System reset

Abstract
The presented circuit makes the balancing of the reset tree with the clock tree superfluous. That saves silicon area and reduced times and effort in the layout team.

Reset tree
Most ASIC engineers know about clocks and clock trees. Not everybody has been involved in balancing clock trees but if you have been in the industry for a few years you have at least heard of it. What is less know is the reset tree balancing. When I started in the industry many years ago it was practice to build a reset tree alongside a clock trees.

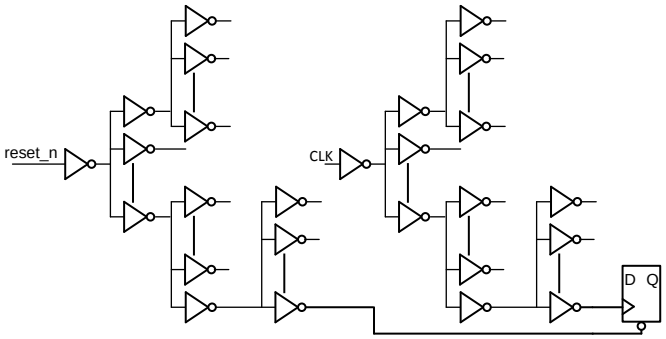

Figure 1: balanced clock and reset tree.

Why is reset tree balancing required?
Again: as designer you will have heard about set-up and hold time of registers. That is, the clock and the input signals should arrive somewhat separated in time. However the same is valid for the reset and the clock. Assertion of the reset can happen asynchronously. However when the reset is removed it should be time-separated from the active clock edge, just as with any other input signal to the register.

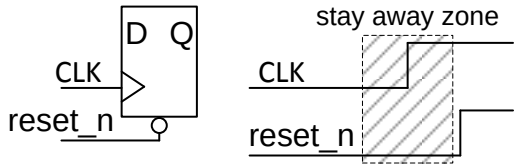

Figure 2: reset release away from the active clock edge.

Adding a reset tree to a chip has a large impact in workload and silicon area.

Workload.
The work poses an extra challenge for the layout engineers. Not only have they design two signal trees, these have to balanced with each other and the rest of the logic over the whole of the voltage, temperature and process range. This quadruples the work in that area. With this circuit timing closure of the reset tree becomes almost trivial.

Silicon area.
The reset tree takes up valuable silicon area. Clock trees are often laid double width and double-spaced to reduce delay and cross talk. To balance the two trees, the reset tree is sometimes laid out the same way. With this circuit the reset tree can be kept extremely small.

Solution.

Just as with some other solutions I present, the answer is ‘obvious’ (in hindsight). Instead of keeping the clock running during a reset, the clock is switched off, after a short delay the reset is release, then after a second delay the clock is switched on again.

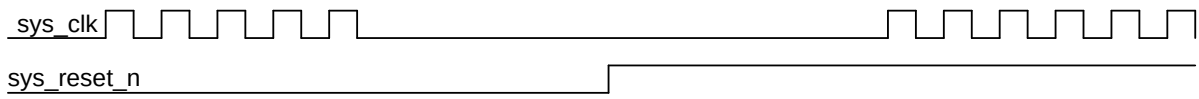

Figure 3: system_reset output signals.

It is necessary to have the clock running with a reset asserted. This is to pass the reset state on if there are any following registers which are not connected to the reset. Therefore the xtal_startup module which I have posted on the web cannot always be used.

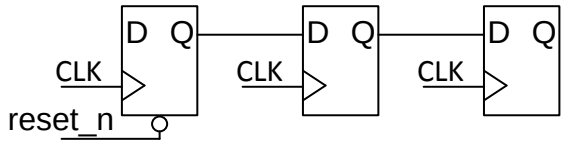

Figure 4: reset state pass-on.

Caveat emptor

Beware that the method described above does NOT work if the system contains reset synchroniser registers. Even if the designer does not use them, they can be present in IP which is bought in. A prime example is a major supplier of advanced risk processors[1], who puts reset registers in the design and clocks the reset through on the negative clock edge. To cope with that I had to develop a slightly modified version of the design described in here.

Copyright

[1] I asked one of their main designers and he told me ‘because our customers want this'.