Model-Based Design and Implementation for Spread Spectrum Communication System

Xiongfei Li [1,2,*], Pengfei Xu [2], Xiaoling Lai [1,2], Shangcheng Li [2], Guochang Zhou [2], and Yangming Guo [1]

[1] School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China
[2] China Academy of Space Technology (Xi'an), Xi'an, China
Email: lxf_li@163.com (X.L.); pfxu0913@163.com (P.X.); 20589149@qq.com (X.L.); 526130675@qq.com (S.L.); zhouguochang2000@163.com (G.Z.); yangming_g@nwpu.edu.cn (Y.G.)
*Corresponding author

***Abstract*—The Model-Based Design (MBD) method was proposed to optimize the conventional software development. The MBD-based software development with SIMULINK tool was introduced to generate code automatically for the Field-Programmable Gate Array (FPGA). Based on spread spectrum communication system, re-development employing MBD was achieved and compared with the manual development. The MBD's validity and feasibility were verified by evaluating the performance of the automatically generated code.**

***Keywords*—software engineering, Model-Based Design (MBD), spread spectrum communication system, Field-Programmable Gate Array (FPGA)**

I. Introduction

For its strong anti-interference performance, outstanding concealing characteristics, and low interception probability, spread spectrum has become a major anti-jamming method in anti-jamming communication [1], especially in satellite communication [2]. Due to the high complexity of the system and limited hardware resources on satellites, high demand was put forward for on-orbit software design and implementation. The Model-Based Design (MBD)-based software development technology have been widely used. And it will become the mainstream of space-borne software development for higher quality, shorter development cycle and lower development cost.

The document driven development mode was generally adopted by traditional software development. After the requirements were brought up from the headquarter, the design, implementation, testing, and verification was accomplished gradually by the professional engineers. The disadvantages of this mode were obvious. Firstly, lots of time was wasted on code development; Secondly, high cost of error correction and the development cycle had to be prolonged; Finally, changes of algorithm requirements or design architecture made re-coding and timing adjustments which were inevitable [3, 4].

The prototype in the early design stage was fully verified in the MBD development method. Meanwhile, errors caused by human in coding stage were effectively avoided and repetitive work in later stages was massively reduced. MBD has been widely deployed in industries like automobile and industrial control for embedded software research and development as an important method to shorten the software development cycle and reduce software costs [5–8].

As the complexity of satellite spread spectrum communication system increases, Field-Programmable Gate Array (FPGA) becomes the main processors carrying out digital signal processing functions. The MBD-based FPGA software development is a new method and still in the early stage. Exploration of MBD-based FPGA design is essential to solve the current software development problems of spread spectrum communication and can meet the needs in the future.

The paper is organized as follows: Section II describes composition of spread spectrum communication systems. Section III presents the design process of MBD-based FPGA software. Section IV illustrates design and implementation of MBD-based spread spectrum communication system. Section V summarizes the results.

II. Composition of Spread Spectrum Communication System

The basic principle of Direct Sequence Spread Spectrum (DSSS) was shown in Fig. 1. At the transmitter, the spread spectrum signal *d(t)* was obtained by multiplying information source *a(t)* with spread spectrum sequence *c(t)*. Then *d(t)* was modulated by the carrier fc and the transmitted signal *s(t)* was generated. At the receiver, the received signal with noise was mixed by local carrier fi and then filtered to obtain *r(t)*. Then *r(t)* was despreaded and demodulated to get $r_{cor}(t)$. Finally, the original information $a'(t)$ was recovered.

Set the symbol interval of a(t) as T_a , *a(t)* was expressed as Eq. (1):

$$a(t) = \sum_{n=0}^{\infty} a_n g(t - nT_a) . \qquad (1)$$

Manuscript received April 20, 2023; revised May 23, 2023; accepted June 7, 2023; published January 25, 2024.

doi: 10.12720/jait.15.1.127-131

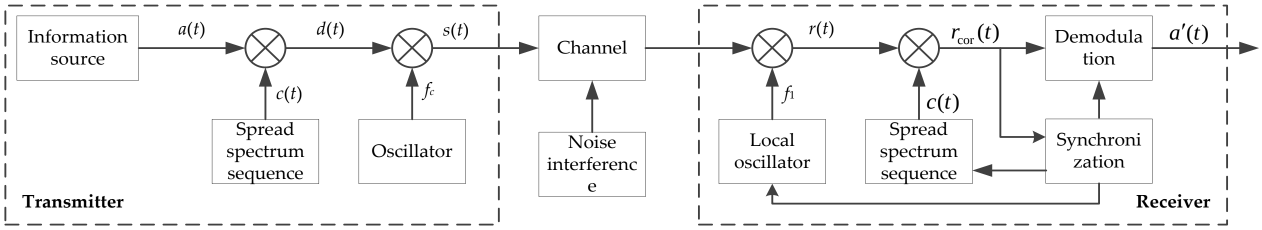


Fig. 1. Functional block diagram of spread spectrum communication system.

where $g(t)=\begin{cases}1, 0 \le t \le T_a \\ 0, others\end{cases}$ denoted the threshold function, a_n represented the original information flow, $a_n \in \{+1,-1\}$.

Given the chip cycle of Pseudo-Random (PN) sequence *c(t)* was T_c, c(t) was expressed as Eq. (2):

$$c(t)=\sum_{n=0}^{N} c_n g(t-nT_c)\,. \qquad (2)$$

The spread spectrum process was regarded as a modulo-2 addition operation between the sequence *a(t)* and the spread spectrum sequence *c(t)*, then the spread spectrum signal *d(t)* was obtained as Eq. (3):

$$d(t)=\mathrm{a}(t)c(t)=\sum_{n=0}^{\infty} d_n g(t-nT_c)\,. \qquad (3)$$

where $d_n=\begin{cases}1, & a_n=c_n \\ -1, & a_n \neq c_n\end{cases}$. Before transmission, *d(t)* was modulated: $s(t)=d(t)\cos\omega_c t$.

In despreading process, most of interferences were uncorrelated to the local spread spectrum sequence. After despreading, most of interferences were filtered out. Thus, the purpose of suppressing interference power was achieved. The output Signal-to-Noise Ratio (SNR) was improved, and the anti-interference performance of the system was enhanced.

III. Design Process of MBD-Based FPGA

Based on graphical and visualized design and emulation with ease, MBD implementation was one of the best methods for algorithm design. While the nonnegligible problem was the available functions were limited, especially for the interface between FPGA and its peripheral hardware, which was relied on device characteristics and difficult for MBD-based development to achieve. Therefore, during MBD-based FPGA development, the overall implementation architecture was divided into MBD and non-MBD implementation in the first step. The code generated by the two methods was merged after finishing coding, software and hardware integrated debug was taken place at last. The specific process was shown in Fig. 2.

The following principles were observed when the architecture was divided into MBD and non-MBD. Numerical or logical calculations like digital signal processing algorithms were implemented in the MBD mode. Non numerical calculations and hardware related functions were implemented in the non MBD mode, such as output signal management, etc.

In this paper, the MATLAB/SIMULINK tool was adopted for MBD-based code generation. After requirements analysis, the detailed design for the algorithm model was carried out, verification was later accomplished by simulation and testing. Finally, the code was generated automatically and tested on hardware [9].

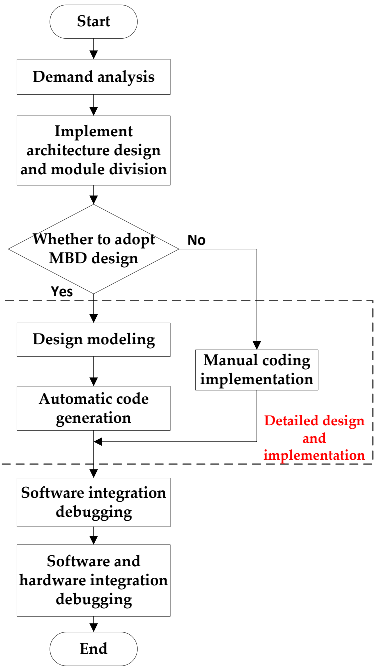

Start
Demand analysis
Implement architecture design and module division
Whether to adopt MBD design
No
Yes
Design modeling
Manual coding implementation
Automatic code generation
Detailed design and implementation
Software integration debugging
Software and hardware integration debugging
End

Fig. 2. Development process of FPGA design based MBD.

A. *Demand Analysis*

An executable model for functional requirements, which corresponded to the paper requirements item by item, was established in the beginning. According to the technical requirements, the functional modules were preliminarily decomposed into subsystems, inputs and outputs of subsystems and handshake signals among subsystems. And the realization procedure of core function for subsystems was built.

B. *Model Design*

In the design and modeling stage, the model was refined step by step, and the functional correctness of each functional unit was verified by simulation, including:

- The subsystem functions refinement, the subsystem test bench model establishment and the subsystem functions test and results exportation.
- Integrated system model test, inputs and outputs of subsystems confirmation, handshake signals

among subsystems processing, system test model establishment, system test execution, result analysis and results exportation.

In the specific modeling process, hardware awareness including clock frequency, data type, resource mapping was necessary. Meanwhile, in order to generate Hardware Description Language (HDL) for all models in the later stage, the models used were required to be included in the model library that could conduct HDL generation. For each level of functional modules, the data types of both inputs and outputs were specified (including symbol bit, integer bit, decimal bit) based on the performance requirements. The fixed-point simulation was accomplished after data type specification for the function and performance confirmation.

C. Hardware Description Language (HDL) Code Generation

The code generation tool HDL Workflow Advisor of SIMULINK was applied to generate HDL code automatically from the model in Model-based automatic code generation [9–14], which consisted of three steps:

1) Set target hardware

Set the generation mode, configuration template, comprehensive tools, chip model and project path. In addition, it was necessary to set the working clock frequency. And the corresponding constraint file would be generated after the code was finally generated.

2) Model static check

Check the rules of the model to meet essential conditions for Very High-Speed Integrated Circuit Hardware Description Language (VHDL) code generation. Errors for corresponding problems would be reported once the conditions failed to meet the rules. Re-checking the global settings and the sampling time was generally needed.

3) HDL code generation

Finish the code generation from the model to the HDL code after setting a series of optimization options, including basic option settings, resource and speed optimization settings and clock port settings, etc. Testbench files for the combined simulation of Simulink and third-party simulation software like MODELSIM were generated in this step too.

D. Integrated Debugging

If the current model corresponded to a complete FPGA project, the whole Integrated Synthesis Environment (ISE) project file was generated by the HDL Workflow Advisor tool. And the synthetization, mapping and routing was finished step by step.

Once the current model was only a part of FPGA project, automatically generated code by the HDL Workflow Advisor tool was combined with manually developed code to build a whole FPGA project. Then the whole project was jointly tested on hardware.

IV. Design and Implementation of MBD-Based Spread Spectrum Communication System

In order to verify the performance of MBD-based FPGA software development, a spread spectrum communication system was re-developed after manual development.

The original information was conducted Cyclic Redundancy Check (CRC) encoding, convolutional coding, interleaving, framing, spectrum spreading and modulation before transmission. The synchronization header in the frame was used for capturing and synchronization at the receiver, and the unique word was adopted for frame synchronization. Synchronous acquisition was one of the key techniques in spread spectrum communication system. Rapid synchronization of carrier and phase within a limited time under certain SNR was achieved at the receiver.

The function of receiver and transmitter was realized on a single FPGA, the core functions including spread spectrum processing algorithm were implemented in MBD mode, and the peripheral device (AD/DA) interface controlling was developed in non-MBD mode.

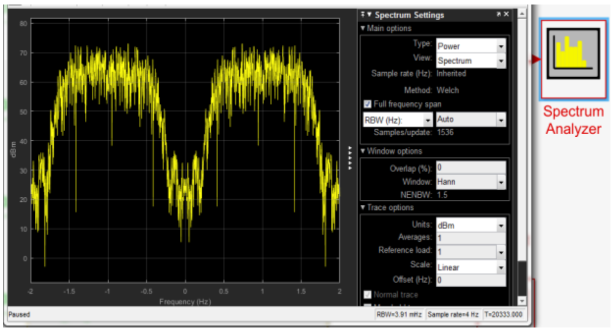
According to the development procedures mentioned above, the MBD development procedure included demand modeling in requirements analysis phase, algorithm modeling in design phase, automatic code generation, joint testing and verification of software and hardware in implementation phase. Finally, the performance of the automatically generated code was evaluated.

A. Demand Modeling

In requirements analysis phase, a top-level model was established, subsystems at all levels were defined, interfaces among subsystems were given. Then preliminary algorithm simulation and verification was carried out.

The FPGA consisted of two core processing functions. Namely, the transmitter and receiver. In order to verify the performance of the transmitter and receiver, the following top-level model was established and divided into three subsystems:

- Data source generation subsystem: the upper layer service data generation and the signal interface with the physical layer simulation.
- Spread spectrum transmitting subsystem: encoding, modulation, spectrum spreading and up converter of the signal.
- Spread spectrum receiving subsystem: demodulation of IF sampled signal, including capturing, tracking, decoding and the original information recovery.



Spectrum Settings
Main options
Type: Power
View: Spectrum
Sample rate (Hz): Inherited
Method: Welch
☑ Full frequency span
RBW (Hz): Auto
Samples/update: 1536
Window options
Overlap (%): 0
Window: Hann
NENBW: 1.5
Trace options
Units: dBm
Averages: 1
Reference load: 1
Scale: Linear
Offset (Hz): 0
☑ Normal trace
Paused RBW=3.91 mHz Sample rate=4 Hz T=20333.000

Fig. 3. Domain simulation results.

In the model, the baseband spread spectrum signal and frequency domain simulation results were shown in Fig. 3.

B. *Algorithm Modeling*

In the design phase, the model in the requirement analysis phase need to be further refined. Considering automatic code generation, all models were developed in HDL Coder library, and data types of the interfaces were set to fixed-point.

1) *Data source generation subsystem*

The interface types between the upper layer were simulated in data source generation subsystem. The test data was also generated like Fig. 4.

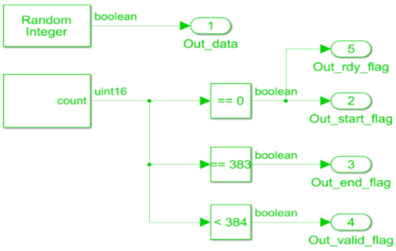


Fig. 4. Data source generation model.

2) *Spread spectrum transmitting subsystem*

The spread spectrum transmitting subsystem mainly contained CRC coding, convolutional coding, framing, data spreading, baseband filtering and digital up converter. The modeling of the sending part was established.

3) *Spread spectrum receiving subsystem*

Spread spectrum receiving process model was mainly consisted by the digital down converter, matched filtering, acquisition module, tracking module, Viterbi decoding module, CRC module and frame decoding module.

Acquisiton played an essential role at the receiver in spread spectrum system. The autocorrelation characteristic of PN code was adopted to determine the code phase of the received signal. Moreover, due to the existence of frequency offset, the correlation peak value would be shortened. Two-dimensional search of carrier frequency offset and spread spectrum phase which was conducted to determine the initial frequency offset and code phase at the receiver was generally required and strongly recommended.

Surmising that the received signal sequence was *r(n)* and the local spread spectrum sequence was *c(n)*, The correlation between *r(n)* and *c(n)* was given by:

$$r_{cor}(n)=\sum_{m=0}^{L\text{-}1} r(m)\bullet c(m\text{-}n)=r(n)\otimes c(\text{-}n)\ . \qquad (4)$$

In Eq. (4), *L* denoted the period length of the spread spectrum sequence; ⊗ represented convolution computation. Since convolution in time domain was equal to multiplication in frequency domain, so Eq. (4) was transformed into:

$$r_{cor}(n)=r(n)\otimes c(\text{-}n)=IFFT[R(K)\bullet C^{*}(K)]\ . \quad (5)$$

where *R(K)* was result of Fast-Fourier Transformation (FFT) of *r(n)*. $C^{*}(K)$ denoted the conjugated results of FFT of *c(n)*. Then the PN code phase of the received signal was obtained by searching the maximum value of r_{cor}(n) [13], Inverse Fast Fourier Transform (IFFT) [14] was the inversed Fast Fourier Transform (FFT).

On this basis, a reasonable searching range and interval for frequency offset was given, and the parallel searching algorithm was adopted, which was shown in Fig. 5.

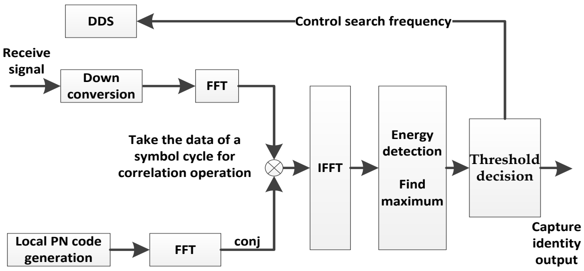


Fig. 5. Spread spectrum receiving model.

C. *Code Generation and Test Verification*

After accomplishing the model simulation, the HDL code generation steps were coming. The automatically generated code of the model possessed the characteristics of conciseness, clearness, legibility, and consistence.

Merging the generated HDL code into a project, then bit file was compiled and generated. After compiling, joint testing of software and hardware was conducted on hardware, the results were shown Fig. 6.

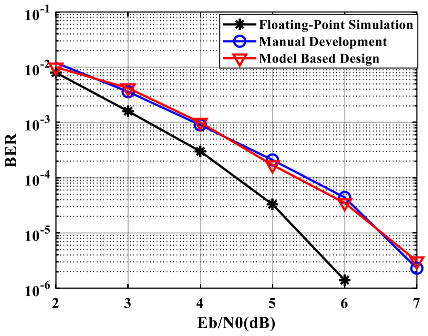


Fig. 6. Bit Error Ratio (BER) performance comparison.

The theoretical performance was obtained by the floating-point simulation with the system parameters (Quadrature Phase Shift Keying (QPSK) and the convolutional code, the communication rate is 32 kbps and the chip rate is 4 Mcps). The function and performance of MBD-based spread spectrum communication system fully meet the specification. the MBD-based code achieved the same BER performance as the manual developed one without extra SNR loss. The practical implementation loss due to fixed-point is about 1 dB.

D. *Code Performance Comparison*

The spread spectrum communication system developed above was categorized as a large-scale FPGA with a scale of 40,000 lines in code amount and nearly ten million gates in resource. The evaluation of the automatic code generation was crucial and indispensable for large-scale software. The performance comparison over key parameters between automatically generated code and manually developed code was detailed in Table I.

TABLE I. PERFORMANCE PARAMETER

Parameters	Model Based Development	Manual Development
Max Freq	201.6 MHz	209.6 MHz
Slice LUTs	35%	30%
Slice Registers	38%	33%
BlockRAM	37%	62%
DSP48E	52%	62%

As shown in Fig. 7, given the manually optimize code performance was unit 1, the performance of the two methods was compared in terms of the maximum operating frequency and different resources occupation.

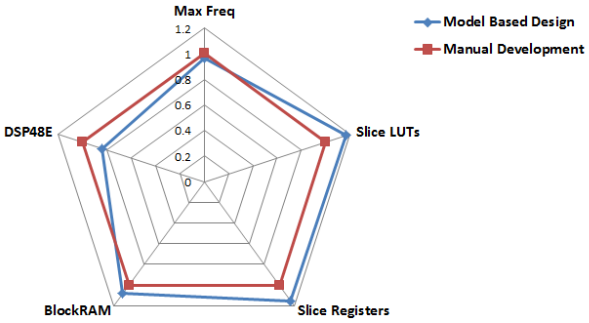


Fig. 7. HDL code performance comparison.

For the maximum operating frequency, the MBD-based code exceeds 200 MHz, and only 3% lower than manual development. Considering the resource expenses, including Slice LUT, Slice Registers, DSP48E and Block RAM, the resource costs of the two methods were almost equal. The two developments were conducted by different developers with different implementation architectures of individual modules, which lead to nuance of resource expense.

Furthermore, as the deeper understanding of the modeling characteristics and methods, the model would be optimized, the enhanced performance of the automatic code generation could be achieved subsequently.

V. CONCLUSION

Through the verification of the large-scale FPGA project of the spread spectrum communication system, the MBD-based FPGA development provides a brand-new but efficient method for FPGA design to automatically generate code. The code based on MBD achieves the same demodulation loss of 1dB in SNR and the similar resource expense as the manually developed code. The MBD method is suitable for space-borne FPGA development by multiple dimensional verifications.

The MBD-based FPGA development provides an excellent method for developers. With deeper understanding of the relationship between models and codes, the MBD-based FPGA development will become the mainstream of space-borne FPGA product development in the future.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Xiongfei Li and Pengfei Xu conducted the research; Xiaoling Lai and Guochang Zhou analyzed the data; Xiongfei Li and Shangcheng Li wrote the paper; Yangming Guo proofread the paper; All authors had approved the final version.

FUNDING

This research was funded by the Fundamental Research Funds for the Central Universities of China. Grant number is D5000220351.

REFERENCES

[1] Z. Ma, T. Chen, M. Zhang, P. D. Kecerski, and S. Dang, “Literature review of spread spectrum signaling: performance, applications and implementation,” *Journal of Communications*, vol. 10, no. 12, pp. 932–938, 2015.

[2] P. Kim, S. Jung, and D.-H. Jung, “Performance analysis of direct sequence spread spectrum aloha for LEO satellite based IoT service,” in *Proc. IEEE 90th Vehicular Technology Conference*, 2019.

[3] Y. L. Lei, Y. F. Zhu, and Y. J. Tan, “Model driven process modeling and simulation of complex man-machine systems,” *Systems Engineering and Electronics*, vol. 1, pp. 223–231, 2016.

[4] F. R. Liu, “Model-based developing approach for airborne software of commercial engine,” *Process Automation Instrumentation*, vol. 6, pp. 26–30, 2017.

[5] C. Feng, G. Li, and G. Zhao, “Research on model-based design method power measurement,” *Electrical Measurement & Instrumentation*, vol. 19, pp. 1–7, 2014.

[6] T. Keleş, S. B. Ö. Yalçın, and Y. K. Alp, “Model based design of software defined and cognitive radio and implementation on an FPGA,” in *Proc. the 29th Signal Processing and Communications Applications Conference (SIU)*, 2021.

[7] R. Muzammil, “Design and implementation of 4-QAM transceiver on FPGA using MATLAB/SIMULINK model,” in *Proc. the International Conference for Advancement in Technology (ICONAT)*, 2023.

[8] D. S. Kaputa and K. A. Derhak, “Model based design of a real time FPGA-based lens undistortion and image rectification algorithm for stereo imaging,” *IEEE Access*, vol. 11, pp. 18972–18982, 2023.

[9] T. Keishi, S. Mina, M. Yasushi, and O. Hitoshi, “Applying Model-Based Development (MBD) to embedded systems,” in *Proc. the 55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference*, 2014.

[10] C. Xin, “Model-based design for software defined radio on an FPGA,” *IEEE Access*, vol. 6, pp. 8276–8283, 2017.

[11] S. L. C. Titri and K. Y. Toumi, “Rapid prototyping of PVS into FPGA: From model based design to FPGA/ASICs implementation,” *Design & Test Symposium*, vol. 9, pp. 162–167, 2014.

[12] G. C. Du, H. H. Sun, and Y. S. Yang, “Design and implementation of frequency hopping communication system based on simulink HDL coder,” *Radio Communications Technology*, vol. 42, no. 5, pp. 96–98, 2016.

[13] F. Tao, J. W. Wang, and M. X. Tian, “Improvement of FFT acquisition algorithm in spread spectrum communication system,” *Radio Engineering*, vol. 46, no. 5, pp. 41–44, 2016.

[14] P. Tulpule, A. Rezaeian, A. Karumanchi, and M. S. Midlam, “Model Based Design (MBD) and Hardware in the Loop (HIL) validation: Curriculum development,” in *Proc. the 2017 American Control Conference*, 2017.