Sensors & Transducers, Vol. 190, Issue 7, July 2015, pp. 58-62

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

The Development of Synchronization Function for Triple Redundancy System Based on SCADE

[1] Moupeng, [2] Duan Xiaojun

[1] AVIC Chengdu Aircraft Industry Company Chengdu, Chen Fei Road Qingyang District, Chengdu, 610000, China

[2] National Laboratory of UAV Special Technology, Northwestern Polytechnical University, West Youyi Road 127, Xi'an, 710072, China

[1] Tel.: +86-29-88451175, fax: +86-29-88451175

[1] E-mail: dxj_nwpu@sina.com

Received: 20 June 2015 /Accepted: 21 July 2015 /Published: 31 July 2015

Abstract: Redundancy technique is an effective approach to improve the reliability and security of flight control system, synchronization function of redundancy system is the key technology of redundancy management. The flight control computer synchronization model is developed by graphical modeling method in the SCADE development environment, the automatic code generation technology is used to generate high level reliable embedded real-time code for synchronization function, omitting the code test process, shorten the development cycle. In the practical application, the program can accomplish the functional synchronization, and lay a well foundation for the redundancy system. *Copyright © 2015 IFSA Publishing, S. L.*

Keywords: SCADE, Triple redundancy, Flight control computer, Synchronization.

1. Introduction

With the use of UAV more and more widely, the reliability of UAV becomes more and more outstanding, and the redundancy technology is the effective way to improve the reliability. Redundancy technique, also called tolerant technology, is the way to improve the reliability of product or system through increasing multiple resources (hardware and software configuration) and the rational management for multiple resources. The key technology of the reasonable management of multiple resources is the synchronization among hardware. Redundancy flight control system consists of several flight control computers, can greatly reduce the task failure rate due to the failure of the flight control computer of UAV. The functional synchronization guarantee the redundancy system sample the input at the same time and the clock error accumulation between the redundant system channels is accumulated; the synchronization of redundancy system is the necessary conditions for the detection of the redundancy system. The functional synchronization directly influences the properties of the redundancy system, so it is very important to solve the synchronization problem among flight control computers. The A SCADE is a high level safety and security development environment, graphical modeling method is used for system primary design and detailed design in SCADE develop environment, the high security embedded code can be generated by the code generator of the SCADE, and omitting the code test process, shorten the development cycle. In this paper, triple redundancy flight control computer system is as an example to introduce functional synchronization based on SCADE.

2. The Hardware Equipment of Triple Flight Control Computer System

Triple redundancy flight control computer system (referred to below as the redundant system) consists of three individual computers, which will be denoted as flight control computer A, flight control computer B and flight control computer C. The redundancy system has a special signal lines for the synchronous signal transmission between flight computers. The DO32 line of flight control computer A transmits the synchronization signals to the other computers; the DI 31 and D32 lines are respectively connected to the other flight control computers to receive these synchronous signals. Each computer has an independent hardware timer that is used for generating the interrupt signal in the every frame. The structure of the redundancy system is shown in Fig. 1.

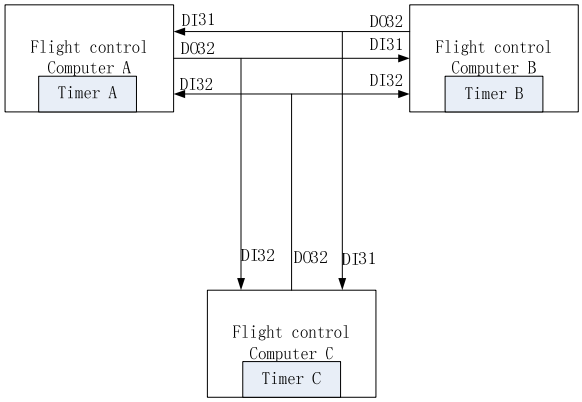

Fig. 1. Triple Redundancy System.

3. Redundancy System Functional Synchronization Requirement Analysis

The synchronize function must be executed once at the beginning of every monitor period in the flight control computer A, B and C. Functional synchronization method is the "double handshake" synchronization based on the software and hardware. When the redundancy system power on, the backplane circuit will finish the timer calibration for each flight control computer firstly in order to make the timer begin counting at same time, then the flight control computer enters the monitor period, the interrupt signal will be generated by the timer every 20 ms. In the monitor period, flight control computer must finish functional synchronization with the other two computers firstly, and then begin to run another functions. The functional synchronization of the redundancy system consists of the first synchronization sub-function and the synchronization recovery sub-function. The flight control computer A is as an example to illustrate the synchronization function. The first synchronization is the first attempt to synchronize in the new period. The DO 32 line of flight control computer A output the high level synchronization signal firstly, the signal through the synchronous signal lines route to the DI 31 line of flight control computer B and the DI 32 line of flight control computer C. Then receive the synchronization signals from DI 31 line (flight control computer B) and DI 32 line (flight control computer C). If the flight control computer has received any high level synchronization signals from the DI 31 line and DI 32 line in the specified time, the first synchronization is successful; otherwise the first synchronization is failure. The first synchronization method flow chart is shown on the left of Fig. 2.

The purpose of the synchronization recovery sub-function is the synchronization attempt again, when the first synchronization sub-function is failed. The synchronization recovery sub-function will take the next whole new frame, the failure counter add 1. The flight control computer receives the synchronization signals from DI 31 line and DI 32 line. If the flight control computer has received the high level synchronization signals from at least one of the DI 31 line and DI 32 line in the specified time, the synchronization recovery is successful; otherwise the synchronization recovery is failure. If the synchronous recovery is failure, only local flight control computer is valid and will be no longer to attempt synchronization with the other flight Control computers. The synchronization recovery method flow chart is shown on the right of Fig. 2.

4. Development of the Redundancy System Synchronization Based on SCADE

4.1. Intro of SCADE

The SCADE (Safety Critical Application Development Environment) is a high security application development environment, which select the graphical modeling for software requirements analysis. The SCADE is based on strict mathematical theory. So through strict modeling for requirements, the Ambiguity and vagueness of the software specification will be cleared by SCADE. The code generator of SCADE (KCG) is the premier code generator which has passed the certification for all software safety and security level requirements of DO-178B and IEC61508. The development cycle of software can be greatly shortened, but also can guarantee high level safety and security of the redundancy system.

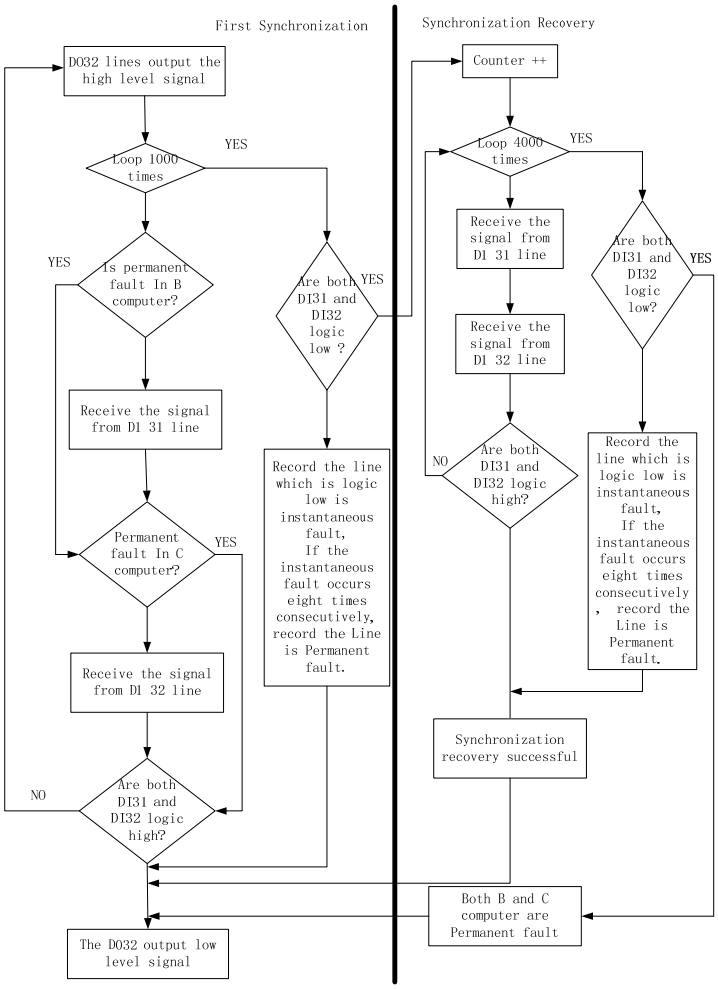

Fig. 2. Synchronization Flow Chart.

4.2. Development Process of the SCADE

When the SCADE is as the development environment, the software requirements need to be analyzed firstly; then the primary design and the detailed design are realized by the graphic modeling method; at last, the source code are generated by the SCADE code generator. Because the code generator of SCADE has already been certified, the code test procedure is not needed for the source code.

Primary Design Based on SCADE Graphic Modeling Method.

Analyze the synchronization methodology flow chart as shown in Fig. 2, the functional synchronization need following modules: SetDO32ture (initialize synchronization function), FSyn (first synchronization), RecSyn (synchronization recovery), EorrorBandC (both flight control computer B and C synchronization are Failure), setgloblefault (set perpetuate fault to true), EndSyn (synchronization exit) etc. The primary design by SCADE is shown in Fig. 3.

Analyze the relationship among these modules; indentify the data flow of synchronization function is that: firstly, the first synchronization sub-function receive the synchronization signals from DI31 and DI32 lines; then when the DI31Syn AND D132Syn is true, the first synchronization is successful, set the switch 1 to true, the EndSyn (synchronization exit) will run and output a low level synchronization signal per D032 line; when the DI31Syn NOTOR D132Syn is true, the RecSyn (synchronization recovery) is enable, the synchronization recovery sub-function can be realized by the internal module of RecSyn, the result of RecSyn is outputted per GloableFaultTepVar. The signals from DI31 and DI32 lines are the inputs of the ErrorBandC, the input of the GloableFaultSet is the output of the ErrorBandC.

4.3. Detailed Design for Synchronization Module

The detailed design of the functional synchronization of redundancy system is developed based on the primary design. ErrorBandC and FistSynchronize are as examples to illustrate the concrete implementation.

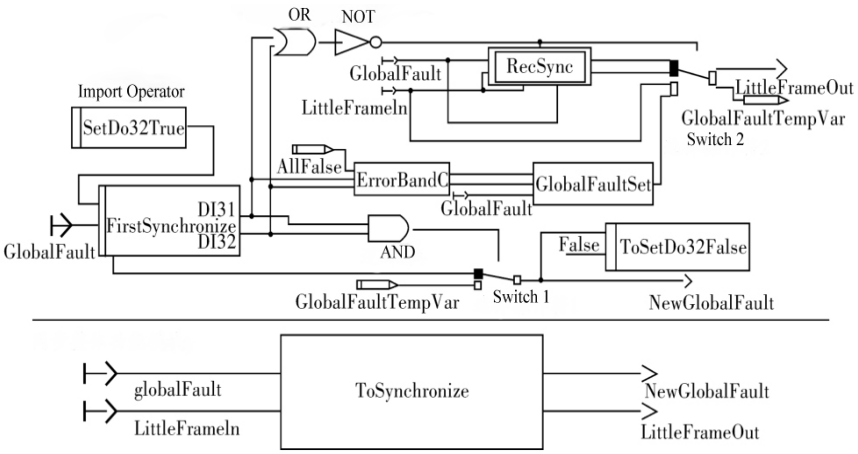

Fig. 3. Primary Design for Synchronization in SACDE.

4.3.1. Detailed Design of ErrorBandC Module

The function of ErrorB and C module is that if the synchronization instantaneous failure occurs in the DI31 or DI32 lines at least 8 consecutive times, then the corresponding line occurs permanent fault.

The model of ErrorBandC in SCADE workspace is shown in Fig. 4, when the instantaneous failure occurs at least 8 consecutive times, the output of Errorcounter block is true. The function of the FBY (follow by) block is holding the previous value of Errorcounter output, the initialization of the Errorcounter is false, it means B and c lines are normal before the execution of functional synchronization.

After the first execution, the output of the BFatalError and CFatalError are false. When the second execution, the output of FBY is result that the previous frame output of BFatalError and CFatalError OR the corresponding current outputs of Errorcounter. When BFatalError and CFatalError are true, the corresponding line occur the permanent fault.

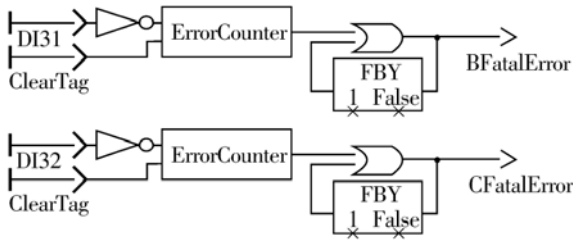

Fig. 4. Detailed Design for the ErrorBandC Module.

4.3.2. Detailed Design for First Synchronization Module

The import operator is a feature and flexible application of SCADE, the main idea of the import operator is that: the certified and functional code is imported to the blocks of the module without any changes. In addition, the hardware related drivers are generally not convenient to be developed by graphics modeling, so the underlying hardware related drivers only be accessed by import operators. SCADE prevent the subscript out of range for safety and security consideration. There is no loop operator in SACDE. When the program is required to drive hardware or use loop operation, the import operator is only method to implement these targets. The following application operators are used in the first synchronization sub-function.

Firstly, define an import operator block in SCADE workspace, named first synchrozine, the input of the block is globlefault which is the structure for recording the fault information of three flight control computers. The file “firstsynchrozine.c” will be generated by SCADE, and the developer can add the code in the file.

Add the functional code in the “firstsynchrozine.c”.

```
Void Firstsynchronize( constant Fault_globle *
globlefault)
    {
          For(i=0,i<1000,i++)
    {
   if                                    (!(glablefault-
>SynchronizeError.BForeverError)
    {
    While(status==0)
    {SynchronizeDI(0,&status, & di31);
    }
    }   if(di31)
    {
       *DI31=di31;
    }
    Status=0;
   if                                    (!(glablefault-
>SynchronizeError.CForeverError)
    {
        While(status==0)
    {
```

```
    SynchronizeDI(0,&status, & di32);
}
}
    if(di32)
{
  *DI32=di32;
}
Status=0;
if(*DI31 && *DI32) break;
}
Return;
}
```

5. Code Generate

SCADE can automatically generate the object oriented embedded code which is the ANSIC language format. After C code generated, it can be use in various systems without any changes just like the hand code. The automatically generated C code of synchronize function be embedded in flight control computer 20 ms processing, generate the build. Then the build is loaded in the three flight control computers.

6. Testing Result

After the redundancy system power on, the backplane circuit completes the timer calibration. Redundancy system enter the monitoring period, flight control computer outputs a logic high synchronous signal, then if the flight control computer can receive the logic high synchronous signals from the other computers in the specified time, the synchronization is successful, and the synchronization function of the computer output a logic low signal in this period. Observed the synchronization signals of three flight control computers through the oscilloscope; all three computers send the synchronization signals in every period, and the functional synchronization is successful in every period, the waves of the synchronization are shown in above of the Fig. 5. As shown in bottom of the Fig. 5, the synchronization function can be finished in 87 ns after the monitor period beginning; the characteristics of the functional synchronization meet the requirement of the redundancy system.

7. Conclusions

When we use the SCADE development environment, there are several advantages: develop process is clear; design structure is convenient to manage; shorten the development time and improve the development efficiency. Generate high level safe and reliable embedded source code automatically; the reliability of the software can be greatly improved. The testing result can indicate that the redundancy system can finish the synchronization; the performance of the synchronization is good; the functional synchronization is stable and reliable in practical application; thus greatly improving the reliability of flight control system, and improve the survivability of UAV.

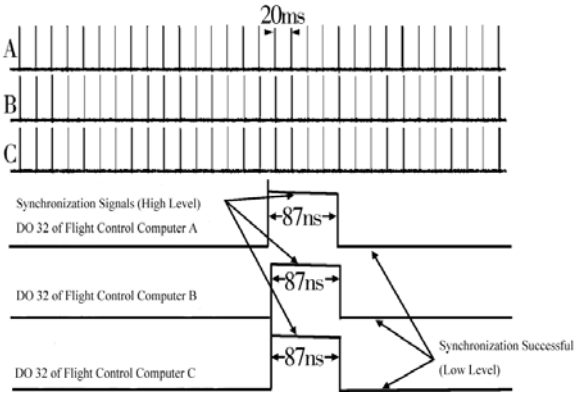

Fig. 5. Test Result.

References

[1]. Peiyi Niu, The failure threshold selection in fault detection of redundant control systems, *Acta Aeronautica et Astronautica Sinica*, Vol. 13, Jan. 1992, pp. 112-116.

[2]. Zhijun Huang, Yong Lei, Simulation Research of Sensor Fault Detection, *Computer Simulation*, Jan. 2005, pp. 110-112.

[3]. Caijuan Jia, Xiaopin Zhu, Zhou Zhou, The Study and Simulation of Sensor Fault Diagnosis for an UAVl Control System, *Computer Simulation*, Nov. 2005, pp. 53-55.

[4]. Esterel Written, The constructive semantics of Esterel, 2000.

[5]. Esterel Written. SCADE Training Course. 2002.

[6]. Naidu Amit K., Case study: Airbus A340 flight control system, *Computing, Department of Computer Science, University of Virginia*, Dec. 2002, pp. 1-12.

[7]. Yeh Y. C., Triple-Triple Redundant 777 Primary Flight Computer, in *Proceedings of the Aerospace Applications Conference*, Aspen, Colorado, 1996, pp. 293-307.

[8]. Briere D., Traverse P., Airbus A320/A330/A340 Electrical Flight Control – A family of fault-tolerant systems, in *Proceedings of the 23rd International Symposium*, Aerospatiale, Toulouse, France, June 1993, pp. 616-623.