



# ——按钮处理设计

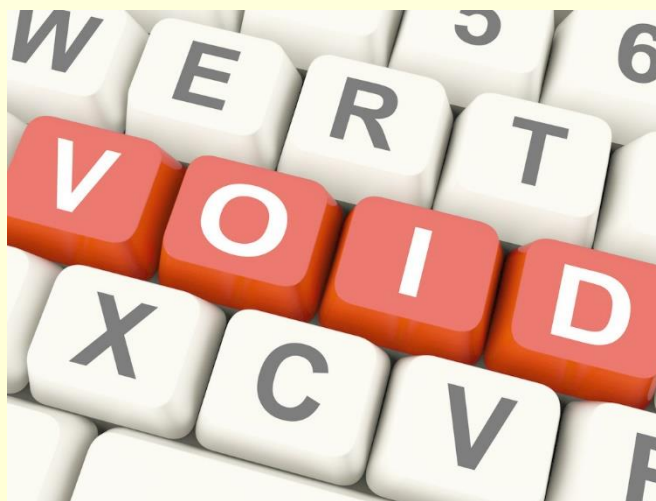
# 本节任务

- 了解按键消抖的意义
- 掌握FPGA边沿检测的设计
- 掌握按键消抖的原理
- 练习状态机的编写
- 掌握矩阵按键的原理及驱动
- 基于小脚丫开发平台实现按键消抖和矩阵键盘的功能

# 概述

键盘作为常用的人机交互工具，用于操作设备运行指令和数据的输入装置，应用于各种电子设备及产品；

电脑、电视、游戏机、电话、计算器、电子琴、仪器仪表、等等；



# 按键

常见的按键有轻触按键和触摸按键：

- 轻触按键：通过内部金属弹片受力弹动实现按键的接通和断开，小脚丫FPGA开发平台使用轻触开关，后面详细介绍
- 触摸按键：通过触摸感应实现，根据原理的不同分为电阻式按键及电容式按键

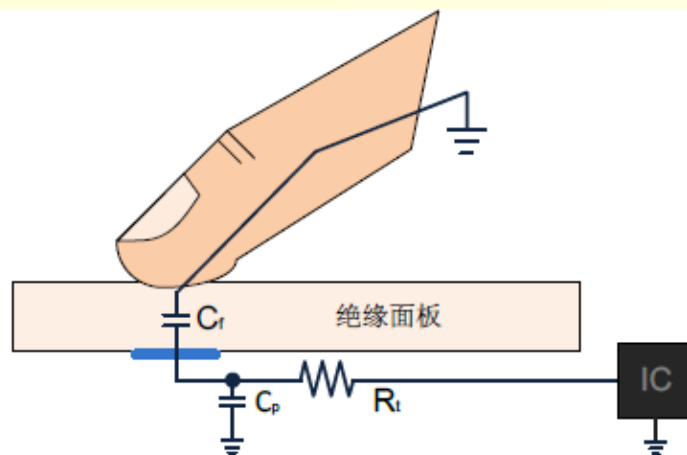
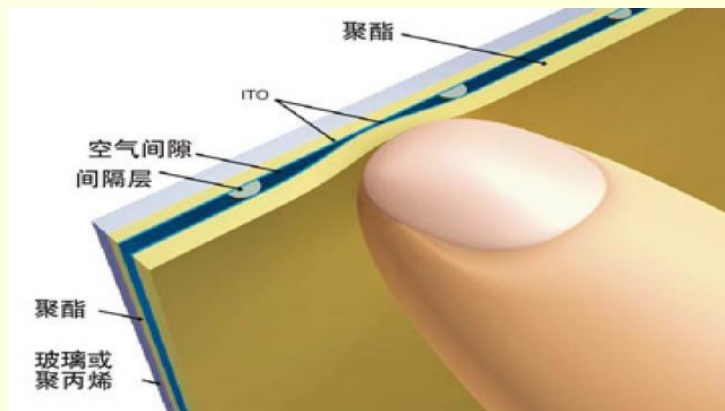


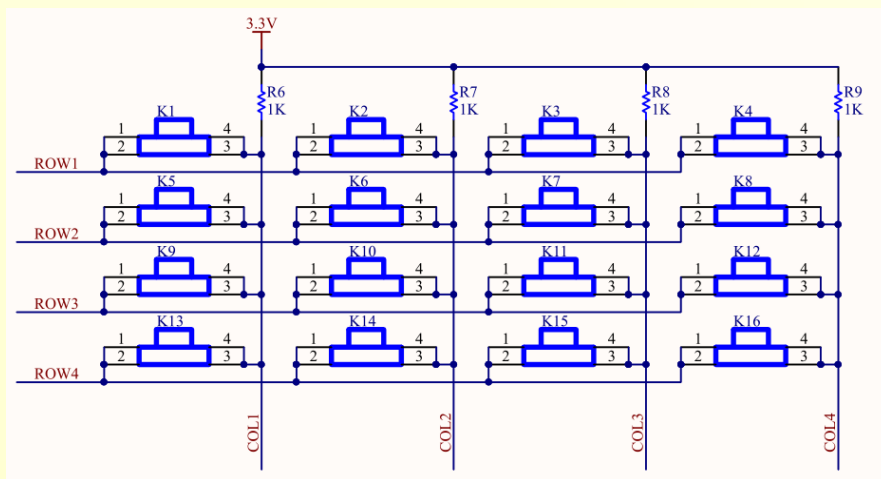
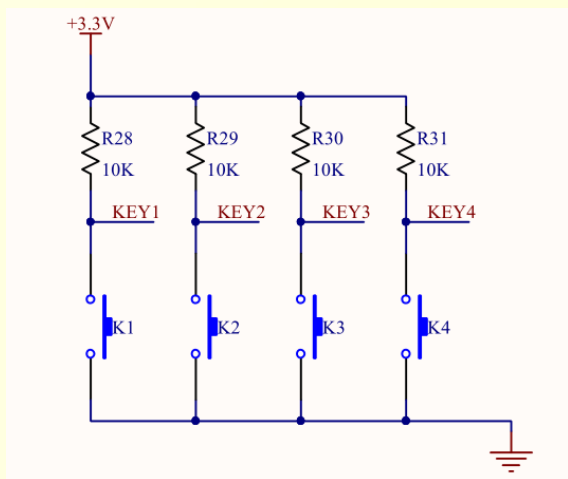
图3 电容式触摸按键原理

# 按键

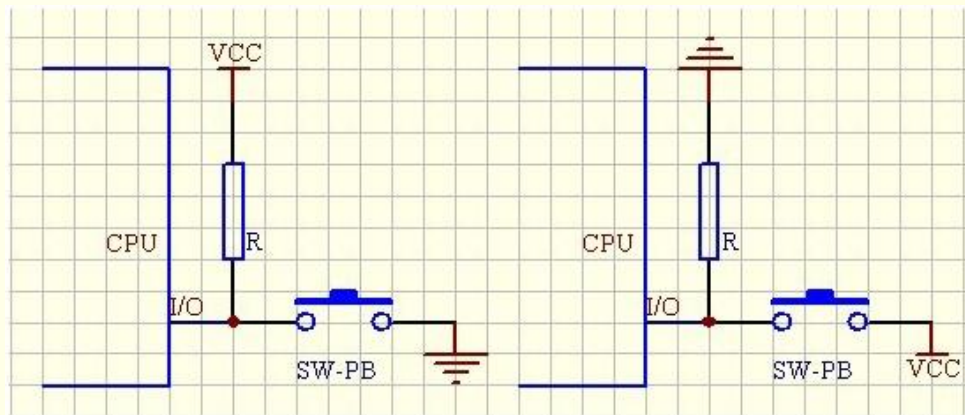
常见的按键连接方式有两种形式：

独立式按键：每个按键单独连接到一个I/O口上，通过判断按键端口的电位识别按键的操作，编程简单，需要更多I/O资源

矩阵式按键：通过行列交叉编码连接，通过分时扫描的方法识别按键的操作，节约I/O资源，编程较复杂



# 轻触按键



根据上图可知：

左侧硬件连接方式：

按下按键：按键接通，按键端口与地导通，为低电平（GND）

松开按键：按键断开，按键端口通过上拉电阻，为高电平（3.3V电压）

右侧硬件连接方式反之



# 按键抖动原理

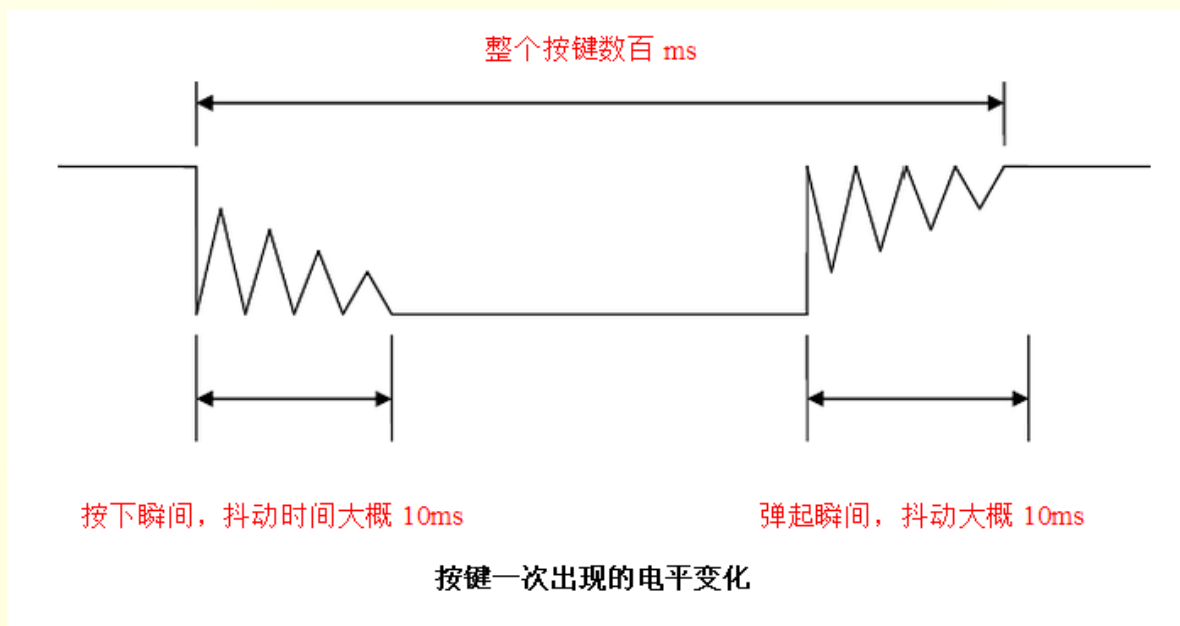
## 抖动的产生：

通常的按键所用的开关为机械弹性开关，当机械触点断开、闭合时，由于机械触点的弹性作用，一个按键开关在闭合时不会马上稳定地接通，在断开时也不会一下子断开。



因而在闭合及断开的瞬间均伴随有一连串的抖动，为了使这种抖动不影响正常的操作，需要在软件中进行优化处理，就是按键消抖。

# 按键抖动原理



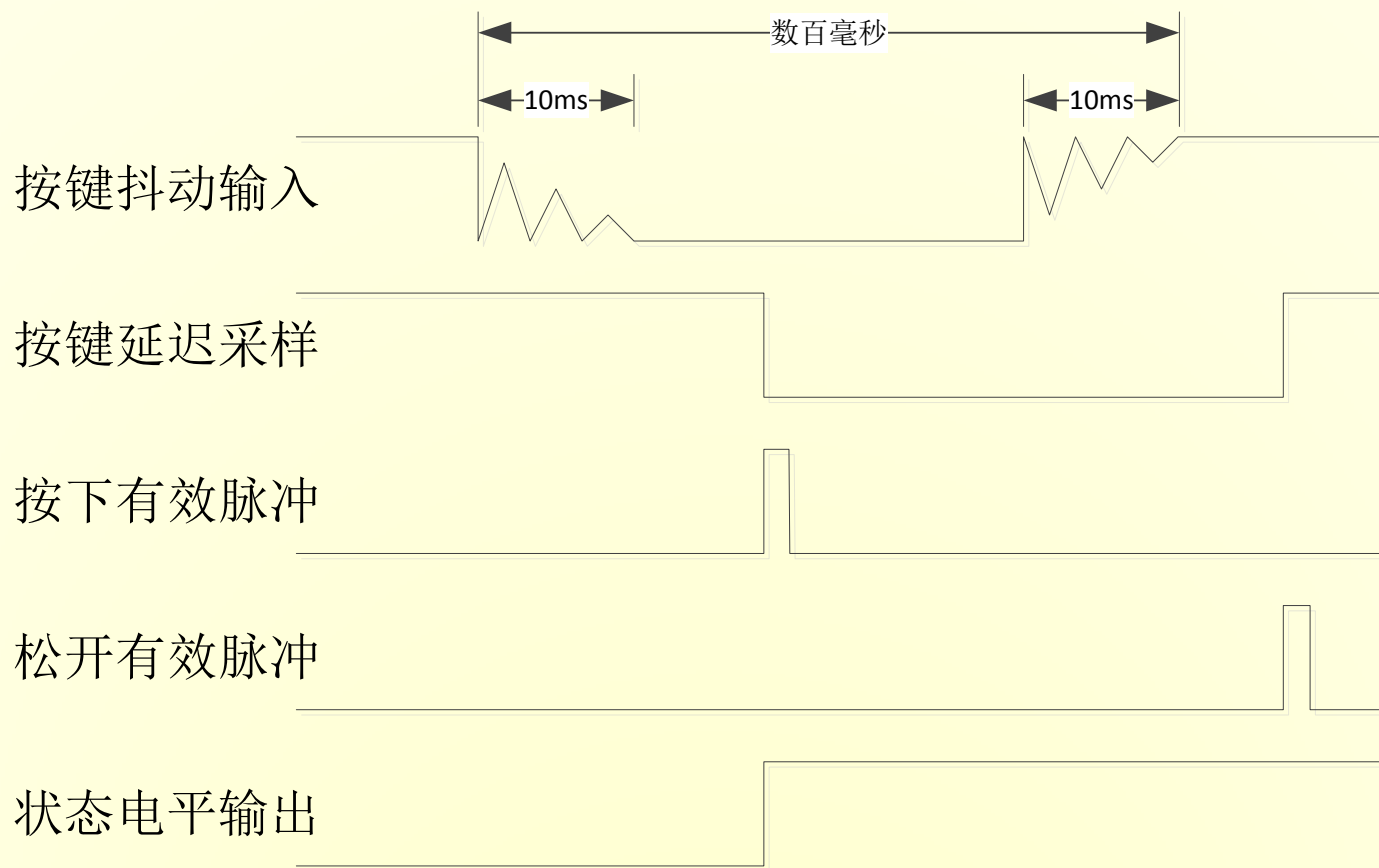
常用的消抖方式有两种:

延时采样: 检测按键电平变化后延时(10ms以上), 采样作为有效信号

周期采样: 每隔固定时间采样(10ms以上), 采样作为有效信号



# 按键消抖示意图



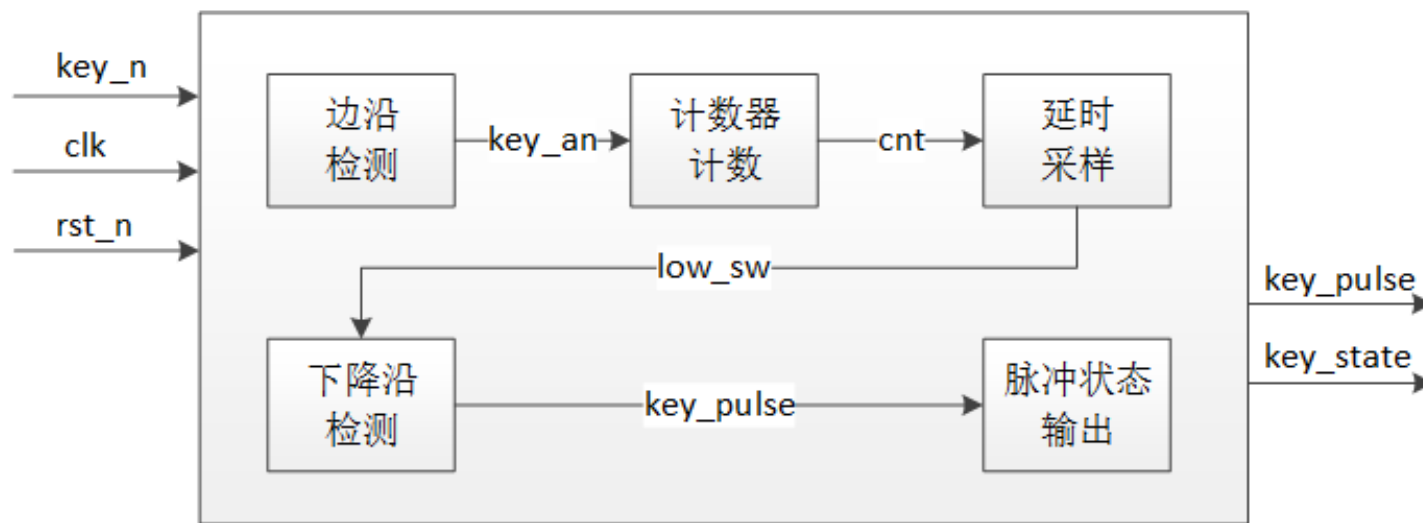
# 延时采样消抖设计框图

边沿检测：按键输入变化检测

计数器计数：计数器计数20ms后采样，将10ms的抖动跨过

延时采样：基于计数器的计时，并采样

下降沿检测：这里使用的是按下有效脉冲输出



# 边沿检测程序实现

边沿检测的方法：

定义一个寄存器对输入信号进行锁存，然后将寄存器的数据与下一个时刻的输入信号做对比，

如果两者不相等则认为输入信号在这个时间段发生了变化，产生了边沿，输出信号key\_an赋值为1，产生一个高电平的脉冲，否则输出0。

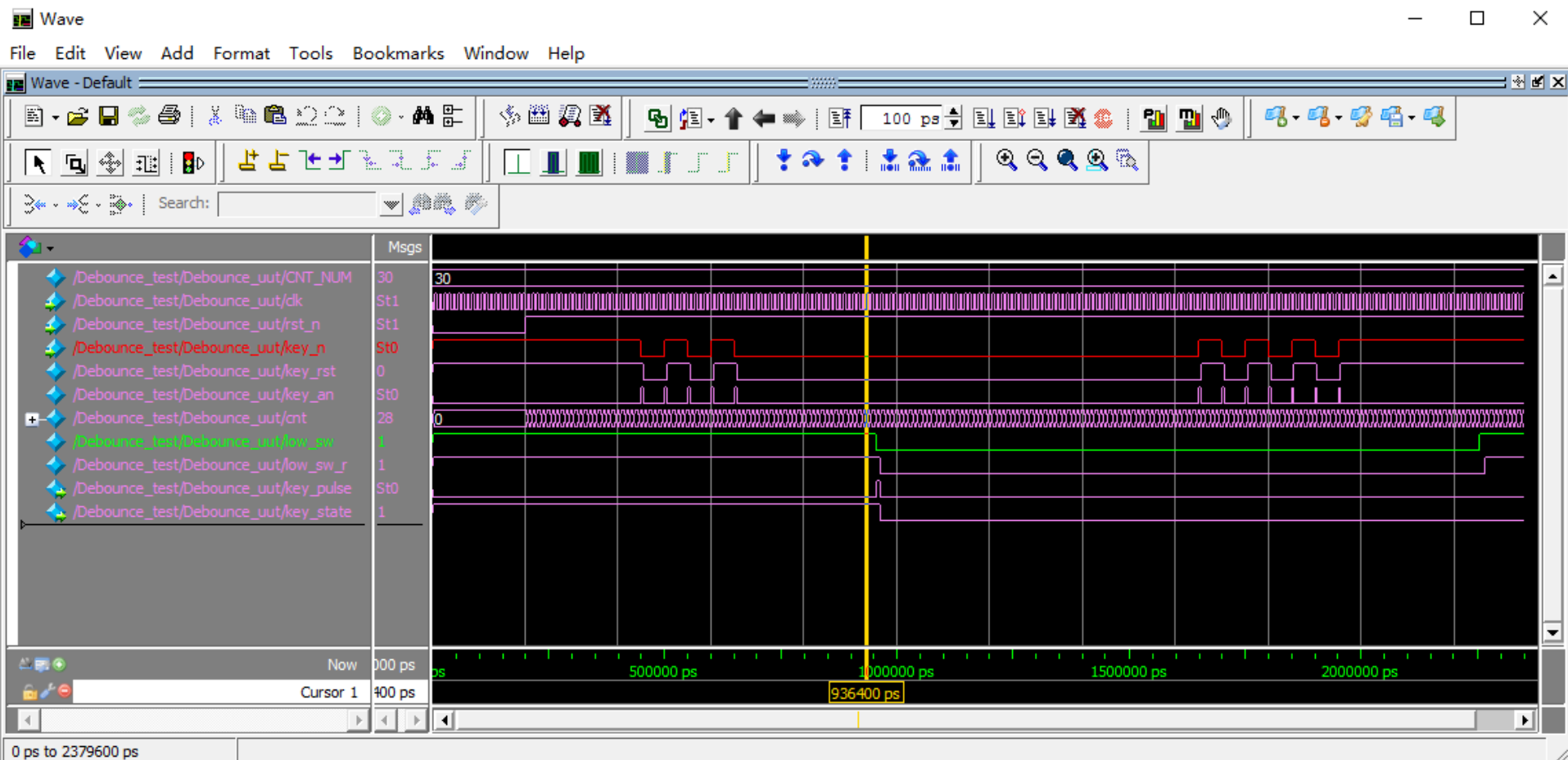
# 延时采样程序实现

边沿检测的方法：

当key\_an产生脉冲时，计数器cnt清零并计数，

因为我们要采集输入信号变化后20ms的状态值，根据系统时钟，设置一个计数器终值 N，计数器计数到N-1时采样key\_an信号得到本模块输出信号low\_sw。

# 延时采样消抖仿真结果



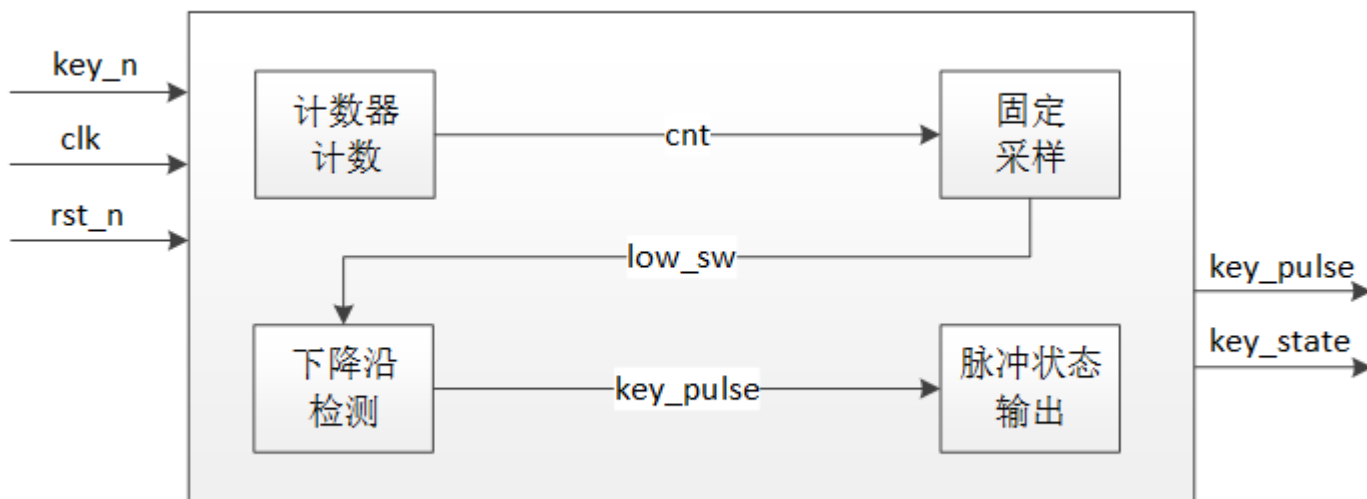
红色key\_n为按键输入，前后各有抖动  
绿色low\_sw为延时采样后的结果，比按键输入稍有延时

# 周期采样消抖设计框图

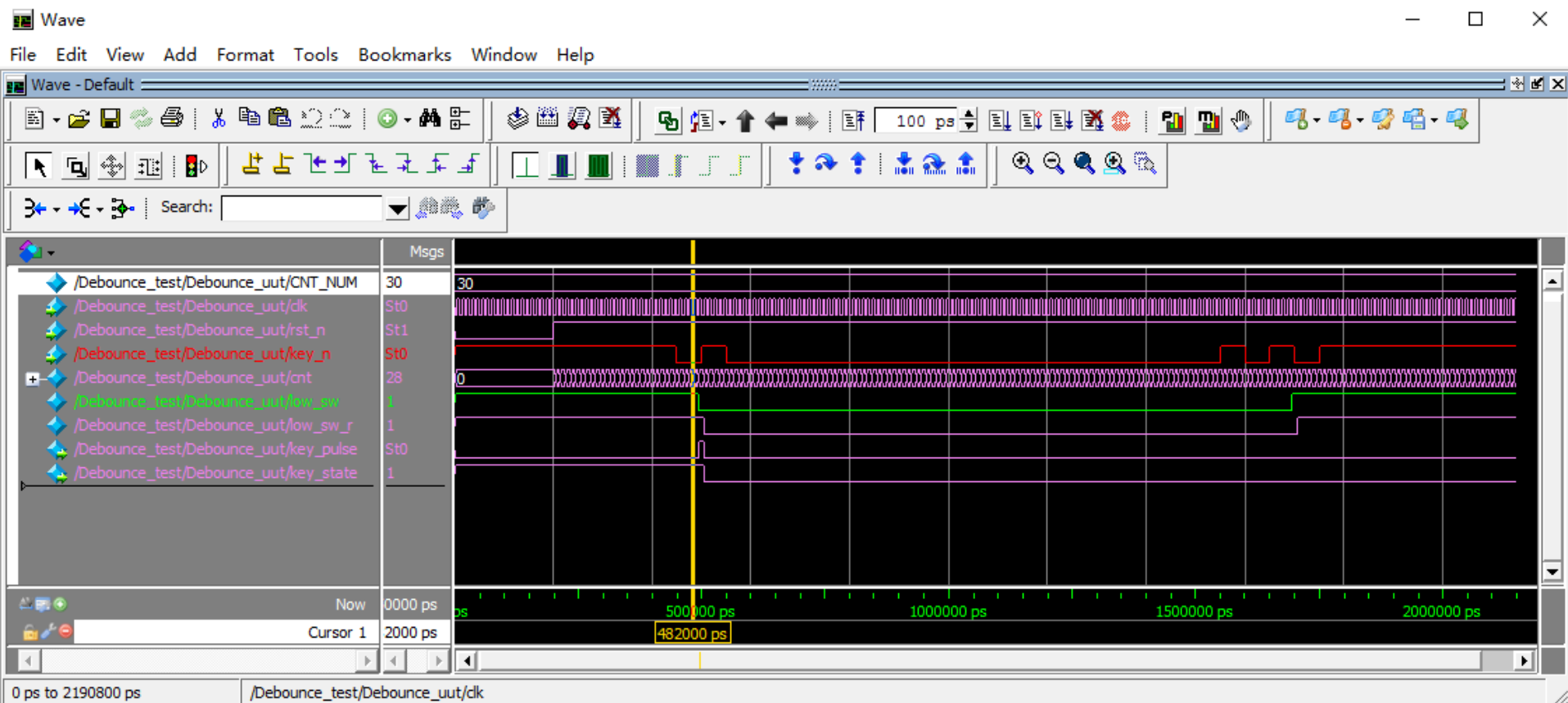
计数器计数：计数器固定计数20ms采样，相邻采样最多只有一个落在不稳定的区间，化不稳定为稳定

固定采样：基于计数器的计时，并采样

下降沿检测：这里使用的是按下有效脉冲输出



# 周期采样消抖仿真结果



红色key\_n为按键输入，前后各有抖动  
绿色low\_sw为周期采样后的结果，无延时

# 矩阵按键原理及优势

## 原理：

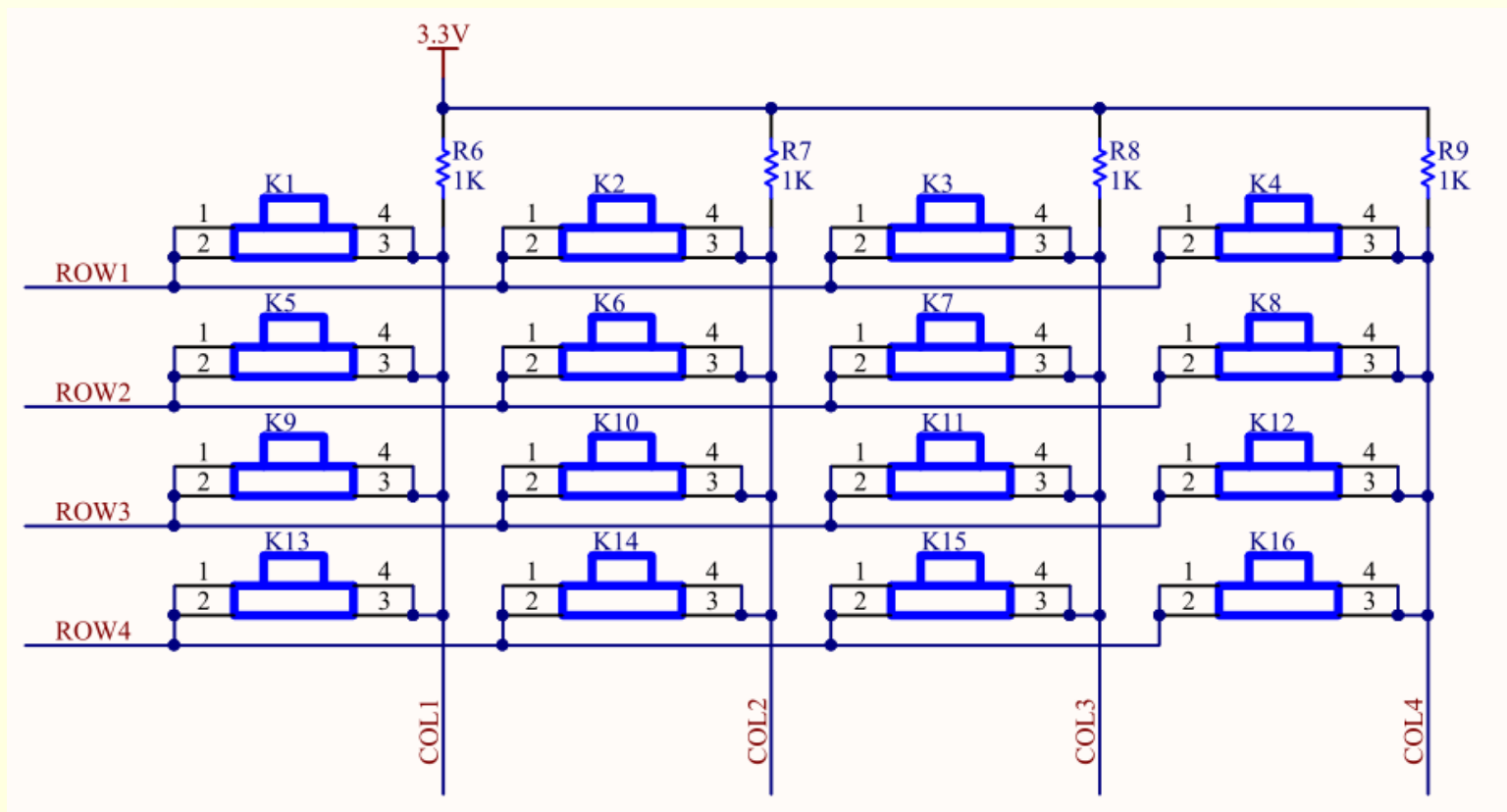
- 将按键按矩阵排列，使用行线和列线分别连接到按键开关的两端，列线通过上拉电阻连接到VCC，
- 当无按键按下时，列线处于高电平的状态，而当有按键按下时，列线电平 由与此列线相连的行线电平决定。
- 最后通过行列扫描法就可以判断各按键的操作状态

## 优势：

- ✓ 相对独立按键来说，当键盘中按键数量较多时，矩阵按键可以节约I/O资源的占用，如4\*4矩阵可以通过8个I/O口实现16个按键的连接，且数量越多优势越明显。



# 矩阵式按键硬件连接

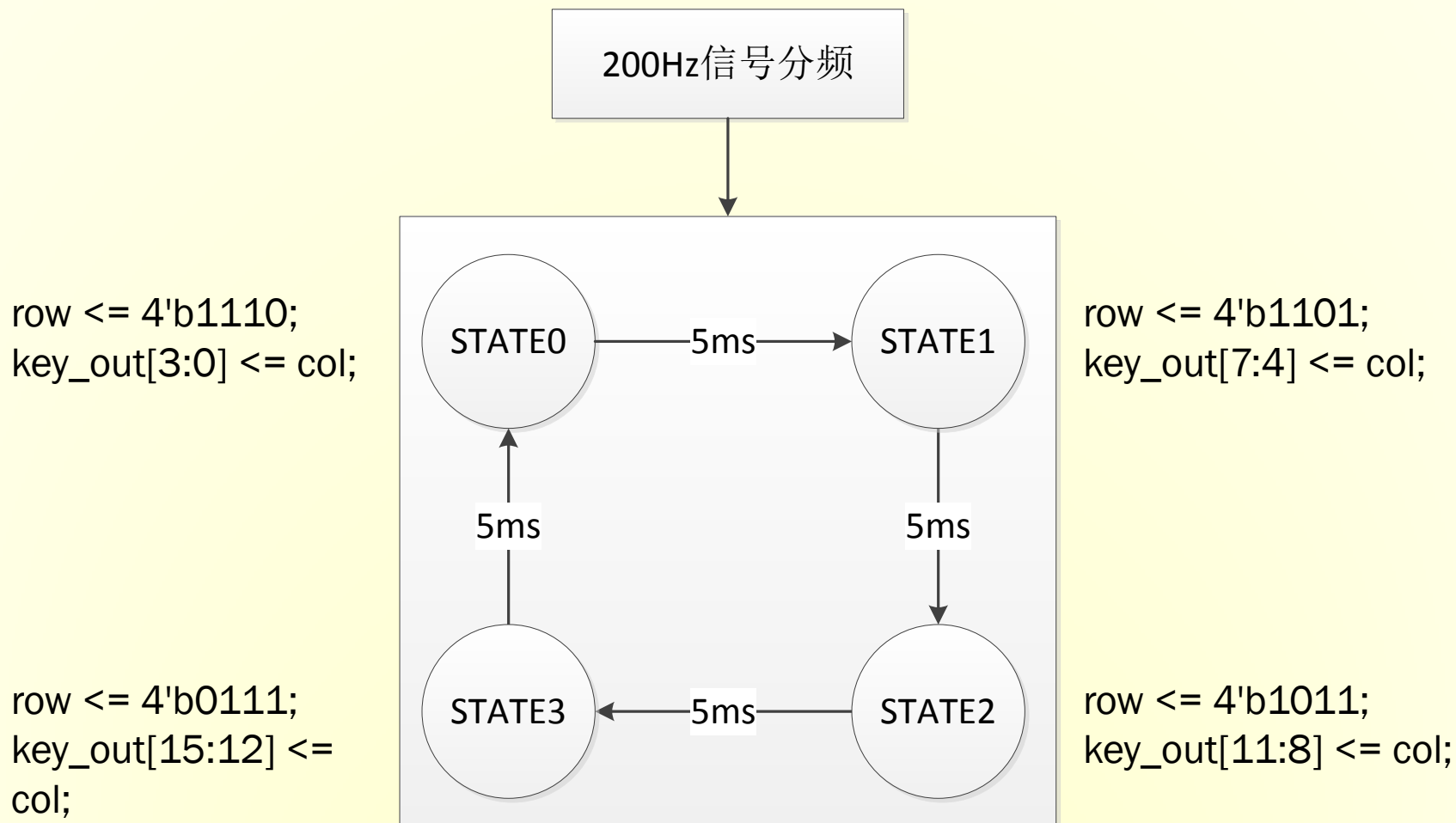


# 矩阵按键行列扫描法

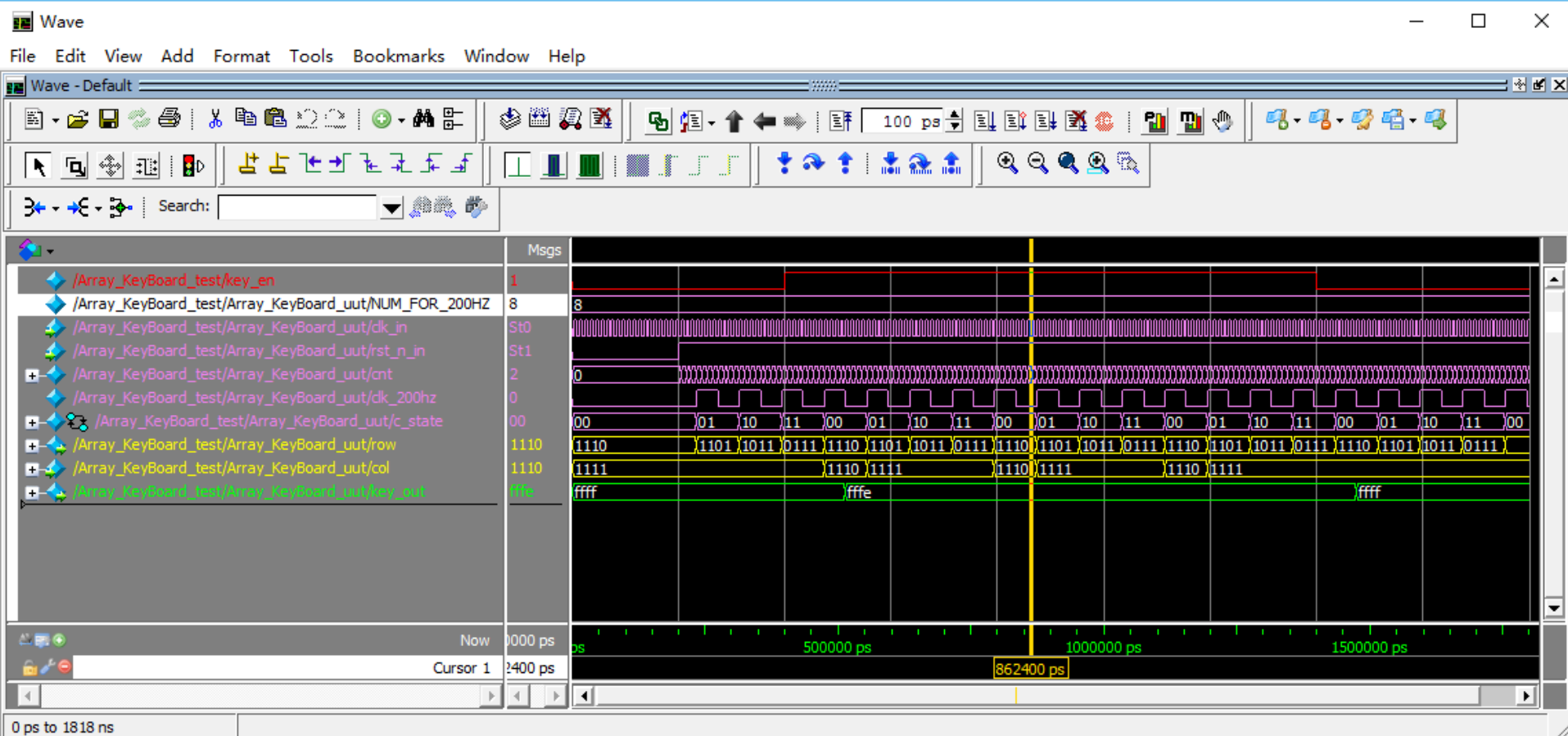
状态	行	列	按键
STATE0	ROW1=0 第1行输出低电平 其余为高电平	判断第1列电平：COL1==0/1	K1按下/松开
		判断第2列电平：COL2==0/1	K2按下/松开
		判断第3列电平：COL3==0/1	K3按下/松开
		判断第4列电平：COL4==0/1	K4按下/松开
STATE1	ROW2=0 第2行输出低电平 其余为高电平	判断第1列电平：COL1==0/1	K5按下/松开
		判断第2列电平：COL2==0/1	K6按下/松开
		判断第3列电平：COL3==0/1	K7按下/松开
		判断第4列电平：COL4==0/1	K8按下/松开
STATE2	ROW3=0 第3行输出低电平 其余为高电平	判断第1列电平：COL1==0/1	K9按下/松开
		判断第2列电平：COL2==0/1	K10按下/松开
		判断第3列电平：COL3==0/1	K11按下/松开
		判断第4列电平：COL4==0/1	K12按下/松开
STATE3	ROW4=0 第4行输出低电平 其余为高电平	判断第1列电平：COL1==0/1	K13按下/松开
		判断第2列电平：COL2==0/1	K14按下/松开
		判断第3列电平：COL3==0/1	K15按下/松开
		判断第4列电平：COL4==0/1	K16按下/松开



# 矩阵式按键设计框图



# 矩阵按键仿真结果



红色key\_n为Testbench中模拟的按键信号  
黄色col和row对应矩阵按键的行列信号  
绿色key\_out为16个按键分别的状态输出