

A Teaching Tool for Designing a Programmable System on Chip Using Hardware Description Language

Sung-Tsun Shih¹, Chin-Ming Hsu², Chian-Yi Chao³

¹Department of Electronic Engineering, Cheng Shiu University, Kaohsiung, Taiwan

²Department of Information Technology, Kao Yuan University, Kaohsiung, Taiwan

³Department of, Electronic Engineering, Kao Yuan University, Kaohsiung, Taiwan
stshih@csu.edu.tw

Abstract: This paper aims to design a simulator which is specifically used to design a programmable system on chip (PSOC) for increasing teaching quality and enhancing students' learning effects. The proposed teaching tool could effectively help students to learn MCS51 microcontroller's principle as well as several input/output peripheral applications in one semester. There are four functions provided by the simulator: (1) using PSOC hardware description language designing an application; (2) capable of simulating designed circuits; (3) allowing compiling the hardware description codes; and (4) capable of checking the data sheets of the MCS51 chip and I/O peripheral devices. The designed teaching tool based on Java programming language can support the characteristics of being easily understood and memorized, thereby enhancing students' interests and system integration concepts while taking the microcontroller principle and practice course.

[Sung-Tsun Shih, Chin-Ming Hsu, Chian-Yi Chao. **A Teaching Tool for Designing a Programmable System on Chip Using Hardware Description Language.** Life Science Journal. 2012; 9(2):1-6] (ISSN:1097-8135).
<http://www.lifesciencesite.com>.

Keywords: Teaching tool, and hardware description language.

1. Introduction

In the era of e-education, an institution has developed various learning education such as distance learning, on-line learning, and adult learning in their certificate programs, undergraduate programs, and graduate programs. Generally, distance learning is heavily based on skill development and sometimes requires laboratory works which need to interface with computers to facilitate a hands-on learning approach [1]; online learning allows students study at home or anywhere they like via computers and the Internet [2]; adult learning is typically for specific age group above the legal age limits, which is related to their daily life, goals, and career needs [3]. Of all these types of learning education, if the learners did not learn the material well initially, they will not retain it well. Therefore, the need of increasing the availability of particular educational resources in various parts of curriculum materials such as motivational training materials for learners is with great importance to help learners retaining what the program delivers to them. In order to achieve this goal, Information technology (IT) in learning education plays an increasingly central role in institutions; specifically, it is placing new demands on expertise [4].

With the rapid developed automated industry in microcontroller related products, developing the computer simulation software becomes more important and gets more attention in recent year because a simulator can provide users to test the

product before making a practical product, thereby cutting the wrong probability. Hence, microcontroller control is one of the mainstream subjects for people developing an automation product in the field of Electrical Engineering, Mechanical Engineering, Computer Science, and Information Technology. In an educational institution, when a professor teaches microcomputer related course, he/she would introduce its main principle and applications in order to unite the theory and the practice together and increase students designing a practical product capability effectively. Currently, Keil 8051 development tool is nowadays the most popular software teaching tool applied in the microcontroller related course, which allows users to use either assembly or C programming language to make a specific control system.

In the reviewing teaching tool related technology, Davies [5] investigated the opinions on using the simulator as teaching tool in the class for the students majoring in Materials Engineering. After interviewing some students with 2000 questionnaire's survey results, the conclusions are the operation process of the existing simulation software is too complex to understand as well as the students don't have enough time practicing the simulation. References [6-8] developed simulators as teaching tools which can be applied on the Internet. Yang and Altj [6] developed a distribution simulator using the Internet as the transmission media, which allows users to transmit information to end-learner via the

Internet and remote control the end-learner's learning progress. This simulator supports the advantage of learning everywhere. Huang [7] developed an interactive teaching platform for students majoring in psychology. The developed platform named PHYSIO has the advantage for students studying on-line. Esquembre [8], an engineering department teacher, developed a graph-base with simple interface simulator as the teaching tool. Literatures [9-11] are the references using JAVA programming language to build a software simulator. Whaley [9] proposed a virtual machine named Joeq which is specifically designed for scholars working on their research. The Joeq system mainly contains four kinds of virtual simulations, including automatic control distribution, time interval calculation, filtering trash information, and compiler. Kreuzinger et al. [10] developed a special virtual microcontroller named Komodo which is mainly designed for the on-line services of multithreading and multiplexing processes which replace the subroutine functions of the interrupt services. Sánchez and Manzoni [11] developed the ad hoc wireless network simulator, named ANEJOS which is compatible with any router. Moreover, Khare et al. [12] developed a simulator named V-ST to analyze and evaluate the system chip hardware structure. Murray [13] proposed a medical consulting and decision making teaching tool.

MCS51 microcontroller is one of the most popular single chips taught in an academic in Taiwan. It allows users to use complex set of assembly instructions or C high level programming language to make some peripheral control applications, including LED, DIP Switch, Timer, interrupt, spot matrix, LCM, motor and UART communication. However, there are two drawbacks existing in the current software teaching tool. They are: (1) the syntax of codes provided in Keil without hardware structure characteristics is hard for students to learn well and remember easily; students are therefore lack of a picture of designed control system and brainstorm stimulation on developing a practical product and (2) one semester generally covers only fundamental principle and software simulation, which lacks the coordination of the theory and practice, thereby decreasing students' interest in learning microcontroller applications.

In order to overcome two weaknesses described above, this paper refers to the advantages of microcontroller-based teaching tools [14-16] and develops a simulator which is specifically used to design a programmable system on chip (PSOC) for increasing teaching quality and enhancing students' learning effects. The proposed teaching tool could effectively help students to learn MCS51 microcontroller's principle as well as several

input/output peripheral applications in one semester. There are four functions provided by the simulator, including using PSOC hardware description language designing an application, simulating designed circuits, compiling the hardware description codes, and checking the data sheets of the MCS51 chip and I/O peripheral devices. The designed teaching tool designed by Java programming language supports users with easy understanding and memorizing and enhances their interests while taking the microcontroller principle and practice course.

2. MCS51 Teaching Simulation Tool

Figure 1 illustrates the structure of the developed MCS51 teaching simulation tool. A user-friendly interface is built between the user and the menu of the proposed teaching tool. As given in Figure 1, there are five functions provided in the system. They are programming MCS51 hardware description language, compiling the codes, I/O simulation, schematic circuit output, and database. The roles of each unit in the menu are described as follows.

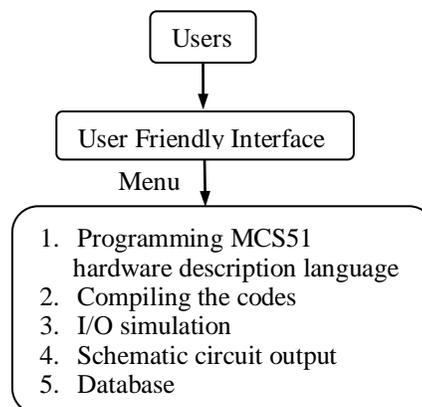


Figure 1. Structure of developed MCS51 teaching tool.

2.1. Programming MCS51 Hardware Description Language

The proposed MCS51 hardware description language (MCS51 HDL) refers to the syntax of current Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) and C programming language. The syntax of MCS51 HDL contains mainly three types of instruction sets as well as some reserved words such as int, print, input, output and so on. One instruction set is related to the mathematic operations, including addition, subtraction, multiplication, division. Another instruction set is related to logical operations, including And, OR, NAND, NOR, XOR, and so on. Loop control is another type of the instruction set, which contains if, if-else, switch-case, for, while, and so on.

Figure 2(a) illustrates the structure of MCS51 HDL program, which basically includes five parts: initialization, input device, output device, hardware pin configuration, and main codes. Initialization defines the fundamental pin configuration to activate an MCS51 chip. As shown in Figure 2(a), pin 9, 12, 19, 20, 31, 40 are the fundamental pin configuration to activate an MCS51. Input and output (I/O) device define what kinds of I/O peripheral devices being used in the application. In this study, fundamental peripheral devices are such as LED, 7-segment buzzer, DIP switch, push button, and 4X4 keypad. Hardware defines the pin configuration of the application project. Main codes define the functions of the application. Figure 2(b) shows an example of MCS51 HDL program, which only uses 8-LED output devices. The example runs 8-LED left shift and right shift, where $i=1$ and $i=0x40$ define two different initial conditions.

2.2. Compiling the Codes

Figure 3 illustrates the process of compiling MCS51 HDL codes, which includes two passes. One pass called pass 1 aims to check the syntax of MCS51 HDL codes, which will output the location and the name of the instruction if the syntax of the instruction is wrong. The other pass called pass2 aims to generate corresponding machine code (Hex file) for users downloading the MCS51 HDL codes on to the chip.

```

BEGIN
{
  DelayX1ms ( unsigned int ) ;
  int i , loop ;
  be(1)
  {
    i=1 ;
    rrr(loop=0;loop<8;loop++)
    {
      P0 = ~i ;
      DelayX1ms(5000) ;
    }
    i = i << 1 ;
  }
  i=0x40 ;
  rrr(loop=0;loop<7;loop++)
  {
    P0 = ~i ;
    DelayX1ms(5000) ;
    i = i >> 1 ;
  }
}
    
```

(b)

Figure 2. (a)Structure of MCS51 HDL program; (b) example of MCS51 HDL program.

<pre> Initialization { PIN40 VCC(+5) PIN31 VCC(+5) PIN20 VSS(GND) PIN18-19 HZ PIN9 RESET } Input Device { } Output Device { } Hardware { } BEGIN { } </pre>	<pre> Initialization { PIN40 VCC(+5) PIN31 VCC(+5) PIN20 VSS(GND) PIN18-19 HZ PIN9 RESET } Input Device { } Output Device { LED:0..7 ; } Hardware { } BEGIN { P0[0,7] <= LED[0,7] ; } </pre>
---	---

(a)

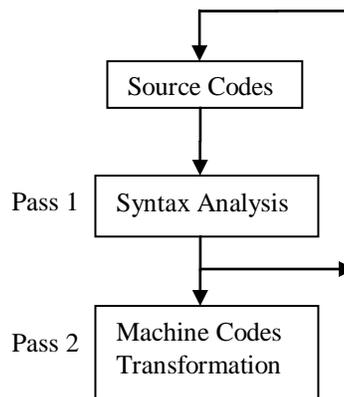


Figure 3. Process of compiling MCS51 HDL.

2.3. I/O Simulation

A user is allowed to define the input signal on specific pin or port; the system runs MCS51 HDL codes and outputs the results on the corresponding port.

2.4. Schematic Circuit Output

The system runs MCS51 HDL codes and outputs the corresponding schematic circuit for users having a picture of designed control system. Figure 4 is the schematic circuit output of MCS51 HDL example given in Figure 2(b).

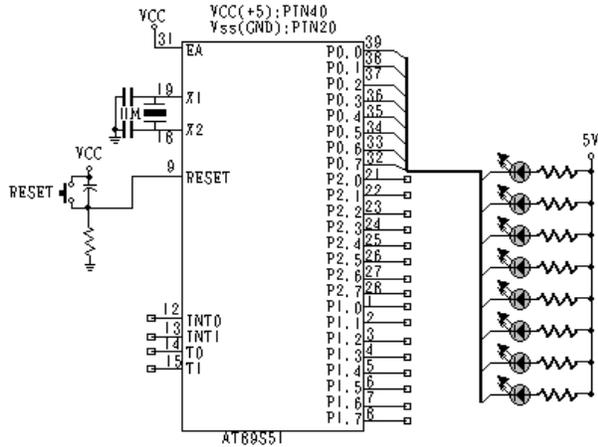


Figure 4. Schematic circuit output of MCS51 HDL example given in Figure 2(b).

2.5. Database

The system provides MCS51 datasheet, fundamental peripheral I/O devices' specifications, including LED, 7-segment, buzzer, switch, keypads etc., and the syntax of MCS51 programming codes. The datasheets of the mathematical operation, logical operation, loop control and reserved word instruction sets are listed in Table 1 to Table 4, respectively.

Table 1. Mathematical operation instruction sets

Sym	Function	Ex.	Explanation
+	Addition	a+b	If a=4,b=2 then a+b=6
-	Subtraction	a-b	If a=4,b=2 then a-b=2
*	Multipli.	a*b	If a=4,b=2 then a*b=8
/	Division	a/b	If a=4,b=2 then a/b=2
%	Modulation	a%b	If a=4,b=2 then a%b=0
+1	Increase	a+1	If a=4, after a+1, a=5
-1	Decrease	a-1	If a=4, after a-1, a= 3

Table 2. Logical operation instruction sets

Symbol	Function	Example
&	AND	F=A&B
#	OR	F= A # B
!	NOT	F = A' = !A
&'	NAND	F = (A & B)' = A&' B
#'	NOR	F = (A # B)' = A#' B

Table 3. Loop control instruction sets

Symbol	Function
if(){}	if
if(){ }else{ }	if-else
switch(){case: }	switch-case
for(){ }	for(){ }
while(){ }	while(){ }
do{ } while()	do{ } while()

Symbol	Example
if(){}	if(A==0){B=0;}
if(){ }else{ }	if(A==0){B=0;}else{C=0;}
switch(){case: }	switch(i){case1:out='A';break;
for(){ }	for(i=0;i<256;i++){A=i}
while(){ }	while(i<10){A=i;i++;}
do{ } while()	do{ A=i;i++; }while(i<10);B=33;

Table 4. Reserved words.

Symbol	Explanation
IN:	Input
OUT:	Output
Title:	Title
Value:	Define Value of Device

3. Experimental Results

The system is implemented by using JAVA programming language. A user can free download the software via <http://www.java.com/en/download/index.jsp> webpage. Figure 5 shows the interface of the proposed teaching simulation tool, which includes seven items: File, Chip device, Input/Output devices, Compile, Simulation, and Database interface functions. Each function of Figure 5 is introduced as follows

- (1) The title of the proposed teaching simulation tool.

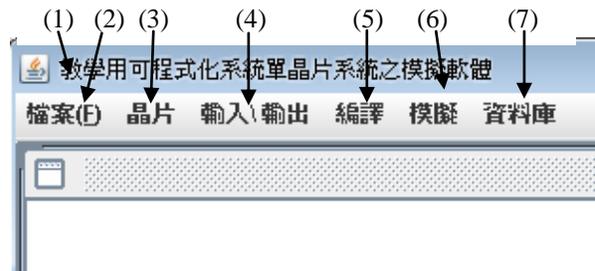


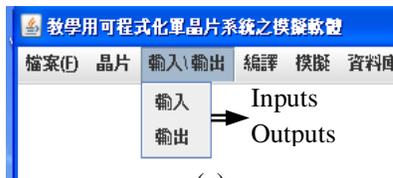
Figure 5. Interface of proposed teaching simulation tool.

- (2) File: Including creating new file, open a file, save file, and exit four sub-functions as shown in Figure 6.



Figure 6. Interface of file sub-functions.

- (3) Chip: It means MCS51 microchip. The contents of this icon illustrate the basic circuitry connections for an MCS51 chip in the working condition.
- (4) Input/Output: It shows input/output devices. As shown in Figure 7, the Inputs devices include push button (SW), 8-pin dip switch (DIP8), 4-pin dip switch (DIP4), and 4 by 4 keyboard. The Outputs devices include color red, green, and blue LEDs, 7 segment display (7SE), and buzzer (SP). A user can check the specifications of input/output devices by clicking the icon on the bottom of the sub-functions.



(a)

符號	說明
SW	按鈕開關
DIP8	8支腳指撥開關
DIP4	4支腳指撥開關
KB4*4	4*4數字鍵盤

SW DIP8 DIP4 KB4*4

(b)

符號	說明
LEDR	紅色LED
LEDG	綠色LED
LEDB	藍色LED
7SE	七段顯示器
SP	蜂鳴器

LEDR LEDG LEDB 7SE
紅色LED 綠色LED 藍色LED 七段顯示器

SP

(c)

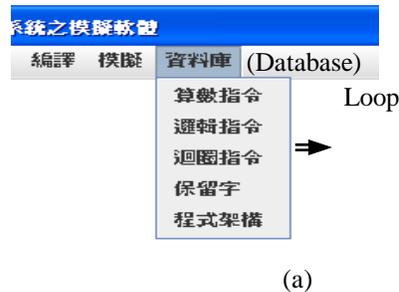
Figure 7. Interface of input/output sub-functions.

- (5) Compile: Compiles MCS51 HDL codes.
- (6) Simulation: The system runs MCS51 HDL codes and outputs the results on the corresponding I/O port. Figure 8 gives the example of parallel port 1 status.



Figure 8. Parallel port 1 status.

- (7) Database: This function includes the introduction of three instruction sets (mathematical logical and loop control operations), reserved words, and program structure, as illustrated in Figure 9. A user can check the specifications of input/output devices by clicking the icon on the bottom of the sub-functions.



(a)

運算元	符號	範例	
加法	ADD	+	F = A + B
減法	SUB	-	F = A - B
乘法	MUL	*	F = A * B
除法	DIV	/	F = A / B
遞增	++	+1	F = A+1
遞減	--	-1	F = A-1

(b)

運算元	符號	範例
AND	&	F = A & B
OR	#	F = A # B
NOT	!	F = A' = !A
NAND	&'	F = (A & B)' = A&'B
NOR	#'	F = (A # B)' = A#'B
XOR	##	F = A ## B = A \$ B

(c)

Figure 9. Interface of mathematical and logical Operations of database Sub-functions.

4. Conclusions

This study proposes a simulator which is specifically used to design a programmable system on chip (PSOC) for increasing teaching quality and enhancing students' learning effects. The main contribution of the proposed teaching tool is to demonstrate the advantages of ahead-of-time compilation design concept on developing an automatic control system. Compared to the existing method, the proposed system supports three distinctive advantages. (1) The proposed teaching tool reduces the testing time for developing an application project, thereby enhancing the efficiency of circuit design. (2) The students have another choice using hardware structure based programming language to design their applications, which has characteristics of easy to learn and to remember. This advantage may have them brainstorm stimulation on developing a practical product and cultivate their competition ability. (3) The system supports the output schematic circuit function, which increases the coordination of the theory and practice, thereby increasing students' interest in learning microcontroller applications.

This work can be extended to build a virtual microcontroller laboratory by using the Internet technology for various learning education such as distance learning, on-line learning, and adult learning in their certificate programs, undergraduate programs, and graduate programs.

Corresponding Author:

Dr. Sung-Tsun Shih
Department of Electronic Engineering,
Cheng Shiu University,
Kaohsiung, Taiwan, R.O.C.
E-mail: stshih@csu.edu.tw

References

- 1 M. Abarashi. Improving education through distance education and online learning. *Nature and Science*, 2011;9(8): 55-58.
- 2 Z. Behzadi, A. Ghaffari. Characteristics of online education and traditional education. *Life Science Journal*, 2011;8(3):54-58.
- 3 A. Badragheh, M. Abedi. Assessing of ways to strengthen adult education. *Life Science Journal*, 2011;8(2):514-519.
- 4 M. R. Rezaei, M. Nazarpour, A. Emami. Challenges of information and communication technology (ICT) in education. *Life Science Journal*, 2011;8(2):595-598.
- 5 C. H. J. Davies. Student engagement with simulations: a case study. *Computers and Education*, 2002; 39(3): 271-282.
- 6 S. H. Yang, J. L. Alty. Development of a distributed simulator for control experiments through the Internet. *Future Generation Computer Systems*, 2002; 18(5): 595-611.
- 7 C. Huang. Changing learning with new interactive and media-rich instruction environments: virtual labs case study report. *Computerized Medical Imaging and Graphics*, 2003; 27(2-3): 157-164.
- 8 F. Esquembre. Easy Java Simulations: a software tool to create scientific simulations in Java. *Computer Physics Comm.*, 2004; 156(2): 199-204.
- 9 J. Whaley. Joeq: A virtual machine and compiler infrastructure. *Science of Computer Programming*, 2005; 57(3): 339-356.
- 10 J. Kreuzinger, U. Brinkschulte, M. Pfeffer, S. Uhrig, T. Ungerer. Real-time event-handling and scheduling on a multithreaded Java microcontroller. *Microprocessors and Microsystems*, 2003; 27(1): 19-31.
- 11 M. Sánchez, P. Manzoni. ANEJOS: a Java based simulator for ad hoc networks. *Future Generation Computer Systems*, 2001; 17(5): 573-583.
- 12 A. Khare, A. Halambi, N. Savoiiu, P. Grun, N. Dutt, A. Nicolau. V-SAT: A visual specification and analysis tool for system-on-chip exploration. *Journal of System Architecture*, 2001; 263-275.
- 13 C. Murray, M. J. Grant, M. L. Howarth, J. Leigh. The use of simulation as a teaching and learning approach to support practice learning. *Nurse Education in Practice*, 2008; 8(1): 5-8.
- 14 S. Y. Huang. A Virtual Laboratory for Microcomputer I/O Control Experiments. Master thesis, Computer Engineering Dept., Fengjia University, Taiwan, 2002.
- 15 H. T. Chang. Simulation Apply in Programming Curriculum. Master thesis. Mathematics Education Dept., National Taipei Education University, Taiwan, 2007.
- 16 N. L. Wang. A Study on the Teaching Performance of Computer Simulation Software Application in the Single-Chip Lab Program for the General Senior High School Information Technology Program. Industrial Education Dept., National Taiwan Teaching University, Taiwan, 2006.

12/30/2011