

CƠ SỞ LÝ THUYẾT MSP430

I. TỔNG QUAN VỀ CẤU TRÚC VÀ CHỨC NĂNG HỌ MSP430

I.1. Giới thiệu

MSP430 chứa 16 bit RISC CPU, các ngoại vi và hệ thống bộ định thời linh hoạt được kết nối với nhau theo cấu trúc VON-NEUMANN, có các Bus liên kết như: Bus địa chỉ bộ nhớ (MAB), Bus dữ liệu bộ nhớ (MDB). Đây là một bộ xử lý hiện đại với các mô đun bộ nhớ tương tự và những kết nối ngoại vi tín hiệu số, MSP430 đã đưa ra được những giải pháp tốt cho những nhu cầu ứng dụng với tín hiệu hỗn tạp.

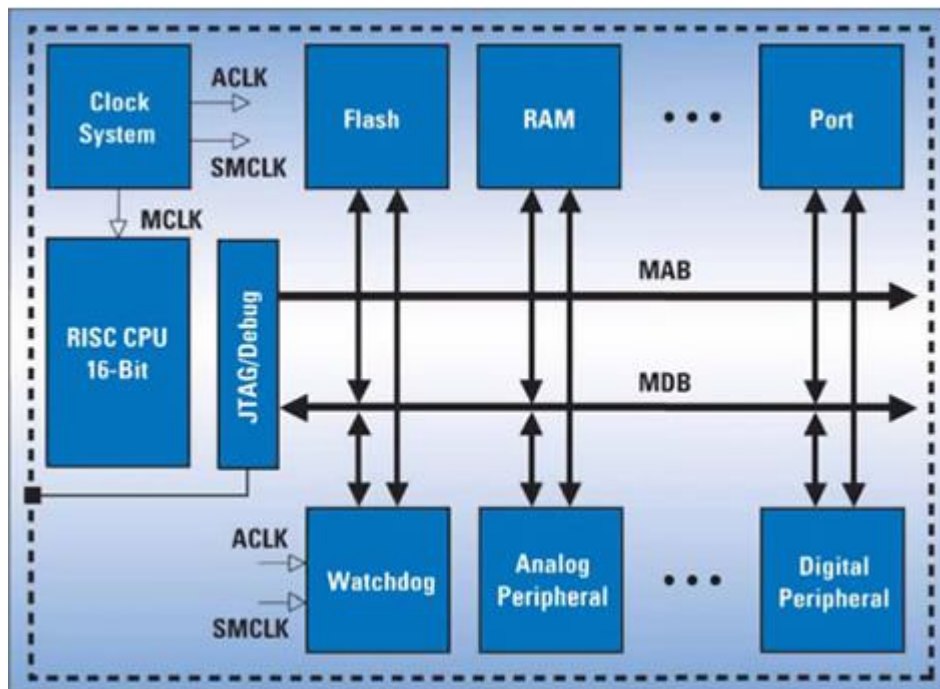
MSP430 có một số phiên bản như: MSP430x1xx, MSP430x2xx, MSP430x3xx, MSP430x4xx, MSP430x5xx. Dưới đây là những đặc điểm tổng quát của họ vi điều khiển MSP430:

- + Cấu trúc sử dụng nguồn thấp giúp kéo dài tuổi thọ của Pin
 - Duy trì 0.1 μ A dòng nuôi RAM.
 - Chỉ 0.8 μ A real- time clock.
 - 250 μ A/ MIPS.
- + Bộ tương tự hiệu suất cao cho các phép đo chính xác
 - 12 bit hoặc 10 bit ADC-200 kskp, cảm biến nhiệt độ, V_{ref} .
 - 12 bit DAC.
 - Bộ giám sát điện áp nguồn.
- + 16 bit RISC CPU cho phép được nhiều ứng dụng, thể hiện một phần ở kích thước Code lập trình.
 - Thanh ghi lớn nên loại trừ được trường hợp tắt nghẽn tập tin khi đang làm việc.
 - Thiết kế nhỏ gọn làm giảm lượng tiêu thụ điện và giảm giá thành.
 - Tối ưu hóa cho những chương trình ngôn ngữ bậc cao như C, C++
 - Có 7 chế độ định địa chỉ.
 - Khả năng ngắt theo véc tơ lớn.
- + Trong lập trình cho bộ nhớ Flash cho phép thay đổi Code một cách linh hoạt, phạm vi rộng, bộ nhớ Flash còn có thể lưu lại như nhật ký của dữ liệu.

I.2. Hệ thống định thời (Clock) linh hoạt

Hệ thống Clock được thiết kế một cách đặc biệt cho những ứng dụng sử dụng nguồn cung cấp là Pin. Một bộ tạo xung nhịp phụ tần số thấp (ACLK) được cung cấp trực tiếp từ một bộ dao động thạch anh 32 KHz. ACLK được sử dụng như là một Real-time Clock nền để kích hoạt các tính năng. Một bộ dao động kỹ thuật số tốc độ cao (DCO) có thể làm nguồn xung đồng hồ chính (MCLK) sử dụng cho CPU và những kết nối ngoại vi tốc độ cao. Bởi thiết kế này, DCO có thể hoạt động ổn định 1MHz trong thời gian ít hơn 2 μ S. MSP430 được thiết kế dựa trên những giải pháp có hiệu quả sử dụng một RISC CPU 16 bit hiệu suất cao.

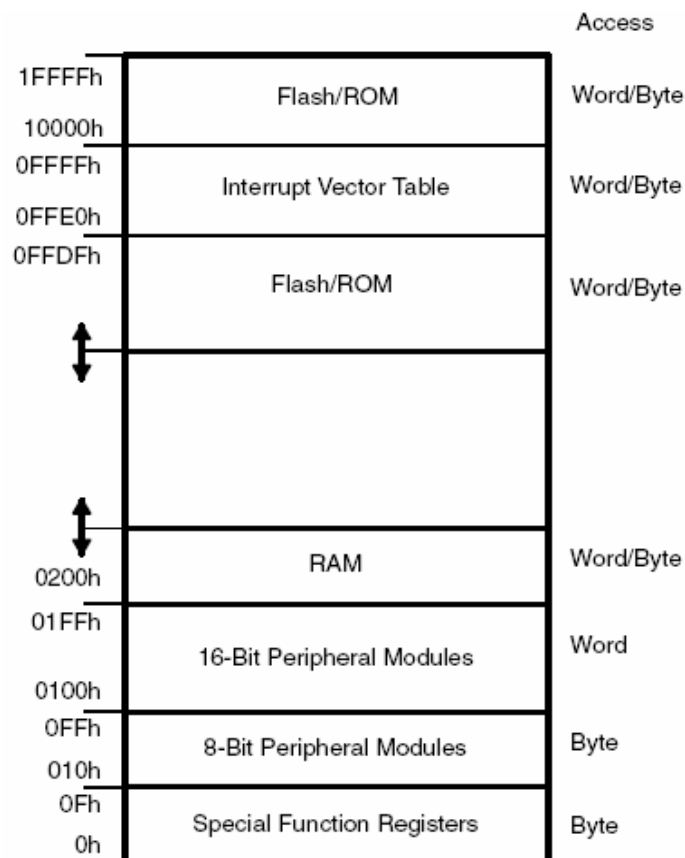
- + Bộ định thời phụ tần số thấp: Hoạt động ở chế độ sẵn sàng sử dụng nguồn cực thấp.
- + Bộ định thời chính (Master Clock) tốc độ cao: Hoạt động xử lý tín hiệu hiệu suất cao.



Hình I.1: Sơ đồ cấu trúc của MSP430

I.3. Các vùng địa chỉ

MSP430 được thiết kế theo cấu trúc Von-Neumann có một vùng địa chỉ được chia thành nhiều vùng như là thanh ghi hàm đặc biệt (SFRs), những ngoại vi, RAM, bộ nhớ Flash/ROM.



Hình I.2: Sơ đồ bộ nhớ của MSP430

I.3.1. Flash/ROM

Địa chỉ bắt đầu của Flash/ROM phụ thuộc vào độ lớn của Flash/ROM và còn tùy thuộc vào từng họ vi điều khiển. Địa chỉ kết thúc của Flash/ROM là 0x1FFFFh. Flash/ROM có thể sử dụng cho cả mã chương trình và dữ liệu. Những bảng Byte hoặc Word có thể được tồn trữ và sử dụng ngay trong Flash/ROM mà không cần copy vào RAM trước khi sử dụng chúng.

Những bảng véc tơ được ánh xạ đến 16 Word phía trên của vùng địa chỉ Flash/ROM với ưu tiên ngắt cao nhất ở vùng địa chỉ cao nhất của Flash/ROM.

I.3.2. RAM

RAM bắt đầu ở địa chỉ 0200h và giới hạn cuối cùng tùy thuộc vào kích thước của RAM. RAM có thể sử dụng cho cả mã chương trình và dữ liệu.

I.3.3. Các module ngoại vi

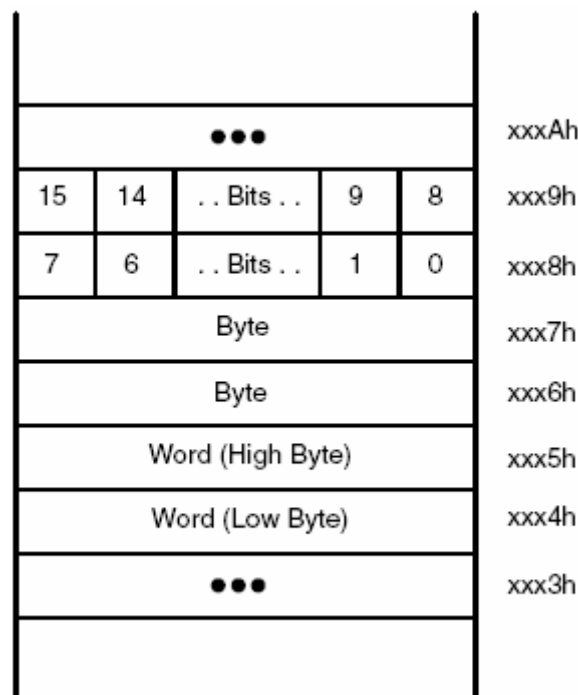
Trong vùng không gian địa chỉ của MSP430 có 2 vùng địa chỉ dành cho những Mô đun ngoại vi. Vùng địa chỉ từ 0100 đến 01FFh sử dụng dành riêng cho những mô đun ngoại vi 16 Bit. Vùng địa chỉ từ 010 đến 0FFh sử dụng dành riêng cho những mô đun ngoại vi 8 Bit.

I.3.4. Thanh ghi hàm đặc biệt

SFRs liên quan nhiều đến sự cho phép những tính năng của một số mô đun ngoại vi và dùng để truyền những tín hiệu ngắt từ ngoại vi. SFRs nằm ở 16 Byte thấp của vùng địa chỉ và được tổ chức bằng Byte. SFRs chỉ có thể được truy cập bởi chỉ thị Byte.

I.3.5. Tổ chức bộ nhớ

Byte thì dùng để định vị trí của địa chỉ lẻ hoặc chẵn, còn Word thì chỉ sử dụng cho địa chỉ chẵn. Vì vậy khi sử dụng những chỉ lệnh Từ thì chỉ có địa chỉ chẵn thì được sử dụng. Byte thấp của một Word luôn là số chẵn, Byte cao thì ở một số lẻ kế tiếp.

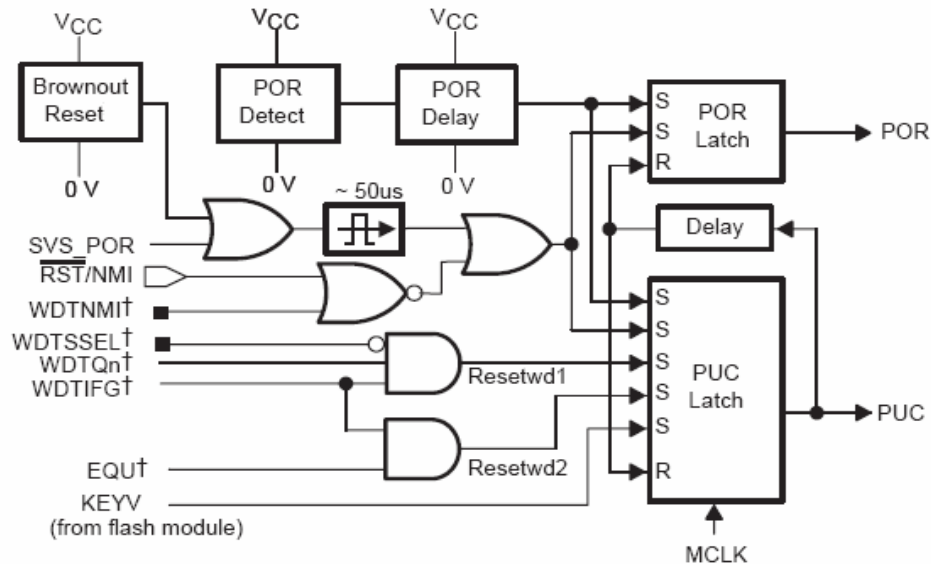


Hình I.3: Bit, Byte, Word trong cấu trúc nhớ của MSP430

II. HỆ THỐNG RESET, NGẮT VÀ CÁC CHẾ ĐỘ HOẠT ĐỘNG

II.1. Hệ thống reset và khởi động

Hệ thống mạch reset bao gồm 2 hệ thống là POR (Power On Reset) và PUC (Power Up Claer).



Hình II.1: Sơ đồ hệ thống Reset của MSP430

+ POR chỉ được sinh ra bởi 3 sự kiện sau:

- Cấp nguồn cho thiết bị.
- Tín hiệu ở chân $\overline{\text{RST}}/\text{NMI}$ ở mức thấp khi thiết lập cấu hình cho hệ thống Reset.
- SVS ở mức thấp khi $\text{PORON} = 1$.

+ PUC thì luôn sinh ra khi POR được sinh ra, nhưng không xảy ra ngược lại. Những sự kiện sau đây sẽ khởi động PUC:

- Khi POR được sinh ra.
- Watchdog timer hết hiệu lực khi trong chế độ watchdog.
- Khi chốt an toàn của Watchdog timer, Flash memory bị vi phạm.

II.1.1. Power On Reset (POR)

POR được sinh ra bởi các điều kiện liên quan đến phần cứng:

- + Thiết bị được cấp nguồn. POR được sinh ra nếu điện áp cung cấp giảm xuống dưới giá trị mà tại đó thiết bị không còn làm việc chính xác nữa, còn gọi là sự sụt áp.
- + Khi chân $\overline{\text{RST}}/\text{NMI}$ ở mức thấp nếu nó được thiết lập chức năng Reset hơn là ngắt không khả năng. Mặc định thì chức năng của chân này là Reset.
- + Ở những phương án lớn hơn có bộ giám sát điện áp cung cấp SVS. Ở đây ta có thể thiết lập cấu hình, không giống như bộ dò sụt áp. Nó đặt cờ SVSFG nếu điện áp sụt dưới mức được lập trình và có thể Reset một cách tùy chọn.

II.1.2. Power Up Clear (PUC)

PUC thì luôn sinh ra khi POR được sinh ra, nhưng không xảy ra ngược lại. Những sự kiện sau đây sẽ khởi động PUC:

- + Watchdog timer bị tràn trong chế độ giám sát.

- + Một sự cố gắn ghi vào thanh ghi điều khiển giám sát WDTCTL mà không đúng từ khóa 0x5A trong Byte cao. Một Reset được khởi động nếu watchdog không cho phép hoặc ở chế độ định thời khoảng.
- + PUC còn được sinh ra bởi một sự truy cập địa chỉ dự trữ của những thanh ghi ngoại vi hoặc đọc bộ nhớ không được thực thi.

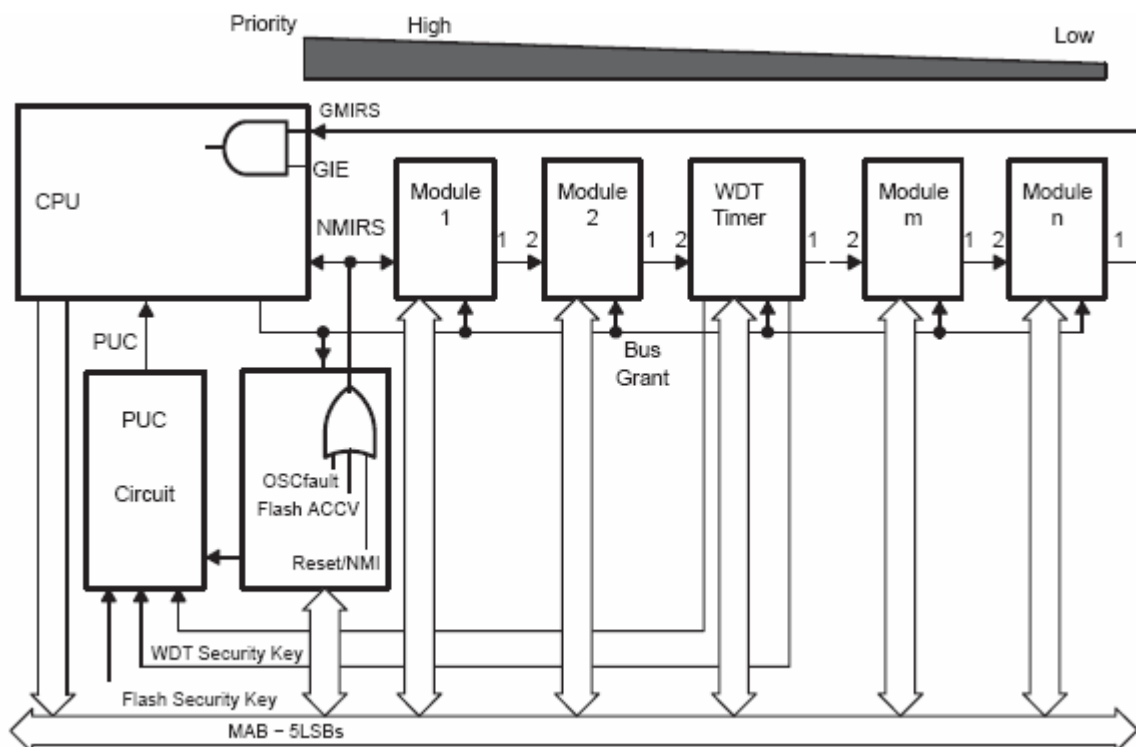
II.1.3. Các điều kiện sau khi reset

Những điều kiện ban đầu cho những thanh ghi và những ngoại vi sau khi Reset POR và PUC được giới thiệu tổng quát như sau:

- + Chân $\overline{\text{RST}}/\text{NMI}$ được thiết lập cho Reset, nó cũng có thể được sử dụng cho kết nối JTAG ở một vài loại như F2013.
- + Phần lớn các chân I/O được thiết lập như là ngõ vào tín hiệu số. Cũng có một vài ngoại lệ là những chân này có thể sử dụng với bộ dao động thạch anh như ở F2013.
- + Thanh ghi trạng thái được Clear. Điều này có nghĩa là IC sẽ hoạt động hết công suất, mặc dù nó có thể có một chế độ nguồn thấp trước khi được Reset trở lại.
- + Watchdog timer bắt đầu ở chế độ giám sát. Đây là một điều cốt yếu vì bởi vì nó là một đặc tính an toàn, nhưng có nghĩa là bạn phải vận hành nó hoặc cho phép nó trước khi nó hết thời gian chờ và Reset lại chip.
- + Bộ đếm chương trình được nạp bởi một vector Reset được lưu ở địa chỉ 0xFFFFE. Nó sẽ cung cấp địa chỉ của lệnh đầu tiên được thực thi.

II.2. Hệ thống ngắt

Hệ thống ngắt được bố trí thành một dãy kết nối gồm nhiều mô đun. Mô đun gần nhất là CPU/NMIRS thì có quyền ưu tiên ngắt cao hơn. Ở đây có 3 loại ngắt là: Reset hệ thống, ngắt khả nguy, ngắt không khả nguy.



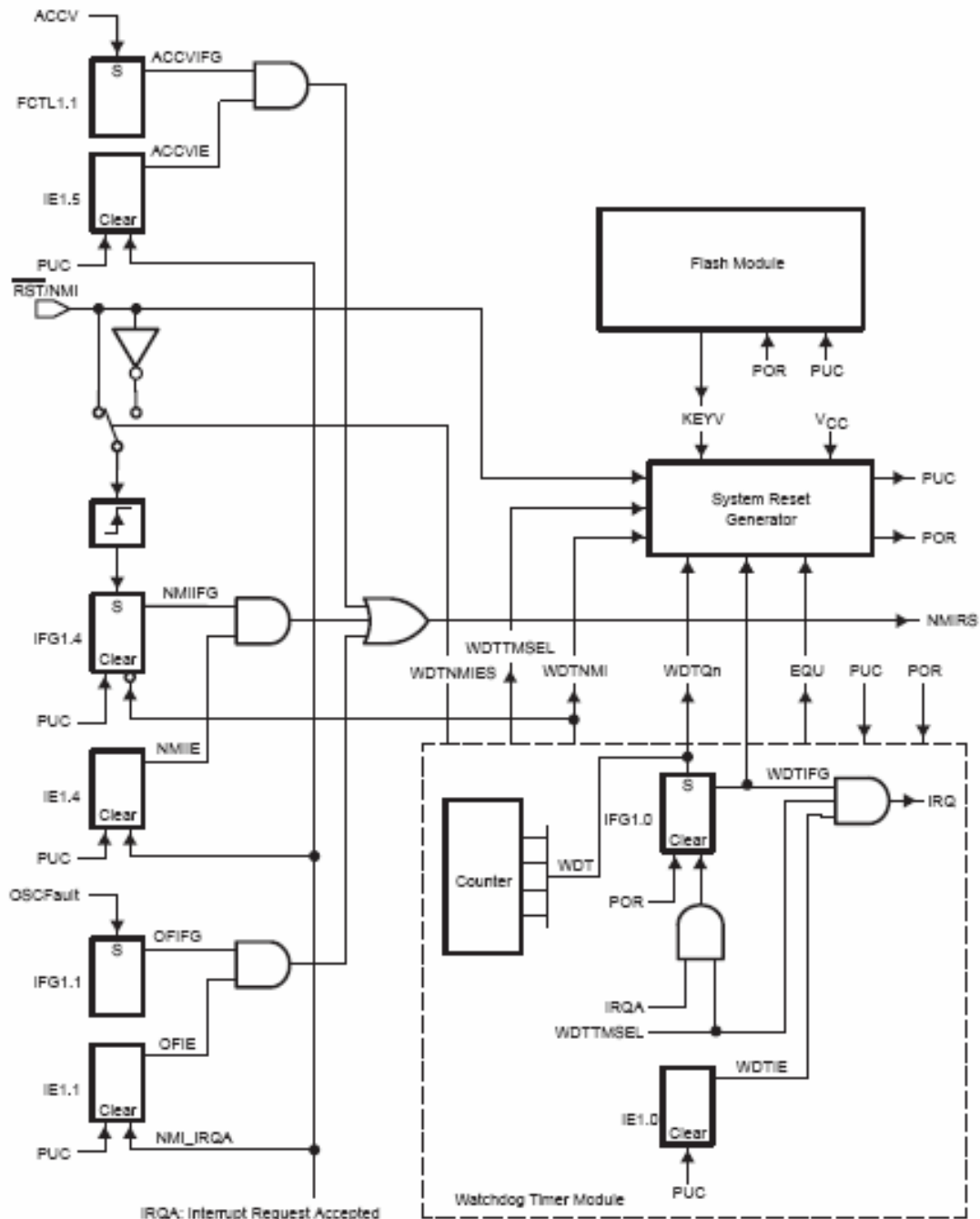
Hình II.2: Sơ đồ hệ thống ngắt

II.2.1. Ngắt không khả nguy

Ngắt không khả nguy NMI thì không bị che bởi bit ngắt cho phép chung GIE nhưng bị che bởi các bit ngắt riêng lẻ như NMIIIE, ACCVIE, OFIE. Khi một ngắt NMI được chấp nhận thì tất cả những bit ngắt cho phép NMI tự động Reset. Chương trình bắt đầu chạy ở địa chỉ được lưu trữ trong vector ngắt không khả nguy là 0FFFCh.

Ngắt không khả nguy có thể được sinh ra bởi 3 nguồn sau:

- + Biên của chân $\overline{\text{RST/NMI}}$ khi được thiết lập ở chế độ NMI.
- + Xuất hiện những sự đứt gãy dao động.
- + Có sự vượt quá dữ liệu bộ nhớ Flash.



Hình II.3: Sơ đồ khối của nguồn ngắt khả nguy

➤ Chân $\overline{\text{RST}}/\text{NMI}$

Khi cấp nguồn, chân $\overline{\text{RST}}/\text{NMI}$ được thiết lập ở chế độ Reset. Chức năng của chân $\overline{\text{RST}}/\text{NMI}$ được chọn trong thanh ghi bộ điều khiển giám WDTCTL. Nếu chân $\overline{\text{RST}}/\text{NMI}$ được thiết lập để khởi động lại chức năng, CPU giữ ở chế độ Reset trong lúc chân $\overline{\text{RST}}/\text{NMI}$ ở mức thấp. Sau khi chân $\overline{\text{RST}}/\text{NMI}$ lên mức cao thì CPU bắt đầu chạy chương trình từ địa chỉ lưu trong vector Reset là 0FFFFh.

Nếu chân $\overline{\text{RST}}/\text{NMI}$ được thiết lập tính năng NMI bởi phần mềm người sử dụng, một tín hiệu biên được chọn bởi bit WDTNMIES sinh ra ngắt NMI nếu bit NMIIIE được Set. Cờ NMIIFG cũng được Set.

➤ Sự vi phạm dữ liệu bộ nhớ Flash

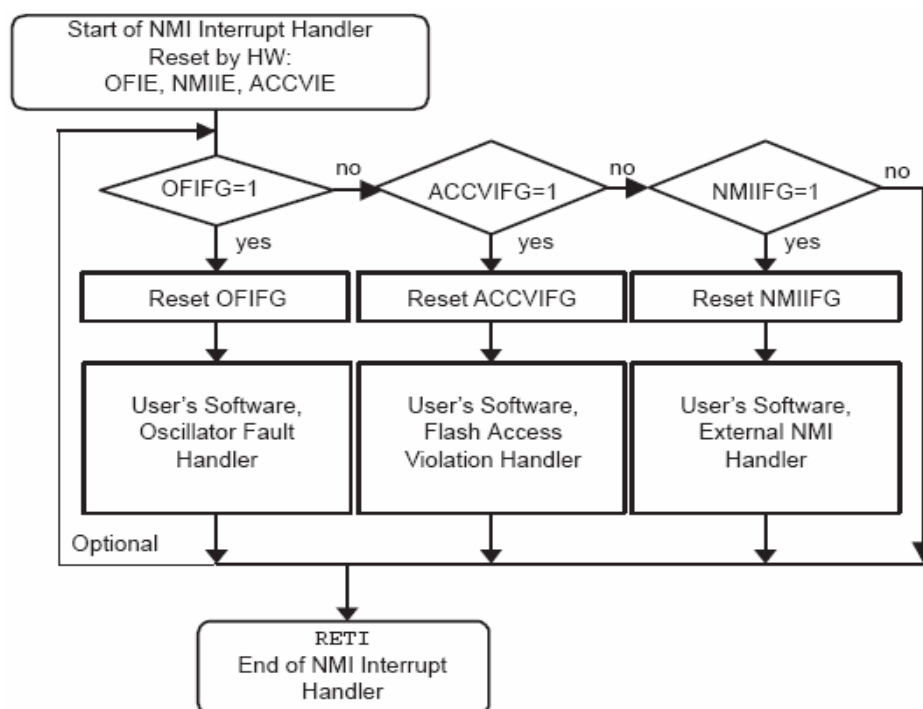
Cờ nhớ ACCVIFG được Set khi xuất hiện những sự vi phạm dữ liệu bộ nhớ Flash. Nó có thể cho phép phát sinh ngắt NMI khi bit ACCVIE được thiết lập. Cờ ACCVIFG có thể được kiểm tra bởi chương trình ngắt NMI nếu NMI được gây ra bởi sự vi phạm bộ nhớ Flash.

➤ Sự đứt gãy dao động

Sự đứt gãy tín hiệu dao động báo cho ta biết có thể bộ dao động thạch anh bị lỗi. Sự đứt gãy dao động này cũng có thể phát sinh ra ngắt NMI bởi việc thiết lập bit OFIE. Tín hiệu PUC có thể kích hoạt sự đứt gãy dao động bởi vì bộ ngắt mạch PUC là LFXT1 ở chế độ LF thành ra tắt đi chế độ HF. PUC cũng tắt đi bộ dao động XT2.

➤ Ví dụ về một bộ điều khiển ngắt NMI

Ngắt NMI là một ngắt nhiều nguồn. Ngắt NMI tự động Reset các Bit ngắt cho phép như NMIIIE, OFIE và ACCVIE. Sử dụng chương trình ngắt NMI để reset các cờ ngắt và cho phép lại các Bit ngắt sao cho phù hợp với những ứng dụng.



Hình II.4: Điều khiển ngắt NMI

II.2.2. Ngắt khả nguy

Ngắt khả nguy được gây ra bởi các thiết bị ngoại vi với khả năng ngắt bao gồm cả bộ định thời giám sát sự tràn của bộ đếm thời gian. Mỗi nguồn ngắt khả nguy có thể mất tác dụng một cách riêng lẻ bởi một bit ngắt cho phép hoặc tắt cả những ngắt khả nguy có thể bị mất tác dụng bởi bit cho phép ngắt GIE ở thanh ghi trạng thái.

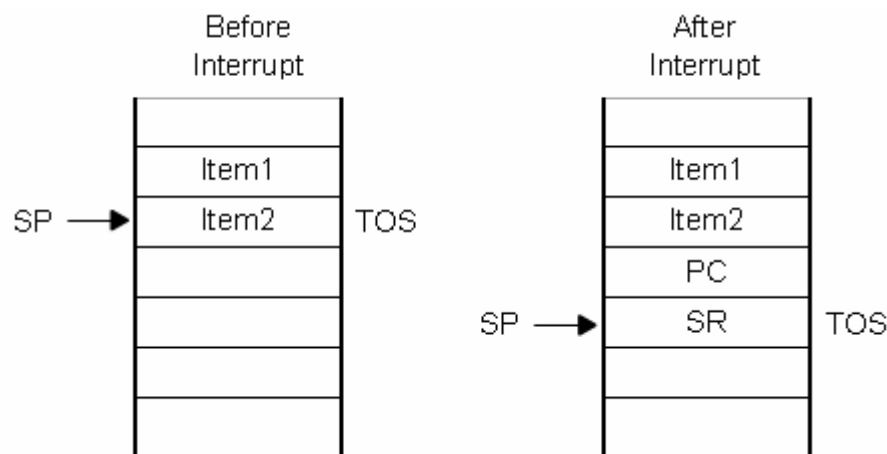
II.2.3. Xử lý ngắt

Khi một ngắt được yêu cầu từ thiết bị ngoại vi, các Bit ngắt được cho phép và Bit GIE được Set, chương trình ngắt được yêu cầu. Chỉ có những Bit cho phép riêng rẽ phải được Set cho ngắt không khả nguy được yêu cầu.

➤ Sự chấp nhận ngắt:

Thời gian trễ ngắt là 5 chu kỳ (CPUx) và 6 chu kỳ (CPU), bắt đầu với sự chấp nhận của một yêu cầu ngắt và kéo dài đến khi thực hiện những lệnh đầu tiên của các chương trình ngắt. Ngắt Logic được thực hiện theo các bước như sau:

1. Bất kì một lệnh nào đang thực thi đều được hoàn thành.
2. Bộ đếm chương trình PC, địa chỉ của lệnh kế tiếp được đẩy lên ngăn xếp.
3. SR được đẩy lên ngăn xếp.
4. Các ngắt với ưu tiên cao hơn thì được chọn nếu các ngắt xảy ra trong suốt quá trình thực thi và làm trì hoãn chương trình.
5. Những cờ ngắt yêu cầu Reset một cách tự động trên cờ nguồn đơn. Nhiều cờ nguồn vẫn Set cho chương trình bằng phần mềm.
6. SR được Clear. Điều này sẽ chấm dứt bất cứ chế độ nguồn thấp nào vì Bit GIE được Clear, hơn nữa các ngắt thì không có tác dụng.
7. Nội dung của véc tơ ngắt được nạp vào trong PC: Chương trình ngắt tiếp tục tại địa chỉ đó.



Hình II.5: Xử lý ngắt

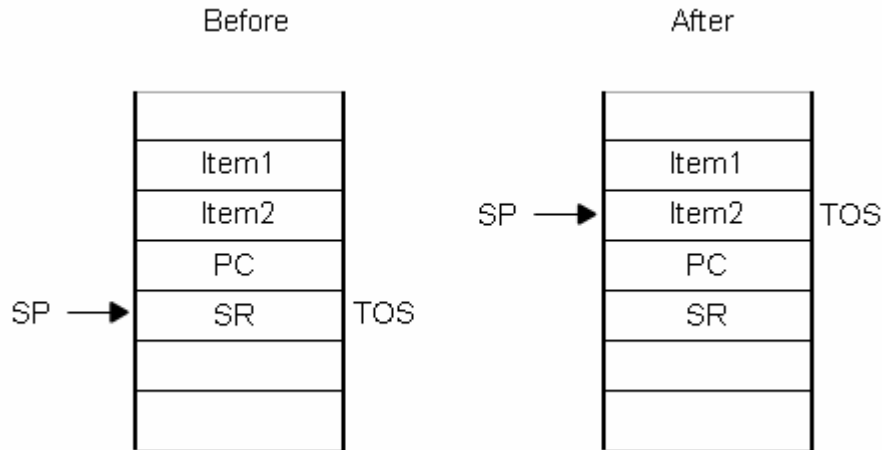
➤ Quay về từ chương trình ngắt:

Chương trình ngắt được chấm dứt với lệnh RETI: Quay về từ một chương trình ngắt.

Sự trở lại từ chương trình ngắt mất 5 chu kỳ (CPU) và 3 chu kỳ (CPUx) để thực thi các hoạt động sau:

1. SR với tất cả các thiết lập trước đó được lấy ra từ ngăn xếp. Tất cả các thiết lập trước đó của GIE, CPUOFF, ... có hiệu lực, bất chấp các cài đặt trong suốt chương trình ngắt.

2. PC được lấy ra từ ngăn xếp và bắt đầu thực thi tại nơi mà nó bị gián đoạn khi ngắt.



Hình II.6: Quay về từ chương trình ngắt

II.2.4. Ngắt có hướng (Ngắt véc tơ)

Ngắt véc tơ khởi đầu ở địa chỉ nằm trong dãy địa chỉ từ 0FFFFh đến 0FFC0h. Một véc tơ được lập trình bởi người sử dụng với 16 bit địa chỉ của ngắt tương ứng.

Nó thì đề nghị cung cấp một thường trình ngắt cho mỗi véc tơ ngắt được gán cho một Mô đun. Một thường trình ngắt có thể bao gồm một lệnh RETI và một vài véc tơ ngắt có thể trở đến nó.

Những véc tơ ngắt không được chỉ định có thể được sử dụng cho mã chương nếu cần thiết.

Một số mô đun Bit cho phép, Bit ngắt cho phép, cờ ngắt thì nằm ở SFRs. SFRs nằm ở phạm vi địa chỉ thấp hơn và được thực hiện ở định dạng Byte. SFRs phải được truy cập bằng những lệnh Byte.

Địa chỉ của các nguồn ngắt và cờ ngắt được trình bày dưới Bảng II.1

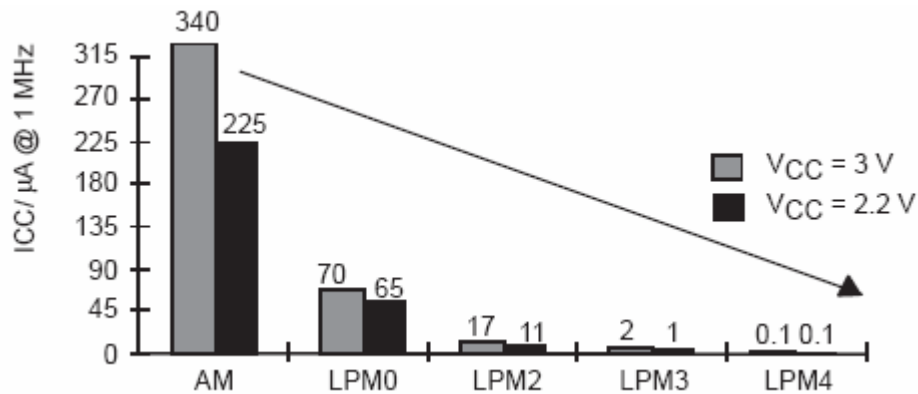
INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-up, external reset, watchdog, flash password, illegal instruction fetch.	PORIFG RSTIFG WDTIFG KEYV	RESET	0FFFEh	31
NMI, oscillator fault, flash memory access violation.	NMIIFG ACCVIFG OFIFG	(non)-maskable (non)-maskable (non)-maskable	0FFFCh	30
device-specific			0FFFAh	29
device-specific			0FFF8h	28
device-specific			0FFF6h	27
Watchdog timer	WDTIFG	maskable	0FFF4h	26
device-specific			0FFF2h	25
device-specific			0FFF0h	24
device-specific			0FFEEh	23
device-specific			0FFECCh	22
device-specific			0FFEAh	21
device-specific			0FFE8h	20
device-specific			0FFE6h	19
device-specific			0FFE4h	18
device-specific			0FFE2h	17
device-specific			0FFE0h	16
device-specific			0FFDEh	15
device-specific			0FFDCCh	14
device-specific			0FFDAh	13
device-specific			0FFD8h	12
device-specific			0FFD6h	11
device-specific			0FFD4h	10
device-specific			0FFD2h	9
device-specific			0FFD0h	8
device-specific			0FFCEh	7
device-specific			0FFCCh	6
device-specific			0FFCAh	5
device-specific			0FFC8h	4
device-specific			0FFC6h	3
device-specific			0FFC4h	2
device-specific			0FFC2h	1
device-specific			0FFC0h	0

Bảng II.1: Những nguồn ngắt, cờ ngắt và véc tơ ngắt

II.3. Các chế độ hoạt động

Họ MSP430 được thiết kế cho những ứng dụng sử dụng nguồn thấp và sử dụng ở nhiều chế độ hoạt động khác nhau. Các chế độ hoạt động khác nhau ở 3 đặc điểm chính:

- + Mức độ sử dụng nguồn thấp.
- + Tốc độ và lưu lượng dữ liệu.
- + Mức độ làm nhỏ đi lượng tiêu thụ điện của các thiết bị ngoại vi.



Hình II.7: Dòng tiêu thụ của MSP430x1xx ở các chế độ hoạt động

Ở chế độ nguồn thấp LPM0 đến LPM4 được thiết lập với các Bit SCG0, SCG1, OSCOFF, CPUOFF trong thanh ghi trạng thái. Ưu điểm của các Bit điều khiển SCG0, SCG1, OSCOFF và CPUOFF trong thanh ghi trạng thái là chế độ hoạt động hiện tại sẽ được lưu trong Stack trong suốt quá trình ngắt diễn ra. Chương trình sẽ quay về chế độ hoạt động trước đó nếu giá trị được lưu trong thanh ghi trạng thái không bị thay đổi trong suốt quá trình ngắt. Chương trình có thể quay về một chế độ hoạt động khác nếu có sự thay đổi giá trị được lưu bên trong Stack trong suốt quá trình ngắt diễn ra. Các Bit điều khiển và Stack có thể được truy cập bởi bất kỳ lệnh nào. Các chế độ hoạt động của họ MSP430 được trình bày ở Bảng II.2

Mode	SCG1	SCG0	OSCOFF	CPUOFF	CPU and Clocks Status
Active	0	0	0	0	CPU on, MCLK on, SMCLK on, ACLK on.
LPM0	0	0	0	1	CPU off, MCLK off, SMCLK on, ACLK on.
LPM1	0	1	0	1	CPU off, MCLK off, SMCLK on, ACLK on.
LPM2	1	0	0	1	CPU off, MCLK off, SMCLK off, DCO off, ACLK on.
LPM3	1	1	0	1	CPU off, MCLK off, SMCLK on, DCO off, ACLK on.
LPM4	1	1	1	1	CPU off, MCLK off, SMCLK off, DCO off, ACLK off.

Bảng II.2: Các chế độ hoạt động cơ bản của MSP430

II.3.1. Cách nhập và thoát khỏi Low-Power Modes

Một sự kiện ngắt được cho phép sẽ kích hoạt MSP430 từ bất kì chế độ hoạt động nguồn thấp nào. Chương trình là:

Nhập chương trình ngắt:

- + PC và SR được lưu trữ trên ngăn xếp.
- + Các Bit CPUOFF, SCG1 và OSCOFF tự động Reset.

Trở về từ chương trình ngắt:

- + SR được lấy ra từ ngăn xếp và khôi phục lại chế độ điều hành trước đó.
- + Các Bit SR được lưu trữ trong ngăn xếp có thể bị thay đổi trong thường trình dịch vụ ngắt quay trở về một chế độ hoạt động khác khi lệnh RETI được thực thi.

➤ **Ví dụ:**

; Chế độ LPM0

BIS #GIE+CPUOFF,SR ; Khởi tạo LPM0

... ; Dừng chương trình

;

; Thoát khỏi chương trình con ngắt LPM0

BIC #CPUOFF,0(SP) ; Trở về chương trình chính bằng lệnh RETI

RETI

; Chế độ LPM3

BIS #GIE+CPUOFF+SCG1+SCG0,SR ; Khởi tạo LPM3

... ; Dừng chương trình

;

; Thoát khỏi chương trình con ngắt LPM3

BIC #CPUOFF+SCG1+SCG0,0(SR) ; Trở về chương trình chính bằng lệnh RETI

RETI

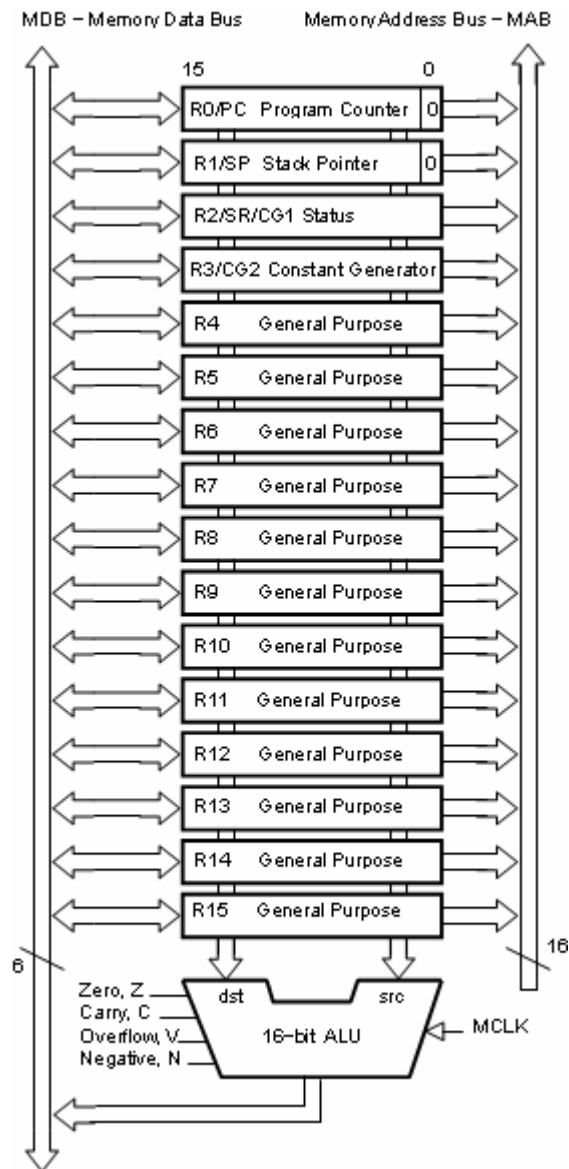
III. RISC 16 BIT CPU

III.1. Giới thiệu bộ xử lý trung tâm CPU

CPU của họ vi điều khiển MSP430 mang những đặc điểm thiết kế đặc biệt như kĩ thuật tính toán rẽ nhánh, sử dụng ngôn ngữ lập trình cấp cao như C, C++. CPU còn có thể định địa chỉ của một vùng địa chỉ bằng cách đánh dấu vùng nhớ.

CPU mang những đặc điểm sau:

- + Cấu trúc RISC với 27 lệnh chỉ thị và 7 chế độ định địa chỉ.
- + Đầy đủ thanh ghi dữ liệu bao gồm thanh ghi bộ đếm chương trình, thanh ghi trạng thái, thanh ghi con trỏ ngăn xếp.



Hình III.1: Sơ đồ các khối chức năng của CPU

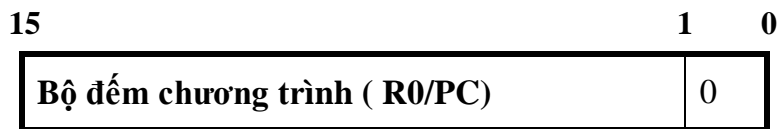
- + Hơn 16 bit thanh ghi tập tin giúp giảm bớt sự truy cập bộ nhớ.
- + 16 bit Bus địa chỉ cấp phát dữ liệu trực tiếp.
- + 16 bit Bus dữ liệu cấp phát trực tiếp những thao tác trên khối thông tin.
- + Bộ sinh hằng (Constant Generator) cung cấp 6 giá trị tức thời thường sử dụng nhất và làm giảm được kích thước của Code lập trình.

III.2. Các thanh ghi CPU

CPU có 16 thanh ghi 16 bit, 4 thanh ghi đầu là R0, R1, R2 và R3 là các thanh ghi có chức năng đặc biệt, trong khi 12 thanh ghi còn lại (Từ R4 đến R15) dùng với mục đích sử dụng chung. Nói chung 16 thanh ghi này là đặc trưng cho một bộ xử lý dùng tập lệnh rút gọn RISC CPU. Sau đây là tóm tắt về các tính năng của các thanh ghi trong CPU:

III.2.1. Thanh ghi bộ đếm chương trình (R0/PC)

Thanh ghi này chứa các lệnh kế tiếp được thực thi. Mỗi lệnh này sử dụng một số chữ của những Byte (hai, bốn hoặc sáu) và PC được gia tăng một cách phù hợp. Các lệnh này chứa dung lượng trong khoảng 64 KB và PC sắp xếp chúng theo địa chỉ.



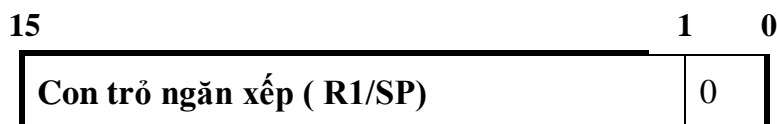
Hình III.2: Bộ đếm chương trình

➤ **Ví dụ:**

MOV #LABEL,PC; Đưa nội dung của LABEL vào PC
 MOV LABEL,PC ; Đưa địa chỉ LABEL vào PC
 MOV @R14, PC ; Đưa gián tiếp địa chỉ của nội dung trong R14 vào PC

III.2.2. Con trỏ ngăn xếp (SP)

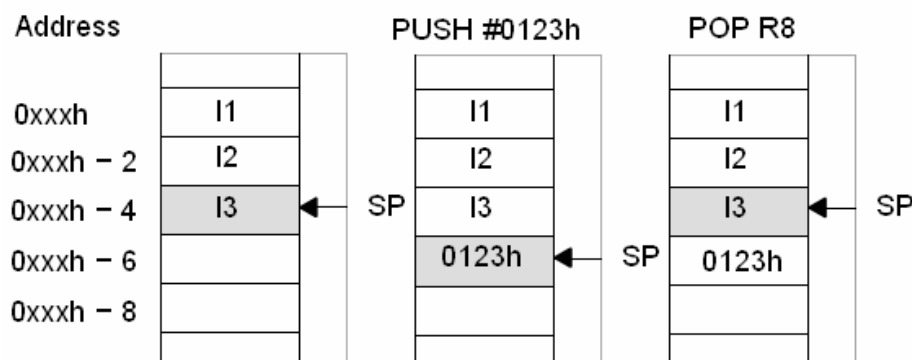
Khi một chương trình con được gọi, CPU nhảy đến chương trình con đó và thực thi nó, sau đó quay trở về chương trình chính. Do đó phải có một bộ nhớ tạm để lưu giữ địa chỉ của chương trình con đó trước khi CPU nhảy đến. Đó là mục đích cơ sở của con trỏ ngăn xếp. Có nhiều bộ xử lý sử dụng một bộ nhớ dành riêng cho con trỏ ngăn xếp nhưng với họ MSP430 thì sử dụng những địa chỉ cao của RAM. Khi nhập địa chỉ vào hay lấy ra thì con trỏ ngăn xếp tự động điều chỉnh tăng lên hay giảm xuống.



Hình III.3: Con trỏ ngăn xếp

➤ **Ví dụ:**

MOV 2(SP),R6 ; SP +2→ R6
 MOV R7,0(SP) ; R7 → SP
 PUSH #0123h ; Đưa giá trị 0123h vào ngăn xếp
 POP R8 ; R8 = 0123h



Hình III.4: Sử dụng ngăn xếp



Hình III.5: Thao tác đưa dữ liệu vào và lấy ra từ ngăn xếp

III.2.3. Thanh ghi trạng thái (SR)

Thanh ghi trạng thái (SR/R2) được sử dụng như là một thanh ghi nguồn hoặc thanh ghi đích gồm hệ thống các cờ (là những Bit đơn). Một số cờ thường sử dụng là C, Z, N và V. Chúng được sử dụng trong các thuật toán hoặc các thao tác logic.

15		8				0			
Dự trữ	V	SCG1	SCG0	OSC OFF	CPU OFF	GIE	N	Z	C

Hình III.6: Mô tả về các Bit của thanh ghi trạng thái

- + **Cờ V**: Cờ tràn. Cờ này được thiết lập mỗi khi kết quả của một phép tính số có dấu quá lớn tạo ra Bit bậc cao làm tràn Bit dấu. Cờ tràn chỉ được dùng để phát hiện lỗi trong các phép toán số học có dấu.
- + **SCG1**: Bộ định thời hệ thống 1. Khi bit này được thiết lập sẽ làm tắt SMCLK.
- + **SCG0**: Bộ định thời hệ thống 0. Khi bit này được thiết lập sẽ làm tắt DCO nếu DCOCLK không được sử dụng cho MCLK và SMCLK.
- + **OSCOFF**: Tắt bộ dao động. Khi bit này được thiết lập sẽ làm tắt bộ dao động thạch anh LFXT1 khi LFXT1CLK không sử dụng cho MCLK hoặc SMCLK.
- + **CPUOFF**: Khi bit này được thiết lập sẽ làm tắt CPU.
- + **GIE**: Ngắt cho phép. Khi bit này được set thì cho phép ngắt khả năng. Khi ta reset thì ngắt khả năng này mất tác dụng.
- + **Cờ Z**: Zero bit. Cờ Z được thiết lập khi kết quả của một Byte hoặc Từ là 0 và được Clear khi kết quả khác 0.
- + **Cờ C**: Cờ carry. Cờ này thông thường dùng cho các lệnh toán học, được thiết lập khi có một số nhớ sinh ra bởi phép cộng hoặc số mượn bởi phép trừ và được Clear nếu không có số nhớ.
- + **Cờ N**: Phủ định bit. Cờ này được thiết lập khi kết quả của một Byte hoặc Word bị phủ định và được Clear khi kết quả ngược lại.

III.2.4. Thanh ghi bộ sinh hằng CG1 và CG2

Có sáu giá trị hằng số thường được sử dụng được sinh ra bởi thanh ghi bộ sinh hằng R2 và R3 mà không cần bổ sung thêm một từ 16 bit của mã chương trình. Những hằng số được chọn với chế độ định địa chỉ thanh ghi nguồn (AS).

Thanh ghi	As	Hằng số	Ghi chú
R2	00	-----	Chế độ thanh ghi
R2	01	(0)	Chế độ định địa chỉ
R2	10	00004h	+4, xử lí Bit
R2	11	00008h	+8, xử lí Bit
R3	00	00000h	0, xử lí Word
R3	01	00001h	+1
R3	10	00002h	+2, xử lí Bit
R3	11	0FFFFh	-1, xử lí Word

Bảng III.1: Các giá trị tức thời của bộ sinh hằng

Những ưu điểm:

- + Không cần những lệnh đặc biệt.
- + Không cần bổ sung từ mã cho 6 hằng số.
- + Mã truy cập bộ nhớ không đòi hỏi phải khôi phục lại hằng số.

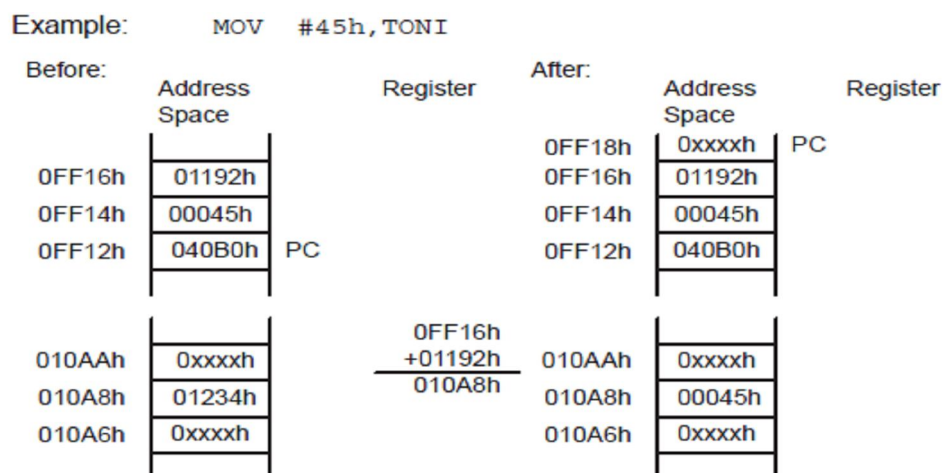
III.2.5. Thanh ghi sử dụng chung R4-R15

12 thanh ghi từ R4 đến R15 không có mục đích chuyên dụng, các thanh ghi này có thể sử dụng làm thanh ghi dữ liệu hoặc địa chỉ đều được vì đều là các giá trị 16 Bit, từ đó giúp đơn giản hóa các hoạt động. Một số qui ước nên được làm theo nếu chương trình được viết bằng assemble. Những thanh ghi đặc biệt này nên được sử dụng để di chuyển các tham số và trả về các kết quả. Tất cả các vấn đề này đều được giải quyết nếu chương trình được viết bằng C.

III.3 Các chế độ định địa chỉ

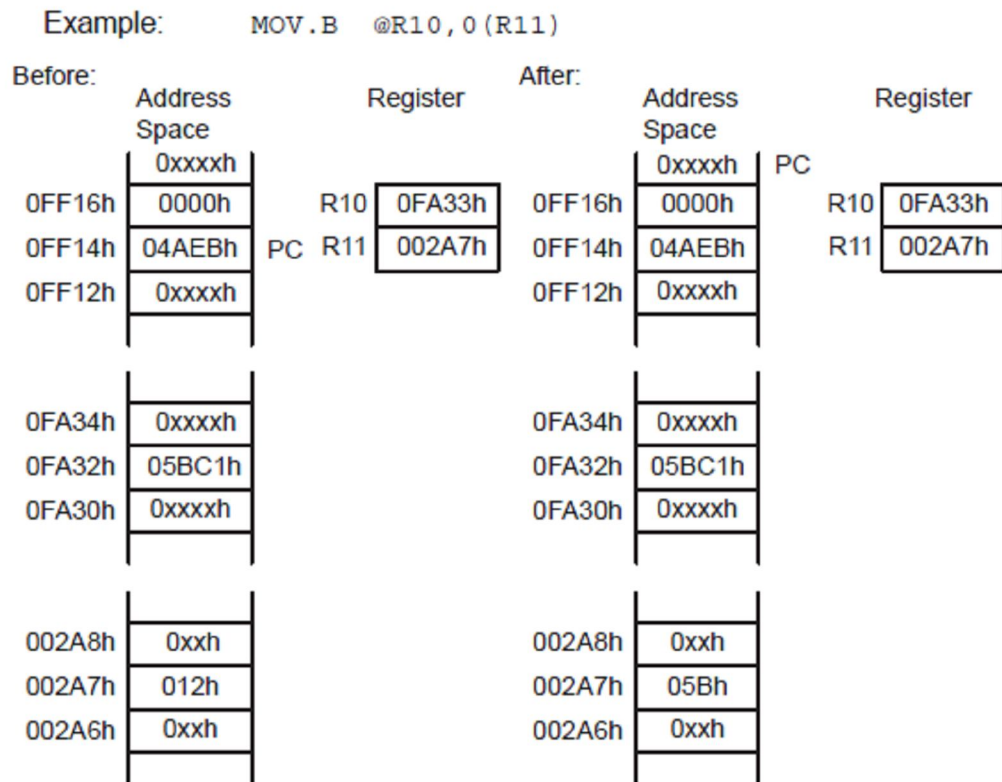
III.3.1 Định địa chỉ trực tiếp (immediate mode)

MOV #30H, R0 ; đưa giá trị 30h vào thanh ghi R0



III.3.2. Định địa chỉ gián tiếp thanh ghi (indirect register mode)

MOV @R10, 0(R0) ; Đưa địa chỉ giá trị nội dung của thanh ghi R10 vào địa chỉ có chứa nội dung thanh ghi R0. Nhưng giá trị thanh ghi không thay đổi.

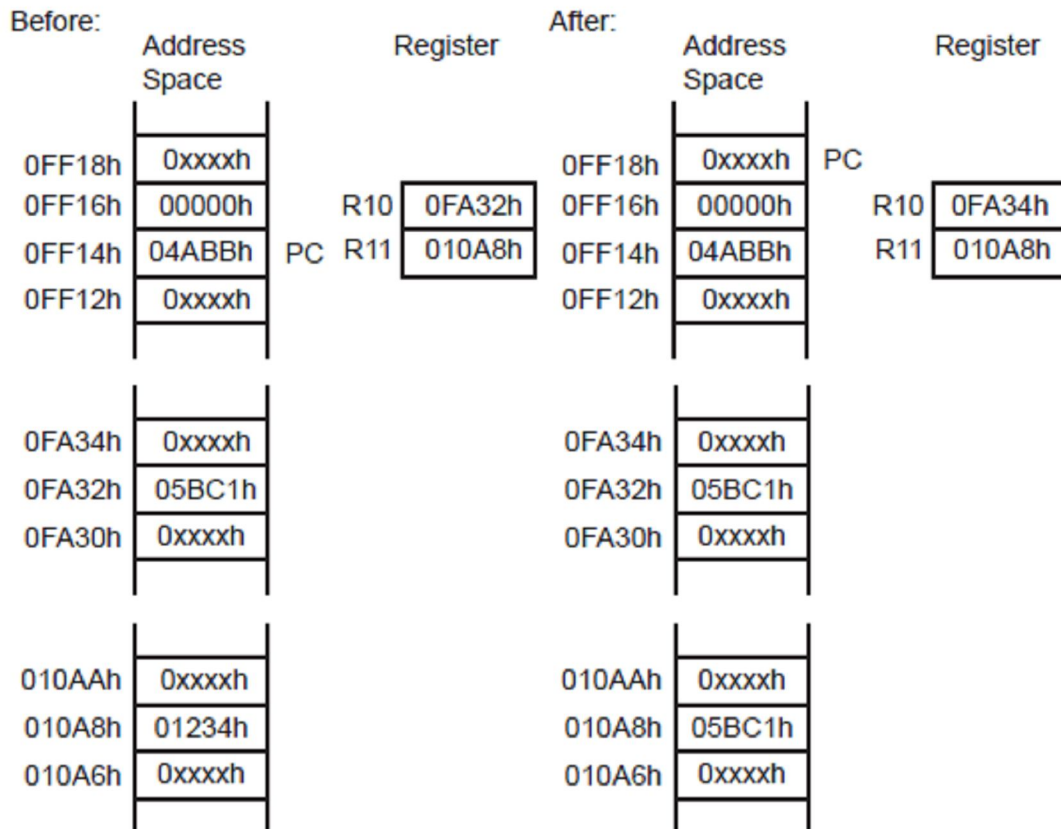


III.3.3. Định địa chỉ gián tiếp tự tăng (indirect autoincrement mode)

MOV @R10+, 0(R0); Lấy nội dung của thanh ghi R10 vào thanh ghi R0 và đồng thời tăng địa chỉ thanh ghi R10 lên 2.

Ví dụ : Lúc đầu Thanh ghi R10 có chứa địa chỉ của ô nhớ (123h) có chứa giá trị là 10h, thanh ghi R0 có chứa địa chỉ là 0AFH thì sao khi thực hiện lệnh đó ta có kết quả như sau: Địa chỉ con trỏ của R10 tăng lên 125h, còn địa chỉ con trỏ của R0 không đổi là 0AFH, Nhưng nó có chứa giá trị là 10h.

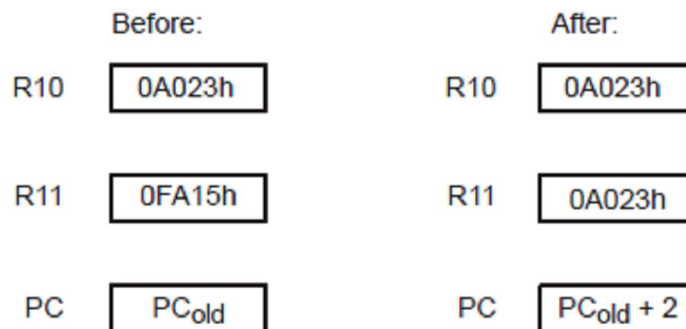
Example: `MOV @R10+, 0 (R11)`



III.3.4. Định địa chỉ trực tiếp thanh ghi(immediate mode)

`MOV R0,R1` ; đưa giá trị thanh ghi R0 vào thanh ghi R1.

Example: `MOV R10,R11`



III.3.5. Định địa chỉ tuyệt đối (absoluto mode)

`MOV &EDE,&TONI` ; đưa giá trị của địa chỉ có chứa nhãn EDE vào địa chỉ có chứa nhãn TONI.

Vd: EDE có địa chỉ là 0FF0h chứa giá trị là 1234h, TONI có địa chỉ là 1FFh có giá trị bất kỳ. Sau khi thực hiện lệnh thì TONI có giá trị là 1234h.

III.3.6. Định địa chỉ giữa các biến (symbolic mode)

`MOV EDE,TONI` ; đưa giá trị của biến có địa chỉ của biến EDE vào biến TONI. Ví dụ: EDE có giá trị 10h, TONI có giá trị bất kỳ. Sau khi thực hiện lệnh TONI có giá trị là 10h.

Before:	Address Space	Register	After:	Address Space	Register
				0xxxxh	PC
0FF16h	011FEh		0FF16h	011FEh	
0FF14h	0F102h		0FF14h	0F102h	
0FF12h	04090h	PC	0FF12h	04090h	
0F018h	0xxxxh	0FF14h +0F102h ----- 0F016h	0F018h	0xxxxh	
0F016h	0A123h		0F016h	0A123h	
0F014h	0xxxxh		0F014h	0xxxxh	
01116h	0xxxxh	0FF16h +011FEh ----- 01114h	01116h	0xxxxh	
01114h	05555h		01114h	0A123h	
01112h	0xxxxh		01112h	0xxxxh	

MOV 2(R5),3(R6) ; đưa giá trị tại địa chỉ của R5 +2 vào địa chỉ R6+3.

Before:	Address Space	Register	After:	Address Space	Register
	0FF16h: 00006h	R5: 01080h		0xxxxh	PC: 01080h
	0FF14h: 00002h	R6: 0108Ch		0FF14h: 00002h	R6: 0108Ch
	0FF12h: 04596h			0FF12h: 04596h	
	01094h: 0xxxxh	0108Ch + 0006h		01094h: 0xxxxh	
	01092h: 05555h	01092h		01092h: 01234h	
	01090h: 0xxxxh			01090h: 0xxxxh	
	01084h: 0xxxxh	01080h + 0002h		01084h: 0xxxxh	
	01082h: 01234h	01082h		01082h: 01234h	
	01080h: 0xxxxh			01080h: 0xxxxh	

IV. CÁC BỘ ĐỊNH THỜI CƠ BẢN

IV.1. Giới thiệu

MSP430 sử dụng bộ định thời mà hệ thống có giá rẻ và sử dụng nguồn cực thấp. Có 3 xung Clock bên trong nên người sử dụng có thể lựa chọn sao cho cân bằng giữa hiệu suất làm việc với nguồn điện thế tiêu thụ. Một mô đun bộ định thời có 2 hoặc 3 nguồn xung Clock:

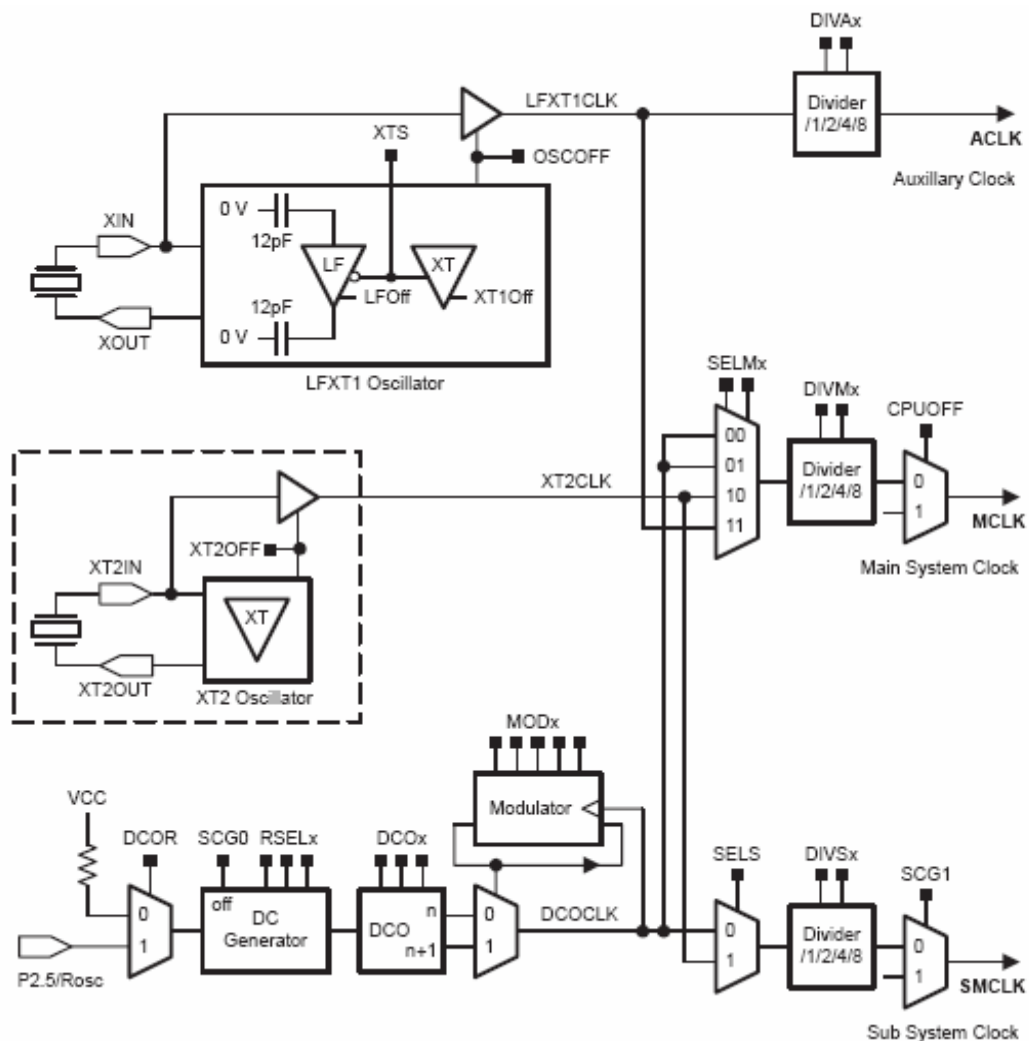
+ **LFXT1CLK:** Bộ dao động tần số thấp/ tần số cao, nó có thể được sử dụng với tần số thạch anh 32768 Hz hoặc tần số thạch anh chuẩn, hoặc bộ cộng hưởng từ 450 KHz đến 8 MHz.

+ **XT2CLK:** Bộ dao động tần số cao. Bộ dao động này có thể được sử dụng với thạch anh chuẩn, bộ cộng hưởng, hoặc nguồn xung Clock bên ngoài có tần số từ 450 KHz đến 8 MHz.

+ **DCOCLK:** Bộ dao động được điều khiển bằng kỹ thuật số (DCO).

Ba tín hiệu xung Clock ở trên có được là từ những mô đun Clock cơ bản như:

+ **ACLK:** Nguồn xung Clock hỗ trợ. ACLK được lưu trong bộ đệm LFXT1CLK và được chia 1, 2, 4 hoặc 8. ACLK ở chương trình có thể được chọn cho nhưng mô đun ngoại vi riêng rẽ.



Hình IV.1: Sơ đồ khối chức năng của bộ định thời MSP430

+ **MCLK**: Bộ định thời chủ. MCLK có thể được chọn như là LFXT1CLK, XT2CLK (nếu sẵn có), hoặc DCOCLK. MCLK được chia 1, 2, 4 hoặc 8. MCLK thì được sử dụng bởi CPU và hệ thống.

+ **SMCLK**: Bộ định thời con. SMCLK có thể được chọn như là LFXT1CLK, XT2CLK (nếu sẵn có), hoặc DCOCLK. SMCLK được chia 1, 2, 4 hoặc 8 và được chọn cho những mô đun ngoại vi riêng rẽ.

IV.2. Chế độ hoạt động của các Clock Module +

PUC, MCLK, SMCLK được bắt nguồn từ Bộ tạo xung tốc độ cao DCO, tần số khoảng 1.1 MHz. ACLK thì được bắt nguồn từ LFXT1CLK ở chế độ TF với một dung kháng 6 pF bên trong.

Những Bit điều khiển SCG0, SCG1, OSCOFF và CPUOFF của thanh ghi trạng thái thiết lập cấu hình các chế độ hoạt động của MSP430, cho phép hoặc không cho phép sự phân đoạn của Clock Module+.

Clock Module+ có thể được thiết lập hoặc tái thiết lập bởi một chương trình trong bất kỳ một khoảng thời gian nào trong suốt quá trình thực thi chương trình.

Ví dụ như:

BIS.B #RSEL2+RSEL1+RSEL0,&BCSCTL1 ; Select range 7

BIS.B #DCO2+DCO1+DCO0,&DCOCTL; Select max DCO tap

IV.2.1. Bộ dao động tần số thấp

Bộ dao động tần số thấp (VLO) cung cấp một tần số 12 KHz mà không cần một thạch anh. Nguồn VLOCLK được chọn bởi việc thiết lập LFXT1Sx = 10 khi XTS = 0. Bit OSCOFF tắt VLO cho LPM4. Bộ dao động thạch anh LFXT1 bị tắt khi VLO được chọn để giảm lượng tiêu thụ của mạch điện. VLO không cần cấp nguồn khi không được sử dụng.

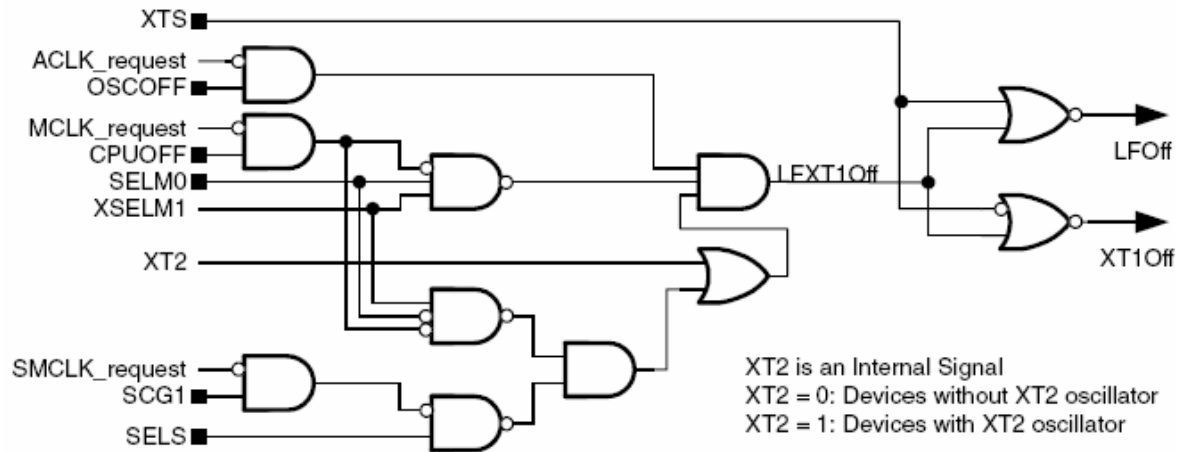
IV.2.2. Bộ dao động LFXT1

Bộ dao động LFXT1 tiêu thụ năng lượng cực thấp, sử dụng dao động thạch anh 32768 Hz trong chế độ LF (XTS = 0). Thạch anh được kết nối vào chân XIN và XOUT. Chương trình lựa chọn Bit XCAPx để thiết lập việc cung cấp một điện dung cho LFXT1. Điện dung này có thể là 1pF, 6pF, 10pF hoặc 12.5pF tùy loại. Ta có thể gắn thêm những tụ điện bên ngoài nếu cần.

LFXT1 cũng cần một thạch anh tốc độ cao hoặc một bộ cộng hưởng khi ở chế độ HF (XTS = 1, XCAPx = 00). Thạch anh hoặc bộ cộng hưởng này kết nối với chân XIN và XOUT và cần mắc thêm một tụ điện. Tụ điện này có giá trị phải phù hợp với chi tiết kỹ thuật của thạch anh hoặc bộ cộng hưởng. Khi LFXT1 ở chế độ HF thì bit LFXT1Sx lựa chọn biên độ hoạt động.

LFXT1 cũng có thể sử dụng nguồn tín hiệu bên ngoài ở chân XIN ở chế độ LF hoặc HF khi LFXT1Sx = 1, OSCOFF = 0 và XCAPx = 00. Khi sử dụng với nguồn tín hiệu bên ngoài thì tần số phải phù hợp với giá trị trong datasheet cho việc lựa chọn chế độ. Nếu tần số vào này thấp dưới giới hạn cho phép thì Bit LFXT1OF có thể được thiết lập cản trở CPU từ việc khóa LFXT1CLK.

Chương trình có thể ngắt LFXT1 bởi việc thiết lập OSCOFF nếu LFXT1CLK không bắt nguồn từ SMCLK hoặc MCLK.

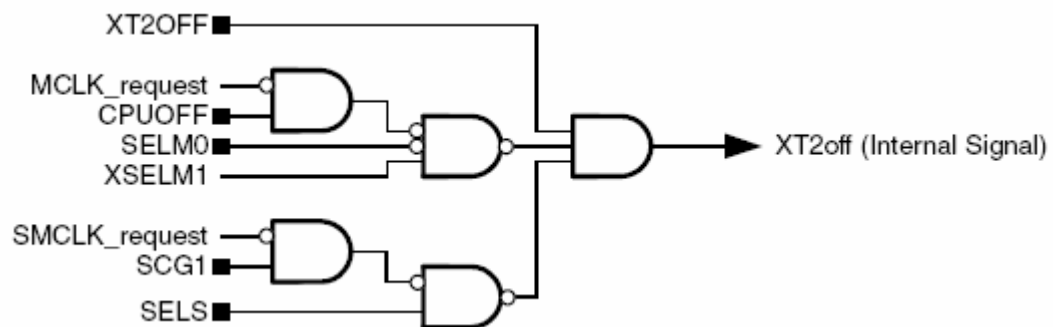


Hình IV.2: Ngắt tín hiệu cho bộ dao động LFXT1

IV.2.3. Bộ dao động XT2

Ở một vài họ MSP430 có thêm bộ dao động thạch anh thứ 2 đó là XT2. XT2 là nguồn của XT2CLK và nó mang những đặc điểm tương đồng với LFXT1 trong chế độ HF. Bit XT2Sx lựa chọn phạm vi hoạt động của XT2. Bit XT2OFF ngắt XT2 nếu XT2CLK không được sử dụng cho SMCLK và MCLK.

XT2 có thể sử dụng nguồn tín hiệu bên ngoài ở chân XT2IN khi Bit XT2Sx = 11 và XT2OFF = 0. Khi sử dụng với tín hiệu bên ngoài thì tần số này phải phù hợp với giá trị của bảng số liệu XT2. Khi tần số này thấp dưới mức qui định thì Bit XT2OF sẽ được thiết lập cản trở CPU bằng cách khóa XT2CLK.

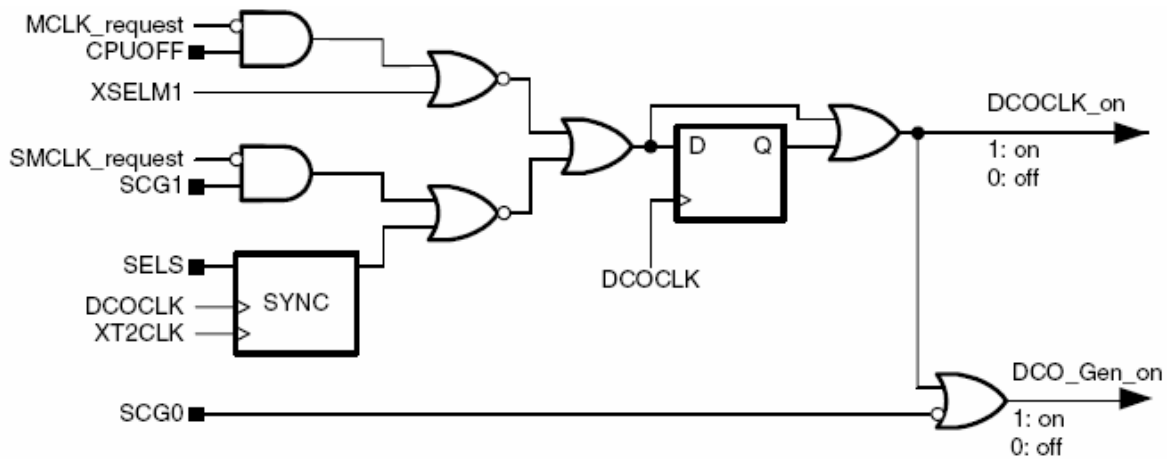


Hình IV.3: Tắt tín hiệu XT2

IV.2.4. Bộ dao động kĩ thuật số (DCO)

DCO là một bộ tích hợp dao động kĩ thuật số. Tần số DCO có thể được điều chỉnh bởi chương trình sử dụng những Bit DCOx, MODx, RSELx.

Chương trình có thể hủy DCOCLK bởi việc thiết lập SCG0 khi nó không từng là nguồn SMCLK và MCLK.



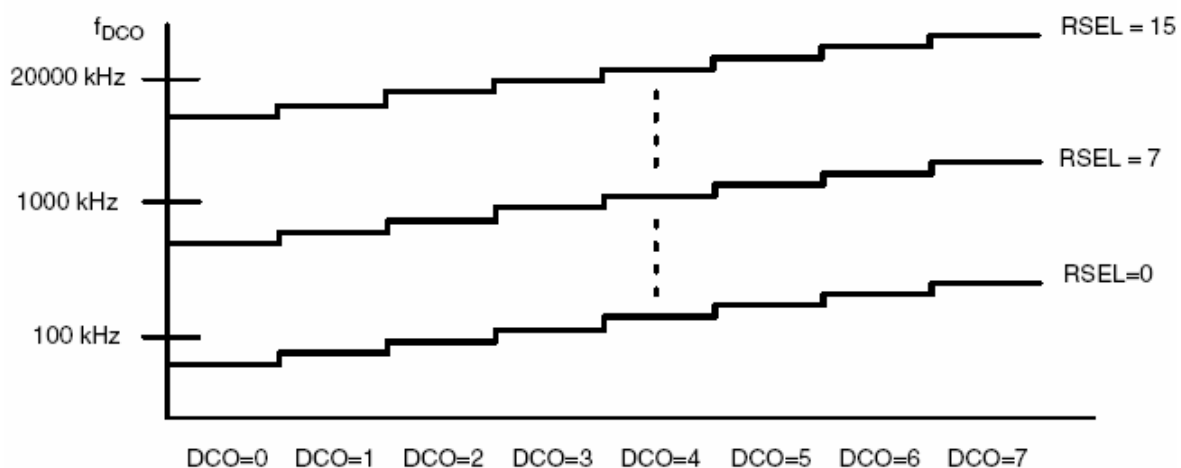
Hình IV.4: Điều khiển ON/OFF của DCO

Điều chỉnh tần số DCO:

Sau một PUC, $RSELx = 7$ và $DCOx = 3$, DCO bắt đầu với một tần số trung bình. SMCLK và MCLK có nguồn từ DCOCLK. Bởi vì CPU thực thi mã chương trình từ MCLK, mà MCLK thì lại bắt nguồn từ DCO. Code chương trình được thực thi bắt đầu từ PUC và chậm hơn $2\mu s$.

Tần số của DCOCLK được thiết lập bởi các tính năng sau:

- + Bốn Bit $RSELx$ cho phép lựa chọn 16 dãy tần số cho DCO. Những dãy tần số này thì được mặc định cho từng họ MSP430 và được trình bày trong Datasheet của từng loại.
- + Ba Bit $DCOx$ của dãy DCO được lựa chọn bởi các bit $RSELx$ đến 8 bước tần số được cách biệt xấp xỉ 10%.
- + Năm bit $MODx$, chuyển đổi tần số được chọn bởi các bit $DCOx$ và tần số kế tiếp cao hơn được thiết lập bởi $DCOx + 1$. Khi $DCOx = 07h$ thì những bit $MODx$ không chịu ảnh hưởng bởi vì DCO đã được thiết lập cao nhất cho việc lựa chọn dãy $RSELx$.



Hình IV.5: Dãy DCOx và bước RSELx

Đối với MSP430F2xx được hiệu chỉnh thành ghi DCOCTL và BCSCTL1 cho việc tồn trữ những tần số đặc biệt trong bộ nhớ thanh ghi A. Sử dụng những hiệu chỉnh thiết lập,

thông tin được sao chép vào thanh ghi DCOCTL và BCSCTL1. Việc hiệu chỉnh này ảnh hưởng đến các Bit DCOx, MODx, RSELx và xóa các Bit đó, ngoại trừ XT2OFF thì được thiết lập trở lại. Những Bit của BCSCTL1 có thể được đặt hoặc xóa bởi lệnh BIS.B hoặc BIC.B

; Set DCO to 1 MHz:

MOV.B &CALBC1_1MHZ,&BCSCTL1 ; Set range

MOV.B &CALDCO_1MHZ,&DCOCTL ; Set DCO step + modulation

IV.2.5. Bộ điều khiển DCO

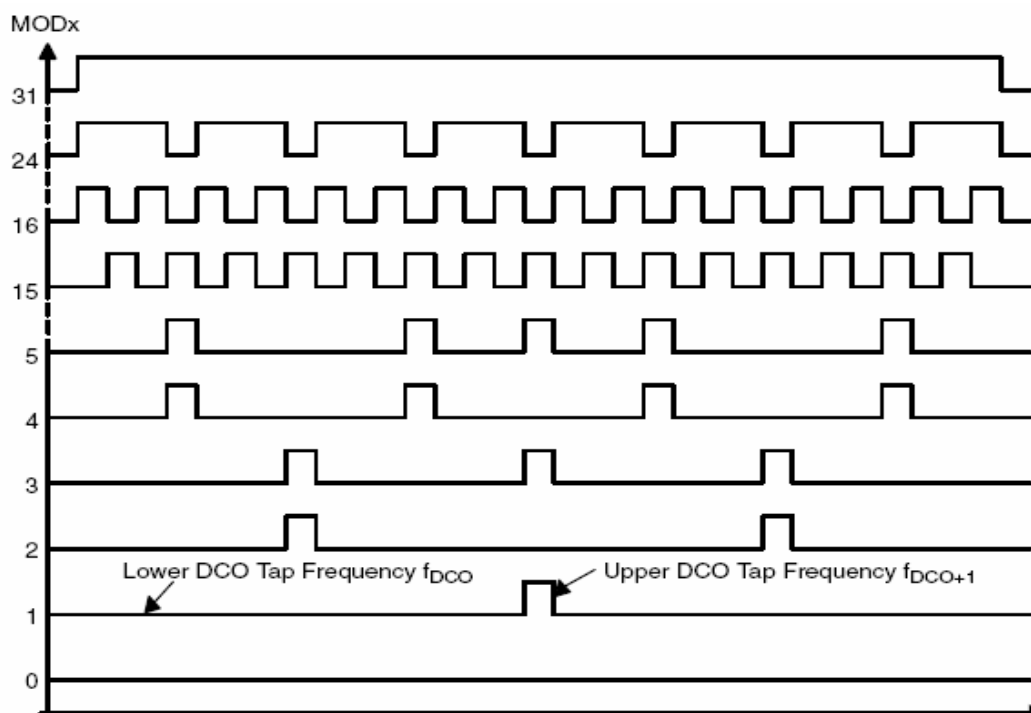
Bộ điều khiển là sự trộn của 2 tần số DCO là f_{DCO} và f_{DCO+} cung cấp một tần số phụ có ích giữa f_{DCO} và f_{DCO+} và mở rộng năng lượng Clock, giảm sự nhiễu điện từ (EMI). Bộ điều khiển pha trộn f_{DCO} và f_{DCO+} cho 32 chu kỳ DCOCLK và được thiết lập với Bit MODx. Khi MODx = 0 thì bộ điều khiển tắt.

Phương trình của bộ điều khiển :

$$t = (32 - \text{MODx}) \times t_{DCO} + \text{MODx} \times t_{DCO+} + 1$$

Bởi vì f_{DCO} thấp hơn tần số hiệu dụng và f_{DCO+} thì cao hơn tần số hiệu dụng, độ sai lệch tần số gần bằng 0. Sự sai lệch tần số hiệu dụng bằng 0 mỗi 32 chu kỳ DCOCLK.

Bộ điều khiển DCO được thiết lập bởi chương trình. DCOCLK có thể được so sánh với tần số ổn định của giá trị đã biết và được hiệu chỉnh với các Bit DCOx, RSELx, MODx.



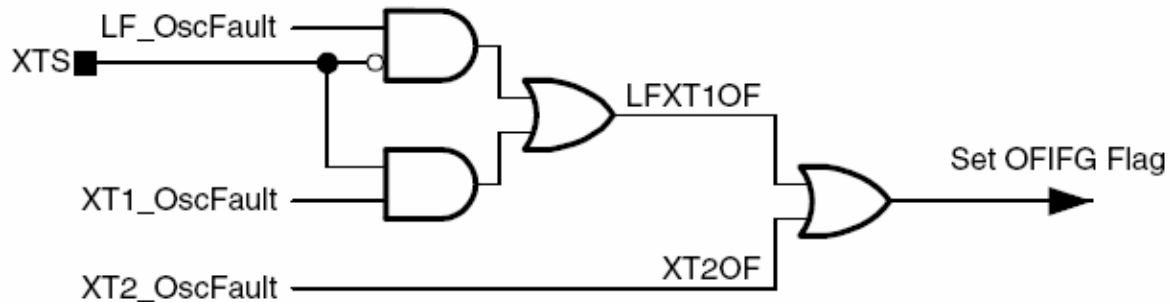
Hình IV.6: Biểu đồ điều khiển DCO

IV.2.6. Chế độ hoạt động an toàn của clock module+

+ Clock module+ được tích hợp những đặc điểm an toàn khi bị lỗi bộ dao động. Nó dò tìm những lỗi dao động của LFXT1 và XT2. Những lỗi dao động như sau:

- Tần số thấp bị đứt gãy (LFXT1OF) của LFXT1 trong chế độ LF.
- Tần số cao bị đứt gãy (LFXT1OF) của LFXT1 trong chế độ HF.

- Tần số cao bị đứt gãy (XT2OF) của XT2.
- + Những Bit LFXT1OF và XT2 sẽ được set nếu dao động thạch anh bị đứt gãy và hoạt động không còn chính xác nữa. Những Bit này sẽ được Set cho đến khi hết những lỗi dao động và sẽ tự động được xóa khi các dao động trở về bình thường.
- + Cờ OFIFG được Set và được chốt ở POR khi một lỗi dao động được tìm thấy (LFXT1OF và XT2OF). Khi OFIFG được Set, MCLK được bắt nguồn từ DCO, và nếu OIFE được Set, OFIFG yêu cầu một ngắt NMI. Khi ngắt được chấp hành thì OIFE tự động Reset trở lại. Cờ OFIFG phải được xóa bằng chương trình.



Hình IV.7: Chế độ hoạt động an toàn của clock module+

Nguồn MCLK từ dao động thạch anh:

Sau một PUC, Clock module+ sử dụng DCOCLK của MCLK. Nếu được yêu cầu, MCLK có nguồn từ LFXT1 và XT2.

Quá trình chuyển đổi MCLK bắt nguồn từ DCO Clock sang dao động thạch anh (LFXT1 và XT2) là:

1. Chuyển đổi trên dao động thạch anh và chọn chế độ tương thích.
2. Xóa cờ OFIFG
3. Chờ đợi ít nhất 50μs
4. Kiểm tra cờ OFIFG, lặp lại các bước cho đến khi cờ OFIFG được xóa hoàn toàn.

; Chọn LFXT1 (chế độ HF) cho MCLK

```
BIC.W #OSCOFF,SR          ; Bật OSC.
BIS.B #XTS,&BCSCTL1        ; Chọn chế độ HF
MOV.B #LFXT1S0,&BCSCTL3    ; 1-3MHz Thạch anh
```

L1:

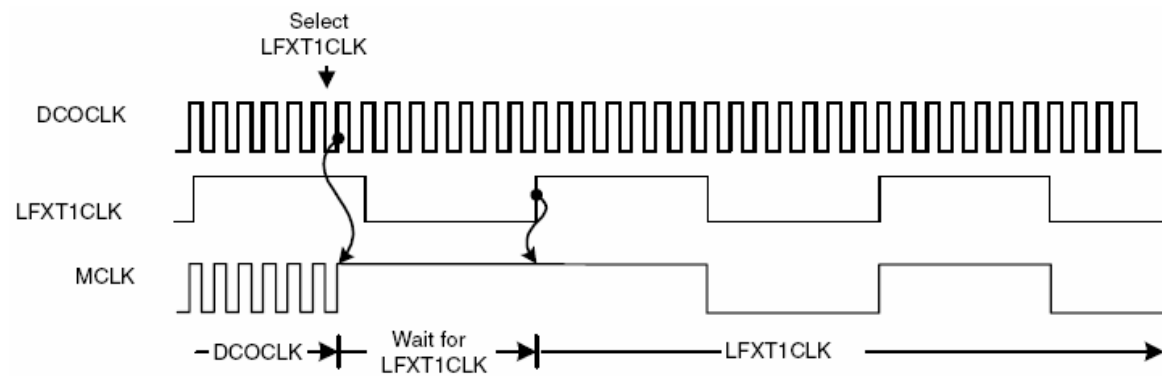
```
BIC.B #OFIFG,&IFG1         ; Xóa cờ OFIFG
MOV.W #0FFh,R15            ; Delay
```

L2:

```
DEC.W R15 ;
JNZ L2 ;
BIT.B #OFIFG,&IFG1         ; Kiểm tra Cờ OFIFG
JNZ L1                     ; Kiểm tra lại nếu khác 0
BIS.B #SELM1+SELM0,&BCSCTL2 ; chọn LFXT1CLK
```

IV.2.7. Sự đồng bộ của các tín hiệu xung Clock

Khi một sự chuyển đổi MCLK và SMCLK từ một nguồn này sang một nguồn khác thì sự chuyển đổi này phải được thực hiện một cách đồng bộ.



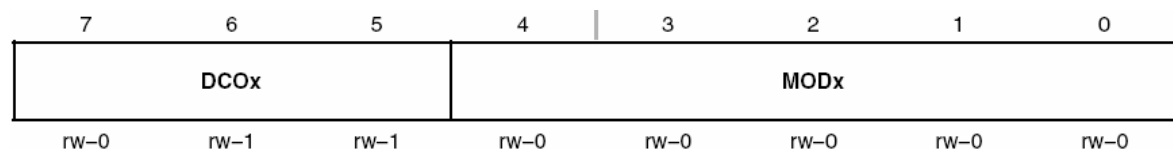
Hình IV.8: Sự chuyển đổi MCLK từ DCOCLK sang LFTX1CLK

IV.3. Các thanh ghi của Clock module+

Register	Short Form	Register Type	Address	Initial State
DCO control register	DCOCTL	Read/write	056h	060h with PUC
Basic clock system control 1	BCSCTL1	Read/write	057h	087h with POR
Basic clock system control 2	BCSCTL2	Read/write	058h	Reset with PUC
Basic clock system control 3	BCSCTL3	Read/write	053h	005h with PUC
SFR interrupt enable register 1	IE1	Read/write	000h	Reset with PUC
SFR interrupt flag register 1	IFG1	Read/write	002h	Reset with PUC

Bảng IV.1: Các thanh ghi của Clock module+

IV.3.1. DCOCTL, DCO control register



+ **DCOx**: Bit 5 đến 7, lựa chọn tần số DCO. Những Bit này lựa chọn 8 tần số DCO riêng biệt trong dãy được xác định bằng việc thiết lập RSELx.

+ **MODx**: Bit 0 đến 4, lựa chọn bộ điều khiển. Bit này xác định tần số f_{DCO+1} xuất hiện bao nhiêu lần trong 32 chu kỳ DCOCLK. Trong suốt chu kỳ còn lại (32-MOD) tần số f_{DCO} thì được sử dụng. Không cho phép khi DCOx = 7.

IV.3.2. BCSTCTL1, Basic Clock System Control Register 1

7	6	5	4	3	2	1	0
XT2OFF	XTS†	DIVAx		RSELx			
rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-0	rw-1	rw-1	rw-1

+ **XT2OFF**: Bit 7, tắt XT2.

XT2OFF = 0: XT2 mở

XT2OFF = 1: XT2 đóng nếu nó không sử dụng cho MCLK và SMCLK

+ **XTS**: Bit 6, lựa chọn chế độ LFXT1.

XTS = 0: Chế độ tần số thấp

XTS = 1: Chế độ tần số cao.

+ **DIVAx**: Bit 4-5, bộ chia ACLK

00: chia 1

01: chia 2

10: chia 4

11: chia 8

+ **RSELx**: Bit 0-3, lựa chọn dãy tần số. Có 16 dãy tần số khả dụng. Dãy tần số thấp nhất được chọn bằng cách cho RSELx = 0. RSEL3 được bỏ qua khi DCOR = 1

IV.3.3. BCSTCTL2, Basic Clock System Control Register 2

7	6	5	4	3	2	1	0
SELMx		DIVMx		SELS	DIVSx		DCOR†
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

+ **SELMx**: Bit 6-7, lựa chọn nguồn của MCLK.

00: DCOCLK

01: DCOCLK

10: XT2CLK khi XT2 đang có sẵn trên Chip. LFXT1CLK hoặc VCLOCK nếu XT2 không có sẵn trên Chip.

11: LFXT1CLK hoặc VCLOCK

+ **DIVMx**: Bit 4-5, bộ chia MCLK.

00: chia 1

01: chia 2

10: chia 3

11: chia 4

+ **SELS**: Bit 3, lựa chọn nguồn SMCLK.

0: DCOCLK

1: XT2CLK khi XT2 đang có sẵn trên Chip. LFXT1CLK hoặc VCLOCK nếu XT2 không có sẵn trên Chip.

+ **DIVSx**: Bit 1-2, bộ chia SMCLK.

00: chia 1
 01: chia 2
 10: chia 3
 11: chia 4

+ **DCOR**: Bit 0, lựa chọn điện trở DCO

0: Điện trở trong.
 1: Điện trở ngoài.

IV.3.4. BCSCTL3, Basic Clock System Control Register 3

7	6	5	4	3	2	1	0
XT2Sx		LFXT1Sx		XCAPx		XT2OF†	LFXT1OF
rw-0		rw-0		rw-0		r0	r-(1)

+ **XT2Sx**: Bit6-7, lựa chọn dãy tần số XT2.

00: 0.4 – 1 MHz dao động thạch anh hoặc bộ cộng hưởng
 01: 1 – 3 MHz dao động thạch anh hoặc bộ cộng hưởng
 10: 3 – 16 MHz dao động thạch anh hoặc bộ cộng hưởng
 11: 0.4 – 16 MHz nguồn dao động từ bên ngoài

+ **LFXT1Sx**: Bit 4-5, lựa chọn loại xung Clock tần số thấp và lựa chọn dãy tần số của LFXT1. Lựa chọn giữa LFXT1 và VLO khi XTS = 0. Lựa chọn dãy tần số LFXT1 khi XTS = 1.

Khi XTS = 0:

00: LFXT1 tần số 32768 Hz dao động thạch anh
 01: Dự trữ
 10: VLOCLK
 11: Nguồn ngoài.

Khi XTS = 1:

00: 0.4 – 1 MHz dao động thạch anh hoặc bộ cộng hưởng
 01: 1 – 3 MHz dao động thạch anh hoặc bộ cộng hưởng
 10: 3 – 16 MHz dao động thạch anh hoặc bộ cộng hưởng
 11: 0.4 – 16 MHz nguồn dao động từ bên ngoài

+ **XCAPx**: Bit 2-3, lựa chọn điện dung.

00: 1pF
 01: 6pF
 10: 10pF
 11: 12.5pF

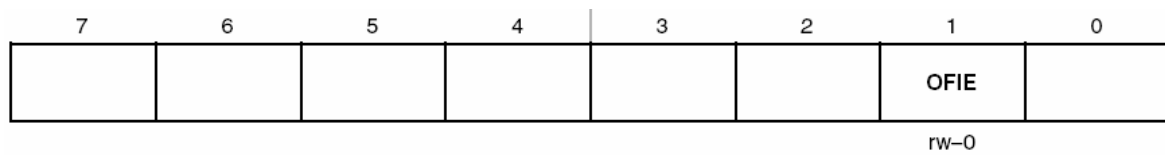
+ **XT2OF**: Bit 1, lỗi đứt gãy dao động XT2

0: Không có lỗi xuất hiện
 1: Có lỗi xuất hiện.

+ **LFXT1OF**: Bit 0, lỗi đứt gãy dao động LFXT1

0: Không có lỗi xuất hiện
 1: Có lỗi xuất hiện.

IV.3.5. IE1, Interrupt Enable Register 1



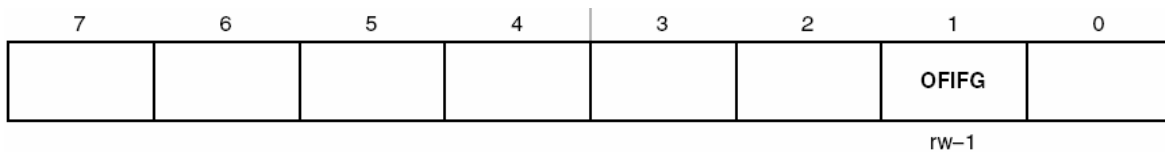
Bit 0 và Bit 2 – 7: Có thể sử dụng cho các mô đun khác.

OFIE: Bit 1, cho phép ngắt khi dao động bị lỗi. Cho phép cờ ngắt OFIEG hoạt động.

0: Không cho phép ngắt.

1: Cho phép ngắt.

IV.3.6. IFG1, Interrupt Flag Register 1



Bit 0 và Bit 2 – 7: Có thể sử dụng cho các mô đun khác.

OFIFG: Bit 1, cho phép cờ ngắt khi dao động bị lỗi.

0: Không cho phép ngắt.

1: Cho phép ngắt.

V. BỘ NHỚ FLASH

V.1. Giới thiệu về bộ nhớ Flash

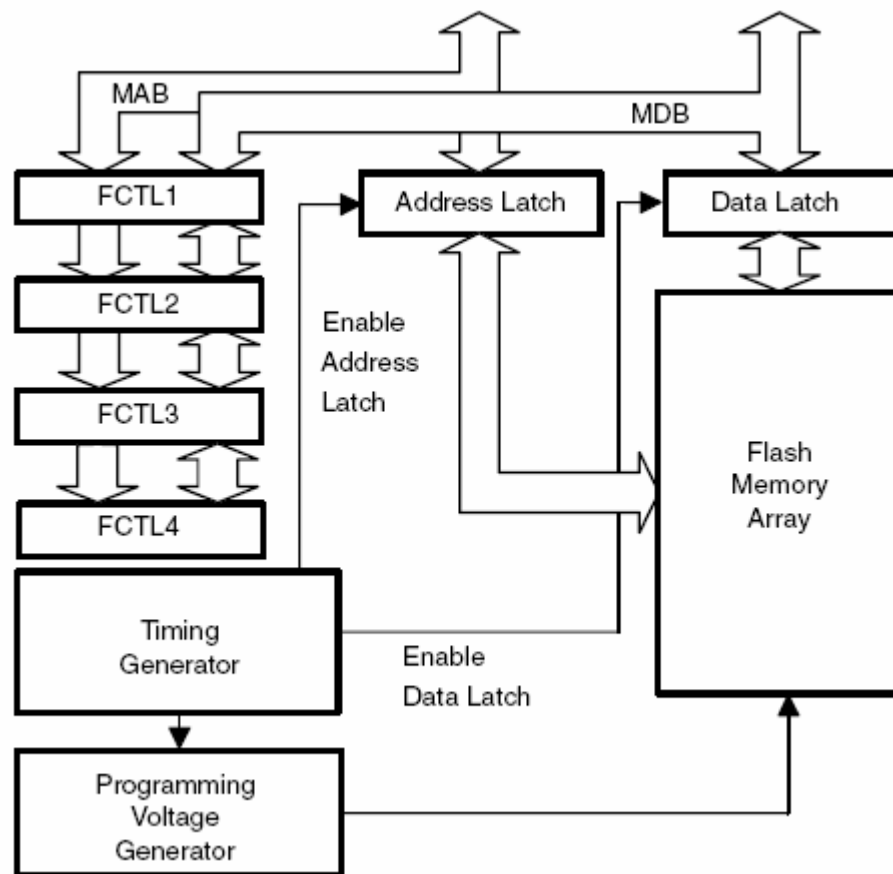
Bộ nhớ thông thường có 2 loại là bộ nhớ chỉ đọc ROM và bộ nhớ truy cập ngẫu nhiên RAM, bộ nhớ Flash là một sự pha trộn của 2 loại bộ nhớ này.

Đặc điểm bộ nhớ Flash là:

- + Chi phí thấp
- + Đọc/Ghi nhanh
- + An toàn
- + Mật độ cao

Bộ nhớ Flash MSP430 bao gồm nhưng Bit, Byte, địa chỉ từ và khả năng lập trình. Mỗi mô đun bộ nhớ Flash được tích hợp sẵn bộ điều khiển dùng để điều khiển các hoạt động ghi, xóa bộ nhớ. Bộ điều khiển này gồm 3 bộ: Bộ tạo dao động chủ, bộ phát điện áp cho lập trình, bộ xóa.

Điện áp nguồn Vcc tối thiểu trong hoạt động viết hoặc xóa bộ nhớ Flash là 2.7V. Nếu điện áp nguồn xuống thấp dưới mức này thì bộ nhớ Flash không thể hoạt động được.



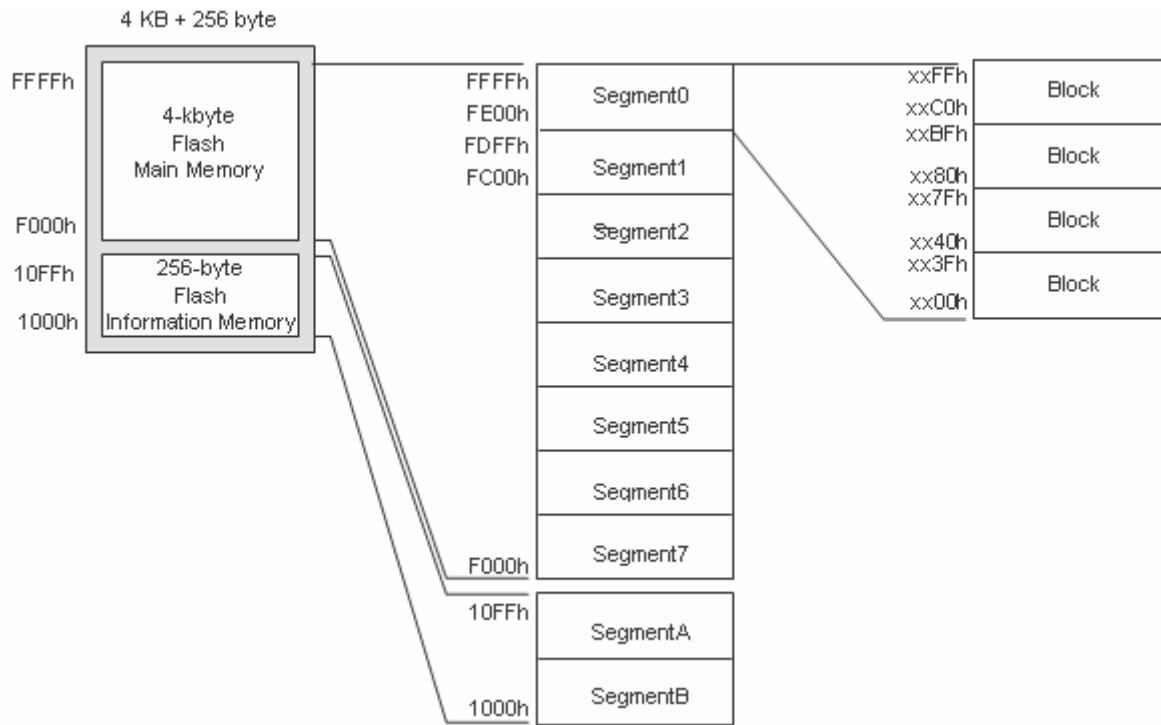
Hình V.1: Sơ đồ khối chức năng của bộ nhớ Flash

V.2. Sự phân đoạn trong bộ nhớ Flash

Bộ nhớ Flash của MSP430 chia thành những phân đoạn như: Bít đơn, byte hoặc Word có thể được ghi vào bộ nhớ Flash. Đây là những kích thước nhỏ nhất của bộ nhớ Flash có thể bị xóa.

Bộ nhớ Flash được phân chia thành một bộ nhớ chính và thành phần bộ nhớ thông tin. Không có sự khác biệt về hoạt động của bộ nhớ chính và thành phần bộ nhớ thông tin. Code chương trình và dữ liệu có thể được lưu trong cả hai bộ phận. Sự khác nhau giữa hai phân vùng này là kích thước của phân đoạn và các địa chỉ vật lý.

Thành phần bộ nhớ thông tin có 2 phân đoạn 128 byte (MSP430F1101 chỉ có một). Bộ nhớ chính có hai hoặc nhiều hơn những phân đoạn 512 byte. Những phân đoạn này chia thành những khối 64 byte bắt đầu ở địa chỉ 0xx00h, 0xx40h, 0xx80h, 0xxC0h và kết thúc ở 0xx3Fh, 0xx7Fh, 0xxBFh, 0xFFh.



Hình V.2: Những phân đoạn trong bộ nhớ Flash

V.3. Các bộ điều khiển bộ nhớ Flash

Bộ nhớ Flash có các bộ điều khiển như sau:

- + Bộ điều khiển lập trình và hoạt động xóa
- + Có 3 hoặc 4 thanh ghi
- + Một bộ định thời có nguồn từ ACLK, MCLK và SMCLK

VI. DIGITAL I-O

VI.1. Giới thiệu về ngõ vào/ra số (Digital I/O)

MSP430 có các Port xuất/ nhập dữ liệu gọi là các Port I/O. Mỗi Port có 8 chân I/O, số lượng Port tùy thuộc vào từng họ MSP430. Mỗi chân I/O được thiết lập các chức năng vào/ra một cách riêng rẽ, và các chân I/O này cũng dùng để đọc hoặc ghi dữ liệu một cách độc lập.

P1 và P2 có thể dùng để ngắt. Mỗi chân I/O của P1, P2 có tác dụng ngắt độc lập và có thể thiết lập ngắt cạnh lên hoặc cạnh xuống của tín hiệu vào.

Các ngõ vào/ra số được mô tả như sau:

- + Những chân I/O riêng rẽ có thể được lập trình một cách độc lập.
- + P1 và P2 có thể thiết lập làm những ngắt riêng rẽ.
- + Những thanh ghi dữ liệu vào/ra độc lập.
- + Có thể thiết lập tăng lên hoặc giảm xuống của điện trở một cách độc lập.

VI.2. Các hoạt động vào/ra số

VI.2.1. Thanh ghi dữ liệu vào PxIN

Mỗi Bit trong mỗi thanh ghi PxIN phản ánh giá trị của tín hiệu vào ở chân I/O tương ứng khi các chân này được thiết lập chức năng I/O.

Bit = 0: Ngã vào mức thấp

Bit = 1: Ngã vào mức cao

VI.2.2. Thanh ghi dữ liệu xuất PxOUT

Mỗi Bit trong mỗi thanh ghi PxOUT là giá trị xuất ra trên các chân I/O tương ứng khi các chân này được thiết lập chức năng I/O. Xuất dữ liệu có hướng và việc tăng giảm điện trở thì không được phép.

Bit = 0: Ngã ra mức thấp

Bit = 1: Ngã ra mức cao

Nếu các chân điều khiển tăng-giảm điện trở bị mất hiệu lực thì các Bit tương ứng trong thanh ghi PxOUT có tác dụng lựa chọn việc tăng giảm điện trở.

Bit = 0: Giảm xuống.

Bit = 1: Tăng lên

VI.2.3. Các thanh ghi định hướng PxDIR

Mỗi Bit trong mỗi thanh ghi PxDIR lựa chọn định hướng của chân I/O tương ứng, bất chấp chức năng đã được chọn của chân. Những Bit PxDIR cho những chân I/O được lựa chọn cho những chức năng khác phải được thiết lập theo yêu cầu của chức năng đó.

Bit = 0: Những chân của Port được định hướng là ngã vào.

Bit = 1: Những chân của Port được định hướng là ngã ra.

VI.2.4. Thanh ghi cho phép tăng/giảm điện trở PxREN

Mỗi Bit của mỗi thanh ghi PxREN dùng để cho phép hoặc không cho phép sự tăng/giảm điện trở của các chân I/O tương ứng. Những Bit tương ứng của thanh ghi PxOUT được chọn nếu chân được tăng lên hoặc giảm xuống.

Bit = 0: Tăng/giảm điện trở không được cho phép.

Bit = 1: Tăng/giảm điện trở được cho phép.

VI.2.5. Thanh ghi chức năng lựa chọn PxSEL và PxSEL2

Các chân này được đa hợp với các chức năng của các mô đun ngoại vi khác. Mỗi Bit PxSEL và PxSEL2 được sử dụng cho việc lựa chọn chức năng của các chân: Chức năng I/O xuất nhập dữ liệu hoặc chức năng mô đun giao tiếp ngoại vi.

PxSEL2	PxSEL	Chức năng của chân
0	0	Chức năng I/O
0	1	Chức năng mô đun giao tiếp ngoại vi ban đầu
1	0	Được dự trữ
1	1	Chức năng mô đun giao tiếp ngoại vi thứ hai

Bảng VI.1: Lựa chọn chức năng của chân

➤ Ví dụ:

;tín hiệu ra ACLK trên P2.0

BIS.B #01h,&P2SEL ; chọn chức năng ACLK

BIS.B #01h,&P2DIR ; chọn chức năng out cho P2.0

VI.2.6. Những ngắt P1 và P2

Mỗi chân của P1 và P2 có thể làm một ngắt yêu cầu. Các chân này được thiết lập với các thanh ghi phụ như:

➤ **Cho phép ngắt (PxIE)**

Thanh ghi này cho phép ngắt trên những chân riêng rẽ.

Bit = 0: không cho phép ngắt.

Bit = 1: Cho phép ngắt.

Mỗi Bit cho phép ngắt PxIE này được liên kết với cờ ngắt PxIFG tương ứng.

Việc ghi vào thanh ghi PxOUT và PxDIR có thể ảnh hưởng đến việc điều chỉnh PxIE.

➤ **Thanh ghi lựa chọn ngắt cạnh (PxIES)**

Lựa chọn loại ngắt khi xuất hiện sự biến đổi tín hiệu (Nếu PxIE và GIE được Set)

Bit = 0: Ngắt cạnh lên

Bit = 1: Ngắt cạnh xuống

➤ **Thanh ghi cờ ngắt (PxIFG)**

Cờ ngắt này được Set tự động theo chương trình đã được lập trình trước khi xuất hiện một sự chuyển đổi tín hiệu. Cờ PxIFG phải sử dụng chương trình để Reset.

Bit = 0: Không có ngắt.

Bit = 1: Có ngắt.

VI.3. Các thanh ghi I/O

Port	Thanh ghi	Từ khóa	Địa chỉ	Loại thanh ghi	Trạng thái ban đầu
P1	Ngõ vào	P1IN	020H	Chỉ đọc	--
	Ngõ ra	P1OUT	021H	Ghi/đọc	Không đổi
	Định hướng	P1DIR	022H	Ghi/đọc	Reset với PUC
	Cờ ngắt	P1IFG	023H	Ghi/đọc	Reset với PUC
	Ngắt cạnh	P1IES	024H	Ghi/đọc	Không đổi
	Cho phép ngắt	P1IE	025H	Ghi/đọc	Reset với PUC
	Chọn cổng	P1SEL	026H	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P1SEL2	041H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P1REN	027H	Ghi/đọc	Reset với PUC
P2	Ngõ vào	P2IN	028H	Chỉ đọc	--
	Ngõ ra	P2OUT	029H	Ghi/đọc	Không đổi
	Định hướng	P2DIR	02AH	Ghi/đọc	Reset với PUC
	Cờ ngắt	P2IFG	02BH	Ghi/đọc	Reset với PUC
	Ngắt cạnh	P2IES	02CH	Ghi/đọc	Không đổi
	Cho phép ngắt	P2IE	02DH	Ghi/đọc	Reset với PUC
	Chọn cổng	P2SEL	02EH	Ghi/đọc	0C0h với PUC
	Chọn cổng 2	P2SEL2	042H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P2REN	02FH	Ghi/đọc	Reset với PUC
P3	Ngõ vào	P3IN	018H	Chỉ đọc	--
	Ngõ ra	P3OUT	019H	Ghi/đọc	Không đổi
	Định hướng	P3DIR	01AH	Ghi/đọc	Reset với PUC
	Chọn cổng	P3SEL	01BH	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P3SEL2	043H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P3REN	010H	Ghi/đọc	Reset với PUC

P4	Ngã vào	P4IN	01CH	Chỉ đọc	--
	Ngã ra	P4OUT	01DH	Ghi/đọc	Không đổi
	Định hướng	P4DIR	01EH	Ghi/đọc	Reset với PUC
	Chọn cổng	P4SEL	01FH	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P4SEL2	044H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P4REN	011H	Ghi/đọc	Reset với PUC
P5	Ngã vào	P5IN	030H	Chỉ đọc	--
	Ngã ra	P5OUT	031H	Ghi/đọc	Không đổi
	Định hướng	P5DIR	032H	Ghi/đọc	Reset với PUC
	Chọn cổng	P5SEL	033H	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P5SEL2	045H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P5REN	012H	Ghi/đọc	Reset với PUC
P6	Ngã vào	P6IN	034H	Chỉ đọc	--
	Ngã ra	P6OUT	035H	Ghi/đọc	Không đổi
	Định hướng	P6DIR	036H	Ghi/đọc	Reset với PUC
	Chọn cổng	P6SEL	037H	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P6SEL2	046H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P6REN	013H	Ghi/đọc	Reset với PUC
P7	Ngã vào	P7IN	038H	Chỉ đọc	--
	Ngã ra	P7OUT	03AH	Ghi/đọc	Không đổi
	Định hướng	P7DIR	03CH	Ghi/đọc	Reset với PUC
	Chọn cổng	P7SEL	03EH	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P7SEL2	047H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P7REN	014H	Ghi/đọc	Reset với PUC
P8	Ngã vào	P8IN	039H	Chỉ đọc	--
	Ngã ra	P8OUT	03BH	Ghi/đọc	Không đổi
	Định hướng	P8DIR	03DH	Ghi/đọc	Reset với PUC
	Chọn cổng	P8SEL	03FH	Ghi/đọc	Reset với PUC
	Chọn cổng 2	P8SEL2	048H	Ghi/đọc	Reset với PUC
	Cho phép điện trở	P8REN	015H	Ghi/đọc	Reset với PUC

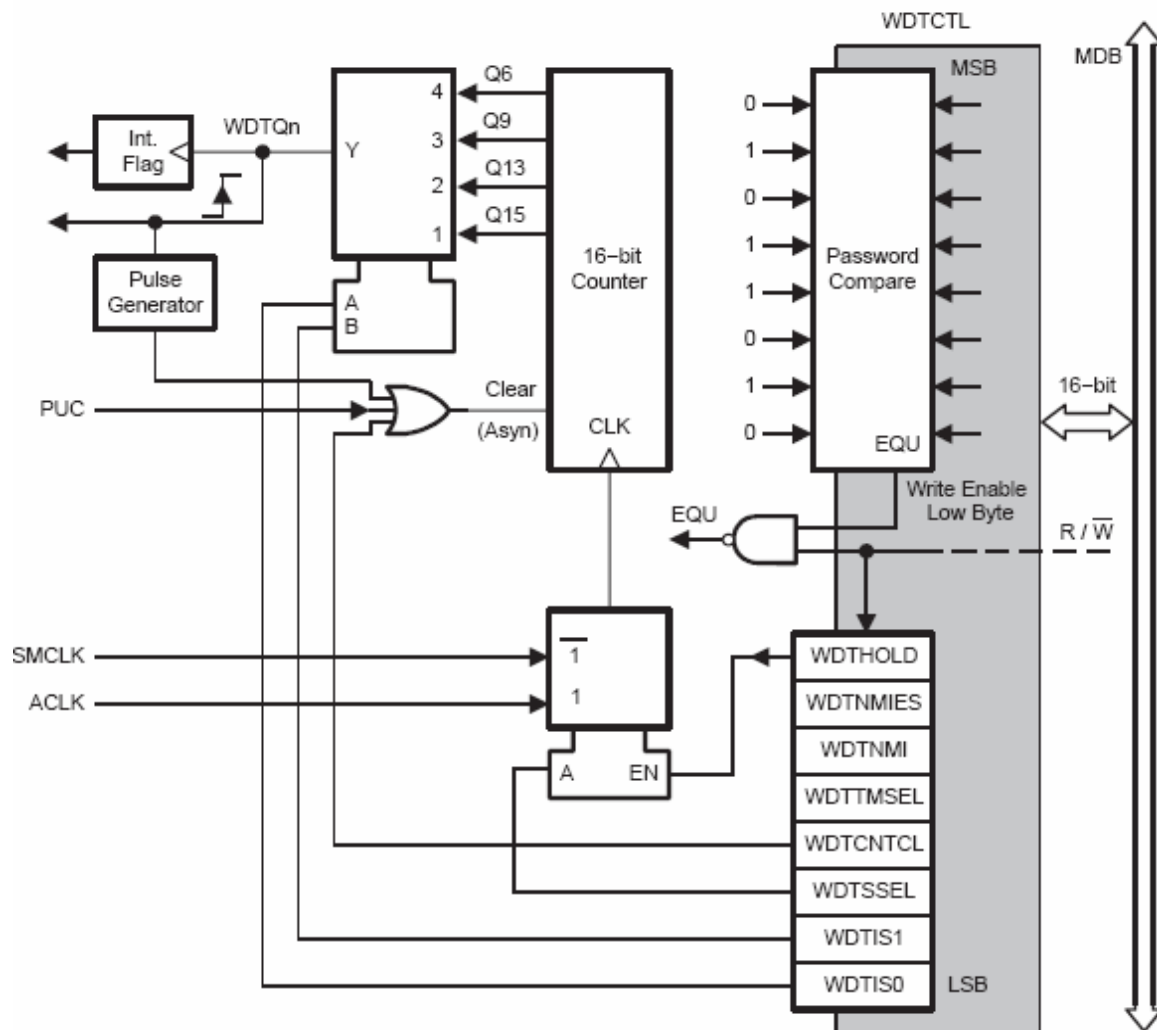
Bảng VI.2: Các cách định nghĩa I/O

VII. BỘ ĐỊNH THỜI GIÁM SÁT

VII.1. Giới thiệu về Watchdog timer

Chức năng cơ bản của Watchdog timer (WDT) là điều khiển hệ thống khởi động lại khi chương trình xuất hiện lỗi. Chẳng hạn như chương trình bị kẹt trong một vòng lặp vô tận thì lúc đó phải dùng đến Watchdog timer để gỡ lỗi. Nếu chức năng giám sát không được sử dụng trong một số ứng dụng thì Watchdog timer có thể được thiết lập như là một bộ định thời khoảng và có thể sinh ra ngắt ở những khoảng thời gian được lựa chọn.

Những Mô đun của watchdog timer được mô tả theo sơ đồ sau:



Hình VII.1: Sơ đồ khối của Watchdog timer

VII.2. Hoạt động của Watchdog timer

Watchdog timer có thể được thiết lập như là một bộ giám sát hoặc bộ định thời khoảng với thanh ghi WDTCTL. Thanh ghi WDTCTL cũng chứa những Bit điều khiển để thiết lập cho chân $\overline{\text{RST}}/\text{NMI}$. WDTCTL là thanh ghi 16 Bit dùng để ghi/đọc dữ liệu và được bảo vệ bằng mật khẩu. Bất kỳ đọc hoặc viết dữ liệu đều sử dụng chỉ lệnh Word và khi ghi dữ liệu phải bao gồm từ khóa 05Ah ở Byte cao. Khi ghi vào thanh ghi WDTCTL bất kỳ giá trị nào hơn 05Ah ở Byte cao thì sẽ vi phạm sự an toàn và khởi động ngắt PUC bất chấp ở chế độ thời gian nào. Còn việc đọc dữ liệu thì chỉ đọc được các giá trị có 069h ở Byte cao.

VII.2.1. Watchdog timer counter (WDTCNT)

- + WDTCNT là một bộ đếm lên 16 Bit và nó không định địa chỉ trực tiếp được.
- + WDTCNT thì được điều khiển thông qua watchdog timer điều khiển thanh ghi WDTCTL.
- + WDTCNT có thể được bắt nguồn từ ACLK hoặc SMCLK. Nguồn xung Clock thì được lựa chọn bởi Bit WDTSEL.

VII.2.2. Chế độ giám sát

Sau một ngắt PUC, mô đun WDT được thiết lập trong chế độ giám sát với độ rộng xung ~32ms sử dụng DCOCLK. Người sử dụng phải cài đặt, tạm dừng hoặc xóa WDT trong khoảng thời gian khởi động ban đầu hoặc một ngắt PUC sẽ được sinh ra. Khi WDT được thiết lập hoạt động trong chế độ giám sát, việc ghi vào thanh ghi WDTCTL với một từ khóa sai hoặc hết thời gian được chọn sẽ sinh ra ngắt PUC. PUC sẽ khởi động lại WDT ở chế độ mặc định và thiết lập chân $\overline{\text{RST}}/\text{NMI}$ ở chế độ Reset.

VII.2.3. Chế độ hẹn giờ

Thiết lập Bit WDTMSEL để chọn chế độ hẹn giờ. Chế độ này có thể sử dụng cho những ngắt có chu kỳ. Trong chế độ hẹn giờ thì cờ WDTIFG thiết lập khoảng thời gian hẹn giờ. PUC thì không được sinh ra trong chế độ hẹn giờ khi hết thời gian chọn và cờ WDTIFG cho phép Bit WDTIE không thay đổi.

Khi Bit WDTIE và GIE được Set thì cờ WDTIFG sẽ yêu cầu ngắt. Cờ ngắt WDTIFG sẽ tự động khởi động lại nếu ngắt của nó yêu cầu được duy trì, hoặc có thể tự khởi động bằng phần mềm. Địa chỉ véc tơ ngắt trong chế độ hẹn giờ này thì khác trong các chế độ giám sát khác.

VII.2.4. Ngắt trong watchdog timer

WDT sử dụng 2 Bit trong SFRs cho điều khiển ngắt:

- Cờ ngắt WDT là WDTIFG được định vị trong IFG1.0
- Ngắt cho phép WDT là WDTIE được định vị trong IE1.0

Khi sử dụng WDT trong chế độ giám sát thì cờ WDTIFG là nguồn của véc tơ ngắt. WDTIFG có thể được sử dụng trong chương trình ngắt nếu WDT là nguyên nhân của việc khởi động lại thiết bị. Nếu cờ này được Set sau khi WDT khởi động chế độ Reset bởi sự hết thời gian định thời hoặc sự vi phạm khóa an toàn. Nếu cờ WDTIFG được xóa thì sự Reset được gây ra bởi một nguồn khác.

Khi sử dụng WDT trong chế độ hẹn giờ, cờ WDTIFG được Set sau khi thời gian hẹn giờ được chọn và yêu cầu phải có một ngắt hẹn giờ khi WDTIE và GIE được Set. Véc tơ ngắt này khác với véc tơ Reset ở chế độ giám sát. Trong chế độ hẹn giờ cờ WDTIFG được Reset tự động khi ngắt được duy trì hoặc có thể từ lập trình.

VII.2.5 Ví dụ

Bất kì việc ghi vào thanh ghi WDTCTL đều phải có một từ 05Ah (WDTPW) ở Byte cao.

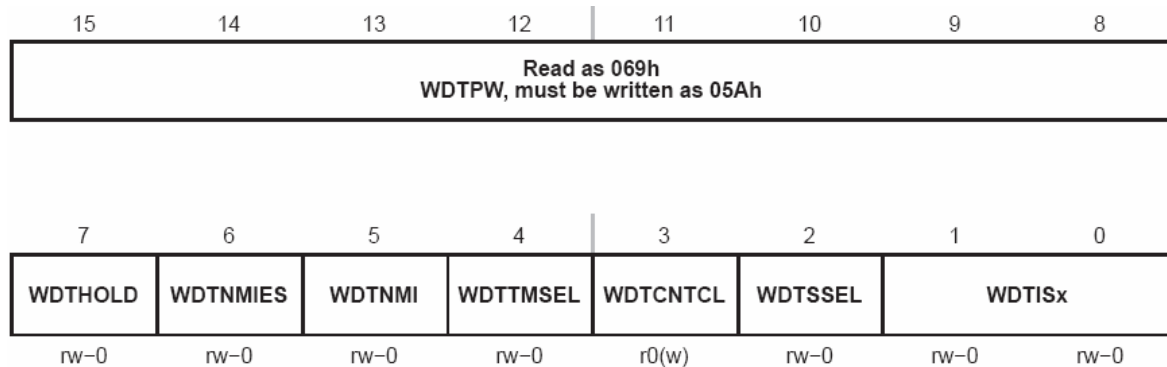
```
; Xóa watchdog timer một cách có chu kỳ
MOV #WDTPW+WDTCNTCL,&WDTCTL
;
; Thay đổi watchdog timer
MOV #WDTPW+WDTCNTL+SSEL,&WDTCTL
;
; Dừng watchdog timer
MOV #WDTPW+WDTHOLD,&WDTCTL
;
; thay đổi WDT trong chế độ hẹn giờ, clock/8192 interval
MOV #WDTPW+WDTCNTCL+WDTMSEL+WDTIS0,&WDTCTL
```


VII.3. Những thanh ghi của watchdog timer

Watchdog timer có 3 thanh ghi là WDTCTL, IE1 và IFG1 như trình bày ở bảng bên dưới.

Register	Short Form	Register Type	Address	Initial State
Watchdog timer control register	WDTCTL	Read/write	0120h	06900h with PUC
SFR interrupt enable register 1	IE1	Read/write	0000h	Reset with PUC
SFR interrupt flag register 1	IFG1	Read/write	0002h	Reset with PUC†

VII.3.1. WDTCTL



+ **WDTPW**: Bit 8 – 15, luôn đọc bằng 069h và ghi bằng 05Ah hoặc PUC sẽ được sinh ra.

+ **WDTHOLD**: Bit 7, đây là Bit dừng của watchdog timer.

WDTHOLD = 1: watchdog timer ngưng.

WDTHOLD = 0: watchdog timer không dừng lại.

+ **WDTNMIES**: Bit 6, lựa chọn cạnh ngắt NMI watchdog timer. Bit này lựa chọn cạnh ngắt cho ngắt NMI khi WDTNMI = 1.

WDTNMIES = 1: Cạnh ngắt rớt xuống thấp.

WDTNMIES = 0: Cạnh ngắt dâng lên cao.

+ **WDTNMI**: Bit 5, lựa chọn chức năng cho chân $\overline{\text{RST}}$ /NMI

WDTNMI = 0: reset

WDTNMI = 1: Ngắt NMI

+ **WDTTMSEL**: Bit 4, Lựa chọn chế độ của watchdog timer.

WDTTMSEL = 0: Chế độ giám sát

WDTTMSEL = 1: Chế độ hẹn giờ

+ **WDTCNTCL**: Bit 3, xóa bộ đếm watchdog timer. WDTCNTCL = 1 thì xóa giá trị đếm về 0000h.

+ **WDTSSEL**: Bit 2, lựa chọn nguồn xung Clock cho watchdog timer.

WDTSSEL = 0: SMCLK

WDTSSEL = 1: ACLK

+ **WDTISx**: Bit 0 -1, lựa chọn watchdog timer để thiết lập cờ WDTIFG và/hoặc sinh ra một PUC.

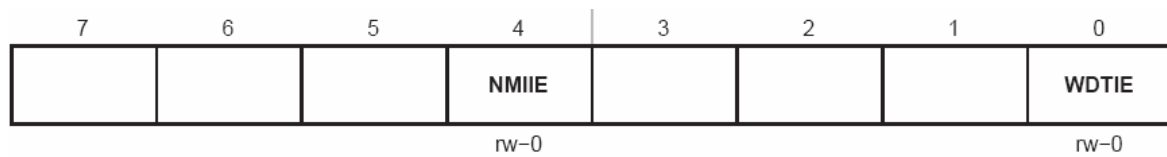
00: Watchdog clock source /32768

01: Watchdog clock source /8192

10: Watchdog clock source /512

11: Watchdog clock source /64

VII.3.2. IE1, thanh ghi cho phép ngắt 1



+ Bit 5 đến 7: Những Bit này sử dụng cho những Mô đun khác.

+ **NMIIE**: Bit 4, cho phép ngắt NMI. Bởi vì Bit này có thể sử dụng cho các mô đun khác nên được đề nghị khi Set hoặc Clear Bit này nên sử dụng lệnh BIS.B hoặc BIC.B hơn là MOV.B hoặc CLR.B.

0: Không cho phép ngắt

1: Cho phép ngắt

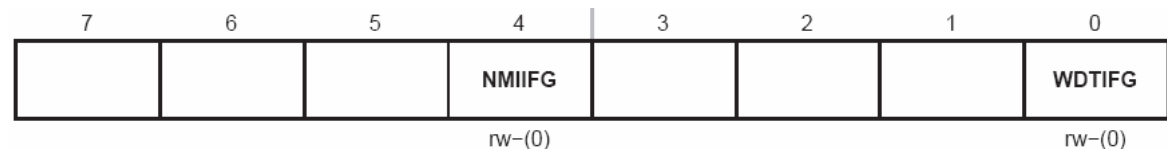
+ Bit 1 đến 3: Những Bit này sử dụng cho những Mô đun khác.

+ **WDTIE**: Bit 0, cho phép ngắt watchdog timer. Bit này cho phép chờ ngắt WDTIFG cho chế độ hẹn giờ. Không cần thiết phải thiết lập Bit này cho chế độ giám sát vì IE1 có thể sử dụng cho các mô đun khác. Nó thì được đề nghị khi Set hoặc Clear Bit này nên sử dụng lệnh BIS.B hoặc BIC.B hơn là MOV.B hoặc CLR.B.

0: Không cho phép ngắt

1: Cho phép ngắt

VII.3.3. IFG1, thanh ghi chờ ngắt 1



+ Bit 5 đến 7 và 1 đến 3: Những Bit này sử dụng cho những Mô đun khác.

+ **NMIIFG**: Bit 4, chờ ngắt NMI.

0: Không cho phép ngắt

1: Cho phép ngắt

+ **WDTIFG**: Bit 0, chờ ngắt watchdog timer. Trong chế độ giám sát thì WDTIFG được giữ cho đến khi được Reset bằng chương trình. Trong chế độ hẹn giờ thì WDTIFG được Reset tự động bởi chương trình ngắt hoặc cũng có thể bởi chương trình. Bởi vì IFG1 có thể sử dụng cho các mô đun khác nên khi xóa cờ WDTIFG ta nên sử dụng lệnh BIS.B hoặc BIC.B hơn là MOV.B hoặc CLR.B.

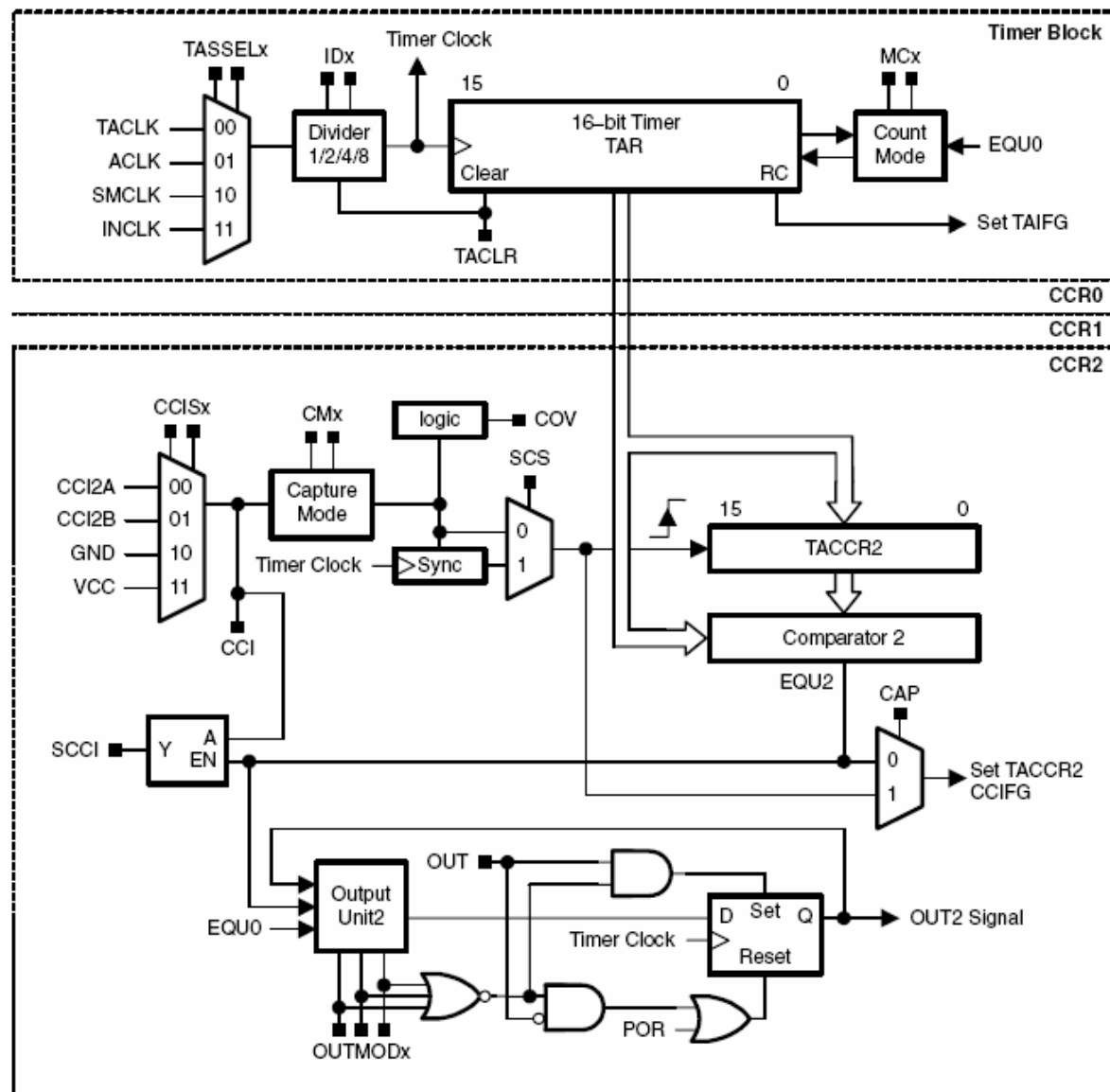
0: Không cho phép ngắt

1: Cho phép ngắt

VIII. TIMER A

VIII.1. Giới thiệu tổng quát Timer_A

Timer_A là một bộ đếm/định thời 16 Bit với 3 thanh ghi capture/compare. Timer_A có thể hỗ trợ tổ hợp capture/compare, những ngõ ra PWM, và sự xác định thời khoảng. Timer_A cũng có những ngắt bao quát. Ngắt có thể được sinh ra từ sự tràn bộ đếm. Timer_A bao gồm các khối chức năng được mô tả như sau:



Hình VIII.1: Biểu đồ khối Timer_A

VIII.2. Chế độ hoạt động của Timer_A

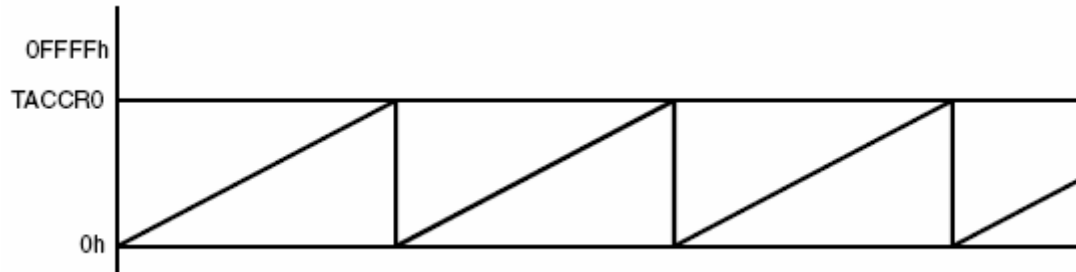
Timer_A có 4 chế độ hoạt động là Stop, Up, Continuous, Up/Down và được điều khiển bởi Bit MCx.

MCx	Chế độ hoạt động	Mô tả hoạt động
00	Stop	Timer_A tạm dừng
01	Up	Đếm lên từ 0x0000 đến giá trị TACCR0
10	Continuous	Tiếp tục chế độ đếm từ 0x0000 đến 0xFFFF
11	Up/Down	Đếm từ 0x0000 đến TACCR0 rồi quay về 0

Bảng VIII.1: Các chế độ hoạt động của Timer_A

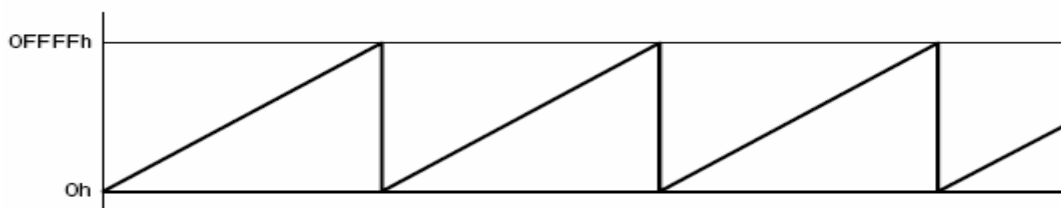
VIII.2.1. Up mode

- + TAR đếm từ 0x0000 lên đến giá trị TACCR0
- + TAR → TACCR0: Cờ ngắt thanh ghi TACCR0 là CCIFG thì được Set.
- + Khi TAR = TACCR0 thì EQU0 = 1 (khởi động lại việc đếm)
- + TACCR0 → 0 : Cờ ngắt TAIFG được Set.



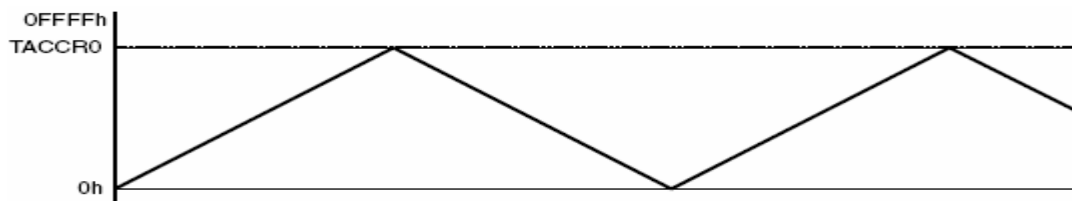
VIII.2.2. Continuous mode

- + TAR đếm lên đến giá trị 0xFFFF
- + Khi TAR = 0xFFFF thì TAR bắt đầu đếm lại từ 0.
- + Khi 0xFFFF → 0: Cờ ngắt TAIFG được Set.



VIII.2.3. Up/Down mode

- + Đếm từ 0x0000 đến TACCR0 rồi quay về 0
- + TACCR0 – 1 → TACCR0: Cờ ngắt CCIFG được Set.
- + Khi TAR = TACCR0 thì việc đếm được đảo ngược.
- + Khi 0x0001 → 0x0000: Cờ ngắt TAIFG được Set



VIII.2.4. Các phương pháp Reset Timer_A

- + Ghi 0 vào thanh ghi TAR
- + Ghi 0 vào thanh ghi TACCR0
- + Thiết lập Bit TACLR trong thanh ghi điều khiển Timer (TACTL)

VIII.2.5. Chế độ Capture

Được sử dụng cho sự đo lường chu kỳ thời gian của các sự kiện với sự can thiệp rất nhỏ của CPU

- + Set Bit CAP để lựa chọn chế độ này.
- + Set Bit SCS để đồng bộ Capture với khối Timer kế tiếp.
- + Tín hiệu vào được lấy mẫu bởi CCIxA, được lựa chọn bởi Bit CCISx trong thanh ghi TACCTLx.
- + Thiết lập Bit CMx để lựa chọn các chế độ Capture
- + Khi một xung hợp lệ được phát hiện trên ngõ vào thì giá trị trong TAR được chốt trong thanh ghi TACCRx cung cấp một mốc thời gian cho sự kiện.
- + Cờ ngắt CCIFG được Set.
- + Bit COV = 1 điều khiển sự tràn sự kiện khi một capture thứ 2 được sử dụng, trước khi giá trị từ capture đầu tiên được đọc.

VIII.2.6. Chế độ Compare

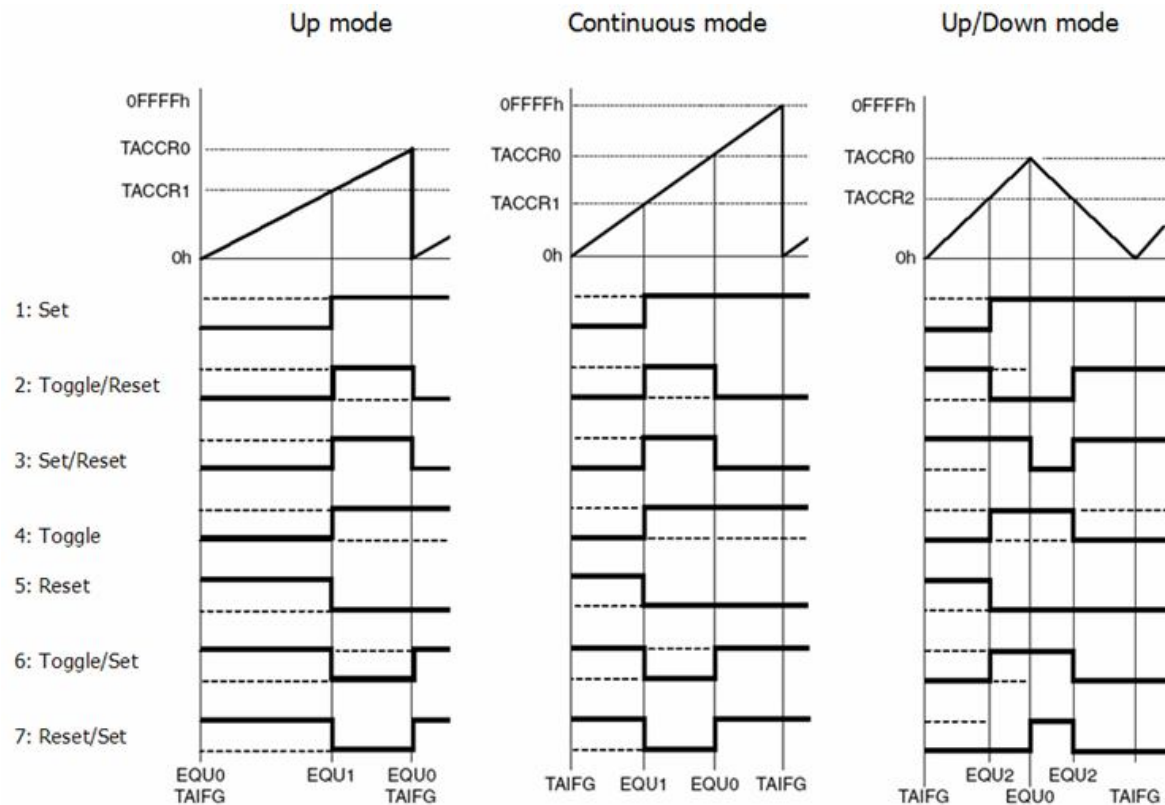
Sử dụng để tạo ra những xung ngõ ra và điều chỉnh độ rộng xung ngõ ra.

- + Reset Bit CAP để lựa chọn chế độ Compare
- + TAR đếm lên đến giá trị được lập trình trong thanh ghi TACCRx.
- + Khi giá trị Timer bằng với giá trị trong thanh ghi TACCRx thì một ngắt được sinh ra. Cờ ngắt CCIFG thì Set, EQUx = 1.
- + EQUx có tác dụng so sánh tín hiệu ngõ ra OUTx phù hợp với OUTMODx.
- + Tín hiệu vào CCI được chốt trong SCCI.

VIII.2.7. Hoạt động ngõ ra

OUTMODx	Chế độ	Mô tả hoạt động
000	Out	Tín hiệu ngõ ra được xác định bởi Bit OUTx
001	Set	OUTx = 1 \Rightarrow timer = TACCRx OUTx = 0 \Rightarrow timer = 0
010	Toggle/Reset	OUTx = toggle \Rightarrow timer = TACCRx OUTx = 0 \Rightarrow timer = TACCR0
011	Set/Reset	OUTx = 1 \Rightarrow timer = TACCRx OUTx = 0 \Rightarrow timer = TACCR0
100	Toggle	OUTx = toggle \Rightarrow timer = TACCRx Chu kỳ ngõ ra gấp đôi chu kỳ Timer
101	Reset	OUTx = 0 \Rightarrow timer = TACCRx OUTx = 1 \Rightarrow Một chế độ ngõ ra khác được lựa chọn và tác động đến ngõ ra.
110	Toggle/Set	OUTx = toggle \Rightarrow timer = TACCRx OUTx = 1 \Rightarrow timer = TACCR0
111	Reset/Set	OUTx = 0 \Rightarrow timer = TACCRx OUTx = 1 \Rightarrow timer = TACCR0

Bảng VIII.2: Các hoạt động ngõ ra Timer_A



Hình VIII.1: Những ngã ra mẫu

VIII.3. Các thanh ghi Timer_A

Các thanh ghi của Timer_A được trình bày dưới bảng sau:

Register	Short Form	Register Type	Address	Initial State
Timer_A control	TACTL	Read/write	0160h	Reset with POR
Timer_A counter	TAR	Read/write	0170h	Reset with POR
Timer_A capture/compare control 0	TACCTL0	Read/write	0162h	Reset with POR
Timer_A capture/compare 0	TACCR0	Read/write	0172h	Reset with POR
Timer_A capture/compare control 1	TACCTL1	Read/write	0164h	Reset with POR
Timer_A capture/compare 1	TACCR1	Read/write	0174h	Reset with POR
Timer_A capture/compare control 2	TACCTL2†	Read/write	0166h	Reset with POR
Timer_A capture/compare 2	TACCR2†	Read/write	0176h	Reset with POR
Timer_A interrupt vector	TAIV	Read only	012Eh	Reset with POR

Bảng VIII.2: Các thanh ghi Timer_A

VIII.3.1. TACTL, Timer_A Control

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx	MCx		Unused	TACLRL	TAIE	TAIFG	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	w-(0)	rw-(0)	rw-(0)

+ **TASSELx**: Lựa chọn nguồn xung Clock Timer_A

00 TACLK

01 ACLK

10 SMCLK

11 INCLK

+ **IDx**: Bộ chia tín hiệu vào

00 /1

01 /2

10 /4

11 /8

+ **MCx**: Bit điều khiển

00 STOP

01 UP

10 CONTINUOUS

11 UP/DOWN

+ **TACLR**: Xóa Timer_A. Set Bit này sẽ Reset TAR, IDx.+ **TAIE**: Cho phép ngắt Timer_A

0 Không cho phép ngắt

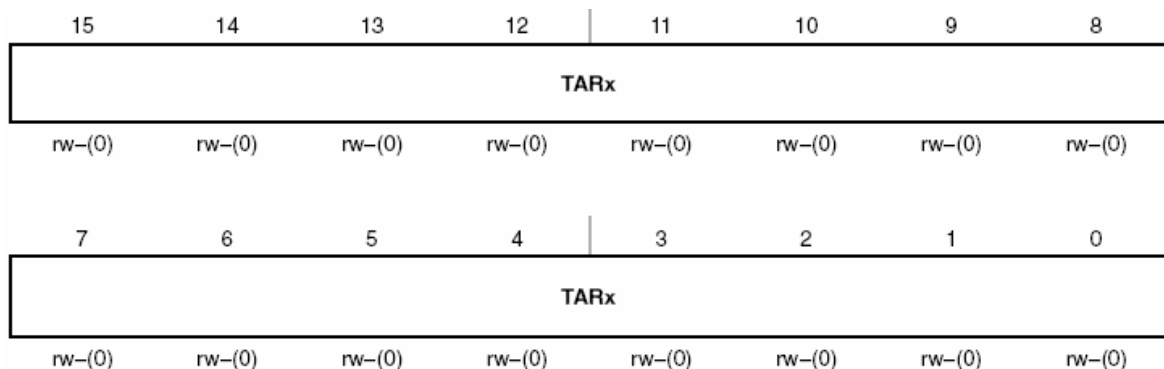
1 Cho phép ngắt

+ **TAIFG**: Cờ ngắt Timer_A

0 Không có cờ hoãn ngắt

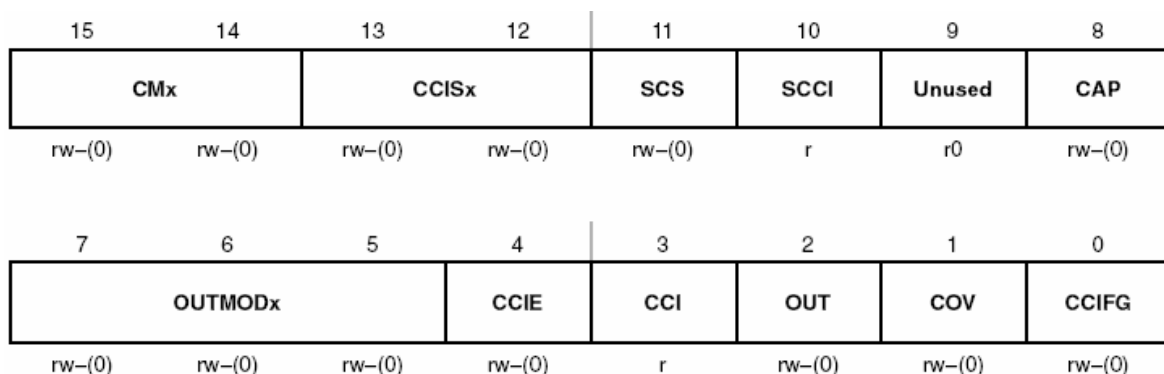
1 Có cờ hoãn ngắt

VIII.3.2. TAR, Timer_A Counter



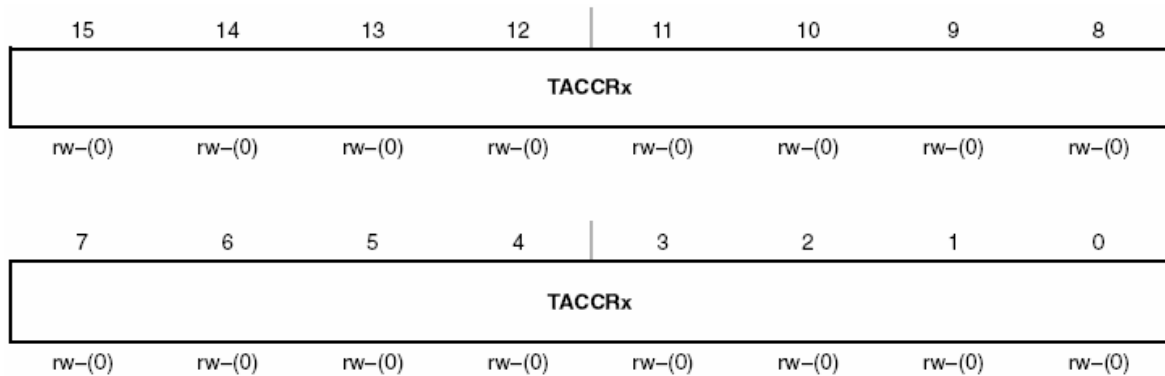
TARx: Thanh ghi đếm của Timer_A

VIII.3.3. TACCTLx, Capture/compare control



- + **CMx**: Chế độ Capture
 - 00 Không Capture
 - 01 Capture trên cạnh lên
 - 10 Capture trên cạnh xuống
 - 11 Capture trên cả hai cạnh lên và cạnh xuống
- + **CCISx**: Lựa chọn ngõ vào Capture/Compare
 - 00 CCIxA
 - 01 CCIxB
 - 10 GND
 - 11 Vcc
- + **SCS**: Đồng bộ nguồn Capture
 - 0 Không đồng bộ
 - 1 Đồng bộ
- + **SCCI**: Đồng bộ ngõ vào Capture/Compare. Lựa chọn tín hiệu vào CCI thì được chốt với tín hiệu EQUx và có thể đọc thông qua Bit SCCI
- + **CAP**: Chọn chế độ
 - 0 Compare
 - 1 Capture
- + **OUTMODx**: Chọn chế độ ngõ ra
 - 000 Out
 - 001 Set
 - 010 Toggle/Reset
 - 011 Set/Reset
 - 100 Toggle
 - 101 Reset
 - 110 Toggle/Set
 - 111 Reset/set
- + **CCIE**: Cho phép ngắt Capture/Compare.
 - 0 Không cho phép ngắt
 - 1 Cho phép ngắt
- + **CCI**: Ngõ vào Capture/Compare có thể đọc bởi Bit này.
- + **OUT**: Ngõ ra
 - 0 Mức thấp
 - 1 Mức cao
- + **COV**: Sự tràn Capture
 - 0 Không xuất hiện tràn Capture
 - 1 Xuất hiện tràn Capture
- + **CCIFG**: Cờ ngắt Capture/Compare
 - 0 Không có trì hoãn ngắt
 - 1 Có trì hoãn ngắt

VIII.3.4. TACCRx, Timer_A capture/compare

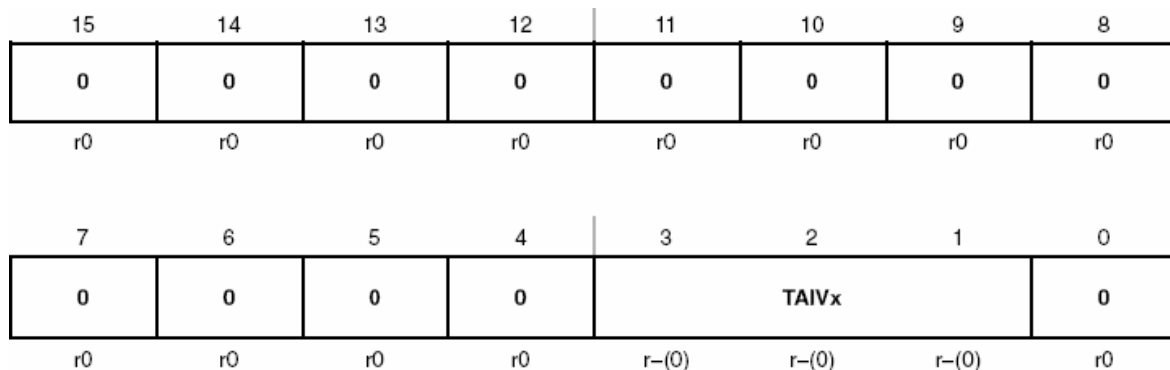


TACCRx: Đây là thanh ghi Timer_A Capture/Compare.

Chế độ Capture: TAR được copy vào thanh ghi TACCRx khi Capture được sử dụng.

Chế độ Compare: Thanh ghi TACCRx chứa dữ liệu của sự so sánh giá trị Timer trong TAR.

VIII.3.5. TAIV, Timer_A Interrupt vector



TAIVx: Giá trị Véc tơ ngắt Timer_A

- 00h Không có trì hoãn ngắt
- 02h Cờ TACCR1 CCIFG
- 04h Cờ TACCR2 CCIFG
- 06h Được dự trữ
- 08h Được dự trữ
- 0Ah Cờ TAIFG
- 0Ch Được dự trữ
- 0Eh Được dự trữ

IX. BỘ BIẾN ĐỔI ADC10

IX.1. Giới thiệu về ADC10

ADC10 là một bộ biến đổi A-D 10 Bit có hiệu suất cao. Mô đun ADC10 lắp đặt 10 bit SAR core, Bộ điều khiển lựa chọn mẫu, Bộ sinh mẫu, Bộ điều khiển chuyển đổi dữ liệu (DTC).

DTC cung cấp cho ADC10 những mẫu được chuyển đổi và lưu trữ ở bất kỳ nơi đâu trong bộ nhớ mà không có bất kỳ sự can thiệp nào của CPU. Mô đun ADC10 có thể được thiết lập bởi người sử dụng cho những ứng dụng đa dạng.

ADC10 bao gồm:

- + Tỷ lệ chuyển đổi lớn nhất hơn 200 ksp/s.
- + Lấy mẫu và giữ với chu kỳ lấy mẫu có thể lập trình được.
- + Sự chuyển đổi được khởi tạo bởi chương trình hoặc timer A.
- + Chương trình lựa chọn điện áp chuẩn trên chip (1.5V hoặc 2.5V).
- + Chương trình lựa chọn sự tham chiếu bên trong hoặc bên ngoài.
- + Có 8 kênh nhập tín hiệu từ bên ngoài (12 đối với MSP430x22xx).
- + Có kênh chuyển đổi cho cảm biến nhiệt độ bên ngoài, Vcc, tham chiếu bên ngoài.
- + Lựa chọn nguồn xung clock chuyển đổi.
- + Nhân ADC và điện áp chuẩn được cấp xuống một cách riêng rẽ.
- + Bộ chuyển đổi dữ liệu lưu trữ kết quả chuyển đổi một cách tự động.

IX.2. Hoạt động của ADC10

IX.2.1. Nhân ADC10 Bit

Nhân ADC10 chuyển đổi một tín hiệu vào tương tự sang số 10 Bit và lưu kết quả chuyển đổi trong thanh ghi ADC10MEM. Nhân ADC10 sử dụng chương trình lựa chọn mức điện thế (V_{R+} và V_{R-}) để xác định giới hạn trên và giới hạn dưới của sự chuyển đổi. Ngã ra số (N_{ADC}) có kích thước đầy đủ (03FFh) khi tín hiệu vào bằng hoặc nhỏ hơn V_{R+} và bằng 0 khi tín hiệu vào bằng hoặc nhỏ hơn V_{R-} . Kênh tín hiệu vào và mức điện áp chuẩn (V_{R+} và V_{R-}) thì được xác định bởi bộ nhớ điều khiển chuyển đổi.

Kết quả chuyển đổi ADC sử dụng định dạng nhị phân là:

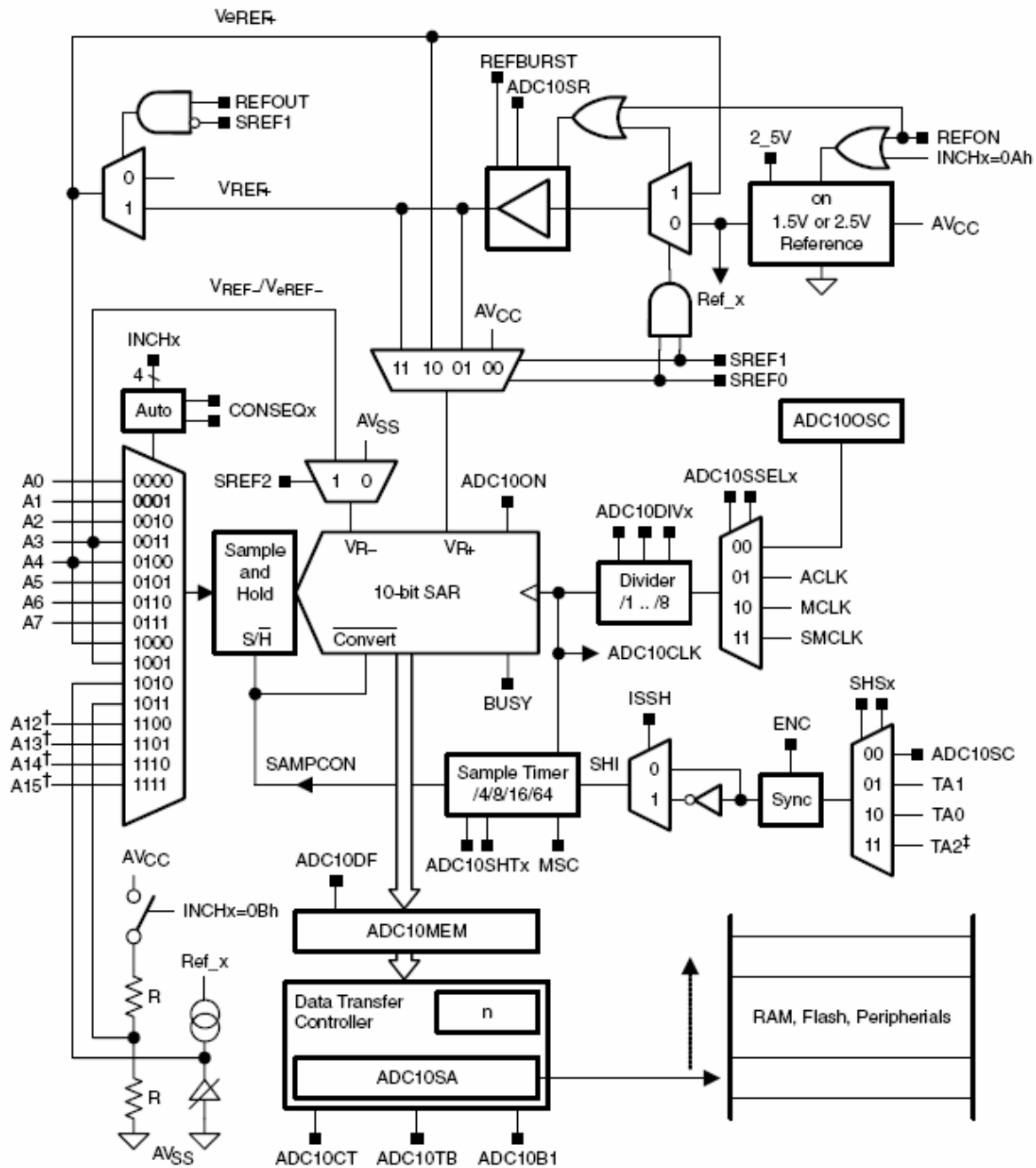
$$N_{ADC} = 1023 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

Nhân ADC10 được thiết lập bởi 2 thanh ghi điều khiển là ADC10CTL0 và ADC10CTL1. Nhân ADC10 được cho phép bởi Bit ADC10ON. Có một vài ngoại lệ là Bit điều khiển ADC10 có thể bị biến đổi khi ENC = 0. ENC phải được Set lên 1 trước khi có bất kỳ sự chuyển đổi nào có thể xảy ra.

Sự chọn lọc xung clock chuyển đổi:

ADC10CLK được sử dụng như là xung clock chuyển đổi và bộ sinh chu kỳ lấy mẫu. Nguồn xung clock của ADC10 được chọn bởi việc sử dụng Bit ADC10SSELx và có thể được chia từ 1-8 sử dụng Bit ADC10DIVx. Có thể nguồn của ADC10CLK là SMCLK, MCLK, ACLK và một bộ dao động bên trong là ADC10OSC.

Người sử dụng phải đảm bảo rằng nguồn xung Clock được chọn cho ADC10CLK phải hoạt động cho đến khi sự chuyển đổi kết thúc. Khi nguồn cấp xung bất chợt bị mất khi đang trong quá trình chuyển đổi thì sự chuyển đổi sẽ không hoàn thành và kết quả thì sẽ bị lỗi.

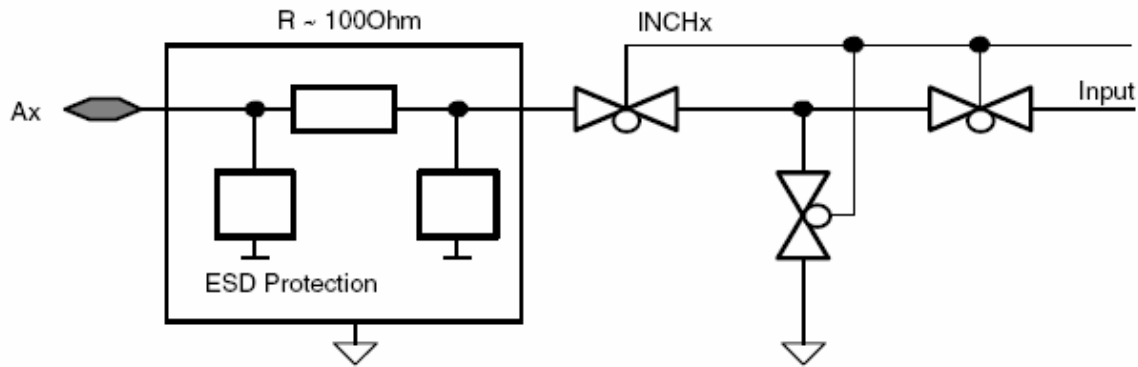


Hình IX.1: Sơ đồ khối chức năng ADC10

IX.2.2. Tín hiệu vào ADC10 và bộ ghép kênh

Có 8 tín hiệu bên trong và 4 tín hiệu bên ngoài tương tự được chọn như là một kênh của sự chuyển đổi bởi bộ ghép kênh tín hiệu vào tương tự. Bộ ghép kênh tín hiệu vào là một kiểu ngắt rồi đóng để làm giảm tín hiệu nhiễu từ những kênh chuyển. Bộ ghép kênh tín hiệu cũng là một T-Switch làm tối thiểu hóa các ghép nối giữa các kênh. Những kênh không được chọn đó thì bị cô lập từ A/D và điểm nối trung gian thì được nối mass để phân tán điện dung nhằm tránh sự giao tiếp chéo.

ADC10 sử dụng phép phân bố điện tích. Khi tín hiệu vào được chuyển mạch bên trong, sự chuyển tiếp này có thể là nguyên nhân của sự quá độ trên tín hiệu vào.



Hình IX.2: Bộ dồn kênh tương tự

Sự lựa chọn cổng tương tự:

Những tín hiệu vào bên ngoài của ADC10 là A_x , V_{REF+} và V_{REF-} thì dùng chung đầu nối với các Port I/O mục đích tổng quát, đó là những cổng CMOS kỹ thuật số. Khi những tín hiệu tương tự được đặt vào các cổng CMOS, dòng nhiễu đi từ Vcc đến GND. Dòng nhiễu này xuất hiện nếu điện thế vào có giá trị gần mức chuyển tiếp của cổng. Bit ADC10AEx cung cấp một khả năng khóa các bộ đệm đầu vào và bộ đệm đầu ra.

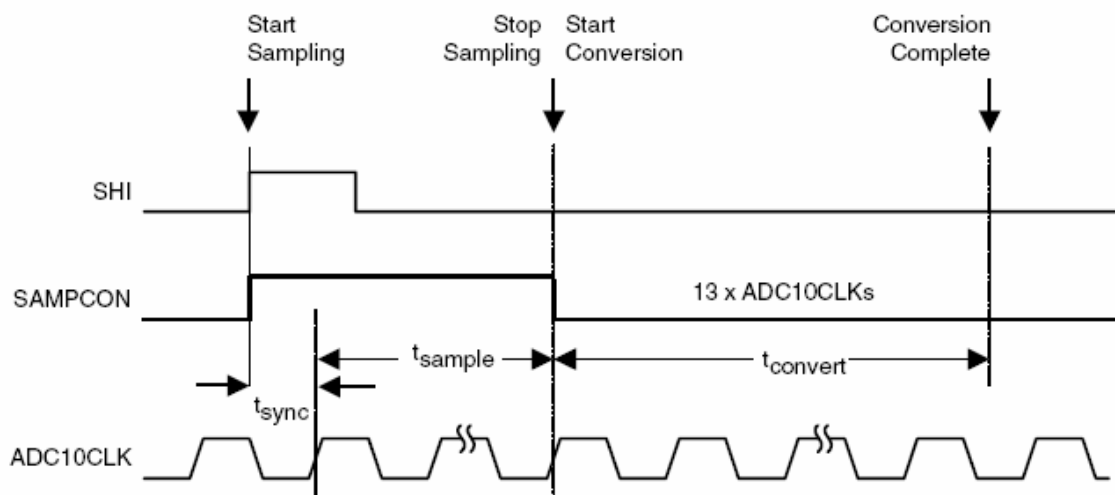
; P2.3 on MSP430x22xx device configured for analog input

BIS.B #08h, &ADC10AE0 ; P2.3 ADC10 function and enable

IX.2.3. Bộ sinh điện áp qui chiếu

ADC10 tích hợp điện áp qui chiếu bên trong với 2 sự lựa chọn mức điện áp. Thiết lập REFON = 1 thì cho phép qui chiếu bên trong. Khi REF2_5 = 1 thì qui chiếu bên trong là 2.5V. Khi REF2_5 = 0 thì qui chiếu bên trong là 1.5V.

Điện áp qui chiếu bên ngoài có thể cung cấp cho V_{R+} và V_{R-} tương ứng ở các chân A4 và A3. Khi sử dụng điện áp bên ngoài hoặc sử dụng Vcc để qui chiếu thì những qui chiếu bên trong có thể được tắt đi để bảo vệ nguồn.

IX.2.4. Quá trình biến đổi và lấy mẫu

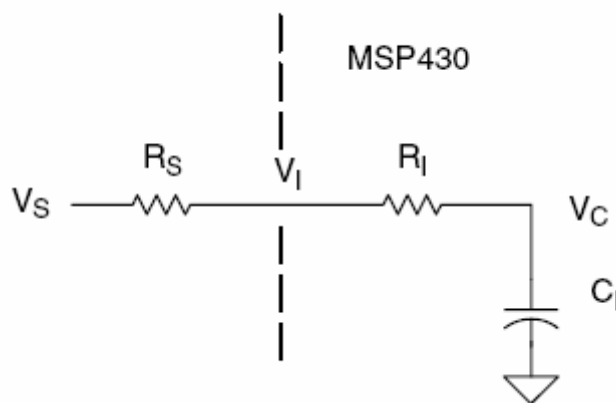
Hình IX.3: Quá trình lấy mẫu

Biến đổi A-D được bắt đầu khi có xung cạnh lên của tín hiệu vào SHI. SHI được lựa chọn bởi bit SHSx từ các nguồn:

- ADC10SC
- Timer_A 1
- Timer_A 0
- Timer_A 2

Bit ISSH dùng để đảo chiều tín hiệu SHI. Bit SHTx dùng lựa chọn chu kỳ lấy mẫu là 4, 8, 16, hoặc 64 ADC10CLK. Định thời lấy mẫu thiết lập SAMPCON ở mức cao cho việc lựa chọn chu kỳ lấy mẫu sau khi đồng bộ với ADC10CLK. T_{sample} là thời gian lấy mẫu. Khi SAMPCON từ mức cao xuống thấp thì quá trình biến đổi A-D bắt đầu.

Khi SAMPCON = 0 thì tất cả các ngõ vào Ax có tổng trở cao. Khi SAMPCON = 1 thì ngõ vào Ax giống như một bộ lọc thông thấp RC trong suốt thời gian lấy mẫu.



V_I : điện áp vào chân Ax

V_S : điện áp nguồn bên ngoài

R_S : điện trở ngoài

C_I : điện dung ngõ vào

V_C : điện áp tụ điện

R_I : điện trở trong

R_S và R_I ảnh hưởng đến thời gian lấy mẫu t_{samlpe} :

$$t_{\text{samlpe}} > (R_S + R_I) \cdot \ln(2^{11}) \cdot C_I$$

Trong thực tế thì thời gian lấy mẫu lớn hơn thời gian lấy mẫu tính toán và thời gian bộ đệm ổn định t_{REFBURST} .

$$t_{\text{samlpe}} > t_{\text{REFBURST}}$$

$$\text{với } t_{\text{REFBURST}} = SR \cdot V_{\text{Ref}} - 0.5\text{ms}$$

SR: tốc độ quét bộ đệm ($1\mu\text{s/V}$ khi ADC10SR = 0 và $2\mu\text{s/V}$ khi ADC10SR = 1)

V_{ref} : Điện áp tham chiếu ngoài

IX.2.5. Các chế độ biến đổi

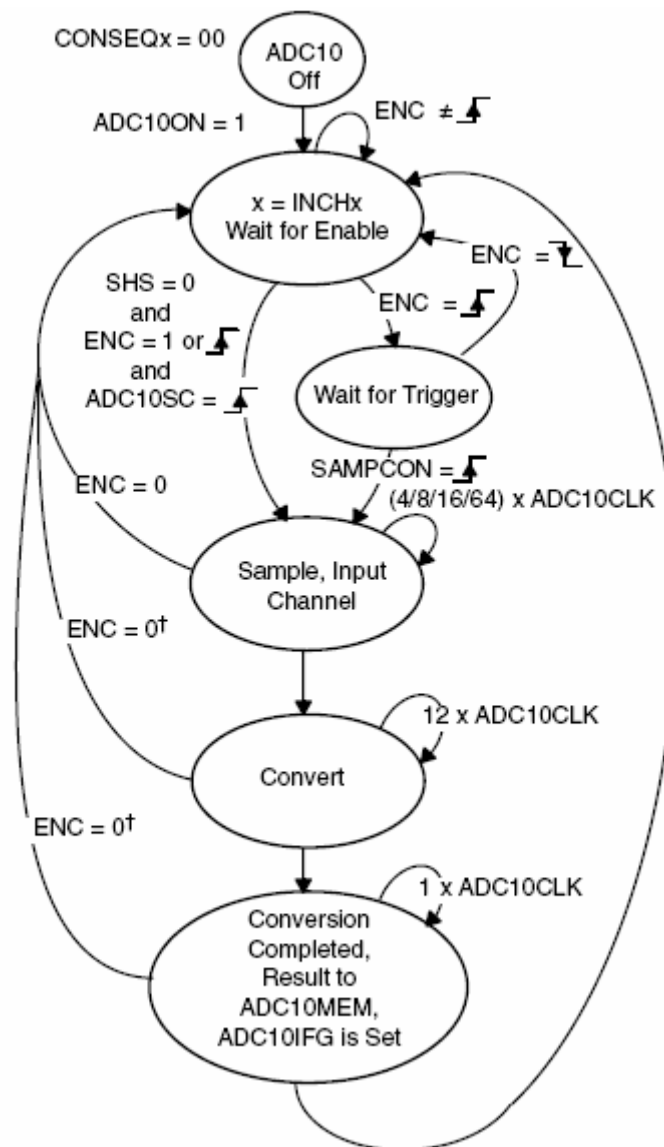
ADC10 có 4 chế độ hoạt động và được lựa chọn bởi Bit CONSEQx.

CONSEQx	Chế độ biến đổi	Hoạt động
00	Single channel single	biến đổi từng kênh
01	Sequence-of-channels	biến đổi một dãy kênh
10	Repeat single channel	biến đổi từng kênh được lặp lại
11	Repeat sequence-of-channels	biến đổi dãy kênh được lặp lại

Bảng IX.1: Các chế độ biến đổi

IX.2.5.1. Single-channel-single

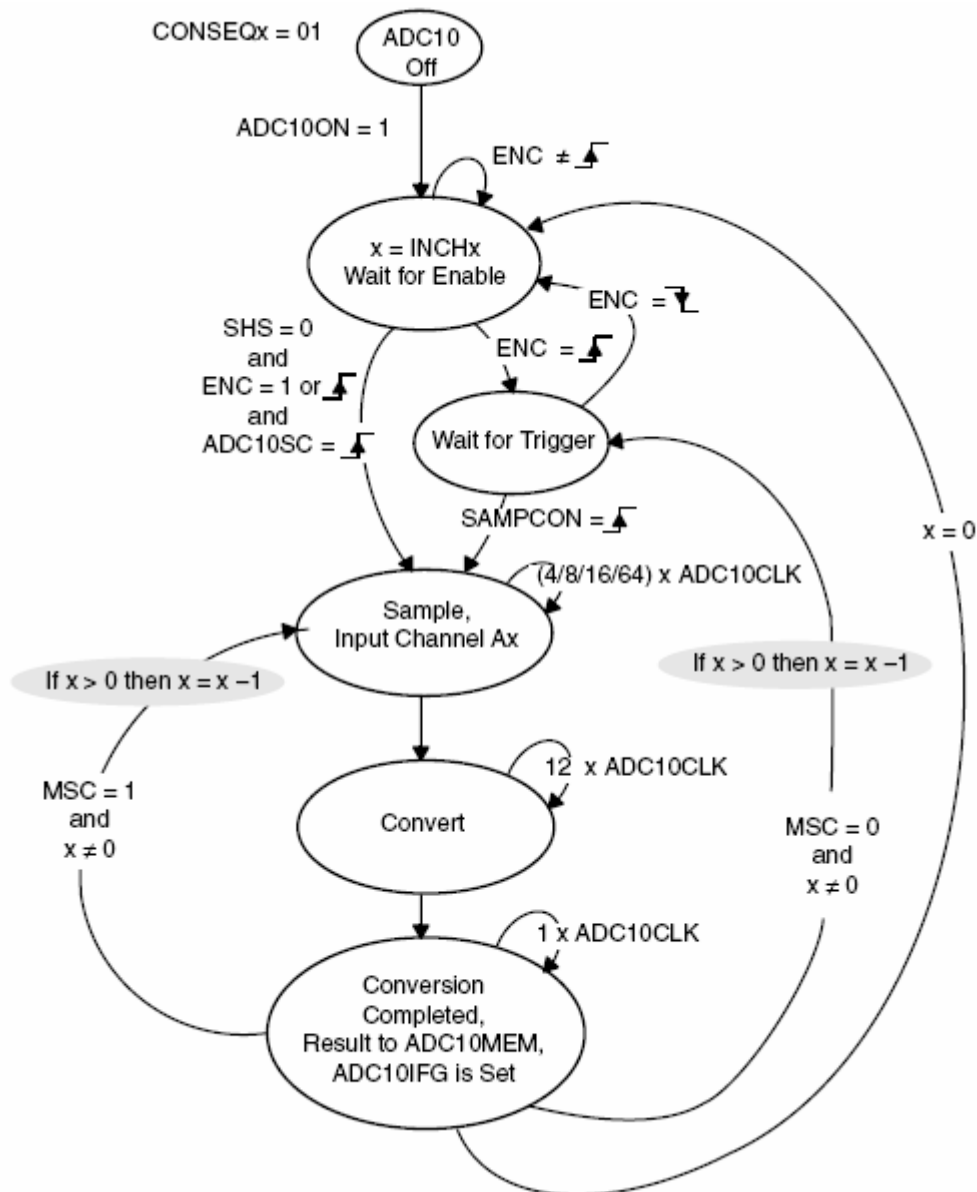
Mỗi kênh đơn được lựa chọn bởi INCHx và được biến đổi từng kênh một. Kết quả được lưu trong thanh ghi ADC10MEM. Bit ADC10SC dùng để kích hoạt sự biến đổi. ENC phải được đảo trạng thái sau mỗi lần biến đổi.



Hình IX.4: Lưu đồ biến đổi ở chế độ Single channel single

IX.2.5.2. sequence-of-channels

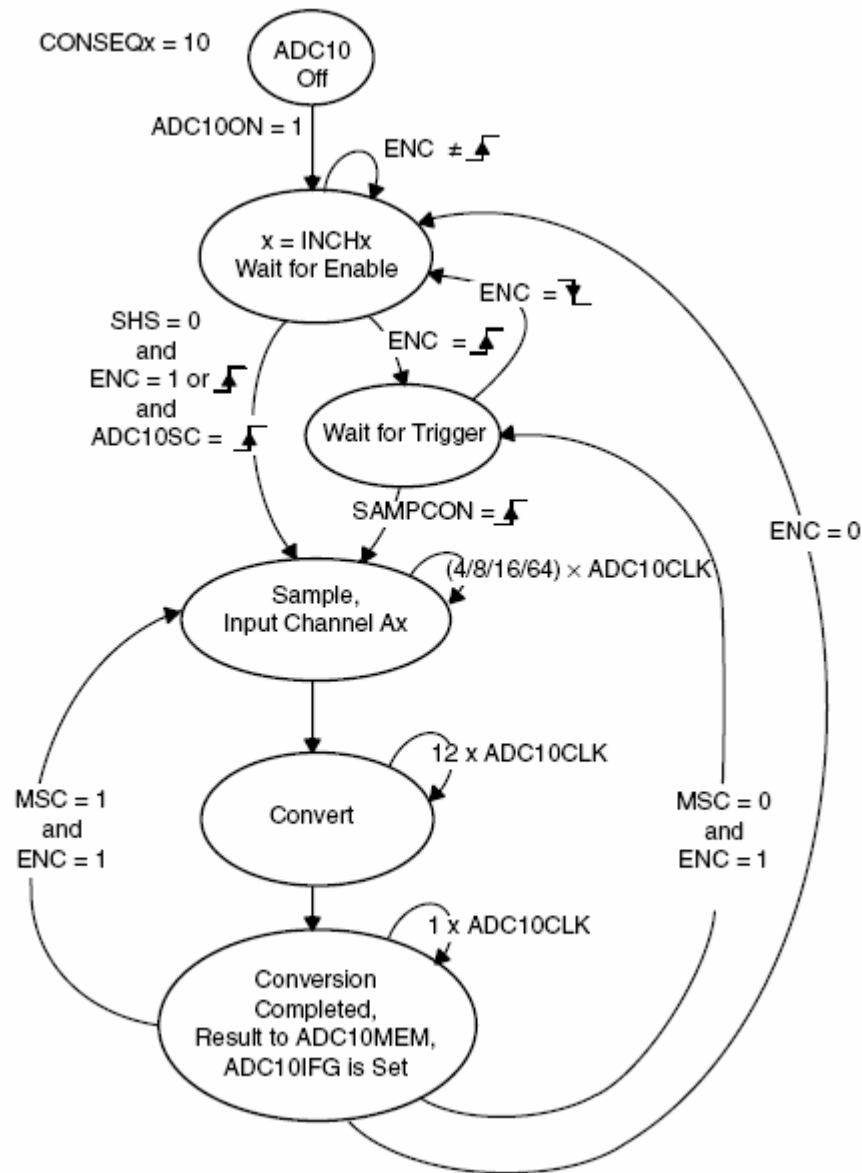
biến đổi một dãy các kênh và kênh bắt đầu được lựa chọn bởi Bit INCHx, kết thúc ở kênh A0. Kết quả được lưu trong thanh ghi ADC10MEM. Bit ADC10SC dùng để kích hoạt sự biến đổi. ENC phải được đảo trạng thái sau mỗi lần biến đổi.



Hình IX.5: Lưu đồ biến đổi ở chế độ Sequence-of-channels

IX.2.5.3. Repeat-single-channel

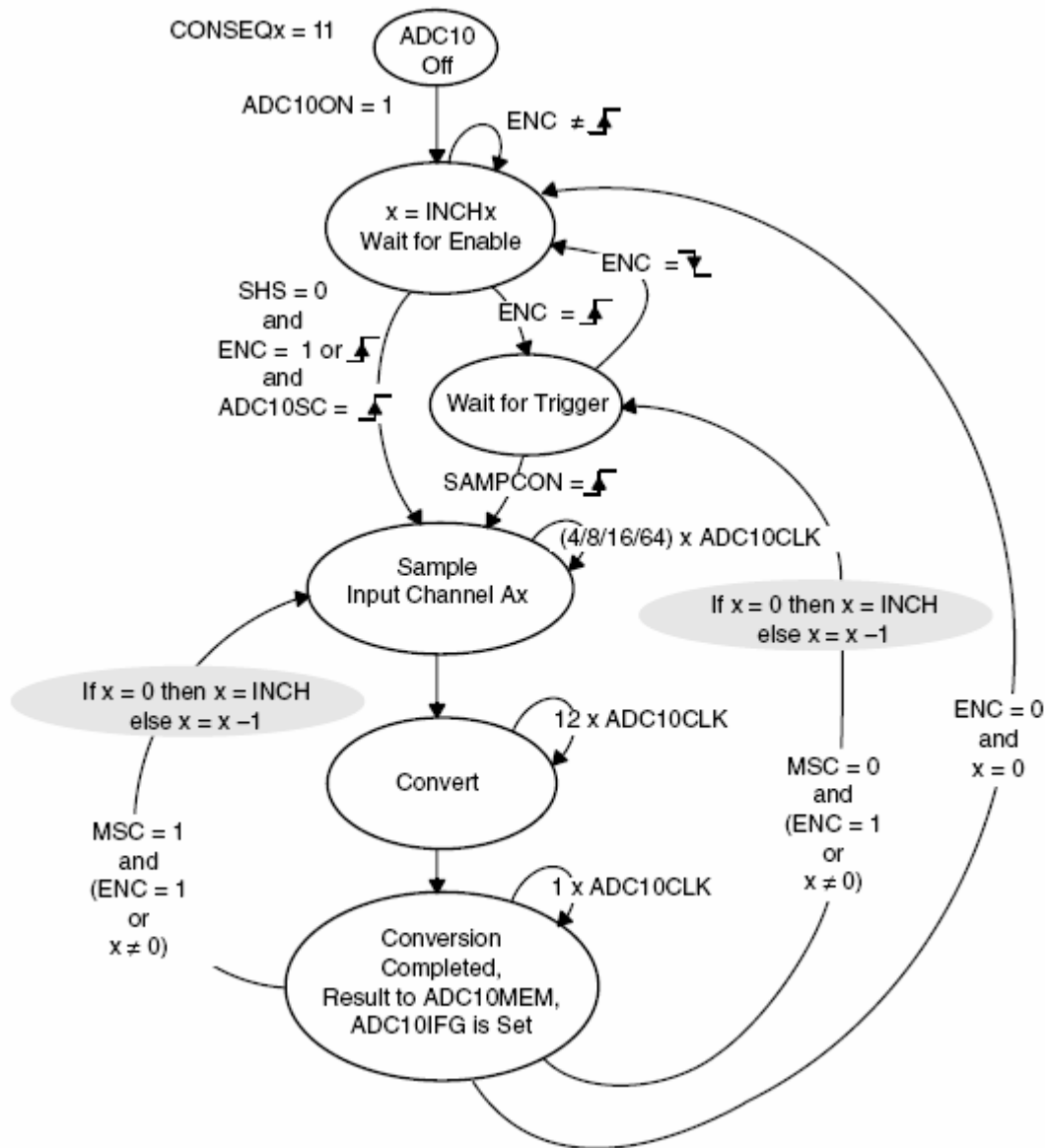
Mỗi kênh đơn được chọn bởi Bit INCHx được lấy mẫu và biến đổi tiếp tục. Kết quả ADC được lưu trong ADC10MEM.



Hình IX.6: Lưu đồ biến đổi ở chế độ Repeat single channel

IX.2.5.4. Repeat-sequence-of-channels

Một dãy các kênh được lấy mẫu và biến đổi trở lại. Kênh bắt đầu được lựa chọn bởi Bit INCHx, kết thúc ở kênh A0. Kết quả được lưu trong thanh ghi ADC10MEM. Dãy kết thúc ở kênh A0 lần kích hoạt kế tiếp sẽ bắt đầu lại dãy.



Hình IX.7: Lưu đồ biến đổi ở chế độ Repeat sequence-of-channels

IX.2.5.5. Sử dụng Bit MSC

Khi $MSC = 1$ và $CONSEQx > 0$ thì cạnh lên đầu tiên của SHI sẽ kích hoạt biến đổi. Sự biến đổi vẫn tiếp tục cho đến khi dãy kênh được biến đổi xong mặc dù trong quá trình biến đổi xuất hiện những xung cạnh lên ở tín hiệu kích hoạt SHI. Chức năng của Bit ENC không thay đổi khi sử dụng Bit MSC.

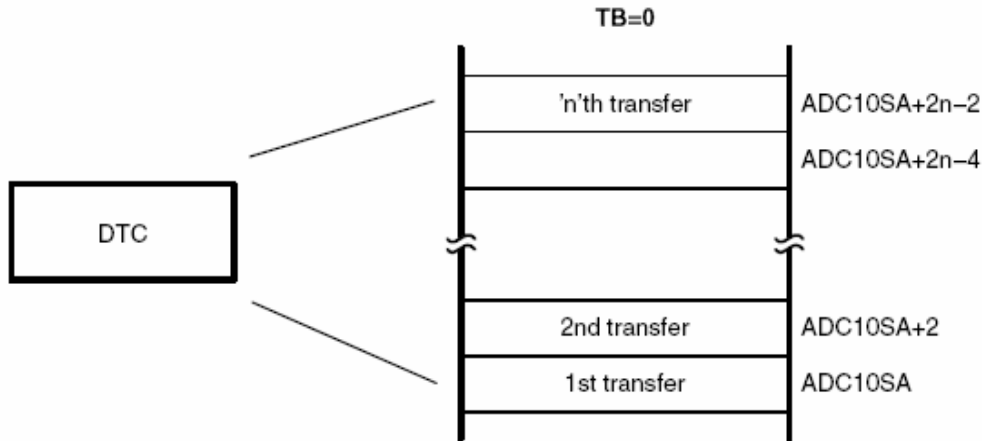
IX.2.6. Bộ điều khiển chuyển giao dữ liệu ADC10

ADC10 có một bộ điều khiển chuyển giao dữ liệu (DTC), nó sẽ tự động chuyển kết quả của biến đổi A-D từ thanh ghi ADC10MEM sang bộ nhớ của Chip. DTC được cho phép khi thiết lập thanh ghi ADC10DTC1 có giá trị khác 0.

Khi DTC được cho phép, mỗi khi ADC10 hoàn thành việc biến đổi và lưu kết quả vào thanh ghi ADC10MEM thì việc chuyển giao dữ liệu được kích hoạt. Mỗi DTC yêu cầu một CPU MCLK. Tránh việc sử dụng các Bus trong suốt quá trình chuyển giao dữ liệu, CPU được tạm ngừng. Sự chuyển giao không được kích hoạt trong khi ADC10 thì bận.

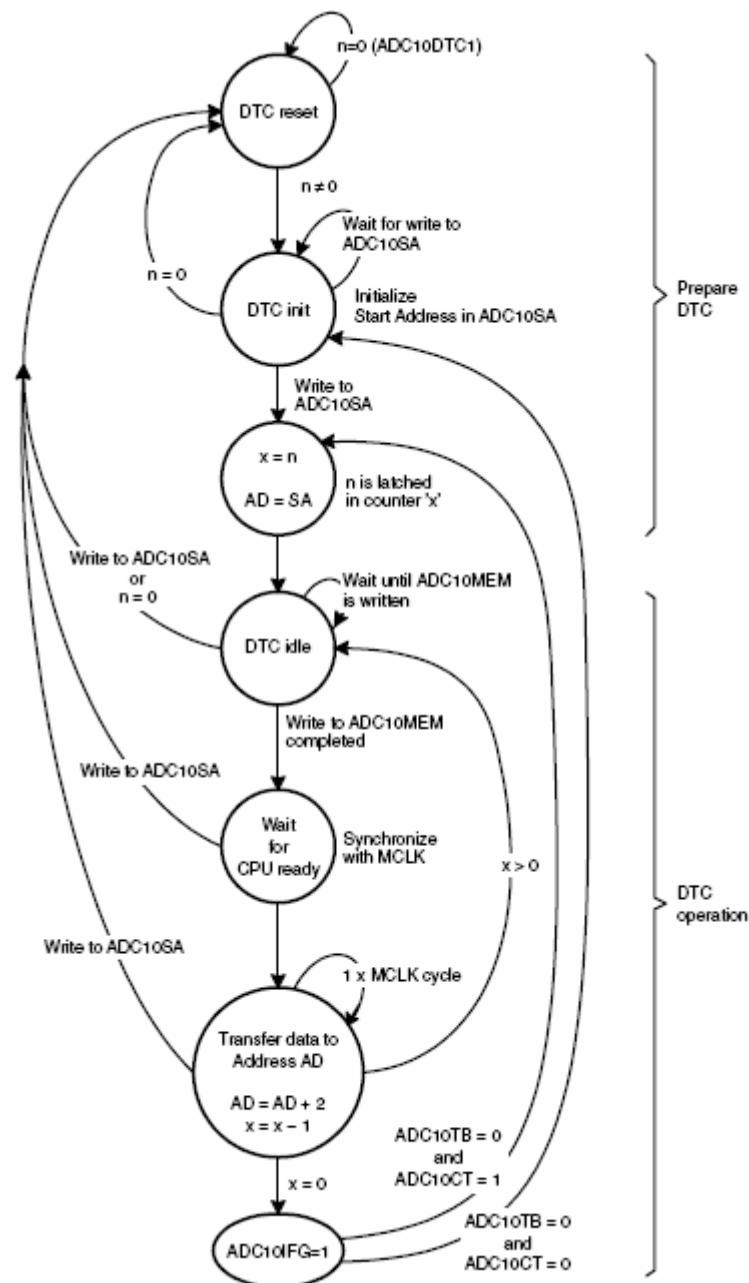
IX.2.6.1. Chế độ chuyển giao một khối

Được kích hoạt khi Bit ADC10TB Reset. Giá trị n trong ADC10TC1 xác định số lượng chuyển giao cho một khối. Địa chỉ bắt đầu của khối là thanh ghi 16 Bit ADC10SA. Địa chỉ kết thúc là $ADC10SA + 2n - 2$.



Hình IX.8: Chế độ chuyển giao một khối

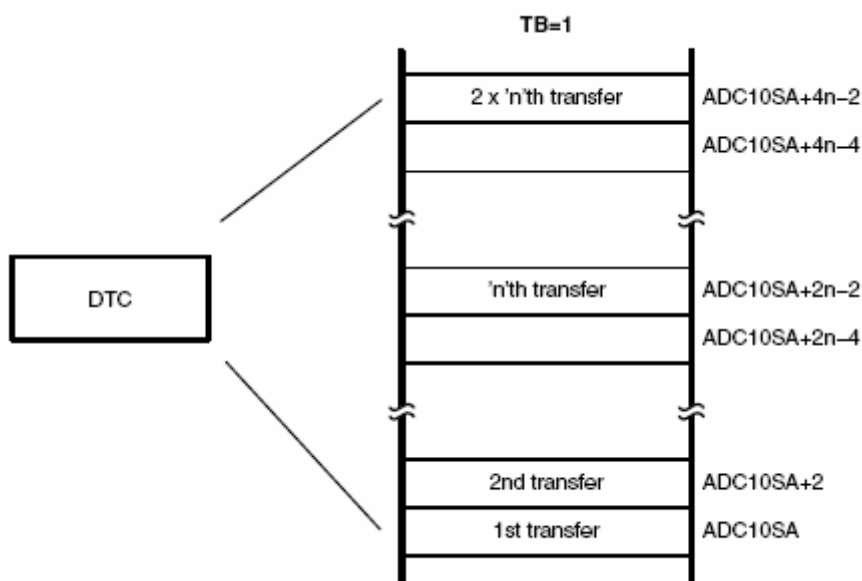
Địa chỉ con trỏ bên trong được khởi tạo bằng ADC10SA và bộ đếm chuyển giao được khởi tạo bằng ' n '. DTC chuyển giao giá trị từ ADC10MEM sang địa chỉ con trỏ ADC10SA. Sau mỗi lần chuyển giao thì địa chỉ con trỏ tăng 2 và bộ đếm giảm 1. DTC tiếp tục chuyển giao từ ADC10MEM cho đến khi bộ đếm giảm xuống bằng 0. Khi sử dụng DTC trong chế độ chuyển giao một khối thì cờ ADC10IFG được Set sau mỗi khối được chuyển giao hoàn thành.



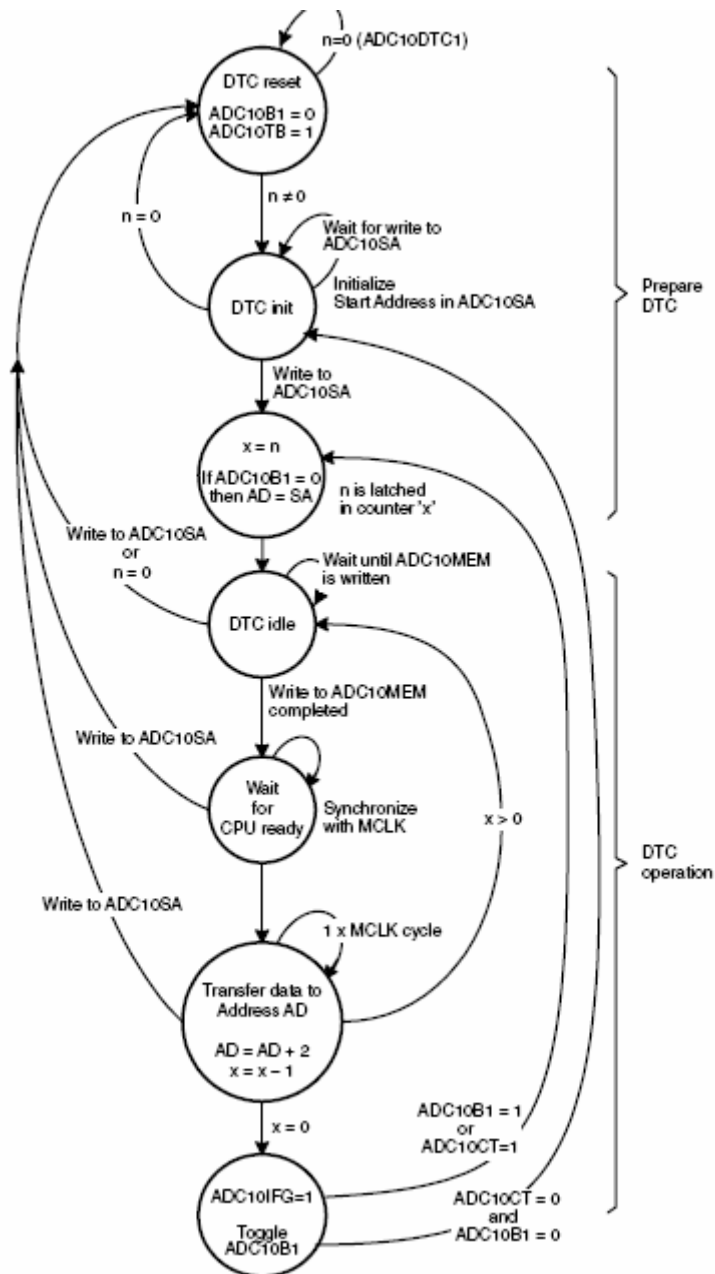
Hình IX.9: Sơ đồ chuyển giao dữ liệu trong chế độ chuyển giao một khối

IX.2.6.2. Chế độ chuyển giao 2 khối

Được chọn khi Bit ADC10TB được Set. Giá trị n trong ADC10TC1 xác định số chuyển giao trong một khối. Địa chỉ đầu tiên trong thanh ghi ADC10SA. Địa chỉ kết thúc của khối đầu tiên là $ADC10SA + 2n - 2$. Dãy địa chỉ của khối thứ 2 được xác định từ $SA + 2n$ đến $SA + 4n - 2$. Địa chỉ con trỏ bên trong được khởi tạo bằng ADC10SA và bộ đếm chuyển giao được khởi tạo bằng ' n '. DTC chuyển giao giá trị từ ADC10MEM sang địa chỉ con trỏ ADC10SA. Sau mỗi lần chuyển giao thì địa chỉ con trỏ tăng 2 và bộ đếm giảm 1. DTC tiếp tục chuyển giao từ ADC10MEM cho đến khi bộ đếm giảm xuống bằng 0. Ở thời điểm này, khối thứ nhất đầy, cờ ADC10IFG và Bit ADC10B1 thì được Set. DTC tiếp tục với khối 2. Bộ đếm chuyển giao bên trong tự động nạp lại giá trị n . DTC bắt đầu chuyển kết quả biến đổi từ ADC10MEM sang khối 2. Sau n lần chuyển giao hoàn thành thì khối 2 đầy. Cờ ADC10IFG Set, Bit ADC10B1 được clear.



Hình IX.10: Chế độ chuyển giao hai khối



Hình IX.11: Sơ đồ chuyển giao dữ liệu trong chế độ chuyển giao hai khối

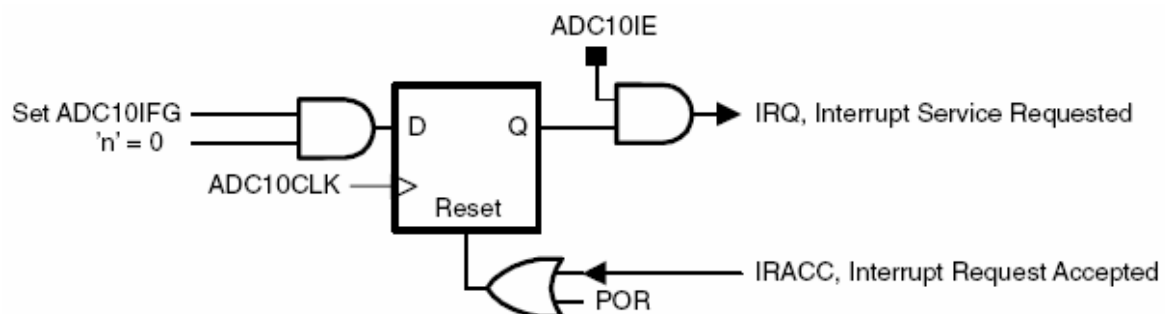
IX.2.6.3. Chu kỳ thời gian chuyển đổi DTC

CPU Operating Mode	Clock Source	Maximum DTC Cycle Time
Active mode	MCLK=DCOCLK	3 MCLK cycles
Active mode	MCLK=LFXT1CLK	3 MCLK cycles
Low-power mode LPM0/1	MCLK=DCOCLK	4 MCLK cycles
Low-power mode LPM3/4	MCLK=DCOCLK	4 MCLK cycles + 2 μ s [†]
Low-power mode LPM0/1	MCLK=LFXT1CLK	4 MCLK cycles
Low-power mode LPM3	MCLK=LFXT1CLK	4 MCLK cycles
Low-power mode LPM4	MCLK=LFXT1CLK	4 MCLK cycles + 2 μ s [†]

Bảng IX.2: Chu kỳ thời gian chuyển đổi DTC

IX.2.8. Ngắt ADC10

Một ngắt và một vec tơ ngắt được ghép với ADC10. Khi DTC không được sử dụng ($ADC10DTC1 = 0$), $ADC10IFG$ được Set khi kết quả biến đổi A-D được tải xuống $ADC10MEM$. Khi DTC được sử dụng ($ADC10DTC1 > 0$), $ADC10IFG$ được Set khi sự chuyển giao một khối được hoàn thành và bộ đếm chuyển giao bằng 0. Nếu cả hai Bit $ADC10IE$ và GIE được Set, $ADC10IFG$ sẽ sinh ra một ngắt yêu cầu. Cờ $ADC10IFG$ tự động Reset khi ngắt yêu cầu được duy trì hoặc có thể được Reset bằng phần mềm.



Hình IX.12: Hệ thống ngắt ADC10

IX.3. Các thanh ghi ADC10

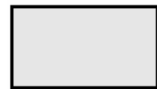
Các thanh ghi của ADC10 được trình bày ở bảng sau:

Register	Short Form	Register Type	Address	Initial State
ADC10 input enable register 0	ADC10AE0	Read/write	04Ah	Reset with POR
ADC10 input enable register 1	ADC10AE1	Read/write	04Bh	Reset with POR
ADC10 control register 0	ADC10CTL0	Read/write	01B0h	Reset with POR
ADC10 control register 1	ADC10CTL1	Read/write	01B2h	Reset with POR
ADC10 memory	ADC10MEM	Read	01B4h	Unchanged
ADC10 data transfer control register 0	ADC10DTC0	Read/write	048h	Reset with POR
ADC10 data transfer control register 1	ADC10DTC1	Read/write	049h	Reset with POR
ADC10 data transfer start address	ADC10SA	Read/write	01BCh	0200h with POR

Bảng IX.3: Các thanh ghi ADC10

IX.3.1. ADC10CTL0, ADC10 control register 0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)			rw-(0)		rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)



Modifiable only when ENC = 0

+ **SREFx**: Lựa chọn tham chiếu.

- 000 $V_{R+} = V_{CC}$ và $V_{R-} = V_{SS}$
- 001 $V_{R+} = V_{REF+}$ và $V_{R-} = V_{SS}$
- 010 $V_{R+} = V_{eREF+}$ và $V_{R-} = V_{SS}$
- 011 $V_{R+} = \text{Buffered } V_{eREF+}$ và $V_{R-} = V_{SS}$
- 100 $V_{R+} = V_{CC}$ và $V_{R-} = V_{REF}/V_{eERF-}$
- 101 $V_{R+} = V_{REF+}$ và $V_{R-} = V_{REF}/V_{eERF-}$
- 110 $V_{R+} = V_{eREF+}$ và $V_{R-} = V_{REF}/V_{eERF-}$
- 111 $V_{R+} = \text{Buffered } V_{eREF+}$ và $V_{R-} = V_{REF}/V_{eERF-}$

+ **ADC10SHTx**: Thời gian giữ và lấy mẫu

- 00 4 x ADC10CLKs
- 01 8 x ADC10CLKs
- 10 16 x ADC10CLKs
- 11 64 x ADC10CLKs

+ **ADC10SR**: Tốc độ lấy mẫu. Thiết lập ADC10SR làm giảm dòng tiêu thụ của bộ đệm qui chiếu.

- 0 Bộ đệm qui chiếu hỗ trợ trên 200 ksps

1 Bộ đệm qui chiếu hỗ trợ trên 50 kpsps

+ **REFOUT**: Ngã ra qui chiếu.

0 Ngã ra qui chiếu tắt

1 Ngã ra qui chiếu mở

+ **REFBURST**: Truyền từng khối qui chiếu

0 Bộ đệm qui chiếu liên tục

1 Bộ đệm qui chiếu chỉ trong quá trình lấy mẫu và biến đổi

+ **MSC**: Tổ hợp lấy mẫu và biến đổi.

0 Quá trình lấy mẫu yêu cầu một xung cạnh lên của SHI để kích hoạt mỗi lần lấy mẫu và biến đổi.

1 Xung cạnh lên đầu tiên của SHI sẽ kích hoạt bộ định thời lấy mẫu.

Nhưng ngoài ra, lấy mẫu và biến đổi được sử dụng một cách tự động đến khi chu kỳ biến đổi được hoàn thành.

REF2_5V: Bộ sinh điện áp qui chiếu. REFON phải được Set.

0 1.5V

1 2.5V

+ **REFON**: Điều khiển tham chiếu.

0 Tắt tham chiếu

1 Mở tham chiếu

+ **ADC10ON**: Điều khiển ADC10

0 Tắt ADC10

1 Bật ADC10

+ **ADC10IE**: Cho phép ngắt ADC10

0 Không cho phép ngắt

1 Cho phép ngắt

+ **ADC10IFG**: Cờ ngắt ADC10

0 Không ngắt

1 Ngắt

+ **ENC**: Cho phép biến đổi A-D

0 ADC10 không được cho phép

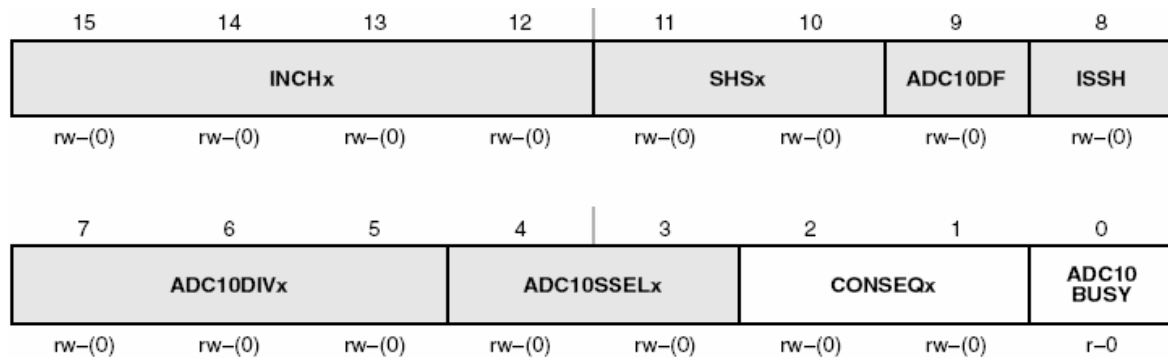
1 ADC10 được cho phép

+ **ADC10SC**: Bắt đầu biến đổi A-D.

0 Lấy mẫu và biến đổi chưa bắt đầu

1 Bắt đầu lấy mẫu và biến đổi

IX.3.2. ADC10CTL1, ADC10 control register 1



Modifiable only when ENC = 0

+ **INCHx**: Lựa chọn kênh ngõ vào.

- 0000 A0
- 0001 A1
- 0010 A2
- 0011 A3
- 0100 A4
- 0101 A5
- 0110 A6
- 0111 A7
- 1000 V_{eERF+}
- 1001 V_{ERF-}/V_{eERF-}
- 1010 Cảm biến nhiệt độ
- 1011 $(V_{cc} - V_{ss})/2$
- 1100 $(V_{cc} - V_{ss})/2$, A12 trên MSP430x22xx
- 1101 $(V_{cc} - V_{ss})/2$, A13 trên MSP430x22xx
- 1110 $(V_{cc} - V_{ss})/2$, A14 trên MSP430x22xx
- 1111 $(V_{cc} - V_{ss})/2$, A15 trên MSP430x22xx

+ **SHSx**: Lựa chọn nguồn thời gian giữ và lấy mẫu.

- 00 ADC10SC
- 01 Timer A.OUT1
- 10 Timer A.OUT0
- 11 Timer A.OUT2

+ **ADC10DF**: Định dạng dữ liệu ADC10

- 0 Nhị phân tiêu chuẩn
- 1 2s số bù

+ **ISH**: Đảo tín hiệu giữ và lấy mẫu

- 0 Không đảo tín hiệu vào lấy mẫu
- 1 Đảo tín hiệu vào lấy mẫu

+ ADC10DIVx: Bộ chia

000 /1
 001 /2
 010 /3
 011 /4
 100 /5
 101 /6
 110 /7
 111 /8

+ ADC10SSELx: Lựa chọn nguồn xung Clock ADC10

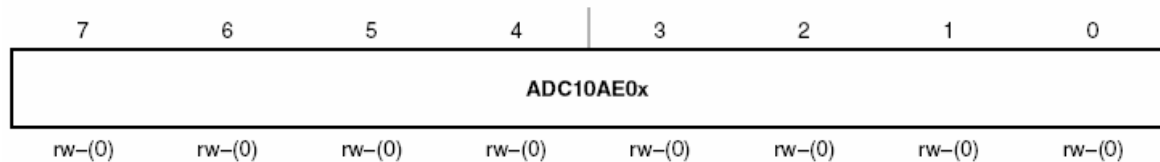
00 ADC10OSC
 01 ACLK
 10 MCLK
 11 SMCLK

+ CONSEQx: Lựa chọn chế độ biến đổi A-D

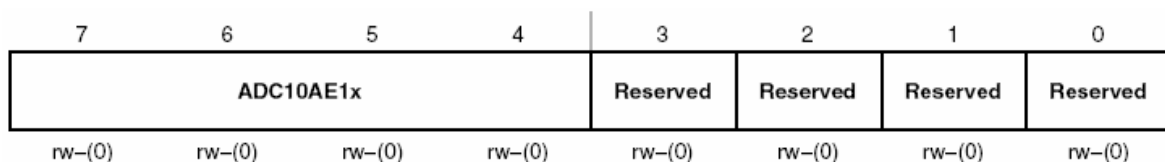
00 Single channel single
 01 Sequence-of-channels
 10 Repeat single channel
 11 Repeat sequence-of-channels

+ ADC10BUSY: ADC10 bận

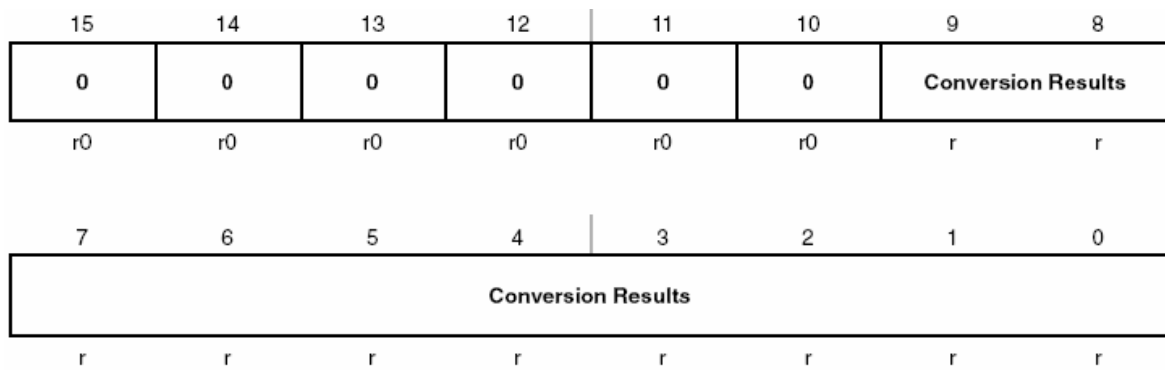
0 Không hoạt động
 1 Tham chiếu, lấy mẫu, hoặc biến đổi thì hoạt động

IX. 3.3. ADC10AE0, Analog enable control register 0**ADC10AE0x:**

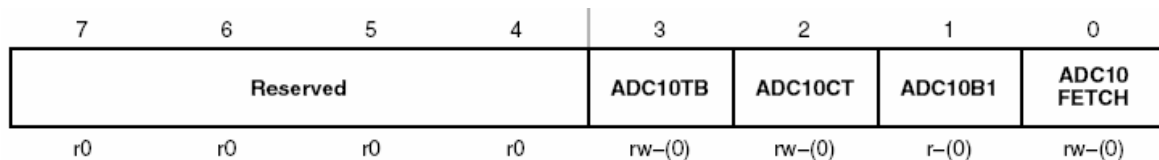
0 Không cho phép tín hiệu vào tương tự
 1 Cho phép tín hiệu vào tương tự

IX.3.4. ADC10AE1, Analog enable control register 1: (chỉ có ở MSP430x22xx)**ADC10AE1x:**

0 Không cho phép tín hiệu vào tương tự
 1 Cho phép tín hiệu vào tương tự

IX.3.5. ADC10MEM

Thanh ghi lưu trữ kết quả của biến đổi A-D và được định dạng nhị phân. Bit 10 đến 15 luôn là 0.

IX.3.6. ADC10DTC0

+ **ADC10TB**: Lựa chọn chế độ chuyển giao khối

- 0 chuyển giao 1 khối
- 1 chuyển giao 2 khối

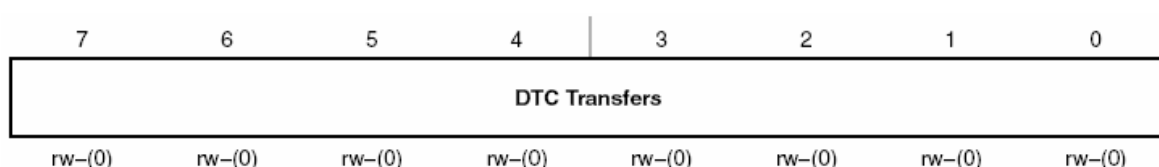
+ **ADC10CT**:

- 0 Dừng chuyển giao dữ liệu khi một khối hoặc 2 khối được hoàn thành
- 1 Dữ liệu được chuyển liên tục. DTC chỉ được dừng nếu ADC10CT được clear hoặc ADC10SA được ghi.

+ **ADC10B1**: Bit này xác định khối được làm đầy bởi kết quả của biến đổi A-D của 2 chế độ chuyển giao khối. ADC10B1 chỉ hợp lệ khi cờ ADC10IFG được set lần đầu trong suốt hoạt động DTC. ADC10TB cũng phải được set.

- 0 Khối 2 đầy
- 1 Khối 1 đầy

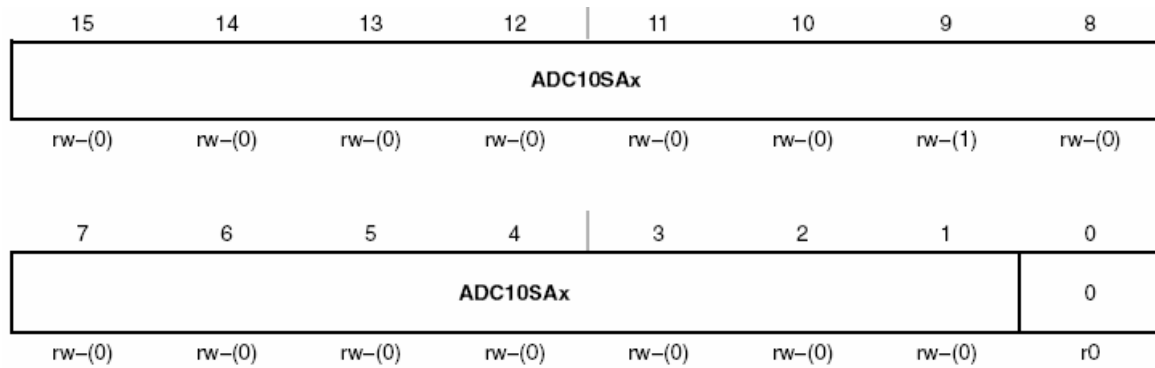
+ **ADC10FETCH**: Bình thường Bit này được Reset

IX.3.7. ADC10DTC1

Bit này xác định số chuyển giao trong mỗi khối. Khi Bit này bằng 0 thì DTC không được cho phép.

01h-0FFh số chuyển giao trên một khối.

IX.3.8. ADC10SA



Đây là thanh ghi địa chỉ bắt đầu của ADC10. Bit này bắt đầu địa chỉ cho DTC. Bit 0 thì không được sử dụng và luôn có giá trị là 0.