ECE321 – Electronics I

Lecture 25: Sequential Logic: D Flip-flop

Payman Zarkesh-Ha

Office: ECE Bldg. 230B
Office hours: Tuesday 2:00-3:00PM or by appointment
E-mail: payman@ece.unm.edu

Review of Last Lecture

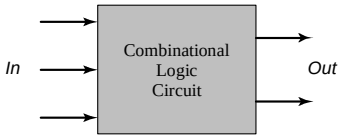
- **CMOS Dynamic Gates**
  - **Circuit Style**
  - **Pros and Cons**
  - **Charge Sharing Issue**

Today's Lecture

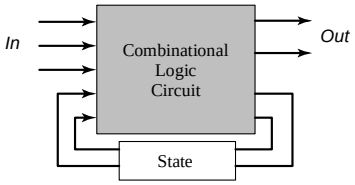
- **Sequential Logic**
  - Latches and Flip-Flops
  - Timing Characteristics
  - Design of Latches and Flip-Flops
  - Setup and Hold Time Issues

Combinational versus Sequential Logic

Combinational Logic: Out=f(in)



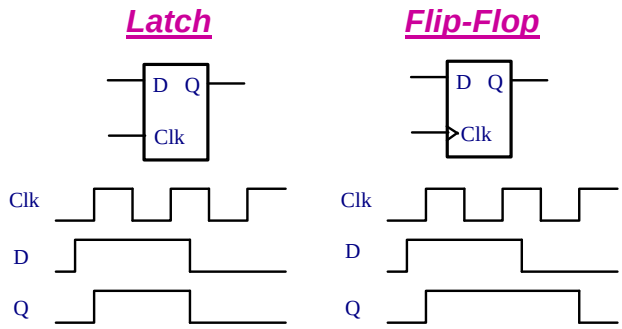
sequential Logic: Out=f(in,previous_in)



- **Combinational Logic:**
  - Output is a function of present inputs (delayed by the propagation delay) i.e., do not contain memory
  - Implements logic functions like NAND, NOR, XOR, Multiplex or any complex functions such as Decoder, adder, shifter
- **Sequential Logic:**
  - Implement memory
  - Stores past values
  - Edge sensitive: Flip-flops
  - Level sensitive: Latches

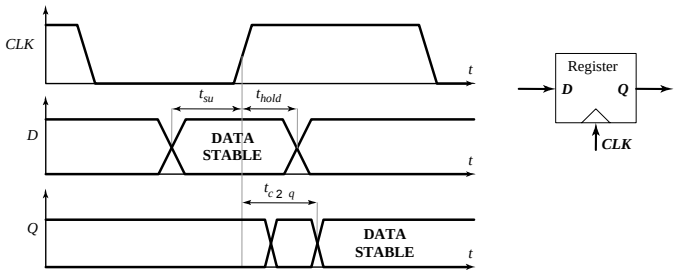
Latches and Flip-Flops

- Latches store data when clock is low
- Flip-flops or registers store data when clock rises
- Usually flip-flops are build by using two latches (we will explain later)



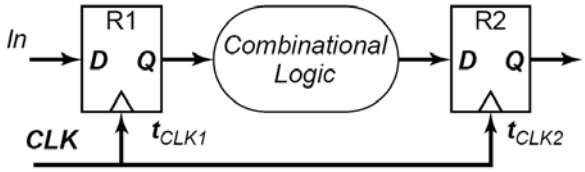
Timing Characteristics of Registers

- Setup time (t_{su}) is the time that data (D) must be valid before clock transition
- Hold time (t_{hold}) is the time that data (D) must remain valid after clock transition
- Propagation delay (t_{C2Q}) is the delay time that the data (D) is copied to output (Q) with reference to clock edge



Timing Constraints

- **Minimum Cycle Time Constraints: $T \geq t_{C2Q} + t_{Logic} + t_{su}$**
  - Worst case is when receiving edge arrives early
  - This constraint set the limit to maximum clock frequency that the circuit will be operable
- **Hold Time Constraints: $t_{(C2Q,\ CD)} + t_{(Logic,\ CD)} > t_{hold}$**
  - CD is contamination delay (fastest possible delay)
  - This is a race between data and clock
  - If this constraint doesn't hold the circuit is ***not functional*** at any clock freq.
- **It is therefore important to minimize t_{su}, t_{hold}, and t_{C2Q}**

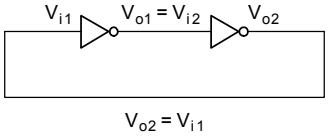


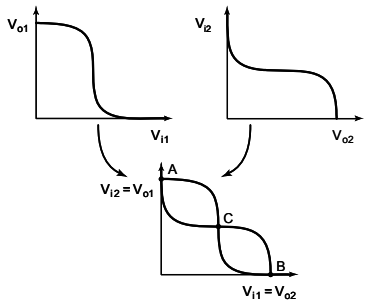
Requirements in Flip-Flop Design

- **High speed:**
  - Small Clk-Output delay
  - Small setup time
  - Small hold time → Inherent race immunity
- **Low power**
- **Small clock load (clock power is very large)**
- **High driving capability**
- **Robustness**
- **Crosstalk insensitivity**

Basic Static Memory Cell

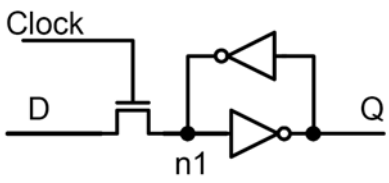
- Static memories use positive feedback to create a *bistable circuit*
- These circuits can hold state and are thus termed "bi-stable"





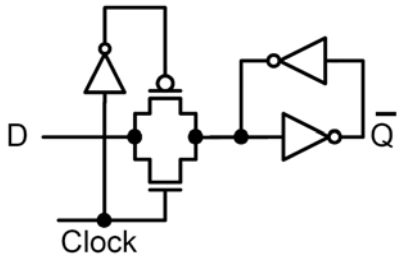
Basic Latch

- We need the ability to set the state of the latch
- A pass-gate input works nicely
- There are a number of problems that it might have…
  - The NMOS pass gate has difficulty driving a logic 1 onto the first node (n1)
  - This can make the gate un-writable at low voltages or with process variation
  - The strength of the input at D is a function of the strength of the previous



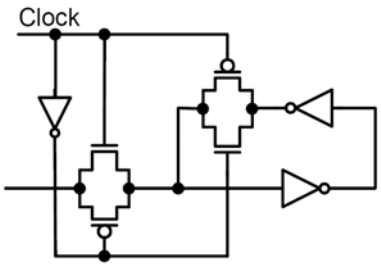
A Better Latch

- Using a CMOS transmission gate improves the write-ability
- NMOS and PMOS help each other to pass logic 0 and 1 strongly
- There are still a number of problems that it might have…
  - The drive strength still depends upon the previous gate
  - The inputs must be strong enough to overpower the feedback in all conditions - To make this gate work, the feedback must be weak
  - There can be substantial short-circuit current due to contention between the feed-forward and feed-back path



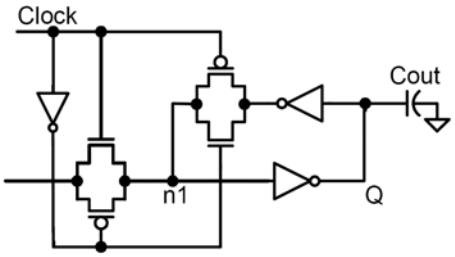
Another Improvement on Latches

- Adding another CMOS transmission gate improves the write-ability
- This removes the feedback during write model
- There are still a number of problems that it might have…
  - Time dependent Setup
  - Charge sharing (back-writing)



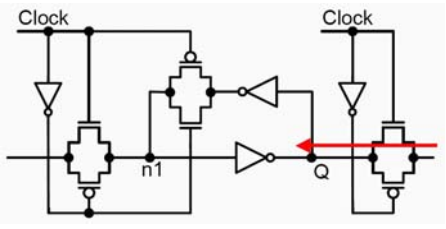
Problem 1: Time Dependent Setup

- In this design, the write speed is dependent on the capacitive load on the output
- The node that must be written to hold state is Q, not n1
  - Just getting n1 to the right point is not enough…
  - When the clock changes, the pass gate from Q turns on and can drive n1 back!
  - Thus, write timing is to the feedback, not the feed-forward node
- Timing should not be dependent on the next gate (if possible)



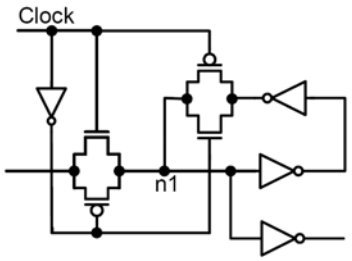
Problem 2: Charge Sharing (Back-Writing)

- This problem occurs when this gate is connected to another pass-gate circuit
  - Pass gates are bidirectional
- If the capacitive load is much higher on the other side of the pass gate attached to Q, this can be stronger than the feedforward path and write the latch from the output towards the input!
  - This is a kind of "charge-sharing" noise



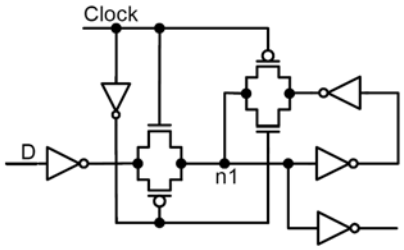
Another Latch Improvement

- This is all fixed by adding transistors (an inverter) to decouple the feedback path from the output
- This also has the advantage that the feedback path can be small, even when the output inverter is quite large
- There are still a number of problems that it might have…
  - n1 is sensitive to input noise



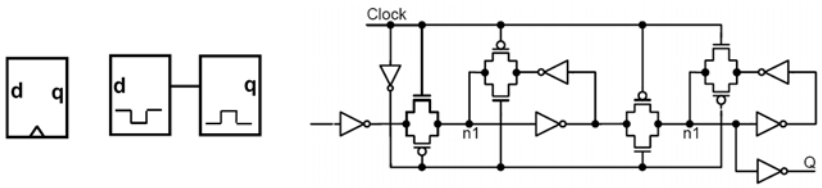
Final Latch Improvement

- The problem with input noise, like the last case is easily fixed with (yet another) inverter
- This lath circuit is very robust to noise
- However, the delay through the latch is long
  - The lath delay is a penalty that we have to live with
  - This delay subtract from circuit speed



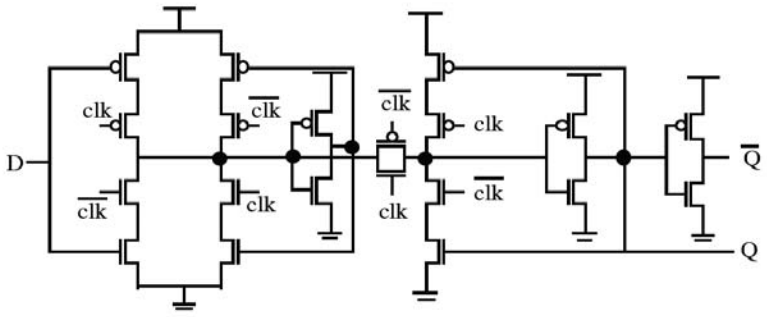
Flip-Flop

- Flip-flops are built out of back to back latches
- They are edge triggered
- Since one lath is transparent in each clock phase, the output can transition only at the clock edge (falling edge in this circuit)
- Flip-flops have quite a bit of delay penalty
  - For example the DEC alpha microprocessor has a maximum logic depth of 12 gates. There are 2-4 gates consumed in the flip-flop.



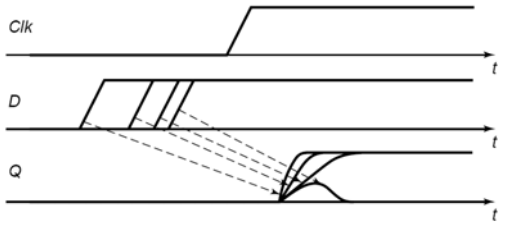
Other Circuits for Flip-Flop

- Flip-Flop can be made using tri-state inverters as shown here.
- How this circuit work?



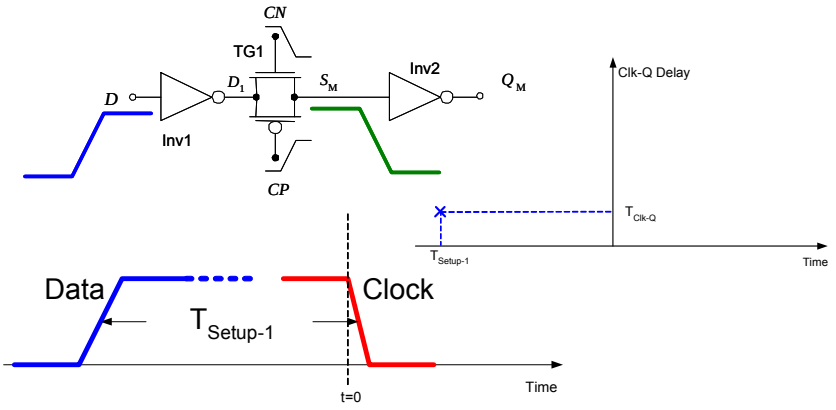
Setup Time Definition

- As the setup time for a FF is violated the failure is soft
- Some behavior like the latch occurs, dependent on the timing at input D
- The net effect is that t_{C2Q} increases as this happens
- The "right" value for t_{su} is the one that minimizes the overall flip-flop overhead, i.e. sum of t_{su} + t_{C2Q}



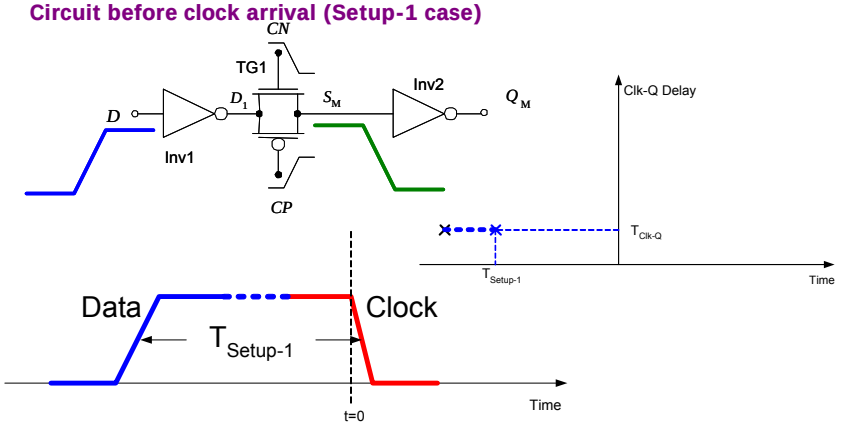
Setup Time Illustrations

Circuit before clock arrival (Setup-1 case)



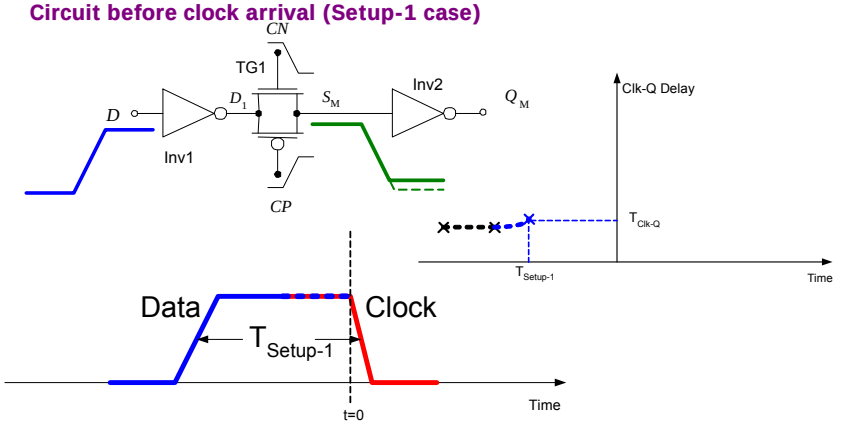
Setup Time Illustrations

Circuit before clock arrival (Setup-1 case)



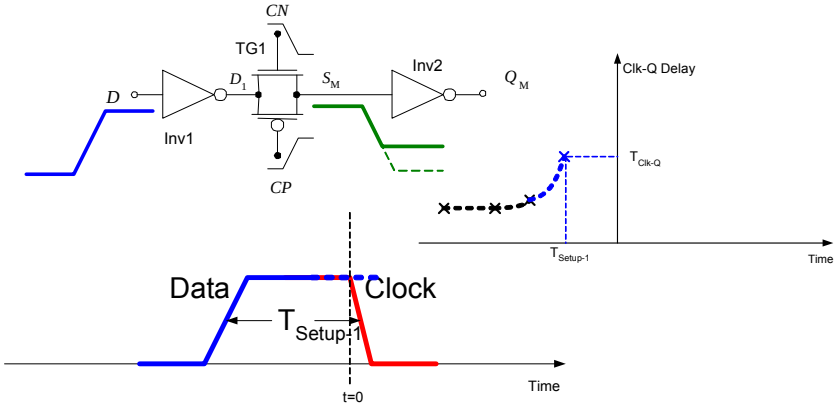
Setup Time Illustrations

Circuit before clock arrival (Setup-1 case)



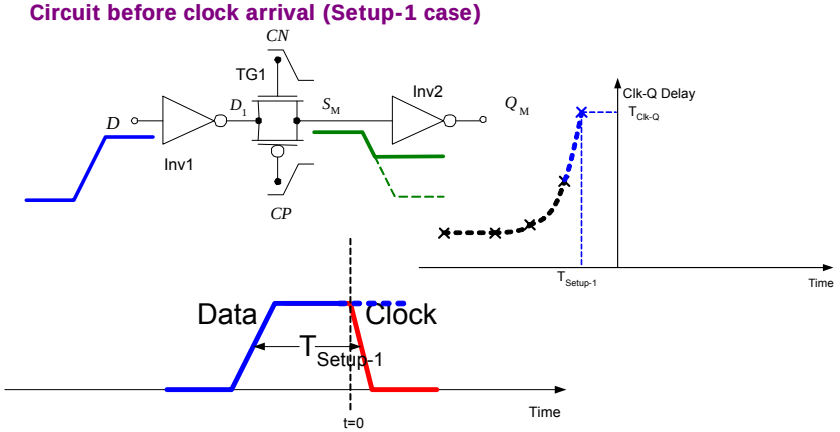
Setup Time Illustrations

Circuit before clock arrival (Setup-1 case)

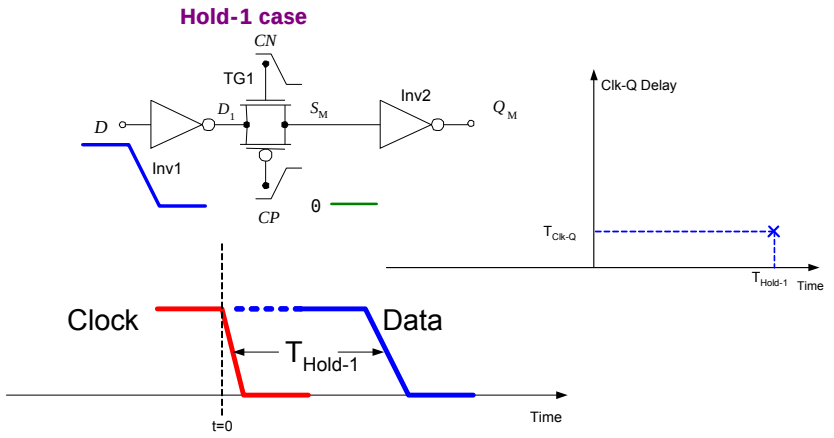


Setup Time Illustrations

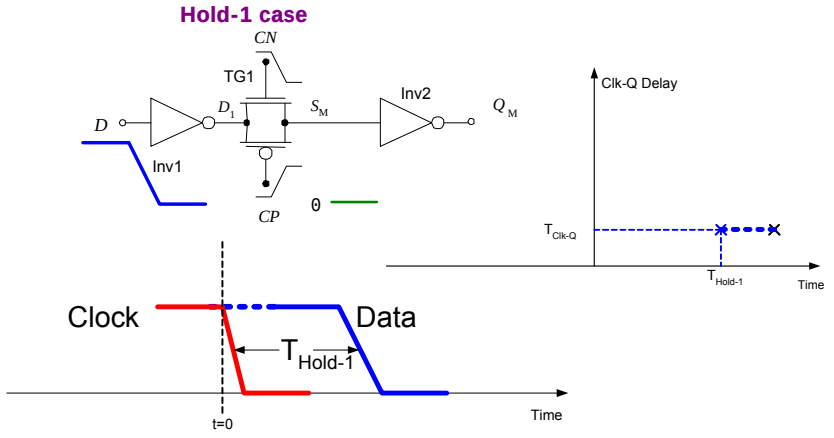
Circuit before clock arrival (Setup-1 case)



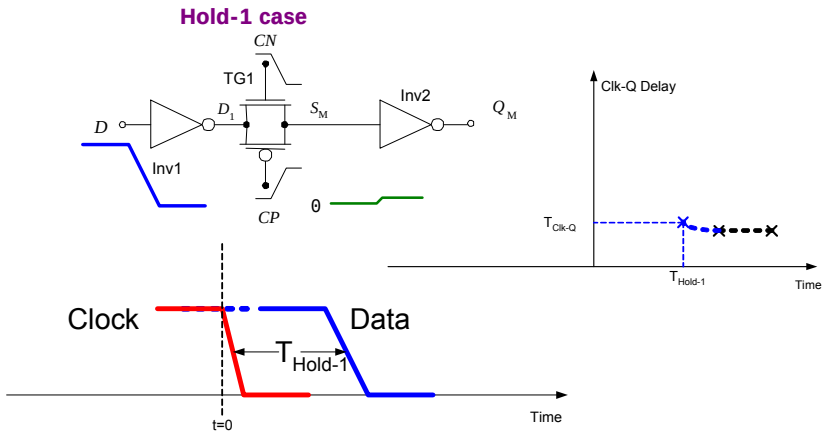
Hold Time Illustrations



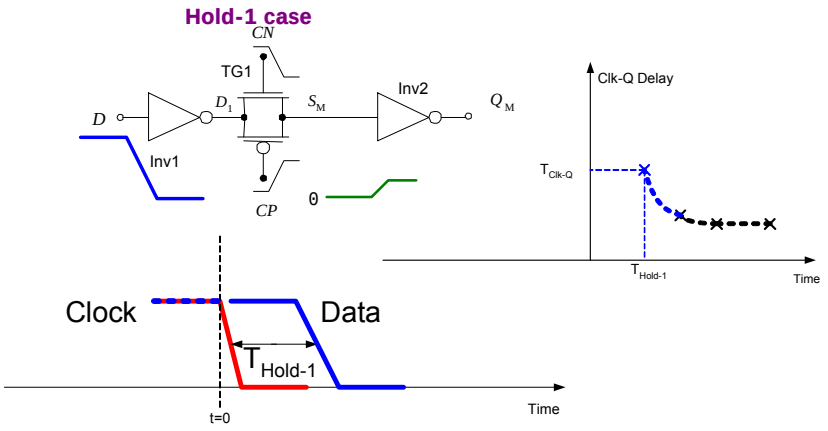
Hold Time Illustrations



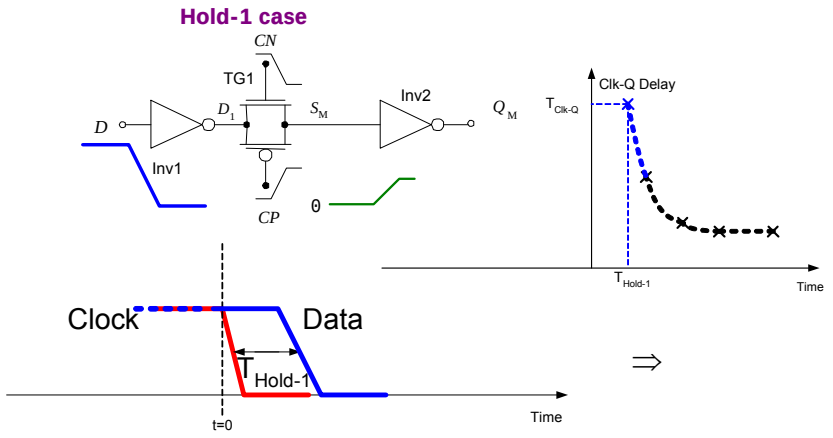
Hold Time Illustrations



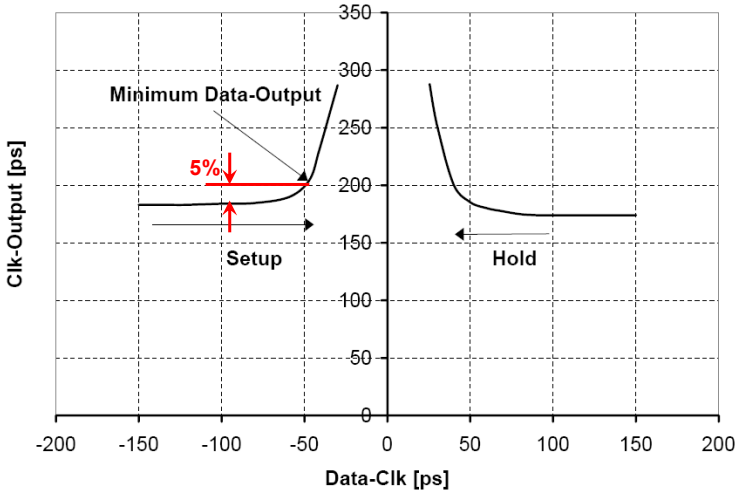
Hold Time Illustrations

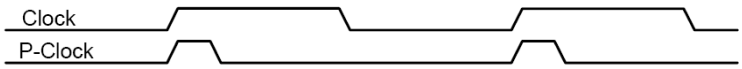


Hold Time Illustrations



CLK2Q versus Setup and Hold Times



Pulse-Clocked Latches

- A latch can simulate a flip flop by using short duty cycle clock
- The latch is transparent only when clock is high, therefore by making the clock high short, it appears to be edge triggered
- This method provides *less delay* and *less load on the clock*
  - About two gate delays are gained per cycle, 10% of the logic depth of 20
  - If done correctly, the clock power can be reduced by about 44% (why?)
- However, the penalty is that the hold time is increased by the pulse width
  - Remember, a hold time violation renders the chip inoperable at all speeds