Low-Power Design Overview

<table>
<thead>
<tr>
<th></th>
<th>Constant Throughput/Latency</th>
<th>Variable Throughput/Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>Design Time</td>
<td>Non-active Modules</td>
</tr>
<tr>
<td>Active</td>
<td>Logic Design</td>
<td>Reduced $V_{dd}$</td>
</tr>
<tr>
<td></td>
<td>Sizing</td>
<td>Multi-$V_{dd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Dynamic Freq, Voltage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scaling)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leakage</td>
<td>+ Multi-$V_{T}$</td>
<td>Sleep Transistors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi-$V_{dd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Variable $V_{T}$</td>
</tr>
</tbody>
</table>

**Voltage Scaling**

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

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

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

**Example: StrongARM SA-1100 processor**

Energy per operation as a function of voltage and clock rate

**CMOS Gates With Capacitive Load**

- Various capacitances are merged into a single load capacitor $C_L$
  - Intrinsic MOS transistor capacitances (driver)
  - External (beneath) MOS transistor capacitances
  - Interconnect capacitance

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

**Energy consumed during one pair of transitions $E_C$:**
- Cross-over currents
- Charge pumped onto the capacitive load (diverted)
$$E_C = (C_L V_{dd}) I_{ddt} = C_L V_{dd}^2$$
- Independent of transistor geometry (width/length)
- Independent of the waveform
- Quadratic dependency on voltage

**Energy/transition**
$$E_t = C_L V_{dd}^2 / 2$$

**Power consumption = Energy/transition * transition/cycle (ct) * frequency ($f_{in}$)**
$$P = \frac{1}{t} E_t f_{in}$$
Extending our calculations to a collection of nodes

- Average energy dissipated per computation cycle for one circuit node
  \[ E_{ch,k} = \frac{\alpha_k}{2} \]
  \[ E_{ch,cyc,k} = \frac{\alpha_k}{2} C_k V_d^2 \]

- Average energy dissipated per computation cycle in a voltage domain of \( K \) nodes
  \[ E_{ch} = \sum_{k=1}^{K} E_{ch,k} = U_d^2 \sum_{k=1}^{K} \frac{\alpha_k}{2} C_k \]

Node activity (aka switching activity)

- Fact: Not all nodes within a (sub)circuits do change state at the same rate.
- Definition:
  A node’s activity \( \alpha_k \) 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 filters.](image)

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

Power consumption is divided into

- Not 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
- Contribution to capacitance of the connected net (input/output load)

\[ C = C_{par} + C_{ext} + C_{int} \]

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

- RTL code → Gate-level netlist → Backend design & CTGen → Parasitic/wiring capacitances → Final gate-level netlist → Delay annotation → Switching activity waveform trace → Power analysis

- 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)

- Clock tree: distribute clock signal with minimum skew to all sequential elements

- Clock input still toggles even when no new data is latched (FF disabled)

- Example for DFF in 0.18um CMOS
  - Output internal power (in [mW]):
    - Example 1: 0.75mW
    - Example 2: 0.375mW

- Clock input still toggles even when no new data is latched (FF disabled) causing significant power consumption

RTL Power Reduction: Clocking

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

- Need special clock-gating cells to protect against glitches in the Enable signal

- Power consumption can be on the order of 2-3 FFs; consider overhead!

- Double-data rate design
  - Clock network has the highest activity factor ($f = 2$)
  - 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

- 50% power reduction in the clock tree. But DFFs can involve a power overhead
Leakage Power

- Transistors leak currents even when in off-state
  - 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 $V_{dd}$

Leakage Power over Temperature

Drain current depends exponentially on thermal voltage $V_{th} = kT/q$

- Exponential $I_{DS}$ increase with temperature

$$ I_{DS} = I_{DS}^{0} e^{\frac{V_{DS} - V_{th}}{V_{th}}} $$

$$ V_{th} = \frac{kT}{q} $$

Example: 0.7V, 100nm process, 15mm2 die
Leakage in Transistor Stacks

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

\[ I_{\text{leakage}} = \frac{V_T}{g_m} \frac{W}{L} \]

Small-signal penalty ~ 20%

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

\[ t_{\text{delay}} = \frac{L}{g_m} \frac{L}{W} (V_{\text{DD}} - V_{\text{T}})^2 \]

\[ I_{\text{leakage}} = \frac{V_T}{g_m} \frac{W}{L} \frac{V_{\text{DD}} - V_{\text{T}}}{g_m} \frac{V_{\text{T}}}{g_m} \]

- Example: 55nm process

<table>
<thead>
<tr>
<th></th>
<th>HVT</th>
<th>SVT</th>
<th>LV</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>20ps</td>
<td>16ps</td>
<td>14ps</td>
<td></td>
</tr>
<tr>
<td>Leakage</td>
<td>30nW</td>
<td>60nW</td>
<td>200nW</td>
<td></td>
</tr>
</tbody>
</table>

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/VT/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

\[ V_{\text{BB}} = V_{\text{DD}} - 4V_{\text{T}} \]

\[ V_{\text{FB}} = V_{\text{BB}} < 0 \]

\[ V_{\text{RBB}} = V_{\text{BB}} > 0 \]

\[ V_{\text{T}} = V_{\text{DD}} - 4V_{\text{T}} \]

\[ V_{\text{FB}} = \text{Effect of body bias decreases for technologies below 100nm} \]

\[ V_{\text{RBB}} = \text{FBB is limited to ~300mV to avoid operating junction diodes in forward direction} \]
A low-power chip structure

A variable threshold inverter

Low-Power, Low-Voltage D-Latch with MTCMOS (Multi-threshold CMOS)

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:

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 $-s_{\text{spec}}$ ($V < V_{\text{spec}}$)

- Minimize power as long as meeting specs
- Start with high $V_T$ everywhere
- But on timing critical paths, lower $V_T$
- Further, you can also adjust $V_{DD}$ at play with $V_T$ (bias voltage programming)
  - multiple $V_{DD}$, multiple $V_T$, etc.