Frequency Synthesis

FPGAs can simplify frequency synthesis without adding complexity, quite often improving performance. Here are some useful techniques.

by Austin Lesea,
Principal Engineer, Xilinx,
austin.lesea@xilinx.com

Frequency synthesis is often used in digital designs, and three major methods are used: dividers, direct digital frequency synthesis (DDFS), and fractional-N synthesis. Sometimes these techniques are combined with traditional analog phase locked loops or other analog elements to remove synthesis by-products such as unwanted side-bands or jitter.

Dividers

Taking an input clock frequency F, and dividing by any modulo of 2, such as 2, 4, 8, and so on, is a common technique. The one problem is that you must choose F in advance such that all of the other frequencies are sub multiples using the modulo 2 factors. There are circuits that rely on asynchronous delays to provide divide by 3, 5, and so on, but their outputs are not square, so they create a large harmonic content. These techniques are found most commonly in digital circuit design today. I will not elaborate further on these, because the following two techniques are more useful.

Direct Digital Frequency Synthesis

One technique that is not usually considered, due to its complexity, is the direct digital frequency synthesizer (DDFS). Here, a constant N is placed on one port of an adder, and the other port of the adder is fed back from a D-type latch whose input is connected to the output of the adder. At every clock of the latch, an incremental phase is added to the previous result. The most significant bit of the latch will transition at a frequency determined by the equation:

$$\text{Fout} = \frac{\text{Fclock} \cdot \text{N}}{2^{\text{K}}}$$

Where K is the number of bits of the adder and latch.

One of the interesting things that is noticed right away is that any arbitrary frequency may be generated to within a resolution of 2^{-K}. For example, with a 48 bit accumulator and latch, the resolution is $3.5*10^{-15}$, or about a thousand times better that the accuracy or resolution of a cesium clock.

So, where do you get a 48-bit latch and accumulator? In an FPGA there is usually plenty of room for not just one, but perhaps three or four DDFS functions. Not all of them have to be 48 bits, they just need to be long enough to synthesize the frequency that is desired. In fact, the Xilinx Foundation Series and Alliance Series tools provide the LogiBLOX feature which will create an adder/accumulator, of any length, with only a few keystrokes.

There are two limitations of this technique: the output frequency must be less than 1/2 the clock frequency, and the output frequency will jitter by the period of the clock frequency. To remove the jitter, you could send the output through a band pass filter, or lock a separate phase locked loop, or complete the synthesis by taking the 12 or 14 most significant bits of the latch, and using them to address a sine wave lookup table ROM. The output of the ROM then goes to a digital to analog (D/A) converter, whose analog output is then a sine wave with much less jitter. The clock jitter will still be present however, and may still require more filtering.

Fractional-N Synthesis

In fractional-N synthesis, counters are used to provide a variable modulus, or counters are used to provide for “pulse swallowing.” If you have a counter that can divide by 10 and divide by 11, then you can make a circuit where the counter divides by one factor for some number of clock cycles and then divides by the other number for perhaps a different number of clocks. The resultant frequency is then the ratio of how many clocks were divided by 10 to how many were

Techniques

The advantages of using FPGAs for frequency synthesis are many: no dependence on voltage, temperature, or aging; and no external analog components.

divided by 11. This technique requires a variable modulo counter, another counter for the selection, and logic for state control.

One disadvantage of this technique is that the jitter is increased due to the difference in the modulo, with the output frequency being 10 times less, then abruptly 11 times less, and so on, in a fine time scale. The advantage is that it is considered a simpler structure (but only from the old point of view using discrete, small scale integrated circuits).

The other commonly used method of fractional-N synthesis is to drop an output clock pulse based on some rule derived from a counter state machine. For example, if you drop every 4th clock out of every 5 input clocks, the output rate is 4/5 the input rate. This is known as "pulse swallowing". This also results in a large amount of jitter due to the missing clock period.

Jitter Reduction

In all of the above techniques where the output is a digital signal transitioning at the desired rate, one can reduce the jitter by dividing the signal down. The jitter time will be less as a function of the resulting period of the signal. For example, if you want a 1.536MHz signal, and you have a 16.384MHz clock, you could synthesize a 6.144MHz signal to an arbitrary resolution, and then divide it by four. The resulting signal would have a jitter of 2.3%. This is an acceptable jitter for any data communications related application, such as T1 Frame Relay Customer Service Unit/Data Service Unit (CSU/DSU).

As mentioned before, using a phase locked loop also reduces the unwanted jitter, where the filtering of the loop is used to attenuate the unwanted side band energy which is the cause of the jitter. One of the simplest means of implementing this, is to use a voltage controlled crystal oscillator whose output is exclusive OR'd with the output of the synthesizer and whose input is the RC filtered XOR output.

Using an FPGA

Even the seemingly complex DDFS is simpler to implement in an FPGA than fractional synthesis because the DDFS circuit consists of replicated units, all interconnected simply. You don't need to find a fractional solution, and realize the state machines and modulo counters. In fact, the DDFS is the most versatile as it may be changed at any time by placing a new N at the adder. Calculating the N may get a bit difficult past 32 bits, because computer math programs are often unable to deal with the resolution and display of hexadecimal or binary numbers beyond this point. I use MathCad ™ from MathSoft, Inc. and split the calculation up into two parts to get arbitrary resolution.

The cost of implementation is the adder and the D-type flip-flop for every bit of synthesis. If examined in this fashion, the cost of a DDFS is more than the other methods. In large FPGAs, the philosophy might be best stated as "gates are free" as long as the design fits into the intended part. It takes 26 CLBs to make a 48 bit DDFS in the XC4000 family. Even for the smallest part, this is about a quarter of the total CLB count (XC4003). For larger parts, such as the XC4085XL, it is hardly a consideration.

Conclusion

The advantages of using FPGAs for frequency synthesis are many: no dependence on voltage, temperature, or aging; and no external analog components.