Technical research of real-time simulation system interface expansion of airborne utility system management

Zhe Liang¹, Jun Zhao¹, Jian Sheng Wang¹ and Jia Wei Han¹∗

¹Xi’an Aviation Institute of Computing Technology Aviation Industry Corporation of China, Xi’an, Shaanxi, 710000, China

∗Corresponding author’s e-mail: hanjw0209@163.com

Abstract. The hardware in the loop is a necessary link during the development of the complex system based on V&V processes. The airborne utility system is featured by numerous types of interfaces, large quantities, and complicated interconnection. During the hardware in the loop test, it is impossible to effectively carry out a large numbers of synchronous simulation, due to the restriction of the simulation equipment resource. This paper proposes an FPGA-based interface expansion technology, and the core idea is to reuse and recombine the signals. The specific steps are as follows: implement the signal reuse algorithm by the FPGA in the interface module, complete the restoration of the recombinered signals according to the synchronization signal and the address signal of the system and transmit the control interface command to the device under test. The simulation results show that the real-time simulation system interface expansion technology proposed in this paper can realize the synchronous simulation function of large-scale signal with a small number of simulation channels, and effectively increase the number of interfaces in the real-time simulation system.

1. Introduction

The airborne utility system is a complex system which integrates mechanical, electrical, hydraulic, and pneumatic all together and it belongs to coupling system with multi-energy domains [1][2]. At present, the aircraft utility system is usually controlled and managed comprehensively to form the utility management system with the computer as the core. Based on the utility integration technology, the computer technology such as multiprocessor plus data bus is used to systematically and comprehensively manage the utility system.

With the constant improvement of the function and performance of the aircraft, the development complexity and cycle of the UMS (Utility Management System) are greatly increased so in order to reduce development risks and improve design efficiency, it is necessary to fully study the influence of various factors on the function and performance of the system in the early stage of development. Based on the V&V development processes, the full-mode modeling and simulation of the system are implemented and the design constraints of the airborne utility management system are established with the simulation technology, forming the verification method. The hardware in the loop interconnects the utility management computer with the utility system object and operates the simulation model of the utility system in the real-time processor to simulate the operation state of the complete utility system and implement the real-time simulation test of the utility management computer.

However, due to the characteristics of the airborne utility system signal such as large quantity and various types, the utility management computer simulation devices are often restricted by the hardware resource and due to the small quantity of simulation channels and difficulty in implementation of
signal synchronization simulation in large quantity, it is impossible to meet the demand of synchronous simulation of utility management computer interface signals.

Taking the rapid prototyping system of the utility management system as the research object, this paper adopts an interface expansion method to expand the hardware interface of the simulation equipment synchronously in high speed and uses the FPGA-based signal multiplexing/demultiplexing together to effectively solve the problems of the real-time simulation equipment of the utility management system such as small quantity of channels and poor expansibility.

2. Real-time rapid prototype system architecture

In the early stage of the development of the utility management computer development, the model of the utility management system was established, and non-real-time and real-time simulation and verification was implemented to the model or algorithm in several times to verify the feasibility of the software/hardware scheme in the rapid prototype of the utility management and management computer products. The generator based on model code is used to generate the automatic code and download the code for the target in the middle and late period of the product development and the utility subsystem simulation capability shall support the local joint simulation based on the UMS finished product to guarantee the UMS product smoothly entering the next development and joint debugging.

The Simulink tool is used to build the UMS_RCP model. The distributed computer measurement and control architecture based on PXI bus is used to implement the modeling and off-line simulation of the UMS rapid prototype and the software platform and rapid prototype hardware platform based on NI Veristand are used to implement the rapid prototyping simulation verification based on the model by encapsulating code or model[3]. The hardware and software platform of this platform has the following characteristics:

1) The high-performance real-time simulation platform is selected to meet the requirements of interface that the synchronization shall be less than 200μs;
2) The self-design signal conditioning circuit as model required can meet the requirements of RIU interface adaptation;
3) Software tools provide modeling environment and basic software simulation package for utility management system and the subsystems[4];
4) Based on the digital model, PXI semi-physical simulation platform and NI Veristand software platform, the model-based rapid prototype design and simulation are realized, and it has some physical load simulation capabilities.

The UMS_RCP simulation environment architecture and operating principle are shown in figure 1.

![Figure 1. UMS rapid prototype system architecture and operating principle](image-url)
simulation[5]. The control model can be loaded into the rapid prototype platform after the completion and the I/O interface and the bus interface in the rapid prototype environment are interconnected with the semi-physical finished product or simulation device of the utility sub-system to realize the joint simulation verification.

As shown in figure 1, the UMS rapid prototype system uses an open PXIe system architecture and with good integrity and reliability, it supports scalability. The platform is mainly composed of PXIe chassis, controller, analog board and reflective memory network and can provide typical signal interfaces of airborne utility system, such as analog, discrete, frequency acquisition, AC analog acquisition, capacitance measurement, and resistance measurement and so on. In addition, it is equipped with the signal conditioning interface circuit to provide analog signal input and output, relay output, discrete signal input and R/LVDT signal at different amplitudes for the RIU.

Although the PXIe system architecture has strong expandability, there are large differences in quantity and type of interfaces of different models of aircraft utility management system. There are the problems such as large system scale, poor flexibility, and low cost-effectiveness by only using the PXIe system architecture. Therefore, for the characteristics such as various types of interfaces, large quantity and large differences between products of the aircraft utility system, the study of interface expansion technology based on PXIe architecture can effectively shorten the development cycle of utility management system products and reduce the research and development risk.

3. Interface expansion algorithm design and implementation

3.1 Interface expansion operating principle
The multiplexing/demultiplexing algorithm implementation includes four steps:

1) In a simulation period T, encapsulate the N-way simulation signals based on the classes in the UMS CPU model and then send them to the expansion substrate sharing memory;

2) Divide the signals of the airborne utility system according to the interface types, and write the X signals of the same type into the reflective memory of the expansion board in the same clock cycle;

3) Call the multiplexing algorithm in the FPGA of the expansion substrate. Read the data package from the sharing memory, divide the N-way signals into M groups, count the FPGA clock signals and generate a synchronous Sync clock signal. Recombine the Sync clock signal, addressing signals of multiplexing algorithm and the simulated signals after grouping and divide them into M segments in a simulation period T before transmitting them to the external devices through simulation channels in batches;

4) The demultiplexing algorithm is implemented in the interface board FPGA of the expansion I/O of the external device. According to the Sync clock signals and the addressing signals in the recombination signals, restore the M segment signals received after demultiplexing to N-way analog signals and send them to the system controller under test.

In a real-time simulation system, each embedded board of the simulation machine has 16 emulation channel and each board just can send the 16-way simulation signal in a simulation step size under the normal mode. It is planned to send the N-way (N>16) simulation signals in one simulation period T with the N1 (N1<16) simulation channels by the multiplexing/demultiplexing algorithm so as to verify the feasibility of the algorithm function.

3.2 FPGA multiplexing algorithm implementation
The FPGA clock cycle T1=10ns and the counting module is used to count the clock signals. The CPU simulation period in the model is set as T=25us and the clock signal count K does not exceed T/T1=2500.

Each embedded board of the simulation machine contains 16 emulation channels CH0–CH15. It is planned to use N1=14 channels and reserve 2 of them as the special signal channels. The simulation channel resource allocation is shown in table 1. The "Sync clock signal" occupies 1 bit to realize multiplexing and demultiplexing data synchronization; the "simulated signal" occupies 10 bits, that is,
Each group can send different signals in 10 ways; "Adr addressing signal" occupies 3 bits, then \( M \leq 2^3 = 8 \). 8 group signals at most can be sent in a simulation cycle. This kind of resource allocation can be used for simulation signals up to \( 10 \times 2^3 = 80 \) channels. Since the number of simulation signals \( N = 40 \), then we take \( M = 4 \).

### Table 1. Simulation channel resource allocation

| Simulation Channel | CH0          | CH1-3        | CH4-13       | CH14-15      |
|--------------------|--------------|--------------|--------------|--------------|
| Signal Type        | Sync clock signal | Adr addressing signal | Simulation signal | Reserved |

Each group of simulation time shall be divided appropriately. If the total time-consuming of FPGA exceeds \( T = 25 \)us in one step size, there will be the CPU model simulation timeout. If the simulation time period of each group of FPGA is too short, this will increase the difficulty of FPGA demultiplexing and cause the simulation signal loss[6]. The multiplexing algorithm timing is shown in figure 2. In the \( n \)th \( T = 25 \)us, there are 0~6us in the first group, 6~12us in the second group, 12~18us in the third group and 18~25us in the fourth group and each group multiplexed signal consists of 10-bit simulation signal and a 3-bit Adr addressing signal; In the \( n+1 \)th simulation step size \( T = 25 \)us, the multiplexing algorithm can be reused.

![Figure 2. Multiplexing algorithm timing](image)

The FPGA multiplexing algorithm model built under the XSG component includes Sel signal function encapsulation module for sequential selection of the multiplexing algorithm and the Sync signal function encapsulation module for data synchronization of the demultiplexing algorithm and the VHDL hardware code generated finally will operate in the FPGA[7].

#### 3.3 FPGA demultiplexing algorithm implementation

The interface board of the external device receives the recombination signals sent by the simulation channel CH0~CH13. The FPGA demultiplexing algorithm performs the demultiplexing function of 10-way analog signals which will be sent to the system controller under test from channels CH4~CH13 according to the addressing signals received by channels CH1~CH3 when detects the rising edge of the Sync clock signal from channel CH0.

The demultiplexing algorithm timing is shown in figure 3 and \( T = 25 \)us at the \( n \)th. There is the rising edge of the Sync clock signal at the 3us which calls the demultiplexing algorithm. The addressing signal is the first group within 0~6us so the first set of 10-way simulation signals are decoded; similarly, the other three sets of demultiplexing algorithms are called at 9us, 15us, and 21us, respectively; \( T = 25 \)us in the \( n \)th simulation step size then the demultiplexing algorithm is reused.
The addressing chip selection signals generated by logical processing of the addressing signals received by the channels CH1 to CH3 and they can be taken as the signal input of the demultiplexing program, as shown in the truth table of table 2.

| Simulation Channel Addressing | Addressing Signal |
|-----------------------------|------------------|
| CH3  | CH2  | CH1  | Adr0  | Adr1  | Adr2  | Adr3  |
| C    | B    | A    | AB'C' | A'BC' | ABC'  | A'B'C |
| 0    | 0    | 1    | 1     | 0     | 0     | 0     |
| 0    | 1    | 0    | 0     | 1     | 0     | 0     |
| 0    | 1    | 1    | 0     | 0     | 1     | 0     |
| 1    | 0    | 0    | 0     | 0     | 0     | 1     |

The demultiplexing function module is encapsulated by the Verilog code. The function module includes three types of signal inputs, i.e. synchronous clock Sync, addressing chip selection Adr, and simulation signal din[9:0]. Among them, the addressing chip selection Adr takes the corresponding value Adr0~Adr3 in table 2 from the first group to the fourth group. The signal output is the simulation signal dout[9:0] after the demultiplexing of each group and the 40-way simulation signals after the demultiplexing are sent to the system controller under test through dout_a[9:0], dout_b[9:0], dout_c[9:0], and dout_d[9:0].

4. Simulation verification of interface extension algorithm

4.1 Simulation verification of multiplexing algorithm

The simulation signals Signal1~40 in the CPU model shall be modified online after the multiplexing algorithm model of CPU model and the FPGA are compiled and downloaded into the simulator.

The multiplexing algorithm simulation waveform is shown in Figure 4. The Sync clock signal output to implement the multiplexing and demultiplexing data synchronization and the signal are sent through the simulation channel CH0. The second line is the recombined Adr addressing signals which are sent through the simulation channels CH1~CH3. The simulation waveform of the output simulation signal channels selected CH4, CH5, CH6 and CH7. In each simulation cycle, the simulation channel outputs 4 signals, that is, signal "0011" of channel CH4, signal "0101" of channel CH5, signal "1010" of channel CH6 and signal "1100" of channel CH7.

The simulation waveform results in figure 4 show that the multiplexing algorithm operating in FPGA can complete the multiplexing and recombination of the 40-way simulation signals correctly in a simulation cycle.
4.2 Simulation verification of demultiplexing algorithm

The demultiplexing algorithm simulation waveform is shown in figure 5. There is the rising edge of the Sync clock signal at the 3us and the demultiplexing algorithm program is called. At this time, the Adr addressing signal is "001", and the first group of simulation signals Sigal_1_4~Sigal_1_7 are output by the demultiplexing of the simulation channels CH4~CH7. The figure shows that Sigal_1_4, Sigal_1_5, Sigal_1_6 and Sigal_1_7 are consistent with the output signals from the corresponding simulation channels CH4, CH5, CH6 and CH7 at this time; The Adr addressing signal is "010" at 9us and the second group of simulation signals Sigal_2_4~Sigal_2_7 are output by demultiplexing. The figure shows Sigal_2_4, Sigal_2_5, Sigal_2_6 and Sigal_2_7 are consistent with the output signals from the corresponding simulation channels CH4, CH5, CH6 and CH7 at the time; The third group of Sigal_3_4~Sigal_3_7 and the fourth group of Sigal_4_4~Sigal_4_7 are sent at 15us and 21us.

The simulation waveform results in figure 5 show that the demultiplexing algorithm can decode the multiplexing signals in one simulation cycle correctly and send the restored simulation signals to the system controller under test.

5. Conclusions

For the technical problem that there are not enough simulation channels in the real-time simulation system, this paper proposes a set of signal multiplexing/demultiplexing algorithm and performs the function simulation verification of the algorithm through the example application. The simulation results show that the algorithm realize the function to interconnect the simulation signals in large quantity with the controller under test through few of simulation channels synchronously in a simulation step size.

This paper uses the set of multiplexing/demultiplexing algorithm for a complex utility system, which is very important for optimizing the design of the complex utility system, the feasibility research and determining the operating conditions of the system and fault analysis.
References
[1] Guo Jianying, Sun Yongquan, Yu Chunyu et al. (2014) Some theories and methods for reliability prediction of complex utility system. J. Chinese Journal of Mechanical Engineering, 50(14):1-13.
[2] Guo Xiwei, You Xiaojie, Xu Congqian et al. (2012) Real-time simulation of hardware in powerful electric traction control system. J. Transactions of China Electrotechnical Society. 27(4):65-70.
[3] Zhang Hongchang, Ding Jianwan. (2015) Research on real-time simulation method of handling stability based on Modleica. J. Computer Engineering and Applications, 51(3): 31-34.
[4] Song Wei. (2008) Research on hardware-in-the-loop simulation of UAV based on MATLAB. D. Nanjing: Nanjing University of Aeronautics and Astronautics.
[5] Liu Chen. (2014) Semi-physical simulation system of CRH5 high-speed EMU D. Beijing: Beijing Jiaotong University.
[6] Hao Qi. (2015) FPGA modeling and hardware-in-the-loop simulation of electric traction AC drive system. D. Sichuan: Southwest Jiaotong University.
[7] Zhou Zhiguo, Yang Liang, Liu Zhiwen. (2012) Realization of real-time semi-physical simulation platform based on FPGA. J. Computer Engineering and Applications. 48(S2):115-118.