



**CE-5401: Proyecto de diseño de ingeniería en
computadores**

Manual técnico

**Proyecto: Implementación de un módulo FFT en FPGA, cómo
módulo acelerador de un computador.**

Estudiante:

Andrés Artavia López - 2017075875

Supervisor:

Luis Alberto Chavarría

Docente: Gustavo Cubas

I semestre 2022

1. En primer lugar los archivos de transformada que se quieran usar, se colocan en la raíz de la carpeta
2. Una vez elegido el archivo, abrir Vivado y abrir el proyecto dentro de la carpeta proyectoVivado.
3. Cuando abre Vivado, se debe asegurar que los operandos usados en el archivo de transformada correspondan a los parámetros del módulo de control. Para eso abrir el *block design*, abrir el bloque de control y cambiar los parámetros similar a cómo se observa en la imagen, para una ventana de 128 y operandos de ancho 32.



4. Una vez configurado esto, hacer el proceso de síntesis e implementación.
5. En caso de querer cambiar la configuración del módulo, se debe referir al archivo multGen.py.
 - a. Cambiar en la línea 10 el ancho de la ventana
 - b. Cambiar en línea 11 el tamaño del operando, cabe destacar que un cambio en este último requiere cambios de diseño en el sistema de control. En la sección de toma y envío de operandos, para el estado de `bufferState` y `returnState`

```

window = 128
operandWidth = 32

```

- c. Si se quiere cambiar el formato del punto fijo, cambiar la línea 14.

```

vectorSize = operandWidth
qformat = [4, 12]
resetMat = [1

```

- d. Una vez hecho el cambio, correr el programa y en caso de querer observar una ejecución con cada pulso de reloj, escribir y a la pregunta de “do tests?”. Caso contrario la generación procederá.
 - e. Es importante considerar que se requiere de gran cantidad de memoria, alrededor de 40GB para modelos con puntos de 512 o más. También el tiempo sube con la cantidad de tiempo considerablemente, para 512 o más se requiere con una estación de trabajo de gama media, 12 horas.
6. Para hacer pruebas simples se puede usar el archivo cmdTester.py, donde manualmente se pueden enviar datos y comandos de control usando la consola.
7. En caso de querer realizar pruebas a la aplicación de usuario, referirse al archivo hwCom.py donde en la clase Config está todo lo que se puede configurar del sistema. Asegurarse que los parámetros correspondan a los del modelo implementado.

```
13 class Config:
14     window = 128
15     fileMaxSize = 1000000000
16     fileMax = 10
17     operWidth = 32
18     qFormat = [4, 12]
19     dictCmd = {'start': 1, 'stop': 2, 'restart': 3, 'ready': 4,
20               'nextRe': 5, 'nextIm': 6, 'give': 7, 'resume': 8}
21     devPath = "/dev/xdma0_user"
22     env = "proc"
23     verbose = False
24
```

8. En caso de querer cambiar el modelo, se utilizará la variable operWidth para el ancho de los operandos, window para el tamaño de ventana y qFormat en caso de cambiar el formato del punto fijo.
9. Si se quiere cambiar los parámetros para crear archivos, se puede modificar el tamaño de cada archivo en líneas con la variable, fileMaxSize.
10. Se puede también modificar la cantidad de archivos por sesión con la variable fileMax.
11. En caso de querer observar prints en cada etapa, se puede cambiar la bandera de verbose a True.
12. El sistema se implementó y validó con un reloj de 100MHz.