ATSC Standard: System Discovery and Signaling

A/321:2026-06
11 June 2026

ATSC®, the Broadcast Standards Association®, is an international, non-profit organization developing voluntary standards and recommended practices for broadcast television and multimedia data distribution. ATSC member organizations represent the broadcast, professional equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society Coordination (JCIC): the Consumer Technology Association (CTA), the Institute of Electrical and Electronics Engineers (IEEE), the National Association of Broadcasters (NAB), the Internet & Television Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). For more information visit www.atsc.org.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at https://www.atsc.org/feedback/.

Revision History

Version	Date
Candidate Standard approved	6 May 2015
Revised CS approved (editorial and substantive changes made)	7 December 2015
Standard approved	23 March 2016
Editorial correction: moved "system_bandwidth" text from before Table 6.3 to after Table 6.3.	12 September 2016
Updated reference to A/331 [2] to point to the current published version	6 December 2017
A/321:2022-03 (references to ATSC documents updated)	31 March 2022
A/321:2023-03 (references to ATSC documents updated)	28 March 2023
A/321:2024-04 (references to ATSC documents updated)	3 April 2024
A/321:2025-07 (references to ATSC documents updated)	17 July 2025
A/321:2026-04 (references to ATSC documents updated)	14 April 2026
A/321:2026-04 Amendment No. 1 approved	11 June 2026
A/321:2026-06 (includes Amendment No. 1 to A/321:2026-04)	11 June 2026

Table of Contents

Index of Figures and Tables

ATSC Standard:
System Discovery and Signaling

1. SCOPE

This Standard constitutes the normative specification for the initial entry point of a physical layer waveform. Syntax and semantics of this specification are for system discovery only and other ATSC Standards may further constrain and/or supplement this physical layer discovery specification.

1.1 Introduction and Background

Broadcasters anticipate providing multiple wireless-based services, in addition to just broadcast television, in the future. Such services may be time-multiplexed together within a single RF channel. As a result, there exists a need to indicate, at a low level, the type or form of a signal that is being transmitted during a particular time period, so that a receiver can discover and identify the signal, which in turn indicates how to receive the services that are available via that signal.

To enable such discovery, a bootstrap signal is required. This comparatively short signal precedes, in time, a longer transmitted signal that carries some form of data. New signal types, at least some of which have likely not yet even been conceived, could also be provided by a broadcaster and identified within a transmitted waveform through the use of a bootstrap signal associated with each particular time-multiplexed signal. Some future signal types indicated by a particular bootstrap signal may even be outside the scope of ATSC.

The bootstrap provides a universal entry point into a broadcast waveform for all signal types conveyed in the RF channel. The bootstrap employs a fixed configuration (e.g., sampling rate, signal bandwidth, subcarrier spacing, time-domain structure) known to all receiver devices and carries information to enable processing and decoding the signal associated with a detected bootstrap. This capability ensures that broadcast spectrum can be adapted to carry new signal types that are preceded by the universal entry point provided by the bootstrap, for public interest to continue to be served in the future.

The bootstrap has been designed to be a very robust signal and detectable even at low signal levels. As a result of this robust encoding, individual signaling bits within the bootstrap are comparatively expensive in terms of the physical resources that they occupy for transmission. Hence, the bootstrap is generally intended to signal only the minimum amount of information required for system discovery (i.e., identification of the associated signal) and for initial decoding of the following signal.

1.2 Organization

This document is organized as follows:

- Section 1 – Outlines the scope of this document and provides a general introduction
- Section 2 – Lists references and applicable documents
- Section 3 – Provides a definition of terms, acronyms, and abbreviations for this document
- Section 4 – Bootstrap overview
- Section 5 – Detailed bootstrap specification
- Section 6 – Contains bootstrap signaling sets that provide bootstrap configurations specific to a particular signal type (such as ATSC 3.0)

- Annex A; Example Method of Gray Code De-mapping at Receiver
- Annex B: Bootstrap Signaling Bit Robustness and Other Characteristics

2. REFERENCES

All referenced documents are subject to revision. Users of this Standard are cautioned that newer editions might or might not be compatible.

2.1 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

[1] IEEE: "Use of the International Systems of Units (SI): The Modern Metric System," Doc. SI 10, Institute of Electrical and Electronics Engineers, New York, NY.

2.2 Informative References

The following documents contain information that may be helpful in applying this Standard.

[2] ATSC: "ATSC Standard: Signaling, Delivery, Synchronization and Error Protection," Doc. A/331:2026-04, ATSC, Washington, DC, 14 April 2026.

[3] ATSC: "ATSC Standard: Physical Layer Protocol," Doc. A/322:2026-06, ATSC, Washington, DC, 11 June 2026.

3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute's published standards [1] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

shall – This word indicates specific provisions that are to be followed strictly (no deviation is permitted).

shall not – This phrase indicates specific provisions that are absolutely prohibited.

should – This word indicates that a certain course of action is preferred but not necessarily required.

should not – This phrase means a certain possibility or course of action is undesirable but not prohibited.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, transport and transmission coding subsystems. These references are typographically distinguished by the use of a different font (e.g., restricted), may contain the underscore character (e.g., sequence_end_code) and may consist of character strings that are not English words (e.g., dynrng).

3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These are used primarily to enable adding new values to a syntactical structure

without altering its syntax or causing a problem with backward compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1.' There is no default value for other reserved elements. Use of reserved elements except as defined in ATSC Standards or by an industry standards setting body is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently-reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently-reserved elements to avoid possible future failure to function as intended.

3.3 Acronyms, Abbreviations and Mathematical Operators

The following acronyms and abbreviations are used within this document.

APLF – Alternative Physical Layer Frame
BSP – Baseband Sample Period
BSR – Baseband Sampling Rate
CAZAC – Constant Amplitude Zero Auto-Correlation
DC– Direct Current
EAS – Emergency Alert System
FFT – Fast Fourier Transform
IEEE – Institute of Electrical and Electronic Engineers
IFFT – Inverse Fast Fourier Transform
kHz – kilohertz
LFSR – Linear Feedback Shift Register
MHz – Megahertz
ms – millisecond
PN – Pseudo-Noise
RCS – relative cyclic shift
µs – microsecond
ZC– Zadoff-Chu
$\lfloor X \rfloor$ The greatest integer less than or equal to X

3.4 Terms

The following terms are used within this document.

Alternative Physical Layer Frame – A Physical Layer Frame that is of one of two or more different types of Physical Layer Frames that alternate within an emission, using Bootstraps as defined herein to introduce each such Physical Layer Frame. Alternative Physical Layer Frame adds to the original structure the construct that has been referenced colloquially by the industry as "Evolutionary Frames," "Future Extension Frames," and the like.

Baseband Sampling Rate – The frequency at which a baseband signal is converted into digital samples.

Baseband Sample Period – The time interval between two consecutive baseband samples, defined as the reciprocal of the Baseband Sampling Rate.

Reserved – Set aside for future use by a Standard.

3.5 Extensibility

This Standard is designed to be extensible via both backward-compatible mechanisms and by replacement syntactical mechanisms that are not backward-compatible. It also establishes means to explicitly signal collections of components to establish services with various characteristics. The enumeration of the set of components that can be used to present a service is established to enable different combinations of the defined components to be offered without altering this Standard.

3.5.1 Backward-compatible Extensibility Mechanisms

The backward-compatible mechanisms are:

Table length extensions – Future amendments to this Standard may include new fields at the ends of certain tables. Tables that may be extensible in this way include those in which the last byte of the field may be determined without use of the section_length field. Such an extension is a backward-compatible addition.

Definition of reserved values – Future amendments to this Standard may establish meaning for fields that are asserted to be "reserved" in a table's syntax, semantic or schema in the initial release. Such an extension is a backward-compatible addition due to the definition of "reserved."

3.5.2 Non-backward-compatible Extensibility Mechanisms

Tables or other structures that can be changed in a non-compatible manner each contain a field or other signaling mechanism labeled major version (or major_version) in order to explicitly signal their syntax. More than one instance (each with a different major version) can be expected to be present wherever such tables, schema, or structures are used.

3.5.3 Extensions with Unknown Compatibility

This Standard establishes a general signaling approach that enables new combinations of components to be transmitted that define a new or altered service offering. Receiver support for such sets is unknown and labeling of such sets of extensions to the service signaling established herein is the responsibility of the document establishing a given set of capabilities.

4. BOOTSTRAP OVERVIEW

4.1 Features

The bootstrap provides a universal entry point into a digital transmission signal. It employs a fixed configuration (e.g., sampling rate, signal bandwidth, subcarrier spacing, time domain structure) known to all receiver devices.

Figure 4.1 shows an overview of the general structure of a physical layer frame, the bootstrap signal, and the bootstrap position relative to the post-bootstrap waveform (i.e., the remainder of the frame). The bootstrap consists of a number of symbols, beginning with a synchronization symbol positioned at the start of each frame period to enable signal discovery, coarse synchronization, frequency offset estimation, and initial channel estimation. The remainder of the bootstrap contains sufficient control signaling to permit the reception and decoding of the remainder of the frame to begin.

Only the bootstrap structure and contents are specified within the present document.

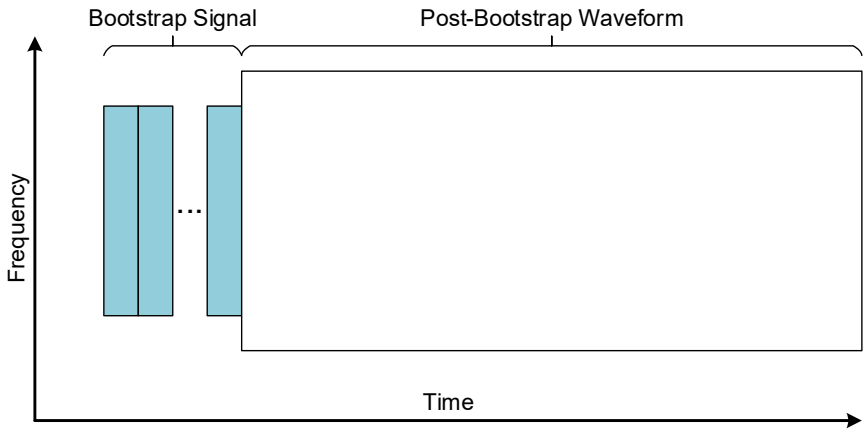

Figure 4.1 General physical layer frame and bootstrap structure.

The structure in Figure 4.1 applies to all signal types, time multiplexed based on this Standard. For all physical layer waveforms present in the RF channel (including physical layer signal types defined by standards organizations other than ATSC), each physical layer frame shall begin with a bootstrap. The start of the bootstrap for the next physical layer frame shall immediately follow the end of the current frame's post-bootstrap waveform. Any gap between adjacent frames shall not exceed the longer of the Baseband Sample Periods of the two frames.

4.2 Central Concepts

The bootstrap design exhibits flexibility via the following core concepts.

- **Versioning:** The bootstrap version is expressed in text as a major version number (decimal digit) followed by a period and a minor version number (decimal digit), e.g., bootstrap version 0.0. The major version and minor version are referenced in code as bootstrap_major_version and bootstrap_minor_version, respectively. A Zadoff-Chu (ZC) root and a pseudo-noise (PN) sequence seed are used for generating the base encoding sequence for bootstrap symbol contents. A major version number (corresponding to a particular signal type) is signaled via selection of the ZC root. A minor version (within a particular major version) is signaled via appropriate selection of the PN sequence seed. The syntax and semantics of signaling fields within the bootstrap are specified within the Standard(s) to which the major and minor versions refer.
- **Scalability:** The number of bits signaled per bootstrap symbol is defined, up to a specified maximum, for a particular major/minor version. The maximum number of bits per symbol is

$$N_{bps} = \left\lfloor \log_2(N_{FFT}/CyclicShiftTol) \right\rfloor,$$

where $\lfloor X \rfloor$ is the greatest integer less than or equal to X (Floor function).

N_{bps} affects the cyclic shift tolerance, and is specified in the Standard(s) for the particular version. The number of signaling bits per symbol can be increased up to the specified maximum as a backward-compatible change when incrementing the minor version within the same major version.

- **Extensibility:** The bootstrap signal duration is extensible in whole symbol periods, with each new symbol carrying up to N_{bps} additional signaling bits. Bootstrap signal termination is signaled by a final symbol having 180° phase inversion relative to the preceding symbol.
- A bootstrap containing undefined signaling information (such as the use of reserved values) is expected to be discarded by the receiver.
- **Universal Accommodation**: A bootstrap prefix identifies the signal type (i.e., physical layer technology) applied to the associated post-bootstrap waveform when multiple signal types are time-multiplexed. A receiver is expected to discard frames when the bootstrap versions of those frames indicate signal type(s) unsupported by the receiver.

5. BOOTSTRAP SPECIFICATION

5.1 Signal Dimensions

The bootstrap sampling rate, bandwidth, FFT size, and symbol length shall remain fixed even as version numbers and/or the other information signaled by the bootstrap evolve.

The bootstrap shall use a fixed sampling rate of 6.144 Msamples/second and a fixed bandwidth of 4.5 MHz, regardless of the channel bandwidth used for the remainder of the frame. The time length of each sample of the bootstrap is fixed by the sampling rate.

$$f_S = 6.144 \text{ Ms/sec}$$

$$T_S = 1/f_S$$

$$BW_{\text{Bootstrap}} = 4.5 \text{ MHz}$$

An FFT size of 2048 results in a subcarrier spacing of 3 kHz.

$$N_{FFT} = 2048$$

$$f_\Delta = f_S/N_{FFT} = 3 \text{ kHz}$$

Each bootstrap symbol shall have time duration of 500 μs.

$$T_{\text{symbol}} = 500\ \mu s$$

The overall time duration of the bootstrap depends on the number of bootstrap symbols, which is specified as N_S. A fixed number of bootstrap symbols shall not be assumed.

5.2 Frequency Domain Sequence

The values used for each bootstrap symbol shall originate in the frequency domain with a ZC sequence modulated by a pseudo-noise (PN) sequence as shown in Figure 5.1. The ZC root and PN seed shall signal the major and minor versions of the bootstrap, respectively.

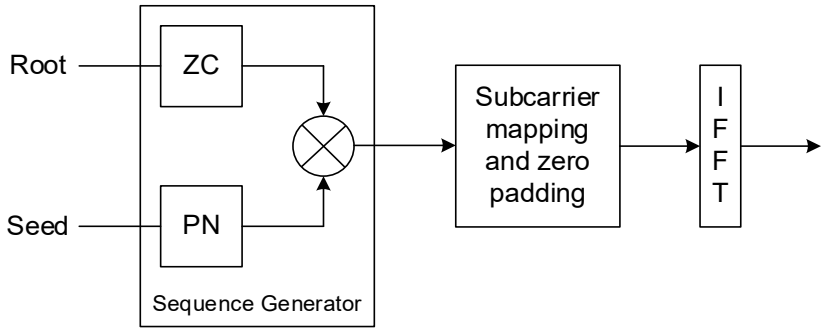

Figure 5.1 Frequency domain processing for bootstrap generation.

The resulting complex sequence shall be applied per subcarrier at the IFFT input. The PN sequence shall introduce a phase rotation to individual complex subcarriers, thus retaining the desirable Constant Amplitude Zero Auto-Correlation (CAZAC) properties of the original ZC sequence. The PN sequence further suppresses spurious peaks in the autocorrelation response, thereby providing additional signal separation between cyclic shifts of the same root sequence.

5.2.1 ZC Sequence Generation

The ZC sequence $z_q(k)$ shall have length $N_{ZC} = 1499$. This is the largest prime number that results in a channel bandwidth no greater than 4.5 MHz with a subcarrier spacing of $f_\Delta = 3$ kHz.

The ZC sequence shall be parameterized by a root, q, that corresponds to a major version number, where

$$z_q(k) = e^{-j\pi q\frac{k(k+1)}{N_{ZC}}}$$

In the above equation, $q \in \{1, 2, \ldots, N_{ZC} - 1\}$ and $k = 0, 1, 2, \ldots, N_{ZC} - 1$.

5.2.2 Pseudo-Noise Sequence Generation

The PN sequence generator shall be derived from a Linear Feedback Shift Register (LFSR) of length (order) $l = 16$ as shown in Figure 5.2. Its operation shall be governed by a generator polynomial g specifying the taps in the LFSR feedback path. Specification of the generator polynomial g and initial state of the registers, r_{init} defines a seed, which corresponds to a minor version number.

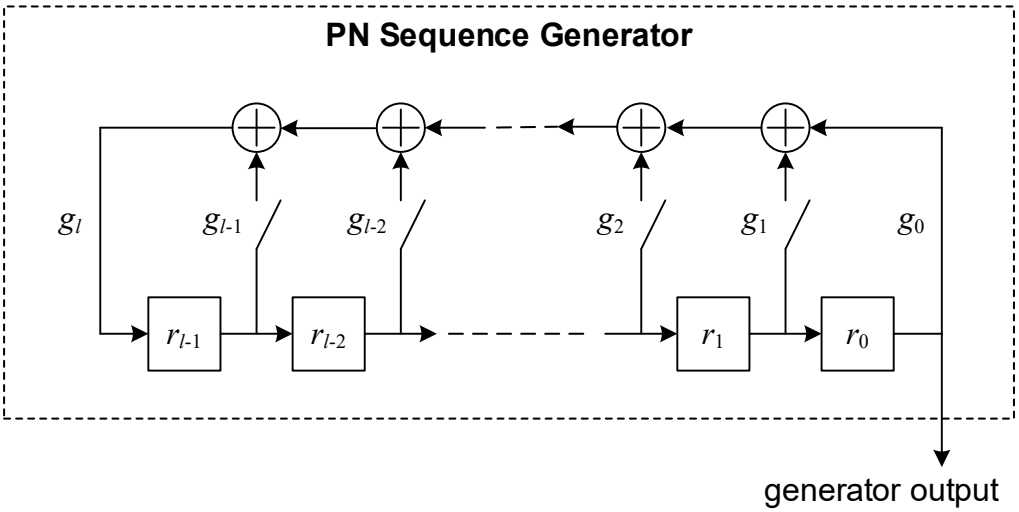

Figure 5.2 Pseudo-noise sequence generator.

The PN sequence generator registers shall be reinitialized with the initial state from the seed prior to the generation of the first symbol in a new bootstrap. The PN sequence generator shall continue to sequence from one symbol to the next within a bootstrap and shall not be re-initialized for successive symbols within the same bootstrap.

The output from the PN sequence generator in Figure 5.2 is defined to be $p(k)$. $p(k)$ will have either the value 0 or 1. $p(0)$ shall be equal to the PN sequence generator output after the PN sequence generator has been initialized with the appropriate seed value and before any clocking of the shift register in Figure 5.2 occurs. A new output bit $p(k)$ shall subsequently be generated every time the shift register in Figure 5.2 is clocked one position to the right.

The generator polynomial for the pseudo-noise sequence generator shall be as follows.

$$\boldsymbol{g} = \{g_l, \ldots, g_0\} = \{1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,1\}$$

$$p(x) = x^{16} + x^{15} + x^{14} + x + 1$$

5.2.3 Subcarrier Mapping and Modulation

Figure 5.3 shows an overview of the mapping of the frequency domain sequence to subcarriers. The ZC sequence value that maps to the DC subcarrier (i.e., $z_q((N_{ZC} - 1)/2)$) shall be set to zero so that the DC subcarrier is null. The subcarrier indices shall be as shown in Figure 5.3 with the central DC subcarrier having index 0.

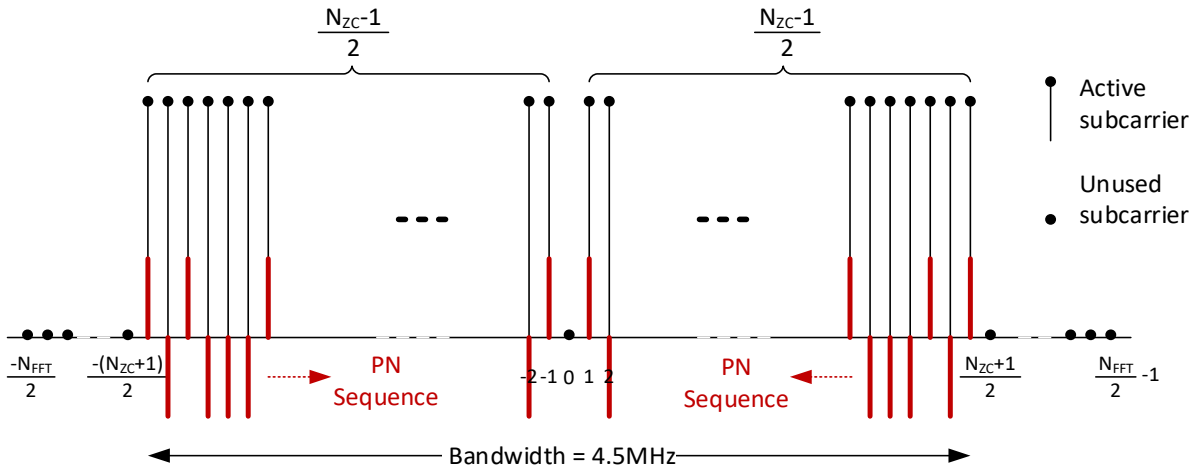

Figure 5.3 Sequence mapping to subcarriers.

The product of the ZC and PN sequences shall have reflective symmetry about the DC subcarrier. The ZC sequence has a natural reflective symmetry about the DC subcarrier. A reflective symmetry of the PN sequence about the DC subcarrier shall be introduced by mirror-reflecting the PN sequence values assigned to subcarriers below the DC subcarrier to the subcarriers above the DC subcarrier. For example, in Figure 5.3 the PN sequence values at subcarriers -1 and +1 are identical, as are the PN sequence values at subcarriers -2 and +2 As a result, the product of the ZC and PN sequences also has reflective symmetry about the DC subcarrier.

As illustrated in Figure 5.3, the subcarrier values for the n-th symbol of the bootstrap ($0 \leq n < N_S$) shall be calculated as follows, where $N_H = (N_{ZC} - 1)/2$. The ZC sequence shall be the same for every symbol, while the PN sequence shall advance with each symbol.

$$s_n(k) = \begin{cases} z_q(k + N_H) \times c\big((n+1) \times N_H + k\big) & -N_H \leq k \leq -1 \\ z_q(k + N_H) \times c\big((n+1) \times N_H - k\big) & 1 \leq k \leq N_H \\ 0 & \text{otherwise} \end{cases}$$

where $c(k) = 1 - 2 \times p(k)$ with $c(k)$ having either the value +1 or -1.

The final symbol in the bootstrap shall be indicated by a phase inversion (i.e., a rotation of 180°) of the subcarrier values for that particular symbol. This bootstrap termination signaling enables extensibility by allowing the number of symbols in the bootstrap to be increased for additional signaling capacity in a backward-compatible manner without requiring the major version number to be changed. Phase inversion is equivalent to multiplying each subcarrier value by $e^{j\pi} = -1$.

$$\tilde{s}_n(k) = \begin{cases} s_n(k) & 0 \leq n < N_S - 1 \\ -s_n(k) & n = N_S - 1 \end{cases}$$

This phase inversion allows receivers to correctly determine the end point of the bootstrap, including the end point of a bootstrap for a minor version (of the same major version) that is later than the minor version for which a receiver was designed and that has been extended by one or

more bootstrap symbols. Receivers are not expected to respond to the signaling bit contents of a bootstrap symbol that the receiver has not been provisioned to decode.

5.2.4 Inverse Fast Fourier Transform

The mapped frequency domain sequence $\tilde{s}_n(k)$ shall be translated to a time domain sequence $\tilde{A}_n(t)$ using a $N_{FFT} = 2048$ point IFFT.

$$\tilde{A}_n(t) = \frac{1}{sqrt(Nzc-1)}\left(\sum_{k=-(N_{ZC}-1)/2}^{-1} \tilde{s}_n(k)\, e^{j2\pi k f_\Delta t} + \sum_{k=1}^{(N_{ZC}-1)/2} \tilde{s}_n(k) e^{j2\pi k f_\Delta t}\right)$$

5.3 Symbol Signaling

5.3.1 Signaling Bits

Information shall be signaled via the bootstrap symbols through the use of cyclic shifts in the time domain of the $\tilde{A}_n(t)$ time domain sequence. This sequence has a length of $N_{FFT} = 2048$ and thus 2048 distinct cyclic shifts are possible (from 0 to 2047, inclusive).

With 2048 possible cyclic shifts, up to $\log_2(2048) = 11$ bits can be signaled. In reality, not all of these bits will actually be used. Let N_b^n specify the number of valid signaling bits that are used for the *n*-th bootstrap symbol ($1 \le n < N_S$), and let $b_0^n, \ldots, b_{N_b^n-1}^n$ represent the values of those bits. Each of the valid signaling bits $b_0^n, \ldots, b_{N_b^n-1}^n$ shall have the value 0 or 1. Each of the remaining signaling bits $b_{N_b^n}^n, \ldots, b_{10}^n$ shall be set to 0.

N_b^n for one or more specific bootstrap symbols may be increased when defining a new minor version within the same major version in order to make use of previously unused signaling bits while still maintaining backward compatibility. A receiver provisioned to decode the signaling bits for a particular major/minor version is not expected to decode any new additional signaling bits that may be used in a later minor version within the same major version.

5.3.2 Relative Cyclic Shift

Let $\widetilde{M}_n$ ($0 \le \widetilde{M}_n < N_{FFT}$) represent the cyclic shift for the *n*-th bootstrap symbol ($1 \le n < N_S$) relative to the cyclic shift for the previous bootstrap symbol. $\widetilde{M}_n$ shall be calculated from the valid signaling bit values for the *n*-th bootstrap symbol using a Gray code created per the following equations. Let $\widetilde{M}_n$ be represented in binary form as a set of bits $m_{10}^n\, m_9^n \ldots m_1^n\, m_0^n$. Each bit of $\widetilde{M}_n$ shall be computed as follows, where the summation of the signaling bits followed by the modulo-two operation effectively performs a logical exclusive OR operation on the signaling bits in question.

$$m_i^n = \begin{cases} \left(\sum_{k=0}^{10-i} b_k^n\right) \bmod 2 & i > 10 - N_b^n \\ 1 & i = 10 - N_b^n \\ 0 & i < 10 - N_b^n \end{cases}$$

The above equation ensures that the relative cyclic shift $\widetilde{M}_n$ is calculated to provide the maximum tolerance to any errors at the receiver when estimating the relative cyclic shift for a received bootstrap symbol. If the number of valid signaling bits N_b^n for a specific bootstrap symbol

is increased in a future minor version within the same major version, the equation also ensures that the relative cyclic shifts for that future minor version bootstrap symbol will be calculated in such a manner that will still allow a receiver provisioned for an earlier minor version to correctly decode the signaling bit values that it is provisioned to decode, and hence backward compatibility will be maintained.

Note: In general, the expected robustness of signaling bit b_i^n will be greater than that of b_k^n if $i < k$.

5.3.3 Absolute Cyclic Shift

The first bootstrap symbol shall be used for initial time synchronization and shall signal the major and minor version numbers via the ZC root and PN seed parameters, respectively. This symbol does not signal any additional information and shall always have a cyclic shift of 0.

The differentially-encoded absolute cyclic shift, M_n ($0 \leq M_n < N_{FFT}$), applied to the n-th bootstrap symbol shall be calculated by summing the absolute cyclic shift for bootstrap symbol n–1 and the relative cyclic shift for bootstrap symbol n, modulo the length of the time domain sequence.

$$M_n = \begin{cases} 0 & n = 0 \\ \left(M_{n-1} + \tilde{M}_n\right) \bmod N_{FFT} & 1 \leq n < N_S \end{cases}$$

The absolute cyclic shift shall then be applied to obtain the cyclically shifted time domain sequence $A_n(t)$ from the output of the IFFT operation.

$$A_n(t) = \tilde{A}_n\left((t + M_n) \bmod N_{FFT}\right)$$

5.4 Time Domain Structure

Each bootstrap symbol shall be composed of three parts: A, B, and C, where each of these parts consists of a sequence of complex-valued time domain samples. Part A shall be derived as the IFFT of the frequency domain structure with an appropriate cyclic shift applied as shown in Figure 5.4 (i.e. part A shall be equal to $A_n(t)$).

Parts B and C shall each be composed of samples taken from part A with a frequency shift of $\pm f_\Delta$ (equal to the subcarrier spacing) and a possible phase shift of $e^{-j\pi}$ introduced to the frequency domain sequence $\tilde{s}_n(k)$ used for calculating the samples of part B.

Parts A, B, and C shall consist of $N_A = N_{FFT} = 2048$, N_B =504, and $N_C = 520$ samples, respectively. Each bootstrap symbol consequently consists of $N_A + N_B + N_C = 3072$ samples for an equivalent duration of 500 µs.

There shall be two variants of the time domain structure: CAB and BCA. The initial symbol of the bootstrap (i.e., bootstrap symbol 0), provided for sync detection, shall employ the CAB variant. The remaining bootstrap symbols (i.e., bootstrap symbol n where $1 \leq n < N_S$) shall conform to the BCA variant up to and including the bootstrap symbol that indicates field termination.

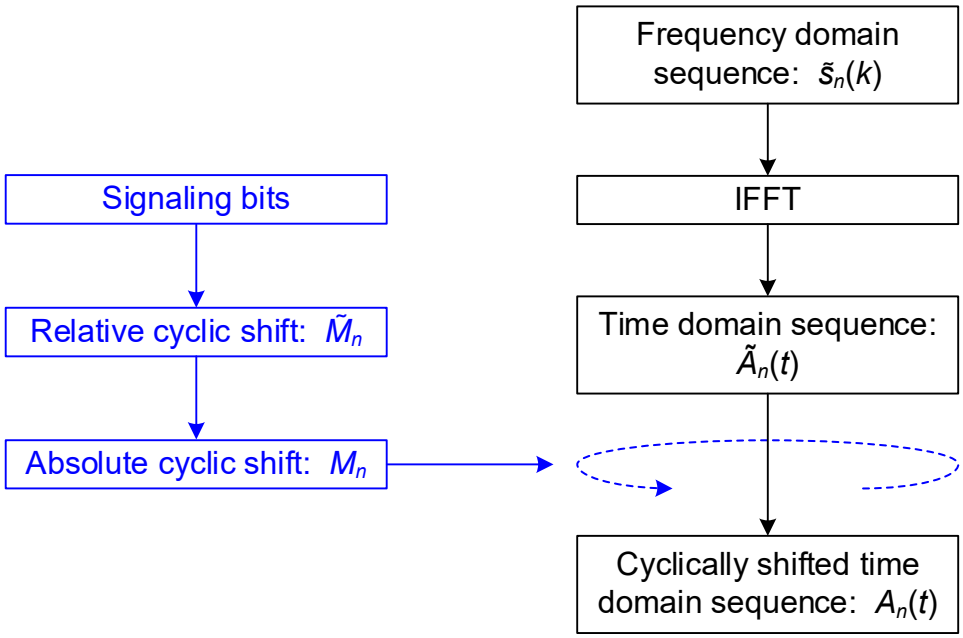

Figure 5.4 Generation of the cyclically shifted time domain sequence from the frequency domain sequence.

5.4.1 CAB Structure

The CAB time domain structure shall be as shown in Figure 5.5.

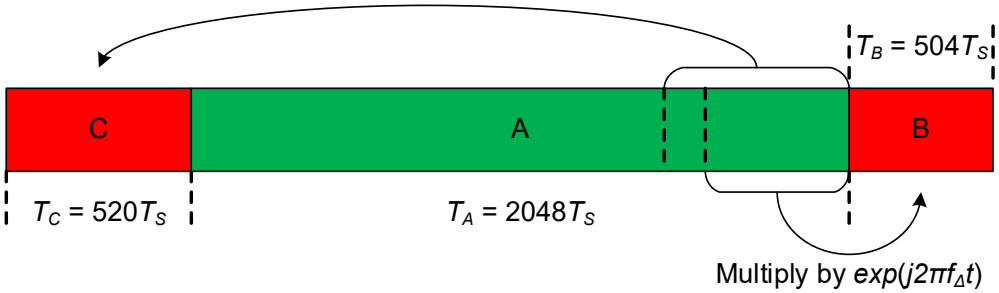

Figure 5.5 CAB time domain symbol structure.

For the CAB structure, part C shall be composed of the last $N_C = 520$ samples of part A, while part B shall be composed of the last $N_B = 504$ samples of part A with a frequency shift of $+f_\Delta$ and a phase shift of $e^{-j\pi}$ applied to the originating frequency domain sequence $\tilde{s}_n(k)$ used for calculating part A. The samples for part B can be taken as the negation of the last N_B samples of a cyclically shifted time domain sequence calculated as shown in Figure 5.4, where the input frequency domain sequence at the top of the block diagram is equal to $\tilde{s}_n(k)$ shifted one subcarrier position higher in frequency (i.e. $\hat{s}_n(k) = \tilde{s}_n\big((k-1+N_{FFT}) \bmod N_{FFT}\big)$, with $\hat{s}_n(k)$ being the input frequency domain sequence for generating the frequency-and-phase-shifted samples for part B). Alternatively, the frequency and phase shifts for generating the part B samples can be introduced in the time domain by multiplying the appropriately extracted samples from part A by $e^{j2\pi f_\Delta t}$ as shown in the following equation.

$$S_{CAB}^{n}(t) = \begin{cases} A_n(t + 1528T_S) & 0 \le t < 520T_S \\ A_n(t - 520T_S) & 520T_S \le t < 2568T_S \\ A_n(t - 1024T_S)e^{j2\pi f_\Delta t} & 2568T_S \le t < 3072T_S \\ 0 & \text{otherwise} \end{cases}$$

5.4.2 BCA Structure

The BCA time domain structure shall be as shown in Figure 5.6.

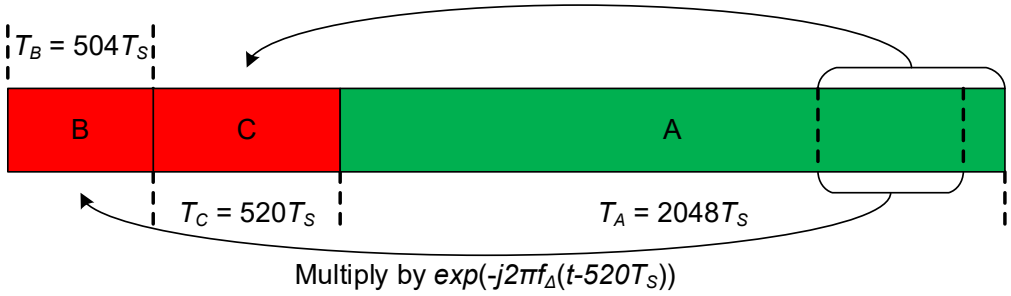

Figure 5.6 BCA time domain symbol structure.

For the BCA structure, part C shall be composed of the last $N_C = 520$ samples of part A, but part B shall be composed of the first N_B =504 samples of part C with a frequency shift of $-f_\Delta$ applied to the originating frequency domain sequence $\tilde{s}_n(k)$ used for calculating part A. In a similar fashion to that described in Section 5.4.1, the samples for part B can be taken as the last N_B samples of a cyclically shifted time domain sequence calculated as shown in Figure 5.4, where the input frequency domain sequence at the top of the block diagram is equal to $\tilde{s}_n(k)$ shifted one subcarrier position lower in frequency (i.e. $\hat{s}_n(k) = \tilde{s}_n\big((k+1) \bmod N_{FFT}\big)$, with $\hat{s}_n(k)$ being the input frequency domain sequence for generating the frequency-shifted samples for part B). The frequency shift for generating the part B samples can alternatively be introduced in the time domain by multiplying the appropriate samples from part A by $e^{-j2\pi f_\Delta t}$ with a constant time offset of $-520T_S$ being included to account for the correct extraction of the appropriate samples of part A, as shown in the following equation.

$$S_{BCA}^{n}(t) = \begin{cases} A_n(t + 1528T_S)e^{-j2\pi f_\Delta(t-520T_S)} & 0 \le t < 504T_S \\ A_n(t + 1024T_S) & 504T_S \le t < 1024T_S \\ A_n(t - 1024T_S) & 1024T_S \le t < 3072T_S \\ 0 & \text{otherwise} \end{cases}$$

Note that the samples for part B are taken from slightly different sections of part A for each of the CAB and BCA symbol structures.

6. BOOTSTRAP SIGNAL STRUCTURE

This section enumerates the signaling sets for specific versions of the general bootstrap structure described in Section 4.2, using the structure defined by the provisions of Section 5.

Each signaling set includes the configuration parameter values, a list of control information fields, and an assignment of those values and fields to specific signaling bits.

A bootstrap containing undefined signaling information (such as the use of reserved values) is expected to be discarded by the receiver.

6.1 Bootstrap Signaling for Major Version Zero (0)

This section and its subsections apply when bootstrap_major_version = 0.

The ZC sequence root (q), as specified in Section 5.2.1, shall be 137 when bootstrap_major_version = 0.

6.1.1 Signaling Minor Versions for Major Version Zero (0)

This section specifies how to signal minor versions when bootstrap_major_version = 0. The number of symbols (N_S) in the bootstrap set shall be greater than or equal to four (including the initial synchronization symbol) for all minor versions.

The initial register state of the pseudo-noise sequence generator for a given bootstrap minor version within bootstrap_major_version = 0 shall be set to a value from Table 6.1 to signal the corresponding bootstrap_minor_version that is in use.

Table 6.1 Initial Register State (pseudo-noise seed) of the Pseudo-Noise Sequence Generator for each respective bootstrap_minor_version

	$r_{init} = \{r_{l-1}, \ldots, r_0\}$	
Bootstrap Minor Version	**Binary**	**Hexadecimal**
0	0000 0001 1001 1101	0x019D
1	0000 0000 1110 1101	0x00ED
2	0000 0001 1110 1000	0x01E8
3	0000 0000 1110 1000	0x00E8
4	0000 0000 1111 1011	0x00FB
5	0000 0000 0010 0001	0x0021
6	0000 0000 0101 0100	0x0054
7	0000 0000 1110 1100	0x00EC

Note: The pseudo-noise seeds in Table 6.1 were generated by first considering a representative set of pseudo-noise seeds from the overall total set of possible pseudo-noise seeds. For each pseudo-noise seed, a metric value was calculated by normalizing the maximum cross-correlation between the frequency-domain sequence generated from the current pseudo-noise seed and the frequency-domain sequences generated from each of the other candidate pseudo-noise seeds with the maximum auto-correlation value for the frequency-domain sequence generated from the current pseudo-noise seed. The candidate pseudo-noise seeds with the minimum metric values were then selected as suitable initial register states for the pseudo-noise sequence generator due to exhibiting low cross-correlation.

6.1.1.1 Minor Version 0 Constraints and Signaling

When the value of r_{init} is set to 0x019D, indicating bootstrap_minor_version = 0, the number of symbols (N_S) in the bootstrap set shall be equal to four (including the initial synchronization symbol).

Bootstrap symbol 1 shall use the $N_b^1 = 8$ most significant signaling bits in order from most significant to least significant: $b_0^1\ b_1^1\ b_2^1\ b_3^1\ b_4^1\ b_5^1\ b_6^1\ b_7^1$. The syntax and semantics of the signaling fields for bootstrap symbol 1 shall be as given in Table 6.2 and the following text.

Table 6.2 Signaling Fields for Bootstrap Symbol 1

Syntax	No. of Bits	Format
bootstrap_symbol_1() {		
ea_wake_up_1	1	uimsbf
min_time_to_next	5	uimsbf
system_bandwidth	2	uimsbf
}		

The signaling fields for bootstrap symbol 1 are defined as follows.

ea_wake_up_1 –Bit 1 of emergency alert wake up field. Bit semantics are given in [2]

min_time_to_next – The minimum time interval to the next frame (B) that matches the same major and minor version number of the current frame (A), defined as the time period measured from the start of the bootstrap for frame A (referred to as bootstrap A) to the earliest possible occurrence of the start of the bootstrap for frame B (referred to as bootstrap B). Bootstrap B is guaranteed to lie within the time window beginning at the signaled minimum time interval value and ending at the next-higher minimum time interval value that could have been signaled. A **min_time_to_next** value of 31, corresponding to a minimum time value of 5700 ms, shall not be indicated. In the signal mapping formulas shown below, an example signaled value of X=10 would indicate that bootstrap B lies somewhere in a time window that begins 700 ms from the start of bootstrap A and ends 800 ms from the start of bootstrap A. The quantity is signaled via a sliding scale with increasing granularities as the signaled minimum time interval value increases.

Let X represent the 5-bit value that is signaled, and let T represent the minimum time interval in milliseconds to the next frame that matches the same version number as the current frame.

$$T = \begin{cases} T = 50 \times X + 50 & 0 \le X < 8 \\ T = 100 \times (X - 8) + 500 & 8 \le X < 16 \\ T = 200 \times (X - 16) + 1300 & 16 \le X < 24 \\ T = 400 \times (X - 24) + 2900 & 24 \le X < 32 \end{cases}$$

See also Table 6.3.

Table 6.3 Minimum Time Interval to Next Frame of the Same Major and Minor Version

Index	Bit Value	Minimum Time Interval (ms)
0	00000	50
1	00001	100
2	00010	150
3	00011	200
4	00100	250
5	00101	300
6	00110	350
7	00111	400
8	01000	500
9	01001	600
10	01010	700
11	01011	800
12	01100	900
13	01101	1000
14	01110	1100
15	01111	1200
16	10000	1300
17	10001	1500
18	10010	1700
19	10011	1900
20	10100	2100
21	10101	2300
22	10110	2500
23	10111	2700
24	11000	2900
25	11001	3300
26	11010	3700
27	11011	4100
28	11100	4500
29	11101	4900
30	11110	5300
31	11111	Not Applicable

system_bandwidth – Signals the system bandwidth used for the post-bootstrap portion of the current PHY layer frame. Values: 00 = 6 MHz, 01 = 7 MHz, 10 = 8 MHz, 11 = Greater than 8 MHz. The "Greater than 8 MHz" option facilitates future operation using a system bandwidth greater than 8 MHz, but is not intended to be used by the version described by the present signaling set. Receivers that are not provisioned to handle a system bandwidth greater than 8 MHz would not be expected to receive any frames where system_bandwidth = 11.

Bootstrap symbol 2 shall use the $N_b^2 = 8$ most significant signaling bits in order from most significant to least significant: $b_0^2\ b_1^2\ b_2^2\ b_3^2\ b_4^2\ b_5^2\ b_6^2\ b_7^2$. The syntax and semantics of signaling fields for bootstrap symbol 2 shall be as given in Table 6.4 and the following text.

Table 6.4 Signaling Fields for Bootstrap Symbol 2

Syntax	No. of Bits	Format
bootstrap_symbol_2() {		
ea_wake_up_2	1	uimsbf
bsr_coefficient	7	uimsbf
}		

The signaling fields for bootstrap symbol 2 are defined as follows.

ea_wake_up_2 – Bit 2 of emergency alert wake up field. Bit semantics are given in [2]

bsr_coefficient – Sample Rate Post-Bootstrap (of the current PHY Layer frame) = $(N + 16) \times 0.384$ MHz. *N* is the signaled value and shall be in the range from 0 to 80, inclusive. Values of 81 to 127 are reserved.

Bootstrap symbol 3 shall use the $N_b^3 = 8$ most significant signaling bits in order from most significant to least significant: $b_0^3\ b_1^3\ b_2^3\ b_3^3\ b_4^3\ b_5^3\ b_6^3 b_7^3$. The syntax and semantics of signaling fields for bootstrap symbol 3 shall be as given in Table 6.5 and the following text.

Table 6.5 Signaling Fields for Bootstrap Symbol 3

Syntax	No. of Bits	Format
bootstrap_symbol_3() {		
preamble_structure	8	uimsbf
}		

The signaling fields for bootstrap symbol 3 are defined as follows.

preamble_structure – This field establishes the capability to signal the structure of one or more RF symbols following the last bootstrap symbol. It is provided to enable such signaling by use of values defined by another Standard. Note: This Standard places no constraint on the contents of this field.

6.2 Bootstrap Signaling for Major Version One (1)

This section and its subsections apply when bootstrap_major_version = 1.

The ZC sequence root (q), as specified in Section 5.2.1, shall be 197 when bootstrap_major_version = 1.

6.2.1 Signaling Minor Versions for Major Version One (1)

This subsection specifies how to signal minor versions when bootstrap_major_version = 1. The number of symbols (N_S) in the bootstrap symbol set shall be greater than or equal to four (including the initial synchronization symbol) for all minor versions.

The initial register state of the pseudo-noise sequence generator for a given bootstrap minor version within bootstrap_major_version = 1 shall be set to a value from Table 6.6 to signal the corresponding bootstrap_minor_version that is in use.

Table 6.6 Initial Register State (pseudo-noise seed) of the Pseudo-Noise Sequence Generator for each respective bootstrap_minor_version

$r_{init} = \{r_{l-1}, \ldots, r_0\}$		
Bootstrap Minor Version	Binary	Hexadecimal
0	1111 0001 0001 0000	0xF110
1	0011 1101 0010 0001	0x3D21
2	1110 0101 0101 0000	0xE550
3	1011 1101 0100 1001	0xBD49
4	0010 0011 1100 1111	0x23CF
5	0000 1011 0101 0000	0x0B50
6	0011 1101 0011 1100	0x3D3C
7	1010 0010 0001 0110	0xA216

6.2.1.1 Minor Version 0 Constraints and Signaling

When the value of r_{init} is set to 0xF110, indicating bootstrap_minor_version = 0, the same constraints, signaling syntax, and meanings as specified in Section 6.1.1.1 shall apply, with minor semantic changes as specified below:

ea_wake_up_1 – Bit 1 of emergency alert wake up field. Bit semantics are defined by the document(s) establishing the corresponding receiver capability set (e.g., [2]).

system_bandwidth – Signals the nominal system bandwidth used for the post-bootstrap portion of the current physical layer frame. Values: 00 = 6 MHz, 01 = 7 MHz, 10 = 8 MHz, 11 = Greater than 8 MHz. Each nominal system bandwidth value may equivalently indicate the actual system bandwidth of the post-bootstrap portion, as defined by the document(s) establishing the corresponding receiver capability set. For example, the following value mapping can be used: 00 = 5.4 MHz, 01 = 6.3 MHz, 10 = 7.2 MHz, 11 = Greater than 7.2 MHz.

ea_wake_up_2 – Bit 2 of emergency alert wake up field. Bit semantics are defined by the document(s) establishing the corresponding receiver capability set (e.g., [2]).

bsr_coefficient – Sample Rate Post-Bootstrap (of the current physical Layer frame) = $(N + 10) \times 0.384$ MHz. N is the signaled value and shall be in the range from 0 to 86, inclusive. Values of 87 to 127 are reserved.

preamble_structure – This field establishes the capability to signal the structure of one or more RF symbols following the last bootstrap symbol. It is provided to enable such signaling by use of values defined by another Standard.

> Note: This Standard does not constrain the contents of this field. This field may therefore be interpreted according to the Standard that specifies the post-bootstrap portion of the signal. For example, this field may indicate the configuration of a channel acquisition signal (CAS): Bandwidth, cyclic prefix length option, and frequency-domain placement (centered, upper-aligned, lower-aligned). As another example, this field may indicate the preamble signal configuration defined in [3].

6.3 Future Major Versions

This section lists the Zadoff-Chu root (q) values that are permitted to be used to indicate future bootstrap_major_version values. The Zadoff-Chu root (q) values within the ranges 0 .. 136, 138 .. 196, and 198 .. 1498 shall be Reserved.

7. MULTIPLEXING OF DIFFERENT SIGNAL TYPES WITH BOOTSTRAP PREFIXING

7.1 General Description

The bootstrap is defined to provide a universal entry point for receivers by enabling identification of the signal type applicable to the waveform that immediately follows it. A signal type herein denotes the format of the post-bootstrap waveform, corresponding to a digital transmission technology or a combination of technology pieces, with ATSC 3.0 specified in [3] as one example. This Section describes how multiple signal types can be multiplexed within a single RF channel by time-division multiplexing and specifies the relationship between bootstrap versioning and identification of the signal type associated with each time-multiplexed frame interval.

Such multiplexing derives from the non-backward-compatible extensibility mechanism described in Section 3.5. The signal type associated with a physical layer frame is identified by the bootstrap prefixed to that frame, specifically by the bootstrap major and minor version value pair. The bootstrap-enabled identification in this Standard accommodates a broad range of digital transmission technologies: A signal type may be specified either within ATSC standards or by documents defined by other organizations.

In time-multiplexed operations with multiple signal types, a physical layer frame is referred to as an Alternative Physical Layer Frame (APLF). APLFs are further classified as follows:

- ATSC-native APLF: An APLF that has a signal type specified by ATSC documents.
- Externally-defined APLF: An APLF that has a signal type specified by a document prepared by an organization other than ATSC.

7.2 Frame and Bootstrap Positioning

Each APLF begins with a bootstrap, followed by a post-bootstrap waveform corresponding to the signal type identified by the major and minor version value pairs carried by that bootstrap. Each consecutive APLF is positioned in time so that start of its bootstrap immediately follows the end of the preceding APLF's post-bootstrap waveform. When adjacent APLFs employ different baseband sampling rates, the next APLF's bootstrap occurs within $\max(T_A, T_B)$, where T_A and T_B denote the baseband sample periods of the adjacent APLF post-bootstrap waveforms.

The baseband sampling rate of each specific signal type follows specifications provided by related documentations, which specifications are defined outside the scope of this Standard.

Figure 7.1 shows an example of time-multiplexed APLFs incorporating different signal types.

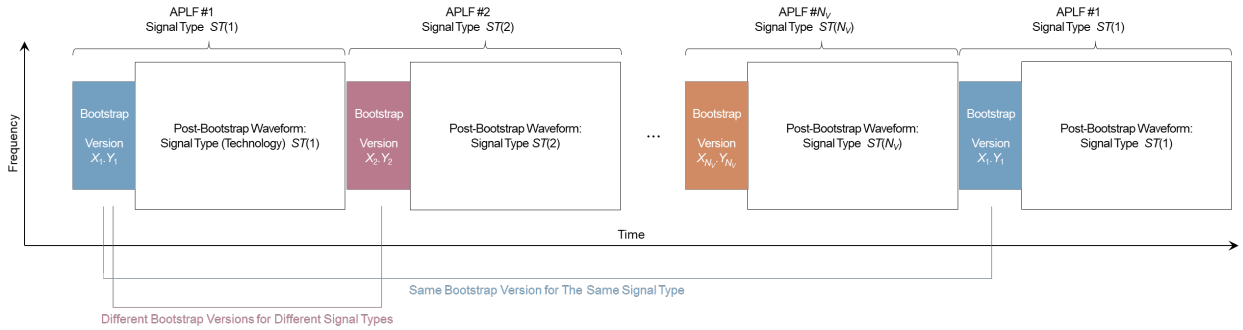

Figure 7.1 Example of time-multiplexed APLFs carrying N_V signal types.

7.3 Identification of Signal Types by Bootstrap Versioning

This section specifies the signal type identification mechanism outlined in Section 4.2, using the versioning structure defined in this Standard. A defined set of major/minor version value pairs is used to identify the signal type associated with each APLF. ATSC 3.0 is identified by Bootstrap version 0.0 (bootstrap_major_version = 0; bootstrap_minor_version = 0) as specified in [3].

Consistent with the extensibility mechanism described in Section 3.5, changes in the major version value correspond to major changes in signal type (e.g., a different underlying digital transmission technology), while changes in the minor version value correspond to evolution, extension, or reconfiguration within the same major signal type.

A bootstrap version allocation table is configured to map each bootstrap version to an APLF signal type identifier. Table 7.1 provides an example using two bootstrap versions defined in Sections 6.1 and 6.2.

Table 7.1 Bootstrap Version Value Allocation Example for APLF Identification (Example Using Two Bootstrap Versions)

bootstrap_major_version Value	bootstrap_minor_version Value	Signal Type of APLF Post-Bootstrap	Notes
0	0	Signal Type A	Note 1
1	0	Signal Type B	

Note 1: The signal type specified in [3] (ATSC 3.0) uses Bootstrap version 0.0 (bootstrap_major_version = 0; bootstrap_minor_version = 0) when employed in an APLF.

Beyond the bootstrap versions 0.0 and 1.0, the constraints and signaling of which are specified in this version of the Standard, the allocation extends to use future major and minor versions. Table 7.2 provides an example.

Table 7.2 Bootstrap Version Value Allocation Example for APLF Identification (Example Using Five Bootstrap Versions)

bootstrap_major_version Value	bootstrap_minor_version Value	Signal Type of APLF Post-Bootstrap	Notes
0	0	Signal Type A	Note 1
1	0	Signal Type B1	
1	1	Signal Type B2	
2	0	Signal Type C	
3	0	Signal Type D	

Note 1: The signal type specified in [3] (ATSC 3.0) uses Bootstrap version 0.0 (bootstrap_major_version = 0; bootstrap_minor_version = 0) when employed in an APLF.

Annex A: Example Method of Gray Code De-mapping at Receiver

A.1 GRAY CODE DE-MAPPING AT RECEIVER

Section 5.3.2 specifies a Gray code mapping of signaling bit values to a corresponding relative cyclic shift value for transmitter operation. This Annex describes an example method of de-mapping at the receiver from an estimated relative cyclic shift to estimated values of the corresponding signaling bits.

Let $\widehat{M}_n$ $(0 \leq \widehat{M}_n < N_{FFT})$ represent an estimated cyclic shift at the receiver for the n-th bootstrap symbol $(1 \leq n < N_S)$ relative to the estimated cyclic shift for the previous bootstrap symbol. Let $\widehat{M}_n$ be represented in binary form as $\widehat{m}_{10}^n\ \widehat{m}_9^n \ldots \widehat{m}_1^n\ \widehat{m}_0^n$. The signaling bit values expected by the receiver can be estimated as follows, where $\oplus$ represents the logical exclusive OR operator.

$$\hat{b}_i^n = \begin{cases} \widehat{m}_{10}^n & i = 0 \\ \widehat{m}_{11-i}^n \oplus \widehat{m}_{10-i}^n & 1 \leq i < N_b^n \\ 0 & N_b^n \leq i < 11 \end{cases}$$

A receiver is expected to decode only the N_b^n signaling bits for which it has been provisioned, even when the receiver is decoding a bootstrap symbol belonging to a later minor version within the same major version.

Annex B: Bootstrap Signaling Bit Robustness and Other Characteristics

B.1 GRAY CODE MAPPING EXAMPLES

One method for illustrating and investigating the Gray code mapping of bootstrap signaling bits to a relative cyclic shift (RCS) value, as described in Section 5.3.2, is to use representative examples.

B.1.1 Gray Code Mapping Example With Four Signaling Bits

In the first example, there are $N_b = 4$ signaling bits ($b_0 b_1 b_2 b_3$, from most significant to least significant) in the bootstrap symbol.

Table B.1.1 shows the mapping from all possible values of the four signaling bits to corresponding relative cyclic shifts, using the procedure described in Section 5.3.2. The four most-significant bits ($m_{10} m_9 m_8 m_7$) of the relative cyclic shift are calculated as a function of the signaling bit values, while the seven least-significant bits ($m_6 m_5 m_4 m_3 m_2 m_1 m_0$) of the relative cyclic shift remain constant for this particular example.

Table B.1.1 Example Mapping of Four Signaling Bits to Relative Cyclic Shifts

Signaling Bits (Binary) $b_0 b_1 b_2 b_3$	Relative Cyclic Shift (Binary) ($m_{10} \ldots m_0$)	Relative Cyclic Shift (Decimal) ($\tilde{M}$)
0 0 0 0	0 0 0 0 1 0 0 0 0 0 0	64
0 0 0 1	0 0 0 1 1 0 0 0 0 0 0	192
0 0 1 0	0 0 1 1 1 0 0 0 0 0 0	448
0 0 1 1	0 0 1 0 1 0 0 0 0 0 0	320
0 1 0 0	0 1 1 1 1 0 0 0 0 0 0	960
0 1 0 1	0 1 1 0 1 0 0 0 0 0 0	832
0 1 1 0	0 1 0 0 1 0 0 0 0 0 0	576
0 1 1 1	0 1 0 1 1 0 0 0 0 0 0	704
1 0 0 0	1 1 1 1 1 0 0 0 0 0 0	1984
1 0 0 1	1 1 1 0 1 0 0 0 0 0 0	1856
1 0 1 0	1 1 0 0 1 0 0 0 0 0 0	1600
1 0 1 1	1 1 0 1 1 0 0 0 0 0 0	1728
1 1 0 0	1 0 0 0 1 0 0 0 0 0 0	1088
1 1 0 1	1 0 0 1 1 0 0 0 0 0 0	1216
1 1 1 0	1 0 1 1 1 0 0 0 0 0 0	1472
1 1 1 1	1 0 1 0 1 0 0 0 0 0 0	1344

Table B.1.2 shows the mapping from relative cyclic shift values back to signaling bit values, using the information from Table B.1.1. The relative cyclic shifts in Table B.1.2 have been sorted into ascending order. As can be seen, the distance between adjacent relative cyclic shifts in this example is 128, and in this case each relative cyclic shift can be incorrectly estimated at the receiver with a tolerance of up to ± 63 without causing an error in the recovery of the signaling bit values.

In general, when N_b signaling bits are in use within a particular bootstrap symbol, the distance between adjacent relative cyclic shifts will be 2^{11-N_b} and the maximum error tolerance in the relative cyclic shift estimation at a receiver will be $\pm(2^{10-N_b}-1)$. That is, the relative cyclic shift signaled by a bootstrap symbol can be incorrectly estimated at a receiver by up to $\pm(2^{10-N_b}-1)$, while still allowing all of the correct signaling bit values for that bootstrap symbol to be recovered.

When the number of signaling bits is $N_b = 7$, the distance between adjacent relative cyclic shifts will be 16 and the maximum error tolerance in the relative cyclic shift estimations at a receiver will be ± 7. Similarly, when the number of signaling bits is $N_b = 8$, the distance between adjacent relative cyclic shifts will be 8 and the maximum error tolerance in the relative cyclic shift estimations at a receiver will be ± 3.

Finally, examination of the signaling bit values in the rightmost column of Table B.1.2 (which have been ordered by their corresponding relative cyclic shift values) clearly illustrates the Gray code mapping, as only one bit position at a time changes value from one row to the next.

Table B.1.2 Example Mapping of Relative Cyclic Shifts to Four Signaling Bits

Relative Cyclic Shift (Decimal)	Relative Cyclic Shift (Binary) ($m_{10} \ldots m_0$)	Signaling Bits (Binary) ($b_0 b_1 b_2 b_3$)
64	0 0 0 0 1 0 0 0 0 0 0	0 0 0 0
192	0 0 0 1 1 0 0 0 0 0 0	0 0 0 1
320	0 0 1 0 1 0 0 0 0 0 0	0 0 1 1
448	0 0 1 1 1 0 0 0 0 0 0	0 0 1 0
576	0 1 0 0 1 0 0 0 0 0 0	0 1 1 0
704	0 1 0 1 1 0 0 0 0 0 0	0 1 1 1
832	0 1 1 0 1 0 0 0 0 0 0	0 1 0 1
960	0 1 1 1 1 0 0 0 0 0 0	0 1 0 0
1088	1 0 0 0 1 0 0 0 0 0 0	1 1 0 0
1216	1 0 0 1 1 0 0 0 0 0 0	1 1 0 1
1344	1 0 1 0 1 0 0 0 0 0 0	1 1 1 1
1472	1 0 1 1 1 0 0 0 0 0 0	1 1 1 0
1600	1 1 0 0 1 0 0 0 0 0 0	1 0 1 0
1728	1 1 0 1 1 0 0 0 0 0 0	1 0 1 1
1856	1 1 1 0 1 0 0 0 0 0 0	1 0 0 1
1984	1 1 1 1 1 0 0 0 0 0 0	1 0 0 0

Figure B.**1.1** shows the values of the four signaling bits as a function of the estimated relative cyclic shift value in a graphical form. This diagram uses the information from Table B.1.2.

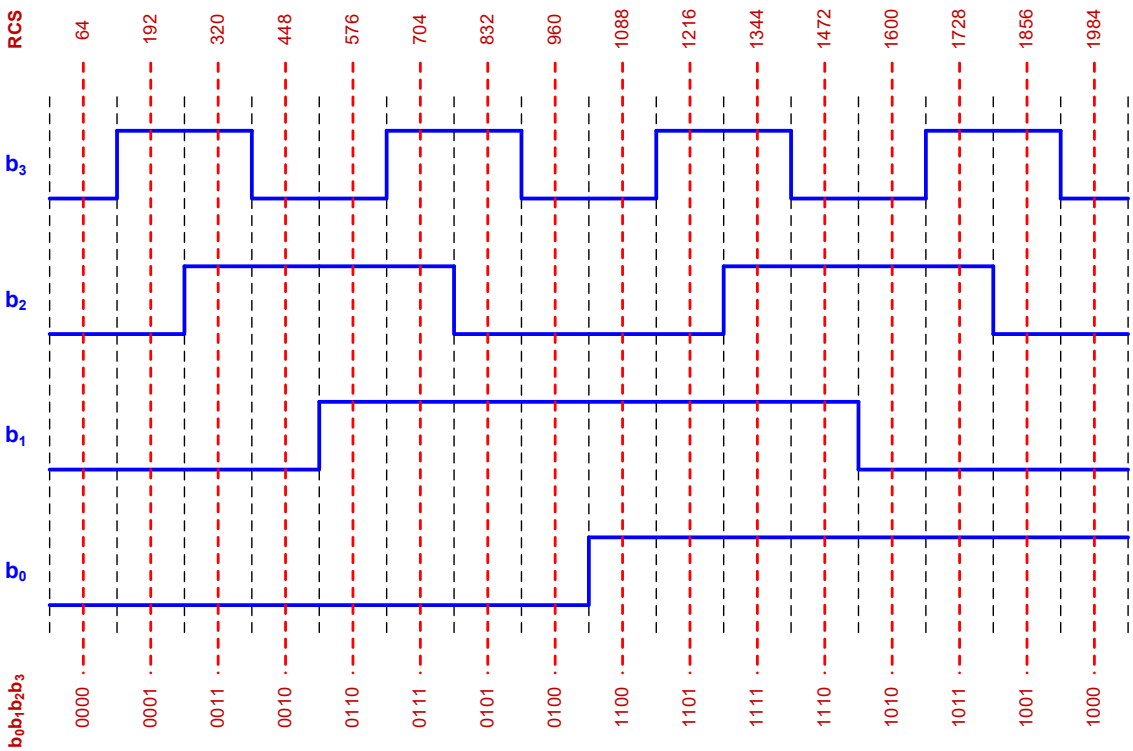

Figure B.1.1 Example Gray code mapping with four signaling bits

B.1.2 Gray Code Mapping Example with Three Signaling Bits

In the second example, there are $N_b = 3$ signaling bits ($b_0 b_1 b_2$, from most significant to least significant) in the bootstrap symbol.

Table B.1.3 shows the mapping from all possible values of the three signaling bits to corresponding relative cyclic shifts, using the procedure described in Section 5.3.2. The three most-significant bits ($m_{10} m_9 m_8$) of the relative cyclic shift are calculated as a function of the signaling bit values, while the eight least-significant bits ($m_7 m_6 m_5 m_4 m_3 m_2 m_1 m_0$) of the relative cyclic shift remain constant for this particular example.

Table B.1.3 Example Mapping of Three Signaling Bits to Relative Cyclic Shifts

Signaling Bits (Binary) $b_0 b_1 b_2$	Relative Cyclic Shift (Binary) ($m_{10} \ldots m_0$)	Relative Cyclic Shift (Decimal) ($\tilde{M}$)
0 0 0	0 0 0 1 0 0 0 0 0 0 0	128
0 0 1	0 0 1 1 0 0 0 0 0 0 0	384
0 1 0	0 1 1 1 0 0 0 0 0 0 0	896
0 1 1	0 1 0 1 0 0 0 0 0 0 0	640
1 0 0	1 1 1 1 0 0 0 0 0 0 0	1920
1 0 1	1 1 0 1 0 0 0 0 0 0 0	1664
1 1 0	1 0 0 1 0 0 0 0 0 0 0	1152
1 1 1	1 0 1 1 0 0 0 0 0 0 0	1408

Table B.1.4 shows the mapping from relative cyclic shift values back to signaling bit values, using the information from Table B.1.3. The relative cyclic shifts in Table B.1.4 have been sorted into ascending order. As can be seen, the distance between adjacent relative cyclic shifts in this

example is 256, and in this case each relative cyclic shift can be incorrectly estimated at the receiver with a tolerance of up to ± 127 without causing an error in the recovery of the signaling bit values.

Table B.1.4: Example Mapping of Relative Cyclic Shifts to Three Signaling Bits

Relative Cyclic Shift (Decimal)	Relative Cyclic Shift (Binary) ($m_{10} \ldots m_0$)	Signaling Bits (Binary) ($b_0 b_1 b_2 b_3$)
128	0 0 0 1 0 0 0 0 0 0	0 0 0
384	0 0 1 1 0 0 0 0 0 0	0 0 1
640	0 1 0 1 0 0 0 0 0 0	0 1 1
896	0 1 1 1 0 0 0 0 0 0	0 1 0
1152	1 0 0 1 0 0 0 0 0 0	1 1 0
1408	1 0 1 1 0 0 0 0 0 0	1 1 1
1664	1 1 0 1 0 0 0 0 0 0	1 0 1
1920	1 1 1 1 0 0 0 0 0 0	1 0 0

Figure B.1.2 shows the values of the three signaling bits as a function of the estimated relative cyclic shift value in a graphical form. This diagram uses the information from Table B.1.4.

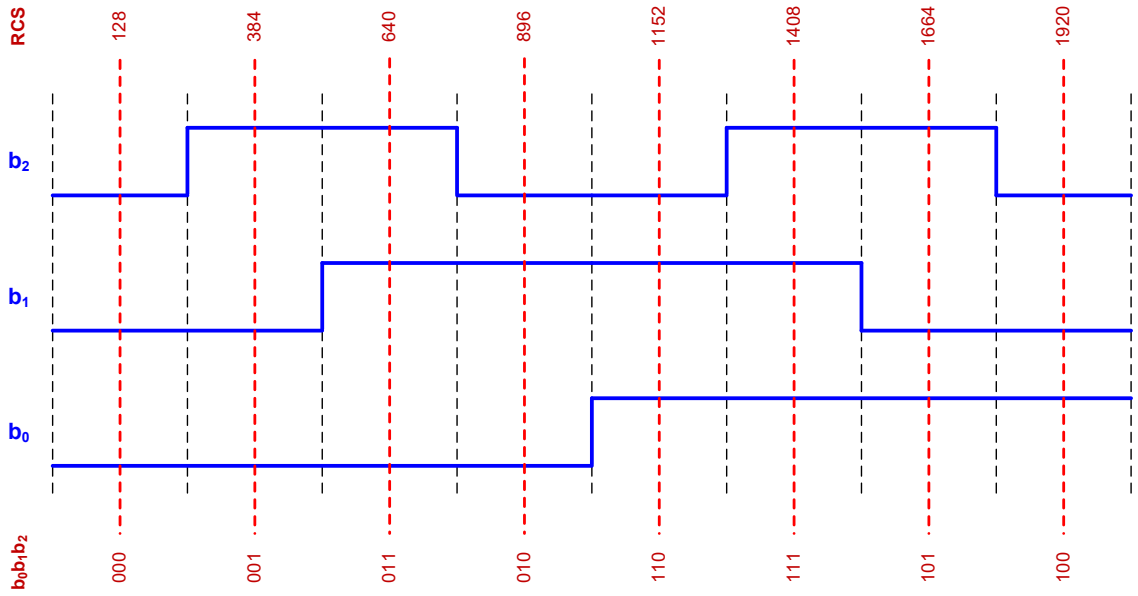

Figure B.1.2: Example Gray code mapping with three signaling bits

B.2 ADDITIONAL OBSERVATIONS ON BOOTSTRAP SIGNALING BITS

One key point to notice from Figure B.1.1 and Figure B.1.2 is that the mapping from a particular relative cyclic shift value to signaling bit values $b_0 b_1 b_2$ is exactly the same for the cases of four signaling bits (Figure B.1.1) and three signaling bits (Figure B.1.2), respectively. This implies that regardless of the number of signaling bits carried by a bootstrap symbol, an individual signaling bit value for a particular signaling bit index will always be the same for a given relative cyclic shift value. For example, b_0 will always be 0 if the relative cyclic shift is in the range $0 \leq \text{RCS} \leq 1023$ or 1 if the relative cyclic shift is in the range $1024 \leq \text{RCS} \leq 2047$, and so on for the other signaling

bit indices, regardless of how many signaling bits are carried by the corresponding bootstrap symbol.

Another robustness consideration is that different signaling bits have different levels of robustness based on the signaling bit index within a bootstrap symbol, with b_k being more robust than b_m when $k < m$. As an illustration of this property, consider the example shown in Figure B.1.2. If an error of ± 128 in the estimation of the relative cyclic shift is made at a receiver, then the value of b_2 will be incorrectly estimated 100% of the time. Conversely, if the same estimation error (± 128) of the relative cyclic shift is incurred and all of the eight possible relative cyclic shifts at the transmitter are equally probable, then the value of b_0 will be incorrectly estimated only 25% of the time.

Coupling the finding of the preceding paragraph with the earlier observation of the maximum error tolerance in the estimation of the relative cyclic shift at a receiver results in the following. When N_b signaling bits are in use within a particular bootstrap symbol, the value of signaling bit b_k will be incorrectly estimated $100/2^{N_b-k-1}$ % of the time when an error of $\pm 2^{10-N_b}$ is made in the relative cyclic shift estimation at the receiver.

B.3 IMPACT OF ERRORS IN THE ESTIMATION OF BOOTSTRAP SIGNALING BIT VALUES AT A RECEIVER

Although different signaling bits within a bootstrap symbol will have different relative levels of robustness, a single bit error when estimating the bootstrap signaling bit values at a receiver will likely cause problems with either decoding the immediately following frame or correctly locating the time window containing the next bootstrap of the same major/minor version. A brief discussion of the effect of estimating an incorrect value for each of the bootstrap signaling fields follows.

- **ea_wake_up_1** and **ea_wake_up_2**
  - The values of these two signaling bits can indicate one of four possible states. One of these states represents a "negative" state where no emergency alert information is available. The other three states represent "positive" states where some form of emergency alert information is available.
  - If **ea_wake_up_1** and **ea_wake_up_2** currently indicate that no emergency alert information is available (i.e. currently in the negative state):
    - A false positive condition would result if **ea_wake_up_1** and/or **ea_wake_up_2** were decoded incorrectly. In this situation a receiver would incorrectly conclude that emergency alert information was available. The receiver would search for that emergency alert information, but would be unable to find it. If the receiver then correctly decoded **ea_wake_up_1** and **ea_wake_up_2** in subsequent bootstraps, the receiver would likely conclude that it had encountered a false positive.
  - If **ea_wake_up_1** and **ea_wake_up_2** currently indicate that emergency alert information is available (i.e. currently in a positive state):
    - A false negative condition would result if **ea_wake_up_1** and **ea_wake_up_2** were decoded incorrectly to indicate that emergency alert information was not available (i.e. that the current emergency alert was over). In this situation a receiver would incorrectly conclude that the current emergency alert was over. If the receiver then correctly decoded **ea_wake_up_1** and **ea_wake_up_2** in subsequent bootstraps, the receiver would likely conclude that a new

emergency alert was beginning. Higher layers would be responsible for the exact handling of this situation, but in the worst case the end user would receive a duplicate warning of a previously-dismissed emergency alert.

- A false positive condition would result if **ea_wake_up_1** and **ea_wake_up_2** were decoded incorrectly to indicate that a different set of emergency alert information was available. In this situation a receiver would likely attempt to receive that "new" emergency alert information and present it to the end user. If the receiver then correctly decoded **ea_wake_up_1** and **ea_wake_up_2** in subsequent bootstraps, the receiver would likely conclude that further "new" emergency alert information was available, and would thus receive that "new" information and present it to the end user. In the worst case, the end user would receive one or two duplicate warnings of a previously-dismissed emergency alert.

- The other situation that could arise is if a receiver incorrectly decoded **ea_wake_up_1** and **ea_wake_up_2** in the exact same frame as a state change in the values of **ea_wake_up_1** and **ea_wake_up_2** occurred. Note that the incorrect decoding of **ea_wake_up_1** and **ea_wake_up_2** would typically be a very rare event, and hence this particular situation (an incorrect decoding coinciding with a state change) would be an extremely rare (i.e. highly improbable) event.
  - If the receiver incorrectly decoded **ea_wake_up_1** and **ea_wake_up_2** to interpret either a negative state (i.e. no emergency alert information is present) or a continuation of the previous positive state (i.e. **ea_wake_up_1** and **ea_wake_up_2** were incorrectly decoded to be equal to their previous values), but **ea_wake_up_1** and **ea_wake_up_2** were decoded correctly in a subsequent bootstrap, then there would be a slight delay (e.g. equal to one frame length) in presenting the new emergency alert to the end user.
  - If the receiver incorrectly decoded **ea_wake_up_1** and **ea_wake_up_2** to interpret a positive state that was different from the previous positive state, then the receiver would decode and present the new emergency alert information (which is the desired action), despite the incorrect decoding.

- **min_time_to_next**
  - Incorrectly recovering the minimum time interval to the next bootstrap of the same major/minor version might result in a receiver using an incorrect time window to search for the next bootstrap of the same major/minor version. If the frame (e.g. preamble) contains additional control signaling indicating when the next bootstrap or frame begins and the receiver actively receives at least the preamble of the frame, then this error would be recoverable. Alternatively, the receiver might need to resync and perform a new initial scan for a bootstrap.
- **system_bandwidth**
  - Incorrectly recovering the **system_bandwidth** value would result in a receiver being unable to correctly decode the immediately following frame. An intelligent receiver might be able to recover the frame if the receiver had cached previously-received **system_bandwidth** values for bootstraps of the same major/minor version for the same RF channel, since it is unlikely that this information would change dynamically.
- **bsr_coefficient**

- Incorrectly recovering the **bsr_coefficient** would result in a receiver being unable to correctly decode the immediately following frame.

- **preamble_structure**
  - Incorrectly recovering the **preamble_structure** would result in a receiver being unable to correctly decode the preamble of the immediately following frame, and hence the payload contents of that same frame would also be non-recoverable.

– End of document –