CHƯƠNG 3: HỌ VI ĐIỀU KHIỂN 8051

GV: LÊ PHƯỢNG QUYÊN
BỘ MÔN: ĐIỆN TỬ VIỄN THÔNG
KHOA: ĐTVT - DTU

GIỚI THIỆU

Các vi bộ điều khiển và các bộ vi xử lý có ba điểm khác nhau chính sau:

- ✓ **Kiến trúc phần cứng**
- ✓ **Phạm vi ứng dụng**
- ✓ **Đặc điểm tập lệnh**

Các họ vi điều khiển phổ biến

- ✓ **Intel 8051**
- ✓ **PIC16C5x**
- ✓ **Motorola MC68xxx**
- ✓ **AVR**
- ✓ **PSoC**

VI ĐIỀU KHIỂN 8051

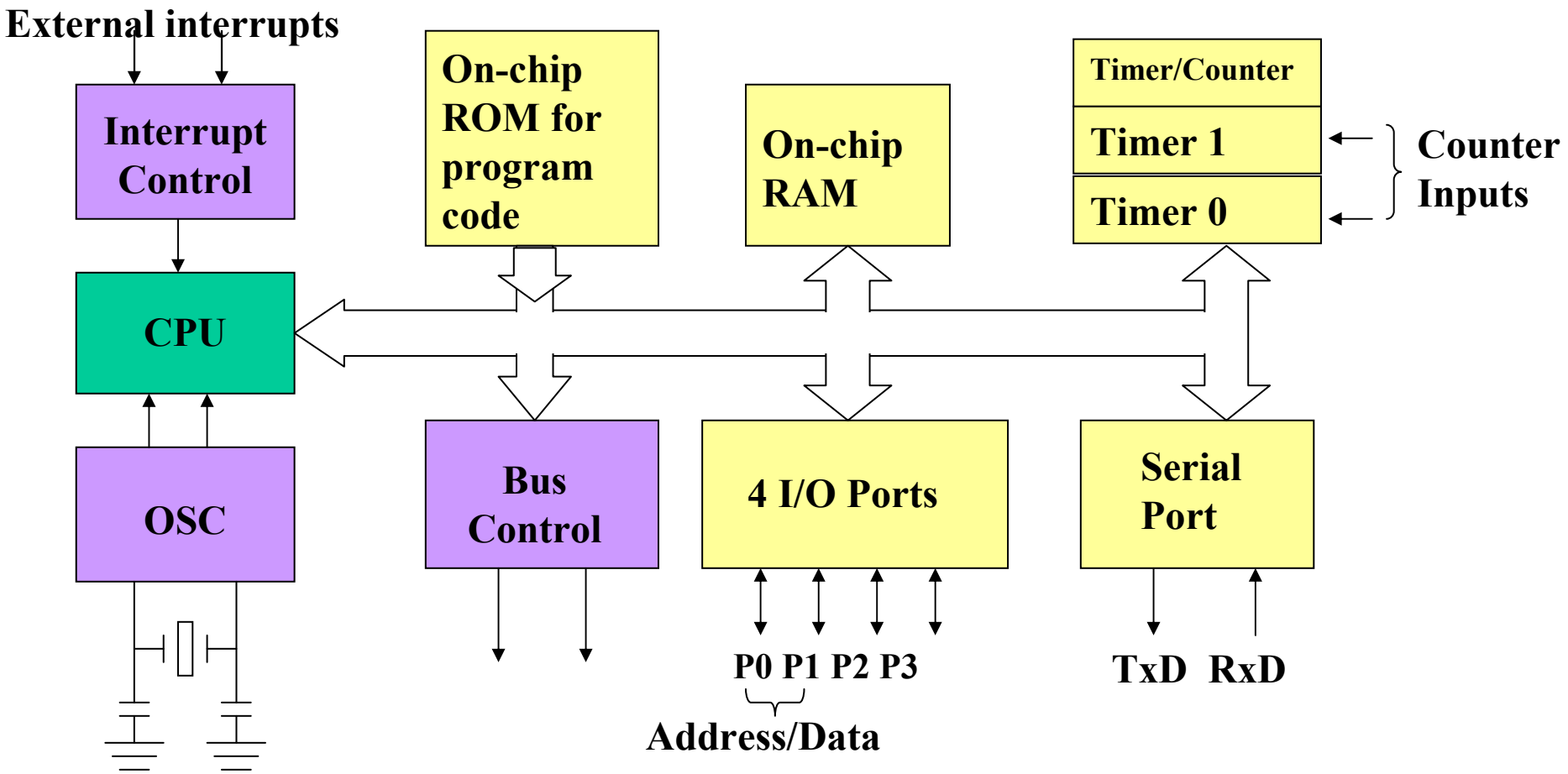
8051 thuộc họ vi điều khiển MCS-51.

MCS-51 được phát triển bởi Intel và các nhà sản xuất khác (như Siemens, Philips) là các nhà cung cấp đứng thứ hai của họ này.

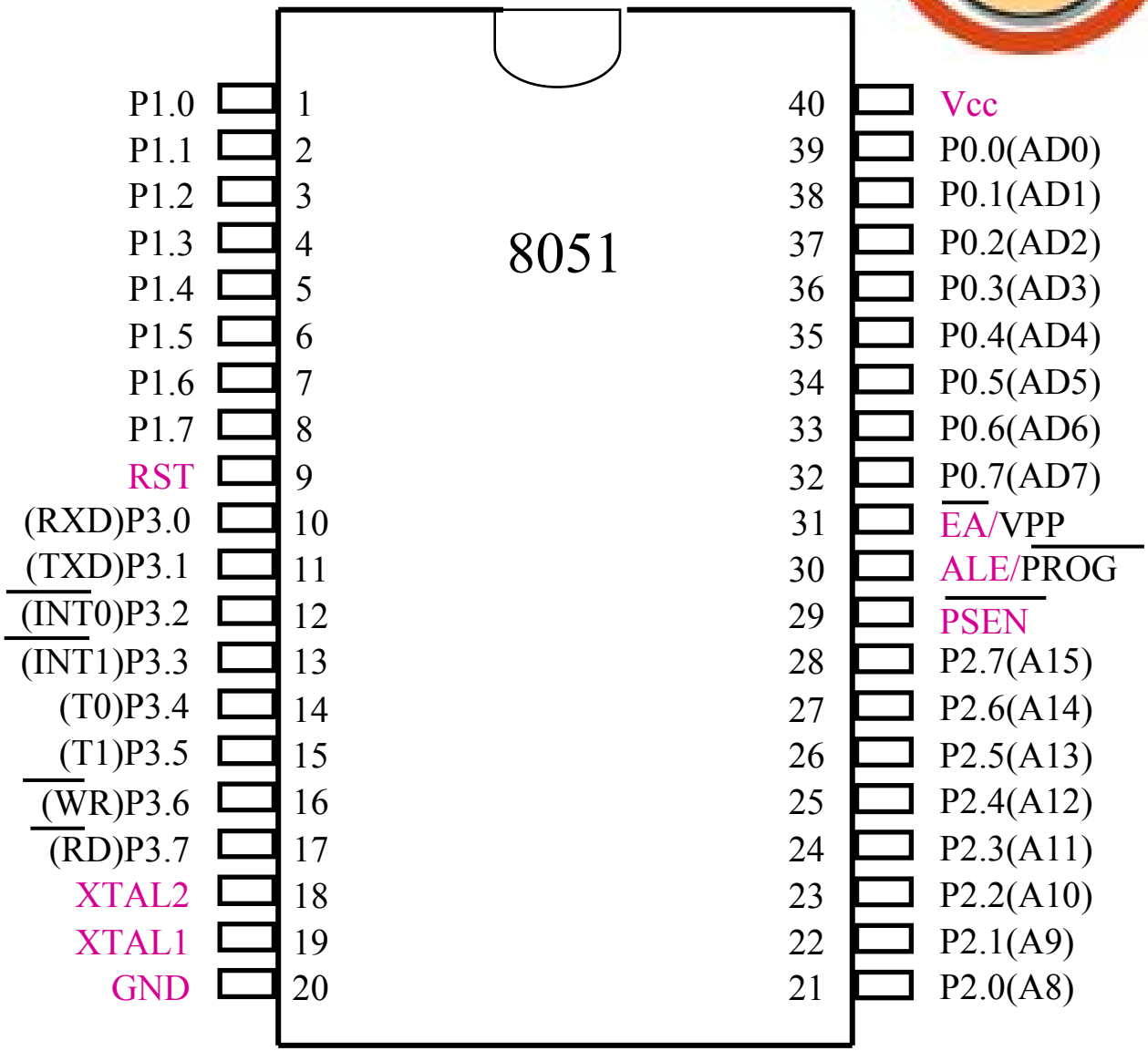
Tóm tắt một số đặc điểm chính của họ 8051:

- **4K bytes ROM trong**
- **128 bytes RAM trong**
- **4 cổng I/O 8-bit**
- **2 bộ định thời 16 bit**
- **Giao diện nối tiếp**
- **Quản lý được 64K bộ nhớ code bên ngoài**
- **Quản lý được 64K bộ nhớ dữ liệu bên ngoài**

SƠ ĐỒ KHỐI 8051



CÁC CHÂN CỦA 8051



CÁC CỔNG I/O

- 4 CỔNG I/O

Port 0 (chân 32-39) : P0 (P0.0 … P0.7)

Port 1 (chân 1-8) : P1 (P1.0 … P1.7)

Port 2 (chân 21-28) : P2 (P2.0 … P2.7)

Port 3 (chân 10-17) : P3 (P3.0 … P3.7)

- Mỗi cổng có 8 chân:

Đánh tên P0.X (X=0,1,...,7), P1.X, P2.X, P3.X

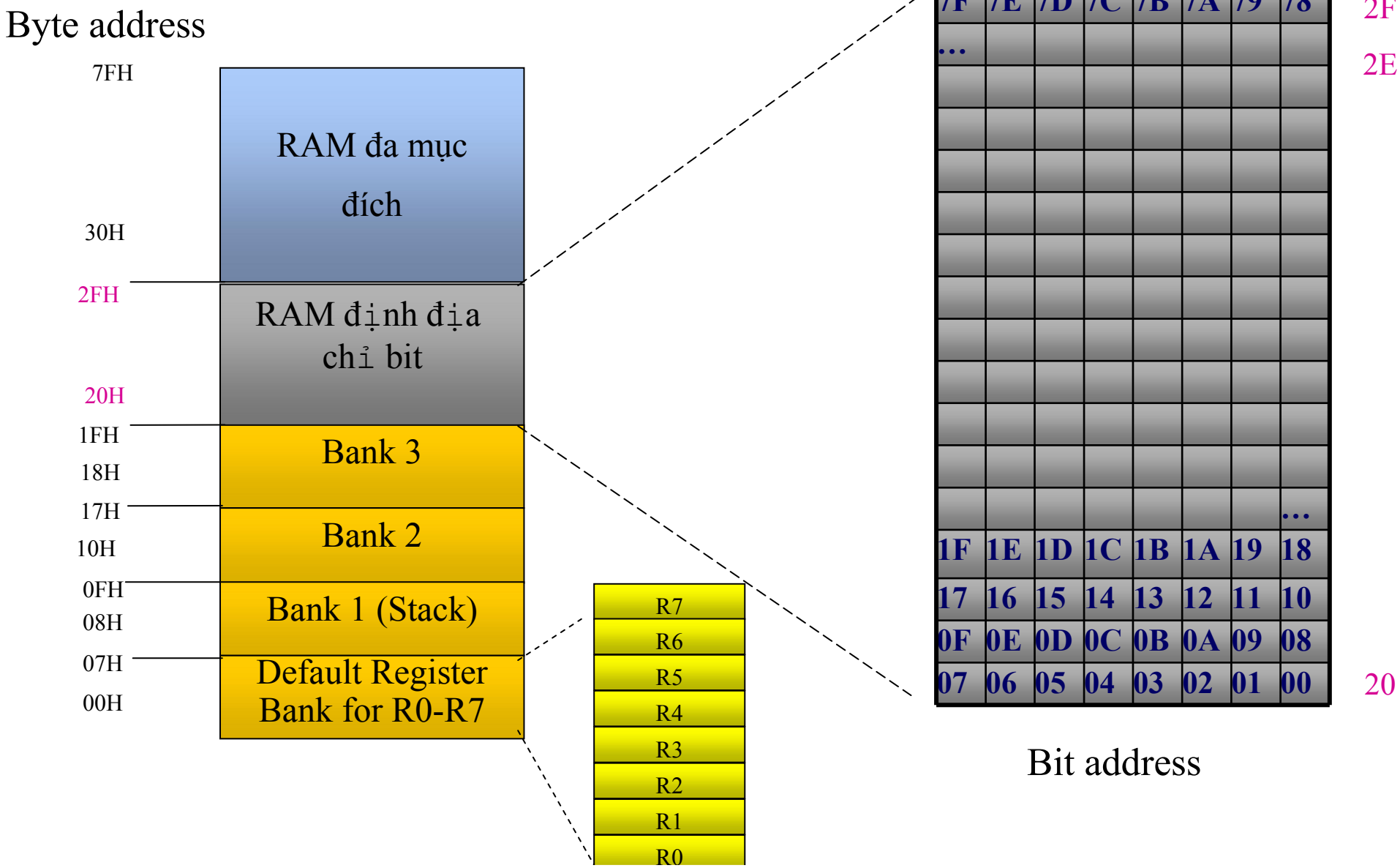
Ex:P0.0 là bit 0 (LSB) của P0

Ex:P0.7 là bit 7 (MSB) của P0

8 bits này cấu thành 1 byte

Mỗi cổng có thể được dùng như input hay output

KHÔNG GIAN BỘ NHỚ RAM NỘI



CÁC CHẾ ĐỘ ĐỊA CHỈ

Các kiểu định địa chỉ cho phép xác định nguồn và đích của dữ liệu theo nhiều cách khác nhau tùy tình huống lập trình

- **Định địa chỉ tức thời :**

  MOV A, #65H ; đưa giá trị 65h vào thanh ghi A

  → A=65h

- **Định địa chỉ thanh ghi**

  MOV R1, A ; đưa dữ liệu trong thanh ghi A vào thanh ghi R1
  → R1=A

- **Định địa chỉ trực tiếp**

  MOV R0, 40H ; lấy dữ liệu tại địa chỉ 40h đưa vào thanh ghi R0

- **Định địa chỉ gián tiếp**

  MOV A, @R1 ; lấy dữ liệu tại địa chỉ được chứa trong thanh ghi R1 đưa vào thanh ghi A

CÁC THANH GHI

80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

Blue background are I/O port SFRs
Yellow background are control SFRs
Green background are other SFRs

THANH GHI CỔNG I/O

- 8051 có 4 thanh ghi cổng I/O : P0, P1, P2, P3
- Tất cả port đều được định địa chỉ bit
- Để truy cập đến các bit ta dùng cú pháp sau :

<Port name> . <bit>

Ví dụ để xóa bit 4 của cổng P0

CLR P0.4

TẬP LỆNH

➢ Lệnh chuyển dữ liệu:

MOV dest, source ; dest = source

Ví dụ : MOV A, #72H ;A=72H

MOV A, R4 ;A=62H

MOV B, 7EH

MOV P1, A ;mov A to port 1

Chú ý :

➢ **MOV A,#72 ≠ MOV A,72**

➢ **MOV DPTR, A**

➢ **MOV #30,A**

➢ **dest và source không được đồng thời là Rn**

MOV R1,R2

TẬP LỆNH

- Truy cập bộ nhớ ROM ngoài

  MOVC A, @A + DPTR

  MOVC A, @A + PC

- Truy cập bộ nhớ RAM ngoài

  MOVX A, @Ri **MOVX A, @DPTR**

  MOVA @Ri, A **MOVX @DPTR, A**

- Tráo đổi dữ liệu

  XCH A, Rn **XCH A, mem** **XCH A, @Ri**

Thao tác ngăn xếp

PUSH mem **POP mem**

TẬP LỆNH CƠ BẢN

- **Lệnh di chuyển dữ liệu MOV**
  - **MOV ĐÍCH,NGUỒN**
- **Ví dụ:**

  MOV A,#55H ; Nạp giá trị 55h vào thanh ghi A

  MOV R0,A ; Sao nội dung trong thanh ghi A vào R0

  MOV R2,#20 ;Nạp giá trị 20 vào thanh ghi R2
- **Lưu ý: Giá trị nạp vào thanh ghi không được quá 256**

  VD: MOV R2,#400 ; Câu lệnh sai

TẬP LỆNH CƠ BẢN

- **Lệnh cộng ADD:**
  - **ADD A,NGUỒN**
  - **Lệnh này ảnh hưởng đến cờ CY,PF,AC**
- **Ví dụ: ADD A,R2;A=A+R2**

  ADD A,3FH ; Cộng A với ô nhớ có địa chỉ 3FH kết quả cất vào A.

  ADD A,@R1 ; Cộng A với ô nhớ có địa chỉ bằng giá trị cất trong R1, kết quả lưu ở A.

  ADD A,#3FH ;Cộng A với hằng số 3FH, kết quả lưu ở A.
- **Lưu ý: Trong phép cộng ADD A,@thanh_ghi chỉ có thanh ghi R1 và R0 là được phép.**

TẬP LỆNH CƠ BẢN

- **Lệnh trừ SUBB: Trừ các số không dấu**
  - **SUBB A,NGUỒN ; A=A-nguồn-CY**
- **Khi CY=0; Các bước tiến hành lệnh như sau:**
  - **Lấy bù 2 của số trừ**
  - **Cộng số vừa tìm được**
  - **Đảo có nhớ**
- **Ví dụ:**

```
CLR C
MOV A,#3FH
MOV R3,#23H
SUBB A,R3
3FH        0011 1111        0011 1111
23H        0010 0011        1101 1101  (BÙ 2)
                          1 0001 1100
```

Kết quả CY=0, AC=0

TẬP LỆNH CƠ BẢN

- Khi CY=1; Lệnh này được dùng trừ các số nhiều byte và có mượn. Nếu CY=1 trước khi thực hiện lệnh SUBB thì kết quả sẽ trừ đi 1
- Ví dụ:

```
CLR C
MOV A,#6
SUBB A,#96H  ;62H-96H=CCH
MOV R7,A
MOV A,#27H
SUBB A,#12H  ; 27H-12H-1=14H
```

TẬP LỆNH CƠ BẢN

- **Lệnh nhân MUL:nhân hai số không dấu**
  - **MUL AB ; kết quả lưu vào 2 thanh ghi, A chứa byte trọng số thấp, B chứa byte trọng số cao**
- **Ví dụ:**
  - **MOV A,#25H**
    MOV B,#65H
    MUL AB ;25H*65H=E99H Với B=0EH A=99H
- **Lệnh chia DIV: 2 số 8 bit**
  - **DIV AB ; phần thương lưu ở A và phần dư lưu ở B**
- **Ví dụ:**

```
MOV B,#20
DIV AB
```

  - **Lưu ý:**
    - **Lệnh này luôn xóa các cờ CY=0 và OV=0 nếu B≠0**
    - **Nếu B=0 thì OV=1 báo lỗi và CY=0**

TẬP LỆNH CƠ BẢN

- **Viết chương trình để nhập dữ liệu dạng HEXA trong phạm vi 00-FFH từ cổng P2 và chuyển đổi sang dạng thập phân. Lưu ý các số vào các thanh ghi R5,R6,R7 trong đó số có nghĩa nhỏ nhất là R5.**

Lệnh số học

- Lệnh INC: tăng giá trị lên một đơn vị.
- Nếu trước khi tăng là 0FFH sau khi tăng là 00H
  - INC BYTE
  - VD: INC A
  - INC R2
- Lệnh INC DPTR ;tăng thanh ghi 16 bit lên 1 đơn vị

KIỂU DỮ LIỆU VÀ CHỈ DẪN

- Chỉ dẫn định nghĩa kiểu byte: DB
  - DATA1: DB 28 ; số thập phân
  - DATA2: DB 010101B ; số nhị phân
  - DATA3: DBB 39H ; mã Hexa
  - DATA DB “2345” ; mã ASCII
- ORG: được dùng để bắt đầu địa chỉ của chương trình.
  - ORG 300H
  - ORG 20

KIỂU DỮ LIỆU VÀ CHỈ DẪN

- **EQU; dùng để định nghĩa một hằng số**
  - **Count EQU 25**
  - **MOV R2,#COUNT ; R2=25**

- **END: Báo cho trình dịch biết mã nguồn đã kết thúc**

LỆNH LOGIC

- **Lệnh AND:**
  - Cú pháp: ANL đích,nguồn
  - Thực hiện phép AND logic, không ảnh hưởng đến cờ
  - Ví dụ: ANL A,#0FH
    ANL A,R1
    ANL A,@R1 ;
- **Lệnh OR:**
  - Cú pháp: OR đích, nguồn
  - Thực hiện OR logic 2 toán hạng, kết quả lưu ở toán hạng đích
  - Ví dụ: MOV A,#55H
    OR A,#FFH

LỆNH LOGIC

- **Lệnh XOR:**
  - Cú pháp: XRL đích,nguồn
  - Thực hiện phép XOR trên hai toán hạng, đặt kết quả vào đích
  - Ví dụ: XRL A,#0FH
    XRL A,R1
    XRL A,@R1 ;
- **Lệnh CPL: lấy bù thanh ghi**
  - Cú pháp: CPL đích, nguồn
  - Thực hiện phép đảo bit 0 thành 1 và 1 thành 0(hay còn gọi là phép bù 1)
  - Ví dụ: MOV A,#55H
    CPL A

LỆNH SO SÁNH

- CJNE đích, nguồn, địa chỉ tương đối
- Lệnh này để so sánh hai toán hạng nguồn và đích, nhảy đến địa chỉ tương đối nếu hai toán hạng này không bằng nhau. Lệnh này làm thay đổi cờ CY khi giá trị đích nguồn lớn hay bé hơn.
- Đích > nguồn CY=0
- Đích < nguồn CY=1

VÒNG LẶP VÀ LỆNH NHẢY

Các lệnh nhảy có điều kiện:

LỆNH	Ý NGHĨA
JZ nhãn	Nhảy nếu A=0
JNZ nhãn	Nhảy nếu A≠0
DJNZ thanh ghi,nhãn	Giảm và nhay nếu thanh ghi khác 0
CJNE A,BYTE, nhãn	Nhảy nếu A≠0
CJNE th ghi1, th ghi 2, nhãn	Nhảy nếu th ghi1≠th ghi 2
JC nhãn	Nhảy nếu CY=1
JNC nhãn	Nhảy nếu CY=0
JB nhãn	Nhảy nếu bit =1
JNB nhãn	Nhảy nếu bit =0
JBC nhãn	Nhảy nếu bit = 1 và xóa nó

VÍ DỤ

- Giả sử có một cảm biến ở cổng P2, hãy viết chương trình đọc nhiệt độ và so sánh với giá trị 60. Dựa vào kết quả kiểm tra đặt các giá trị nhiệt độ vào các thanh ghi sau:
  - Nếu T=60 thì A=T
  - Nếu T>60 thì R1=T
  - Nếu T<60 thì R2=T

GiẢI

```
        MOV P1,#0FFH
        MOV A,P1
        CJNE A,#60,LON
        SJMP THOAT
LON:    JC NHO          ;C=1
        MOV R1,A
        SJMP THOAT
NHO:    MOV R2,A
THOAT:  END
```

LỆNH QUAY

- **Quay phải: quay các bit của thanh ghi A qua phải**
  - **RR A**
  - **Ví dụ:**

MOV A,#36H	**A=0011 0110**
RR A	**A=0001 1011**
RR A	**A=1000 1101**
RR A	**A=1100 0110**

LỆNH QUAY

- Quay trái: quay các bit của thanh ghi A qua trái
  - RL A
  - Ví dụ:
    - MOV A,#72H A=0111 0010
      RL A A=1110 0100
      RL A A=1100 1001
      RL A A=1001 0011

QUAY CÓ NHỚ

- Quay phải có nhớ: các bit của thanh ghi A được quay từ trái sang phải và bit thấp nhất được đưa vào cờ CY
  - RRC A
  - Ví dụ:
    - CLR C

MOV A,#26H	A=0010 0110	
RRC A	A=0001 0011	CY=0
RRC A	A=0000 1001	CY=1
RRC A	A=1000 0100	CY=1

QUAY CÓ NHỚ

- Quay trái có nhớ: các bit của thanh ghi A được quay từ phải sang trái và bit cao nhất được đưa vào cờ CY
  - RLC A
  - Ví dụ:
    - SETB C

MOV A,#15H	A=0001 0101	
RLC A	A=0010 1011	CY=0
RLC A	A=0101 0110	CY=0
RLC A	A=1010 1100	CY=0
RLC A	A=0101 1000	CY=1

LỆNH TRÁO ĐỔI THANH GHI

- **Lệnh tráo đổi thanh ghi A:**
  - **SWAP A**
  - **Lệnh này chỉ làm việc với thanh ghi A theo cách tráo đổi phần byte cao và byte thấp của thanh ghi A với nhau**
  - **Ví dụ:**
    - **MOV A,#72H** **A=0111 0010**
      SWAP A **A=0010 0111**

VÍ DỤ

- Viết chương trình tìm tổng các số 1 trong một byte đã cho
- Giải:

```
        MOV R1,#0  ; lam bien dem so bit 1
        MOV R2,#8  ; dung lap 8 lan
        ;MOV A,#97H
AGAIN:  RLC A      ; dich trai co co CY
        JNC NEXT   ; nhay den NEXt neu CY=0
        INC R1     ; tang R1 len 1
NEXT: DJNZ R2,AGAIN
```

BÀI TẬP

- Giả sử RAM ở địa chỉ 40H-44H chứa nhiệt độ năm ngày được chỉ ra như sau, hãy tìm xem có giá trị nào bằng 55 hay không? Nếu có giá trị bằng 55 hãy đặt địa chỉ ngăn nhớ vào R4, nếu không thì R4=0
  - 40H=(76)
  - 41H=(55)
  - 42H=(69)
  - 43H=(65)
  - 44H=(50)

Giải

```
        MOV R0,#40H ; GÁN ĐỊA CHỈ CHO R0
        MOV B,#5  ; DÙNG LÀM BiẾN ĐẾM Ô NHỚ
        MOV A,#55
BACK:   CJNE A,@R0,NEXT ; SS A VỚI NỘI DUNG Ô NHỚ
        MOV r4,#55
        jmp exit
NEXT:   MOV R4,#0
        INC R0
        DINZ B,BACK  ; KiỂM TRA ĐÃ THỰC HiỆN 5?
EXIT:   END
```

Dạng chương trình sử dụng ngắt

```
ORG   0H
LJMP   MAIN          ; nhảy tới Main
ORG   03H            ; external interrupt 0
JMP EXINTER0
ORG   0BH            ; timer 0 interrupt
JMP COUNTER0
ORG   13H            ; external interrupt 1
JMP EXINTER1
ORG   1BH            ;timer 1 interrupt
JMP COUNTER1
```

```
MAIN:
EXINTER0
; chương trình phục vụ ngắt 0
RETI
EXINTER1
; chương trình phục vụ ngắt 1
RETI
COUNTER0
; chương trình phục vụ bộ đếm 0
RETI
COUNTER1
; chương trình phục vụ bộ đếm 1
RETI
END
```

TẠO VÀ TÍNH TOÁN THỜI GIAN GIỮ CHẬM

- **Chu kỳ máy:Độ dài của chu kỳ máy của họ 8051 phụ thuộc vào tần số bộ thạch anh. Một chu kỳ máy gồm 12 chu kỳ dao động thạch anh.Chu kỳ máy của họ 8051 thường là 1,085µs**

- **Ví dụ: Tính toán chu kỳ máy trong các trường hợp sau:**

1. **11,0592 MHz**
2. **16 MHz**

TÍNH THỜI GIAN TRỄ

- Thời gian trễ được xác định dựa vào thời gian thực hiện các lệnh bên trong vòng lặp và bỏ qua các lệnh bên ngoài vòng lặp.
- Ví dụ: Tính thời gian trễ của chương trình

```
        ORG 00H
DELAY:  MOV R3,#250
HERE:   NOP
        NOP
        NOP
        NOP
        DJNZ R3,HERE
        RET
```

LẬP TRÌNH CỔNG VÀO RA

- Cổng P0: có 8 chân từ 32-39: Bình thường đây là cổng ra.

```
        MOV A,#55H
BACK:   MOV P0,A
        ACALL DELAY
        SJMP BACK
```

- Để có thể vừa làm đầu vào vừa làm đầu ra nó phải được nối với một điện trở kéo R=10K. Lúc này khi khởi động lại R0 sẽ trở thành đầu vào. Muốn nó là đầu vào cần phải lập trình cho nó.

LẬP TRÌNH CỔNG VÀO RA

```
        MOV A,#0FFH
        MOV P0,A
BACK:   MOV A,P0
        MOV P1,A
        SJMP BACK
```

- Chuyển 8 bit địa chỉ AD0-AD7. Khi nối với bộ nhớ ngoài, nếu ALE=0 thì P0 cấp dữ liệu D0-D7, nếu ALE=1 P0 là địa chỉ.

LẬP TRÌNH CỔNG VÀO RA

- Cổng P1: có 8 chân từ 1-8, có thể sử dụng làm đàu vào và ra. Khi Reset cổng P1 là cổng ra.
- Để chuyển đổi cổng P1 làm đầu vào ta cần lập trình cho nó như sau:
  - MOV P1,#0FFH
    MOV A,P1
- Cổng P2:giống chân P1 về chức năng cổng vào ra.

LẬP TRÌNH CỔNG VÀO RA

- Có thể truy cập trực tiếp các bit địa chỉ của các cổng
  - SETB P1.2
  - CLR P1.2

LẬP TRÌNH CỔNG VÀO RA

- **Cổng P2: còn có chức năng chuyển dữ liệu và địa chỉ.**
  - **Cổng P2 kết hợp cổng P0 để tạo ra địa chỉ 16 bit của bộ nhớ ngoài.**
- **Cổng P3: chiếm 8 chân từ 10-17 với các chức năng sau:**
  - **P3.0: Nhận dữ liệu (RXD)**
  - **P3.1: Phát dữ liệu (TXD)**
  - **P3.2: Ngắt 0(INT0)**
  - **P3.3: Ngắt 1(INT1)**
  - **P3.4: Bộ định thời 0(T0)**
  - **P3.5: Bộ định thời 1(T1)**
  - **P3.6: Ghi WR**
  - **P3.7: Đọc RD**

Bài tập

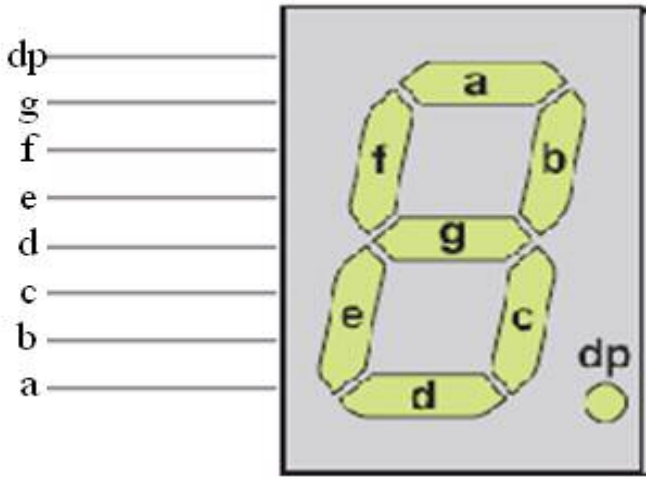
- **BT1: Viết chương xuất LED (cổng P1) các LED sáng tắt xen kẻ nhau. Với đèn sáng ở mức logic 1 và tắt ở mức logic 0.**

- **BT2: Viết chương trình tạo thời gian trễ nạp thanh ghi A giá trị 55H, và tăng giá trị này lên 700 lần.**

- **VN: Viết chương trình điều khiển 16 led(sáng dồn từ hai bên vào).**

- **OUTPUT**
  - Led đơn
  - Led 7 đoạn
  - Led ma trận
  - LCD
- **INPUT**
  - Phím đơn
  - Bàn phím ma trận (bàn phím số Hex)
- **In/Out**
  - RS232
  - ADC/DAC

LED 7 ĐOẠN



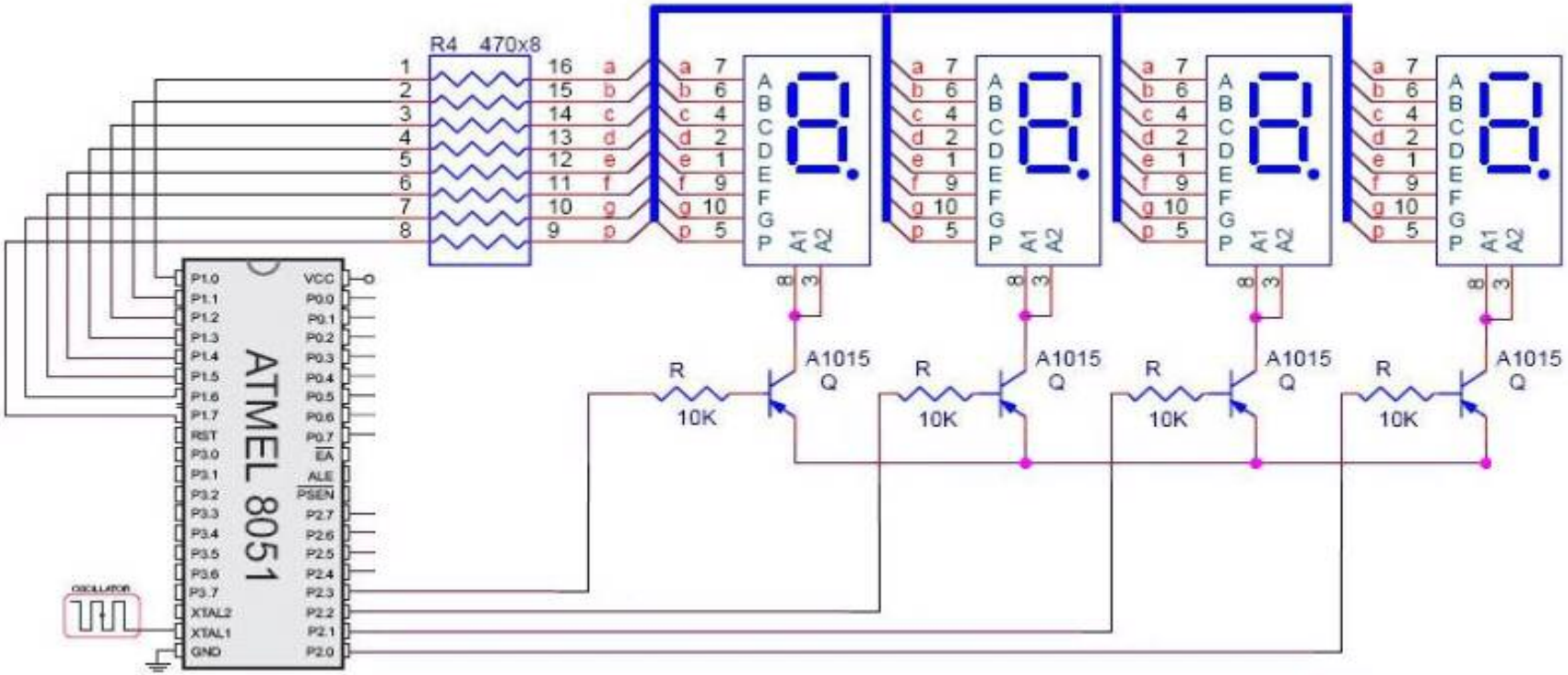
Có 2 loại chính

- ✓ **Anod chung**
- ✓ **Catod chung**

Hiển thị	Anod	Catod
0	C0h	3Fh
1	F9h	06h
2	A4h	5Bh
3	B0h	4Fh
4	99h	66h
5	92h	6Dh
6	82h	7Dh
7	F8h	07h
8	80h	7Fh
9	98h	67h
A	88h	77h
B	c6h	39h
C	86h	79h
D	8Eh	71h
E	82h	70h
F	89h	76h
[trắng]	FFh	00h

GHÉP LED CHUNG DATA & PHƯƠNG PHÁP QUÉT LED

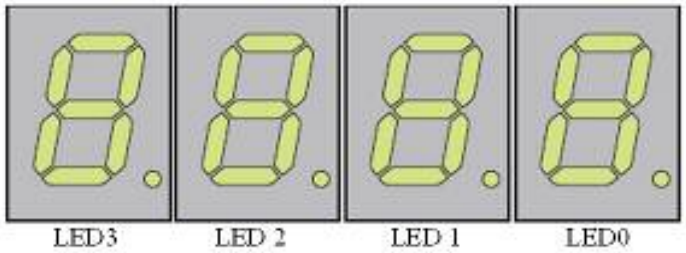
- **Mạch nguyên lý**



Phương pháp quét led

- Phương pháp này dựa trên nguyên tắc hình ảnh trên võng mạc mắt vẫn còn được lưu thêm 1 thời gian sau khi hình ảnh thực tế đã mất.
- Tại mỗi thời điểm chỉ bật sáng và hiển thị 01 led
- Điều khiển bật sáng từng led 1 với số hiển thị tương ứng trong 1 thời gian rất ngắn rồi tắt
- Do tần số quét rất cao nên mắt người không thấy sự thay đổi đó mà chỉ thấy tất cả các led cùng sáng.

Ví dụ 8.1: Hiển thị số “9876”



Thuật toán :

Tắt led 0, Bật led 3 (P2.0=1 & P2.3=0)

Đưa mã số 9 ra port 1

Delay

Tắt led 3, Bật led 2 (P2.3=1 & P2.2=0)

Đưa mã số 8 ra port 1

Delay

Tắt led 2, Bật led 1, (P2.1=0 & P2.1=0)

Đưa mã số 7 ra port 1

Delay

Tắt led 1, Bật led 0, (P2.1=1 & P2.0=0)

Đưa mã số 6 ra port 1

Delay

Lặp lại từ đầu

```
Org 00
LAP:
    SETB P2.0          ; tắt led 0
    CLR P2.3           ; bật led 3
    MOV P1,#98H        ; hiển thị số 9
    LCALL DELAY        ; chờ
    SETB P2.3          ; tắt led 3
    CLR P2.2           ; bật led 2
    MOV P1,#80H        ; hiển thị số 8
    LCALL DELAY        ; chờ
    SETB P2.2          ; tắt led 2
    CLR P2.1           ; bật led 1
    MOV P1,#0F8H       ; hiển thị số 7
    LCALL DELAY        ;chờ
    SETB P2.2          ; tắt led 1
    CLR P2.1           ; bật led 0
    MOV P1,#82H        ; hiển thị số 6
    LCALL DELAY        ; chờ
    SJMP LAP
```

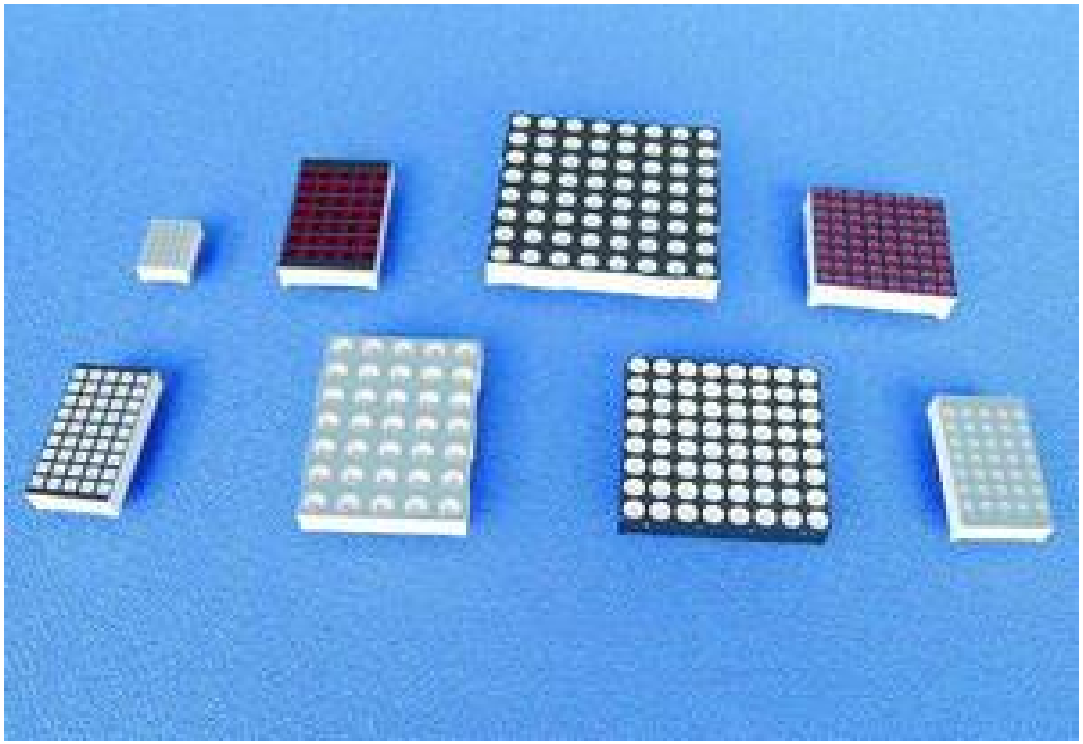
```
DELAY:
    PUSH 01
    MOV R1,#0FFH
    DJNZ R1,$
    POP 01
RET
END.
```

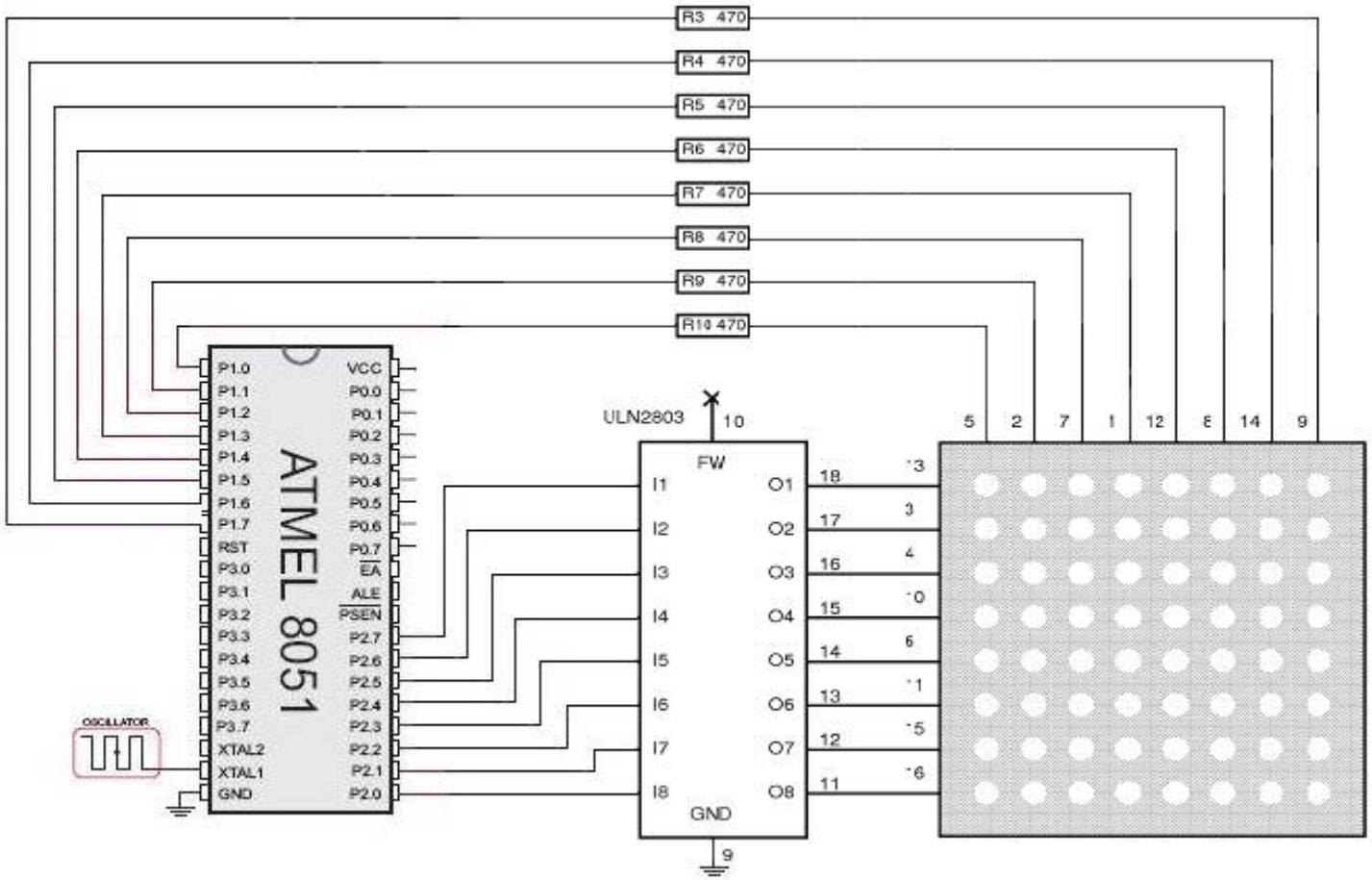
RUNG PHÍM & PHƯƠNG PHÁP CHỐNG RUNG PHÍM

- Bàn phím là chuyển mạch cơ khí
- Khi ấn và nhả nút → rung phím : như là nhấn và nhả nhiều lần trong thời gian rất ngắn
- Có 02 cách chống xung nảy khi rung phím
  - Bằng phần cứng
  - Bằng phần mềm
- Chống xung nảy bằng phần mềm : Kiểm tra nhiều lần liên tiếp (vd: 50 lần) có phím ấn thì mới xem như nhấn phím 1 lần

LED MA TRẬN

- Tập hợp các led đơn được tổ chức theo kiểu ma trận và đóng gói lại
- Ứng dụng nhiều trong lĩnh vực quang báo
- Nhiều loại với nhiều kích cỡ khác nhau
  - 8x8
  - 5x7
  - …




R3 470
R4 470
R5 470
R6 470
R7 470
R8 470
R9 470
R10 470
ATMEL 8051
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
RST
P3.0
P3.1
P3.2
P3.3
P3.4
P3.5
P3.6
P3.7
XTAL2
XTAL1
GND
VCC
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
EA
ALE
PSEN
P2.7
P2.6
P2.5
P2.4
P2.3
P2.2
P2.1
P2.0
OSCILLATOR
ULN2803
10
FW
I1
I2
I3
I4
I5
I6
I7
I8
O1
O2
O3
O4
O5
O6
O7
O8
GND
9
18
17
16
15
14
13
12
11
5
2
7
1
12
8
14
9