

Lập trình nhúng cơ bản

-----Phần thực hành----

Tuần 8

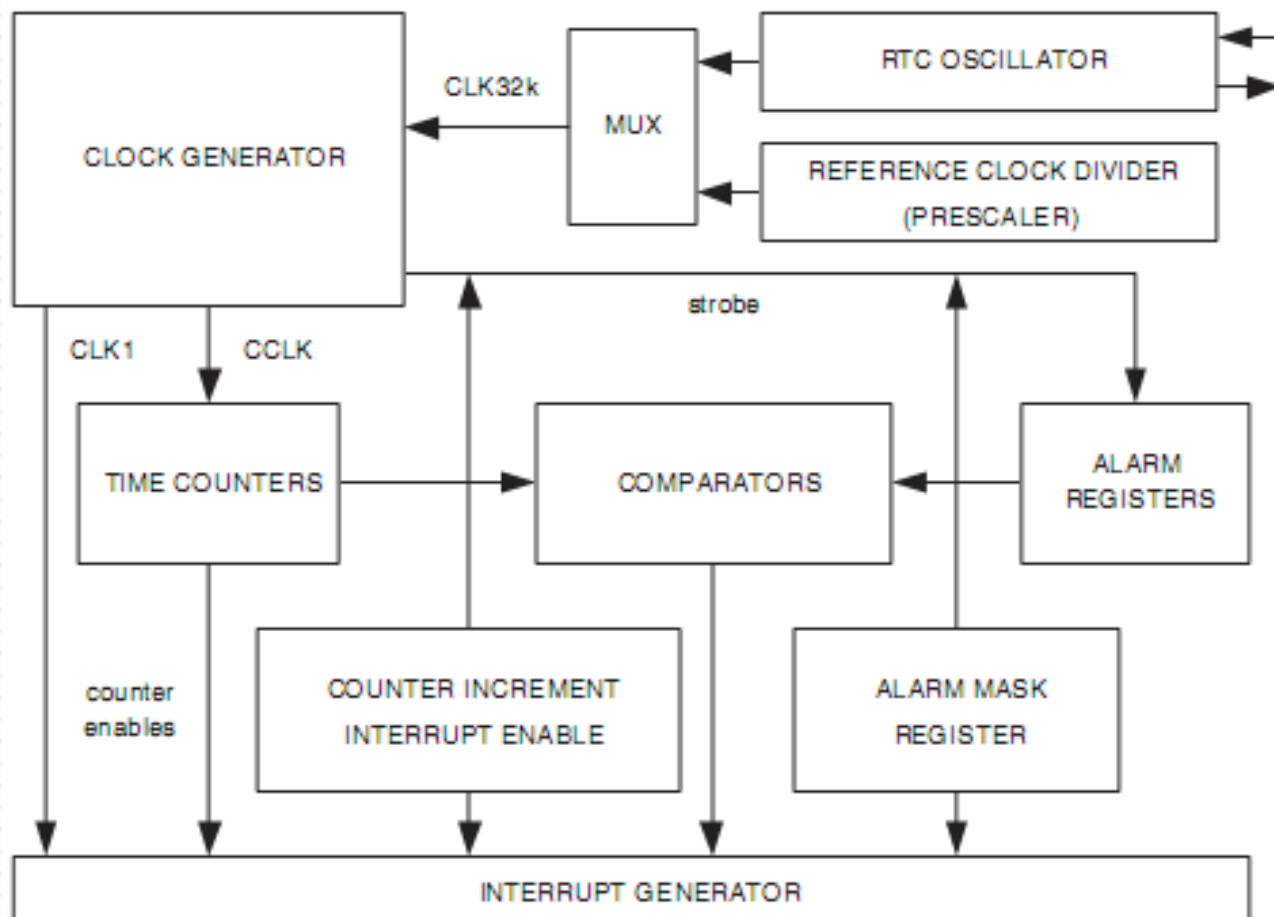
Nội dung

- ▶ **RTC (Real Time Clock)**
- ▶ **WDT (Watch Dog Timer)**

RTC – Real Time Clock

- ▶ RTC (Real Time Clock) là tập các bộ đếm dùng để đo thời gian khi hệ thống khởi động.
- ▶ RTC thỏa các điều kiện của một hệ thống real time :
 - Kết quả đúng
 - Hành vi đúng
 - Đúng thời điểm

Kiến trúc RTC



Mô tả các thanh ghi

Name	Size	Description	Access	Reset Value ^[1]	Address
CCR	4	Clock Control Register	R/W	*	0xE002 4008
PREINT	13	Prescaler value, integer portion	R/W	0	0xE002 4080
PREFRAC	15	Prescaler value, integer portion	R/W	0	0xE002 4084
SEC	6	Seconds Counter	R/W	*	0xE002 4020
MIN	6	Minutes Register	R/W	*	0xE002 4024
HOUR	5	Hours Register	R/W	*	0xE002 4028
DOM	5	Day of Month Register	R/W	*	0xE002 402C
DOW	3	Day of Week Register	R/W	*	0xE002 4030
DOY	9	Day of Year Register	R/W	*	0xE002 4034
MONTH	4	Months Register	R/W	*	0xE002 4038
YEAR	12	Years Register	R/W	*	0xE002 403C

CCR

- ▶ CCR (Clock Control Register) và 1 thanh ghi 4 bit để điều khiển Real Time Clock.
 - ▶ Gồm các bit quan trọng :
 - Bit 0 (CLKEN): Dùng để Enable bộ đếm thời gian. Cần phải Disable khi trước khi khởi động. (*)
 - Bit 1:(CTCRST) : Clock Tick Counter Reset.
 - Bit 2&3 (CTTEST) : bit kiểm tra, nên đặt là 0 trong quá trình vận hành.
 - ...
- (*) : *chỉ cần lưu ý bit này*

Prescaler

Name	Size	Description	Access	Address
PREINT	13	Prescale Value, integer portion	R/W	0xE002 4080
PREFRAC	15	Prescale Value, fractional portion	R/W	0xE002 4084

- ▶ **PREINT** = $\text{int}(\text{PCLK}/32768) - 1$ // ≥ 1
 - Chỉ dùng 13 bit (0–12)
- ▶ **PREFRAC** = $\text{PCLK} - ((\text{PREINT} + 1) \times 32768)$
 - Chỉ dùng 15 bit (0–14)

Prescaler (VD)

- ▶ Chip LPC2378 có PCLK = 15Mhz
- ▶ PREINT = $\text{int}(\text{PCLK}/32768) - 1$
= $[15,000,000/32768] - 1$ (*)
= $456 = 0x1C8$
- ▶ PREFRAC = $\text{PCLK} - ((\text{PREINT} + 1) \times 32768)$
= $25024 = 0x61C0$

(*) : $[\text{num}] = \text{floor}(\text{num})$

Các thanh ghi liên quan tới datetime

- ▶ RTC_SEC
- ▶ RTC_MIN
- ▶ RTC_HOUR
- ▶ RTC_DOM
- ▶ RTC_MONTH
- ▶ RTC_YEAR

Name	Size	Description	Access	Address
SEC	6	Seconds value in the range of 0 to 59	R/W	0xE002 4020
MIN	6	Minutes value in the range of 0 to 59	R/W	0xE002 4024
HOUR	5	Hours value in the range of 0 to 23	R/W	0xE002 4028
DOM	5	Day of month value in the range of 1 to 28, 29, 30, or 31 (depending on the month and whether it is a leap year). ^[1]	R/W	0xE002 402C
DOW	3	Day of week value in the range of 0 to 6 ^[1]	R/W	0xE002 4030
DOY	9	Day of year value in the range of 1 to 365 (366 for leap years) ^[1]	R/W	0xE002 4034
MONTH	4	Month value in the range of 1 to 12	R/W	0xE002 4038
YEAR	12	Year value in the range of 0 to 4095	R/W	0xE002 403C

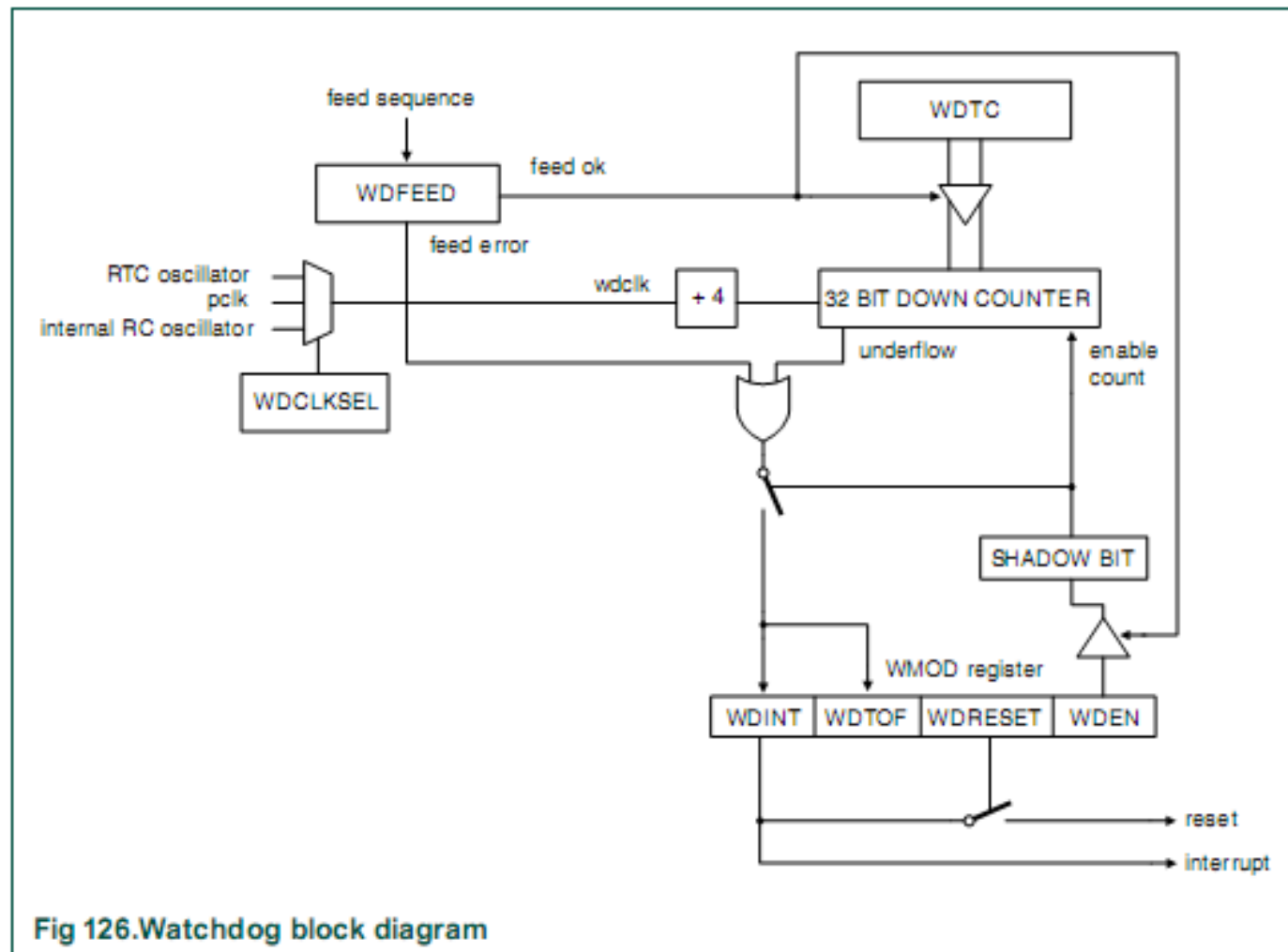
Khởi tạo RTC

```
RTCInit()  
{  
    Disable CCR;  
    Set Prescaler;  
    Enable CCR;  
}
```

WDT – Watchdog Timer

- ▶ Watchdog Timer được dùng để reset lại board/chạy interrupt sau một thời gian xác định sau khi WDT được kích hoạt.
- ▶ Việc kích hoạt WDT được gọi là feed.

Kiến trúc WDT



Kiến trúc WDT

- ▶ Thanh ghi đếm lùi thời gian là 32 bit WDTCx4.
- ▶ Có thể thiết lập thời gian timeout để chạy lùi thời gian trước khi WDT kích hoạt.

WDT – Các thanh ghi chính

Name	Description	Access	Reset Value ^[1]	Address
WDMOD	Watchdog mode register. This register contains the basic mode and status of the Watchdog Timer.	R/W	0	0xE000 0000
WDTC	Watchdog timer constant register. This register determines the time-out value.	R/W	0xFF	0xE000 0004
WDFEED	Watchdog feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the Watchdog timer with the value contained in WDTC.	WO	NA	0xE000 0008
WDTV	Watchdog timer value register. This register reads out the current value of the Watchdog timer.	RO	0xFF	0xE000 000C
WDCLKSEL	Watchdog clock source selection register.	R/W	0	0xE000 0010

WDT – Các bước thiết lập

- ▶ Xác định giá trị thanh ghi thời gian time-out WDTC.
- ▶ Chọn mode (WDMOD) (Reset/Interrupt).
- ▶ Bật WDT bằng cách ghi trị 0xAA vào WDFEED và sau đó ghi 0x55 vào WDFEED để kích hoạt WDT (kích hoạt bộ đếm lùi thời gian và đợi time-out sẽ reset board/interrupt).

WDT – Init()

```
DWORD WDTInit( void )
{
    VICVectAddr0 = (unsigned long)WDTHandler; /*Set Interrupt Vector*/
    VICVectCntl0 = 0;                          /*use it for Timer0 Interrupt*/
    VICIntEnable = (1 << 0);                    /*Enable Timer0 Interrupt*/
    WDTC = 0xff;
    WDMOD = 0x03;

    / * 0x01= Interrupt */
    / * 0x02= reset */

    WDFEED = 0xAA;
    WDFEED = 0x55;
    return( TRUE );
}
```


WDT – WDTFeed()

```
void WDTFeed( void )  
{  
    WDFEED = 0xAA;           /* Feeding sequence */  
    WDFEED = 0x55;  
    return;  
}
```

