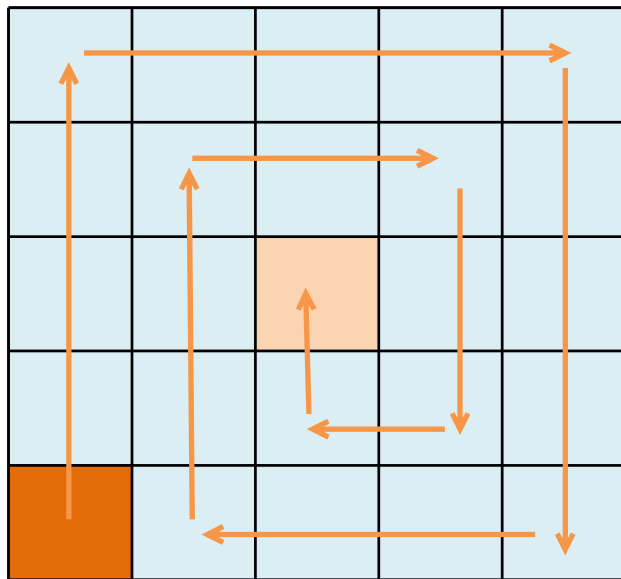


一笔画游戏

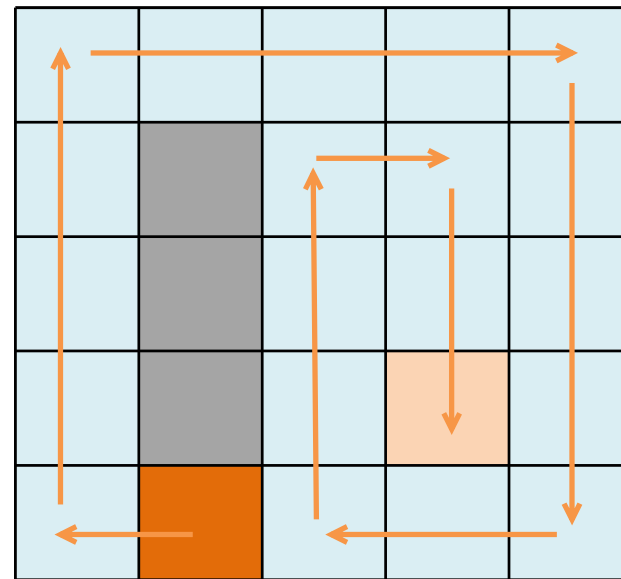
61010126 张居谋

游戏简介

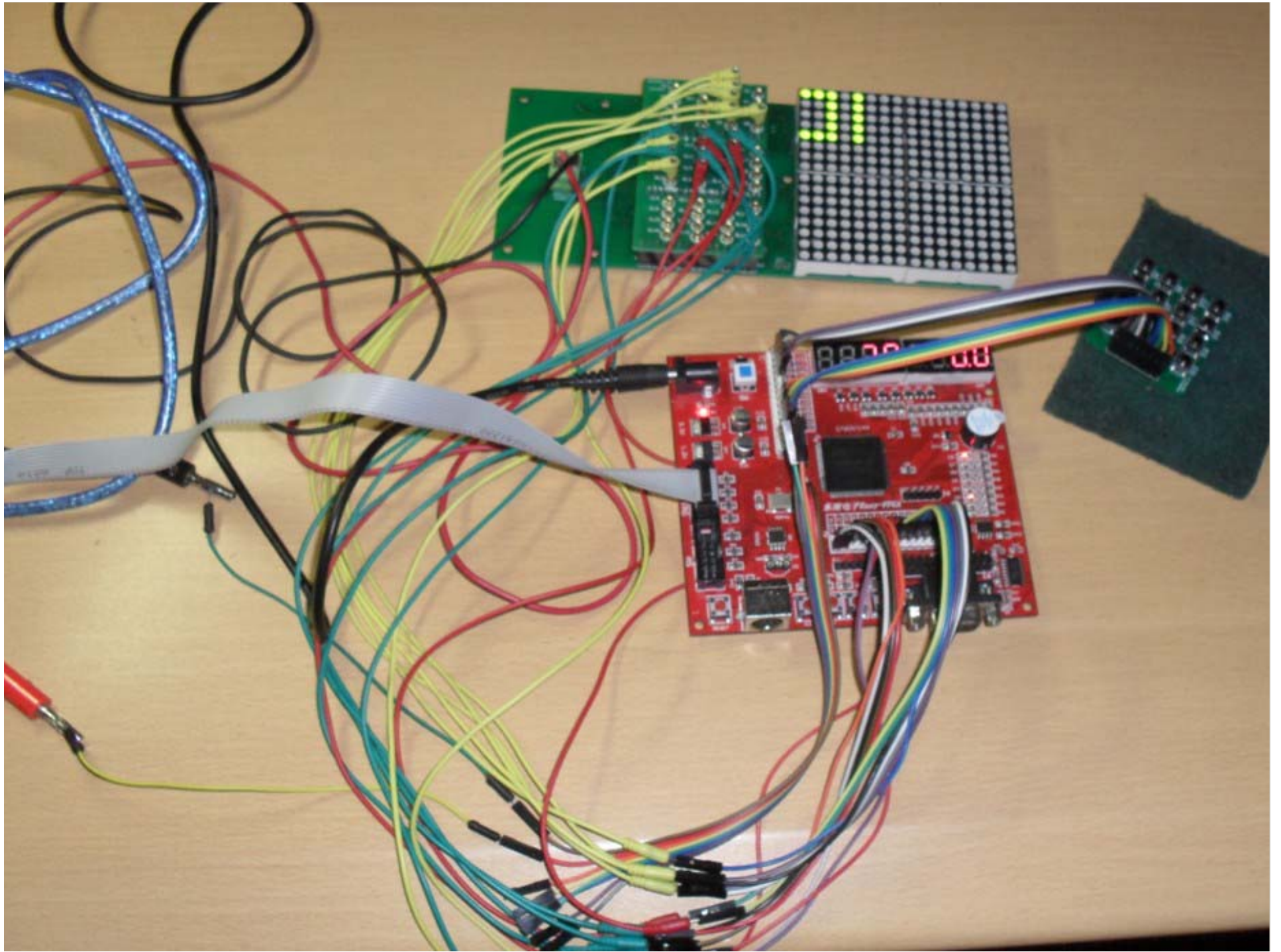
在5*5的点阵上有一些障碍点，要求将剩下的点连接起来，每个点必须走一次且只能走一次。



WIN!!



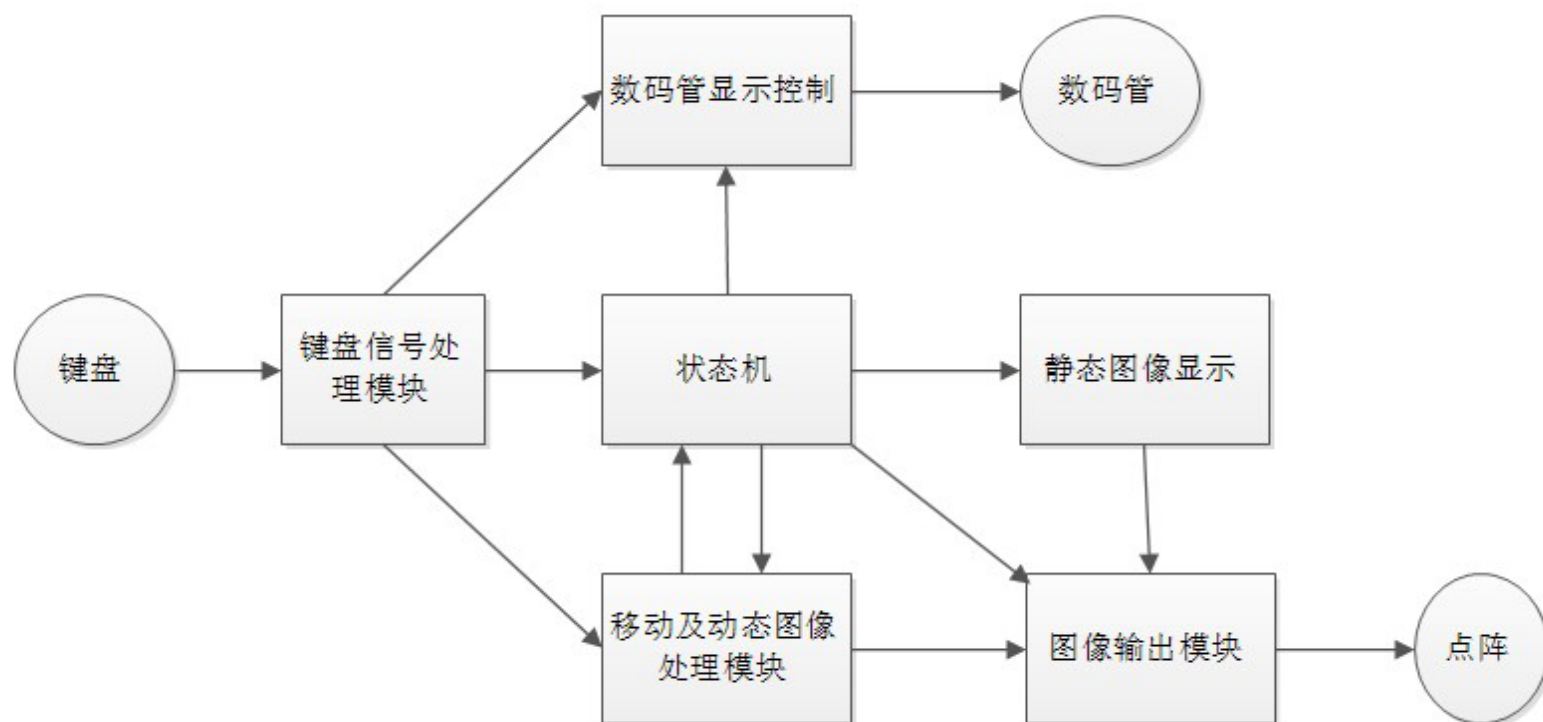
FAIL..



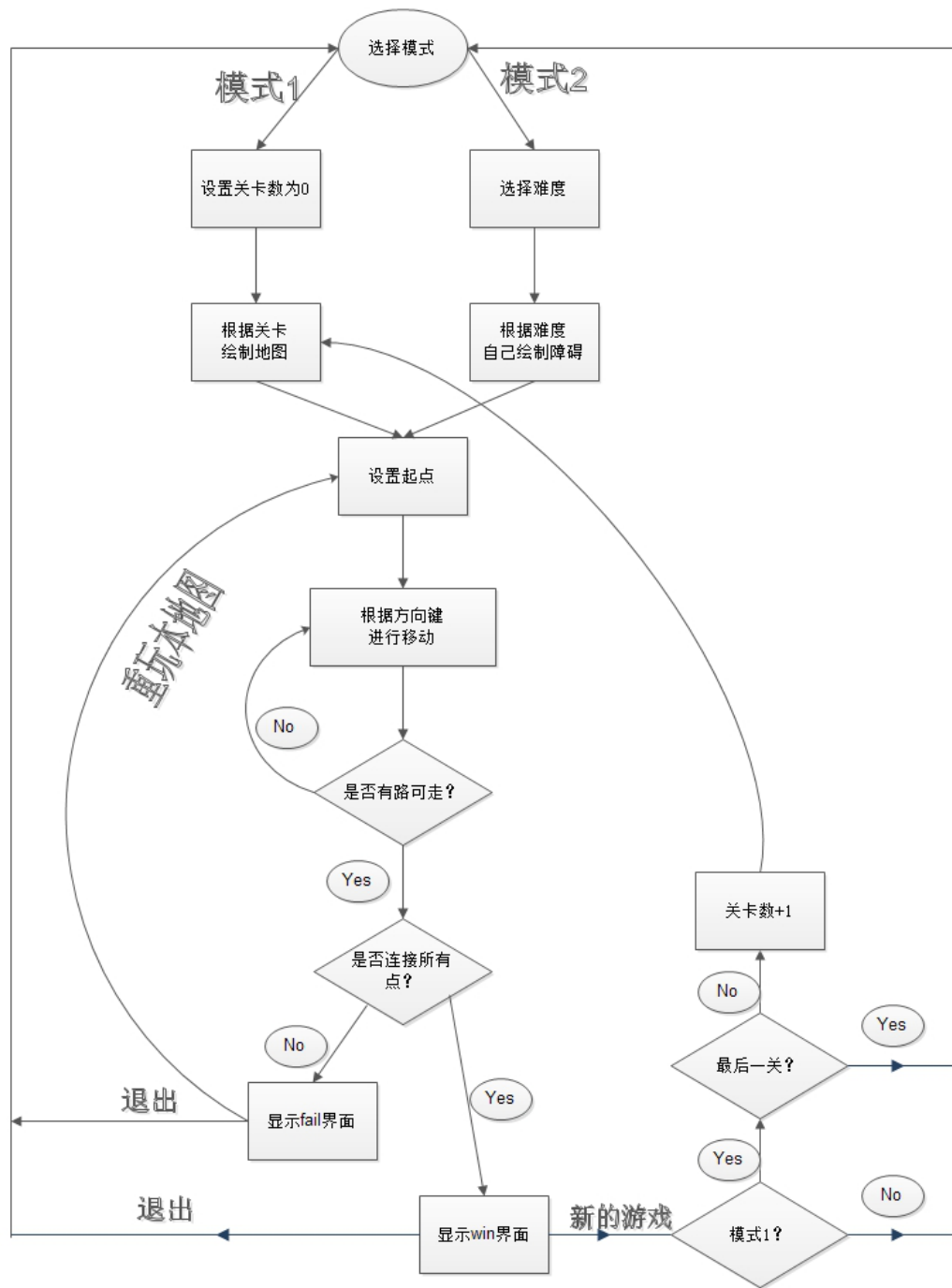
主要工作

- ◆ 熟悉自购的FPGA开发板
- ◆ 熟悉数码管、点阵板的使用
- ◆ 编写程序
- ◆ 调试与优化程序

结构框图



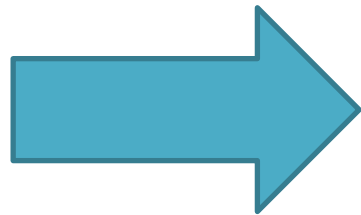
流程图



项目亮点

- ◆ 实现了以红、绿、橙**3**种颜色滚动显示“WIN!!”、“FAIL..”等提示文字
- ◆ 矩阵的转换
- ◆ 同一段代码实现不同的功能

三色滚动显示



7位寻址
abxxxxx

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	31	2	28	2	31	0	31	17
8	17	31	0	31	17	17	14	0
16	31	21	21	0	0	0	31	2
24	28	2	0	0	0	0	0	0
32	31	17	17	14	0	31	5	5
40	0	1	31	1	0	1	2	28
48	2	1	0	0	0	0	31	17
56	17	14	0	0	0	0	0	0
64	31	8	7	8	31	0	17	31
72	17	0	31	2	4	8	31	0
80	23	0	23	0	0	0	31	8
88	7	8	0	0	0	0	0	0
96	31	5	5	0	30	5	5	30
104	0	17	31	17	0	31	16	16
112	0	16	16	0	0	0	31	5
120	5	0	0	0	0	0	0	0

MODE

DFTY

WIN!!

FAIL..



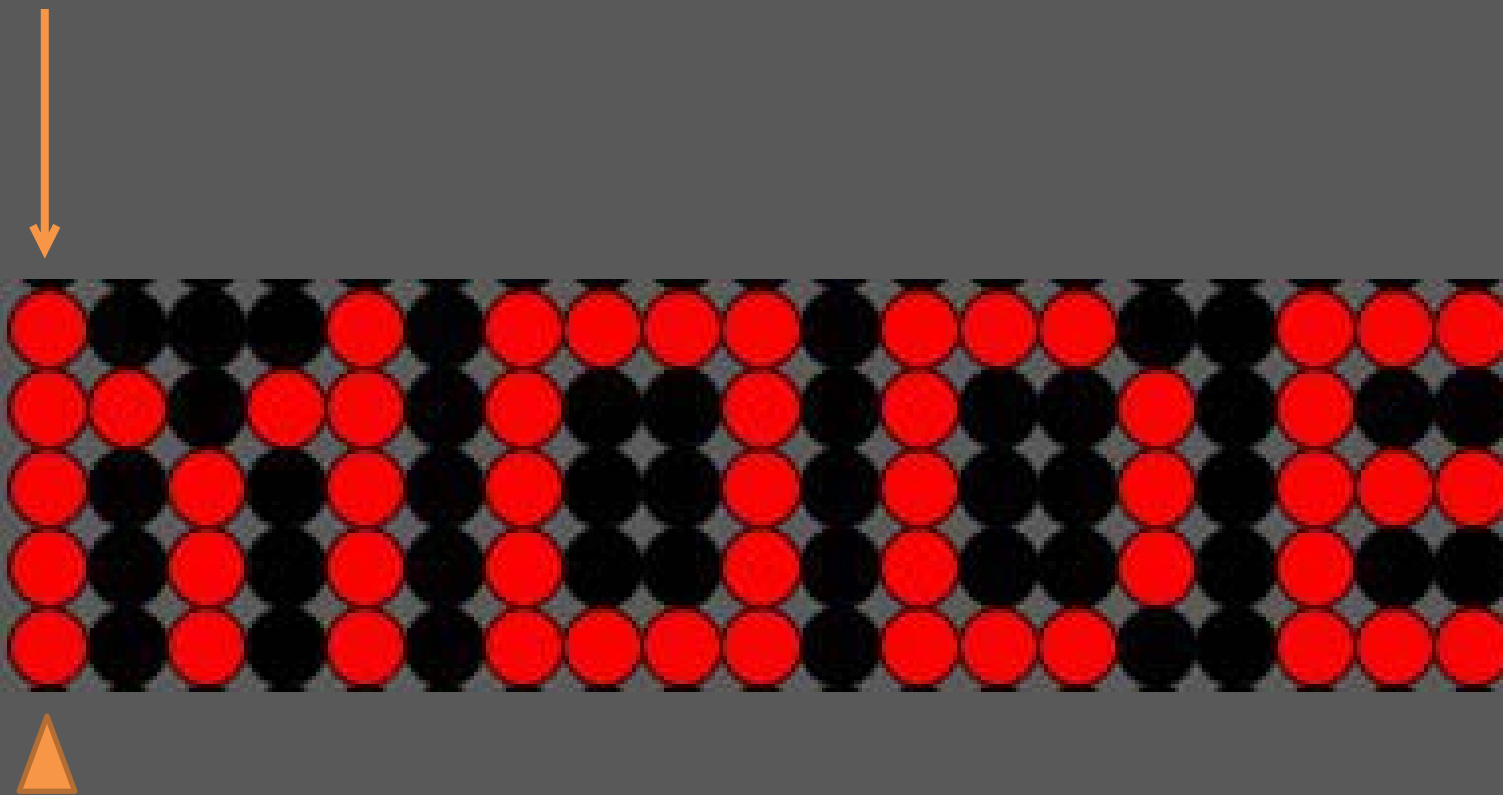
5位输出

ROM

滚动原理

CNT1: 0

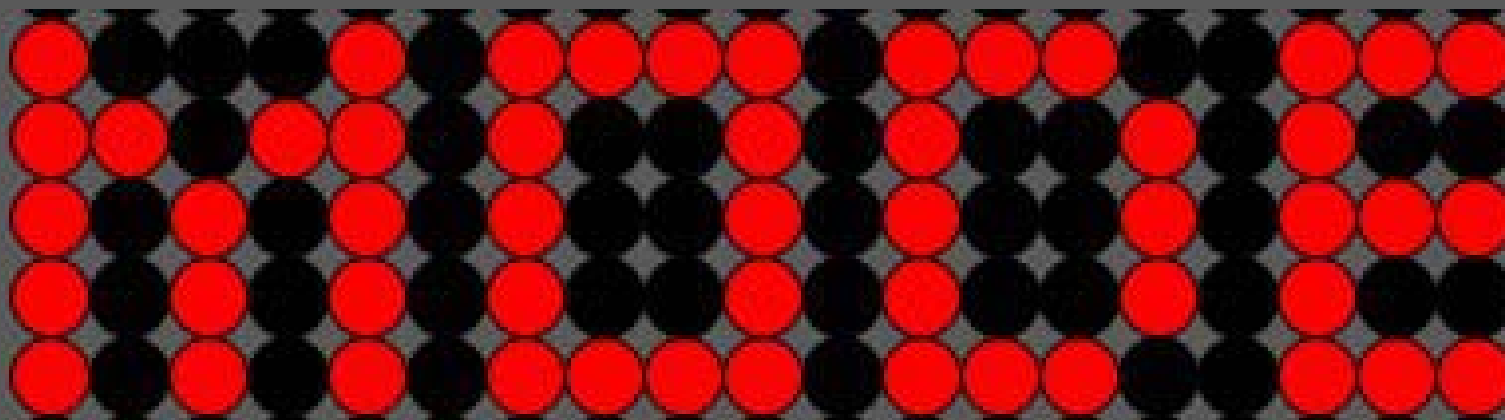
CNT2: 0



滚动原理

CNT1: 1

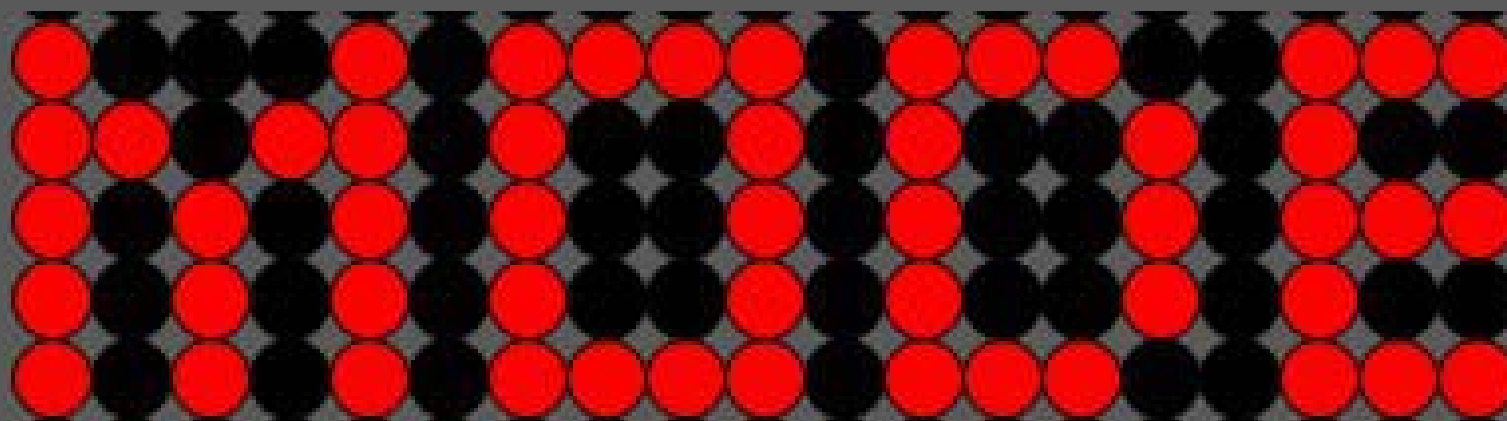
CNT2: 0



滚动原理

CNT1: 2

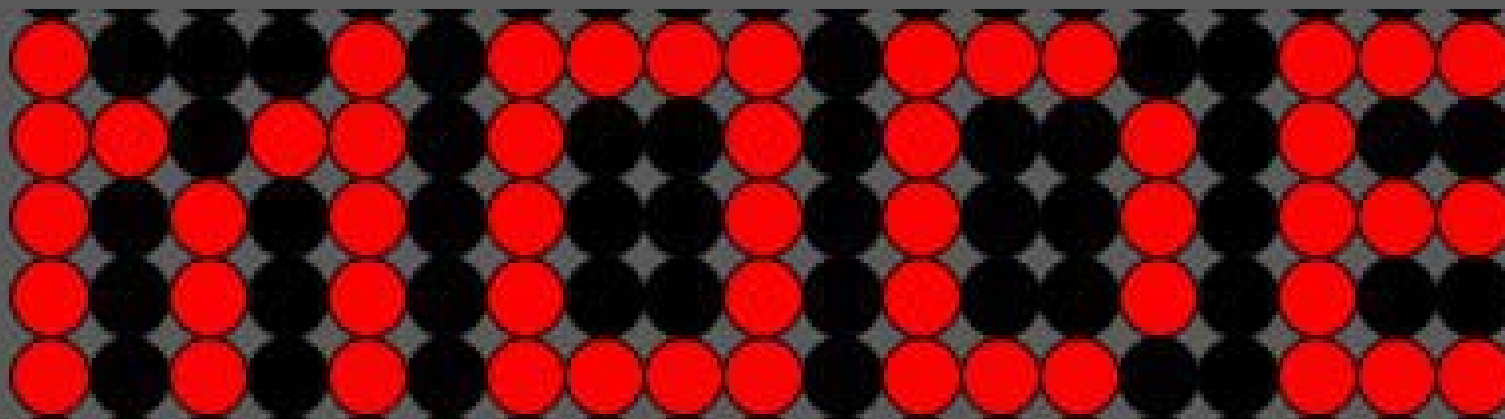
CNT2: 0



滚动原理

CNT1: 3

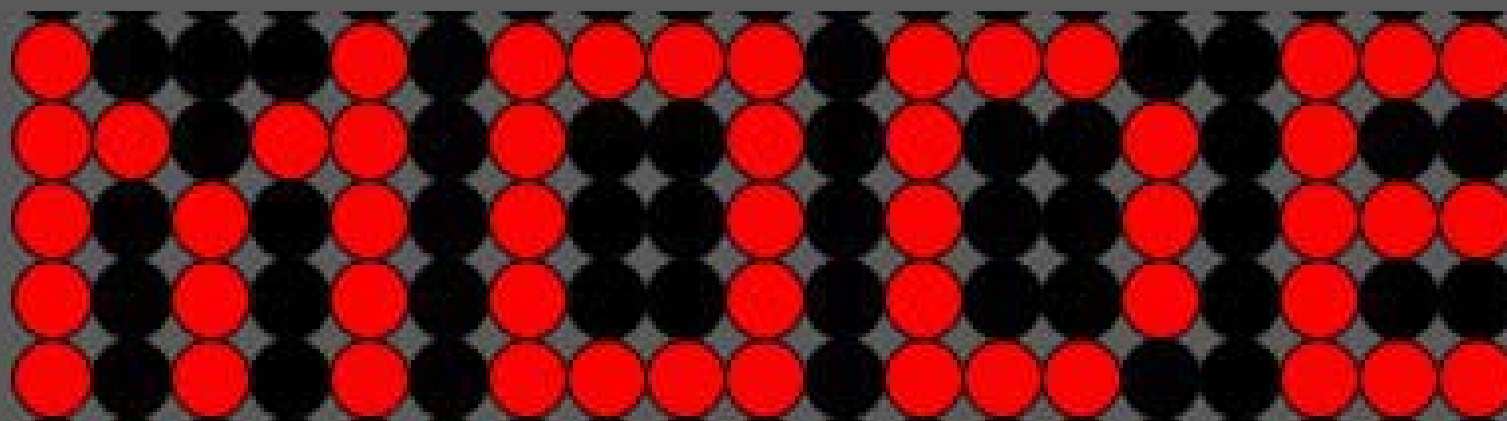
CNT2: 0



滚动原理

CNT1: 4

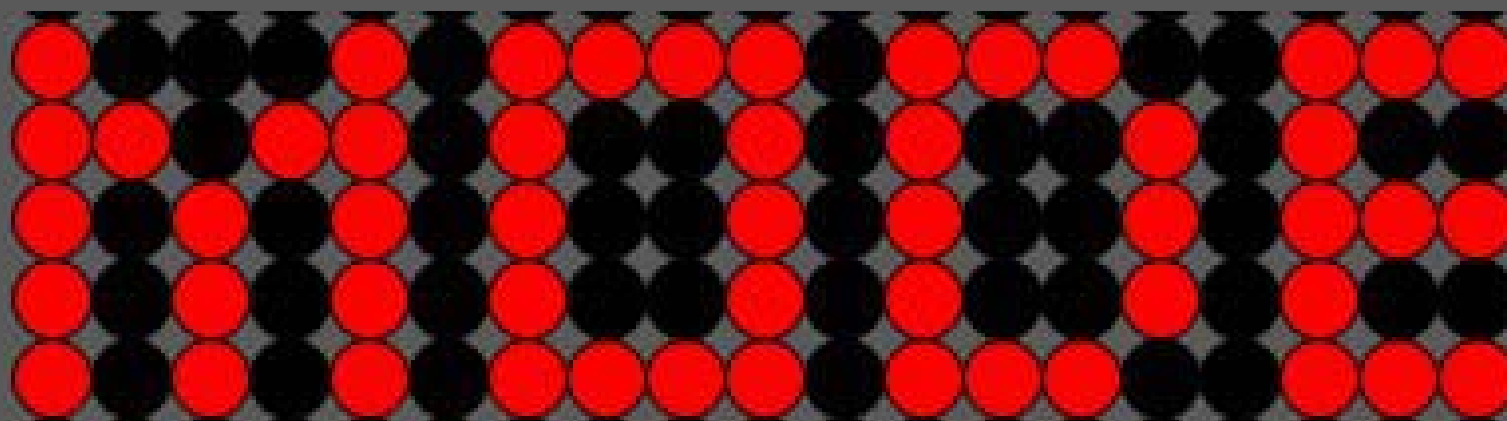
CNT2: 0



滚动原理

CNT1: 4

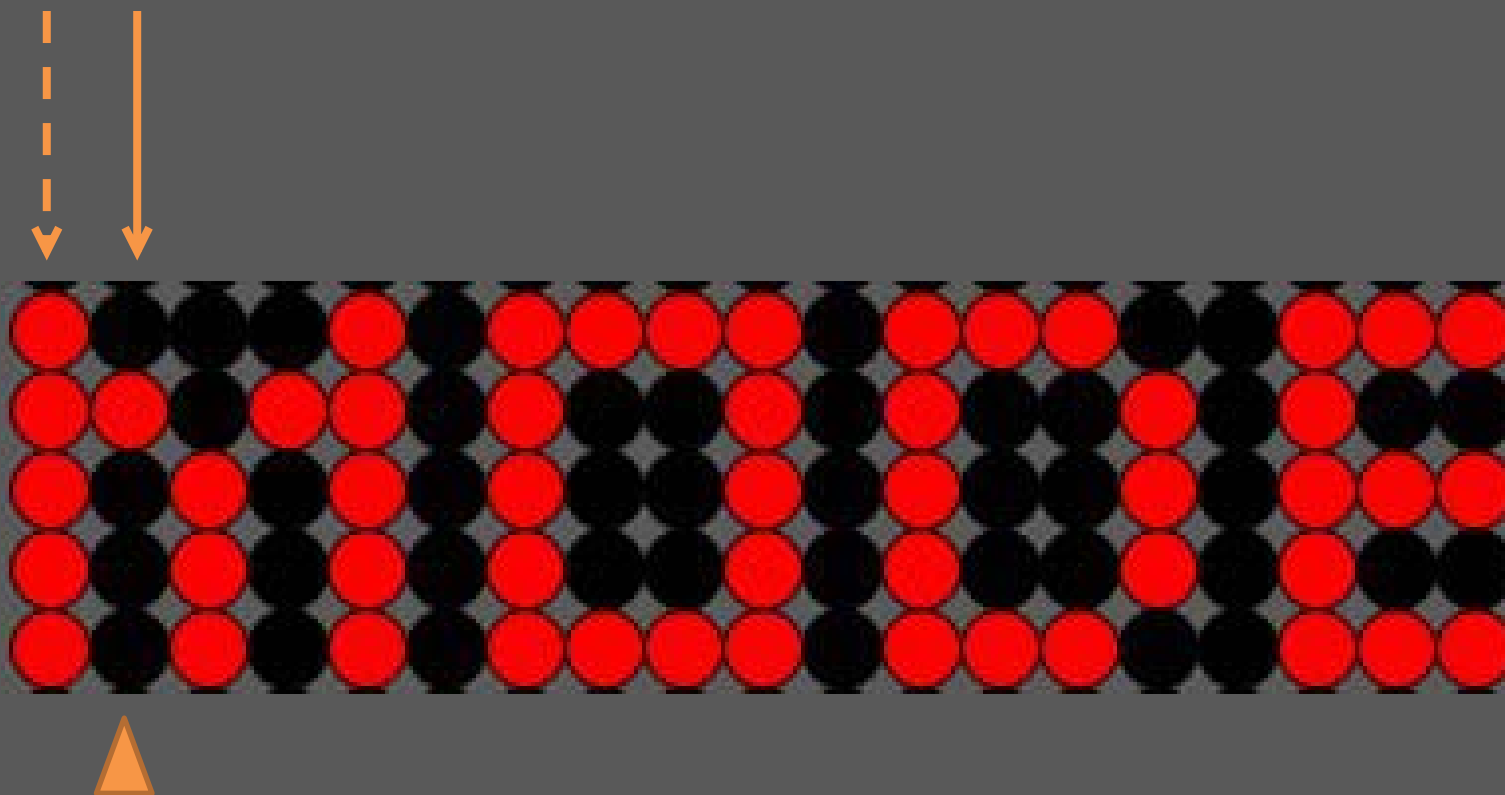
CNT2: 1



滚动原理

CNT1: 0

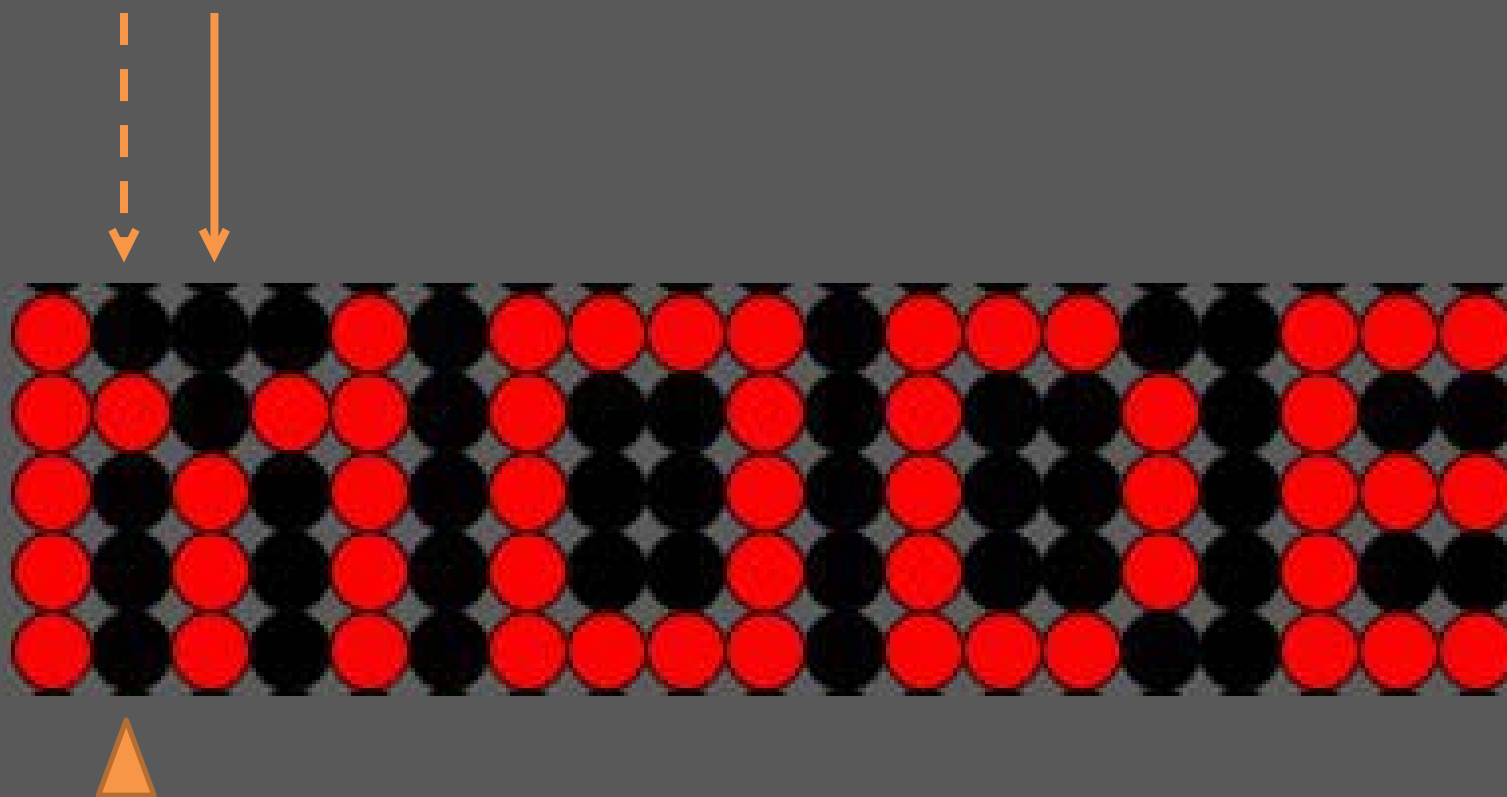
CNT2: 1



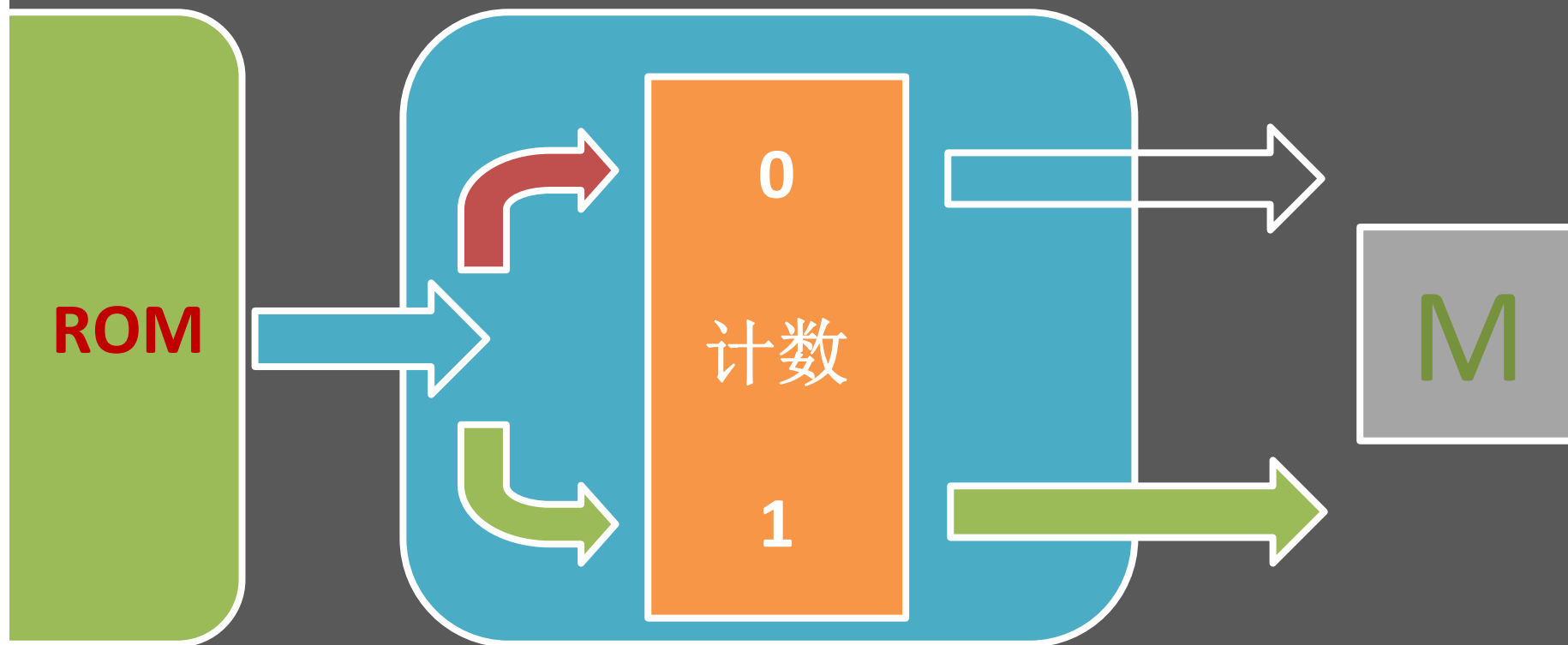
滚动原理

CNT1: 1

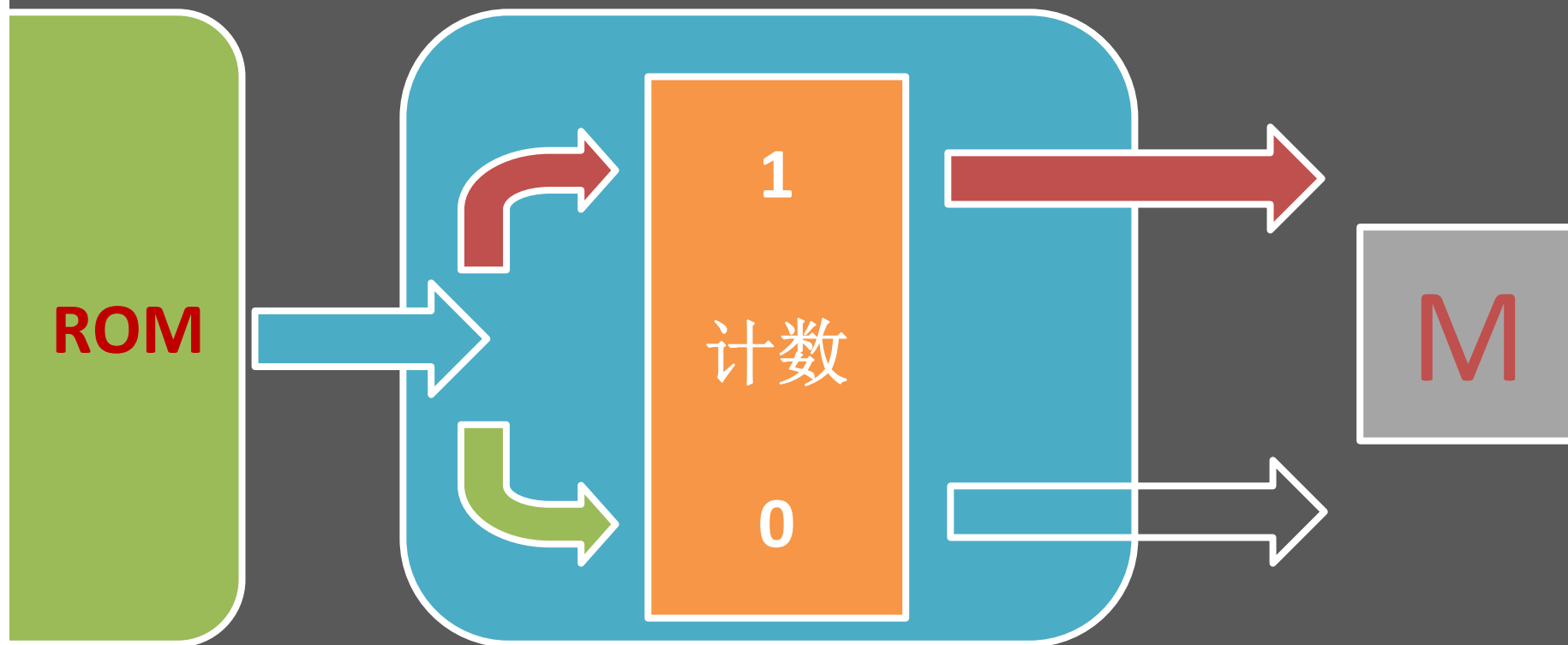
CNT2: 1



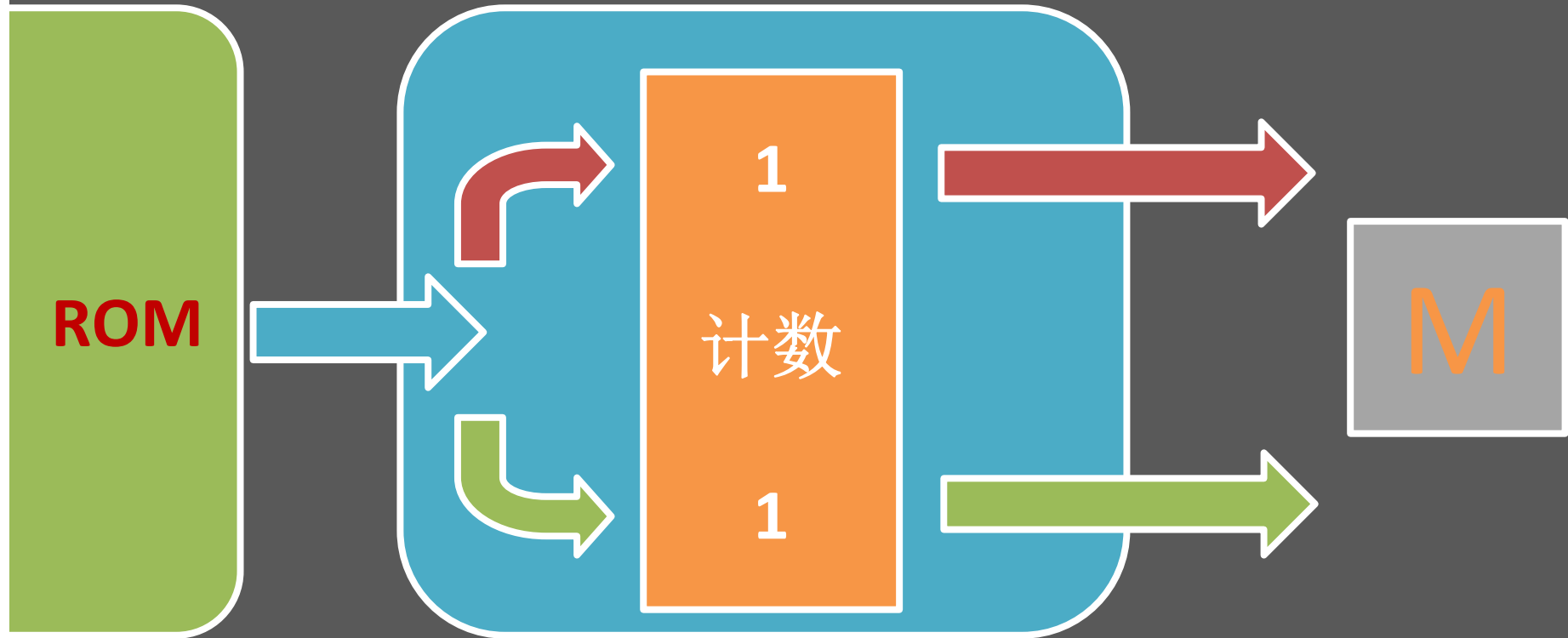
三色显示原理



三色显示原理



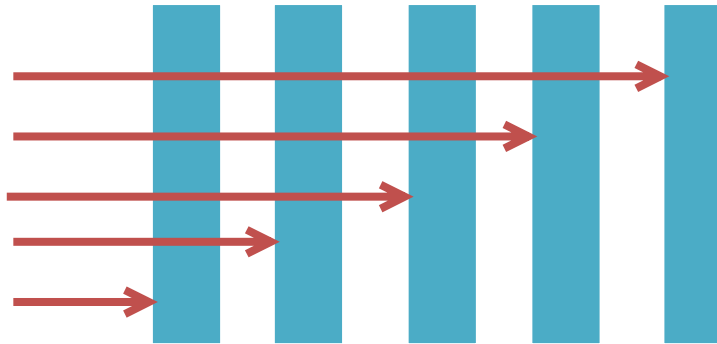
三色显示原理



矩阵的转换

```
reg [4:0] Matrix [4:0];
```

```
Matrix[4] = 5'bxxxxxx;  
Matrix[3] = 5'bxxxxxx;  
Matrix[2] = 5'bxxxxxx;  
Matrix[1] = 5'bxxxxxx;  
Matrix[0] = 5'bxxxxxx;
```



赋值:

```
begin  
Matrix[4] = 5'b00100;  
Matrix[3] = 5'b11000;  
Matrix[2] = 5'b01001;  
Matrix[1] = 5'b11101;  
Matrix[0] = 5'b10110;  
end
```

判断:

```
if(  
Matrix[4] == 0 &&  
Matrix[3] == 0 &&  
Matrix[2] == 0 &&  
Matrix[1] == 0 &&  
Matrix[0] == 0  
)  
... ..
```

矩阵的转换

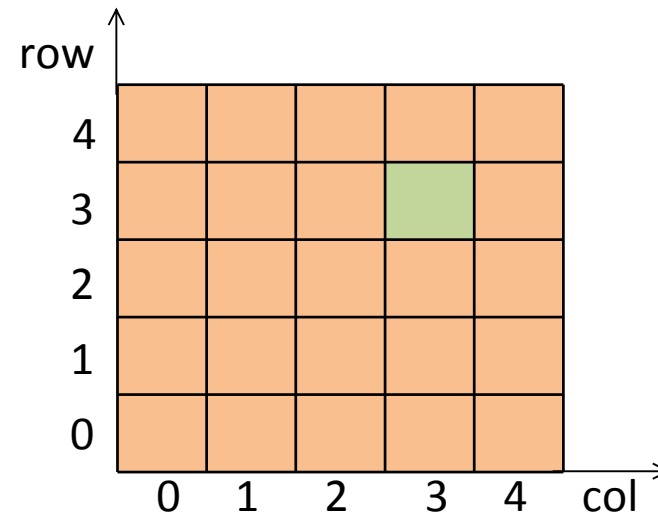
```
reg [24:0] Matrix2;
```

```
Matrix2 = 25'bxxxxx_xxxxx_xxxxx_xxxxx_xxxxx;
```



```
integer col , row;
```

```
Matrix2[5*col+row]
```



一码二用

```
always@(posedge clk)
if(... ..)
case(key)
up:  if( currow !=0 && (red[ curcol*5+currow-1]|green[curcol*5+currow-1])==0)
      nextrow=currow-1;
down:if(currow!=3'b100 && (red[curcol*5+currow+1]|green[curcol*5+currow+1])==0)
      nextrow=currow+1;
left:  if(curcol!=0 && (red[(curcol-1)*5+currow]|green[(curcol-1)*5+currow])==0)
      nextcol=curcol-1;
right: if(curcol!=3'b100 && (red[(curcol+1)*5+currow]|green[(curcol+1)*5+currow])==0)
      nextcol=curcol+1;
default: begin nextrow=currow;nextcol=curcol; end
endcase
```

设置障碍、设置起点：按一下方向键移动一格



游戏：按一下方向键一直移动，直至无法移动



一码二用

实现此效果的原理：合适的触发条件

键盘特性：

键盘按下时，输出4位码电平 与 keydown脉冲

时钟上升沿：

- 设置障碍状态 && keydown 高电平
- 设置起点状态 && keydown 高电平
- 游戏状态



调用移动模块

