

Verilog编码规范

1. 范围

本规范规定了采用verilog设计时的代码书写规范。

2. 规范性引用文件

Verilog Coding Style <http://www.ee.ncu.edu.tw/~jfli/vlsidi/lecture/VerilogCoding-2009.pdf>

3. 术语与定义

4. 规范内容

4.1 设计风格

1. 【必须】目录名小写，单词中间用下划线“_”连接，不能用空格
2. 【必须】模块名和文件名保持一致
3. 【必须】每个文件只包含一个模块
4. 【必须】每个文件有一个文件头
5. 【必须】每个模块加timescale
6. 【必须】模块名小写，单词中间用下划线“_”连接
7. 【必须】模块例化名采用“名称_模块名_inst”，如果有多次例化采用“名称_模块名_inst_n”，n从0开始递增
8. 【必须】采用大写字母定义参数，单词中间用下划线“_”连接
9. 【必须】使用小写字母定义wire、reg和input/output/inout，单词中间用下划线“_”连接
10. 【必须】不要书写空的模块，即一个模块至少要有有一个输入和一个输出
11. 【必须】时钟信号应后缀“_clk”，复位信号应后缀“_rst”
12. 【必须】低电平有效的信号，信号名后缀"_n"
13. 【必须】时钟事件的表达式要用“negedge <clk_name>”或“posedge <clk_name>”的形式

14. 【必须】异步复位，高电平有效用“if(<asynch_reset>==1'b1)”，低电平有效用“if(<asynch_reset>==1'b0)”
15. 【必须】代码中不能使用verilog保留字
16. 【必须】程序块采用缩进风格编写， 每级缩进为 4 个空格，不使用TAB

4.2 设计可靠性

1. 【必须】采用同步设计，避免使用异步逻辑（全局异步复位除外）
2. 【必须】不要在模块内部生成时钟信号，使用PLL产生时钟信号
3. 【必须】采用单一的全局同步复位电路或者单一的全局异步复位电路
4. 【必须】不要将时钟信号作为数据信号输入
5. 【必须】组合逻辑语句块敏感表中的敏感变量必须和该块中使用的一致，不能多也不能少
6. 【必须】时序always块的敏感时间列表中必须都是沿触发事件，不允许出现电平触发事件
7. 【必须】时序逻辑的always block中有且只有一个时钟信号，并且在同一个沿动作
8. 【必须】组合逻辑语句块中统一使用阻塞赋值
9. 【必须】时序逻辑语句块中统一使用非阻塞赋值
10. 【必须】非阻塞型赋值不加单位延时
11. 【必须】避免产生latch
12. 【必须】避免组合反馈
13. 【必须】不允许直接使用魔鬼数字，代码难以理解；如果一个有含义的数字多处使用，一旦需要修改这个数值，代价惨重
14. 【必须】输出信号必须被寄存（只对顶层模块）
15. 【必须】在表达式内使用括号表示运算的优先级
16. 【必须】端口申明、比较、赋值等操作时，数据位宽要匹配
17. 【必须】禁止使用casex，case语句item必须使用常数
18. 【必须】条件表达式必须是1bit value
19. 【必须】if语句嵌套不能超过3层
20. 【必须】如果没有优先级编码需求，建议采用case语句代替if语句(The multiplexer is a faster circuit. If the priority - encoding structure is not required, we recommend using the case statement)

21. 【必须】不可以对任何信号进行初始化赋值，应采用复位的方式进行初始化
22. 【必须】不能含有inital结构，应采用复位的方式进行初始化
23. 【必须】不使用用户自定义单元UDP
24. 【必须】不使用disable语句
25. 【必须】不使用任务task
26. 【必须】不使用===,!===等不可综合的运算符
27. 【必须】不使用forever、repeat、while等循环语句
28. 【必须】不使用integer类型寄存器
29. 【必须】不使用real、time、realtime类型
30. 【必须】不要使用wait语句
31. 【必须】不要使用fork-join语句
32. 【必须】不要使用specify模块

4.3 接口设计规范

1. 【必须】AXI4-Stream接口，master不允许等待TREADY再给TVALID，一旦TVALID拉高必须保持到握手成功；slave允许等待TVALID有效后拉高TREADY；slave拉高TREADY后，在TVALID拉高前运行拉低TREADY

5. 说明

5.1 避免组合反馈

组合反馈会导致输出不可预期

```
1 input a;
2 reg b;
3
4 always @* begin
5     b = b ^ a;
6 end
```

或者

```
1 input a;
2 wire b;
3 assign b = b ^ a;
```

5.2 防止latch产生的方法

- 使用完全的条件判断语句，if…else…，否则会产生锁存器
- 检查设计中是否含有combinational feedback loops，发现及时修改
- 为每个input 设计一个output，在case语句中设置default语句
- State machine中，尽量用完全条件语句（full case），如果不是，应加上default语句

修订记录

日期	版本号	修改原因	姓名
2023/4/3	0.1	初版	刘磊
2023/4/6	0.2	1，增加目录名称风格	刘磊
2023/4/6	0.3	1，增加了AXIS接口设计规范	刘磊