

Verilog红宝书_编程规范



电子发烧友

www.elecfans.com

阿东

恒创科技

积聚天下电子发烧友
服务天下电子发烧友！



电子发烧友
www.elecfans.com

电子产业链 电子工程师互动平台

简介

- 大家可以叫我阿东，我在通信行业做了6年的芯片设计，做了几款大型路由器和交换机芯片，写了6年的Verilog，对Verilog是熟悉的不能再熟悉了，对数据通信、QOS有深入研究和实现，精通数据通信各种协议，对通信网络有较深理解。精通ASIC、FPGA和Verilog架构、方案、实现设计。
- 希望我的经历能让大家掌握项目开发的编码规范和方案设计，给大家的学习和发展略进绵薄之力。大家使用过程中遇到什么问题，可以发邮件:1530384236@qq.com，QQ:1530384236。
- 后续还会推出更多FPGA相关资料，包括方案设计、视频教程、项目实战等，请大家关注我们。

› 电子论坛 › EDA设计论坛 › FPGA/CPLD/ASIC论坛 › 阿东Verilog技术专版 ›

- **淘宝店铺（本店专注于FPGA开发板开发）：**

<http://shop67541132.taobao.com>

<http://shop69029874.taobao.com>

- **开发板介绍（位于电子发烧友的阿东Verilog技术专版）：**

http://bbs.elecfans.com/jishu_348985_1_1.html



电子发烧友
www.elecfans.com

目 录

1. 往期回顾
2. 为什么要编程规范
3. 工程的组织形式
4. 设计风格
5. 可综合性
6. 编程规范实例
7. 本期总结
8. 附录-阿东开发板简介



1、往期回顾

第一期和第二期主要讲解了**Verilog**的历史、重要性、基础语法。

硬件描述语言的重点：

- 1、**Verilog**语言是商业公司绝对主流的设计语言。
- 2、**Verilog**语言掌握基础的语法即可，项目开发也是只使用简单的语法。
- 3、基于**Verilog**进行方案设计是**FPGA**设计的重中之重。



电子发烧友

www.elecfans.com

2、为什么要编程规范？

1、当前数字电路设计越来越复杂，一个项目需要的人越来越多，当几十个设计同事完成同一个项目时候，大家需要互相检视对方代码，如果没有一个统一的编程规范，那么是不可想象的。大家的风格都不一样，如果不统一的话，后续维护、重用等会有很大的困难，即使是自己写的代码，几个月后再看也会变的很陌生，也会看不懂（您可能不相信，不过阿东和同事交流发现大家都是这样的，时间长不看就忘记了），所以编程规范的重要性显而易见。

2、养成良好的编程规范，对于个人的工作习惯、思路等都有非常大的好处。可以让新人尽快融入项目中，让大家更容易看懂您的代码。



3、工程的组织形式

1、当前项目越来越复杂，项目的文件需要统一的存放在一个统一的文件夹下。根据各自功能不同，分门别类的存放。以项目XX为例，其文件存储方式如下所示。

```
XX├──doc├──ip├──rtl├──sim└──flow
```

● doc

- 存放项目相关的文档，包括该项目用到的datasheet，芯片规格书(specification、设计方案)等等。

● ip:

- 存放商业上购买的成熟IP以及项目中一些成熟的可复用的代码。



工程的组织形式（续）

- **rtl:**

- 存放项目的rtl代码。这是项目的核心，文件名与**module**名称应当一致，建议按照模块的层次/子系统分别新建目录分开存放。

- **sim:**

- 存放项目的仿真代码。一般**sim**下面还是需要根据子系统继续划分子目录。

- **flow:**

- 存放后端的一些动作，包括**DC**综合、**STA**时序分析、**FV**形式验证等后端动作。



4、设计风格

- 1) 文件声明
- 2) 输入输出定义
- 3) **Parameter**定义
- 4) **Wire/Reg**定义
- 5) 信号命名
- 6) **Always**块描述方式
- 7) **Assign**块描述方式
- 8) 空格和**TAB**
- 9) 注释
- 10) 例化
- 11) 状态机



文件声明

每一个Verilog文件的开头，都必须有一段声明的文字。包括文件的版权，作者，创建日期以及内容介绍等，如下表所示。

```
// *****  
// Copyright (C)    xx Coporation  
  
// File name:      xx.v  
// Author:         xxxxxxxxxxxx  
// Date:           2012-01-01  
// Version:        1.0  
// Abstract:       xxxxxxxx  
  
//*****
```

如果对该文件进行了修改，请在开头声明中添加以下语句，如下表所示。

```
// Version:        1.1  
// Date:           2012-01-01  
// Abstract:       XXX  
// Author:         xxxxxxxxxxxx
```

- 本文规定：**
- ① 不建议使用include包含文件；
 - ② 一个.V只包括一个module；



输入输出定义

输入输出有Verilog 95和2001两种格式，本文采用Verilog 2001语法格式。下面是一个例子，包括module名字、输入输出、信号名字、输出类型、注释。

```
module divider7_fsm (
//input
input      sys_clk      ,      // system clock;
input      sys_rst_n    ,      // system reset, low is active;

//output
output reg  clk_divide_7      // output divide 7 clk
);
```

本文规定：

- ① 一行只定义一个信号；
- ② 信号需要对齐；
- ③ 同一组（一个种类）的信号放在一起；

Parameter 定义

一个module中的Parameter声明，不建议随处乱放。

本文规定：

- ① 将Parameter定义放在紧跟着module的输入输出定义之后；
- ② Parameter等常量命名全部使用大写。
- ③ 一行只定义一个信号；

```
// Parameter ↵  
parameter S0 = 7' b0000000;↵  
parameter S1 = 7' b0000001;↵  
parameter S2 = 7' b0000010;↵  
parameter S3 = 7' b0000100;↵  
parameter S4 = 7' b0001000;↵  
parameter S5 = 7' b0010000;↵  
parameter S6 = 7' b0100000;↵
```



电子发烧友

www.elecfans.com

WIRE/REG定义

一个module中的变量声明需要集中放在一起，不建议随处乱放。

本文规定：

- ① 将reg与wire的定义放在紧跟着parameter之后；
- ② 建议具有相同功能的信号集中放在一起；
- ③ 信号需要对齐，reg和位宽需要空2格，位宽和信号名字至少空四格；
- ④ 位宽使用降序描述，[6:0]；
- ⑤ 时钟使用前缀clk，复位使用后缀rst；
- ⑥ 不能使用Verilog保留字作为信号名字；
- ⑦ 一行只定义一个信号；

```
//reg define ↵  
reg [6:0]    curr_st    ;    // FSM current state↵  
reg [6:0]    next_st    ;    // FSM next    state↵  
reg          clk_divide_7 ;    // generated clock, divide by 7↵  
  
//wire define ↵  
wire [6:0]   aa        ;    ↵
```

信号命名

信号命名每个人都有不同的喜好，尤其是新员工，命名是五花八门，但是作为一个项目必须统一风格。

本文规定：

- ① 信号命名需要体现其意义，比如fifo_wr代表FIFO读写使能；
- ② 可以使用“_”隔开信号，比如sys_clk；
- ③ 内部信号不要使用大写，也不要使用大小写混合，建议全部使用小写；
- ④ 顶层PAD信号名字建议全部使用大写，比如CPU_WR代表CPU写使能管脚；
- ⑤ 模块名字使用小写；
- ⑥ 低电平有效的信号，使用_n作为信号后缀；
- ⑦ 异步信号，使用_a作为信号后缀；
- ⑧ 纯延迟打拍信号使用_dly作为后缀；



Always块描述方式

一个module中的变量声明需要集中放在一起，不建议随处乱放。

本文规定：

- ① if需要空四格；
- ② 一个always需要配一个begin和end；
- ③ always前面需要有注释；
- ④ 信号和逻辑运算符有空格；
- ⑤ begin建议和always放在同一行；
- ⑥ 一个always和下一个always空一行即可，不要空多行；
- ⑦ 时钟复位触发描述使用posedge sys_clk和negedge sys_rst_n
- ⑧ 一个always块只包含一个时钟和复位；
- ⑨ 时序逻辑使用非阻塞赋值；
- ⑩ 信号逻辑处理的位宽需要一致，即使+1，1的位宽定义也要和信号保持一致；



Always块描述方式（续）

一个例子：

```
// gen a 1S counter , 1S = 50000000 * 20ns ↵
always @(posedge sys_clk or negedge sys_rst_n) begin ↵
    if (sys_rst_n ==1'b0) ↵
        clk_cnt <= 26'b0;↵
    else if ( clk_cnt == 26'd50000000 )↵
        clk_cnt <= 26'b0;↵
    else↵
        clk_cnt <= clk_cnt + 26'b1;↵
end↵
    ↵
always @(posedge sys_clk or negedge sys_rst_n) begin ↵
    xxx↵
```



Assign块描述方式

一个module中的变量声明需要集中放在一起，不建议随处乱放。

本文规定：

- ① assign模式的逻辑不能太复杂，否则易读性不好；
- ② assign前面需要有注释；
- ③ 组合逻辑使用阻塞赋值；

```
// wfq_sch_en enable is active 1 second ↵  
assign wfq_sch_en = ( clk_cnt == 26'b0 ) ? 1'b1:1'b0 ;↵
```



空格和TAB

由于不同的解释器对于**TAB**翻译不一致，所以建议不要使用**TAB**，全部使用空格。

本文规定：

① 不使用**TAB**，全部使用空格



电子发烧友

www.elecfans.com

例化

例化名字如果随意命名也会导致代码可读性变差。

本文规定：

- ① module例化使用U_XX_0、U_XX_1标示；
- ② 例化使用大写；

```
// instance RR_CORE
RR_CORE #( 4,2 ) U_RR_CORE (
    .rr_en      ( rr_en      )      ,
    .port_vld   ( key        )      ,
    .last_sel_port ( last_sel_port )      ,

    .port_win   ( port_win   )
);
```



状态机

状态机使用不容易想到具体的电路结构，且容易进入异常状态，另外还可能导致覆盖率不好覆盖。

本文规定：

- ① 状态机使用三段式描述；
- ② 尽量避免使用状态机，减少电路复杂性；



电子发烧友

www.elecfans.com



5、可综合性

RTL代码（仿真/模型除外）编写一定要使用可以综合的语法，否则会造成不可实现和后端实现困难等问题。

本文规定：

- ① 不使用不可综合语句；
- ② 不使用repeat等循环语句；
- ③ 避免使用异步反馈环电路；
- ④ RTL级别代码里面不使用initial语句，仿真代码除外；
- ⑤ 避免产生Latch锁存器；

锁存器会导致电路的可测性变小
影响DFT和STA分析。

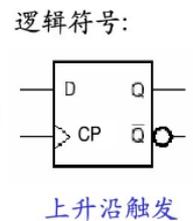
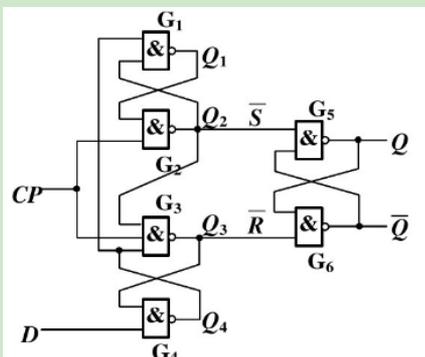
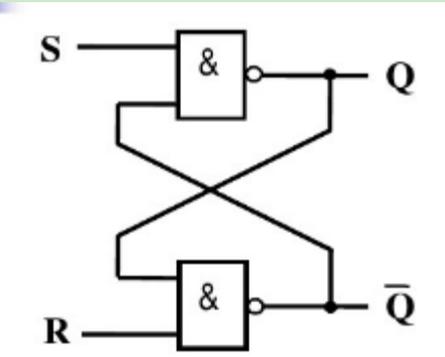
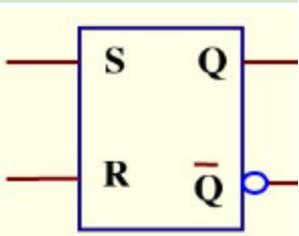
产生 Latch 例子:↵

① 条件没有补全↵

```
always @(*) begin↵
  if (condition) ↵
      latch_i = aa;↵
end↵
```

② 组合逻辑赋值给自己↵

```
always @(*) begin↵
  if (condition) |↵
      aa = aa;↵
end↵
```





6、编程规范实例

```
/* *****  
** Copyright :      Dongdong  Studio  
  
** File name :      LED  
** Auther   :      dongdong  
** DATA    :      20131110  
** Version  :      1.0  
** Abstract  :      Pipeline LED  
  
**-----  
** Version:      1.1  
** Modified by:  dongdong  
** DATA     :   20131111  
** Abstract  :   Add note  
**-----  
*****  
module LED (  
    //input  
input      sys_clk      ,    //system clock;  
input      sys_rst_n    ,    //system reset, low is active;  
  
    //output  
output reg [7:0] LED  
);  
  
//Parameter define  
parameter WIDTH0 = 3      ;  
parameter WIDTH1 = 25    ;  
  
//Reg define  
reg [WIDTH0-1:0] counter  ;  
reg [WIDTH1-1:0] count    ;
```

```
/* *****  
/**                               Main Program  
/**  
*****  
// count for add counter, 0.125S/20ns = 6,250,000, need 23 bit cnt  
always @(posedge sys_clk or negedge sys_rst_n) begin  
    if ( sys_rst_n ==1'b0 )  
        count <= 23'b0;  
    else if ( count == 23'd6250000 )  
        count <= 23'b0;  
    else  
        count <= count + 23'b1;  
end  
  
// counter for pipeline LED display  
always @(posedge sys_clk or negedge sys_rst_n) begin  
    if ( sys_rst_n ==1'b0 )  
        counter <= 3'b0;  
    else if ( count == 23'd6250000 )  
        counter <= counter + 3'b1;  
    else ;  
end  
  
// ctrl LED pipeline display when counter is equal 10 or 20 ....  
always @(posedge sys_clk or negedge sys_rst_n) begin  
    if ( sys_rst_n ==1'b0 )  
        LED <= 8'b0;  
    else begin  
        case ( counter )  
            3'd0      : LED <= 8'b10000000 ;  
            3'd1      : LED <= 8'b01000000 ;  
            3'd2      : LED <= 8'b00100000 ;  
            3'd3      : LED <= 8'b00010000 ;  
            3'd4      : LED <= 8'b00001000 ;  
            3'd5      : LED <= 8'b00000100 ;  
            3'd6      : LED <= 8'b00000010 ;  
            3'd7      : LED <= 8'b00000001 ;  
            default   : LED <= 8'b00000000 ;  
        endcase  
    end  
end  
  
endmodule  
//end of RTL code
```

- ✓ 文件头。
- ✓ 端口声明。
- ✓ 注释编写。
- ✓ 空格

7、本期总结

- 1、编程规范是大型项目必须的。
- 2、编程规范是一个人的修养。
- 3、编程规范是无数个芯片流片失败总结出来的。



8、暴风系列开发板简介

附录1

中端2C5 FPGA开发板套餐(包括开发板+USB Blaster+1602 +电源线+亚克力):
适合没有任何基础、有一定基础,可以用来做学习和一般项目、NIOS。



● 淘宝网址:

<http://item.taobao.com/item.htm?spm=a230r.1.14.37.3IZ4fw&id=18309760936>



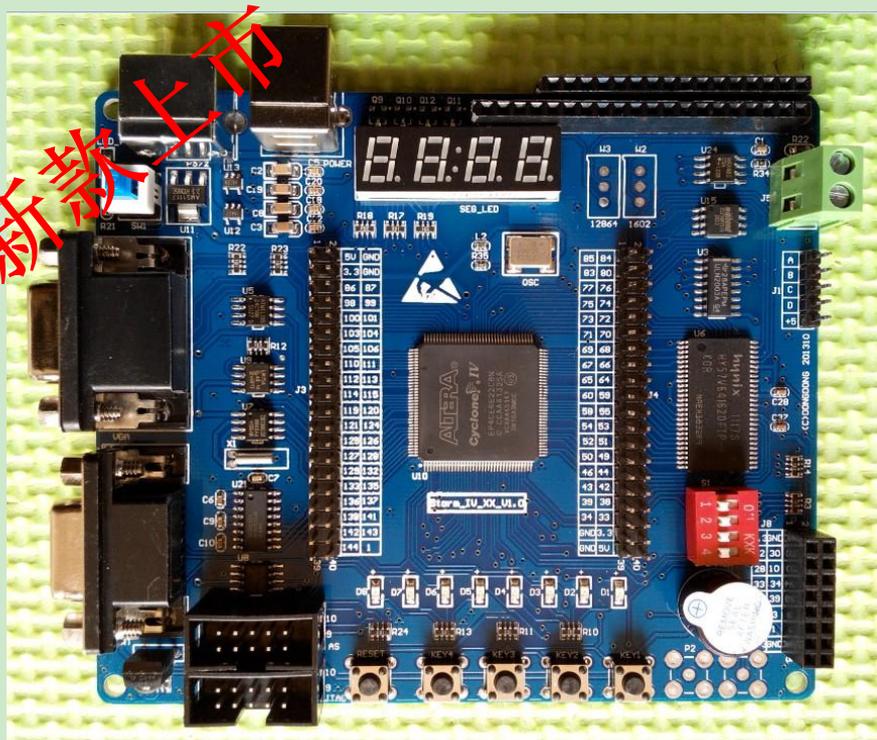
电子发烧友
www.elecfans.com

附录2

暴风系列开发板简介

中端EP4CE6 FPGA开发板套餐(包括开发板+USB Blaster+电源线+亚克力):

适合没有任何基础、有一定基础,可以用来做学习和一般项目、NIOS。



● 淘宝网址:

<http://item.taobao.com/item.htm?spm=a230r.1.14.131.v3ax3E&id=35911884243>



电子发烧友

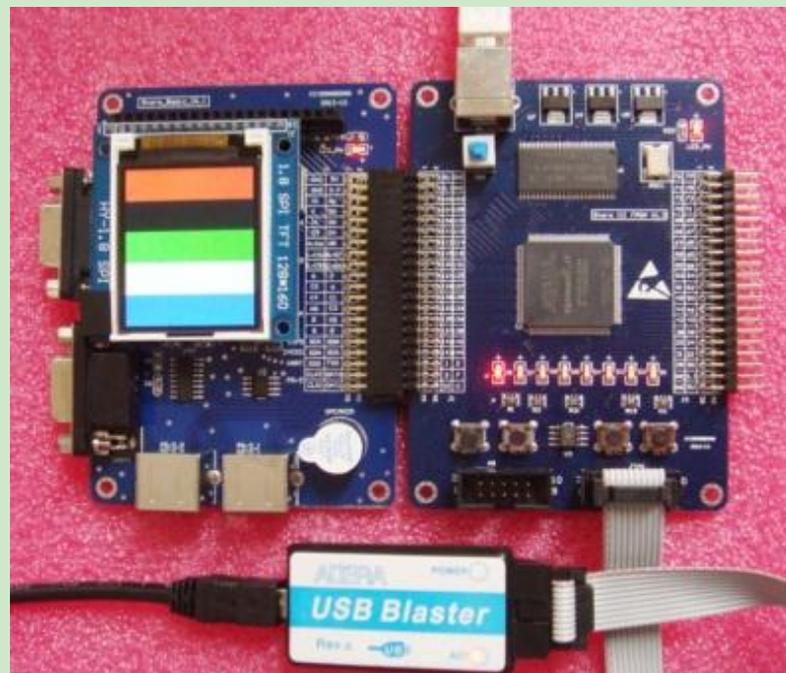
www.elecfans.com

附录3

暴风系列开发板简介

高端3C10 FPGA开发板 (包括核心板+扩展板+USB Blaster+亚克力壳):

适合没有任何基础、有一定基础、基础较好,可以用来做学习和高级项目、NIOS。



● 淘宝网址:

<http://item.taobao.com/item.htm?spm=a1z10.1.w4004-1006554551.5.Z5bNtc&id=17385063452>



电子发烧友

www.elecfans.com

附录4

暴风系列开发板简介

高端4CE15 FPGA开发板420元套餐(包括核心板+扩展板+USB Blaster+亚克力壳)：

适合有一定基础、基础较好，想一步到位（学习和复杂项目都可以）的同学，可以用来做学习和高级项目、NIOS。

板载SDRAM/SRAM/FLASH



● 淘宝网址：

<http://item.taobao.com/item.htm?spm=a1z10.1.w4004-2613814661.5.IHzdr5&id=25969336978>



电子发烧友

www.elecfans.com

播放结束，谢谢观看！



电子发烧友

www.elecfans.com

积聚天下电子发烧友
服务天下电子发烧友！



电子发烧友

www.elecfans.com

电子产业链 电子工程师互动平台