EECS251B : Advanced Digital Circuits and Systems

Lecture 23 – Dynamic Voltage Scaling

Borivoje Nikolić

Introducing Google Axion Processors, our new Arm-based CPUs

April 9, 2024. Amin Vahdat. "…Today, we are thrilled to announce the latest incarnation of this work: Google Axion Processors, our first custom Arm®-based CPUs designed for the data center. Axion delivers industry-leading performance and energy efficiency and will be available to Google Cloud customers later this year."

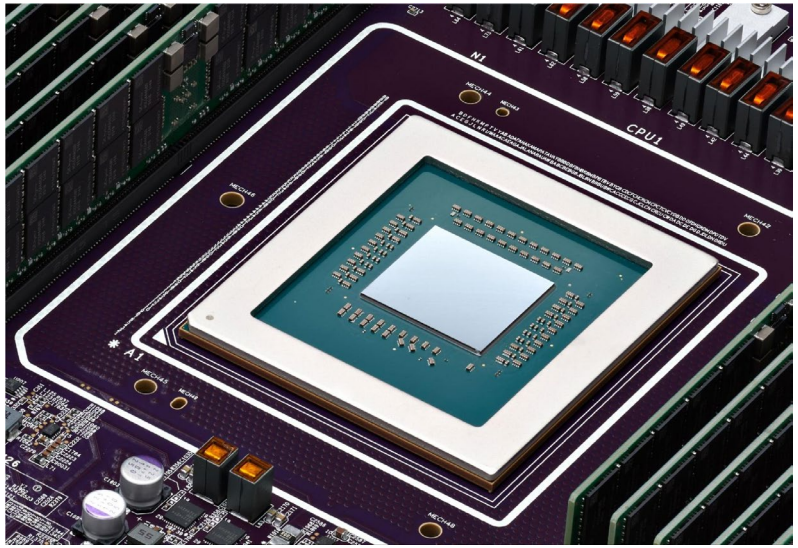


Image source: Google

Announcements

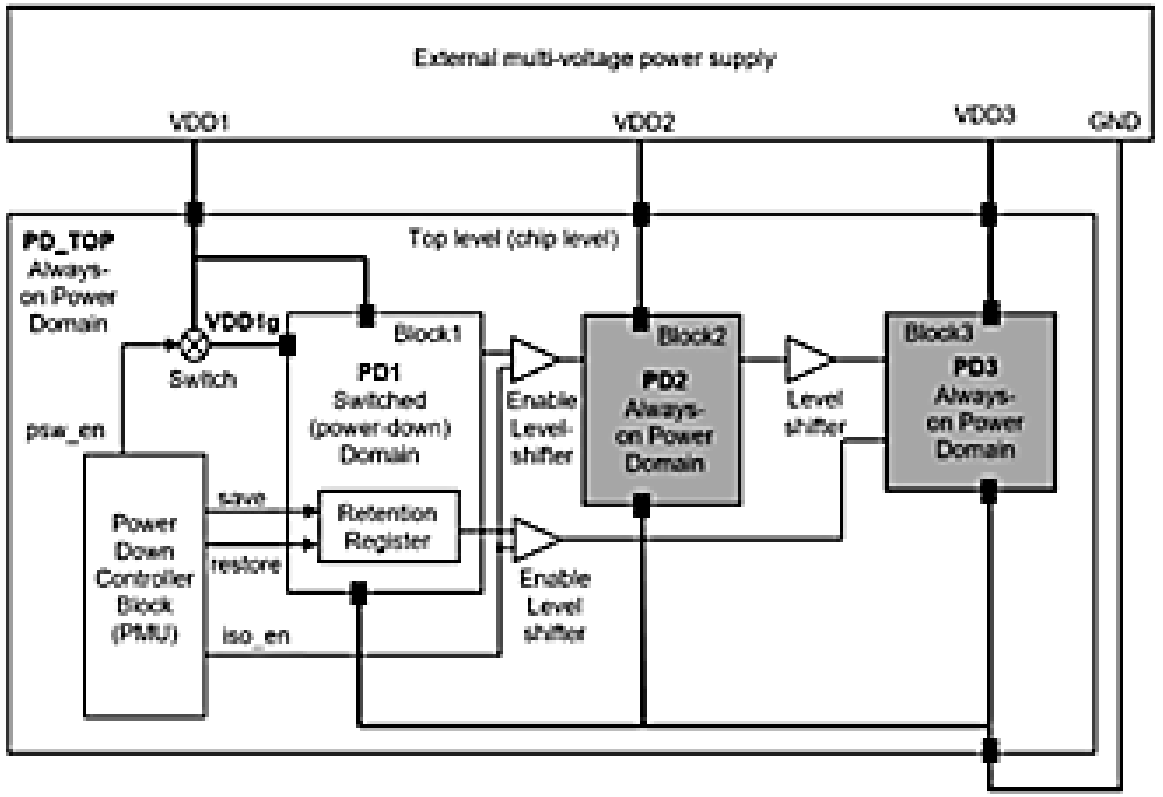
- Homework 4 due this week
  - Quiz 4 next Tuesday, in class
- Project
  - Good preliminary design review on Tuesday
  - Pay attention to integration with other teams!
  - Final presentations: May 2, 9am-12pm
- Final exam: April 26, in class

Multiple Supplies

Multiple Supply Voltages

- Block-level supply assignment ("power domains" or "voltage islands")
  - Higher throughput/lower latency functions are implemented in higher V_{DD}
  - Slower functions are implemented with lower V_{DD}
  - Often called "Voltage islands"
  - Separate supply grids, level conversion performed at block boundaries
- Multiple supplies inside a block
  - Non-critical paths moved to lower supply voltage
  - Level conversion within the block
  - Physical design challenging
  - (Not used in practice)

Power Domains

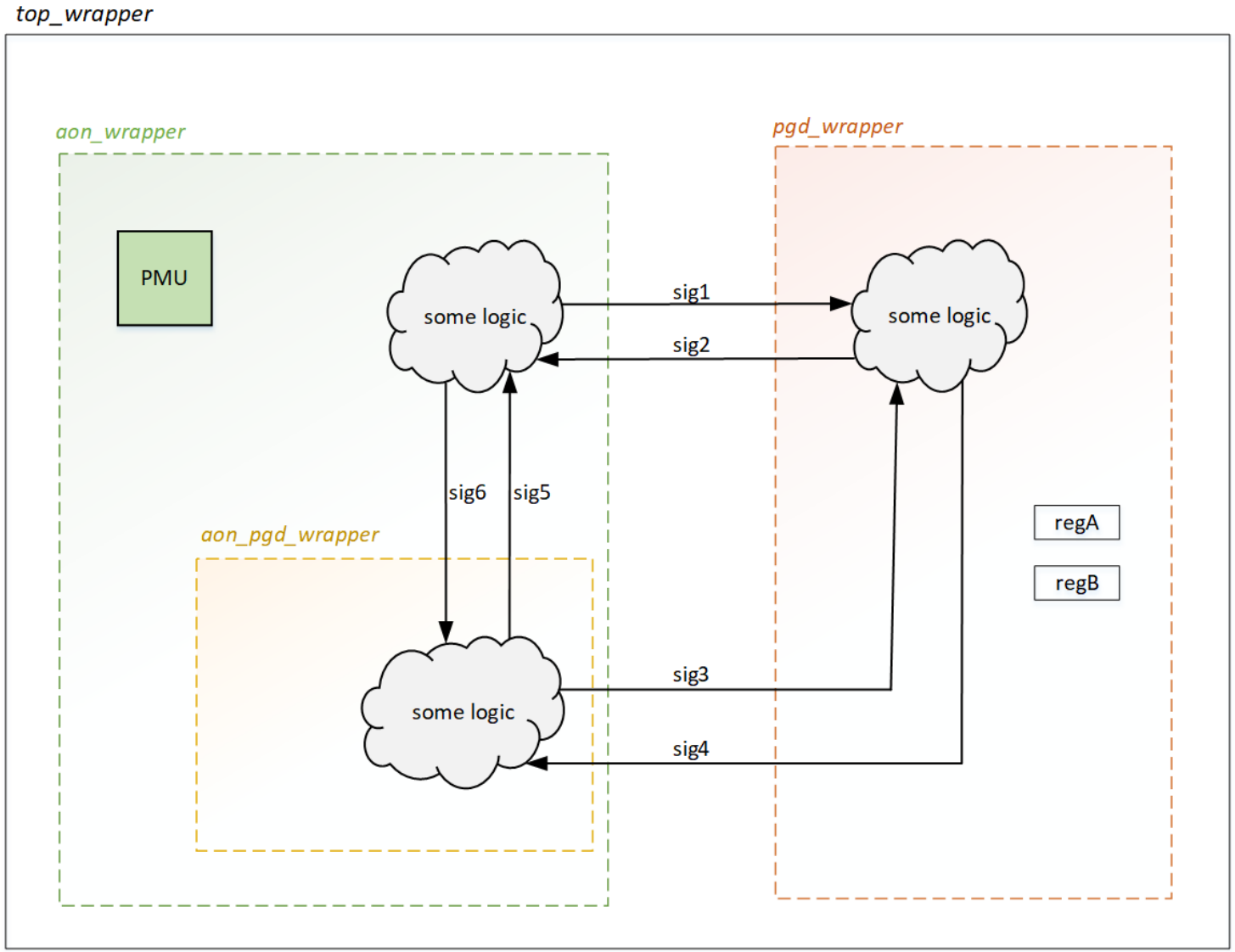


[Synopsys]

Typical Low Power SoC

Utilize Unified Power Format (UPF) to capture design intent
Common Power Format (CPF) is similar

Power Domain Design Intent



https://vlsitutorials.com/upf-low-power-vlsi/

There are primarily 3 power domains –

- Logic inside aon_wrapper [but not inside aon_pgd_wrapper] is always-on.
- Logic inside pgd_wrapper can be power gated.
- Logic inside aon_pgd_wrapper can be power gated but won't be power gated when pgd_wrapper is powered ON.

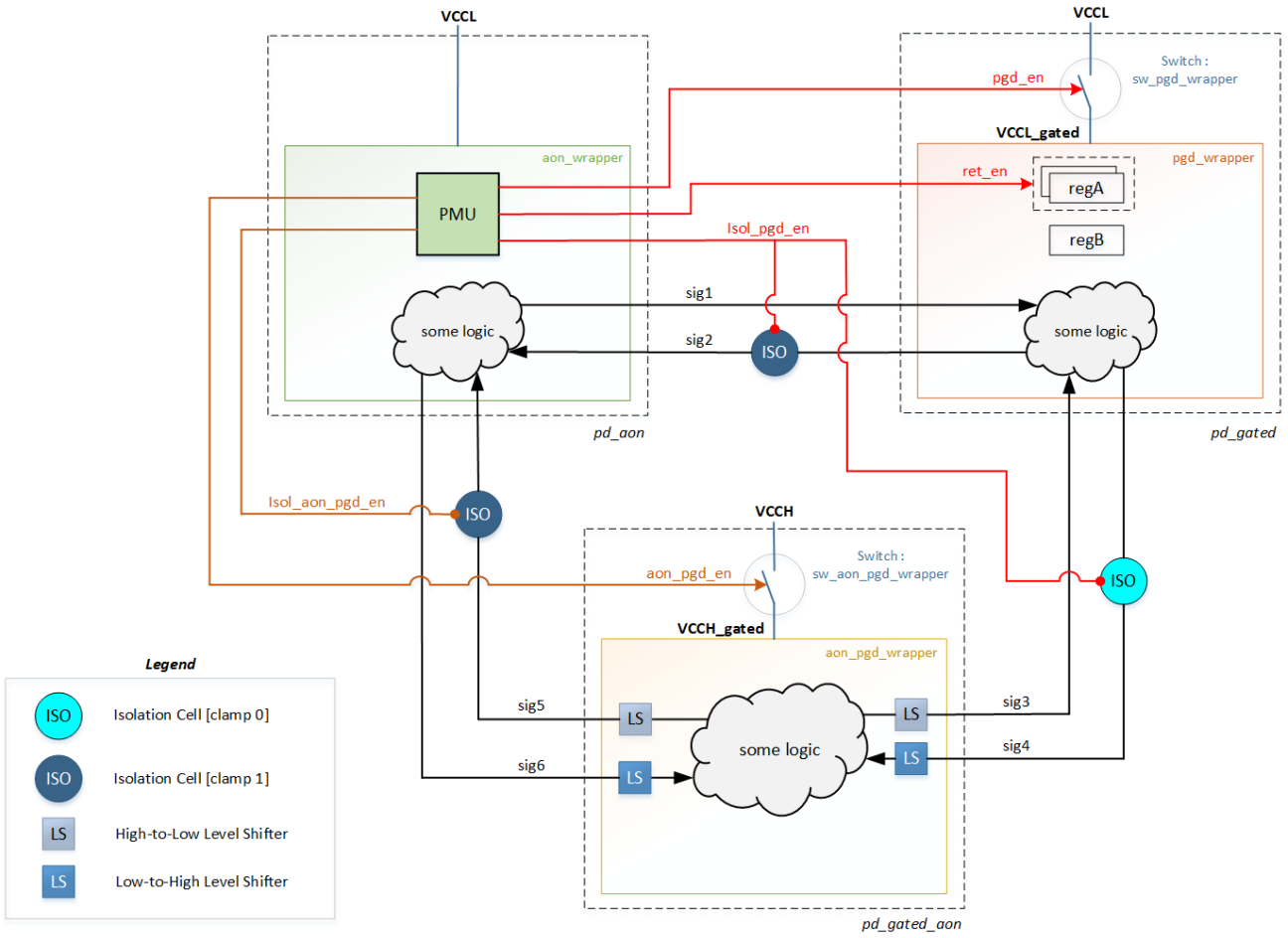
There are two voltage domains –

- The supply voltage to logic inside aon_wrapper [but not inside aon_pgd_wrapper] and logic inside pgd_wrapper is 0.9V.
- The supply voltage to logic inside aon_pgd_wrapper is 1.1V.

There are two registers – reg A and reg B. The state of reg A needs to be retained in power gated state.

There are six signals sig1-sig6 coming to and from different logic blocks.

Unified Power Format



 # Create Power Domains

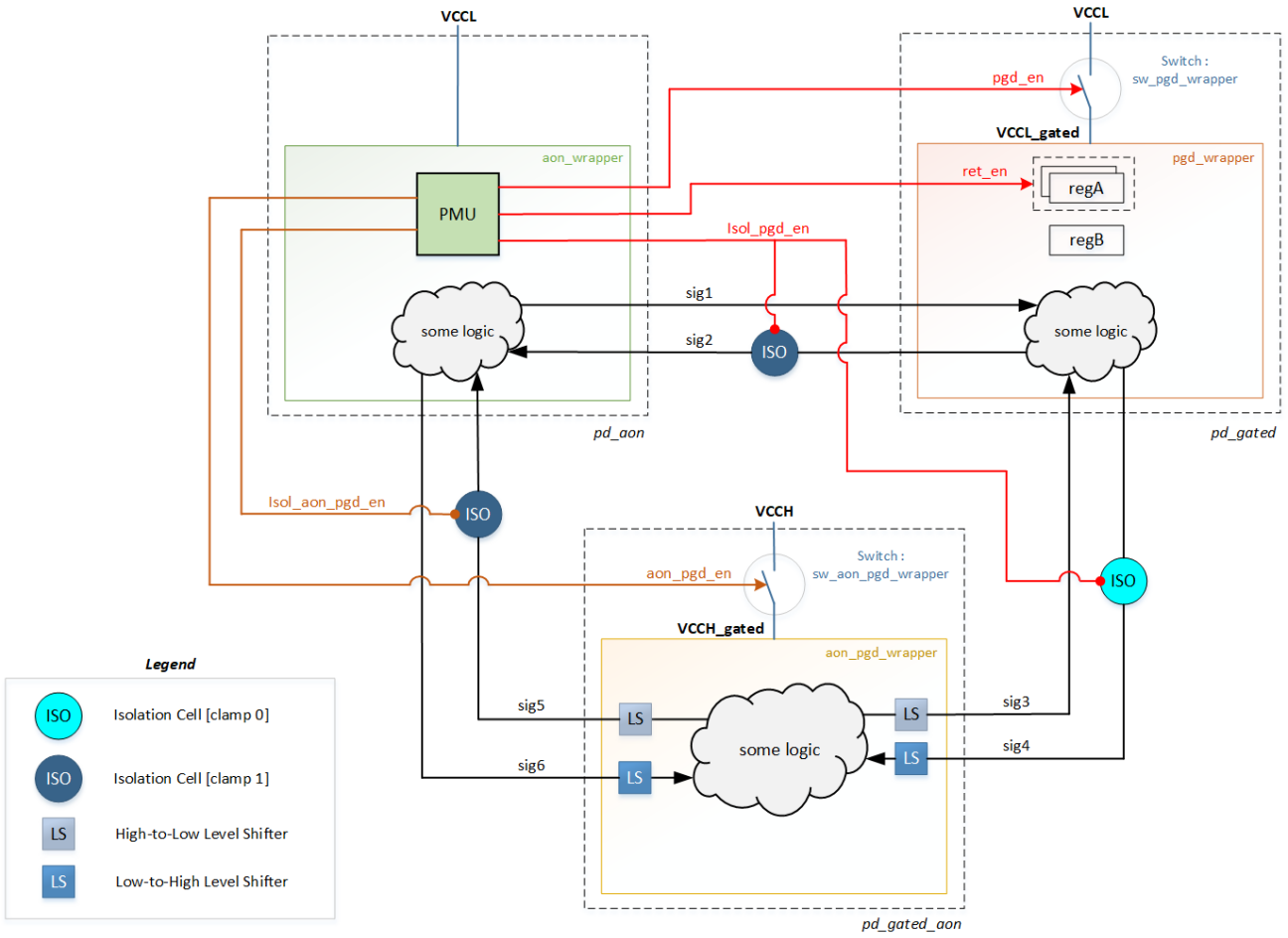
create_power_domain pd_top -include_scope
create_power_domain pd_aon -elements {aon_wrapper}
create_power_domain pd_gated -elements {pgd_wrapper}
create_power_domain pd_gated_aon -elements {{aon_wrapper/aon_pgd_wrapper}}

There are primarily 3 power domains –

- Logic inside aon_wrapper [but not inside aon_pgd_wrapper] is always-on.
- Logic inside pgd_wrapper can be power gated.
- Logic inside aon_pgd_wrapper can be power gated but won't be power gated when pgd_wrapper is powered ON.

https://vlsitutorials.com/upf-low-power-vlsi/

Unified Power Format



 # Create Supply Ports

create_supply_port VCCL -direction in -domain pd_top
create_supply_port VCCH -direction in -domain pd_top
create_supply_port GND -direction in -domain pd_top

 # Create Supply Nets

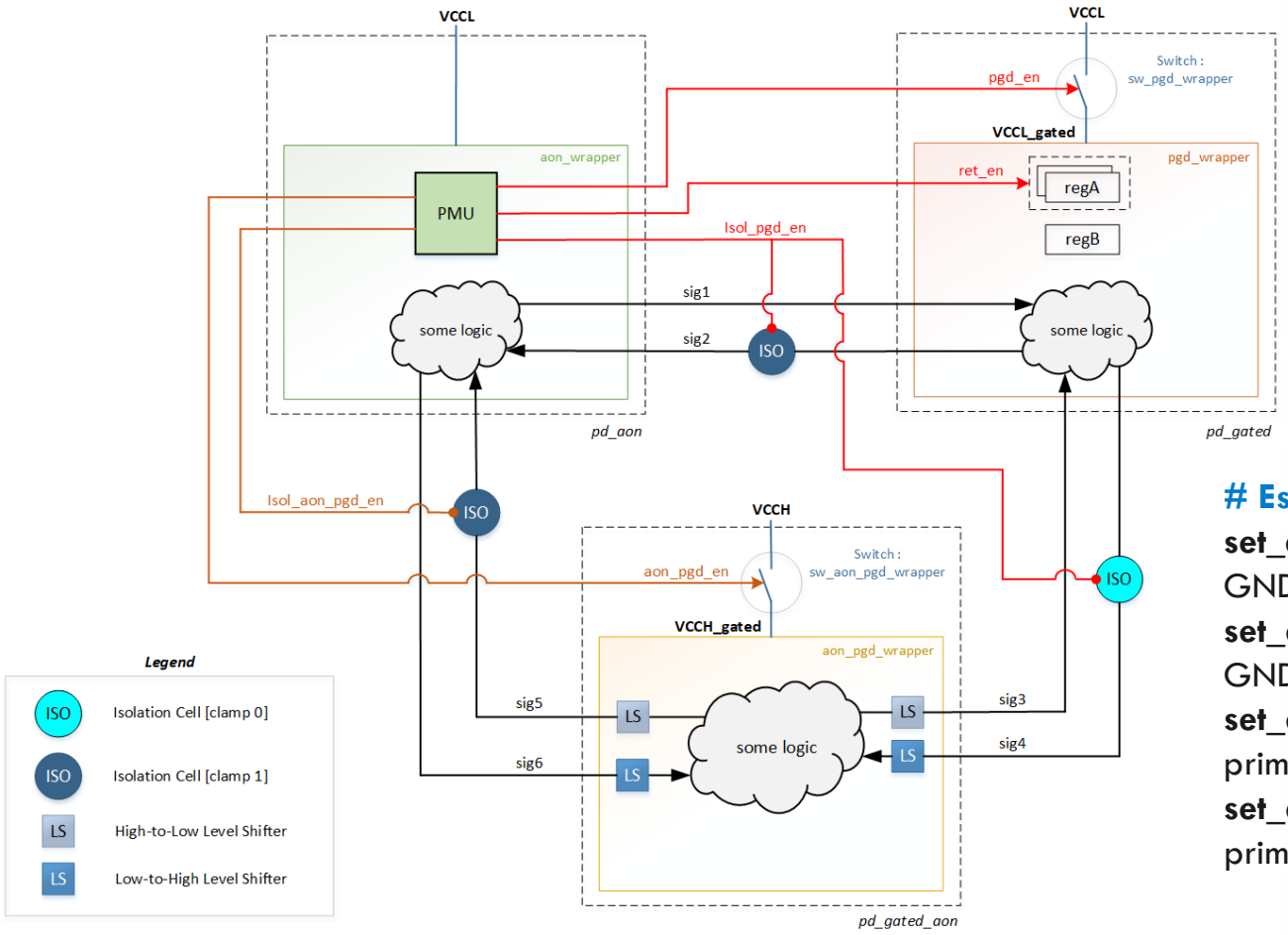
create_supply_net VCCL -domain pd_top
create_supply_net VCCH -domain pd_top
create_supply_net GND -domain pd_top
create_supply_net VCCL -domain pd_aon -reuse
create_supply_net GND -domain pd_aon -reuse
create_supply_net VCCH -domain pd_gated_aon –reuse
create_supply_net VCCH_gated -domain pd_gated_aon
create_supply_net GND -domain pd_gated_aon -reuse
create_supply_net VCCL -domain pd_gated -reuse
create_supply_net VCCL_gated -domain pd_gated
create_supply_net GND -domain pd_gated -reuse

https://vlsitutorials.com/upf-low-power-vlsi/

There are two voltage domains –

- The supply voltage to logic inside aon_wrapper [but not inside aon_pgd_wrapper] and logic inside pgd_wrapper is 0.9V.
- The supply voltage to logic inside aon_pgd_wrapper is 1.1V.

Unified Power Format



 # Connect Supply Nets with corresponding Ports

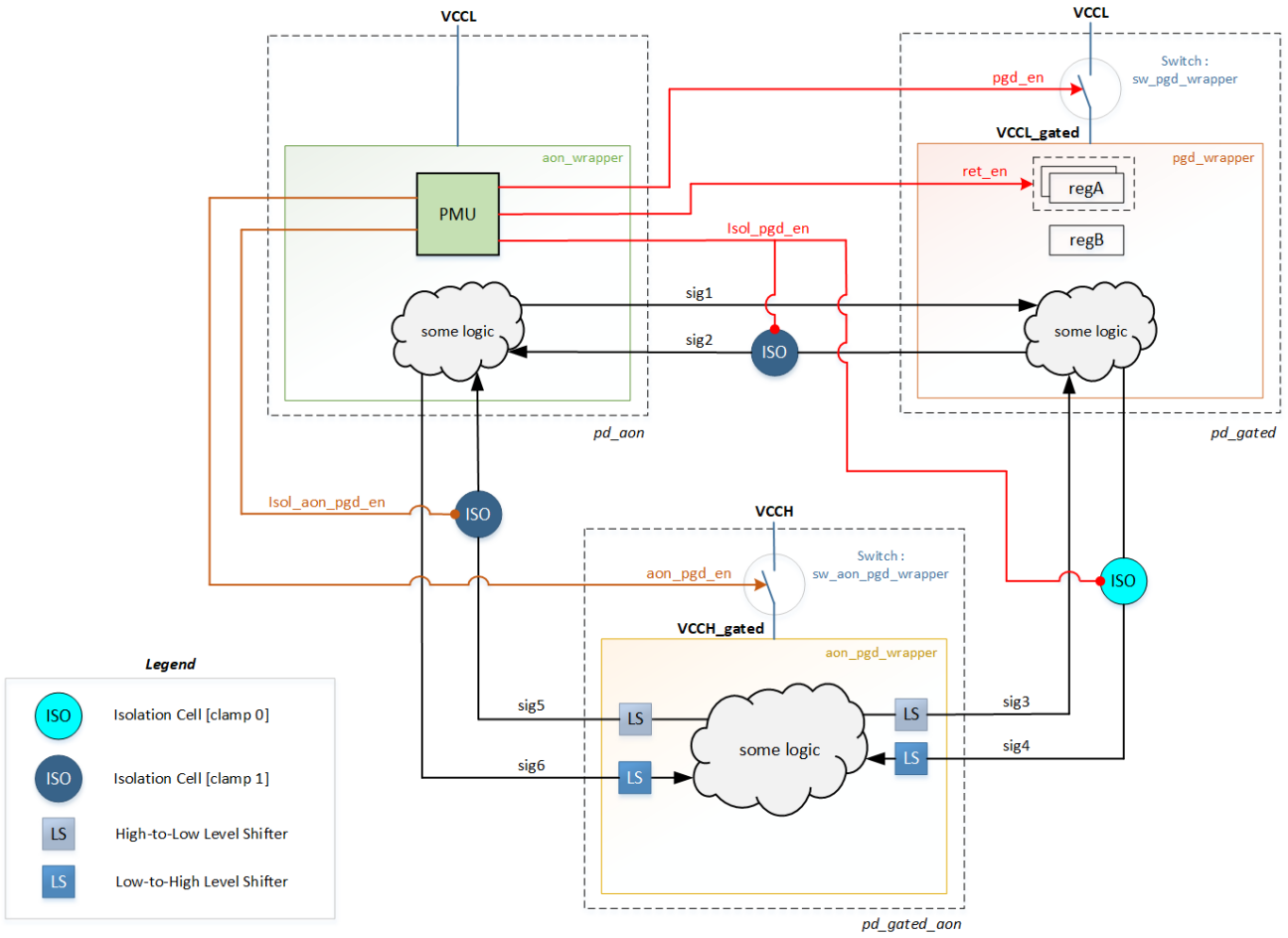
connect_supply_net VCCL -ports VCCL
connect_supply_net VCCH -ports VCCH
connect_supply_net GND -ports GND

 # Establish Connections

set_domain_supply_net pd_top -primary_power_net VCCL -primary_ground_net GND
set_domain_supply_net pd_aon -primary_power_net VCCL -primary_ground_net GND
set_domain_supply_net pd_gated_aon -primary_power_net VCCH_gated -primary_ground_net GND
set_domain_supply_net pd_gated -primary_power_net VCCL_gated -primary_ground_net GND

https://vlsitutorials.com/upf-low-power-vlsi/

Unified Power Format

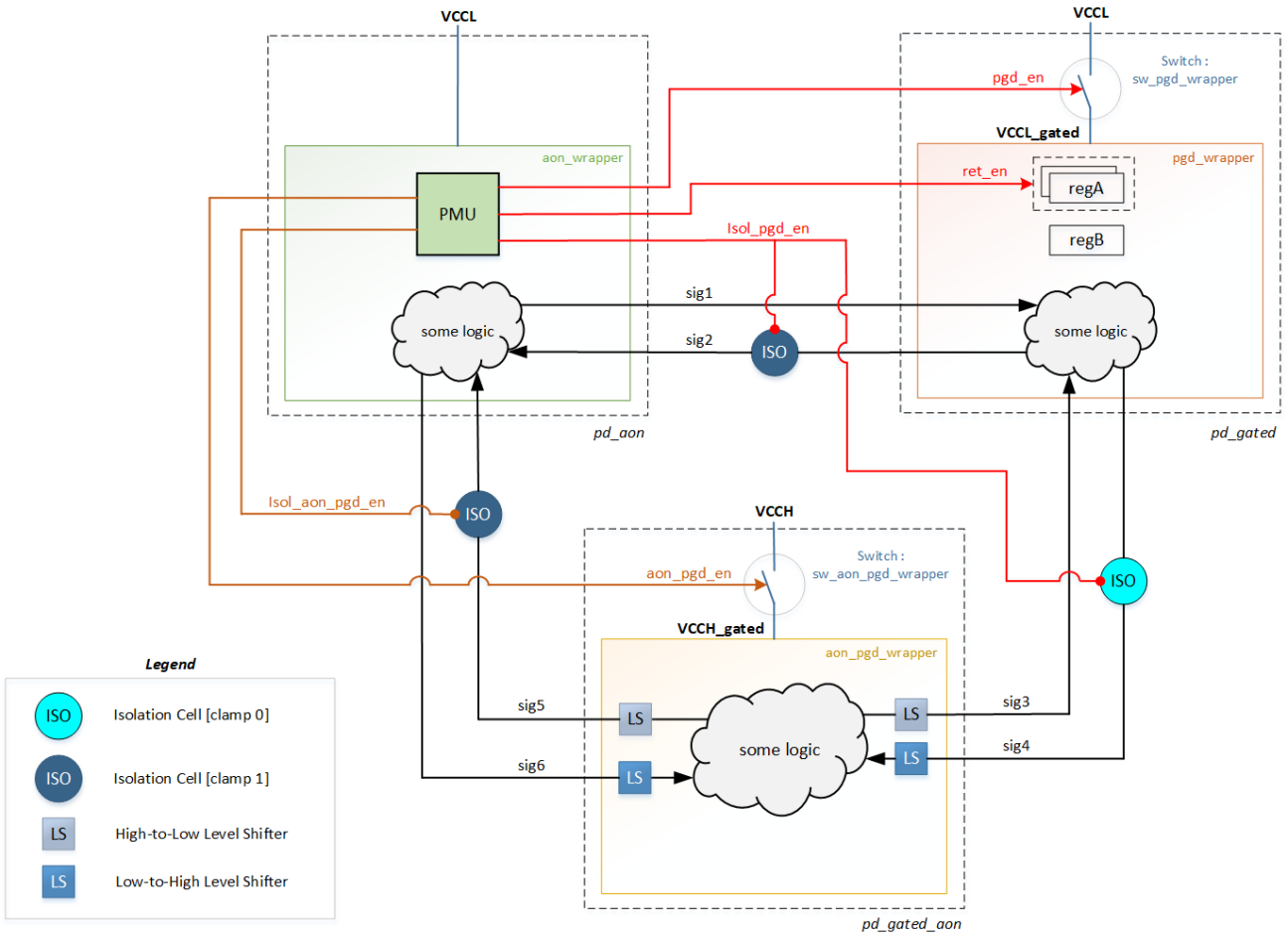


 # Shut-Down Logic for pgd_wrapper & aon_pgd_wrapper

create_power_switch sw_pgd_wrapper \
-domain pd_gated \
-input_supply_port “sw_VCCL VCCL ” \
-output_supply_port “sw_VCCL_gated VCCL_gated” \
-control_port “sw_pgd_en aon_wrapper/pmu/pgd_en” \
-on_state “SW_PGD_ON sw_VCCL {!sw_pgd_en}”
create_power_switch sw_aon_pgd_wrapper \
-domain pd_gated_aon \
-input_supply_port “sw_VCCH VCCH ” \
-output_supply_port “sw_VCCH_gated VCCH_gated” \
-control_port “sw_aon_pgd_en aon_wrapper/pmu/aon_pgd_en” \
-on_state “SW_AONPGD_ON sw_VCCH {!sw_aon_pgd_en}”

https://vlsitutorials.com/upf-low-power-vlsi/

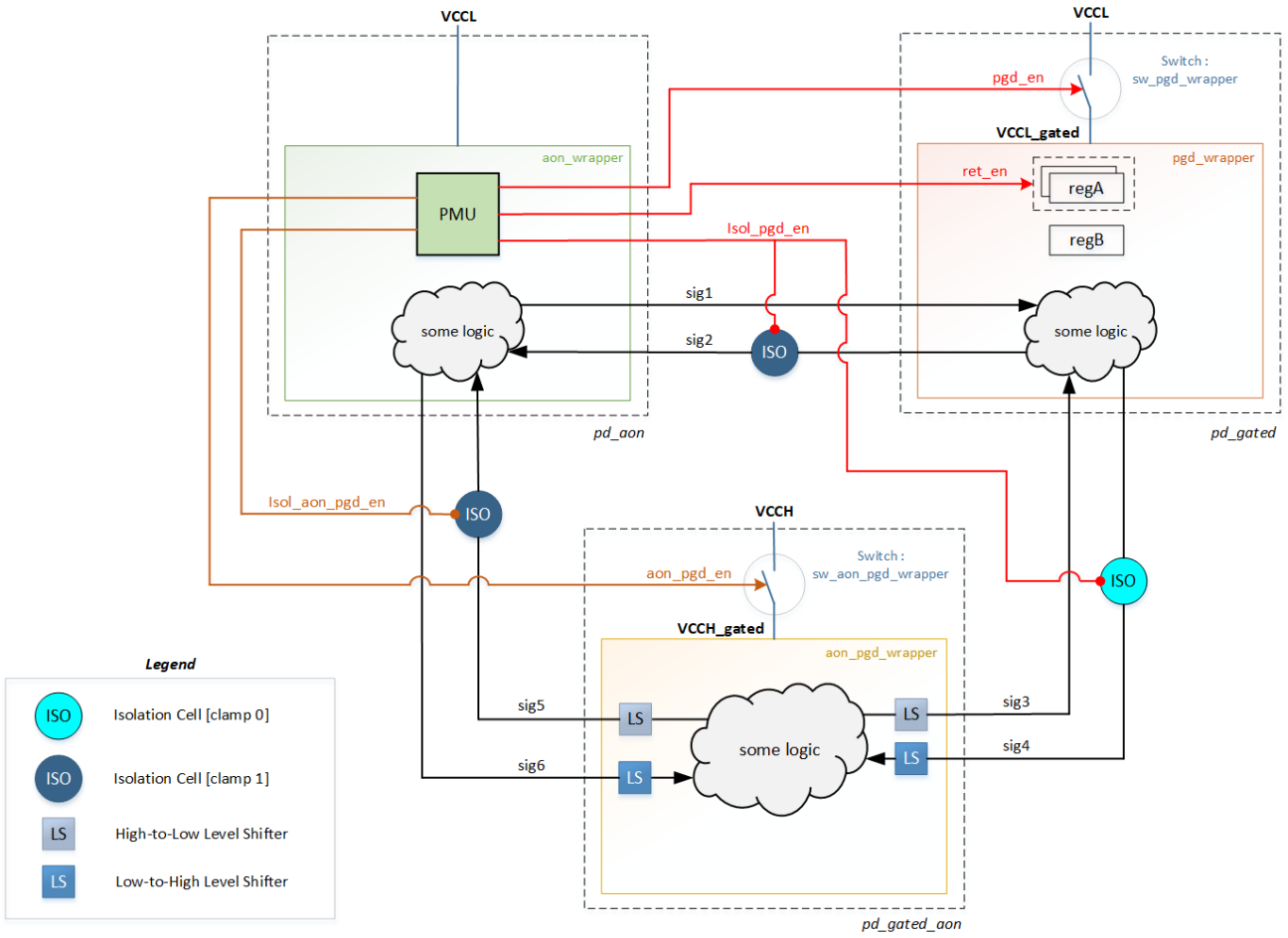
Unified Power Format




```
# Isolation strategy
set_isolation isol_clamp1_sig_from_pgd \
-domain pd_gated \
-isolation_power_net VCCL \
-isolation_ground_net GND \
-clamp_value 1 \
-elements {pgd_wrapper/sig2}
set_isolation_control isol_clamp1_sig_from_pgd \
-domain pd_gated \
-isolation_signal aon_wrapper/pmu/isol_pgd_en \
-isolation_sense low \
-location parent
set_isolation isol_clamp0_sig_from_pgd \
-domain pd_gated \
-isolation_power_net VCCL \
-isolation_ground_net GND \
-clamp_value 0 \
-elements {pgd_wrapper/sig4}
set_isolation_control isol_clamp0_sig_from_pgd \
-domain pd_gated \
-isolation_signal aon_wrapper/pmu/isol_pgd_en \
-isolation_sense low \
-location parent
...
```

https://vlsitutorials.com/upf-low-power-vlsi/

Unified Power Format

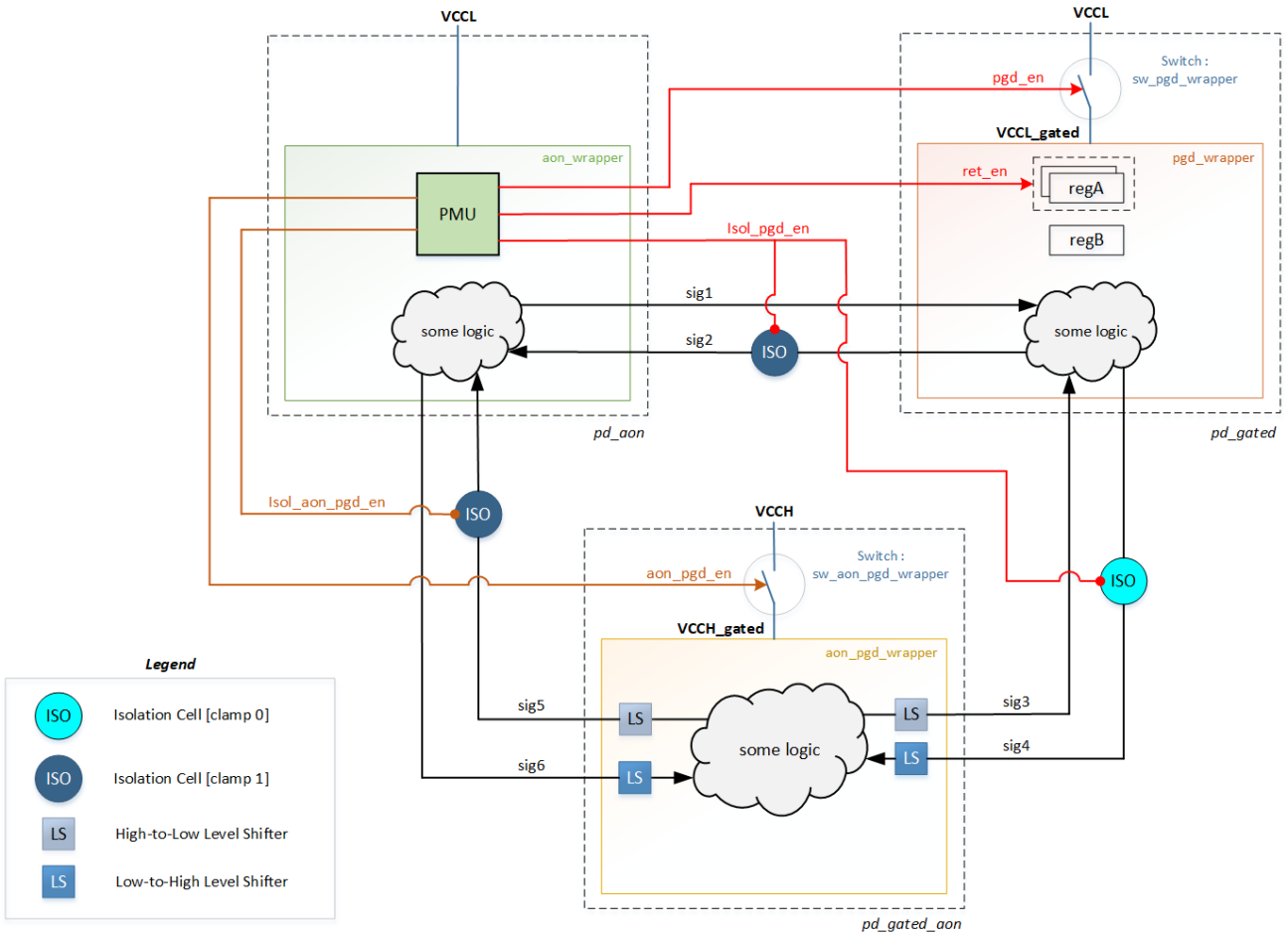


 # Level Shifter strategy

set_level_shifter LtoH_sig_to_aonpgd \
-domain pd_gated_aon \
-applies_to inputs \
-rule low_to_high \
-location self
set_level_shifter HtoL_sig_from_aonpgd \
-domain pd_gated_aon \
-applies_to outputs \
-rule high_to_low \
-location self

https://vlsitutorials.com/upf-low-power-vlsi/

Unified Power Format



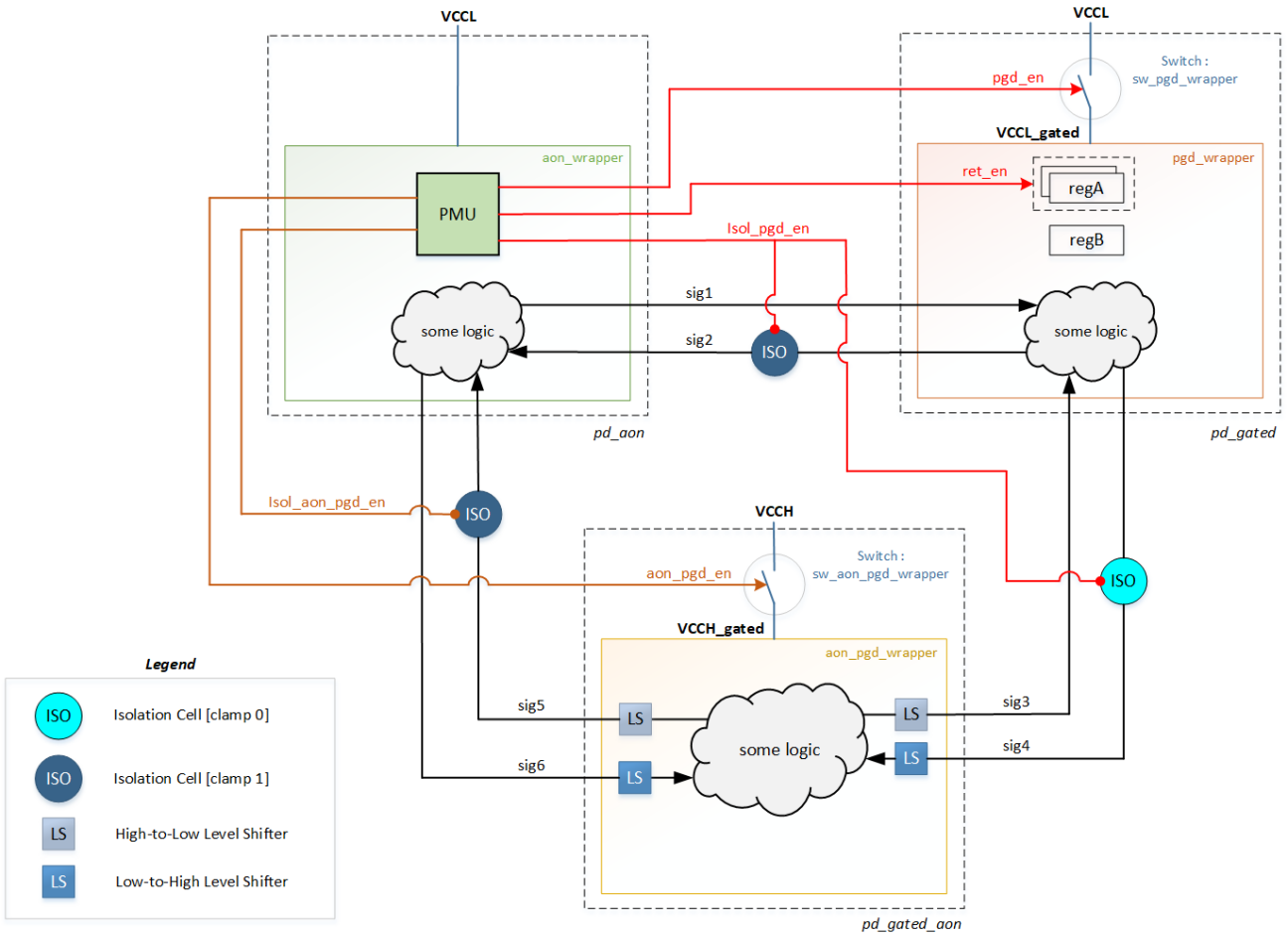
 # Retention strategy

set_retention pgd_retain \
-domain pd_gated \
-retention_power_net VCCL \
-retention_ground_net GND \
-elements {pgd_wrapper/regA}
set_retention_control pgd_retain \
-domain pd_gated \
-save_signal {aon_wrapper/pmu/ret_en high} \
-restore_signal {aon_wrapper/pmu/ret_en low}

There are two registers – reg A and reg B.
The state of reg A needs to be retained in power gated state.

https://vlsitutorials.com/upf-low-power-vlsi/

Unified Power Format

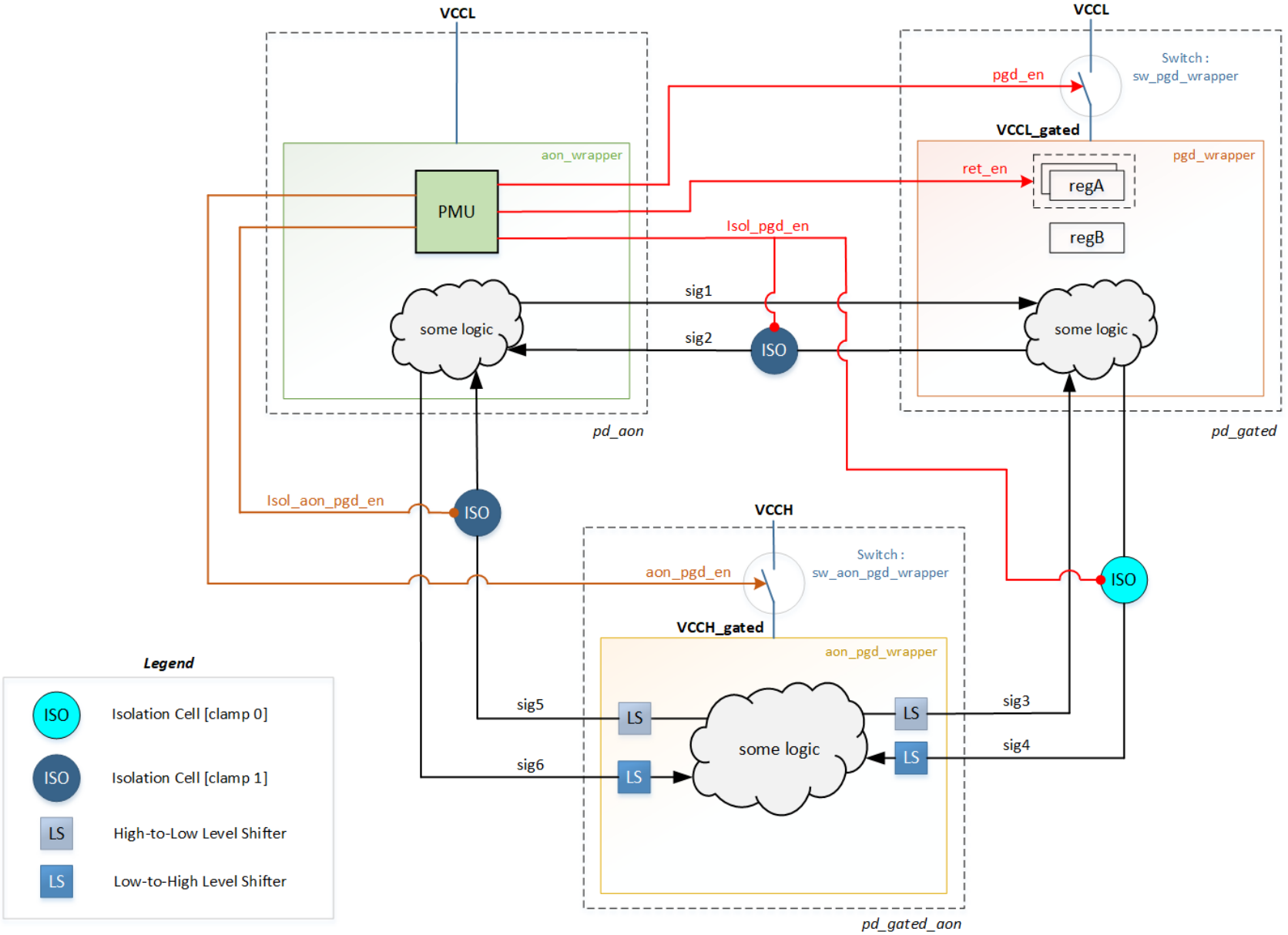


https://vlsitutorials.com/upf-low-power-vlsi/

```
# Create Power State Table
add_port_state VDDH \
-state {HighVoltage 1.1}
add_port_state VDDL \
-state {LowVoltage 0.9}
add_port_state sw_aon_pgd_wrapper/sw_VCCH_gated \
-state {HighVoltage 1.1} \
-state {aonpgd_off off}
add_port_state sw_pgd_wrapper/sw_VCCL_gated \
-state {LowVoltage 0.9} \
-state {pgd_off off}

create_pst pwr_state_table \
-supplies {VCCH VCCL VDDH_gated VDDL_gated}
add_pst_state PRE_BOOT \
-pst pwr_state_table \
-state { HighVoltage LowVoltage aonpgd_off pgd_off}
add_pst_state AONPGD_ON \
-pst pwr_state_table \
-state { HighVoltage LowVoltage HighVoltage pgd_off}
add_pst_state PGD_ON \
-pst pwr_state_table \
-state { HighVoltage LowVoltage aonpgd_off LowVoltage}
add_pst_state ALL_ON \
-pst pwr_state_table \
-state { HighVoltage LowVoltage HighVoltage LowVoltage}
```

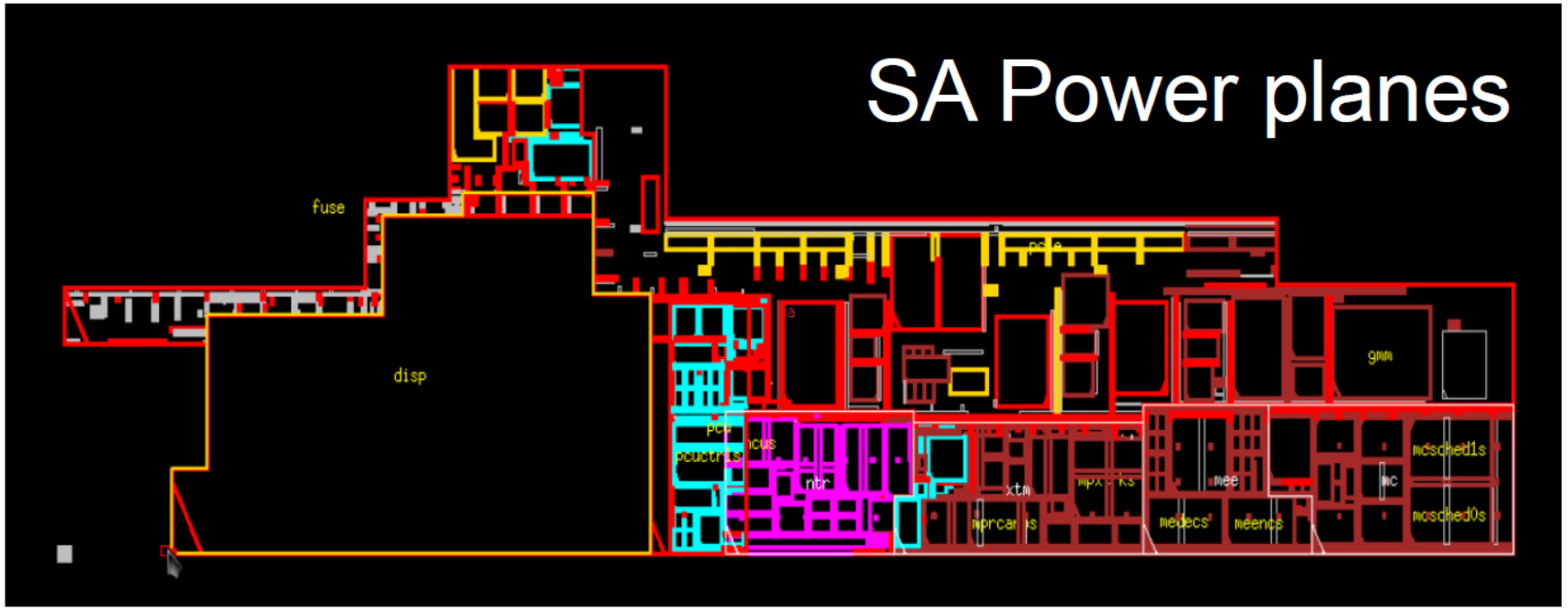
Unified Power Format



https://vlsitutorials.com/upf-low-power-vlsi/

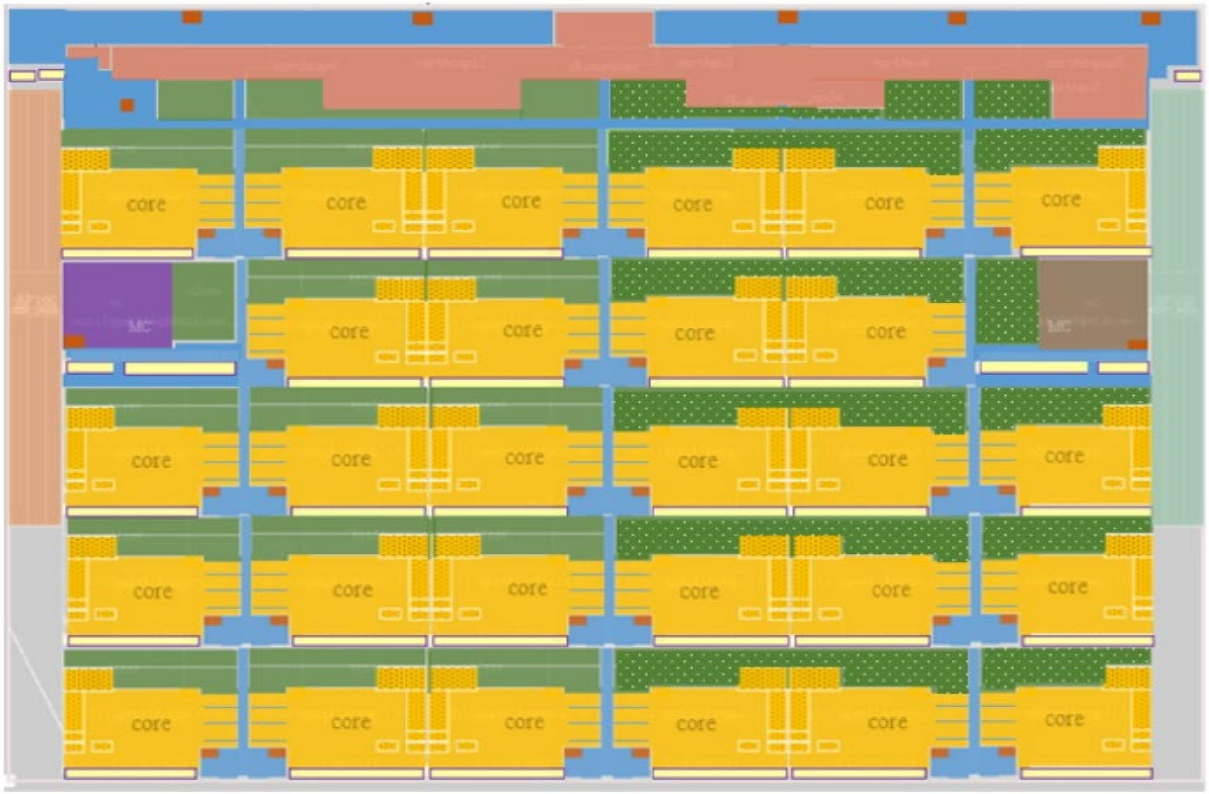
Practical Examples

- Intel Skylake (ISSCC'16)
  - Four power planes indicated by colors



Practical Examples

- Intel 28-core Skylake-SP (ISSCC'18)

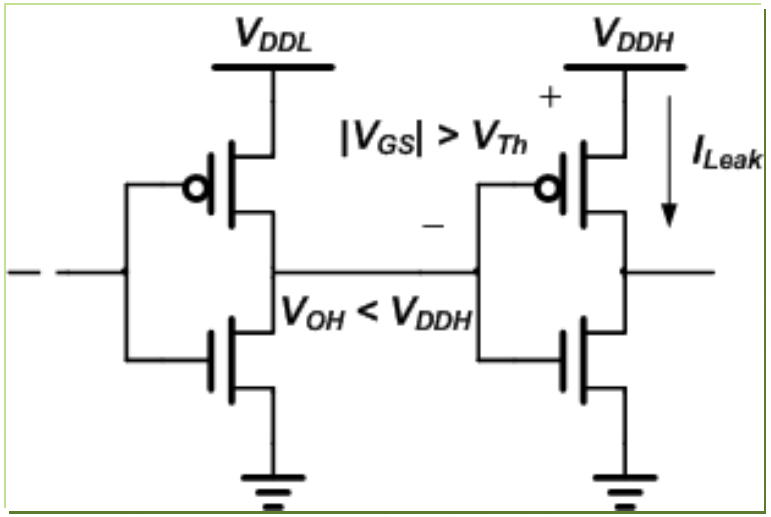


Vcc: core supply (per core)
Vccclm: Un-core supply
Vccsa: System Agent supply
Vccio: Infrastructure supply
Vccsfr: PLL supply
Vccddrd: DDR logic supply
Vccddra: DDR I/O supply

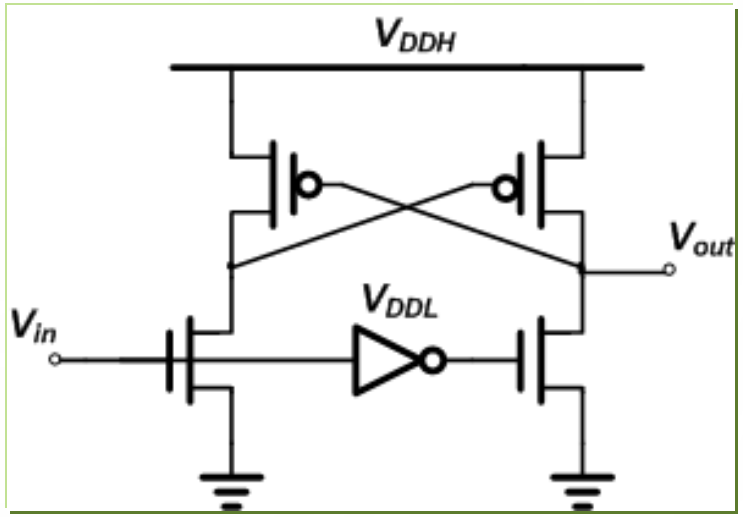
- 9 primary VCC domains are partitioned into 35 VCC planes

Leakage Issue

- Driving from V_{DDL} to V_{DDH}

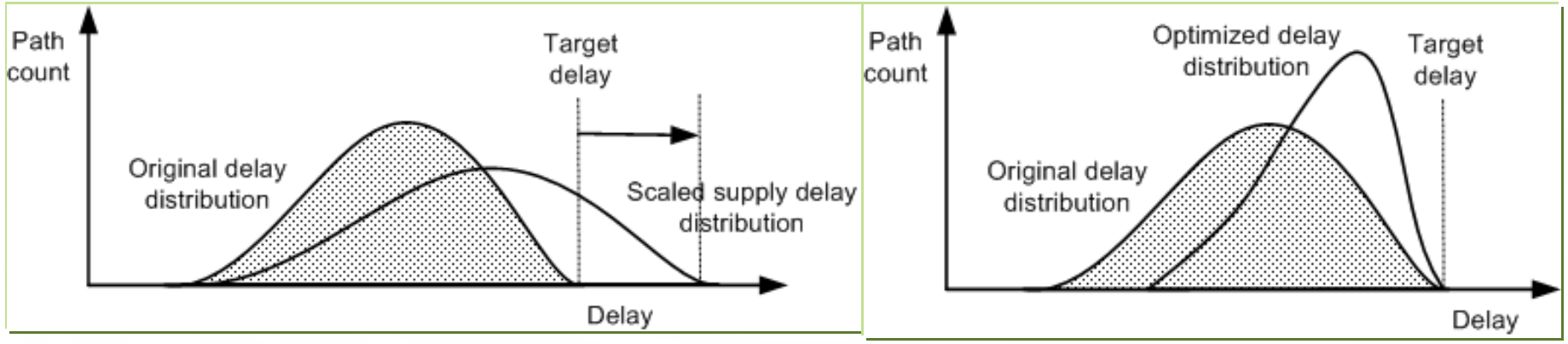


- Level converter

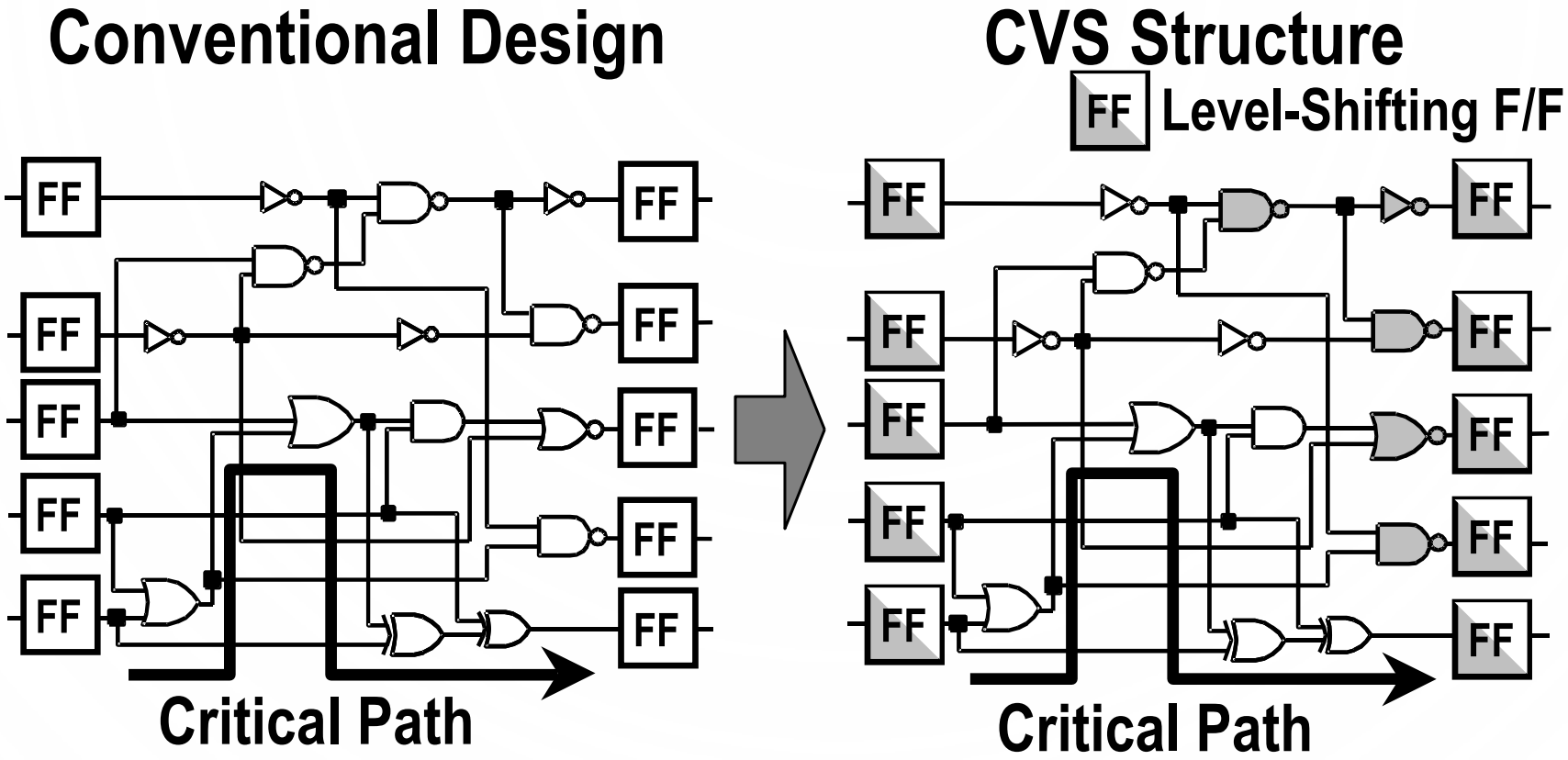


Multiple Supplies Within A Block

- Downsizing, lowering the supply on the critical path will lower the operating frequency
- Downsize (lowering supply) non-critical paths
  - Narrows down the path delay distribution
  - Increases impact of variations



Multiple Supplies in a Block



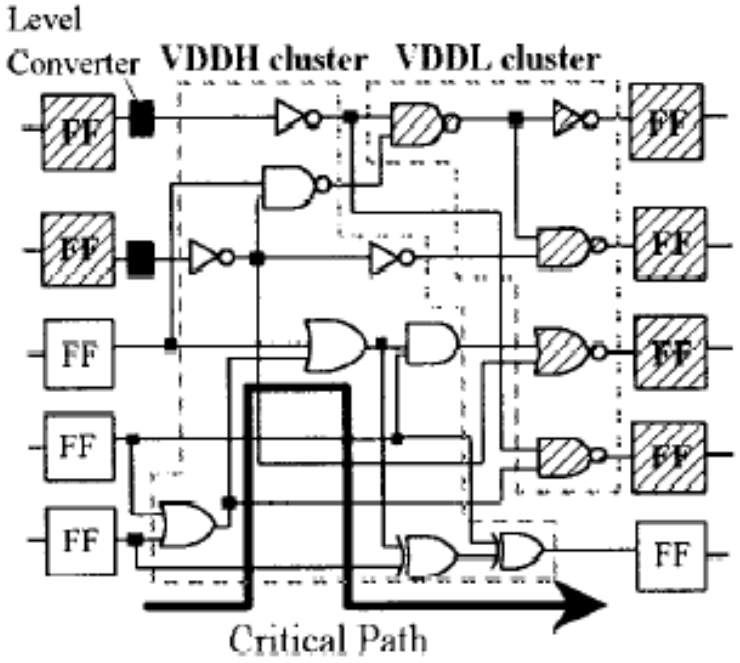
Lower V_{DD} portion is shaded

"Clustered voltage scaling"

M.Takahashi, ISSCC'98.

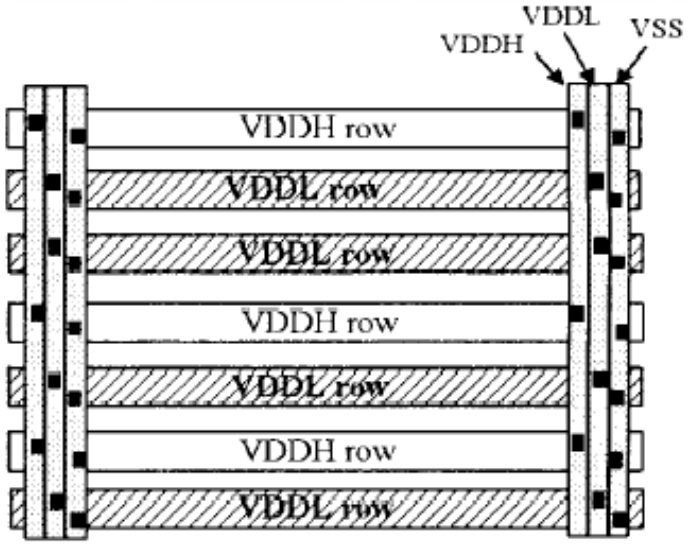
Multiple Supplies in a Block

CVS

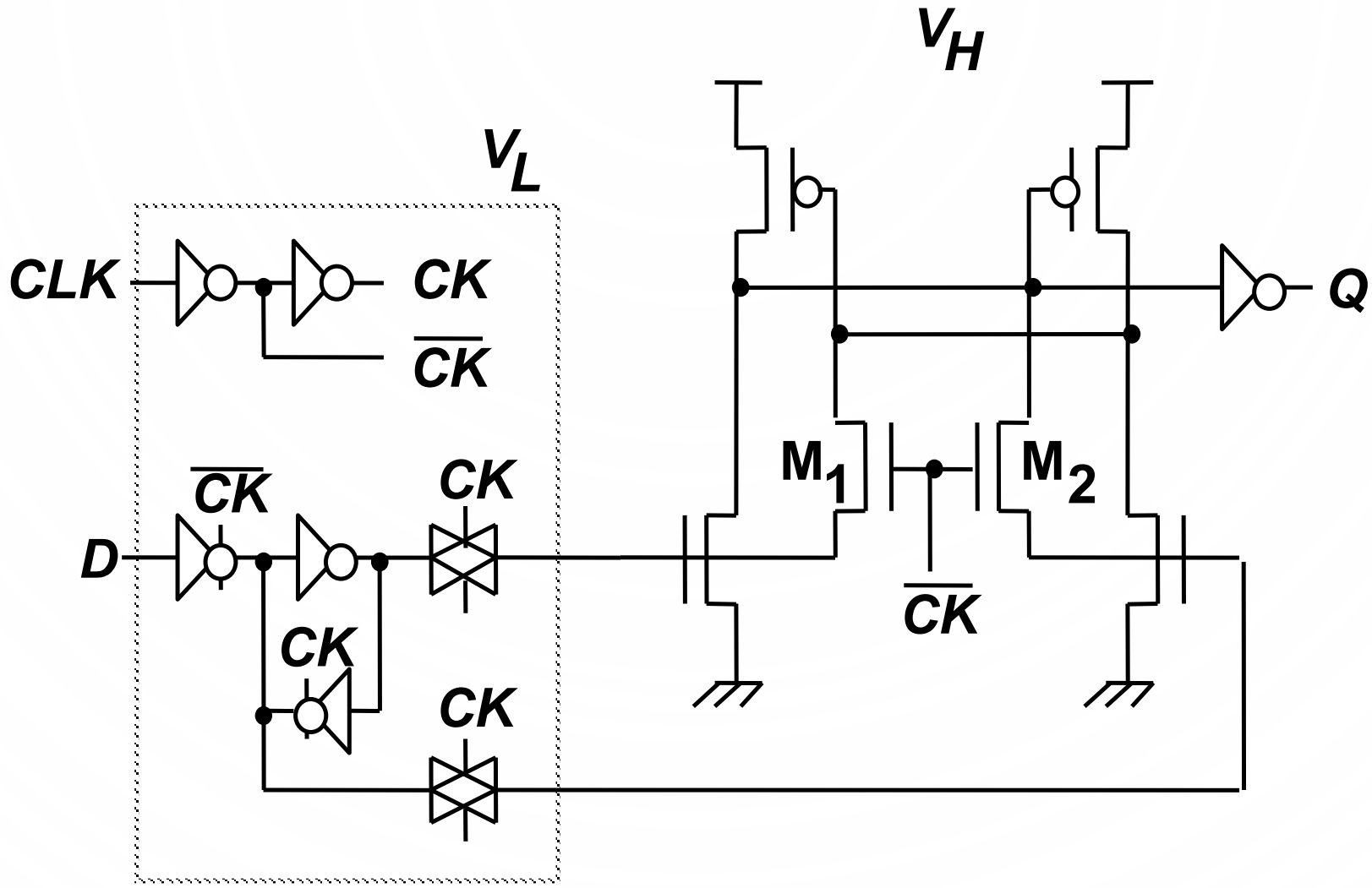


Usami'98

Layout:



Level-Converting Flip-Flop

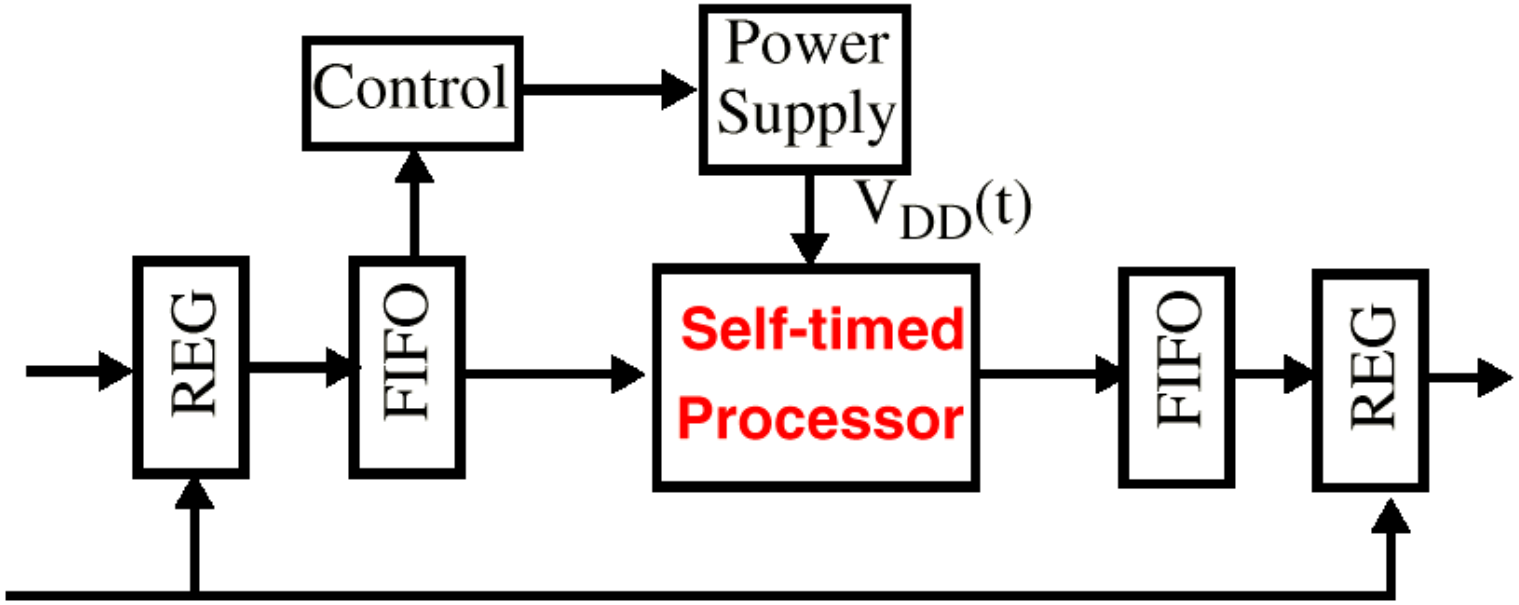


Dynamic Voltage-Frequency Scaling (DVFS)

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

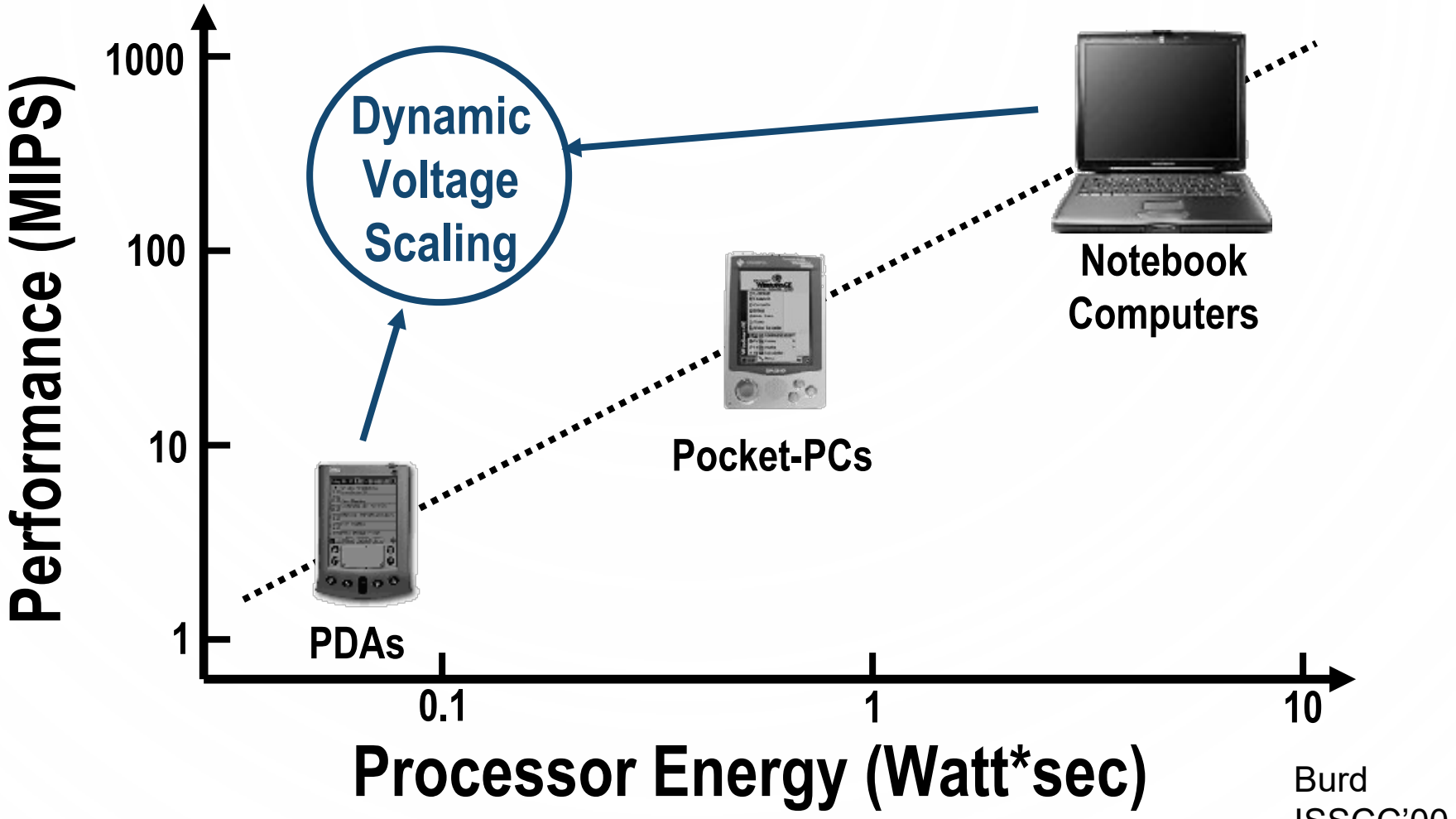


Exploit Data Dependent Computation Times To Vary the Supply

from [Nielsen94]

(*IEEE Transactions on VLSI Systems*)

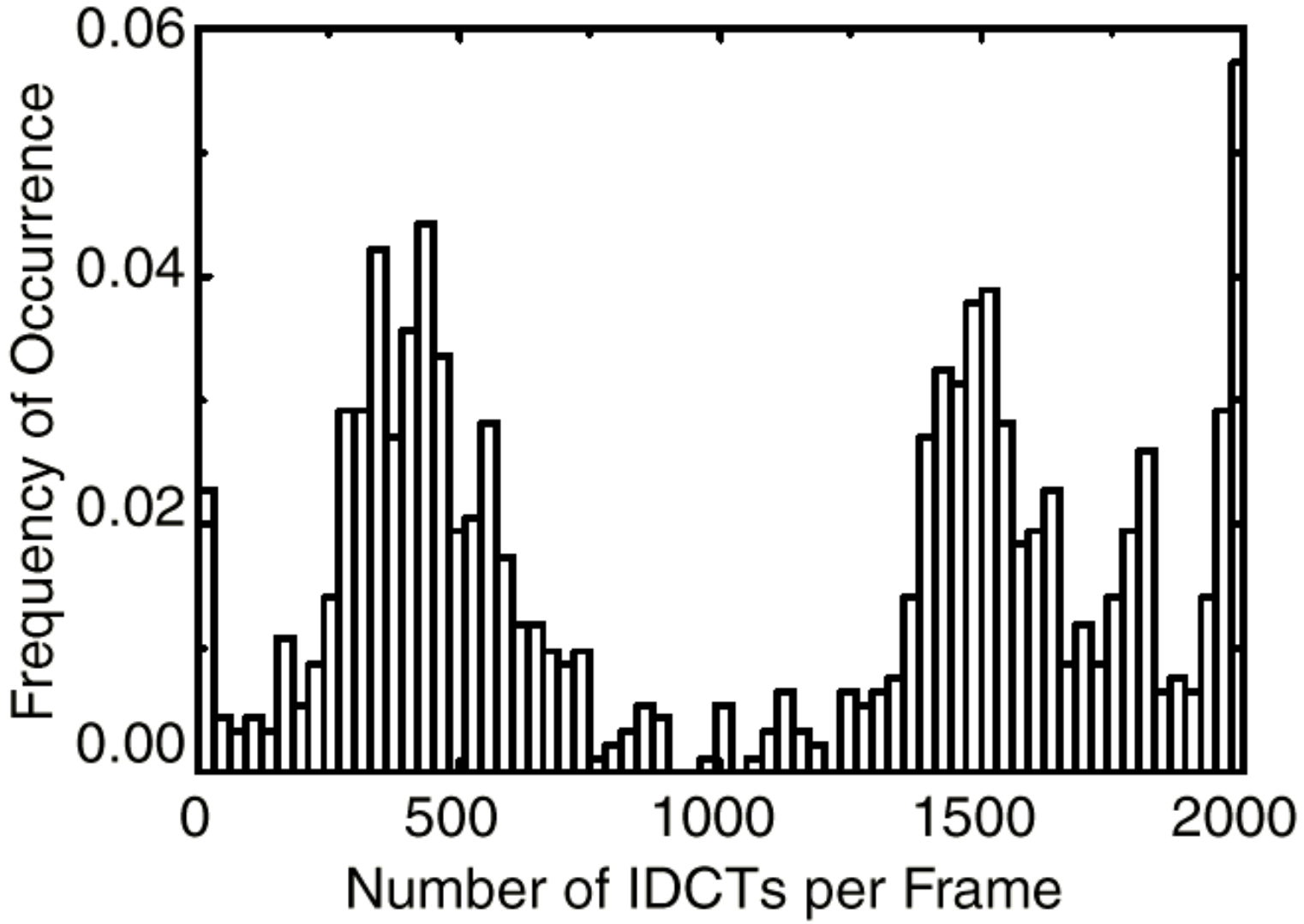
Processors for Portable Devices



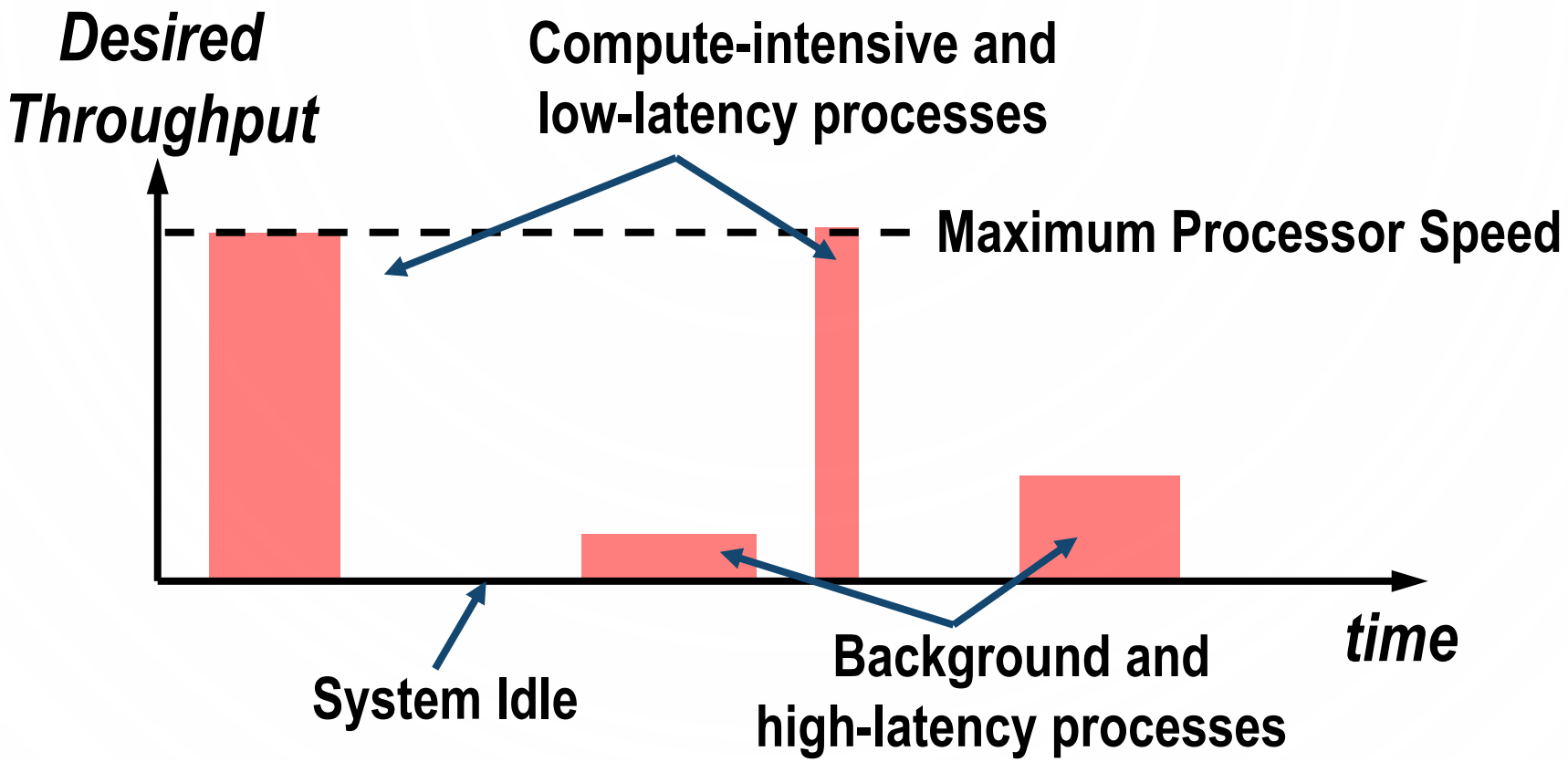
Burd
ISSCC'00

- **Eliminate performance ↔ energy trade-off**

Typical MPEG IDCT Histogram



Processor Usage Model

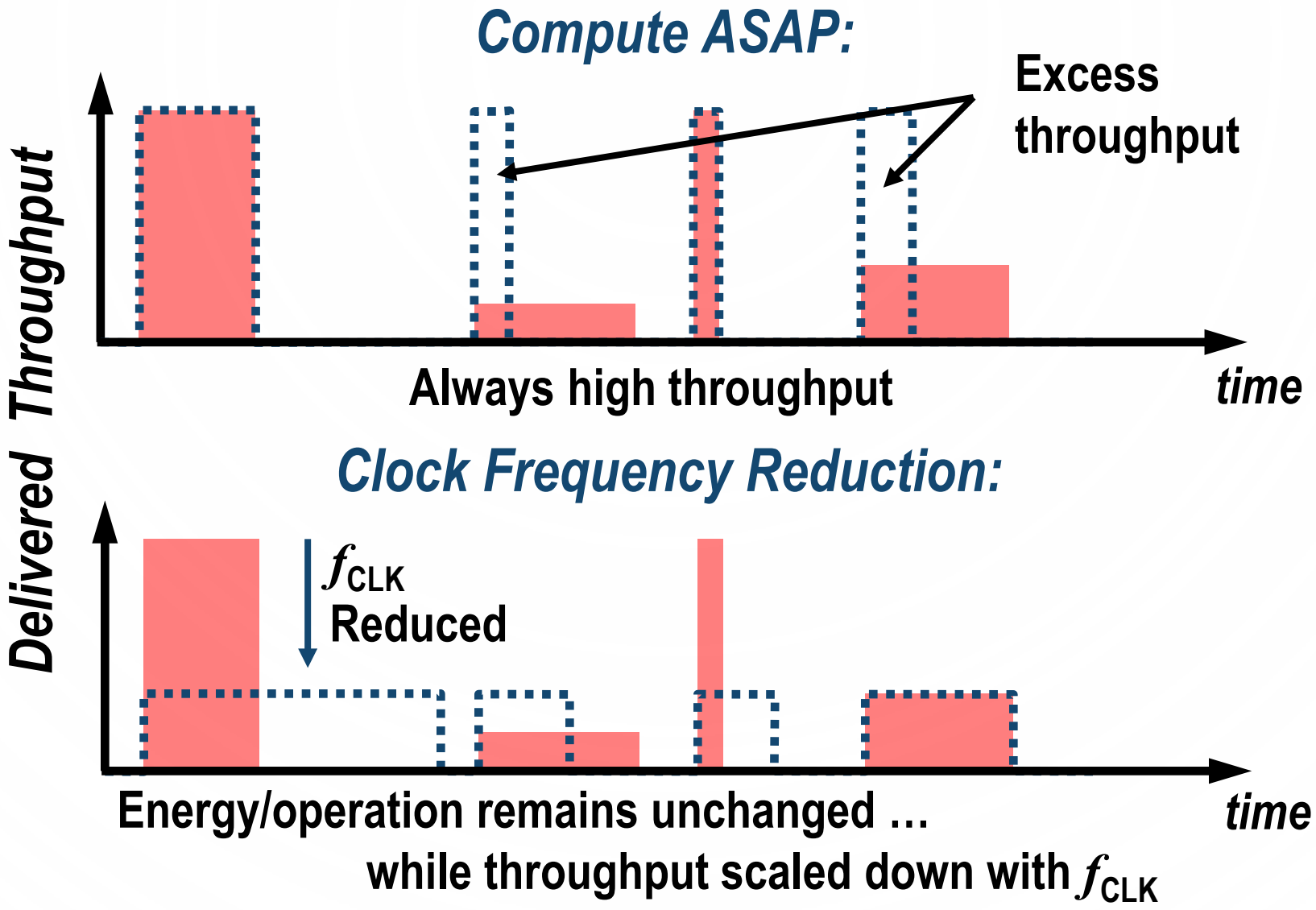


System Optimizations:

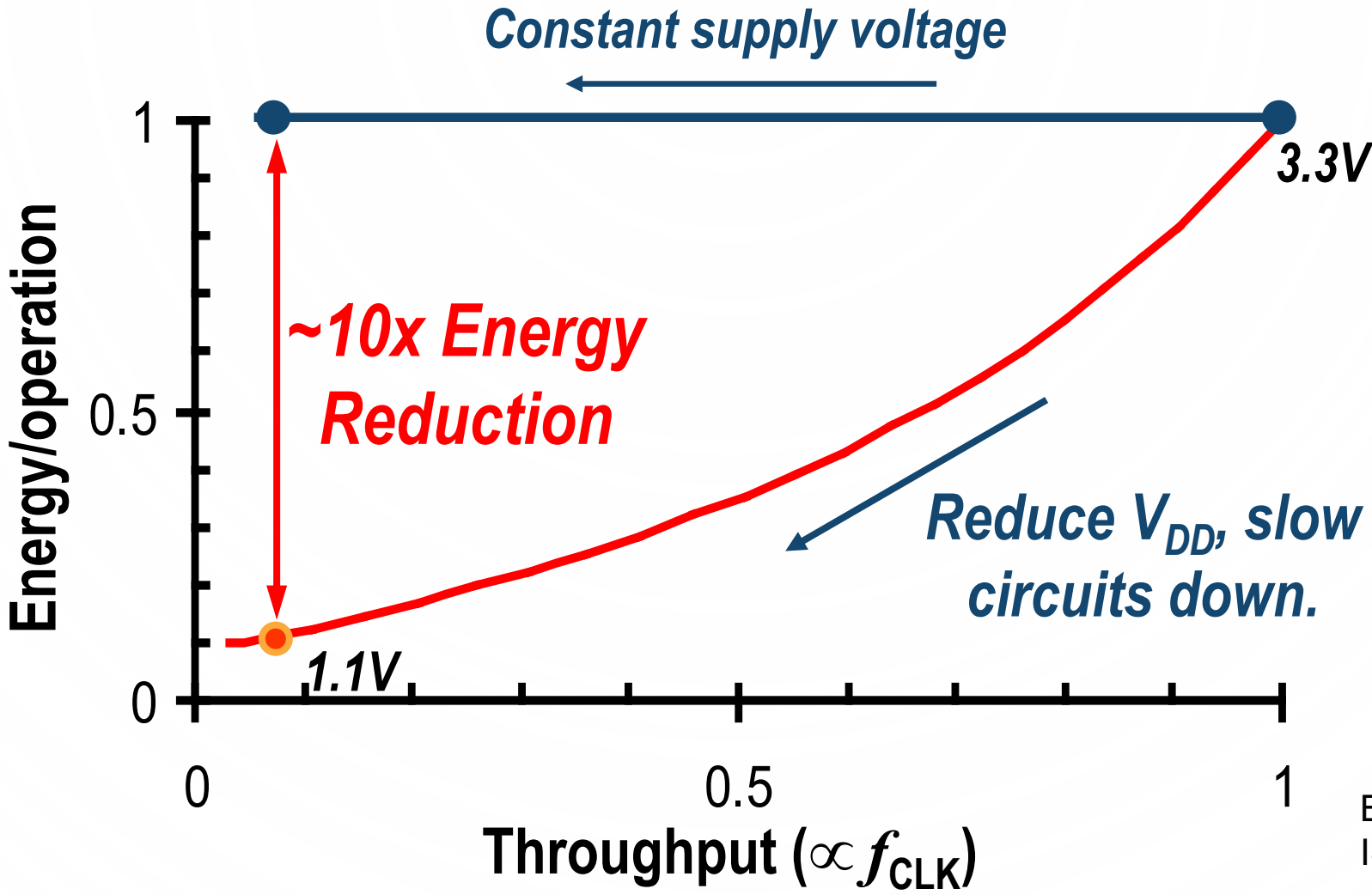
- **Maximize Peak Throughput**
- **Minimize Average Energy/operation**

Burd
ISSCC'00

Common Design Approaches (Fixed VDD)

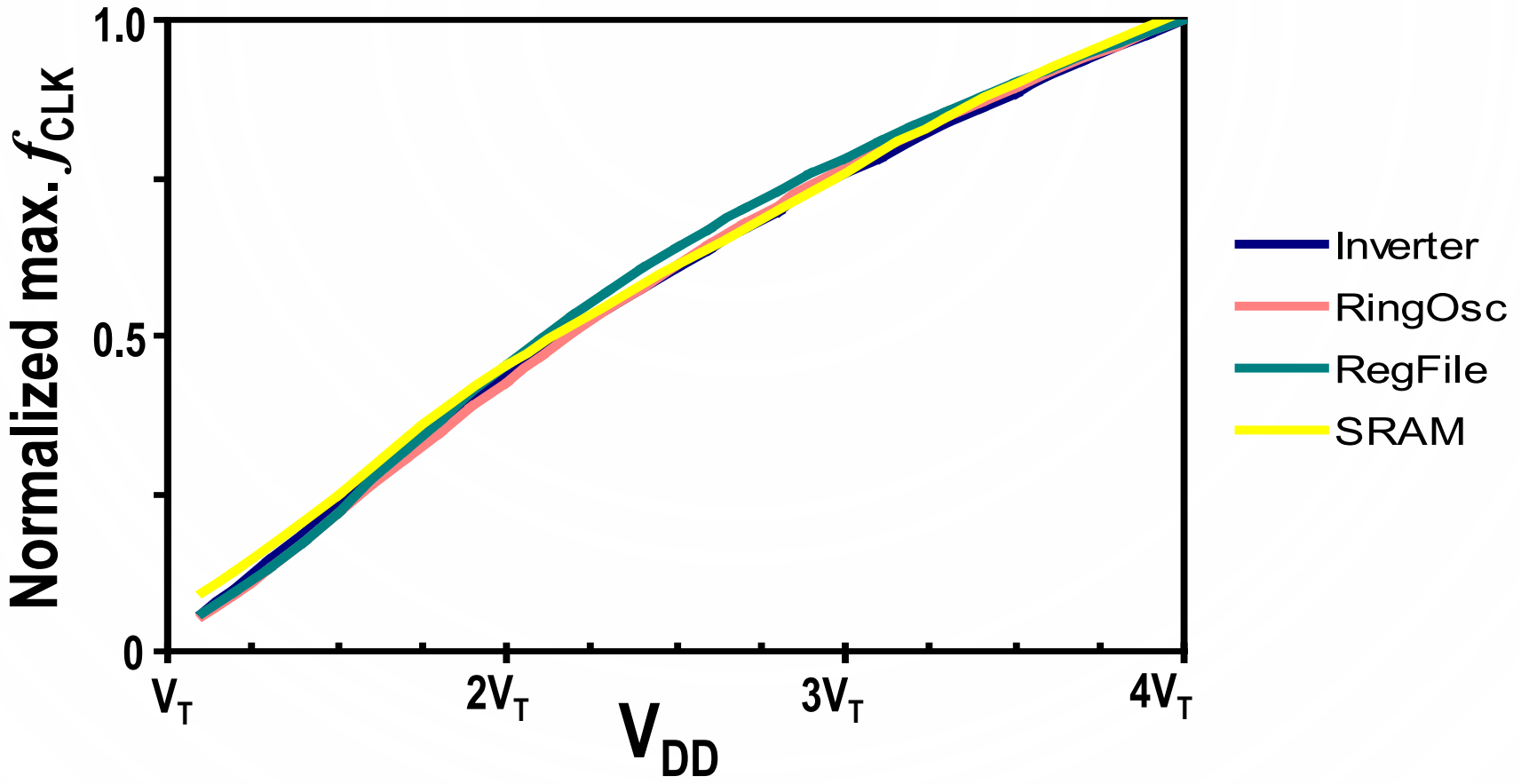


Scale V_{DD} with Clock Frequency



Burd ISSCC'00

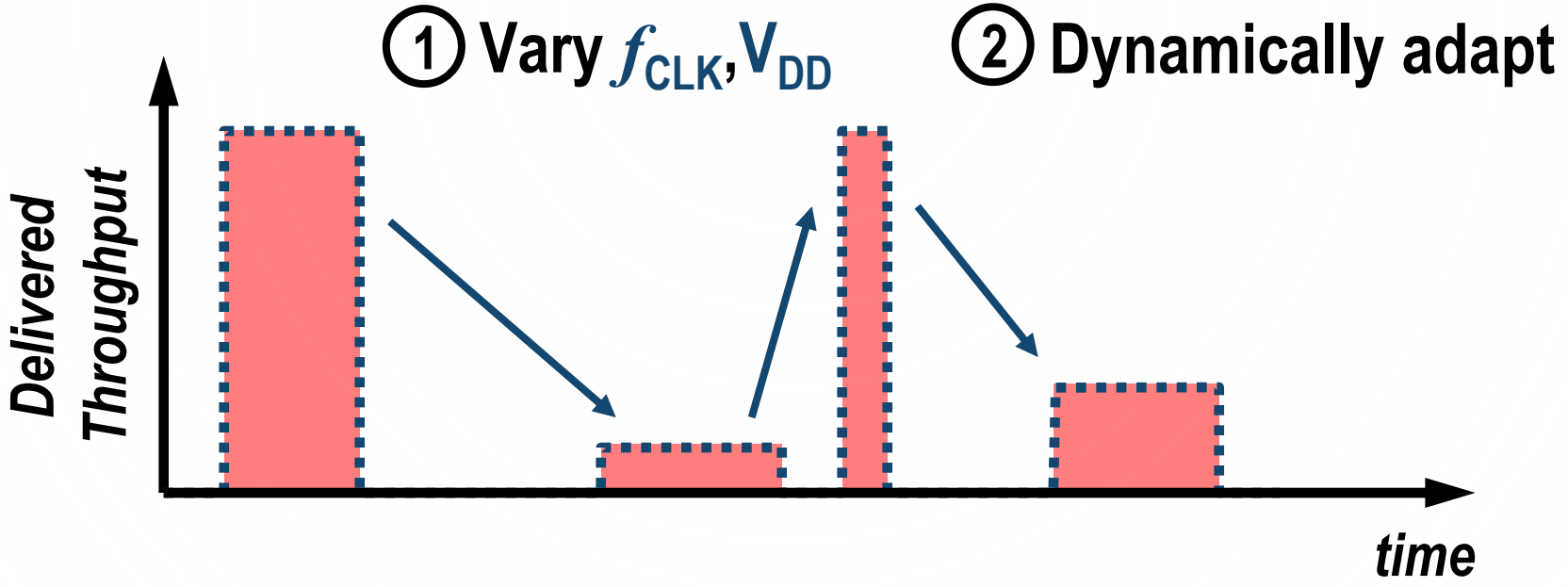
CMOS Circuits Track Over V_{DD}



← Delay tracks within +/- 10% →

Burd
ISSCC'00

Dynamic Voltage Scaling (DVS)



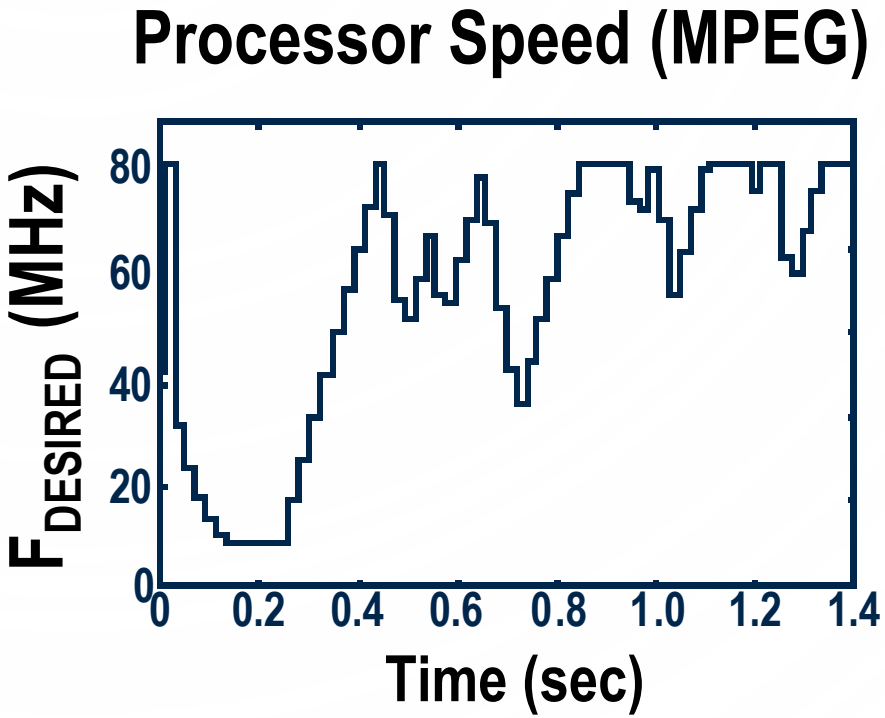
Burd
ISSCC'00

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

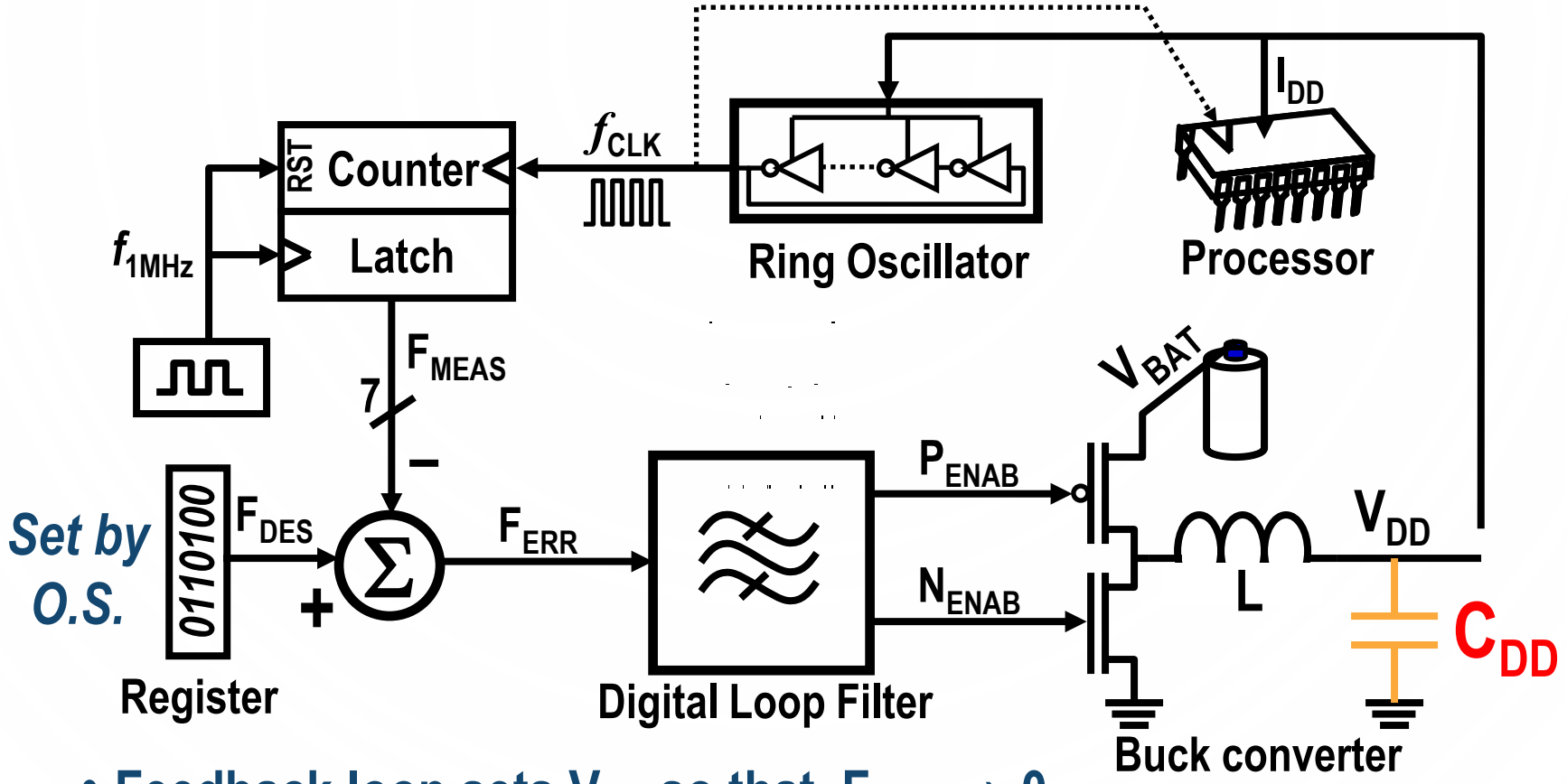
Operating System Sets Processor Speed

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

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



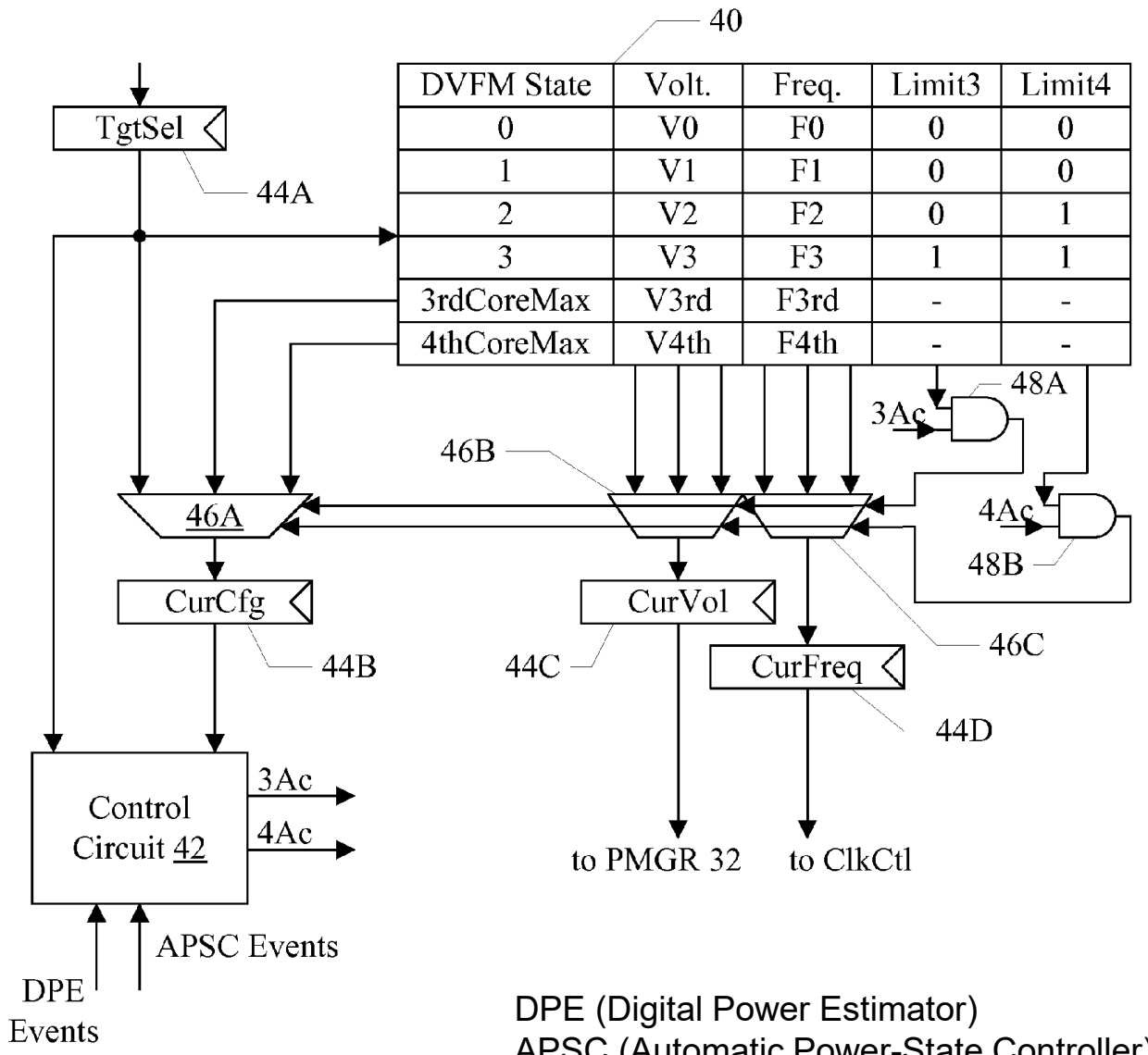
Converter Loop Sets V_{DD}, f_{CLK}



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

Burd
ISSCC'00

Power Estimator Example for HW-Based DVFS Control



DPE (Digital Power Estimator)
APSC (Automatic Power-State Controller)

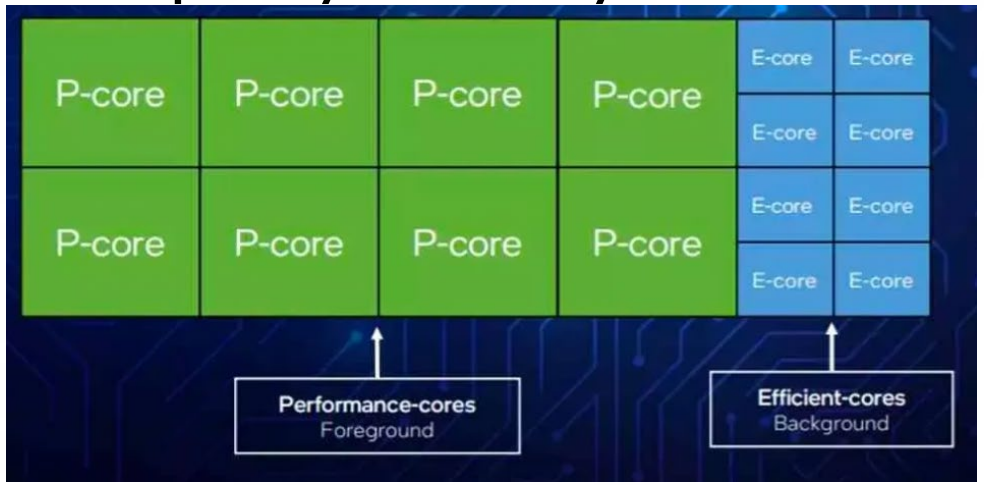
- Receives ongoing counts from each CPU of "significant" events
  - eg how many vector instructions executed per cycle, aggregates these to a per-CPU instantaneous power estimate
- Sends per-CPU throttle requests as necessary.
- The number of "power events" over some period of time is tracked
  - If this is too high the fact is reported to the PMU and PMGR, and the frequency/voltage settings for the offending core(s) are reduced; and similarly if a core has been operating within spec for some period of time, its frequency/voltage is allowed to rise.

DVFS Scheduling

- First stage – establish performance goals (achieve state X by time Y)
  - Use closed-loop performance controller to adjust DVFS to meet these performance goals.
  - E.g. performance described as a monotonic map from n-cores running at max frequency down to 1 core running at minimum frequency.
  - Move up/down this performance map depending on how the target is being met
- Second stage - energy and performance issues.
  - Track short-term power draw and temperature to back-off high-performance goals if needed
  - Track an energy per instruction metric and try to optimize this while not impairing the performance goals

DVFS as a Systems Problem for the OS

- Based on thread groups and matching target metrics to measured metrics, a map of the best performance schedule for the thread group is created (i.e. each thread gets mapped onto a P vs E core at certain voltage/frequency)
- These maps are then read by the thread scheduler (Thread Director on Adler Lake CPU, for example) which puts each thread into an E vs P queue, with the appropriate DVFS info attached to it. The queues are adjusted across different thread groups to ensure maximal occupancy of every core.



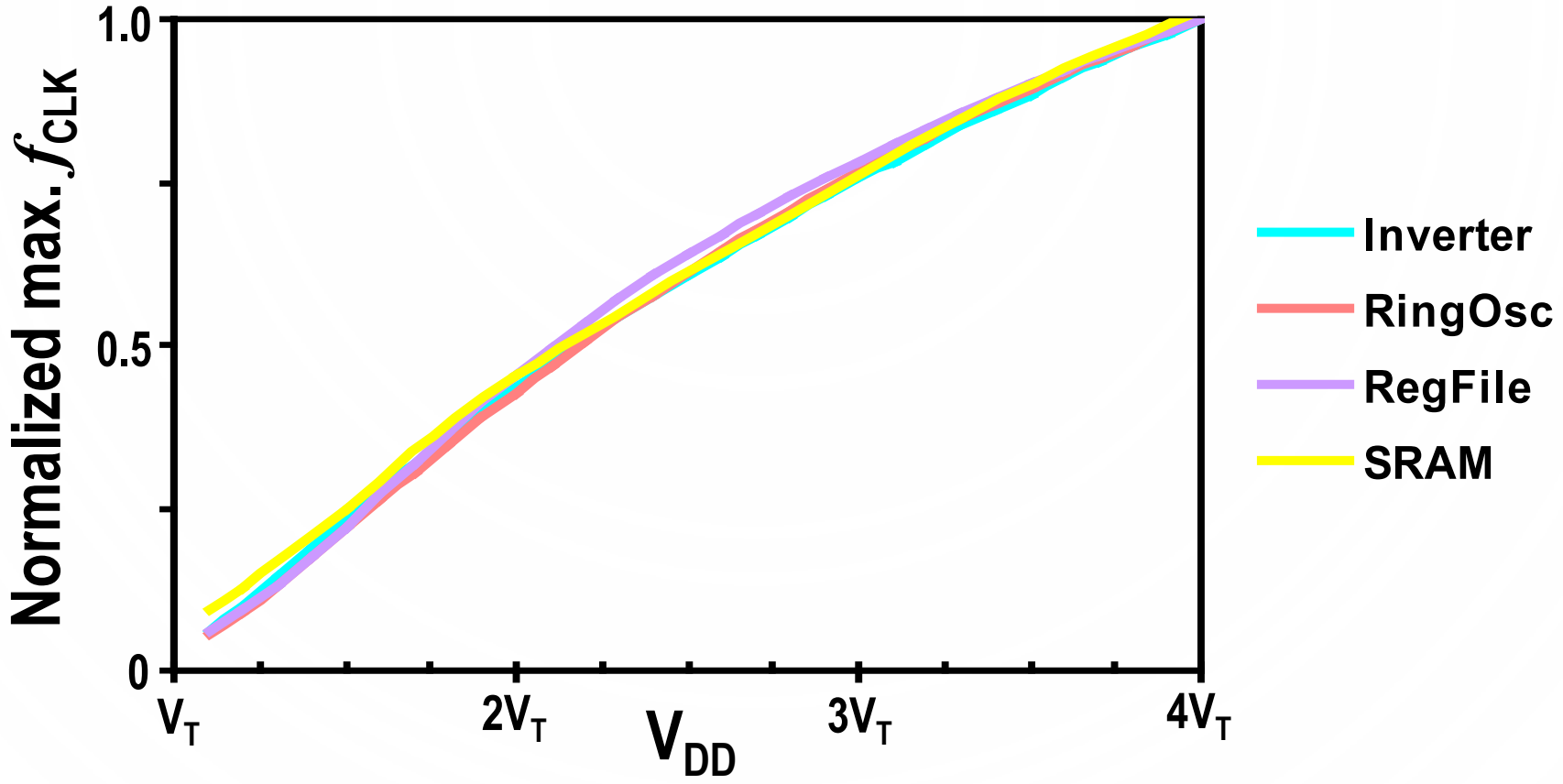
Adler Lake CPU

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}

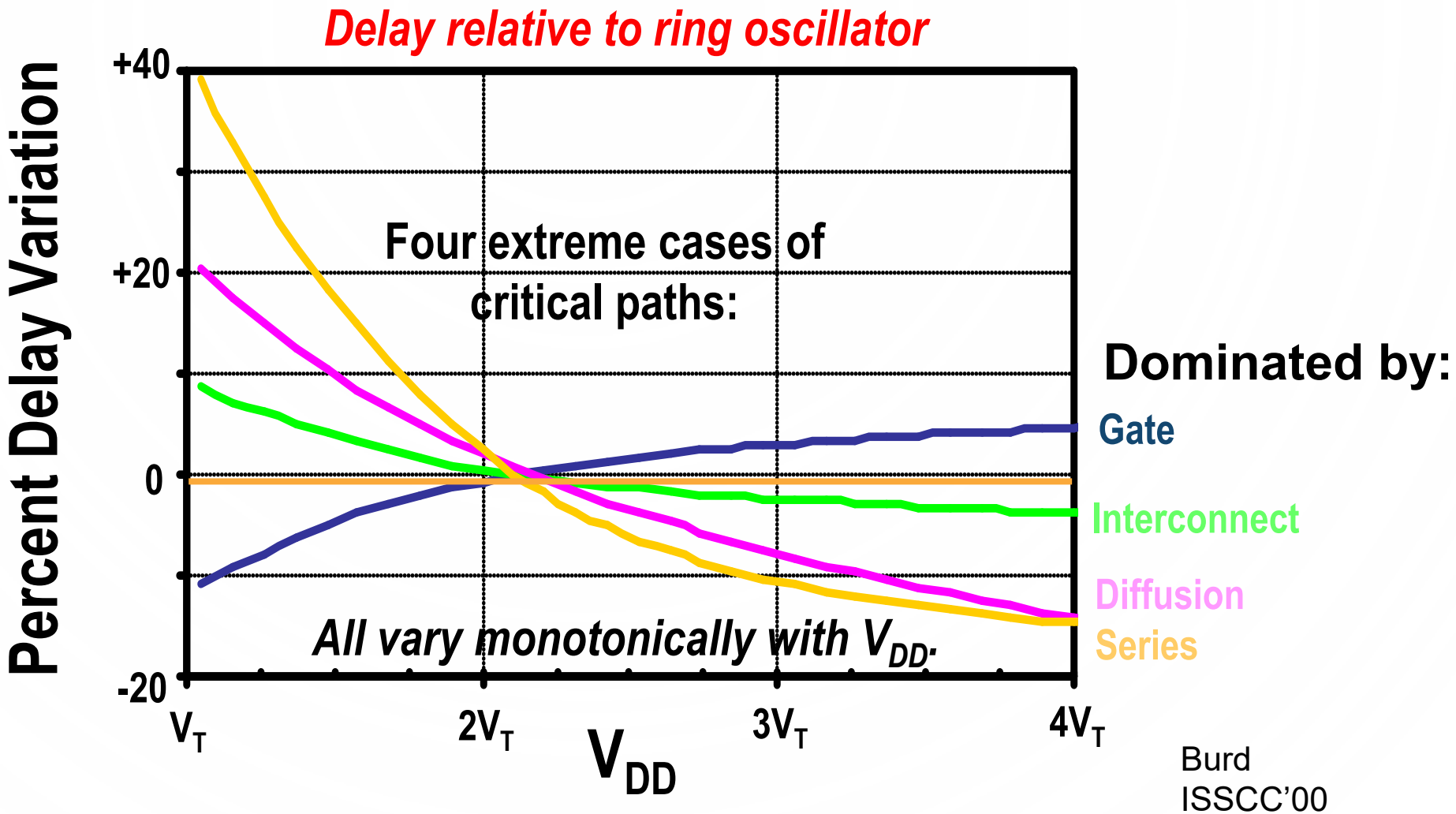
Delay Variation & Circuit Constraints



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

Burd
ISSCC'00

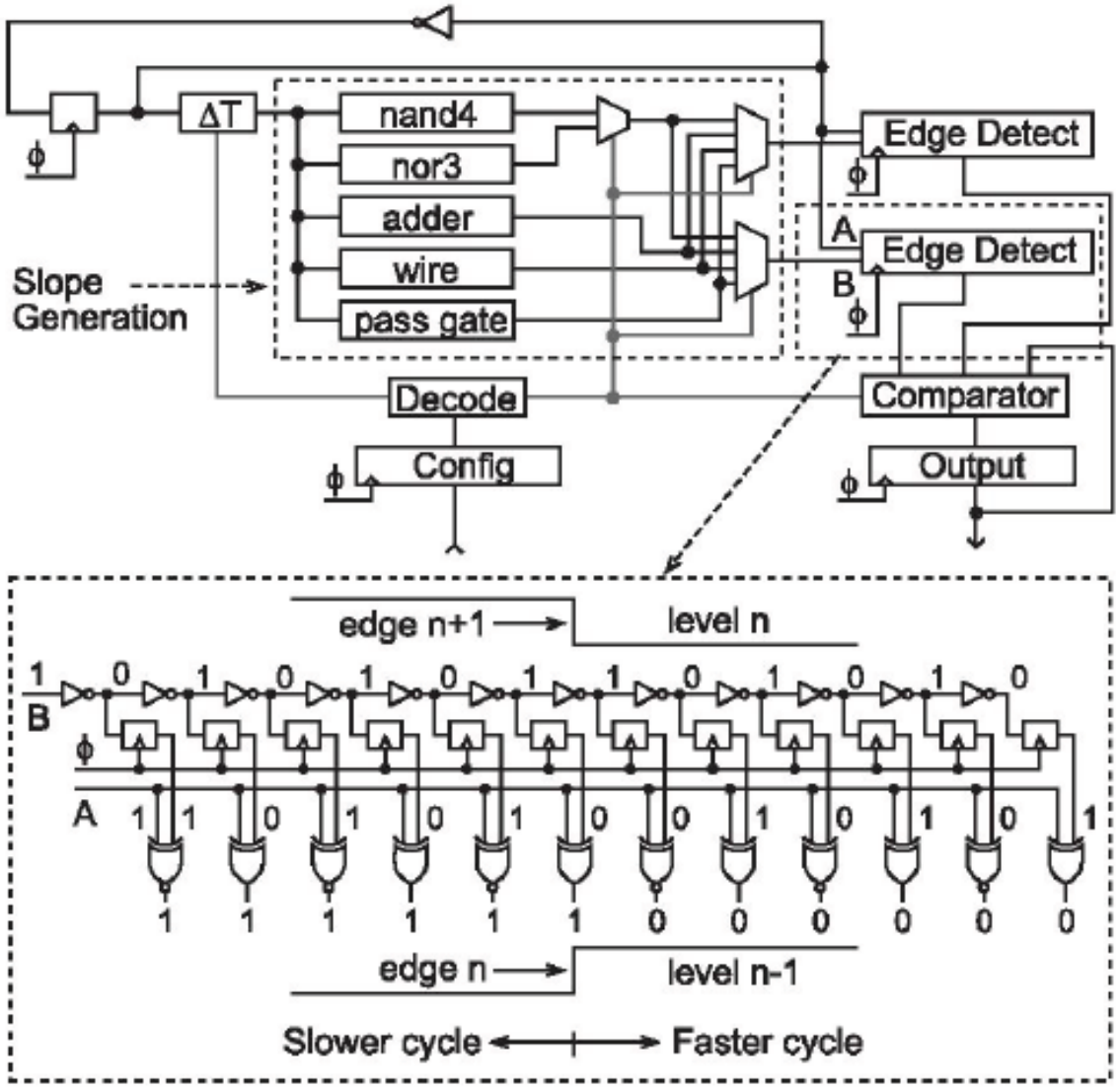
Relative Delay Variation



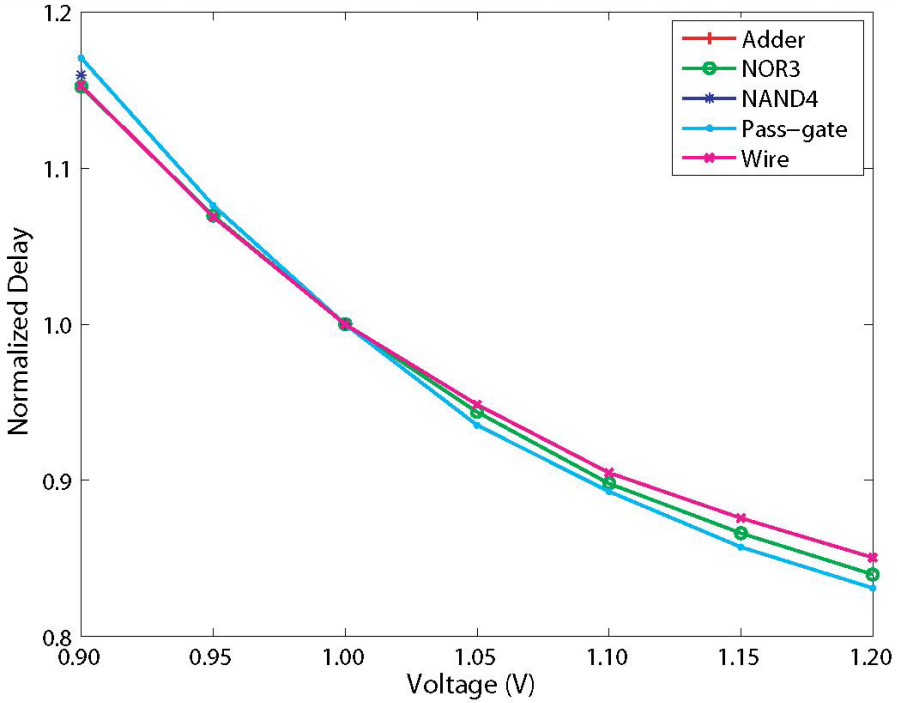
Burd ISSCC'00

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

Multiple Path Tracking



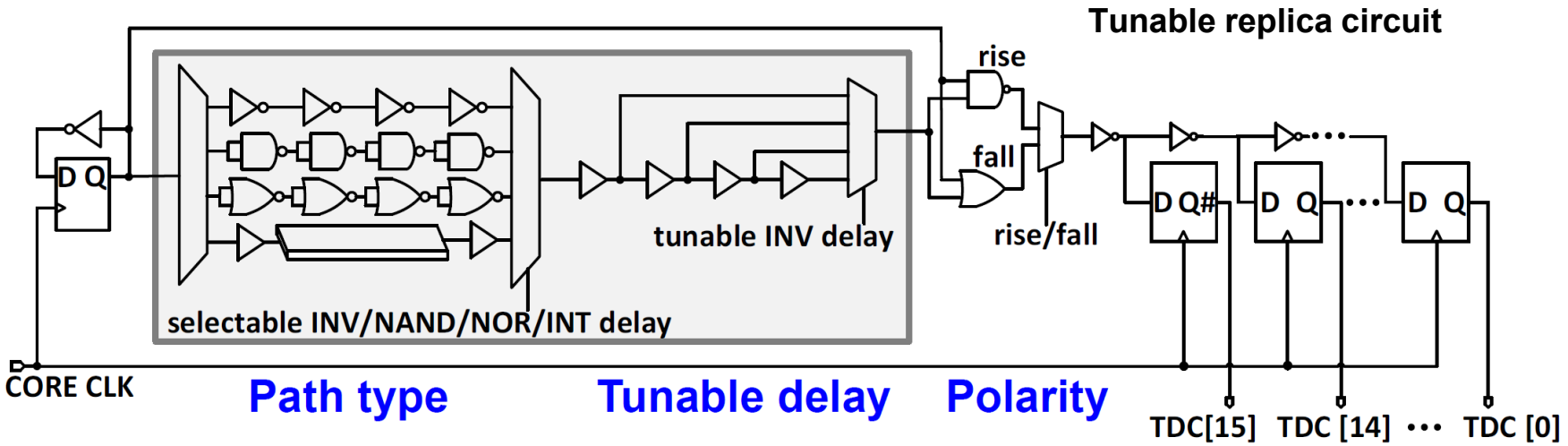
A. Drake, ISSCC'07



ps	adder	nor	nand	pass-gate	wire
core 0: σ	3.78	3.45	3.91	3.28	2.82
core 1: σ	3.55	3.16	2.25	3.55	4.21
chip: σ	4.09	4.03	3.90	4.80	4.10
core 0: Δ	10	9	11	10	7
core 0: Δ	12	9	6	10	12
core 0: Δ	13	11	14	16	15

Path delay variation at nominal voltage

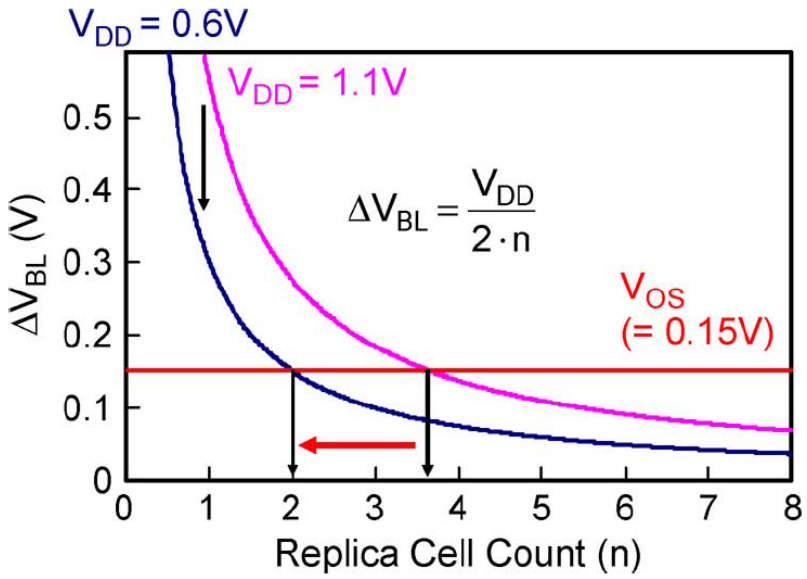
Multiple Path Tracking



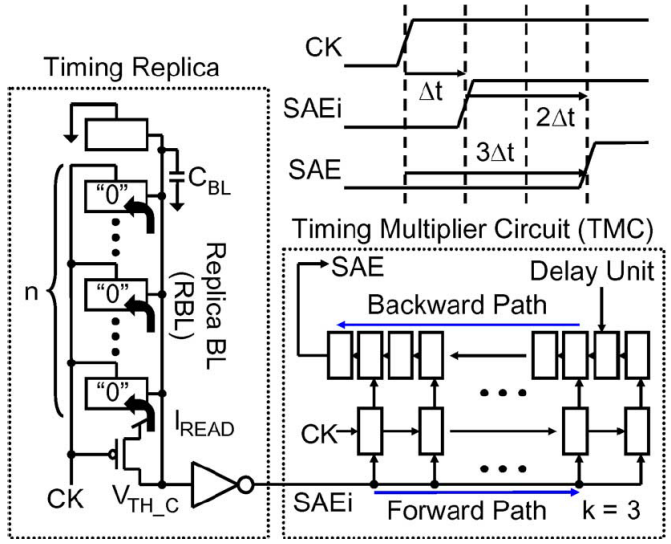
Cho, ISSCC'16

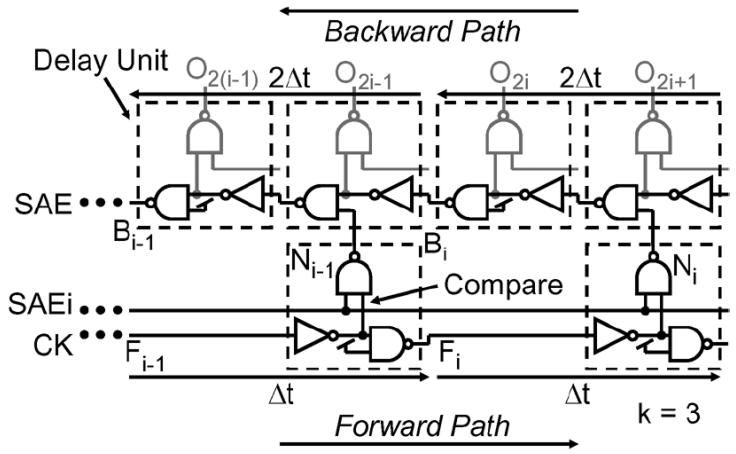
Tracking with SRAM in Critical Path

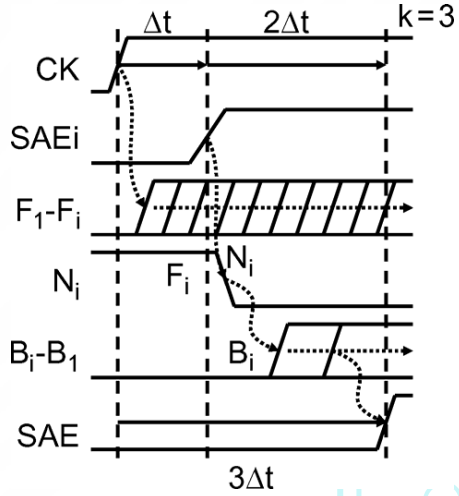
- **Mismatch between logic and SRAM**



- **SRAM multiplicative replica**

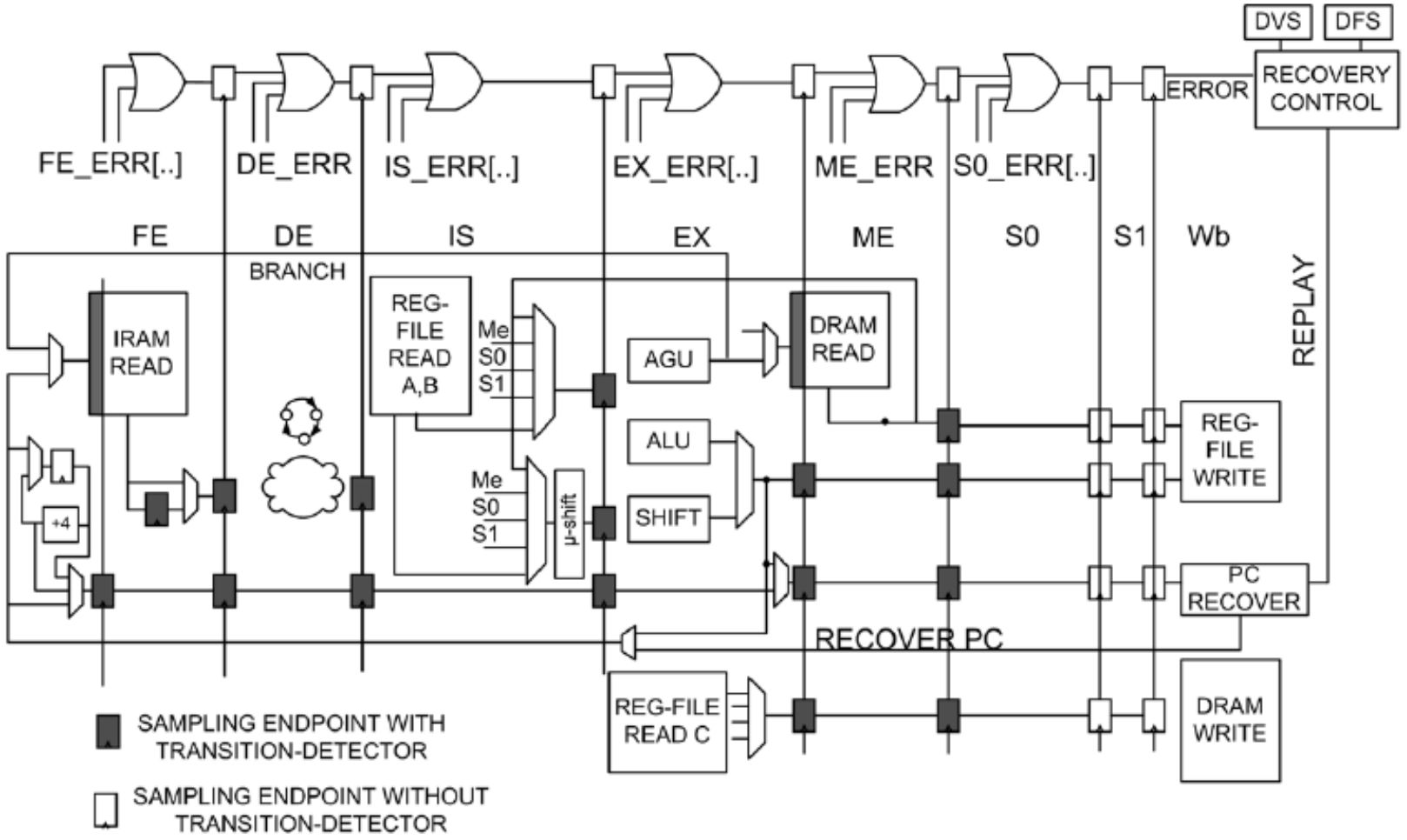






Niki, JSSC'11

Alternative: Error Detection



Bull, ISSCC'2010

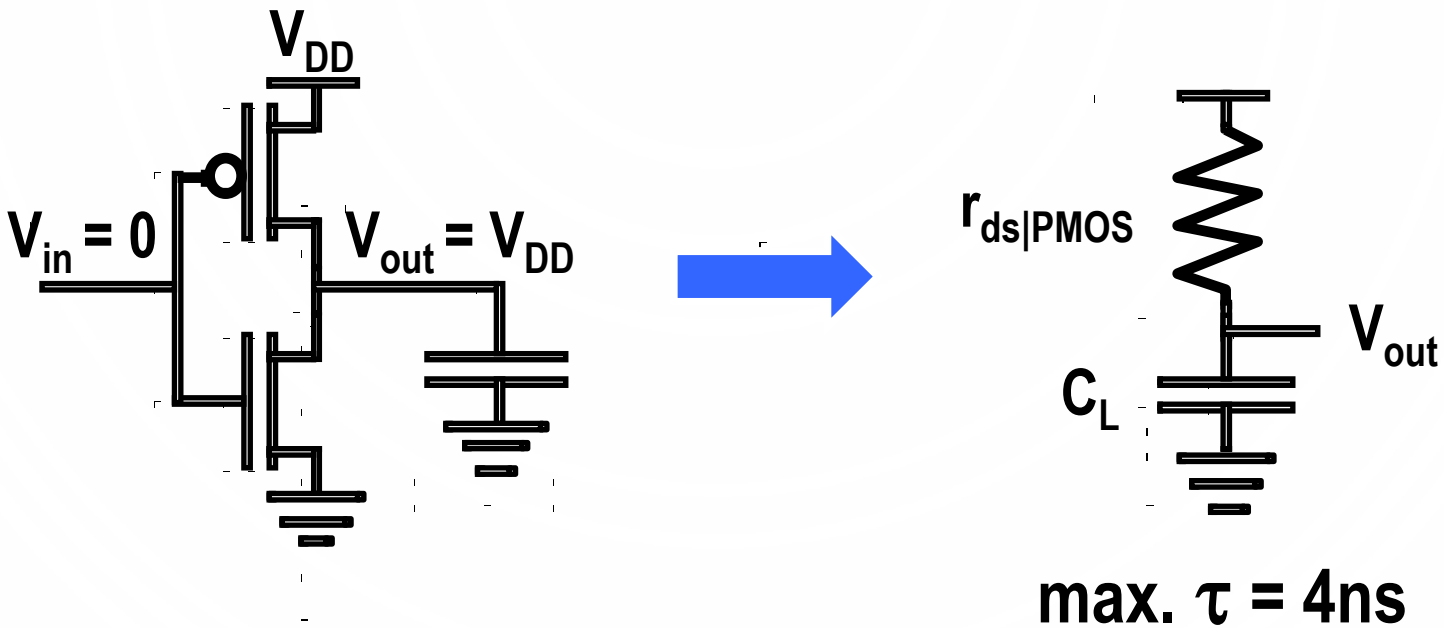
Design for Dynamically Varying VDD

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

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

Circuits continue to properly operate as V_{DD} changes

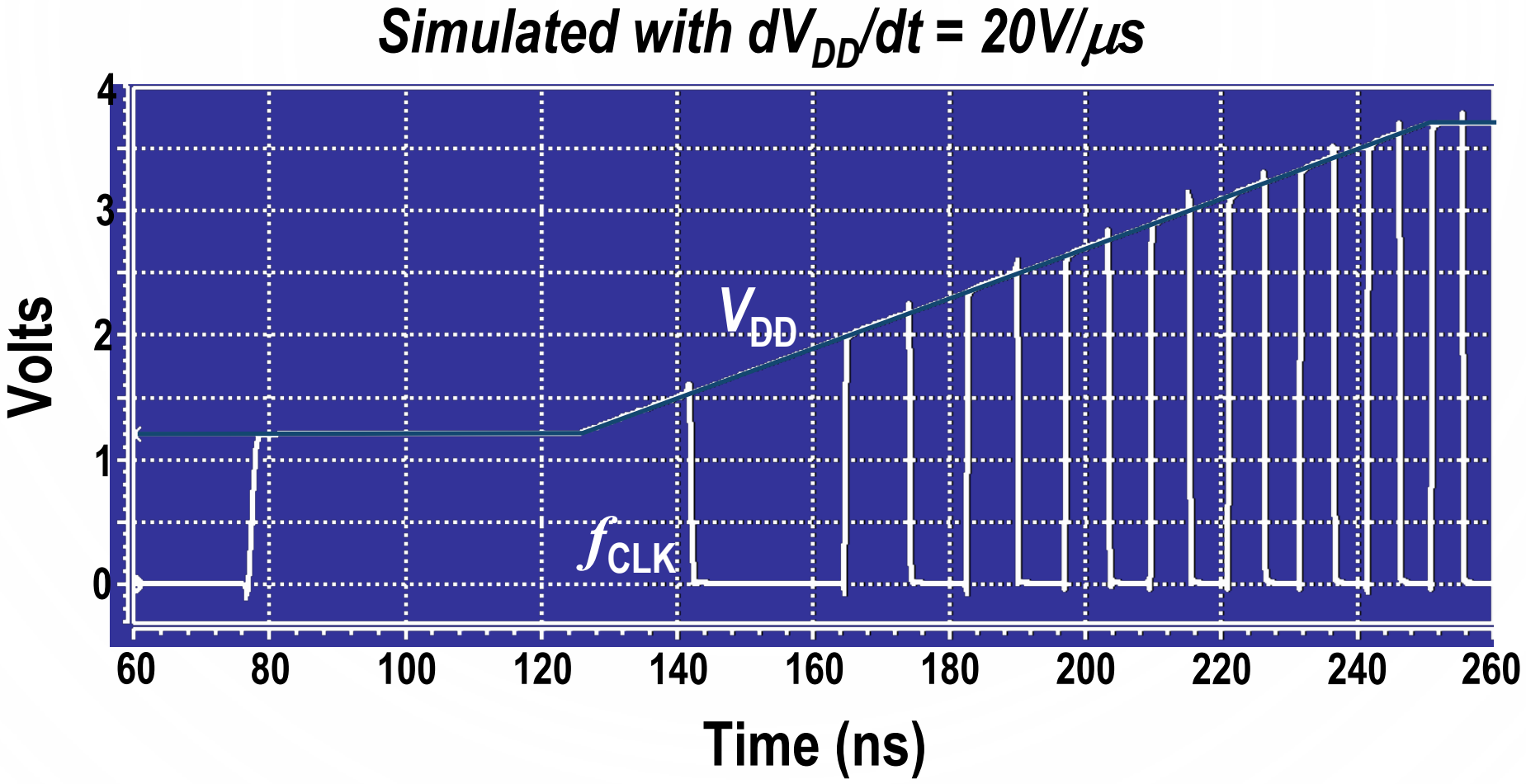
Static CMOS Logic



0.6mm CMOS: $|dV_{DD}/dt| < 200V/ms$

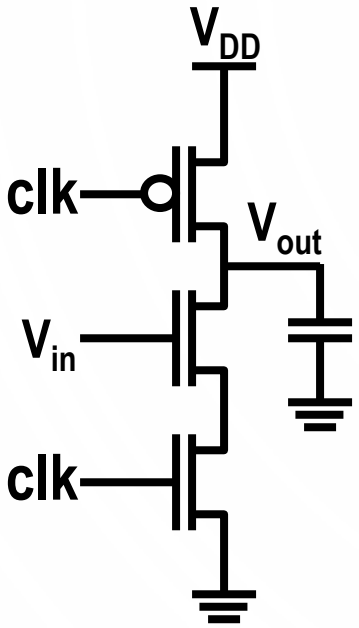
- **Static CMOS robustly operates with varying V_{DD}.**

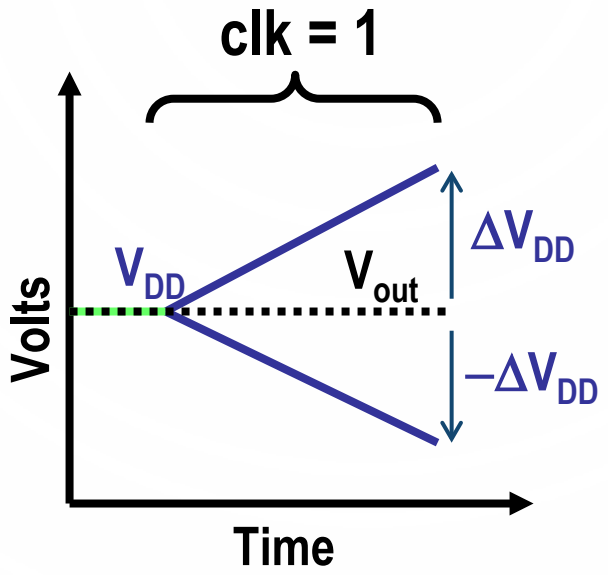
Ring Oscillator



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

Dynamic Logic





Errors

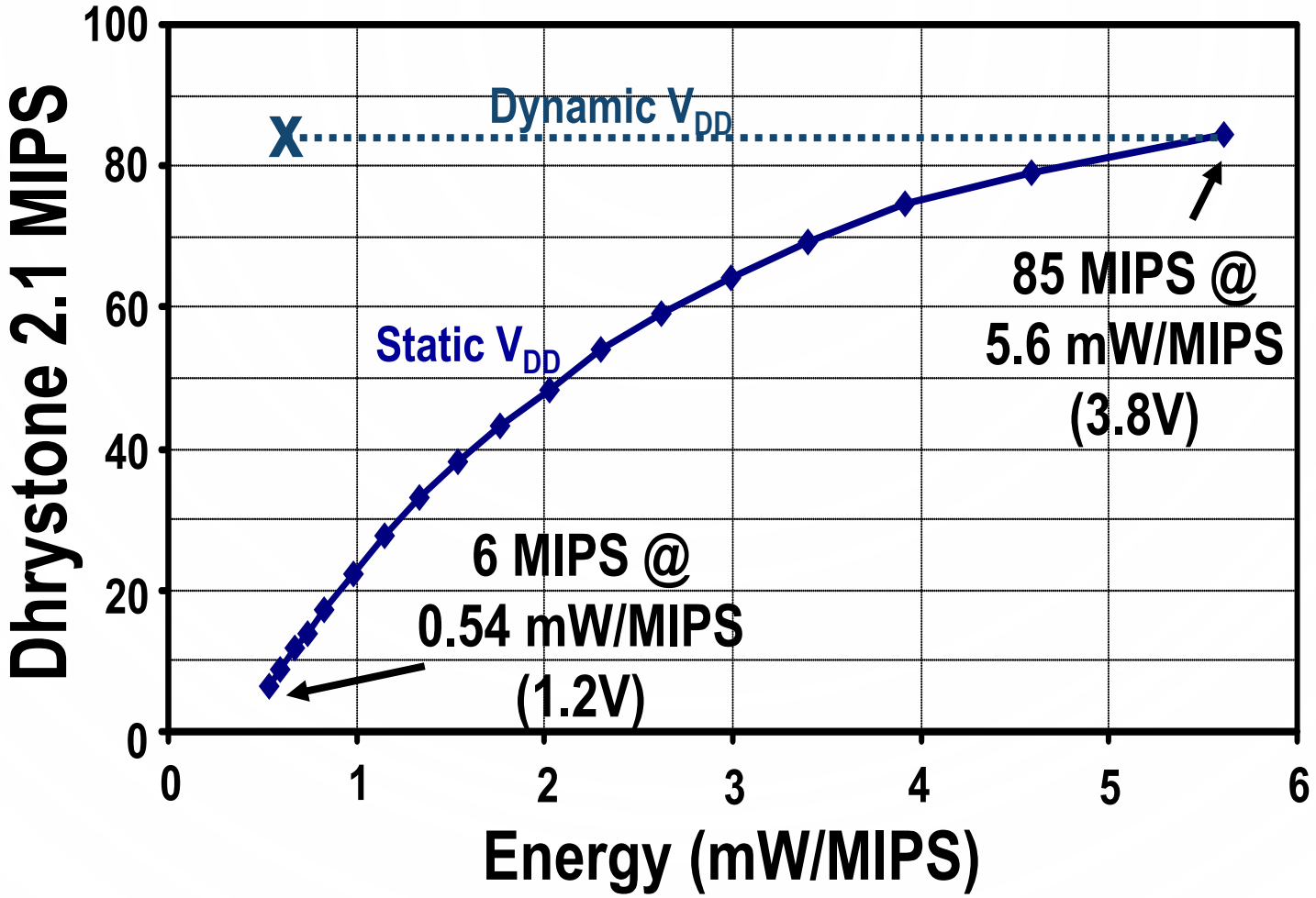
False logic low: $DV_{DD} > V_{TP}$

Latch-up: $DV_{DD} > V_{be}$

0.6mm CMOS: $|dV_{DD}/dt| < 20V/ms$

- Cannot gate clock in evaluation state.
- Tri-state busses fail similarly → Use hold circuit.

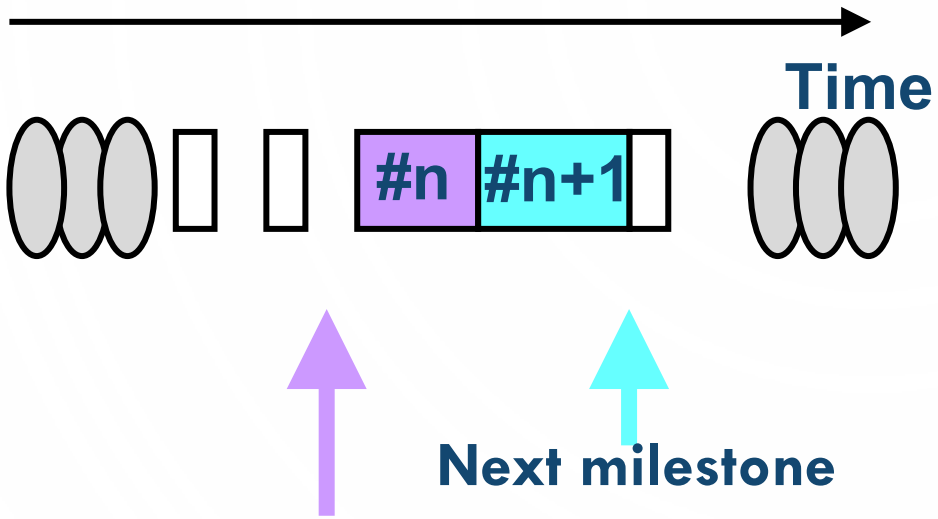
Measured System Performance & Energy

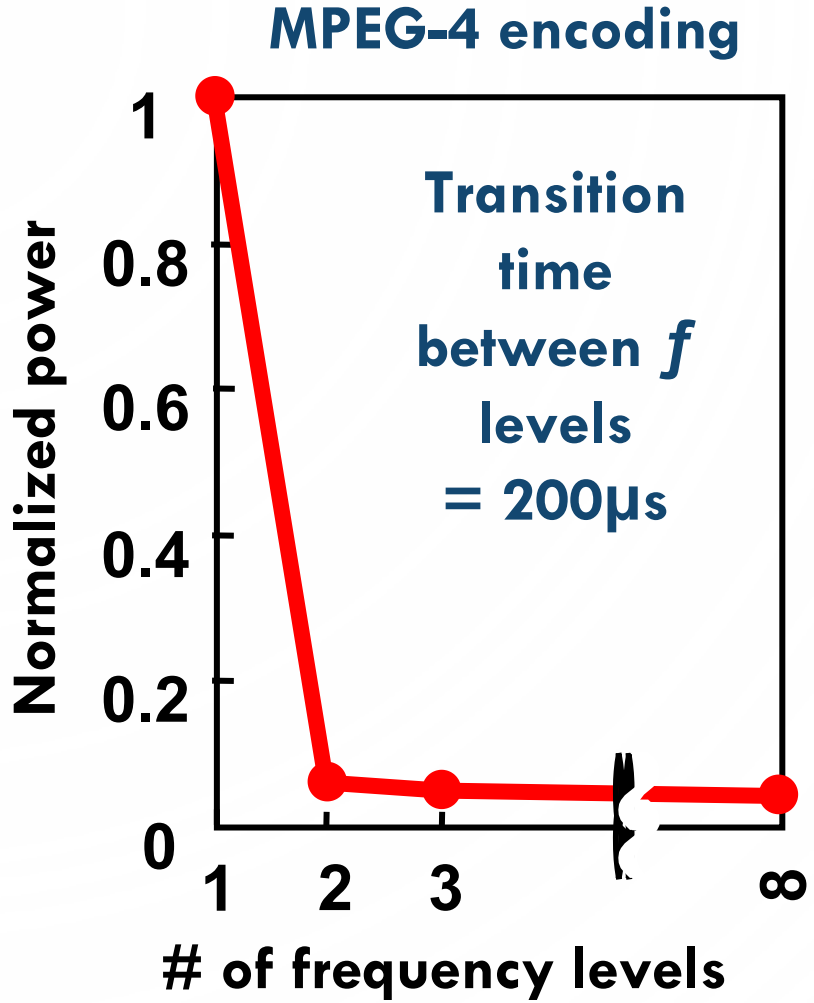


- **Dynamic operation can increase energy efficiency > 10x.**

Burd
ISSCC'00

V_{DD}-Hopping

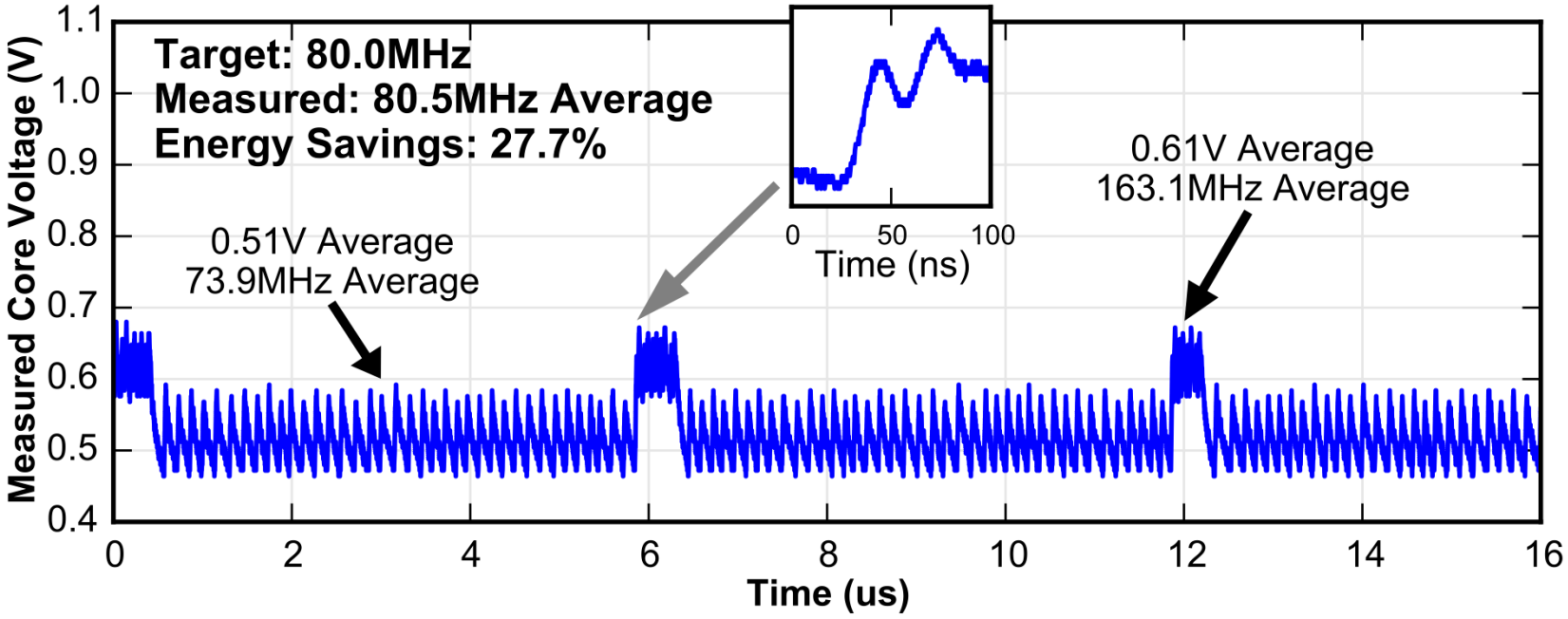




Application slicing and software feedback guarantee real-time operation.

Two hopping levels are sufficient.

Dithering Between Supply Levels

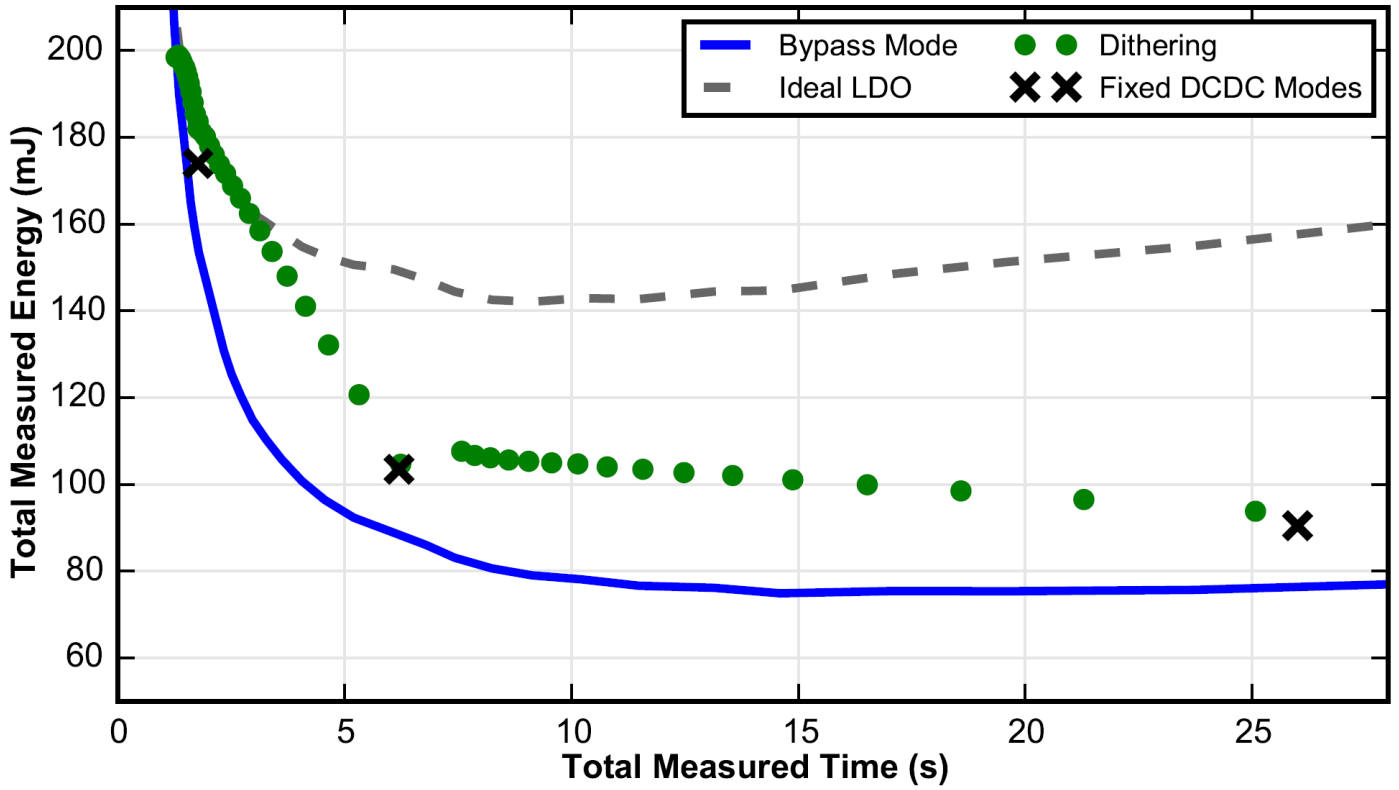


- Done with switched-capacitor DC-DC converters which efficiently work only at discrete levels

Keller et al, ESSCIRC'16

Dithering Between Supply Levels

- Dithering fills in between fixed DC-DC modes



Keller et al, ESSCIRC'16

Summary

- Multiple supply voltages
- Dynamic voltage scaling

Next Lecture

- Low-power design
  - Clock gating
  - Leakage management