Wide Range, Essentially Linear Control Circuit for Control of the Reference Frequency in Digital Phase-Locked Loops

T. O. Anderson
Communications Systems Research Section

A simple all-digital control circuit intended for control of the frequency of the reference signal in all-digital phase-locked loops is presented. Such control circuits, of prior art, are often nonlinear and provide only for narrow frequency range. The present control circuit, though very simple, provides for essentially linear control for a wide frequency range. The circuit presented in this article is intended for use in all-digital phase-locked loops and described in this context.

I. Introduction

A phase-locked loop implemented entirely with digital circuits references all its operations to a fixed high-frequency service clock. This clock is derived from a simple crystal oscillator operating at the maximum operating speed of the digital circuits. To keep the step size as small as possible and also of the same general size for correction in either direction, the general principle of the frequency correction mechanism is that of deleting every nth pulse in the pulse train from the oscillator to an output divider for the steady state and $(n - m)$ and $(n + m)$, respectively, for correction in either direction.

The output from a simple counter/divider which can be programmed to divide by one of a set of integers controls a pulse delete gate resulting in the instantaneous frequency of the reference signal: $f_n = [(n - 1)/n]\, f_0$, where f_0 corresponds to the uncontrolled frequency of the reference signal. A change of n to the next larger or next smaller integer does not result in the same frequency change, i.e., for small n's the changes are large and for large n's the changes are small. For $(n < 10)$ the frequency changes are too large and too nonlinear to be usable in a control system. For these reasons n must be limited to $(n > 10)$, which in turn limits the acquisition range to $\sim 5\%$ about the center frequency.

In Ref. 1 an entirely digital implementation of a bit synchronizer for noiseless binary data is described. This synchronizer includes one search and one track mode, both using the control circuit indicated above. For the search operation n is varied from 10 to 20 and for track operation from 20 to 30. For many purposes the search range of only $\pm 5\%$ may, however, be too narrow.

This article presents a simple control circuit which makes the control of the frequency of the reference signal essentially linear and extends the acquisition range from 10 to 50%.

The design strategy is explained first as it was conceived for the simplest possible implementation. The new control circuit is then discussed in reference both to its block diagram and detail logic diagram.

II. Design Strategy

Deleting one pulse every nth pulse requires that n be an integer. For smallest possible control steps n must then include all possible integers $\geqslant 2$. The implementation of $f_n = [(n-1)/n]\, f_0$, which is shown as plot B in Fig. 1, contains a programmable counter/divider which can be programmed to divide by $2,3,4,5,\cdots,n$. The output of this counter controls a pulse delete gate in the pulse train from the oscillator to the output divider.

Another simple counter is a binary ripple counter which divides by $2,4,8,16,32,\cdots,2^n$. Selection of any one stage of such a counter for control of a pulse delete gate results in function $f_n = (2^n - 1/2^n) f_0$. This function, although simple to implement, is by itself essentially unusable as a control function. For reference see plot A, Fig. 1.

However, if one substracts the difference $A - B$ from B, the result, shown in plot C Fig. 1, becomes a desirable function. For control purposes this function is essentially linear and usable for all n's $(2 \leqq n)$ resulting in a 50% control range.

As will be shown, both the $A - B$ and $B - (A - B)$ functions are very simple to implement, each requiring only a flip-flop (F-F) and a gate.

III. Functional Block Diagram

For references see Fig. 2. The inputs to the divide by n divider and the divide by 2^{n-1} divider are derived from the true and complement output from the oscillator, respectively. This will assure that A and B are of different phase and that for $n = 2$, $A - B = 0$. As is readily apparent from the block diagram (Fig. 2), if the same phase were used for both A and B, $A - B$ would be ambiguous.

IV. Detail Logic Connections

The $A - B$ and $2B - A$ circuits are identical and very simple, as shown in Fig. 3. An F-F is set and reset on the A and B pulses, respectively. If two B pulses occur within the period between two A pulses, an output pulse is registered.

The divide n counter/divider is implemented by simply setting a number of counter F-Fs to the binary equivalent of n. The set operation takes place over set gates and the F-Fs are connected to count down. The output of an all-zero gate will enable the set gates and again set the F-Fs to n. Setting takes place from a zero condition requiring only a single set term for each F-F. For references see logic diagram (Fig. 4). Only four bits are shown in the figure; however, the design is iterative.

The counter/divider which divides by 2^n is in itself trivial. Only with n being the selection number and available in binary notation a base-two to base-ten conversion network is required.

As apparent from the expression $f_n = [(2^n - 1)/2^n]\, f_0$, the frequency change for $(n > 8)$ is negligible. n can then be limited to $(2 < n < 8)$. For $(n < 8)$ the $(n_{(2)} \rightarrow 2^n_{(10)})$ conversion network simply reduces to a readily available binary-to-decimal decoder. Two design versions of the divide by 2^n counter programmable by n in binary notation and where $(n < 8)$ are considered. The first is identical to the divide by n counter, the only difference being that the $(n_{(2)} \rightarrow 2^n_{(10)})$ decoder is inserted between the selection terms and the set gates as shown in Fig. 5.

In the second version shown in Fig. 6, the output from the $(n_{(2)} \rightarrow 2^n_{(10)})$ converter selects the output from any one stage of the counter as the final output. With the output from any one F-F being a square wave, a circuit is required to form a one clock period pulse of that periodicity.

For $(n > 8)$ the divide by 2^n counter/divider or its effect on the system should be inhibited. The inhibit term may be the output from an OR-function whose inputs are all the $(2^n_{(10)})$ terms for $(n > 3)$.

Reference

1. "Digital Data Transition Tracking Bit Synchronizers for Noiseless Binary Waveforms," *Communications Designer's Digest*, p. 12, Nov.–Dec. 1971.

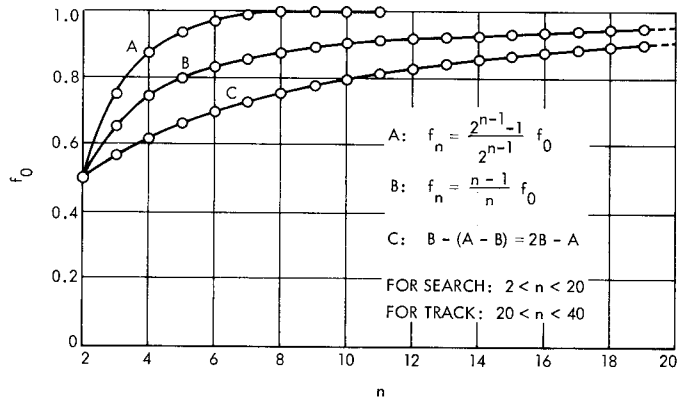

Fig. 1. Control function graphs

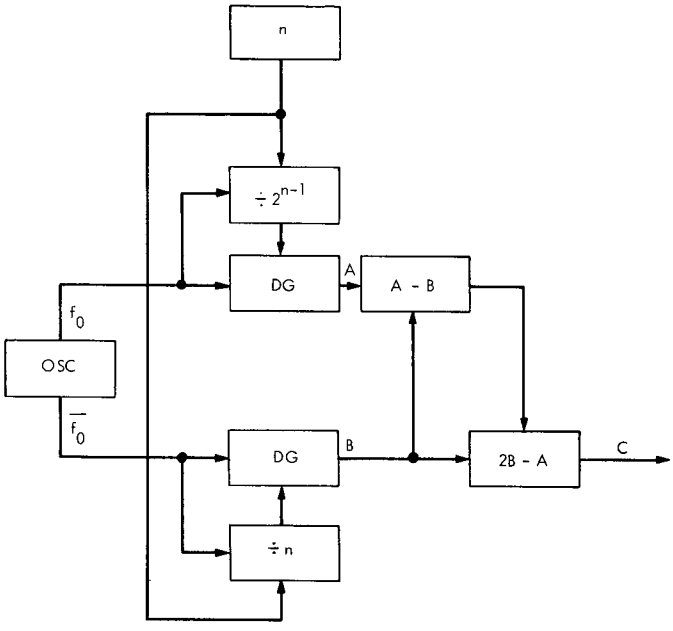

Fig. 2. Functional block diagram for implementing the 2B — A function

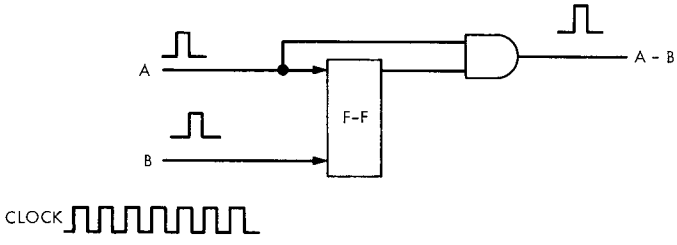

Fig. 3. Detail logic diagram showing an A — B logic

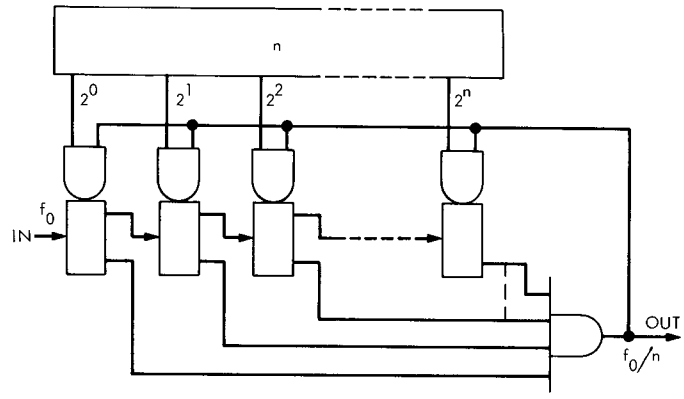

Fig. 4. Logic diagram of a programmable divide by *n* counter

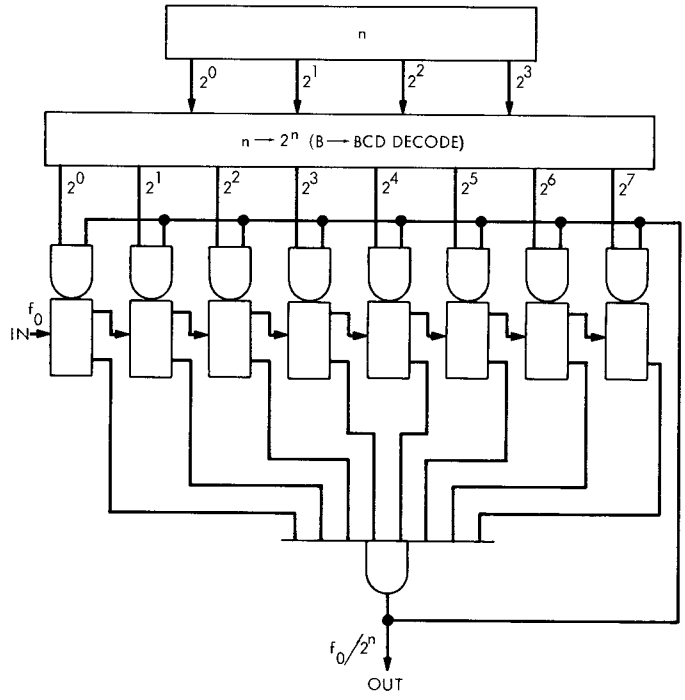

Fig. 5. Logic diagram of a programmable divide by 2^n counter with parallel set gates

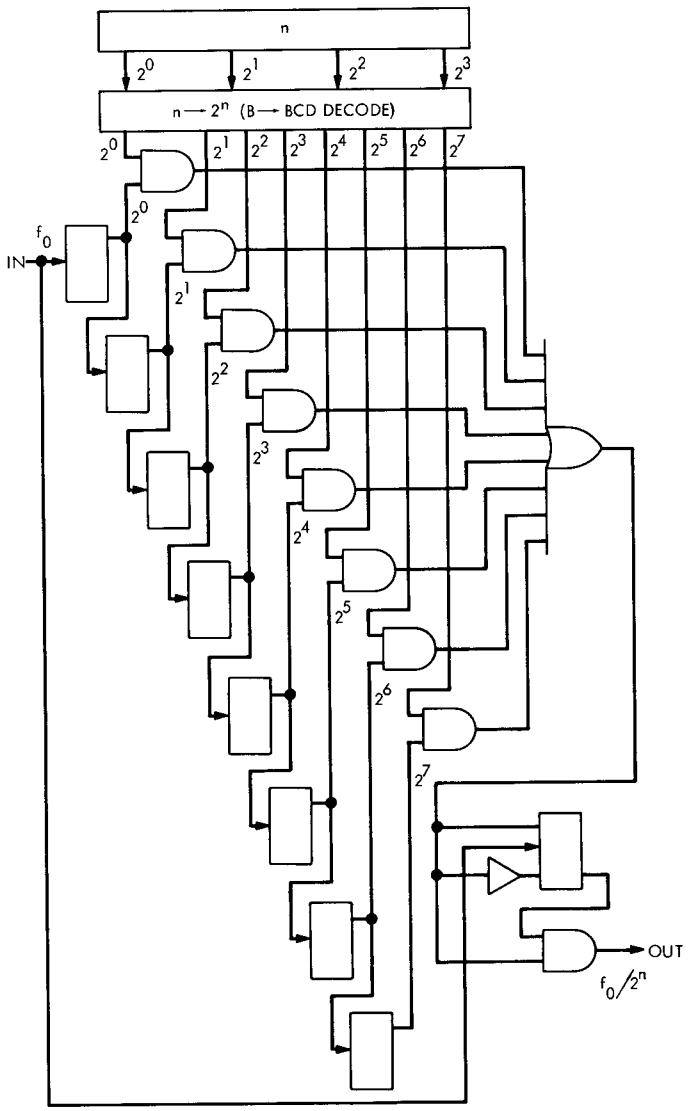

Fig. 6. Logic diagram of programmable divide by 2^n counter with output stage selection