The Design and Implementation of an Embedded Virtual Experiment System Kernel

Kelian Peng, Fuan Wen, Lu Song

Beijing University of Posts and Telecommunications, Beijng Key Laboratory of Network System and Network Culture Beijing, China

pengkelian@126.com, fawen@bupt.edu.cn, songlu310@foxmail.com

Abstract - With the rapid development of embedded technology, embedded simulation technology also followed it. In this paper, with single chip microcomputer of 8051 as an example, it introduces an implementation process of 8051 simulation kernel by using discrete event model and object-oriented modeling approach, then a comprehensive test of the 8051 system kernel was tested to verify the reasonableness of the core model**.**

Index Terms - embedded systems, device simulation, instruction emulation, 8051 MCU

1. Introduction

With the arrival of the post-PC era, embedded technology has been rapid development, it resulting in a huge talent gap. The course of embedded system is a very practical course, which focus on the study of embedded system theory, it is also requires students to improve the practical application development ability through do a lot of experiments. However, due to lack of funds and continuous upgrading of embedded devices, some teaching institutions unable to provide students with good experimental equipment and environment, especially for distance education institutions. Therefore, building a virtual experiment system embedded courses become a good solution. Based on the in-depth research of China virtual experiment system and embedded simulation technology, on the basis of previous studies and the combination of innovative embedded systems and object-oriented modelling features, this paper design an embedded system, which the interface and kernel is segregated ,with a high degree of scalability, good reusability between modules. This paper has a detailed description for the design and implementation process of core and instruction emulation.

2. Research Situation

Virtual experiments has an important significance, and has broad prospects to promote the development of distance education, so there are many universities and research institutions to carry out virtual experiments in the system research and development work, resulting in a number of good virtual experiment system results[1], which greatly promoted the development of virtual experimental techniques.

As some famous universities and research institutions to carry out research in virtual experiment early, now, many developed countries which have virtual experiment system is widely used it in education and teaching activities. Virtual experiment is already very popular, especially the United States that first proposed the concept of virtual experiment Room, which has been in a very leading position in the field of research and application[2]. For instance, SPICE (Simulation Program with Integrated Circuit Emphasis) and IBIS(Input and Output Buffer Information Specification) is a circuit simulation for circuit description and a language and an emulator for detecting circuit connections and functional integrity and predicting the behavior of the circuit, which was originally used to help students understand the performance of integrated circuits, mainly used in analog circuit simulation and mixed signal circuits.

3. The Key Technologies of System

The 8051 embedded simulation system use the object-oriented thought to realize all the functional modules with object-oriented programming language. In the process of instruction simulation, it often involves the operation of the data field. The use of lower-level programming language is easier to handle and more efficient [3]. Considering these factors, we use the C++ programming language for the part of implementation embedded virtual experiment system kernel.

The 8051 processor device is the core of the system equipment. In addition to the common attributes and behaviors of common device, it needs to complete the core mission: execute user program instruction. Thus, the processor device class needs to inherit from a base class device and add its own unique instruction simulation.

4. The Realization of Processor and Instructions

A. The Architecture of Processor

Processor module is the core of embedded virtual experiment system, it achieves the goal that can execute user programs through simulation instruction system. According to the design of the processor, we can safely draw the universal model based on the explanation of the processor. From this model, we can build the processor class structure, as shown in Figure 1.

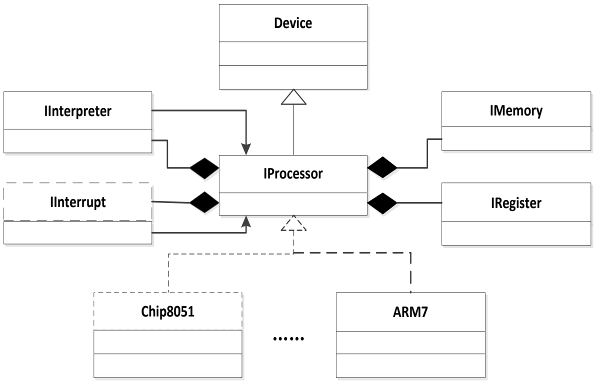


Fig. 1 UML diagrams of processor structure

Remove the methods inherited from the Device class, Characteristic method of IProcessor interface as shown in Table I.

TABLE I The Characteristic Method of IProcessor Interface

The name of method	Describe
LoadProgram	To load an executable program file
GetRegister	According to the register names for register
GetMemory	According to the storage place for storage
GetInterrupt	To get the interrupt controller

DoLogic method of class Device used to perform the logic functions. Its core logic function is executing user programs for the processor. Therefore, interface IProcessor inherits from the class Device, the DoLogic method is used to perform user program instructions. For example, class 8051 processor of chip Chip8051, its implementation of DoLogic method is as follows:

```
void Chip8051::DoLogic()
{
   if (this->pins_[39]->GetState() == HIGH_LEVEL)
   //If the chip has access to electricity
   {
      pInterpreter_->Interpret();
      //An instruction explained
   …
}
```

B. *The Command Interpreter*

Through the analysis of the role of command interpreter and process of instructions, the command interpreter model is obtained, the two models is tightly linked, because the command interpreter is relying on a series of instructions model object to complete the user program interpretation and implementation[4]. Command interpretation system is obtained by these two models in the implementation of the class structure, as shown in Figure 2.

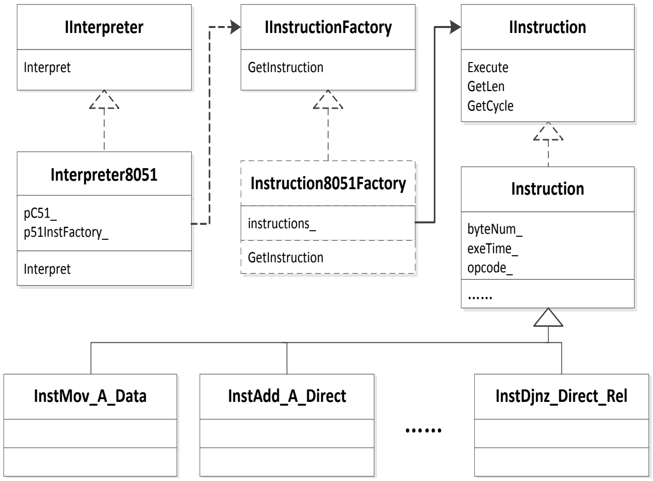


Fig.2 The class structure UML diagram of instruction interpretation system

We can take the instruction class InstMov_A_Data as an example. This is the introduction of the execution process of instruction. The Execute method implementation code is as follows:

```
void InstMov_A_Data::Execute(IProcessor *pProc)
{
IRegister *pA   = pProc->GetRegister("A");
IRegister *pPc  = pProc->GetRegister("PC");
IMemory *pRom=pProc->GetMemory("ROM");
// the address of immediate number in Rom
addr8_type dataAddr = (addr8_type)(pPc->GetValue() +
1);
// From Rom to retrieve the Data value
byte_type data = pRom->GetByte(dataAddr);
pA->SetValue(data);
// Make the PC value to the next instruction address
pPc->SetValue(pPc->GetValue() + byteNum_);
}
```

The implementation of function is to take immediate digital to the accumulator for class MOV A, #data, after instructions interpreter determined by analysis of the current user program for MOV. A, #data, then it call the Execute method of class InstMov_A_Data. Firstly, find the program counter object, to get the address of immediate digital in Rom for dataAddr, on the basis of the immediate digital dataAddr, it can get the data as a new value to the accumulator, the final set instruction counter into the first address of an instruction.

5. Realization of Timers and Counters

Microcontroller of MCS-51 has two internal 16-bit programmable timers and counter-T0 and T1.They can be used as a timer mode or counter mode.T0 consists of TH0 and TL0 counter and T1 consists of TH1 and TL1 counter.

When work on the timer mode, the system work by counting the machine cycle that every machine cycle timer plus one to achieve timing. Therefore, the system oscillator frequency directly affects regular time. If the oscillator frequency is 12MHZ, the intervals of timer will plus one.

When work on the count mode, the system work by counting the negative transition for P3.4 P3.5 pin (1 → 0) .The external input is sampled during S5P2 of every machine cycle, when the sample values is high in this machine cycle and the next machine cycle is low, the counter is incremented by one. Therefore, two machine cycles needed to identify an effective transition, the maximum count frequency is 1/24 of oscillator frequency.

Special Function Register TMOD controls the timer and counter. The high bit is to set timer T1, the low bit is to set timer T0. As shown in Figure3

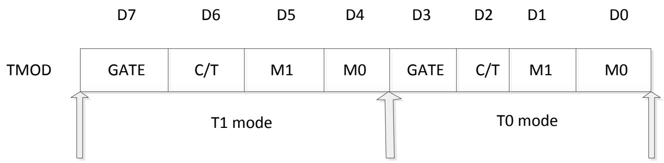


Fig.3 The Control format of Timer Mode

TCON register is used to control the timer count and interrupt flag. As shown in Figure 4.

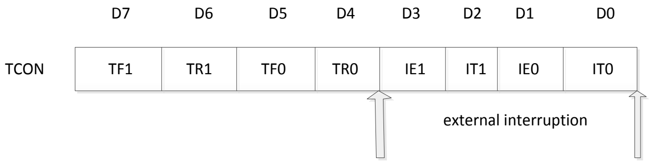


Fig. 4 Timing control register data format

Write a program to control these two registers can control the timer runs.

In the system, we need to simulate corresponding register of timers and counters, and add the appropriate resources in the 8051 class, the position of the timer class in the system as described below.

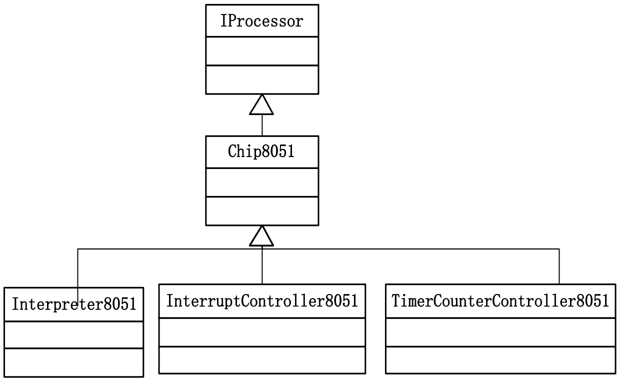


Fig. 5 Relationship of the MCU class and Timer class

After each instruction execution time for 8051 class, the timer will determine whether the class is set to open, if open, it will follow the pattern set by the timer count for processing, if not open, it will ignore the existence of the timer, the timer core algorithm is as follows as follows:

```
void TimerCounterController8051::CountByMode(int no, time_type step)
{
    if (no == 0)
    {
        pTh  = pC51_->GetRegister("TH0");
        PTl  = pC51_->GetRegister("TL0");
        mode = mode_.first;
        pos  = 5;
    }
        …
    switch (mode)
    {
     case 0:
        tmp = (pTh->GetValue() << 5) + (PTl->GetValue() & 0x1F) + step;
        if (tmp > 0x1FFF)
        {
         pC51_->GetRegister("TCON")->SetBit(pos, 1);
        }
        pTh->SetValue(tmp >> 5);
        PTl->SetValue((PTl->GetValue() & 0xE0) + (tmp & 0x1F));
        break;
        case 1:
            …
}
```

User set timers and counters for the initial value and the operating mode in the program, the each machine cycle of microcontroller is processing, we can achieve the simulation timer and counter functions.

6. An Test of Embedded System Kernel

Kernel virtual experiment system provide the core logic function for the whole of virtual experiment system, it must be stable enough to ensure that the virtual experiment system run correctly and effectively, so it is necessary to strictly test the function.

According to the experiment of test cases, it is overall test system kernel for the matrix keyboard scanning and key value of 8051, then analysis the result of the test.

The following experimental design for timer is used to verify system availability.

Experiment shows that: according to the Figure 6 to set up the experimental circuit.P1.0 port placed LED lights. The experimental procedure is that the level of P1.0 is inverted after 30 cycles, the corresponds led blinking lights appear for situation.

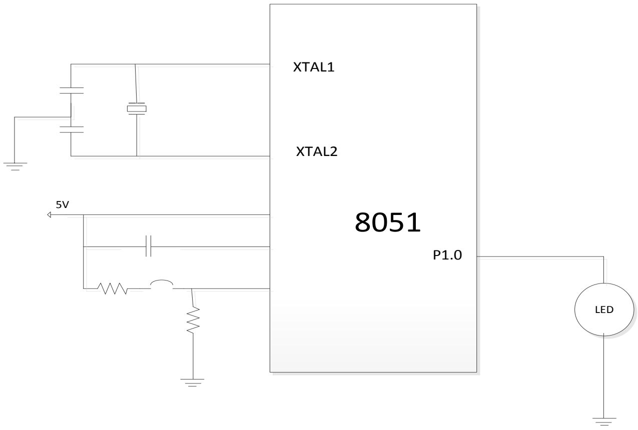
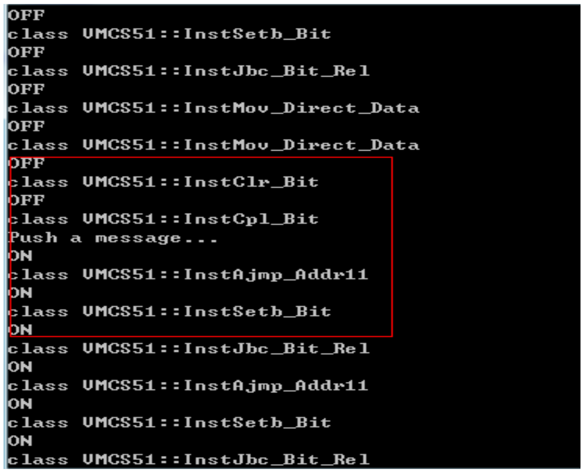


Fig. 6 Matrix keyboard scanning and key display circuit diagram

The results are shown in Figure 7.



```
OFF
class VMCS51::InstSetb_Bit
OFF
class VMCS51::InstJbc_Bit_Rel
OFF
class VMCS51::InstMov_Direct_Data
OFF
class VMCS51::InstMov_Direct_Data
OFF
class VMCS51::InstClr_Bit
OFF
class VMCS51::InstCpl_Bit
Push a message...
ON
class VMCS51::InstAjmp_Addr11
ON
class VMCS51::InstSetb_Bit
ON
class VMCS51::InstJbc_Bit_Rel
ON
class VMCS51::InstAjmp_Addr11
ON
class VMCS51::InstSetb_Bit
ON
class VMCS51::InstJbc_Bit_Rel
```

Fig.7 Keyboard operation screenshots of test

The Fig.7 shows that when the value of the timer that set by the program reaches zero, the microcontroller sends a signal to reverse the P1.0 port and use a jump instruction to return to the initial value of the timer and resume timing. P1.0 appeared state “off” and “on”. It is namely the desired effect.

7. Conclusions

As the development of the modern experiment teaching mode, virtual experiment technology can well solve the problem of lack of funding. The 8051 single-chip microcomputer is the focus of the teaching experiment, and the part of 8051 kernel is the core simulation arithmetic logic for the embedded virtual experiment system [5]. On the basis of in-depth study to object-oriented modeling technology and the discrete event system simulation technology, this paper has carried on the simulation modeling to the embedded system kernel and mainly introduced the virtual equipment system, command simulation system. In a virtual device system, it established abstract universal equipment model based on practical equipment, the connection attributes and behaviors. The equipment manager makes it easy to create and manage on them. In instruction simulation system, according to the way an explanation to the instruction system simulation, it set up the general structure of the processor and instructions by the processor's instruction interpreter components to complete simulation of the instruction execution process. The simulation system is a good way to finish some of the basic experiments, and has great practical significance.

References

[1] Wen Zhichen, Zonghuiwong. Principle and design of embedded system, Tsinghua university press, 2011(in Chinese)

[2] Yang Ning, Zhou Yulin. Based and the application of embedded system ,Tsinghua university press, 2012(in Chinese)

[3] Aihu Zhang. The design and implementation of embedded virtual experiment system kernel [dissertation], Beijing university of posts and telecommunications, 2012(in Chinese)

[4] Di Suo. The study of the embedded kernel simulation technology of the virtual experiment system [dissertation], Beijing university of posts and telecommunications, 2012(in Chinese)

[5] Donald O. Pederson, L. Nagel, SPICE-A Simulation Program with Integrated Circuits Emphasis, Sixteenth Midwest Symposium on Circuit Theory, Waterloo, Canada, Paper VI-1, April 12, 1973.