Using FPGAs for Digital PLL Applications

In addition to purely digital applications, many designs use Field Programmable Gate Arrays (FPGAs) for DSP. We'll examine one such application, digital PLLs, to show various ways of implementing PLL designs using FPGAs.

Pulse Steal PLL

In telecommunications applications, it is often desirable to generate a digital signal that is locked to an incoming signal and is some multiple of its frequency. A drawing of a pulse steal PLL, which is a simple way to generate such a signal, is shown in Figure 1. Note that the design contains an ordinary oscillator but no VCO. Except for the crystal, the entire design will operate in an FPGA.

Note the frequency relationship that holds at points A and B in the figure, where

$$\text{OSC}/(K*M) = \text{Input}/N = \text{Comparison Frequency} \qquad (1)$$

The technique is based on selecting a reference oscillator frequency slightly higher than OSC. This frequency (OSC+) should be chosen so that

$$1/\text{Comparison Freq.} - (K*M)/(\text{OSC+}) = .5 * (1/\text{OSC}) \qquad (2)$$

The right side of equation 2 equals one-half the period of the reference oscillator.

The reference oscillator frequency delta will cause point B (the detector flip-flop D input) to begin to precede point A (the detector flip-flop clock input) by half a period. When the edge of the D input is sufficient, the detector will clock true and begin a pulse train through the two deglitching flip-flops. The output of the second of these clears all three flip-flops and steals a pulse by disabling the divide by K output. Stealing the pulse puts point B behind A until the reference oscillator delta can move it ahead by one period, repeating the cycle. Points A and B are always within one-half a cycle of each other.

The circuit allows the frequency of the output signal to be selected simply by adjusting the values of the dividers K and M. The lock range of the loop is given by the following:

$$\text{Lock Range} = \pm\ (\text{OSC+}/\text{osc})/\text{Input} \qquad (3)$$

Jitter-Bounded Digital PLL

Another technique[1] for generating a wide variety of synchronized clock frequencies with low jitter employs an accumulator Digital Controlled Oscillator (DCO) and phase and frequency comparators. The system, shown in Figure 2, can lock to any division of a reference frequency (F ref.) as selected by the data loaded into frequency divider counters.

The Successive Approximation Register (SAR) and its controller serve as a low-pass filter supplying the DCO with frequency and phase correction data. Among the three inputs to the SAR controller is the F ref. divided by a factor Q to form F q.

The other two inputs come from the phase and frequency (zero) comparators. The frequency comparator output is the DCO frequency divided by P to form F p. When the system is in lock, the following equation is true:

$$\text{F dco} = (P/Q) * \text{F ref.} \qquad (4)$$

The heart of the system is the accumulator DCO, which determines the ability to lock to a frequency and the amount of jitter allowed. The DCO consists of a four-bit accumulator whose input is fed by the SAR. The DCO input value is determined from the phase and frequency comparison feedback loops. The most significant bit of the accumulator output is the DCO output signal. It is generated by successively adding the SAR value to itself at the high-frequency system clock rate. The frequency comparator uses the value of P to divide the DCO frequency. If the frequency is out of lock during a period of F ref., the comparator asserts greater-than-zero or less-than-zero to the SAR controller to modify the value of the register. If the P counter output is zero, the DCO has the correct frequency.

The DCO latch acts as a phase register indicating the phase of the DCO with respect to F ref. The DCO phase is calculated by the N most significant accumulator output bits. When the DCO is out of phase, the jitter, or phase difference, is detected by the phase comparator and accumulates with time until it equals one period. The feedback loops then cause the SAR register controller to load a correcting value into the register or to clear the accumulator with a synchronizing pulse.

The Jitter-Bounded DPLL may be implemented entirely on an Actel FPGA. The resource requirements vary with the relationships of the system input and output frequencies, but for any F ref., system clock, and desired output frequency, the design is easily accommodated on an Actel FPGA.

References:

1. S. Walters and T. Troudet, "Digital Phase-Locked Loop with Jitter Bounded," *IEEE Transactions on Circuits and Systems*, Vol 36, No. 7, July 1989.

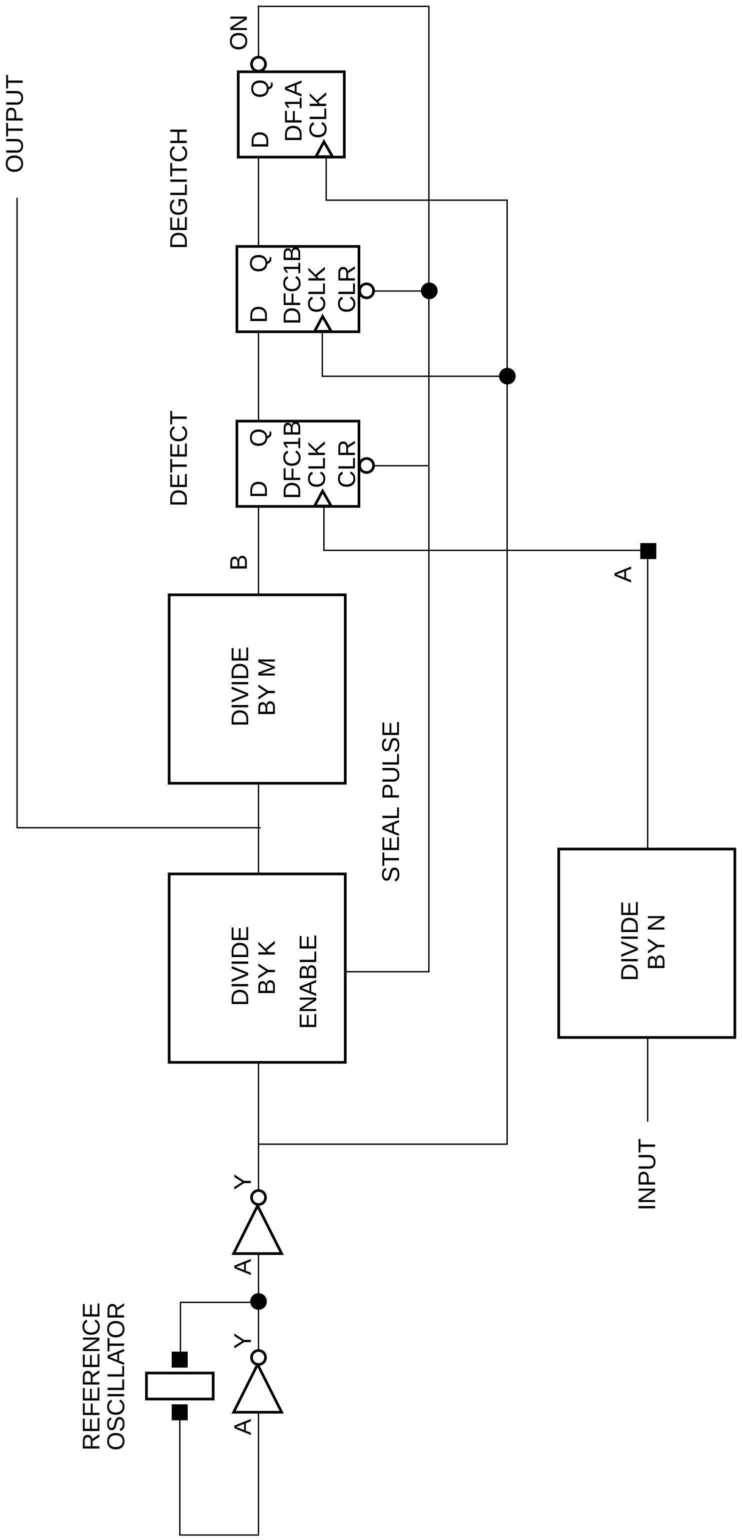

Figure 1 • *Pulse Steal PLL*

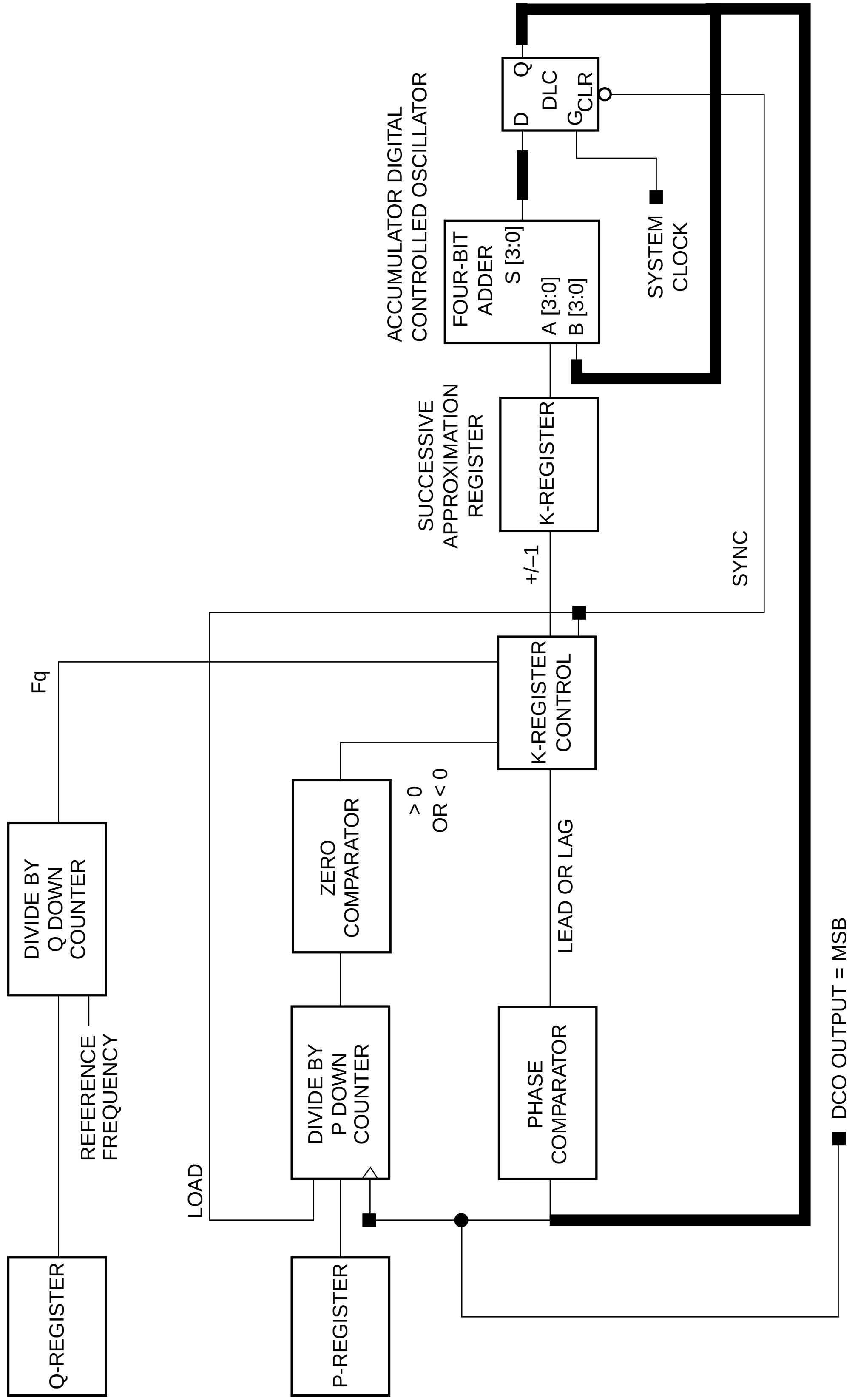

Figure 2 • *Jitter-Bounded Digital PLL*