Las tarjetas gráficas o GPU de Nvidia que fueron las utilizadas en esta investigación, contienen cuatro tipos de memoria denominadas como, memoria global, constante, compartida y registro.

Para esta investigación se decidió hacer uso de estas memorias mediante tres implementaciones del algorítmo de la convolución y el filtro de sobel sobre una imagen llevada a escala de grises. Es importante resaltar que en las tres implementaciones se está haciendo uso de la memoria de registro que contiene cada hilo de un stream multiprocessor, estos registros son usados cuando se declara variables o arreglos dentro de una kernel que no contengan un qualifier que indique la declaración de la variable fuera de ese contexto, como por ejemplo el identificador \_\_shared\_\_.

En cada uno de las tres implementaciones, el uso de la memoria respectiva se llevó acabo de la siguiente manera:

Para el algorítmo que utiliza memoria global, los datos de la imagen en escala de grises reside en la memoria global así como las dos máscaras del filtro de sobel, con las que se realizará la convolución, y es aquí donde se evidencia que existirá un problema de cuello de botella, ya que habrá que acceder a la información de las máscaras varias veces por cada cada pixel de la imágen, así como el acceso a los pixeles hallo al pixel que está siendo procesado, además la situación empeora si consideramos los otros hilos de los demás streaming multiprocessor que están accediendo a las máscaras.

Para el algorítmo que utiliza memoria compartida, el problema del cuello de botella en acceso a los datos se solventa en gran medida, debido a que parte de los datos de la imagen en escala de grises se llevan primero a la memoria compartida ( separada previamente dentro de la kernel), y luego las operaciones de acceso serán con respecto a ese espacio de memoria, reduciendo el acceso a la memoria global considerablemente para cada pixel y los pixeles hallo cuando estemos haciendo la convolución, las máscaras todavía residen en la memoria global, por lo que el cuello de botella en esta parte todavía existe.

Para el algoritmo que utiliza la memoria constante, las máscaras son trasladadas a la memoria constante, se sigue usando la memoria compartida para trasladar las partes de la imagen desde la memoria global, aquí se puede visualizar que todavía persiste el problema del cuello de botella, pero al tratarse de una memoria que posee un canal de transmisión de datos más ancho de alta velocidad, y como es memoria de solo lectura el delay en el acceso se reduce notablemente en comparación con el acceso en la memoria global.

El algorítmo donde se utilizó memoria global, presentó una inmensa mejoría en velocidad en comparación con la velocidad obtenida por el algorítmo “secuencial”(en donde se hizo uso de la herramienta opencv para su implementación) como se puede apreciar en la gráfica #. Podemos notar además que el algorítmo donde se utilizó memoria compartida, tubo aún una mayor mejoria en velocidad en comparación con el de la memoria global como lo muestra la gráfica#. Y el algorítmo donde se utilizó memoria constante obtuvo la mayor velocidad como lo muestra la grafica# que el de los otros dos tipos de memoria.

Entre las razones por las cuales se presentan estas mejorías en velocidad, más hayá de que estamos trabajando con un algorítmo altamente paralelizable y así haciendo uso de many-cores, se debe a la velocidad de acceso a los datos en cada tipo de memoria, de los cuatro tipos de memoria, la global es la más lenta, y es la memoria que posee mayor probabilidad de presentar un cuello de botella, ya que es la memoria que es accesada por cualquier hilo de cualquier stream multprocessor de la tarjeta, el segundo tipo de memoria vendría siendo la memoria compartida, esta es una memoria pequeña que está dentro de cada stream multiprocessor, y es solo accedida por lo hilos pertenecientes a ese stream multiprocessor, la velocidad de acceso a la memoria compartida es más rápida comparada con la memoria global, debido a que es una memoria que está más cerca a los hilos a nivel de hardware, y además solventa en gran medida el cuello de botella que se presenta en la memoria global, al llevar trozos de la imagen a esa memoria para un respectivo procesamiento, la siguiente memoria es la memoria constante, esta memoria es accedida por cualquier hilo de cualquier stream multiprocessor como la memoria global, pero es una memoria de mayor velocidad de acceso al tratarse de una memoria de un gran canal de transmisión de datos de alta velocidad y de solo lectura y por último tenemos la memoria de mayor velocidad de acceso que es el registro de cada hilo de un stream multiprocesor, y es la de mayor velocidad porque ese la más cercana en hardware al hilo.