BIT SYNCHRONIZERS FOR PSK AND THEIR DIGITAL IMPLEMENTATION

Jack K. Holmes
Holmes Associates, Inc.
1338 Comstock Avenue
Los Angeles, California 90024

ABSTRACT

Bit synchronizers play an important role in phase-shift-keyed systems (as well as noncoherent systems) with the trend towards all-digital versions. This paper discusses the various types of bit synchronizers and the additional functions and subsystems that must be used to make them efficient at low values of SNR and bit-transition density. It also discusses the digitization of bit synchronizers, along with the performance measures commonly used.

INTRODUCTION

All communication systems require the use of bit synchronizers to provide a clock for the detected data. Perhaps, the most common high-quality bit synchronizer is the data-transition tracking loop (DTTL) which is based on on the work of Lindsey, Tausworthe, Hurd and Anderson [1,2]. Harmonic-generating devices such as filter and squarers, followed by a phase-locked-loop (PLL) provide a simple bit synchronizer with relatively good performance. The first section of the paper discusses numerous different bit synchronization loops.

TYPES OF BIT SYNCHRONIZERS

First, we discuss the various types of bit synchronizers; they can be divided into two classes. The first is a synchronizer which uses some type of nonlinearity (square law) and filtering to generate a line spectrum at the data rate and is then followed by an ordinary PLL or bandpass filter. In the second type, the bit-timing error is detected directly and that, in turn, controls--through a loop filter--a closed-loop error control. The two types of bit synchronizers are illustrated in Figure 1.

In Figure 2, a group of nonlinear clock-generating bit synchronizers are illustrated. We consider as an input to the bit synchronizers a baseband data stream embedded in thermal noise of the form

$$y(t) = P\, d(t) + n(t) \tag{1}$$

where P is the signal power, d(t) is the baseband NRZ or Manchester data stream, and n(t) is white Gaussian noise. In Figure 2(a), the filtered baseband signal plus noise is delayed one-half of a bit time and multiplied by itself to generate a line spectral component at the data rate that can be tracked by a PLL. Figure 2(b) shows another method of generating a data-rate component. A lowpass filter (LPF) distorts y(t), which is differentiated (highpass filter), then squared to produce pulses at the data rate that yields a spectral component at that data rate. In Figure 2(c), the roles of the squarer and differentiator are interchanged. In Figure 2(d), the delayed signal plus noise is multiplied by a differentiated version of the signal plus noise to yield a clock component at the data rate which can be tracked by the PLL located at the output. Finally, in Figure 5(e), an LPF distorts the signal plus noise and is then nonlinearly modified by the square-law detector and followed by a PLL.

Holmes[5,6] has analyzed the synchronizer in Figure 2(e) and shown that it has relatively poor performance with a one-pole RC LPF located before the squarer. The optimum bandwidth of the one-pole filter was found to be about 3/16 of the data rate. Subsequently, McCallister and Simon[7] showed that, with a one-pole predetection filter, the synchronizer of Figure 5(a) was somewhat more efficient in data tracking. Later in the year, however, McCallister [8] determined that, when the LPF preceding the squarer is a matched filter, the filter-and-square synchronizer shown in Figure 2(e) is superior to the cross-symbol synchronizer shown in Figure 2(a). He further showed that the performance was about on a par with the DTTL that is commonly employed for high-qualtiy bit synchronizers. It is to be noted that the filter-and-square synchronizer is quite simple and, hence, is a desirable synchronizer.

Now let us consider some error-tracking bit synchronizers. First we consider the early/late gate synchronizer illustrated in Figure 3. It operates by integrating over a bit time with two integrators, as shown in Figure 3(b). The early gate has the end of the integration ξT seconds after the bit-transition point, whereas the late gate has the transition ξT seconds after the start of the late gate. When no data transition occurs, the average error signal is zero; however, when a transition does occur within the early/late gate time period, the average error signal is given by[1]

$$\bar{e} = 8\,\varepsilon P(1 - 2\xi)T \quad , \quad |\varepsilon| \leqslant \xi T$$

where ϵ is the timing error between the received signal and the bit-synchronizer loop estimate. Thus, the loop has an error signal which provides the feedback necessary to track the received data stream.

Now consider a very popular type of bit synchronizer--one that has been implemented often--the DTTL (or bit synchronizer), as shown in Figure 4 for NRZ symbols. For Manchester symbols, the loop can be run at twice the rate of an NRZ bit synchronizer and the clock can be divided down by two.

The DTTL utilizes two integrators--one integrates across a bit and one integrates across the point where the loop estimates that a transition point occurs. A bit-transition detector located in the inphase arm serves to cancel any nondata transitions so that noise will not be entered into the loop-tracking function. Also, a plus/minus bit transition and a minus/plus transition are taken account of by this detector, so that the error signal is invariant with a plus/minus or a minus/plus transition. The error-control signal can be shown to be well approximated by

$$e = \sqrt{P}\,\epsilon\,\mathrm{erf}(\sqrt{R}) \tag{2}$$

where P is the baseband signal power, ϵ is the timing error, and

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt.$$

The tracking performance of the DTTL can be minimized by reducing the integration time across the bit-transition point to about 1/2, 1/4 or even 1/8 of a bit time. One trades acquisition time by this reduction, but the timing-error variance is reduced by this design approach. The value of 1/2 is a common compromise value between tracking and acquisition.

ANCILLARY FUNCTIONS REQUIRED FOR EFFICIENT OPERATION

In this section, we discuss ancillary functions used on a bit synchronizer. In particular, we discuss data-bit scramblers, baseline-offset correctors, automatic-gain controls (AGC's), ambiguity or false-lock detectors, and lock detectors.

First, we consider data-bit scramblers; these are used to randomize data entering the bit synchronizer, as illustrated in Figure 5. A self-synchronizing scrambler and descrambler are illustrated in Figure 6. The unscrambled data d_i enters the scrambler and, via the feedback output r_i. produces a scrambled output s_i. This scrambled signal is then

modulated, transmitted , received in the receiver, demodulated, detected, then descrambled.

This self-synchronizing descrambler requires n bits to descramble correctly, where n is the number of shift-register stages in the scrambler or descrambler. Consequently, the first n bits are lost in each transmission. Furthermore, this scrambling function costs a penalty in bit-error rate (BER) performance. A single detection error produces an error in every non-zero tap coefficient c_i i=1,2,...,n so that, if there are K nonzero taps, the bit-error rate is multiplied up by (K+1).

Next consider baseline-offset correction; this refers to the removal of the DC offset present in the baseline signal out of the carrier demodulator. Mathematically, we can describe the received, demodulated signal by

$$y(t) = A \sum_{i=-\infty}^{\infty} d_i \, u(t - iT) + n(t) + b \qquad (3)$$

where A is the signal baseband amplitude, d_i is the ith data-bit polarity, u(t) is an NRZ or Manchester data symbol, n(t) is white Gaussian noise, and b is the demodulator-output bias. Conceptually, for NRZ data, the baseline-offset-correction algorithm adds alternate polarity pairs of bits for M pairs, then divides by 2ATM to obtain an estimate of b which can be subtracted out to remove the bias. In the same manner, Manchester symbols can be added by M bits and divided by ATM since a Manchester symbol will (ideally) have a value of zero DC. This DC offset correction is not necessary for nonlinear (harmonic-generating) synchronizers since it does not affect the PLL error signal.

Now consider the AGC system which functions to provide an approximately constant loop bandwidth and set the loading to the quantizer on digital bit synchronizers. DTTL or ELBS synchronizers often utilize a noncoherent AGC at the input operating on the signal plus noise, whereas harmonic-generating synchronizers can use the CAD in the PLL following the nonlinearity to control an AGC loop in a coherent manner. However, summing the magnitude of the matched-filter outputs and comparing them to a threshold can produce essentially coherent AGC.

Another important function is lock detection, which indicates to the rest of the system that the bit synchronizer is--or is not--still in lock. In a DTTL bit synchronizer, the lock detector for NRZ data is based on samples of the magnitude of an integration over the present bit time, minus the integration over the last half of the previous bit, plus the first half of the present bit. These differences are then summed for a fixed number of samples and the sum is compared to a threshold. Clearly, when in synchronization, the sum is positive and, when out of synchronization, is near zero. For Manchester symbols, the

magnitude of integration over the first half of a symbol has subtracted from it a one-half-bit integration across the midpoint of the bit. Thus, again, when in synchronization, the sum of these differences will be positive. Further, when out of synchronization, the sum of these differences will be positive. Further, when out of synchronization, the differences will tend to sum to zero.

For harmonic-generating bit synchronizers, the CAD output in the PLL following the nonlinearity provides a coherent lock-detector output that needs only to be filtered and compared to a threshold.

Now we consider an ambiguity detector for Manchester data. Since a Manchester symbol has transitions at both the middle and end of a data bit, all bit synchronizers which track Manchester data are susceptible to false lock. That is, either transition point can be locked to by a bit synchronizer. In order to determine the correct position, it is necessary to count transitions at what the loop thinks is the midbit and what it thinks is the end of bit. Since midbit transitions occur 100% of the time (when no half-symbol errors are made) and between-bit transitions typically occur about 50% of the time, counting the two transition points for a fixed time allows for the determination as to which is the midbit point since it will have the higher transition density.

DIGITIZATION OF BIT SYNCHRONIZERS

In order to digitize the signal, it must be filtered and gain controlled, then sampled and analog-to-digital converted. Typical A/D converters used are 4- or 8-bit units, although the actual number of bits needed depends upon the AGC characteristics, the amount of degradation that can be tolerated, etc. The booklike report by D. Martin and D. Secor of TRW discusses A/D converters in considerable detail[13]. Their report shows the optimum loading factor, which is defined as

$$\text{LF} = \frac{\text{rms amplitude of total input}}{\text{zero-to-peak quantizer}} \tag{4}$$

as a function of the number of bits used in the quantizer.

Besides quantizing the bit-sample amplitude, the bits must also be quantized in time. At least eight samples per bit must be taken in order to keep the BER degradation down to around 0.5 dB. Commonly, 16, 32 or even 64 samples per bit are used. In order to employ the higher sample-rate-to-bit-rate ratio requires a wider predetection bandwidth. If we let R_b be the data rate, BW the predetection LPF bandwidth, and M the number of samples per bit, then

$$\text{BW} \cong \frac{M}{2} R_b \tag{5}$$

Hence, the larger number of samples requires a wider bandwidth in order to decrease the degradation.

The data output can be hard quantized when coding is not used or, with coding, three bits usually gives close to ideal performance with Viterbi decoders.

Figure 7 illustrates a typical DTTL digitally implemented bit synchronizer. The data bit stream is filtered and AGC'd, then fed into the A/D converter where each sample is quantized to eight bits. After correction by the baseline-correction circuit, the samples are sent to the matched filter, then to the transition detector to yield either 1, 0, or -1 as its output. In addition to the output from the transition detector, the phase-detector output determines the timing-error estimate when multiplied by the transition detector output. This error signal is then fed to the digital-loop filter which is typically second order, so it provides a direct measure of the filter output plus a term proportional to the sum of all past values. The loop-filter output controls the number-controlled oscillator (NCO) which outputs an analog frequency proportional to the digital value of its input.

PERFORMANCE MEASURES IN BIT SYNCHRONIZERS

The two most important measures of performance in a bit synchronizer are the bit-error rate (BER) and the bit-slip rate (BSR). The BER reflects degradations to ideal by the carrier demodulator, filtering and, of course, the bit synchronizer (assuming no bit slip). Factors which affect the BER degradation of the bit synchronization include the number of bits used, the loop SNR and the amount of filtering done prior to the A/D quantizer. Typical values of degradation can run from 0.5 to 1.5 dB.

The second parameter of prime importance in specifying a bit synchronizer is the BSR, which is the number of bit slips per second. Typically, this number is less than 1×10^{-10}. Holmes [5] recently developed an exact result for the BSR for bit synchronizers in which the S-curve and the power spectral density are known. His results indicate that the BSR is given by

$$BSR = 4\sigma^2 B_L \left\{ \int_0^{1/2} \int_x^{1/2} \eta^{-1}(x) \exp\left[-\frac{\hat{G}(x)}{\sigma^2} \right] \exp\left[\frac{\hat{G}(x')}{\sigma^2} \right] dx dx' \right\}^{-1}$$

where

σ^2 = linearized tracking variance

$n(x) = N_0(x)/N_0(0)$, relative noise spectral density

$$\hat{G}(x) = \int^{x} \frac{g_n(u)}{\eta(u)}\, du$$

$g_n(x)$ = normalized S-curve

Hence, evaluating this result for different bit synchronizers allows one to compare BSR's. As the loop SNR is increased, BSR decreases in a faster-than-exponential manner.

CONCLUSIONS

In this paper, we have discussed the types of bit synchronizers that can be used, the refinements needed to make them efficient at low values of SNR and transition density' and the means to digitize them. Finally, we discussed two measures which are useful in appraising their performance.

REFERENCES

1. Lindsey, W.C. and Tausworthe, R.C., "Digital Data-Transition Tracking Loops," JPL SPS No. 37-50, Volume III, April 1968.

2. Hurd, W.J. and Anderson, T.O., "Digital Transition Tracking Symbol Synchronizer for Low SNR-Coded Systems," IEEE Trans. Comm. Tech., Volume 18, April 1970.

3. Layland, J.W., "Telemetry Bit Synchronizer Loop," JPL SPS No. 37-46. Volume 3, pp. 204 to 215.

4. Simon, M.K., "An Analysis of the Steady-State Phase-Noise performance of a Digital Data-Transition Tracking Loop," IEEE Trans. Comm. Tech., Volume 18, October 1970.

5. Holmes, J. K., Coherent Spread Spectrum Systems, Chapter 12, Wiley Interscience, 1982.

6. Holmes, J. K., "Tracking Performance of the Filter-and-Square-Bit Synchronizer," IEEE Trans. Comm., Vol. COM-28, Part I, August 1980.

7. McCallister, R.D. and Simon, M.K., "Cross-Spectrum Symbol Synchronizer," ICC'81, Denver, Colorado, June 1981.

8. McCallister, R.D., "Optimum Tracking Performance of Delay-and Multiply Cross-Spectrum Symbol Synchronizers," NTC'81, New Orleans, Louisiana, November 29 to December 2, 1981.

9. Spilker, J.J., Digital Communications by Satellite, Prentice-Hall, Inc.

10. Savage, J.E., "Some Simple Self-Synchronizing Digital Data Scrambles," B.S.T.J., February 1967, pp. 449-487.

11. Lindsey, W.C. and Chie, C.M., "A Survey of Digital Phase-Locked Loops, Proceedings of the IEEE, Volume 69, No. 4, April 1981.

12. Natali, F.D., "All-Digital Coherent Demodulator Techniques," Proceedings of the ITC, Volume VIII, October 1972, pp. 89-108.

13. Martin, D. and Secor, D., "High-Speed Analog-to-Digital Conversions in Communication Systems," TRW Defense and Space Systems Group Report, November 1981.

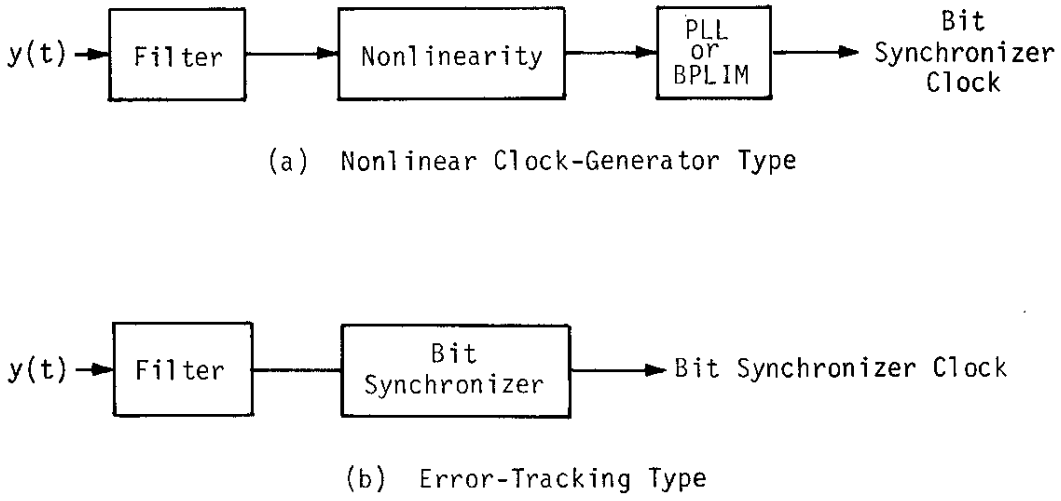

Figure 1. The Two Basic Types of Bit Synchronizers

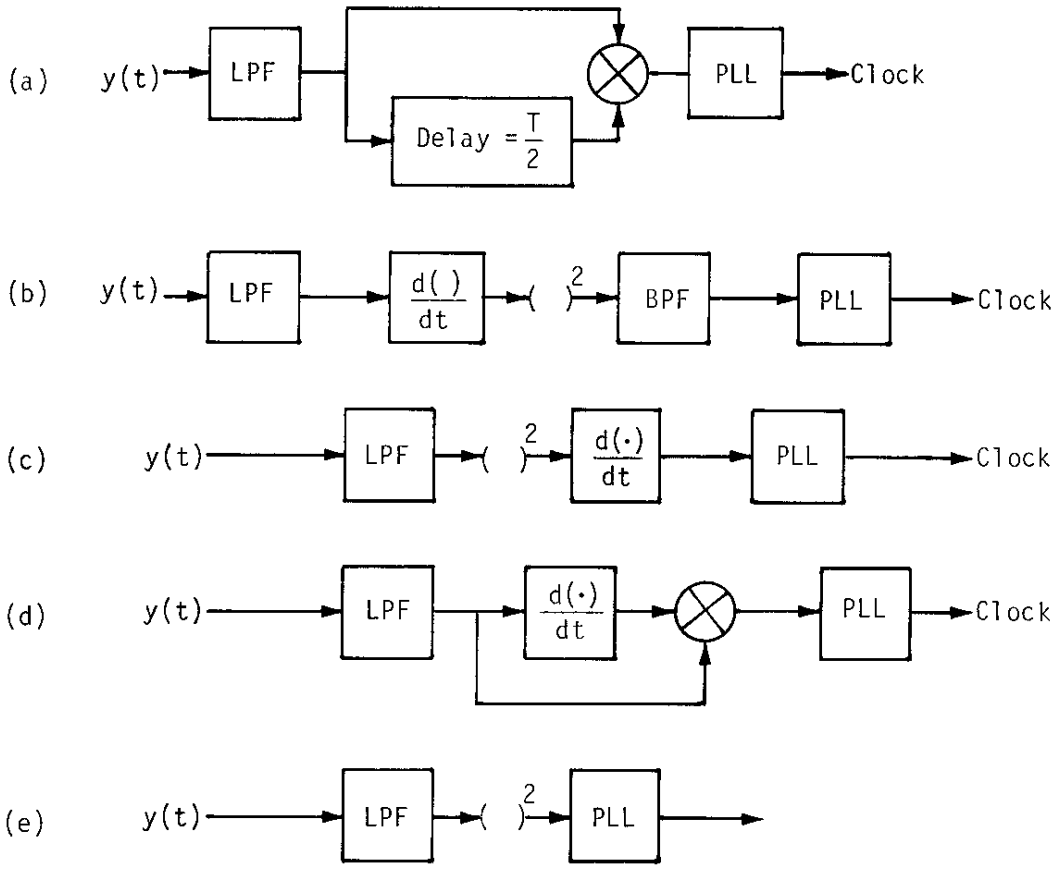

Figure 2. Some Nonlinear Clock-Generating Bit Synchronizers

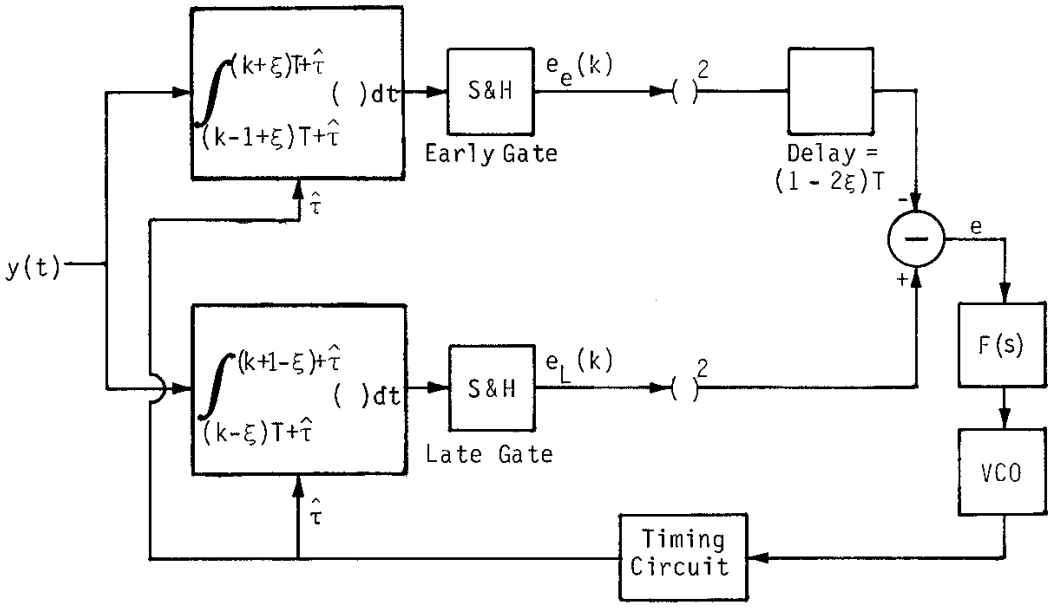

Figure 3(a). Early/Late Symbol Synchronization for NRZ Symbols with Square Law

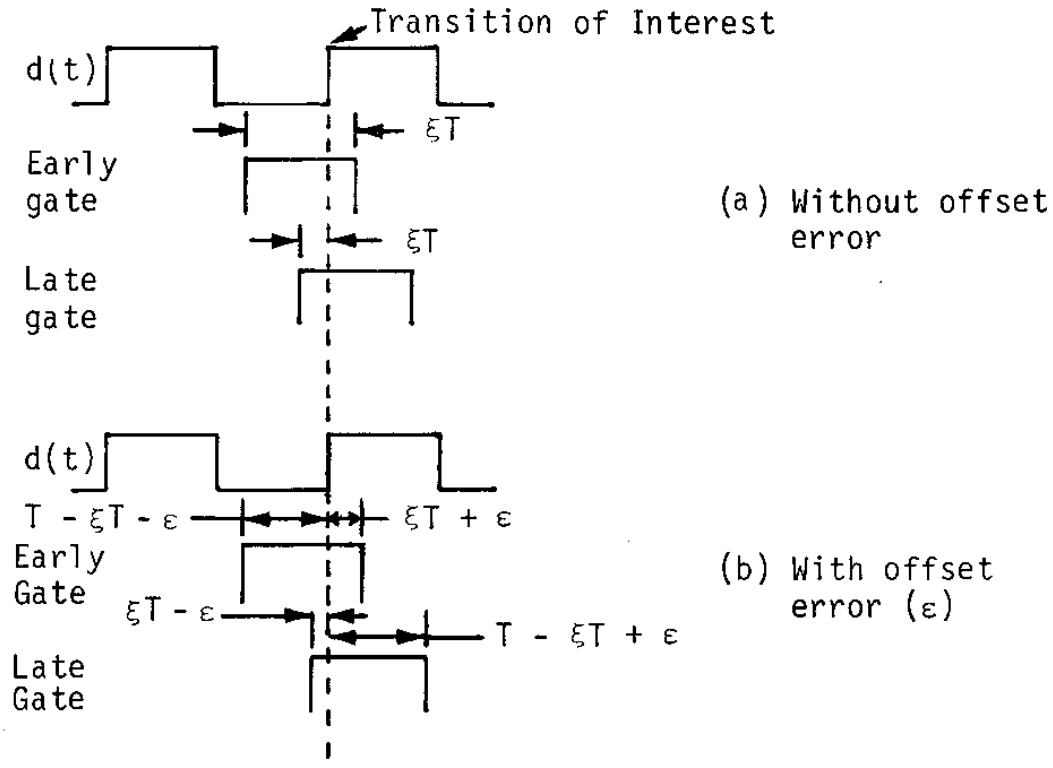

Figure 3(b). Data and Gating Waveforms in the Early/Late Gate Synchronizer

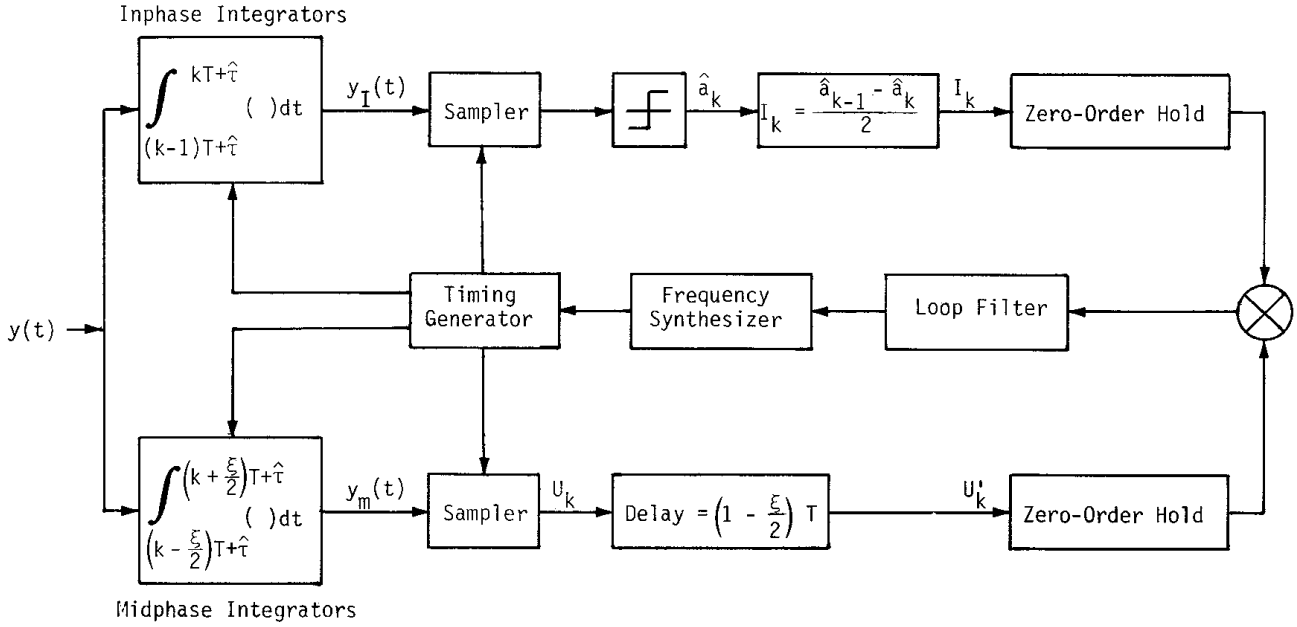

Figure 4. A Digital Data-Transition Tracking Loop for NRZ Symbols

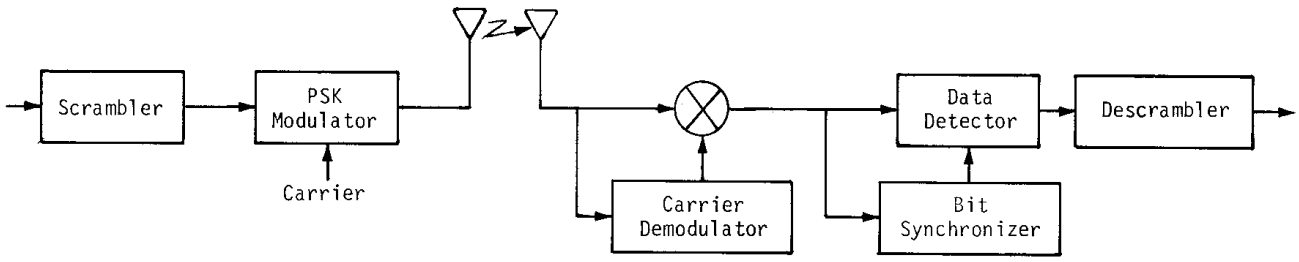

Figure 5. PSK Transmitter and Receiver with Scrambling and Descrambling

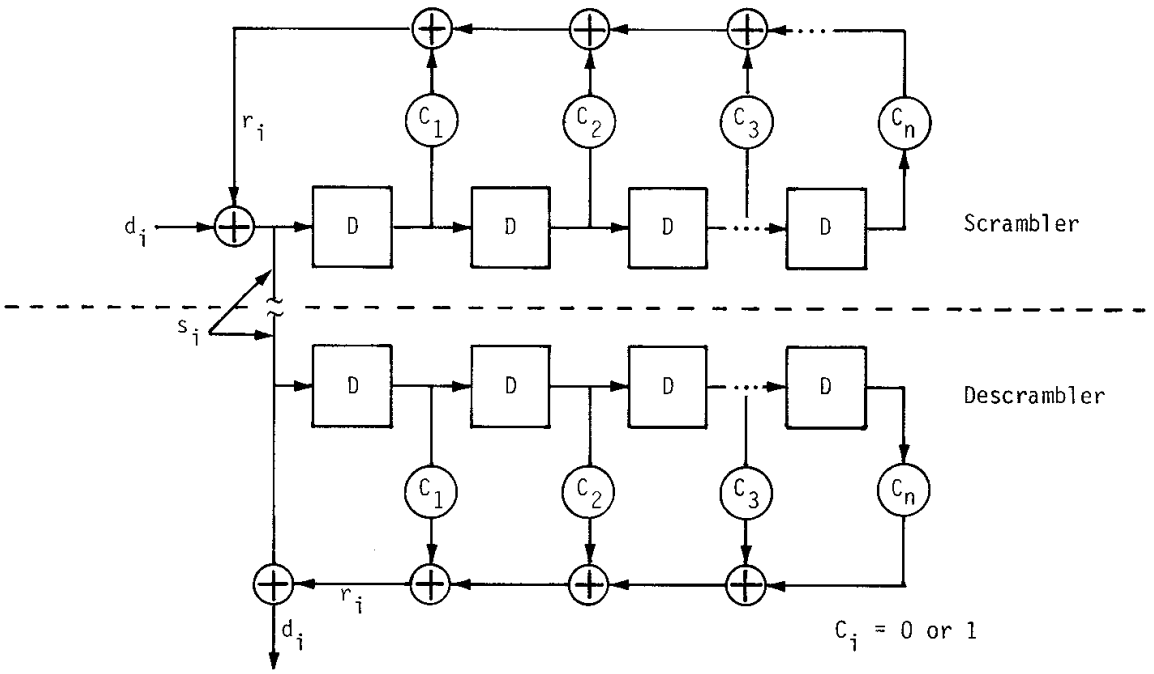

Figure 6. Scrambler and Descrambler

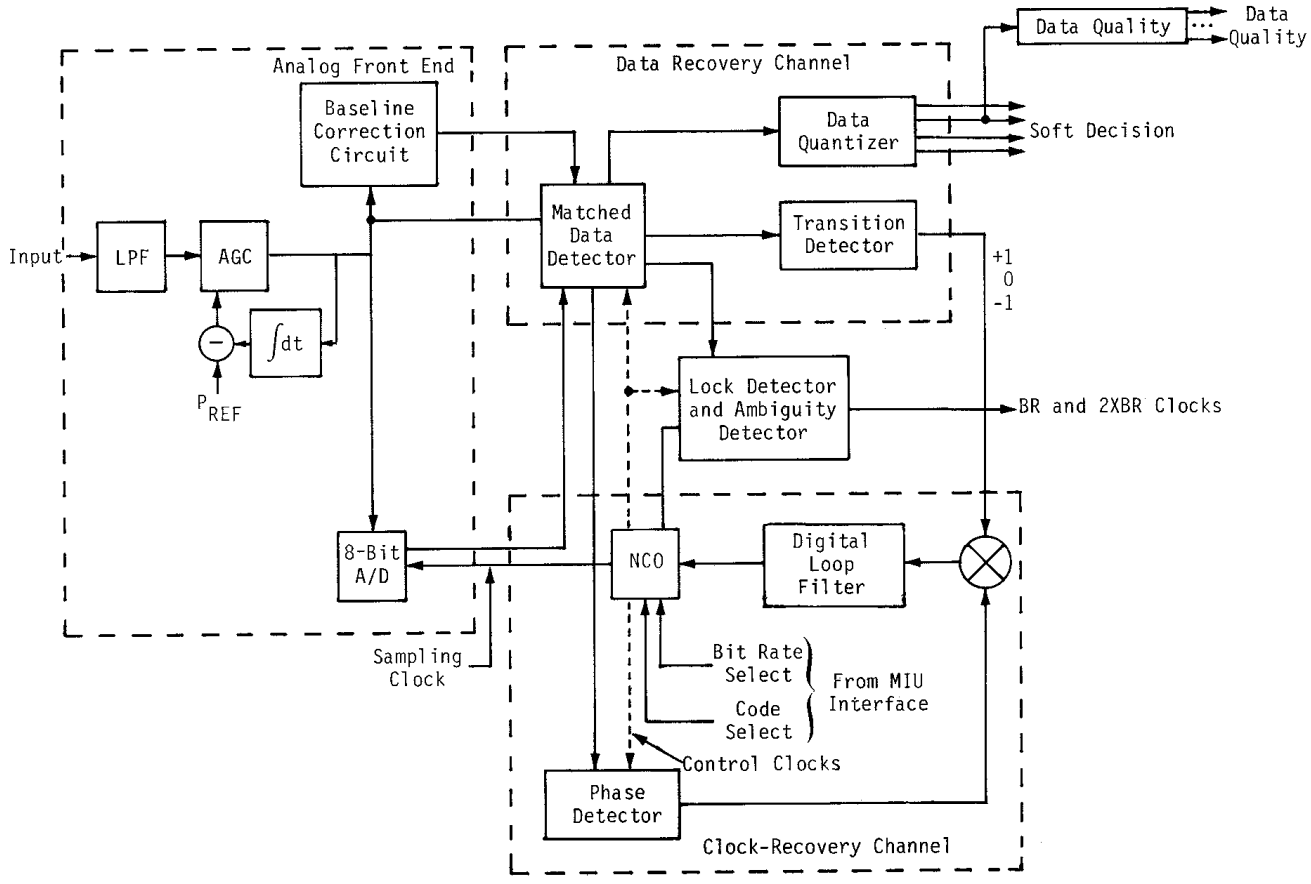

Figure 7. Typical Digitally Implemented DTTL Bit Synchronizer Block Diagram