

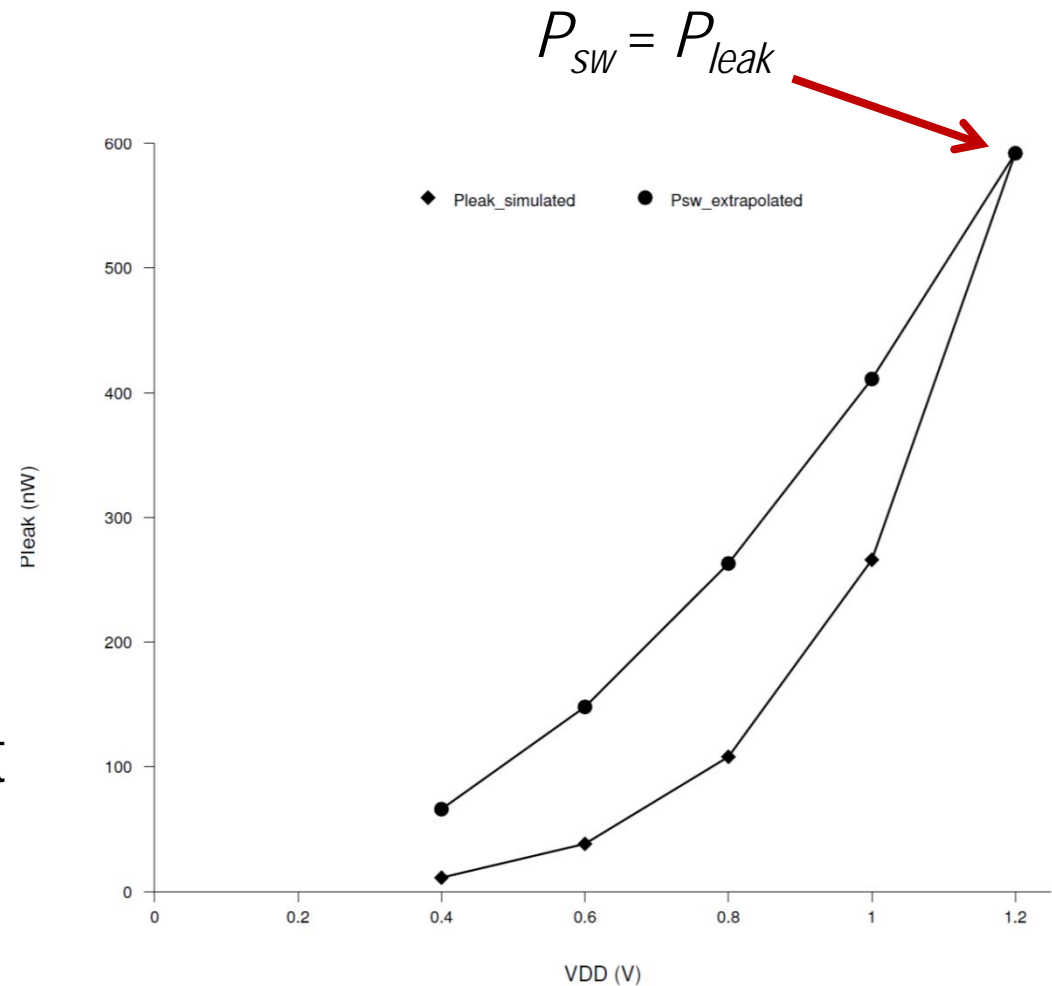
# Energy-Aware Computing: Low-Power Circuit Techniques

Per Larsson-Edefors

Computer Science and Engineering  
Chalmers University of Technology

# Two Mechanisms of Power Dissipation

- In previous lectures, we've seen two distinct mechanisms causing power to be dissipated:
  - Dynamic - switching.
  - Static - subthreshold.
- Important to note that both mechanisms benefit more than linearly from reduced  $V_{DD}$ .



$P_{leak}$  simulated,  $P_{sw}$  extrapolated from 1.2 V.

# Common Low-Power Circuit Techniques

- Body biasing
- Multi-  $V_T$
- Multi-  $V_{DD}$
- DVFS
- Clock gating
- Power gating
- But beside the above techniques, designers always strive to make systems power efficient at all levels, e.g., by
  - minimizing switched capacitance.
  - balancing IC technology used and performance needs.

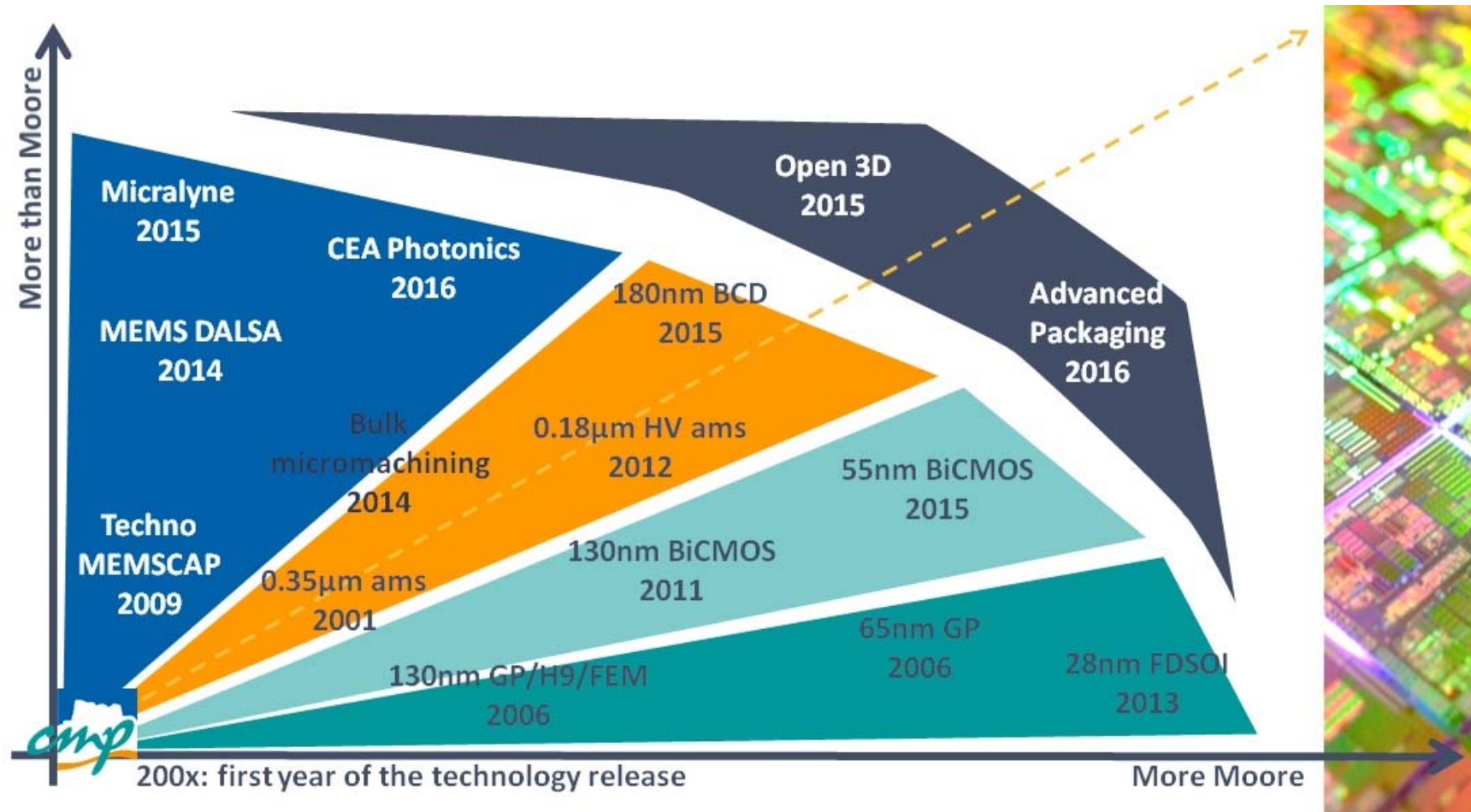
# Best Design Practice for Power Reductions

- Priority on  $P_{sw}$  ( $P_{leak}$  requires more invasive solutions).
- Consider  $P_{sw} = f \alpha C V_{DD}^2$ . Broadly, implementation decisions to reduce  $P_{sw}$  via ...
  - $f$  and  $V_{DD}$  are system wide and need to be negotiated early on in a project.
  - the switched capacitance ( $\alpha C$ ) can be handled at later implementation stages, at RTL and gate level (slide 18 previous lecture).
- Approximate computing:  
Inspired by SNR-guided DSP implementation, adjust data precision to application needs.

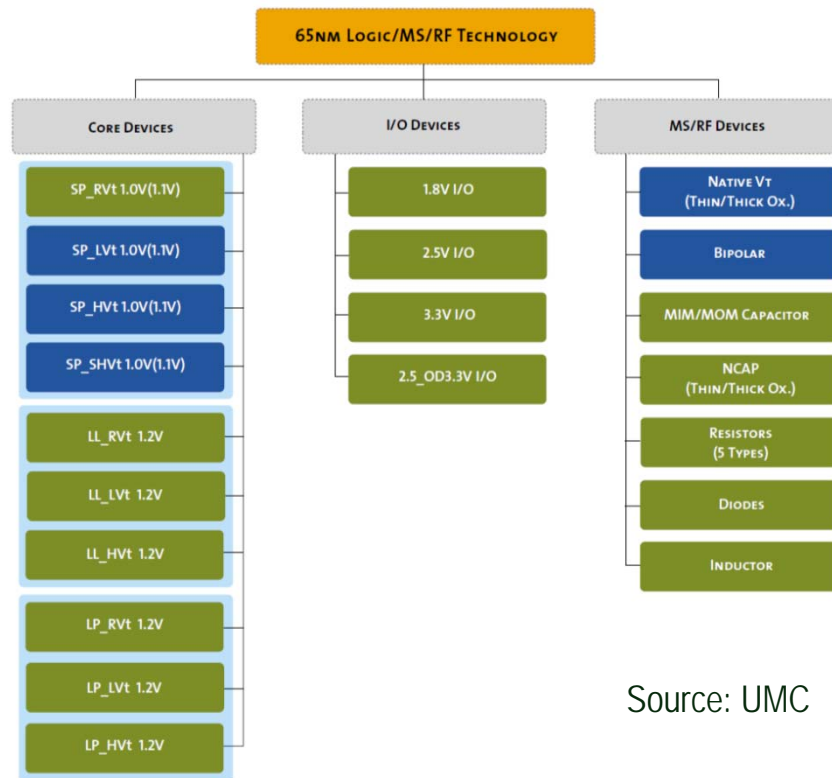
# Technology Flavors

- Select adequate IC technology.
  - Main question: How powerful need the transistors be in terms of current delivery  $I_{ON}$ ? Powerful  $\Rightarrow$  power dissipating!
- Different flavors:
  - Low power (LP) with higher  $V_T$ s.
  - High performance (in our case, GP) with lower  $V_T$ s and more aggressive design rules.
- Each flavor has got different  $V_T$  options:
  - Low  $V_T$  (LVT), Standard  $V_T$  (SVT), High  $V_T$  (HVT).
  - LP\_LVT: 0.40-0.49 V.
  - GP\_LVT: 0.25-0.36 V (slide 27 of my previous lecture).

# Technologies via CMP – our IC Broker

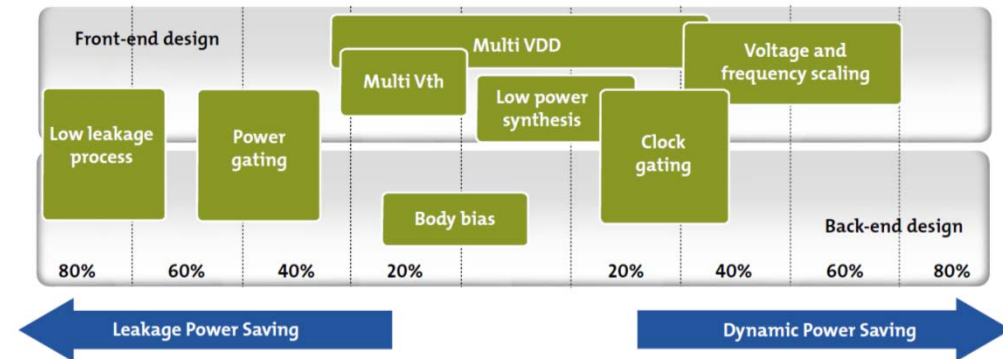


# A Smorgasbord of IC Options



Source: UMC

## LOW POWER DESIGN SUPPORT



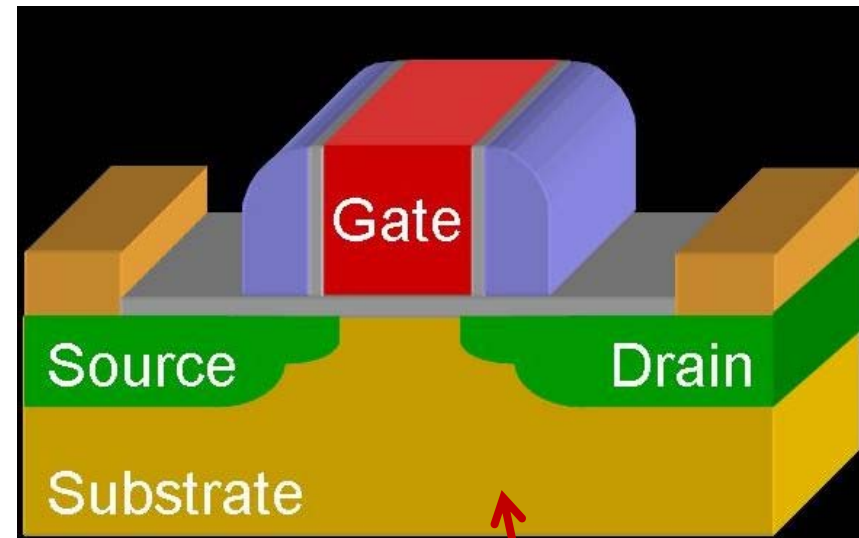
Source: UMC

- Technology options  
+  
EDA tool support  
⇒  
Different low-power techniques supported.

# The MOSFET Body Terminal

- The MOSFET actually has four terminals, not three (gate, drain and source).
- The fourth is called body.
- In  $I_{sub} \propto e^{V_T}$ ,  $V_T$  depends on MOSFET terminal voltages:

$$\Delta V_T = \frac{\sqrt{2q\epsilon N_a}}{C_{ox}} (\sqrt{2\phi_F + V_{SB}} - \sqrt{2\phi_F})$$



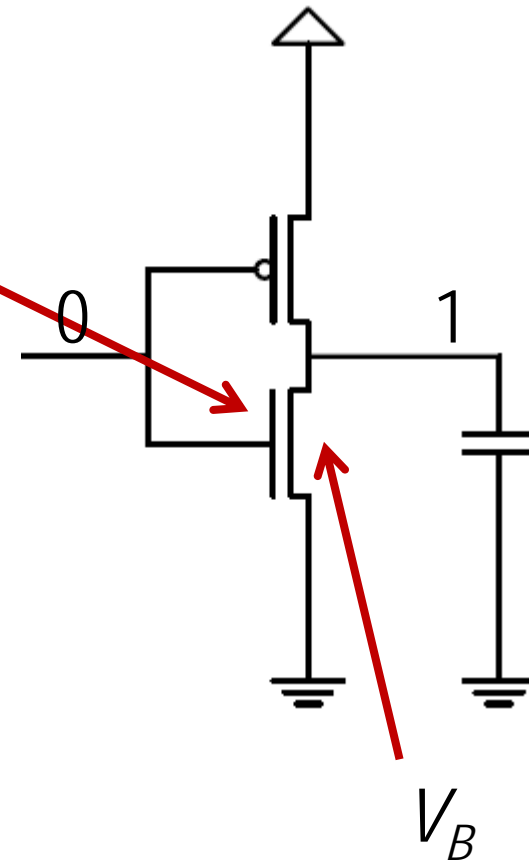
Source: USC

Body electrode



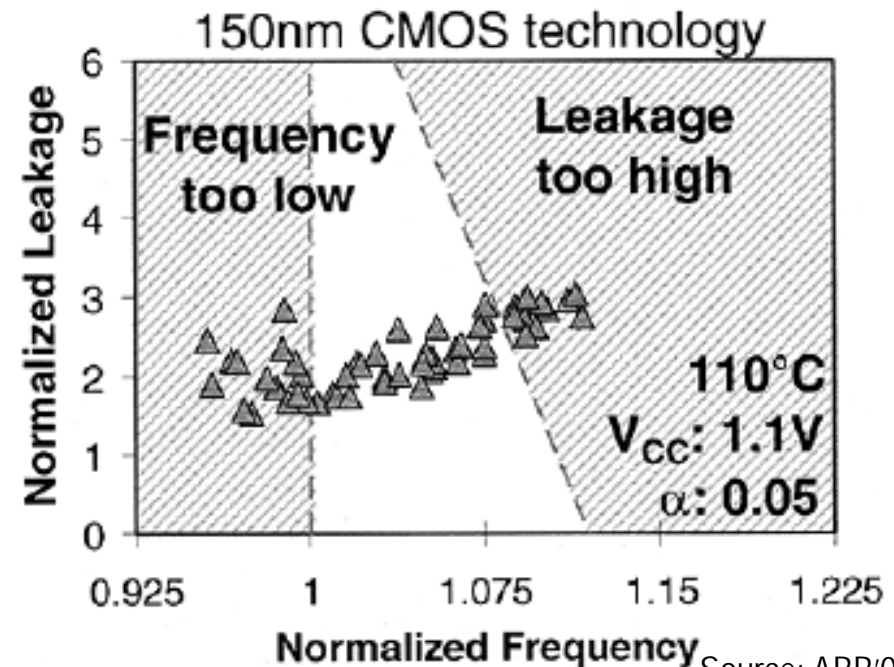
# Low-Power Technique 1: Body Biasing

- In an NMOSFET,  $V_T$  decreases with an increasing body voltage ( $V_B$ ).
  - For FinFETs from previous lecture, back biasing does not work well.
  - For Fully Depleted-Silicon on Insulator (FD-SOI), thin BOX (buried oxide) allows for back bias control.



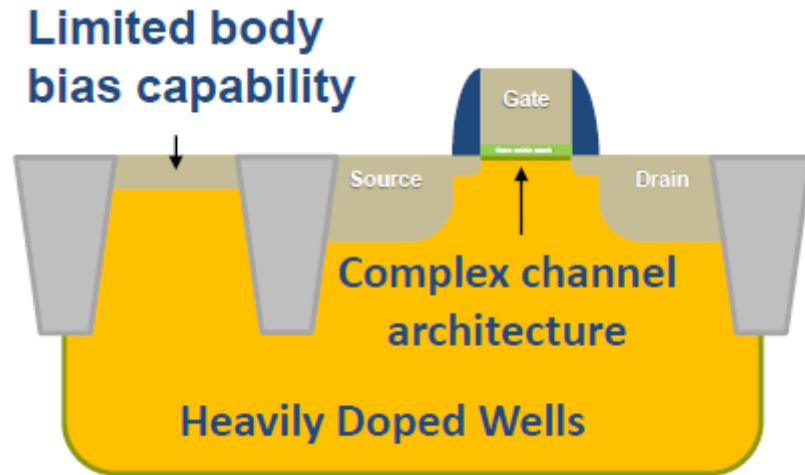
# Reverse and Forward Body Biasing

- Reverse body biasing (RBB):  
 $V_B < 0$  V (NMOSFET)  
 $\Rightarrow V_T$  increases.  
 $\Rightarrow$  leakage decreases.
- Forward body biasing (FBB):  
 $V_B > 0$  V (NMOSFET)  
 $\Rightarrow V_T$  decreases.  
 $\Rightarrow$  higher speed.

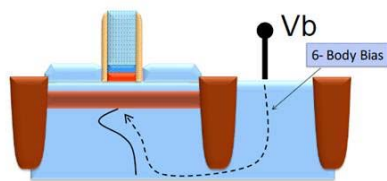


- Remember variations?
- Body bias allows for tuning at fab: Performance and power binning...

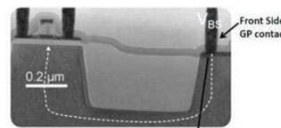
# Fully Depleted-Silicon on Insulator (FD-SOI)



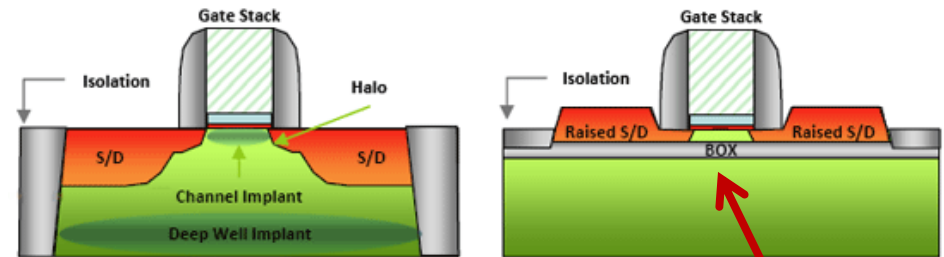
FD-SOI: 1-V  $V_B$  change  $\Rightarrow$   
85-mV  $V_T$  change



Back-gate contact



Source: STM



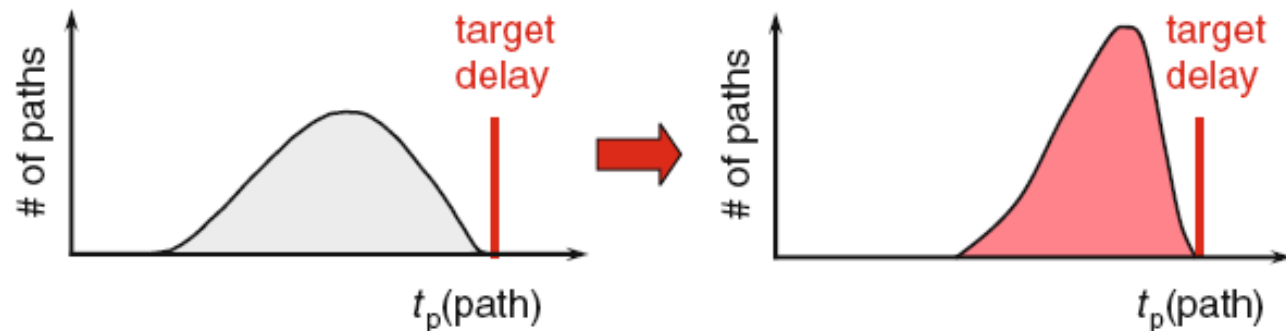
Bulk Device

FD-SOI Device

The fully depleted SOI transistor at 20 nm is significantly simpler than even a simplified version of the bulk CMOS transistor.

- FD-SOI and FinFET; BOX  
the main alternatives for scaled CMOS:
  - FD-SOI mainly for low power; good body bias control.
  - FinFET mainly for high speed; limited leakage control.

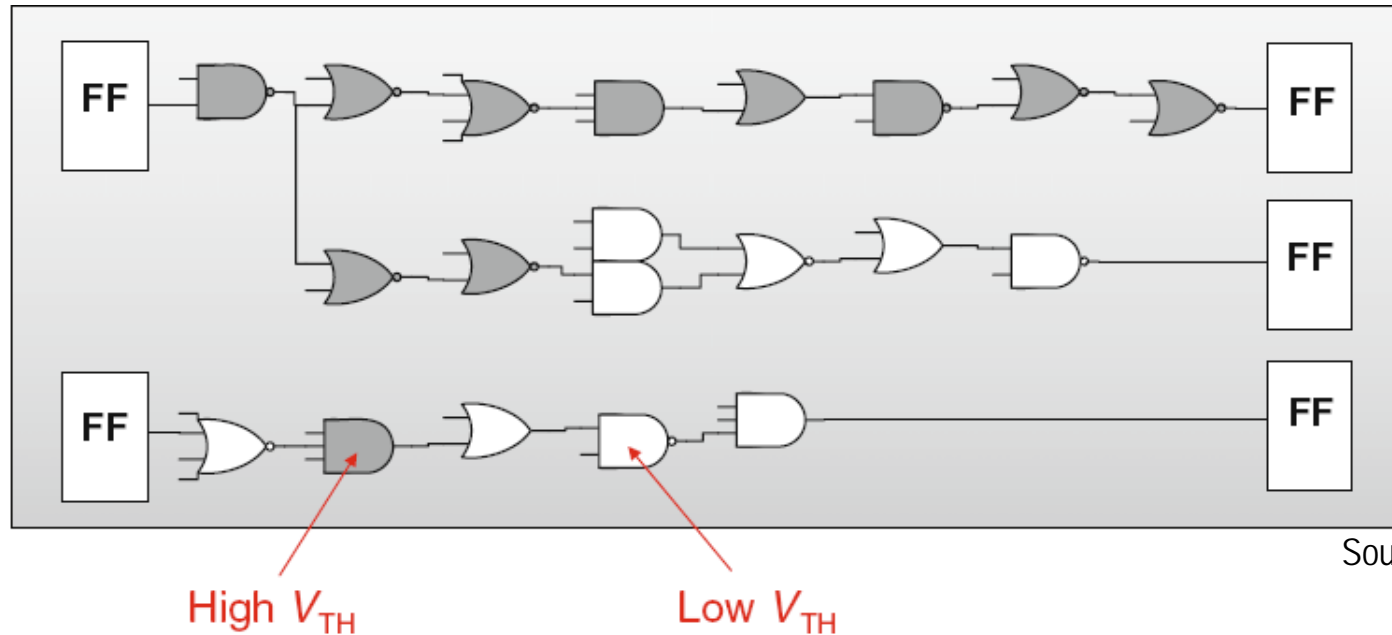
# Delay Distribution of Logic



Source: LPDE'09/Ch 4

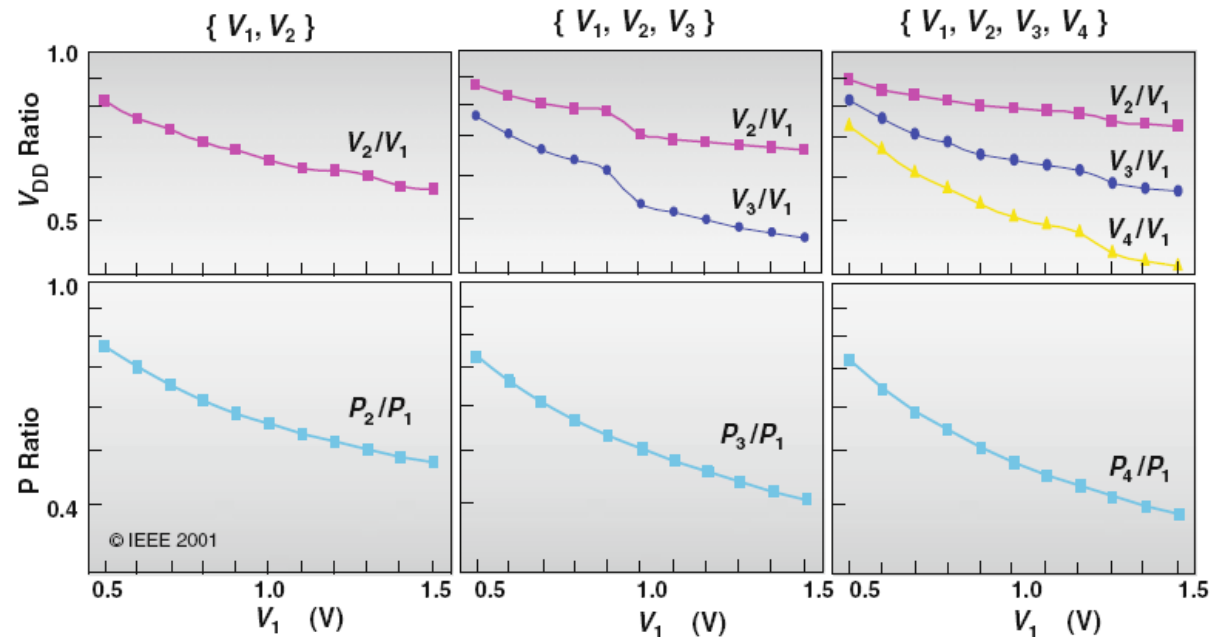
- Logic paths exhibit different delays.
- Critical paths must satisfy clock rate constraint  $\Rightarrow$  implementation must ensure gates are fast enough.
- But what about the fast paths ... can their intrinsic speed be converted to power reductions?

# Low-Power Technique 2: Multi- $V_T$



- Assign slow transistors to fast paths (makes these slower)  $\Rightarrow$  use transistors with high  $V_T \Rightarrow P_{leak}$  is reduced (but  $P_{sw}$  more or less unchanged).

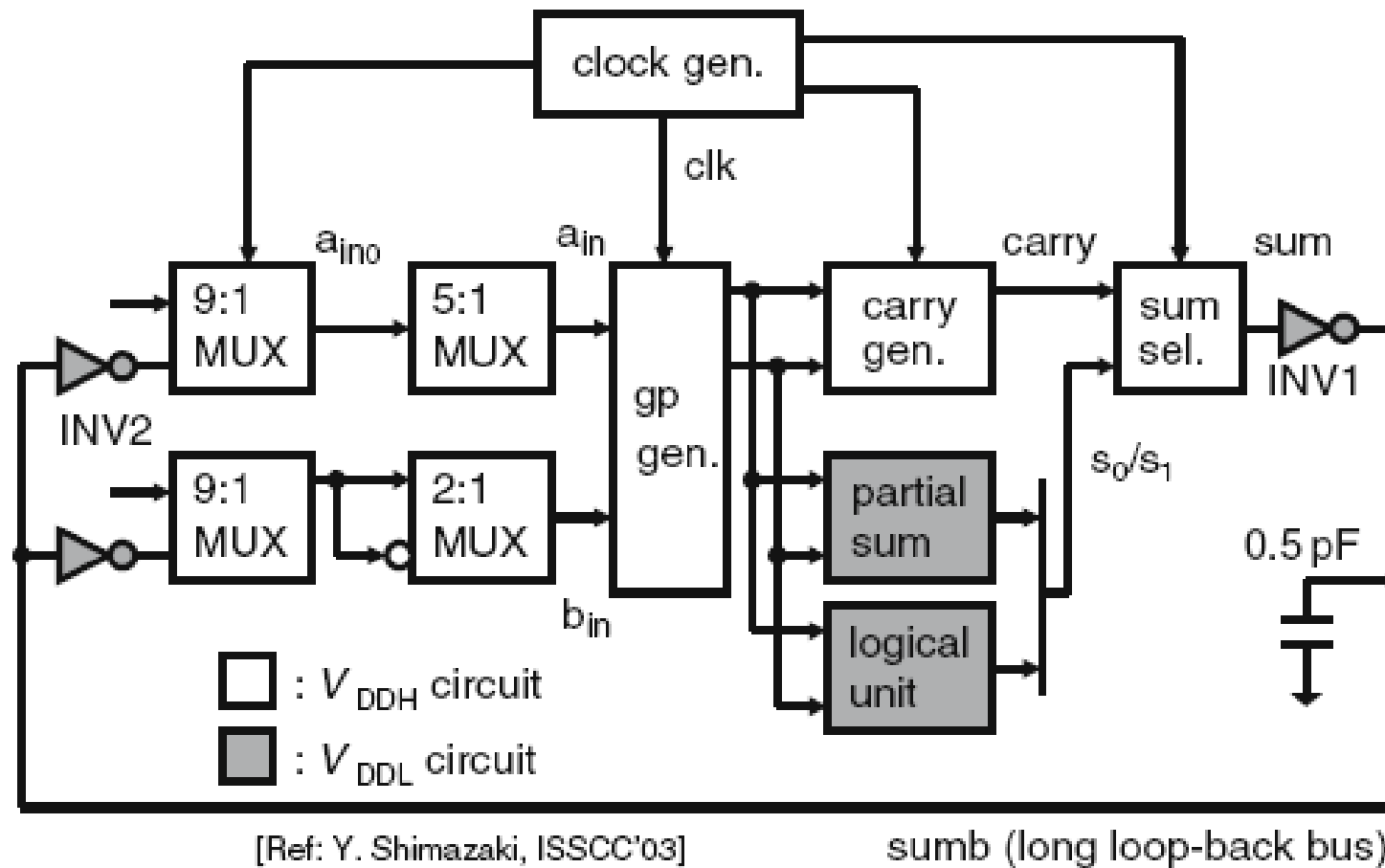
# Match $V_{DD}$ to Performance Need



Source: LPDE'09/Ch 4

- First order delay  $\propto 1/V_{DD}$  and  $P_{sw} \propto V_{DD}^2$ :
  - Reduce  $V_{DD}$  for circuits that are not timing critical  $\Rightarrow$  both  $P_{sw}$  and  $P_{leak}$  are reduced.
  - Optimal number of  $V_{DD}$  levels? Consider infrastructure overheads like voltage generation.

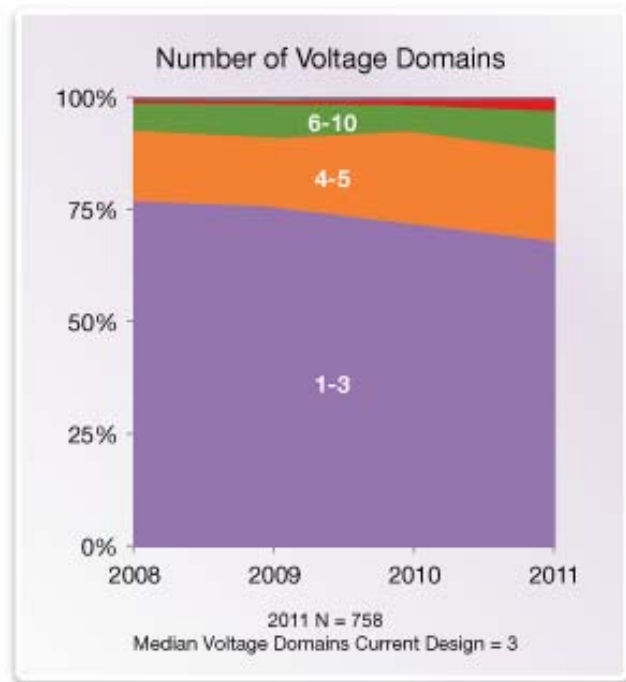
# Low-Power Technique 3: Multi- $V_{DD}$



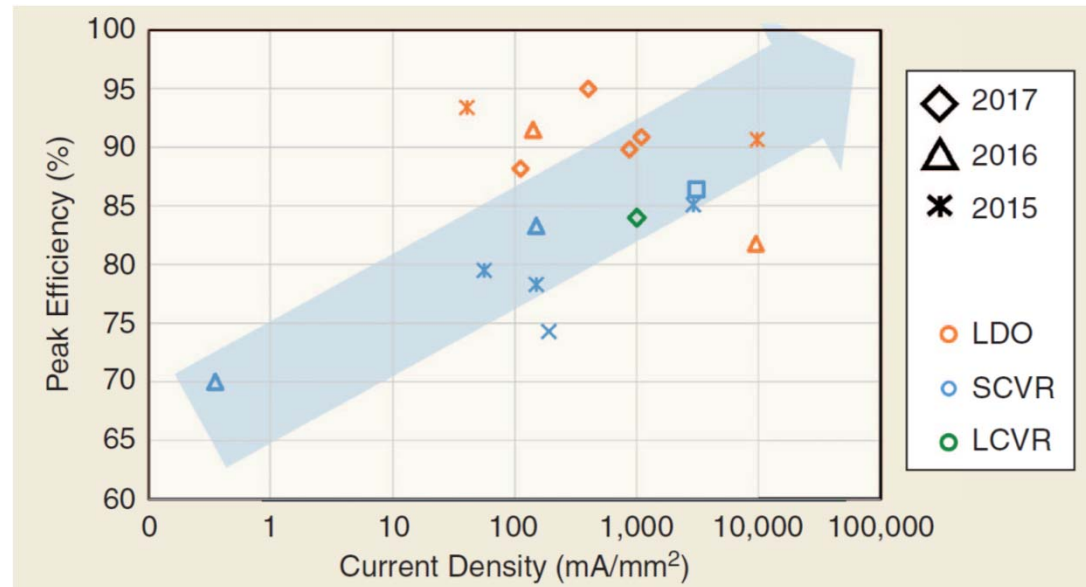
Dual-  $V_{DD}$  ALU example from LPDE'09/Ch 4

# System-on-Chips (SoCs) with Several $V_{DD}$ s

Steadily improving on-chip voltage converters.



Source: Synopsys



Source: IEEE SOLID-STATE CIRCUITS MAGAZINE



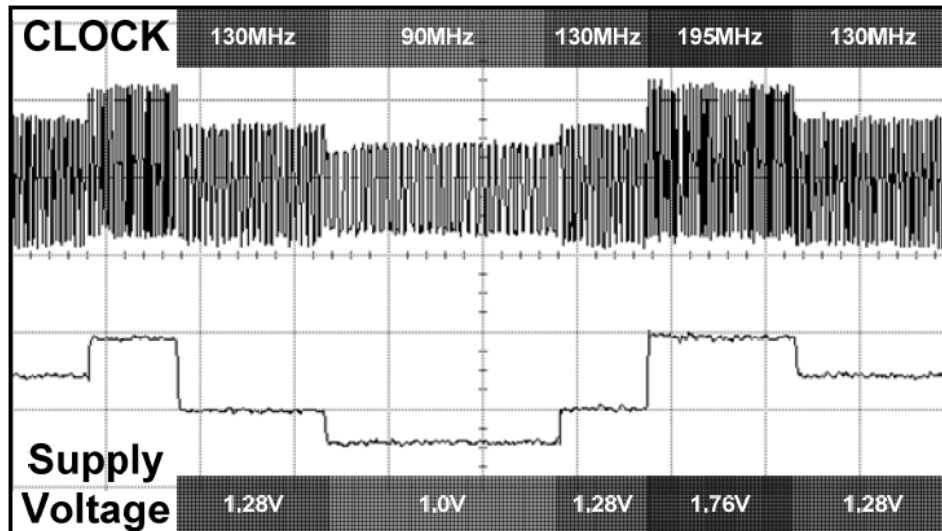
# SoC Implementation Challenges

- Multiple  $V_{DD}$ s
  - makes implementation and verification more complex.
  - requires different voltages; how are they generated?
  - requires voltage-level conversions between domains, which creates an overhead.
- What about synchronization and timing between blocks?
  - Domains may have different clock rates.
  - Domains may have varying clock rates.

# Planning Supply Voltages

- Overall, use minimal  $V_{DD}$  to limit power dissipation.
  - High performance  $\Rightarrow$  low  $V_T$ .
  - Low standby power  $\Rightarrow$  high  $V_T$ .
- To simplify integration, logic and memory should operate under the same  $V_{DD}$ . However, logic and memory have very different  $V_{DD}$  -  $V_T$  tradeoffs:
  - Unused SRAM portions are leaking  $\Rightarrow$  ought to use high  $V_T$  + low  $V_{DD}$  to reduce standby power (otherwise very significant in huge memories).
  - However, due to read disturb and write failures, high  $V_T$  + low  $V_{DD}$  spells big problems for SRAM cells.

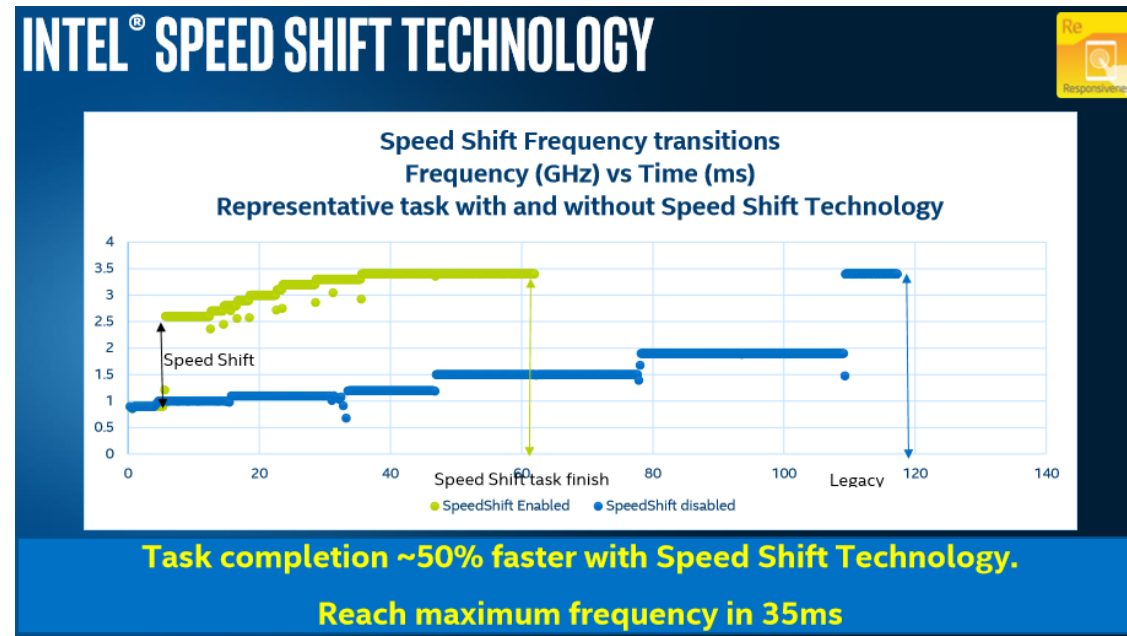
# Low-Power Technique 4: DVFS



- Timing slack can be used for power reductions:  
Dynamic Voltage and Frequency Scaling (DVFS).

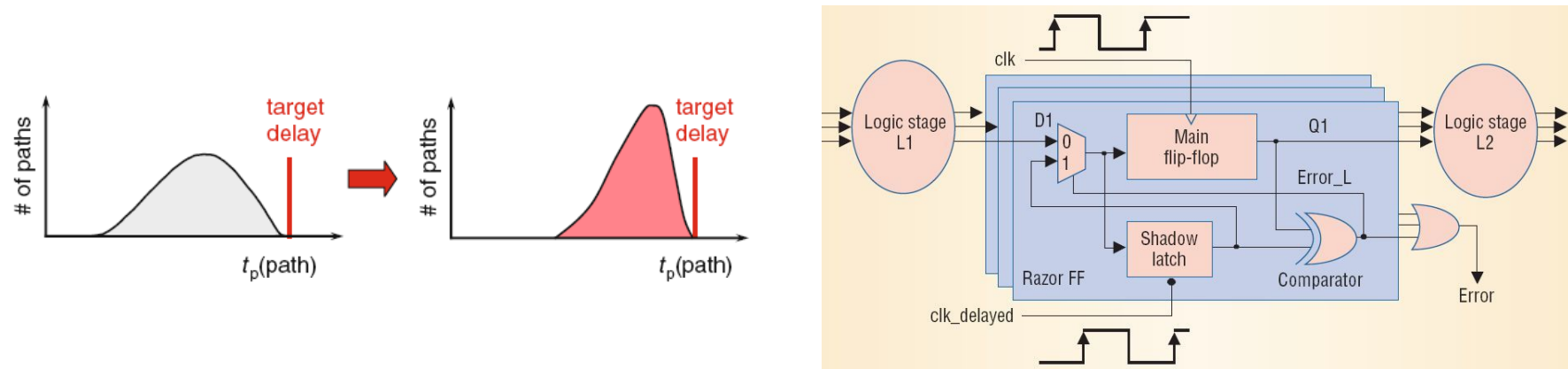
Read more in CATPE'08/Ch 3

# Faster DVFS State Transitions



- Conventional DVFS: Predefined performance states, controlled by OS  $\Rightarrow$  slow transitions.
- Intel's Speed Shift succeeded Speed Step in 2015: CPU handles transitions (faster); OS to relinquish (some) control.
  - Main gain is performance, not so much power dissipation.

# Circuit Adaptation for DVFS

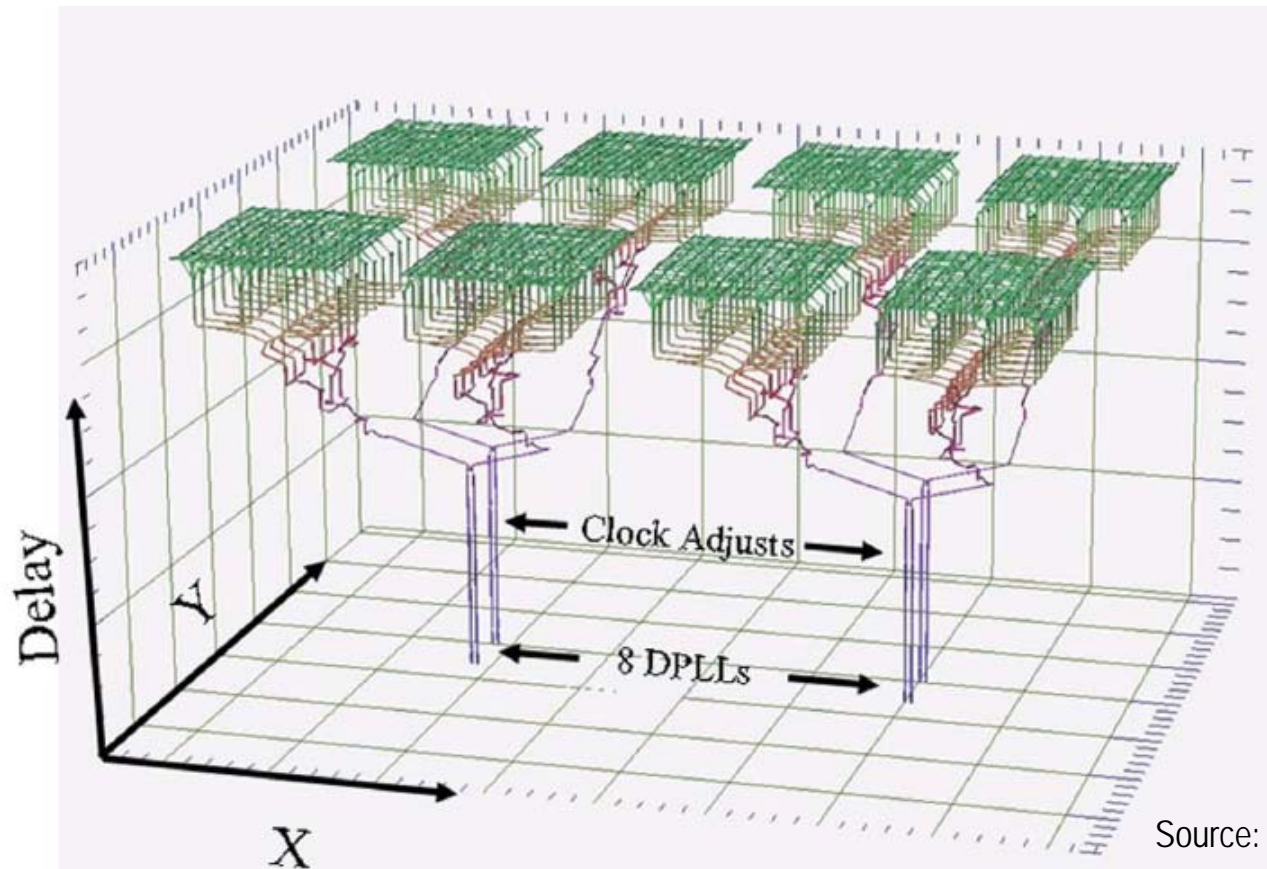


- Aggressive  $V_{DD}$  reduction would cause timing violations and, thus, computation errors.
- Solution: Implement a feedback system that regulates speed, in the process also handling variations.

Read more in CATPE'08/Ch 3.5

# Example on Variations

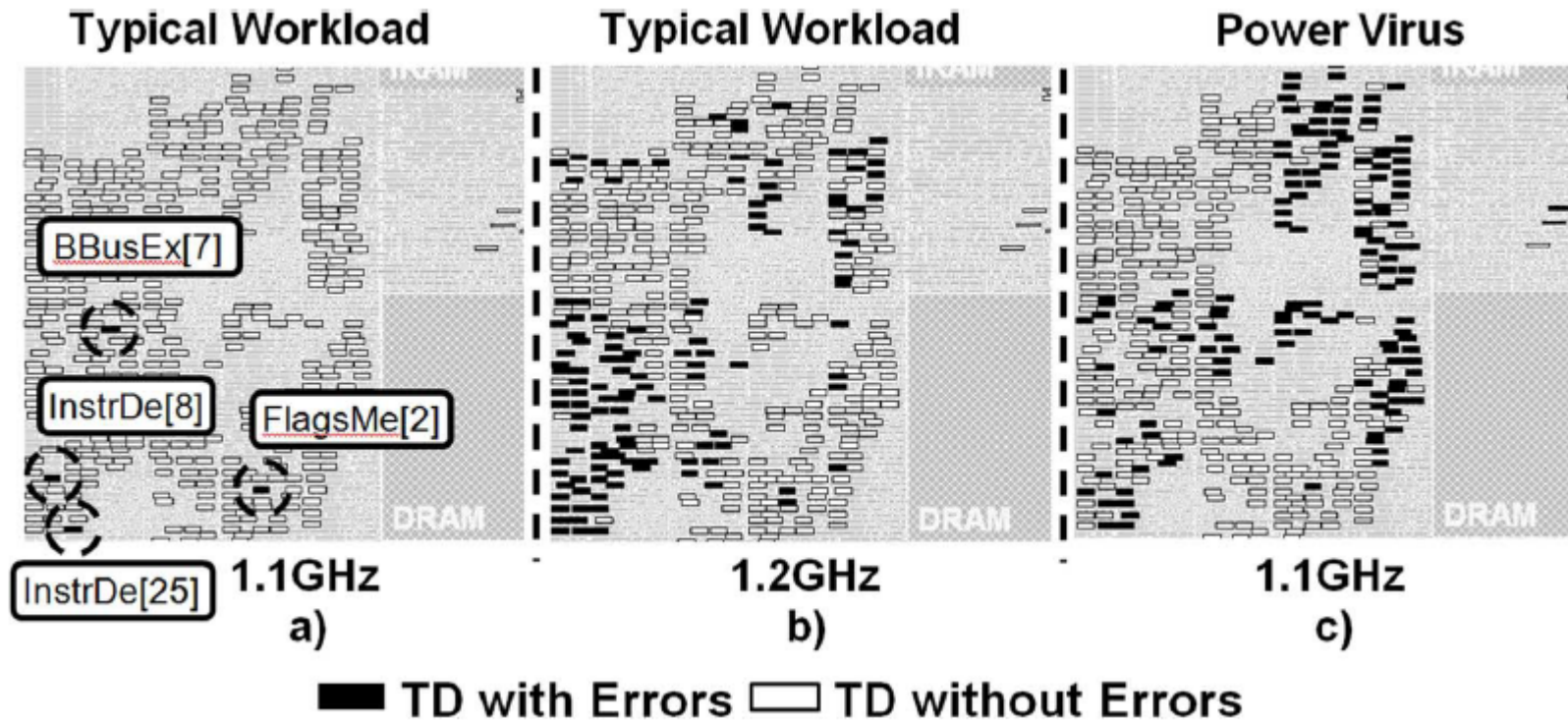
Clock arrival times are hard to synchronize.  
Such static variations are handled as a side effect.



Source: POWER'11

# Detection of Timing Failures

- High clock rates or extremely compute-intensive code can expose timing issues.



Source: ARM'11

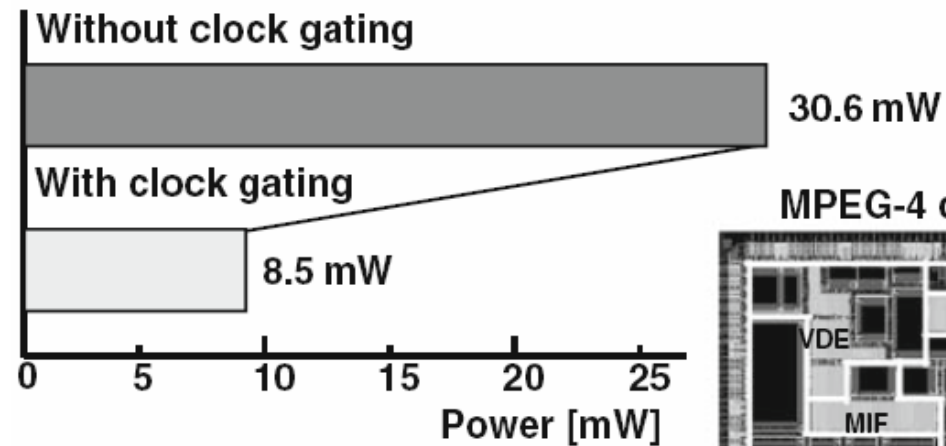
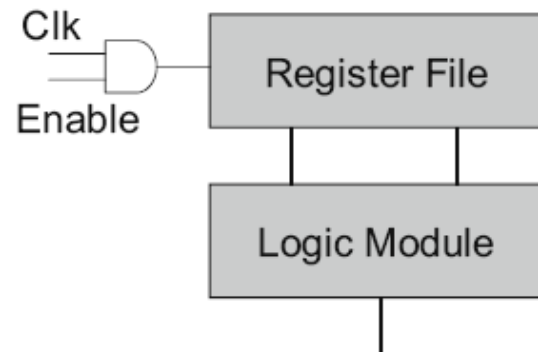


# Clock Tree Design



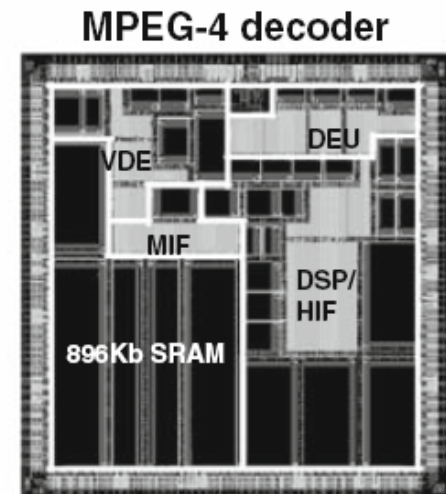


# Low-Power Technique 5: Clock Gating



90% of FFs clock-gated.

70% power reduction by clock gating alone.



[Ref: M. Ohashi, ISSCC'02]

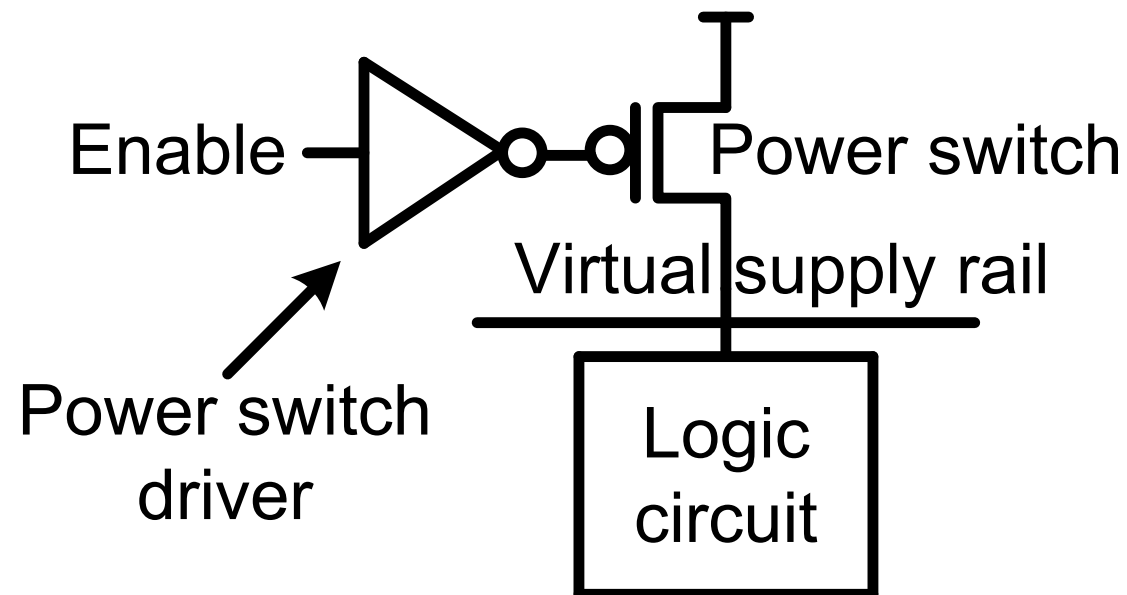
© IEEE 2002

Source: LPDE'09/Ch 8

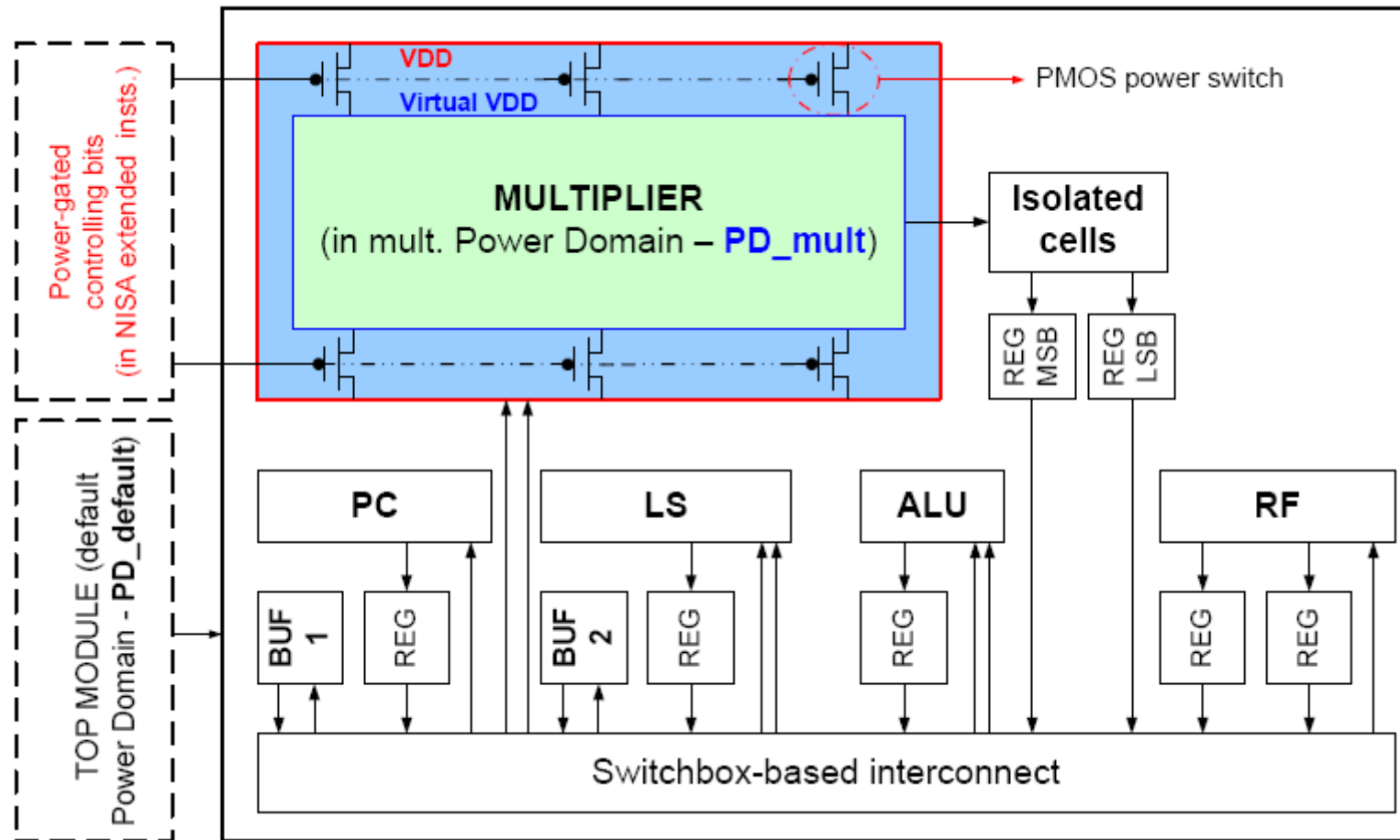
# Recap: Low-Power Techniques

- Body biasing ✓
- Multi-  $V_T$  ✓
- Multi-  $V_{DD}$  ✓
- DVFS ✓
- Clock gating ✓
- Power gating

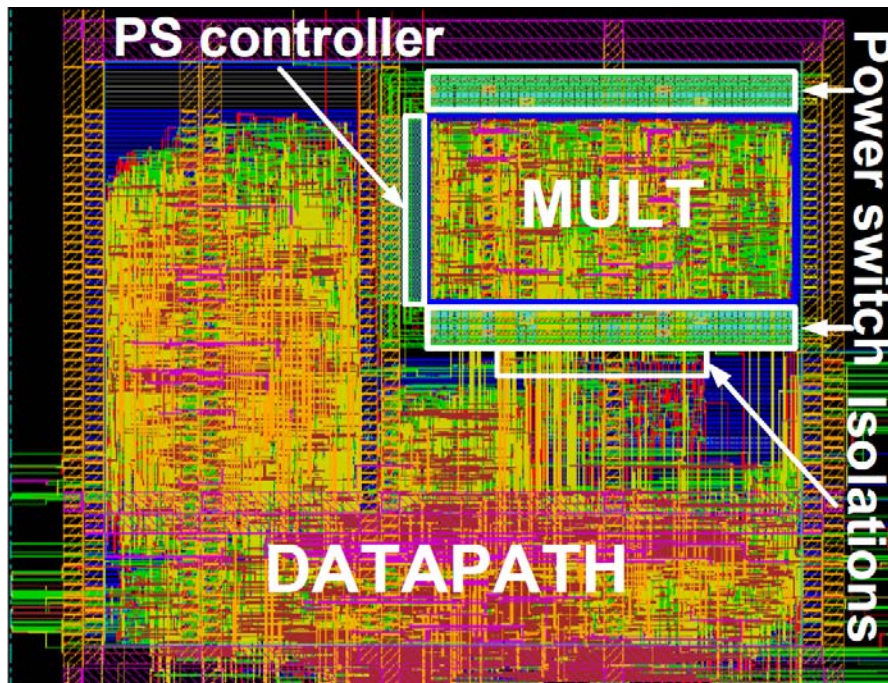
# Low-Power Technique 6: Power Gating



# Power Gating of Execution Unit

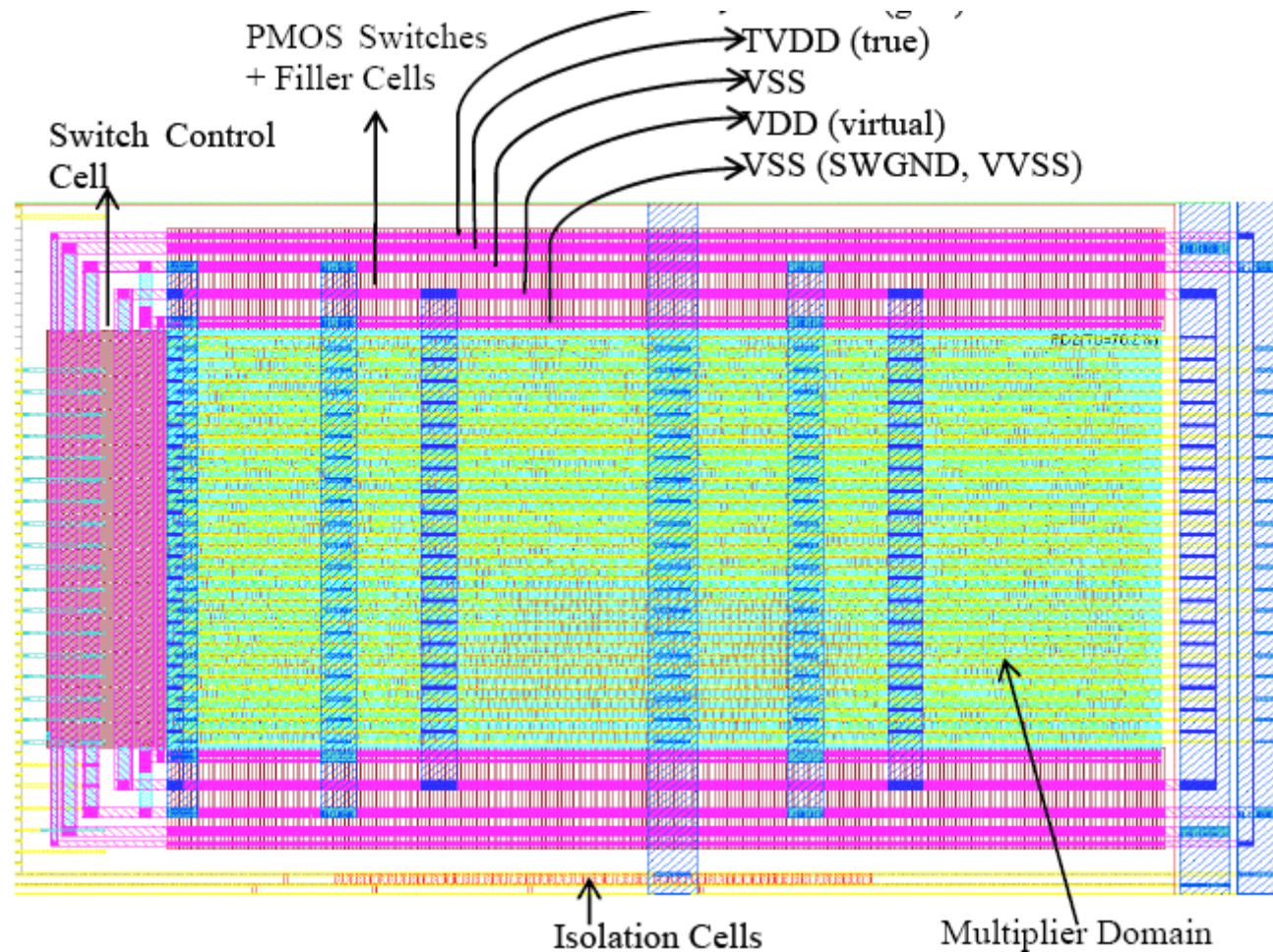


# Power Gating after Place&Route



- Need IC cell library support to implement power gating:
  - Level shifters, isolation level shifters, isolation cells, always-on buffers and inverters, retention cells, and power switches.
- EDA tool must support transition control:
  - Trade off between rush current and wake-up time.

# Power Gating Impacts Area Significantly





# Identify Multiply Activity

```

1319 00000000AE4240000000340101880 - 000057 [PCGetPC,PCJumpSA 13,RegRead2 ...
1320 000000F04000000000000039003E0 - 000058 [LSWrite LSW 4 Alu Rslt ...
1321 000000000000000000000000029000 - 000013 [RegRead1 R5,RegRead2 R4]
1322 0EC000000000800000002000F8000 - 000014 [MultRegWrite,Mult Regbank_Out2 ...
1323 000C00040000C0000000005800000 - 000015 [PCJumpDA Regbank_Out1,LSRead LSW_4
1324 00000000AA0000000000003900040 - 000016 [LSWrite LSW 4 Ls Read Mult LSW ...
1325 000000000000000000000080004000 - 000059 [PCImm (Just 2),RegRead2 R16]
1326 0000000000E4000000000080088001 - 000060 [PCImm (Just 2),RegRead1 R17,ALU0pc
      ⋮                ⋮                ⋮                ⋮
1337 00000000AE4240000000340101880 - 000057 [PCGetPC,PCJumpSA 13,RegRead2 R6 ...
1338 000000F04000000000000039003E0 - 000058 [LSWrite LSW 4 Alu Rslt ...
1339 000000000000000000000000029000 - 000013 [RegRead1 R5,RegRead2 R4]
1340 0EC000000000800000002000F8000 - 000014 [MultRegWrite,Mult Regbank_Out2 ...
1341 000C00040000C0000000005800000 - 000015 [PCJumpDA Regbank_Out1,LSRead LSW_4
1342 00000000AA0000000000003900040 - 000016 [LSWrite LSW 4 Ls Read Mult LSW ...
1343 000000000000000000000080004000 - 000059 [PCImm (Just 2),RegRead2 R16]
1344 0000000000E4000000000080088001 - 000060 [PCImm (Just 2),RegRead1 R17,ALU0pc

```

# Limited Mult Utilization Allows for Savings

Benchmark: EEMBC Autocorrelation



- Limited multiply activity, early in application  $\Rightarrow$  multiplier can be power gated since it becomes idle.
- Trade off: Static power reductions during idle vs power overhead for power gating.

Benchmark: EEMBC FFT



More extensive multiply activity.

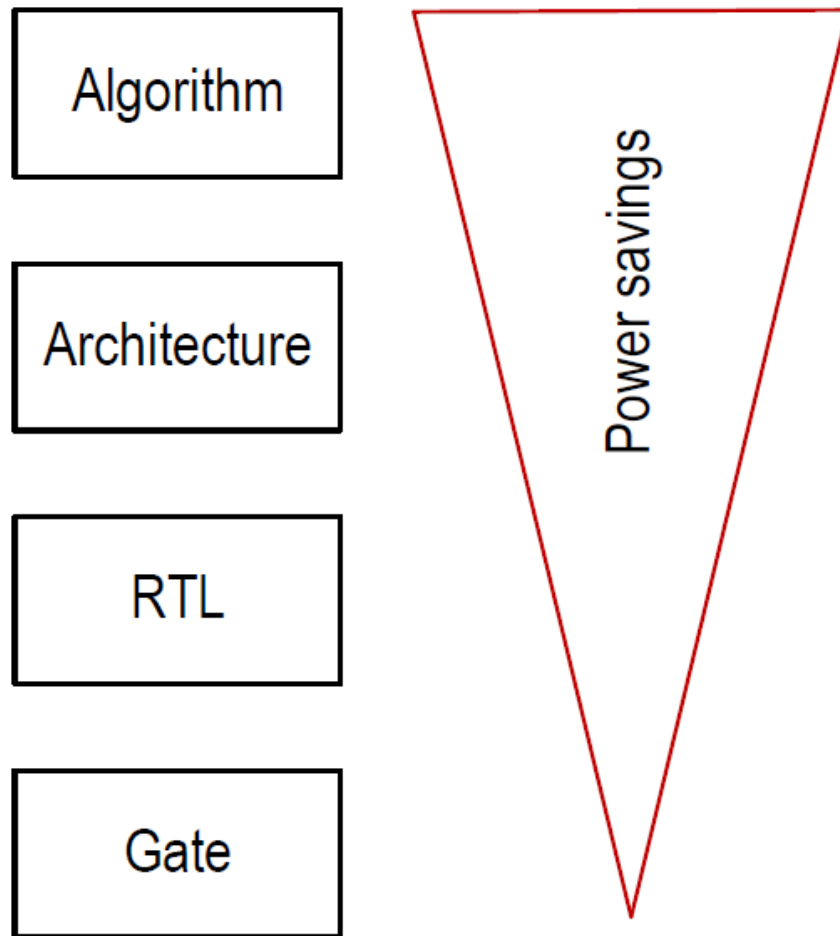


# Low-Power Options – Pros and Cons

Power-reduction Technique	Power Benefit	Timing Penalty	Area Penalty	Methodology Impact			
				Architecture	Design	Verification	Implementation
Multi-Vt Optimization	Medium	Little	Little	Low	Low	None	Low
Clock Gating	Medium	Little	Little	Low	Low	None	Low
Multi-supply Voltage	Large	Some	Little	High	Medium	Low	Medium
Power Shut-off	HUGE	Some	Some	High	High	High	High
Dynamic and Adaptive Voltage Frequency Scaling	Large	Some	Some	High	High	High	High
Substrate Biasing	Large	Some	Some	Medium	None	None	High

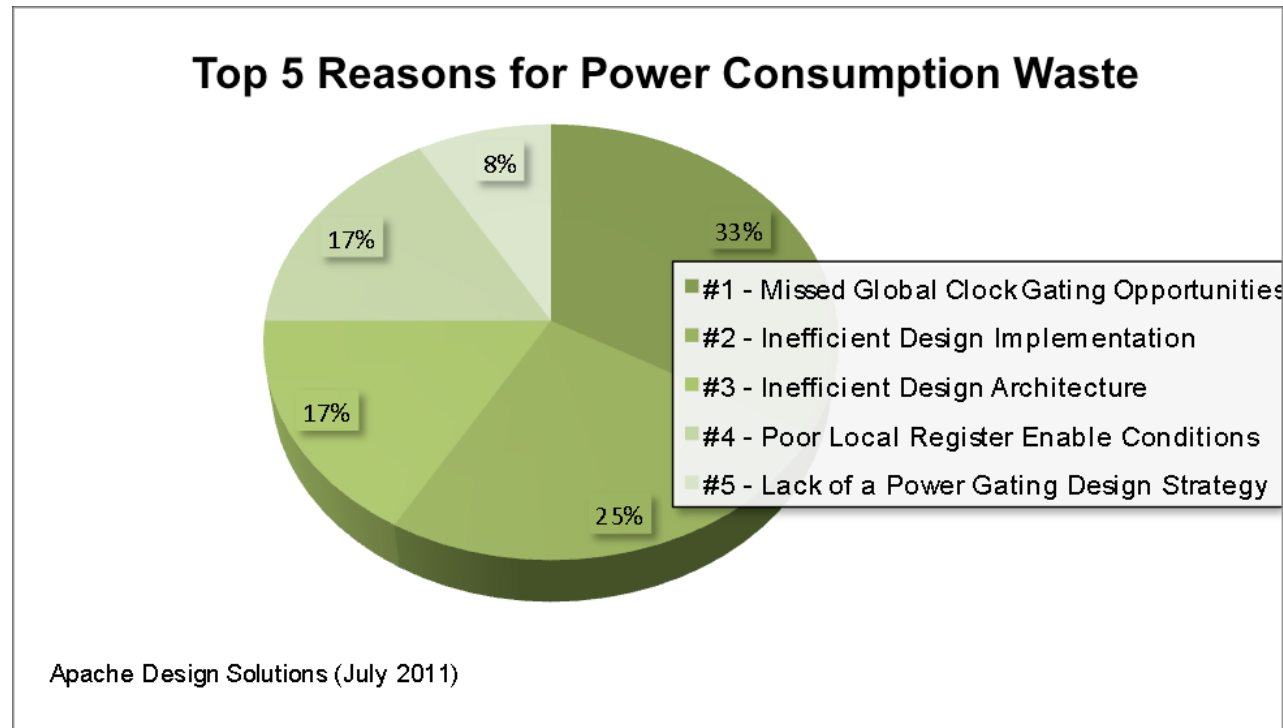
Source: Cadence

# Power Reductions in Design Flow



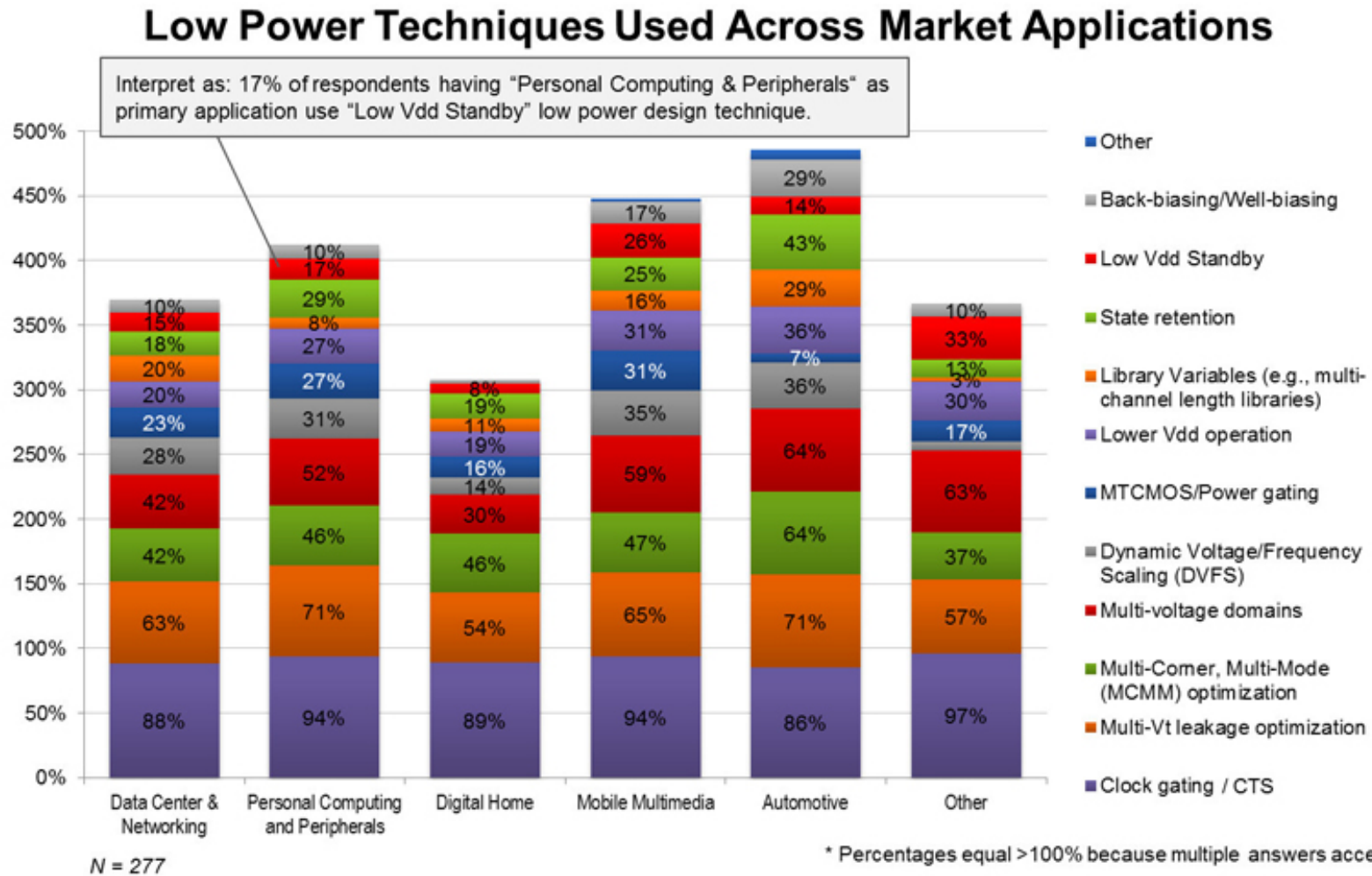
- Early design decisions yield higher power reductions than late decisions.
  - Decisions based on holistic view even better.
  - Co-optimization across levels is complex; depends on EDA tool support.
- System architects should be aware of what low-power techniques exist.

# Missed Opportunities?



- Low-power techniques clearly exist. But how do we make use of them in complex systems?
- Designer's competence + IP/cell infrastructure + EDA tools.

# Industry View on Low-Power Techniques



Source: Synopsys, Inc. Global User Survey, 2011

# Conclusion

- Reducing circuit power dissipation is done by using best design practices and by employing a few well known low-power techniques.
- Support from EDA (CAD) tools and IC technology (cell libraries) is essential to handle low-power design in an efficient manner.
- Common to all techniques is that reducing  $V_{DD}$  is effective for reducing power.

# References

- **ABB'02:** "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage", J. Tschanz et al., IEEE JSSC, Nov. 2002.
- **ARM'11:** "A Power-Efficient 32 bit ARM Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT Variation", D. Bull et al., IEEE JSSC, 2011.
- **CATPE'08:** "Computer Architecture Techniques for Power-Efficiency", S. Kaxiras and M. Martonosi, Morgan & Claypool, 2008.
- **LPDE'09:** "Low Power Design Essentials", J. Rabaey, Springer, 2009.
- **POWER'11:** "POWER7™, a Highly Parallel, Scalable Multi-Core High End Server Processor", D. F. Wendel et al., IEEE JSSC, 2011.
- And many local Chalmers papers.