Lectures 17: Project 2 Hints

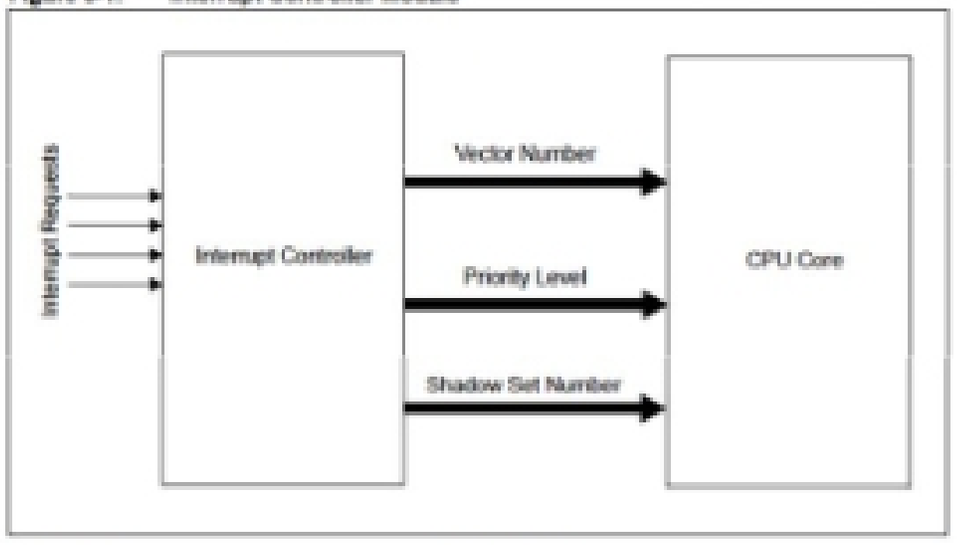
- Interrupts in our board
- Interrupt configurations for project 2
- Project 2 discussion

Reading assignment

- Reference manual
  - Sections 8.5, 8.6, 8.7
  - Sections 12.2.6, 12.3.9, 12.4
- Textbook
  - Chapter 5

Interrupt controller on our board

Figure 8-1: Interrupt Controller Module



Interrupts for Project 2

- Our job for project 2 is relatively easier.
- Since we only use the change notice interrupt, we do not have to dealing with multiple interrupts or nested interrupts…
- So what we need are two things:
  1. Configure I/O pins and interrupt settings
  2. Write the corresponding ISR

Change Notification (CN) pins

- Generate interrupt requests to the CPU upon a change of state on CN pins (input).
- Pin are sampled on every internal system clock cycle, SYSCLK.
- Pin values compared with the values sampled during the last read operation of the designated PORT register. Any mismatch will cause a "interrupt-on-change" signal.
- Each CN pin has a pull up connected to it. The pull ups act as a current source that is connected to the pin, and eliminate the need for external resistors when push button or keypad devices are connected.

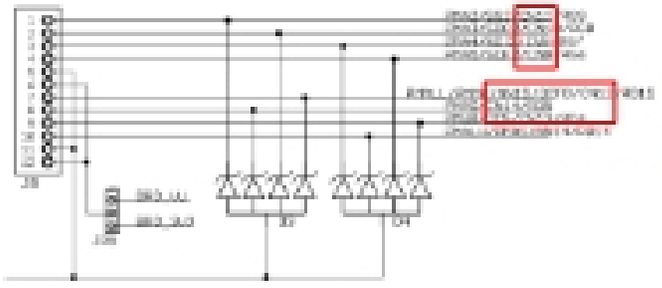
How to configure CN pins?

Configure CN pins through 3 control registers

- **CNCON:** CN Control Register <bit 15>
  - 1 = CN Module is enabled, 0 = disabled
- **CNEN:** CN Interrupt Enable Register <bit 21-0>
  - If a port pin is configured as an input (corresponding TRISx bit = 1)
    - 1 = Port pin input change notice enabled, 0 = disabled
  - If a port pin is configured as an output, CNENx bits have no effect.
- **CNPUE:** CN Pull-up Enable Register <bit 21-0>
  - If a port pin is configured as an input (corresponding TRISx bit = 1)
    - 1 = port pin pull-up enabled, 0 = disabled
  - If a port pin is configured as an output, the corresponding CNPUEx bit should be disabled.

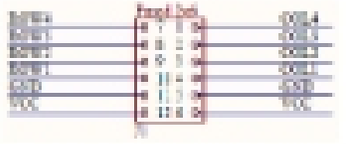
Connect Keypad to the Board

- Keypad outputs need to be connected to **CN pins** of the MCU
- **CNEN:** CN Interrupt Enable Register <bit 21-0>
  - Implies a total of 22 CN pins. Where exactly are they?
  - We can get all CN pins from board schematic



Connect Keypad to the Board

- Keypad outputs need to be connected to **CN pins** of the MCU



  - To which port should we connect the keypad?
    - Need 4 consecutive positions either 1~4 or 7~10
    - Keybad ⇔ JB or JJ
  - Whether to use row or column as outputs?
    - Use COLs as outputs

How to configure CN interrupts?

- Each interrupt has 7 control bits. The ones for CN interrupts are:

Control bits	Name	IO register	Bits	Setting
Interrupt Enable Control	CNIE	IEC1	0	1 = enable interrupt 0 = disable interrupt
Interrupt Flag Status	CNIF	IFS1	0	1 = has interrupt request 0 = has interrupt request
Interrupt Priority Control	CNIP<2:0>	IPC6	20-18	001-111 = priority levels 000 = Interrupt disabled
Interrupt Sub-priority Control	CNIS<1:0>	IPC6	17-16	01-11 = priority levels 00 = Interrupt disabled

Set by HW
Reset by SW

- How do we get such information?
  - Use keyword "change notice" to search the reference manual

Configure CN pins and interrupts

Done in main(), before you enter the while(1) loop

9 steps:

CN pins:
2. Set desired CN I/O pin as input (TRISx register bits = 1).
   Note: If the I/O pin is shared with an analog peripheral, it may be necessary to set the corresponding AD1PCFG register bit = 1 to ensure that the I/O pin is a digital input.
3. Enable CN Module ON (CNCON<15>) = 1.
4. Enable individual CN input pin(s), enable optional pull up(s).
5. Read corresponding PORT registers to clear mismatch condition on CN input pins.

CN interrupts:
6. Configure the CN interrupt priority.
7. Clear CN interrupt flag, CNIF = 0.
8. Enable CN interrupt enable, CNIE = 1.

1. Disable CPU interrupts.
9. Enable CPU interrupts.

CN Configuration Example (textbook)

Example 12-1: Change Notice Configuration Example

```
/*
    The following code example illustrates a Change Notice interrupt configuration for pins
    CN1(PORTC.BIT1), CN4(PORTB.BIT2) and CN15(PORTF.BIT5).
*/

    /* NOTE: disable vector interrupts prior to configuration */

    CNCON = 0x8000;         // Enable Change Notice module
    CNEN = 0x00140012;      // Enable individual CN pins CN1, CN4 and CN15
    CNPUE = 0x00140012;     // Enable weak pull ups for pins CN1, CN4 and CN15

    /* read port(s) to clear mismatch on change notice pins */
    PORTB;
    PORTC;
    PORTF;

    IPC6SET = 0x00140000;   // Set priority level=5
    IPC6SET = 0x00030000;   // Set Subpriority Level=3
                            // Could have also done this in single
                            // operation by assigning IPC6SET = 0x00170000

    IFS1CLR  = 0x0001;      // Clear the interrupt flag status bit
    IEC1SET  = 0x0001;      // Enable Change Notice interrupts

    /* re-enable vector interrupts after configuration */
```


ISR Code Example (textbook)

- When a CN interrupt occurs, the user should read the PORT register associated with the CN pin(s). This will clear the mismatch condition and set up the CN logic to detect the next pin change.

Example 12-2: Change Notice ISR Code Example

```
/*
    The following code example demonstrates a simple interrupt service routine for CN
    interrupts. The user's code at this vector can perform any application specific
    operations. The user's code must read the CN corresponding PORT registers to clear the
    mismatch conditions before clearing the CN interrupt status flag. Finally, the CN
    interrupt status flag must be cleared before exiting.
*/
void __ISR(_CHANGE_NOTICE_VECTOR, ipl5)ChangeNoticeHandler(void)
{
    ... perform application specific operations in response to the interrupt

    readB = PORTB           // Read PORTB to clear CN4 mismatch condition
    readC = PORTC           // Read PORTC to clear CN1 mismatch condition
    readF = PORTF           // Read PORTF to clear CN15 mismatch condition
    ...
    IFS1CLR  = 0x0001;      // Be sure to clear the CN interrupt status
                            // flag before exiting the service routine.
}
```