CMPE 663 Project 4

ADC & DAC

Andrei Tumbar

Instructor: Wolfe
TA: Nitin Borhade

By submitting this report, you attest that you neither have given nor have received any assistance (including writing, collecting data, plotting figures, tables or graphs, or using previous student reports as a reference), and you further acknowledge that giving or receiving such assistance will result in a failing grade for this course.

Your Signature: ______________________________

Analysis/Design

This project looked at the wave degeneration created from feeding the DAC a 1Hz triangular wave signal read from the ADC and connecting the DAC output pin to the ADC input pin. Using an oscilloscope connected in parallel, the live signal degeneration can be seen clearly. The sharp edges of the original wave begin to dengerate as the error produced by the DAC, ADC and line impedance compounds after every iteration.

Hardware Operation

The simplest design for the operation of the DAC and ADC is to write directly to the DAC hardware output register and trigger ADC conversion via software. To make the timing work properly, `TIM2` was used to interrupt at a constant known rate for each of the DAC samples. The DAC is designed to output to the maximum operational voltage and use the entire dynamic range of 12-bits. To generate a triangle wave of this type, 8192 sample buckets are required. The calculation to set up the timer is done below:

$$F_{sysclk} = F_{wave} \cdot N_{samples} \cdot (ARR + 1)(PSC + 1) \tag{1}$$

We have two variables and only one equation, we can therefore solve for $ARR = PSC$ to keep things simple. F_{wave} is defined at 1 Hz.

$$\begin{aligned} ARR &= PSC \quad &(2)\\ x &= ARR + 1 \quad &(3)\\ x &= \sqrt{\frac{80\,\text{MHz}}{8192 \cdot 1\,\text{Hz}}} = 98.82 \approx 99 \quad &(4)\\ ARR &= PSC = \mathbf{98} \quad &(5) \end{aligned}$$

Now that the `TIM2` interrupt will trigger at the desired rate, an interrupt handler is written to perform two main functions: write the current sample to the DAC, trigger the ADC conversion.

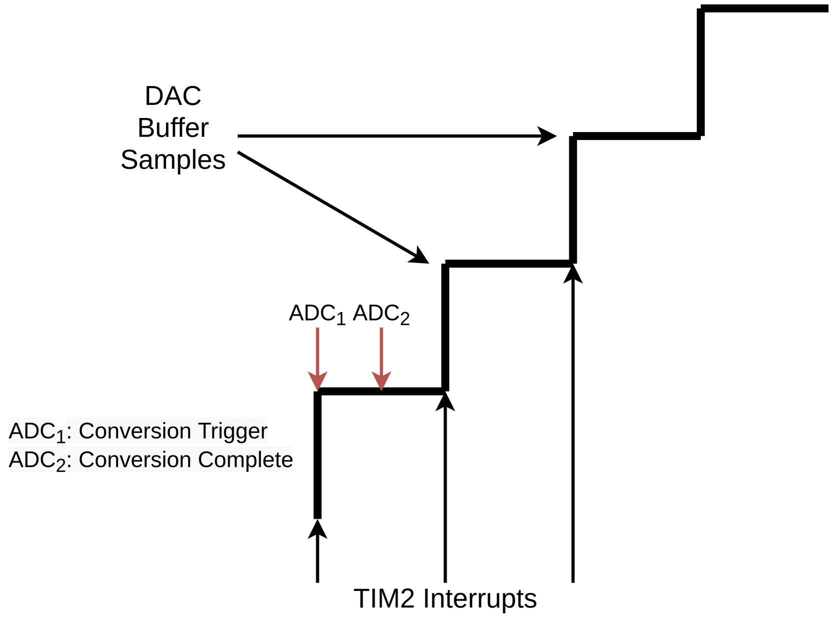

Figure 1: DAC & ADC operation with `TIM2` interrupt

As can be seen in Figure 1, the `TIM2` interrupt will write to the DAC output register. The ADC conversion trigger is also signaled during this interrupt. When the ADC has finished sampling and converting the read value, it will trigger an interrupt of its own, which allows the program to read the value from the peripheral into the ADC buffer.

Metrics & Peripheral Buffers

Metrics will take a significant amount of time to compute. This is due to the need to sort the data to compute the median data point. This means that the computation should not be performed inside the timer interrupt. When a cycle completes, the buffers pointed to by the DAC and ADC, are rotated. The two old buffers are queued to compute metrics. Once the `TIM2` interrupt finishes, the superloop will see the queued buffers and perform the computation.

Because the ADC will write to its buffer immediately, at least three buffers are required when computing metrics. The DAC peripheral and metrics can both read from the same buffer at the same time.

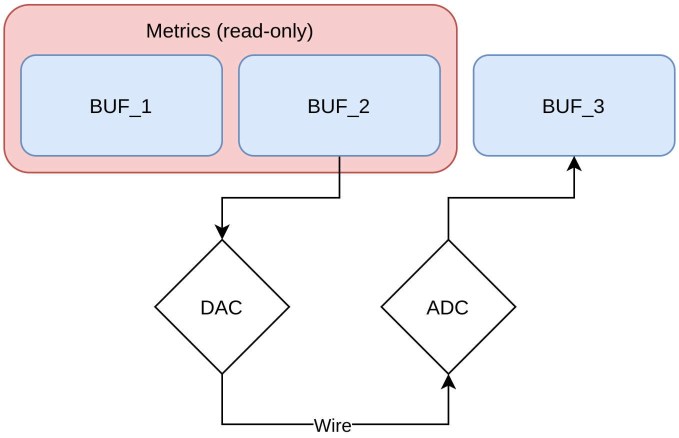

Figure 2: Buffer design in combination with metrics

Figure 2 shows an example of the buffer usage during a DAC/ADC cycle and metrics computation. The computation is reading from the buffer that was previously written to by the ADC (`BUF_2`) as well as comparing this buffer to the the ADC output from two cycles prior (`BUF_1`). An important thing to note is that nothing is reading from the ADC buffer while the peripheral is writing to it.

Test Plan

Testing the code was fairly straighforward. A wire was connected from the DAC output pin, `PA4` to the ADC input pin `PA0`. An oscilloscope was connect in parallel to show the voltage on the line in realtime.

Project Results

To check the functionality of the program, the data from the first and tenth iterations were saved and dumped to the UART. This data was then used to generate a graph with three plots: reference, first and tenth iteration lines. The reference plot is meant to show the ideal wave that was initially passed to the DAC.

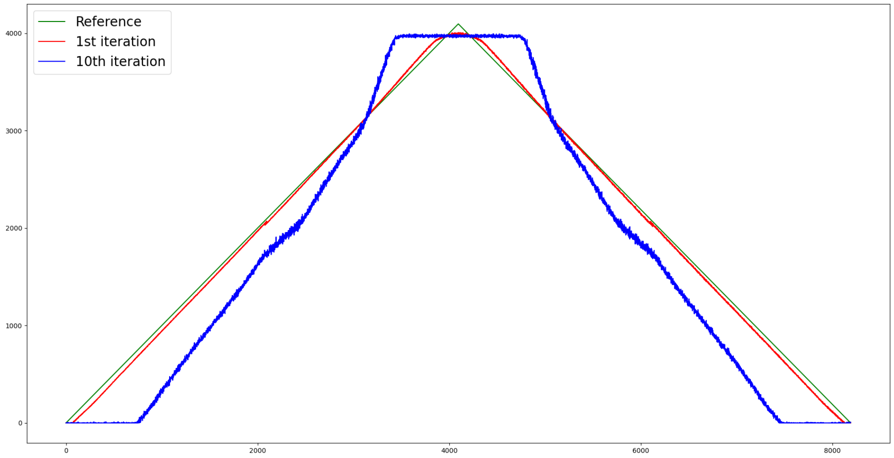

Figure 3: ADC capture comparison of 1 vs 10 iterations

Looking at Figure 3 we can see the degredation we would expect. The corner at top the will flatten and the shape of the original triangle will begin to flatten. After a single iteration, the shape is mostly kept intact when comparing to the reference curve.

An interesting result of the degradation is that the start and end of the wave is nulled to zero. This means when connecting to the oscilloscope, the wave does not look like the tails of each period are aligned. However, we can be sure that the timing is correct because the center of period is aligned for the reference, first and tenth iterations meaning the ADC is filling the buffers properly.

Lessons Learned

This project explored the operation of the DAC and ADC. It also showed how the error of the DAC and ADC can compound very quickly. To operate the DAC, a timer interrupt was used to write the sample to output register. This interrupt also triggers the ADC to begin conversion. An ADC interrupt was used to copy the value at the end of conversion the ADC respective buffer. Smart handling of the buffers was required to make sure that metrics of each iteration could be calculated while the DAC and ADC were running.