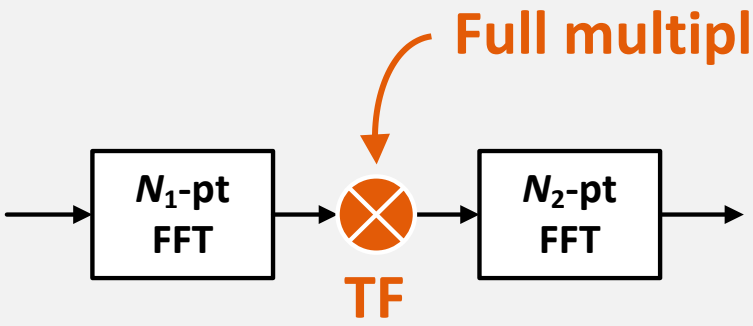
| Opt. Problem | For a **given performance**<br>**Minimize energy** and/or **area** |
|---|---|

*Variables / Techniques*

- **Radix factorization**
- **Parallelism**
- **DL realization**

# FFT Factorization:

## $N = N_1 \times N_2$




Algorithm

Hardware

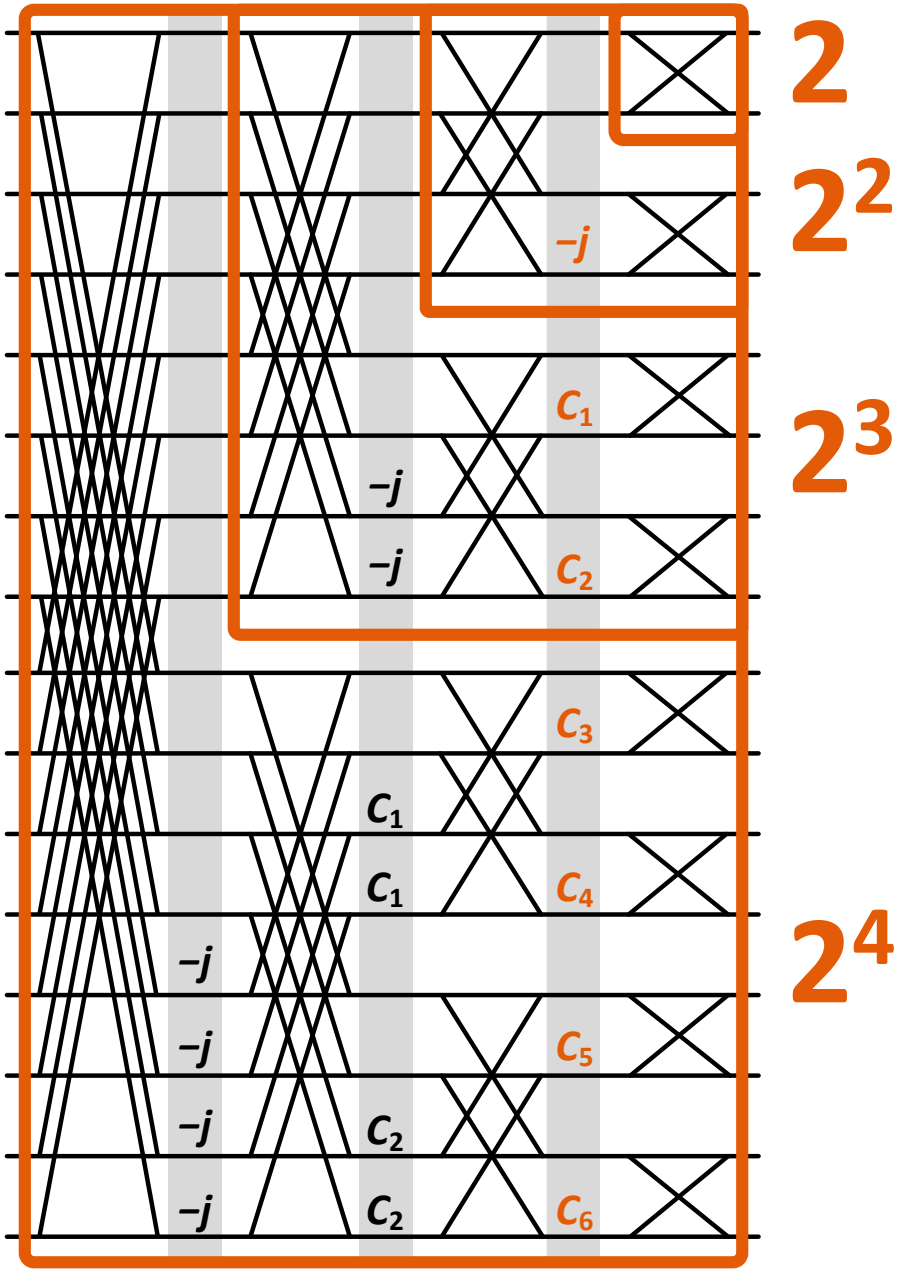
$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}}$$

$$= \sum_{n_2=0}^{N_2-1} \left\{ \left\{ \sum_{n_1=0}^{N_1-1} x(n_1 N_2 + n_2) e^{-j\frac{2\pi n_1 k_1}{N_1}} \right\} e^{-j\frac{2\pi n_2 k_1}{N_1 N_2}} \right\} e^{-j\frac{2\pi n_2 k_2}{N_2}}$$

Twiddle factor (TF)

$N_1$-point DFT

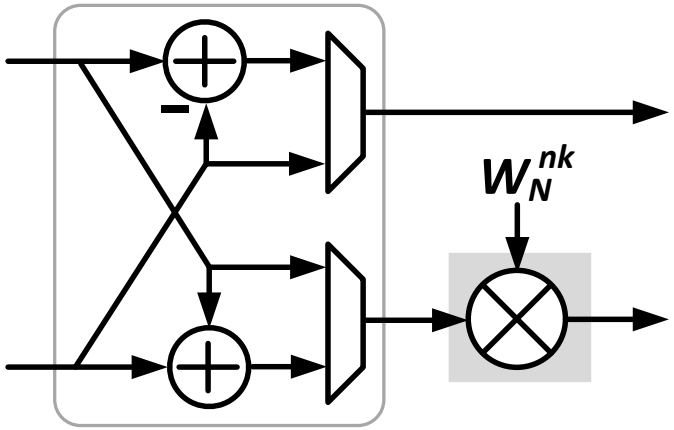
$N_2$-point DFT



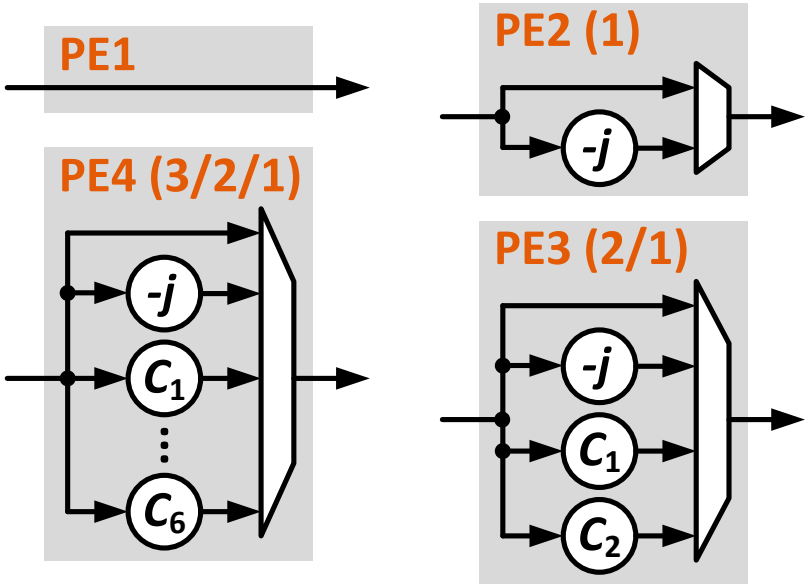

# Radix
## Factorization

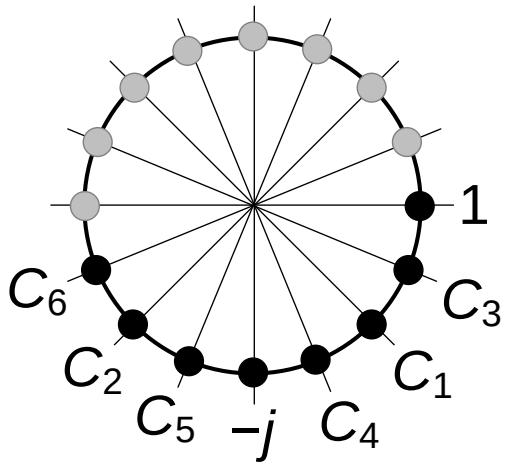
| Radix | CONSTANT multipliers |
|---|---|
| $2$ | $1$ |
| $2^2$ | $1, -j$ |
| $2^3$ | $1, -j, C_1, C_2$ |
| $2^4$ | $1, -j, C_1, C_2, C_3, C_4, C_5, C_6$ |

# Flexible Processing Element (PE)

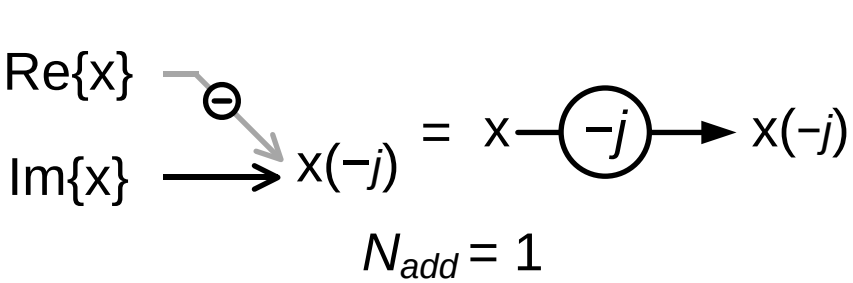


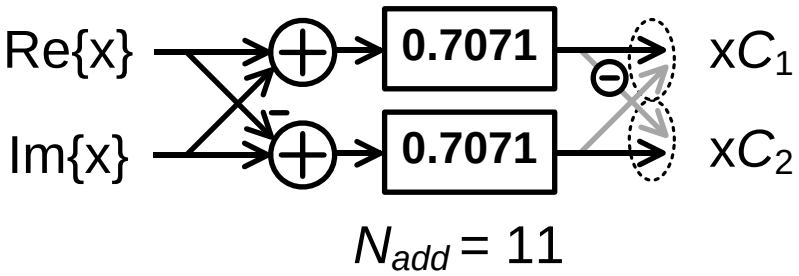

| Radix | Implementation |
|---|---|
| 2 | PE1 |
| 4 | PE2+PE1 |
| 8 | PE2+PE3+PE1 |
| 16 | PE2+PE3+PE4+PE1 |

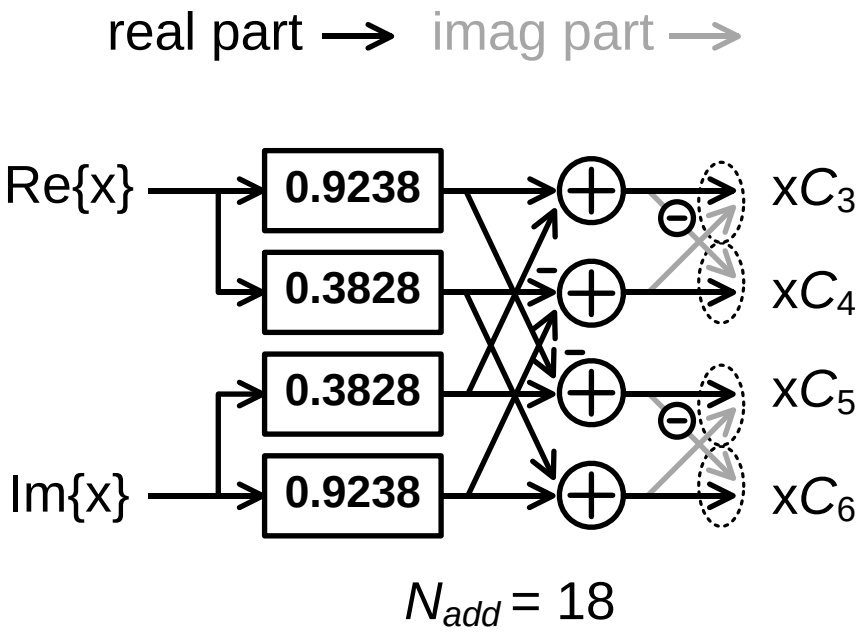


# Only 3 Constant Factors to Build PUs




| Factor | CSD realization | Adders |
|---|---|---|
| **0.7071** | $2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8}$ | 4 |
| **0.9238** | $2^{0} - 2^{-3} + 2^{-5} + 2^{-6} + 2^{-9}$ | 4 |
| **0.3828** | $2^{-2} + 2^{-3} + 2^{-7}$ | 2 |

# Hardware Cost of PUs

| Radix | Implementation | Butterfly adders | Const "x" (equiv. "+") |
|---|---|---|---|
| 2 | PU1 | 4 | **0** |
| $2^2$ | PU2+PU1 | 8 | **1** |
| $2^3$ | PU2+PU3+PU1 | 12 | **13** |
| $2^4$ | PU2+PU3+PU4+PU1 | 16 | **43** |

- **Normalize to an equivalent adder** (of certain WL)
- **Calculate relative cost for each radix**

# Impact of TW Wordlength

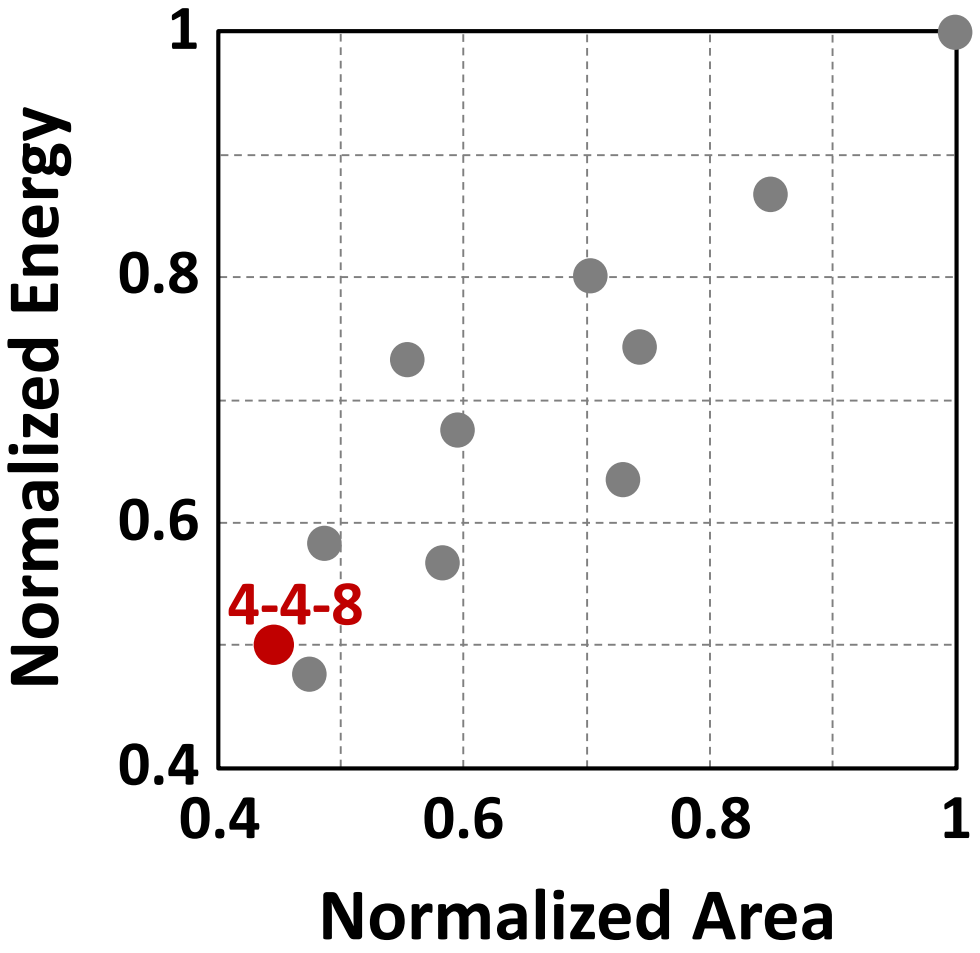
| FFT size | 4 points | | | | | 8 points | | | | | 16 points | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TW bits | 8 | 10 | **12** | 14 | 16 | 8 | 10 | **12** | 14 | 16 | 8 | 10 | **12** | 14 | 16 |
| Radix-2 | 29 | 35 | 41 | 47 | 53 | 58 | 70 | 82 | 94 | 106 | 87 | 105 | 123 | 141 | 159 |
| Radix-$2^2$ | 1 | 1 | 1 | 1 | 1 | - | - | - | - | - | 31 | 37 | 43 | 49 | 55 |
| Radix-$2^3$ | - | - | - | - | - | 11 | 13 | 13 | 13 | 15 | - | - | - | - | - |
| Radix-$2^4$ | - | - | - | - | - | - | - | - | - | - | 37 | 43 | 43 | 43 | 49 |

- **Complexity: the number of equivalent adders**
- **Example: 16-point FFT**
  - Two radix $2^2$ stages (1 add/stage)
  - Inter-stage full multiplier (3 mults + 5 adds)
  - Complexity (8-bit TW) = 2 + (3·8 + 5) = 31

legend: Uses full multipliers

# Radix Structure of 128-pt FFT

| Arch | R2 | R4 | R8 | R16 |
|---|---|---|---|---|
| A1 | 7 | | | |
| A2 | 5 | 1 | | |
| A3 | 3 | 2 | | |
| A4 | 1 | 3 | | |
| A5 | 4 | | 1 | |
| A6 | 2 | 1 | 1 | |
| A7 | 1 | | 2 | |
| A8 | | 2 | 1 | |
| A9 | 3 | | | 1 |
| A10 | 1 | 1 | | 1 |
| A11 | | | 1 | 1 |

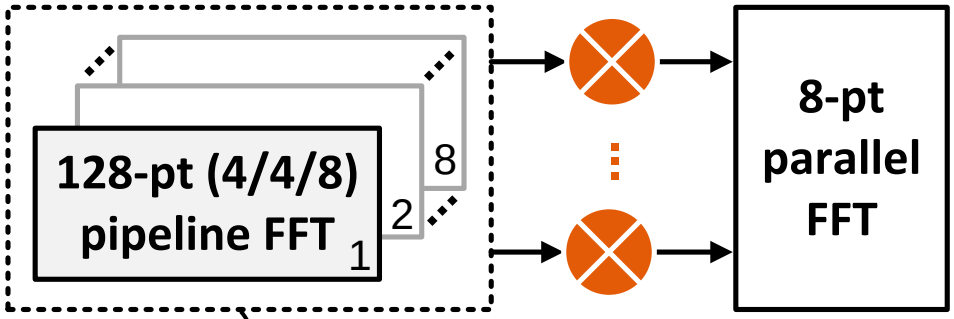


# Radix-Optimized 1024-pt FFT

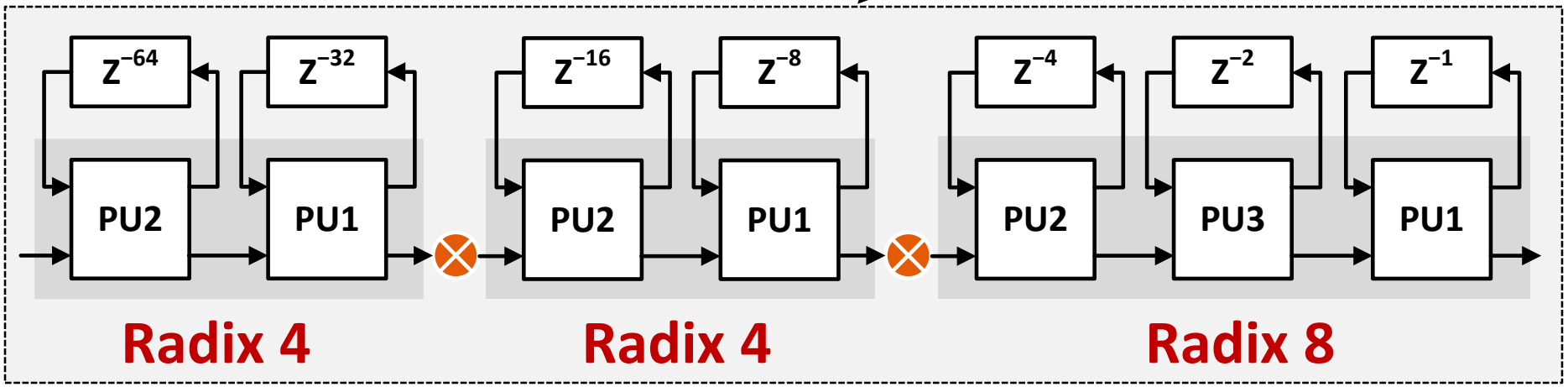
**Final architecture**

$N_1 = 128$, $N_2 = 8$

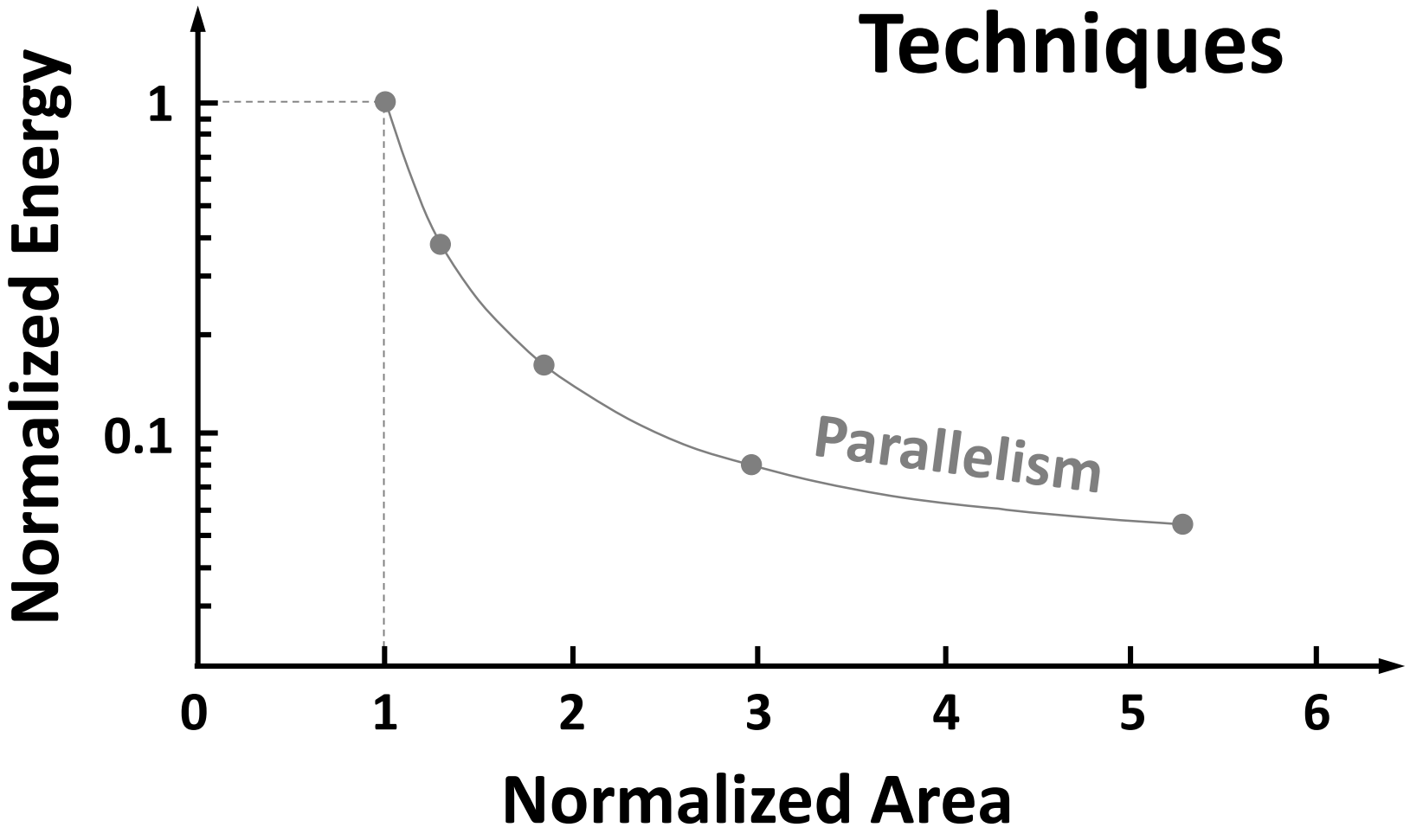
(out of 68 arch.)



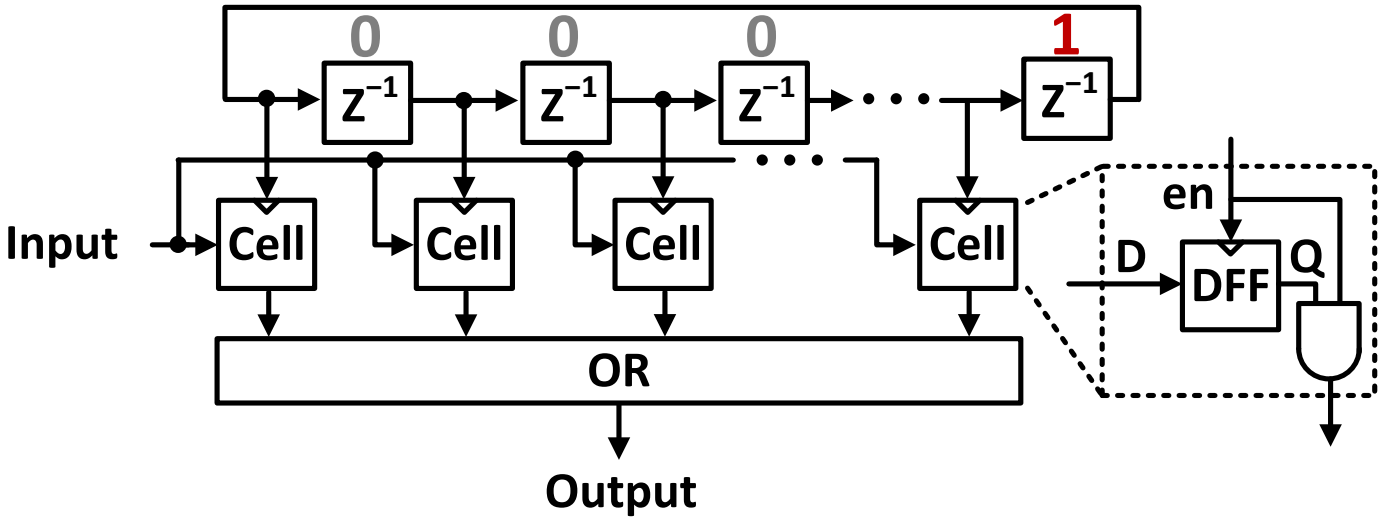

**128-pt FFT**



# Optimization Techniques

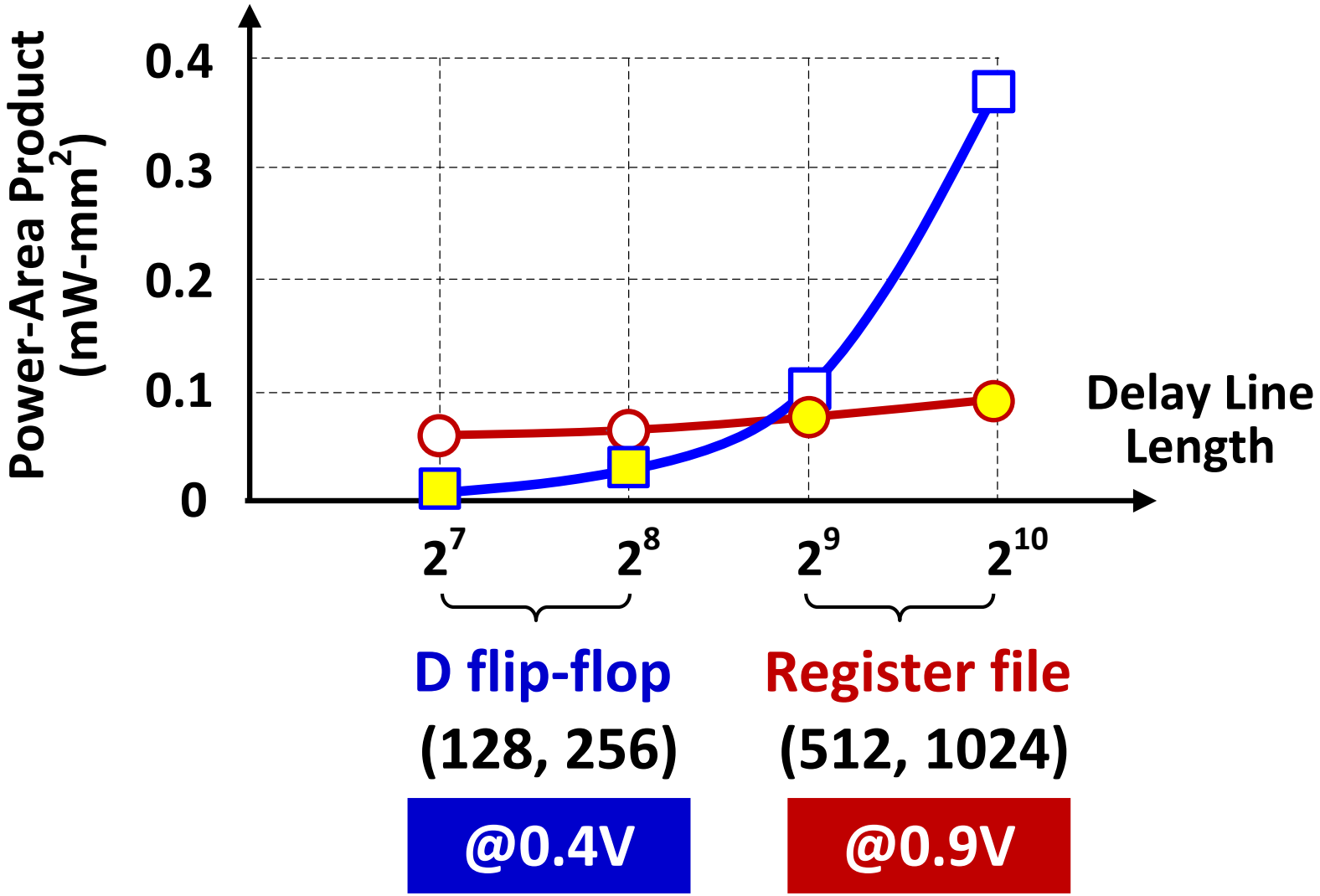


# Pointer-based DFF Delay Line

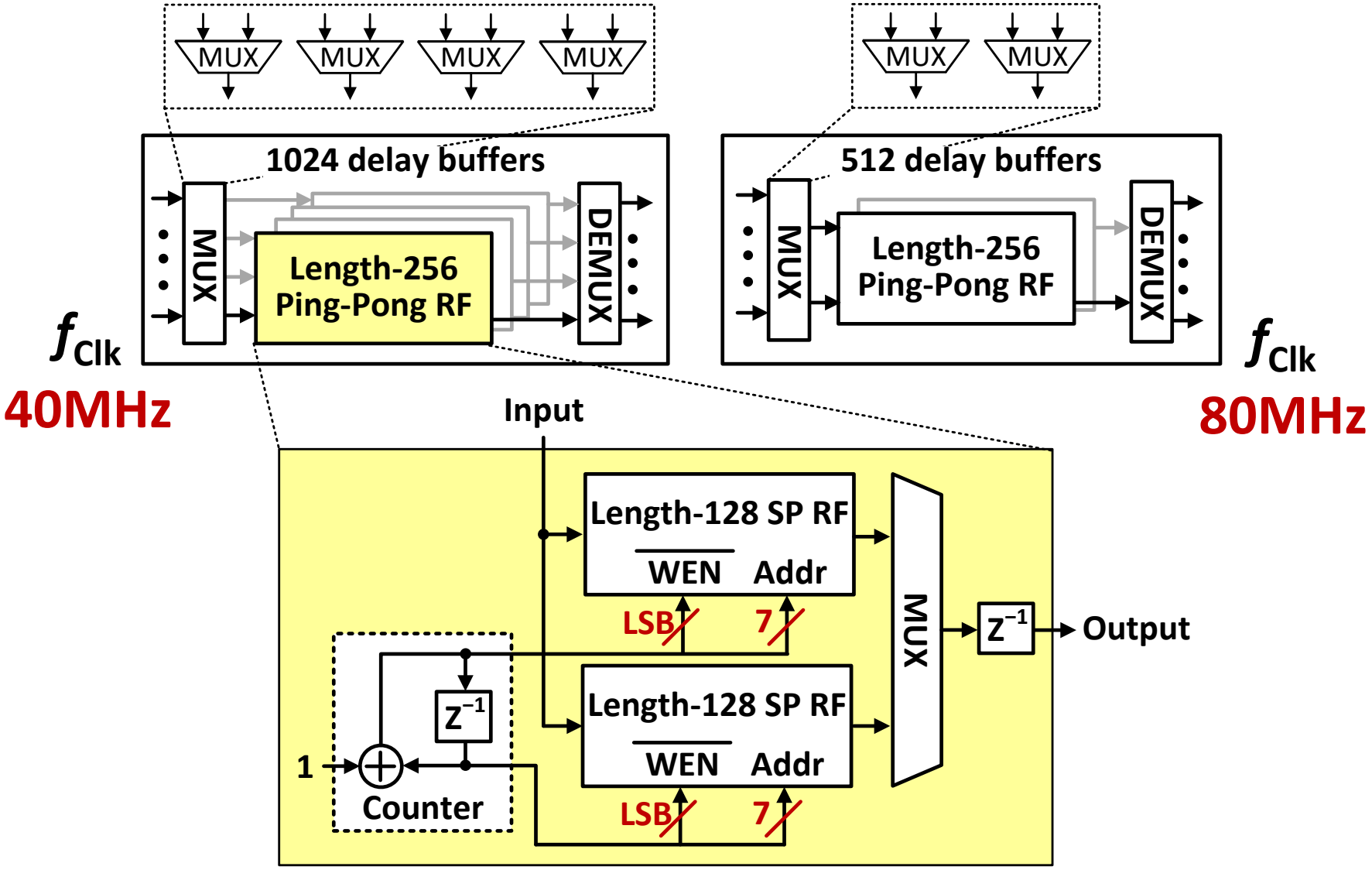



- **Shift DL chooses a register cell to read/write**
  - Remaining cells are clock-gated to save power
- **DFF good for small size**
  - No periphery
  - Can scale $V_{DD}$

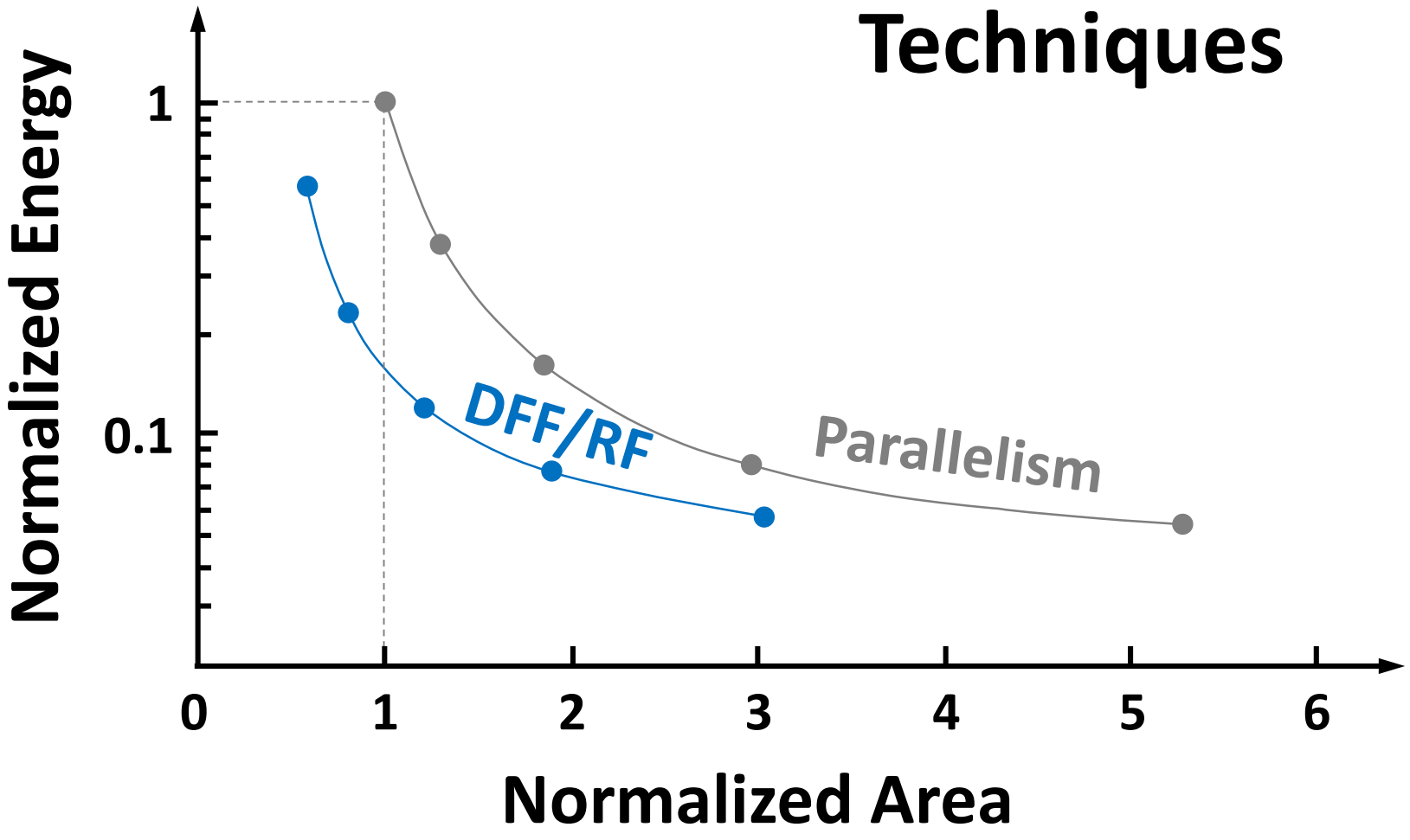
# Optimal Delay Line Partition



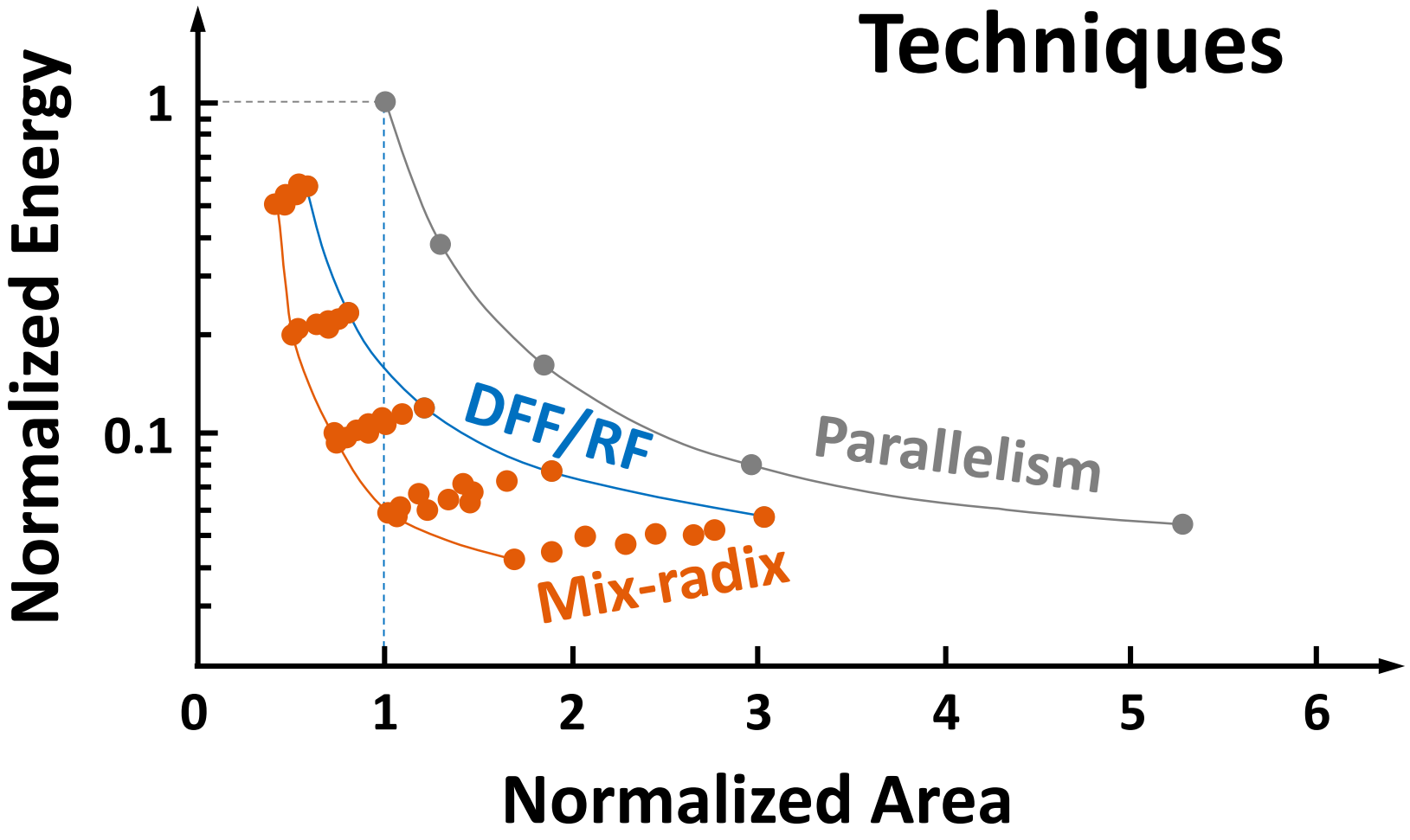
# Ping-pong Register File Delay Line

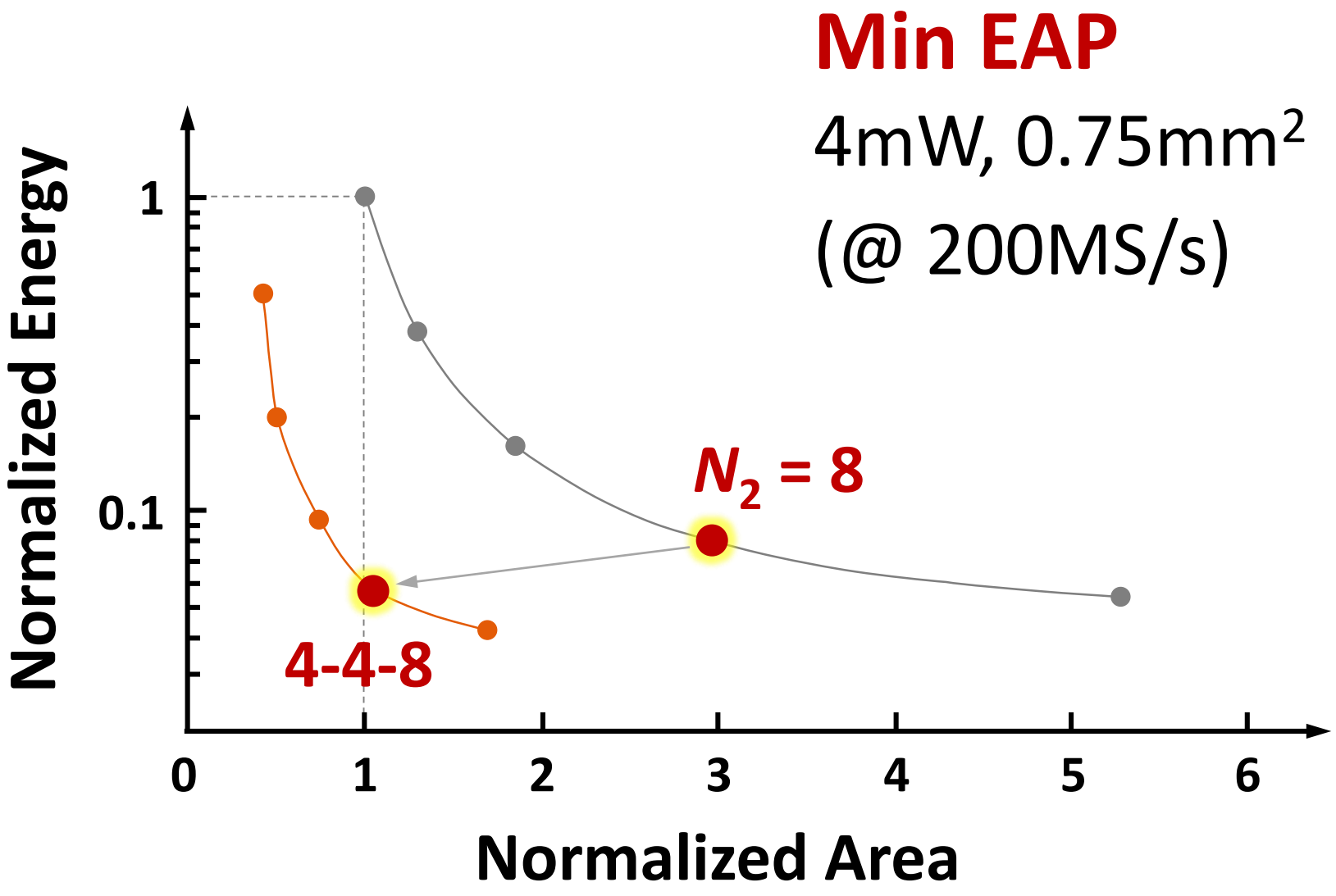


# Optimization Techniques



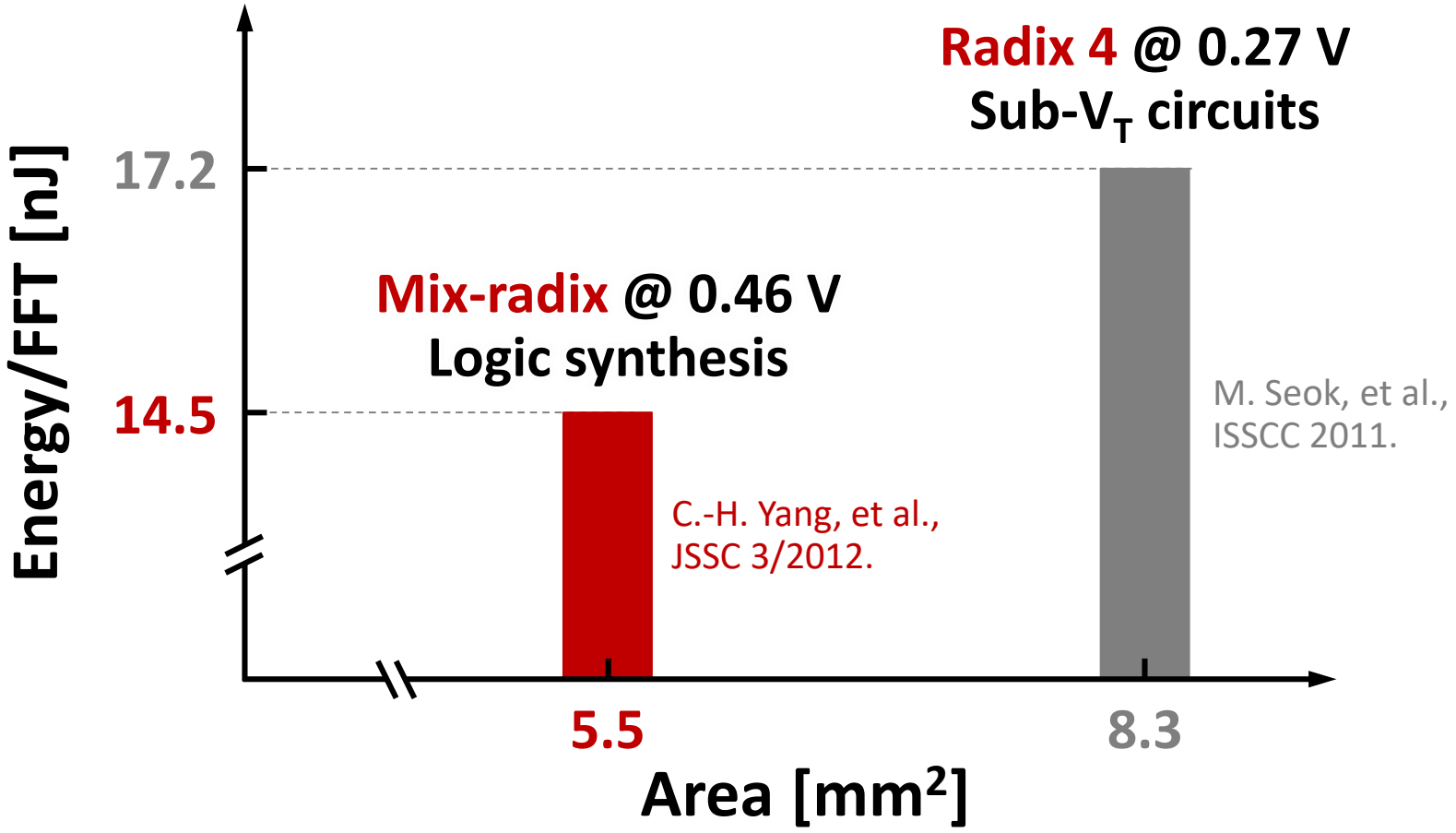
# Optimization Techniques




Min EAP
4mW, 0.75mm²
(@ 200MS/s)
Normalized Energy
1
0.1
$N_2$ = 8
4-4-8
0
1
2
3
4
5
6
Normalized Area

# Mix-Radix vs. Sub-$V_T$ Circuits

1024 points, 240 MS/s



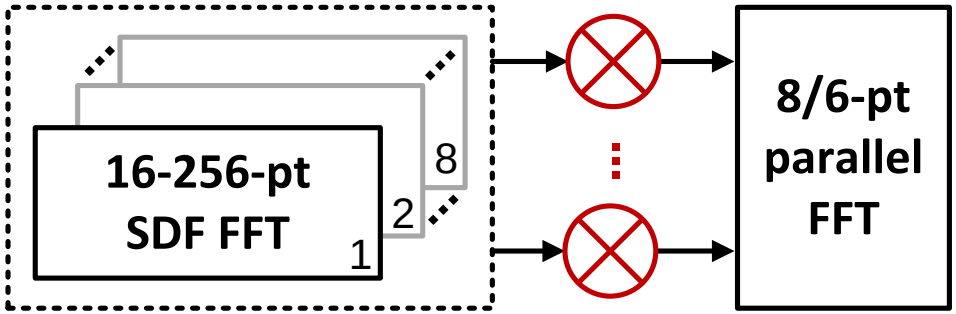
# Flexible FFT:
# LTE Example

# 3GPP-LTE Specifications

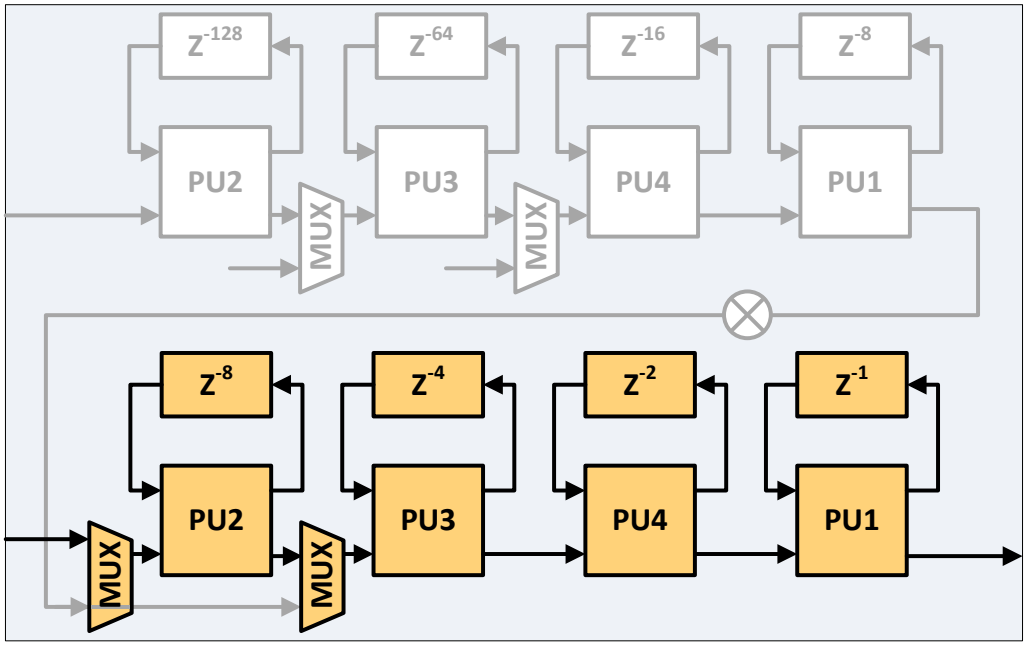
| Parameter | Configuration | | | | | |
|---|---|---|---|---|---|---|
| FFT size | 128 | 256 | 512 | 1024 | 1536 | 2048 |
| BW (MHz) | 1.25 | 2.5 | 5 | 10 | 15 | 20 |

- **Consider all configurations to find best architecture**
- **15 MHz band requires support for radix 3**

# Flexible FFT (128-2048 pts) for 3GPP-LTE



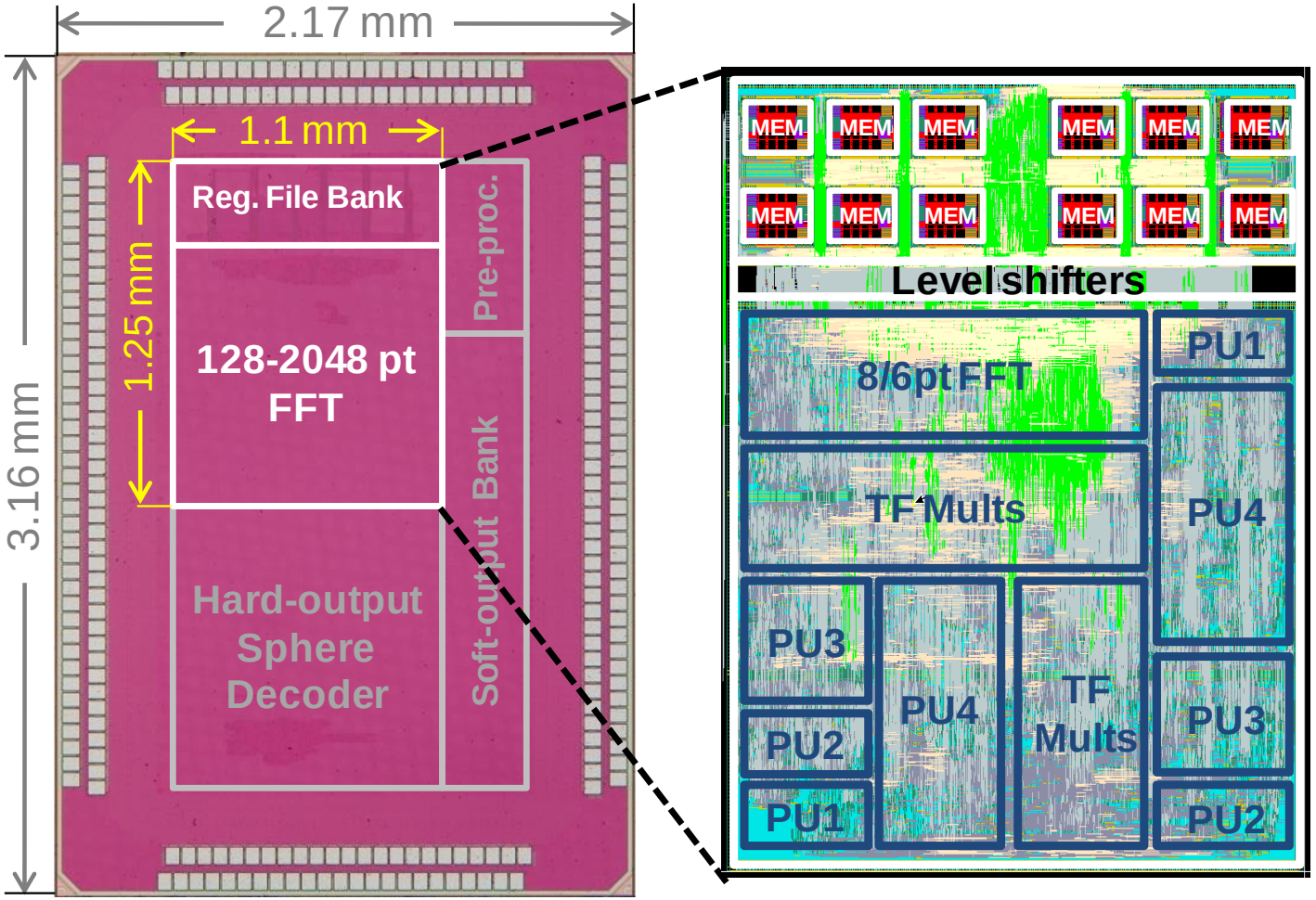

**Flexible FFT:**
$N_1$ = 16 to 256
$N_2$ = 8 or 6




16-pt FFT

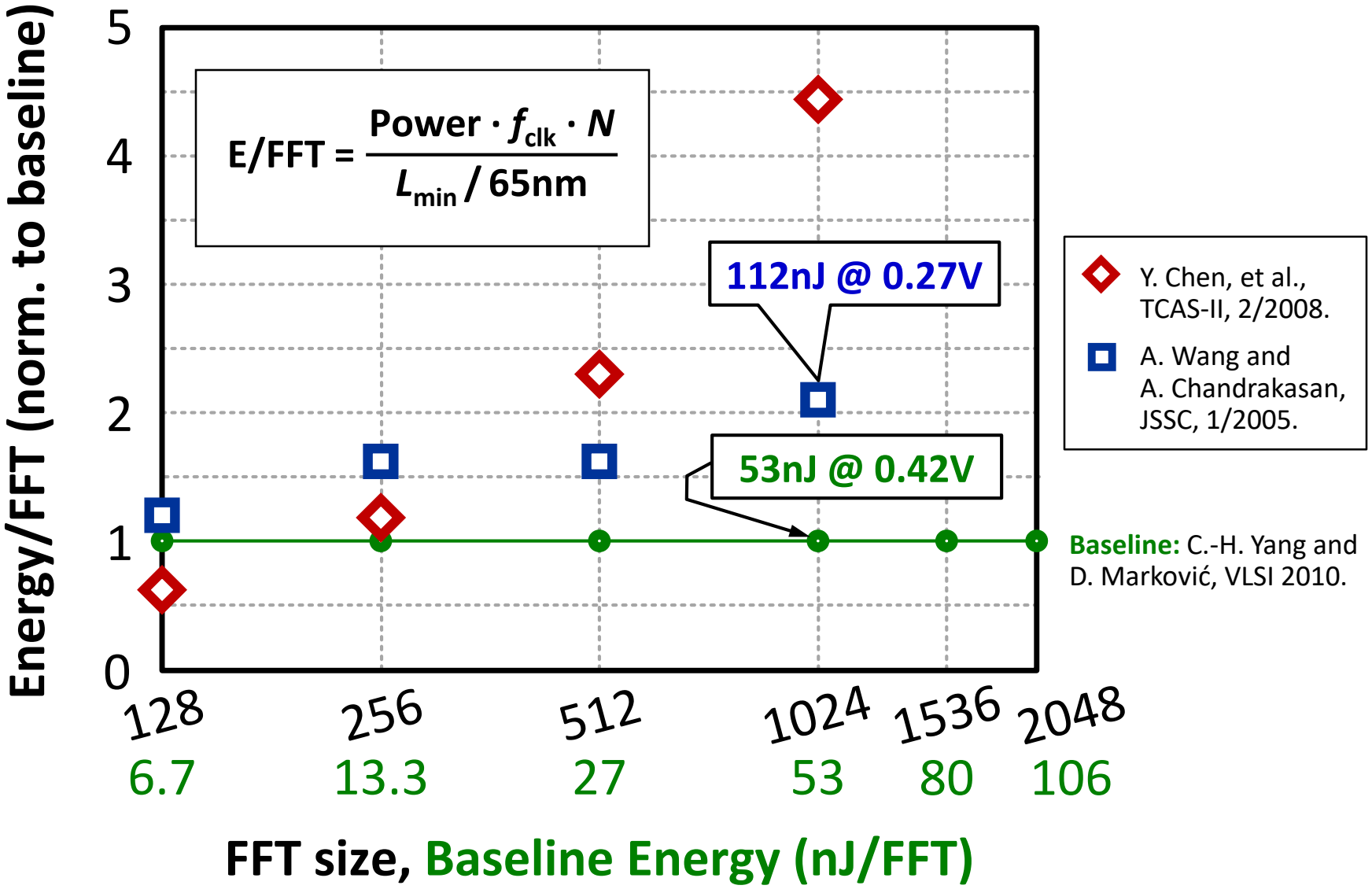
| BW (MHz) | FFT size | Optimal Factorization |
|---|---|---|
| 20 | 2048 | (16x16)x8 |
| 15 | **1536** | (16x16)x**6** |
| 10 | 1024 | (8x16)x8 |
| 5 | 512 | (8x8)x8 |
| 2.5 | 256 | (4x8)x8 |
| 1.25 | 128 | (16)x8 |

# Flexible FFT – Chip Layout




C.-H. Yang, et al., VLSI 2011.

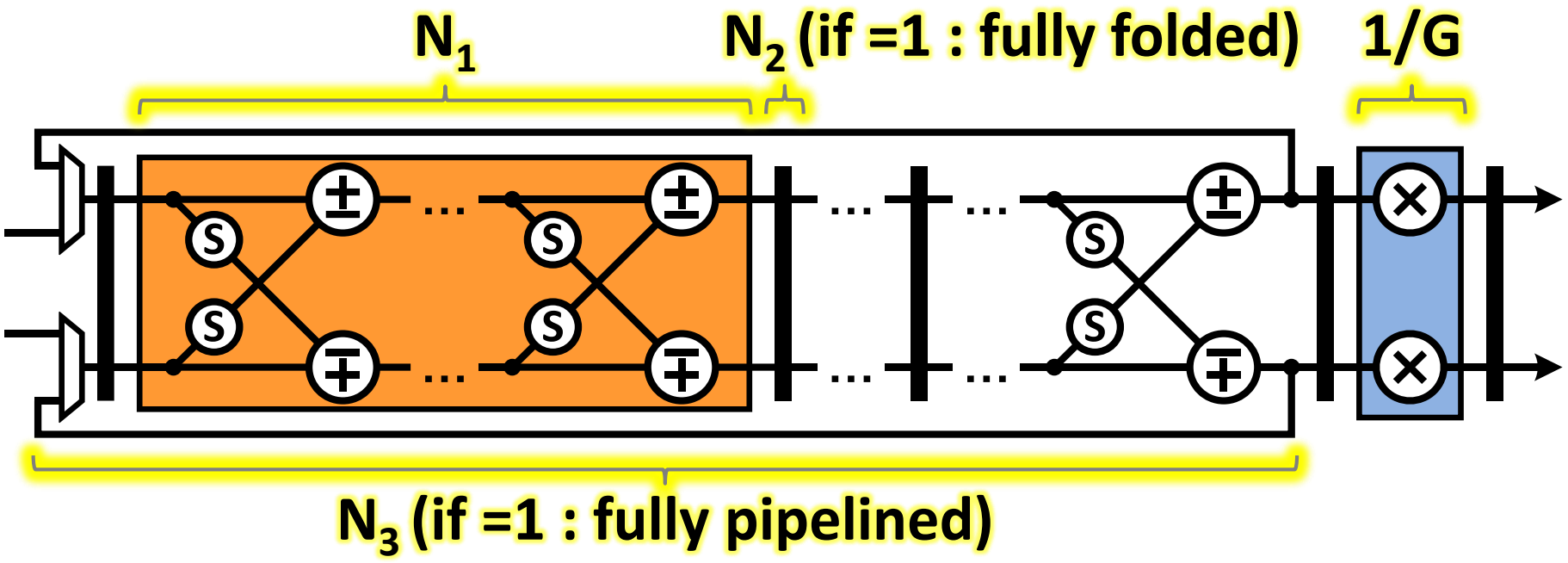
# Energy of Flexible FFTs



# CORDIC

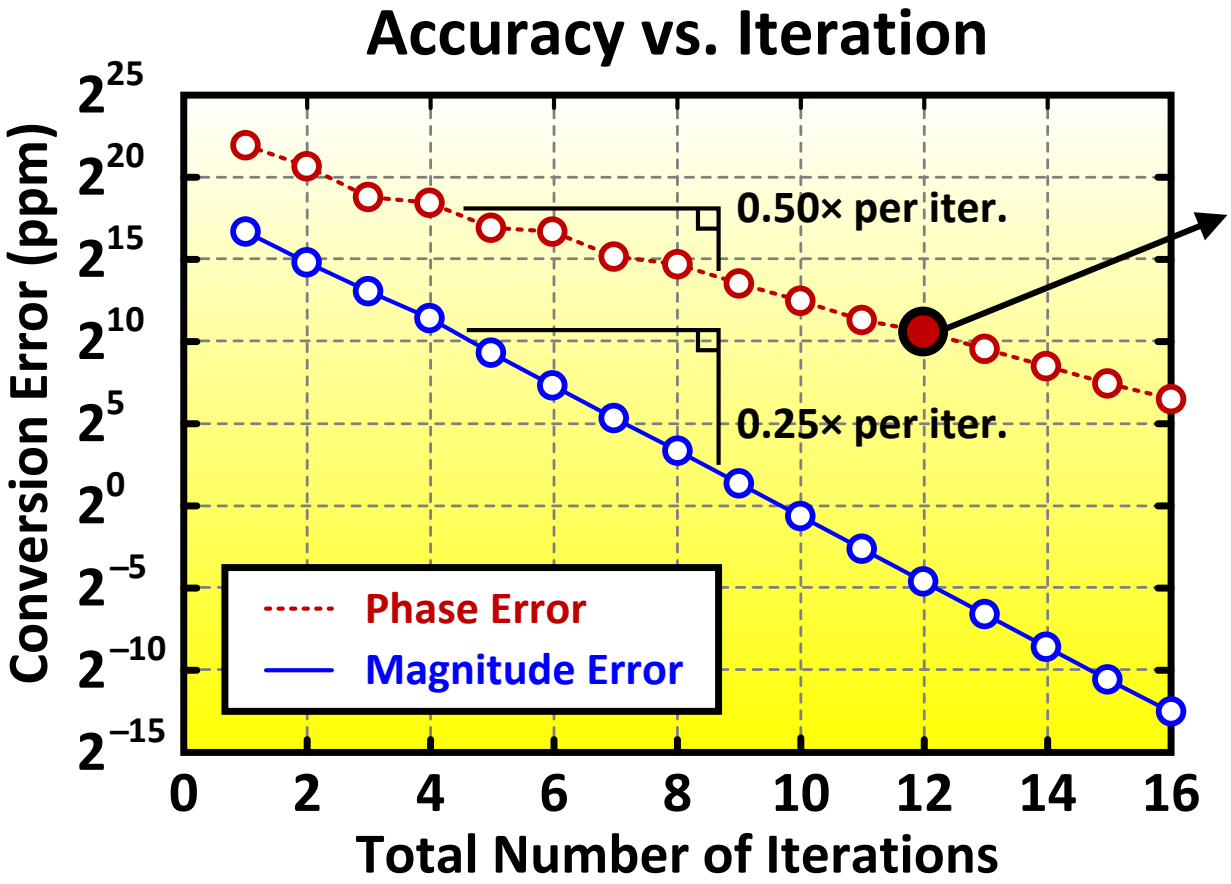
## Example

# Generalized CORDIC Architecture




| Design Variable | Notation | Depending on |
|---|---|---|
| Iteration per Pipeline Stage | $N_1$ | Operating Frequency |
| Number of Pipeline Stage | $N_2$ | Throughput, Area, Power |
| Folding Factor | $N_3$ | Throughput, Area, Power |
| Total Number of Iterations | $N$ ($=N_1N_2N_3$) | Accuracy |
| Gain-factor Normalization | 1/G | Accuracy & Application |

# System Requirements Affect Implementations




## Example

**Phase Err. < 0.2%**

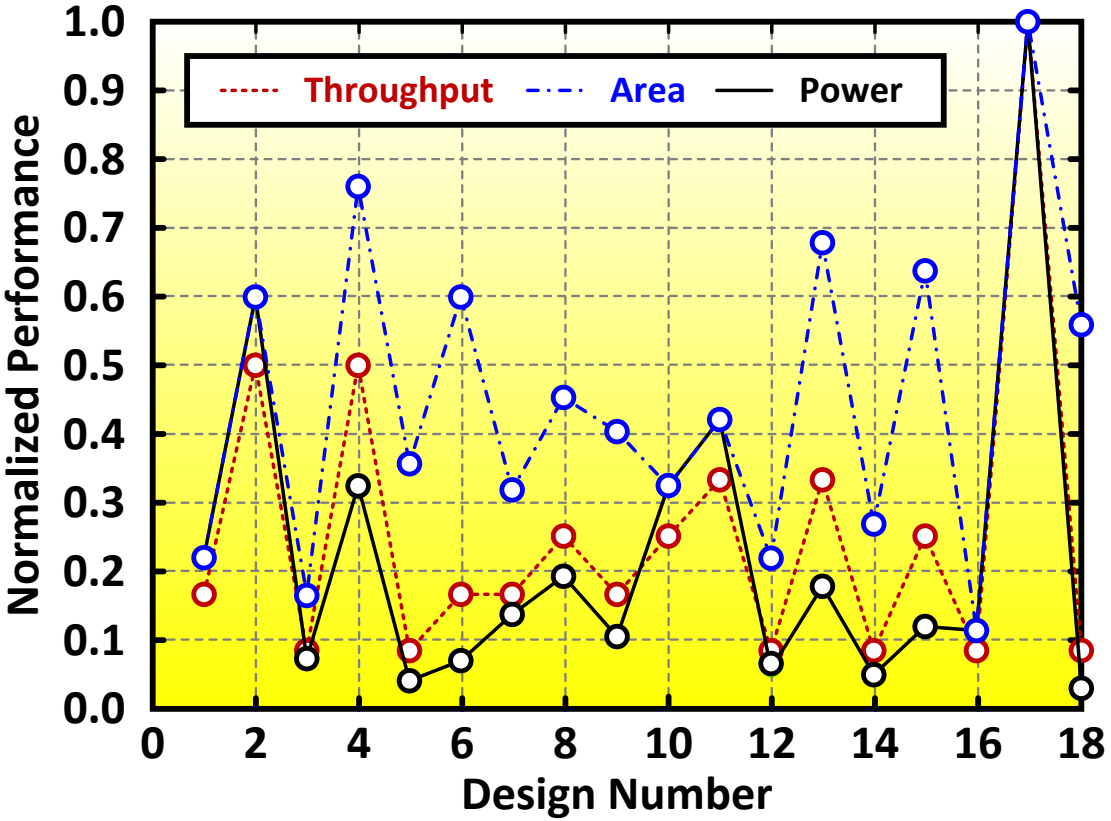
⇩

**12 CORDIC Iterations**

⇩

**18 Different Implementations**

# Design for Power, Area, or Throughput?




| Design | $N_1$ | $N_2$ | $N_3$ | Design | $N_1$ | $N_2$ | $N_3$ |
|---|---|---|---|---|---|---|---|
| D1 | 1 | 2 | 6 | D10 | 1 | 3 | 4 |
| D2 | 1 | 6 | 2 | D11 | 1 | 4 | 3 |
| D3 | 2 | 1 | 6 | D12 | 3 | 1 | 4 |
| D4 | 2 | 6 | 1 | D13 | 3 | 4 | 1 |
| D5 | 6 | 1 | 2 | D14 | 4 | 1 | 3 |
| D6 | 6 | 2 | 1 | D15 | 4 | 3 | 1 |
| D7 | 2 | 2 | 3 | D16 | 1 | 1 | 12 |
| D8 | 2 | 3 | 2 | D17 | 1 | 12 | 1 |
| D9 | 3 | 2 | 2 | D18 | 12 | 1 | 1 |

**Select suitable design based on requirements**

# Example 15.1a: Blind Parameter Estimation

- **Cognitive radio (CR): opportunistic spectrum sensing, classification and utilization**
- **Cyclic auto-correlation (CAC): universal way to extract signal features without prior knowledge (i.e. blindly)**
  - Modulation-type classification (PSK, QAM, MSK, AM…)
  - Symbol rate ($\mathbf{F_{sym}}$) and carrier frequency ($\mathbf{F_{car}}$) estimation

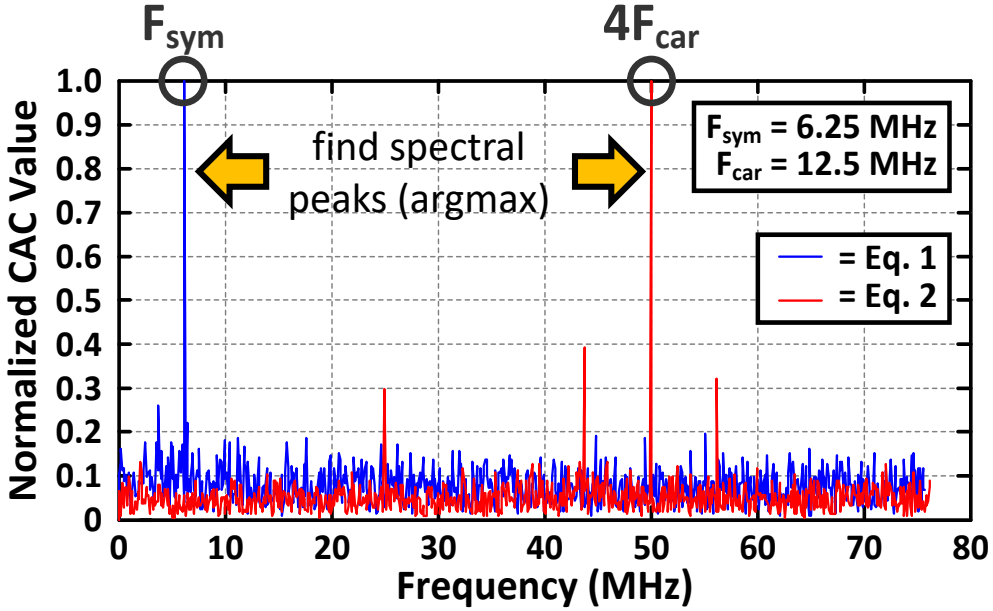
$$\hat{\mathbf{F}}_{\mathbf{sym}} = \arg\max_{\alpha \in W_T} \left| \sum_{n=0}^{N_T-1} |x[n]|^2 e^{-j2\pi\alpha T_{sam} n} \right|$$

**(Eq. 1)**

$$4\hat{\mathbf{F}}_{\mathbf{car}} = \arg\max_{\alpha \in W_F} \left| \sum_{n=0}^{N_F-1} x[n]^4 e^{-j2\pi\alpha T_{sam} n} \right|$$

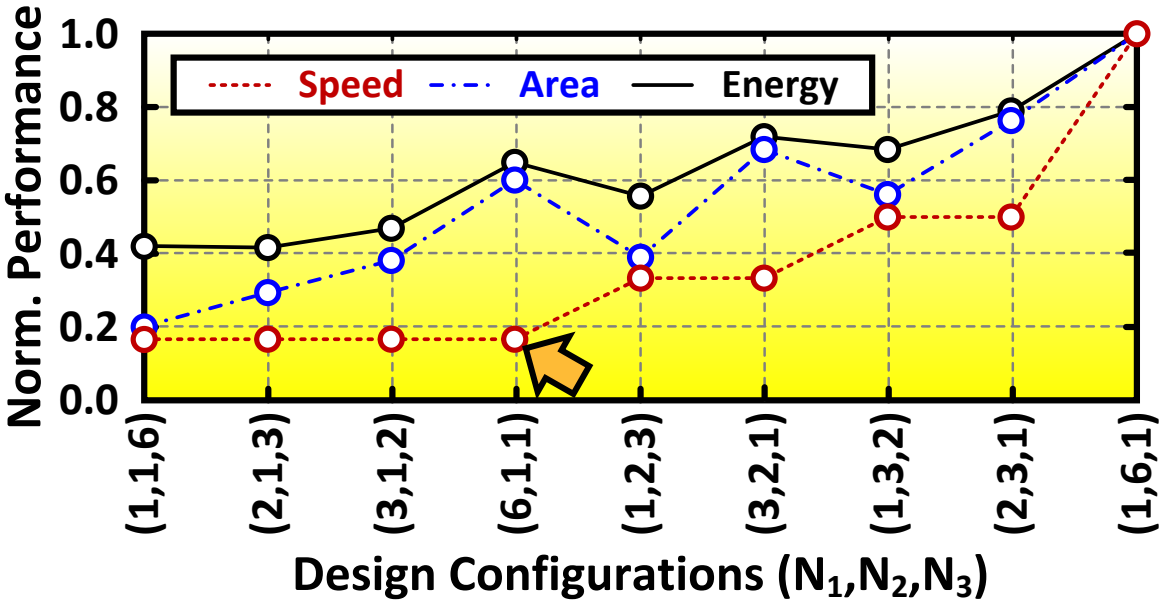
**(Eq. 2)**

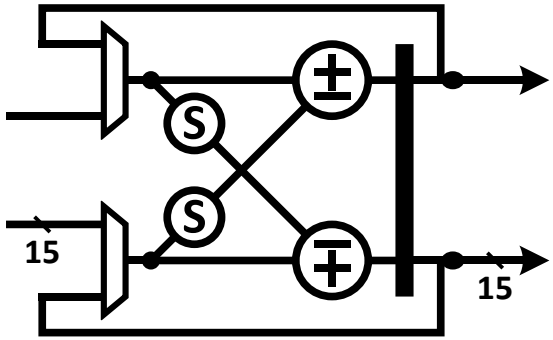
need **CORDIC** here



# Example 15.1b: Blind Parameter Estimation

- **Implementation considerations**
  - No gain-factor normalization: **argmax** doesn't care
  - Medium precision on magnitude: 6 iterations enough
  - Low throughput: frame sizes ($\mathbf{N_T}$ and $\mathbf{N_F}$) always >> iter. latency
  - Low leakage (area): only active every $\mathbf{N_T}$ and $\mathbf{N_F}$ clock cycles




**40nm Result**

**Area = 300µm²**

**$E_{dyn}$ + $E_{leak}$ = 2pJ**

# FFT Architecture: Takeaways

- **FFT hardware model has to consider the use of constant multipliers for twiddle factors**
  - Such model guides selection of radix
- **Radix factorization is essential for low energy & area**
  - Proper radix granularity from 2 to 16
- **Flexible processing element allows radix factorization**
  - It also supports flexible number of FFT points
- **DFF and RF should be carefully implemented**
  - DFF up to 256 points, RF for 512 and above