A Method to Improve the Availability of Unmanned Aerial Vehicle Cluster

RuiBin OuYang, QingZhong Jia
Key Ministry Education Laboratory of Dynamic and Control of Flight Vehicle, Beijing Institute of Technology, Beijing, China
Ruikiy@outlook.com

Abstract—Taking unmanned aerial vehicles as background, this paper introduces a design method of a flexible master-slave working mode according to the thoughts of redundancy technology to improve the vehicles' availability[1] in order to reduce the influence of the breakdown of vehicles caused by terrible environments on the cluster. And the results shows that the system can continue to work through a host cooperating with other devices when one or more devices of the system break down. And when new device is added to the system, the system can absorb it. Thus, the abilities of unmanned aerial vehicles working under terrible conditions are improved.

Keywords-Unmanned aerial vehicle cluster; Availability; Redundancy technology; FPGA

I. Introduction

Unmanned aerial vehicles have caused wide attention and research because they are in small size, have low cost, and cost no casualties. And using unmanned aerial vehicle cluster to perform tasks has been applied in many fields like military and agriculture. For example, they can be used to spray pesticide for farmland in large scale, to transport rescue goods for remote disaster areas and to scout enemies' areas. However, when working under terrible conditions like bad weather or being attached by enemies, the vehicles damages easily. Especially when several vehicles have to work together, the cluster would fail to complete the task due to the damage of the host. So, to improve the availability of the unmanned aerial vehicle cluster has been an important research direction.

II. Redundancy Technology

In practical project, redundancy technology is often used to improve a system's availability. Redundancy technology includes static redundancy, dynamic redundancy and mixed redundancy. Static redundancy means to use additional subsystems or components to replace the broken ones. Its basic principle is to ignore the fault through majority voting. For unmanned aerial vehicle cluster, using static redundancy would burden the system and waste its power. The working vehicle cluster care little about the condition of a single machine but the whole system's ability to complete the task. Dynamic redundancy is based on reconstruction technology and is made up of several same or different modules. The system's redundancy construction changes with the case of the fault. It is very convenient to form a cluster with dynamic redundancy using several unmanned aerial vehicles. This paper designs a master-slave working mode for the vehicle cluster with dynamic redundancy.

III. System Design

Usually, unmanned aerial vehicles communicate through wireless devices. Because of its broadcasting feature, the vehicles working in master-slave mode connect with each other in a star type[2]. See Fig. 1. If the host breaks down, the subordinate vehicles could not get orders then. We often use hot or cold standby for hosts in a system if dynamic redundancy is required, that is to prepare one or more spare machines for the hosts. However, it is unrealistic and useless to use vehicles working as spare ones for the hosts. What's more, under terrible conditions, it is common that large scale damage would occur and many vehicles including the host and standby vehicles would break down at the same time. We hope that the vehicle cluster in master-slave mode can work in any conditions through a host cooperating with other vehicles. When new vehicles come, the host can identify then and add them into the cluster.

In the design here, every vehicle can work as the host in turn to control and monitor the others' conditions. Under the designed judging method, if one vehicle breaks down, the cluster could generate a host automatically and continue to work. A new vehicle can also be identified by the host and join the cluster. Thus, resources of every vehicle can be made full use of and the availability of the cluster is largely improved.

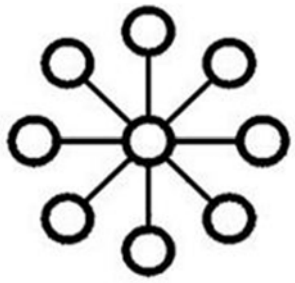


Figure 1. Star Type Constructure

A. Design of a Single Vehicle

Usually, unmanned aerial vehicle consists of main control chip like a micro control unit and other function modules. In this design, several FPGA evaluation boards are used to replace the main control chip to realize the function of logic control and wireless communication modules like HM-TRP is also used to simulate a vehicle to test the design.

B. Design of the Working Mode

The working mode for the vehicle is realized through the programming of FPGA. Fig. 2 shows the design modules. The main logic module cooperates with other modules so that the vehicle can work as we expect. The clock module is used to synchronize the clock of all the vehicles in the cluster. Universal asynchronous receiver and transmitter module[3] is used for vehicle to communicate. Every vehicle transmits data in time-sharing mode. The form of the 8 bits data is: 3 bit self- number, 1 bit master-slave mark and 4 bit operand. The increase or decrease of vehicles can be realized by transmitting data with self-number in the form designed so that they can be identified by the host. Besides, the self-number works as the heartbeat of the vehicles. It is easy for the host in the star type to judge whether a vehicle is normal or broken. The delay module generates delay signals and clear signals for the buffer of the receiver and transmitter. LEDs can be used to observe the status directly.

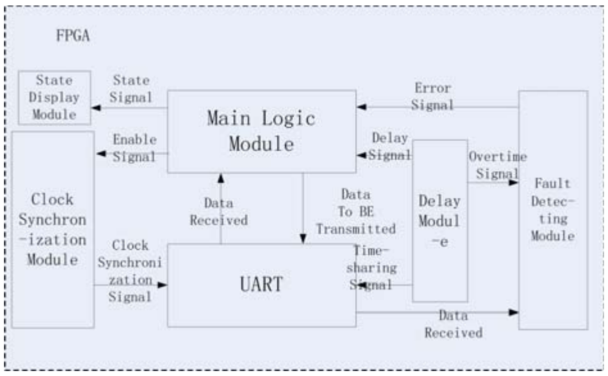


Figure 2. Modules in The Design

Fig. 3 shows the state machine of the main logic module. State0 and state1 are the waiting states after power up. They are used to set the first host during the initialization of the system. Here, the host can be any of the vehicle. State2 means that the vehicle works as a master when it can control and monitor other vehicles. State3 means that the vehicle works as a slave when it follows the order of the host. State4 means that the vehicle finds itself damaged. In this state, the vehicle can stop itself or hang. Fig. 4 and Fig. 5 show the design of state2 and state3 respectively.

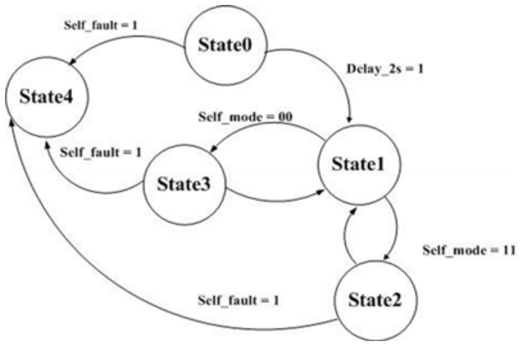


Figure 3. State Machine of State

In master mode, the host vehicle monitors the conditions of other vehicles and saves them in a register. If a certain bit in the register is 1, it means that the relevant vehicle works well. For example, if data in the register is “00000110”, it means vehicle marked number 2 and 3 work well (including itself). When the data is “00000010”, it means the vehicle whose self-number is 2 does not receive any signal and will treat itself as damaged. In every state, if the vehicle finds itself damaged, then it will go to s_master5 and transmit “00001111”. Otherwise, the host will send orders to the next normal vehicle and make it as the next host after confirmation by shaking hands.

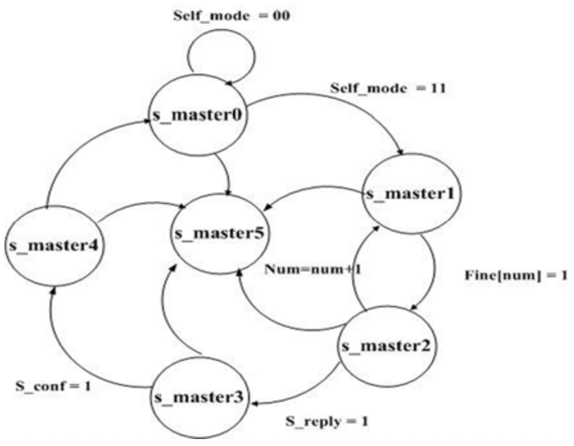


Figure 4. State Machine of Master Mode in State 2

In slave mode, the vehicle waits for the orders from the host. When receiving the orders from the host, it would become the host after shaking hands. If it does not receive the signal from the host, the rest normal vehicles would become the host after a different delay determined in the order of their mark number. After one slave vehicle becomes the host, the cluster could continue to work. It would go to s_slave7 if it finds itself damaged and will transmits “00001111”.

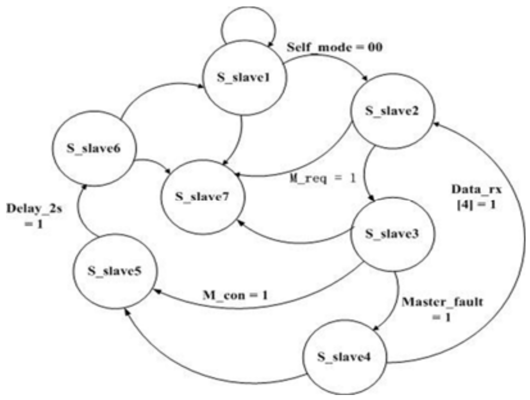
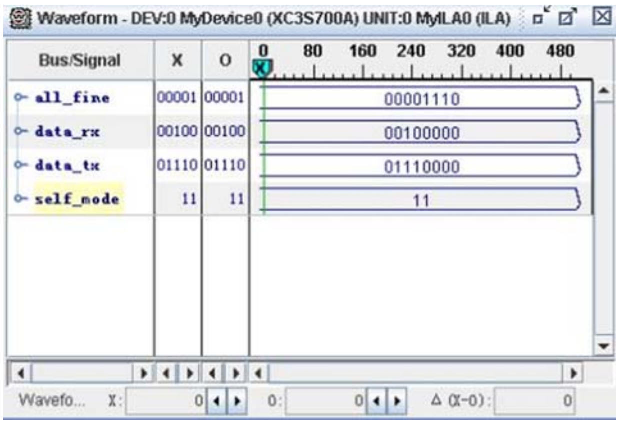


Figure 5. State Machine of Slave Mode in State 3

C. *Simulation Results*

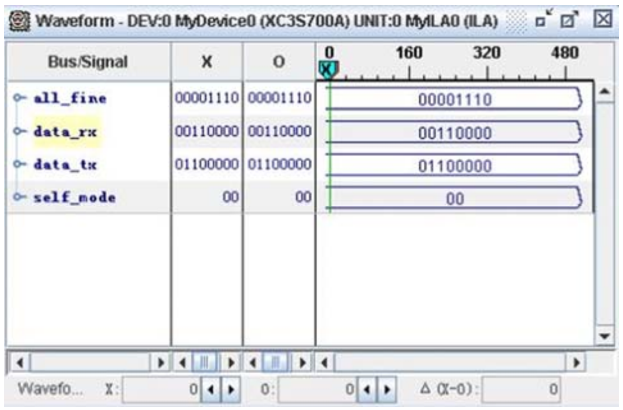
Three FPGA evaluation boards are used here to replace three vehicles and the working states are observed by LEDs. As it is shown in the result, when all the three vehicles work well, they work as the host in turn. Fig. 6 and Fig. 7 show the working state of vehicle 011 (marked number3). We can see that the information of the vehicle is saved in the register "all_fine" and it works in master mode. Fig. 7 shows that it is working in slave mode. After turning off the power of the wireless device, Fig. 8 shows its working state. And then, vehicle marked number 2 becomes the host. See Fig. 9.


Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	0
all_fine	00001	00001	00001110
data_rx	00100	00100	00100000
data_tx	01110	01110	01110000
self_mode	11	11	11

Wavefo... X: 0 O: 0 Δ (X-O): 0

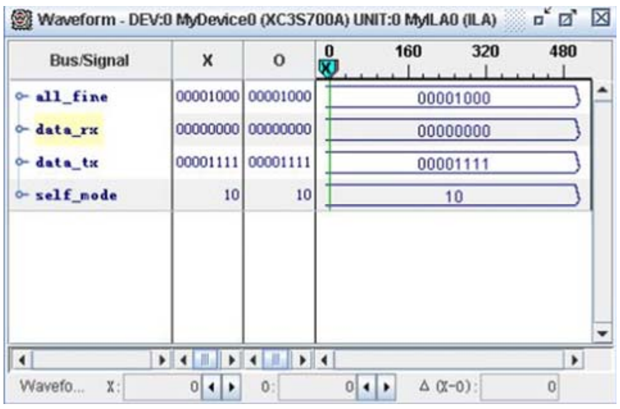
Figure 6. Working State of Vehicle 011 in Master Mode


Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	0
all_fine	00001110	00001110	00001110
data_rx	00110000	00110000	00110000
data_tx	01100000	01100000	01100000
self_mode	00	00	00

Wavefo... X: 0 O: 0 Δ (X-O): 0

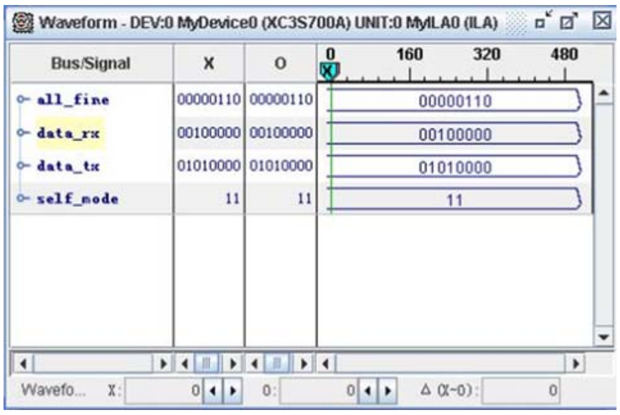
Figure 7. Working State of Vehicle 011 in Slave Mode


Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	0
all_fine	00001000	00001000	00001000
data_rx	00000000	00000000	00000000
data_tx	00001111	00001111	00001111
self_mode	10	10	10

Wavefo... X: 0 O: 0 Δ (X-O): 0

Figure 8. Working State After Damaged


Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)

Bus/Signal	X	O	0
all_fine	00000110	00000110	00000110
data_rx	00100000	00100000	00100000
data_tx	01010000	01010000	01010000
self_mode	11	11	11

Wavefo... X: 0 O: 0 Δ (X-O): 0

Figure 9. Working State of Vehicle 010

In order to observe clearly, here puts the working states of the three vehicles together. See Fig. 10. When one vehicle is not working. See Fig. 11.

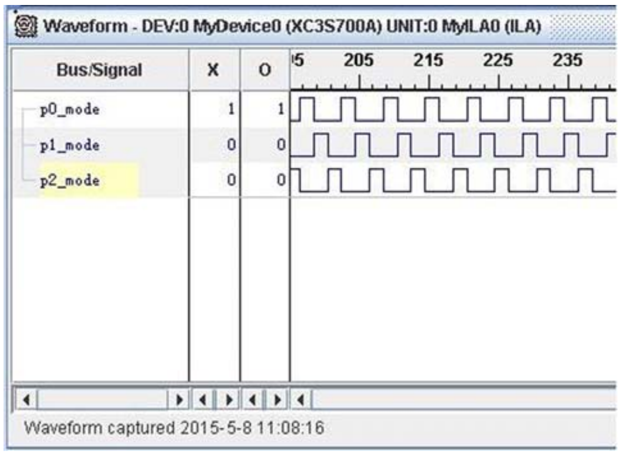

Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)
Bus/Signal
X
O
5
205
215
225
235
p0_mode 1 1
p1_mode 0 0
p2_mode 0 0
Waveform captured 2015-5-8 11:08:16

Figure 10. Working State of Three Vehicles

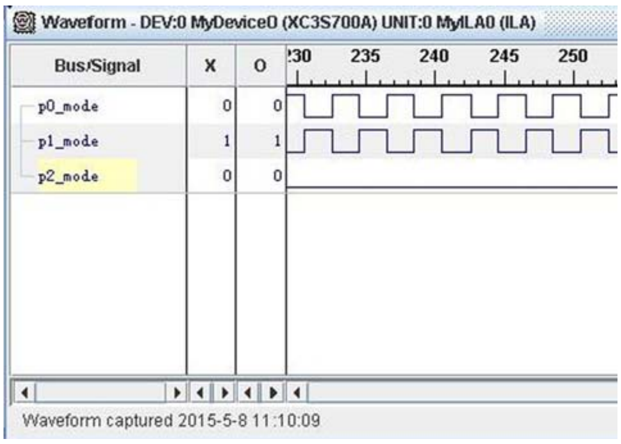

Waveform - DEV:0 MyDevice0 (XC3S700A) UNIT:0 MyILA0 (ILA)
Bus/Signal
X
O
230
235
240
245
250
p0_mode 0 0
p1_mode 1 1
p2_mode 0 0
Waveform captured 2015-5-8 11:10:09

Figure 11. After One Vehicle Breaking Down

IV. EXTENSION OF THE DESIGN

Logic design here is simple and simulation of the result is displayed in an easiest way, the design can actually be extended greatly. For example, we can use more than 3 bits to mark the number. Complicated way to detect the conditions of the vehicle can be designed. Besides, thoughts of the working mode can also be expanded in many other ways like the formation of unmanned aerial vehicles where a master is essential to carry out the arithmetic to control other vehicles. Once the master vehicle is broken, another one would replace it and continue the formation.

V. CONCLUSION

This paper takes unmanned aerial vehicles as background and designs a master-slave working method according to redundancy technology to improve the availability of the vehicle cluster. Vehicle cluster working in this mode can avoid the collapse because of the breakdown of the host. Besides, the cluster can reconstruct itself by absorbing new vehicles.

REFERENCES

[1] FengXing Shao, XiangPing Zhang, ZhiQiang Long and Xun Li, “The Base of Fault Diagnosis and Reliability Technology for Computer Application System”, 2nd ed, China Water&Power Press, February, 2011, pp.9-20

[2] ChenXi Zhang, ZhiYin Wang, ChunYuan Zhang, Kui Dai and HaiBin Zhu, “Computer System Constructure”, High Eduction Press, pp.298-305

[3] WenBo Xu and Yun Tian, Xlinx FPGA: Development and Application(Second Edition), TsingHua University Press, July, 2012.pp.261-284

[4] YuWen Xia,Verilog Digital System Design Tutorial,Second Edition,BeiHang University Press,June 2008.