**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

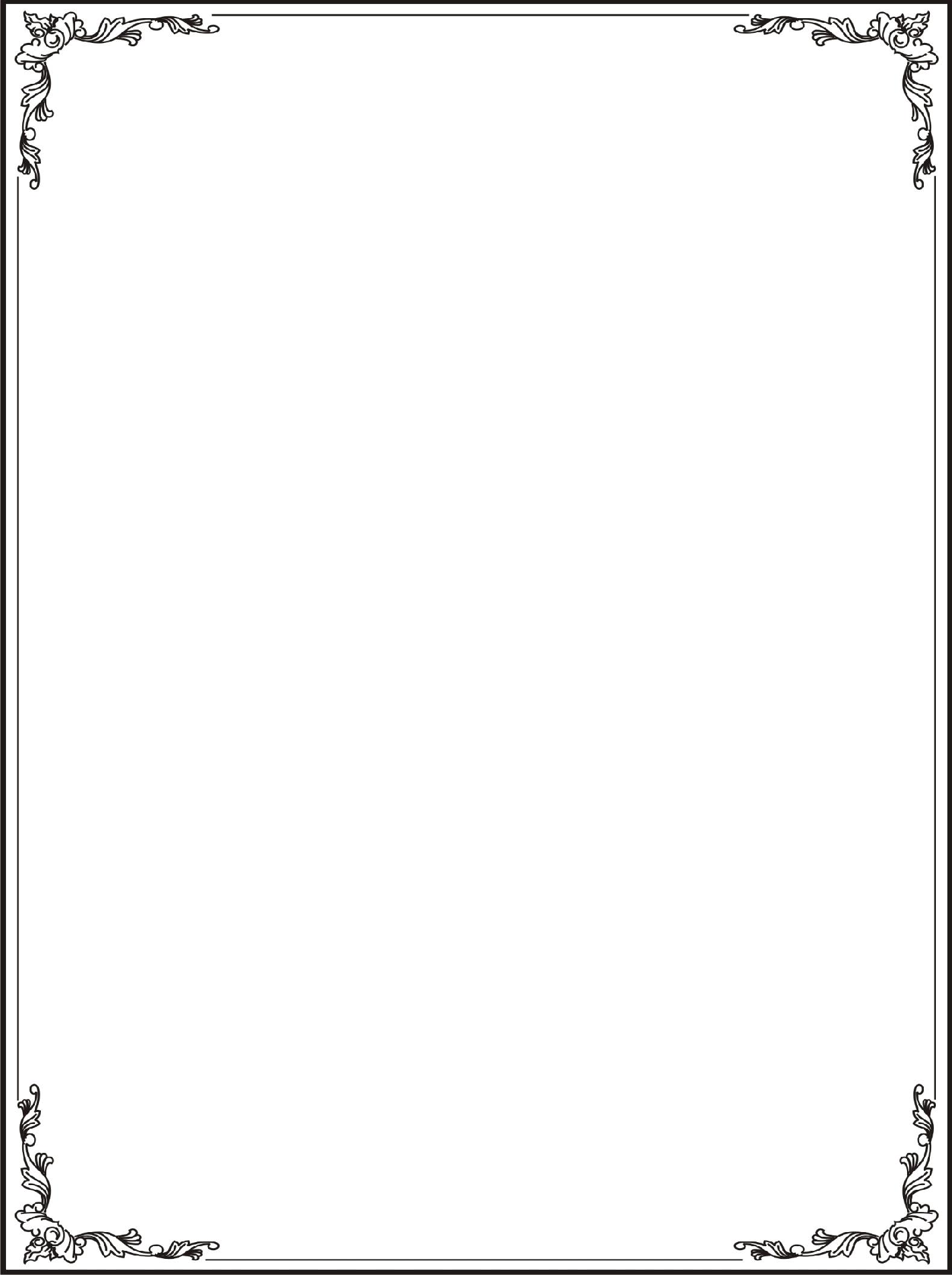
**THỰC TẬP KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH**

GVHD: Huỳnh Hoàng Hà

HTSV: Nguyễn Đức Tài

MSSV: 21139071

**TP. Hồ Chí Minh - 11/2023**



Timer 8051

1. Timer 0 và Timer 1:

* Timer 0 và Timer 1 là hai bộ đếm độc lập trong vi điều khiển 8051.
* Timer 0 thường được sử dụng để tạo ra các chức năng thời gian trong các ứng dụng, trong khi Timer 1 có thể được sử dụng cho các mục đích khác.

1. Các thanh ghi cấu hình Timer

|  |  |
| --- | --- |
| **Thanh ghi** | **Ý nghĩa** |
| TMOD | Chọn chế độ hoạt động cho Timer (Không cho truy xuất từng bit) |
| IE | Cấu hình cho phép ngắt hoạt động (Ngắt toàn cục, ngắt Timer) |
| THx, TLx | Thanh ghi chứ giá trị đếm của Timer (x=0 hoặc 1) |
| TCON | Chứa cờ báo hiệu tràn khi đếm lên, và bit cho phép Timer bắt đầu đếm |

* Nguyên lý hoạt động: Timer là 1 bộ đếm lên, với giá trị đếm được nạp vào thanh ghi THx, TLx. Khi đếm tràn thì cờ TFx bật lên 1, hàm xử lý ngắt Timer được gọi.
* Tốc độ đếm của Timer = fCrystal/12 = 12MHz/12 = 1MHz.  
  => Thời gian của 1 lần đếm (chu kì): TTimer = 1/f = 1us.

1. Cấu hình Timer:

* Chọn chế độ làm việc cho Timer (TMOD)

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

* Timer 0 hoạt động ở 4 chế độ. Có hai chế độ thông dụng là Mode 1 và Mode 2.
* Mode 1: là bộ đếm lên 16 bit. Giá trị bắt đầu đếm được nạp vào 2 thanh ghi TH0 (byte cao) và TL0 (byte thấp). Mỗi lần đếm xong (đếm lên tối đa 0xFFFF+1), cần nạp lại giá trị ban đầu cho TH0, TL0.
* Mode 2: là bộ đếm lên 8 bit, sử dụng thanh ghi TL0 để đếm lên (0xFF+1). Thanh ghi TH0 không sử dụng đến mà chỉ dùng để lưu gái trị bắt đầu đếm. Mỗi lần đếm xong, thì vi điều khiển tự động nạp lại giá trị từ TH0 sang TL0.

1. Cấu hình cho phép ngắt Timer 0(IE):

* Ngắt toàn cục (EA=1): Đây là ngắt tổng cho tất cả các ngắt ngoại vi.
* Ngắt Timer 0 (ET0=1): Đây là bit phép Timer 0 ngắt.

1. Tính giá trị nạp cho thanh ghi TH0, TL0:

Ví dụ: Xác định TH0, TL0 để tạo bộ đếm tràn sau 1ms?

T = TTimer \* (2^16 – TH0TL0)

=> TH0TL0 = 2^16 - T/TTimer = 65536 – 1ms/1us = 65536 – 1000 = 64536

=> TH0TL0 = 0xFC18   
=> TH0 = 0xFC,  TL0 = 0x18.

1. Chương trình mẫu:

* Bai1: 8 LED sáng dần thời gian khoảng 0,5s.
* Code mẫu:
* #include<reg51.h>
* #include<intrins.h>
* #define GPIO\_LED P2
* void TimerConfiguration();
* unsigned char Time;
* void main(void)
* {
* Time=0;
* GPIO\_LED=0XFE;
* TimerConfiguration();
* while(1)
* {
* if(Time==10)
* {
* GPIO\_LED=\_crol\_(GPIO\_LED,1);
* Time=0;
* }
* }
* }
* void TimerConfiguration()
* {
* TMOD = 0x01;
* TH0 = 0x3C;
* TL0 = 0xB0;
* EA = 1;
* ET0 = 1;
* TR0 = 1;
* void Timer0() interrupt 1
* {
* TH0 = 0x3C;
* TL0 = 0xB0;
* Time++;
* }
* Giải thích: chương trình cấu hình timer mỗi chu kỳ khoảng 50ms. Khi tăng lên 10 lần khoảng 500ms thì sẽ vào hàm ngắt để cấu hình lại Timer và biến Time tăng lên 1. Khi Time=10 thì sẽ gán lại bằng 0 và tiếp tục đếm.
* Mô phỏng Proteus:

Ảnh có chứa văn bản, biểu đồ, hàng, số

Mô tả được tạo tự động

* Bai2: Chương trình cài đặt thời gian bằng nút nhấn K1(chọn nơi cài đặt), K2 để tăng giá trị lên và K3 để bật/tắt Timer để bắt đầu đếm thời gian. Thời gian được hiển thị trên màn hình LCD.
* Code mẫu:
* #include<reg51.h>
* #include"lcd.h"
* sbit K1=P3^0;
* sbit K2=P3^1;
* sbit K3=P3^2;
* sbit K4=P3^3;
* unsigned char Time;
* void Delay1ms(unsigned int c);
* void TimerConfiguration();
* void Int0Configuration();
* unsigned char SetPlace;
* void main(void)
* {
* unsigned char hour,minit,second;
* unsigned int i;
* TimerConfiguration();
* Int0Configuration();
* LcdInit();
* hour=12;
* LcdWriteData('0'+hour/10);
* LcdWriteData('0'+hour%10);
* LcdWriteData('-');
* LcdWriteData('0'+minit/10);
* LcdWriteData('0'+minit%10);
* LcdWriteData('-');
* LcdWriteData('0'+second/10);
* LcdWriteData('0'+second%10);
* while(1)
* {
* if(TR0==0)
* {
* if(K1==0)
* {
* Delay1ms(10);
* if(K1==0)
* {
* SetPlace++;
* if(SetPlace>=3)
* SetPlace=0;
* }
* while((i<50)&&(K1==0))
* {
* Delay1ms(1);
* i++;
* }
* i=0;
* }
* if(K2==0)
* {
* Delay1ms(10);
* if(K2==0)
* {
* if(SetPlace==0)
* {
* second++;
* if(second>=60)
* second=0;
* }
* else if(SetPlace==1)
* {
* minit++;
* if(minit>=60)
* minit=0;
* }
* else
* {
* hour++;
* if(hour>=24)
* hour=0;
* }
* }
* while((i<50)&&(K2==0))
* {
* Delay1ms(1);
* i++;
* }
* i=0;
* }
* }
* if(Time>=20)
* {
* Time=0;
* second++;
* if(second==60)
* {
* second=0;
* minit++;
* if(minit==60)
* {
* minit=0;
* hour++;
* if(hour==24)
* {
* hour=0;
* }
* }
* }
* }
* LcdWriteCom(0x80);
* LcdWriteData('0'+hour/10);
* LcdWriteData('0'+hour%10);
* LcdWriteCom(0x83);
* LcdWriteData('0'+minit/10);
* LcdWriteData('0'+minit%10);
* LcdWriteCom(0x86);
* LcdWriteData('0'+second/10);
* LcdWriteData('0'+second%10);
* }
* }
* void Delay1ms(unsigned int c)
* {
* unsigned char a,b;
* for (; c>0; c--)
* {
* for(b=199;b>0;b--)
* {
* for(a=1;a>0;a--);
* }
* }
* } Hàm tạo độ trễ 1ms.
* void TimerConfiguration()
* {
* TMOD = 0x01;
* TH0 = 0x3C;
* TL0 = 0x0B0;
* EA = 1;
* ET0 = 1;
* TR0 = 1;
* } Hàm này để cấu hình cho Timer.
* void Timer0() interrupt 1
* {
* TH0 = 0x3C;
* TL0 = 0x0B0;
* Time++;
* } Khi timer hoàn thành 1 chu kỳ đếm thì đến hàm ngắt này để cấu hình lại timer.
* void Int0Configuration()
* {
* IT0=1;
* EX0=1;
* EA=1;
* } Hàm này cấu hình ngắt ngoại vi 0.
* void Int0() interrupt 0
* {
* Delay1ms(10);
* if(K3==0)
* {
* TR0=~TR0;
* SetPlace=0;
* }
* } Khi nút K3 được nhấn hàm này đảo ngược trạng thái của Timer và đặt “SetPlace” về 0 để xác định thiết lập giờ.
* Giải thích Code: Khi bắt đầu chương trình đầu tiên hiển thị 12h00p00s trên LCD. Nếu mà K3 được nhấn thì ‘TR0=0’ khi đó mỗi lần nhấn K1 thì sẽ chọn chỗ để thiết lập thời gian tương ứng với biến “SetPlace” nếu SetPlace >= 3 thì sẽ về 0. SetPlace =0 thì chỉnh giây, SetPlace=1 thì chỉnh phút, SetPlace=2 thì chỉnh giờ. Khi nhấn K2 tùy theo SetPlace thì các biến hour, minit, second sẽ tăng lên 1 theo biến SetPlace tương tứng. Khi ‘TR0=1’ thì Timer sẽ được chạy nếu đủ 20 lần thì biến second sẽ tăng lên 1. Các biến minit, second khi được tăng đến 60 thì được quay về 0. Biến hour tăng đến 24 thì quay về 0. Thanh ghi TH0=0x3C, TL0=0x0B0 thì thời gian đếm của timer khoảng 50ms. 50ms x 20 lần = 1s.

1. Mô phỏng trên proteus:

* Khi trương trình đếm bình thường:

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

* Khi dừng việc đếm bằng K3 và điều chỉnh thời gian bằng các nút K1, K2

Ảnh có chứa văn bản, biểu đồ, số, hàng

Mô tả được tạo tự động