

# Lecture 13 :

## Synchronization in Synchronous / Asynchronous Systems

EE3801 — Advanced Logic  
Circuits

1

## Session Overview

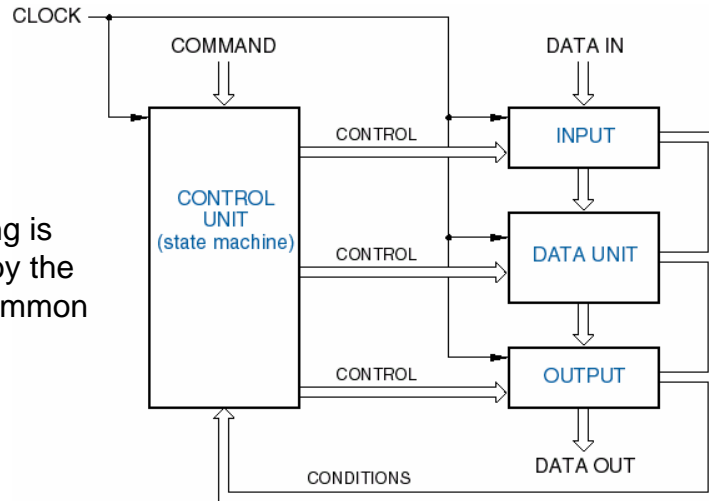
- Synchronous systems
  - Clock Skew
- Asynchronous Systems
  - Input Synchronization
  - Metastability Revisited
  - Synchronizers

EE3801 — Advanced Logic  
Circuits

2

## Synchronous System Structure

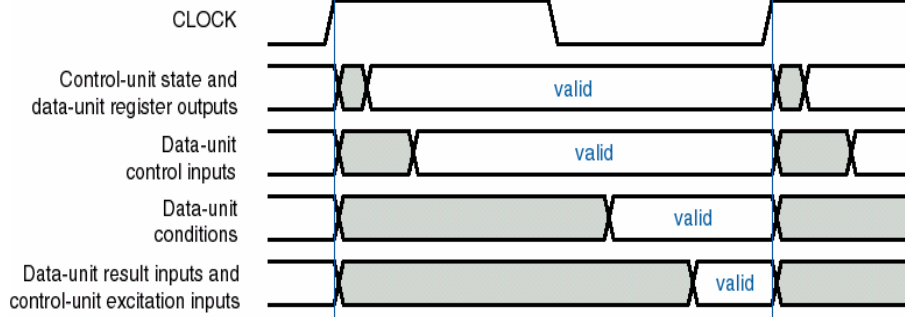
Everything is clocked by the same, common clock



EE3801 — Advanced Logic Circuits

3

## Typical Synchronous-System Timing



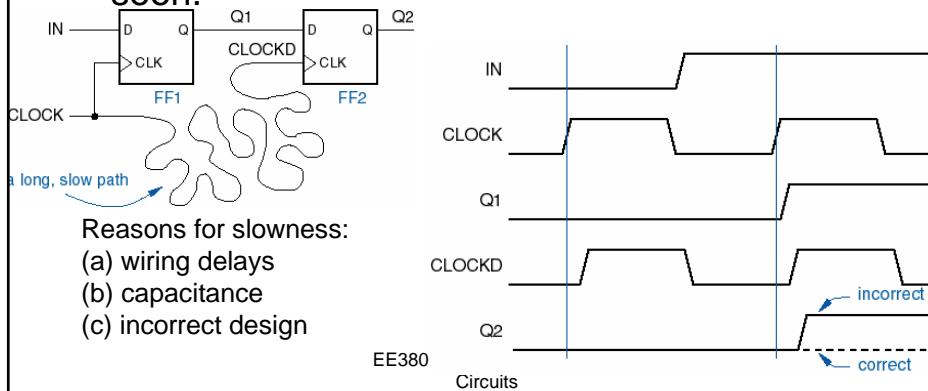
- Outputs have one complete clock period to propagate to inputs.
- Must take into account flip-flop setup times at next clock period.

EE3801 — Advanced Logic Circuits

4

## Clock Skew

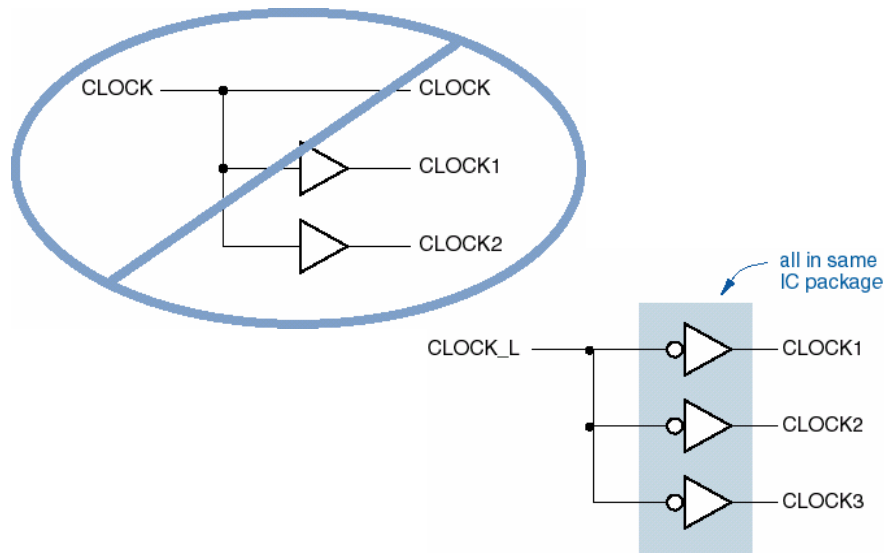
- Clock signal may not reach all flip-flops simultaneously.
- Output changes of flip-flops receiving “early” clock may reach D inputs of flip-flops with “late” clock too soon.



## Clock-Skew Calculation

- $t_{ffpd(\min)} + t_{comb(\min)} - t_{hold} - t_{skew(\max)} > 0$
- First two terms are minimum time after clock edge that a D input changes
- Hold time is earliest time that the input may change
- Clock skew subtracts from the available hold-time margin
- Compensating for clock skew:
  - Longer flip-flop propagation delay
  - Explicit combinational delays
  - Shorter (even negative) flip-flop hold times

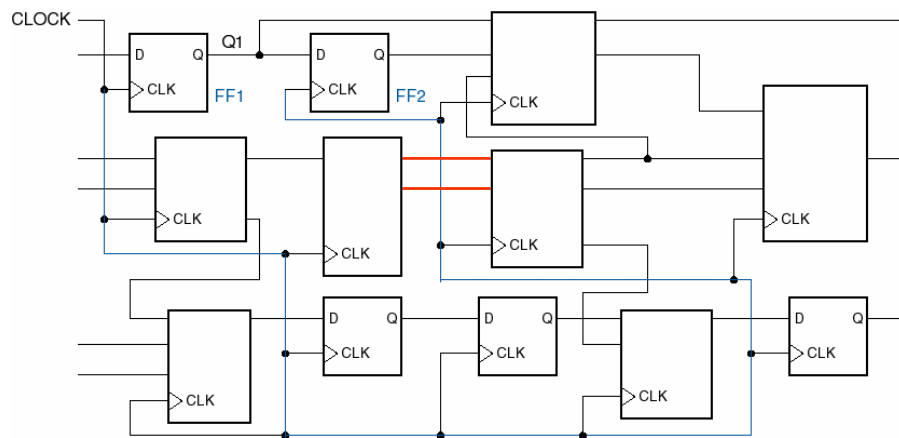
## Example of Bad Clock Distribution



EE3801 — Advanced Logic Circuits

7

## Clock Distribution in ASICs

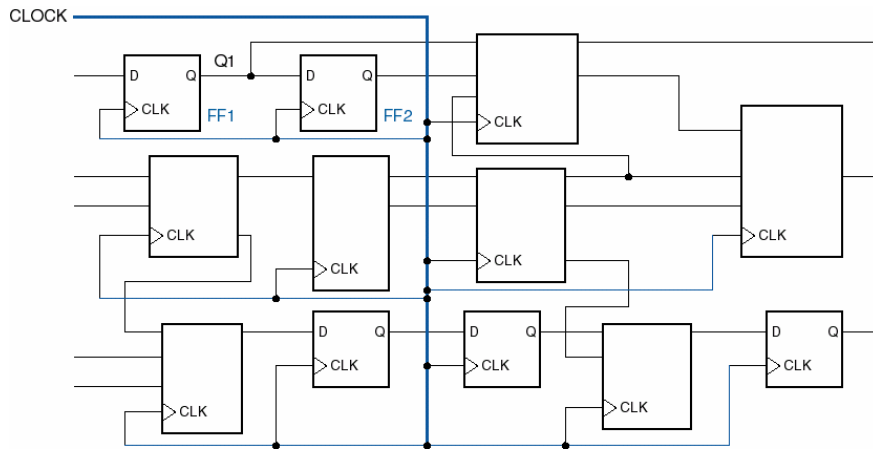


- This is what a typical ASIC router will do if you don't lay out the clock by hand.

EE3801 — Advanced Logic Circuits

8

## “Clock-Tree” solution

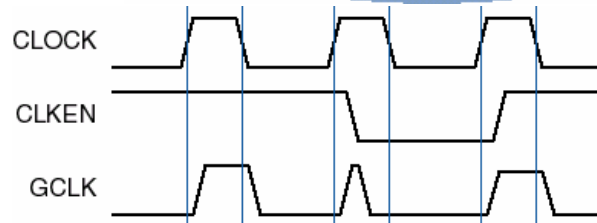
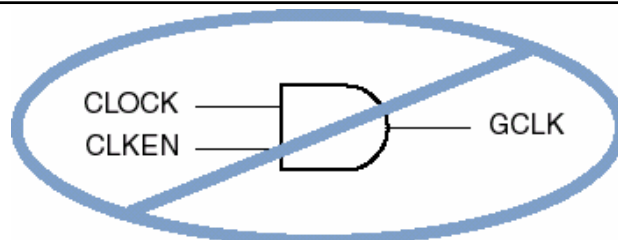


- Often laid out by hand
- Wide, fast metal (low  $R \implies$  fast RC time constant)

EE3801 — Advanced Logic Circuits

9

## Gating the clock

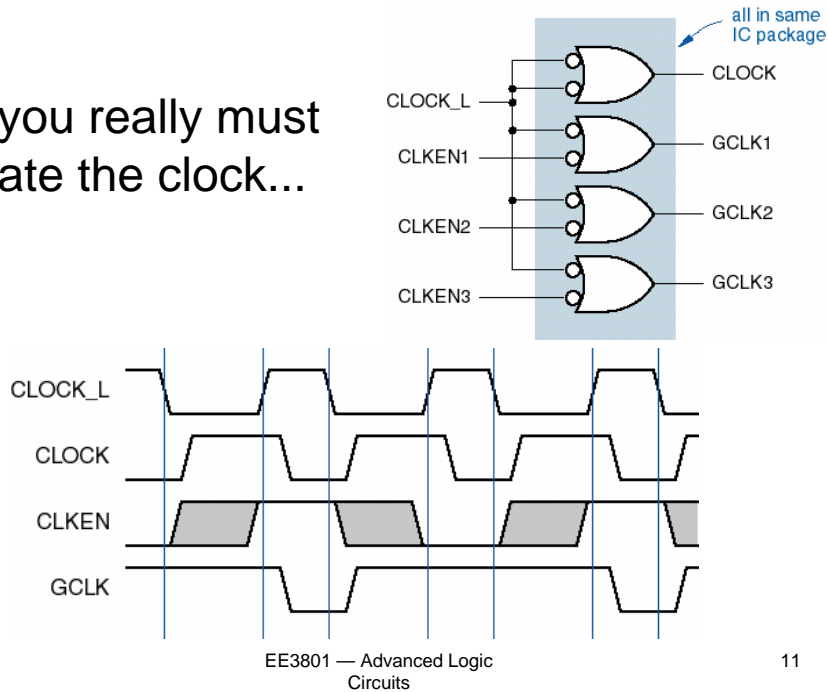


- Definitely a no-no
  - Glitches possible if control signal (CLKEN) is generated by the same clock
  - Excessive clock skew in any case.

EE3801 — Advanced Logic Circuits

10

If you really must gate the clock...

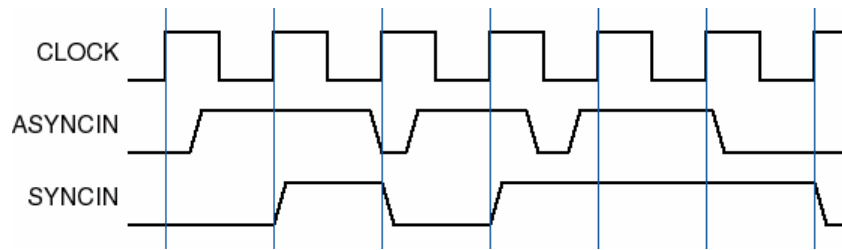
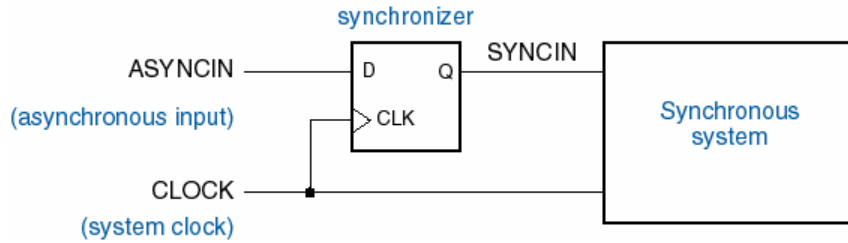


11

## Asynchronous inputs

- Not all inputs are synchronized with the clock
- Examples:
  - Keystrokes
  - Sensor inputs
  - Data received from a network (transmitter has its own clock)
- Inputs must be synchronized with the system clock before being applied to a synchronous system.

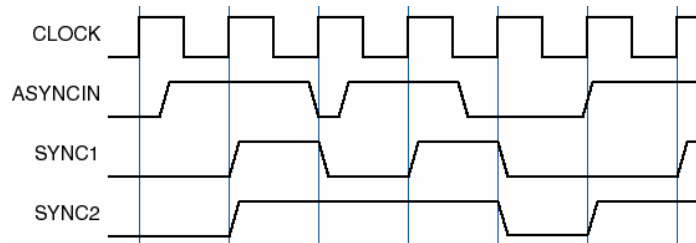
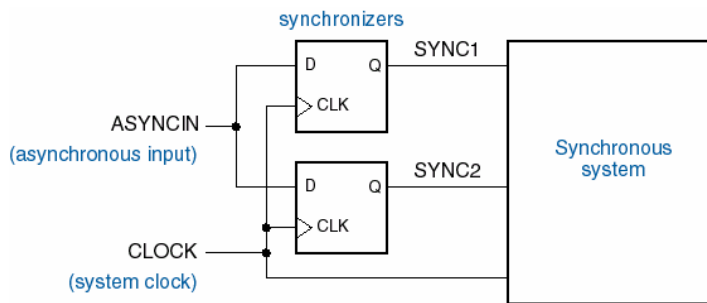
# A Simple Synchronizer



EE3801 — Advanced Logic Circuits

13

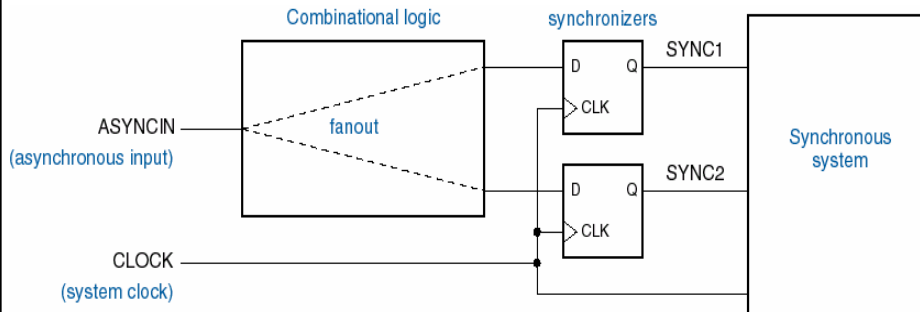
# Only **one** Synchronizer per Input



EE3801 — Advanced Logic Circuits

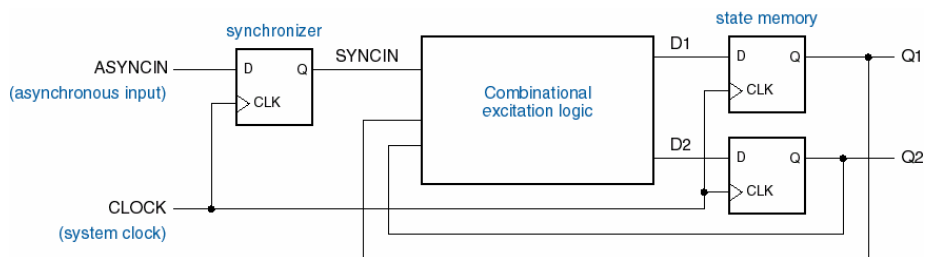
14

## Even Worse



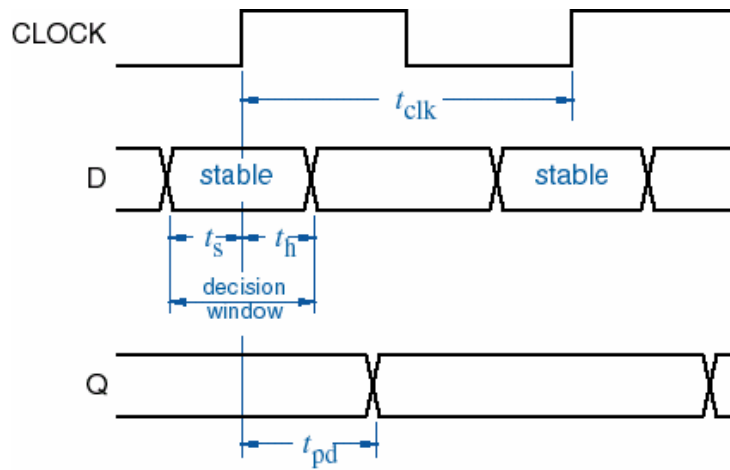
- Combinational delays to the two synchronizers are likely to be different.

## The Way to Do it



- One synchronizer per input
- Carefully locate the synchronization points in a system.
- But still a problem -- the synchronizer output may become metastable when setup and hold time are not met.

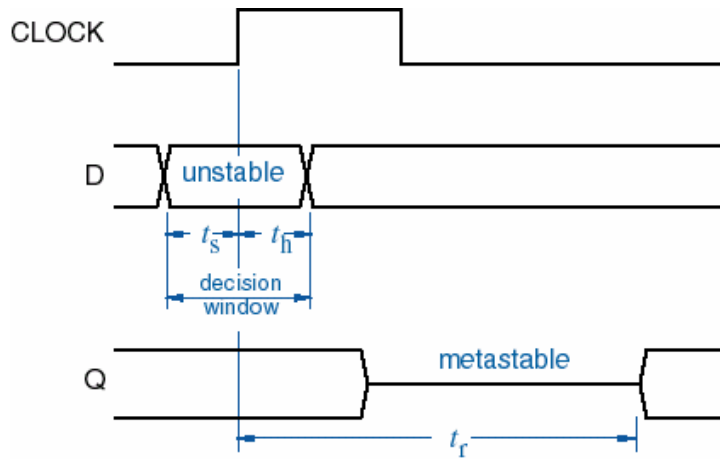
## Metastability Decision Window



EE3801 — Advanced Logic Circuits

17

## Metastability Resolution Time

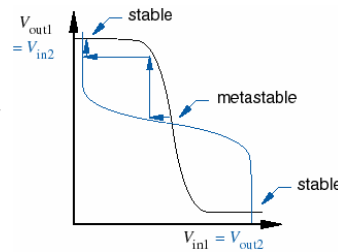


EE3801 — Advanced Logic Circuits

18

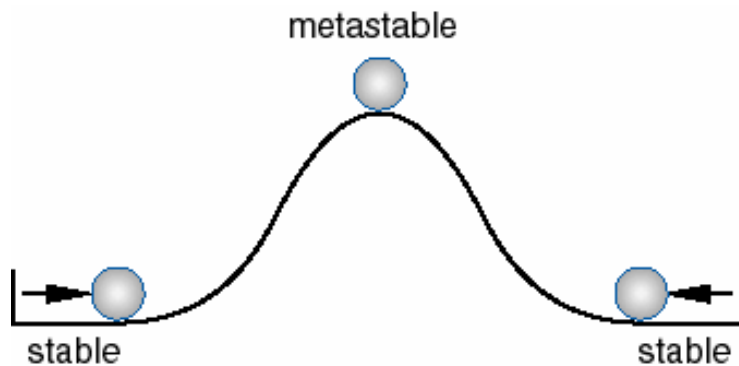
## Metastability - Revisited

- Two stable states
  - $V_{o1}=L, V_{o2}=H$
  - $V_{o1}=H, V_{o2}=L$
- One metastable state
  - $V_{o1} = V_{o2}$
- Ugly characteristic: unbounded recovery time  $t_r$



Transfer function:  
 $V_{out1} = T(V_{in1})$   
 $V_{out2} = T(V_{in2})$

## Metastability



## Metastability

- Bad news
  - Metastability is unavoidable
  - Recovery time is theoretically unbounded
- Good news
  - Can empirically measure recovery times
  - Can use statistics from recovery times to make failure probability arbitrarily small

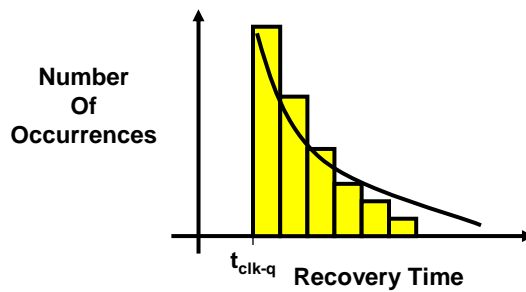
## Flip-flop Metastable Behavior

- Probability of flip-flop output being in the metastable state is an exponentially decreasing function of  $t_r$  (time since clock edge, a.k.a. “**Resolution Time**”).
- Stated another way,

$$\text{MTBF}(t_r) = \exp(t_r / \tau) / [T_0 f a]$$

Where  $\tau$  and  $T_0$  are parameters for a particular flip-flop  
 $f$  is the clock frequency, and  
 $a$  is the number of asynchronous transitions / sec

$$\text{MTBF}(t_r) = \exp(t_r / \tau) / [T_0 f a]$$



## Resolution Time Example

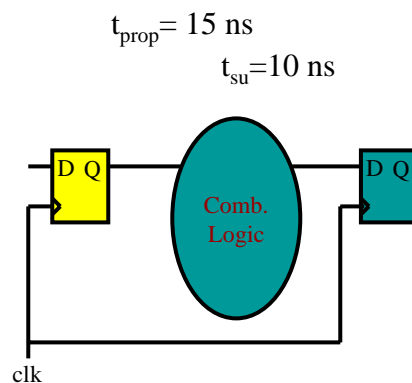
- Example:  $f = 5 \text{ MHz}$  ( $t_{\text{clk}} = 200 \text{ ns}$ )

- $a = 5 \text{ KHz}$

- $t_{\text{prop}} = 15 \text{ ns}$

- $t_{\text{su}} = 10 \text{ ns}$

- $t_r = t_{\text{clk}} - t_{\text{su}} - t_{\text{prop}}$   
 $= 200 \text{ ns} - 10 \text{ ns} - 15 \text{ ns}$   
 $= 175 \text{ ns}$

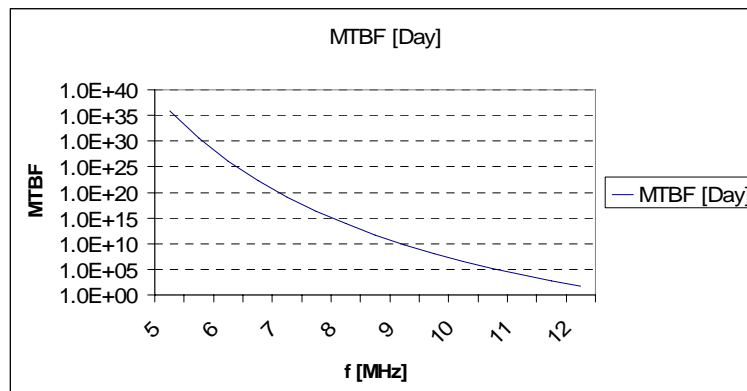


## MTBF Calculation

- “Typical” values for a flip-flop
  - $\tau = 1.5 \text{ ns}$
  - $T_o = 0.4 \text{ s}$
  - $t_r = 175 \text{ ns}$
- MTBF =  $1.47 \times 10^{33}$  Years

$$\text{MTBF}(t_r) = \frac{e^{(t_r/\tau)}}{T_o \times f_{\text{clk}} \times a}$$

## MTBF Vs. Clock Frequency



## Typical flip-flop metastability parameters

$$\text{MTBF} = \frac{\exp(t_r / \tau)}{[T_0 f a]}$$

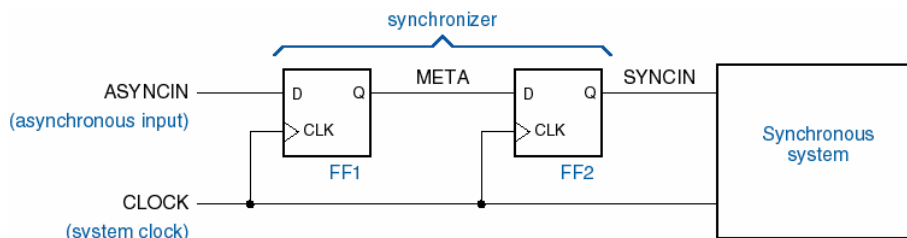
MTBF = 1000 yrs.  
 F = 25 MHz  
 a = 100 KHz  
 $t_r = ?$

Device	$\tau$ (ns)	$T_0$ (s)	$t_r$ (ns)
74LS74	1.50	$4.0 \cdot 10^{-1}$	77.71
74S74	1.70	$1.0 \cdot 10^{-6}$	66.14
74S174	1.20	$5.0 \cdot 10^{-6}$	48.62
74S374	0.91	$4.0 \cdot 10^{-4}$	40.86
74F74	0.40	$2.0 \cdot 10^{-4}$	17.68
74LSxx	1.35	$4.8 \cdot 10^{-3}$	63.97
74Sxx	2.80	$1.3 \cdot 10^{-9}$	90.33
74ALSxx	1.00	$8.7 \cdot 10^{-6}$	41.07
74ASxx	0.25	$1.4 \cdot 10^3$	14.99
74Fxx	0.11	$1.9 \cdot 10^8$	7.90
74HCxx	1.82	$1.5 \cdot 10^{-6}$	71.55
PALC16R8-25	0.52	$9.5 \cdot 10^{-12}$	14.22*
PALC22V10B-20	0.26	$5.6 \cdot 10^{-11}$	7.57*
PALCE22V10-7	0.19	$1.3 \cdot 10^{-13}$	4.38*
7300-series CPLD	0.29	$1.0 \cdot 10^{-15}$	5.27*
9500-series CPLD	0.17	$9.6 \cdot 10^{-18}$	2.30*

EE3801 — Advanced Logic Circuits

27

## Recommended Synchronizer Design



- Hope that FF1 settles down before “META” is sampled
  - In this case, “SYNCIN” is valid for almost a full clock period.
  - Can calculate the probability of “synchronizer failure” (FF1 still metastable when META sampled)

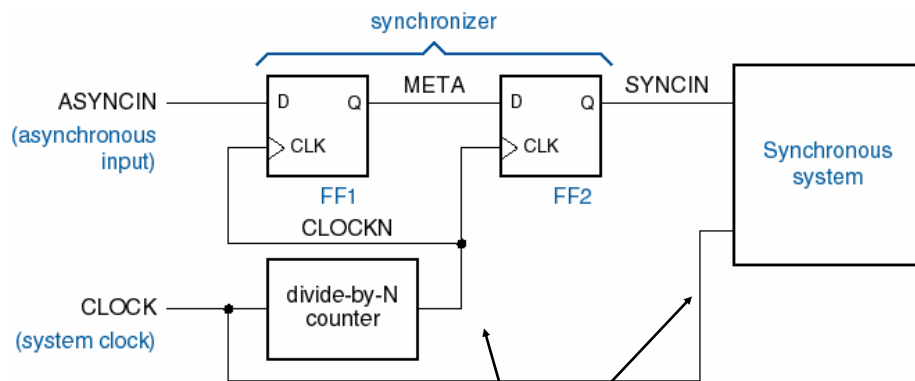
EE3801 — Advanced Logic Circuits

28

## Is 1000 Years Enough?

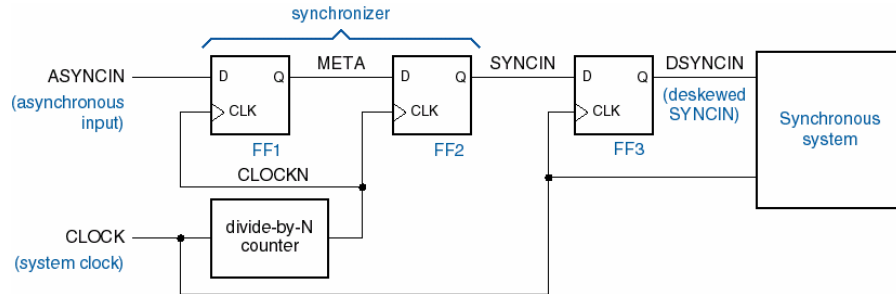
- If MTBF = 1000 years and you ship 52,000 copies of the product, then some system experiences a mysterious failure every week.
- Real-world MTBFs must be much higher.
- How to get better MTBFs?
  - Use faster flip-flops
    - But clock speeds keep getting faster, thwarting this approach.
  - Wait for multiple clock ticks to get a longer metastability resolution time
    - Waiting longer usually doesn't hurt performance
    - ...unless there is a critical "round-trip" handshake.

## Multiple-cycle synchronizer



- Clock-skew problem

## Deskewed Multiple-Cycle Synchronizer



- Necessary in really high-speed systems
- DSYNCIN is valid for almost an entire clock period.

## Summary

- Synchronous systems
  - Clock Skew
- Asynchronous Systems
  - Input Synchronization
  - Metastability Revisited
  - Synchronizers