

BÀI 2. KIẾN TRÚC PHẦN CỨNG PIC18F4520, LẬP TRÌNH GHI/ĐỌC TRÊN CÁC CỔNG/CHÂN VÀO/RA ĐA NĂNG (GPIO)

Mục tiêu bài học:

Sinh viên nhớ và hiểu:

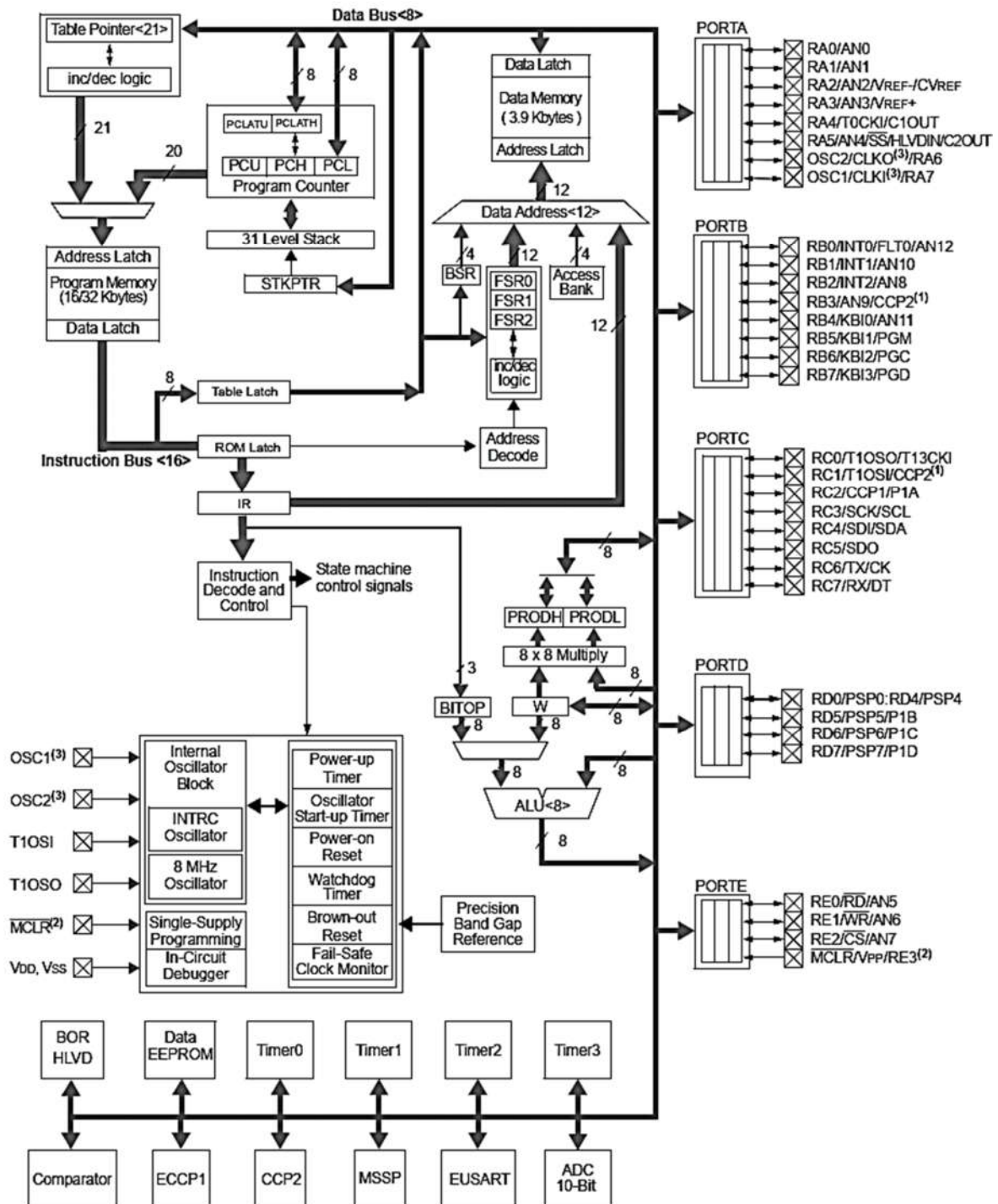
- Khái quát về PIC18F4520.
 - + Các khối cơ bản trong vi điều khiển
 - + Các ngoại vi của vi điều khiển: GPIO, Timer, PWM, ADC, USART
- Hoạt động Reset.
 - + Các chế độ hoạt động
 - + Mạch điện sử dụng trong chế độ Reset bằng nguồn và Reset trên chân MCLR
- Bộ tạo dao động.
 - + Các chế độ hoạt động
 - + Mạch điện sử dụng trong chế độ HS
- Các cổng GPIO và các thanh ghi liên quan.
 - + PORT A, B, C, D, E.
 - + Cách thiết lập thanh ghi TRISx, PORTx
 - + Thiết lập ADCON1=0x0f để các chân ANx (PORT A, B, E) là đầu vào tín hiệu số.
- Lập trình và mô phỏng ghi/đọc theo byte, theo bit.
 - + Khung chương trình
 - + Các bước lập trình và mô phỏng

Sinh viên vận dụng các kiến thức đã học để lập trình, mô phỏng hoạt động của GPIO, bao gồm:

- Thiết lập chiều vào ra dữ liệu trên các chân GPIO
- Lập trình, mô phỏng hoạt động vào/ra sử dụng if...else (*)
- Thiết kế 01 ứng dụng sử dụng while (*)

****: Học trên lớp***

2.1. Kiến trúc phần cứng PIC 18F4520



Hình 2.1 Sơ đồ khối Pic 18F4520 (trang 11 – datasheet pic 18f4520)

- Cấu tạo của vi điều khiển PIC 18F4520 bao gồm CPU, khối bộ nhớ, khối thiết bị ngoại vi.

- **Bộ xử lý trung tâm CPU(Central Processing Unit):**

- Tần số làm việc tối đa 40 MHz, sản xuất bằng công nghệ nanoWatt.
- Thiết kế theo kiến trúc Havard, tập lệnh RISC. (xem thêm mục 2.1 trong giáo trình hoặc mục 1.1 trong tài liệu PIC18F4520 datasheet)

- **Bộ nhớ (Memory):**

- Bộ nhớ chương trình (Program Memory) bao gồm 32 Kbytes bộ nhớ ROM (Read-only Memory) kiểu Flash.
- Bộ nhớ dữ liệu (Data Memory) bao gồm 1536 byte SRAM (Static Random Access Memory).
- EEPROM có dung lượng 256 byte

- . **Bộ nhớ chương trình**

Bộ nhớ Flash của PIC 18F4520 có dung lượng 32 Kbyte nên chứa được 16384 lệnh từ đơn (single-word instructions), với dải địa chỉ từ 0000h đến 7FFFh. Nếu đọc ở vùng nhớ ngoài 32Kbyte của PIC 18F4520 và trong khoảng 2 Mbyte mà nó có thể quản lý thì giá trị dữ liệu trả về sẽ là “0”. PIC 18F4520 có 31 mức ngăn xếp.

- Vector Reset của PIC 18F4520 được đặt ở địa chỉ 0000h, khi reset nội dung của thanh ghi đếm chương trình PC sẽ được xóa về 0. Các thanh ghi của PIC 18F4520 sẽ được tải lại các giá trị mặc định.

- Vector ngắt ưu tiên cao (High-Priority Interrupt Vector) được đặt ở địa chỉ 0008h, vector ngắt ưu tiên thấp (Low-Priority Interrupt Vector) được đặt ở địa chỉ 0018h. (xem thêm mục 2.3.1 trong giáo trình hoặc mục 5.1; 6.0 trong tài liệu PIC18F4520 datasheet).

- . **Bộ nhớ dữ liệu RAM**

Bộ nhớ dữ liệu RAM 1536 byte(SRAM) được chia thành 2 vùng chức năng riêng biệt, vùng RAM đa dụng GPR (General Purpose Registers) sử dụng để chứa dữ liệu, vùng các thanh ghi chức năng đặc biệt SFR(Special Function Registers) chứa các thanh ghi chức năng điều khiển ngoại vi và CPU. (xem thêm mục 2.3.2 trong giáo trình hoặc mục 5.3 trong tài liệu PIC18F4520 datasheet).

- . **Bộ nhớ dữ liệu EEPROM**

Bộ nhớ dữ liệu EEPROM của PIC 18F4520 là bộ nhớ mảng không bị mất dữ liệu khi mất điện, độc lập với bộ nhớ chương trình và bộ nhớ dữ liệu RAM, được sử dụng để lưu trữ dữ liệu lâu dài. Nó có thể ghi/đọc được 1.000.000 lần, dữ liệu có thể lưu trữ trong bộ nhớ 100 năm. Điều khiển và đọc/ghi bộ nhớ dữ liệu EEPROM không truy cập trực tiếp vào tệp thanh ghi hay khoảng trống bộ nhớ chương trình mà được truy cập, điều khiển gián tiếp qua các thanh ghi chức năng đặc biệt SFR. (xem thêm mục 2.3.3 trong giáo trình hoặc mục 7.0 trong tài liệu PIC18F4520 datasheet).

- **Bộ phát xung hệ thống (Oscillator):**

Nguồn xung từ bên ngoài hoặc từ bộ phát xung của hệ thống sẽ được đi qua bộ nhân hoặc chia tần số để lựa chọn lấy tần số thích hợp để làm xung hệ thống.

- Nguồn xung chính được đưa vào chip qua chân OSC1 và OSC2.
- Nguồn xung phụ được đưa vào chip qua các chân T1OSI, T1OSO.
- Bộ phát xung nội INTRC tần số 31kHz.
- Bộ phát xung nội trên chip tần số 8 MHz.

- **Watchdog Timer(WDT):**

WDT là một bộ timer có chức năng đặc biệt. Nếu được “cho phép” WDT sẽ hoạt động và khi tràn sẽ khởi động lại hệ thống. Thời gian khởi động lại hệ thống có thể lựa chọn được từ 4ms đến 131,072s. (xem thêm mục 2.1 trong giáo trình hoặc mục 23.2 trong tài liệu PIC18F4520 datasheet).

Khi không sử dụng Watchdog Timer cần thiết lập lệnh cấu hình :

#pragma config WDT=OFF

- **Bộ nạp chương trình:**

Bộ nạp chương trình nối tiếp trên chip (Single-Supply InCircuit Serial Programming) sẽ giúp nạp chương trình từ mạch nạp vào bộ nhớ ROM qua các chân PGM, PGC và PGD.

Nếu sử dụng nguồn của mạch nạp cần thiết lập lệnh cấu hình :

#pragma config LVP = OFF

- **Bộ Debugger (In-Circuit Debugger):**

Mạch Debugger trên chip sẽ giúp người lập trình kiểm soát lỗi chương trình bằng cách cho vi điều khiển hoạt động ở chế độ chạy từng lệnh, nhóm lệnh hay toàn bộ chương trình.

- **Khởi phát hiện tín hiệu reset:**

Mạch phát hiện tín hiệu reset có khả năng phát hiện 03 nguồn reset:

- Reset từ chân MCLR.
- Reset khi bật nguồn (POR: Power-on Reset).
- Reset khi nguồn yếu (BOR : Brown-out Reset).

- **Khởi quản lý lỗi bộ phát xung (Fail-Safe Clock Monitor):**

Khởi này được sử dụng để quản lý an toàn bộ phát xung hệ thống.

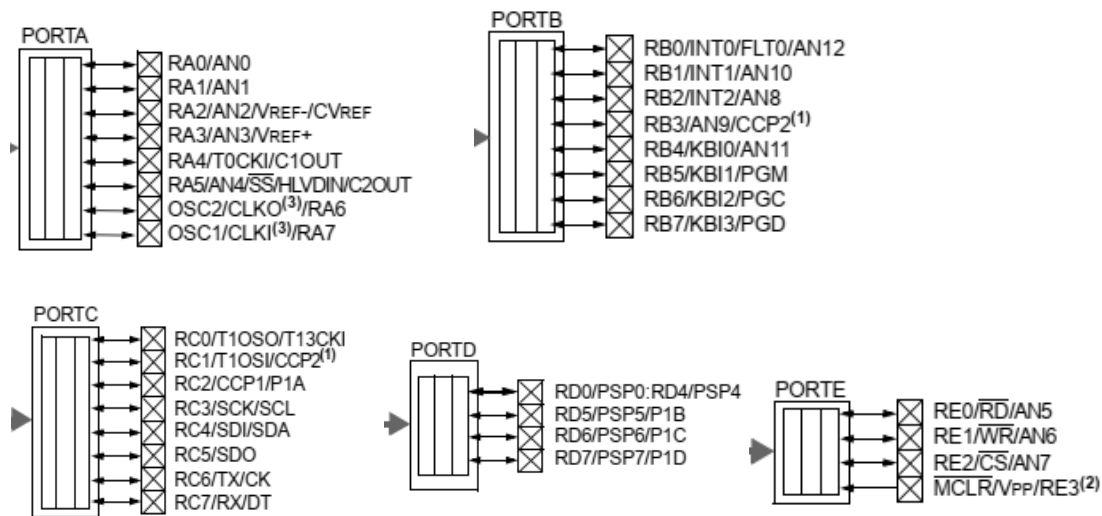
- **Khởi định thời khởi động bộ phát xung (Oscillator Start-up Timer):**

Khối này sử dụng để tạo thời gian trễ chờ cho bộ phát xung ổn định.

• **Thiết bị ngoại vi (Peripheral):**

- Bộ phát hiện điện áp cao/thấp HLVD(High/low-Voltage Detect).
- Bộ nhớ lưu dữ liệu khi tắt nguồn EEPROM.
- 04 bộ đếm, định thời 16 bit: Timer0, Timer1, Timer2 và Timer3.
- 01 bộ so sánh tín hiệu tương tự (Comparator).
- 02 bộ CCP1, CCP2 (Capture, Compare, Pwm : Chụp, So sánh, xung Pwm); 01 bộ ECCP (Enhanced CCP).
- 01 cổng truyền thông nối tiếp đồng bộ (Master Synchronous Serial Port) có thể hoạt động được ở chế độ SPI hoặc I2C.
- 01 cổng truyền thông nối tiếp đồng bộ/không đồng bộ tăng EUSART(Enhanced Universal Synchronous Asynchronous Receiver Transmitter), giúp vi điều khiển PIC có thể giao tiếp với nhau hoặc giao tiếp với cổng COM của máy tính.
- 13 kênh biến đổi tương tự - số (ADC) độ phân giải 10 bit.
- Khối ghép nối vào ra:

+ Các chân vào ra dữ liệu : PORTA, PORTB, PORTC, PORTD, PORTE



Hình 2.2. Sơ đồ chân GPIO

. Sơ đồ chân

- Sơ đồ chân dạng PDIP (Lead Plastic Dual In-Line Package) hai hàng chân cắm hai bên.

- TQFP (Thin Quad Flat Package) bốn hàng chân dán vỏ mỏng.

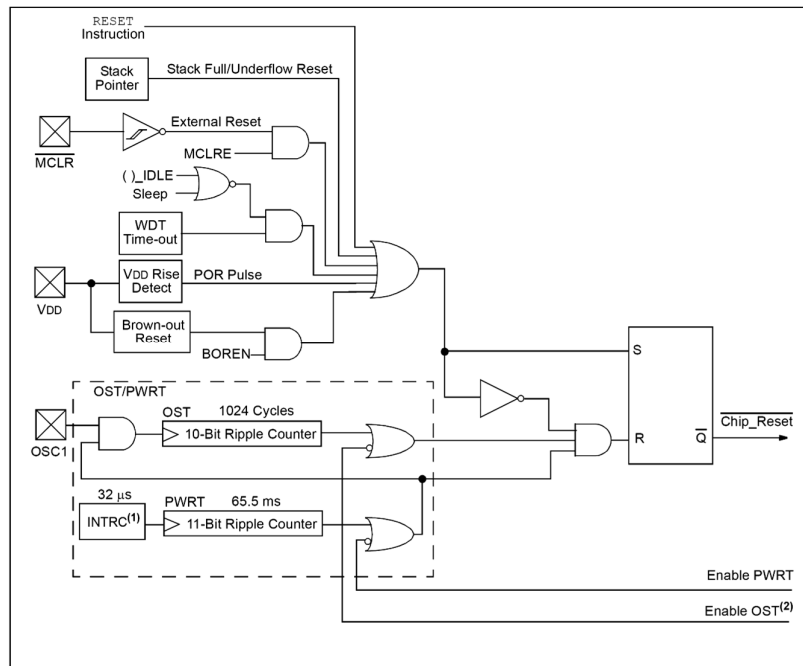
- Sơ đồ chân dạng QFN(Quad Flat No-lead) bốn hàng chân dán dưới đế.

(xem thêm mục 2.2 trong giáo trình hoặc trang số 3, số 4 trong tài liệu PIC18F4520 datasheet).

2.2. Hoạt động Reset

Vi điều khiển PIC 18F4520 có 8 nguồn Reset:

- Reset do bật nguồn POR (Power-on Reset).
- Reset từ chân MCLR, sử dụng trong quá trình hoạt động bình thường.
- Reset từ chân MCLR, sử dụng trong chế độ quản lý nguồn.
- Reset do Watchdog Timer (WDT).
- Reset do yếu nguồn BOR (Brown-out Reset), reset này có thể lập trình được.
- Reset bằng lệnh RESET.
- Reset do đầy ngăn xếp (Stack Full Reset).
- Reset do rỗng ngăn xếp (Stack Underflow Reset).

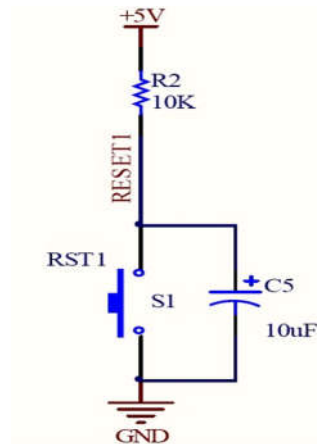


Hình 2.3. Sơ đồ khối Reset của PIC 18F4520

2.2.1. Các chế độ Reset

2.2.1.1. Nguồn Reset từ chân MCLR

Nguồn reset MCLR được nối từ mạch reset bên ngoài qua chân MCLR/RE3. Nếu bit MCLRE trong thanh ghi CONFIG3H được đặt bằng 1 thì chân MCLR/RE3 là chân reset, đặt MCLRE = “0” chân MCLRE là chân vào/ra của PORTE. Khi có mức điện áp thấp được đặt lên chân MCLR thì sẽ thực hiện reset vi điều khiển, reset MCLR có chức năng lọc nhiễu nên nó sẽ bỏ qua xung reset tần số cao.

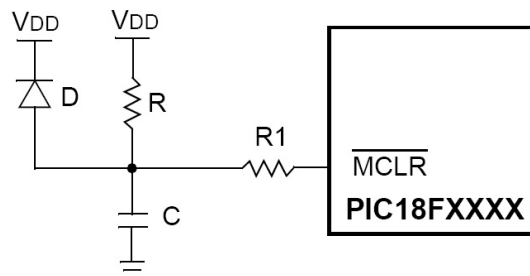


Hình 2.4. Mạch reset ngoài MCLR

2.2.1.2. Reset do bật nguồn POR (Power-on Reset)

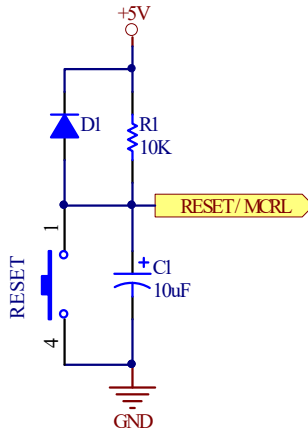
Khi nguồn được cấp vào chân V_{DD} , khối reset POR sẽ phát hiện sự sụt giảm trên chân V_{DD} sau đó phát tín hiệu reset vi điều khiển. Bit POR trong thanh ghi RCON sẽ báo trạng thái của reset POR, POR = ‘1’ là không phát hiện tín hiệu reset, POR = ‘0’ là phát hiện tín hiệu reset POR, bit POR cần được đặt bằng ‘1’ sau reset POR (không được thiết lập bằng phần cứng).

Giá trị linh kiện: $R < 40\text{ k}\Omega$; $R1 \geq 1\text{ K}\Omega$.



Hình 2.5. Mạch reset do bật nguồn POR.

- Hoạt động RESET – lựa chọn reset trên chân MCLR – RE3. Sử dụng câu lệnh cấu hình: **#pragma config MCLRE = ON**



Hình 2.6. Sơ đồ mạch Reset của PIC 18F4520

2.2.1.3. Reset do sụt nguồn BOR (Brown-out Reset)

Reset do sụt nguồn BOR (Brown-out Reset) được sử dụng để Reset lại vi điều khiển khi phát hiện nguồn cấp bị sụt áp. Reset BOR được điều khiển thông qua các bit BORV1: BORV0 và BOREN1: BOREN2 của thanh ghi CONFIG2L. (xem thêm mục 2.5.1.3 trong giáo trình hoặc mục 4.4 trong tài liệu PIC18F4520 datasheet).

2.2.1.4. Bộ đếm thời gian Reset hệ thống

Vi điều khiển sử dụng bộ đếm thời gian PWRT(Power-up Timer) và OST (Oscillator Start-up Timer) để thiết lập thời gian Reset hệ thống sau khi có tín hiệu yêu cầu reset do sụt nguồn hay bật nguồn. (xem thêm mục 2.5.1.4 trong giáo trình hoặc mục 4.5 trong tài liệu PIC18F4520 datasheet).

2.2.2. Các thanh ghi của Reset

- Thanh ghi điều khiển Reset: **RCON**

IPEN	SBOREN	—	RI	TO	PD	POR	BOR
bit 7				bit 0			

- Thanh ghi cấu hình 3 byte cao: **CONFIG3H**

MCLRE	—	—	—	—	LPT1OSC	PBADEN	CCP2MX
bit 7				bit 0			

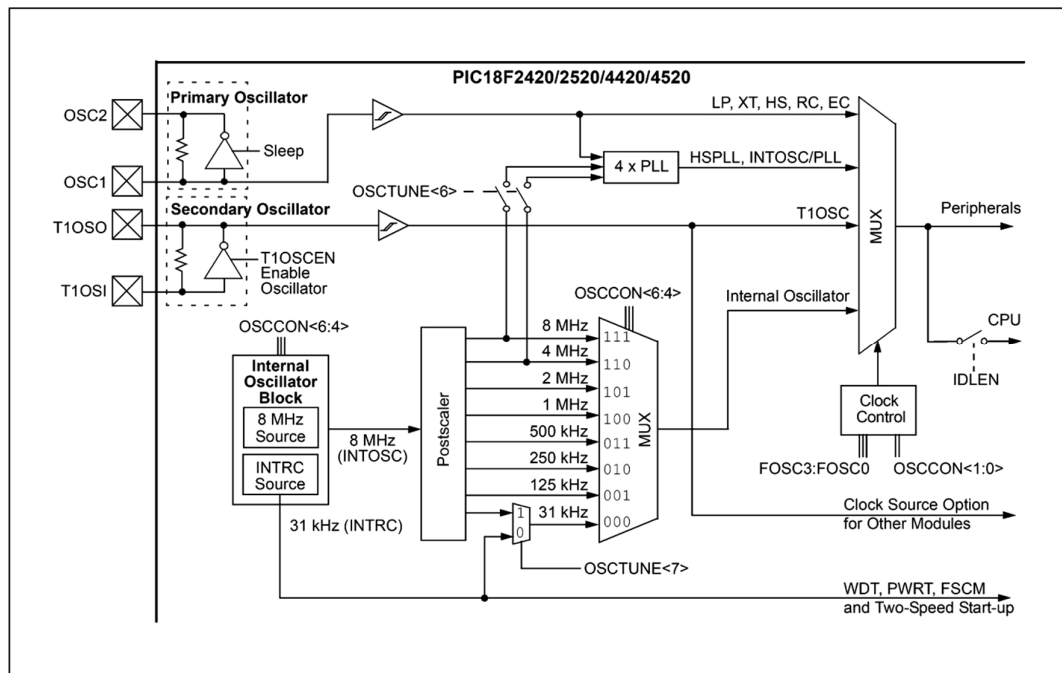
(xem thêm mục 2.5.2 trong giáo trình hoặc mục 4.1 và trang 253 trong tài liệu PIC18F4520 datasheet).

2.3. Khối tạo dao động

PIC 18F4520 có thể hoạt động ở một trong 10 chế độ tạo dao động khác nhau. Việc lựa chọn các chế độ tạo dao động nhờ các bit FOSC3:FOSC0 trong thanh ghi CONFIG1H.

Các chế độ tạo dao động:

- LP (Low-Power Crystal) nguồn xung thạch anh ngoài, nguồn thấp
- XT (Crystal/Resonator) thạch anh/bộ cộng hưởng bên ngoài
- HS (High-Speed Crystal/Resonator) thạch anh/bộ cộng hưởng bên ngoài tốc độ cao
- HSPLL nhân 4 lần tần số HS bằng vòng khóa pha (Phase Locked Loop)
- RC (External Resistor/Capacitor) tạo dao động bằng mạch RC bên ngoài, phát xung $F_{osc}/4$ ra chân RA6.
- RCIO tạo dao động bằng mạch RC ngoài, vào/ra trên chân RA6.
- INTIO1 bộ tạo dao động nội, phát xung $F_{osc}/4$ ra chân RA6, vào/ra trên chân RA7.
- INTIO2 bộ tạo dao động nội, vào/ra trên chân RA6 và RA7.
- EC bộ phát xung ngoài, phát xung $F_{osc}/4$ ra chân RA6.
- ECIO bộ phát xung ngoài, vào/ra trên chân RA6.

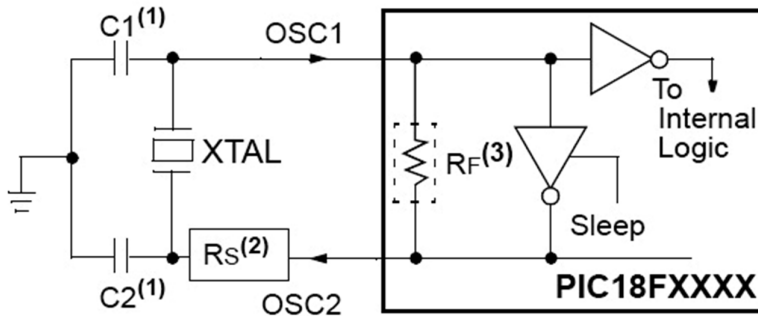


Hình 2.7. Sơ đồ khối bộ tạo dao động trên PIC 18F4520

2.3.1. Tạo dao động bằng thạch anh ngoài (Crystal/ Ceramic Resonator)

Trong chế độ tạo dao động LP, XT, HS, HSPLL sử dụng thạch anh (Crystal) chưa có tụ điện hoặc mạch cộng hưởng thạch anh bọc gốm đã có tụ điện (Ceramic Resonator). Ở các chế độ này bộ tạo dao động kết nối với vi điều khiển PIC 18F4520 qua hai chân OSC1 và OSC2.

- Sơ đồ kết nối giữa vi điều khiển với bộ phát xung ngoài:



Hình 2.8. Sơ đồ kết nối với bộ dao động thạch anh/mạch cộng hưởng ngoài

- Lựa chọn giá trị tụ điện khi sử dụng mạch cộng hưởng thạch anh bọc gốm (Ceramic Resonator).

Chế độ	Tần số	OSC1	OSC2
XT	3.58 MHz	15 pF	15 pF
	4.19 MHz	15 pF	15 pF
	4 MHz	30 pF	30 pF
	4 MHz	50 pF	50 pF

Bảng 2.1. Bảng lựa chọn giá trị tụ điện khi sử dụng Ceramic Resonator

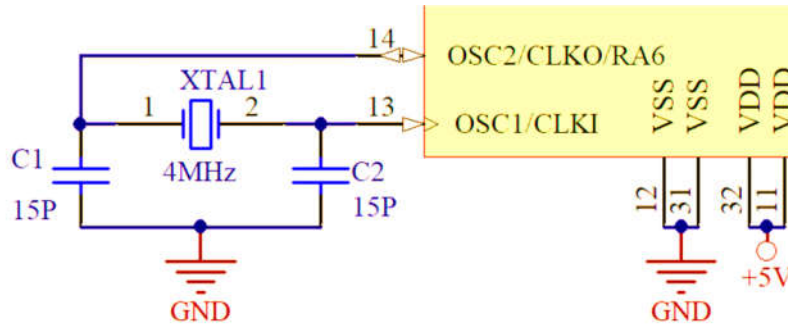
- Lựa chọn giá trị tụ điện khi sử dụng thạch anh chưa có tụ điện (Crystal).

Chế độ	Tần số	Giá trị tụ điện	
		C1	C2
LP	32 kHz	30 pF	30 pF
XT	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	10 MHz	15 pF	15 pF
	20 MHz	15 pF	15 pF
	25 MHz	0 pF	5 pF
	25 MHz	15 pF	15 pF

Bảng 2.2. Bảng lựa chọn tụ điện khi sử dụng thạch anh (Crystal)

- Khởi tạo dao động : Sử dụng bộ tạo dao động trong chế độ HS, sử dụng thạch anh để tạo dao động.

Sử dụng câu lệnh cấu hình: **#pragma config OSC = HS**



Bảng 2.9. Sơ đồ mạch tạo dao động sử dụng thạch anh

2.3.2. Nguồn xung ngoài (External Clock)

Chế độ EC và ECIO sử dụng nguồn xung ngoài làm xung hệ thống và được nối qua cổng NOT trước khi đưa vào chân OSC1/CLKI.

(xem thêm mục 2.4.2 trong giáo trình hoặc mục 2.3 trong tài liệu PIC18F4520 datasheet).

2.3.3. Tạo dao động bằng mạch RC ngoài

Chế độ tạo dao động RC và RCIO của PIC 18F4520 sử dụng mạch tạo dao động RC (một điện trở R_{EXT} và một tụ điện C_{EXT}) để tạo dao động. Chế độ RC và RCIO được sử dụng cho các ứng dụng không nhạy cảm với thời gian để tiết kiệm chi phí thiết kế. (xem thêm mục 2.4.3 trong giáo trình hoặc mục 2.4 trong tài liệu PIC18F4520 datasheet).

2.3.4. Khởi phát xung nội

Vi điều khiển PIC 18F4520 có hai bộ dao động nội, một bộ INTOSC 8Mhz và một bộ dao động INTRC 31Khz, cả hai có thể được thiết lập làm xung của hệ thống. Sử dụng nguồn xung nội sẽ tiết kiệm được chi phí sử dụng bộ dao động bên ngoài, hai chân OSC1/ RA7 và OSC1/RA6 có thể sử dụng làm chân vào/ra. Để lựa chọn sử dụng nguồn xung nội thì phải thiết lập cho khối phát xung hệ thống hoạt động ở chế độ INTIO1 hoặc INTIO2.

2.3.4.1. Nguồn xung nội

- Bộ dao động nội INTOSC 8Mhz sẽ được đưa qua bộ chia tần (Postscaler) để tạo ra dải tần số từ 31 Khz đến 8 Mhz. Sử dụng các bit OSCCON<6:4> để lựa chọn tần số phù hợp cho hệ thống.

- Bộ dao động RC nội INTRC 31Khz không những có thể làm xung cho hệ thống mà nó có làm nguồn xung cho bộ định thời gian bật nguồn (Power-up timer), bộ đảm bảo an toàn nguồn xung (Fail-Safe Clock Monitor), bộ định thời gian Reset hệ thống (Watchdog Timer) và bộ tăng tốc khởi động (Two-Speed Start-up).

2.3.4.2. Chế độ INTIO

Chế độ INTIO sử dụng nguồn xung nội INTOSC hoặc INTRC làm xung hệ thống, chế độ INTIO được chia thành hai chế độ INTIO1 và INTIO2.

- Với chế độ INTIO1, chân OSC2 phát xung $F_{osc}/4$ còn chân OSC1 là chân vào ra số RA7.

- Với chế độ INTIO2, cả hai chân OSC2/RA6 và OSC1/RA7 đều là hai chân vào/ra số.

2.3.5. Vòng khóa pha PLL

Vòng khóa pha PLL (Phase Locked Loop) của vi điều khiển PIC 18F4520 sẽ giúp nhân tần số xung nhịp cấp cho hệ thống lên 4 lần, vi điều khiển sẽ hoạt động nhanh hơn 4 lần so với chế độ thông thường. Có hai chế độ có thể sử dụng vòng khóa pha PLL là HSPLL và chế độ nguồn xung nội INTOSC/PLL, với chế độ INTOSC/PLL chỉ sử dụng vòng khóa pha PLL cho tần số xung nội 8Mhz hoặc 4 Mhz.

2.3.6. Các thanh ghi của bộ phát xung

- Thanh ghi chuyển chế độ bộ phát xung : OSCTUNE
- Thanh ghi điều khiển bộ phát xung OSCCON
- Thanh ghi cấu hình 1 byte cao: CONFIG1H

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1
IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0
bit 7							bit 0

bit 3-0 **FOSC3:FOSC0**: Bit lựa chọn bộ phát xung

11xx = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

101x = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

1001 = Chế độ phát xung nội, chức năng phát xung CLKO trên chân RA6, vào/ra trên chân RA7

1000 = Chế độ phát xung nội, vào/ra trên chân RA6 và RA7

0111 = Chế độ dao động RC ngoài, vào/ra trên chân RA6

0110 = Chế độ HS, cho phép PLL (xung hệ thống được nhân 4)

0101 = Chế độ EC, vào/ra trên chân RA6

0100 = Chế độ EC, phát xung trên chân RA6

0011 = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

0010 = Chế độ HS

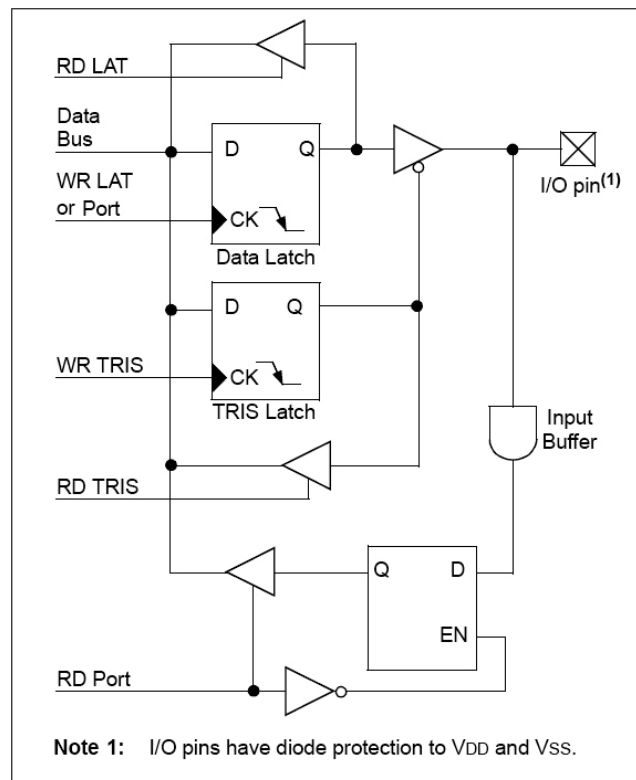
0001 = Chế độ XT

0000 = Chế độ LP

2.4. Các cổng GPIO và các thanh ghi liên quan

Vi điều khiển PIC 18F4520 có 36 chân vào/ra được chia thành 5 cổng là PORTA, PORTB, PORTC, PORTD có 8 chân và PORTE có 4 chân. Các chân vào/ra của vi điều khiển PIC 18F4520 mang nhiều chức năng, nó có thể được thiết lập là chân vào/ra dữ liệu hay là các chân chức năng đặc biệt của các bộ ngoại vi, sử dụng các thanh ghi điều khiển của ngoại vi để lựa chọn chức năng cho các chân.

2.4.1. Cấu trúc chung của một chân vào/ra



Hình 2.10. Cấu trúc chung của một chân vào/ra.

Trong hoạt động vào ra dữ liệu, cần thiết lập thanh ghi **TRIS** và thanh ghi **PORT**, thiết lập thanh ghi **ADCON1** để các chân đầu vào tương tự ANx có đầu vào ra số .

- Thanh ghi **TRISx** là thanh ghi lựa chọn hướng dữ liệu (bằng ‘0’ chiều ra, bằng ‘1’ chiều vào).

Ví dụ: Khi sử dụng chân RB0 có chiều vào, RB1 có chiều ra.

Bit	7	6	5	4	3	2	1	0
TRISB							0	1

Các bit không sử dụng nên thiết lập = 0;

TRISB=0bxxxxxx01; hoặc TRISBbits.TRISB0=1; TRISBbits.TRISB1=0;

- Thanh ghi **PORTx** là thanh ghi dữ liệu, được định địa chỉ theo byte và theo bit, sử dụng để đọc/ghi dữ liệu trên các cổng.

Ví dụ: Thiết lập giá trị RC0=1; RC7=0;

Bit	7	6	5	4	3	2	1	0
PORTC	0							1

Các bit không sử dụng nên thiết lập = 0;

PORTC=0b0xxxxxx1; hoặc PORTCbits.RC0=1; PORTCbits.RC7=0;

- **Thanh ghi ADCON1:** Thiết lập các chân AN0-AN12 là vào/ra số hoặc vào tương tự.

Bit	7	6	5	4	3	2	1	0
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
Giá trị	0	0	0	0	1	1	1	1

ADCON1=0b00001111; //0x0f;

2.4.2. Cổng A – PORTA

PORTA có độ rộng 8 bit, tương ứng với 8 chân được ký hiệu từ RA0 đến RA7. Các chân của PORTA có hai chiều dữ liệu và người lập trình có thể đọc/ghi theo từng bit hoặc cả byte.

Các thanh ghi liên quan đến PORTA

Các thanh ghi liên quan đến PORTA gồm 6 thanh ghi sau:

- PORTA: Là thanh ghi dữ liệu PORTA.
- LATA: Là thanh ghi chốt dữ liệu đầu ra của PORTA.

- TRISA: Là thanh ghi lựa chọn hướng dữ liệu của PORTA (bit tương ứng trên thanh ghi đặt bằng '0' thì chân tương ứng có chiều ra, bằng '1' là chiều vào).
- ADCON1: Là thanh ghi điều khiển A/D, thiết lập các chân vào/ra là số hay tương tự.
- CMCON: Là thanh ghi điều khiển bộ so sánh.
- CVRCON là thanh ghi điều khiển điện áp tham chiếu của bộ so sánh..

(xem thêm mục 2.7.2 trong giáo trình hoặc mục 10.1 trong tài liệu PIC18F4520 datasheet)

2.4.3. Cổng B – PORTB

PORTB có độ rộng 8 bit, tương ứng với 8 chân được ký hiệu từ RB0 đến RB7. Các chân của PORTB có hai chiều dữ liệu và người lập trình có thể đọc/ghi theo từng bit hoặc cả byte. Các chân PORTB <7:4> của PORTB còn được sử dụng làm nguồn ngắt do có sự thay đổi trên mức trên các chân này.

Các thanh liên quan đến PORTB

Có 7 thanh ghi được sử dụng để điều khiển và chọn chức năng cho PORTB:

- PORTB: Là thanh ghi dữ liệu của PORTB.
- LATB: Là thanh ghi chốt dữ liệu của PORTB.
- TRISB: Là thanh ghi hướng dữ liệu của PORTB.
- INTCON: Là thanh điều khiển ngắt.
- INTCON2: Là thanh ghi điều khiển ngắt 2.
- INTCON3: Là thanh điều khiển ngắt 3.
- ADCON1: Là thanh ghi điều khiển bộ biến đổi A/D. Thanh ghi này được sử dụng để chọn các chân AN0 đến AN12 là các chân vào/ra số hay tương tự.

(xem thêm mục 2.7.3 trong giáo trình hoặc mục 10.2 trong tài liệu PIC18F4520 datasheet)

2.4.4. Cổng C – PORTC

PORTC có độ rộng 8 bit, tương ứng với 8 chân được ký hiệu từ RC0 đến RC7. Các chân của PORTC có hai chiều dữ liệu và người lập trình có thể đọc/ghi theo từng bit hoặc cả byte.

Các thanh ghi liên quan đến PORTC

Có 3 thanh ghi được sử dụng để điều khiển và chọn chức năng cho PORTC:

- PORTC: Là thanh ghi dữ liệu của PORTC.

- LATC: Là thanh ghi chốt dữ liệu của PORTC.
- TRISC: Là thanh ghi hướng dữ liệu của PORTC.

(xem thêm mục 2.7.4 trong giáo trình hoặc mục 10.3 trong tài liệu PIC18F4520 datasheet)

2.4.5. Cổng D – PORTD

PORTD có độ rộng 8 bit, tương ứng với 8 chân được ký hiệu từ RD0 đến RD7. Các chân của PORTD có hai chiều dữ liệu và người lập trình có thể đọc/ghi theo từng bit hoặc cả byte. Các chân của PORTD có chức năng vào/ra dữ liệu và chức năng đặc biệt của các ngoại vi, đặc biệt PORTD còn được sử dụng để vào/ra dữ liệu song song với bên ngoài khi kết hợp với các chân RD, CS và WR.

Các thanh liên quan đến PORTD

Có 5 thanh ghi được sử dụng để điều khiển và chọn chức năng cho PORTD:

- PORTD: Là thanh ghi dữ liệu của PORTD.
- LATD: Là thanh ghi chốt dữ liệu của PORTD.
- TRISD: Là thanh ghi hướng dữ liệu của PORTD.
- TRISE : Thanh ghi hướng dữ liệu của PORTE. Thanh ghi này có bit PSPMODE được sử dụng để chọn chế độ vào/ra song song PSP hay vào/ra dữ liệu trên PORTD.
- CCP1CON: Thanh ghi điều khiển ECCP. Thanh ghi này có 2 bit P1M1:P1M0 liên quan tới PORTD, nó sử dụng để cấu hình đầu ra cho bộ PWM tăng cường.

(xem thêm mục 2.7.5 trong giáo trình hoặc mục 10.4 trong tài liệu PIC18F4520 datasheet)

2.4.6. Cổng E – PORTE

PORTE có độ rộng 4 bit, tương ứng với 4 chân được ký hiệu từ RE0 đến RE3. Các RE0/RD/AN5, RE1/WR/AN6 và RE2/CS/AN7 của PORTE có thể được cấu hình với chức năng vào/ra dữ liệu hoặc các chức năng đặc biệt khác của các ngoại vi, riêng chân thứ bốn MCLR/VPP/RE3 chỉ có chiều vào dữ liệu và các chức năng đặc biệt khác..

Các thanh liên quan đến PORTE

Có 4 thanh ghi được sử dụng để điều khiển và chọn chức năng cho PORTE:

- PORTE: Là thanh ghi dữ liệu của PORTE.
- LATE: Là thanh ghi chốt dữ liệu đầu ra của PORTE.
- TRISE: Là thanh ghi hướng dữ liệu của PORTE, riêng chân RE3 mặc định có chiều vào.

- ADCON1: là thanh ghi điều khiển A/D. Được sử dụng để lựa chọn vào/ra số hay tương tự cho các chân của PORTE.

(xem thêm mục 2.7.6 trong giáo trình hoặc mục 10.5 trong tài liệu PIC18F4520 datasheet)

2.5. Lập trình và mô phỏng ghi/đọc theo byte, theo bit

- *Lệnh điều khiển vào dữ liệu theo byte*

```
x = PORTD;    // x là một biến kiểu byte
```

- *Lệnh điều khiển ra dữ liệu theo byte*

```
PORTD = x;    // x là một biến kiểu byte
```

- *Lệnh điều khiển vào dữ liệu theo bit*

```
x = PORTDbits.RD0;    // x là một biến kiểu bit
```

Hoặc:

```
#define SW PORTBbits.RB2 // định nghĩa RB2 = SW  
x = SW;    // lệnh đọc
```

- *Lệnh điều khiển ra dữ liệu theo bit*

```
PORTDbits.RD0 = x;    // x là một biến kiểu bit
```

Hoặc:

```
#define LED PORTCbits.RC2 // định nghĩa RC2 = LED  
LED = x;    // lệnh xuất
```

2.5.1. Khung chương trình

Khai báo thư viện định danh phần cứng: `#include<p18f4520.h>`

Cấu hình vi điều khiển:

- + Sử dụng bộ tạo dao động chế độ HS:

```
#pragma config OSC = HS
```

- + Sử dụng chân RE3 làm chân Reset:

```
#pragma config MCLRE = ON
```

- + Không dùng Watchdog timer:

```
#pragma config WDT=OFF
```

- + Không sử dụng chế độ nguồn cấp từ mạch nạp ICSP:

```
#pragma config LVP=OFF
```

...

- Khai báo biến số, hằng số, cấu trúc, chương trình con, int x;
- Viết chương trình chính

Khai báo biến số, hằng số, cấu trúc, chương trình con:

```
int x; char m[10];
void high_isr (void)
{    //các câu lệnh    }
```

Viết chương trình chính:

```
void main()
{ //các câu lệnh }
```

Khung chương trình cụ thể như sau:

```
#include<p18f4520.h>

#pragma config OSC = HS
#pragma config MCLRE = ON
#pragma config WDT=OFF
#pragma config LVP = OFF

int x; char m[10];

void high_isr (void)
{    //các câu lệnh    }


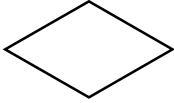


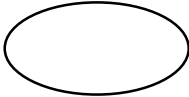
void main()
{ //các câu lệnh }
```

TIPs: Sử dụng “Tài liệu tra cứu PIC” hoặc tài liệu “MPLAB® C18

C COMPILER USER’S GUIDE” để viết khung chương trình.

2.5.2. Lập lưu đồ thuật toán

Lưu đồ thuật toán là một biểu đồ thể hiện trình tự và các bước để tạo ra một chương trình cụ thể. Bảng dưới đưa ra các hình dạng, ký hiệu và ý nghĩa của các thành phần trong lưu đồ thuật toán.

Hình dạng	Ý nghĩa
	Xử lý
	Quyết định, điều kiện
	Luồng xử lý
	Gọi chương trình con, hàm
	Bắt đầu, kết thúc (Begin, end)

2.5.3. Viết các lệnh trong chương trình chính.

a. Khởi tạo

Viết các lệnh khởi tạo giá trị thích hợp cho thanh ghi TRIS và ADCON1.

b. Các lệnh theo đề bài

Các lệnh được viết ở chương trình chính sẽ là “các công việc tuần tự, lặp lại”.

Ví dụ với yêu bài toán sau:

Vẽ mạch điện và lập trình theo yêu cầu:

Mạch điện gồm 1 nút nhấn **BTN1** kết nối với chân RB3, 1 Led đơn **D1** kết nối với chân RB7. Lập trình khi nhấn nút **BTN1** thì Led **D1** sáng, khi không nhấn **BTN1** thì Led **D1** tắt.

- Vẽ lưu đồ thuật toán và viết chương trình theo yêu cầu:

- Khi bật nguồn hoặc reset: **LED1** tắt;
- Nhấn **BNT1** lần **LED1** sáng, khi không nhấn **LED1** tắt.

Lập lưu đồ thuật toán:

