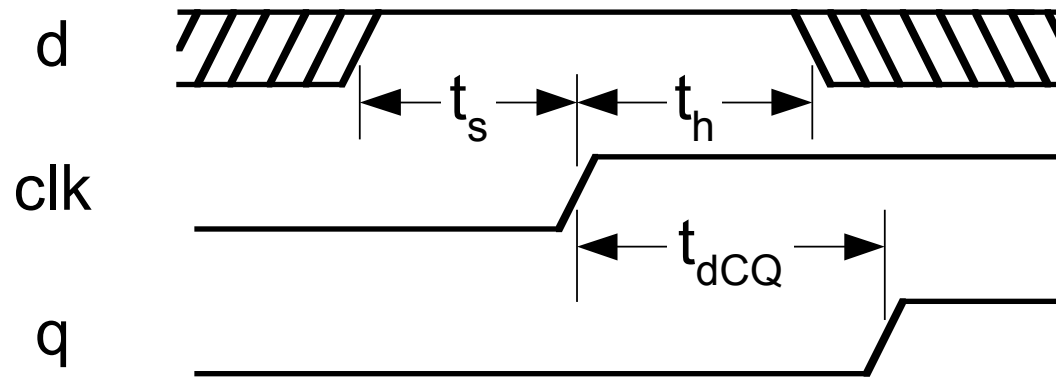
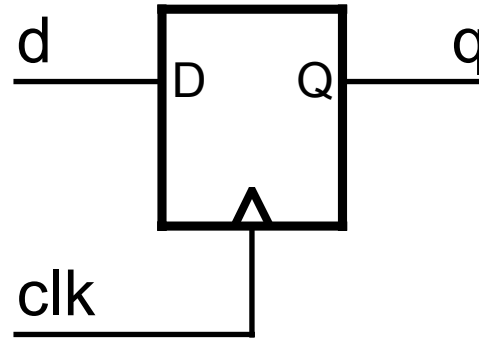

Digital Design: A Systems Approach

Lecture 13: Metastability and Synchronization Failure (or When Good Flip-Flops go Bad)

Readings

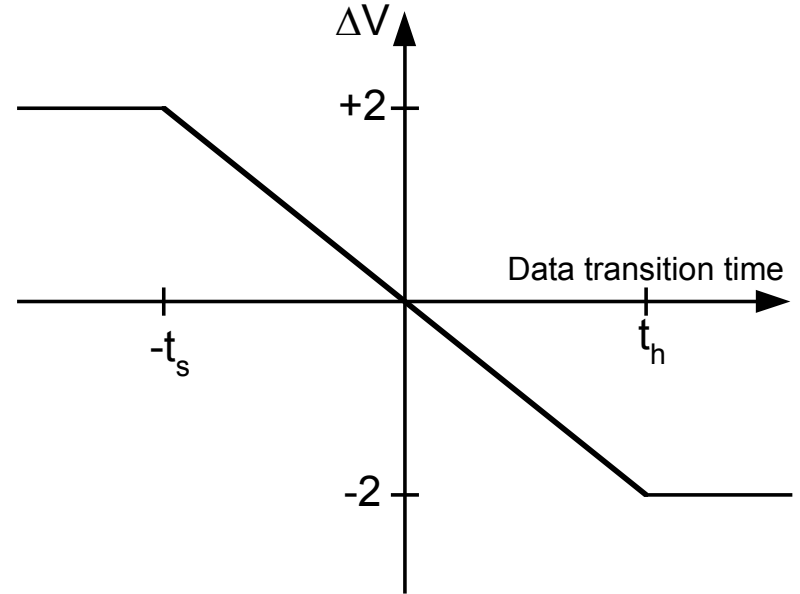
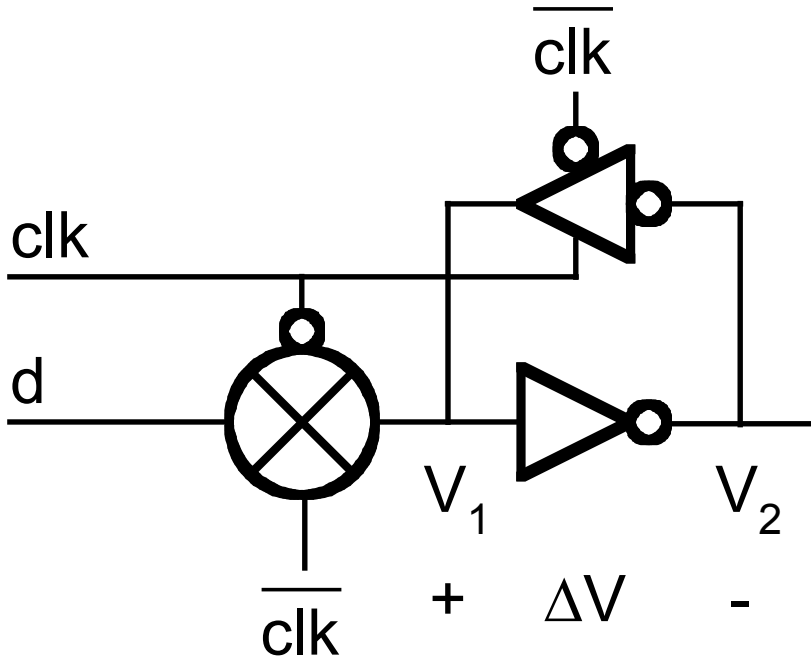
- L13: Chapters 27 & 28
- L14: Sequential Logic Review

What happens when we violate setup and hold time constraints?

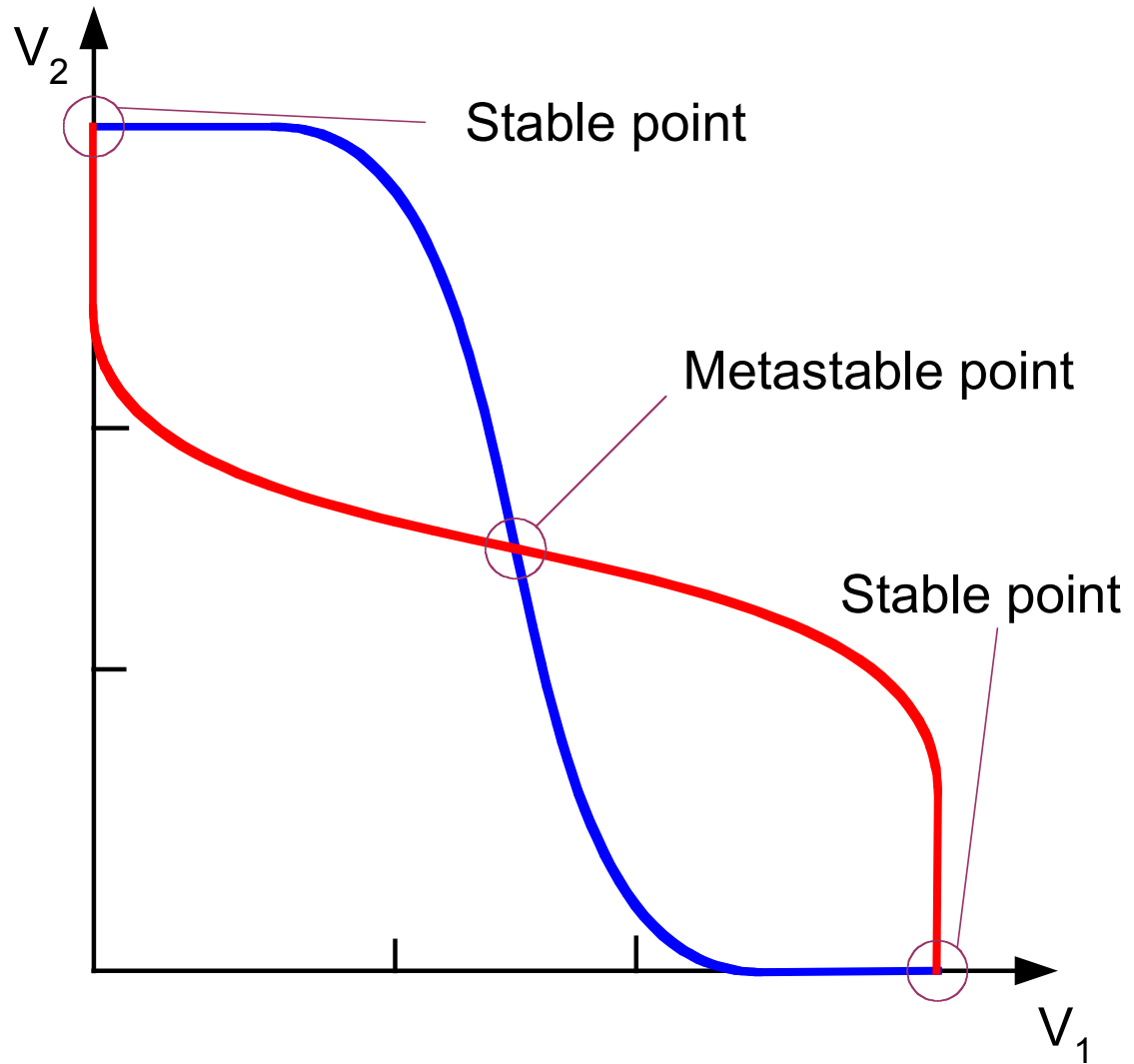
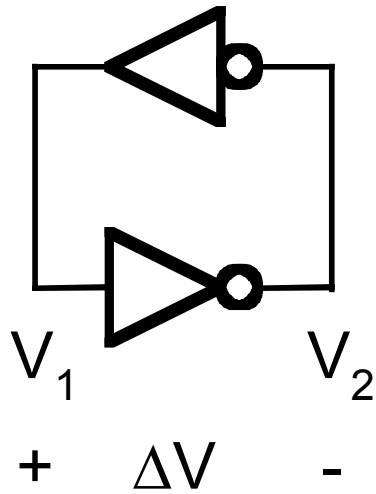


Look at structure of CMOS latch

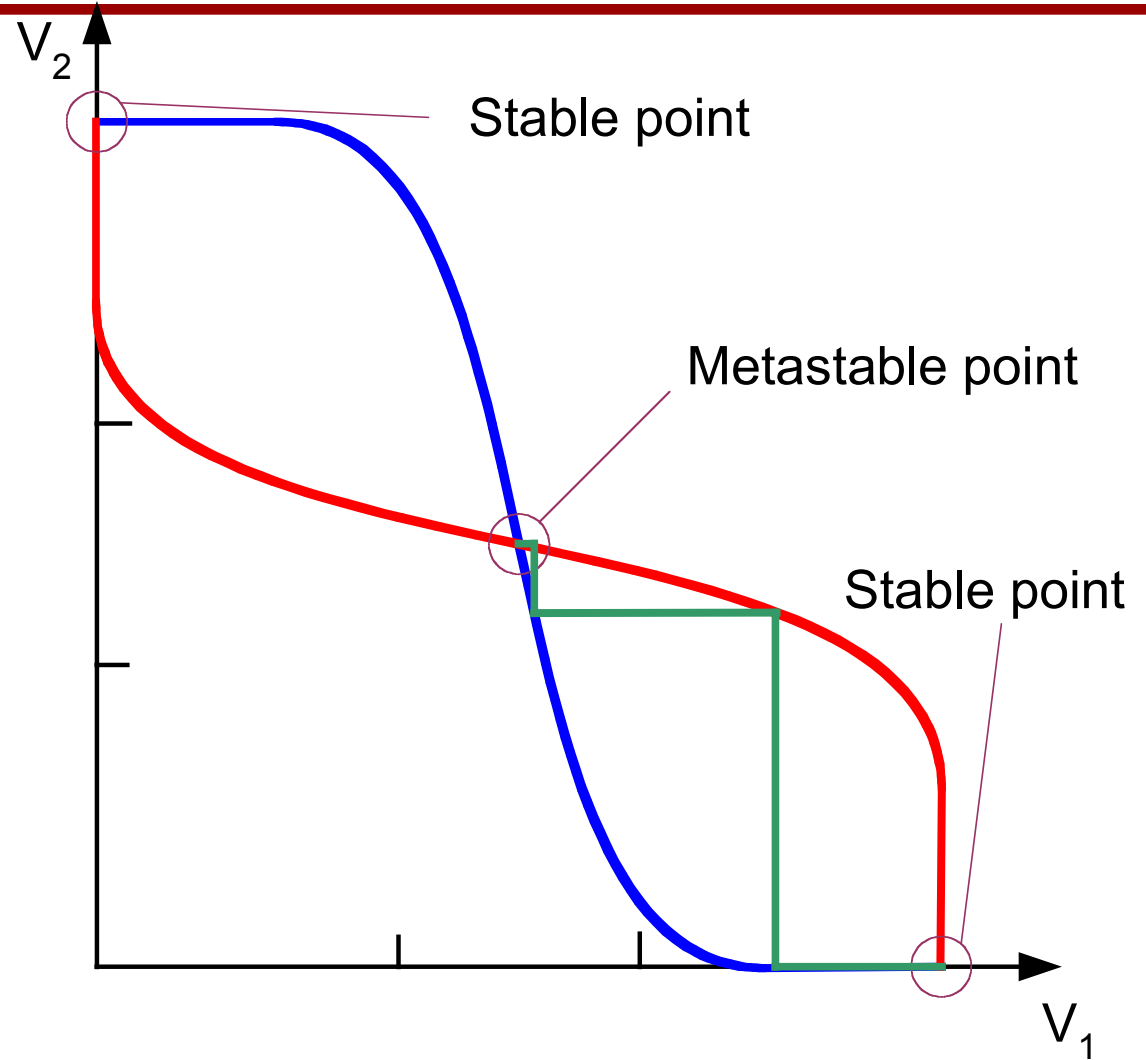
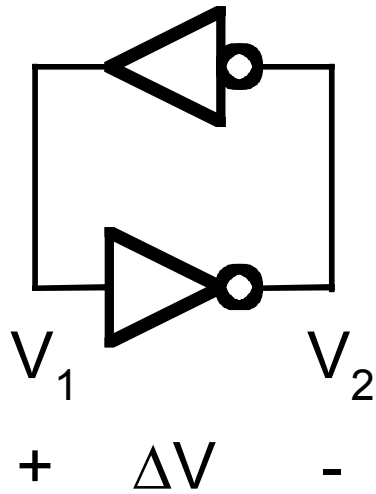
- Storage loop gets initialized with an ‘analog’ value
- Latch is a “time-to-voltage” converter



Storage loop has a *metastable* state between 0 and 1



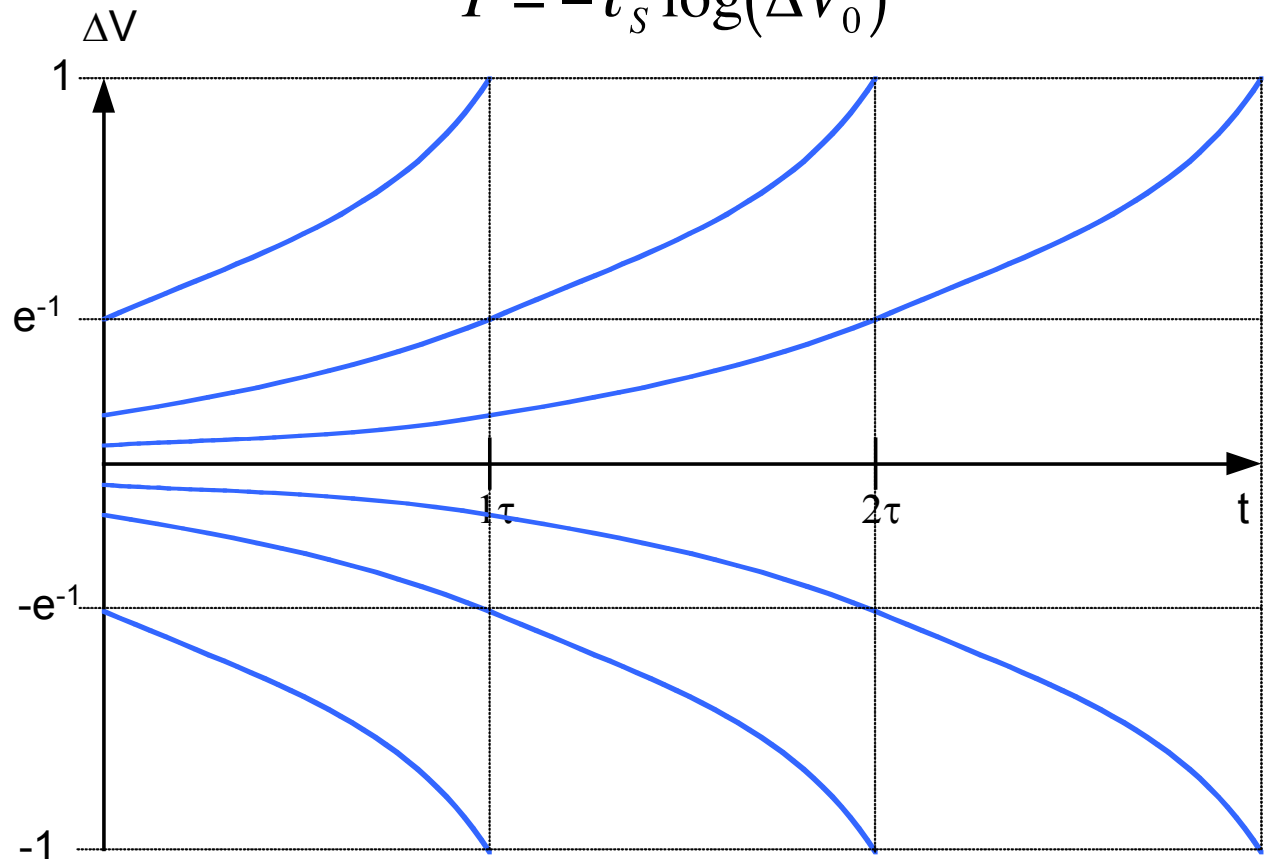
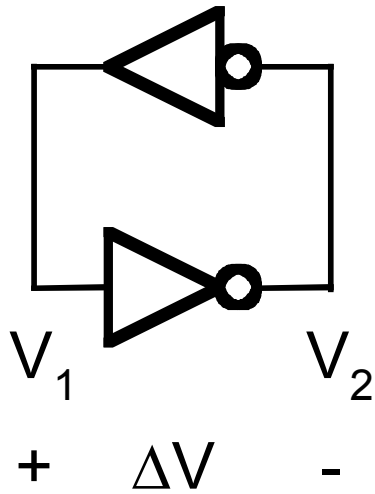
Dynamics of ΔV



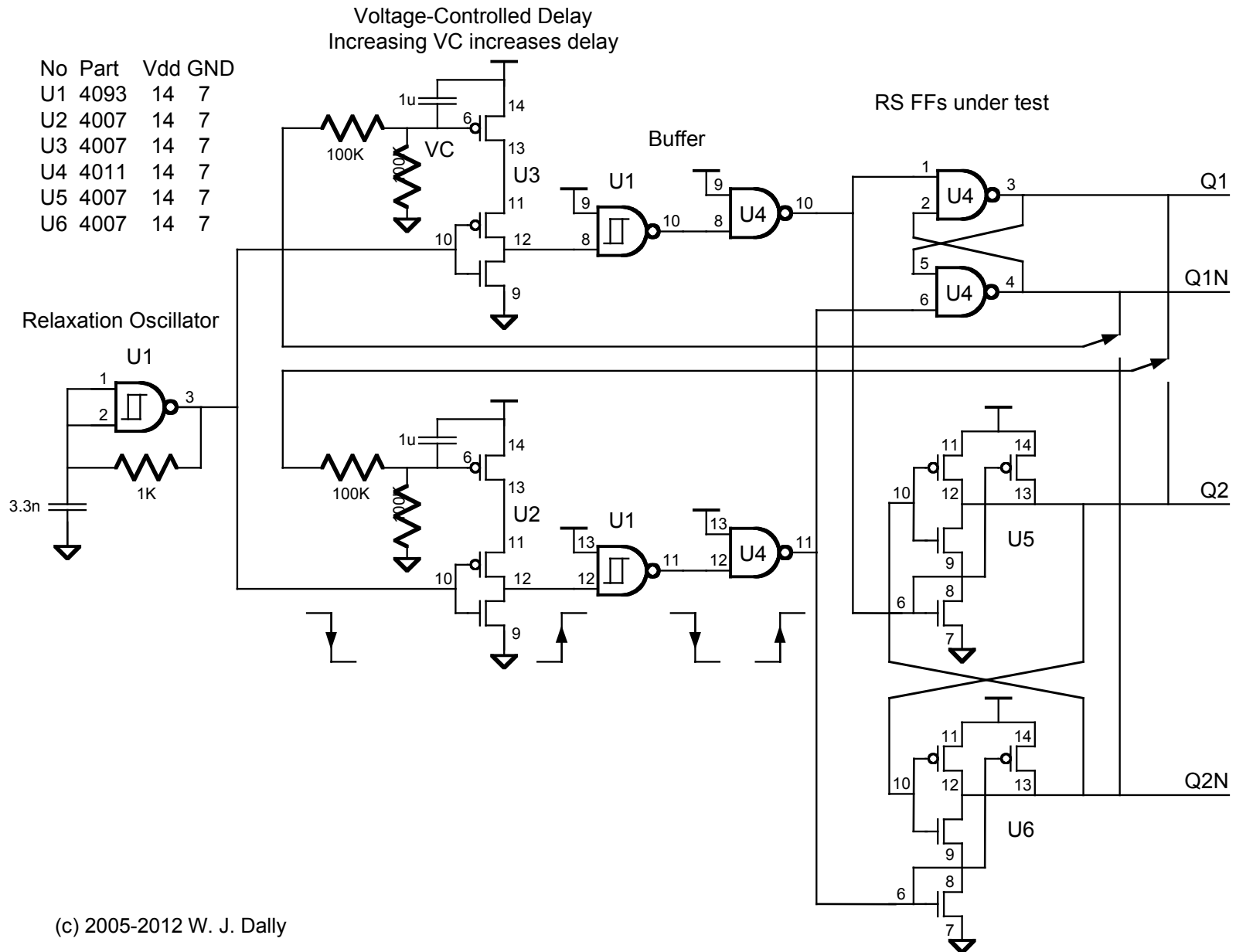
Dynamics of ΔV

$$\Delta V(t) = \Delta V_0 \exp\left(\frac{t}{\tau_s}\right)$$

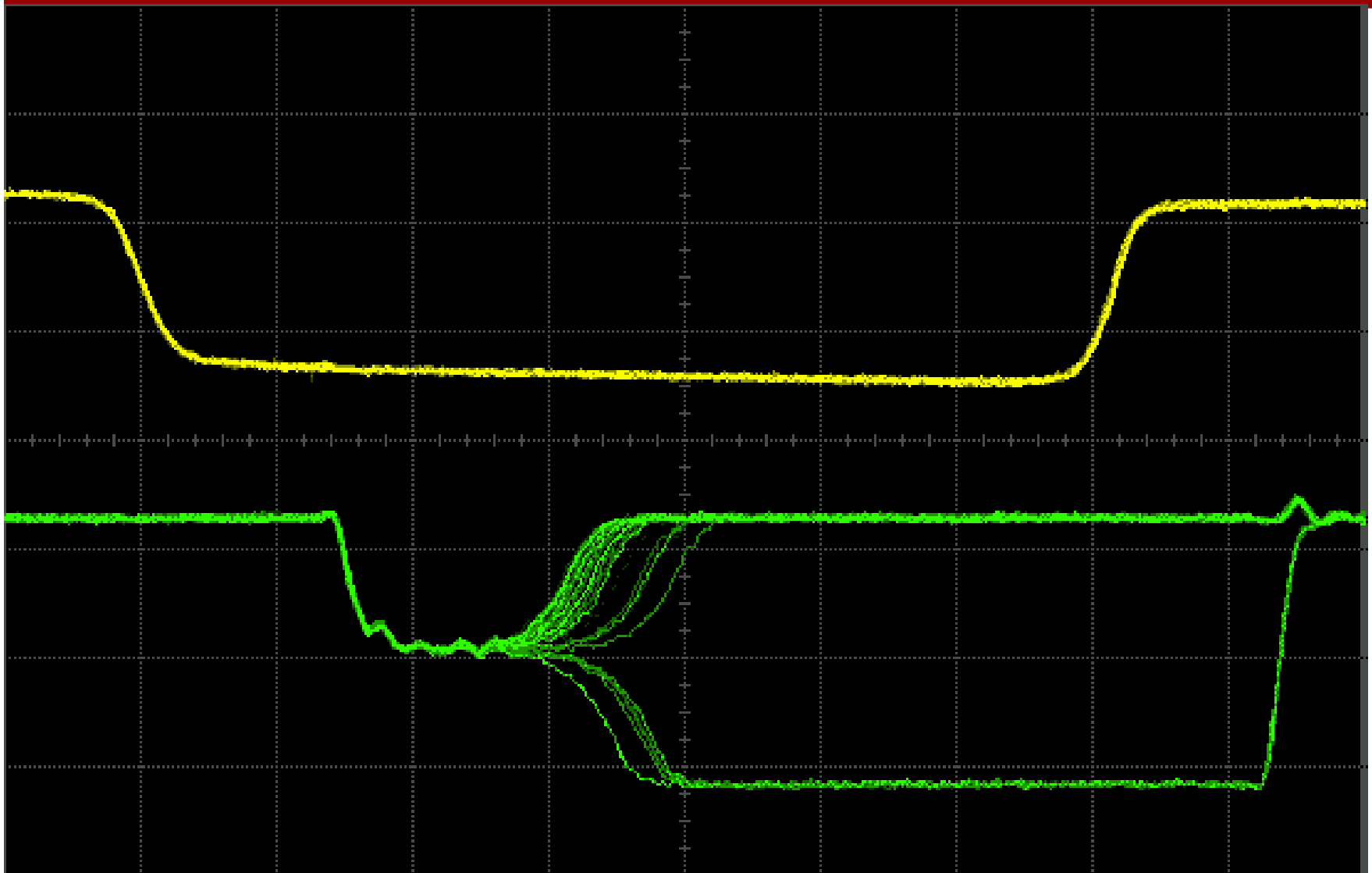
$$T = -\tau_s \log(\Delta V_0)$$



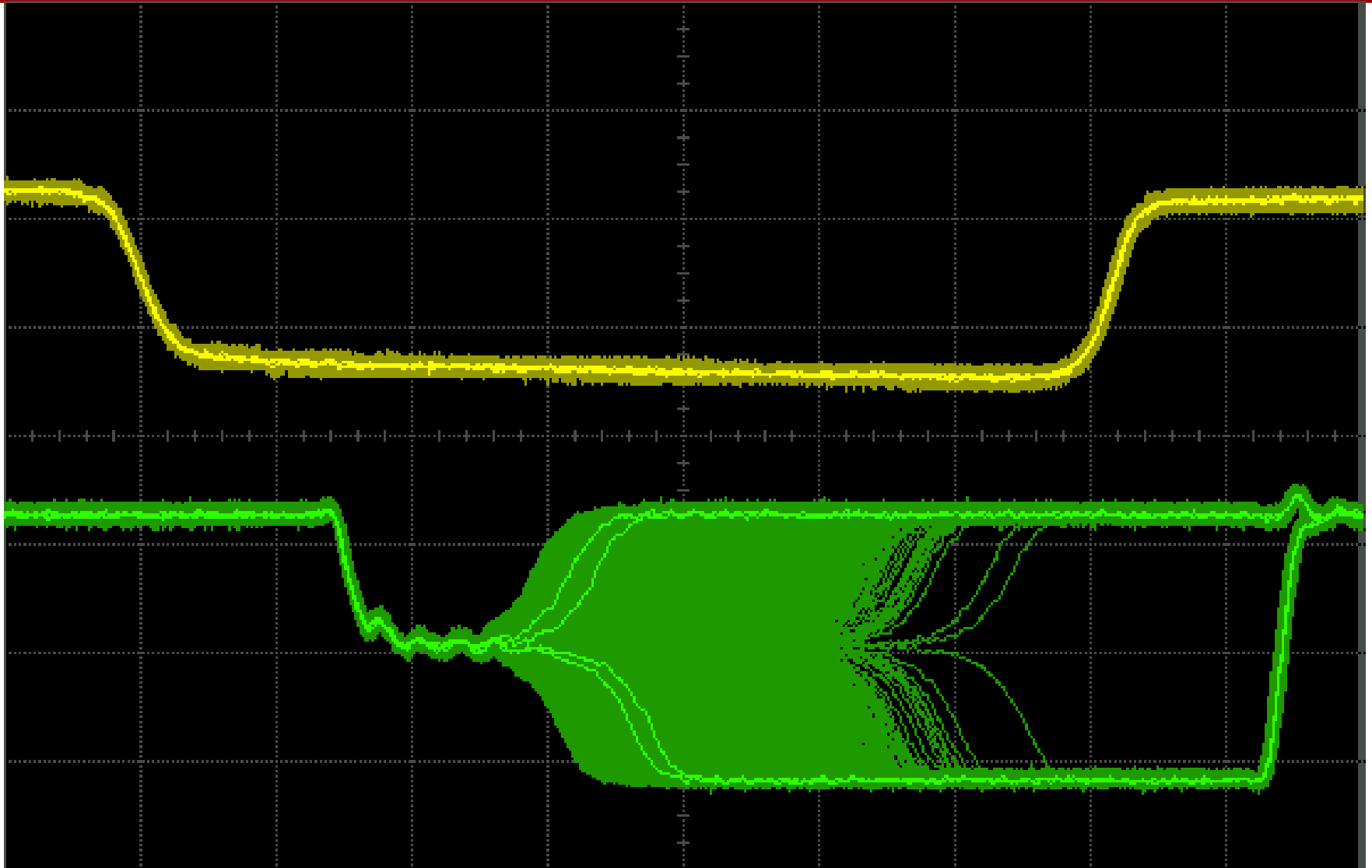
Metastability Demonstration Circuit



Metastable state of FF1 – 4007 Nand RS Latch

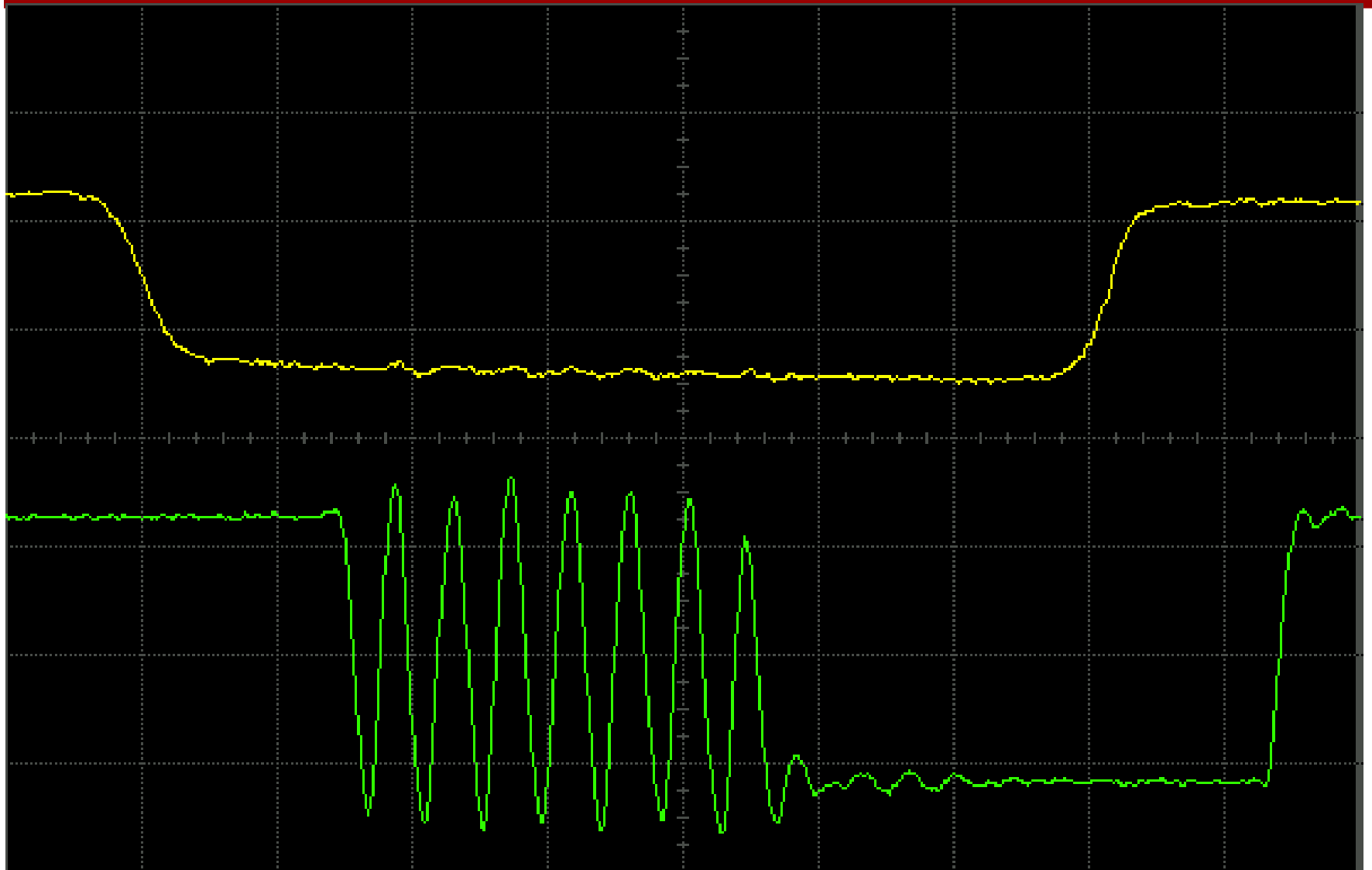


Over time the waveform fills in

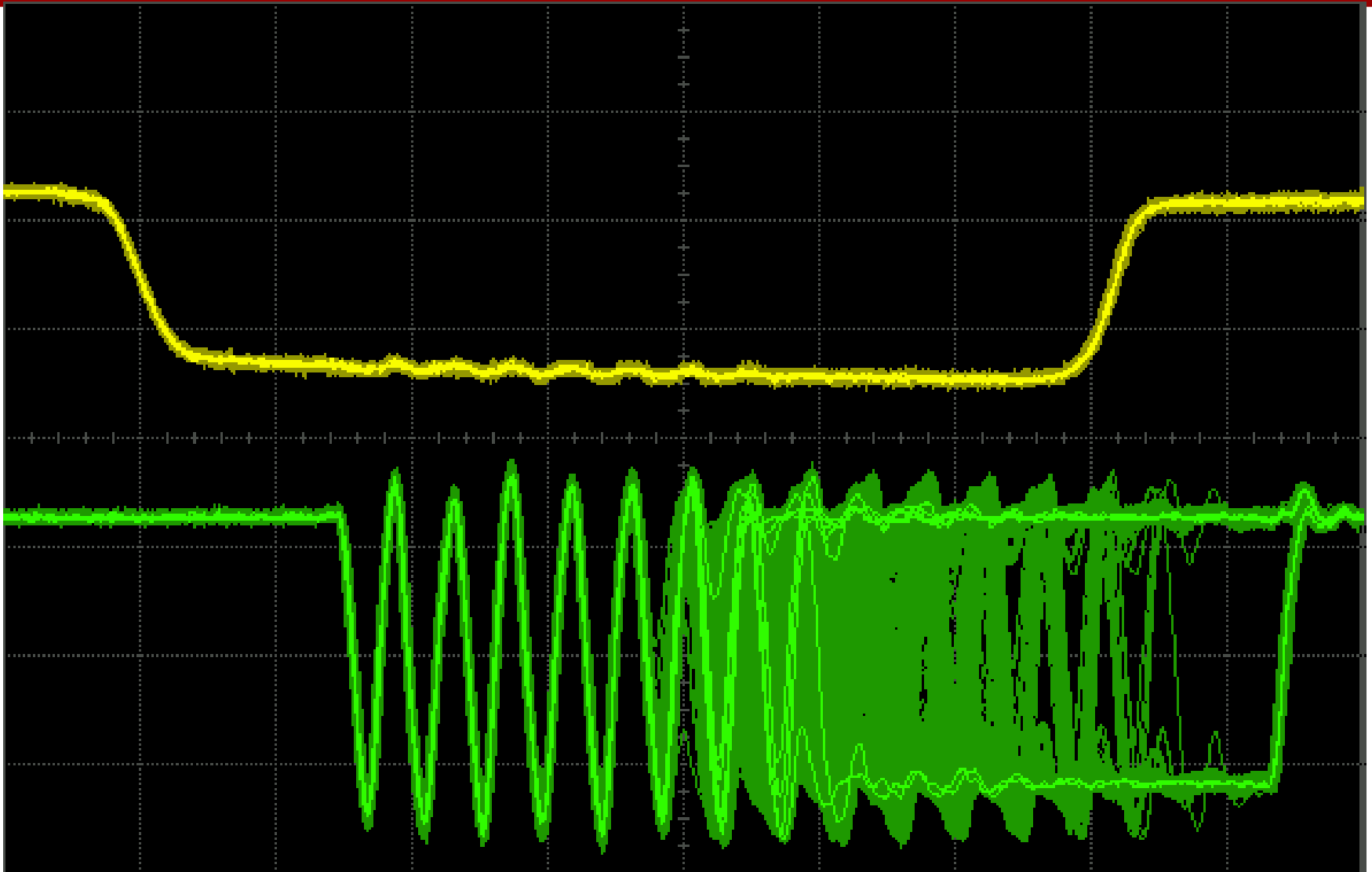


Metastable state of 4011 Nand RS Latch

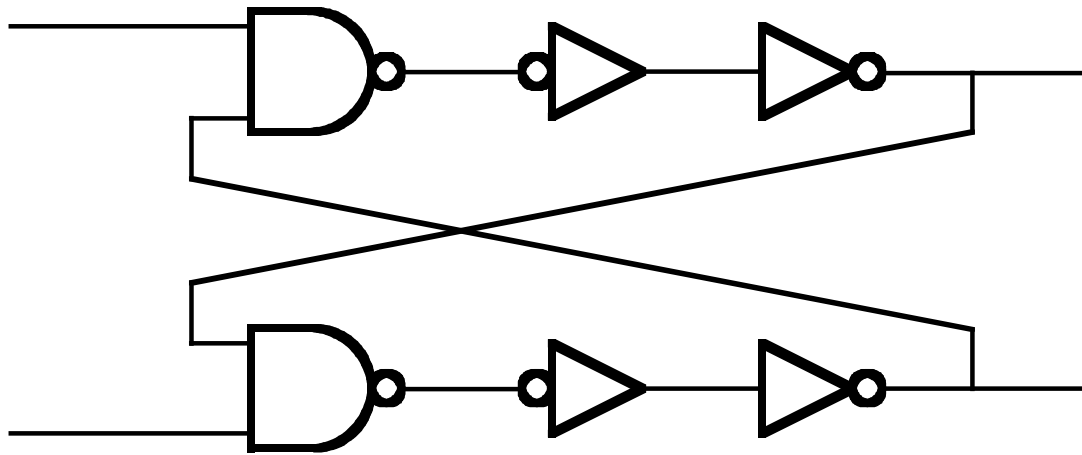
What's going on here?



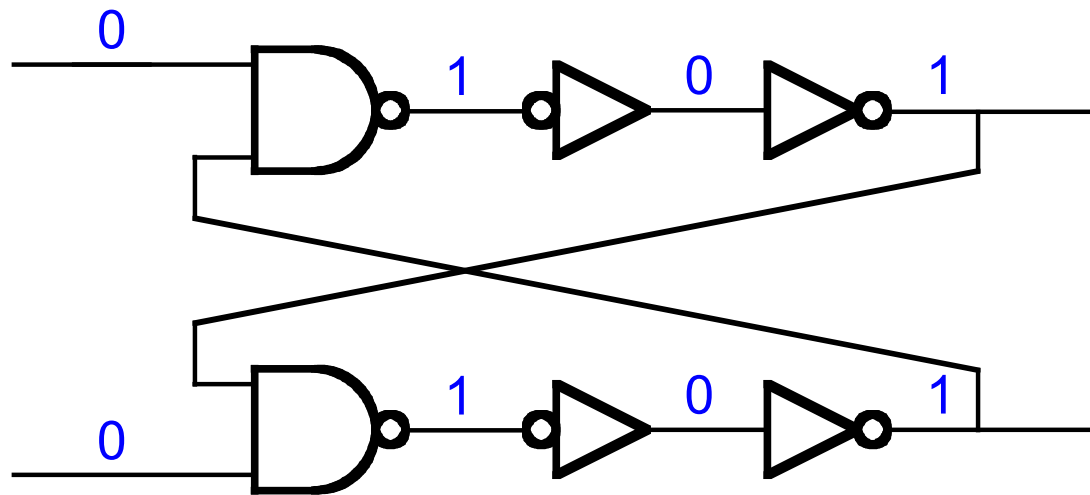
Over time this waveform fills in too



Actual circuit of 4011 Nand RS Latch

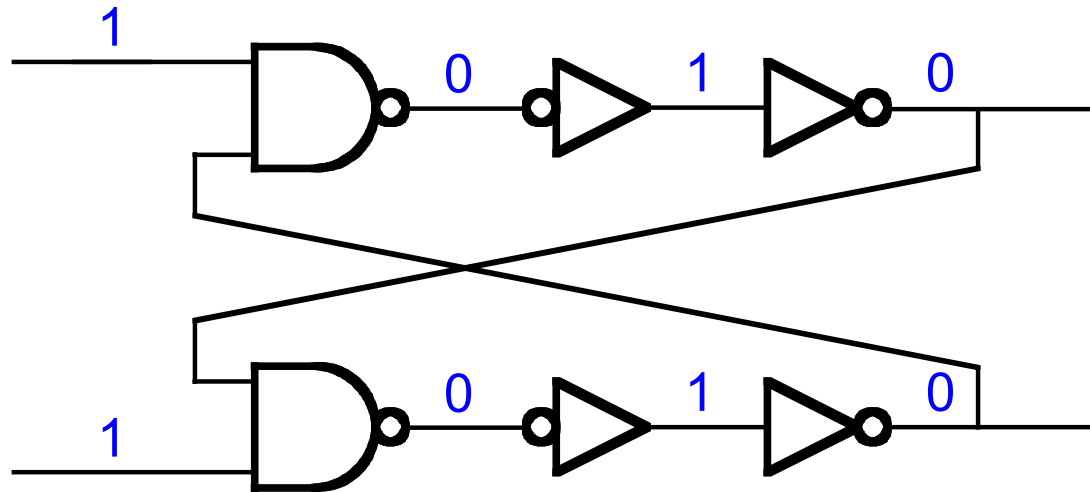


Initial state when both inputs are low

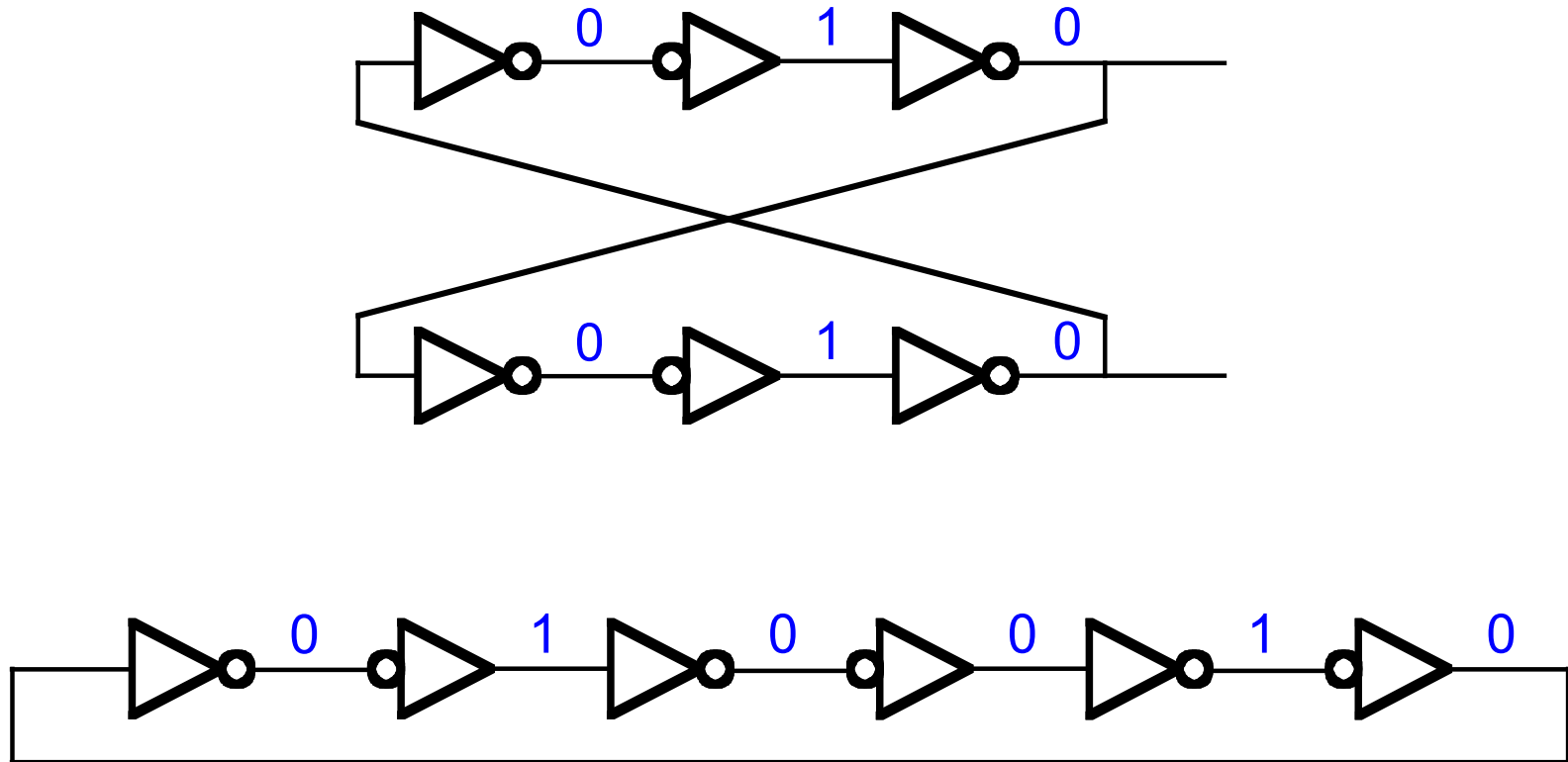


When both inputs go high, it becomes a 6-stage ring oscillator

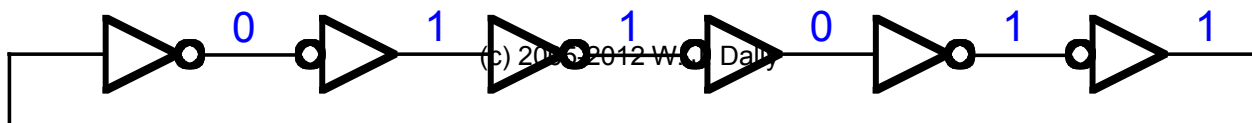
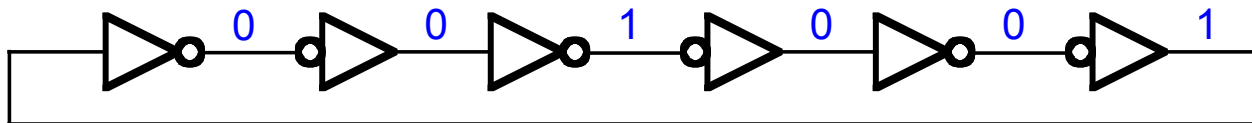
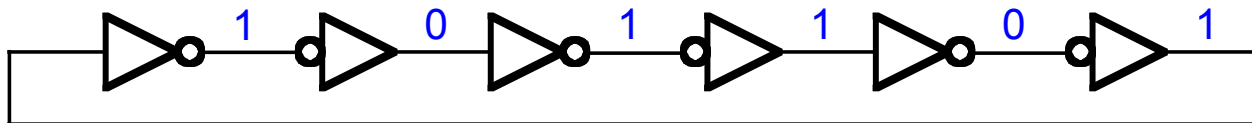
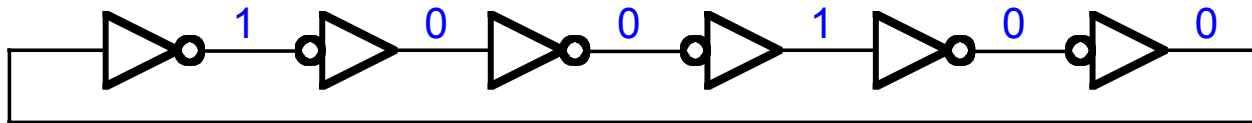
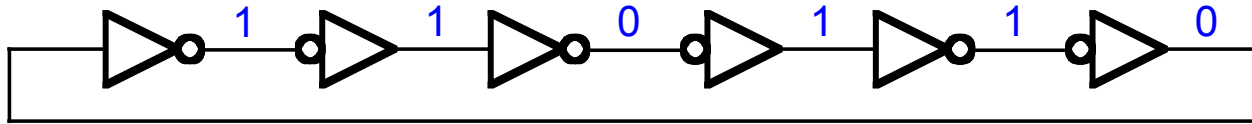
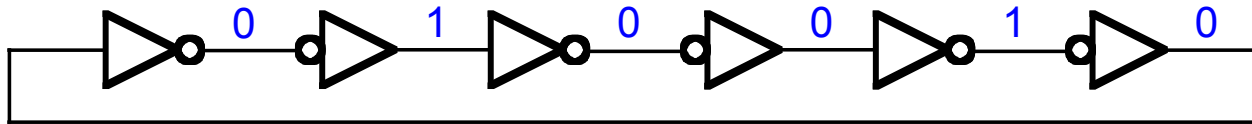
Oscillation is metastable



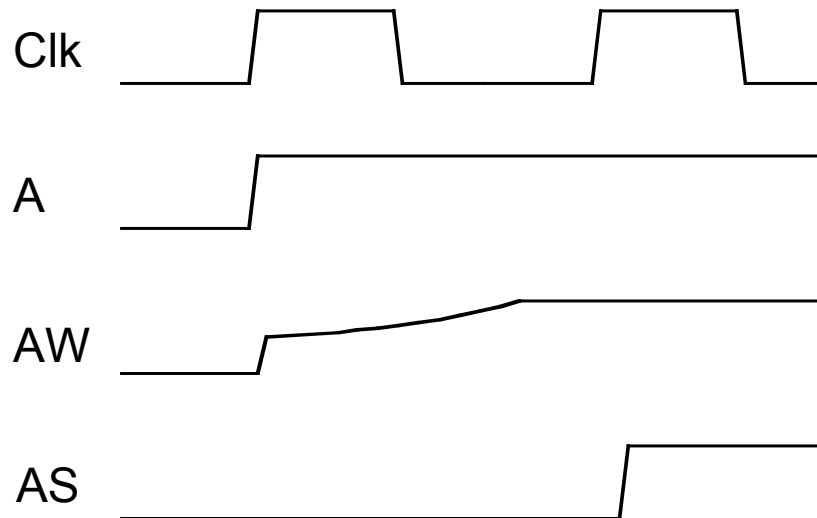
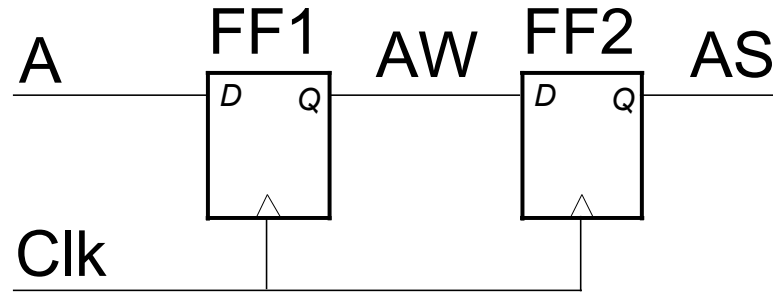
Oscillation is metastable



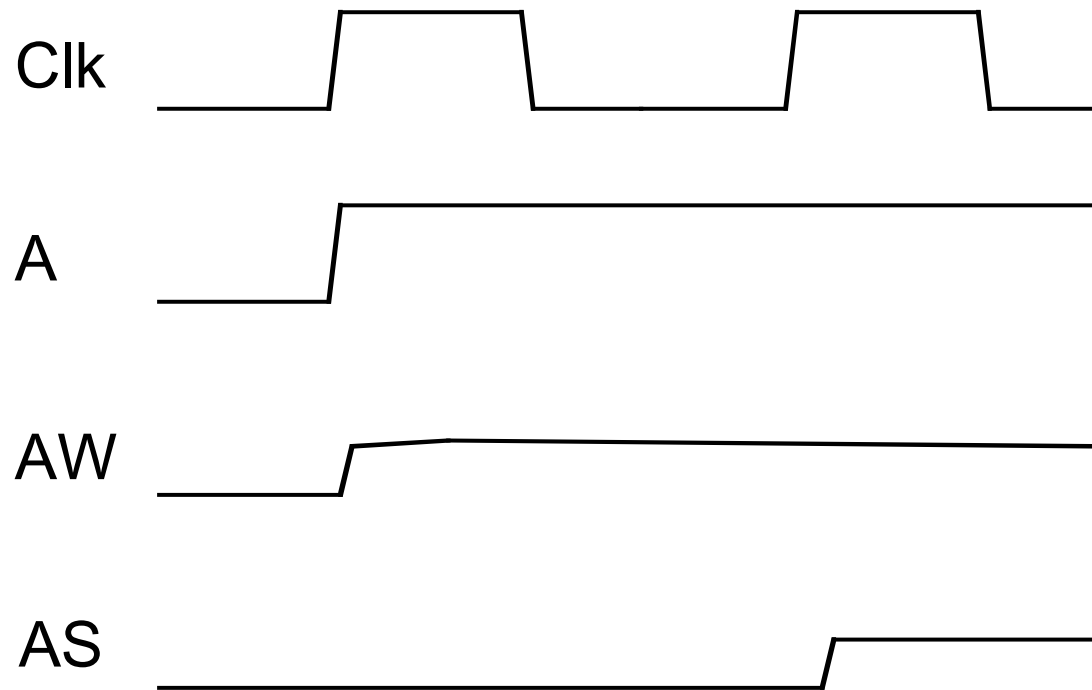
If delays are balanced, ring sequences through six states repeatedly



A Brute-Force Synchronizer



What if *AW* is *still* in a metastable state when FF2 is clocked?

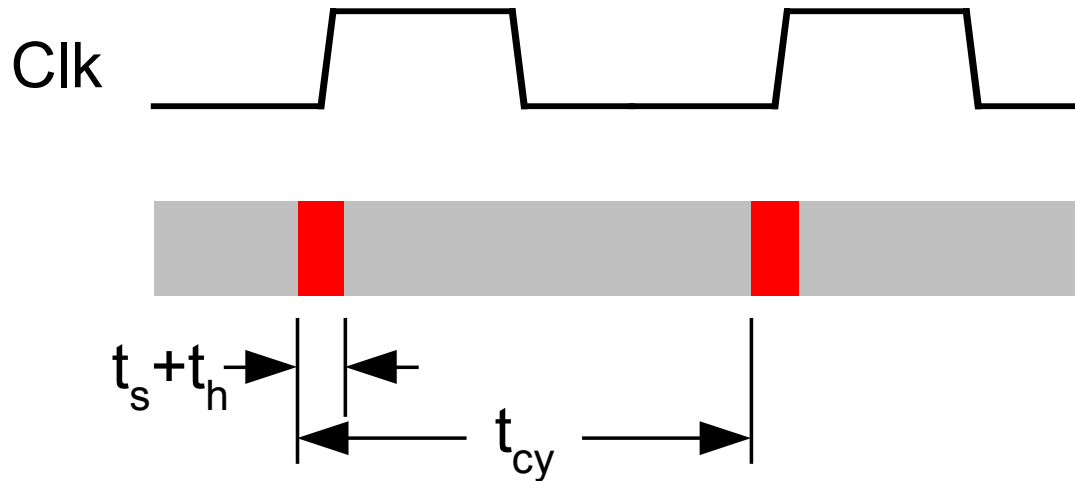


Calculating Synchronization Failure (The Big Picture)

$$P(\text{failure}) = P(\text{enter metastable state}) \times P(\text{still in state after } t_w)$$

Probability of Entering a Metastable State

- FF1 may enter the metastable state if the input signal transitions during the *setup+hold window* of the flip flop
- Probability of a given transition being in the setup+hold window is the fraction of time that *is* setup+hold window



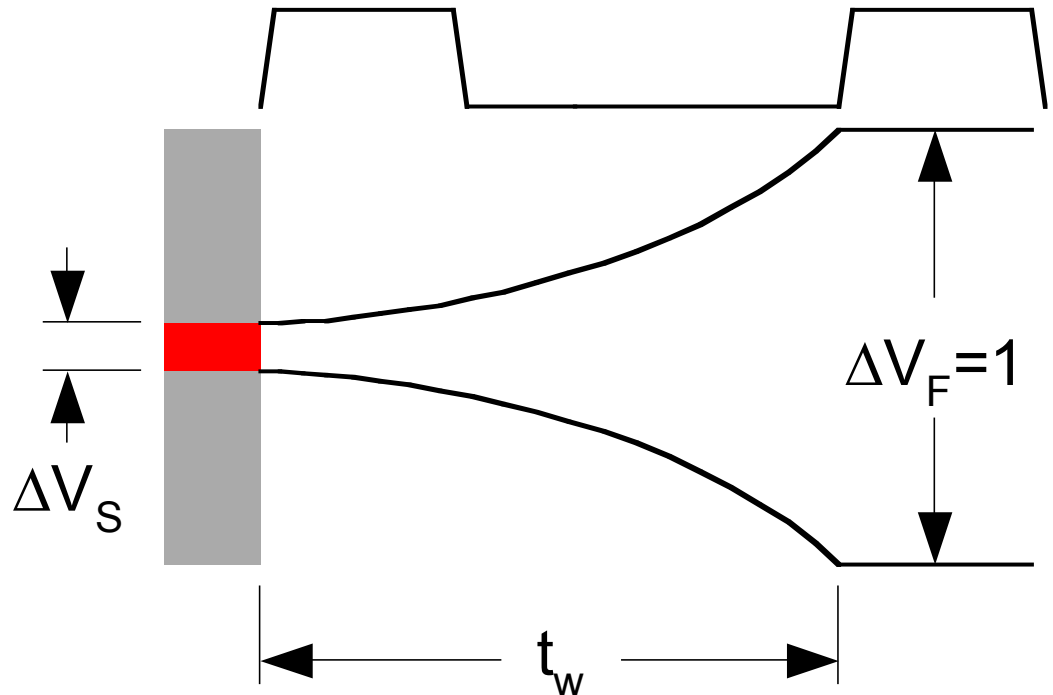
$$P_E = \frac{t_s + t_h}{t_{cy}} = f_{cy} (t_s + t_h)$$

Probability of Staying in the Metastable State

- Still in metastable state if initial voltage difference was too small to be exponentially amplified during wait time
- Probability of starting with this voltage is proportion of total voltage range that is 'too small'

$$\Delta V_S = \Delta V_F \exp\left(\frac{-t_w}{\tau_S}\right)$$

$$P_S = \exp\left(\frac{-t_w}{\tau_S}\right)$$

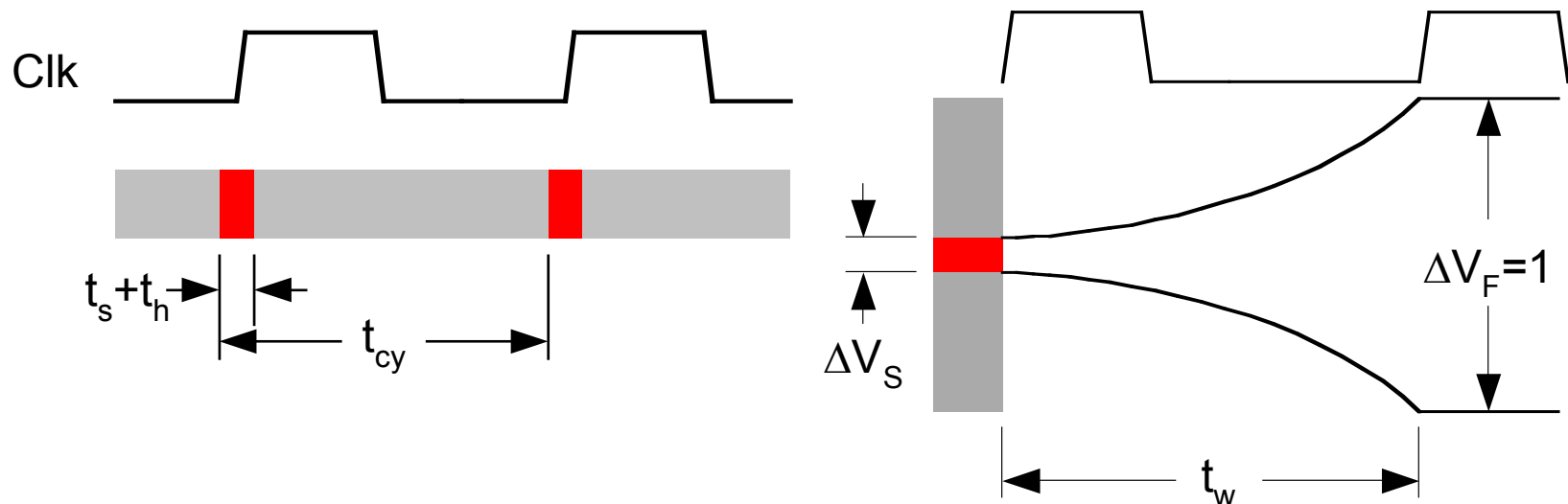


Failure Probability and Error Rate

- Each event can potentially fail.
- Failure rate = event rate x failure probability

$$P_F = P_E P_S = (t_s + t_h) f_{cy} \exp\left[-\frac{t_w}{\tau}\right]$$

$$f_F = f_e P_F = (t_s + t_h) f_e f_{cy} \exp\left[-\frac{t_w}{\tau}\right]$$

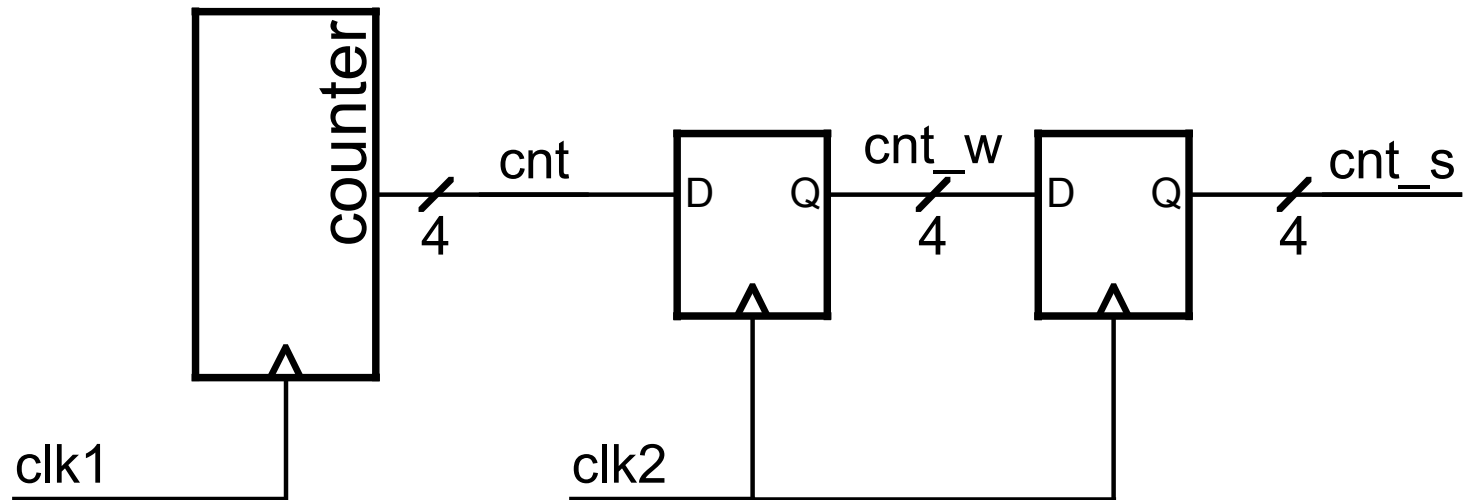


Example

- $t_s = t_h = t_{dCQ} = \tau = 100\text{ps}$
- $t_{cy} = 2\text{ns}$
- must sample a $f_E = 1\text{MHz}$ asynchronous signal
- $P_E = (.1+.1)/2 = 0.1$
- $P_S = \exp(-1.8/.1) = \exp(-18) = 1.5 \times 10^{-8}$
- $P_F = P_S P_E = 1.5 \times 10^{-9}$
- $f_F = f_E P_F = 1.5 \times 10^{-3}$
- 1 failure every 656 seconds ~ every 11 minutes
- This is not adequate. How do we improve it?
- How do we get failure rate to one every 10 years ~ $3 \times 10^8\text{s}$ ($f_F < 3 \times 10^{-9}$)

Synchronizing multi-bit signals

Consider a 4-bit counter running on `clk1` you need the value of this counter sampled by `clk2`. Will the following circuit work? (assume $t_w \gg \tau$)



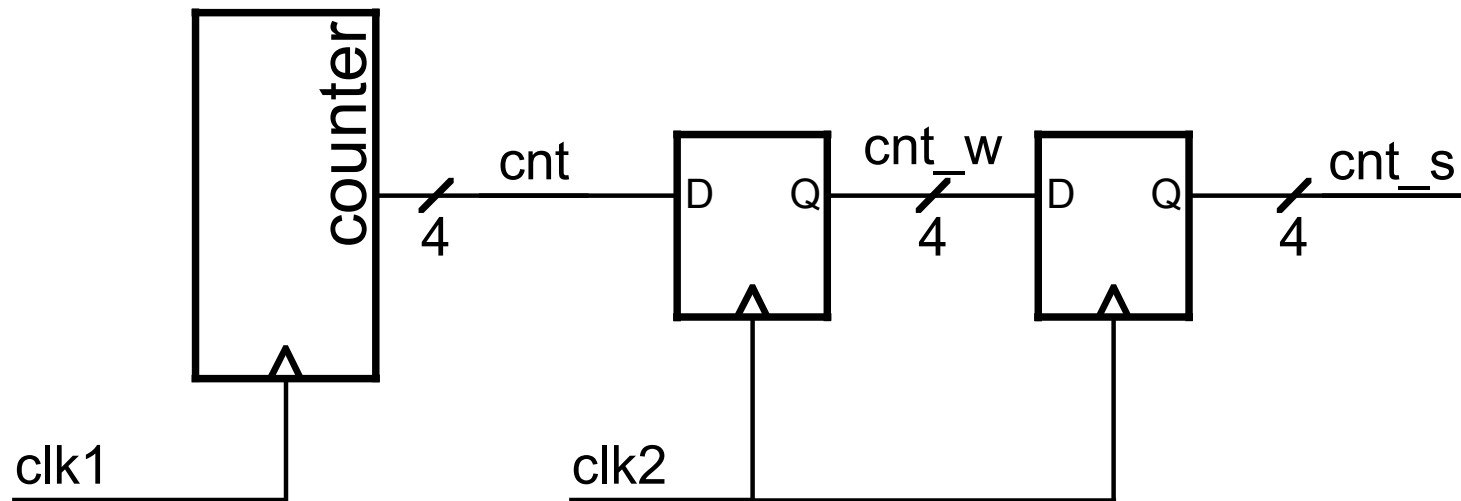
This happens, for example, in a FIFO where the head and tail pointers are in different *clock domains*.

Multi-bit signals (2)

When synchronizing a multi-bit signal, each changing bit is *independently* synchronized

Consider what happens on the 0111 to 1000 transition. All bits are changing. Each can independently fall either way.

How do you fix this?



Warning: The Surgeon General has determined that passing binary-coded and one-hot signals through a brute-force synchronizer can be hazardous to your circuits.

Solution:

Use a Gray code counter

For all but the MSB:

```
next_b[i] = (b[i-1] & !(|b[i-2:0])) ? !xor(b[n-1:i+1]) : b[i];
```

For the MSB:

```
next_b[i] = (|b[i-2:0]) ? b[i] : b[i-1] ;
```

Can we use this Gray code for the head and tail pointers of our FIFO?

```
module GrayCount4(clk, rst, out) ;
  input clk, rst ;
  output [3:0] out ;
  wire [3:0] out, next ;

  DFF #(4) count(clk, next, out) ;

  assign next[0] = !rst & !(out[1]^out[2]^out[3]) ;
  assign next[1] = !rst & (out[0] ? !(out[2]^out[3]) : out[1]) ;
  assign next[2] = !rst & ((out[1] & !out[0]) ? !out[3] : out[2]) ;
  assign next[3] = !rst & (!(|out[1:0]) ? out[2] : out[3]) ;
endmodule
```

xxxx
0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000
0000
0001
0011
0010

Metastability and Synchronization Failure Summary

- Clocking a flip-flop during the “keepout” interval may leave the storage node in an “illegal state”
- Some “illegal states” are **Metastable**
- Time to decay to a legal state depends on log of initial voltage

$$t = -\tau \log(\Delta V(0))$$

- Probability of entering metastable state is probability of hitting “keepout” interval.

$$P_E = \frac{t_s + t_h}{t_{cy}}$$

- Probability of staying in metastable state after time T is probability that initial voltage was too small to decay in time T

$$P_S = \exp\left[-\frac{t_W}{\tau}\right]$$

- Brute-force synchronizer – sample signal and wait for metastable states to decay.
- Don't use on multi-bit signals unless they are Gray coded