CHAPTER 12

Randomization Methods for Telemetry Systems

Table of Figures

This page intentionally left blank.

Acronyms

ASM	attached synchronization marker
BER	bit error rate
CCSDS	Consultative Committee for Space Data Systems
dB	decibel
IRIG	Inter-Range Instrumentation Group
LDPC	Low Density Parity Check
XOR	exclusive-or

This page intentionally left blank.

CHAPTER 12

Randomization Methods for Telemetry Systems

12.1 Introduction

A randomizer is a device that manipulates the source data stream before transmission to apply a pseudo-random binary sequence; this provides maximum bit transition density. A de-randomizer reverses these manipulations at the destination.

This chapter specifies bit-level randomization methods that improve the spectral qualities, demodulation, and decoding of telemetry signals. Maximum bit transition density is needed to maintain the spectral qualities of IRIG 106 modulations, aid in telemetry signal acquisition, and maintain bit (or symbol) synchronization of a received telemetry signal. The randomizer prevents degenerative data patterns from degrading data quality.

The following randomization and de-randomization methods are recommended for wireless serial streaming telemetry data links.

12.2 Randomizer Types and Applications

There are two types of randomizers used in serial-streaming links: a self-synchronizing randomizer known as the IRIG randomizer; and a non-self-synchronizing randomizer known as the CCSDS randomizer.[1] The choice of randomizer depends on the application.

12.2.1 Self-Synchronizing Randomizers and Applications

Self-synchronizing randomizers apply a pseudo-random sequence to a continuous stream of input data. These randomizers are applicable to bit streams when no known identifiers or markers exist to aid in synchronizing the de-randomizer. Self-synchronizing randomizers do not need frame or block synchronization. The IRIG randomizer is a self- synchronizing randomizer.

The self-synchronizing de-randomizer has the characteristic of creating additional bit errors when a bit error is received at the de-randomizer input. See Subsection 12.3.3 for additional information.

The self-synchronizing randomizer should be applied to data for the direct recording of unencrypted pulse code modulation data. This randomizer should also be applied to all serial-streaming telemetry where encryption is not used. Randomization is not required or recommended when encryption is used in a telemetry link that is not using Low Density Parity Check (LDPC) forward error correction. In the case of adding LDPC to an encrypted link, the codeblock (data block plus parity block) will be randomized with the CCSDS randomizer.

12.2.2 Non-Self-Synchronizing Randomizers and Applications

Non-self-synchronizing randomizers apply a pseudo-random sequence to a block-structured input data stream. Pilot bits or a synchronization word is placed in the data stream at

[1] International Telecommunications Union. *General Requirements for Instrumentation for Performance Measurements on Digital Transmission Equipment*. ITU-T Recommendation O.150. May 1996. May be superseded by update. Retrieved 28 March 2024. Available at https://www.itu.int/rec/T-REC-O/en.

equal intervals, which maintains the synchronous operation of the randomizer and de-randomizer. The CCSDS randomizer is a non-self-synchronizing randomizer.

The CCSDS randomizer should be applied to serial-streaming links that incorporate LDPC forward error correction codes. This randomizer is also recommended for serial-streaming links exhibiting a block structure with synchronization markers.

Performance of the non-self-synchronizing randomizer will exceed that of a self-synchronizing randomizer, making this type a better choice for coded links or links requiring data-aided synchronization. Non-self-synchronizing randomizers do not create additional bit errors when a bit error is received at the de-randomizer input.

12.3 IRIG Randomizer Implementation

12.3.1 IRIG Randomization

The IRIG randomizer uses a network of shift registers and modulo-2 adders (exclusive-OR [XOR]).

The IRIG randomizer generates the randomized bit stream by applying an XOR operation using the input bit stream and the output of another XOR operation applied to the outputs of the 14th and 15th stages of a shift register. The output bit stream is also the input to the shift register. The pseudo-random sequence is generated using the polynomial $h(x) = x^{15} + x^{14} + 1$ (see Figure 12-1).

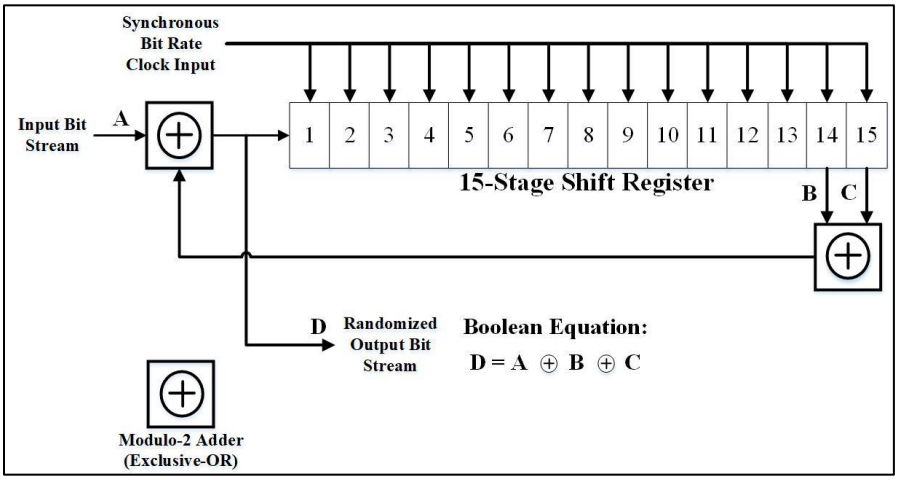

Figure 12-1. IRIG Randomizer Block Diagram

The properties of the output bit stream are similar to a pseudo-random sequence. A 15-stage IRIG randomizer will generate a maximum length pseudo-random sequence of $2^{15}-1$ (32,767) bits if the input data stream consists of zeros and the shift register contains at least a single "1". Conversely, a maximum length pseudo-random sequence occurs when the input bit stream consists of only "1's" and the shift register contains at least a single zero.

However, if the shift register contains all zeros when the input bit stream is all zeros, the randomized output bit stream will also be all zeros. The converse is also true: when the shift register contains all "1's" and the input bit stream also contains all "1's", the output bit stream from the randomizer will also be all "1's". In these two cases, the contents of the shift register

did not change, and the randomization of the output bit stream did not occur. Note that the randomizer is not permanently in this state because a change in the content of the input bit stream will again produce a randomized output.

In general, if the input bit stream contains runs of X bits without a transition with a probability of occurrence of p(X), the output will contain runs having a length of up to (X+15) bits with a probability equal to (2^{-15} • p(X)). Therefore, the output bit stream can contain long runs of bits without a transition, but the probability of occurrence is low.

NOTE 	This randomizer uses the same polynomial as the 32,767-bit pseudo-random test sequence (ITU, *General Requirements for Instrumentation*…), more commonly referred to as the PN15 or PRBS15 test pattern used throughout the telemetry industry. The concurrent use of IRIG randomization with the PN15 test sequence can yield degenerative results and should therefore be strictly avoided.

12.3.2 IRIG De-Randomizer Implementation

The de-randomizer shown in Figure 12-2 receives a randomized input data stream and synchronized bit-rate clock. De-randomization is accomplished by applying an XOR operation using the input bit stream and the output of another XOR operation applied to the outputs of the 14th and 15th stages of the shift register.

NOTE 	A reverse playback mode of the de-randomizer was available when magnetic tape recording was prevalent.[2]

[2] Range Commanders Council. "Magnetic Tape Recorder and Reproducer Information and Use Criteria." Annex A-2 in *Telemetry Standards*. RCC 106-23. Retrieved 28 March 2024. Available at https://www.trmc.osd.mil/wiki/x/9AAGCg.

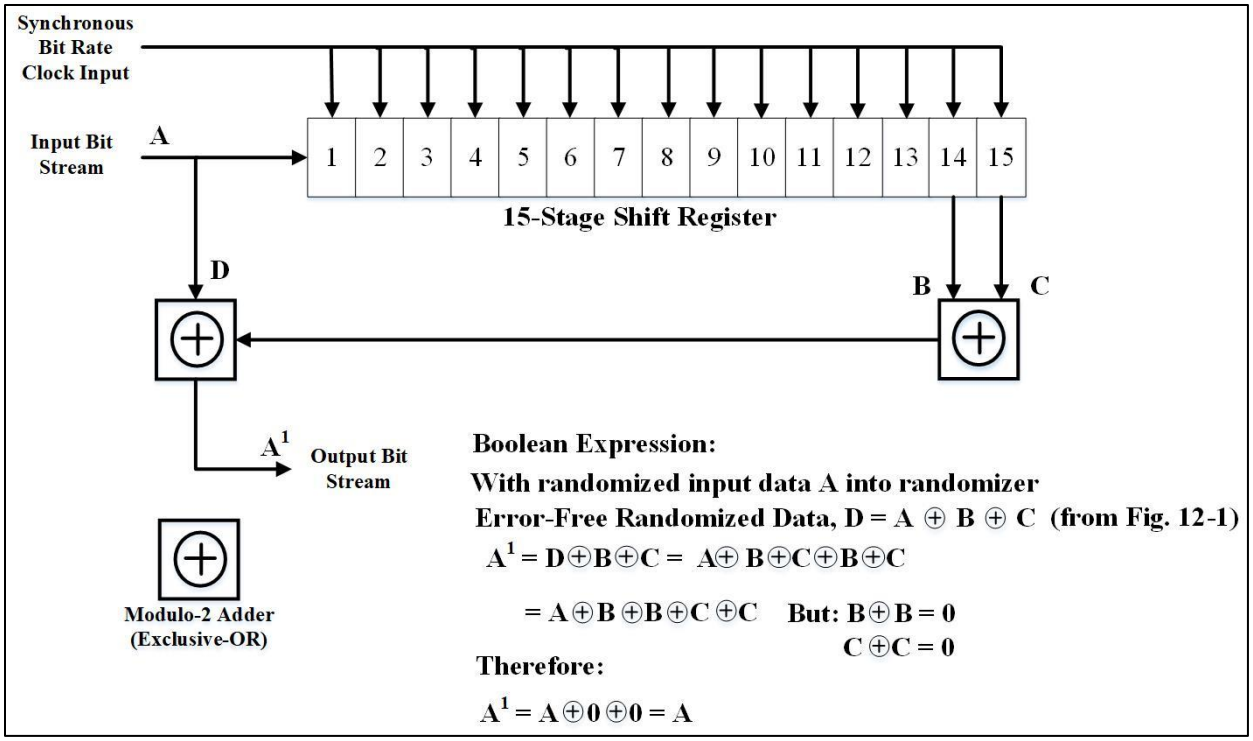

Figure 12-2. IRIG De-randomizer Block Diagram

12.3.3 IRIG De-Randomizer Response to Bit Errors

Although the IRIG de-randomizer is self-synchronizing, 15 consecutive error-free bits must be loaded into the shift register before the output bit stream will be valid.

A bit slip will cause the de-randomizer to lose synchronization, and 15 consecutive error-free data bits must again be loaded into the shift register before the output bit stream is valid. The de-randomized output bit stream, although correct, will contain the bit slip, which causes a shift in the data with respect to the frame synchronization pattern. Therefore, frame synchronization must be reacquired before the output bit stream provides meaningful data.

The IRIG de-randomizer has an error multiplication factor of 3 for isolated bit errors (separated from adjacent bit errors by at least 15 bits). A bit error received in the input bit stream will cause three errors in the output bit stream. One error will be the original bit in error, while two additional bit errors will occur 14 and 15 bits later. This error extension will degrade the detection efficiency of the telemetry link by an amount dependent on the steepness of the bit error rate (BER) curve of the system. For IRIG 106 modulation schemes near a BER of $1x10^{-5}$, the effective degradation is approximately 0.5 dB. For coded schemes using LDPC per IRIG 106 near a BER of $1x10^{-5}$, the effective degradation is approximately 0.04 dB.

In addition, a burst of errors occurring after the data has been randomized will produce a burst of errors in the de-randomized output. The number of errors in the output depends on the distribution of errors in the burst and can be greater than, equal to, or less than the number of errors in the input to the de-randomizer; however, the de-randomization process always increases the number of bits between the first and last error in the burst by 15.

Errors introduced prior to randomization are not affected by either the randomizer or the de-randomizer.

Though not standardized, IRIG randomization can be used in conjunction with LDPC. One example of this would be for backward compatibility. In this case, the order of decoding and de-randomization is critical. Due to error extension described above, de-randomization prior to decoding would severely degrade LDPC decoder performance. Therefore, randomization must be applied prior to LDPC encoding, and de-randomization must be applied subsequent to LDPC decoding.

12.4 CCSDS Randomizer Implementation

The CCSDS randomization/de-randomization generally describe the process of randomizing and de-randomizing a block-based bit stream with a pseudo-random sequence generator. The CCSDS randomization and de-randomization processes require block synchronization provided by a synchronization marker.[3]

12.4.1 CCSDS Randomization (Sending End)

The CCSDS randomizer ensures sufficient bit transitions to maintain bit (or symbol) synchronization with the received signal in a data block-based bit stream. This is accomplished by performing an XOR operation of each bit from the data block with the standard pseudo-random sequence. The first bit of the of the data block is XOR'ed with the first bit of the pseudo-random sequence followed by the second bit of the data block with the second bit of the pseudo-random sequence, and so on.

After randomization, an attached synchronization marker (ASM) or other bits used for receiver synchronization is prepended to each data block. The ASM is not randomized. This process is illustrated in Figure 12-3.

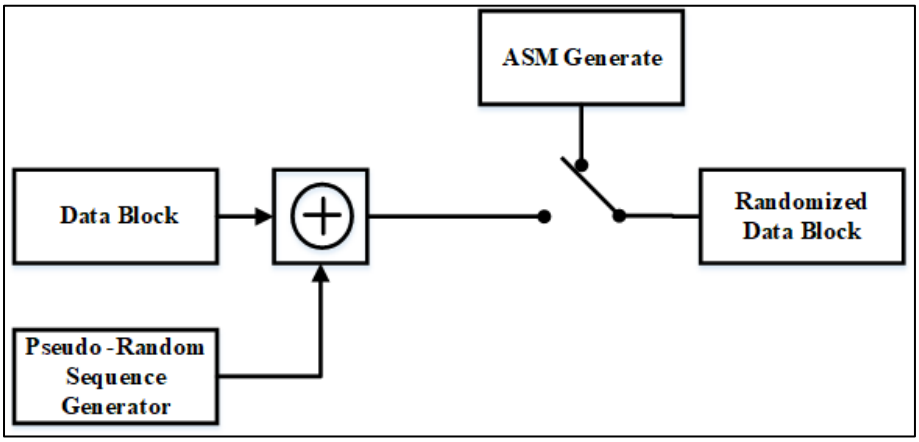

Figure 12-3. CCSDS Randomizer Block Diagram

12.4.2 CCSDS De-Randomization (Receiving End)

The CCSDS de-randomizer reconstructs the original data block using the same pseudo-random randomization sequence used for randomization.

After locating the ASM in the received data stream, the pseudo-random sequence is XOR'ed with the data bits immediately following the ASM. The pseudo-random sequence is

[3] CCSDS. *TM Synchronization and Channel Coding*. CCSDS 131.0-B-4. Blue Book. September 2023. May be superseded by update. Retrieved 28 March 2024. Available at https://public.ccsds.org/Pubs/131x0b5.pdf.

applied by XOR'ing the first bit following the ASM with the first bit of the pseudo-random sequence, followed by the second bit of the data stream with the second bit of the pseudo-random sequence, and so on. This process is shown in Figure 12-4. Unlike the IRIG de-randomizer, there is no error extension related to the CCSDS de-randomizer.

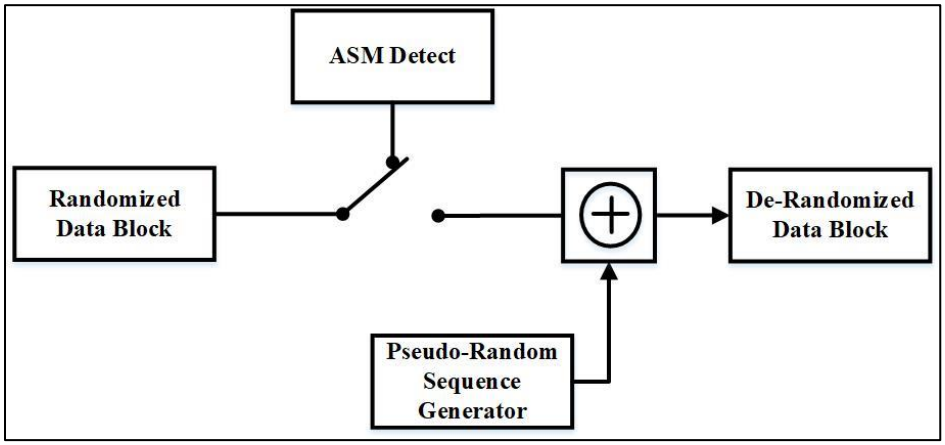

Figure 12-4. CCSDS De-Randomizer Block Diagram

12.4.3 CCSDS Pseudo-Random Sequence Generator

The 255-bit randomization sequence repeats until the end of the input data block. There is no limit to the length of the input data block. In coded systems, the data block length would be equal to the length of the code block or code word.

The pseudo-random sequence used for the CCSDS randomization and de-randomization process is generated by using the following polynomial: $h(x) = x^8 + x^7 + x^5 + x^3 + 1$ and is illustrated in Figure 12-5.

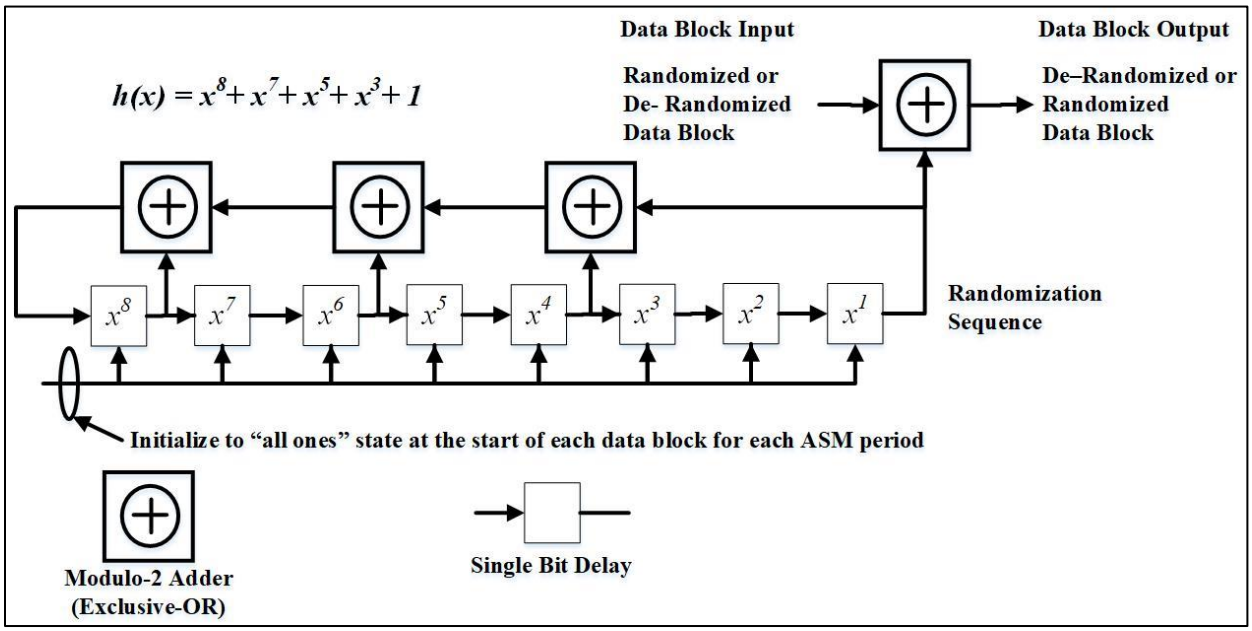

Figure 12-5. CCSDS Pseudo-Random Sequence Generator Block Diagram

This sequence begins at the first bit of the block of data to be randomized and repeats after 255 bits, continuing repeatedly until the end of the data block. The sequence generator is initialized to the all "1's" state at the start of each data block.

The first 40 bits of the pseudo-random sequence from the generator are shown below. The leftmost bit is the first bit of the sequence to be XOR'ed with the first bit of the data block, the second bit of the sequence is XOR'ed the second bit of the data block, and so on.

The initial sequence of 40 bits is 1111 1111 0100 1000 0000 1110 1100 0000 1001 1010…….

This page intentionally left blank.

APPENDIX 12-A

Citations

International Telecommunications Union. *General Requirements for Instrumentation for Performance Measurements on Digital Transmission Equipment*. ITU-T Recommendation O.150. May 1996. May be superseded by update. Retrieved 28 March 2024. Available at https://www.itu.int/rec/T-REC-O/en.

Range Commanders Council. "Magnetic Tape Recorder and Reproducer Information and Use Criteria." Annex A-2 in *Telemetry Standards*. RCC 106-23. Retrieved 28 March 2024. Available at https://www.trmc.osd.mil/wiki/x/9AAGCg.

The Consultative Committee for Space Data Systems. *TM Synchronization and Channel Coding*. CCSDS 131.0-B-5. Blue Book. September 2023. May be superseded by update. Retrieved 28 March 2024. Available at https://public.ccsds.org/Pubs/131x0b5.pdf.

****** END OF CHAPTER 12 ******