EE 222 Lecture 11, Feb. 20, 2018

Low-Power Design Overview

	Constant Throughput/Latency		Variable Throughput/Latency
Energy	Design Time	Non-active Modules	Run Time
Active	Logic Design Reduced V_{dd} Sizing Multi-V_{dd}	Clock Gating	DFS, DVS (Dynamic Freq, Voltage Scaling)
Leakage	+ Multi-V_T	Sleep Transistors Multi-V_{dd} Variable V_T	+ Variable V_T

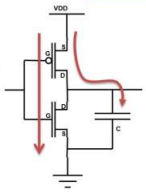
Voltage Scaling

$$E = \int \frac{C_L V_{dd}^2}{2} \beta f_{clk} + V_{dd} I_{leak} dt$$

Total Energy Consumption

$\int V_{dd} I_{leak} dt$ **Static Energy Consumption** | **Dynamic Energy Consumption** $\int \frac{C_L V_{dd}^2}{2} \beta f_{clk} dt$

Minimize leakage energy by:
- **Reducing voltage**
- **Reducing Vdd to GND paths**



Minimize active energy by:
- **Reducing voltage**
- **Switching activity**
- **Capacitance**

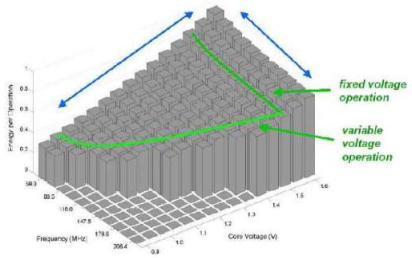
$E = \int \frac{C_L V_{dd}^2}{2} \beta f_{clk} + V_{dd} I_{leak} dt$

Total Energy Consumption

$\int V_{dd} I_{leak} dt$ **Static Energy Consumption** | **Dynamic Energy Consumption** $\int \frac{C_L V_{dd}^2}{2} \beta f_{clk} dt$

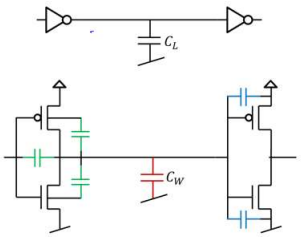
- **Reducing supply voltage below nominal**
  - **Most popular and most effective low-power strategy**
  - **Voltage-scaling**
    - **Reduces active power**
    - **Reduces leakage power (but not necessarily energy/Op)**
    - **Reduces speed : need to compensate with architectural changes (e.g., parallel processing)**

Example: StrongARM SA-1100 processor



Energy per operation as a function of voltage and clock rate

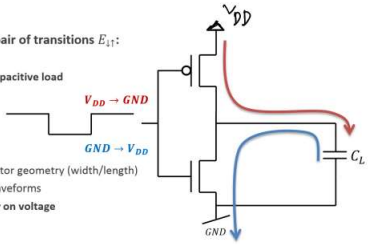
CMOS Gates With Capacitive Load



Wider transistors increase the gain factor (drive) but also increase the load (capacitance)

- **Various capacitances are merged into a single load capacitor C_L**
  - **Intrainsic MOS transistor capacitors (driver)**
  - **Extrinsig (fanout) MOS transistor capacitances**
  - **Interconnect capacitance**

- **Energy consumed during one pair of transitions E_{1T}:**
  - **Cross-over currents**
  - **Charge pumped onto the capacitive load (dominant):**



- $E_{1T} = (C_L V_{dd}) V_{dd} = C_L V_{dd}^2$
  - independent of transistor geometry (width/length)
  - Independent of the waveforms
  - **quadratic dependency on voltage**
- Energy/transition
- $E_t = C_L V_{dd}^2 / 2$
- Power consumption = Energy/transition * transition/cycle (α) * frequency (f_{clk})
- $P = \frac{\alpha}{2} C_L V_{dd}^2 f_{clk}$

Extending our calculations to a collection of nodes

- Average energy dissipated per computation cycle for one circuit node

$$E_{ch\,k} = \frac{\alpha_k}{2} E_{ch\,cyc\,k} = \frac{\alpha_k}{2} C_k U_{dd}^2$$

- Average energy dissipated per computation cycle in a voltage domain of K nodes

$$E_{ch} = \sum_{k=1}^{K} E_{ch\,k} = U_{dd}^2 \sum_{k=1}^{K} \frac{\alpha_k}{2} C_k$$

Node activity (aka switching activity)

- Fact: Not all nodes within a (sub)circuit do change state at the same rate.
- Definition:
  A node's activity α_k indicates how many times per computation cycle node k switches from one logic state to the opposite one when averaged over many computation cycles.
- Examples:
- Ungated clock in single-edge-triggered clocking: $\alpha_k = 2$
- Ungated clock in dual-edge-triggered clocking: $\alpha_k = 1$
- Output of a T-type Flip-Flop if permanently enabled: $\alpha_k = 1$
- Output of a D-type Flip-Flop fed with random data: $\alpha_k = 1/2$

Impact of Glitching

- In a synchronous (single-edge triggered) circuit, the activity factor of each node should never rise above $\alpha_k = 1/2$
- Reality: activity factors up to 6 or more can be observed:
  - Increased activity due to glitches: signals reconverge after having propagated along paths of markedly different depths
- Glitching explains why the isomorphic architecture often dissipates more (dynamic) energy than more sophisticated architectures do.
- Activity caused by glitches is very difficult to predict (depends heavily on timing)
  - Analytical prediction almost impossible

- Node activities are distributed very unevenly in most circuits.

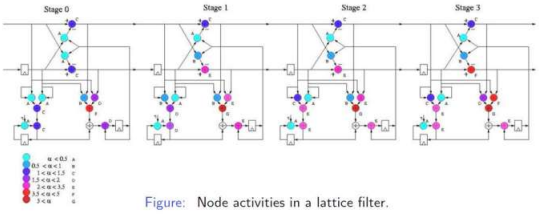
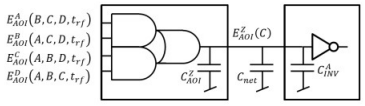


Figure: Node activities in a lattice filter.

  - Activity increases with the number of preceeding logic stages (increased glitching)

- Power consumption is divided into
  - Net switching power
  - Internal power
    - Internal power depends on actual input values
    - Power is consumed even if output does not change
- Library files: internal energy characterization for each cell at given supply voltage
  - Internal energy (cross-current, switching) per change in each input and output (as functions of input slope t_{rf} and output load C)
  - Contribution to capacitance of the connected net (input/output load)

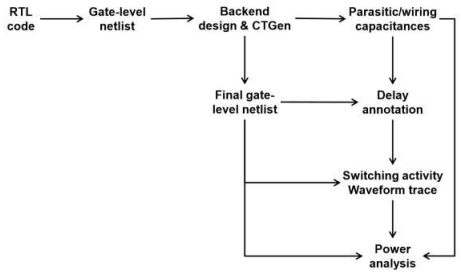


$$C = C^Z_{AOI} + C_{net} + C^A_{INV}$$

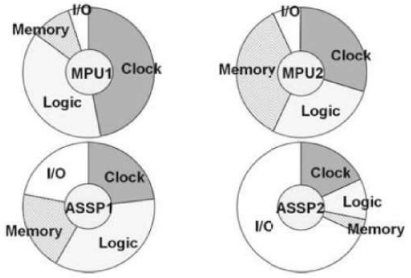
What about the activity factor(s)?

- Fixed activity:
  Assume a constant activity factor for all nodes in the circuit
  - Very rough estimate and highly inaccurate
- Statistical power analysis:
  Assumes a given toggle activity at the input and propagates the activity throughout the circuit using statistical models of the gates
  - Does not account for correlation between signal values
  - No accounting for glitching activity
- Simulation based:
  Obtains toggle statistics from gate level simulations
  - Most accurate method
  - Slow

Gate-Level Power Analysis Flow



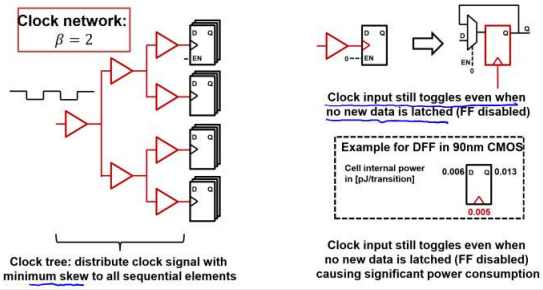
- The clock is a major source of power consumption in many synchronous designs

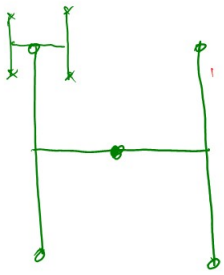


J. Rabaey: Power figures from sever microprocessors and DSPs

RTL Power Reduction: Clocking

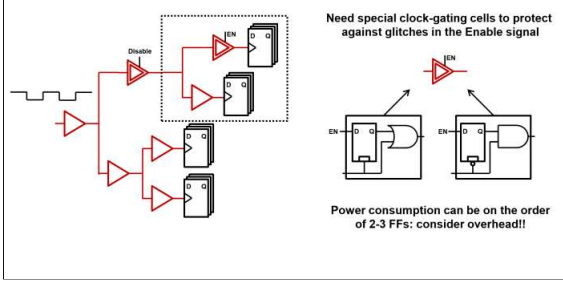
- The clock is a major source of power consumption in many synchronous designs
  - Clock distribution network (clock tree)
  - Intrinsic power of sequential elements (even when data input is constant)





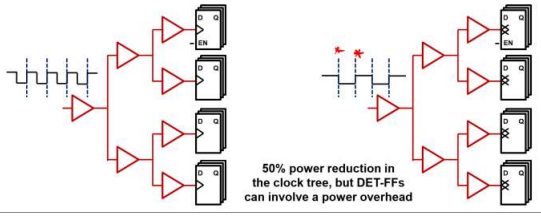
RTL Power Reduction: Clocking

- Clock gating: reduce power consumption by disabling the clock for
  - Inactive parts of the design (coarse grained)
  - Disabling FFs without consuming internal power (fine-grained)

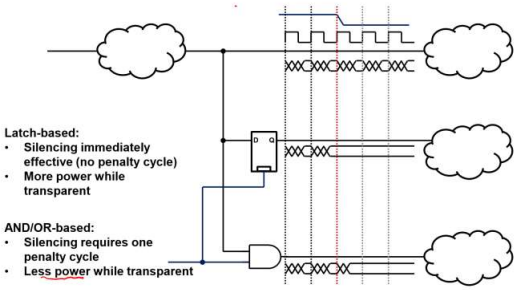


RTL Power Reduction: Clocking

- Double-data rate design
  - Clock network has the highest activity factor ($\beta = 2$)
  - Two transitions per clock period with only one transition triggering a state change
- Replace FFs with double-edge triggered FFs
  - Clock frequency can be cut in ½ for same number of operations

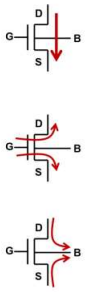


- **Silencing:** avoid activity in unused logic
  - Unused logic is not always immediately preceded by registers
  - Avoid changes to the input of unused parts of the logic



Leakage Power

- Transistors leak currents even when in off-state
- Sources for leakage
  - Sub-threshold leakage
    - Dominant component in most circuits
  - Gate tunneling
    - Generally low, even in modern technologies due to high-k gate dielectrics
    - Decreases very rapidly with decreasing V_{dd}
  - Junction current
    - Generally low
    - Decreases very rapidly with decreasing V_{dd}

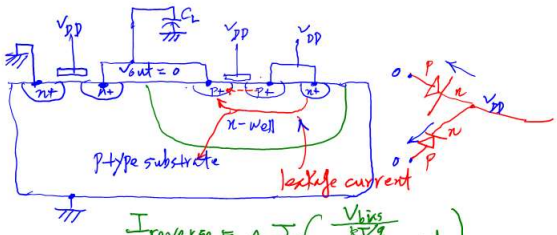


$$P_{total} = \alpha C_{load} V_{DD} \Delta V f_{clock} + V_{DD}(I_{short\,ckt} + I_{leakage} + I_{static})$$

I_{static}: dc current drawn from V_{DD} supply

Some statistics

	Intel 80386	DEC Alpha 21064	cell-based ASIC
Design rule	1.5 μm	0.75 μm	0.5 μm
No of gates	36,808	263,686	50,000
f_{clock}	16 MHz	200 MHz	110 MHz
V_{DD}	5 V	3.3 V	3 V
P_{total}	1.4 W	32 W	0.8 W
Logic gates	32%	14%	9%
clock distrib.	9%	32%	30%
Interconnect	28%	24%	15%
I/O	26%	30%	46%



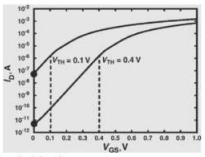
$$I_{reverse} = A J_S \left(e^{\frac{V_{bias}}{kT/q}} - 1\right)$$

J_S reverse saturation current density
A junction area

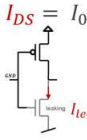
typically $I_{reverse} = 1 \sim 5\ pA/\mu m^2$ which increases significantly with temp. T

Leakage Power

- Long channel deices (>130nm): $I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}}$
  - I_{DS} mostly independent from Drain-Source Voltage
  - Leakage current depends strongly on $V_{GS} - V_{th}$
    - Decreasing threshold voltage increases leakage
- Impact of technology scaling on sub-threshold leakage (<130nm)
  - Drain-Induced Barrier Lowering (DIBL): V_{DS} modulates threshold voltage
  - I_{DS} becomes a function of V_{DS}



- $I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}+\lambda_{DS}V_{DS}}{v_t n}}$



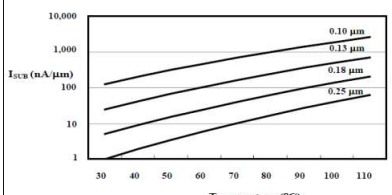
$I_{leak} = I_0 e^{\frac{-V_{th}+\lambda_{DS}V_{DD}}{v_t n}}$ ← Voltage scaling reduces leakage

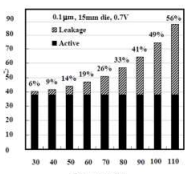
Leakage Power over Temperature

Drain current depends exponentially on thermal voltage $v_t = kT/q$

$$I_{DS} = I_0 e^{\frac{V_{GS}-V_{th}}{v_t n}}$$

- Exponential I_{DS} increase with temperature





Example: 0.7V, 100nm process, 15mm2 die

Vivek De, Intel

Leakage in Transistor Stacks

- Stacking occurs
  - In many logic gates (> 1 input)
  - When introduced intentionally for leakage reduction

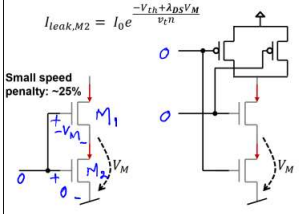
$$I_{leak,M1} = I_0 e^{\frac{-V_M - V_{th} + \lambda_{DS}(V_{dd} - V_M)}{v_t n}}$$

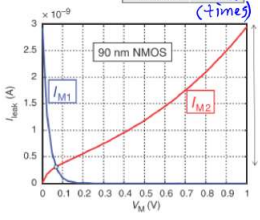
$$I_{leak,M2} = I_0 e^{\frac{-V_{th} + \lambda_{DS} V_M}{v_t n}}$$

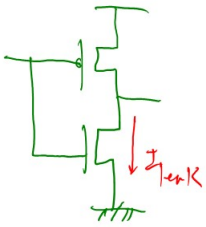
Leakage Reduction	
2 NMOS	9 X
3 NMOS	17 X
4 NMOS	24 X
2 PMOS	8 X
3 PMOS	12 X
4 PMOS	16 X

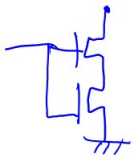
(times)

Small speed penalty: ~25%









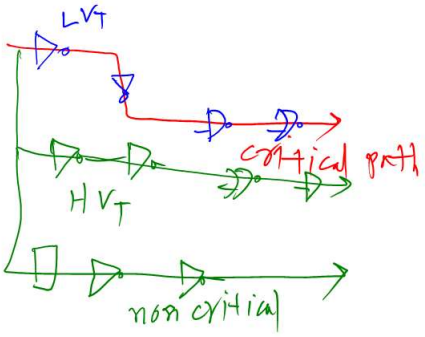
Threshold Voltage Selection

- Modern process technologies support devices with different threshold voltages
  - Typically three flavors: low-VT, standard-VT, high-VT
  - Often all three flavors can be mixed in the same design
- **VT-selection:** tradeoff between speed and leakage

$$t_{pd} = \frac{t_{OX}}{\mu \varepsilon_{OX}} \frac{L}{W} C_L \frac{V_{DD}}{(V_{DD} - V_{th})^{\alpha}} \qquad I_{leak} = I_0 e^{\frac{-V_{th} + \lambda_{DS} V_{DS}}{v_t n}}$$

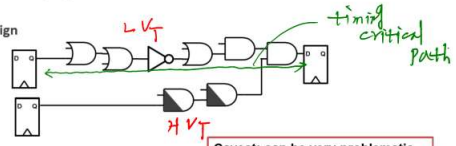
- **Example**: 55nm process

	HVT	SVT	LVT
			
Delay	20ps	16ps	14ps
Leakage	30nW	60nW	200nW



Multi-VT Design

- Design tradeoff when choosing a VT flavor:
  - Less leakage (high-VT) increases delay and vice versa
  - Threshold voltage types can often be mixed
- Multi-VT design



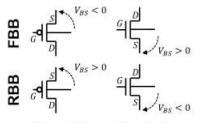
  - Use low-VT cells only on critical paths
  - High-VT cells are used in all other paths

Caveat: can be very problematic for near-VT or sub-VT design: path delays scale very differently

- Methodology:
  - Either done by replacing non-critical cells in the backend OR already during synthesis by providing multiple libraries (HVT/SVT *and* LVT)

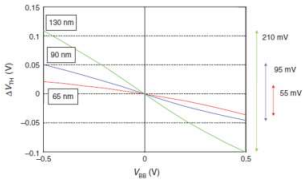
Body Bias Modulates Threshold Voltage

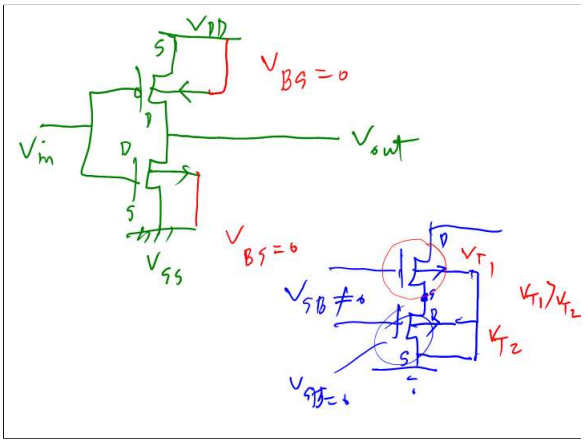
- Body of the transistor is often connected to the source (no body bias)
- Introducing a body bias modulates threshold voltage
  - Forward Body Bias (FBB): increases threshold voltage
  - Reverse Body Bias (RBB): reduces threshold voltage

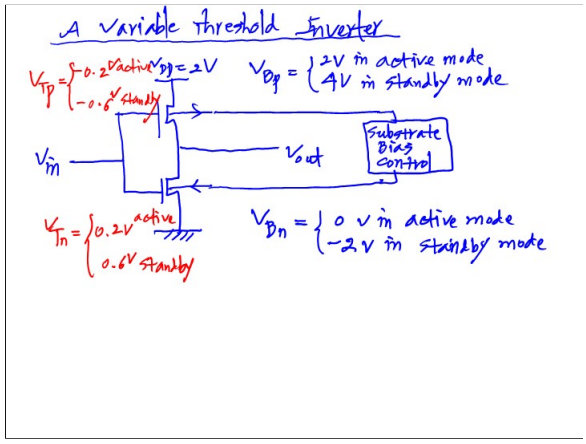


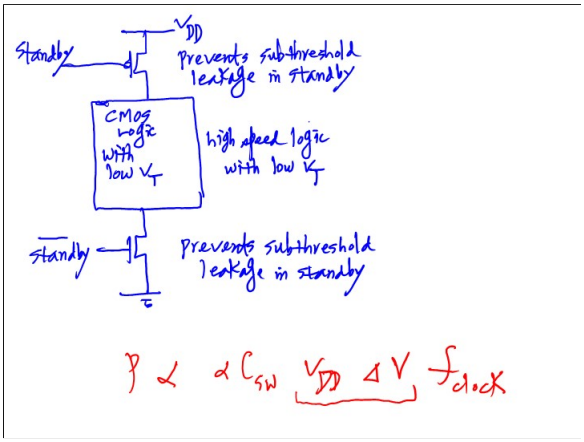
- $V_{th} = V_{th0} - \lambda_{BS} V_{BS}$ (nMOST)
- BULK CMOS:
  - Effect of body bias decreases for technologies below 100nm
  - FBB is limited to ~300mV to avoid operating junction diodes in forward direction

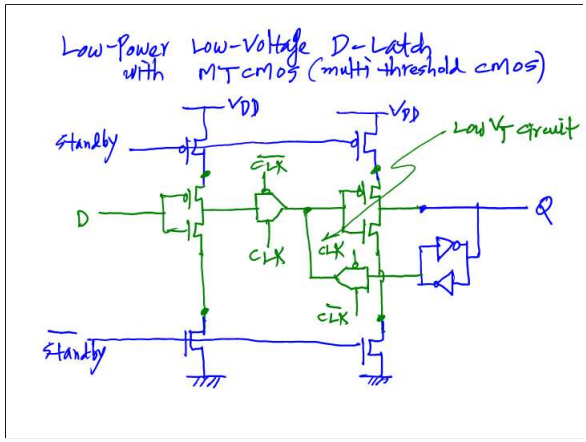
$V_{th0} < 0$ $V_{th0} > 0$

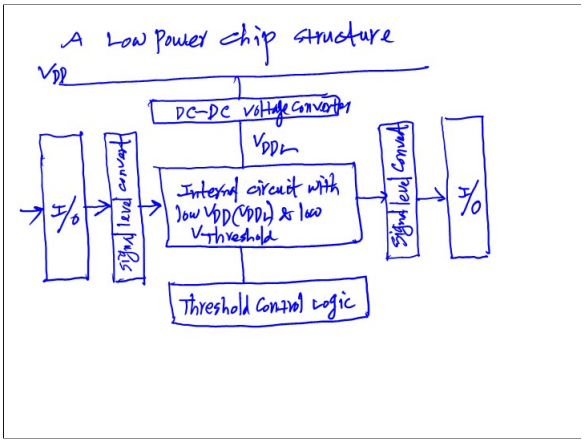






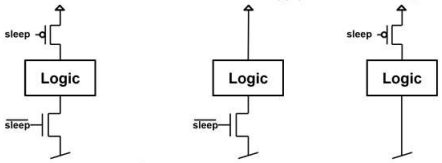






Power Gating

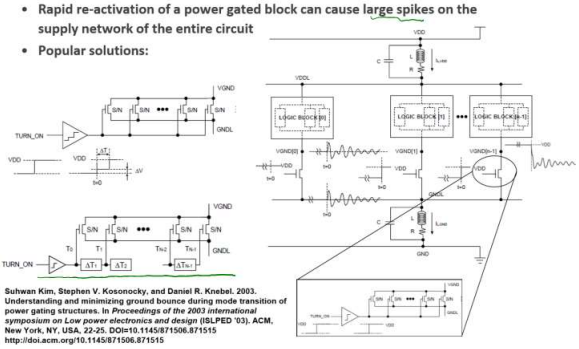
- Avoid leakage almost completely when individual design units are not used:
  - Disconnect entire modules from the supply with headers and/or footers)



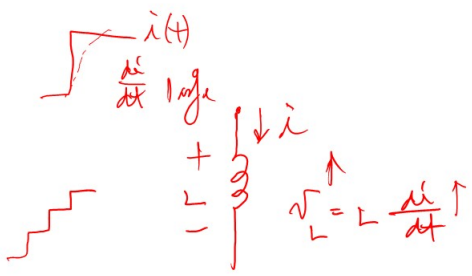
- Objectives with conflicting requirements
  - Sleep mode: large off-resistance to avoid leakage (stacking)
    - PMOS preferred over NMOS and HVT over LVT, header+footer
  - Active mode: minimize on-resistance to reduce negative impact on timing
    - Sleep transistors require large area
    - NMOS preferred over PMOS, LVT over HVT, footer-only

Power Mode Transition

- Rapid re-activation of a power gated block can cause large spikes on the supply network of the entire circuit
- Popular solutions:



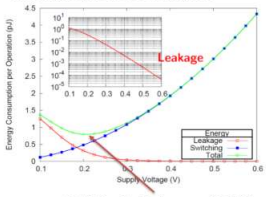
Suhwan Kim, Stephen V. Kosonocky, and Daniel R. Knebel. 2003. Understanding and minimizing ground bounce during mode transition of power gating structures. In *Proceedings of the 2003 international symposium on Low power electronics and design* (ISLPED '03). ACM, New York, NY, USA, 22-25. DOI=10.1145/871506.871515 http://doi.acm.org/10.1145/871506.871515



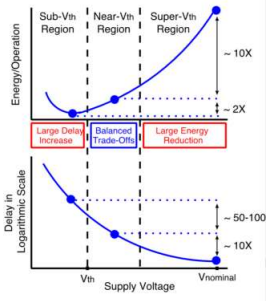
Ultra-Low-Power Design: Sub-Threshold Operation

- Near/below VT operation:
  - Exponential delay/leakage increase
  - Minimum energy voltage: balance between leakage and active power consumption

J. Rodrigues, PATMOS 2011, Keynote



Relatively flat around EMV



Given S_{spec} ($\tau < \tau_{spec}$)
Minimize Power
so long as meeting specs

Start with high V_T everywhere
But for timing critical paths,
lower V_T
Further, you can also adjust V_{DD}
& play with V_T (bias voltage programming)
multiple V_{DD}, multiple V_T, etc.