Metastability & Synchronizers

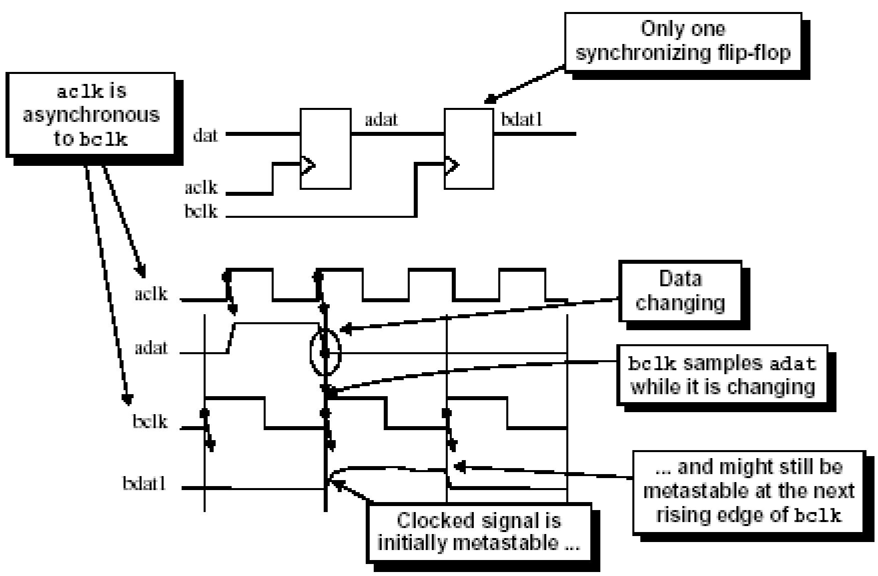
In the real design world there are very few designs with a single clock. Most of the ASIC's designed are driven by multiple Asynchronous Clocks which require special handling and understanding to ensure their timely completion. Some key issues to synchronize the design in spite of multiple asynchronous clocks are discussed here.

What is Metastability?

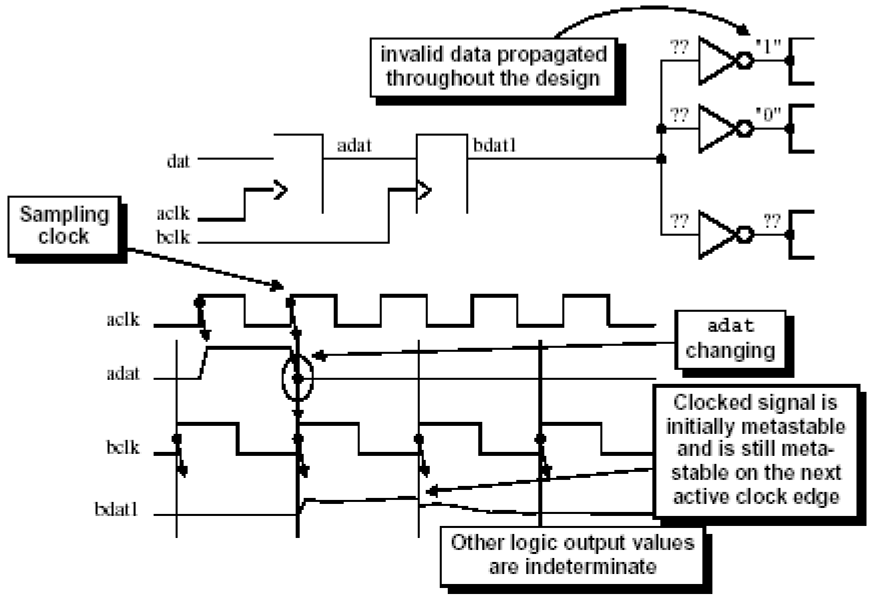
Quoting from Dally and Poulton's book…

"When sampling a changing data signal with clock, the order of the events determines the outcome. The smaller the time difference between events, the longer it takes to determine which came first. When two events occur very close together, decision process takes a longer time than allotted and it results in synchronization failure".

Synchronization failure occurs when the output goes into an unknown (Metastable) state and does not converge into a legal stable state by the time output has to be sampled again. Taking an example to explain the loss of synchronization…



In the above figure there are two clocks, namely **aclk** and **bclk** asynchronous to each other. Since the **adat** signal generated in **aclk** domain is sampled too close to the rising edge of the **bclk** clock, synchronization failure occurs and results into Metastable signal as output **bdat1**. Sometimes when the output does not converge to a legal stable state, metastable output causes illegal signal values to propagate throughout rest of the design.



As shown above, the output **bdat1** remains in Metastable state in the next active clock edge, instead of going to a legal stable value of zero due to which illegal values are transmitted.

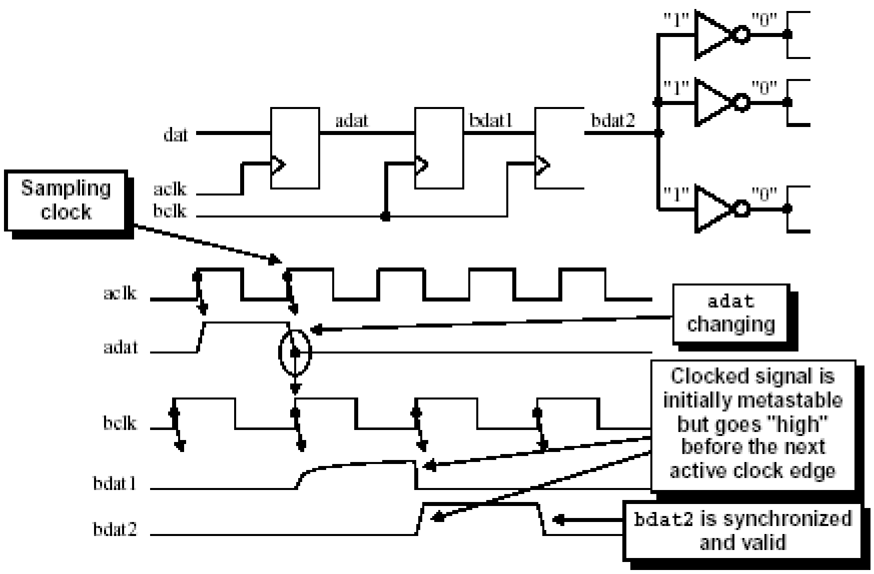
To overcome the problem of Metastability to some extent, each flip flop in the design has a specified setup and hold time, It is the time during which data input is not legally permitted to change before and after the rising clock edge and is specified as a design parameter in a precise manner to keep a data signal from changing too close to another synchronizing signal that could result into Metastable output.

If the Metastable output comes to a stabilized legal value within a single clock period, still synchronization is needed so as to get the actual input, which is fed, for there is a need of **synchronizer**.

What are Synchronizers?

Quoting from Dally and Poulton's book…
"A synchronizer is a device that samples an asynchronous signal and outputs a version of the signal that has transitions synchronized with the sample clock".

The most common synchronizer is a two flip-flop synchronizer…



The second flip-flop samples the input and waits for a full clock cycle to permit Metastability in the **bdat1** signal. This signal is again sampled by the same **bclk** in the third flip-flop to get a valid and synchronized signal **bdat2** in the **bclk** clock domain.

There can be possibilities that **bdat1** signal still is in Metastable condition when sampled. For such cases the calculation of the probability of the time between the synchronization failures is a function of variables involved in generating input signals and clocks, but for most of the synchronization applications, the two flip-flop synchronizer works well.