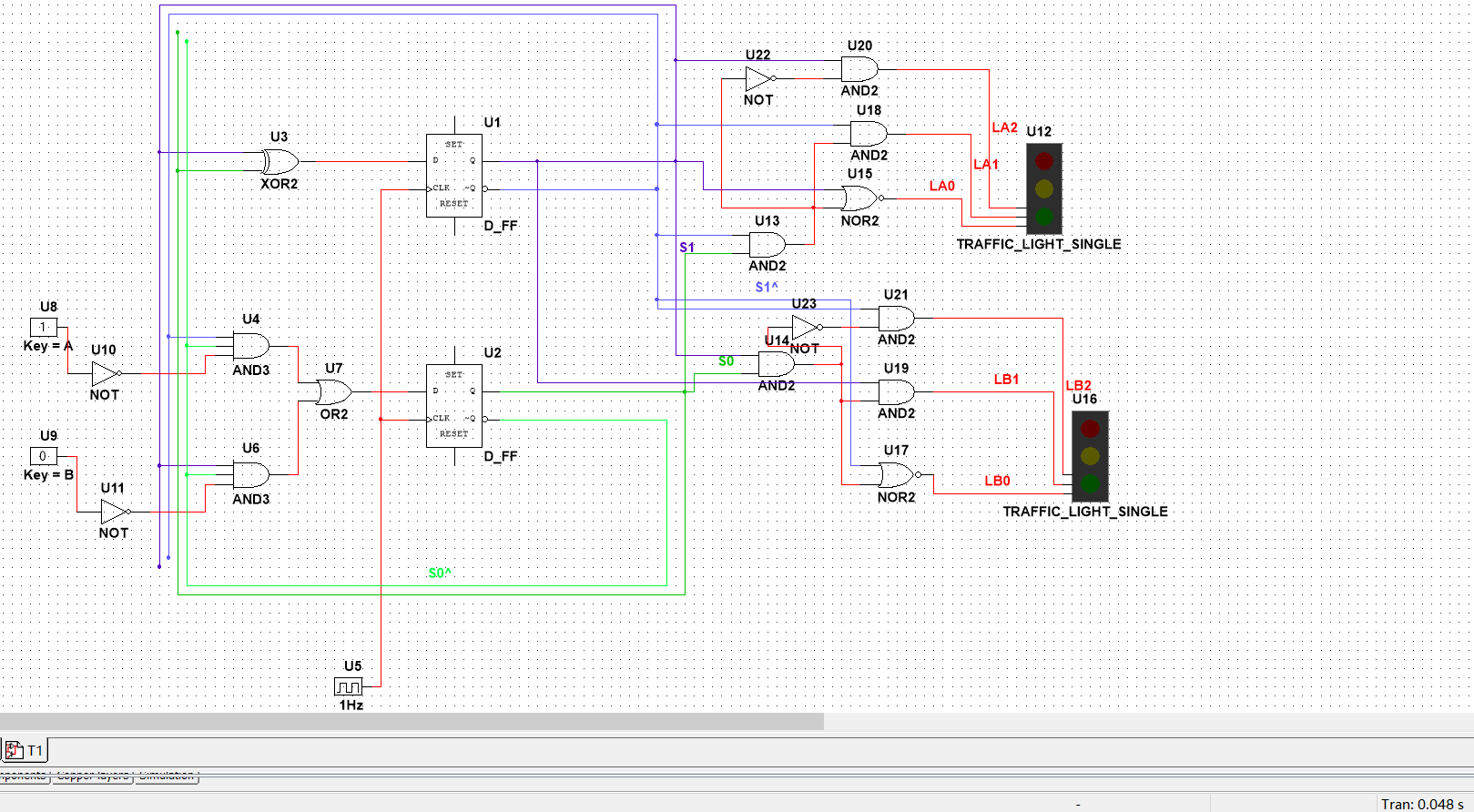
***1.FSM I***

Question 1



Let the frequency of the digital\_clock be 1Hz.

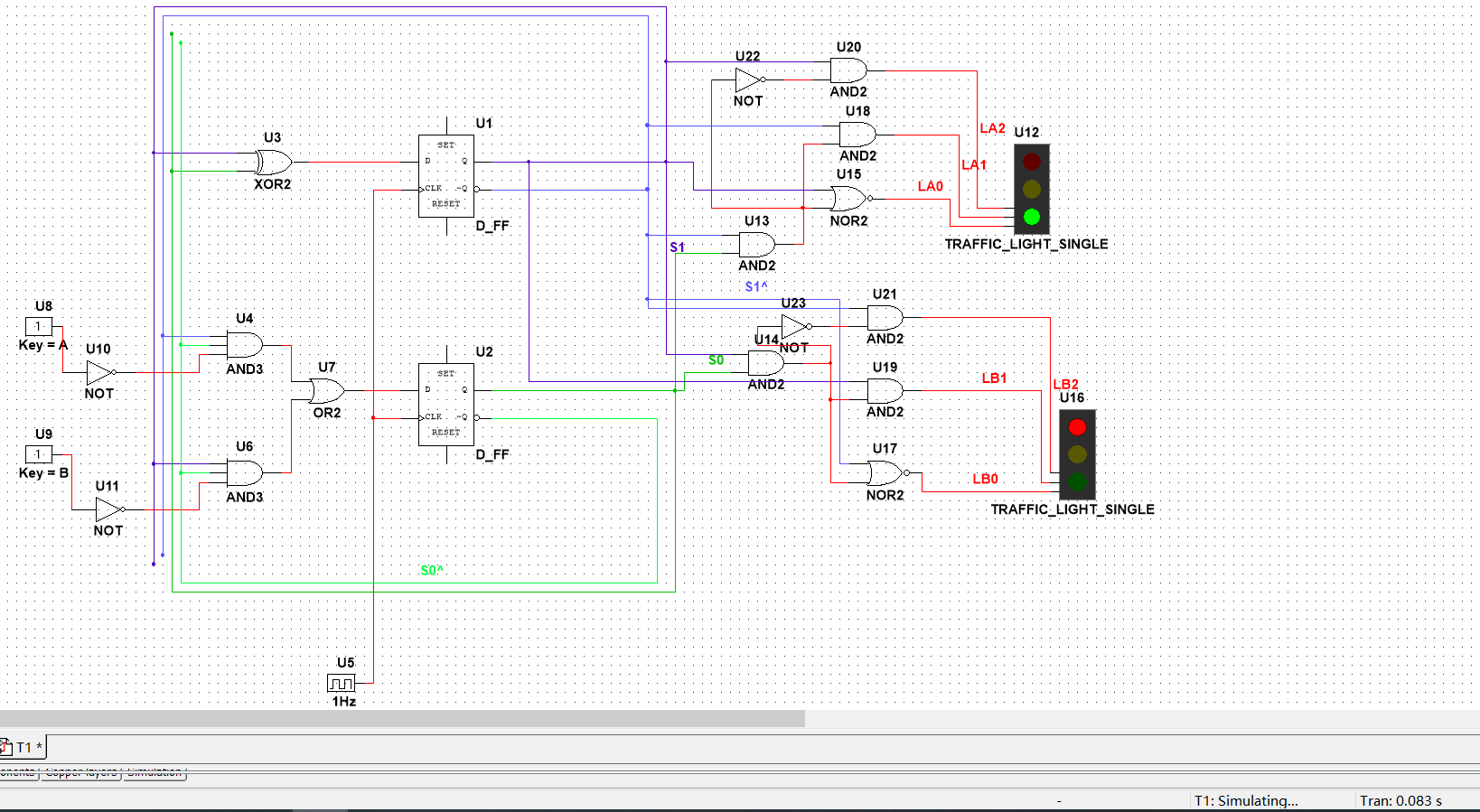
So the clock signal will change in every 1 second.

And it will follow as a loop.

In one cycle:

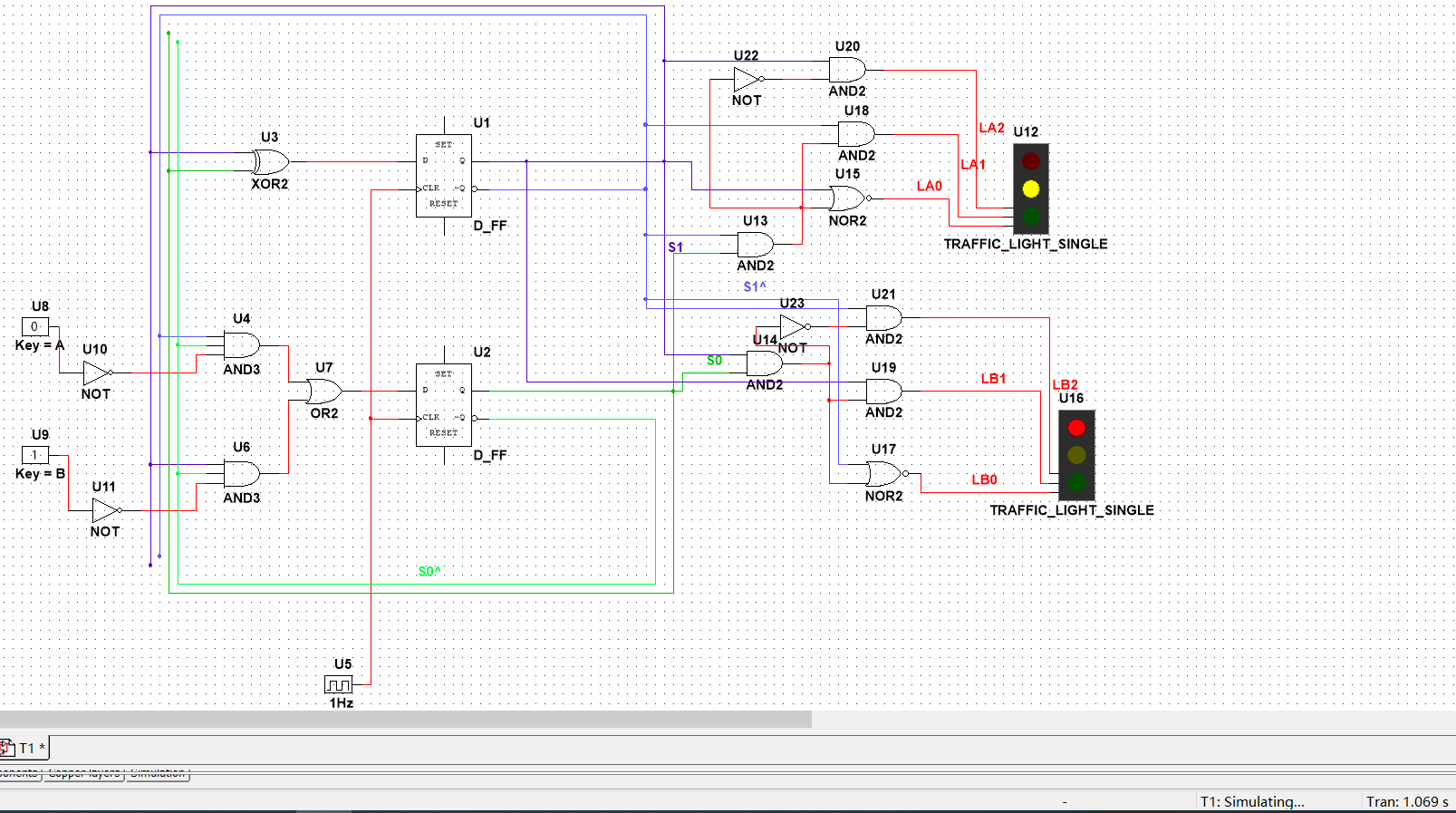
A=1,B=1

time=0s



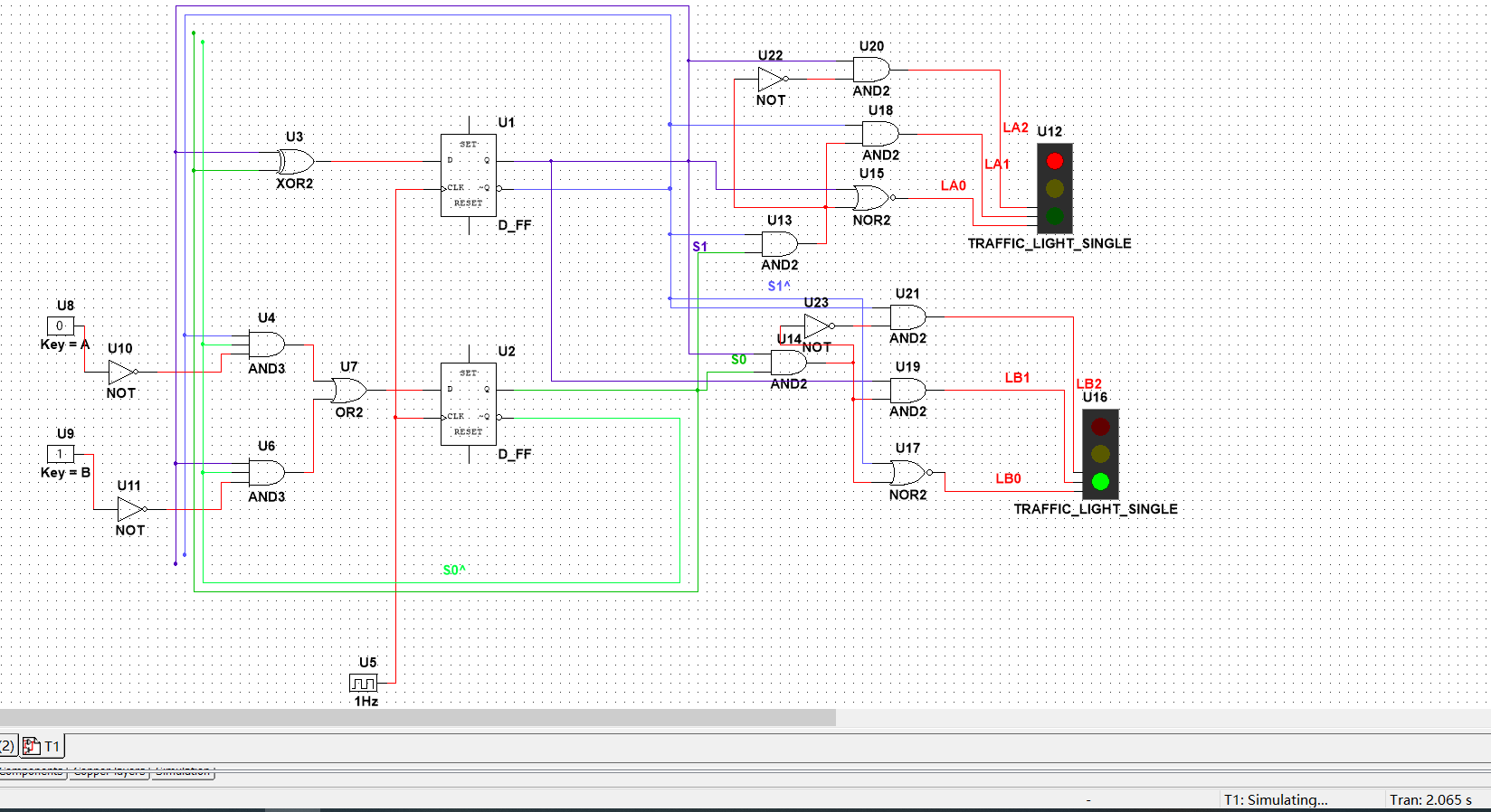
A=0 B=1

time=1s



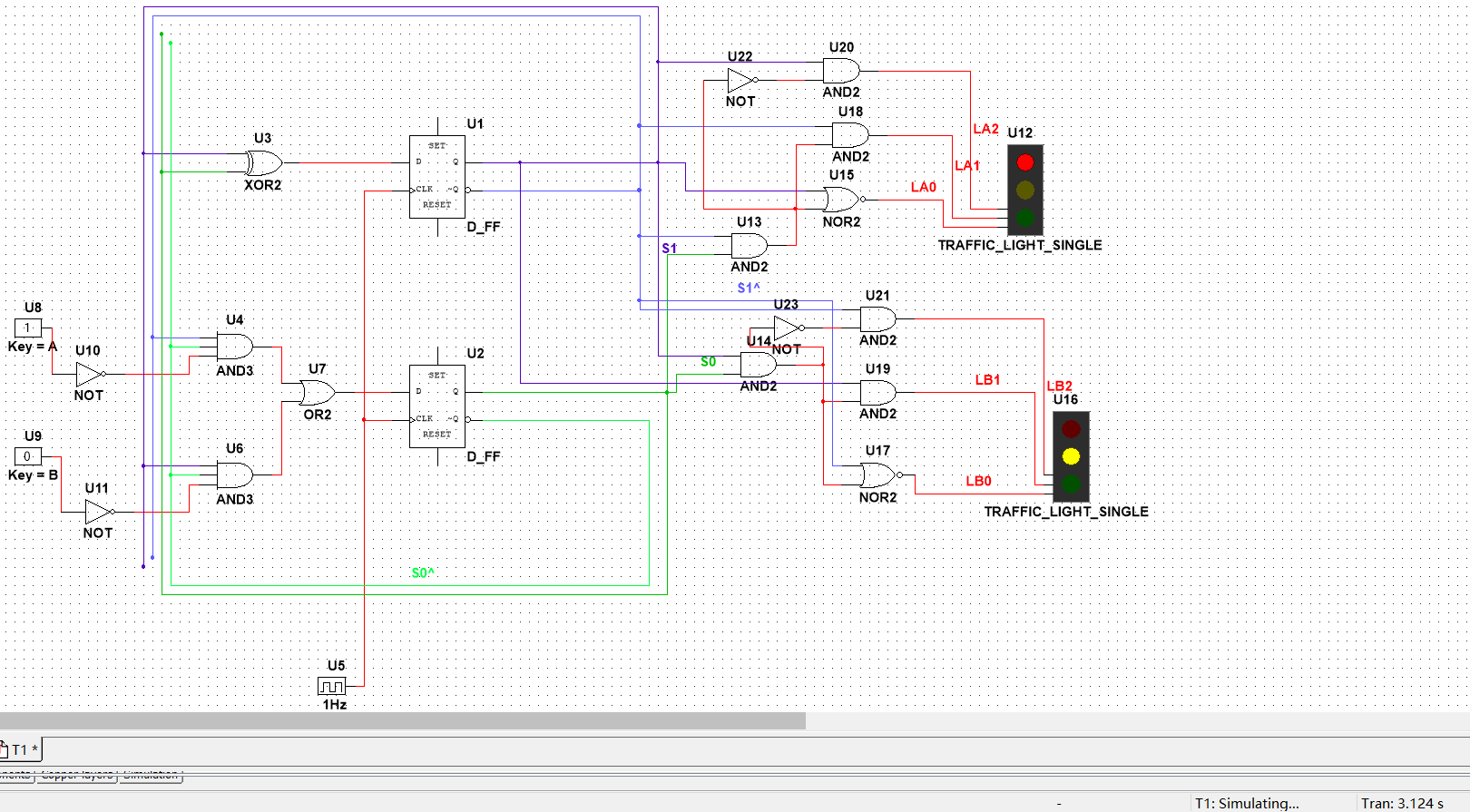
A=0 B=1

time=2s

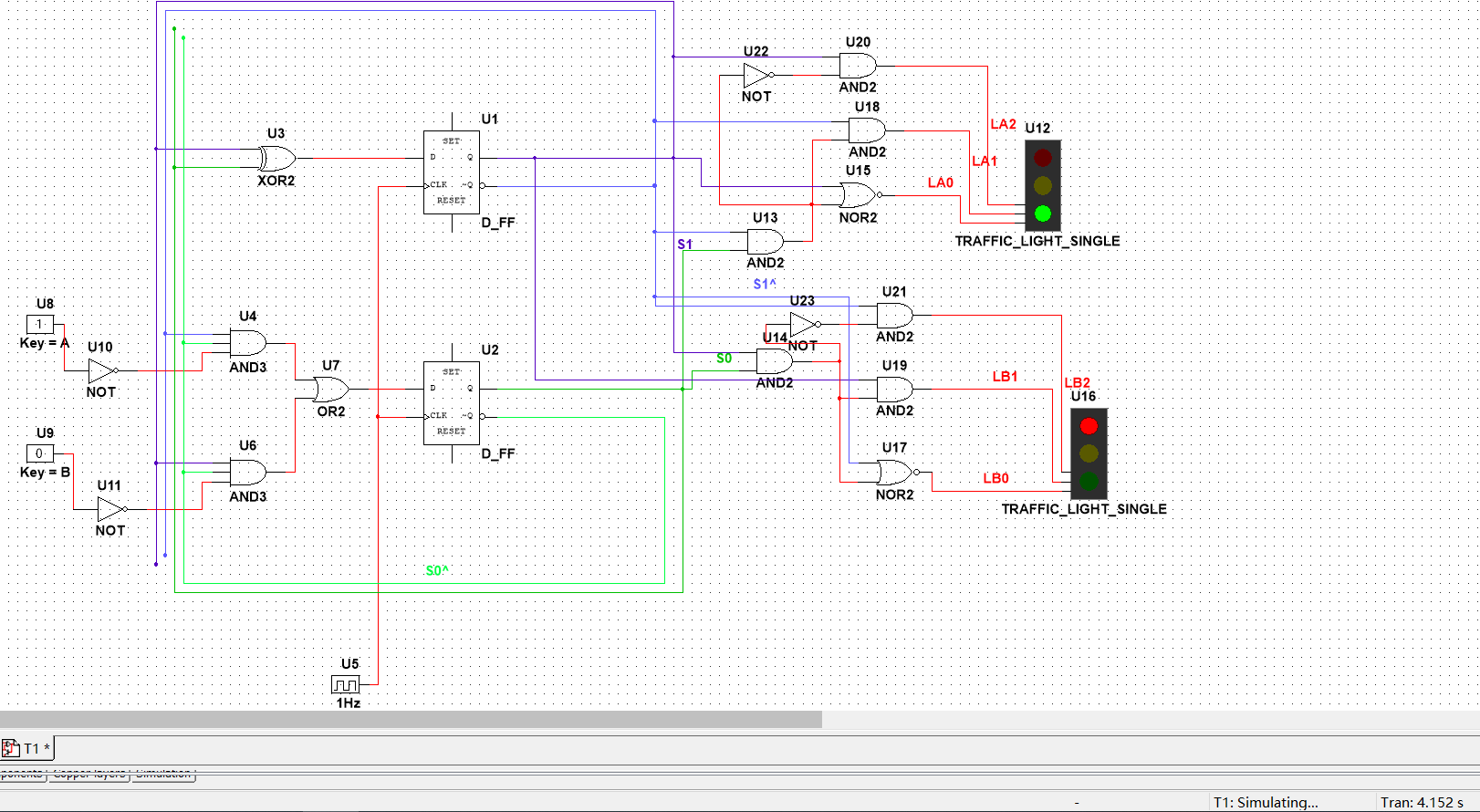


A=1 B=0

time=3s



time=4s

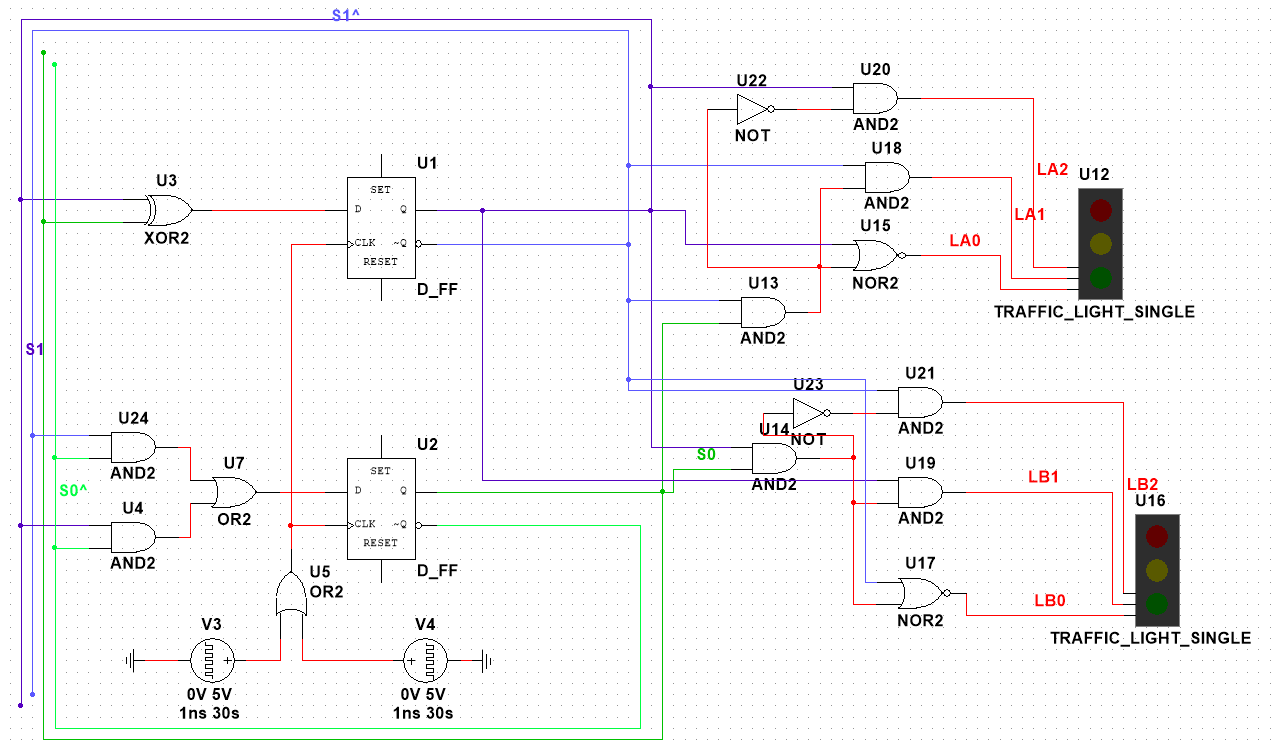


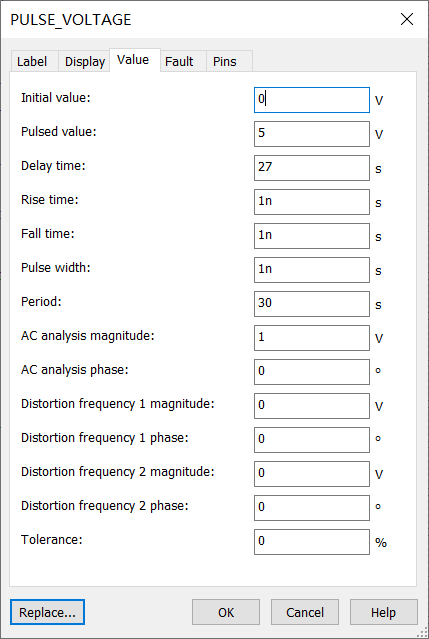
Question 2

Use pulse\_voltage to change the signal in 1ns when the time is 27s and 30s in one loop.

The loop is 30s.

And the whole cycle is 60s for one loop.





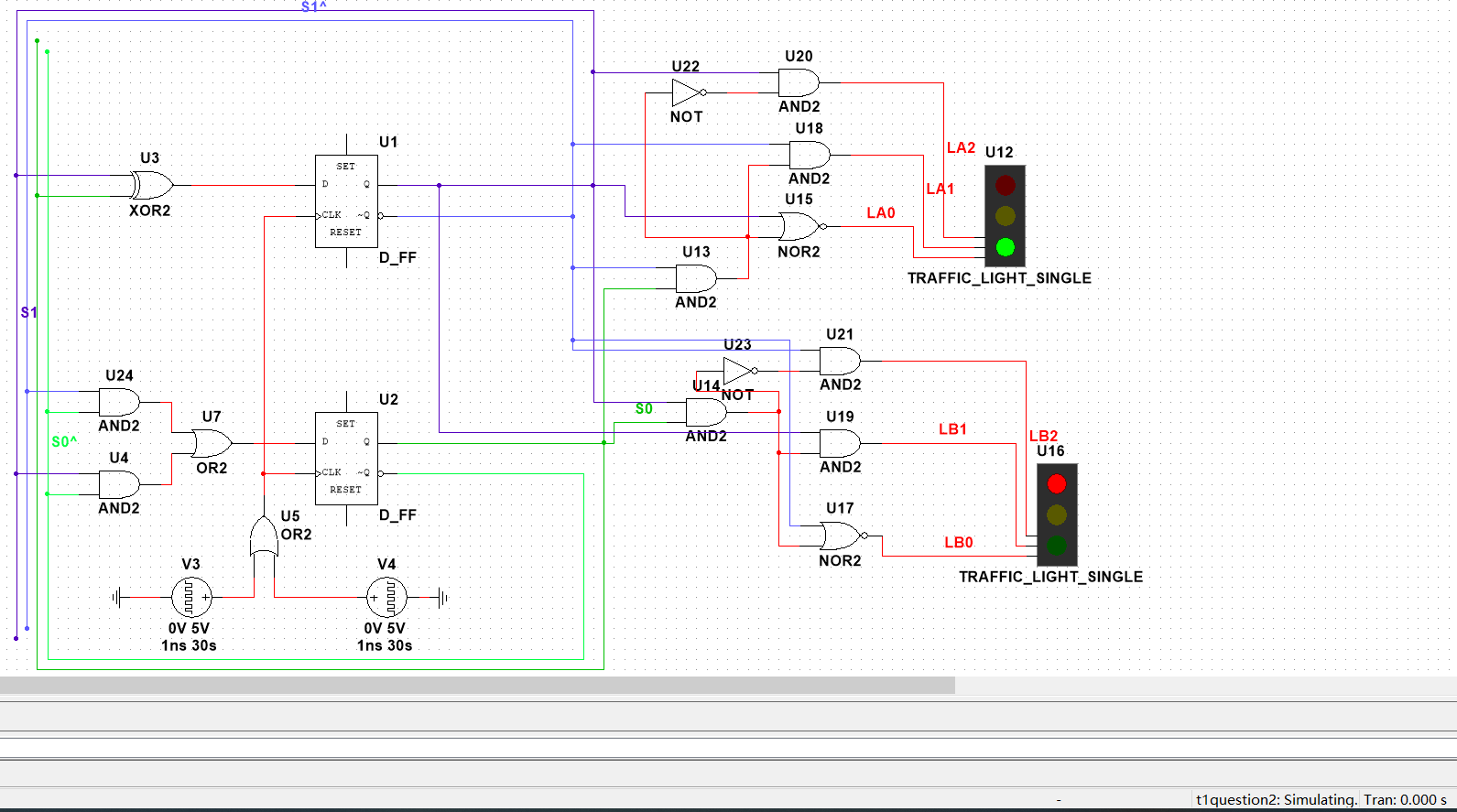
One of the PULSE\_VOLTAGE.

The initial value is 0V, and Pulsed value is 5V.

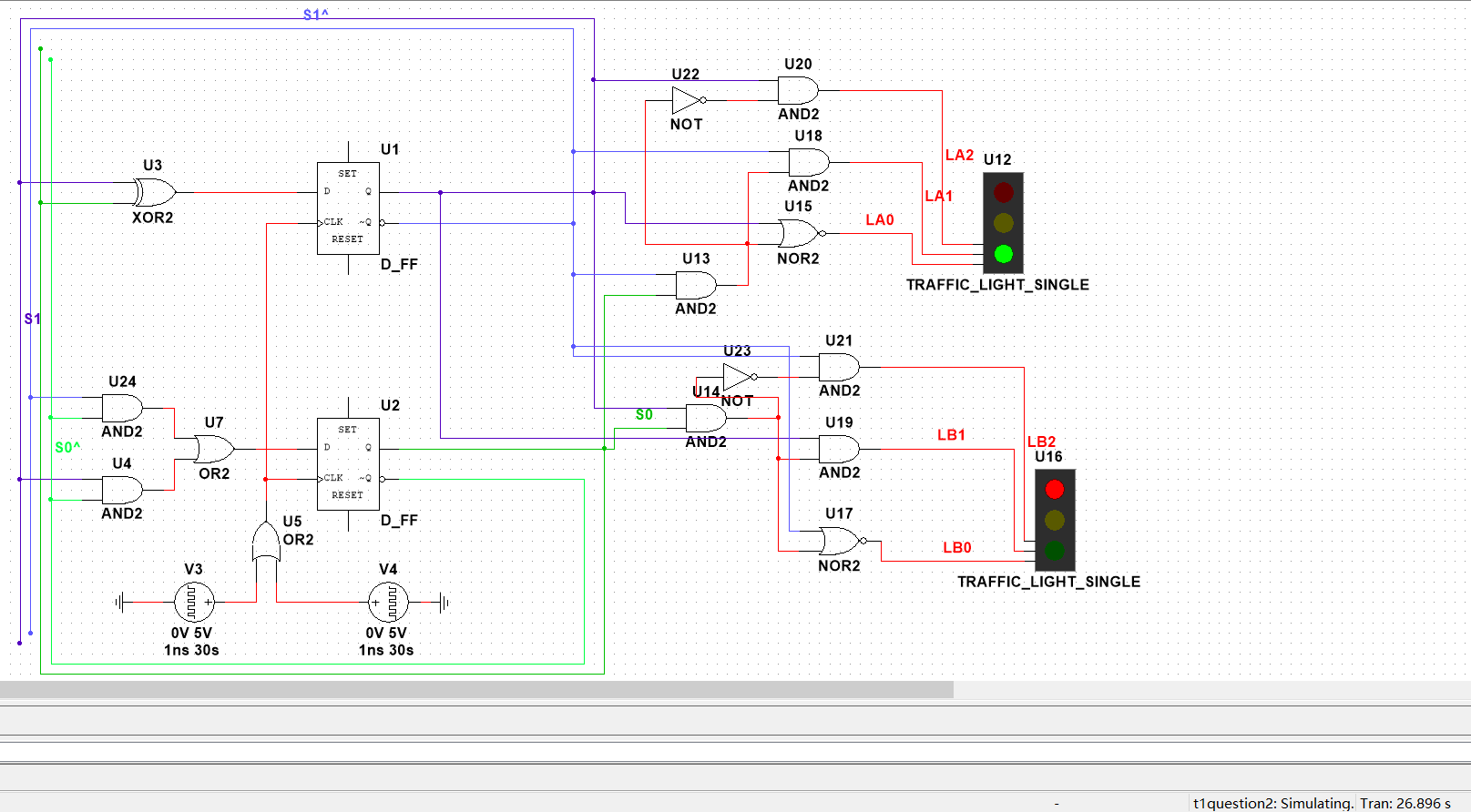
Let the signal delay 27s, change in 1ns, and the period is 30s.

In one 60s’ cycle:

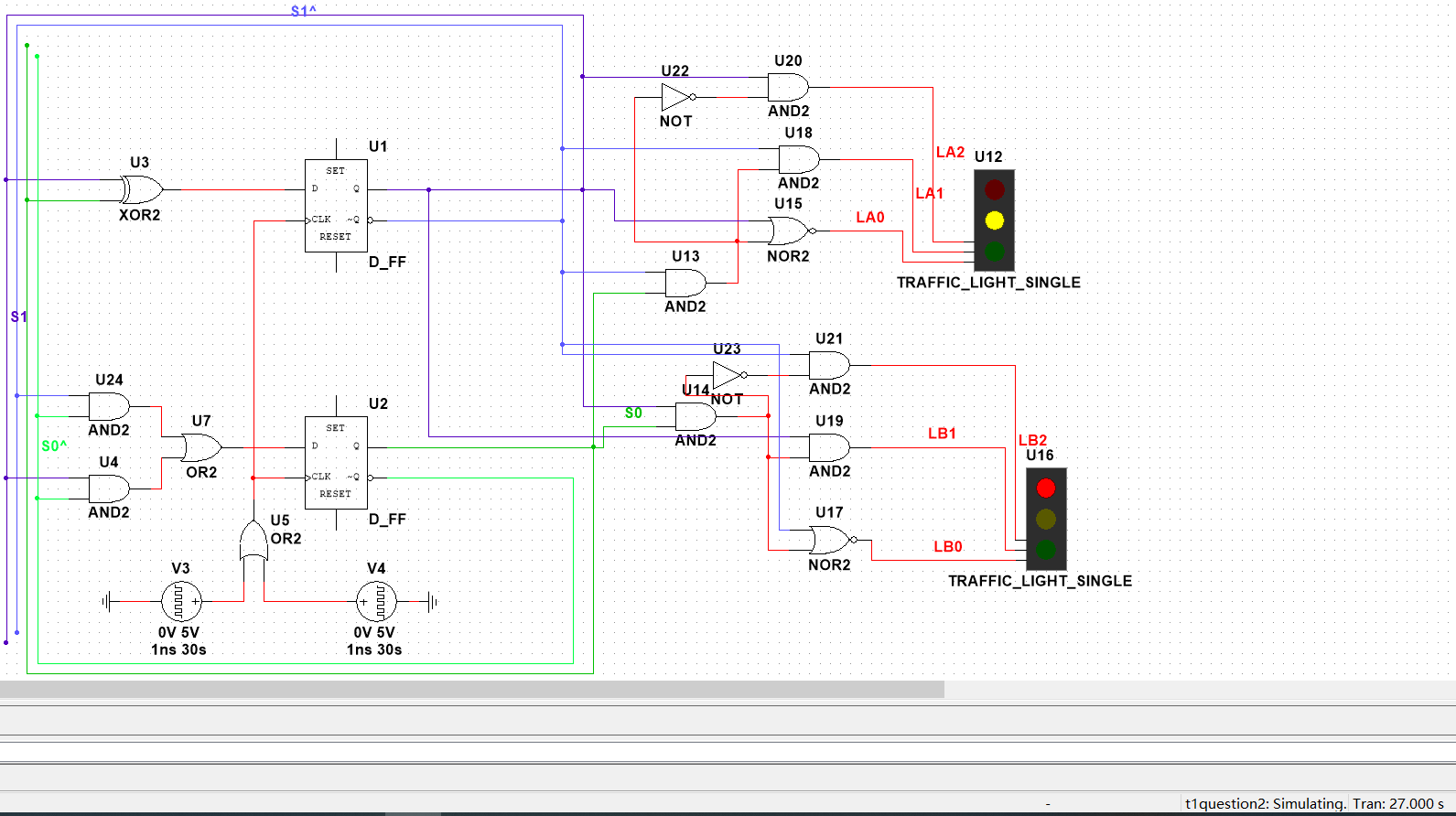
The L1’s green light and L2’s red light light up at the beginning.



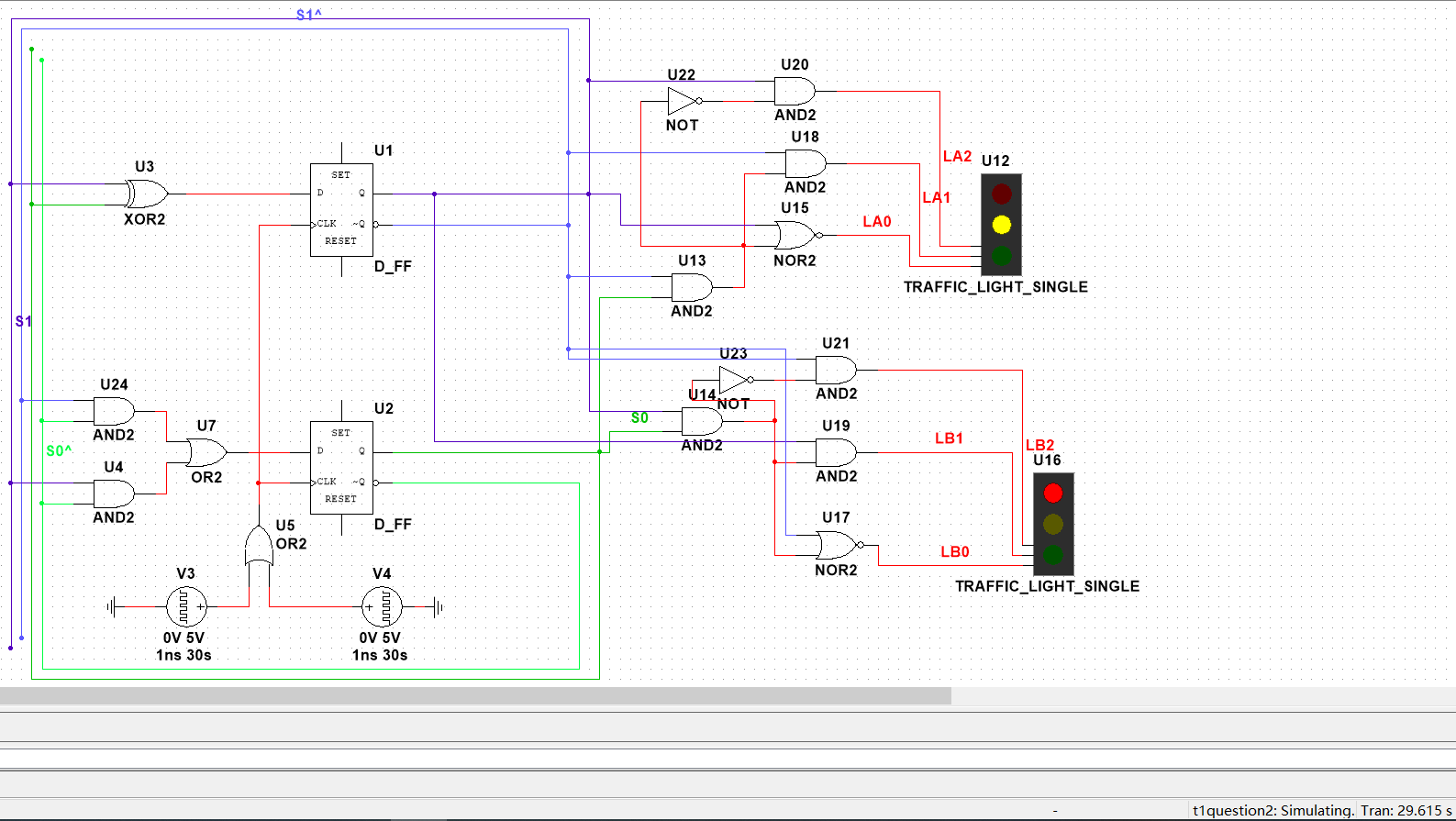
At 27s, the L1’s green light will extinguish immediately after this.



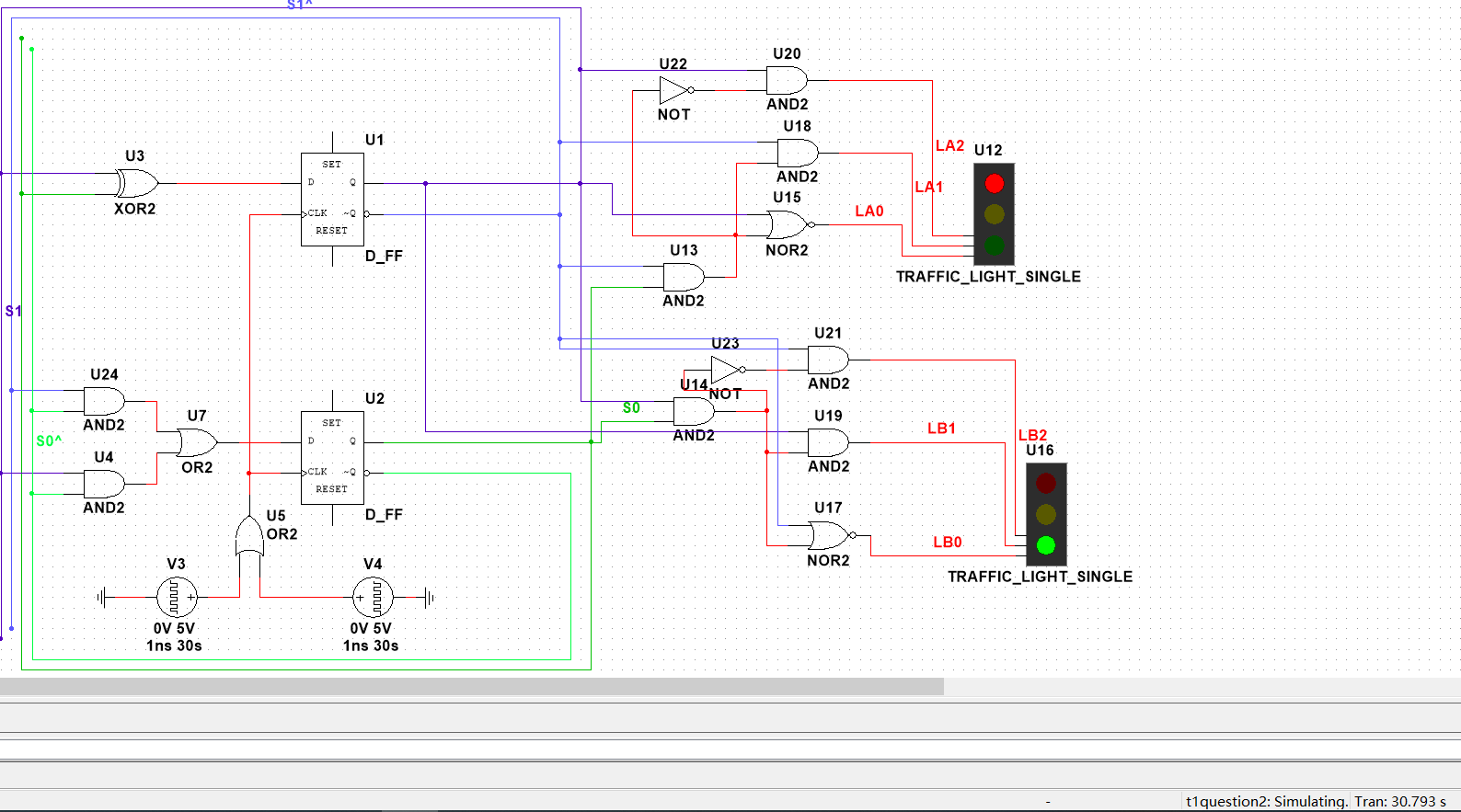
And the L1’s yellow light lights up following closely.



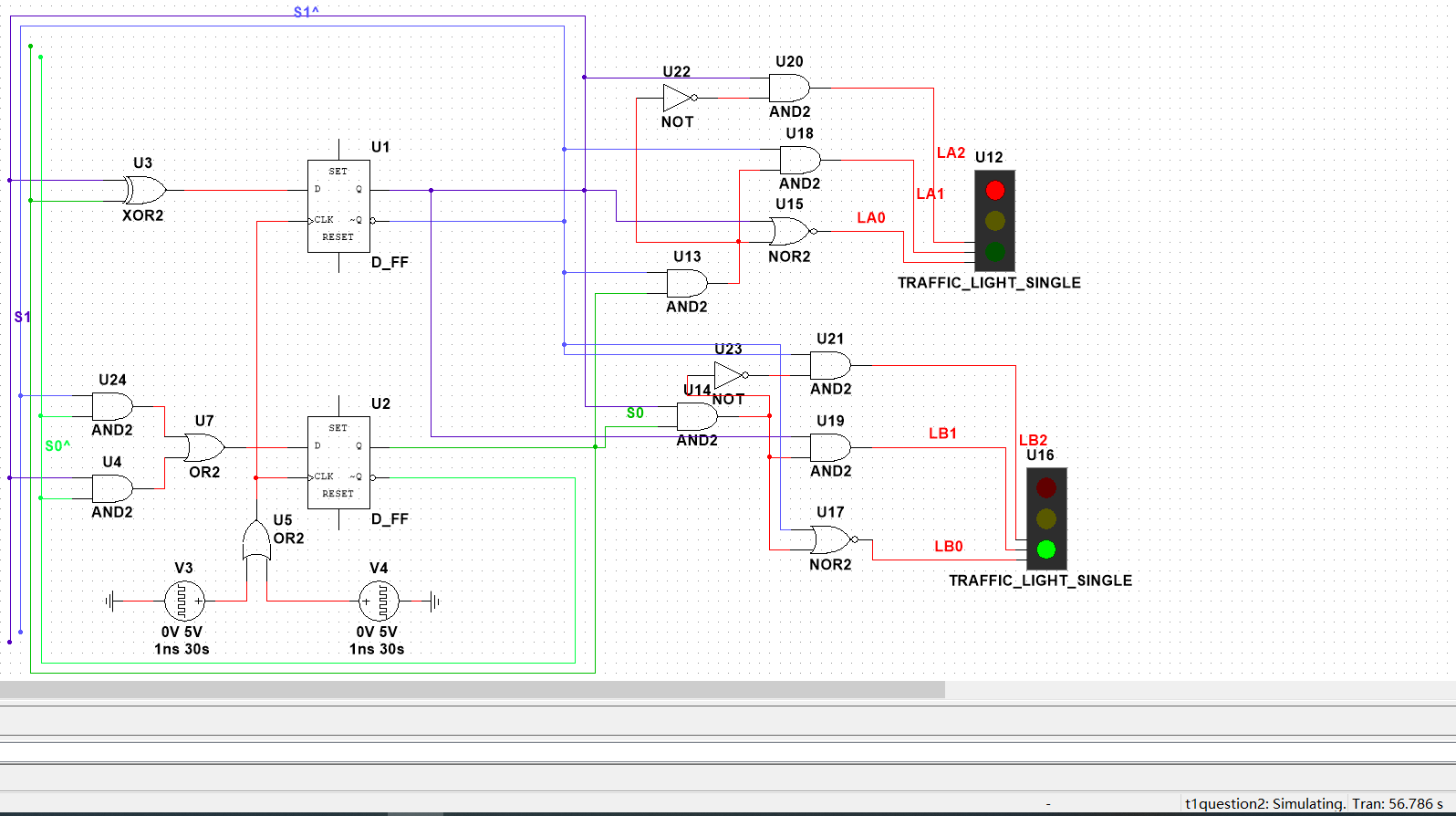
At 30s, the L1’s yellow light and L2’s red light will extinguish immediately after this.



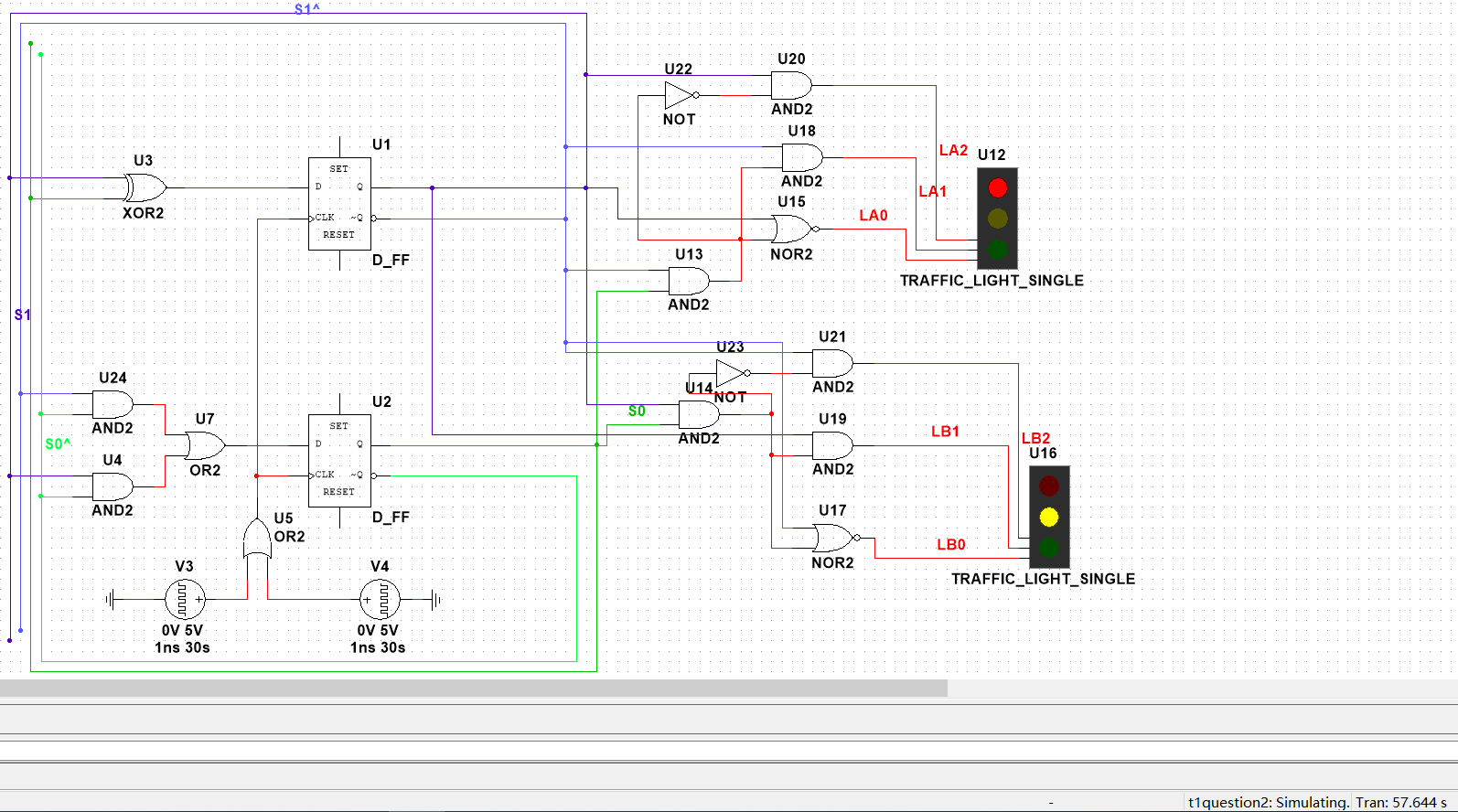
And the L1’s red light and L2’s green light light up following closely.



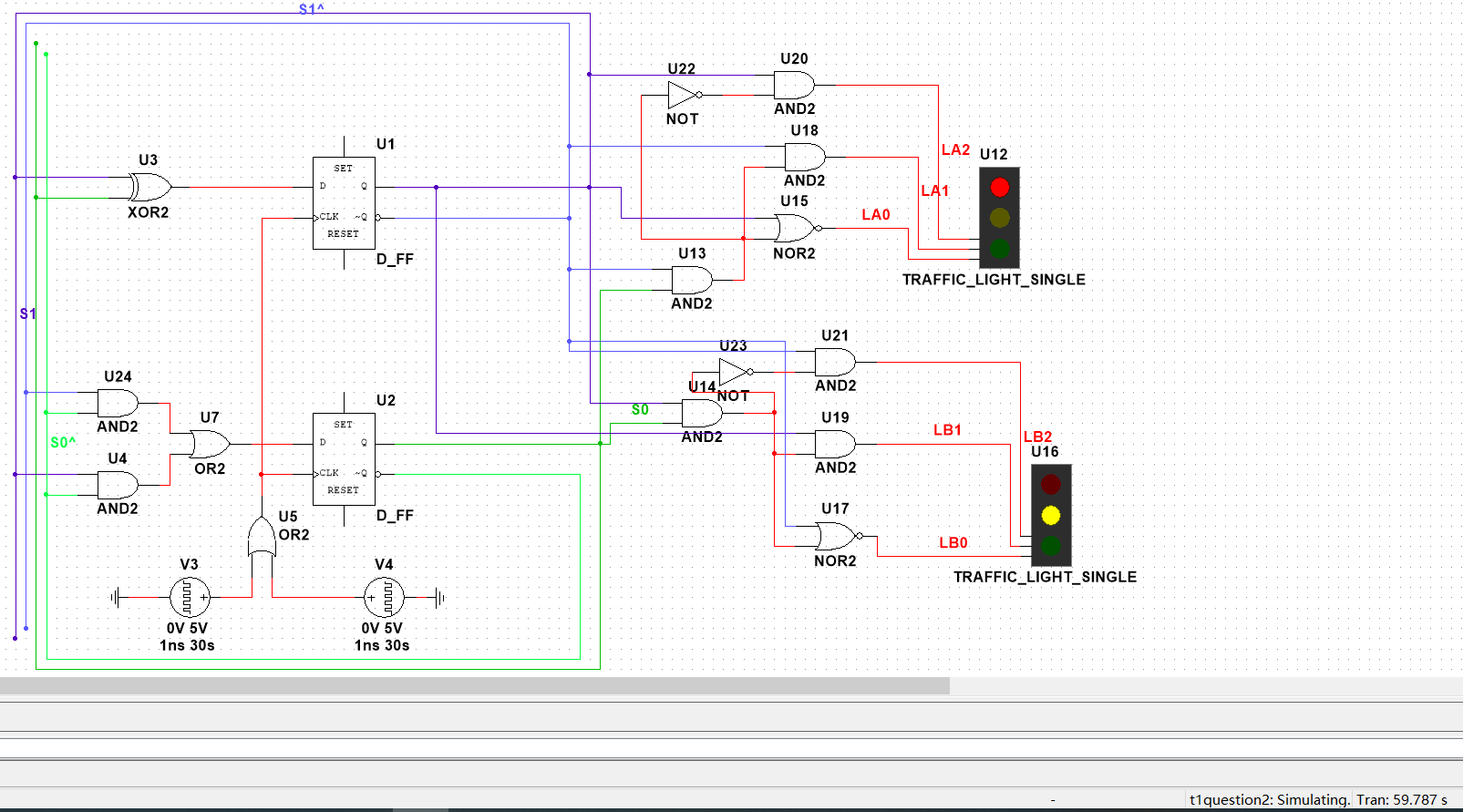
At 57s, the L2’s green light will extinguish immediately after this.



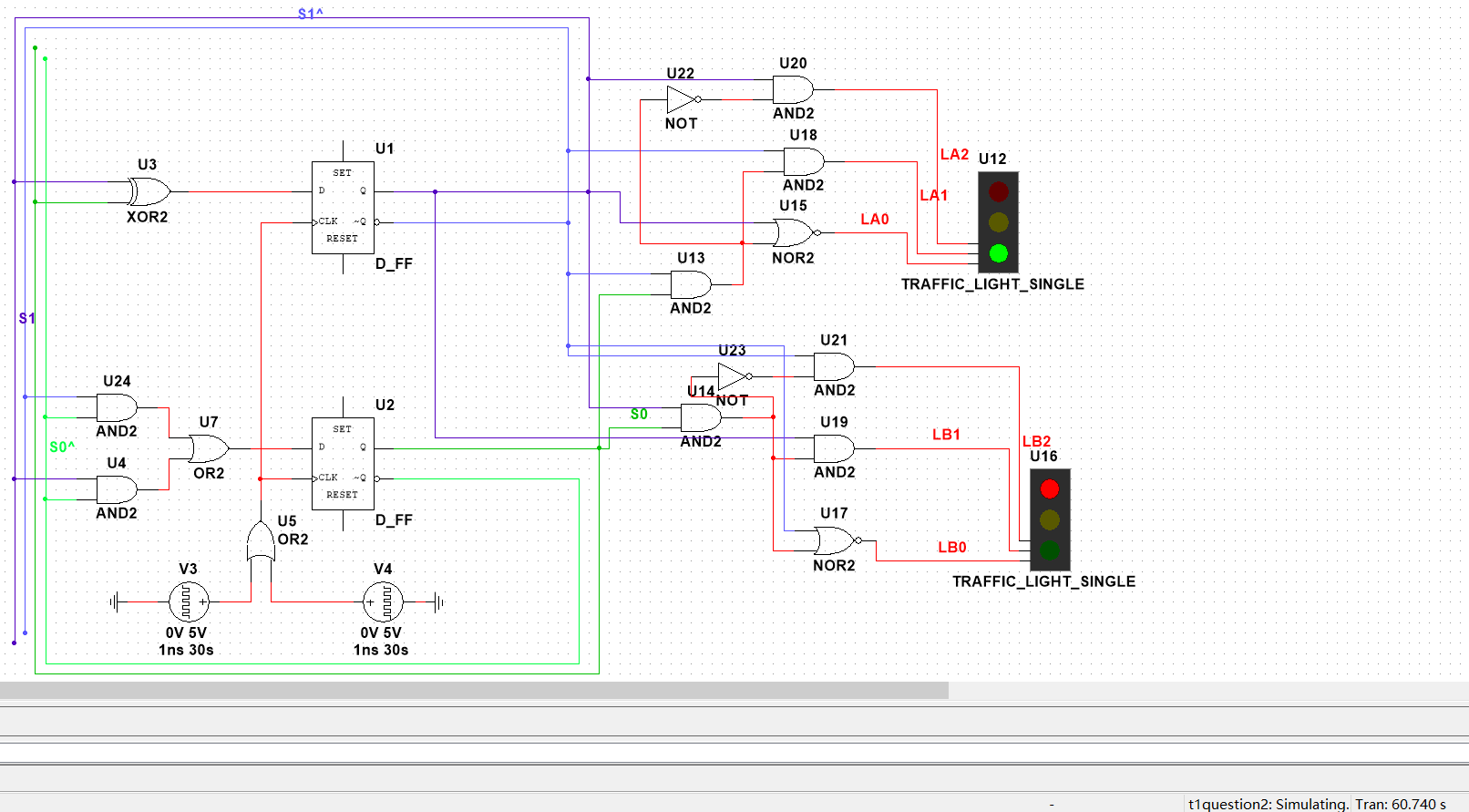
And the L2’s yellow light lights up following closely.



At 60s, the L1’s red light and L2’s yellow light will extinguish immediately after this.



And the L1’s green light and L2’s red light light up following closely.



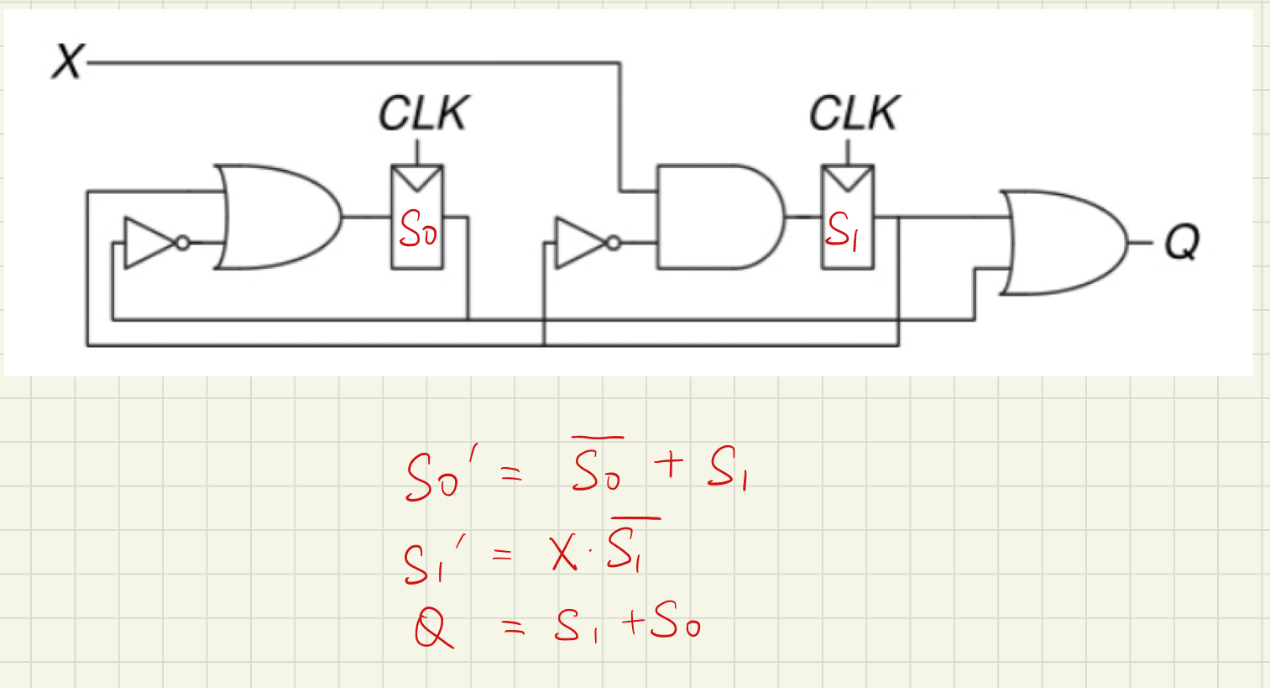
Then a now cycle starts, and the lights will follow the steps like the first cycle.

***2.FSM II***

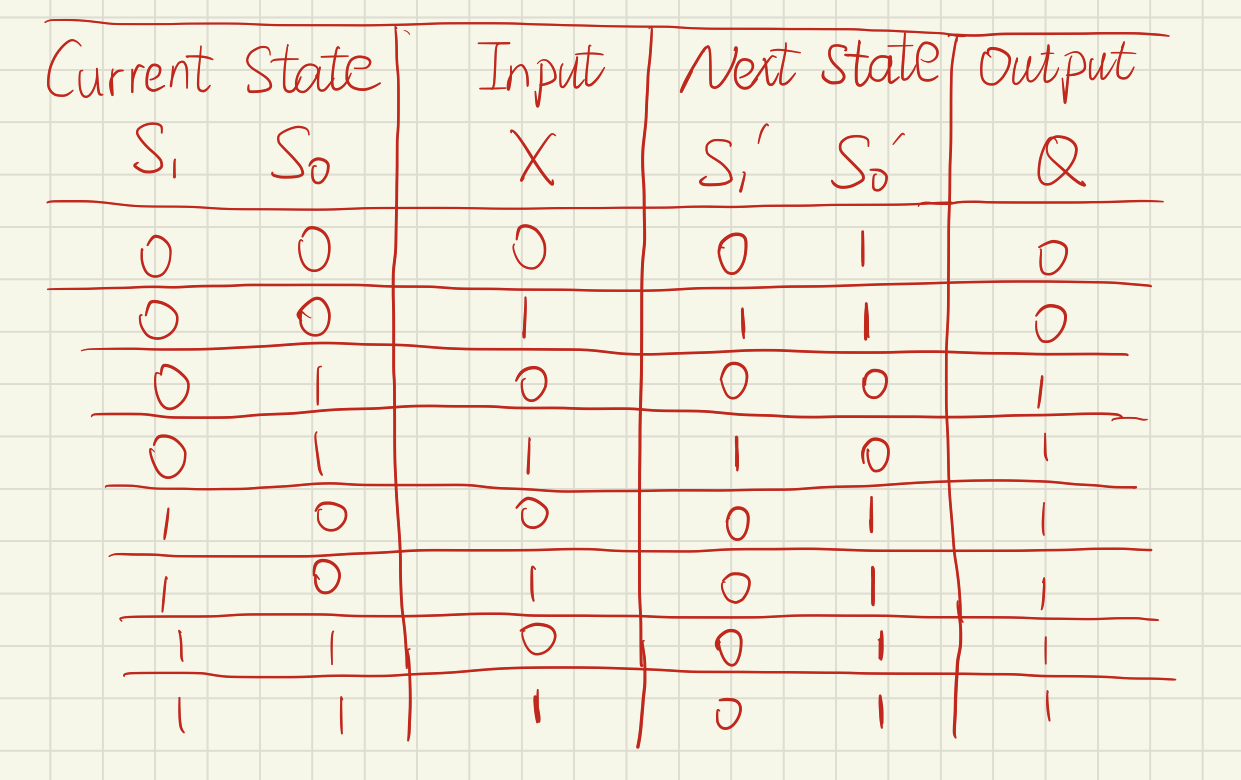
The FSM is as followed.

Let the left D\_FF be S0, the right D\_FF be S1.

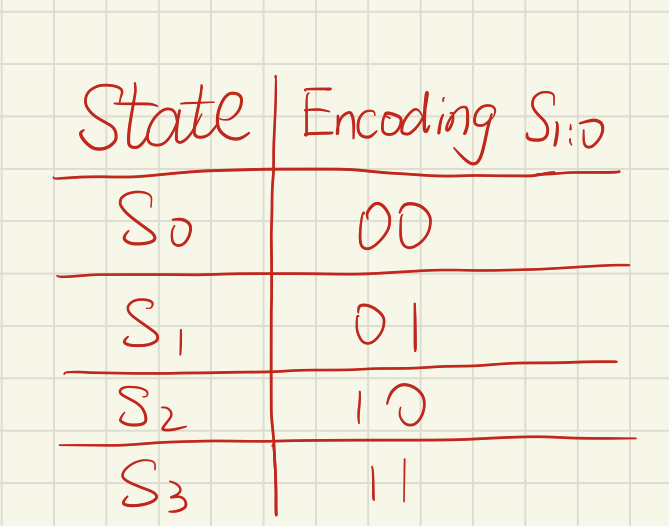
The S0,S1,Q’s transform are as followed.

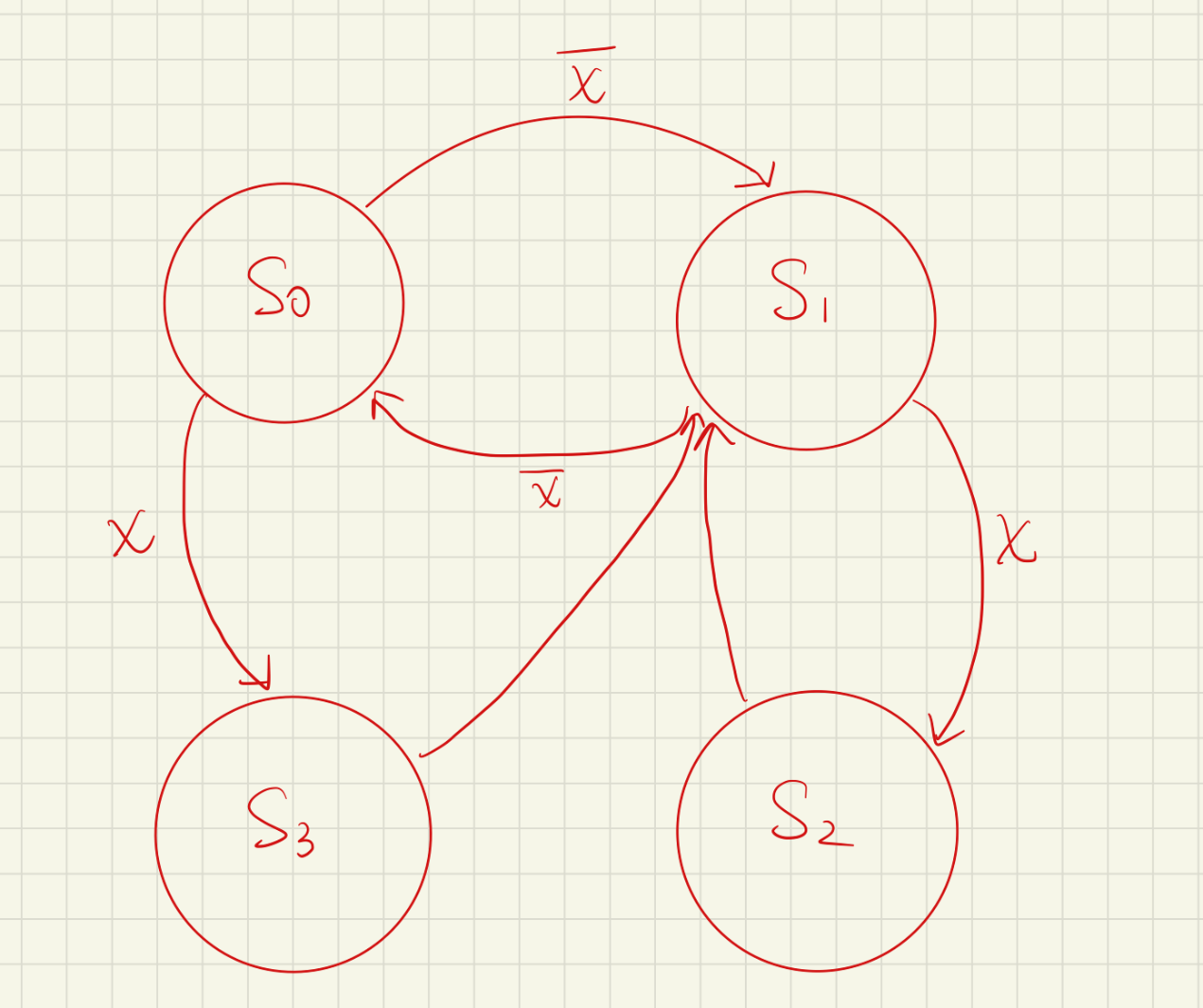


The state table

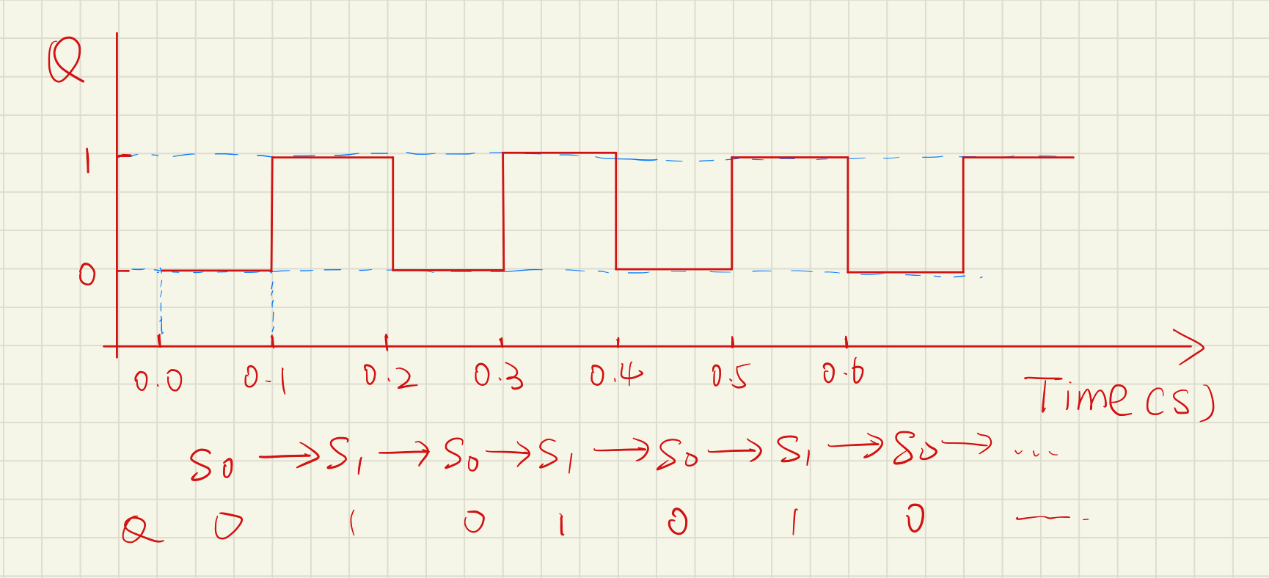


The output encoding

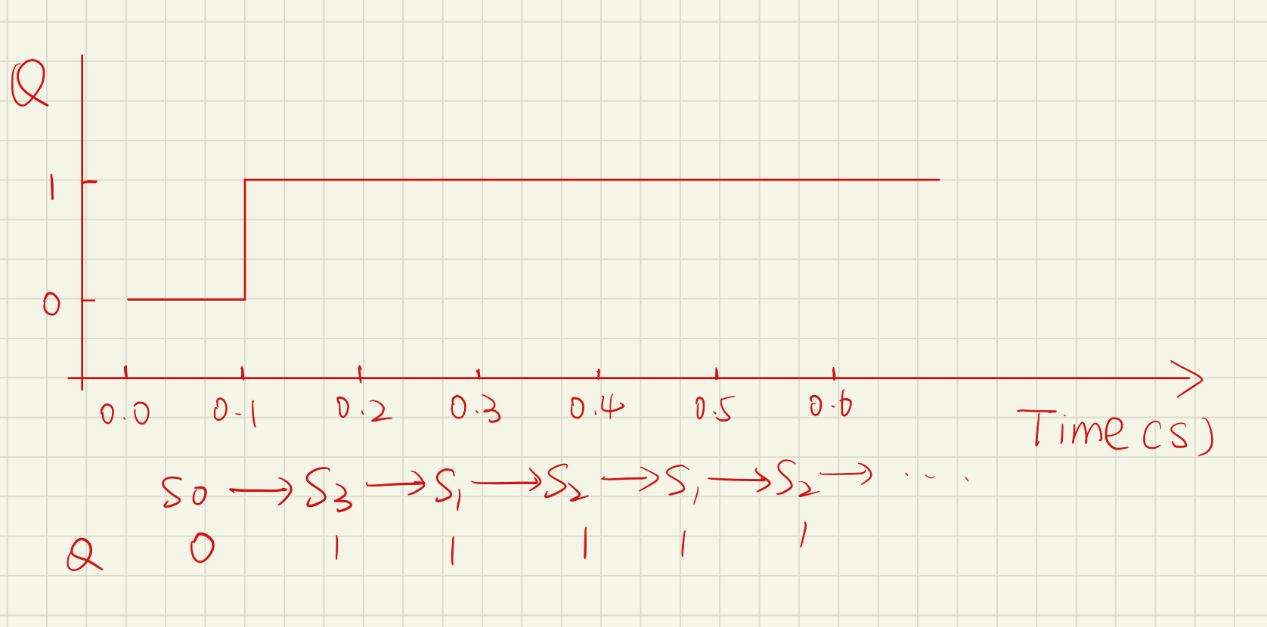


The state transition 

The output waveform when X=0



The output waveform when X=1

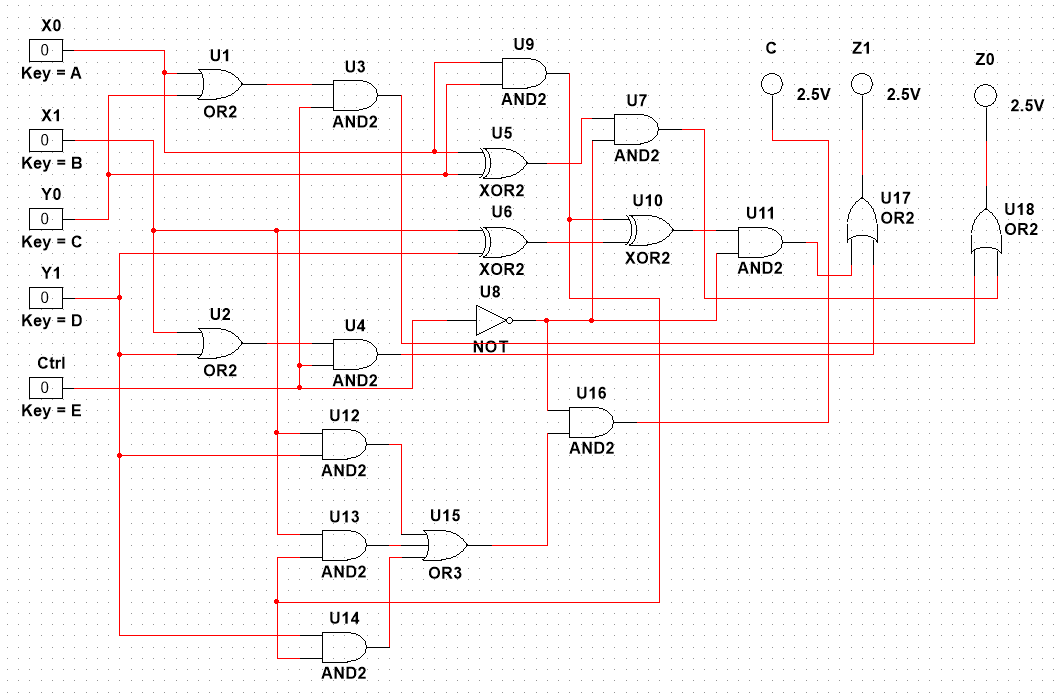


***3.2 bits easy ALU***

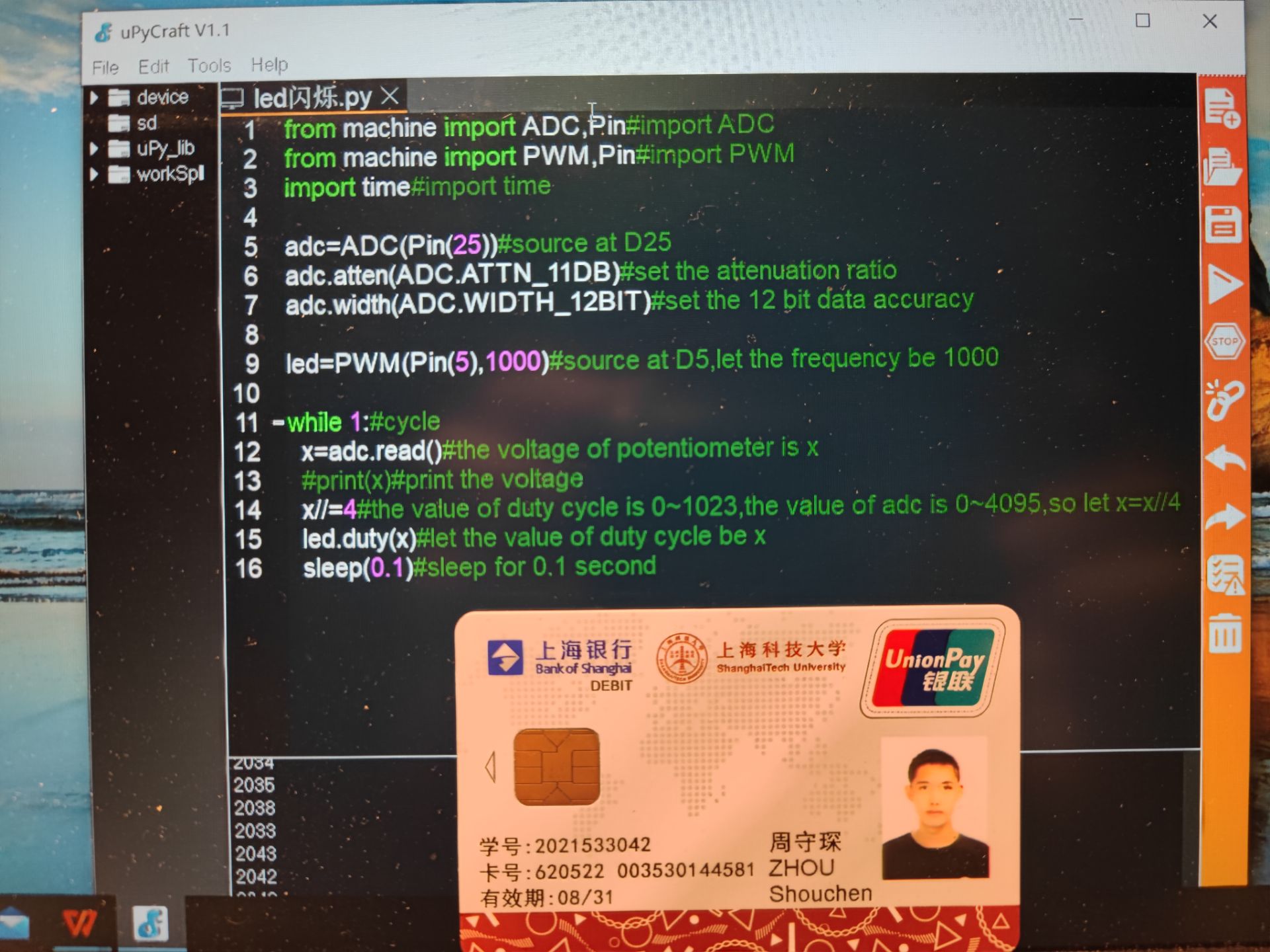
If the red(left) led is bright, then it means c=1,otherwise c=0

If the yellow(middle) led is bright, then it means z1=1,otherwise z1=0

If the green(right) led is bright, then it means z0=1,otherwise z0=0



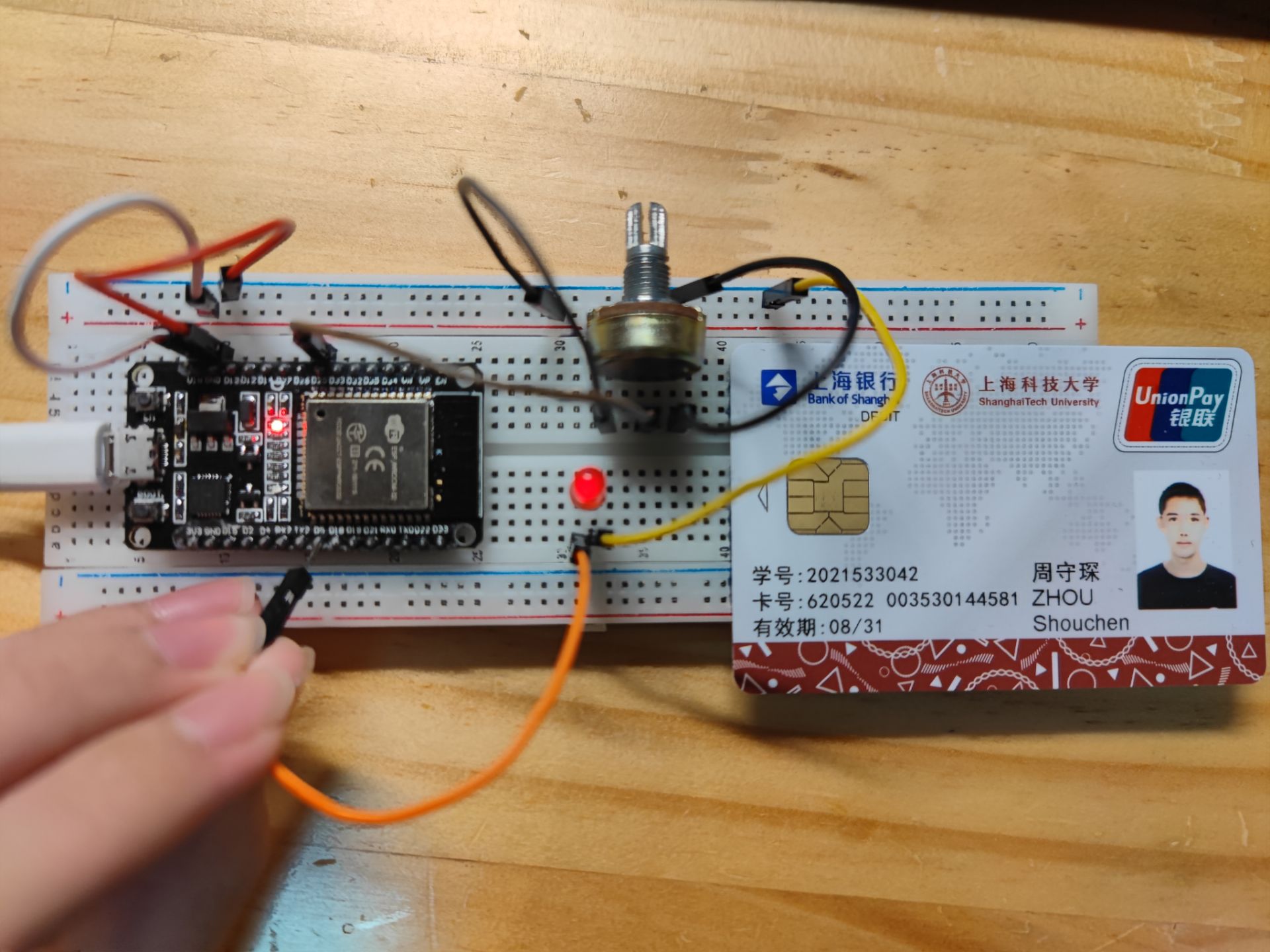
***4.MCU development***



from machine import ADC,Pin#import ADCfrom machine import PWM,Pin#import PWMimport time#import timeadc=ADC(Pin(25))#source at D25adc.atten(ADC.ATTN\_11DB)#set the attenuation ratioadc.width(ADC.WIDTH\_12BIT)#set the 12 bit data accuracyled=PWM(Pin(5),1000)#source at D5,let the frequency be 1000while 1:#cycle x=adc.read()#the voltage of potentiometer is x print(x)#print the voltage x//=4#the value of duty cycle is 0~1023,the value of adc is 0~4095,so let x=x//4 led.duty(x)#let the value of duty cycle be x sleep(0.1)#sleep for 0.1 second

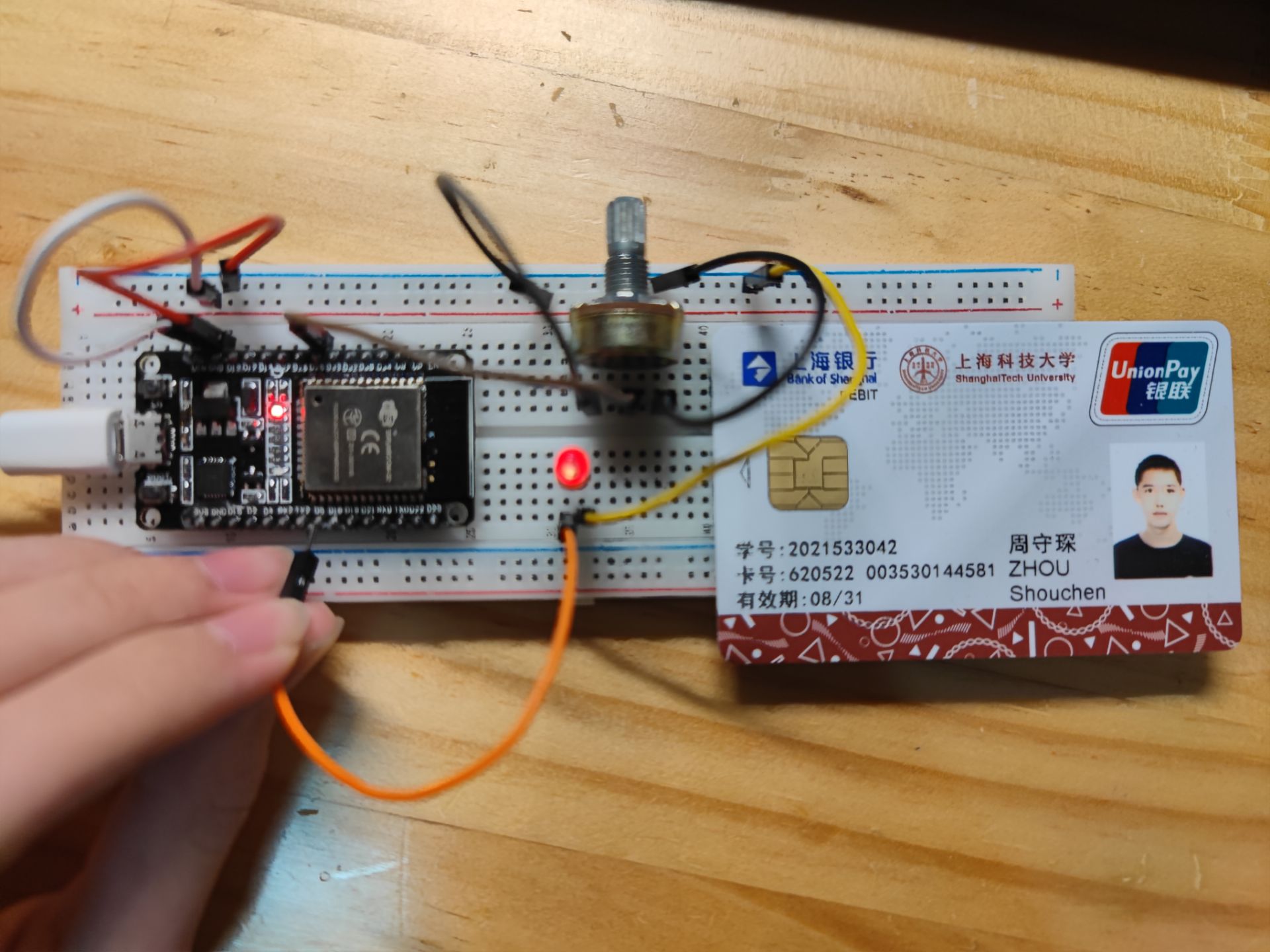
Adjust the potentiometer, let it get low voltage.

So brightness the of the led is low.

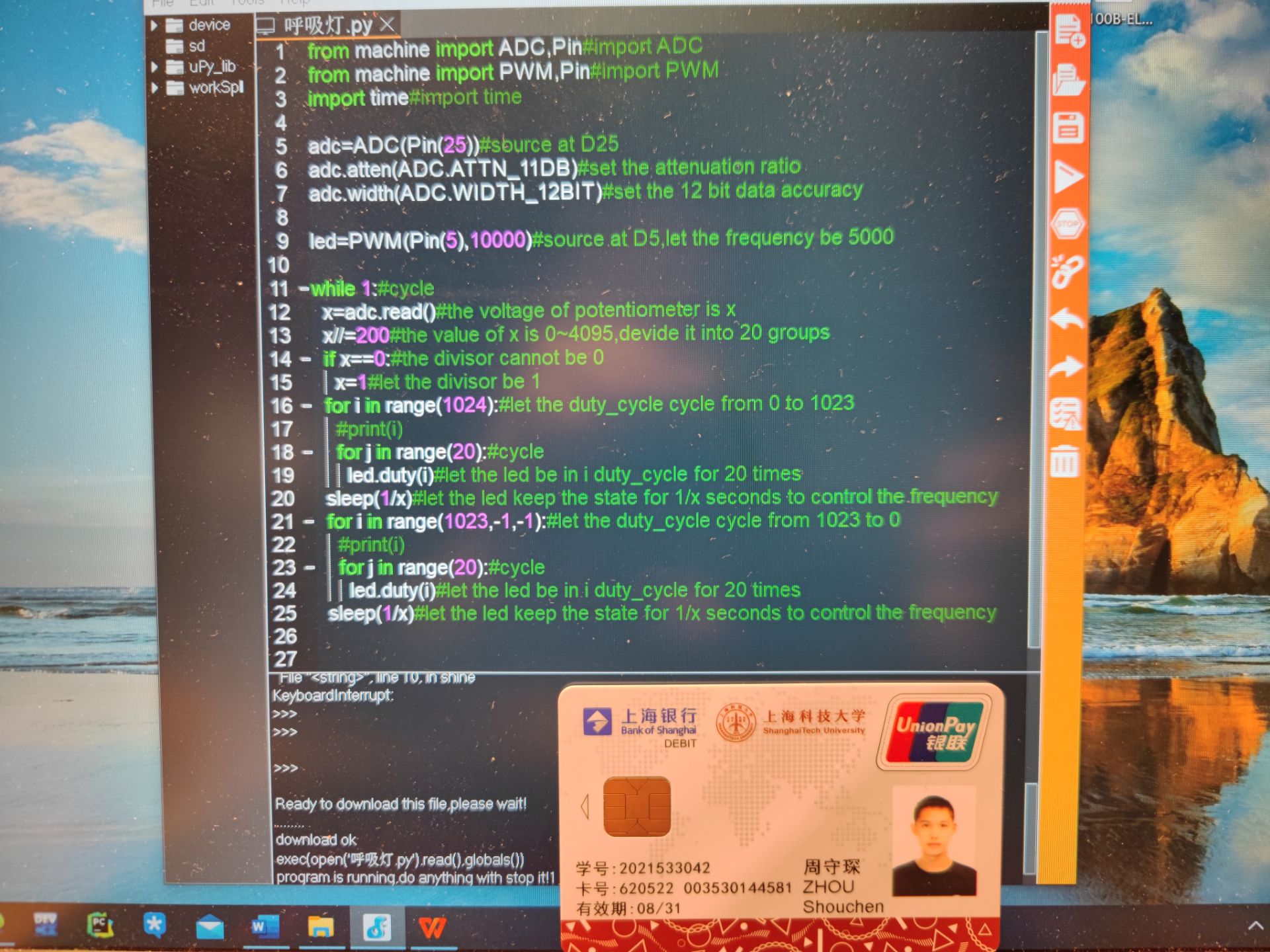


Adjust the potentiometer, let it get high voltage.

So brightness the of the led is high.



***Bonus question***



from machine import ADC,Pin#import ADCfrom machine import PWM,Pin#import PWMimport time#import timeadc=ADC(Pin(25))#source at D25adc.atten(ADC.ATTN\_11DB)#set the attenuation ratioadc.width(ADC.WIDTH\_12BIT)#set the 12 bit data accuracyled=PWM(Pin(5),10000)#source at D5,let the frequency be 5000while 1:#cycle x=adc.read()#the voltage of potentiometer is x x//=200#the value of x is 0~4095,devide it into 20 groups if x==0:#the divisor cannot be 0 x=1#let the divisor be 1 for i in range(1024):#let the duty\_cycle cycle from 0 to 1023 #print(i) for j in range(20):#cycle led.duty(i)#let the led be in i duty\_cycle for 20 times sleep(1/x)#let the led keep the state for 1/x seconds to control the frequency for i in range(1023,-1,-1):#let the duty\_cycle cycle from 1023 to 0 #print(i) for j in range(20):#cycle led.duty(i)#let the led be in i duty\_cycle for 20 times sleep(1/x)#let the led keep the state for 1/x seconds to control the frequency