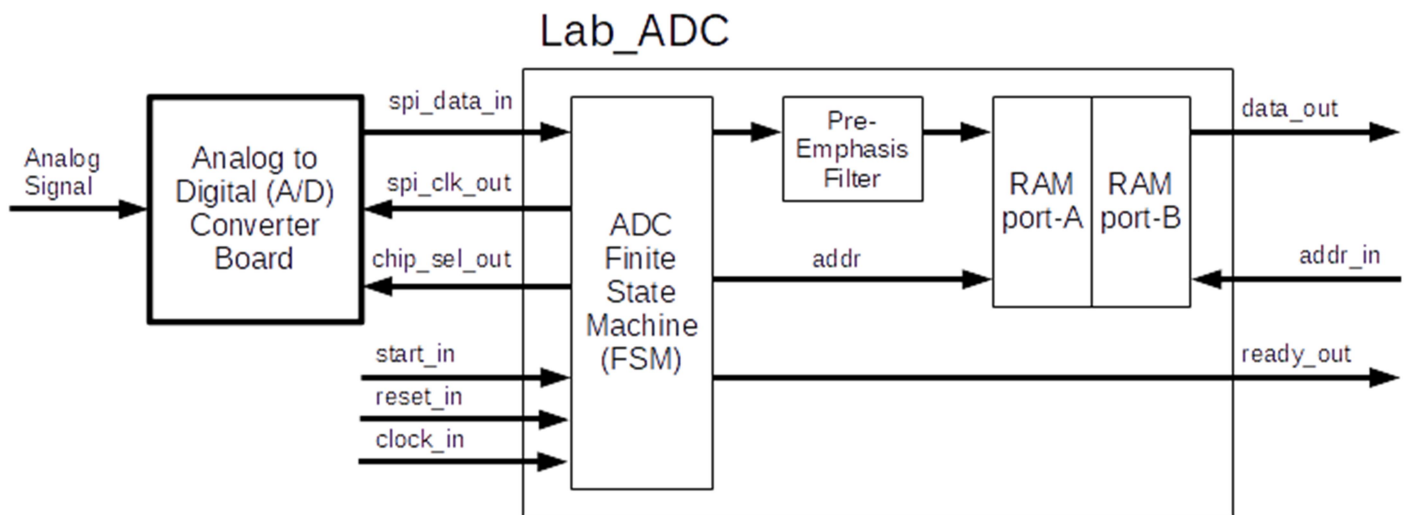


EEE 491 Lab Assignment

Lab-ADC: SPI Interface and Pre-Emphasis Filter

1. Technical Specifications/Requirements

A Serial Peripheral Interface (SPI) to an analog to digital converter (ADC) board, pre-emphasis filter, and a dual-port RAM shall be designed and implemented on the FPGA of Basys-3 board. Refer to the figure below.



The Lab-ADC shall access to ADC board via SPI signals in order to sample analog audio signals at the input of the ADC. The ADC board has ADC121S101 12-bit A/D converter IC. ADC sample rate shall be at 16K samples per second. The SPI shall operate at 12.5MHz clock rate.

Since ADC is single power supplied by 3.3V and 0V (GND), each sample at the output of ADC converter has a DC level. Offset this DC level and represent the signals as signed data (two's complement).

Lab-ADC shall detect existence of voice from the samples. Propose a method for detecting start and end of voice. Only the samples between start and end of voice shall be stored in the RAM. Fill unused remaining space of the RAM with silence data value.

Pre-emphasis refers to filtering that emphasizes the higher frequencies. Its purpose is to balance the spectrum of voiced sounds that have a steep roll-off in the high-frequency region. Pre-emphasis shall use $H(z)=1 - a(1 / z)$ discrete function, where $a=0.5$. The signed samples shall pass through this filter and shall be stored in dual-port RAM.

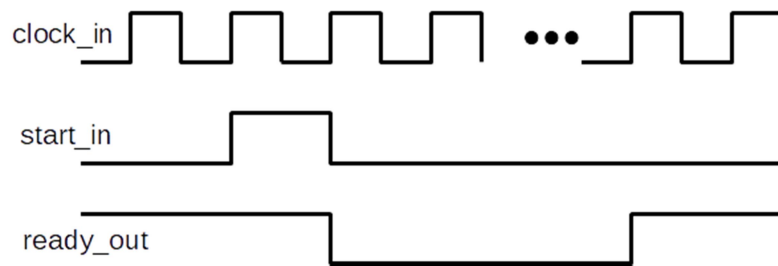
The size of the RAM shall be capable of storing about one-second duration of samples. A "simple dual-port RAM" shall be selected from the "Block Memory Generator" in Vivado "IP Catalog". The dual-port RAM has two ports; port-A and port-B. Port-A shall be write-only port, and Port-B shall be read-only port.

The ADC board shall be connected to one of PMOD-JA, PMOD-JB, PMOD-JC ports on the Basys-3 board. PMOD ports and pin assignments of FPGA are given in reference manual of the Basys-3 board.

Lab-ADC shall have the following input/output (IO) signals:

- reset_in: input control signal, which resets all the registers in the design.
- clock_in: input clock signal of the 100MHz clock oscillator on the Basys-3 board.

- spi_clk_out: output clock signal, which is 12.5MHz clock of SPI interface of ADC board.
- chip_sel_out: output signal, which is active-low chip select of SPI interface of ADC board.
- spi_data_in: input signal, which is serial data of SPI interface of ADC board.
- addr_in: 14-bit input signal that addresses the read-only port-B of the dual-port RAM in Lab-ADC.
- data_out: 12-bit output signal of the read-only port-B of the dual-port RAM in Lab-ADC.
- start_in: input control signal, which starts the sampling and ADC conversion. This signal shall be active high asserted as a single clock pulse for one clock-cycle duration (refer to waveform below).
- ready_out: output status signal that indicates if Lab-ADC is done with the sampling and storing of all the samples in its dual-port RAM and ready for the next execution. This signal is active high and asserted low just after the start signal is asserted and becomes active after the last sample is stored in the dual port RAM (refer to waveform below).



Write a test-bench in VHDL for the simulation of your design. Verify by using your test-bench simulation results that your design satisfies the technical specifications.

Integrate Lab-ADC with Lab-CTRL, Lab-DEBUG and the other available sub-systems whenever possible in a new top level design by using “component” statements in VHDL.

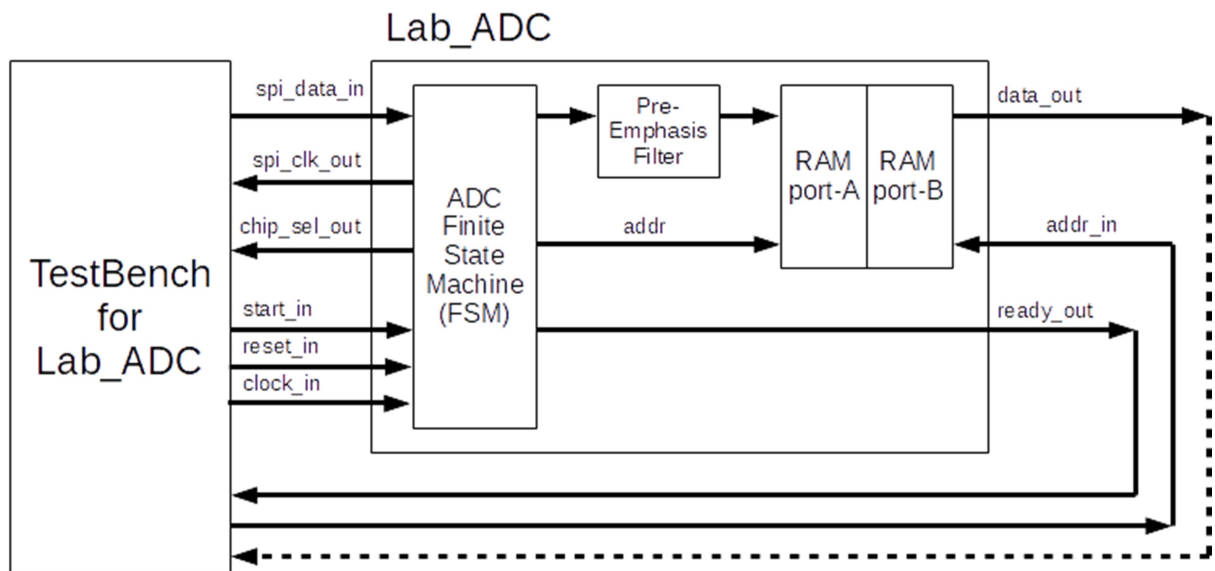
Lab-DEBUG accesses the read-only Port-B of the dual-port RAM of Lab-ADC and transfers the data to PC/MATLAB. Control and observe the process sequence using start_in and ready_out signals of the components. On the Basys-3 board, use a button for each start_in signaling and a LED for each ready_out signals of the components.

2. Demonstration

- Show that your design is implemented on FPGA.
- Present your test bench simulation results: inputs, outputs (waveforms). Clearly show the SPI signals on the waveforms. Observe the input, output waveforms, and verify the technical specifications listed above.
- Configure your integrated design on Basys-3 board. Use start button to start the operation. Use ready LEDs to observe/verify end of processes. If needed you may add more LEDs for further observations.
- Apply DC voltage levels (0.0V, 1.0V, 2.0V, and 3.3V) to the input of the ADC board. Transfer the all the samples in the dual-port RAM to MATLAB by using the Lab-DEBUG. Plot samples and plot frequency spectrum on MATLAB, and verify the results. Check the spectrum plot for noise performance.
- Use signal generator and sample various sinusoidal signals at different frequencies (100Hz, 300Hz, 1KHz, 3KHz) and amplitudes (0.1V, 1V, 2V, 3V). Transfer the all the samples in the dual-port RAM to MATLAB by using the Lab-DEBUG. On MATLAB, plot samples, plot frequency spectrum and calculate signal to noise ratio (SNR). Verify results.
- Show that the detection of voice existence function is properly working.
- Show that the pre-emphasis filter properly works.

3. Guidance

- During an FPGA design, always write a test-bench code in order to simulate your design. A test-bench and the design under test block diagram are shown in the figure below as an example. The test-bench generates serial SPI data on SPI interface depending on the signals `spi_clk_out` and `chip_sel_out`. Lab-ADC reads the SPI data and writes all the samples data to dual-port RAM. Test bench checks the `ready_out` signal and when it is active then test bench start to generate `addr_in` signal and `data_out` signal is observed on the simulation output waveform and validated. Note that test bench may also check the validity of the data content.



- Configure “Basys3_Master.xdc” Xilinx Design Constraints (XDC) file provided by the Basys-3 board manufacturer (Digilent). The file includes FPGA pin assignments on Basys-3 board. Uncomment the lines corresponding to used pins and rename the used ports (in each line, after “get_ports”) according to the top level signal names in your project.
- Download and use the data sheet of the ADC121S101 for SPI interface. Be careful not to put the ADC in to shutdown mode unintentionally; better to run a single dummy sampling at the beginning of sampling.
- Study on ADC board schematics (inputs and outputs). Note that VREF input of ADC is at pin VCC and it is connected to 3.3V, and it is supplied from the Basys-3 board (3.3V and 0V). The analog signal at the input of the ADC has a bias level at about $3.3V/2$ and its converted digital value shall be subtracted from the data.
- Note that audio signal at the input of ADC contains noise. Straightforward method to reduce noise is to make a twisted cable connection to ADC board input from the audio source, and to keep the cable length as short as possible.
- The pre-emphasis filter function has 0.5 multiplication operation, which is a division by two and simply calculated by a right shift operation with sign bit feed to most significant bit of the data.
- The start input signal from a button on Basys-3 board is noisy that has multiple transitions of logic-0s and logic-1s, which last in several milliseconds. Your design must avoid multiple re-starts for its operation.
- An example for a simple dual-port RAM memory configuration is shown below. Be careful for the memory block data latency at the data out (check the summary tab) which may be one or two clock cycles that can be observed during the simulations.
- Your design will change multiple times during your development and your design may have multiple configurations. Be careful for the versions and configurations of your design. In order to be aware of which version to use, take notes and records.

Block Memory Generator (8.4)

[Documentation](#)
[IP Location](#)
[Switch to Defaults](#)

IP Symbol

Power Estimation

☒ Show disabled ports

```

+ AXI_SLAVE_S_AXI
+ AXILite_SLAVE_S_AXI
- BRAM_PORTA
  ▶ addr[13:0]
  ▶ clka
  ▶ dina[11:0]
  ▶ douta[11:0]
  ▶ ena
  ▶ rsta
  ▶ wea[0:0]
- BRAM_PORTB
  ▶ addrb[13:0]
  ▶ clk
  ▶ dinb[11:0]
  ▶ doutb[11:0]
  ▶ enb
  ▶ rstb
  ▶ web[0:0]
  regcea
  regceb
  injectsbiterr
  injectdbiterr
  eccpipece
  sleep
  deepsleep
  shutdown
  
```

Component Name blk_mem_gen_0

Basic

Port A Options

Port B Options

Other Options

Summary

Memory Size

Port A Width 12 Range: 1 to 4608 (bits)

Port A Depth 16384 Range: 2 to 1048576

The Width and Depth values are used for Write Operations in Port A

Operating Mode No Chan...

Enable Port Type Always Enabled

Port A Optional Output Registers

☐ Primitives Output Register ☐ Core Output Register☐ SoftECC Input Register ☐ REGCEA Pin

Port A Output Reset Options

☐ RSTA Pin (set/reset pin) Output Reset Value (Hex) 0☐ Reset Memory Latch Reset Priority CE (Latch or Register Enable)

READ Address Change A

☐ Read Address Change A