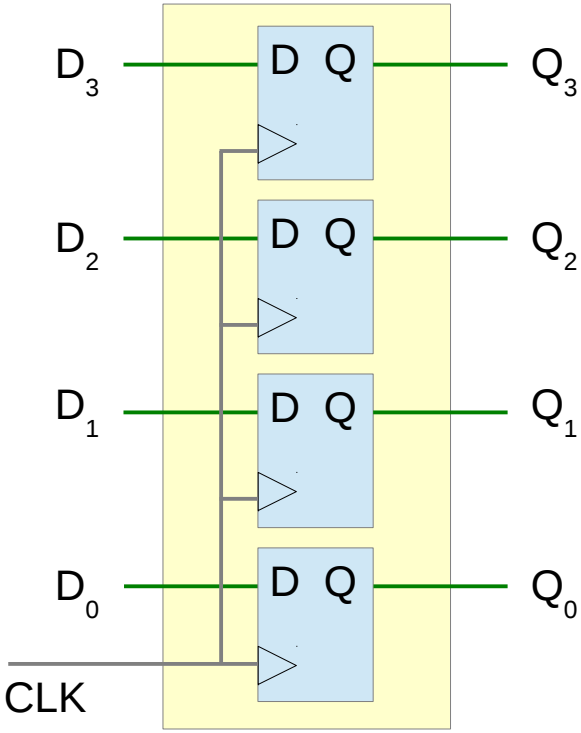
Sequential Circuit Timing

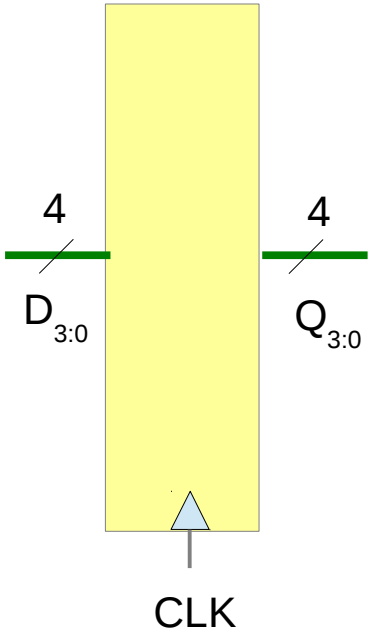
Please send corrections (or suggestions) to youngwlim@hotmail.com.

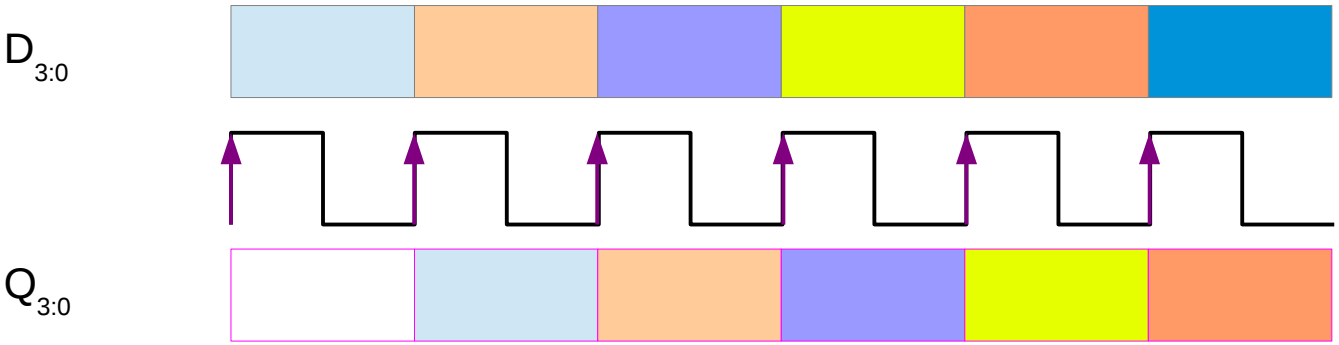
This document was produced by using OpenOffice and Octave.

Latches and FF's

Register





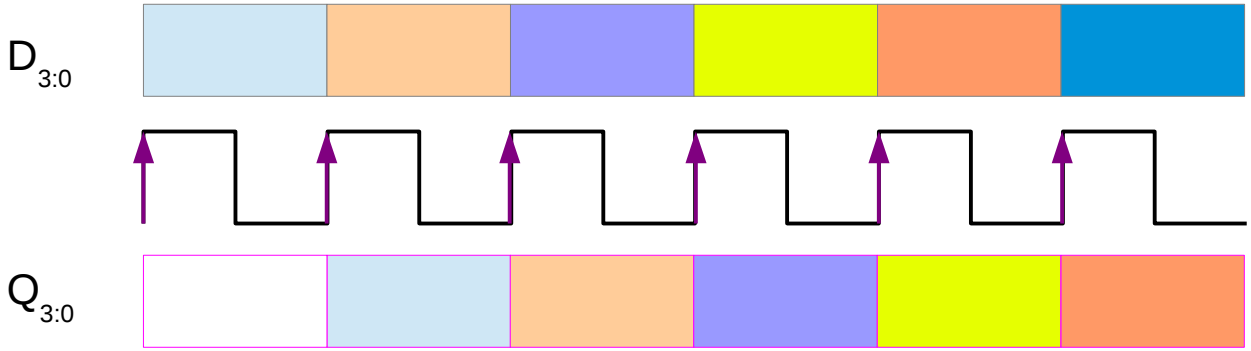


without delay

without delay

Types of Timing Diagrams

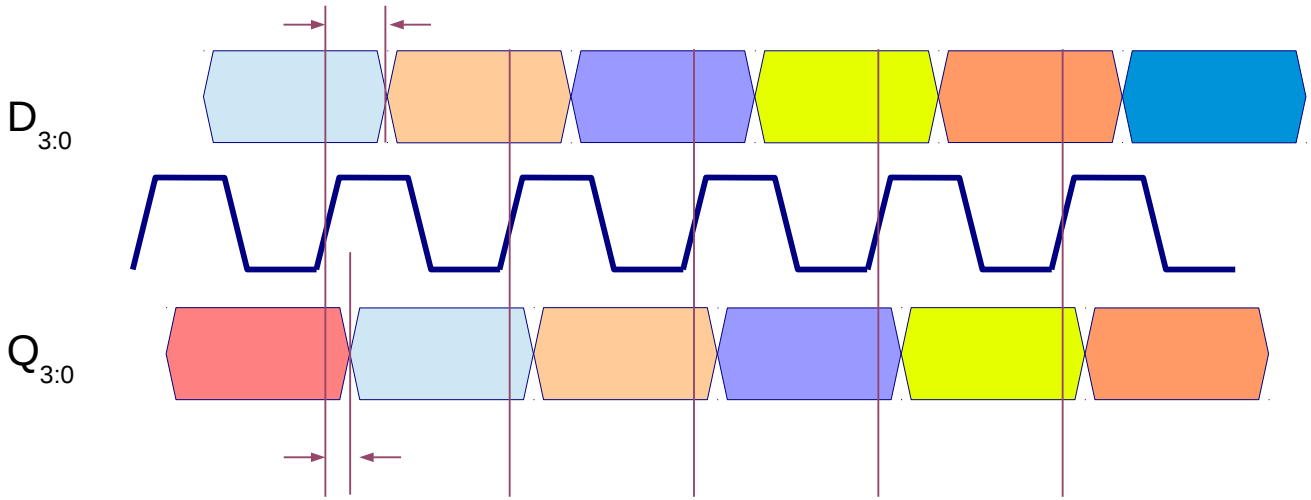
a timing diagram without delays



no delay (ideal case)

no delay (ideal case)

a timing diagram with delays



input delay

output delay Clk->Q

DFF Testbench

```
module dff(d, clk, rst, q, qb);
  input d, clk, rst;
  output q, qb;
  reg    q;

  always @(posedge clk)
  begin
    if (~rst) q = 0;
    else      q = d;
  end

  assign qb = ~q;
endmodule
```

Nonblocking Assignments <=

Blocking Assignments =

```
initial
begin
        clk =0;
        d   =0;
        rst =1;
        rst =0;
  #20   rst = 1;

  #10  d <=  1;
  #10  d <=  0;
  #10  d <=  1;
  #10  d <=  0;
  #10  d <=  1;
  #10  d <=  1;

  $finish;
end
```

```
#10  d =  1;
#10  d =  0;
#10  d =  1;
#10  d =  0;
#10  d =  1;
#10  d =  1;
```

```
`timescale 1ns/100ps

module dff_tb;

reg d, clk, rst;

dff U1 (d, clk, rst, q, qb);

always #10 clk = ~clk;

initial
begin
   $dumpfile("test.vcd");
   $dumpvars(0, dff_tb);
end

endmodule
```

Testbench with Nonblocking Assignments

```
module dff(d, clk, rst, q, qb);
  input d, clk, rst;
  output q, qb;
  reg    q;

  always @(posedge clk)
  begin
    if (~rst) q = 0;
    else      q = d;
  end

  assign qb = ~q;
endmodule
```

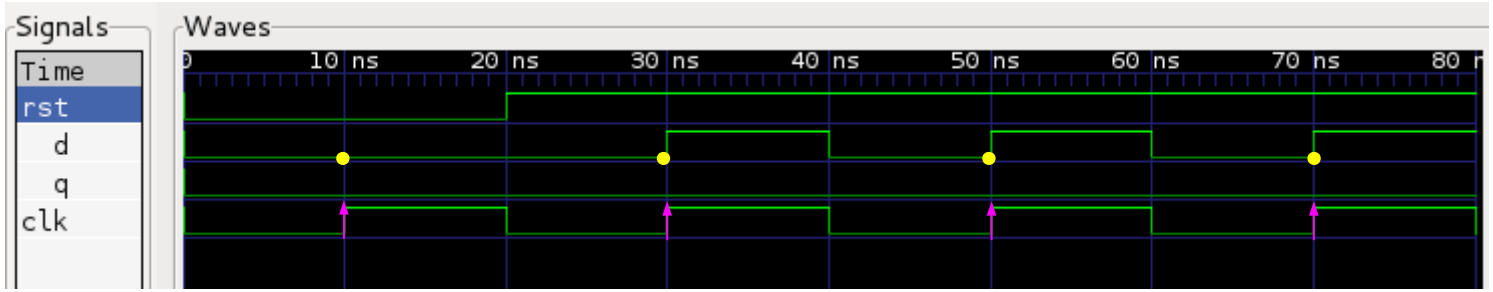
```
initial
begin
        clk =0;
        d   =0;

        rst =1;
        rst =0;
   #20  rst = 1;

   #10  d <=  1;
   #10  d <=  0;
   #10  d <=  1;
   #10  d <=  0;
   #10  d <=  1;
   #10  d <=  1;
  $finish;

end
```

```
`timescale 1ns/100ps

module dff_tb;

reg d, clk, rst;

dff U1 (d, clk, rst, q, qb);

always #10 clk = ~clk;


initial
begin
   $dumpfile("test.vcd");
   $dumpvars(0, dff_tb);
end

endmodule
```

DFF Testbench Waveforms

Nonblocking Assignments

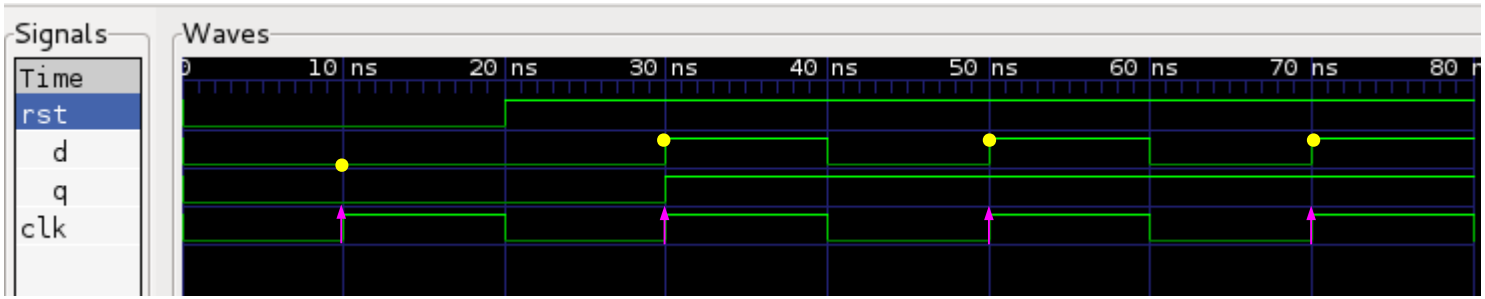
```
#10 d <= 1;
#10 d <= 0;
#10 d <= 1;
#10 d <= 0;
#10 d <= 1;
#10 d <= 1;
```



samples the unchanged d input values at the posedge of clk

Blocking Assignments

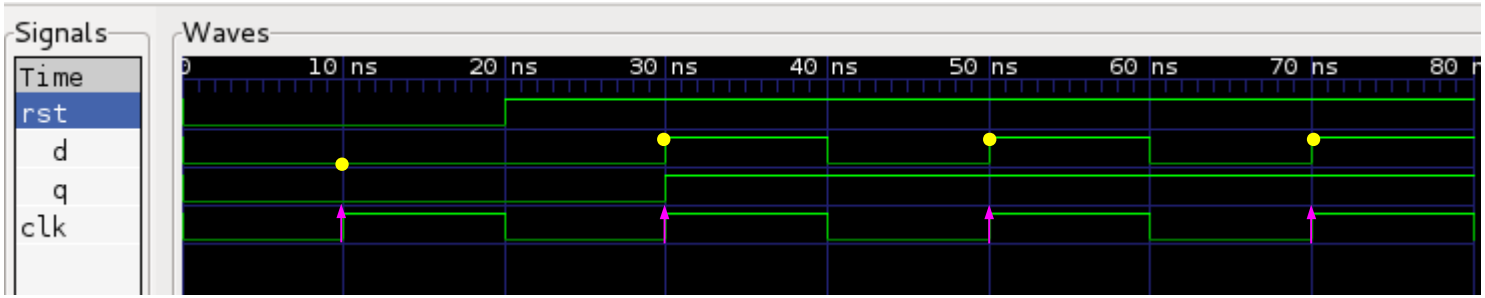
```
#10 d = 1;
#10 d = 0;
#10 d = 1;
#10 d = 0;
#10 d = 1;
#10 d = 1;
```



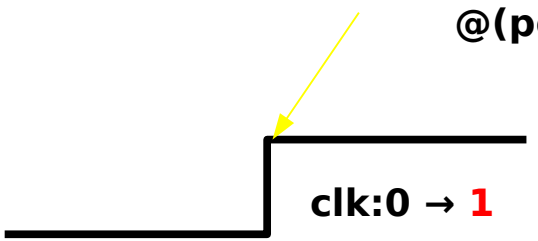
samples the changed d input values at the posedge of clk

Blocking Assignments

```
#10  d = 1;
#10  d = 0;
#10  d = 1;
#10  d = 0;
#10  d = 1;
#10  d = 1;
```




```
always @(posedge clk)
begin
  if (~rst) q = 0;
  else      q = d;
end
```

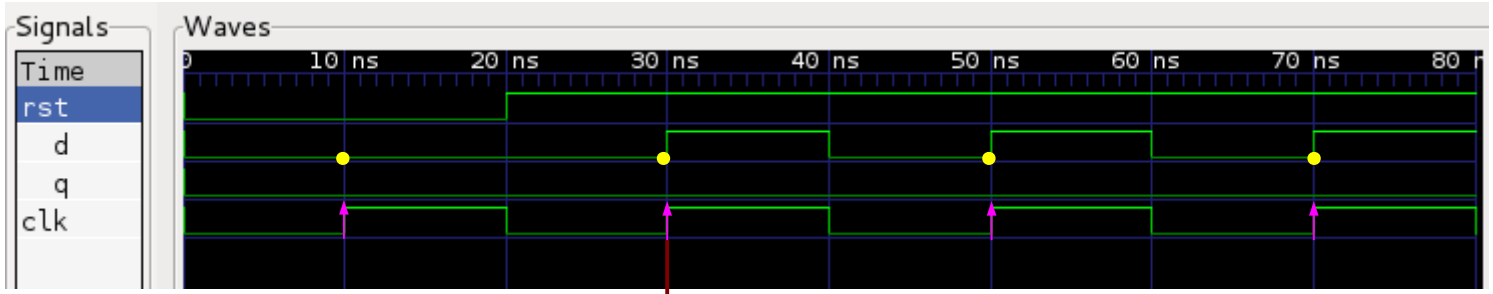


At the posedge of clk
clk has changed to 1
Also, d has changed to
its new value

Nonblocking Assignments

Nonblocking Assignments

```
#10 d <= 1;
#10 d <= 0;
#10 d <= 1;
#10 d <= 0;
#10 d <= 1;
#10 d <= 1;
```



In each of time slot
which is delayed by 10 time units,
there is only one assignment
that can be scheduled simultaneously,

samples the unchanged d input values
at the posedge of clk

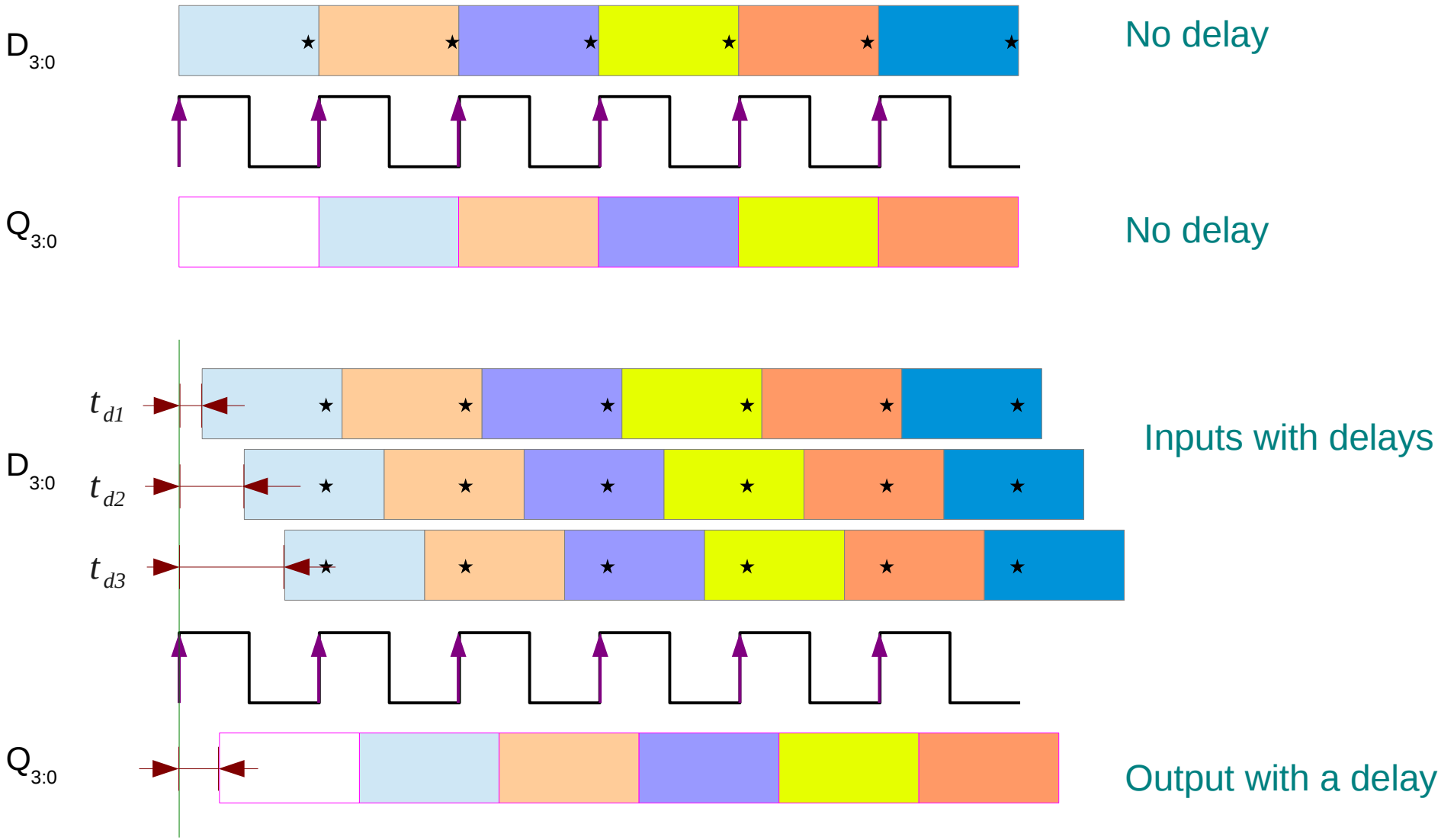
(1) Allowing scheduling of assignments without waiting for its completion
(2) Executed last in the time step in which it is scheduled (after all blocking assignments' execution)

Regular Events	Nonblocking Update Events	Monitor Events

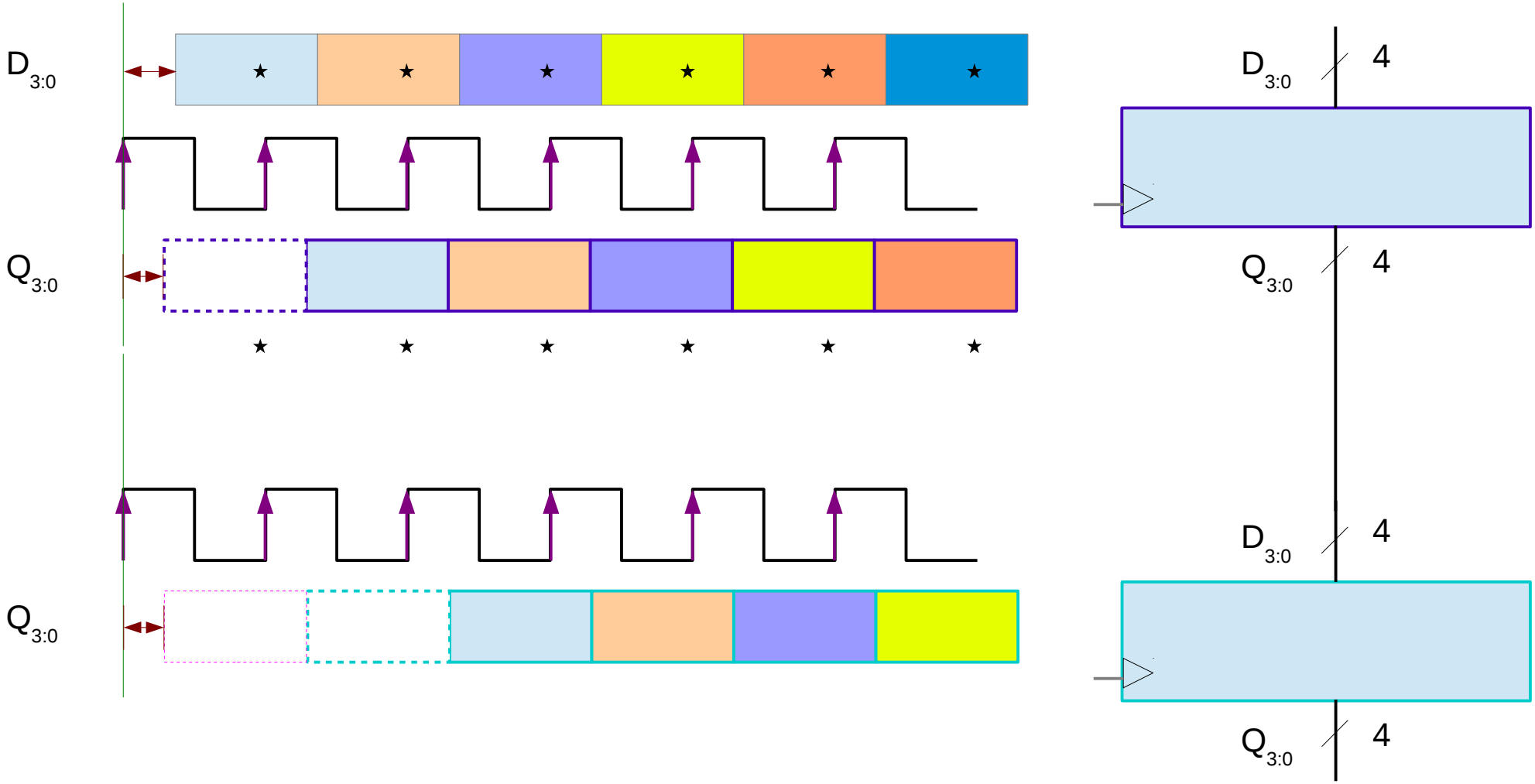
- Evaluate
- Update

So, the input d update is done after the posedge clk

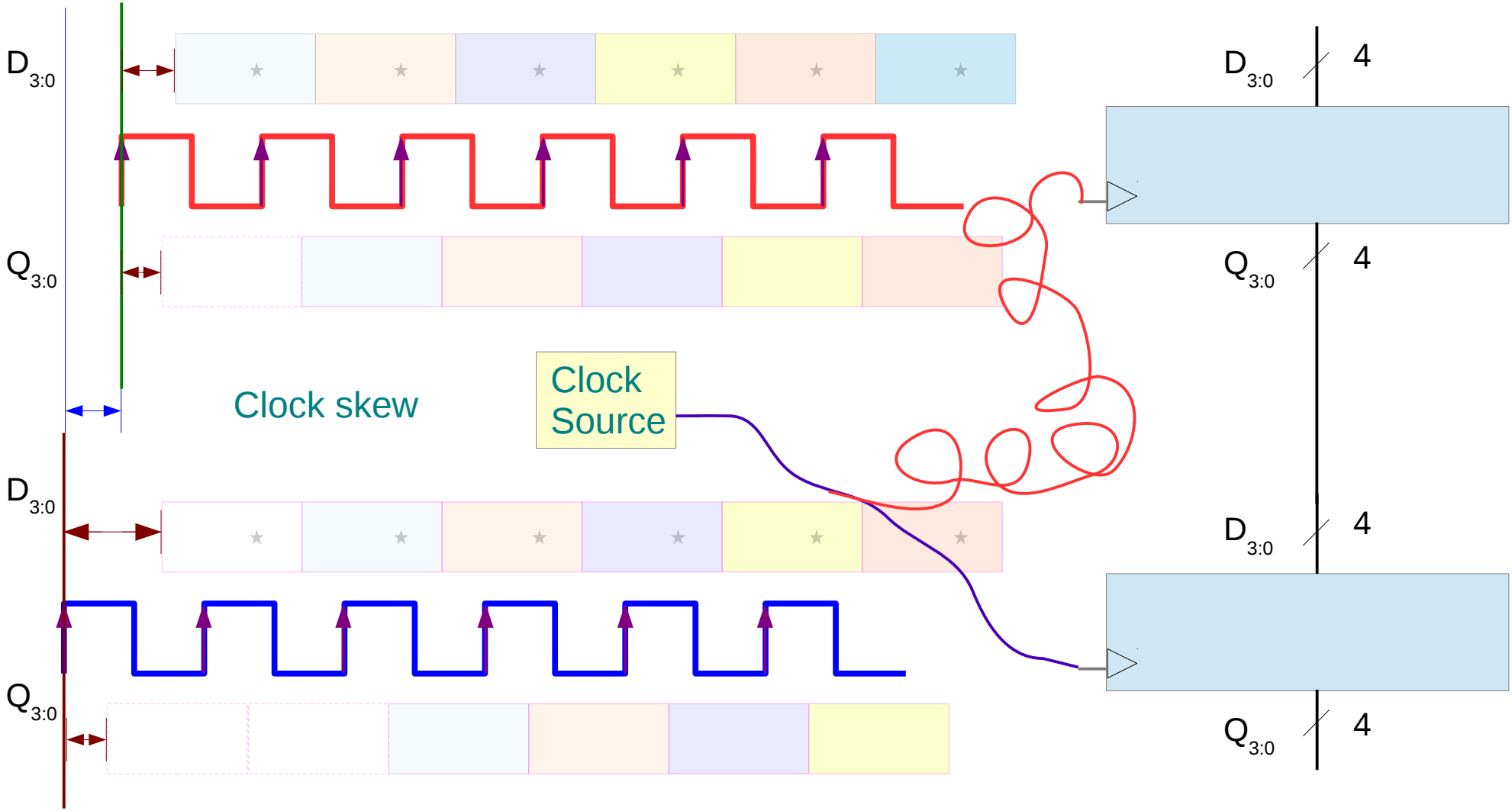
FF Timing – Input and Output Delays



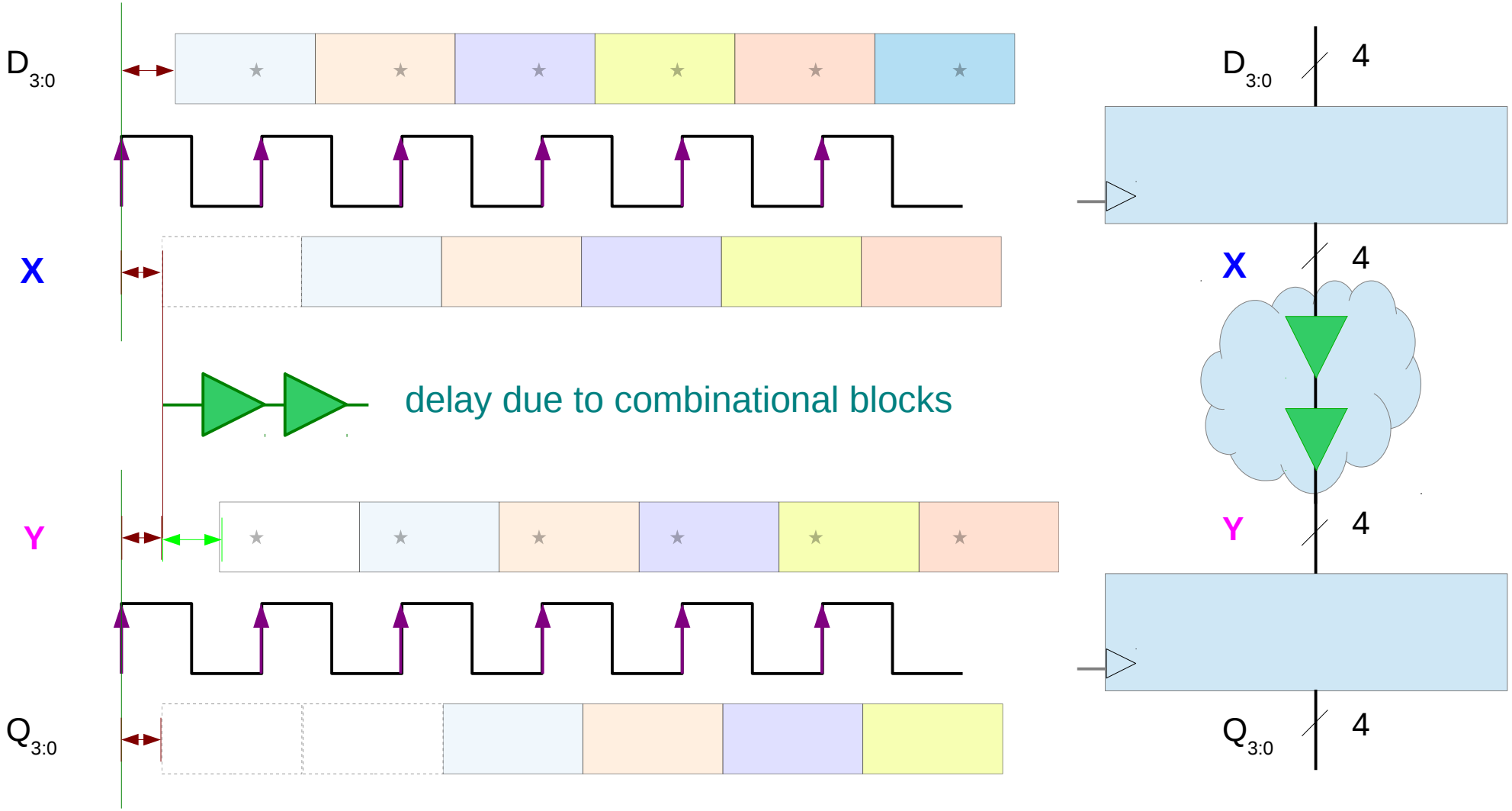
Reg to Reg Timing



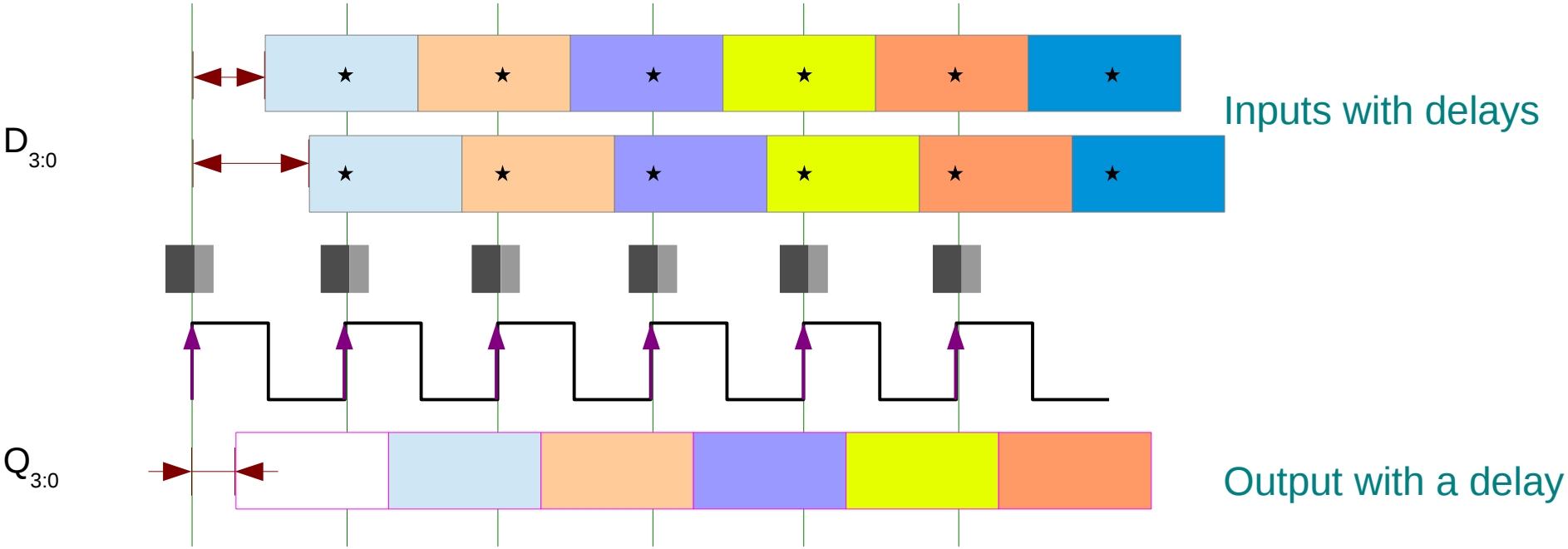
Clock Skew

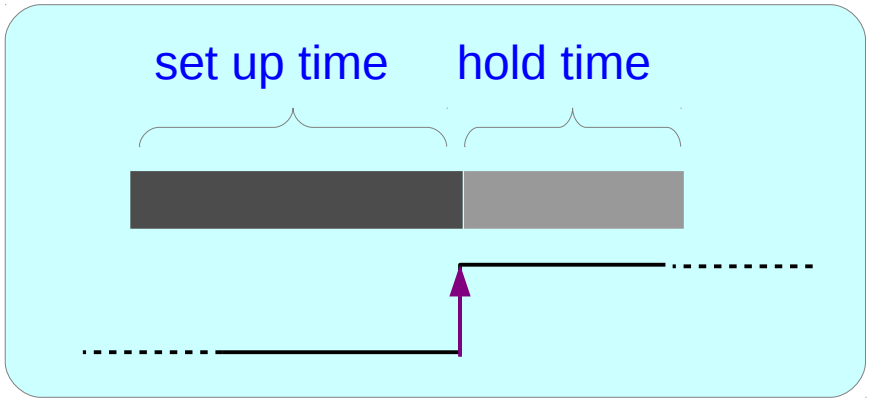


Path Delay

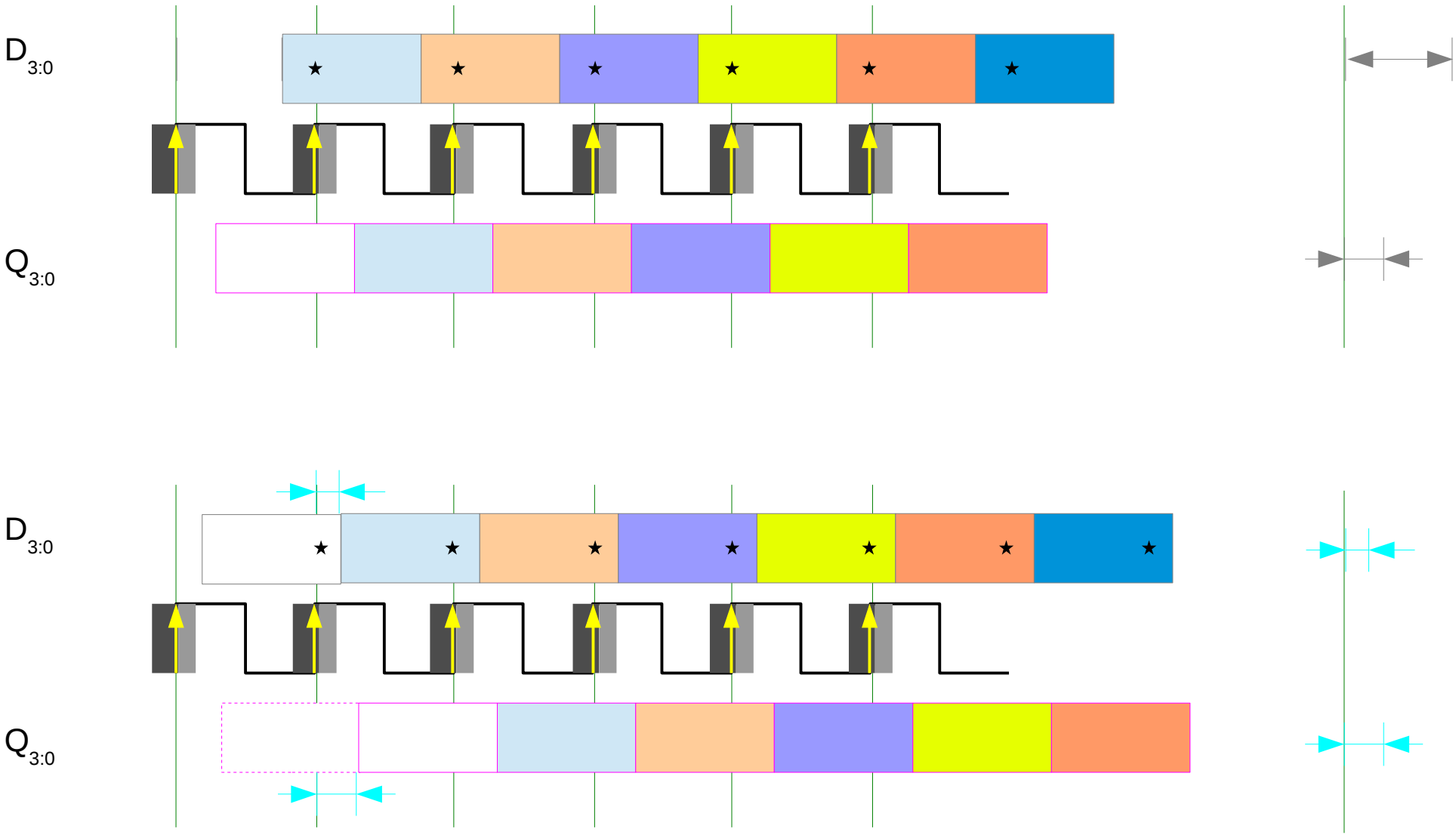


Setup & Hold Time (1)

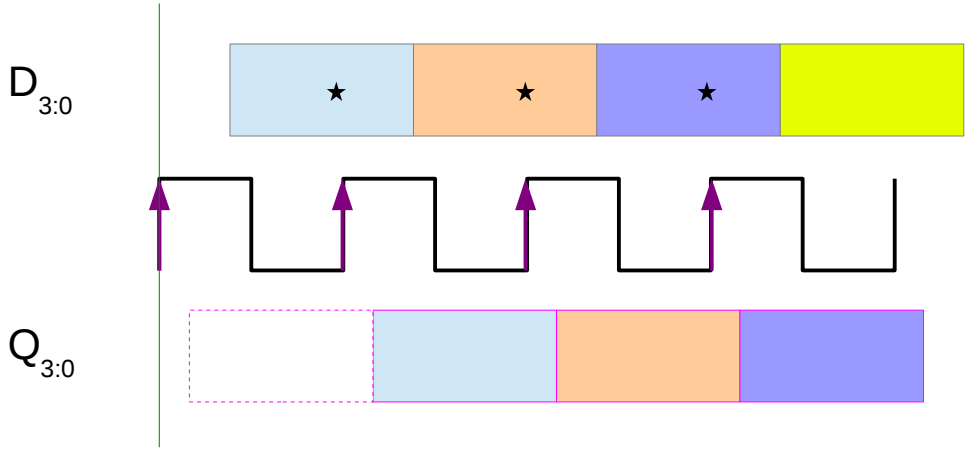


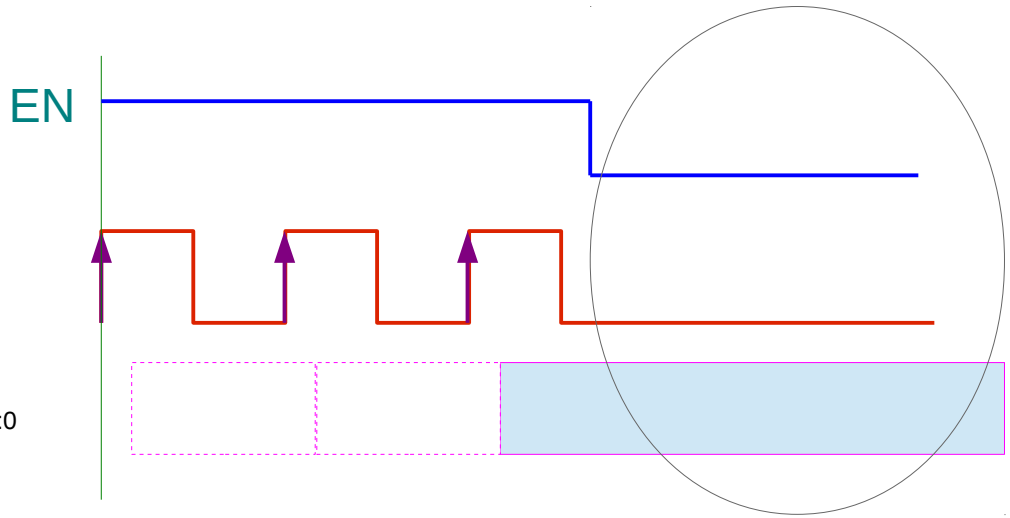


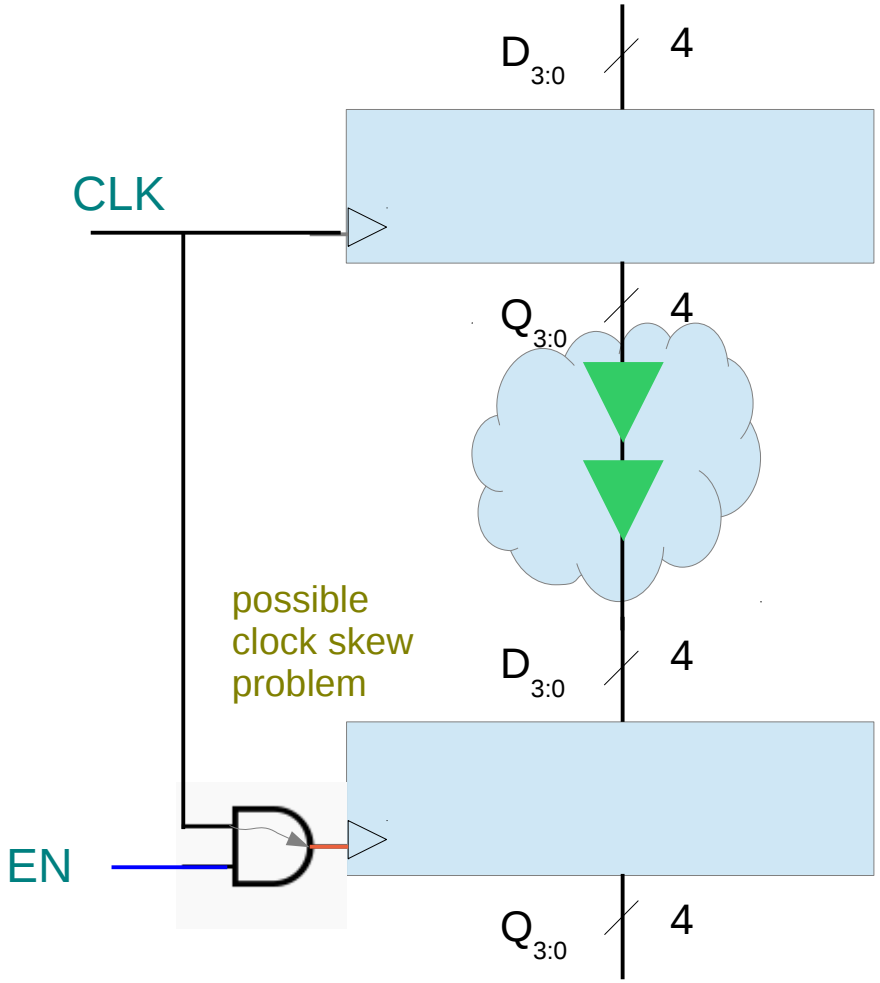
Setup & Hold Time (2)



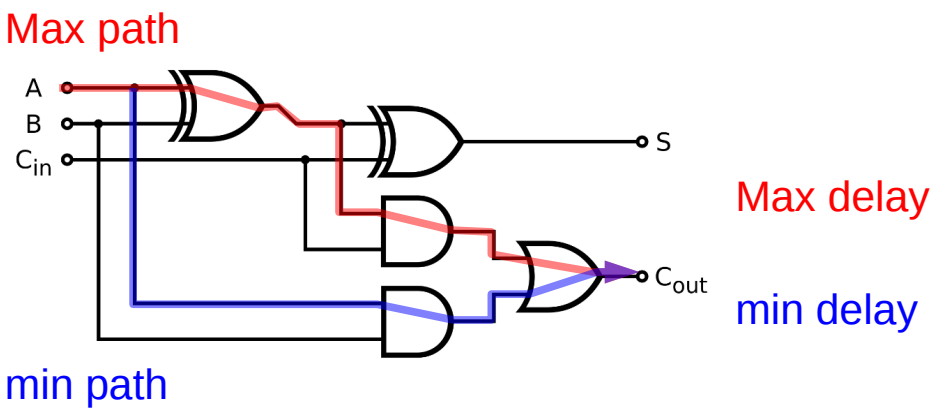
Clock Gating

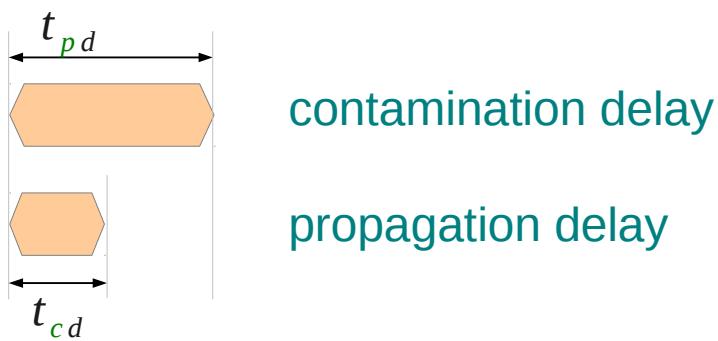


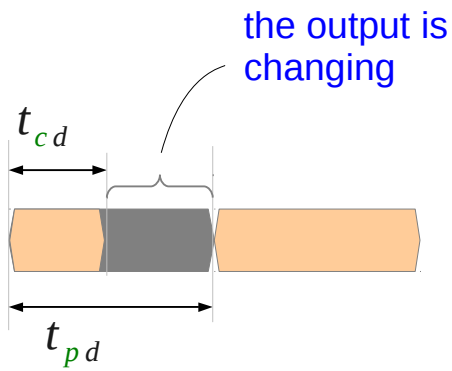




Max Path / Min Path



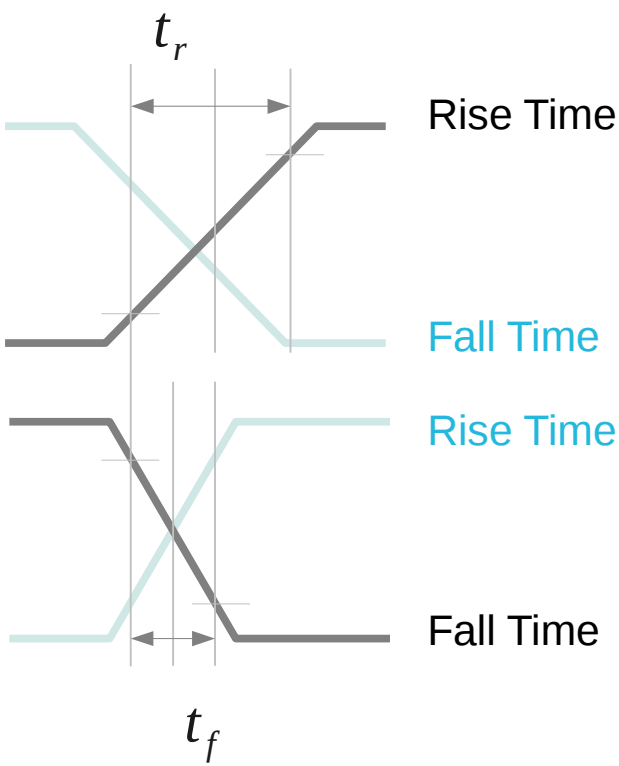




$$t_{cd} \leq t_{delay} \leq t_{pd}$$

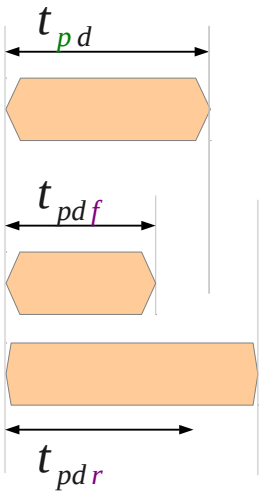
min delay Max delay

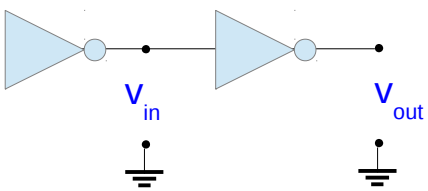
Rise / Fall Times



Max delay

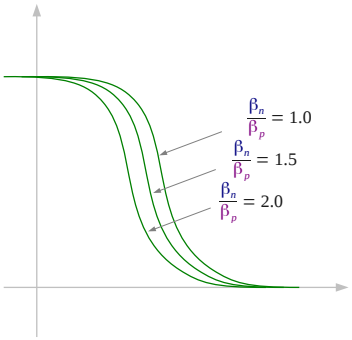
min delay

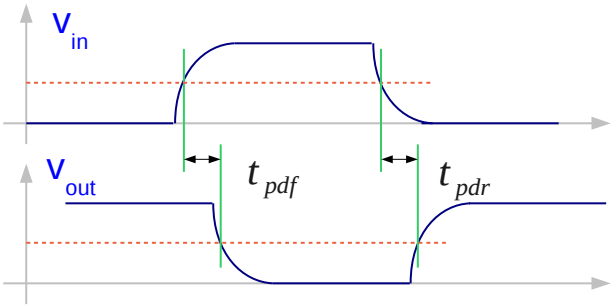




$$\frac{\beta_n}{\beta_p} > 1 \qquad \frac{R_n}{R_p} < 1$$

$$\frac{t_f}{t_r} = \frac{2.2\,\tau_n}{2.2\,\tau_p} \qquad \frac{\tau_n}{\tau_p} = \frac{R_n C_{out}}{R_p C_{out}} = \frac{R_n}{R_p} < 1$$



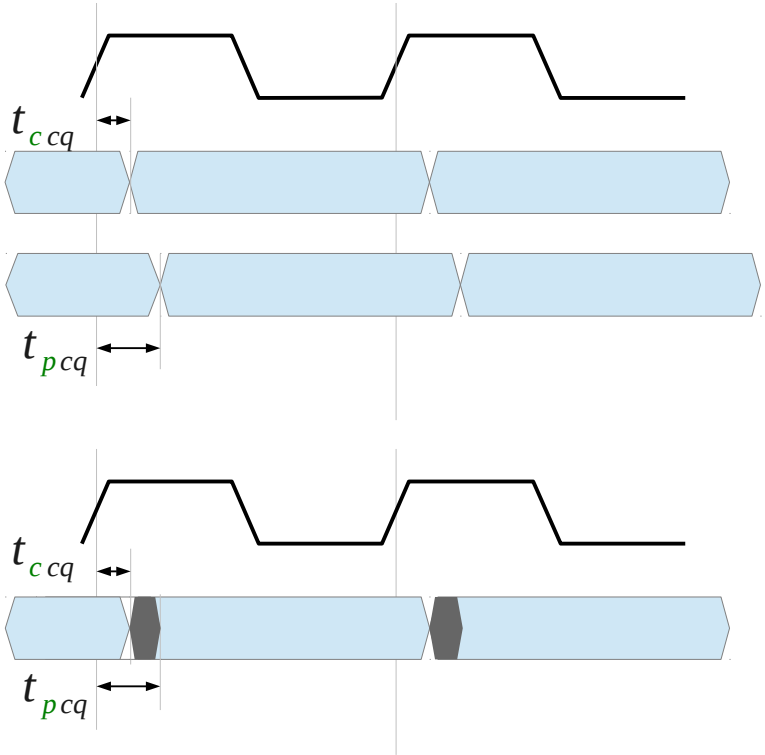


PVT Variation

- Process
- Voltage
- Temperature

High temperature	Max delay
Low temperature	min delay

FF Output Delay



contamination delay

propagation delay

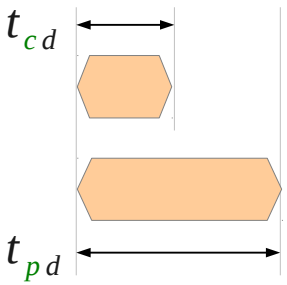
flipflop clock-to-q

$$t_{ccq} \leq t_{delay} \leq t_{pcq}$$

min delay

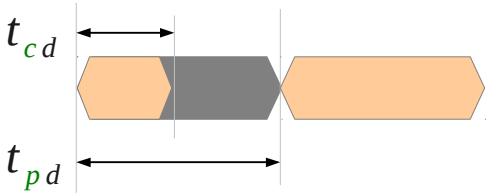
Max delay

Path Delay



contamination delay

propagation delay

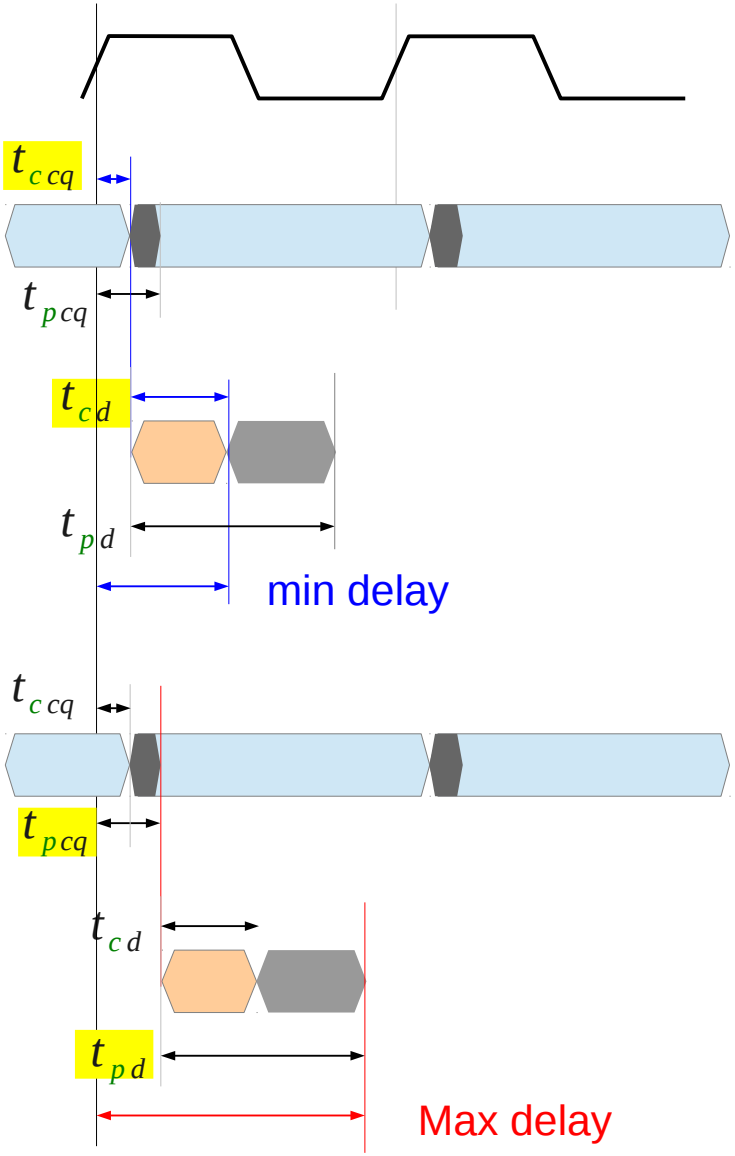


combinational logic delay

$$t_{cd} \leq t_{delay} \leq t_{pd}$$

min delay Max delay

Reg-to-Reg Delay (1)



$$t_{ccq} \le t_{FF} \le t_{pcq}$$

min delay Max delay

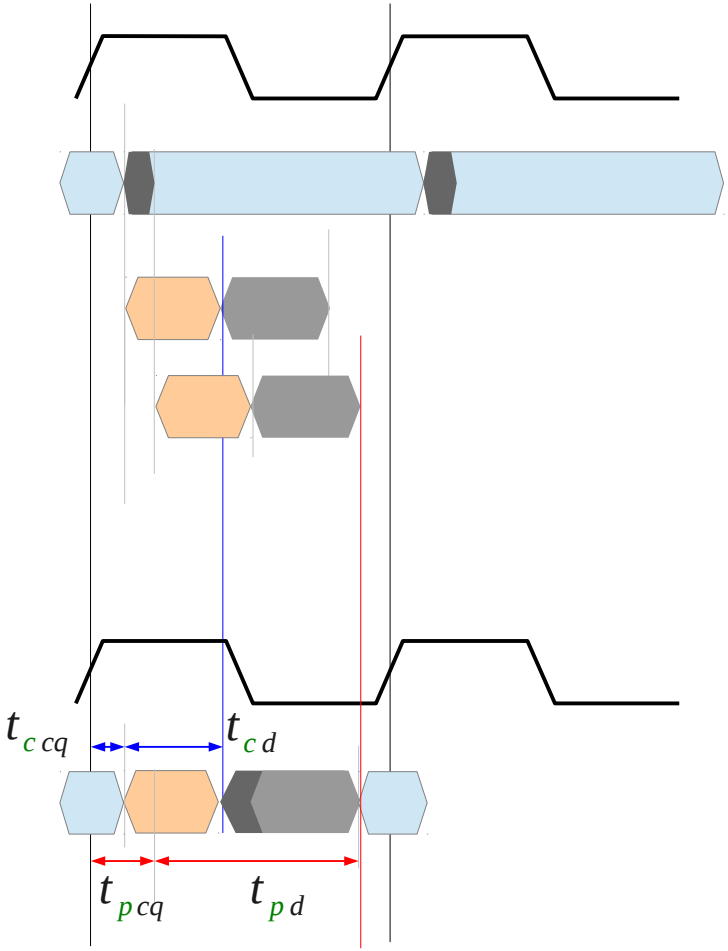
$$t_{cd} \le t_{comb} \le t_{pd}$$

min delay Max delay

$$t_{ccq} + t_{cd} \le t_{delay} \le t_{pcq} + t_{pd}$$

min delay Max delay

Reg-to-Reg Delay (2)



$$t_{ccq} \le t_{FF} \le t_{pcq}$$

min delay Max delay

$$t_{cd} \le t_{comb} \le t_{pd}$$

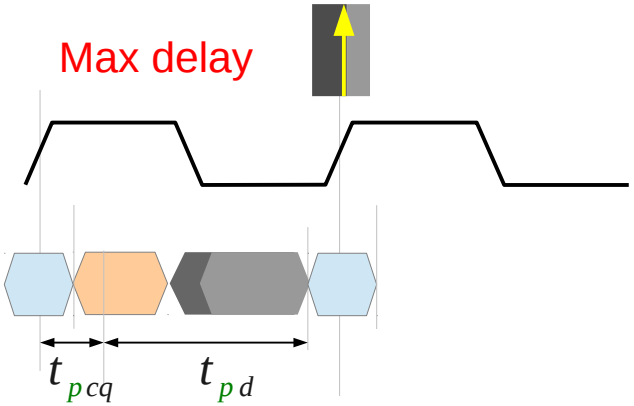
min delay Max delay

$$t_{ccq} + t_{cd} \le t_{delay} \le t_{pcq} + t_{pd}$$

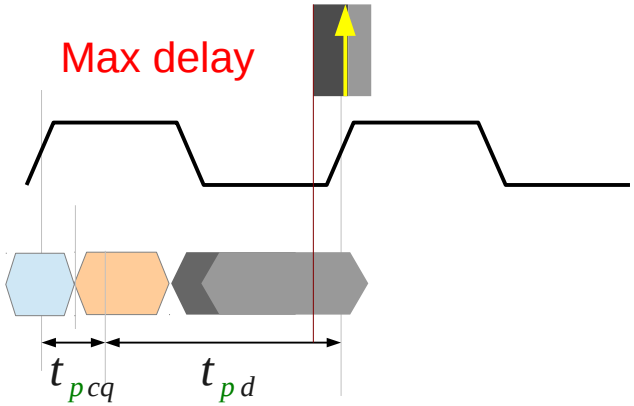
min delay Max delay

Setup Time / Hold Time

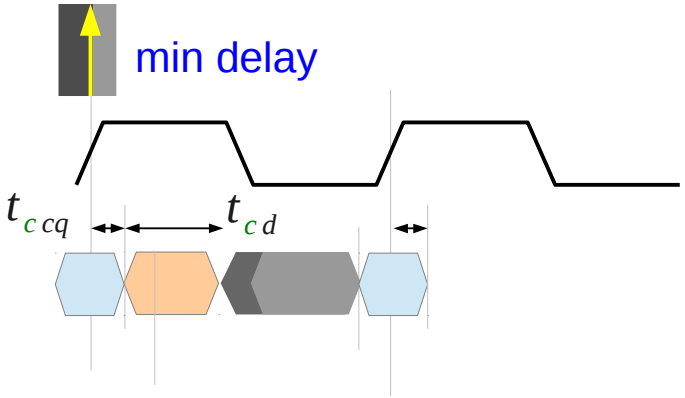
Setup Time OK



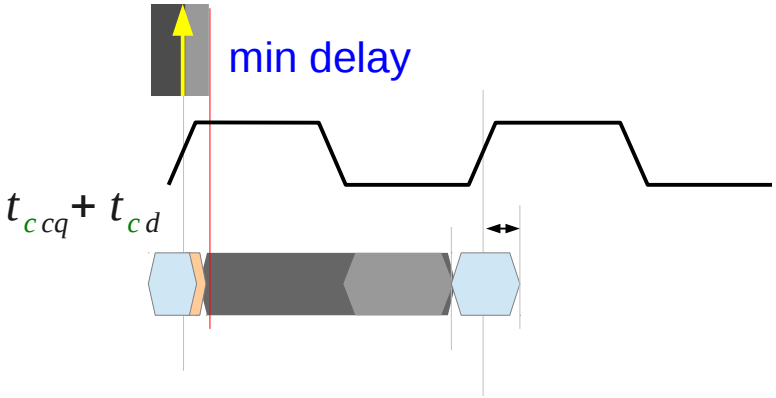
Setup Time <u>Violation</u>



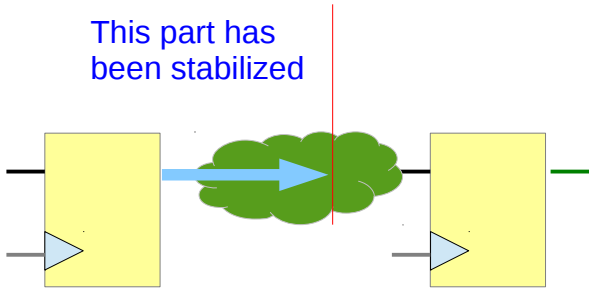
Hold Time OK



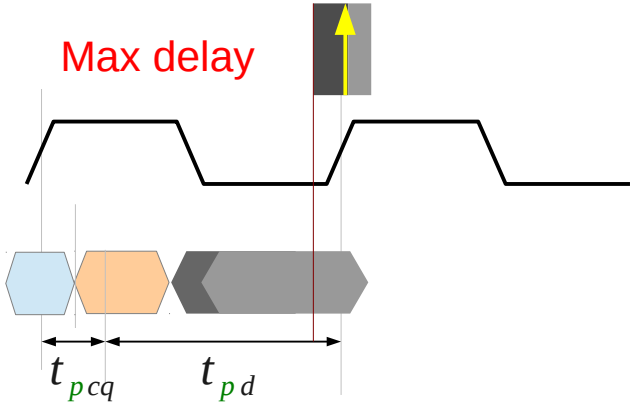
Hold Time <u>Violation</u>

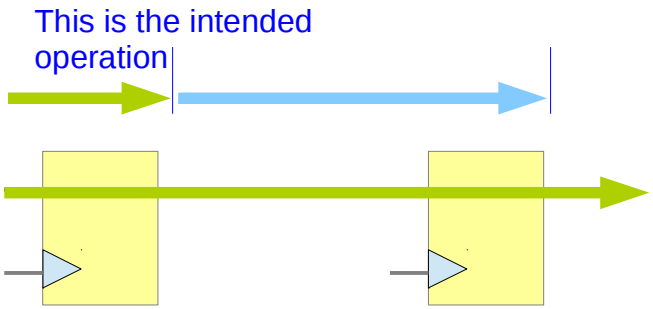


Setup Time / Hold Time



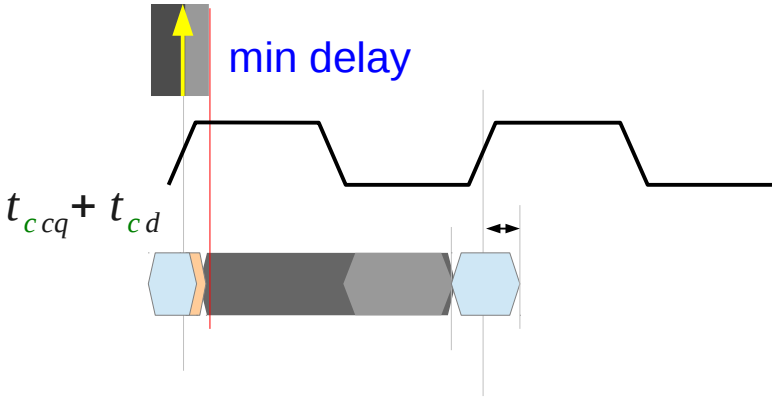
Setup Time Violation





Since the delay is too
small signal passes
through the 2nd FF

Hold Time Violation



Resolution Time

References

[1] http://en.wikipedia.org/
[2] M. M. Mano, C. R. Kime, “Logic and Computer Design Fundamentals”, 4th ed.
[3] J. Stephenson, Understanding Metastability in FPGAs. Altera Corporation white paper. July 2009.