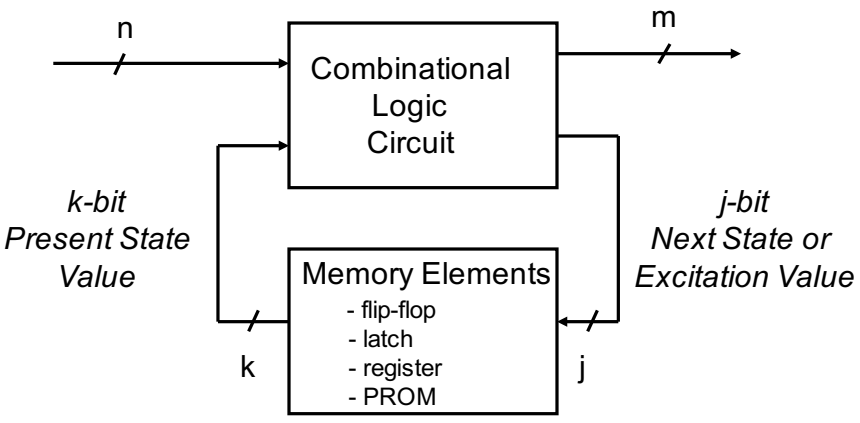
Sequential Systems Review

- Combinational Network
  - Output value only depends on input value
- Sequential Network
  - Output Value depends on input value and present state value
  - Sequential network must have some way of retaining state via memory devices.
  - can be synchronous or asynchronous
- Synchronous Sequential Network
  - Use a clock signal in a **synchronous** sequential system to control changes between states

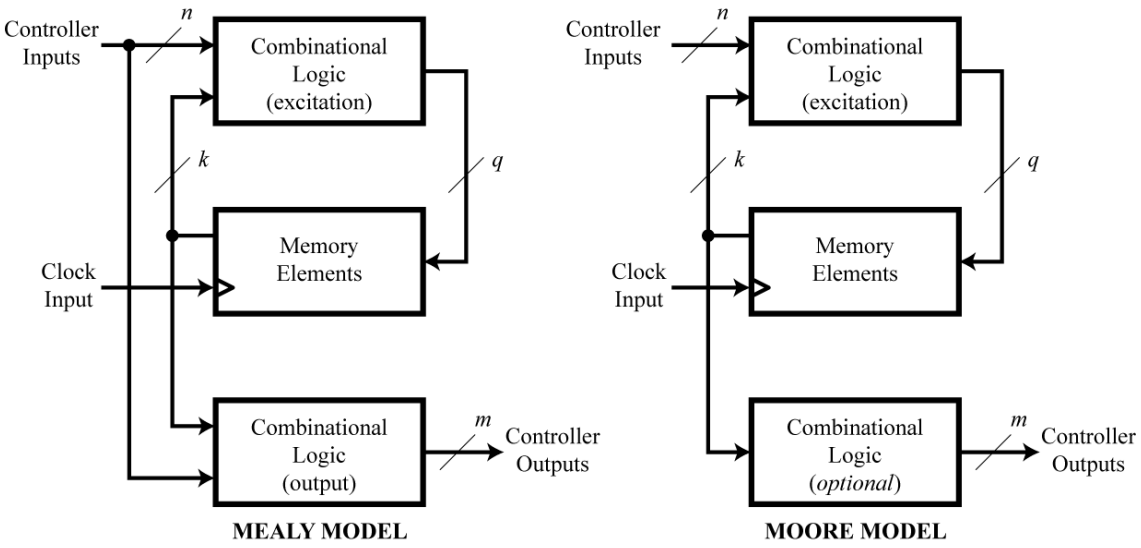
Sequential System Diagram



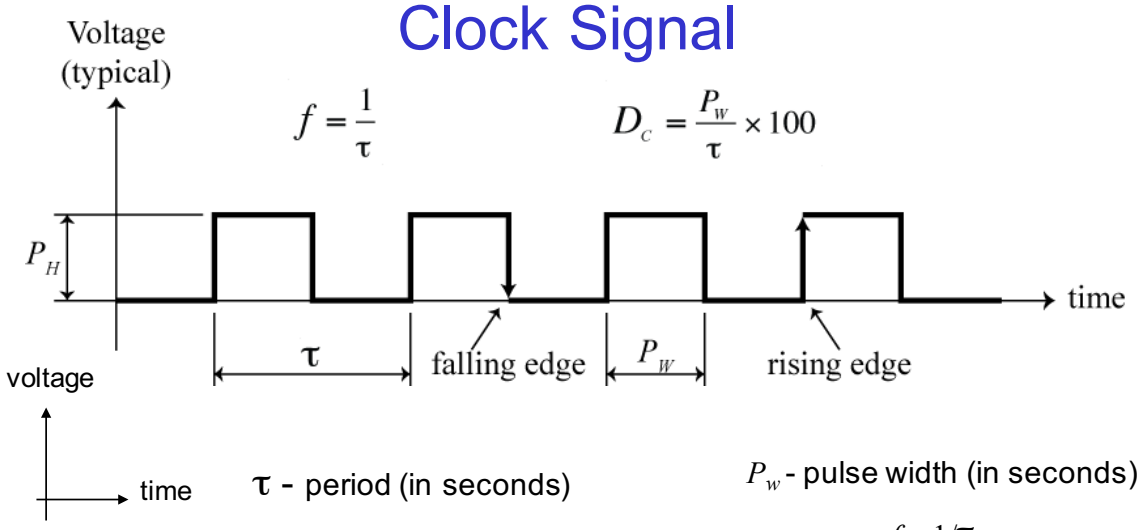
- m outputs only depend on k PS bits - Moore Machine
  - *REMEMBER: Moore is Less !!*

- m outputs depend on k PS bits AND n inputs - Mealy Machine

Controller Block Diagrams



Clock Signal



τ - period (in seconds)

P_w - pulse width (in seconds)

f - frequency (in Hertz)

$f = 1/\tau$

duty cycle - ratio of pulse width to period (in %)

$D_C = P_w/\tau$

millisecond (ms) 10^{-3}	Kilohertz (kHz) 10^{3}
microsecond (μs) 10^{-6}	Megahertz (MHz) 10^{6}
nanosecond (ns) 10^{-9}	Gigahertz (GHz) 10^{9}

Clock Signal Example

What is the pulse-width of a 4.77 MHz clock with a 30% duty cycle?

Clock Signal Example

What is the pulse-width of a 4.77 MHz clock with a 30% duty cycle?

$$\tau = 1/f = (4.77\times10^{6})^{-1} = 2.096\times10^{-7} = 210 \text{ ns}$$

Clock Signal Example

What is the pulse-width of a 4.77 MHz clock with a 30% duty cycle?

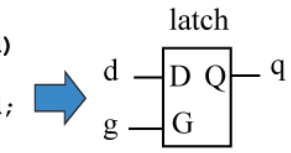
$$\tau = 1/f = (4.77\times10^{6})^{-1} = 2.096\times10^{-7} = 210 \text{ ns}$$

$$Pw = (\text{duty cycle}) \times \tau = (0.3) \times (210 \text{ ns}) = 63 \text{ ns}$$

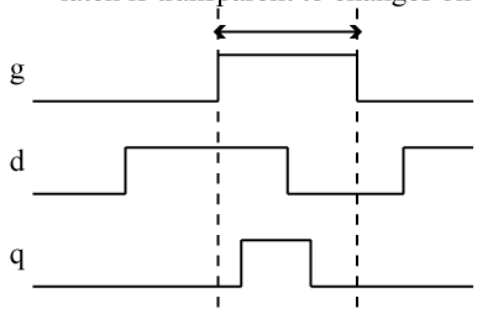
Storage Elements

(a) level-sensitive storage element
(D-latch)

```
always @(g or d)
begin
  if (g) q <= d;
end
```



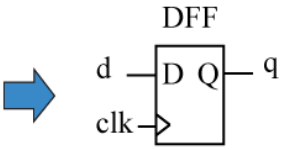
latch is transparent to changes on *d*



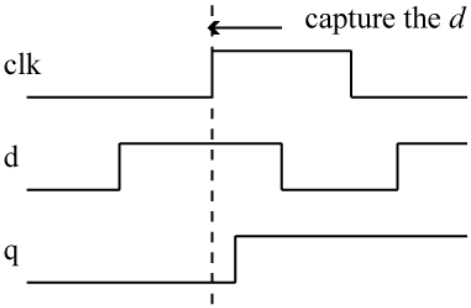
q follows *d* when *g* is high

(b) edge-triggered storage element
(data flip-flop, or DFF)

```
always @(posedge clk)
begin
  q <= d;
end
```

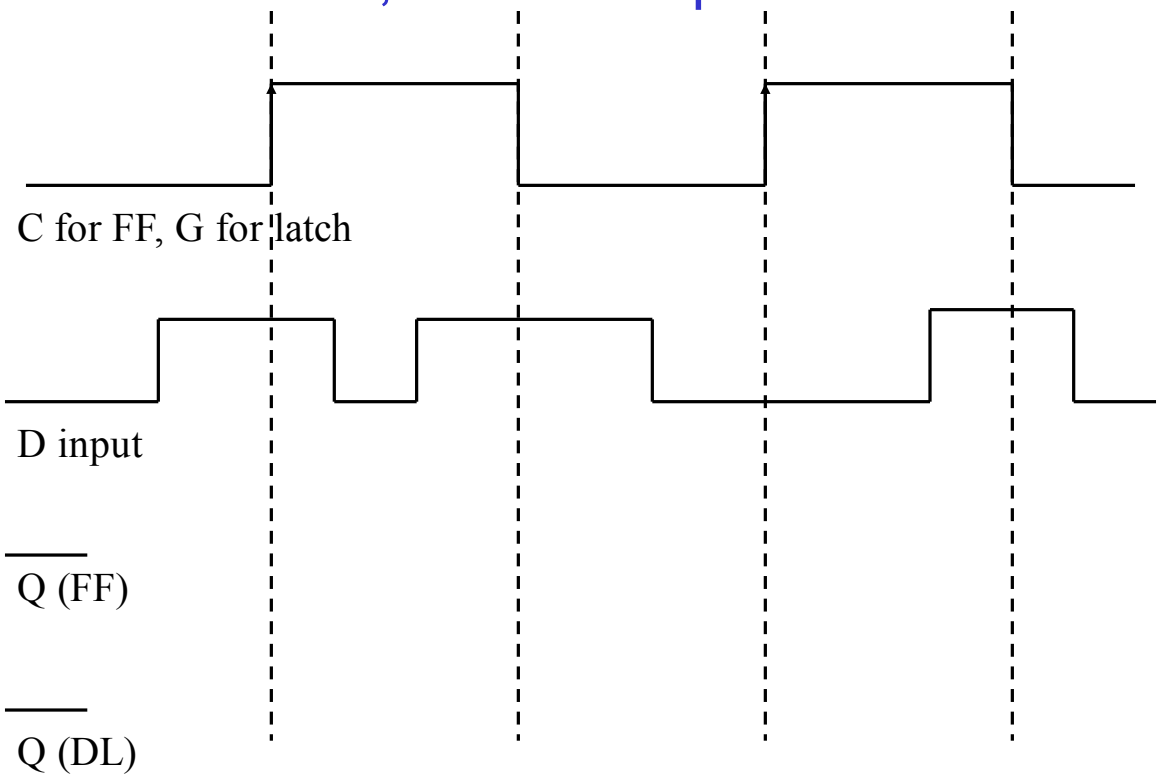


capture the *d* input

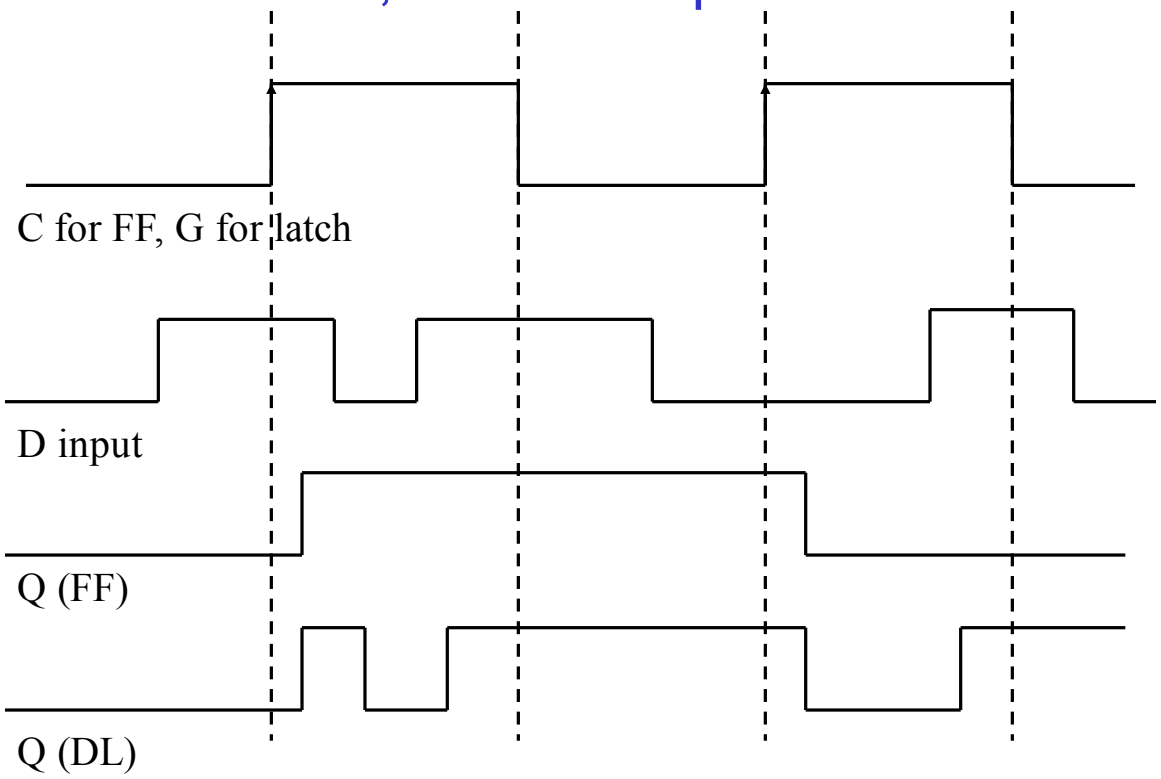


q follows *d* on rising edge of *clk*

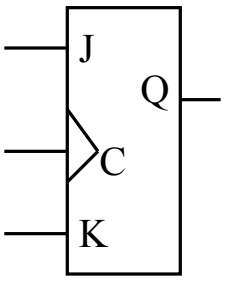
D FF, D Latch operation



D FF, D Latch operation



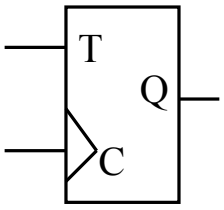
Other State Elements



J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

JK can be useful for single bit flags with separate set(J), reset(K) control.

RARELY USED



T	Q(t+1)
0	Q(t)
1	Q'(t)

Can be useful for asynchronous counter design.

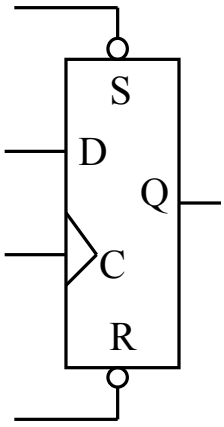
RARELY USED

DFFs are most common

- Most FPGA families only have DFFs
- DFF is fastest, simplest (fewest transistors) of FFs
- Other FF types (T, JK) can be built from DFFs
- We will use DFFs almost exclusively in this class
  - Will always use edge-triggered state elements (FFs), not level sensitive elements (latches).

Synchronous vs Asynchronous Inputs

Synchronous input: Output will change after active clock edge
Asynchronous input: Output changes independent of clock

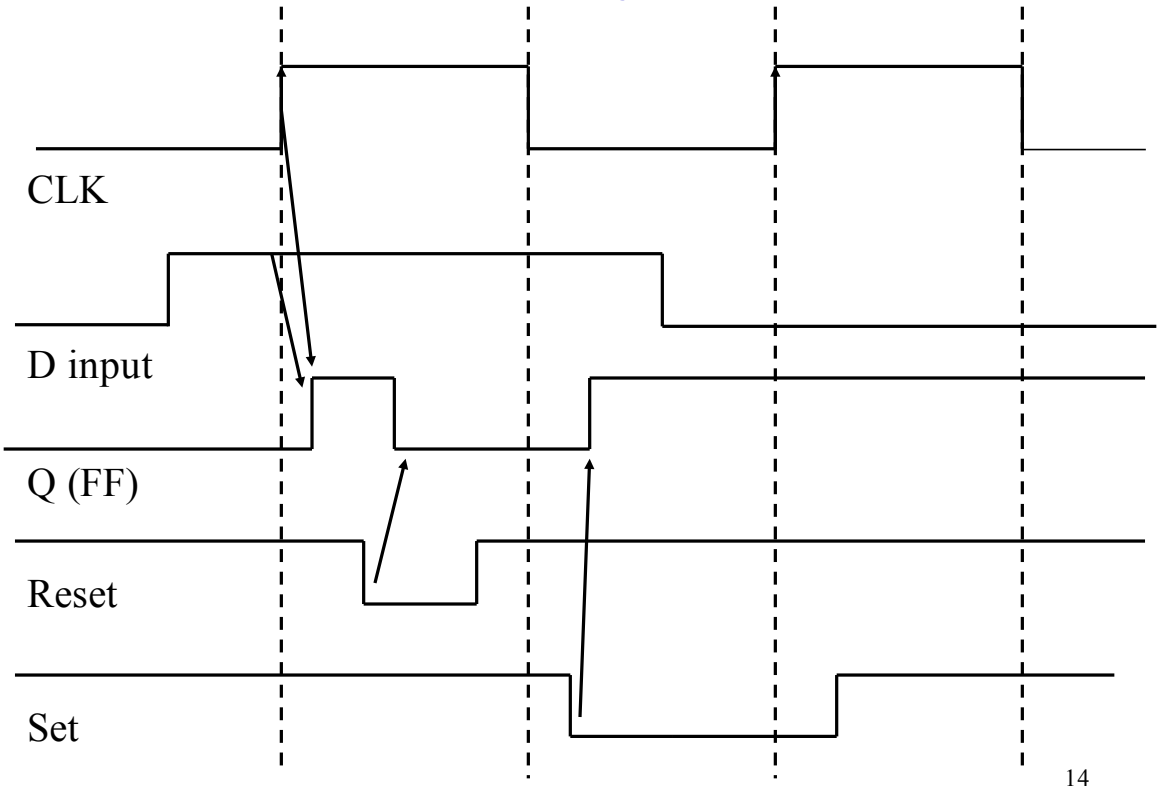


State elements often have async set, reset control.

D input is synchronous with respect to Clk

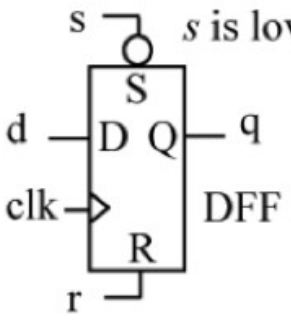
S, R are asynchronous. Q output affected by S, R independent of C. Async inputs are dominant over Clk.

D FF with async control



D-Flip-Flop with Async. Set/Res

(a) DFF with asynchronous Set/Reset

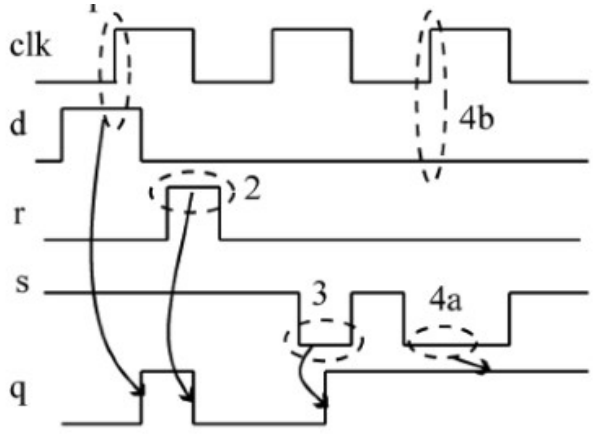


s is low-true, *r* is high-true

r	*s*	*clk*	*q*
1	x	x	0
0	0	x	1
0	1	↑	d
0	1	1 or 0	q_{old}

```
always @(posedge clk or
         posedge r or
         negedge s)
begin
  if (r) q <= 1'b0;
  else if (!s) q <= 1'b1;
  else q <= d;
end
```

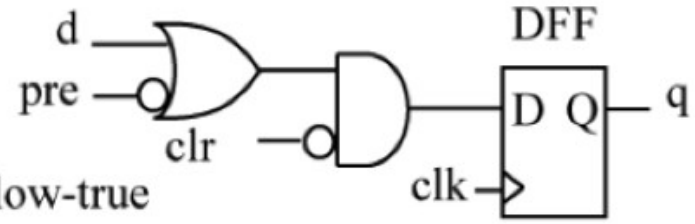

D-Flip-Flop with Async. Set/Res



1. *d* input is captured on rising clock edge as *r*, *s* are negated; *q* becomes '1'.
2. *r* input is asserted; *q* becomes '0'.
3. *s* input is asserted; *q* becomes '1'.
4. assertion of *s* input (a) overrides clock input (b), so *q* output remains as '1'.

D-Flip-Flop with Sync. Set/Res

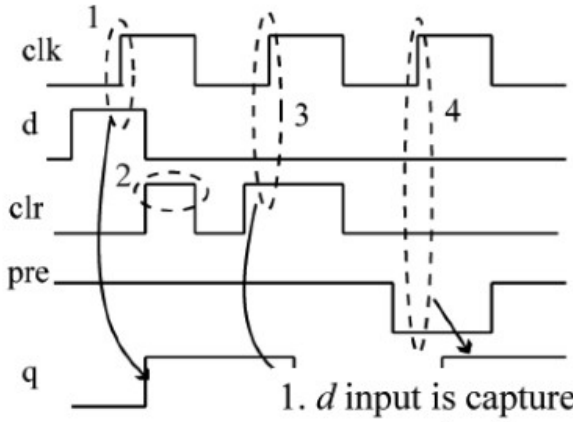
(b) DFF with synchronous preset/clear



pre is low-true
clr is high-true
clr has priority over *pre*

```
always @(posedge clk)
begin
  q <= d;  //lowest priority
  if (!pre) q <= 1'b1;
  if (clr) q <= 1'b0; //highest priority
end
```


D-Flip-Flop with Sync. Set/Res



1. *d* input is captured on rising clock edge as *clr*, *pre* are negated; *q* becomes '1'.
2. assertion of synchronous input has no effect if does not occur on the active clock edge
3. *clr* input is asserted on rising clock edge; *q* becomes '0'.
4. *pre* input is asserted on rising clock edge; *q* becomes '1'.

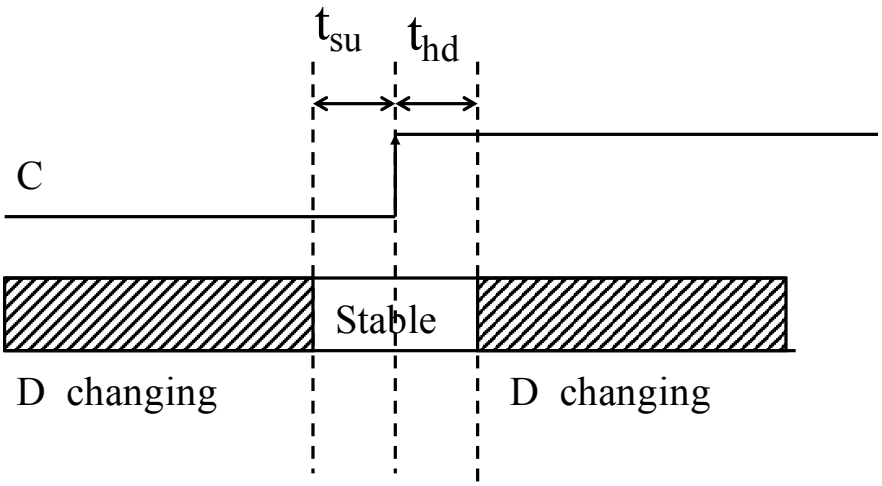
FF Timing

- Propagation Delay
  - C2Q: Q will change some propagation delay after change in C. Value of Q is based on D input for DFF.
  - S2Q, R2Q: Q will change some propagation delay after change on S input, R input
  - Note that there is NO propagation delay D2Q for DFF!
  - D is a Synchronous INPUT, no prop delay value for synchronous inputs

Setup, Hold Times

- Synchronous inputs (e.g. D) have Setup, Hold time specification with respect to the CLOCK input
- Setup Time: the amount of time the synchronous input (D) must be *stable before* the active edge of clock
- Hold Time: the amount of time the synchronous input (D) must be *stable after* the active edge of clock.

Setup, Hold Time



If changes on D input violate either setup or hold time, then correct FF operation is not guaranteed.

Setup/Hold measured around active clock edge.