Student Name: Nguyễn Tiến Đạt

ID Student: 20119125

Exam 1: : write 8 LED control program in turn flashing, staggered

C	ASM
Size: 218 bytes	**Size: : 120 bytes**
#include <at89x51.h> void delay(int interval){ int i, j; for(i = 0; i < 255; i++){ for(j = 0; j < interval; j++); } } void main(){ while(1){ P0 = 0x55; delay(100); P0 = 0xAA; delay(100); } }	**$NOMOD51 $INCLUDE (8051.MCU) org 0000h jmp Start org 0100h Start: MOV TMOD, #01h Loop: MOV P0 , #055h ACALL delay1 MOV P0, #0AAh ACALL delay1 jmp Loop delay1: MOV TH0 , #HIGH(-50000) MOV TL0 , #LOW(-50000) SETB TR0 JNB TF0 , $ CLR TR0 CLR TF0 RET END**

Exam 2: Create the 8 LED control program that blinks from left to right.

C	ASM
Size: 733 bytes	**Size: : 118B**
#include <at89x51.h> #include <stdio.h> #define LED0 P1_0 #define LED1 P1_1 #define LED2 P1_2 #define LED3 P1_3 #define LED4 P1_4	**$NOMOD51 $INCLUDE (8051.MCU) org 0000h jmp Start org 0100h**

```
#define LED5 P1_5
#define LED6 P1_6
#define LED7 P1_7
#define sang 1
#define tat 0



void delay(unsigned int ms){
      unsigned int i,j;
      for(i=0;i<ms;i++){
            for(j=0;j<120;j++){}
      }
}

void display_LED(unsigned char
number){
      switch(number){
            case 1:
                  LED0 = sang;
                  LED1 = LED2 = LED3 =
LED4 = LED5 = LED6 = LED7 = tat;
                  break;
            case 2:
                  LED1 = sang;
                  LED0 = LED2 = LED3 =
LED4 = LED5 = LED6 = LED7 = tat;
                  break;
            case 3:
                  LED2 = sang;
                  LED0 = LED1 = LED3 =
LED4 = LED5 = LED6 = LED7 = tat;
                  break;
            case 4:
                  LED3 = sang;
                  LED0 = LED1 = LED2 =
LED4 = LED5 = LED6 = LED7 = tat;
                  break;
            case 5:
                  LED4 = sang;
                  LED0 = LED1 = LED2 =
LED3 = LED5 = LED6 = LED7 = tat;
                  break;
            case 6:
                  LED5 = sang;
```

```
Start:
   cjne A, #00h, Main
   mov A, #001H

Main: MOV R5, #10D

DELAY: MOV TMOD, #01h
   MOV TH0 , #03CH
   MOV TL0, #08FH
   SETB TR0

OverBit: JNB TF0, OverBit
   CLR TR0
   CLR TF0
   DJNZ R5, DELAY
   MOV P1, A
   RL A
END
```

```
                LED0 = LED1 = LED2 =
LED3 = LED4 = LED6 = LED7 = tat;
                break;
           case 7:
                LED6 = sang;
                LED0 = LED1 = LED2 =
LED3 = LED4 = LED5 = LED7 = tat;
                break;
           case 8:
                LED7 = sang;
                LED0 = LED1 = LED2 =
LED3 = LED4 = LED5 = LED6 = tat;
                break;
     }
}

void main (){
     unsigned char m;
     while(1){
          for(m=0;m<9; m++){
                display_LED(m);
                delay(500);
          }
     }
}
```

Conclusion: High level languages require more CPI from computers than low level languages. However, Assembly performs better than C language in the same software because we require a cross-compiler at the high level to assemble code and produce HEX files. However, in order to create large features, we need a language that is simple to comprehend and easy to code, which is C.