·Verilog小技巧记录

1. 在写较长的常数时，为了增加可读性，可以在数字中间插入下划线“\_”，数字里的下划线会被编译器忽略。（例：8’b01010101 写成 8’b0101\_0101）
2. 编译器一般不允许向量在多位赋值时使用变量来表示位数范围（一位是可以的），也就是说不允许out[i+3:i]的写法。但是可以用运算符“+:”“-:”代替“:”，写成out[i+:4]或者out[(i+3)-:4]。“+:”表示向高位溯位，“-:”表示向低位溯位，数字表示总位数。
3. 异或与同或逻辑运算虽然可以用与或非逻辑运算表示，但是它们在Verilog里有自己的运算符“^”“~^”，“^”表示按位异或，“~^”表示按位同或（也即异或取非）。
4. 在表达常量时，无论使用何种进制书写，基数部分的数字都表示数值所对应的二进制数的位数，而不是当前进制形式下的位数。（例：十六进制数ff要写成8’hff而不是2’hff，因为两位十六进制数至少要用八位二进制数表示）
5. always语句的敏感事件表中信号默认是双边沿触发的（即对信号电平变化敏感），因此如果写成always @(clk)则表示clk的上升沿和下降沿都会触发always语句块执行。如果写成always @(\*)则表示任何输入信号变化都会触发此always语句块。
6. 时延语句“#10”的数字代表延时的时间单位数，Verilog的时间单位在’timescale语句设置。（’timescale 10ns/10ps表示一个时间单位为10ns）
7. 时延语句“#10”是不可综合的，因此任何描述电路的代码都不应该出现“#10”，要实现延时功能，只能使用时钟信号配合其他时序语句。（而且综合工具会忽略所有“#10”的同时不报错）
8. 部分无关项条件语句casex，casez也是不可综合的。
9. 尽量不要把模块中的变量名命名为“单个字母\_数字”，这样很容易和FPGA中的端口名称冲突。
10. 模块端口（即输入输出变量）的定义可以写在模块声明的括号外面，像是普通的变量声明一样，“input/output + 端口名；”，但是括号里要先写上这些变量的名称。
11. 实例化模块的时候，一般情况下，端口变量的连接写成“.原端口变量(外部接入变量)”的格式；也有省略版写法，就是只写外部接入变量名，按照模块声明时的端口变量顺序一一对应。
12. 关于wire型和reg型变量的用法规范。第一点，reg型变量不能作为assign语句的左值，wire型变量不能在always结构中赋值；第二点，实例化模块时，输入端口可以接wire或reg型变量，但是输出端口只能接wire型变量。