An Introduction to Lab2

Lecture 4 for Information Processing

Aaron Zhao, Imperial College London, a.zhao@imperial.ac.uk

What is in this lab?

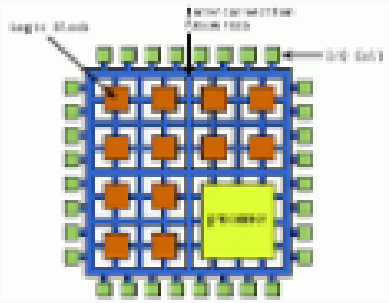
- Design a NIOSII system.
- Understand the design process of a NIOSII system.
- Program the Max 10 FPGA chip on the DE10-Lite board with your soft processor.
- Write code that runs on the NIOS processor to display a message on a terminal.
- Explore and test the capabilities of your NIOS II system design.

Soft Vs. Hard processors

- Hard processors normally have a fixed architecture
- NIOSII is a soft processor
  - customizable (*eg.* size, performance)
  - add or remove certain features (*eg.* floating-point units)
  - custom instructions or extensions of the instruction set
- Soft processors are normally on FPGAs, running at a slower clock rate

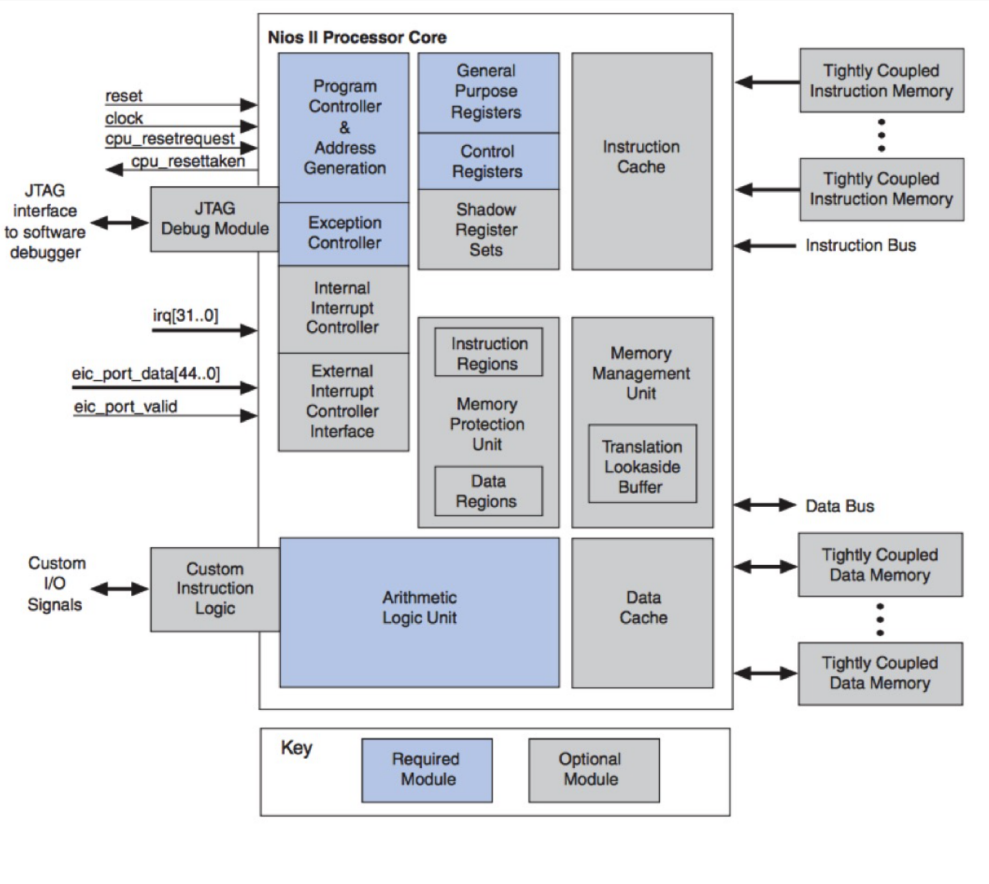
Hard Processors (eg. Intel, ARM, AMD)

Soft Processors (MicroBlaze, NIOS II)



Processor architecture

- Register files
- Arithmetic logic unit (ALU)
- icache and dcache
- instruction bus and data bus ...



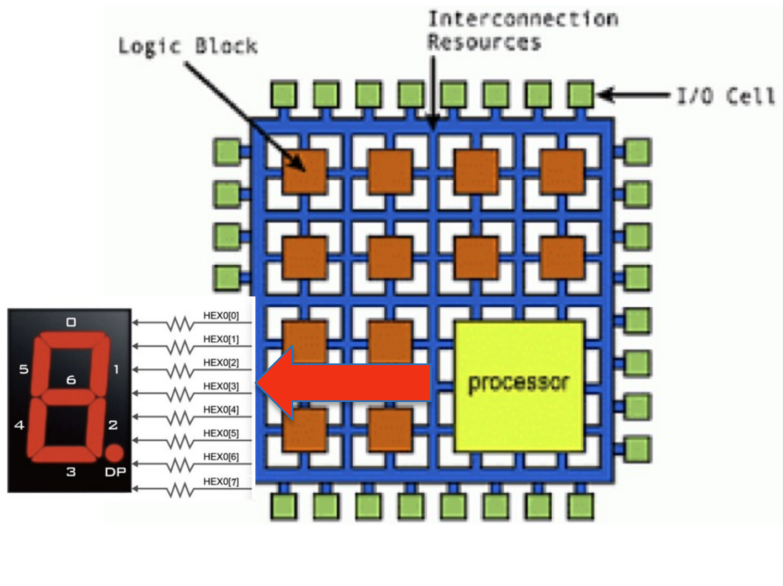
Processor implementation

- A NIOSII implementation is a set of design choices
- A functional unit (i.e. fp-mult) can be implemented in
  - hardware
  - emulated in software
  - omitted completely...
- Another example is division support

What do you do in Lab2?

- Control the lighting sequence on LEDs through the NIOSII processor

```c
int main()
{
  int switch_datain;
  alt_putstr("Hello from Nios II!\n");
  alt_putstr("This is my first application!\n");

  char msg[20];

  /* Event loop never exits. */
  while (1){
    switch_datain = IORD_ALTERA_AVALON_PIO_DATA(BUTTON_BASE);
    switch_datain &= (0b0000001111);

    if (switch_datain==0)
        alt_putstr("both switches pressed\n");
    else {
    if (switch_datain==3)
        alt_putstr("no switches pressed\n");
    if (switch_datain==1)
        alt_putstr("first switch pressed\n");
    if (switch_datain==2)
       alt_putstr("second switch pressed\n");
    }

    int i;
    for (i=0; i<300000; i++)
        msg[0]='2';

    IOWR_ALTERA_AVALON_PIO_DATA(LED_BASE, switch_datain);
    IOWR_ALTERA_AVALON_PIO_DATA(HEX0_BASE, switch_datain);
  }
```



Questions?

Questions?

An Introduction to Lab3

Lecture 5 for Information Processing

Aaron Zhao, Imperial College London, a.zhao@imperial.ac.uk

What is in this lab?

- Design a NIOS II system that interfaces with the accelerometer on DE10-lite board.
- Understand the SPI interface.
- Learn how to read the acceleration value provided by the accelerometer.
- Design a low-pass FIR filter to process the readings.
- Investigate the impact of using low arithmetic precision to the quality of the results and the performance of your system.

Accelerometer

- Analog Devices' ADXL345 chip
- 3-axis accelerometer, it measures acceleration in three directions, which are referred to as x-axis, y-axis and z-axis.
- Serial Peripheral Interface (SPI) / I2C
- 16-bit digital output

How does NIOS interact with the accelerometer

- Add accelerometer_spi IP
  - IP controls the accelerometer and provides an SPI interface to NIOS
  - 58 internal registers
  - Memory mapped through two 8-bit registers: Address and Data
  - Memory mapped means they are mapped to specific memory addresses at the time the core is instantiated in a Qsys-developed system.

Address	7 · · · 6	5 · · · 0	
0x10004020	0	Addr	Address register
0x10004021	Data		Data register

Address	Register Name	Description
0x32	*DATAX0*	Low-order byte of *x*-axis acceleration.
0x33	*DATAX1*	High-order byte of *x*-axis acceleration.
0x34	*DATAY0*	Low-order byte of *y*-axis acceleration.
0x35	*DATAY1*	High-order byte of *y*-axis acceleration.
0x36	*DATAZ0*	Low-order byte of *z*-axis acceleration.
0x37	*DATAZ1*	High-order byte of *z*-axis acceleration.

Serial Peripheral Interface (SPI)

- Synchronous Serial Communication
- Short distances
- Embedded systems
- Duplex communication and a Master-Slave architecture

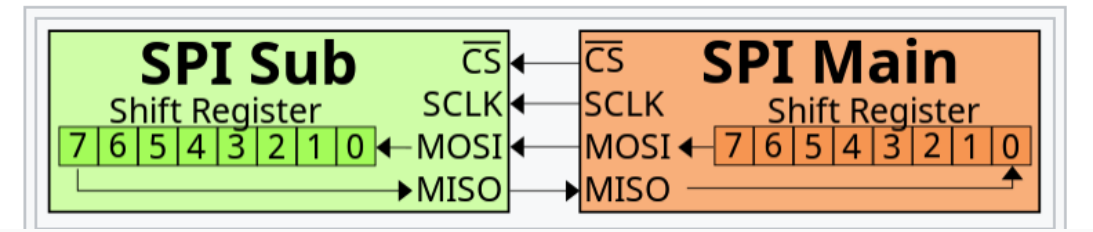
Serial Peripheral Interface (SPI)

SPI has four logic signals (which go by alternative namings):

- SCLK : Serial Clock (clock signal from main)
- MOSI : Main Out Sub In (data output from main)
- MISO : Main In Sub Out (data output from sub)
- CS : Chip Select

SPI communication

- Chip Select (CS) first, it is possible to have multiple slaves.
- Slave does not have the clock, you must provide one.
- Duplex communication, you are transferring and receiving at the same time.



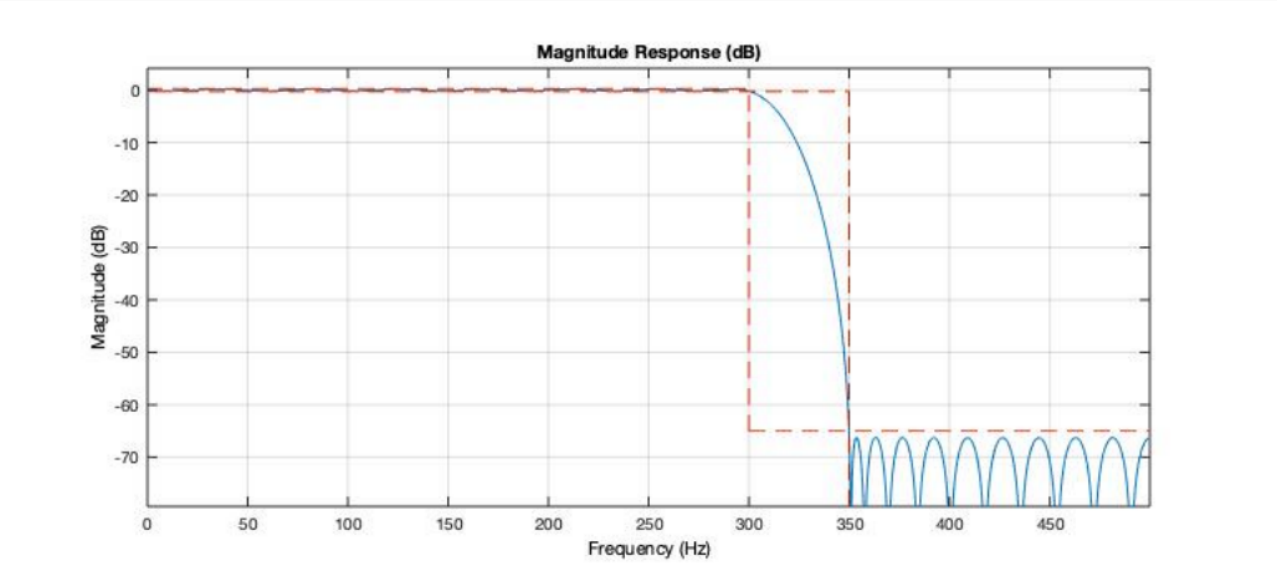
Program NIOS to read accelerometer values

- Understand code to interface with accelerometer
- Drive the LEDs with the accelerometer value

```
int main() {

    alt_32 x_read;
    alt_up_accelerometer_spi_dev * acc_dev;
    acc_dev = alt_up_accelerometer_spi_open_dev("/dev/accelerometer_spi");
    if (acc_dev == NULL) { // if return 1, check if the spi ip name is "accelerometer_spi"
        return 1;
    }

    timer_init(sys_timer_isr);
    while (1) {

        alt_up_accelerometer_spi_read_x_axis(acc_dev, & x_read);
        // alt_printf("raw data: %x\n", x_read);
        convert_read(x_read, & level, & led);

    }

    return 0;
}
```

FIR filter implementation and optimisation

- Moving average with a 5-tap filter
- Extend this to a low-pass N-tap filter
  - Use Matlab to design your filter
- Optimise you program
  - Convert floats to fixed-point values
  - Observe the impact on performance and results



Questions?

Questions?

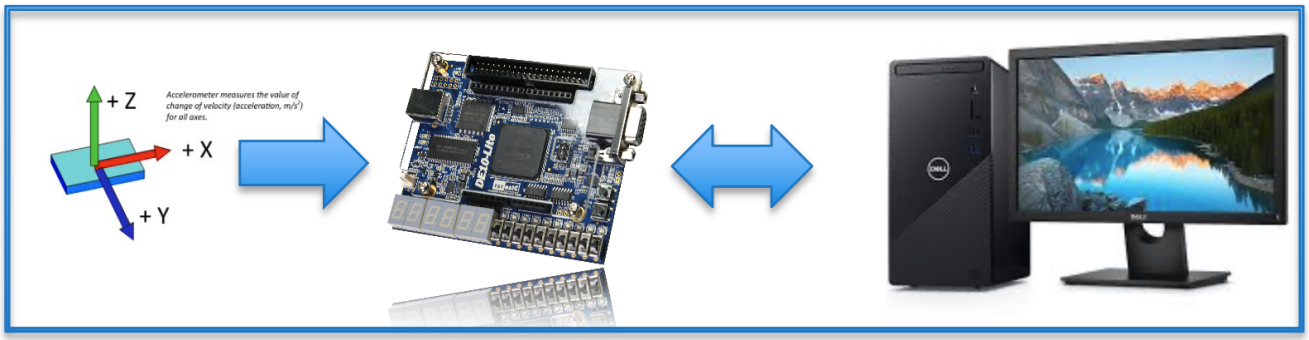
An Introduction to Lab4

Lecture 6 for Information Processing

Aaron Zhao, Imperial College London, a.zhao@imperial.ac.uk

What is in this lab?

- Understand how to establish a communication process of a NIOSII system with a host PC.
- Establish a number of functions/commands that would allow you to communicate between the board and the host PC.
- Extra: Learn how to add off-chip memory to your system



UART communication

- UART (universal asynchronous receiver-transmitter)
- Device to Device communication
- Asynchronous Serial Communication – (2 wires)
- Agreed frequency of reading – Baud rate
- PC is the master

1	5-9	0-1	1-2
Start Bit	Data Frame	Parity Bits	Stop Bits

Lab structure

- Part 1: Give you an example to understand the communication
- Part 2: Integrate UART with Lab3
- Part 3: Add command to update the coefficient. Conversion of characters to numbers
- Part 4: Plot received accelerometer data at real time

1	5-9	0-1	1-2
Start Bit	Data Frame	Parity Bits	Stop Bits

Questions?

Questions?