**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **D4 Comments Resolution Based PHY-VI PHY Specification** | |
| Date Submitted | September, 2017 | |
| Source | Jaesang Cha (SNUST), Vinayagam Mariappan (SNUST), Sangsung Choi (ETRI), Hyeongho Lee (ETRI), Chanhyung Chung (RAPA), Ilkyoo Lee (Kongju Nat’ Univ.), Sooyoung Chang (CSUS) | Voice: [ ] Fax: [ ] E-mail: [chajs@seoultech.ac.kr]1 |
| Re: | Draft D4 Comment Resolution based PHY-VI PHY Specification Revision | |
| Abstract | Details of Resolutions regarding to the submitted Comments on D4 are suggested for PHY-VI PHY Specification Revision. The PHY VI is designed to operate on the application services like LED ID, LiFi/CamCom, Digital Signage with Advertisement Information etc. | |
| Purpose | Draft D4 Comments Resolutions and Editorial Revision. | |
| Notice | This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. | |
| Release | The contributor acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

# **1. PHY VI SPECIFICATIONS**

# **15. PHY VI Specifications**

# **15.2. VTASC Specifications**

The VTASC works with variable transparency levels, sizes, shapes, and colors of the patterns. The VTASC PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.2.1 VTASC Reference Architecture**

The reference PHY architecture for VTASC is illustrated in Figure 215. The data embedded on visual frame by overlaying visual patterns in defined displays visual area. After spread spectrum, data is transformed into VTASC coded patterns according to the mapping rule on the transparency levels, sizes, shapes, and colors by the coding pattern blocks.

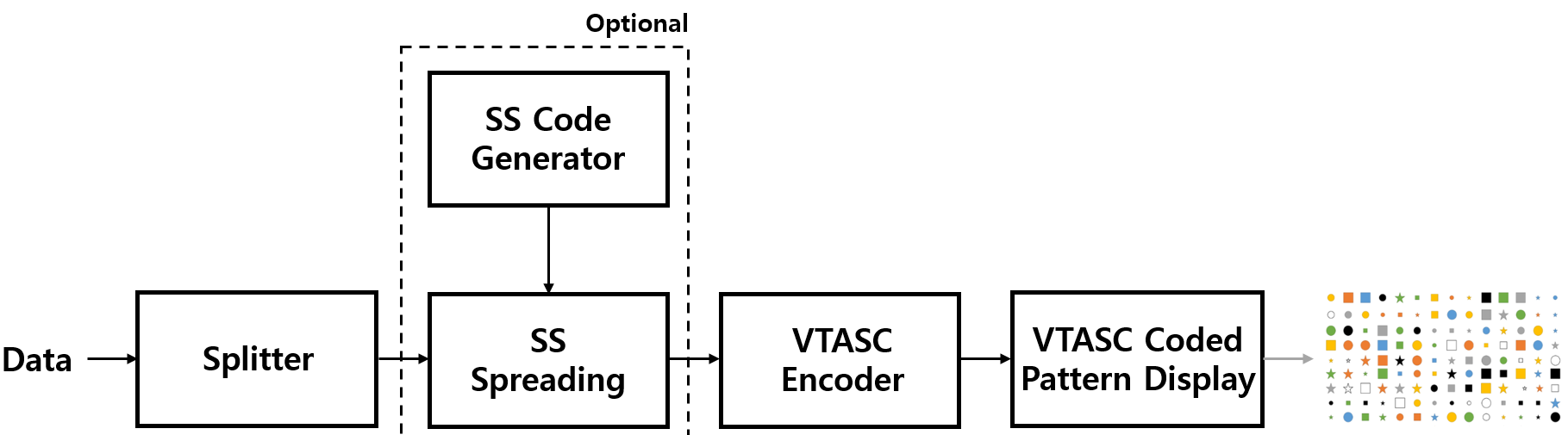
****

Figure 215 – Reference architecture for VTASC PHY System

The spread spectrum used with VTASC to have effective asynchronous, distance adaptive scalable data rate controlled OWC. The VTASC is used for enhanced display to camera communication in the real-time application usage scenario. The VTASC specific working features are given in Annex I.3.1.

The receiver specific information for VTASC Data Decoder is given in Annex J.9.

**15.2.2 Spread Spectrum**

The spread spectrum used with VTASC, SS2DC, and IDE based display to camera OWC to have effective asynchronous, distance adaptive scalable data rate controlled communication. The display to camera communication adopted the binary zero-correlation duration (ZCD) code sequences as an optical spread code with the spreading code length. The initial basic matrix G used to generate binary ZCD is defined as,

1 1 1 -1

1 1 -1 1

1 -1 1 1

-1 1 1 1

G =

The binary ZCD sequences constructed cyclically from the chip-shift operation using family of codes {SN(a),SN(b)} shown in (xyz). Any row of G or –G is denoted as S4(a) = (S0(a), S1(a), S2(a), S3(a)), S4(b) = (S0(b), S1(b), S2(b), S3(b)) is generated from S4(a), where Sq(b) = Sq(a) (q = 0, 1, 2, 3).

{SN(a),SN(b),TΔ[SN(a)],TΔ[SN(b)], T2Δ[SN(a)],T2Δ[SN(b)] ,…,T(k-1)Δ[SN(a)],T(k-1)Δ[SN(b)], TkΔ[SN(a)],TkΔ[SN(b)] } ----------------------------------------------------- (xyz)

Where,

* SN(a),SN(b) are the pair of family sequence and N is family size
* Tl is chip shift operator, which shifts a sequence cyclically to the left by l chips
* Δ is a chip-shift increment and k is a the maximum number of chips-shifts for a sequence and Δ and k should satisfy |(k+1) Δ| ≤ |N/4 + 1| ,Δ is a positive and k a non-negative integer

The binary ZCD based optical spreading code used for a specific data rate or distance transmission is defined in Table 147.

Table 147 – Optical Spreading Code for different data rate or receiver distance

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Spread Sequence** | **Spreading Code** | | | | | | | | **Distance (meters)** | **Frame Refresh Rate (Hz)** |
| SC1#00 | 1 | 1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 30 |
| SC1#01 | 1 | -1 | 1 | 1 | -1 | 1 | 1 | 1 |
| SC2#00 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | -1 | 2 | 30 |
| SC2#01 | 1 | 1 | 1 | -1 | 1 | 1 | -1 | 1 |
| SC3#00 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | 1 | 3 | 30 |
| SC3#01 | -1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 |
| SC4#00 | -1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | above 4 | 30 |
| SC4#01 | -1 | -1 | -1 | 1 | -1 | -1 | 1 | -1 |

There are four set of code used for scalable and distance adaptive transmission (see in 15.2.6) and each set of code coupled with pair of codes for synchronization (see in 15.2.5). The data spreading with spreading factor 1 is illustrated in Figure 216.

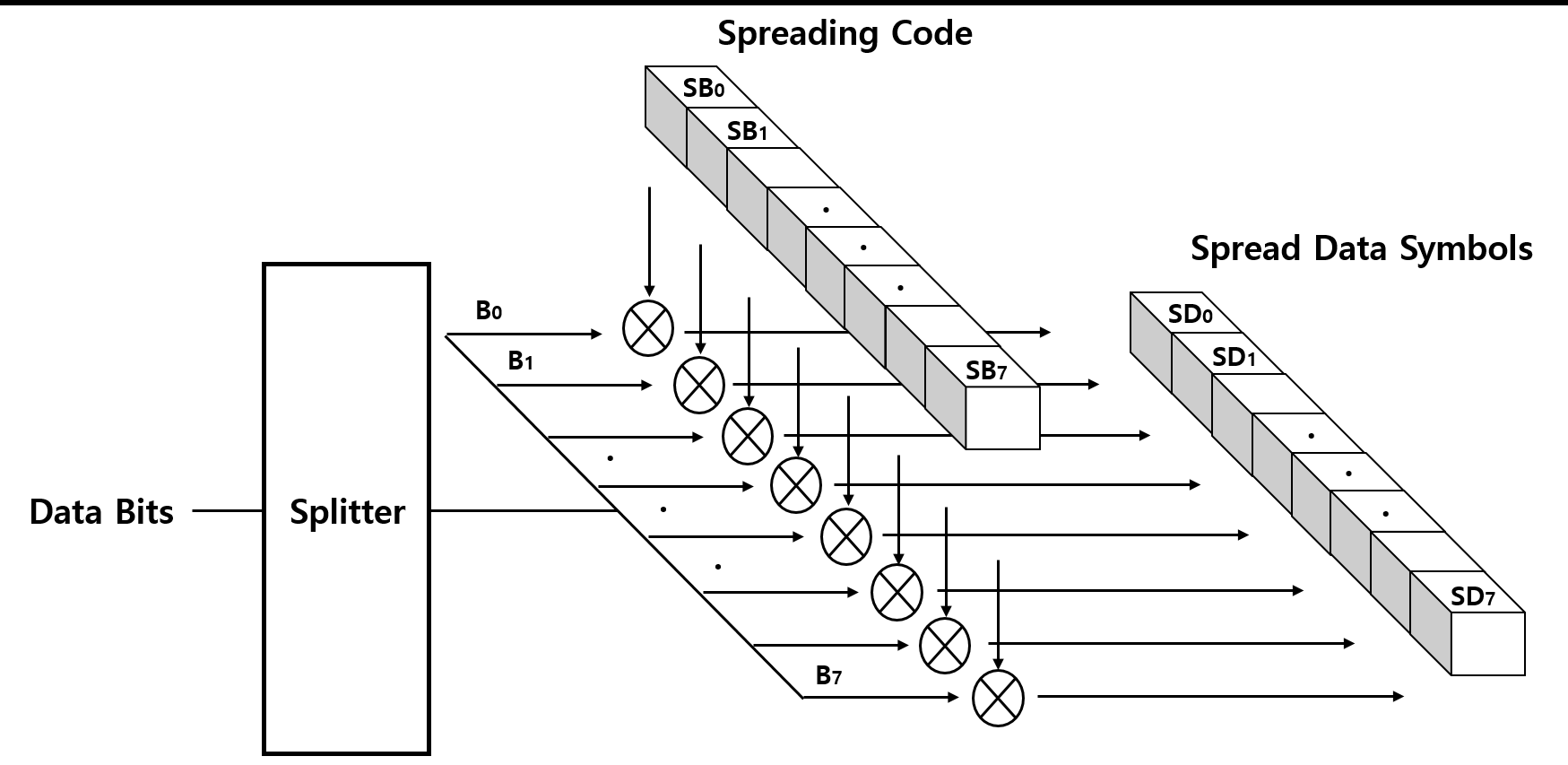


Figure 216 - SS Spreading Example

The SS code length configurable over PHY PIB attributes PhySSCode1Len, PhySSCode2Len, PhySSCode3Len, PhySSCode4Len and SS code pair is configurable over the PHY PIB attributesPhySSCode1FP00,PhySSCode1FP01,PhySSCode2FP00,PhySSCode2FP01,PhySSCode3FP00,PhySSCode3FP01,PhySSCode4FP00,PhySSCode4FP01.

**15.2.3 VTASC Code Design**

VTASC is a modulation scheme for visible-light communication involving single or multiple screens with variable transparency levels, sizes, shape models, and colors. VTASC enhances the OWC system performance with improved OWC throughput by increasing the bit per symbol rate, and avoiding the single color interference.

The VTASC is coded by T (Transparency level) / A (Amplitude nothing but block size) / S (Shapes) / C (Colors) State as described in the Figure 217.



Figure 217 - VTASC Code Design

The number of code levels in the VTASC modulation is (TxAxSxC) with two transparency levels, four block sizes, four shape models, and eight colors is 256 = 28 and this makes place to code 8 bit symbol with two levels of transparency, four size of blocks, four models of shape, and eight colors. The shape model design shown in Figure NEW1.

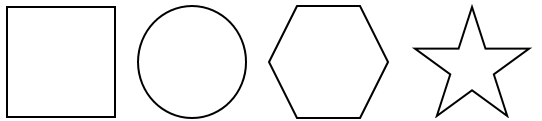


Figure NEW1 – VTASC Shape Models Design

In the VTASC coded screen symbol, the shapes inside the block pixel region is equally spaced in coding region. The coded region background color shall be white. The zero padded VTASC coded pattern generated if the available number of data bits is less than the coding region pattern mapping. The VTASC coded example model is given Figure 218.

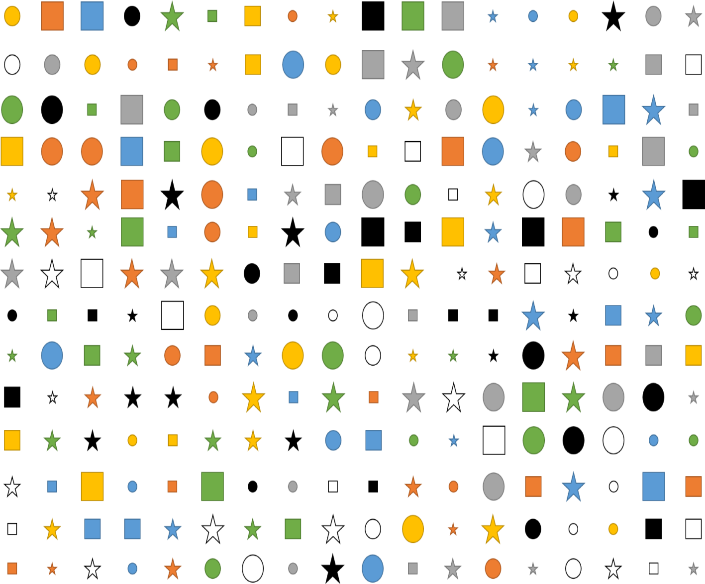
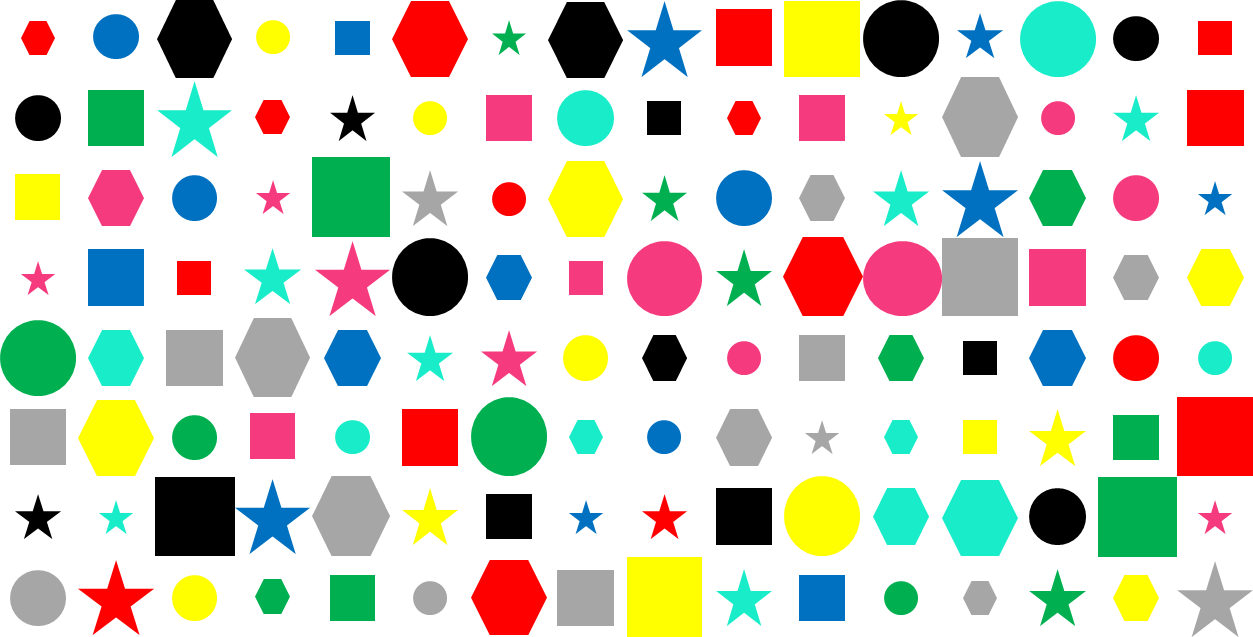


Figure 218 - VTASC Coded Pattern Example

The coding states are configurable over PHY PIB attributes phyVTASCTLevel, phyVTASCALevel, phyVTASCSLevel, and phyVTASCCLevel .Table 148 describes the bits per symbol for VTASC code design.

Table 148 - Bits per symbol for VTASC coded block models

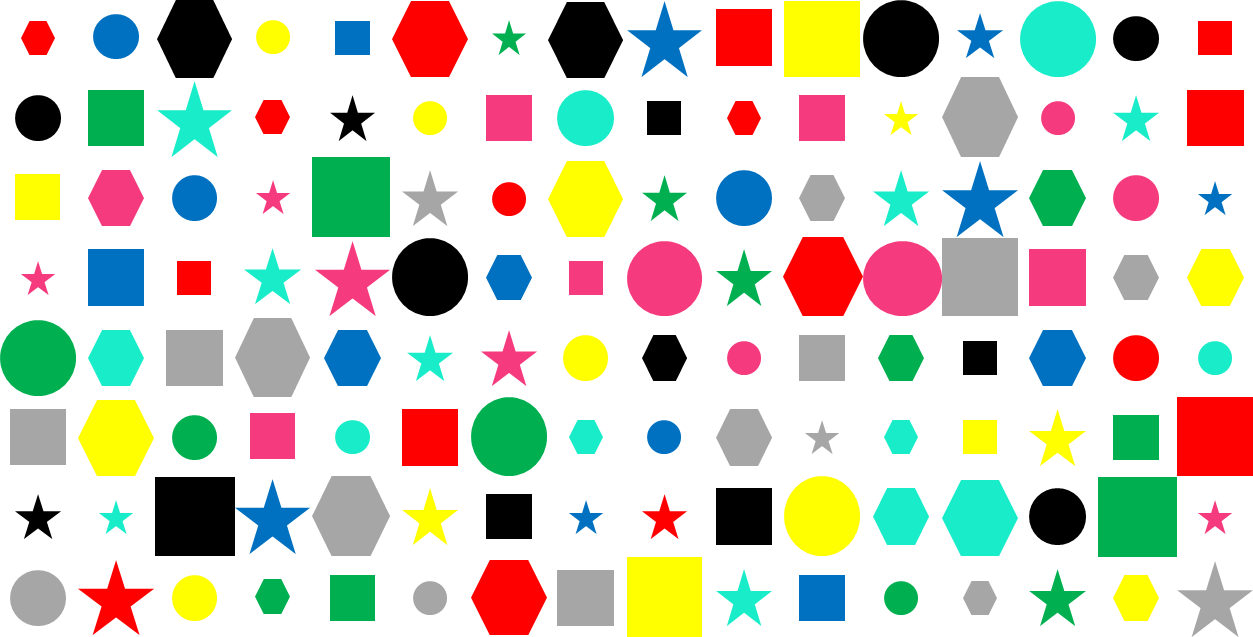
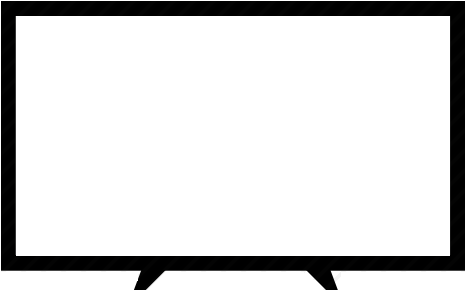
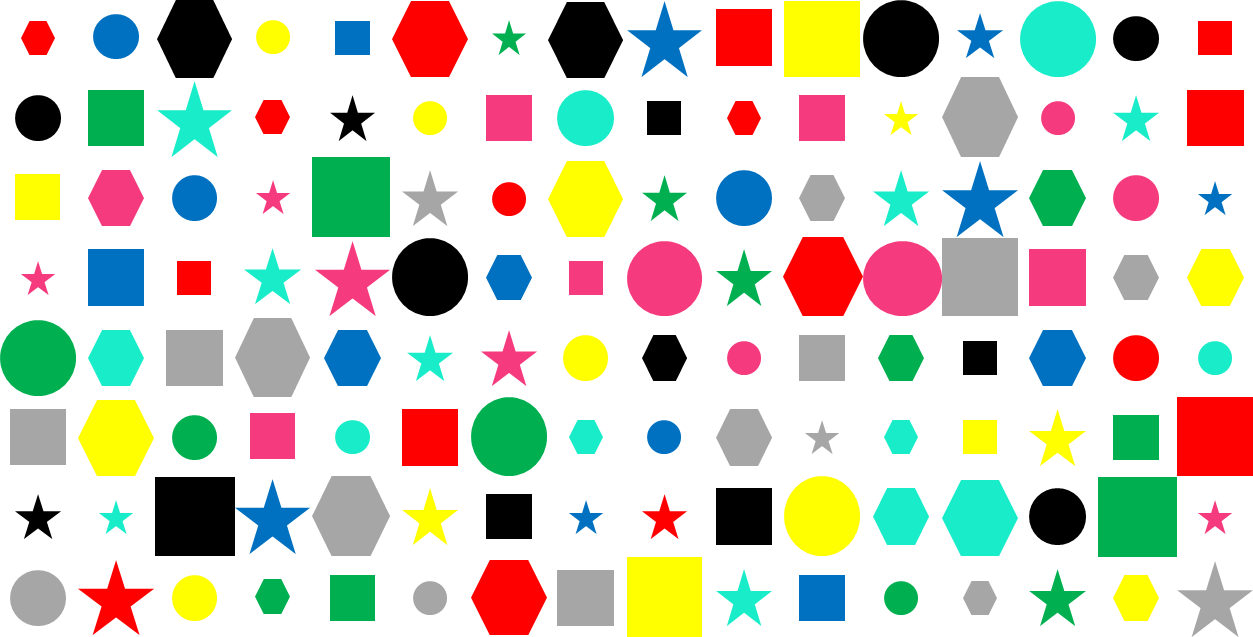
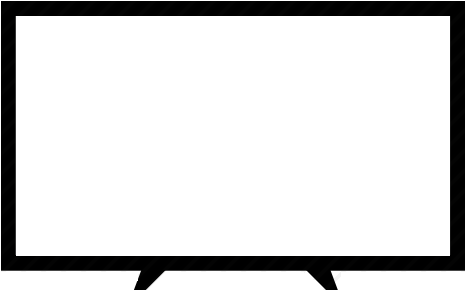
|  |  |  |
| --- | --- | --- |
| **Coding (T,A,S,C) States** | **Number of Coded patterns (T\*A\*S\*C)** | **Bits per symbol** |
| T = 2, A = 4,S = 4, C = 2 | 64 = 26 | 6 |
| T = 2, A = 4,S = 4, C = 4 | 128 = 27 | 7 |
| T = 2, A = 4,S = 4, C = 8 | 256 = 28 | 8 |

Table 149 describes the data bits to coding states mapping for VTASC code design.

Table 149 - VTASC coded symbol bit mapping with coding states (T, A, S, C)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bits Per Symbol** | **Data Bits** | | | | | | | |
| **B7** | **B6** | **B5** | **B4** | **B3** | **B2** | **B1** | **B0** |
| 6 | - | - | T | A | A | S | S | C |
| 7 | - | T | A | A | S | S | C | C |
| 8 | T | A | A | S | S | C | C | C |

The number of horizontal and vertical blocks depends on the partial or full screen coded mode by PHY PIB attributes phyVTASCCodedArea. The example VTASC coded pattern of the full and partial screen coded mode is shown in Figure NEW2 (a) & NEW2 (b).



1. Full Screen Coded Mode (b) Partial Screen Coded Mode

Figure NEW2 – VTASC Shapes

In partial screen mode, the number of horizontal and vertical blocks configurable over the PHY PIB attributes phyVTASCAHSize, phyVTASCAVSize. In full screen mode, the number of horizontal and vertical blocks estimated based on the screen size, resolution, aspect ratio, and the relative pixel ratio in reference with 42 inches full HD (1920 pixels of width and 1080 pixels of height) display.

The size of the block is vary with screen size and aspect ratio. For an example the block size of 21 inches full HD display looks compared to reference 42 inches full HD display. To generate exact block size as reference with 42 inches full HD display, need to dynamically calculate the pixel ratio so that all display transmitter can generate same block size VTASC coded pattern. The pixel ratio for another displays is,

Pixelratio = (hNewResolutiion / hRefResolutiion) \* (InchesRef/InchesNew) \* (Squareroot (1-AspectRatioNew) / Squareroot (1-AspectRatioRef))

Where,

* HNewResolutiion is horizontal resolution of display pixel to be estimated
* hRefResolutiion is horizontal resolution of reference display
* InchesRef is inches of display pixel to be estimated
* InchesNew in inches of reference display
* AspectRatioNew is aspect ratio of display pixel to be estimated
* AspectRatioRef is aspect ratio of reference display

**15.2.4 VTASC Encoder**

The display light based transmitter with VTASC encoder works by overlaying the data mapped color code on visual scene as show in Figure 219.

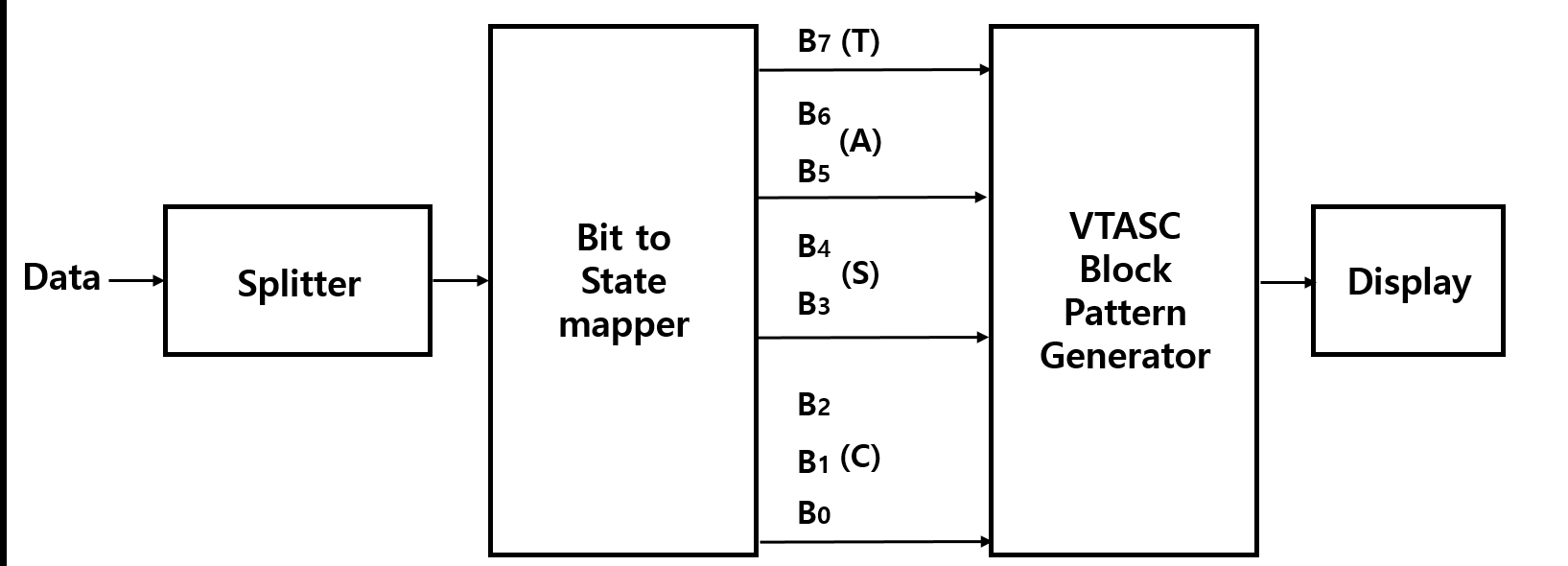


Figure 219 – VTASC Data Encoder

The data coded on display by overlaying visual patterns in displays visual area. The overlaying visual pattern means that updating coded region pixel value according to VTASC coded pattern on display frame buffer to refresh on display screen. The overlaying coded pattern on frame buffer and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. Table 150 describes the example data rate supported by VTASC code design with block size of 32x32 pixels on 42 inches full HD display with 16:9 aspect radio.

Table 150 – VTASC Data Rate Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation**  **(TxAxSxC)** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| 2 Color VTASC Code  (T = 2,A=4,S=4,C=2) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 384 Kbps  (FEC None) |
| 4 Color VTASC Code  (T = 2,A=4,S=4, C=4) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 448 Kbps  (FEC None) |
| 8 Color VTASC Code  (T = 2,A=4,S=4,C=8) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 512 Kbps  (FEC None) |
| 2 Color SS VTASC Code  (T = 2,A=4,S=4,C=2) | None | 30Hz | None | 192 Kbps1 |
| 4 Color SS VTASC Code  (T = 2,A=4,S=4,C=4) | None | 30Hz | None | 224 Kbps1 |
| 8 Color SS VTASC Code  (T = 2,A=4,S=4,C=8) | None | 30Hz | None | 256 Kbps1 |

1 where spreading factor is 2

The data rate calculation is described below,

DataRate = (NoofBlocks \* BitsPerSymbol \* OpticalClockrate \* FECRate) / CodeLength)

Where,

* CodeLength is 1 for without SS spreading and respective spreading code factor used for with SS spreading
* ScreenWidth = 1920; ScreenHeight = 1080;
* BlockWidth = 32; BlockHeight = 32 ;
* NoOfHorizontalBlocks = (ScreenWidth / BlockWidth) = 60 (Approx. to even for coding efficiency)
* NoOfVerticalBlocks = (ScreenHeight / BlockHeight) = 32 (Approx. to even for coding efficiency)
* NoOfBlocks = (NoofHorizontalBlocks\* NoofVerticalBlocks)
* BitsPerSymbol = 7 (Refer Table 206)
* OpticalClockrate = 30 Hz
* FECRate = 1 (Refer Table 206)

The Data Rate for 2 Color VTASC Code with 8 size scalability & 4 shapes & 2 transparency Level without SS spreading code (CodeLength is 1)

DataRate = ((ScreenWidth / BlockWidth)\* (ScreenHeight / BlockHeight) \* BitsPerSymbol \* OpticalClockrate \* 1) / 1) = 403200 = 390 Kbps (Approx.)

VTASC uses the two transparency levels in code design. The transparency defines the pixel with an observed color when given the pixel and a background VTASC coded block color. The symbol to bitmapping for transparency is shown in Table 151.

Table 151 – Symbol to bit mapping for transparency level

|  |  |
| --- | --- |
| **Symbol Bit (B7)** | **Transparency Level (%)** |
| 1 | 100 |
| 0 | 50 |

VTASC block size represented by amplitude state in code design. The symbol to bitmapping for block size is shown in Table 152.

Table 152 – Symbol to Bit Mapping for Amplitude Level

|  |  |
| --- | --- |
| **Symbol Bits (B6,B5)** | **Block Size (MxN Pixels)** |
| 00 | 128x128 |
| 01 | 96x96 |
| 10 | 64x64 |
| 11 | 32x32 |

VTASC uses the four shape model in code design. The four shape models are shown in Figure NEW1. The symbol to bitmapping for shape is shown in Table 153.

Table 153 – Symbol to bit mapping for shape model

|  |  |
| --- | --- |
| **Symbol Bits (B4,B3)** | **Shapes** |
| 00 | square |
| 01 | circle |
| 10 | hexagon |
| 11 | star |

VTASC uses the eight color in code design. The symbol to bitmapping for color channel is shown in Table 154. The coded region background color is white.

Table 154 – Symbol to bit mapping for color channel

|  |  |
| --- | --- |
| **Symbol Bits (B2,B1, B0)** | **Color Channel** |
| 000 | Black |
| 001 | Red |
| 010 | Green |
| 011 | Blue |
| 100 | Yellow |
| 101 | Magenta |
| 110 | Cyan |
| 111 | Gray |

**15.2.5 Asynchronous Communication**

Transmitter does not use any reference block for receiver synchronization with transmitter. The spreading codes used to support receiver to perform asynchronous data decoding irrespective of receiver frame rate variation. To provide efficient receiver synchronization, every frame in the video sequence spreading with one spreading code and the alternative frames use the spreading code pairs sequentially. The spreading sequence order in the video frame sequence is shown in Figure 220.

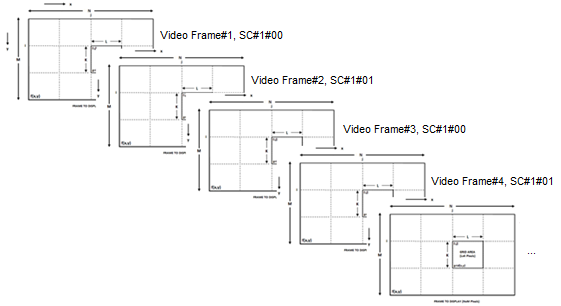


Figure 220 – Video Frame Sequence SS Code Assignment

The receiver decoding for asynchronous communication and receiver error mitigation due to rolling effect is given in Annex J.9.1.

**15.2.6 Scalable Bitrate Controller**

The VTASC and SS2DC PHY for display based OWC designed with built-in scalable bitrate controller. There are two types of scalable bitrate controller supported,

* Receiver framerate adaptive multirate controller
* Receiver distance adaptive data rate controller

The scalable bitrate control mode selection configurable over the PHY PIB phyVTASCScalRateCtrl.

**15.2.6.1 Receiver frame rate adaptive multirate controller**

The screen is divided into 2x2 regions and each region encode with different optical rate and renders the visual scene on screen. The different optical rate encoded region is spreaded by pair of spread code as defined in Table 147. The same encoded pattern rendered repeatedly at the rate of (displayRefreshRate / OpticalClockRate) to control the multirate data rate control on single screen. To achieve robust communication, the scalable multirate data transmission in PHY model design is shown in Figure 222. The region based optical clock rate and SS code con configurable over the PHY PIB attributes phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhySSCode1FP00 to PhySSCode4FP01.

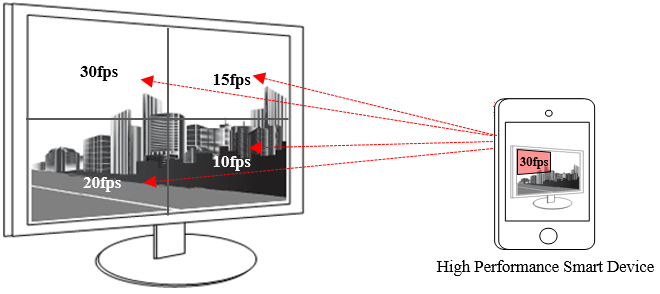
****

Figure 222 – Scalable Bitrate Controller

**15.2.6.2 Receiver distance adaptive data rate controller**

The receiver distance adaptive data rate control is by changing the block pattern size small for short distance (example: 32x32 but vary with display) and big for long distance (example: 128x128 but vary with display). The region based distance adaptation is configurable PHY PIB attributes phyVTACScalRegion1DistanceRange, phyVTACScalRegion2DistanceRange, phyVTACScalRegion3DistanceRange, and phyVTACScalRegion4DistanceRange.

The different distance range encoding is spreaded by pair of spread code as defined in Table 147. In this case the transmitter built-in with camera features as shown in Figure 223 to estimate the receivers distance using camera. The receiver distance estimation is not part of this standard. The distance based optical clock rate and SS code configurable over the PHY PIB attributes phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhyVTACSSCode1FP00 to PhyVTACSSCode4FP01.

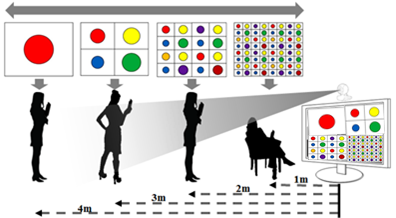
****

Figure 223 – Distance Adaptive Data rate Control

**15.3 SS2DC Specification**

The Sequential Scalable 2D Code (SS2DC) works with different 2D codes organized in a combination of one or more codes sequentially in row and column manner. The SS2DC uses QR, VTASC, A-QL, HA-QL, and IDE 2D codes. The SS2DC PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.3.1 SS2DC Reference Architecture**

The reference PHY architecture for SS2DC is illustrated in Figure 224. The data embedded on visual frame by overlaying 2D code patterns in defined displays visual area. After spread spectrum, data is transformed into SS2C coded 2D patterns according to the 2D Code encoder.

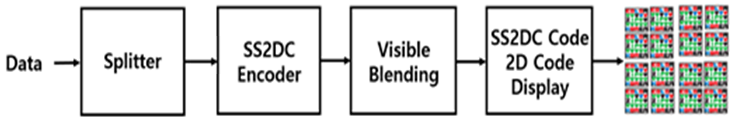
****

Figure 224 – Reference architecture for SS2DC PHY System

The SS2DC is used for enhanced display to camera communication in the real-time application usage scenario. The SS2DC specific working features are given in Annex I.3.2.

The receiver specific information for SS2DC Data Decoder is given in Annex J.10.

**15.3.2 SS2DC Code Design**

SS2DC is a two dimensional design using different 2D codes for improved OWC throughput by sequentially arranging the coded screen symbols in a 2D order as shown in Figure 225. The data is encoded as per 2D code principle and displayed on the display screen or panels. The number of horizontal and vertical 2D code blocks is configurable over the PHY PIB phyVTASCAHSize, phyVTASCAVSize. The horizontal and vertical size of the 2D code is configurable over the PHY PIB PhySS2DCCODEHSIZE and PhySS2DCCODEVSIZE.



(a) 2D QR Code (b) 2D Color Code (c) Combination of 2D Codes

Figure 225 – SS2DC Code Design Examples

**15.3.3 SS2DC Encoder**

The display light based transmitter with SS2DC encoder works by overlaying the data mapped color code on visual scene as show in Figure 226.

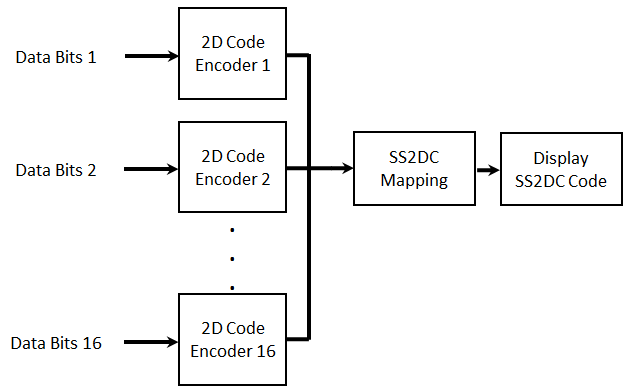


Figure 226: Sequential Scalable 2D Code Data Encoder

The data coded on display by overlaying visual patterns in displays visual area. The overlaying visual pattern means that updating coded region pixel value according to SS2DC coded pattern on display frame buffer to refresh on display screen. The overlaying coded pattern on frame buffer and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. SS2DC uses one or more 2D codes from QR, VTASC, A-QL, HA-QL, and IDE 2D codes.

The QR based data encoder uses QR code version 40 and follows the ISO/IEC 18004 standard. The A-QL based data encoding informations a described in subclause 15.1. The VTASC based data encoding informations a described in subclause 15.2. The IDE based data encoding informations a described in subclause 15.4. The VTASC based data encoding informations a described in subclause 15.6. The minimum QR code size must be equal to (scanning distance / 10) to have an effective QR detection.

Table 155 describes the data rate supported by SS2DC code design with QR code. The Table 155 data rates differs according to the 2D code specification.

Table 155 – SS2DC Data Rate Table Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| 2x2 SS2DC | None | 2DCodeDecodingRate | RS(64,32)/ RS(160,128)/None | 92 Kbps  (FEC None) |
| 4x4 SS2DC | None | 2DCodeDecodingRate | RS(64,32)/ RS(160,128)/None | 368 Kbps  (FEC None) |

The data rate calculation is described below,

DataRate = NoOfCodeSequence\* (2DCodeDataCapacity \* OpticalClockrate \* FECRate)

Where,

* NoofHorizontalBlocks is no of horizontal 2D code sequence
* NoofVerticalBlocks is no of vertical 2D code sequence
* NoOfCodeSequence = (NoofHorizontalBlocks\* NoofVerticalBlocks)

The data rate for 2x2 SS2DC with QR code (The maximum data capacity is 2953 bytes) and DataRate = 4\* (2953 \* 8)\* 1 \* 1) / 1) = 94494 = 92 Kbps (Approx.).

**15.4 IDE Specification**

The Invisible Data Embedding (IDE) works on embedding data on visual frame in unobtrusive mode using blending and watermarking. The IDE PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 81.

# **15.4.1 IDE Reference Architecture**

The reference PHY architecture for IDE is illustrated in Figure 298. The data embedded on visual frame by invisible image blending and watermarking in the defined displays visual area. After spread spectrum, data is transformed into IDE encoding according to the invisible blending and watermarking rules as described in 227.

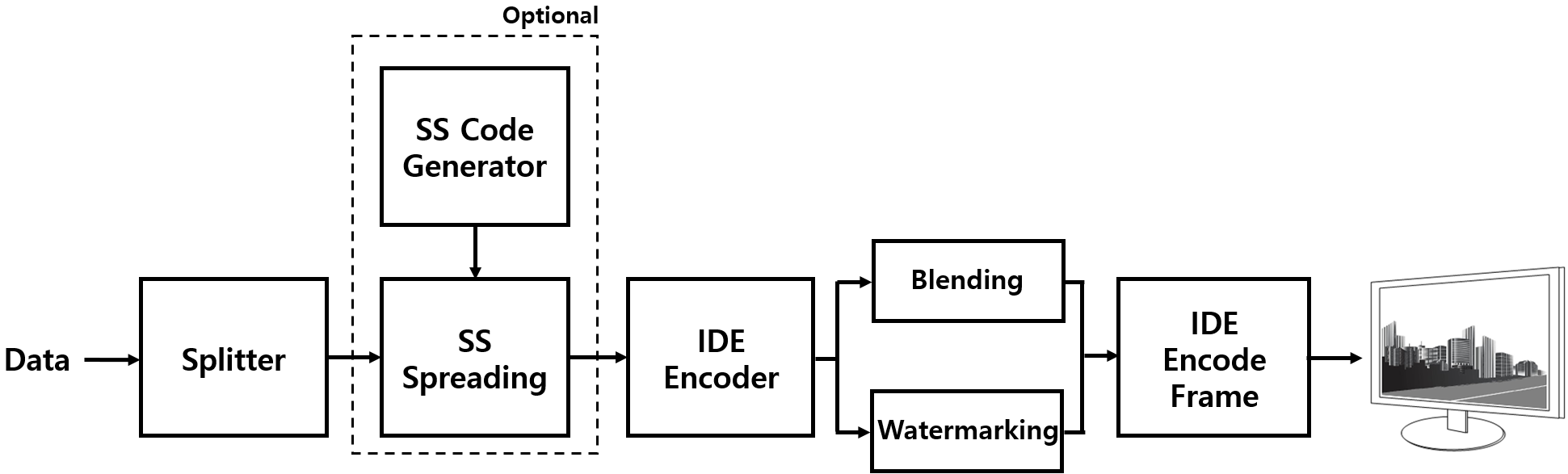
****

Figure 227 – Reference architecture for IDE PHY System

The spread spectrum used with IDE to have effective asynchronous and communication, distance adaptive scalable data rate controlled OWC. The IDE is used for enhanced display to camera communication in the real-time application usage scenario. The IDE specific working features are given in Annex I.3.3.

The receiver specific information for IDE data decoder is given in Annex J.11.

**15.4.2 Spread Spectrum**

The spread spectrum used with IDE coded display based transmitter to add built-in adaptation on data recovery in addition to achieve the asynchronous communication with angle free and distance adaptive communication between transmitter and receiver. The subclause 15.2.2 contains more information about spread spectrum.

**15.4.3 IDE Encoder**

IDE is a two dimensional block based imperceptible data encoder for unobtrusive OWC communication between screen and camera. The visual display frame is divided into MxN pixel blocks and the human imperceptible data encoded on visual scene block using M-FSK/PSK by image blending and 2D-Binary code by image watermarking. The display light based transmitter with IDE encoder works by invisible overlaying the data mapped on visual scene as show in Figure 228. The M-FSK, Hybrid-MPFSK, 2D Binary Code, Blending, and Watermarking are described in 15.4.3.1, 15.4.3.2, 15.4.3.3, 15.4.3.4, 15.4.3.5 respectively.

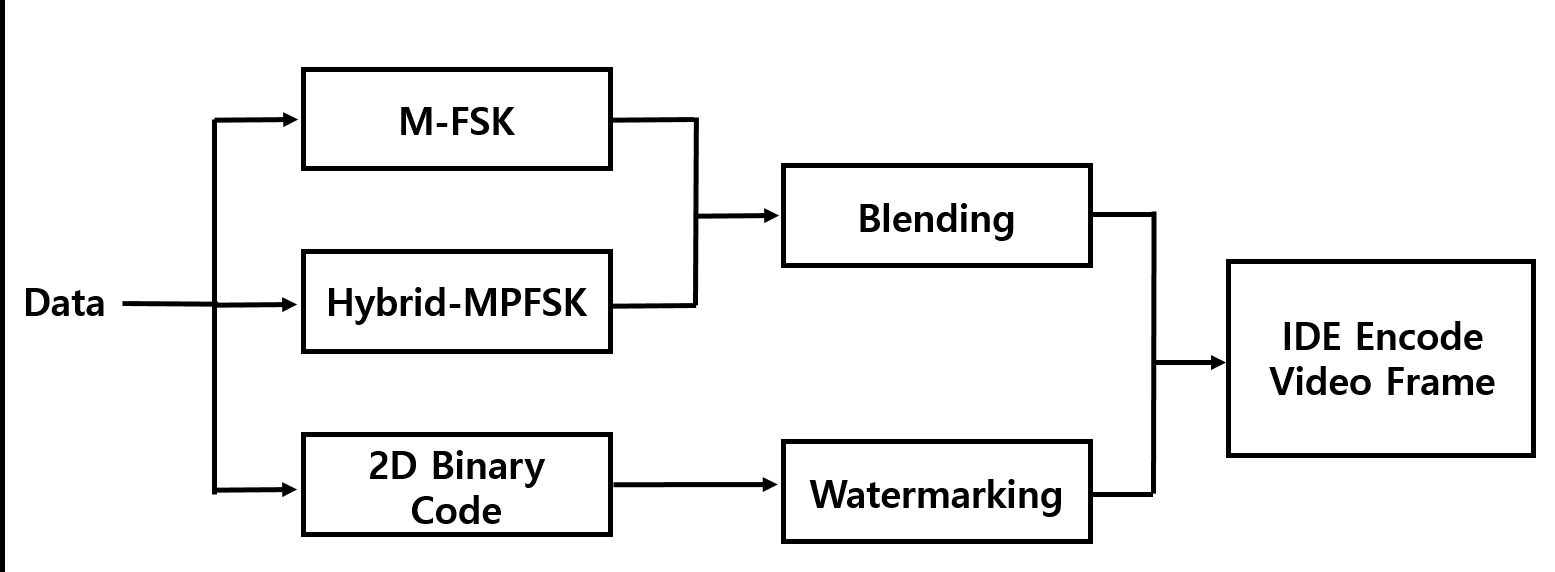


Figure 228 – IDE Data Encoder

The IDE transmitter mode, modulation are configurable over the PHY PIB phyIDETxMode, phyIDEModulation. The 16x16, 32x32, and 64x64 are three blocks size is used and configurable over the PHY PIB phyIDEMxNBlockSize.

Table 156 describes the data rate supported modulation schemes. The IDE encoder data rate estimated with reference to full HD display. The aspect ratio, size (inches) does not impact the distance does change the scale of the block design.

Table 156 – IDE PHY Data Rate Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modulation** | **RLL Code** | **Optical Clock Rate** | **FEC** | **Data Rate (Kbps)** |
| M-FSK-BLEND | None | 30Hz | RS(64,32)/ RS(160,128)/None | 16 Kbps  (FEC None) |
| HYBRID-PFSK-BLEND | None | 30Hz | RS(64,32)/ RS(160,128)/None | 32 Kbps  (FEC None) |
| 2DBIN-WM | None | 30Hz | RS(64,32)/ RS(160,128)/None | 128 Kbps  (FEC None) |
| SS-M-FSK-BLEND | None | 30Hz | None | 8 Kbps1 |
| SS-HYBRID-PFSK-BLEND | None | 30Hz | None | 16 Kbps1 |
| SS- 2DBIN-WM | None | 30Hz | None | 64 Kbps1 |

1 where spreading factor is 2

The IDE PHY transmitter supports asynchronous communication by spreading data stream using SS code. The subclause 15.2.4 contains more information about asynchronous communication.

**15.4.3.1 M-FSK Modulation**

The M-FSK uses the 16 frequency ranges to map data symbol. The SS spreaded data bits sequence is splited as a 4 bit symbol and map into a selected 16 frequencies as shown in Figure 229. The number of frequencies used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency.

The M-FSK encoder generate the sine waveform for the symbol mapped frequency according to Table 157 and store the sine waveform in an MxN 2D pattern. The M-FSK encoded symbol 2D pattern of MxN pixel block is blended with visual scene to be rendered on screen. The data blending is described in 15.4.3.4.

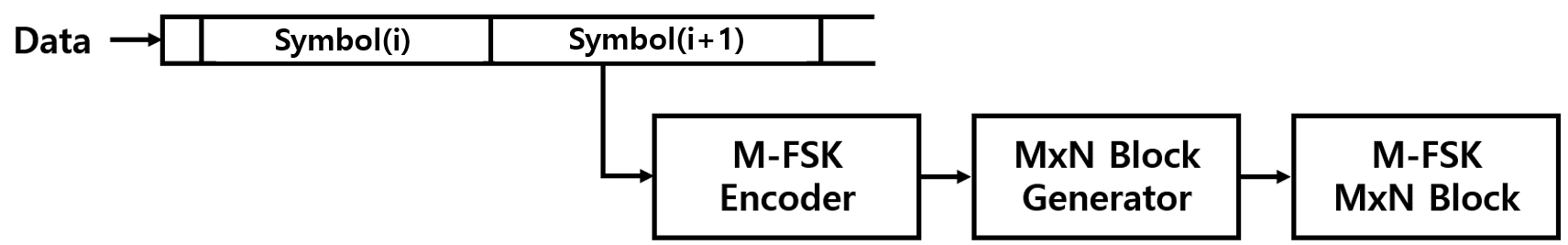


Figure 229 – IDE M-FSK Data Encoder

The 4 bit symbol to bitmapping for M-FSK is shown in Table 157. The symbol coded sine waveform patterns are generated dynamically based on PHY PIB attributes and stored in lookup table (LUT).

Table 157 – Symbol to bit mapping for M-FSK

|  |  |
| --- | --- |
| **Symbol Bits (B3, B2,B1, B0)** | **Frequency Mapping** |
| 0000 | f0 |
| 0001 | f1 |
| 0010 | f2 |
| 0011 | f3­ |
| …. | … |
| 1101 | f13 |
| 1110 | f14 |
| 1111 | f15 |

The frequencies f0~f15 are selected to encode data and the relationship between frequencies are,

fi = f0 + i.df (I = 1.2……..15) , Where df is the selected frequency separation value.

The selection of all frequencies shall be configured over the first frequency (f0) which shall be implemented over the PHY PIB attribute phyIDEFreqBase (200Hz by default) and the frequency separation which shall be implemented over the PHY PIB attribute phyIDEFreqSeparation (50Hz).

**15.4.3.2 Hybrid-MPFSK Modulation**

Hybrid scheme used to achieve double the data rate ~~of M-PSK or F-FSK~~ by combining frequency and phase on the modulation. The Hybrid-MPFSK uses the 16 frequency and two phase ranges to map data symbol. The data bits spreaded with SS sequence is splited as a 5 bit symbol and map into a selected 16 frequencies conjunction with two phases as shown in Figure 230. The number of frequencies and phase used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency, and phyIDEPSKNoPhase.

The Hybrid-MPFSK encoder generate the sine waveform for the symbol mapped frequency and phase according to Table 158 and store the sine waveform in an MxN 2D pattern. The Hybrid-MPFSK encoded symbol generates 2D pattern of MxN pixel block to blend with visual scene to be rendered on screen. The data blending is described in 15.4.3.4.

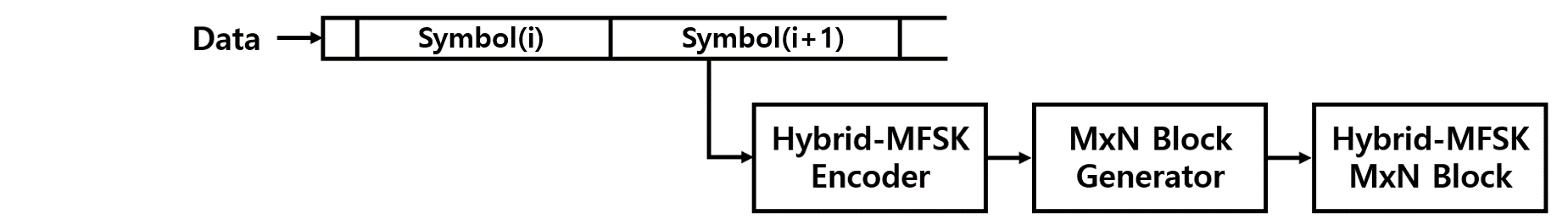


Figure 230– IDE Hybrid-MPFSK Data Encoder

The 5 bit symbol to bitmapping for Hybrid-MPFSK is shown in Table 158. The symbol coded patterns are generated dynamically based on PHY PIB attributes and stored in lookup table (LUT).

Table 158 – Symbol to bit mapping for Hybrid-MPFSK

|  |  |  |
| --- | --- | --- |
| **Symbol Bits (B4, B3 B2,B1, B0)** | **Phase Mapping** | **Frequency Mapping** |
| 00000 | P0 | f0 |
| 00001 | P0 | f1 |
| … | … | … |
| 01111 | P0­ | f15­ |
| 10000 | P1 | f0 |
| 10001 | P­1 | f1 |
| … | P1 | … |
| 11111 | P1 | f15 |

The frequency selection and configuration are described in 15.4.3.1.

The phase P0~P1 are selected to encode data and the relationship between phases are,

Pi = P0 + i.dP (I = 1.2), Where dP is the selected phase separation value.

The selection of all phases shall be configured over the first phase (P0) which shall be implemented over the PHY PIB attribute phyIDEPhaseBase, (0 by default) and the phase separation which shall be implemented over the PHY PIB attribute phyIDEPhaseSeparation (180).

**15.4.3.3 2D Binary Code**

The horizontal and vertical encoding area pixel ranges shall be configured over the PHY PIB attribute phyIDEENCHozAreaSize, phyIDEENCVerAreaSize. The 2D binary encoder, first calculate the number of MxN pixels blocks in visual display by dividing phyIDEENCHozAreaSize and phyIDEENCVerAreaSize by 8. The MxN number of encoding blocks sized SS spreaded data is extracted and converted to 2D format of MxN dimension. The described 2D Binary Data encoder is shown in Figure 231.

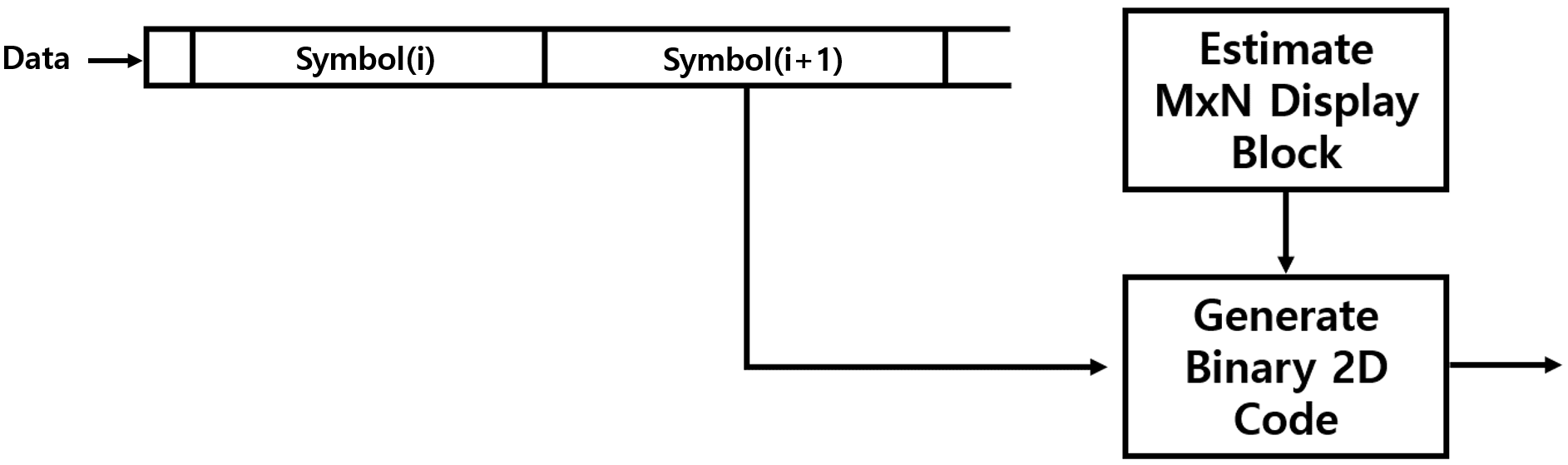


Figure 231 – IDE 2D Binary Data Encoder

**15.4.3.4 Invisible Data Blending**

The M-FSK or Hybrid-MPFSK coded MxN block is blended with in visual frame sequential in every block by cell row to cell column manner and rendered on display screen ~~as shown in~~ as shown in Figure 232.

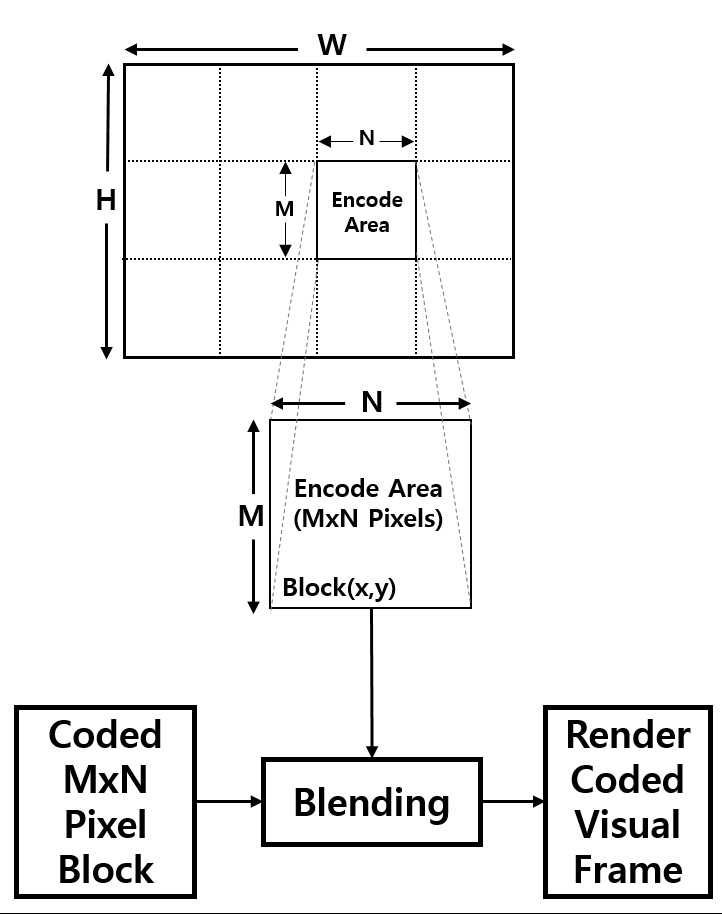


Figure 232 – IDE Encoder Blending

The visual blending rule is,

IDEEncodedFRAME = α. VisualFrameBlock(x, y) + (1-α) M-PFSKCodeBlock.

Where

* α is blending factor ~~α is~~ and α = 0.0~0.3 for invisible blending
* x is current row of MxN block in visual frame
* y is current column of MxN block in visual frame

**15.4.3.5 Invisible Watermarking**

The human eye is more sensitive to lower frequency components than to higher-frequency components. This means that most of the important information in an image is contained in the lower-frequency components. So, the higher frequency components can be discarded without visible degrading the image. The invisible watermarking utilize the human eye visual imperceptibility in middle-high frequency component of every 8x8 block of the visual frame.

The binary 2D coded MxN block is watermarked imperceptibly with in visual frame of every 8x8 block by row to column manner and rendered on display screen as shown in as shown in Figure 233.

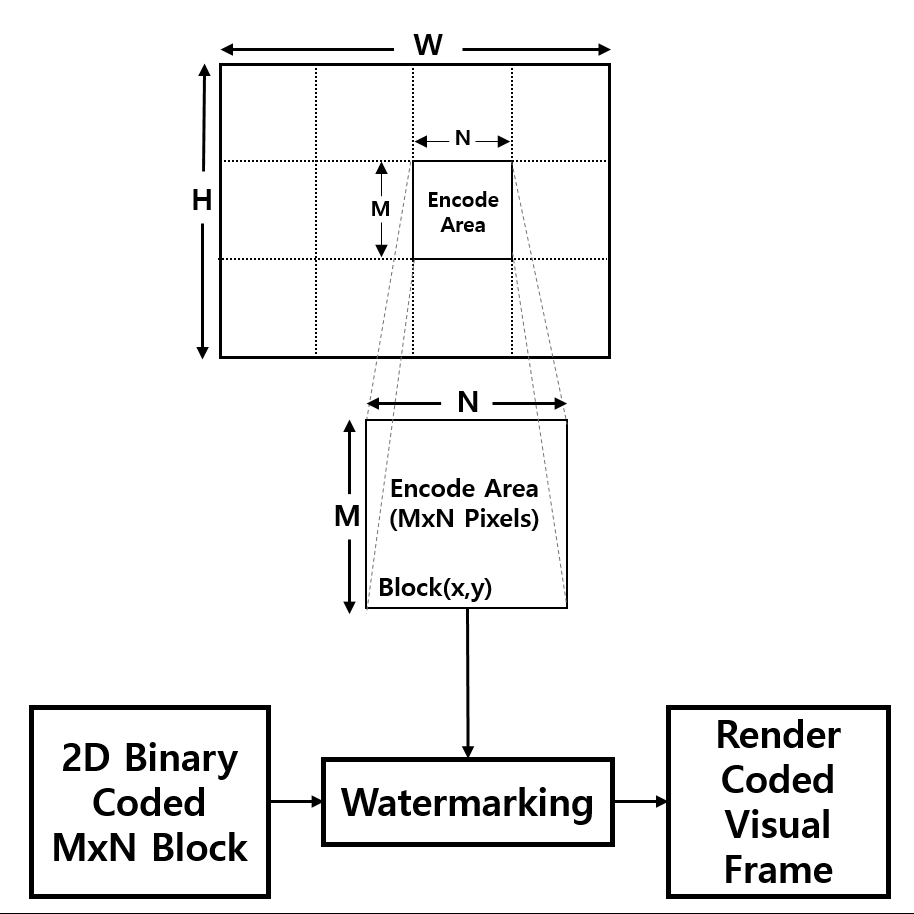


Figure 233 – IDE Encoder - Watermarking

This frequency based Transform domain watermarking is dominant on for invisible data embedding using the discrete cosine transform (DCT). The detailed frequency based invisible data embedding procedure using DCT is described in Annex M.

**Annex K (Normative)**

**J.9 VTASC Decoder Method**

VTASC data decoding is shown Figure J.13.

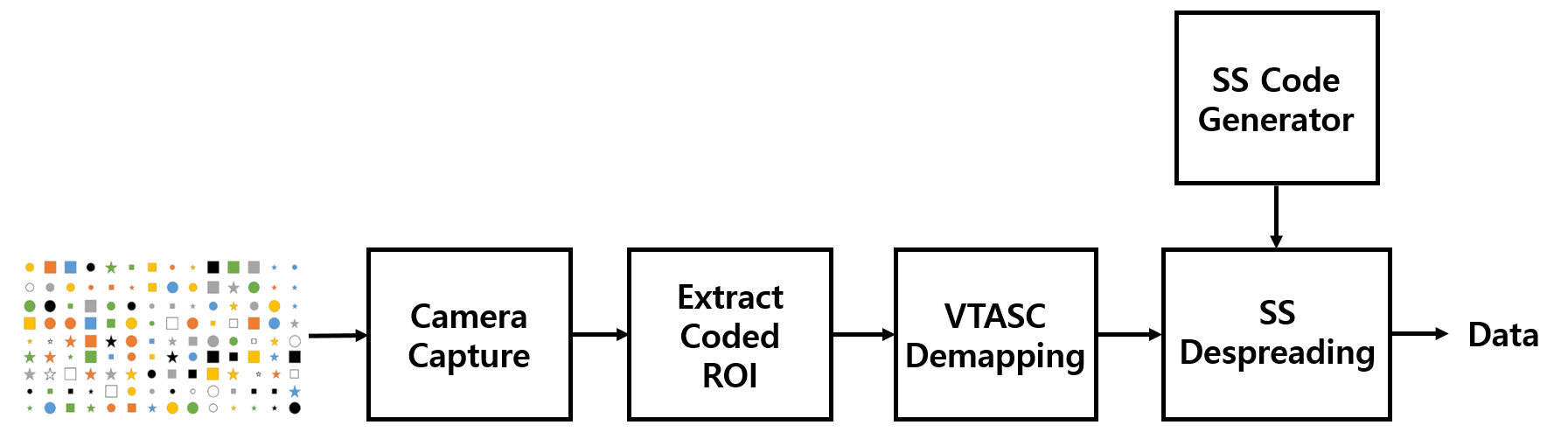


Figure J.13 – VTASC Receiver Functional Block Diagram

The ROI of screen visual area is extracted from the captured visual frame and then detect the VTASC coded patterns based on mapping scheme applied on the transmitter. The data recovered by applying SS despreading on the VTASC decoded data streams.

**J.9.1 Asynchronous Communication**

The optical clock rate and SS codes are configurable over the PHY PIB phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhySSCode1FP00 to PhySSCode4FP01.The receiver synchronized using SS code (any one of the four pair SS code at first time) and decoded the data.

If camera captured same frame receive twice in a sequence, then receiver will discard the second video frames when SS detection fails with next code of pair of SS code that means SS detection is true with previous frames SS code. If the current frame decoding detects the both SS codes in a single frame, then that frame is rolling effect fault capture frame and must be discarded.

**J.9.2 Angle Free Communication**

The angle free communication between transmitter and receiver is shown in Figure J.14.



Figure J.14 – Angle free and distance adaptive communication

The angle free communication is achieved by warping the ROI of the transmitter to get the original shape alignment and then the decoded data synchronizing with spread code to extract original information transferred on transmitter. The kind automatic synchronization in receiver is time consuming function but the communication is robust.

In case a rolling shutter camera captures in between the transition time of two adjacent blocks of data, four Ab bits at four reference cells on the captured image shall not be the same. The image is detected as a rolling affected image as shown in Figure 262, and be discarded.

Also, to support a rotated camera decoding, the transmission of four rotation-indication bits via reference cells over the blue channel allows a receiver in identifying the starting corner of the code. The starting corner shall have the similar values with its two adjacent reference cells.

**J.10 SS2DC Decoder Method**

SS2DC data decoding is shown Figure J.15.

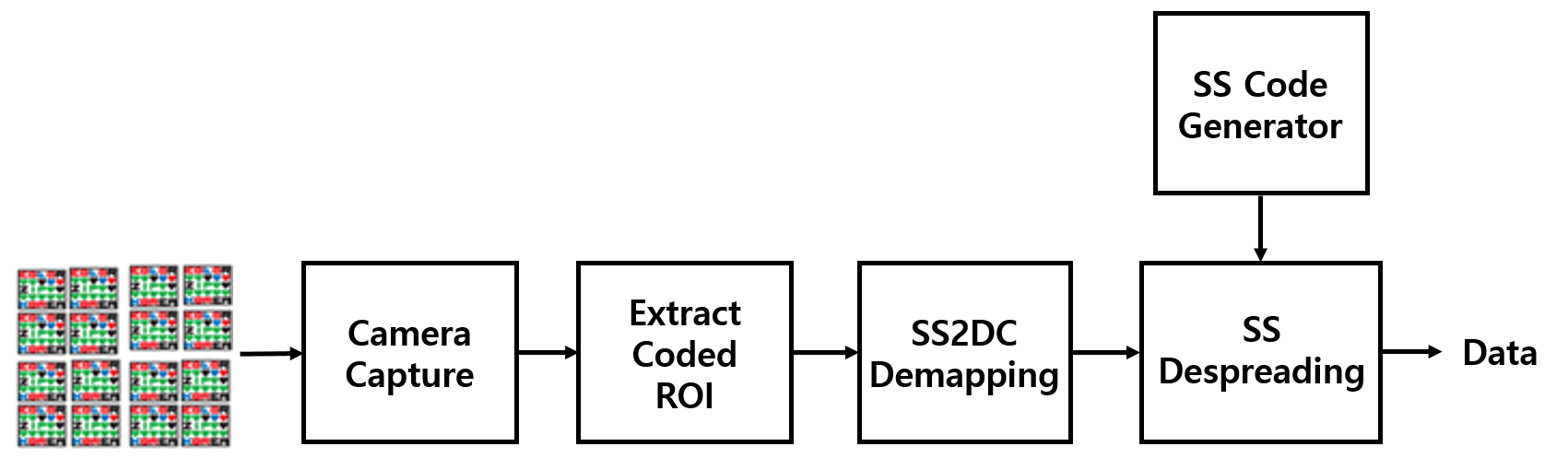


Figure J.15 – SS2DC Receiver Functional Block Diagram

The ROI of Screen Visual Area is extracted from the captured visual frame and then apply the Sequential Scalable 2D Code detector based on mapping scheme applied on the transmitter. The data recovered by applying SS despreading on the SS2DC data decoded.

**J.11 IDE Decoder Method**

IDE data decoding is shown Figure J.16.

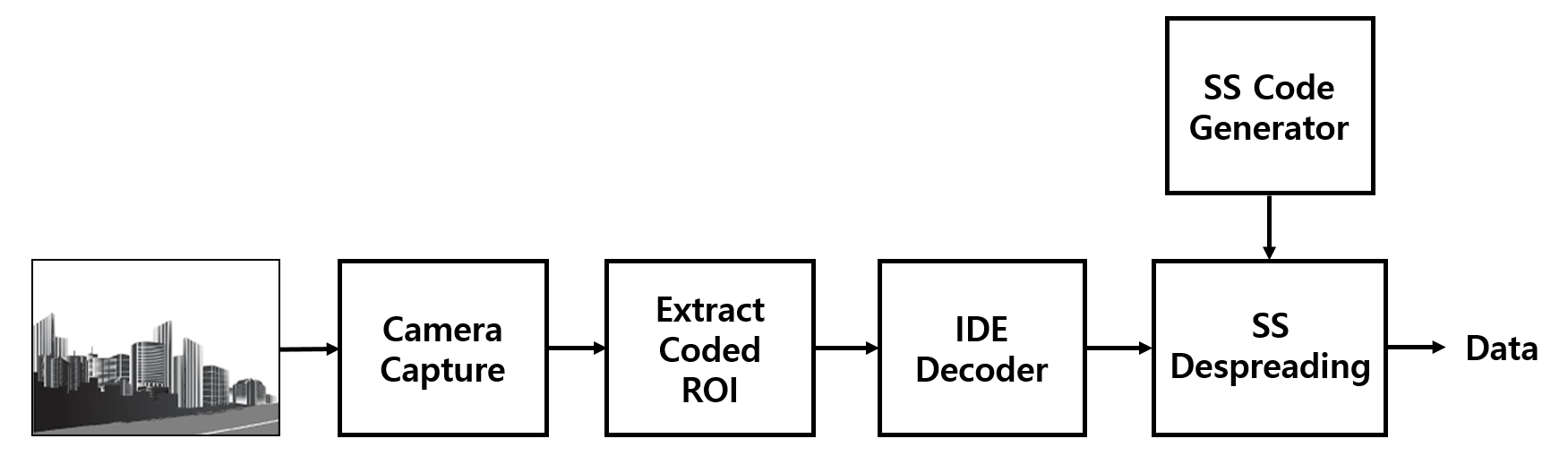


Figure J.16 – IDE Receiver Functional Block Diagram

To decode the data stream, the ROI of display visual area is extracted from the captured visual frame using image processing methods (like canny edge detection , Gaussian blurring, Contour Extraction, Perspective Transform, and Warping) and then invisibly embedded data extracted using blending or watermark extraction procedure.

The blending or watermark based data extraction procedure is applied based on modulation scheme used to invisibly embedding the data on the transmitter system (modulation scheme is described in 2.1). The blending works with combination of M-PSK and M-FSK and the decoder uses the FFT to detect the coded frequency and phase to decode the data.

The data embedded using high frequency visual coefficients on visual frame is extracted by applying DCT on every 8x8 block and then extracted data by extracting high frequency coefficient values. The recovered high frequency coefficient based data is SS coded data so SS decoding is applied to recover original data from the visual sequence.

**Annex I (Informative)**

**I.3 PHY Mode Specific Characteristics**

**I.3.1 VTASC**

The VTASC works on,

* Receiver angle free and distance adaptive communication
* Receiver distance adaptive communication achieved by screen with interactive Camera
* Asynchronous and receiver frame rate independent communication
* Scalable bitrate controller for distance adaptive data rate control
* Enhanced multi-display model for transmission

**I.3.2 SS2DC**

The SS2DC works on,

* Receiver angle free and distance adaptive communication
* Receiver distance adaptive communication achieved by screen with interactive Camera
* Asynchronous and receiver frame rate independent communication
* Scalable bitrate controller for distance adaptive data rate control
* Enhanced multi-display model for transmission

**I.3.3 IDE**

The IDE works on,

* Unobtrusive to screen viewer on dynamic visual Scene
* Receiver angle free and distance adaptive communication
* Asynchronous and receiver frame rate independent communication
* Enhanced multi-display model for transmission

**Annex M (Normative)**

**M.1 Frequency Based Invisible Watermarking**

The human eyes are more sensitive to noise in lower frequency range than its higher frequency counterpart, while the energy of most natural images are concentrated on the lower frequency range. In order to invisibly embed the watermark and to survive the lossy data compression, a reasonable trade-off is to embed the watermark into the middle-high frequency range of the image.

The DCT-based watermarking method has been used for frequency based image watermarking which could survive several kinds of image processing and lossy compression. The frequency based invisible watermarking using DCT is described in principle by the block diagram in figure M.1.

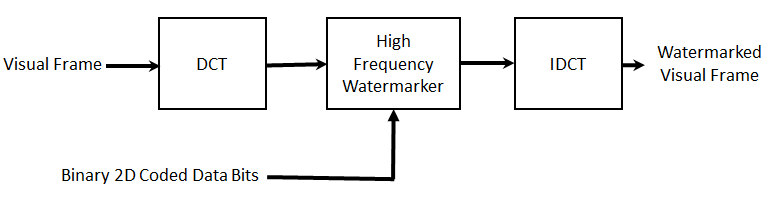


Figure M.1 – DCT based Watermarking Block Diagram

The visual rendering frame is divided into 8x8 blocks of pixels, and the 2-D DCT is applied independently to each block. Then, select the four coefficients of high -frequency range from the DCT coefficients for watermarking, an example of defining the high-frequency coefficients is shown in Figure M2.

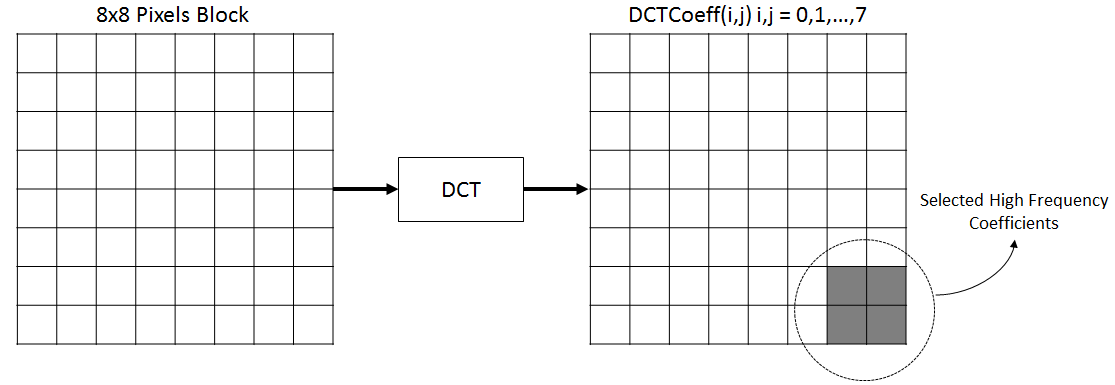


Figure M.2 – High-Frequency Coefficients Selection for Watermarking

The selected DCT coefficients for embedding data using watermarking are DCTCoeff (6, 6), DCTCoeff (6, 7), DCTCoeff (7, 6), and DCTCoeff (7, 7).