數位系統之低功率設計技術簡介

Introduction to Low-Power Design Techniques for Digital Systems

Prof. Chien-Nan Liu
Institute of Electronics
National Chiao-Tung Univ.

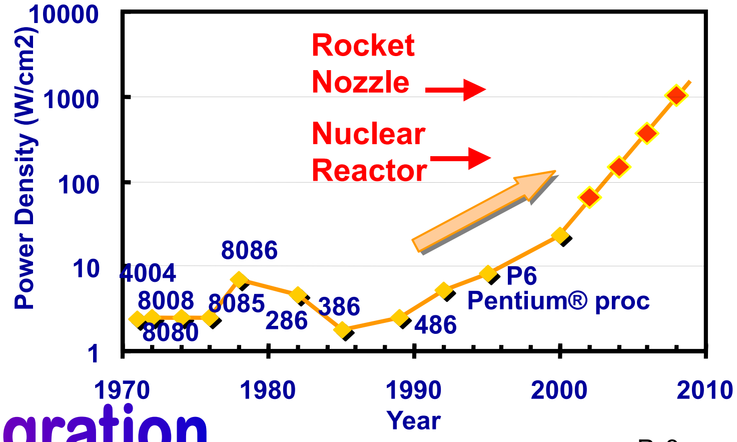
Tel: (03)5712121 ext:31211
E-mail: jimmyliu@nctu.edu.tw
http://www.ee.ncu.edu.tw/~jimmy

Outline

- **Introduction**
- **Multiple Power Domains**
- **Dynamic Voltage and Frequency Scaling**
- **Clock Gating**
- **Dual Threshold Voltages**
- **Power Gating**
- **Conclusion**

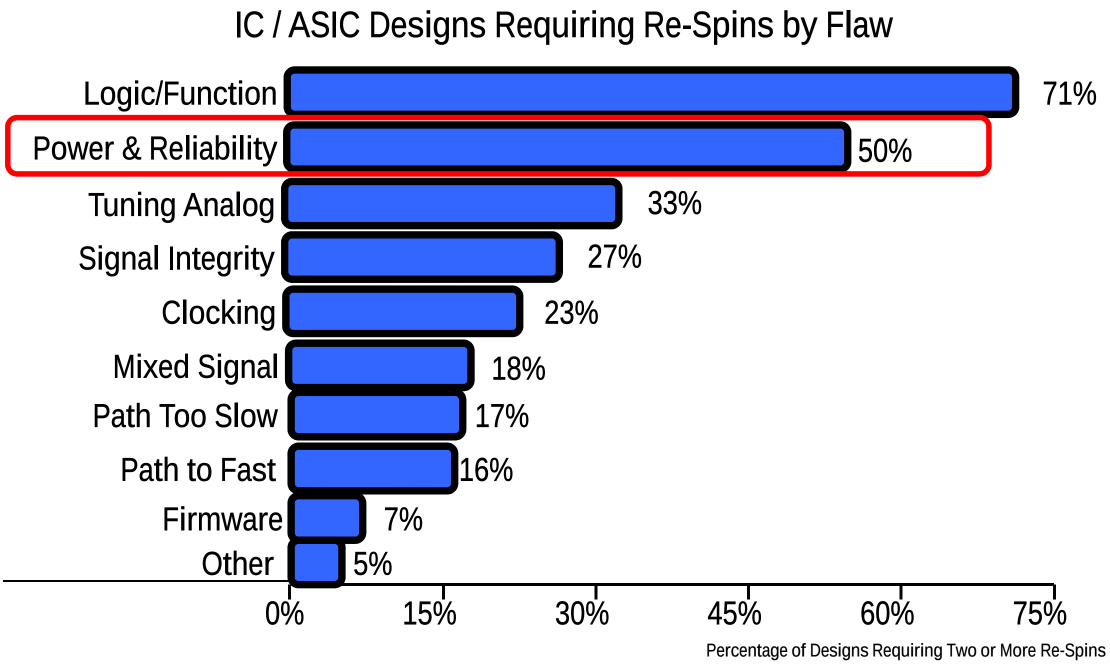
Motivation

- **Portability requirement is increasing**
  - Longer battery life with increased functionality
- **Power related issues are getting serious**
  - Heat dissipation requirement (rocket nozzle?)
  - Electromigration
  - Power/Ground bounce due to IR-drop
- **Power is as important as performance**
  - Should be considered in the design phase



Source: Courtesy, *Intel*

Serious Power-Related Issues



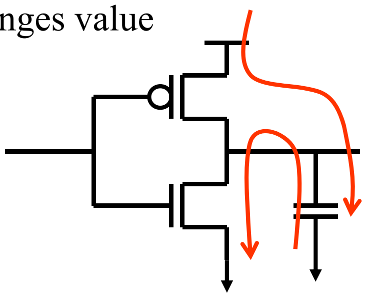
Source: Courtesy, *Synopsys*

Where is Power Consumed?

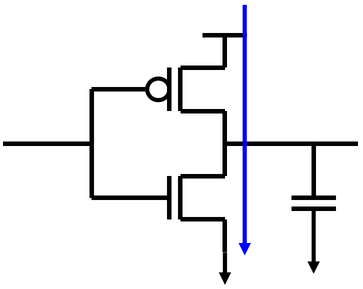
- **Dynamic power (switching power)**
  - Power consumed when the output of a gate changes value
  - Quadratically dependent on voltage
  - **$P = C_L V_{dd}^2 f$**

    C_L: loading capacitance; V_{dd}: supply voltage

    f: transition frequency



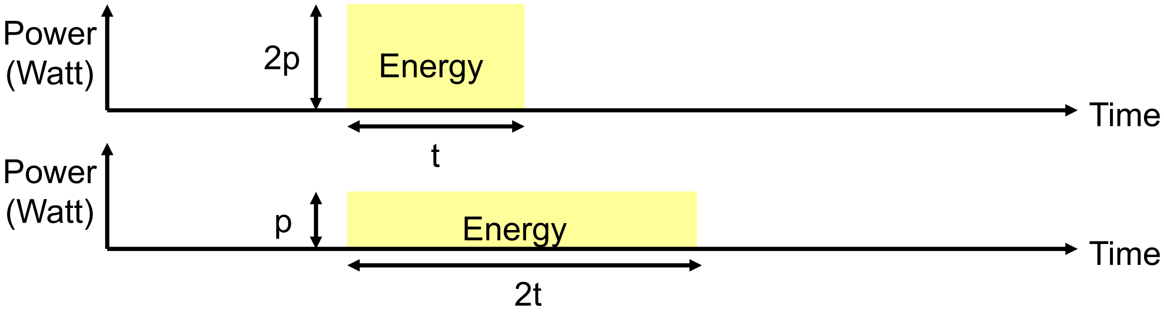
- **Static power (leakage power)**
  - Power consumed by each element at all times
    - From several sources
  - Grows exponentially when voltage is reduced
  - Increases as transistor size shrinks



Heterogeneous Integration

Power Vs. Energy

$$Power = \frac{Energy}{T}$$

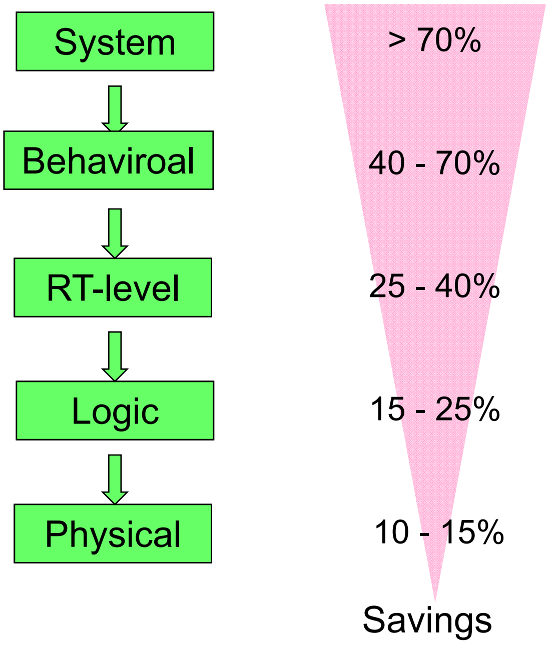


- **Two approaches require the same energy**
- **Lower power design could simply be slowed down ??**

Heterogeneous Integration

Power Saving Opportunities

- **A power-conscious design methodology addresses power at every level of design hierarchy**
- **Applying low-power techniques at higher design levels can reduce more power**
- **The focuses of this slide are the low-power technique at RTL or higher design levels**



Ref : [1]

Power Reduction Techniques

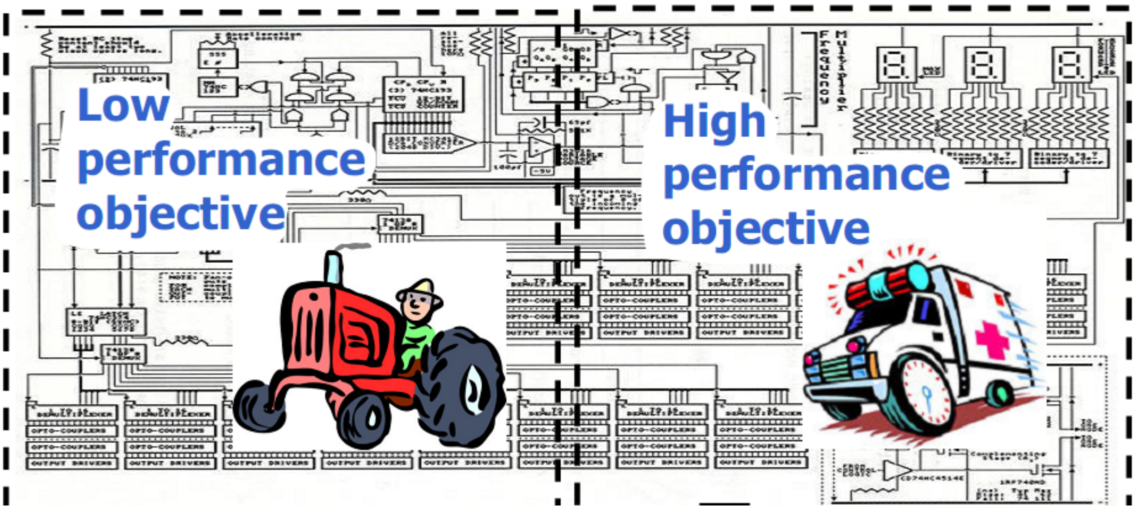
- **Reduce supply voltage** (Vdd↓)
  - **Multiple power domains**
    - Lower Vdd value for non-critical blocks
- **Reduce signal transitions** (f↓)
  - **Dynamic voltage and frequency scaling (DVFS)**
    - Reduce frequency for light-loading blocks
  - **Clock gating**
    - Freeze clock signal
- **Use high-Vt cells for lower leakage power** (leakage↓)
  - **Dual threshold voltages**
    - Replace the cells on non-critical paths with high-VT cell
- **Shut down power supply when not used** (Vdd↓, leakage↓)
  - **Power gating (MTCMOS)**

Outline

- Introduction
- Multiple Power Domains
- Dynamic Voltage and Frequency Scaling
- Clock Gating
- Dual Threshold Voltages
- Power Gating
- Conclusion

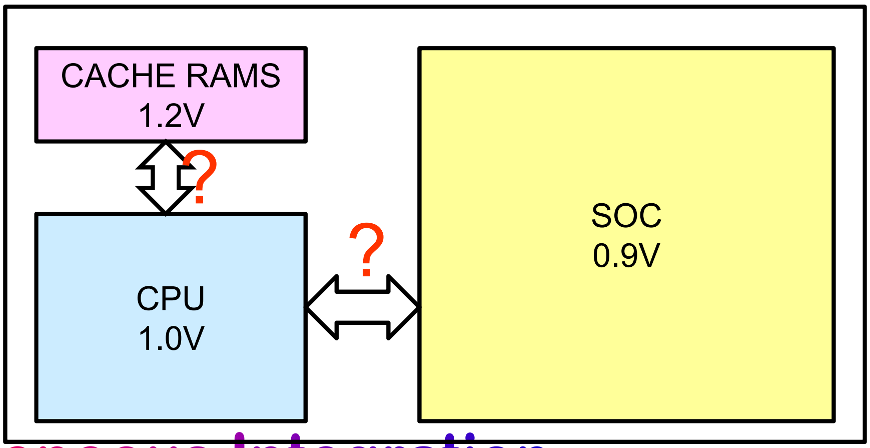
Multiple Power Domain Concept

- Different components run with different voltages
- Also known as multi-supply-voltage (MSV) designs



Adv. of Multiple Power Domains

- Since dynamic power is proportional to V_{dd}^2, lowering V_{dd} on selected blocks helps reduce power significantly.
- Unfortunately, lowering the voltage also increases the delay of the gates in the design
- Level shifters are used to propagate the signals between different domains



Heterogeneous Integration

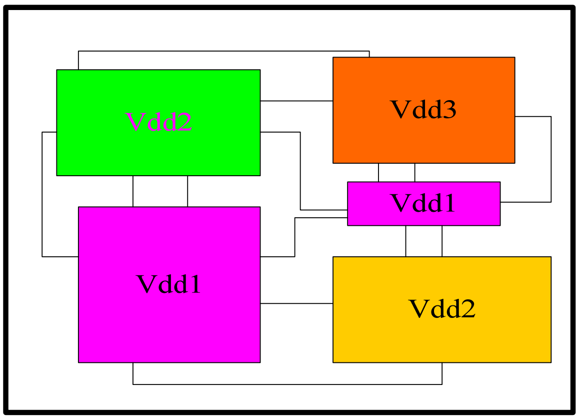
Basic MSV Design Principles

- Adopt multiple supply voltages (MSV) to trade performance and power saving
- Assign higher VDD to timing-critical cells for timing optimization
  - GOOD for performance
  - BAD in power consumption
- Assign lower VDD to non-timing-critical cells for power saving
  - BAD for performance
  - GOOD in power consumption

Heterogeneous Integration

Physical Implementation Requirements

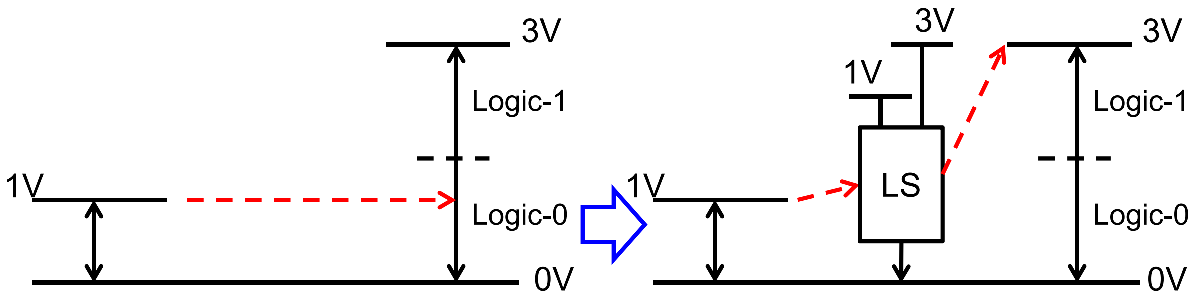
- **The devices with the same supply voltage will be put together**
  - Form a "voltage island"
- **Creation of voltage islands**
  - During high-level synthesis or physical synthesis
- **Physical implementation**
  - Floorplanning of voltage islands
  - Power network planning
- **As few voltage levels as possible**
  - Reduce the types of level shifters and ease the power/ground distribution
  - Level shifter itself occupies area and consumes power
  - Level shifter takes time to switch → increases delay



Three supply voltages and five voltage islands

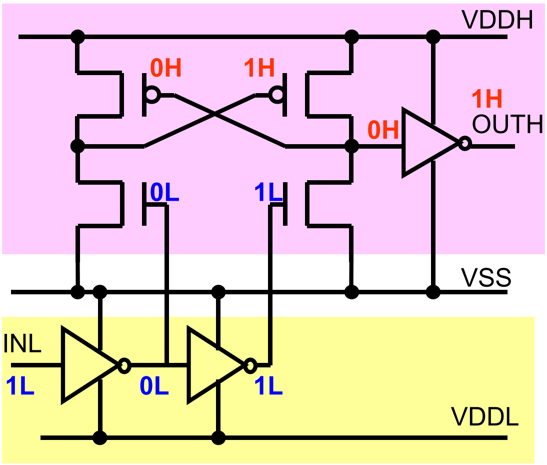
Low-to-High Level Shifter – Case 1

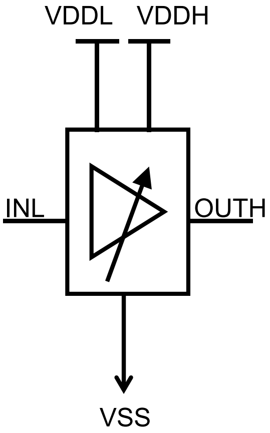
- **1V swing may not reach threshold in the 3V domain**
  → Add Level Shifters



Low-to-High Level Shifter – Case 1

- **"Up-shifting" level converters require two supply rails and share a common ground**



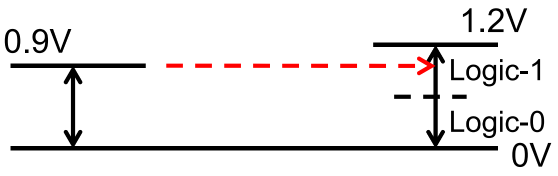


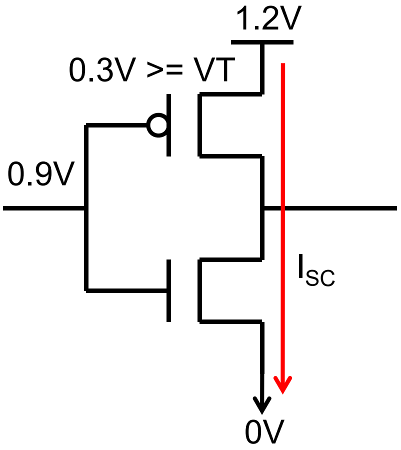
Ref:[2]

Heterogeneous Integration

Low-to-High Level Shifter – Case 2

- **A 0.9V signal driving a 1.2V gate will turn on both the NMOS and PMOS networks, causing short-circuit currents**

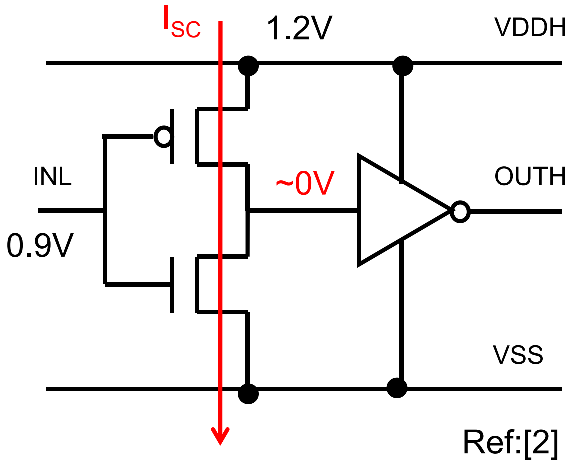




Heterogeneous Integration

Low-to-High Level Shifter – Case 2

- **A low voltage swing input signal would not necessarily be strong enough to turn the NMOS input transistor fully on.**
  - Internal signal cannot be discharged to 0V

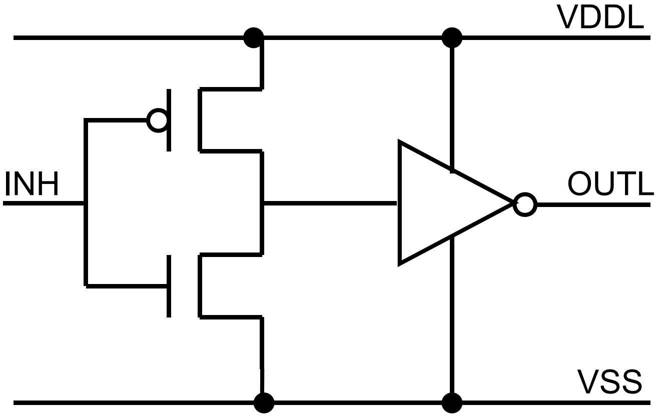


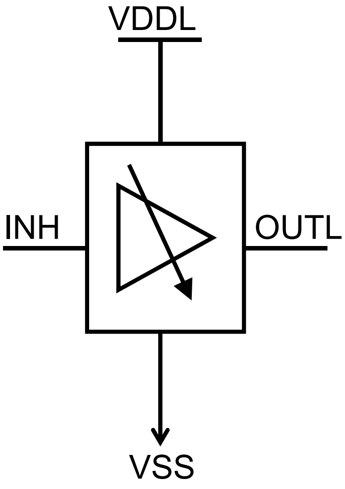
Ref:[2]

Heterogeneous Integration

High-to-Low Level Shifter

- **CMOS gate can be driven higher than the power supply voltage without problems, up to the gate breakdown voltage**
  - Additional level shifter is not required



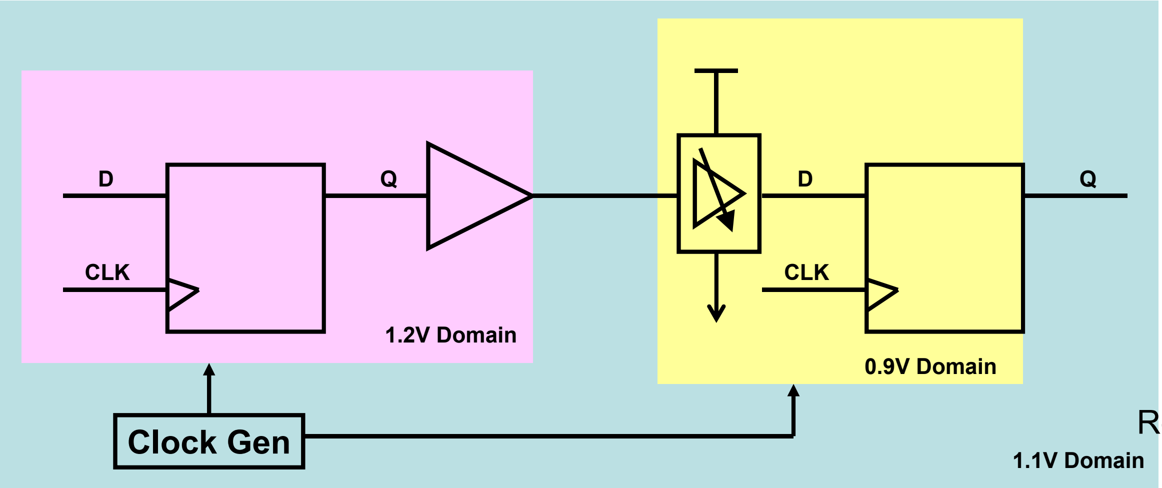


Ref:[2]

Heterogeneous Integration

Timing Issues

- **Gate delay is different with different supply voltages**
  - Clock arrival time may be quite different, too
  - Level shifters will induce extra delay as well
  - Timing analysis becomes more complex
- **Optimization with accurate timing analysis**



Ref:[2]

Other Issues in MSV Approach

- **Level shifter design**
  - Reduce its power, area, and delay
- **Library characterization**
  - Support every possible supply voltages in a library requires too much characterization efforts
  - When a cell is running at different voltage that are not characterized in the library, timing analysis becomes much more complex
- **Floorplanning, power planning, power grids**
  - Voltage-island awareness is a new consideration in physical design
  - Reduce the required level shifters is an additional target
- **Power up and power down sequencing**
  - A pre-designed sequence for powering up the design may be required to avoid deadlock

Outline

- Introduction
- Multiple Power Domains
- Dynamic Voltage and Frequency Scaling
- Clock Gating
- Dual Threshold Voltages
- Power Gating
- Conclusion

Dynamic Voltage & Frequency Scaling

- **Dynamically control the voltage and frequency**
  - Increase voltage when high performance is required
  - Reduce the voltage when not needed

- **DVFS is a method to reduce the energy for a task by scaling the operating voltage/frequency**
  - Power consumption if a CMOS-based circuit is

$$P = \alpha \bullet C \bullet V^2 \bullet f$$

α : switching factor
C : effective capacitance
V : operating voltage
f : operating frequency

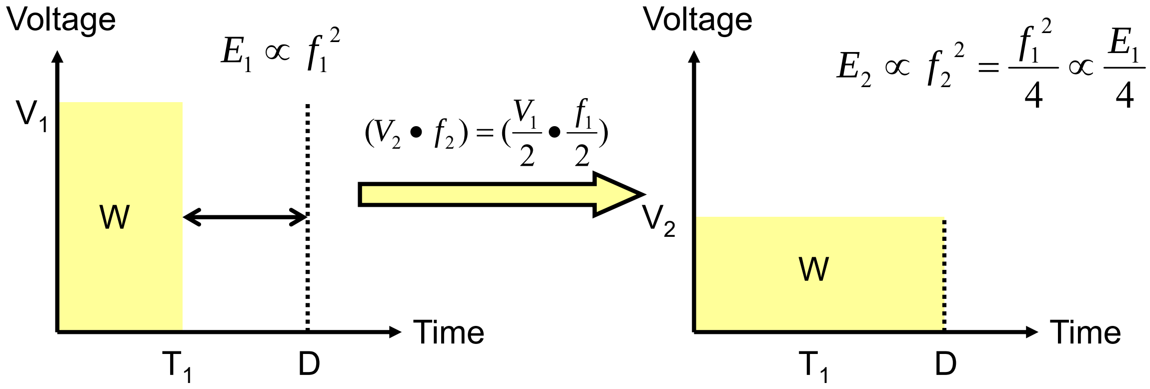
  - Energy required to run a task during T is

$$E = P \bullet T \propto V^2 \qquad \textit{Assuming } V \propto f,\ T \propto f^{-1}$$

- **By lowering CPU frequency, CPU energy can be reduced**

Energy Reduction with DVFS

- **Ex: a task with workload W should be completed by a deadline, D**



Voltage
$E_1 \propto f_1^{\,2}$
V_1
W
T_1
D
Time
$(V_2 \bullet f_2) = (\frac{V_1}{2} \bullet \frac{f_1}{2})$
Voltage
$E_2 \propto f_2^{\,2} = \frac{f_1^{\,2}}{4} \propto \frac{E_1}{4}$
V_2
W
T_1
D
Time

- **DVFS reduces the CPU energy by providing “just enough” computation power**

Choosing a Frequency in DVFS

- **Workload of a task, $\mathbf{W_{task}}$ is defined as the total number of CPU clock cycles required to finish the task.**

$$W_{task} = \sum_{i=1}^{N} CPI_i$$

N : total number of instructions in a task
CPI : clock cycles per instruction

- **Task execution time, $\mathbf{T_{task}}$ is a function of the CPU frequency, $\mathbf{f_{CPU}}$**

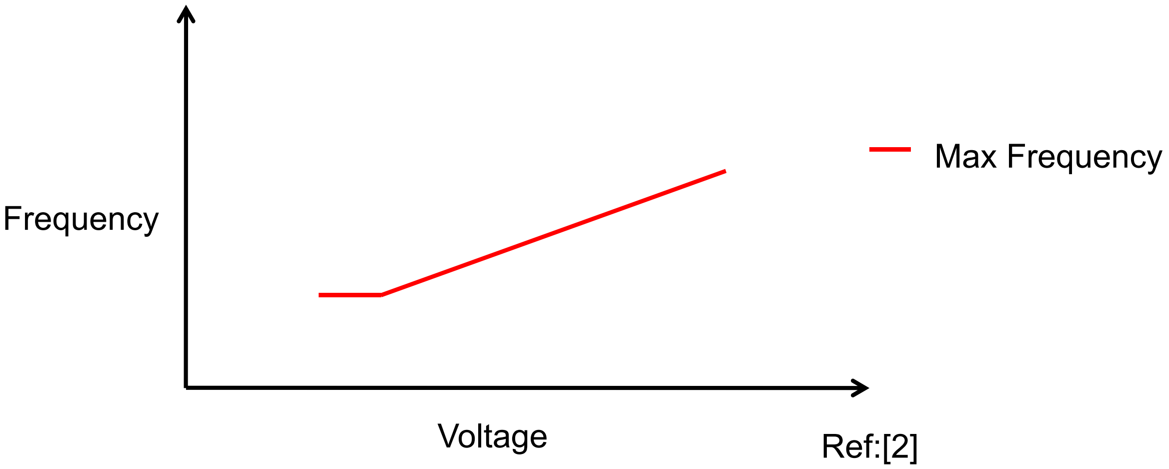
$$T_{task} = \frac{W_{task}}{f_{CPU}}$$

- **Given a deadline of D, $\mathbf{f_{target}}$ denotes the CPU frequency that results in $\mathbf{T_{task}}$ closest to D**

$$f_{target} = \frac{W_{task}}{D} \Rightarrow T_{task} = D$$

Dynamic Operating Voltage Range

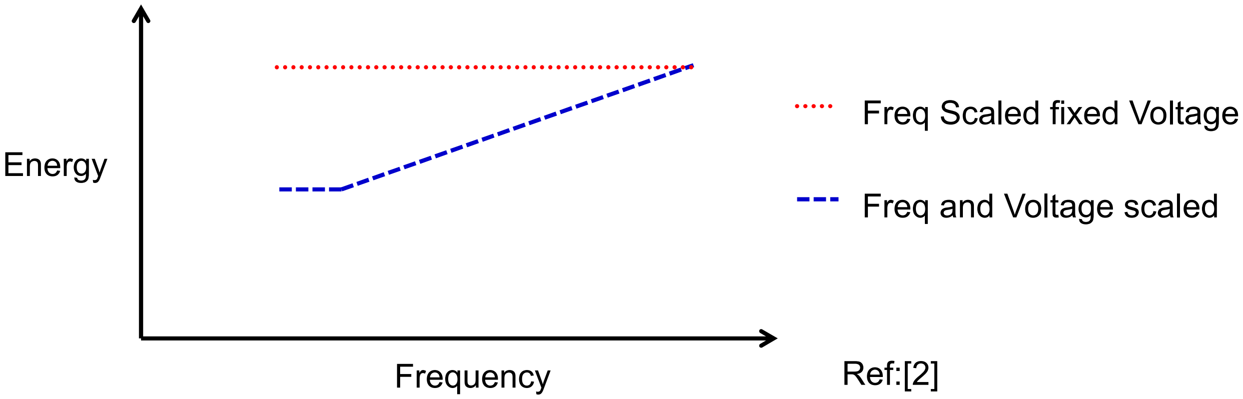
- **Best operation region is the period that frequency increases monotonically over voltage**
  - Make sure the circuits can operate reliably and the delay paths vary monotonically



Ref:[2]

Dynamic Energy Dissipation

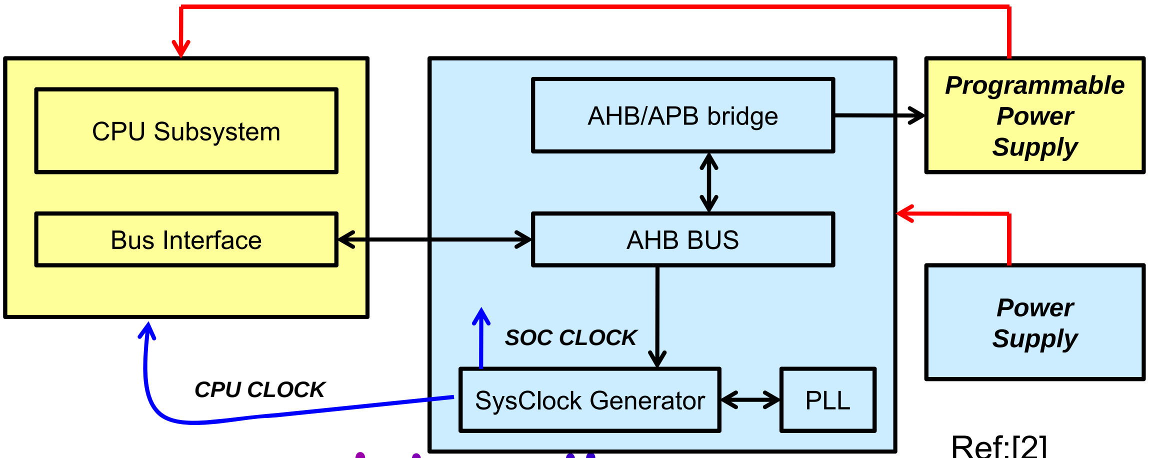
- **Energy is the integration of power over the time taken to complete a task of work**
  - Lowering frequency can reduce power, but total energy is the same
- **Lowering the supply voltage can reduce the energy dissipation simultaneously**



Ref:[2]

DVFS Block Diagram

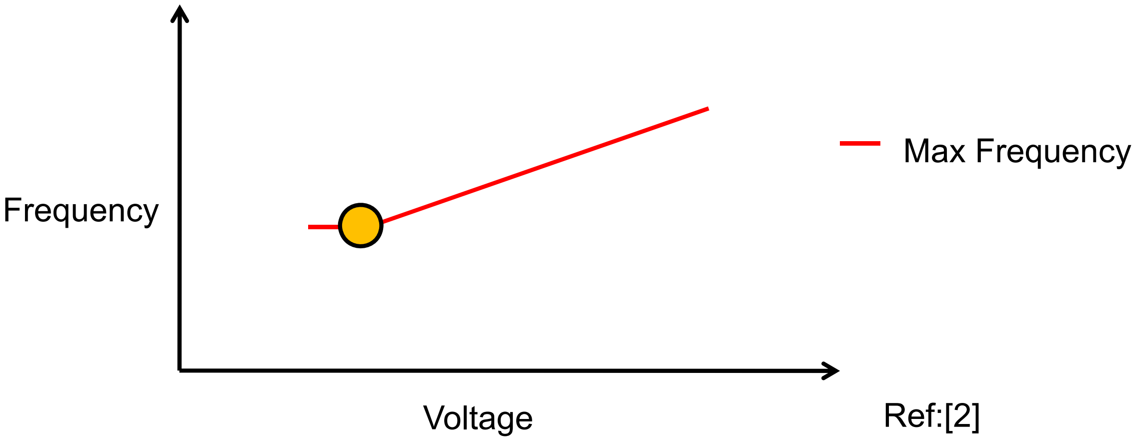
- **Supply Voltage**
  - The CPU subsystem is powered by a programmable power supply.
  - The rest of the chip is powered by fixed power supply
- **System Clock**
  - The SysClock Generator dynamically control the PLL to generate required clock signals with different frequency



Ref:[2]

Minimum Operating Point

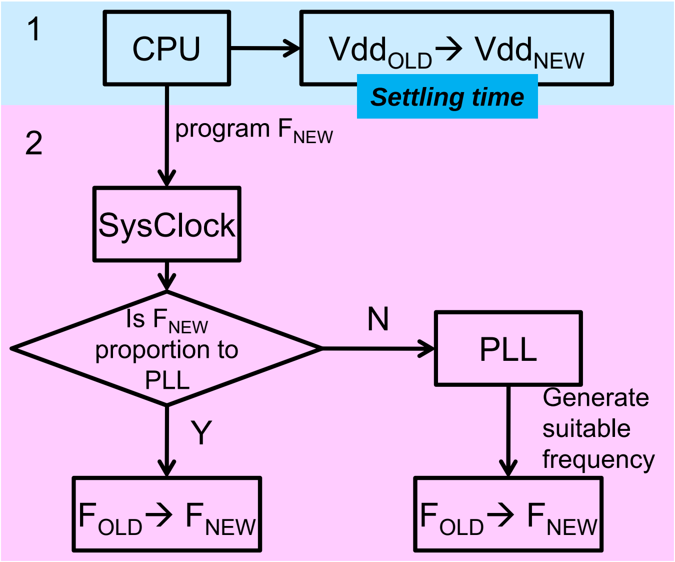
- **Decide the minimum CPU clock speed that meets the workload requirements**
- **Decide the lowest supply voltage that will support that clock speed**



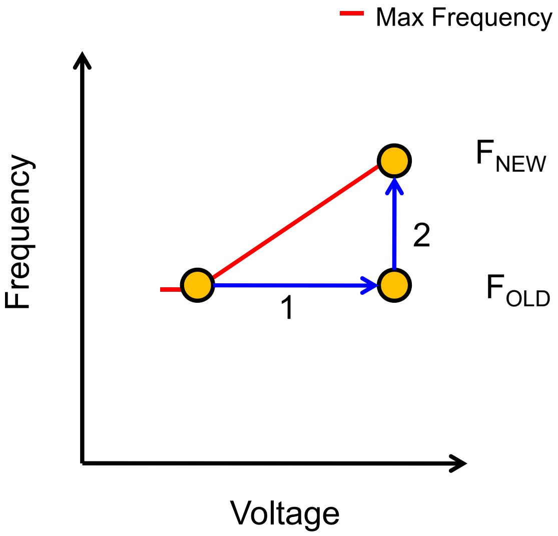
Ref:[2]

DVFS Operation ($F_{NEW} > F_{OLD}$)

- **The target clock is higher than current clock**
  - Supply voltage increases
  - Clock frequency increases



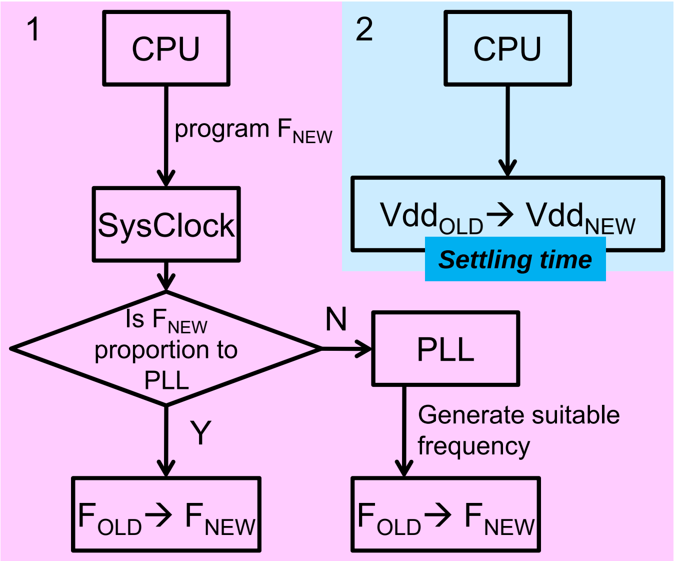
1
CPU
$Vdd_{OLD} \rightarrow Vdd_{NEW}$
Settling time
2
program F_{NEW}
SysClock
Is F_{NEW} proportion to PLL
N
PLL
Generate suitable frequency
Y
$F_{OLD} \rightarrow F_{NEW}$
$F_{OLD} \rightarrow F_{NEW}$



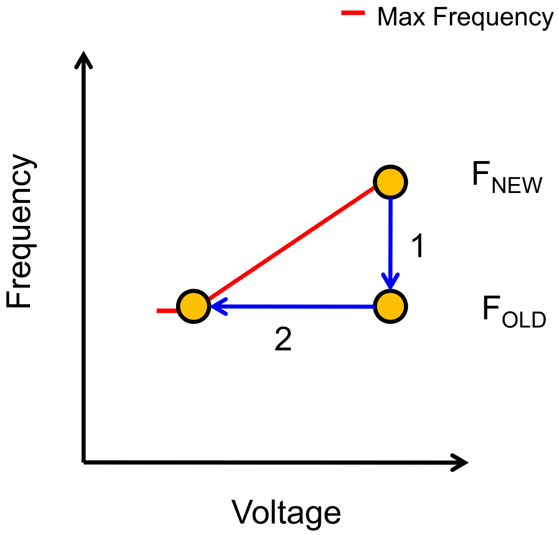
Heterogeneous Integration

DVFS Operation ($F_{NEW} < F_{OLD}$)

- **The target clock is lower than current clock.**
  - Clock frequency decreases
  - Supply voltage decreases



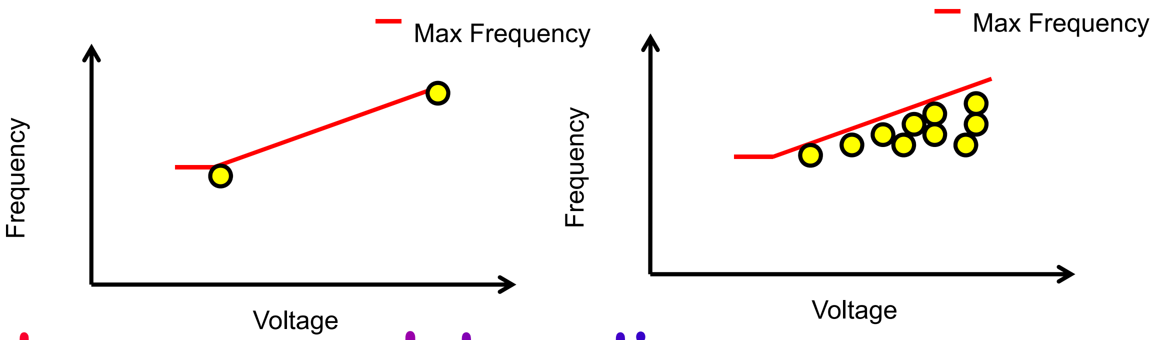
1
CPU
program F_{NEW}
SysClock
Is F_{NEW} proportion to PLL
N
PLL
Generate suitable frequency
Y
$F_{OLD} \rightarrow F_{NEW}$
$F_{OLD} \rightarrow F_{NEW}$
2
CPU
$Vdd_{OLD} \rightarrow Vdd_{NEW}$
Settling time



Heterogeneous Integration

DVFS Implementation Issues

- **Determining which voltages and clock values to support**
  - Too few operating points
    - Spend a significant time ramping between two levels
    - The energy saving efficiency during the ramping times are less than the steady-state values
  - Too many operating points
    - Power supply will spend most of time “hunting” between different target voltage levels



Clock Speed/Supply Voltage Values

- **Rules to determine the number of operating points**
  - What are the appropriate clock frequencies for different workloads
  - Which frequencies have clock periods that are multiples of the PLL period
    - Just change the clock divider, not the PLL frequency
  - What voltage is required to support each target frequency

- **Refine the selection of operating points**
  - An FPGA implementation
  - High level simulation model

Other DVFS Issues

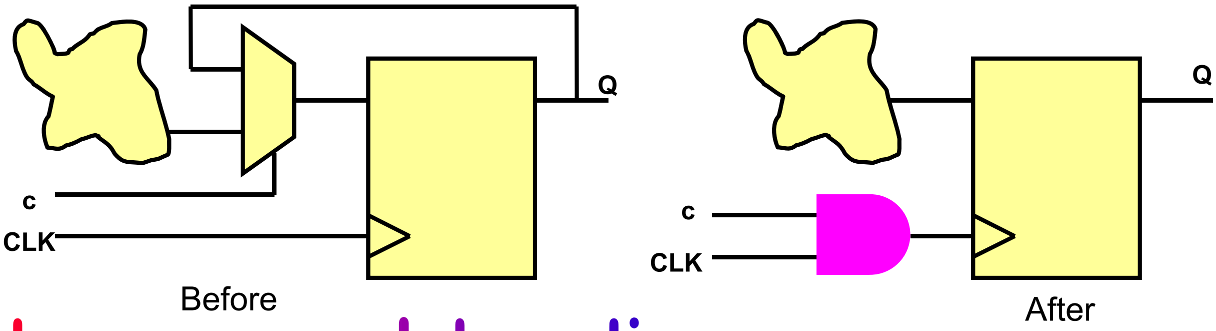
- Determining the minimum voltage to meet a particular (sub)system performance level
- Achieving timing closure over a range of voltages and clock speeds
- Control sequencing (power management)
- Generate required clock speed/supply voltage values
- Verification across different situations
  - Especially at transition cases

Outline

- Introduction
- Multiple Power Domains
- Dynamic Voltage and Frequency Scaling
- Clock Gating
- Dual Threshold Voltages
- Power Gating
- Conclusion

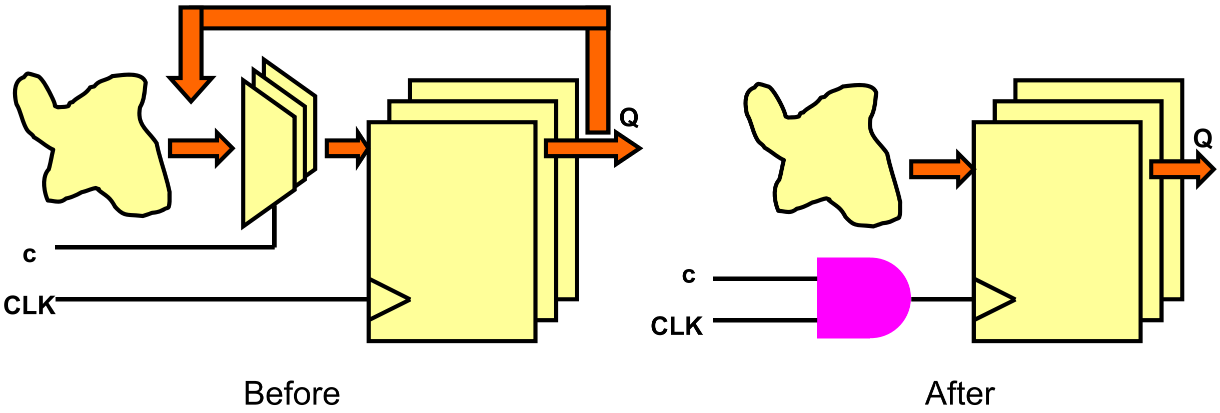
Clock Gating (1/2)

- **Up to 50% or even more of the dynamic power in a chip is in the distribution network of the clock**
  - Clock buffers have the highest toggle rate in the system
  - Typically there are lots of clock in a design
  - Clock buffers often have a high driving strength to minimize clock delay
- **The most common technique to reduce this power is to turn clocks off when they are not required → Clock Gating**
  - No need for Muxes to re-circulate the data for these flip-flops



Clock Gating (2/2)

- **Much more effective if the same gating function is applied to large set of registers**

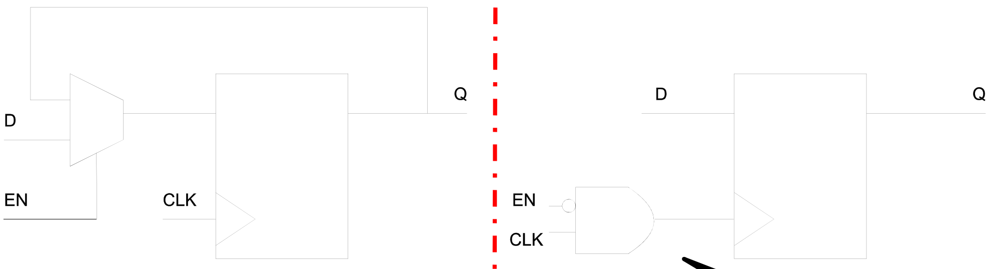


How Clock Gating Works ?

- In the original RTL, the register is updated or not depending on a variable (EN)
- The same result can be achieved by gating the clock based on the same variable

Typical Compile

Compile with Clock Gating Insertion



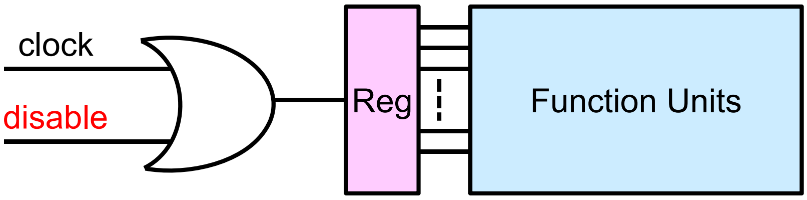
```
always@(posedge CLK)
   if(EN)
      Q <= D;
```

Low activity

Ref:[2]

Issues of Clock Gating

- Clock gating also idle the succeeding functional units
- Need logic to generate ***disable*** signal
  - Increase the complexity of control logic
  - Timing critical to avoid glitches at OR gate output
- Additional gate delay on clock signal
  - Gated OR gate can replace a buffer in the clock distribution tree
- May generate an extra clock edge while ***disable*** is deasserted
  - (clock=0, disable=1) → 1 ; (clock=0, disable=0) → 0
  - Should be careful about the deassertion time

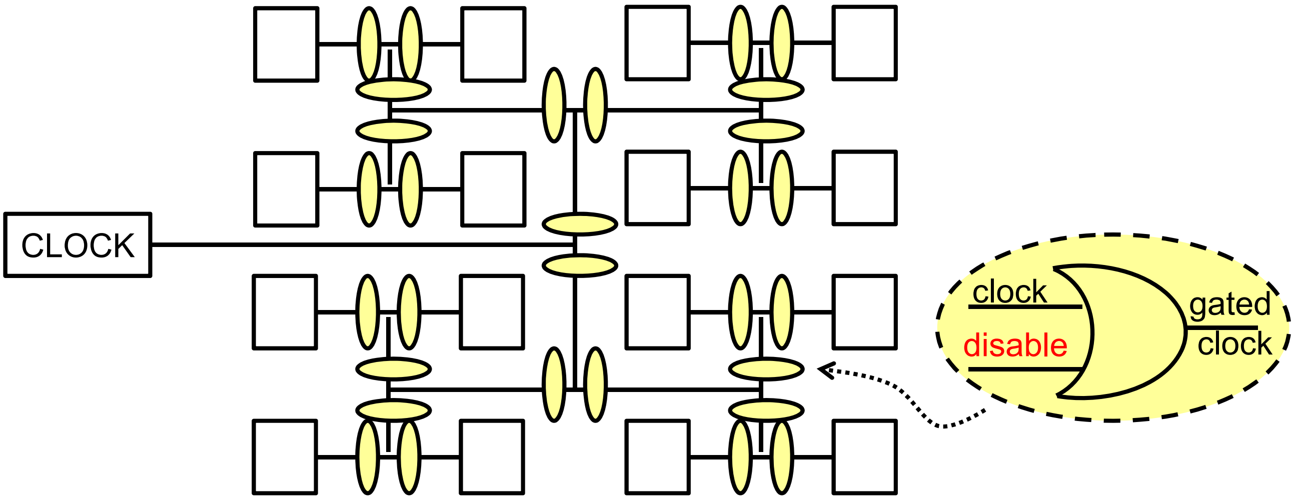


Clock Gating Levels

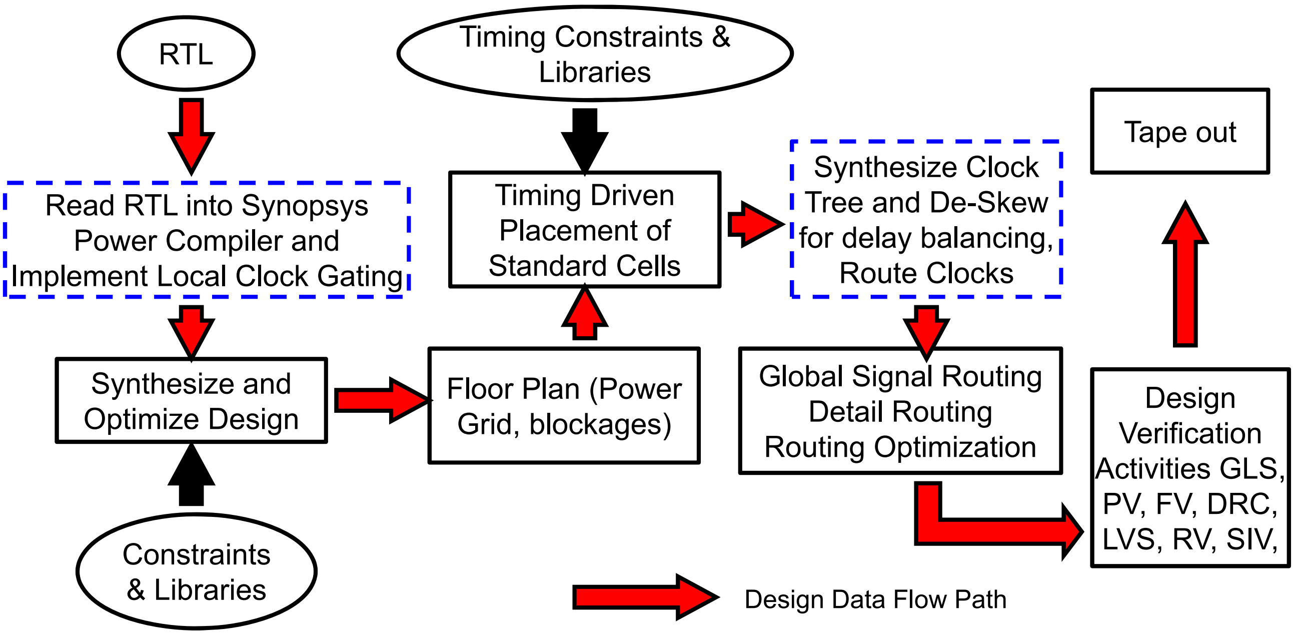
- **Fine Grain**
  - Portions of the pipeline registers are disabled depending on whether the information they hold is used in the next stage
- **Medium Grain**
  - Disable cache pre-charging during cache miss
- **Coarse Grain**
  - Eliminate switching of the clock's main driver
- **Depending on the target applications**

Gated Clock Distribution

- **If the paths are perfectly balanced, clock skew is zero**
  - Can insert clock gating at multiple levels in clock tree
  - Can shut off entire sub-tree if all gating conditions are satisfied



Clock Gating in the Design Flow



RTL

Timing Constraints & Libraries

Read RTL into Synopsys Power Compiler and Implement Local Clock Gating

Timing Driven Placement of Standard Cells

Synthesize Clock Tree and De-Skew for delay balancing, Route Clocks

Tape out

Synthesize and Optimize Design

Floor Plan (Power Grid, blockages)

Global Signal Routing Detail Routing Routing Optimization

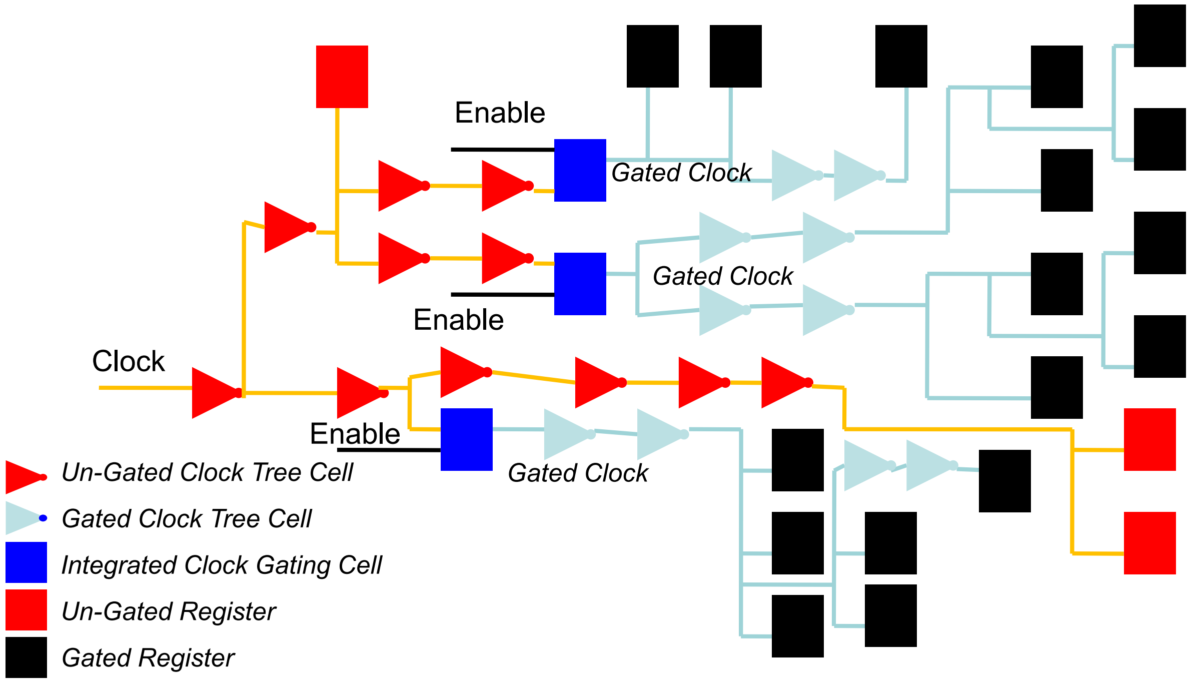
Design Verification Activities GLS, PV, FV, DRC, LVS, RV, SIV,

Constraints & Libraries

Design Data Flow Path

Ref:[3]

Clock Gating Example



Enable

Gated Clock

Gated Clock

Enable

Clock

Enable

Gated Clock

Un-Gated Clock Tree Cell

Gated Clock Tree Cell

Integrated Clock Gating Cell

Un-Gated Register

Gated Register

Ref:[3]

Other Clock Gating Issues

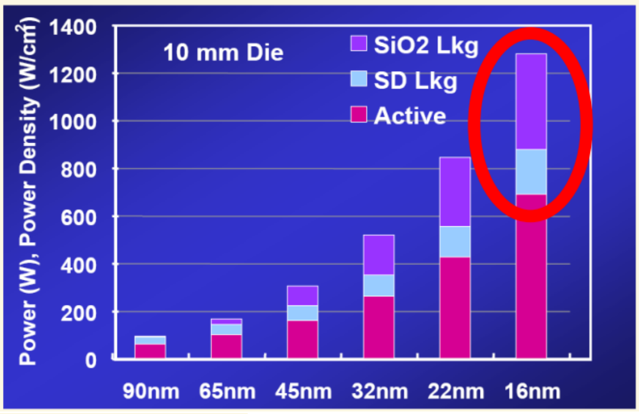
- **Where and when can we gate the clock?**
  - Find suitable gating functions for many latches
- **Need careful checking to prevent the function being changed**
  - Formal equivalence checking may be a solution
  - Scalability of sequential equivalence checking is a problem
- **Do we really reduce power?**
  - Need better power estimation capabilities to consider clock gating effects

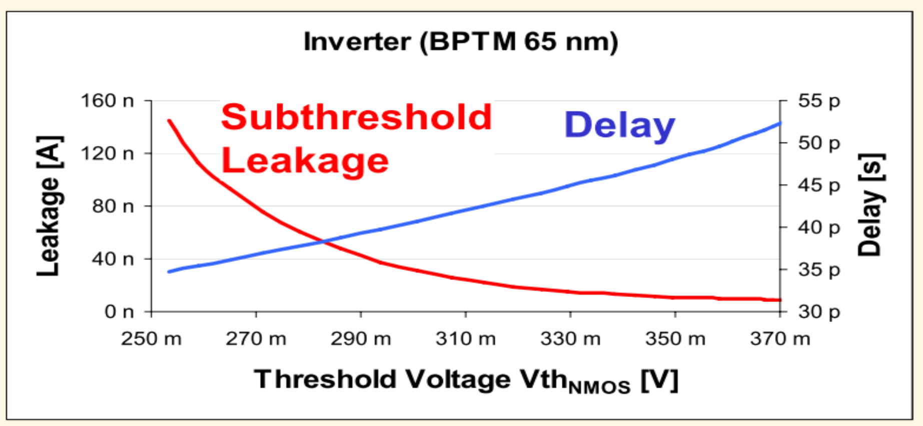
Outline

- **Introduction**
- **Multiple Power Domains**
- **Dynamic Voltage and Frequency Scaling**
- **Clock Gating**
- **Dual Threshold Voltages**
- **Power Gating**
- **Conclusion**

Leakage Power Reduction

- **Leakage power dissipation has become a major part in advance process**
- **Reducing Vt can reduce the leakage current**
  - Also reducing performance



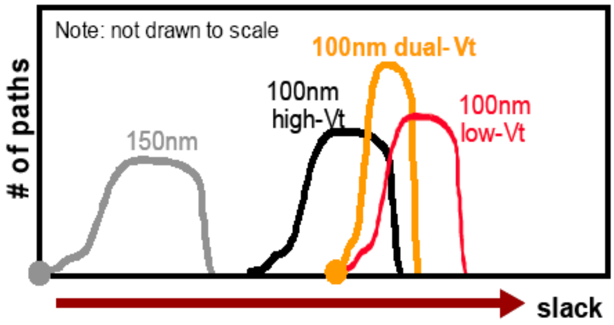


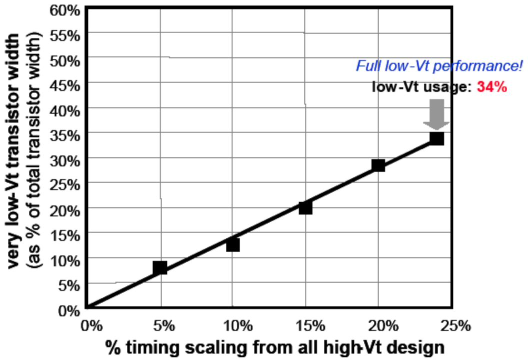
SiO2 Lkg : Gate Oxide Tunneling Leakage
SD Lkg : Sub-threshold Leakage

Ref: [4]

Dual-VT Design for Leakage Control

- **"Dual VT" is a technique to minimize the total number of fast, leaky low-VT cells by deploying them only when required to meet timing**
- **Dual-VT can increase timing slack and also can reduce the leakage power**

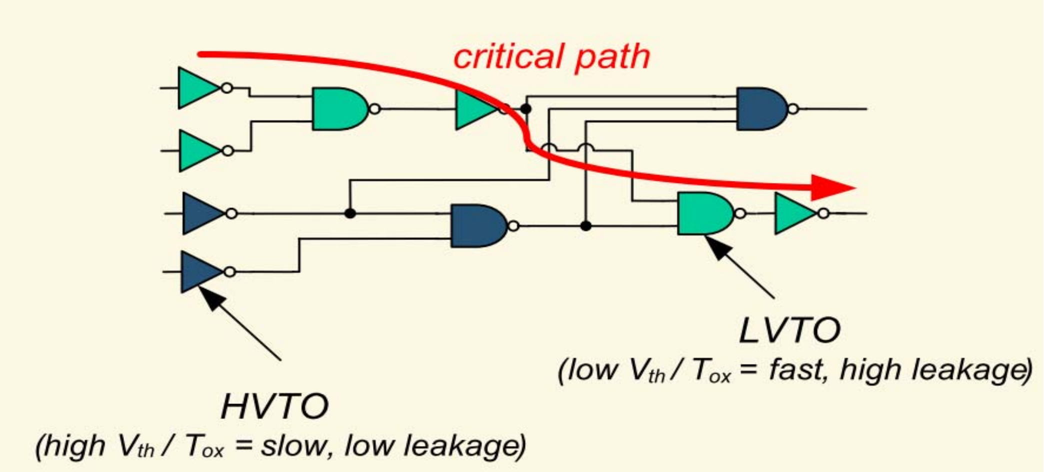




Dual-Threshold Voltage (Dual-VT)

- **Dual Threshold Voltage CMOS (DVTCMOS)**
  - Use lower threshold for devices within the critical paths
  - Use higher threshold for devices outside the critical paths

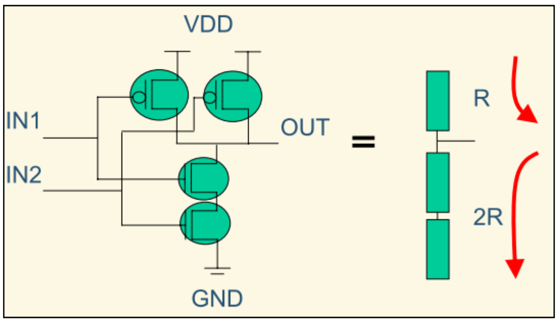
➔ **Decrease leakage power without performance penalty**


critical path
LVTO
(low V_{th} / T_{ox} = fast, high leakage)
HVTO
(high V_{th} / T_{ox} = slow, low leakage)

Ref: [4]

Mixed-Vt Gates

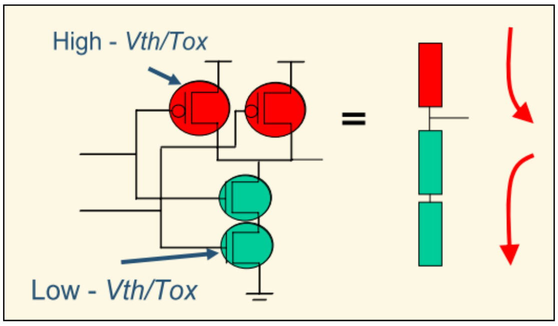
- **Preserve the delay while decreasing the leakage**



0→1 delay
(Output from GND to VDD)

1→0 delay
(Output from VDD to GND)

➔**delay$_{0\to1}$ < delay$_{1\to0}$**

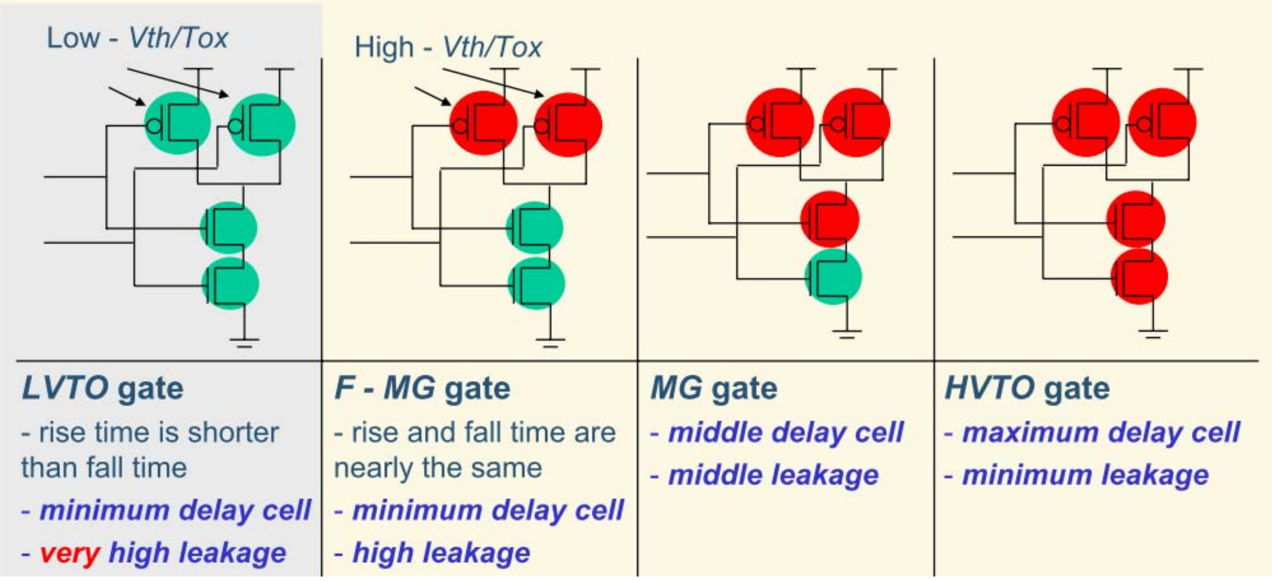


Typical timing analysis
➔ only maximum delay is considered!!!!

Slow down the rising case
➔**delay$_{0\to1}$ = delay$_{1\to0}$**

Ref: [4]

Example : Mixed-Vt Gates (NAND2)



LVTO gate
- rise time is shorter than fall time
- ***minimum delay cell***
- ***very high leakage***

F - MG gate
- rise and fall time are nearly the same
- ***minimum delay cell***
- ***high leakage***

MG gate
- ***middle delay cell***
- ***middle leakage***

HVTO gate
- ***maximum delay cell***
- ***minimum leakage***

Ref: [4]

Summary of Dual-Vt CMOS

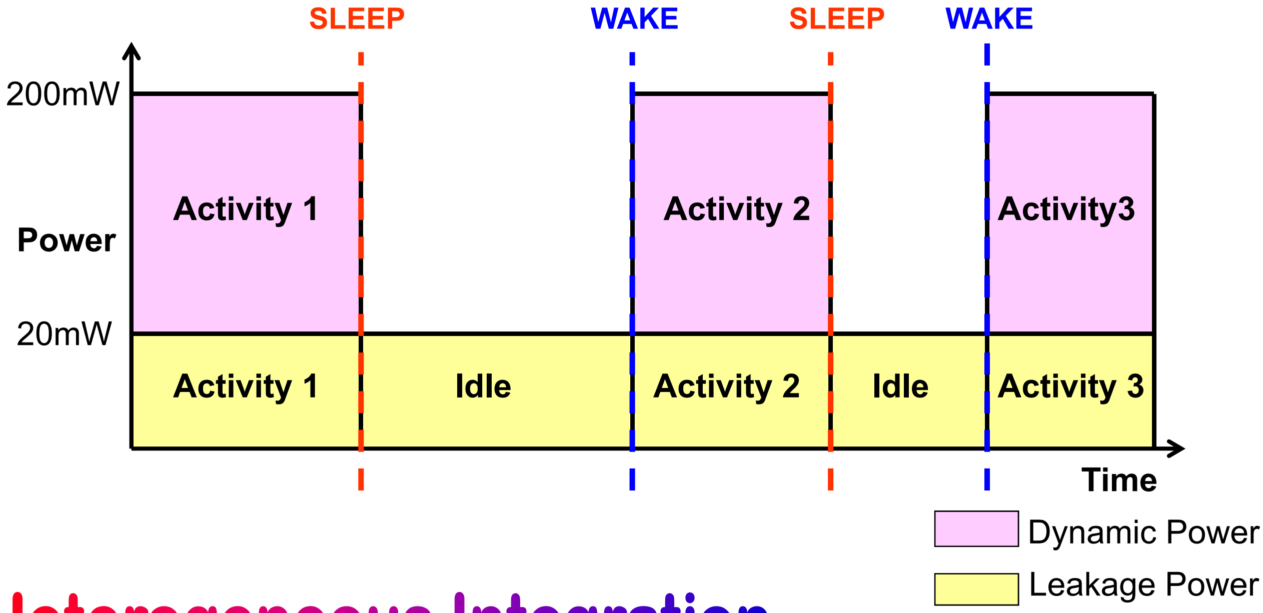
- **High-Vt cell VS low-Vt cell**
  - High-Vt cell has lower leakage current and lower performance
  - Low-Vt cell has higher performance and higher leakage current
- **Usually, there is a minimum performance which be met before optimizing power**
  - First, synthesizing with the highest performance, using low-Vt library
  - Then, swapping the cells on non-critical paths to low-leakage cells because their performance degradation will not affect the overall speed
- **Library supports for different Vt is an extra cost**
  - Similar to preparing two different library sets
- **Mixed-Vt gates combines the advantages of DVTCMOS at transistor level**
  - The cell library with Mixed-Vt gates is not popular yet

Outline

- Introduction
- Multiple Power Domains
- Dynamic Voltage and Frequency Scaling
- Clock Gating
- Dual Threshold Voltages
- Power Gating
- Conclusion

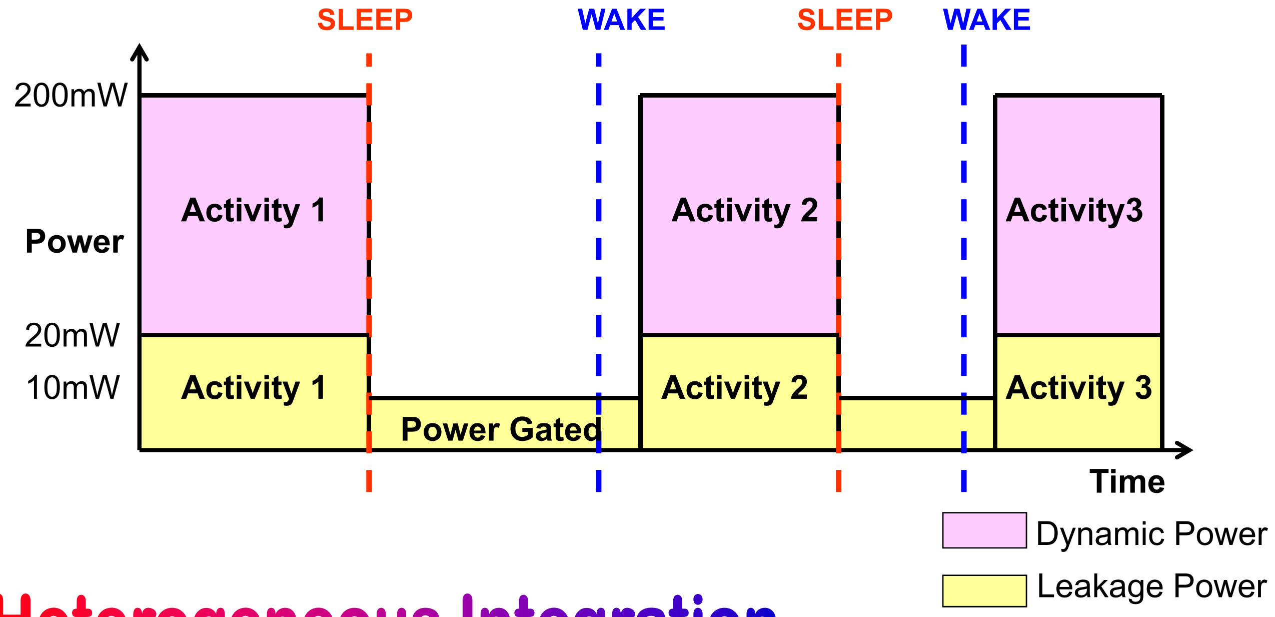
Power Profiles (No Power Gating)

- SLEEP events initiate entry to the low power mode
- WAKE events initiate entry to active mode



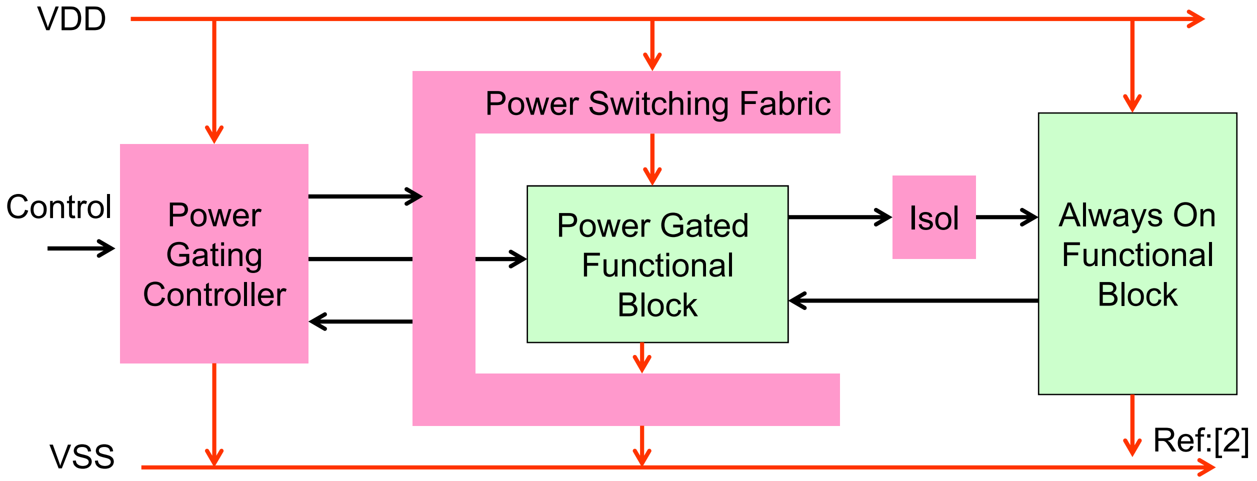
Power Profiles (Power Gating)

- **Power gating further reduces the standby power**
- **Require longer wakeup time in the WAKE event**



Block Diagram of Power Gating Design

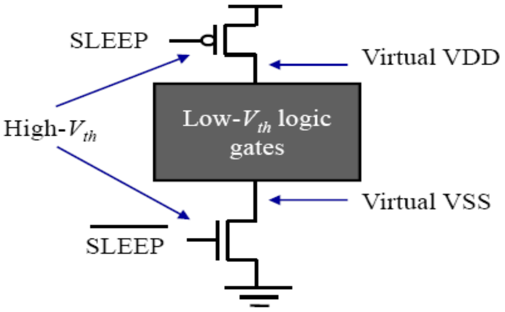
- **The basic components are composed with**
  - Power switching fabric
  - Isolation cell
  - Power gating controller

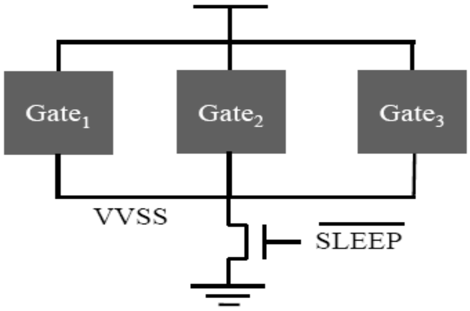


Ref:[2]

Multi-Threshold-Voltage CMOS (MTCMOS)

- **High-Vth power switches are controlled by SLEEP signal**
  - Low-Vth logic gates are used to achieves high performance
  - Reduces leakage power dramatically due to the series-connected high-Vth power switch
- **Typically only a header or a footer sleep transistor is used.**
- **A single sleep transistor may be shared among several logic gates**





Heterogeneous Integration

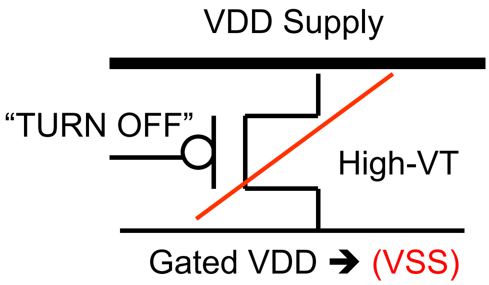
Sleeper Transistor (MTCMOS)

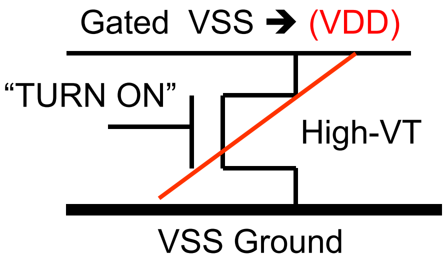
- **Header**

  The internal nodes and outputs of a power gated block collapse down to the ground rail when the switch is turned off

- **Footer**

  The internal nodes and outputs all charge to the supply rail when the switch is turned off

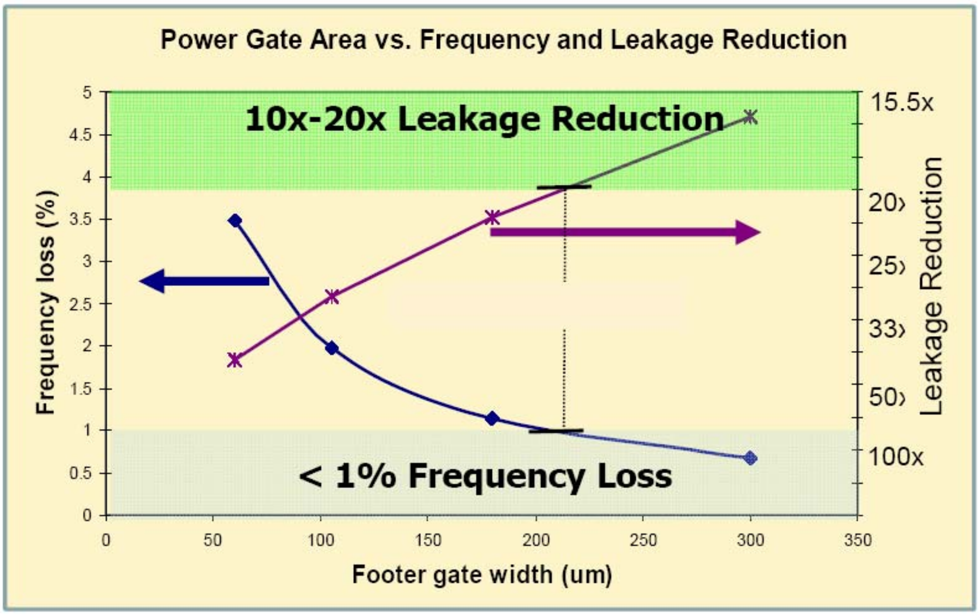




Heterogeneous Integration

Tradeoff on Sleep Transistor Sizing

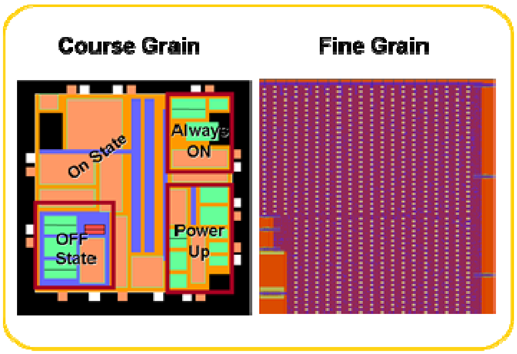
- **Maximize the saving of power consumption**
- **Minimize the impact of performance**

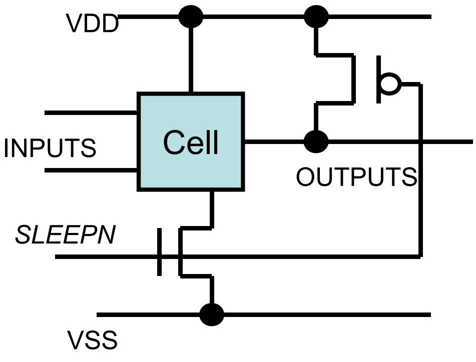


A 2-stage pipelined 40-bit ALU (IBM)

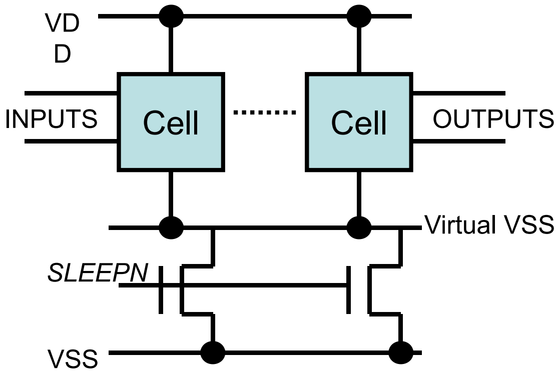
Granularity of Power Switches

- **Fine Grain**
  - The switch is placed inside each cell
  - Better control, but larger area overhead
- **Coarse Grain**
  - Connected on the permanent power
  - Easier to implement, but have performance penalty during power-up





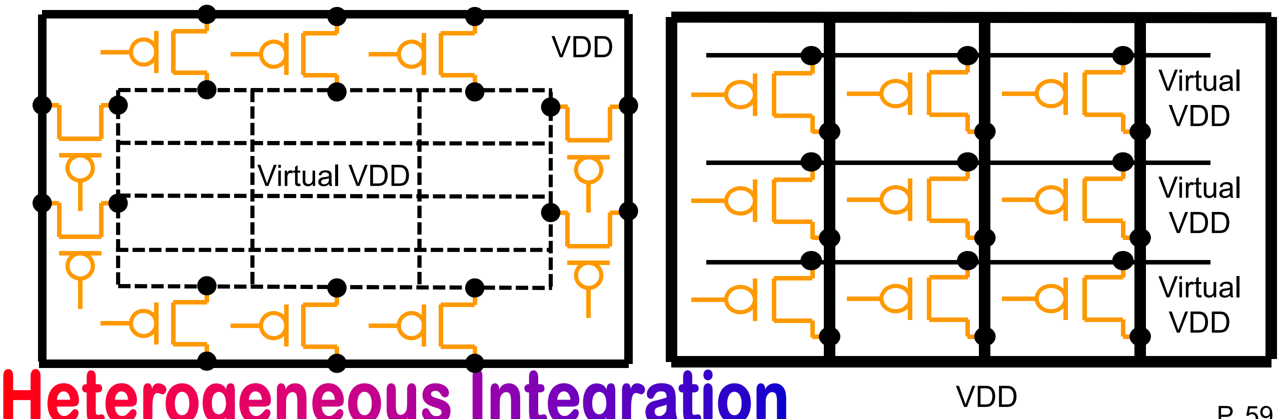
Fine Grain



Coarse Grain

Arrangement of Power Switches

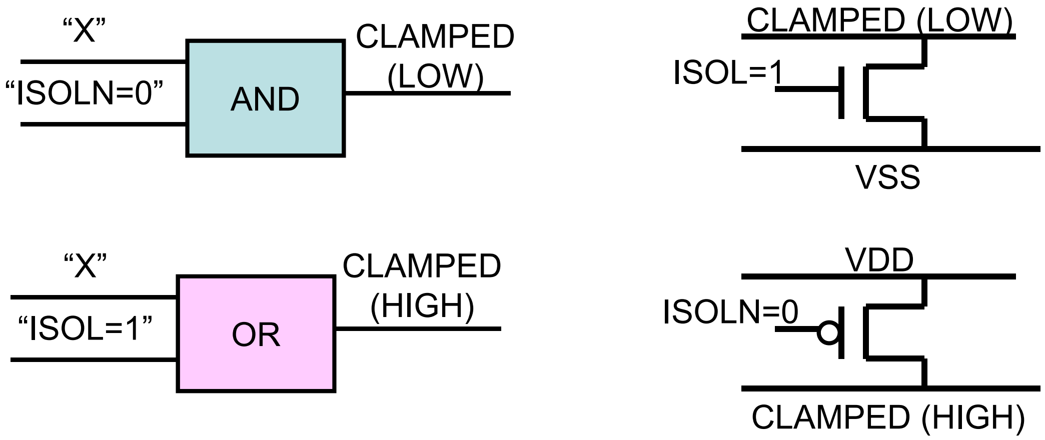
- **Ring Style**
  - A ring of switches connects VDD to a switched or virtual VDD power mesh that covers the power gated block
  - Simpler power plan, but larger area overhead
- **Grid Style**
  - The sleep transistors are distributed throughout the power gated region.
  - Less area and performance overhead, but increase routing complexity



Heterogeneous Integration

Signal Isolation

- **Without power supply, the output signals will become floating**
  - Isolation cells are required to stop the propagation of those signals
- **For timing consideration, the isolation transistor can be used to clamp the output signal**



Heterogeneous Integration

State Retention and Restoration

- **Before shutting down the power supply, the register states must be kept for future restoration**

- **To resume the operation of the gated blocks**
  - The retention state must be restored when the block is powered up

- **State retention and restoration strategies**
  - A software approach that reads and writes registers
  - A scan-based approach that uses scan chains to store states off chip
  - A register-based approach that uses built-in retention registers

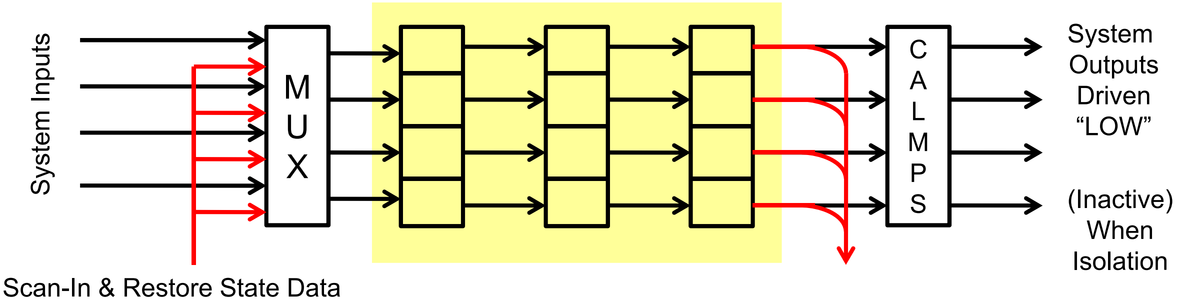
Software Approach

- **Software approach flow**
  - During the power shutdown sequence, a processor reads the registers of the power gated blocks
  - The states are stored in the processor's memory
  - During the power up sequence, the processor reads its memory and write the states back into the power gated blocks

- **Drawbacks**
  - The bus traffic slows the power down and up sequence
  - Bus conflicts can make the store/restore times no-deterministic
  - Software must be written and integrated into the system's software for handling power down and up

Scan-Based Approach

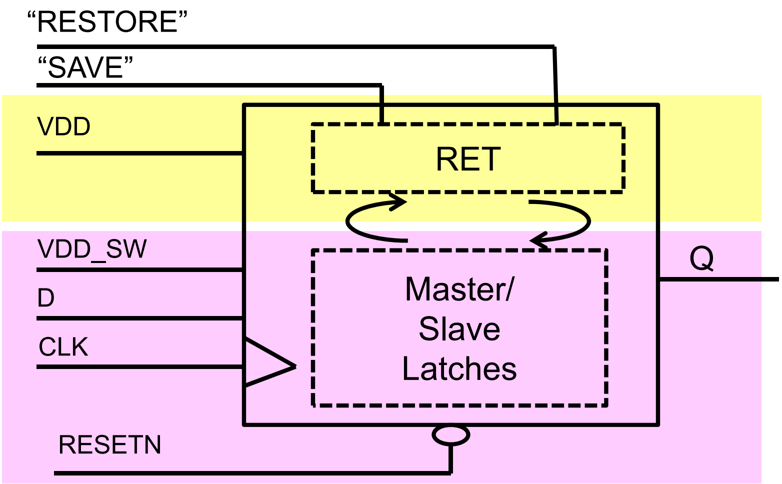
- **Reusing scan chains for manufacturing test to perform state retention has no area overhead**
- **Scan-based approach flow**
  - During power down, the registers are shifted out as in scan testing mode and stored into the memory
  - During power up, the memory states are shifted in by scan chains
- **Shifting the register states out and back is extra energy cost**


System Inputs
M U X
C A L M P S
System Outputs Driven "LOW"
(Inactive) When Isolation
Scan-In & Restore State Data

Heterogeneous Integration

Register-Based Approach

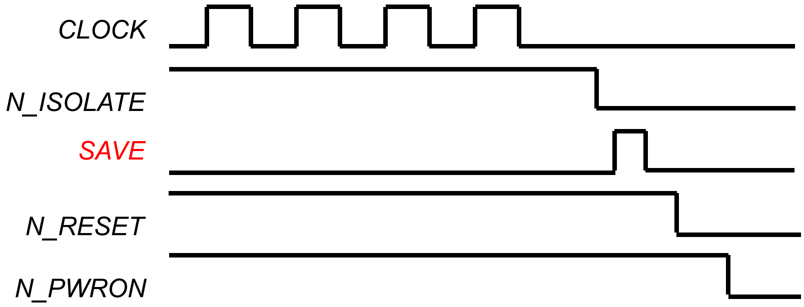
- **Build in a "shadow" register to preserve the register states during power down and restore it at power up**
  - The shadow register is always powered on for state retention
- **Easier for implementation, but have larger area overhead**



Heterogeneous Integration

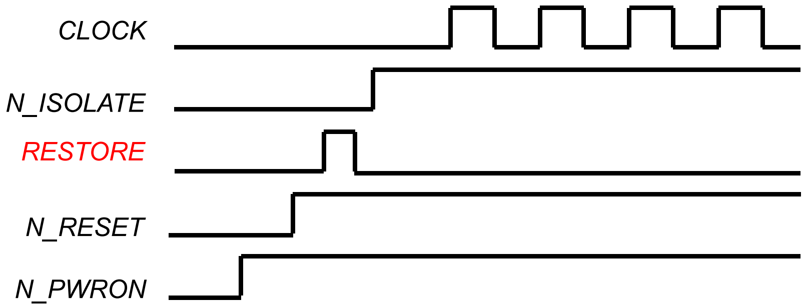
Power Control Sequence (1/2)

- **To shut down power**
  - Flush through any bus or external operations in progress
  - Stop the clocks in the appropriate phase to minimize leakage into the power-gated region
  - Assert the isolation control signal to park all outputs in a safe conditions
  - Assert the state retention save condition
  - Assert reset to the block, so that it powers up in the reset condition
  - Assert the power gating control signal to power down the block



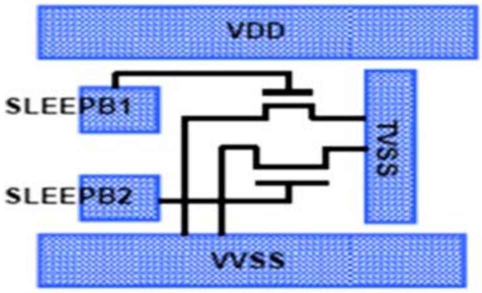
Power Control Sequence (2/2)

- **To restore power**
  - De-assert the power gating control signal to restore power back
  - De-assert reset to ensure clean initialization following the gated power-up
  - Assert the state retention restore condition
  - De-assert the isolation control signal to restore all outputs
  - Restart the clocks



Power-On Rush Current

- **There is a large current rush in sleep-to-active transition which can cause EM and IR-drop issues**
  - Also called “in-rush current”
- **Using parallel (mother-daughter) sleep transistors can reduce rush current**
  - Optimizing the ratio of the daughter and mother transistor widths
  - Scheduling the turn-on times of the two switches so as to minimize the wakeup delay



- **Power-up scheduling is another useful technique**
  - Scheduling the turn-on times of different blocks to avoid current rush and reduce the peak current

Summary of Power Gating Approach

- **Power gating design can reduce the leakage power when the block is powered off**
  - Using High-VT transistors to limit the leakage current
- **Power gating design is composed of**
  - Power switch fabric
  - Isolation and retention cell
  - Power controller
- **Performance degradation should be prevented**
  - Performance becomes worse because IR-drop effects on normal operation mode
  - In-rush current during power-up can cause extra supply noise issues

Conclusion

- **Low-power design is a major trend on today’s applications**
  - Portable devices, medical devices, bio-electronics, …
  - Low-power techniques at higher design levels can reduce more power
- **Low-power techniques mentioned in this slide**
  - Reduce supply voltage (Vdd↓)
    - Multiple power domains
  - Reduce signal transitions (f↓)
    - Dynamic voltage and frequency scaling (DVFS)
    - Clock gating
  - Use high-Vt cells for lower leakage power (leakage↓)
    - Dual threshold voltages, Mixed-Vt gates
  - Shut down power supply when not used (Vdd↓, leakage↓)
    - Power gating (MTCMOS)
- **Power management is important to control those mechanisms**

Reference

- **[1] Anand Raghunathan, Niraj K. Jha, Sujit Dey, “High-level power analysis and optimization” Boston : Kluwer Academic, c1998.**
- **[2] Michael Keating, David Flynn, Rob Aitken, Alan Gibbons, Kaijian Shi, “Low Power Methodology Manual: For System-on-Chip Design “ Berlin, Germany : Spring-Verlag, 2007.**
- **[3] Pokhrel, K., “Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study”,SNUG, 2007.**
- **[4]Frank Sill, Claas Cornelius, Stephan Kubisch, Dirk Timmermann, “Mixed Gates: Leakage Reduction techniques applied to Switches for Networks-on-Chip”, ReCoSoC, 2006.**
- **[5]Karen Yorav, “The challenges of low power design”, Tutorial , Haifa Verification Conference 2008.**