I/O design and FPGA implementation based on DW8051

Bin Zhang, Zhengyan Wang, Yuliang Guo

College of Electtronic Information, Qingdao University, Qingdao 266071, China

Keywords: DW8051; FPGA; IO interface; Verilog HDL

Abstract: Based on the IP (Intellectual Property) core DW8051 Core provided by Synopsys, the IO interface (Input/Output Interface) for transmitting data between Core and peripherals is added. The interface is designed and implemented by the hardware description language Verilog HDL as DW8051. A peripheral that is mounted on the DW8051's bus. The design is fully compatible with the Intel 8051 series of microprocessors, which can be controlled by the user program on the DW8051 and verified for correctness on Altera's FPGA chip.

1. Introduction

With the continuous development of semiconductor technology, more and more components are integrated per unit area, and the size and power consumption of the chip are continuously decreasing. However, the performance has been continuously improved, and more and more design solutions are available at this stage. Starting from the system level, the chip structure and algorithm are fully integrated with the embedded system as the core, and finally realize the required functions on a whole chip. This article is based on the idea of this design, based on Synopsys' DW8051 soft core, designed a general-purpose I / O interface to achieve data interaction between DW8051 and other peripherals.

2. System Overview

DW8051_core is a soft core designed and developed by SYNOPSYS, which can be configured according to actual needs. The soft core is compatible with the industry standard 803x/805x microcontroller [1]. The DW8051_core has higher performance than the standard 8051 microcontroller. Its instruction cycle is only 4 clocks, and the execution time of the instruction is 2.5 times faster than the standard 8051 microcontroller. The address range of RAM and ROM can be configured by internal parameters. As a synthesizable soft core, DW8051_core only provides basic standard 8051 microcontroller peripherals, such as: interrupt control, timer/counter, serial port. For peripheral expansion, the DW8051_core provides a set of peripheral bus SFRs (Special Function Registers) through which the user can freely customize the peripheral interface.

The I/O interface (Input/Output Interface) studied in this paper is mainly used to exchange information between the host and external devices. General external input/output devices, such as mice, keyboards, monitors, etc., are much slower in data transfer speed than high-speed CPUs. Moreover, different devices have different data transmission formats and signal formats, so the peripherals cannot be directly connected to the central processing unit. In this case, corresponding circuits are needed to complete the speed matching, signal conversion between them, and complete some Some control functions. These buffer circuits between the host and peripherals are called I/O interface circuits[2].

In the design of this paper, the I/O interface circuit interacts with the DW8051_core through the data bus SFR of the DW8051. Its system block diagram is shown in Figure 1.

DOI: 10.25236/dpaic.2018.012

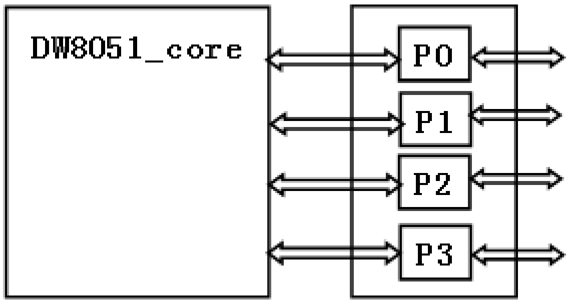

Figure 1 system structure diagram

3. Design implementation

3.1 P0 port

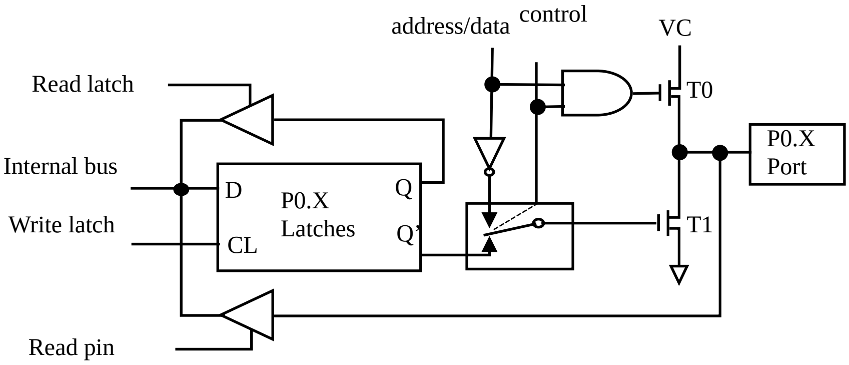

Figure 2 P0 Bit structure diagram

In the standard 8051 microcontroller, the P0 port is a three-state bidirectional port, which can be used both as a normal I/O and as an address/data multiplexing bus. The bit structure diagram of P0 is shown in Figure 2.

Analysis of Figure 2 shows that the bit structure of P0 includes two MOS transistors to form the push-pull structure to realize the output drive of P0. Two tri-state buffers are used to implement the read pin, the read latch, and one output latch. , a multiplexer control circuit[3].

If it is necessary to access the external storage space, the P0 port is used as the address/data multiplexing bus, the multiplex switch is placed on it, and the signal state of the address/data is transmitted to the T1 through the inverter; at the same time, the door is unlocked, and the gate is The output signal is completely dependent on the state of the address/data line, thus enabling the output of the address/data signal to the P0.X pin.

If the MCU has its own ROM space, the P0 port can be used as a normal I/O port, and the multi-way switch is placed below. At this time, the internal bus data can be reversed according to the input D and P0.X of P0.X. The output terminal Q', the multiplexer, and the T1 are sequentially output to the P0.X pin. When P0 is used as the input port, the CPU will automatically write 1 to the P0.X latch, and then the two MOS transistors T0 and T1 will be turned off. At this point, the signal of the pin can be read to the internal bus through the tri-state input buffer. Thereby completing the read data operation.

By analyzing the internal circuit structure of the P0 port of the standard 8051 MCU, using the knowledge of the hardware circuit design, combined with the instructions of the DW8051, the hardware circuit diagram of P0 can be obtained.

The main pin signals in the P0 hardware circuit diagram are described in Table 1:

Table 1 Description of signal lines at port p0

p0_addr_data_n	Address/data selection signal of port p0
mem_addr	External RAM or ROM address line
mem_data_out	External RAM data output line
p0_mem_reg_n	P0 port output register and address/data selection signal line
sfr_data_out	DW8051 data output bus
sfr_wr	DW8051 data output bus write indication signal
sfr_addr	DW8051 peripheral address bus
80h	Device address of port p0
sfr_rd	DW8051 data input bus read indication signal
port_pin_reg_n	Read pin and read latch selection signal lines
sfr_cs0	The chip select signal of the data input from the p0 port to the data bus
sfr_data_in_p0	P0 data line for inputting data to the data bus
mem_data_in	External RAM or ROM to input data signal lines to the cpu
mem_rd_n	External RAM read indication signal
p0	double-sided p1 port

According to the hardware circuit diagram of the P0 port, the hardware circuit description language Verilog is used to design the hardware circuit description of the P0 port. The following is the main description code of P0.0:

```
 assign sfr_cs0=(sfr_rd&&(sfr_addr==8'h80))?1'b1:1'b0;
 wire [7:0] exmem_out;
 assign exmem_out = p0_addr_data_n? mem_addr[7:0]:mem_data_out;
 inout_pad p0_port0(
                              .I((p0_mem_reg_n & exmem_out[0])? 1'b1:1'b0),
                              .OEN ((p0_mem_reg_n)? 1'b0:p0_l[0]),
                              .PAD (p0[0]),
                              .C    ()
                            );
 assign mem_data_in = (mem_rd_n==1'b1)? 8'bz : p0;
if(sfr_wr)
    begin
      case(sfr_addr)
            8'h80: begin p0_l<=sfr_data_out; end
            default: ;
      endcase
    end
```

The module description of inout_pad is as follows. This module mainly implements the bidirectional input function of the I/O port.

```
module inout_pad (I, OEN, PAD, C);
      input I, OEN;
      inout PAD;
      output C;
      bufif0   (PAD, I, OEN);
      buf      (C, PAD);
endmodule
```

3.2 P1 Port

In the standard 8051, the P1 port function is relatively simple, generally used as a normal I/O port, and can realize bit operation at the same time, that is, each bit of the P1 port can realize independent input and output, and its bit structure is as shown in FIG 3 .

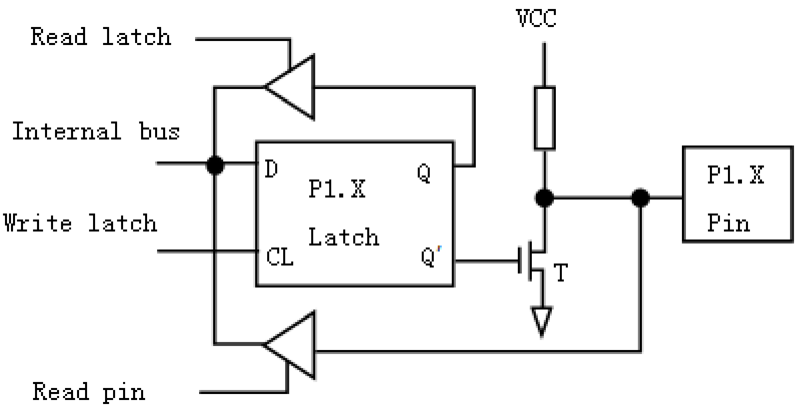

Figure 3 P1 mouth structure diagram

When the P1 port is output as shown in Figure 3, the data is output from the internal bus through the input terminal D of the P1.X latch, the inverted output of the P1.X latch, and the T and P1.X pins. When inputting P1 port, it must first write 1 to the latch to make the factory effect transistor cut off, then the data is read into the internal bus[4] through the P1.X pin and three buffer gates. By analyzing the data flow of the standard 8051 MCU P1 port, combined with the DW8051 core user manual, the hardware structure of the P1 port is obtained by the design method of the hardware circuit.

The main pin descriptions of the hardware circuit diagram of the P1 port are shown in Table 2. The pins that have been described in the P0 port are not repeated here.

Table 2 Description of the signal line of the P1 port

90H	P1 port device address
sfr_cs1	The chip select signal of the data input from the p1 port to the data bus
sfr_data_in_p1	P1 data line for inputting data to the data bus
p1	double-sided p1 port

According to the hardware structure diagram of the P1 port, the hardware circuit description language Verilog is used to design the hardware circuit description of the P1 port. The following is the main description code of P1.0.

```
assign sfr_cs1=(sfr_rd&&(sfr_addr==8'h90))?1'b1:1'b0;
inout_pad p1_port0(.I(1'b0),.OEN(p1_l[0]),.PAD(p1[0]),.C());
assign sfr_data_in    =    sfr_cs1?(port_pin_reg_n? p1 : p1_l)
if(sfr_wr)
   begin
      case(sfr_addr)
           8'h90: begin p1_l<=sfr_data_out; end
           default: ;
      endcase
    end
```

3.3 P2 Port

The P2 port of the standard 8051 is a dual function port. The P2 port can be used as a normal I/O port or as a high 8 bit of the address bus. The bit structure of P2 is shown in Fig. 6.

By analyzing Figure 4, it can be seen that the one-bit circuit of P2 includes the P2.X latch that latches the data bits, and two three-state input data buffers, which are mainly used for reading the data buffer of the pin and the latch, one more Road switch MUX, output drive circuit. When the P2 port is used as a normal I/O port, the multiplexer MUX is turned to the left, and the output Q terminal of the P2.X latch is connected to the output driving circuit. At this time, the working mode is exactly the same as that of the P1 port; when P2 When the port is used as an address bus, the multiplexer MUX is turned to the right, and a bit of the upper 8-bit address can be input through the address line. It can be seen from the figure that the MUX switch can only be played on one side at the same time, so that the P2 port can only use one function at the same time, that is, if P2 is used as the address line, it can no longer be used as a normal I/O port. Also. By analyzing the P2 port function of the standard 8051, combined with the design knowledge of the hardware circuit, we can design the P2 port hardware circuit structure[5].

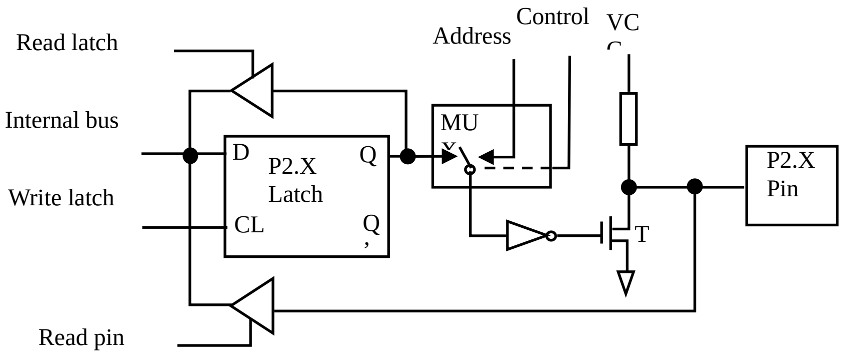

Figure 4 Bit structure diagram of P2

The main pin descriptions of the hardware circuit diagram of the P2 port are shown in Table 3. The pins that have been described above are not repeated here.

Table 3 Description of P2 port signal lines

p2_addr_data_n	Address/data selection signal of port p2
p2_mem_reg_n	P2 port output register and address/data selection signal line
A0h	P2 port device address
sfr_cs2	Chip select signal for input data from p2 port to data bus
sfr_data_in_p2	P2 port inputs data data lines to the data bus
p2	double-sided p2 port

According to the hardware circuit diagram of the P2 port, the hardware circuit design language Verilog is used to design the P2 port hardware circuit description. The following is the main description of P2.0[6].

```
  assign sfr_cs2=(sfr_rd&&(sfr_addr==8'ha0))?1'b1:1'b0;
  assign sfr_data_in = sfr_cs2?(port_pin_reg_n? p2 : p2_l): 8'bz;
 inout_pad p2_port0(.I ((p2_mem_reg_n       & mem_addr[8])? 1'b1:1'b0),
                        .OEN ((p2_mem_reg_n      )? 1'b0:p2_l[0]),
                        .PAD (p2[0]),
                        .C     ()
                        );
 if(sfr_wr)
      begin
         case(sfr_addr)
            8'ha0: begin p2_l<=sfr_data_out; end
            default: ;
         endcase
       end
```

3.4 P3 port

The P3 port in the standard 8051 microcontroller is a dual function port. Each of its bits can be used as a second output pin or a second input pin. The circuit structure of a P3 port is shown in Figure 5.

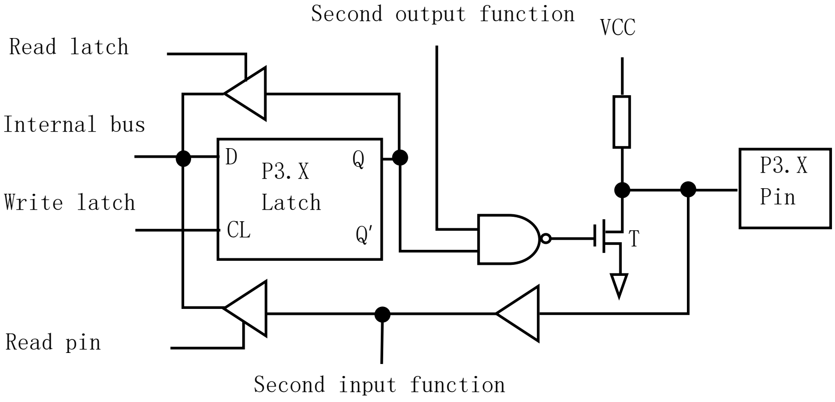

Figure 5 Bit structure of port P3

By analyzing Figure 5, it is seen that the P3 port of the standard 8051 includes a data latch P3.X, three input buffers, and an output driving circuit. When P3 is used as a normal I/O port, the output function control line must be high. At this time, the latch P3.X and the output of the NAND gate are consistent, and the output operation of the P3.X pin is the same as that of the P1 port; When P3 is used as the second function port, the internal bus must be output as one, and the output port of the NAND gate is consistent with the second output function. Through the analysis of the bit structure function of the P3 port and the knowledge of the hardware circuit design, we can design the P3 hardware circuit structure.

The main pin descriptions of the P3 port hardware circuit diagram are shown in Table 4. The pins that have been described above are not repeated here.

Table 4 Signal description line of port P3

B0H	P3 port device address
second_function_in	Second input function
second_function_out	Second output function
sfr_data_in_p3	P3 port inputs data data lines to the data bus
sfr_cs3	Chip select signal for input data from p3 port to data bus
p3	double-sided p3 port

According to the hardware structure of the P3 port, the hardware circuit design language Verilog is used to design the P3 port hardware circuit description. The following is the main description of P3.0

```
assign sfr_cs3=(sfr_rd&&(sfr_addr==8'hb0))?1'b1:1'b0;
assign sfr_data_in    =    sfr_cs3?(port_pin_reg_n? p3 : p3_l): 8'bz;
inout_pad p3_port0(.I(1'b0),
                        .OEN ((rxd0_out & p3_l[0])? 1'b1 : 1'b0),
                        .PAD (p3[0]),
                        .C     ()
                          );
assign rxd0_in = p3[0];
if(sfr_wr)
      begin
           case(sfr_addr)
                 8'hb0: begin p3_l<=sfr_data_out; end
                 default: ;
```

```
            endcase
        end
```

4. Simulation implementation

In the design process, the software timing simulation is first performed. The specific steps are as follows:

1) Select the FPGA device EP4CE115F29C7 in Quartus II 13.0, then compile the project, check and eliminate syntax errors.

2) Write the running program in C language in Keil4, including the read and write operations of each I/O port of P0 to P3. After compiling, load the generated HEX file into the ROM in the design.

3) Write a test file in Quartus II, call the top-level design, and compile it all.

4) After compiling, call Modelsim in Quartus II to simulate the waveform and observe the input and output waveforms of each I/O.

By observing the results of the software simulation, the waveform output of each I/O is consistent with the expected result.

5. Conclusion

In the design, according to the specification of hardware circuit design, the I/O port expansion of DW8051 is realized, and the integrity and correctness of the design are proved by software simulation. By extending the I/O port of the DW8051, it is compatible with the Intel standard 8051 instruction set, which further simplifies the use cost of the DW8051, that is, obtains higher MCU performance, whether to increase the learning cost of the standard 51 MCU user, and promote the use of the DW8051 core. Brought great convenience.

References

[1] Synopsys Corp. DesignWare DW8051 MacroCell Databook. 2004.

[2] Yigang Zhang. The principle and application of single chip microcomputer [M]. Beijing: Higher Education Press,2010: 5-30.

[3] Xuehai Li. Standard 80C51 single-chip basic tutorial - principle articles [M]. Beijing: Beijing Aerospace University Press,2006

[4] Zhenzhong Dou, Peng Song. New developments in embedded system design and its challenges. MCU and embedded system

[5] Yuwen Xia. Verilog Digital System Design Tutorial [M]. Beijing: Beijing Aerospace University Press,2009.

[6] Samir Palnitkar. Verilog HDL Digital Design and Synthesis [M]. Beijing: Publishing House of Electronics Industry,2009.

[7] Xuehai Li. Standard 80C51 single-chip basic tutorial - principle articles [M]. Beijing: Beijing Aerospace University Press,2006

[8] Xiaolin Li, Shuguang Zhu. Principle and Interface Technology of Single Chip Microcomputer [M]. Beijing: Publishing House of Electronics Industry,2011.

[9] Shi Yan. Digital Electronic Technology Foundation [M]. Beijing: Higher Education Press,2005.

[10] Ru Zhang, Songlin Sun, Xiaogang Yu. Embedded System Technology Foundation [M]. Beijing: Beijing University of Posts and Telecommunications Press,2006.

[11] GregOsbom. Embedded Microcontroller and Processor Design [M]. Beijing: Mechanical Industry Press,2011:79—81.

[12] Wei Guo, Jizeng Wei. Soc design method and implementation [M]. Beijing: Publishing House of Electronics Industry,2011.

[13] Liang Zhang. Digital Circuit Design and Verilog HDL [M]. Beijing: People's Posts and Telecommunications Press,2000: 20-35.

[14] Song Pan. EDA Technology Practical Tutorial - Verilog HDL Edition [M]. Beijing: Science Press,2011: 6-15.

[15] Altera Corp. DE2_115_User_manual.2010.