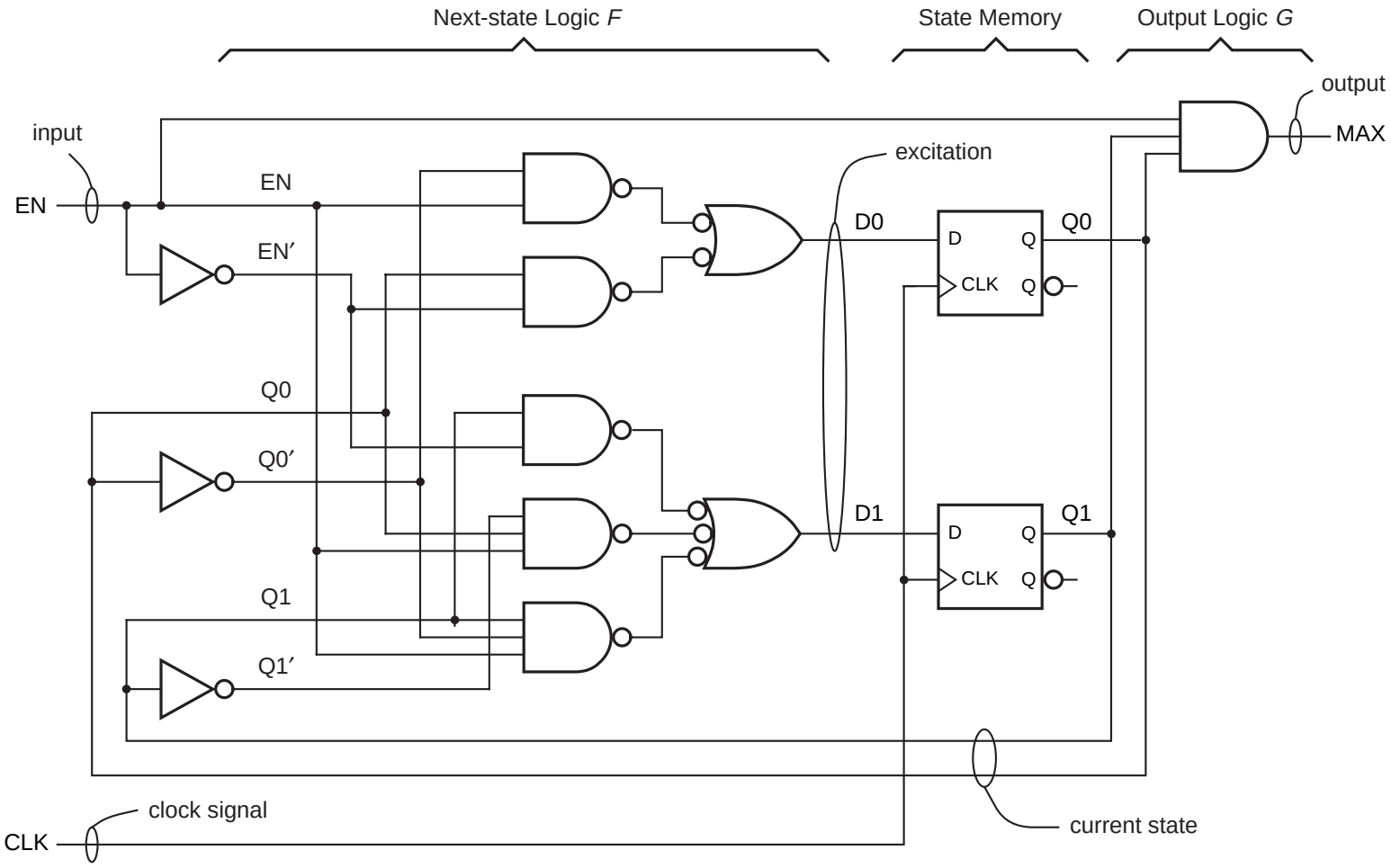
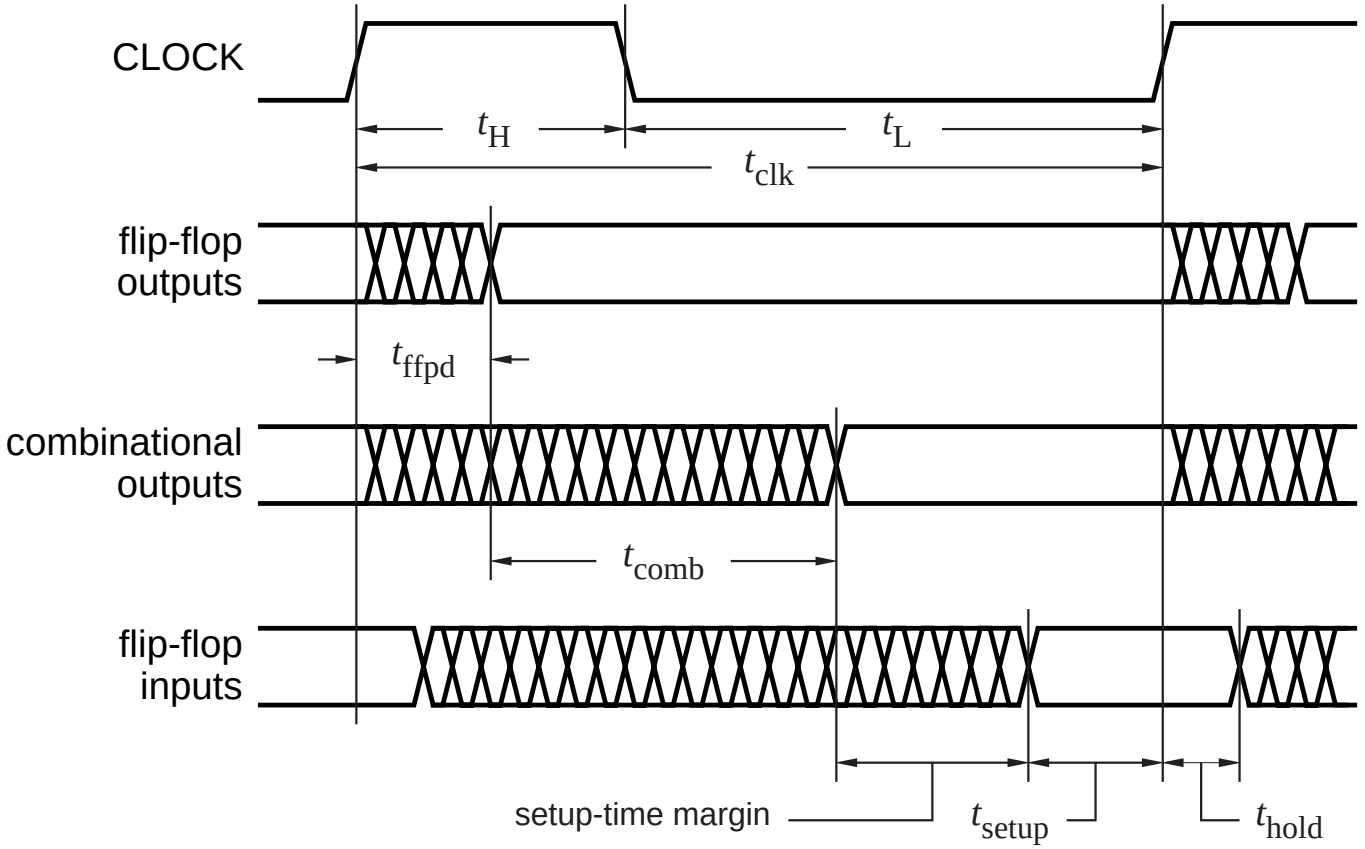
Clocked synchronous state machine example



Setup and hold times requirements for state flip-flops must be satisfied.

Clocked synchronous state machine timing



Timing margin equation: $t_{\text{clk}} \geq t_{\text{ffpd}} + t_{\text{comb}} + t_{\text{setup}}$

Setup time margin $= t_{\text{clk}} - t_{\text{ffpd}} - t_{\text{comb}} - t_{\text{setup}}$

Satisfying timing requirements

The setup time margin can be made positive by making t_{clk} large enough.

Simple solution: slow down the system.

Hold time requirement is independent of system clock.

- Guarantee that minimum combination logic delay is larger than hold time:

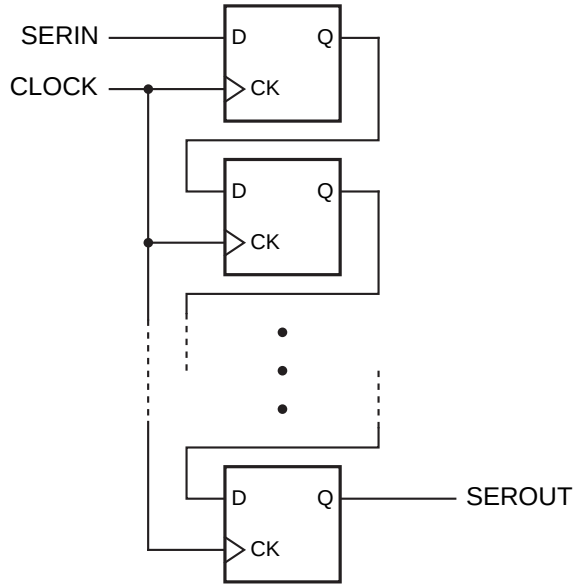
$$t_{\text{ffpd}} + t_{\text{comb}} \geq t_{\text{hold}}$$

  Note that manufacturer's *minimum* delay specifications are needed.

- Use "good" flip-flops (hold time ≤ 0)
- Kludge: add delay to guarantee proper operation.
- Important: avoid clock skew.

Hold times

To guarantee that shift registers work, hold time t_{hold} must be smaller than flip-flop propagation delay t_{ffpd}.

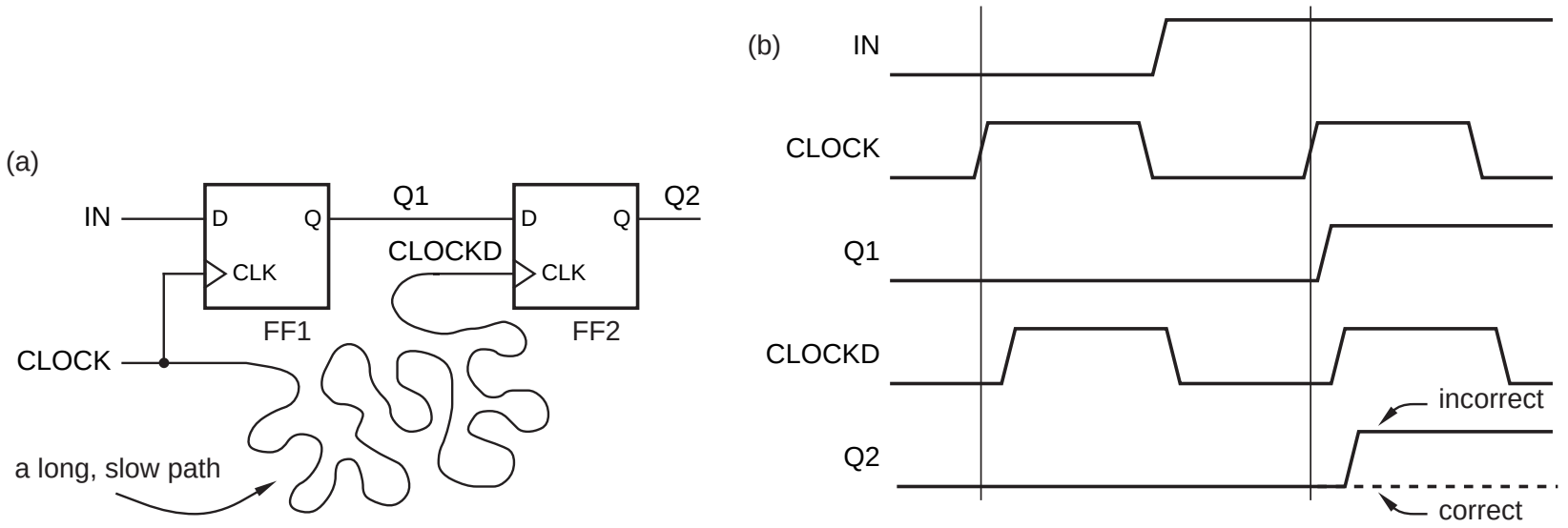


From DDPP Table 8-1, 74FCT273 has min $t_{\text{ffpd}} = 2$ and $t_{\text{hold}} = 1.5$.

Early ('70s) databooks listed "typical" but not minimum propagation delays.

Clock skew example

All flip-flops in clocked synchronous state machine should be clocked at the "same" time. Violating this rule may result in hold time violations.

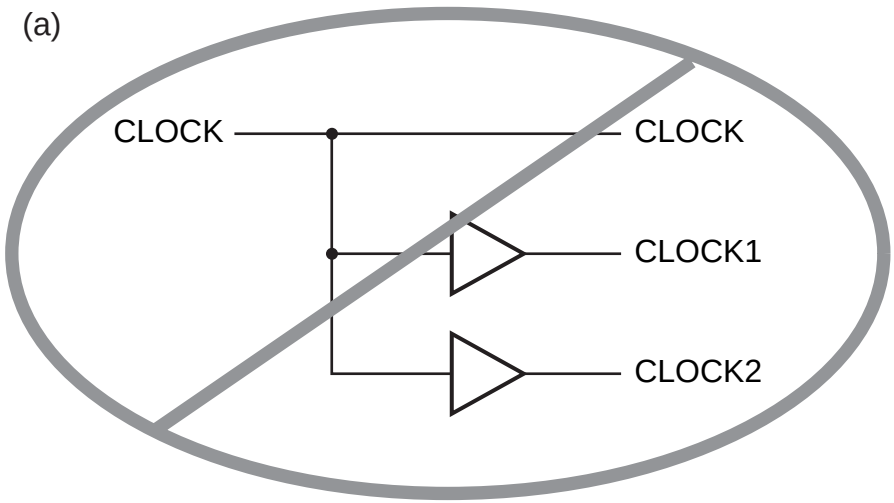


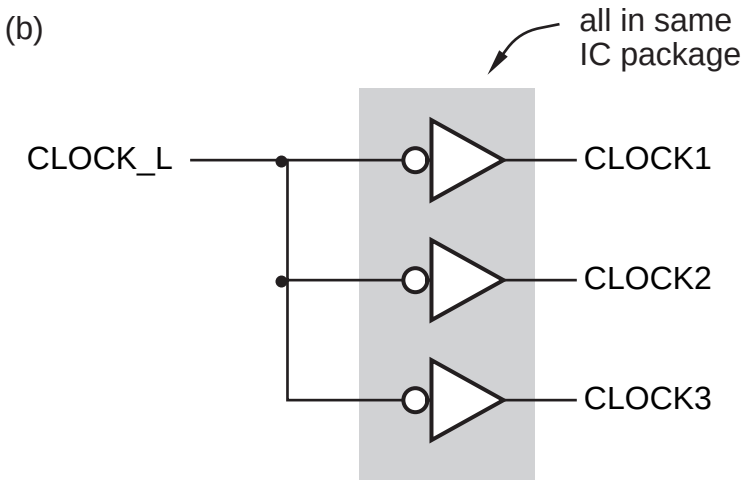
"Same" time means that difference between active edges should be small compared to hold time.

Clock rise and fall times should be short, in case flip-flops respond to different voltage levels. (Use similar flip-flops when possible.)

Clock distribution

In real systems (such as Xilinx XCS200) clock distribution is an important consideration. Goal: simultaneous clocking of all flip-flops.

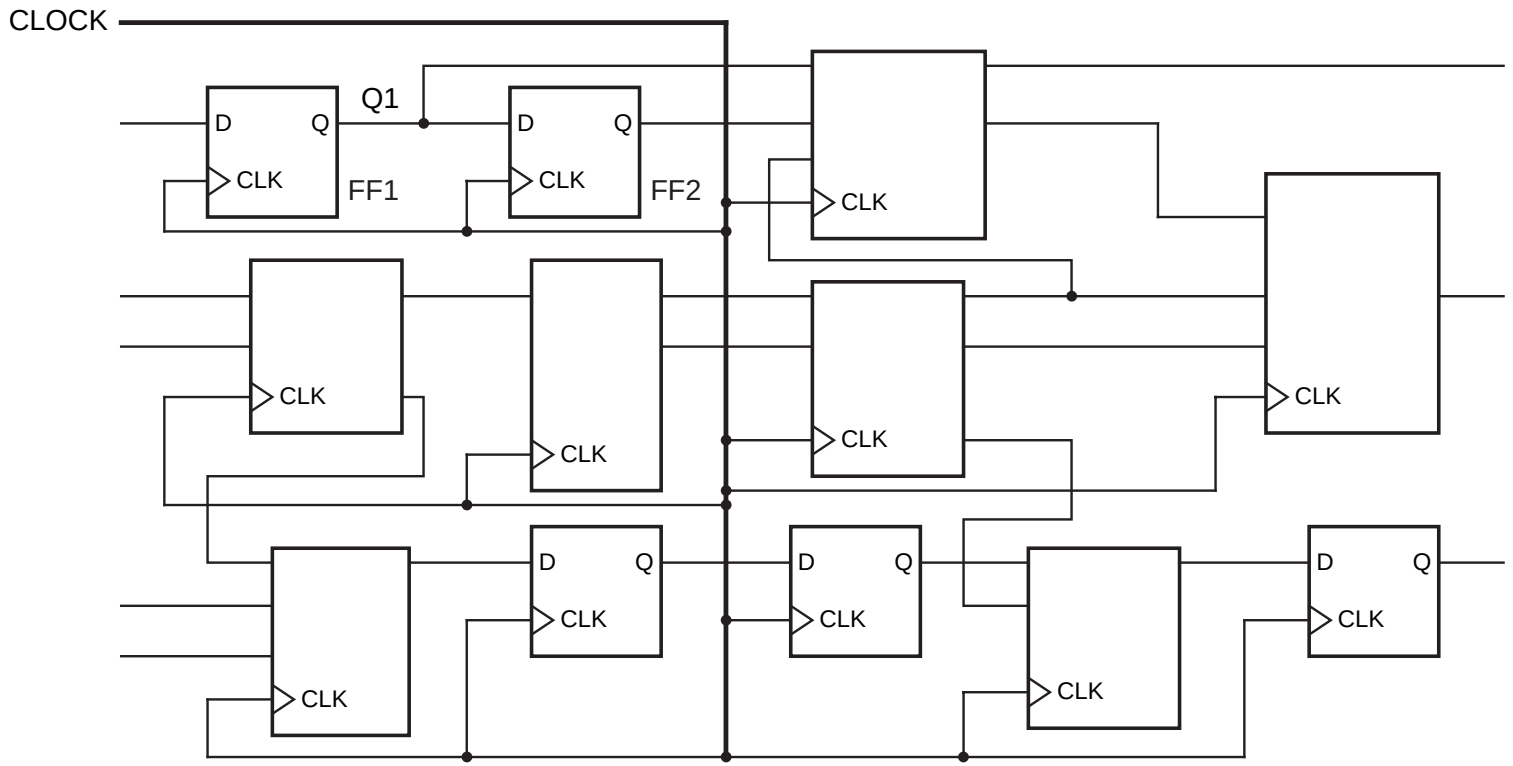




Master clock must be buffered through a tree so that slave clocks have same delay. (Done for you in Xilinx FPGAs.)

Clock distribution

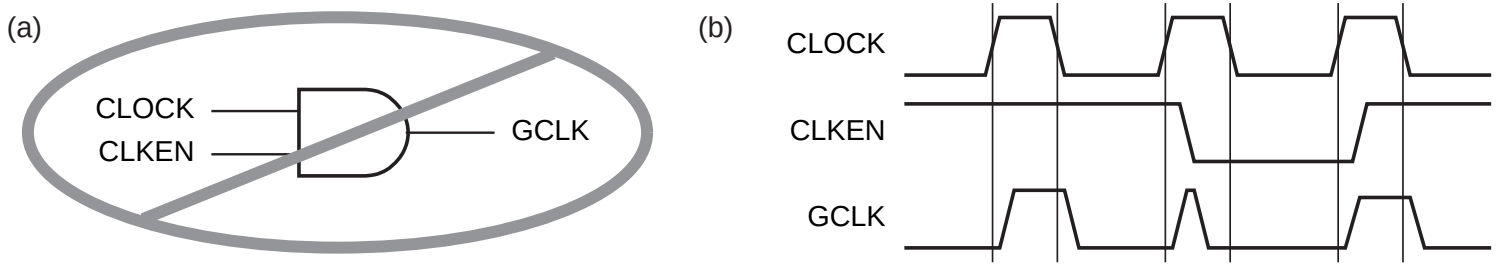
On a printed circuit board, clock traces are laid out to minimize discrepancies.



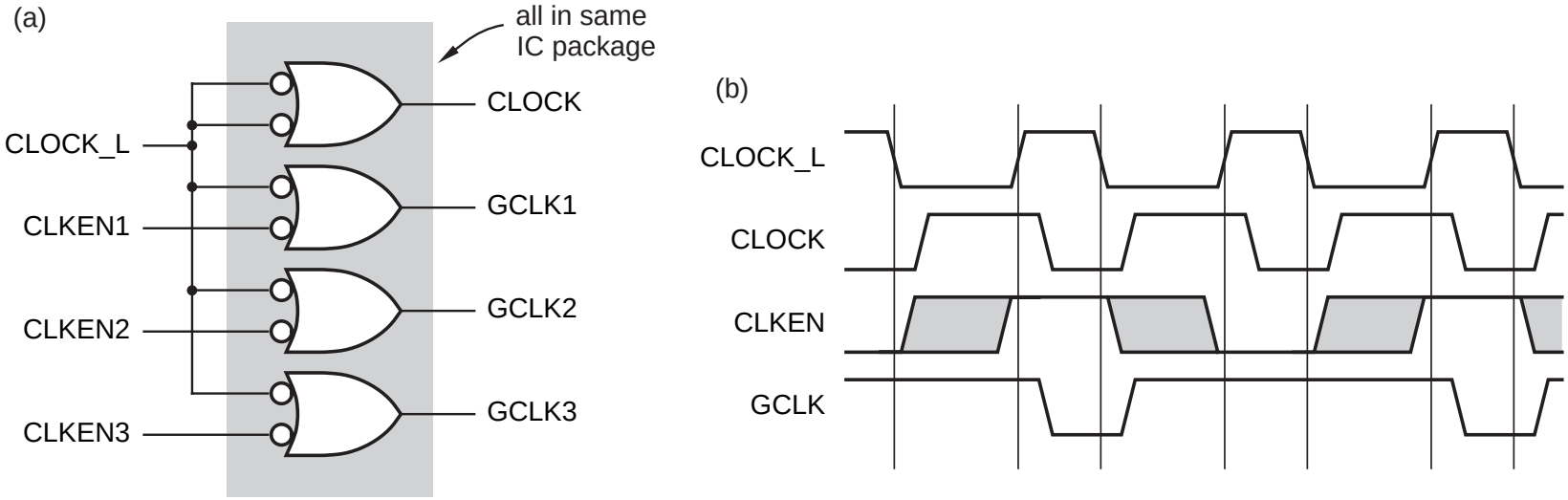
Cultural note: high performance ECL boards were laid out so that all signals traveled over traces with same length.

Don't gate the clock!

How not to gate the clock:

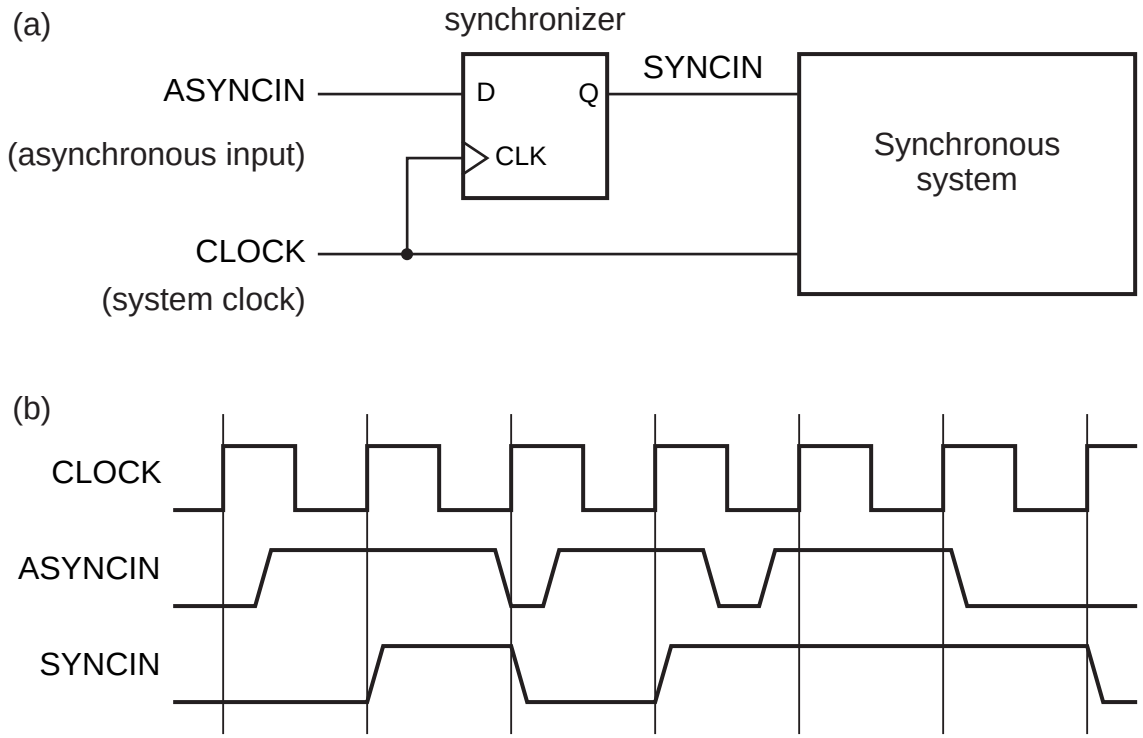


An "acceptable" way to gate the clock (reduces clock skew). Not in EE 121.



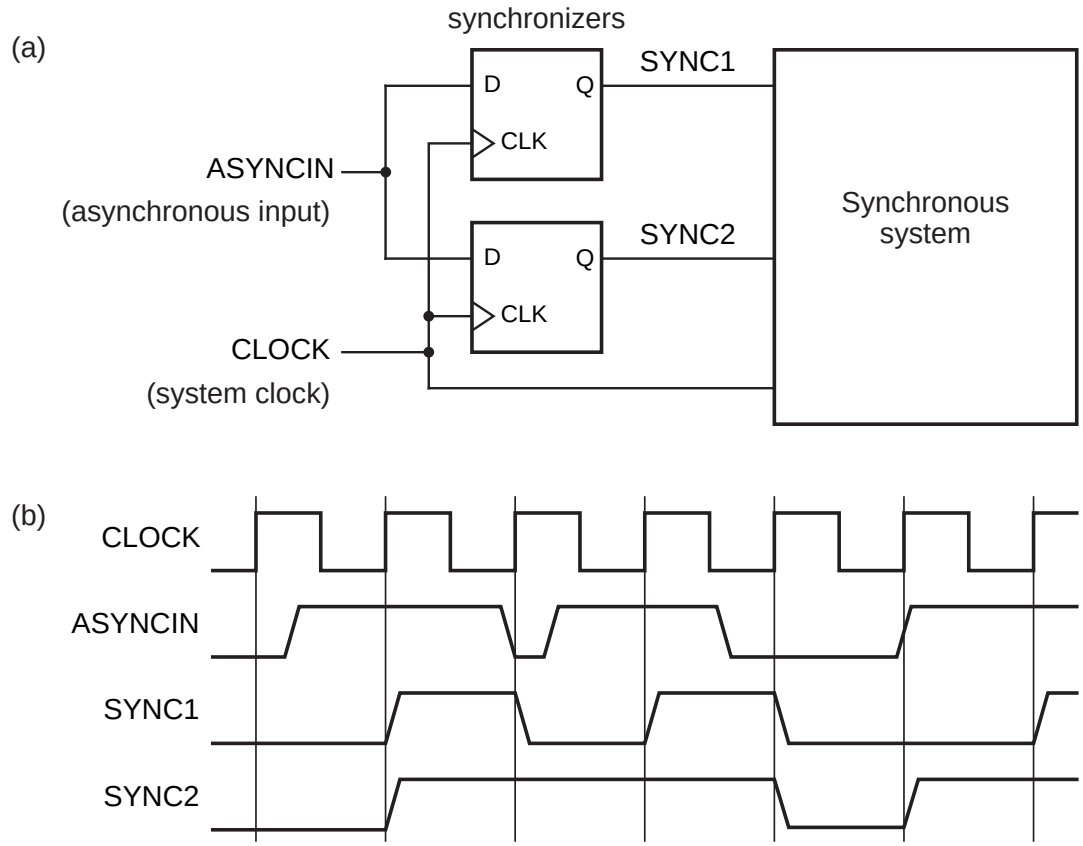
Asynchronous inputs

Very simple synchronizer circuit:



This works most of the time (failure rate proportional to system clock).

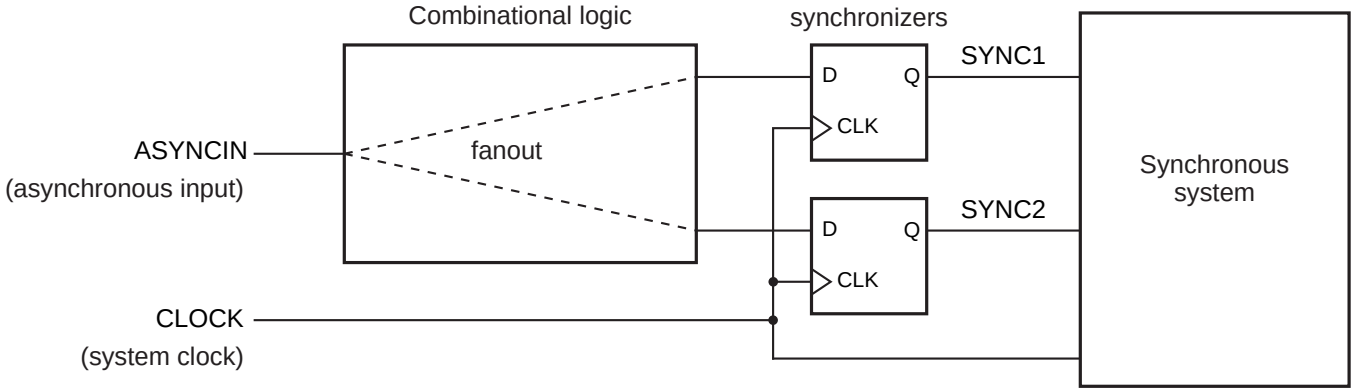
Asynchronous inputs: multiple synchronizers



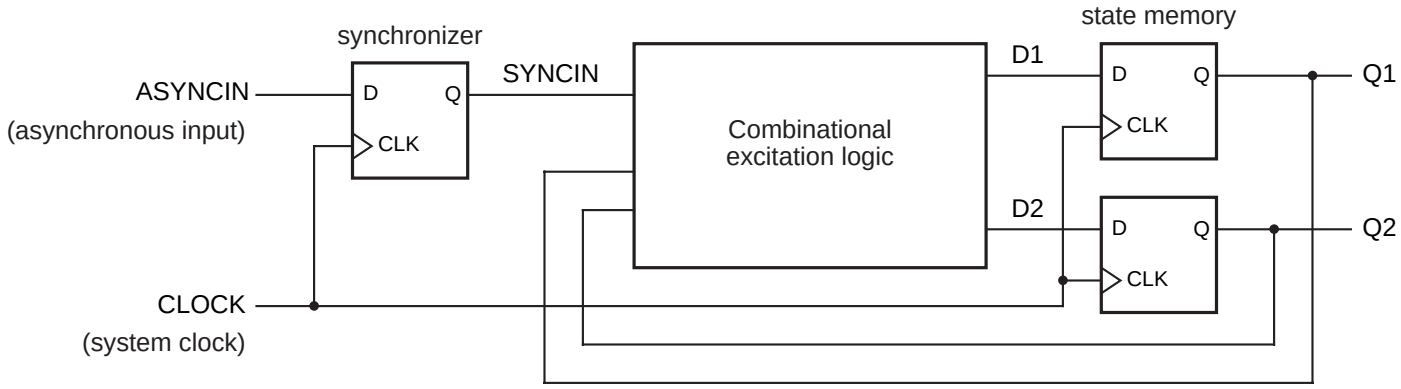
Sample an asynchronous signal at one place in your circuit. Otherwise, system might see inconsistent values of input.

Asynchronous inputs: subtle problem

We might indirectly get two views of a variable.

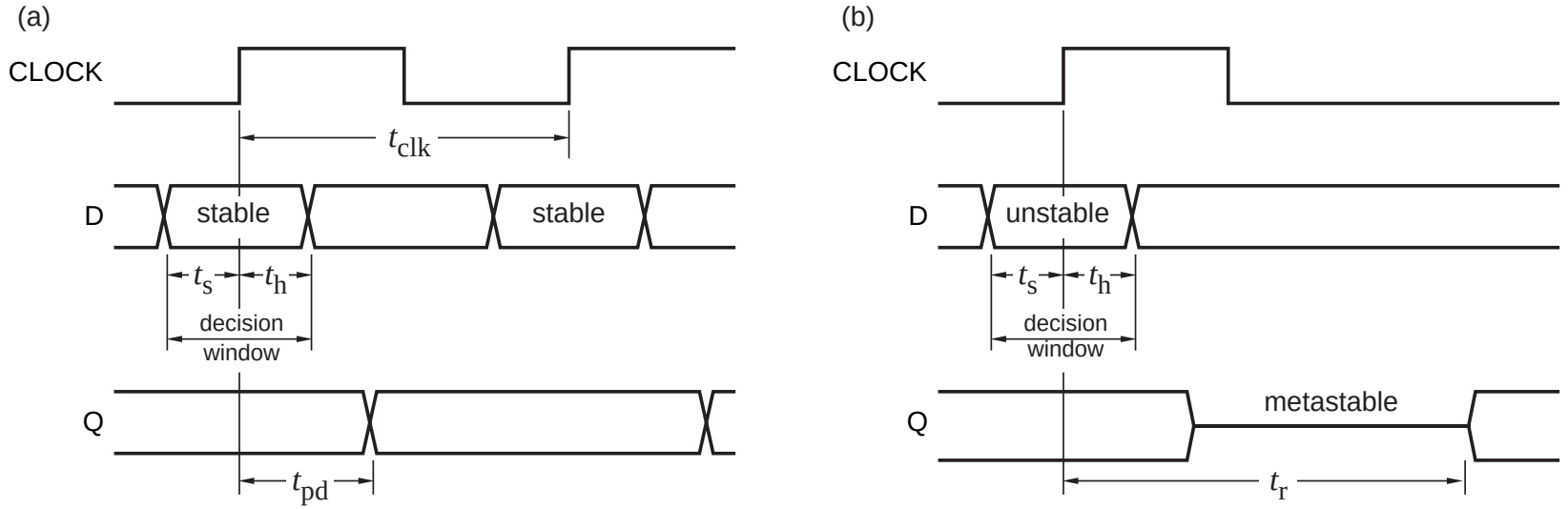


Solution: synchronize the variable for supplying it to combinational logic.



Metastability

What can go wrong when we sample data during the decision window:

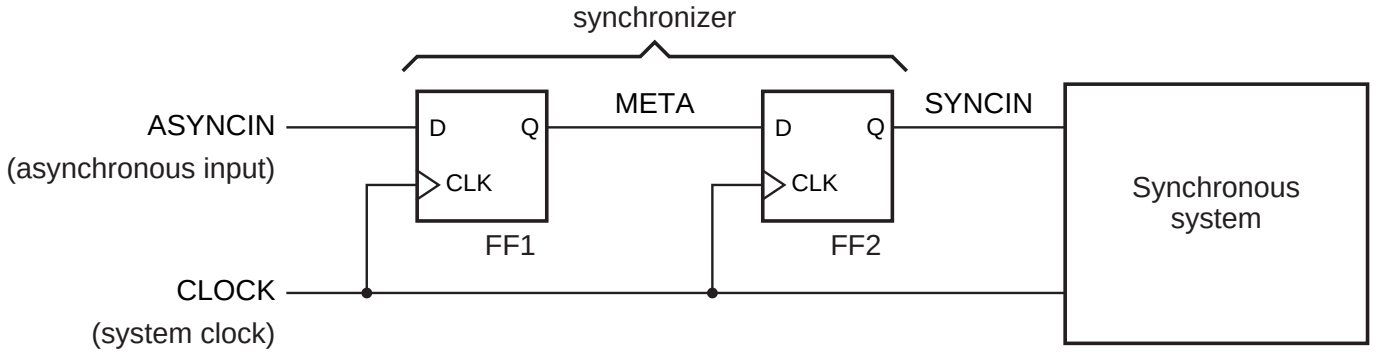


Duration of metastable outputs: $\text{MTBF}(t_r) = \dfrac{\exp(t_r/\tau)}{T_o \cdot f \cdot a}$

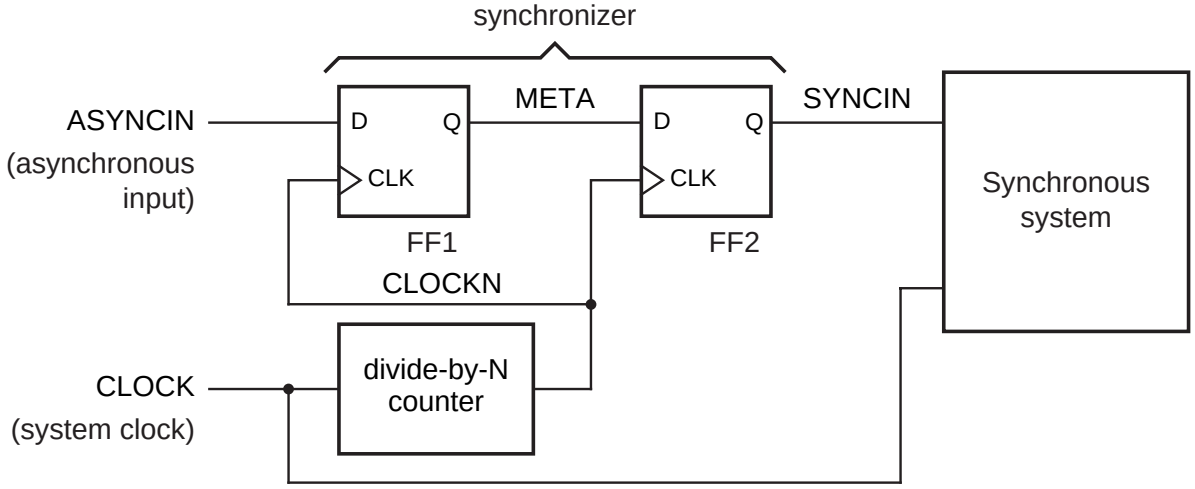
"MTBF" is *mean time between failure*; $f =$ flip-flop clock frequency; $a =$ asynchronous input changes per second; T_o and τ are parameters depending on flip-flop technology.

Synchronizers

Two-level synchronizer:



Multiple-cycle synchronizer:

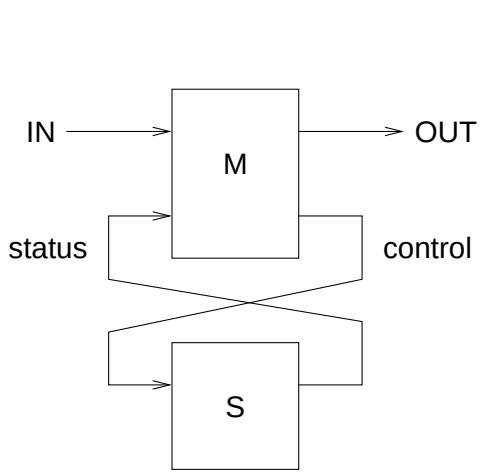


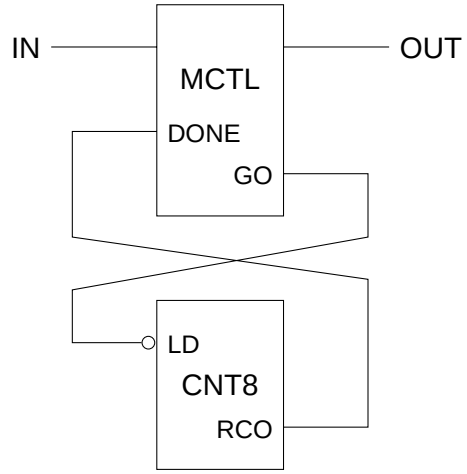
State machine decomposition

Important logic design principle: break down the problem into

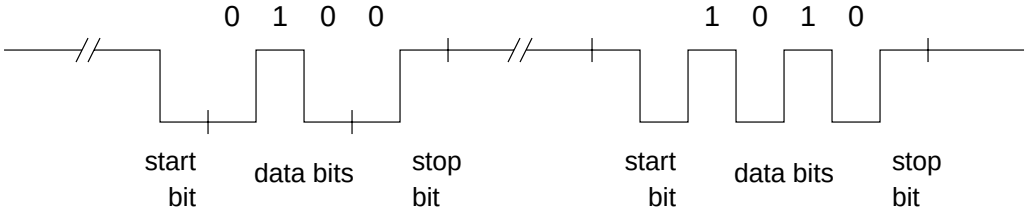
- simple(r) components, with
- simple (and well understood) connections between components

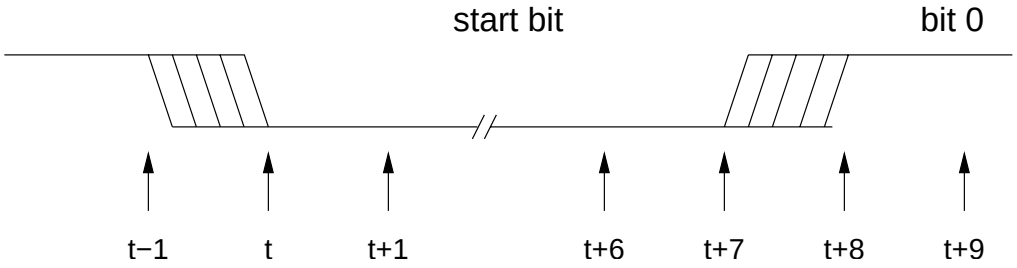
Typical master controller and subordinate state machine and concrete example:

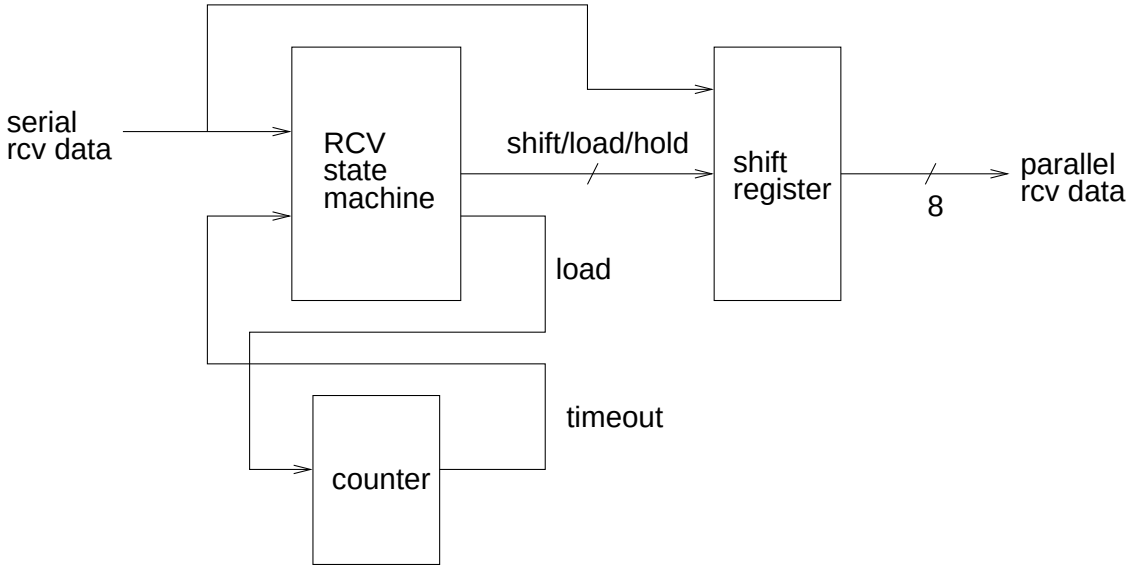




Asynchronous serial communications

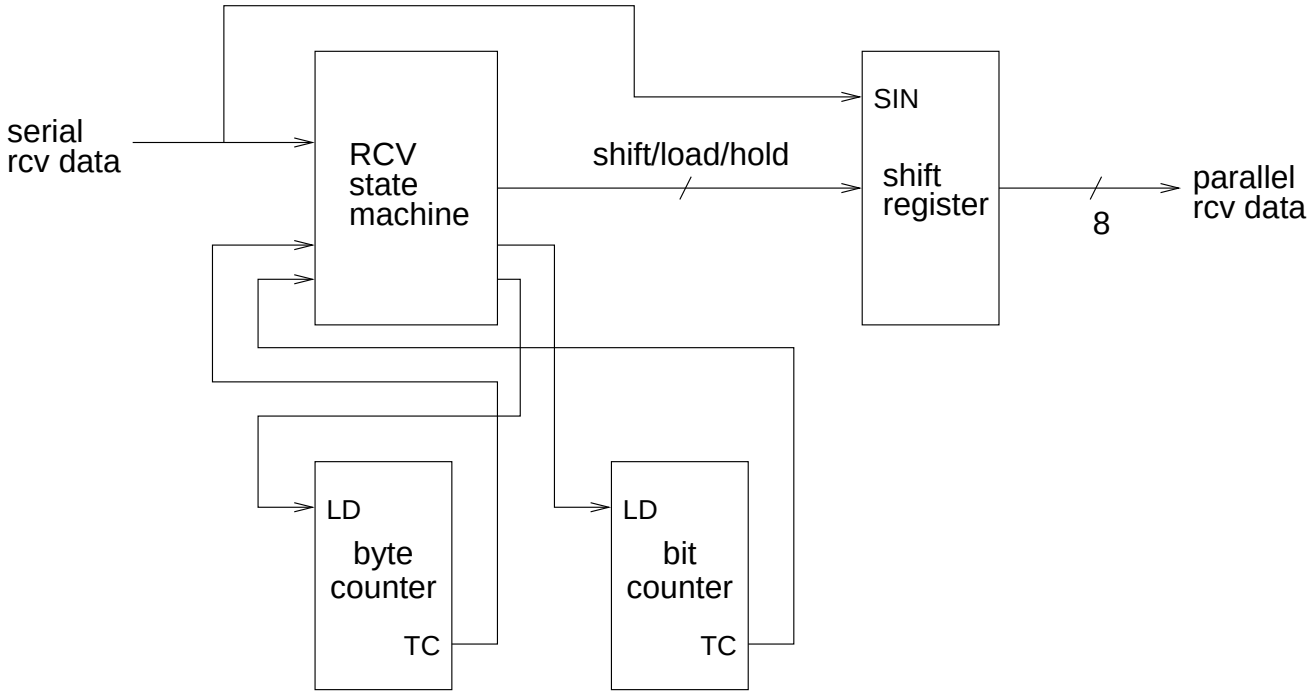






State machine decomposition

Improved receiver state machine: move states from master controller to counter.



Simplified master controller has only four states, IDLE, START, DATA, STOP.