# 实验5

#### 状态机 sim

						0.000 113														5.000 n
Value	0.000 ns				ns				500.000	ns 6			700.000 ns	800.000			ns 1	,000.000 ns		
0					Ш					Ш		Ш		тин	Ш	ППП	Ш		imi	ППП
1																				
1111	1111	X 1116	tx	1111	1111 1101		1111		1811		1111		0111	i		1111		1110	1111	
1111	1111	X 1110	1101	1011	8111	1110	118	1011	8111	1118	1181	X 18	11 0111	1118	118	1011	011	1 1110	1101	1011
0000	0	888	TX	0881				8882				8884				0008				8818
0		9	TX	X 4								ь								
0							hΙ				<u> </u>				л					
0																		П		
00	88	81	82	83	84	(85 X	86	( 87 X	88	89	0a	X 80	) 0c	8a	0e	) er	10	\ 11 \	12	13
0							Ш	П					П					ПП		
00000005				aaaooops																
0																	ш	П		
f	+		X · X	<b>b</b>	7	X	d	X • X	7	$\longrightarrow$ X	d	X	7		d	χь	χ	χ • χ	d	χ
	0 1 1111 1111 0000 0 0 0	0 1 1111 1111 0000 0 0 0 0	Value 0 000 ns 130 0 1 1111 1111 1111 1111 111 0000 0000 0 00 00 01 0 00 00 01	Value 0 000 ns 1000 000 ns 10 1000 000 ns 10 1000 000 ns 11 100 1111 1111	0 1 1111 1110 1111 1111 1111 1111 1111	Value 0 000 ns 1000 600 ns 200 000 ns 11111 1111 1111 1111 1111 1111	0 000 ns 1000 000 ns 200 000 ns 200 000 ns 100 00 ns 100 ns 1	Value 0 000 ns 100 600 ns 200 000 ns 100 000 ns 100 110 1111 1111 1	Value	Value 0 000 ns 1000 080 ns 200 080 ns 300 080 ns 400 080 ns 500 080 ns 500 080 ns 100 080 ns 500	Value 0 000 ns   100 000 ns   200 000 ns   200 000 ns   400 000 ns   500 000 ns   60 000 n	Value 0 000 ns 100 000 ns 200 ns 200 ns 200	Value 0 000 ns 1501 982 ns 200 000 ns 1500	Value 0 000 ns 1001000 ns 200 000 ns 200 ns 200 000 ns 200	Value 0 000 ns   100   600 ns   200 000 ns   200 000 ns   400 000 ns   400 000 ns   200 000005	Value	O	Value 0 000 ss 100 300 75 200 000 ns 200 ns 200 000 ns 200 ns 200 000 ns 200 ns 20	Value 0 000 cs 100 3 000 cs 200 cs 200 000 cs 200 cs 200 000 cs 200 cs 200 000 cs 200 00	Value 0 000 ns 1501958 ns 200 000

#### ZZ

- (1) 65ns, cnt\_end=1, 状态 f->e, 经过一个周期, col = 1110, row = 1110 (按下), 直到松开, 有 keyboard\_num = d
- (2) 365ns, cnt\_end=1, 状态 e -> d, 经过一个周期, col = 1110, row = 1101 (按下), 直到松开, 有 keyboard\_num = c
- (3) 555ns, cnt\_end=1, 状态 e -> d, 经过一个周期, col = 1110, row = 1101(按下), 直到松开, 有 keyboard\_num = b
- (4) 605ns, cnt\_end=1, 状态 e -> d, 经过一个周期, col = 1110, row = 1011(按下), 直到松开,有 keyboard\_num = a
- (5) 1145ns, cnt\_end=1, 状态 e -> d, 经过一个周期, col = 1101, row = 1101 (按下), 直到松开, 有 keyboard\_num = f(#)
- (6) 1325ns, cnt\_end=1, 状态 e -> d, 经过一个周期, col = 1101, row = 1101 (按下), 直到松开, 有 keyboard\_num = 9
- (7) 1565ns, cnt\_end=1, 状态 e -> d, 经过一个周期(10ns), col = 1101, row = 1011(按下), 直到松开(1635ns), 有 keyboard\_num = 6

# 状态定义

```
parameter IDLE = 4'b1111;
parameter S0 = 4'b1110;
parameter S1 = 4'b1101;
parameter S2 = 4'b1011;
parameter S3 = 4'b0111;

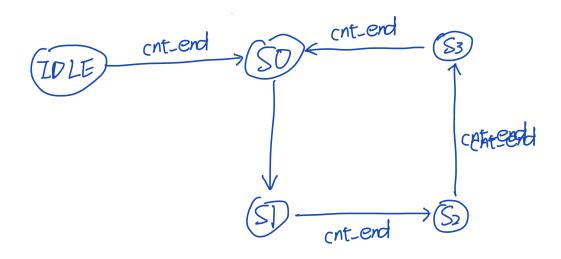
// 这些是条件,但是其实条件是差不多的,都是 cnt_end
wire idle_s0_con = current_state == IDLE && cnt_end;
wire s0_s1_con = current_state == S0 && cnt_end;
wire s1_s2_con = current_state == S1 && cnt_end;
```

```
wire s2_s3_con = current_state == S2 && cnt_end;
wire s3_s0_con = current_state == S3 && cnt_end;
```

cnt\_end 表示: 计数器 计算到头的时候。

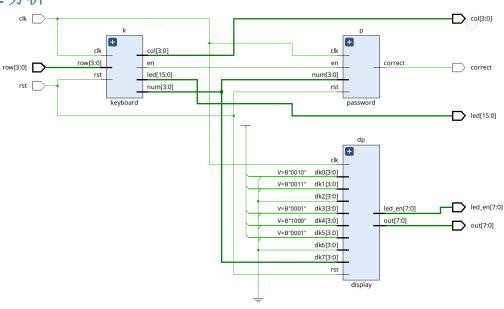
# 状态转移图

状态转移图 # TODO



#### Alt text

# RTL 分析

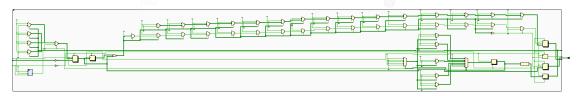


ZZ

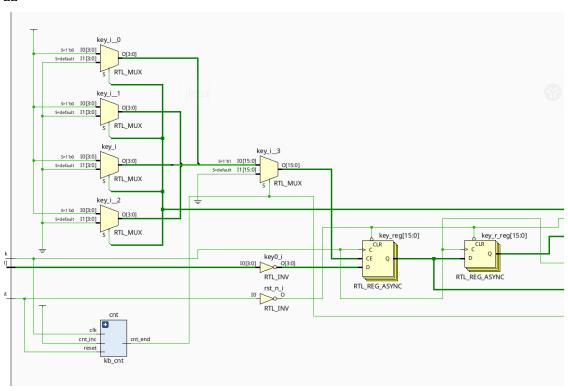
keyboard 是用来 检测按下了什么 键

password 是用来 检测密码是否输入正确的,里面也是一个状态机 display 是用来 显示的,其中 keyboard 的输出 num,只连接了 display 的 dk7

# keyboard

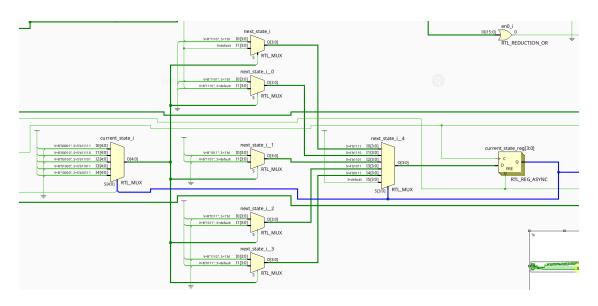


#### ZZ



ZZ

左边这一块是用来:给 寄存器堆 的相应的 row 赋值

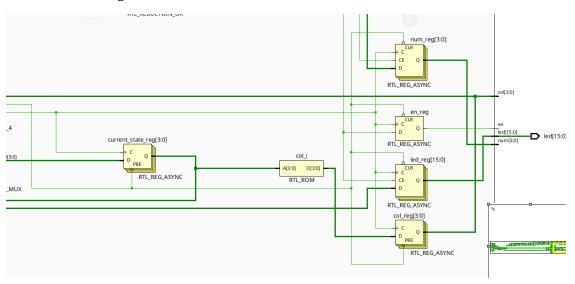


ZZ

这段就对应的是状态机

current\_state\_reg 是存储状态的

current\_state\_reg 左边是: 状态转移的



ZZ

这里靠右边的部分就是:输出逻辑

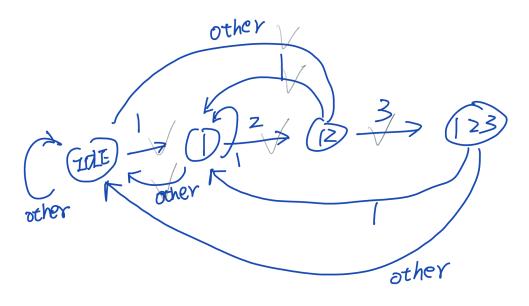
# 附加题 (密码锁)

# 设计说明

正确的密码是: "123"

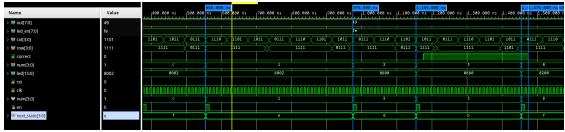
如果输入正确,K17 也就是最左边的 led 会亮(也就是 correct 变量为 1)

#### 状态转移图



### Alt text

#### 仿真分析



ZZ

- (1) 555ns 时, col = 0111 时, row = 0111 弹起, 状态: f-> e
- (2) 975ns 时, col = 1011 时, row = 0111 弹起, 状态: e -> d
- (3) 1155ns 时, col = 1101 时, row = 0111 弹起, 状态: d -> b
- (4) 1175ns 时, 由输出逻辑, correct 根据状态 correct = 1
- (5) 1455ns 时, col = 1011 时, row = 1101 弹起, 状态: b -> f
- (6) 1635ns 时, col = 1011 时, correct = 0