

Title	命令セットによるマイクロアーキテクチャへの影響に関する研究 [課題研究報告書]
Author(s)	桑田, 正明
Citation	
Issue Date	2016-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/13635
Rights	
Description	Supervisor: 田中 清史, 情報科学研究科, 修士

Study on Influence of Instruction Set on Microarchitecture

Masaaki Kuwata (1310704)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 6, 2016

Keywords: Instruction Set Architecture, FPGA, MIPS, ARM, SPARC.

Instruction sets that von-Neumann-type processors execute are diverse, but the superiority or inferiority is barely discussed because there are many similarities in order to provide general-purpose programmability. However, instructions to be used vary according to target applications, and it is expected that difference in performance appears from ISA (Instruction Set Architecture) to ISA, considering instruction efficiency.

In this study, for the purpose of clarifying how ISA influences microarchitecture, I targeted FPGA (Field Programmable Gate Array) and developed microarchitecture of several ISAs. Specifically, I clarified how the difference in ISA affected speed / hardware size by actually designing the processor for each ISA, and comparing the results that I implemented. In addition, I arranged common components which were independent of ISA, similar components, and special components, and showed that processors for different ISAs could be designed by choosing and connecting components with simple modification.

I adopted instruction sets of MIPS, ARM and SPARC as the target ISAs in this study. (Although SPARC is not common or practical in these days, I involved SPARC for a comparison to MIPS and ARM which are widely used.)

In chapter 1, I show the background, the purpose, the study methodology, and the organization of this report.

In chapter 2, I show the history and background of ISAs and the instruction formats of each ISA, and show basic design of the processor based on each ISA and results of the simulation. I show MIPS first. As a basic design of the MIPS processor, I designed it with a single cycle implementation where an instruction is executed in one clock cycle without cache memories. The following instructions are implemented.

- add,sub,and,or,slt,addi,sw,lw,beq,bne

I designed this processor by using Verilog HDL in this study and referring to David A. Patterson and John L. Hennessy's "Computer Organization and Design", 4th edition, first volume, figure 4.21. I aimed at running a program that calculates $1+2+3+4+5+6+7+8+9=45$ using add instructions, store instructions, branch instructions, and load instructions, whose correctness was confirmed to in simulation in this study.

I show ARM next. As a basic design of the ARM processor, I designed it with a single cycle implementation as the MIPS one, without cache memories. The following instructions are implemented.

- MOV,ADD,SUB,AND,CMP,STR,LDR,BNE,BEQ,BAL

(Although ARM processors in general have 16-bit wide instructions, called Thumb instructions, that are aimed at improving the code efficiency, the Thumb instructions are out of the scope of this study.) I designed this processor with Verilog HDL and referred to "ARM Architecture Reference Manual version6 (ARM DDI 0100HJ-00)" by ARM Ltd. Similar to the MIPS's case, I tried to calculate $1+2+3+4+5+6+7+8+9=45$ using move instructions, store instructions, add instructions, compare instructions, branch instructions, and load instructions, in the simulation.

I show SPARC next. As a basic design of the SPARC processor, I designed it with a single cycle implementation and no cache memories. The following instructions are implemented.

- ADD,SUB,SUBcc,AND,OR,ST,LD,BE,BNE,BL,SAVE,RESTORE

In addition, I designed two cases, with the register window and without the register window. I designed it with Verilog HDL and referred to SPARC International Inc.'s "The SPARC Architecture Manual Version 8 Revision

SAV080SI9308”. I executed a calculation of $1+2+3+4+5+6+7+8+9=45$ using add instructions, store instructions, subtract instructions, branch instructions, and load instructions in the simulation.

In chapter 3, I show the evaluation of the source files, the evaluation of the circuits which I implemented, and the relations between ISA and the modules. I show the evaluation of the source files first. The source files are Verilog module files designed in ISE Design Suite 14.7 of Xilinx, Inc. The file extension is .v. The file sizes and the number of lines of the source files of each ISA are as follows.

- MIPS, SPARC (without register window), ARM, and SPARC (with register window) are in order of increasing sizes and lines.

I show the evaluation of the circuits next. I implemented the circuits in ISE Design Suite 14.7 of Xilinx, Inc. for Spartan6 FPGA XC6SLX45. The results of the evaluation of the circuits are as follows.

- MIPS, SPARC (without register window), ARM, and SPARC (with register window) are in order of increasing FPGA resources.
- MIPS, SPARC (without register window), SPARC (with register window), and ARM are in order of decreasing frequency.
- MIPS exhibits the highest frequency with the fewest resources.

I show the relations between ISAs and the modules next. I classified modules of each ISA into common components, similar components, and special components. The common components are independent of ISA and they are modules which have the same source codes in all the ISAs. They can be used to build processors of different ISAs. The similar components are modules which have partly different source codes in all the ISAs. They can be used to build processors of different ISAs with a small modification. The special components are modules which have different source codes from ISA to ISA or which only one ISA has. They depend on ISA and show special characteristics of ISA. In addition, I showed differences of similar components and special components for each module, and showed special characteristics of ISA. Processors of different ISAs could be organized by choosing and connecting these common components, similar components, and special components, with appropriate, slight modifications.

In chapter 4, I show the summary of this study.