Finite-Word-Length-Effects in Practical Block-Floating-Point FFT

Gil Naveh
Tel-Aviv Research Center
Huawei Technologies Co. Ltd, Tel-Aviv, Israel
Email: gil.naveh@huawei.com

SIGNAL 2025

Gil (Nave) Naveh – Short Bio

Education

Gil Received his M.Sc. in EE 1992 from Ben-Gurion University, Israel.

Areas of expertise

Signal processing, Digital communications and Digital Signal Processor

Experience

Held positions of Chief Scientist in multiple Hi-tech companies

Now serving as the DSP CTO of Huawei, Israel

Publications

Hold few publications in signal processing, digital communications and biomedical signal processing

Why Finite-Word-Length-Effects in BFP-FFT **Again**?

- **SW implementation of FFT gain huge momentum with the dominance of OFDM modems and the giant wave of sensing applications in 5.5G and 6G**
- **Floating point processors are generally more expensive (silicon die and power consumption) and less efficient for classical signal processing applications**
- **High-Accuracy-High-Efficiency SW FFT on fixed-point processor requires Block-Bloating-Point (BFP)**
- **Finite-Word-Length-Effects of BFP-FFT for Radix-2 Cooley Tuckey FFT were deeply investigated back in 1969 by Clifford J. Weinstein (Bell Labs)**
- **That analysis covered an ideal version of BFP-FFT which is not suitable for many real-time embedded use-cases**

Why Finite-Word-Length-Effects in BFP-FFT **Again**?

- **In practice we all use a different derivative of BFP-FFT that is more suitable for real-time embedded use-cases (called herein practical BFP-FFT)**
- **This commonly used derivative has never been analyzed for accuracy (SQNR)**
- **Most of us (the engineers) believe that it's performance are very close to that of the ideal BFP-FFT, But**
  - **This is not the case !!**

HUAWEI

Why Finite-Word-Length-Effects in BFP-FFT **Again**?

- **In this work we:**
  - **Derive the mathematical model of the practical BFP-FFT, and**
    - Refine the model of the ideal BFP-FFT
    - Extend to radix-4
  - **Adapt the models of the ideal and practical BFP-FFT to the modern fixed-point processors (DSPs and CPUs)**
  - **Compare the performance of the practical BFP-FFT to that of the ideal and observe the performance loss**
  - **Compare the SQNR performance of Radix-2 to Radix-4 of the ideal and practical BFP-FFT**
  - **Analyze the effects of Twiddle-Factors of the set $\{1, -1, j, -j\}$**

Agenda

- **Why Block Floating Point (BFP) FFT**
- **Ideal (theoretical) BFP**
- **Practical BFP**

Introduction

- **Models: Processor, FFT and Quatization noise**

Models

- **Accuracy (finite word length effects) model of generic BFP-FFT policy**
- **Derivation of the SQNR of Ideal BFP-FFT**
- **Derivation of the SQNR Practical BFP-FFT**

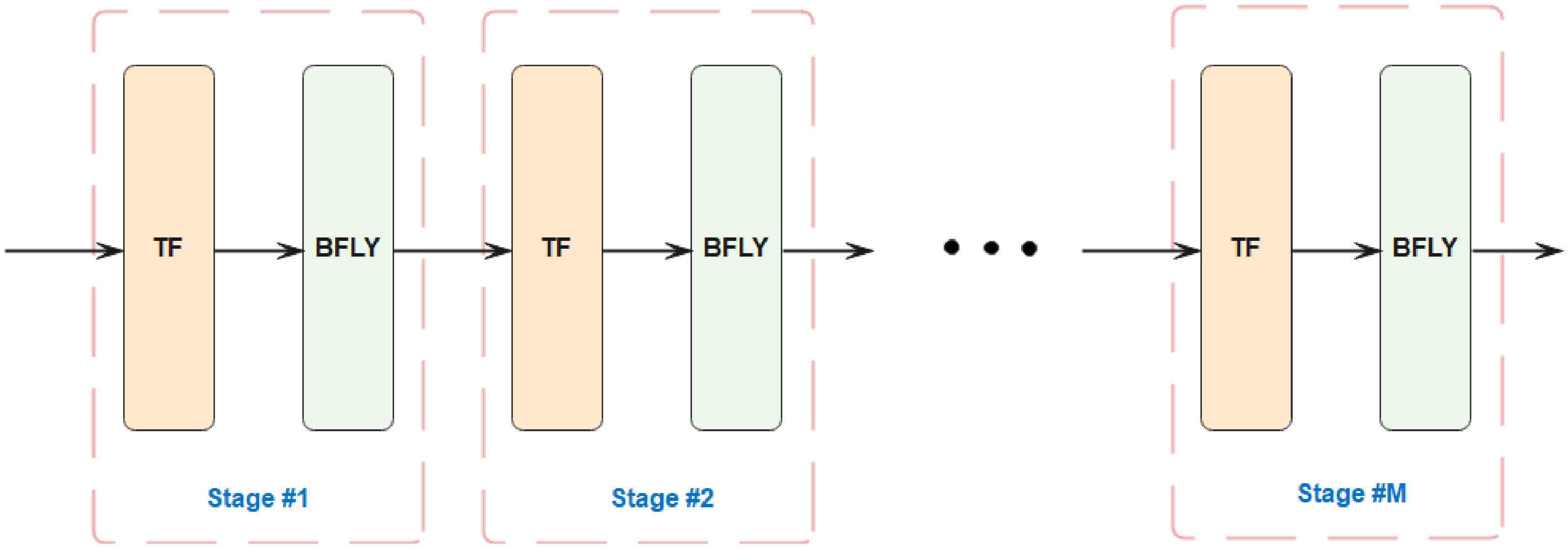
Derivation

- **Summary & Results**

Results

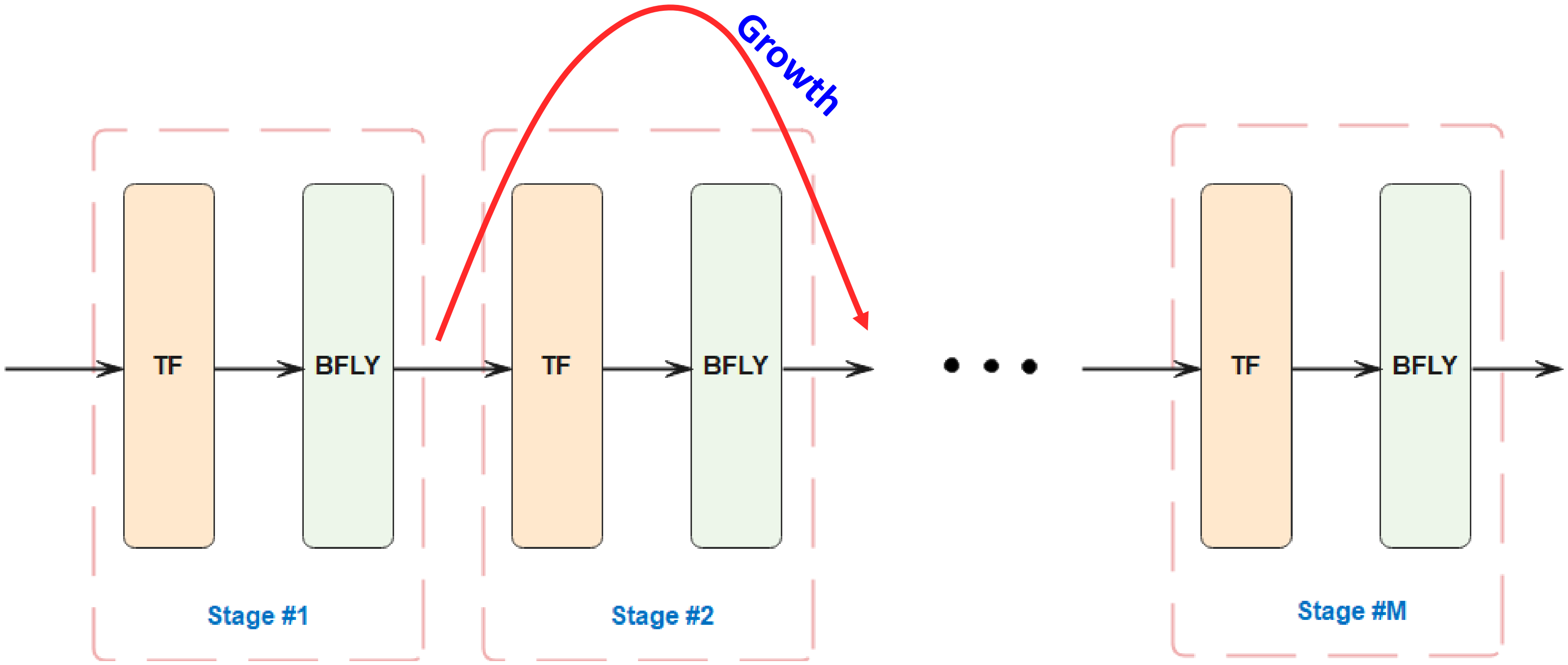
General Cooley-Tuckey FFT structure

- **The classical Cooley-Tuckey FFT is composed of Stages, Each stage is composed of**
  - Twiddle Factors (TF), marked as $w_N^{kn} = e^{-j\frac{2\pi kn}{N}}$ that rotate the complex values by a known angle of $\frac{2\pi kn}{N}$
  - A Butterfly (BFLY) that outputs a weighted sums of the inputs



General Cooley-Tuckey FFT structure

- **The maximal output (magnitude) of each stage is larger or equal to the maximal output of the previous stage**

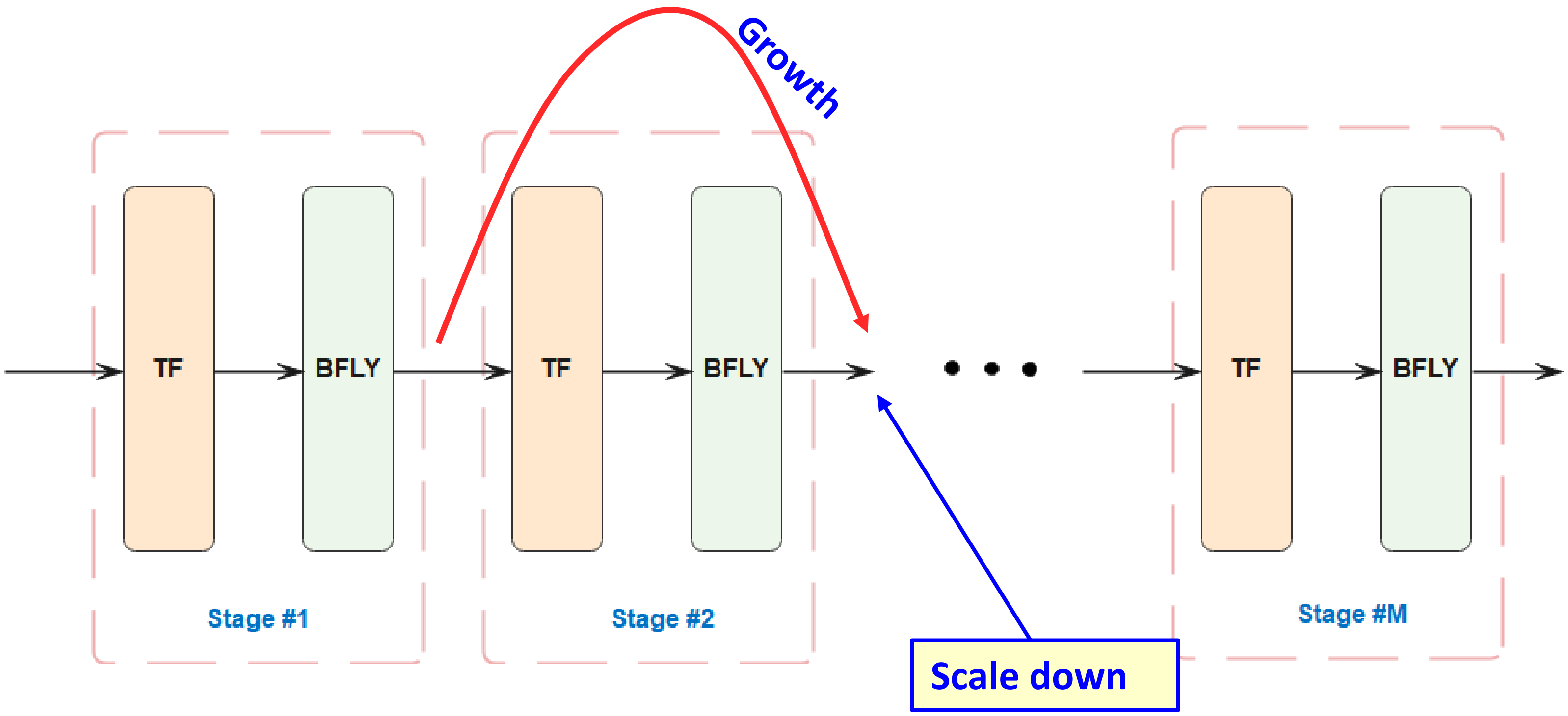


➔ **The word-length at the output of a stage is larger than that at the input**

➔ **Not suitable for SW implementation**

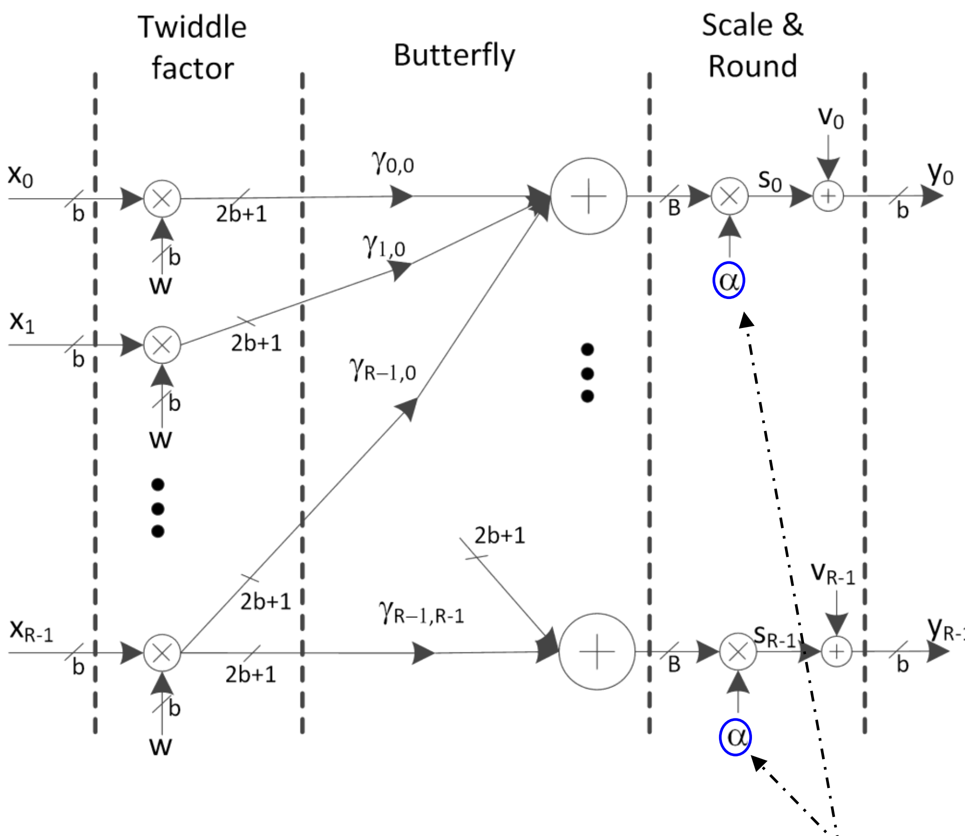
General Cooley-Tuckey FFT structure

- **The common approach in SW implementations is to scale down the stage outputs before being stored to memory**
- **Scheme known as Block-Floating-Point-FFT (BFP-FFT)**
  - Sometimes also called “dynamic scale” or “dynamic shift”



General Radix-R DIT Butterfly

- **Three sections:**
  - **Twiddle Factors**
  - **Butterfly**
  - **Scale & Round**



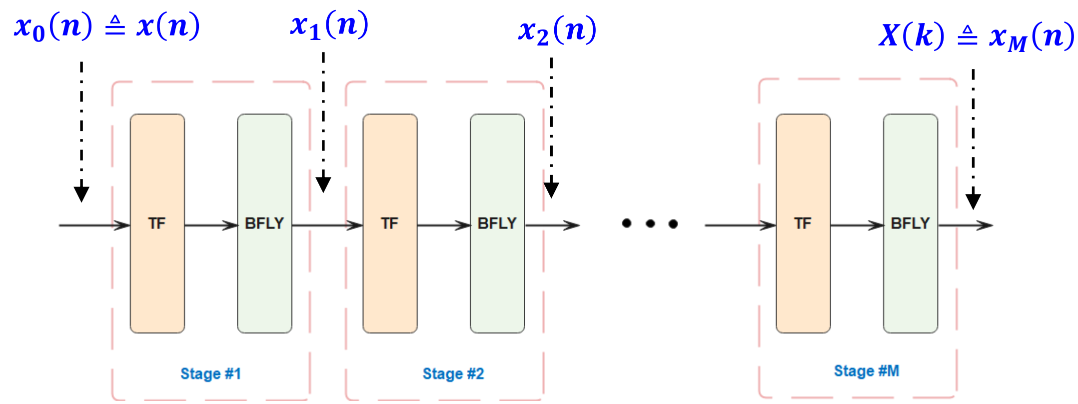
- **The output of the butterflies are scaled down (by right shift, $\alpha = 2^{-q}$) and rounded before stored to memory**

Scaling at the BFP-FFT

$$X(k) = FFT\{x(n)\} \triangleq \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}$$

$$x_0(n) \triangleq x(n)$$

$$X(k) \triangleq x_M(n)$$



In the ideal BFP-FFT a scale down at stage m will happen iff one of the stage's outputs is larger than 1 (re-calculation is required)

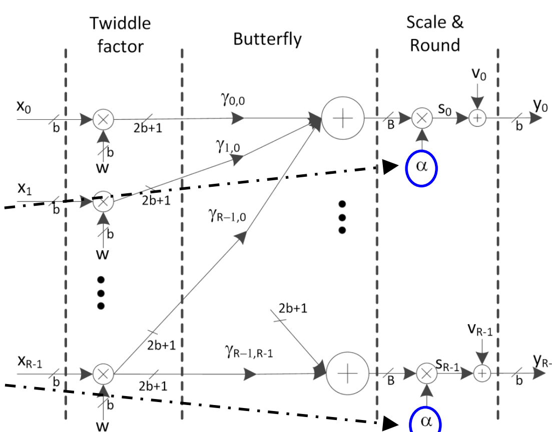
$$\tilde{s}_m = \max_n(max\{|Re[x_m(n)]|, |Im[x_m(n)]|\})$$

$$q_m = \max\{0, \lceil \log_2 \tilde{s}_m \rceil\}$$

The down scale value is

$$\alpha_m = 2^{-q_m}$$

Down scale take place only if $\tilde{s}_m > 1$



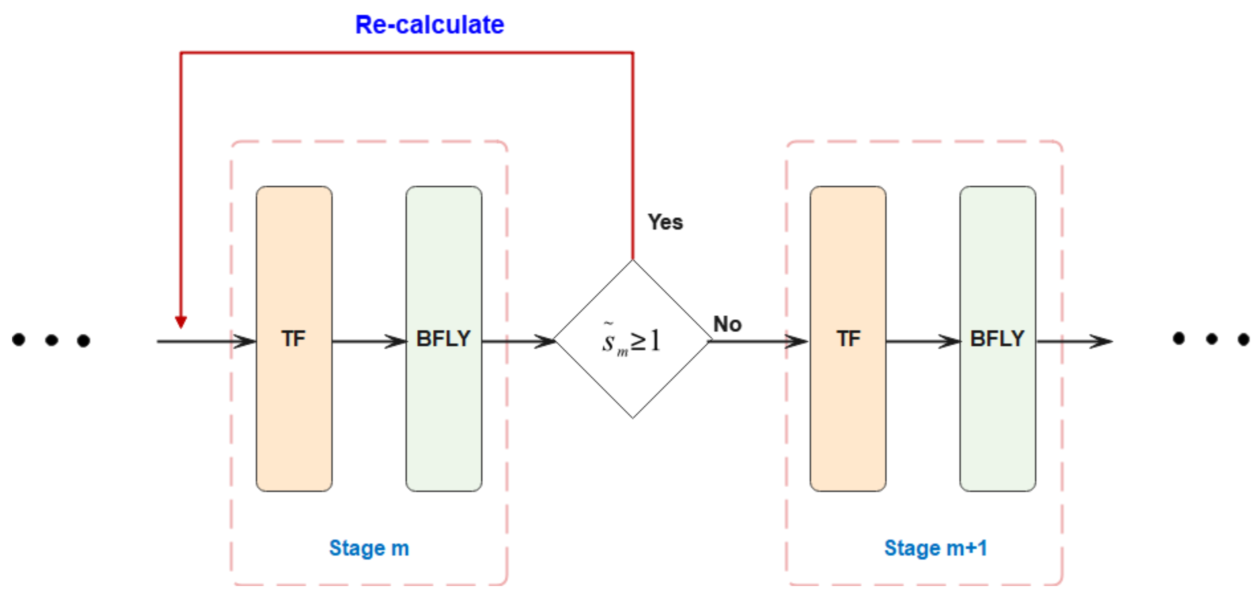
HUAWEI

Main Drawback of the Ideal BFP-FFT

The scale decision (whether to scale and by what factor) of stage m is a function of the outputs of that stage:

Scale down of stage m take place if and only if one of the outputs of that stage overflows

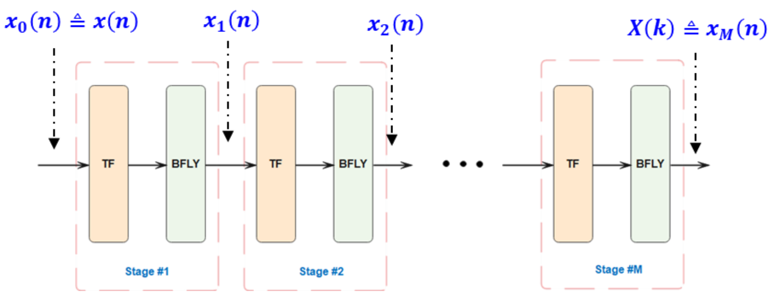
- **Recalculation of the stage is required if overflow is detected**
- **The entire FFT calculation latency is non-deterministic**



Not suitable for strict real-time, pipelined systems like demodulation of OFDM modem

Scaling at the Practical BFP-FFT

To eliminate the non-deterministically, Shively(*) proposed a scheme at which the scaling at stage m depend on the outputs of stage $m-1$



The maximal growth is $\sqrt{2}R$:

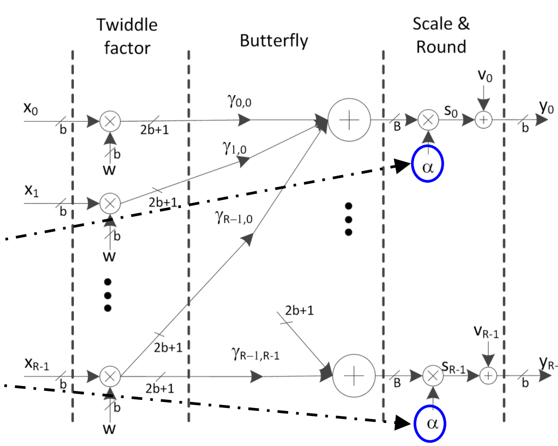
- Adding R values → growth by $\leq R$
- Rotation by $e^{\frac{-j2\pi kn}{N}}$ → growth by $\leq \sqrt{2}$

The scaling criteria is:

$$\tilde{x}_{m-1} = \max_n\left(max\{|Re[x_{m-1}(n)]|, |Im[x_{m-1}(n)]|\}\right)$$

$$q_m = \max\{0, \lceil \log_2 \sqrt{2}R\tilde{x}_{m-1} \rceil\}$$

$$\alpha_m = 2^{-q_m} = 2^{-\max\{0, \lceil \log_2 \sqrt{2}R\tilde{x}_{m-1} \rceil\}}$$

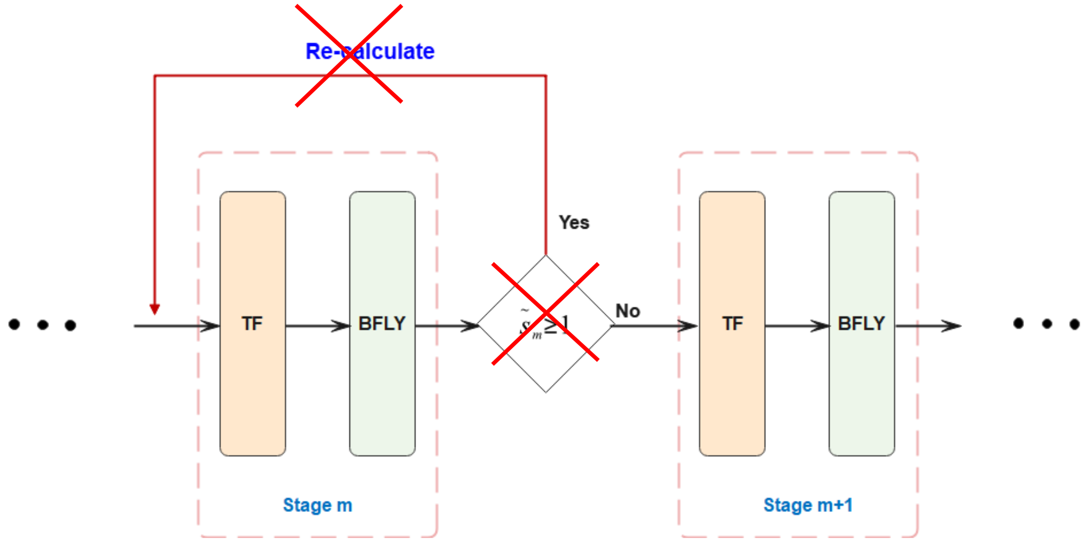


(*) R. R. Shively, "A Digital Processor to Generate Spectra in Real Time," IEEE Transactions on Computers, Vols. C-17, no. 5, pp. 485-491, 1968.

Scaling at the Practical BFP-FFT

Shively's scheme guarantee the following:

- **Deterministic latency**
- **No overflows will occur**



The cost – degraded accuracy (SQNR) → The main goal of this work

Ideal vs Practical BFP-FFT

- **The scaling mechanism of both schemes is identical:**

  Scale the whole stage by right shift and round

- **Both schemes guarantee that no overflows will occur**
- **The difference is the scaling policy – when to scale and by what factor**

---

practical

$$\alpha_m = 2^{-q_m} \quad where$$

$$\boxed{q_m = \max\{0, \lceil \log_2 \sqrt{2} R \tilde{x}_{m-1} \rceil\}}$$

and

$$\tilde{x}_{m-1} = \max_n(max\{|Re[x_{m-1}(n)]|, |Im[x_{m-1}(n)]|\})$$

ideal

$$\alpha_m = 2^{-q_m} \quad where$$

$$\boxed{q_m = \max\{0, \lceil \log_2 \tilde{s}_m \rceil\}}$$

$$\tilde{s}_m = \max_n(max\{|Re[x_m(n)]|, |Im[x_m(n)]|\})$$

Contribution of this work

1. **Derivation the analytical SQNR formulas of the practical BFP – main contribution**
   - Serves as a design tool for FFT implementations on CPUs and DSPs
   - Serves as a design tool for DSP (and CPU) processor architectures
2. **Refine the classical noise model for BFP-FFTs**
   - Calculation the contribution of all possible the scale patterns
   - Incorporate the effects of Twiddle-Factors from the set $\{1, -1, j, -j\}$
3. **Assess the performance loss of the practical BFP-FFT compared to the ideal BFP-FFT**
4. **Compare the SQNR of BFP-FFT of Radix-2 to Radix-4 in the ideal and the practical BFP-FFT**

Reference Models

Processor model

Have an **embedded complex multiplier**

Processor register of b bits

Accumulator registers of $B \geq 2b + \lceil \log_2 R \rceil + 1$ bits,
(where R is the FFT Radix)

Arithmetic is 2's complement

Data written to memory is always of b bits

Quantization noise model

Rely on the **Rounding-Half-Up** (RHU)

$$\boldsymbol{y} = \boldsymbol{Q}[\boldsymbol{s}] \triangleq \mathbf{2}^{-\boldsymbol{b}} \cdot \lfloor \boldsymbol{s} \cdot \mathbf{2}^{\boldsymbol{b}} + \mathbf{0.5} \rfloor$$

Quantization noise

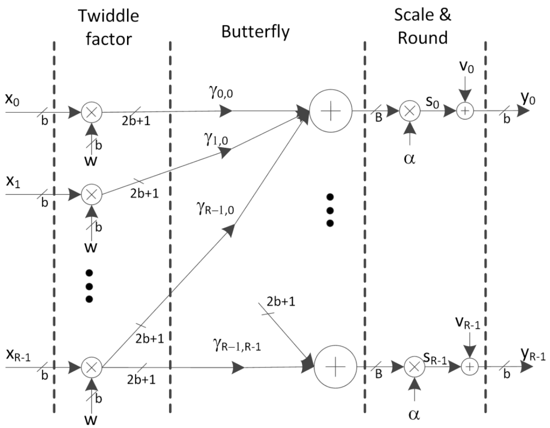
$$v = s - y = s - Q[s]$$

$$v \sim U[-2^{-b}, 2^{-b})$$

$$\sigma_v^2 = \frac{2^{-2(b-1)}}{12}$$

FFT model

Fixed-Radix DIT Cooley-Tuckey FFT



Terminology

scaling pattern $\boldsymbol{q} = [q_1, q_2, \ldots, q_M]$
(The pattern depends on the scaling policy)

Probability of a scaling pattern - $Pr(\boldsymbol{q}; \sigma_{x_0}^2)$
(Depends on the input signal power $\sigma_{x_0}^2$)

$\mathrm{SQNR}(\boldsymbol{q}; \sigma_{x_0}^2)$ - $\sigma_{x_M}^2(\boldsymbol{q}; \sigma_{x_0}^2)/\rho_E^2(\boldsymbol{q})$
(Depends on the input signal power $\sigma_{x_0}^2$
and the scaling pattern)

Averaged SQNR Calculation

The SQNR calculation is composed of 3 steps:

a) Calculating the SQNR for a specific scaling pattern, $\boldsymbol{q} = [q_1, q_2, \dots, q_M]$, as

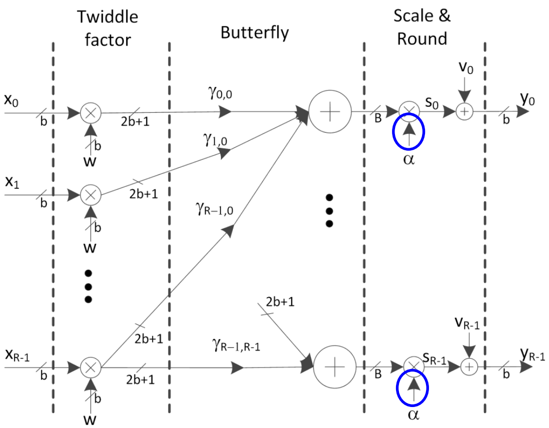
$$\mathrm{SQNR}\left(\boldsymbol{q}; \sigma_{x_0}^2\right) = \frac{\sigma_{x_M}^2\left(\boldsymbol{q}; \sigma_{x_0}^2\right)}{\rho_E^2(\boldsymbol{q})}$$

b) Calculating the probability of each scaling pattern $Pr\left(\boldsymbol{q}; \sigma_{x_0}^2\right)$

c) Calculating the averaged (expected value) SQNR

$$SQNR = \sum_{\boldsymbol{q}} Pr\left(\boldsymbol{q}; \sigma_{x_0}^2\right) \cdot SNR\left(\boldsymbol{q}, \sigma_{x_0}^2\right) = \sum_{\boldsymbol{q}} Pr(\boldsymbol{q}; \sigma_{x_0}^2) \cdot \frac{\sigma_{x_M}^2(\boldsymbol{q}, \sigma_{x_0}^2)}{\rho_E^2(\boldsymbol{q})}$$

SQNR Calculation – Output Signal Power



The variance (power) of the FFT's output is

$$\sigma_{x_M}^2 = \underbrace{N\sigma_{x_0}^2}\underbrace{\prod_{m=1}^{M}\alpha_m^2} = N\sigma_{x_0}^2\, 2^{-2\sum_{m=1}^{M} q_m}$$

The output variance if no scaling were done.
From the FFT definition

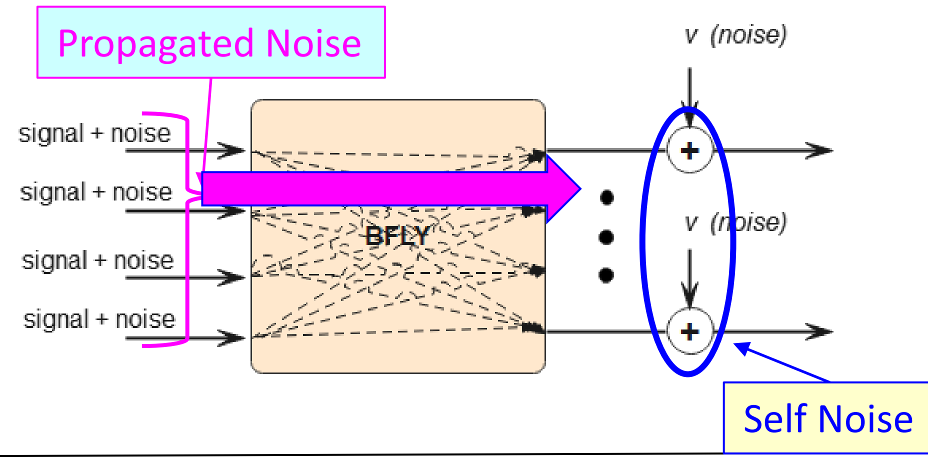
$$x_M(k) \triangleq \sum_{n=0}^{N-1} x_0(n) e^{\frac{-j2\pi kn}{N}}$$

The attenuation of the scale pattern

SQNR Calculation – Output Noise Power

The noise at the output of a butterfly is composed of **two components**:

- *The **Self-Noise** – noise that is generated at the particular butterfly*
- *The Propagated-Noise – noise that was generated in earlier stages and propagated through the particular butterfly*



Noise generated at 1st stage propagates to the output through $M-1$ following stages
Results in accumulation of R^{M-1} noise sources
Each attenuated by a factor of $\prod_{m=2}^{M} \alpha_m^2$

$$\sigma_v^2 R^{M-1} \prod_{i=2}^{M} \alpha_i^2$$

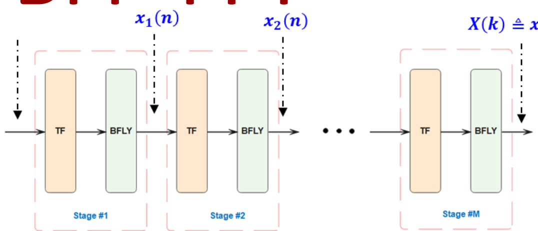
Noise generated at 2nd stage propagates to the output through $M-2$ following stages
Results in accumulation of R^{M-2} noise sources
Each attenuated by a factor of $\prod_{m=3}^{M} \alpha_m^2$

$$\sigma_v^2 R^{M-2} \prod_{i=3}^{M} \alpha_i^2$$

The variance (power) of the Quatization noise at the output

$$\rho_E^2 = \sigma_v^2 \left(1 + \sum_{m=1}^{M-1} R^{M-m} \prod_{i=m+1}^{M} \alpha_i^2 \right) = \sigma_v^2 \left(1 + \sum_{m=1}^{M-1} \prod_{i=m+1}^{M} R\alpha_i^2 \right)$$

Scale pattern probability : Practical BFP-FFT



The probability of a scale pattern $\boldsymbol{q} = [q_1, q_2, \dots, q_M]$ by the chain rule is

$$\boxed{\begin{aligned} Pr(\boldsymbol{q}; \sigma_{x_0}^2) &= Pr(q_1; \sigma_{x_0}^2) Pr(q_2 | q_1; \sigma_{x_0}^2) Pr(q_3 | q_1, q_2; \sigma_{x_0}^2) \dots Pr(q_M | q_1, q_2, \dots, q_{M-1}; \sigma_{x_0}^2) \\ &= Pr(q_1; \sigma_{x_0}^2) \prod_{m=2}^{M} Pr(q_m | q_1, q_2, \dots, q_{m-1}; \sigma_{x_0}^2) \end{aligned}}$$

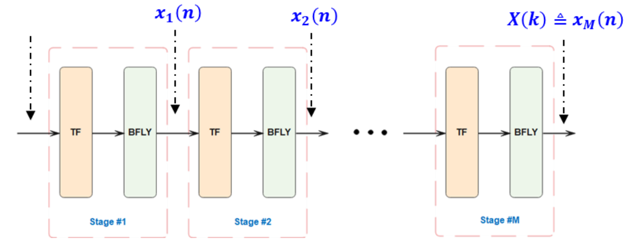
The probability of q_m given all previous scaling $q_1, q_2, \dots, q_{m-1}$ for an input signal variance $\sigma_{x_0}^2$ can be written as

$$Pr(q_m | q_1, q_2, \dots, q_{m-1}; \sigma_{x_0}^2) = \boxed{Pr(q_m; \sigma_{x_{m-1}}^2)}$$

where

$$\sigma_{x_{m-1}}^2 = \sigma_{x_0}^2 \cdot R^{m-1} \cdot 2^{-2 \sum_{k=1}^{m-1} q_k} = \sigma_{x_0}^2 \cdot R^{m-1} \cdot \prod_{i=1}^{m-1} \alpha_i^2$$

Scale pattern probability : Practical BFP-FFT



The practical BFP-FFT scale policy

$$q_m = max\{0, \log_2[\sqrt{2}R\tilde{x}_{m-1}]\}$$

$$\tilde{x}_{m-1} = \max_n(max\{|Re[x_{m-1}(n)]|\,, |Im[x_{m-1}(n)]|\})$$

Defining

$$x_m^c(2n) = real\big(x_m(n)\big)$$
$$x_m^c(2n+1) = Imag\big(x_m(n)\big)$$

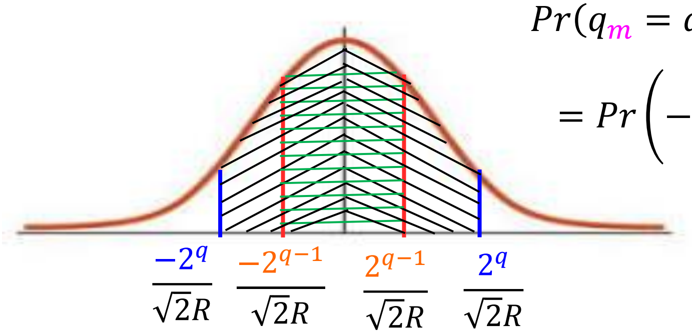
The probability that there will be exactly $q = 0$ right shifts at stage m is equal to

$$Pr(q_m = 0) = Pr\big(\sqrt{2}R\tilde{x}_{m-1} \le 1\big) = Pr\left(\tilde{x}_{m-1} \le \frac{1}{\sqrt{2}R}\right)$$

Whereas for $q > 0$

$$Pr(q_m = q) = Pr\big(2^{q-1} \le \sqrt{2}R\tilde{x}_{m-1} \le 2^q\big) = Pr\left(\frac{2^{q-1}}{\sqrt{2}R} \le \tilde{x}_{m-1} \le \frac{2^q}{\sqrt{2}R}\right)$$

$$= Pr\left(-\frac{2^q}{\sqrt{2}R} \le \underset{n}{all}\{x_{m-1}^c(n)\} \le \frac{2^q}{\sqrt{2}R}\right) - Pr\left(-\frac{2^{q-1}}{\sqrt{2}R} \le \underset{n}{all}\{x_{m-1}^c(n)\} \le \frac{2^{q-1}}{\sqrt{2}R}\right)$$

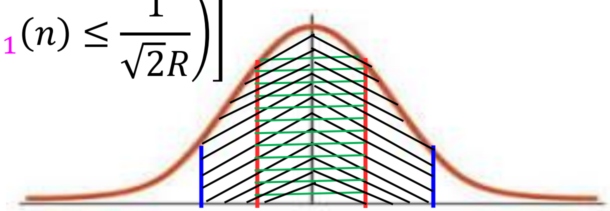


Scale pattern probability : Practical BFP-FFT

By the assumption that the input, $\boxed{x_0(n)\text{, is i.i.d.}}$ → the arrays $x_m(n)$ are i.i.d. in n $\forall m$
Therefore,

The probability $Pr(q_m = 0)$ becomes

$$Pr(q_m = 0) = Pr\left(-\frac{1}{\sqrt{2}R} \le \underset{n}{\boldsymbol{all}}\{x^c_{m-1}(n)\} \le \frac{1}{\sqrt{2}R}\right) = \left[Pr\left(-\frac{1}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{1}{\sqrt{2}R}\right)\right]^{2N}$$



Similarly $Pr(q_m = q)$ reads

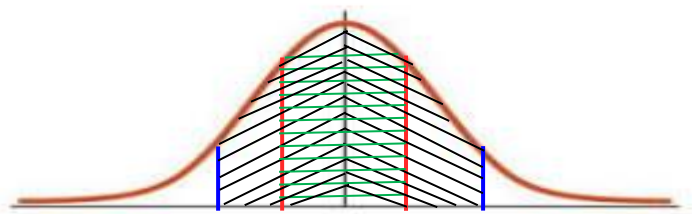
$$Pr(q_m = q) = Pr\left(-\frac{2^q}{\sqrt{2}R} \le \underset{n}{\boldsymbol{all}}\{x^c_{m-1}(n)\} \le \frac{2^q}{\sqrt{2}R}\right) - Pr\left(-\frac{2^{q-1}}{\sqrt{2}R} \le \underset{n}{\boldsymbol{all}}\{x^c_{m-1}(n)\} \le \frac{2^{q-1}}{\sqrt{2}R}\right)$$

$$= \boxed{\left[Pr\left(-\frac{2^q}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{2^q}{\sqrt{2}R}\right)\right]^{2N} - \left[Pr\left(-\frac{2^{q-1}}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{2^{q-1}}{\sqrt{2}R}\right)\right]^{2N}}$$

Practical BFP-FFT scale pattern prob for Gaussian inputs

For Gaussian distributions the probability $Pr\left(-\frac{2^q}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{2^q}{\sqrt{2}R}\right)$ is

$$Pr\left(-\frac{2^q}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{2^q}{\sqrt{2}R}; \sigma^2_{\mathrm{x}_{m-1}}\right) = \boldsymbol{erf}\left(\frac{2^q}{\sqrt{2}R\sigma_{\mathrm{x}_{m-1}}}\right)$$



Plugging

$$\sigma^2_{x_{m-1}} = \sigma^2_{x_0} R^{m-1} T_{m-1} \quad ; \quad T_m = 2^{-2\sum_{i=1}^m q_i} = \prod_{i=1}^m \alpha_i^2$$

We have

$$Pr\left(-\frac{2^q}{\sqrt{2}R} \le x^c_{m-1}(n) \le \frac{2^q}{\sqrt{2}R} | q_1, q_2, \dots, q_{m-1}; \sigma^2_{\mathrm{x}_0}\right) = erf\left(\frac{2^q}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)$$

such that

$$Pr(q_m = q | q_1, q_2, \dots, q_{m-1}; \sigma^2_{\mathrm{x}_0}) = \left[erf\left(\frac{2^q}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N} - \left[erf\left(\frac{2^{q-1}}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N} \quad ; \quad q > 0$$

$$Pr(q_m = 0 | q_1, q_2, \dots, q_{m-1}; \sigma^2_{\mathrm{x}_0}) = \left[erf\left(\frac{1}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N}$$

Practical BFP-FFT SQNR

Finally, the averaged SQNR is the sum of the SQNR of all the $\boldsymbol{q}$ sequences weighted by their appearance probabilities

$$\boldsymbol{SQNR}(\sigma_{x_0}^2) = \sum_{\boldsymbol{q}} Pr(\boldsymbol{q};\sigma_{x_0}^2)\cdot SQNR(\boldsymbol{q},\sigma_{x_0}^2) = \sum_{\boldsymbol{q}} Pr(\boldsymbol{q};\sigma_{x_0}^2)\cdot \frac{\sigma_{x_M}^2(\boldsymbol{q},\sigma_{x_0}^2)}{\rho_E^2(\boldsymbol{q})}$$

$$\boxed{= \sum_{\boldsymbol{q}} \frac{\sigma_{x_M}^2(\boldsymbol{q},\sigma_{x_0}^2)}{\rho_E^2(\boldsymbol{q})}\cdot Pr(q_1;\sigma_{x_0}^2)\prod_{m=2}^{M} Pr(q_m|q_1,q_2,\dots,q_{m-1};\sigma_{x_0}^2)}$$

$$\sigma_{x_M}^2(\boldsymbol{q};\sigma_{x_0}^2) = N\sigma_{x_0}^2\prod_{m=1}^{M}\alpha_m^2 = N\sigma_{x_0}^2 2^{-2\sum_{m=1}^{M} q_m}$$

$$\rho_E^2 = \sigma_v^2\left(1+\sum_{m=1}^{M-1}\prod_{i=m+1}^{M} R\alpha_i^2\right) = \sigma_v^2\left(1+\sum_{m=1}^{M-1} R^{M-m}\prod_{i=m+1}^{M}\alpha_i^2\right)$$

$$Pr(q_m|Q_{m-1};\sigma_{x_0}^2) = \left[erf\left(\frac{2^{q_m}}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N} - \left[erf\left(\frac{2^{q_m-1}}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N} \quad ; \quad q_m > 0$$

$$Pr(q_m = 0|Q_{m-1};\sigma_{x_0}^2) = \left[erf\left(\frac{1}{\sigma_{x_0}\sqrt{2R^{m+1}T_{m-1}}}\right)\right]^{2N}$$

$$Pr(\boldsymbol{q};\sigma_{x_0}^2) = Pr(q_1;\sigma_{x_0}^2)\prod_{m=2}^{M} Pr(q_m|Q_{m-1};\sigma_{x_0}^2)$$

Ideal BFP-FFT SQNR

The formulas for $\sigma^2_{x_M}\left(\boldsymbol{q};\sigma^2_{x_0}\right)$ and $\rho^2_E(\boldsymbol{q})$ are identical to those of the practical BFP-FFT

The only difference is the **scaling pattern probabilities**

→A result of the different scaling policy

Repeating the same derivation steps as for the practical BFP-FFT we get the scaling probabilities:

$$Pr\left(q_m = q | q_1, q_2, \dots, q_{m-1}; \sigma^2_{x_0}\right) = \left[erf\left(\frac{2^{q_m}}{\sigma_{x_0}\sqrt{2R^m T_{m-1}}}\right)\right]^{2N} - \left[erf\left(\frac{2^{q_m-1}}{\sigma_{x_0}\sqrt{2R^m T_{m-1}}}\right)\right]^{2N} \quad ; \quad q > 0$$

$$Pr\left(q_m = 0 | q_1, q_2, \dots, q_{m-1}; \sigma^2_{x_0}\right) = \left[erf\left(\frac{1}{\sigma_{x_0}\sqrt{2R^m T_{m-1}}}\right)\right]^{2N}$$

The difference to the practical BFP-FFT is the R^m instead of R^{m+1} in the denominator

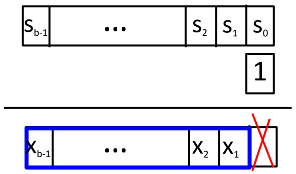
Twiddle-Factors of the set $\{1, -1, j, -j\}$

Multiplication of a b-bits number by TF of the set $\mathcal{T}_1 \triangleq \{1, -1, j, -j\}$ results is b-bits number

Rounding after scale down by very few bits results in a non-zero-mean discrete random variable

For example: shift by one bit and round

$$x = s \gg 1$$
$$y = Q[x] = 2^{-b} \cdot \lfloor x \cdot 2^b + 0.5 \rfloor = (s \gg 1) + \varepsilon_1$$



where

$$\varepsilon_1 = \begin{cases} 0 & w.p.\, 0.5 \\ -\frac{1}{2} 2^{-(b-1)} & w.p.\, 0.5 \end{cases},$$

It’s mean is

$$E[\varepsilon_1] = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \left(-\frac{1}{2} 2^{-(b-1)} \right) = -\frac{1}{4} 2^{-(b-1)} \neq 0$$

and power

$$\rho_{\varepsilon_1}^2 = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \left(\frac{1}{2} 2^{-(b-1)} \right)^2 = \frac{2^{-2(b-1)}}{8} > \frac{2^{-2(b-1)}}{12}$$

Twiddle-Factors of the set $\{1, -1, j, -j\}$

→The Multiplication of a b-bits number by TF of the set $\mathcal{T}_1$ results in higher noise power if scale down of the output takes place !

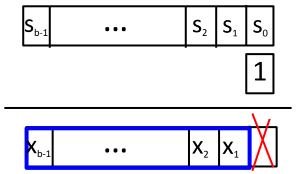
The noise power of scaled down samples multiplied by TF of the set $\mathcal{T}_1$ is

$$\rho_{\varepsilon_q}^2 = \begin{cases} 0 & ; \quad q = 0 \\ \frac{1}{8} 2^{-2(b-1)} & ; \quad q = 1 \\ \frac{3}{32} 2^{-2(b-1)} & ; \quad q = 2 \\ \frac{11}{128} 2^{-2(b-1)} & ; \quad q = 3 \end{cases}$$

For $q \geq 4$ the noise power is almost as that of the uniform RV, hence for those cases we use

$$\rho_{\varepsilon_q}^2 = \frac{1}{12} 2^{-2(b-1)} \quad ; \quad q \geq 4$$

Twiddle-Factors of the set $\{\mathbf{1}, -\mathbf{1}, \boldsymbol{j}, -\boldsymbol{j}\}$



Define $\mathcal{B}_1$-set - Butterflies that all their R inputs are of the set $\mathcal{T}_1$

The number of butterflies belonging to the $\mathcal{B}_1$-set vary between the FFT stages

β_m - The fraction of the butterflies belonging to the $\mathcal{B}_1$-set at stage m

$\rho_{q_m}^2$ - The self-noise power of the butterflies belonging to the $\mathcal{B}_1$-set at stage m

The updated noise power that incorporate the noise model of the butterflies belonging to the $\mathcal{B}_1$-set:

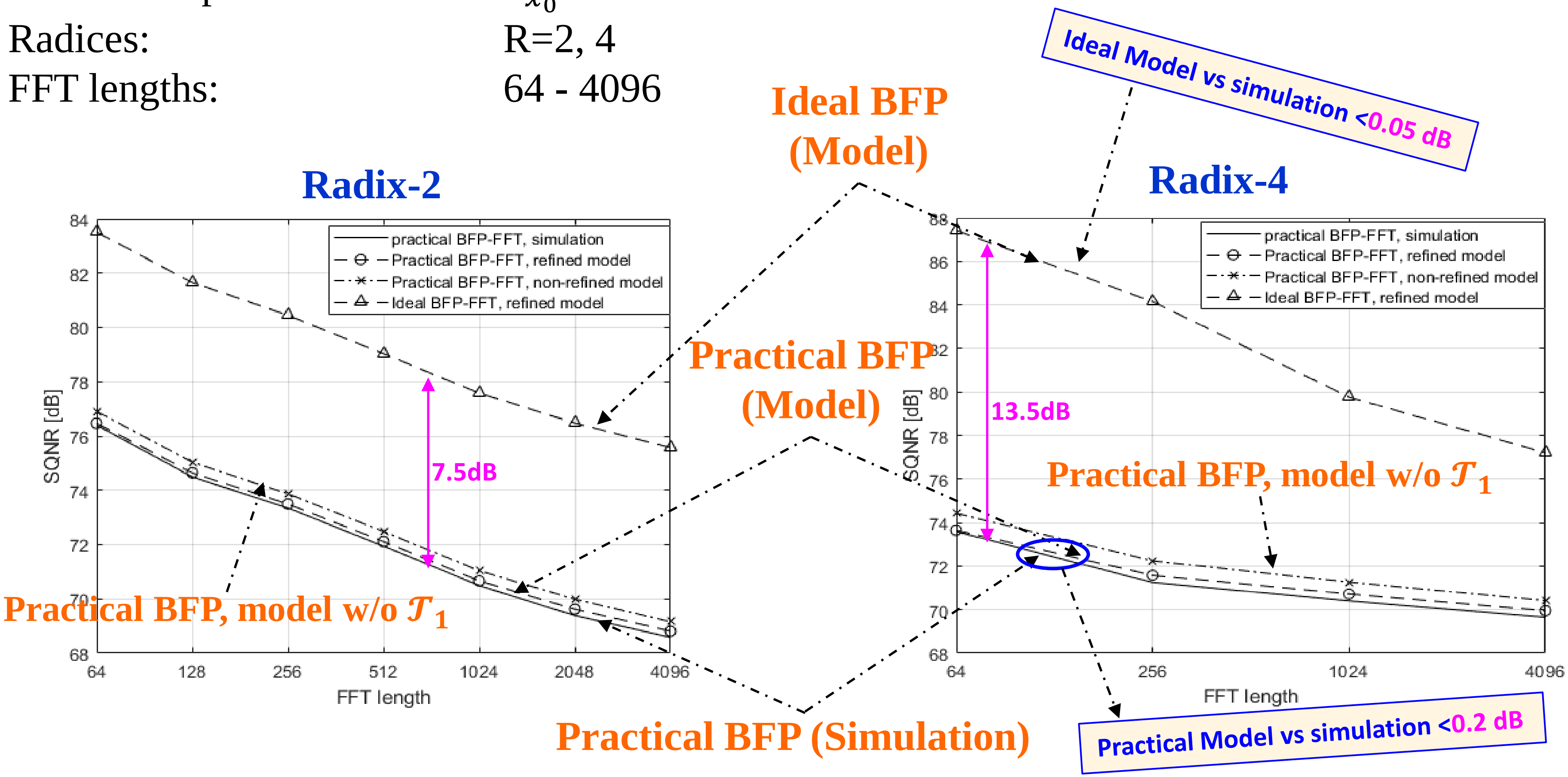
$$\rho_E^2 = \underbrace{\sigma_v^2 \sum_{m=1}^{M} R^{M-m+1} \prod_{i=m+1}^{M+1} \alpha_i^2}_{\text{The output noise power w/o incorporating the } \mathcal{B}_1 \text{ butterflies model}} + \underbrace{\sum_{m=1}^{M} \beta_m \overbrace{(\rho_{q_m}^2 - \sigma_v^2)}^{\text{Allways positive since } \rho_{q_m}^2 \geq \sigma_v^2} R^{M-m+1} \prod_{i=m+1}^{M+1} \alpha_i^2}_{\text{The increased output noise power when incorporating the } \mathcal{B}_1 \text{ butterflies model}}$$

Results

Results

Simulation conditions

Register width: $b = 16$
Gaussian inputs with $\sigma_{x_0} = 0.15$
Radices: R=2, 4
FFT lengths: 64 - 4096



Takeaways

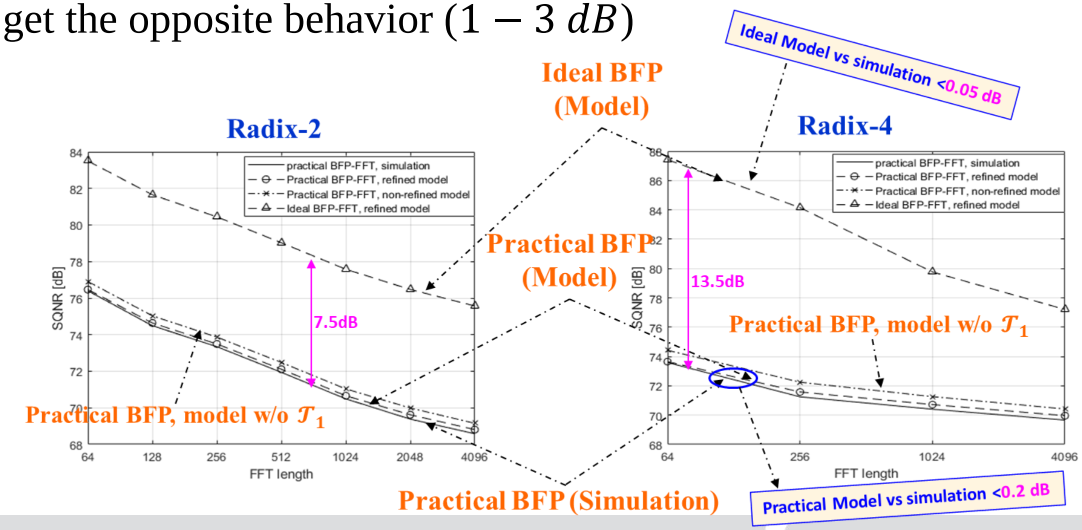
1. The developed models have **very good match** to the simulation results

   $(gap \leq 0.2\ dB\ /\ 0.05\ dB)$

2. The gap between the practical to the ideal BFP-FFT is $7 - 13.5\ dB$ (depending on the FFT size)

3. Not incorporating the model of the $\mathcal{T}_1$-set results in model optimistic by 0.5dB for radix-2 and by 1dB for radix-4

4. SQNR comparison between radix-2 and radix-4 BFP-FFT
   a. At the ideal BFP-FFT, radix-4 results in better SQNR than radix-2 $(2 - 4\ dB)$
   b. At the practical BFP-FFT, we get the opposite behavior $(1 - 3\ dB)$



Thank you

Backup