

Summary

The design strategies for loadable and non-loadable binary counters are significantly different. This application note discusses the differences, and describes the design of a loadable binary counter. Up, down and up/down counters are described, with lengths of 16 and 32 bits. Design files are available for all six versions.

Specifications

Length	16 Bits
Maximum Clock Frequency XC3100A-2	60 MHz
Number of CLBs	23

Xilinx Family

XC3000A/XC3100A

Demonstrates

Fast Counter Technique

Background

When designing a non-loadable counter, the fastest designs use some form of prescaler technique to exploit the fact that the more significant bits toggle much more slowly than the less significant bits.

The carry chain for the first few bits of the counter can usually be implemented in parallel and is very fast. However, the carry chain for the more significant bits usually requires multiple levels of gating and is much slower. Using prescaler techniques, the counter can operate at the speed of the less significant bits, by giving the more significant bits several clock periods in which to settle.

Typically, a 2- or 3-bit prescaler generates a high-speed count-enable signal that is broadcast through the more significant bits every four or eight clocks. In between these enables, the more significant bits are stable; the carry chain for these bits, therefore, has four or eight clocks periods in which to settle, instead of one.

These techniques depend upon the predictability of the binary sequence, and the implied low-speed operation of the more significant bits. When a counter is loaded, however, the binary sequence is disturbed, and its predictability is lost. To ensure correct operation following a load, the carry chain for the entire counter must settle before the next clock.

This reduces the speed of a prescaler counter significantly. Its operating frequency becomes constrained by the slow more significant bits rather than by the fast prescaler.

There are techniques such as pulse-swallowing and state-skipping that can be used to load a prescaler counter without loss of speed. However, these result in non-binary operation for a short time after loading, and some load values are not permitted.

Loadable Binary Up Counter

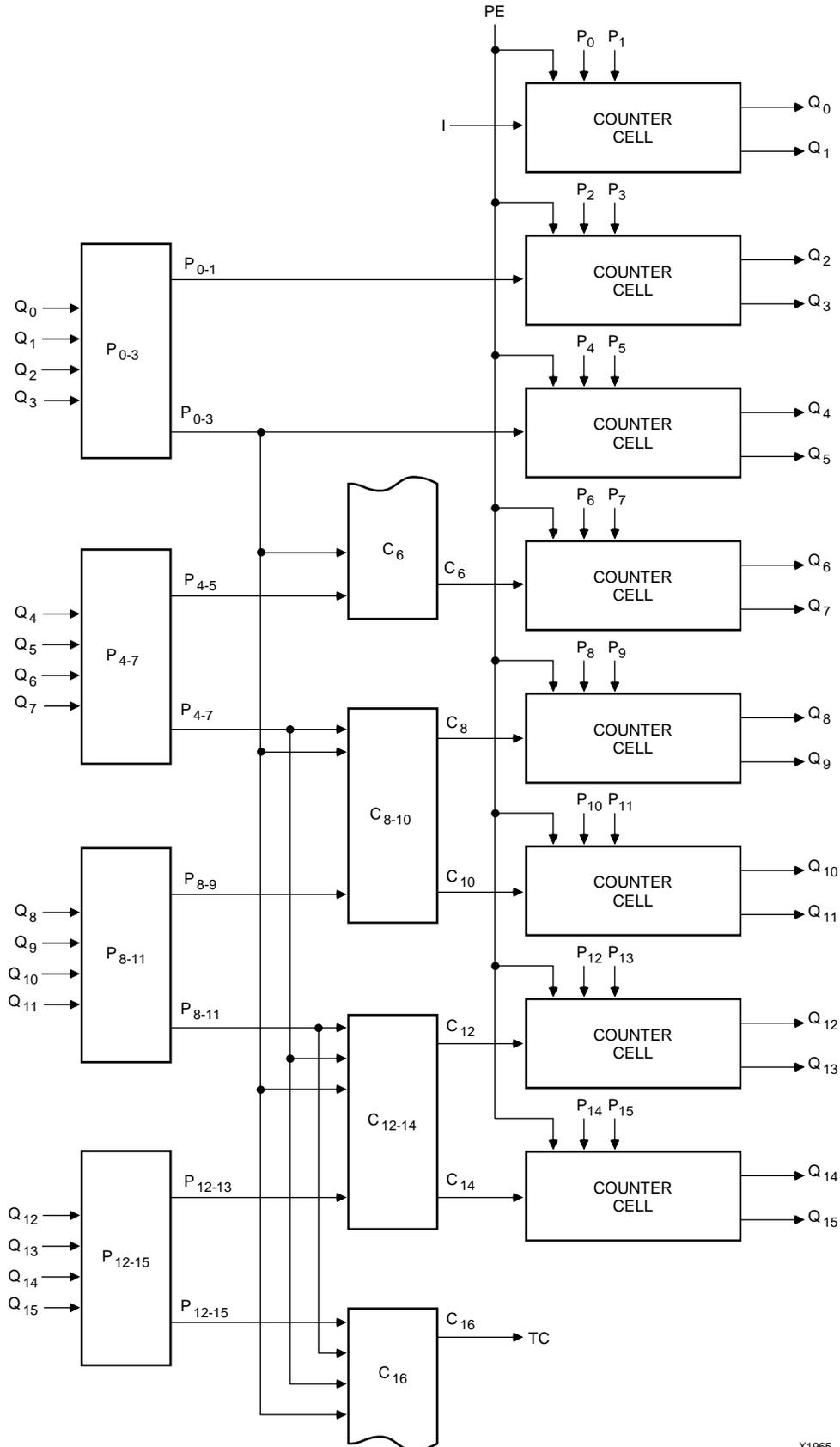
When designing a loadable binary counter, emphasis must be placed on balancing the carry delays. Unlike the prescaler counter, high-speed paths are of no benefit, and slow paths cannot be hidden. Figure 1 shows a good example of a loadable binary counter.

This counter is based on a 2-bit cell, as shown in Figure 2. The two bits are implemented in two CLBs, using loadable T-type flip-flops. Only one carry-in is required, the second carry-in being derived within the cell. The CLB clock enable may be used as Count Enable; however, the bits cannot be loaded while disabled. To overcome this, Parallel Enable must be ORED into the Count Enable line.

To form the carry chain, output bits are ANDed into groups of two and four, using the propagate cell shown in Figure 3. The propagate outputs are then ANDed together to form the even carries, according to the formulae of Table 1. Carries to the odd-weighted bits are generated within the counter cell.

With the exception of the trivial less significant bits, all carry delays comprise two levels of combinatorial CLB. This is longer than the direct paths from the less significant bits found in prescaler counters. However, prescaler counters typically have longer more-significant-bit delays, which is the chief speed constraint of a loadable counter.

The partitioning of the carry logic into the CLBs allows the counter to be implemented in an N-shaped configuration. A suggested placement of the CLBs is shown in Figure 4. Restricting the carry chain to a 2 x 4 block of CLBs minimizes the routing delays among them. With this organization, simulations show the counter will operate at 54 MHz.



X1965

Figure 1. 16-Bit Loadable Binary Counter

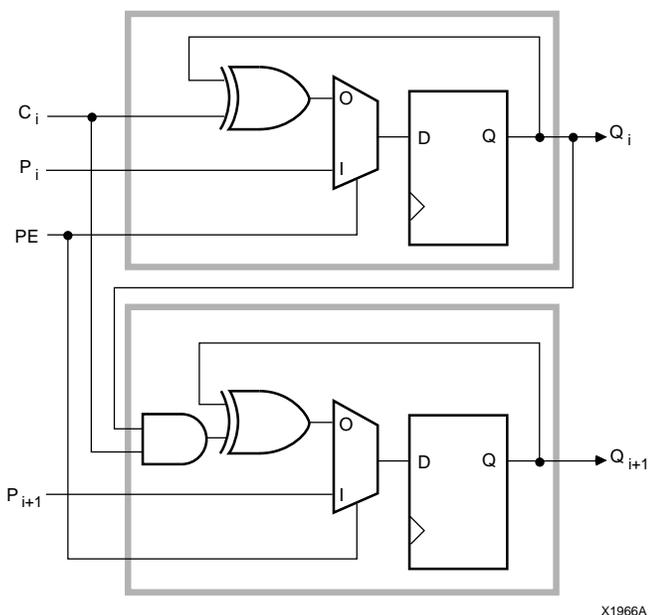


Figure 2. 2-Bit Counter Cell

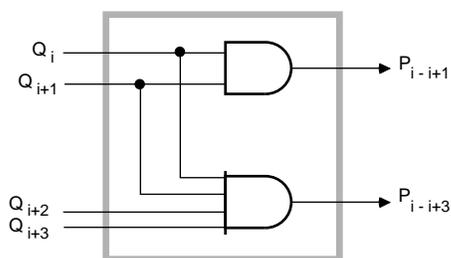


Figure 3. Propagate Cell

$$C_6 = P_{0-3} \cdot P_{4-5}$$

$$C_8 = P_{0-3} \cdot P_{4-7}$$

$$C_{10} = P_{0-3} \cdot P_{4-7} \cdot P_{8-9}$$

$$C_{12} = P_{0-3} \cdot P_{4-7} \cdot P_{8-11}$$

$$C_{14} = P_{0-3} \cdot P_{4-7} \cdot P_{8-11} \cdot P_{12-13}$$

$$C_{16} = P_{0-3} \cdot P_{4-7} \cdot P_{8-11} \cdot P_{12-15}$$

X2652

Table 1. Carry Logic Equations

Q ₀	Q ₁		C _{8/10}	Q ₈	Q ₉
Q ₂	Q ₃	P ₀₋₃	P ₈₋₁₁	Q ₁₀	Q ₁₁
Q ₄	Q ₅	P ₄₋₇	P ₁₃₋₁₅	Q ₁₂	Q ₁₃
Q ₆	Q ₇	C _{6/16}	C _{12/14}	Q ₁₄	Q ₁₅

X1969A

Figure 4. CLB Placement

For an 18-bit counter, C₁₆ may be used as carry-in to bits 16 and 17, as shown in Figure 5. Additional TC logic must also be included. This extension does not involve additional levels of logic, but may incur additional routing delays.

The 18-bit counter may easily be extended to 32 bits by replicating bits 4 through 17, and using TC/C₁₈ in the upper section in place of what was P₀₋₃. This entails one additional combinatorial delay, which reduces the maximum operating frequency to 37 MHz.

If this additional delay is unacceptable, two 16-bit counters may be concatenated, using C₁₆ as the clock enable to the counter bits in the upper half. However, this creates two problems. Clock enable can no longer be used to provide count enable, and the counter may only be loaded when the lower half is at terminal count.

Both of these problems can be overcome separately, but not together. If C₁₆ is moved to a separate CLB, a fifth input may be added. This could be Count Enable, which should be ANDed with the existing C₁₆, or Parallel Enable which should be ORed with it.

Loadable Binary Down Counter

If the counter bits are viewed as T-type flip-flops, the purpose of the carry chain is to determine which bits of the counter are to be toggled. For an up counter, a contiguous group of bits is toggled, starting with the least significant bit and extending up to, and including, the first zero. For a down counter, this group extends up to, and includes, the first one. The operation of the carry chain is the same in each case, but with the role of input ones and zeros reversed. Consequently, an up-counter may be converted into a down counter by simply inverting the output bits into the carry chain.

This requires two modifications to the up counter. First, all inputs to the propagate cells must be inverted, as shown in Figure 6. Second, the counter cell must be modified so that the direct path from the even bit to the odd bit becomes inverting, as shown in Figure 7. In all other respects, the counter remains the same. Performance and expandability are unaffected.

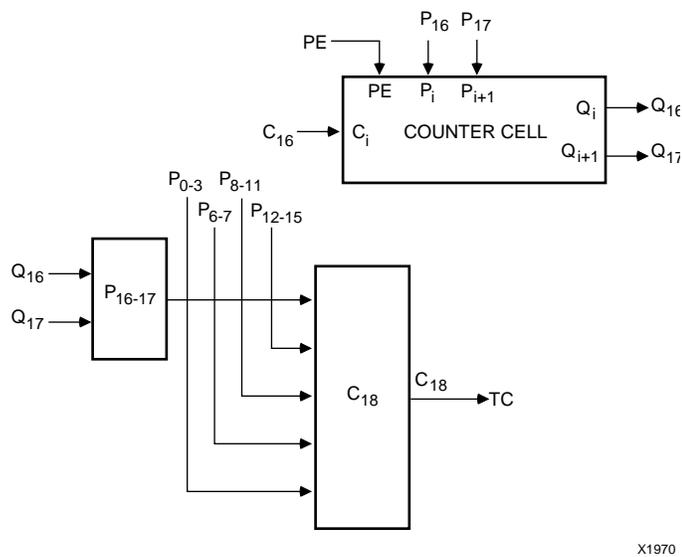


Figure 5. Extension to 18 Bits

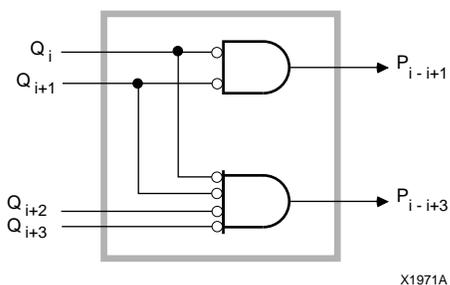


Figure 6. Down-counter Propagate Cell

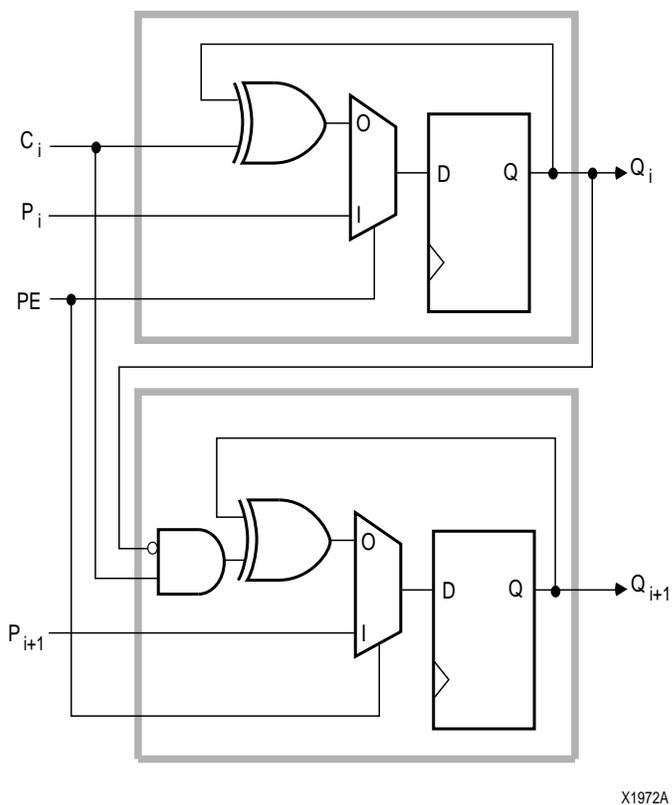


Figure 7. 2-Bit Down-counter Cell

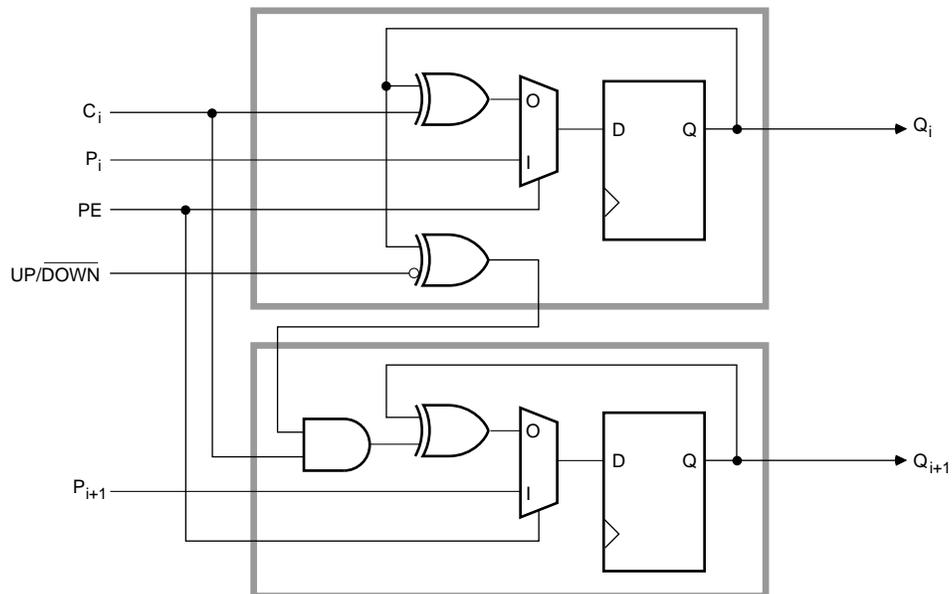
Loadable Binary Up/Down Counter

To create an up/down counter, simply make the above inversions programmable. For the counter bits, this is not a problem. An XOR gate is placed in the direct path, as shown in Figure 8.

The propagate cells are more of a problem. The 2- and 4-input functions become 3- and 5-input when the up/down control is added; they can no longer share a single CLB.

The propagate cells must be split in two CLBs each, and the 3-input functions combined if necessary. Two or four additional CLBs are required, and additional routing delays might be created due to the higher fan-outs and the longer signal paths among the greater number of CLBs.

This design results in 16-bit up/down counters that operate at 46 MHz, and 32-bit up/down counters that operate at more than 37 MHz



X1973

Figure 8. 2-Bit Up/Down-counter Cell