# 实验3.2 扩展并行I/O口实验

姓名：孟麟芝 学号：201800121050 实验时间：2020.6.21

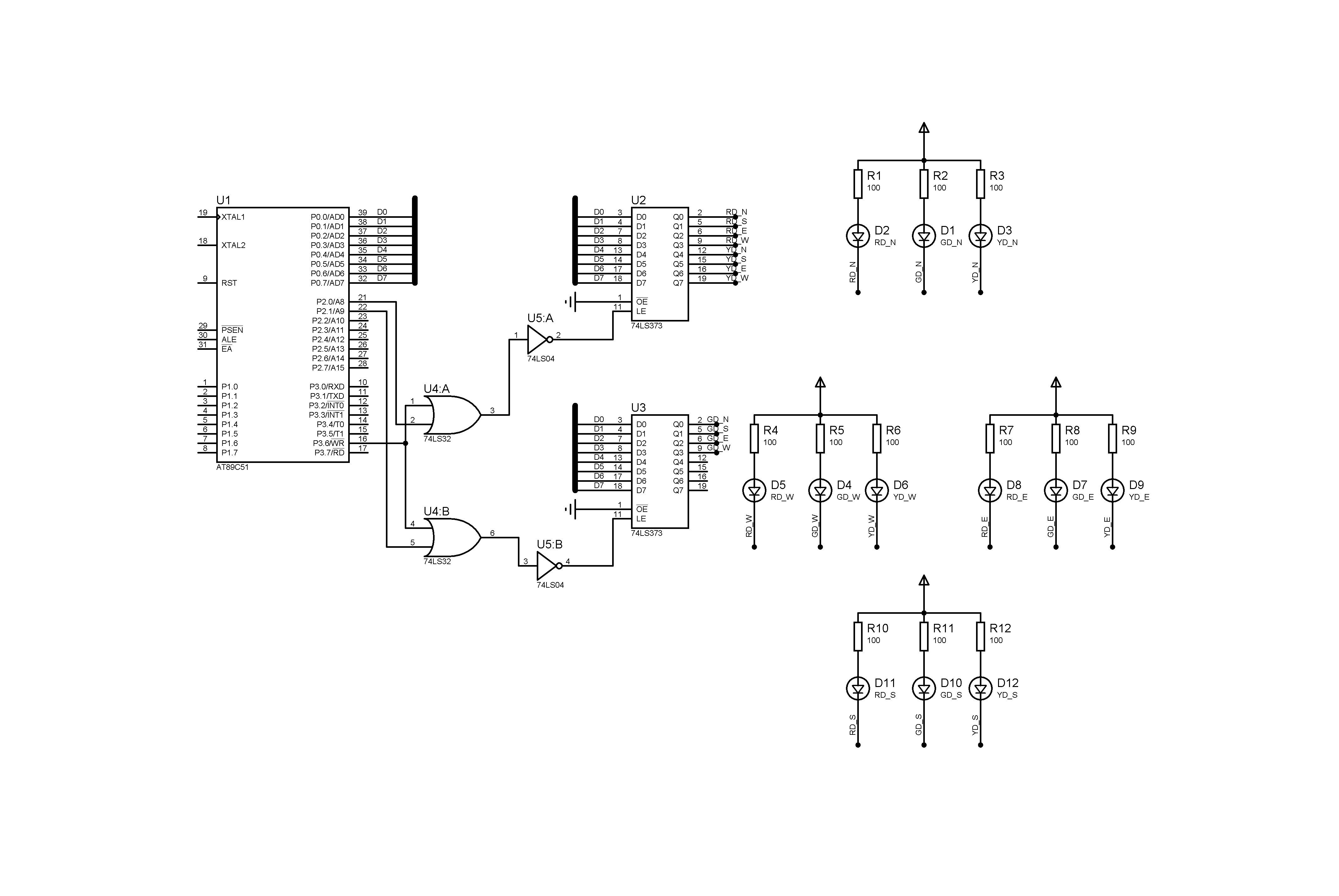
## 实验要求

（1）编写源程序并进行注释，叙述实验原理，画出流程图

（2）记录实验过程

（3）记录Proteus仿真结果

## 实验电路及功能



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **选件编号** | **元件名称** | **参数** | **所在元件库类名** | **子类名** | **生产厂家** |
| U1 | AT89C51 |  | Microprocessor ICs | 8051 Family | ATMEL |
| 微处理器 | 8051 家族 |
| U2，U3 | 74LS373 |  | TTL 74LS series | Flip-Flops & Latches |  |
| TTL 74LS 系列 | 触发与锁存 |
| U4 | 74LS32 |  | TTL 74LS series | Gates& Inverters |  |
| TTL 74LS 系列 | 门与反相器 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| U5 | 74LS04 |  | TTL 74LS series | Gates& Inverters |  |
| TTL 74LS 系列 | 门与反相器 |
| RD\_N  RD\_S  RD\_E  RD\_W | LED-RED |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |
| YD\_N  YD\_S  YD\_E  YD\_W | LED-YELLOW |  | Optoelectronics | LEDs |  |
| 光电子器件 | 发光二极管 |

实验功能为，仿真实现交通信号灯控制功能。

控制顺序为：

（1）南北绿灯亮，同时东西红灯亮10s；

（2）南北黄灯亮，同时东西红灯亮2s；

（3）南北红灯亮，同时东西绿灯亮10s；

（4）东西黄灯亮，同时南北红灯亮2s；

（5）重复（1）～（4）。

## 实验原理

（1）进行初始化工作，包括设置堆栈指针SP，将两个373的输出口所有位均设置为1，使所有发光二极管全部熄灭。

（2）分析两个74LS373的选通地址： 假定所有无关地址均定义为1，那么U2的选通地址为：#0FE00H，U3的选通地址为：#0FD00H。

（3）分析4个状态下两个373的输出数据值，假定：

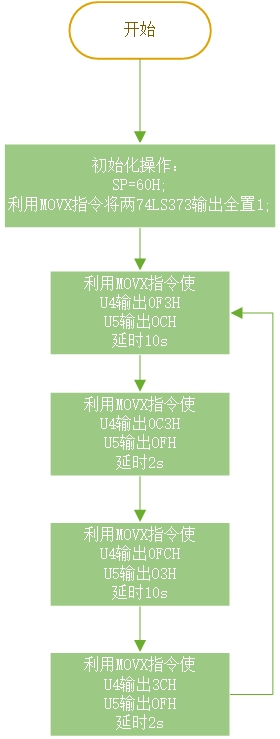
“南北绿灯亮，同时东西红灯亮”为状态1，即：Stat1。这时U2输出0F3H，U3输出0CH；

“南北黄灯亮，同时东西红灯亮”为状态2，即：Stat2；这时U2输出0C3H，U3输出0FH；

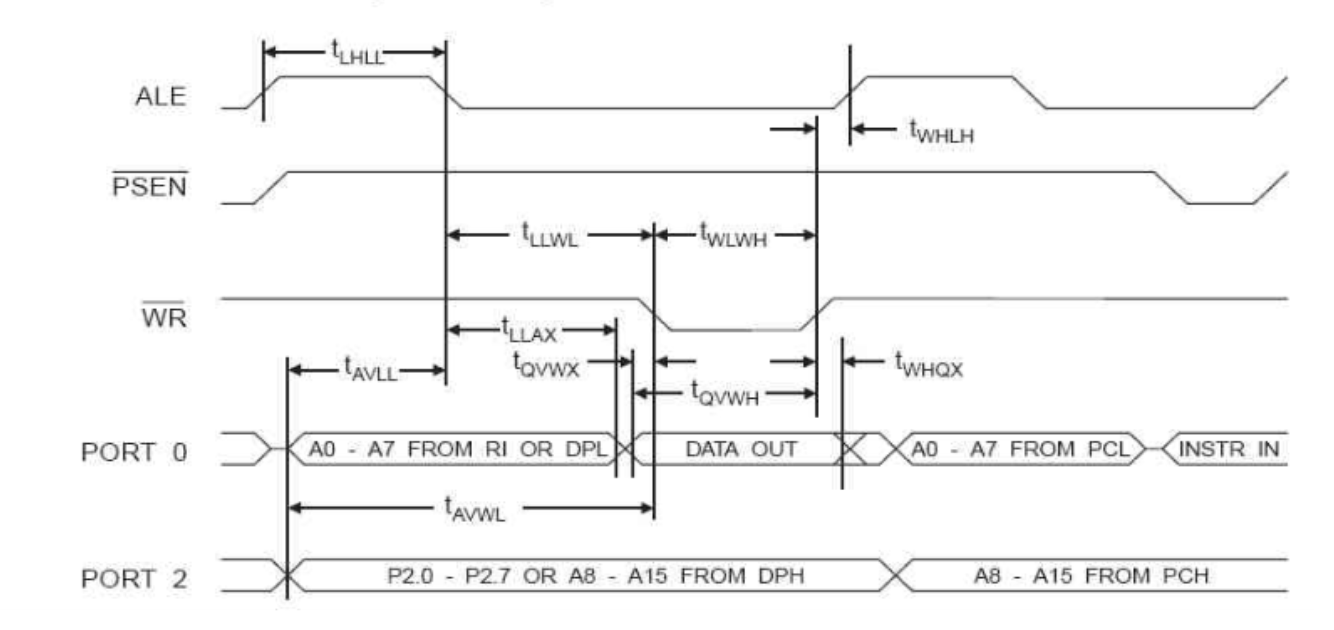
“南北红灯亮，同时东西绿灯亮”为状态3，即：Stat3；这时U2输出0FCH，U3输出03H；

“东西黄灯亮，同时南北红灯亮”为状态4，即：Stat4。这时U2输出3CH，U3输出0FH；

实验流程图如下：



下面是MOVX指令执行的时序



## 实验过程

（1）根据上述实验内容，参考1.2.2，在Proteus 环境下建立图3.7 所示原理图

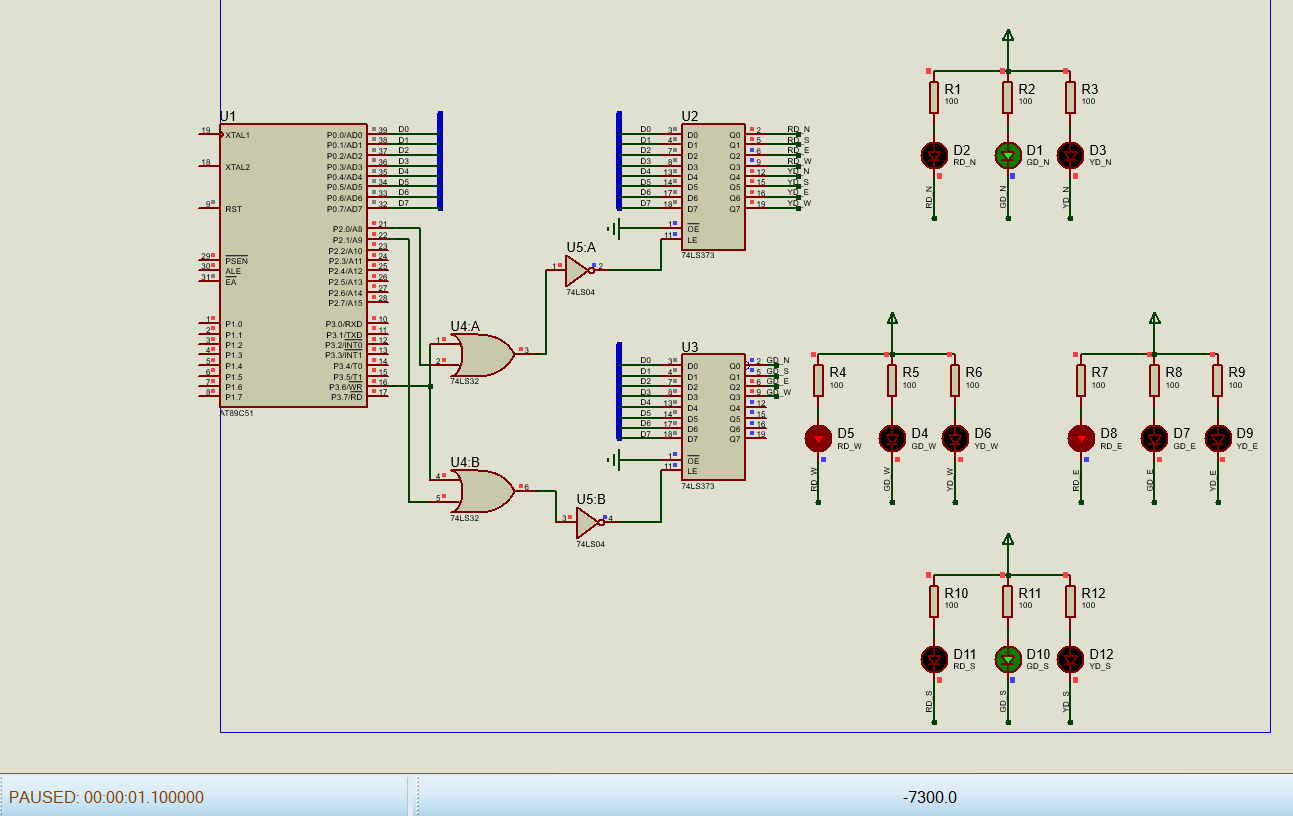
（2）根据（2）和（3）编写控制源程序。

（3）将源程序添加到U1 中，并构造（build）该程序。

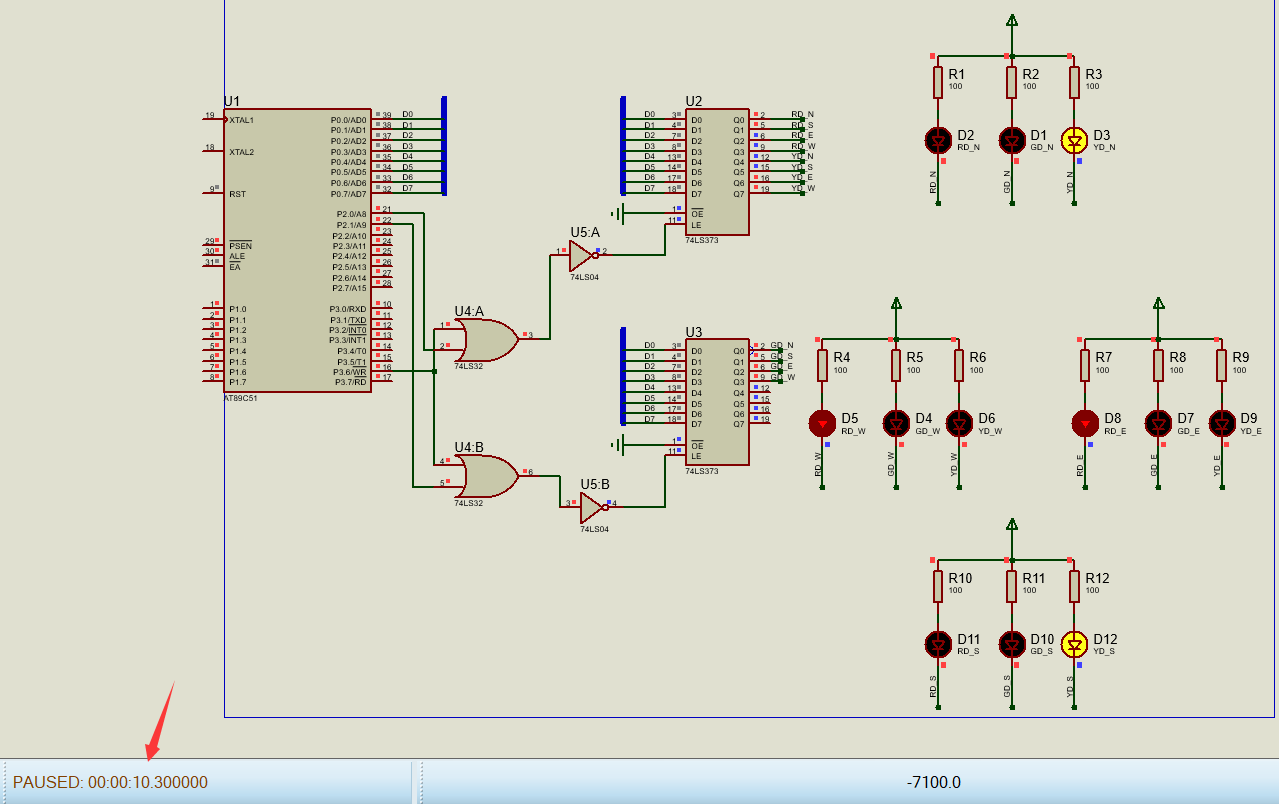
（4）执行仿真过程观察各个方向的交通信号灯指示，查看程序功能是否正确。

## 实验结果

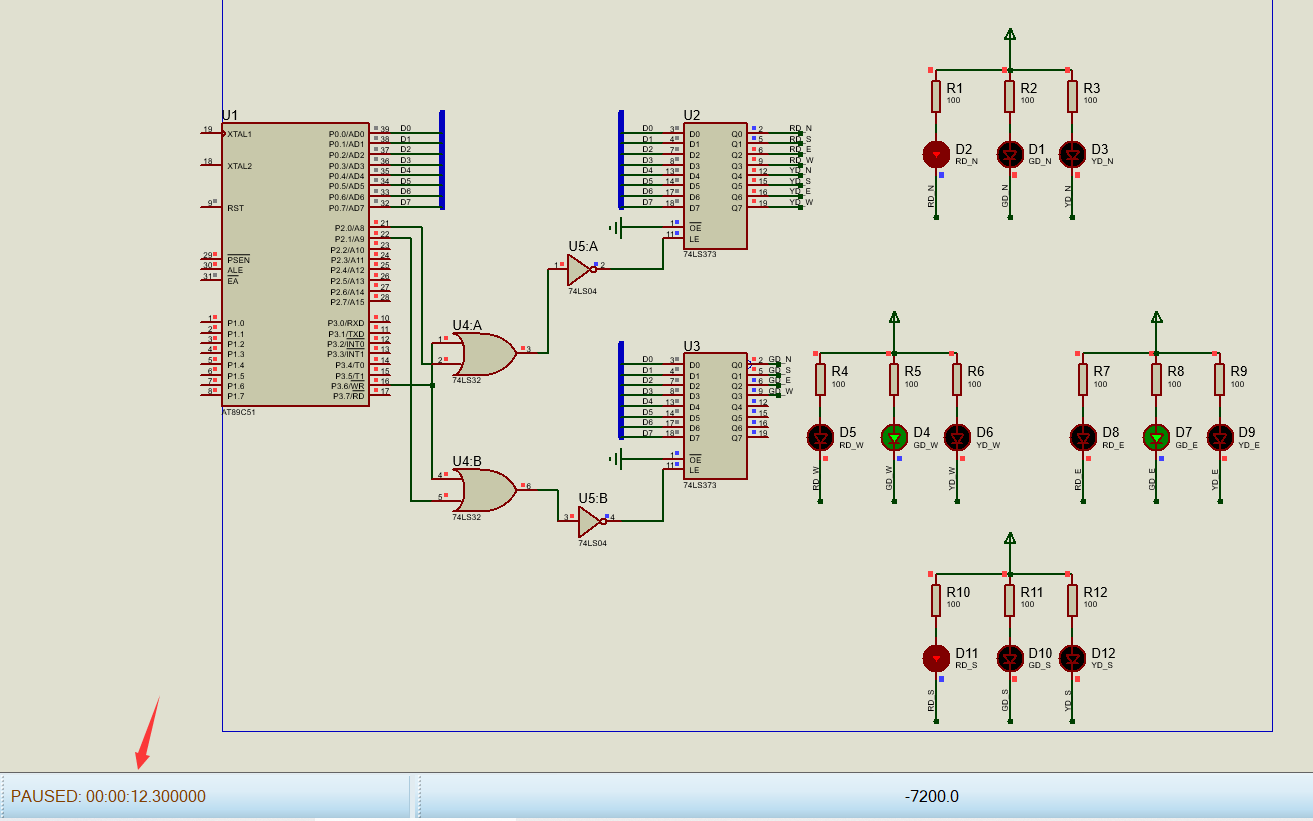
（1）最初为状态1



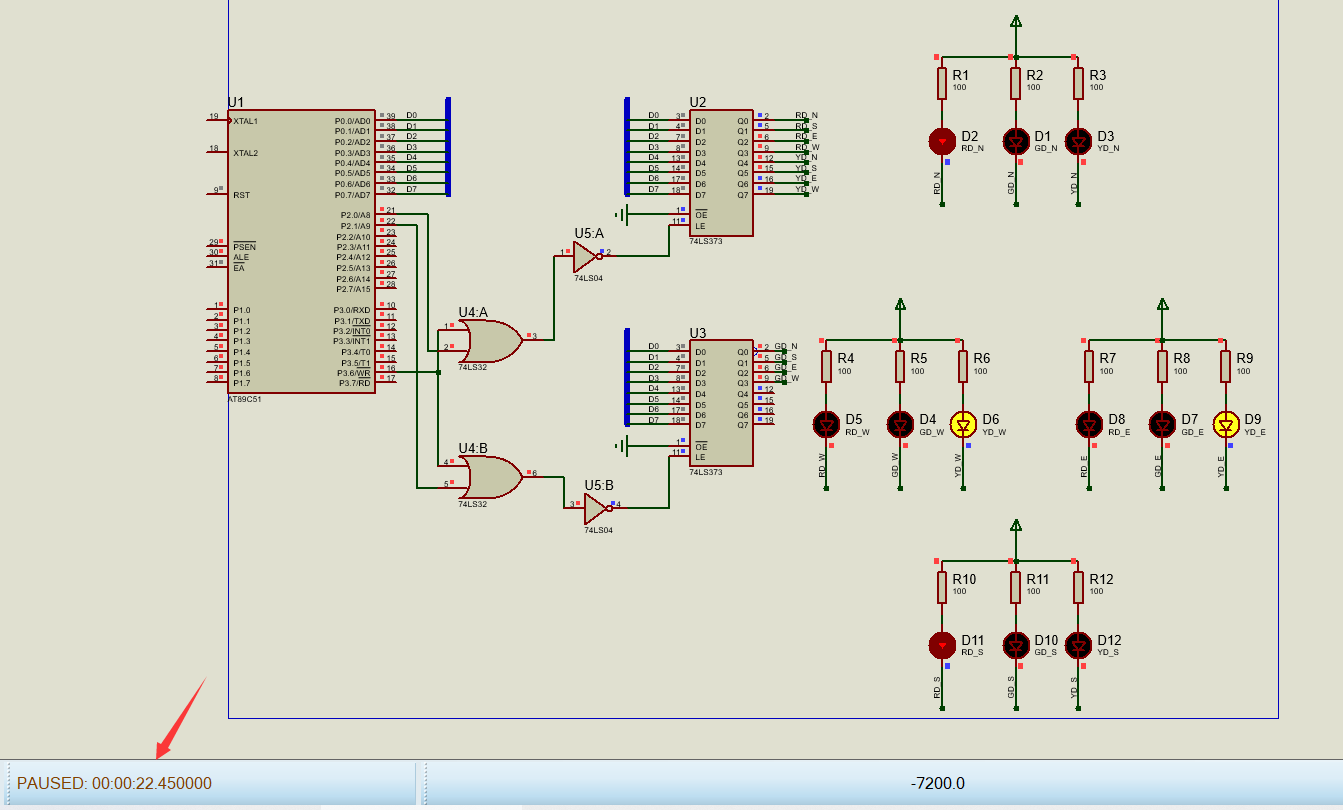
（2）经过10s，转入状态2



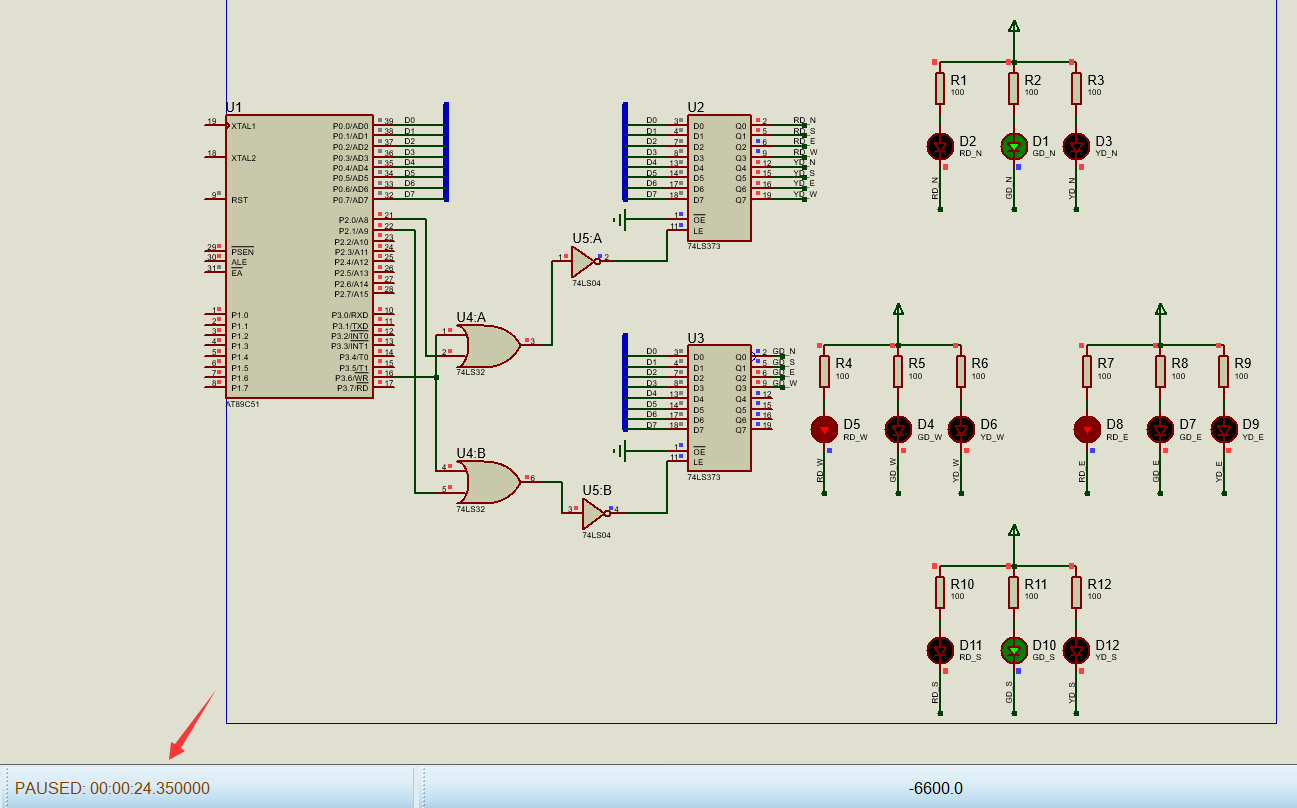
（3）经过2s，转入状态3



（4）经过10s，转入状态4



（5）又经过2s，回到状态1



## 实验源程序

ORG 0000H

AJMP MAIN

ORG 0030H

MAIN:

MOV SP,#60H

MOV A,#0FFH

MOV DPTR,#0FE00H

MOVX @DPTR,A

MOV DPTR,#0FD00H

MOVX @DPTR,A

STATE1:

MOV A,#0F3H

MOV DPTR,#0FE00H;利用@DPTR使用MOVX时，P2口输出DPH，P0在输出数据前输出DPL

MOVX @DPTR,A;利用@DPTR使用MOVX时，会在P0口输出数据时使WR端为低电平，配合P2对DPH输出，LE为高电平，74LS373被选通

MOV A,#0CH

MOV DPTR,#0FD00H

MOVX @DPTR,A

LCALL DL10S

STATE2:

MOV A,#0C3H

MOV DPTR,#0FE00H

MOVX @DPTR,A

MOV A,#0FH

MOV DPTR,#0FD00H

MOVX @DPTR,A

LCALL DL2S

STATE3:

MOV A,#0FCH

MOV DPTR,#0FE00H

MOVX @DPTR,A

MOV A,#03H

MOV DPTR,#0FD00H

MOVX @DPTR,A

LCALL DL10S

STATE4:

MOV A,#03CH

MOV DPTR,#0FE00H

MOVX @DPTR,A

MOV A,#0FH

MOV DPTR,#0FD00H

MOVX @DPTR,A

LCALL DL2S

LJMP STATE1

DL10S:

MOV R7,#100

DL2:MOV R6,#200

DL1:MOV R5,#250

DJNZ R5,$

DJNZ R6,DL1

DJNZ R7,DL2

RET

DL2S:

MOV R7,#20

DL4:MOV R6,#200

DL3:MOV R5,#250

DJNZ R5,$

DJNZ R6,DL3

DJNZ R7,DL4

RET

SJMP $

END