Proceedings of 2005 5th IEEE Conference on Nanotechnology
Nagoya, Japan, July 2005

FPGA Platform for CPU Design and Applications

Chi-Jeng Chang, Chi-Wu Huang, Ying-Ping Lin, Zen-Yi Huang and Teng-Kuei Hu

Department of Industrial Education, National Taiwan Normal University, Taipei, Taiwan, R.O.C.
e-mail : t07022@cc.ntnu.edu.tw

Abstract — **This paper presents a CPU design of 25 MIPS instructions in addition to the interface controller circuitries of LCD, 7-sequment and key pad and all are downloaded on a 200k gate-count FPGA board for system verification. Then an image process device developed in another FPGA board was connected to the CPU as an image accelerator. By using the same way, other mechantronic or nano devices could also be connected to the CPU with proper designed controllers.**

The FPGA board could be used for teaching CPU design, controlling applications and also for system-on-chip (SoC) designing since all circuitries might be incorporated in a signal FPGA chip. A multifunctional platform is gradually evolving for teaching and applications.

Index Terms — **MIPS, CPU, FPGA, CMOS Image Sensor, SoC**

I. Introduction

As the gate-count in FPGA chips getting larger and larger, more educational applications in reconfigurable design were found at literatures in recent years as follows:

1. Hiroyuki Ochi in 1997 [1] at Hiroshima University, Japan, proposed a 10-instruction 16-bit RISC processor was designed and downloaded in a self-developed FPGA board with 15K gate-count Xilinx chip. Pipelined design was his emphasis.

2. William Richard in 1999 [2] at Washington University, St. Louis, USA, proposed an 8–bit microprocessor. It was designed and implemented in ACTEL commercial FPGA board. Due to limited FPGA chip capacity, only a few circuitries designed with lower gate-count could be downloaded to the board.

3. Calazan in 2001 [3] at the Catholic University of Rio Grande do Sul, Brazil, proposed a 16-bit processor simulation. A commercial FPGA board (XS40) with 5k gate-count was used for the download of RAM controller circuit. He perceived that student's lack of motivation for the subsequent hardware course is due to lack of application.

4. Murray Pearson [4] in 2002 at the University of Waikato, Hamilton, New Zealand, proposed a 32-bit simplified MIPS-liked [5] CPU design. The CPU was downloaded to self-designed 100k gate-count FPGA just completed, no further commons about using the board.

The main concerns of the above educational FPGA applications were to introduce the CPU design concepts, the further applications of their designed CPU were not much considered. Only Calazan mentioned that the lack of application is perceived as the main source of student's lack of motivation for the subsequent hardware course. Therefore, application driven was also an important factor for the educational training of CPU design.

In this paper, we present a 32-bit CPU design capable of adding many applications on it, to continue practicing the hardware design, and 32-bit CPU is more suitable in real applications than 8 or 16 bit CPU designs in the above first three cases mentioned previously. Hopefully, this practice could also be motivations to students for their subsequent hardware course.

MIPS CPU structure with 25 instructions listed in Fig. 1 were designed following the design procedures in Patterson's text book. The completed circuitry was then downloaded in a self-designed FPGA board with 200K gate-count chip for function verification. Some practical applications were made by using the 25 instructions, though the full instruction set is 146 for MIPS 2000. After finishing the verification the CPU board was connected with an Image Process Device (IPD), also developed in our lab using other FPGA (200K gate-count) board, to capture and process the image from a CMOS image sensor. This kind of connection made MIPS become a core processor for I/O devices. Using the similar connection stepper motor or other nano-mechantronic devices could also be connected to this design, which made this MIPS become a core processor (or microcontroller) for a SoC [6] based application design. It will be described more detail in later sections

0-7803-9199-3/05/$20.00 ©2005 IEEE

No.	Mnemonic	OP Code	Operation
1	LW	100011	$t = MEM[$s + offset]
2	SW	101011	MEM[$s + offset] = $t
3	ADD	000000	$d = $s + $t
4	ADDU	000000	$d = $s + $t
5	SUB	000000	$d = $s - $t
6	SUBU	000000	$d = $s - $t
7	AND	000000	$d = $s & $t
8	OR	000000	$d = $s | $t
9	XOR	000000	$d = $s ^ $t
10	NOR	000000	$d = ~($s | $t)
11	SLL	000000	$d = $t << shamt
12	SRL	000000	$d = $t >> shamt
13	SRA	000000	$d = $t << shamt
14	ADDI	001000	$t = $s + immed
15	ADDIU	001001	$t = $s + immed
16	ANDI	001100	$t = $s & immed
17	ORI	001101	$t = $s | immed
18	XORI	001110	$d = $s + immed
19	BEQ	000100	If $s = $t advance_pc(offset); Else advance_pc
20	BGEZ	000001	If $s >= 0 advance_pc(offset); Else advance_pc
21	BGTZ	000111	If $s > 0 advance_pc(offset); Else advance_pc
22	BLEZ	000110	If $s <= 0 advance_pc(offset); Else advance_pc
23	BLTZ	000001	If $s < 0 advance_pc(offset); Else advance_pc
24	BNE	000101	If $s != $t advance_pc(offset); Else advance_pc
25	J	000010	If $s != $t advance_pc(offset); Else advance_pc

Fig. 1. 25 instruction implemented in FPGA board

Section Ⅱ describes some detail about the design and verification about MIPS. Section Ⅲ describes the FPGA board design and function verification. Section Ⅳ describes how IPD is connected to MIPS to become a dedicated processing accelerator for MIPS. Section V mentions the prototype of SoC design. Some concluding remarks are made in Section Ⅵ.

II. MIPS CPU Design

MIPS functional block diagram and datapath with 25 instructions of Fig. 1. were shown in Fig. 2. It could be roughly combined into four units, namely, MEM (Memory unit), RGB (Register Bank unit), ALU (Arithmetic and Logic Unit), and CTR (Control unit). The control functions in CTR were decided by the ALU operations as well as the datapath between MEM, RGB and ALU.

Using bottom-up design procedure from layer 1 to layer 4 shown in Fig.3. layer 1 included mem.xco file, basic arithmetic and logic functions, and a barrel shifter. File mem.xco was used for generating a memory block in Xilinx FPGA chip to constitute MEM in layer 2. Basic arithmetic and logic functions with barrel shifter constituted ALU in layer 2, layer 2 also included RGB which was an register file containing 32 registers, the connections between MEM, RGB and ALU constitutes a datapath in layer 3. CTR which controlled the data flow inside the datapath was also in layer 3. CTR combines the data path between other units, constituted the 32 bit MIPS in layer 4. Each of the 25 instructions of MIPS and some programs were downloaded in a FPGA board, described in next section, for system verification.

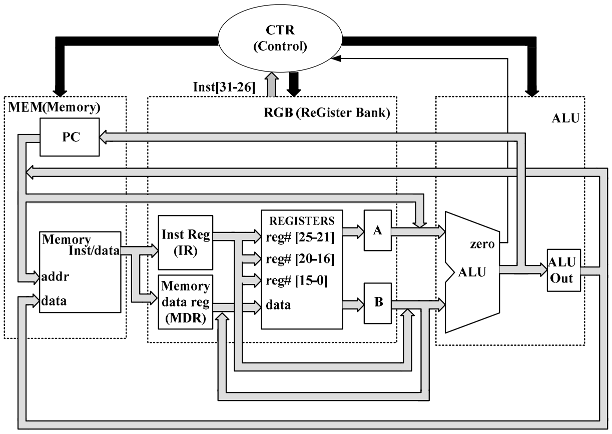


Fig. 2. MIPS functional block diagram and internal data path from layer1 to layer 4

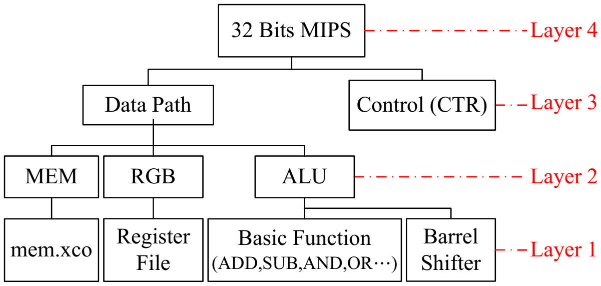


Fig. 3. Bottom up design procedure form layer 1 to 4

III. FPGA Board and Verification

A FPGA board shown in Fig.4. was developed for demonstrating CPU operations. It included a 200K gate-count FPGA chip (Xilinx XCS200), 4×4 Keypad, 7-seg LED display, text LCD, SRAM and EEPROM, 2 printer ports, memory switches and LEDs.

The I/O device controller circuitries should be designed and loaded to the FPGA chip before the CPU verification could be started. Fig.5 shows how MIPS were connected to all the controllers. The FPGA circuitries design of keypad, 7-seg LED display and LCD were mostly followed the popular algorithms used in 8051 programs.

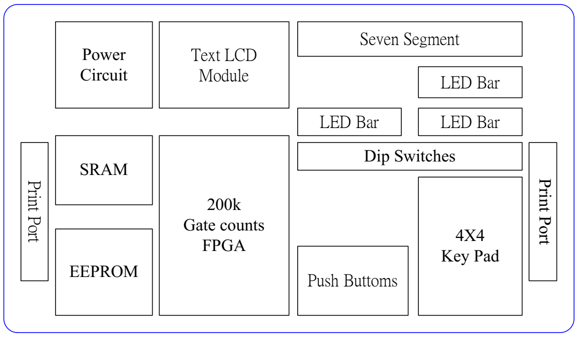


Fig. 4. FPGA board for demonstrating CPU operations

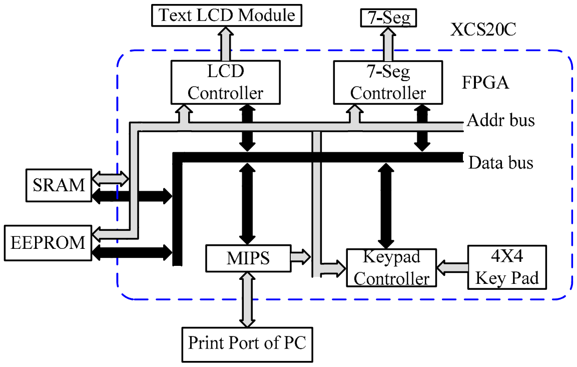


Fig.5. MIPS connection with device controllers, ROM and SRAM

There were two steps for MIPS CPU verification:

1. Instruction verification: each of the 25 instruction was tested step by step.

2. Program verification: many programs were written for running in FPGA board. Binary counter and 24-hour clock were two of the testing programs that could be dynamically observed for their operations while the program was running. Fig. 6a. showed the snap-shot of the 24-hour program running at 2 o'clock, 55 minutes and 22 seconds (2:55:22) form LCD display. The photo of FPGA board was shown in Fig. 6b.

IV. MIPS AS CORE PROCESSOR

Image Processing Device (IPD) was a device developed in another FPGA board connected with a CMOS sensor in our lab. It could do pipelined image convolution and sorting while capturing image from the CMOS sensor. Image convolution and sorting helped further filtering out the FPN (Fixed Pattern Noise) and the dark current noise from CMOS image sensor for gaining a better image quality. Its connection with MIPS was shown on the top of Fig. 7. The image displayed at PC through this experiment was shown in Fig.8, using photo of Lena.

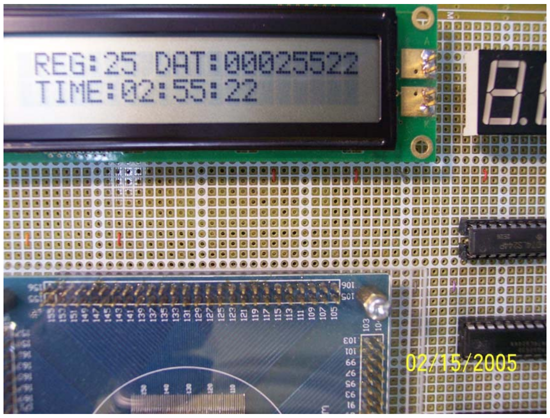


Fig.6a. 24-hour clock running at 02:55:22

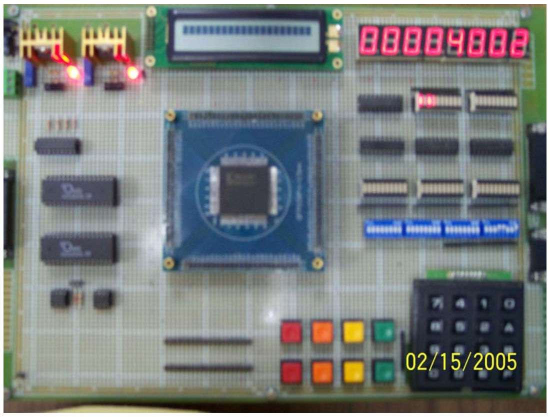


Fig.6b. Photo of the Self-designed FPGA board

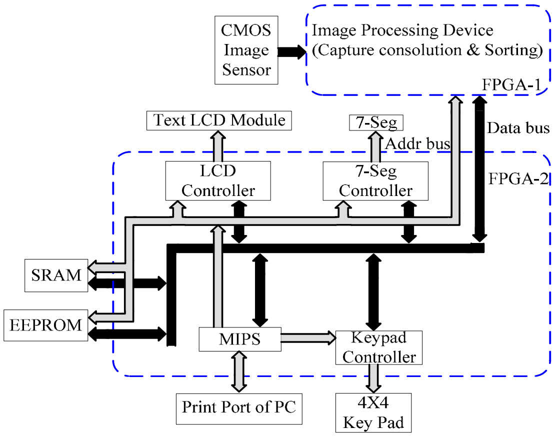
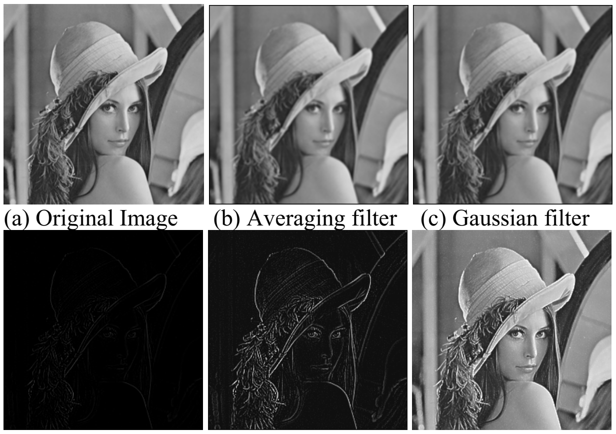


Fig. 7. MIPS and Image Processing Device (IPD)


(a) Original Image (b) Averaging filter (c) Gaussian filter

(d) High pass filter (e) Enhance from d (f) a plus d

Fig.8. Photo of Lena through this experiment

V. Entering SoC Design

The two FPGAs in Fig. 7. could be put together in one FPGA if its gate-count was over 600K as shown in Fig. 9. actually more device controllers, such as USB, DC motor, stepper motor, nano-mechantronic devices,…could be put into a FPGA as long as the chip capacity was large enough. Then MIPS embedded with many device controllers in a FPGA chip shown inside the dash-lined area in Fig. 9. It demonstrated a kind of SoC architecture design. The architecture in Fig. 9. might gradually become a platform for CPU design capable of adding controllers for controlling special devices, such as image processing, mechantronic, nana-optics,...etc.

VI. Concluding Remarks

From the description of previous sections. This CPU and application design includes three steps; CPU design, I/O interface design and controller design for special devices.

First the CPU design is based on Patterson's text book. Then the I/O interface is the interaction between controller circuitries (including keypad, LED and LCD) and MIPS. The interface design, usually using software or dedicated IC chips, is now being put into FPGA circuit. Then MIPS interface design in FPGA might be considered as MIPS' general applications.

As for the controllers design for special devices (including mechantronic devices) are the special applications interacting with CPU instructions and are more challenging design. All these CPU and device controller circuitries design will finally be the basis for SoC design.

Actually more MIPS instructions can be added to this original 25 instructions for more complicated applications, or taken off some instructions for simple application, since FPGA is reconfigurable. For more flexible, we can use other CPU instructions such as ARM, 8051 or self-developed CPUs.

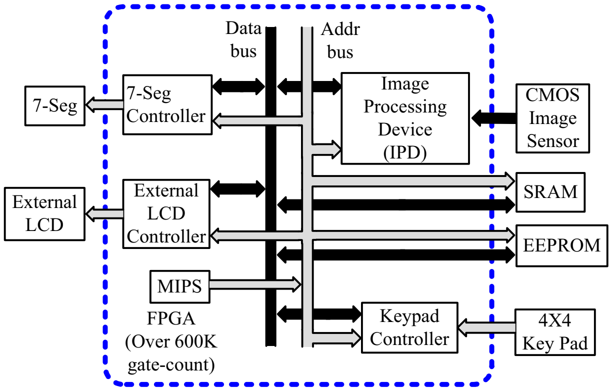


Fig. 9. MIPS and many device controller merged in a FPGA chip demonstrating a kind of SoC design.

References

[1] H. Ochi, "ASAver.1: an FPGA-based education board for computer architecture/system design," *Design Automation Conference 1997. Proceedings of the ASP-DAC '97*. Asia and South Pacific, 28-31, pp. 157-165, January 1997.

[2] W. D. Richard, D. E. M. Taylor, D. M. Zar, "A capstone computer design course," *Education, IEEE Transactions on,* vol. 42, Issue: 4, pp. 288 – 294, November 1999.

[3] N. L. V. Calazans, F. G. Moraes, "Integrating the teaching of com organization and architecture with digital hardware design early in undergraduate courses," *Education, IEEE Transactions on,* vol. 44, Issue: 2, pp. 109 – 119, May 2001.

[4] M. Pearson, D. Armstrong, T. McGregor, "Design of a processor to support the teaching of computer systems," *Electronic Design, Test and Applications, 2002. Proceedings. The First IEEE International Workshop on* , vol. 29-31, pp. 240-244, January 2002.

[5] David A. Patterson and John L. Hennessy, "Computer Organization and Design (3rd Edition)," San Francisco : Morgan Kaufmann, 2005.

[6] Steve Furber, "ARM System-on-Chip Architecture (2nd Edition)," Maidstone, Kent, USA : Addison-Wesley, 2000.

[7] Zen-Yi Huang, "FPGA Platform for Capturing and Filtering CMOS Image Sensor," Master of Technology, National Taiwan Normal University, 2004.

[8] Ying-Ping Lin, "FPGA Platform for CPU Design and Education Training Using MIPS," Master of Technology, National Taiwan Normal University, 2005H.