A Platform for System-on-a-chip Design Prototyping

Yang Xun, Zhu Ming, Xue Hongxi, Bian Jinian, Hong Xianlong
yangxun@cs.ucla.edu, [zhum, xuehx, bianjn, hongxl]@tiger.cs.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing, China, 100084

Abstract

System prototyping is an important step in embedded system design, it can be used to validate the system functionality, performance and real time response to environment. A platform containing a microcomputer and a prototyping board is built for system-on-a-chip prototyping. A simple system-on-a-chip prototype reacting with its environment is built demonstrating the usage of the platform. The on chip system is built on prototyping board. The system environment is modeled with SystemC running on microcomputer. SystemC is a simulation environment that can be used to model hardware and software system. Some problems encountered during system prototyping is discussed. Further research on system-on-a-chip prototyping platform is presented.

Keywords

System-on-a-chip(SOC), SystemC, Prototype, Field Programmable Gate Array(FPGA), 8255 Parallel Programmable Interface, PPI

By the year 2002, it is estimated that more information appliance will be sold to consumers than PCs(Business Week, March 1999). This new market includes small, mobile devices that provide information, entertainment, and communications capabilities to consumer electronics, industrial automation, retail automation and medical market. These devices are highly complex with large amount of software contents including application code, device driver and operating system. On the other hand, the increase rate of IC complexity abides by the moore's law. Now complex system can be put on a single chip. A chip can accommodate a system with microprocessor, DSP, ASIC, A/D D/A converter, RAM, ROM, I/O interface , analog device, sensor and actuator etc.

At the same time, the adoption of hardware description language, high level synthesis tools and design methodology based on IP reuse has greatly increased the design capability and rendered the design of such complex system possible. Field programmable system-on-a-chip provides a convenient means of design system-on-a-chip. Users buy such IP as microprocessor and integrate the IP with programmable logic. According to Dataquest, this kind of design methodology will come on strong in 2003 and will account for about 80% of SOC design in 2008.

Here we developed a platform for SOC prototyping, it can be used to validate the system functionality, performance and real time response to environment. The paper is organized as follows. The first part introduces the platform, its construction and functionality. The second introduces the usage of the platform through the construction of a simple system prototype. The third part introduces the problem encountered and solved during system prototyping. The last part discusses the future research focus for prototyping platform.

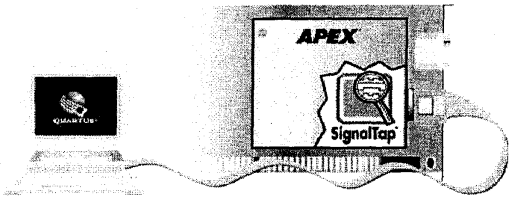
1.Prototyping Platform

The platform consists of two parts, a microcomputer and a prototyping board, as shown in figure1(a). The microcomputer and the prototyping board are connected via parallel port. A SystemC simulation environment is

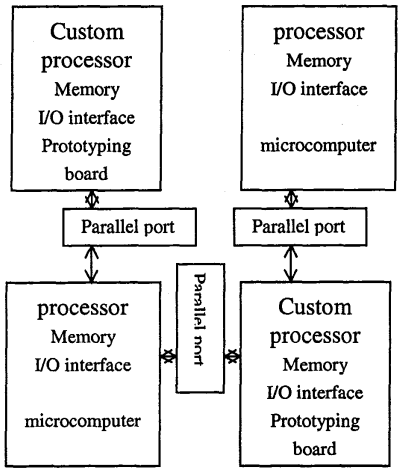
0-7803-6677-8/01/$10.00©2001 IEEE.

running on the microcomputer. SystemC is a system level description language extended from C++ with the capability of describing hardware and software. It is proposed by Synopsys and is provided with a simulation environment for hardware and software modeling.

The microcomputer can model a system with software. The prototyping board can model a system with hardware. The platform can be used to model a one processor system communicating with its environment, a two processors system. communicating with each other. The platform can have other configurations, with multiple microcomputers and multiple prototyping boards interconnected to model complex system, as shown in figure1(b).



(a)Basic Prototyping Platform Configuration



(b)Other Prototyping Platform Configuration

Figure 1 Prototyping Platform

The prototyping board has two ALTERA FPGA(apex EQ20k200EQC240-3, has volume of about 200,000 equivalent gates) on it. Hardware design can be put into the chip. The two chips are strongly connected. Any design that can't be accommodated in a chip can be partitioned into two parts and put into the two chips. On the prototyping board, there are other components such as power supply, oscillator, download circuit, parallel communication port etc. The FPGA is low power chip, it uses 1.8V internally and 3.3V to interface with other devices. These two voltages are gained from 5V with linear regulator. There are three modes of configuration for the FPGAs. Passive serial and JTAG can be used to configure the FPGAs, every time the prototyping board is powered up. The design file is downloaded from microcomputer into the FPGAs. The third mode of configuration is device configuration, in this mode several FLASH EPC2 devices are used. With JTAG configuration, the design file can be download into the devices. Every time the prototyping board is powered up, the EPC2 devices automatically configures the FPGAs. The on board parallel port is connected to the FPGA pins via 74LS245. The 74LS245 devices are used as a protection for the FPGA. Since the FPGA interface voltage is 3.3V and the microcomputer has a interface voltage of 5V.

2.Prototyping a Simple System-on-a-chip

2.1System Structure and Functionality

The system consists two subsystems, one SOC subsystem A and its environment subsystem B. Subsystem A comprises an 8bit 8085 compatible microprocessor, 2K memory and an 8255 parallel programmable interface. The 8bit 8085 compatible microprocessor CE(chip engine) is a software core that we developed. We also developed an 8255 parallel programmable interface for the system. Memory can be realized with FPGA development system. The other subsystem B comprises a general processor, memory and 8255 interface. The two subsystems are connected and communicate via the 8255 interface.

2.2Virtual Prototype for System Verification

During system verification phase, a system virtual

prototype is built. The two subsystem are modeled with CE, 8255 and memory model. And the two subsystems are connected to construct the system virtual prototype. For simulation, the initial content of memory should be defined with memory image file. The program that is running on the computer is converted into memory image file. Upon simulation initialization, the memory model reads the file and initializes the memory.

2.3System Prototyping with the Platform

Subsystem A is modeled with prototyping board. Subsystem B is modeled with microcomputer. Subsystem A is designed with hardware design tools as Design Compiler, VSS from Synopsys, Quartus from Altera. First subsystem A is verified with VSS simulator using system virtual prototyping. Then it is optimized with Design Compiler synthesizer and mapped to Altera technology library. The synthesis output is again verified with VSS simulator. After that, the synthesis output is fed to Altera FPGA development tool Quartus. After detail placement and routing, a design file is generated.

```
#include "src\systemc.h"
#include "cpu.h"
#include "8255.h"
void main(void)
{
  sc_signal<sc_int<8> > tData ;
  sc_signal<bool>
STB,enWR,nWR,nRD,WRack,RDack,
RESET,IRQ ;
  CPU     cpuM("cpu");
  M8255   m8255("8255");
  cpuM(STB,enWR,WRack,RDack,n
WR,nRD,RESET,IRQ,tData);
  m8255(nWR,nRD,RESET,IBF,nOB
F,nSTB,WRack,RDack,tData);
  sc_start(10000);
}
```

Figure 2 The SystemC Description of Subsystem B

At the same time, Quartus generates a timing accurate VHDL description. The VHDL description is simulated with VSS. If the result is right, the design file can be used to build the prototype subsystem A.

Subsystem B is modeled with SystemC. A model m8255 is described according to the behavior of 8255. The behavior of general processor and memory is described with cpuM model. The SystemC description of subsystem B is shown in figure 2.

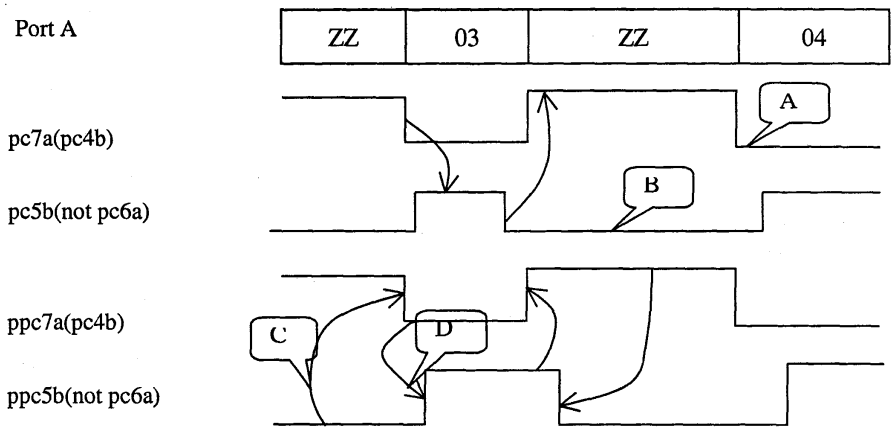
3. Communication Protocol Design for System Prototyping

The two subsystems communicate via 8255 in mode 2 through port A. The port C's seventh of subsystem A(denoted as pc7a) is connected to port C's forth bit of subsystem B(pc4b), the inversion of pc5b is connected to pc6a. They constitute a group of request/acknowledge signal for transmission of data from subsystem A to subsystem B. The request/acknowledge signal for transmission of data from subsystem B to subsystem A is connected likewise.

The communication protocol between subsystem A and subsystem B is shown in figure 3 with signals pc7a and pc5b. When there is data to be transmitted in subsystem A, the data is put on port A and pc7a becomes low, then pc4b becomes low. The falling edge of pc4b latches the data on port A into the register in subsystem B and sets pc5b to high indicating there was data in 8255 to be fetched by processor of subsystem B. The processor of subsystem B reads the pc5b signal. If it is high, it reads the data in register and set pc5b to low indicating that the data has been fetched.

The change of pc5b from high to low causes a rising edge in signal pc6a. The rising edge of pc6a sets pc7a to high indicating that the data sent to subsystem B has been fetched. The processor of subsystem A reads pc7a. If it is high, it sends new data on port A. Thus sets pc7a to low and begins a new round of data transmission.

This protocol works quite well in system virtual prototype. But it does work in system prototype, because the prototyping board and microcomputer have different



A:set to low by processor in subsystem A, B:set to low by processor in subsystem B
C:and data is ready, D:and data has been consumed by receiver
Figure 3 communication protocol between two subsystems

rates of execution. The protocol uses the falling edge of pc7a and rising edge of pc5b for triggering. If the pc5b changes to high and quickly return to low, the other subsystem can't detect the edge, an error will occur.

To solve the problem, we set up two extra signals in the respective subsystem ppc7 and ppc5 adding a handshaking protocol upon the 8255 communication protocol as shown in figure 4 with signals ppc7a and ppc5b. The initial values of ppc7a and ppc5b are high and low.

When there is data to be transmitted, the data is put on port A, pc7a is set to low. If pc7a is low and pc6a is high, ppc7a is set to low then pc4b becomes low. The falling edge of pc4b latches the data on port A into the register in subsystem B and sets pc5b to high indicating there was data in 8255 to be fetched by processor of subsystem B. The processor of subsystem B reads the pc5b signal. If it is high, it reads the data in register and set pc5b to low indicating that the data has been fetched. On the falling edge of pc5b and if pc4b is low, ppc5b is set to high.

Thus pc6a is low, which changes ppc7a to high. Afterwards ppc5b is set to low. Thus pc6a is changed to high. The rising edge of pc6a sets pc7a to high indicating that the data sent to subsystem B has been fetched. The processor of subsystem A reads pc7a. If it is high, it sends new data on port A. Thus sets pc7a to low and begins a new round of data transmission.

4.Conclusion

We have discussed the construction of system prototype with the platform that demonstrates great flexibility. But there is limitation in the communication speed between the subsystems. Further research should be on improving the communication speed between the subsystems.

References

1.The MCS-80/85 Family User's Manual, Intel Corporation, Intel Corporation, 1983
2.John Uffenbeck, Microcomputers and microprocessors — The 8080,8085, and Z-80 Programming, Interfacing, and Troubleshooting, 1985
3.ALTERA APEX data book and it's application, 2000
4.Quartus Software User's Guide, 1999