A 0.75 Million-Point Fourier Transform Chip for Frequency-Sparse Signals

Ezz El-Din Hamed

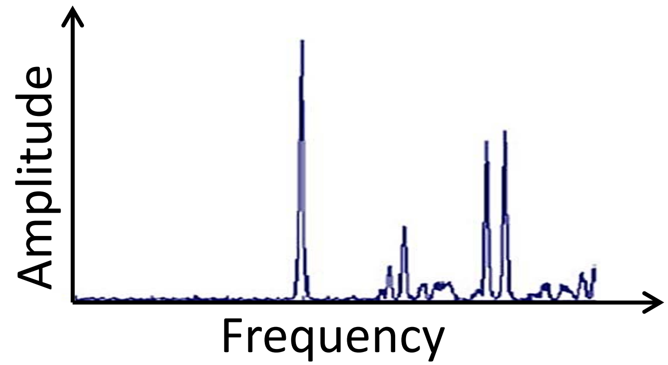
Omid Abari, Haitham Hassanieh, Abhinav Agarwal, Dina Katabi, Anantha Chandrakasan, Vladimir Stojanović

Massachusetts Institute of Technology

wireless @mit

Leverage Sparsity

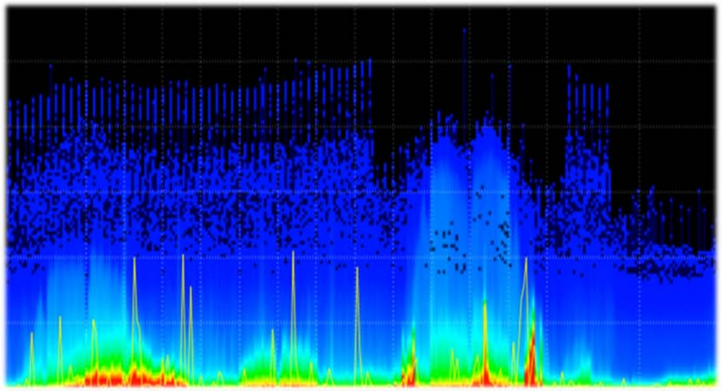
- Output of FFT is Sparse:
  – Most frequencies have zero energy or noise



  – Only few frequencies are active and have energy

- Use sparse FFT algorithm
  – Find and compute the active frequencies

Applications



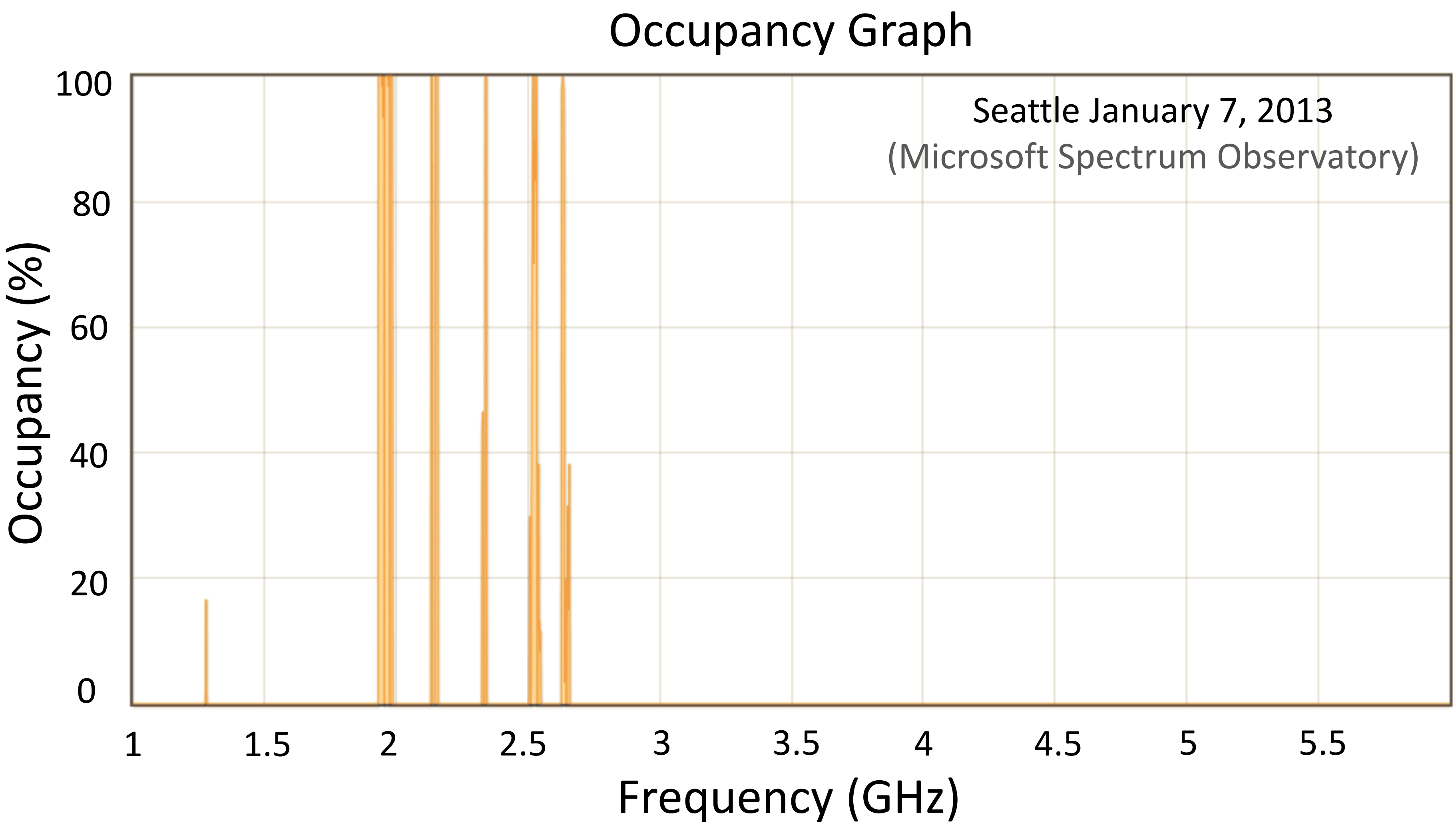
Spectrum Sensing

GPS

Radio Astronomy

- GHz spectrum sensing and acquisition
- Pattern matching by convolution with long code (e.g. GPS)
- Radio astronomy: data compression

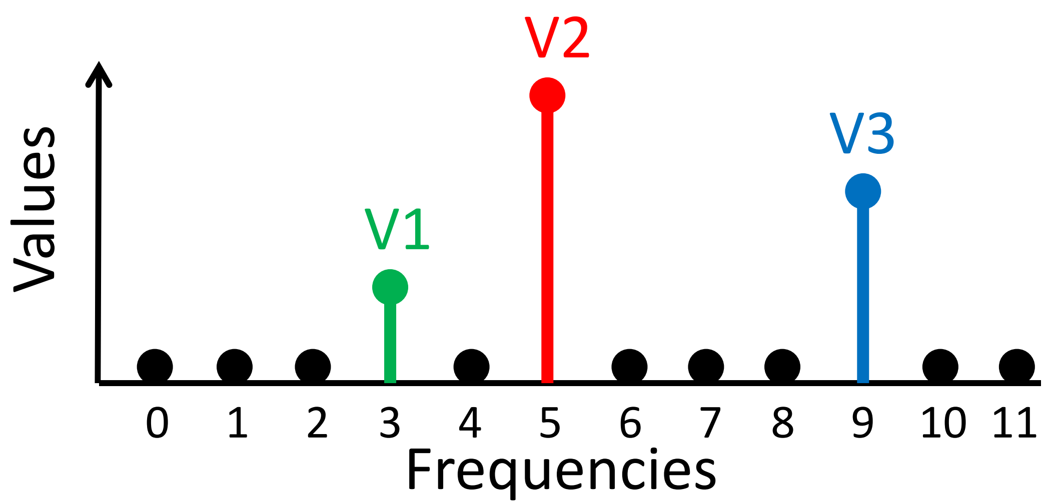
Wideband Spectrum Sensing



Outline

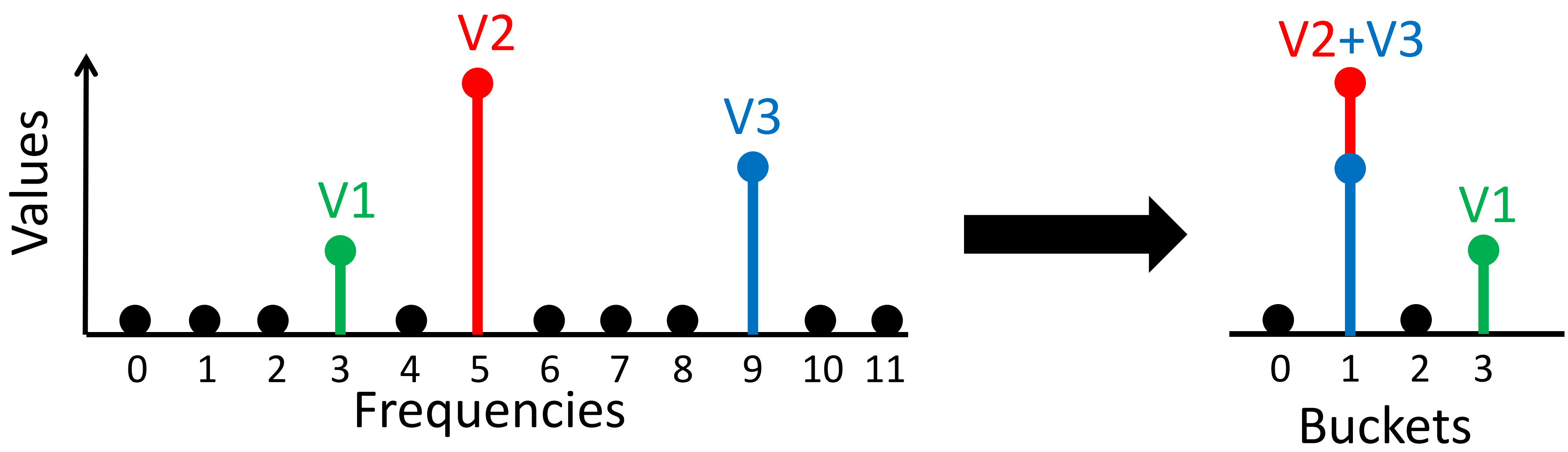
- Sparse FFT Algorithm
- Micro-architecture and Implementation
- Measurement Results
- Conclusion

How Does Sparse FFT Work?



1. **Bucketize**
2. **Estimate**
3. **Resolve collisions**

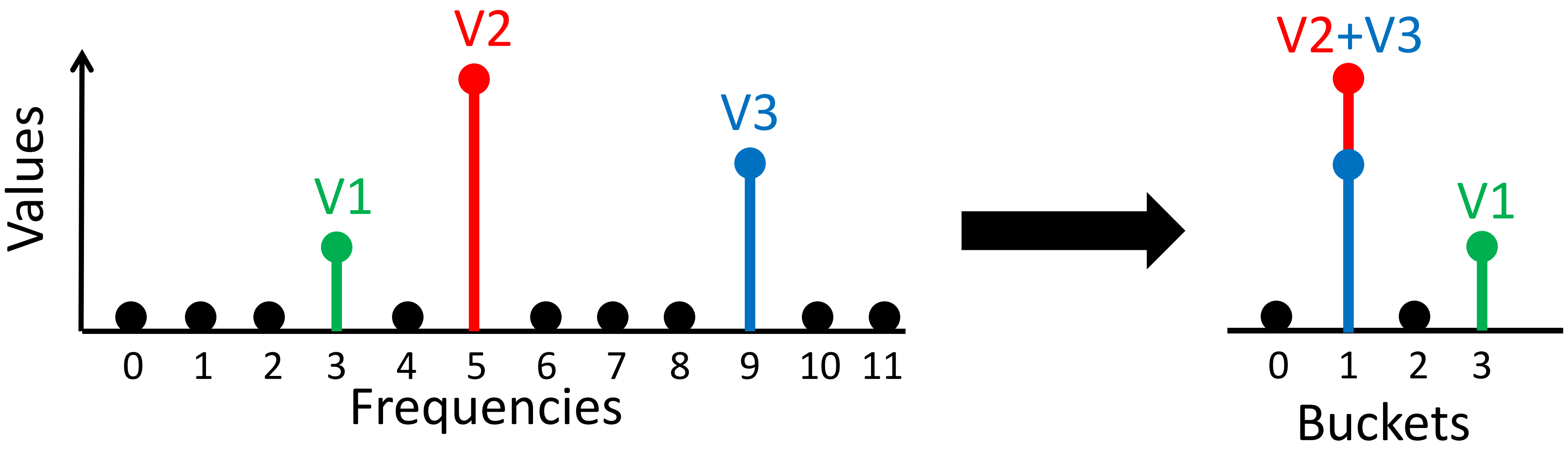
How Does Sparse FFT Work?



1. Bucketize

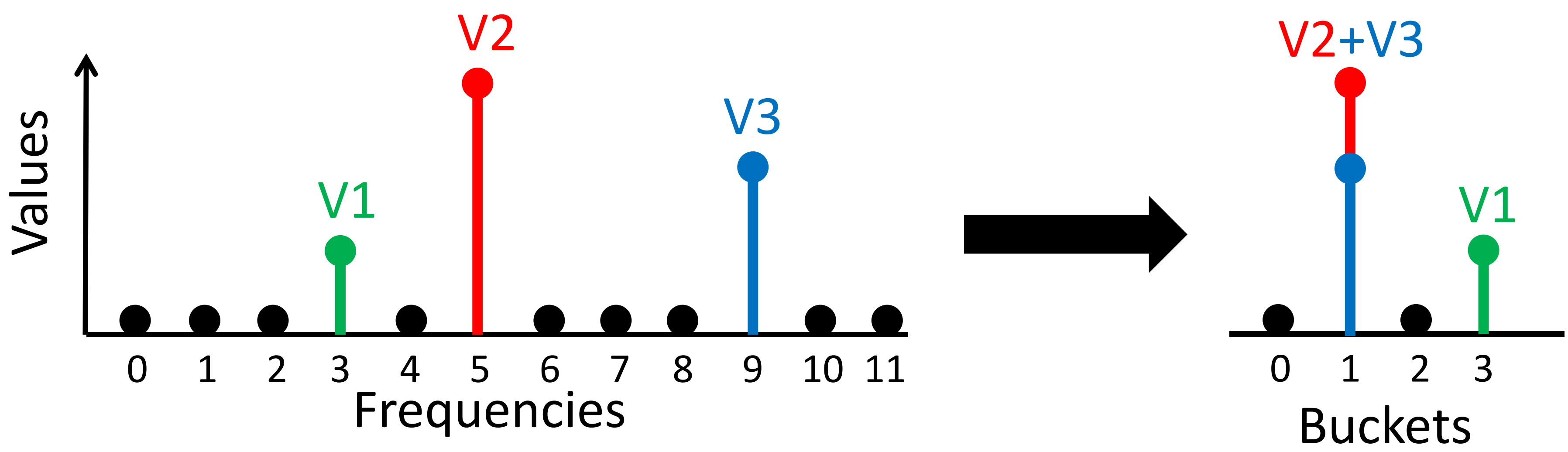
Sub-sampling time → Aliasing the frequencies

How Does Sparse FFT Work?



2. Estimate

How Does Sparse FFT Work?



2. Estimate

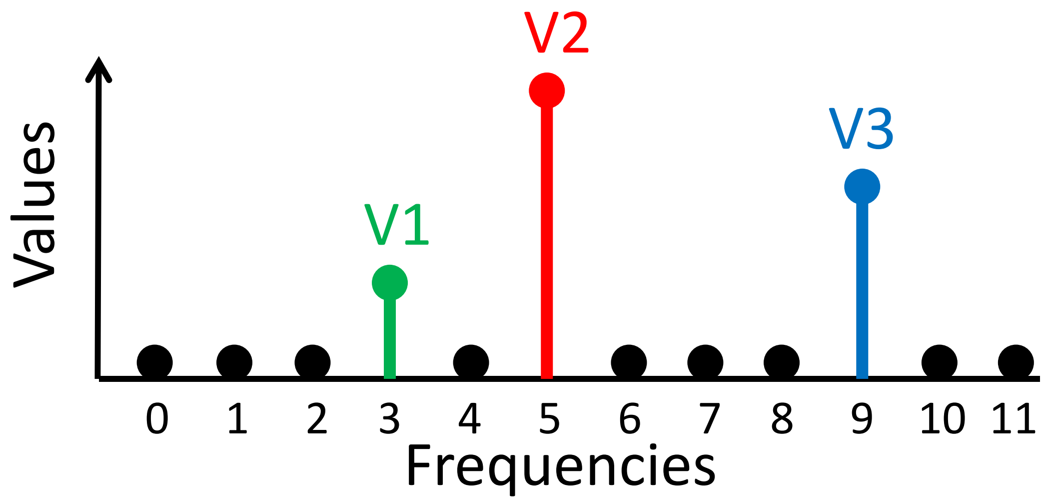
Repeat bucketization with a **time shift τ**

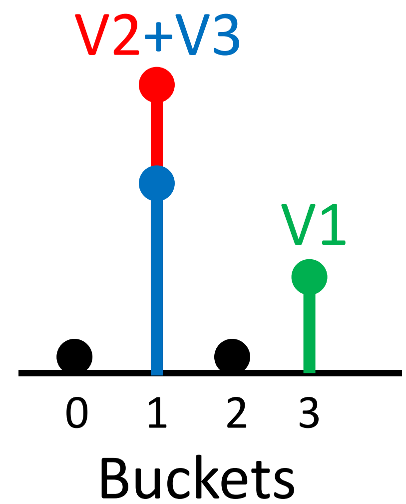
Time-Domain	**Freq-Domain**
$x(t)$	$X(f)$
$x(t-\tau)$	$X(f)e^{-j\theta}$

Phase Rotation

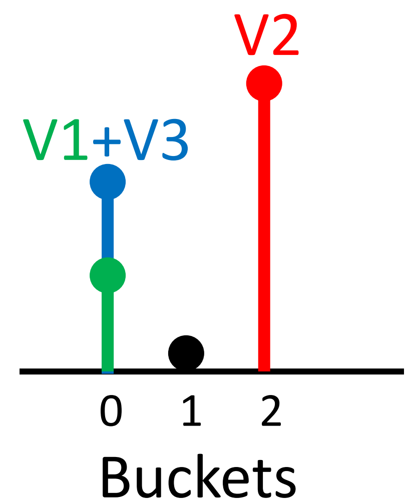
$$\theta = \frac{2\pi f \tau}{N}$$

How Does Sparse FFT Work?





3. Resolve Collisions



Repeat bucketization with a different subsampling rate (co-prime)

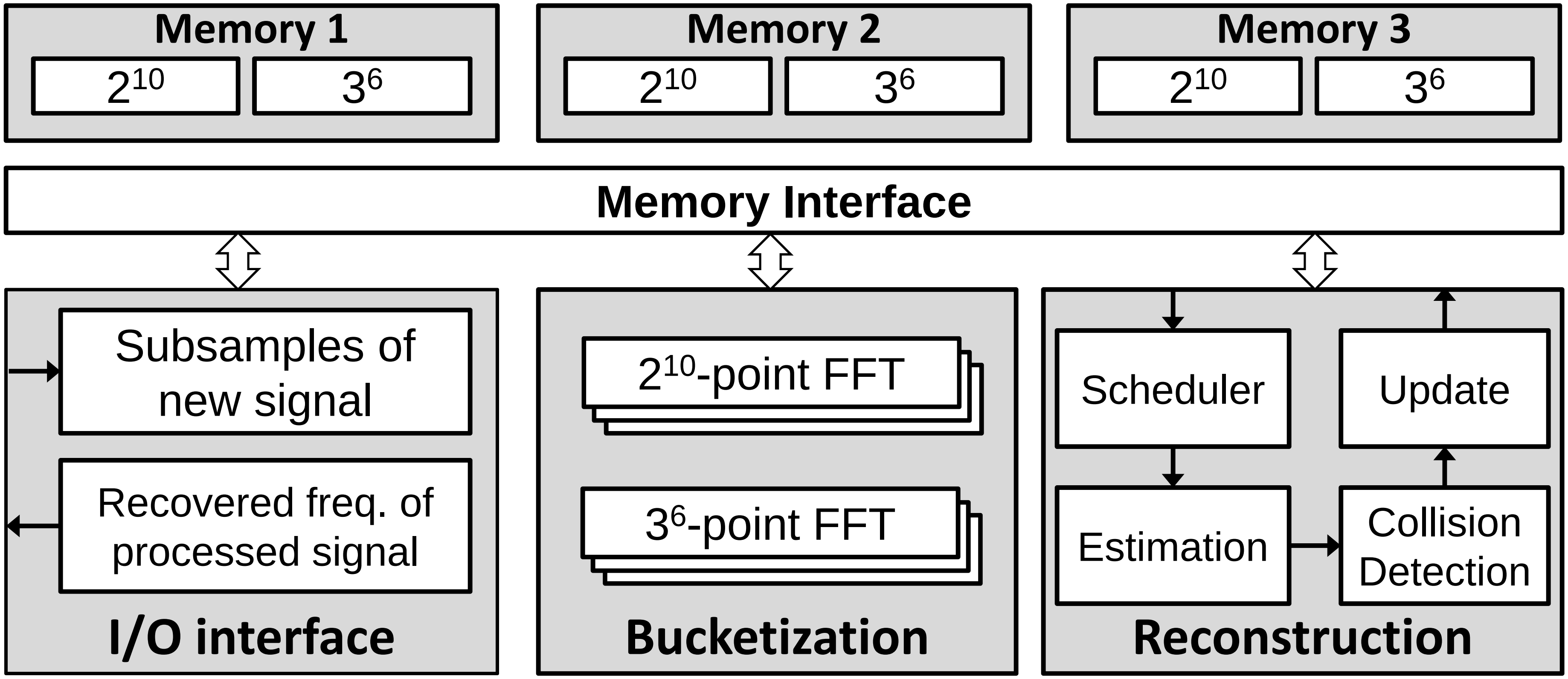
Implementation of Sparse FFT

- Sparse FFT size: $\boldsymbol{N=3^6 \times 2^{10}=746496}$
  - Subsample by 3^6 in time → **FFT of size 1024 (2^{10})**
  - Subsample by 2^{10} in time → **FFT of size 729(3^6)**

- Sparsity: outputs up to **750** active frequencies

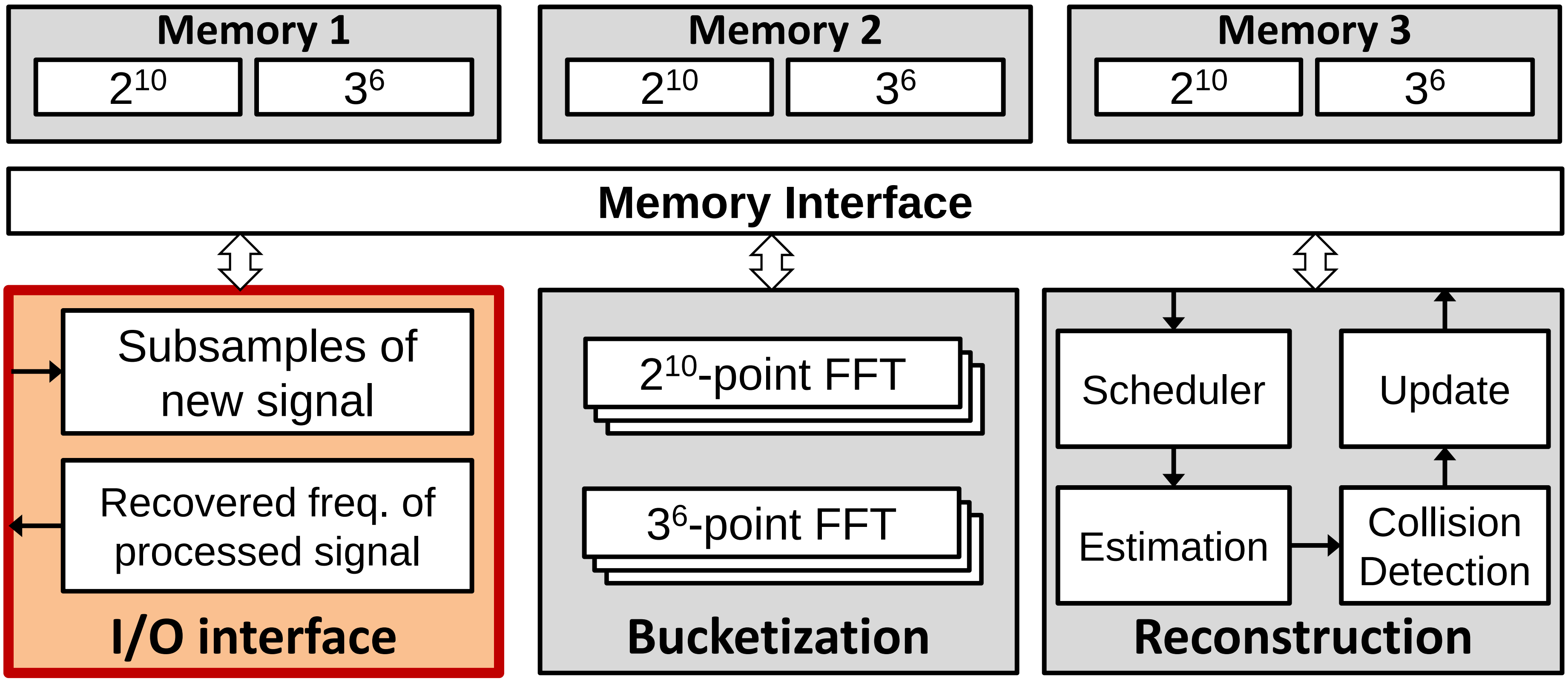
- Output resolution: **12 bit** real, **12 bit** imaginary

Sparse FFT Chip Block Diagram



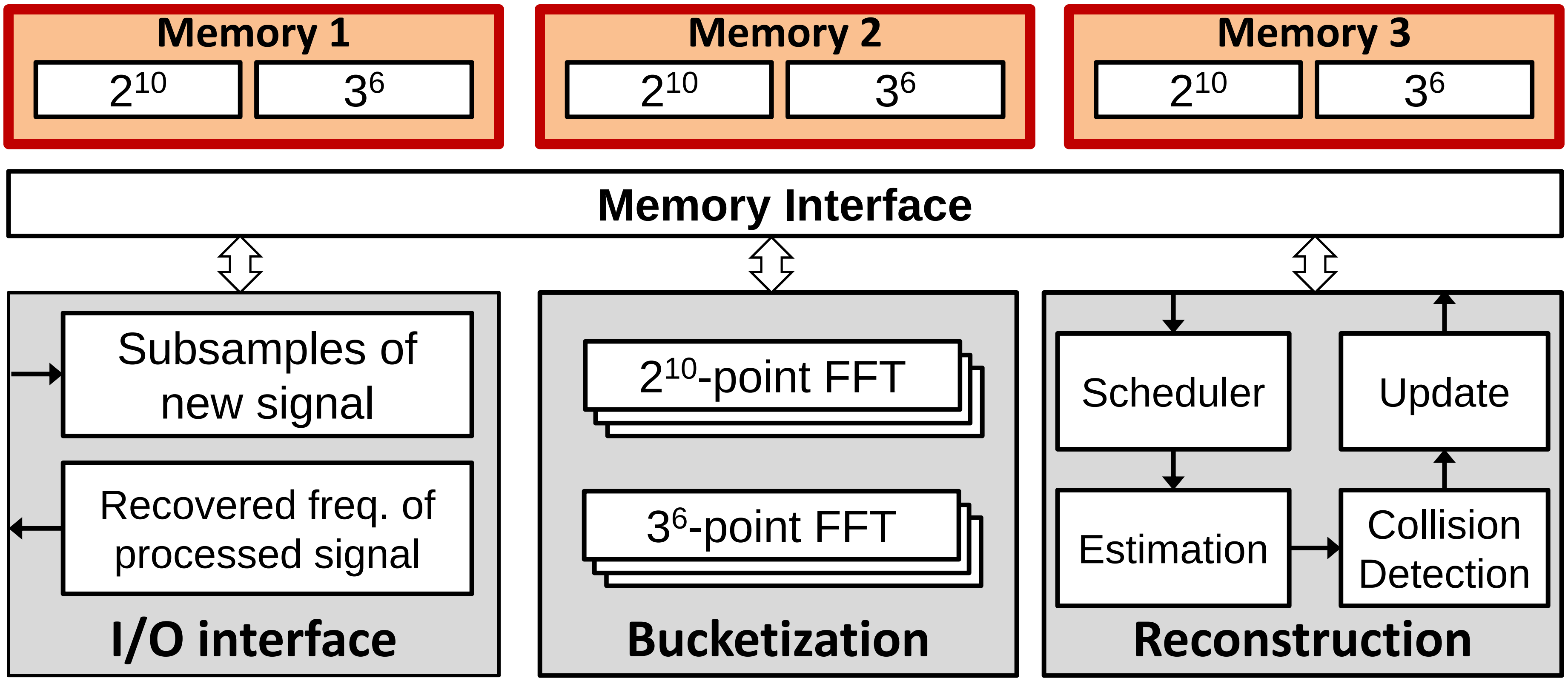
3 Pipelined Stages

Sparse FFT Chip Block Diagram



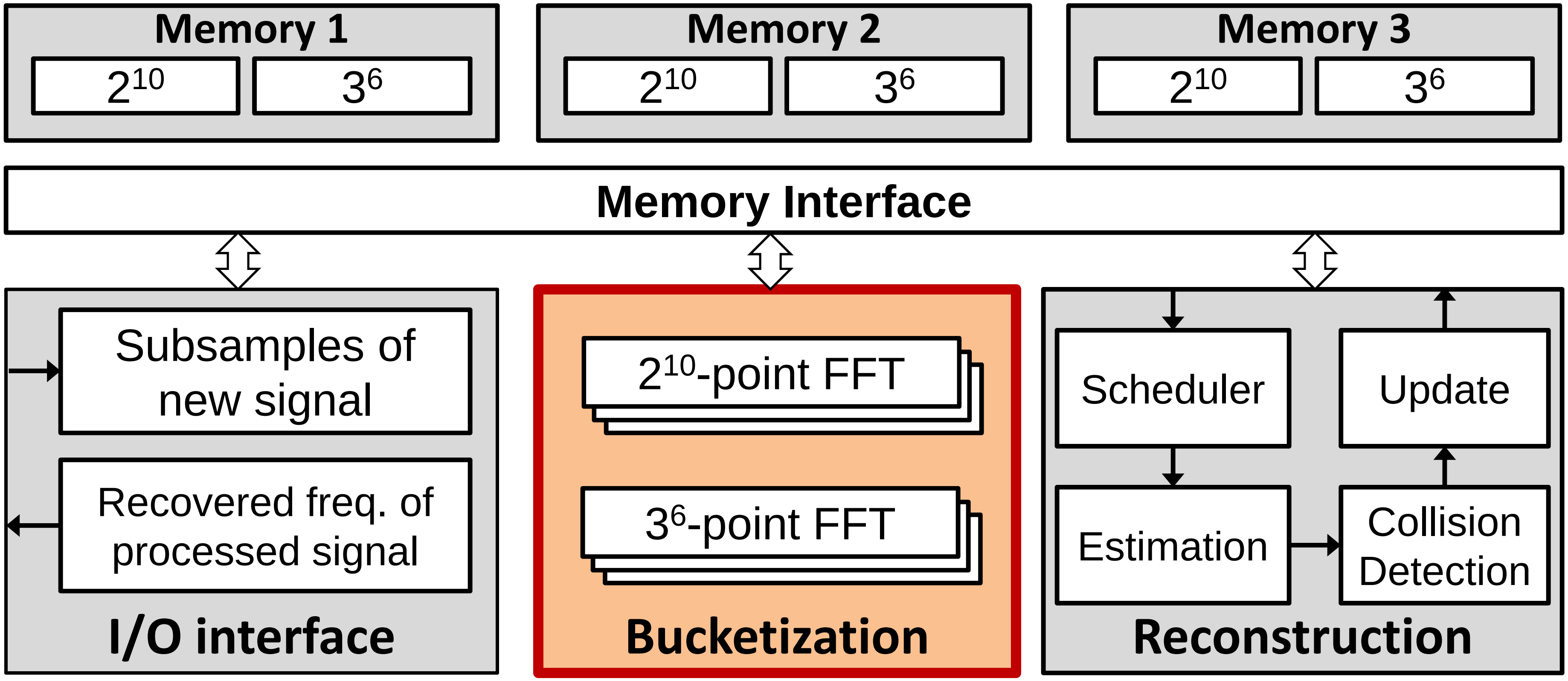
- **Input:** sub-sampled signal (by 2^{10} and 3^6)
- **Output**: positions and complex values of active frequencies

Sparse FFT Chip Block Diagram



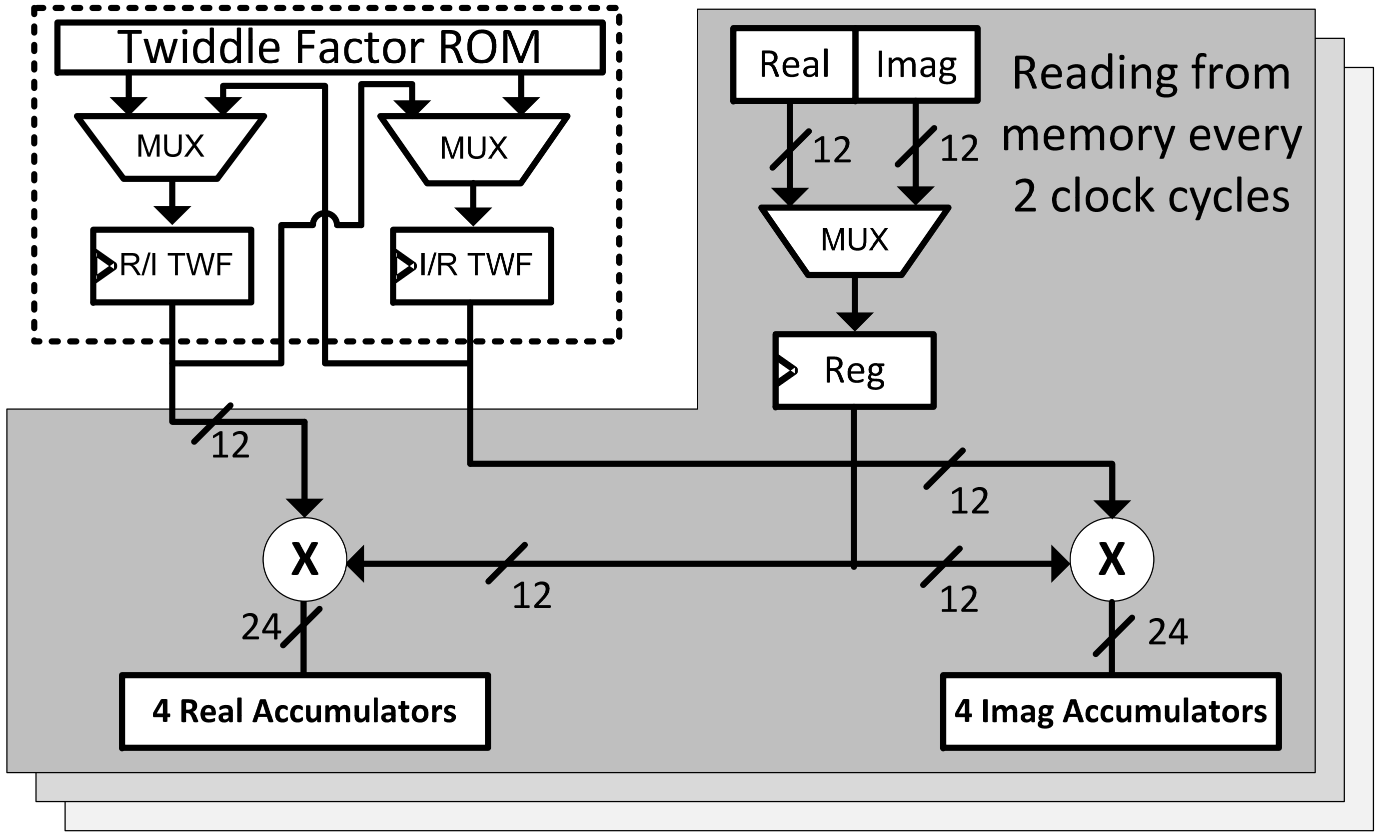
- 3 Memory sets, 2 SRAM blocks each
- Rotated across the 3 pipeline stages

Sparse FFT Chip Block Diagram



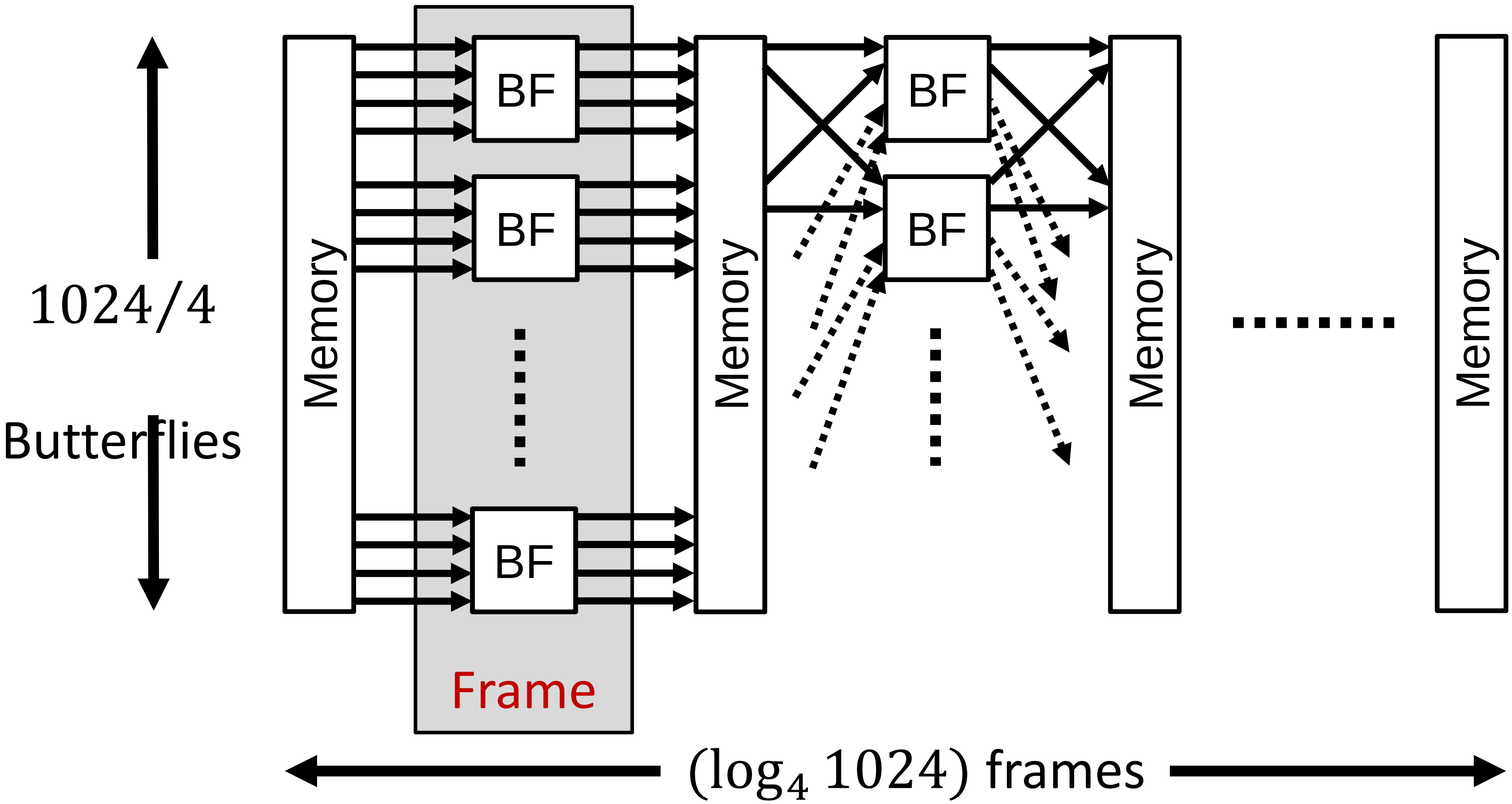
- In-place FFT
  - Radix-4 butterfly for 2^{10}-point FFT
  - Radix-3 butterfly for 3^6-point FFT
- Three FFTs of each type sharing the memory

2^{10} FFT Architecture



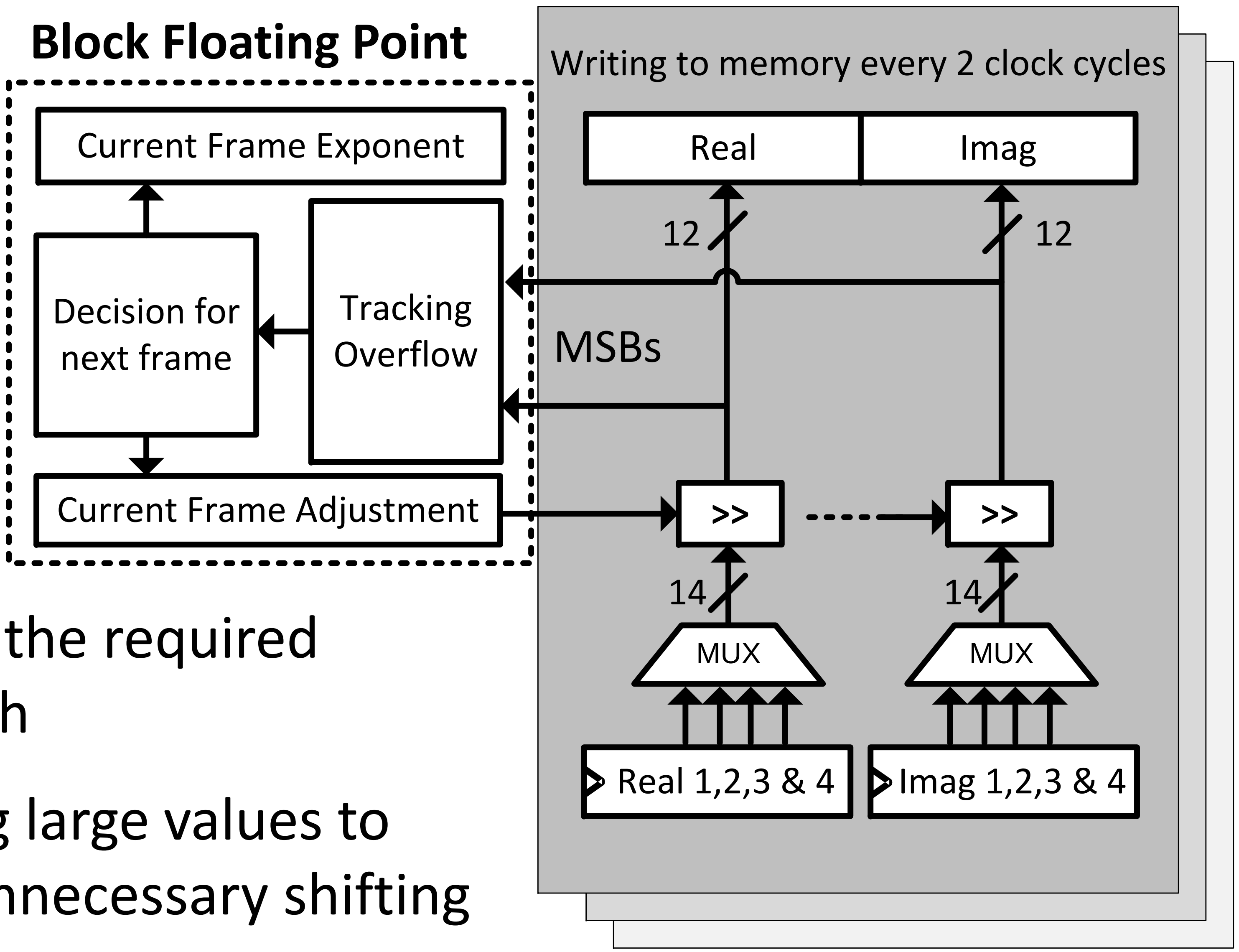
- Three radix-4 butterflies running in parallel
- Two 12b x 12b real multipliers per butterfly
- Multiplexing Real and Imaginary

2^{10} FFT Architecture



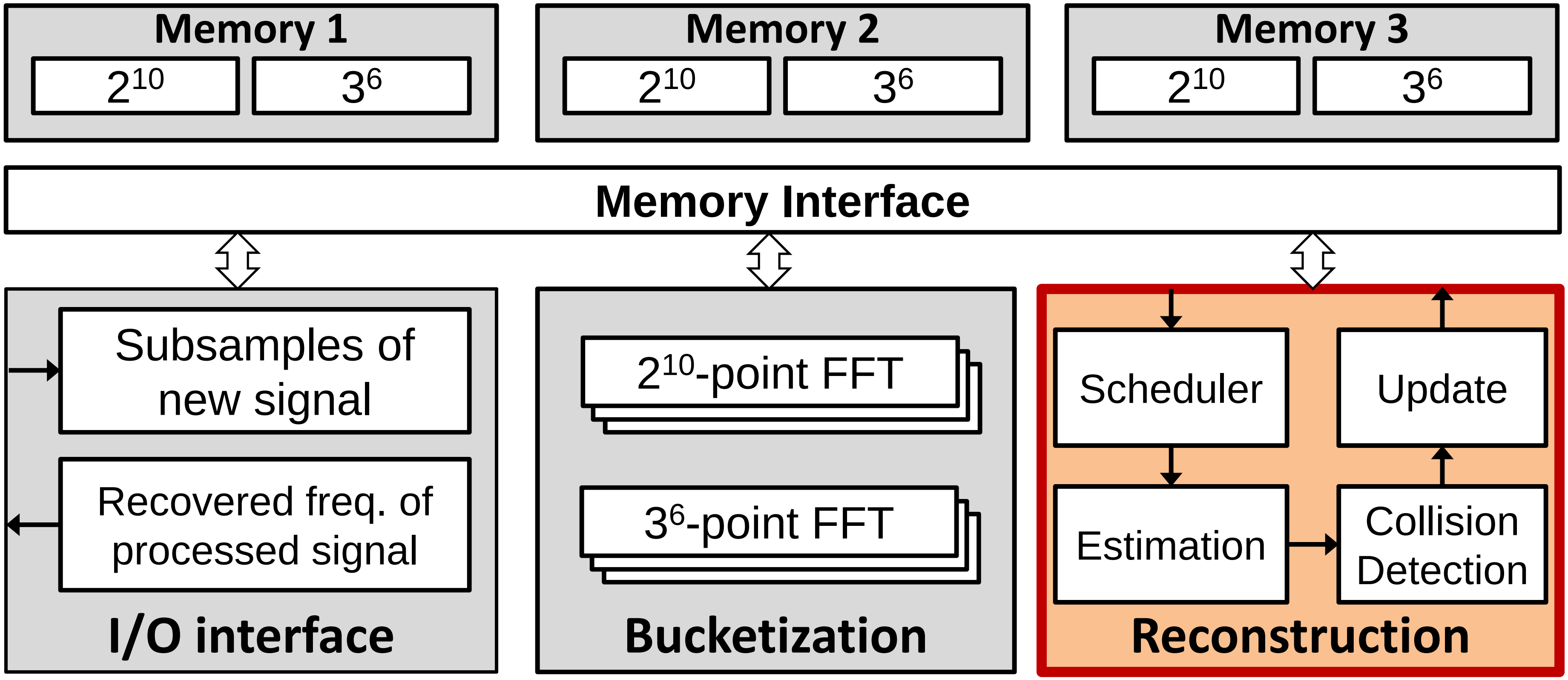
- In-place 1024-point FFT using a radix-4 butterfly
- 1024 additions (10 extra bits to account for overflow)

Dynamic Scaling



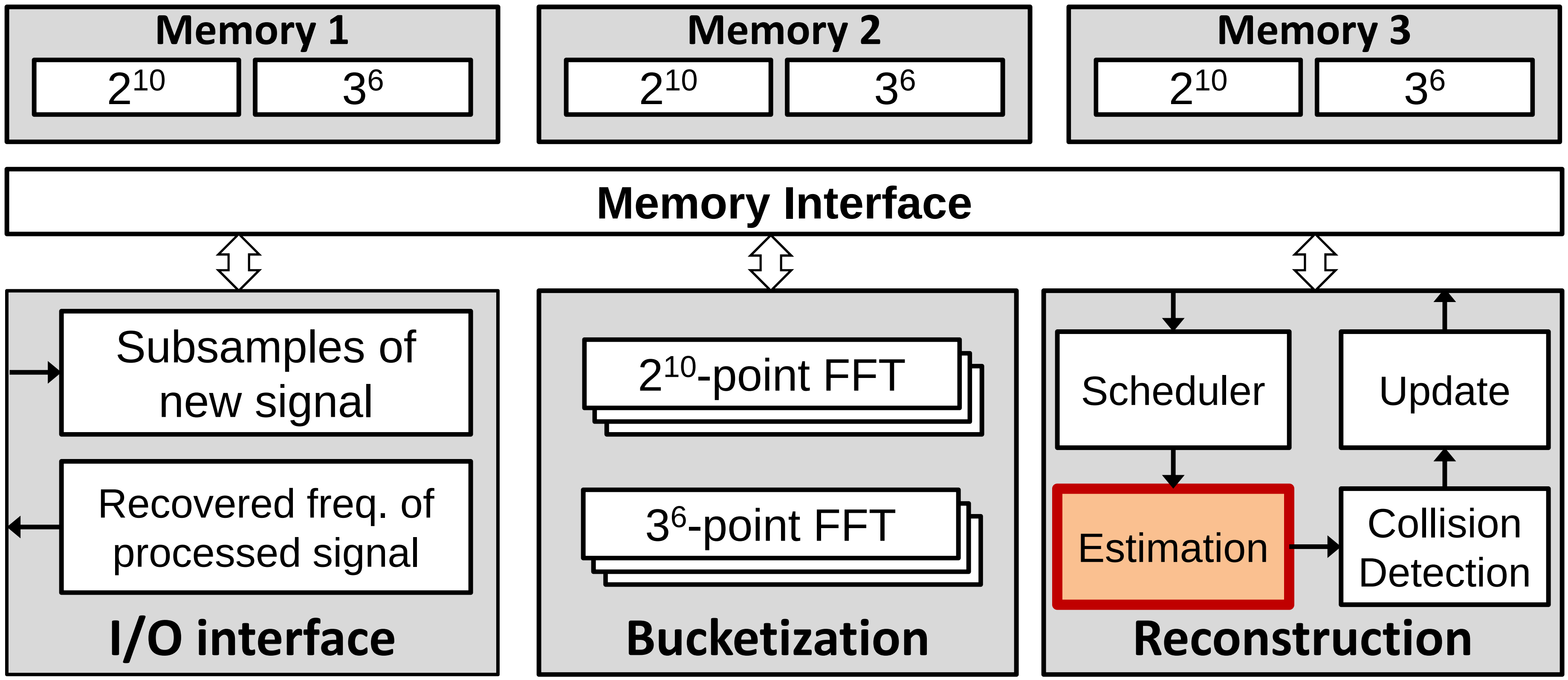
- Reduce the required bit width
- Tracking large values to avoid unnecessary shifting

Sparse FFT Chip Block Diagram



Reconstruction stage has 4 sub-blocks

Sparse FFT Chip Block Diagram



Estimation Sub-block

- Estimate frequency position f where $0 \leq f \leq N - 1$
- For $N \approx 0.75$ million ➔ Use 20 bits to represent f

Estimation Sub-block

- Estimate frequency position f where $0 \leq f \leq N-1$
- For $N \approx 0.75$ million ➔ Use 20 bits to represent f
- For 2^{10} bucketization: frequency f aliases to bucket number b:

$$b = f \bmod 2^{10}$$

⬇

10 LSBs of f can be estimated from bucket number b

10 remaining MSBs are estimated using phase rotation

Estimation Sub-block

Use phase rotation from a time shift τ to estimate the 10 MSBs of f :

$$\theta = \frac{2\pi f \tau}{N} \quad \Longrightarrow \quad f = \frac{N\theta}{2\pi\tau}$$

Estimation Sub-block

Use phase rotation from a time shift τ to estimate the 10 MSBs of f :

$$\theta = \frac{2\pi f \tau}{N} \quad \longrightarrow \quad f = \frac{N\theta}{2\pi\tau}$$

If $\tau > 1$, phase wraps around 2π

⬇

$f_1 \neq f_2$ create same phase rotation
→loose highest MSBs

⬇

Use time shift $\tau = 1$ to get MSBs

Estimation Sub-block

Use phase rotation from a time shift τ to estimate the 10 MSBs of f :

$$\theta = \frac{2\pi f \tau}{N} \longrightarrow f = \frac{N\theta}{2\pi\tau}$$

If $\tau > 1$, phase wraps around 2π

⬇

$f_1 \neq f_2$ create same phase rotation

→loose highest MSBs

⬇

Use time shift $\tau = 1$ to get MSBs

⬇

Noise in θ creates error

⬇

Use large time shift τ to average noise in θ

Estimation Sub-block

Use phase rotation from a time shift τ to estimate the 10 MSBs of f :

$$\theta = \frac{2\pi f \tau}{N} \quad \longrightarrow \quad f = \frac{N\theta}{2\pi\tau}$$

If $\tau > 1$, phase wraps around 2π

$f_1 \neq f_2$ create same phase rotation

→loose highest MSBs

Use time shift $\tau = 1$ to get MSBs

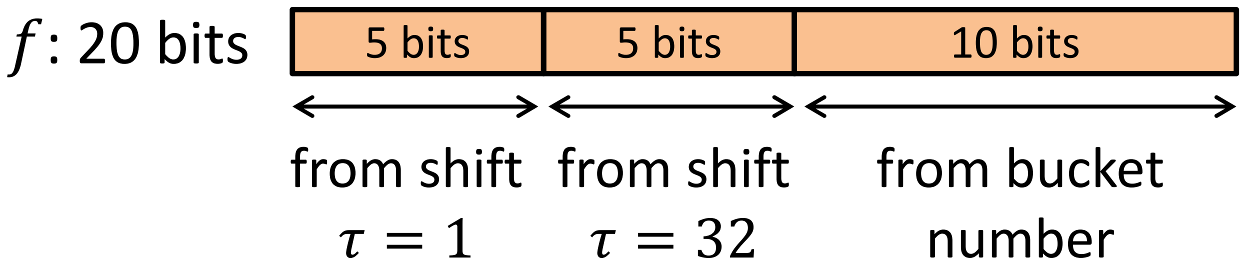
Noise in θ creates error

Use large time shift τ to average noise in θ

Use two different time shifts
$\tau = 1$ and $\tau = 32$ to estimate the 10 MSBs of f

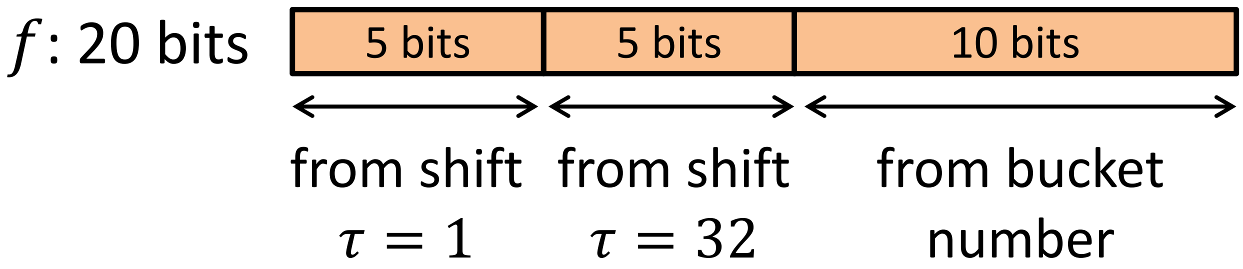
Estimation Sub-block

Frequency Recovery from 2^{10} bucketization



Estimation Sub-block

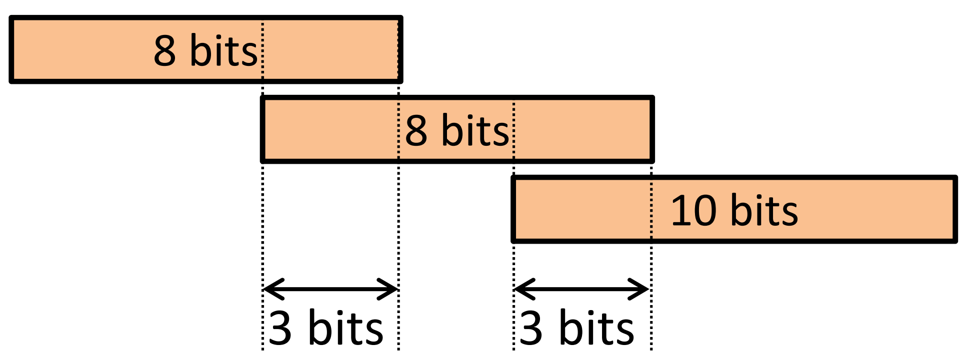
Frequency Recovery from 2^{10} bucketization



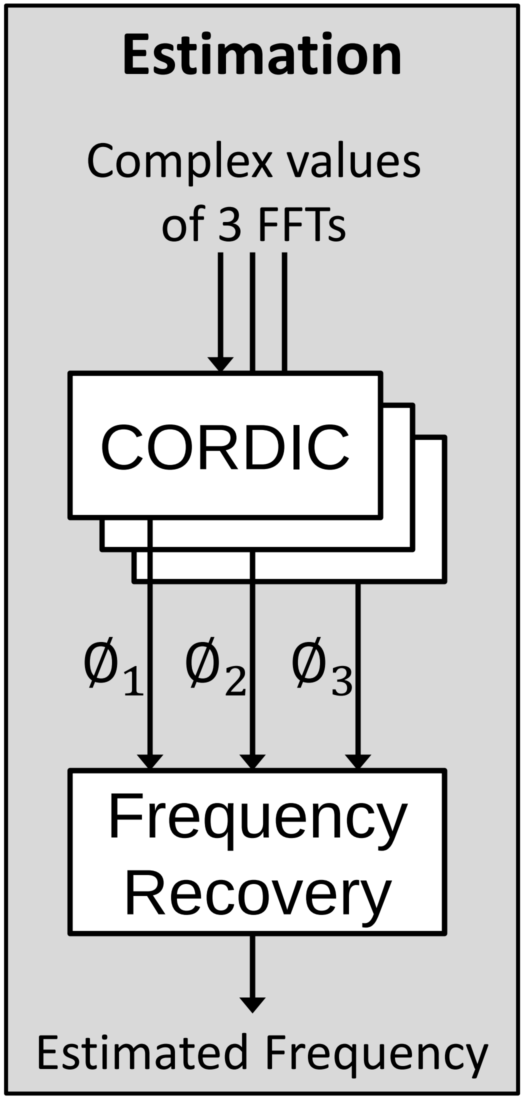
Problem: Small error propagates

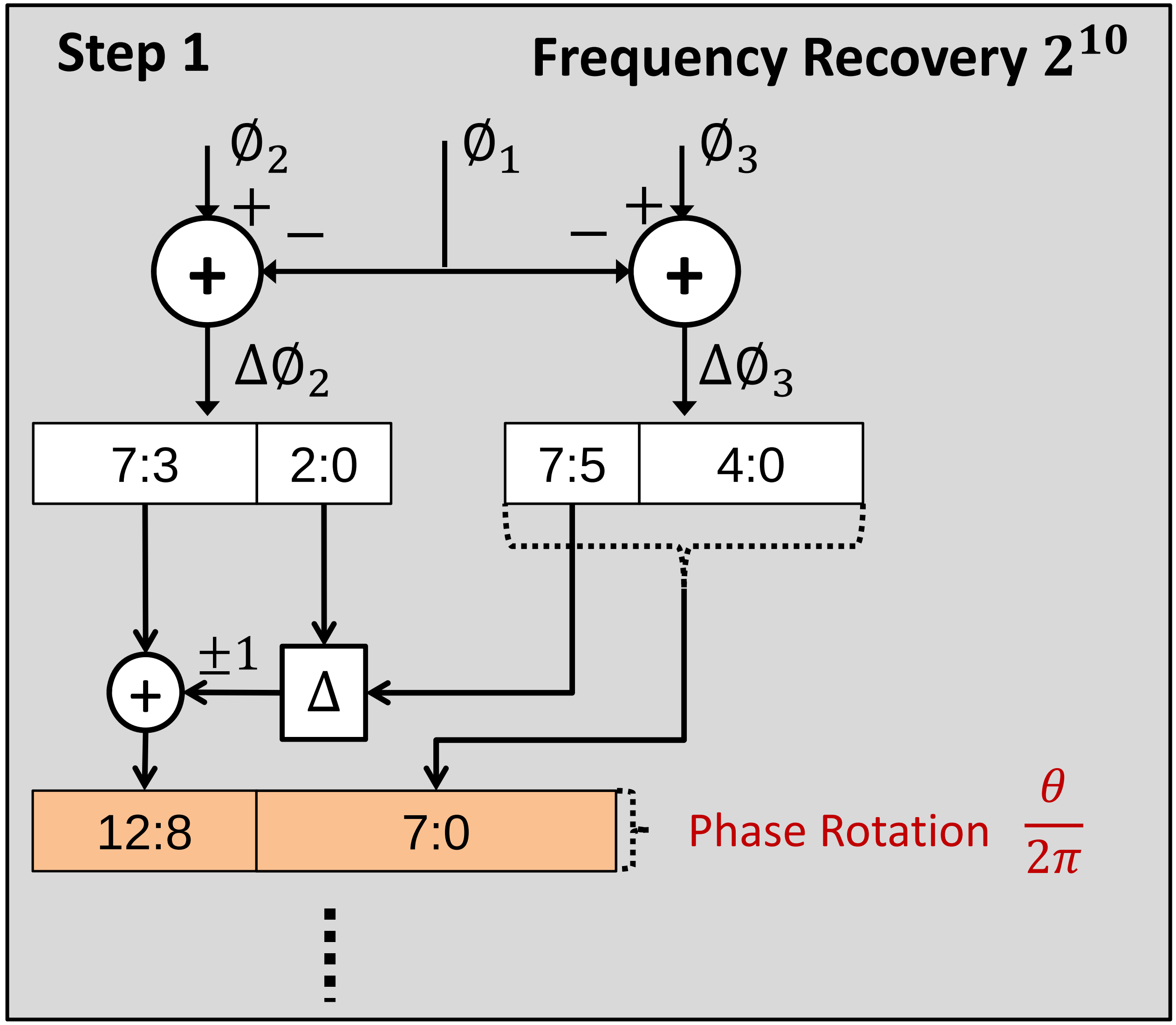
→ Affect MSBs

Solution: Use 3 bit overlap to detect errors

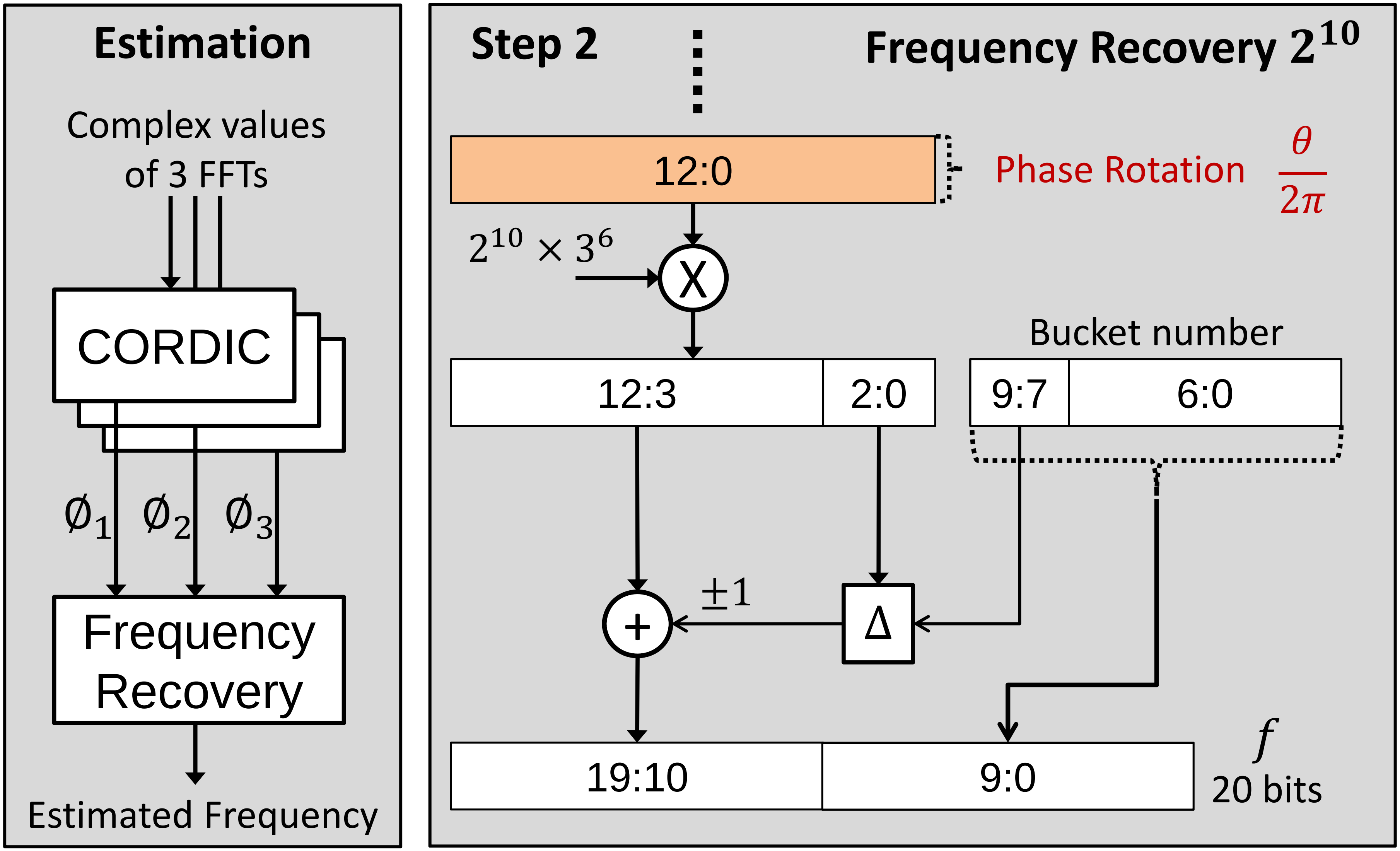


Estimation Sub-block





Estimation Sub-block



Estimation Sub-block

- For 3^6 bucketization: frequency f aliases to bucket number b:

$$b = f \bmod 3^6$$

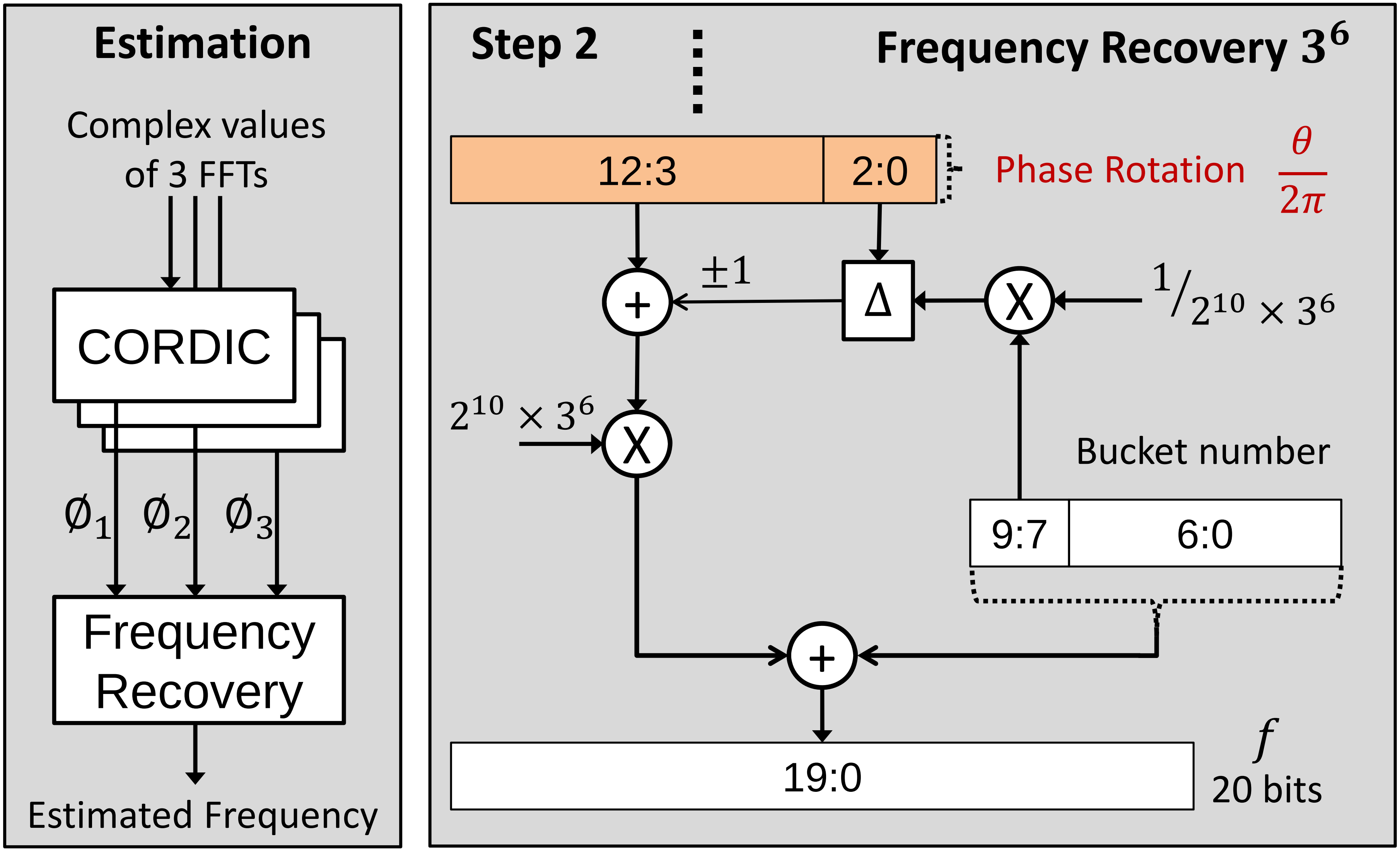
- LSBs cannot be replaced directly by the bucket number

- Naïve implementation:
  1. Estimate f from the phase rotation
  2. Calculate the remainder ($f \bmod 3^6$)
  3. Subtract this remainder from f
  4. Then add the bucket number *b*

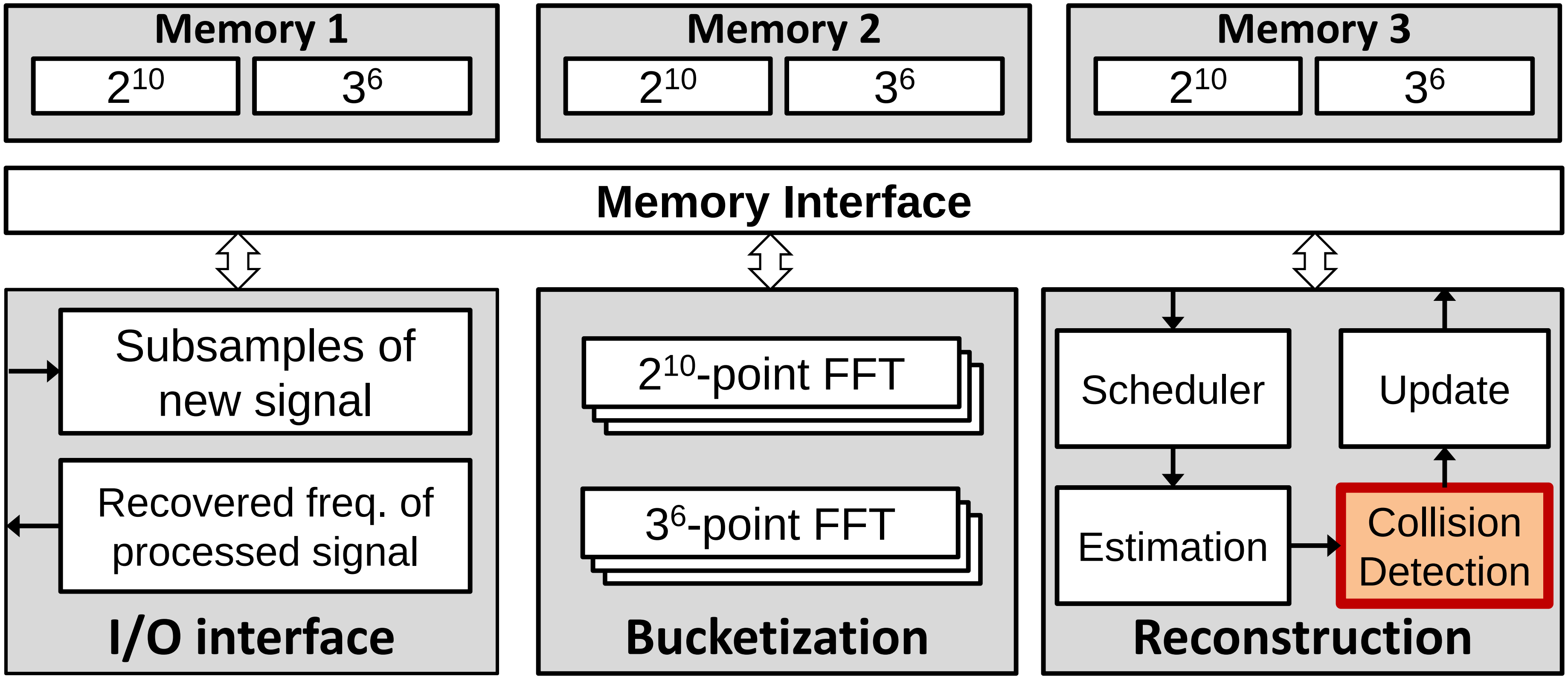
$$f = \frac{N\theta}{2\pi\tau}$$

- Steps 2 and 3 are done indirectly by truncating the LSBs of θ

Estimation Sub-block



Sparse FFT Chip Block Diagram



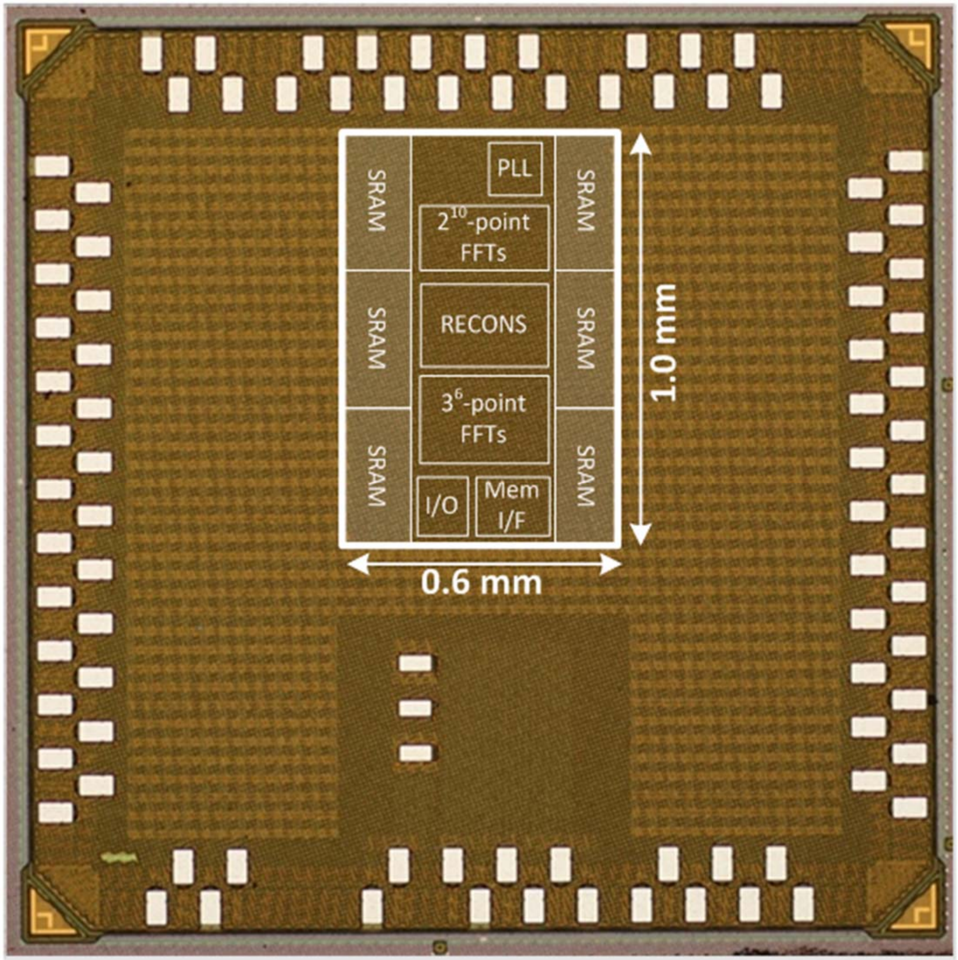
Collision Detection

- No Collision:
  - Single active frequency
  - Time shift causes only change in phase and not in magnitude of the bucket

Collision Detection

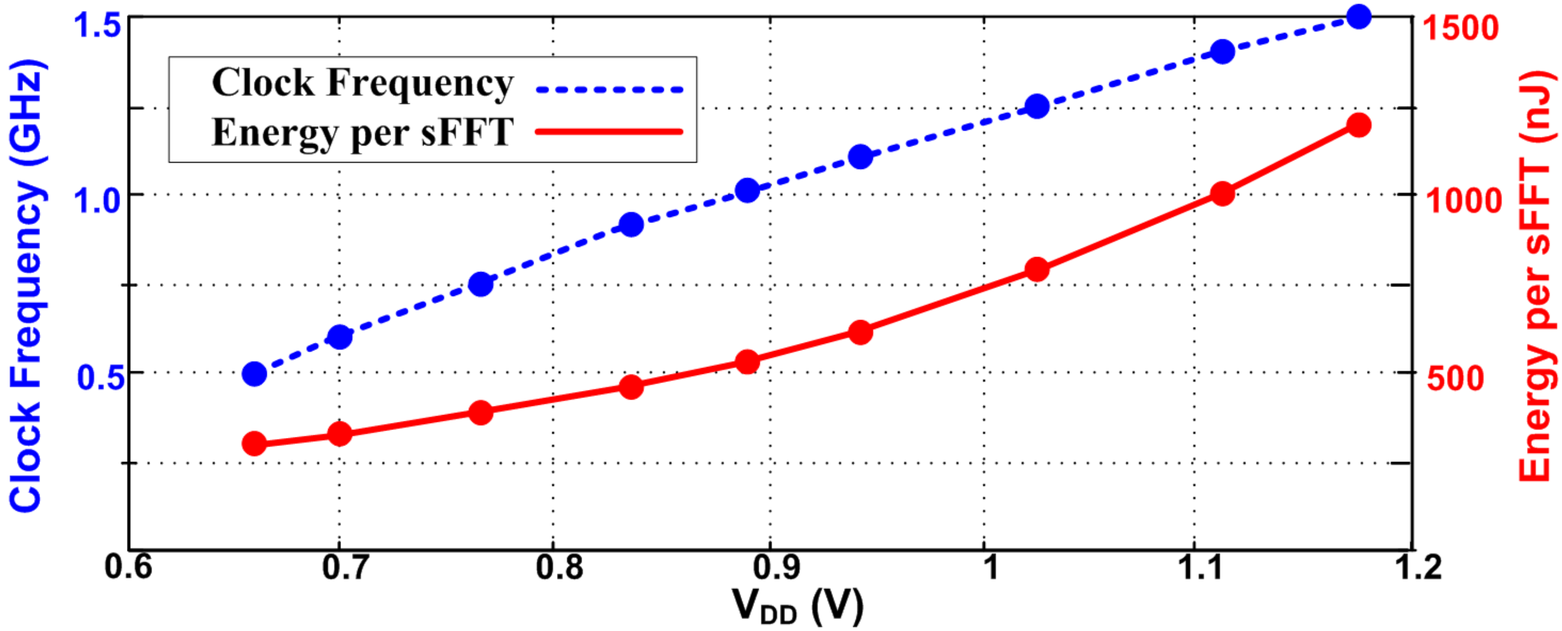
- No Collision:
  - Single active frequency
  - Time shift causes only change in phase and not in magnitude of the bucket
- Collision:
  - Multiple active frequencies sum up with different phase rotations
  - Time shift causes the magnitude of the bucket to change

Die photo

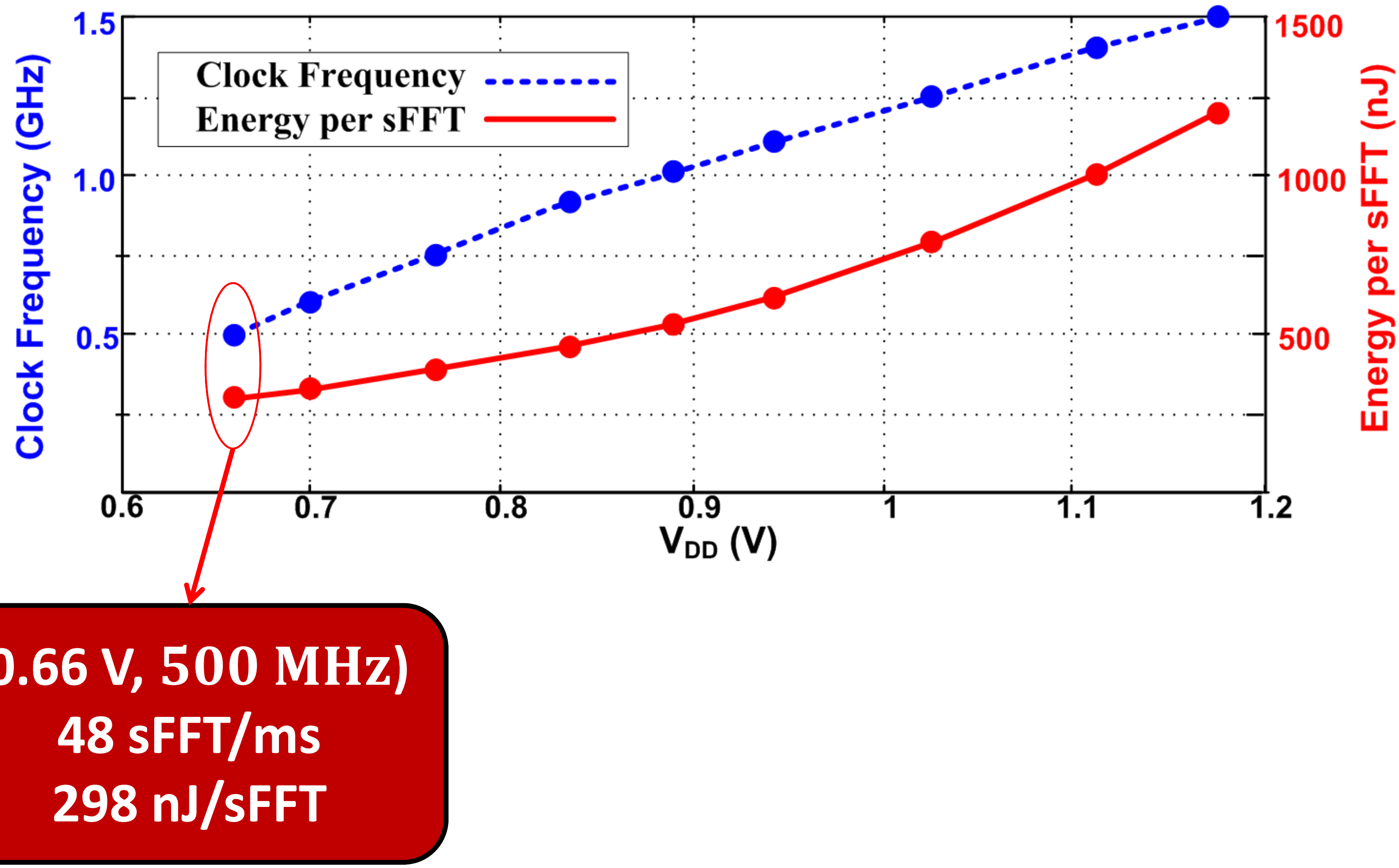


Technology	45nm SOI CMOS
Core Area	0.6mm x 1.0mm
Core Supply Voltage	0.66-1.18 V
Clock Frequency	0.5-1.5 GHz
Core Power	14.6 - 174.8 mW
Latency	63 - 21 µs

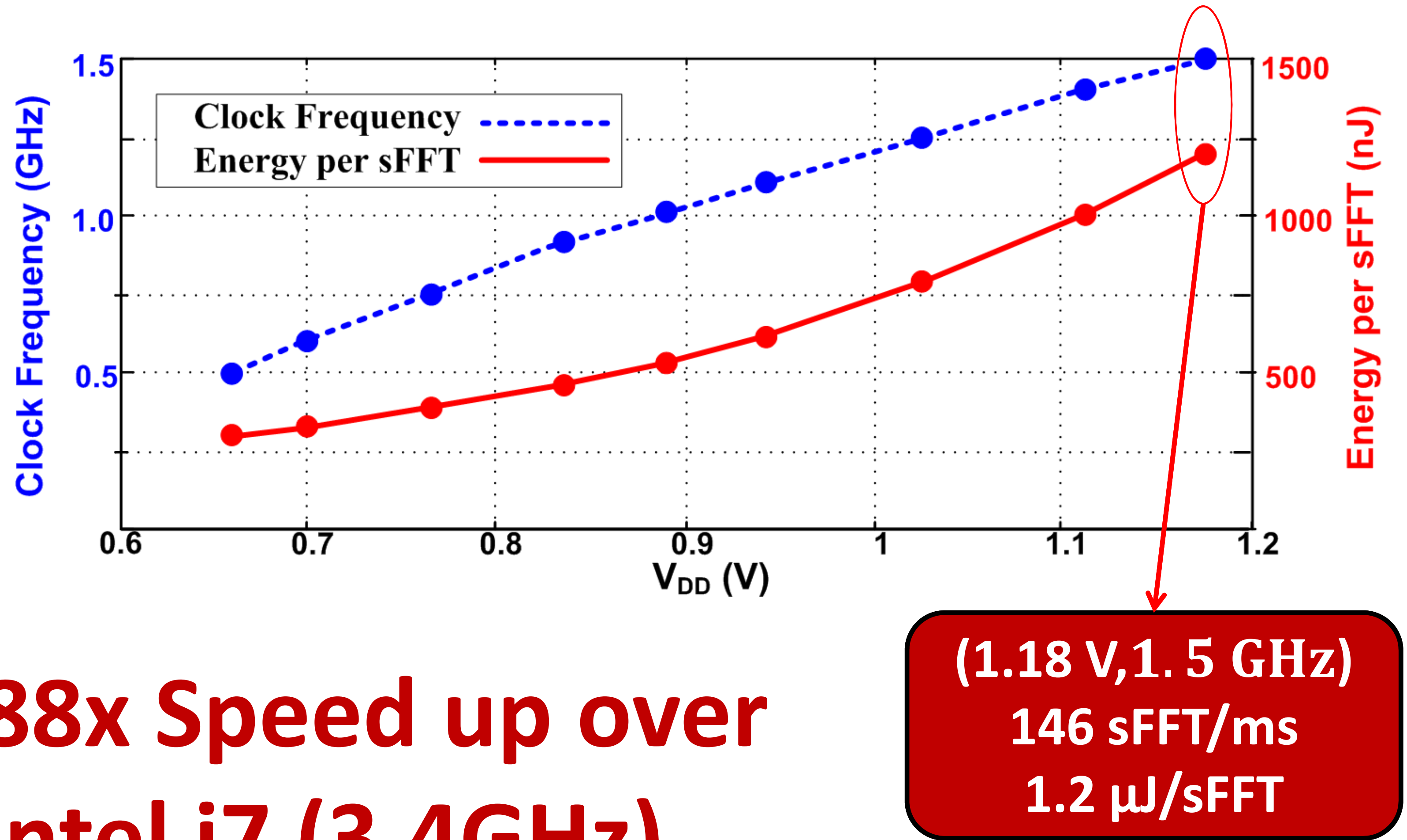
Measurement Result



Measurement Result

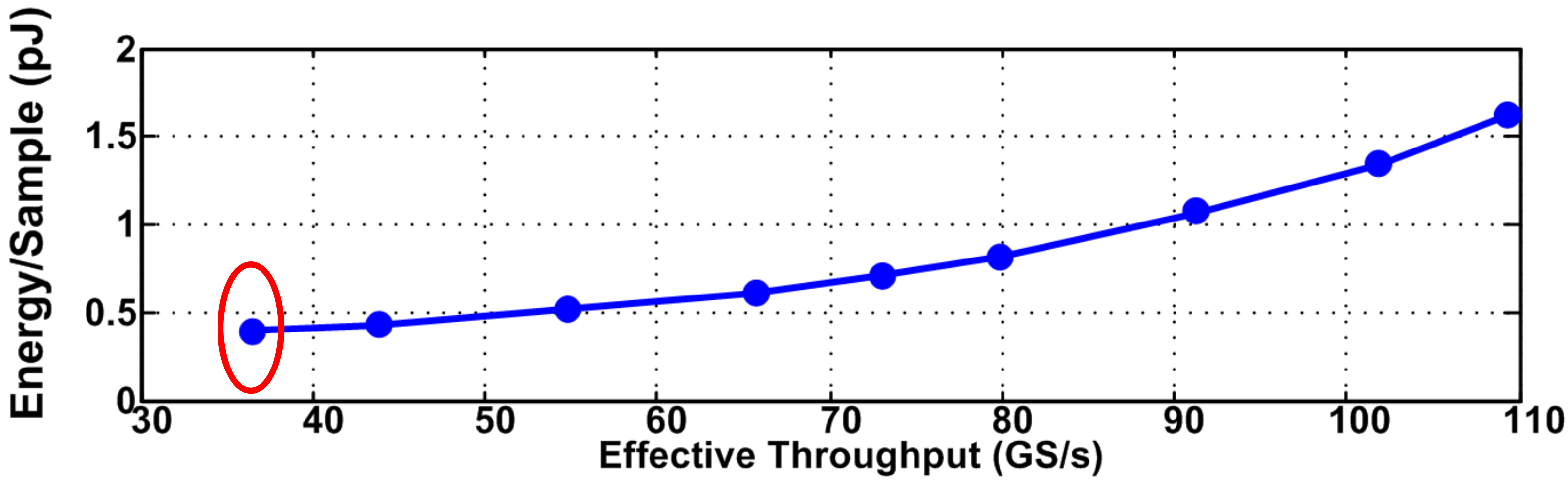


Measurement Result



88x Speed up over intel i7 (3.4GHz)

Measurement Result



At 0.66V:

- Effective throughput 36GS/s
- Consuming 0.4pJ/sample

Comparison

	ISSCC'11	JSSC'08	JSSC'12	**This work**
Technology	65 nm	90 nm	65nm	45 nm
Signal Type	Any Signal	Any Signal	Any Signal	Freq.-Sparse signal
Size	2^{10}	2^{8}	2^{7} to 2^{11}	3^{6}x2^{10}
Word width	16 bits	10 bits	12 bits	12 bits
Area	8.29mm^2	5.1mm^2	1.37mm^2	0.6mm^2
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36-109GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6pJ	0.4-1.6pJ

Comparison

	ISSCC'11	JSSC'08	JSSC'12	**This work**
Technology	65 nm	90 nm	65nm	45 nm
Signal Type	Any Signal	Any Signal	Any Signal	Freq.-Sparse signal
Size	2^{10}	2^{8}	2^{7} to 2^{11}	3^{6}x2^{10}
Word width	16 bits	10 bits	12 bits	12 bits
Area	$8.29mm^2$	$5.1mm^2$	$1.37mm^2$	$0.6mm^2$
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36-109GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6pJ	0.4-1.6pJ

Sparse FFT works only for frequency-sparse signals

Comparison

	ISSCC'11	JSSC'08	JSSC'12	**This work**
Technology	65 nm	90 nm	65nm	45 nm
Signal Type	Any Signal	Any Signal	Any Signal	Freq.-Sparse signal
Size	2^{10}	2^{8}	2^{7} to 2^{11}	3^{6}x2^{10}
Word width	16 bits	10 bits	12 bits	12 bits
Area	8.29mm^2	5.1mm^2	1.37mm^2	0.6mm^2
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36-109GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6pJ	0.4-1.6pJ

Comparison

	ISSCC'11	JSSC'08	JSSC'12	**This work**
Technology	65 nm	90 nm	65nm	45 nm
Signal Type	Any Signal	Any Signal	Any Signal	Freq.-Sparse signal
Size	2^{10}	2^{8}	2^{7} to 2^{11}	3^{6}x2^{10}
Word width	16 bits	10 bits	12 bits	12 bits
Area	8.29mm^2	5.1mm^2	1.37mm^2	0.6mm^2
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36-109GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6pJ	0.4-1.6pJ

Comparison

	ISSCC'11	JSSC'08	JSSC'12	**This work**
Technology	65 nm	90 nm	65nm	45 nm
Signal Type	Any Signal	Any Signal	Any Signal	Freq.-Sparse signal
Size	2^{10}	2^{8}	2^{7} to 2^{11}	3^{6}x2^{10}
Word width	16 bits	10 bits	12 bits	12 bits
Area	8.29mm^2	5.1mm^2	1.37mm^2	0.6mm^2
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36-109GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6pJ	0.4-1.6pJ

Conclusion

- 0.75 million-point sparse FFT in 45nm CMOS IBM SOI
- More than 40x better energy efficiency than traditional FFTs
- 88x improvement in run-time over the software implementation

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

ISSCC®

UNIVERSITY OF PENNSYLVANIA