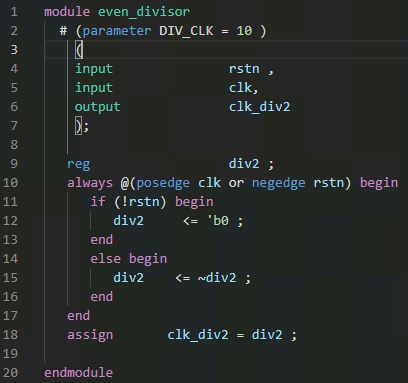
# 功能简述

# 二、处理过程



一个Verilog代码示例如上图所示，将代码解析按照上图分步获取对应信息

各状态以及对应含义如下

0:待获取”module”

获得“module”，后面无内容返回1

获得”module”，后面还有内容，返回1.5以及剩余内容

1/1.5:待获取module名

获得module名，后面无内容，返回2

获得module名，后面还有内容，返回2.5以及剩余内容

2/2.5:

获取“#” 后面无内容，返回3

获取“#” 后面还有内容，返回3.5

获得“(”，后面无内容，返回11

获得“(”，后面还有内容，返回11.5

3/3.5:

获得“(”后面无内容，返回4

获得“(”，后面还有内容，返回4.5

4/4.5:

获得“parameter”，后面无内容，返回5

获得“parameter”，后面还有内容，返回5.5

5/5.5:

获得parameter名，后面无内容，返回6

获得parameter名，后面还有内容，返回6.5

6/6.5:

获得“=”,后面无内容，返回7

获得“=”，后面还有内容，返回7.5

7/7.5:

获取参数值，后面无内容，返回8

获取参数值，后面还有内容，返回8.5

8/8.5:

获得“,”，后面无内容，返回9

获得“,”后面还有内容，返回9.5

获得“)”，后面无内容，返回10

获得“)”, 后面还有内容，返回10.5

9/9.5:

获得“parameter”，后面无内容，返回5

获得“parameter”，后面还有内容，返回5.5

否则返回原字符，返回5/5.5

10/10.5:

获得“(”，后面无内容，返回11

获得“(”，后面无内容，返回11.5

11/11.5:

增加一个标志，verilog2001 or 1995

获得input/output/inout，后面无内容，返回12 这里要设置一个全局变量保存端口方向

获得input/output/inout，后面还有内容，返回12.5

获得信号名，后面无内容，返回20

获得信号名，后面有内容，返回20.5

---------2001标准port-----

12/12.5:

获得wire/reg/logic，后面无内容，返回13

获得wire/reg/logic，后面还有内容，返回13.5

获得[，后面无内容，返回14

获得[，后面有内容，返回14.5

获得信号名，后面无内容，返回17

获得信号名，后面又内容，返回17.5

13/13.5:

获得[，后面无内容，返回14

获得[，后面有内容，返回14.5

获得信号名，后无内容，返回17

获得信号名，后又内容，返回17.5

14/14.5:

设置一个变量保存高位信息

把所有有效字符push到该变量中，直到：（在没有遇到：前可以保持14）

:后无内容，返回15

:后有内容，返回15.5

15/15.5:

设置一个变量保存低位信息

把所有字符push到该变量中，直到](在没有遇到[前，可以保持15)

]后无内容，返回16

]后有内容，返回16.5

16/16.5:

获得信号名，无内容，返回17

有内容，返回17.5

17/17.5:

获得”,”，后面无内容，返回18

获得“,”，后面有内容，返回18.5

获得“)”，后面无内容，返回19

获得“)”，后面有内容，返回19.5

18/18.5:

获得input/output/inout，后面无内容，返回12

后面有内容，返回12.5

获得wire/reg/logic，后面无内容，返回13

后面有内容，返回13.5

获得[，后面无内容，返回14

后面有内容，返回14.5

-------1995标准，port声明------

20/20.5:

获得“，”,后面无内容，返回21

有内容21.5

获得“)”，后面无内容，返回19

有内容，返回19.5

21/21.5:

获得信号名，后面无内容返回20

获得信号名，后面有内容，返回20.5

19/19.5:

获得“;”，无内容，返回22

有内容，返回22.5