HARDWARE LIBRE DE PROCESAMIENTO DIGITAL DE SEÑALES PARA SISTEMAS EN UN SOLO CHIP BASADOS EN EL BUS WISHBONE

Manual de Usuario

Este manual muestra una descripción de los Core DSP (Filtro de respuesta finita al impulso FIR, Filtro de respuesta infinita al impulso IIR y la transformada rápida de Fourier (FFT)); Ejemplo para cada Core de su instanciación del hardware en lenguaje Verilog. Y ejemplo su acceso desde lenguaje C.

Contenido

I. Core FIR

A. Descripción del Bloque

Este Core realiza la función de un filtro de respuesta finita al impulso (FIR), la estructura de implementación que se utiliza para este filtro es la tipo II transpuesta o también llamada tipo III, cuenta con una interfaz de comunicación bajo las especificaciones del Bus Wishbone. Este Core cuenta con la facilidad de configurar el ancho de palabras en el momento de la instanciación; tanto de la señal de entrada, como de salida, la cantidad de coeficientes, el ancho de palabra de los coeficientes, el ancho de palabra del valor de cuantización (Q), y la cantidad de bits de crecimiento para que no haya desborde o sobre flujo, este último es importante ya que dependiendo de la cantidad de bits que se instancien las señales internas van a tener el ancho de palabra más los bits de crecimiento; Las demás entradas tipo lógicas corresponden al reloj (CLK), señal que proviene de un reloj global haciendo que el módulo trabaje sincrónicamente; una señal de reinicio (*RESET*); una de limpieza (*CLEAR*); y finalmente una señal de habilitación (*ENABLE*) que hace que el filtro comience el proceso de filtrado.

Los coeficientes del filtro se obtienen mediante un script hecho en Matlab, el cual crea un archivo punto H (*.h) con formato de lenguaje C donde se guardan estos en un vector.

B. Descripción de Registros internos y su configuración

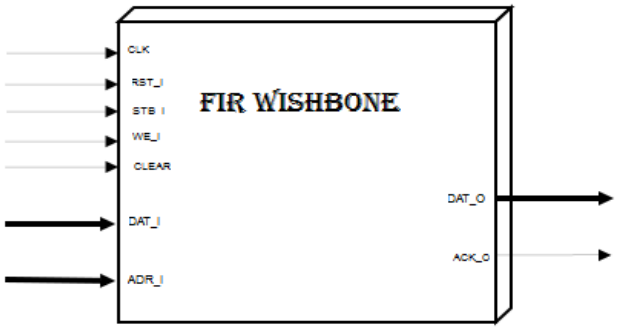
Este Core cuenta con registros internos para su buen funcionamiento, estos registros los vistos en la tabla, para su acceso es necesario conocer la dirección base del modulo y a raíz de esta se accede a los registros sumándole la dirección que se ve en la tabla, un ejemplo de acceso a los registros se mostrara en la parte de acceso mediante lenguaje C.

Para escribir cada uno de los registros se hace mediante el bus de direcciones, situando el valor de la dirección base del modulo mas el valor de cada registro en él. Los registro se escriben solo accediendo a ellos sin la necesidad de enviarles un valor especifico y estos se escriben para: el registro de control para indicarle al modulo que comience el filtrado; el registro de datos para indicar la transferencia de datos desde el maestro hacia el filtro; el registro de estados para indicar si el modulo se encuentra procesando datos; el registro Q para indicar la transferencia del valor de cuantización y el registro de coeficientes para escribir en el banco de registros de los coeficientes. El registro de estados es el único que se lee y se hace para saber si el modulo se encuentra procesando datos o no.

Direcciones Core FIR	
Registro de Control	0x 00000004
Registro de Datos	0x 00000008
Registro de Estado	0x 000000012
Registro Q	0x 000000016
Registro Coeficientes	0x 000000020

C. Descripción de Pines

Para conocer las conexiones que se deben tener en cuenta del Core con el procesador o más bien con el Arbitro del Bus Wishbone, es necesario conocer los Pines externos del Core.



Estas señales se deben conectar directamente con una de las señales que tiene como entrada o salida el Arbitro teniendo en cuenta con cual modulo de periférico se esté conectando este Core al bus, ya que el Arbitro posee señales que realizan la misma función pero tienen diferentes nombre de acuerdo al puerto.

CLK: La señal CLK es la señal que mantiene el sincronismo del modulo con el Bus esta se debe conectar con la señal de reloj del Árbitro del Bus.

RST_I: Esta señal se encarga de poner en ceros todas las señales del Core, hace un RESET general al modulo.

STB_I: Es una señal proveniente del Bus con el fin de seleccionar el Core FIR específicamente.

WE_I: Esta señal le indica al Core si en él se realzara una escritura o una lectura.

Clear: Esta señal limpia los registros internos en el Core.

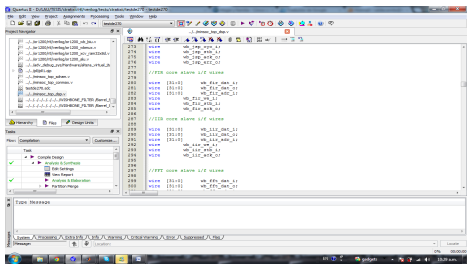
DAT_I: Por esta señal ingresan los datos provenientes desde el Bus.

ADR_I: Esta señal transporta las direcciones tanto base como de cada registro interno del Core.

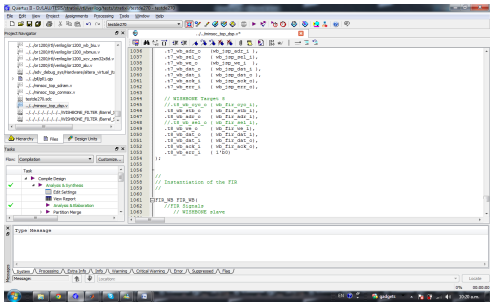
DAT_O: Por esta señal salen los datos procesador por el filtro FIR hacia el procesador.

ACK_O: Señal requerida por el Bus para saber si la lectura o la escritura en el Core se realizo con éxito, es decir saber si la transferencia de datos se está realizando adecuadamente.

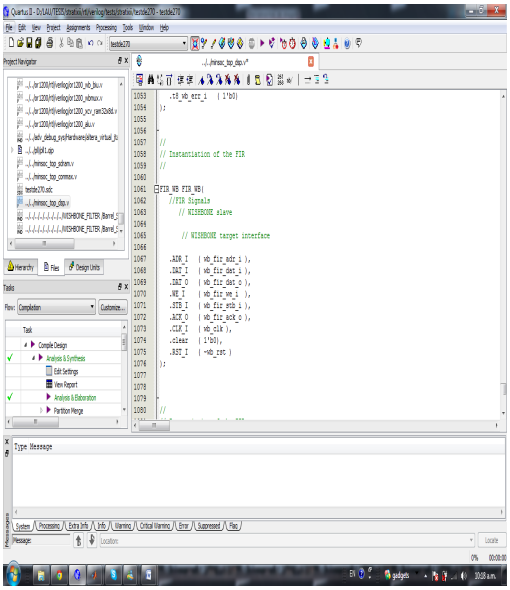
D. Ejemplo de Instanciación



Declaración de señales del Core FIR en lenguaje Verilog

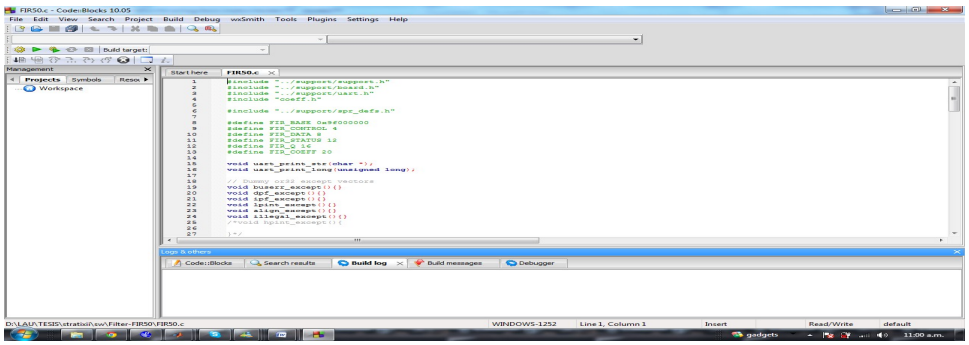


Conexión de las señales del Core FIR con las señales del Arbitro (Traffic COP) usando lenguaje Verilog



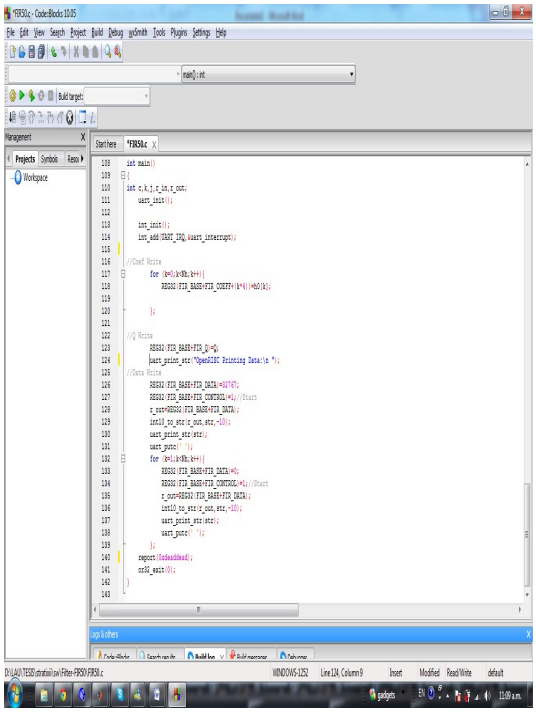
Instanciación del modulo utilizando lenguaje Verilog.

E. Ejemplo de Acceso desde lenguaje C



Declaración de Librerías y declaración de macros para las direcciones tanto base como de los registros internos.

En la figura se observa la declaración de las librerías y los macros de direcciones, además del llamado del archivo de cabecera coeff.h que contiene los coeficientes del filtro en un vector y una variable con el valor de cuantización para el filtro FIR.



Programa principal para la escritura y la lectura del Core FIR

II. Core IIR

A. Descripción del Bloque

Este Core realiza la función de un filtro de respuesta infinita al impulso (IIR), la estructura de implementación para este filtro es en cascada o *Second Order Sections* (SOS), conectando filtros de segundo orden implementados mediante la estructura directa tipo II, cuenta con una interfaz de comunicación bajo las especificaciones del Bus Wishbone. En el momento de la instanciación tiene la facilidad de configurar el ancho de palabra; tanto de la señal de entrada, como de salida, la cantidad de secciones de orden dos, el ancho de palabra de los coeficientes, el valor de cuantización (Q), y la cantidad de bits de crecimiento para que no haya desborde o sobre flujo; Además cuenta con señales lógicas de control correspondientes al reloj (CLK), señal que proviene del reloj global haciendo que el módulo trabaje sincrónicamente; una señal de reinicio (*RESET*), una de limpieza (*CLEAR*), y finalmente una señal de habilitación (*ENABLE*) que hace que el filtro comience el proceso de filtrado; Emplea un multiplexor con el propósito de determinar en cuál celda debe estar la salida, de acuerdo con el filtro diseñado y poder seleccionarlo dependiendo de la cantidad de secciones que se hallan instanciado, ya que se recomienda instanciar una cantidad fija de celdas entre 13 y 18 para la buena utilización de los recursos del FPGA.

B. Descripción de Registros internos y su configuración

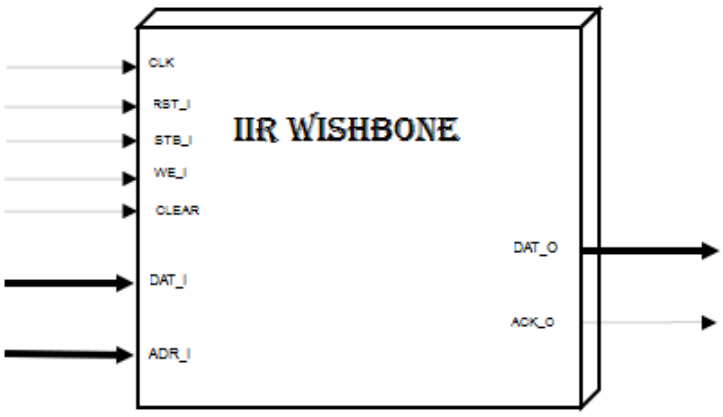
Este Core cuenta con registros internos para su buen funcionamiento, estos registros los vistos en la tabla, para su acceso es necesario conocer la dirección base del modulo y a raíz de esta se accede a los registros sumándole la dirección que se ve en la tabla, un ejemplo de acceso a los registros se mostrara en la parte de acceso mediante lenguaje C.

La escritura de cada uno de los registros se hace situando el valor de la dirección base del modulo mas el valor de cada registro. Los registro se escriben solo accediendo a ellos sin la necesidad de enviarles un valor especifico y estos se escriben para: el registro de control para indicarle al modulo que comience el filtrado, el registro de datos para indicar la transferencia de datos desde el maestro hacia el filtro, el registro de estados para indicar que el modulo se encuentra procesando datos, el registro Nsect para indicar cual etapa del filtro tiene los datos validos filtrados, el registro de ganancia para escribir la ganancia para cada etapa y el registro de coeficientes para escribir en el banco de registros de los coeficientes. Los únicos registros de este Core que se leen son el registro de datos con el fin de

obtener los datos procesados y el registro de estados para saber el estado del Core.

Direcciones Core IIR	
Registro Control	0x 00000000
Registro Datos	0x 00000004
Registro Estado	0x 00000008
Registro Nsect	0x 000000012
Registro Ganancia	0x 000000016
Registro Coeficientes	0x 000000020

C. Descripción de Pines



Estas señales se deben conectar directamente con una de las señales que tiene como entrada o salida el Arbitro teniendo en cuenta con cual modulo de periférico se esté conectando este Core al bus, ya que el Arbitro posee señales que realizan la misma función pero tienen diferentes nombre de acuerdo al puerto.

CLK: La señal CLK es la señal que mantiene el sincronismo del modulo con el Bus esta se debe conectar con la señal de reloj del Árbitro del Bus.

RST_I: Esta señal se encarga de poner en ceros todas las señales del Core, hace un RESET general al modulo.

STB_I: Es una señal proveniente del Bus con el fin de seleccionar el Core FIR específicamente.

WE_I: Esta señal le indica al Core si en él se realzara una escritura o una lectura.

Clear: Esta señal limpia los registros internos en el Core.

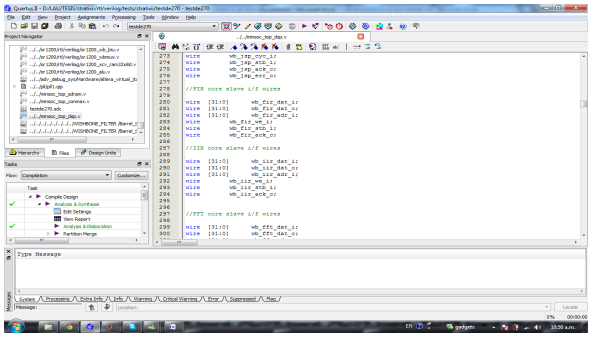
DAT_I: Por esta señal ingresan los datos provenientes desde el Bus.

ADR_I: Esta señal transporta las direcciones tanto base como de cada registro interno del Core.

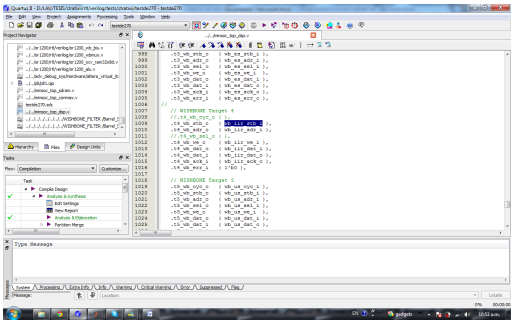
DAT_O: Por esta señal salen los datos procesador por el filtro FIR hacia el procesador.

ACK_O: Señal requerida por el Bus para saber si la lectura o la escritura en el Core se realizo con éxito, es decir saber si la transferencia de datos se está realizando adecuadamente.

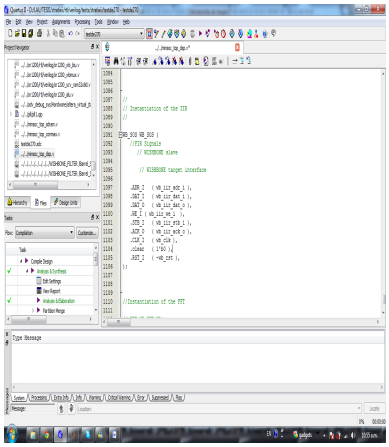
D. Ejemplo de Instanciación



Declaración de señales del Core FIR en lenguaje Verilog

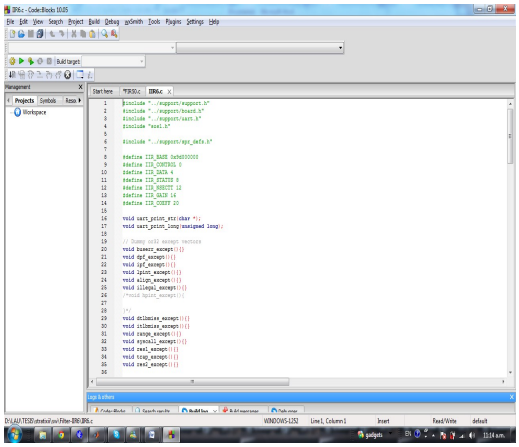


Conexión de las señales del Core FIR con las señales del Arbitro (Traffic COP) usando lenguaje Verilog



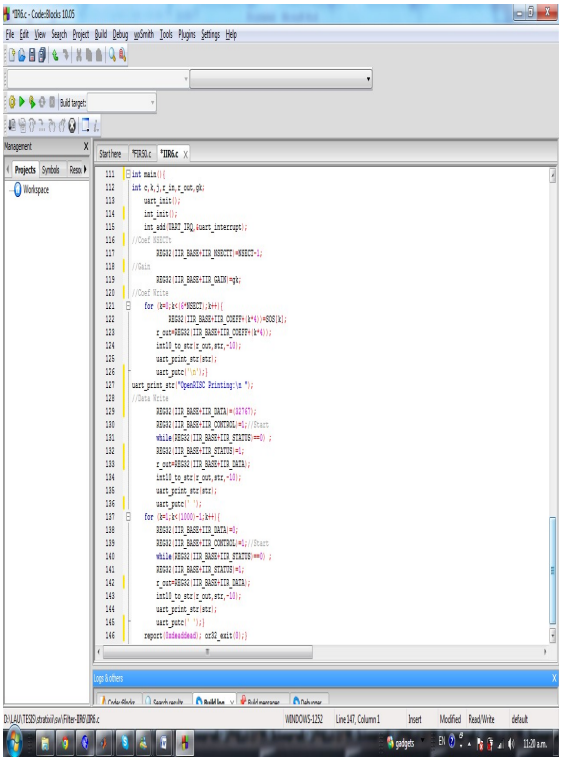
Instanciación del modulo utilizando lenguaje Verilog.

E. Ejemplo de Acceso desde lenguaje C



Declaración de Librerías y declaración de macros para las direcciones tanto base como de los registros internos.

En la figura se observa la declaración de las librerías y los macros de direcciones, además del llamado del archivo de cabecera sos1.h que contiene los coeficientes del filtro en un vector para cada etapa y una variable con el valor de ganancia y una variable con el numero de secciones del filtro IIR.



Programa principal para la escritura y la lectura del Core IIR

III.

IV. Core FFT

A. Descripción del Bloque

2^2

Este Core realiza la función de la función de una transformada rápida de Fourier (FFT), la estructura implementada en esta transformada es la Radix Single-Path Delay Feedback (Radix SDF) ya que es la forma más eficiente de realizar dicha

implementación, cuenta con una interfaz de comunicación bajo las especificaciones del Bus Wishbone.

Para la instanciación de este Core se cuenta con un script en Matlab en el cual configurando la cantidad de puntos que se desea de la FFT crea un archivo VHDL, este crea la cantidad necesaria de memorias ROM que contienen los valores de giro correspondientes para cada etapa generados igualmente en el script.

Posee señales de control del módulo, permitiendo borrar los registros mediante la señal CLEAR, también reiniciar completamente el módulo a través de RESET, igualmente habilitarlo para que comience la transformación de los datos por medio de ENABLE y finalmente la señal de reloj CLK para mantener la sincronización con el sistema.

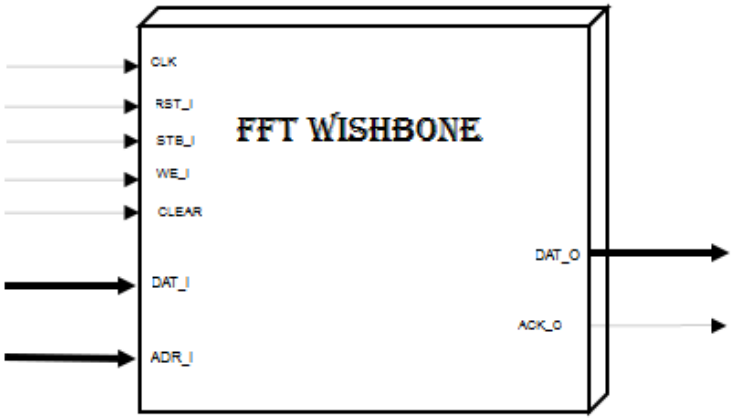
B. Descripción de Registros internos y su configuración

Este Core cuenta con registros internos para su buen funcionamiento, estos registros los vistos en la tabla, para su acceso es necesario conocer la dirección base del modulo y a raíz de esta se accede a los registros sumándole la dirección que se ve en la tabla, un ejemplo de acceso a los registros se mostrara en la parte de acceso mediante lenguaje C.

La escritura de cada uno de los registros se hace de la misma forma de los Cores anteriores, situando el valor de la dirección base del modulo mas el valor de cada registro en él. Los registro se escriben solo accediendo a ellos sin la necesidad de enviarles un valor especifico y estos se escriben para: el registro de control para habilitar el proceso de transformación, el registro de datos para indicar la transferencia de datos desde el maestro hacia el modulo FFT, y el registro de memoria para indicar que se realizara una acción de escritura en la memoria RAM. El único registros de este Core que se lee es el registro de estados que indica la finalización de la transformación.

Direcciones del Core FFT	
Registro Control	0x 00000000
Registro Datos	0x 00000004
Registro Estado	0x 00000008
Registro Memoria	0x 000000012

C. Descripción de Pines



Estas señales se deben conectar directamente con una de las señales que tiene como entrada o salida el Arbitro teniendo en cuenta con cual modulo de periférico se esté conectando este Core al bus, ya que el Arbitro posee señales que realizan la misma función pero tienen diferentes nombre de acuerdo al puerto.

CLK: La señal CLK es la señal que mantiene el sincronismo del modulo con el Bus esta se debe conectar con la señal de reloj del Árbitro del Bus.

RST_I: Esta señal se encarga de poner en ceros todas las señales del Core, hace un RESET general al modulo.

STB_I: Es una señal proveniente del Bus con el fin de seleccionar el Core FIR específicamente.

WE_I: Esta señal le indica al Core si en él se realzara una escritura o una lectura.

Clear: Esta señal limpia los registros internos en el Core.

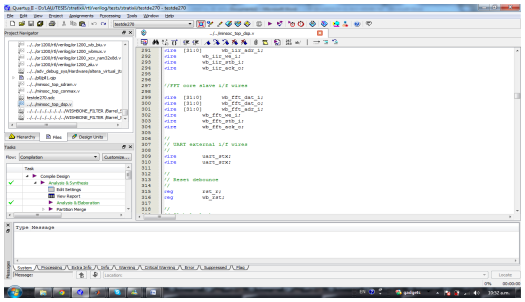
DAT_I: Por esta señal ingresan los datos provenientes desde el Bus.

ADR_I: Esta señal transporta las direcciones tanto base como de cada registro interno del Core.

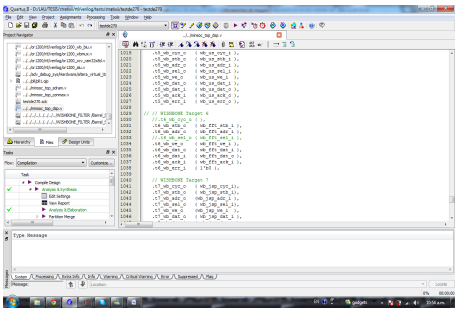
DAT_O: Por esta señal salen los datos procesador por el filtro FIR hacia el procesador.

ACK_O: Señal requerida por el Bus para saber si la lectura o la escritura en el Core se realizo con éxito, es decir saber si la transferencia de datos se está realizando adecuadamente.

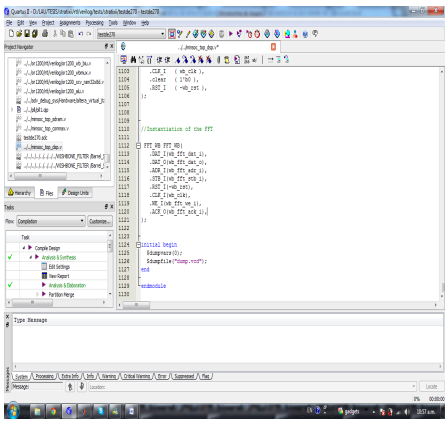
D. Ejemplo de Instanciación



Declaración de señales del Core FIR en lenguaje Verilog

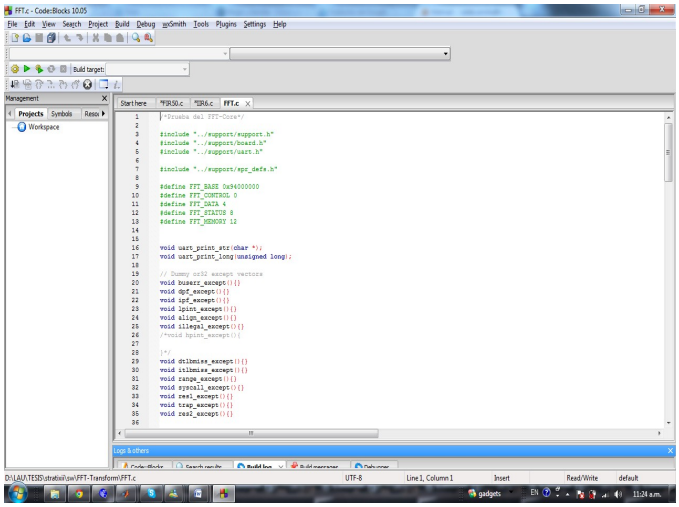


Conexión de las señales del Core FIR con las señales del Arbitro (Traffic COP) usando lenguaje Verilog



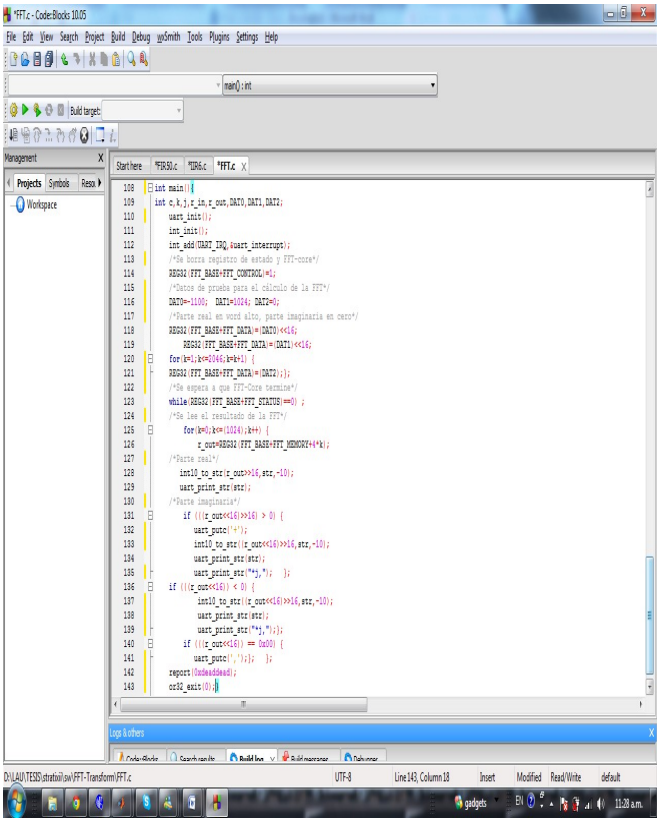
Instanciación del modulo utilizando lenguaje Verilog.

E. Ejemplo de Acceso desde lenguaje C



Declaración de Librerías y declaración de macros para las direcciones tanto base como de los registros internos.

En la figura se observa la declaración de las librerías y los macros de direcciones, además del llamado del archivo de cabecera sos1.h que contiene los coeficientes del filtro en un vector para cada etapa y una variable con el valor de ganancia y una variable con el numero de secciones del filtro IIR.



Programa principal para la escritura y la lectura del Core IIR