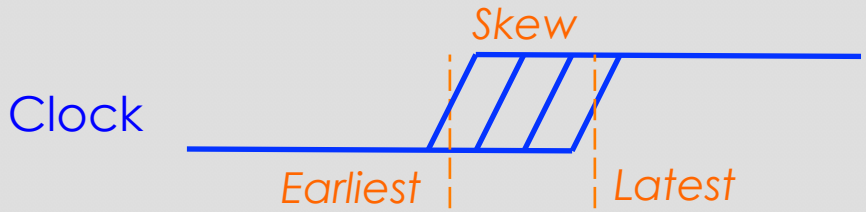
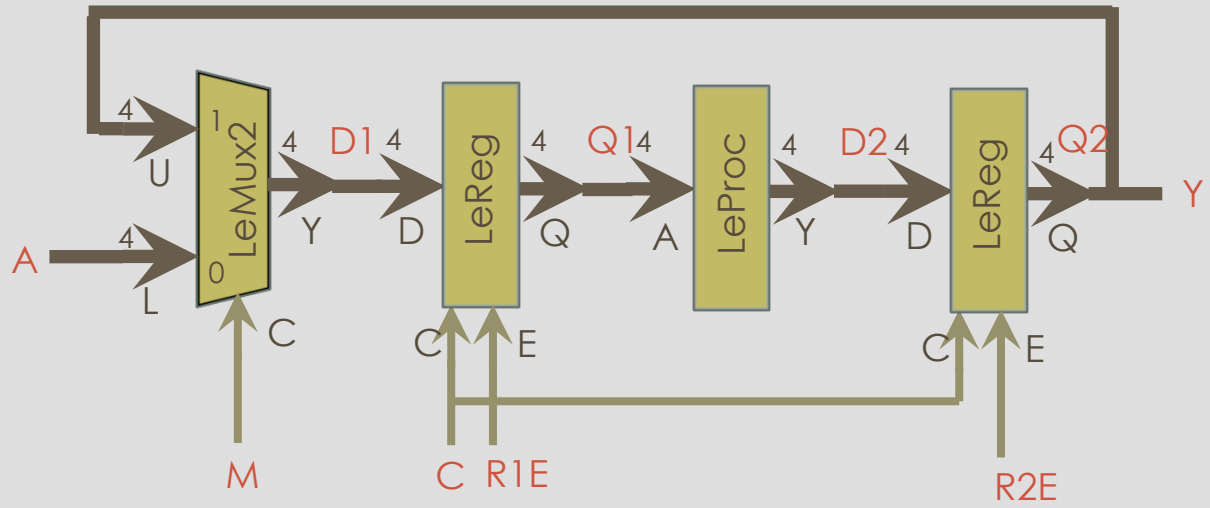
COMPUTER ARCHITECTURE & DESIGN

5

CLOCK SKEW

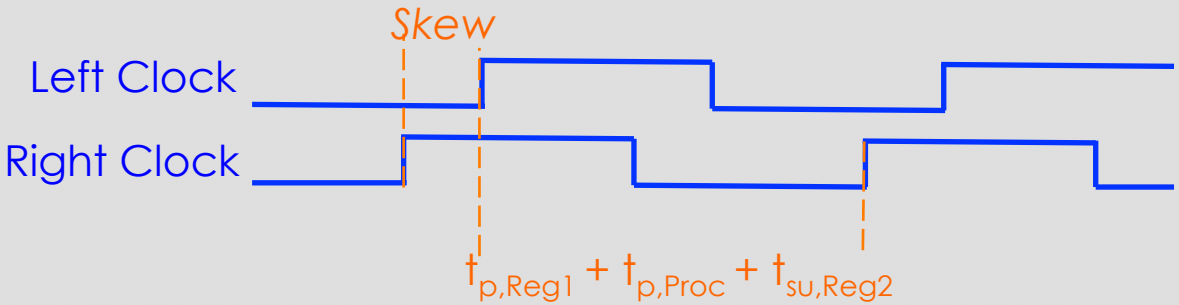
- We might want to have all storage devices clocked at the same time, but this might not occur even if all storage devices are clocked on the same edge of the same clock
  - Different wire lengths from the clock source to the storage devices present different delays
  - Clock gating on different paths can introduce different delays





Nominal time from left register to right register: $t_{p,Reg1} + t_{p,Proc} + t_{su,Reg2}$

Suppose the right register is clocked earlier than the left register by t_{skew}.



Time needed from left register to right register becomes:

$t_{p,Reg1} + t_{p,Proc} + t_{su,Reg2} + t_{skew}$

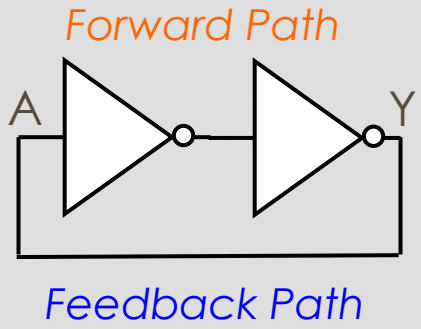
The clock must be slower to ensure operation with skew.

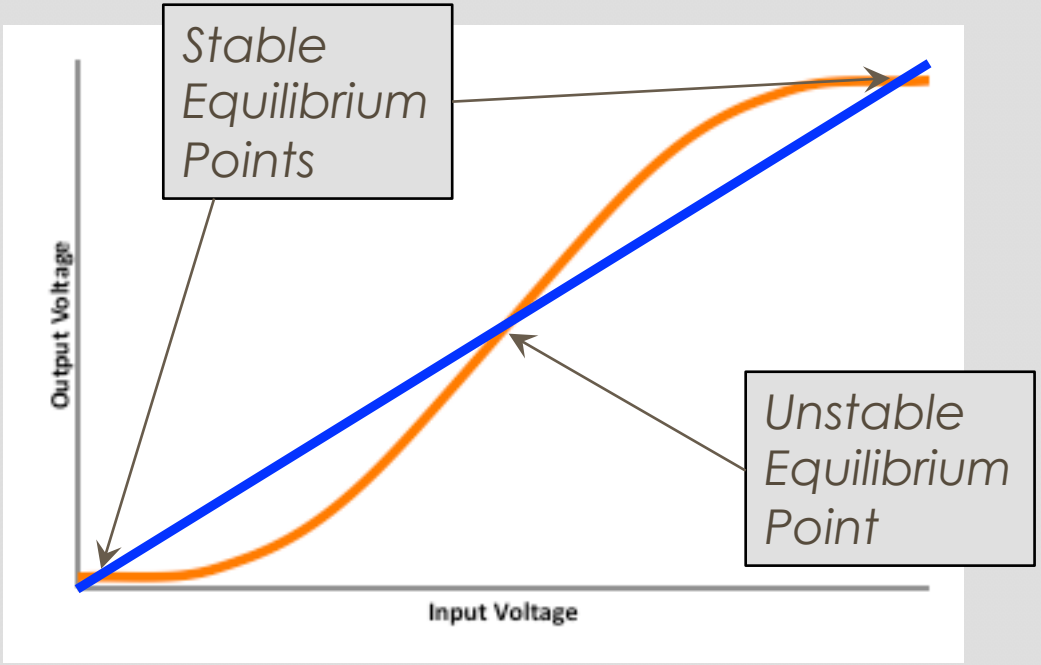
CLOCK SKEW AND HOLD TIME

- Clock skew shifts the controlling clock edge applied to one storage device with respect to the controlling edge of another storage device
- Clock skew increases both setup and hold time requirements
  - Skew may be in either direction – early or late
- Design efforts are applied to limit clock skew

APERTURE TIME VIOLATION

- What happens if aperture time constraints are violated?
- Consider a simple bistable storage element





METASTABILITY

- The simple bistable storage device has three points of equilibrium between the forward path and the feedback path
  - Two of these equilibrium points are are stable
  - One equilibrium point is unstable – this is the metastable point

THREE PROBLEMS OF METASTABILITY

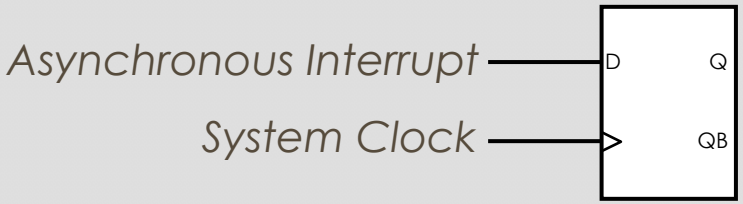
- The metastable point is in the forbidden zone.
  - The output voltage does not map to a valid logic value
- The equilibrium is unstable, and the device will leave the metastable point with slight perturbation.
  - We cannot predict how long the device will remain at the metastable point.
- When the device leaves the metastable point, we do not know which way it will go.
  - The device could go to the stable high value or the stable low value.

REACHING METASTABILITY

- Inputs must deliver enough energy to move the device away from one stable equilibrium point but not enough to get to the other stable equilibrium point.
- An aperture violation can yield metastability.
- While metastability is an unlikely outcome, many aperture violations may eventually yield a few metastable outcomes.

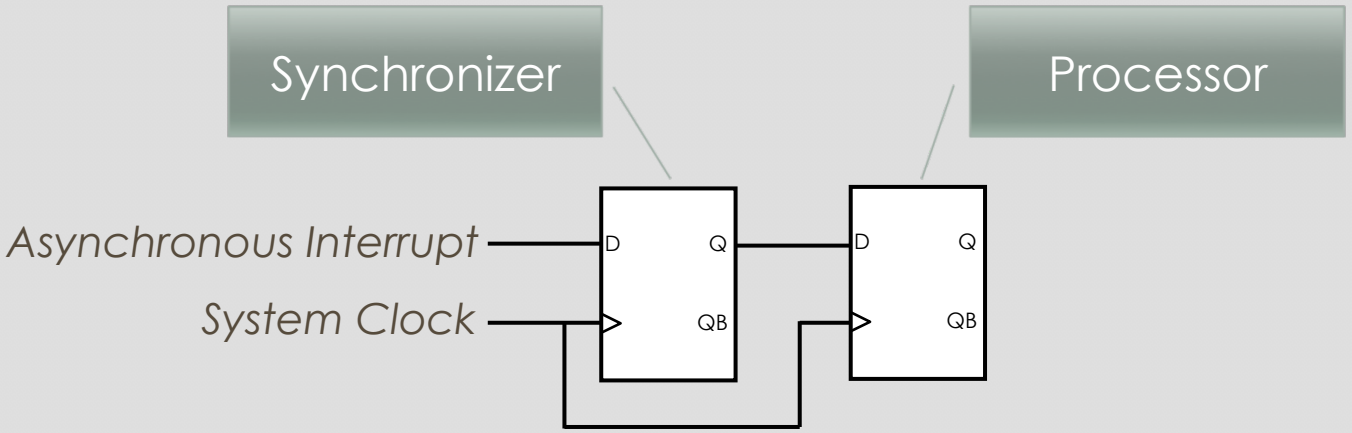
AVOIDING METASTABILITY

- Synchronous designs with clock specifications to avoid aperture violations
- Consider synchronous designs with external asynchronous inputs such as interrupts



Can this storage device become metastable?

ADDING A SYNCHRONIZER



The synchronizer D-type flip-flop is loaded using the same clock edge that is used to load the input storage within the processor.

SYNCHRONIZER OPERATION

- If the asynchronous interrupt input changes within the aperture time of the synchronizer, there are three possible results.

1) The synchronizer captures the new value and presents this value to the output. The new value is transferred to the processor synchronously on the following clock.
2) The synchronizer misses the new value. The asynchronous input does not change and is captured by the synchronizer without aperture violation on the following clock.

SYNCHRONIZER METASTABILITY

- The third possible outcome is synchronizer metastability.
- If the metastability resolves in time to satisfy t_{SU} before the next clock edge, then
  - If the resolution is to the new value, the new value is loaded into the processor.
  - If the resolution is to the old value, then the new value will loaded into the synchronizer without an aperture violation for transfer to the processor on the following clock.
- If the synchronizer metastability does not resolve in time, then the processor input might become metastable.

SYNCHRONIZER VALUE

- Adding a synchronizer cannot guarantee that the processor input storage will avoid all metastabilities.
- Adding a synchronizer significantly decreases the likelihood that the processor input storage will avoid most mestabilities.
- The synchronizer adds at least one clock cycle to the recognition of an asynchronous input change.