The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM) 1(2): 68-80
The Society of Digital Information and Wireless Communications, Apr - 2015 (ISSN: 2410-0439)

FPGA and System on Chip Education for Embedded Engineer

Kenta Shimizu*‡||, Yoshifumi Ohtsuka*‡¶, Yuichi Okuyama†§**, Naohiko Shimizu†‡††
School of Information and Telecommunication Engineering Tokai University*
Graduate School of Computer Science and Engineering University of Aizu†
2-3-23, Takanawa, Minatoku, Tokyo, 108-0074, Japan‡
Tsuruga, Ikki-machi, Aizu-Wakamatsu, Fukushima 965-8580, Japan§
4bjnm007@mail.tokai-u.jp¶, 4bjnm012@mail.tokai-u.jp||,
okuyama@u-aizu.ac.jp**, nshimizu@keyaki.cc.u-tokai.ac.jp††

ABSTRACT

We present the author's education cases. As an introductory education, we applied hardware development with Linux and FPGA to the 3rd and 4th grade students newly assigned to our laboratory in 2012 and 2013. As the next step, we carried out a Study Camp for developing an Apple-I System in 2014. Furthermore, we applied hardware development with Linux, System on Chip (SoC) of the Apple-I System and Field Programmable Gate Array (FPGA) to 3rd grade students newly assigned to our laboratory in 2014. This paper reports four case studies of the education conducted in 2012, 2013 and 2014.

KEYWORDS

Education, Hardware, FPGA, System on Chip, Apple-I System

1 INTRODUCTION

Embedded systems include microprocessors to execute specific functions. The hardware and software of the systems control several peripherals collaboratively to provide useful functions. The rapid and worldwide dissemination of recent embedded systems inevitably requires expertise both in hardware and software of engineers. In recent years, engineers had to develop embedded systems in high performance and a short period. The industry of the embedded system reported issues of systems with high functionality and high performance developed in a short period [1]. Embedded engineers were shorthanded [1]. For these reasons, embedded engineers are required for the contribution of society. Embedded engineers are necessary some skills that are as follows[2]:

- Knowledge of hardware
- Knowledge of software
- knowledge of how to use operating system
- knowledge of Development tools
- Communication ability

Authors have carried out educations of embedded systems to students assigned or related to author's laboratory. Several articles report conventional university educations dealing with following applications:

- Video games for social and economic applications for programming education [3].
- Android application and distributed processing of the application for FPGA in 2014 [4].
- CPU with pipeline for FPGA in computer architecture course [5].
- Video games and an image decoding on an FPGA [6].

FPGAs were used in these studies to rewrite their hardware. These educations are carried out with the small number of participants or the large number of participants in the long term [3], [4], [5], [6]. In contrast, we carried out education to develop a hardware in a short term. Table.1 illustrate our education in recent years to develop applications on FPGA. The goal of our education is to let participants get the knowledges of software and the hardware required to embedded engineers.

Table 1. Our education in 2012, 2013, and 2014

Year	Application(s)	Working style	Term
2012	Games or SDRAM controller	Group	11 days
2013	Games	Individual	10 days
2014	Apple-I system	Group	4 days
	Games	Group	19 days

After the development, participants answered to questionnaires in 2012, 2013 and 2014. The remainder of the paper is structured as follows: Chapter 2 presents the education cases in 2012 and 2013. Next, chapter 3 presents an education of a study camp with other universities in 2014. Then, Chapter 4 presents the education cases in 2014. In following, chapter 5 illustrates the results of educations in 2012, 2013 and 2014. Chapter 6 illustrates the questionnaire results in 2012, 2013 and 2014. Chapter 7 describes the discussion of educations. Finally, chapter 8 provides a conclusion.

2 EXAMPLE OF INTRODUCTORY EDUCATION IN 2012 AND 2013

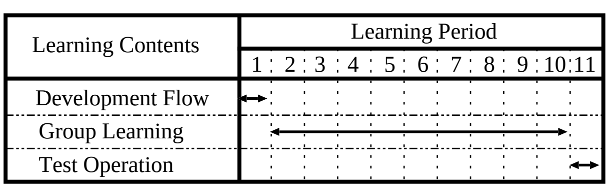

Figure 1. Learning schedule in 2012 [12]

2.1 Introductory Education in 2012

We carried out 11 days of hardware education, as shown in Table.2 [10]. Table.3 illustrates the participants experiences in 2012. The 3rd and 4th grade newly assigned students had some knowledge of C language. However, they had never participated in a short term hardware development; therefore, we carried out hardware development education in 2012. Figure.1 shows their learning schedule, which is as follows:

1) On the first day, we educated the students on hardware design flow using Linux.
2) From the second day on, the participants freely developed each challenge.
3) On the final day, an upper-class student checked the problems of each group.

When the participants were faced with issues, they asked for help from the upper-class students. The participants of Group 2 already had knowledge of hardware. For this reason, they developed the SDRAM controller because they wanted to learn the FPGA, and thought SDRAM could be widely applicable and useful.

Table 2. Introductory education in 2012[12]

Group ID	Number of persons	Grade	Application(s)
1	3 persons	3rd grade students	Hockey game
2	1 person	3rd grade students	SDRAM controller
3	3 persons	3rd grade students	Breakout game
4	2 persons	4th grade students	Snake game

Table 3. Participants experiences in 2012 [12]

Group ID	Linux experience	C language experience	Hardware experience
1	No	Yes	No
1	No	Yes	No
1	No	yes	No
2	Yes	Yes	Yes
3	No	Yes	No
3	No	Yes	No
3	No	Yes	No
4	Yes	Yes	No
4	Yes	Yes	No

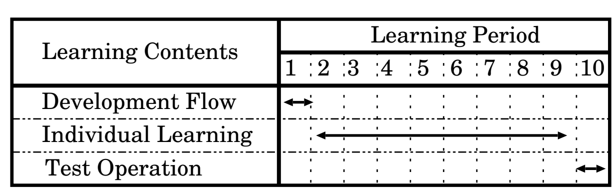

Figure 2. Learning schedule in 2013 [12]

2.2 Introductory Education in 2013

We carried out 10 days of hardware education in Table.4 [11]. Table.5 illustrates the participants experiences in 2013. The 3rd grade newly-assigned students had some knowledge of C language. However, they had never participated in short term hardware development; therefore, we carried out hardware development education in 2013. Figure.2 shows their learning schedule, which is as follows:

1) On the first day, we educated the students on hardware design flow using Linux.
2) From the second day on, the participants freely developed each challenge.
3) On the final day, an upper-class student checked the problems of each group.

When the participants were faced with issues, they asked for help from the upper-class students. Some participants already had knowledge of hardware. The participants could make games; and some participants were successful in using VGA controllers.

Table 4. Introductory education in 2013 [12]

Person ID	Grade	Development condition	Application(s)
1	3rd grade students	Reuse	Breakout game
2	3rd grade students	Reuse	Hockey game
3	3rd grade students	Fresh	Gobang game
4	3rd grade students	Fresh	Trump game

Table 5. Participants experiences in 2013 [12]

person ID	Linux experience	C language experience	Hardware experience
1	No	Yes	No
2	No	Yes	No
3	Yes	Yes	Yes
4	Yes	Yes	Yes

2.3 Hardware Design Flow using Linux Operating System

We educated a hardware design flow using Linux Operating System on the first day. The hardware design flow are as follows:

1) Explain a specifications of an FPGA
   We explained to the participants a specification of an FPGA.
2) How to use the Linux
   We taught them how to use Linux. They were difficult to an operation of command line at first. However, they become very proficient.
3) Method of convert NSL to Verilog-HDL
   We taught them to convert NSL to Verilog-HDL. They used NSL because NSL is similar to C language. The C language of knowledge is necessary for NSL.
4) How to use the QuartusII
   We taught them the how to use QuartusII (e.g. Project Creation, compile, pin assignment and forwarding method).
5) we ran a sample programs of NSL with them.
   They ran a sample programs of NSL with us. After that, We explained the sample programs.
6) Explain a VGA controller
   We explained a modular structure of the VGA controller.

2.4 Educational Material:VGA Controller

We provided a VGA controller in Figure.3 which is written with NSL [7]. The VGA controller has two modules and It can be expanded up. The participants learn VGA controller and FPGA of specification.

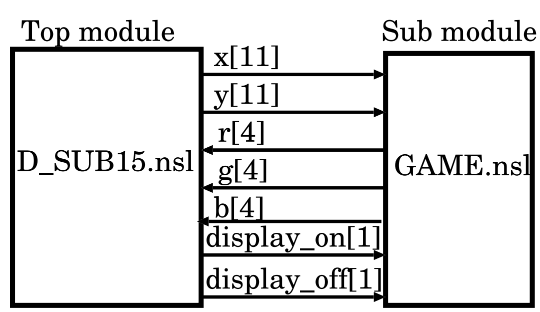

Figure 3. Educational material:VGA controller [12]

2.5 Development Process

Development processes of the game are as follows:

1) The participants made the specification, the screen configuration diagram (e.g. Figure.4) and the block diagram (e.g. Figure.5).
2) They wrote the game module in hardware description language.
3) They developed some games. Figure.6 illustrates System requirements of the game.

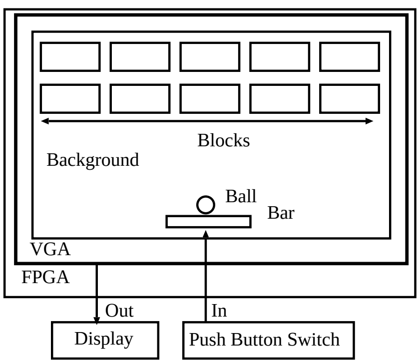

Figure 4. Screen structure [12]

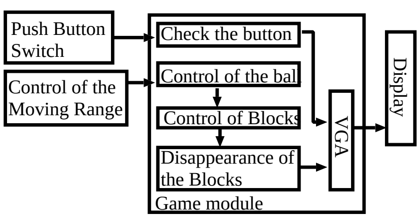

Figure 5. Block diagram [12]

2.6 Educational Method

Figure.7 illustrates an author's educational method. When the participants were faced with issues, they asked some questions to the upper-class student.

2.7 When the Participants were Faced with Issues

The participants issues are as fallows:

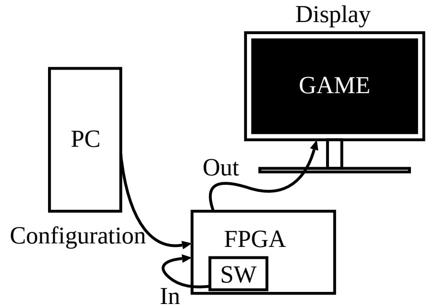

Figure 6. System requirements of the game [12]

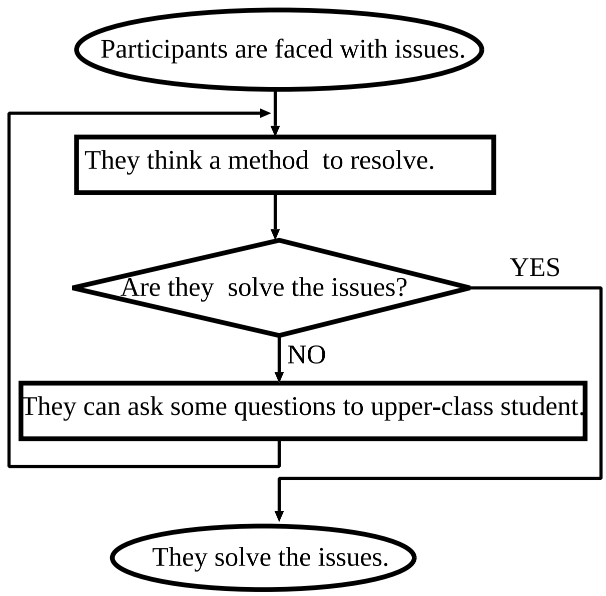

Figure 7. Educational method [12]

- The participants in Tables.3 and 5 took time to learn for Linux and NSL. This is because they didn't have enough experience.
- When the participants in Table.3 carried out learning in group work, they didn't gather to our laboratory.
- The participants in Table.3 could not share information, because they didn't gather to our laboratory.
- The participants in Tables.3 and 5 were difficult to adjust the schedule.
- The participants in Tables.3 and 5 could not communicate with group members or upper-class student.

2.8 There is Method of Solution when the participants were Faced with Issues

When the participants were faced with issues, as illustrated in Figure.7, they asked some questions to the upper-class student.

3 EXAMPLE OF EDUCATION OF STUDY CAMP IN 2014

3.1 Apple-I System

The Apple-I System is an educational system for programmers. Its system can write some program to memory, and those written programs can run. The Apple-I System has a monitor program which has 16 functions. The monitor program can write Value to specified memory. Furthermore, the program can read memory's value from a specified memory. Figure.8 illustrates the system requirements of the Apple-I System.

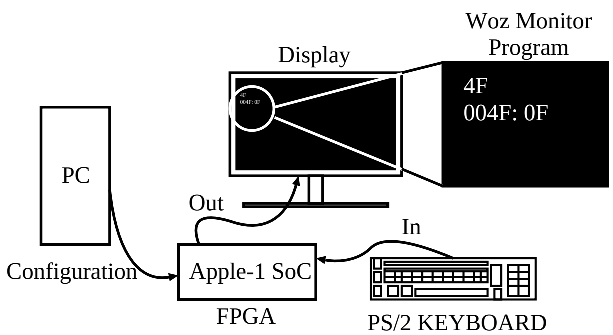

Figure 8. System requirements of Apple-I System [12]

3.2 Educational Material:Apple-I System on Chip (AISoC)

Figure.9 illustrates a Structure of a SoC educational material. The SoC of an educational Material was provided by our teachers in which it has 6502CPU, ROM(4KB), RAM(8KB) and Peripherals. The educational material on runs on FPGA(DE2-115).

Its actions are as follows:

1) It performs Read/Write to data-memory.
2) It stores byte-code about assembler instruction of 6502CPU.
3) It executes byte-code as functions.

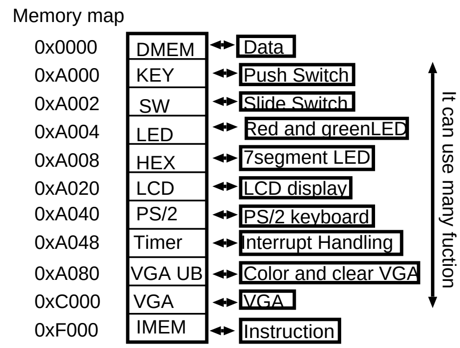

Figure 9. Structure of educational material:AISoC [12]

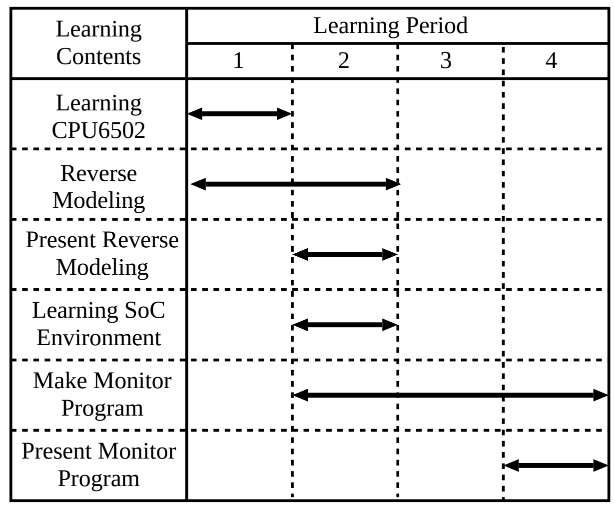

Figure 10. Learning schedule in 2014 [12]

3.3 Study Camp in 2014

We carried out a Study Camp to develop the Apple-I System with Tokai University and the University of Aizu. In this paper, we present examples of an educational program in Tokai University. Table.6 illustrates the participants in 2014. They developed a monitor program with C language before a Study Camp. Figure.10 illustrates their learning schedule. The schedule was as follows:

1) On the first day, they received a lecture of the cpu6502 [8] by our teachers. Then, they read the manual of cpu6502 and the Apple-I operation manual [9], and carried out a reverse modeling.
2) On the second day, they continued to perform the reverse modeling, and announced

Table 6. The participants experiences in 2014 [12]

Person ID	Grade	UNIX or Linux experience	C language experience	Hardware experience
1	1st grade students	No	Yes	No
2	1st grade students	No	Yes	No
3	2nd grade students	No	Yes	No
4	3rd grade students	No	Yes	No
5	3rd grade students	Yes	Yes	Yes
6	3rd grade students	Yes	Yes	Yes
7	3rd grade students	Yes	Yes	Yes
8	3rd grade students	Yes	Yes	Yes
9	4th grade students	Yes	Yes	Yes
10	4th grade students	Yes	Yes	Yes
11	4th grade students	Yes	Yes	Yes
12	2nd year graduate student	Yes	Yes	Yes

the results of the reverse modeling. Then, they received a description of the SoC for the development on FPGA. Next, they started to create a monitor program.

3) On the third day, they continued to create the monitor program.
4) On the fourth day, they continued to create the monitor program. Then, they announced the results of the monitor program.

3.4 Development Process

The participants developed the Apple-I System. The development processes were as follows:

1) They read the manual of cpu6502 and the assembly code of the monitor program of Apple-I operation manual. Furthermore, they carried out the reverse modeling.
2) They created the monitor program based on the reverse modeling with the C language on FPGA.

3.5 Educational Method

3.5.1 Reverse modeling

The participants carried out the reverse modeling in a group work. The reverse modeling involves converting from an assembly language code to C language.

1) The participants read the manual of cpu6502 and the assembly code of the monitor program of the Apple-I operation manual.
2) They converted the assembly language to flowcharts or UML.
3) They converted the flowcharts or UML to C language.

3.5.2 Monitor Program

The participants created the monitor program in a group work.

1) The participants showed some words using the VGA module.
2) They performed character entry using PS/2 module.
3) They carried out the Apple-I System with the character entry and C language of the reverse modeling.

This is difficult; therefore, they exchanged opinions with other groups.

4 EXAMPLE OF INTRODUCTORY EDUCATION IN 2014

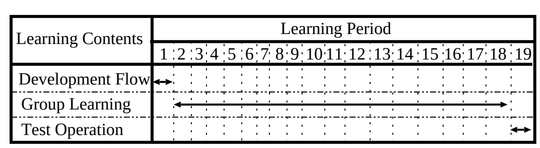

Figure 11. Learning schedule in 2014

4.1 Introductory Education in 2014

We carried out 19 days of hardware education for newly students who was assigned to our laboratory. Educational participants designed and developed several games in a table.7 using AISoC on FPGA. The 3rd grade newly assigned students have some knowledge of C language and some knowledge of a little hardware. Furthermore, they have never been a short term hardware development, therefore we carried out hardware development education in 2014.

Table 7. Introductory education in 2014

Group ID	Number of persons	Grade	Application(s)
1	2 persons	3rd grade students	Badminton game.
2	3 persons	3rd grade students	Typing game.

Figure.11 shows their learning schedule which is as follows:
1) On the first day, we educated the hardware design flow using AISoC.
2) From the second day, the participants freely developed each challenges.
3) On the final day, an upper-class student checked the problem of each groups.

During the development, the participants exchanged views with another group. They also asked some questions to the upper-class student when the participants are faced with issues.

Before and after the development, the participants answered the questionnaire. Table.8 illustrates the questionnaire items and Table.9 illustrates the answer method of Questionnaires. Furthermore, Figures.12, 13 and14 illustrate the participant's experience.

4.2 Hardware design flow using Linux Operating System in 2014

We educated a hardware design flow using Linux Operating System on the first day. The hardware design flow are as follows:

1) Explain a specifications of an FPGA
We explained to the participants a specification of an FPGA.
2) Explain specifications of an AISoC
We explained to the participants specifications of an AISoC.

Table 8. Questionnaire items before our lesson

I. Do you understand C language with following items?
1. Character's I/O 2. Variable 3. Operator symbol 4. Control statement 5. Function 6. Array 7. Pointer 8. Struct
II. Do you know some knowledge of computer with following items?
1. Difference between windows and Linux 2. Experience of Linux 3. About CPU 4. How to use memory 5. Experience of FPGA
III. Do you have knowledge in hardware description language?
1. Experience of Verilog-HDL 2. Difference between C language and NSL

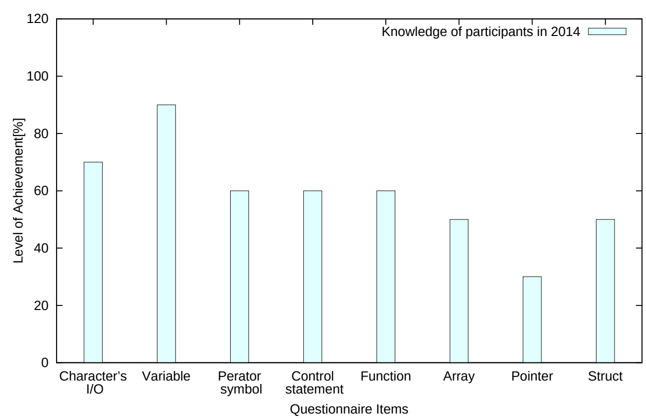

Figure 12. Knowledge of C language

3) How to use the Linux
We taught them how to use Linux. They had difficult to use the command line at first. However, they become very proficient.
4) We ran a sample programs of NSL with them.
The upper-class student ran a sample program of AISoC and after that, we explained the sample programs to them.

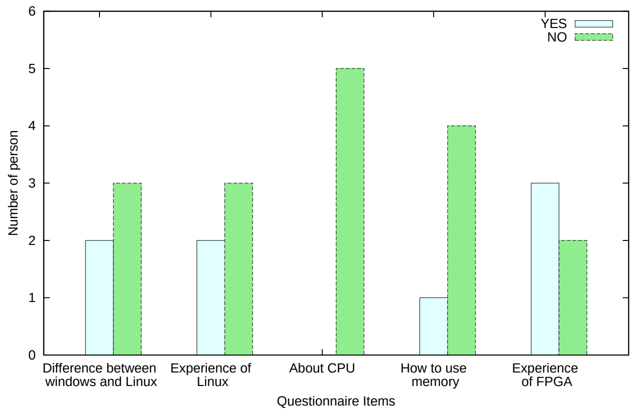

Figure 13. Knowledge of computer

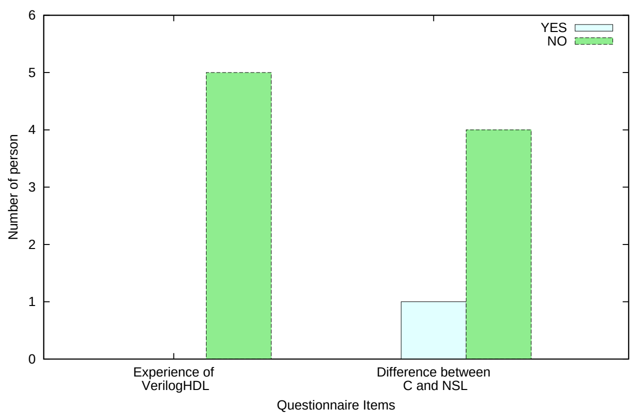

Figure 14. Knowledge of hardware

4.3 Development process in 2014

4.3.1 Participants in Group1

Group 1 member had never been development experience using AISoC. Therefore, they defined specification with upper-class student. After define the specification, they decided order of implementation of function. When group 1 member developing its game, there were an upper-class student near them to answer some of the questions. They developed all function with all members of the group and sharing information in their group.

Table 9. Answer method of questionnaires

I. Do you understand C language with following items?
1. I don't think so. (0%) 2. Neither. (50%) 3. I think so. (100%)
II. Do you know some knowledge of computer with following items?
1. Yes (0%) 2. I No (100%)
III. Do you have knowledge in hardware description language?
1. Yes (0%) 2. I No (100%)

4.3.2 Participants in Group2

Group 2 member had been development experience using AISoC. Therefore, they defined specification by themselves. Furthermore, they increased a working efficiency when they applied role sharing for the development.

5 RESULTS

5.1 Implementation's Results of Introductory Education in 2012

Many participants were able to implement the applications for 11 days, as shown in Table.2. However, the participants of Group 4 were not able to implement the applications because it was difficult to adjust the schedule.

5.2 Implementation's Results of Introductory Education in 2013

Many participants were able to implement the applications in Table.4. However, the participants of Group 4 were not able to implement the applications because it was difficult to adjust the schedule.

5.3 Development Scale of Introductory Education in 2012 and 2013

As illustrated in Table.10, there were code lineages of the development scale in 2012 and 2013. For motion synthesis, the Verilog-HDL code blended with the NSL code with NSLcore

[7]. The Verilog-HDL code runs on FPGA. The participants in 2013 developed games in individuals. For that reason, there was much flexibility in the design. In the results, the number of lines of code was increased.

Table 10. Code linage of development scale in 2012 and 2013 [12]

Group or Person	Developed NSL linage	Blended Verilog-HDL linage
Group 1	662	3557
Group 2	702	4234
Group 3	544	4842
Group 4	342	2958
Person 1	1423	6232
Person 2	659	3837
person 3	784	4832
person 4	1164	6675

5.4 Working Hours in 2012 and 2013

Table.11 shows the working hours for both 2012 and 2013. The working hours were the total number of hours worked by each individual. The participants of Group 4 found it difficult to adjust the schedule; therefore, they were not able to develop part of the functions.

Table 11. Working hours in 2012 and 2013 [12]

Group or person	Working hours
Group 1	12.5
Group 2	35.0
Group 3	12.8
Group 4	3.5
person 1	10.0
person 2	20.0
person 3	15.0
person 4	15.0

5.5 Implementation's Results from Study Camp in 2014

The participants carried out reverse modeling and created the monitor program of the Apple-I System. After that, they announced the results of deliverables with the group. Two groups out of seven could make the monitor program in two universities.

5.6 Implementation's Results of Introductory Education in 2014

The participants in Group 1 made the badminton game on an FPGA. They could implement a display drawing and an operation using keyboard input. However, they could not implement a collision detection of a shuttlecock and other functions because they did not have enough time and had to control ROM below 4 KB. Group 2 made the typing game. The participants in Group 2 had knowledge of AISoC; therefore, they could implement all functions of specification. Furthermore, they tried to increase the game's function but they could not.

6 QUESTIONNAIRE RESULTS

6.1 Questionnaire Results of Introductory Education in 2012 and 2013

After the development, the participants answered the questionnaires. Table.12 illustrates Common Questionnaires. Table.13 illustrates an answer method of the questionnaires.

Table 12. Common questionnaires [12]

The teaching effectiveness
1. could you use NSL language?
2. could you use Linux?
3. could you use QuartusII?
4. Did you understand hardware design?
The one's problem-solving powers
5. Did you get one's problem-solving powers?
6. Did you get one's assessment powers of the situation?
7. Did you get communication ability?
8. Did you get logical thinking ability?

Table 13. Answer method of questionnaires [12]

Value of Evaluate	Evaluate Contents (%)
1	I don't think so. (0%)
2	I don't think that it says either so. (25%)
3	Neither. (50%)
4	I think that is says either so. (75%)
5	I think so. (100%)

Figure.15 illustrates the results of teaching effectiveness. Figure.16 illustrates the results of one's problem-solving powers. An analysis of questionnaire results is as follows:

The International Journal of E-Learning and Educational Technologies in the Digital Media (IJEETDM) 1(2): 68-80
The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2410-0439)

6.1.1 Teaching effectiveness

As illustrated in Figure.15, the participants in 2012 increased in knowledge of hardware design and how to use Linux. The purpose of this class was to create the application. They enjoyed that a lot and we obtained good results. However, the hardware knowledge of the participants in 2013 was low, because we came up short in explaining the FPGA and the hardware design.

6.1.2 One's problem-solving powers

The participants in 2012 developed games with the individuals. As illustrated in Figure.16, the participants in 2012 could not communicate with upper-class students because they could not adjust their schedules to coincide with the upper-class students schedules.

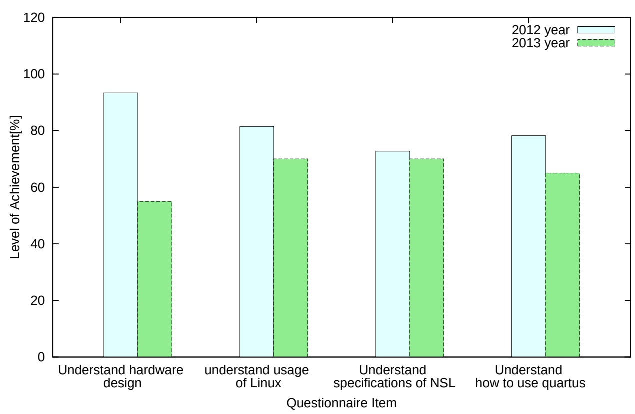

Figure 15. The Result of Teaching Effectiveness [12]

6.2 Questionnaire Results of Study Camp

After the development, we gave a questionnaire on the Study Camp to our university students. The participants filled out the questionnaires during a free writing time after our lesson. Items from the questionnaires are as follows:

- About reverse modeling
- About monitor program
- About group work

The questionnaire results are as follows:

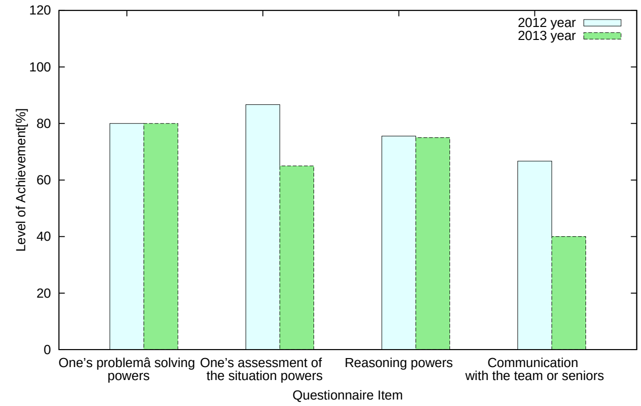

Figure 16. The Result of One's Problem-Solving Powers [12]

6.2.1 Questionnaire results of reverse modeling

- It was difficult to understand assembly code and the instructions for cpu 6502 for many students, because many participants did not have assembly code knowledge.
- It took a lot of time, but the participants could eventually understand assembly code and instructions for cpu 6502.

6.2.2 Questionnaire results of monitor program

- It was difficult to design hardware with the rules of material.
- It was difficult for many students because the participants had to control ROM below 4 KB.
- It was difficult for many students because the participants understood an admission control of PS/2.
- Many participants could not develop part or all of the monitor program.
- The participants found it difficult to adjust the times.
- The participants could not share their work.

6.2.3 Questionnaire results in group work

- When a group member wrote the code, other group members wrote the other code. Therefore, they could not share the work.

- They all thought the same thing about group work; they thought it was very efficient.

6.3 Questionnaire Results of Introductory Education in 2014

After the development, the participants filled out the questionnaires of free writing after our lesson. Table.14 illustrates the questionnaire items during free writing time. The questionnaires of free writing are as follows:

6.3.1 About some skills

- All participants answered yes to the statement We can review C language.
- The greater part of participants answered We understand all functions.
- All participants answered We can study the operations of Linux.
- The greater part of participants answered We understand the operating environment of SoC.
- All participants answered We can develop for memory management.
- All participants answered We can understand how to use educational materials.

6.3.2 About communication

- The greater part of participants answered We think appropriate description.
- The greater part of participants answered We can make specifications before implementation.
- The greater part of participants answered We couldn't make the game according to your specifications.
- The greater part of participants answered We can communicate with group members.
- All participants answered We can communicate with upper-class students.

7 DISCUSSION

7.1 Introductory Education in 2012 and 2013

We carried out hardware education in the short term. The goal was to improve understanding and knowledge of hardware. The questionnaire results show an improvement in knowledge of hardware. When their education was completed, the participants had some issues. The participants in 2012 had the issues of scheduling. As a result, they could not develop some functions. They had never developed the hardware. Furthermore, they had to develop in a short period of time. Some of the participants in 2013 gained some knowledge of a portion of the hardware. This is because the codes were reused from 2012. They could not communicate with group members or upper-class students because they could not adjust their schedule. We will take steps to improve upon these issues from now on. To manage a schedule, we will plan the schedule and increase the learning abilities of participants next year.

Table 14. Questionnaire Items after Our lesson

I. Did you understand some of the skills with the following items?
1. Did you understand afresh C language?
2. Did you understand the functions of FPGA?
3. Did you understand how to use Linux?
4. Did you understand the operating environment of SoC?
5. Did you develop the memory management?
6. Did you understand on how to use the development tools?
II. Communication ability
1. Was the prior explanation of upper-class student appropriate?
2. Did you design your game's specification before game's coding?
3. Did you design game according to your specifications?
4. Did you communicate with group member in group work?
5. Did you communicate with upper-class student?

7.2 Study Camp

From the questionnaire results, we can see that the participants had a hard time understanding the instructions of CPU 6502 and assembly codes of the monitor program; however they were able to understand the instructions of CPU 6502 and assembly language. This has been determined from the prior learning from our SoC of educational and the material that was provided by our teachers. The participants should make a monitor program of educational materials, Therefore, they have thought about specifications and have developed the Apple-I System. When they carried out the group work, they had issues. When one group member wrote the code, the other group members wrote the other codes. Therefore, they could not share their work. Some participants could not develop the whole monitor program because they were having difficulties adjusting the times. We would like to teach them how to improve their problem-solving abilities and communicative competence.

7.3 Introductory Education in 2014

We carried out education in 2014 that divided the participants into two groups because of different purposes. The purpose of Group 1, which included the participants who had not yet learned AISoC, was for the participants to learn the basic skills of AISoC. Group 2, which included the participants who had already learned some of those basic skills, improved their knowledge. The participants of Group 1 only made a part of the function. They could not make the game completely. We think the reason for this was a ROM's restriction of AISoC and a lack of time. The participants of Group 2 made the necessary functions of a complete game. We think the reason for success is that they already had made a game in the same environment. According to the questionnaire survey, we learned that the knowledge of C language and the operating skills of Linux of the participants developed. They didn't have an opportunity to learn C language without a class. When they were faced with issues, they learned how to operate Linux by looking at the operations of the upper-class students. The reflection point was that none of the groups could make the game that each other had thought of as a complete game. We think the main reasons for this were the difficulty of the ROM's restriction and the memory interruption.

8 CONCLUSION

We presented the examples of education in 2012, 2013 and 2014. The four cases implemented applications on a FPGA board in a short period of time. The conclusions of the four cases are as follows:

8.1 Introductory Education in 2012 and 2013

The two cases implemented applications on the FPGA board in a short period of time. The participants were faced with some issues during development, but they were able to solve some issues through the advice given by the upper-class students. Some participants do not have the knowledge necessary for advanced preparation, but they were able to create the simple hardware in this case. The participants with an extensive knowledge challenged the hard lessons to increase their skills; they had give good results. We are planning to educate new students.

8.2 Study Camp

The participants from our university participated in a Study Camp for 1st grade students to 2nd year graduate students. They developed the Apple-I System for four days. They were able to understand the overall system and implement the Apple-I System. It was a good experience for the participants because they had a relationship with another university. We would like to carry out the Study Camp in our future work.

8.3 Introductory Education in 2014

Our education in 2014 had two different points. The first point was hardware education using an AISoC environment. For the second point, we divided the participants into two groups for developing games. The participants could develop games with peripherals (VGA, PS/2 keyboard and others) and C language in AISoC environment on FPGA. The peripherals are used through the memory-mapped I/O as shown in Figure.9. We think a simple development environment for the participants provided a little knowledge of the peripherals. Our education in 2014 helped the experienced participants and the inexperienced participants with the AISoC environment. As in Section 7.3, we divided the participants into two groups. Therefore, the participants who had no knowledge increased in knowledge because they were working with educated upper-class students.

REFERENCES

[1] IPA:Information-technology Promotion Agency, Japan, ”The twenty-twelve fiscal year-Investigations on fact finding survey of the software industry-reconnaissance report”, (2013)

[2] Japan Embedded Systems Technology Association, organ Bulletin JASA, ”An embedded technology for a beginning employee”, (2011)

[3] T. Tamama, H. Tomita, ”Programming Education Based on Development of Serious Games”, Information Processing Society of Japan Technical Reports, IPSJ SIGCE, 2009-CE-102(20), 1-5, 2009-12-04

[4] T. Ohkawa, S. Takano, D. Uetake, T. Yokoto ,K. Ootsu, ”HW/SW Hybrid Design Environment for a Cooperative System of Android Application and FPGA”, IEICE technical report, CPSY IPSJ-EMB IPSJ-SLDM DC, 295-300, (2014)

[5] J. H. Lee, S. E. Lee, H. C. Yu, T. Suh, ”Pipelined CPU Design With FPGA in Teaching Computer Architecture”, IEEE TRANSACTIONS ON EDUCATION, VOL. 55, NO. 3, AUGUST 2012

[6] A. Kumar, S. Fernando, R. C Panicker, ”Project-Based Learning in Embedded Systems Education Using an FPGA Platform”, IEEE TRANSACTIONS ON EDUCATION, VOL. 56, NO. 4, NOVEMBER 2013

[7] Overtore Corporation, ”NSL reference manual”, (2011)

[8] MOS TECHNOLOGY, ”MOS6500 MICROCOMPUTER FAMILY PROGRAMING MANUAL”, (1976)

[9] APPLE COMPUTER COMPANY, ”APPLE-1 OPEARATION MANUAL”, (1976)

[10] T. Kazihara, K. Nomoto, C. He, T. Aoyama, R. Tamaki, S. Hirose, Y. Tsubata, H. Akiyama, Y. Ohtsuka, K. Shimizu, N. Shimizu, ”Development Cases in Introductory Education of Our Laboratory”, PARTHENON Technical Society, Vol.38, pp.43-50, 2012

[11] M. Kaziyama, S. Hayakawa, T. Furukawa, S. Kawano, N. Shimizu, ”Development Cases in Introductory Education of Our Laboratory”, PARTHENON Technical Society, Vol.39, pp.47-54, 2013

[12] Y. Ohtsuka, K. Shimizu, Y. Okuyama and N. Shimizu, ”Rapid Method for Embedded Systems Hardware and Software Education”, Proceedings of the International Conference on Computer Science, Computer Engineering, and Education Technologies, Kuala Lumpur, Malaysia,p90-98, (2014)