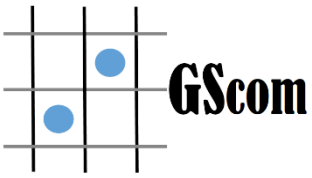


Broadband Wireless 4096-QAM Modem

System Document

Broadband Wireless 4096-QAM Modem

Table of Contents

1. Introduction

The need to increase the channel data rate in data communication systems without increasing the signal bandwidth drives the development towards a more spectrally efficient modulation formats. QAM modulation is widely used in digital wireless communication systems for achieving high data transmission rates over a relatively narrow signal bandwidth. The 4096-QAM is currently one of the most advanced modulation technology used in the wireless communication system.

GScom developed a broadband wireless 4096-QAM modem that achieves Giga bit range high speed data communication for both microwave and mm-wave radio systems. The broadband wireless modem is a single carrier QAM modem for frequency efficient modulation and can be used in broadband wireless communication systems for fixed wireless backhaul communication, cellular access networking, or small cell backhaul communication.

The broadband wireless modem was developed to support modulation levels up to 4096-QAM, to be very robust to phase noise, and for point-to-point or point-to-multi point microwave and mm-wave radio systems. The 4096-QAM modem was designed to support various modulation technologies (such as 4-QAM, 16-QAM, 64-QAM, 256-QAM, 1024-QAM, and 4096-QAM), Low Density Parity Check (LDPC) coding, Reed-Solomon Coding, Adaptive Coding and Modulation (ACM), Hybrid Auto Request (HARQ), etc.

GScom carried out a performance test of the GScom's modem in a fading environment using the guide line and test parameters (please refer to Table 1 in Section 4.2.1.1) specified in the specification (ETSI EN 302 217-2-1). GScom's modem yielded an excellent performance in terms of the "receiver sensitivity" in the frequency selective fading environment specified in the specification.

2. System Description

This section provides a system overview and functional description of each system's operational mode.

2.1 System Overview

The broadband wireless modem (or small cell backhaul modem) is a base band modem and can support any radio frequency communication system. The modem supports very high speed data communication capability with very low latency to meet the requirements for next generation wireless communication systems.

The main features of the 4096-QAM Modem IP Core are listed below:

- Various modulation schemes: 4QAM/16QAM/64QAM/256QAM/1024QAM/4096QAM
- Various channel bandwidth: 14/28/56/112MHz
- Maximum symbol rate on each of the bandwidth: 12.6/25.2/50.4/100.8M Symbols/second
- Real time Gray code mapper/De-mapper with very low latency LLR decoding
- Reed-Solomon Codec with (N,K,S+1) format used;
    - Block length; N = 255
    - Data length; K is variable
    - Number of parity bits; S = 2t, where t is the number of correctable bits
    - Parallel pipelined structure Reed-Solomon (R-S) code
- LDPC (low density parity check) codec;
    - Code length: 4320/8640 bits, & Coder rate: 0.5 & 0.75
    - Architecture Aware Parallel TDMP architecture
    - Horizontal and Vertical Partitioned LDPC
    - Optimized node degree distribution LDPC
- Fractional spaced DFE (decision feedback equalizer) combined with carrier recovery.
- Joint estimation/correction of carrier recovery and equalizer
- Pulse Shape Filter (PSF): Squared-Root-Raised-Cosine with 96 taps
- Roll-Off factor in PSF: 10% - 18%
- ACM, HARQ, ATPC, XPIC, etc.,

The broadband wireless 4096QAM modem (or small cell backhaul modem) performs four operational modes to get the data communication from the initial operation of acquisition:

- Carrier frequency acquisition mode
- Symbol synchronization mode
- Frame synchronization mode
- Data communication mode

The broadband wireless modem shows excellent performance such as:

- Very reliable structured system operation architecture
- Excellent performance of phase noise mitigation by combining DFE and Carrier Recovery
- Excellent coding gains from R-S and LDPC codes
- Excellent receiver sensitivity over all the modulation schemes (or formats)
    - Receiver sensitivity (SNR) in dB for 1E-6 (in AWGN)

Modulation	QPSK	16-QAM	64-QAM	256QAM	1024QAM	4096QAM
SNR(dB)	6	13	20	25	28	32

2.1.1. High Level System Block Diagram

A variety of data transport types can be used as the interface to the broadband wireless modem (or small cell backhaul modem) such as E1/T1 ports, Ethernet port, and General Purpose Interface (GPI) port. Figure 1 shows the high level system block diagram of the small cell backhaul modem showing the input and output ports of the communication system.

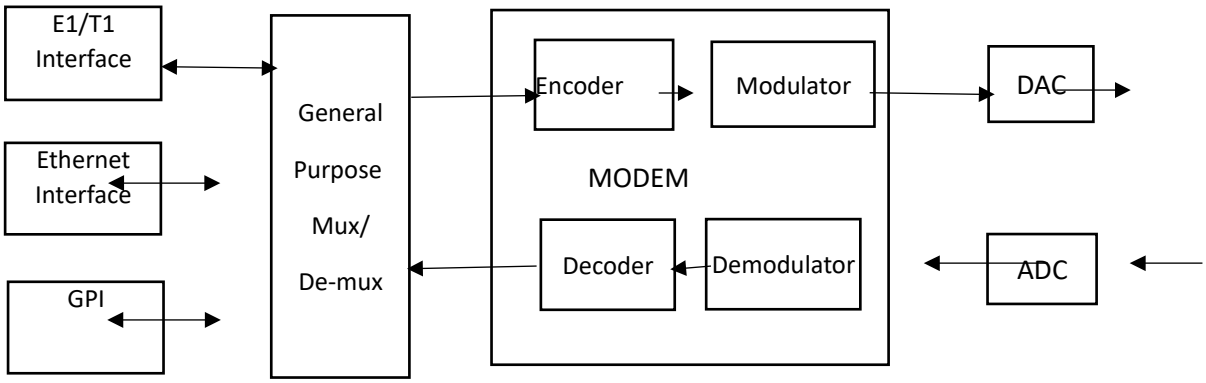


Figure 1: High Level Block Diagram of Wireless Communication System

For the transmit direction (TX), the user data is obtained from the physical interface and transferred to the general purpose multiplexer (GPM). The GPM organizes data from multiple sources and builds data frames to be transmitted and transfers the data frames to the encoder of the modem. The modem encodes and modulates the data, and then sends the modulated digitally formatted data to the digital-to-analog converter (DAC) which converts the data into an analog signal for radio frequency modulation and then sends the analog data to the transmit antenna.

For the receive direction (RX), the analog-to-digital converter (ADC) receives the analog signal, converts it into a digital signal and then sends it to the modem. The modem demodulates and decodes the digital data, and then sends it to GPM. The GPM receives the digitally formatted data from the modem and then distributes the data to the corresponding addressed user data interface.

2.1.2. High Level System Operation

The broadband wireless 4096QAM modem (or small cell backhaul modem) performs four operational steps to set up the data communication link following initialization of the communication system- coarse carrier acquisition, symbol synchronization, frame synchronization and data communication.

2.1.2.1. Coarse Carrier Acquisition and AGC Operation Mode

Oscillators in two separate communication systems (transmitter and receiver sides of each system) may have more than a few tens of kHz of frequency offset between them at the initial stage of communication. The coarse carrier acquisition process achieves the frequency offset reduction within a few KHz using feed-forward carrier estimation/correction techniques. In the carrier acquisition mode the automatic gain control (AGC) process keeps the ADC input signal at a certain level to avoid signal clipping. The coarse carrier acquisition process employs the carrier acquisition frame and then takes the cross correlation of two consecutive and repeated preamble symbol blocks of the carrier acquisition frame, and then obtains the offset frequency estimate by taking the arc tangent of the averaged correlation value.

The coarse carrier acquisition uses the preamble symbols in the carrier frequency acquisition frame. The carrier frequency acquisition frame size is 4096 symbols. The carrier frequency acquisition frame consists of 4096 symbols of repeated (0, 2) QPSK symbols as can be seen in Figure 2.

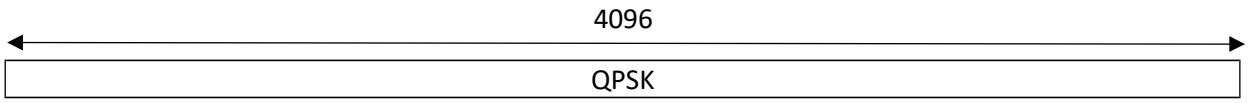


Figure2. Carrier frequency acquisition frame configuration

2.1.2.2. Symbol Synchronization and AGC Operation Mode

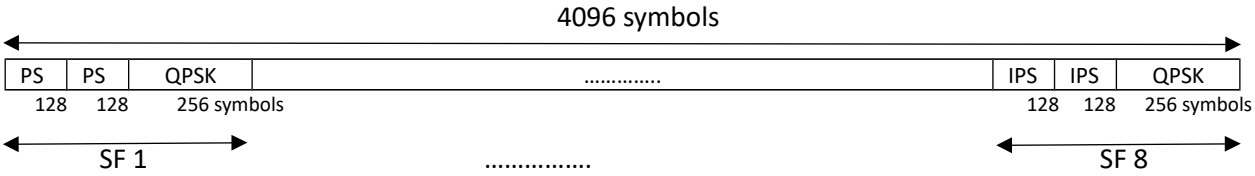
Symbol synchronization uses the preamble symbols in the acquisition frame and calculates the timing error using the following delayed locked loop technique:

$$\text{Timing Error} = (s(n+2) - s(n)) * s(n+1)$$

Where, s(n) is a complex symbol signal and n is the symbol number of the timing sequence. The type II second-order loop filter uses the timing error estimate to track the symbol timing. The phase of the symbol timing drives the ADC output timing. Note that the symbol synchronization block also works together with the AGC which keeps the input signal level constant with a second-order loop filter.

The symbol synchronization mode uses the acquisition frame. The acquisition frame size is 4096 symbols. The Acquisition Frame consists of eight sub frames where each sub frame consists of identical 256 preamble symbols and 256 symbols of repeated (0, 2) QPSK symbols from sub-frame 1 to sub-frame 7. The last sub-frame (sub-frame 8) has a sign inverted 256 symbols of preamble, instead of the same 256 preamble symbols as those in the previous sub-frames. The 256 preamble symbols consist of two identical blocks of 128 preamble symbols. The block of 128 preamble symbols are obtained from the 256 Gold Sequence Generator after taking the QPSK modulation of the 256 Gold sequence. The 256 Gold Sequence Generator block consists of two 8-bit shift registers with several XOR gates between those two registers.

Figure 3 illustrates the Acquisition frame configuration.



Note: PS: Preamble symbols, IPS: inverted signed preamble symbols, SF 1: sub-frame 1, SF8: sub-frame 8

Figure3. Acquisition frame configuration

2.1.2.3. Frame Acquisition and Carrier Frequency Acquisition Mode

The frame acquisition and carrier frequency acquisition process also uses the acquisition frame shown in Figure 3 in the symbol synchronization section. The acquisition frame consists of eight sub-frames. Each sub-frame, from sub-frame 1 to sub-frame 7 in the acquisition frame consists of identical 256 preamble symbols. The last sub-frame (sub-frame 8) has a sign inverted 256 preamble symbols, instead of the same 256 preamble symbols as those in the previous sub-frames. The 256 preamble symbols consist of 128 repeated (0, 2) QPSK symbols. The 256 preamble symbols consist of two identical blocks of 128 preamble symbols. The 128 preamble symbols are obtained from the 256 Gold Sequence Generator and then taking the QPSK modulation on the 256 Gold sequence data.

The preamble symbols in each of the first seven sub-frames of the acquisition frame are used to identify the timing information of the sub-frame. The preamble symbols in the last sub-frame (sub frame 8) in the acquisition frame are used to identify the timing information of the acquisition frame. Fine carrier frequency acquisition can be obtained by applying a feed-forward carrier frequency estimation/correction technique to the 256 symbols of the repeated (0, 2) QPSK symbols in each sub-frame in the acquisition frame.

Continuous fine carrier tracking is performed using the phase locked loop (PLL) in part of analog baseband modulation process through a voltage controlled oscillator (VCO).

2.1.2.4. Data Communication Mode

After obtaining the symbol timing, frame timing and carrier frequency acquisition, data communication takes place using the Data Frame. The Data Frame is 4744 symbols long and consists:

128 preamble symbols used for the training equalizer

128 pilot symbols used for the training equalizer and the channel estimation

16 physical layer control (PLC) symbols used for communication with the upper layer

16 automatic coding and modulation (ACM) symbols used for communication with the upper layer

2160 Data symbols

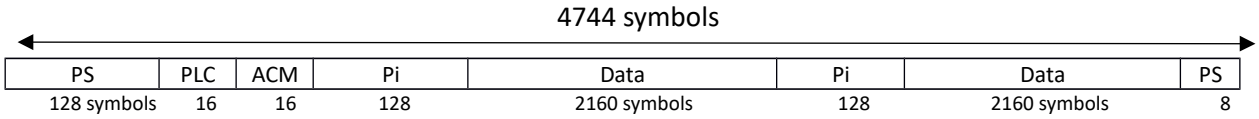
128 pilot symbols used for the training equalizer and the channel estimation

2160 Data symbols

8 preamble symbols used to take care of the delay in the equalizer

The preamble, pilot, PLC, and ACM symbols are all QPSK modulated symbols for reliable control communication. The data symbols are QAM symbols for high speed data communication. Note that the data communication block also works together with the phase recovery for phase noise mitigation and the AGC which keeps the input signal level constant using a second-order loop filter.

Figure 4 illustrates the data frame configuration.



Note:

PS: Preamble symbols for frame tracking

PLC: Physical layer control for communication with upper layer control software

ACM: Adaptive control and modulation parameter for communication with upper layer control software

Pi: pilot signal used for channel condition estimation and carrier frequency tracking

Data: Coded symbol data for data communication

Figure 4. Data Frame configuration

3. Functional Description

3.1 High Level System and Function Block Description

As seen in Figure 5, the high level modem system consists of a transmitter (Tx) and a receiver (Rx) section. The encoder and modulator in the transmitter receives data from the general purpose multiplexer (GPM) and creates the data frame for transmission through the DAC and analog radio frequency devices to the antenna. The encoder consists of two forward error correction codes; one is a Reed-Solomon (R-S) encoder and the other is a low density parity check (LDPC) encoder. GScom uses concatenated encoders of R-S and LDPC for high order modulation QAM signals and a single LDPC encoder for low order QAM signals. The encoder adds parity symbols and bits to inputted data symbols and bits, respectively, to make R-S encoded data and LDPC encoded data formats for forward error correction. The encoded data bits go to a Mapper to map those bits into symbols according to the selected modulation scheme. The symbols are then modulated into real signals (I) and imaginary signals (Q). The digital format of the airframe is converted to analog signals by a digital to analog converter (DAC) and is then sent for transmission.

At the receiver, the analog to digital converter (ADC) receives the analog signal, converts the received analog signal to a digital signal and then sends it to the demodulator block. The digital signals are processed by the demodulator, sent to the De-mapper for converting the demodulated symbols to bits according to the selected modulation scheme, and then sent to the decoder for error correction and for decoding. The decoder consists of two forward error correction codes; one is a Reed-Solomon (R-S) decoder and the other is a low density parity check (LDPC) decoder. GScom uses concatenated decoders of R-S and LDPC for high order modulation QAM signals and a single LDPC decoder for low order QAM signals.

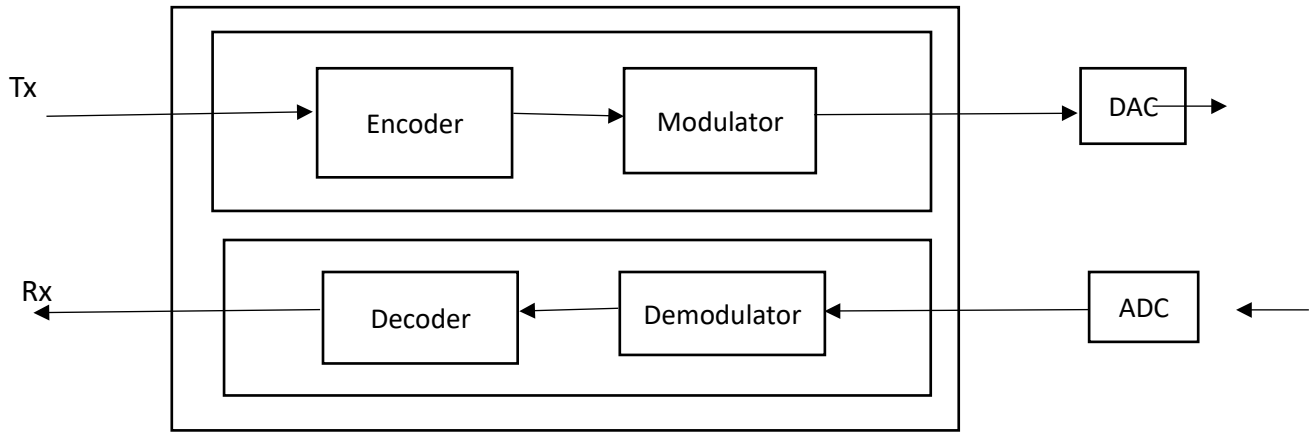


Figure 5, High level modem system

3.2 Encoder Block Description

GScom uses two error correction codes; an R-S code and a LDPC code. The R-S code is a non-binary code and is well suited for taking care of burst errors. The LDPC code is a binary code and is suited for taking care of identically and independently distributed (iid) errors. GScom uses concatenated coders of R-S and LDPC for high order modulation QAM signals and a single LDPC coder for low order QAM signals.

Figure 6 shows the functional block diagram of the encoder/decoder with adjacent functional blocks (denoted as green boxes).

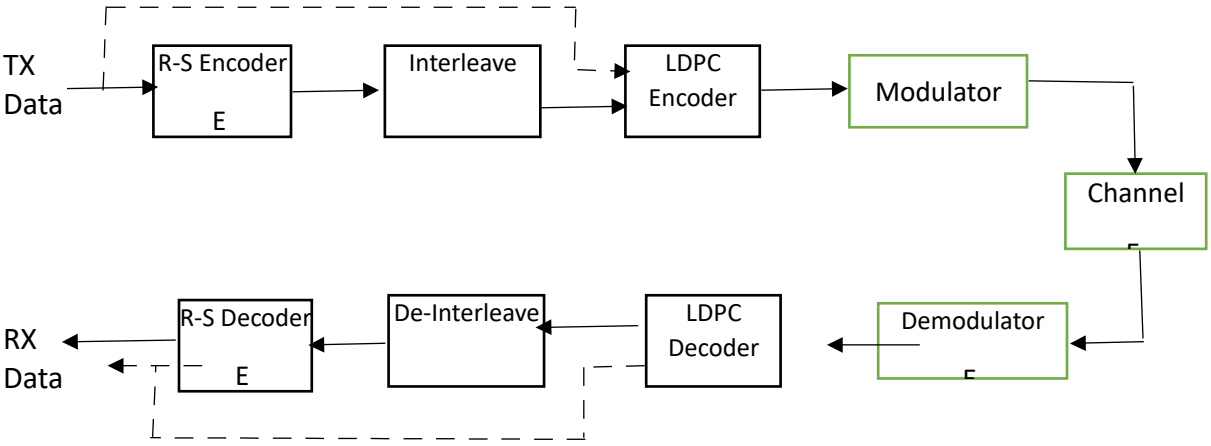


Figure 6: Encoder/Decoder Functional Block Diagram

GScom uses a dotted signal line path to encode/decode for lower order modulation signals (4 to 256-QAM signal) and a solid signal line path to encode/decode for high order modulation signals (1024-QAM and 4096-QAM).

The encoder block consists of the following function blocks:

- Reed-Solomon Encoder
- Low Density Parity Check (LDPC) Encoder
- Mapper

3.2.1 Reed-Solomon Encoder

R-S codes are non-binary cyclic codes with symbols made up of m-bit sequences, where m is any positive integer having a value greater than 2. A R-S code is a block code, i.e. the message to be transmitted is divided up into separate blocks of data. Each block of data has parity protection information added to form a self-contained code word. An R-S code is a systematic code, i.e. the

encoding process does not alter the message symbols and the protection symbols are added as a separate part of the block.

The values of the message and parity symbols of an R-S coder are the elements of a Galois field. Thus for a code based on m-bits symbols, the Galois field has 2^m elements. GScom uses (255, k, t) R-S code so that the k=255-2t symbols of the input packet will be extended with 2t parity symbols to produce a code block length of 255 symbols. For this code, the Galois filed has 256 (m=8) elements and GScom used the following field generator polynomial to construct the 256 elements of GF(256);

$$p(x)=x^8+x^4+x^3+x^2+1 \quad \text{---------- (1)}$$

3.2.1.1 The Code Generator Polynomial

An (n, k) R-S code is constructed by forming the code generator polynomial g(x), consisting of n-k=2t factors, the roots of which are consecutive elements of the Galois field. Choosing the consecutive elements ensures that the distance properties of the code are maximized.

Thus the code generator polynomial of a t-error-correcting R-S code of length 2^m -1 takes the following form;

g(X) = (X+ α) (X+ α^2) …. (X+ α^{2t}) ---------- (2)

where α is a primitive element of GF(2^m).

Note that the primitive element α is the root of the code generating polynomial.

3.2.1.2 Reed-Solomon Encoding

The k information symbols that form the message to be encoded as one block can be represented by a polynomial M(x) of order k-1, so that;

$$M(x)=M_{k-1}x^{k-1}+\ldots+M_1x+M_0 \ ,$$

where each of the coefficients $M_{k-1}\ldots..M_1,M_0$ is an m-bit message symbols and is an element of GF(2^m).

To encode the message, the message polynomial is multiplied by x^{n-k} and the result is divided by the generator polynomial, g(x). Division by g(x) produces a quotient q(x) and a remainder r(x) which is of a degree up to n-k-1. Thus:

$$\frac{M(x)x^{n-k}}{g(x)}=q(x)+\frac{r(x)}{g(x)} \quad \text{---------- (3)}$$

Having produced the r(x) by division, the transmitted code word (R-S code word), T(x), can then be formed by combining M(x) and r(x) as following:

$$T(x) = M(x)x^{n-k} + r(x) = M_{k-1}x^{n-1} + \ldots + M_0 x^{n-k} + r_{n-k-1}x^{n-k-1} + \ldots + r_0 \quad \text{-------- (4)}$$

which shows that the R-S code word is produced in the required systematic form.

Adding the remainder, r(x), ensures that the encoded message polynomial will always be divisible by the generator polynomial without a remainder.

3.2.1.3 Implementation of Reed-Solomon Encoder

R-S encoding is obtained by the division of the message polynomial M(x) by generating polynomial g(x) to produce the parity symbols r(x), and then by combining the message and the remainder as seen in the equation (3) and (4). The division of the M(x) by the g(x) can be accomplished by a polynomial division process in the R-S field.

GScom uses a hardware encoder to perform the polynomial long division process. The hardware encoder operates on pipelined data, so the division calculation is made using the message symbols one at a time as they are presented. The pipelined division calculation is performed using the conventional linear feedback shift register (LFSR) encoder which enables the encoding to perform a real time processing.

3.2.2 LDPC Encoder

Low-Density-Parity-Check (LDPC)is the one of the most powerful linear block codes in terms of performance (coding gain) and its decoding algorithm is inherently parallel which is attractive for high-speed applications.

The Turbo-Decoding-Message-Passing (TDMP) algorithm is one of the LDPC decoding algorithms. The TDMP processes the rows of the parity check matrix (H) sequentially which results in a lower processing throughput compared to the LDPC-like schedule. The TDMP algorithm can be parallelized by embedding some structure into the parity check matrix (H) that allows the parallel processing of multiple rows without communicating messages between the nodes corresponding to these rows. The resulting matrix (H) is composed of several bands of non-overlapping rows, referred to as a Parallel Turbo-Decoding-Message-Passing (P-TDMP) algorithm. The P-TDMP algorithm was chosen for decoding the Architecture Aware Low Density Parity Check (AA-LDPC) codes due to its faster convergence speed and corresponding throughput advantage over the standard LDPC and the standard TDMP codes. The modem employs a reduced complexity message computation mechanism (Soft-Input-Soft-Output (SISO) decoder), which is free of lookup tables and features a programmable network for message interleaving based on the code structure.

To design finite length LDPC codes that can be efficiently implemented on the target multi-processor architecture for high speed communication, we utilize a design process that combines the optimal

degree distribution for asymptotic performance, characteristics of structured sub-matrices, finite length code optimization criteria, architectural constraints such as the number of processors, and the width of the SIMD unit. The design process is performed in two levels- the block matrix level which constructs the H_b matrix, and the sub-matrix level which assigns the shift values to the sub-matrices in the H_b block matrix.

We use AA-LDPC code with the parity check matrix which has a D x B array of S x S sub-matrices. The (n,k) LDPC code has k information bits, n coded bits, and (n-k) parity bits with a code rate of r=k/n. The parity check matrix H is of dimension (n-k) x n, which defines a set of equations as follows:

$$H.v^T=0 \qquad (5)$$

Denote H = [$H_1 H_2$¿, where $H_1 \wedge H_2$ have dimensions (n-k) x k and (n-k)x(n-k), respectively.

Note that $H_1 \wedge H_2$ can also be represented as an m x b array of S x S sub matrices and m x m of S x S sub-matrices, respectively, where m = (n-k)/S and b = k/S. Denote the code word $v=[s\,p]$, where s is the k information bits and p is the n-k parity bits. Using the decomposed check matrix H and the decomposed code word v in the above check matrix equation, we have

$$H_1.s^T+H_2.p^T=0 \quad \text{(mod 2)} \qquad (6)$$

$$p^T=H_2^{-1}H_1.s^T \quad \text{(mod 2)} \qquad (7)$$

The Quasi Cyclic Architecture Aware (QC AA) LDPC code provides efficient encoding and decoding with excellent performance. For the parity check matrix in the QC AA-LDPC code, H_2 has a simple deterministic structure where the encoding can be performed recursively using the structure of the QC AA-LDPC code as shown in below:

The deterministic structure of H_2 enables the encoding procedure to be recursively done. The first column of the sub-matrix in h = [$h_0, h_1, \ldots, h_{m-1}$¿^{T} satisfies $\sum_{i=0}^{c-1} h_i$ = I_{sxs} (mod2). The other column in H_2 has two identity sub-matrices.

As discussed before, H_1 consists of an m x b array of S x S sub-matrices, which are either zero or shifted identity matrices. Given a block of information bits s, if we decompose the s information bits into a 1 x b array of 1 x S sub-matrices, and also decompose the p parity bits into a 1 x m array of 1 x S sub-matrices, we have the following recursive parity check vector equation using the decomposed (H, v) in equation (6) above.

$$p_0^T=\sum_{i=0}^{m-1} H_1^{(row\,i)}.s^T \quad \text{(mod2)}$$

$$p_1^T=H_1^{(row\,0)}.s^T+h_0.p_0^T \quad \text{(mod2)}$$

$$p_2^T=H_1^{(row\,1)}.s^T+h_1.p_0^T+p_1^T \quad \text{(mod2)}$$

……

$$p_{m-1}^{T} = H_{1}^{(row\,m-2)} . s^{T} + h_{m-2} . p_{0}^{T} + p_{m-2}^{T} \quad \text{(mod2)}$$

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the LDPC design.

3.2.3 Mapper

The Mapper module takes the stream of coded bits, divides them into groups of *m* bits, maps each group of *m* bits into one of M (= 2^{m}¿ signal constellations, and generates the stream of corresponding symbols. The Mapper is programmable to support different constellations and different modulations up to 4096QAM. The Mapper supports six standard Gray mapped constellations- QPSK, 16QAM, 64QAM, 256QAM, 1024QAM and 4096QAM.

The Gray mapping for mapping the coded binary bits into symbols can be done using the following recursive formula:

gray (k) = $2^{n-1} + ¿\left(2^{n} - 1 - k\right), \qquad 2^{n-1} \le k < 2^{n}$

with gray (0) = 0, gray (1) = 1, as initial values.

The Mapper module takes the stream of coded bits, divides them into groups of *m* bits, and maps each group of *m* bits into one of the M (= 2^{m}¿ signal constellations using the Gray code mapping rule as follows:

$$s_{k,m-1}\, s_{k,m-2}\, s_{k,m-3} \quad ..... \quad s_{k,0} \xrightarrow{\text{map}} \quad I_k \;,\; Q_k$$

In the above equation, $s_{k,i}$ (i = 0,1,2, …., m-1) represents the i^{th} bit in the k^{th} symbol to be coded by the Gray code mapping, and I_k and Q_k represents the in-phase and quadrature component of the k^{th} symbol, respectively.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the mapper design.

3.3 Modulator Functional Block Description

The modulator in the transmitter receives a stream of symbol data from the Mapper. The symbols are then modulated into real signals (I) and imaginary signals (Q). The digital format of the airframe is converted to analog signals by a digital-to-analog converter (DAC) and is then sent for transmission.

Figure 7 shows the functional block diagram of the modulator with adjacent functional blocks (which are in green boxes).

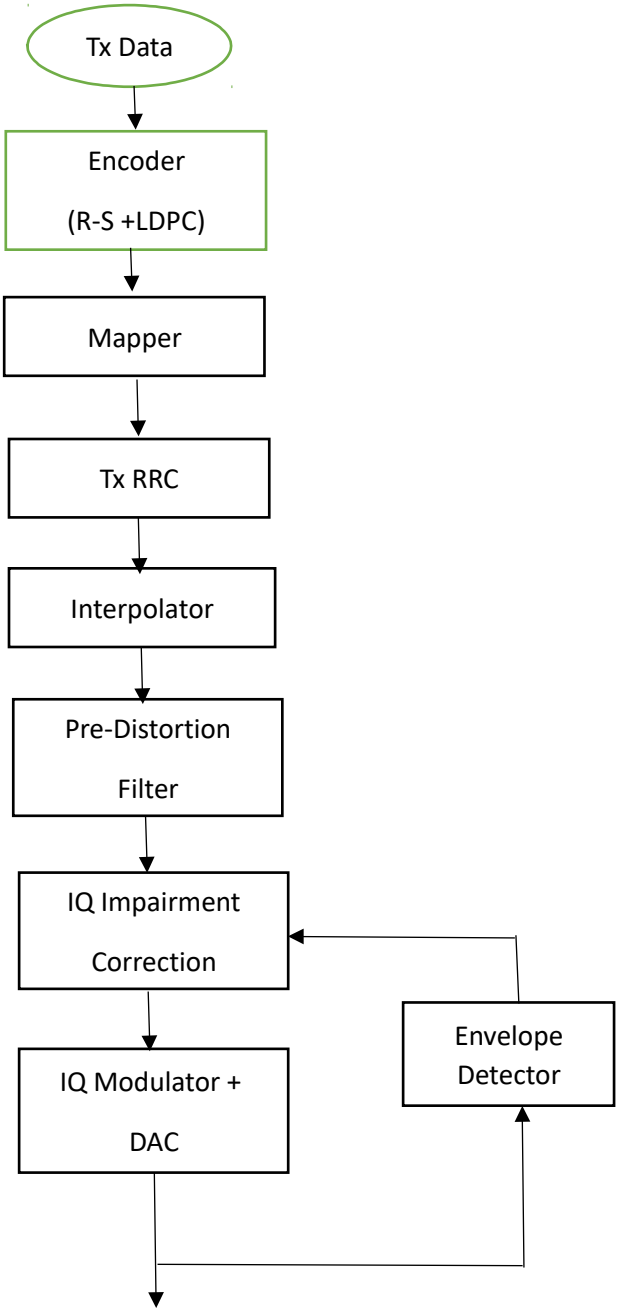


Figure 7: Modulator Functional Block Diagram

3.3.1. Tx RRC

The Transmitter's squared Root Raised Cosine (Tx RRC) filter is an interpolating 64-tap symmetric Finite Impulse Response (FIR) filter. It is used to reduce the inter-channel interference by shaping the modulated signal spectrum. Inputs to the filter are symbols provided at the baud rate. Outputs are sampled at twice the symbol rate.

3.3.2. Interpolator

The interpolator is a 12-tap Poly-Phase interpolator filter which provides the interpolation for a different time offset, from zero to half of a symbol interval (with 1/2048 symbol accuracy). The interpolator module adjusts the rate between the sampling clock of the DAC (digital-to-analog converter) and twice the symbol clock.

3.3.3. Pre-Distortion Filter

The pre-distortion filter is designed to compensate for the power amplifier's non-linearity. Successful pre-distortion reduces the required amplifier back-off, and thus increases the transmit power and amplifier efficiency. The module implements a 5^{th} order complex polynomial to compensate for 3^{rd} and 5^{th} order inter-modulation. The user has to update the coefficients of the polynomial according to the transmission power.

3.3.4. Tx IQ Impairment Correction

The IQ correction module corrects the IQ imbalance of the transmitter IQ modulator. The algorithm used in this module uses the feedback returning from the envelope detector. The algorithm is an adaptive algorithm which estimates the IQ gain, phase imbalances and DC offset, and then adaptively corrects them.

3.4 Demodulator Functional Block Description

The demodulator in the receiver receives a stream of symbol data from the analog-to-digital converter (ADC). The demodulator performs both frequency and timing (symbol and frame) synchronization processes while also performing the automatic gain control (AGC) process on the received symbols. After obtaining the synchronization, the demodulator performs an Equalization process to mitigate the

channel effect on the received symbols. Then the output of the Equalizer is sent to the De-mapper (or Slicer) and then sent to the LDPC decoder for data correction.

The Demodulator is consists of the following sub-blocks:

- Rx Impairment Correction
- Rough Frequency Synchronization
- Symbol Synchronization
- Frame Synchronization
- Equalization
- Carrier Synchronization

Figure 8 shows the functional block diagram of the demodulator with adjacent functional blocks (which are in green boxes).

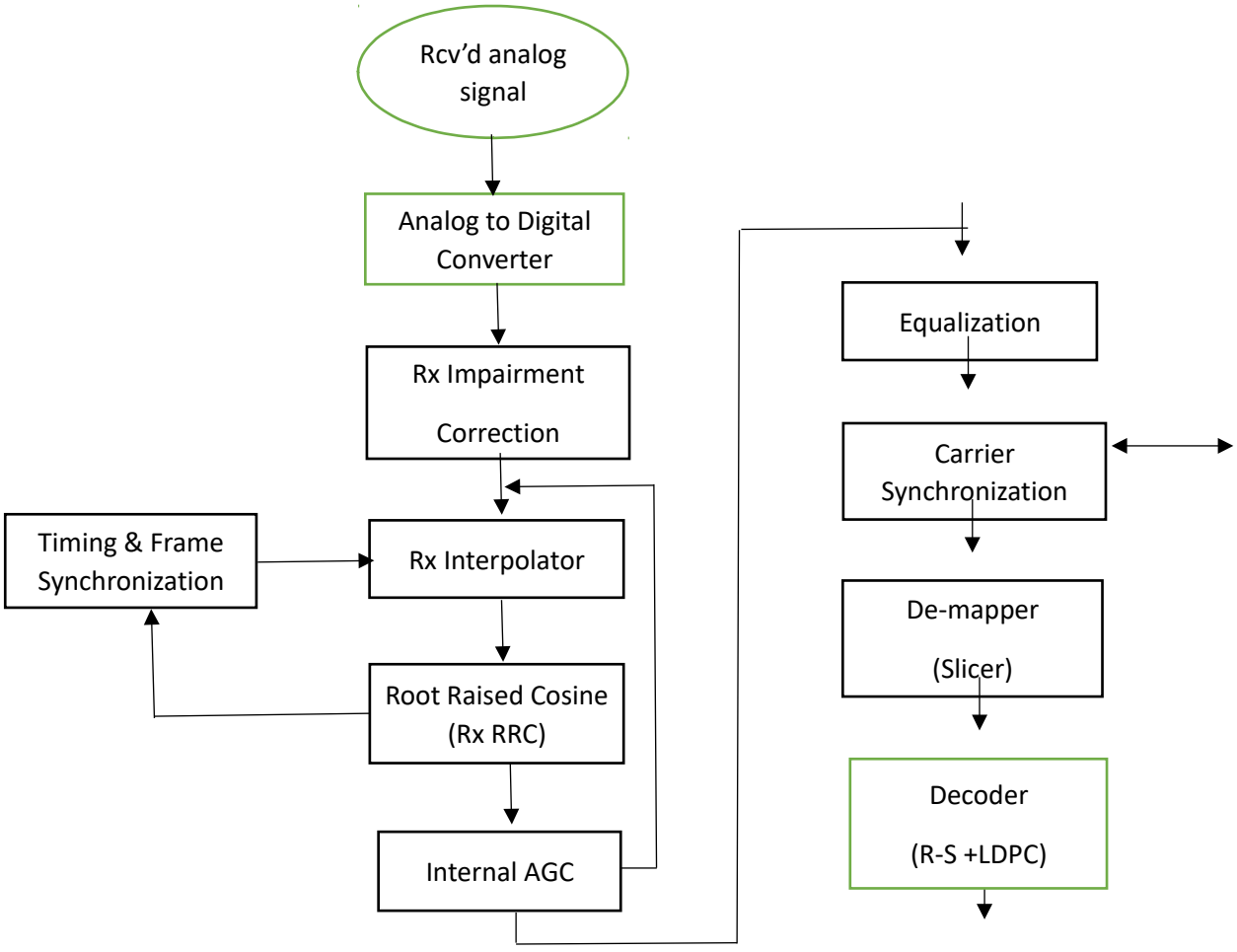


Figure 8: Demodulator Functional Block Diagram

3.4.1. Rx Impairment Correction

The Rx Impairment correction module corrects the gain, phase and DC offset IQ imbalances, caused by the receiver IQ demodulation. The algorithm is an adaptive algorithm which estimates the IQ gain imbalance, phase imbalance and DC offset, and adaptively corrects them.

3.4.2. Rx Interpolator

There are two functionalities to be performed in the Rx Interpolator. The first is the sampling rate reduction and the second is the timing synchronization between the ADC sampling time and the symbol timing. The Rx Interpolator reduces the data rate from the ADC sampling rate to twice the symbol rate. The Rx Interpolator is a 12-tap Poly-Phase interpolator filter which provides the interpolation for a different time offset, from zero to half of a symbol interval (with 1/2048 symbol accuracy). The interpolation filter is controlled by the timing loop from the Symbol Synchronization module.

3.4.2. Rx RRC

The Receiver squared Root Raised Cosine (Rx RRC) filter is an interpolating 96-tap symmetric Finite Impulse Response (FIR) filter. It is used to reduce the inter-channel interference by shaping the modulated signal spectrum. Inputs and outputs of the filter are sampled at twice the symbol rate.

3.4.3. Timing and Frame Synchronization

There are two synchronization function modules used at the output of the Rx RRC- The first is the Symbol Timing Synchronization module and the second is the Frame Timing Synchronization module. Note that we need to achieve the symbol timing synchronization first and then the frame timing synchronization for correct system operation.

3.4.3.1. Symbol Timing Synchronization

Symbol timing synchronization is performed to synchronize the symbol timing of the receiver with the transmitted symbol timing. The synchronizer uses the eye pattern of the received data, where there exists one half of a symbol period between the eye opening and the eye transition timing of the data.

3.4.3.2. Frame Synchronization

Frame Synchronization is performed to synchronize the frame timing of the receiver with the transmitted frame timing. This function block identifies the start of the airframe. The frame synchronization is performed after the symbol synchronization.

The frame synchronization module uses an Acquisition Frame whose frame format is known between the transmitter and receiver. The Acquisition Frame is 4096 symbol size frame and consists of eight sub-frames as can be seen in Figure 9. Each sub-frame consists of 256 preamble symbols and 256 QPSK symbols. The 256 preamble symbols in the sub-frame repeat a block of 128 preamble symbols twice. The 128 preamble symbols are obtained by generating a 256 Gold Sequence (GS) and then mapping the 256 GS into 128 QPSK constellation signals. The generator for the 256 Gold Sequence block consists of two 8-bit shift registers and several XOR gates between those two registers.

Sub-frames 1 through 7 contain identical symbols. The symbols in Sub-frame 8 however differ from the symbols of the preceding 7 Sub-frames. The 256 preamble symbols in Sub-frame 8 have their sign-bits inverted compared to the 256 preamble symbols for the Sub-frames 1 through 7 of the Acquisition Frame. The inverted sign is used to indicate that the sub-frame is the last sub-frame in the Acquisition Frame.

The Frame Synchronization module uses a preamble correlator to search for the block of 256 preamble symbols and locks onto it. Once the last sub-frame is detected, a state machine changes the frame timing counter to synchronize the receiver frame timing with the transmitted frame timing of the transmitter.

Figure 9 shows the Acquisition Frame configuration

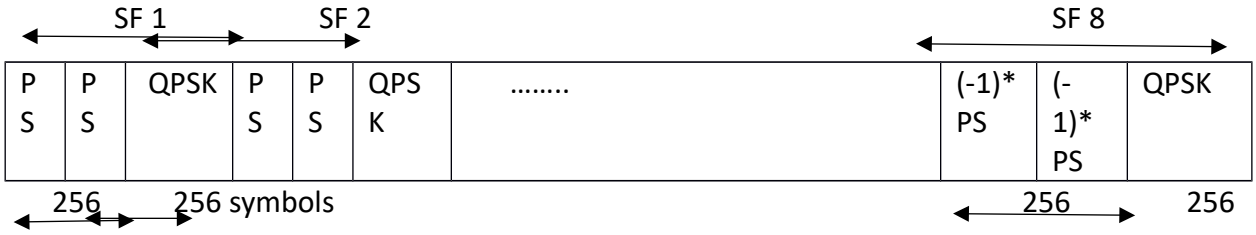


Figure 9: Acquisition Frame Format

3.4.5. Internal Automatic Gain Control (AGC)

The AGC is used to accommodate a wide dynamic range on the input signal and to provide a constant signal level for the communication system to ensure that the communication system operates reliably and effectively. Usually, the receiver is required to accommodate a dynamic range of over 100 dB for the signal in backhaul RF systems.

In general, the AGC needs to keep the output of the variable gain amplifier (VGA) constant by keeping the sum of the input power, low noise amplifier (LNA) gain, and the VGA input. In this approach, we need to ensure that the numerical control signal matches the physical signal level consistently.

In backhaul or point-to-point (PTP) systems, a fast AGC should not impair the function of the receiver's algorithms. Unless the AGC is capable of perfectly tracking the fast fading or 100 dB/sec fast block rate of the signal, we may encounter headroom problems in the AGC and the signal could be clipped by the A/D converter. The fast AGC will however impair the soft information collected for decoding. In this case, we may need to use weighting with the received signal strength information (RSSI) to update the AGC on a frame-by-frame basis.

The AGC power control loop consists of the antenna, the LNA, the VGA, and the AGC loop. In general, the output signal from the AGC is used to control the VGA input signal from the LNA to keep the VGA output signal constant. Figure 10 models the general AGC loop diagram representing the LNA and VGA as summation nodes.

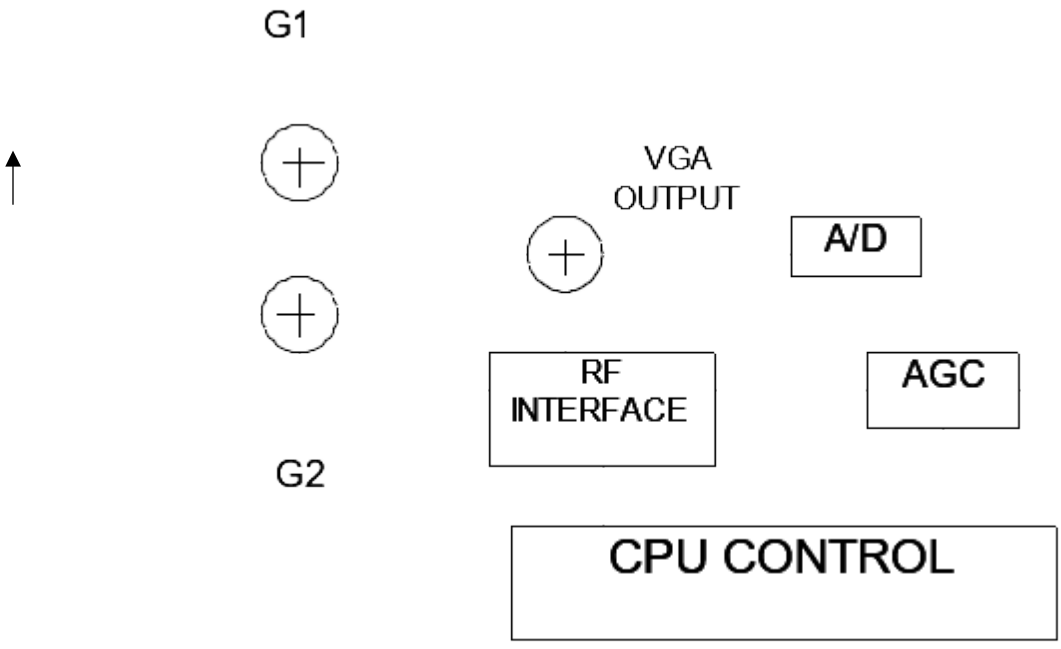


Figure 10. AGC Loop System Diagram

From the perspective of power level, we can summarize the signal flow as follows:

P_{IN} + G_{LNA} + G_{VGA} = VGA OUT = Constant.

Where, P_{IN} = Input signal power from antenna, G_{LNA} = LNA gain, G_{VGA} = VGA gain.

In other words, as the power of the input signal changes, the gain of the VGA should change accordingly. If the power of the input signal to the VGA is increasing, the gain of the VGA should be decreased to keep its output constant. We use the AGC loop to control the gain of the VGA, G_{VGA}.

The AGC control loop consists of the signal energy estimation block, the energy comparison block with the AGC reference (or target), and the RF interface block. The signal energy estimation block estimates the input signal energy, where the energy estimate is then used to compare with the AGC reference signal and the resulting difference is used to estimate the input signal power of P_{IN} (which is the received signal strength indicator (RSSI) or AGC loop output power estimate). The RF interface block transforms the output of the power estimation block into the VGA control input signal, VGA_{IN}. The VGA_{IN} is used to control the VGA gain, G_{VGA}, and to keep the output of the VGA constant.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the AGC parameter design.

3.4.6. Equalization

Equalization is used to mitigate the channel effect on the received signal for clear communication. The Equalizer estimates the communication channel distortion (such as amplitude distortion, phase distortion, fading, and channel interference, etc.) and mitigates the channel effect.

In micro-wave or millimeter wave communication systems which use high carrier frequency, the equalizer combined with the carrier recovery is one of the most powerful solutions to fight against phase noise through the latency reduction between the phase noise estimation and correction.

We use a fractional spaced decision feedback equalizer (DFE) to mitigate the channel effect and inter-symbol interference. The DFE consists of two sections in the equalizer. The first is the Feed Forward Filter (FFF) and the second is the Feedback Filter (FBF). The FFF consists of a 24-tap T/2 spaced finite impulse response (FIR) filter. The FBF consists of either a 1-tap T spaced FIR filter. In general, the number of taps used is determined by the modulation order (or bit rate) that is used.

The phase correction is applied at the input of the FFF and at the input of decision device (DD) in the DFE using the phase error estimate which is obtained from a digital phase locked loop (DPLL). The SNR estimate is obtained by applying a signal and noise energy measurement algorithm at the output of the DD in the DFE.

For the operation of the equalizer during the acquisition mode, the preamble symbols located in the acquisition frame are used as a training sequence for the equalizer. This precedes the data mode operation (Pease refer to acquisition frame format in section 3.4.3.2).

For the operation of the equalizer during the data mode, the preamble symbols and pilot symbols located in the data frame are used for channel estimation and mitigation (Pease refer to the data frame format in figure 4).

For updating the filter coefficients in both the FFF and FBF a least mean square (LMS) algorithm is used. The step size (μ¿ used for the filter coefficient update in the LMS algorithm is programmable.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the equalizer system design.

3.4.7. Carrier Synchronization

The carrier synchronization loop tracks the phase error at the input of the decision device (DD) in the decision feedback equalizer (DFE). The carrier synchronization loop is a 2nd order type II digital phase locked loop (DPLL) operating at the symbol rate. The phase error estimate is obtained from the phase difference between the input and output of the decision device and is decision directed. Figure 11 is the functional block diagram of the carrier recovery loop (CRL) combined with the part of the DFE that is designed for joint detection, estimation, and compensation of the phase noise. The CRL tracks the phase error at the input of the DD in the DFE.

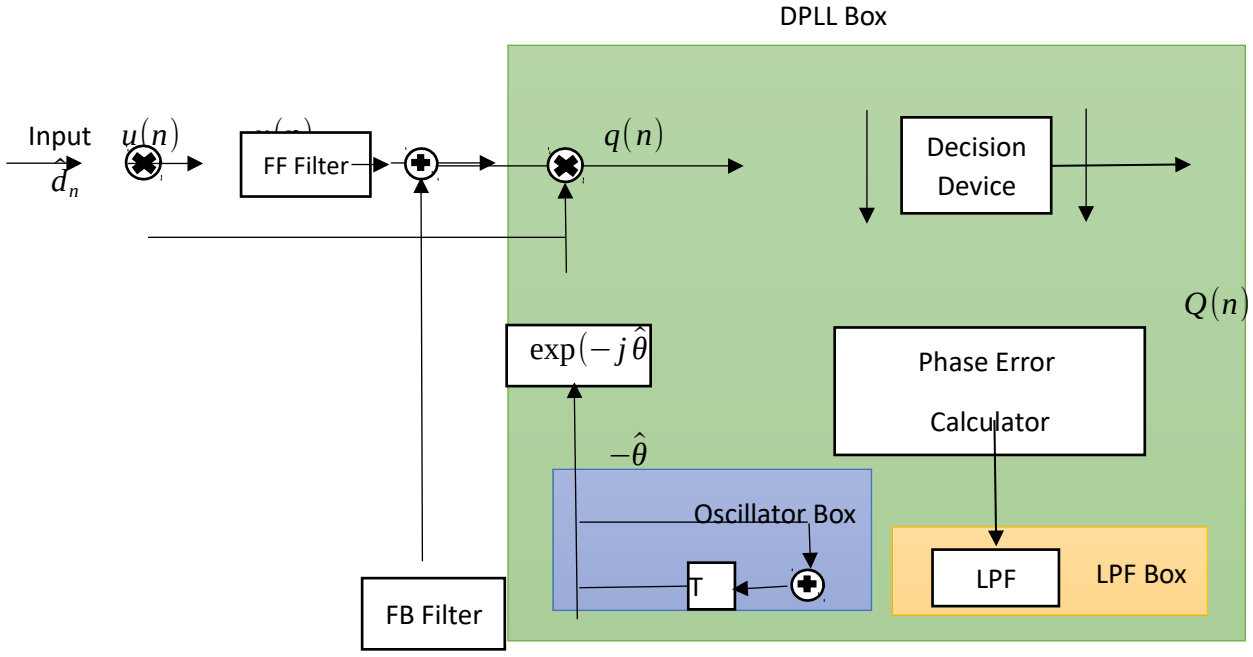


Figure11. Carrier Synchronization Loop block diagram

The low-pass-filter (LPF) in the Digital Phase Locked Loop (DPLL) in figure11 has two signal paths. One is the proportional path and the other is the integration path. The proportional path gain K_p and the integration path gain K_i in the LPF determine the bandwidth of the carrier synchronization loop filter.

The carrier synchronization loop supports a range of loop bandwidths designed to maximize the phase noise tracking capability and minimize the effect of additive white Gaussian noise (AWGN). The loop bandwidth requires optimization based on the expected signal to noise ratio (SNR) and the expected amount of the phase noise.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the carrier synchronization system design.

3.5 Decoder Block Description

The Rx decoder receives demodulated symbols from the demodulator, converts the received symbols into a bit stream by the de-mapping process, and then performs the error correction on the received bit streams of data. The Rx Decoder consists of a De-Mapper (or Slicer) and the LDPC Decoder.

3.5.1. De-Mapper Description

The Receiver receives the transmitted signal through an analog-to-digital converter (ADC) and demodulates the received symbols. The De-Mapper receives the demodulated symbols from the demodulator and converts the received symbols into the soft decision values of the corresponding bit stream using a soft decision decoding process. The De-Mappers output signals are the soft decision value of the bit stream of the corresponding symbols. As it is well known, an LDPC provides its best performance with the soft decision value as its input.

We can represent the demodulated receive signal, R_k ,, as follows:

$$R_k = X_k + jY_k = g_k\left(I_k + jQ_k\right) + \left(n_k^I + jn_k^Q\right)$$

Where, X_k and Y_k are the in-phase and quadrature components of the modulated signal, g_k is the transmission path gain of the signal, and $n_k^I \wedge n_k^Q$ are independently distributed zero mean white Gaussian noise.

The estimation of bit, $S_{k,i}$ (i = 0, 1, 2 … m), in symbol S_k is obtained by taking the log likelihood ratio (LLR) of the conditional probability of the bit being correct, as follows:

$$\Lambda(S_{k,i}) = \log \frac{Pr\{S_{k,i}=0 \mid X_k, Y_k\}}{Pr\{S_{k,i}=1 \mid X_k, Y_k\}}, \quad i = 0, 1, 2, \ldots, m-1$$

The above LLR equation can be approximated by a dual minimum metric approximation as follows:

[Equation 8]

$$\Lambda(S_{k,i}) = \log \frac{\sum_{z_i} \exp\{-1/\sigma_n^2 \cdot \{R_k - z_k(S_{k,i}=0)\}^2\}}{\sum_{z_i} \exp\{-1/\sigma_n^2 \cdot \{R_k - z_k(S_{k,i}=1)\}^2\}}$$

$$\approx \{\min[R_k - z_k(S_{k,i}=1)]^2 - \min[R_k - z_k(S_{k,i}=0)]^2\}$$

$$= (2n_{k,i} - 1) \cdot \{[R_k - z_k(S_{k,i} = n_{k,i})]^2 - \min[R_k - z_k(S_{k,i} = \acute{n}_{k,i})]^2\}$$

Where, $z_k(S_{k,i}=0)$ and $z_k(S_{k,i}=1)$ are the values of $I_k + jQ_k$ when $S_{k,i}=0$ and $S_{k,i}=1$, respectively. In the above equation $n_{k,i}$ represents the value of the i^{th} bit of the symbol, which is the closest to the received symbol, R_k. The $\acute{n}_{k,i}$ represents the negative value of $n_{k,i}$.

As can be seen from the above equation, the LLR can be calculated by finding the values of $z_k(S_{k,i}=1)$ and $z_k(S_{k,i}=0)$ which minimize the values of $[R_k - z_k(S_{k,i}=1)]^2$ and $[R_k - z_k(S_{k,i}=0)]^2$, respectively. The values of $z_k(S_{k,i}=1)$ and $z_k(S_{k,i}=0)$ which minimize the values of $[R_k - z_k(S_{k,i}=1)]^2$ and $[R_k - z_k(S_{k,i}=0)]^2$ can be determined by the range of the values of the in-phase and quadrature component of the received symbol, R_k.

The first term in the third line of the [Equation 8] can be represented using the in-phase and quadrature component of received symbol (or signal) as follows:

[Equation 9]

$$[R_k - z_k(S_{k,i} = n_{k,i})]^2 = (X_k - U_k)^2 + (Y_k - V_k)^2$$

Where, U_k and V_k are the in-phase and quadrature components of the received symbol on a signal constellation.

The second term in the third line of the [Equation 8] can be represented using the in-phase and quadrature component of received symbol (or signal) as follows:

[Equation 10]

$$\min [R_k - z_k (S_{k,i} = \acute{n}_{k,i}) ¿^2 = (X_k - U_{k,i} ¿^2 + (Y_k - V_{k,i} ¿^2$$

Where, $U_{k,i}$ and $V_{k,i}$ are the in-phase and quadrature components of the received symbol on the signal constellation which minimizes [$R_k - z_k (S_{k,i} = \acute{n}_{k,i}) ¿^2$.

Inserting [Equation 9] and [Equation 10] into [Equation 8], the LLR can be obtained as follows:

[Equation 11]

$$\Lambda(S_{k,i}) = (2 n_{k,i} -1) [\{ (X_k - U_k ¿^2 + (Y_k - V_k ¿^2 \} - \{ (X_k - U_{k,i} ¿^2 + (Y_k - V_{k,i} ¿^2 \}] .$$

$$= (2 n_{k,i} -1) [(U_k + U_{k,i} - 2 X_k) (U_k - U_{k,i}) + (V_k + V_{k,i} - 2 Y_k) (V_k - V_{k,i})]$$

We can use [Equation 11] to derive the soft decision values of the LLR of any M-ary QAM for the input into the decoder. As we support from 4QAM up to 4096QAM in our modem, we need to derive 6 sets of soft decision values for those LLRs derived for the 4QAM, 16QAM, 64QAM, 256QAM, 1024QAM and 4096QAM modulation rates.

As can be seen in [Equation 11], we need to find the values of U_k , $U_{k,i}$, V_k , and $V_{k,i}$ for the corresponding received symbol $S_{k,i}$ (= X_k + j Y_k) as a function of the bit position of i (= 0, 1, ….., *m-1*) to calculate the LLR of the received symbol Λ($S_{k,i}$).

Please refer to the white paper on the De-Mapper attached on the home page of the GSCom website for a more detailed description of the derivation of the soft decision values of the LLR for the various QAM levels.

3.5.2. LDPC Decoder

The classic LDPC decoding algorithm is a two-phase message passing (TPMP) algorithm or the so called belief propagation (BP) algorithm. The decoding procedure revolves around the two-phase transmission of extrinsic information between check nodes and bit nodes on the Tanner Graph. The message passing decoding algorithm utilizes the bit-to-bit dependence required in check-sum equations of the LDPC code to adjust the probability of each bit until obtaining a valid code-word.

The Turbo Decoding Message Passing (TDMP) algorithm outperforms the standard two-phase decoding algorithm with its faster convergence speed of roughly a factor of two, in terms of the number of decoding iterations required, and with its memory savings of more than 50%. With the TDMP scheme both variable and check messages collapse into a single type of message, and compared to the TPMP algorithm the TDMP algorithm requires lower complexity and is therefore more suitable for hardware implementation.

The Quasi Cyclic LDPC (QC-LDPC), used in our system, is an Architecture Aware LDPC (AA-LDPC). The ones in its each block row (or every s row) in the QC-LDPC are not overlapped. As a result, decoding can be processed for s rows simultaneously. The TDMP algorithm can be modified to a parallel version referred to as P-TDMP using the architecture of the AA-LDPC. The P-TDMP algorithm provides an improvement in decoding throughput over the ordinary TDMP algorithm and is attractive for high speed applications.

In the LDPC we used, the parity matrix H is decomposed into SxS binary sub-matrices. These sub-matrices satisfy the following two conditions- each column has at most a single 1, and each row has at most a single 1. Note that zero columns, zero rows and null matrices are allowed under this definition.

By decomposing the parity check matrix H of an LDPC code in such a way as to restrict the column positions of the ones, the LDPC decoding problem is transformed into a turbo-decoding problem where messages flow in tandem only between the adjacent super-codes as opposed to potentially all the sub-codes on the parity check matrix. The inter-leavers are factored into smaller inter-leavers that are more practical to implement.

3.5.2.1. LDPC Soft Decoding Algorithm

We review the LDPC decoding algorithm using a classical two phase message passing (TPMP) approach, called iterative layered belief propagation approach, using the following parity check matrix as an example:

$$\begin{bmatrix} 01011001 \\ 10010110 \\ 11001001 \\ 00100110 \end{bmatrix}$$

In the above parity check matrix, we have four check nodes (m = 4) and eight variable nodes (j=8). Let $L(q_{mj})$ denote the variable node log likelihood ratio (LLR) message sent from variable node j to the check node m, then:

$$L(q_{mj}) = L(q_j) - R_{mj} ,$$

¿
¿
$$\Psi\left(¿ L\left(q_{mj'}\right)\right)$$
$$\sum_{j' \epsilon N(m)\}} ¿$$

$$R_{mj} = \prod_{j' \epsilon N(m)\}} sign\left(L\left(q_{mj'}\right)\right) \Psi ¿$$

$$\begin{matrix} q_j \\ L¿ \end{matrix}) = L(q_{mj}) + R_{mj} ,$$

$$\Psi(x) = -\log\left(\tanh\left(\frac{x}{2}\right)\right)$$

Where, R_{mj} is the check node LLR message sent from the check node m to the variable node j, and $L(q_j)$ (j=1,2,…..,N) represents the a posteriori probability ratio (APP) for all variables. The APP messages are initialized with the channel reliability values of the coded bits. N(m) is the set of all variable nodes connected to the check node m.

The LDPC is an iterative decoding scheme which performs the above three parameter estimations and updates in each iteration step to improve the reliability of the bit estimation using LLR information and parity check equations. If all parity check equations are satisfied or the pre-determined maximum number of iteration is reached, then the decoding stops.

3.5.2.2. Turbo Decoding Message Passing (TDMP) Algorithm

The previous decoding algorithm operates using two types of messages and requires the saving all intermediate messages between both rounds at every iteration. Moreover, newly computed messages in a round of computations do not participate in further message computations until the decoding iteration is over. If updated messages are used directly within an iteration to compute new messages, then refined estimates will speed up the convergence behavior of the algorithm. Further, new check messages become directly variable messages within the iteration, hence, both variable and check messages collapse into a single type of message leading to a significant savings in the memory storage required.

The TDMP algorithm utilizes the most recently updated message directly within the iteration to compute a new message, which speeds up the convergence speed in terms of total iterations required. Furthermore, the two phase message computation in the TPMP algorithm is substituted by a single computation, and the memory savings is significant. As a result, the TDMP algorithm outperforms the TPMP algorithm in terms of throughput (20%-50%), coding gain, and hardware efficiency.

The TDMP algorithm can be described with the help of the parity check matrix in the above figure in Section 3.5.2.1 which has four parity check equations corresponding to a code of length 8. The algorithm is based on decoding the rows (parity check equations) of the parity check matrix sequentially. Extrinsic messages generated from decoding earlier rows are used as prior messages to decode subsequent rows. To each row i in H, we associate the vector $\lambda^i = \left[\lambda_1^i, \ldots\ldots, \lambda_{c_i}^i\right]$ of extrinsic messages corresponding to the nonzero entries in that row. The number of nonzero c_i in a row is called the weight. Let I_i denote the set of indexes of ones in row I and $\gamma(I_i)$ represent the posterior messages of row i. For example, in the parity check matrix H above in the figure shown in Section 3.5.2.1, the weight of row 2 is c_2 =4, while its index set I_2 ={1,4,6,7}, and its extrinsic message vector λ^2 =[$\lambda_1^2, \lambda_2^2, \lambda_3^2, \lambda_4^2$]. Where λ_1^2 corresponds to bit 1 and λ_2^2 corresponds to bit 4, etc. The extrinsic messages λ^i are associated to the extrinsic estimates about the bits from all parity check equations that these bits participate. Let γ = [$\gamma_1, \ldots\ldots, \gamma_N$] denote posteriori messages that stores the sum of all messages generated by the rows in which each bit participates. For example, N=8, γ_1 =

$\lambda_1^2+\lambda_1^3$, γ_2 = $\lambda_1^1+\lambda_2^3$,..., γ_8 = $\lambda_4^1+\lambda_4^3$. The posterior messages of row i are indexed as $\underline{\gamma}(I_i)$ and the hard decisions are determined by slicing the $\underline{\gamma}$ vector.

Decoding the *i* th parity check row involves the following four steps which constitute a decoding sub-iteration:

1) Read: $\underline{\lambda}^i$ and $\underline{\gamma}(I_i)$ are read for row i from memory.
2) Subtract: $\underline{\lambda}^i$ are subtracted from $\underline{\gamma}(I_i)$ to generate prior messages $\underline{\rho}$ = [$\rho_1,\ldots..,\rho_{c_i}$].
3) Decode: Decode row i using a SISO algorithm with $\underline{\rho}$ as input and $\underline{\Lambda}^i$ = [$\Lambda_1^i,\ldots,\Lambda_{c_i}^i$] as output.
4) Write back: Replace the original extrinsic message $\underline{\lambda}^i$ with $\underline{\Lambda}^i$ and update $\underline{\gamma}(I_i)$ by $\underline{\gamma}(I_i)$ = $\underline{\rho}$ + $\underline{\lambda}^i$.

A decoding iteration comprises multiple sub-iterations corresponding to the rows of H. A round of sub-iterations over all rows of H constitutes a decoding iteration. The whole decoding procedure will be terminated when it satisfies all the parity check equations in the matrix or the number of iterations has reached a predefined number. The TDMP algorithm requires storage memory for

$$M_{TDMP}=\sum_{i=1}^{M} c_i+N$$

messages, assuming H has M rows, N columns, and row weights c_i .

A formal description of the TDMP algorithm is given using the SISO algorithm which is a reduced-complexity message computation algorithm. The SISO algorithm will be explained in detail in the next section. The inputs to the TDMP algorithm are a sparse parity check matrix H_{MXN} representing a repeated accumulated (RA) code, LDPC code (where N=n+k), input channel observations $\underline{\delta}$ = [$\delta_{u_1},\ldots..,\delta_{u_k};\delta_{y_1},\ldots..,\delta_{y_n}$], and maximum number of decoding iterations T. The outputs are hard decision estimates of the information bits $\underline{u}=[u_1,u_2,\ldots..,u_k]$.

3.5.2.2.1. SISO algorithm for message decoding in TDMP algorithm

The SISO (soft input soft output) algorithm is adopted for message decoding in the TDMP algorithm as can be seen in the decoding step in its sub-iteration procedure. Unlike other schemes of

$$\sum_{¿} ¿, i=1,\ldots.,M\,;\,j=1,\ldots.,c_i \quad \begin{matrix} ¿ \\ l\epsilon l\} \Psi(\rho_l) \\ \Lambda_j^i=\Psi^{-1}¿ \end{matrix}$$

the SISO decoder need not use a lookup table, and can be implemented using simple logic gates. Moreover, the SISO circuit can be implemented using the “max-quartet” function $Q(x,y)$ as can be seen in Figure 11 resulting in memory savings, an improvement

in efficiency for the hardware implementation, and it provides a more accurate approximation than any other available at this time.

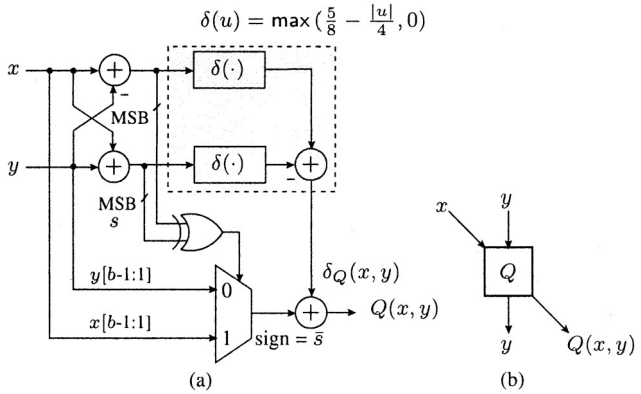


Figure 11. Max-quartet function $Q(x, y)$: (a) Logic circuit, and (b) Symbol

3.5.2.3. Parallel Turbo Decoding Message Passing (P-TDMP) Algorithm

The TDMP algorithm processes the rows of H sequentially which results in a lower processing throughput compared to the LDPC-like schedule. The TDMP algorithm can be parallelized by embedding some structure into the parity check matrix, H, that allows parallel processing of multiple rows without communicating messages between the nodes corresponding to these rows. The resulting H is composed of several bands of non-overlapping rows. We refer to the LDPC code with such a structure as an architecture-aware LDPC (AA-LDPC) code, and the parallel TDMP as a P-TDMP algorithm.

The P-TDMP algorithm is based on having the ones in every s number of rows of sub-matrices as non-overlapped, so the decoding procedure can be handled in parallel using s SISO decoders. The s SISO decoders constitute a decoder group working simultaneously. Decoder i processes row i in each sub-matrix and maintains the extrinsic messages $\lambda^{i,j}$ in local memory, while posteriori messages are passed to the decoders s messages at a time from a global memory. The decoding procedure runs over the rows of H in c iterations and processes s rows simultaneously for every iteration which improves the de-coding throughput by a factor of s as compared to the TDMP algorithm. The AA-TDMP reduces the required number of multiplexers/de-multiplexers by an order of n (O(n)), and the required number of control overhead by an order of O(log n).

The parallel Turbo-decoding message-passing (P-TDMP) algorithm is chosen for the AA-LDPC codes providing a faster convergence speed and hence a throughput advantage over the standard TDMP codes. We employ a reduced complexity message computation mechanism (SISO algorithm) and the features of a programmable network for message interleaving, based on the code structure.

To design finite length LDPC codes that can be efficiently implemented on the target multi-processor architecture for high speed communication, we utilize a design process that combines the optimal degree distribution for asymptotic performance, characteristics of structured sub-matrices, finite length code optimization criteria, architectural constraints such as the number of processors, and the width of the SIMD unit. The design process is performed at two levels. First is the block matrix level design which

constructs the H_b matrix, and second is the sub-matrix level design which assigns shift values to the sub-matrices.

The S rows of ones in each row of sub-matrices in an AA_LDPC H are non-overlapping, and hence can be processed in parallel using S SISO decoders. Decoder s processes row s in each row of sub-matrices, for a total of D rows, and maintains the extrinsic messages denoted by $\lambda^{d,s}, d=1,2,\ldots,D,$ in a local memory. Posterior messages are stored in a global memory of size N and passed in parallel (S messages at a time) to the decoders using a network that implements the factored edge permute process. The innermost parallel loop runs over the rows of H in D iterations processing S rows during each iteration period. This results in a factor of S improvement in decoding throughput over the TDMP algorithm.

3.5.2.3.1. Programmable P-TDMP Decoder Architecture

Figure 12 shows one architecture option for a high level programmable decoder implementing the P-TDMP algorithm. We assume that the corresponding parity check matrix is for an AA-LDPC code and has the following parameters:

1) S is the size of the sub-matrix partitions (assuming permutation matrices).

2) B = n/S denotes the number of sub-matrices per block row.

3) D = m/S denotes the number of sub-matrices per block column.

4) r is the maximum number of permutation matrices per block row.

5) $c_1, \ldots\ldots, c_B,$ corresponds to the number of permutation matrices in the block columns.

The architecture includes B memory modules for storing the messages, S MPUs (message processing units) that operate in parallel, and read/write networks for transporting the messages between the memory and the MPUs. The parameter S acts as a scaling parameter.

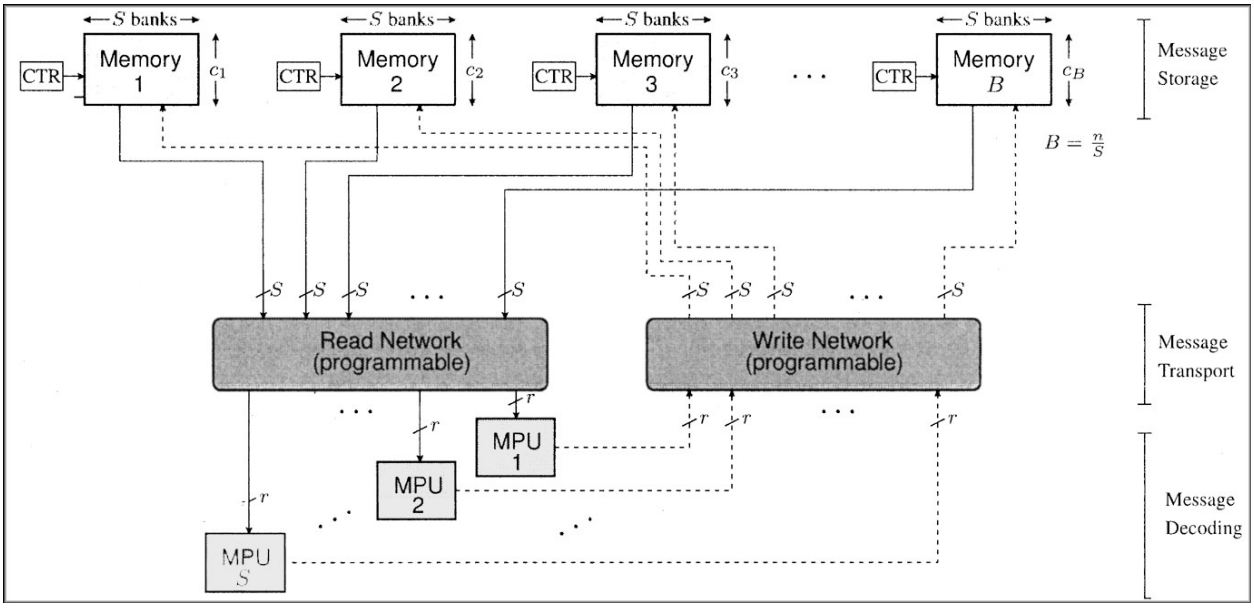


Figure 12. Decoder architecture implementing the P-TDMP algorithm

The decoder completes a single decoding iteration by performing a round of D updates across the super-codes. An update corresponding to a single super-code C^j constitutes a sub-iteration which involves the following four steps:

1) The network performs c read operations from memory, where c is the maximum node degree of C^j. It then forwards r messages to each of the S MPUs.
2) The MPUs update the messages in parallel using the SISO algorithm.

3) The updated messages are routed by the write network to the appropriate memory modules scheduled to receive message related to C^j .
4) The messages are written in the designated memory modules and the address counters are updated.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the LDPC system design.

3.5.3. Reed-Solomon Decoder

Reed-Solomon (R-S) codes are block, systematic, and cyclic code with a finite field of size of (n, k). R-S decoder recovers the transmitted signal, T(x), which was encoded by the R-S encoder from a received signal through a communication channel. The channel may introduce errors, E(x), into the transmitted signal in a communication environment. Thus the received signal, R(x), is given by;

R(x) = T(x) + E(x),

where T(x) is given in equation (4) and

$$E(x) = E_{n,-1}x^{n-1} + \ldots + E_1 x + E_0 \quad \text{---------- (12)}$$

and each of the coefficients $E_{n-1}, \ldots, E_0$ is an m-bit error symbol represented by an element of the GF(2^m) with the positions of the errors in the code word determined by the degree of x for that term. If more than t=(n-k)/2 of the E values are non-zero, then the capacity of the code is exceeded and the errors are not correctable.

The R-S decoder corrects the error signal if the number of error symbol is less than or equal to t=(n-k)/2 using the following four major steps: 1) calculating the syndromes; 2) finding the error location polynomial; 3) determination of the error location number; 4) calculation of the error values.

3.5.3.1 Syndrome Calculation

Section 2.1.2 showed that the encoded message polynomial or transmitted code word, T(x), is divisible with the generator polynomial without remainder and that this property extends to the individual factors of the generator polynomial.

The first step of decoding a code is to calculate the syndrome from the received signal R(x). The syndrome can be obtained by dividing the received signal polynomial, R(x), by each of the factors ($x + \alpha^i$) of the generator polynomial, g(x), in equation (2). This produces a quotient and a remainder as follows;

$$\frac{R(x)}{(x+\alpha^i)} = Q_i(x) + \frac{S_i}{(x+\alpha^i)} \; for \; 0 \le i \le 2t-1$$ ---------- (13)

The remainders, S_i , resulting from these division are known as the syndromes.

The syndromes S_i can be represented as follows by rearranging the equation (13);

$$S_i = Q_i \times (x+\alpha^i) + R(x)$$ ---------- (14)

The syndrome, S_i , is reduced to the following equation (15) when x = α^i in equation (14);

$$\begin{aligned} &\alpha^i ¿^{n-2} + \ldots + R_1 \alpha^i + R_0 \\ &\quad \alpha^i ¿^{n-1} + R_{n-2} ¿ \quad \text{, for i = 1, 2, ..., 2t} \\ &S_i = R(\alpha^i) = R_{n-1} ¿ \end{aligned}$$ ---------- (15)

where the coefficients $R_{n-1} \ldots . R_0$ are the symbols of the received code word. The physical meaning of the above equation (15) is that each of the syndrome values can also be obtained by substituting x= α^i in the received signal polynomial, as an alternative to the division of the R(x) by ($x - \alpha^i ¿$ to form the remainder.

Note also that R($\alpha^i ¿ = T(\alpha^i) + E(\alpha^i)$ and $T(\alpha^i) = 0$ because $x+\alpha^i$ is a factor of g(x), which is a factor of T(x). As a result

$$S_i = R(\alpha^i) = E(\alpha^i)$$, for i = 1, 2, ..., 2t ---------- (16)

This means that the syndrome values are only dependent on the error pattern and are not affected by data values. In addition, all the syndrome values are zeros when no errors have occurred.

3.5.3.2 Error Location Polynomial

Assume that E(x) is an error pattern of v , $where \; v \le t$, say;

$$E(x) = Y_1 x^{e_1} + Y_2 x^{e_2} + \ldots + Y_v x^{e_v}$$ ---------- (17)

where $e_1, \ldots, e_v$ are the locations of the errors in the code word as the corresponding powers of x, while $Y_1, \ldots, Y_v$ represent the error values at those locations. Substituting the equation (17) into the equation (16) produces;

$$S_i = E(\alpha^i) = Y_1 \alpha^{ie_1} + Y_2 \alpha^{ie_2} + \ldots + Y_v \alpha^{ie_v} = Y_1 x_1^i + Y_2 x_2^i + \ldots + Y_v x_v^i$$, for i=1,..,2t -------- (18)

Where $x_1 = \alpha^{e_1}, \ldots x_v = \alpha^{e_v}$ are known as error locators. Note that the syndromes are written as $S_1, \ldots . , S_{2t}$ to correspond with the roots of $\alpha^1, \ldots . \alpha^{2t}$ and the powers of x are dependent on the chosen roots in the generator polynomial of the equation (2).

GScom uses the error location polynomial to represent the number of errors and their locations in the received signal (or polynomial). The error location polynomial is constructed to have v factors of the

form (1+ x_j x) and therefore has the inverses $x_1^{-1},\ldots\ldots,x_v^{-1}$ of the v error locators as its roots;

$$\Lambda(x)=(1+x_1x)(1+x_2x)\ldots(1+x_vx)=1+\Lambda_1x+\ldots+\Lambda_{v-1}x^{v-1}+\Lambda_vx^v \quad \text{---------- (19)}$$

For each error, there is a corresponding root x_j^{-1} that makes $\Lambda(x)$ equal to zero

$$1+\Lambda_1x_j^{-1}+\ldots+\Lambda_{v-1}x_j^{-v+1}+\Lambda_vx_j^{-v}=0$$

Multiplying through by $Y_jx_j^{i+v}$;

$$Y_jx_j^{i+v}+\Lambda_1Y_jx_j^{i+v-1}+\ldots+\Lambda_{v-1}Y_jx_j^{i+1}+\Lambda_vY_jx_j^i=0, for\ i=1,2,\ldots,2t$$

GScom obtains similar equations for all error and by taking the summation of all the terms;

$$\sum_{j=1}^{v}Y_jx_j^{i+v}+\Lambda_1\sum_{j=1}^{v}Y_jx_j^{i+v-1}+\ldots+\Lambda_v\sum_{j=1}^{v}Y_jx_j^i=0$$

or

$$S_{i+v}+\Lambda_1S_{i+v-1}+\ldots+\Lambda_vS_i=0, for\ i=1,2,\ldots,2t \quad \text{---------- (20)}$$

The 2t sets of simultaneous linear equation have v unknown coefficients of the error location polynomial. The first v simultaneous equations can be used to find the coefficients of $\Lambda_{1,\ldots,}\Lambda_v$ of the error location polynomial (19). There are several methods to find the coefficients of the error location polynomial. The Berlekamp's algorithm and the Euclidean algorithm are the most efficient algorithms to solve the coefficients of the error location polynomial. The Berlekamp's algorithm is a concise and efficient algorithm to find the coefficients using micro-processor. GScom believes that the Euclidean algorithm is a more efficient algorithm to find the coefficients by implementing with hardware for high speed data and for low-power application.

GScom describes the Berlekamp's algorithm very briefly and the Euclidean algorithm in more detail. GScom will use the Euclidean algorithm to implement the error location polynomial and its related process to implement R-S decoder in a hardware based system for high speed and low power purposes.

3.5.3.2.1 The Coefficient of the Error Location Polynomial

3.5.3.2.1.1. Berlekamp's Algorithm

Berlekamp's algorithm is an efficient iterative technique for solving equation (20). This is done by forming an approximation to the error location polynomial, starting with $\Lambda(x)=1$. Then at each stage, an error value is formed by substituting the approximate coefficients into the equations corresponding to that value of v . The error is then used to refine a correction polynomial, which is then added to improve the approximated $\Lambda(x)$. The process ends when the approximate error location polynomial checks consistently with the remaining equations.

3.5.3.2.1.2 Euclidean Algorithm

The Euclidean algorithm is another efficient algorithm for obtaining the coefficient of the error location polynomial. This algorithm uses the relationship between the errors and the syndromes expressed in the form of an equation based on polynomials. The Euclidean algorithm is also referred to as the key equation and requires two new polynomials, the syndrome and error magnitude polynomials, which will be introduced in the following sub-sections.

Syndrome Polynomial

The syndrome polynomial is defined as

$$S(x)=S_{2t}x^{2t-1}+\ldots+S_2x+S_1$$

where the coefficients are the 2t syndrome values obtained from the received code word using equation (9).

Error Magnitude Polynomial

The error magnitude polynomial is defined as

$$\Omega(x)=\Omega_v x^{v-1}+\ldots+\Omega_2 x+\Omega_1$$

Key Equation

The key equation can be written as;

$$\Omega(x)=[S(x)\Lambda(x)]mod\,x^{2t}$$

Where S(x) is the syndrome polynomial and Λ(x) is the error location polynomial. So that

$$\Omega_1=S_1$$

$$\Omega_2=S_2+S_1\Lambda_1$$

.

$$\Omega_v=S_v+S_{v-1}\Lambda_1+\ldots+S_1\Lambda_v$$

Applying the Euclid's Method to the Key Equation

Euclid's method can find the highest common factor d of two elements a and b, such that:

ua + vb = d ---------- (21)

where u and v are coefficients produced by the algorithm.

The product of S(x) and Λ(x) have degree 2t+v-1. So the product can be expressed as:

S(x) X Λ(x) = $F(x)\,X\,x^{2t}+\Omega(x)$

in which the terms of x^{2t} and above are represented by the F(x) term and the remaining part is represented by $\Omega(x)$. This can be rearranged as:

$S(x)\,X\,\Lambda(x)+F(x)\,X\,x^{2t}=\Omega(x)$. ---------- (22)

so that the known terms S(x) and x^{2t} correspond to the a and b terms in equation (21). The algorithm consists of dividing the x^{2t} by S(x) to produce a remainder . S(x) then becomes the dividend and the remainder becomes the divider to provide a new remainder. This process is continued until the degree of the remainder becomes less than t. At this point, both the remainder $\Omega(x)$ and the multiplying factor Λ(x) are available as terms in the calculation.

Note that the equation (22) has the same form as of the equation (3).

Solving the Error Location Polynomial – The Chien Search

After obtaining the coefficients values, $\Lambda_1,\ldots,\Lambda_v$, of the error location polynomial, it is possible to find its roots. If the polynomial is written in the form;

$\Lambda(x)=x_1\left(x+x_1^{-1}\right)x_2\left(x+x_2^{-1}\right)\ldots$

then the function value will be zero if x = $x_1^{-1},x_2^{-2},\ldots,$ that is $x=\alpha^{-e_1},\alpha^{-e_2},\ldots.$.

The roots, and hence the values of $x_1,\ldots,x_v$, are found by trial and error, known as the Chien search, in which all the possible values of the roots (the field values $\alpha^i, 0\le i\le n-1$) are substituted into equation (19) and the results are evaluated. If the expression reduces to zero, then that values of x is a root and identifies the error position.

The Forney's Algorithm for Calculating the Error Values

This is a means of calculating the error value Y_j having established the error location polynomial $\Lambda(x)$ and the error magnitude polynomial, $\Omega(x)$. The algorithm makes use of the derivative of the error location polynomial.

For a polynomial f(x) given by:

$f(x)=1+f_1x+f_2x^2+\ldots+f_vx^v$

the derivative is given by:

$$f'(x) = f_1 + 2f_2 x + \ldots + v f_v x^{v-1}$$

However, for the error location polynomial, $\Lambda(x)$, for $x = x_j^{-1}$, the derivative reduces to:

$$\Lambda'\left(x_j^{-1}\right) = \Lambda_1 + \Lambda_3 x_j^{-2} + \Lambda_5 x_j^{-4} + \ldots$$

which amounts to setting even-powered terms of the error location polynomial to zero and dividing through by $x = x_j^{-1}$.

Forney's algorithm is very efficient for the calculation of error values $Y_1 \ldots . Y_v$ and is very easily implemented into hardware effectively. According to the Forney's algorithm, the error value is given by:

$$Y_j = x_j^1 \frac{\Omega\left(x_j^{-1}\right)}{\Lambda'\left(x_j^{-1}\right)}$$

where $\Lambda'(x_j^{-1})$ is the derivative of $\Lambda(x)$ for $x = x_j^{-1}$.

Having located the symbols containing errors, identified by x_j through the usage of the Chien Search, and calculated the values of Y_j of those errors, the errors can be corrected by adding the error polynomial E(x) to the received signal polynomial R(x). It should be noted that conventionally the highest term of the received polynomial corresponds to the first symbol of the received code word.

3.5.3.3 Implementation of a Reed-Solomon Decoder

There are several approaches to implement the Reed-Solomon(R-S) decoder. For the high speed backhaul communication applications, GScom decided to implement the hardware based R-S decoder for the purpose of high speed and low power.

The following figure 13 is the arrangement of main units of the R-S decoder.

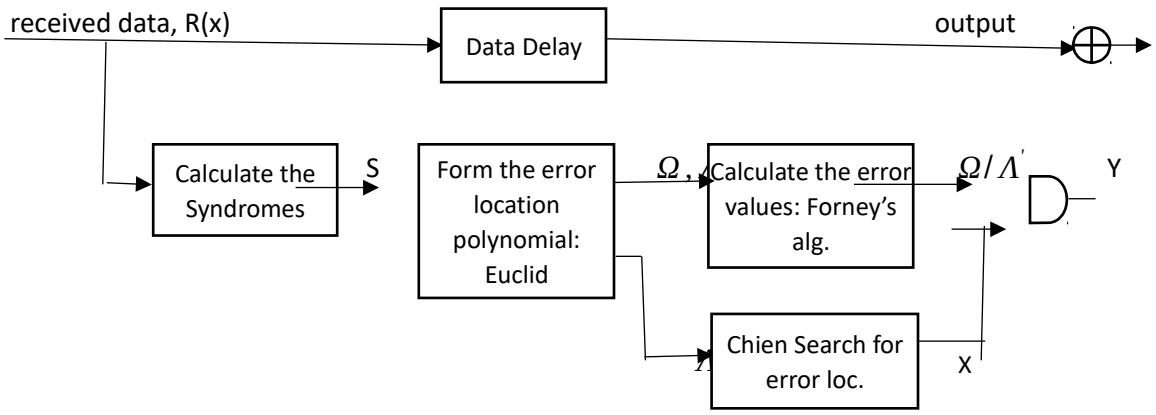


Figure 13. Main Processes of a Reed-Solomon Decoder

The first process in the R-S decoder is to calculate the syndrome values from the incoming code word (received data). These are then used to find the coefficients of the error location polynomial $\Lambda_1,\ldots,\Lambda_v$ and the error value (or magnitude) polynomial $\Omega_1,\ldots.,\Omega_v$ using the Euclidean algorithm. The error locations (i.e. positions) are identified by the Chien search and the error values are calculated using Forney's algorithm. As these calculations involve all the symbols of the received code word, it is necessary to store the message (or the received data) until the results of the calculation are available. Then, to correct the errors, each error values is added (modulo 2) to the symbol at the appropriate location in the received data (or code word).

GScom designed all the function blocks of the R-S decoder in a hardware based system for high speed and low power purposes.

Please refer to the white paper attached on the home page of the GSCom website for a more detailed description of the Reed-Solomon Codec system design.

4. Performance Test Result

GScom carried out a performance test of the 4096-QAM modem using computer simulation techniques using two types of radio channels in additive Gaussian noise and phase noise environments; an AWGN channel and a fading channel. The AWGN channel was used to test the performance of GScom's modem in an AWGN environment and the fading channel was used to test those of the modem in a fading environment.

GScom carried out a system performance simulation of the broadband wireless modem on 4QAM, 16QAM, 64QAM, 256QAM, 1024QAM, and 4096QAM modulation levels using various SNR levels of additive white Gaussian noise, along with phase noise masking signal. The simulation environment was as follows

- The Performance simulation of the 4096-QAM modem includes all the system operation modes and all the function blocks in the transmitter and receiver, except the functions of pre-distortion filter and IQ Impairment in digital domain
- Simulation is based on floating point simulation using MATLAB
- GScom used the phase noise mask obtained from a tier-1 backhaul equipment vendor in the simulation
- GScom used a Rummler channel model to simulate a fading channel. The Rummler channel includes those parameters of notch placement frequench, notch depth, minimum phase, non-mnimum phase, and relative time delay tau
- GScom used a pulse shaping filter with roll-off factor of 0.12 in the performance simulation test for this report
- GScom used R-S code with RS(255,235,21) format in the performance simulation test for this report
- GScom used an LDPC (CR of 0.5) in the performance simulation test for this document

GScom carried out a system performance simulation to test the performance of the 4096QAM modem in an AWGN environment and in a fading environment. Gscom uses LDPC codec for low order QAM modulated signals (from QPSK up to 256-QAM signals) and uses a concatenated code of the LDPC and R-S codecs for high oreder QAM modulated (1024-QAM and 4096-QAM) signals. GScom illustrated the performace test result in an AWGN environment in Section 4.1 and in a fading environment in Section 4.2.

4.1 Performance Simulation Results in AWGN Environment

GScom carried out a system simulation to test the performance of the 4096QAM modem. GScom illustrates the performace test results using the LDPC and R-S codecs developed by GScom. GScom generated a set of performance tables for each QAM modulation format which shows the performance simulation test results in each test environment. The performance table shows the symbol error rate (SER), un-coded bit error rate (UC BER), and coded bit error rates (CBER) at the output of LDPC (LD BER) and R-S (RS BER) codes as a function of SNR for the corresponding QAM modulated signal. Performance parameters such as the receiver sensitivity, coding gain, performance improvement gain due to mitigation technologies, etc. can be estimated by exploiting the performance table. The summary of the

performance parameters in each test environment will be in the following sub-sections. The coding gain is obtained from the performance improvement of the modem due to the usage of the R-S codec and the LDPC codec designed by GScom.

4.1.1 Summary of Performance Simulation Results in AWGN Environment

The performance parameters such as receiver sensitivity, coding gain, performance improvement gain due to mitigation technologies, etc. can be estimated by exploiting the performance tables in Section A.4.1.1 in the Appendix. The coding gain from the usage of the GScom's LDPC (CR of 0.5) and R-S (RS format of (255,235,21)) codecs can be evaluated by by exploiting the performance simulation tables in Section A.4.1.1 in the Appendix. The performance improvement of the GScom's carrier recovery system can be obtained by exploiting the performance simulation tables in Section A.4.1.2 in the Appendix.

The following table shows the receiver sensitivity of the 4096QAM modem in each QAM modulation format in AWGN environment.

Receiver Sensitivity in AWGN environment (using LDPC and R-S codes)

Modulation Format	4QAM	16QAM	64QAM	256QAM	1024-QAM		4096-QAM	
					w/o R-S	w/ R-S	w/o R-S	w/ R-S
RCVR Sensitivity	6 dB	13 dB	20 dB	25 dB	30 dB	28 dB	36 dB	32 dB

The following table shows the performance improvement of the modem due to the usage of the LDPC codec and the phase noise mitigation.

LDPC coding gain in reference to 1e-6 BER

	4QAM	16QAM	64QAM	256QAM	1024QAM	4096QAM
AWGN	7 dB	7 dB	6 dB	8 dB	8 dB	9 dB
PN+AWGN w/o CR	6 dB	7 dB	6 dB	5 dB	0 dB	0 dB
PN+AWGN w/CR	7 dB	7 dB	6 dB	7 dB	5 dB	5 dB

NOTE: BER=bit error rate, AWGN=additive white Gaussian noise, PN= phase noise

CR= carrier recovery using digital phase locked loop for phase noise mitigation

The GScom's 4096QAM modem provided an excellent performance in the AWGN and phase noise environment.

Note that the phase noise mask used in the above performance simulation is in the following table.

Phase noise mask used in the simulation

Frequency	100 Hz	2 KHz	10 KHz	100 KHz	100 MHz
Attenuation (dBc)	-37	-37	-58	-88	-148

The coding gain obtained from the usage of the RS coder is 2-4 dB depending upon the communication environment. The following table shows the additional coding gain obtained from the usage of the R-S coder on top of the LDPC coding gain.

R-S coding gain in reference to 1e-6 BER (using RS(255,235,21) format)

Communication Environment	1024-QAM	4096-QAM
AWGN	2 dB	4 dB

4.2 Performance Simulation Results in Fading Environment

The performance degradation of the digital radio systems is caused by the combination of the following performance degradation factors: interference, thermal noise, phase noise, and waveform distortion due to the multipath propagation in the fading environment.

Waveform distortion due to the multipath propagation affects the performance of the digital radio ystems as a function of the following parameters: transmission bandwidth (BW), carrier frequency, and modulation format. It is well known that the frequency selective fading is caused by the multipath propagation of the signal.

There are two countermeasures to the propogation distortion: diversity techniques and adaptive channel equalizers, which attempts to combat the attenuation and the distortion caused by the transmission medium. Equalization is effective against the waveform distortion caused by fading.

GScom designed a partial response digital feedbak equalizer which is effective in the fading environment. GScom carried out a performane simulation test in a fading environment using the Rummler channel model, Normalized System Parameter, and Signature Curve.

4.2.1 Fading Channel

There are various methods to evaluate the performance of digital radio systems. One of the performance evaluation methods to assess the waveform distion is to use the multipath fading channel in the performance test. Rummler channel model is one of the most popular channel models used in P-P fixed point & line-of-sight communication system design. Rummler model is effectively a two-path model.

The transfer function of the Rummler channel model can be written as follows.

$$H(f)=\alpha-\beta e^{-j2\pi(f-f_0)\tau} = a(1-be^{-j2\pi(f-f_0)\tau}) \quad (23)$$

Where,

f = frequency,

f_0 = notch frequency,

α = attenuation parameter for direct signal path,

β = attenuation parameter for reflected signal path,

τ = time delay between the direct and reflected signal paths.

The transfer function is a minimum phase when $\alpha>\beta$ (or positive τ) and is a non-minimum phase when $\alpha\le\beta$ (or negative τ). There is a flat fade when $¿H(f)¿^2=1$. The notch depth (ND) is relative to the flat fade and is related with the attenuation parameters in the transfer function as follows.

$$ND=10\log\left[¿\frac{1}{¿H(f_0)¿^2}¿\right]$$

The performance of digital radio system in fading environment is a function of the notch depth and notch freqeucy of the fading signal.

4.2.1.1 Performance Parameter used

There are various methods for calculating the outage time or the performance degration of digital systems. The performance degradation due to the waveform distortion can be estimated using a signature approach. GScom used the signature concept and normalized system parameter (NSP), K_n , to evaluate the performance of GScom's modem in a fading environment.

Signature can be used to measure the outage and to compare the relative sensitivity of different digital radio systems to the effects of frequency selective fading. Signature can be measured by approximating the actual fading by a two-ray simulator. The simplified three-ray model has the transfer function of Eq(23) in the previouse Section 4.2.1.

$$H(f)=\alpha-\beta e^{-j2\pi(f-f_0)\tau} = a(1-be^{-j2\pi(f-f_0)\tau}) \quad \text{---------- (23)}$$

Where a unity amplitude of direct ray and a second ray of amplitude of b, delayed by τ is assumed and 'a' is a scaling factor. The notch point (or notch frequency) is f_0 away from the center frequency and the notch depth,B, is B= -20 log (1-b). The signature is the plot of the critical value of B_c , as a function of f_0 at the outage ratio or BER of 10^{-6} point.

Reference is made to the signature concept, measurement, the representative parameters of signal bandwidth (W), notch depth (B_c) and normalized system parameter (K_n) defined in Recommendation ITU-R F.1093.

Equipment for nominal carrier space, CS $\geq 14\ MHz$, should have a signature within one of the limits provided below;

- For a reference delay of 6.3 ns and a BER of 10^{-6} the signature shall exhibit a normalized system paramter (NSP), K_n , calculated with the actually declared system parameters (W and B_c) and symbol rate of the system under test, equal to or less than K_n limits defined in the Table 1 below as appropriate.
- The limits are intended as a mean value of K_n seperately calculated for minimim phase and non-minimum phase cases.
- For mixed-mode systems the limits apply only for the reference modes.

The relationships between W, B_c and K_n defined in ETSI specification are summarized below.

$$K_n = \left(T^2 X W X \lambda_a\right)/\tau_r \quad \text{---------- (24)}$$

Where:
T: system baud period (ns) which is 1/(symbol rate)
W: signature width (GHz)
λ_a : average of linear signature depth (λ_c) variable with frequency (f) as:

$$\lambda_a = \frac{\int_{f=\frac{-W}{2}}^{f=\frac{W}{2}} \lambda_c(f)\,df}{W}$$

Where:
$\lambda_c(f) = 1 - b_c(f) = H_c(f)$,
$b_c = 1 - 10^{-B_c/20}$,
B_c : signature depth at outage rate expressed in dB.
τ_r : reference delay for λ_a (ns).

Table 1 gives the limits for generic-mode systems.

Table 1: K_n limits for the reference modes of mixed-mode equipment

Spectral Efficiency (reference index)	**Maximum** K_n (Reference to ITU-R F.1093)
2 (4QAM)	0.3
3	0.45
4 (16QAM)	0.6
5	0.75
6 (64QAM)	0.9
7	1.05
8 (256QAM)	1.2
9	Supplier declaration
10 (1024QAM)	

11	
12 (4096QAM)	

The notch frequency and notch depth in the fading signal affects the performance (or the sensitivity) of digital radio systems in the frequency selective fading environment. As seen in the equation (24) above, K_n is obtained by exploiting both of the notch frequency and notch depth.

4.2.2 Performance Simulation in Fading Environment

GScom used two test environments to evaluate the performance of the GScom's 4096QAM modem in a fading environment: flat fading environment and frequency selective fading environment. GScom measured the receiver sensitivity of the modem in both of the flat fading and frequency selective fading environments to get the objective performance evaluation of the 4096QAM modem. GScom used the receiver sensitivity, signature concept, and normaliezed system parameter in the frequency selective fading environment to get a performance evaluation of the 4096QAM modem.

4.2.2.1 Summary of the Performance Simulation Results in Flat Fading Environment

In the flat fading environment, the transfer function of the communication channel is constant, i.e. $¿H(f)¿^2=1$ in the equation (23).

GScom implemented a practical flat fading channel by setting the relative delay of the multi-path to be very small. Gscom obtained less than 0.1 dB notch depth variation in the passband of the signal by using a very small delay in the equation (23).

GScom used LDPC as the forward error correction code for lower order QAM modulated signals (from 4QAM up to 256QAM) and used a concatenated forward error correction code (R-S coder+LDPC) for high order QAM modulated signals (1024QAM and 4096QAM signals).

GScom used a LDPC (CR of 0.5) and a RS(255,235,21) format code in the simulation test. The set of the performance simulation test result, which supports the following receiver sensitivity table and the coding gain table, are in Section A.4.2.2.1 in the Appendix.

The following tables show the receiver sensitivity and coding gain of the 4096QAM modem in each of the QAM modulation format in a flat fading environment.

Receiver Sensitivity in flat fading environment (using LDPC and R-S codes)

Modulation Format	4QAM	16QAM	64QAM	256QAM	1024-QAM		4096-QAM	
					w/o R-S	w/ R-S	w/o R-S	w/ R-S
RCVR Sensitivity	6 dB	13 dB	20 dB	25 dB	31 dB	28 dB	36 dB	32 dB

Coding gain of the LDPC & R-S codes in reference to 1e-6 BER

	4QAM	16QAM	64QAM	256QAM	1024QAM	4096QA

						M
LDPC code	7 dB	7 dB	7 dB	7 dB	6 dB	7 dB
R-S code	NA	NA	NA	NA	3 dB	4 dB

As seen in the receiver sensitivities in Section 4.1.1 and those in this section, GScom's 4096QAM modem has almost the same receiver sensitivity in the flat fading environment as those in the AWGN environment. GScom's 4096QAM modem demonstrated excellent performance in a flat fading environment.

4.2.3 Performance Simulation in Frequency Selective Fading Environment

GScom carried out the performance test of the 4096QAM modem in a frequency selective fading environment under two test criteria: compliance test with ETSI specification (Table 1 in Section 4.2.1.1) and performance test in an aggressive fading environment from the perspective of notch frequency.

4.2.3.1 Summary of the Compliance Test with ETSI Specification

GScom carried out a performance simulation test based on the guide line on the usage of " K_n limits" specified in the specifications (Table 1 in Section 4.2.1.1). GScom used the "normalized system parameters (NSP)" specified in the ETSI specfication in the compliance test. GScom generated performance tables to illustrate the performance simulation test results. The performance tables are in the corresponding sections of Section A.4.2.3.1 in the Appendix.

The performance parameter, such as the receiver sensitivity, can be estimated by exploiting the performance tables in Section A.4.2.3.1 in the Appendix. The following table shows the receiver sensitivity of the 4096QAM modem obtained from the performance test using the system parameter specified in the specification (Table 1 in Section 4.2.1.1).

Receiver Sensitivity of 4096QAM modem in compliance test with specification (ETSI EN 302 217-2-1)

Modulation Format	4QAM	16QAM	64QAM	256QAM
RCVR Sensitivity	6 dB	13 dB	20 dB	26 dB

GScom's 4096QAM modem provided almost the same receiver sensitivity in the frequency selective fading environment as those obtained in the AWGN environment. The GScom's 4096QAM modem provided excellent performance in the frequeny selective fading environment which is specified in the specification.

4.2.3.2 Summary of Performance Test in Aggressive Fading Environment

The performance of digital radio systems in a fading environment is dependent upon the notch frequency and notch depth of the received fading signal. GScom carried out a perfromance test of the 4096QAM modem to check its performance in an aggressive fading environment.

GScom measured the receiver sensitivity and coding gain of the modem in a fading environment. GScom also measured the normalized system parameter using the signature concept in the frequency selective fading environment to get the objective performance evaluation of the 4096QAM modem. GScom generated performance tables to illustrate the performance simulation test results. The set of the performance table can be found in the corresponding Section A.4.2.3.2 in Appendix.

The following tables show the receiver sensitivity and the coding gain of the 4096QAM modem in each QAM modulation format in an aggressive frequency selective fading environment.

Receiver Sensitivity in aggressive frequency selective fading environment (using LDPC and R-S codes)

Modulation Format	4QAM	16QAM	64QAM	256QAM	1024-QAM		4096-QAM	
					w/o R-S	w/ R-S	w/o R-S	w/ R-S
RCVR Sensitivity	7.5 dB	13 dB	20 dB	26 dB	31 dB	29 dB	37 dB	33 dB

Coding gain of the LDPC & R-S codes in reference to 1e-6 BER

	4QAM	16QAM	64QAM	256QAM	1024QAM	4096QAM
LDPC code	7 dB	8 dB	8 dB	7 dB	7 dB	7 dB
R-S code	NA	NA	NA	NA	2 dB	3 dB

The performance degradation due to the multi-path fading environment, from the perspective of an aggressive notch frequency, can be evaluated by comparing the receiver sensitivity table in AWGN in Section 4.1.1 and those above in this section

As seen in the receiver sensitivities in Section 4.1.1 and those in this section, the performance degradation of the GScom's 4096QAM modem in an agressive frequency selective fading environment is at most 1 dB when compared to those in the AWGN environment. GScom's 4096QAM modem provided excellent performance in an aggressive frequency selective fading environment.

4.3 Performance Test Summary

GScom's broadband wireless 4096QAM modem provided very reliable functionality and excellent performance while being very robust to any kind of noise (such as AWGN, phase noise, etc.) and to any kind of communication channel.

The following is a summary of the strong points of GScom's broadband wireless 4096QAM modem verified during the performance testing of the modem:

- Excellent coding gain from the Reed-Solomon codec
- Excellent coding gain from the LDPC codec
- Two LDPC iterations are typically sufficient in most of the practical communication environments
- Excellent performance of the phase noise mitigation from the DFE combined with the Carrier Recovery in terms of the signal quality recovery and the latency
- Excellent and the best receiver sensitivity in AWGN environment
- Excellent receiver sensitivity in any kind of fading environment
- Very reliable Structured System Operation Architecture