Clock Signal Definitions

- **Rising-edge**
- **Falling-edge**

**Period**

\[ P = \tau_{\text{HIGH}} + \tau_{\text{LOW}} \]

**Frequency**

\[ f = \frac{1}{P} = \frac{1}{\tau_{\text{HIGH}} + \tau_{\text{LOW}}} \]

**Duty Cycle**

\[ \text{Duty Cycle} = \frac{\tau_{\text{HIGH}}}{\tau_{\text{HIGH}} + \tau_{\text{LOW}}} \]
Latches Work Best In Pairs

The first stage operates while the clock is low and inverts and amplifies the arriving signal and charges or discharges $C_{L1}$.

The second stage operates while the clock is high and inverts the signal on $C_{L1}$ to charge or discharge $C_{L2}$ and downstream logic gate inputs.

Latch Implementation: Lumped

Latch 0  Gate 1  Gate 2  Latch 1
Latch Operation: Lumped

Latency and Throughput

Latency $L$ is the delay between the rising edge of the clock on $L0$ and the data being valid internally in the last latch.

$$L_{LUMPED} = \tau_{L_{EXT}} + \tau_{GATE1} + \tau_{GATE2} + \tau_{L_{INT}}$$

$$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 8\tau_{INV}$$

Throughput $T$ is the bits per second through the latches and is the maximum clock frequency.

$$P_{LUMPED} = \tau_{L_{EXT}} + \tau_{GATE1} + \tau_{GATE2} + \tau_{L_{INT}}$$

$$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 8\tau_{INV}$$

$$F_{LUMPED} = 1/8(345ps) = 0.36 \text{ GHz}$$
Latch Implementation: Pipelined

Latch Operation: Pipelined
Latch Timing Diagram

A1 = 0  B2 = 1  C2 = 0  B1  1 => 0

Latency 32 inverter delays

Throughput = 1/(20 x 345ps) = 0.145 GHz

Clock Optimization: Pipelined

\[ \tau_{\text{HIGH}} = \tau_{L,\text{EXT}} + \max (\tau_{\text{GATE1}}, \tau_{\text{GATE2}}) \]

\[ \tau_{\text{LOW}} = \tau_{L,\text{INT}} \]
Latency and Throughput

Latency $L$ is the delay between the rising edge of the clock on $L0$ and the data being valid internally in the last latch.

$L_{LUMPED} = \tau_{L\_EXT} + \tau_{GATE1} + \tau_{GATE2} + \tau_{L\_INT}$

$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 8\tau_{INV}$

$L_{PIPILED} = \tau_{L\_EXT} + \tau_{L\_INT} + \tau_{GATE1} + \tau_{GATE2} + \tau_{L\_INT}$

$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 12\tau_{INV}$

Throughput $T$ is the bits per second through the latches and is the maximum clock frequency.

$P_{LUMPED} = \tau_{L\_EXT} + \tau_{GATE1} + \tau_{GATE2} + \tau_{L\_INT}$

$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 8\tau_{INV}$

$F_{LUMPED} = 1/8(345\text{ps}) = 0.36 \text{ GHz}$

$P_{PIPILED} = \tau_{L\_EXT} + \max(\tau_{GATE1}, \tau_{GATE2}) + \tau_{L\_INT}$

$= 2\tau_{INV} + 2\tau_{INV} + 2\tau_{INV} = 6\tau_{INV}$

$F_{PIPILED} = 1/6(345\text{ps}) = 0.48 \text{ GHz}$

Logic Pipeline: Example #2

$\tau_{HIGH} = \tau_{L\_EXT} + \max(\tau_{GATE1}, \tau_{GATE2})$

$\tau_{LOW} = \tau_{L\_INT}$
Logic Limitations: Power Consumption

• The resistive load of NMOS results in D.C. current and hence static power consumption given by the product of current times voltage.
• CMOS avoids this static loss as the pull-up device shuts off the current completely.
• CMOS still suffers a.c. power consumption that is proportional to the switching frequency.
• The energy expended per cycle of charging and discharging can never be less than $CV^2$

$$P_{Dynamic} = \left(\frac{1}{2}\right)CV^2N_{Gates}F_{Active}f_{Clock}$$

Logic Gates: CMOS

• The inverting style gates are the smallest (hence cheapest) and fastest.
• Pull-up is PMOS only and Pull-down is NMOS only
• Pull-up and Pull-down networks have the same inputs
• The Pull-up and pull-down are the dual of each other. [Learn more about this in EECS 141. One network can be determined from the other by placing nodes at the centers of the windows and to the left and right of the networks. The dual topology is then found by connecting the nodes via arcs through the transistors]
Logic Limitations: Delay

• Any node that changes voltage has capacitance that must be charged or discharged.
• The charging and discharging rate is set by the available current.
• As the number of logic inputs increases the capacitance to be charged increases and the average available current decreases. The delay grows at a power larger than unity.
• For >5 inputs cascading can be used to keep the growth in delay with the number of inputs nearly linear.
• Delayed is modeled many ways. Typically the output must change by $V_{DD}/2$. Often the delay is taken as RC where $R = 0.75V_{DD}/I_{SAT}$ and corresponds to the a switched equivalent resistance model the transistor.

Logic Limitations: Delay Data Dependence

• The best and worst case delays for logic circuits are inherently data dependent
• This can lead to race conditions in which an input signal that arrives earlier than other input signals can trigger an incorrect output that then continues to propagate further
• A latch circuit can synchronize the signals by interrupting the signal flow until all of the inputs are correct and a clock signal releases them
• A two inverter style latch works by storing the inverse of the arriving signal on the capacitance between the two-inverters on the clock low and creating its inverse to drive the latch output on the clock high
• Pipelining consists of using smaller logic functions with more latches inbetween them. While pipelining increases overall delay (latency) it can increase the completed evaluations per unit time (throughput) when multiple sets of data are being processed