

RESEARCH

Open Access

Design and implementation of AD9361-based software radio receiver



Feng Tian*, Hanqing Li and Liangchen Yuan

Abstract

Existing software radio platforms constructed by discrete devices have many disadvantages, such as high power consumption, high cost, and poor portability. In this study, an AD9361-based software radio communication system was designed on the basis of the zero-IF bandpass sampling software radio structure to solve the poor universality and expansibility problem of traditional software radio receivers. In the AD9361-based software radio communication system, the influences of channels on received signals and the inter-symbol interference caused by the multipath configuration are offset and eliminated by the minimum mean-squared error (MMSE) equalization algorithm. The simulation analyses on the core functions, including group detection, frame synchronization, channel estimation, and frequency-domain equilibrium, of the designed receiver was performed by ModelSim. The receiving functions of the software radio were realized by the core radio frequency (RF) board of AD9361 and the digital baseband development board of ZC706. The signal frequency spectra received and sent by the designed receiver overlap on the basis of the joint debugging and testing of the RF and digital baseband modules. Test results demonstrate that the designed software radio receiver has a reasonable structural design and can meet the design requirements in terms of overall performance. Additionally, the repeated development process of traditional software radio receivers is simplified, and the integration level and expansibility of the system can be improved. The results can provide valuable references for the development of universal software radio receivers.

Keywords: Software radio, Receiver, AD9361, Frequency-domain equalization, Channel estimation

1 Introduction

Software radio has overcome the disadvantage that previous communication platforms with different communication functions and frequency bands cannot communicate mutually. Existing software radio platforms are constructed by discrete devices, which are constrained by high power consumption and high cost. This situation not only requires technicians to be experienced in hardware design and radio frequency (RF) signal processing, but it also poses high access requirements for software. Given the coexistence of 3G and 4G communication standards, and even that of 5G, there are multiple frequency bands around the world. Traditional software radio designs require different hardware platforms to support varied communication protocols and frequency bands, and they require long development periods and high design costs.

Andrews et al. [1] suggested the use of a mixer as the first-level structure of a software radio to reduce noise at the cost of power with the supply voltage of the baseband low-noise amplifier increased. Murphy et al. (2012) put forward the use of a noise-canceling technology as the software radio structure. Useful signals were enhanced by increasing the auxiliary channels and offset noises at the output terminal, thus improving the noise reduction capability and the overall performance of the software radio system [2]. Zhang et al. propounded a software radio technique that can eliminate the spatial incident angle interference and frequency interference in signal filtering [3]. Loubser and Swart [4] encoded two existing CR-specific media access control protocols by using a CR-specific simulator. Kamaleldin and Ahmed put forward that hardware platforms of software radio system which support many wireless standards could be realized by dynamical program reconfiguration [5]. Marwanto et al. [6] proposed ARDUINO UNO and

* Correspondence: tianfeng0306@163.com

College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an, China

X-Bee technologies for software radio systems to reduce the costs of spectrum exchange information based on OFDM. Sahoo et al. propounded a multi-channel finite impulse response filter for software radio, which can reduce power consumption effectively by the launcher umbilical tower, and can be applied to software radio systems with multichannel filter efficiently [7].

Tsinghua University (2014) completed a chip for software radio receivers in the working frequency band of 0.1–5 GHz. Qin and Wang et al. constructed a radio communication system platform in Matlab and a universal software radio peripheral to increase the utilization of spectrum resources. Spectrum sensing and available spectral bandwidth estimation of signals of master users were obtained by performing an energy detection method. Thus, spectrum detection was realized, and a set of judgment criteria was provided for the spectral access of secondary users [8]. Xu and Yu designed and completed a fault prediction software platform for an airborne software radio system by analyzing its structure [9]. Cui [10] designed a communication terminal for the time-hopping spread spectrum of the TDMA system based on the software radio concept, realized the single-channel launching and multichannel reception of RFs, and accomplished the design of RF modules and their link terminals. Zhang et al. [11] designed a monitoring system over interferences and multipath in the signal bands and adjacent frequencies of current four navigation systems based on software-defined radio concept. Yin and Cheng [12] built a new hardware design program for the software radio processing platform with high-performance and low power consumption based on the requirements of special radio communication systems with low power consumption.

Extant studies have reported that software radio systems are generally limited by their structures. Studies based on AD9361 RF modules and processing modules of SOC digital basebands remain underexplored. Covering most bands with the charter and license-free bands, the working frequency range of AD9361 is from 70 MHz to 6.0 GHz. The supported channel bandwidth is from less than 200 kHz to 56 MHz. With the RF front end and the baseband of the flexible mixed signal integrated, AD9361 provides a configurable digital interface for the processor and integrates frequency synthesizer, thus simplifying the import of the design, which can achieve lower noise and higher accuracy of modulation with the high programmability. In this study, an AD9361-based software radio structure was constructed by using the broadband zero-IF bandpass sampling software radio structure. The corresponding software radio receiver was designed, which achieved many core functions,

such as group detection, frame synchronization, channel estimation, and frequency-domain equilibrium. The results can provide new universal platforms and methods for software radio receivers.

2 Methods

2.1 Structure of software radio system

The software radio system is mainly composed of RF module and digital baseband processing module. The RF module converts RF analog signals to baseband digital signals and vice versa, and it facilitates RF receiving and transmitting. The communication protocol in the physical layer is realized by the digital baseband processing module, which ensured signal encoding/decoding, and facilitated modulation and demodulation [13]. The structure of the software radio system is shown in Fig. 1.

2.2 Implementation of software radio receiving terminal

The receiving terminal of the software radio performs group detection, frame synchronization, channel estimation, frequency-domain equilibrium, RS decoding, inverse mapping, and so on. In the main signal processing of the digital baseband receiving module, the first step is to conduct group detection, followed by signal synchronization and channel estimation based on training sequences. Then, signals are balanced on the basis of data from channel estimation to compensate for the frequency-selective fading of signals caused by multipath transmission. Finally, RS decoding and 16QAM demodulation are accomplished. The structure of the software radio receiver is shown in Fig. 2.

2.3 The structure of AD9361

AD9361 is composed of the 2×2 transceiver, the configuration interface of Serial Peripheral Interface (SPI), the AUXADC, internal PLLs, the programmable GPO, and the data bus of 2×12 bits which can be configured as differential or single terminal. Each transmitting and receiving channel of AD9361 is independent. The two receiving channels are composed of low-noise amplifier, mixer, amplifier, filter, and ADC. The two transmitting channels are composed of DAC, filter, amplifier, low-noise amplifier, and mixer. The configuration interface of SPI is compatible with the standard mode of four lines. The structure of AD9361 is shown in Fig. 3.

2.4 Group detection

Group detection is performed by using the leading structure. The circuit used for group detection is shown in Fig. 4, which is designed and realized according to the circuit procedure [14].

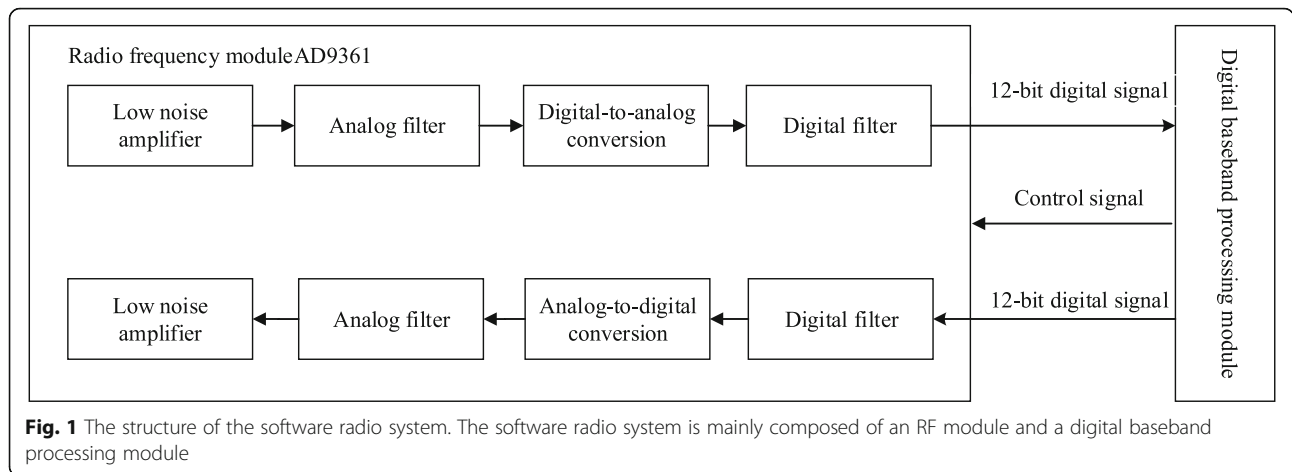


Fig. 1 The structure of the software radio system. The software radio system is mainly composed of an RF module and a digital baseband processing module

The value of delay correlation C_n is:

$$C_n = \sum_{k=0}^{L-1} r_{n-k} r_{n-k-D}^* \tag{1}$$

where r_n is the received signal, and C_n is the mutual correlation between the currently received L data and the L data received before D .

The value of received signal energy P_n is:

$$P_n = \sum_{k=0}^{L-1} r_{n-k-D} r_{n-k-D}^* = \sum_{k=0}^{L-1} |r_{n-k-D}|^2 \tag{2}$$

The decision variable m_n of the delay correlation algorithm is:

$$m_n = \frac{C_n}{P_n} = \frac{|\sum_{k=0}^{L-1} r_{n-k} r_{n-k-D}^*|}{\sum_{k=0}^{L-1} r_{n-k} r_{n-k-D}^*} \tag{3}$$

The value of m_n for group detection can be determined by the leading structure when the signal-to-noise ratio (SNR) is 15 dB. A value of C_n approaching 0 indicates that effective data transmission has not been achieved and only noise exists. Parameter m_n begins to increase with the

occurrence of the secondary short-training symbol and begins to decrease when the ninth period is reached.

A simulation on the basis of the ModelSim platform is conducted. The simulation results of group detection are shown in Figs. 5 and 6.

DataInRe and DataInIm are the real part and imaginary part of the current data, respectively. SumMagnitude is the sum of relevant window energies, and SumDelayCorrelation is the sum of correlation coefficients of relevant windows. BufferForDetection represents the initial judgment samples of 32 continuous groups, and BufferForDetection represents the judgment samples at the end of 48 continuous groups. As shown in Figs. 4 and 5, SumMagnitude and SumDelayCorrelation are immediately calculated after a system reset and when the effective signal of the grouping detection elevates. When 32 samples are detected continuously, data grouping of judgment arrives. When the SumMagnitude of the 48 continuous samples is smaller than the threshold, the data grouping is completed.

2.5 Frame synchronization

The frame synchronization of signals requires the calculation of the cross-correlation coefficient between the received data groups and the locally known short-training

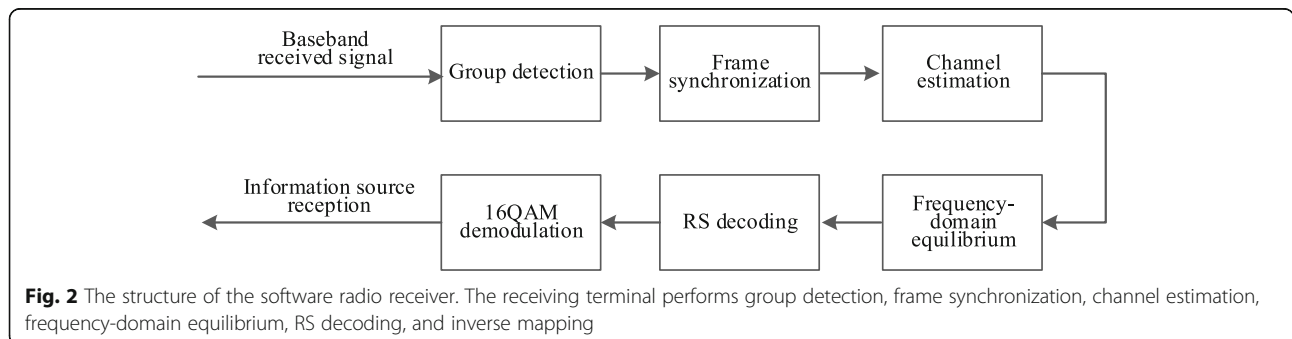


Fig. 2 The structure of the software radio receiver. The receiving terminal performs group detection, frame synchronization, channel estimation, frequency-domain equilibrium, RS decoding, and inverse mapping

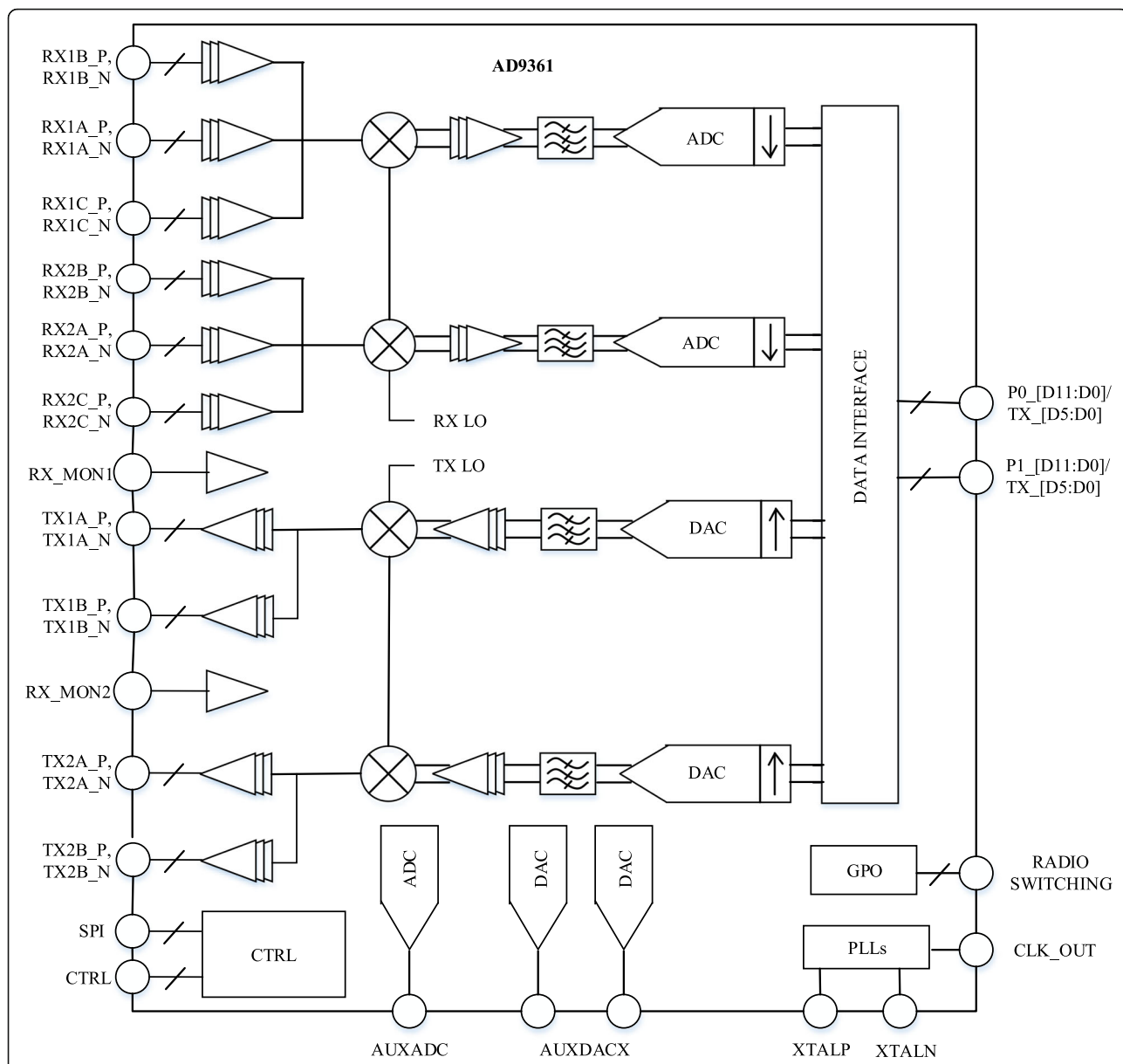


Fig. 3 The structure of AD9361. The whole circuit of RF and IF signals is integrated into one chip by adopting zero-IF architecture, which is composed of the transceiver, the configuration interface of SPI, the AUXADC, the internal PLLs, the programmable GPO, and the data bus

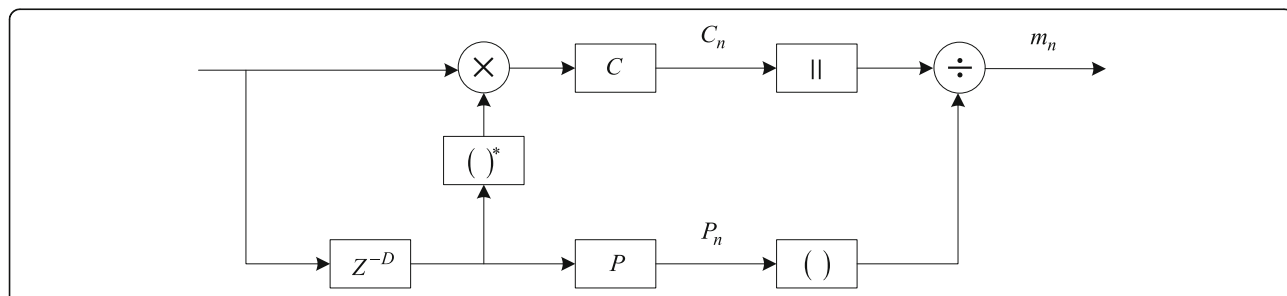


Fig. 4 Circuit procedure for group detection. Group detection is performed by using the leading structure. C_n is the mutual correlation between the currently received L data and the L data received before D . P_n is the value of received signal energy. m_n is the decision variable of the delay correlation algorithm

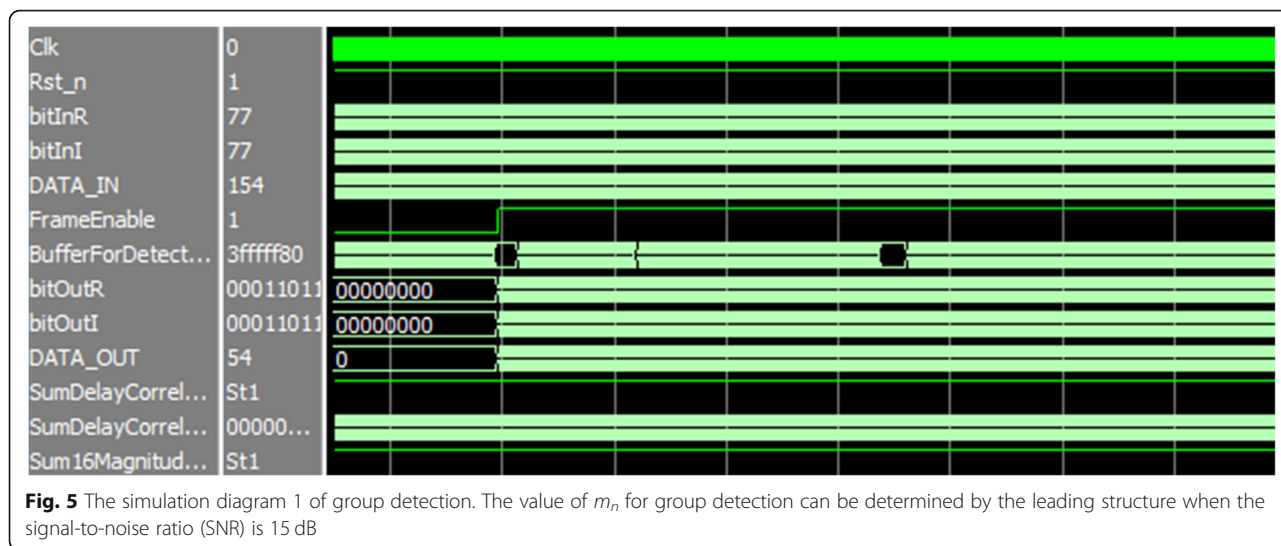


Fig. 5 The simulation diagram 1 of group detection. The value of m_n for group detection can be determined by the leading structure when the signal-to-noise ratio (SNR) is 15 dB

symbols [15]. Cross-correlation coefficient C_k can be expressed as:

$$C_k = \sum_{m=0}^{D-1} r_{k-m} \times S_m^* \tag{4}$$

where the superscript * is a conjugation, and D is the length of the cross-correlation coefficient, which is determined to be 16. The positions of the short-training symbols are judged according to the value of $|C_k|$. The moment of the last peak of $|C_k|$ is designated as the end point of the short-training symbols.

The simulation results of frame synchronization based on the ModelSim platform are shown in Fig. 7.

DataInRe and DataInIm are the real part and imaginary part of input data, respectively. DataInEnable is the enable signal of input data, and PCouter is the number of detected peaks. First, quantization is implemented when the data to be synchronized arrives. Then, the correlation is calculated on the basis of the 16 local short-training

symbols, and the moment at the ninth peak is viewed as the end point of the short-training symbols. Finally, long-training symbols and data symbols are designated with serial output according to the output format with the cyclic prefix eliminated at the same time. As shown in Fig. 7, DataOutEn is the effective time for outputting one symbol denoted as 1 and 2 successively which calculates from the long-sequence, with the data symbols started from 3. This scheme is viewed as one cycle of output data.

2.6 Channel estimation

Channel estimation is first performed to estimate the received signals from the time domain, according to which the estimation of frequency domain can be easily obtained [16]. Then, the received signals of the estimator can be expressed as:

$$r(t) = s(t) \times h(t) + n(t), \tag{5}$$

where $h(t)$ is the impulse response, $r(t)$ represents the received signals, $s(t)$ denotes the theoretically received

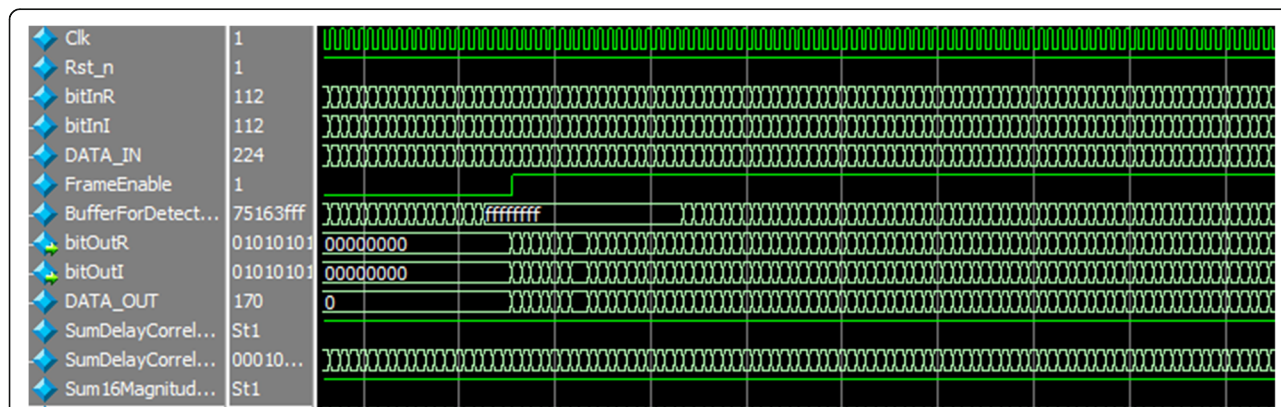
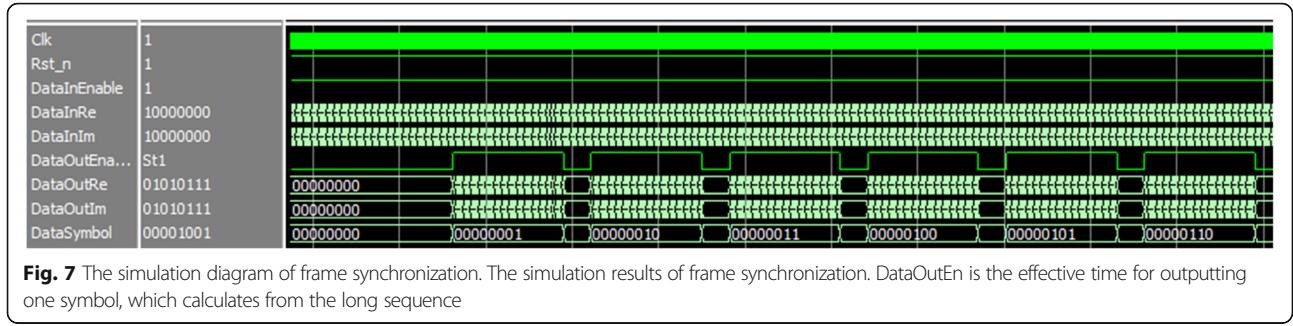


Fig. 6 The simulation diagram 2 of group detection. The value of m_n for group detection can be determined by the leading structure when the SNR is 15 dB



signals, and $n(t)$ is the signal noise. The estimated value of the input signal $\hat{s}(t)$ is produced by the convolution of inverse channel system $\hat{h}(t)$ that is composed of $r(t)$ and $h(t)$, where $h(t) \otimes \hat{h}(t) = \delta(t)$.

$$\begin{aligned} \hat{s}(t) &= [s(t) \otimes h(t) + n(t)] \otimes \hat{h}(t) \\ &= s(t) \otimes h(t) \otimes \hat{h}(t) + n(t) \otimes \hat{h}(t) \\ &= s(t) + \hat{n}(t) \end{aligned} \quad (6)$$

The estimation of the channel frequency response of the frequency domain can be deduced directly from the time domain. Channel frequency response $H(j\omega)$ is estimated by using $r(t)$, and the inverse channel system $\hat{H}(j\omega)$ is constructed by using $H(j\omega)$. Therefore,

$$\begin{aligned} \hat{s}(t) &= F^{-1}\{[F\{S(t)\} \cdot H(j\omega)] \cdot \hat{H}(j\omega)\} \\ &= F^{-1}\{F\{s(t)\}H(j\omega)\hat{H}(j\omega) + N(j\omega) \cdot \hat{H}(j\omega)\} \\ &= s(t) + F^{-1}\{N(j\omega) \cdot \hat{H}(j\omega)\}, \end{aligned} \quad (7)$$

where $\hat{H}(j\omega) = \frac{1}{H(j\omega)}$.

Channel estimation is realized by the unique words (UW) inserted into the data sequence. If UW is $\{x_m\}$ with a length of P , then the channel frequency response \hat{H}_k can be estimated by FFT from the transmitting sequences $\{x_m\}$ and $\{y_m\}$ to $\{x_m\}$ and $\{Y_m\}$.

$$\hat{H}_k = \frac{Y_k}{X_k} \quad k = 1, 2, 3, K, P-1 \quad (8)$$

The corresponding time-domain discrete signal $\{h_m\}$ can be initially obtained from the IFFT operation of \hat{H}_k at the point P , where P denotes the length. Then, the zero-padding operation of $\{h_m\}$ is performed, thus obtaining the $\{h_m\}$ of the point M . Finally, the frequency response value \hat{H}_k is acquired from the FFT operation of $\{h_m\}$ of the point M .

The frequency domain is calculated on the basis of the time domain as part of the channel estimation. The received signals can be expressed by the $Y = XH + V$ matrix as follows:

$$J_{LS} = (Y - XH)^H (Y - XH), \quad (9)$$

where $X = \text{diag}[X(0), X(1), \dots, X(N_{p-1})]$, $Y = [Y(0), Y(1), \dots, Y(N_{p-1})]^T$, and $H = [H(0), H(1), \dots, H(N_{p-1})]^T$. N_p is the number of UW. The first-order derivative and the second-order derivative of J_{LS} for H are calculated as follows:

$$\frac{\partial}{\partial H} J_{LS} = -2X^H(Y - XH), \quad (10)$$

$$\frac{\partial}{\partial H} \left(\frac{\partial}{\partial H} J_{LS} \right)^H = 2X^H X \geq 0. \quad (11)$$

The minimum can be derived from Eq. (11). If Eq. (11) is 0, then:

$$\hat{H}_{LS} = (X^H X)^{-1} X^H Y = X^{-1} Y = \frac{Y}{X}. \quad (12)$$

Thus, the estimated value denoted by \hat{H}_{LS} can be expressed as:

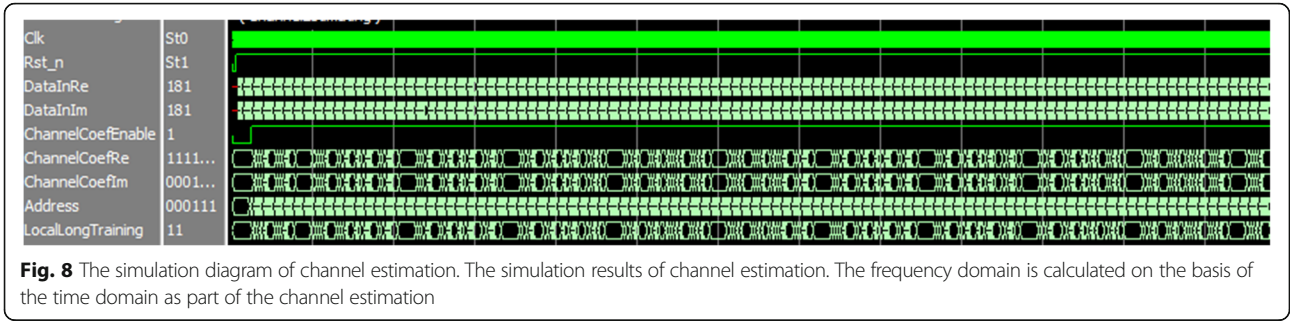
$$\hat{H}_{LS} = X^{-1}(XH + V) = H + X^{-1}V = H + n, \quad (13)$$

where n is the estimation error, and $n = X^{-1}V$. The simulation results of channel estimation based on the Model-Sim platform are shown in Fig. 8.

DataInRe and DataInIm are the real part and imaginary part of input data, respectively. DataInEnable is the enable signal of input data, and ChannelcoeEnable is the enable signal of output data. ChannelcoeIm and ChannelcoeRe are the real part and imaginary part of output channel estimation, respectively.

2.7 Frequency-domain equilibrium module

Frequency-domain equilibrium is performed to offset the effects of channels on the received signals. Here, the minimum mean-squared error (MMSE) equilibrium



algorithm is applied as the frequency-domain equilibrium [17].

Suppose that the set of transmitting data is denoted by s ($s = [s_0, s_1, \dots, s_{N-1}]^T$), N is the number of FFT points ($h = [h_0, \dots, h_{L-1}, 0, \dots, 0]^T$), and L is the length of impulse response. Then, the received signal vector is $r = [r_0, r_1, \dots, r_{N-1}]^T$. Accordingly,

$$r = h \otimes s + v \tag{14}$$

where \otimes is the cyclic convolution, and $v = [v_0, v_1, \dots, v_{N-1}]^T$ is the channel noise. On the basis of the FFT of Eq. (14),

$$R_k = S_k H_k + V_k, \tag{15}$$

where $H_k = \sum_{m=0}^{L-1} (h_m e^{-j2\pi km/N})$, and R_k , S_k , H_k , and V_k are the frequency domain values of received signals, transmitting signals and channel impulse response function, and additive white Gaussian noise.

If the equilibrium coefficient is W_k , then the frequency-domain output after equilibrium is:

$$\tilde{S}_k = W_k H_k S_k + W_k V_k. \tag{16}$$

According to its definition, the mean square error (MSE) after the equilibrium can be deduced.

$$\begin{aligned} \text{MSE} &= E \sum_{k=0}^{N-1} |\tilde{S}_k - S_k|^2 \\ &= E \sum_{k=0}^{N-1} |[W_k H_k - 1] S_k + W_k V_k|^2 \\ &= \sum_{k=0}^{N-1} E |[W_k H_k - 1] S_k + W_k V_k|^2 \\ &= \sum_{k=0}^{N-1} F_k \end{aligned} \tag{17}$$

Suppose that σ_N^2 is the noise power on the frequency domain and σ_S^2 is the signal power on the frequency domain, then:

$$\sigma_N^2 = E |V_k|^2, \tag{18}$$

$$\sigma_S^2 = E |S_k|^2. \tag{19}$$

On the basis of Eq. (18) and Eq. (19), the following can be derived:

$$F_k = |W_k H_k - 1|^2 \sigma_S^2 + |W_k|^2 \sigma_N^2. \tag{20}$$

If $\arg z_1 = \arg z_2$, $|z_1 - z_2| \geq ||z_1| - |z_2||$ ($z_1, z_2 \in W$) is true, then:

$$F_k \geq |W_k H_k - 1|^2 \sigma_N^2 = (|W_k H_k| - 1)^2 \sigma_S^2 + |W_k|^2 \sigma_N^2, \tag{21}$$

where F_k uses the lower limit value based on the condition of $\arg W_k H_k = \arg 1 = 0$.

$$\arg W_k = -\arg H_k \tag{22}$$

If the lower limit of F_k is y , then the minimum y should be calculated, such that:

$$\begin{aligned} y &= (|W_k H_k| - 1)^2 \sigma_S^2 + |W_k|^2 \sigma_N^2 \\ &= |H_k^2 \sigma_S^2 - \sigma_N^2| |W_k|^2 - 2 H_k \sigma_S^2 |W_k| + \sigma_S^2. \end{aligned} \tag{23}$$

To derive the minimum of y :

$$W_k = \frac{H_k^* \sigma_S^2}{|H_k|^2 \sigma_S^2 + \sigma_N^2} = \frac{H_k^*}{|H_k|^2 + \frac{\sigma_N^2}{\sigma_S^2}}. \tag{24}$$

Given that $\frac{\sigma_S^2}{\sigma_N^2} = \text{SNR}$, the MMSE equilibrium coefficient can be expressed as $W_k = \frac{H_k^*}{|H_k|^2 + 1/\text{SNR}}$. In Eq. (24), $0 \leq k \leq N-1$ and SNR denotes the signal-to-noise ratio of the transmitting terminal.

The simulation of signals based on the MMSE algorithm is conducted in Matlab. The following parameters are included:

Multipath channel: the corresponding power of the SUI.3 channel model is [0, -5, -10 dB];

Modulation mode: 16QAM;

SC-FDE system parameters: UW uses the Chu sequence, and the length is $N = 64$;

$M = 256$, and the MMSE equilibrium algorithm is used.

The hardware implementation block diagram of channel equalization module based on FPGA is shown in Fig. 9.

SC-FDE symbols in the time domain are read from "buffer of RX frame sample" by the channel equalization module and sent to the FFT module to calculate the frequency domain values of SC-FDE symbols.

Frequency domain values are read from "buffer of CSI" by the CSI_ACQ module, which can complete the integration of corresponding samples meanwhile.

With the complex multiplication of the corresponding sample points completed by the FDE_CORE module, the frequency domain equalization is achieved.

Meanwhile, with the subsequent IFFT_256 module controlled by the FDE_CORE module, the sample points in the frequency domain after the equilibration are restored to the time domain and stored in the symbol buffer of SIG domain and the time buffer of DATA domain, respectively.

The simulation results of the channel equilibrium based on the ModelSim platform are shown in Fig. 10.

DataInRe and DataInIm are the real part and imaginary part of the input data, respectively. DataInEnable is the enable signal of input data. DataOutRe and DataOutIm are the real part and imaginary part of output data, respectively.

2.8 RS decoding

The design procedures of the RS decoder are as follows:

1. The adjoint polynomial $s(x)$ of RS codes is calculated from the receiving codes.
2. The error position polynomial $a(x)$ and error value polynomial $\delta(x)$ are solved by an adjoint polynomial.
3. The error position can be acquired by using the Chien searching method to calculate the roots of error location polynomials.
4. The error magnitude corresponding to each error location can be obtained from the error value polynomial by using the Fomey algorithm, namely $C(x) = R(x) - E(x)$.
5. After decoding, the adjoint formula of the codeword is calculated again, and the adjoint formula is determined by detecting whether the adjoint formula is zero or not.

According to the above procedures, RS decoder should include four parts: the adjoint polynomial calculation module, the key equation solving module, the money search module, and the Fomey algorithm module [18].

Specific procedures of RS decoding design are as follows:

1. Solving the adjoint polynomial of RS decoding. The parameters of RS (255,191) are as follows:

Encoding length: $n = 255$
 Information bit length: $k = 191$
 Parity bit length: 64
 Error correcting capability: $t = 32$
 Primitive polynomial:

$$p(x) = x^8 + x^4 + x^3 + x^2 + x^1 + 1. \tag{25}$$

The generating polynomial of RS (255,191) is as follows:

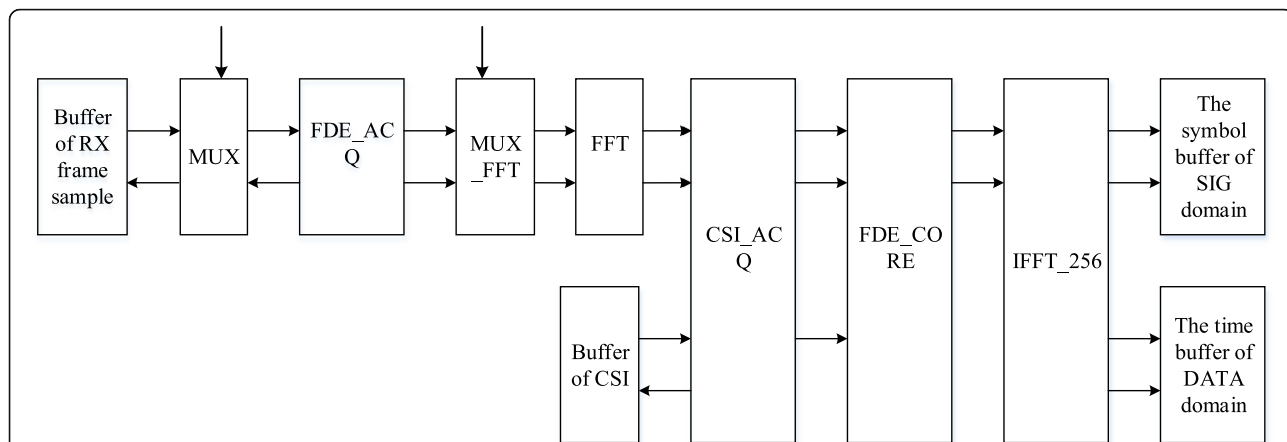
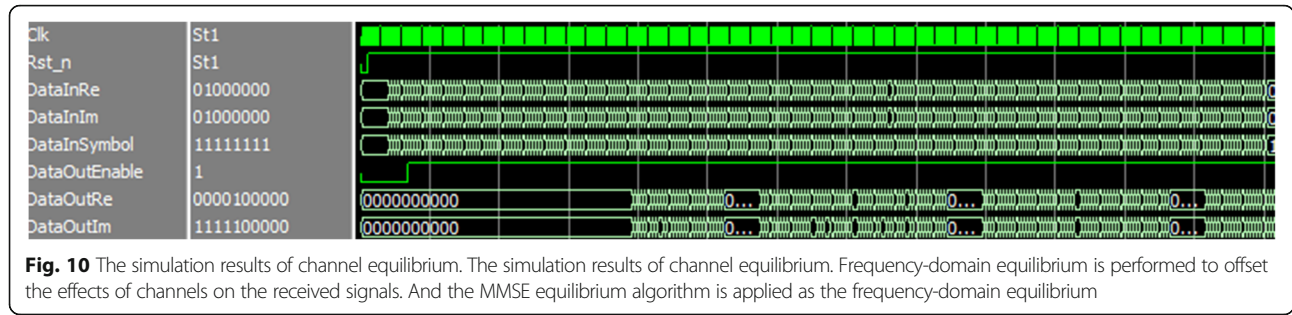


Fig. 9 The hardware implementation block diagram of channel equalization module. The hardware implementation block diagram of channel equalization module. SC-FDE symbols in the time domain are read from "buffer of RX frame sample" by the channel equalization module and sent to the FFT module to calculate the frequency domain values of SC-FDE symbols. Frequency domain values are read from "buffer of CSI" by the CSI_ACQ module, which can complete the integration of corresponding samples meanwhile. With the complex multiplication of the corresponding sample points completed by the FDE_CORE module, the frequency domain equalization is achieved



$$g(x) = (x-a^0)(x-a^1)(x-a^2)\Lambda(x-a^{15}). \tag{26}$$

$$\delta(x) = 1 + \delta_1x + \delta_1x^2 + \Lambda + \delta_1x^t. \tag{30}$$

The RS is solved by using $a, a^2, a^3, \dots, a^{32}$, in which $R(x) = r_0 + r_1x + r_2x^2 + \dots + r_{n-1}a^{(n-1)}$. The 32 adjoint expressions of RS (255,191) codes are acquired, namely $s_1, s_2, s_3, \dots, s_{32}$.

$$s(x) = \sum_{i=1}^{2t} s_i x^{i-1}, s_i = R(a^i), (1 \leq i \leq 2t) s_i = R(a^i) \tag{27}$$

2. Solving the error position polynomial. Firstly, the error location polynomial $\delta(x)$ is obtained, then the error location polynomial and the error value polynomial are obtained. The error location polynomial $\delta(x)$ can be defined as:

$$\delta(x) = (1-\theta_1x)(1-\theta_2x)\Lambda(1-\theta_t x). \tag{28}$$

The error location is $\theta_1, \dots, \theta_t$. The right part of the expansion equation is simplified as follows:

$$\delta(x) = 1 + (\theta_1 + \theta_2 + \Lambda + \theta_t)x + (\theta_1\theta_2 + \theta_1\theta_3 + \Lambda + \theta_{t-1}\theta_t)x^2 + \Lambda + \theta_1\theta_2\Lambda + \theta_1x^t. \tag{29}$$

The error location polynomial is acquired:

3. Solving the error position. The error location of receiving polynomial $R(x) = r_0 + r_1x + \Lambda + r_{n-2}x^{n-2} + r_{n-1}x^{n-1}$ is acquired according to the root of δ_1x .

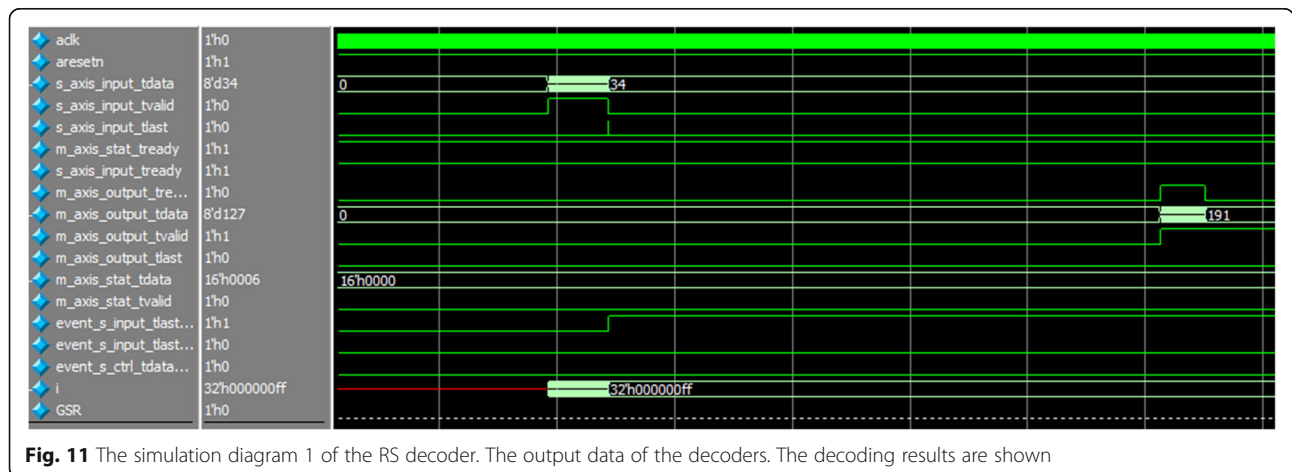
4. Ascertain the error pattern $E(x)$ and the polynomial of $c(x)$. The error value polynomial is defined as follows: $\omega(x) = S(x)\delta(x)$, which is simplified as follows:

$$\omega(x) = s_1x + (s_2 + \delta_1s_1)x^2 + \Lambda + (s_t + \delta_1s_{t-1} + \Lambda + \delta_{t-1}s_1)x^t + (s_{t+1} + \delta_t s_1 + \Lambda + \delta_1s_1)x^{t+1} + \Lambda. \tag{31}$$

The following equations can be verified.

$$\begin{aligned} s_{t+1} + \delta_1s_t + \Lambda + \delta_t s_1 &= 0 \\ s_{t+2} + \delta_1s_t + \Lambda + \delta_t s_2 &= 0 \\ s_{2t} + \delta_1s_{2t+1} + \Lambda + \delta_t s_t &= 0 \end{aligned} \tag{32}$$

The error value polynomial $\omega(x) = \omega_1x + \omega_2x^2 + \omega_t x^t$ can be obtained if $\omega_1 = s_1, \omega_2 = s_2 + \delta_1s_1, \Lambda, \omega_t = s_t + \delta_1s_{t-1} + \Lambda\delta_{t-1}s_1$. The error pattern $E(x) = \sum_{i=1}^t Y_i x_i^t$ can be obtained by $Y_j = \frac{-x_j \omega(x_j^{-1})}{\delta(x_j^{-1})}$. Here, x_j is the root of the Chien searching method. The final actual code $C(x)$ is obtained with $E(x)$ and the receiving code $R(x)$ superposed.



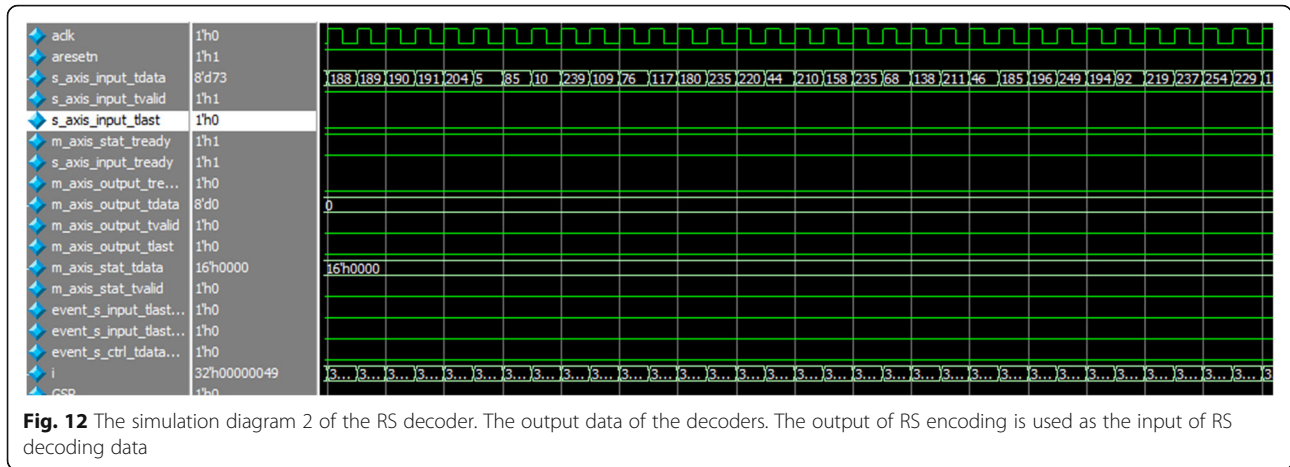


Fig. 12 The simulation diagram 2 of the RS decoder. The output data of the decoders. The output of RS encoding is used as the input of RS decoding data

5. Calculating the adjoint formula of the codeword again. The decoding result is determined by detecting whether the adjoint formula is zero or not.

The RS decoding is implemented on FPGA. The simulation results are shown in Figs. 11, 12, and 13.

As is shown in Figs. 11 and 12, the output of RS encoding is used as the input of RS decoding data. The input is as follows: (1, 2, 3, ..., 190, 191, 204, 5, 85, 10, 239, 109, 76, 117, 180, 235, 220, 44, 210, 158, 235, 68, 138, 211, 46, 185, 196, 249, 194, 92, 219, 237, 254, 229, 151, 239, 246, 19, 26, 219, 66, 100, 210, 157, 6, 208, 187, 169, 68, 168, 78, 28, 34, 163, 42, 134, 149, 43, 0, 88, 70, 90, 93, 129, 173, 131, 235, 192, 66, 34). If the output data of RS decoding is the input of RS encoding, the output is correct. Namely, the output is (1, 2, 3, ..., 190, 191). As is shown in Fig. 13, the RS decoding decodes the encoded data. The RS encoding is correct.

2.9 16QAM demodulation module

RS decoding is followed by the 16QAM demodulation. With the orthogonal coherent demodulation

method applied, the signal is judged, detected, and converted in series and parallel, and the final output is generated.

The expressions of demodulated *I* branch and *Q* branch are shown in Eq. (33).

$$\begin{cases} I_{\text{branch}} = S_{\text{QAM}} \cos at = \frac{1}{2}X_k + \frac{1}{2}X_k \cos 2at + \frac{1}{2}Y_k \sin 2at \\ Q_{\text{branch}} = S_{\text{QAM}} \sin at = \frac{1}{2}Y_k + \frac{1}{2}Y_k \cos 2at + \frac{1}{2}X_k \sin 2at \end{cases} \quad (33)$$

With $\frac{1}{2}X_k \cos 2at + \frac{1}{2}Y_k \sin 2at$, $\frac{1}{2}Y_k \cos 2at + \frac{1}{2}X_k \sin 2at$, $\frac{1}{2}X_k$, and $\frac{1}{2}Y_k$ filtered by the low pass filter, the output of 16QAM demodulation is obtained. The expression is as follows:

$$\begin{cases} I_{\text{branch}} = \frac{1}{2}X_k \\ Q_{\text{branch}} = \frac{1}{2}Y_k \end{cases} \quad (34)$$

The constellation of the 16QAM modulation is shown in Fig. 15, and its mapping output value is $d = (I$

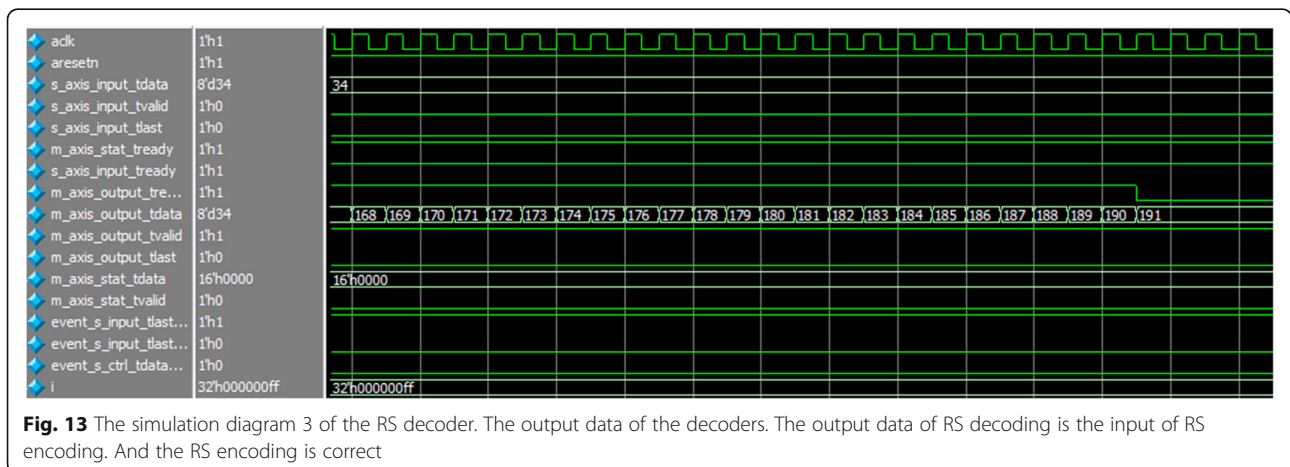
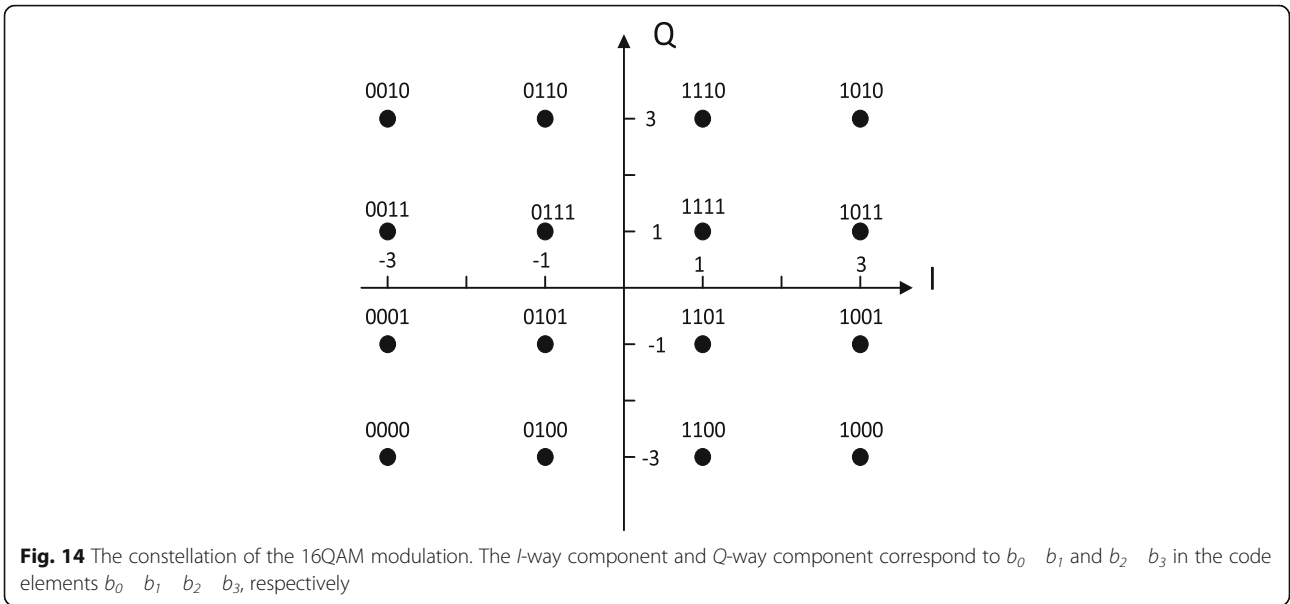


Fig. 13 The simulation diagram 3 of the RS decoder. The output data of the decoders. The output data of RS decoding is the input of RS encoding. And the RS encoding is correct



$+jQ) \times K_{MOD}$, where $K_{MOD} = 1/\sqrt{10}$. The constellation of the 16QAM modulation is shown in Fig. 14.

The I -way component and Q -way component correspond to b_0, b_1 and b_2, b_3 in the code elements b_0, b_1, b_2, b_3 , respectively. With the decision thresholds set as $-2 \times K_{MOD}$, 0 and $2 \times K_{MOD}$, respectively, the I and Q can be demodulated now.

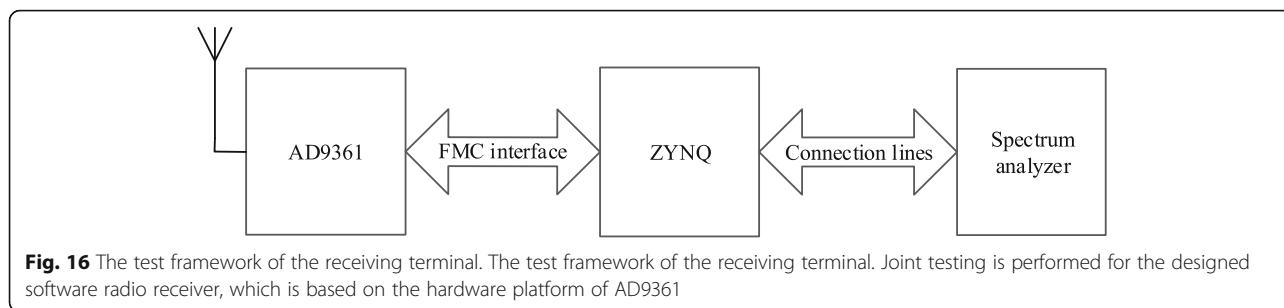
3 Experiment results and discussion

With the AD9361 used as the RF module, ZC706 applied as the digital baseband processing module of SOC,

and ZC706 utilized as the ARM+FPGA framework, the hardware platform of the software radio system is built. The physical connection between AD9361 and ZC706 is shown in Fig. 15. The AD9361 board card and the ZYNQ ZC706 development board are connected by FMC. A spectrum analyzer is used as the tester of the transmitting and receiving terminals during the system test. The accuracy of the system test is evaluated by observing the frequency spectra.

Joint testing is performed for the designed software radio receiver, which is based on the hardware platform of AD9361. The test framework of the receiving terminal





is shown in Fig. 16. The transmitting central frequency, transmit gain, and bandwidth are 1.435 GHz, 15 dB, and 20 MHz, respectively. AD9361 is connected to ZYNQ via FMC, and ZYNQ is connected to the spectrum analyzer via JTAG. The final results are displayed through the spectrum analyzer. The detailed procedure can be described as follows: signals are received, signals are inputted into AD9361 via the antenna, and these input signals are sent into the digital baseband processing module to complete the processing after amplification, mixing, filtering, and A/D conversion based on AD9361.

AD9361 is set as the working modes of 1R1T, LVDS, and TDD. The ADC frequency is set as 13 MHz, and the local frequency is set as 1.435 GHz, which can be displayed through the spectrum analyzer. The frequency spectra at the receiving terminal are shown in Fig. 17.

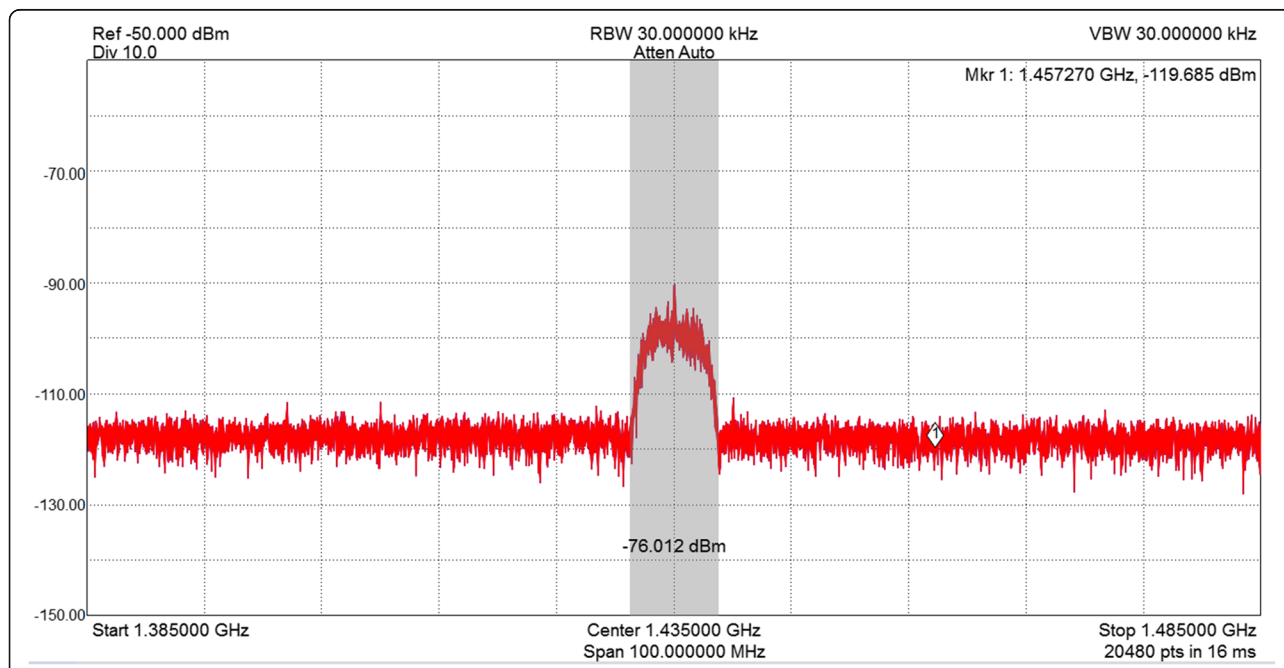
Joint testing is also performed for the transmitting signals and the receiver, with the transmitting and receiving frequency spectra examined, which are shown in Fig. 18.

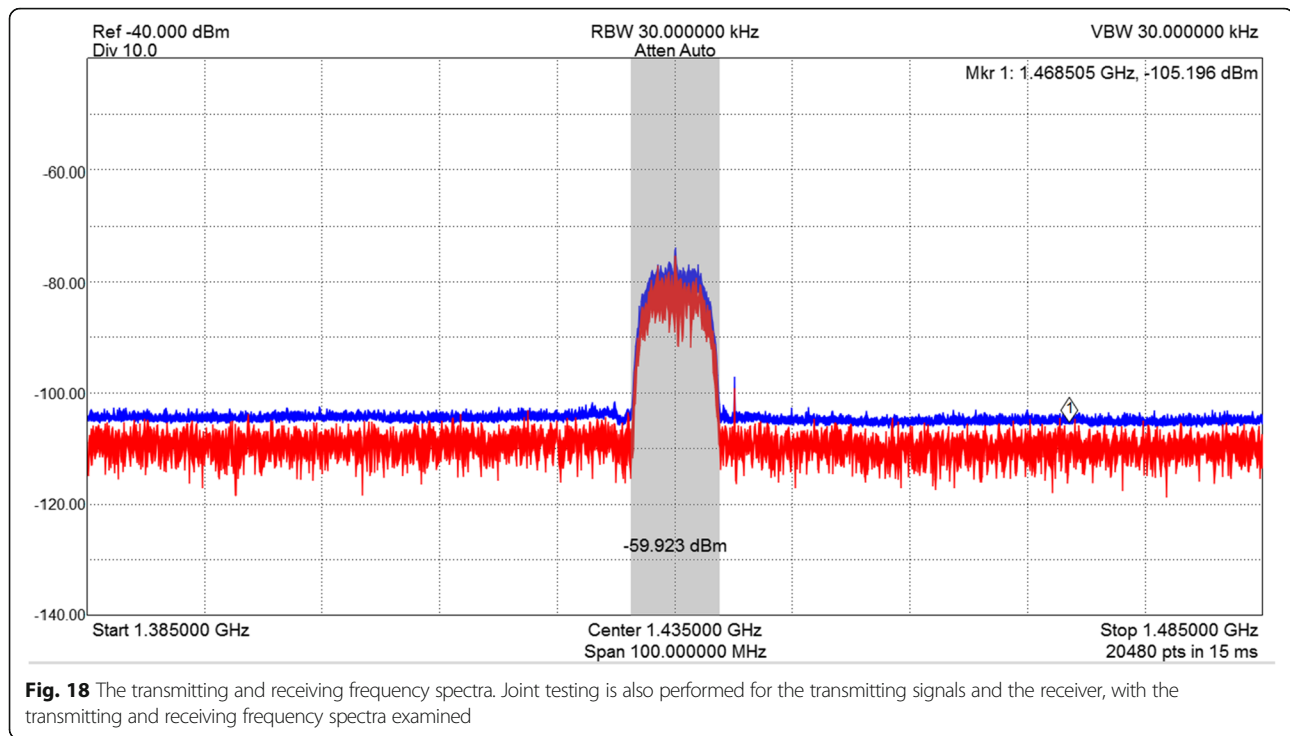
As shown in Fig. 16, the transmitting frequency spectrum is located in the upper position, while the receiver frequency spectrum is located in the lower position. The overlapping of the frequency spectra of the transmitting signals and receiver indicates the consistency of parameters between the transmitting and receiving signals. Therefore, transmitting signals are received accurately.

In this study, a joint experiment of RF module and digital baseband processing module is carried out. The experiment is carried out by two modules combined with a signal source and a spectrum analyzer. But there is no video display part and signal compression part, which can be added to the system to promote the applications in our future research.

4 Conclusions

An AD9361-based software radio system was designed by using AD9361 as the hardware platform. The receiving





function is realized, with the modules for group detection, frame synchronization, channel estimation, and frequency-domain equilibrium designed. Finally, the results of the software radio receiver were optimized and verified. The major conclusions from this study are as follows:

1. The single-carrier frequency domain equilibrium is applied on the digital baseband physical layer, which is optimized on the basis of the SC-FDE communication protocol. Then, the receiving function on the basis of the software radio receiver is realized by using the optimized communication protocol. The system can effectively resist the frequency-selective fading of channels, thus achieving high-rate and large-capacity communication transmission.
2. Peak to average power ratio and the sensitivity to the phase noise are decreased by the MMSE equilibrium algorithm, thus enhancing the resistance against multipath interference.

The AD9361-based software radio receiver presented in this study has a reasonable structure, and its performance indexes are satisfactory. The proposed system can effectively increase the communication speed and capacity, remarkably ameliorate the reduction of signal quality caused by multipath fading, and excellently offset the poor universality and expansibility of traditional radio receivers. The results of this study can serve as a useful reference in the development of next-generation universal software radio receivers.

Abbreviations

FFT: Fast Fourier transform algorithm; FMC: FPGA Mezzanine Card; FPGA: Field programmable gate array; IF: Intermediate frequency; JTAG: Joint Test Action Group; MMSE: Minimum mean-squared error; MSE: Mean square error; QAM: Quadrature amplitude modulation; RF: Radio frequency; SC-FDE: Single-carrier frequency domain equalization; SNR: Signal-to-noise ratio; SOC: System on chip; SPI: Serial Peripheral Interface; TDMA: Time-division multiple address; UW: Unique words

Acknowledgements

This work was supported by the Project of Science and Technology of Shaanxi (No.2018GY-151).

Funding

This work was supported by the Project of Science and Technology of Shaanxi (No.2018GY-151).

Availability of data and materials

The authors declare that all the data and materials in this manuscript are available.

Authors' contributions

The contributions of all authors are equal in this manuscript, and all authors read and approved the final manuscript.

Authors' information

Feng Tian is currently an associate professor of College of Communication and Information Engineering, Xi'an University of Science and Technology. Hanqing Li is currently pursuing an MS degree at the College of Communication and Information Engineering, Xi'an University of Science and Technology. Liangchen Yuan is currently pursuing an MS degree at the College of Communication and Information Engineering, Xi'an University of Science and Technology.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 14 January 2019 Accepted: 28 March 2019

Published online: 15 April 2019

References

1. C. Andrews, A passive mixer-first receiver with digitally controlled and widely tunable RF interface. *IEEE J. Solid State Circuits* **45**(12), 2696–2708 (2010). <https://doi.org/10.1109/JSSC.2010.2077151>
2. H. Darabi, D. Murphy, M. Mikhemar, H. Wu, *2.1 A Highly Linear Inductor Less Wideband Receiver with Phase- and Thermal-Noise Cancellation* (IEEE International Solid-State Circuits Conference, San Francisco, 2015), pp. 1–3. <https://doi.org/10.1109/ISSCC.2015.7062850>
3. L. Yang, Z. Zhang, W. Hou, B. Zhao, H. Zheng, Papyrus: a software platform for distributed dynamic spectrum sharing using SDRs. *ACM SIGCOMM Computer Communication Review* **41**(1), 32–37 (2017). <https://doi.org/10.1145/1925861.1925866>
4. C.J. Loubser, T.G. Swart, *Cognitive Radio DAB MAC Protocol Performance Using a CR Specific Simulator and Software Defined Radio* (IEEE AFRICON, Cape Town, 2017), pp. 139–144. <https://doi.org/10.1109/AFRCON.2017.8095470>
5. A. Kamaleldin, S. Hosny, K. Mohamed, M. Gamal, A. Hussien, E. Elnader, A. Shalash, A.M. Obeid, Y. Ismail, H. Mostafa, *A Reconfigurable Hardware Platform Implementation for Software Defined Radio Using Dynamic Partial Reconfiguration on Xilinx Zynq FPGA* (2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, 2017), pp. 1540–1543. <https://doi.org/10.1109/MWSCAS.2017.8053229>
6. A. Marwanto, S.K.S. Yusof, M.H. Satria, Orthogonal frequency-division multiplexing-based cooperative spectrum sensing for cognitive radio networks. *Telkomnika (Telecomm Comput Electron Contr)* **12**(1), 143–152 (2014). <https://doi.org/10.12928/TELKOMNIKA.v12i1.310>
7. R. Srinivasa, S. Kotha, S. Kumar, An approach for fixed coefficient RNS-based FIR filter. *Int. J. Electron.*, 1–19 (2017). <https://doi.org/10.1080/00207217.2017.1296593>
8. X. Qin, L. Wang, H. Zhao, Y. Deng, Realization of spectrum sensing based on MATLAB and general software radio peripherals. *Comput. Eng.* **41**(5), 106–110 (2015)
9. S. Xu, Z. Yu, K. Chang, Y. Wu, B. L., Design and implementation of fault prediction software for airborne software radio system. *Meas Contr Technol* **35**(8), 111–114 (2016). <https://doi.org/10.3969/j.issn.1000-8829.2016.08.028>
10. Z. Cui, Design of frequency hopping communication terminal based on software radio principle. *Modern Navigation* **8**(5), 365–371 (2017)
11. Y. Zhang, X. Cui, T. Da, G. Wang, Design of GNSS jamming and multipath monitoring system based on software radio. *Telecommun. Eng.* **57**(3), 288–295 (2017)
12. Z. Yin, L. Cheng, Design of low power software radio platform based on ZYNQ7000. *Wireless Internet Technol* **37**(3), 31–32 (2018)
13. M. Mishra, A. Potnis, P. Dwivedy, S.K. Meena, *Software Defined Radio Based Receivers Using RTL — SDR: a Review* (2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, 2017), pp. 62–65. <https://doi.org/10.1109/RISE.2017.8378125>
14. Y. Qi, X. Chen, Z. Xie, S. Yuan, *An Improved MMSE-RISIC Equalization Algorithm Based on STBC-SC-FDE* (2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, 2018), pp. 310–314. <https://doi.org/10.1109/SIPROCESS.2018.8600476>
15. K. Kumar, S. Pillai, S. Sagar, S.K. Parambeth, N. Prem Krishnan, *Implementation of Digital Pre-Distortion for Power Amplifier Linearisation in Software Defined Radio* (Twenty-third National Conference on Communications (NCC), Chennai, 2017), pp. 1–6
16. J. David Cepeda, S.I. Rodríguez, M. Rico-Martínez, C. Daniel Muñoz, M. Varón, I.T. Monroy, *Performance Evaluation of a Real Time OFDM Radio over Fiber System at 2.5 GHz Using Software Defined Radio SDR* (SBMO. IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), 2017), pp. 1–5. <https://doi.org/10.1109/IMOC.2017.8121094>
17. E. Balevi, A.Ö. Yilmaz, Analysis of frequency domain oversampled MMSE SC-FDE. *IEEE Commun. Lett.* **20**(2), 232–235 (2016)
18. W. Lu, Y. Liang, Y. Liu, Z. Liang, W. Zhang, The design of an RS decoder based on the mCS-RiBM algorithm for 100 Gb/s optical communication systems. *IEEE Trans Circuits Syst Express Briefs* **66**(1), 76–80 (2019)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)