|  |  |  |  |
| --- | --- | --- | --- |
| UNLOGOh  项目名称：Verilog编码规范  文件编号：ZHJD-YF-046-0  Verilog语言编码规范  浙江众合机电股份有限公司  技术中心 | | | |
| 编制 | 祁香兵 | 保密级别 | 普通（） 机密（√） 绝密（） |
| 审核 | 户贯涛 | 保存期限 | 短期（） 长期（√） 永久（） |
| 批准 | **凌祝军** | 发布日期 | 2014-01-09 |
| 本文件版权属于本公司所有，未经许可，不得复制引用 | | | |

文档修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 修改内容 | 修改人 |
| V1.0.0 | 2013-07-15 | 初稿 | 祁香兵 |
| V1.0.1 | 2014-01-09 | 根据反馈意见修改：   1. 1.5章节的术语； 2. 2.2.1章节的命名规范； 3. 2.3.2章节的文件头注释规范； 4. 6.2章节的vhdl保留字。 | 祁香兵 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 概述 1](#_Toc377043460)

[1.1 目的 1](#_Toc377043461)

[1.2 范围 1](#_Toc377043462)

[1.3 阅读对象 1](#_Toc377043463)

[1.4 格式说明 1](#_Toc377043464)

[1.5 术语与缩写解释 2](#_Toc377043465)

[2 代码风格 2](#_Toc377043466)

[2.1 程序版式 2](#_Toc377043467)

[2.1.1 通用 2](#_Toc377043468)

[2.1.2 表达式 3](#_Toc377043469)

[2.1.3 模块 3](#_Toc377043470)

[2.2 命名约定 4](#_Toc377043471)

[2.2.1 文件命名 4](#_Toc377043472)

[2.2.2 命名规范 5](#_Toc377043473)

[2.3 注释风格 5](#_Toc377043474)

[2.3.1 通用 5](#_Toc377043475)

[2.3.2 文件头注释 6](#_Toc377043476)

[2.3.3 功能块注释 7](#_Toc377043477)

[2.3.4 关键代码注释 7](#_Toc377043478)

[3 语言规范 7](#_Toc377043479)

[4 度量标准 8](#_Toc377043480)

[4.1 嵌套 8](#_Toc377043481)

[5 规则列表 8](#_Toc377043482)

[6 附件 11](#_Toc377043483)

[6.1 HDL编译器不支持的Verilog结构 11](#_Toc377043484)

[6.1.1 不支持的定义和说明 11](#_Toc377043485)

[6.1.2 不支持的语句 11](#_Toc377043486)

[6.1.3 不支持的操作符 12](#_Toc377043487)

[6.1.4 不支持的门级结构 12](#_Toc377043488)

[6.1.5 不支持的其他结构 12](#_Toc377043489)

[6.2 Verilog保留字 12](#_Toc377043490)

[6.3 VHDL保留字 14](#_Toc377043491)

[7 参考资料 16](#_Toc377043492)

# 概述

## 目的

执行该规范的目的在于保证编码的规范性，确保代码质量，以及提高软件代码的可读性、可修改性，避免在开发和调试过程中出现不必要的麻烦。此外，还可以统一代码风格，便于研发人员之间的技术交流。

## 范围

本编程规范适用于本公司所有采用Verilog作为编程语言的安全相关性软件(SIL1/SIL2/SIL3/SIL4)项目，从以下几个方面进行了规范：

* 代码风格
* 语言规范
* 度量标准
* HDL编译器不支持的Verilog结构（附件）
* Verilog保留字（附件）

## 阅读对象

本规范的阅读对象主要为：

* 项目经理
* FPGA软件开发人员
* 代码审核人员
* 验证确认人员

## 格式说明

规则的格式如下：

【<规则类型><编号>，<要求>，<检查方法>】<规则描述>

对括号里的内容介绍如下：

* <规则类型>说明规则的分类，如“风格”、“语言”等。
* <编号> 每一个规则都会有一个唯一的编号。
* <分类> 格式为：强制/推荐，强制性规则是要求公司程序员都需要认真遵守的编码约定；推荐为一些重要性不高的，或很难在所有项目中适用的规则，程序员可选择性采用。
* <检查方法> 格式为：自动/评审，“自动”是通过静态检查工具能自动检查；“评审”是通过代码评审来检查是否符合规范。
* <规则描述> 描述的是规则的具体内容。

## 术语与缩写解释

表 1术语与缩写

|  |  |  |
| --- | --- | --- |
| **缩写、术语** | **英文** | **中文** |
| RTL | Register Transfer Level | 寄存器传输级 |
| PLL | Phase Locked Loop | 锁相环 |
| UDP | User Defined Primitive | 用户自定义原语 |
| HDL | Hardware Description Language | 硬件描述语言 |

# 代码风格

## 程序版式

### 通用

【风格1-1-1，推荐，评审】编写代码格式要整齐，代码缩排的格式要一致（即相同类型的代码语句要对准）。

【风格1-1-2，推荐，自动】用缩进体现代码结构，只使用空格，不使用Tab。推荐缩进的空格数为4，但在同一文件应该一致。

### 表达式

【风格1-2-1，推荐，评审】每行只写一条语句。

【风格1-2-2，推荐，评审】端口定义应该是独立的，每行只写一个端口声明语句，端口定义方向要包含在端口定义中，无需额外添加。

【风格1-2-3，推荐，评审】将端口进行分类进行定义，先按照模块分组，各个组间空一行，便于理解和调用。

【风格1-2-4，推荐，评审】端口声明顺序应与模块声明中端口的顺序一致。

【风格1-2-5，推荐，评审】所有内部net必须声明，不能隐含。

【风格1-2-6，推荐，评审】内部net的声明要跟在模块的I/O端口声明之后。

【风格1-2-7，推荐，评审】代码流中不同结构之间用一空行隔开。比如：文件头、功能块注释与代码之间应空行；相对独立的代码块之间应空行；信号的定义与代码之间应空行。

### 模块

【风格1-3-1，推荐，评审】不能访问模块外部的net和variable。

【风格1-3-3，推荐，评审】模块的端口信号尽可能少。

【风格1-3-4，强制，评审】时钟产生电路单独构成一个模块。

【风格1-3-5，推荐，评审】不同的时钟域分配到不同的模块。

【风格1-3-6，推荐，评审】设计应划分得使模块的边界与其详细设计框图的边界相匹配。

【风格1-3-7，强制，评审】将异步逻辑与同步逻辑分开。

【风格1-3-8，强制，评审】将状态机写成单独模块。把状态机代码分成两个部分，一是组合逻辑，一是时序逻辑。

【风格1-3-9，推荐，评审】FPGA内部复位处理，使用posedge复位，如果外部是下降沿复位，先反相一下再给各个模块使用。

【风格1-3-10，强制，评审】状态机的默认状态必须直接给出。

【风格1-3-11，推荐，评审】状态机状态优先使用one-hot方式，即每个状态只有一个二进制位为1，其它位为0。

## 命名约定

### 文件命名

【风格2-1-1，推荐，评审】顶层文件的命名规范：如果有2片或2片以上的FPGA时采用<项目名称\_单板名\_FPGAn.v>，n表示第几片FPGA。只有1片FPGA时采用<项目名称\_单板名\_FPGA.v>。

举例（正例）：

|  |
| --- |
| 如BILOCK项目的VOB板FPGA顶层文件命名为BiLOCK\_VOB\_FPGA.v。 |

【风格2-1-2，推荐，评审】功能模文件块的命名要求采用功能命名，用下划线分开多个单词。

举例（反例）：

|  |
| --- |
| 如信号产生功能模块的文件命名为Sig\_Gen.v。 |

【风格2-1-3，强制，评审】命名只能由26个英文字母（a-z，A-Z）、数字（0-9）或者下划线( \_ )组成。不允许使用破折号和连续的下划线。

【风格2-1-4，推荐，评审】文件名的首字符必须是字母。

【风格2-1-5，推荐，评审】模块名和文件名保持一致。

### 命名规范

【风格2-2-1，强制，评审】各种类型的命名需参照统一的命名规则。

【风格2-2-2，推荐，评审】常量用大写字母表示。

【风格2-2-3，推荐，评审】net、variable、construct及instance以小写命名。

【风格2-2-4，推荐，评审】construct、net、variable的命名要唯一，不能用大小写来区分命名。

【风格2-2-5，强制，评审】不允许用Verilog和VHDL的关键词来命名。

【风格2-2-6，强制，评审】低电平有效的信号的后缀为\_n。

【风格2-2-7，推荐，评审】时钟信号的后缀为clk\_。寄存器变量使用r\_开头

【风格2-2-8，推荐，评审】寄存器变量使用r\_开头。

【风格2-2-9，推荐，评审】输出悬空信号的后缀为\_nc。

【风格2-2-10，推荐，评审】三态信号用后缀\_z命名, 三态信号只存在于端口，内部不允许也不能使用三态信号。

【风格2-2-11，强制，评审】信号名最长为32个字符。

## 注释风格

### 通用

【风格3-1-1，强制，评审】代码行使用单行注释“//”，不使用多行“/\*..\*/”注释（文件头注释除外）。

【风格3-1-2，推荐，评审】代码行注释跟在注释代码之后，处于同一行。

【风格3-1-3，推荐，评审】若注释过长，可以分行注释。放在下一行的注释应从前行注释左侧对齐，注意分行的注释内容要独占一行，该行不能有其它的代码。

【风格3-1-4，推荐，评审】若代码本身较长，难以在同一行加以注释，可以在代码的前一行放置注释内容。注意这行注释要独占一行。

【风格3-1-5，推荐，评审】每个端口声明必须有注释说明，和端口声明处在同一行。

【风格3-1-6，推荐，评审】其它声明如net、variable等也必须加以注释说明，和声明处在同一行。

【风格3-1-7，推荐，评审】注释不应该是对代码的简单翻译, 不能使用中文注释。

【风格3-1-8，推荐，评审】注释内容要清楚、明了，含义准确，防止注释二义性

【风格3-1-9，推荐，评审】对编译directive语句`ifdef、`ifndef、`else等使用时必须予以注释，说明其用途。

【风格3-1-10，推荐，评审】对于有嵌套结构复杂和带有case语句中的END语句要加以注释，说明模块结束。

### 文件头注释

【风格3-2-1，推荐，评审】每个RTL和Verilog文件必须有文件头说明，文件头应该包含版权和公司保密等信息。

每一个文件都要对文件内容进行注释说明，说明顺序如下：

* 文件名；
* 文件内容的简单说明；
* 历史记录，包含日期、版本、作者、和变更记录等信息。

文件头的结尾要包含以下版本信息项：

* 体现公司的版权声明，如：Copyright (C), Zhejiang United Electronic Industry Co.,Ltd.

举例（正例）：

|  |
| --- |
| /\*\*  \* \brief 文件简述  \* \copyright 1999-2013, Zhejiang United Electronic Industry Co.,Ltd.  \* \author 张三  \* \author 李四  \* \file 文件名、  \*  \*  \* 该文件的详细描述  \* 该文件的详细描述  \* 该文件的详细描述  \*  \*  \* \date 2011-02-17 张三：新建文件  \* \date 2011-03-01 李四：添加函数  \*  \*/ |

【风格3-2-2，推荐，评审】每次修改文件（包括第一次创建文件），都需要在文件头注释部分增加历史记录。

### 功能块注释

【风格3-3-1，推荐，评审】对代码的每个功能块必须加以注释说明，说明其目的及功能。

### 关键代码注释

【风格3-4-1，推荐，评审】对不常用或不容易理解的代码及其使用限制要进行注释，说明其功能、作用和其它特征信息。

# 语言规范

【语言1-1，强制，评审】代码中不能使用Verilog保留字，也不能使用VHDL保留字，具体保留字请参考附件。

【语言1-2，推荐，评审】不使用`include编译指令。

【语言1-3，推荐，评审】不使用任务。

【语言1-4，推荐，评审】不使用time、real、realtime类型。

【语言1-5，推荐，评审】不使用integer类型。

【语言1-6，推荐，评审】常量使用PARAMETER来定义。

【语言1-7，推荐，评审】if判断时，把常量放置左边。

【语言1-8，推荐，评审】禁止魔鬼数字，就所有用到常数的地方，需要使用PARAMETERS。

【语言1-9，强制，评审】总线定义，统一为[位宽-1:0]方式，不能使用[0:位宽-1]。

【语言1-10，推荐，评审】禁止魔鬼数字，就所有用到常数的地方，需要使用PARAMETERS。

# 度量标准

## 嵌套

【度量1-1，推荐，评审】IF语句嵌套层数小于4层。

# 规则列表

本章列出了所有上文提到的规则：

【风格1-1-1，推荐，评审】编写代码格式要整齐，代码缩排的格式要一致（即相同类型的代码语句要对准）。 2

【风格1-1-2，推荐，自动】用缩进体现代码结构，只使用空格，不使用Tab。推荐缩进的空格数为4，但在同一文件应该一致。 2

【风格1-2-1，推荐，评审】每行只写一条语句。 3

【风格1-2-2，推荐，评审】端口定义应该是独立的，每行只写一个端口声明语句，端口定义方向要包含在端口定义中，无需额外添加。 3

【风格1-2-3，推荐，评审】将端口进行分类进行定义，先按照模块分组，各个组间空一行，便于理解和调用。 3

【风格1-2-4，推荐，评审】端口声明顺序应与模块声明中端口的顺序一致。 3

【风格1-2-5，推荐，评审】所有内部net必须声明，不能隐含。 3

【风格1-2-6，推荐，评审】内部net的声明要跟在模块的I/O端口声明之后。 3

【风格1-2-7，推荐，评审】代码流中不同结构之间用一空行隔开。比如：文件头、功能块注释与代码之间应空行；相对独立的代码块之间应空行；信号的定义与代码之间应空行。 3

【风格1-3-1，推荐，评审】不能访问模块外部的net和variable。 3

【风格1-3-3，推荐，评审】模块的端口信号尽可能少。 3

【风格1-3-4，强制，评审】时钟产生电路单独构成一个模块。 3

【风格1-3-5，推荐，评审】不同的时钟域分配到不同的模块。 3

【风格1-3-6，推荐，评审】设计应划分得使模块的边界与其详细设计框图的边界相匹配。 3

【风格1-3-7，强制，评审】将异步逻辑与同步逻辑分开。 3

【风格1-3-8，强制，评审】将状态机写成单独模块。把状态机代码分成两个部分，一是组合逻辑，一是时序逻辑。 3

【风格1-3-9，推荐，评审】FPGA内部复位处理，使用posedge复位，如果外部是下降沿复位，先反相一下再给各个模块使用。 4

【风格1-3-10，强制，评审】状态机的默认状态必须直接给出。 4

【风格1-3-11，推荐，评审】状态机状态优先使用one-hot方式，即每个状态只有一个二进制位为1，其它位为0。 4

【风格2-1-1，推荐，评审】顶层文件的命名规范：如果有2片或2片以上的FPGA时采用<项目名称\_单板名\_FPGAn.v>，n表示第几片FPGA。只有1片FPGA时采用<项目名称\_单板名\_FPGA.v>。 4

【风格2-1-2，推荐，评审】功能模文件块的命名要求采用功能命名，用下划线分开多个单词。 4

【风格2-1-3，强制，评审】命名只能由26个英文字母（a-z，A-Z）、数字（0-9）或者下划线( \_ )组成。不允许使用破折号和连续的下划线。 4

【风格2-1-4，推荐，评审】文件名的首字符必须是字母。 4

【风格2-1-5，推荐，评审】模块名和文件名保持一致。 4

【风格2-2-1，强制，评审】各种类型的命名需参照统一的命名规则。 5

【风格2-2-2，推荐，评审】常量用大写字母表示。 5

【风格2-2-3，推荐，评审】net、variable、construct及instance以小写命名。 5

【风格2-2-4，推荐，评审】construct、net、variable的命名要唯一，不能用大小写来区分命名。 5

【风格2-2-5，强制，评审】不允许用Verilog和VHDL的关键词来命名。 5

【风格2-2-6，强制，评审】低电平有效的信号的后缀为\_n。 5

【风格2-2-7，推荐，评审】时钟信号的后缀为clk\_。寄存器变量使用r\_开头 5

【风格2-2-8，推荐，评审】寄存器变量使用r\_开头。 5

【风格2-2-9，推荐，评审】输出悬空信号的后缀为\_nc。 5

【风格2-2-10，推荐，评审】三态信号用后缀\_z命名, 三态信号只存在于端口，内部不允许也不能使用三态信号。 5

【风格2-2-11，强制，评审】信号名最长为32个字符。 5

【风格3-1-1，强制，评审】代码行使用单行注释“//”，不使用多行“/\*..\*/”注释（文件头注释除外）。 5

【风格3-1-2，推荐，评审】代码行注释跟在注释代码之后，处于同一行。 5

【风格3-1-3，推荐，评审】若注释过长，可以分行注释。放在下一行的注释应从前行注释左侧对齐，注意分行的注释内容要独占一行，该行不能有其它的代码。 6

【风格3-1-4，推荐，评审】若代码本身较长，难以在同一行加以注释，可以在代码的前一行放置注释内容。注意这行注释要独占一行。 6

【风格3-1-5，推荐，评审】每个端口声明必须有注释说明，和端口声明处在同一行。 6

【风格3-1-6，推荐，评审】其它声明如net、variable等也必须加以注释说明，和声明处在同一行。 6

【风格3-1-7，推荐，评审】注释不应该是对代码的简单翻译, 不能使用中文注释。 6

【风格3-1-8，推荐，评审】注释内容要清楚、明了，含义准确，防止注释二义性 6

【风格3-1-9，推荐，评审】对编译directive语句`ifdef、`ifndef、`else等使用时必须予以注释，说明其用途。 6

【风格3-1-10，推荐，评审】对于有嵌套结构复杂和带有case语句中的END语句要加以注释，说明模块结束。 6

【风格3-2-1，推荐，评审】每个RTL和Verilog文件必须有文件头说明，文件头应该包含版权和公司保密等信息。 6

【风格3-2-2，推荐，评审】每次修改文件（包括第一次创建文件），都需要在文件头注释部分增加历史记录。 7

【风格3-3-1，推荐，评审】对代码的每个功能块必须加以注释说明，说明其目的及功能。 7

【风格3-4-1，推荐，评审】对不常用或不容易理解的代码及其使用限制要进行注释，说明其功能、作用和其它特征信息。 7

【语言1-1，强制，评审】代码中不能使用Verilog保留字，也不能使用VHDL保留字，具体保留字请参考附件。 7

【语言1-2，推荐，评审】不使用`include编译指令。 7

【语言1-3，推荐，评审】不使用任务。 8

【语言1-4，推荐，评审】不使用time、real、realtime类型。 8

【语言1-5，推荐，评审】不使用integer类型。 8

【语言1-6，推荐，评审】常量使用PARAMETER来定义。 8

【语言1-7，推荐，评审】if判断时，把常量放置左边。 8

【语言1-8，推荐，评审】禁止魔鬼数字，就所有用到常数的地方，需要使用PARAMETERS。 8

【语言1-9，强制，评审】总线定义，统一为[位宽-1:0]方式，不能使用[0:位宽-1]。 8

【语言1-10，推荐，评审】禁止魔鬼数字，就所有用到常数的地方，需要使用PARAMETERS。 8

【度量1-1，推荐，评审】IF语句嵌套层数小于4层。 8

# 附件

## HDL编译器不支持的Verilog结构

这里列出HDL编译器不支持的Verilog特殊结构：

### 不支持的定义和说明

1. primitive定义；
2. time说明；
3. event说明；
4. triand,trior,tri1,tri0和trireg网络类型。

### 不支持的语句

1. defparem语句；
2. initial语句；
3. forever语句；
4. repeat语句；
5. while语句；
6. delay控制语句；
7. event控制语句；
8. wait语句；
9. fork语句；
10. deassign语句；
11. force语句；
12. release语句；

### 不支持的操作符

(1) “＝＝＝”与“！＝＝”操作符；

(2) 变量的除法与取模操作符。

### 不支持的门级结构

不支持的文件结构有：

nmos，pmos，cmos，rpmos，rcmos，pullup，pulldown，tranif0，tranif1，rtran，rtranif0，rtranif1。

### 不支持的其他结构

(1) 模块中的分层命名；

(2) `ifdef，`ndif和`else编译指示

## Verilog保留字

表 2Verilog保留字列表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| always | | and | | assign | automatic | |
| begin | | buf | | bufif0 | bufif1 | |
| case | | casex | | casez | cell | |
| cmos | | config | |  |  | |
| deassign  disable | | default | | defparam | design | |
| endconfig | | endfunction | | endgenerate | endmodule | |
| endprimitive  event | | endspecify | | endtable | endtask | |
| for | | force | | forever | fork | |
| function | |  | |  |  | |
| generate | | genvar | |  |  | |
| highz0 | | highz1 | |  |  | |
| if | | ifnone | | incdir | include | |
| initial | | inout | | input | instance | |
| integer | |  | |  |  | |
| join | |  | |  |  | |
| large | | liblist | | library | localparam | |
| macromdule | | medium | | module |  | |
| nand | | negedge | | nmos | nor | |
| noshowcancelled | not | | notif0 | | |  | |
| notif1 |  | |  | | |  | |
| or | output | |  | | |  | |
| parameter | pmos | | posedge | | | primitive | |
| pull0 | pull1 | | pulldown | | | pullup | |
| pulsestyle\_onevent | pusestyle\_ondetect | |  | | |  | |
| rcmos | real | | realtime | | | reg | |
| release | repeat | | rnmos | | | rpmos | |
| rtran | rtranif0 | | rtranif1 | | |  | |
| scalared | showcancelled | | signed | | | small | |
| specify | specparam | | strong0 | | | strong1 | |
| supply0 | supply1 | |  | | |  | |
| table | task | | time | | | tran | |
| tranif0 | tranif1 | | tri | | | tri0 | |
| tri1 | triand | | trior | | | trireg | |
| unsigned | use | |  | | |  | |
| vectored |  | |  | | |  | |
| wait | wand | | weak0 | | | weak1 | |
| while | wire | | wor | | |  | |
| xnor | xor | |  | | |  | |

## VHDL保留字

表 3VHDL保留字列表

|  |  |  |
| --- | --- | --- |
| Abs | If | Reject |
| Access | Impure | Rem |
| After | In | Report |
| Alias | Inertial | Return |
| All | Inout | Rol |
| And | Is | Ror |
| Architecture |  |  |
| Array | Label | Select |
| Assert | Library | Shared |
|  | Literal | Signal |
| Begin | Loop | Sla |
| Block |  |  |
| Body | Map | Arl |
| Buffer | Mod | Sra |
| Bus |  | Srbyte |
|  | Nand | Then |
| Case | New |  |
| Component | Next | To |
| Configuration | Nor | Transport |
| Constant | Not | Type |
|  | Null |  |
| Disconnect |  | Unaffected |
| Downto | Of | Units |
|  | On | Until |
| Else | Open | Use |
| Elsif | Or |  |
| End | Others | Variable |
| Entity | Out | Wait |
| Exit |  | When |
|  | Package | While |
| File | Port | With |
| For | Procedure |  |
| Function | Process | Xnor |
|  |  | Xor |
| Generate | Range |  |
| Generic | Record |  |
| Guarded | register |  |

# 参考资料

1. The Verilog Golden Reference Guide, DOULOS, 1996.
2. The Verilog Hardware Description Language, Fifth Edition, Thomas & Moorby’s, 2002