辽宁大学学报
自然科学版
第35卷 第1期 2008年

JOURNAL OF LIAONING UNIVERSITY
Natural Sciences Edition
Vol.35 No.1 2008

中断控制器a8259的子模块—中断请求寄存器的代码与仿真

吴春瑜[1*],刘 晶[1],王 宏[2],董丽凤[1]
(1.辽宁大学 物理学院,辽宁 沈阳 110036;2.东北微电子研究所,辽宁 沈阳 110032)

摘 要:基于中断控制器a8259的子模块—中断请求寄存器的工作原理编写了verilog HDL代码,给出了用modelsim5.7c软件仿真的波形,并对波形进行了分析,结果表明功能正确.该模块可移植性强,可以用在同类功能的其他系统中.

关键词:中断请求寄存器;verilog HDL;代码;仿真.

中图分类号:TP391.9 **文献标识码**:A **文章编号**:1000-5846(2008)01-0014-03

为了解决快速的CPU与慢速的外设之间速度不匹配的问题,实现CPU与外设并行工作,提高CPU的利用率.确保CPU在运行过程中具有实时响应和处理随机事件的能力,常使用中断控制器,它能够实时地接收外部信号,配合CPU实现对外设的控制[1,5].中断技术的使用,明显提高了计算机系统中信息处理的并行度和处理器的效率,改善了计算机系统的性能[6].中断控制器a8259又称为优先级控制器,常用来对中断系统中的可屏蔽中断进行中断管理.中断请求寄存器(IRR)是中断控制器的子模块,它的功能就是接收外部中断源的中断请求.本文用verilog HDL语言编写了这个子模块的代码,进行了功能仿真,给出了仿真波形,经过分析表明功能正确.

1 中断请求寄存器IRR及其工作原理

IRR是一个8位寄存器,用于接收外部中断请求.引入中断请求的方式有边沿触发方式和电平触发方式.在边沿触发方式下,a8259将中断请求输入端出现的上升沿作为中断请求信号.中断请求输入端出现上升沿触发信号以后,可以一直保持高电平.在电平触发方式下,便把中断请求输入端出现的高电平作为中断请求信号.当某一个IR端接收中断请求信号时,则IRR的相应位将被置"1"(即对该中断请求作了锁存);若最多有8个中断请求信号同时进入IR0~IR7端,则IRR将被置为全"1"[1,2].

如果CPU的中断允许位IF为1,那么CPU执行完当前指令后,就可以响应中断.这时,CPU(对8086而言)从INTA线上往a8259线上会送两个负脉冲.第一个负脉冲到达时,使IRR的锁存功能失效.这样,在IR7~IR0线上的中断请求信号就不予接收,直到第二个负脉冲到达时,才又使IRR的锁存功能有效,同时使IRR寄存器中相应位(即刚才设置的位)清0.a8259的引脚见图1.

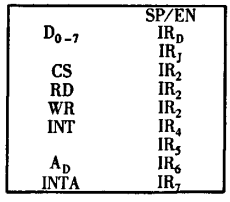

图1 a8259引脚

* 作者简介:吴春瑜(1953-),男,北京人,辽宁大学教授,从事集成电路、电力电子器件的设计与研究.
基金项目:辽宁省科技厅自然科学基金(002021);辽宁省教育厅科学基金(2004D026)
收稿日期:2006-10-12

2　代码

用 verilog HDL 语言对 IRR 进行功能描述[7,8]

```
module IRR (mrst_n, init, clk, freeze, ltim, isr_set_stb, isr_set, ir_mask, ir, irr, masked_irr);
    input mrst_n;          //system reset (active low)
    input init;            //sw initialization reset (active high)
    input clk;             //clock input (positive edge)
    input freeze;          //freeze ir input (active high)
    input ltim;            //level sensitive (icw1(3))
    input isr_set_stb;     //isr set strobe (active high)
    input[7:0] isr_set;    //set isr vector
    input[7:0] ir_mask;    //ir mask (ocw1)
    input[7:0] ir;         //interrupt request inputs (active high)
    output[7:0] irr;       //irr output for data reads
    wire[7:0] irr;
    output[7:0] masked_irr;//masked irr for data reads
    wire[7:0] masked_irr;
    reg[7:0] qsynch1;      //first ir synch register output
    reg[7:0] dsynch2;      //second ir sync register input
    reg[7:0] qsynch2;      // second ir sync register output
    reg[7:0] dedge;        //ir edge register input
    reg[7:0] qedge;        // ir edge register output
    reg[7:0] dirr;         // irr input
    reg[7:0] qirr;         // irr output
    assign irr = qirr;     //reassign intermediate signal to output
    assign masked_irr = qirr & (~ir_mask);  //and mask register and irr before sending to pr
    always @(init or qsynch1 or qsynch2 or qedge or qirr or isr_set_stb or isr_set or ltim or freeze)
    begin : comb_prc
      dsynch2 <= qsynch1;
      if (init == 1'b1) //clear edge detect reg at initiation
    dedge <= {8{1'b0}};
      else if (isr_set_stb == 1'b1)     //hold edge detect until cleared when setting
      dedge <= (qedge | (qsynch1 & (~qsynch2))) & (~isr_set);
    else
      dedge <= qedge | (qsynch1 & (~qsynch2));
      if (ltim == 1'b0) //edge sensitive
    begin
      if (isr_set_stb == 1'b1)
        dirr <= qirr & (~isr_set);
      else if (freeze == 1'b1)
        dirr <= qirr;
      else
        dirr <= qedge;
    end
      else //level sensitive
    begin
      if (isr_set_stb == 1'b1)
        dirr <= qirr & (~isr_set);
    else if (freeze == 1'b1)
      dirr <= qirr;
    else
      dirr <= qsynch2;
      end
    end
    always @(clk)
    begin : synch_reg_prc
    if (clk == 1'b1)
      begin
        qsynch1 <= ir;
        qsynch2 <= dsynch2;
      end
    end
    always @(mrst_n or clk)
    begin : reg_prc
      if (mrst_n == 1'b0)
      begin
        qedge <= {8{1'b0}};
        qirr <= {8{1'b0}};
      end
      else if (clk == 1'b1)
      begin
        qedge <= dedge;
        qirr <= dirr;
      end
    end
endmodule
```

3　仿真实现

3.1　仿真波形

用 modelsim5.7c 对 IRR 所作的仿真波形见图 2.

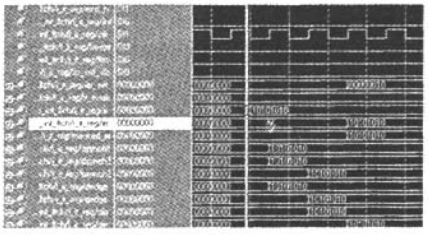

图 2　IRR 的仿真波形

3.2 波形分析

由图 2 可见 ltim = 0 使 IRR 处于边沿触发请求模式下，当 ir 有信号输入即 ir = 10101010 时，irr = 00000000；在之后 clk 的第三个上升沿时 irr = 10101010. 这是因为在第一个时钟上升沿时序部分 qsynch1 < = ir，使 qsynch1 = 10101010，同时组合逻辑 dsynch2 < = qsynch1，dedge < = qedge | (qsynch1 & (~qsynch2))使 dsynch2 = 10101010，dedge = 10101010；第二个时钟上升沿时序部分 qedge < = dedge 使 qedge = 10101010，同时组合逻辑 dirr < = qedge 使 dirr = 10101010；第三个时钟上升沿时序部分 qirr < = dirr 使 qirr = 10101010，同时组合逻辑 irr = qirr 使 irr = 10101010. 仿真结果表明功能正确. 通过以上的分析不仅可以看到各个信号的工作原理，也可以明确一个既有时序逻辑又有组合逻辑的过程中信号与时钟的关系.

4 结论

通过研究了 IRR 中断请求寄存器的工作原理，用 verilog HD 硬件描述语言做了功能描述. 结合仿真波形的分析明确了一个既有时序逻辑又有组合逻辑的过程中信号与时钟的关系，这对 IC 设计人员编写这一类进程的代码分析波形时避免错误，节约时间，加快工程进度都是有益的.

参 考 文 献 :

[1] 龚义建，严运国. 微机原理与接口技术[M]. 北京：科学出版社，2005.

[2] 戴梅萼. 微型计算机技术及应用[M]. 北京：清华大学出版社，1991.

[3] 王 诚，薛小刚，钟信潮. FPGA/CPLD 设计工具[M]. 北京：人民邮电出版社，2005.

[4] 简弘伦. 精通 verilog HDL：IC 设计核心技术实例详解[M]. 北京：电子工业出版社，2005.

[5] 王 伟. verilog HDL 程序设计与应用[M]. 北京：人民邮电出版社，2005.

[6] 王月姣. 单片机扩展 8259 中断控制器的研究[J]. 中南民族大学学报(自然科学版)，2004，2：23.

[7] IEEE standard Hardware Description Language Based on the Verilog Hardware Desription Language. IEEE computer society，IEEEstd，1995.

[8] Bhasker J. verilog HDL synthesis a Practical Primer [M]. 孙海平译. 北京：清华大学出版社，2004.

The Sub – module of Interrupt Controller a8259 – the Code and Simulation of the Interrupt Request Register IRR

WU Chun-yu[1], LIU Jing[1], WANG Hong[2], DONG Li-feng[1]

(1. *The College of Physics, Liaoning University, Shenyang* 110036, *China*;
2. *NorthEast Microelectronics Institute, Shenyang* 110032, *China*)

Abstract: In this paper, code of verilog HDL is complied based on the principle of operation of which is the sub – module of Interrupt Controller a8259, the simulation waveform of the software of modelsim5.7c is given, and the waveform is analyzed. The result indicates the exactness of the function. The module can be transplanted easily so it can be used in some other systems with the same function.

Key words: IRR; verilog HDL; code; simulation.

(责任编辑 郑绥乾)