

Phân Lý thuyết

1. Giới thiệu chung - Mô hình hệ VXL - Nguyên tắc hoạt động
2. Cấu trúc và hoạt động của vi xử lý 8085
3. Quá trình thực hiện 1 lệnh trong VXL 8085
4. Giới thiệu về vi điều khiển PIC
5. Bộ công cụ nạp chương trình, công cụ mô phỏng vi điều khiển
6. Bộ định thời Timer
7. Ghép nối với bộ hiển thị
8. ADC
9. Giao tiếp truyền dữ liệu
10. Ngắt
11. PWM

Ngắt trong PIC16F877A

- 15 nguồn ngắt khác nhau
- Thanh ghi điều khiển ngắt INTCON

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

- Bit cho phép ngắt toàn cục GIE

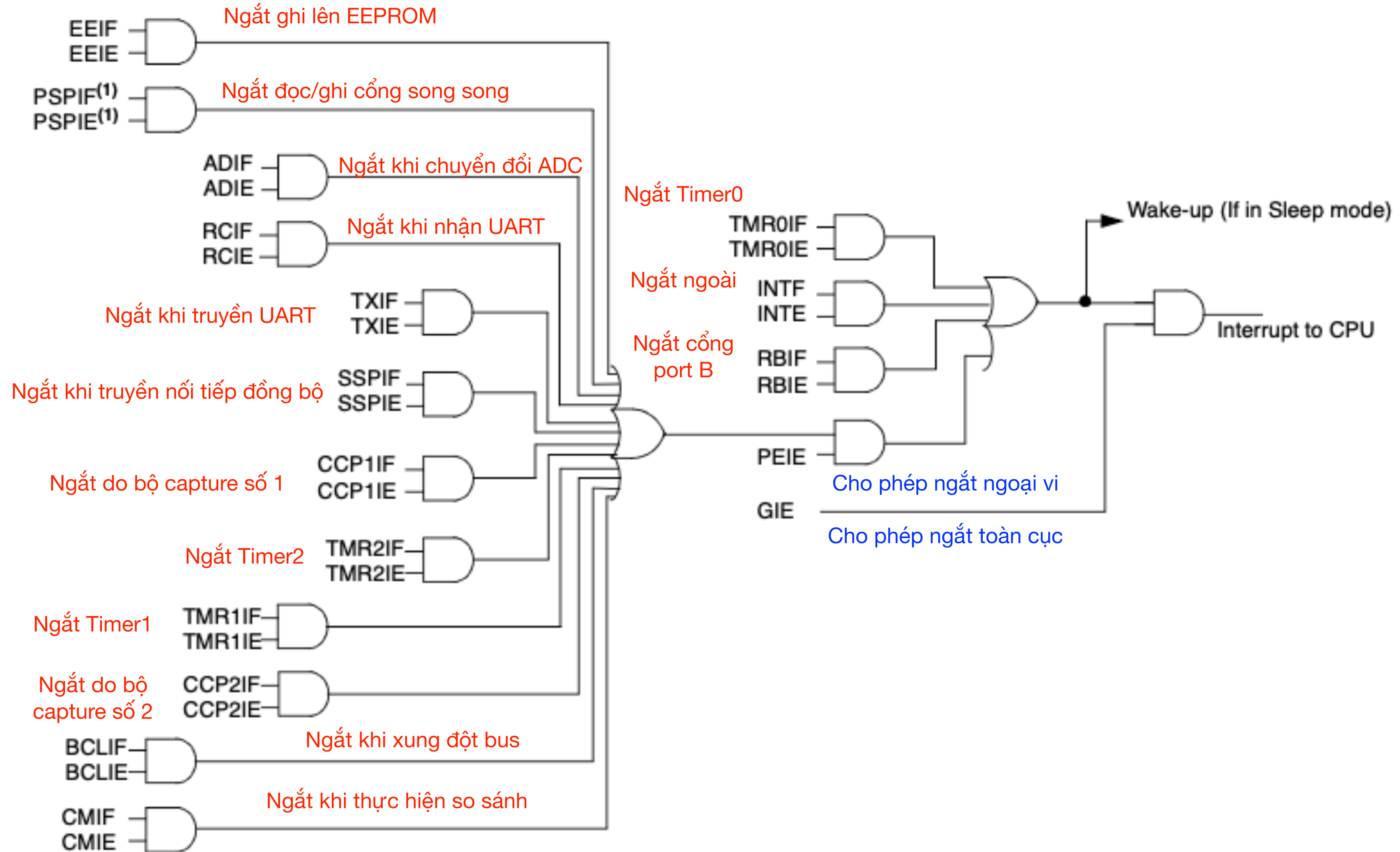
GIE =1 Cho phép tất cả các nguồn ngắt

GIE=0 không cho phép tất cả các nguồn ngắt

Khi GIE=1, các nguồn ngắt độc lập có thể được set ở dạng cho phép hoặc cấm ngắt bởi các bit cho phép tương ứng

Bit GIE chỉ được xóa khi reset

Ngắt trong PIC16F877A



Ngắt Timer0

- Khi thanh ghi TMR0 tràn từ FFh sang 00h thì bit TM0IF được set

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

TMR0IF=1: khi TMR0 tràn từ FFh sang 00h

TMR0IE=1: Cho phép ngắt Timer0
TMR0=0: Cấm ngắt Timer0

Cấu hình cho Timer0 vui lòng xem lại phần bộ định thời Timer

Ngắt ngoài chân RB0/INT

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

INTF=1: khi có 1 xung vào chân 33

INTE=1: Cho phép ngắt chân RB0
INTE=0: Cấm ngắt

OPTION_REG REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

INTEDG=1: Tác động theo sườn lên
INTEDG=0: tác động theo sườn xuống

40	<input type="checkbox"/>	↔	RB7/PGD
39	<input type="checkbox"/>	↔	RB6/PGC
38	<input type="checkbox"/>	↔	RB5
37	<input type="checkbox"/>	↔	RB4
36	<input type="checkbox"/>	↔	RB3/PGM
35	<input type="checkbox"/>	↔	RB2
34	<input type="checkbox"/>	↔	RB1
33	<input type="checkbox"/>	↔	RB0/INT

Ngắt ngoài cổng portB thay đổi

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

RBIF=1: khi có sự thay đổi các bit **RB4:7**

40	<input type="checkbox"/>	↔	RB7/PGD
39	<input type="checkbox"/>	↔	RB6/PGC
38	<input type="checkbox"/>	↔	RB5
37	<input type="checkbox"/>	↔	RB4
36	<input type="checkbox"/>	↔	RB3/PGM
35	<input type="checkbox"/>	↔	RB2
34	<input type="checkbox"/>	↔	RB1
33	<input type="checkbox"/>	↔	RB0/INT

RBIE=1: Cho phép ngắt khi portB thay đổi
RBIE=0: Cấm ngắt

Khi nào sử dụng ngắt

Xét bài toán: Đưa ra chân RA0 tín hiệu xung vuông chu kỳ 1ms

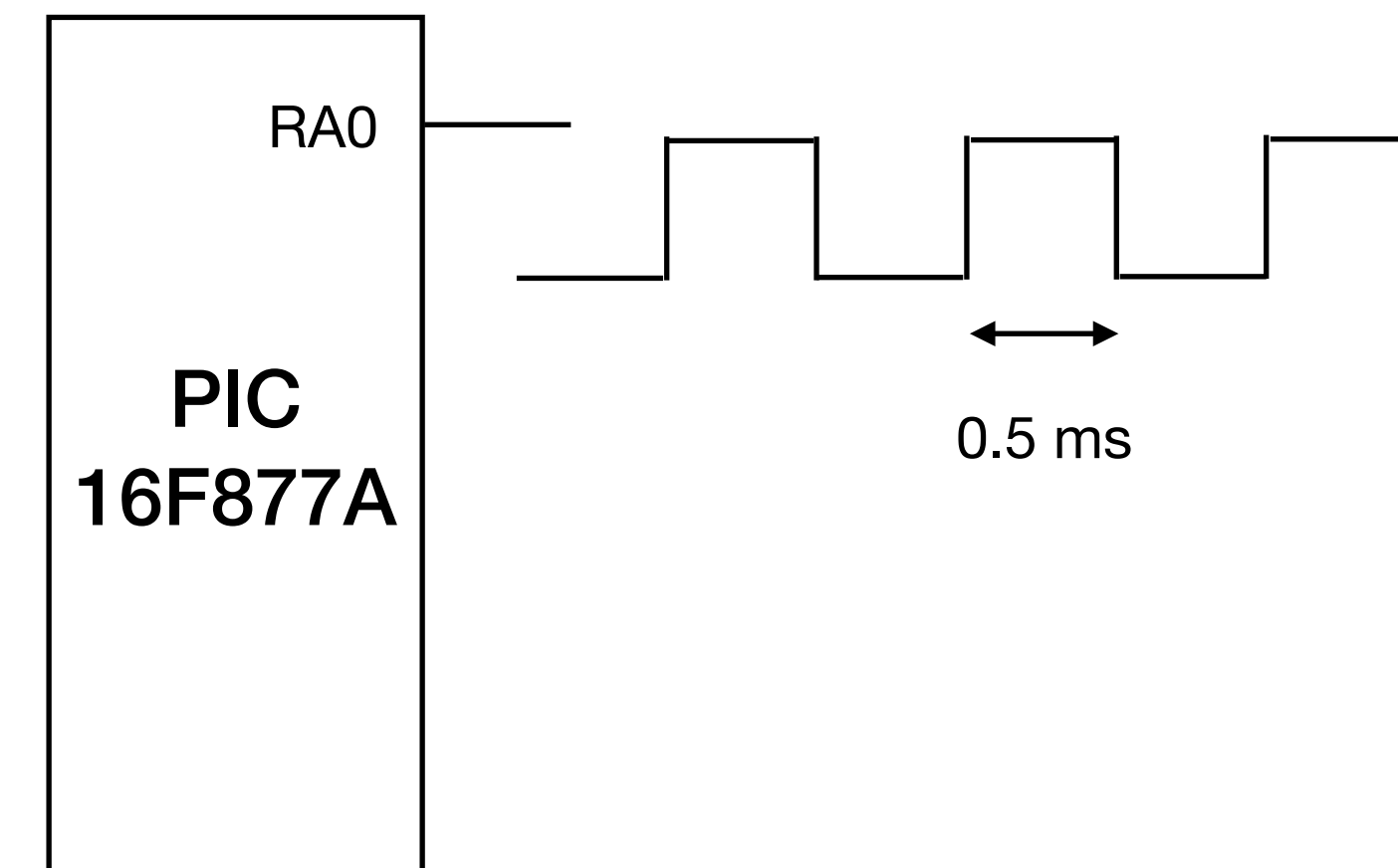
Phương án xử lý

Xác định chân RA0 là output

Void main()

Thiết lập vòng lặp while(1)

- set chân RA0 lên 1
- Trễ 0.5 ms
- Reset chân RA0 về 0
- Trễ 0.5 ms



Tiến trình được quản lý bằng chương trình chính main()

Khi nào sử dụng ngắt

Xét bài toán:

Đưa ra chân RA0, RA1 tín hiệu xung vuông chu kỳ 1ms và 2 ms

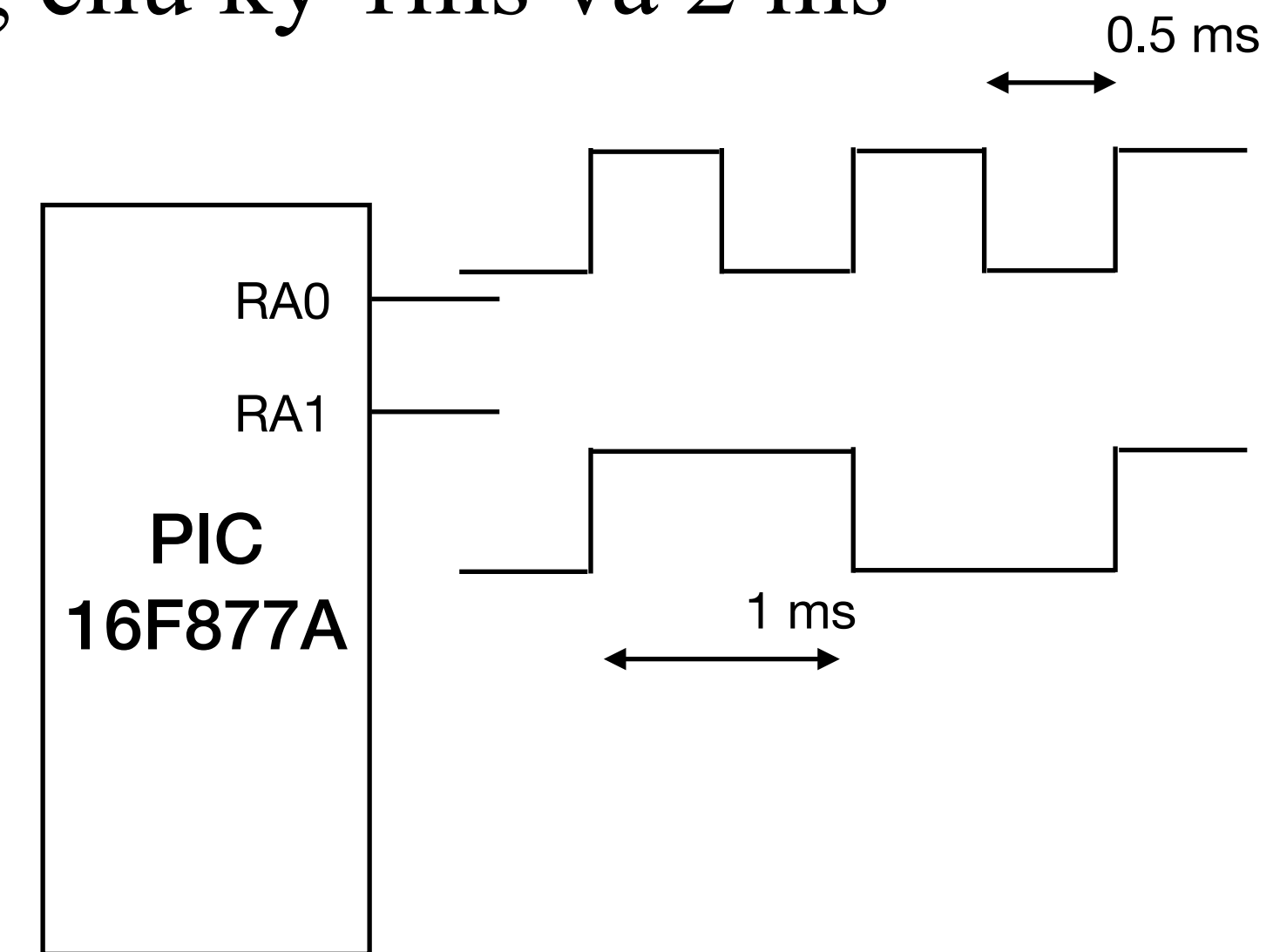
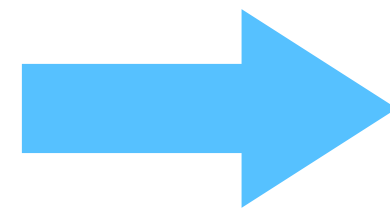
Nếu coi tại thời điểm đầu tiên, 2 chân cùng có sườn lên

Xác định chân RA0, RA1 là output

Void main()

Thiết lập vòng lặp while(1)

- set chân RA0, RA1 lên 1
- Trễ 0.5 ms
- Reset chân RA0 về 0
- Trễ 0.5 ms
- Set chân RA0 lên 1, reset RA1 về 0
- Trễ 0.5 ms
- Reset RA0 về 0
- Trễ 0.5 ms



- CPU quản lý đồng thời 2 tiến trình
- Có thời điểm 2 xung bắt buộc có cùng sườn lên hoặc xuống
- Tần số các xung phải là bội số của nhau

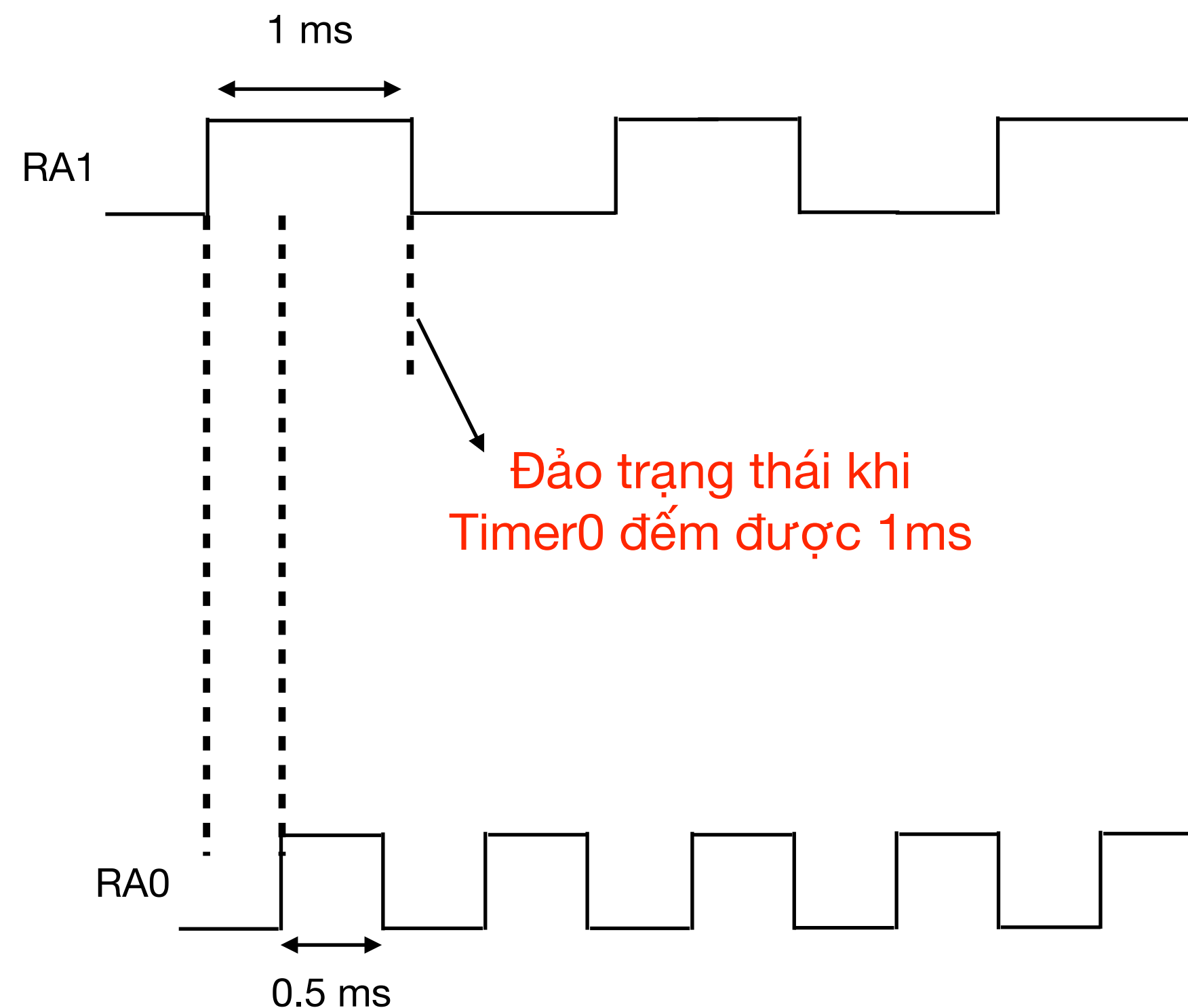
Kém linh hoạt

Khi nào sử dụng ngắt

Xét bài toán:

Đưa ra chân RA0, RA1 tín hiệu xung vuông chu kỳ 1ms và 2 ms

Sử dụng ngắt để quản lý tiến trình



Xác định chân RA0, RA1 là output

Khai báo sử dụng ngắt **Timer0**

Nạp các thông số cho Timer0 để timer sẽ chạy được 1 ms

```
int_timer0()
```

```
RA1 = ~RA1
```

```
Xoá cờ ngắt TMR0IF
```

```
Nạp lại thông số cho Timer0
```

Quản lý tiến trình tạo xung RA1

```
Void main()
```

```
Thiết lập vòng lặp while(1)
```

- set chân RA0
- Trễ 0.5 ms
- Reset chân RA0 về 0
- Trễ 0.5 ms

Quản lý tiến trình tạo xung RA0

Khi nào sử dụng ngắt

Xét bài toán:

Đưa ra chân RA0, RA1 tín hiệu xung vuông chu kỳ 1ms và 2 ms

Sử dụng ngắt để quản lý tiến trình

Xác định chân RA0, RA1 là output

Khai báo sử dụng ngắt **Timer0, Timer1**

Nạp các thông số cho Timer0 để timer sẽ chạy được 1 ms, Timer1 để chạy 0.5 ms

```
int_timer0()
```

```
RA1 = ~RA1
```

```
Xoá cờ ngắt TMR0IF
```

```
Nạp lại thông số cho Timer0
```

Quản lý tiến trình tạo xung RA1

```
int_timer1()
```

```
RA0 = ~RA0
```

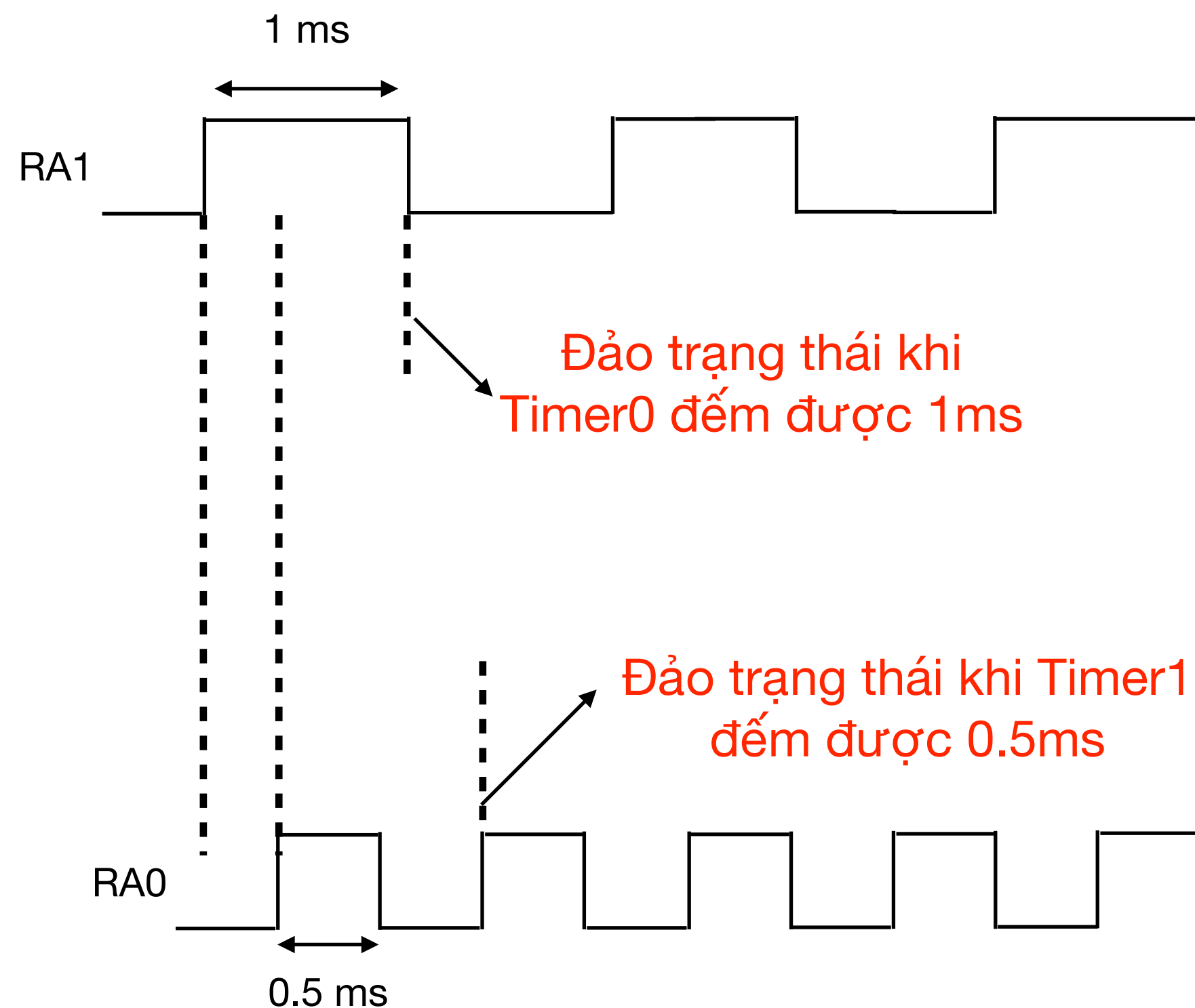
```
Xoá cờ ngắt TMR1IF
```

```
Nạp lại thông số cho Timer1
```

Quản lý tiến trình tạo xung RA0

```
Void main()
```

Main nghỉ hoặc quản lý tiến trình khác



Khi nào sử dụng ngắt

- Khi chương trình cần quản lý nhiều hơn 1 tiến trình hoạt động độc lập
- Khi cần xử lý các tiến trình mà điều kiện xảy ra là ngẫu nhiên

Ví dụ

Xây dựng hệ thống hiển thị số 3 chữ số trên led 7 thanh. Số cần hiển thị là số lần bấm một phím bấm

