

# Taller 5: OpenCL

Malcolm Davis, Miembro Estudiantil, IEEE  
mdavis.cr@ieee.org

**Resumen—** En este documento se encuentran las respuestas a las preguntas propuestas en el taller 5 del curso Arquitectura de Computadores II relacionadas con el paralelismo a nivel de datos de datos con CUDA.

**Index Terms—** TEC, Arquitectura De Computadores II, AMD, OpenCL, GPU.

## 1. MICRO-INVESTIGACIÓN

### 1.1. ¿Qué es OpenCL?

OpenCL, según lo describen en la página de Kronos es un estándar multi plataforma que permite la programación paralela en diferentes procesadores. [2]

### 1.2. ¿Qué es un kernel en OpenCL y cómo se define?

Los kernels en OpenCL son piezas de código que se van a ejecutar en los procesadores paralelamente. Se definen utilizando la palabra clave `__kernel`. [1]

### 1.3. ¿Cómo se diferencia entre asignación de trabajo a CPU y asignación a GPU?

Si se modifica el `CL_DEVICE_TYPE_CPU` por un `CL_DEVICE_TYPE_GPU` se puede definir el tipo de procesador que se usa.

## 2. EJEMPLO 1: HOLA MUNDO EN OPENCL

**2.1. Analice el código hello.c. A partir del análisis del código, extraiga cuáles son los pasos generales para la generación de aplicaciones utilizando OpenCL. Además, determine, ¿qué debería cambiar del código, para que el kernel sea ejecutado en un GPU en lugar de un CPU?**

Los pasos principales son:

- Declarar el Kernel
- Inicializar Dispositivos
- Declarar y reservar memoria
- Inicializar los datos en el CPU
- Transferir los datos del CPU al GPU
- Ejecutar el Kernel declarado anteriormente
- Transferir los resultados del GPU al CPU
- Liberar la Memoria

Como se mencionó en el punto anterior, `CL_DEVICE_TYPE_CPU` se cambia por `CL_DEVICE_TYPE_GPU` para ejecutarlo en GPU.

**2.2. Analice el código fuente del kernel hello.cl. A partir del análisis del código, determine ¿Qué operación se realiza con los vectores de entrada? ¿Cómo se identifica cada elemento a ser procesado en paralelo y de qué forma se realiza el procesamiento paralelo?**

Se puede observar en el kernel que la operación que se realiza sobre los vectores es una suma de vectores:

```
__kernel void vector_add(__global const int *A,
__global const int *B, __global int *C) {

    // Get the index of the current element
    //to be processed
    int i = get_global_id(0);

    // Do the operation
    C[i] = A[i] + B[i];
}
```

En la octava línea de este código se aprecia la operación.

**2.3. Realice la ejecución de la aplicación hello. ¿Qué hace finalmente la aplicación?**

Como se mencionó anteriormente, la aplicación hace una suma de vectores y muestra un hello world.

## REFERENCIAS

- [1] M. Ebersole, "What is OpenCL? - StreamHPC". StreamHPC. [Online]. Available: <https://streamhpc.com/knowledge/what-is/opencl/>. [Accessed: 23- May- 2018].
- [2] M. Ebersole, "OpenCL - The open standard for parallel programming of heterogeneous systems". The Khronos Group. [Online]. Available: <https://www.khronos.org/registry/OpenCL/sdk/1.1/docs/man/xhtml/function>. [Accessed: 23- May- 2018].