CLOCKS
Clocks in Digital Systems

• Clocks pace the flow of data inside digital processors
• The exact speed of data through circuits is impossible to predict accurately due to factors such as:
  – Fabrication process variations
  – Supply voltage variations
  – Temperature variations
  – Countless parasitic effects (e.g., wire-to-wire capacitances)
  – Data-dependent variations (e.g., 1 OR 1 = 1 versus 1 OR 0 = 1)
• Clocked pipeline memory elements slow down the fastest signals, wait until all signals have finished propagating through logic, and release them into the next pipeline stage simultaneously
Robust Clock Design

- Edge-triggered memory elements (flip-flops) are generally more robust than level-sensitive memory elements (latches) if the clock can be delivered with low skew.

- Safe design practices (always follow these rules in this class):
  - *Only clock signals may connect to flip-flop or latch clock inputs*
    - A simpler circuit may sometimes be possible if a logic signal is connected to a clock input, but do not do it for robust design.
  - *Clock signals may not connect to any node other than a flip-flop or latch clock input*
    - No logic gate inputs
    - No flip-flop or latch inputs other than the clock input
  - A few common exceptions that must be very carefully designed include:
    - Clock generation circuits
    - Clock gating circuits
    - Clock tree buffers
Multi-rate Hardware Clocking

- There are three main approaches to clocking multi-rate systems

1) Build slower divided clocks with FFs
   - Some FFs are clocked by the real clock signal, others are clocked by a delayed slower clock\_freq-half signal coming from a frequency divider. Significant clock skew \(\rightarrow\) potential for dead chip 😞
   - Could risk your job security (moderate exaggeration)

2) Use multi-frequency clocks
   - Requires an independent clock tree for each frequency and possibly an independent phased-locked loop (PLL)
   - May save significant power in main processing circuits
   - Each PLL uses significant power
Multi-rate Hardware Clocking

There are three main approaches to clocking multi-rate systems

3) Clock all logic with highest-rate clock
   - Utilize small simple counters that load registers or route signals on only certain clock edges (for example, every fourth clock edge for $freq/4$).
   - Counters must be reset simultaneously and reset signal must meet timing requirements at the highest frequency.
   - Definitely the simplest and most robust
   - Design in only this way in this class

Enable = 1 every fourth cycle is equivalent to using $freq/4$