

Summary

This Application Note provides guidance for implementing high performance multiplexers and barrel shifters in XC3000 LCA devices.

Xilinx Family

XC3000A/XC3100A

Introduction

Since the function generator in the XC3000 series CLB has only five inputs, it cannot directly implement a four-input multiplexer, which requires four data inputs and two select inputs. The CLB does, however, have the logic capability to implement a 4-input multiplexer.

This applications shows how to access the full logic capability of the CLB for 4-input multiplexers. It also shows how best to implement larger multiplexers and barrel shifters.

Multiplexers

Four-Input Multiplexer

CLB function generators have a base-FGM operating mode that permits certain functions of more than five variables to be implemented. The restriction on the function is that it must be implementable as a multiplexer selecting between two functions, each of four variables. Clearly, a 4-input multiplexer meets this requirement; each 4-input function implements a 2-input multiplexer, and the final multiplexer selects one of the outputs.

Since the CLB only has five logic inputs to the function generators, the sixth input to the multiplexer must reach the function generators via the CLB .di pin, a flip-flop and the internal feedback path. Routing through a flip-flop has obvious timing implications, but using this path can result in through delay and resource savings of 50%. Often the additional select delay can easily be accommodated, and sometimes it even saves storage resources elsewhere.

One approach is to pipeline the select lines, Figure 1. Two bits of the 4-input multiplexer are implemented in two CLBs. In one CLB, the S_0 select line is registered, while in the other the S_1 select line is registered. In addition to being used within the CLB, the registered versions are

output for use in the other CLB. This balances the delay in the select lines. Notice that the order of the multiplexer ranks is reversed in the two CLBs.

Alternatively, if the design requires one of the multiplexer inputs to be pipelined, this input may use the flip-flop route, thus saving an external pipeline register, Figure 2. In either case, one CLB flip-flop remains available for optional use registering the multiplexer output.

Wider Multiplexers

If the multiplexer select line can be pipelined, large multiplexers are best implemented using multiple ranks of the 4-input multiplexer described above, together with a 2-input multiplexer, if required. Even if a completely combinatorial circuit is absolutely necessary, there are better alternatives to using multiple ranks of 2-input multiplexers.

While 4-input multiplexers cannot be implemented in a single CLB, it is possible to implement a 3-input multiplexer in one CLB. If this 3-input multiplexer is considered part of a 4-input multiplexer that is completed elsewhere, it can be used in expansion schemes, and binary encoding of the select lines can be retained.

The 8-input multiplexer, Figure 3, uses two 3-input multiplexers and a 2-input multiplexer to select one bit from six; on the two outstanding select codes, Zeroes are selected. These two select codes are also used to AND the corresponding inputs into a 2-input multiplexer. The output of this multiplexer is Zero whenever one of the other six select codes is asserted, and consequently, it is only necessary to OR the two outputs to complete the multiplexer.

This structure requires four CLBs, as does the 2-input multiplexer approach. However, the delay is only two CLBs instead of three, a reduction of 33%.

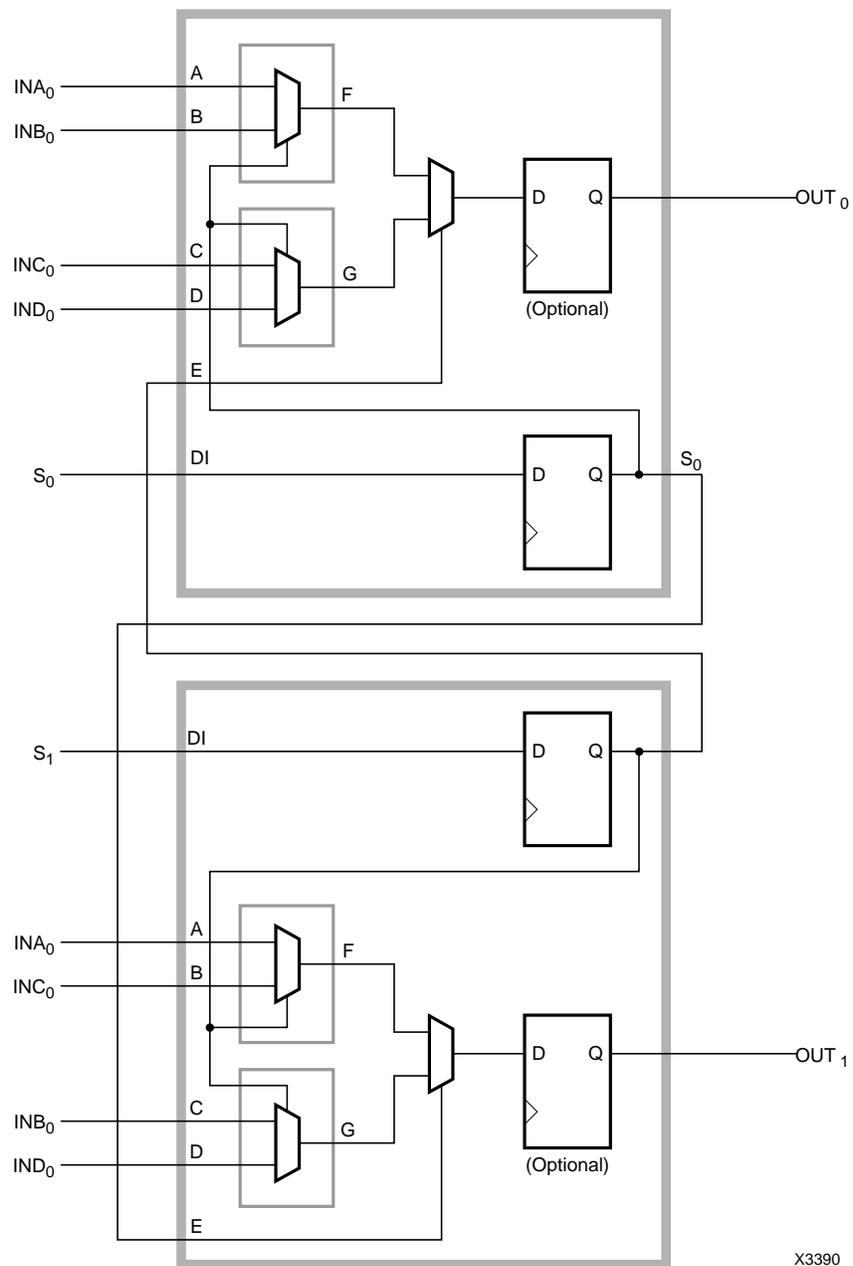


Figure 1. Dual 4:1 Multiplexer with Pipelined Select (Two CLBs)

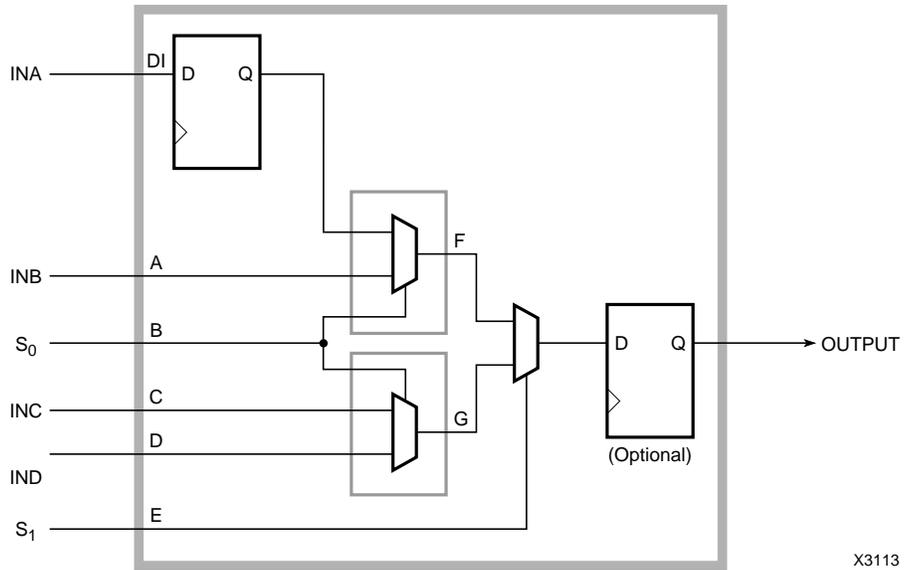


Figure 2.4:1 Multiplexer with Pipelined Input

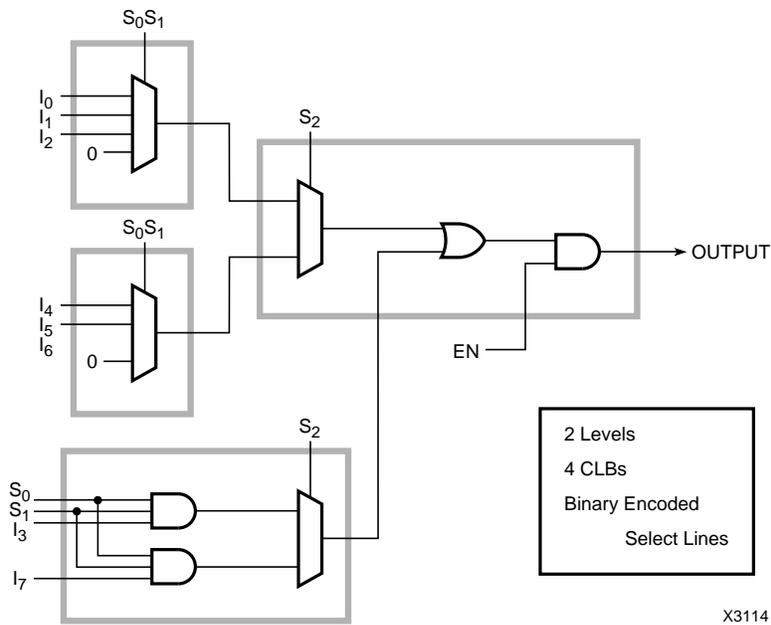


Figure 3.8:1 Multiplexer

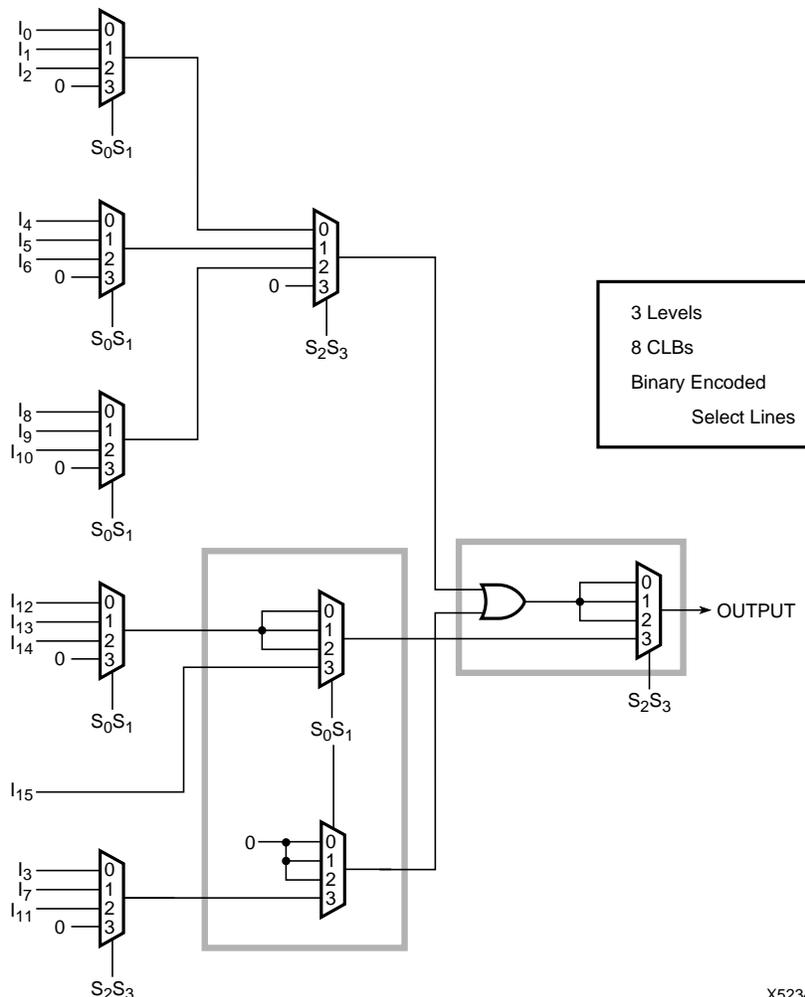


Figure 4. 16:1 Multiplexer

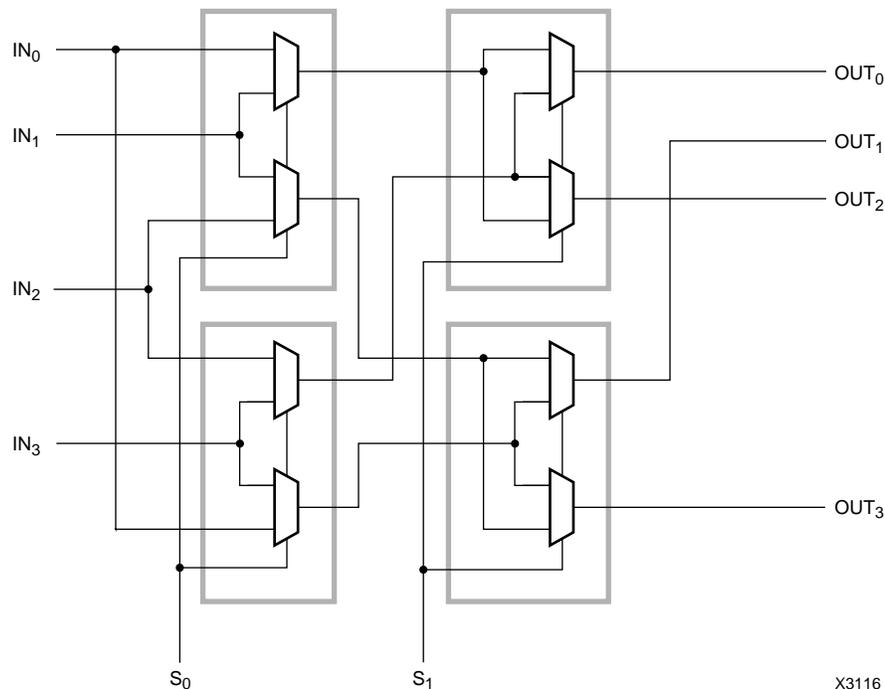
An output enable control is provided that permits the multiplexer to be expanded by ORing the outputs in an additional level of logic. A single CLB can implement a 5-input OR gate. Consequently, this expansion scheme can accommodate up to 40-input multiplexers within three levels of CLBs. The more significant select lines must be decoded to provide individual enables to each 8-input multiplexer, but this logic settles in parallel with the first level of CLBs.

For 16-input multiplexers, the design shown in Figure 4 may be used. It requires eight CLBs in three levels, which is one CLB fewer than is needed to combine two 8-input multiplexers, and one less level of CLB than a design based on 2-input multiplexers.

Barrel Shifters

A four-input barrel shifter has four data inputs, four data outputs and two control inputs that specify rotation by 0, 1, 2 or 3 positions. A simple approach would use four 4-input multiplexers, since each output can receive data from any input. This approach yields the best solution only if the select lines can be pipelined, and the 4-input multiplexer design described above is used. The complete barrel shifter can be implemented in one level of four CLBs.

If the barrel shifter must be fully combinatorial, it is better to decompose the barrel shifter into 2-stages, Figure 5. The first stage rotates the data by 0 or 1 positions, and


Figure 5. 4-Bit Barrel Shifter

the second rotates the result by 0 or 2 positions. Together, these two shifters provide the desired rotations of 0, 1, 2 or 3 positions. As in the previous design, four CLBs are required, but the number of levels increases to two. A combinatorial 4-input multiplexer approach would have used six CLBs in two levels.

This binary decomposition scheme can be used for any number of bits. The number of levels required for an N -bit shifter is $\log_2 N$, rounded to the next higher number if N is not a power of two. Each level requires $N/2$ CLBs. The first level rotates 0 or 1 positions, and subsequent levels each rotate by twice as many positions as the preceding level. The select bits to each level form a binary-encoded shift control.

For example, an 8-bit barrel shifter can be implemented in three levels of 2-input multiplexers that rotate by 1, 2 and 4 positions. Each level requires four CLBs, for a total of

12. For a 12-input barrel shifter, four levels of multiplexer are required. These multiplexers rotate by 1, 2, 4 and 8 positions, and require a total of 24 CLBs.

The 16-bit barrel shifter shown in Figure 6 has only two levels of CLB, and is, therefore, twice as fast as one using the 2-input multiplexer approach. However, the shift control must be pipelined, since it uses the 4-input multiplexer shown in Figure 1. The first level of multiplexers rotates by 0, 1, 2 or 3 positions, and the second by 0, 4, 8 or 12 positions. Each level requires 16 CLBs, and the total of 32 is the same as for the 2-input approach. The shift control remains binary.

Again, this scheme can be expanded to any number of bits using $\log_4 N$ rotators that successively rotate by four times as many bit positions. For sizes that are odd powers of two, the final level should consist of less costly 2-input multiplexers.

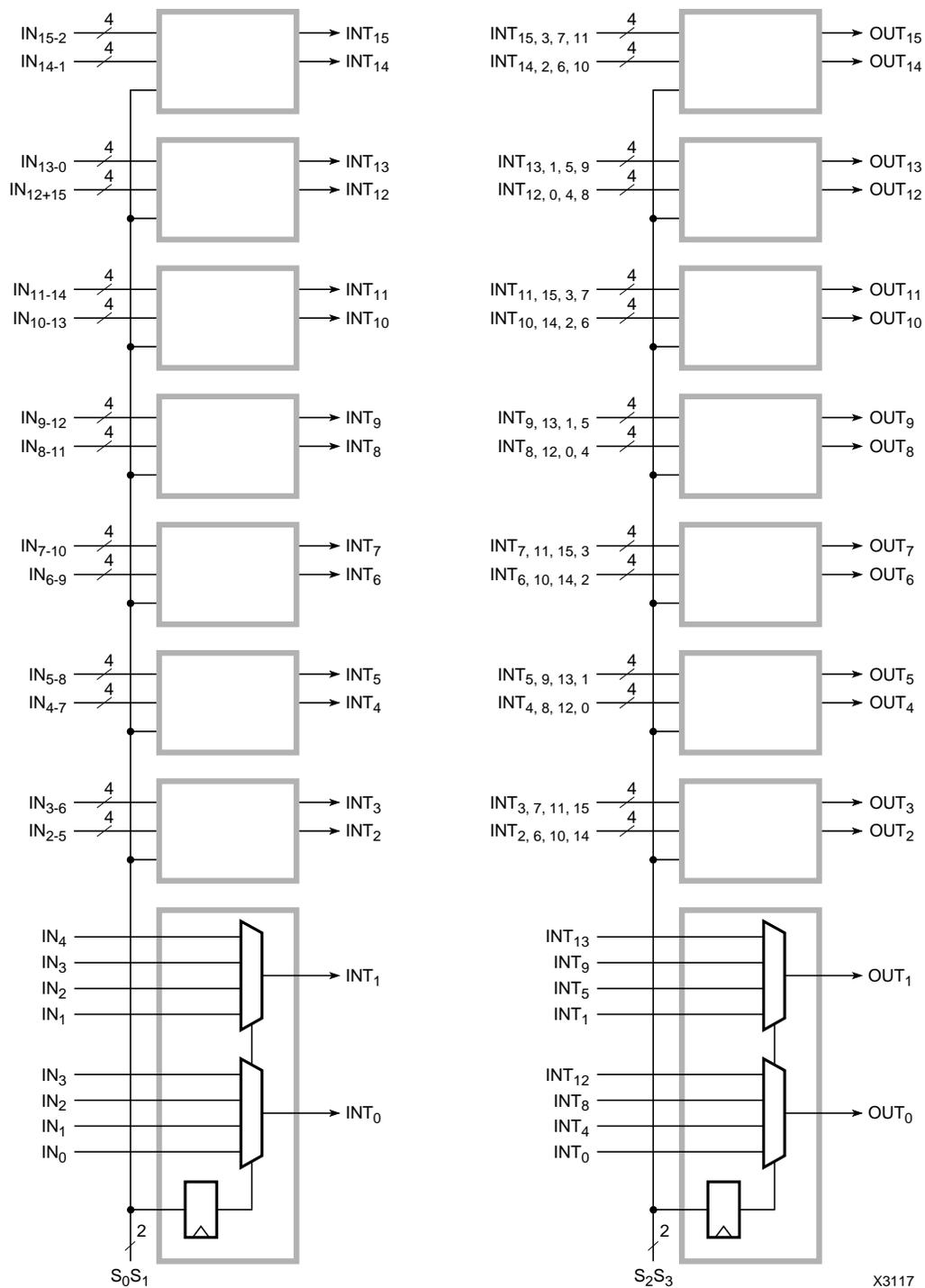


Figure 6. 16-Bit Barrel Shifter