Hardware Description Document

Requirements

- Less than 16.67 ms for total delay
- Pass through tuser and tlast
- Syncs with output Video Timing Controller

Device Functional description

A video stream data processor that detects edges on the incoming video stream using a Sobel edge detection algorithm.

Block Diagram

Development plan

Crawl

Pass HDMI video in and buffer it out.

This stage only checks that the FIFOs work, the clock domain crossing is okay, and to monitor what happens to timing. Not worrying about TID, TUSER, TLAST, or stages.

Walk

Implement SOL, Row Manager, and a simple pass out module.

Top level interfaces

VTC Timing Controller

A Module called the VTC Timing Controller was implemented to make sure the Video timing signals are synced with the video data signals from the axis edge detection module.

cold sparing interfaces

Internal Block descriptions

All inputs and outputs are registered beside the tready signals. This is to reduce the delay caused by a clock.

SOF

This module of the design pauses processing until the start of a frame is found, on net tuser. This is done to set a known starting point for the design. It also implements some pipeline registers to keep the AXIS slave interfaces in sync.

The SOL module waits for the tuser line to go high, indicating the start of an HDMI frame. Otherwise it simply passes everything through. The output is also registered by this module and has a mux implemented. It outputs a blank frame while waiting for a new frame.

This section will be responsible for dropping frames if needed. (TBD) could do 1 out of 6 or something

Row Manager

The Row manager's purpose to allow parallel processing of various stages. It keeps rows in temporary memory but allows them to be overwritten as data is shifted in from middle stages. It is composed of a few main blocks.

1. Oversight manager: This section handles where each submodule is in the process
2. Prepper: Adds padding and stage info to the incoming data based on the oversight's orders. It converts the FIFO AXIS data to be written into the BRAM
3. Block Ram (BRAM):
4. Kernel Shifter: Reads 72 bits worth of data from the BRAM and sends it out based on the stage to the appropriate output module

Prepper

This module accepts data from FIFO1 when it is ready, and writes it into BRAM when it is allowed to. It is watched by the oversight manager when overwriting is allowed.

It adds padding TBD

Kernel Shifter

The shifter was changed to be absorbed by the row manager. This is to allow it to act on the stored memory instead of needing to have multiple rows shifted over.

The kernel shifter is responsible for managing the video frame itself.

A state machine is used in its design to handle shifting rows, pixels, and resetting after each frame.

For tuser, it sends out a high signal on the second pixel of each frame. This is because tuser goes high on the first pixel.

Tlast is pulsed on the second to last pixel of each row. This is before the end padding pixel.

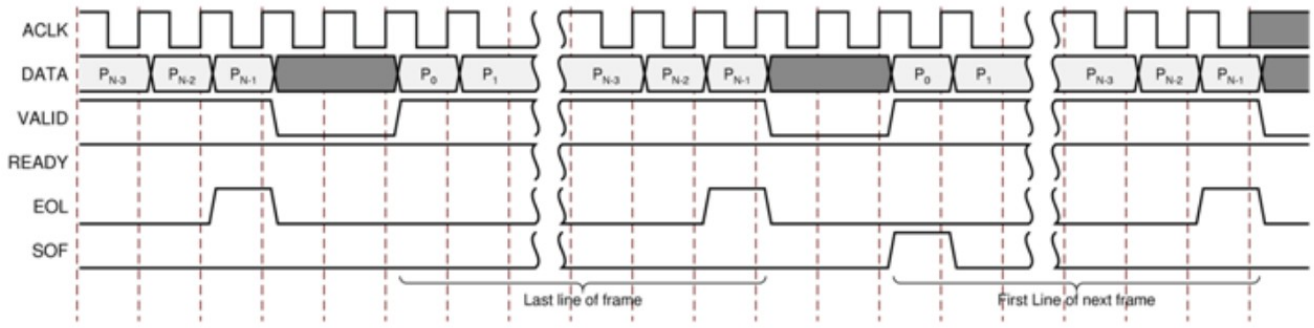

Figure 2-7: **Use of EOL and SOF Signals**

Processing Module

These are various modules that can be placed after a kernel shifter. They must use a 3x3 matrix operation.

Tuser and Tlast are passed through/pipelined with the pixel data they came in with.

FIFO2 Arbiter

This module

Reset Architecture

Clocks

This design features two clock domains, a pixel clock and a higher frequency clock.

The pixel clock is from the DVI IP block and runs at 74.25 MHz. This is a result of 60Mhz at 720p.

The fast clock is for the heavy processing done in the design. This is to reduce timing delay from HDMI in to HDMI out. Initially 400MHz was chosen, but after learning that the FIFOs can’t handle this rate I switched it to a lower 200MHz.

Clock Name	MHz

Pix_clk	74.25
Fast_clk	200

Memory

FIFO1 is used to change from the slower pixel clock to 400MHz and to allow the input data to store up if needed. This is a small buffer as its outputs are fed into the deserializer module.

FIFO2 is a small buffer for data that has been processed coming back to memory to overwrite the existing pixels. MIGHT NOT BE NEEDED

FIFO3 is used to reverse the clock change back to the pixel clock. This is a small buffer.

The FIFOs are instantiated through an AXIS FIFO XPM from Vivado.

Design for Test

Multiple stages of block diagrams and bitstreams are saved at incremental points in development like crawl, crawl w/ extra, etc. This was done to provide good reference for development and learning.

Another key features is a testing environment. This features a block diagram with a test pattern generator for video AXIS data. This entire design is made for simulation.

Estimated Power

Estimated Timing

Name	Time from input to output for 720p @ 60Hz	
(ms)	Edge detection	Top
Base HDMI no mod		
Crawl - passthrough		
Crawl – passthrough w/ SOF	2380 ns	
Walk		
Run		

Future Developments

Implement a USB to AXIS data conversion for live USB Camera data from an ESP32 camera module.