Example 1: Polling Pushbuttons with NIOS II

In this example, the function polls the pushbutton specified by the variable `button`.

```
void get_button_push( int button )
{
    // Declare a variable to store the bit mask used to select the desired bit
    alt_u8 mask;
    // Setup the mask based on the desired button (0 to 3)
    mask = 0x1;
    mask = mask << button;

    // Wait in a tight polling loop for the desired button to be pushed
    while((IORD(BUTTON_PIO_BASE, 0) & mask) != 0);

    // Display on the console that the button was pushed
    printf("You pushed button %d\n", button);

    // Wait in a tight polling loop for the desired button to be released
    while((IORD(BUTTON_PIO_BASE, 0) & mask) == 0);
}
```

This function above assumes that the hardware has a parallel port called `button_pio`. The port is initialized prior to calling the `get_button_push` function. The `get_button_push` function is called with a `button` parameter ranging from 0 to 3. In the function, the `mask` is set so that only the button of interest will be polled. The mask will be set as follows:

Button	Mask Value (in Binary)
0	0000 0001
1	0000 0010
2	0000 0100
3	0000 1000

When a button is pushed, the corresponding bit in the data register is a 0. When a button is not pushed, the corresponding bit in the data register is a 1. The first loop reads the data register for the button_pio using the following macro:

```
IORD(BUTTON_PIO_BASE, 0)
```

The IORD macro reads the register specified by the base address (in this case BUTTON_PIO_BASE) in combination with the register number. The Altera Peripherals documentation on PIOs indicates that register offset 0 corresponds with the data register in a PIO.

The bits from the register are then masked so that all bits are 0 except for the one corresponding to the button of interest which will be set (equal to 1) or cleared (equal to 0) depending on the state of the button. By masking the other bits, it means that you can push the other buttons and nothing will happen. If the button of interest is not pushed, then the masked value will not be 0, and execution will keep looping. Otherwise the loop will be exited, and the message is output. The function then polls the button until it is released using another while loop.