# CSE477
# VLSI Digital Circuits
# Fall 2002

## Lecture 26: Low Power Techniques in Microarchitectures and Memories

Mary Jane Irwin ( www.cse.psu.edu/~mji )
www.cse.psu.edu/~cg477

[Adapted from Rabaey's *Digital Integrated Circuits*, ©2002, J. Rabaey et al.]



## Review: Energy & Power Equations

$$E = C_L V_{DD}^2 P_{0\to1} + t_{sc} V_{DD} I_{peak} P_{0\to1} + V_{DD} I_{leakage}$$

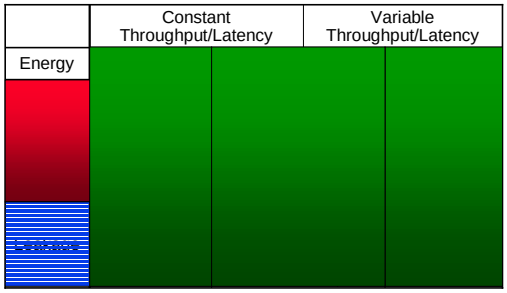
$$f_{0\to1} = P_{0\to1} * f_{clock}$$

$$P = C_L V_{DD}^2 f_{0\to1} + t_{sc} V_{DD} I_{peak} f_{0\to1} + V_{DD} I_{leakage}$$

| Dynamic power (~90% today and decreasing relatively) | Short-circuit power (~8% today and decreasing absolutely) | Leakage power (~2% today and increasing) |
|---|---|---|

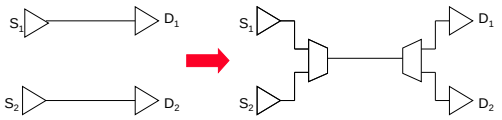


## Power and Energy Design Space



## Bus Multiplexing

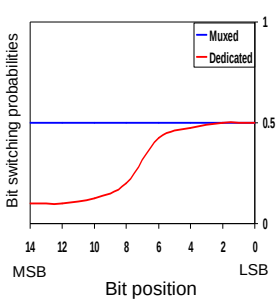
- Buses are a significant source of power dissipation due to high switching activities and large capacitive loading
  - 15% of total power in Alpha 21064
  - 30% of total power in Intel 80386
- Share long data buses with time multiplexing ($S_1$ uses even cycles, $S_2$ odd)




- But what if data samples are correlated (e.g., sign bits)?



## Correlated Data Streams

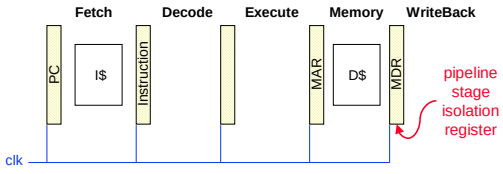



- For a shared (multiplexed) bus advantages of data correlation are lost (bus carries samples from two uncorrelated data streams)
  - Bus sharing should not be used for positively correlated data streams
  - Bus sharing may prove advantageous in a negatively correlated data stream (where successive samples switch sign bits) - more random switching



## Glitch Reduction by Pipelining

- Glitches depend on the logic depth of the circuit - gates deeper in the logic network are more prone to glitching
  - arrival times of the gate inputs are more spread due to delay imbalances
  - usually affected more by primary input switching
- Reduce logic depth by adding pipeline registers
  - additional energy used by the clock and pipeline registers



## Power and Energy Design Space

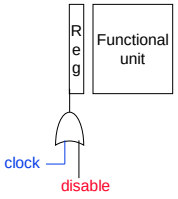
| | Constant Throughput/Latency | | Variable Throughput/Latency |
|---|---|---|---|
| Energy | Design Time | Non-active Modules | Run Time |
| Active | Logic Design<br>Reduced $V_{dd}$<br>Sizing<br>Multi-$V_{dd}$ | Clock Gating | DFS, DVS<br>(Dynamic Freq, Voltage Scaling) |
| Leakage | + Multi-$V_T$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



## Clock Gating

- Most popular method for power reduction of clock signals and functional units

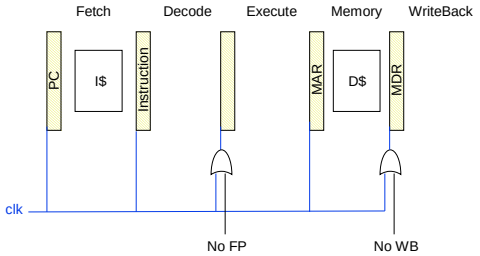
- Gate off clock to idle functional units
  - e.g., floating point units
  - need logic to generate disable signal
    - increases complexity of control logic
    - consumes power
    - timing critical to avoid clock glitches at OR gate output
  - additional gate delay on clock signal
    - gating OR gate can replace a buffer in the clock distribution tree






## Clock Gating in a Pipelined Datapath

- For idle units (e.g., floating point units in Exec stage, WB stage for instructions with no write back operation)



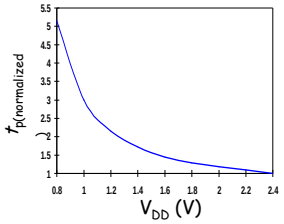
## Power and Energy Design Space

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



## Review: Dynamic Power as a Function of $V_{DD}$

- Decreasing the $V_{DD}$ decreases dynamic energy consumption (quadratically)
- But, increases gate delay (decreases performance)




- Determine the critical path(s) at design time and use high $V_{DD}$ for the transistors on those paths for speed. Use a lower $V_{DD}$ on the other logic to reduce dynamic energy consumption.



## Dynamic Frequency and Voltage Scaling

- Intel's SpeedStep
  - Hardware that steps down the clock frequency (dynamic frequency scaling – DFS) when the user unplugs from AC power
    - PLL from 650MHz → 500MHz
  - CPU stalls during SpeedStep adjustment
- Transmeta LongRun
  - Hardware that applies both DFS and DVS (dynamic supply voltage scaling)
    - 32 levels of $V_{DD}$ from 1.1V to 1.6V
    - PLL from 200MHz → 700MHz in increments of 33MHz
  - Triggered when CPU load change is detected by software
    - heavier load → ramp up $V_{DD}$, when stable speed up clock
    - lighter load → slow down clock, when PLL locks onto new rate, ramp down $V_{DD}$
  - CPU stalls only during PLL relock (< 20 microsec)

## Dynamic Thermal Management (DTM)

Trigger Mechanism:
When do we enable DTM techniques?

Initiation Mechanism:
How do we enable technique?

Response Mechanism:
What technique do we enable?



## DTM Trigger Mechanisms

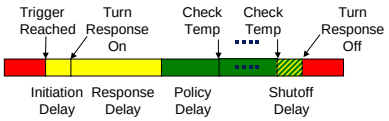
- Mechanism: How to deduce temperature?
- Direct approach: on-chip temperature sensors
  - Based on differential voltage change across 2 diodes of different sizes
  - May require >1 sensor
  - Hysteresis and delay are problems
- Policy: When to begin responding?
  - Trigger level set too high means higher packaging costs
  - Trigger level set too low means frequent triggering and loss in performance
- Choose trigger level to exploit difference between average and worst case power



## DTM Initiation and Response Mechanisms

- Operating system or microarchitectural control?
  - Hardware support can reduce performance penalty by 20-30%
- Initiation of policy incurs some delay
  - When using DVS and/or DFS, much of the performance penalty can be attributed to enabling/disabling overhead
  - Increasing policy delay reduces overhead; smarter initiation techniques would help as well
- Thermal window (100Kcycles+)
  - Larger thermal windows “smooth” short thermal spikes

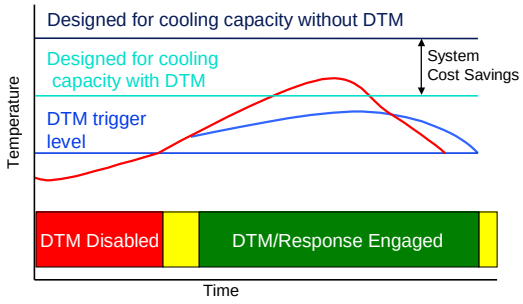
## DTM Activation and Deactivation Cycle




- Initiation Delay – OS interrupt/handler
- Response Delay – Invocation time (e.g., adjust clock)
- Policy Delay – Number of cycles engaged
- Shutoff Delay – Disabling time (e.g., re-adjust clock)



## DTM Savings Benefits

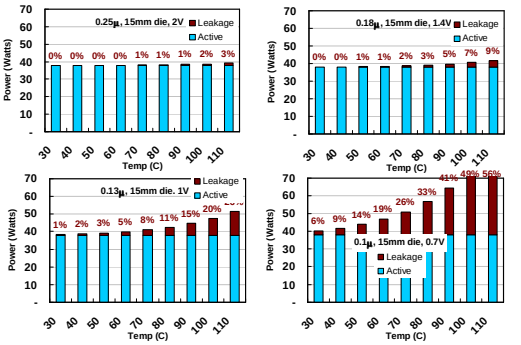





## Power and Energy Design Space

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

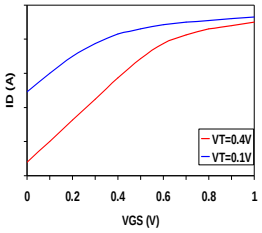
## Speculated Power of a 15mm µP






## Review: Leakage as a Function of Design Time $V_T$

- Reducing the $V_T$ increases the sub-threshold leakage current (exponentially)
- But, reducing $V_T$ decreases gate delay (increases performance)




- Determine the critical path(s) at design time and use low $V_T$ devices on the transistors on those paths for speed. Use a high $V_T$ on the other logic for leakage control.



## Review: Variable $V_T$ (ABB) at Run Time

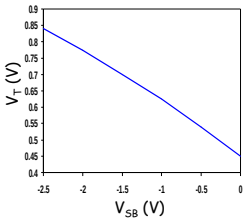
- $V_T = V_{T0} + \gamma(\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|})$

  where $V_{T0}$ is the threshold voltage at $V_{SB} = 0$

  $V_{SB}$ is the source-bulk (substrate) voltage

  $\gamma$ is the body-effect coefficient

- For an n-channel device, the substrate is normally tied to ground
- A negative bias causes $V_T$ to increase from 0.45V to 0.85V
- Adjusting the substrate bias at run time is called adaptive body-biasing (ABB)



## Next Lecture and Reminders

- Next lecture
  - System level interconnect
    - Reading assignment – Rabaey, et al, xx
- Reminders
  - Project final reports due December 5th
  - Final grading negotiations/correction (except for the final exam) must be concluded by December 10th
  - Final exam scheduled
    - Monday, December 16th from 10:10 to noon in 118 and 121 Thomas