

NIOS Interrupts Example

Last updated 6/1/20

NIOS Interrupts – Example

These slides describe the implementation of an interrupt controller for the NIOS system

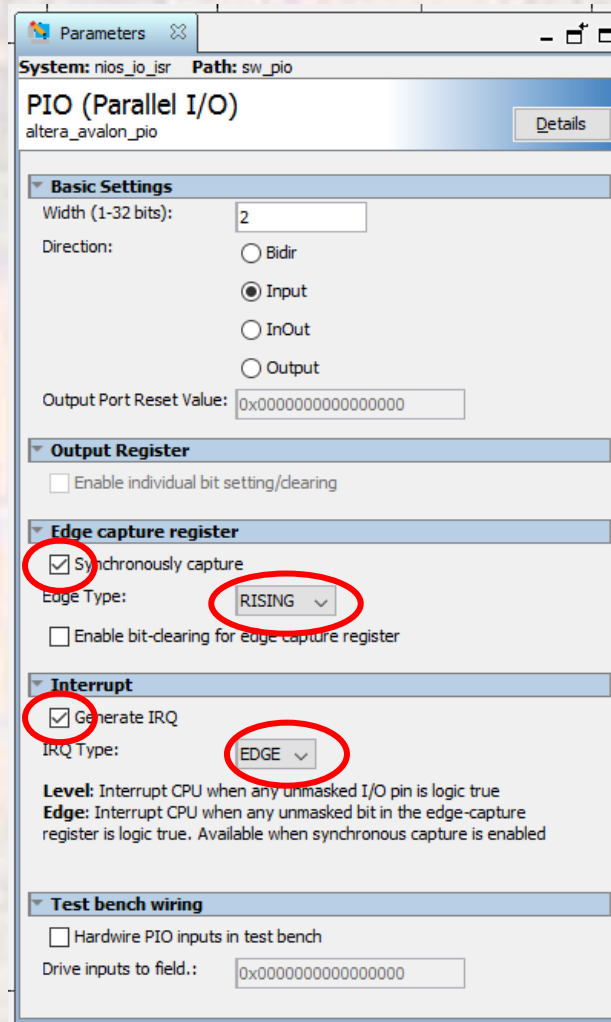
Upon completion: You should be able to implement an interrupt controller in a NIOS system

NIOS Interrupts - Example

- HAL Framework – Interrupts
 - Modify our counter project to increment the count based on a switch rising edge – via interrupt
 - Use the edge capture flag as the signal to increment the counter

NIOS Interrupts - Example

- HAL Framework – Interrupts
 - Edit sw_pio



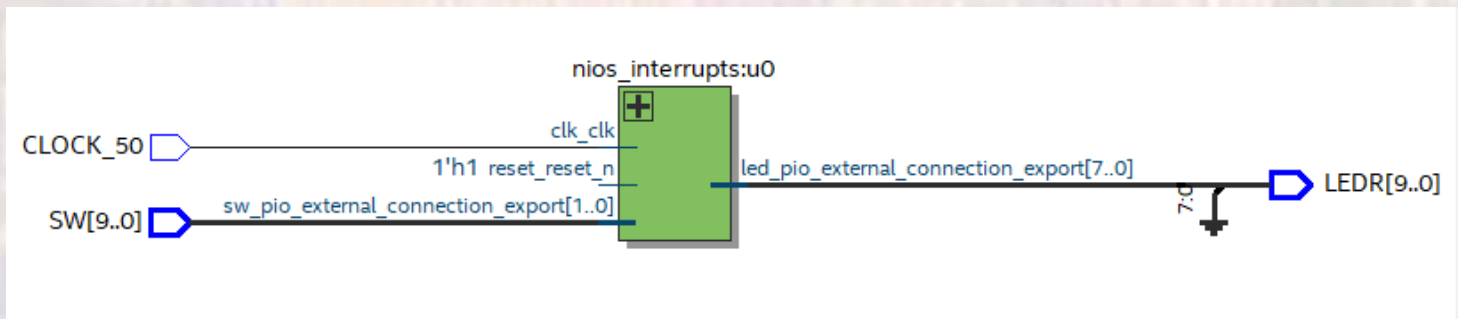
NIOS Interrupts - Example

- HAL Framework – Interrupts

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags
<input checked="" type="checkbox"/>		clk_0	Clock Source						
<input checked="" type="checkbox"/>		clk_in	Clock Input	clk	exported				
<input checked="" type="checkbox"/>		clk_in_reset	Reset Input	reset					
<input checked="" type="checkbox"/>		clk	Clock Output	clk_0					
<input checked="" type="checkbox"/>		clk_reset	Reset Output	reset					
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)						
<input checked="" type="checkbox"/>		clk1	Clock Input	clk_0					
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	[clk1]		0x0000_8000	0x0000_cfff		
<input checked="" type="checkbox"/>		reset1	Reset Input	[clk1]					
<input checked="" type="checkbox"/>		nios2_gen2_0	Nios II Processor						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		data_master	Avalon Memory Mapped Master	[clk]					
<input checked="" type="checkbox"/>		instruction_master	Avalon Memory Mapped Master	[clk]					
<input checked="" type="checkbox"/>		irq	Interrupt Receiver	[clk]				IRQ 0	IRQ 31
<input checked="" type="checkbox"/>		debug_reset_request	Reset Output	[clk]					
<input checked="" type="checkbox"/>		debug_mem_slave	Avalon Memory Mapped Slave	[clk]					
<input checked="" type="checkbox"/>		custom_instruction_m...	Custom Instruction Master	[clk]		0x0001_0800	0x0001_0fff		
<input checked="" type="checkbox"/>		jtag_uart_0	JTAG UART						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		avalon_jtag_slave	Avalon Memory Mapped Slave	[clk]		0x0001_1048	0x0001_104f		
<input checked="" type="checkbox"/>		irq	Interrupt Sender	[clk]					16
<input checked="" type="checkbox"/>		timer_0	Interval Timer						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	[clk]		0x0001_1000	0x0001_101f		
<input checked="" type="checkbox"/>		irq	Interrupt Sender	[clk]					1
<input checked="" type="checkbox"/>		sysid_qsys_0	System ID Peripheral						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		control_slave	Avalon Memory Mapped Slave	[clk]		0x0001_1040	0x0001_1047		
<input checked="" type="checkbox"/>		led_pio	PIO (Parallel I/O)						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	[clk]		0x0001_1030	0x0001_103f		
<input checked="" type="checkbox"/>		external_connection	Conduit	led_pio_external_conne...					
<input checked="" type="checkbox"/>		sw_pio	PIO (Parallel I/O)						
<input checked="" type="checkbox"/>		clk	Clock Input	clk_0					
<input checked="" type="checkbox"/>		reset	Reset Input	[clk]					
<input checked="" type="checkbox"/>		s1	Avalon Memory Mapped Slave	[clk]		0x0001_1020	0x0001_102f		
<input checked="" type="checkbox"/>		external_connection	Conduit	sw_pio_external_conne...					
<input checked="" type="checkbox"/>		irq	Interrupt Sender	[clk]					10

NIOS Interrupts - Example

- HAL Framework – Interrupts



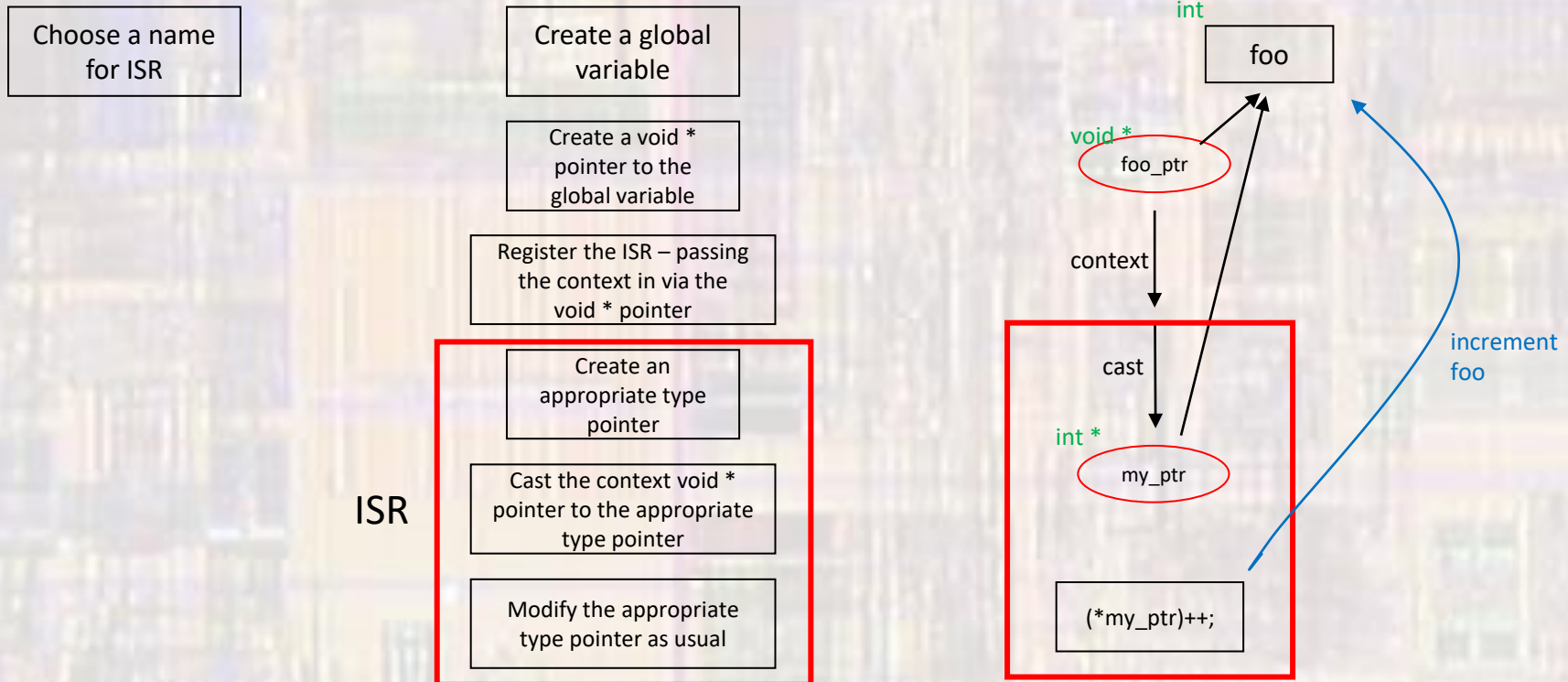
NIOS Interrupts - Example

- HAL Framework – Interrupts

```
#include "system.h"  
#include "altera_avalon_pio_regs.h"  
#include "sys/alt_irq.h"  
#include "alt_types.h"  
#include <stdio.h>
```

NIOS Interrupts - Example

- HAL Framework – Interrupts



NIOS Interrupts - Example

- HAL Framework – Interrupts

```
// ISR Prototype  
void io_switch_isr(void * context);
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
// Global variable to hold the value of the  
// edge capture  
volatile int edge_val;
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
// Cast the global variable to the required ISR  
// type of pointer - void *  
void * edge_val_ptr = (void *) &edge_val;
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
// Register the interrupt
alt_ic_isr_register(
    SW_PIO_IRQ_INTERRUPT_CONTROLLER_ID,
    SW_PIO_IRQ,
    io_switch_isr,
    edge_val_ptr,
    0x00
);
```

system.h

```
-----
#define SW_PIO_HAS_IN 1
#define SW_PIO_HAS_OUT 0
#define SW_PIO_HAS_TRI 0
#define SW_PIO_IRQ 10
#define SW_PIO_IRQ_INTERRUPT_CONTROLLER_ID 0
#define SW_PIO_IRQ_TYPE "EDGE"
#define SW_PIO_NAME "/dev/sw_pio"
#define SW_PIO_RESET_VALUE 0
-----
```

sys/alt_irq.h

```
int alt_ic_isr_register(
    alt_u32 ic_id,           interrupt controller ID
    alt_u32 irq,            interrupt ID
    alt_isr_func isr,      isr name
    void * isr_context,    pointer to any passed context
    void * flags,          reserved – 0
);
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
void io_switch_isr(void * context){
    // expect the context passed to be a pointer
    // to the variable to hold the edge capture information

    // create a pointer variable to hold the context
    volatile int * edge_ptr;
    edge_ptr = (volatile int *) context;

    // Read the edge capture register and assign the
    // value to the ptr variable
    *edge_ptr = IORD_ALTERA_AVALON_PIO_EDGE_CAP(SW_PIO_BASE);

    // Clear the edge capture register
    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(SW_PIO_BASE, 0);
} // end io_switch_isr
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
/*
 * nios_interrupts.c
 *
 * Created on: Sep 20, 2018
 * Author: johnsontimoj
 */

#include "system.h"
#include "altera_avalon_pio_regs.h"
#include "sys/alt_irq.h"
#include "alt_types.h"
#include <stdio.h>

// ISR Prototype
void io_switch_isr(void * context);
// Switch setup prototype
void io_switch_setup();

// Global variable to hold the value of the
// edge capture
volatile int edge_val = 0;
```

```
int main(void){
    printf("Entered main\n");

    int count = 0;

    // Configure the IO switches
    io_switch_setup();

    // Loop and wait for edges to update the count
    while(1){
        if(edge_val & 0x01){
            count++;
            edge_val = 0;
            printf("incrementing count : %i\n", count);
        } else if (edge_val & 0x02){
            count--;
            edge_val = 0;
            printf("decrementing count : %i\n", count);
        }
        // Output the count to the LEDs
        IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_BASE, count);
    }

} // end main
```

NIOS Interrupts - Example

- HAL Framework – Interrupts

```
void io_switch_setup(void) {  
    // Enable interrupts on 2 switches  
    IOWR_ALTERA_AVALON_PIO_IRQ_MASK(SW_PIO_BASE, 0x03);  
  
    // Clear any existing interrupts  
    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(SW_PIO_BASE, 0x00);  
  
    // Cast the global variable to the required ISR  
    // type of pointer - void *  
    void * edge_val_ptr;  
    edge_val_ptr = (void *) &edge_val;  
  
    // Register the interrupt  
    alt_ic_isr_register(SW_PIO_IRQ_INTERRUPT_CONTROLLER_ID,  
                      SW_PIO_IRQ,  
                      io_switch_isr,  
                      edge_val_ptr,  
                      0x00);  
} // end io_sw_setup
```

```
void io_switch_isr(void * context){  
    // expect the context passed to be a pointer  
    // to the variable to hold the edge capture information  
    //  
    // create a pointer variable to hold the context  
    volatile int * edge_ptr;  
    edge_ptr = (volatile int *) context;  
  
    // Read the edge capture register and assign the  
    // value to the ptr variable  
    *edge_ptr = IORD_ALTERA_AVALON_PIO_EDGE_CAP(SW_PIO_BASE);  
  
    // Clear the edge capture register  
    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(SW_PIO_BASE, 0);  
} // end io_switch_isr
```