Imperial College London

Lecture 14

Timing Constraints & Timing Analysis

Peter Cheung
Imperial College London

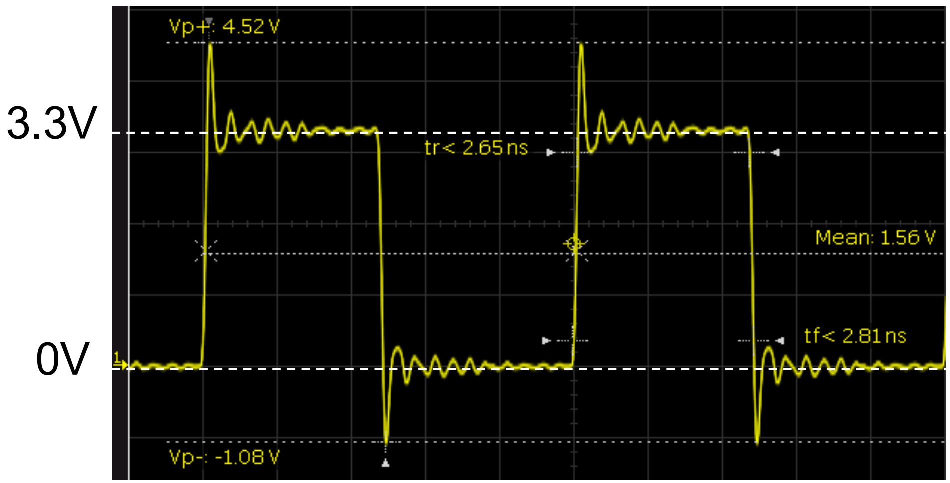
URL: www.ee.imperial.ac.uk/pcheung/teaching/E2_CAS/
E-mail: p.cheung@imperial.ac.uk

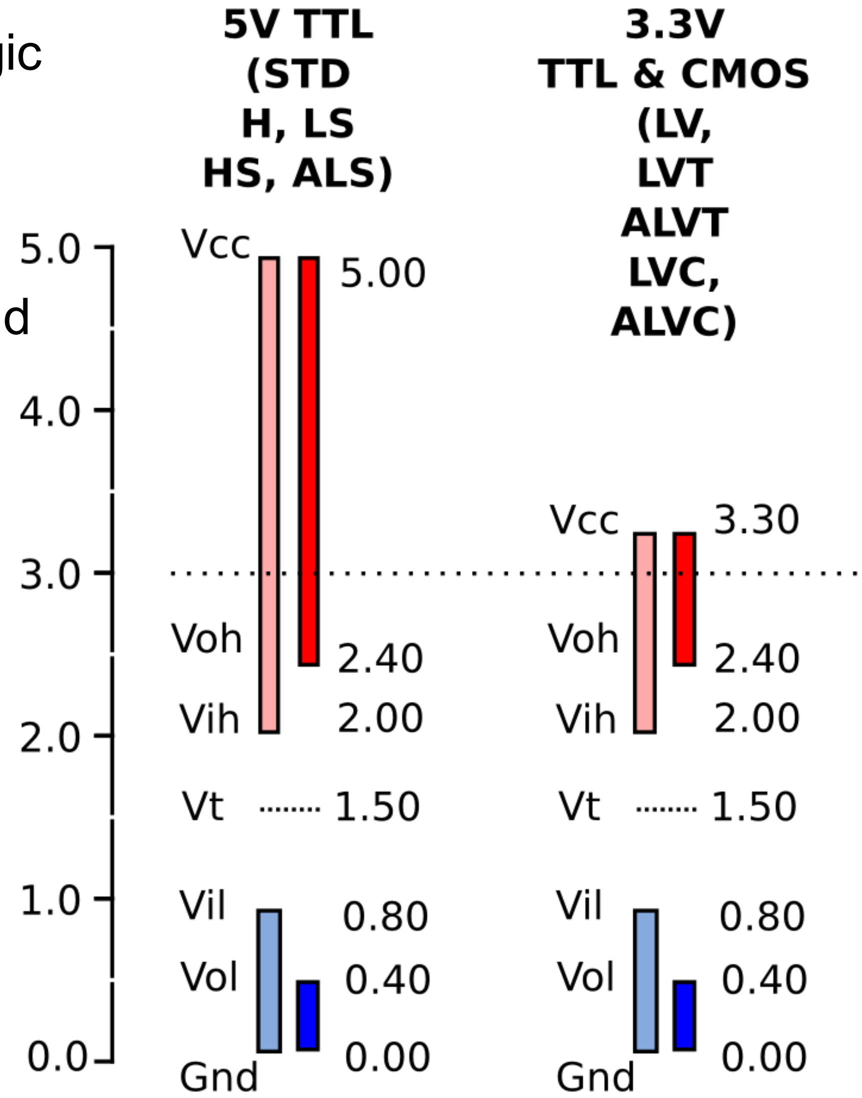
Lecture Objectives

- Appreciate the difference between theoretical and real digital signals
- Understand the low and high logic level thresholds for input and output digital signals
- Understand the meaning of noise margin and why they are needed
- Explain the meaning of setup and hold times in flipflops
- Explain how data is sent between two digital systems using a synchronous bit-serial protocol
- Investigate the timing constraints in a transmission system
- Explore the **TimeQuest** timing analyser used in the Quartus system

Typical digital signal

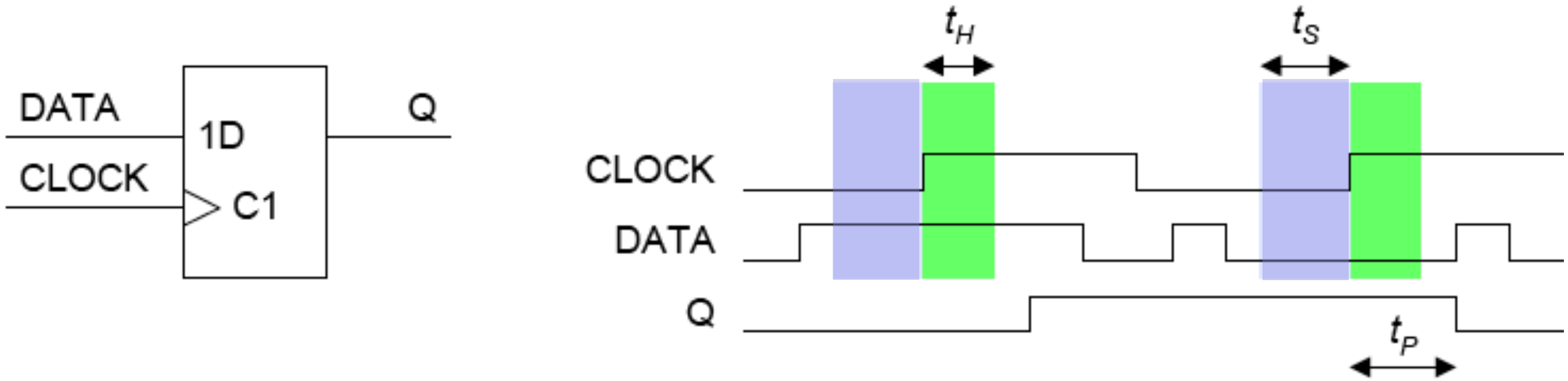
- Real digital signals are generally far from ideal.
- Shown here is a 4MHz digital signal using 3.3V logic as measured on a digital oscilloscope.
- There are overshoots and undershoots in voltage levels and finite rise and fall times.
- That's why logic circuits have well-defined threshold voltages for high and low levels as shown on the right.
- For 3.3V logic, Voh ≥ 2.4V and Vih ≥ 2V, therefore the high level margin (noise margin) is 0.4V





Setup and Hold Times

The DATA input to a flipflop or register must not change at the same time as the CLOCK.

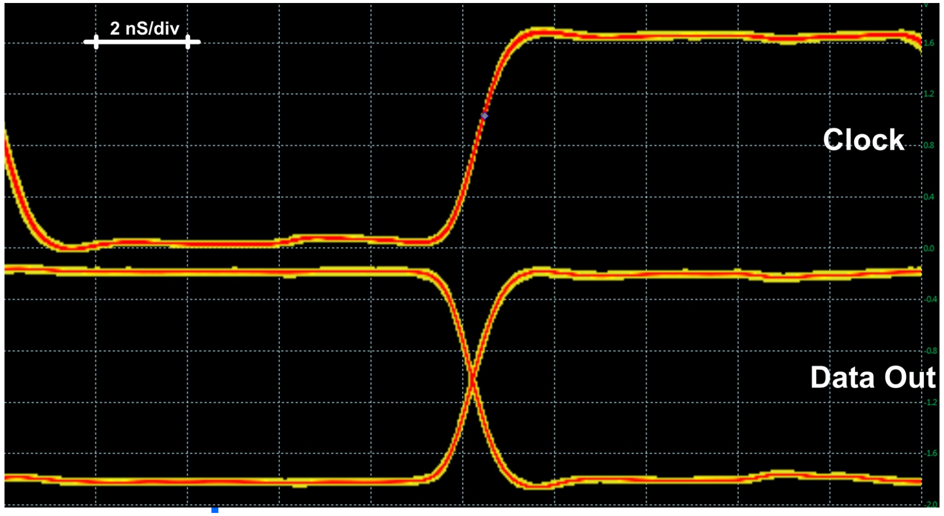


Setup Time: DATA must reach its new value at least t_S before the CLOCK↑ edge.

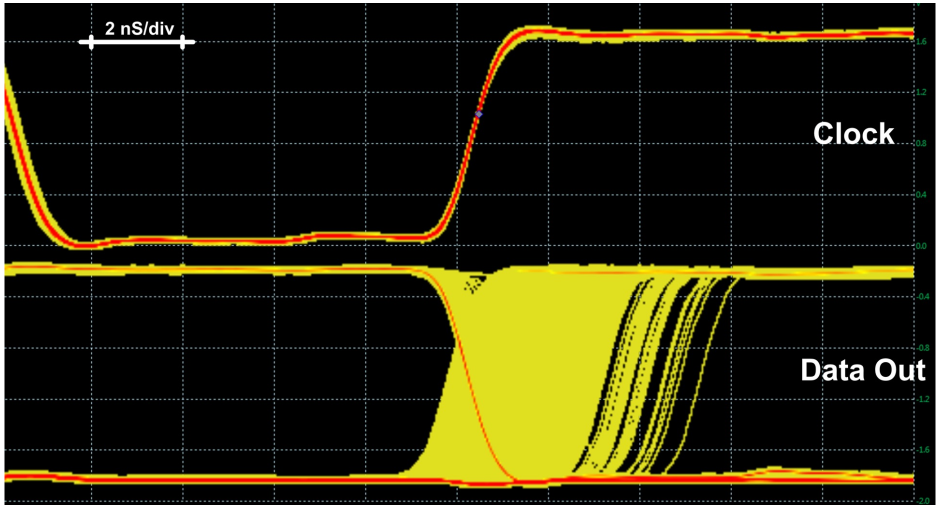
Hold Time: DATA must be held constant for at least t_H after the CLOCK↑ edge.

- Typical values for a register: t_S = 5 ns, t_H =3 ns (discrete logic/ I/O circuit)
  t_S = -50ps, t_H = 0.2 ns (internal LE)
- The setup and hold times define a window around each CLOCK ↑ edge within which the DATA **must not change**.
- If these requirements are not met, the Q output may oscillate for many nanoseconds before settling to a stable value.

Setup time violation and metastability



- No setup time violation
- Input data arrives earlier than t_S before rising edge of Clock
- Data Out changes cleanly to either 0 or 1

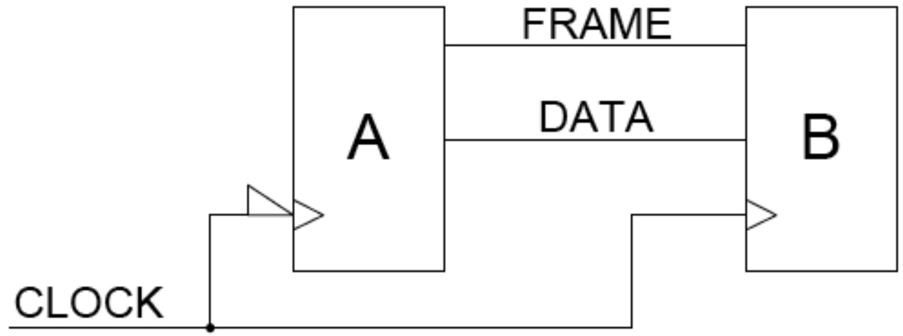


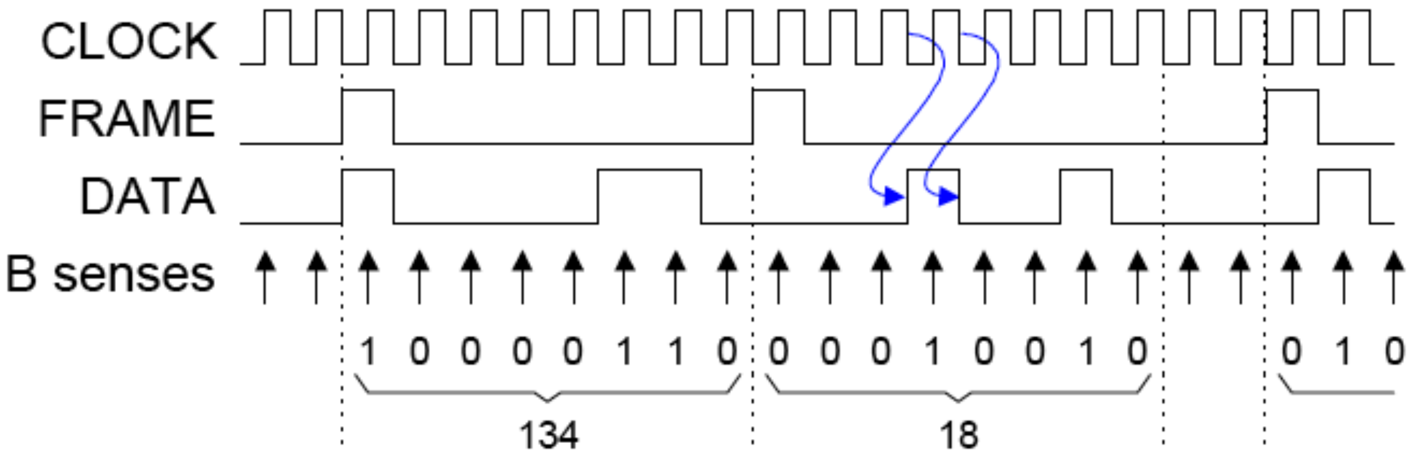
- Set up time violation
- Input data arrives within the setup time window t_S
- Data Out becomes undefined (0 or 1 or somewhere in between) for a random period time before settling down to either 0 or 1
- This can cause the digital circuit to fail

Synchronous Bit-Serial Transmission

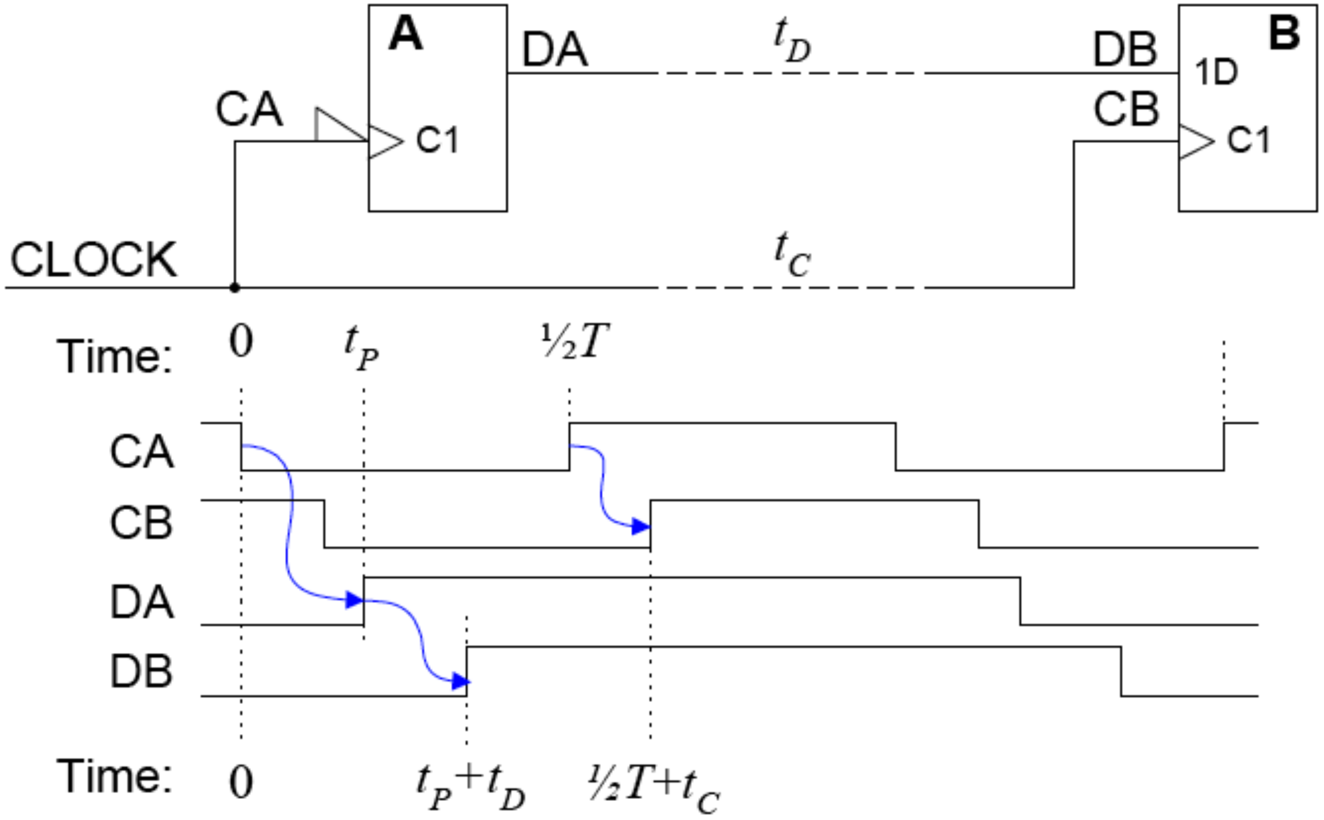
Transmitting 8 bit values from A to B:

- FRAME indicates the first bit of each value; the other 7 bits follow on consecutive clock cycles. The FRAME signal is often called **a frame sync pulse**.
- DATA changes on the *falling* CLOCK edge
- Propagation delays are often omitted from diagram.
- DATA is sensed by system B on the *rising* CLOCK edge to maximise tolerance to timing errors. We must always clock a flipflop at a time when its DATA input is not changing.





Timing Specifications

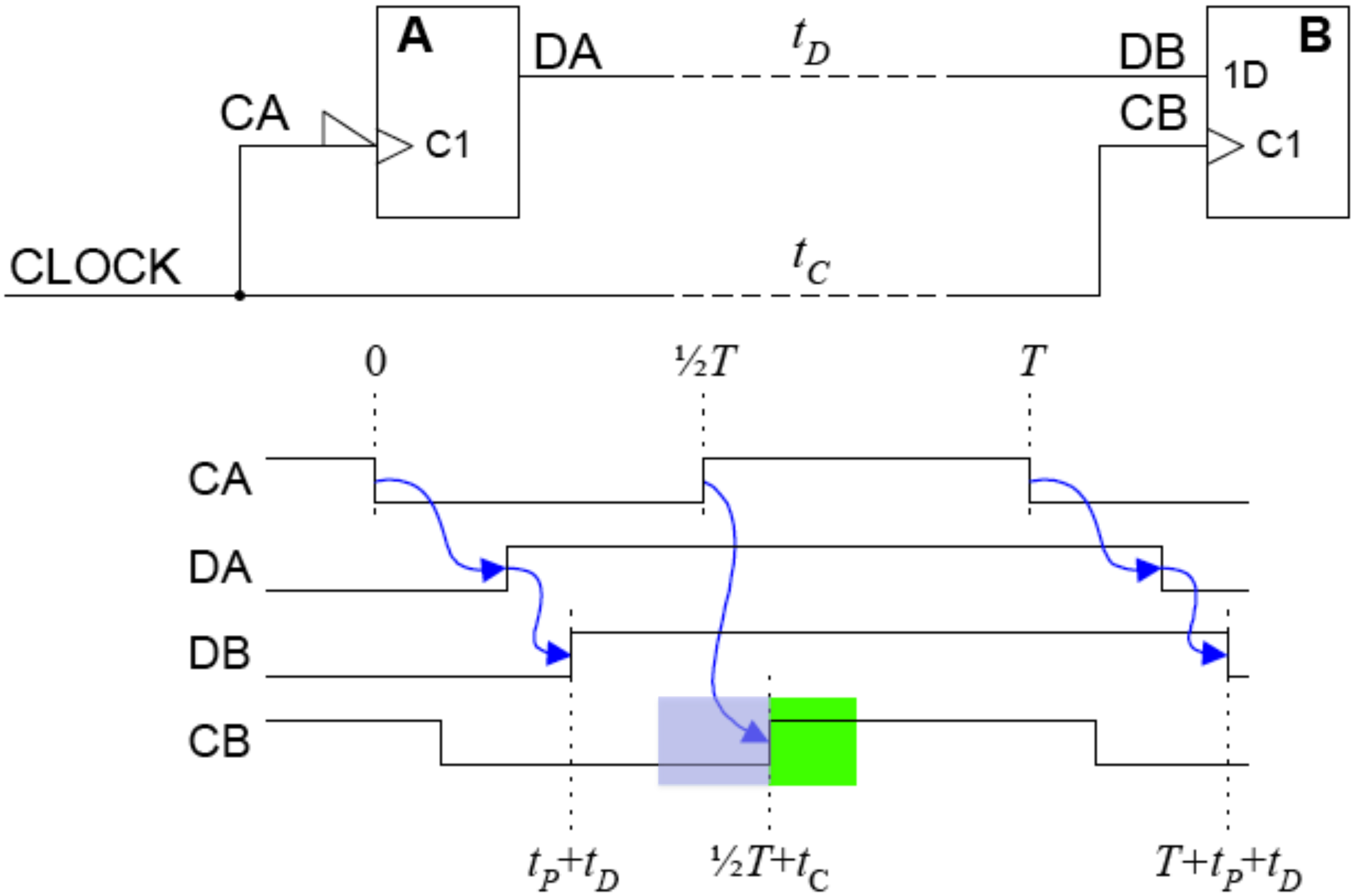


For Device B:

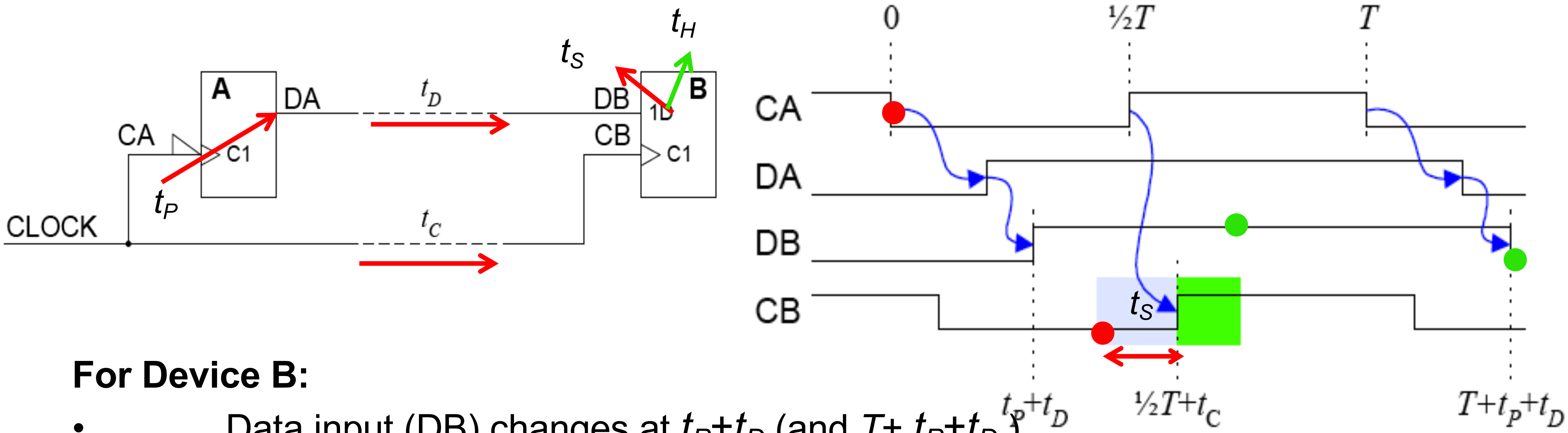
- Data input changes at time t_P+t_D
- Clock input changes ↑ at time $\frac{1}{2}T+t_C$

t_P	*Propagation* delay for device A.
T	Clock Period.
t_C, t_D	Transmission line delays for CLOCK and DATA

Timing Constraints (1)



Timing Constraints (2)



For Device B:

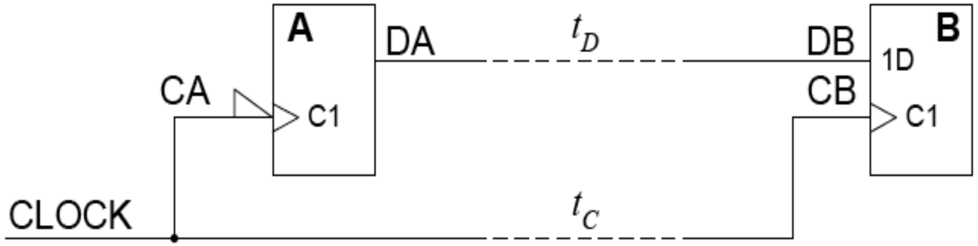
- Data input (DB) changes at t_P+t_D (and $T+ t_P+t_D$)
- Clock↑ (CB) at time $½T+t_C$

For reliable operation:

- Setup Requirement: $t_P + t_D + t_S < ½T + t_C$
- Hold Requirement: $½T+t_C + t_H < T + t_P + t_D$

Get a pair of inequalities for each flipflop/register in a circuit.
You never get both t_S and t_H in the same inequality.

Example



For a given DSP processor:

$$0 < t_P < 10 \text{ ns}, \ t_S = 5 \text{ ns}, \ t_H = 3 \text{ ns}$$

Suppose differential delay: $-10 < (t_D - t_C) < +10$

<u>Find maximum CLOCK frequency</u> (min CLOCK period):

Setup Requirement: $t_P + t_D + t_S < ½T + t_C$

- $\max(t_P + t_D) + t_S < \min(½T + t_C)$

$10 + 10 + 5 < ½T + 0$ $\quad (t_D = 10, t_C = 0)$

$½T > 25 \quad \Rightarrow \quad T > 50 \text{ ns}$

Hold Requirement: $½T + t_C + t_H < T + t_P + t_D$

- $\max(½T + t_C) + t_H < \min(T + t_P + t_D)$

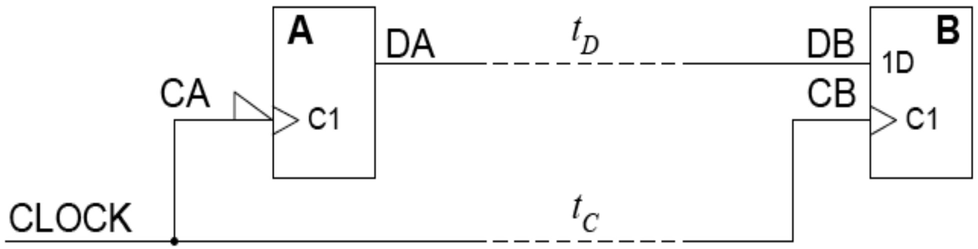
$½T + 10 + 3 < T + 0 + 0$ $\quad (t_D = 0, t_C = 10)$

$½T > 13 \quad \Rightarrow \quad T > 26 \text{ ns}$

- Hence $f_{CLOCK} < 1/50\text{ns} = 20 \text{ MHz}$

- To test for worst case: make the left side of the inequality as big as possible and the right side as small as possible.

Propagation Delay Constraint Inequalities



When do they arise?

Whenever a flipflop's clock and data input signals originate from the same ultimate source. Here CB and DB both originate from CLOCK. You normally get two inequalities for each flipflop in a circuit.

Relationship between setup and hold inequalities:

- Setup Requirement: $t_P + t_D + t_S < \frac{1}{2}T + t_C$
- Hold Requirement: $\frac{1}{2}T + t_C + t_H < t_P + t_D + T$

IMPORTANT:
These inequalities applies ONLY to this circuit.
IT IS NOT UNIVERSAL!

Are both t_S and t_H ever in the same inequality?

- No.

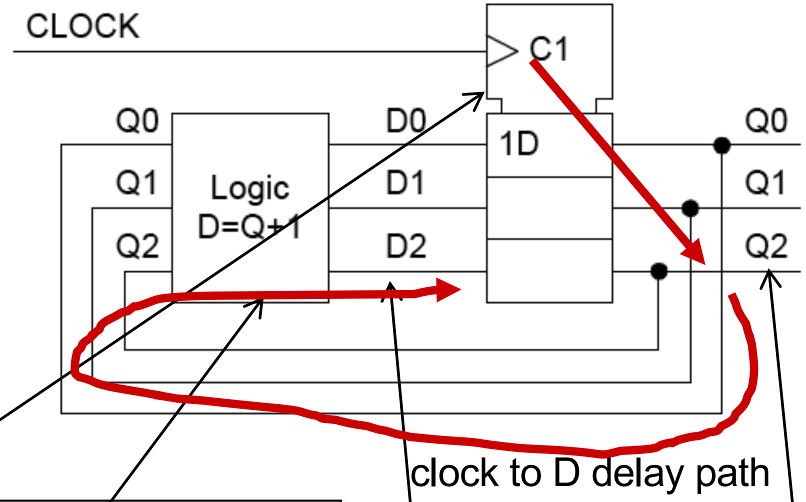
How do you decide to take the max or the min?

- For a <, take max of everything on the left and min of everything on the right.
- max = most positive: for example, max(–7,–2) = –2

The 16-bit up-counter

```
module counter (
clock,       // Clock input to the design
reset,       // active high, synchronous Reset input
enable,      // Active high enabel signal for counter
count        // 4 bit vector output of the counter
);           //    End of port list

//-------------Input Ports-----------------------------
input clock;
input reset;
input enable;

//-------------Output Ports----------------------------
output [15:0] count;

//--------
// All the
wire clock
wire reset
wire enabl

//--------
// Output
reg [15:0]
```

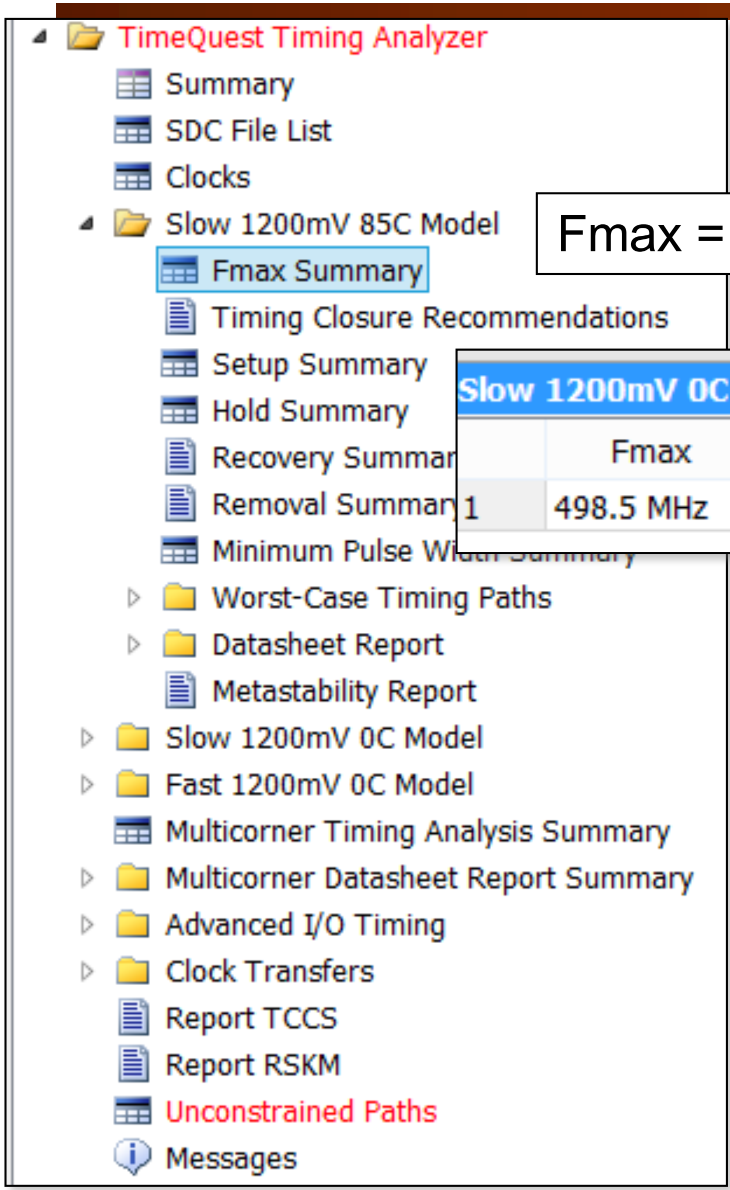
```
//------------ Main Body of the module ------------------------
  always @ (posedge clock)  begin
  // At every rising edge of clock we check if reset is active
  // If active, we load the counter output with "0000"
      if (reset == 1'b1) begin
        count <= 16'b0;
      end
  // If enable is active, then we increment the counter
      else if (enable == 1'b1) begin
        count <= count + 1'b1;
      end
    end // End of Block

endmodule // End of Module counter
```



count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock

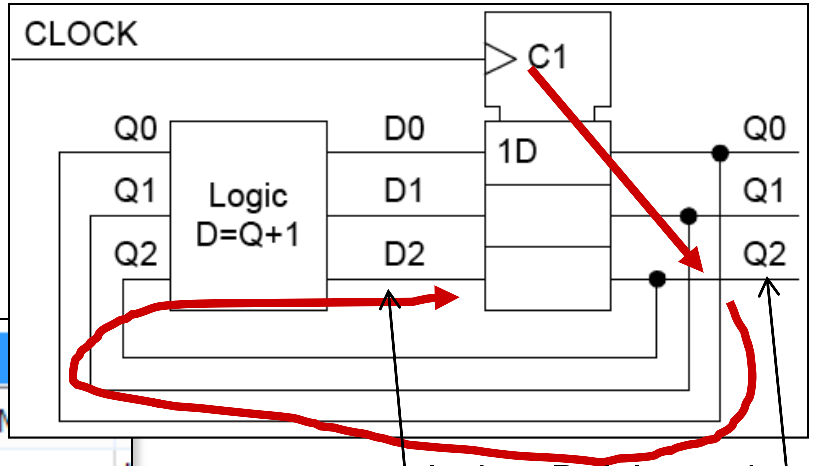
TimeQuest Report (1) - Fmax



$$F_{max} = 1/(t_{c\text{-}q} + t_p + t_{setup})$$

Slow 1200mV 0C Model Fmax Summary

	Fmax	Restricted Fmax	Clock N
1	498.5 MHz	250.0 MHz	clock



clock to D delay path

Slow 1200mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name
1	438.79 MHz	250.0 MHz	clock

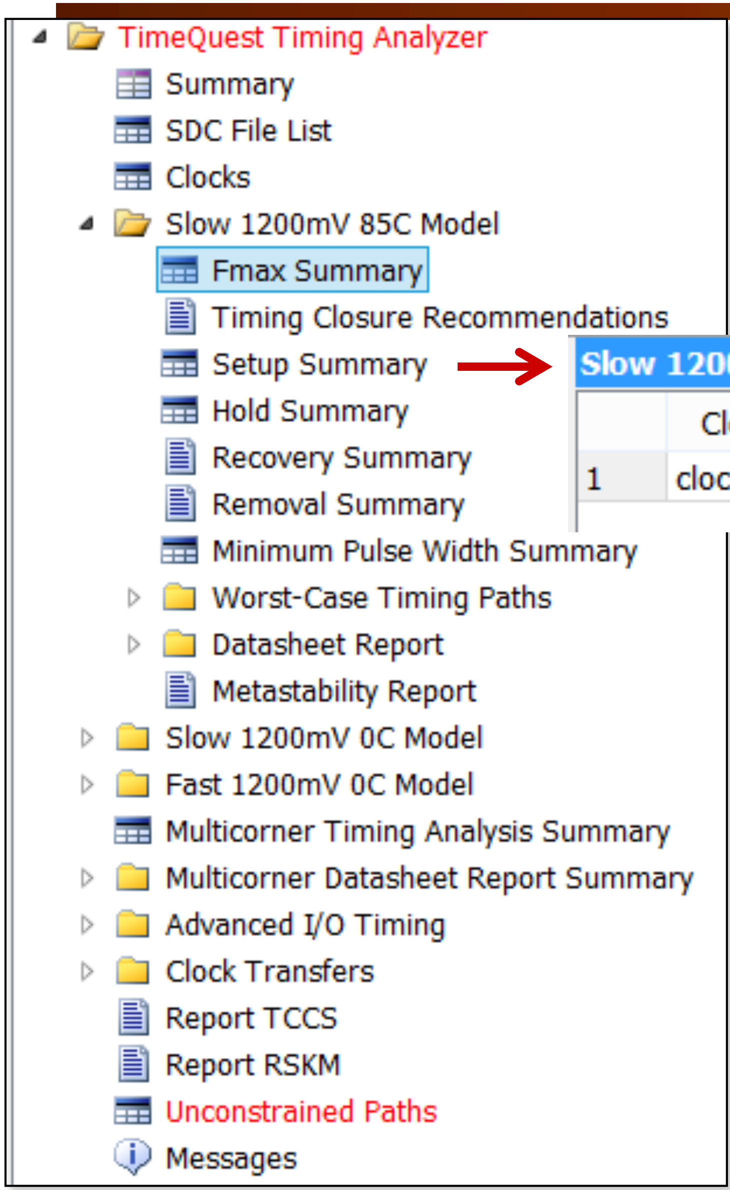
count[15:0] updated at end of **always**

Slow 1200mV 85C Model Setup Summary

	Clock	Slack	End Point TNS
1	clock	17.721	0.000

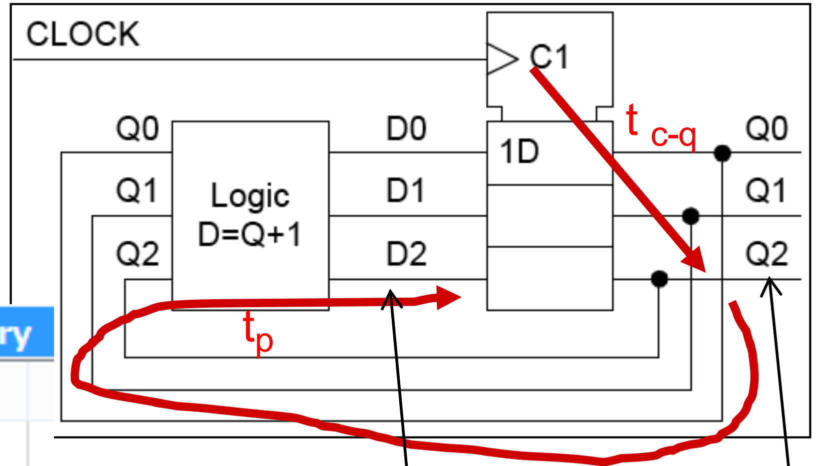
count + 1'b1 evaluated immediately after +ve edge of clock

TimeQuest Report (2) – Setup Summary



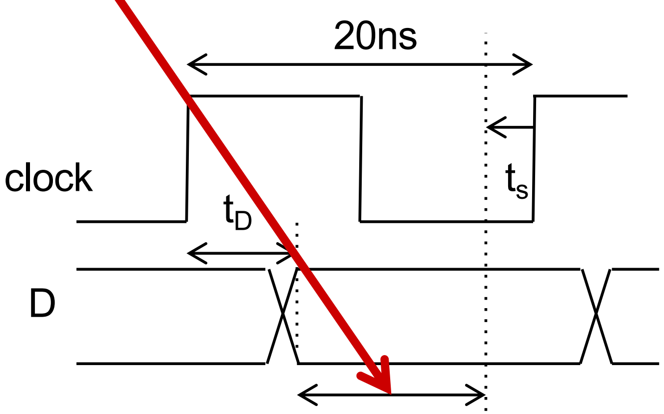
Slow 1200mV 85C Model Setup Summary

	Clock	Slack	End Point TNS
1	clock	17.721	0.000



clock to D delay path t_D

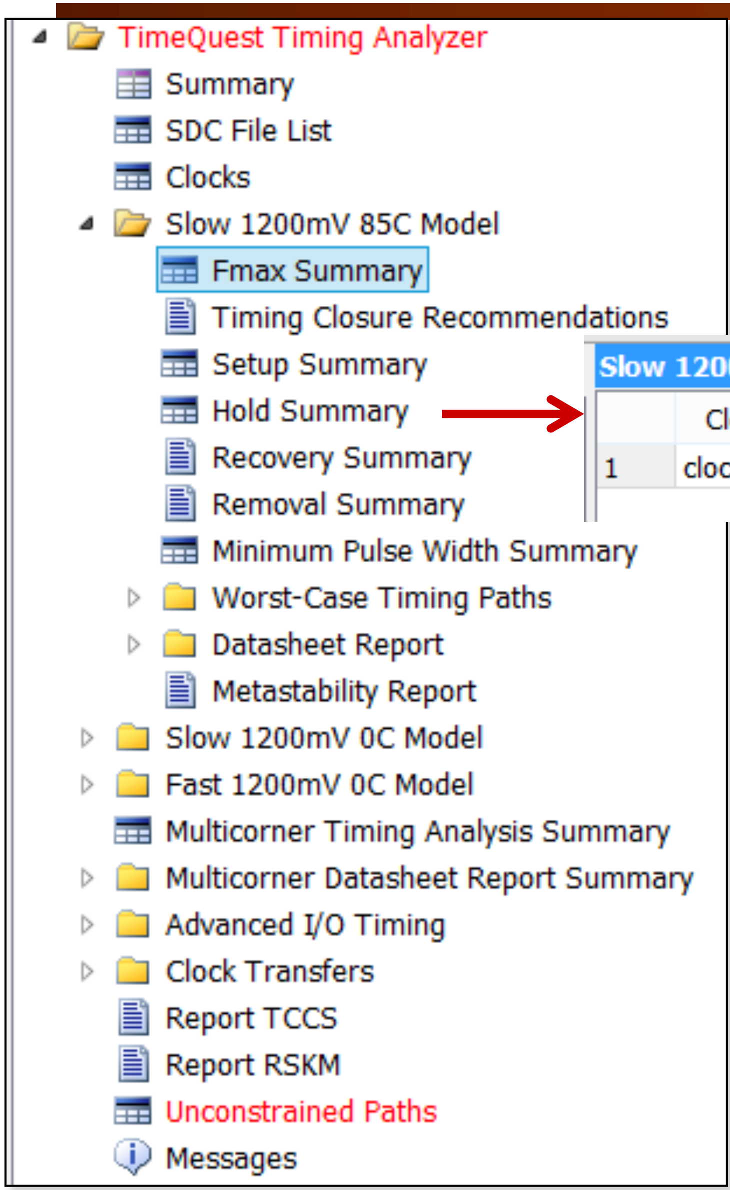
$$t_D = t_{c\text{-}q} + t_p$$



count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock

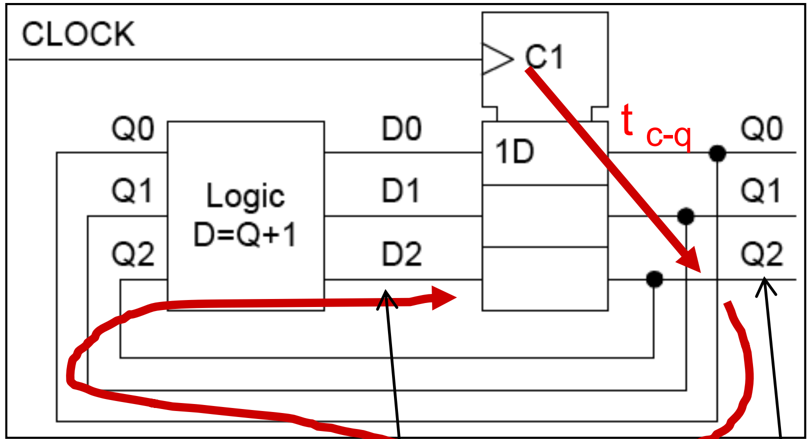
TimeQuest Report (2) – Hold Summary



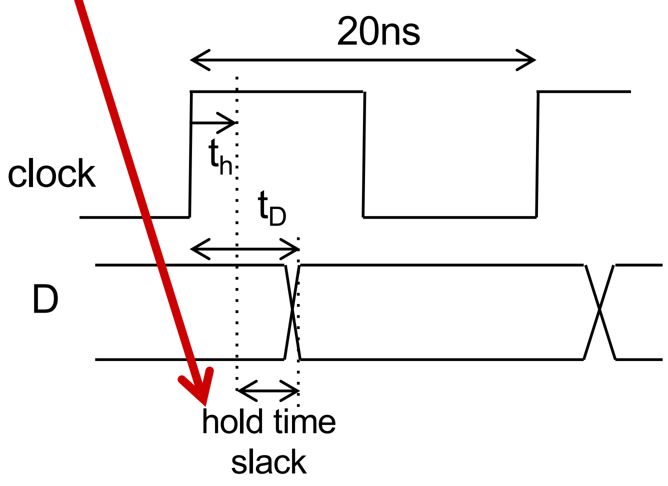
Slow 1200mV 85C Model Hold Summary

	Clock	Slack	End Point TNS
1	clock	0.570	0.000

$t_D = t_{c\text{-}q} + t_p$



clock to D delay path t_D



count[15:0] updated at end of **always**

count + 1'b1 evaluated immediately after +ve edge of clock