**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | **D2 Comments Resolution Based PHY-VI PHY Specification Revision** | |
| Date Submitted | June, 2017 | |
| Source | Jaesang Cha (SNUST), Minwoo Lee (SNUST), Soonho Jung (SNUST), Kim Chan (SNUST), Ilkyoo Lee (Kongju Nat’Univ.), Gilsik Lee (The Univ. of Texas at Dallas), Soo-Young Chang (CSUS), Vinayagam Mariappan (SNUST), | Voice: [ ] Fax: [ ] E-mail: [chajs@seoultech.ac.kr]1 |
| Re: | Draft D2 Comment Resolution based PHY-VI PHY Specification Revision | |
| Abstract | Details of Resolutions regarding to the submitted Comments on D2 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 D2 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 116.

# **15.2.1 VTASC Reference Architecture**

The reference PHY architecture for VTASC is illustrated in Figure 279. 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.

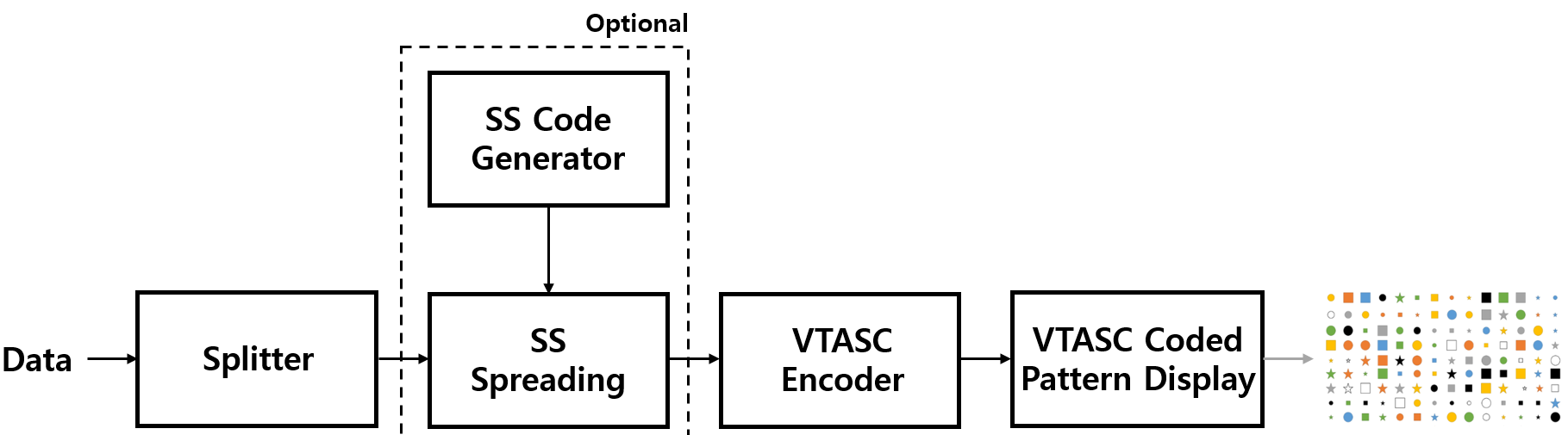
****

Figure 279 – 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 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

The receiver specific information for VTASC Data Decoder is given in Annex M.1.

**15.2.2 Spread Spectrum**

The spread spectrum used with VTASC, SS2DC, and IDE based display to camera OWC to have effective asynchronous and communication, distance adaptive scalable data rate controlled communication. The spread spectrum can use any orthogonal codes (like Walsh sequences, ZCD) or non-orthogonal codes (like PN, Gold, and Kasami sequences). 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 PQR.

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Spread Sequence** | **Spreading Code** | | | | | | | | **Distance (meters)** | **Optical Clock 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 | 20 |
| SC2#01 | -1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |
| SC3#00 | -1 | -1 | -1 | 1 | 1 | 1 | 1 | -1 | 3 | 15 |
| SC3#01 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 |
| SC4#00 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | above 4 | 10 |
| 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 ABC.

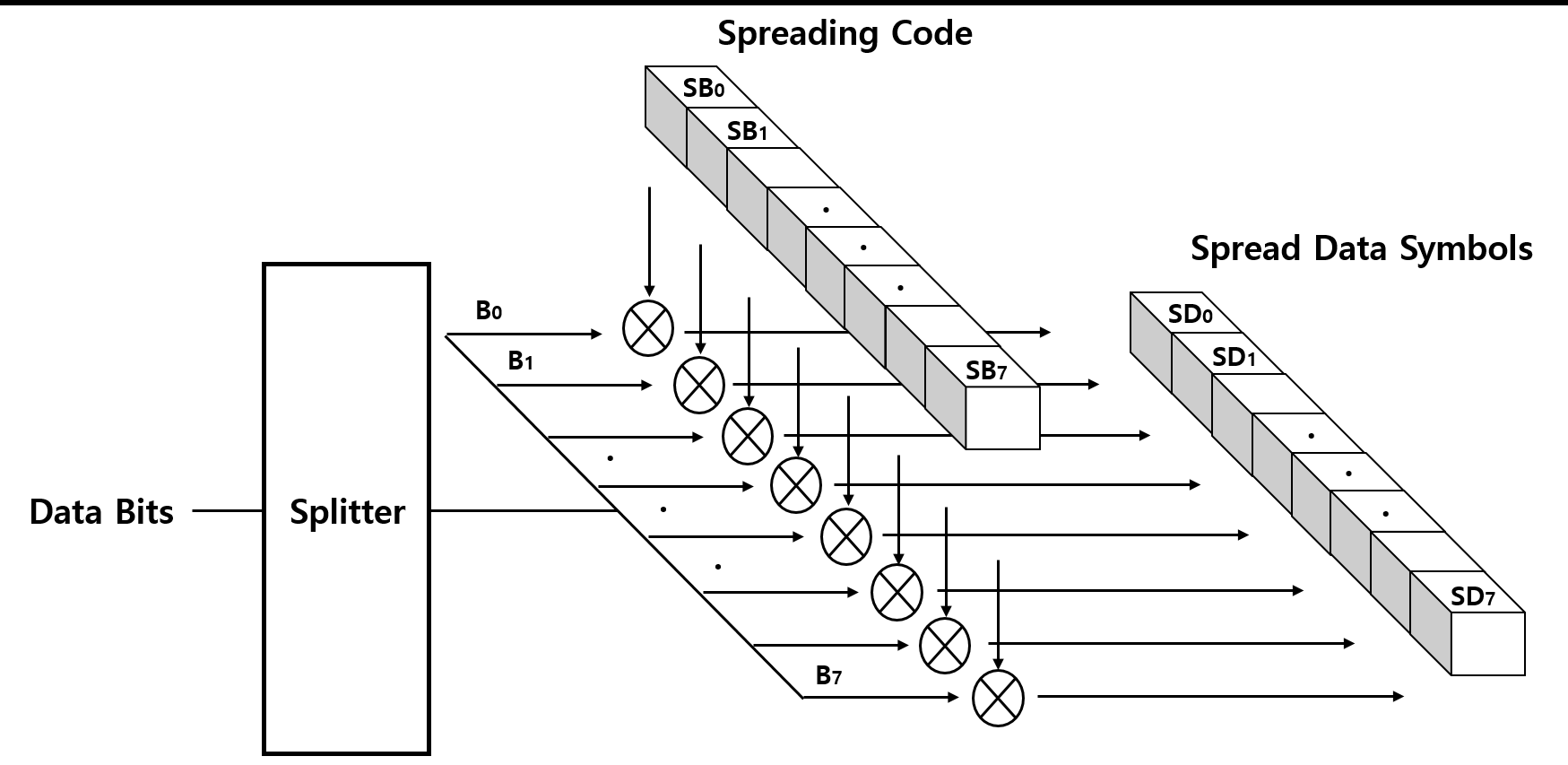


Figure ABC – SS Spreading Example

**15.2.3 VTASC Code Design**

VTASC is a modulation scheme for visible-light communication involving single or multiple light sources 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 size of the blocks) / S (Shapes) / C (Colors) State as described in the Figure 280.

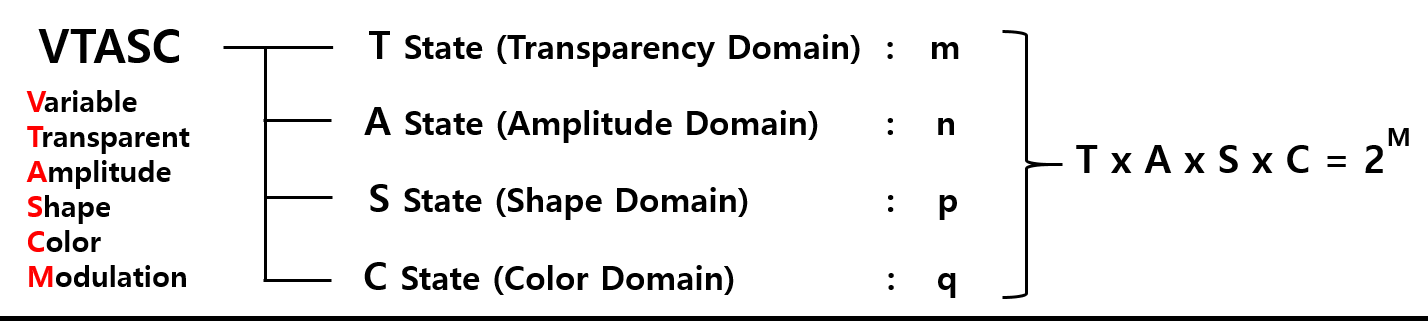
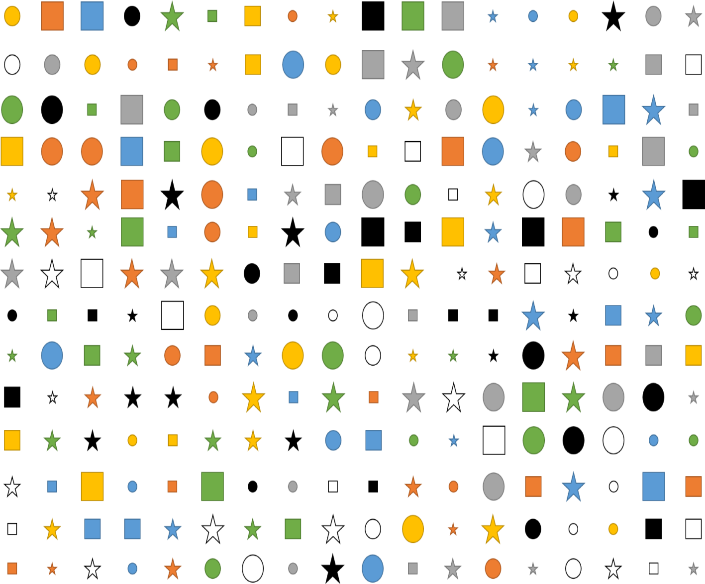


Figure 280—VTASC Code Design

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 coded example model is given Figure 281.

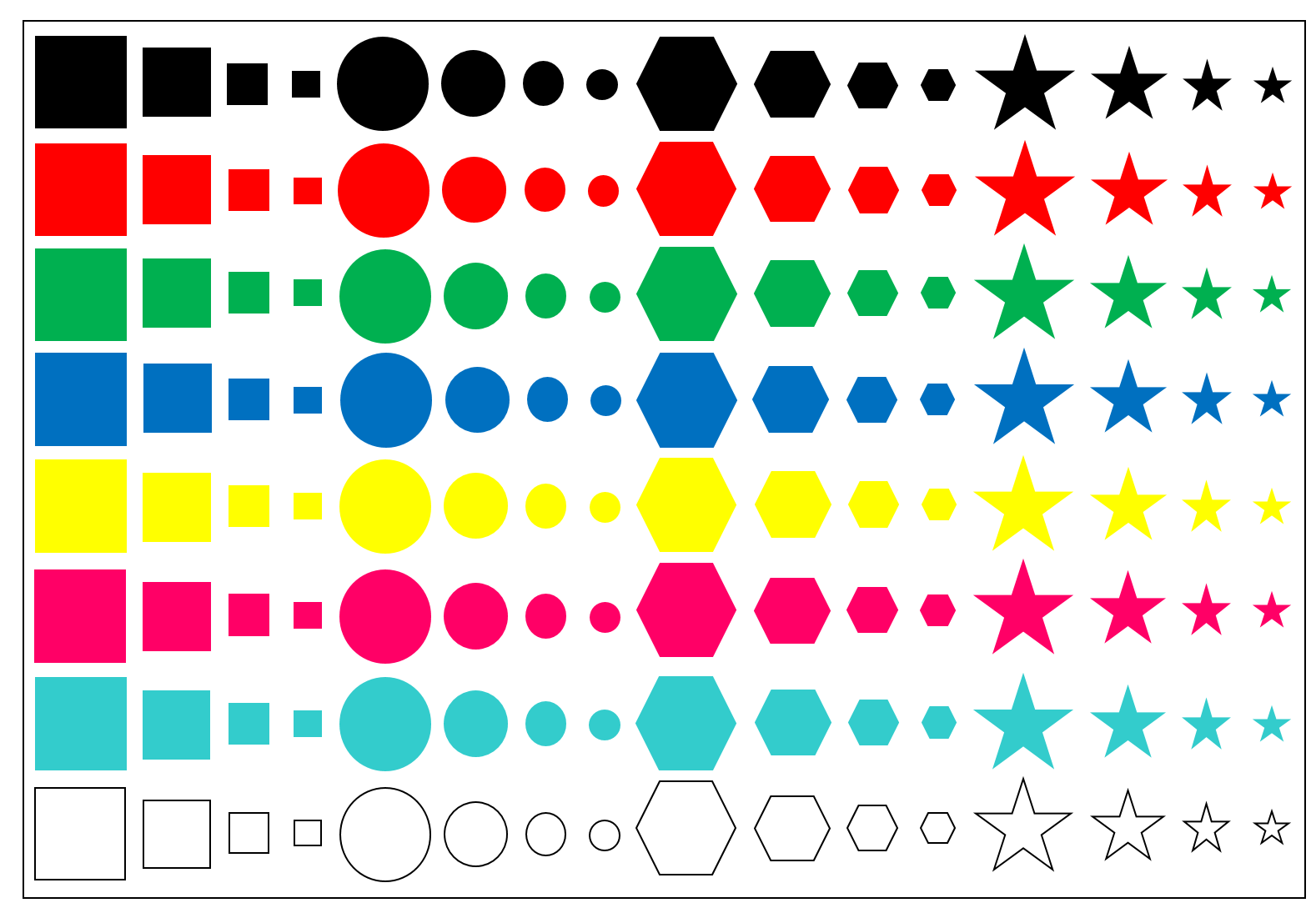


Figure 281—VTASC Coded Pattern Example

Table 206 describes the bits per symbol for VTASC code design.

Table 206 - 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 PQR1 describes the data bits to coding states mapping for VTASC code design.

Table PQR1 - 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 phyVTASCCodedArea. In partial screen mode, the number of horizontal and vertical blocks configurable over the PHY PIB phyVTASCAHSize, phyVTASCAVSize. In full screen mode, the number of horizontal and blocks estimable based on the screen size, resolution, aspect ratio, and the relative pixel ratio with 42 inches full HD display. 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 XYZ.

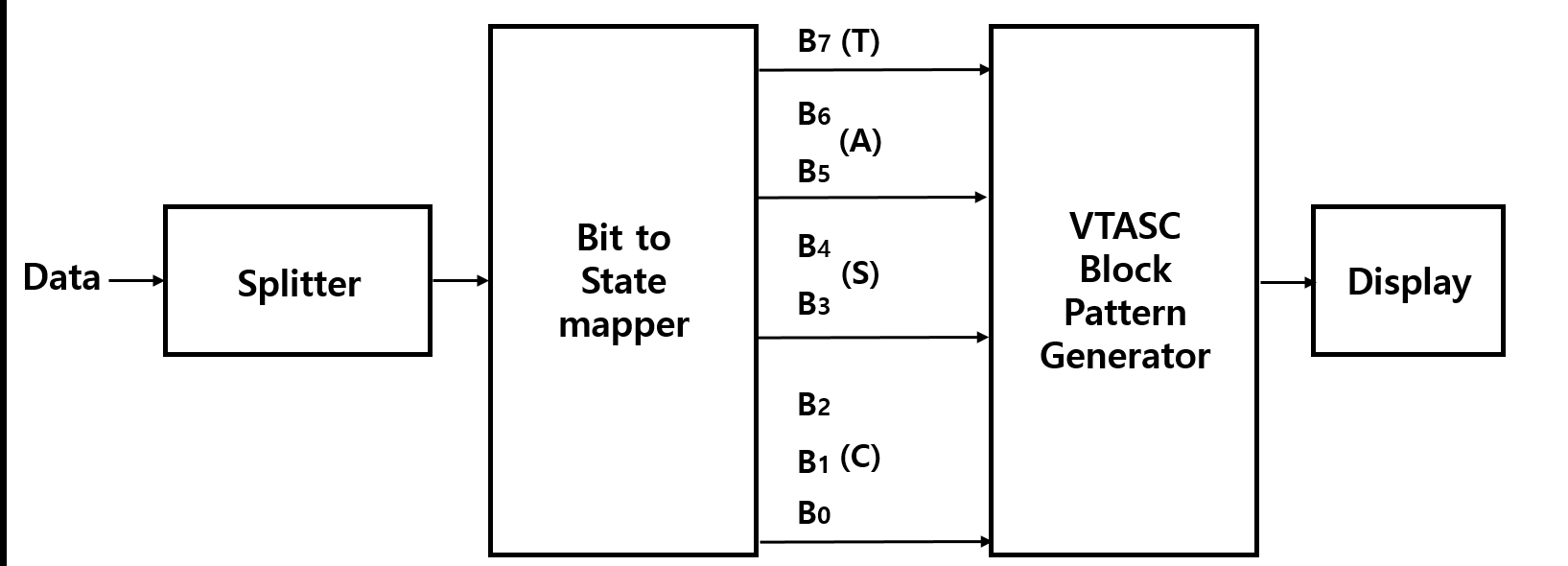


Figure XYZ – VTASC Data Encoder

The data coded on visual frame by overlaying visual patterns displays visual area. The rule to overlaying data and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. Table PQR2 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 PQR2 – 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 |
| 4 Color VTASC Code  (T = 2,A=4,S=4, C=4) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 448 Kbps |
| 8 Color VTASC Code  (T = 2,A=4,S=4,C=8) | None | 30Hz | RS(64,32)/ RS(160,128)/ None | 512 Kbps |
| 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 1

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
* NoofHorizontalBlocks = (1920/32) = 60 (Approx. to even for coding efficiency)
* NoofVerticalBlocks = (1080/32) = 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 = ((1920/32)\* (1080/32) \* 7 \* 30 \* 1) / 1) = 403200 = 390 Kbps (Approx.)

VTASC uses the two transparency levels in code design. The symbol to bitmapping for transparency is shown in Table PQR3.

Table PQR3 – Symbol to bit mapping for transparency level

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

VTASC block size represented by amplitude state in code design. The symbol to bitmapping for block size is shown in Table PQR4.

Table PQR4 – 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 symbol to bitmapping for shape is shown in Table PQR4.

Table PQR4 – 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 PQR6.

Table PQR6 – 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 | White |

**15.2.5 Asynchronous Communication**

Transmitter does not use any reference block for receiver synchronization with receiver. The purpose of spreading codes is 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 XYZ1.

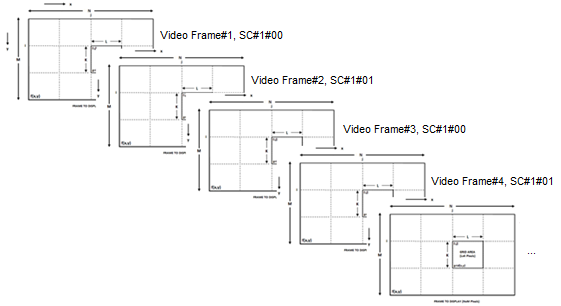


Figure XYZ1 – Video Frame Sequence SS Code Assignment

The receiver decoding for asynchronous communication and receiver error mitigation due to rolling effect is given in Annex M.2.

**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 PQR. 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 286. The region based optical clock rate and SS code con configurable over the PHY PIB phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhyVTACSSCode1FP00 to PhyVTACSSCode4FP01.

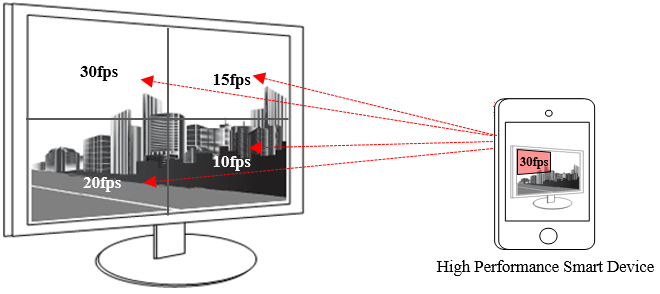
****

Figure 286 – 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 different distance range encoding is spreaded by pair of spread code as defined in Table PQR. In this case the transmitter built-in with camera features as shown in Figure 287 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 con configurable over the PHY PIB phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhyVTACSSCode1FP00 to PhyVTACSSCode4FP01.

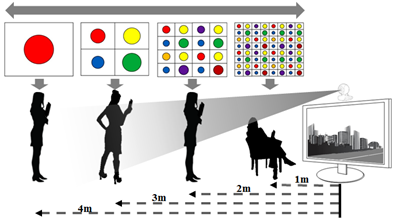
****

Figure 287 – Distance Adaptive Data rate Control

For multiuser distance adaptive is supported combining with receiver frame rate adaptive multirate controller. The receiver frame rate adaptive multirate controller is described in subclause15.2.6.2.

**15.3 SS2DC Specification**

The Sequential Scalable 2D Code (SS2DC) works with different 2D codes (like QR, Color Code, VTASC, A-QL, etc.) organized in a combination of one or more codes sequentially in row and column manner. The SS2DC PHY supported data rates and operating conditions are shown in PHY VI operating modes Table 116.

# **15.3.1 SS2DC Reference Architecture**

The reference PHY architecture for SS2DC is illustrated in Figure 279. 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.

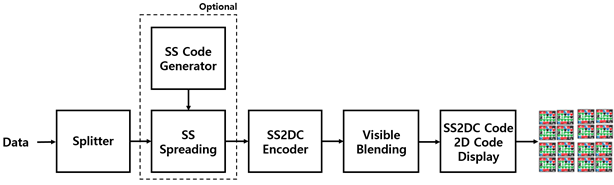
****

Figure 288 – Reference architecture for SS2DC PHY System

The spread spectrum used with SS2DC to have effective asynchronous and communication, distance adaptive scalable data rate controlled OWC. The SS2DC is used for enhanced display to camera communication in the real-time application usage scenario. 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

The receiver specific information for SS2DC Data Decoder is given in Annex M.4.

**15.3.2 Spread Spectrum**

The spread spectrum used with SS2DC 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.

**15.3.3 SS2DC Code Design**

SS2DC is a two dimensional design using different 2D codes for improved OWC throughput by sequentially arranging horizontally and vertically as shown in Figure 289. 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 289 – SS2DC Code Design Examples

**15.2.4 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 ZYX.

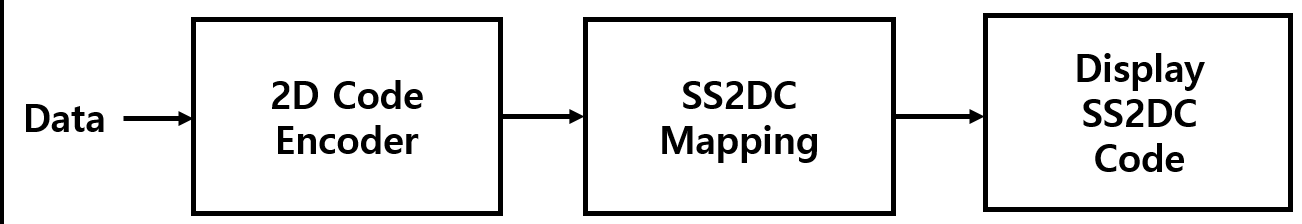


Figure ZYX: Sequential Scalable 2D Code Data Encoder

The data coded on visual frame by overlaying visual patterns displays visual area. The rule to overlaying data and data rate achievement vary based on the kind of display used to design the transmitter and the distance between transmitter and receiver. Table ZYX describes the data rate supported by SS2DC code design with QR code. The QR code size must be minimum of scanning distance / 10 (in inches). The table data rate differs according to the 2D code specification (2D code encoders not scope of this standard).

Table ZYX – 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 |
| 4x4 SS2DC | None | 2DCodeDecodingRate | RS(64,32)/ RS(160,128)/None | 368 Kbps |
| 2x2 SS2DC | None | 2DCodeDecodingRate | None | 46 Kbps1 |
| 4x4 SS2DC | None | 2DCodeDecodingRate | None | 184 Kbps1 |

1 where spreading factor is 1

The data rate calculation is described below,

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

Where,

* CodeLength is spreading code factor used for SS spreading (1 for without SS)
* 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 without SS spreading with QR code (The maximum data capacity is 2953 bytes) and DataRate = 4\* (2953 \* 8)\* 1 \* 1) / 1) = 94494 = 92 Kbps (Approx.).

**15.3.4 Asynchronous Communication**

The SS2DC PHY transmitter supports asynchronous communication by spreading data stream using SS code. The subclause 15.2.5 contains more information about asynchronous communication.

**15.3.6 Scalable Bitrate Controller**

The SS2DC PHY Transmitter with SS2DC designed with built-in Scalable bitrate Controller. The subclause 15.2.6 contains more information about scalable bitrate controller.

**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 116.

# **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 298.

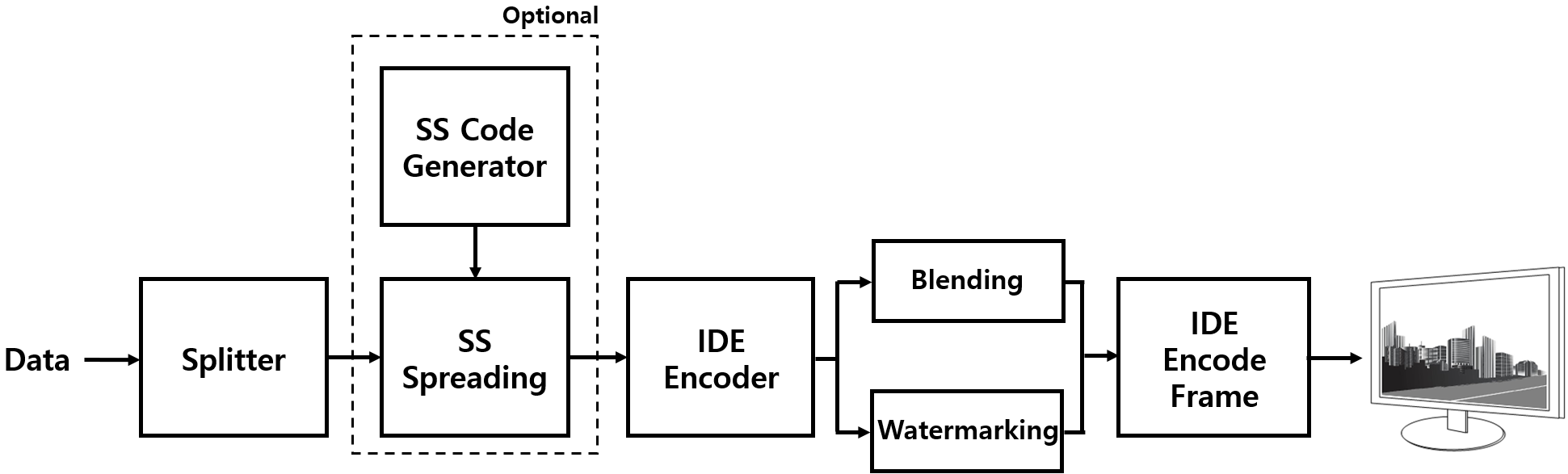
****

Figure 298 – 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 works on,

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

The receiver specific information for IDE data decoder is given in Annex M.5.

**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 ZYX.

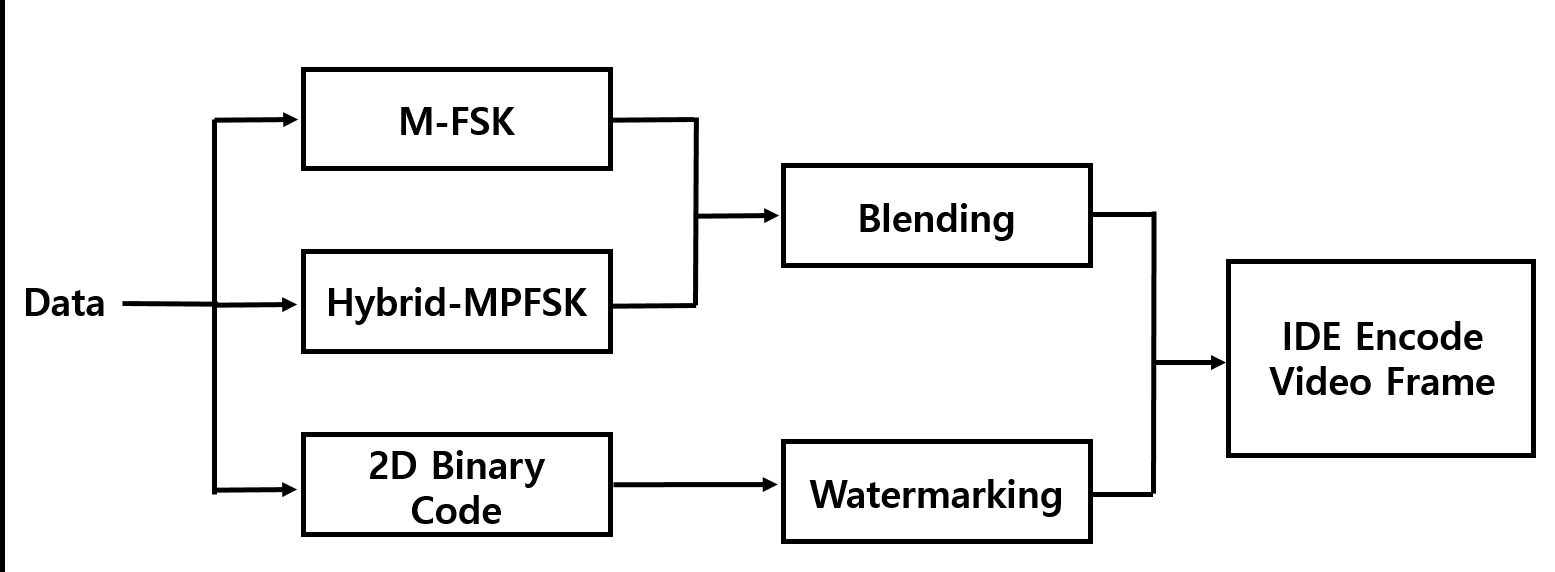


Figure ZYX – IDE Data Encoder

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 ZYX1 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 ZYX1 – 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 |
| HYBRID-PFSK-BLEND | None | 30Hz | RS(64,32)/ RS(160,128)/None | 32 Kbps |
| 2DBIN-WM | None | 30Hz | RS(64,32)/ RS(160,128)/None | 128 Kbps |
| 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 1

**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 ZYX1. The number of frequencies used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency. The M-FSK encoded symbol generates 2D pattern of MxN pixel block to blend with visual scene to be rendered on screen.

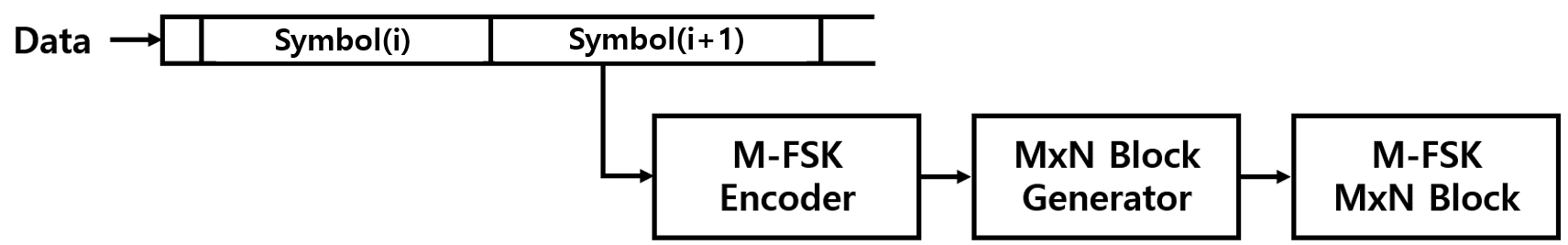


Figure ZYX1 – IDE M-FSK Data Encoder

The 4 bit symbol to bitmapping for M-FSK is shown in Table PQR6.

Table PQR6 – Symbol to bit mapping for M-FSK

|  |  |
| --- | --- |
| **Symbol Bits (B4, B2,B1, B0)** | **Frequency Mapping** |
| 0000 | f0 |
| 0001 | f1 |
| 0010 | f2 |
| 0011 | f3­ |
| …. | … |
| 1101 | f13 |
| 1110 | f14 |
| 1111 | f15 |

**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 ZYX2. The number of frequencies and phase used to map data shall be configured over the PHY PIB attribute phyIDEFSKNoFrequency, and phyIDEPSKNoPhase. The Hybrid-MPFSK encoded symbol generates 2D pattern of MxN pixel block to blend with visual scene to be rendered on screen.

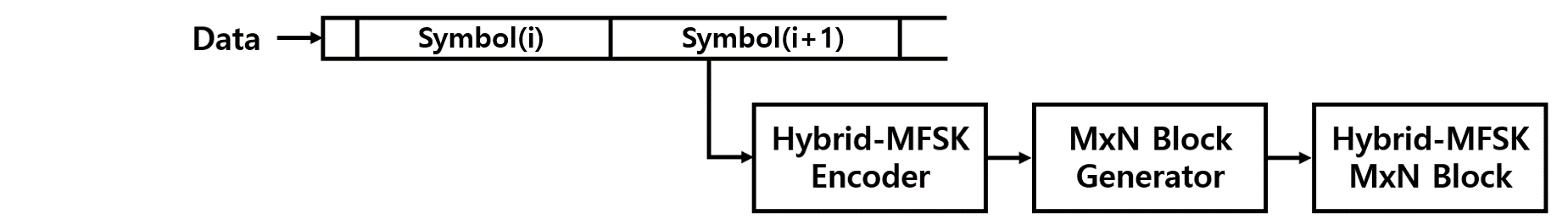


Figure ZYX2– IDE Hybrid-MPFSK Data Encoder

The 5 bit symbol to bitmapping for Hybrid-MPFSK is shown in Table PQR7.

Table PQR7 – Symbol to bit mapping for Hybrid-MPFSK

|  |  |  |
| --- | --- | --- |
| **Symbol Bits (B4, 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 |

**15.4.3.3 2D Binary Code**

The 2D binary encoder, estimates the number of MxN pixels blocks in visual display. The horizontal and vertical encoding area pixel ranges shall be configured over the PHY PIB attribute phyIDEENCHozAreaSize, phyIDEENCHozAreaSize. The total number of encoding blocks sized data bits spreaded with SS sequence is extracted and creates two dimensional binary coding according number of row MxN blocks and number of column MxN blocks estimated as shown in as shown in Figure ZYX3.

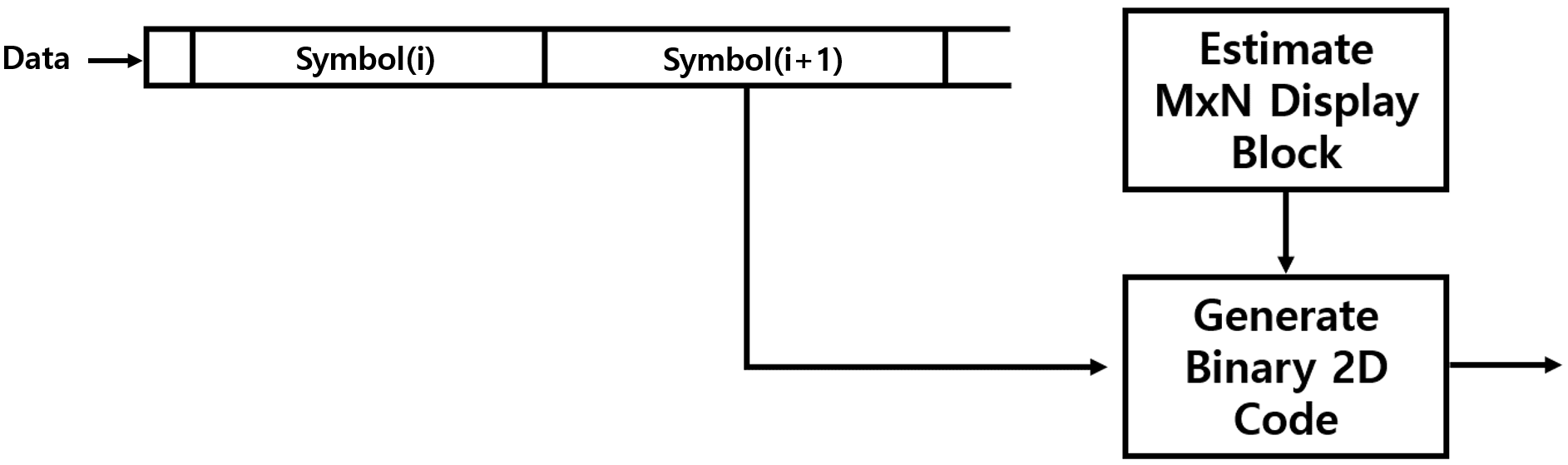


Figure ZYX3– IDE 2D Binary Data Encoder

**15.4.3.4 Invisible Data Blending**

**Process of stitching two different images as a single image**

The M-FSK or Hybrid-MPFSK coded MxN block is blended with in visual frame sequential in every block by row to column manner and rendered on display screen as shown in as shown in Figure ZYX4.

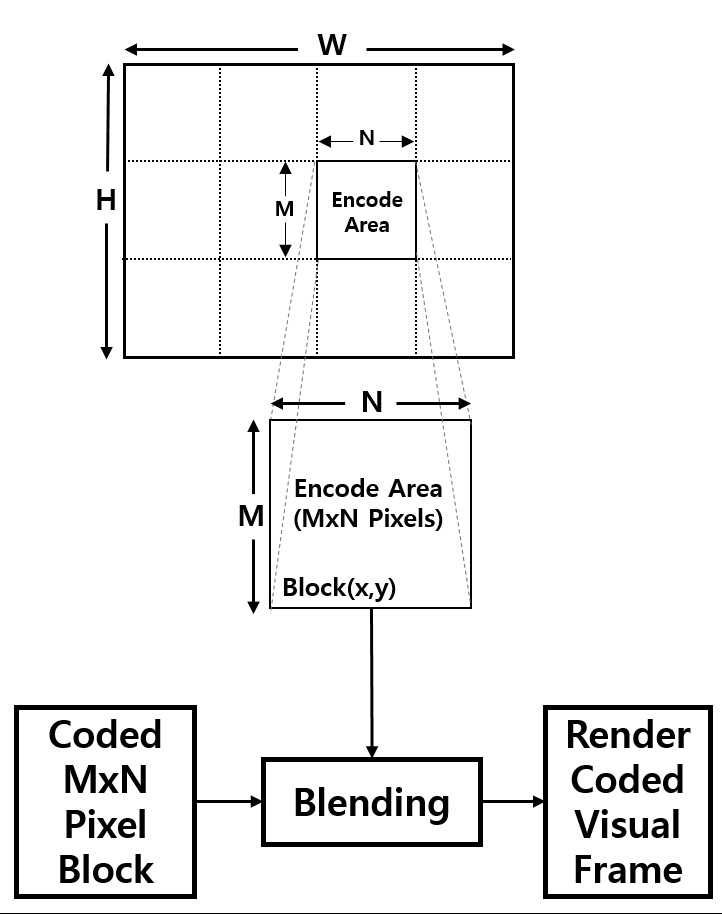


Figure ZYX4– IDE Encoder Blending

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 binary 2D coded MxN block is watermarked with in visual frame sequential in every block by row to column manner and rendered on display screen as shown in as shown in Figure ZYX5. The invisible watermarking utilize the human eye visual imperceptibility in middle-high frequency component of the visual frame. The frequency based watermarking is dominant on for invisible data embedding. The frequency based invisible data embedding procedure is not part of the standard.

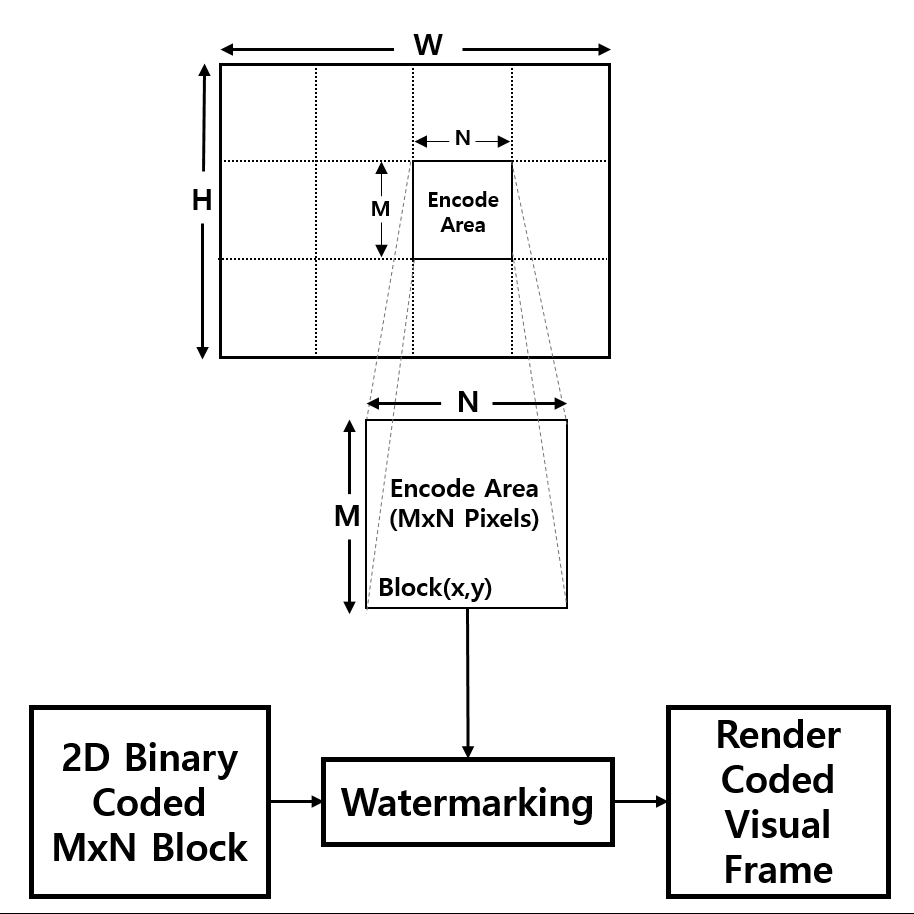


Figure ZYX5– IDE Encoder - Watermarking

**15.4.4 Asynchronous Communication Mode**

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.

**Annex M (Normative)**

**M.1 VTASC Decoder**

VTASC data decoding is shown Figure M.1.

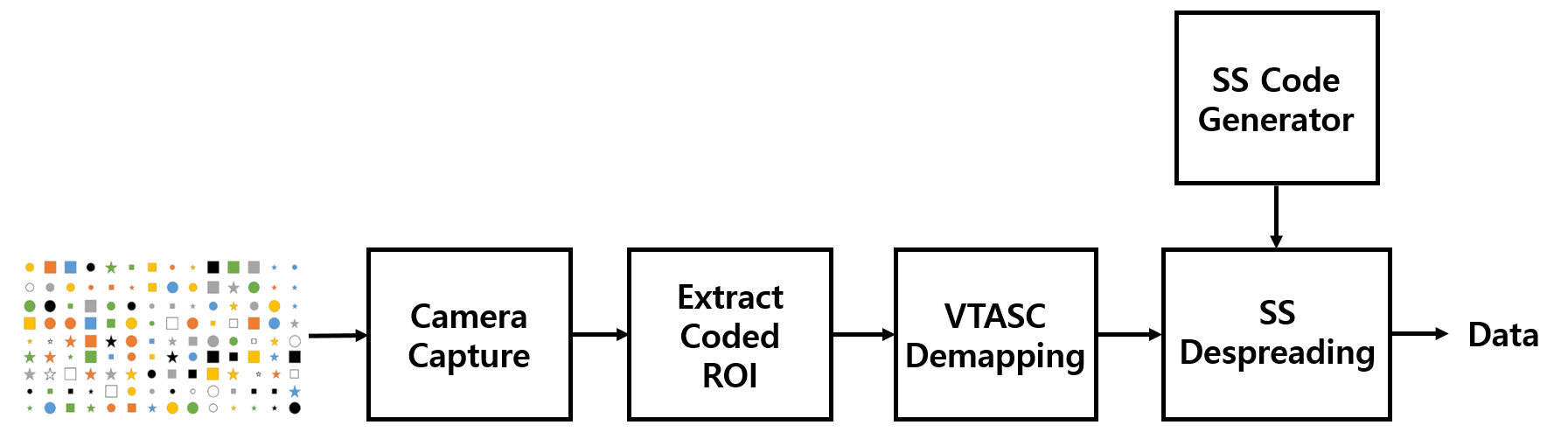


Figure M.1 – VTASC Receiver Functional Block Diagram

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.

**M.2 Asynchronous Communication**

The optical clock rate and SS codes are configurable over the PHY PIB phyVTACScalRegion1OpticalClockRate to phyVTACScalRegion4OpticalClockRate, PhyVTACSSCode1FP00 to PhyVTACSSCode4FP01.The receiver synchronize using SS code (any one of the four pair SS code at first time) and decoded the data.

In the next consecutive capture frames, if camera receive the same frame, for example #N video frame receive twice, then receiver will discard video frames by despreading the next code in SS pair code detection fails. If the current processing frame detect pair of SS codes in a single frame, then that frame is rolling effect fault capture frame and be discarded.

**M.3 Angle Free Communication**

The angle free communication between transmitter and receiver is shown in Figure M.3.



Figure M.3 – 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.

**M.4 SS2DC Decoder**

SS2DC data decoding is shown Figure M.4.

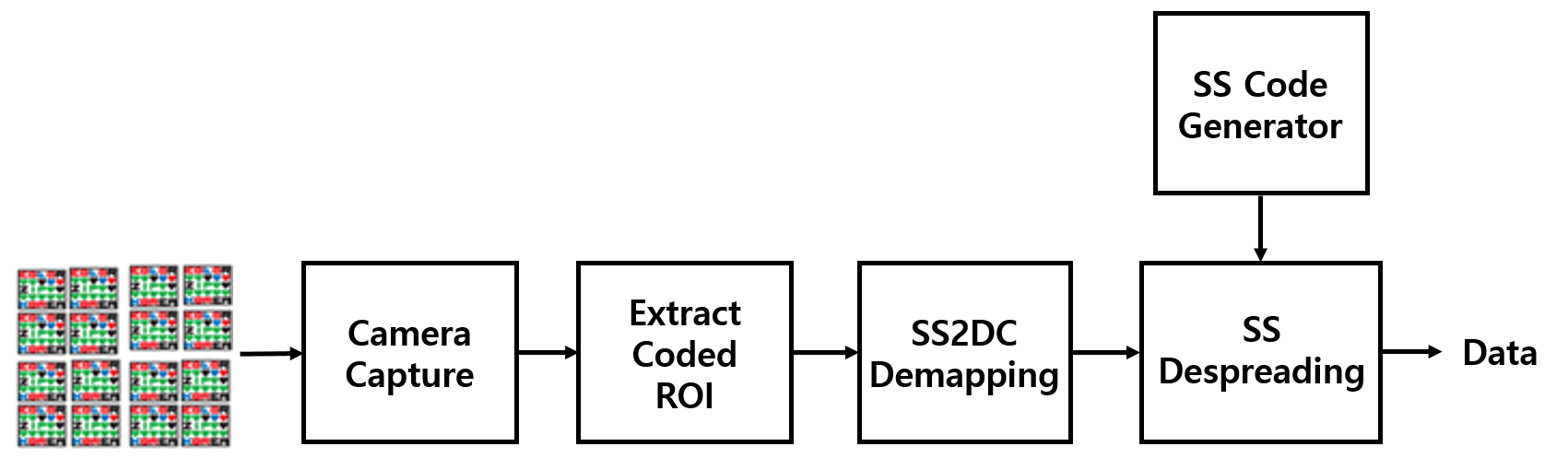


Figure M.4 – SS2DC Receiver Functional Block Diagram

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.

**M.5 IDE Decoder**

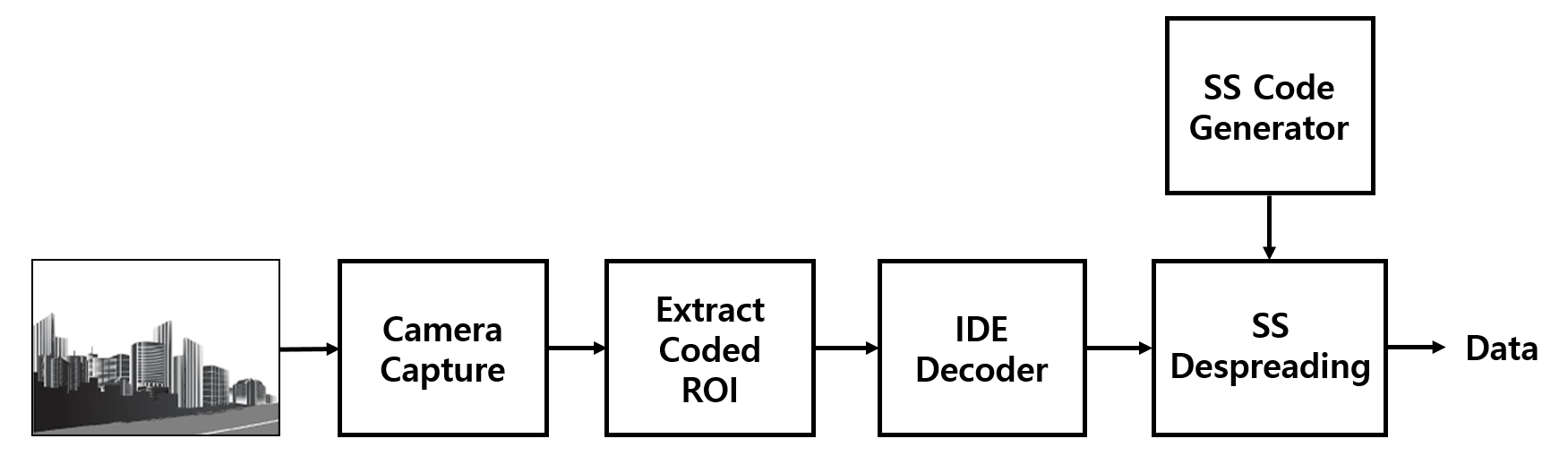
IDE data decoding is shown Figure M.5.

Figure M.5 – 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 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 water marking uses the high frequency visual coefficients to embed the data on visual frame so need to use frequency based frequency based watermark decoding is used to decode the data. The data embedded on display is SS Coded data so SS decoding is applied to recover original data from the visual sequence.