

Báo cáo bài tập lớn Kỹ thuật vi xử lý: Mạch mô phỏng đèn giao thông

Ý tưởng thực hiện:

Nhận thấy giao thông hiện nay còn tình trạng tắc, nghẽn nên nhóm em thực hiện mô phỏng đèn giao thông ngã tư với 2 đường 1 chiều giao nhau sử dụng vi điều khiển 8051, nhằm nghiên cứu, mô phỏng quá trình hoạt động của đèn giao thông, đồng thời đưa ra một số ý kiến về chức năng của đèn giao thông để giảm thiểu tình trạng trên.

Chức năng của mạch:

Mạch thực hiện đếm lùi, số đếm được hiển thị qua led 7 đoạn. Khi mạch thực hiện đếm lùi, 6 led đỏ vàng và xanh sẽ lần lượt sáng biểu thị cho 2 cột đèn giao thông ở ngã tư.

Mạch có thể thay đổi thời gian sáng của các đèn thuận tiện điều tiết giao thông. Đồng thời cùng một chương trình, ta có thể áp dụng cho nhiều đèn giao ở các ngã tư có hai tuyến đường một chiều giao nhau mà không phải sửa chương trình để thay thời gian đếm.

Mạch có chức năng điều tiết lưu lượng giao thông, khi lượng xe lưu thông của hai tuyến đường không đồng đều bằng cách giảm thời gian đèn đỏ của tuyến đường có nhiều phương tiện tham gia (tăng lưu lượng xe qua tuyến đường), tăng thời gian chờ đèn đỏ cho tuyến đường còn lại (giảm lưu lượng xe qua tuyến đường). Chức năng được thực hiện nhờ các nút bấm. Khi ở chế độ này thời gian đếm ở nửa chu kì bị giảm một nửa.

Với các chức năng trên ta có thể kết hợp với một bộ điều khiển từ xa giúp cho các người điều khiển giao thông có thể theo dõi, điều khiển lượng xe lưu thông từ xa mà không phải đến địa điểm xảy ra tắc nghẽn.

Linh kiện làm mạch:

- Vi điều khiển 8051.
- Led xanh, đỏ, vàng mỗi loại 2 cái.
- Tụ thạch anh 12MHz, tụ gốm 33pF, Trở 330 Ω , 1 k Ω .
- Nút bấm.
- 2 led 7 đoạn.
- Nguồn 5V.

Quá trình hoạt động:

Chế độ 0:

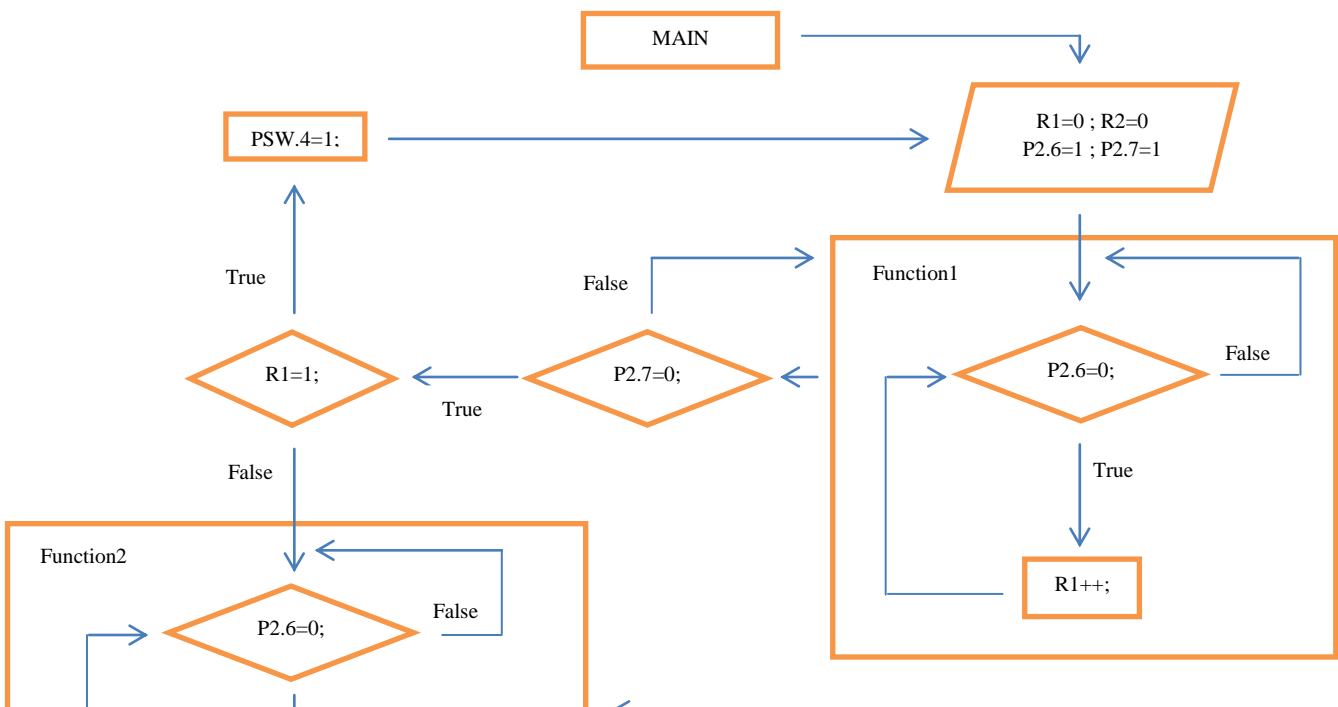
Khi mạch bắt đầu hoạt động ta thực hiện lựa chọn số đếm cho đèn giao thông. Ban đầu led 7 đoạn hiển thị giá trị 00, chân p2.6 và p2.7 được nối qua một nút bấm và nối xuống đất. Hai chân được đưa lên mức 1 qua lệnh được lập trình trên 8051, khi ấn nút bấm, chân p2.6 được nối đất chuyển xuống mức 0, vi điều khiển sẽ nhảy tới chương trình con thực hiện tăng số đếm được hiển thị qua led 7 đoạn, thời gian sáng của đèn đỏ bằng với giá trị hiển thị trên led, việc tăng số đếm được lặp lại khi ta tiếp tục ấn nút p2.6. Sau khi kết thúc lựa chọn thời gian sáng cho đèn đỏ, ta ấn nút p2.7 để tiếp tục lựa chọn thời gian sáng cho đèn vàng. Sau khi hoàn tất ta ấn tiếp nút p2.7 để mạch bắt đầu hoạt động, trong quá trình hoạt động ta có thể ấn tiếp nút p2.7 để đưa mạch về trạng thái ban đầu.

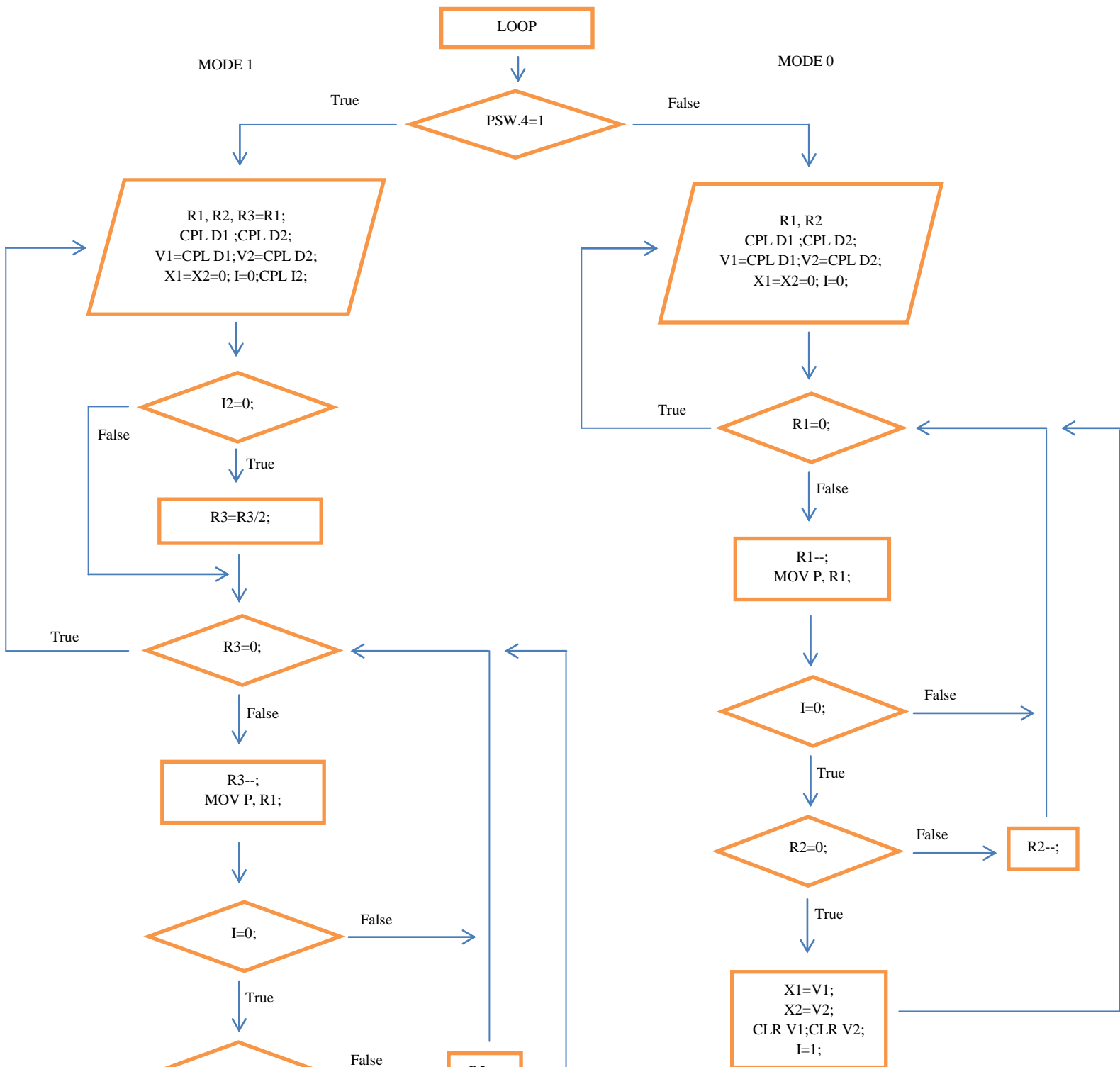
Cổng p1, p3 dùng hiển thị số đếm ra led 7 đoạn, cổng p2 dùng nối với các led biểu thị đèn giao thông, cổng p0 sử dụng làm các chân điều kiện chạy chương trình.

Chế độ 1:

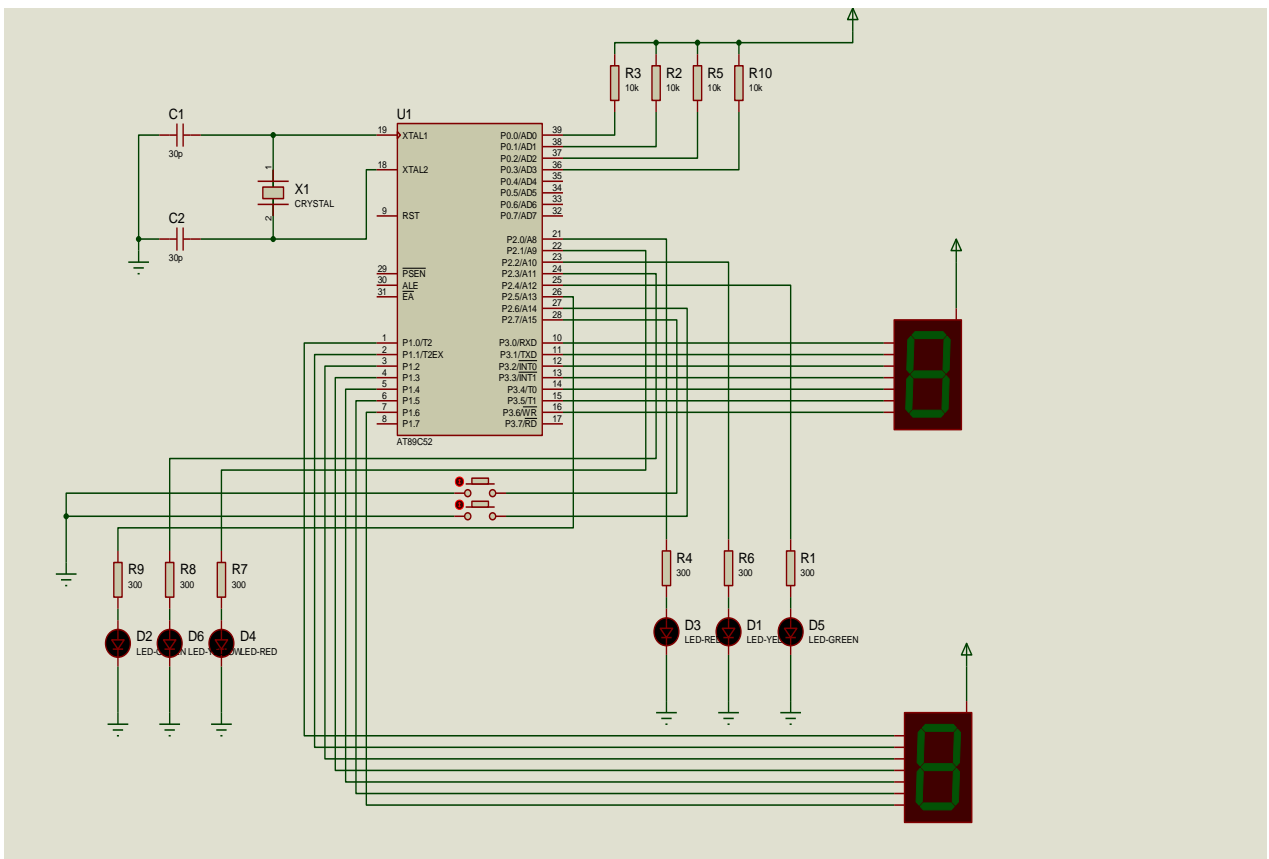
Khi thời gian sáng của đèn đỏ bằng 01, mạch chuyển sang chế độ 1. Ở chế độ này vi điều khiển chuyển băng thanh ghi, thời gian sáng tiếp tục được lập trình như chế độ 0. Sau khi lập trình xong trong nửa chu kỳ đầu mạch thực hiện đếm bình thường, nửa chu kỳ sau thời gian sáng của đèn đỏ giảm một nửa, nút p2.6 có nhiệm vụ chuyển đổi việc lựa chọn giảm thời gian sáng của đèn đỏ giữa hai cột đèn giao thông. Khi ấn nút p2.7 mạch được chuyển về trạng thái ban đầu ở chế độ 0.

Thuật toán:





Mạch mô phỏng trên proteus:



CODE:

```
org 00H
MAIN:
LCALL DELAY
MOV DPTR, #MALED
MOV A, #0
MOVC A, @A+DPTR
MOV P1, A
MOV P2, #0
MOV A, #0
MOVC A, @A+DPTR
MOV P3, A
MOV R0, #0
CLR P0.0
CLR P0.3
SETB P2.6
SETB P2.7
MOV R7, #0
MOV 34H, #1
MOV R1, #0
MOV R2, #0
MOV A, #0
MOV R6, #10
LJMP TIM
NHAN1:
DEC R0
MOV A, R0
JNZ TIM40
MOV R0, #9
DEC 34H
TIM40:
MOV A, 34H
DEC A
MOV 34H, A
```

```

MOV R7, 34H
LJMP TIM26
TIM27:
MOV A, R7
JNZ TIM15
MOV A, R0
DEC A
JNZ TIM15
SETB PSW.4
SETB P0.1
LJMP MAIN
TIM15:
MOV A, #0
MOV R6, #10
MOV R1, #0
LCALL DELAY
LJMP TIM2
TIM4:
MOV P2, #0
SETB P2.6
SETB P2.7
DEC R2
CLR P2.0
SETB P2.3
LOOP:
JNB PSW.4, TIM19
JNB P0.1, TIM25
JB P0.1, TIM16
TIM19:
MOV A, #0
MOVC A, @A+DPTR
MOV P1, A
LCALL DELAY
LCALL DELAY
CPL P2.0
CPL P2.3
MOV C, P2.0
CPL C
MOV P2.1, C
MOV C, P2.3
CPL C
MOV P2.4, C
CLR P2.2
CLR P2.5
JNB PSW.4, TIM22
CLR P0.0
JB P0.1, TIM17
TIM22:
MOV R7, 34H
TIM17:
JNB PSW.4, TIM23
JNB P0.1, TIM23
MOV R7, 35H
TIM23:
MOV A, R7
MOVC A, @A+DPTR
MOV P3, A
TIM24:
MOV A, R2
MOV R6, A
TIM21:

```

```

MOV A,R0
MOV R1,A
JNB PSW.4,LOOP1
JNB P0.1,TIM20
MOV R1,36H
TIM20:
CPL P0.1
LOOP1:
JNB P2.7,TIM36
MOV A,R1
MOVC A,@A+DPTR
MOV P1,A
LCALL DELAY
LCALL DELAY
MOV A,R7
JZ TIM11
TIM12:
JNB P2.6,TIM13
MOV A,R6
DEC A
JZ NHAN
LOOP4:
MOV R6,A
JNB P2.7,TIM36
DJNZ R1,LOOP1
JBC P0.3,TIM19
MOV A,R7
DEC A
MOV R7,A
LJMP TIM8
TIM11:
SETB P0.3
LJMP TIM12
TIM25:
LJMP TIM18
TIM36:
CLR PSW.4
LCALL DELAY
LCALL DELAY
LJMP MAIN
TIM16:
JNB P0.1,TIM19
MOV A,34H
MOV B,#2
DIV AB
MOV R7,A
MOV 35H,A
MOV A,B
MOV B,#10
MUL AB
ADD A,R0
MOV B,#2
DIV AB
MOV 36H,A
LJMP TIM19
NHAN:
MOV A,R1
JZ TIM7
SETB P0.0
MOV C,P2.1
MOV P2.2,C

```

```

MOV C, P2.4
MOV P2.5, C
CLR P2.1
CLR P2.4
LJMP LOOP4
TIM18:
SETB P0.1
LJMP LOOP
TIM13:
JB PSW.4, TIM35
LJMP MAIN
TIM35:
CPL P2.0
CPL P2.3
CLR P0.3
LJMP TIM16
TIM8:
MOV A, R6
DEC A
MOV R6, A
JZ NHAN
TIM7:
MOV A, #00H
MOVC A, @A+DPTR
MOV P1, A
LCALL DELAY
LCALL DELAY
MOV R1, #9
MOV A, R7
MOVC A, @A+DPTR
MOV P3, A
JB P0.0, TIM32
MOV A, R6
JZ TIM31
TIM32:
LJMP LOOP1
TIM31:
MOV C, P2.1
MOV P2.2, C
MOV C, P2.4
MOV P2.5, C
CLR P2.1
CLR P2.4
LJMP LOOP1
TIM:
JNB P2.6, TIM1
MOV A, R1
JNZ TIM37
MOVC A, @A+DPTR
MOV P1, A
MOV P3, A
INC R1
TIM37:
JNB P2.7, TIM10
AJMP TIM
TIM1:
MOV A, R1
MOVC A, @A+DPTR
MOV P1, A
INC R1
MOV A, R1

```



```

MOV R0,A
MOV A,R6
DEC A
MOV R6,A
JZ TIM5
MOV A,#0
LCALL DELAY
LJMP TIM
TIM2:
JNB P2.6,TIM3
MOV A,R1
JNZ TIM38
MOVC A,@A+DPTR
MOV P1,A
MOV P3,A
INC R1
TIM38:
JNB P2.7,TIM9
AJMP TIM2
TIM3:
MOV A,R1
MOVC A,@A+DPTR
MOV P1,A
INC R1
MOV A,R1
MOV R2,A
MOV A,R6
DEC A
MOV R6,A
JZ TIM6
MOV A,#0
LCALL DELAY
LJMP TIM2
DELAY:
MOV R5,#02
LAP1:
MOV R4,#0FFH
LAP2:
MOV R3,#0FFH
LAP3:
NOP
DJNZ R3,LAP3
DJNZ R4,LAP2
DJNZ R5,LAP1
RET
TIM5:
MOV R1,#0
MOV A,R1
MOVC A,@A+DPTR
MOV P1,A
MOV A,34H
MOVC A,@A+DPTR
MOV P3,A
MOV A,34H
INC A
MOV R7,A
MOV 34H,A
MOV A,#0
MOV R6,#11
LJMP TIM
TIM6:

```

```

MOV R1, #0
MOV A, R1
MOVC A, @A+DPTR
MOV P1, A
MOV R6, #11
LJMP TIM2
TIM9:
LJMP TIM4
TIM10:
LJMP NHAN1
TIM26:
MOV A, R0
CLR C
SUBB A, #10
JNZ TIM28
MOV A, 34H
DEC A
MOV 34H, A
MOV R0, #9
TIM28:
LJMP TIM27
MALED: DB 0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H
        END

```