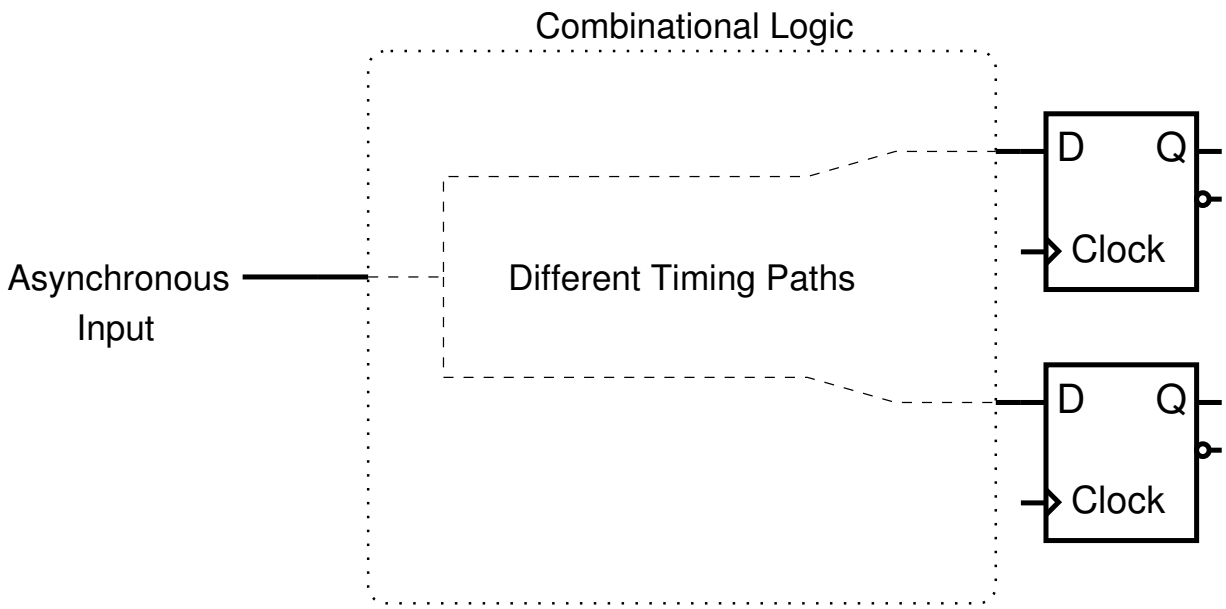
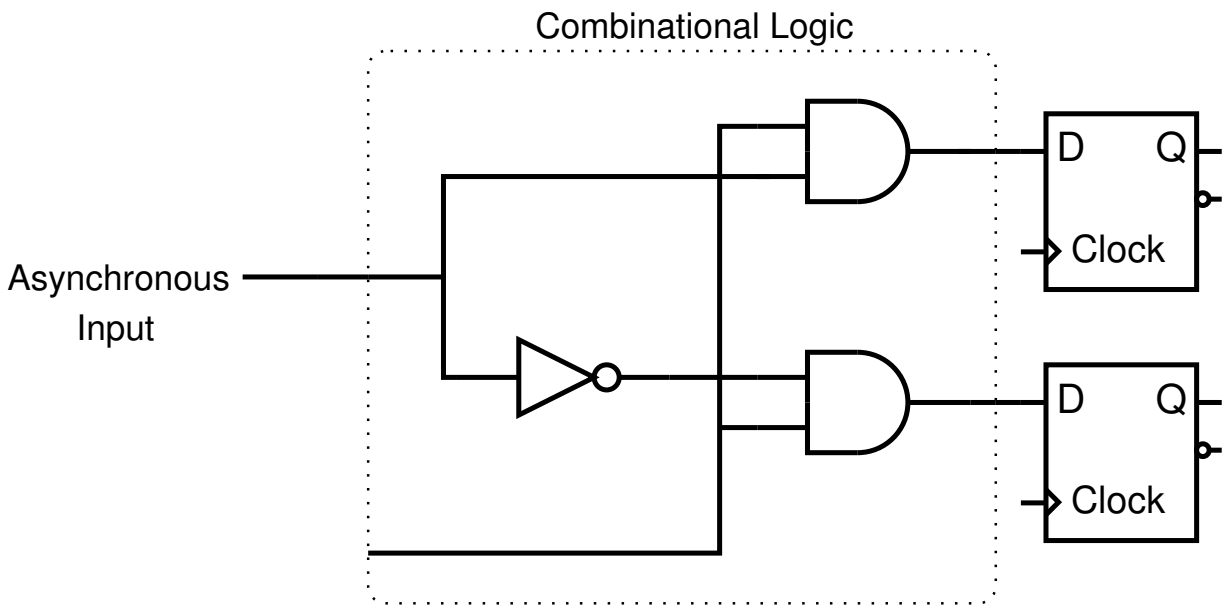
Asynchronous Inputs



With different timing paths through combinational logic, we can get unexpected results.

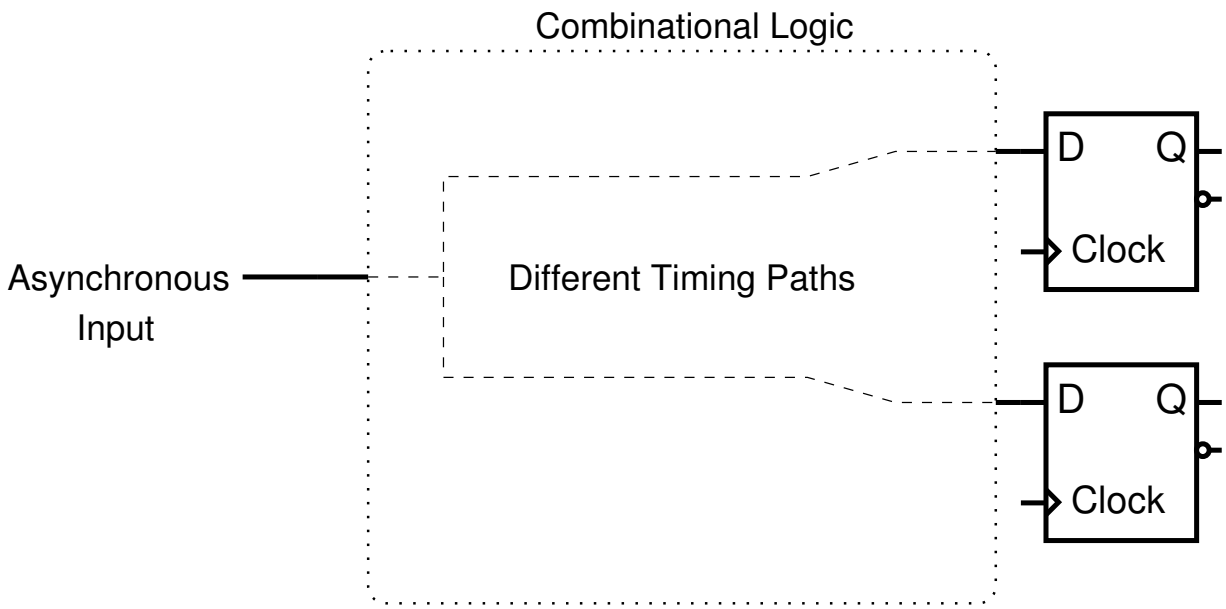
Asynchronous Inputs



With different timing paths through combinational logic, we can get unexpected results.

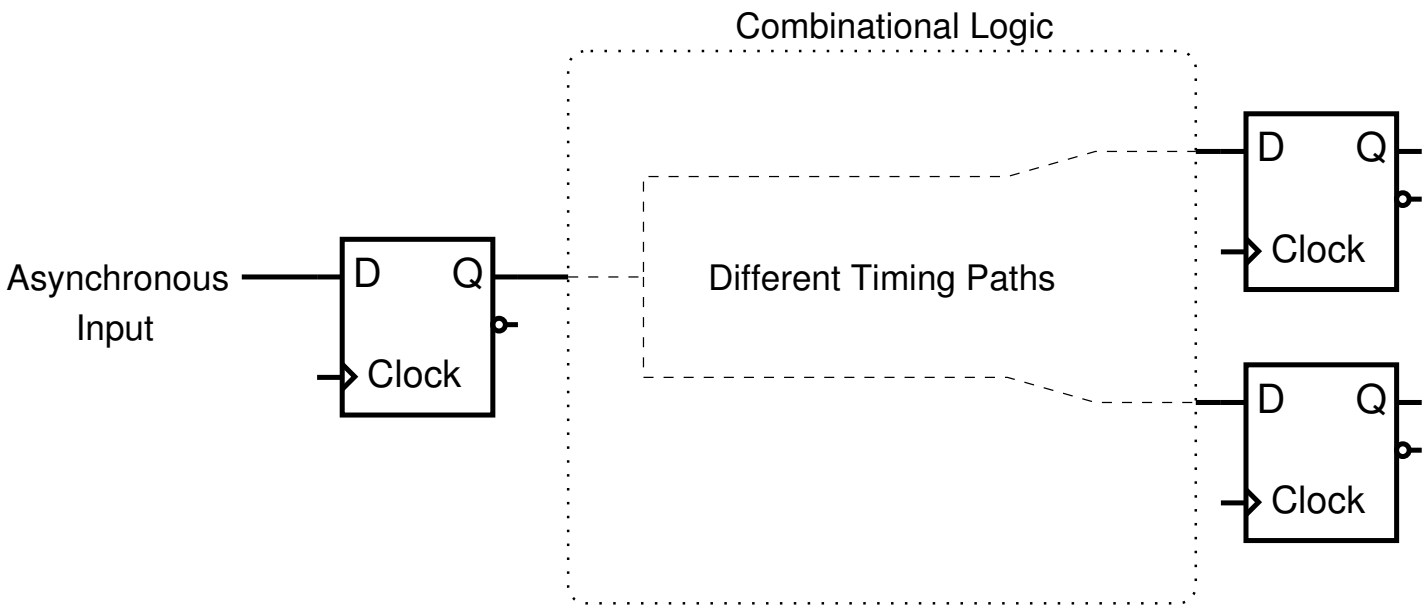
In this example, a falling input we may be registered as neither high nor low.

Asynchronous Inputs



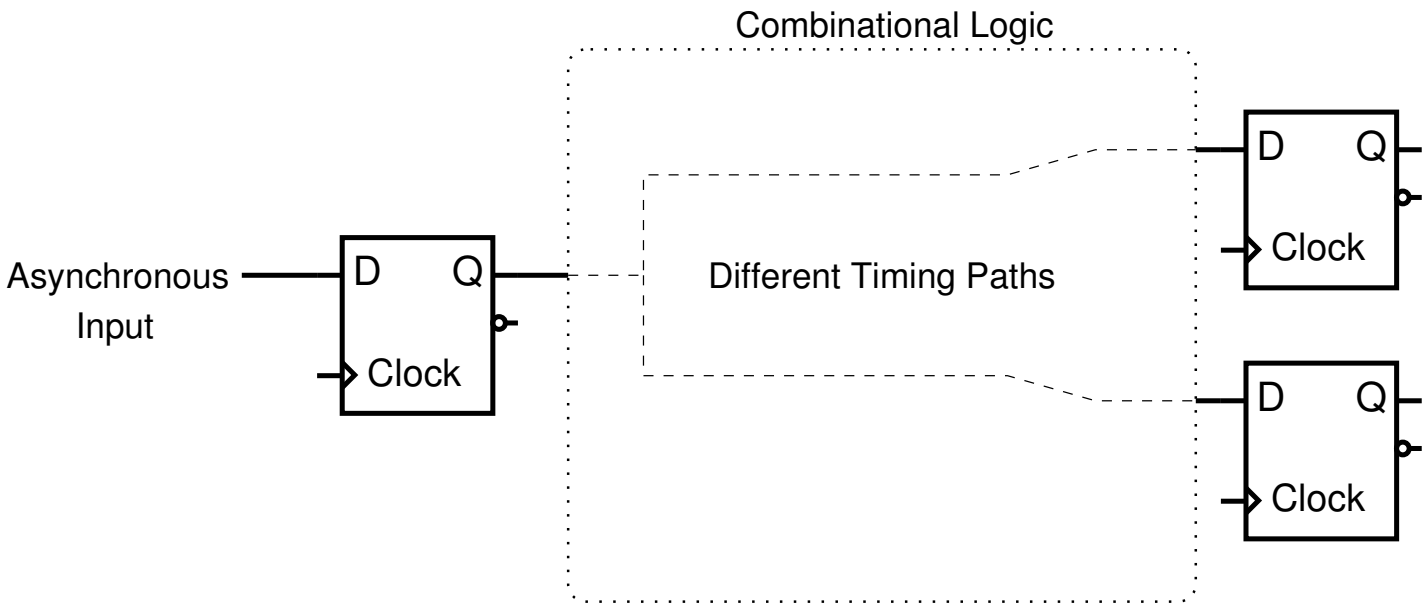
Retiming the asynchronous input before the combinational logic block should give more predictable results.

Asynchronous Inputs



Retiming the asynchronous input before the combinational logic block should give more predictable results.

Asynchronous Inputs

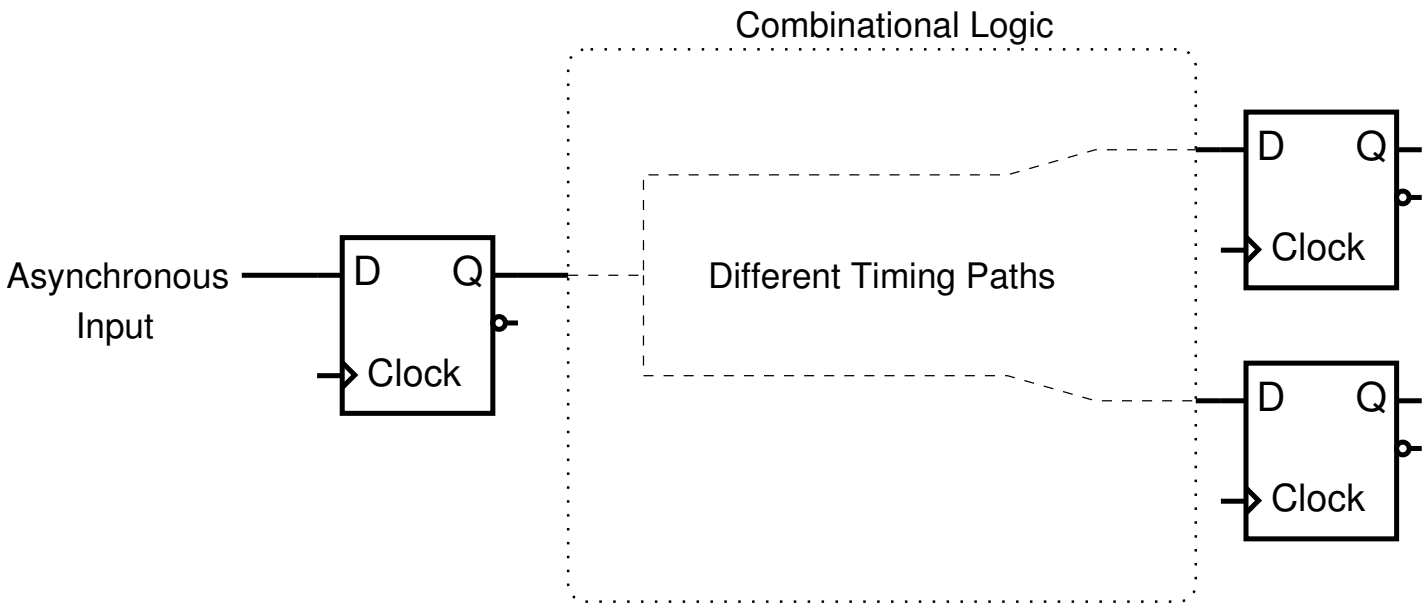


To avoid a setup violation[1]:

$$ClockPeriod > t_{pQ} + t_{critical_path} + t_{setup}$$

[1]assuming ideal clock

Asynchronous Inputs



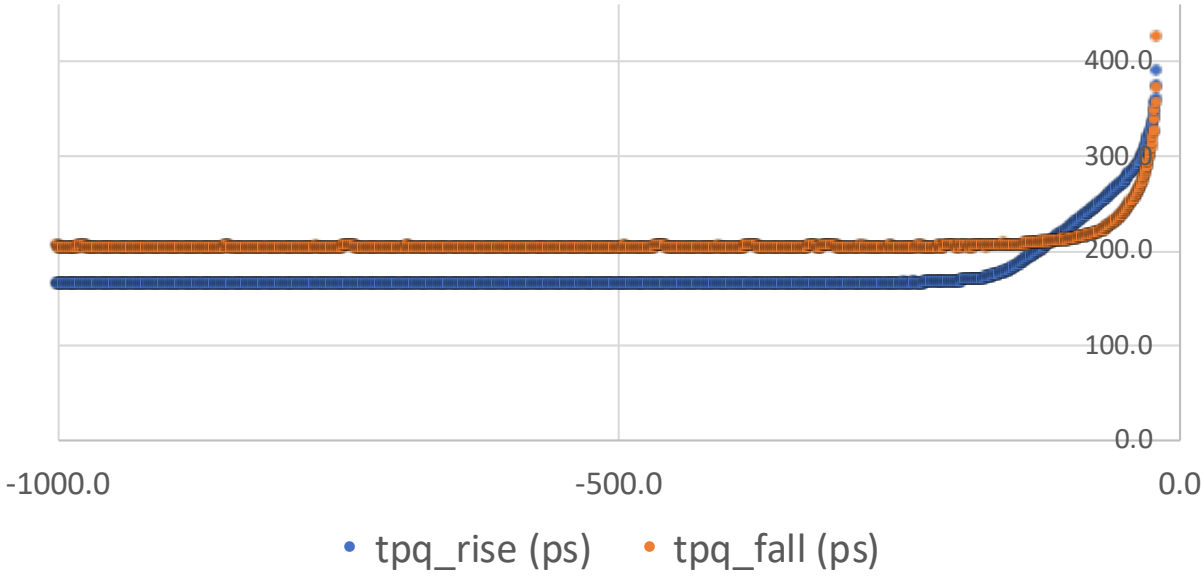
To avoid a setup violation[1]:

$$ClockPeriod > t_{pQ} + t_{critical_path} + t_{setup}$$

but t_{pQ} may now be unpredictable

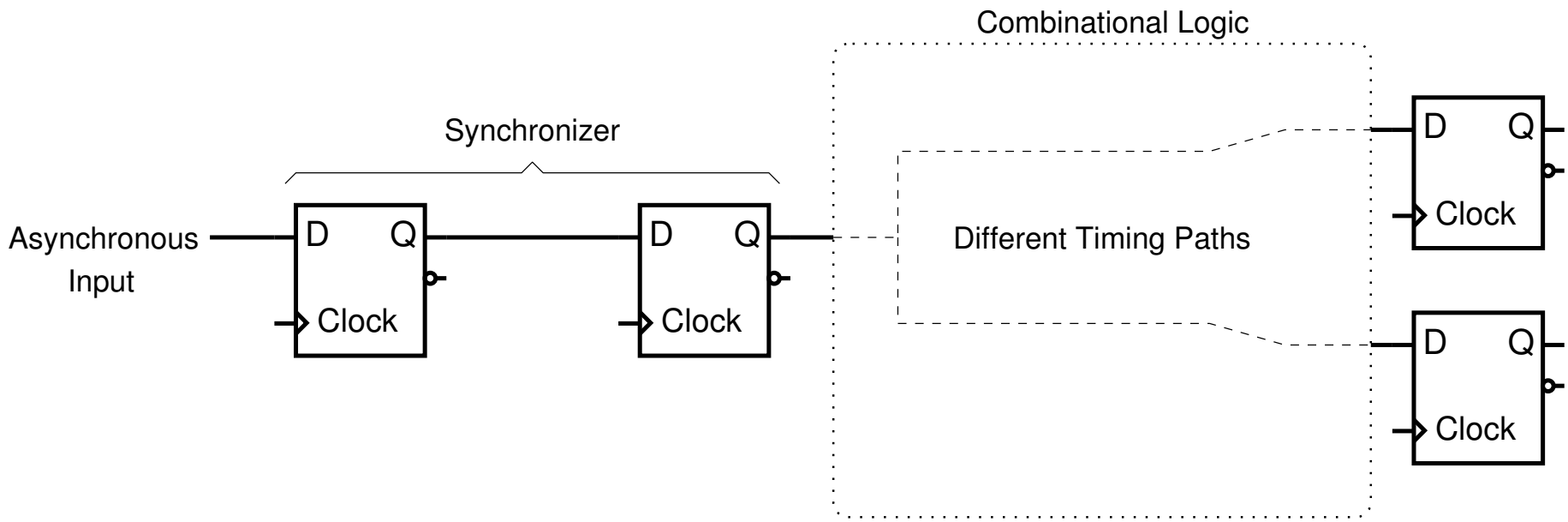
[1] assuming ideal clock

t_{pQ} Variation with Late Arriving Data



- As the setup time (t_{setup}) is reduced (close to it's absolute minimum value), the clock-to-Q delay (t_{pQ}) increases sharply.
- A flip-flop dasheet will quote a t_{pQ} value for a minimum t_{setup} value. When we violate the minimum specified t_{setup} value, the quoted t_{pQ} value is no longer valid.

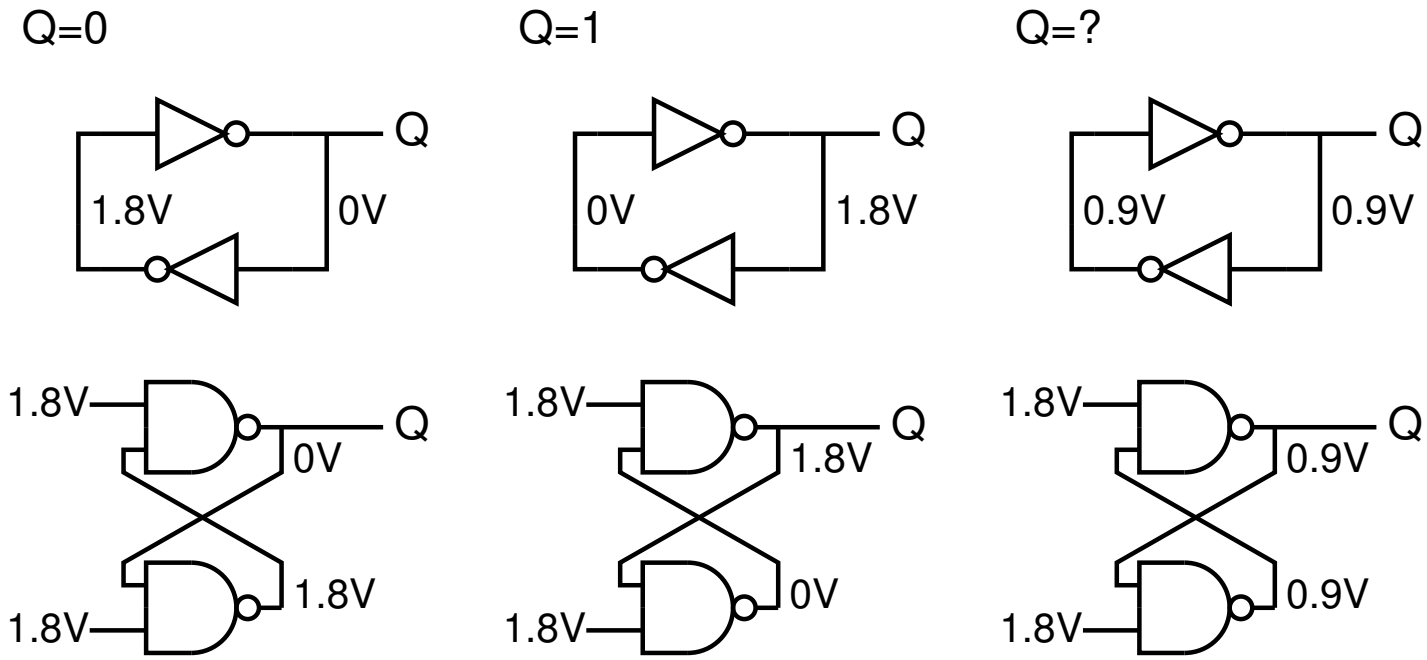
Asynchronous Inputs



A second flip-flop in the synchronizer gives a predictable t_{pQ} for our equation:

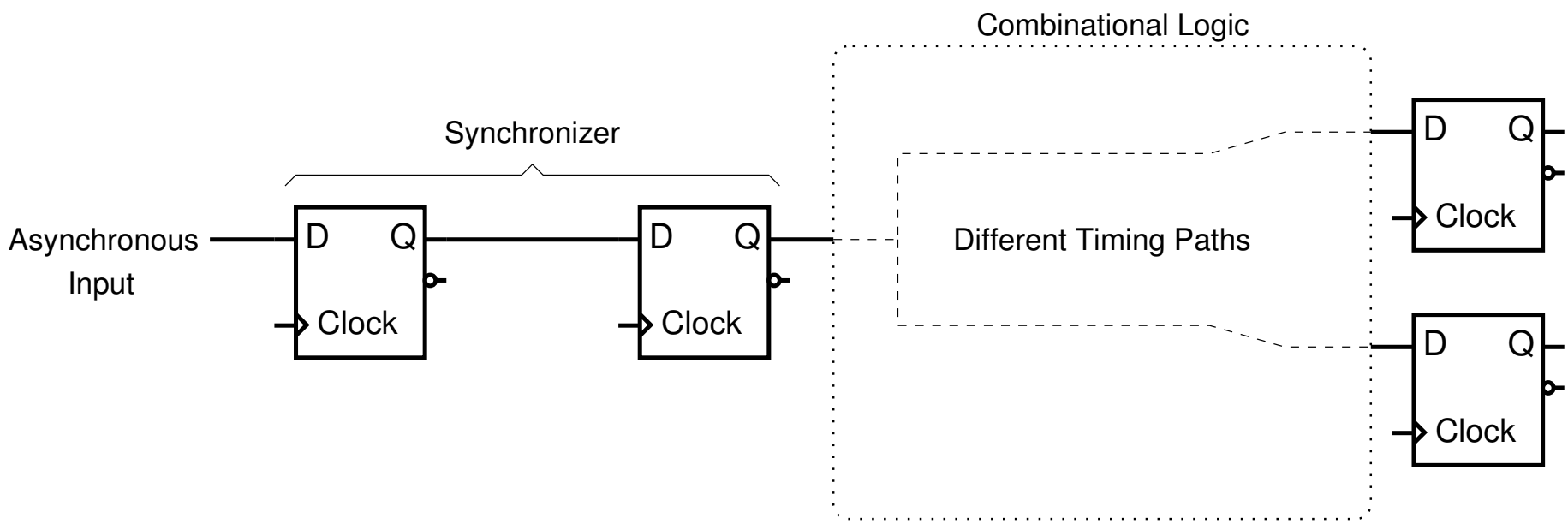
$$ClockPeriod > t_{pQ} + t_{critical_path} + t_{setup}$$

Flip-Flop Stable States



The third state is described as *metastable* since a slight perturbation in one of the 50%-of-V_{DD} voltages will result in a move to one of the other states.

Metastability



- The first flip-flop in our synchronizer may go metastable in the extreme case where the set-up time for D is reduced and t_{pQ} increases.
- The two flip-flop synchronizer gives a full clock cycle for the metastable state to resolve itself into one or other normal state.