Elastic Pipeline Design of High Performance Micro-Controller YL8MCU for Signal Processing of Digital Home Appliances

Fulong Chen[1,*], Yunxiang Sun[2], Chuanxin Zhao[2], Jie Yang[2], Heping Ye[2], Junru Zhu[2], and Qimei Tang[2]

[1]*Department of Computer Science & Technology, Anhui Normal University, Wuhu, Anhui 241002, China*

[2]*Network and Information Security Engineering Research Center, Anhui Normal University, Wuhu, Anhui 241002, China*

Abstract-With the growing complexity of the structure and function of the digital home appliance controllers, the higher requirements on the design of integrated circuits are expected. In view of the lower efficiency of the traditional large scale integrated circuit design, such as long design cycle, high investment and bad flexibility, therefore, the traditional design method is unable to meet the rapid product development. A design and its implementation scheme based on System-on-Programmable-Chip (SoPC) technologies with the Intellectual Property (IP) core parametric technology and elastic design method are proposed for the Micro-controller Unit (MCU) YL8MCU applied in digital home appliances. It makes used of Reduced Instruction Set Computer (RISC) framework and 5-stage pipeline technology for designing the instruction system and hardware structure so as to improve the speed of processing data and instructions. Finally the correctness and feasibility of its function is verified with the integrated verification, the IP function module verification based on coverage and the simulation test of the controlling flow for the washer. The results show that YL8MCU meets the requirements of digital home appliances and supports system integration on chip.

Keywords: Digital Home Appliances, Micro-controller Unit, Pipeline, IP Core.

1. INTRODUCTION

With the continuous development of microelectronics technology, especially the emergence of the large-scale integrated circuits and embedded system, these technologies have brought a new technological revolution in the modern industrial control field. Micro-controller Unit (MCU) is the core component of embedded system and is known as monolithic (single chip) microcomputer. There are microcomputer components, such as central processing unit (CPU), program memory (ROM), the working memory (RAM), input and output interface circuit (I/O), the timer/counter (TIMER) integrated on a single chip to form a complete microcomputer. Micro-controller has a widely application domain. In addition, it also has many advantages, such as single-chip, small size, low power, reliability and rich peripheral resources [1-6]. But in recent years, the integration level of Micro-controller is more and more high, and its function is more and more complex. Using traditional design method of Micro-controller will lead to some problems, such as long design cycle, high investment and bad flexibility. Therefore, we need to further improve the manufacturing process and design method to overcome these difficulties.

Nowadays, Field Programmable Gate Array (FPGA) has been rapidly developed in product performance and manufacturing process [7]. System-on-Programmable-Chip (SoPC) technology is the reconfigurable System-on-Chip (SoC) based on programmable logic devices, and it can be designed elastically. System designers can convert traditional board-level system design to chip-level system design. Each functional unit in a system are designed as an Intellectual Property (IP) core and integrated into FPGA. The hardware design becomes more flexible. FPGA brings programmable, upgradable, scalable, extensible advantages to current digital design [8]. Xilinx and Altera are two major FPGA manufacturers which provide a variety of SoC embedded system solutions based on FPGA, e.g., MicroBlaze, Power PC440, NIOS II and ARM Cortex-A9 MPCore [9-12].

In this paper, we use FPGA as the way to realize the Micro-controller design. According to the problems of traditional Micro-controller unable to meet the demands of modern home appliance systems, which require fast coming into the market, various interfaces and timely data processing. This paper proposes a Micro-controller design and implementation scheme based on elastic and pipeline for digital appliance micro controllers. On the one hand, elastic technology is used to improve the efficiency of design and development cycle. On the other hand, using pipeline technique to process instruction pipelining, its purpose is to improve cycle per instruction (CPI), and reduce the design

*Address correspondence to this author at the Department of Computer Science and Technology, Anhui Normal University, 189 Jiuhua East Rd., Wuhu, Anhui Province 241002, P. R. China; Tel: (+86-553) 5910757;
E-mail: long005@mail.ahnu.edu.cn

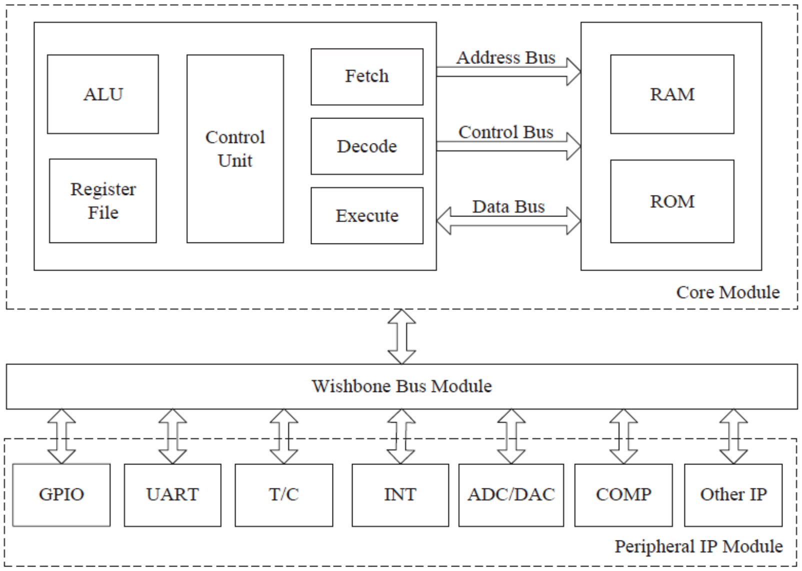


Fig. (1). System structure of YL8MCU.

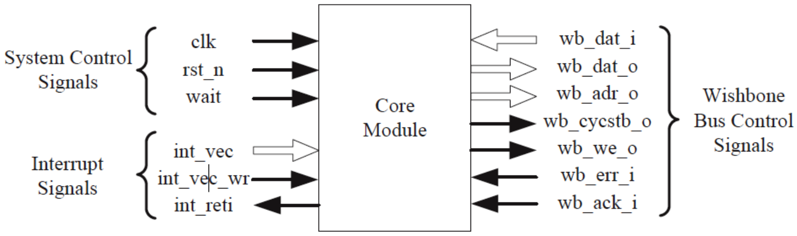


Fig. (2). Top interface of core module.

complexity of Micro-controller. Finally we verify its functionality and requirements *via* function verification and application simulation.

2. YL8MCU MICRO-CONTROLLER DESIGN

2.1. Structure Design

The designed Micro-controller YL8MCU adopts Harvard architecture in which the program and data have their own separate memory and bus. It can pre-fetch the next instruction when an instruction is being executed so as to reduce performance bottlenecks while the program is running. It is divided into the Micro-controller core module, the peripheral IP modules and the bus module. The structure of Micro-controller is shown in Fig. (**1**). Pipeline architecture is used in the Micro-controller core module, and the top interface of core module is shown in Fig. (**2**). Peripheral IP modules include Timer/Counter (T/C), Universal Asynchronous Receiver/Transmitter (UART), Interrupt Controller (INT), General Purpose I/O (GPIO), etc. The data spaces are addressed continuously, including 16 general registers, 32 special function registers, memory spaces of user data and stack spaces. The high-speed addressable register file consists of general registers. Such register file is distinguished by having dedicated reading and writing ports, whereas ordinary register file will usually read and write through the same ports. Similarly, fetch and write-back can be always done in one clock cycle. This effectively simplifies the Micro-controller structure, and completes the data operations conveniently.

The major works of YL8MCU are internal data manipulation and peripheral control. For the internal data manipulation and peripheral control, there exist some differences in digits processing and speed processing. Therefore the independent read/write channel and separate control channel are used in the core module for DEC, ALU, MEM and special function registers like SP, SREG, PC. Peripheral IP modules are controlled by Wishbone bus. It increases the processing capacity, improves the speed, and reduces the bus conflicting and delay caused by bus occupation while the core is reading and writing.

2.2. Pipeline Design

In digital systems, pipeline is an implementation technique whereby multiple instructions are overlapped in

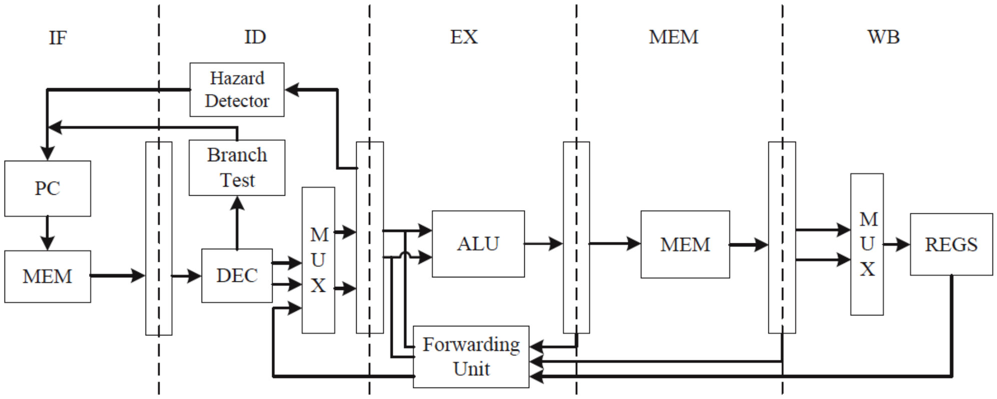


Fig. (3). Pipelined datapath.

execution. RISC architecture incorporate pipeline as a key implementation technique to improve working speed and enhance system stability [13, 14]. The data path of instruction is divided into multiple parts during pipeline in which each part processes data flow at the same time, so as to achieve the purpose of efficient processing. If the pipeline has more number of stages, the logical path of each stage will be shorter, and the performance will be higher. However the delay between different stages will be increased, and it gradually offset the performance gain. Traditional Micro-controller adopts 3-stage pipeline structure. According to the three steps of Micro-controller in the work, the whole circuit is divided into fetch, decode and execute, and insert pipeline between each module. However the scale of three modules and the delay of data path are very difficult to balance. The execute module take too much data operation, such as data reading, data writing, logic operation and so on. Therefore, the path delay is far greater than fetch module and decode module.

This paper adopts 5-stage pipeline. Every instruction in RISC architecture can be implemented using 5 clock cycles. The five stages are instruction fetch (IF), instruction decode (ID), execute (EX), memory access (MEM) and write back (WB). The pipelined datapath is shown in Fig. (**3**).

During the execution of the pipeline, the result of pipelined hazard problem directly affects its performance. There are three types of pipeline hazards: structural hazard, data hazard and control hazard.

(1) As the RISC architecture supports multiple instruction pipeline execution, there is no structural hazard.

(2) Forwarding unit can solve most of data hazards. If the forwarding condition is not satisfied, the operand or operation result will be fetched from the pipeline register. If the forwarding conditions are met, the operation data will be fetched from the data bypass. However forwarding is not useful if the store instruction is followed by one load instruction, or the next instruction fetches the same value written in register by the load instruction. In such a case we need to stall the pipeline. Therefore, we need hazard detection unit to find this type of hazards so as to process them, and soon after, the forwarding unit can continue the follow-up task. If the hazard condition is satisfied, the hazard detection can insert a pipeline bubble after load instruction. In this way, one clock cycle delay will be produced. After that, we can use forwarding unit to complete forwarding.

(3) Control hazard is also called branch hazard. We use pre-branch instruction technique to solve it. The branch judgment circuit and branch address calculation circuit are shifted from the EX state to the ID state. Compared with static branch prediction technique, it can minimize the branch cost and improve the branch efficiency.

2.3. Bus Design

Selection of on-chip bus is the most critical problem for design with semiconductor IP cores [15]. On-chip bus provides a unified interface standard for different IP cores. For this reason, some on-chip bus standard appeared, such as Advanced Micro-controller Bus Architecture (AMBA), Wishbone, Core Connect and Avalon. All of them are shared bus interconnection, and have multiple primary controllers in which request responses are executed synchronously. In particular, the Wishbone bus also can realize Off-chip interconnection and point to point interconnection, and therefore, users can choose the best interconnection according to their own needs. In the arbitration mechanism, AMBA, Core Connect and Avalon are defined by system, while the Wishbone is defined by user. And only the Wishbone bus is completely free. Therefore, we adopt Wishbone bus proposed by Silicore Company. It has the characteristics of flexibility, lightweight and open source [16]. This is accomplished by creating a standard data exchange and bus cycle between IP cores. It does not attempt to regulate the application-specific functions of the IP core. This improves the reusability of design and reliability of the system, reduces the integration problems of system-level chip, and results in faster time-to-market.

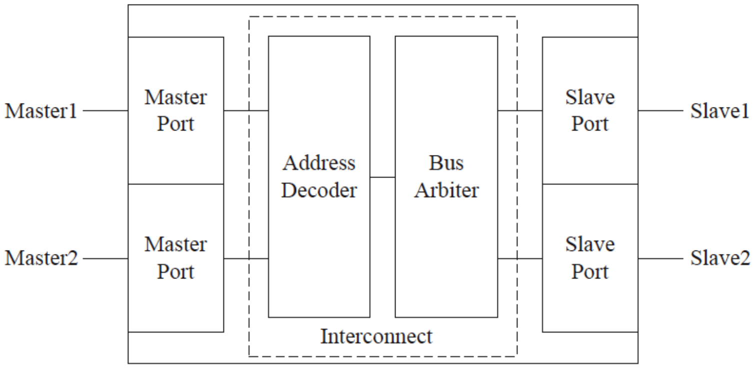


Fig. (4). Shared bus interconnection.

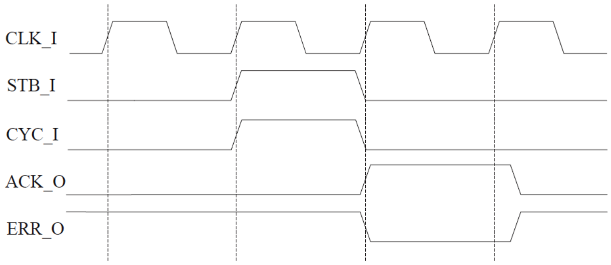


Fig. (5). Wishbone handshaking protocol with master.

The Wishbone interconnection employs Master/Slave architecture for flexible system designs. The modular data bus widths and operand vary in size from 8-bits to 64-bits, and address widths size up to 64-bits. Handshaking protocol allows each IP core to throttle its data transfer speed. The Wishbone interconnection methods support point-to-point, shared bus, crossbar switch, and switched fabric interconnections. The shared bus interconnection is a system where multi-master connect to multi-slave. The bus arbiter works when multi-master will access shared bus. The shared bus interconnection is shown in Fig. (**4**).

The clock signal CLK_I and reset signal RST_I of master and slave come from the external input after synchronization. Address signal ADR_I connects with output of master, data input DAT_I and output DAT_O cross connect with data port of master. Under the control of the write enable signal WE_I, data are exchanged between bus and master. It read the control signal or effective data sent by master device, and return to confirm information, status information, or external data. Error signal ERR_O will be sent to master when there is illegal address signals. Only the normal bus cycle signal CYC_I and data transmission cycle signal STB_I are effective at the same time, and if there is no error signal, the signal ACK_O will be return. Wishbone handshaking protocol with master is shown in Fig. (**5**).

In this design, there is only one master device, and therefore CYC and SYB signals can be combined into a signal CYCSTB, and internal bus arbiter can also be removed. The top interface of Wishbone bus module is shown in Fig. (**6**).

2.4. Elastic Design

The Top-Down design method is adopted by traditional Micro-controller design. The system level is the highest level, and can be divided into sub-modules hierarchically until it can be implemented by EDA tools. Designers must carefully consider the composition of each module in the system and clearly understand the working process of all modules, so as to design the system reasonably. Due to the traditional design process using custom-designed method to design system, each module design must contain the following process: structural design, code design, simulation and verification. If an error occurs in the module verification, the entire procedure must be repeated. A lot of time and effort are spent during modifying the whole structure [17]. The SoC design based on IP core brings a new concept to integrated circuit design, and it will shift the focus from functional design to functional integration. It also can take full advantage of the early accumulation to accelerate the design process and reduce the risk of product development [18, 19].

The YL8MCU Micro-controller design not only make use of reusable IP core to construct the overall architecture but also use parametric and elastic design method to design IP cores and peripherals IP modules. The purpose is to increase the flexibility of the IP core and expand its scope of application, so that designers can change the functions of IP cores and control the size of IP cores according to the needs. Ultimately, the IP cores could be designed as accurately as possible to match the design goals.

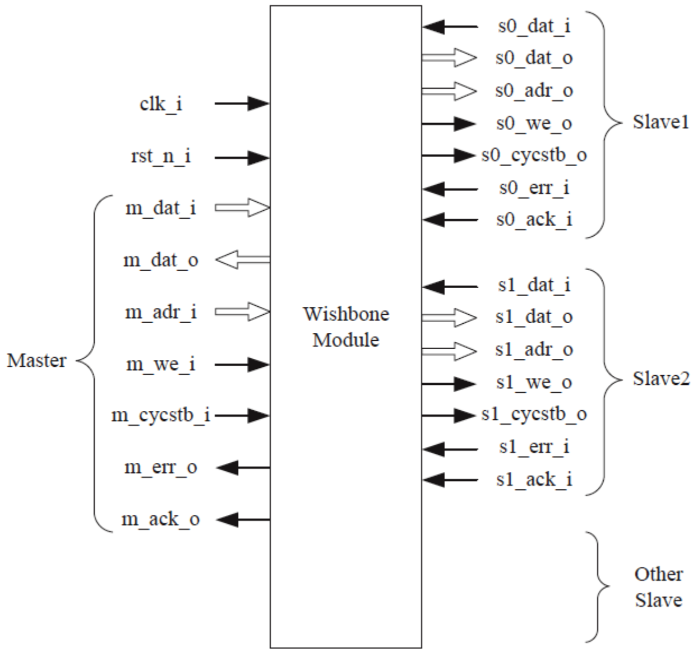


Fig. (6). Top interface of Wishbone bus module.

Parametric and elastic design method can be divided into elasticity of interface parameters and elasticity of function parameters. The former can configure interfaces and timing, and that usually can be seen in Hard Description Language (HDL) source codes. In case of IP core integration, interface parameters are one-time set as static parameters. Parameters specify the interface characteristics of IP core, such as operating mode, data width, address width, delay parameter and so on. The elasticity of function parameters can configure functions of IP core in the HDL source codes. On the one hand, static parameters can set the functions of IP core when it is implemented in FPGA. On the other hand, static parameters can set the way of IP core implementation, e.g., insert the configurable parameters when IP core integrate or save the parameters into the configurable register, and then the configurable register determines the concrete implementation of IP core.

During the design process, IP core program space and data space are designed by parametric technology. Addressing range of program space can be modified by the parameter of program address width between 8KB and 128KB. Addressing range of data space can be modified by the parameter of data address width, between 256B and 64KB. The designer can choose the memory size according the requirement, and control its hardware cost.

The peripheral IP modules such as common GPIO, UART, T/C, INT, which can be configured as optional devices in the top-level module of YL8MCU micro-controller. We define the IP core of GPIO in the top layer if GPIO module needs to be linked. The GPIO module will be added when IP core integration. The GPIO module is defined as follows:

```
IO MCU_IO(
.clk_i(clk)
,.rst_n_i(rst_n)
,.s0_dat_i(s0_dat_o)
,.s0_dat_o(s0_dat_i)
,.s0_adr_i(s0_adr_o)
,.s0_we_i(s0_we_o)
,.s0_cycstb_i(s0_cycstb_o)
,.pina_in(pina_in)
,.pina_out(pina_out)
,.pinb_in(pinb_in)
,.pinb_out(pinb_out)
,.pinc_in(pinc_in)
,.pinc_out(pinc_out)
);
```

3. INSTRUCTION SYSTEM

3.1. Instruction Cycle and Structure

The instruction system determines the basic operations of the Micro-controller. Fig. (**7**) shows the steps of instructions, and it implicates the factors of machine instruction, such as opcode, source operand reference, target operand reference or next instruction reference. The designed system contains 58 instructions, and every instruction can be divided into two

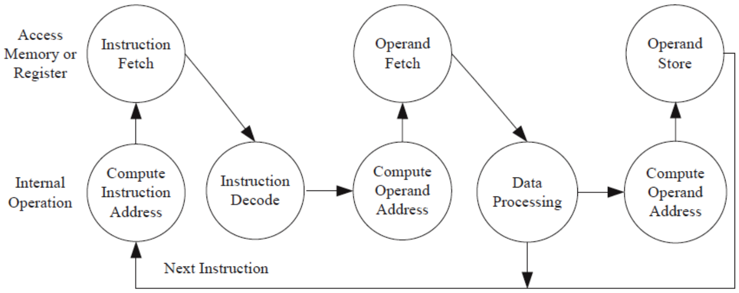


Fig. (7). Instruction cycle status.

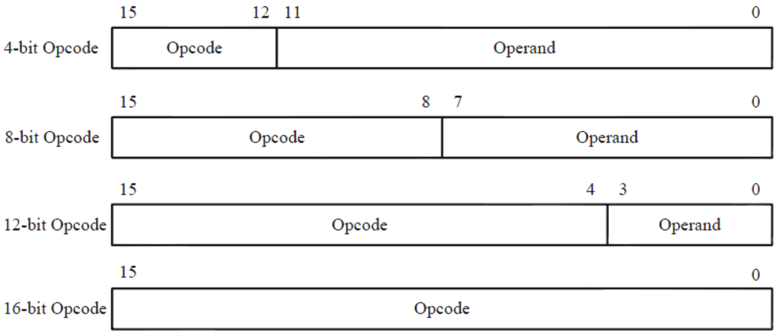


Fig. (8). Instruction formats.

segments. One segment is the opcode segment which represents the function of instruction, and the other one is the operand segment which represents the parameters required for the operation.

The general operation and branch instruction use two operand fields. However some instructions only use a single operand field, e.g., the instruction INC Rd only has an operand for denoting which is used to do the add operation. Especially, some instructions don't give the operand fields, namely the operand implicates in the opcode segment, e.g., the RET instruction and RETI instruction.

3.2. Instruction Format and Addressing Mode

The traditional micro-controllers' instruction complexity is not stable, some instructions' usability is low, and the execution speed is slow. Therefore, some micro-controllers cannot reach the goals of modern home electronics products and system applications. The designed YL8MCU micro-controller adopts RISC instruction system, which has the following advantages such as single cycle executing, pipelining, oriented operation of register groups, simple addressing mode, simple instruction format and hard-wired controlling circuit.

For the advantages that RISC instruction systems have the fixed-size instruction, the designed all instructions' length is 16-bits. This method is propitious to decode instruction, fetch instruction and increase the speed of fetch instructions, by the way, decrease the faults occurred in fetching progress and improve the reliability of designed system. Due to the different operand bits affect the opcode bits, the instruction system has 4 instruction encoding formats: 4-bits opcode, 8-bits opcode, 12-bits opcode and 16-bits opcode. The instruction formats are shown in Fig. (**8**).

The instruction execution for the micro-controller has three steps:

(1) Obtain the operated operand according to address;

(2) Execute the operate corresponding to operand;

(3) Save the results to related register or memory according to address.

Therefore, the program execution is a progress to search operands and do the related operation. Generally, the modes of instruction addressing vary: the more instruction addressing modes, the stronger instruction function. However, complex instruction addressing modes need longer machine cycle, especially worst for the pipeline executing. That is why the instruction addressing is getting easier, and register addressing is getting popular. It can improve the access speed and decrease the pipelined hazard. The instruction addressing of instruction operand has the following 5 categories:

(1) Single register direct addressing: the instruction appoints the content of a register as the operand, and the register is in the specified register groups, e.g., DEC Rd and NEG Rd.

(2) Double register direct addressing: this mode is similar to single register direct addressing. It appoints the content of register Rd and Rr as operands, and store the result in register Rd, e.g., DEC Rd and NEG Rd.

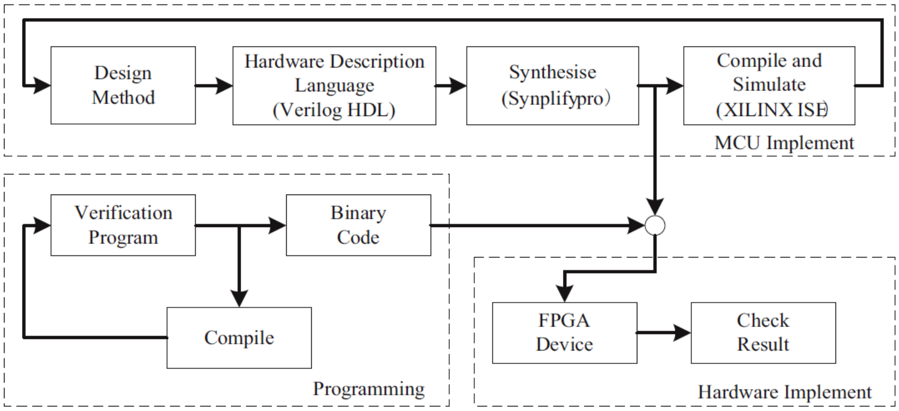


Fig. (9). System implement flowchart.

(3) I/O register direct addressing: this mode appoints the content of a I/O register as one operand, and appoints the content of Rd or Rr register as the other operand. The IN instruction and OUT instruction are used to read and write the data of I/O registers, e.g., IN Rd, P and OUT P, Rr.

(4) Register indirect addressing: this mode regards the 8-bit and 16-bit general register as address pointer register Y for RAM (256B-64KB). The instruction LOAD and STORE are used to read and write the data memory, e.g., LD Rd, @ Rr and LD Rd, Y.

(5) ROM-related addressing: the instruction JMP can jump through the all ROM (8KB) space, and the instruction RJMP can jump to any space of ROM (8KB) space, e.g., JMP addr12 and RJMP rel12.

4. IMPLEMENTATION AND APPLICATION

4.1. System Implementation

The YL8MCU Micro-controller system implementation can be divided into three parts: MCU implementation, programing and hardware implementation as shown in Fig. (**9**). All modules and IP cores adopt Verilog HDL to specify with Xilinx ISE 12.0. This software can executing all the steps from synthesis to simulation, whereas the designed Micro-controller uses the Synplify pro to synthesis because it has the advanced optimization method to reduce the coverage rate of logic netlist and decrease the resource consumption. The program part adopts the compiler developed specifically for this Micro-controller to compile the assembly instructions to binary configuration file for downloading into FPGA. Hardware implementation uses the JTAG to download the correct code to FPGA, and verifies the correctness of the designed system through truth circuit.

Table **1** and Table **2** are the utilization of resources summary and performance report for the designed Micro-controller which has been synthesized and layout. This design contains MCU IP core, GPIO, UART, T/C and interrupt controller. The target FPGA is Xilinx Sparten 3E XC3S1200EFG320 [20]. The reports indicate that the chip resource slices which this Micro-controller have used account for 70% of the target devices resources, and this Micro-controller can work under maximum clock frequency of 73MHz.

Table 1. Utilization of resources summary.

Logic Utilization	Used	Available	Utilization
Number of Slices	6,156	8,672	70%
Number of Slice Registers	3,061	17,344	17%
Number of 4 input LUTs	11,690	17,344	67%
Number of bonded IOBs	57	250	23%
Number of BUFGMUXs	5	24	20%
Number of MULT18X18SIOs	1	28	3%

Table 2. Performance report.

Performance	Index
Minimum period	13.672ns
Maximum Frequency	73.144MHz
Minimum input arrival time before clock	15.541ns
Maximum output required time after clock	4.134ns

4.2. System Verification

Verification work occupies a large proportion in the system design. With the increasing of system scale, the verification vector also exponentially increasing. Although the traditional verification method is used to verify the basic functions of system, it will miss some test space. At the same

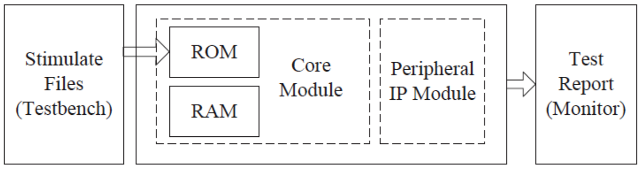
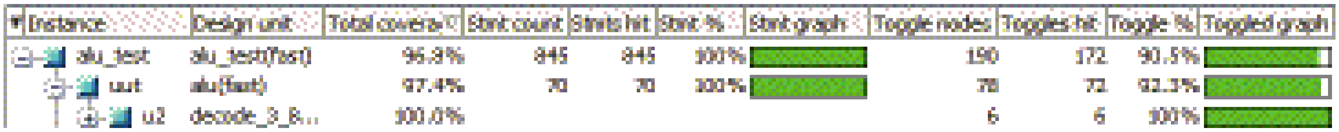


Fig. (10). Verification platform of system integration.



Instance	Design unit	Total covera...	Stmt count	Stmts hit	Stmt %	Stmt graph	Toggle nodes	Toggles hit	Toggle %	Toggled graph
alu_test	alu_test(fast)	96.3%	845	845	100%	🟩	190	172	90.5%	🟩
uut	alu(fast)	97.4%	70	70	100%	🟩	78	72	92.3%	🟩
u2	decode_3_8...	100.0%					6	6	100%	🟩

Fig. (11). Verification result of ALU module based on coverage-driven.

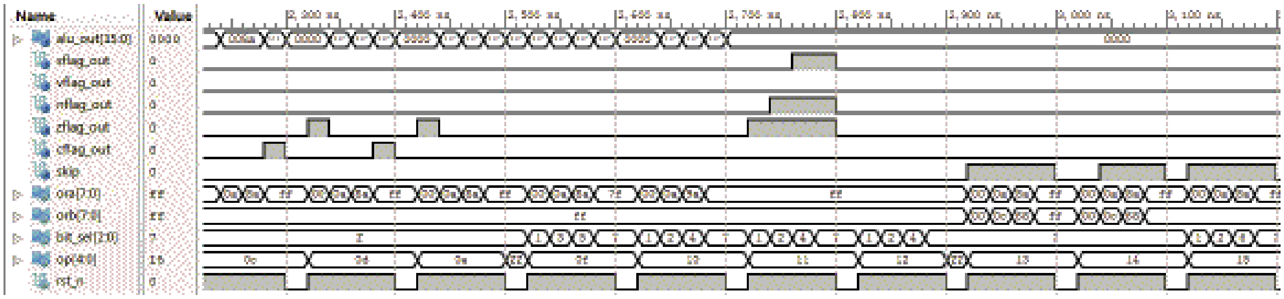


Fig. (12). Simulation waveform of ALU module.

time, it lacks the hierarchical concept, and thus it increases the workload of designer and extends the design cycle of system. At present, the commonly used method is the module verification based on coverage rate. This method analyzes the coverage rate through the existing tools, and then adjusts the verification vectors according to the coverage report.

The paper adopts the IP function module verification based on coverage rate and system integration verification to verify the designed Micro-controller. The former one verifies the module's function via choosing the appropriate incentives produced way according to the scale of IP module. The latter connects the IP core with external interface modules, such as GPIO, UART, T/C and interrupt controller, instantiates the IP cores, and then writes the test bench into the memory, and lastly, executes the binary instruction files that have been stored previously after the system reset [21].

Fig. (**10**) is the verification platform of system integration. The result can be observed by loading stimulation files in this platform, and the stimulation files are machine (.hex) files. The testing result is reserved in the monitor files. System integration verification is mainly used to do function verification for the designed instruction system.

Fig. (**11**) is the verification result of ALU module based on coverage-driven. The code coverage hits 100%, and the coverage of flip-flop and total module is higher than 90%. The test vectors given by the test bench are complete and the logic timing function of ALU module reach the expected goals. Fig. (**12**) is the simulation waveform of ALU module.

4.3. Comparison and Analysis

In the field of Micro-controller, many manufacturers have their products, including Microchip Company, STC Company and ATMEL Company. The products of these companies are most widely used. The three companies also have 8-bit Micro-controllers which are representative. The Micro-controller PIC16F914, STC90C51R and AT90S1200-12 have similar functions. And these Micro-controllers are used to compare with the YL8MCU which is designed in this paper according to all performance indicators. Table **3** shows the comparisons between the designed YL8MCU and other micro-controllers.

According to the comparisons, YL8MCU has simplified instruction system, the users do not memory a variety of complex instructions and the reliability is improved. The number of external ports is changed with I/O ports. The number of external pins is 20 when I/O ports are 8 bits. The number of external pins will be up to 36 when the I/O ports increased to 24 bits. YL8MCU can be configured conveniently and flexibly on ROM and RAM, so as to match varies of application demands. Because YL8MCU adopts elastic design method , we can increase or decrease different

Table 3. Comparisons of different micro-controllers.

MCU	PIC16F914	STC90C51RC	AT90S1200-12	YL8MCU
Instructions	35	111	89	58
Pins	40	40	20	16-32
ROM	7KB	4KB	1KB	8-128KB
RAM	256B	256B	None	256B-64KB
I/O	3(35pins)	3(35pins)	2(15pins)	1-3(8-24pins)
Performance	20MHz	40MHz	8MHz	73MHz

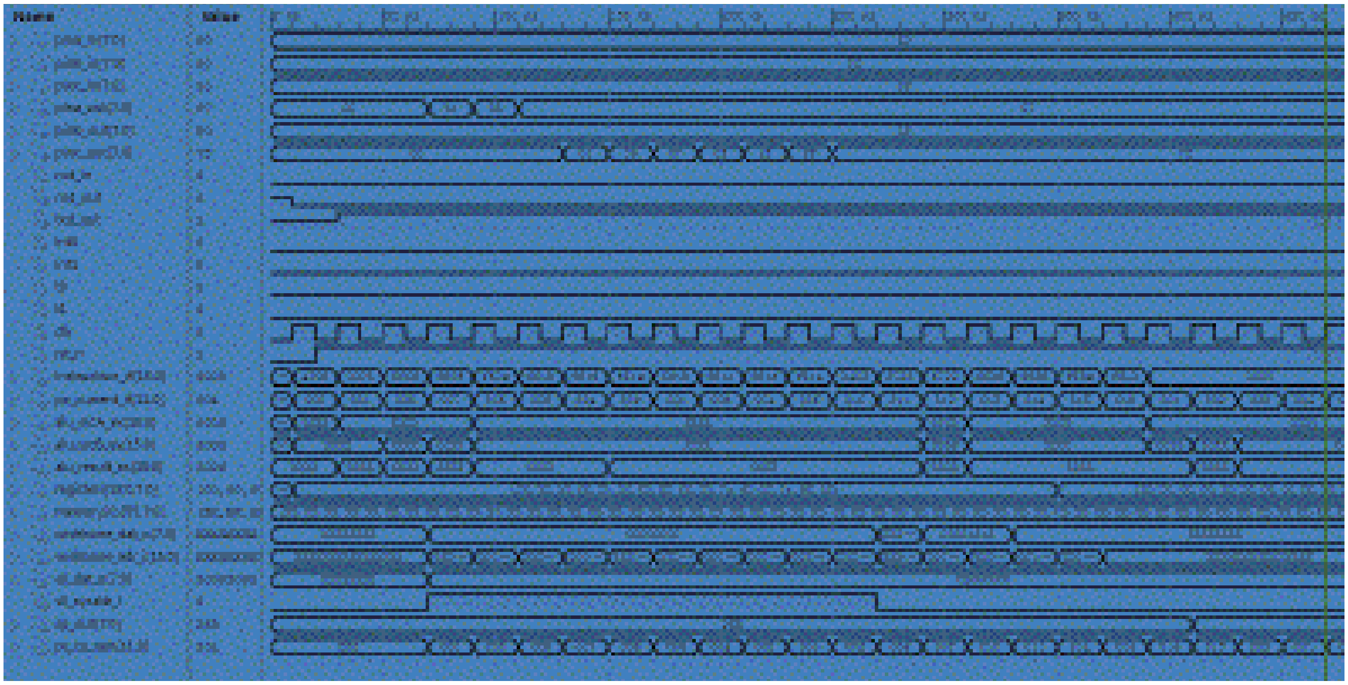


Fig. (13). Simulation waveform of washing machine.

function. The working frequency of YL8MCU can reach 73MHz. Compared with the other three Micro-controllers, it has certain performance advantages.

4.4. System Application

As important members of digital appliances, Washing machines become the essential electronic products that replace the artificial domestic activities, such as automatic washing machines. The designed YL8MCU Micro-controller is used to implement full automatic washing machine, that is, the manpower is not needed in the transforming process during washing, rinsing and dehydration. This system has higher intelligence and reliability.

We use ISIM of Xilinx ISE to simulate the developed MCU. The simulation result can match the function design goals, and shows the MCU can control the whole progress well, which contain input user data for washing machine, washing, rinsing and dehydration. Fig. (**13**) shows the simulation the waveform of the washing machine.

At last, this paper uses Xilinx ISE to choose device and assign pins according to the correct digital appliance Micro-controller, and then downloads to XC3S1200EFG320. The output results consist of simulation results.

CONCLUSION AND FUTURE WORKS

Our work investigates an elastic pipeline design of high performance Micro-controller for digital home appliance signal processing, completes the design of instruction set and hardware configuration, describes the implement of parameterized, elastic IP core and pipelining technology, and combines various verification methods with application examples to simulate the designed system. The design scheme provides a reference for future Micro-controller development.

Two works can be carried forward in the future. One is to study the on-chip bus model based on IP core, integrate more functions on IP core, and then achieve a complete SoC system. The other one is to strengthen the compiler's function, e.g., mainly the work is to complete the process of compiling C code to binary.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 61370050), Anhui Provincial Natural Science Foundation (No. 1308085QF118), University Natural Science Research Project of Anhui Province (No. KJ2014A084), Wuhu City Science and Technology Key Project (No. 2014cxy04), and Anhui Normal University Postdoctoral Project (No.161-071214).

REFERENCES

[1] Z. K. Huang, and K. W. Chau, "A new image thresholding method based on Gaussian mixture model", *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 899-907, 2008.

[2] R. Taormina, K. Chua, and R. Sethi, "Artificial Neural Network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon", *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1670-1676, 2012.

[3] C. L. Wu, K. W. Chau, and Y. S. Li, "Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques", W*ater Resources Research*, vol. 45, no. 8, pp. 1670-1676, 2009.

[4] J. Zhang, and K. W. Chau, "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization", *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 840-858, 2009.

[5] C. Cheng, K. W. Chau, and Y. Sun, "Long-term prediction of discharges in Manwan Reservoir using artificial neural network models", *Advances in Neural Networks–ISNN*, pp. 1040-1045, 2005.

[6] K. W. Chau, "Application of a PSO-based neural network in analysis of outcomes of construction claims", *Automation in Construction*, vol. 16, no. 5, pp. 642-646, 2007.

[7] E. Monmasson, and M. N. Cirstea, "FPGA design methodology for industrial control systems-a review", *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824-1842, 2007.

[8] D. Saha, and S. Sur-Kolay, "SoC: a real platform for IP reuse, IP infringement, and IP protection", *VLSI Design*, vol. 2011, no. 5, pp. 1-10, 2011.

[9] PowerPC Processor Reference Guide. Version 1.3. Xilinx Corporation, 2010.

[10] MicroBlaze Processor Reference Guide. Version14.1. Xilinx Corporation, 2012.

[11] Nios II Processor Reference Handbook. Version11.0. Altera Corporation, 2011.

[12] SoC FPGA. ARM Cortex-A9 MPCore Processor Advance Information Brief. Altera Corporation, 2012.

[13] A. Yazdanbakhsh, M. E. Salehi, and S. M. Fakhraie, "Customized pipeline and instruction set architecture for embedded processing engines", *The Journal of Supercomputing*, vol. 6, no. 2, pp. 948-977, 2014.

[14] C. H. Cheng, "Design example of useful memory latency for developing a hazard preventive pipeline high-performance embedded-microprocessor", *VLSI Design*, vol. 2013, no. 6, pp. 1-10, 2013.

[15] E. Salminen, V. Lahtinen, and K. Kuusilinna, "Overview of bus-based system-on-chip interconnections", In: *Proceedings of the IEEE International Symposium on Circuits and Systems* (*ISCAS 2002*), vol.II, Phoenix-Scottsdale, Arizona, USA, May 26-29, pp. 372-375, 2002.

[16] Wishbone System-on-Chip (SoC). Interconnection Architecture for Portable IP Cores. Version B4. OpenCores, 2010.

[17] R. Saleh, S. Wilton, and S. Mirabbasi, "System-on-chip: reuse and integration", *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1050-1069, 2006.

[18] P. Coussy, A. Baganne, and E. Martin, "A design methodology for integrating IP into SoC systems", In: *Proceedings of the IEEE Conference on Custom Integrated Circuits*, 2002, pp. 307-310.

[19] P. Trivedi, and D. Asati, "A new approach for 8 bit core processor & its IP design for small scale system", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 7, pp. 121-124, 2012.

[20] Spartan-3E FPGA. Family:Complete Data Sheet. Xilinx Corporation, 2008.

[21] J. Ributzka, Y. Hayashi, and F. Chen, "DEEP: An iterative fpga-based many-core emulation system for chip verification and architecture research", In: *Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2011, pp. 115-118.

Received: May 07, 2015 Revised: May 19, 2015 Accepted: June 20, 2015