You can implement a digital UWB (ultrawideband) pulse generator in most FPGAs. The design lets you create a pulsed signal with a frequency that’s twice the FPGA’s clock frequency (Figure 1). A previous design relies on asynchronous delays to make pulses of the desired frequency. That design, however, requires an FPGA that supports tristate pullups, such as the Xilinx Virtex 2 (Reference 1). That approach also requires manual placement and routing. Today’s FPGAs don’t support tristate pullups. In addition, asynchronous delays vary with temperature. This Design Idea uses a synchronous-delay approach employing a combination of multiple clock phases. You can implement this design in all types of FPGAs.

The maximum clock frequency of the DCM (digital clock manager) and the flip-flops are the main limiting factors in this design. For example, the DCM of a Xilinx Virtex 4 can’t exceed 400 MHz. An FPGA can generate signals of frequencies that are half the clock frequency because it takes two clock pulses to toggle the signal from zero to one and back. Thus, you can’t directly generate frequencies greater than half the clock frequency. This design lets you generate pulsed signals higher than half the clock frequency and reaches twice the clock frequency by using multiple clock phases from the DCM and synchronous delays smaller than one clock period.

Figure 2 shows the proposed pulse generator. It consists of three functional blocks, an OOK (on/off-key) modulator, a synchronous-delay generator, and an edge combiner comprising an exclusive OR gate. The OOK modulator comprises an inverter that the pulse-repetition-frequency signal triggers at every start of a new pulse. When a trigger occurs, the OOK circuit inverts a preinitialized signal to a time equal to a count value derived for a pulse width and remains zero until the next trigger. The OOK block generates a frequency that is one-half the clock frequency. This OOK output passes through the synchronous-delay generator, which generates three delayed versions of the OOK output.
These delays are smaller than one clock period. The clock phases in turn clock flip-flops $\text{FF}_1$, $\text{FF}_2$, and $\text{FF}_3$, which lag by 90, 180, and 270°, respectively. These delayed pulses then combine with the output of the OOK modulator using combinatorial logic to generate the desired frequency for the UWB pulses. The edge combiner performs an XOR (exclusive-OR) operation, which generates signal frequencies that depend on the edges you want to combine. Combining the output of the OOK edge with the output of $\text{FF}_1$ generates a signal frequency equal to the clock frequency. Combining the edges of all outputs generates a signal frequency equal to two times the clock frequency. The DCM synchronizes these delays, producing an accurate signal frequency. This design is less complex than the asynchronous-delay approach in Reference 1.

Reference