**10/25/2020 –** Looked int AXI DMA accessing. I think this should be the method to input data to a design. Followed this guide: <http://www.fpgadeveloper.com/2014/08/using-the-axi-dma-in-vivado.html>

I was also looking into the guide provided by Kasem: <https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug1393-vitis-application-acceleration.pdf> (page 518), I believe I need to setup a PetaLinux installation.

Using this guide to setup PetaLinux, got to page 15 and having an error opening settings source <https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug1144-petalinux-tools-reference-guide.pdf>

**10/26/2020 –** Continued to look at the guide Kasem sent. Did not spend much time but got petalinux installed. Vivado would not export the xsa file so I went to sleep.

**10/28/2020 –** Found out the ubuntu installation I had was not supported for vitis/petalinux. Changed to Ubuntu 14.04.1 LTS. Installed Vitis, Vivado, PetaLinux. Started a document to cumulate reference papers on accelerated hardware designs. Finally got PetaLinux to open but when I tried to create the project I got a failure. For some reason Vitis does not want to launch due to some eclipse error. I put a posting to the Zynq forum here <https://discuss.pynq.io/t/image-processing-acceleration/1955>

**10/30/2020 –** After reading the forum post I made, it seems in Vivado, you can easily integrate your design with the AXI protocol. I started this process by testing a simple FIFO design. I downloaded the ISO for the Pynq SD card image.

Example of hardware software link using AXI: <https://github.com/Xilinx/PYNQ-HelloWorld>

Video for simple tutorial on AXI: <https://www.youtube.com/watch?v=8hzzVhPw6uw&t=516s>

Video that creates a buffer and tests it out: <https://www.youtube.com/watch?v=R8MSpEU7UKE&t=613s>

Debugging AXI Streams: <https://zipcpu.com/dsp/2020/04/20/axil2axis.html>

AXI Signal Description: <http://www.gstitt.ece.ufl.edu/courses/fall15/eel4720_5721/labs/refs/AXI4_specification.pdf>

I started working on a simple AXI interface device that will take in data over AXI, then update an LED blinker counter. Ran into an issue where Vivado say I was driving the signals incorrectly. I made a detailed forum post here: <https://discuss.pynq.io/t/image-processing-acceleration/1955/8?u=monkeyboyfr3sh>

**10/31/2020 –** I fixed my LED\_timer design to take in data via AXI and update a period/duty cycle. This allows for me to create a “breathing LED” just for practice. I then started updating my design to utilize the AXI stream interface for getting pixel values via DMA and an AXI-lite port to set control information about the convolution.

**11/4/2020 –** I continued work on the DMA controller device that will stream in pixels. I got pretty far with implementing logic to interface with the device but ran into an issue when trying to actually complete a DMA stream. I believe the base is there, but I need to find some resources on interacting with the DMA device properly.

**12/1/2020 –** Previously I was trying to implement the FIFO inside my own device. I think it would be more effective to just use the FIFO built into Vivado. I began prototyping with a design that should invert images that are fed to it.

Follow this tutorial series:

Video 1: <https://www.youtube.com/watch?v=Xkpu8BXi3aI&t=947s>

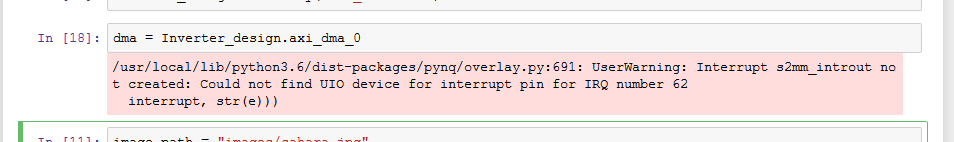
Video 2: <https://www.youtube.com/watch?v=chs5mdwMchQ>

Video 3: <https://www.youtube.com/watch?v=5MCkjKhn1DM>

Video 4: <https://www.youtube.com/watch?v=x3KyWuhGmJg&t=1046s>

Since this design is working with SDK, I need to port this to work with the Jupyter notebook.

Tried some more to create a simple BD to pass data with DMA. For some reason when I use the FIFO IP from Vivado, I get the following error in jupyter



**12/4/2020 –** DMA FINALLY WORKS. Kinda. As of right now the only thing I can successfully do is take in an image, invert every pixel value, then print this image.



So I started implementing a new AXI controller. I was not using the register feature of AXI devices properly. When trying to edit the files a bit, Vivado kept crashing. Gonna close for today.

**12/7/2020 –** Made some good progress and can now set IP control registers as well stream data. This is a good explanation on normal AXI transactions: <https://zipcpu.com/blog/2019/04/27/axi-addr.html>

So I’ve updated my inverting code to basically take the image and check a control register if the data should be inverted or left alone. This was successful.

I tried to further develop the controller to complete use the state machine. I think it is close but still needs modifications. As of now the DMA stalls at the receiving channel. I believe this means that my tlast signal is not setting properly. Should setup some more debugging registers that while return he number of convolutions that have occurred within the device. This should be much easier now that I can successfully complete a write transaction. Will need to do a separate test on read transactions tho ☹