

Summary

The XC4000 dedicated carry logic provides for very compact, high-performance counters. This Application Note describes a technique for increasing the performance of these counters using minimum additional logic. Using this technique, the counters remain loadable.

Xilinx Family

XC4000/A/D/H

Demonstrates

Dedicated Carry Logic
High-performance Counter Design

Introduction

The dedicated carry logic in XC4000 LCA devices provides a mechanism for very fast and efficient counters. While the ripple-carry scheme appears simplistic, the hardware implementation of the dedicated carry logic is very fast, and requires few CLBs. In fact, the implementation is so efficient that it defeats most attempts to replace it. It is possible, however, to augment the operation of the carry logic and obtain higher performance.

To reduce the ripple-carry delay, the effective length of the carry path must be shortened. This is achieved by dividing the counter into two sections that settle in parallel, as shown in Figure 1. The carry output of the less-significant section provides a parallel Count Enable (CEP) to the more-significant section.

The carry delay is reduced to the settling time of the more significant section, or the settling time of the less significant section plus the subsequent routing and count-enable times, whichever is greater. For optimum

performance, these times should be balanced, requiring that the counter be divided into two unequal parts.

The use of CEP does not imply that these are prescaler techniques. In a prescaler counter, CEP is typically decoded from the least significant two or three bits. The CEP signal is then used to enable the remaining bits, such that their effective clock rate is one fourth or one eighth of the actual clock rate. This allows multiple clock periods for the remaining bits to settle, and the whole counter can be operated at the speed of the prescaler.

Using the prescaler technique, it is not possible to load the counter and guarantee that it will count correctly on the following clock cycle. The carry chain in the more significant bits is designed to settle in multiple clock periods. If the loaded data causes these bits to be enabled on the clock following the load operation, the carry path will not, in general, have had adequate settling time. Depending on the value loaded, it might not be possible to resume counting for several clock periods after the load operation.

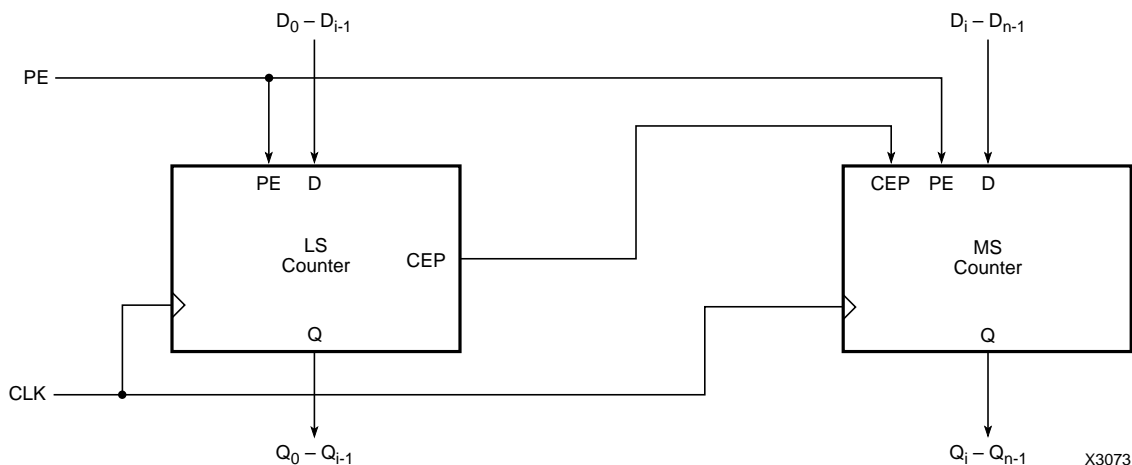


Figure 1. Accelerated N-Bit Counter

The acceleration technique described in this Application Note does not depend upon carry chains having multiple clock periods in which to settle; the entire carry chain settles within one clock period. However, the clock period is reduced because parallelism is introduced into the carry chain. The improvement is not as dramatic as with a prescaler, but loadability is retained.

Two versions of the technique are described below. One version uses two dedicated carry-logic chains, and is increasingly effective in longer counters. For shorter counters, a second version uses CLBs for the less significant section, and decreases the clock period by a fixed amount (1.5 ns in an XC4000-5). While the benefit from this second version is small, it can sometimes be crucial. Figure 2 illustrates the benefits derived from the two versions. In either case, one additional CLB is required to accelerate the counter.

Operating Description

Long-Counter Version

To accelerate long counters, the carry chain must be divided into two unequal parts. The less significant section should be shorter to accommodate the distribution and set-up times of CEP. For optimum performance, each section of the counter should contain an odd number of bits.

If the counter length is an exact multiple of four, the more-significant section should be 10 bits longer than the less-significant section. A 32-bit counter, for example, should be split into sections of 11 and 21 bits.

This split creates a 7.5-ns difference in settling times to accommodate the additional delay. The set-up time is 4 ns, and consequently, 3.5 ns is available for routing. A Longline should easily meet this requirement, leaving the speed controlled by the more-significant section of the counter.

As described in the Application Note, Estimating the Performance of XC4000 Adders and Counters (XAPP 018), the estimated minimum clock period for an N-bit counter is the following.

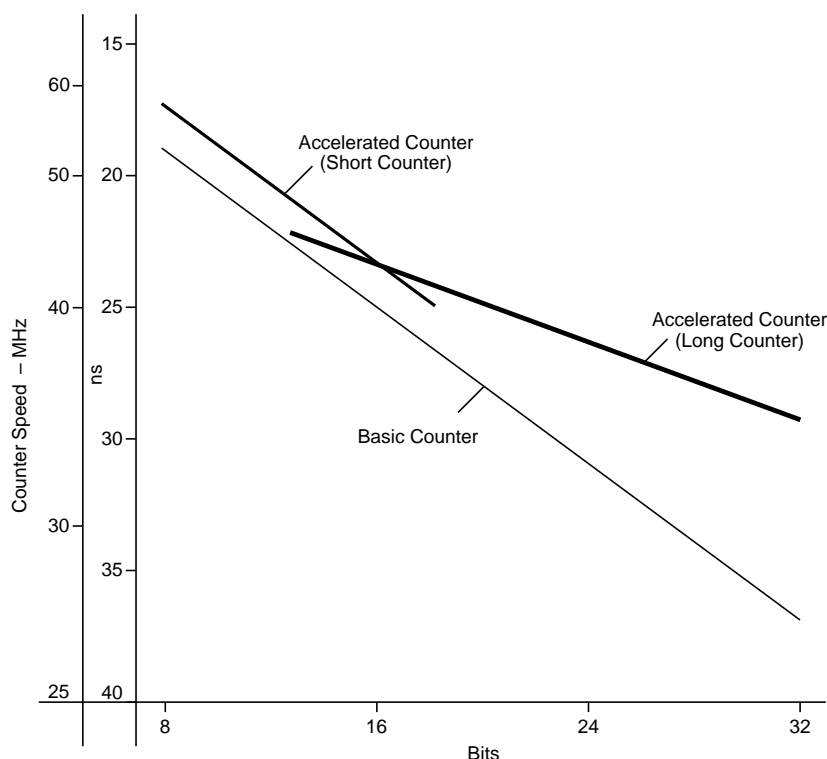
$$t_{\text{CLK-CLK}} = 13 + 0.75N \text{ ns}$$

Assuming that the speed of the accelerated counter is determined by the more-significant section, this reduces to the following.

$$t_{\text{CLK-CLK}} = 17.5 + 0.375N \text{ ns}$$

As a result, the clock period of a 32-bit counter is reduced from 37 ns to 29.5 ns.

For counters with an even length that is not divisible by four, the more-significant section should contain eight



X3074

Figure 2. Counter Speed Comparison (Max Speed vs Counter Length)

more bits than the less-significant section. In this case, the speed of the counter will be controlled by its less significant section plus the additional CEP delays. While the minimum clock period is no longer as well-defined, it is again approximated by the above formula.

Splitting the counter into odd-length sections, one function generator is available in each section. As shown in Figure 3, these function generators can be used to generate CEP and Terminal Count (TC). To permit this, they should be G function generators, and share CLBs with the MSBs of each section.

The CEP signal uses CLB Enable Clock pins to control counting in the more significant section. Consequently, it must be forced to a one while the counter is being loaded. CEP is, therefore, defined as $C_{OUT0} + PE$.

The carry input to the more-significant section of the counter is forced to a one, and the carry chain in this section is independent of the less significant bits. In order for TC to reflect the state of the entire counter, it must be generated as $MSCOUNT0 \bullet LSCOUNT0$.

One benefit of this counter is that TC is available without additional time delay or CLB cost. The CLB count of the accelerated counter matches that of the unaccelerated counter if TC is generated. If TC is not required, the unaccelerated counter can be one CLB smaller.

Short-Counter Version

For counters shorter than 16 bits, the following design should be used. It is based on the same fundamental approach as the counter described above, but offers greater benefit in short counters.

As shown in Figure 4, the less significant section of the counter is two bits long, and is implemented using function generators instead of the dedicated carry logic. The more significant section of the counter is $N-2$ bits long and is implemented using the carry logic.

As in the previous design, CEP is forced to a one while the counter is loaded. This permits the enable clock pin to be used as Count Enable with the Parallel Enable taking priority.

The 1.5-ns performance advantage requires that the counter speed be dominated by the more significant section, which is two bits shorter than the unaccelerated counter and, therefore, faster. With good routing, this requirement can be met in counters of six or more bits.

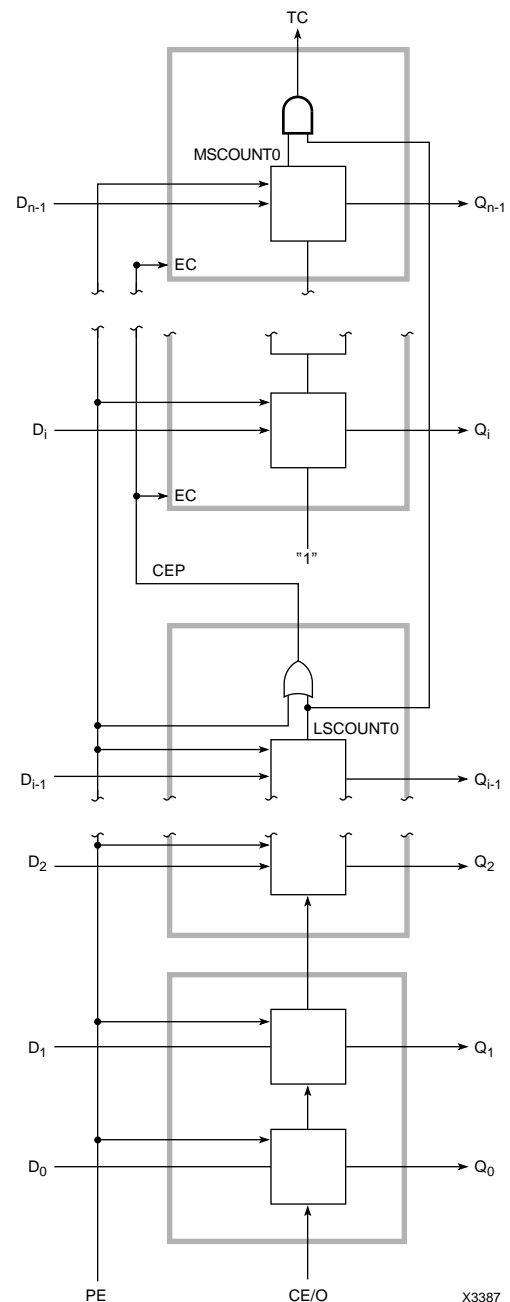


Figure 3. Long Accelerated Counter

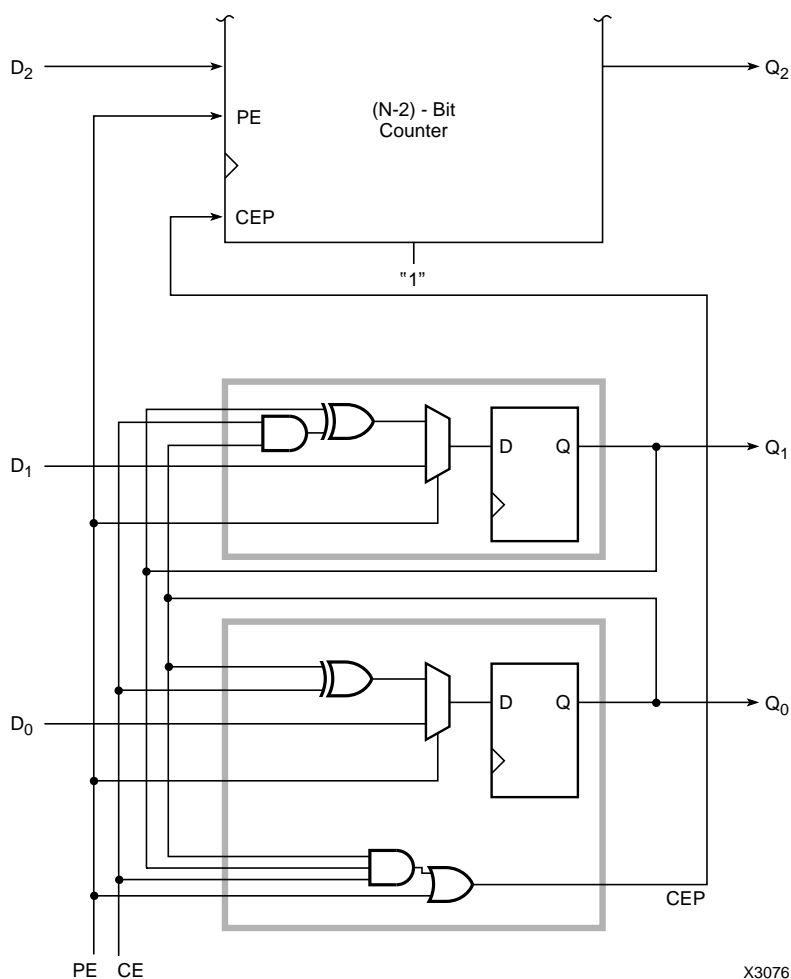


Figure 4. Short Accelerated Counter