**Desarrollo**

3.2.4 Diseño del modelo de referencia en alto nivel

3.2.4.1 Implementación de algoritmos en C

3.2.4.2 Pruebas a nivel de software

3.2.4.3 Diseño de arquitecura de alto nivel

3.2.5 Implementación a nivel de hardware

3.2.5.2 Descripción de hardware en Verilog

3.2.5.3 Pruebas tipo test bench con herramientas de XIlinx

**Arquitectura del convolucionador**

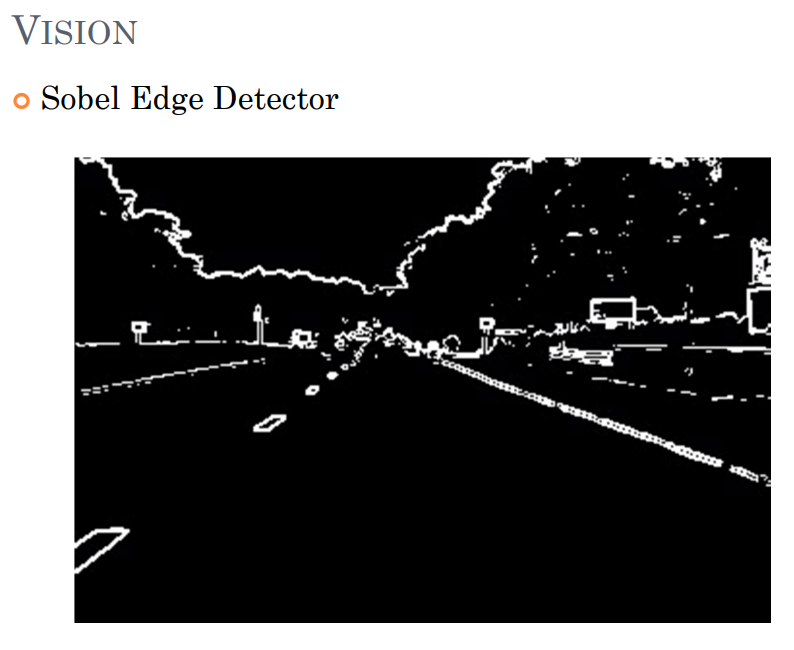
**Poner el HLA aquí**

Para resolver la convolución de dos señales, se decidió utilizar el método de Systolic Arrays el cual

Idea para el convolucionador:

* Hacer un convolucionador 2D. Que tenga una aplicación en el área de Computer Vision (especificamente para el área de self driving cars)

Por ejemplo, que sea el Sobel Edge Detector y que de entrada tenga una imagen y de salida, de otra imagen pero ya procesada. Como en la siguiente imagen:



La ventaja de todo esto, es que el procesamiento de imágenes se hace a nivel de hardware y no ha nivel de software, lo cual lo convierte en un procesamiento MUCHO más rápido.

La arquitectura es muy específica, pero al final del día, lo que se busca es un convolucionador rápido, no un convolucionador general. Es decir, cuando se diseña un acelerador de hardware, no importa que no sea general sino que sea rápido.

Esa imagen, ya procesada, se la puedes mandar a un algoritmo de state estimation o lo que sea a más alto nivel (en software) capaz de tomar decisiones.

Kung - Ciclo de diseño

Task definition: Resolver la convolución

Design: Convolucionador basado en systolic arrays - para 8, 16, 32 o 64 bits?

Implementation: For an FPGA using Verilog