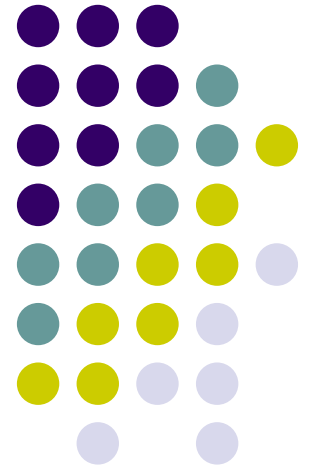
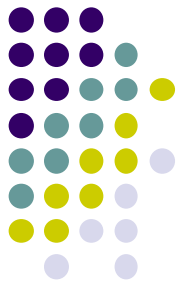


Low Power Design

Awais M. Kamboh
kambohaw@msu.edu
Department of Electrical and Computer Engineering
Michigan State University, East Lansing

May 30th, 2008





Power Dissipation in CMOS

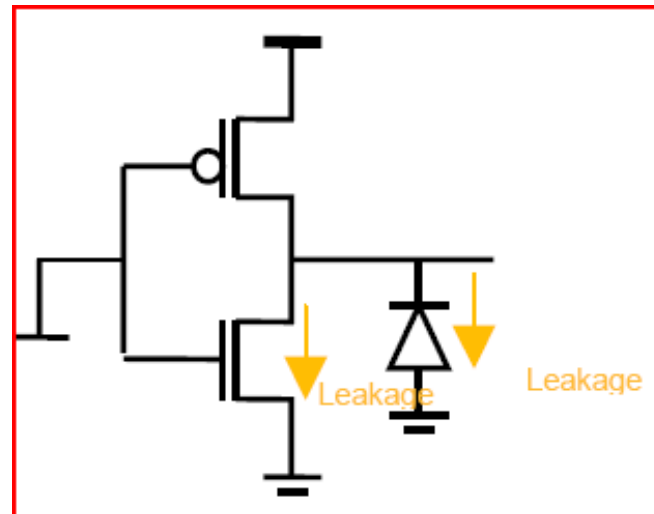
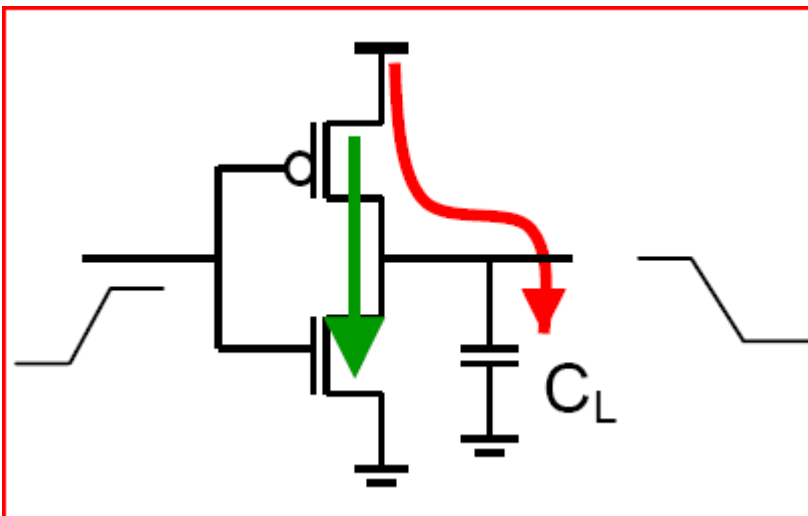
- Total Power = Dynamic + Leakage

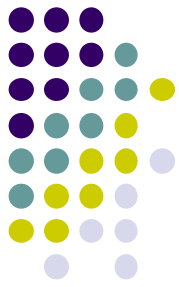
$$P_{\text{switching}} = a.f.C_{\text{eff}}V_{\text{dd}}\Delta V$$

$$P_{\text{short-circuit}} = I_{\text{sc}}V_{\text{dd}}f$$

$$P_{\text{leakage}} = f(V_{\text{dd}}, V_{\text{th}}, W/L) = I_{\text{static}}V_{\text{dd}}$$

- Total Power: Function of switching activity, capacitance, voltage and transistor structure





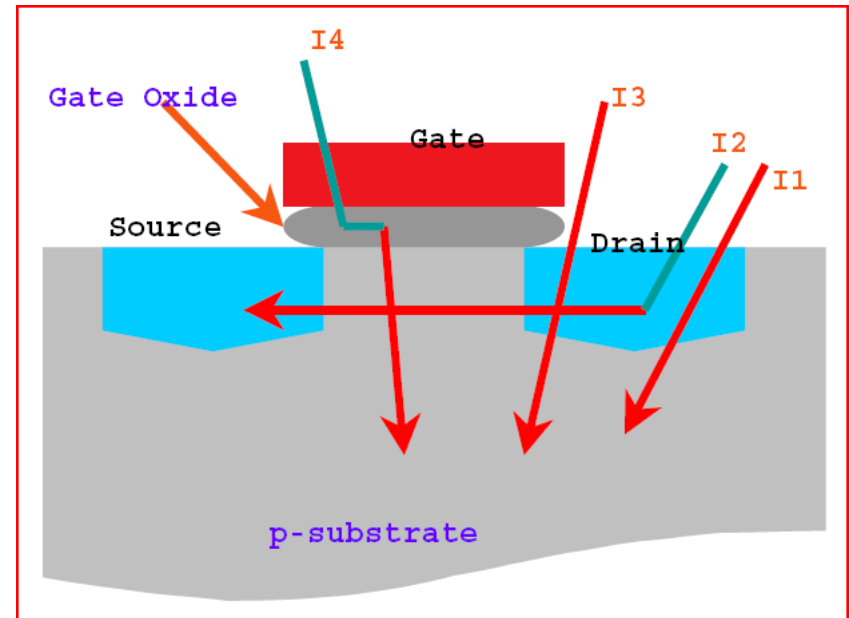
Power Dissipation in CMOS

Dynamic Power

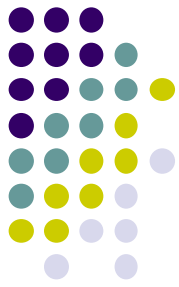
- Switching: when charging/discharging internal and wire capacitances
- Short Circuit: Instantaneous short circuit when gate switches state

Leakage Power

- I1: Diode reverse bias current (Drain-Body)
- I2: Sub-threshold current (Drain-Source)
- I3: Gate-induced drain leakage (Drain-Body)
- I4: Gate oxide leakage (Gate-Body)



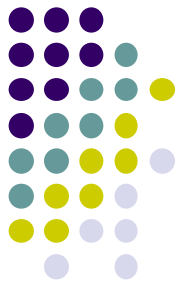
Dynamic + Leakage Power Reduction Techniques



- **Multiple Supply Voltages**
- **Dynamic Power Savings** 40–50%
- **Leakage Power Savings** 2X
- **Timing Penalty** ~0% Adds level shifters; clock scheduling issues due to latency Changes
- **Area Penalty** <10% Power routing and power interconnect; level shifters
- **Complexity Penalty** High (Design time, turnaround time, TTM)

Selected functional blocks are run at different supply voltages.

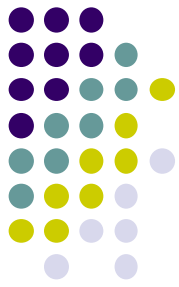
Multiple Supply Voltage



Multi-supply voltage techniques operate different blocks at different voltages. Running at a lower voltage reduces power consumption, but at the expense of speed. Designers use different supply voltages for different parts of the chip based on their performance requirements. MSV implementation is key to reducing power since lowering the voltage has a squared effect on active power consumption. MSV techniques require level shifters on signals that go from one voltage level to another. Without level shifters, signals that cross voltage levels will not be sampled correctly.

A power domain is a collection of design blocks or instances that share the same supply voltage. The power domain concept is used to describe switchable blocks with different supply voltages. Level shifters must ensure the proper drive strength and accurate timing as signals transition from one voltage level to another.

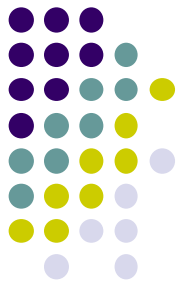
Dynamic + Leakage Power Reduction Techniques



- **DVFS (Dynamic Voltage Frequency Scaling)**
- **Dynamic Power Savings** 40–70%
- **Leakage Power Savings** 2–3X
- **Timing Penalty** ~0% Adds level shifters, power-up sequence; clock scheduling issues due to dynamic latency changes
- **Area Penalty** <10% Adds level shifters and a power management unit
- **Complexity Penalty** High (Design time, turnaround time, TTM)

Selected portions of the device are dynamically set to run at different voltages and frequencies on the fly while the chip is running. Used for dynamic power reduction.

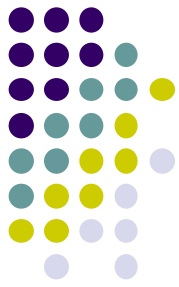
Dynamic Voltage Frequency Scaling



Dynamic voltage and frequency scaling (DVFS) techniques—along with associated techniques such as dynamic voltage scaling (DVS) and adaptive voltage and frequency scaling (AVFS)—are very effective in reducing power, since lowering the voltage has a squared effect on active power consumption. DVFS techniques provide ways to reduce power consumption of chips on the fly by scaling down the voltage (and frequency) based on the targeted performance requirements of the application. Since DVFS optimizes both the frequency and the voltage, it is one of the only techniques that is highly effective on both dynamic and static power.

Dynamic voltage scaling is a subset of DVFS that dynamically scales down the voltage (only) based on the performance requirements. Adaptive voltage and frequency scaling is an extension of DVFS. In DVFS, the voltage levels of the targeted power domains are scaled in fixed discrete voltage steps. Frequency-based voltage tables typically determine the voltage levels. It is an open-loop (no feedback) system with large margins built in, and therefore the power reduction is not optimal. On the other hand, AVFS deploys closed-loop voltage scaling and is compensated for variations in temperature, process, and $V=IR$ drop using dedicated circuitry (typically analog in nature) that constantly monitors performance and provides active feedback. Although the control is more complex, the payoff in terms of power reduction is higher.

Dynamic Voltage Frequency Scaling

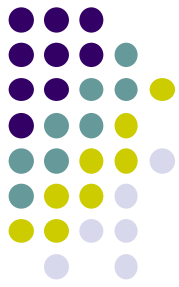


To reduce the total power consumption of the chip, the design uses variable supply voltages for different parts of the chip based on their performance requirements.

Here are the requirements for DVFS:

- Variable power supply, Capable of generating required voltage levels,
- Minimal transition energy losses,
- Quick voltage-transient response,
- Voltage scaling (voltage-independent design),
- Scale the frequency in the same proportion to meet signal propagation delay requirements,
- Power scheduler that intelligently computes the appropriate frequency and voltage levels needed to execute the various tasks or jobs.

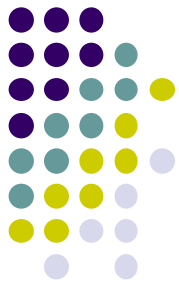
Dynamic Power Reduction Techniques



- ***Clock gating***
- **Dynamic Power Savings 20%**
- **Leakage Power Savings ~0X**
- **Timing Penalty ~0% (*Clock tree insertion delay*)**
- **Area Penalty <2%**

Portions of the clock tree(s) that aren't being used at any particular time are disabled.

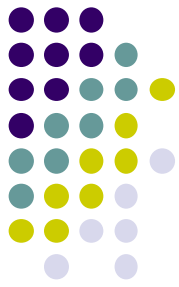
Clock Gating



In normal operation, the clock signal continues to toggle at every clock cycle, whether or not its registers are changing. Clock trees are a large source of dynamic power because they switch at the maximum rate and typically have larger capacitive loads. If data is loaded into registers only infrequently, a significant amount of power is wasted. By shutting off blocks that are not required to be active, clock gating ensures power is not dissipated during the idle time. Clock gating can occur at the leaf level (at the register) or higher up in the clock tree. When clock gating is done at the block level, the entire clock tree for the block can be disabled. The resulting reduction in clock network switching becomes extremely valuable in reducing dynamic power.

Power is not dissipated during the idle period when the register is shut off by the gating function. The logic on the enable circuitry in the original design is removed.

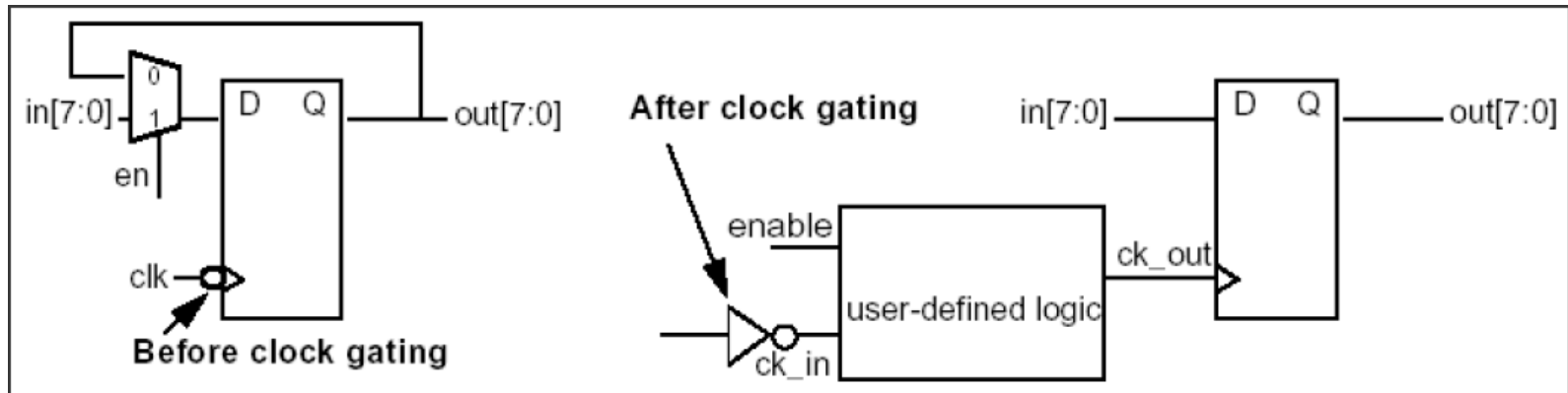
Clock Gating



Clock-Enabled Register Example

Consider a multiplexer (MUX) at the data input of a register. This MUX is controlled by an enable signal (select line).

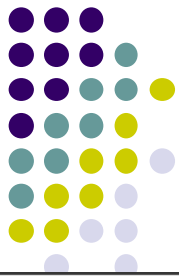
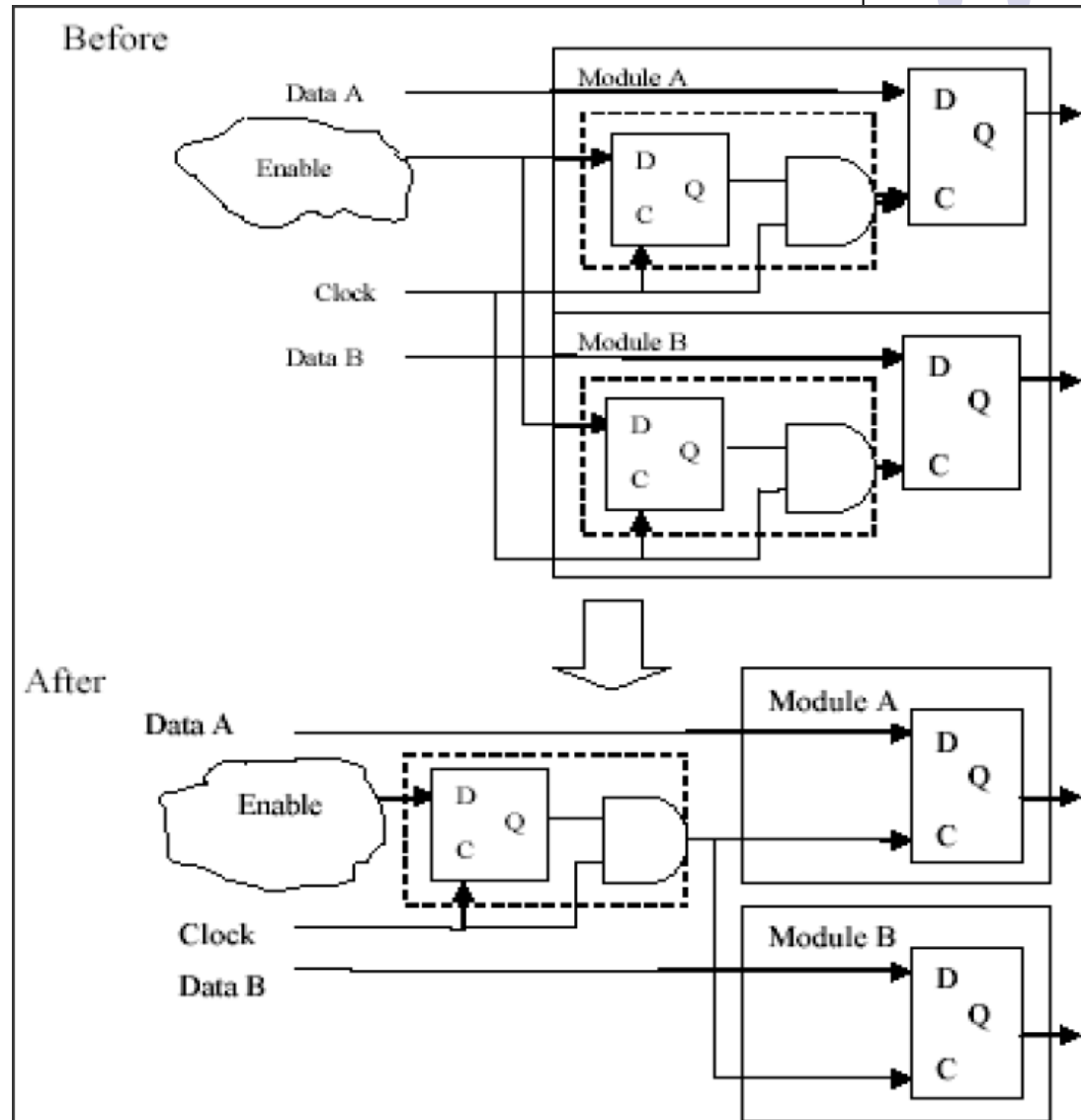
This type of description is a perfect candidate for clock gating. If the data input to a flip-flop can be reduced to a MUX between the data pin and the output pin of the flip-flop, the synthesis tool can model this flip-flop by connecting the “data input” directly to the data pin of the flip-flop, and by using the MUX enable to gate the clock signal of the flip-flop via an inserted clock-gating element.



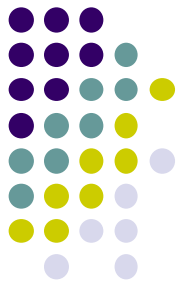
Clock Gating

De-Cloning Local Clock Gating

If the clock-gating logic of different registers in the design uses the same enable signal, these clock-gating instances can be merged for any such identically gated registers. This process is called clock-gating de-cloning.

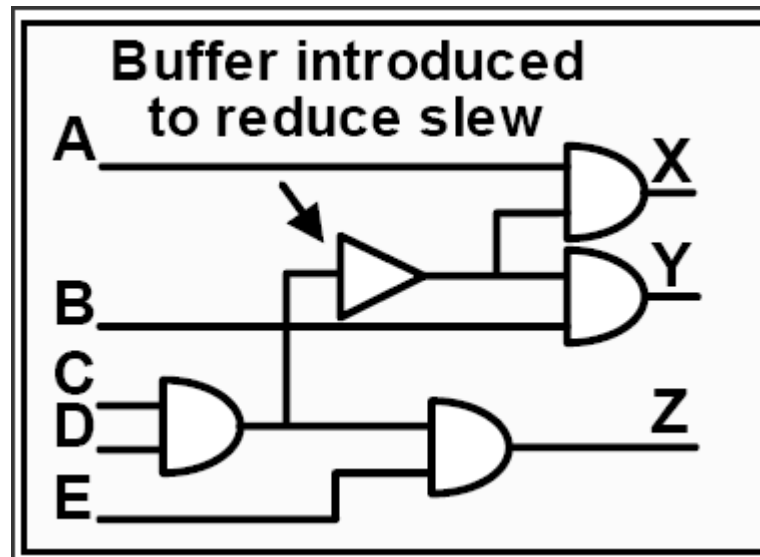


Dynamic Power Reduction Techniques

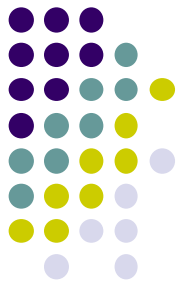


- **Transition rate buffering**
- **Dynamic Power Savings** <5%
- **Leakage Power Savings** ~0X
- **Timing Penalty** ~0%
- **Area Penalty** *Little*

Buffer manipulation reduces dynamic power by minimizing switching times.



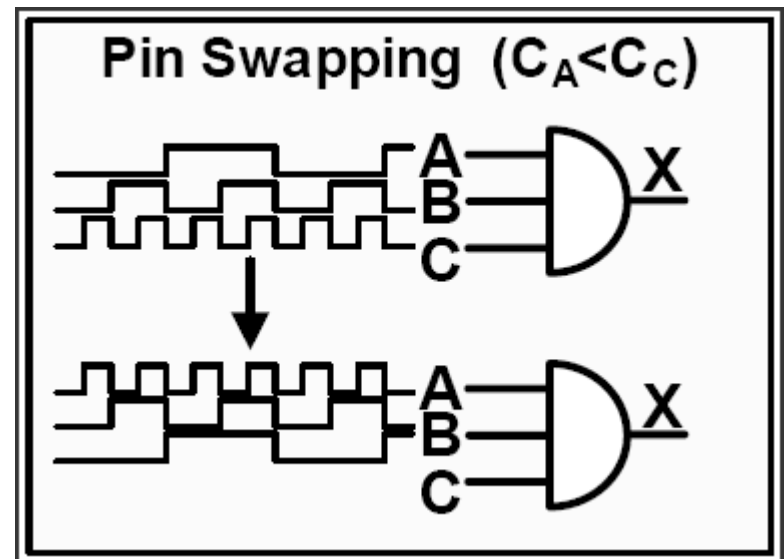
Dynamic Power Reduction Techniques



- **Pin swapping**
- **Dynamic Power Savings** <5%
- **Leakage Power Savings** ~0X
- **Timing Penalty** ~0%
- **Area Penalty** None

By swapping gate pins, switching occurs at gates/pins with lower capacitive loads.

The pins are swapped so that most frequently, switching occurs at the pins with lower capacitive load. Since the capacitive load of pin A is lower, there is less power dissipation.



Dynamic Power Reduction Techniques

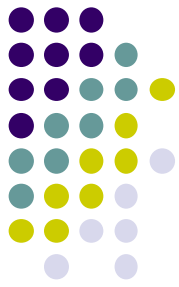


- ***Operand isolation***
- **Dynamic Power Savings** <5%
- **Leakage Power Savings** ~0X
- **Timing Penalty** ~0% (May add a few gates to pipeline)
- **Area Penalty** None (May add a few gates to pipeline)

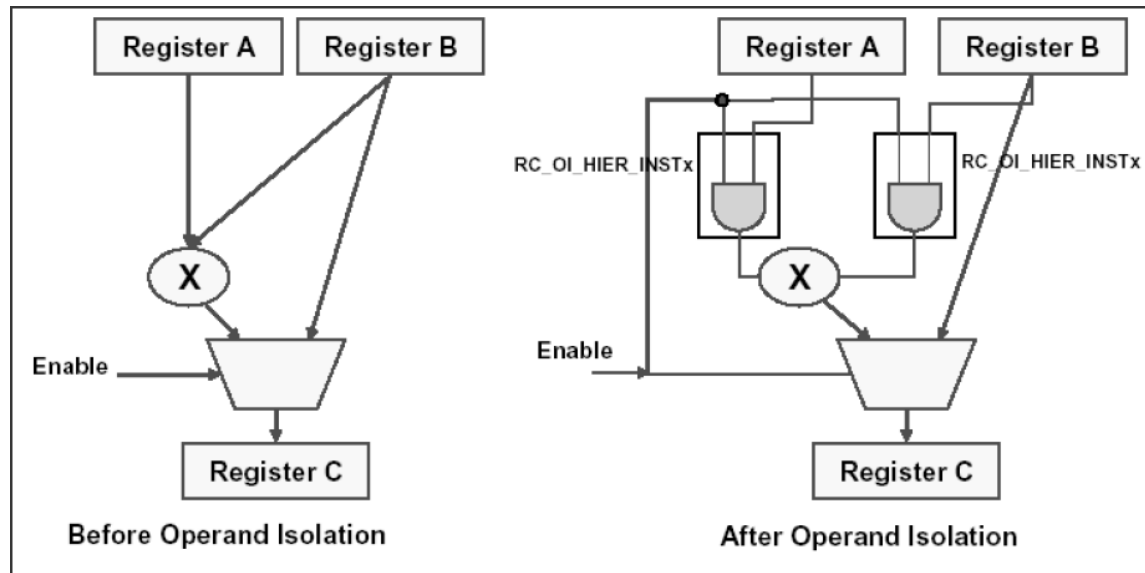
Reduce power dissipation in datapath blocks controlled by an enable signal; when the datapath element is not active, prevent it from switching.

Often, datapath computation elements are sampled only periodically. This sampling is controlled by an enable signal. When the enable is inactive, the datapath inputs can be forced to a constant value (disabled) so that unnecessary switching power is not wasted in the datapath. The result is that the datapath will not switch, saving dynamic power.

Operand Isolation



In the figure, in the digital system shown as Before Operand Isolation, register C uses the result of the multiplier when the enable is on. When the enable is off, register C uses only the result of register B, but the multiplier continues its computations. Because the multiplier dissipates the most power, the total amount of power wasted is quite significant. One solution to this problem is to shut down (isolate) the function unit (operand) when its results are not used, as shown in After Operand Isolation. AND gates are inserted at the inputs of the multiplier and the enable logic of the multiplier is used to gate the signal transitions. As a result, no dynamic power is dissipated when the result of the multiplier is not needed.



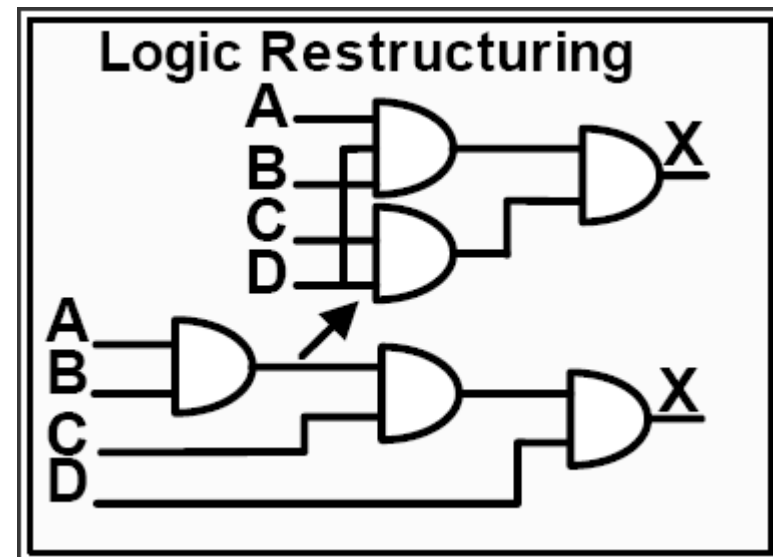
Dynamic Power Reduction Techniques



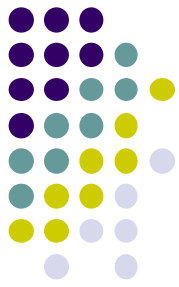
- **Logic restructuring**
- **Dynamic Power Savings** <5%
- **Leakage Power Savings** ~0X
- **Timing Penalty** ~0%
- **Area Penalty** Little

Move high switching operations up in the logic cone, and low switching operations back in the logic cone.

A gate-level dynamic power optimization technique, logic restructuring can, for example, reduce three stages to two stages through logic equivalence transformation, so the circuit has less switching and fewer transitions.



Leakage Power Reduction Techniques



- **Memory splitting**
- **Dynamic Power Savings** ~0%
- **Leakage Power Savings** *Varies*
- **Timing Penalty** *Varies Adds isolation cells for power shutoff*
- **Area Penalty** *Varies*
- **Complexity Penalty** *Medium-high*

If the software and/or data are persistent in one portion of a memory but not in another, it may be appropriate to split that block of memory into two or more portions. One can then selectively power down those portions that aren't in use.

In many systems, the memory capacity is designed for peak usage. During normal system activity, only a portion of that memory is actually used at any given time. In many cases, it is possible to divide the memory into two or more sections, and selectively power down unused sections of the memory. With increasing SoC memory capacity, reducing the power consumed by memories is increasingly important.

Leakage Power Reduction Techniques



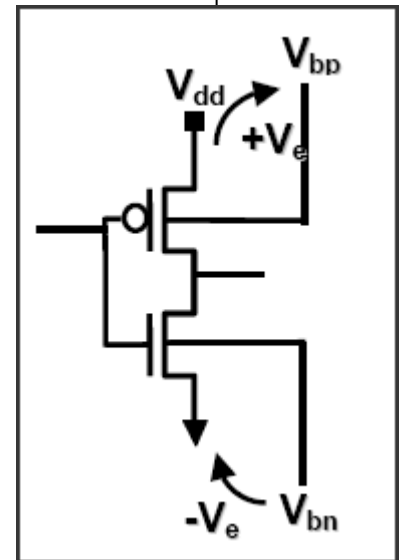
- ***Substrate biasing***
- **Dynamic Power Savings** *~0%*
- **Leakage Power Savings** *10X*
- **Timing Penalty** *10%*
- **Area Penalty** *<10%*
- **Complexity Penalty** *High*

Substrate biasing in PMOS biases the body of the transistor to a voltage higher than V_{dd} ; in NMOS, to a voltage lower than V_{ss} .

Substrate Biasing

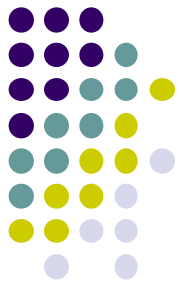
Since leakage currents are a function of device V_{th} , substrate biasing, also known as back biasing, can reduce leakage power.

With this technique, the substrate or the appropriate well is biased to raise the transistor thresholds, thereby reducing leakage. In PMOS, the body of transistor is biased to a voltage higher than V_{dd} . In NMOS, the body of transistor is biased to a voltage lower than V_{ss} . Since raising V_{th} also effects performance, some advanced techniques allows the bias to be applied dynamically, so during an active mode of operation the reverse bias is small, while in standby the reverse bias is stronger.



Area and routing penalties are incurred. An extra pin in the standard cell library is required and special library cells are necessary. Body-bias cells are placed throughout the design to provide voltages for transistor bulk. To generate the bias voltage, a substrate-bias generator is required, which also consumes some dynamic power, partially offsetting the reduced leakage. Substrate bias returns are diminishing at smaller processes in advanced technologies. At 65nm and below, the body-bias effect decreases, reducing the leakage control benefits. TSMC has published information pointing to a factor of 4x reduction at 90nm, and only 2x moving to 65nm. Consequently, substrate biasing is predicted to be overshadowed by power gating.

Leakage Power Reduction Techniques

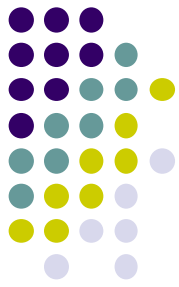


- **Multi-V_{th}**
- **Dynamic Power Savings** 0%
- **Leakage Power Savings** 2–3X
- **Timing Penalty** ~0%
- **Area Penalty** 2 to –2%
- **Complexity Penalty** Low

Individual logic gates use transistors with low switching thresholds (faster with higher leakage) or high switching thresholds (slower with lower leakage).

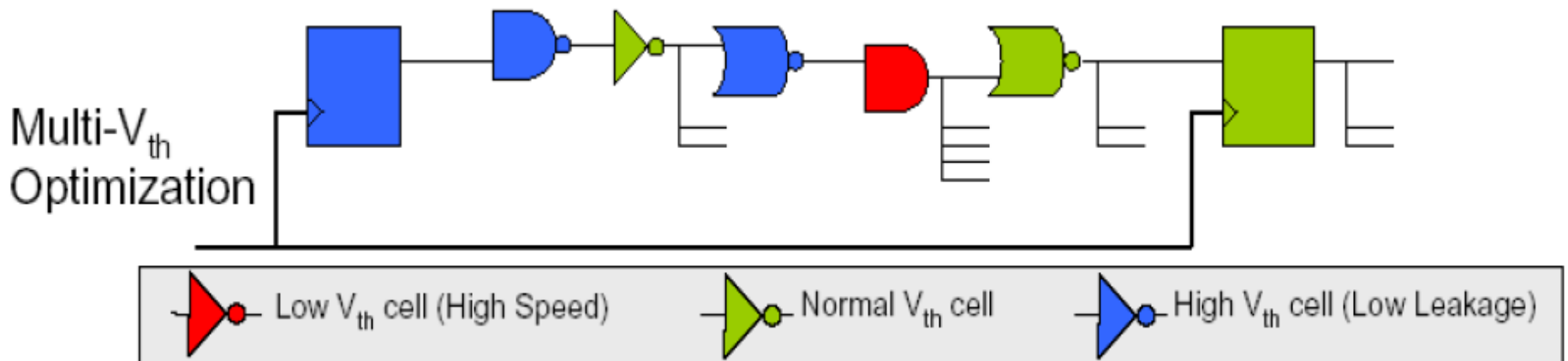
Multi-V_{th} optimization utilizes gates with different thresholds to optimize for power, timing, and area constraints. Most library vendors provide libraries that have cells with different switching thresholds. Good synthesis tools for low-power applications are able to mix available multi-threshold library cells to meet speed and area constraints with the lowest power dissipation. This complex task optimizes for multiple variables and so is automated in today's synthesis tools.

Multi-V_{th}

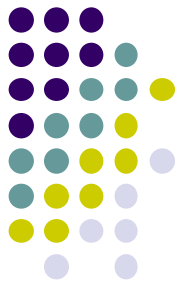


The most common leakage reduction technique is to use specially designed high-V_{th} cells where possible. The low-V_{th} gates switch more quickly in response to their input signals, but consume more leakage power. The high-V_{th} gates switch more slowly, but consume less leakage power.

The synthesis tool should be able to limit the maximum leakage power for the design by performing multi-V_{th} leakage optimization. The compiler chooses cells with high V_{th} to replace the cells with low V_{th} in areas where it won't affect critical timing paths. Low-V_{th} cells are placed in areas that do not meet timing.



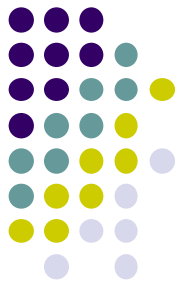
Leakage Power Reduction Techniques



- **Power shutoff (PSO)**
- **Dynamic Power Savings** ~0%
- **Leakage Power Savings** 10–50X
- **Timing Penalty** 4–8% *Adds isolation cells, complex timing, wakeup time, rush currents*
- **Area Penalty** 5–15% *Adds isolation cells, state retention cells, always-on cells; may have wider power grid due to rush currents; power management unit*
- **Complexity Penalty** *High (System architecture, support for power control, verification, synthesis, implementation, Design For Test)*

When not in use, selected functional blocks are individually powered down.

Power Shut Off



One of the most effective techniques, PSO, also called *power gating*, switches off power to parts of the chip when these blocks are not in use. This technique is increasingly being used in the industry and can eliminate up to 96 percent of the leakage current. Power gating is employed to shut off power in standby mode. A specific powerdown sequence is needed, which includes isolation on signals from the shut-down domain. Erroneous power-up/down sequences are the root cause of errors that can cause a chip re-spin. This needs to be correctly and exhaustively verified along with functional netlist to ensure that the chip functions correctly with sections turned off and that the system can recover after powering up these units. Deploying power shutoff also requires isolation logic and possibly state retention of key state elements or, in other words, state retentive power gating (SRPG). For multi-supply voltage (MSV), level shifters are also needed.

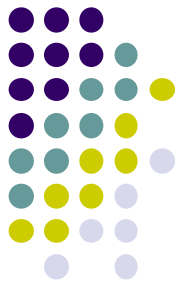
There are three main components of Power Shut Off

Isolation

State Retention

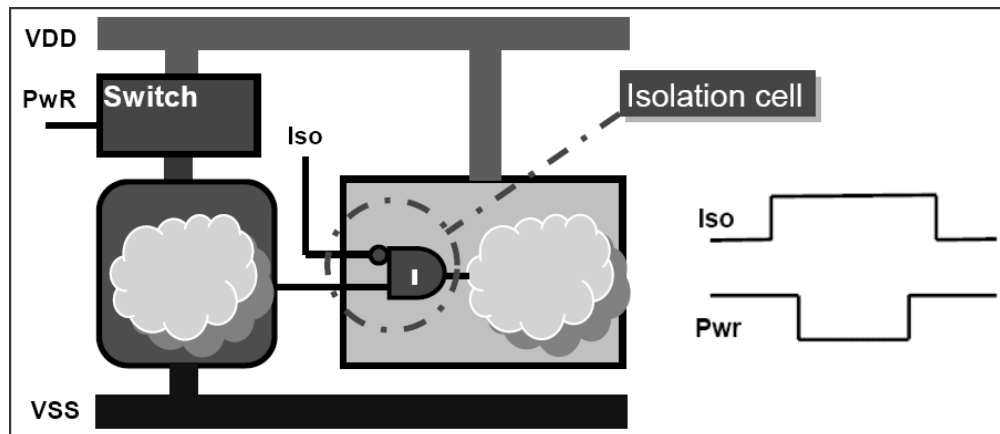
Power Cycle Sequence

Power Shut Off

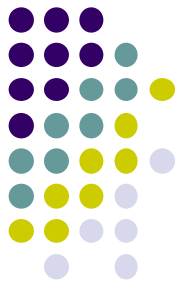


Isolation

Isolation logic is typically used at the output of a powered-down block to prevent floating, unpowered signals (represented by unknown or X) from propagating from powered-down blocks. The outputs of blocks being powered down need to be isolated before power can be switched off; and they need to remain isolated until after the block has been fully powered up. Isolation cells are placed between two power domains and are typically connected from domains powered off to domains that are still powered up. In some cases, isolation cells may need to be placed at the block inputs to prevent connection to powered-down logic. If the driving domain can be OFF when the receiving domain is ON, the receiving domain needs to be protected by isolation. The isolation cells may be located in the driving domain, with special isolation cells, or they may be in the receiving domain.

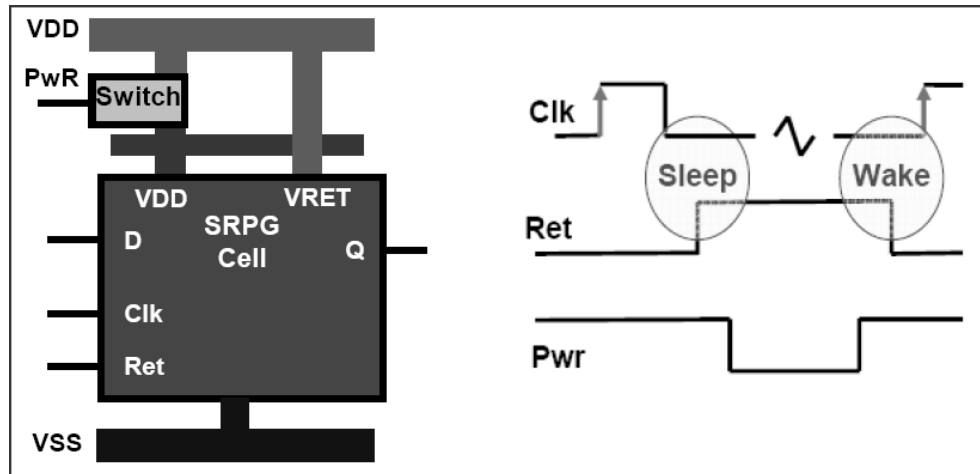


Power Shut Off

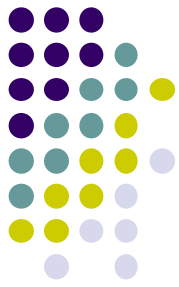


State Retention

In certain cases, the state of key control flops needs to be retained during poweroff. To speed power-up recovery, state retention power gating (SRPG) flops can be used. These retain their state while the power is off, provided that specific control signaling requirements are met. Cell libraries today include such special state retention cells. A key area of verification is checking that these library-specific requirements have been satisfied and the flop will actually retain its state.

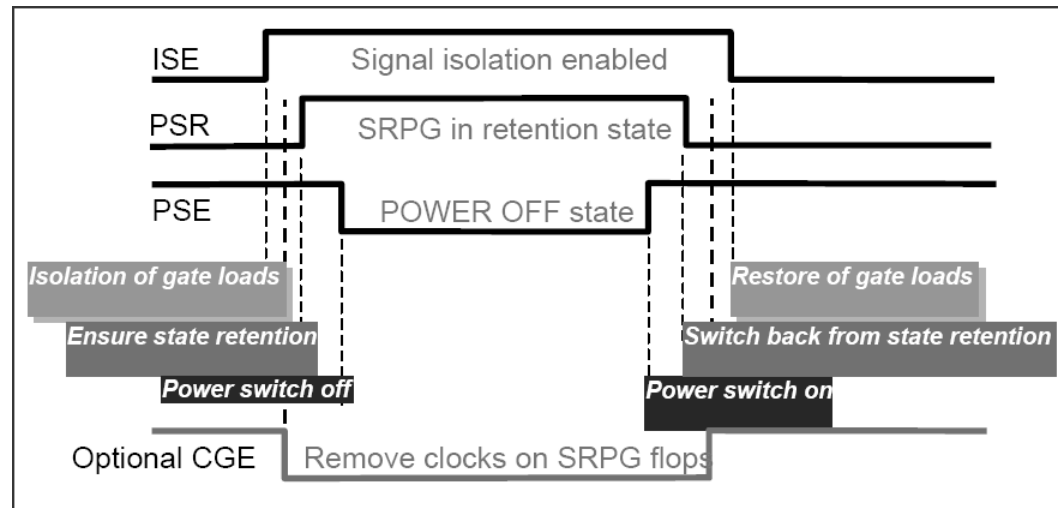


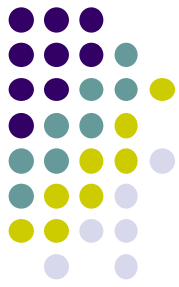
Power Shut Off



Power Cycle Sequence

For power-down, a specific sequence is generally followed: isolation, state retention, power shutoff. For the power-up cycle, the opposite sequence needs to be followed. The power-up cycle can also require a specific reset sequence. Given that there are multiple—possibly nested—power domains, coupled with different power sequences, some of which may share common power control signals and multiple levels of gated clocks, the need for verification support is tremendous. The complexity and possible corner cases need to be thoroughly analyzed; functional and power intent must be analyzed and thoroughly verified together using advanced verification techniques.



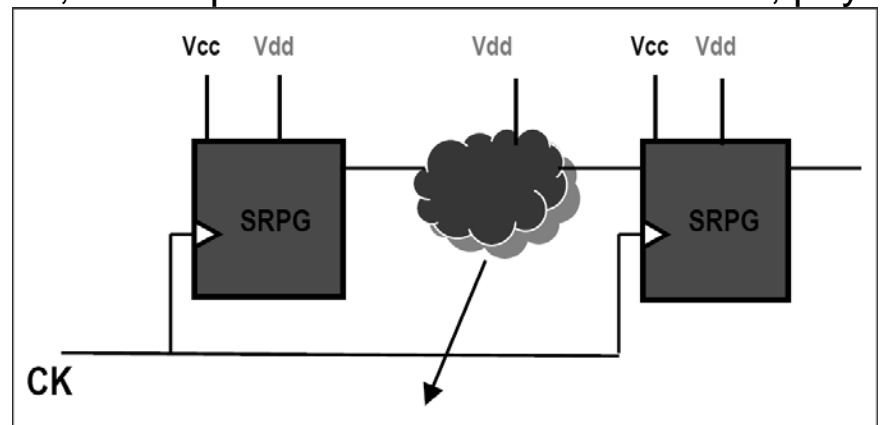


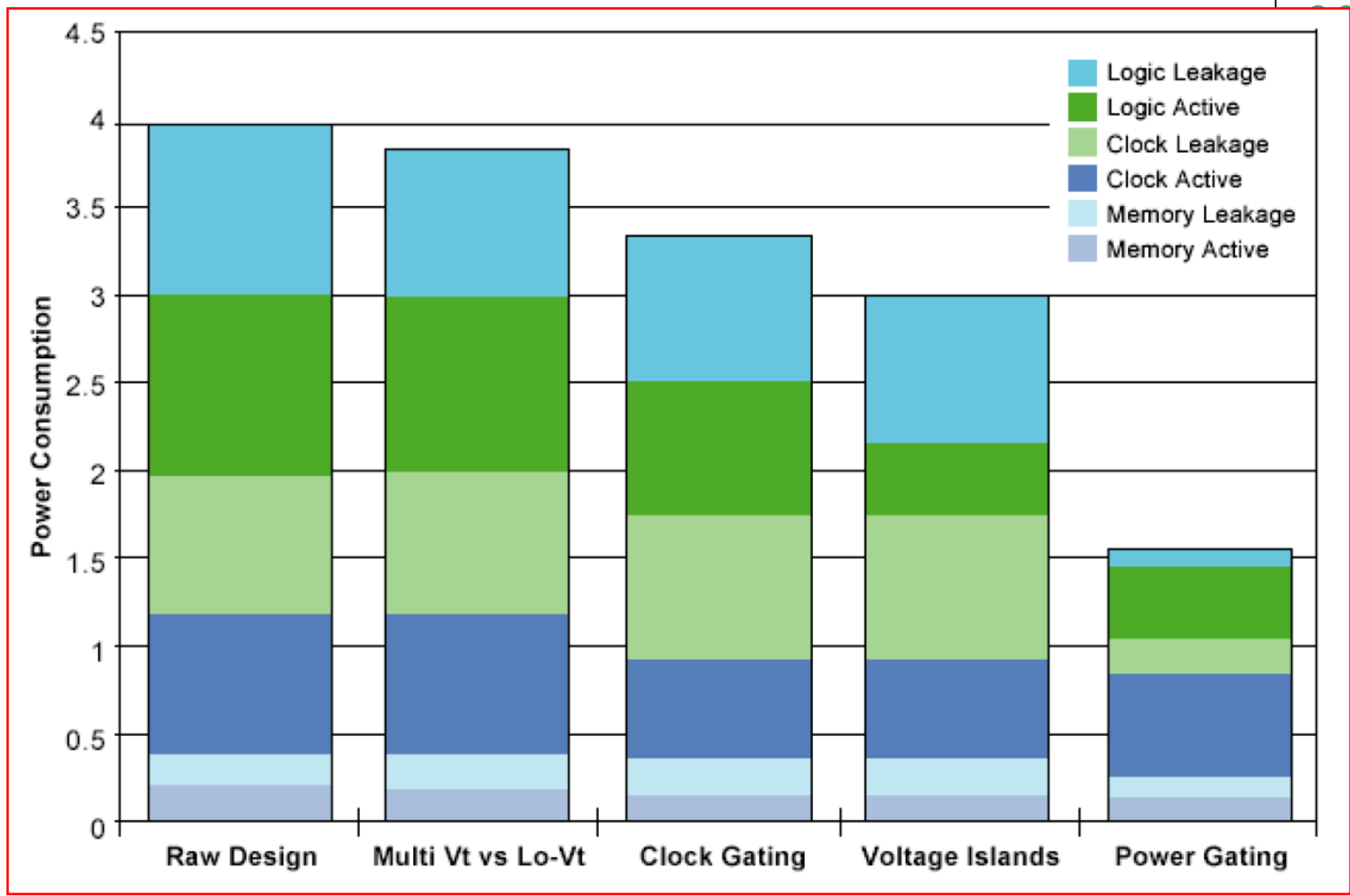
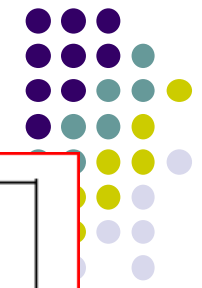
Power Shut Off

State Retention Power Gating (SRPG)

The use of power-gating state retention cells allows a system to shut down power to certain block(s) in a design, and recover the prior states after a power-up sequence. To implement power gating, special state retention cells are required to store prior state(s) of the blocks before power-down. The basic flip-flop has been modified in SRPG, and the master latch runs on the same power supply V_{dd} as combinational logic, while the slave latch runs on the different power supply V_{cc} . The state of the system will be retained in the flip-flops during power down and all the combinational logic will be turned off during sleep mode.

The advantages of SRPG include shutdown leakage savings, which can be independent of process variations. It allows for faster system power-on because the state is preserved in the slave latch. Disadvantages include increased area and die size; timing penalties such as increased signal and clocking delays; increased routing resources (power routing for V_{cc} and a power-gating signal tree with on buffers); specialized library models for SRPG cells; additional power overhead in the active mode; and impacts to functional verification, physical integration, and DFT.





@ 90nm technology node.