Lab 2

Kyle Jeffrey

Abstract—The second lab explores using DMA.

1 Introduction

Direct Memory Access or DMA is a hardware implementation of data transfer. For operations that use repeated data movement, DMA can handle this with hardware. This lab explores using the PSoC 6's DMA Component

2 Part 1: Memory to Memory Block Transfer

In this part the DMA component is used to transfer data from one 4096 byte array to another. The next three sections transfer the data and manipulate them before reaching the destination. All three subsections use the same component design, and the top level design has no wire connections.

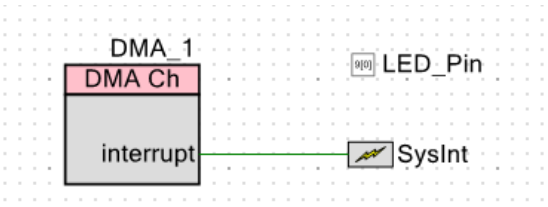

Fig. 1. Component Design

2.1 Data Replication

In this part, the data from the source to the destination array is exact. To setup the DMA, the descriptor settings must be changed for transfer. In this part, it is a one to one transfer. The DMA uses a Y loop, X loop structure, where x is the interior loop. This part uses 4 descriptors chained together.

Descriptor	
Descriptor Name	**Descriptor_1**
Trigger output	**Trigger on completion of entire descriptor chain**
Interrupt	**Trigger on completion of entire descriptor chain**
Chain to descriptor	**Descriptor_2**
Channel state on completion	Enable
Input trigger options	
Trigger input type	**Entire descriptor chain per trigger**
Trigger deactivation and retriggering	Retrigger immediately (pulse trigger)
Transfer setting	
Data element size	**Byte**
Source and destination transfer width	**Byte to Byte**
X loop transfer	
Number of data elements to transfer	**4**
Source increment every cycle by	1
Destination increment every cycle by	1
Y loop transfer	
Number of X-loops to execute	**256**
Source increment every cycle by	**4**
Destination increment every cycle by	**4**

Fig. 2. Part 1a Descriptor settings

2.2 Data Copy With Endian Conversion

In this part, the Endianess of the source to destination array is switched, meaning that the most significant byte switches with the least significant byte. The descriptor is set as follows:

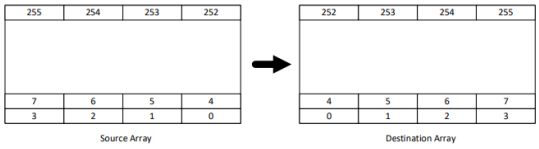

Fig. 3. Part 1b Data Diagram

Descriptor	
Descriptor Name	**Descriptor_1**
Trigger output	**Trigger on completion of entire descriptor chain**
Interrupt	**Trigger on completion of entire descriptor chain**
Chain to descriptor	**Descriptor_2**
Channel state on completion	Enable
Input trigger options	
Trigger input type	**Entire descriptor chain per trigger**
Trigger deactivation and retriggering	Retrigger immediately (pulse trigger)
Transfer setting	
Data element size	**Byte**
Source and destination transfer width	**Byte to Byte**
X loop transfer	
Number of data elements to transfer	**4**
Source increment every cycle by	1
Destination increment every cycle by	**-1**
Y loop transfer	
Number of X-loops to execute	**256**
Source increment every cycle by	**4**
Destination increment every cycle by	**4**

Fig. 4. Part 1b Descriptor

2.3 Data Copy with Byte Level Replication

In this part, the DMA copies the data four times. To achieve that replication, the descriptor

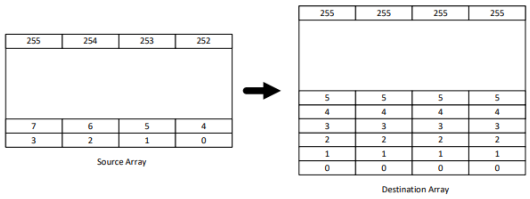

Fig. 5. Part 1c Data Transfer

settings were set as:

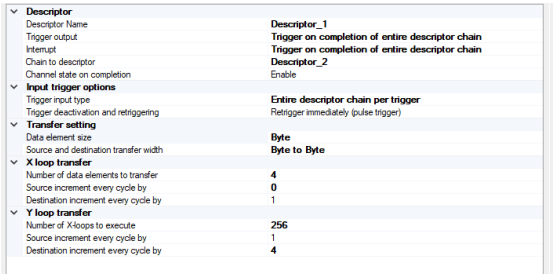

Fig. 6. Part 1c Descriptor

3 Part 2: Measuring Frequency of an Analog Waveform

In this part, the PSoC will read an analog signal input, store the data in buffers and then process the data to determine the frequency of the analog signal. The UART is used for terminal debugging and the I2C is used to print to the LCD screen.

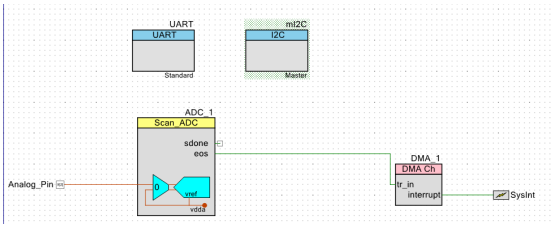

Fig. 7. Component Design

3.1 ADC and Ping Pong Buffers

The ADC uses an EOS signal to signify End of Scan and to signal to the DMA to begin transferring. To process the data though, a filled array needs to not be manipulated while processing. To do this, a ping-pong buffer is used, i.e. two arrays that have flags set for when they are ready for processing and then the other array is filled.

3.2 Caluclating Frequency

To calculate the frequency, an array was used to detect edges. The first rising edge is detected

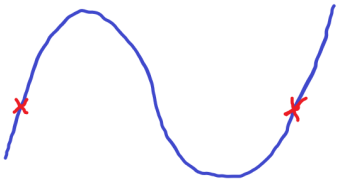

Fig. 8. Signal Sketch

by finding a value in the array where array[i] < array[i+1]. That value is stored and then the array continues until it finds another rising edge where the $|array[i]| < threshold$. This should give the period of the array.

4 Conclusion

This lab explored using DMA as a technique for data transfer. As shown in the last part, using an ADC, transferring data from the ADC would require far more software than using a DMA.

Kyle Jeffrey is a Senior Robotics Engineering Student at 6the University of California Santa Cruz. He is the Secretary of the Engineering Fraternity Tau Beta Pi and the lead Hardware Engineer at the on campus startup Yektasonics.