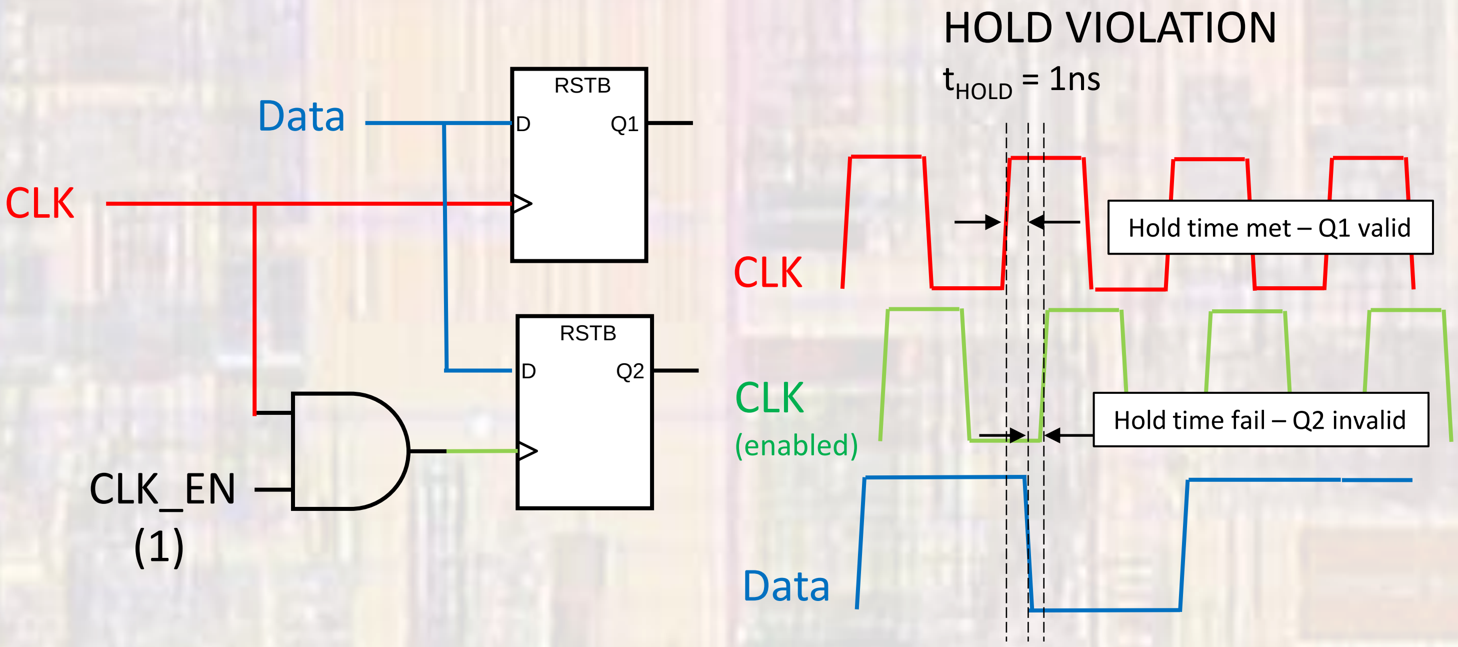
# Dynamic Voltage and Frequency Scaling

Last updated 1/25/24

# Clock Gating

- Clock Gating - Local
  - Local clock gating creates the potential for generating clock skew
    - Leads to Setup/Hold timing violations and functional failures

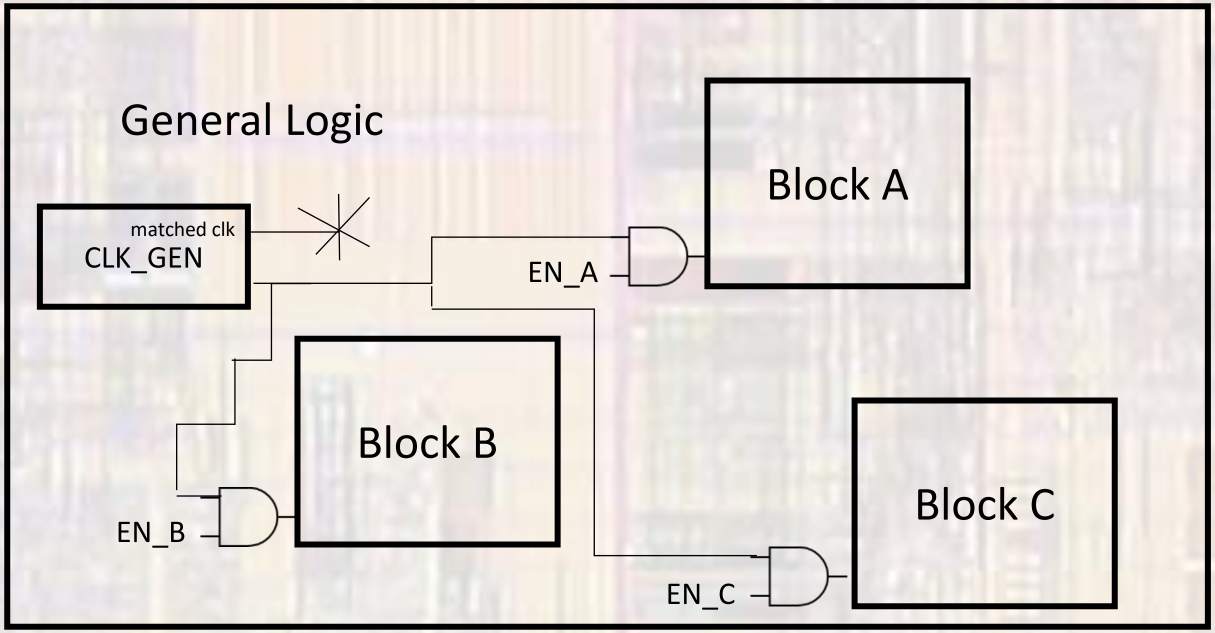


Hold time met – Q1 valid

Hold time fail – Q2 invalid

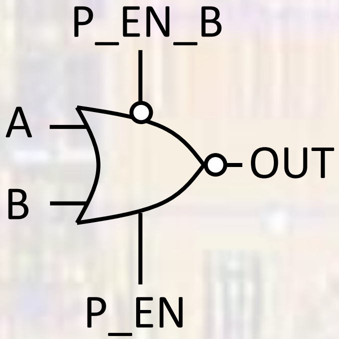
# Clock Gating

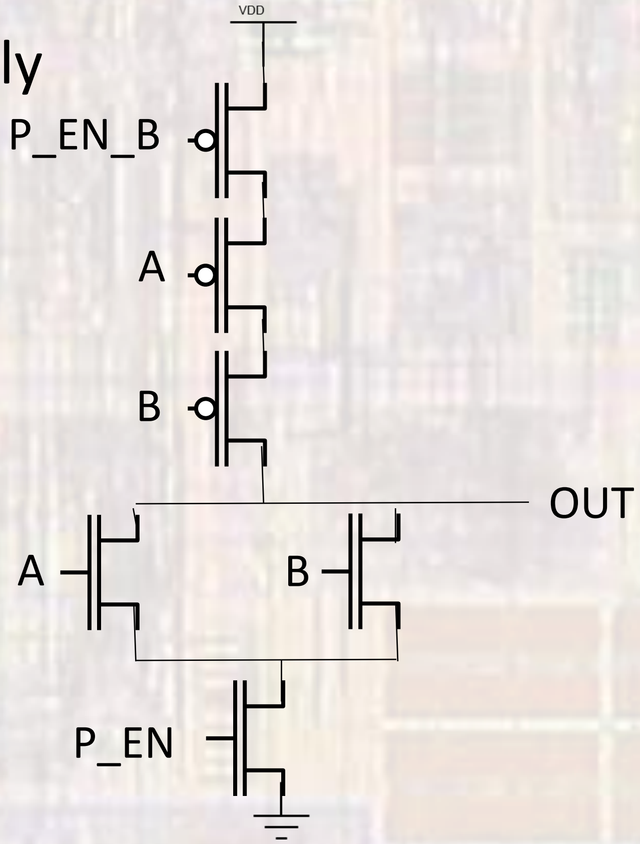
- Clock Gating - Global
  - Gate the clock to entire blocks
    - Removes the switching current components of power
    - Data is retained
    - Requires all blocks to have matched clock paths
    - Restart (enable) must be managed carefully



# Power Gating

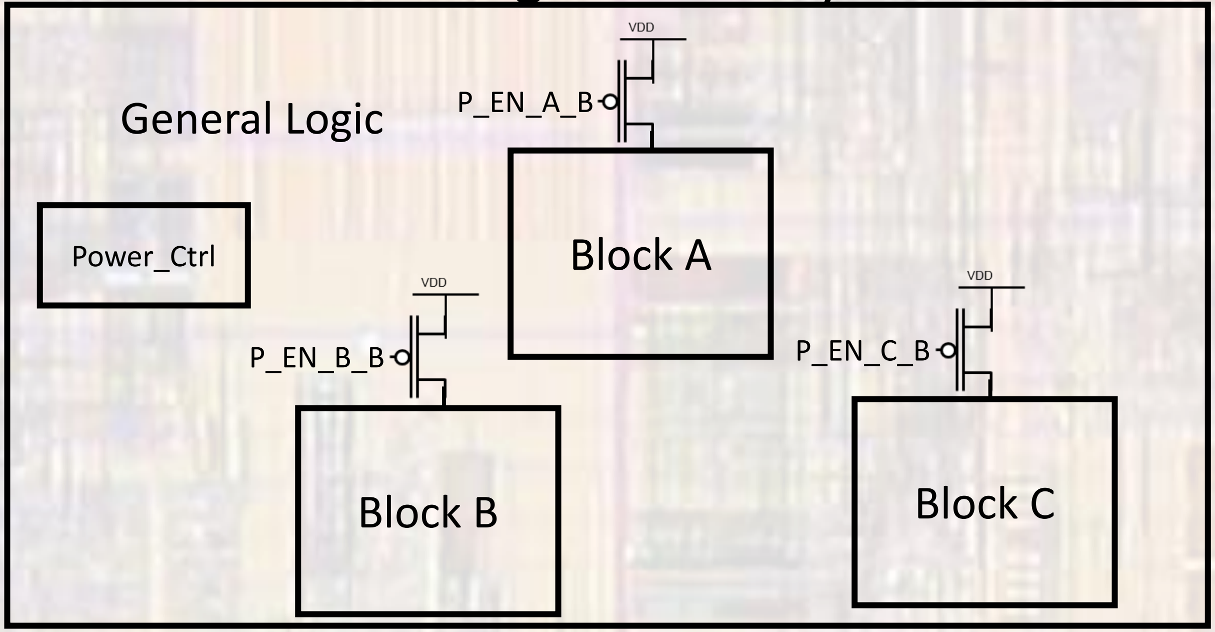
- Power Gating - Local
  - Removes switching and leakage power
  - Add power gating transistors to gates
  - Data is lost when disabled
  - Restart must be managed carefully





# Power Gating

- Power Gating - Global
  - Removes switching and leakage power
  - Add LARGE power gating transistors to the power supply line in blocks
  - Data is lost when disabled
  - Restart must be managed carefully



# DVFS

- Dynamic Voltage and Frequency Scaling (DVFS)
  - Provide just enough capability to get the job done
    - Minimize power in the process
  - From our gate delay discussion

$$t_{pd} = \frac{C_{load}\, V_{dd}}{\frac{k'_n}{2}\frac{W}{L}(V_{dd} - V_t)^2}$$

Increasing Vdd reduces gate delays
→ higher operating frequency

  - From our power discussion

$$P = P_{DC} \quad + \quad P_{SW} \qquad + \qquad P_{Shoot-through}$$

$$P = \beta V_{DD} + \alpha C_{load} V_{DD}^2 F + \alpha I_{peak} V_{DD}\left(\frac{t_r + t_f}{2}\right) F$$

All terms a function of $V_{DD}$

# DVFS

- DVFS – Static implementation
  - Given
    - A fixed set of hardware, e.g. (CPU + memory + gates + peripherals)
    - A required set of tasks to be performed in a specific amount of time
  - The minimum required clock frequency can be calculated
  - Design the system to provide the minimum Vdd required to allow the minimum clock frequency
    - Lowest possible power

$$P = \beta V_{DD} + \alpha C_{load} V_{DD}^2 F + \alpha I_{peak} V_{DD} \left(\frac{t_r + t_f}{2}\right) F$$

# DVFS

- DVFS – Frequency implementation
  - Given
    - A fixed set of hardware, e.g. (CPU + memory + gates + peripherals)
    - A changing set of tasks to be performed in various required amounts of time
  - The required clock frequency for the worst case (highest performance) scenario can be calculated
  - Design the system to provide the minimum Vdd required to allow the maximum required clock frequency
  - Design the system to reduce the clock frequency (PLL) when not in the worst case scenario

$$P = \beta V_{DD} + \alpha C_{load} V_{DD}^2 F + \alpha I_{peak} V_{DD} \left( \frac{t_r + t_f}{2} \right) F$$

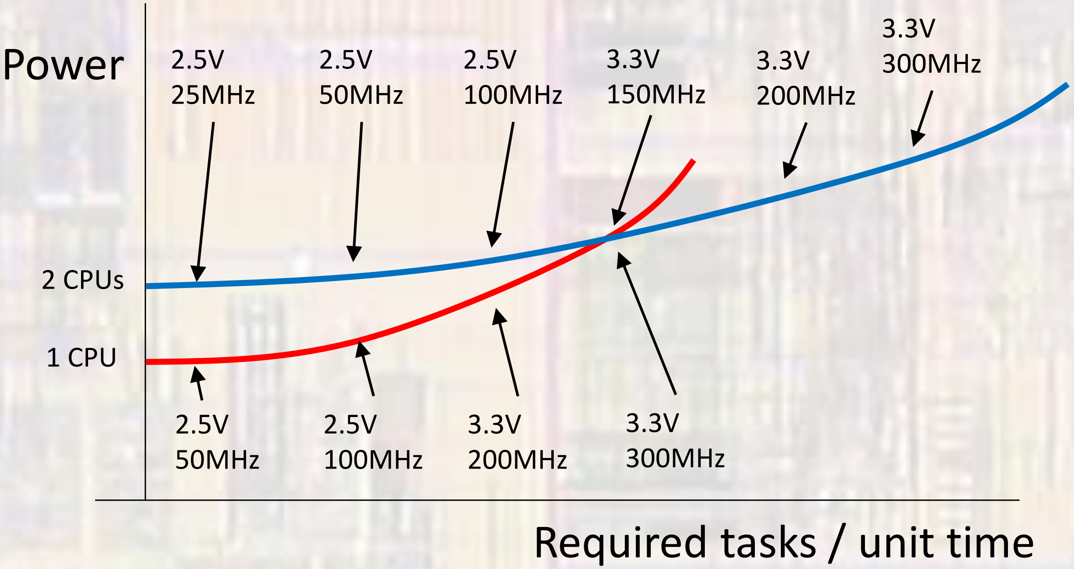
# DVFS

- DVFS – Voltage and Frequency implementation
  - Given
    - A fixed set of hardware, e.g. (CPU + memory + gates + peripherals)
    - A changing set of tasks to be performed in various required amounts of time
  - The required clock frequency for the worst case (highest performance) scenario can be calculated
  - Design the system to reduce Vdd and the clock frequency (PLL) when not in the worst case scenario

$$P = \beta \enclose{circle}{V_{DD}} + \alpha C_{load} \enclose{circle}{V_{DD}^2} \enclose{circle}{F} + \alpha I_{peak} \enclose{circle}{V_{DD}} \left(\frac{t_r + t_f}{2}\right) \enclose{circle}{F}$$

# DVFS

- DVFS – System implementation
  - Given
    - A variable (through clock or power gating) set of hardware, e.g. (2 CPUs + memory + gates + peripherals)
    - A changing set of tasks to be performed in various required amounts of time
    - Modify Vdd, F, and resources as required


Power
2 CPUs
1 CPU
2.5V 25MHz
2.5V 50MHz
2.5V 100MHz
3.3V 150MHz
3.3V 200MHz
3.3V 300MHz
2.5V 50MHz
2.5V 100MHz
3.3V 200MHz
3.3V 300MHz
Required tasks / unit time