# CHƯƠNG 2: TỔ CHỨC BỘ NHỚ, THANH GHI

### GIỚI THIỆU

Ở chương này khảo sát tổ chức bộ nhớ bên trong, các thanh ghi của vi điều khiển 8 bit. Sau khi kết thúc chương này thì người đọc có thể biết tổ chức bộ nhớ bên trong, chức năng của từng loại bộ nhớ, tên và chức năng của các thanh ghi đặc biệt.

### KIẾN TRÚC BỘ NHỚ

Có 2 loại kiến trúc bộ nhớ cơ bản là kiến trúc Von Neumann và Harvard

A picture containing text, rectangle, design

Description automatically generated

A picture containing text, screenshot, design

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Tên kiến trúc | Tính chất | Ưu điểm | Khuyết điểm |
| Kiến trúc Von Neumann | - Bộ nhớ giao tiếp với CPU thông qua 1 bus dữ liệu 8 bit  -Bộ nhớ có các ô nhớ chứa dữ liệu 8 bit  -Bộ nhớ vừa lưu trữ chương trình và dữ liệu. | Kiến trúc đơn giản | -Do chỉ có 1 bus nên tốc độ truy suất chậm  -Khó thay đổi dung lượng lưu trữ của ô nhớ |
| Kiến trúc Harvard | -Bộ nhớ được tách ra làm 2 loại bộ nhớ độc lập: bộ nhớ lưu chương trình và bộ nhớ lưu dữ liệu  -CPU giao tiếp với 2 bộ nhớ độc lập nên cần 2 bus độc lập. -Có thể thay đổi số bit lưu trữ của từng bộ nhớ mà không ảnh hưởng lẫn nhau. | -Do chỉ có 2 bus nên tốc độ truy suất nhanh  -Tùy ý thay đổi số bit của ô nhớ. | Kiến trúc phức tạp |

### TỔ CHỨC BỘ NHỚ CỦA VI ĐIỀU KHIỂN PIC 16F887

#### TỔ CHỨC BỘ NHỚ CHƯƠNG TRÌNH VÀ NGĂN XẾP

##### Bộ nhớ chương trình

Bộ nhớ chương trình của PIC16F8xx có **dung lượng 8K** được chia làm 4 trang bộ nhớ, **mỗi trang 2K**

A picture containing text, receipt, number, font

Description automatically generated

Bộ nhớ có dung lượng 8K ô nhớ sẽ có 13 bit địa chỉ, để quản lý địa chỉ của bộ nhớ chương trình thì do thanh ghi bộ đếm chương trình (Program Counter - PC) đảm nhận.

##### Thanh ghi PC và PCLATH

Thanh ghi bộ đếm chương trình PC sẽ quản lý địa chỉ của bộ nhớ chương trình, thanh ghi PC có độ dài 13 bit sẽ quản lý 8192 ô nhớ tương đương với 8K ô nhớ. Số bit của thanh ghi PC có mối quan hệ với dung lượng bộ nhớ chương trình.

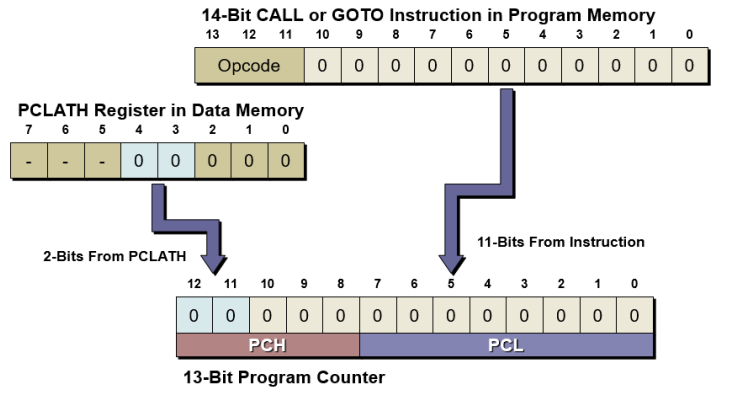
Với vi điều khiển PIC 16F887 thì mỗi ô nhớ chương trình có **độ dài 14 bit.**

Khi PIC **bị reset** thì thanh ghi PC **có giá trị là 0000H** và vi điều khiển PIC sẽ bắt đầu thực hiện chương trình tại **địa chỉ 0000H**

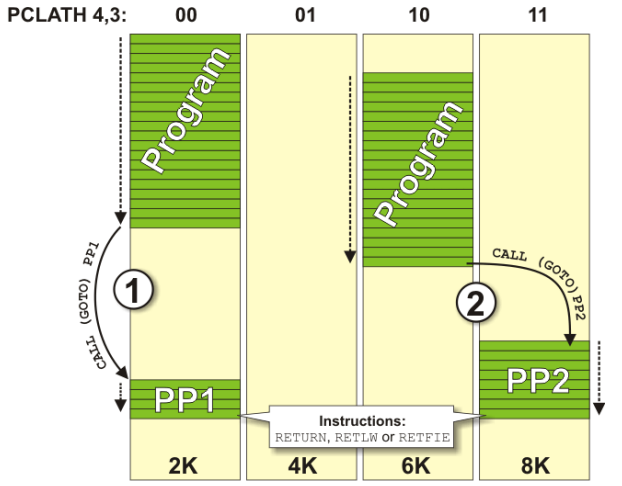
Khi có bất kỳ ngắt nào tác động thì vi điều khiển PIC sẽ thực hiện chương trình phục vụ ngắt tại **địa chỉ 0004H.**

Việc phân chia theo trang bộ nhớ chỉ có tác dụng đối với **lệnh nhảy** và **lệnh gọi** chương trình con, khi truy xuất **trong nội bộ mỗi trang** thì chỉ có **11 bit địa chỉ thấp** **thay đổi**, **2 bit địa chỉ cao** **giữ nguyên.**

Khi thực hiện **lệnh gọi** hoặc **lệnh nhảy** thì từ mã lệnh 14 bit trong đó chứa **11 bit địa chỉ thấp** và **2 bit địa chỉ cao** lấy từ **thanh ghi PCLATH**

****

Khi nơi nhảy đến hoặc khi gọi chương trình con nằm trong cùng 1 trang thì lệnh sẽ viết ngắn gọn chỉ có 1 lệnh và mã lệnh chỉ là 1 từ bao gồm cả mã lệnh và địa chỉ 11 bit là địa chỉ của trang hiện tại



Khi nhảy hoặc gọi trong cùng 1 trang thì chỉ thay đổi 11 bit địa chỉ trong thanh ghi PC nên chỉ nhảy đến trong phạm vi 1 trang 2K. Hai bit địa chỉ thứ 12 và 13 lưu trong thanh ghi PCLATH giữ nguyên.

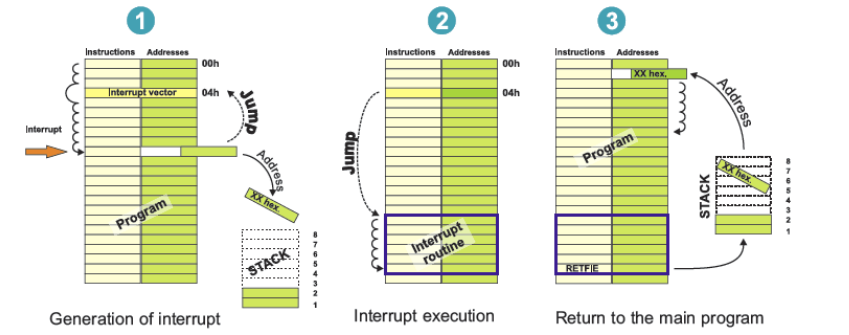
Khi nơi nhảy đến hoặc khi gọi chương trình con nằm ở trang khác thì trước khi thực hiện lệnh nhảy hoặc gọi thì ta phải biết được nơi nhảy đến nằm ở trang nào để tiến hành thay đổi 2 bit địa chỉ cao trong thanh ghi PCLATH cho đúng rồi mới thực hiện lệnh nhảy hoặc gọi. Do có nhiều lệnh hơn nên mã lệnh dài hơn

* Nói tóm lại cùng trang thì không thay đổi **2 bit địa chỉ cao trong thanh ghi PCLATH** còn khác trang thì thay đổi

##### Bộ nhớ ngăn xếp

Trong các họ vi điều khiển khác thì bộ nhớ ngăn xếp **dùng chung** với bộ nhớ dữ liệu, ưu điểm là cấu trúc đơn giản, khuyết điểm là việc dùng chung nếu không biết giới hạn sẽ lấn chiếm lẫn nhau và làm mất dữ liệu lưu trong bộ nhớ ngăn xếp, khi đó dẫn đến vi điều khiển thực hiện sai chương trình.

**Ở vi điều khiển PIC** thì nhà thiết kế **tách** bộ nhớ ngăn xếp **độc lập** với bộ nhớ dữ liệu và chỉ để dùng lưu địa chỉ trở về của thanh ghi PC khi thực hiện lệnh gọi chương trình con và khi thực hiện ngắt



Bộ đếm ngăn xếp(Stack) giống như 1 nơi được **mượn** để **lưu trữ địa chỉ của lệnh**. Khi thực hiện xong yêu cầu ngắt thì địa chỉ của lệnh sẽ được **trả lại**

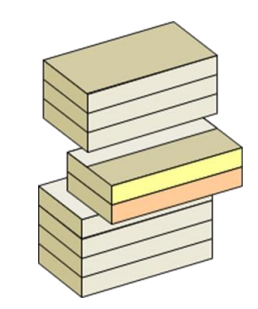
Do chỉ có 8 ô nhớ nên khi thực hiện các chương trình con lồng vào nhau tối đa là **8 cấp (Mượn có giới hạn)**

Tránh **vùng nhớ bắt đầu** tại **địa chỉ 0004H**

#### MÃ LỆNH 14 BIT

##### Vi điều khiển khác

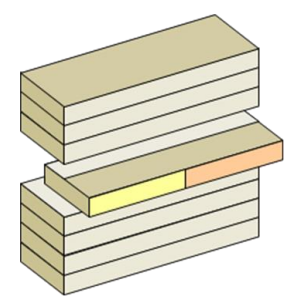
Với các vi điều khiển 8 bit của các hãng khác thì bộ nhớ chương trình tổ chức theo đơn vị là byte, mỗi ô nhớ lưu trữ dữ liệu 1 byte



Nếu mã lệnh 2 byte gồm 1 byte mã lệnh và 1 byte dữ liệu hay địa chỉ thì dùng 2 ô nhớ liên tiếp để lưu và khi CPU đọc mã lệnh để thực hiện lệnh thì CPU phải thực hiện 2 lần đọc - mỗi lần 1 byte.

##### Vi điều khiển PIC

Với vi điều khiển PIC thì mỗi ô nhớ của bộ nhớ chương trình có thể **lưu trữ dữ liệu nhiều bit** bao gồm **cả mã lệnh và dữ liệu.** Khi CPU đọc mã lệnh để thực hiện thì vi điều khiển CPU chỉ thực hiện 1 lần đọc cả mã lệnh và dữ liệu.



Vậy với tổ chức bộ nhớ của vi điều khiển PIC thì **tiết kiệm được 1 chu kỳ đọc dữ liệu** vì thế vi điều khiển PIC sẽ có **tốc độ thực hiện chương trình nhanh hơn.**

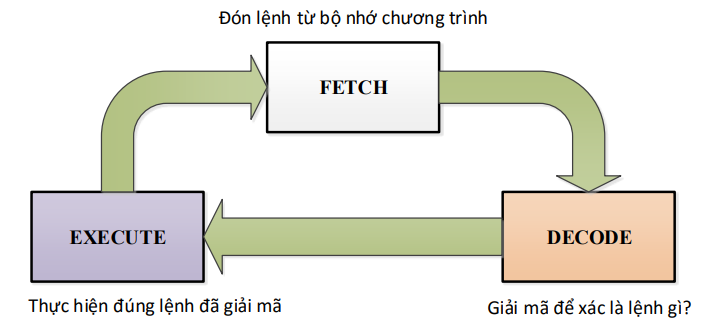
Tập lệnh của vi điều khiển PIC là **tập lệnh rút gọn** nên chỉ dùng có **6 bit** nhị phân để mã hóa các lệnh, cùng với dữ liệu xử lý là byte - 8 bit nên **tổng cộng là 14 bit.**

#### CẤU TRÚC PIPELINE

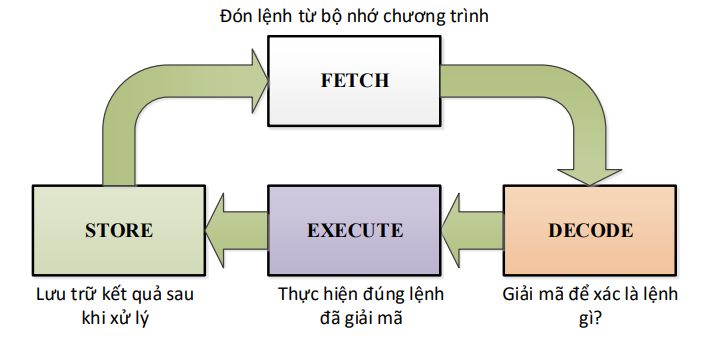
##### Các bước thực hiện lệnh của các vi xử lý

Các vi xử lý 8 bit như 8085, Z80 trước đây thì quá trình thực hiện lệnh theo nhiều nguồn tài liệu được chia thành 3 bước hoặc 4 bước hoặc 5 bước.

**3 bước**

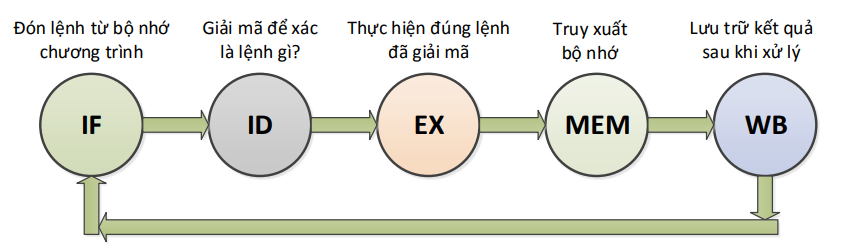
****

**4 bước**

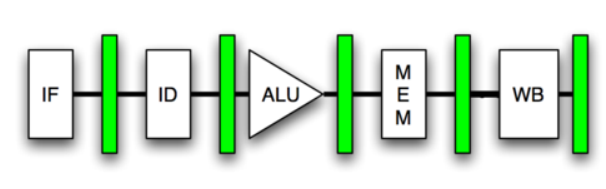
****

**5 bước**

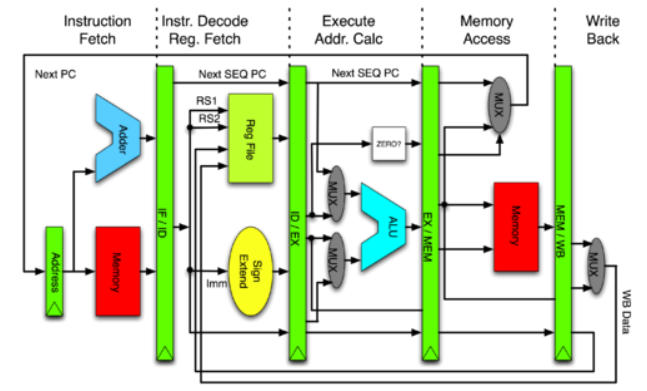
* **Quá trình**

****

* **Ký hiệu**

****

* **Các bước thực hiện chi tiết**

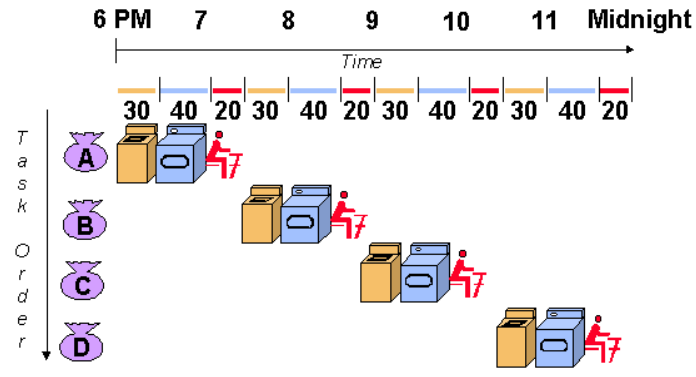
****

Quá trình 5 bước thực ra là cách trình bày đầy đủ nhất

Mỗi một lệnh được thực hiện đầy đủ các bước trên, mỗi bước có thể thực hiện với thời gian tính theo chu kỳ bộ dao động thường là 1 chu kỳ. Lệnh đầy đủ năm bước sẽ thực hiện hết 5 chu kỳ, tương tự cho các lệnh khác.

Vi điều khiển có tập lệnh và cách thức vừa trình bày ở trên được gọi là tập lệnh có cấu trúc phức tạp CISC (Complex Instruction Set Computer).

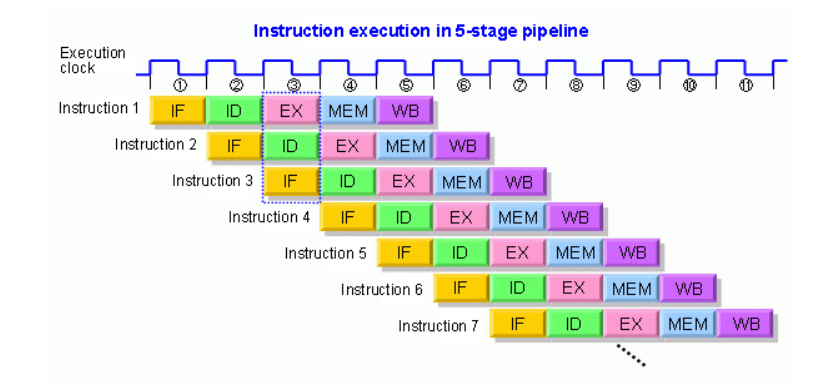
Ở vi xử lý hay vi điều khiển có tập lệnh CISC thì thực hiện từng lệnh: thực hiện lệnh này xong thì mới bắt đầu thực hiện lệnh tiếp theo. Với cách thực hiện này thì dẫn đến hiệu suất không cao hay tốc độ thực hiện chậm



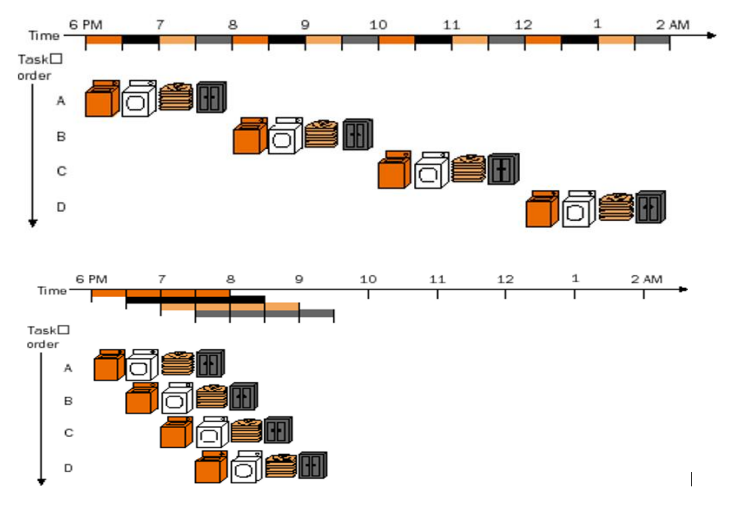
##### Thực hiện lệnh theo cấu trúc PIPELINE

Với sự nghiên cứu của còn người nhằm làm cho vi xử lý thực hiện càng nhanh càng tốt thì có rất nhiều vấn đề cần cải tiến, cải tiến về cấu trúc bộ nhơ như đã trình bày ở trên cũng là một cách, tăng số bit của từ dữ liệu thành 14 bit, 16 bit, 32 bit, …, cải tiến tập lệnh, cải tiến quy trình thực hiện lệnh cũng là một cách.

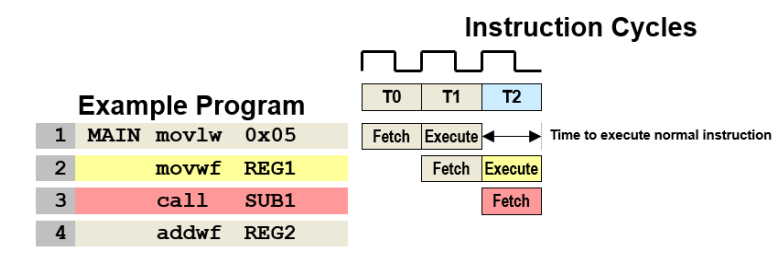
Phần này trình bày cách cải tiến quy trình thực hiện lệnh theo cấu trúc pipeline, tạm dịch là đường ống lệnh. Để thực hiện cấu trúc pipeline thì các nhà nghiên cứu đã cải tiến quy trình đón lệnh, giải mã và thực hiện lệnh sao cho chúng gối đầu lên nhau



So sánh hiệu quả của cấu trúc pipeline mang lại về thời gian so với không dùng pipeline



Ở vi điều khiển PIC thì quá trình thực hiện lệnh được rút gọn theo cấu trúc RISC (Reduce Instruction Set Computer) chỉ còn có **2 bước** là đón lệnh và thực hiện lệnh. Khi áp dụng vào cấu trúc pipeline thì tổ chức có dạng như sau



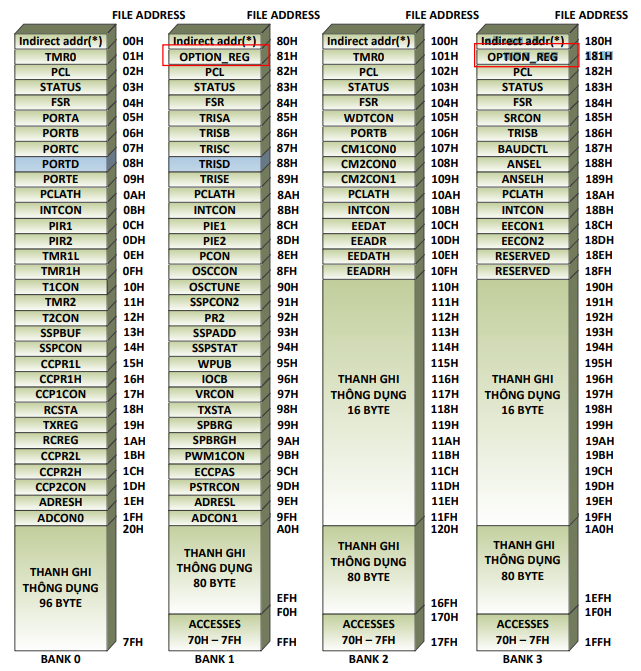
#### KHẢO SÁT BỘ NHỚ DỮ LIỆU VÀ THANH GHI TRẠNG THÁI

##### Cấu trúc bộ nhớ dữ liệu

Bộ nhớ dữ liệu được phân chia thành 4 Bank, mỗi bank có 128byte bao gồm một số thanh ghi chức năng đặc biệt, còn lại là các ô nhớ thông dụng có chức năng lưu trữ dữ liệu.

Bộ nhớ RAM có 512 ô nhớ và khi đó số bit địa chỉ được dùng là 9 bit (2 mũ 9 = 512), mỗi 1 ô nhớ có 1 địa chỉ duy nhất.

Toàn bộ các ô nhớ của bộ nhớ dữ liệu được gọi là File thanh ghi.



Các thanh ghi có chức năng đặc biệt nằm ở vùng địa chỉ thấp, các ô nhớ còn lại không có gì đặc biệt nằm ở cùng địa chỉ bên trên các thanh ghi chức năng đặc biệt – xem như các ô nhớ RAM dùng để lưu dữ liệu. Tất cả các bank thanh ghi đều chứa những thanh ghi đặc biệt

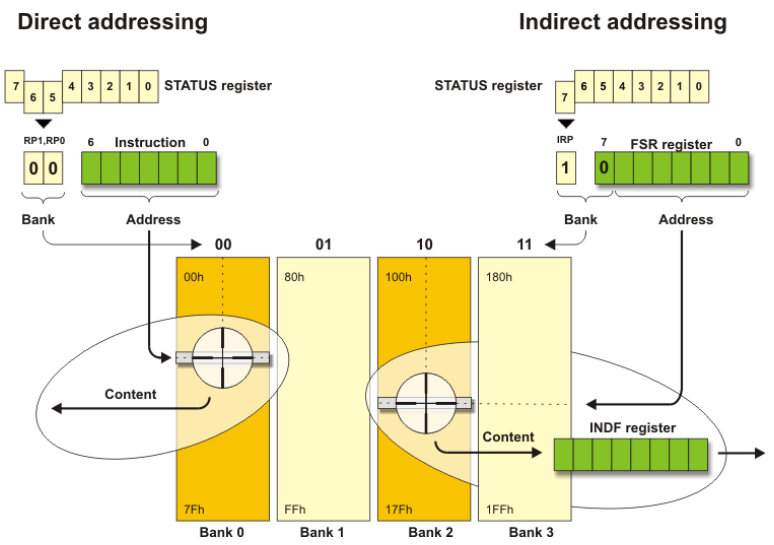
Theo hình thì bộ nhớ dữ liệu được chia làm 4 bank thanh ghi, mỗi bank có 128, tổng cộng là 512 ô nhớ, nhưng do có 1 số thanh ghi có chức năng đặc biệt ở bank nào cũng có nên làm **giảm số lượng**. Ví dụ thanh ghi trạng thái (status) ở 4 bank đều có, thay vì 4 thanh ghi thì chỉ xem là 1, tương tự cho các thanh ghi khác. Số lượng thực nhỏ hơn 512 ô nhớ. Số lượng thực là 368 ô nhớ.

Do sử dụng cấu trúc pipeline nên tập lệnh phải rút gọn dẫn đến bộ nhớ chương trình được tổ chức theo trang để đơn giản khi truy xuất trong phạm vi 1 trang, tương tự bộ nhớ dữ liệu cũng được phân chia thành 4 bank để đáp ứng cho các yêu cầu trên.

Bộ nhớ RAM 4 bank có 2 cách truy xuất bộ nhớ dữ liệu: truy xuất trực tiếp và truy xuất gián tiếp.

###### Truy xuất trực tiếp

**Khi truy xuất trực tiếp**: thì các lệnh chỉ được phép truy xuất 1 ô nhớ RAM hay 1 thanh ghi của 1 bank. Địa chỉ 9 bit thì 2 bit địa chỉ cao thứ 7 và thứ 8 cố định, 7 bit chỉ thấp từ 6 đến 0 được phép thay đổi nên chỉ truy xuất trong phạm vi 128 byte tương ứng 1 bank



Truy xuất trực tiếp và truy xuất gián tiếp

Hai bit địa chỉ cao chính lưu ở 2 bit có tên là **RP1 và RP0** nằm trong thanh ghi trạng thái. Nếu muốn truy xuất các ô nhớ hay thanh ghi nằm ở bank khác thì phải đổi bank bằng cách **thay đổi địa chỉ** của 2 bit này. Cách truy xuất này **giống như trang bộ nhớ chương trình**.

###### Truy xuất gián tiếp

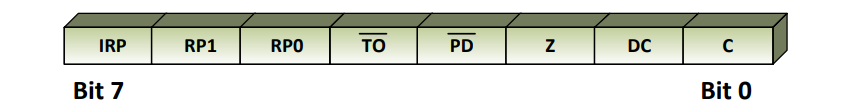
**Khi truy xuất gián tiếp**: thì các lệnh truy xuất được phép truy xuất 1 ô nhớ RAM hay 1 thanh ghi nằm trong 2 bank: bank 0, 1 hoặc bank 2, 3. Địa chỉ 9 bit thì bit địa chỉ thứ 8 cố định, 8 bit thấp được phép thay đổi nên truy xuất trong phạm vi 256 byte **tương ứng 2 bank**, xem hình phần truy xuất gián tiếp.

Bit địa chỉ thứ 8 chính lưu ở **bit IRP** nằm trong thanh ghi trạng thái.

Cách truy xuất trực tiếp thường dùng để truy xuất một vài ô nhớ, cách truy xuất gián tiếp dùng để truy xuất một vùng nhớ liên tục để phục vụ cho cách viết chương trình ở dạng vòng lặp.

##### Thanh ghi trạng thái – STATUS REGISTER ĐỊA CHỈ 03H, 83H, 103H, 83H

Thanh ghi trạng thái chứa trạng thái của khối ALU, trạng thái Reset và các bit chọn bank bộ nhớ dữ liệu. Cấu trúc thanh ghi trạng thái như hình



Chức năng của các bit trong thanh ghi trạng thái:

Bit 7 IRP: bit lựa chọn thanh ghi (dùng địa chỉ gián tiếp).

* 1 = bank 2, 3 (100h-1FFh)
* 0 = bank 0, 1 (000h- 0FFh)

Bit 6-5 RP1:RP0: các bit lựa chọn thanh ghi (dùng địa chỉ trực tiếp)

* 11 = bank 3 (180h-1FFh)
* 10 = bank 2 (100h- 17Fh)
* 01 = bank 1 (80h- FFh)
* 00 = bank 0 (00h- 7Fh)

Bit 4 TO : Time-out bit (Bit thời gian chờ)

* 1 = sau khi mở nguồn, lệnh CLRWDT hoặc SLEEP
* 0 = thời gian chờ của WDT được thực hiện

Bit 3 PD : Power-down bit (bit tắt nguồn)

* 1= sau khi mở nguồn hoặc bằng lệnh CLRWDT
* 0= thực thi lệnh SLEEP

Bit 2 Z: Zero bit (bit 0)

* 1 = khi kết quả bằng 0.
* 0 = khi kết quả khác 0.

Bit 1 DC: Digit carry/ borrow bit (các lệnh ADDWF, ADDLW, SUBLW, SUBWF) (bit tràn / mượn)

* 1 = khi cộng 4 bit thấp bị tràn.
* 0 = khi cộng 4 bit thấp không bị tràn.

Bit 0 C: Carry/ borrow bit (các lệnh ADDWF, ADDLW, SUBLW, SUBWF)

* 1 = khi kết quả phép toán có tràn.
* 0 = khi kết quả phép toán không bị tràn.

**Chú ý:** Nếu phép toán trừ thì trạng thái của cờ C như sau: nếu phép trừ lớn hơn 0 thì cờ C bằng 0, nếu kết quả trừ nhỏ hơn hay bằng 0 thì cờ C bằng 1.

#### BỘ NHỚ DỮ LIỆU EEPROM

Bộ nhớ dữ liệu Eeprom có dung lượng 256 byte dùng để lưu dữ liệu quan trọng khi mất điện thì dữ liệu này vẫn còn.

Cách thức ghi dữ liệu vào bộ nhớ Eeprom sẽ được trình bày ở phần bộ nhớ Eeprom.

#### TÓM TẮT

Vi điều khiển PIC đang khảo sát có 4 loại bộ nhớ được tích hợp bên trong gồm:

* Bộ nhớ chương trình có dung lượng 8k word dùng để lưu chương trình và thanh ghi chứa địa chỉ để quản lý bộ nhớ chương trình là PC. Chương trình điều khiển sau ki viết xong và được biên dịch thành công sẽ được nạp vào vùng nhớ này để vi điều khiển thực thi.
* Bộ nhớ ngăn xếp có 8 cấp hay 8 ô nhớ, chỉ dùng để lưu địa chỉ trở về trong thanh ghi PC khi thực hiện lệnh gọi chương trình con và ngắt. Người dùng không thể dùng vùng nhớ này để lưu dữ liệu.
* Bộ nhớ RAM xem như có 512 byte được chia làm 4 bank, bao gồm các thanh ghi đã có chức năng và chưa có chức năng là các ô nhớ dùng để lưu dữ liệu phục vụ cho việc viết chương trình điều khiển.
* Bộ nhớ dữ liệu Eeprom có dung lượng 256 byte dùng để lưu dữ liệu quan trọng khi mất điện thì dữ liệu này vẫn còn.

