

# Vivado AXI Timer and Interrupts

Leonardo Aviles, [tue51778@temple.edu](mailto:tue51778@temple.edu)

**Abstract**—This project that will be an introduction to the Vivado AXI Timer and a continuation of Interrupts. In exercise 2D the tutorial continues to use the project from 2A-2C, which were the button interrupts. In 2D the Zynq book adds a AXI Timer IP block as a additional interrupt and concat IP block which will allow two interrupts to be connected.

## I. INTRODUCTION

IN the Vivado AXI Timer and Interrupt Lab the objective was to increment the number of times the AXI timer will interrupt the process before it increments the count autonomously. In order to do this the file `interrupt_controller_tut_2D.c` was modified.

The specifications of the project are as follows.

- Reconfigure the project files so button or the right most button would not only reset the count but also increment the number of times the AXI Timer will occur before it increase the count.
- A method was found that would allow the default expiration time to be increased. The method that can be used is using a while loop that would have nothing in it.

To complete the project, code needs to be manipulated in the following way.

- A global variable called `CountVal` was created which will change depending on the reset button.

## II. DISCUSSION

*A. interrupt\_controller\_tut\_2D.c: "Incrementing the number of times AXI will interrupt the process"*

In order to achieve this goal a global value was created. The code used was `static int CountVal = 3;`, this was the default value of the number of times the interrupt process will happen. In order to increment this value the code `CountVal++` was placed inside the else statement that would reset the values of the `LED_Data`. The variable `CountVal` will be places in the if statement `if(tmr_count == CountVal)` this statement will only occur when the timer expires and the `tmr_counter` reaches the value set in `CountVal`.

In order to add an AXI Timer and make it a interrupt two IP blocks are needed. The first is the AXI Timer block, which will be the actual timer. Clicking on this block will open a menu that will allow the programmer to enable interrupts. The second IP Block was the concat block. This IP blocked allowed us to connect two interrupts to the zynq processing system.

The default expiration time of the AXI Timer is one second if the process is interrupted three times. In order to change the

expiration time the line in the code `XTmrCtr_SetResetValue(&TMRInst, 0, TMR_LOAD);` can be change. The variable `TMR_LOAD` is the initial value of the timer and can be changed based on the users input. This value by default is 32 bits wide based on the Xilinx manual. A method that can be used in order to increase the timer is using an empty while loop. The problem with is the amount of time will change depending on the processor speed or system the program is running on.

## III. CONCLUSION

The project was implemented into the zybo board but the results were not what they were expected. Based on the code it was expected that increasing the number of times the interrupt process the time between LED increments would increase but instead the time between the LED increments stayed the same. Regardless of the number of times the reset button was pressed the time would not increase or decrease.

## IV. APPENDIX

Figure 1: Shows the block diagram created for the lab.

