Design of Flexible Interface Unit for Functional Reconfiguration of Measurement and Control System

Penghui Zhao [a, *] and Haibin Yuan [b]

School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China.

[a, *] zhaophl@buaa.edu.cn, [b]yuanhb@buaa.edu.cn

Abstract. Aiming at the problems of static fixed network architecture, low resource utilization and difficult maintenance of existing measurement and control system network, this paper introduces a design scheme of flexible interface unit which can realize functional reconfiguration. The design uses the FPGA as the core control element, and completes the configuration and normal work of the FPGA by communicating with the host computer. In this paper, the hardware design of the circuit and the whole workflow are given in detail. The realization of logic control circuit, power-driven module and initialization module are mainly introduced. The experimental data show that the flexible interface unit has fast switching speed and high precision, and can quickly and effectively complete functional reconfiguration in the measurement and control system with high reliability and configurable redundant components.

Keywords: Measurement and Control System; Relay-array; Functional Reconfiguration; FPGA.

1. Introduction

At present, computer, communication, command and information technology jointly promote the rapid development of measurement and control (TT&C) systems. The TT&C bus with measurement and control system as its main objective has gone through the development process of GPIB, VXI and PXI bus. It has been widely used in automobile industry, military equipment, aerospace testing and other fields [1]. But in the complex large-scale measurement and control system, the device connection is a static hierarchical structure, and a large number of test cables need frequent manual switching. In addition, due to the solidification of the node relationship, the upper nodes depend on the function realization of the lower nodes, any node failure may lead to the failure of the entire subsystem function, and the reliability and flexibility of the system is poor [2].

In order to solve the above problems, we can apply the strategy of system functional reconfiguration. For the system architecture that cannot meet the current requirements of measurement and control, we can remove, add or switch some of the equipment circuits, and reconstruct the functions needed by the link connection to restore the system. Therefore, this paper designs a flexible interface unit for functional reconfiguration of measurement and control system.

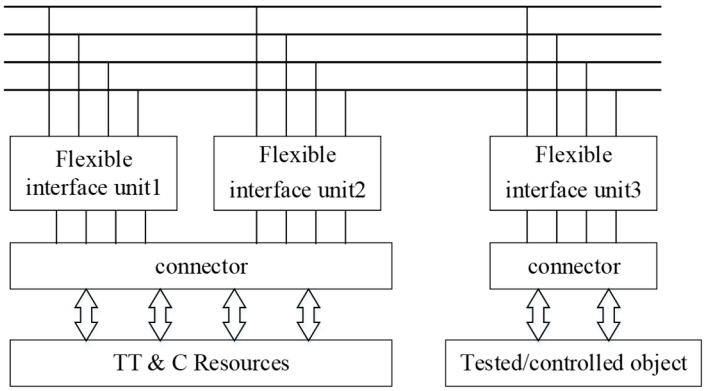

Fig. 1 System connection of the flexible interface unit

As shown in Fig.1, the tested/controlled units that need frequent switching, the measurement and control resources that are idle of the system, and other backup resources are connected to the system

network through the flexible interface unit. When the system has component failure or needs to switch the measurement and control object, the flexible interface unit is disconnected from the unrelated device, and then the required device is connected to the system network to realize the flexible connection of the system interface. The flexible interface unit can effectively improve the resource utilization efficiency and fault handling capability of the system, reduce the cost and time of system fault repair, and greatly enhance the reliability and maintainability of the measurement and control equipment, and improve intelligence, flexibility and safety of the test system.

2. Principle Model of the Flexible Interface Unit

The flexible interface unit connects the redundancy and backup resources in the measurement and control equipment to the system network. When components in the system are damaged, the faulty components can be isolated through the flexible interface unit to initiate backup to quickly restore the system. Therefore, according to the requirements of functional reconstruction, the flexible interface unit should have the following characteristics [3]: 1) The unit has enough interfaces to connect as many measurement and control resources as possible to the network; 2) The flexible unit has a faster response speed and higher precision to quickly switch components and reduce the disturbance when signal passing; 3)The unit has high security and avoids control errors such as line shorting and channel conflict. According to the needs of the measurement and control system, the functional block diagram of the designed flexible unit is shown in Fig.2.

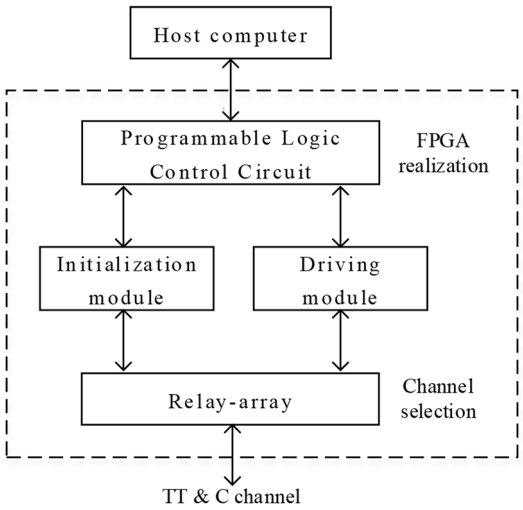

Fig. 2 Principle block diagram of flexible interface unit

The flexible interface unit is mainly composed of relay-array, initialization module, driving module and programmable logic control circuit. When there is a component failure, the host computer issues an instruction to the programmable control circuit, and the control circuit decodes and then controls the switch of the relay array through the driving module. The initialization module is used to start or reset to verify whether the flexible unit itself is working properly.

3. Design in Hardware of the Flexible Interface Unit

3.1 The Design of Relay-array Module

After analysis, the SRD-05VDC-SL-C relay was selected for its voltage range is large, and the cost performance is high. The rated voltage of the relay is 5V, the coil power is 360mW and the response time is less than 10ms. The internal structure of the relay is shown in Fig.3. Pin1 and Pin2 are the control signals of the relay, Pin5 is the input signal, Pin3 is the normally closed output, and Pin4 is the normally open output. When voltage of Pin1 is 5V higher than Pin2, the inductor coil is energized, and the generated electromagnetic force attracts the metal switch downward to close the

normally open output, and the Pin 4 and Pin5 are connected electrically. After the 5V control voltage is disconnected, the electromagnetic force disappears and the metal switch bounces back to the normally closed output again.

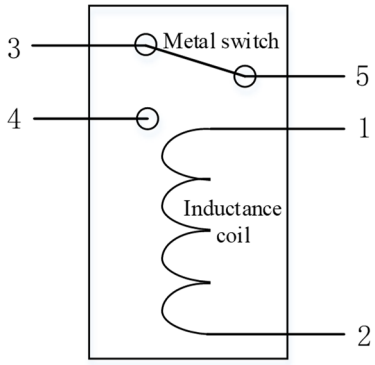

Fig. 3 The internal structure of relay

The relay array can be set to various specifications depending on the number of backup components in the measurement and control system. For different types of backup components, we can continue to add relay-arrays or expand the array size to connect more components to the measurement and control network. This paper designs an 8-input and 8-output relay-array model as shown in Fig.4. When the measurement and control signal is transmitted from an input channel, the switch of the relay can be controlled to select the channel of the output so that the connection of different input ports to the output port can be flexibly realized, and the flexible interface configuration of the measurement and control system is completed.

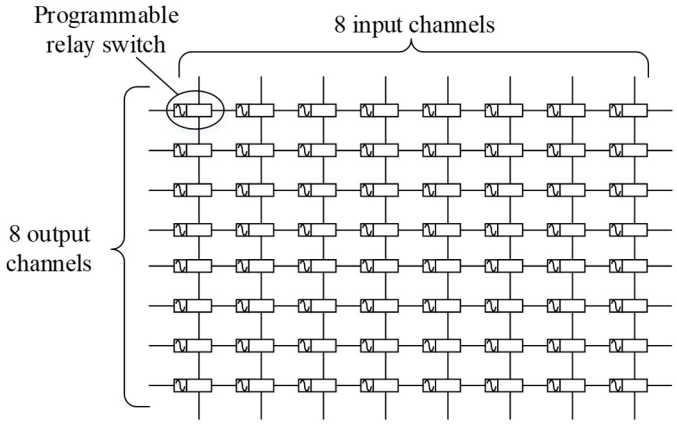

Fig. 4 Relay-array model

3.2 The Design of Programmable Logic Control Circuit

The main function of the programmable logic control circuit module is to realize data decoding, verification and latching [4]. Taking into account the dynamic changes in demand, we use FPGA to build circuits. With high integration, short development cycle, and rich internal trigger and I/O resources, FPGA is a good choice for designing control circuits, which can be programmed online. The host computer sends an initialization command and a relay on/off command to the FPGA so we can realize the safe start of the flexible interface unit and the function reconstruction when some component fails [5].

3.2.1 Introducing the Communication Protocol

For the 8×8 relay-array, the row number and column number are both marked as 0~7, and the relay can be uniquely determined according to the row number and column number. The host computer communicates with the FPGA through the UART serial port, each communication uses 1 byte of data and the data format of the serial communication is shown in Fig.5. The first two data formats are

applied to the host computer to send instructions to the FPGA, and the FPGA responds to the instructions and returns the result using the third data format. Since the row number and the column number each occupy 3 bits which indicate the labels of 0 to 7(from 000 to 111), so that we specify that in all data communication formats D2 to D4 represent the row numbers and D5 to D7 represent the column numbers. In the host computer control instruction, D0 indicates the control mode: when D0 is 1, if D1 is 1, it means to turn on the relay, and if D0 is 1, it means to turn off the relay. When D0 is 1, FPGA is required to check whether the relay can work normally. In the signal returned by the FPGA, that D0 is 0 indicates that the instruction received by the FPGA is legal and that D0 is 1 indicates the instruction is illegal. When D1 is 0, we can get the flexible interface unit executed successfully, and D1 is 1 indicates the execution failed. The specific relay label in all of the formats of transmitted data discussed above is given by D2~D7.

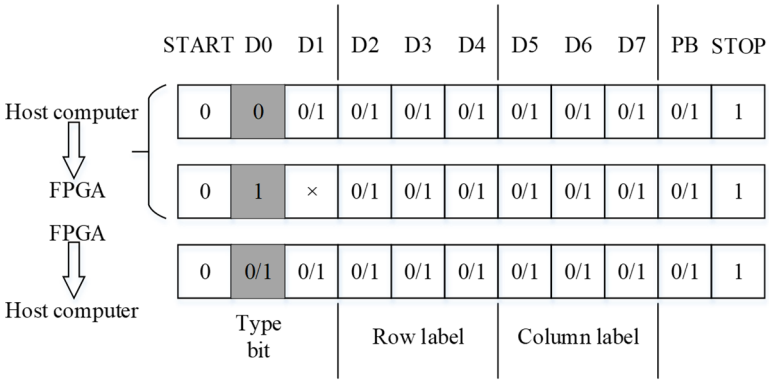

Fig. 5 UART serial communication frame format

3.2.2 FPGA Internal Programming

The FPGA device we selected is Xilinx's XC6SLX9, with XCF04 as configured Flash, which is programmed to decode, verify, and latch the circuit. We use the hardware design tool ISE to complete the input, synthesis, implementation, verification, download and other aspects of FPGA product development. After the design is completed, the simulation software ModelSim is used to simulate and analyze the logic circuit.

The decoding circuit analyzes the control commands received by the FPGA from the host computer, and determines operations to be performed according to the communication protocol. The latch circuit which is simple to implement in the FPGA can define Verilog register variables and assign them to the I/O port. Each relay is controlled by an I/O port for the corresponding output pin is locked while the register variable is unchanged [6].

Inspection circuit is used to determine whether the received instructions are legitimate. Any input row of the relay-array can only have one relay on at the same time. Otherwise, signal crosstalk may be disturbed and even serious faults may occur such as power shortage, device burnout, etc. Therefore, for the received instruction, the FPGA internally generates a status table to record whether rows and columns of relay-array have been occupied. For example, after the initialization is completed the host computer sends the instruction “01011101” to the FPGA, that is, the relay of row 3 and has been turned on, and the recording line number 3 is occupied at this time. After that we assumes the FPGA received the instruction of “01011001” which means the relay at row 3 and column 1 in array should be turned on. Comparing the status table at this time, we find the row 3 has been already occupied so that the instruction is illegal and cannot be executed, and the FPGA sends error feedback to the host computer.

3.3 The Design of Driving Module

The driving circuit module needs to be designed for the I/O port of FPGA can only lock the high or low level, but its current and sink current are insufficient, and the relay cannot be driven to complete the switching action. Here we choose the inverting driver MC1413, which consists of 7 NPN Darlington tubes, can be directly connected to TTL and CMOS circuits at 5V operating voltage.

It sinks up to 500mA and can withstand 50V in the off state, fully meeting the control requirements of the I/O port for relays [7]. The specific circuit schematic is shown in Fig.6.

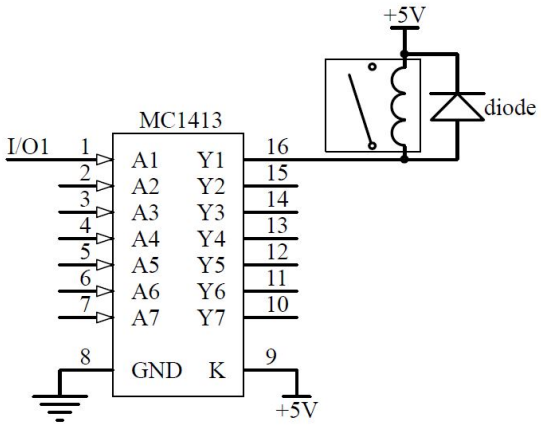

Fig. 6 Driving circuit schematic diagram

A1~A7 of MC1413 are input pins, Y1~Y7 are output pins, and K is connected to +5V voltage. The input pins are connected to the I/O ports of FPGA, the output pins are connected to negatives of relays, and the anodes of relays are connected to +5V. The diode is used for circuit freewheeling to prevent the back electromotive force generated by the coil from damaging the device when the relay is disconnected [8]. For example, the I/O1 of the FPGA is set to a high level of 3.3V. At this time, the output of the inverting driver Y1 is low, the current of the relay coil is passed, the metal piece is closed, and the relay is turned on. Similarly, when I/O1 is set to a low level, Y1 is connected to +5V due to the action of the inverter. Both ends of the relay are high level, no current flows through the coil, and the relay is disconnected.

3.4 The Design of Initialization Module

The initialization module can detect whether the relay-array works normally when the flexible interface unit is started or reset to avoid the problems of the switch failure, poor contact, abnormal resistance of the relay-array itself, which may have an influence on the signal transmission of the measurement and control system, and can effectively improve the flexible interface unit safety and reliability [9].

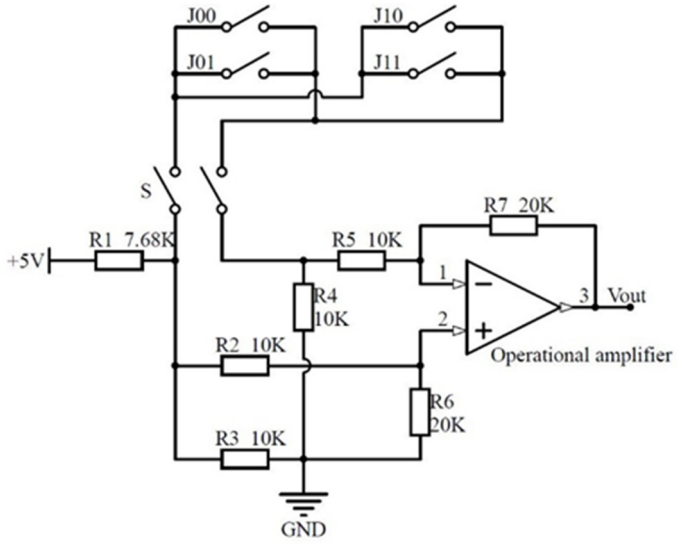

Fig. 7 Initialization circuit schematic diagram

Fig.7 is the circuit diagram of the initialization module which uses a +5V DC power supply. The figure shows four relays J00, J01, J10, J11 in the first two rows and the first two columns of the relay-array. The output port of the operational amplifier, Vout, is connected to the I/O port of FPGA, and the level characteristic of the I/O port is set to LVCMOS 3.3V. When the relay is closed, the resistance effect is ignored, the voltage of the non-inverting input terminal and the inverting input terminal of the operational amplifier are equal, the output voltage Vout is equal to 0, and the I/O port of the FPGA is detected as a low level. When the relay is disconnected, the voltage of the non-inverting input terminal of the operational amplifier is 1.647V and the inverting input terminal is grounded through R4 and R5 and connected with the output terminal through R7. We calculate that Vout is equal to 3.294V at this time, and the I/O port of the FPGA is detected as high level.

After the FPGA receives the initialization command, it closes the corresponding relay. For the initialization of a single relay, the I/O port connected to Vout is "1" when the relay is disconnected, and Vout is "0" when the relay is closed. A detection level error indicates that the relay cannot be normally switched or the resistance is abnormal. For a faulty relay, the FPGA sends an error message and the corresponding relay number to the host computer.

4. Design in Software of Flexible Interface Unit

The main program flow chart of the flexible interface unit is shown in Fig.8.

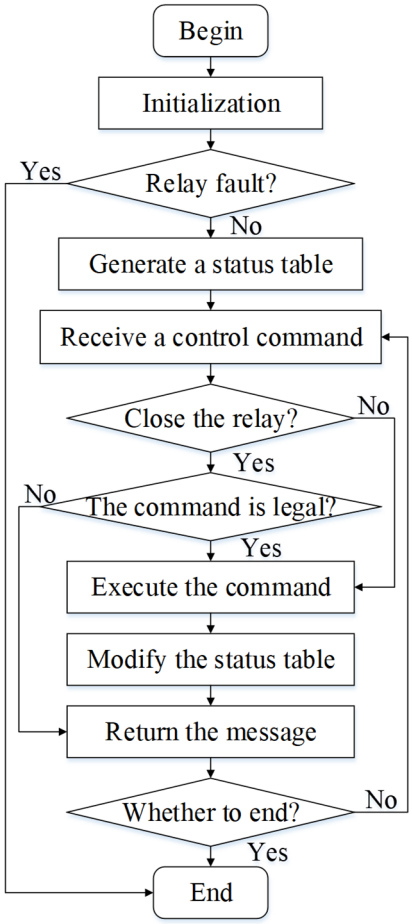

Fig. 8 Main program flow chart of flexible interface unit

After the flexible interface unit is started, the host computer sends an initialization command. The FPGA sequentially checks whether each relay is normal and when the relay is found to be faulty the system stops working. If no fault is found, the FPGA generates a status table of relay-array and records whether the relays are turned on in each row. Since all relays are disconnected at this time, the row and column values in the status table are all 0, indicating that they are not occupied. If the

relay is required to be disconnected by the control signal, directly perform the operation to set the corresponding I/O port low and change the status of the row to 0. If the command requires the relay to close, check whether the row is occupied in the status table. If it is closed, the command is illegal and cannot be executed. For legitimate operations, the FPGA executes control instruction and sets the I/O port high and changes the value of the row and column in the status table to 1. After the action is completed, the execution result is sent to the host computer, and then the host computer is requested to end the process. If the request is successful, the process ends, otherwise it continues to wait for the control command.

5. The Results of Experiment

In order to determine the safety and feasibility of the above design, the flexible interface unit is connected to the measurement and control system for verification by experiment. We add +12V DC power supply to the system, and control the 3×2 relays to turn on and off in turn, and record the execution time of the host computer program, that is, the time from when the command is issued until the last disconnected feedback signal is received. The above operation was repeated 4 times to obtain Table 1. The average time taken to continuously complete 6 relays is 75.843ms and the average switching time of each relay is 6.321ms.

Table 1. Execution time of relay switching

Experiment number	Execution time(ms)	Mean of execution time(ms)
1	75.996	6.332
2	75.731	6.310
3	75.775	6.312
4	75.851	6.320
Average value	75.843	6.321

The experiment also tested the voltage error of the flexible interface unit. The above 3×2 relay array was closed, and the input voltage and output voltage values were measured for comparison. The data results are shown in Table 2. The maximum difference between the test input voltage and the output voltage is 13mV, and the maximum relative error is 0.108%.

Table 2. Voltage measurement results

Relay number	Input voltage(V)	Output voltage(V)	Relative error(%)
J00	12.012	12.008	-0.033
J01	12.009	12.015	0.050
J10	12.005	12.018	0.108
J11	12.003	12.000	-0.025
J20	12.008	12.001	0.058
J21	12.011	12.007	-0.033

6. Conclusion

The above experimental results show that the flexible interface unit controls the relay on and off by the drive circuit through the host computer control, effectively completes its own initialization function and the function reconstruction task of the measurement and control system, and can be applied to the systems whose architecture is prone to failure, inconvenient maintenance or have redundant device. In the experiment, the test relay switching time is within 6.332ms, and relative error of the voltage is within 0.108%, which ensures the rapidity and reliability of the system reconstruction task. Due to the free implementation of the programmable control circuit, the flexible interface unit can be developed in terms of integration, intelligence and low power consumption, and has a good application prospect.

References

[1]. Hong C, Xin-Yu J. The Analysis and Actuality of Measure-control System Bus Technology[J]. Mechanical & Electrical Engineering Magazine, 2003.

[2]. Neves, Saotome. Comparison between Redundancy Techniques for Real Time Applications[C]// International Conference on Information Technology: New Generations. IEEE Computer Society, 2008.

[3]. Xiang Xuezhi, Kai Xianglong, Zhang Zhenyu. Design of switching matrix module based on PXI bus[J]. Foreign Electronic Measurement Technology, 2013, 32(12):76-79.

[4]. Wang Jie. Xilinx FPGA / CPLD Design Manual [M]. The Posts and Telecommunications Press, 2011.

[5]. Zhang H G, Xing H G, Wang D S, et al. Design of Massive Switching Matrix Module Based on Field Programmable Gate Array (FPGA)[J]. Torpedo Technology, 2006.

[6]. Ren Xiaodong, Wen Bo. CPLD/FPGA Advanced Application Development Guide[M]. Electronic Industry Press, 2003.B. R. Jackson and T. Pitman, U.S. Patent No. 6,345,224 (8 July 2004).

[7]. Chen Jiandong. A Practical Circuit of MC1413[J]. Home electronics, 1998(01):21. R. C. Mikkelson (private communication).

[8]. Han Zhiqiang, Zhang Jianjun, Yan Ruliang. Relay Fault Analysis and MATLAB Simulation [J]. Electrician technology, 2006(10):13-14.

[9]. Ge Genhua, Zang Chunhua. Horizontator measurement and control cabinet with self-inspection and self-protection functions[J]. Electronic Technique, 2006(08):44-48.