

HỆ NHÚNG

Ngô Lam Trung

Bộ môn Kỹ thuật Máy tính

Viện CNTT&TT- ĐHBK HN

Chương 3: Ghép nối ngoại vi

2.1 Ghép nối cổng vào ra song song

2.2 Ghép nối ngắt ngoài

2.3 Ghép nối nút bấm, bàn phím

Ghép nối cổng vào ra song song

- 8051 có 4 cổng vào ra GPIO (mỗi cổng 8 bit): P0, P1, P2, P3
- Sau khi reset, các cổng ở chế độ mặc định là cổng ra (output)
- Để các cổng/chân làm việc ở chế độ cổng/chân vào (input) phải tiến hành ghi các bit 1 ra các cổng/chân tương ứng

8051 ko có set 1 cổng vào ra 1 cách tường minh

- Ví dụ: `MOV P1,#0FF`; Cổng 1 thành cổng vào
`SETB P1.0` ; Chân P1.0 làm chân vào
`MOV P1,#03` ; Chân P1.0 và P1.1 làm chân vào
; các chân còn lại làm chân ra

Xuất dữ liệu ra cổng/chân

■ Xuất dữ liệu ra cổng ra

MOV **tên_cổng**, giá trị

• Ví dụ:

✓ MOV P1, #55h

■ Xuất dữ liệu ra từng chân

• Đưa chân cổng lên mức cao:

SETB **bit**

Ví dụ: SETB P1.0

bit 0 ở cổng P1

• Đưa chân cổng xuống mức thấp:

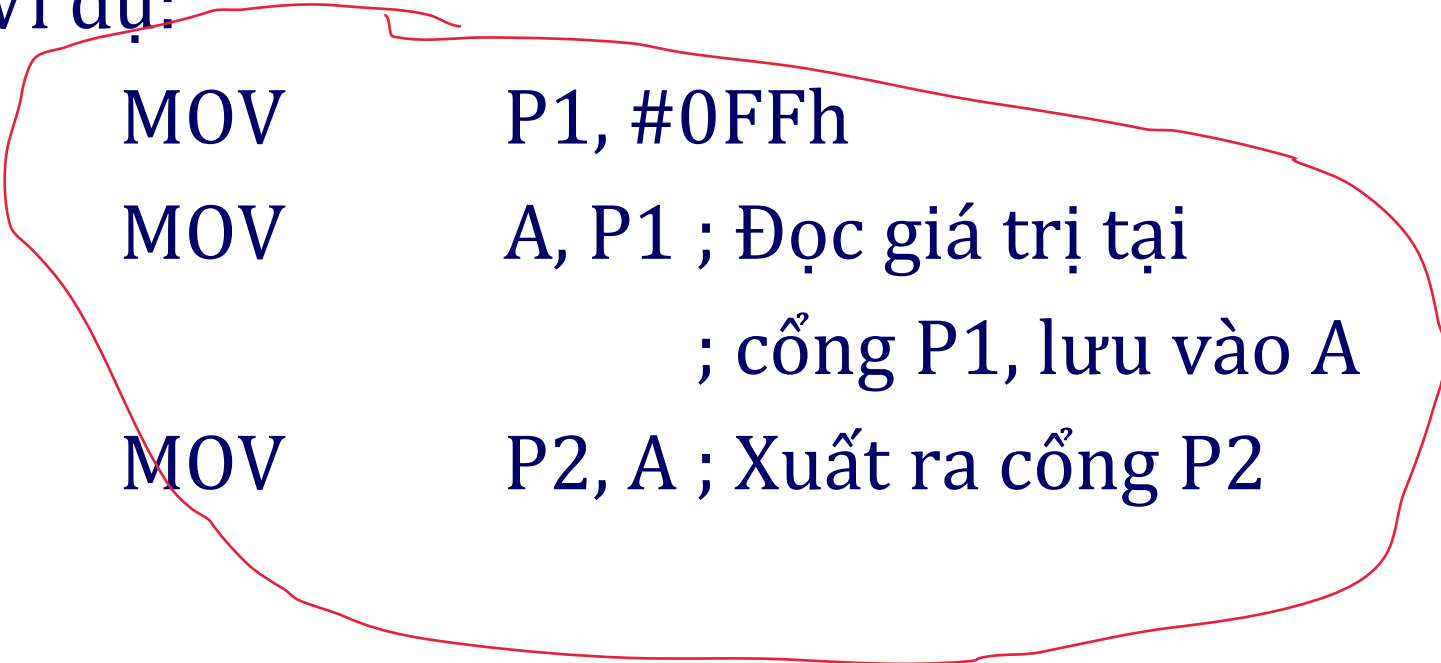
CLR **bit**

Ví dụ: CLR P1.0

Đọc dữ liệu từ cổng vào

- Bước 1: Thiết lập cổng làm việc ở chế độ input
- Bước 2: Đọc dữ liệu từ chân cổng

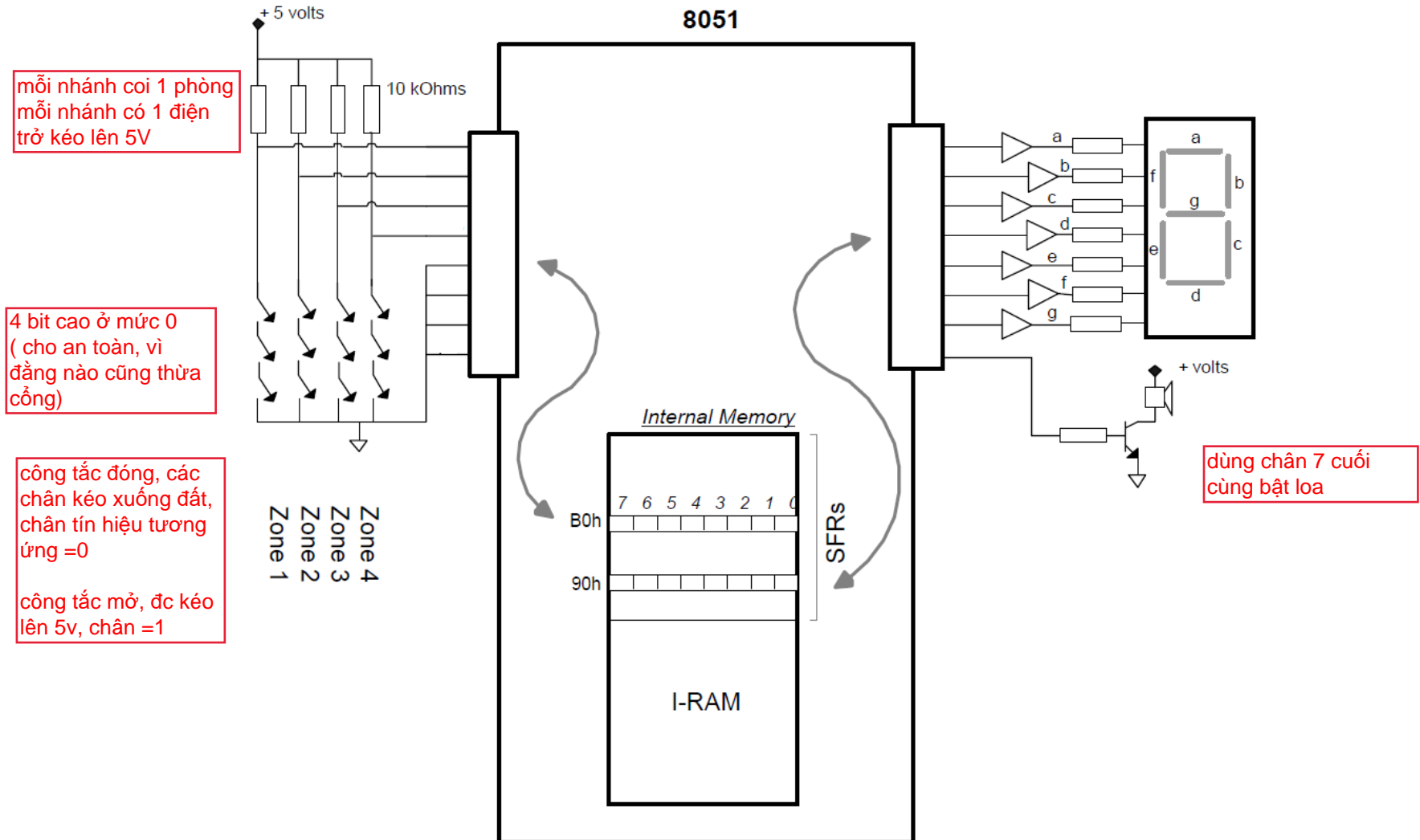
Ví dụ:



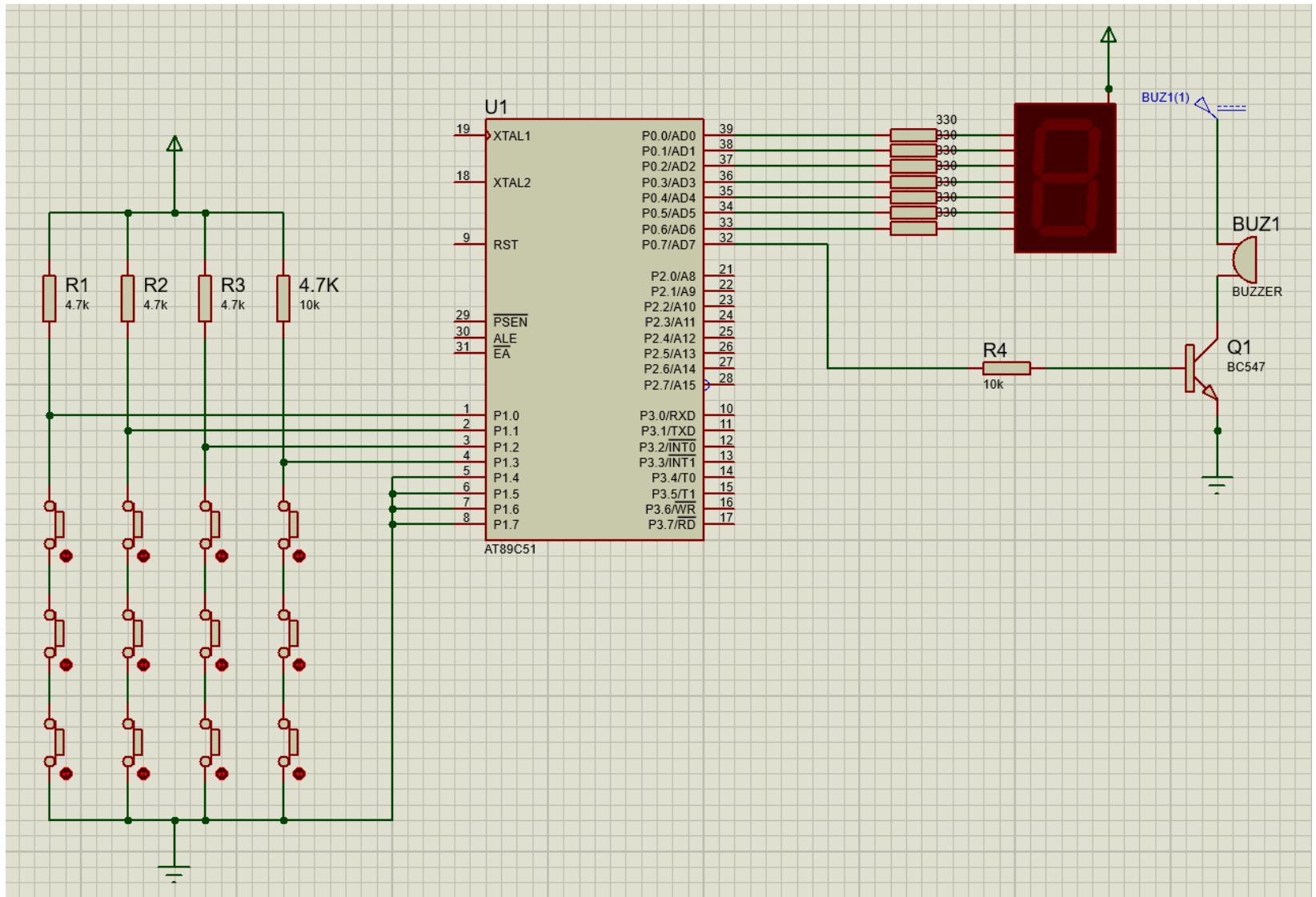
```
MOV      P1, #0FFh
MOV      A, P1 ; Đọc giá trị tại
              ; cổng P1, lưu vào A
MOV      P2, A ; Xuất ra cổng P2
```

- Thiết kế mạch phát hiện đột nhập dùng công tắc đóng/mở
 - Nhà có 4 khu vực, mỗi khu vực có 3 công tắc lắp nối tiếp.
 - Tủ trung tâm lắp 1 chuông báo động, và một đèn LED 7 thanh báo vị trí khu vực phát hiện đột nhập

Sơ đồ khối



Maç Proteus



Code version 1: phát hiện đột nhập

ORG 0000h

MOV P1, #0ffh ;set P1 as input

MOV P0, #00 ;initialize P0

POLL:

MOV A, P1 ;read sensors

CJNE A, #00h, ALARM ;fire alarm if detected

LJMP POLL ;otherwise polling sensor

ALARM:

SETB P0.7 ;turn on buzzer

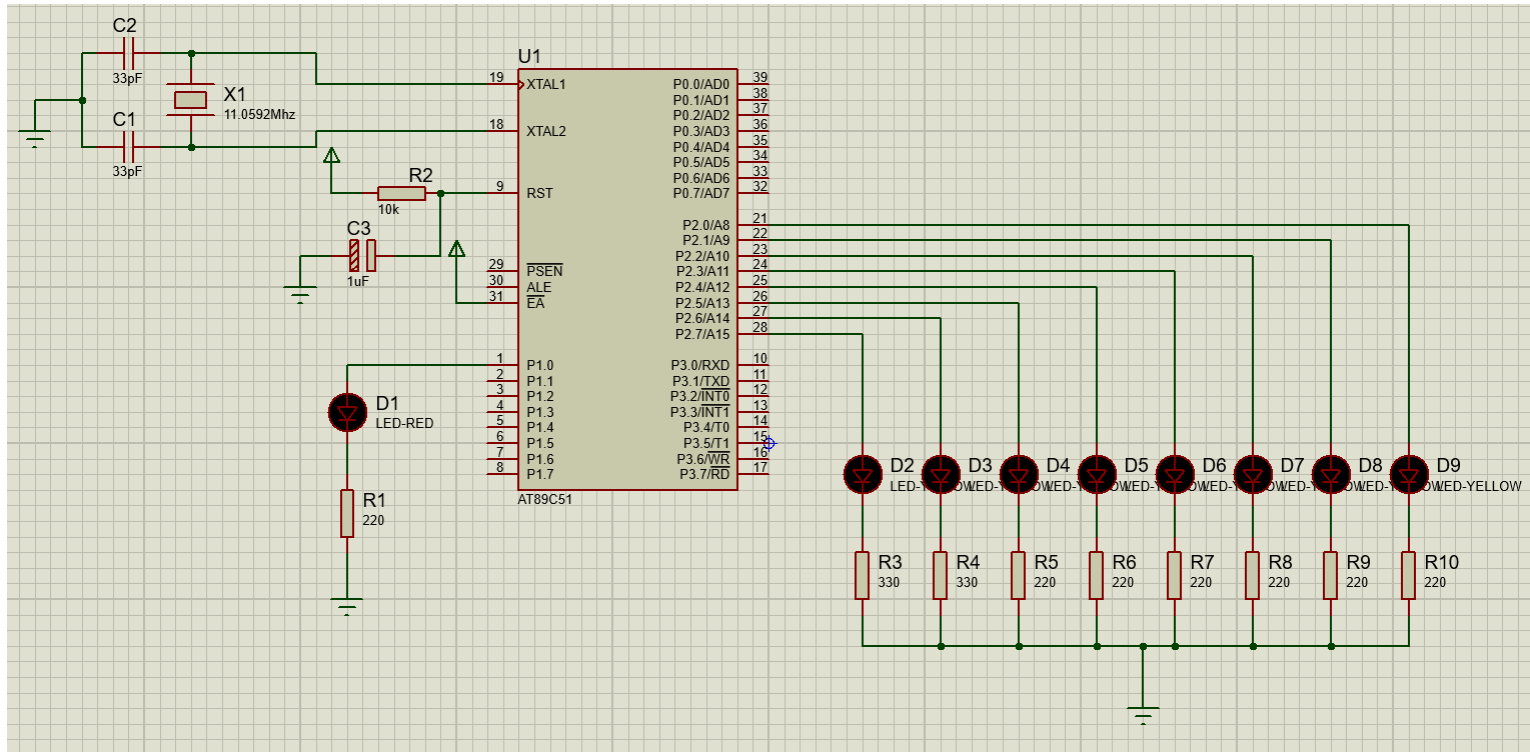
END_LOOP:

LJMP END_LOOP ;final endless loop

END

Bài tập: Hiển thị vị trí có đột nhập

- Lập trình cho dãy đèn LED sáng lần từ trái sang phải



Code version 1: Delay 1ms routine

thời gian
thực thi (us)

để chạy đc 1ms thì 8051 chạy ở 12HZ

ONE_MILLI_SUB:

2 PUSH 07h

2 MOV R7, #250d

LOOP_1_MILLI:

1 NOP No operation

1 NOP

2 DJNZ R7, LOOP_1_MILLI

2 POP 07h

2 RET

; save R7 to stack

; ctr con đặt R7=250

; loops 250 times

; two NOPs

;

; loop until zero

; restore R7

; return to caller

cất thanh ghi R7 vào Stack
(Cất R7 đi để có thể R7 đc sd
ở ctr con và ctr chính)

$(1+1+2) \cdot 250$
1000 us

lấy ra, thì sau khi vào ctr con và ra ngoài
ctr con ko bị thay đổi dù ctr con
ONE_MILLI thay đổi nó

1008 us

Bài tập: tính thời gian delay chính xác

- CPU 8051 chạy ở tần số 12 MHz
→ Chu kỳ lệnh là 1us

Bài tập: điều khiển LED

3.2. Lập trình ngắt với 8051

8051 có 5 nguồn
ngắt lập trình
được
(INT0, INT1,
TF0, TF1, TI, RI)

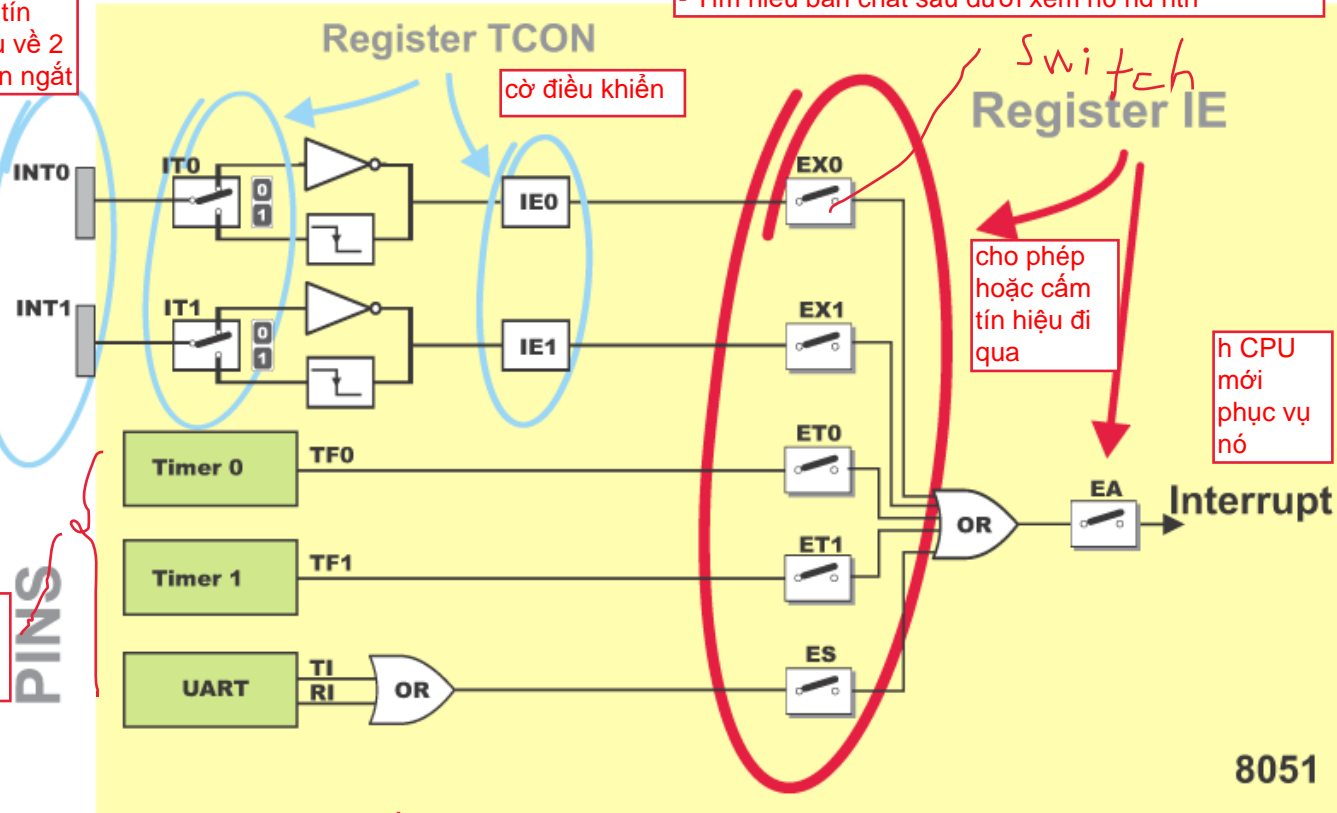
ngắt ngoài 0,1
ngắt timer số 0,1
ngắt bộ ngoài ra: truyền, nhận

3 nguồn ngắt ko đến từ
tín hiệu ngoài mà từ
ngoại vi bên trong

gửi tín
hiệu về 2
chân ngắt

cờ điều khiển

- Để thực hiện nhiều ctr cùng 1 lúc
- Tìm hiểu bản chất sâu dưới xem nó hỡ ntn



Cấu hình ngắt sử dụng thanh ghi IE (Interrupt Enable)

	0	X	0	0	0	0	0	0	Value after Reset
IE	EA		ET2	ES	ET1	EX1	ET0	EX0	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

chỉ hỡ trên 89C52 có
Thêm timer2

Các ngắt của 8051

- Các nguồn ngắt lập trình được
 - 2 ngắt cho bộ đếm và định thời
 - 2 ngắt ngoài (INT0, INT1)
 - 1 ngắt cho bộ truyền thông nối tiếp UART
- Bảng vector ngắt của 8051

nguồn ngắt 6, nhưng ko lập trình được

địa chỉ nhảy vào

Ngắt	Địa chỉ ROM (Hexa)	Chân
RESET	0000	9
Ngắt phần cứng ngoài (INT0)	0003	12 (P3.2)
Ngắt bộ Timer 0 (TF0)	000B	
Ngắt phần cứng ngoài 1 (INT1)	0013	13 (P3.3)
Ngắt bộ Timer 1 (TF1)	001B	
Ngắt COM nối tiếp (RI và TI)	0023	

Lập trình xử lý ngắt

■ Trình phục vụ ngắt:

con pic cũng có ngắt nhưng ko từng
minh như 8051

- Mỗi ngắt luôn có một trình phục vụ ngắt
- Khi một ngắt được kích hoạt thì vi điều khiển thực thi trình phục vụ ngắt
- Trình phục vụ ngắt của mỗi ngắt có một vị trí cố định trong bộ nhớ
- Tập hợp các ô nhớ lưu giữ địa chỉ của tất cả các trình phục vụ ngắt gọi là bảng vector ngắt (Interrupt Table)

Trình tự phục vụ ngắt của 8051

khi lệnh ngắt xảy ra

1. Kết thúc lệnh hiện tại, lưu địa chỉ của lệnh kế tiếp (PC) vào ngăn xếp
ngăn xếp: vài byte
ngắt lồng nhiều quá => Tràn stack => nhận lại ko là mã lệnh => Treo ctr
2. Nhảy đến vị trí cố định trong bảng vector ngắt
3. Nhận địa chỉ của trình phục vụ ngắt, nhảy tới địa chỉ đó và bắt đầu thực thi chương trình con phục vụ ngắt cho đến lệnh cuối cùng là lệnh RETI
4. Kết thúc chương trình con phục vụ ngắt, bộ vi điều khiển trở về thực thi tiếp lệnh nơi nó đã bị ngắt
lệnh đã đc lưu ở ngăn xếp

Trình tự phục vụ ngắt

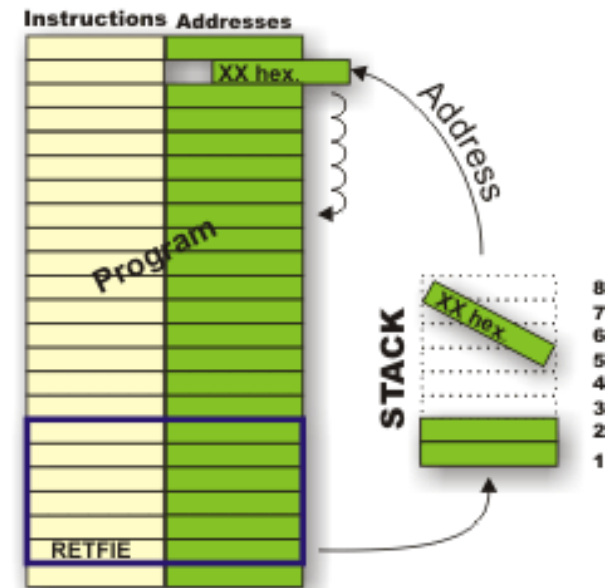
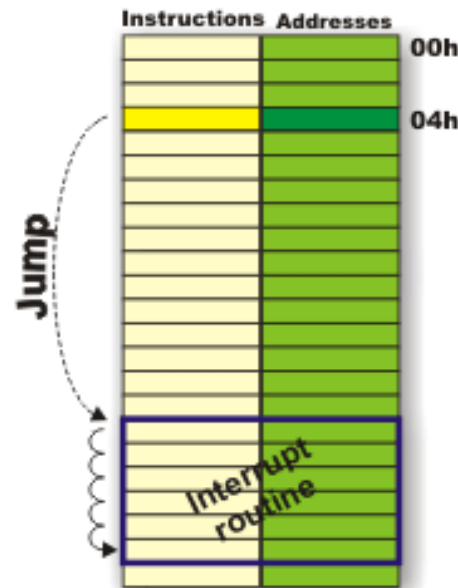
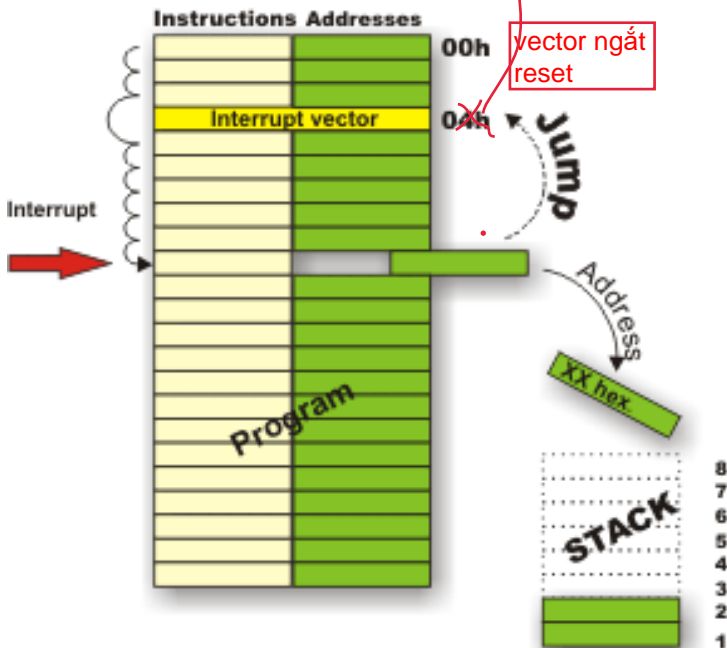
thời điểm 1: Có 1 ngắt xảy ra
- gtr địa chỉ vào stack
- Thanh ghi PC trở vào vector ngắt

địa chỉ ở vector ngắt nhảy đến ctr con phục vụ ngắt (vì nếu ko Jump vị trí đó chỉ phục vụ đc 1 ctr con dài 8 word)

1

2

3



địa chỉ của ngắt:
Trang 16

cuối ctr con bh cũng có
RETIE => lệnh này lấy
đỉnh stack ra

Nguồn ngắt ngoài

- Ngắt phần cứng ngoài (0 và 1)
 - Tín hiệu yêu cầu ngắt được gửi đến chân **INT0** (P3_2) và chân **INT1** (P3_3)
phải đặt các chân này ở chế độ input và ko đc gán gtr khác cho nó nữa
 - 8051 có thể nhận ngắt theo 2 kiểu
 - ✓ Ngắt theo mức thấp: ngắt được kích hoạt khi có tín hiệu mức thấp đưa đến chân ngắt (**Đây là chế độ mặc định**)
ngắt khi bằng =0, ngắt kéo dài
 - ✓ Ngắt theo sườn âm: ngắt được kích hoạt khi có sườn âm (chuyển từ mức cao xuống mức thấp) đưa đến chân ngắt
ngắt khi có 1 sự kiện chạy từ 1 xuống 0, ngắt ở 1 thời điểm ngắn
 - Thiết lập chế độ kích hoạt ngắt qua thanh ghi **TCON** (**bit IT0 cho INT0 và IT1 cho INT1**)

Thanh ghi TCON

D7				D0			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Cờ tràn của Timer 1, được thiết lập bởi phần cứng khi bộ đếm/bộ định thời 1 tràn, và được xoá bởi phần cứng khi bộ xử lý nhảy đến trình phục vụ ngắt.					
TR1	TCON.6	Bit điều khiển hoạt động của Timer 1, được thiết lập và xoá bởi phần mềm để bật/tắt bộ đếm/bộ định thời 1.					
TF0	TCON.5	Tương tự như TF1 nhưng là cho Timer 0.					
TR0	TCON.4	Tương tự như TR1 nhưng là cho Timer 0.					
IE1	TCON.3	Cờ ngắt ngoài 1 kích phát sườn, được CPU thiết lập khi phát hiện có sườn xuống ngắt ngoài và được CPU xoá khi ngắt được xử lý. Lưu ý: Cờ này không chốt ngắt kích phát mức thấp.					
IT1	TCON.2	Bit điều khiển kiểu ngắt 1 (Interrupt 1 Type Control Bit) được thiết lập và xoá bởi phần mềm để xác định kiểu ngắt ngoài kích phát sườn xuống hay mức thấp.					
IE0	TCON.1	Tương tự như IE1 nhưng là cho ngắt ngoài 0.					
IT0	TCON.0	Tương tự như bit IT1 nhưng là cho ngắt ngoài 0.					

xã ngắt
theo
mức hay
sườn

code mẫu
- vector ngắt
- Cách xử lí ngắt ngoài

Lập trình với ngắt ngoài

;Interrupt table

```
ORG 0000h          ; entry address for 8051 RESET
    LJMP MAIN       ; move MAIN away from interrupt vector table
ORG 0003h          ; vector address for interrupt 0
    LJMP ISR0       ; jump to ISR0
ORG 0013h          ; vector address for interrupt 1
    LJMP ISR1       ; jump to ISR1
ORG 0100h          ; MAIN starts here
```

đặt địa chỉ hàm main ở đó
(ở 256=0100h)

Bảng vector ngắt

MAIN:

```
    MOV IE, #10000101B ; enable external interrupts IE0, IE1
    SETB IT0           ; negative edge trigger for interrupt 0
    SETB IT1           ; negative edge trigger for interrupt 1
    SETB P1.0          ; LED ON
```

LOOP:

```
LJMP LOOP           ; end loop
```

Lập trình với ngắt ngoài (tiếp)

```
=====
; ISR0 code mẫu ctr con phục vụ ngắt số 0
;
=====
ISR0:
    SETB P1.0                ; LED ON
    RETI                       ; return from interrupt
;
=====
; ISR1
;
=====
ISR1:
    CLR P1.0                ; LED OFF
    RETI                       ; return from interrupt
END                           ; end of program
```