
EE273 Lecture 13

Synchronization

November 9, 1998

William J. Dally
Computer Systems Laboratory
Stanford University
billd@csl.stanford.edu

Today's Assignment

- Reading
 - Sections 10.3
 - Complete before class on Wednesday 11/11

A Quick Overview

- Synchronization
 - determining an event order
 - used for
 - moving a signal into a clock domain
 - asynchronous arbitration
- Synchronization Failure
 - as the time between two signals decreases it becomes more difficult to tell which came first
 - synchronizer may hang in a metastable state, unable to decide
- Failure Probability
 - is proportional to fraction of *vulnerable* time
 - exponentially decreases with waiting period
 - exponentially increases with flip-flop time constant
 - failure rate is proportional to event rate

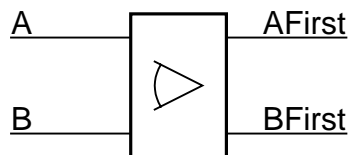
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

3

What is Synchronization?

- A *synchronizer* determines the order of events on two signals
- Which event came first?
 - Does it matter? Some times synchronization is unnecessary
- Often one signal is a clock
 - did the data go high before or after the clock went high?
- Why is this problem hard?



EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

4

Uses of Synchronization

- Sampling asynchronous inputs with a clock
 - e.g., particle counter or pushbutton
- Crossing clock domains
 - sampling a synchronous signal with a *different* clock
 - this is an easier problem if both clocks are periodic
- Arbitration of asynchronous signals
 - e.g., request line for shared resource
 - game-show pushbutton

EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

5

Synchronization Failure

- Which came first, event on A or event on B?
- The closer the race, the harder it is to call
- When the events are very close, the synchronizer may enter a *metastable* state
- The synchronizer may take an arbitrary amount of time to *exit* this state
- Synchronizer output may be interpreted inconsistently in the meantime

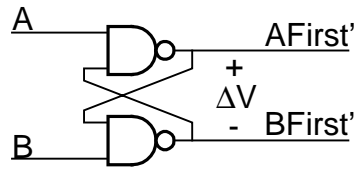
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

6

A NAND Arbiter

- Consider a NAND RS flip-flop
- We can attempt to use this as an arbiter.
 - If A arrives much earlier than B, AFirst' goes low and locks B out
 - If B arrives much earlier than A, BFirst' goes low and locks A out.
- What happens if A and B go high at nearly the same time?



EE273, L13, Nov 9, 1998

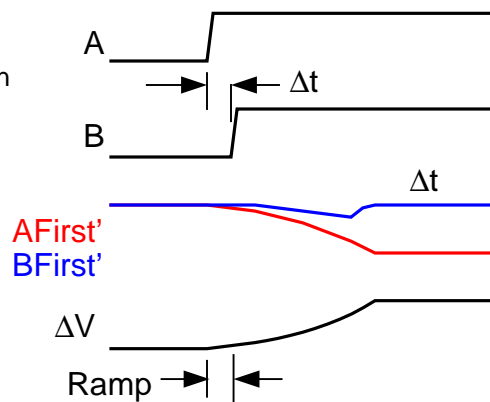
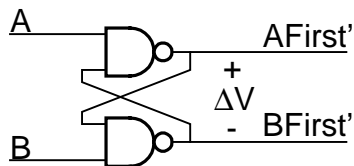
Copyright (C) by William J. Dally, All Rights Reserved

7

A NAND Arbiter Dynamics Step 1: Initial Voltage

- When A and B go high at nearly the same time, difference in voltage is proportional to difference in time

$$\Delta V_1 = K_S \Delta t$$



EE273, L13, Nov 9, 1998

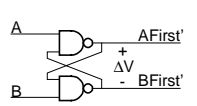
Copyright (C) by William J. Dally, All Rights Reserved

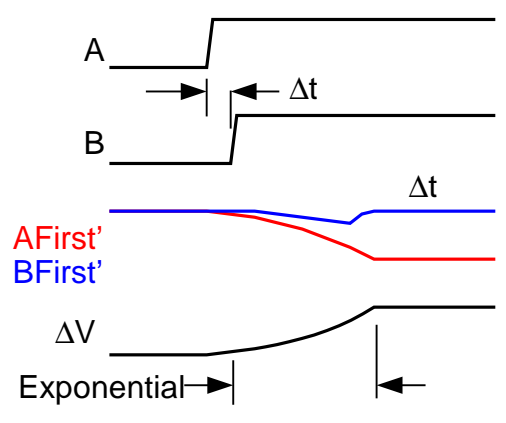
8

A NAND Arbiter Dynamics Step 2: Exponential Regeneration

- After A and B are both high, initial voltage difference is exponentially amplified.

$$\Delta V_1 = K_S \Delta t$$

$$\Delta V(t) = \Delta V_1 \exp\left(\frac{t}{\tau_S}\right)$$




EE273, L13, Nov 9, 1998 Copyright (C) by William J. Dally, All Rights Reserved 9

A NAND Arbiter Dynamics Settling Time

- After A and B are both high, initial voltage difference is exponentially amplified.

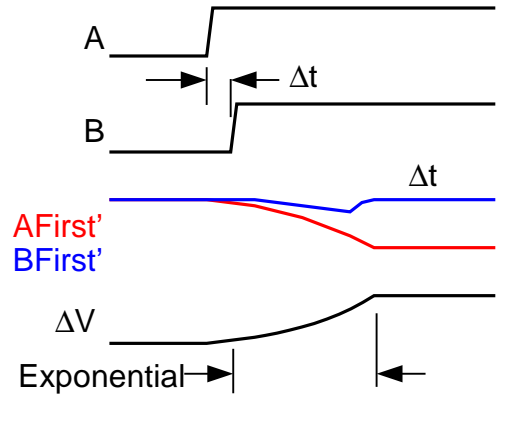
$$\Delta V_1 = K_S \Delta t$$

$$\Delta V(t) = \Delta V_1 \exp\left(\frac{t}{\tau_S}\right)$$

$$\Delta V_1 = \exp\left(-\frac{t}{\tau_S}\right)$$

$$t_x = -\tau_S \log(\Delta V_1)$$

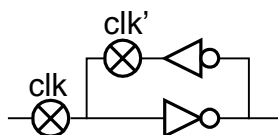
$$= -\tau_S \log(K_S \Delta t)$$



EE273, L13, Nov 9, 1998 Copyright (C) by William J. Dally, All Rights Reserved 10

Static Flip-Flop Dynamics are Similar

- Initial voltage difference depends on Δt
- Voltage difference increases exponentially after clock rises



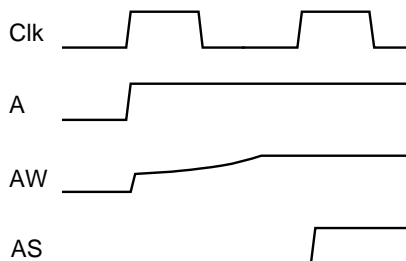
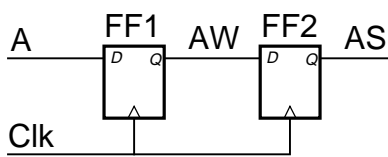
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

11

A Brute-Force (Waiting) Synchronizer

- To sample an asynchronous signal with a clock
- Sample signal with FF1
 - may go into a metastable state
- Wait for possible metastable stages to decay
 - time t_w
- Sample output of FF1



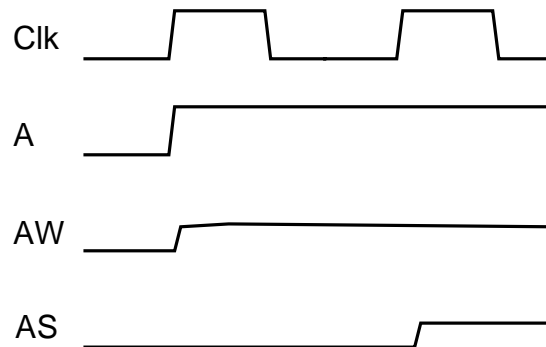
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

12

Synchronization Failure

- What happens if FF1 is still in a metastable state when FF2 is clocked?
- What is the probability that this will happen?



EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

13

Calculating Synchronization Failure (The Big Picture)

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

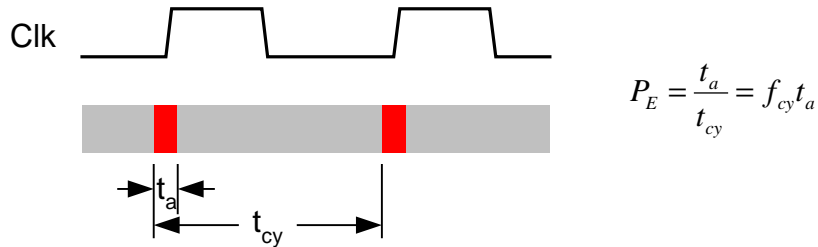
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

14

Probability of Entering a Metastable State

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



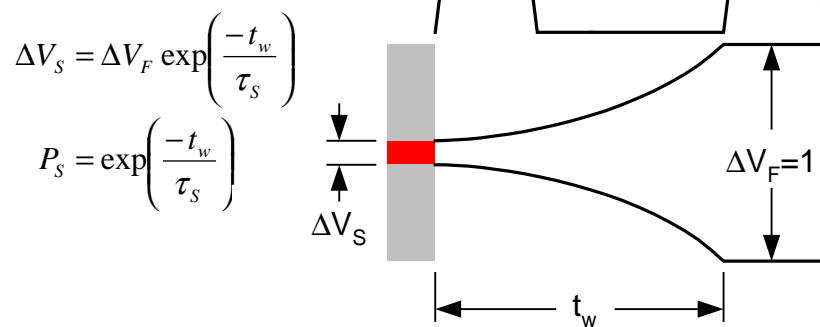
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

15

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'



EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

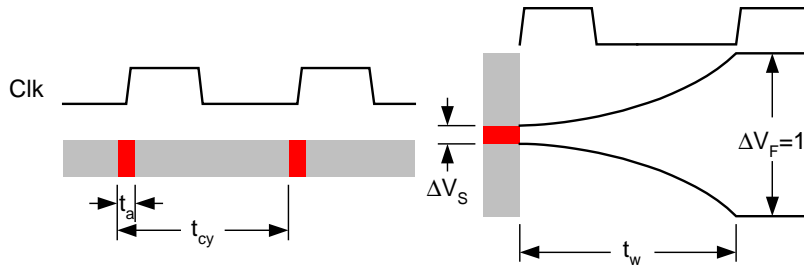
16

Failure Probability and Error Rate

- Each event can potentially fail.
- Failure rate is given by the event rate times the failure probability

$$P_F = P_E P_S = t_a f_{cy} \exp\left(\frac{-t_w}{\tau_S}\right)$$

$$f_F = f_e P_F = t_a f_e f_{cy} \exp\left(\frac{-t_w}{\tau_S}\right)$$



EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

17

Example Failure Rate Calculation

- Suppose a 500MHz clock samples a 10MHz asynchronous signal
- Flip-flops have aperture and regeneration time of 100ps
- What is the probability of synchronization failure?
- What is the frequency?

t_a	1.0E-10
f_cy	5.0E+08
τ_s	1.0E-10
t_w	2.0E-09
P_E	5.0E-02
P_S	2.1E-09
P_F	1.0E-10
f_e	1.0E+07
f_F	1.0E-03

EE273, L13, Nov 9, 1998

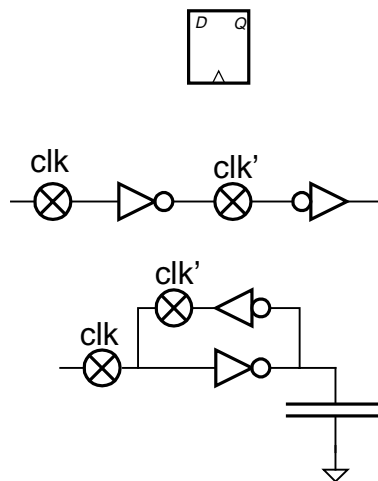
Copyright (C) by William J. Dally, All Rights Reserved

18

Common Pitfalls

- Its easy to get a synchronizer design wrong
- The two most common pitfalls are:
 - using a non-restoring (or slowly restoring) flip-flop
 - not isolating the flip-flop feedback loop

$$P_F = t_a f_{cy} \exp\left(\frac{-t_w}{\tau_S}\right)$$



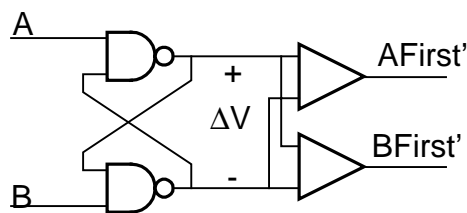
EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

19

Completion Detection

- It is not possible to bound the amount of time needed for a synchronizer to settle.
- It is, however, possible to detect when the synchronizer has settled!
- This is only useful if the downstream logic can use this *asynchronous* completion signal



EE273, L13, Nov 9, 1998

Copyright (C) by William J. Dally, All Rights Reserved

20

Next Time

- Synchronizer Design