Postgraduate Coursework Program using Network Processors at Osaka University

Go Hasegawa and Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Phone: +81-6-850-6616, Fax: +81-6-850-6589
E-mail: {hasegawa,murata}@ist.osaka-u.ac.jp

Abstract

We introduce a postgraduate coursework program where students learn to use network processors. This is part of the Master's curriculum for Osaka University's Graduate School of Information Science and Technology (IST). First we briefly explain the Intel IXP1200 network processor and its evaluation kit, the ENP-2505, which are used in this program. We next describe the content of the network-processor coursework program in detail. We also present the results obtained for coursework evaluation, done by students enrolled in the course.

1 Introduction

This paper discusses a postgraduate coursework program where students learn to use the Intel IXP1200 network processor. It is part of the Master's curriculum for Osaka University's Department of Information Networking at the Graduate School of Information Science and Technology (IST).

The IST was established in April 2002 [1]. It combined and reorganized the existing curricula and organizations relating to information and network technologies, which had been distributed through several graduate schools including the Graduate School of Engineering, the Graduate School of Engineering Science, and the Graduate School of Science of Osaka University. The IST has seven IT-related departments for Pure and Applied Mathematics, Information and Physical Sciences, Computer Science, Information Systems Engineering, Information Networking, Multimedia Engineering, and Bioinfomatic Engineering.

The Master's course at the Department of Information Networking involves eleven lectures and two coursework programs each year. The coursework component of the master course is one of the unique features of the IST because these are usually only been required in the undergraduate schools of Japanese universities. We believe that more advanced coursework to expose postgraduate students to IT-related content is necessary at the IST.

The coursework program we discuss is conducted in the first semester (April-September) and its called "Information Networking Exercise I" [2]. The main objective of the coursework is to learn design architecture, implementation, and programming of a network processor. We use the Intel IXP1200 network processor [3] and its evaluation board, the ENP-2505 [4], both of which were donated by the Intel Corporation as part of its IXA University Program [5].

The IXP1200 has a unique CPU instruction set especially designed for network packet processing, fast packet input/output, and header processing. The ENP-2505 also has various devices designed to assist packet processing for the IXP1200, such as three kinds of memory, network interfaces and fast shared buses. Through coursework program-

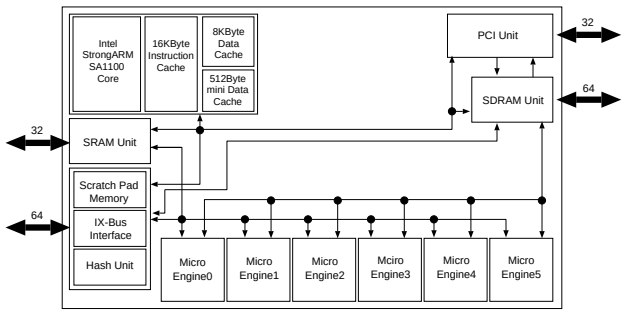

Figure 1: IXP1200 Architecture

Figure 2: ENP-2505: IXP1200 evaluation board

ming exercises with the IXP1200, students can gain experience of network processing, and knowledge of the design and implementation of an actual system as practiced by system vendors. However, general programming exercises for popular computer languages such as C and Java are done by students at the undergraduate schools. We should note that the goal of the postgraduate coursework is not only to improve the programming skills of the students, but also for them to learn about network design, implementation, and evaluation in a hands-on fashion by using the network processor system. The obtained knowledge and skills can also be used in the industry, since the architecture is adopted in the actual products of the switches and routers.

The rest of this paper is organized as follows. Section 2 briefly describes the architecture and development environment for the IXP1200 and ENP-2505. In Section 3, we discuss the coursework content, followed by the results of evaluating the course done by the students in Section 4. In Section 5, we conclude this paper with considerations for on how to improve the quality of the coursework.

2 Intel IXP1200 Network Processor

2.1 Architecture

Figure 1 shows the internal architecture of the Intel IXP1200. The IXP1200 has seven RISC processors for the CPUS, which consist of six microengines (μEs), and one StrongARM SA1100 core [6]. An IXBus is used for data transfer within the IXP1200, and to/from network interfaces and other IXP1200s. For the interfaces to external systems, it has an SRAM unit and an SDRAM unit for memory access, and a PCI unit for other PCI devices. The scratch pad memory and hash unit are utilized by the microengines to do various operations for network packets.

We use RadiSys ENP-2505 [4] (Figure 2) as the evaluation board for the IXP1200. It has one IXP1200 processor with a 232-MHz StrongARM, four 10/100-Mbps Ethernet ports for packet input/output, a 256-MByte SDRAM and 8-MByte SRAM which are connected to the IXP1200 via the SDRAM unit and the SRAM unit, all of which are installed on a standard PCI board. References [3] and [4] have more detailed information on the IXP1200 and ENP2505.

2.2 Development Environment

The IXP1200 microengines have a unique instruction set especially designed for network packet processing, fast packet input/output, header processing, and other features. However, since the programming language of microengines resembles assembler languages, it is difficult to understand and develop programs to operate the IXP1200. Consequently, we used WorkBench [7] software provided by the Intel Corporation to develop source codes for the microengines as the development environment. It could also simulate the behaviors of the IXP1200 by using developed

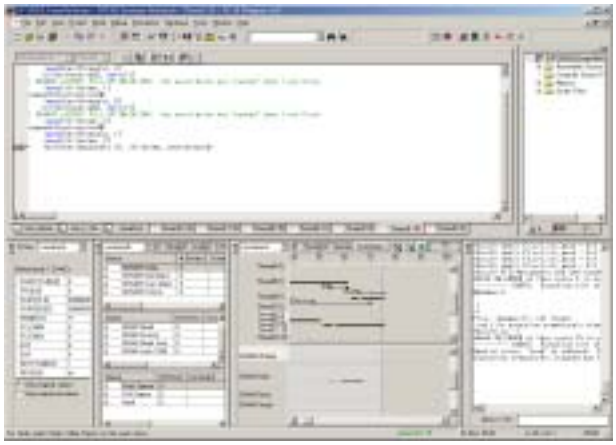

Figure 3: Snapshot of development environment software, WorkBench

source codes, and graphically display the step-by-step actions of each thread of the microengines and interactions with external memory systems, as Figure 3 shows.

Using this software, the source code for the microengines could easily be developed. An example of source code can be seen in Figure 4.

```
immed_w1 [PacketBase,
         (SDRAM_PACKET_BASE >> 16)]
immed_w0 [PacketBase,
          (SDRAM_PACKET_BASE & 0xFFFF)]
immed_w1 [DescriptorBase,
          (QUEUE_LINK_ARR >> 16)]
immed_w0 [DescriptorBase,
          (QUEUE_LINK_ARR & 0xFFFF)]
immed_w1 [SecondBankBase,
          ((QUEUE_LINK_ARR
            + PACKETS_PER_BANK) >> 16)]
immed_w0 [SecondBankBase,
          ((QUEUE_LINK_ARR
            + PACKETS_PER_BANK) & 0xFFFF)]

.local tmp
immed [tmp, 128]
alu [TFifoControlAddr, tmp, OR,
     TFifoElem, <<1];
      128 + element num <<1
.endlocal

immed [QueueIndexMask, (QUEUE_SIZE - 1)]
immed [MaxMPacketSize, 64]

// Masks for the absolute port registers
.local tmp
alu [DescriptorValidMask, --, B, 1, <<31]
immed [tmp, 0x7FF]
alu [PacketLengthMask, --, B, tmp]
alu [PortMask, --, B, 0x0F]
.endlocal
```

Figure 4: Example of the source code for microengine program

3 Coursework Content

This section provides details on Information Networking Exercise I, which is part of the postgraduate coursework.

3.1 Environments

The course is taken over 13 weeks from April to September excluding the summer holidays. Each week, three hours (two hours in class) are assigned for the exercises, which is quite long compared with the time that students spend on general exercises in undergraduate schools. It allows the students to experience and understand the specialized content of the exercises.

The students participating in the course do not have to join a class as such, except at commencement (1st and 2nd week for guidance and introduction). The exercises can be done in the laboratory to which each student has been assigned. Two sets of ENP-2505 boards and personal computers are distributed throughout the laboratories (Figure 5). Each PC has an Intel Pentium 4 1.7-GHz CPU and a 256-MByte memory. The ENP-2505 is installed on the PC via a PCI bus. The PCs are also used for developing the source codes with the WorkBench software.

Each week, the materials and instructions on the exercise, which are not so detailed, are distributed through a Web page (Figure 6) [8] and e-mail. As the students are provided with limited, rather than thorough information, they are expected to try the exercises by themselves. Questions from students are also received by e-mail, which are open to the other students with prompt answers.

We have an associate professor and a research assistant for the teaching staff of coursework exercise. Although We have no TA (Teaching Assis-

Figure 5: Coursework Exercises in Laboratory

Figure 6: Web Page of the Exercises Course

tant) for supporting the exercise in this year, but we plan to introduce some TAs in the next year's coursework.

In what follows, we discuss a detailed program for the exercises undertaken by participating students.

3.2 Weekly Plan

Preparation 1 (1st week): Guidance

We give a lecture on how the students are guided through the exercises and introduce network processors and their characteristics. We also provide a brief explanation on the Intel IXP1200 network processor and the ENP-2505. The students gather in the lecture room.

Preparation 2 (2nd week): Introduction

We next introduce the Intel IXP1200 and ENP-2505 in more detail, including the IXP1200 design architecture, use of the ENP-2505 board and the WorkBench software.

We also provide a short series of instruction on the programming language for microengines, including the instruction set and its usage. Again, the students gather in the lecture room.

Step 1 (3rd–5th weeks): Simple Programming

The first step is simple programming to produce a counter program. It only uses two microengines and two threads, and there is no interaction between the microengines.

The students then shift to a program that is a little more complicated, which uses more than two microengines and threads, SDRAM and SRAM memory units for storing the counter values.

Through this step, the students study the fundamentals of microengine programming, how to control concurrent processing with the six microengines, how to run four threads in each microengine, and how to use interfaces to the memory units.

Step 2 (6th–9th weeks): Understanding SRD

SRD (Simplified Reference Design) [9], which is provided by Intel for simple IP packet forwarding (input and output) operations, is used to teach how network elements are implemented. We provide materials on the detailed mechanisms for SRD and its source code with copious comments and explanation. The students first read and understand the SRD source codes, and execute them on WorkBench to investigate their behavior and packet forwarding performance.

They also implement SRD on an actual PC with ENP-2505, and try data transfer experiments using actual machines. The data transfer speed is also monitored with various settings for packet length, packet input rate and protocols (UDP and TCP), and the results are compared with those obtained on WorkBench.

Through this step, the students obtain knowledge on the basic packet forwarding architecture of the IXP1200. They are also expected to experience throughput monitoring in data transfer experiments with actual machines.

Step 3 (10th-13th weeks): Extending SRD

As the final step in the coursework exercises, the students try to extend the function of SRD to implement various kind of packet processing at the switch/router.

This involves packet dropping/buffering disciplines such as TailDrop, RED (Random Early Detection) [10] and WRR (Weighted Round Robin) [11], bandwidth limitations, emulation of packet loss, and packet delay. Other functions which have been considered by the students themselves as being important can be accepted.

These additional functions are first tested on WorkBench, and then implemented on the actual system. Through actual tests, the differences between the simulation results on WorkBench and the implementation results are evaluated, and the students discuss how to improve packet processing speed on the actual systems. Since this step takes much longer, we expect students use their summer vacation to complete the exercises.

In this step, the students learn a great deal about the mechanisms for packet input, output, buffering, and forwarding, on actual network processors.

At every step, we provide mini-projects, so that the students can check their progress through the exercises. We also ask the students to evaluate the coursework component of the Master's degree program itself. Their evaluations are discussed in the next section.

3.3 Credit Requirements

In addition to attending each exercise session, students are required to submit the following materials to obtain credit for the course. All reports are to be submitted by e-mail. They need to submit;

- A short report on each week's progress
- Complete reports for all Steps

4 Evaluation

In this section, we assess student evaluations of the exercises, which we obtained through questionnaires. During this year's coursework exercises, we asked the students to submit their evaluations in the following four parts;

Part 1: Evaluation of Step 1

Part 2: Evaluation of Step 2

Part 3: Evaluation of understandings of SRD with WorkBench in Step 3

Part 4: Evaluation of SRD implementation on the actual machine in Step 3

Note that Step 4 is currently in progress. The evaluation results will be obtained at the end of September. For each evaluation, the students have to complete a Likert scale using scores from 1 to 5;

- Degree of difficulty (1: easy, 5: difficult)
- Degree of understanding (1: poor, 5: excellent)
- Difficulty of materials (1: hard to understand, 5: easy to understand)

We also ask how many hours it took the students to complete each exercise.

Table 1: Evaluation Results

Part 1	Score	1	2	3	4	5	Avg.
	Difficulty	0	5	13	10	1	3.24
	Understanding	0	3	16	8	2	3.31
	Materials	1	9	13	5	1	2.86
	Hours	4.96					
Part 2	Score	1	2	3	4	5	Avg.
	Difficulty	0	0	5	18	6	4.03
	Understanding	0	1	10	16	2	3.62
	Materials	0	4	16	8	1	3.21
	Hours	10.95					
Part 3	Score	1	2	3	4	5	Avg.
	Difficulty	0	0	7	16	6	3.97
	Understanding	0	10	11	8	0	2.93
	Materials	3	19	9	8	0	2.72
	Hours	11.02					
Part 4	Score	1	2	3	4	5	Avg.
	Difficulty	0	1	4	12	12	4.21
	Understanding	0	7	10	11	1	3.21
	Materials	3	7	12	6	1	2.83
	Hours	15.86					

The evaluation results are summarized in Table 1. Each number on the lines labelled "Difficulty", "Understanding", and "Materials" indicates the number of students who scored 1 though 5 on the degree of difficulty, the degree of understanding, and the degree of difficulty of the materials. We also show the average scores and the average hours it took to complete each step.

These results clearly indicate that the exercises had a sufficiently high level of difficulty since almost all the scores ranged from 2 to 4. Particularly, in Parts 3 and 4, the students rated the exercises to be a little difficult. However, their degree of understanding did not deteriorate very much. This means that the students gained a great deal of knowledge about SRD, and its implementation architecture for packet input/output procedures.

In terms of the difficulty of the materials, we can say that those in Step 3 (understanding of SRD on WorkBench), seemed to be a little too difficult for the students. This is because the mechanism for SRD is quite complex for students who have a limited knowledge of the IXP1200 at this stage. However, once they understand the mechanism, its implementation (Part 4) became quite easy, compared with Part 3. By comparing the evaluation results for Parts 3 and 4, it can be seen clearly where the understanding level has improved significantly.

Finally, we noted the hours taken for each step. We assigned one, two, two, and three weeks for Step 1, 2, 3, and 4, respectively. Since each week allowed three hours for exercises, we concluded that each step took a little longer than expected. However, we think that we can strengthen this coursework component by improving the quality/quantity of the materials, and by increasing the interaction between teachers and students. We also plan to introduce TAs to assist the students.

5 Conclusion

This paper discussed a postgraduate coursework program that involved the completion of exercises undertaken on Intel IXP1200 network processors. The program was designed by the Department of Information Networking of the Graduate School of Information Science and Technology (IST) at Osaka University. We first provided short explanations on the IXP1200 and its evaluation board, the ENP-2505, and described the coursework exercises in detail. We also provided results for the evaluations of the exercises done by the students, and concluded that our postgraduate coursework program is currently achieving good results in terms of the students' level of understanding of the design architecture, implementation issues, and programming languages for network processors.

In addition to using in the coursework, we use the Intel IXP1200 for research activities, where we implemented the proposed schemes for the switch/router architectures on the IXP1200 to confirm their effectiveness [2].

For the future work, we plan to improve the quality and quantity of the materials distributed to students to increase their degree of understanding of the exercises. This does not mean that we will provide thorough excessively detailed information to the students. The information we provide will remain limited to encourage the students to learn by themselves. We also plan to collect even more evaluations in more detail, by interviewing additional students on the coursework exercises.

Acknowledgements

This work was partly supported by the IXA University Program sponsored by Intel Corporation. We also thank Mr. Akio Tomobe of Intel for kind support.

References

[1] M. Murata and S. Nishio, "New organizations for IT-related R&D at Osaka University," in *Proceedings of 4th AEARU Workshop on Computer Science (Kyoto)*, Dec. 2001.

[2] G. Hasegawa and M. Murata, "Research and educational activities using IXP1200 at Osaka University," in *IXA University Program Workshop*, June 2002.

[3] Intel IXP1200 Network Processor Family. available at `http://www.intel.com/design/network/products/npfamily/ixp1200.htm`.

[4] RadiSys ENP-2505 PCI Packet Processing Engines. available at `http://www.radisys.com/oem_products/ds-page.cfm?ProductDatasheetsID=105%5`.

[5] Intel IXA University Program. available at `http://intel.com/research/university/comm/index.htm`.

[6] Intel StrongARM Processors. available at `http://www.intel.com/design/strong/collateral.htm?iid=strongarm+leftnav%&`.

[7] Intel IXA Software Developers Kit 2.0 for IXP1200. available at `http://www.intel.com/design/network/products/npfamily/sdk2.htm`.

[8] Web Page of Information Networking Exercise I. available at `http://www-ana.nal.ics.es.osaka-u.ac.jp/~hasegawa/japanese/lessons/exer%ciseI/` (in Japanese).

[9] Charles Preston, "IXP1200EB Simplified Reference Design," January 2001.

[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[11] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round–robin cell multiplexing in a general–purpose ATM switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1265–1279, Oct. 1991.