LCA Speed Estimation: Asking the Right Question

XAPP 011.001 Application Note By BERNIE NEW

Summary

A simple algorithm is described for determining the depth of logic, in CLBs, that can be supported at a given clock frequency. The algorithm is suitable for XC3000/XC3100 or XC4000 LCA devices.

Speed is always a consideration when deciding whether a design can be implemented in an LCA devices. Often, an initial logic design is created and the question asked, “How fast will this run in an LCA device?”

This is not an easy question to answer. A good speed estimate requires careful analysis of the logic design; performance will vary with the logic implementation. To complicate matters, routing delays are always unknown at this stage.

When the estimate is complete, it is usually compared to a given system requirement simply to determine adequacy, and the exact number becomes irrelevant. If a system requires 30 MHz, for example, being able to operate at 35, 40 or even 50 MHz makes no difference.

A better question is “Will an LCA implementation meet the system speed requirements?”

This can often be answered much more easily. Given a required clock rate, it is easy to estimate the level of complexity that can be supported. This complexity can then be compared to the functional requirements to make an initial determination of feasibility. Only in marginal cases does a full speed estimate become necessary.

A typical data path runs from a register, through some combinatorial logic to another register. In an LCA device, this requires, as a minimum, a CLB clock-to-output delay plus a set-up time. In an XC3000-125 part, these total 10.5 ns. Including routing, 15 ns should be typically allowed. If combinatorial CLBs are added into the path, each level of CLBs adds 5.5 ns. Additional routing delays are also created. Including a typical routing allowance, 10 ns should be added for each level of combinatorial CLBs.

This simple speed-estimating procedure can also be reversed. If, for example, the system clock frequency is 30 MHz, the 33 ns period typically provides for two combinatorial CLBs.

Clock period	33 ns
Minimum delay	-15 ns
	18 ns
Combinatorial delay	÷10 ns
	~2 CLBs

Including the function generator in the destination CLB, a total of three function generators can be cascaded. If the number of function generators that can be cascaded is known, the design can be analyzed to determine whether or not it is feasible.

This should not be considered a hard limit. Shorter routing delays can be achieved, allowing deeper logic. However, dependence on short routing delays will probably necessitate optimization of both the logic design and the routing.

Nor is the number of function generators guaranteed. Longer routing delays may be encountered, especially if a chip is fully utilized or if high fan-out signals are used. Elimination of these long routing delays may necessitate manual routing or logic design changes. In any case, the timing of all LCA designs should be analysed after routing to determine worst-case performance.

Table 1 shows typical minimum delays for various LCA devices. Also shown are typical increments for combinatorial CLBs. To allow for higher routing delays, these figures should be increased by 5 ns, if more that 60 – 75% of the CLBs are to be used. If a large LCA device is to be used and the CLBs are placed automatically, a separate 3 – 5 ns should be added to each delay.

This technique not only simplifies the feasibility study, it also provides valuable information on which to base the logic design. Critical areas can be identified prior to starting the design. It is better to design around the critical areas than to have to accommodate them during implementation. Conversely, if a design only requires a fraction of the capability available, it might be possible to multiplex some functions to provide a less costly implementation.

Table 1. Delays, Including Typical Routing

	XC3000			XC3100			XC4000		
	-70	-100	-125	-5	-4	-3	-6	-5	
Minimum delay	21	18	15	10	9	7	17	12	ns
Combinational delay	15	12	10	8	7	6	12	9	ns
To each delay add:	5 ns for high utilization 3 – 5 ns for large LCA Devices								