Tarea filtro FIR con CCS

La práctica consiste en la implementación de un filtro FIR, aplicando las mejoras de optimización de código en C y la comprobación de los ciclos de reloj que tarda cada versión, utilizando el Code Composer Studio. El filtro FIR consiste en un algoritmo del tipo:

$$Y(i) = \sum_{j=0}^{N-1} c(k) * x(i-j)$$

Donde i se encuentra entre 0 y M. M es el número de puntos de la señal. N es el número de coeficientes del filtro.

Tareas a realizar

Se tienen que realizar las siguientes tareas:

Implementación código

- Implementar el programa para que se pueda trabajar con un vector de datos y un vector de coeficientes cuyo tamaño se pueda modificar fácilmente. Se puede realizar un programa de prueba con pocos datos para comprobar la corrección del algoritmo pero, para la práctica, el tamaño mínimo debe ser de 1000 valores de datos y 50 coeficientes.
- 2. En la primera versión el bucle de cálculo estará implementado en el main de forma desenrrollada.
- 3. En la segunda versión, el bucle de cálculo en forma desenrrollada se colocará en una función.
- 4. En la tercera versión el bucle de cálculo estará en una función separada implementado como un doble bucle.
- 5. En la cuarta versión, los parámetros se pasarán mediante punteros.
- 6. Por último hacemos la lectura de los valores de los coeficientes (archivo coeficientes.csv) y de los valores del archivo musica4.csv. Son 21 coeficientes. Del archivo musica4.csv no hace falta obtener todos los valores, porque hablamos de un fichero muy grande. Se puede jugar con el tamaño de los datos. El mínimo será 3500.

Optimización

- 1. Se llevará a cabo la depuración hasta asegurarse de que funciona el código correctamente.
- Se debe llevar a cabo un profiling del código que nos dé información de los ciclos de reloj que tarda en ejecutarse cada uno de los pasos programados anteriormente.
- 3. Se comenzará por la opción sin ninguna optimización y se apuntarán los ciclos de reloj para cada caso.
- 4. A continuación se añadirá Restrict a los punteros de los vectores y se comprobará nuevamente los ciclos de reloj.
- 5. Se utilizarán los intrinsics que el alumno crea convenientes para acelerar dicho código y se volverá a comprobar los ciclos de reloj.
- 6. Con la situación que menos ciclos de reloj utilice se comienzan a aplicar los diferentes niveles de optimización.
- 7. Aplicar los pragmas que se crean necesarios y comprobar de nuevo los ciclos de reloj.

Documentación a entregar

Se creará un fichero ZIP con la siguiente denominación de NOMBRE_DSP_2020 donde NOMBRE representa el nombre y primer apellido del alumno para poder identificarlo. El fichero debe contener:

- Los proyectos sin ejecutables para poder ser compilados por el profesor.
- Un informe, con una buena estructura, es decir, introducción, desarrollo y conclusión, que indique.
 - Características del proyecto
 - Explicación del código C.
 - Tabla comparativas (también se puede complementar con gráficas) de ciclos y tiempo para cada una de las optimizaciones llevadas a cabo.
 - Valoración personal.