Lecture 2. Analog Abstraction

Jaeha Kim
Mixed-Signal IC and System Group (MICS)
Seoul National University
jaeha@ieee.org

Key to Effective Design: “Abstraction”

- Note that digital tools leverage “abstractions” effectively
  - Digital abstraction: Boolean (value), synchronous (time)
  - Tools that simulate abstraction models, convert between abstraction layers, check equivalence, measure coverage, …

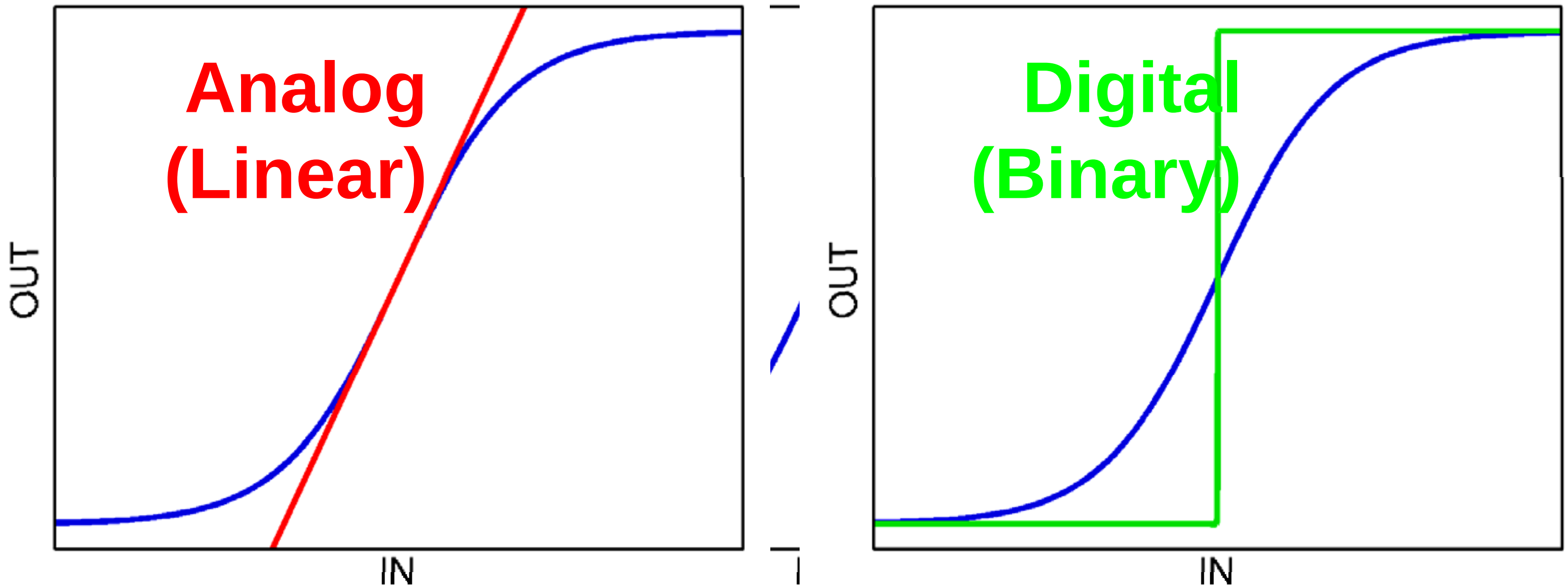
So, analog tools must leverage abstractions – but what is the proper abstraction for analog?

  - No notion on analog abstraction; SPICE treats any circuit as a general nonlinear system
  - Designers scream for *faster* SPICE; although it’s not the solution

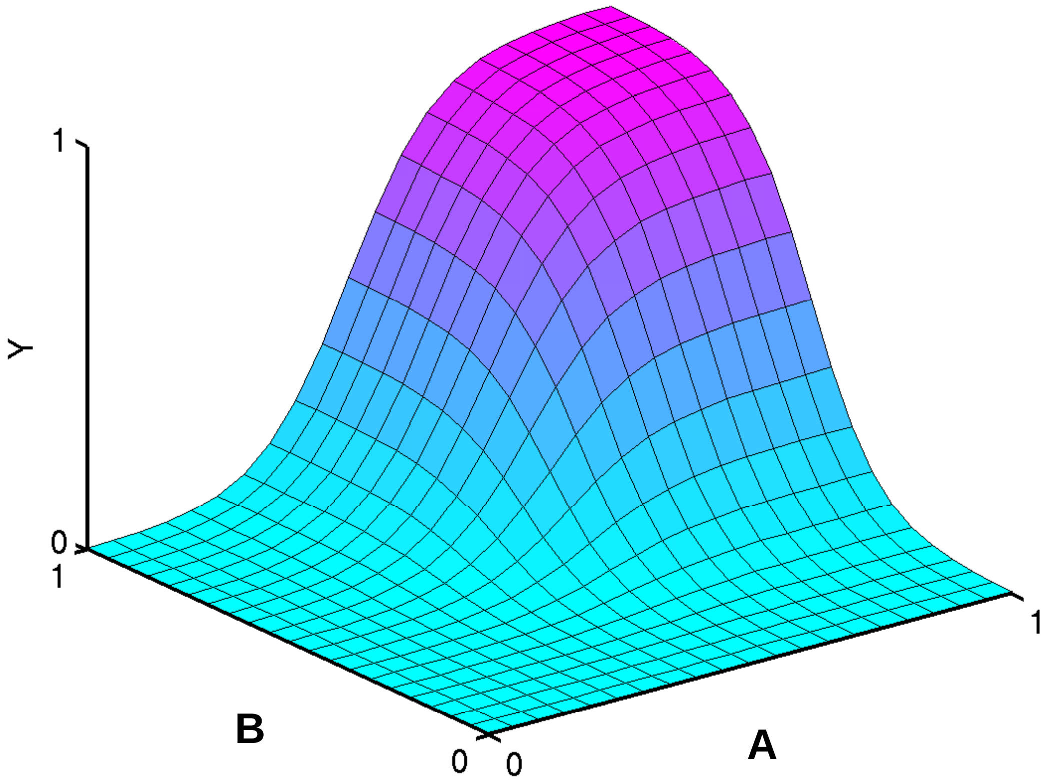
Analog vs. Digital

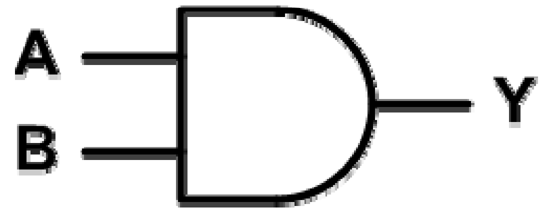
- Continuous vs. discrete?
- A and D are different in their world views

What do you see in this picture?



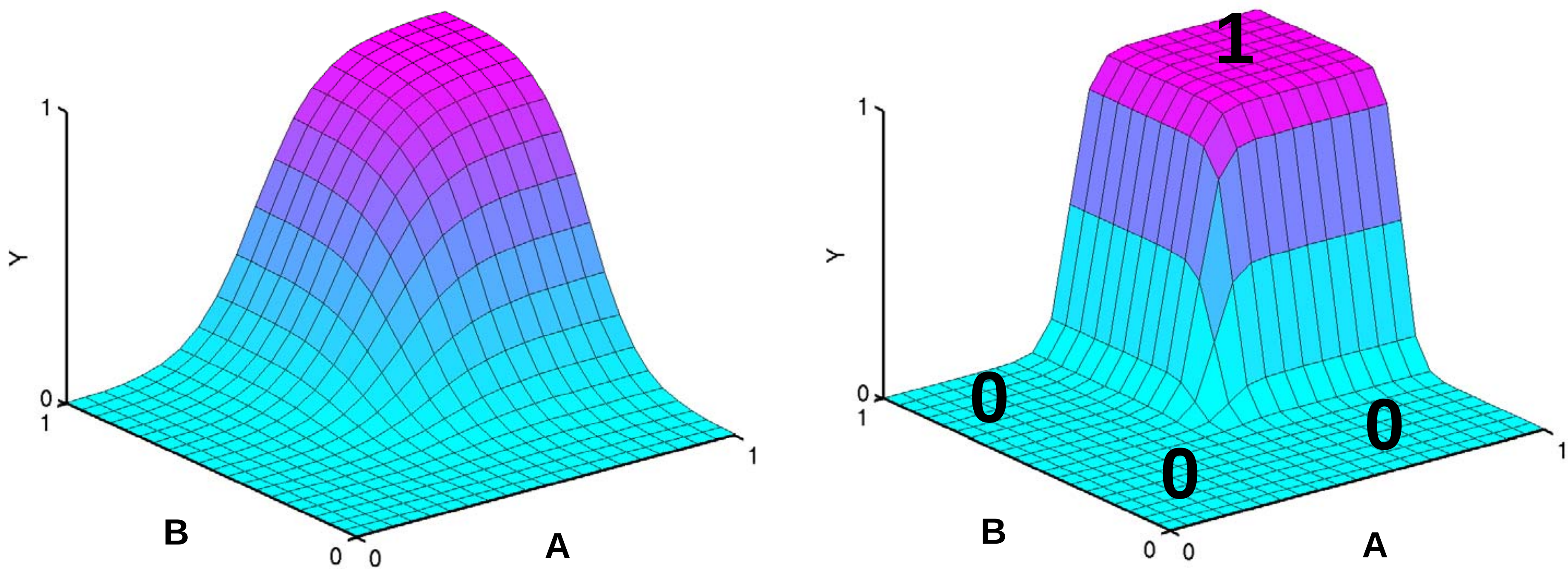
Let's Learn from the Digital World





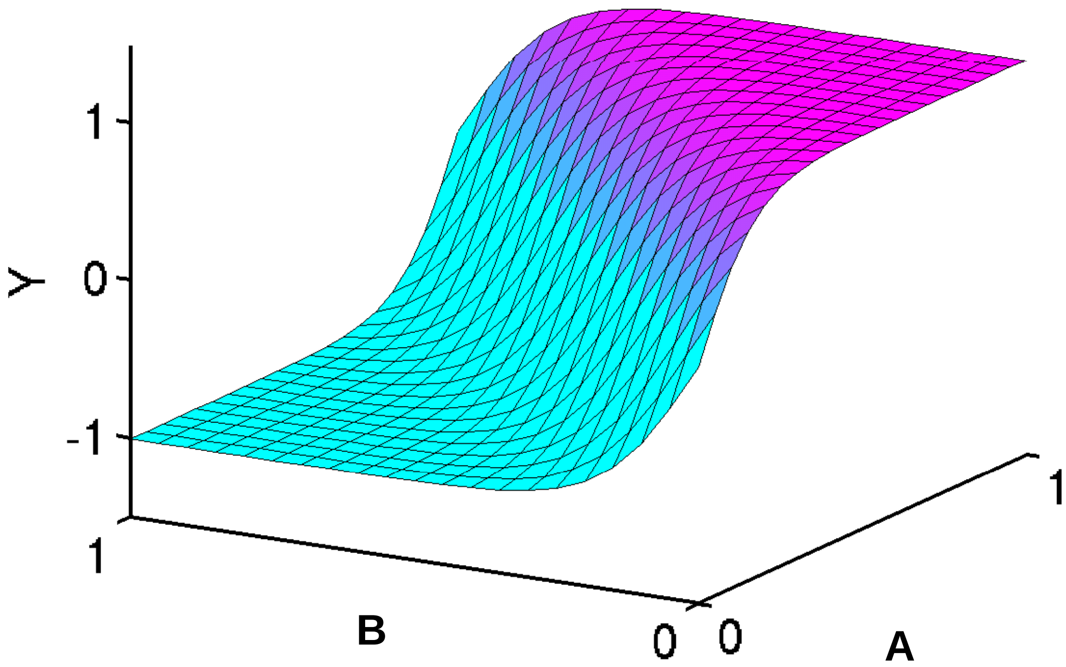
Q: Verify this two-input AND gate

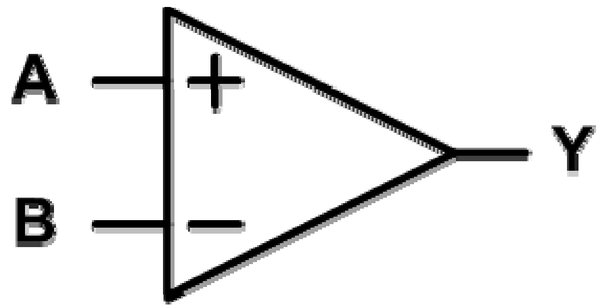
Digital Abstraction is the Key



- States become discrete and countable
- Verify property (Y=A·B) for each state

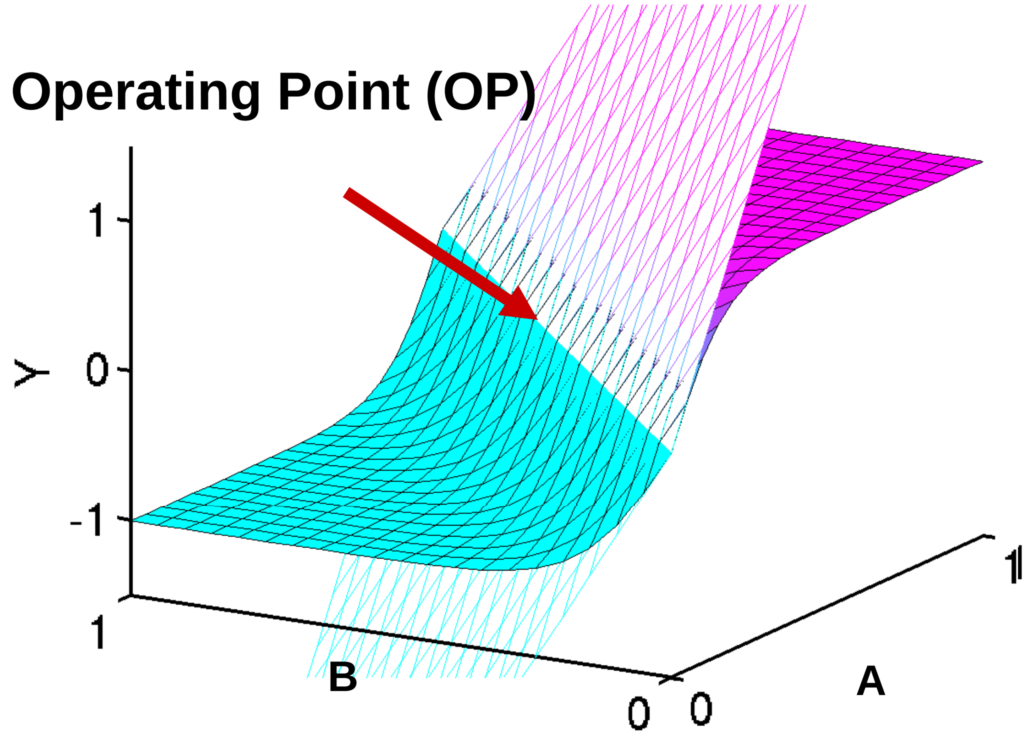
Verification in Analog Circuits





Q: Verify this op-amp

Analog Abstraction: Linear System

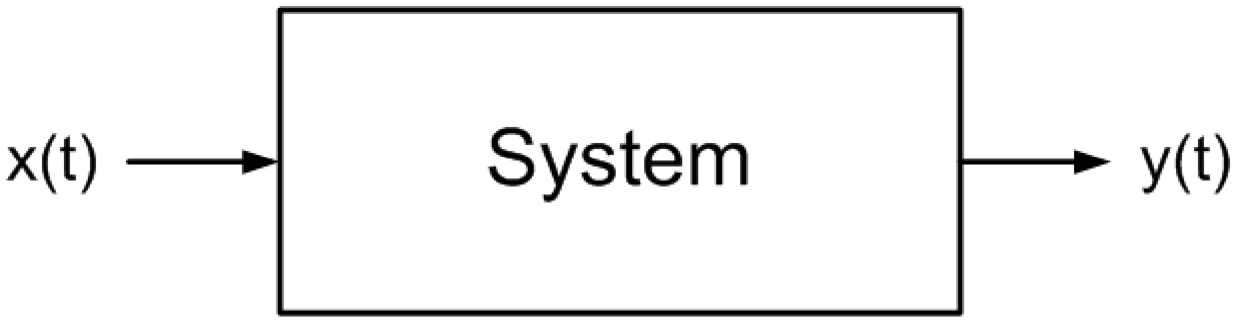


- Design intent is to use the linear region around the OP
- An ideal circuit has linear I/O relationship $\Delta Y = \alpha \cdot \Delta A + \beta \cdot \Delta B$
- In general, it's a linear dynamical system

- Our conjecture: all analog circuits have *linear intent*!
  - Then, the proper abstraction for analog is a linear system

Linear System Review

- What defines a linear system?



  - If $y_1(t)$ is the system's response to an input $x_1(t)$ (i.e. $x_1(t) \rightarrow y_1(t)$), then $a \cdot x_1(t) \rightarrow a \cdot y_1(t)$
  - If $x_1(t) \rightarrow y_1(t)$ and $x_2(t) \rightarrow y_2(t)$, then $x_1(t)+x_2(t) \rightarrow y_1(t)+ y_2(t)$
  - Called ***"superposition principle"***

Linear System Review (2)

- Superposition principle basically says the output of a linear system can be expressed as ***a linear sum*** of the inputs at different times:

$$y(t) = \int_{-\infty}^{+\infty} h(t, \tau) \cdot x(\tau) d\tau$$

  - $h(t, \tau)$ denotes the gain between input $x(\tau)$ and output $y(t)$
  - Causality implies: $h(t, \tau) = 0$ if $t < \tau$ if t
  - Time-invariance implies: $h(t, \tau) = h(t-\tau)$, i.e., the gain is a function of $t-\tau$ only
  - $h(t-\tau)$ is called the ***impulse response*** of a system

Leverage Linear Abstraction

- As Boolean abstraction did for digital, linear abstraction can greatly simplify analog verification

- For example, abstraction lets us define “coverage”
  - How thoroughly do we need to run the simulations
  - Boolean reduces # of points along each state dimension (2)
  - Linear allows each input to be tested independently

$$y = \sum_i \alpha_i \cdot x_i \qquad \textbf{(superposition)}$$

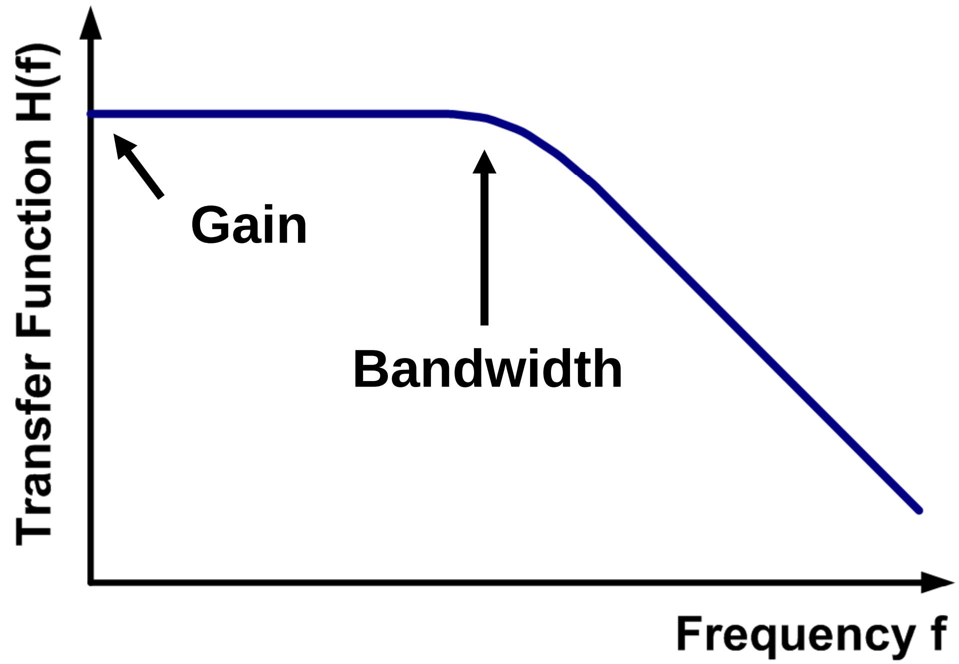
  - # of tests required is actually far less for analog (N+1 vs. 2^N)

* J. Kim, et al., “Leveraging Designer’s Intent: A Path Toward Simpler Analog CAD Tools,” CICC’09.

Linear AC: Formal Method for Analog

- If the intent is linear, AC analysis is the most effective way of validating it
  - AC analysis can measure the transfer function (TF) of a circuit
  - TF is a Fourier transform of the impulse response; therefore it can ***completely describe*** the linear system of interest

- Therefore, AC analysis *formally* verifies the linear intent



Is LS Indeed the Abstraction for Analog?

- Note that not everyone agrees with my conjecture
- Designers easily agree because:
  - Linear system theory is the only tool they are trained with in order to understand the real world
  - “Engineers see the world as a first-order system; although it’s really a second-order one”
  - Designers don’t know how to analyze general nonlinear systems: how can you design a system you don’t understand?

Is LS Indeed the Abstraction for Analog?

- CAD researchers don’t agree easily
  - They are people who write numerical simulators like SPICE
  - They are very good with nonlinear equations; in fact linear equations are too elementary to them
- They say, “no real circuits are linear!”
  - And my designer friend said nonlinearity is really important!
  - True, but no real gates are Boolean, either
  - That does not mean the circuit doesn’t have a linear intent
  - Question is, can we describe the circuit as
    - an approximate linear system and
    - its deviations from that linear system?

Weakly Nonlinear Systems

- A system whose behavior can be approximated mostly as linear system yet may possess mild nonlinearities are called "weakly nonlinear systems"
  - Can be expressed with Volterra series (~ Taylor series for dynamical systems)

$$
\begin{aligned}
y(t) = y_0 &+ \int_{-\infty}^{+\infty} h_1(\tau_1) \cdot x(t-\tau_1) d\tau_1 \\
&+ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 \\
&+ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_3(\tau_1, \tau_2, \tau_3) x(t-\tau_1) x(t-\tau_2) x(t-\tau_3) d\tau_1 d\tau_2 d\tau_3 \\
&+ \cdots
\end{aligned}
$$

  - Don't be scared by the equation; point is that there exists a systematic way of analyzing a weakly nonlinear system

Example: Low-Noise Amplifier (LNA)

- Its specification parameters can be categorized into:
- Linear parameters:
  - Gain (S_{21})
  - Center frequency
  - Noise figure
- Nonlinear parameters
  - 1-dB compression point: gain saturation
  - 3rd-order interception point (IP3): harmonic generation
- Others
  - Power, area, supply voltage, technology node, …

Note: most circuits strive to achieve zero nonlinearities!

What about ADC and DAC?

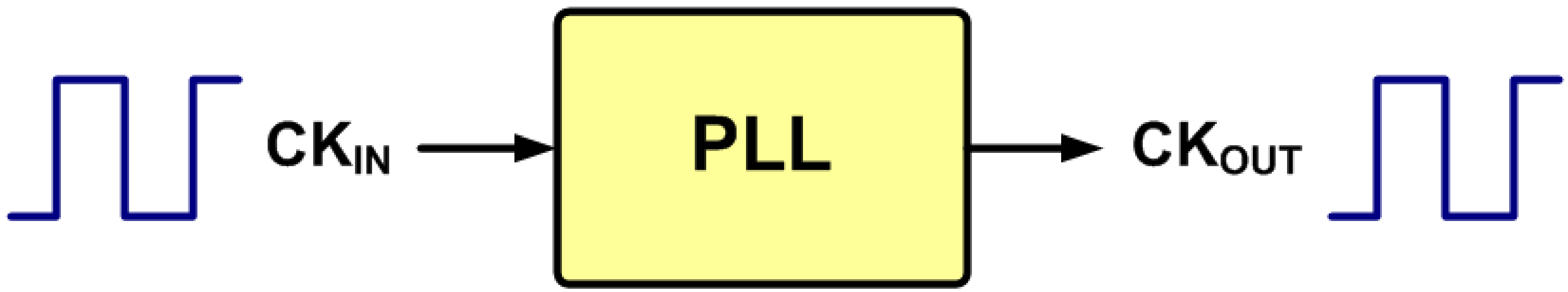
- A/D and D/A converters are one exception to our conjecture – they possess “strong nonlinearity”
- And it is an intended behavior; even an ideal ADC should quantize the input signal!

- Nonlinearity in a DC sense is easy to describe and understand
  - An extended framework to include strong DC-nonlinearities within the realm of “weakly nonlinear system” is called “modified Volterra Series”

D. Mirri, “A modified Volterra series approach for nonlinear dynamic systems modeling”, TCAS-I, Mar 2006.

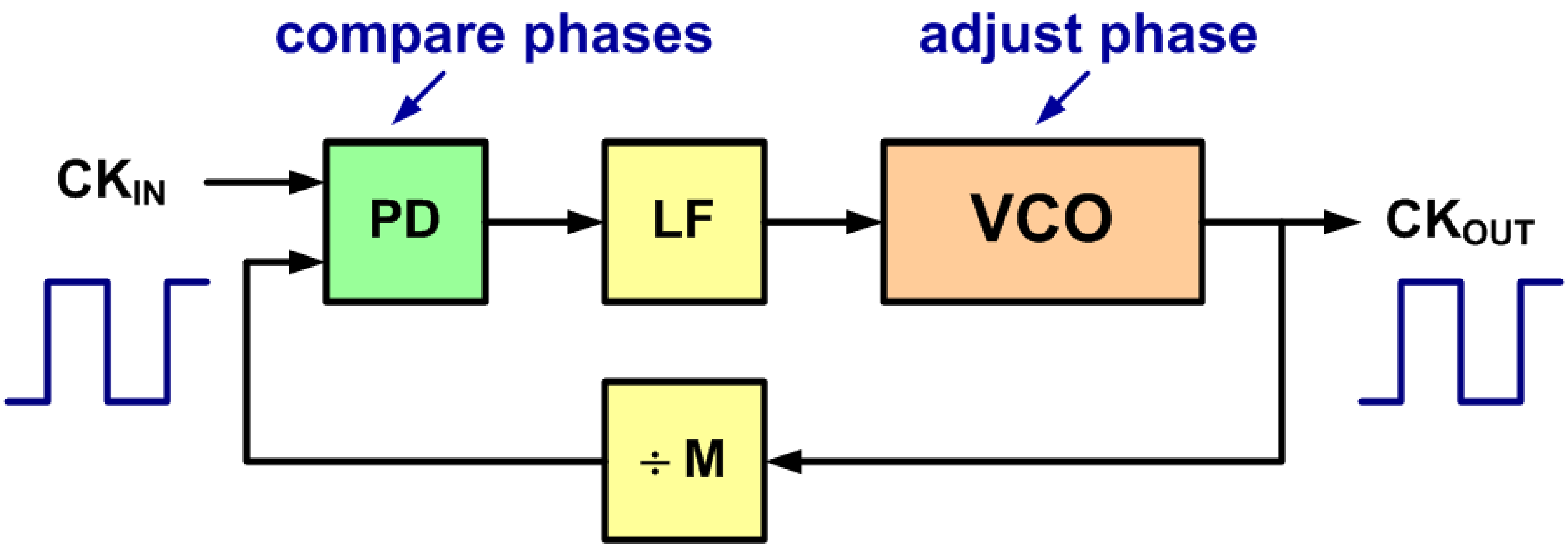
What About PLL/DLL?

- A PLL/DLL is highly nonlinear from a voltage perspective
  - Large-signal clock in, large-signal clock out
  - Isn’t this a counter-example to my conjecture?



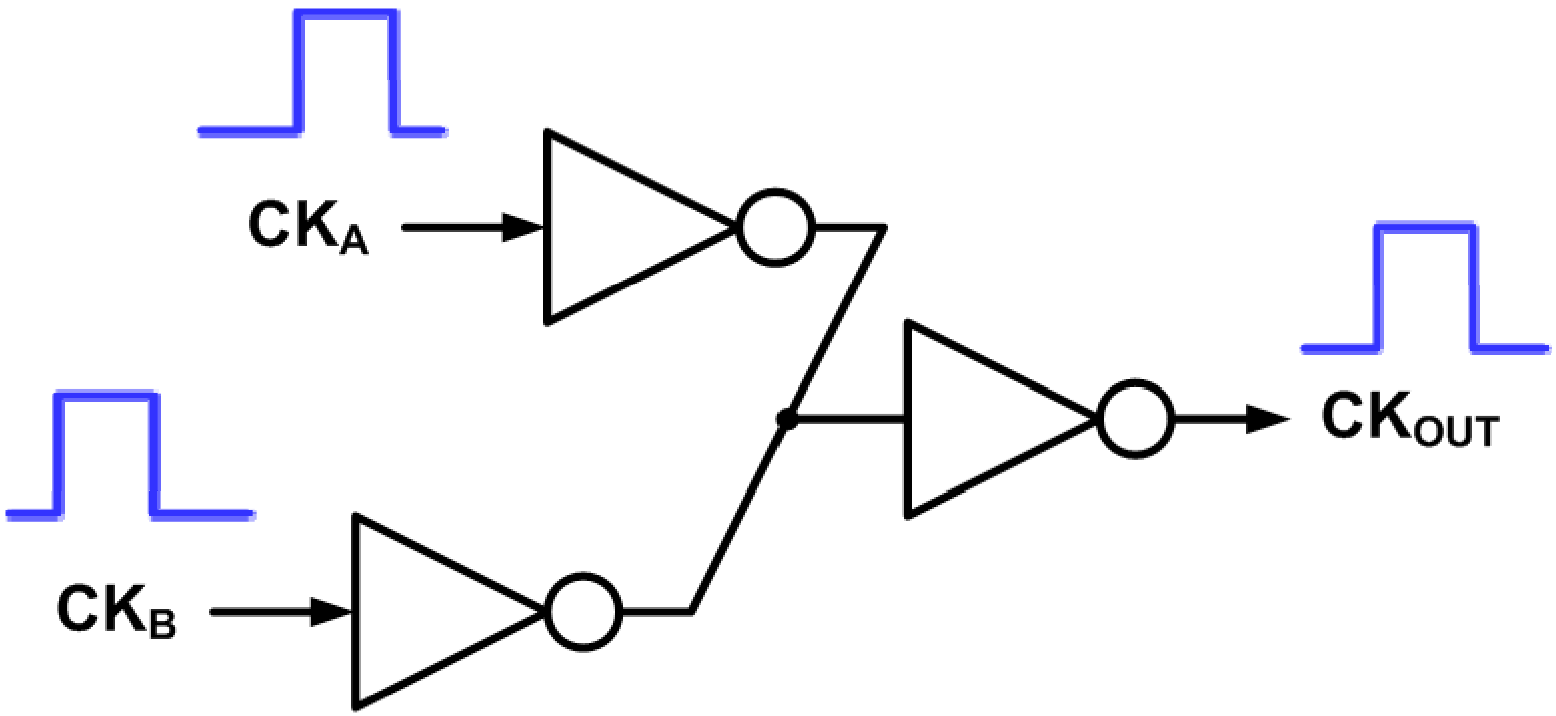
PLL/DLL is a Linear System

- A PLL/DLL is highly nonlinear from a voltage perspective
  - Large-signal clock in, large-signal clock out
- But it is linear in its phase/delay variables
  - The design intent is to control the “phase or delay” not voltage



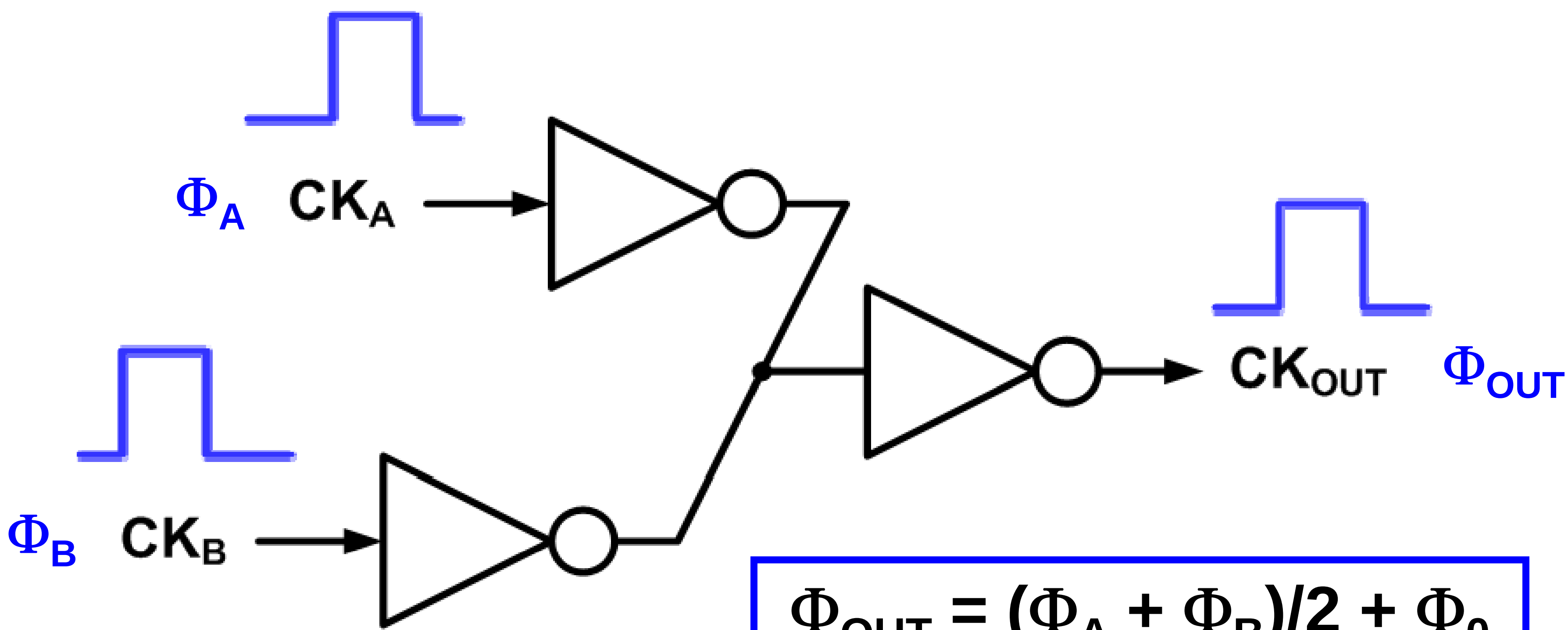
What About Phase Interpolators?

- A phase interpolator takes two clock inputs and produces a clock with the phase in-between
  - Is this circuit analog or digital? Is it linear?



Phase Interpolators are Linear

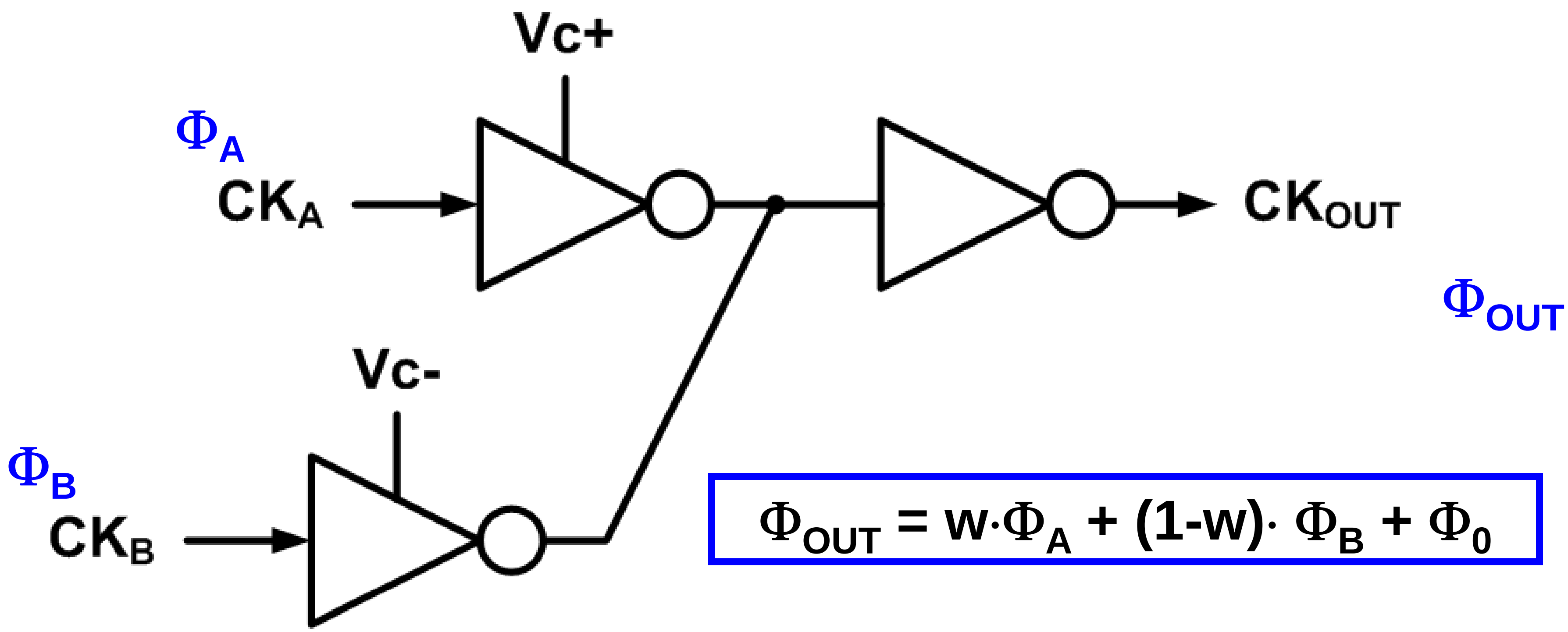
- It has a linear relationship between the input phases and the output phase



$$\boxed{\Phi_{OUT} = (\Phi_A + \Phi_B)/2 + \Phi_0}$$

Phase Interpolator with Control Inputs

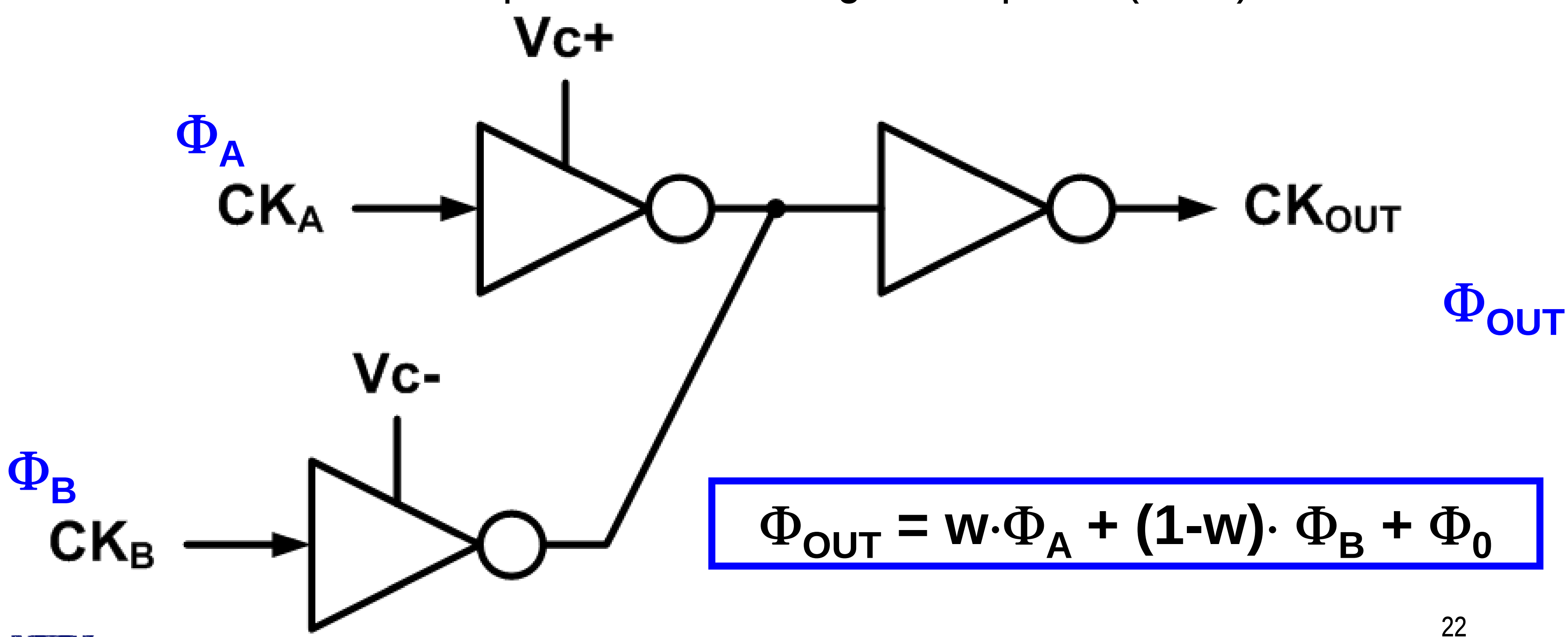
- Some phase interpolators have control inputs which can adjust the interpolation weight *w*
- Q: is this circuit still linear?



$$\boxed{\Phi_{OUT} = w \cdot \Phi_A + (1-w) \cdot \Phi_B + \Phi_0}$$

Circuits with Control Inputs

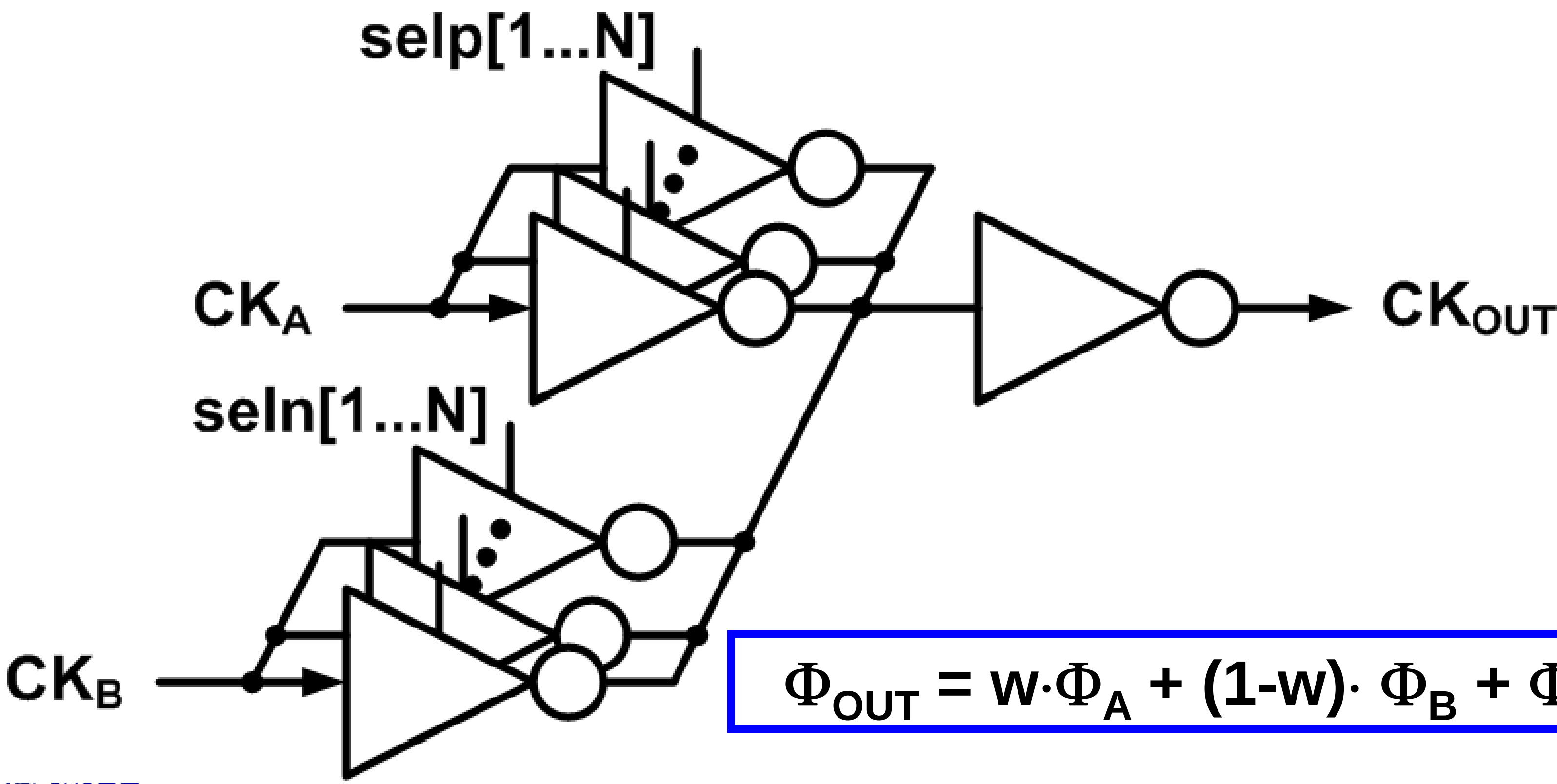
- We can understand these circuits as ***two linear systems***
  - One between the main inputs (Φ_A, Φ_B) and output (Φ_{OUT})
  - The other between the control inputs (Vc+/-) and the weight (w)
  - A similar example is a variable gain amplifier (VGA)



$$\boxed{\Phi_{OUT} = w\cdot\Phi_A + (1-w)\cdot\,\Phi_B + \Phi_0}$$

Phase Interpolator with Digital Control

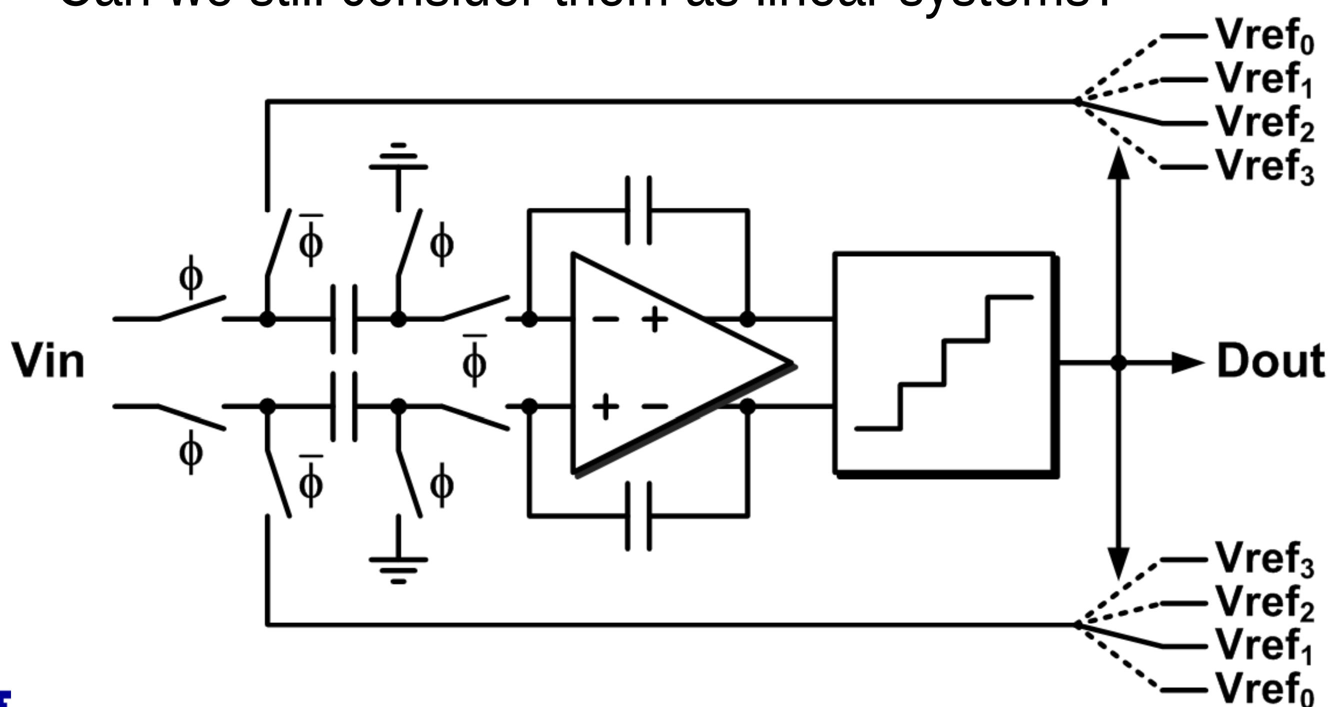
- Now, is this circuit still linear?
  - Functionally yes, except that we have an implicit DAC between the digital control inputs and the interpolation weight



$$\boxed{\Phi_{OUT} = w\cdot\Phi_A + (1-w)\cdot \Phi_B + \Phi_0}$$

What About a Delta-Sigma ADC?

- A Δ-Σ ADC or a bangbang PLL has a strongly nonlinear element (i.e. the quantizer or binary PD) within the loop
  - Makes the loop behavior strongly nonlinear, too
  - Can we still consider them as linear systems?



Their Behaviors are Random

- Due to nonlinear (digital) components in the feedback
  - e.g. binary PLL/CDRs, digital calibration loops, etc.
  - Aperiodic dithering near locked states

- Often, randomness is intentional
  - $\Delta\Sigma$ data converters: quantization errors ⇨ out-of-band noise
  - Dynamic element matching: mismatch ⇨ out-of-band noise
  - Aperiodic calibration: periodic tones ⇨ random noise
  - Dithering: to improve linearity and suppress periodicity

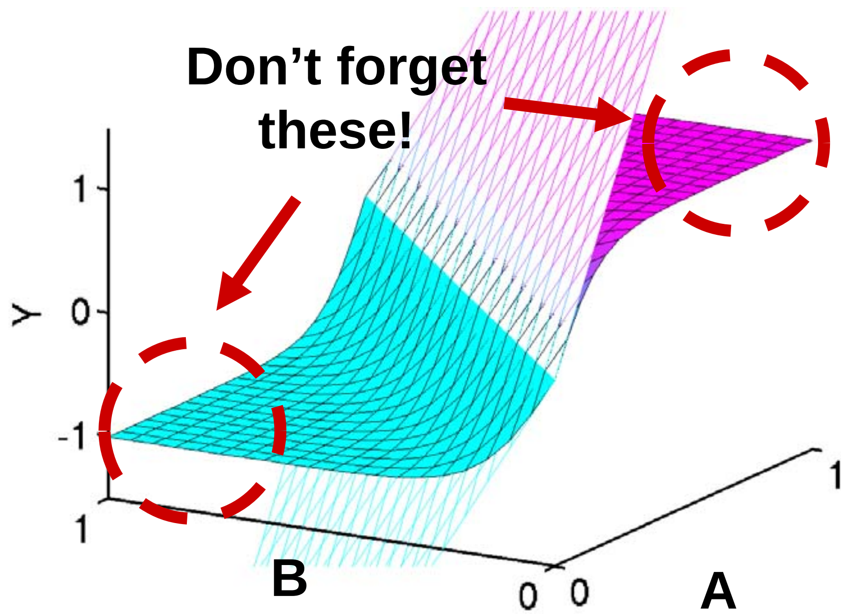
Yet, the Intent is Still a Linear System

- Despite the randomness, these nonlinear feedback loops are designed with a certain “linear system” in mind
- Most of them have “analog” counterparts
  - Bangbang PLL/CDRs
  - Digital calibration/correction loops
  - $\Delta\Sigma$ data converters
- Hence, the functionality is “analog (linear)”
  - Phase transfer function (bandwidth, peaking)
  - Calibration bandwidth, stability
  - Signal or noise transfer functions

Extending LS to Stochastic Systems

- Bangbang PLLs and $\Delta\Sigma$ data converters:
  - Intent is linear but
  - They have neither DC nor periodic steady states
- They do have steady states – in a *stochastic* sense!
  - Steady state is an ensemble of waveforms with probabilities
  - e.g. PDF (jitter histogram), PSD (noise spectrum), etc.
  - In other words, their operating point is probablistic
- Given the steady state (OP), we can find its equivalent linear system
  - J. Kim, et al., "Stochastic Steady-State and AC Analyses of Mixed-Signal Systems", DAC 2009.

Limitation of Linear Analysis

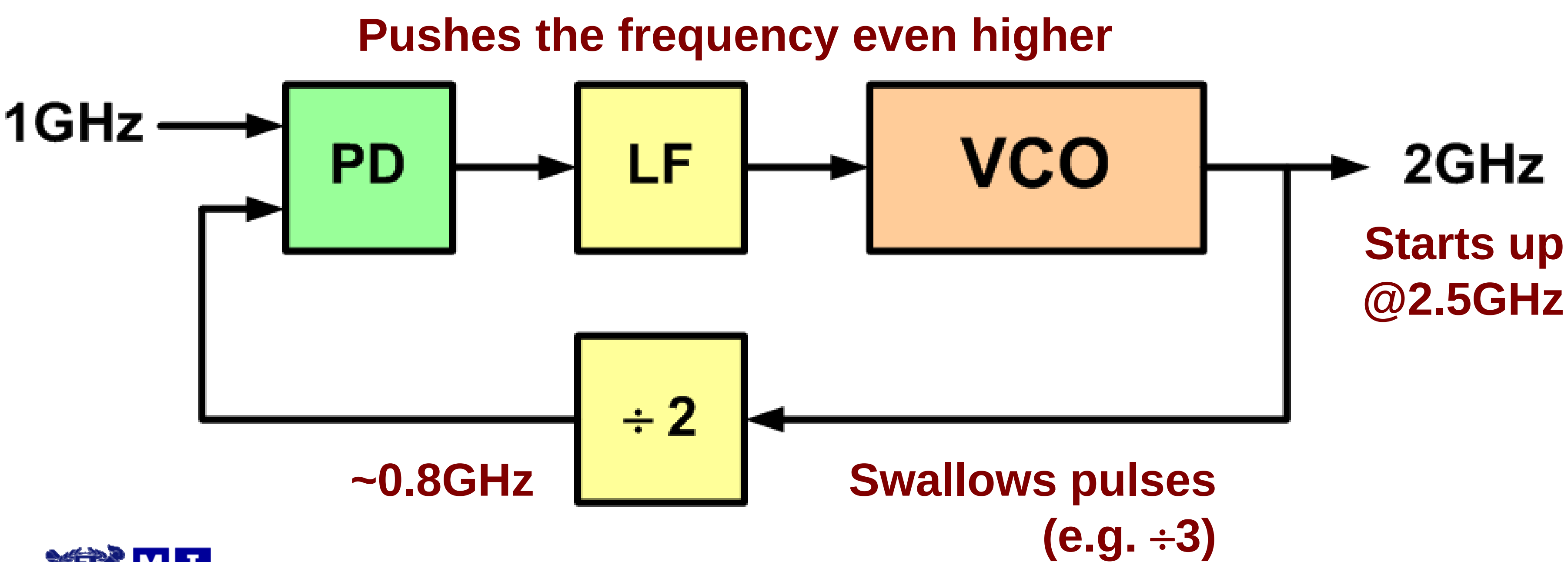


- Linear analysis verify only the <u>*LOCAL*</u> linear properties
- All circuits are eventually nonlinear

- Serious failures can occur when the circuits do not operate at the desired operate points
  - The system can behave completely different

Functional Failure Example in PLLs

- When the VCO starts at too high a frequency, a PLL may get into a dead-lock condition
- But the PLL functions correctly otherwise
  - And it's very easy to overlook this bug during design time



It's Global Convergence Problem

- Happens in nonlinear systems: converge to different equilibrium points depending on the initial points
- It calls for a way to ensure that the system will always converge to the desired OP and the desired LS