American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Design of GMSK Frequency Hopping Modulation Scheme Based on FPGA

Lijun Ge[a*], Xiaoying Li[b], Xiaoxiao Wang[c]

[a,b,c]*School of Electronics and Information Engineering, Tianjin Polytechnic University, Tianjin 300387, China*

[a]*Email: gelj@mail.nankai.edu.cn*

[b]*Email: 1002477935@qq.com*

[c]*Email: yan7317@qq.com*

Abstract

An FPGA-based GMSK frequency hopping modulation scheme is designed. The design of the frequency hopping module is added to modulation scheme, and used the digital implementation of GMSK modulation. The pre-encoded symbols are first processed by a Gaussian filter and then sent to a phase accumulator to calculate the accumulated phase. The accumulated phase is summed with the phase generated by frequency-hopping carriers of frequency hopping module to obtain the total phase, and finally we use the CORDIC algorithm to perform the cosine operation on the total phase to realize the GMSK frequency hopping modulation of the signal. This scheme is implemented on the FPGA. After testing and verification, this scheme can realize the frequency hopping function of the system. The design of the phase accumulator ensures the phase continuity of GMSK signal, and the CORDIC algorithm eliminates a large number of multiplication operations. The hardware implementation is simpler and improves the feasibility of the scheme.

Keywords: GMSK modulation; frequency hopping communication; FPGA; phase accumulator; CORDIC algorithm.

---

* Corresponding author.

1. Introduction

In the national defense of security and military war, the role of communication technology is increasingly important, if the transmitted information is discovered by enemy, it will cause an inestimable damage. So, it is necessary to use a communication technology to ensure the validity and security of the communication. Frequency hopping communication is widely used in various strategic and tactical communication systems due to its interference immunity and confidentiality [1].

In the frequency hopping communication, the carrier is pseudo-randomly hopped with the change of the hopping sequence, and the transmission bandwidth is increased, so it is difficult for enemy to intercept data when the enemy is in an unknown hopping rule. Therefore, the frequency hopping communication system has strong interference immunity and high reliability [2]. At the same time, in the frequency hopping communication system, the excellent spectral characteristics and high frequency efficiency of GMSK (Gaussian Minimum Shift Keying) modulation can effectively improve the transmission speed of the system [3].

The existing GMSK modulation generally adopts the orthogonal modulation method of waveform storage. Firstly, the input signal is encoded to generate two orthogonal waveform data, and then the information of the waveform in various phase paths is stored to make a look-up table, and the look-up table method is used to implement GMSK modulation [4]. However, this modulation method does not implement the frequency hopping function of the system, and there is a quadrature phase error of the two tributary signals, which eventually leads to a phase error of the modulated signal, so it cannot ensure the phase continuity of the modulated signal. At the same time the operation of this method is more complicated and requires a lot of storage space.

To solve these problems, a new GMSK frequency hopping modulation method is proposed in this paper, which adds the frequency hopping module to the GMSK modulation system, so it can easily realize the control of frequency hopping and increased the security and confidentiality of the frequency hopping system. At the same time, the system can enhance the controllability of frequency hopping, and the GMSK modulation method can ensure the phase continuity of the GMSK signal, simplify the modulation process and save a lot of storage space.

2. Modulation Principle of GMSK

GMSK is an improved modulation technique based on minimum frequency shift keying, which means that the input signal need to be preprocessed by Gaussian low-pass filter before modulation[5]. The impulse response expression of the Gaussian low-pass filter is expressed as follows :

$$h(t) = \frac{\sqrt{\pi}}{\alpha} \exp\left[(-\frac{\pi}{\alpha} t)^2 \right] \qquad (1)$$

In equation (1) α is a constant related to the 3dB bandwidth of the transfer function.

The Gaussian impulse response of a rectangular pulse of width T_b is:

$$g(t)=\frac{1}{2T_b}[Q(\frac{2\pi B(t-T_b/2)}{\sqrt{\ln 2}}-Q(\frac{2\pi B(t+T_b/2)}{\sqrt{\ln 2}}] \qquad (2)$$

Where function $Q(x)$ is expressed as:

$$Q(x)=\int_x^{\infty}\frac{1}{\sqrt{2\pi}}\exp(-\tau^2/2)d\tau=\frac{1}{2}erfc\frac{x}{\sqrt{2}} \qquad (3)$$

In equation (2), *B* is the 3dB bandwidth of the Gaussian filter, and the input of the Gaussian filter is Bipolar non-return to zero code, the expression is:

$$m(t)=\sum a_n b(t-nT_b) \qquad (4)$$

Where $a_n=\pm 1, b(t)=\begin{cases}1/T_b & 0\le|t|T_b/2\\ 0 & \text{else}\end{cases}$ so the output signal of the Gaussian low-pass filter is:

$$X(t)=m(t)*h(t)=\sum a_n g(t-nT_b) \qquad (5)$$

In summary, the final output of the GMSK modulated signal is:

$$S_{GMSK}(t)=\sqrt{\frac{2E}{T_b}}\cos\left\{\omega_c t+\frac{\pi}{2T_b}\int_{-\infty}^{t}[\sum_{k=-\infty}^{n}a_k g(\tau-kT_b-\frac{T_b}{2})]d\tau\right\}=\sqrt{\frac{2E}{T_b}}\cos[\omega_t t+\phi(t)] \qquad (6)$$

E is the signal energy in the symbol period T_b, $\phi(t)$ is the baseband phase, and the expression of baseband phase can be obtained from equation (6):

$$\phi(t)=\frac{\pi}{2T_b}\int_{-\infty}^{t}[\sum_{k=-\infty}^{n}a_k g(\tau-kT_b-\frac{T_b}{2})]d\tau \qquad (7)$$

3. Design Scheme of GMSK Frequency Hopping Modulation

Normalizing the amplitude of GMSK modulated signal of equation (6) and obtaining the expression of the GMSK signal:

$$S_{GMSK}=\cos[\omega_c(t)+\phi(t)] \qquad (8)$$

The schematic diagram of GMSK frequency hopping modulation according to the above formula is shown in Figure 1.

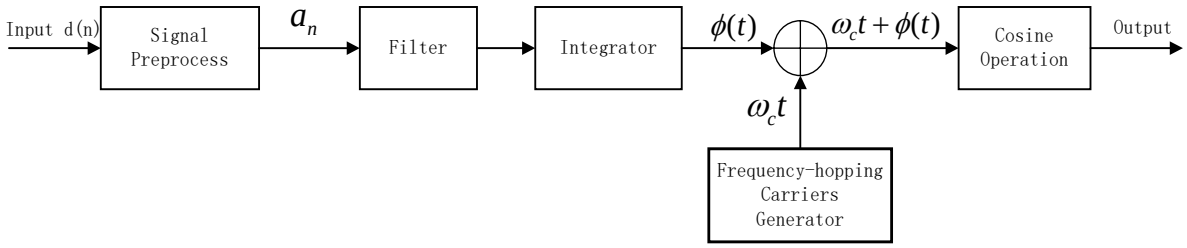

Figure 1: The schematic diagram of GMSK frequency hopping modulation

The working process is that the user inputs the code elements of the sequence combination of {0, 1}, then the code elements is preprocessed to obtain the bipolar non-return to zero code a_n, next the preprocessed a_n is filtered by the Gaussian low-pass filter, and through the integrator obtains the accumulated phase $\phi(t)$ of the waveform. Then sums the accumulated phase and the phase $\omega_c(t)$ generated by the frequency hopping carrier to obtain a total phase, and the total phase calculates the cosine values through the CORDIC operator, thereby the GMSK modulation of the input signal can be realized.

4. Implementation of GMSK Frequency Hopping Modulation on FPGA

According to the Figure 1, the GMSK frequency hopping modulation can be divided into three parts: the frequency hopping module for generating the frequency hopping carrier phase, the phase accumulator for calculating the accumulated phase, and the CORDIC operator for phase cosine values, so the design of this scheme mainly discusses the FPGA implementation from these three parts. The FPGA implementation principle of GMSK frequency hopping modulation is shown in Figure 2.

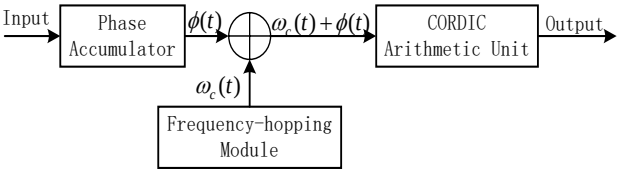

Figure 2: The FPGA implementation principle of GMSK frequency hopping modulation

4.1 Design of Frequency Hopping Module

The frequency hopping module is composed of a PN code generator and a DDS frequency synthesizer. The PN code generator can generate pseudo-random hopping code elements, and different code elements are input to the DDS frequency synthesizer as different frequency control words, controlling the DDS exports the phase information of the sine wave, the cosine wave and the phase information of waveform. The principle of the frequency hopping module is shown in Figure 3.

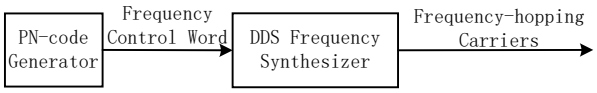

Figure 3: The principle of the frequency hopping module

The PN code is generated by the m-sequence, and the primitive polynomial corresponding to m-sequence is set to $f(x) = x^4 + x + 1$. This is an m-sequence of order 4 and period of 15, so that 15 different pseudo-random codes can be generated, corresponding to 15 hopping points. The initial state of m-sequence is set to 0001, and the 15 pseudo-random codes generated according to the law of the m-sequence are {0001, 1000, 0100, 0010, 1001, 1100, 0110, 1011, 0101, 1010, 1101, 1110, 1111, 0111, 0011}, eventually jump back to 0001, 15 pseudo-random codes are 15 hopping points, cyclic jump according to this rule.

When the frequency hopping carrier is generated by pseudo random codes, the output waveform frequency f_{out} of the DDS frequency synthesizer and the input system clock f_{clk}, the frequency control word K, and the data bit width *N* of the phase accumulator in the DDS have the relationship of the following equation (9) [6]:

$$f_{out} = \frac{f_{clk} \times K}{2^N} \quad (9)$$

The frequency resolution Δf of DDS is:

$$\Delta f = \frac{f_{clk}}{2^N} \quad (10)$$

The system index is 50MHz for intermediate frequency, 20MHz for bandwidth, and 400MHz for DDS input clock. In order to make the frequency hopping points fill the bandwidth as much as possible and meet the requirements of system specifications, the data bit width N of the phase accumulator is designed to 8 bits, and the frequency control word K is set to [25, 39], input the generated pseudo-random codes as the frequency control word K, 15 frequency control words corresponding to the DDS to generate 15 different frequency carriers, according to the formula (10), the frequency hopping interval, that is Δf of the DDS is 1.5625MHz, so the 15 hopping points generated are 39.0625M, 40.625M, 42.1875M, 43.75M, 45.3125M, 46.875M, 48.4375M, 50M, 51.5625M, 53.125M, 54.6875M, 56.25M, 57.8125. M, 59.375M, 60.9375M, 15 frequency hopping points generate frequency hopping carrier according to the law of pseudo random codes.

4.2 Design of Phase Accumulator Module

The code elements input by the user terminal can be processed by phase accumulator to obtain the accumulated phase $\phi(t)$ of the signal. The phase accumulation process is mainly to perform integral operations of the input signal waveform.

The scheme sets the code element rate to 1 Mbps, and the sampling rate takes 50 times of the code element rate, that is 50 Mbps, which means that the signal is sampled 50 times in one symbol period. The input of the phase accumulator is the code elements of {0, 1}, and the calculation process of accumulated phase is shown in Figure 4.

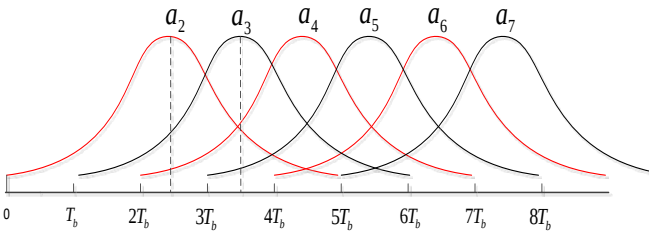

Figure 4: Calculation process of the accumulated phase

Since the ideal Gaussian impulse response *g(t)* is infinitely long, it is unrealizable in physics, and the energy of *g(t)* is concentrated only in a limited interval, so we can truncate it in actual calculation [7], according to the waveform characteristic intercepts the length of $5T_b$, so the truncated $g_T(t)$ satisfies the following formula:

$$\int_{-\infty}^{\infty} g_T(t)dt = T_b \qquad (11)$$

The curve in Figure 4 is the Gaussian impulse response of the rectangular pulse corresponding to the code elements a_2、a_3.....a_7. The truncation of $5T_b$ is expressed as $g_T(t-nT_d-T_b/2)$. By accumulative phase $\phi(t)$ expressed in the equation (7), the computational process of accumulative phase is as follows:

$$\phi(t)=\frac{\pi}{2T_b}\sum_{n=0}^{k-1}a_n\int_{-\infty}^{(k-1)T_b}g_T(\tau-nT_b-\frac{T_b}{2})d\tau+\frac{\pi}{2T_b}a_k\int_{(k-1)T_b}^{t}g_T(\tau-kT_b-\frac{T_b}{2})d\tau \qquad (12)$$

When $0\le t\le T_b$,

$$\phi(t)=\frac{\pi}{2T_d}a_2\int_0^t g_T(\tau-2T_b-\frac{T_b}{2})d\tau \qquad (13)$$

When $T_b\le t\le 2T_b$,

$$\phi(t)=\frac{\pi}{2T_d}a_2\int_0^t g_T(\tau-2T_b-\frac{T_b}{2})d\tau+\frac{\pi}{2T_d}a_3\int_{T_b}^t g_T(\tau-3T_b-\frac{T_b}{2})d\tau \qquad (14)$$

When $2T_b\le t\le 3T_b$,

$$\phi(t)=\frac{\pi}{2T_d}a_2\int_0^t g_T(\tau-2T_b-\frac{T_b}{2})d\tau+\frac{\pi}{2T_d}a_3\int_{T_b}^t g_T(\tau-3T_b-\frac{T_b}{2})d\tau+\frac{\pi}{2T_d}a_4\int_{2T_b}^t g_T(\tau-4T_b-\frac{T_b}{2})d\tau \qquad (15)$$

And so on, because $\int g(t)dt=T_b$, so when $nT_b\le t\le(n+1)T_b$,

$$\begin{aligned}\phi(t)&=\frac{\pi}{2T_d}a_2\int_0^t g_T(\tau-2T_b-\frac{T_b}{2})d\tau+\cdots+\frac{\pi}{2T_d}a_n\int_{nT_b}^t g_T(\tau-nT_b-\frac{T_b}{2})d\tau\\&=\frac{\pi}{2}\sum_2^{k-2}a_n+\cdots+\frac{\pi}{2T_d}a_{n+1}\int_{(n-2)T_b}^t g_T(\tau-(n+1)T_b-\frac{T_b}{2})d\tau+\frac{\pi}{2T_d}a_{n+2}\int_{nT_b}^t g_T(\tau-(n+2)T_b-\frac{T_b}{2})d\tau\end{aligned} \qquad (16)$$

It can be obtained from the above calculation process that the accumulative phase is divided into two parts, the instantaneous phase generated by the last five code elements and the accumulative phase of each symbol input before the last five symbols, the value of the accumulative phase is finally constant to $\pm\pi/2$. Every time you enter a new code element, the calculation is performed according to the above calculation process. Except for the last five code elements entered, the integral value corresponding to the previous code elements is $\pi/2$, to calculate the integral, simply multiply the corresponding symbol value by $\pi/2$ and then sum. For the last five code elements, the sampling frequency of 50MHz corresponds to 50 sampling points for each symbol, and a $5T_b$ truncated Gaussian impulse response corresponds to 250 sampling points, so 250 integral values can be generated, and the integral value is calculated as shown in the following equation (17).

$$J\_g(t)=\frac{1}{2T_b}\int_0^t g_T(\tau)d\tau \qquad (17)$$

In MATLAB, evenly take 250 points to calculate the integral value, and store it in the register of FPGA for calling. When there is code element input, the code element is processed into NRZ code first, according to calculation formula of the accumulative phase (16), the last five NRZ codes as a group multiply the Gaussian integral values in the register then added to calculate the phase value, plus the accumulation of phase values of the code elements before the last five code elements, the sum of accumulated phase can be obtained, this value is the scale value of π , which is $1/\pi$ of the actual phase value.

4.3 Design of CORDIC Arithmetic Unit

CORDIC is a method of numerical approximation by a finite number of iterative operations. It uses a series of yaws of angle associated with the operation base to approximate the angle of the rotation required [8]. It performs a plane rotation, the vector (X_i,Y_i) is transformed into the vector (X_j,Y_j) by the rotation The angle of rotation θ can be obtained by a finite number of iterative calculations, and each iteration is as follows:

$$\begin{bmatrix}X_{n+1}\\Y_{n+1}\end{bmatrix}=\begin{bmatrix}\cos\theta_n & -\sin\theta_n\\ \sin\theta_n & \cos\theta_n\end{bmatrix}\begin{bmatrix}X_n\\Y_n\end{bmatrix} \qquad (18)$$

When calculating the sine and cosine of the angle by the CORDIC algorithm, the initial vector (x_0, y_0) is set to (1, 0), and the sine and cosine values of the phase angle can be obtained by using the CORDIC algorithm to find the vector (x_n, y_n) with a rotation angle of θ.

In the FPGA implementation of this scheme, since the input angle of rotation exceeds the operation range of $[-\pi, \pi]$ of the CORDIC algorithm, so a CORDIC arithmetic unit structure with extended phase operation range is designed, which can perform CORDIC operation at any angle without being limited by the quadrant, the structure of this arithmetic unit is shown in Figure 5.

Figure 5: CORDIC arithmetic unit structure with extended phase operation

The pre-processor inputs a phase signal with a scale of π. According to the periodicity of the trigonometric function, the value is first converted to the value in the interval [0, 2] by 2 cycles, and then the first phase of the binary phase is detected. If it is 0 means that in [0, 1] interval, that is the first and second quadrants. If it is 1 means that in [1, 2] interval, that is in the three or four quadrants. Signing which the quadrant is located, then send the data to CORDIC core processor to calculate the cosine value, and then the result is processed according to the flag bit to the corresponding quadrant to obtain the final cosine value.

5. Simulation and results

The modulation scheme is simulated in MATLAB, taking the intermediate frequency 50MHz, the bandwidth is 20MHz, the frequency hopping points are 15. Select 20 input symbols d = [1 1 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0], and the simulation result of the phase accumulator module is shown in Figure 6. In the figure, when the input symbol is 1, the output phase is increased, and when the input symbol is 0, the output phase is decreased, we can conclude that the phase information corresponding to the input symbol is generated by the operation of the phase accumulator module.

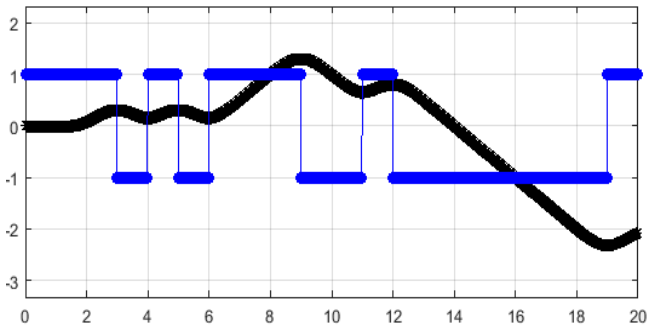

Figure 6: MATLAB simulation diagram of phase accumulator module

The overall simulation result of the system is shown in Figure 7. It can be seen that the modulation of the

present scheme generates a GMSK modulated signal of carrier hopping corresponding to the input code elements.

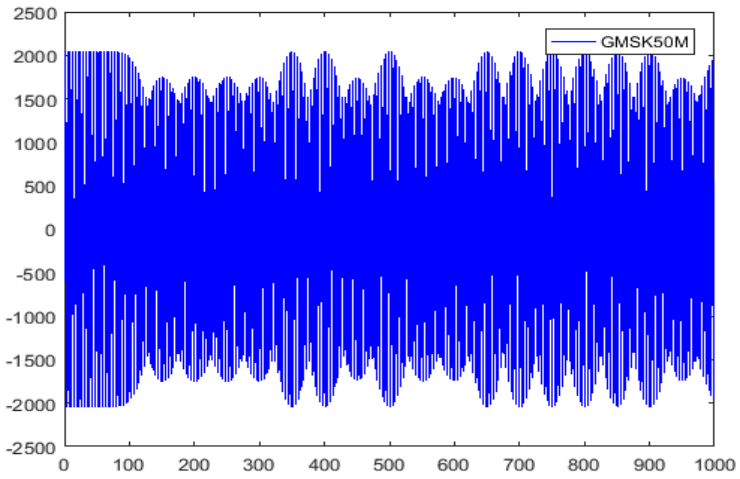

Figure 7: MATLAB simulation diagram of GMSK modulation

To further verify the correctness of the overall scheme, the design is implemented on the ISE platform of the FPGA, then the whole system is simulated and verified by Modelsim software. The result is shown in Figure 8 below. The input data din is the code elements sequence of {0, 1} combination. It can be seen that the value of the phase gmsk_phase output by the phase accumulator is continuous, and when the input data din is 1, the value of gmsk_phase increases, when din is 0, the value of gmsk_phase decreases. It is consistent with the waveform of the phase in the MATLAB simulation, which proves the correctness of the FPGA implementation of the scheme. Phase_out is the phase information of the frequency hopping carrier generated by the frequency hopping module. The gmsk_phase of the output of the phase accumulator is added to the phase hop signal phase_out of the frequency hopping generated by the frequency hopping module to obtain the phase sum_phase, and then the CORDIC operation is performed to get the cosine value, and the GMSK modulation signal gmsk is generated, finally implements GMSK modulation of carrier frequency hopping for the input data.

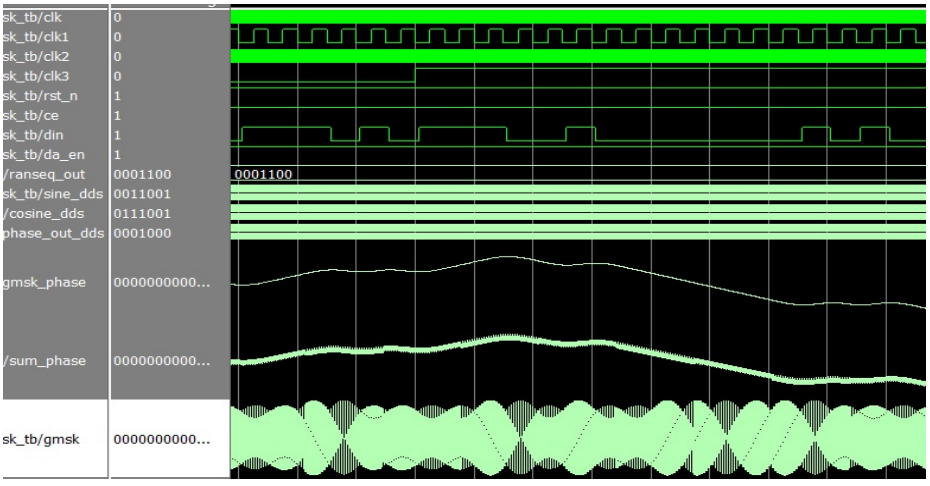

Figure 8: Simulation diagram of GMSK modulation

The above results show that the GMSK frequency hopping modulation method proposed in this paper realizes the frequency hopping function of GMSK carrier, ensures the phase continuity of modulated signal, and saves a lot of complicated multiplication operations, which makes the realization of frequency hopping communication system simpler.

6. Conclusion

In this paper, an FPGA-based GMSK frequency hopping modulation scheme is designed. The frequency hopping module is added to the GMSK modulation system to realize the control of frequency hopping. The design of phase accumulator and CORDIC algorithm simplify the modulation scheme, which not only ensures the continuity of the phase but also saves a lot of multiplication operations,it also improves the feasibility of the scheme and saves a lot of storage space of the system. The scheme is implemented in the Verilog HDL language on the Xilinx ISE software and simulated on Modelsim software. The simulation results show the correctness and feasibility of this scheme.

References

[1] Wu X, Shi J H. Anti-Interference Performance Analysis in Bluetooth Frequency Hopping System[C]. IEEE International Workshop on Anti-Counterfeiting, Security, Identification. IEEE, 2007: 328-331.

[2] Popovski P, Yomo H, Prasad R. Dynamic adaptive frequency hopping for mutually interfering wireless personal area networks[J]. IEEE Transactions on Mobile Computing, 2006, 5(8): 991-1003.

[3] Iqbal R, Akhtar M W, Junaid M, et al. A design framework for frequency hopped communication system[C]. Proceedings of 2016 13th International Bhurban Conference on Applied Sciences and Technology, IBCAST, 2016, 9: 699-703.

[4] De-Xin L I, Gao X J, Zhuang Z. Design for GMSK frequency-hopping communication system based on Simulink[J]. Journal of Jilin University (Information Science Edition), 2007, 25(4): 391-7.

[5] Babu K M N, Vinaymurthi K K. GMSK modulator for GSM system, an economical implementation on FPGA[C]. International Conference on Communications and Signal Processing. IEEE, 2011: 208-212.

[6] Huang C, Ren L X, Mao E K, et al. A systematic frequency planning method in Direct Digital Synthesizer (DDS) design[C]. International Conference on Wireless Communications and Signal Processing. IEEE, 2009: 1375-1378.

[7] Peng W J, Song W T, Luo H W. Application and performance analysis of GMSK modulation in frequency-hopping communication[J]. Journal of China Institute of Communications, 2000, 21(11): 41-47.

[8] Liu X Y, Liang X M. Theory and Implementation of GMSK Tamed Spread Spectrum Modulation Technology[J]. China Academy of Electronics and Information Technology, China, 2010, 5(4): 415-418.