中国科学技术大学
University of Science and Technology of China

I2C Management Based on IPbus

Author:Luo Shiyu
Speaker: Yu Hongwei

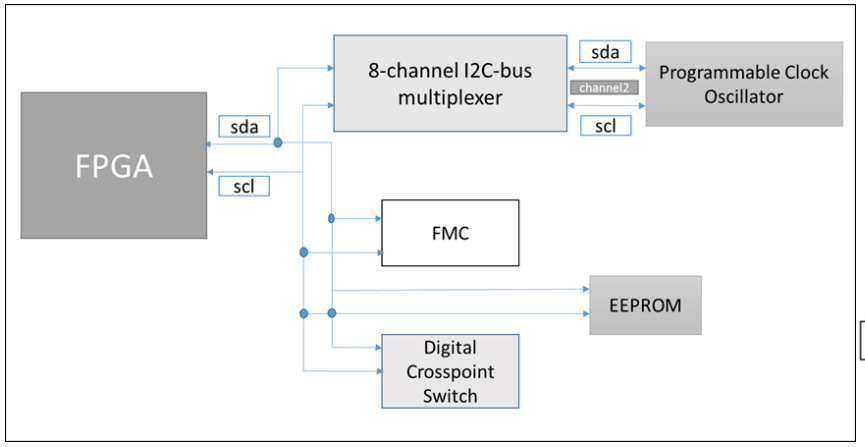

Fig. 1 I2C device connection diagram

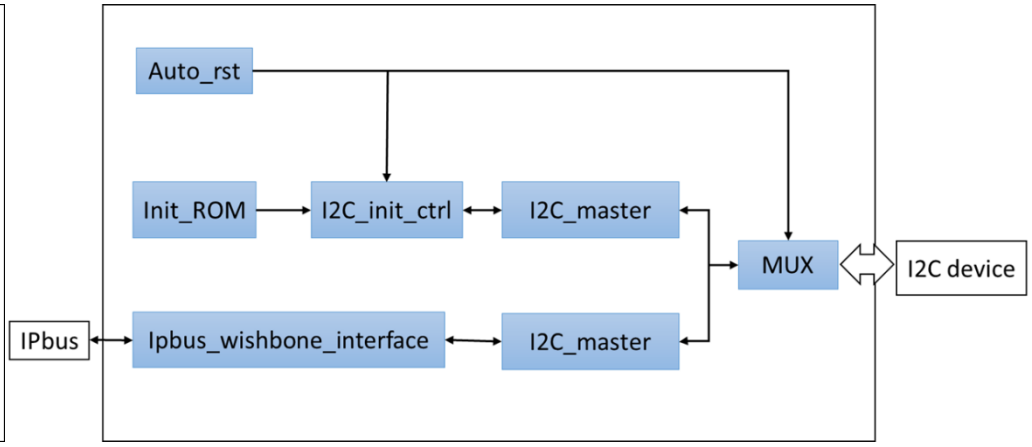

Fig. 2 Logical connection diagram

Poster

中国科学技术大学
University of Science and Technology of China

I2C Managenent Base on IPbus

Shiyu Luo[1,2], Junfeng Yang[1,2], Kezhu Song[1,2], Hongwei Yu[1,2], Tengfei Chen[1,2], Tianbo Xu[1,2]

1. State Key Laboratory of Particle Detection and Electronics, University of Science and Technology of China,Hefei,230026,China
2. Department of Modern Physics, University of Science and Technology of China,Hefei,230026,China

Introduction

1. Introduction

The CBM experiment are composed of kinds of detectors, including Micro Vertex Detector (MVD), Silicon Tracking System (STS), Muon Chamber (MUCH), Ring Imaging Cherenkov Detector (RICH), Transition Radiation Detector (TRD),Time of Flight Detector (TOF), and Projectile Spectator Detector (PSD). All these detectors are equipped with the appropriate Front End Electronics (FEE) boards and Readout Boards (ROB) which are located near to the detectors in the irradiated area.

Before the next generation GBTx based CBM DAQ system is built up, the data from the FEE and ROB are injected into a separate intermediate layer named DPB layer, where more complex functionalities, such as slow control, time synchronization, data readout, data pre-process and data format converting, are implemented. In current CBM DAQ architecture, all these functions will be implemented in a Field Programmable Gate Array (FPGA). Considering the optimal balance between the cost and performance, a versatile FPGA-based platform, the AMC FMC Carrier Kintex (AFCK) is used in kinds of the DPB projects to support different detectors.

2. System architecture

On AFCK, a set of I2C devices can be configured from the Module Management Controller (MMC) software or from the FPGA, including: I2C bus multiplexer, Clock crosspoint switch, Programmable Clock, I2C serial EEPROM with a EUI-48 address, System Monitor Ics, and Other I2C devices.

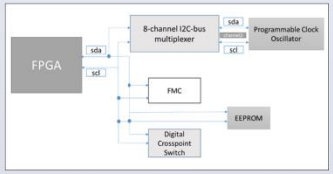

Figure 1. I2C connection diagram

Although these I2C devices can be configured via MMC when power up, it is important for DAQ system that all these I2C devices can be on-line configured via the slow control bus. In CBM DAQ, the slow control is based on IPBus , an open-source FPGA core which controls a Wishbone-like bus via Ethernet with fully dedicated software package (c++ or Python). But the settings of IPBus depends on the correct configuration of related I2C devices, including the I2C-bus multiplexer (choosing correct I2C bus), the clock crosspoint switch, the serial EEPROM with a EUI-48 address (providing the AFCK MAC address), which must be configured before the IPBus can be used. Hence a stand-alone, FPGA-based, programmable I2C configuration module is needed to implement the basic I2C initialization when power-up. After the IPBus link is setup, an IPBus I2C slave will fulfill the control needs for all the I2C devices connected. All these two parts consist the I2C management module for CBM DPB layers.

3. Ipbus_I2C_ctrl Module

As shown in Figure 2, there are two I2C controller in the module, one is for I2C device initialization, the other is for the in-system operation on I2C device via IPbus. There is an automatical reset submodule, in which the reset signal will be asserted when power-up. This reset signal will trigger the I2C device initialization module, in which a set of I2C configuration command stored in a ROM can be read out one by one and be sent to the appointed I2C buses.

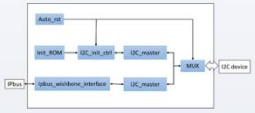

Figure 2. Logical connection diagram

After the I2C device initialization is finished, the I2C controller switches to the IPbus-based I2C controller automatically. The I2C controller in ipbus_i2c_ctrl core is an I2C master with WISHBONE interface, so it can be easily either accessed directly or connected to IPbus. A set of use-defined i2C configuration commands are used in the I2C device initialization. These commands should be saved in a ROM and be read out one by one by the I2C device initialization module. At the same time, the i2C device initialization module supports the input and output port for complex I2C configuration.

4. I2C_init_ctrl Module

The initialization controller module is controlled by a state machine. The state machine consists of a total of 11 states. The command is read in the initialization module ROM. When the 11th bit of the instruction is logic 0, the state of the state machine will be transferred from READ_ROM to READ_WRITE_REG, and then to CMD_END. According to the register port part of the command, the corresponding register in the I2C core is read or written. The specific state transition diagram is shown in Figure 2. When the 11th bit of the command is logical one, different operations are performed according to the register port portion of the command.

Hardware and logic architecture

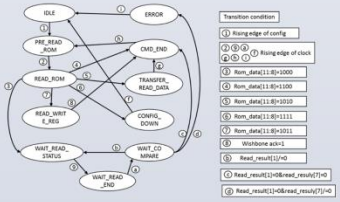

Figure 3. Initialize module state transition diagram.

5. Software and ROM command file programming

Before initialization, the programmable crystal oscillator is not connected to the FPGA. So the initialization module needs a separate clock to drive it. After initialization, the system clock will connect to FPGA, and the IPBUS control module is driven by the system clock.

6.Conclusion

The initialization control module can complete the configuration of some I2C devices. However, some I2C devices need to be configured in real time, so it is necessary to use IPbus to configure I2C devices online. The I2C device can establish connection with IPbus only after the initialization is completed. Therefore, the configuration of the I2C device on the AFCK board can be completed only when two configuration modules are used together.

Conclusion

lichao37@mail.ustc.edu.cn