PROJECT REPORT ON
ADC, DMA AND DSP IMPLEMENTATION

By

SIDDHANT JAJOO
SATYA MEHTA

UNDER THE GUIDANCE OF:

PROFESSOR KEVIN GROSS,
UNIVERSITY OF COLORADO BOULDER

DATED:12/12/2018.

TABLE OF CONTENTS

DOCUMENTATION

BLOCK DIAGRAM

REPORT QUESTIONS

APPENDIX

DOCUMENTATION

The following are the different modules used in the implementation of the Project

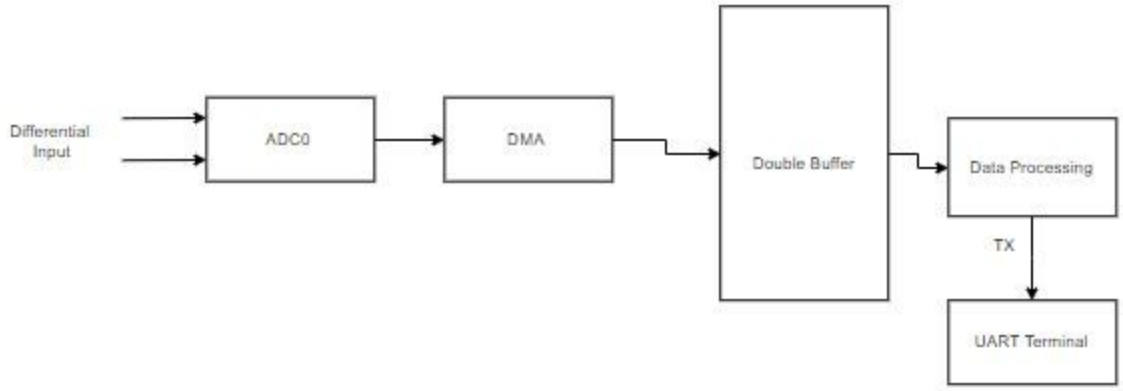
1. Analog to Digital Converter configuration and features.
    a. Differential Input enabled.
    b. 16 bit resolution.
    c. DMA enabled.
    d. Long Sampling enabled.
2. Double Buffer
    a. 128 * 16 bit Buffer.
    b. Two headers used for memory processing.

3. DMA configuration and features.
    a. Source register - ADC output register.
    b. Destination register - Array pointers.
    c. Load byte count register to 128 bytes.
    d. DMA interrupt handler called every 128 bytes transfer.

. .

4. Application.
    a. Process data in main function to calculate peak values and logarithmic values of sampled data.
    b. Send the peak value, ADC data and its corresponding Log values to UART transmitter.

BLOCK DIAGRAM



REPORT QUESTIONS

1.What is the behavior of the system when no input signal is applied?

Ans. When no signal is applied then the ADC gives the digital output of the noise which is present at its differential input. In our case we get some noise value at the output which comes up in the range of -3000 to -6000.

2. What source and destination DMA read and write size did you choose and why?

Ans. The source and destination DMA read and write size is kept as 2 bytes. The buffer implemented is of int16_t data type and also the ADC configured is of 16 bits which writes the ADC0_RA register to 16 bit output digital data. Hence, it will be useful and optimized way to keep the source and destination write size to 16 bits. If we keep the destination write to size to more than 16 bit we have to make a buffer of int32_t type which will reserve more memory and fragmentation will also increase.

3.What is the consequence of a lengthy or delayed interrupt service for the data retrieved by the DMA controller?

Ans. By using a lengthy or delayed interrupt service we get enough time to process the data in the main function. This helps in the case where a single buffer is used.

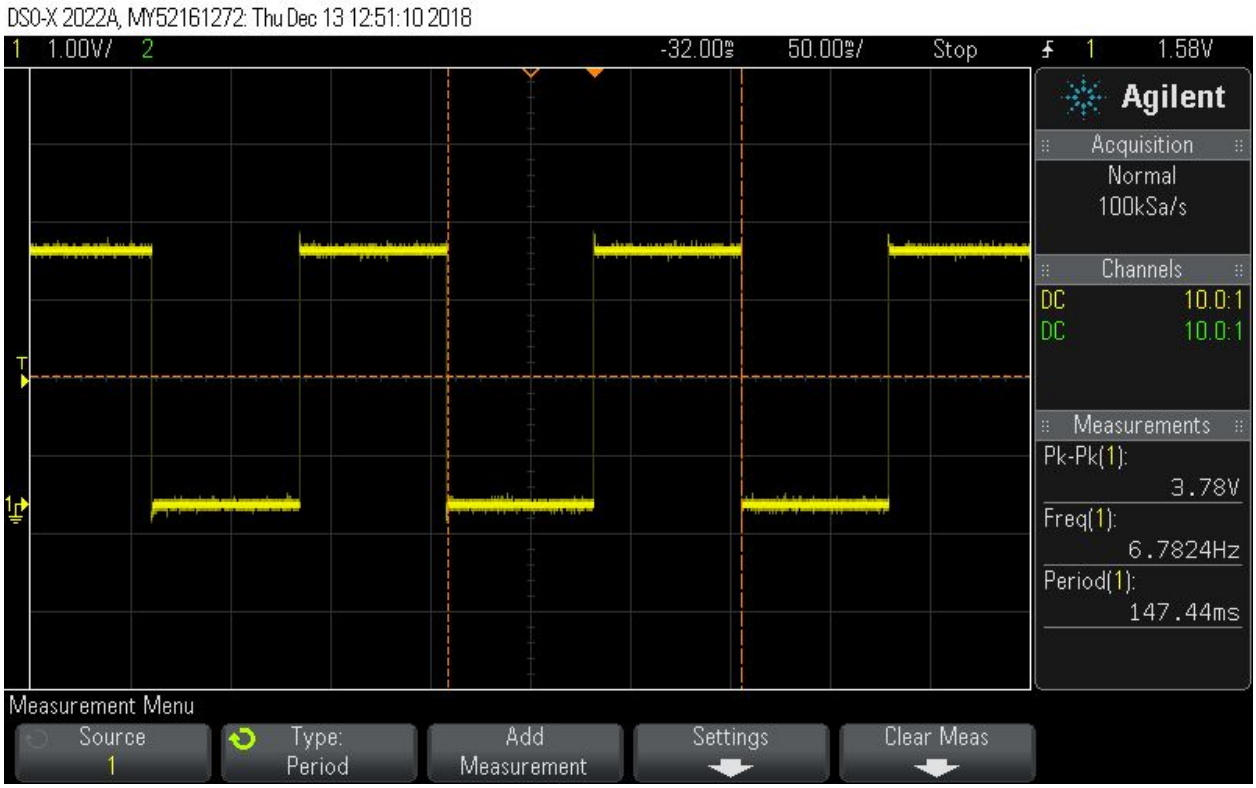
4. What effect does changing the buffer size have on system processing efficiency?

Ans. By reducing the buffer size the DMA interrupt will come more frequently and we have to service it more frequently. This will give us less to execute the main function and frequent dma interrupts might stall the functions to be executed in the main loop. If we increase the buffer size then the frequency of interrupt will decrease and the processor will be executing the main function most of the time.

5. What other effect(s) does changing the buffer size have on the system?

Ans. By reducing the buffer size the DMA interrupt will come more frequently and we have to service it more frequently. This will lead to data overwrite and we may lose some data. If we increase the buffer size then the frequency of interrupt will decrease and the processor will be executing the main function most of the time.

OSCILLOSCOPE OUTPUT:



APPENDIX

This report has comprehensively covered all the modules , functions , their description, the different modes implemented in UART and answers to relevant questions.

This project covers implementation of 16 bit ADC in differential mode. The ADC configuration includes selection of proper clock and bits in the ADC register to run it at more than 8 KHz sampling frequency. During the initial phase of the project the ADC reading was obtained using Adc_read function to check whether the ADC is properly configured or not. The next step was to configure the ADC to enable the DMA request. The output of the ADC module is directly transferred to the buffer using DMA.

The DMA configuration includes clocks, selecting the source and destination registers and selecting DMA write sizes. The data that is written using DMA is of type int16_t. The DMA byte counter is set to 128 bytes. Hence, the interrupt will be generated after 128 bytes of data write. A double buffer has also been implemented where at a given time a part of the buffer will be used by the processor and a part by the DMA to load the ADC data.

The buffer data is processed to calculate peak, logarithmic values and displayed on the terminal using UART.