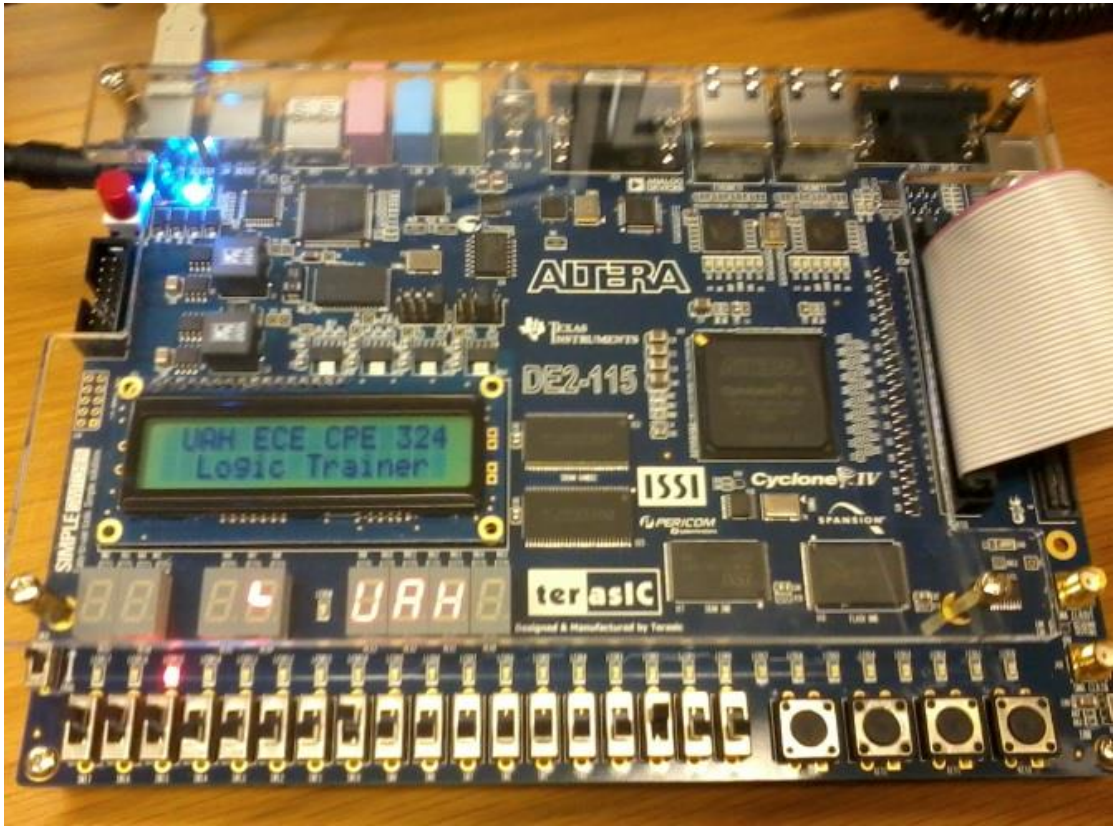
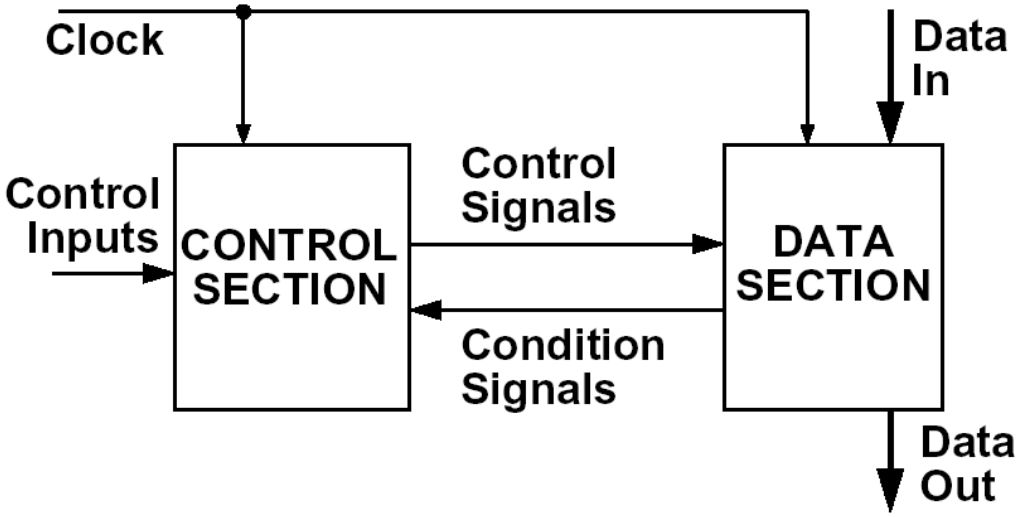
CPE 322
Digital Hardware Design Fundamentals

Timing Considerations in Synchronous Sequential Design

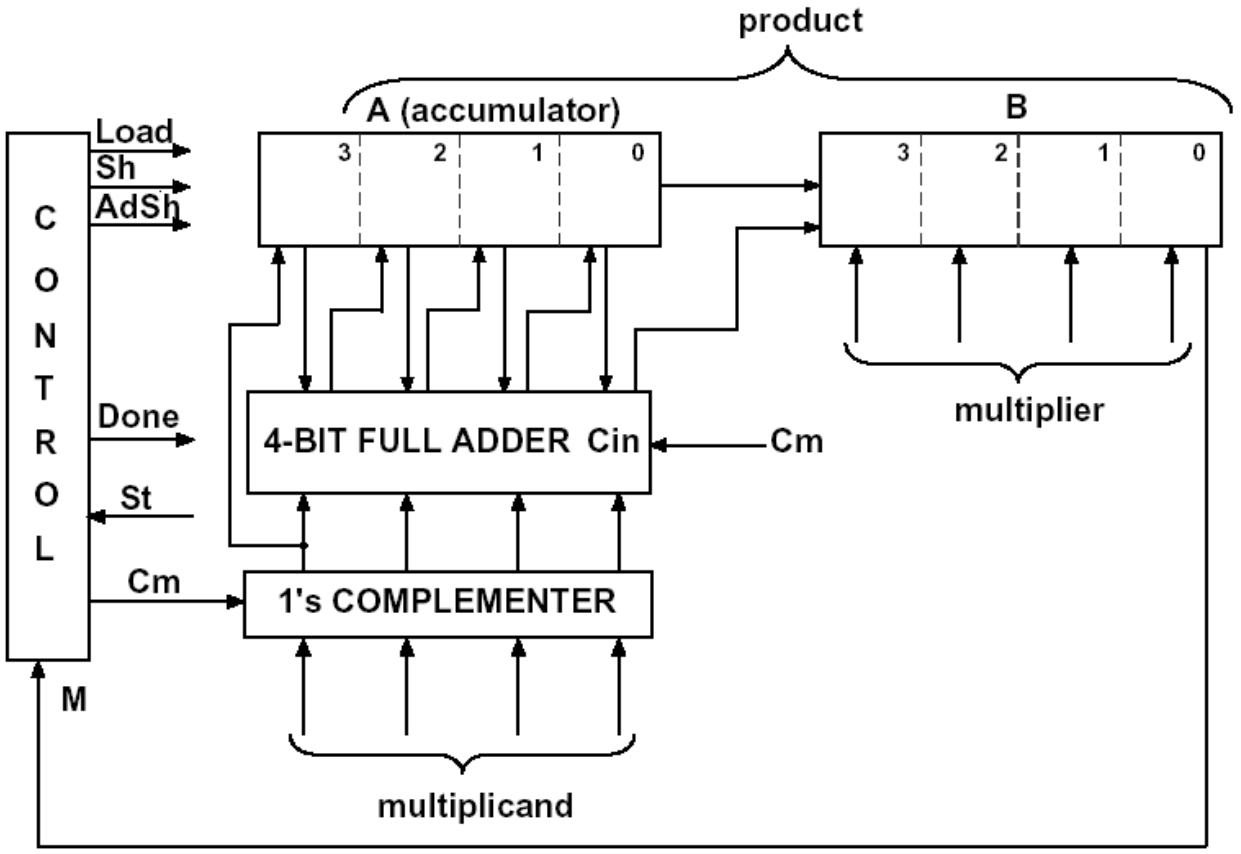


Synchronous Design

- Use a clock to synchronize the operation of all flip-flops, registers, and counters in the system
  - all changes occur immediately following the active clock edge
  - clock period must be long enough so that all changes flip-flops, registers, counters will have time to stabilize before the next active clock edge
- Typical design: Control section + Data Section



Example: Faster Multiplier

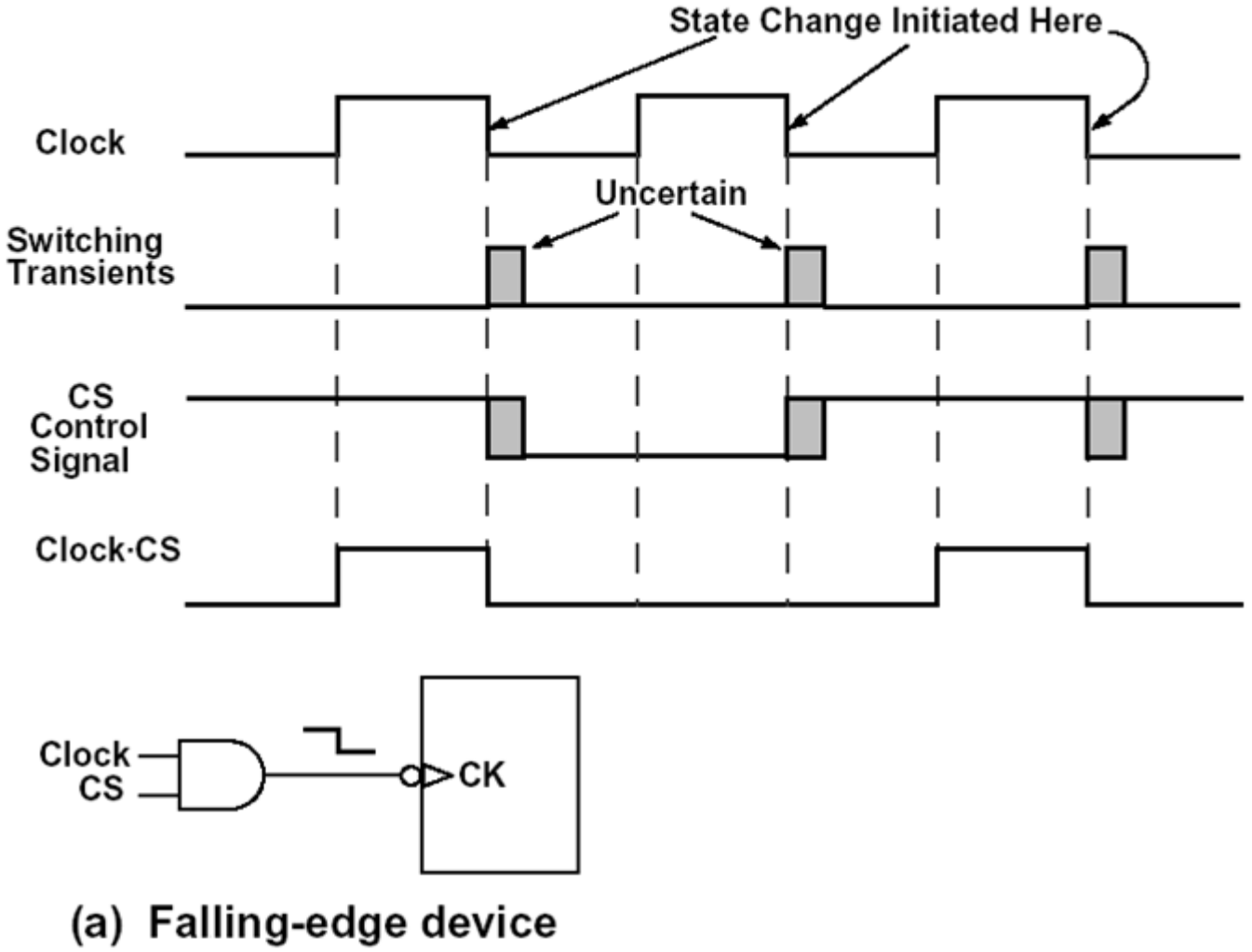


- Move wires from the adder outputs one position to the right => add and shift can occur at the same clock cycle

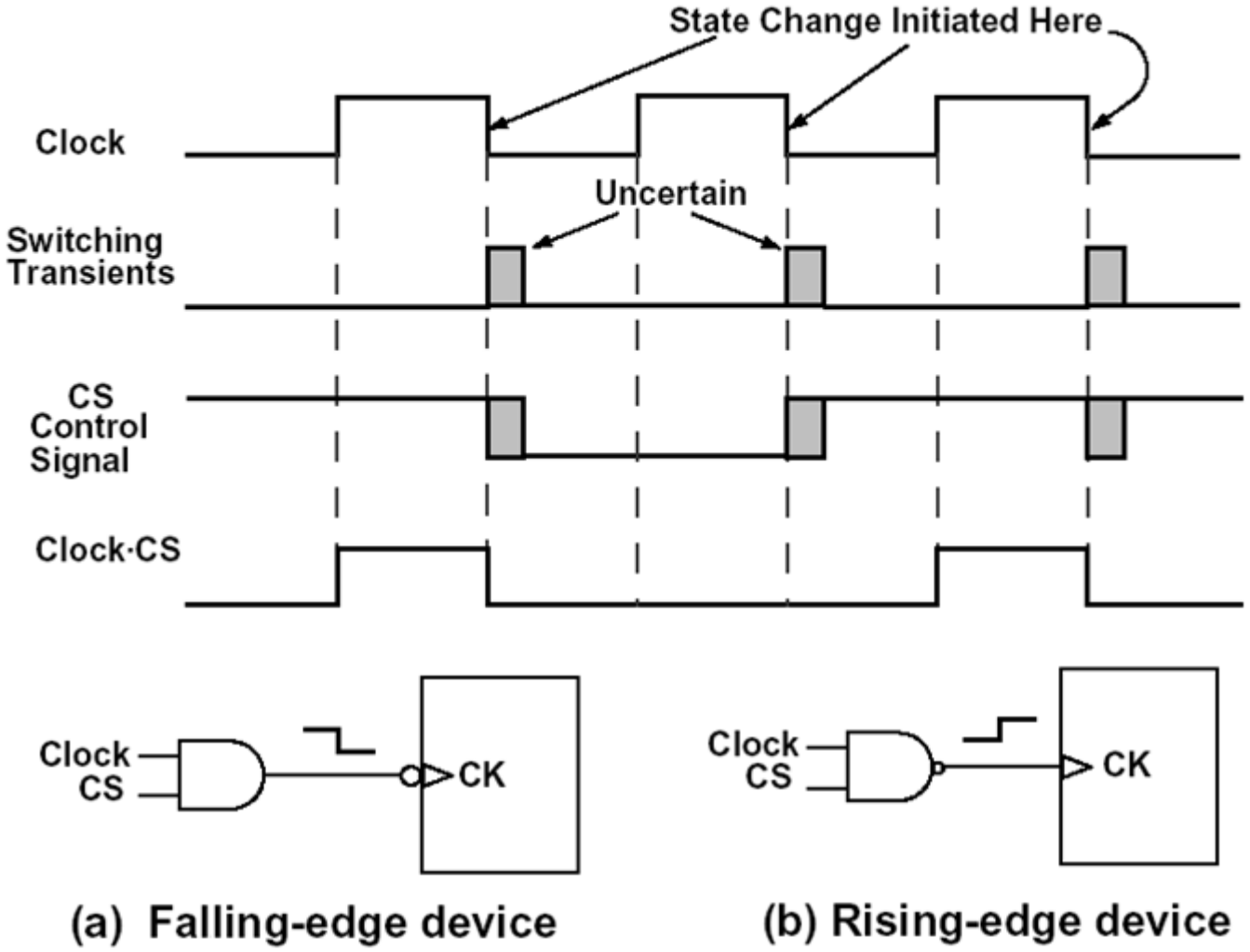
Control Signal Timing Issues

- Change in state of the flip-flops in control section determined by the propagation delay
- Time control signals change depend upon this FF propagation delay and combination network delay
- Glitches and spikes may occur in the control signals due to hazards in the network
- Noise may be introduced on the control signals by changing signals in another part of the circuit
- THIS MEANS THAT THERE IS A TIME INTERVAL AFTER THE ACTIVE EDGE OF THE CLOCK WHERE THE STATE OF THE CONTROL SIGNAL IS NOT KNOWN AND MAY NOT BE STABLE

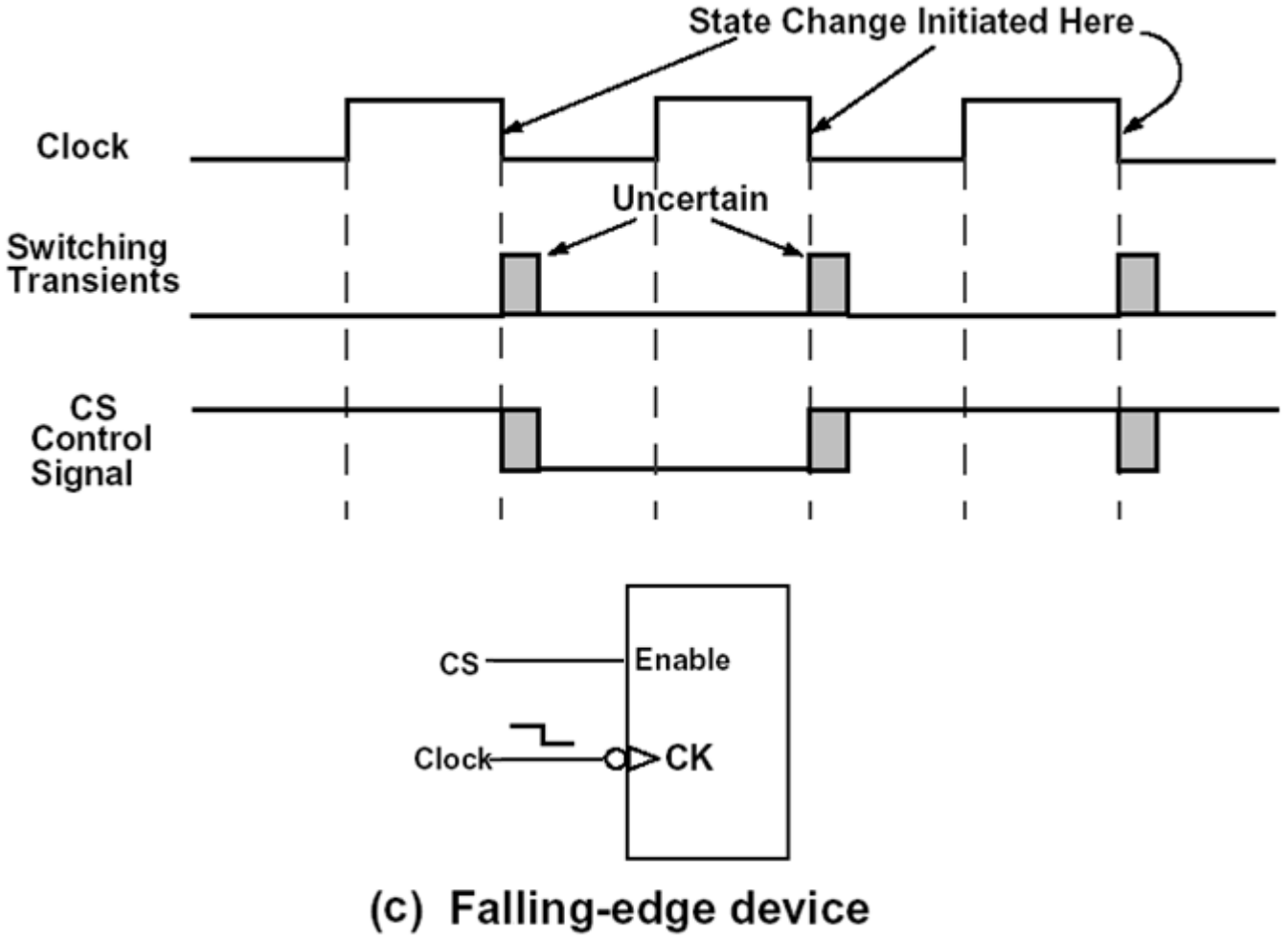
Control Signal Clock Gating when Active Edge is the Falling-Edge



Control Signal Gating when Active Clock Edge is the Falling-Edge



Use of an Enable when Active Clock Edge is the Falling-Edge



(c) Falling-edge device

Control Signal Gating when Active Clock Edge is the Rising-Edge

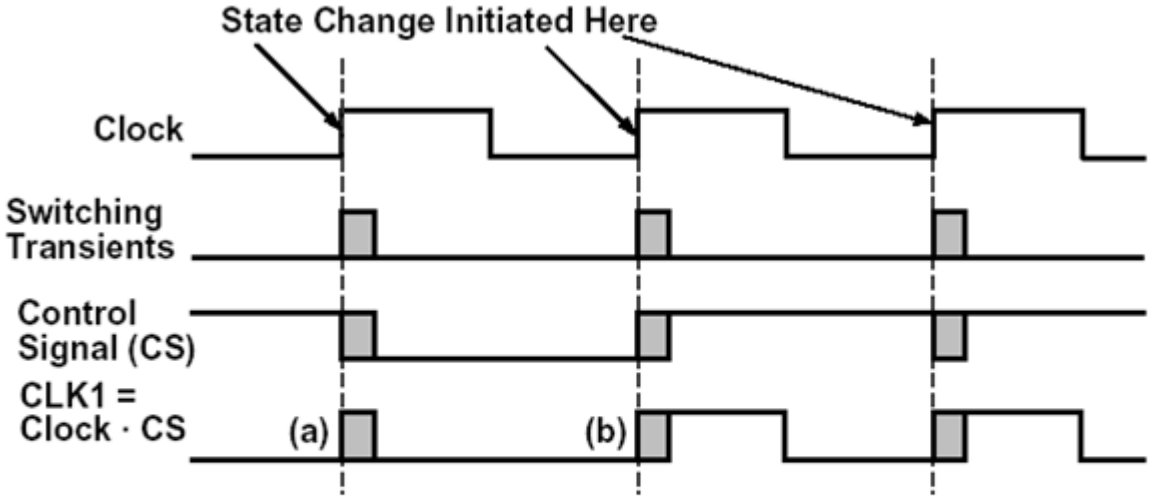
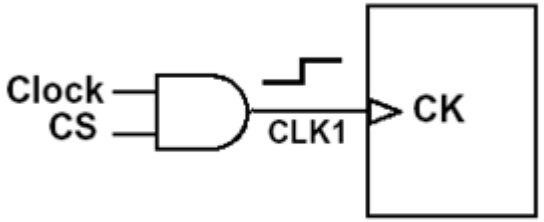


Figure 1-35 Incorrect Design



(a) Rising-edge device

Control Signal Gating when Active Clock Edge is the Rising-Edge

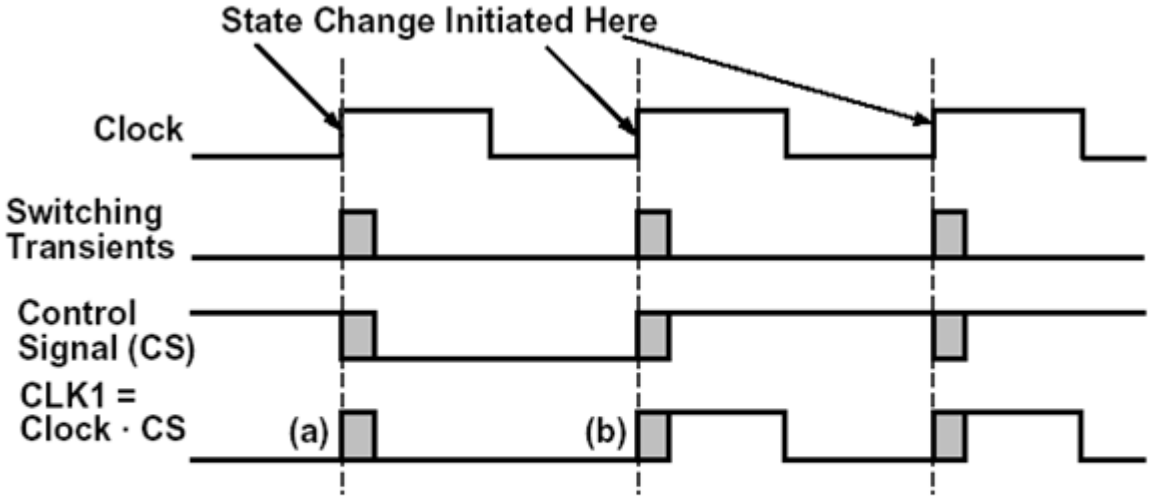
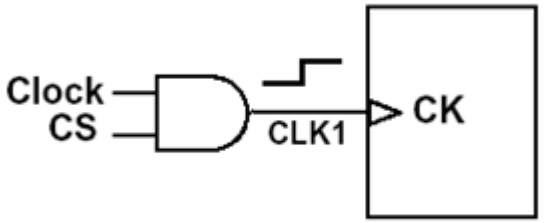
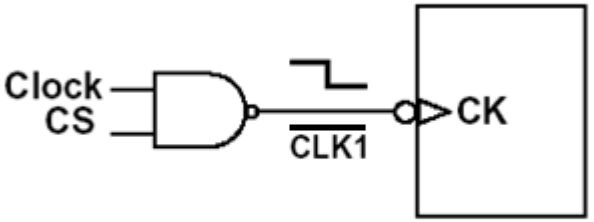


Figure 1-35 Incorrect Design



(a) Rising-edge device

Figure 1-35 Incorrect Design



(b) Falling-edge device

Control Signal Gating when Active Clock Edge is the Rising-Edge

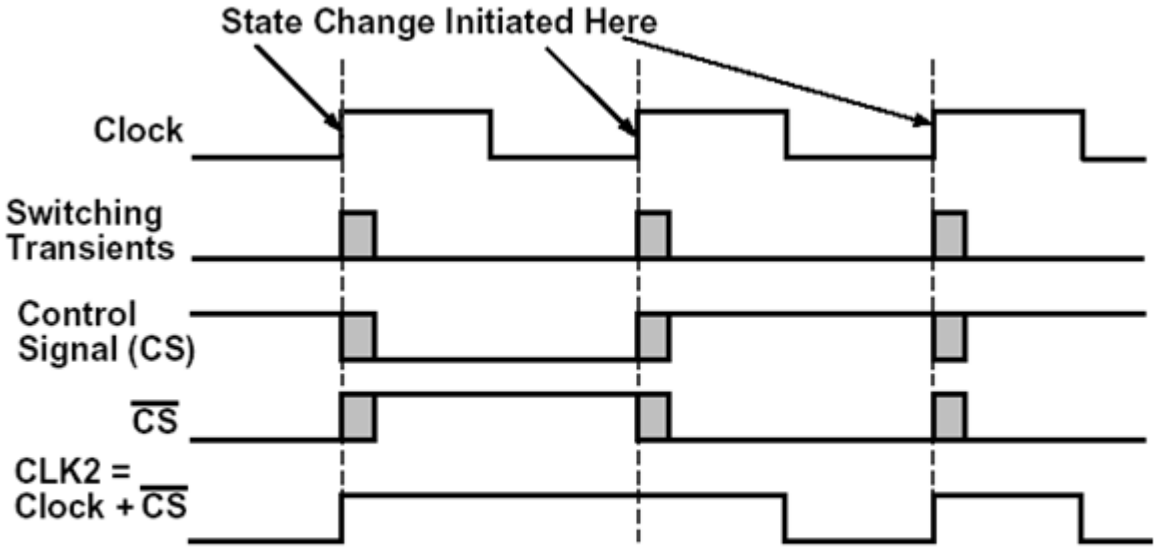
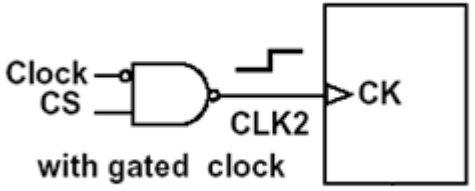


Figure 1-36 Correct Design



(a) Rising-edge device

Control Signal Gating when Active Clock Edge is the Rising-Edge

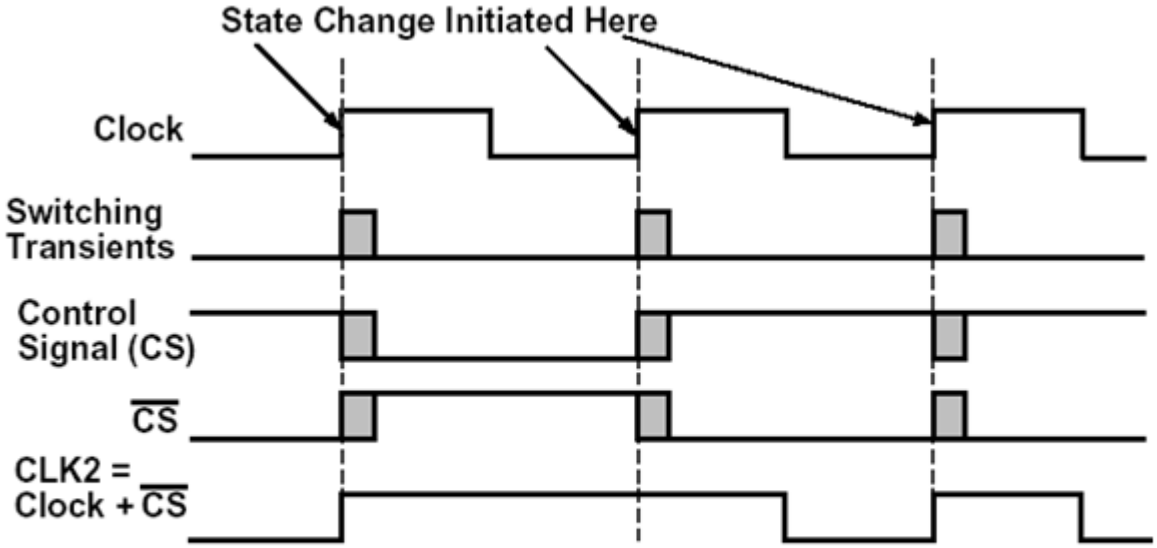
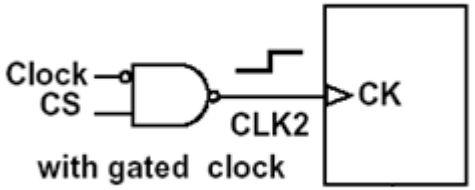
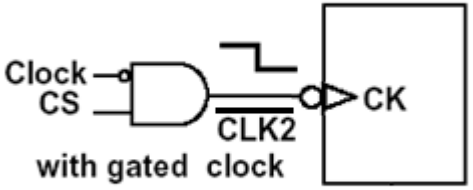


Figure 1-36 Correct Design



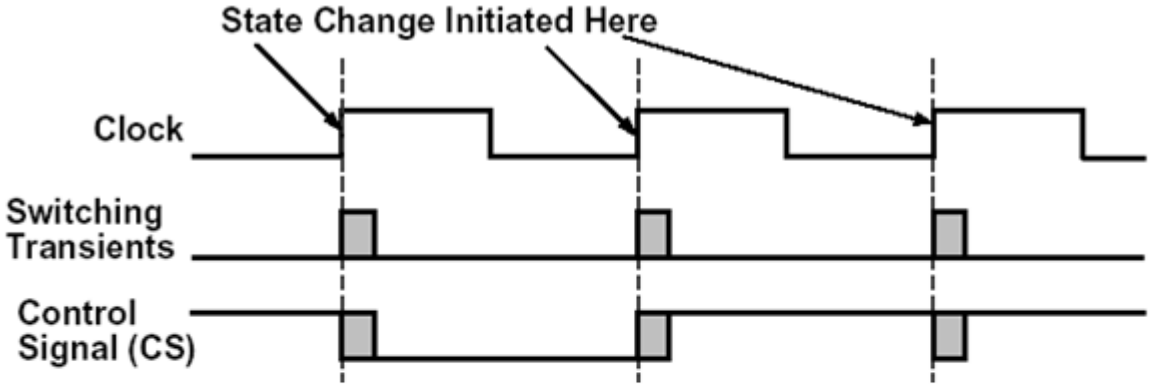
(a) Rising-edge device

Figure 1-36 Correct Design



(b) Falling-edge device

Control Signal Gating when Active Clock Edge is the Rising-Edge



Control Signal Gating when Active Clock Edge is the Rising-Edge

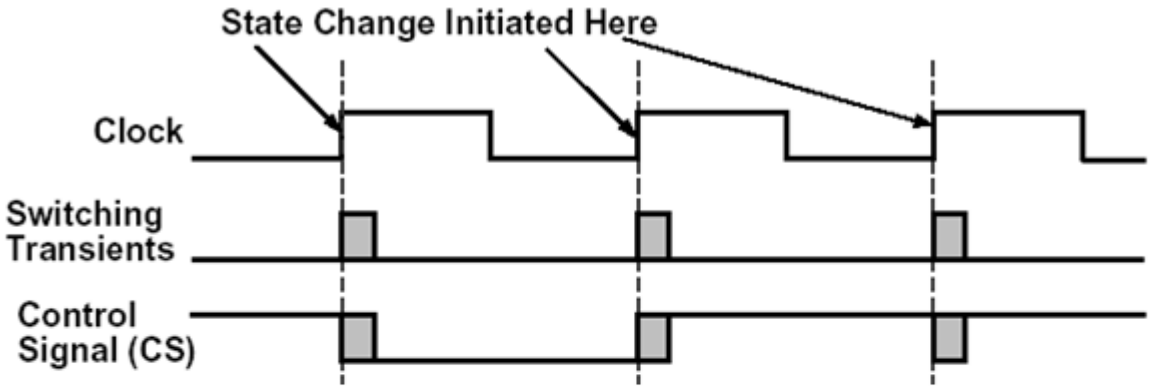
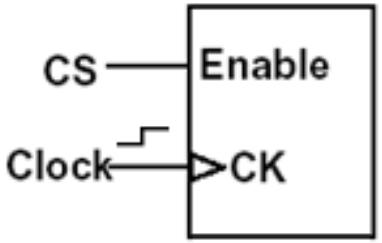
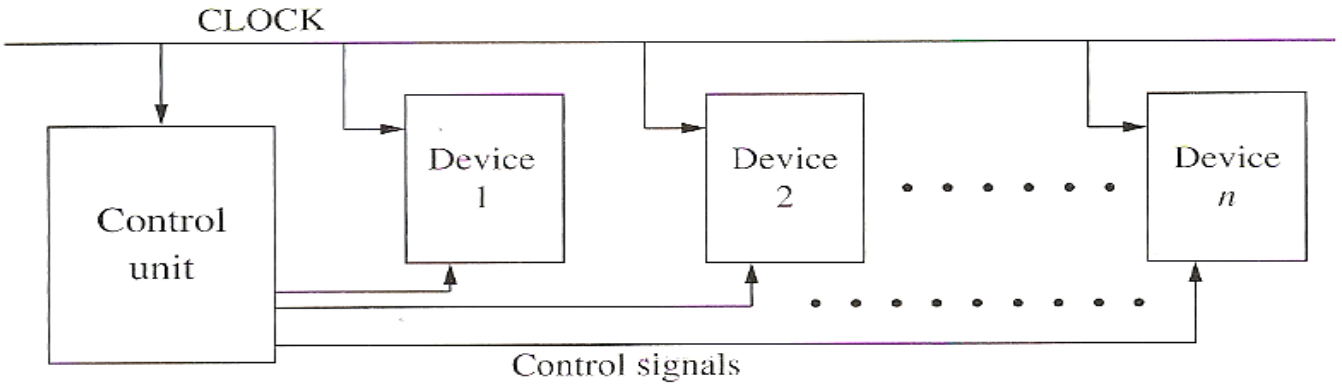


Figure 1-36 Correct Design



(c) Rising-edge device

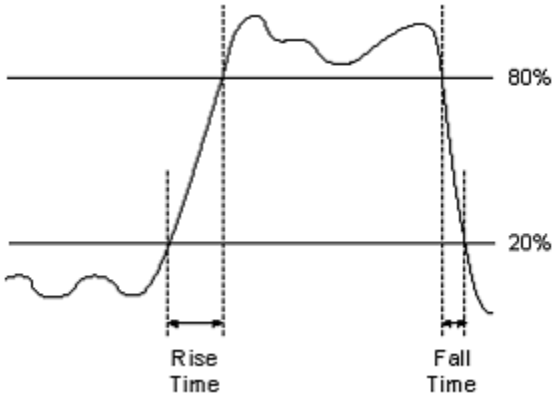
Synchronous Digital Systems



- Use a clock to synchronize the operation of all flip-flops, registers, and counters in the system
  - all changes occur immediately following the active clock edge
  - clock period must be long enough so that all changes flip-flops, registers, counters will have time to stabilize before the next active clock edge
- Control section + Data Section
- Mealy and Moore Machines typically form the control sections of synchronous designs

Rise/Fall Times & Pulse Width

- The rise time or fall time of a signal is usually defined as the time it takes the signal to change between the 10% to 90% values of the transition.

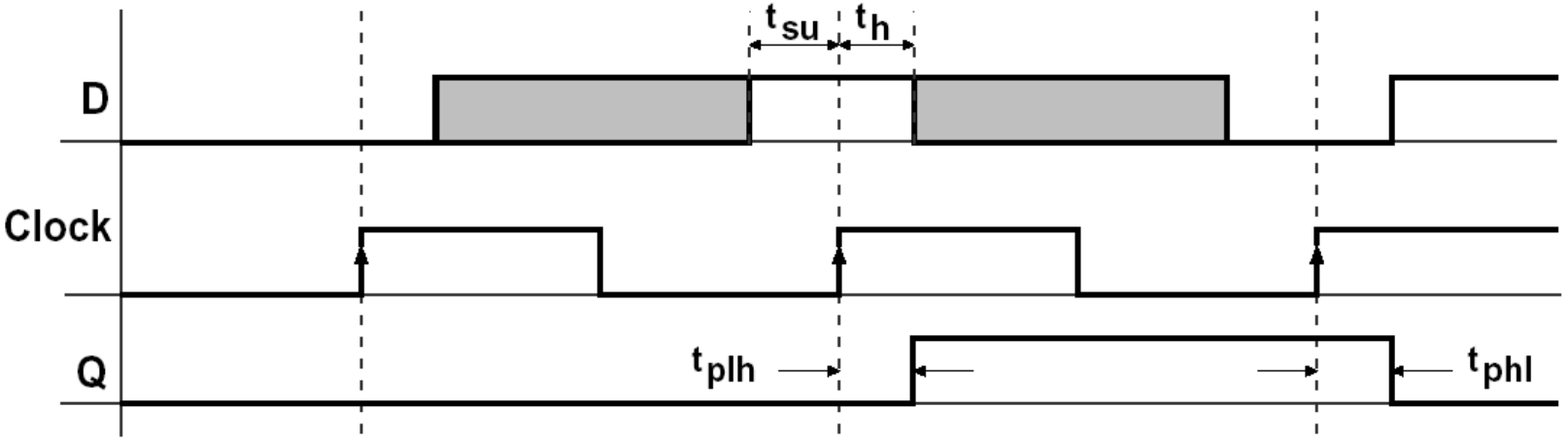


- An ideal timing diagram does not show rise times and fall times and each gate is assumed to have a 0 gate delay.

- The pulse width of a positive pulse is usually defined as the time it takes the signal to go from its 50% value on the rising edge of the pulse to its 50% value on the falling edge of the pulse.

Setup and Hold Times

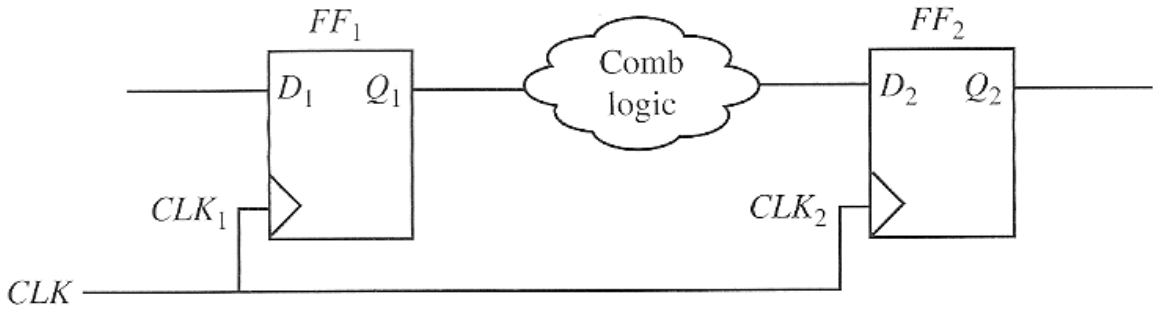
- For a real D-FF
  - D input must be stable for a certain amount of time before the active edge of clock cycle => *Setup time*
  - D input must be stable for a certain amount of time after the active edge of the clock => *Hold time*
- Propagation time: from the time the clock changes to the time the output changes



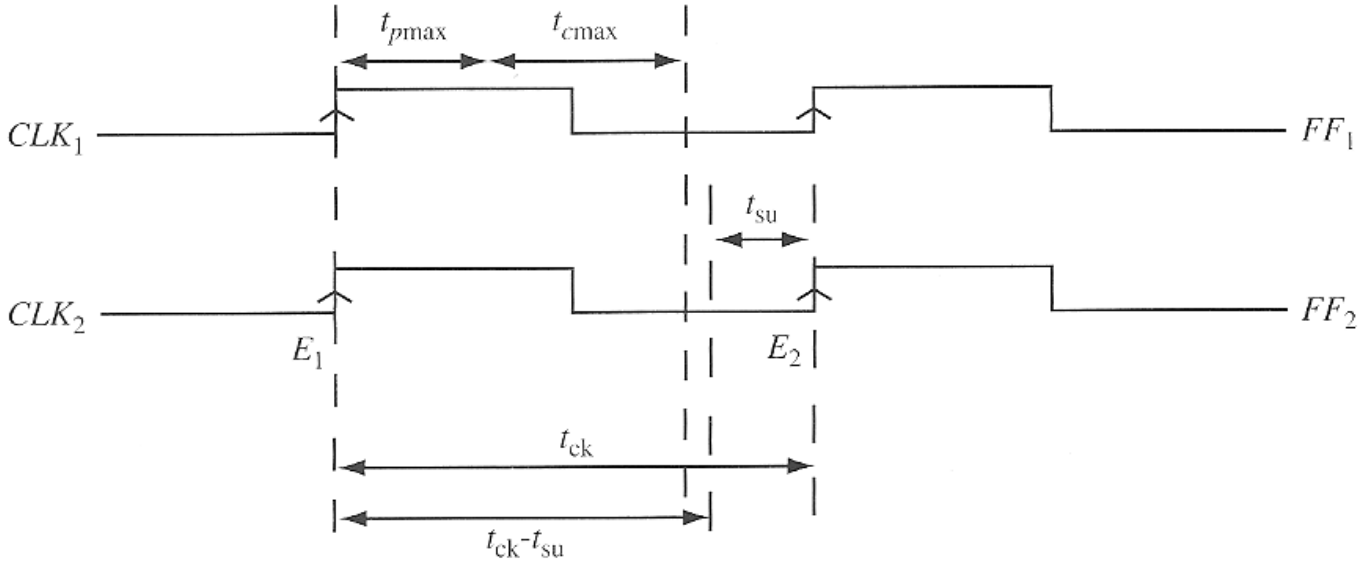
Manufacturers provide minimum t_{su}, th, and maximum tplh, tphl

FF to FF Path via Combinational Logic

(when setup time is met)



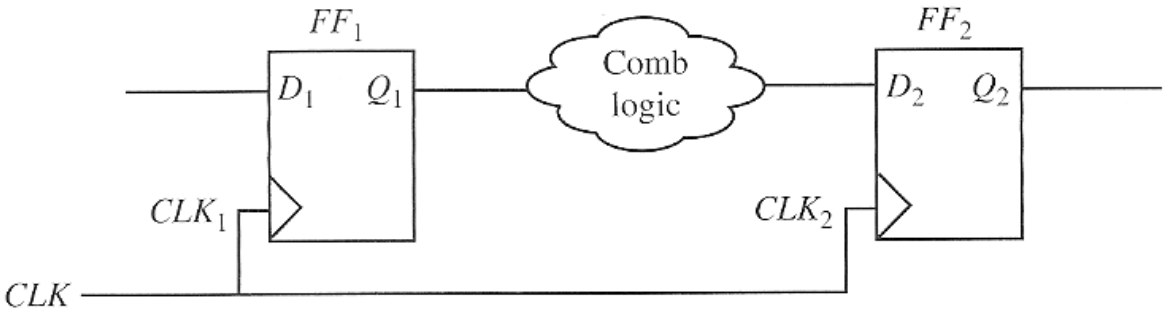
(a) Circiut



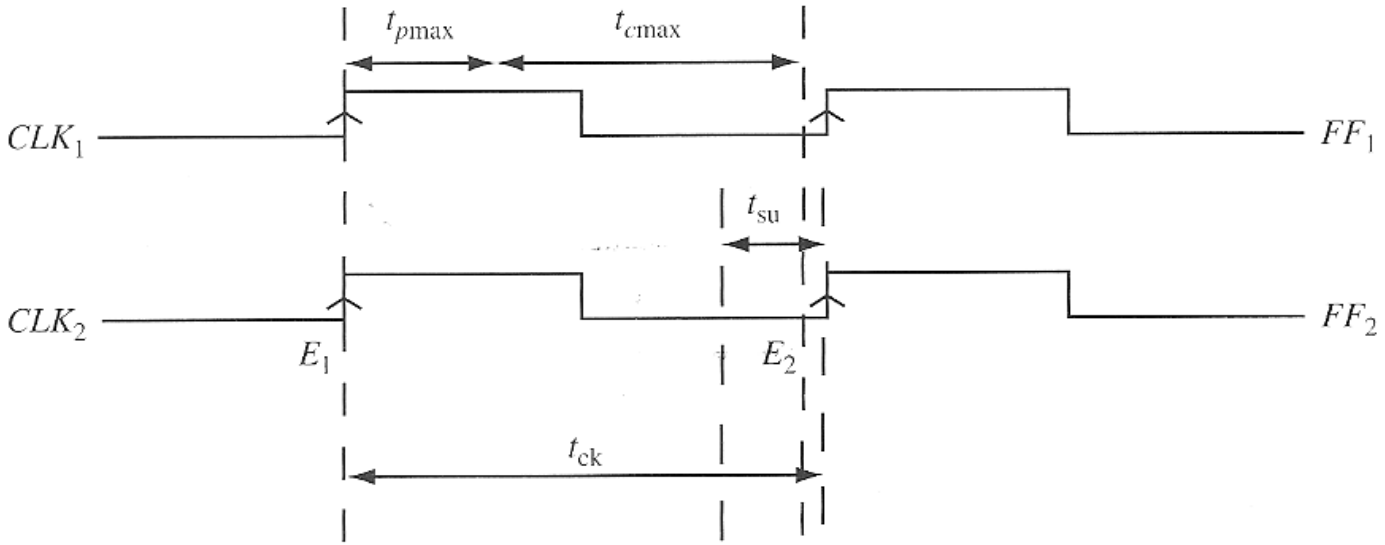
(b) Timing diagram when setup time is met

FF to FF Path via Combinational Logic

(when setup time is not met)



(a) Circiut



(c) Timing diagram when setup time is violated

Maximum Clock Frequency

$t_{c\,max}$-Max propagation delay through the combinational network
(path from register outputs back to register inputs – ignoring inputs)

$t_{p\,max}$- Max propagation delay from the time the clock changes to the flip-flop output changes { = max(tplh, tphl)}

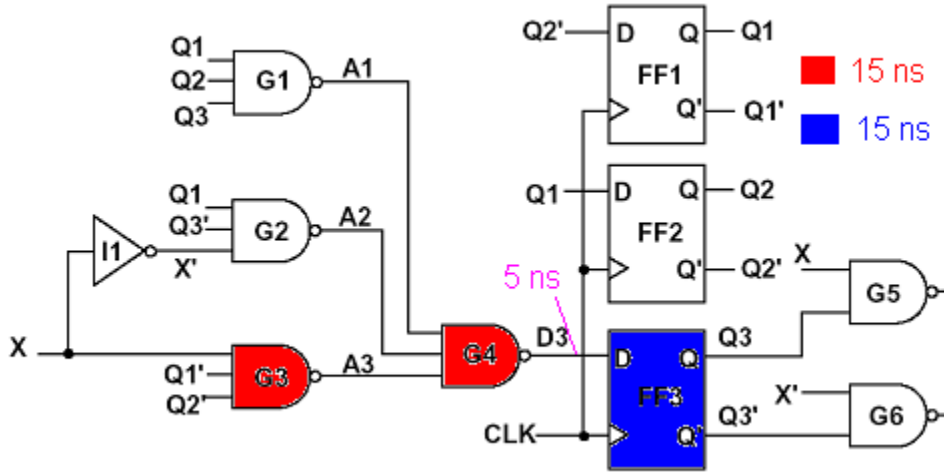
t_{ck} - Clock period

$$t_{ck} \geq t_{c\,max} + t_{p\,max} + t_{su}$$

Example:

$$t_{p\,max} = 15\ ns, t_{su} = 5\ ns,$$

$$t_{gate} = 15\ ns$$



$$t_{ck\,min} = 2 \bullet 15ns + 15ns + 5ns$$

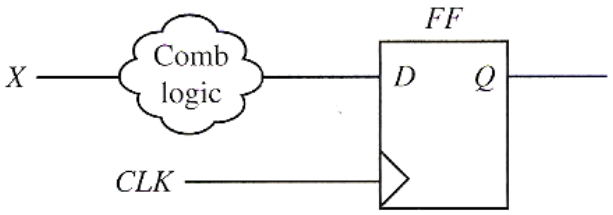
$$= 50ns$$

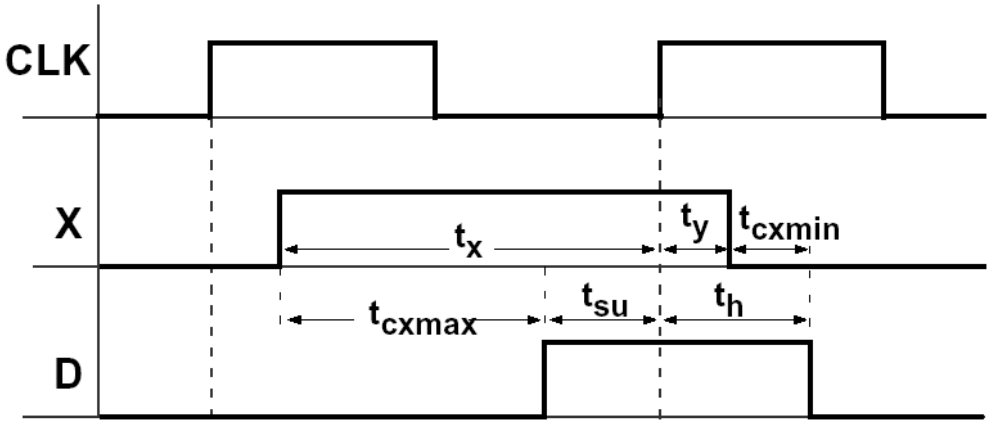
$$f_{ck\,max} = \frac{1}{t_{ck\,min}} = \frac{1}{50ns}$$

$$= 20Mhz$$

Setup Time Violations, Input X point of view

- Occurs if the change in X that is fed forward through the combinational network causes input D to change too late before the clock edge





What about X?

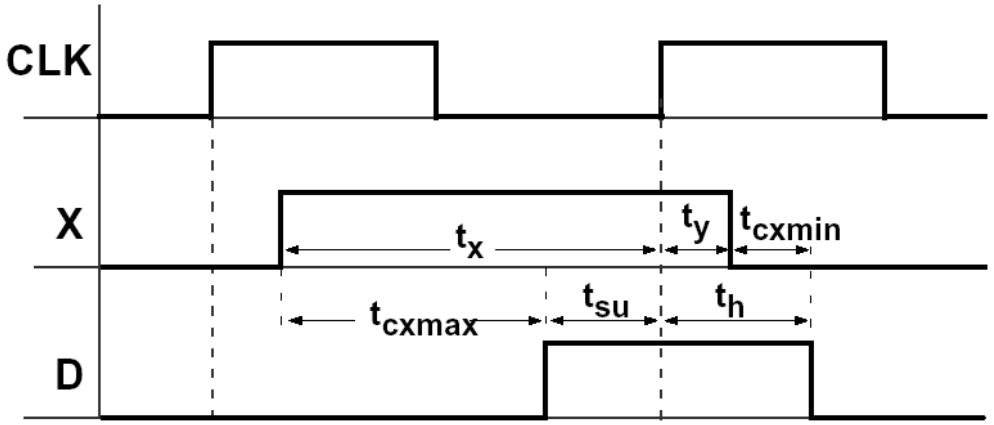
Make sure that input changes propagate to the flip-flops inputs such that setup time is satisfied.

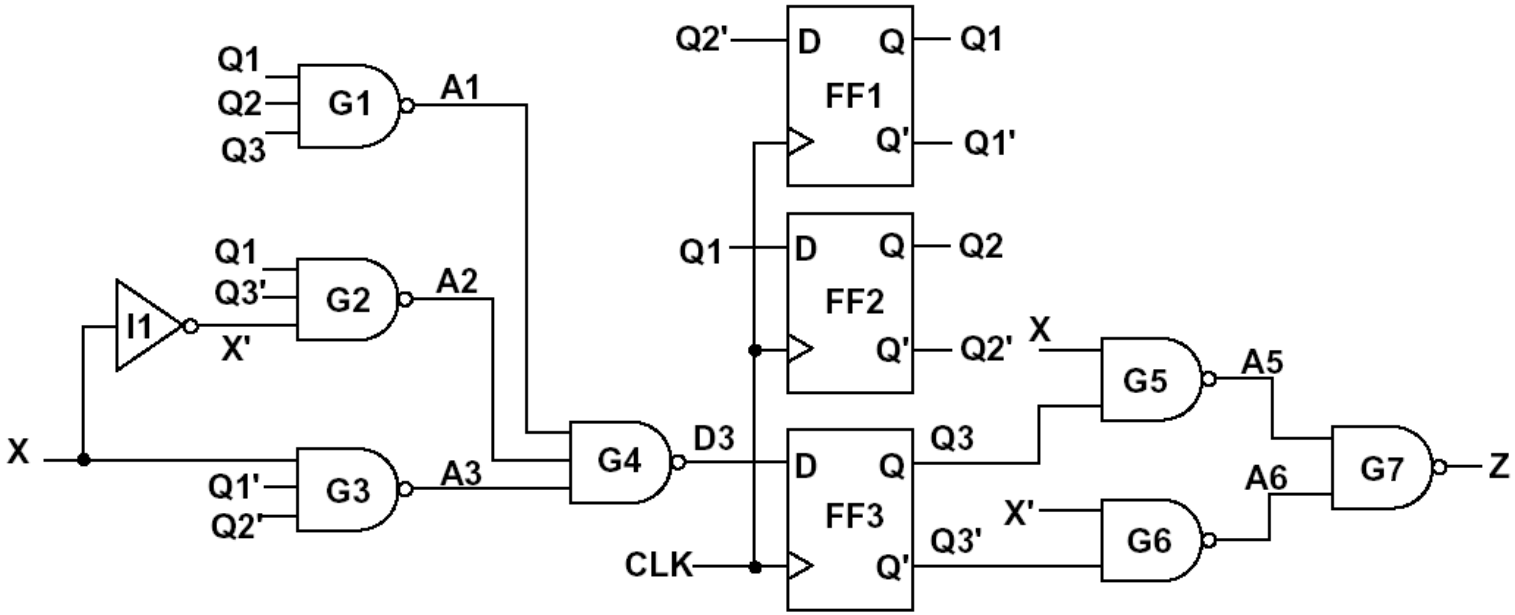
$$t_x \geq t_{cxmax} + t_{su}$$

$t_{cx\max}$ - Max propagation delay through the combinational network from input X (or any other input) to the flip-flop input D

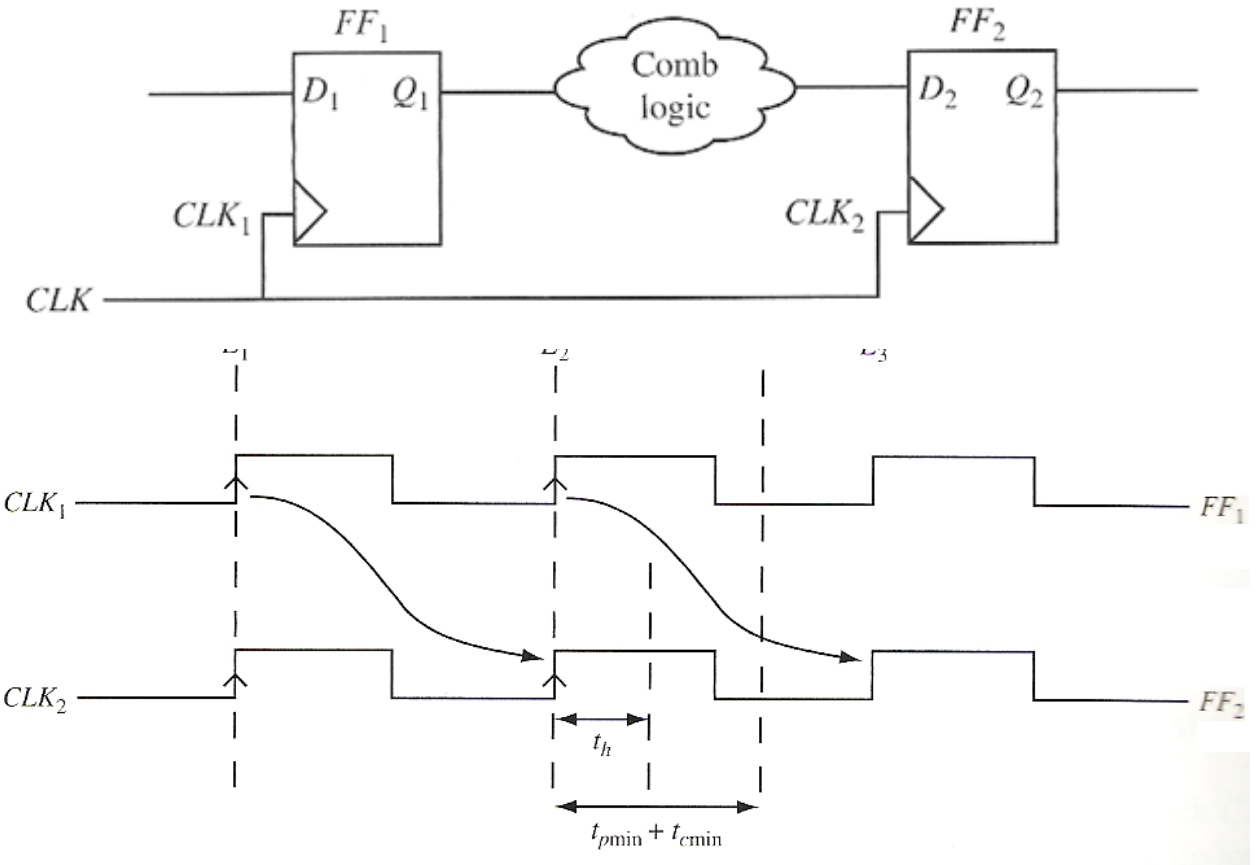
Setup Time Violations, Input X point of view

$$t_x \geq t_{cxmax} + t_{su}$$





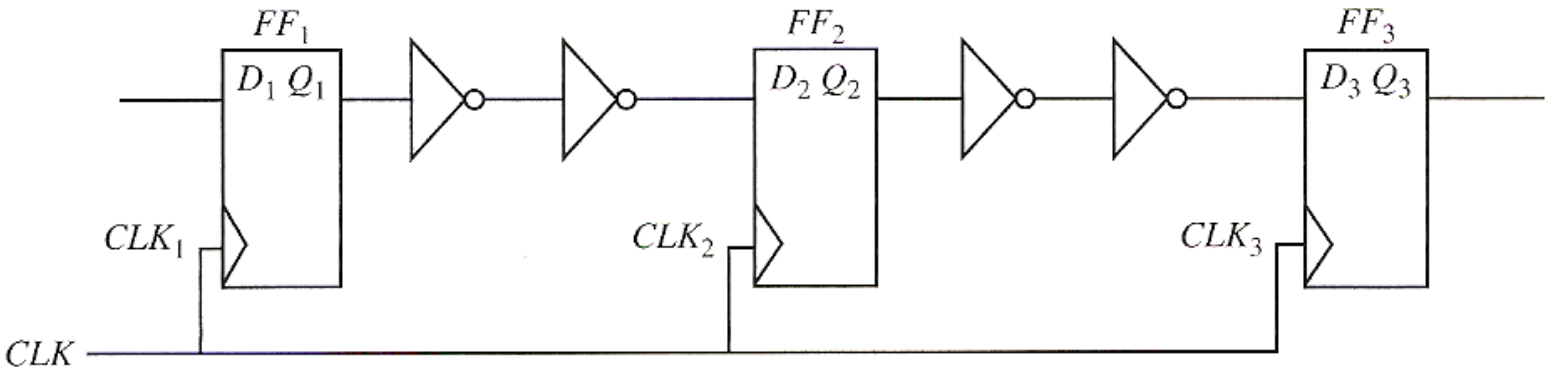
Hold Time Rule for FF



$$t_{pmin} + t_{cmin} \geq t_h$$

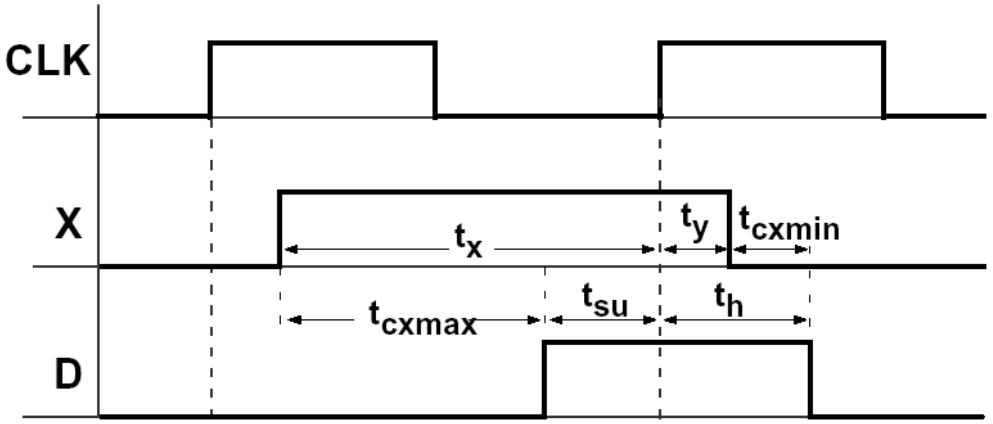
Note: if a circuit has a hold-time violation, it cannot be corrected by changing the clock frequency – must be redesigned by adding combinational delays!

Fixing Hold Time Problems in a Shift Register Design using Buffers



Hold Time Violations
(not considering the Input X)

- Occurs if the change in Q that is fed back through the combinational network causes input D to change too soon after the clock edge



Hold time is satisfied if:

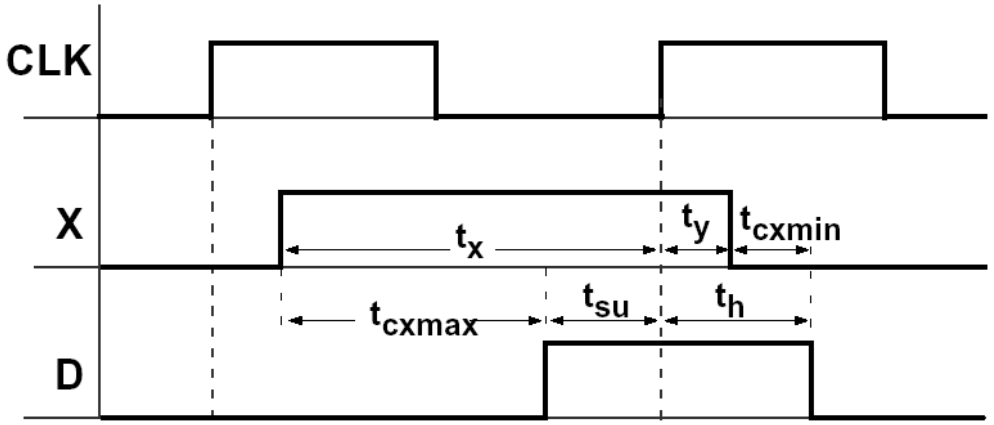
$$t_{p\,min} + t_{c\,min} \geq t_h$$

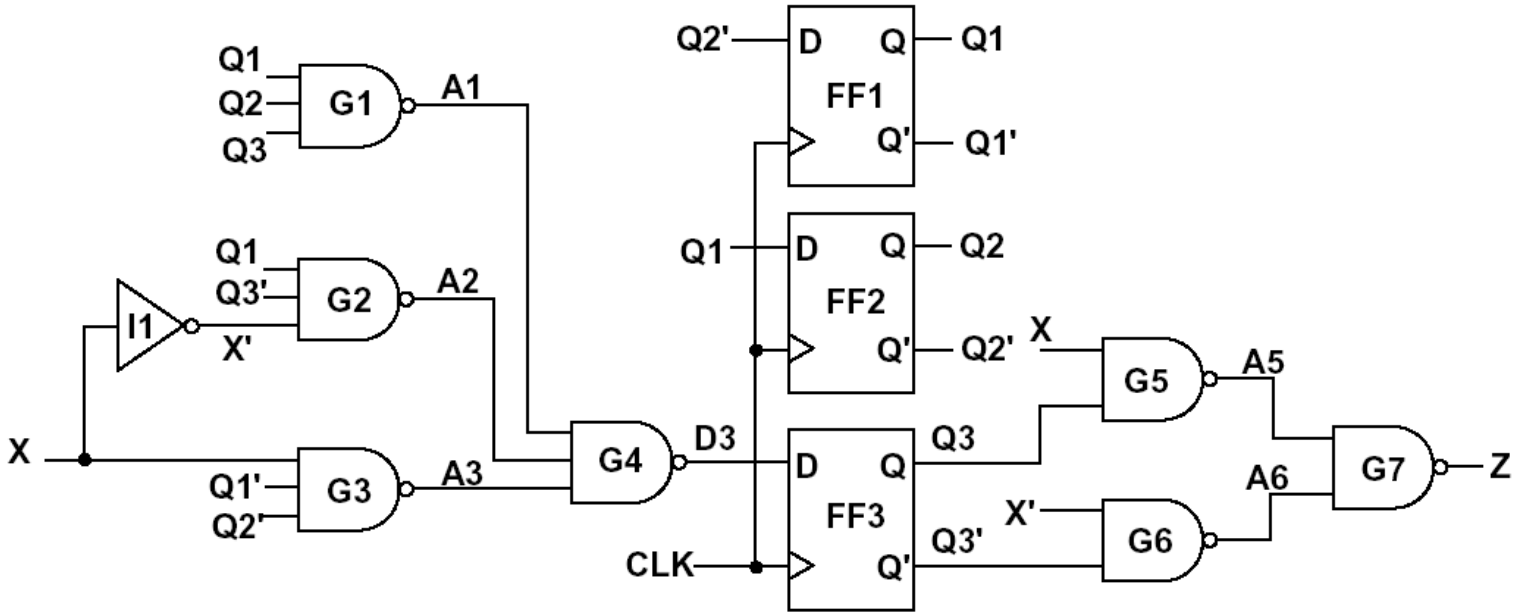
$t_{c\,min}$ - Min propagation delay through the combinational network

$t_{p\,min}$ - Min propagation delay from the time the clock changes to the flip-flop output changes { = min(tplh, tphl)}

Hold Time Violations
(not considering the Input X)

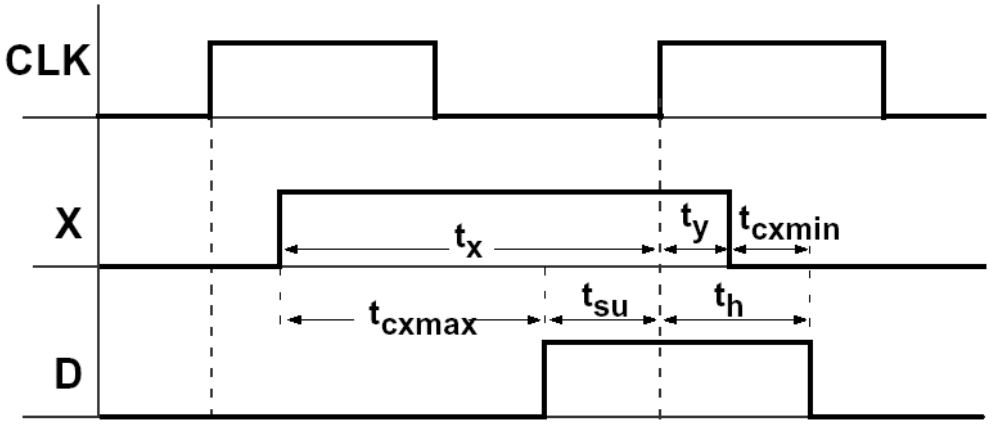
$$t_{p\,min} + t_{c\,min} \geq t_h$$





Hold Time Violations, Input X point of view

- Occurs if the change in X that is fed back through the combinational network causes input D to change too soon after the clock edge



What about X?

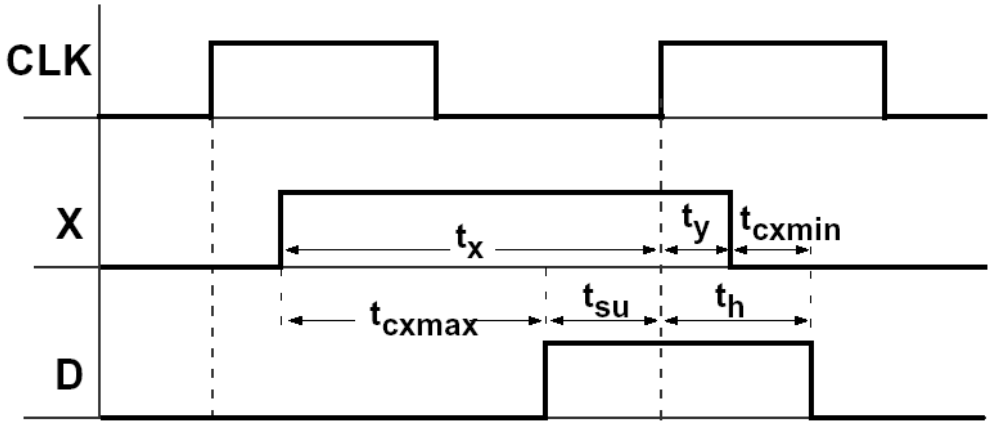
Make sure that X does not change too soon after the clock.
If X changes at time ty after the active edge, hold time is satisfied if

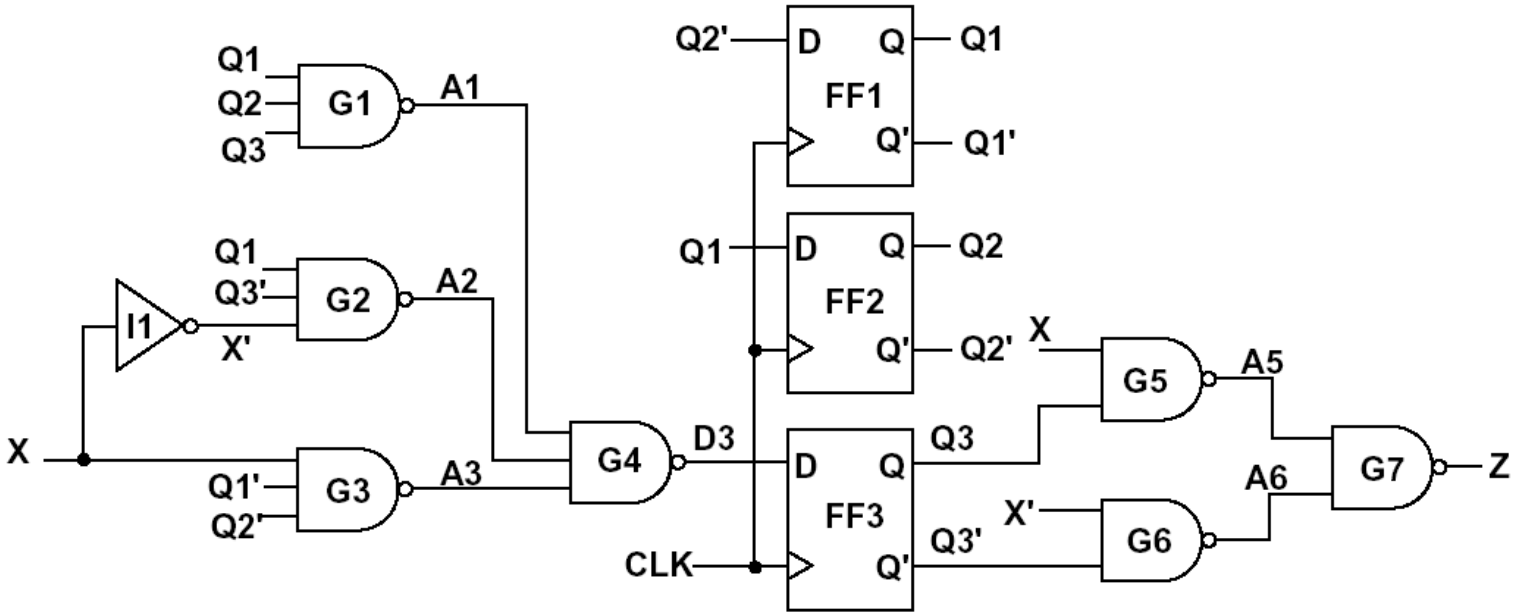
$$t_y \geq t_h - t_{cxmin}$$

$t_{cx\min}$ - Min propagation delay through the combinational network from input X (or any other input) to the flip-flop input D

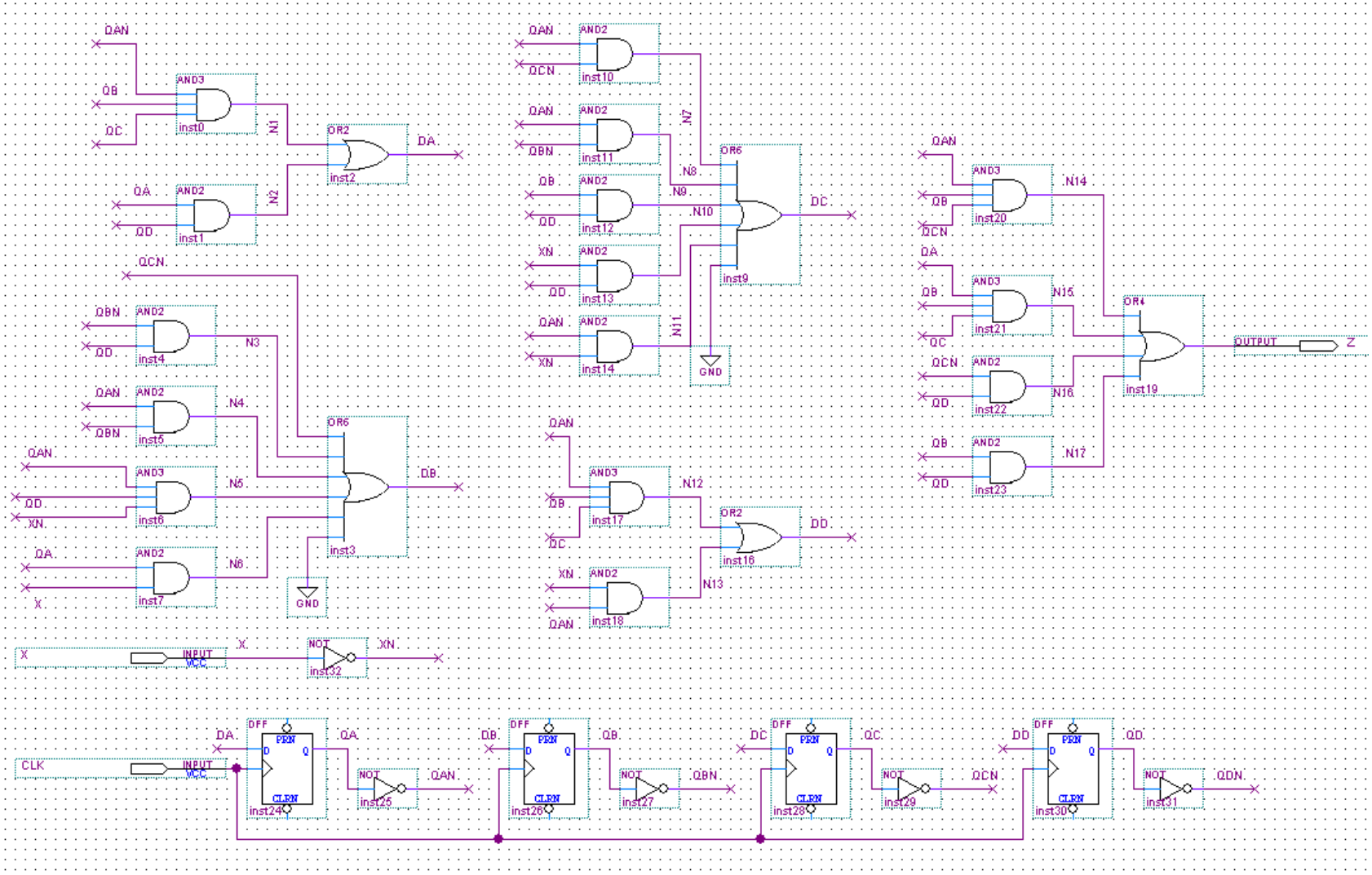
Hold Time Violations, Input X point of view

$$t_y \geq t_h - t_{cxmin}$$





Quartus II Moore Implementation of BCD to Excess 3

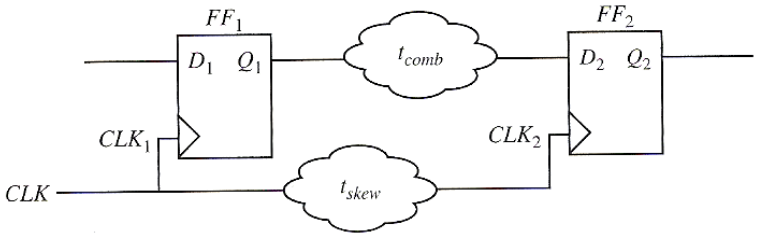


Clock Skew in Synchronous Designs

- **Clock Skew** - absolute time difference in clock signal arrival between two points in the clock network.
  - Often caused by delays in the interconnect within the clock distribution network.
  - Can also be caused by combinational logic used to selectively gate the clock of certain devices
- **Positive Skew** –capturing flip-flop gets the clock delayed with reference to the launching flip-flop
- **Negative Skew** – launching filip-flop get the clock delayed with reference to the capturing flip-flop

Setup & Hold Equations with Clock Skew

Positive Clock Skew



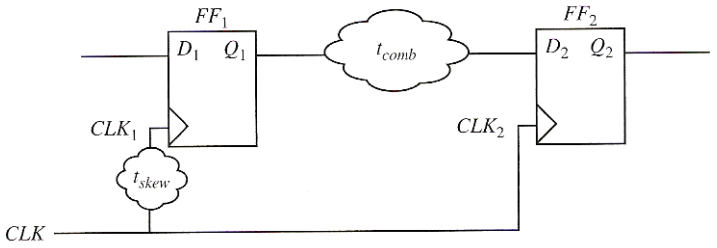
(a) Positive Skew

$$t_{ck} \geq t_{pmax} + t_{cmax} - t_{skew} + t_{su}$$

$$t_{pmin} + t_{cmin} \geq t_h + t_{skew}$$

Positive skew is good for setup time, but it is bad for hold time.

Negative Clock Skew

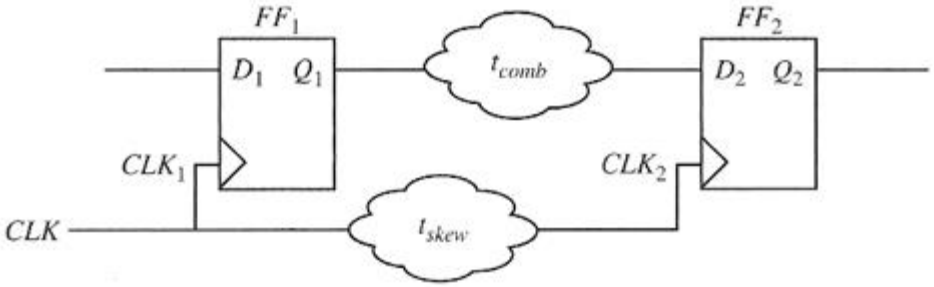


$$t_{ck} \geq t_{pmax} + t_{cmax} + t_{skew} + t_{su}$$

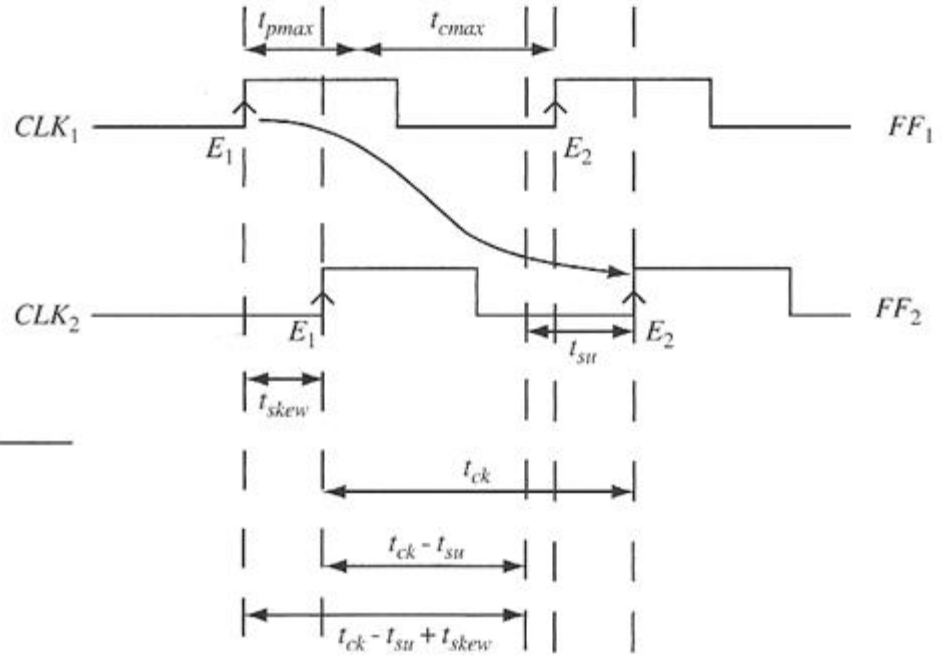
$$t_{pmin} + t_{cmin} \geq t_h - t_{skew}$$

Negative skew is good for hold time, but it is bad for setup time.

Positive Clock Skew – Setup Time



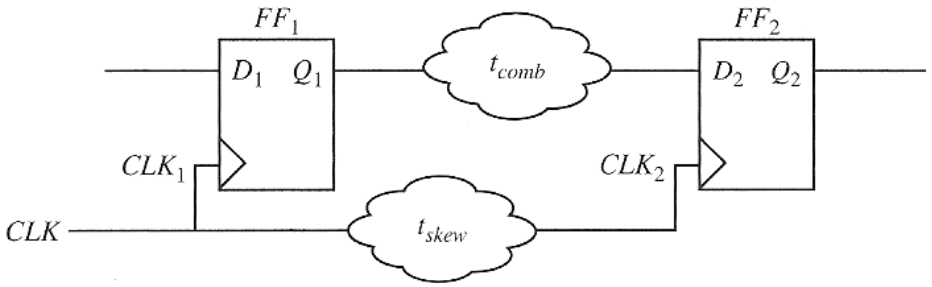
(a) Positive Skew



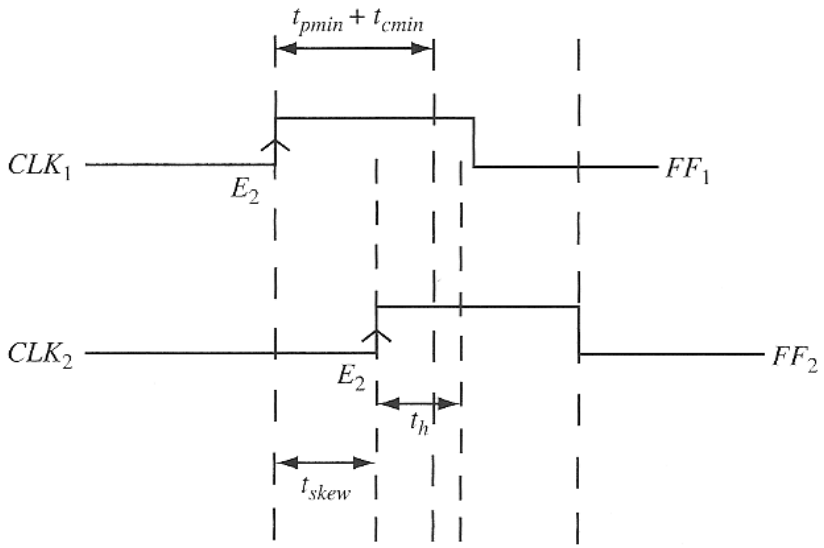
Condition $t_{pmax} + t_{cmax} \leq t_{ck} - t_{su} + t_{skew}$ not satisfied

(b) Setup-Time Violation

Positive Clock Skew – Hold Time



(a) Positive Skew



Condition $t_{pmin} + t_{cmin} \geq t_{skew} + t_{hold}$ not satisfied

(c) Hold-Time Violation

Timing Analysis Example

Timing Analyzer Summary

Type	Slack	Required Time	Actual Time	From	To
Worst-case tsu	N/A	None	11.900 ns	X	inst26
Worst-case tco	N/A	None	19.100 ns	inst24	Z
Worst-case th	N/A	None	-1.700 ns	X	inst28
Clock Setup: 'CLK'	N/A	None	93.46 MHz (period = 10.700 ns)	inst28	inst26
Total number of failed paths					

Timing Analyzer Settings

Option	Setting	From	To	Entity Name
Device Name	EPF10K70RC240-4			
Timing Models	Final			
Default hold multicycle	Same as Multicycle			
Cut paths between unrelated clock domains	On			
Cut off read during write signal paths	On			
Cut off feedback from I/O pins	On			
Report Combined Fast/Slow Timing	Off			
Ignore Clock Settings	Off			
Analyze latches as synchronous elements	On			
Enable Recovery/Removal analysis	Off			
Enable Clock Latency	Off			
Number of source nodes to report per destination node	10			
Number of destination nodes to report	10			
Number of paths to report	200			
Report Minimum Timing Checks	Off			
Use Fast Timing Models	Off			
Report IO Paths Separately	Off			

Setup/Combinational/Hold Time Report

tsu

Slack	Required tsu	Actual tsu	From	To	To Clock
N/A	None	11.900 ns	X	inst26	CLK
N/A	None	10.900 ns	X	inst28	CLK
N/A	None	7.700 ns	X	inst30	CLK

tco

Slack	Required tco	Actual tco	From	To	From Clock
N/A	None	19.100 ns	inst26	Z	CLK
N/A	None	19.100 ns	inst30	Z	CLK
N/A	None	19.100 ns	inst24	Z	CLK
N/A	None	18.800 ns	inst28	Z	CLK

th

Minimum Slack	Required th	Actual th	From	To	To Clock
N/A	None	-1.700 ns	X	inst28	CLK
N/A	None	-2.000 ns	X	inst30	CLK
N/A	None	-4.600 ns	X	inst26	CLK

```
Warning: Found pins functioning as undefined clocks and/or memory enables
    Info: Assuming node "CLK" is an undefined clock
Info: Clock "CLK" has Internal fmax of 93.46 MHz between source register "inst24" and destination register "inst26" (period= 10.7 ns)
    Info: + Longest register to register delay is 6.700 ns
        Info: 1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = LC6_D30; Fanout = 6; REG Node = 'inst24'
        Info: 2: + IC(0.500 ns) + CELL(2.000 ns) = 2.500 ns; Loc. = LC4_D30; Fanout = 1; COMB Node = 'inst3~74'
        Info: 3: + IC(0.000 ns) + CELL(2.000 ns) = 4.500 ns; Loc. = LC5_D30; Fanout = 1; COMB Node = 'inst3~71'
        Info: 4: + IC(0.500 ns) + CELL(1.700 ns) = 6.700 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
        Info: Total cell delay = 5.700 ns ( 85.07 % )
        Info: Total interconnect delay = 1.000 ns ( 14.93 % )
    Info: - Smallest clock skew is 0.000 ns
        Info: + Shortest clock path from clock "CLK" to destination register is 7.000 ns
            Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_91; Fanout = 4; CLK Node = 'CLK'
            Info: 2: + IC(4.100 ns) + CELL(0.000 ns) = 7.000 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
            Info: Total cell delay = 2.900 ns ( 41.43 % )
            Info: Total interconnect delay = 4.100 ns ( 58.57 % )
        Info: - Longest clock path from clock "CLK" to source register is 7.000 ns
            Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_91; Fanout = 4; CLK Node = 'CLK'
            Info: 2: + IC(4.100 ns) + CELL(0.000 ns) = 7.000 ns; Loc. = LC6_D30; Fanout = 6; REG Node = 'inst24'
            Info: Total cell delay = 2.900 ns ( 41.43 % )
            Info: Total interconnect delay = 4.100 ns ( 58.57 % )
    Info: + Micro clock to output delay of source is 1.400 ns
    Info: + Micro setup delay of destination is 2.600 ns
Info: tsu for register "inst26" (data pin = "X", clock pin = "CLK") is 11.900 ns
    Info: + Longest pin to register delay is 16.300 ns
        Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_212; Fanout = 5; PIN Node = 'X'
        Info: 2: + IC(7.200 ns) + CELL(2.000 ns) = 12.100 ns; Loc. = LC4_D30; Fanout = 1; COMB Node = 'inst3~74'
        Info: 3: + IC(0.000 ns) + CELL(2.000 ns) = 14.100 ns; Loc. = LC5_D30; Fanout = 1; COMB Node = 'inst3~71'
        Info: 4: + IC(0.500 ns) + CELL(1.700 ns) = 16.300 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
        Info: Total cell delay = 8.600 ns ( 52.76 % )
        Info: Total interconnect delay = 7.700 ns ( 47.24 % )
    Info: + Micro setup delay of destination is 2.600 ns
    Info: - Shortest clock path from clock "CLK" to destination register is 7.000 ns
        Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_91; Fanout = 4; CLK Node = 'CLK'
        Info: 2: + IC(4.100 ns) + CELL(0.000 ns) = 7.000 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
        Info: Total cell delay = 2.900 ns ( 41.43 % )
        Info: Total interconnect delay = 4.100 ns ( 58.57 % )
```

```
Info: tco from clock "CLK" to destination pin "Z" through register "inst26" is 19.100 ns
    Info: + Longest clock path from clock "CLK" to source register is 7.000 ns
        Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_91; Fanout = 4; CLK Node = 'CLK'
        Info: 2: + IC(4.100 ns) + CELL(0.000 ns) = 7.000 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
        Info: Total cell delay = 2.900 ns ( 41.43 % )
        Info: Total interconnect delay = 4.100 ns ( 58.57 % )
    Info: + Micro clock to output delay of source is 1.400 ns
    Info: + Longest register to pin delay is 10.700 ns
        Info: 1: + IC(0.000 ns) + CELL(0.000 ns) = 0.000 ns; Loc. = LC1_D30; Fanout = 6; REG Node = 'inst26'
        Info: 2: + IC(0.500 ns) + CELL(2.700 ns) = 3.200 ns; Loc. = LC3_D30; Fanout = 1; COMB Node = 'inst19~72'
        Info: 3: + IC(2.500 ns) + CELL(5.000 ns) = 10.700 ns; Loc. = PIN_86; Fanout = 0; PIN Node = 'Z'
        Info: Total cell delay = 7.700 ns ( 71.96 % )
        Info: Total interconnect delay = 3.000 ns ( 28.04 % )
Info: th for register "inst28" (data pin = "X", clock pin = "CLK") is -1.700 ns
    Info: + Longest clock path from clock "CLK" to destination register is 7.000 ns
        Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_91; Fanout = 4; CLK Node = 'CLK'
        Info: 2: + IC(4.100 ns) + CELL(0.000 ns) = 7.000 ns; Loc. = LC7_D30; Fanout = 5; REG Node = 'inst28'
        Info: Total cell delay = 2.900 ns ( 41.43 % )
        Info: Total interconnect delay = 4.100 ns ( 58.57 % )
    Info: + Micro hold delay of destination is 3.100 ns
    Info: - Shortest pin to register delay is 11.800 ns
        Info: 1: + IC(0.000 ns) + CELL(2.900 ns) = 2.900 ns; Loc. = PIN_212; Fanout = 5; PIN Node = 'X'
        Info: 2: + IC(7.200 ns) + CELL(1.700 ns) = 11.800 ns; Loc. = LC7_D30; Fanout = 5; REG Node = 'inst28'
        Info: Total cell delay = 4.600 ns ( 38.98 % )
        Info: Total interconnect delay = 7.200 ns ( 61.02 % )
Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 1 warning
    Info: Allocated 100 megabytes of memory during processing
    Info: Processing ended: Tue Feb 20 08:57:44 2007
    Info: Elapsed time: 00:00:00
```

Metastability Issues

- Metastability in digital systems can occur when the data input to a flip-flop is asynchronous to the clock, which can lead to setup or hold time violations.
  - It can cause flip-flops to switch late or not at all.
  - It can present a brief pulse at a flip-flop output (called a runt pulse) or cause flip-flop output oscillations.
  - This can cause system failures

- Source: Xilinx Application notes: XAPP077 January, 1997

Metastability Issues

- To ensure reliable operation, the input to a register must be stable for a minimum time before the clock edge (register setup time or t_{SU}) and for a minimum time after the clock edge (register hold time or t_H).
- The register output then is available after a specified clock-to-output delay (t_{CO}).
- If a data signal transition violates a register's t_{SU} or t_H requirements, the output of the register may go into a metastable state.
- In a metastable state, the register output hovers at a value between the high and low states for some arbitrary period of time
- This means the output transition to a defined high or low state is delayed beyond the specified t_{CO}.

- Source: Altera White paper, "Understanding Metastability FPGAs" http://www.altera.com/literature/wp/wp-01082-quartus-ii-metastability.pdf, 2009

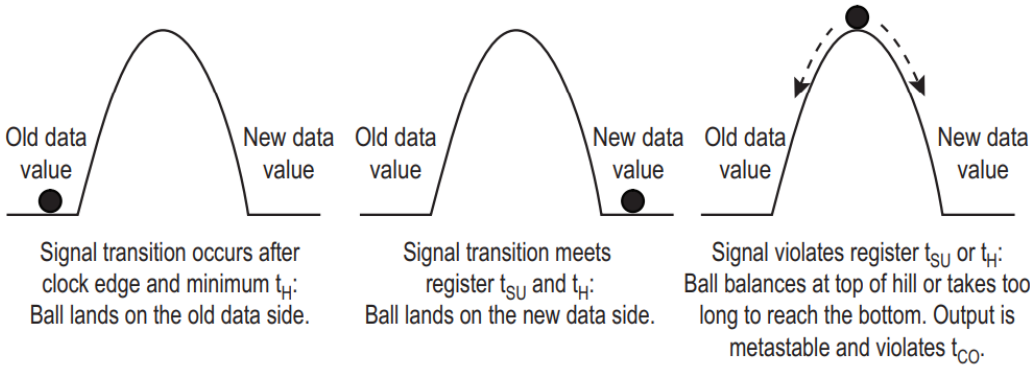
Metastability Issues

- Not every signal transition that violates a register's t_{SU} or t_H results in a metastable output.
- More often than not, these timing violations result in nondeterministic but stable output from the register.
- The likelihood that a register enters a metastable state and the time required to return to a stable state vary depending on the process technology used to manufacture the device and on the operating conditions.

Metastability Issues

- Metastability does not always cause a problem.
  - If the data output signal resolves to a valid state before the next register captures the data, then the metastable signal does not negatively impact the system operation.
  - But if the metastable signal does not resolve to a low or high state before it reaches the next design register, it can cause the system to fail.

Figure 1. Metastability Illustrated as a Ball Dropped on a Hill



Source: Intel (Altera) WP 01082

Metastability Issues

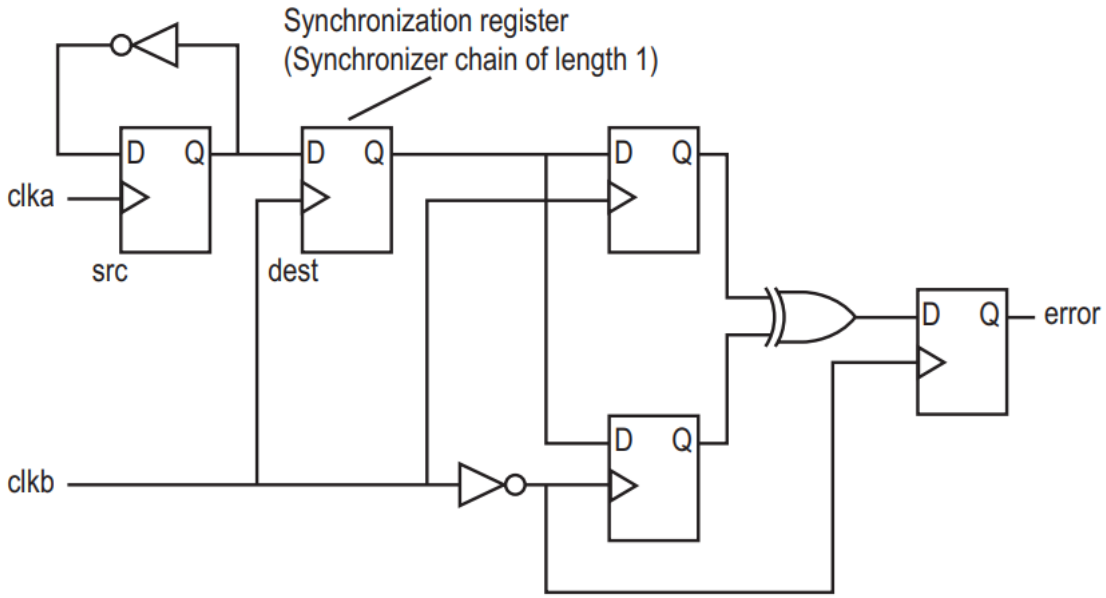
- In properly designed synchronous systems, the input signals must always meet the register timing requirements, so metastability does not occur.
- Metastability can occur when
  - When the input signal is an asynchronous signal.
  - When the clock skew is too large (rise and fall time are more than the tolerable values).
  - When interfacing two domains operating at two different frequencies or at the same frequency but with different phase.
  - When the combinational delay is such that flip-flop data input changes in the critical window (setup or hold time window)

Source: ASIC World, “What is Metastability”
http://www.asic-world.com/tidbits/metastablity.html

Metastability Issues

- In safety critical applications it may be necessary to quantify the risks involved.
  - Mean Time Between Failures is the accepted metric

Figure 4. Test Circuit Structure for Metastability Characterization



Source: Intel (Altera) WP 01082

Determining if Metastability is a problem
MTBF Calculation

The metastability MTBF for a specific signal transfer, or all the transfers in a design, can be calculated using information about the design and the device characteristics. The MTBF of a synchronizer chain is calculated with the following formula and parameters:

$$MTBF = \frac{e^{t_{MET}/C_2}}{C_1 \cdot f_{CLK} \cdot f_{DATA}}$$

The C_1 and C_2 constants depend on the device process and operating conditions.

The f_{CLK} and f_{DATA} parameters depend on the design specifications: f_{CLK} is the clock frequency of the clock domain receiving the asynchronous signal and f_{DATA} is the toggling frequency of the asynchronous input data signal. Faster clock frequencies and faster-toggling data reduce (or worsen) the MTBF.

The t_{MET} parameter is the available metastability settling time, or the timing slack available beyond the register's t_{CO}, for a potentially metastable signal to resolve to a known value. The t_{MET} for a synchronization chain is the sum of the output timing slacks for each register in the chain.

The overall design MTBF can be determined by the MTBF of each synchronizer chain in the design. The failure rate for a synchronizer is 1/MTBF, and the failure rate for the entire design is calculated by adding the failure rates for each synchronizer chain, as follows:

$$failure\_rate_{design} = \frac{1}{MTBF_{design}} = \sum_{i=1}^{number\ of\ chains} \frac{1}{MTBF_i}$$

Source: Intel (Altera) WP 01082

Metastability Measurements

The circuit in Figure 1 was implemented in an XC2VP4 device using 0.13 micron, 9-layer-metal technology. Two different implementations put QA, the flip-flop under test, into a CLB and into an IOB.

$$MTBF = \frac{e^{K2 * \tau}}{F1 * F2 * K1}$$

Table 1: **Virtex-II Pro Metastability Measurements and Calculations (1997 data for XC4005E added for comparison)**

Device	XC2VP4			XC2VP4			XC4005E –3
V_{CC} (V)	1.5	1.35	1.65	1.5	1.35	1.65	5.0
Flip-Flop Type	CLB	CLB	CLB	IOB	IOB	IOB	CLB
Low Frequency (MHz)	300	310	390	310	300	420	109
Half Period (ps)	1667	1613	1283	1613	1667	1190	4587
MTBF1 (ms)	60,000	20,000	60,000	30,000	30,000	30,000	1,000
High Frequency (MHz)	390	420	490	420	430	500	124.4
Half Period (ps)	1282	1190	1020	1190	1163	1000	4019
MTBF2 (ms)	1.69	1.046	5.16	0.987	1.84	6.96	0.016
Half Period Difference (ps)	385	423	262	423	504	190	568
Ln (MTBF1 / MTBF2)	10.478	9.86	9.36	10.322	9.70	8.37	11.09
K2 (per ns)	27.2	23.3	35.7	24.4	19.24	44.05	19.52
1 / K2 = tau (ps)	36.8	42.9	28.0	41.0	52.0	22.7	51.2
MTBF multiply / 100 (ps)	**15.2**	**10.3**	**35.6**	**11.5**	**6.85**	**81.8**	**7.04**

Table 1 lists the experimental results from which the exponential factor K2 was derived. The clock frequency was adjusted manually, while counting errors. Measurements were taken at room temperature, but testing at V_{CC} extremes gives an indication of performance at higher and lower temperature.

Tolerating Metastability

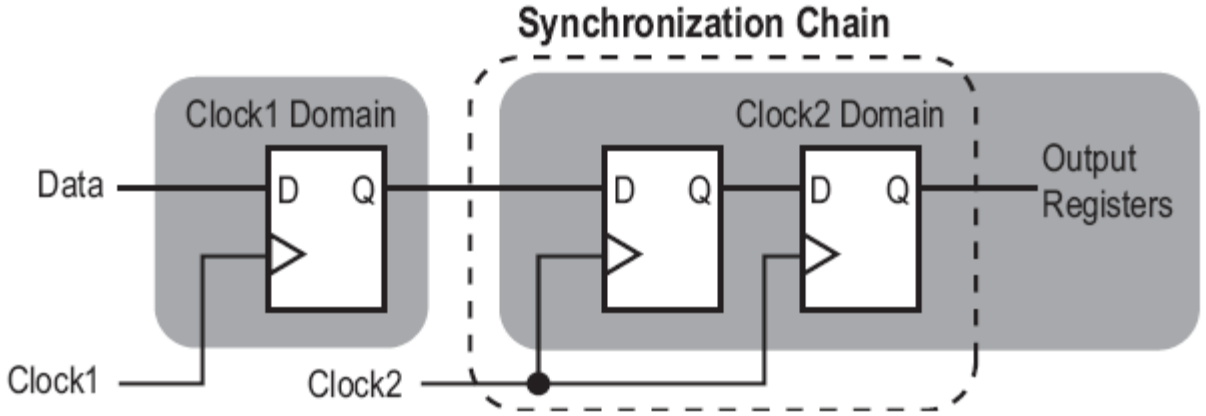
- Method 1: Make sure the clock period is long enough to allow for the resolution of quasi-stable states and for the delay of whatever logic may be in the path to the next flip-flop.
- Method 2: Add one or more successive synchronizing flip-flops to the synchronizer.
  - This approach allows for an entire clock period (except for the setup time of the second flip-flop) for metastable events in the first synchronizing flip-flop to resolve themselves.
  - Disadvantage added latency

Metastability Solutions

- Neither of these approaches can guarantee that metastability cannot pass through the synchronizer; they simply reduce the probability to practical levels.

Cascaded Registers to Tolerate Metastability (from Altera Literature)

Sample Synchronization Register Chain



- The registers in the chain are all clocked by the same or phase-related clocks
- The first register in the chain is driven from an unrelated clock domain, or asynchronously
- Each register fans out to only one register, except the last register in the chain

Source: Intel (Altera) WP 01082

Review: Principles of Synchronous Design

- Method
  - All clock inputs to flip-flops, registers, counters, etc., are driven directly from the system clock or from the clock ANDed with a control signal
- Result
  - All state changes occur immediately following the active edge of the clock signal
- Advantage
  - All switching transients, switching noise, etc., occur between the clock pulses and have no effect on system performance

Review: Asynchronous Design

- Disadvantage - More difficult
  - Problems
    - Race conditions: final state depends on the order in which variables change
    - Hazards
  - Special design techniques are needed to cope with races and hazards
- Advantages = Disadvantages of Synchronous Design
  - In high-speed synchronous design propagation delay in wiring is significant => clock signal must be carefully routed so that it reaches all devices at essentially same time
  - Inputs are not synchronous with the clock – no need for synchronizers
  - Clock cycle is determined by the worst-case delay