**BÀI THÍ NGHIỆM 2**

**LẬP TRÌNH HIỂN THỊ LED 7 ĐOẠN VÀ GIAO TIẾP LED MA TRẬN**

Họ và tên : Dương Phúc Nguyên

MSSV : 1811109

Nhóm lớp: L10

1. **Thí nghiệm 1**

***Yêu cầu***:

* Viết chương trình con **DisplayLED** xuất giá trị 9 lên LED 7 đoạn thứ 0.

DisplayLED: MOV A,#0E9h

MOV DPTR,#0000h

MOVX @DPTR,A

RET

* Sau đó, dùng chương trình con này viết chương trình xuất giá trị 9 lên LED 7 đoạn số 0.

ORG 2000H

MAIN: ACALL DisplayLED

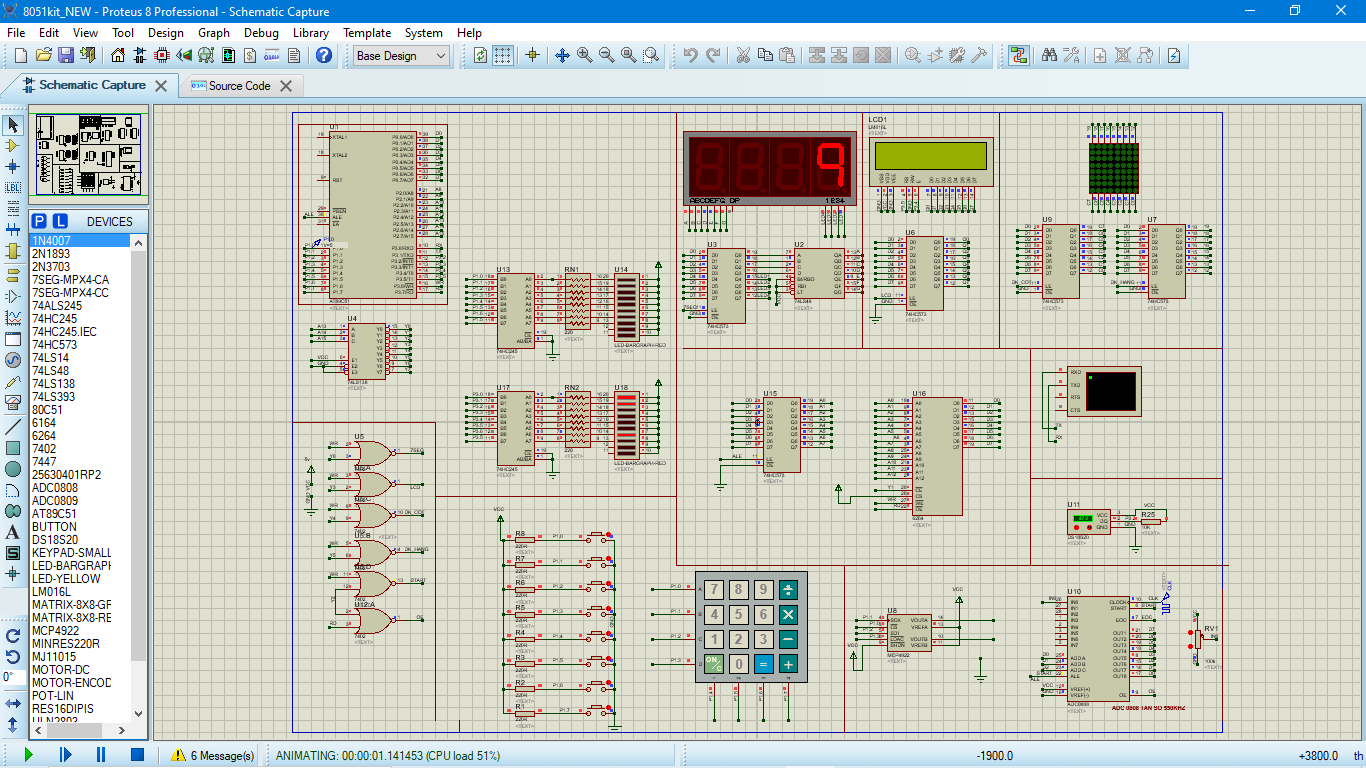
DisplayLED: MOV A,#0E9h

MOV DPTR,#0000h

MOVX @DPTR,A

RET

END

***Kết quả thí nghiệm:***

*Led 7 đoạn thứ 0 hiển thị số 9*

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình.
* Để giao tiếp với LED 7 đoạn, ta phải ghi ra ngoại vi ở vùng nhớ (giá trị nạp vào thanh ghi DPTR) bao nhiêu??
* Để giao tiếp với LED 7 đoạn, ta phải ghi ra ngoại vi ở 0x0000 - 0x1FFF ở vùng nhớ dữ liệu ngoài của 8051.
* Byte được ghi ra phải có định dạng như thế nào? Muốn hiển thị lên các LED còn lại, bit nào sẽ bằng 0?
* Byte được ghi ra phải có định dạng mã hex. Muốn hiển thị lên các LED còn lại bit số 4, 5, 6, 7 phải bằng 0 tương ứng với LED 0, 1, 2, 3.

**Thí nghiệm 2**

***Yêu cầu***: Sửa lại chương trình con **DisplayLED** ở Thí nghiệm 1 (có thể tham khảo đoạn chương trình ở dưới) để thực hiện xuất giá trị chứa trong thanh ghi R0 lên LED 7 đoạn có số thứ tự chứa trong thanh ghi R1. (R0 nằm trong tầm từ 0-9 và R1 trong tầm từ 0-3. Sau đó, dùng chương trình con này viết chương trình xuất 1 giá trị bất kỳ lên LED 7 đoạn số 3.

ORG 2000H

MAIN:

MOV R1,#1011B

MOV R0,#0111B

ACALL DisplayLED

DisplayLED:

MOV B, R1

MOV A, #11110111B

SHIFTLOOP:

RL A

DJNZ B, SHIFTLOOP

ANL A, #0F0H

PUSH ACC

MOV A, R0

ANL A, #0FH

MOV R0, A

POP ACC

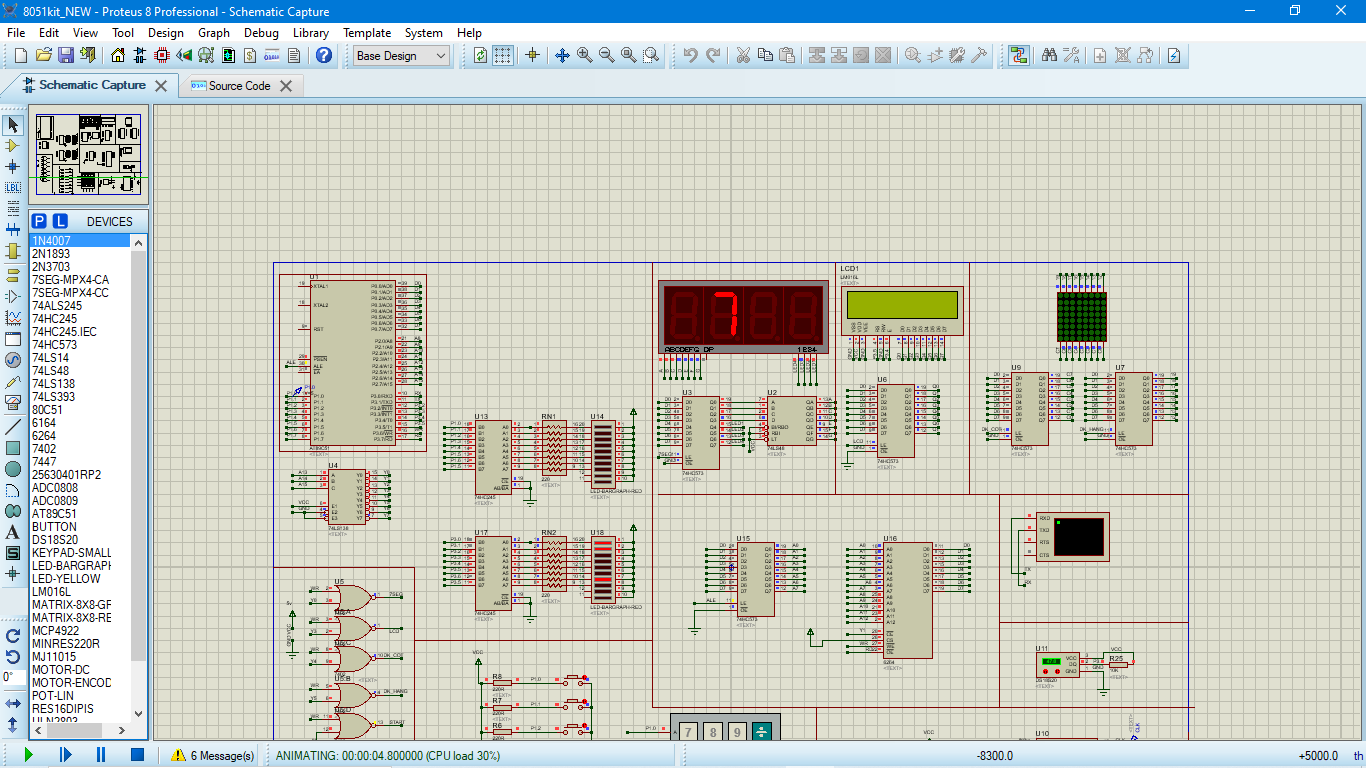
ORL A, R0

MOV DPTR,#0001H

MOVX @DPTR,A

RET

END

***Kết quả thí nghiệm:***

*Led 7 đoạn thứ 3 hiển thị giá trị 7.*

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình.
* Giải thích ý nghĩa các lệnh của đoạn chương trình trong hướng dẫn.

MOV B, R1 ; lưu giá trị của R1 vào thanh ghi B

MOV A, #11110111B ;bit số 3 bằng 0, bốn bit cao bằng 1.

SHIFTLOOP:

RL A ;Dịch chuyển các bit của giá trị chứa trong thanh ghi A sang trái.

DJNZ B, SHIFTLOOP ; Vòng lặp để dịch chuyển bit 0 trong thanh ghi A sang vị trí tương ứng với LED 7 đoạn cần hiển thị.

ANL A, #0F0H ; phép AND giá trị trong thanh ghi A và 11110000b nhằm xóa 4 bit thấp của giá trị đó và giữ lại 4 bit cao.

PUSH ACC ;đẩy giá trị trong thanh ghi A vào ô nhớ ngăn xếp.

MOV A, R0 ;lưu giá trị thanh ghi R0 ( lưu giá trị được hiển thị) vào A;

ANL A, #0FH ;phép AND giá trị trong thanh ghi A VÀ 00001111b nhằm xóa giá trị 4 bit cao và giữ lại 4 bit thấp của A.

MOV R0, A ;lưu giá trị của A vào thanh ghi R0

POP ACC ;xuất giá trị đã lưu vào ngăn xếp vào thanh ghi A.

ORL A, R0 ;phép OR bit giá trị của thanh ghi A (chứa vị trí LED 7 đoạn cần hiển thị) và giá trị của thanh ghi R0 (chứa giá trị cần hiển thị lên LED 7 đoạn).

* Viết lại chương trình DisplayLED sử dụng phương pháp tra bảng.

ORG 2000H

MAIN:

ACALL DisplayLED

DisplayLED:

MOV DPTR, #TABLE

MOV A, #7

MOVC A, @A+DPTR

MOV DPTR, #0000h

MOVX @DPTR, A

TABLE:

DB 0B0h,0B1h,0B2h,0B3h,0B4h,0B5h,0B6h,0B7h,0B8h,0B9h

RET

END

1. **Thí nghiệm 3**

***Yêu cầu***: Viết chương trình hiển thị lên LED 7 đoạn số 0 các con số từ 0 đến 9 sau các khoảng thời gian 1s.

ORG 2000H

MAIN:

MOV R1,#1H

REVERSE: MOV R0,#0000B

INCLOOP: ACALL DisplayLED

ACALL Delay1sIns

INC R0

CJNE R0,#0Ah,INCLOOP

SJMP REVERSE

DisplayLED:

MOV B, R1

MOV A, #11110111B

SHIFTLOOP:

RL A

DJNZ B, SHIFTLOOP

ANL A, #0F0H

PUSH ACC

MOV A, R0

ANL A, #0FH

MOV R0, A

POP ACC

ORL A, R0

MOV DPTR,#0001H

MOVX @DPTR,A

RET

Delay1sIns:

MOV R7,#8;

LP2: MOV R6,#250;

LP1: MOV R5,#250

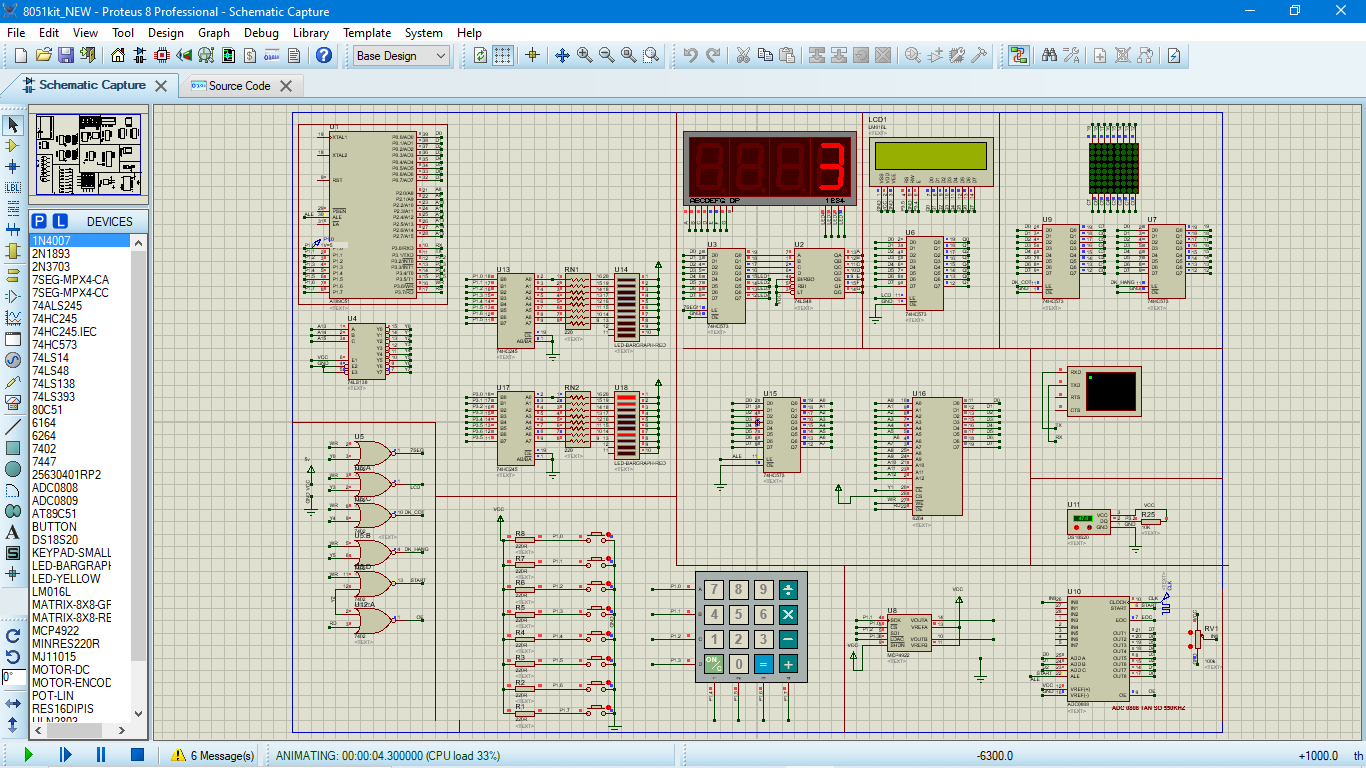
DJNZ R5,$;

DJNZ R6, LP1;

DJNZ R7, LP2;

RET

END

***Kết quả thí nghiệm:***

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình.
* Để kiểm tra R0 lớn hơn 9 cần sử dụng lệnh nào? Vòng lặp nhảy như thế nào??
* Để kiểm tra R0 lớn hơn 9 cần sửa dụng lệnh CJNE.

CJNE R0,#0Ah,INCLOOP

* Ta so sánh giá trị của R0 với #0Ah nếu không bằng thì lệnh tiếp theo được thực hiện, còn nếu bằng thì sẽ nhảy đến nhãn phía sau của CJNE

1. **Thí nghiệm 4**

***Yêu cầu:***

* Đầu tiên, viết chương trình hiển thị số 1 lên LED 7 đoạn 0, sau đó 1s hiển thị số 2 lên LED 7 đoạn 1, sau đó 1s hiển thị số 3 lên LED 7 đoạn 2, sau đó 1s hiển thị số 4 lên LED 7 đoạn 3. Quá trình này lặp đi lặp lại. Việc hiển thị này sử dụng chương trình con DisplayLED đã viết ở trên.

ORG 2000H

MAIN:

REVERSE: MOV R1,#1H

MOV R0,#1H

INCLOOP: ACALL DisplayLED

ACALL Delay1sIns

INC R0

INC R1

CJNE R0,#5h,INCLOOP

SJMP REVERSE

DisplayLED:

MOV B, R1

MOV A, #11110111B

SHIFTLOOP:

RL A

DJNZ B, SHIFTLOOP

ANL A, #0F0H

PUSH ACC

MOV A, R0

ANL A, #0FH

MOV R0, A

POP ACC

ORL A, R0

MOV DPTR,#0001H

MOVX @DPTR,A

RET

Delay1sIns:

MOV R7,#8;

LP2: MOV R6,#250;

LP1: MOV R5,#250

DJNZ R5,$;

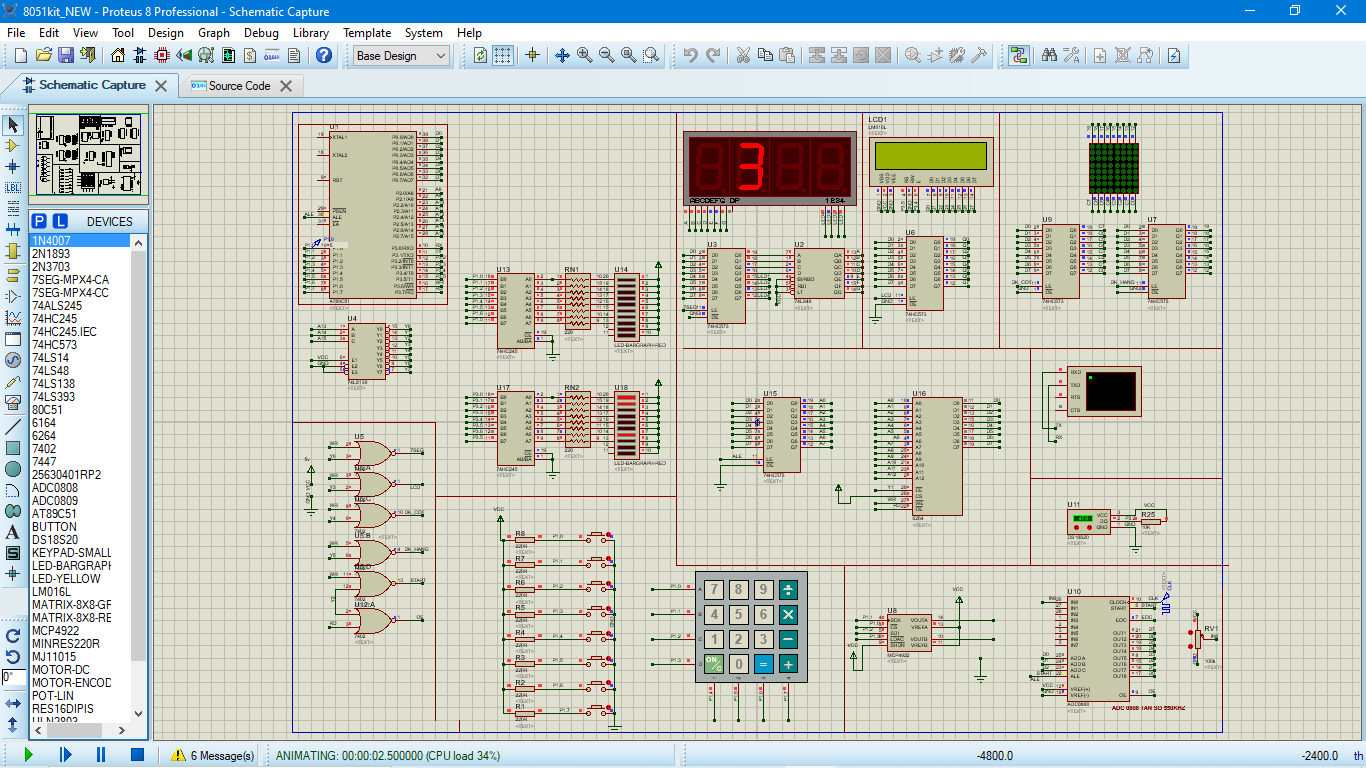
DJNZ R6, LP1;

DJNZ R7, LP2;

RET

END

* Giảm thời gian trễ xuống còn 100 ms. Quan sát hiện tượng.
* Các số trên LED hiển thị và dịch chuyển nhanh hơn.
* Tính toán thời gian trễ sao cho không còn thấy LED nhấp nháy và áp dụng vào chương trình.
* Giảm thời gian trễ xuống bằng 0 (không sử dụng hàm delay). Quan sát hiện tượng.
* Khi giảm thời gian trễ xuống bằng 0, ta quan sát Led dường như không sáng.

***Kết quả thí nghiệm:***

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình
* Thời gian trễ để không còn thấy LED nhấp nháy là bao nhiêu theo lý thuyết? Giải thích cách tính?
* Thời gian trễ để không còn thấy LED nhấp nháy theo lý thuyết là 1/24s. Mắt người chỉ có thể phân biệt được tần số dưới 24Hz, nếu trên 24Hz do hiện tượng lưu ảnh nên chúng ta không còn tháy LED nhấp nháy.
* Khi giảm rất nhỏ thời gian trễ, hiện tượng xảy ra là gì? Giải thích.
* Khi giảm rất nhỏ thời gian trễ, trên LED 7 đoạn hiển thị các số 4,3,2,1 và không có hiện tượng nhấp nháy. Vì khi giảm thời gian trễ rất thấp, các LED vẫn nhấp nháy nhưng với tốc độ mà mắt người không thể quan sát được, do đó chúng ta thấy được hiện tượng trên.

1. **Thí nghiệm 5**

***Yêu cầu***: Viết chương trình làm sáng các LED ở vị trí bất kỳ của hàng y trên LED ma trận. Giá trị y được chứa trong thanh ghi.

ORG 2000H

MAIN:

MOV R0,#0

MOV A,R0

MOV DPTR,#COT

MOVC A,@A+DPTR

MOV DPTR,#08000H

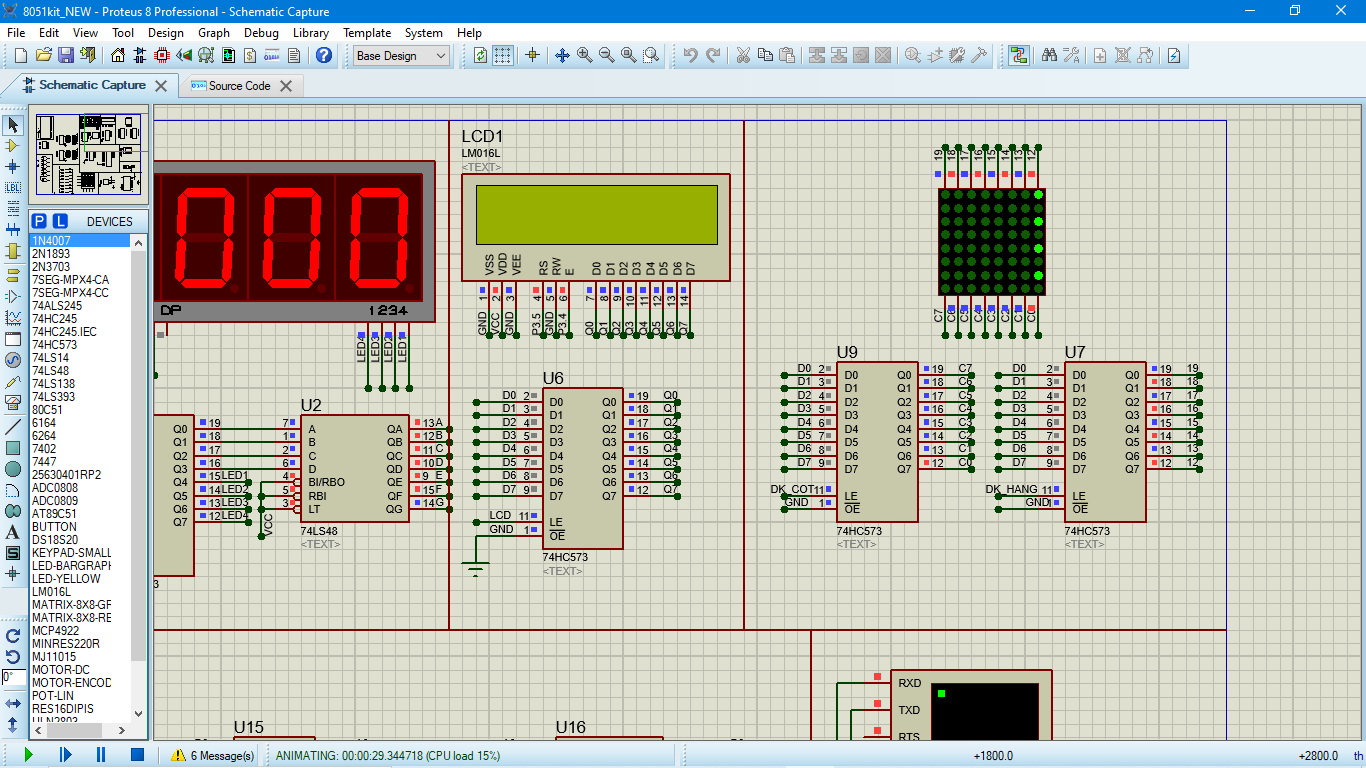
MOVX @DPTR,A

MOV DPTR,#0A000H

MOV A,#10101010b

MOVX @DPTR,A

COT: DB 10000000B, 01000000B, 00100000B, 00010000B, 00001000B, 00000100B, 00000010B, 00000001B

END

***Kết quả thí nghiệm:***

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình.
* Để chốt hàng và chốt cột của LED ma trận, ta lần lượt phải ghi ra ngoại vi ở vùng nhớ (giá trị nạp vào thanh ghi DPTR) bao nhiêu?
* Điều khiển chốt ‘573 chốt 8 bit dữ liệu của khối led matrix cột ở vùng nhớ 8000-9FFFh.
* Điều khiển chốt ‘573 chốt 8 bit dữ liệu của khối led matrix hàng ở vùng nhớ A000h-BFFFh.
* Để làm sáng LED ở vị trí bất kỳ thì dữ liệu xuất trên cột và hàng phải là mức cao hay thấp ?
* Để làm sáng LED ở vị trí bất kì thì dữ liệu xuất trên cột là mức cao và trên hàng là mức thấp.

1. **Thí nghiệm 6**

***Yêu cầu***: Viết chương trình làm hiển thị chữ B lên LED ma trận.

ORG 2000H

MAIN:

LOOP: MOV R2,#0

MOV R1,#7

QUET: MOV A,R2

MOV DPTR,#COT

MOVC A,@A+DPTR

MOV DPTR,#08000h

MOVX @DPTR,A

MOV A,R1

MOV DPTR,#ChuB

MOVC A,@A+DPTR

MOV DPTR,#0A000h

MOVX @DPTR,A

ACALL Delay1ms

INC R2

DJNZ R1,QUET

MOV A, #7

MOV DPTR, #COT

MOVC A,@A+DPTR

MOV DPTR,#8000H

MOVX @DPTR,A

ACALL Delay1ms

MOV A, #0

MOV DPTR, #ChuB

MOVC A, @A+DPTR

MOV DPTR,#0A000H

MOVX @DPTR,A

ACALL Delay1ms

SJMP LOOP

MOV DPTR,#0A000H

MOV A,#10101010b

MOVX @DPTR,A

COT: DB 10000000B, 01000000B, 00100000B, 00010000B, 00001000B, 00000100B, 00000010B, 00000001B

ChuB: DB 00h,00h,76h,76h,76h,71h,8FH,0FFh

Delay1ms:

MOV R6,#4;

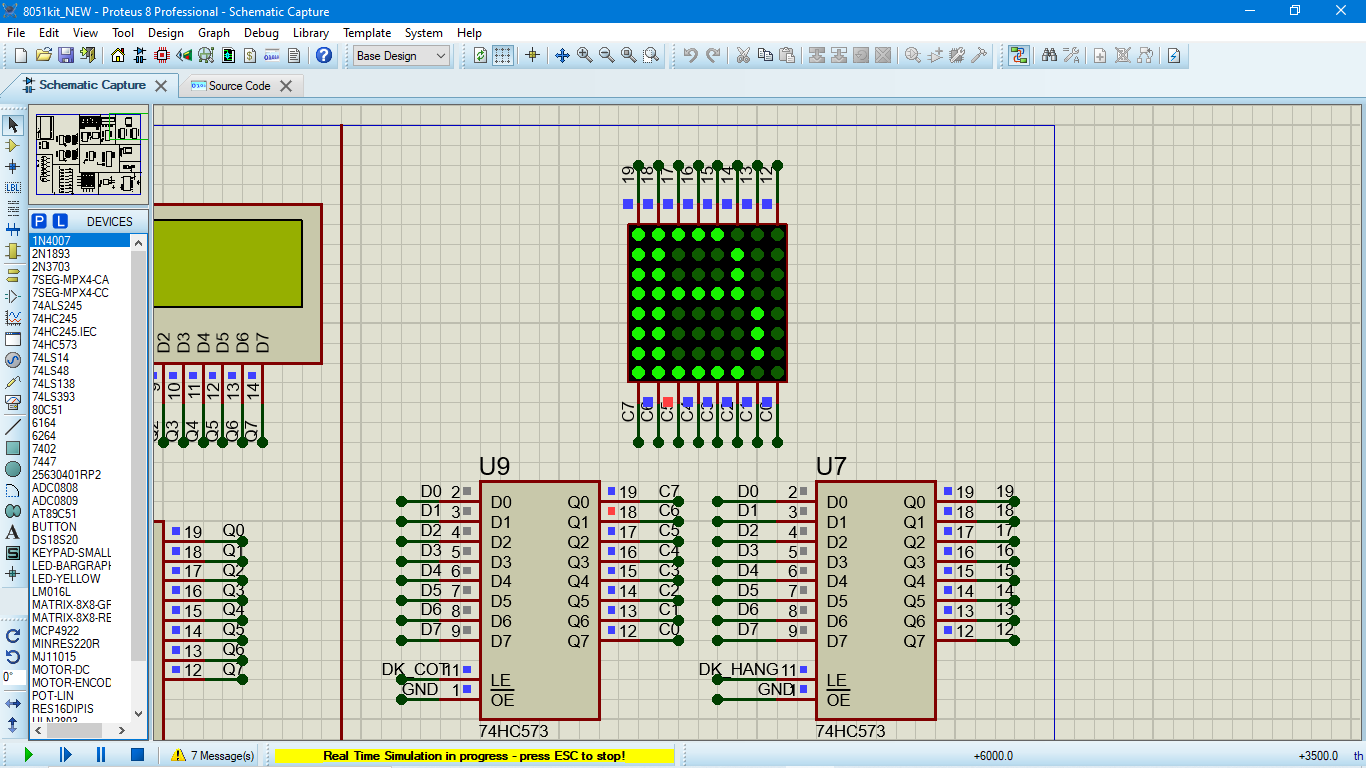
LAP1: MOV R5,#250

DJNZ R5,$;

DJNZ R6, LAP1;

RET

END

***Kết quả thí nghiệm:***

***Kiểm tra:***

* Biên dịch, thực thi và kiểm tra chương trình.
* Khi tăng hoặc giảm thời gian delay, hiển thị trên LED ma trận sẽ thay đổi như thế nào?
* Khi tăng thời gian delay chữ B trên Led ma trận hiển thị rõ ràng, khi giảm thời gian delay chữ B nhấp nháy.