

BCD 设计

本次设计的测试未采用之前所描述的方法，后文将进行介绍。

算法

整体流程的伪代码如下：

```
set(from=bin_reg, to=reg)
set(hun_reg, 0)
set(ten_reg, 0)
set(one_reg, 0)

bundle(hun_reg, ten_reg, one_reg, bin_reg, as=shift_reg)

for i in range(bit_width_of_in):
    check_bundle(bundle)
    bundle = bundle << 1

def check_bundle(bundle):
    for part in [hun_reg, ten_reg, one_reg]:
        if part > 4:
            part += 3
```

注意到，对每一位的检测实际上是检测上一次移位的结果，防止最后一次的结果被修改，如 259 -> 25 (9+3)。

设计

为了使得逻辑更清楚，使用 `task` 封装每一步的操作。由于不存在复用的问题，因此没有设计 `task` 的接口。

测试

8 位二进制码转 BCD 码局限在 0-255 之间，因此直接遍历取值，并进行验证。

通过则显示 passed，未通过则显示错误发生的位置与状态，方便在波形中定位。输出波形在 `test.wlf` 与 `test.vcd` 中，数量过大，因此未添加到本文档中。