2002년도 대한전자공학회/대한전기학회 시스템 및 제어분야 합동 추계학술대회 논문집(2002.11.30)

The implementation of DC motor controller based on SoC

Sung Ui Lee, Jae Kwan Seo, Sung Nam Oh, Kyi Kae Park, Kab Il Kim
Dept. of Electrical Engineering, Myong-Ji University
Tel : 031-330-6356 / Cell phone : 016-438-6426
E-mail : sulee74@msn.com

Abstract

In this paper, DC motor controller has been designed by using SoC. SoC is short for System on a chip. This is a methodology that both a processor and some applications are integrated in a chip. In order to design this system based on SoC, PIC 16C57 has been selected as a processor because it has not too many instruction sets and simple data path named a harvard structure. And motor control module has been programmed by using VHDL. The advantages of the design based on SoC are as follows: simple structure, high speed working, easily verifying and simulating the system.

I. Introduction

At the beginning of 1980's, according to the industrial development of semiconductor, It has been the relative importance on the rise and fall of the nation. Korea assumed the leadership in the filed of memory chip within very short time. But Korea still lagged behind U.S, Europe, and some countries in the filed of non-memory. The comparativeness of IT industry plays an important role in 21th century. So it is no wonder the improvement of digital circuit including both memory and non-memory design skill is very important.

There have been several remarkable changes in the technology of a digital circuit design. Firstly, a digital circuit had been designed by using only transistors. Secondly, a digital circuit had been designed by the cell such as gate and flip-flop(SSI Level). And thirdly, functional modules such as ALU or REGISTER had been designed(LSI Level). and next, complex modules like as PROCESSOR or MEMORY had designed(VLSI Level). Recently, system modules which is needed over 100k gate in a chip has been designed(SoC Level).

VHDL or Verilog HDL is used in the level of system such as SoC. VHDL describes the behavior of digital circuits. The appearance and necessity of VHDL is as follows. In accordance with more complexity and increasing of digital circuit's integrated rate, it is necessary to design and test the huge circuits altogether. So VHDL has been developed by the department of defense in U.S and becomes the standard language for hardware description officially-adopted by IEEE. Now, it is widely used in Europe, Asia including Korea and Japan due to the strengths that easy to design on a large system and reuse IP. Therefore, from the view of digital circuit design, the important reasons of using VHDL boardly are as follow two points. One is to be reduced 30%-50% on a term of design comparing with the schematic capture which was previous design method. Another is a fact that the probability of First Success is much higher.

A design method that large scale gates over 100k integrate in a chip, such as SoC, has a simple structure and high speed working characteristic because a processor and some application modules are been designed together in a chip.

The programmed module using VHDL is IP(Intellectual Property). It is a fact that nowadays our country depends on export when we get very expensive IP . But the trend of semiconductor design technology develops day after day. Therefore, independent development of IP is very important.

This paper presents the design of DC motor controller using FPGA on the basis of SoC.

II. Overall structure of DC motor controller

This paper is divided into two parts of module: the processor module and the motor control module. Figure 1 shows the block diagram of overall system.

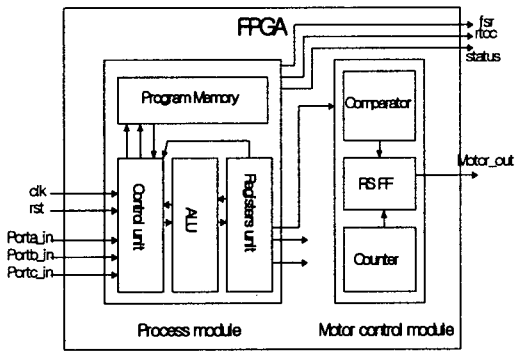

Figure 1. Block diagram of overall system.

In this paper, One of Flex10k series-EPF10K50RC240-3 is used as FPGA, and Max-plus 9.5 and Active HDL are also used as VHDL compiler, and Symplify and ModelSIM 5.5 are used as a verification tool for simulation.

2.1 Processor Module

(1) Overall block diagram

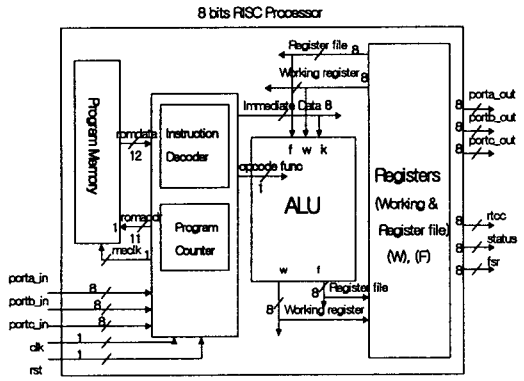

Figure 2. Overall block diagram of Processor

In this paper, PIC16C57 of Micro-chip Corporation is selected as the model of the processor. PIC16C57 is a 8bits RISC(Reduced Instruction Set Code). It has not too many instruction sets(33) and also data path that is separated into two paths for program memory and data memory named a harvard structure.

Figure 2 shows the block diagram of overall processor. Looking at Figure 2, Register File has used as a substitute of Data Memory. As two MUX are added in parts of input/output of ALU module, Data path is separated into BUS_A and BUS_B to work more efficiently. System clock enters this module as input and it is divided into 4 clocks in a period. The divided clocks control the each module. Namely, system clock is 4MHz and processing time of an instruction is 1us. 8bits three input/output have also been designed.

(2) Control module

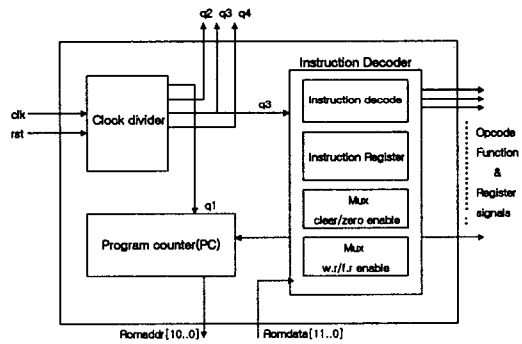

Figure 3. Block diagram of Control module

The control module generates several control signals for ALU module and Registers module through fetching and decoding.

Looking at Figure 3, clock divider module makes internal clocks such as q1, q2, q3, q4. q1 is input of Program counter module ,and if q2 happens, instruction decoder module fetches a data from program memory, Next by q3, instruction decoder module works and generates several control signals for ALU and Registers File module. After q4 enters Registers module, it can work.

One of the input of instruction decoder module -Romdata is 12 bits. It is classified into three types

of instructions which are shown in table1.

Byte-Oriented	11 6 5 4 0 opcode | d | f d : Indicate destination d='0' : Working Register d='1' : Register File f : Register File Address	ADDWF, ANDWF, CRLF,CLRW,COMF, DECF,DECFSZ, INCF, INCFSZ, IORWF,MOVF,MOVWF, NOP,RLF,RRF,SUBWF, SWAPF,XORWF
Bit-Oriented	11 8 7 5 4 0 opcode | b | f b : bit number f : Register File Address	BCF BSF BTFSC BTFSS
Literal	11 8 7 0 opcode | k k : immediate data	ANDLW,CALL,CLRWDT, GOTO,IORLW,MOVLW, OPTION,RETLW, SLEEP,TRIS,XORLW

Table 1. Three instruction types

The instruction decoder module has been designed through an analysis of 33 instructions. OPCODE has saved package file in order to refer this at any files.

(3) ALU module

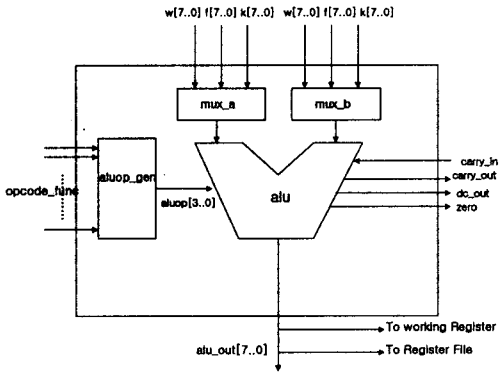

Figure 4. Block diagram of ALU module

The ALU module calculates two values between Registers module's output and control module's constant number(k). Finally these results go into Registers module again to be saved.

Looking at Figure 4, aluop[3..0] presents the result of what operation ALU does. So aluop, mux_a and mux_b go into ALU module. And after opration is over, the results of operation go to Working Register or Registers File.

When ALU module is been designed, the library must refer to "std_logic_1164.vhd" due to figure the subtraction of 2's complement. Also ALU module has 13 arithmetic and logic operations.

(4) Register module

The register module saves the operation results. There are Working Register, Registers File and SFR. SFR contains ioport Register, status Register and fsr Register.

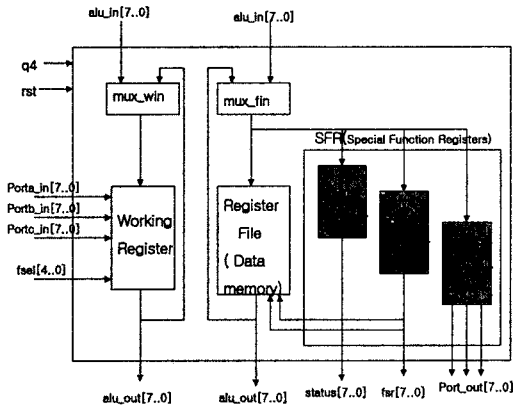

Figure 5. Block diagram of Register module

Looking at Figure 5, Register module works after q4 enters into it, and this module has two MUX in order to send ALU's results to both Working Register and Registers File through different data path separately.

fsel[4..0] is the input of Working Register, and its role is to select one of registers. Table 2 shows the part of program about selecting method.

```
Case fsel is
        when "00011" =>  reg_status<='1';
        when "00100" =>  reg_fsr<='1';
        when "00111" =>  reg_ioport_c<='1';
        when "01100" =>  reg_ioport_b<='1';
        when "01011" =>  reg_ioport_a<='1';
End case;
```

Table 2. Selection of Registers

SFR Register's outputs have some information about system. Addressing of bank selection is done by 6bit and 5 bit of fsr.

2.2 DC motor control module

In this paper, PWM(Pulse Width Modulation) method is used to run DC motor. DC motor has the steps from 0 to 255 because the input of motor is 8bits. And select_generating module was designed due to selecting one of clock-wise/counter clock-wise rotations.

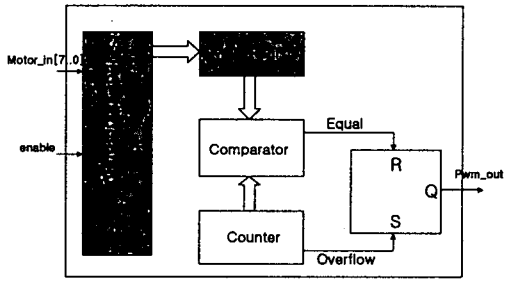

Figure 6. Block diagram of Motor Control Module

Looking at Figure 6, It is available to select one of rotation method by the enable signal in select_generating module. pwm register value and 8 bits counter's value enter the comparator module. At the same time, if two values is equal, pwm_out is '1'. otherwise if two values is not equal, pwm_out is '0'. So motor runs with various speeds during a period because of changing of duty-rate. The reason of using FS-FlipFlop is to load pwm register's value to previous value.

2.3 Simulation and Test of overall system

(1) Compiling

```
--motor_con_top.rpt--

***** Project compilation was successful
** DEVICE SUMMARY **
Chip/                Input Output Bidir Memory Memory LCs
POF    Device        Pins Pins  Pins  Bits % Utilized LCs % Utilized
Motor_con_top
    EPF10K30RC208-3  38   60    0     0      0 %      1361   78 %
User Pins:           38   60    0
****Embedded         Column    Row ******
Array   Embedded Interconnect Interconnect      Read/    External
Block   Cells    Driven      Driven      Clocks Write    Interconnect
Total dedicated input pins used:          4/6    ( 66%)
Total I/O pins used:                     94/141  ( 66%)
Total logic cells used:                1361/1728  ( 78%)
Total embedded cells used:                0/48   (  0%)
Total EABs used:                          0/6    (  0%)
Average fan-in:                        3.65/4    ( 91%)
Total fan-in:                       4970/6912    ( 71%)
```

```
Compilation Times
-----------------
Compiler Netlist Extractor          00:00:15
Database Builder                    00:00:59
Logic Synthesizer                   00:01:40
Partitioner                         00:00:23
Fitter                              00:00:48
Timing SNF Extractor                00:00:03
Assembler                           00:00:02
------------------------------      --------
Total Time                          00:04:11
```

Table 3. Compiling report

Table 3 shows the part of report file that is made after compiling. It presents the used pins, used rate of logic cell, and estimated time during compiling.

(2) Simulation and Test

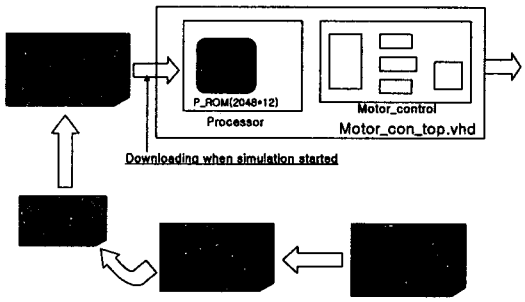

Figure 7. Simulation and Test of overall system

Looking at Figure 7, Program memory is in motor_con_top module which equals to overall system.

First of all, an algorithm of motor control was programmed by assembler. Next hex and ttb file got through compiling by using RC8ASM.EXE. Next the value of ttb file put in P_ROM(2048*12). Finally we got a result which is needed after simulation.

Figure 8 shows array structure of P_ROM.vhd.

```
entity prom is
  generic ( ,
    DEPTH : in integer := 2048;
    DATA_WIDTH : in integer := 12;
    ADDR_WIDTH : in integer := 11 );
.....
package table_rom_pack is
          Type rom_type is array ( 0 to 2047 ) of STD_LOGIC_VECTOR ( 11 downto 0 );

          constant ROM : rom_type,:= (
                    "110011111101",
                    "000000101010",
                    "001010101010",
                                .
                                .
                                .
                    "000000000000");
End package;
```

Figure 8. Array structure of P_ROM.vhd

III. Conclusion and Future

DC motor controller has been designed based on SoC. A processor and motor control module have been integrated in a chip based on SoC. 8bits PIC RISC Processor has been used as processor. And the modules have been programmed by using VHDL. Then we can draw a conclusion: with the method mentioned in this paper. the design can work more fast, can simulate and verify easily, and reuse easily.

In the future works, we have a plan to design another system by reusing IP. When we reuse IP, it is very important to verify the exact work of system in order to prevent the serious wrong work of system in advance. Therefore, the fact that more verification tools are needed is considered.

Reference

[1] M.Mano, "Computer System Architecture", second edition, Prentice Hall, 1993.

[2] Lee Kang-hyun, Cho Tae-won, Song Sang-seup, "Design of computer architecture", Dae-young-sa, 2002.

[3] Charles H, Roth,Jr, "Digital system design using VHDL", Thomson learning, pp219-227, 383-405, 1998.

[4] S.Sjoholm, L.Lindh, "VHDL for designers", Prentice Hall, pp30-162, 1997.

[5] Jarrod D.Luker and Vinod B.Prasad. "RISC System design in an FPGA", 2001 IEEE, pp532-536, 2001.

[6] Sin Keul-ho, "One chip micom PIC16C5X handbok", Sung-an-dang,1994.

[7] Frank Lien, Jason Feng, Eddy Huang, Chung Sun, Tong Liu, Naihui Liao, David Hightower, "A Hardware/software solution for Embeddable FPGA", IEEE 2001 Custom Integrated Circuits Conference, pp71-74, 2001.

[8] Jin-Fu Li, Hsin-Jung Huang, Jeng-Bin Chen, Chin-Ping Su, Cheng-Wen Wu, Chnang Cheng, Shao-I Chen, Chi-Yi Hwang, and Hsiao-Ping Lin, "A hierarchial test scheme for system-on-chip design", 2002. Design, Automation and Test in Europe Conference and Exhibition(DATE'02), pp1530-1534, 2002.