



Interrupts

Kai Huang, Biao Hu, Long Chen



Polling

- Constantly reading a memory location, in order receive updates of an input value

```
#include "system.h"
#include "io.h"

int main() {
    unsigned char key;
    unsigned char val = 0x01;
    IOWR( LED_BASE, 0, val );
    while(1) {
        key=IORD(KEY_BASE,0);
        if( key == 0x02 ){
            // key number 1 pressed
            // slide the bit one position to the left
            val = val << 1 | val >> 7;

            // update the LEDs
            IOWR( LED_BASE, 0, val );
            // wait until the key is released
            while( IORD( KEY_BASE != 0x03 ) ){}
        }
    }
    return 0;
}
```

Polling

Polling

Altera Interrupts

- Initialization
- Enable the interrupt for the specific input

```
IOWR_ALTERA_AVALON_PIO_IRQ_MASK(<BASE>, <MASK>);
```

- Set the edge capability

```
IOWR_ALTERA_AVALON_PIO_EDGE_CAP(<BASE>, <VAL>)
```

- Register handler

```
alt_ic_isr_register( <IRQ_Controller_ID>, <IRQ>,  
                   <isr_function>, <isr_context>, <flags> )
```

Tasks

- Control the LED with KEY on your board
- Try out and understand the Interrupt based KEY-LED package

