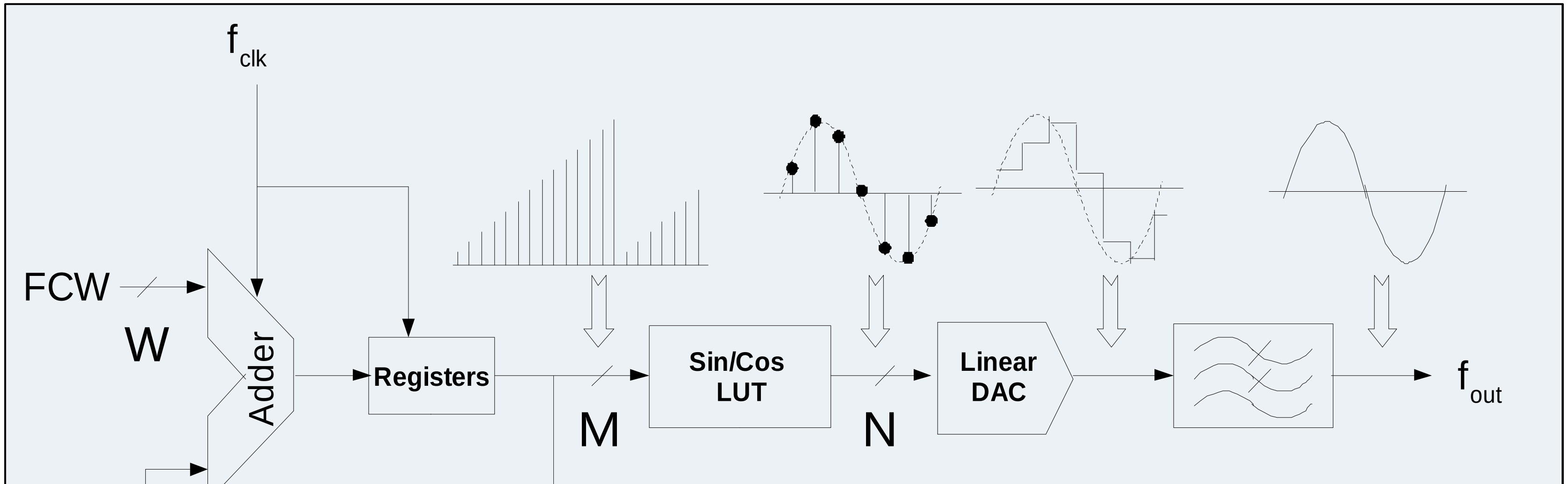
CORDIC-based Numerically Controlled Oscillator (NCO)

Jie Qin, Charles Stroud, Foster Dai
Auburn University

Outline

- Overview of traditional NCO
- Introduction to traditional CORDIC
- Hybrid CORDIC with partial dynamic rotation and LUT
- Experimental Results

Direct Digital Synthesis

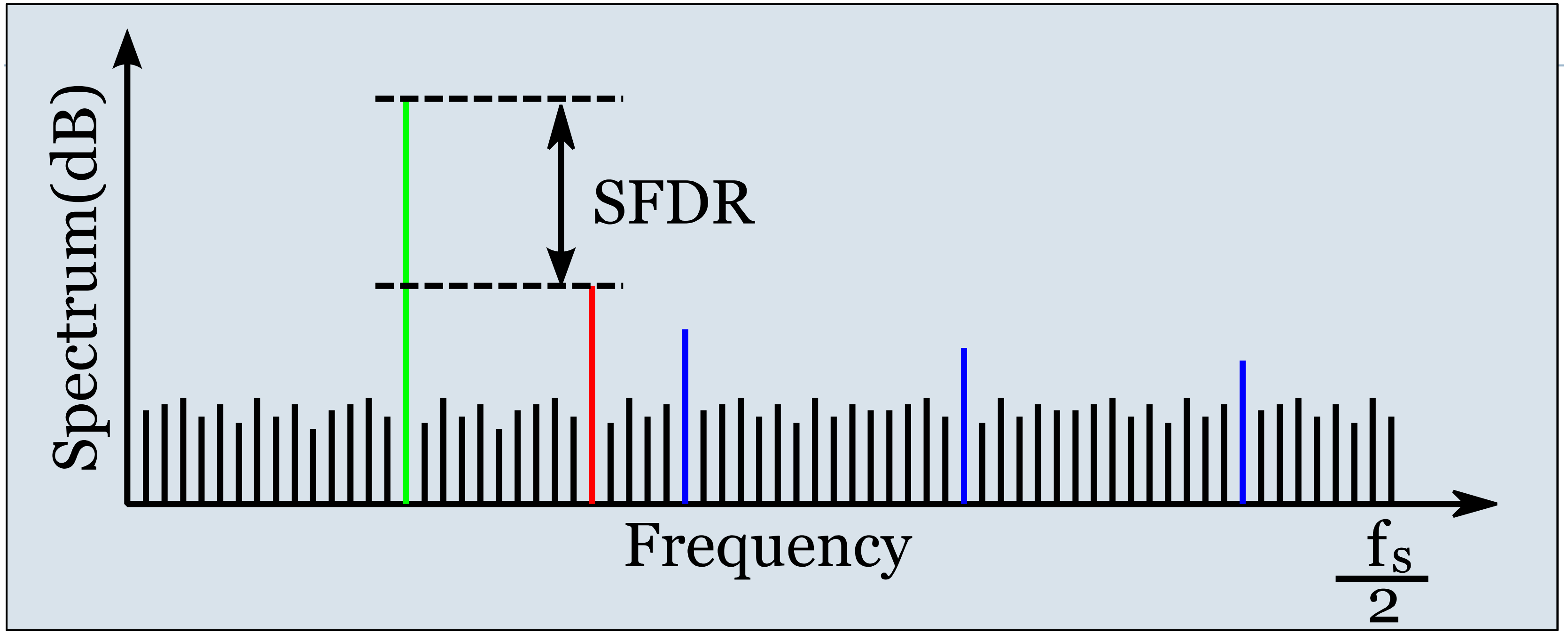


$$f_{out} = f_{clk} \frac{FCW}{2^N}$$

- A complete DDS consists of NCO, DAC and LPF
- NCO[1] Transform the linear phase word into a digital sin/cos word
  - M: bit-width of phase address to LUT
  - N: bit-width of the DAC

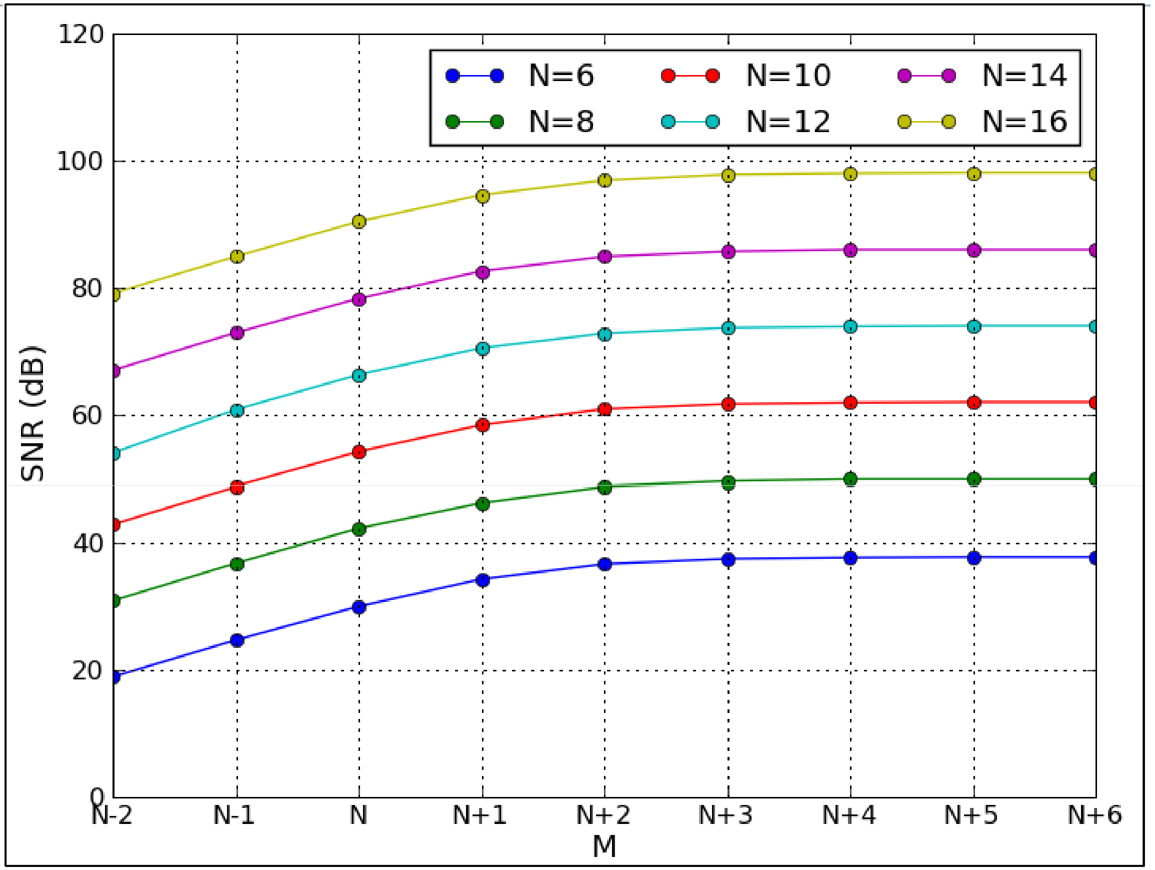
NCO: numerically controlled oscillator

Performance Merits of DDS and NCO



- **Signal-to-noise ratio (SNR):** Ratio between the signal power and noise power over $(0, f_s/2)$ **excluding** spurs
- **Signal-to-noise and distortion ratio (SINAD):** Ratio between the signal power and noise power over $(0, f_s/2)$ **including** spurs
- **Spur-free dynamic range (SFDR):** Ratio between the signal power and the worst spur

SNR of NCO



- M: bit-width of the phase address to LUT
- N: bit-width of the DAC

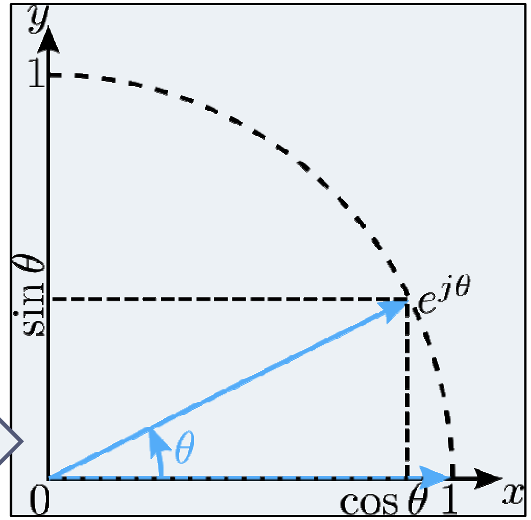
- NCO performance depends on both M and N
  - To fully utilize the dynamic range of the DAC, M > N
  - LUT size increases exponentially as N increases

Introduction of CORDIC

- What is CORDIC?
  - An acronym for **CO**ordinate **R**otation **DI**gital **C**omputer
- What can CORDIC do?
  - Calculate sine, cosine, magnitude, and phase
    - using only LUT, shift and addition/subtraction operations
- How does CORDIC calculate these functions?
  - Through successive vector rotations basically
- Potential Applications in BIST
  - NCO (Numerically Controlled Oscillator)
  - BIST Calculation (square root and arctangent)

View Point of Vector

- Two forms to represent a vector
  - Polar form: $Ae^{j\theta}$
  - Cartesian form:
    $$(a, b) = a + b \cdot j = A cos\theta + j \cdot A sin\theta$$
  - Cartesian forms are used in CORDIC
- NCO: **polar form** to **Cartesian form**
  - Knowing θ, needs $cos\theta + j \cdot sin\theta$
  - Can be obtained by rotating a unit vector $(1, 0)$ by θ
- BIST calculation: **Cartesian from** to **polar form**
  - Knowing $DC_1 + j \cdot DC_2$, needs A and θ
  - Can be obtained by rotating the vector back to x-axis



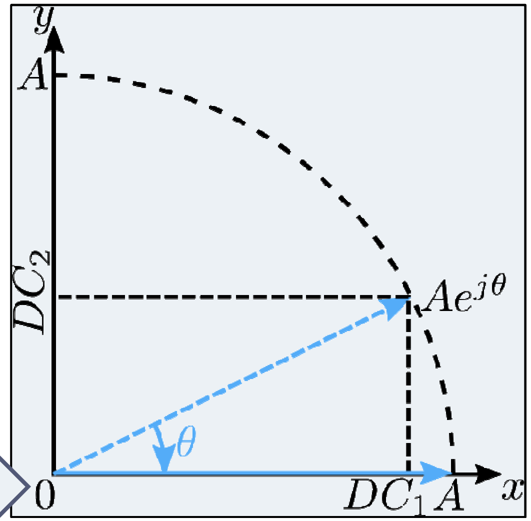
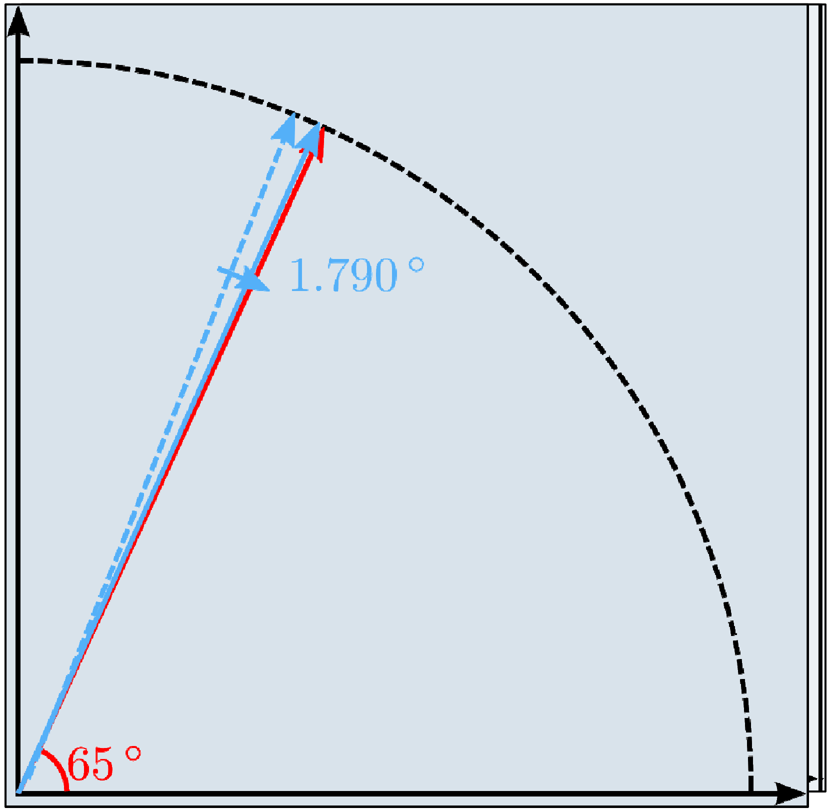


Illustration of Successive Rotation

- Achieve 65° through a series of rotations
  - The phase step θ_i every rotation takes is given that $\tan\theta_i = 2^{-i}$
  - The rotation starts from 0° whose cosine and sine are 1 and 0



Phase	Tangent
45°	1
26.565°	½
14.026°	¼
7.125°	1/8
3.576°	1/16
1.790°	1/32
0.895°	1/64
0.448°	1/128
0.224°	1/256
…	

Accumulative Phase: Φ = [illegible]

How CORDIC Performs Rotation

- Rotating a vector $e^{j\varphi i}$ by θ_i gives

$$\begin{aligned} e^{j(\phi_i+\theta_i)} &= \cos(\phi_i+\theta_i) + j\sin(\phi_i+\theta_i) \\ &= (\cos\phi_i\cos\theta_i - \sin\phi_i\sin\theta_i) + j(\sin\phi_i\cos\theta_i + \cos\phi_i\sin\theta_i) \\ &= \cos\theta_i[(\cos\phi_i - \tan\theta_i\sin\phi_i) + j(\sin\phi_i + \tan\theta_i\cos\phi_i)] \end{aligned}$$

- Rotations of θ_i are purposely chosen that $\tan\theta_i = 2^{-i}$

$$e^{j(\phi+\theta_i)} = \cos\theta_i[(\cos\phi - 2^{-i}\sin\phi) + j(\sin\phi + 2^{-i}\cos\phi)]$$

Scaling factor *K* ($\cos\theta_i$)

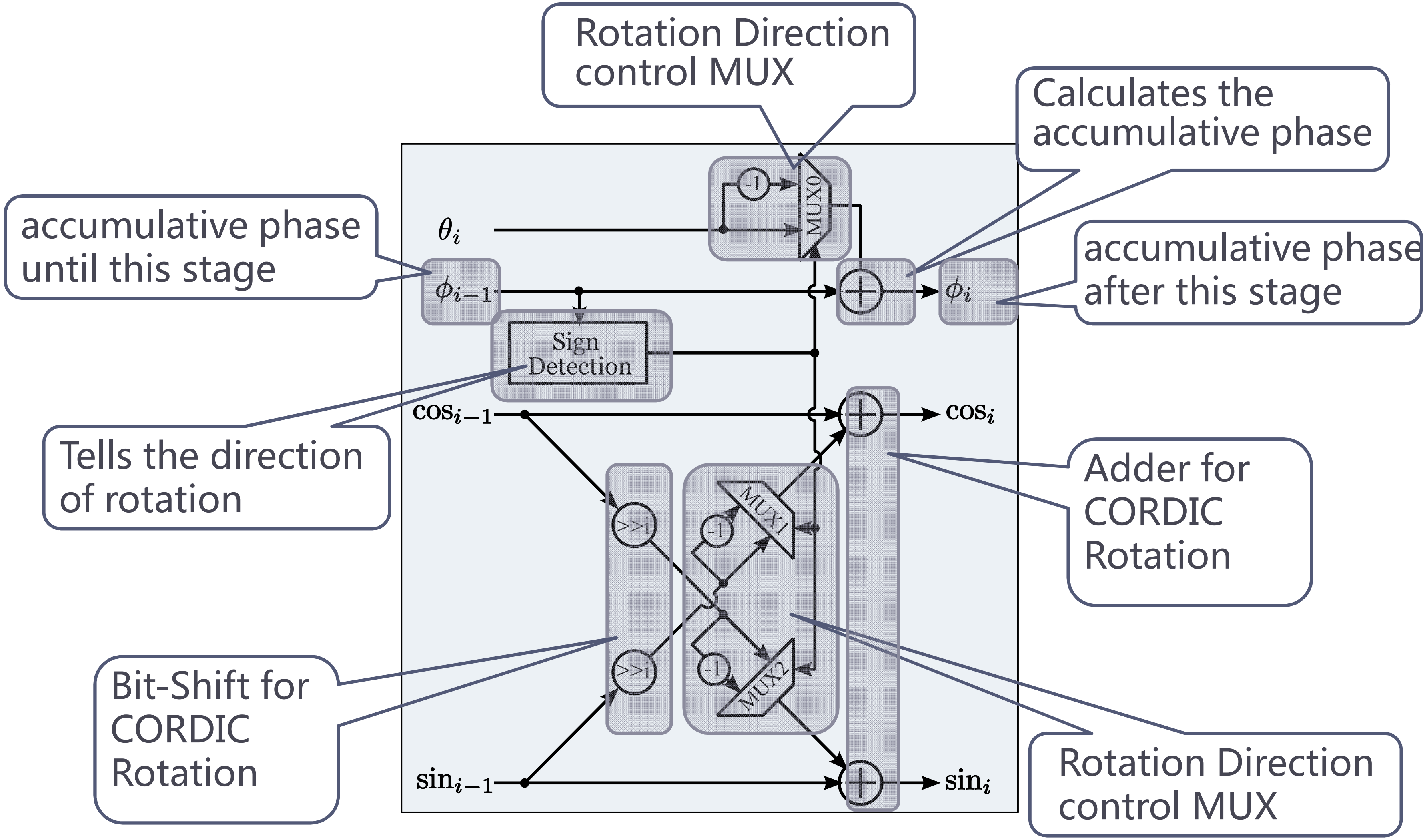
Bit-shift and Subtraction ($\cos\phi - 2^{-i}\sin\phi$)

Bit-shift and Adder ($\sin\phi + 2^{-i}\cos\phi$)

CORDIC Rotation ($[(\cos\phi - 2^{-i}\sin\phi) + j(\sin\phi + 2^{-i}\cos\phi)]$)

- CORDIC algorithm only utilizes CORDIC rotation
  - Scaling factor *K* is discarded, thus $\dfrac{e^{j(\phi_i+\theta_i)}}{\cos\theta_i}$
  - Vector (1, 0), after N rotations, becomes $\dfrac{e^{j(\sum_{i=0}^{N-1}\pm\theta_i)}}{\prod_{i=0}^{N-1}\cos\theta_i}$
  - Not a problem as long as N is same

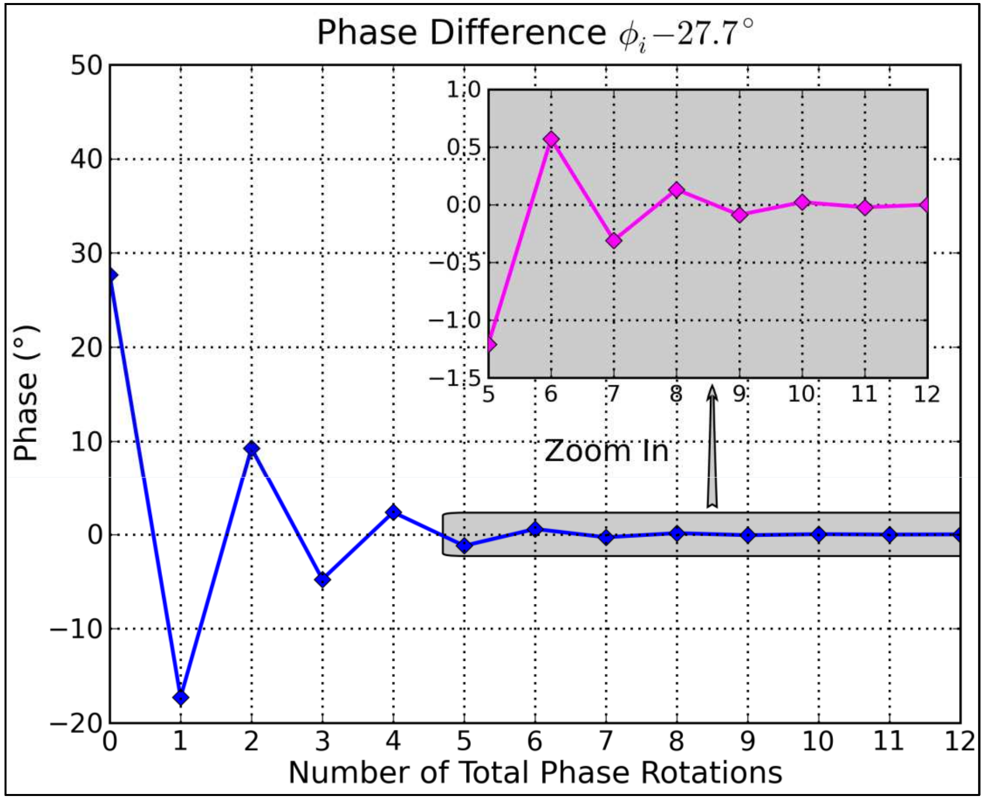
CORDIC Rotation Stage



Pros and Cons of CORDIC

- Pros of CORDIC
  - No cos/sin ROM needed
  - Only a small phase LUT, shifts and adders needed
- Cons of CORDIC
  - A number of rotations required
    - Low speed if the rotation stage is reused
    - Heavily Pipelined design for high-speed requirement
- Two solutions are proposed to reduce # of rotations
  - Partial dynamic rotation (PDR)
  - Hybrid architecture to incorporate LUT and CORDIC rotation

Phase Oscillation in CORDIC Rotation



Desired phase: 27.7°
Different between the accumulative phase and the desired phase *versus* number of total phase rotations

- Phase Oscillation makes slow phase convergence
  - Rotation step is fixed in each stage
  - Dynamic rotation is needed for fast phase convergence
    - Find the optimistic (closest) rotation step on-the-fly

Issues with Dynamic Rotation

- Scaling factor issue for dynamic rotation

$$e^{j(\phi+\theta_i)} = \cos\theta_i \left[(\cos\phi - 2^{-i}\sin\phi) + j(\sin\phi + 2^{-i}\cos\phi) \right]$$

Scaling factor K (from $\cos\theta_i$) · Bit-shift and Subtraction · Bit-shift and Adder · → CORDIC Rotation

- K is ignored to eliminate the needs for multipliers
  - Not a problem for static CORDIC rotation
    - since all θ_i in LUT will be gone through
  - Serious issue for dynamic rotation
    - No constant amplitude for output vectors

$$\frac{e^{j(\sum_{i=0}^{N-1} \pm\theta_i)}}{\prod_{i=0}^{N-1} \cos\theta_i}$$

- Issue of hardware overhead
  - Dynamic rotation selection and programmable shifter required
    - More hardware overhead than static rotation stage

Partial Dynamic Rotation

- Partially Dynamic Rotation (PDR)
  - If θ_i small enough, no scaling factor issue since $\cos\theta_i \approx 1$.
    - Static rotation for large θ_i
    - Only dynamic rotation for small θ_i
  - Speed up the phase convergence
- It is safe to use PDR from 3.576° for a 12-bit NCO

Phase	Tangent	Cosine
45°	1	0.7071
26.565°	½	0.8944
14.026°	¼	0.9701
7.125°	1/8	0.9923
3.576°	1/16	0.9981
1.790°	1/32	0.9995
0.895°	1/64	0.9999
0.448°	1/128	≈1
0.224°	1/256	≈1
	…	

Hybrid Structure

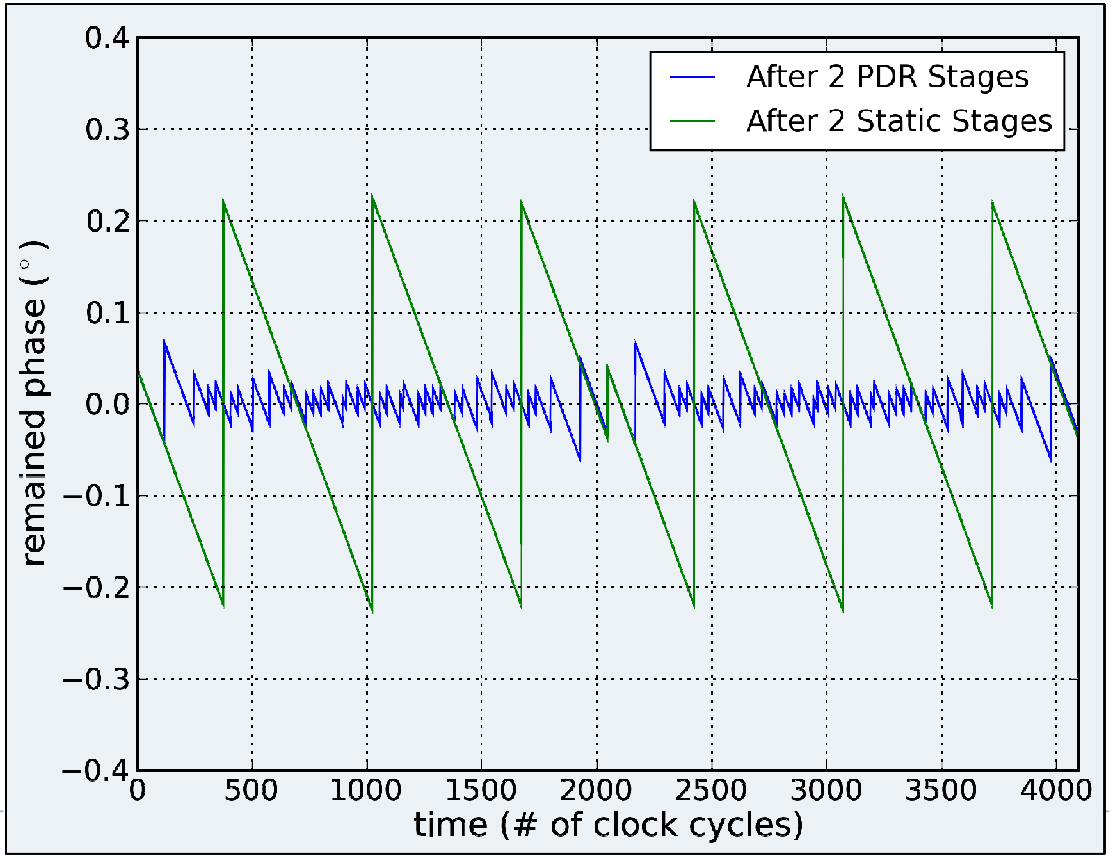
(M[1], N[2])	Hardware Resource	LUT	Static CORDIC	PDR CORDIC
9 and 8	# of 4-input LUTs	142	314	337
	# of DFFs	0	318	228
11 and 10	# of 4-input LUTs	508	448	464
	# of DFFs	0	451	307
13 and 12	# of 4-input LUTs	1534	590	578
	# of DFFs	0	598	466

- LUT is much more efficient when N is small
  - LUT and PDR are combined to achieve the best result
- It is hard to synthesize a LUT with wide address bus

1. M: bit-width of the phase address to LUT
2. N: bit-width of the DAC

Pros of Dynamic Rotation

- Converge faster, thus less # of rotations required
- Natural dithering effect in phase domain
  - Thus clean spectrum



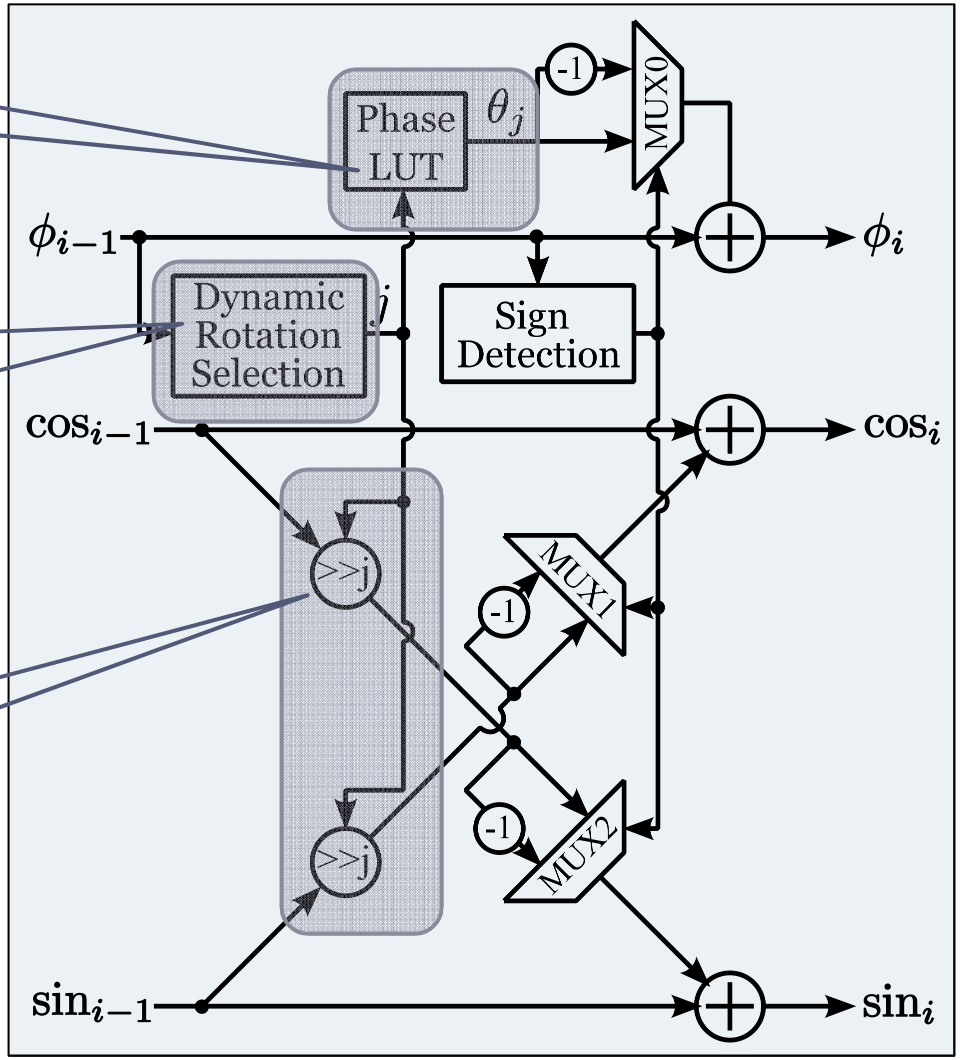
FCW=8193

CORDIC Dynamic Rotation Stage

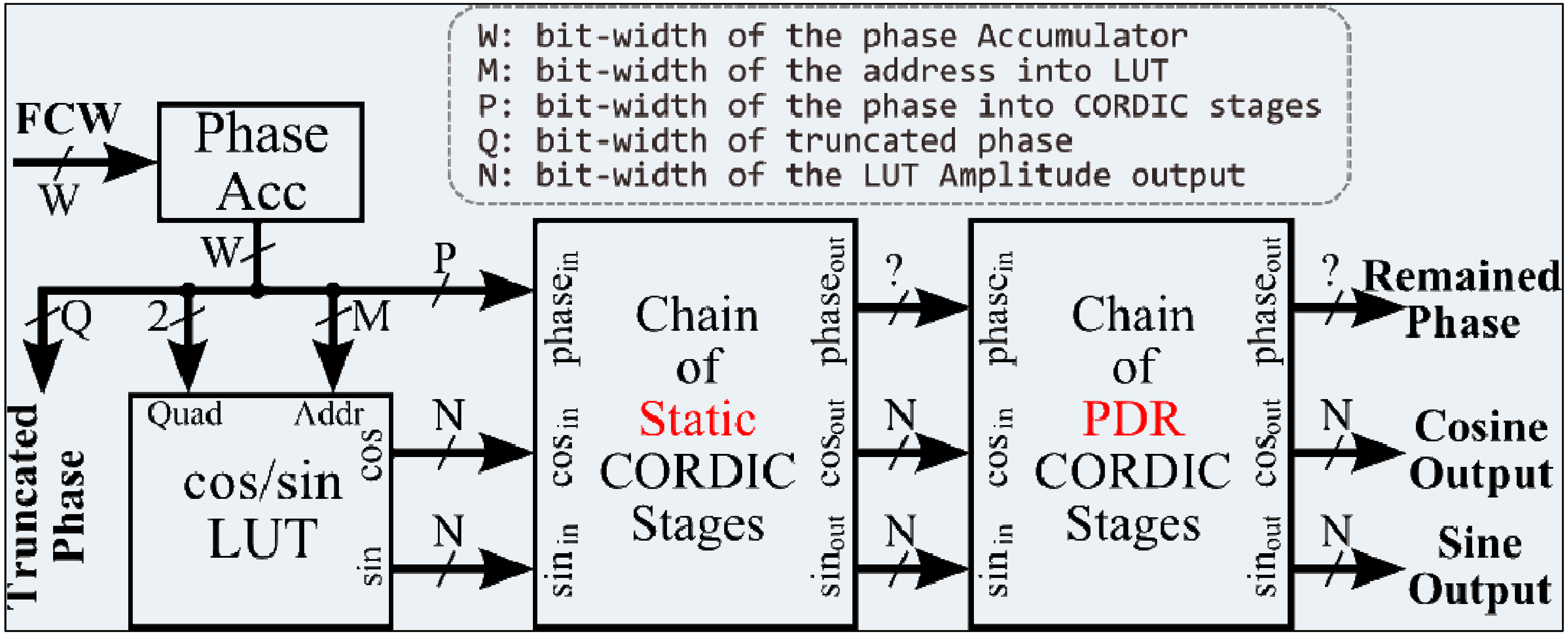
Picks the rotation step dynamically

Determines a phase index for phase LUT and bit-shifter

Programmable Bit-shifter

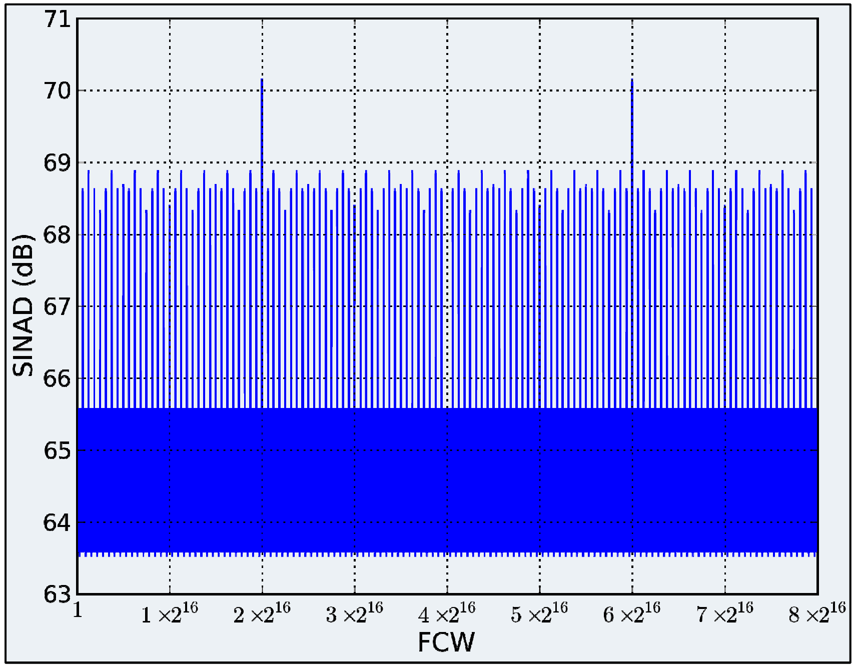


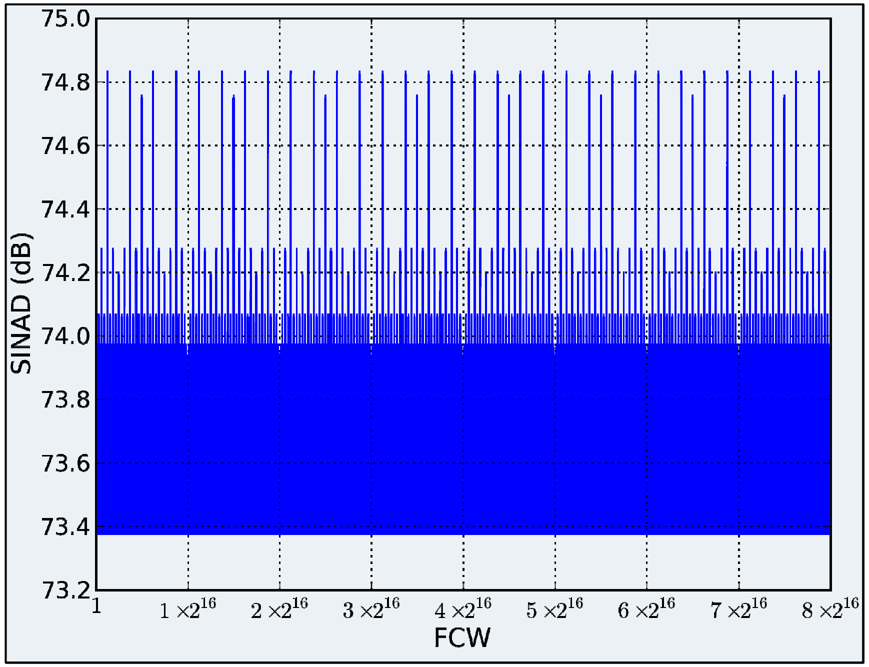
Architecture of PDR-CORDIC



	W	M	P	Q	N	# of Static Stages	# of PDR Stages	RC Synthesized Area (um^2)
CORDIC_ATO (without Σ-Δ)	32	9	8	13	15	0	2	28,203+52,423
CORDIC_DTO	32	6	9	15	12	0	2	15,767+24,010
BTM (Two NCOs)	32							26,231+49,621

SINAD vs. FCW

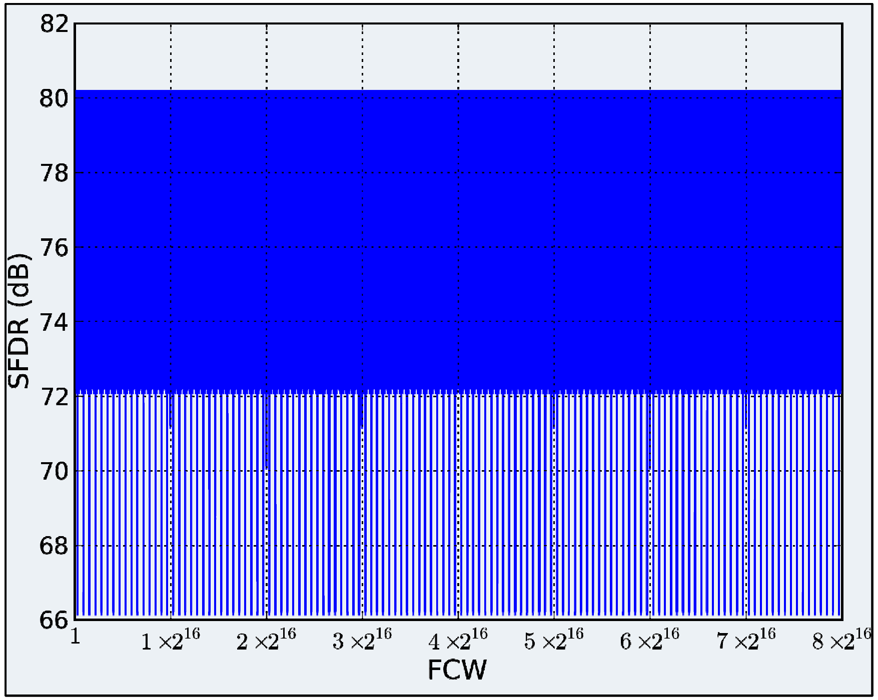


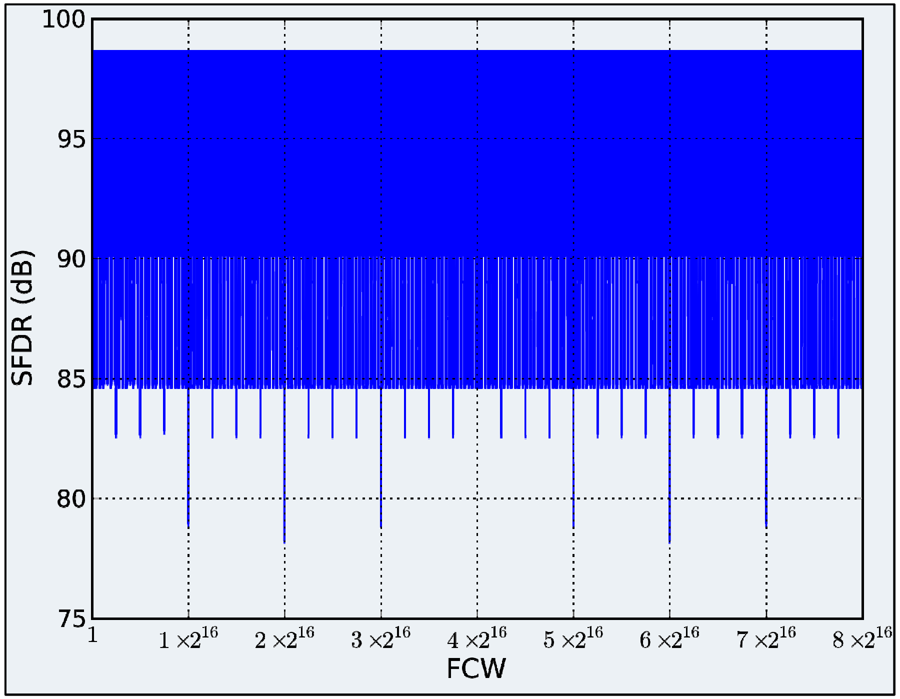


- CORDIC for DTO
  - Worst SINAD
    - About 63.5dB

- CORDIC for ATO
  - Worst SINAD
    - About 73.4dB

SFDR vs. FCW

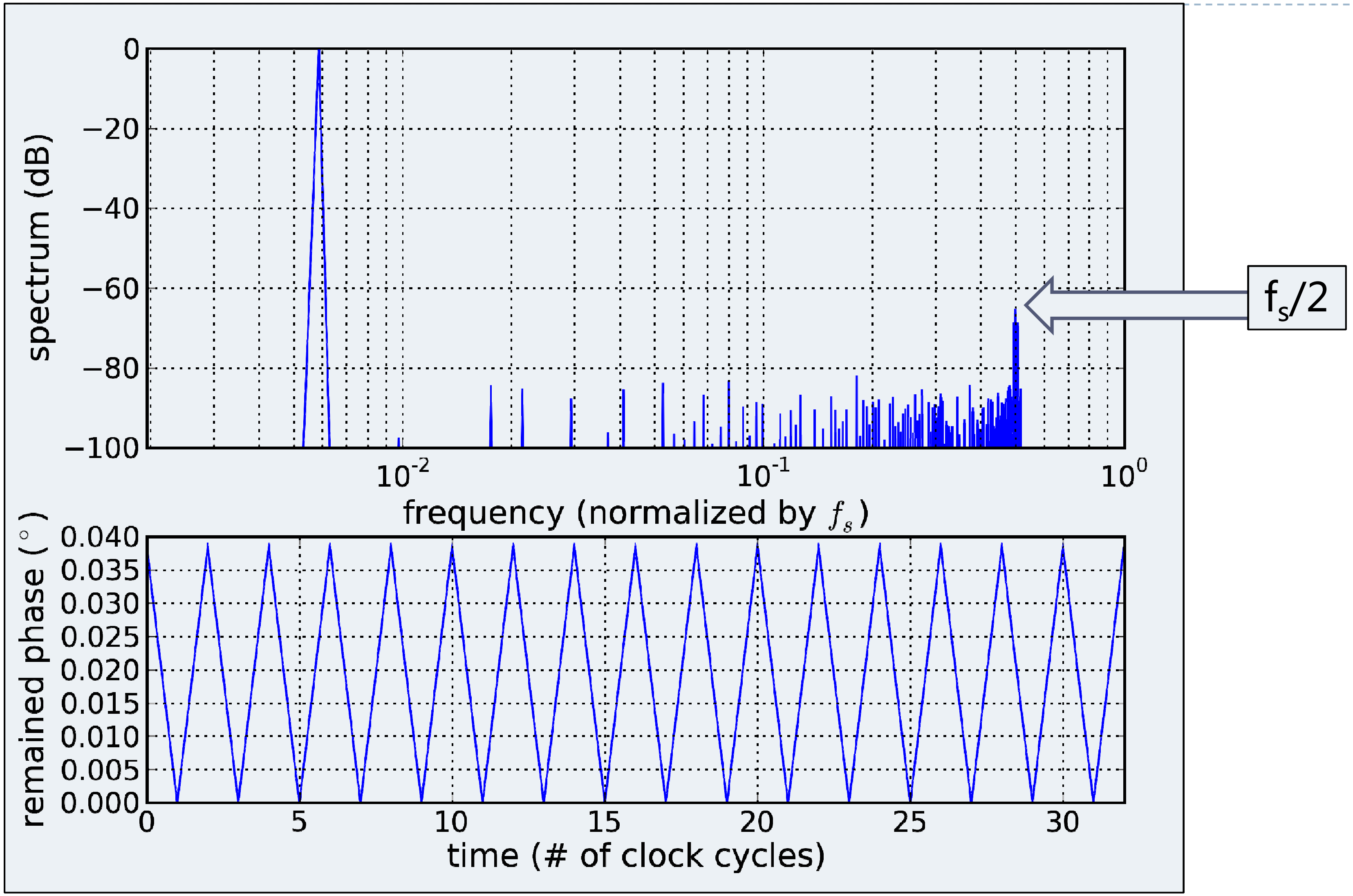




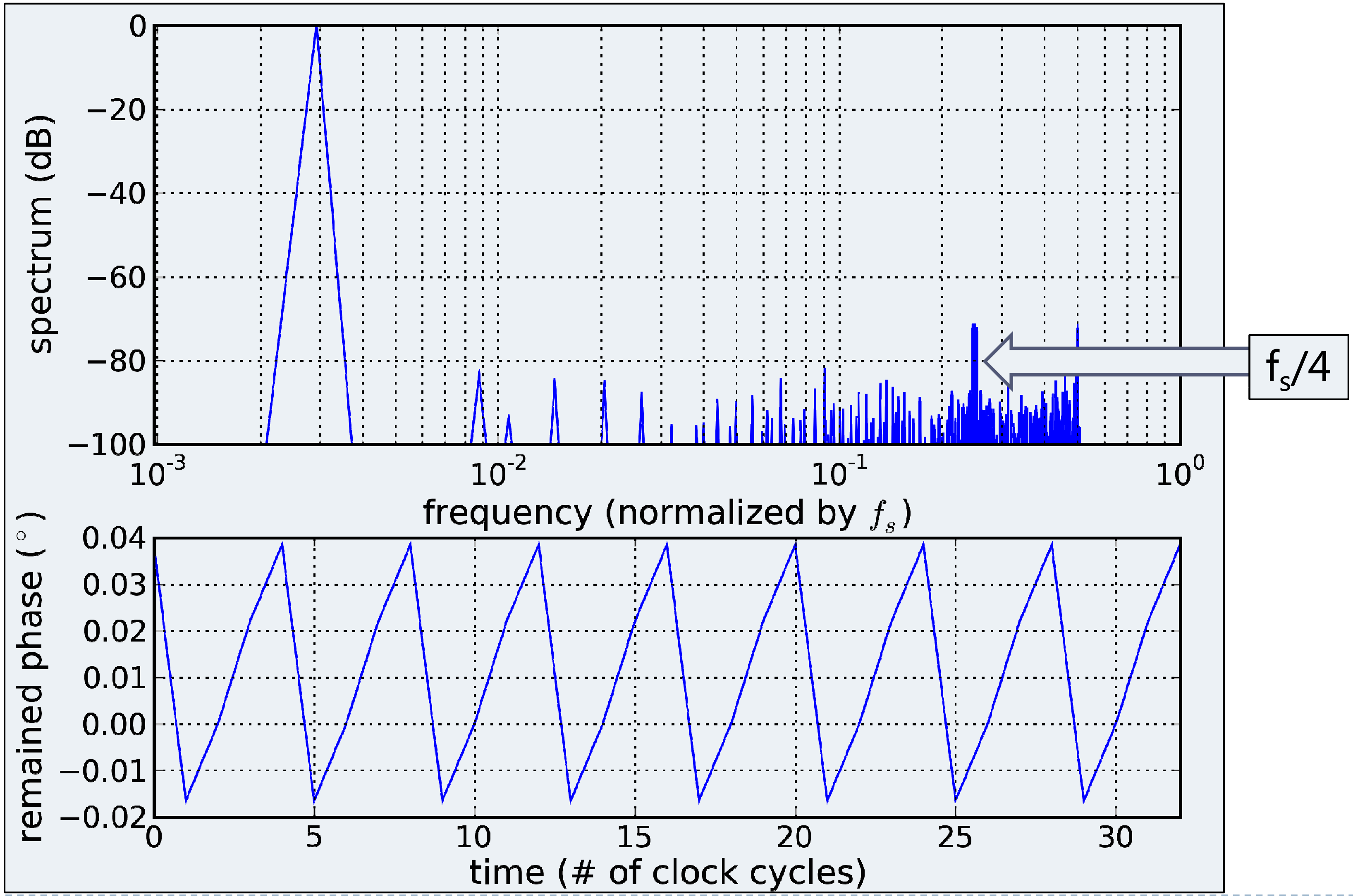
- CORDIC for DTO
  - Worst SFDR
    - About 66dB

- CORDIC for ATO
  - Worst SFDR
    - About 78dB

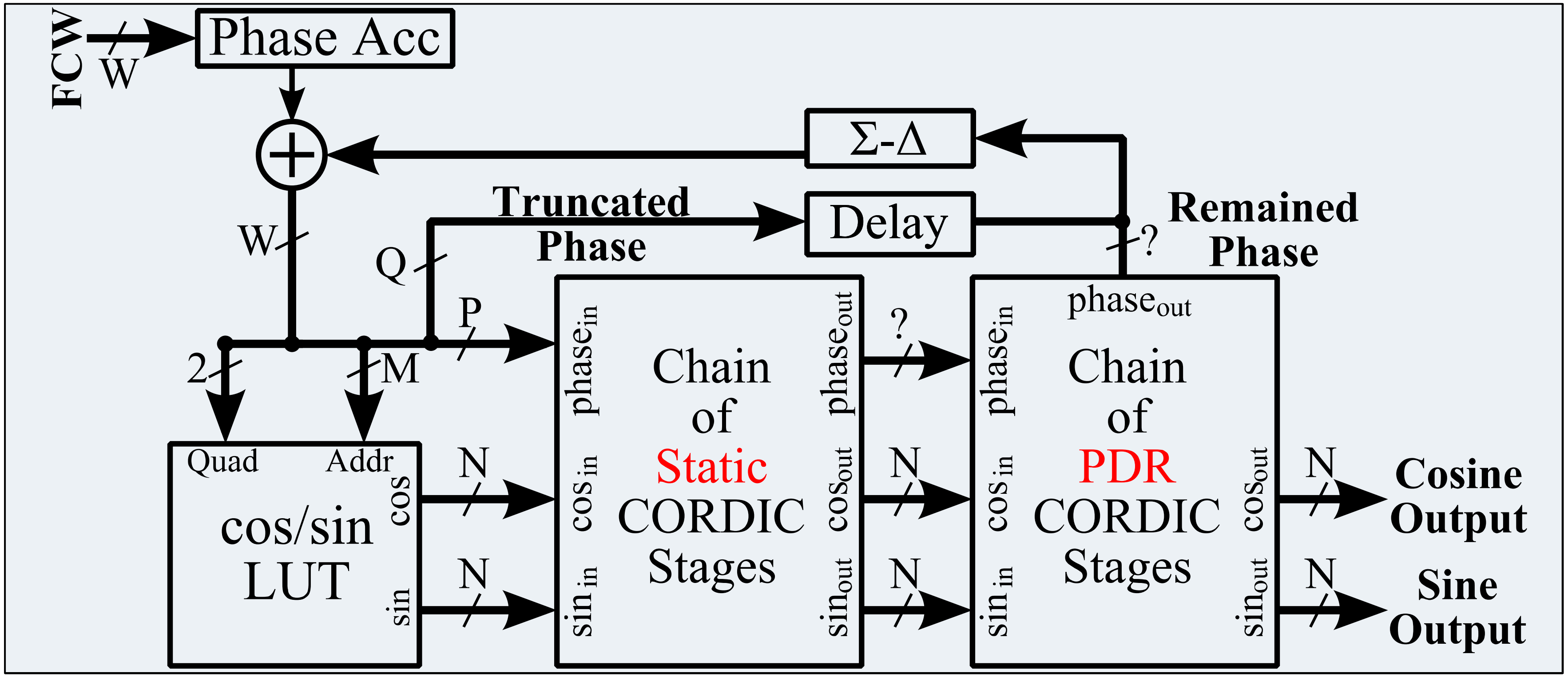
Worst-Case SFDR



2^{nd} Worst-Case SFDR

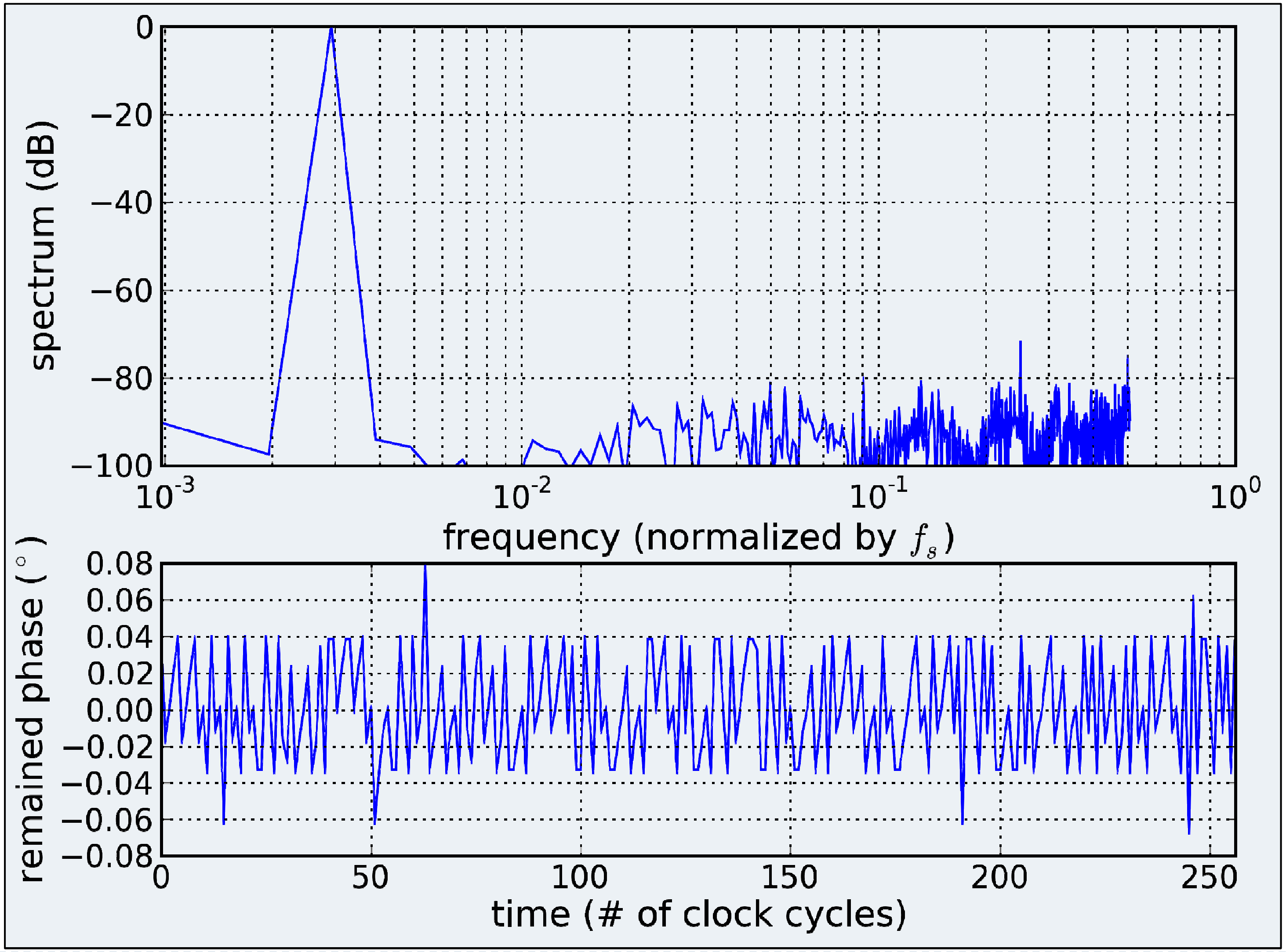


Architecture of PDR-CORDIC with Σ-Δ



- Σ-Δ is adopted to further randomize the phase residue
  - For better spectrum performance

2nd Worst-Case SFDR after ΣΔ



Conclusion

- Hybrid CORDIC with PDR and LUT is a very strong candidate for implementing high speed and high-resolution NCO
  - Much faster convergence speed than traditional CORDIC
  - Less area overhead than traditional CORDIC
    - Comparable to BTM ROM compression technique
  - Quiet Spectrum