Circuit Design for a 2.2 GByte/s Memory Interface

Stefanos Sidiropoulos

Work done at Rambus Inc with A. Abhyankar, C. Chen, K. Chang, TJ Chin, N. Hays, J. Kim, Y. Li, G. Tsang, A. Wong, D. Stark

Increasing Chip I/O Bandwidth

- **Computers:**

  Main memory:

  - SDRAM100 (100 Mbps) → RDRAM (0.8-1.1 Gbps)

  Peripherals:

  - PCI (66 Mbps) → Infiniband (2.5 Gbps)

- **Networks:**

  Physical Front End:

  - LAN: Fast-Eth (100 Mbps) → Gigabit-Eth (1Gbps)
  - WAN: OC-12 (625 Mbps) → OC-48 (2.4 Gbps)

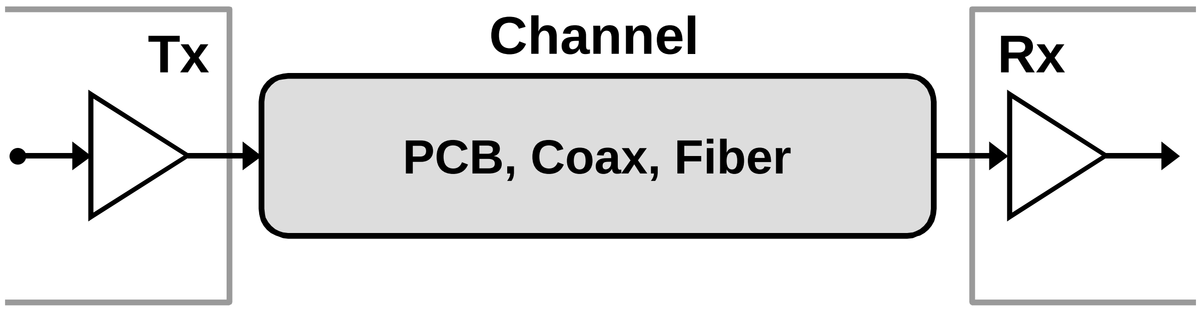
  Switch Fabric:

  - 625 Mbps → 2.5 Gbps

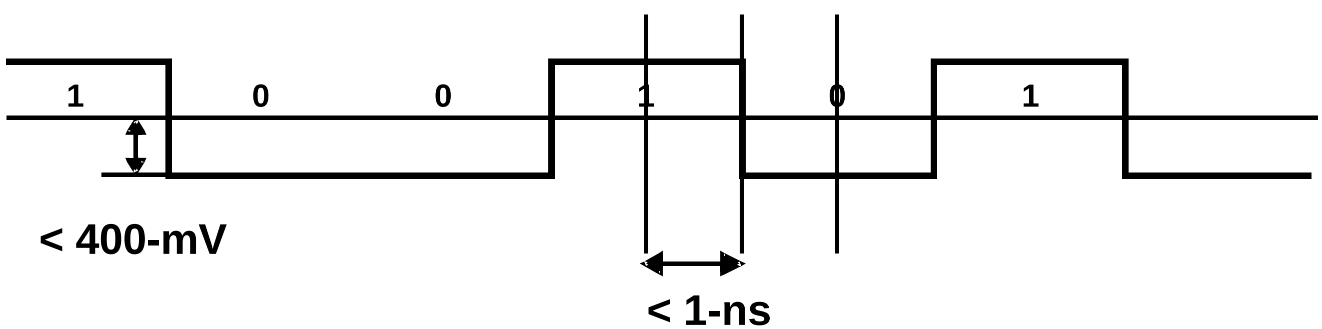
Outline

- Overview
  - Timing Methods
  - Signaling Methods
- Timing Circuits
- Signaling Circuits
- Results

Main Issues



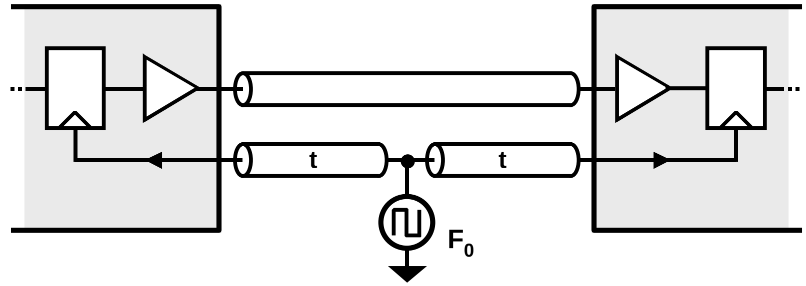
- Drive and capture signals at the correct time
  - Bit times are as small as 2-3 gate delays
- Send and receive signals robustly
  - Noise is a large fraction of the signal



Timing Architectures

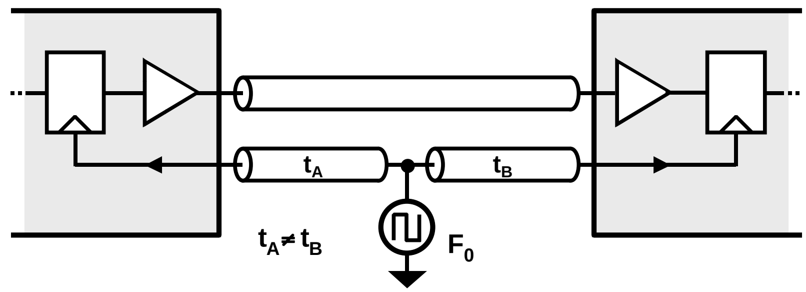
- **Synchronous:**

  Same frequency and phase
  - Conventional busses
  - Conventional Memories



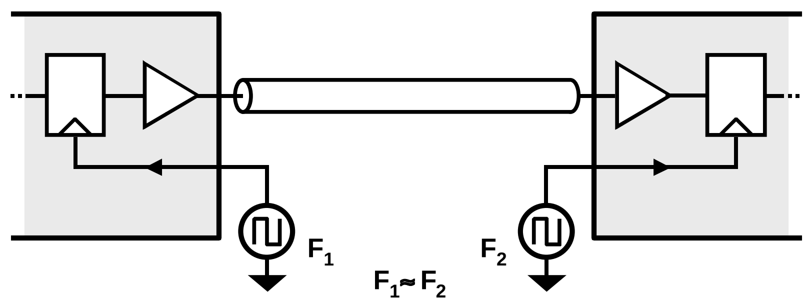
- **Mesochronous:**

  Same frequency, unknown phase
  - Fast memories/busses
  - MP networks
  - Interconnection networks

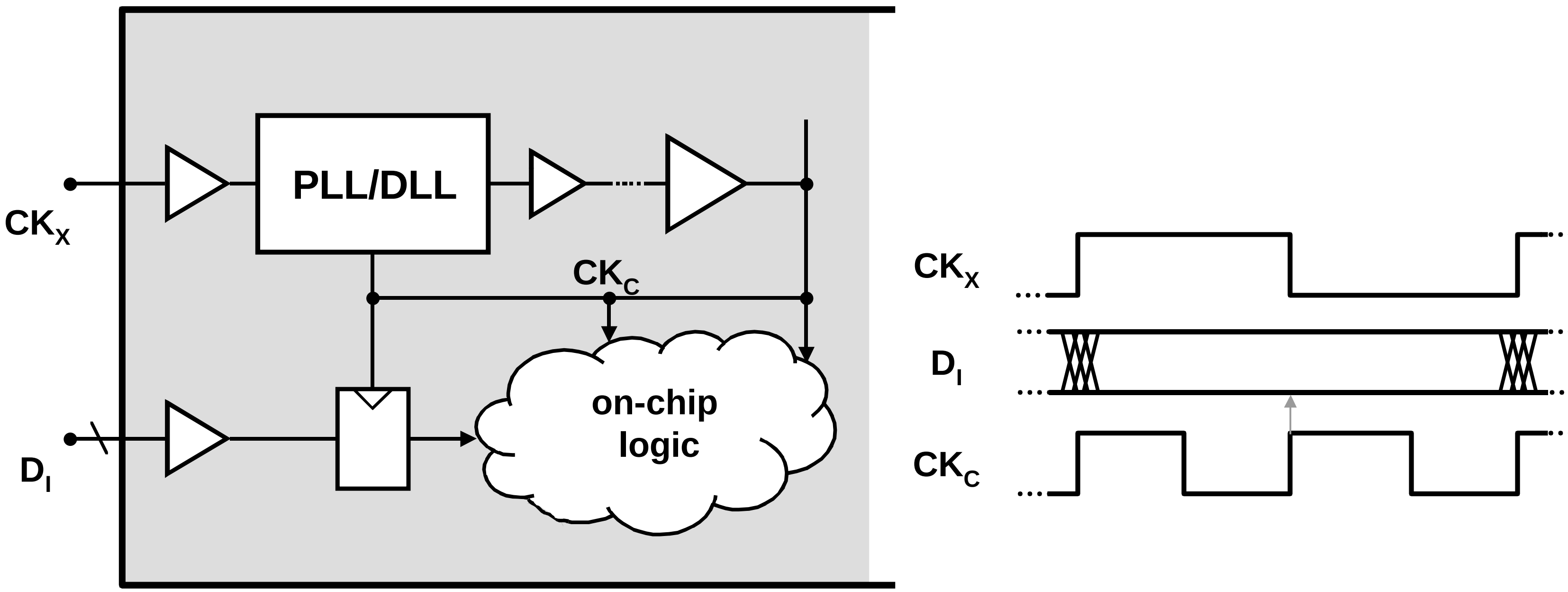


- **Plesiochronous:**

  Almost the same frequency
  - Network front-end
  - Router core

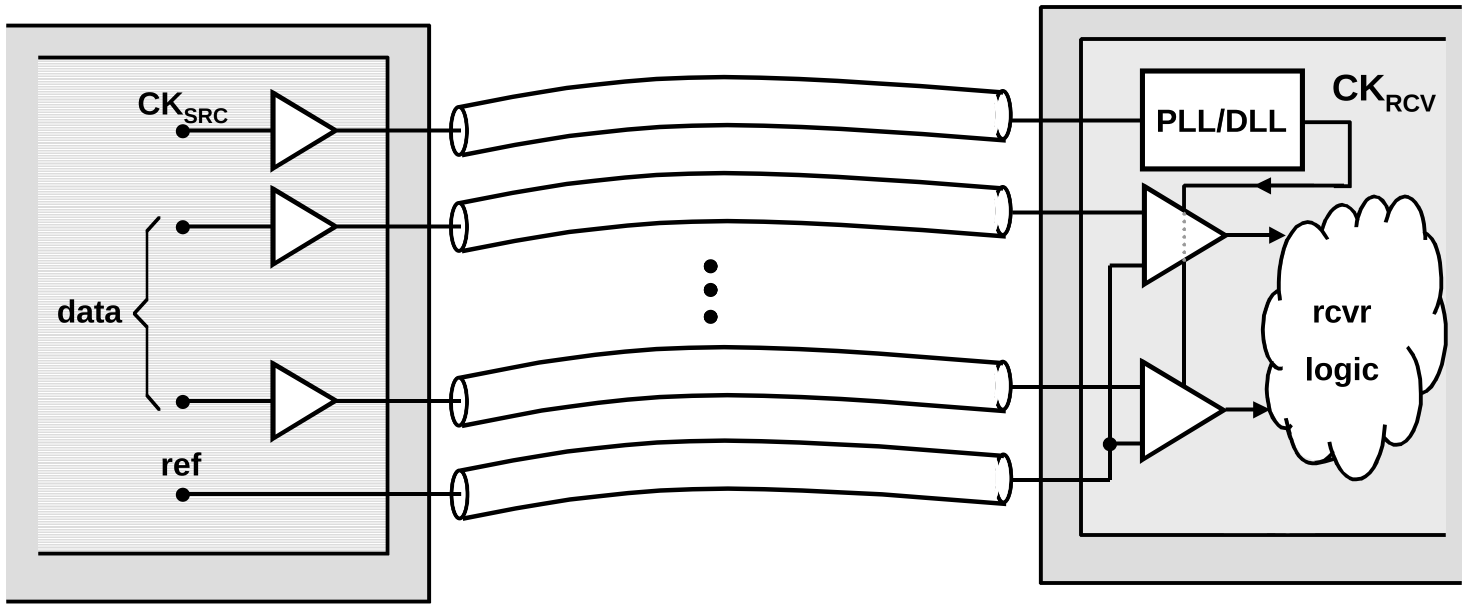


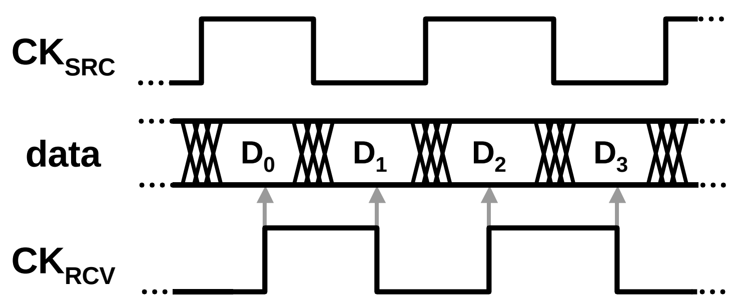
Synchronous Systems



- On-chip clock is a multiple of system clock:
  - → Synthesize on-chip clock frequency
- On-chip clock phase varies:
  - → Cancel clock buffer delay

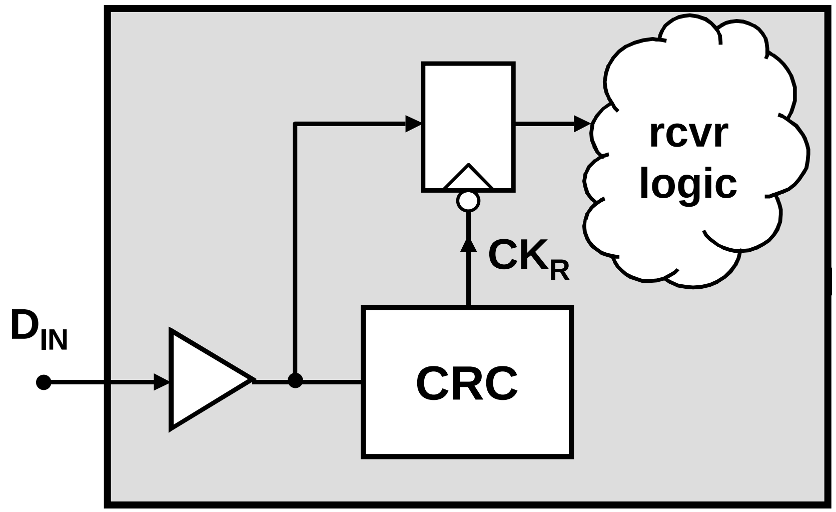
Mesochronous Systems

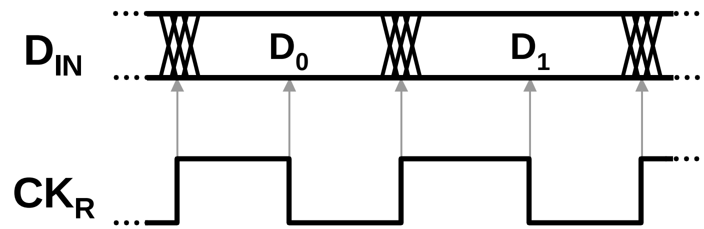




- Position on-chip sampling clock at the optimal point
  - i.e. maximize “timing” margin

Plesiochronous Systems

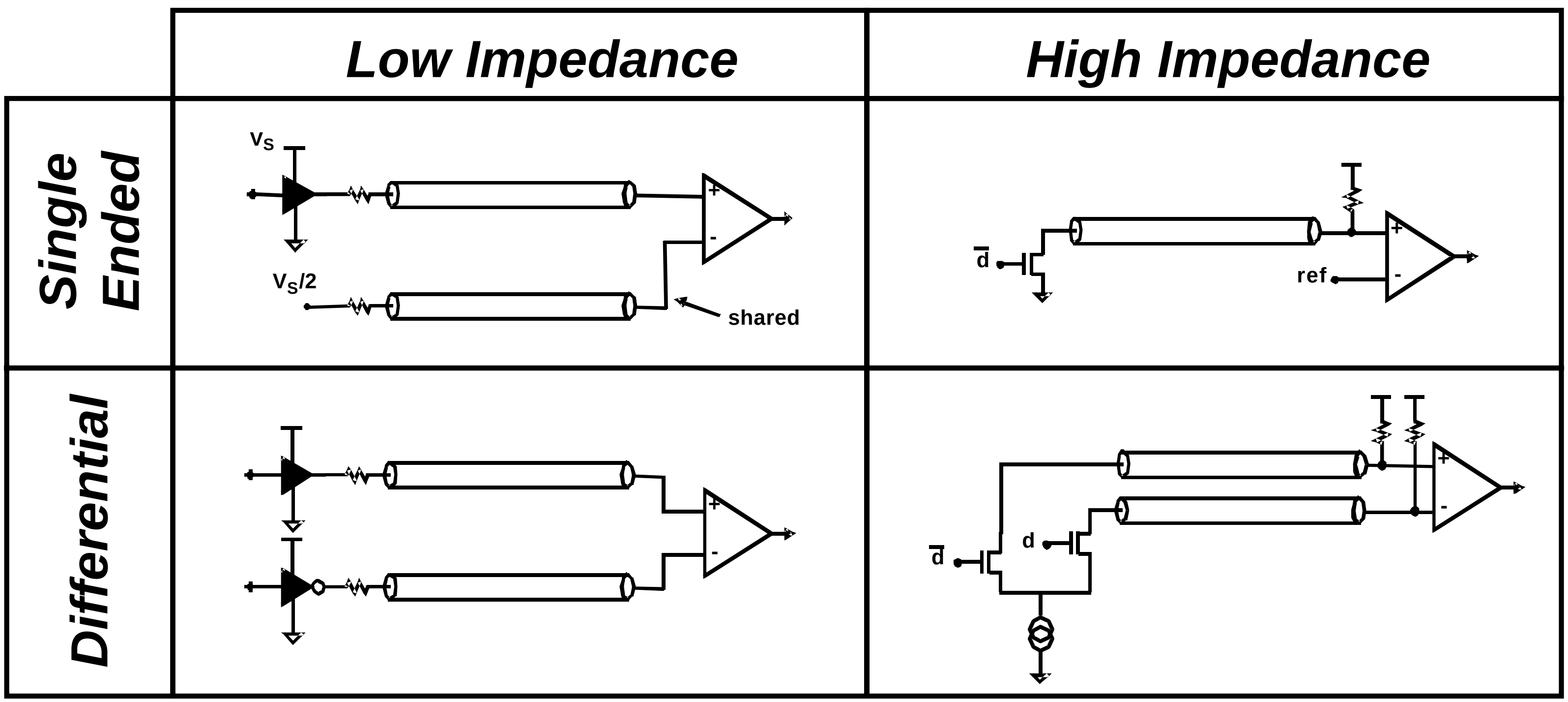




- Recover incoming data fundamental frequency
- Position sampling clock at the “optimal” point

Signaling

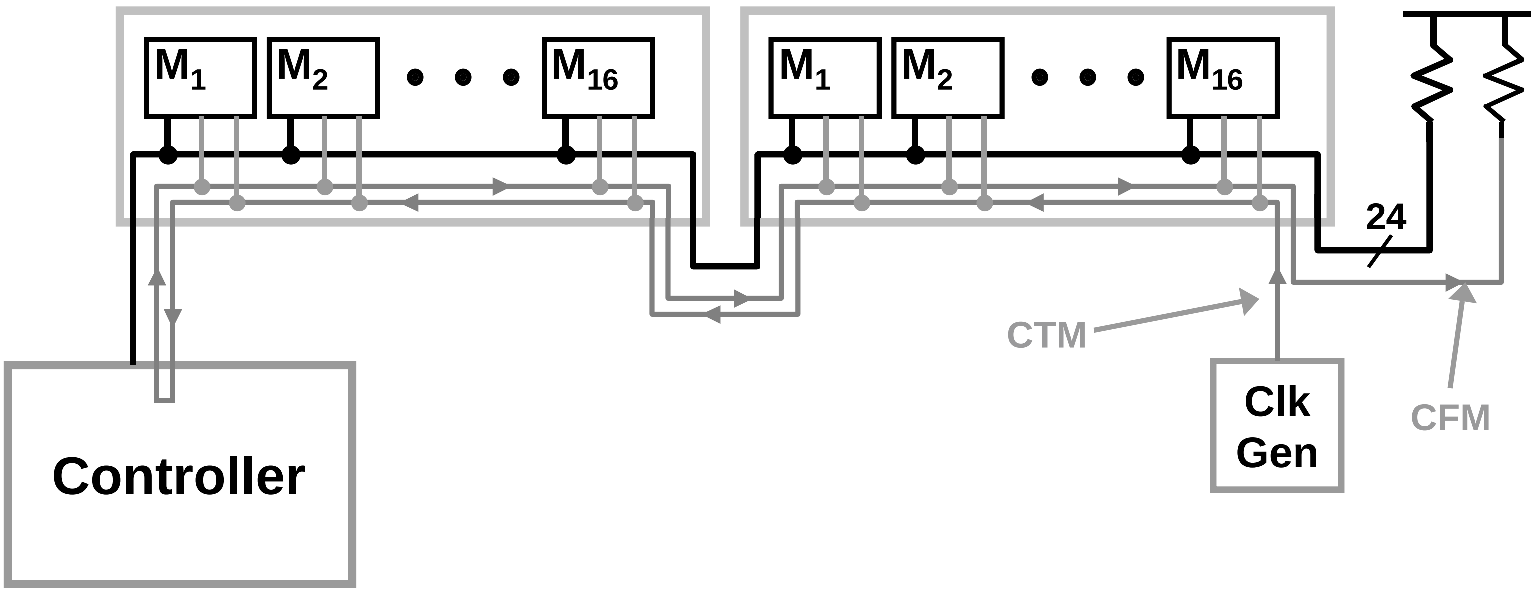
- Send and receive the data impaired by noise:
  - Independent noise sources:
    - Thermal and uncorrelated system noise
  - Proportional noise sources:
    - Reflections, cross-talk, signal-return noise

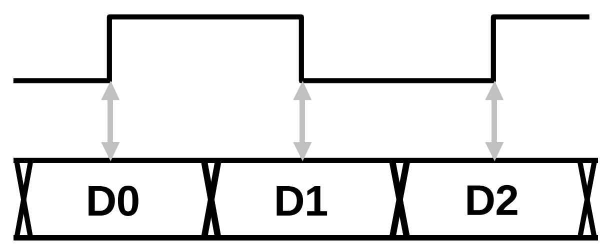


Outline

- Background
- Timing Circuits
- Signaling Circuits
- Results

Rambus Memory Channel





- 1.6-GB/s (800 Mbps/pin):
  - Current mode signaling
  - Source synchronous clocking

Increasing System Performance

- **Increase transfer rate:**

  System Clock: **400 → 533 MHz (800 → 1066 Mbps/pin)**

  Peak Bandwidth: **1.6 → 2.2 GB/s**

- **Challenges:**
  - **Timing Margin**
    - **Device Variations**
    - **Channel Imperfections**
  - **Voltage Errors**
    - **Bus Hand-off**

Prototype DRAM Interface Chip

Technology:	**0.25-μm, 2.5-V CMOS**
Supply:	**1.8-V**
Active Area:	**11.2 x 1.3 mm^2**
Package:	**LGA, μBGA**

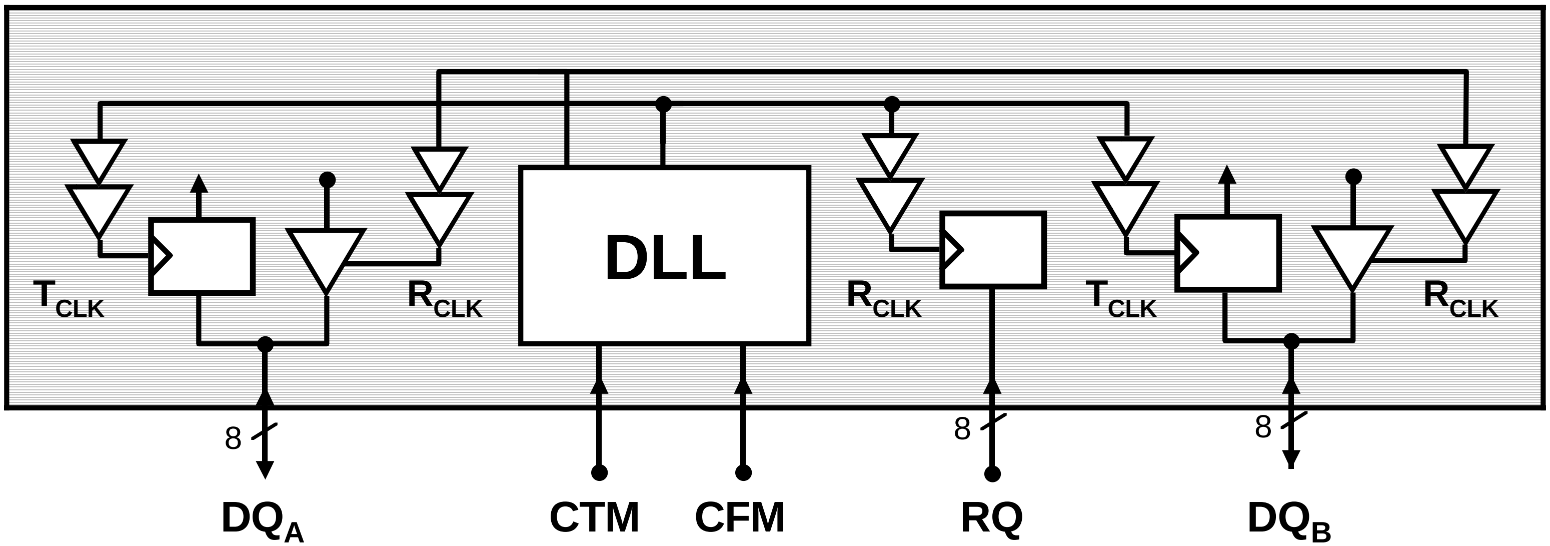
Chip Includes:

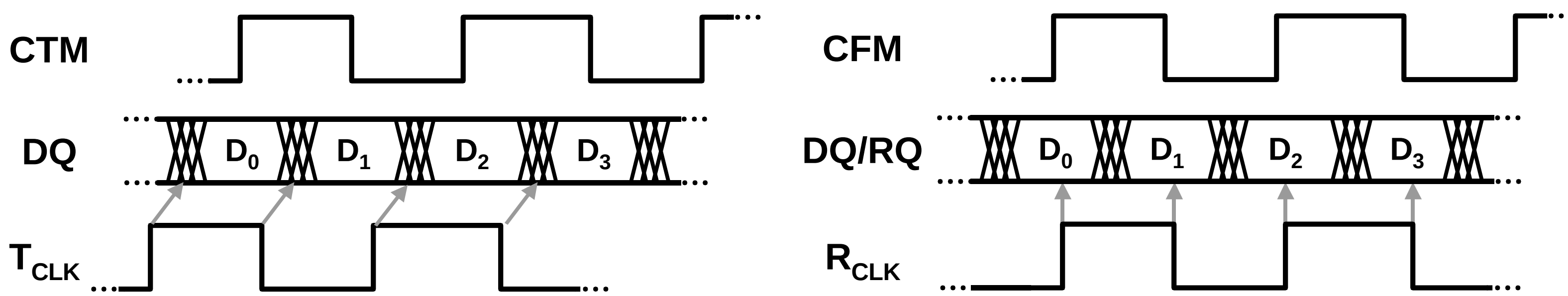
- **T/R DLL**
- **2-Data bytes, 1-Address byte**
- **Packet Protocol Logic**
- **18 KB SRAM**

Outline

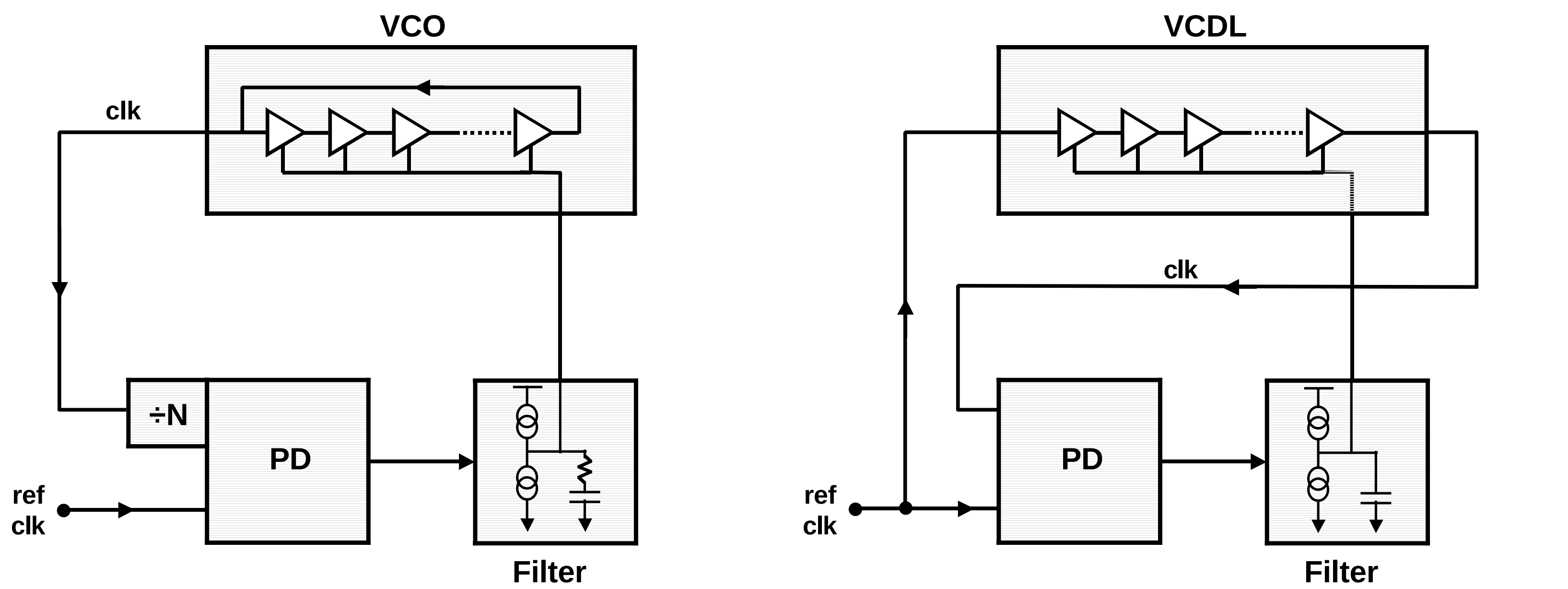
- Background
- Timing Circuits
  - Requirements
  - Architecture
  - Timing Error Sources
- Signaling Circuits
- Results

RDRAM Timing Circuit Requirements





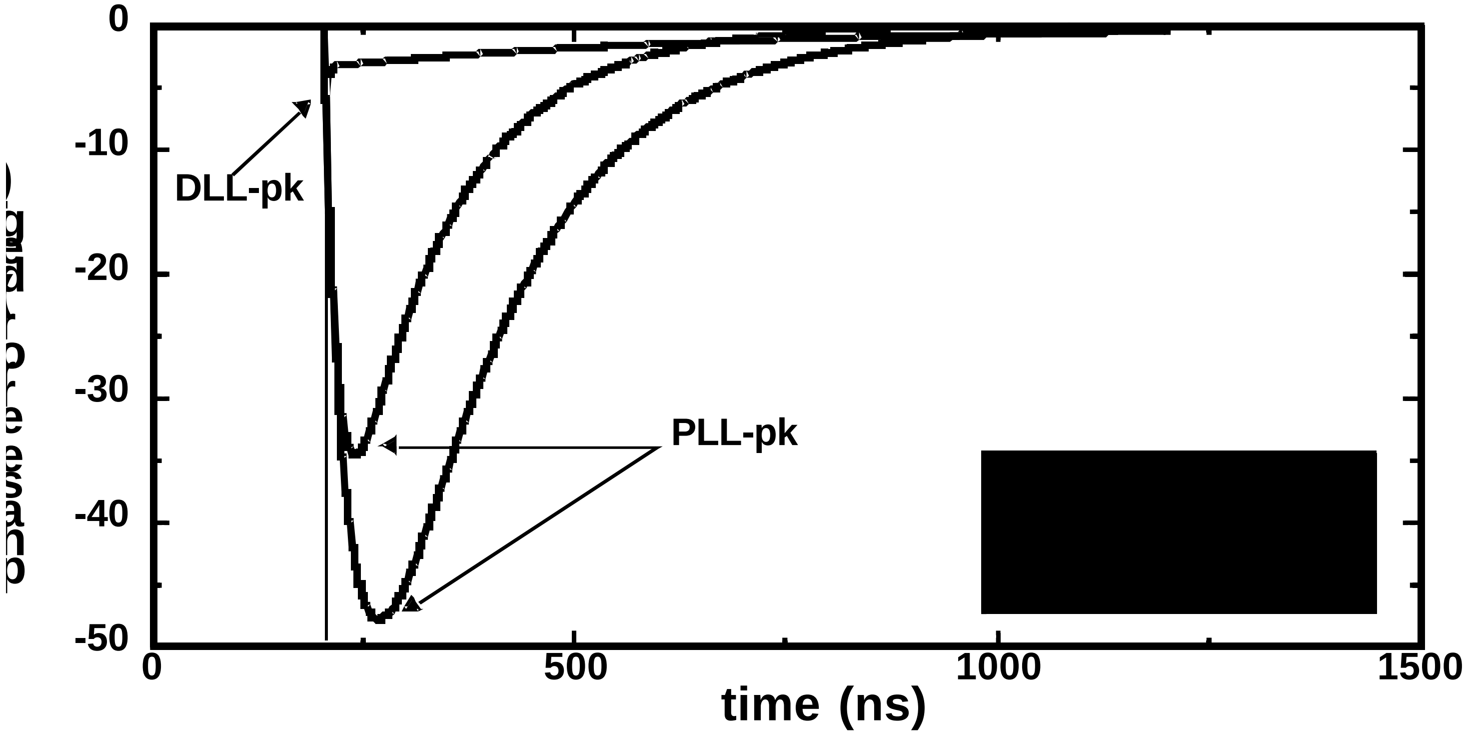
PLLs vs DLLs



- **Second/third order loop:**
  - → **Stability is an issue**
  - → **Frequency synthesis easy**
  - → **Ref. Clk jitter gets filtered**
  - → **Phase error accumulates**
- **First order loop:**
  - → **Stability guaranteed**
  - → **Frequency synthesis problematic**
  - → **Ref. Clk jitter propagates**
  - → **Phase error does not accumulate**

Supply Noise: DLL vs PLL

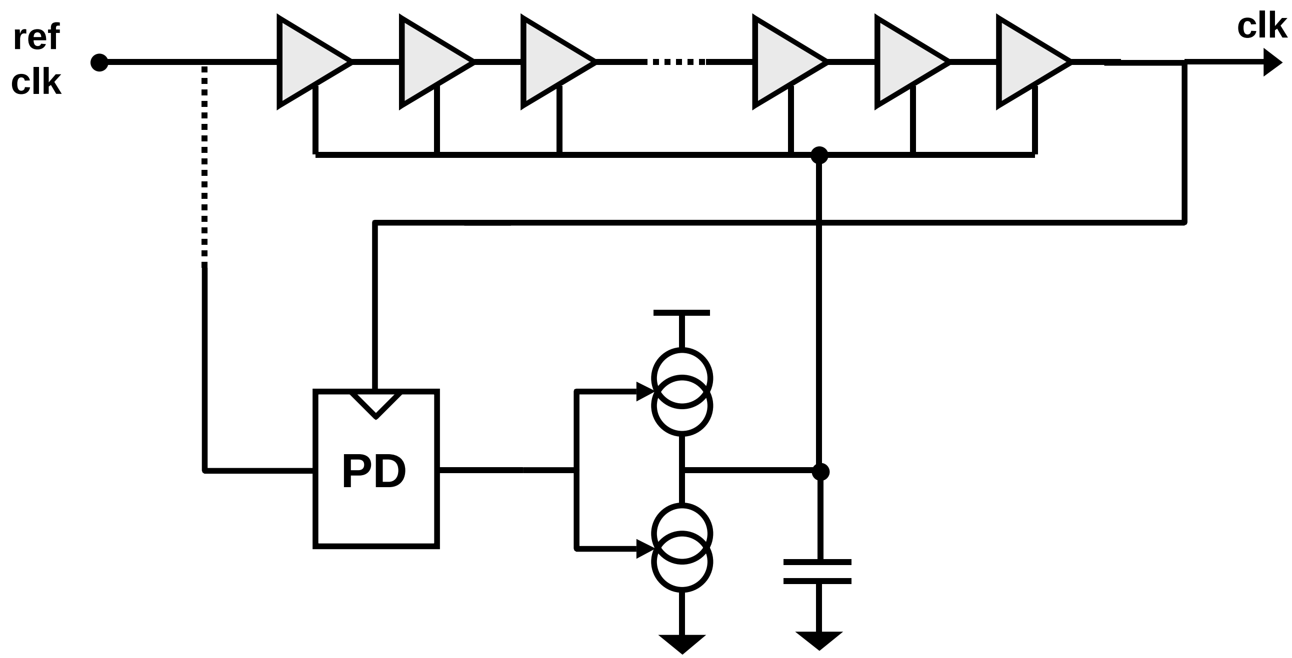
6-stage DLL vs 6-stage PLL



*** Supply sensitivity: 0.1%-delay/%-supply/element**

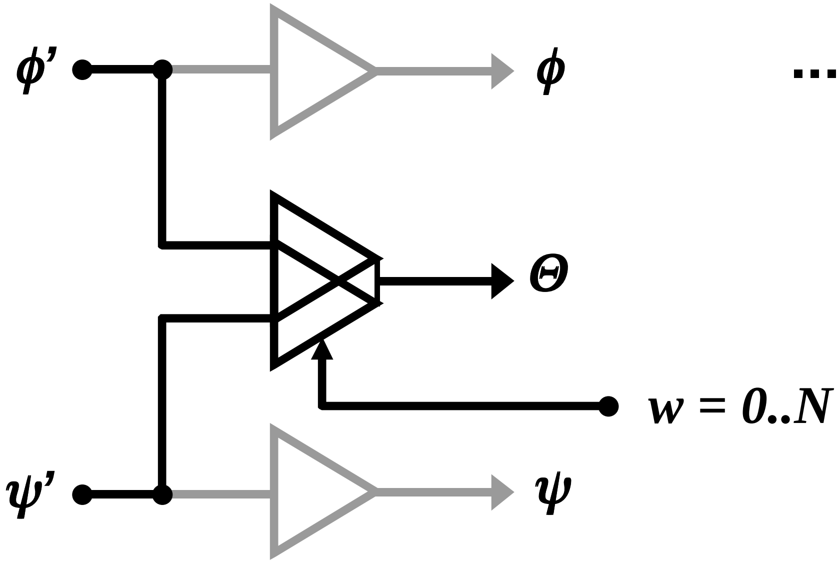
- **No need for clock multiplication ➜ use a DLL**

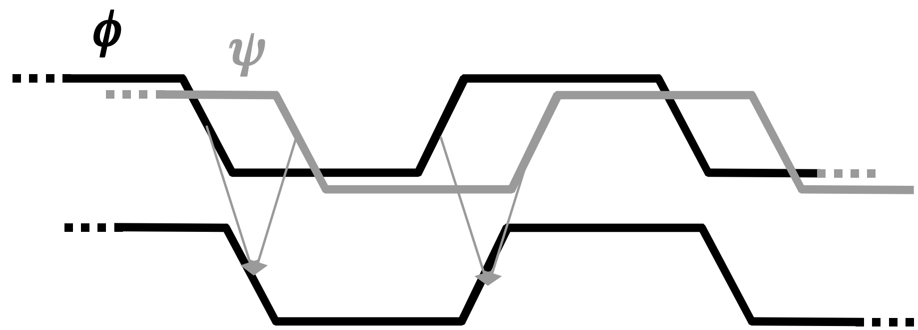
Conventional DLL



- Limited phase acquisition range
  - → Generate delay by using phase interpolation

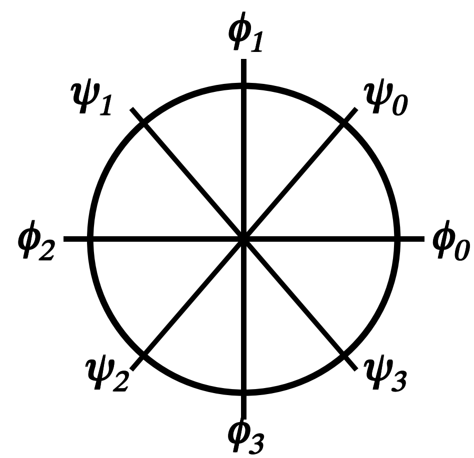
Variable Phase Interpolation



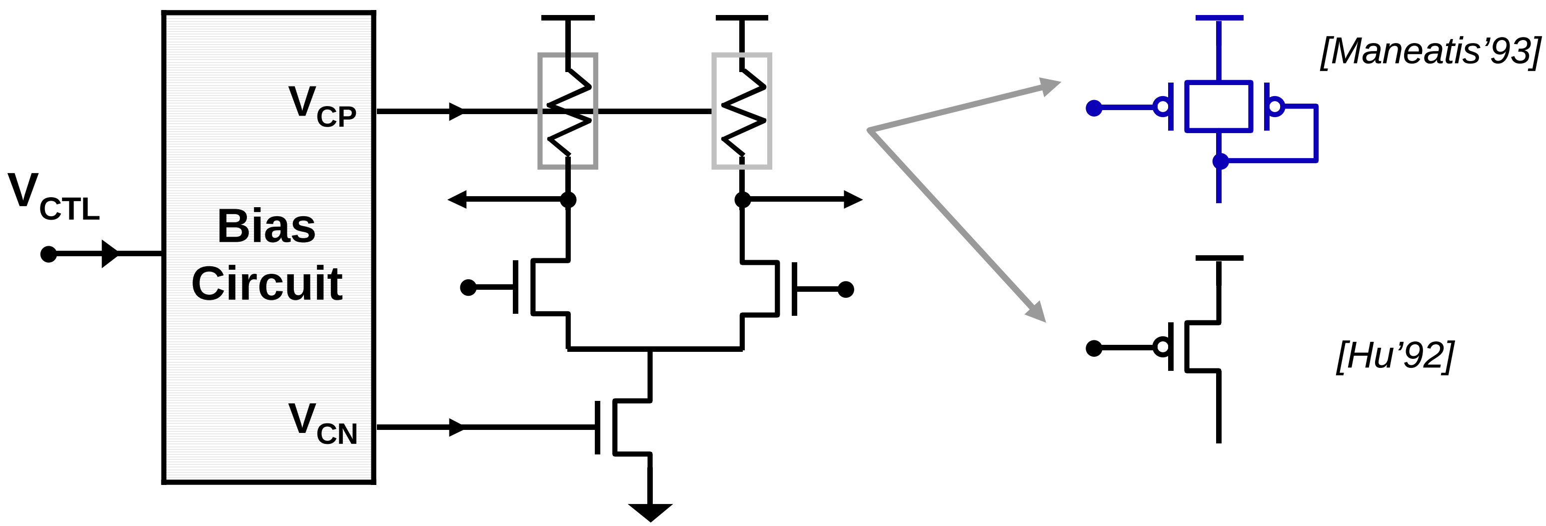


$$\Theta = \frac{(N - w)\cdot\phi + w\cdot\psi}{N}$$

- If ϕ, ψ selectively span 2π:
  - → Can generate any Θ
- ϕ, ψ can be generated by a DLL

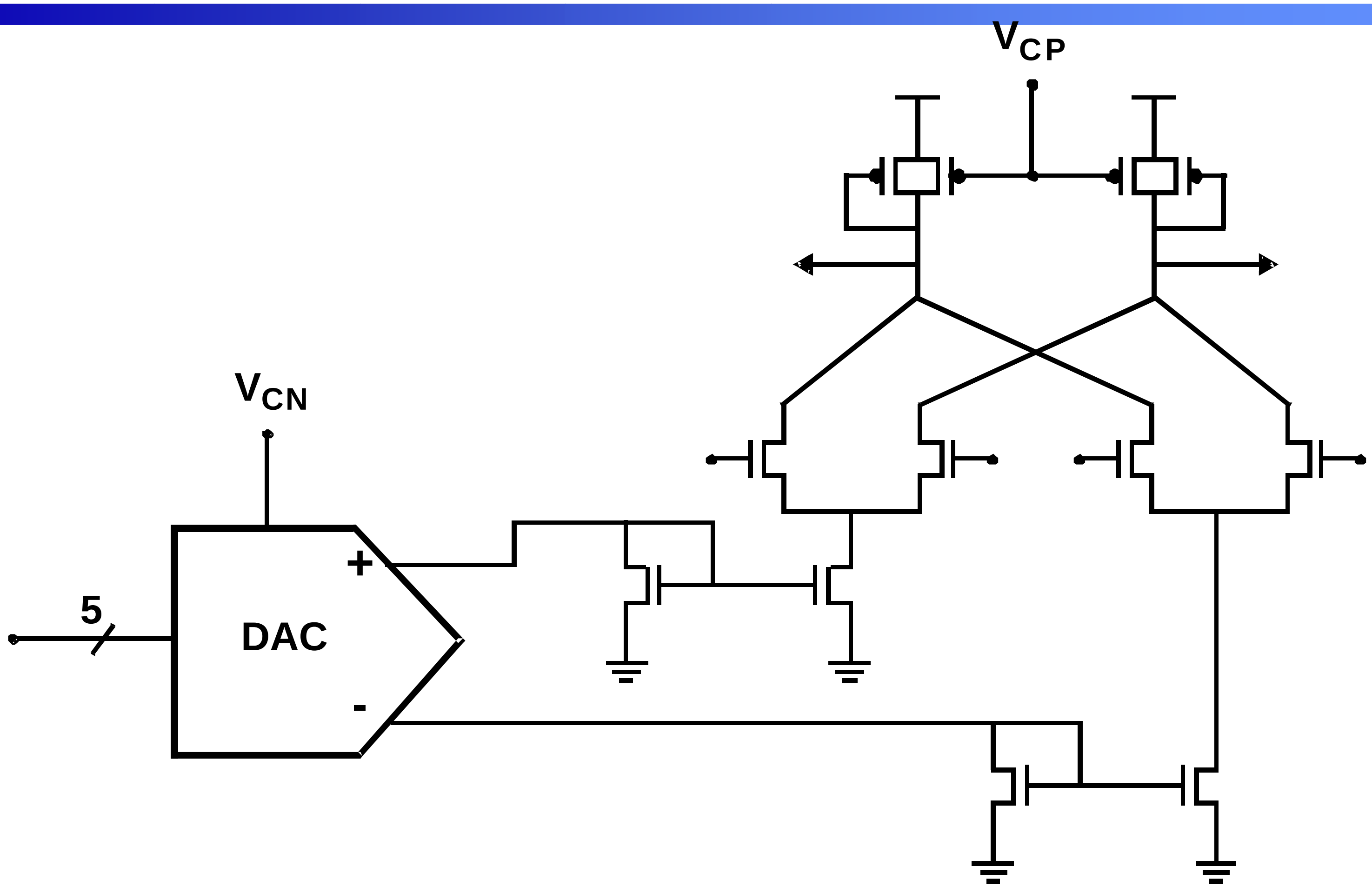


RDRAM Delay Buffers



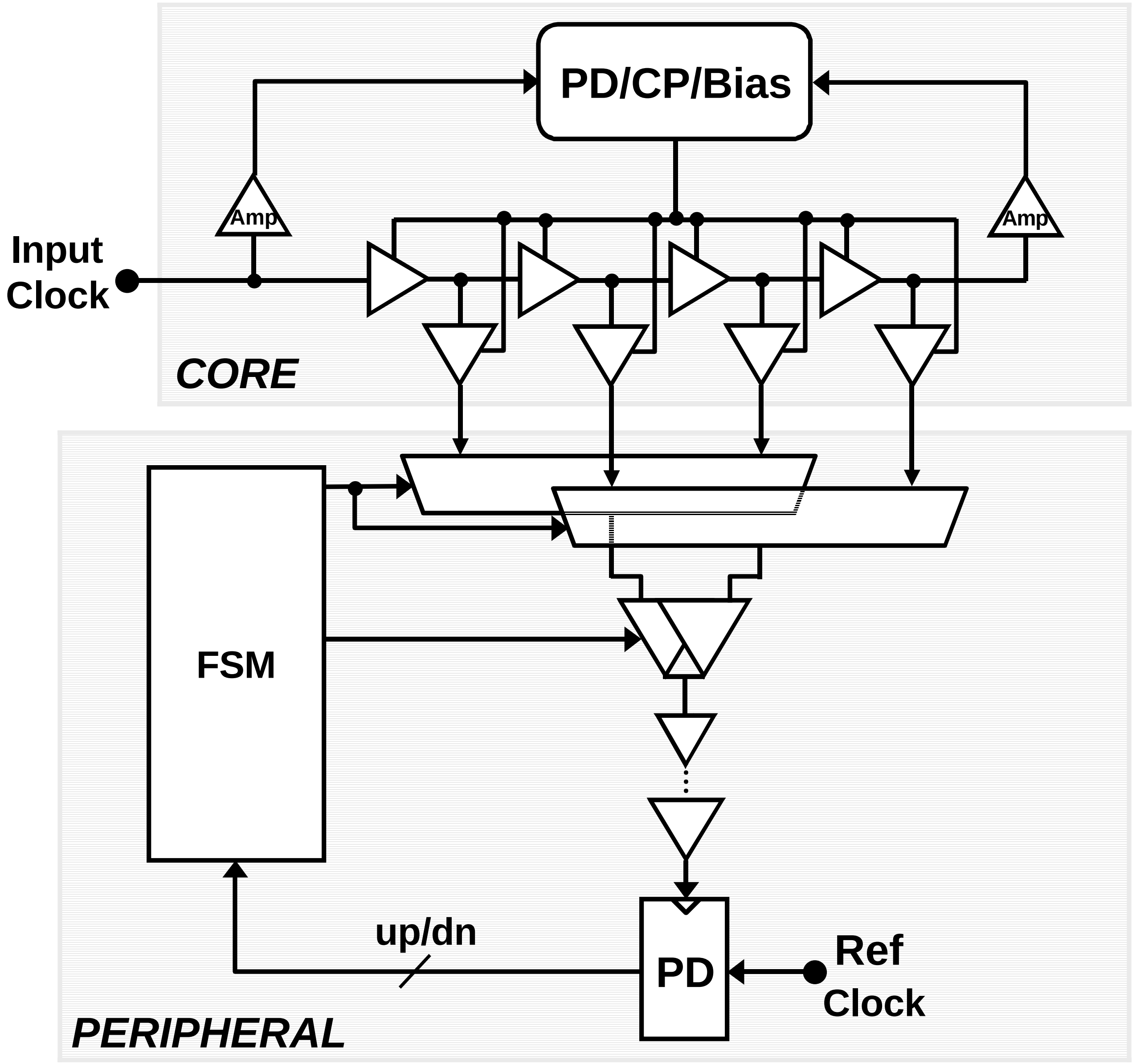
- Use differential elements with replica biasing:
  - ✓ Increased noise immunity
  - ✗ Not easily portable
  - ✗ Require larger supply head-room but ok for 1.8-V

Interpolator Design

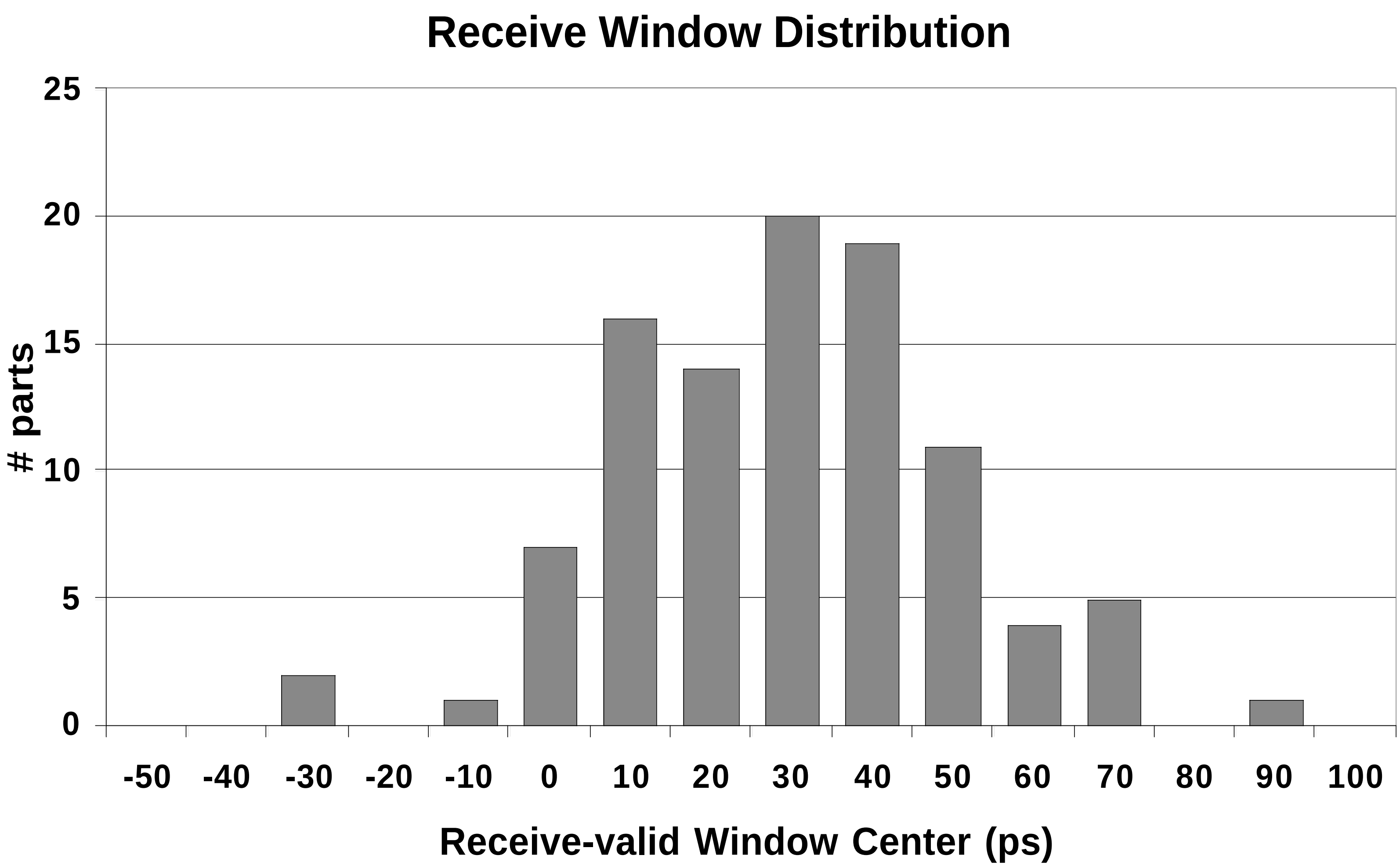


- Interpolator bias and input/output time constant scales
  - → TDC remains linear over large frequency range

Dual DLL Block Diagram

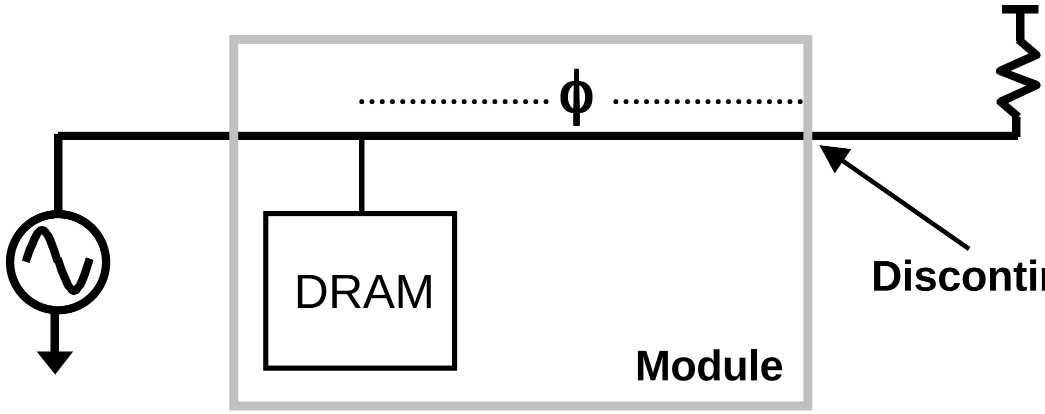


Device Timing Variations

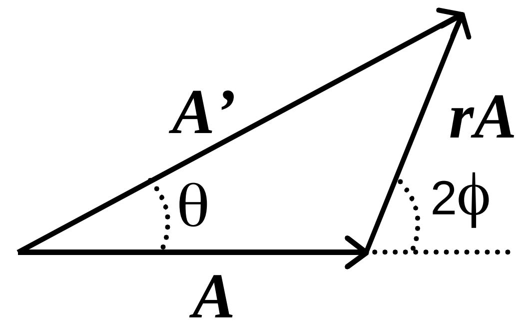


- 100 parts: $\mu \cong$ 30-ps, $\sigma \cong$ 20-ps

Propagation Delay Mismatch

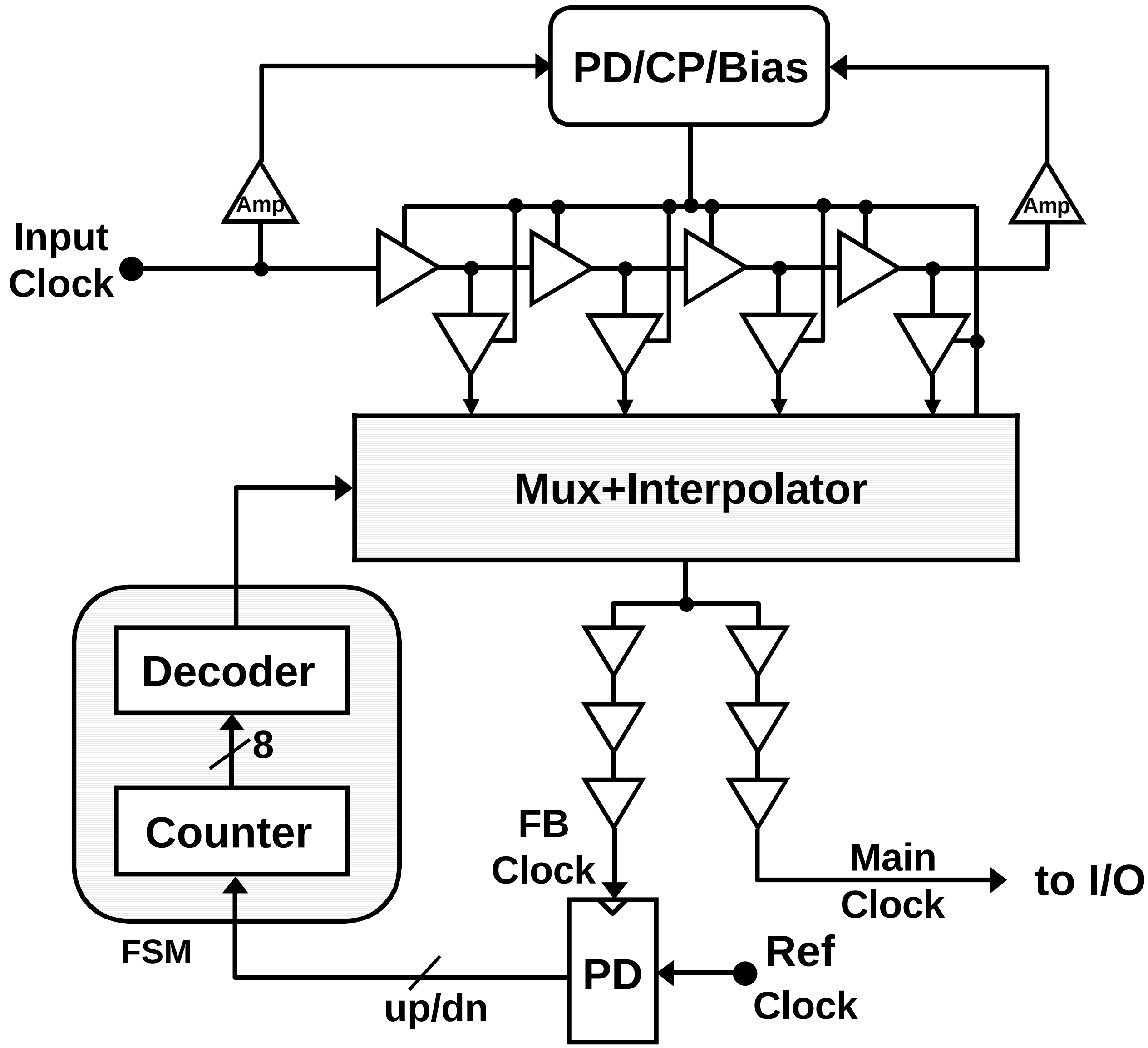


$$v(t) = A \cdot [\sin(\omega \cdot t) + r \cdot \sin(\omega \cdot t - 2\varphi)]$$
$$\Rightarrow v(t) = A' \cdot \sin(\omega \cdot t + \theta)$$

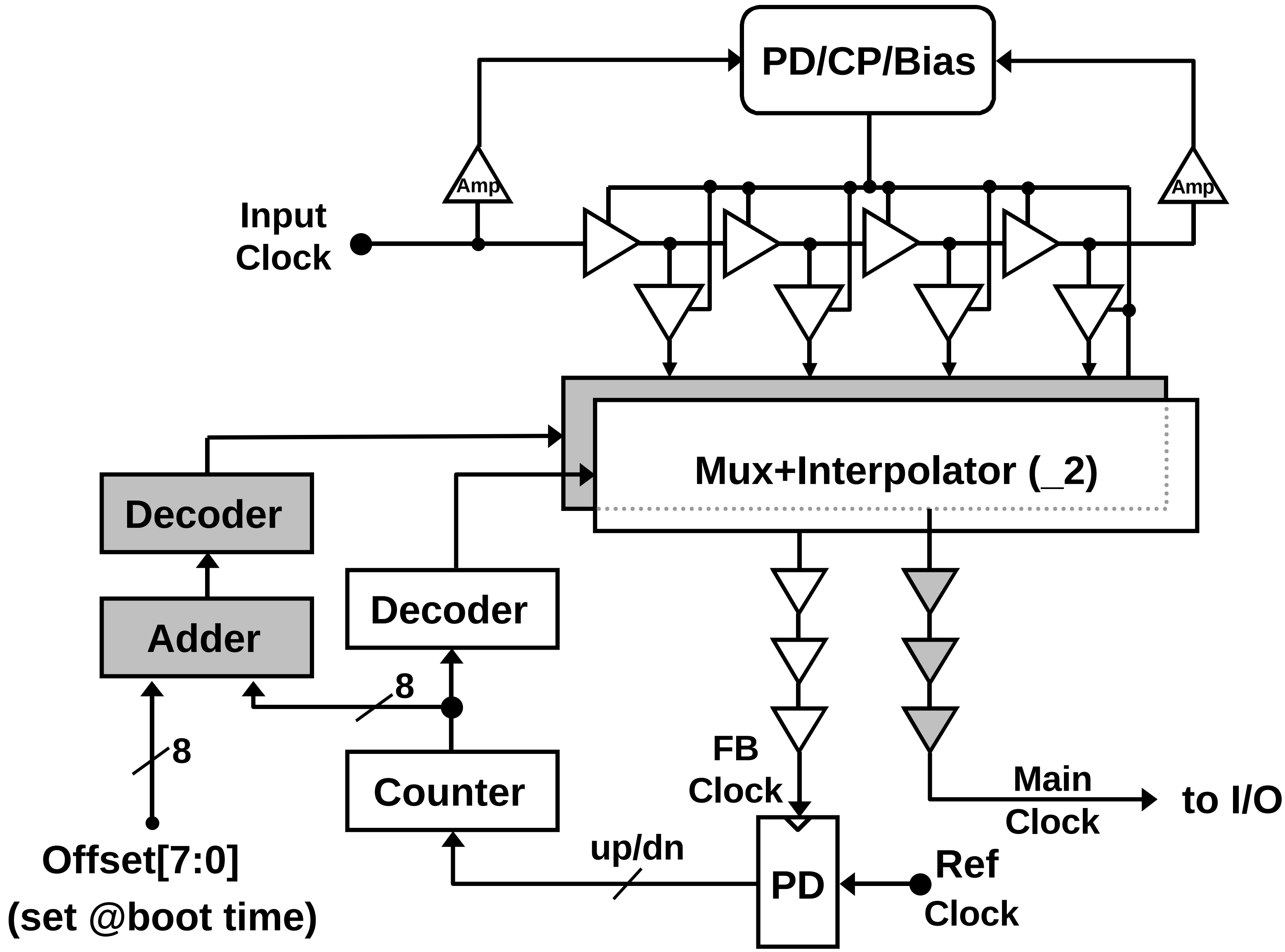


- Clock and data channels different
- Clock and data spectral components different
  - → Propagation delays can differ by ~ 100-ps
- Regain margin: every DRAM transmit/receive timing must be offset from its lock point

Original Dual-DLL



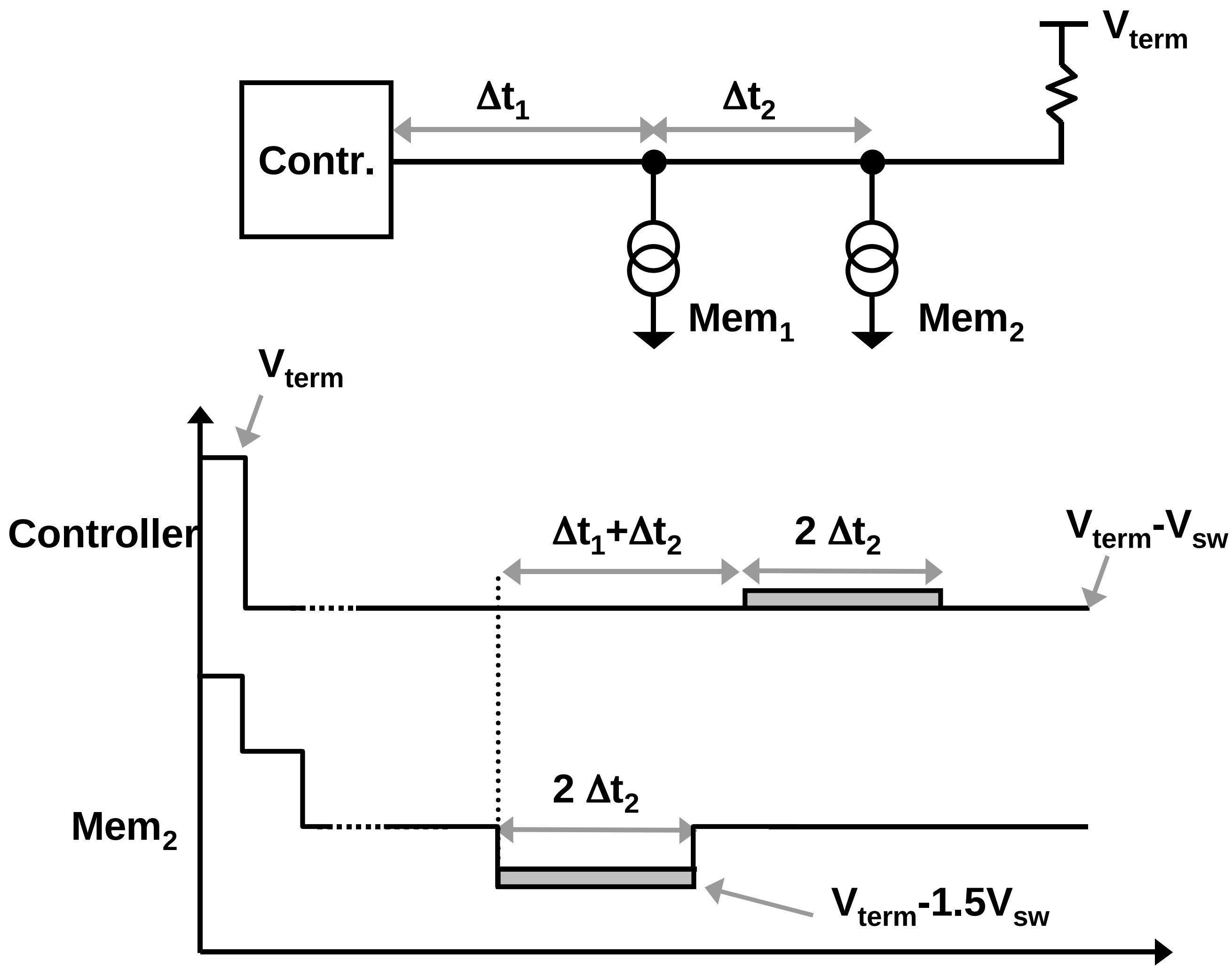
DLL for "in-system" Calibration



Outline

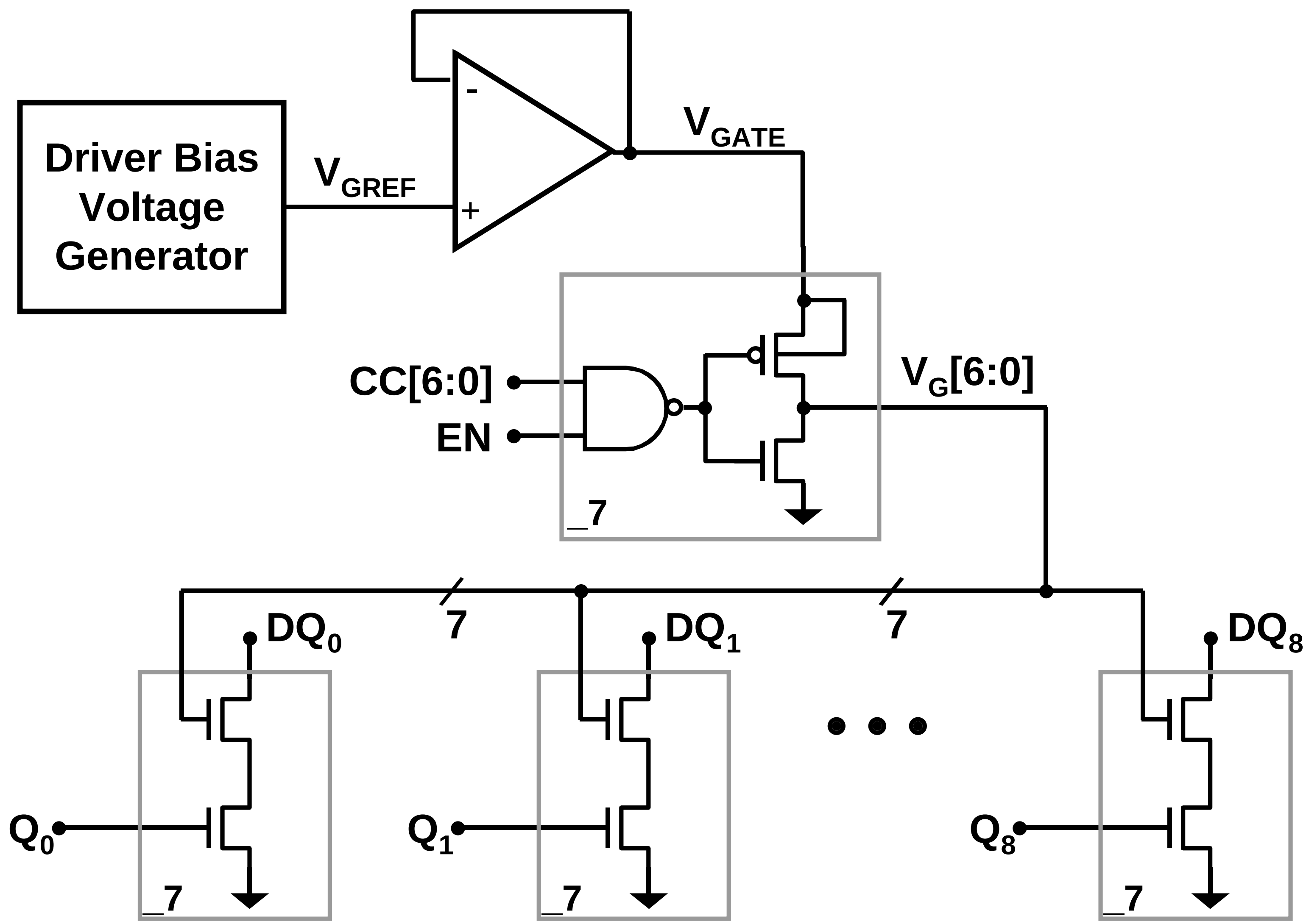
- Background
- Timing Circuits
- Signaling Circuits
  - Bus Environment Challenges
  - Output Subsystem Design
- Results

“Back-to-Back” Reads

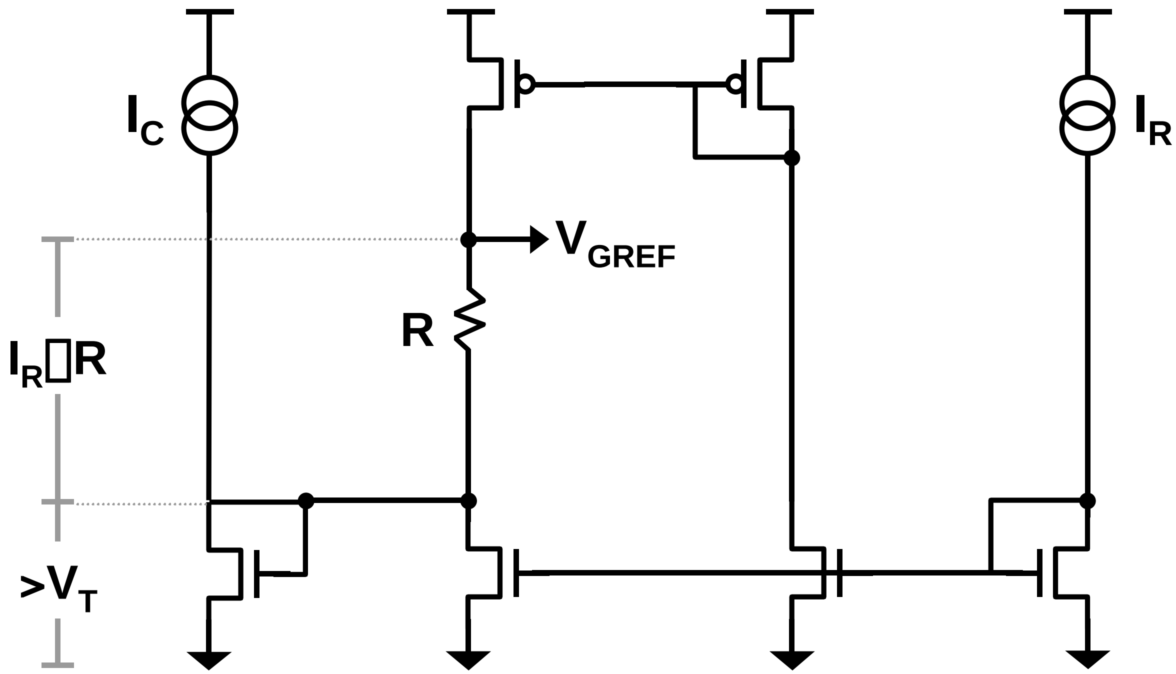


- Compliance voltage for M_2 as low as 0.5-V

Output Driver Subsystem

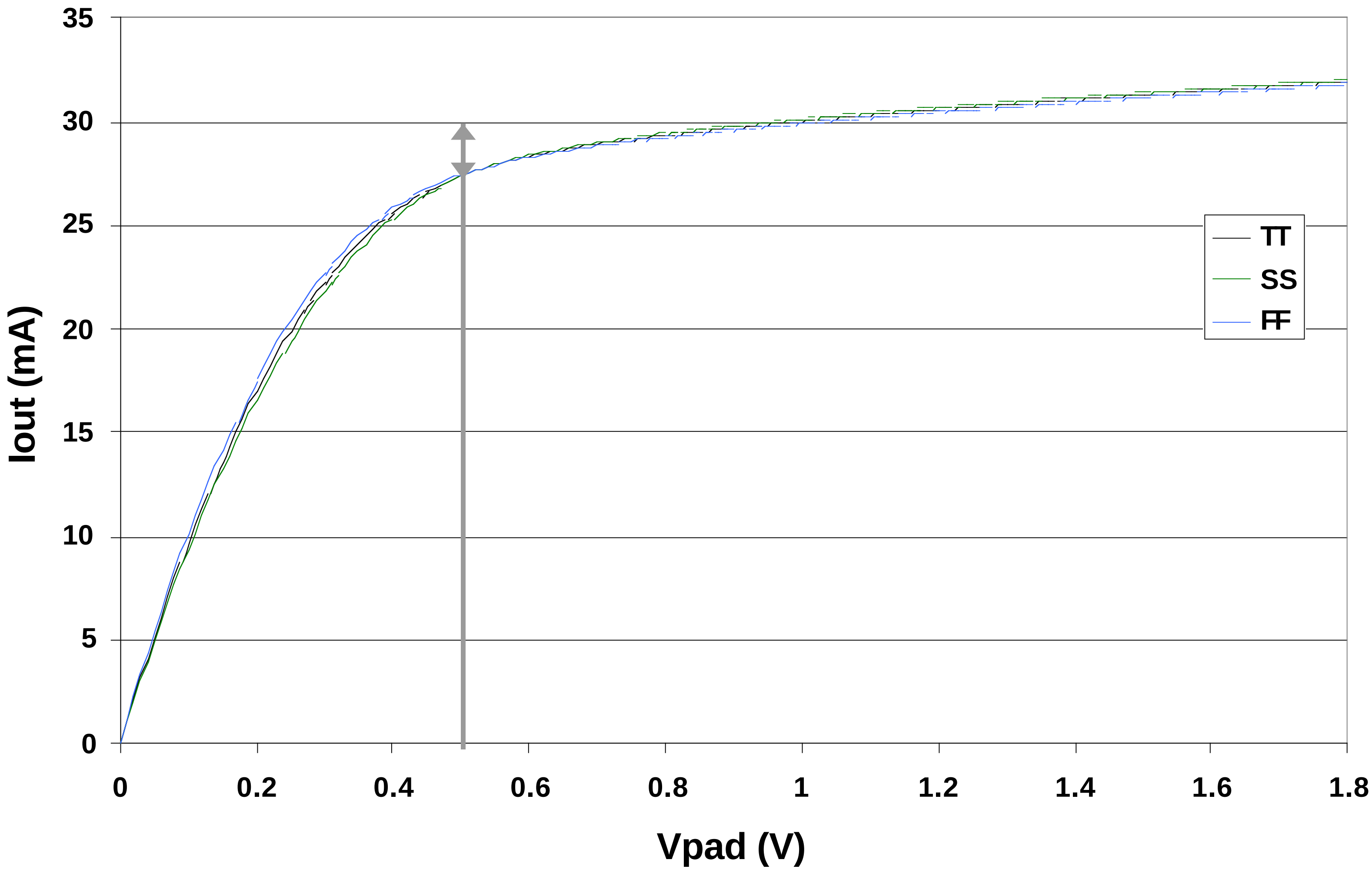


Driver Bias Voltage Generator

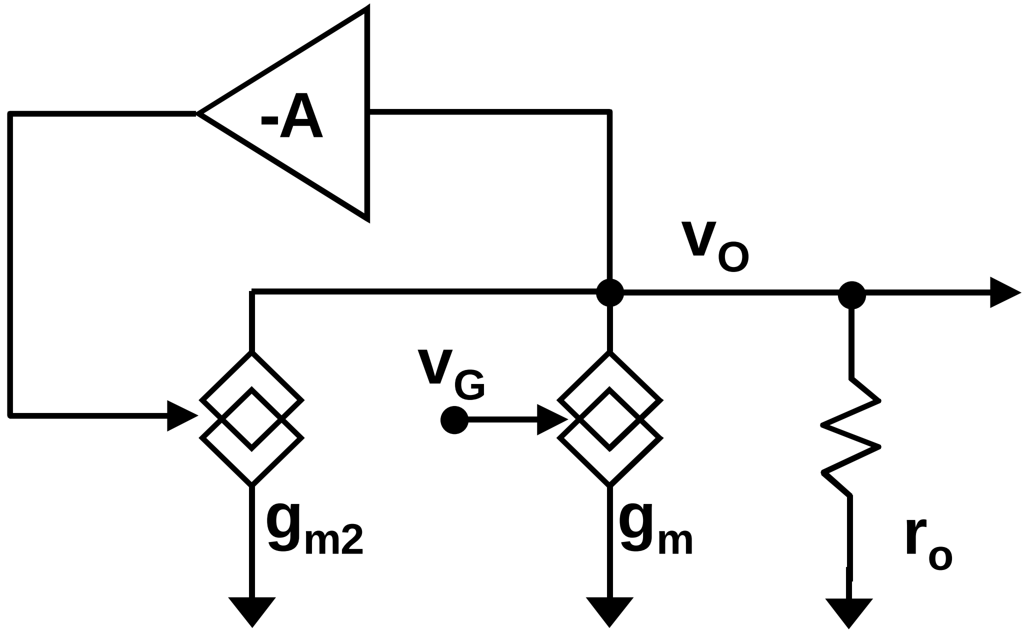


- Constant gate overdrive:
  - Increase noise immunity
  - Constant saturation margin over PVT

Driver IV Characteristics



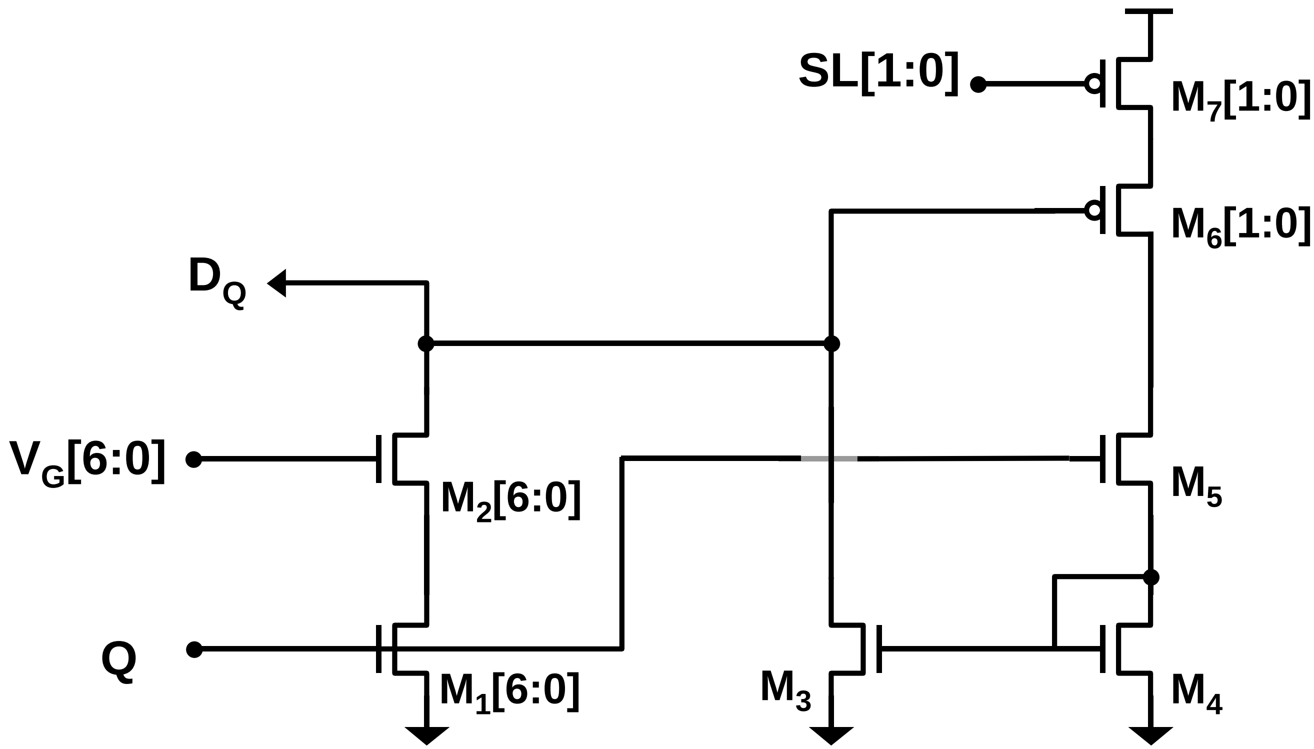
Output Driver Model



$$i_{out} = g_m \cdot v_g + v_o / r_o - A \cdot g_{m2} \cdot v_o$$

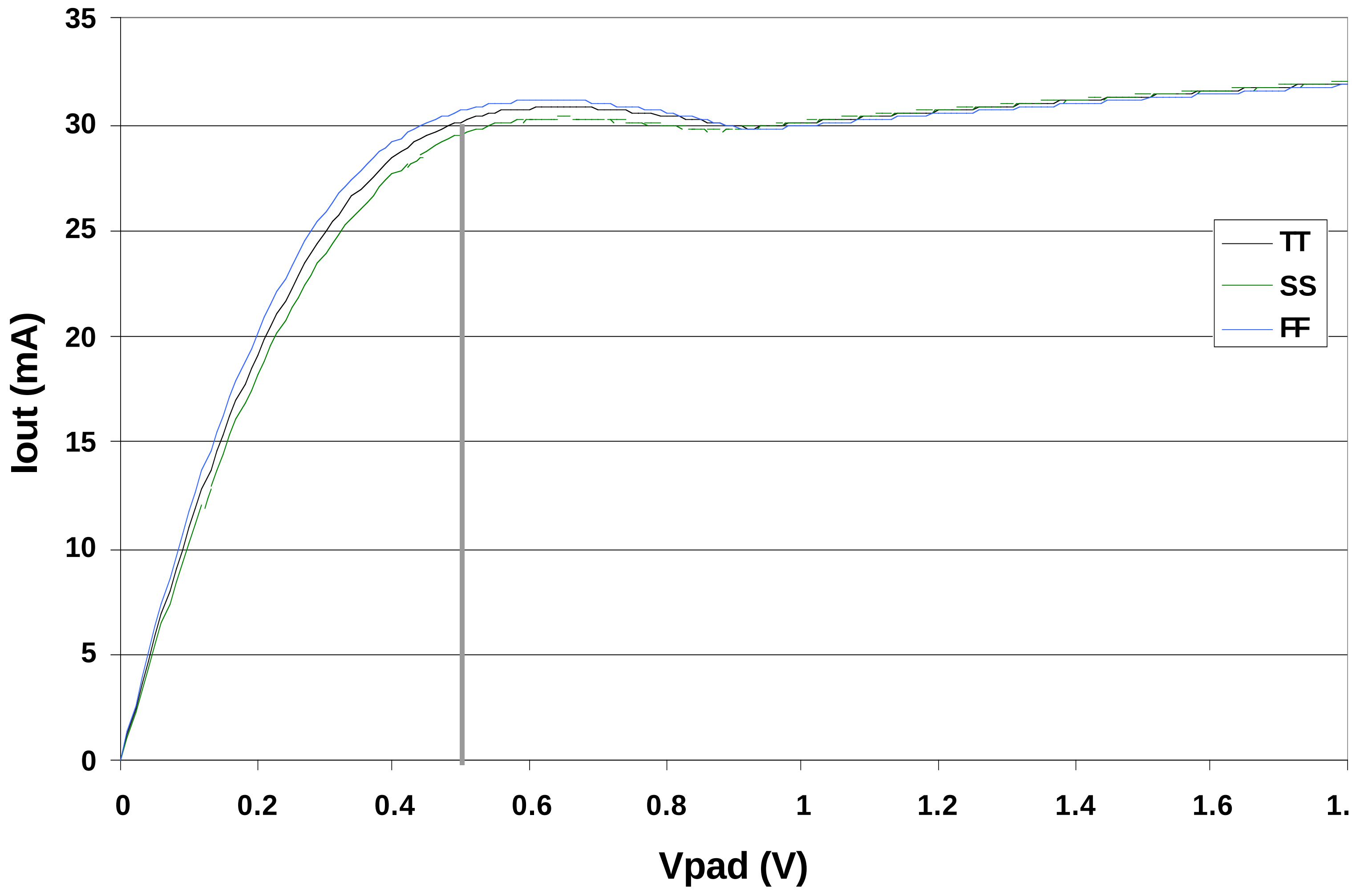
- Negative resistance compensates for finite r_o

Output Driver Schematic



- M6-M7 control maximum feedback current
- M3/M4 ratio constrained to minimize time constant

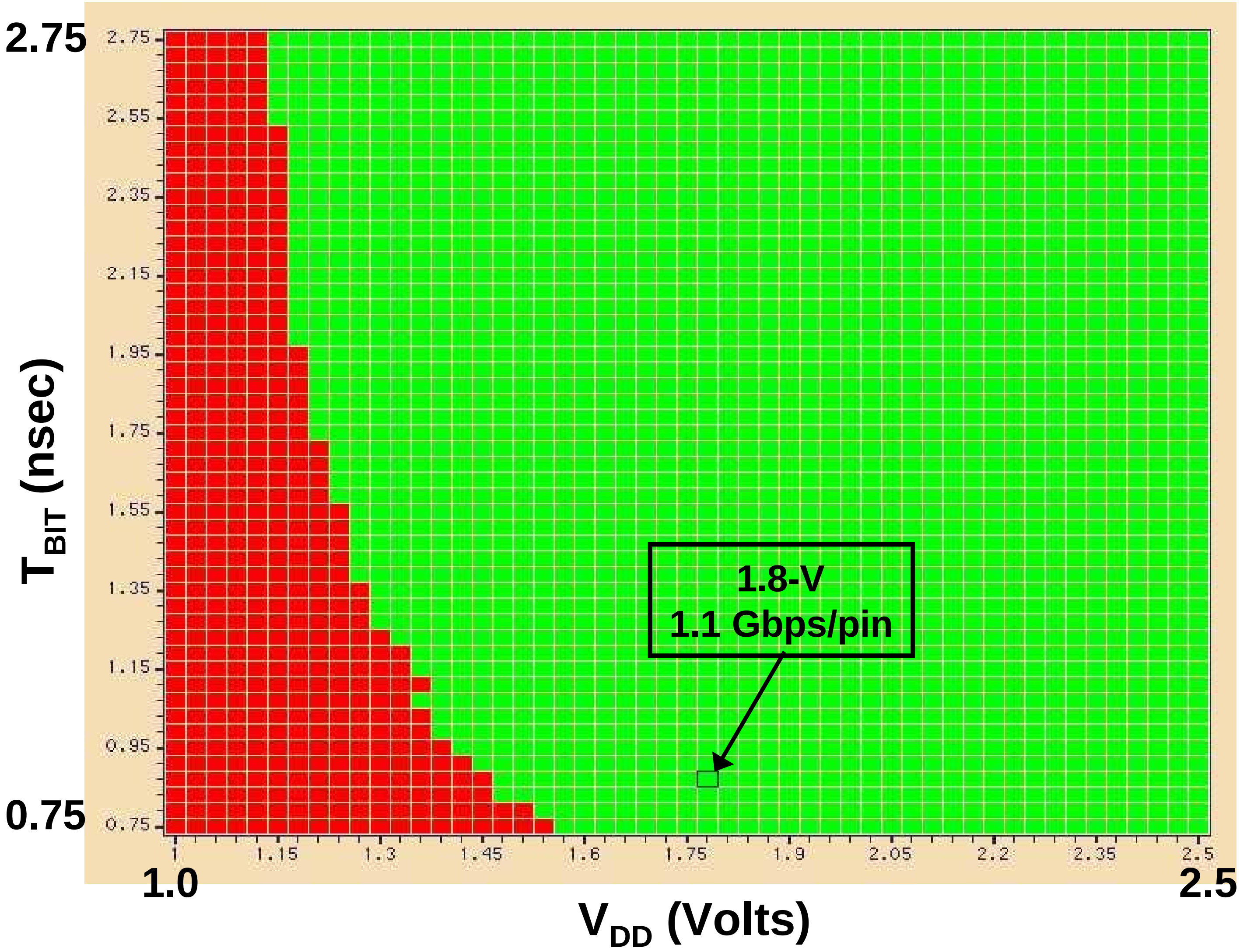
Driver IV Characteristics



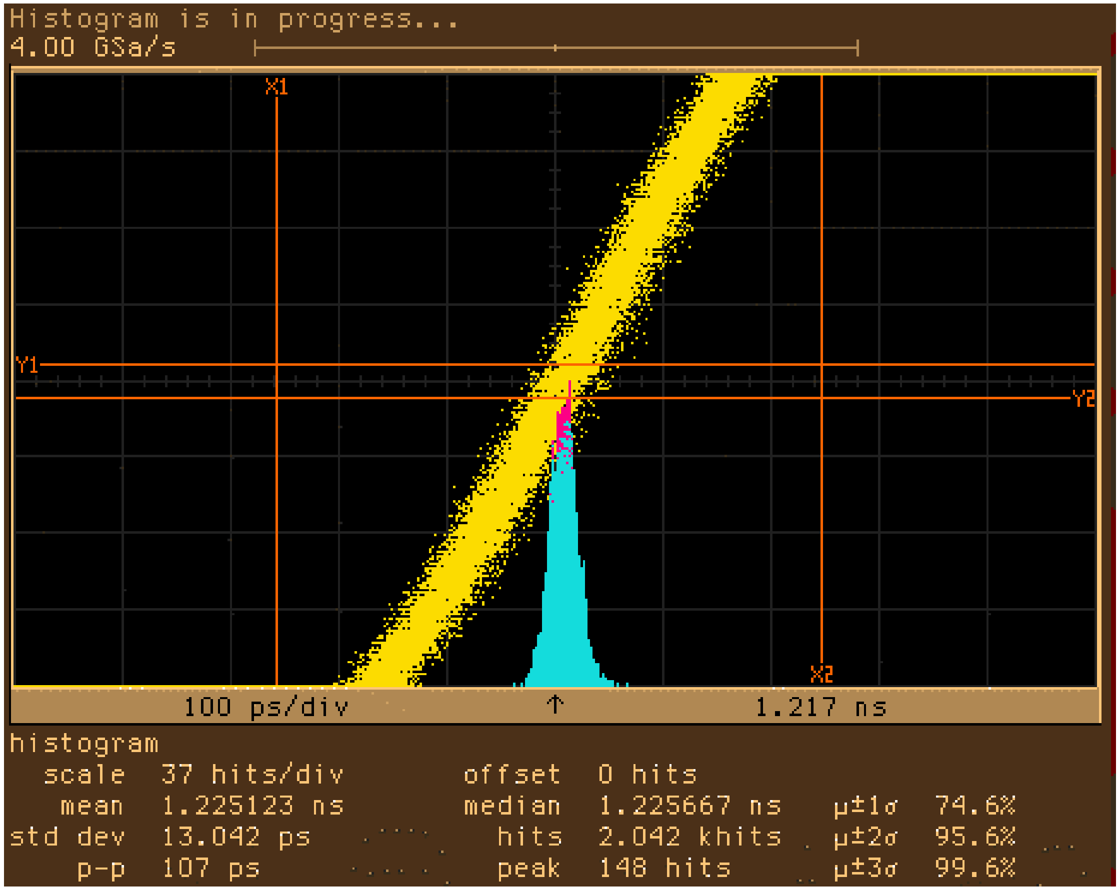
Outline

- Introduction
- Timing
- Signaling
- Results

Operating Range

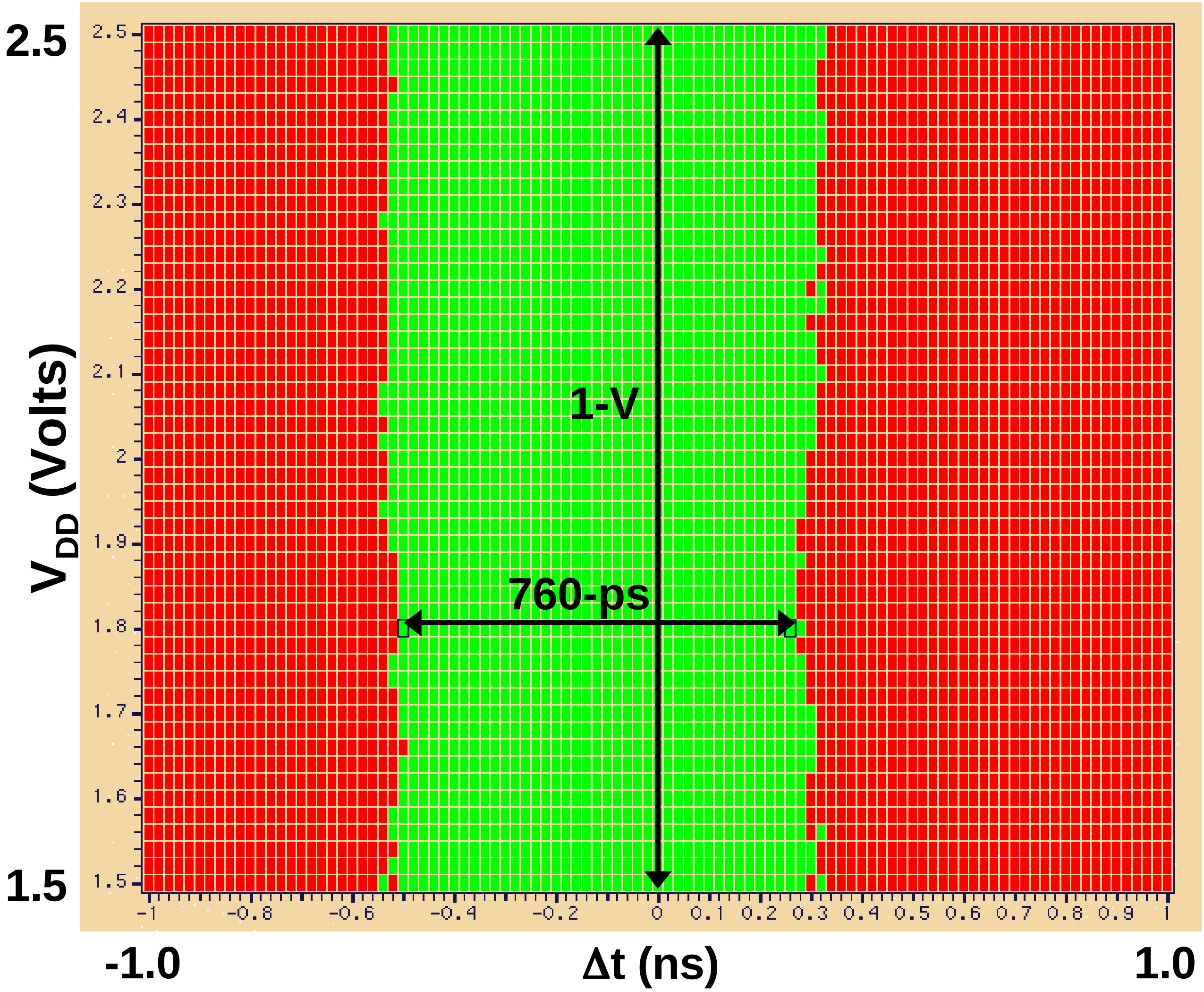


Measured DLL Jitter



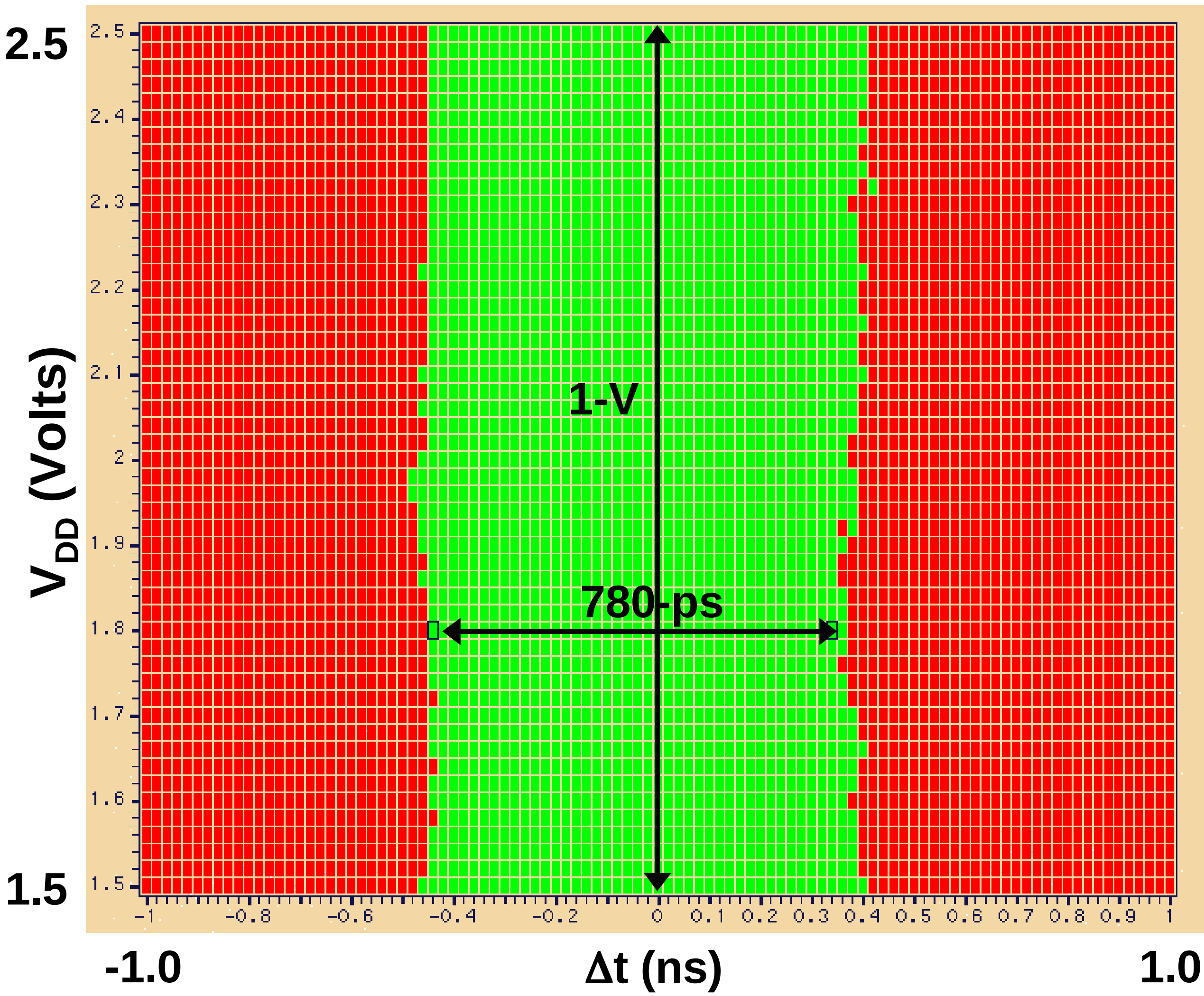
< 100-ps peak-peak with interface and core active

Uncalibrated Output Data-valid Window



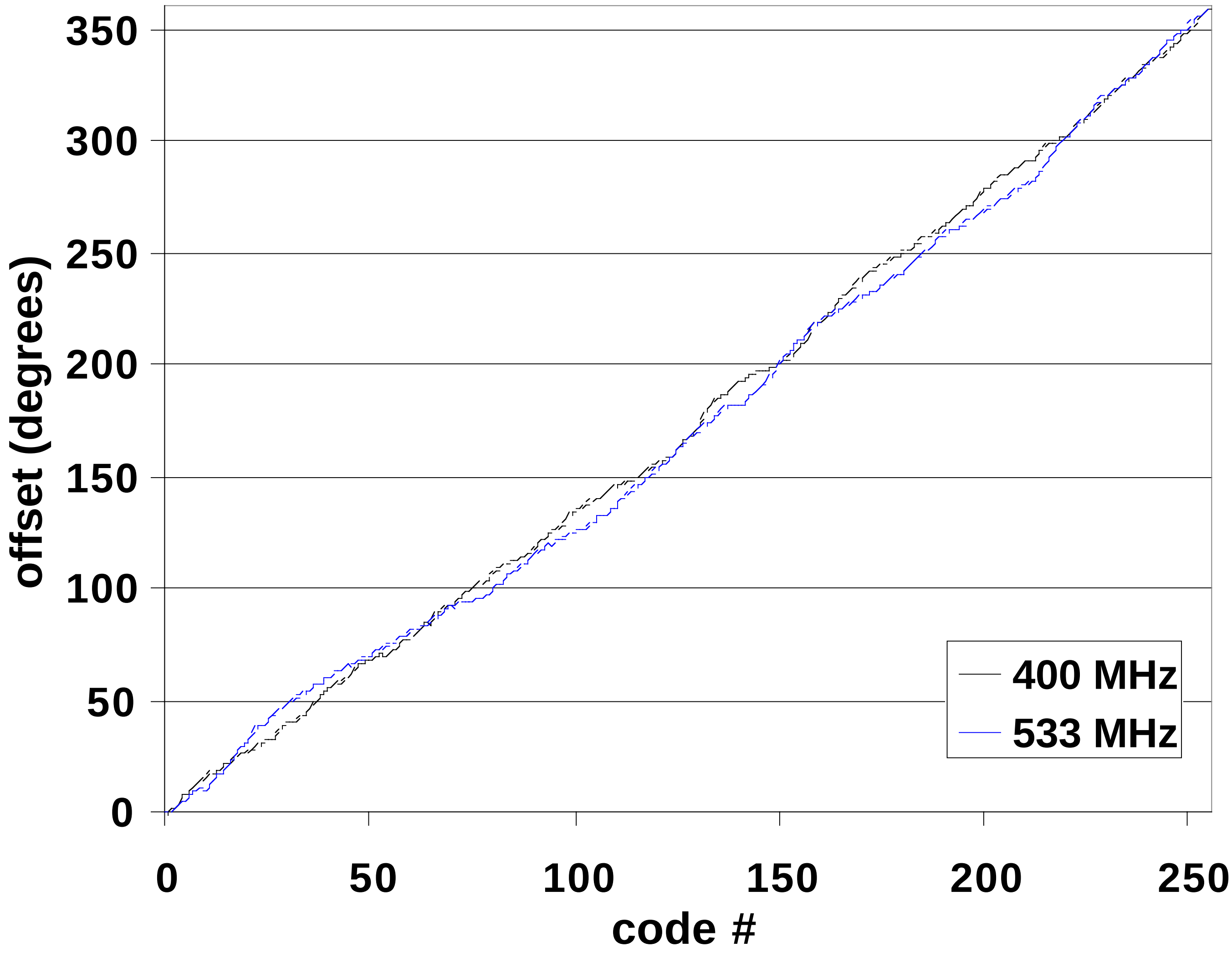
- T_{BIT} = 900-ps, T_{OFFS} = default → T_Q offset ~ 150-ps

Calibrated Output Data-valid Window



- T_{BIT} = 900-ps, calibrated T_{OFFS} → T_Q offset < 20-ps

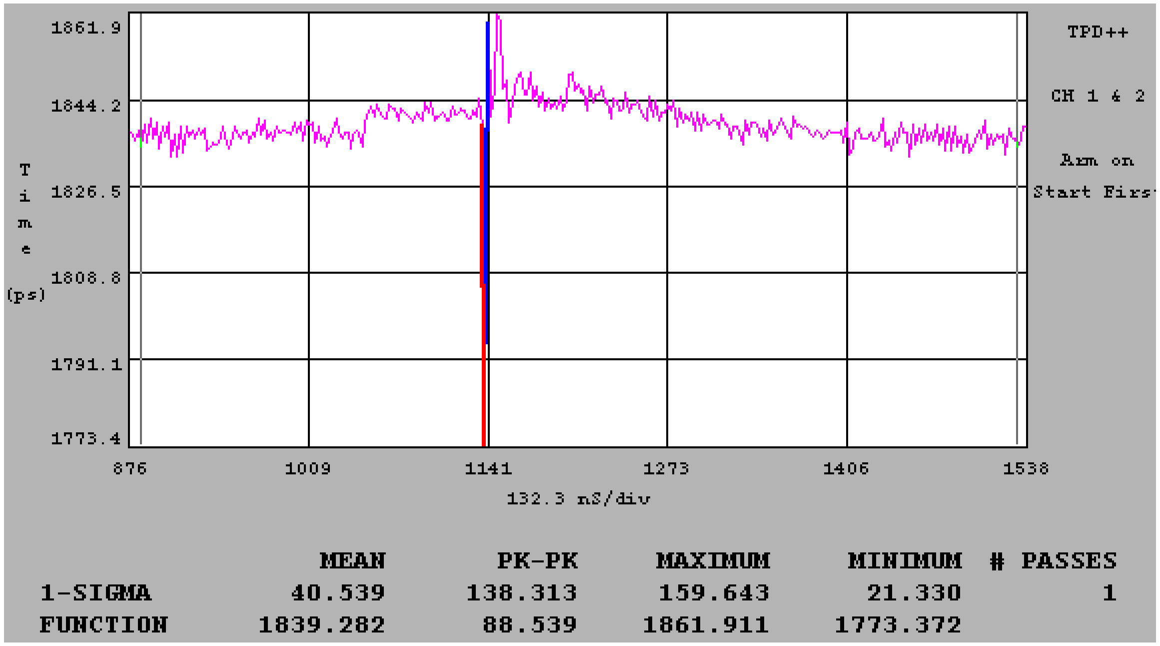
Measured Calibration Accuracy



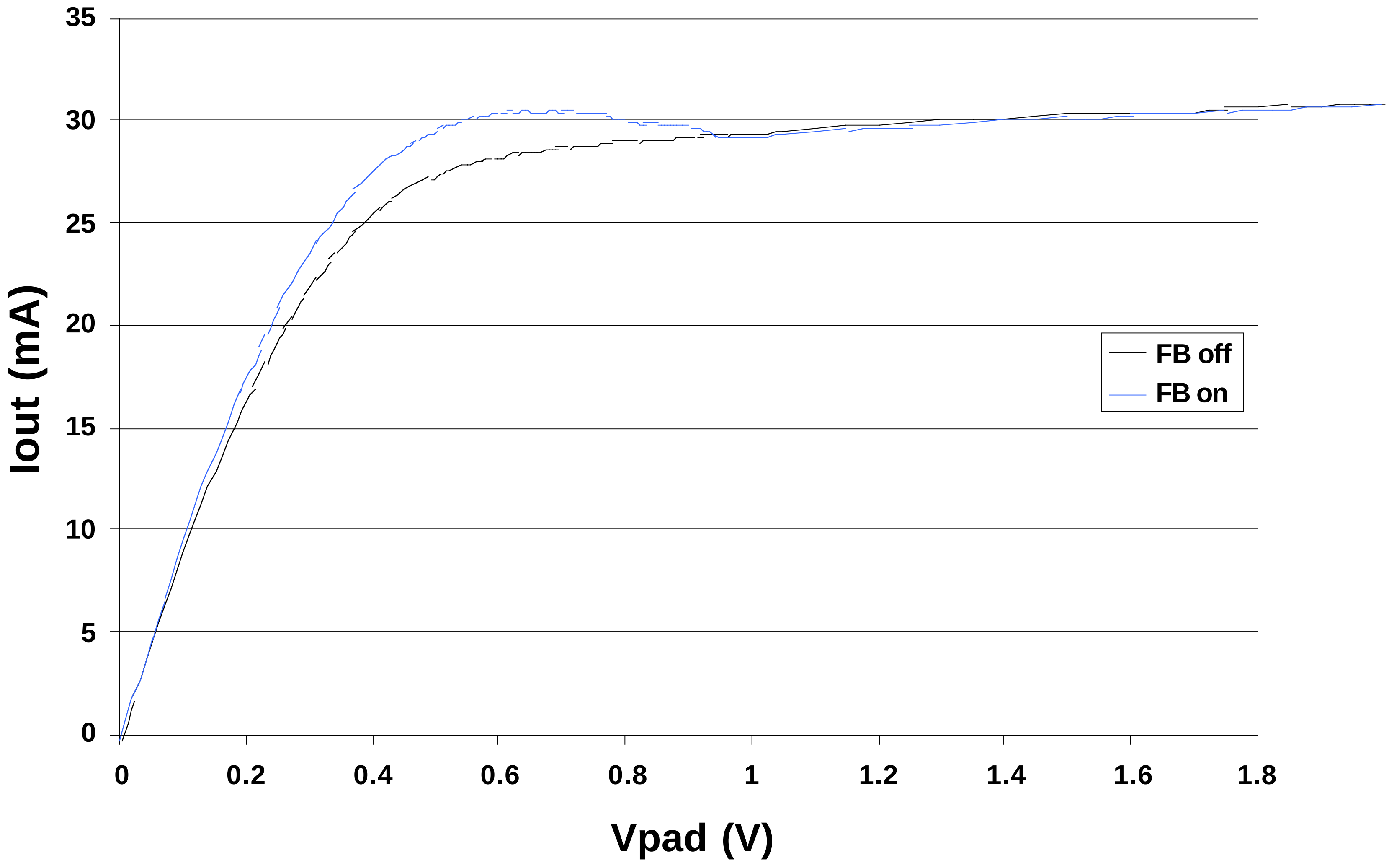
- DNL, INL < 2-LSB

RDRAM Power Modes

- DLL must go into low-power “nap” mode
  - IVDD < 4-mA
  - Restore clock phase within 80-ns
- Digital peripheral loop logic naturally holds state
- Hold state of core loop on 25-pF charge-pump capacitor



Measured Driver I-V Characteristics



Summary

- **Increasing memory interface bandwidth:**
  - → **Minimize both voltage and timing errors:**
  - **Voltage errors are systematic**
    - → **Compensated with new driver design**
  - **Timing Errors are unpredictable**
    - → **Compensated with “in-system” calibration**
- **Expect to see more digital “calibration” in high speed links:**
  - **Challenge is minimize overhead:**
    - **Area, Power, Yield..**
    - **System bring-up and ease of use..**