Demultiplexing 200 MHz Data Streams

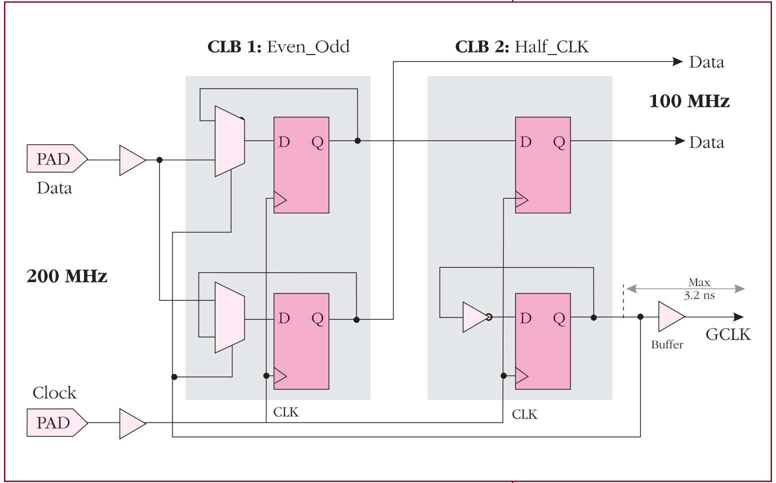
Modern serial data protocols (e.g., FireWire, SONET, ATM, T4) sometimes require clocks that are faster than maximum FPGA global clock speeds. To solve this problem, the incoming clock (200 MHz in the example below) can be used to demultiplex the incoming single data stream into two parallel data streams clocked at one-half the speed (100 MHz here).

The circuit diagrammed below avoids the set-up time of the XC3100A-09 FPGA's IOB input flip-flop and the delay of the global clock distribution network. Incoming data and clock are both routed to nearby CLBs. The delay on the data path from pad to CLB function generator is 2.1 ns (worst case), while the delay on the clock path from pad to CLB input is 2.3 ns. The CLB register's set-up time (function generator input to CLB clock input) is 1.5 ns. The longest possible input pad-to-CLB register set-up time is 2.1 + 1.5 - (0.7 x 2.3) = 2.0 ns. (This assumes 70% delay tracking; i.e., if one delay is at its worst-case value, the other delay cannot be shorter than 70% of its maximum value.)

The possibility of a pad-to-register hold time requirement is assessed by evaluating the opposite situation: (0.7 x 3.6) - 2.3 = 0.2 ns. Since the result is a positive value, the pad-to-register set-up time will always be positive, so there is no hold time requirement.

The circuit also can be cascaded to demultiplex the incoming data stream into four parallel streams (for example, for storage in the slightly slower XC4000E distributed RAM). Such a 1-to-4 demultiplexing structure uses just six CLBs.

The corresponding output multiplexing is more difficult, since the delay between clock input-to-output pad and clock-to-out in the XC3100A-09 is greater than 5 ns. An external ECL clocked multiplexer may be the most practical solution. ◆



CLB 1: Even_Odd
CLB 2: Half_CLK
Data
100 MHz
Data
PAD
Data
D Q
D Q
200 MHz
D Q
D Q
Max
3.2 ns
Buffer
GCLK
Clock
PAD
CLK
CLK