Design of Simple CPU Based on Hardware Description Language

Jianming Liu[1], Yunjie Zhang[2]
1:Computer Department
Weifang Medical University
Weifang, China
2:Electronic & Information Engineering Department
Tianjin Institute of Urban Construction
Tianjin, China
e-mail:sdwfljm@126.com

Lili Xu[3], Pengtao Liu[1]
3:Weifang Vocational College
Weifang, China

***Abstract*—Based on FPGA, the hardware description language Verilog and the top-to-down design method of modularization are adopted to achieve the design of a simple CPU logic controller, in which the software design and simulation are conducted for each module of the CPU, and then each module is synthesized. The principle analysis is performed for the key function modules of CPU. The function-sequence simulation diagrams are given by means of compilation and adaptation with software QuartusII. The result suggests that the design has a strong flexibility, reliability and easiness for extension. The CPU can achieve more functions as long as extending or amending the instructions appropriately.**

Keywords-FPGA; CPU; EDA;VHDL.

I. INTRODUCTION

Hardware Description Languages is used to design the behavior of logic circuits. In the older days, early Intel's and other processors were designed by hand, laying out the layers of an IC's substrate masks using regular drafting techniques. There were little or no computer-aided design tools to help the chip developer. This method was so tedious that the least and few people had the patience and skills for such a task. Thankfully, times have changed and designing custom processors are within reach of many designers of such hobby. HDLs give a way to describe and implement a hardware design in a similar manner as developing a software program. Besides, there are similar tools like compilers, debuggers and simulators. There are two predominate HDL language, Verilog and VHDL. VHDL is adopted in this paper.

Based on the relevant knowledge on the principles of the CPU composition, the paper designs a simple CPU by using top-down approach, and completed all of the functional modules using the hardware description language VHDL and EDA technology. The functional modules include the control module, memory module, arithmetic and logic module, the general register group, the program counter, address registers, instruction register, timing components, input device, as well as output device [2][8]. It is designed, compiled and simulated under the integrated development environment of Quartus II, and after that it is downloaded to the FPGA experimental platform for units and system testing.

The paper is organized as follows. In section II, the paper introduces the architecture of the simple CPU. In section III and IV, main logic components/modules of the CPU internal structure are described in detail and how the logic components can work collaboratively is thoroughly discussed. The implementation and simulation of the CPU with basic functions will be shown in Section V followed by short conclusion and future work in Section VI.

II. DESIGN OF INSTRUCTION SET

Single-address instruction format is used in our simple CPU design. The instruction word contains two sections: the operation code (opcode), which defines the function of instructions (addition, subtraction, logic operations, etc.); the address part, in most instructions, the address part contains the memory location of the datum to be operated, which is called direct addressing. In some instructions, the address part is the operand, which is called immediate addressing.

For simplicity, the size of memory is 256× 16 in the computer. The instruction word has 16 bits. The opcode part has 8 bits and address part has 8 bits. The instruction word format can be expressed in Figure 1

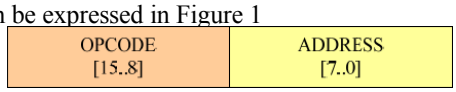

Figure 1. the instruction format

The opcode of the relevant instructions is listed in Table 1.

In Table 1, the notation [x] represents the contents of the location x in the memory. For example, the instruction word 00000011101110012 (03B916) means that the CPU adds word at location B916 in memory into the accumulator (ACC); the instruction word 00000101000001112 (050716) means if the sign bit of the ACC (ACC [15]) is 0, the CPU will use the address part of the instruction as the address of next instruction, if the sign bit is 1, the CPU will increase the program counter (PC) and use its content as the address of the next instruction.

Table 1 the relevant instructions system

INSTRUCTION	OPCODE	COMMENT
IN Rd, [addr]	0001 ×× Rd addr	io→Rd
OUT [addr], Rs	1000 Rs ×× addr	Rs→io

ADD Rd, Rs	0101	Rs	Rd	Rs+Rd→Rd
CMP Rd , Rs	0011	Rs	Rd	Rs−Rd, CF
INC Rd	0110	××	Rd	(Rd)+1→Rd
DEC Rd	1001	××	Rd	(Rd)−1→Rd
MOV1 Rd, Rs	0010	Rs	Rd	Rs→Rd
MOV2 Rd, data	1010	××	Rd	data→Rd
	data			
LOAD Rd, [Rs]	1011	Rs	Rd	[Rs]→Rd
STORE [Rd], Rs	1100	Rs	Rd	Rs→[Rd]
JMP addr	0111	××××		addr→PC
	addr			
JC addr	0100	××××		If CF=1, addr→PC
	addr			

III. THE DESIGN OF TOP MODULE FORM

The basic structure of microprocessor generally consists of data path and controller, which is shown in Figure 1. The basic structure of the CPU generally consists of data path and controller [1]. Data path is for processing data information, which represents the structure and layout of the CPU as a whole. The data path and instruction system provide necessary basis for controller design. Different data path structure has a direct impact on the running speed of the CPU. The structure of the data path designed in this paper is shown in Figure 1. The functional modules of it mainly include data bus(DB), arithmetic and logic unit(ALU), memory module(RAM), general register(R0), program counter(PC), address registers(AR), instruction register(IR), data buffer register(DR1,DR2), timing pulse generator, input device(SWICH), as well as output device(LED).

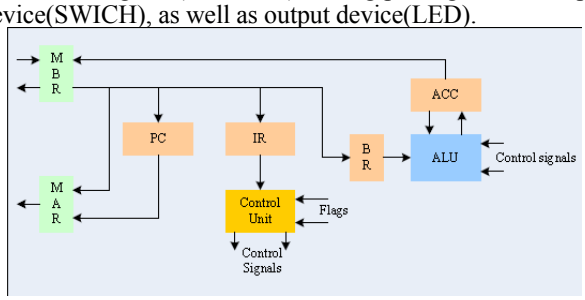

Figure 2. The structure of the data path

Data path of single bus is adopted in the design, therefore the data and address share the same bus [9]. In order to avoid the data conflict occurred in the bus, it is through tristate logic gate to connect to data bus for the out ports of the most modules. Data path of single bus is convenient for extension; however, both the data and the instructions are all transferring through the same bus. So the assignment of the data bus should be made in a proper way.

The design of the data path is processed in the integration environment of Quartus II. The ALU is designed in VHDL; however, all kinds of registers, the RAM, the PC and the tristate logic gate could be designed through calling the LPM (Library Parameterized Modules) provided by Quartus II. As for timing pulse generator, it is designed by 4 D-type Flip flops and its Pulse-width is regulated by the timing source.

Using the approach of top-down and the method of modular design, in the top schematic of system it calls and links the symbols of all kinds of functional modules through the single bus. Besides, the corresponding control signals of the functional modules are marked in the top schematic. Consequently, the whole circuit of the CPU is accomplished. Top-level schematic of the data path is shown in figure 2:

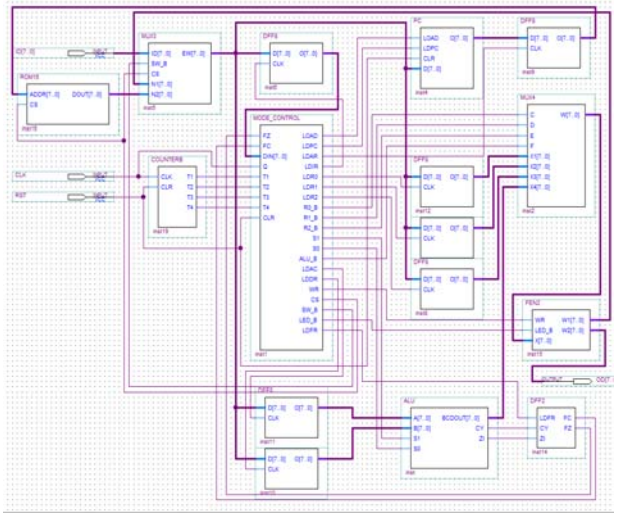

Figure 3. Top-level schematic of the data path

IV. THE MAIN MODULES OF CPU

Controller is the command center of CPU and controls all kinds of signals used in the data path to control data transferring or data processing sent from controller. The main function of it is generating the various operation control signals, which make the data path be established correctly and the instructions be fetched and executed cyclically [4][5]. There are two ways for controller design, which are hardwired controlling and micro program controlling. By comparing the two methods, the first is widely used. So in this paper, the controller adopts micro program controlling.

A. Design of the datapath

MAR contains the memory location of the word to be read from the memory or written into the memory. Here, READ operation is denoted as the CPU reads from memory, and WRITE operation is denoted as the CPU writes to memory. In our design, MAR has 8 bits to access one of 256 addresses of the memory.

16 bits MBR contains the value to be stored in memory or the last value read from memory. MBR is connected to the address lines of the system bus.8 bits PC keeps track of the instructions to be used in the program. In our design, PC has 8 bits. 8 bits IR contains the opcode part of an instruction. BR is used as an input of ALU, which holds other operand

for ALU. ACC holds one operand for ALU, and generally ACC holds the calculation result of ALU. MR is used for implementing the MPY instruction, holding the multiplier at the beginning of the instruction. When the instruction is executed, it holds part of the product.

LPM_RAM_DQ is a RAM with separate input and output ports, which works as memory, and its size is 256× 16. Although it's not an internal register of CPU, we need it to simulate and test the performance of CPU.

B. Design of the Control Unit

CU is used for controlling all of the register-control-signals outside and doing the steps of the micro-operation inside.Based on the machine instruction, the flow chat of programs should be designed. The flow chat of programs is shown in Figure 4. As shown in Figure 2, the controller module is connected via a number of inputs and outputs to the 'datapath' module. The controller is responsible for decoding the current instruction in the IR (instruction register) and set up the datapath control signals so that the data is correctly routed from the source to the destination register accordingly to the semantics of the instruction. As the instruction in the IR changes， so does the functionality of the Controller. The Controller is essentially a simple state machine that asserts a pre-determined number of signals for the various instructions.

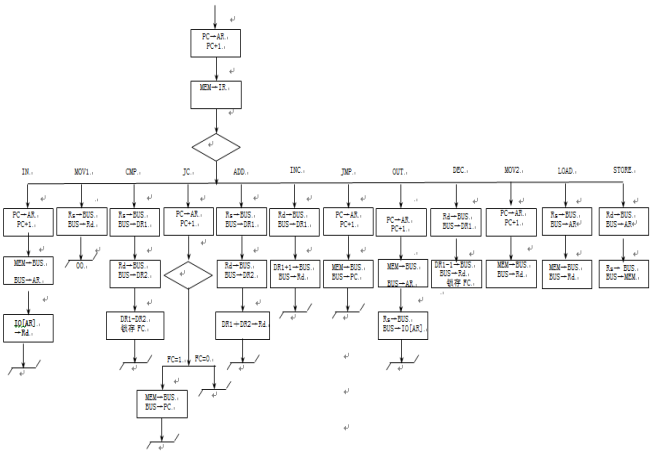

Figure 4. Flow chat of micro programs

In the flow chat of programs, a box represents one micro instruction. If one operation should be performed, the corresponding field would be set '1', otherwise '0'. When the micro instruction is sent to data path from control memory, the corresponding module will work correctly. The machine instruction is interpreted and executed after the IR receives it. That is to say, it will execute the corresponding micro program segment. In accordance with the 3 higher bits (IR7-IR5), several branches are produced. Each of the branches presents one machine instruction. Finally, all of the codes of the micro instructions are obtained.

V. SIMULATION AND IMPLEMENTATION BASED ON FPGA

The simple CPU, which is composed of the data path and the controller, is designed and compiled under the integrated development environment of Quartus II. After that the simulation for the process of the machine instructions is performed [10]. To meet the requirement of the simulation, a section of assembly language code should be given. The function of the code fragment mainly includes receiving an 8-bit data from input device (SWITCH), adding it with the data from a cell of the main memory, returning the result to another cell of the main memory, sending the result to output device (LED), and jumping back to the beginning. The assembly language testing code is listed as follows:

IN1 RO，[80H]	8010	[80H] →R0
MOV2 R2,0	00A2	0→R2
ADD R2,R0	52	(R0)+(R2)→R2
DEC R0	90	(R1)-1→R1
DEC R0	90	(R1)-1→R1
JC L2	0B40	
JMP L1	0470	L2→PC
OUT [90H], R2	9088	R2→LED
JMP L2	0B70	

In accordance with the instruction format, the assembly language testing code is translated into machine language code which is written into the RAM as binary form.

Besides the schematic file, a wave file should be established which includes many excitation signals. In accordance with the excitation signal, the function simulation of the CPU can be performed after the machine language code is written into the RAM. The function simulation waveforms are stated in Figure 5:

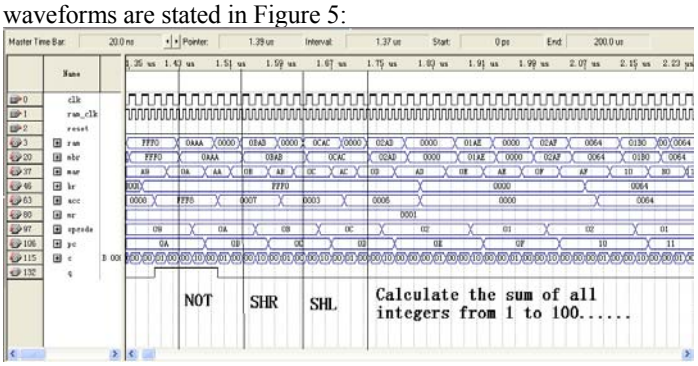

Figure 5. Function simulation waveforms

In Figure 5, it indicates the execution flow of 5 machine instructions, and 2 instructions of that is explained as follows:

Machine cycle 1: the '0001141' micro instruction in the micro address of 00H is executed, and the corresponding operations include 00 AR, PC=PC+1=1. When the cycle is gone, the next micro address is 01H.

Machine cycle 2: the '00020A0' micro instruction in the micro address of 01H is executed, and the corresponding operations include 20H IR, P1=1. When the cycle is gone, the next micro address is 04H. The next micro address is not sequential because the branch entrance of micro programs is different.

After the design is complied, a netlist file will be produced. If there are no problems in the process of the function simulation, the netlist can be downloaded into the FPGA development board which is named cyclone EP1C12Q240C8. The board-level hardware test is made on the FPGA development platform. The hardware test result shows the simple CPU can execute the testing code segment composed of some machine instructions rapidly and accurately.

VI. CONCLUSION

By using top-down approach and modular design method, a simple CPU is designed and implemented based on FPGA, which has achieved the expected goals through the simulation and verification. This CPU design can complete most of the instructions. And from the simulate wave, we can distinguish every step of the instruction clearly. The weakness of this CPU design is that there are too many control signals. When watching the experiment results (the output waveform), it is not very convenient.

We may treat the data signal themselves as the sensitive signals in their symbol files.reading or writing according to the change of them, the output waveform would be more simple .But it would bring much more problems in design. There is another weakness that the method of DIV is easy but it repeats too many times and will spend too much time.

REFERENCES

[1] Xing Yuhua, Wang Ru, "An Optimized Design of MCU in CPU Soft-core Based on the FPGA," The Eighth International Conference on Electronic Measurement and Instruments(ICEMI' 2007), China, 2007. China, pp.16-29

[2] Li Jingpeng, "An optimized design of MCU including predication," Microelectronics and computer, vol.23, pp.25-27, 2006

[3] Pan Song, Pan Ming, Principles of Modern Computer Architecture. Science Press, 2007.

[4] Pan Song, Huang Jiye, EDA Tecnologey and VHDL, Tsinghua University Press, 2002

[5] Zhang Kai, Tang Zhizhong, "Design and implement of 16-bit general CPU," Computer Engineering and Applications, vol.32, pp.116-117, 2003.

[6] William Stallings. The Structure of Computer. Tsinghua Publishing Company, China, 2000

[7] Wayne Wolf, Modern VLSI Design, Printice Hall, 2001

[8] Tian Hongli, Yan Huiqiang, Geng Hengshan, Liu Su, "Design and implementation of 8 bit micro-controller," Computer Engineering and Applications, Vol.46, pp.60-63, 2010

[9] Wang Peilin, "Design of a simple CPU based on FPGA," Coal Technology, Vol.29, pp.10, 2010

[10] Qi Mei, Zhang Peng, Dong Ye, Chang Lei, "Design and realization of model machine based on CPLD combinational logic controller," Research and Exploration in Labortaory, Vol.29, pp.7, 2010