50 channel FPGA based ADC and TDC Data Acquisition System for a Neutron Detector Experiment

Claudio Protano

Summer Internship Program
Fermi National Accelerator Laboratory

September 27, 2013
DAMIC (Dark Matter In CCDs) is a Fermilab experiment that makes use of CCDs to search for **light dark matter candidates** with masses in the order of a few GeVs.
DAMIC (Dark Matter In CCDs) is a Fermilab experiment that makes use of CCDs to search for **light dark matter candidates** with masses in the order of a few GeVs.

The new CCD technology allows us to set a **lower threshold** than other experiments have been able to do so far, making us more sensitive to lower mass dark matter particles.
We are interested in designing and constructing a **scintillator neutron detector** in order to use neutron beams to make silicon quenching factor calibrations for the DAMIC experiment (University of Notre Dame).
We are interested in designing and constructing a scintillator neutron detector in order to use neutron beams to make silicon quenching factor calibrations for the DAMIC experiment (University of Notre Dame).
My task @ Fermilab

Designing the Electronics for the DAQ and in particular:

- Programming the FPGA board using Simulink and ISE Design Suite;
- Acquiring data from the memory by means of the DOOCS interface;
- Analyzing the stored data using MATLAB.
Neutron Detector Experiment (2)

My task @ Fermilab

Designing the Electronics for the DAQ and in particular:
- Programming the FPGA board using Simulink and ISE Design Suite;
- Acquiring data from the memory by means of the DOOCS interface;
- Analyzing the stored data using MATLAB.
The equipment we worked on basically consists of:

1. an FPGA Board
2. a VMIVME-7805 Processor

They were previously used for a cavity controller (ESE LLRF controller) for the ILC project.
The equipment we worked on basically consists of:

1. an FPGA Board
Equipment

The equipment we worked on basically consists of:

1. an FPGA Board
2. a VMIVME-7805 Processor
The equipment we worked on basically consists of:

1. an FPGA Board
2. a VMIVME-7805 Processor

They were previously used for a cavity controller (ESE LLRF controller) for the ILC project.
The input real signal (centered at an IF $f_{IF} = 53.104$ MHz), is sampled at $f_s = 92.4$ MHz and digitized. The DDC then converts it to a baseband complex signal centered at zero frequency.
The input real signal (centered at an IF $f_{IF} = 53.104$ MHz), is sampled at $f_s = 92.4$ MHz and digitized. The DDC then converts it to a **baseband complex signal** centered at zero frequency.

The Direct Digital Synthesizer (DDS) generates a complex sinusoid at the chosen frequency of $f_{DDS} = 39.296$ MHz.
To select the signal component centered at zero frequency we use a **Cascaded Integrator-Comb (CIC) filter**, a digital filter made up of many stages.
To select the signal component centered at zero frequency we use a **Cascaded Integrator-Comb (CIC) filter**, a digital filter made up of many stages.

**Chosen parameters:**

- Number of stages: $N = 3$;
- Differential delay: $M = 2$;
- Sample rate change factor: $R = 16$. 
To select the signal component centered at zero frequency we use a **Cascaded Integrator-Comb (CIC) filter**, a digital filter made up of many stages.

Chosen parameters:

- Number of stages: $N = 3$;
- Differential delay: $M = 2$;
- Sample rate change factor: $R = 16$. 
**Sum block**: we are interested in a measure of the input pulse intensity, so we make the sum of the samples of $i$ and $q$. 

![Diagram of sum block and FIFO register](image-url)
**Sum block**: we are interested in a measure of the input pulse intensity, so we make the sum of the samples of \( i \) and \( q \).

**FIFO register**: a FIFO is needed to decrease the sampling frequency to 5.775 MHz, which is compatible with DOOCS processor.
Reflection is the software that lets us interact with the processor.
Reflection is the software that lets us interact with the processor.

Through DOOCS interface we can:

- Check clock and trigger status;
- Modify delay and width of the configurable windows;
- Look at channel outputs that constantly update;
- Acquire the data and save them into a text file.
We wrote several MATLAB scripts and functions to handle the acquired data and in particular to:

```matlab
function [] = histograms( data_i2, data_q2, data_i3, data_q3 )

complex2 = complex(data_i2, data_q2); % calculates amplitude and phase for channel 2
ampl2 = abs(complex2);
phase2 = angle(complex2);

complex3 = complex(data_i3, data_q3); % calculates amplitude and phase for channel 3
ampl3 = abs(complex3);
phase3 = angle(complex3);

phase_diff = phase3 - phase2; % calculates phase difference between the two channels

figure;
s(1) = subplot(3,1,1);
s(2) = subplot(3,1,2);
s(3) = subplot(3,1,3);

hist(s(1),ampl2,300); % plots the histograms for ampl2
title(s(1), 'Amplitude (channel 2)');
xlabel(s(1), 'points');
ylabel(s(1), 'repetitions');

hist(s(2),ampl3,300); % plots the histograms for ampl3
title(s(2), 'Amplitude (channel 3)');
xlabel(s(2), 'points');
ylabel(s(2), 'repetitions');

[q, h] = hist(s(3),phase_diff,300); % plots the histograms for phase_diff after rescaling the x axis in the basis of the TDC
m = 0.31002;
h_scaled = h / m;
stairs(h_scaled, q);
xlim([-10 10]);
title(s(3), 'Phase difference');
xlabel(s(3), 'Xs');
ylabel(s(3), 'repetitions');
end
```
We wrote several MATLAB scripts and functions to handle the acquired data and in particular to:

- Filter the samples, discarding the corrupted or double ones;

```matlab
function [ ] = histograms( data_i2, data_q2, data_i3, data_q3 )

complex2 = complex(data_i2, data_q2); % calculates amplitude and phase for channel 2
ampl2 = abs(complex2);
phase2 = angle(complex2);

complex3 = complex(data_i3, data_q3); % calculates amplitude and phase for channel 3
ampl3 = abs(complex3);
phase3 = angle(complex3);

phase_diff = phase3 - phase2; % calculates phase difference between the two channels

figure;
s(1) = subplot(3,1,1);
s(2) = subplot(3,1,2);
s(3) = subplot(3,1,3);

hist(s(1), ampl2, 300); % plots the histograms for ampl
xlabel(s(1), 'Amplitude (channel 2)');
ylabel(s(1), 'repetitions');

hist(s(2), ampl3, 300); % plots the histograms for ampl
xlabel(s(2), 'Amplitude (channel 3)');
ylabel(s(2), 'repetitions');

[Q, H] = histograms(s(3), phase_diff, 300);
% plots the histograms for phase_diff after rescaling the x axis on the basis of the TDC
m = 0.31002;
h_scaled = H / m;
stairs(h_scaled, Q);
xlim([-10 10]);
title(s(3), 'Phase difference');
ylabel(s(3), 'Hz');
ylabel(s(3), 'repetitions');
end
```
We wrote several MATLAB scripts and functions to handle the acquired data and in particular to:

- Filter the samples, discarding the corrupted or double ones;
- Calculate the amplitude and the phase of the signal starting from $i$ and $q$;
We wrote several MATLAB scripts and functions to handle the acquired data and in particular to:

- Filter the samples, discarding the corrupted or double ones;
- Calculate the amplitude and the phase of the signal starting from $i$ and $q$;
- Plot them by means of **histogram bars** to compare the results with the expected behavior.
A **TDC** is a device commonly used to measure a time interval and convert it into a digital output.

We took several measurements on two channels using different input delays (0 to 7 ns) and plotted the difference in phase between them, obtaining a linear relationship as expected.
Time-to-Digital Converter (TDC)

A **TDC** is a device commonly used to measure a time interval and convert it into a digital output.

We took several measurements on two channels using different input delays (0 to 7 ns) and plotted the difference in phase between them, obtaining a **linear relationship** as expected.
We tested the Data Acquisition and Analysis system at the Fermilab Silicon Detector Facility (SiDet) using scintillators and Photo Multiplier Tubes (PMTs) to detect muons.
We tested the Data Acquisition and Analysis system at the Fermilab Silicon Detector Facility (SiDet) using scintillators and Photo Multiplier Tubes (PMTs) to detect muons.

We chose muons for the test because there is plenty of them and because they are safer.
We took measurements on two channels, using the signals coming from the PMTs as inputs. The **trigger** was generated by an AND port between two PMTs outputs.
We took measurements on two channels, using the signals coming from the PMTs as inputs. The trigger was generated by an AND port between two PMTs outputs.

Amplitudes of channels 2 and 3 and phase difference (zero delay and input voltage of -1900 V)

FWHM $\approx 1.5$ ns
Amplitudes of channels 2 (light blue) and 3 (dark blue) with input voltages of -1900 V and -1950 V
Amplitudes of channels 2 (light blue) and 3 (dark blue) with input voltages of -1900 V and -1950 V

Phase difference with zero delay (blue) and 8 ns delay (red) between the two channels