1. module ···· endmodule

① 端口模式(定义)

- input
- output
- inout 双向,但并不智能.

② I/O 说明

信号类型
- wire
- reg

③ 内部信号

④ 功能定义

- 结构方式 ······ e.g. 门级建模
- 数据流方式 ······ e.g. assign 数据的赋值.
  assign (wire)
- 行为方式.

---

2. 操作数类型.

① parameter, 單个模块内定义.

def para 位置与模块内声明同级

② wire：线网型.

紧随赋值，输出紧随输入改变

③ reg：register

reg [n-1:0] m 单位/多位寄存器.

还可用于定义存储器.

---

3. 运算符.

① 拼接 {}

② 条件 (boolean)?(true):(false)

可嵌套. 数据流方式.

---

4. Verilog 行为描述方式.

① initial 语句：只执行一次

(一般用于仿真测试)

```
initial
begin
```

……

end.

② always 语句. 循环重复执行.

always @ (事件列表)

Rmk: 以上两种语句在同一个module中可写多个.
但不支持嵌套, 并行执行.

5. 时序控制方式.

① 基于延迟的控制.
常用于产生信号

② 基于敏感电平的控制.
always @ (*)
↓全部输入

③ 基于边沿敏感的控制.

always @ (posedge clock
negedge clock)

在单个 always 语句中，敏感信号列表最好只写一类.

6. 块语句.

① begin-end 语句顺序执行.

② fork-join 语句并行执行 (不被综合)

7. if-else

条件需完备，否则将综合出 latcher.

即使 else 为空语句.

Rmk: 代码风格..

reg temp.

temp = ...

↑ 只能用于 reg?

assign wire = ( )

保持 in.out 为 wire

而使 reg 中间变量.

从而方便用 assign

8. case [express]

[分支1]:
[分支2]: → 平级. 可认为是并行.

......

default: 防止 latchor.

endcase

全等比较：按位全等比较

其余还有：0、1、X、z，
不定 高阻.

casex / casez. 局部分支控制.

---

9. 过程赋值语句.

区别于 assign 的连续赋值语句.

① 阻塞.（组合）

=

② 非阻塞（时序）

阻塞/非阻塞 区分的是时间顺序.

而非数据的传递

10、FSM 的设计流程

11、Testbench 测试模块.

常见形式

```
module Test… ()
  待测试信号名.
  initial、always 产生 激励/控制信号
  待测模块实例化，注意传参.

endmodule
```

所有输入：reg. 代码定义.
所有输出：wire