

Lập trình nhúng cơ bản

-----Phần thực hành-----

Tuần 4

Nội dung

- ▶ Giới thiệu về ngắt trong LPC23xx
- ▶ Các thanh ghi liên quan tới điều khiển ngắt
- ▶ Hàm cài đặt ngắt `install_irq()`
- ▶ Ví dụ
- ▶ Bài tập

Giới thiệu về ngắt trong LPC23xx

- ▶ LPC2378 hỗ trợ 32 ngắt với 16 mức độ ưu tiên
- ▶ Có 2 loại ngắt : Interrupt request (IRQ) và Fast Interrupt Request (FIQ)
- ▶ Fast Interrupt Request là ngắt có độ ưu tiên cao nhất. IRQ có độ ưu tiên thấp hơn
- ▶ Nếu có nhiều ngắt cùng độ ưu tiên thì ngắt nào ở vector có chỉ số nhỏ hơn sẽ được thực thi trước.

Giới thiệu về ngắt trong LPC23xx

Bit	31	30	29	28	27	26	25	24
Symbol	I2S	I2C2	UART3	UART2	TIMER3	TIMER2	GPDMA	SD/MMC
Bit	23	22	21	20	19	18	17	16
Symbol	CAN1&2	USB	Ethernet	BOD	I2C1	AD0	EINT3	EINT2
Bit	15	14	13	12	11	10	9	8
Symbol	EINT1	EINT0	RTC	PLL	SSP1	SPI/SSP0	I2C0	PWM1
Bit	7	6	5	4	3	2	1	0
Symbol	UART1	UART0	TIMER1	TIMER0	ARMCore1	ARMCore0	-	WDT

- ▶ Bảng trên thể hiện vị trí thứ tự các bit tương ứng với ngắt cho từng loại đối tượng

Nội dung

- ▶ Giới thiệu về ngắt trong LPC23xx
- ▶ Các thanh ghi liên quan tới điều khiển ngắt
- ▶ Hàm cài đặt ngắt `install_irq()`
- ▶ Ví dụ
- ▶ Bài tập

Các thanh ghi liên quan tới điều khiển ngắt

- ▶ VICIRQStatus (32bits) : Thanh ghi trạng thái của ngắt loại IRQ, bit bật lên 1 cho biết đối tượng ngắt tương ứng được kích hoạt và loại ngắt là IRQ
- ▶ VICFIQStatus (32bits) : Thanh ghi trạng thái của ngắt loại FIQ, bit bật lên 1 cho biết đối tượng ngắt tương ứng được kích hoạt và loại ngắt là FIQ

Các thanh ghi liên quan tới điều khiển ngắt

- ▶ VICVectAddr n (32bits) : thanh ghi địa chỉ, chỉ tới hàm sẽ thực hiện nếu có ngắt xảy ra
- ▶ VICIntEnable (32bits) : cho giá trị bằng 1 ở bit muốn cài đặt ngắt
- ▶ VICVectPriority n (32bits) : thanh ghi xét độ ưu tiên

▶ n : bit thứ n (với $n: 0 \rightarrow 31$)

Ví dụ

```
VICVectAddr4 = (unsigned long) T0_IRQHandler;  
VICVectCntl4 = 15;  
VICIntEnable = (1 << 4);
```

- ▶ Với n=4: ngắt của Timer0
- ▶ Hàm thực hiện khi ngắt xảy ra là T0_IRQHandler
- ▶ Độ ưu tiên cho ngắt là 15
- ▶ Khởi động Bộ điều khiển ngắt ở bit thứ 4

Khai báo

- ▶ Hàm ngắt: có từ khóa `__irq` hoặc `__fiq`

- Ví dụ:

```
void T0_IRQHandler( void ) __irq
{
    ...
    VICVectAddr = 0;
}

__irq void T0_IRQHandler( void )
{
    ...
    VICVectAddr = 0;
}
```

Ví dụ

- ▶ Hiển thị giá trị các thanh ghi liên quan tới ngắt trong quá trình debug

The screenshot shows a debugger's Watch window with the following data:

Name	Value
VICVectAddr4	0x000007CC
VICVectPriority4	0x0000000F
VICIntEnable	0x00000010
VICIRQStatus	0x00000010
VICFIQStatus	0x00000000
TO_IRQHandler	0x000007CC
<type F2 to edit>	

Below the table are navigation buttons (back, forward, etc.) and tabs for 'Locals', 'Watch #1', 'Watch #2', and 'Call Stack'. The 'Watch #1' tab is currently selected.

Red boxes highlight the values 0x000007CC in the 'VICVectAddr4' and 'TO_IRQHandler' rows, and 0x00000010 in the 'VICIntEnable' and 'VICIRQStatus' rows. Red arrows point from these highlighted values to a 'TIMER0' register in a separate window at the top right, which also shows a value of 4.

Nội dung

- ▶ Giới thiệu về ngắt trong LPC23xx
- ▶ Các thanh ghi liên quan tới điều khiển ngắt
- ▶ Hàm cài đặt ngắt **install_irq()**
- ▶ Ví dụ
- ▶ Bài tập

Hàm cài đặt ngắt

```
▶ DWORD install_irq( DWORD IntNumber, void *HandlerAddr, DWORD Priority )
▶ {
▶     DWORD *vect_addr;
▶     DWORD *vect_prio;
▶
▶     VICIntEnClr = 1 << IntNumber;          /* Disable Interrupt */
▶     if ( IntNumber >= VIC_SIZE )
▶     {
▶         return ( FALSE );
▶     }
▶     else
▶     {
▶         /* find first un-assigned VIC address for the handler */
▶         vect_addr = (DWORD *)(VIC_BASE_ADDR + VECT_ADDR_INDEX + IntNumber*4);
▶         vect_prio = (DWORD *)(VIC_BASE_ADDR + VECT_PRIO_INDEX + IntNumber*4);
▶         *vect_addr = (DWORD)HandlerAddr; /* set interrupt vector */
▶         *vect_prio = Priority;
▶         VICIntEnable = 1 << IntNumber;    /* Enable Interrupt */
▶         return( TRUE );
▶     }
▶ }
```

Nội dung

- ▶ Giới thiệu về ngắt trong LPC23xx
- ▶ Các thanh ghi liên quan tới điều khiển ngắt
- ▶ Hàm cài đặt ngắt `install_irq()`
- ▶ Ví dụ
- ▶ Bài tập

Ví dụ

- ▶ Tạo ngắt cho UART
- ▶ Tạo ngắt cho Interrupt0 Button (INT0)

Ví dụ – tạo ngắt cho UART

▶ Tự làm

- `install_irq()..`
- Viết xử lý trong hàm ngắt...

Tạo ngắt cho Interrupt0 Button (INT0)

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P2.0	GPIO Port 2.0	PWM1.1	TXD1	TRACECLK ^[1]	00
3:2	P2.1	GPIO Port 2.1	PWM1.2	RXD1	PIPESTAT0 ^[1]	00
5:4	P2.2	GPIO Port 2.2	PWM1.3	CTS1	PIPESTAT1 ^[1]	00
7:6	P2.3	GPIO Port 2.3	PWM1.4	DCD1	PIPESTAT2 ^[1]	00
9:8	P2.4	GPIO Port 2.4	PWM1.5	DSR1	TRACESYNC ^[1]	00
11:10	P2.5	GPIO Port 2.5	PWM1.6	DTR1	TRACEPKT0 ^[1]	00
13:12	P2.6	GPIO Port 2.6	PCAP1.0	RI1	TRACEPKT1 ^[1]	00
15:14	P2.7	GPIO Port 2.7	RD2 ^[2]	RTS1	TRACEPKT2 ^[1]	00
17:16	P2.8	GPIO Port 2.8	TD2 ^[2]	TXD2	TRACEPKT3 ^[1]	00
19:18	P2.9	GPIO Port 2.9	USB_CONNECT1 ^[2]	RXD2	EXTIN0 ^[1]	00
21:20	P2.10	GPIO Port 2.10	EINT0	Reserved	Reserved	00
23:22	P2.11	GPIO Port 2.11	EINT1	MCIDAT1	I2STX_CLK	00
25:24	P2.12	GPIO Port 2.12	EINT2	MCIDAT2	I2STX_WS	00
27:26	P2.13	GPIO Port 2.13	EINT3	MCIDAT3	I2STX_SDA	00
29:28	P2.14	Reserved	Reserved	Reserved	Reserved	00
31:30	P2.15	Reserved	Reserved	Reserved	Reserved	00

Tạo ngắt cho Interrupt0 Button (INT0)

- ▶ `PINSEL4 = 0x01 << 20;`
- ▶ `VICVectAddr14 = (unsigned long)abc;`
- ▶ `VICVectCntl14 = 14;`
- ▶ `VICIntEnable = (1 << 14);`
- ▶

```
__irq void abc(void) {  
    .....// xuất ten cua ban ra HyperTerminal  
    EXTINT0=0x01;  
    VICVectAddr = 0;  
}
```

Nội dung

- ▶ Giới thiệu về ngắt trong LPC23xx
- ▶ Các thanh ghi liên quan tới điều khiển ngắt
- ▶ Hàm cài đặt ngắt `install_irq()`
- ▶ Ví dụ
- ▶ Bài tập

Bài tập

- ▶ Xây dựng chương trình truyền và nhận thông tin qua UARTs kết nối qua cổng COM trên PC (dùng ngắt cho UART)
 - Mở HyperTerminal nhập vào một số n. Firmware xuất ra màn hình HyperTerminal số n và cho đèn LED thứ n sáng
 - Bài toán cộng, trừ, nhân, chia với 2 toán tử. Xuất đáp số ra màn hình LCD và ra HyperTerminal
 - Menu thực hiện các chức năng

Chúc các bạn làm bài tốt

