

Design and Implementation of a General-Purpose MCU Simulation Software

Yao Chen¹, Yichun Huang^{1,2}, Jie Lu¹, and Hua Lu¹

¹ School of Electronics and Information Engineering, Tongji University, Shanghai, China

² Shanghai Hanzhi Information Technologies Co., Ltd., Shanghai, China

chenyao@tongji.edu.cn

Abstract. In this paper, a general-purpose MCU simulation software is introduced. The software utilizes an open architecture based on plug-in mechanism. It can simulate various MCU and peripherals in high accuracy and support for diversified compilers through different plug-in modules. In this way, a MCU simulation environment can be established on PC, it can provide powerful and convenient debugging and analysis function than in the real hardware. Actual applications especially in education area proved that the software can satisfy the multiple MCU needs and possess excellent flexible and practical features. Furthermore, we can develop new plug-ins to expand the software function in the future.

Keywords: MCU, simulation, plug-in, peripheral.

1 Introduction

MCU is a major branch of computer especially for embedded systems. In MCU area, there are more families and types than other processors. There are also many different develop software environments for different MCU types common provided by the manufacturers, they can only support one families MCU usually. Some software even hasn't simulation function that can't run and debug programs on PC without the target hardware.

Simulators are power tools for computer learning, research and development; simulation techniques play an important role in the computer area now. There are many researches and implementations of simulation software for different processors, platforms and systems, for example, ArmSim for ARM processors[1], MOOSE for PowerPC[2], PTLSim for x86 processors[3], Simics-a functional complete system simulation software[4], SimOS-a complete computer system simulation software from Stanford University[5], SimSoc-a full system simulation software for embedded systems from Tsinghua University[6], a hardware/software co-simulation system[7], a synchronization profiler for hybrid full system simulation platform[8], SkyEye-a full system simulator which can simulate many kinds of Processors[9] and Proteus VSM (Virtual System Modeling)-a famous co-simulation of complete microcontroller combines mixed mode SPICE circuit based designs[10].

In education area, students may learn different MCU families in some courses. Shifting between different develop software environments is difficult and inefficient, a multiple MCU simulator maybe a good choice, but it isn't easy to find a simulation software satisfy our needs. For this reason, we developed a universal MCU simulation software, it's an open source software registered in SourceForge.

2 Software Architecture

As a general-purpose software, the architecture must flexible enough to accommodate different MCU simulation needs in a uniform GUI. The famous music player Winamp is a good reference, it can support various music format, multi output mode and extend other functions easily depend on its distinctive plug-in architecture [11].

Our MCU simulation software also uses the plug-in architecture to develop the different function modules. The main program is the integrated development environment (IDE) of MCU simulator, it provides these functions such as project management, source code edit, memory and registers check & modify, code debugging etc. The unique function can be realized in these corresponding plug-in modules. The architecture of the simulation software is shown in Fig.1.

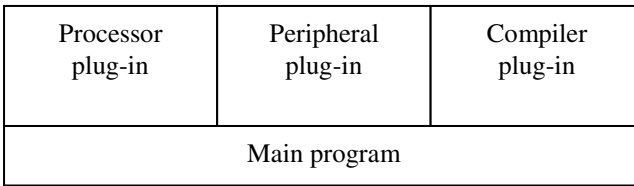


Fig. 1. Architecture of the simulation software

The plug-in module involves three types; they are processor plug-in, peripheral plug-in and compiler plug-in. We can realize the support of various processors, peripherals and compilers through writing different plug-in modules. The main program is written in Visual C++ and the plug-in modules are written in MASM32, Visual C++ and LCC-win32.

2.1 Processor Plug-in

The processor plug-in is used to simulate the target processor. The processor simulation includes instruction set simulation, memories simulation, timers' simulation, UART simulation, I/O simulation of the chip. The software now can support Intel 8051, Freescale 68HC08, Microchip PIC12, AVR and ARM7TDMI, etc., more processors are developing.

The processor plug-in can not only act as a processor simulator, it can also communicate with the real hardware emulator. These operations in the simulation software interface will reflect to the hardware emulator.

2.2 Peripheral Plug-in

The peripheral plug-in used to realize peripheral simulation. The peripheral simulation includes external status simulation, external logic simulation and external I/O simulation, etc. It has the same structure as the processor plug-in although the target they simulation are different.

The peripheral plug-in and processor plug-in can both simulate the pins of the device. Users can connect these pins freely between processor and peripheral. The peripheral plug-in not always simulate the actual peripherals, it can also simulate the virtual peripherals you want, for example, a pin state viewer or a monitor of CPU internal state.

2.3 Compiler Plug-in

Normally, different processors have different instruction sets, so we need different compilers translate the source code into machine code. The compiler plug-in is used to unify the difference between diverse compilers and call the compiler as the same mode. The structure and programming of the compiler plug-in is easier than the processor's. Its work mainly includes generating compiler parameters according the users needs, calling compiler and transfer the compiling results to the main program.

Only one compiler plug-in can be set active at one time, all compiling work will be completed by the active compiler this time. The compiler plug-in may include one or several window interface for user's compiler configuration. The MCU simulation software now has several compiler plug-ins for different MCU support various languages including assembler, C and C++.

3 Functions Implementation

As a general-purpose MCU simulator, the software simulation targets include 8 bit, 16 bit or 32 bit microcontrollers, various peripherals or even actual embedded systems. In addition, it can also connect and drive the real equipments through dedicated plug-ins, for example, experimental apparatus and development kits, realize visual debugging.

The software functions mainly consist of three modules: equipment simulation, code management/compiling and system debugging.

3.1 Equipment Simulation

The equipment simulation system is used to simulate the target equipment, complete data transfer and synchronization between these simulated equipments. The equipment simulation system includes processor simulation, peripheral simulation and redirection of serial equipment. The software completes the processor and peripheral simulation by calling corresponding plug-ins, and synchronizes the actions between equipments through controlling the calling time. These are the key techniques ensure efficient and accurate system simulation. The serial equipment redirection is used to redirect the serial I/O of simulated equipments to main machine's various equipments, for example, screen, disk file, named pipes, actual serial ports and so on.

3.2 Code Management/Compiling

The code management/compiling system is used for source codes management and compiling. The software uses the project concept as popular IDE in source codes

management. Besides source files, library files, project description and documents, a project file also includes compiler configuration, hardware selection and configuration (processors, peripherals and serial equipment) information. The compiling function is completed by calling the compiler through the compiler plug-in.

3.3 System Debugging

The debugging system supply debugging and analysis function for source codes. It is tightly integrated with equipment simulation system. The debugging function includes step, trace, run and breakpoint, etc. Fig.2. gives the basic flowchart of trace running.

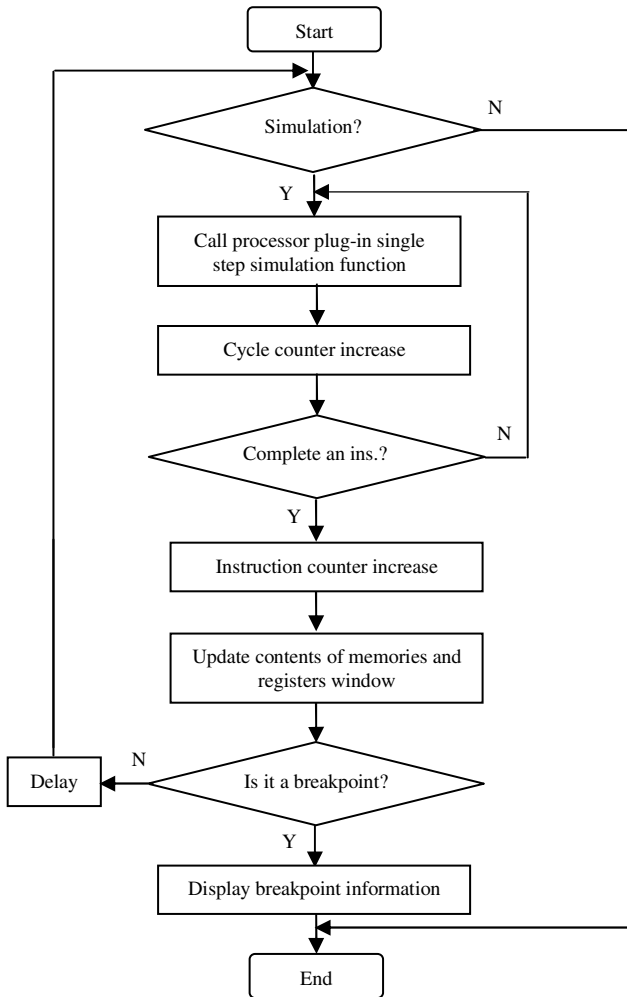


Fig. 2. Flowchart of trace running

All display content of memories, registers and ports will be updated during step and trace debugging. The speed of trace debugging can be set between 0 to 500 ms.

The full-speed running process similar to tracing mode but not update the any display content and execute the next instruction immediately.

4 Application and Features

4.1 Application

The universal MCU simulation software can be used in many areas such as software development, architecture design, software testing and hardware verification. It is also a powerful tool for embedded learning, students can make hardware configuration and program design freely under the simulation software environment. The MCU simulation software already used in MCU teaching practice in Tongji University and good results have been achieved. Fig.3 gives an example of a M68HC908GP32 MCU simulation with an eight 7-segment LED display.

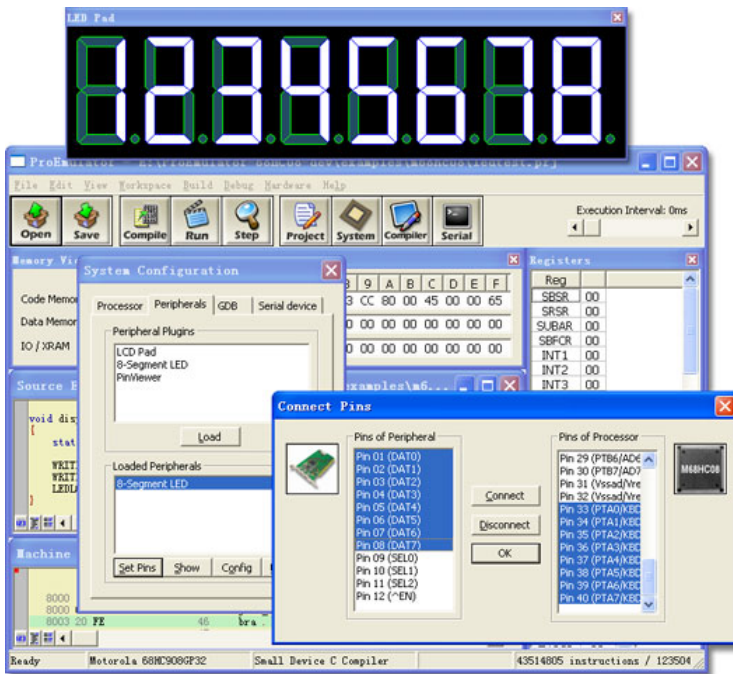


Fig. 3. MCU simulation with an eight 7-segment LED display

4.2 Features

Some features comparison with two famous simulation software Proteus and SkyEye, as shown in Table 1.

Table 1. Features compare with Proteus and SkyEye

Features	Proteus	SkyEye	Our software
Is it an open source software?	No	Yes	Yes
Is it a commercial software?	Yes	No	No
Processor support	PIC, 8051, MSP430, AVR, HC11, ARM7/LPC2000 and Basic Stamp Processors, etc.	ARM, Coldfire, MIPS, PowerPC, SPARC, x86 and Blackfin DSP, etc.	8051, M68HC08, PIC12, AVR and ARM7, etc.
Peripheral support	Yes	Yes	Yes

5 Conclusions

This paper introduces the design and implementation of an open structure general-purpose MCU simulation software based on the plug-in mechanism. The simulator now has realized many MCU series simulation such as Intel 8051, Freescale M68HC08, Microchip PIC12, AVR and ARM7TDMI, some external equipment simulation(e.g. keyboard, LED, LCD) and support various compiler including assembler, C and C++ languages. It can provide high precision simulation of multiple types MCU and peripherals in system clock and pin signal level.

In the simulation environment, programs can run and debug in the same way they are running in the real hardware. It can provide powerful and convenient debugging and analysis function than the real hardware to promote the embedded developing process. Through the applications especially in teaching practice proved that the simulation software has good practicability and flexibility. As a general-purpose MCU simulator, it has an open architecture which can realize simulation rapidly for new devices or equipments. We can develop more types of processors, peripherals and compiler plug-ins to expand the simulator function in the future.

References

1. Wan, H., Gao, X.P.: ArmSim: A Complete System Simulation Environment of the ARM Embedded System. In: Fifth International Conference on Information Technology: New Generations, Queensland, pp. 1261–1262 (2008)
2. Robinson, L., Whisenhunt, G.: A PowerPC Platform Full System Simulation-from The MOOSE up. In: IEEE International Performance, Computing and Communications Conferences, p. 458. IEEE Press, Texas (1999)
3. Yourst, M.T.: PTLsim: A Cycle Accurate Full System x86-64 Microarchitectural Simulator. In: IEEE International Symposium on Performance Analysis of Systems & Software, pp. 23–34. IEEE Press, California (2007)

4. Magnusson, P.S., Christensson, M., Eskilson, J., Forsgren, D., Hållberg, G., Högberg, J., Larsson, F., Moestedt, A., Werner, B.: Simics: A full system simulation platform. *Computer*, 50–58 (2002)
5. Rosenblum, M., Herrod, S.A., Witchel, E., Gupta, A.: Complete Computer System Simulation: The SimOS Approach. *Parallel & Distributed Technology: Systems & Applications* 3, 34–43 (1995)
6. Helmstetter, C., Joloboff, V., Xiao, H.: SimSoC: A full System Simulation Software for Embedded Systems. In: *IEEE International Workshop on Open-source Software for Scientific Computation (OSSC)*, pp. 49–55. IEEE Press, Guiyang (2009)
7. Ishikawa, M., McCune, D.J., Saikal, G., Oho, S.: CPU Model-Based Hardware/Software Co-design, Co-simulation and Analysis Technology for Real-Time Embedded Control Systems. In: *13th IEEE Real Time and Embedded Technology and Applications Symposium*, pp. 3–11. IEEE Press, Washington States (2007)
8. Chen, K.C., Chen, C.H.: A Synchronization Profiler for Hybrid Full System Simulation Platform. In: *2010 International SoC Design Conference (ISOCC)*, pp. 91–94. IEEE Press, Incheon (2010)
9. SkyEye: A Very Fast Full System Simulator,
http://sourceforge.net/projects/skyeye/?_test=b
10. Proteus Virtual System Modeling (VSM) co-simulation software,
http://www.labcenter.com/products/vsm_overview.cfm
11. Wikimedia Foundation, Inc.: Plug-in (computing). *Wikipedia free encyclopedia*,
[http://en.wikipedia.org/wiki/Plug-in_\(computing\)](http://en.wikipedia.org/wiki/Plug-in_(computing))