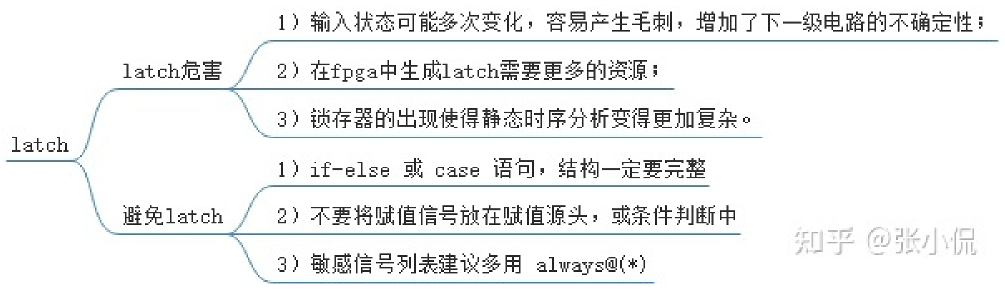
Basic Digital Circuits

1. Latch

> **From: https://zhuanlan.zhihu.com/p/456254623**
> **From: https://www.cnblogs.com/yiquwange/p/15160529.html?ivk_sa=1024320u**



latch
latch危害
1）输入状态可能多次变化，容易产生毛刺，增加了下一级电路的不确定性；
2）在fpga中生成latch需要更多的资源；
3）锁存器的出现使得静态时序分析变得更加复杂。
避免latch
1）if-else 或 case 语句，结构一定要完整
2）不要将赋值信号放在赋值源头，或条件判断中
3）敏感信号列表建议多用 always@(*)
知乎 @张小侃

Cases Produce Latches

1. 组合逻辑中，**不完整**的 `if - else` 结构会综合成 latch

```
always@(*)
if(a_i)
   out_o = c_i;
else if(b_i)
   out_o = 1'b1;
```

2. 组合逻辑电路中将值赋值给自己

```
always@(*)
if(a_i)
   out_o = c_i;
else if(b_i)
   out_o = 1'b0;
else
   out_o = out_o;
```

Solutions

1. `case` 语句： 加 `default`
2. `if` 语句： 加 `else`

3. `always` 语句: 在赋值表达式右边参与赋值的信号都必须在 `always@（敏感电平列表）` 中列
4. 赋初值

2. Divider

> From the Lab2.2 of Course CS-473

Some modules in our design need to have a slower clock. So we need **Clock Dividers.**

Here are two ways of implement a clock divider: a wrong way and a correct way

- Indeed, **clock signals are routed through special channels** to guarantee that the clock arrives at all **components roughly at the same time** (avoids clock skew)

Wrong Way

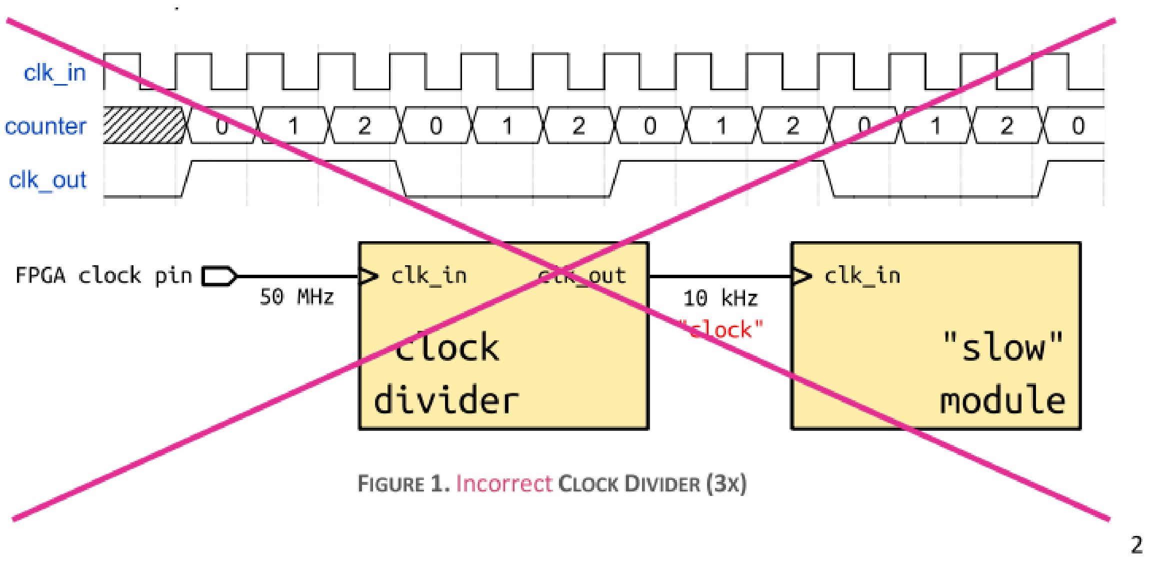


FIGURE 1. Incorrect Clock Divider (3x)

If you instead decide to “create” a clock manually by using logic within the FPGA, your clock will not be routed on the dedicated clock channels and will suffer from clock skew, therefore causing all logic driven by the custom clock to be unstable.

Correct Way

The correct way to generate slow clocks in FPGAs is to not generate a clock, but to instead generate a clock divider. And use the divider to provide `enable` signals

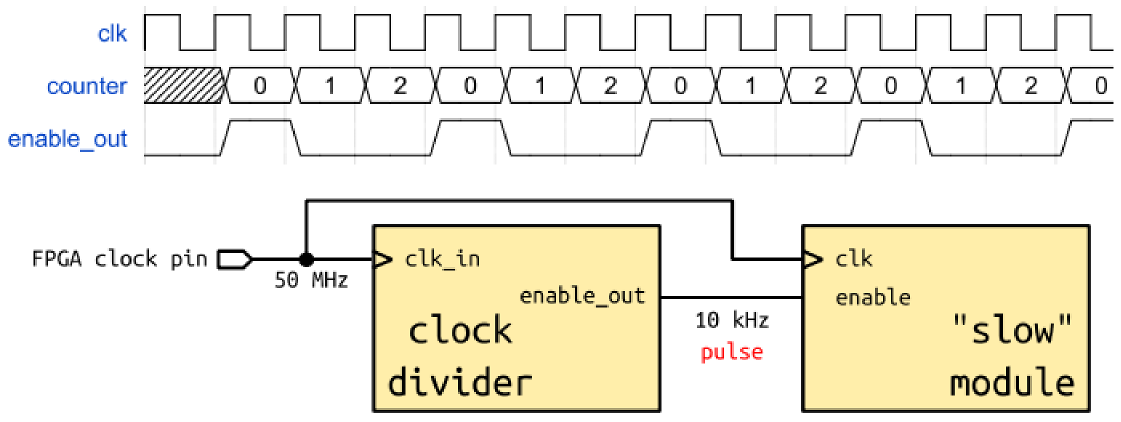


FIGURE 2. Correct CLOCK DIVIDER (3x)