

# EE241B : Advanced Digital Circuits

## Lecture 21 – DVS II

### Borivoje Nikolić



**April 8, NY Times: Computers Already Learn From Us. But Can They Teach Themselves?**



Scientists are exploring approaches that would help machines develop their own sort of common sense

Features Pieter Abbeel and Sergey Levine from UC Berkeley

# Announcements

- Assignment 4 due next Friday.
- Reading
  - T. Burd, et al, JSSC, Nov 2000.



# Outline

- **Module 5**
  - Dynamic voltage and frequency scaling

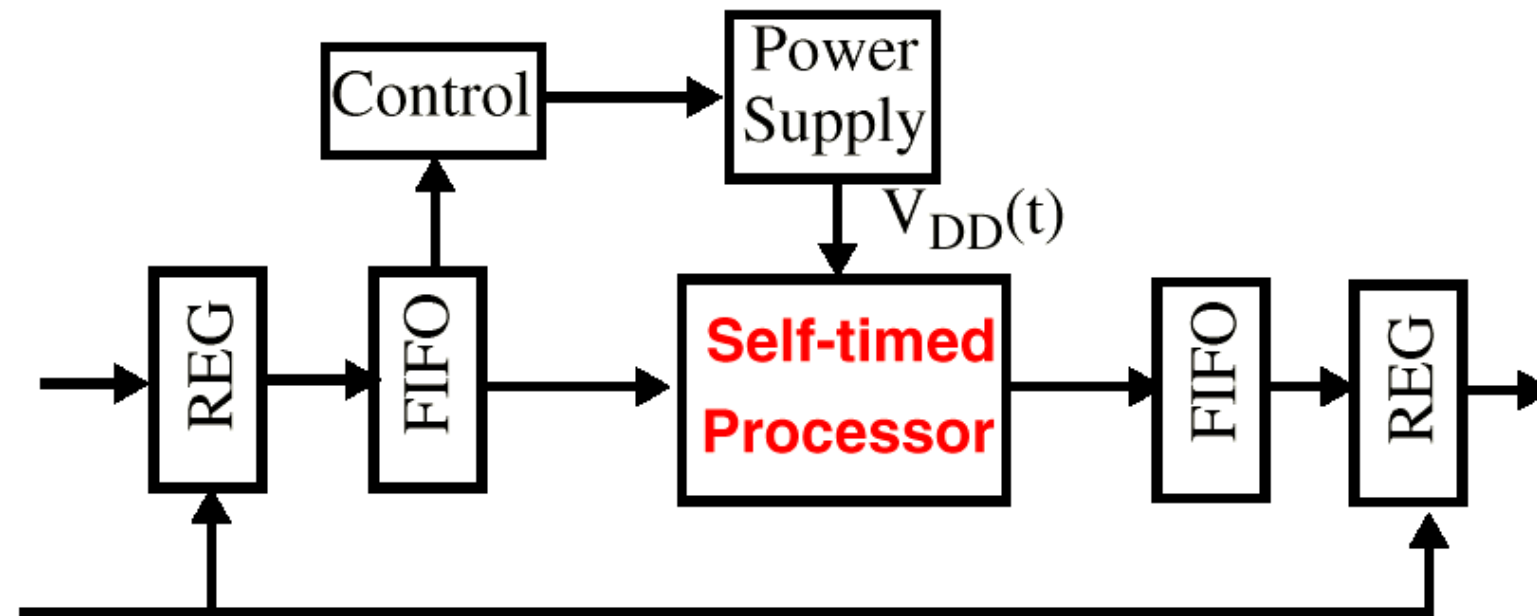


## 5.F Dynamic Voltage Scaling

# Power /Energy Optimization Space

	Constant Throughput/Latency	Variable Throughput/Latency	
Energy	Design Time	Sleep Mode	Run Time
Active	Logic design Scaled $V_{DD}$ Trans. sizing Multi- $V_{DD}$	Clock gating	DFS, DVS
Leakage	Stack effects Trans sizing Scaling $V_{DD}$ + Multi- $V_{Th}$	Sleep T's Multi- $V_{DD}$ Variable $V_{Th}$ + Input control	DVS Variable $V_{Th}$

# Adaptive Supply Voltages

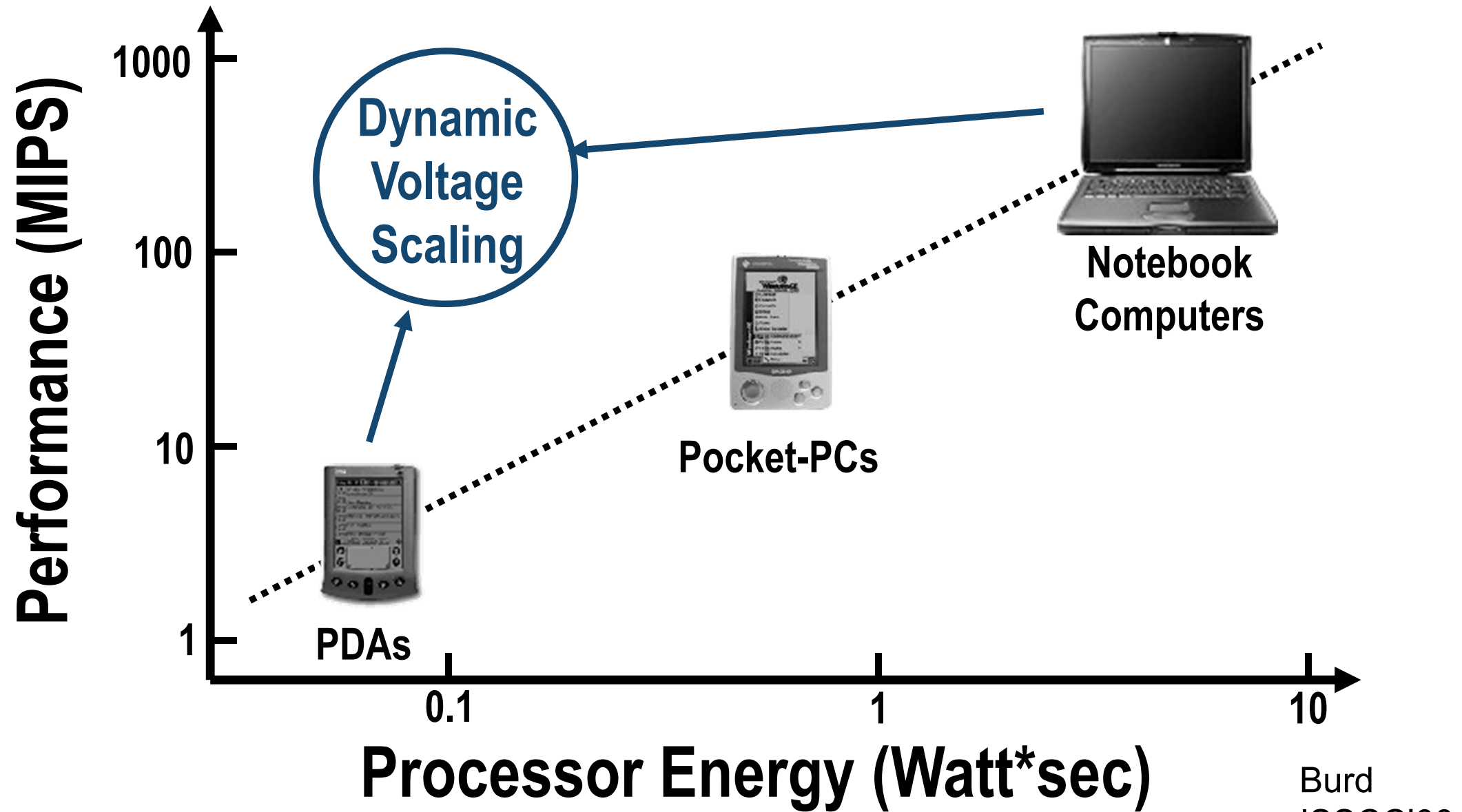


***Exploit Data Dependent Computation Times To Vary the Supply***

**from [Nielsen94]**

***(IEEE Transactions on VLSI Systems)***

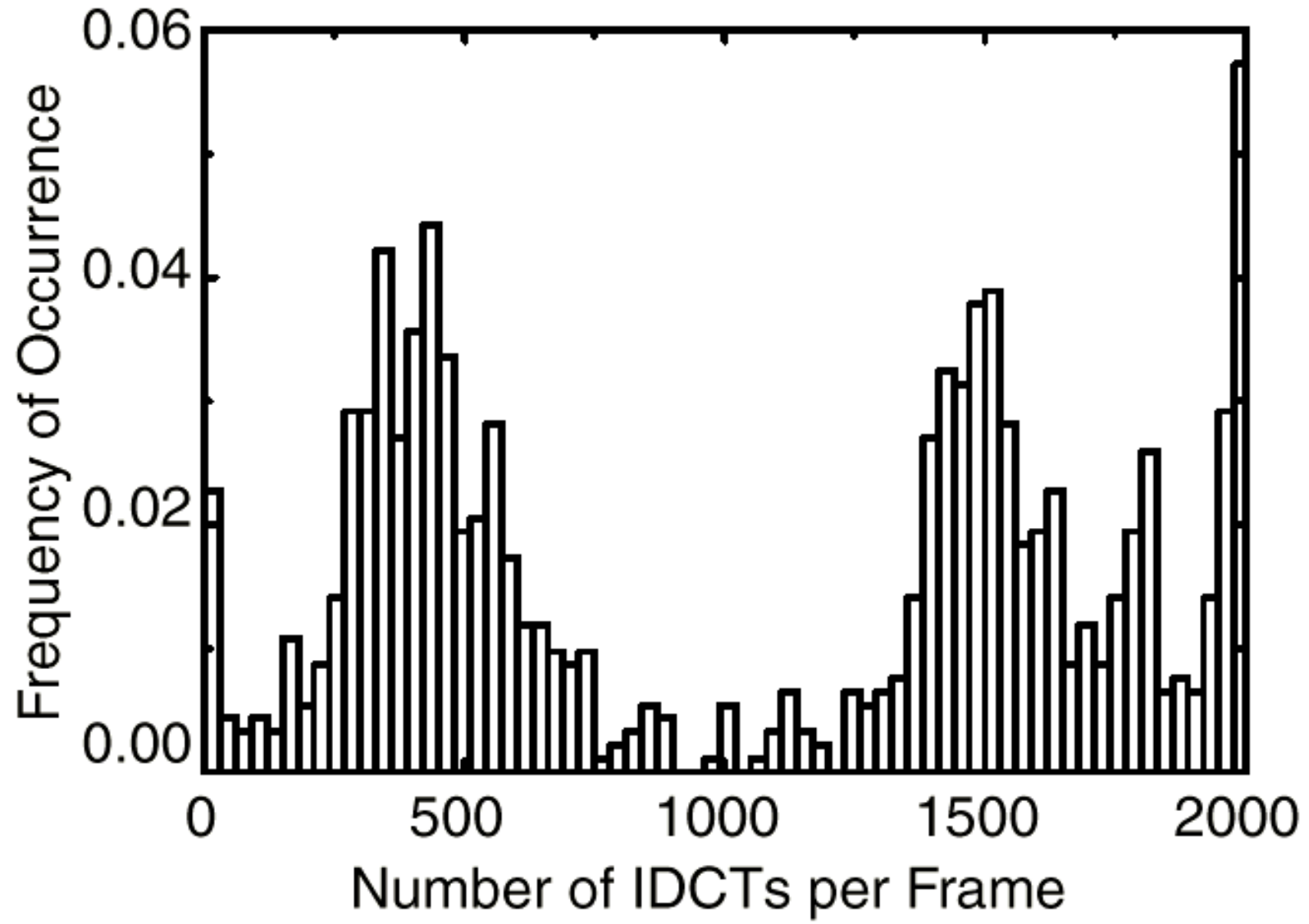
# Processors for Portable Devices



- Eliminate performance ↔ energy trade-off

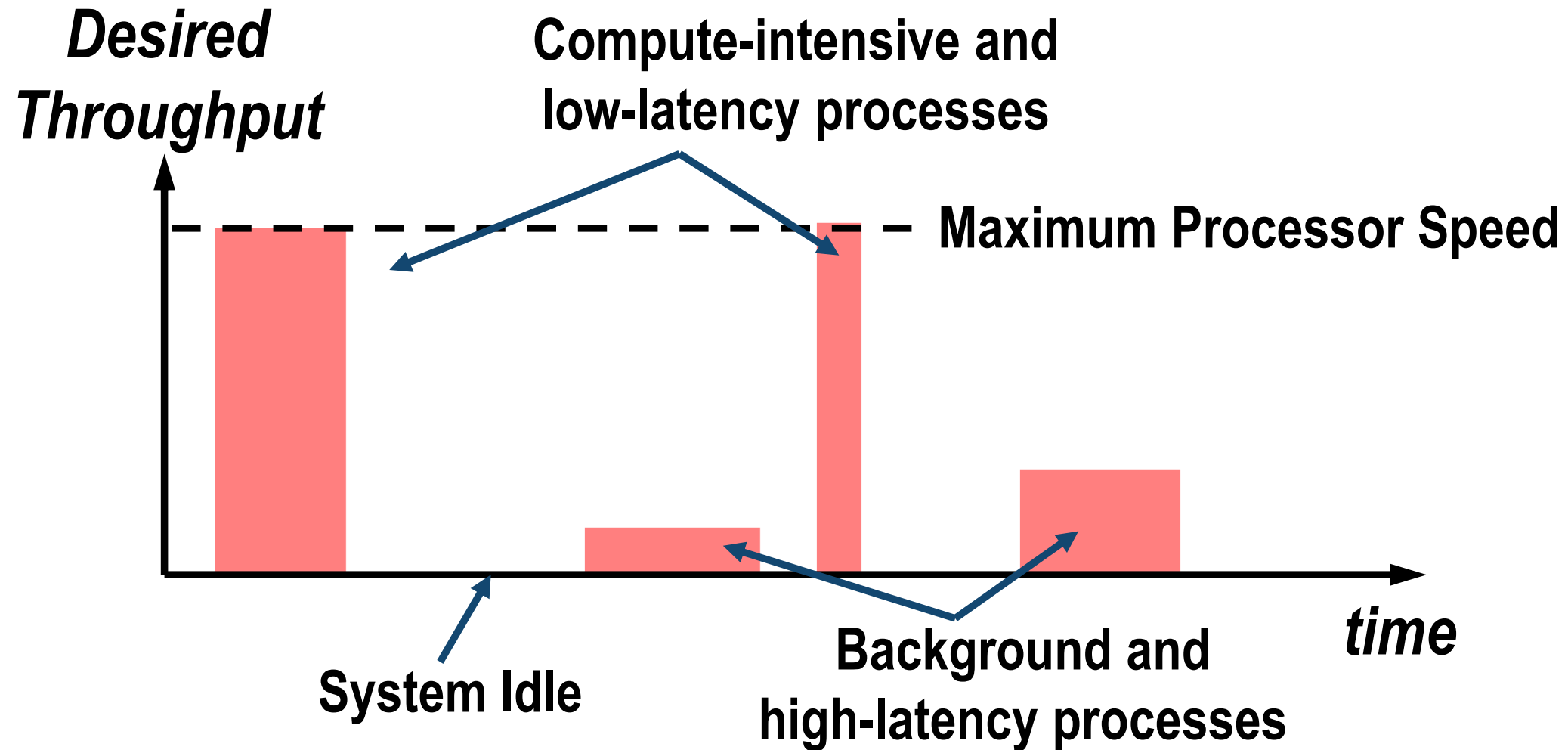
Burd  
ISSCC'00

# Typical MPEG IDCT Histogram





# Processor Usage Model

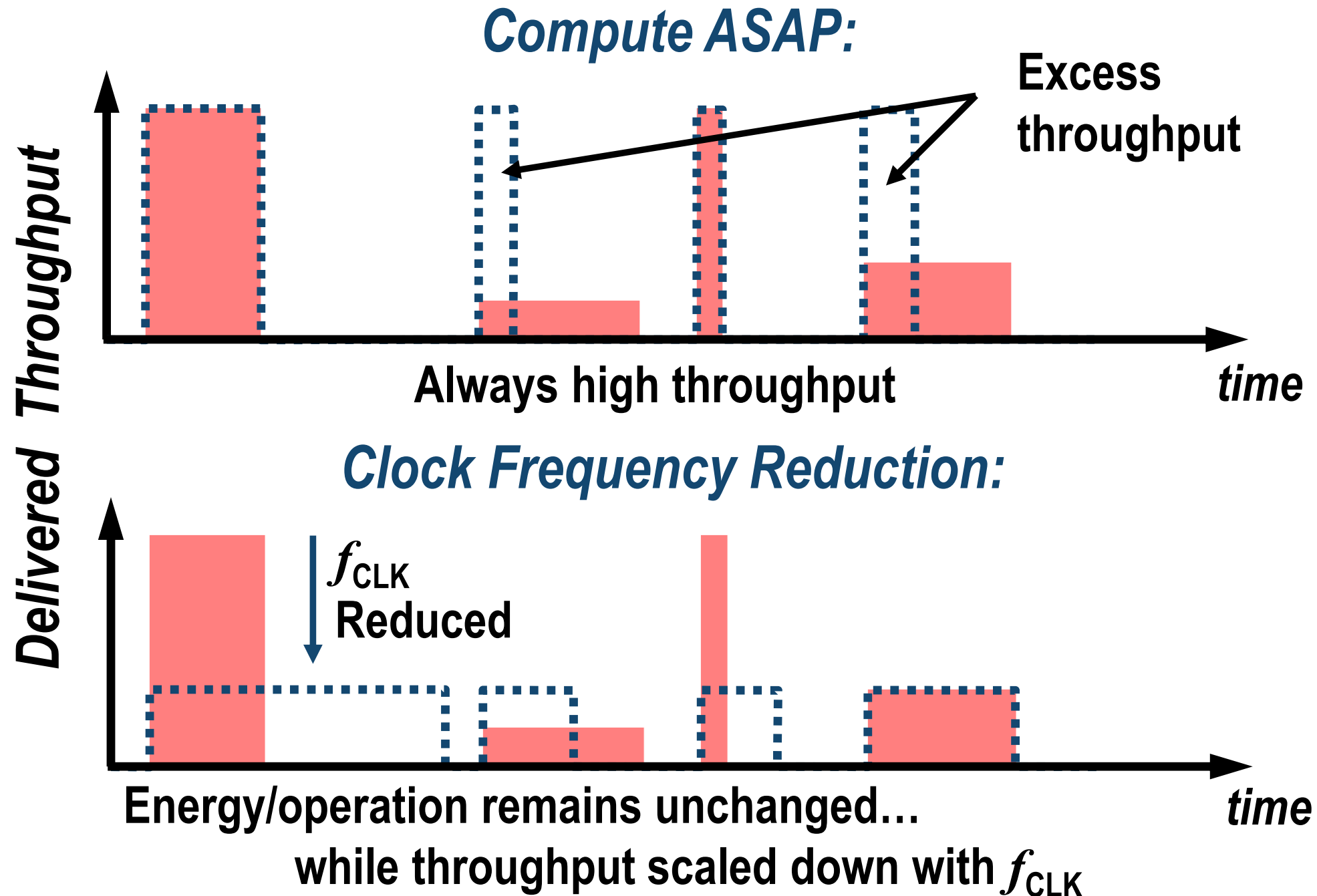


## System Optimizations:

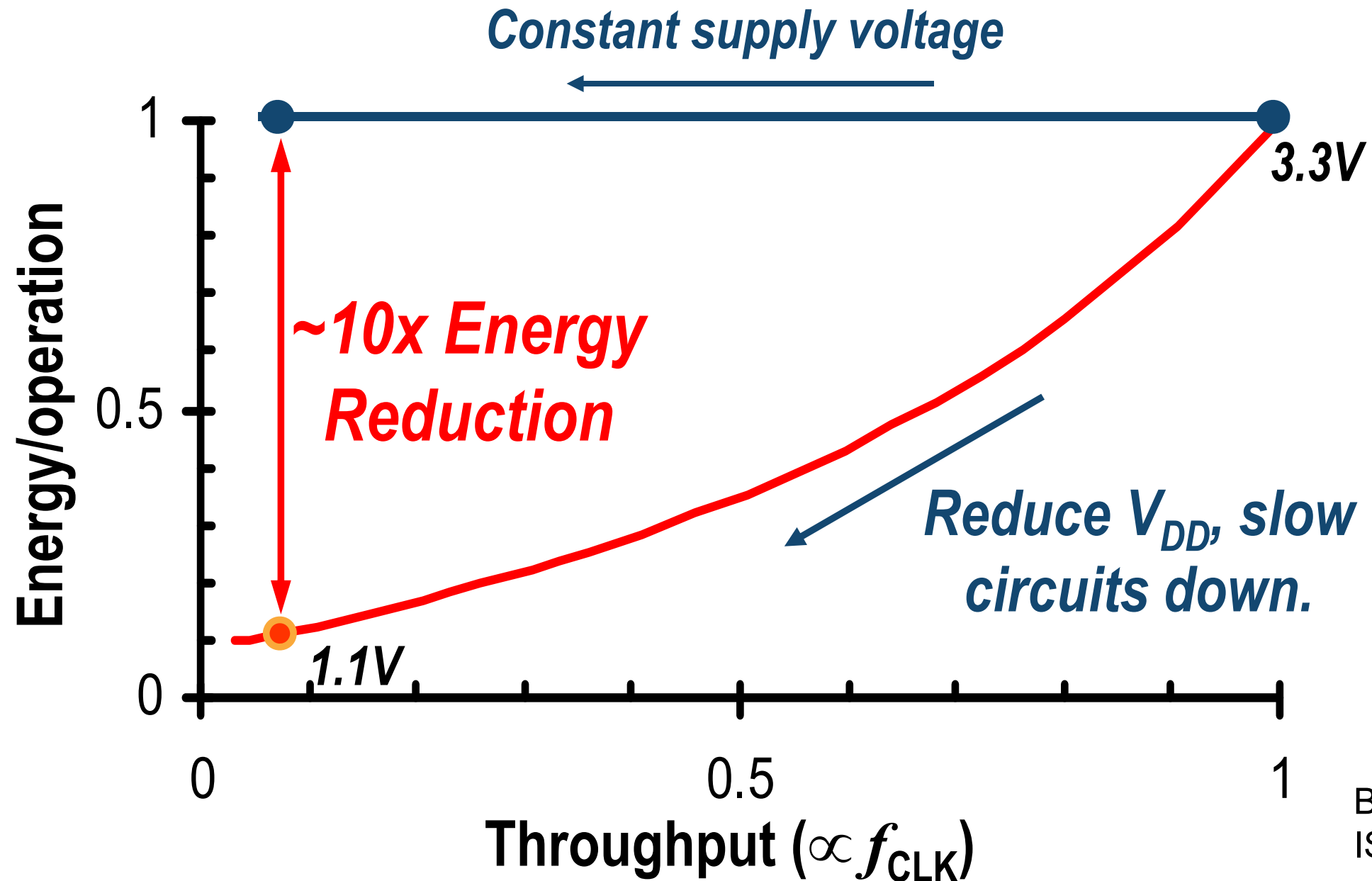
- **Maximize Peak Throughput**
- **Minimize Average Energy/operation**

Burd  
ISSCC'00

# Common Design Approaches (Fixed VDD)

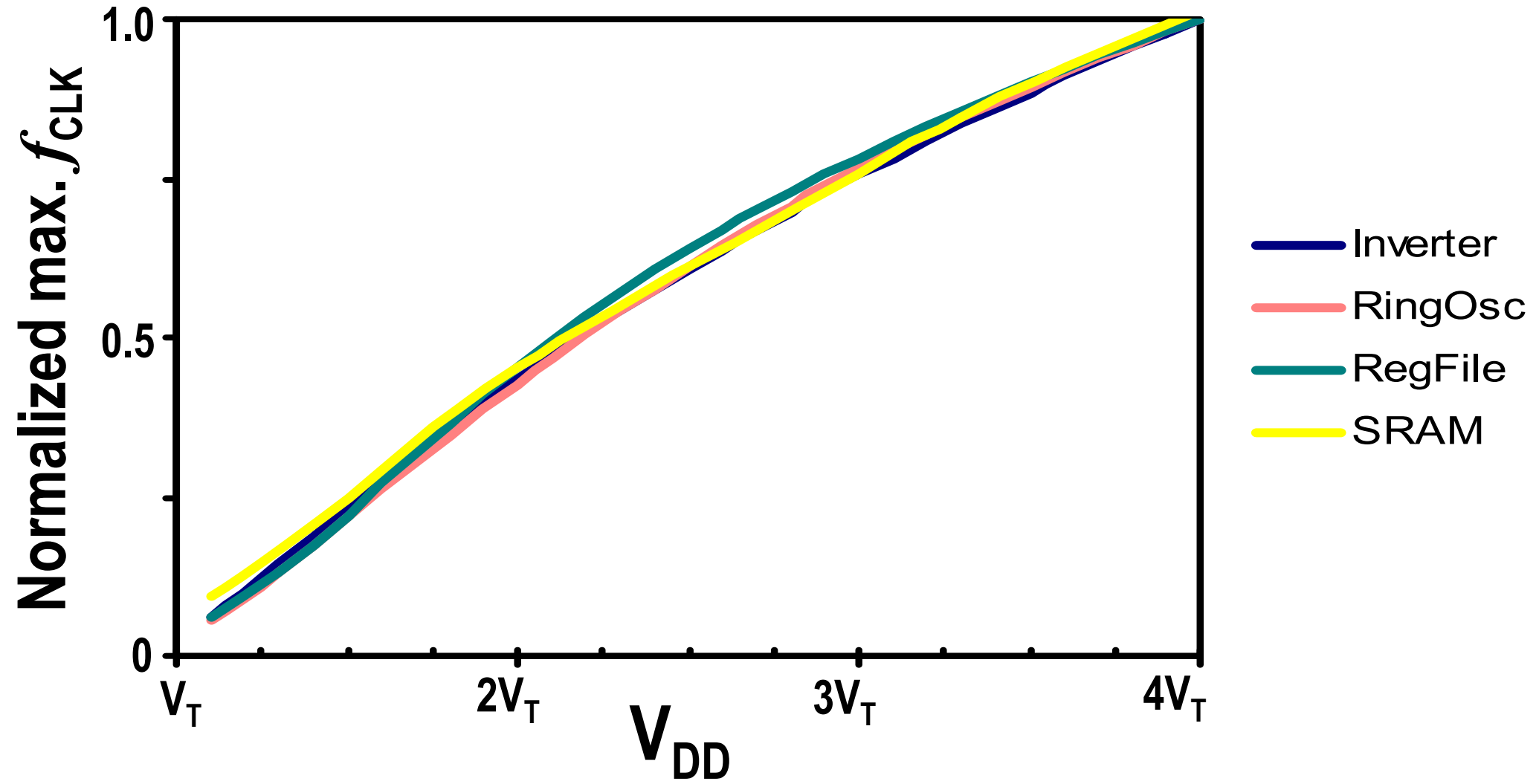


# Scale $V_{DD}$ with Clock Frequency



Burd  
ISSCC'00

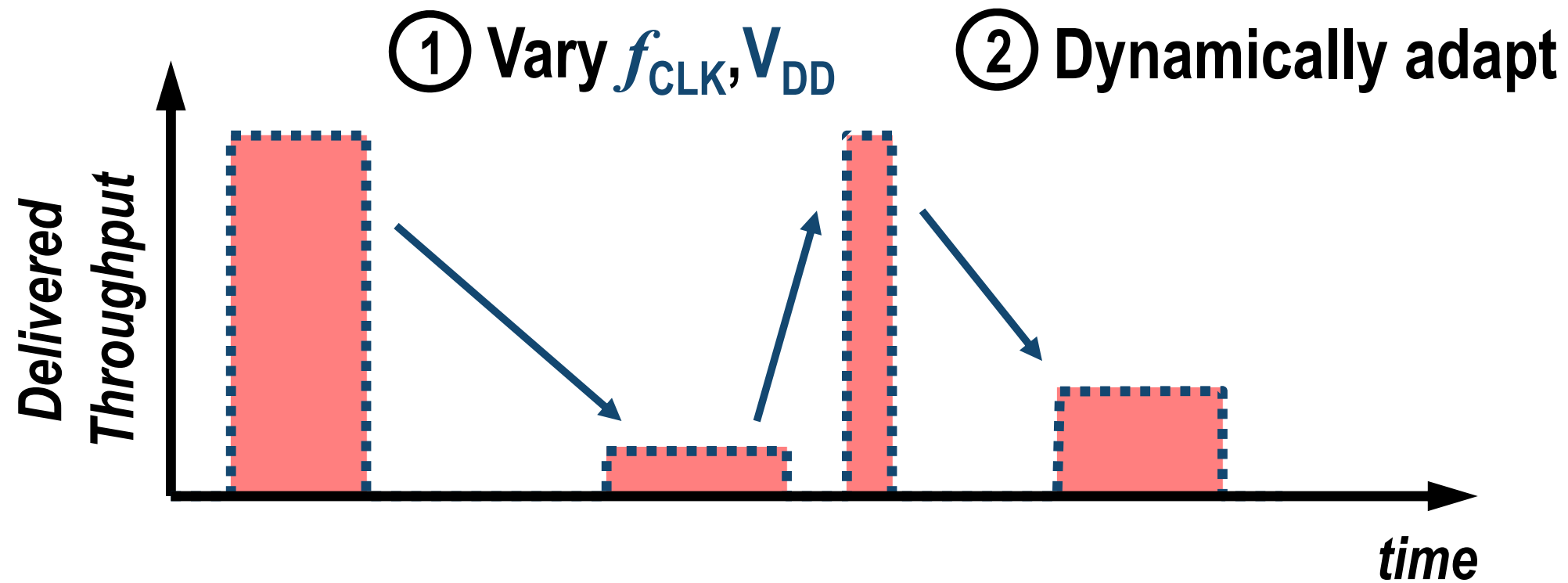
# CMOS Circuits Track Over $V_{DD}$



← Delay tracks within +/- 10% →

Burd  
ISSCC'00

# Dynamic Voltage Scaling (DVS)



- Dynamically scale energy/operation with throughput.
- Always minimize speed → minimize average energy/operation.
- Extend battery life up to 10x with the exact same hardware!

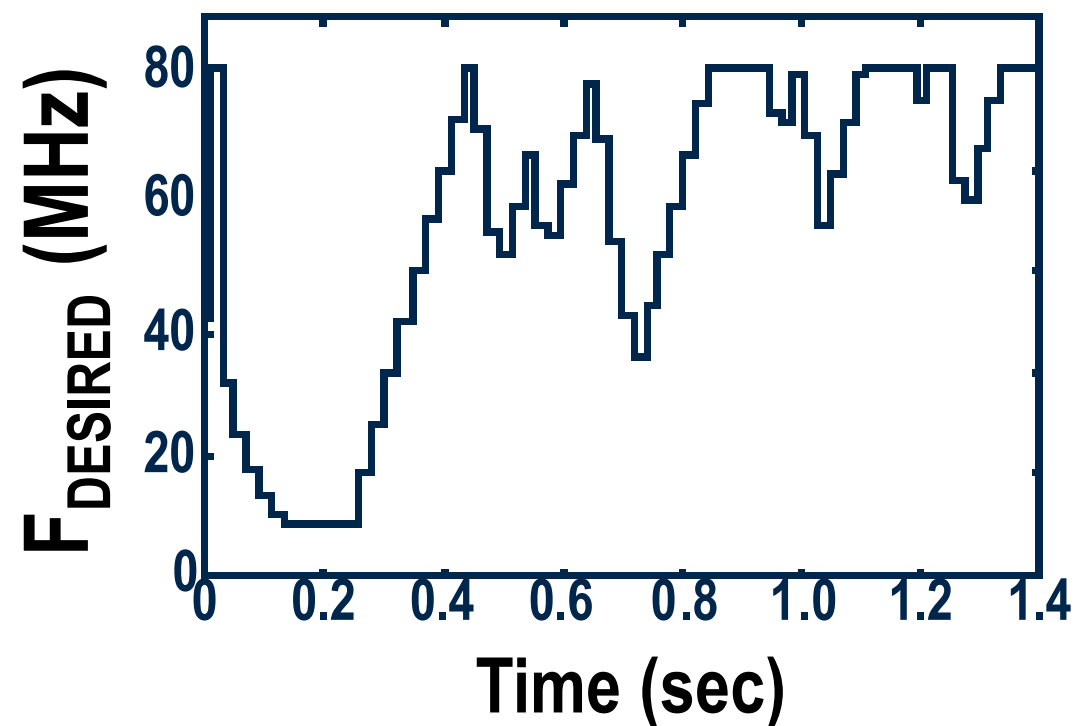
Burd  
ISSCC'00

# Operating System Sets Processor Speed

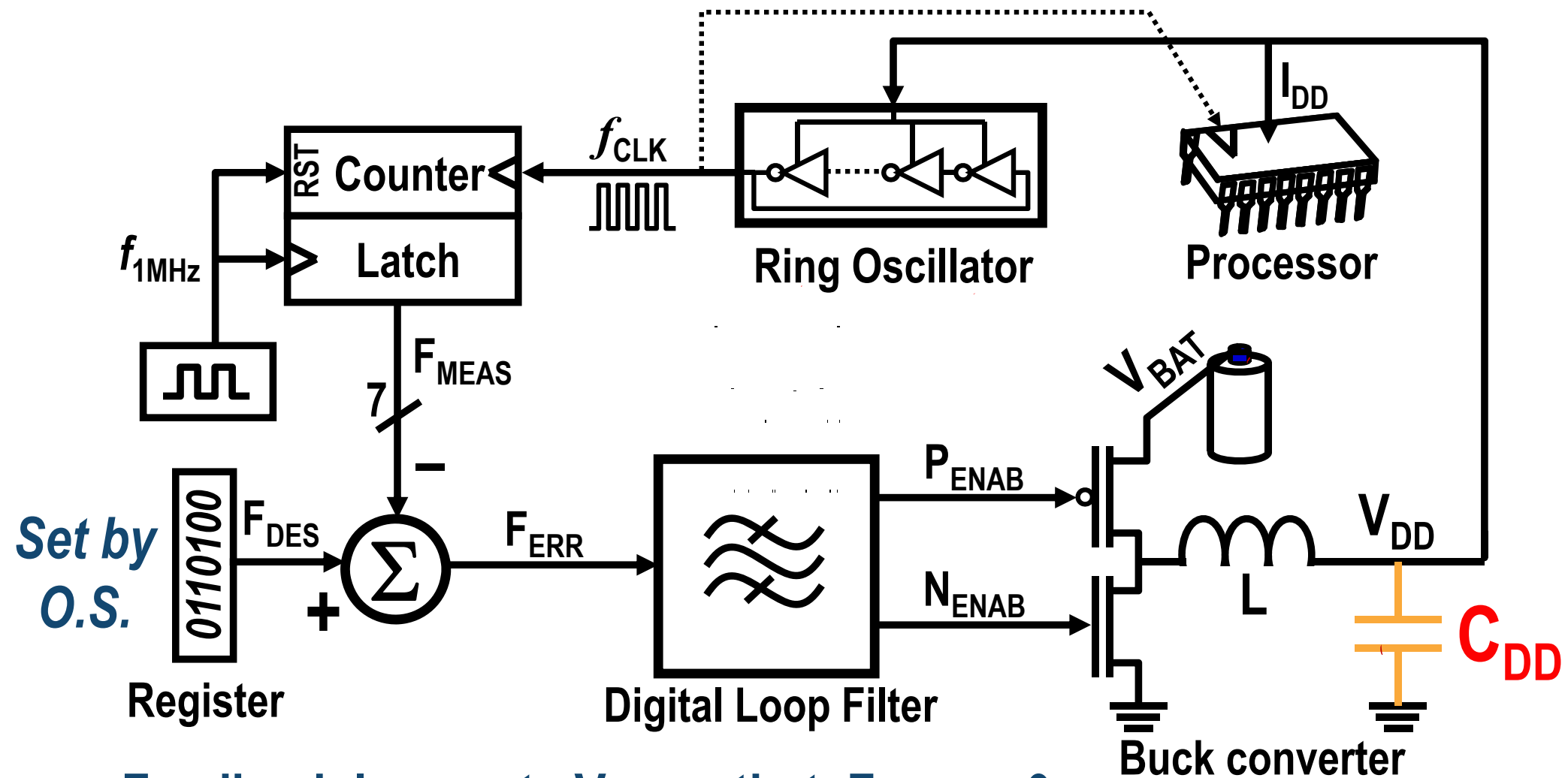
- DVS requires a *voltage scheduler (VS)*.
- VS predicts workload to estimate CPU cycles.
- Applications supply completion deadlines.

$$\frac{\text{CPU cycles}}{\Delta \text{ time}} = F_{\text{DESIRED}}$$

## Processor Speed (MPEG)



# Converter Loop Sets $V_{DD}$ , $f_{CLK}$



- Feedback loop sets  $V_{DD}$  so that  $F_{ERR} \rightarrow 0$ .
- Ring oscillator delay-matched to CPU critical paths.
- Custom loop implementation  $\rightarrow$  Can optimize  $C_{DD}$ .

Burd  
ISSCC'00

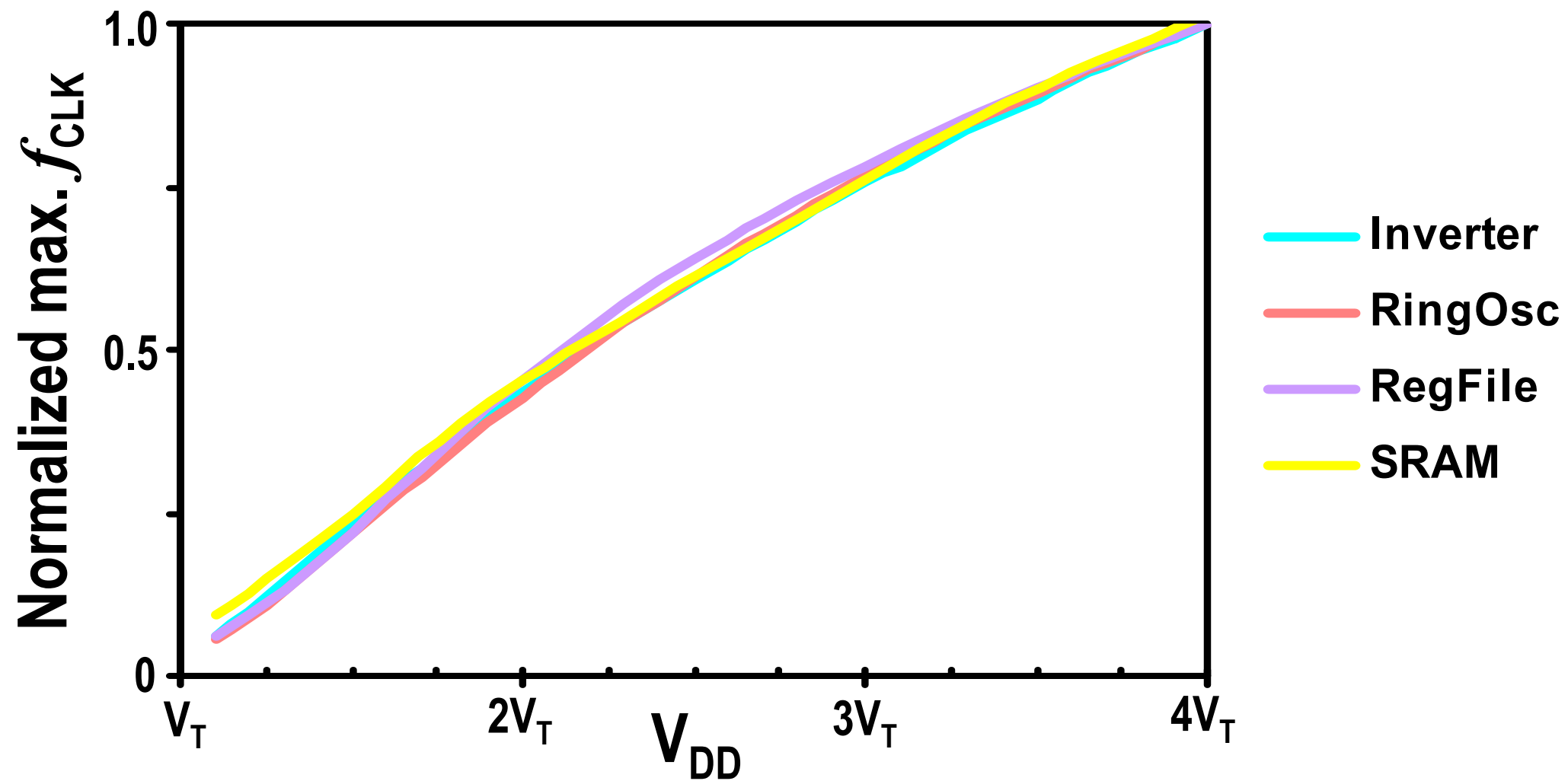
# Design Over Wide Range of Voltages

- **Circuit design constraints. (Functional verification)**
- **Circuit delay variation. (Timing verification)**
- **Noise margin reduction. (Power grid, coupling)**
- **Delay sensitivity. (Local power distribution)**

**Design verification complexity similar to  
high-performance processor design @ fixed  $V_{DD}$**



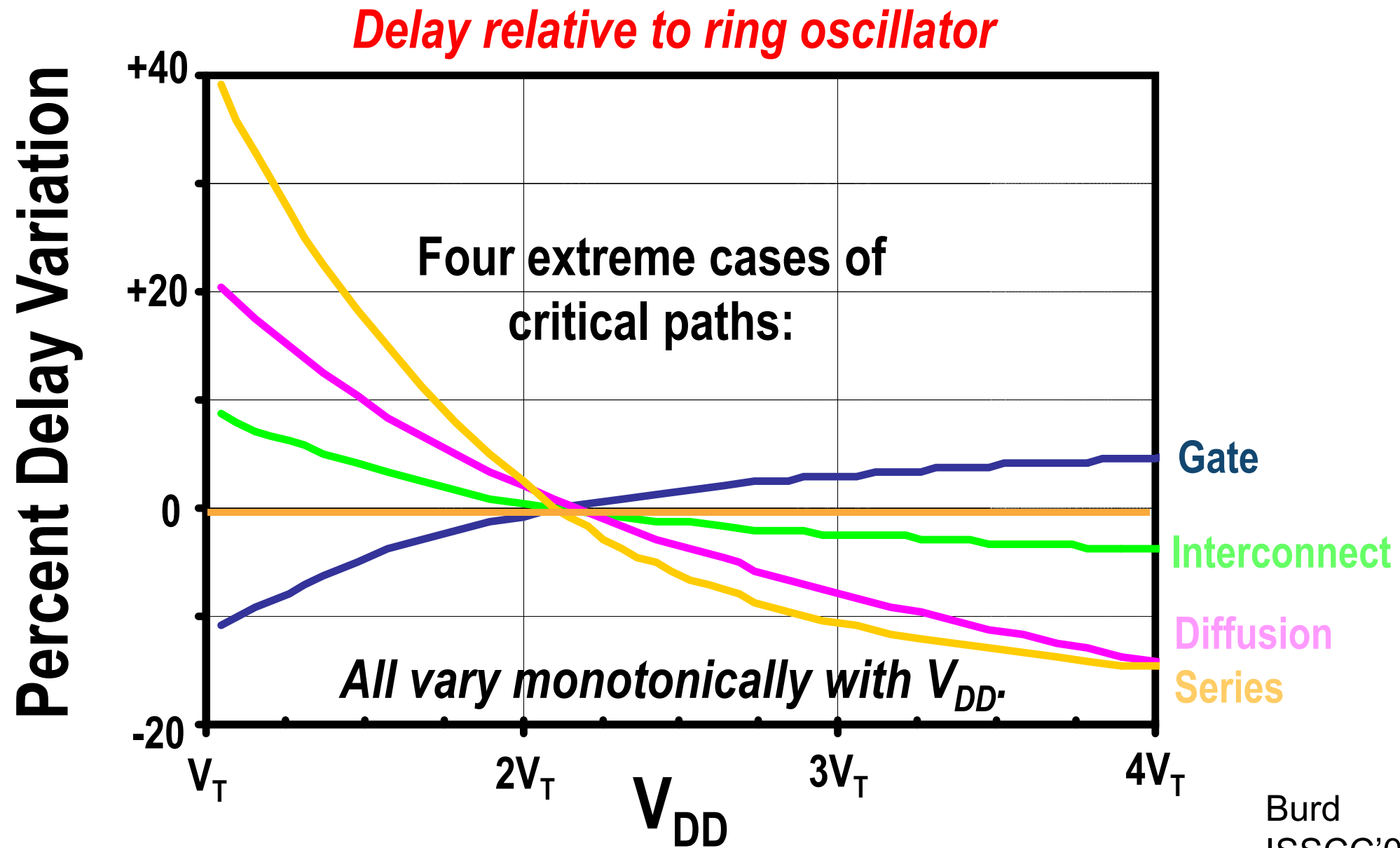
# Delay Variation & Circuit Constraints



- Cannot use NMOS pass gates – fails for  $V_{DD} < 2V_T$ .
- Functional verification only needed at one  $V_{DD}$  value.

Burd  
ISSCC'00

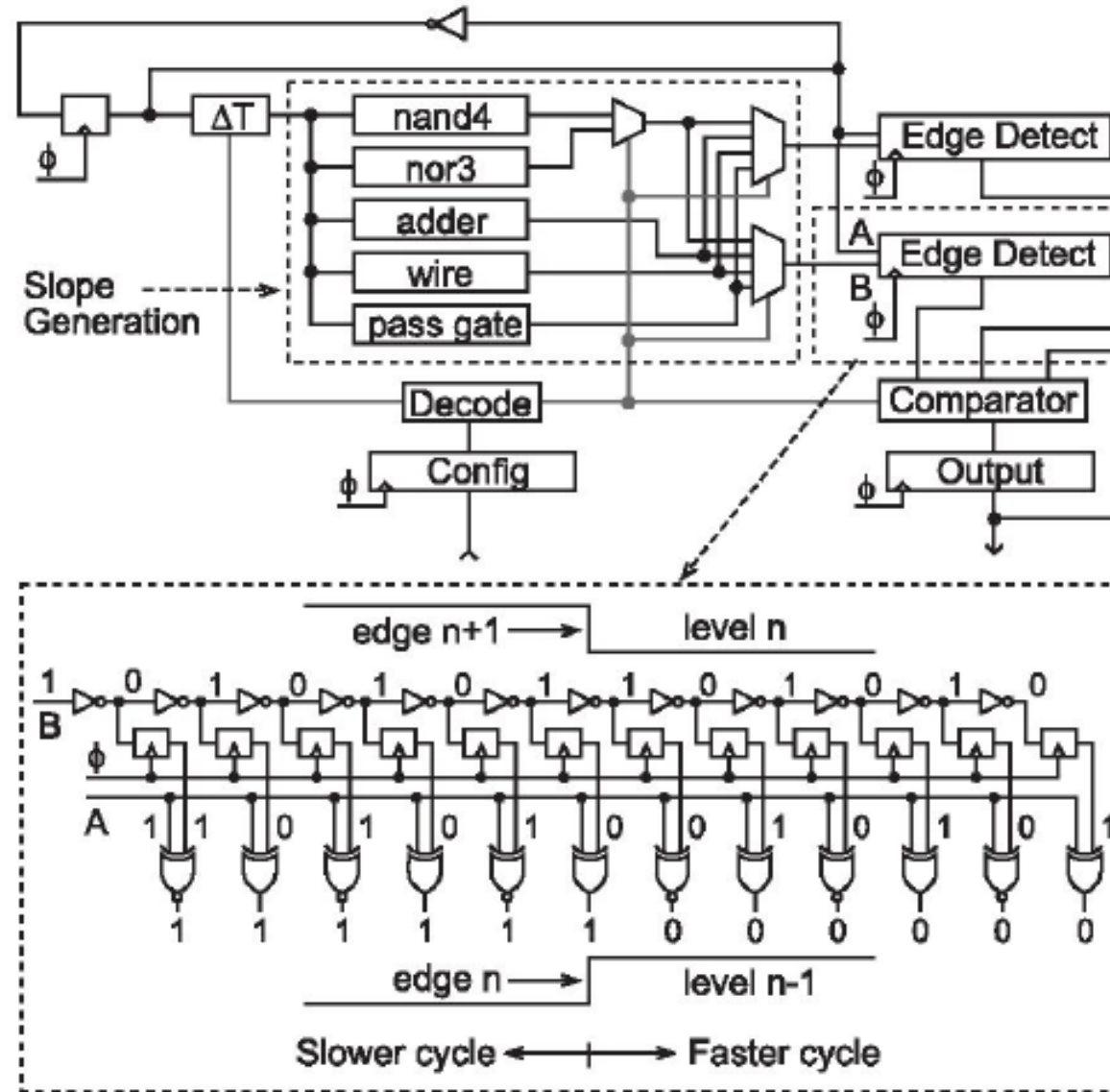
# Relative Delay Variation



Burd  
ISSCC'00

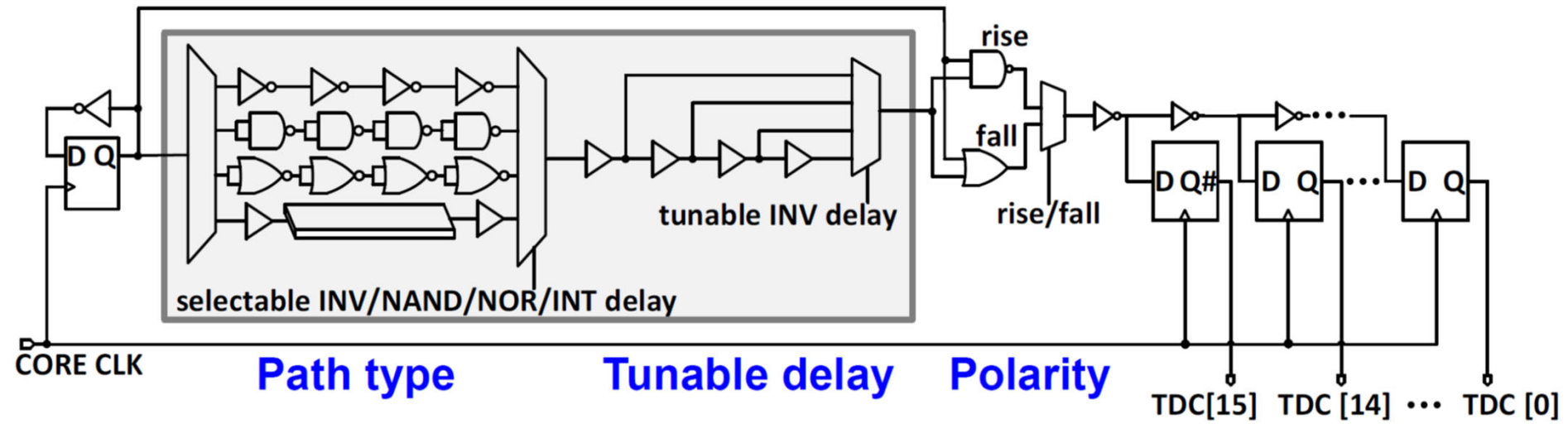
- Timing verification only needed at min. & max.  $V_{DD}$ .

# Multiple Path Tracking



A. Drake, ISSCC'07

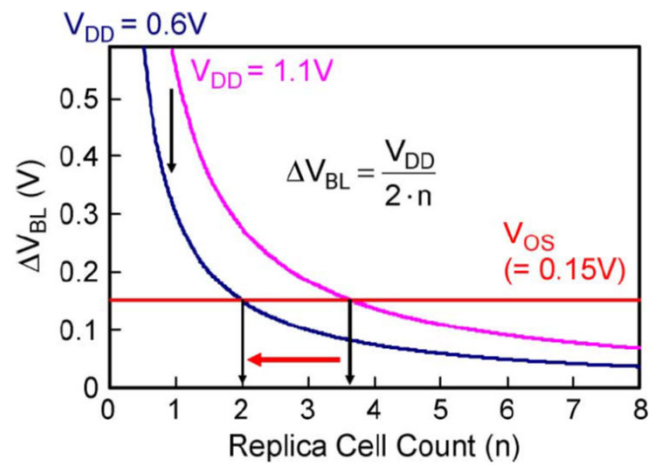
# Multiple Path Tracking



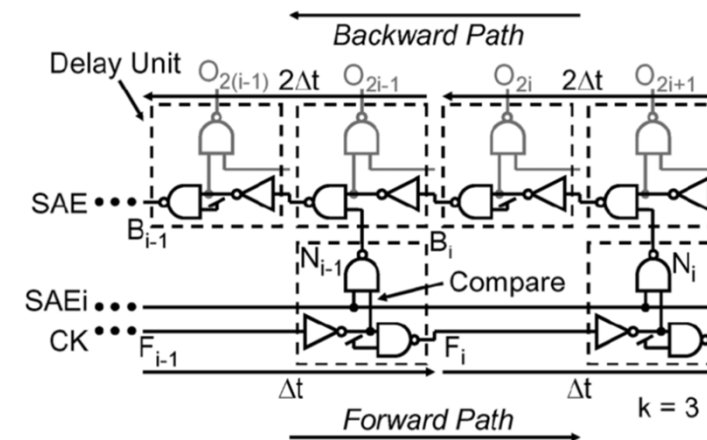
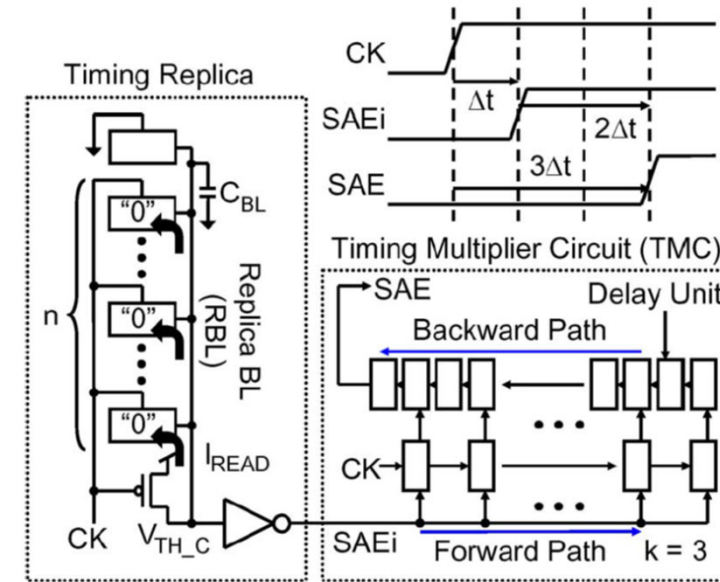
Cho, ISSCC'16

# Tracking with SRAM in Critical Path

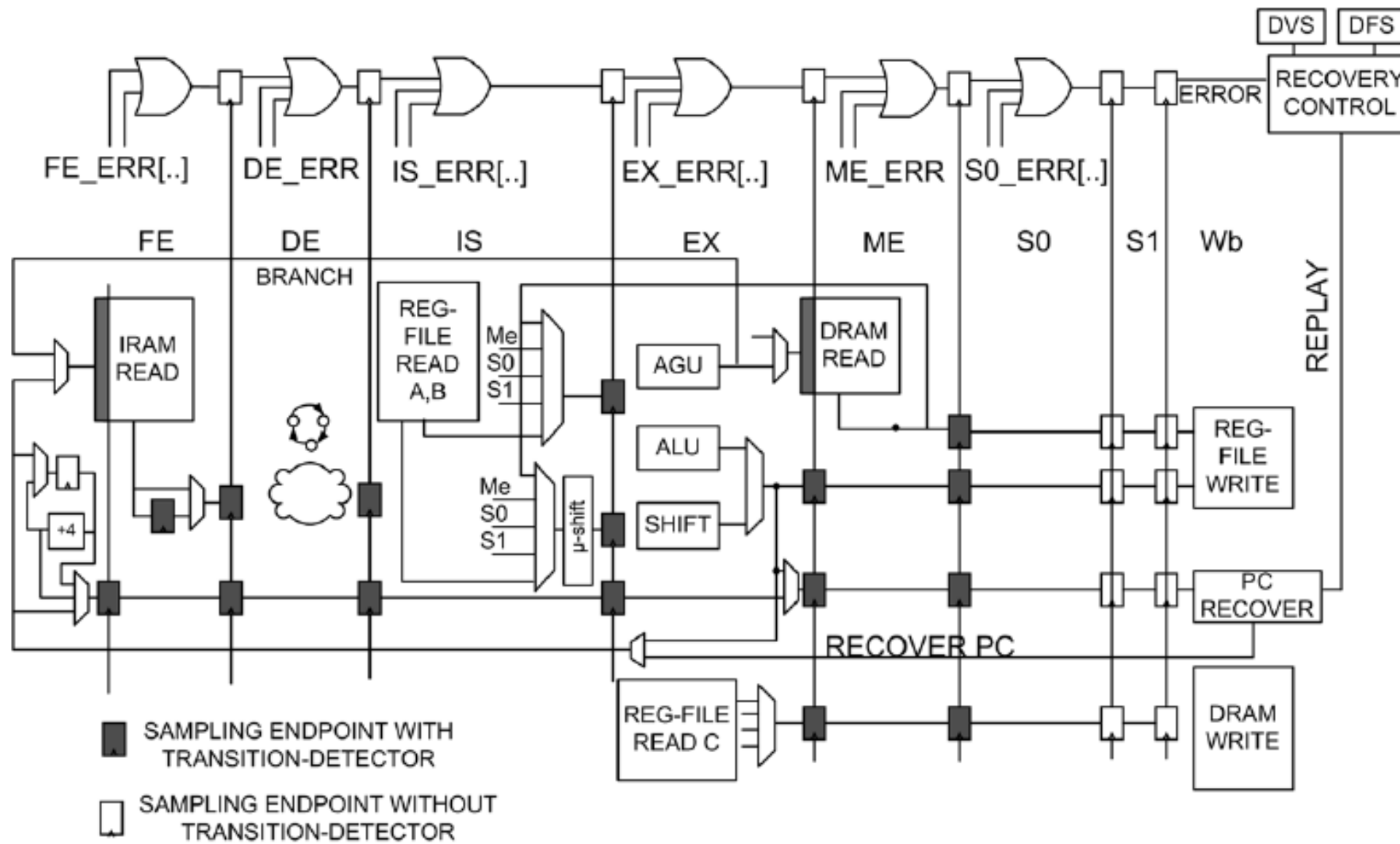
## ➤ Mismatch between logic and SRAM



## ➤ SRAM multiplicative replica



# Alternative: Error Detection



Bull, ISSCC'2010

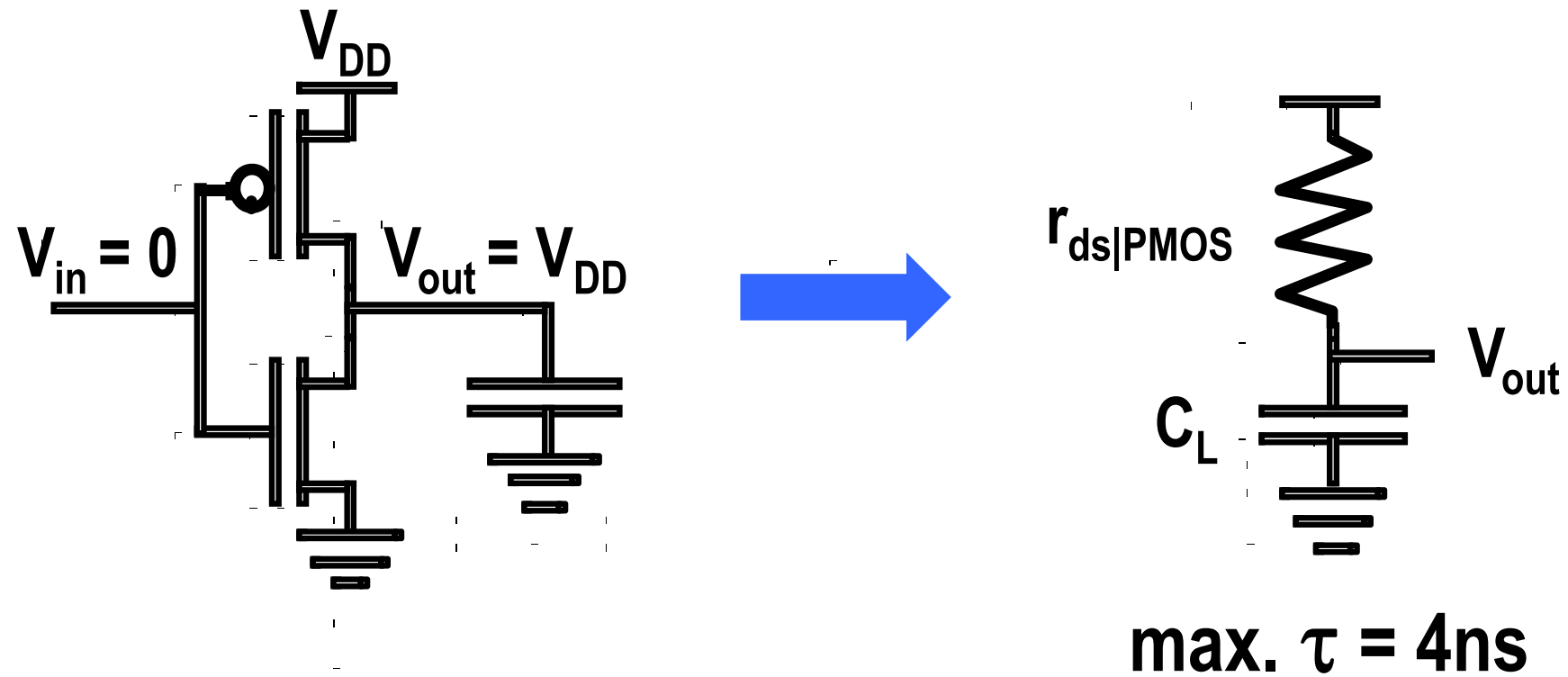
# Design for Dynamically Varying VDD

- **Static CMOS logic.**
- **Ring oscillator.**
- **Dynamic logic (& tri-state busses).**
- **Sense amp (& memory cell).**

**Max. allowed  $|dV_{DD}/dt| \rightarrow \text{Min. } C_{DD} = 100\text{nF} (0.6\mu\text{m})$**

**Circuits continue to properly operate as  $V_{DD}$  changes**

# Static CMOS Logic



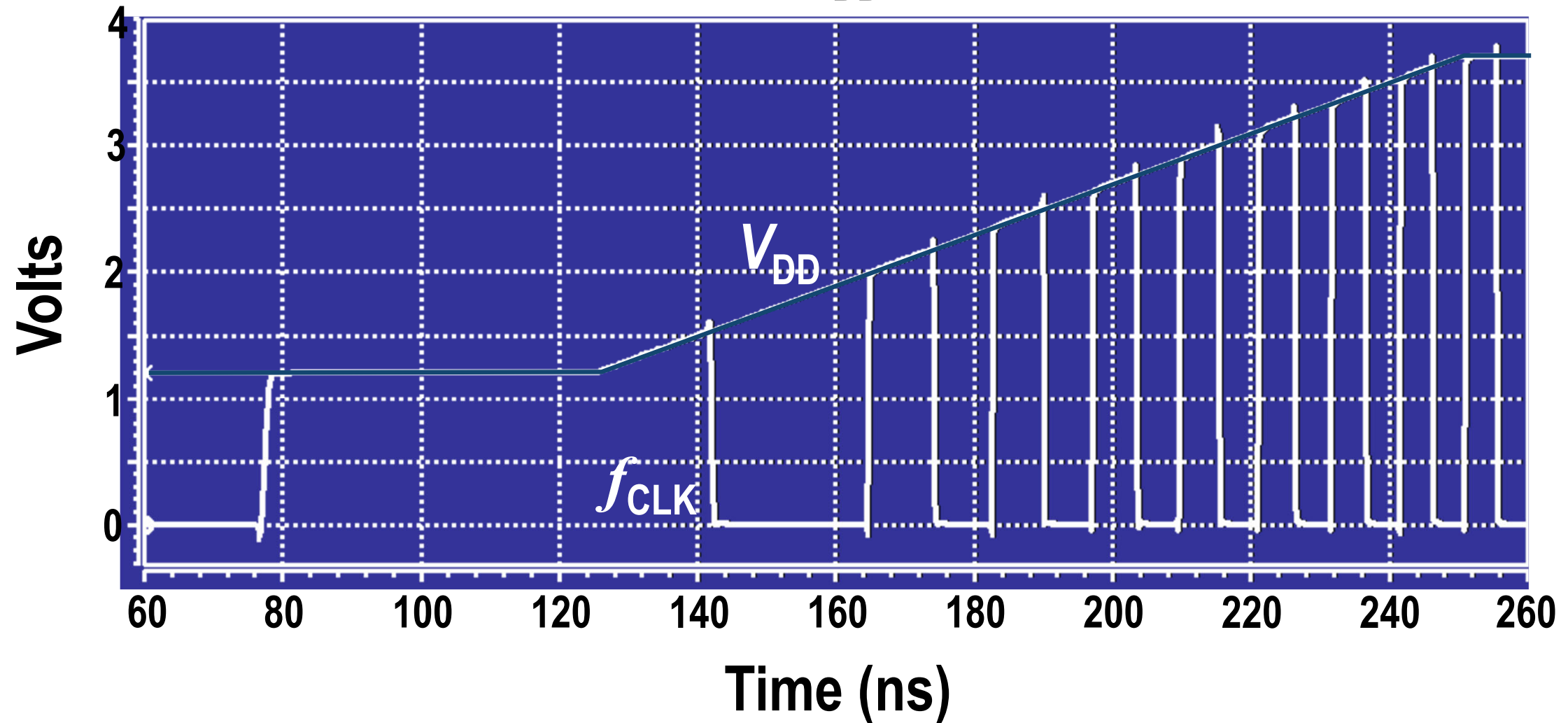
**$0.6\mu\text{m CMOS: } |dV_{DD}/dt| < 200\text{V}/\mu\text{s}$**

- **Static CMOS robustly operates with varying  $V_{DD}$ .**



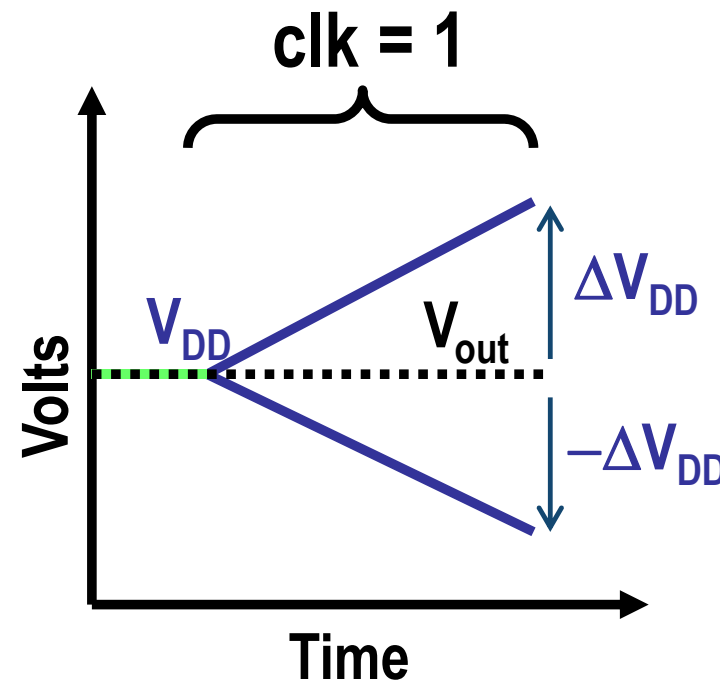
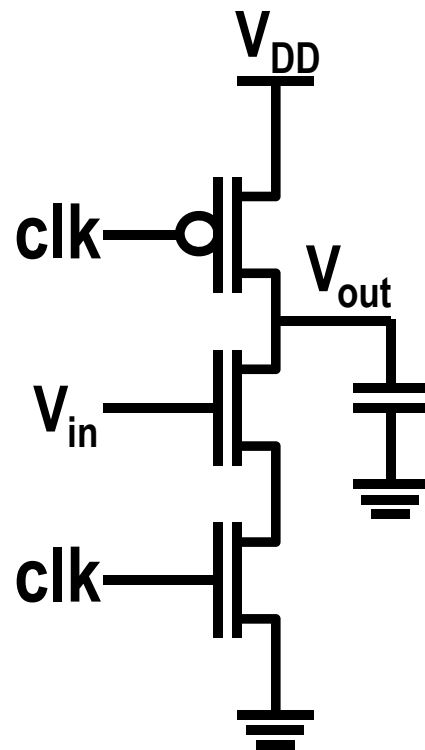
# Ring Oscillator

*Simulated with  $dV_{DD}/dt = 20V/\mu s$*



- Output  $f_{CLK}$  instantaneously adapts to new  $V_{DD}$ .

# Dynamic Logic



## Errors

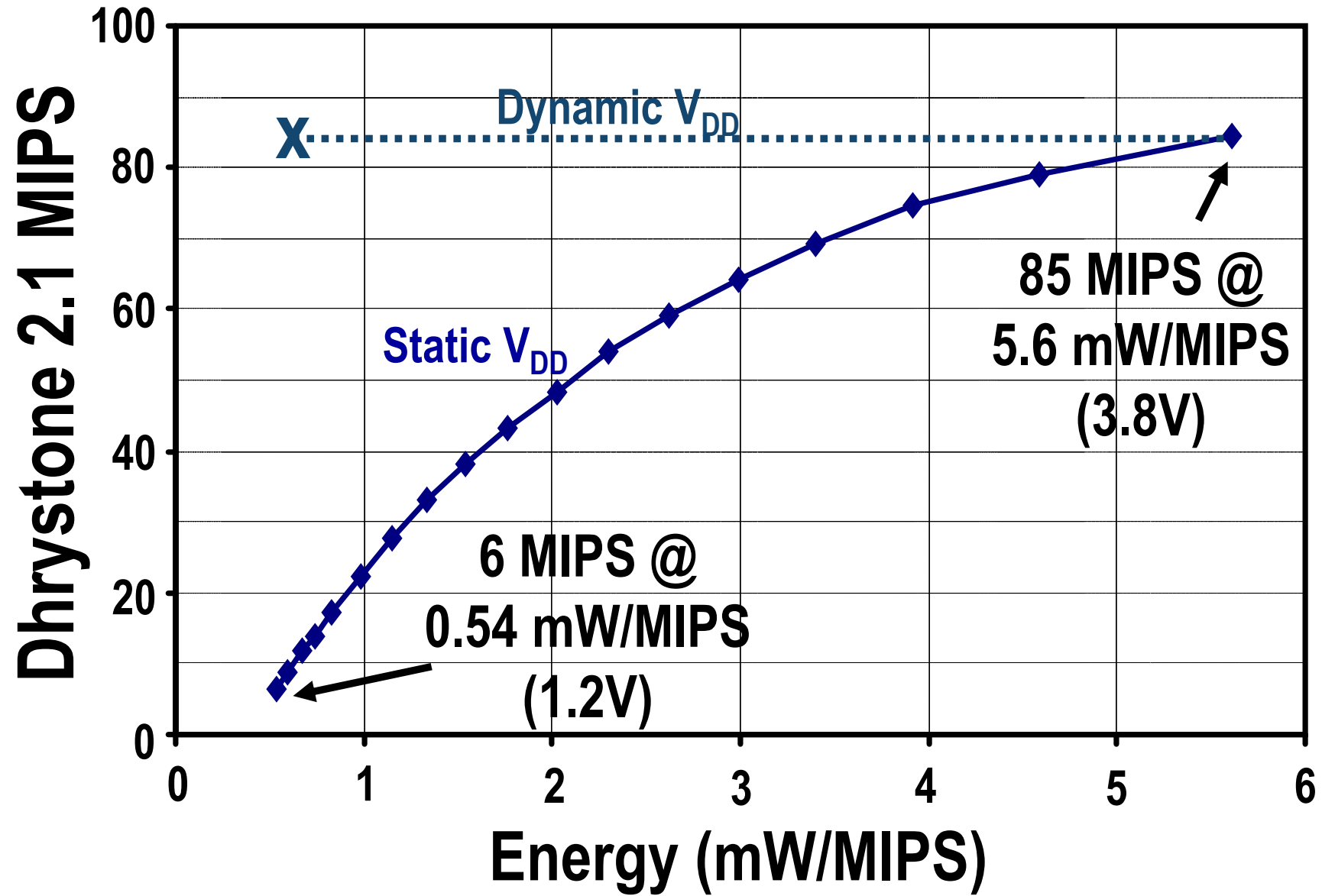
False logic low:  $\Delta V_{DD} > V_{TP}$

Latch-up:  $\Delta V_{DD} > V_{be}$

**0.6 $\mu$ m CMOS:  $|dV_{DD}/dt| < 20V/\mu s$**

- **Cannot gate clock in evaluation state.**
- **Tri-state busses fail similarly  $\rightarrow$  Use hold circuit.**

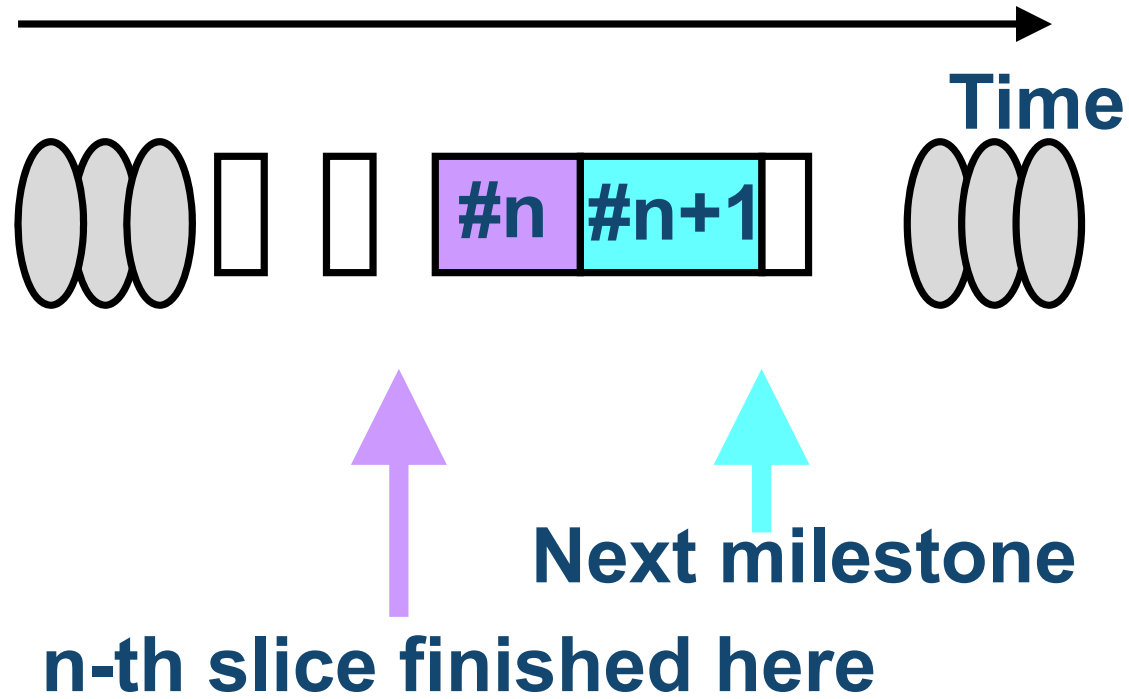
# Measured System Performance & Energy



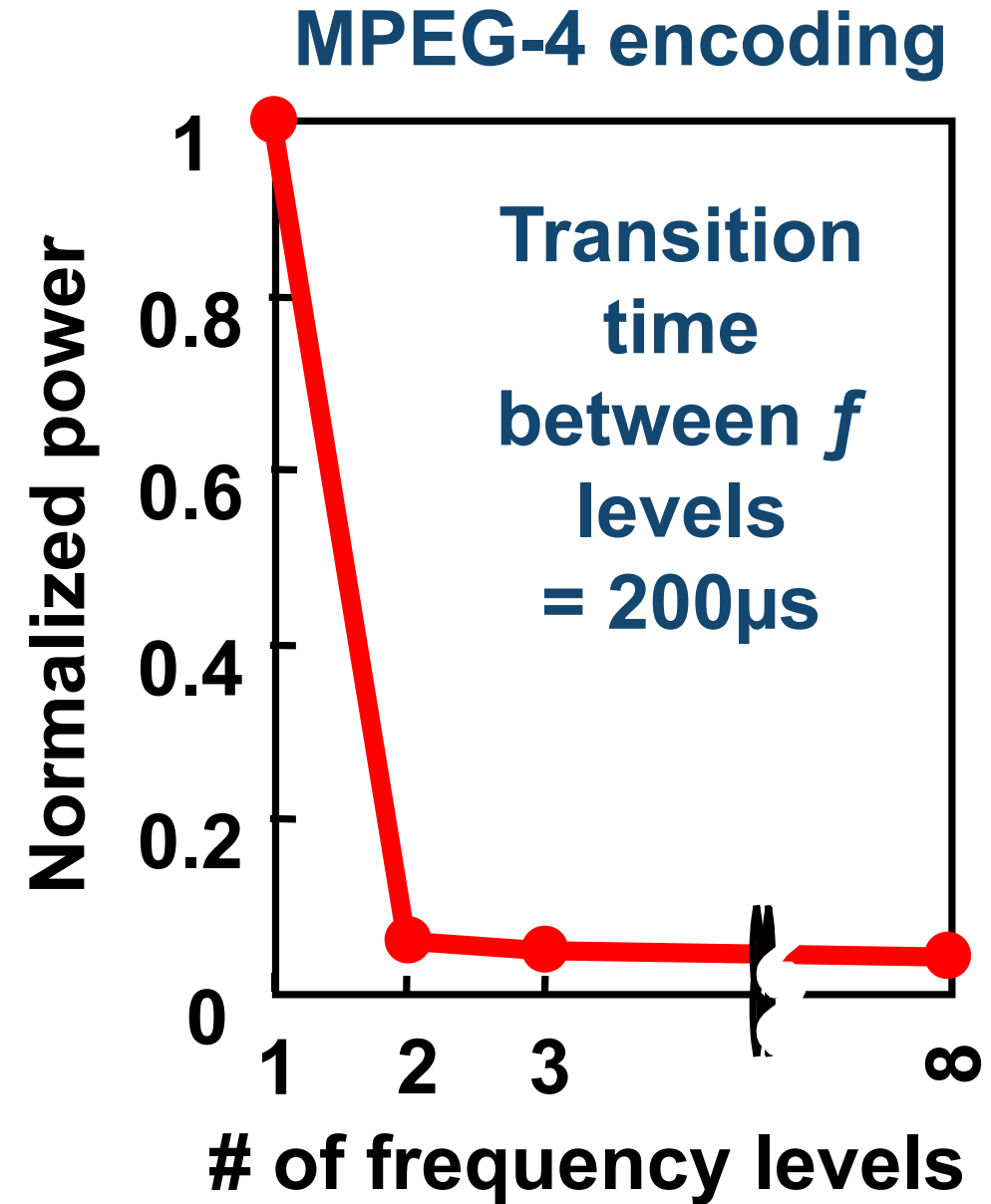
- Dynamic operation can increase energy efficiency > 10x.

Burd  
ISSCC'00

# V<sub>DD</sub>-Hopping

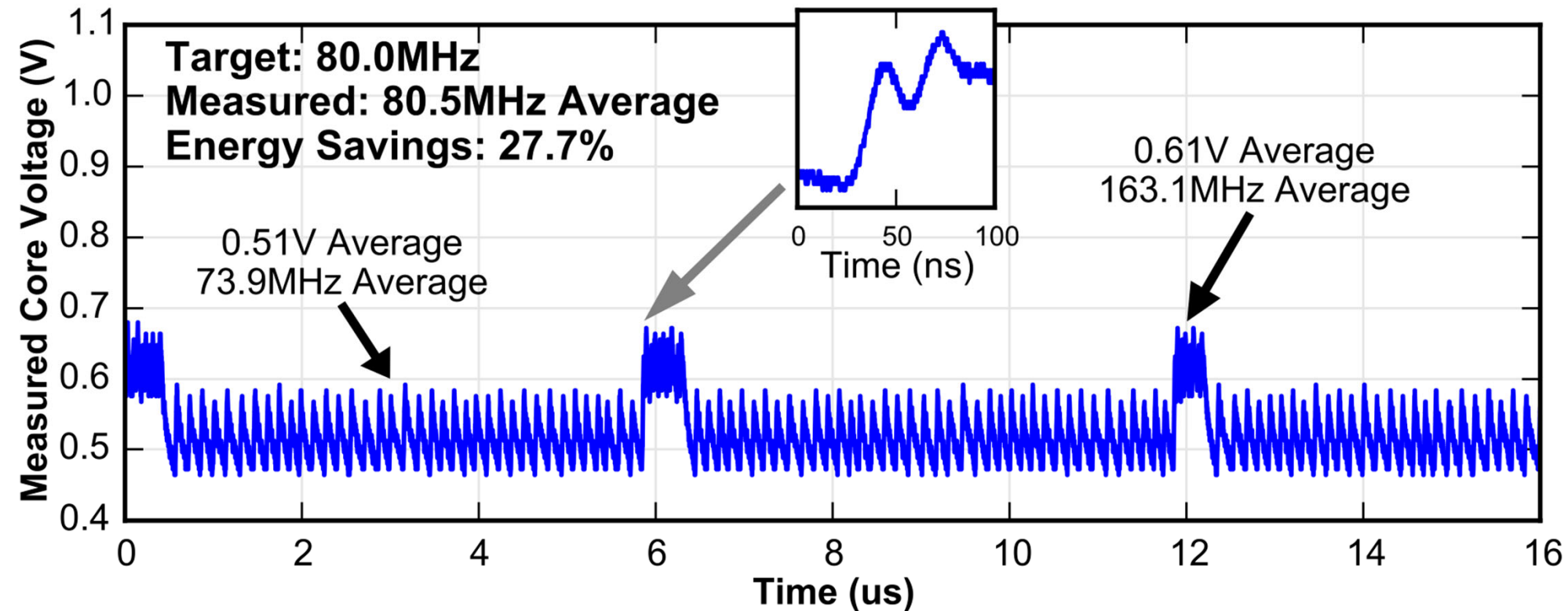


**Application slicing and software feedback guarantee real-time operation.**



**Two hopping levels are sufficient.**

# Dithering Between Supply Levels

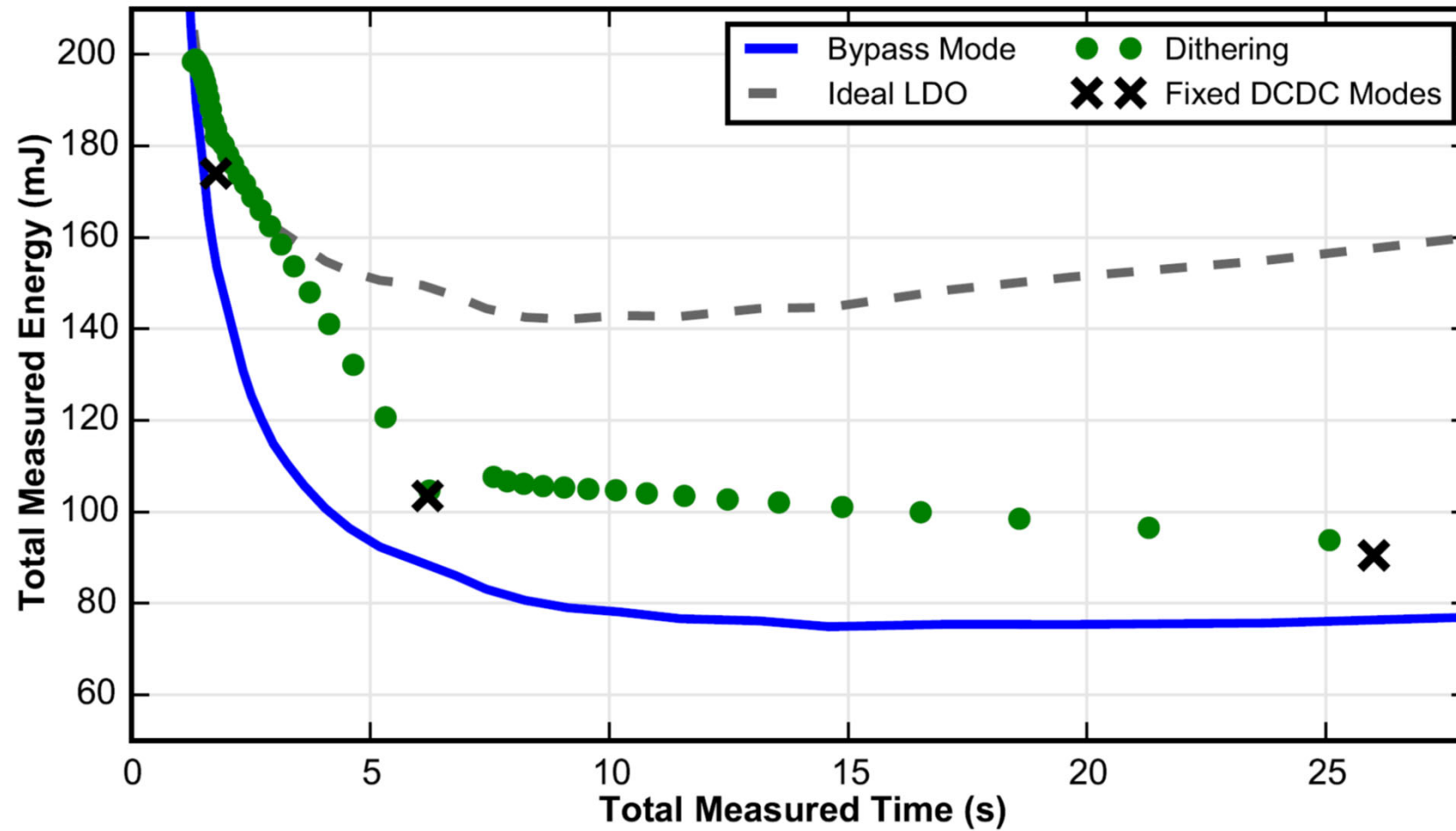


- Done with switched-capacitor DC-DC converters which efficiently work only at discrete levels

Keller et al, ESSCIRC'16

# Dithering Between Supply Levels

- Dithering fills in between fixed DC-DC modes



Keller et al, ESSCIRC'16

# Next Lecture

- Low-power design
  - Clock gating
  - Power gating