



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
IEE2463 SISTEMA ELECTRÓNICOS PROGRAMABLES

Ayudantía 05 2S23

ZYBOZ7- Protocolo AXI y RGB driver

Ayudante: Reimundo Alcalde reimundo.alcalde@uc.cl

Prof. Dr.-Ing. Félix Rojas - felix.rojas@uc.cl

1. Objetivo de la Ayudantía

- Primera interacción con periféricos de la tarjeta Booster.
- Ejercitar protocolo I2C.
- Accionar BUZZER tarjeta Booster.
- Configurar los registros, según hoja de datos del fabricante, del sensor de temperatura TMP006 y de luz OPT3001 de la tarjeta Booster.

2. Actividades Previas a la Ayudantía

En el vídeo asociado a nuestra ayudantía 10 se instancia en la lógica programable un modulo de comunicación I2C para adquirir datos del sensor de temperatura TMP006 y de luz OPT3001 con los que cuenta la tarjeta BOOSTER. Se aprende a acceder correctamente a los registros de cada IC y sus configuraciones según la hoja de datos del fabricante. Además, se realiza un código en VHDL para accionar el BUZZER de la tarjeta, específicamente, se establecen 4 tonos (cambio de frecuencia) y 4 amplitudes (cambio Duty Cycle) distintas.

- Incluya el modulo Zynq en su proyecto de Vivado.
- Incluya el modulo IPCore AXI IIC, el clock *SCL* del protocolo I2C debe estar configurado en 100KHz, velocidad estándar.
- Vea el esquemático de la tarjeta BOOSTER y haga el *Match* entre los pines *SDA* y *SCL* de la BOOSTER y los de la ZYBO. Deberá descomentar en el archivo Constraint estos pines y asociarlo al bloque AXI IIC.

- Cree una nueva fuente **BUZZER** que reciba como entrada *clk*, *BTN0*, *BTN1* y *BTN_JY*. Y de salida *LEDS* y *BUZZERPWM*. Los botones 0, 1 y JY son de la tarjeta *BOOSTER*, por lo que también deberá asociarlo correctamente a los pines de la Zybo.
- Incluya el código *BUZZER.VHD* disponible en el repositorio de la ayudantía 10 dentro de la fuente *BUZZER*. Finalmente, empaquete el IPCore y inclúyalo en su diseño de bloques.
- Cree el HDL Wrapper, genere el Bitstream y exporte el archivo *.xsa*.
- Finalmente, cree un proyecto de plataforma en *Vitis* utilizando el *.xsa*. Cree también un proyecto de aplicación e incluya los archivos *.c* disponibles en Github.

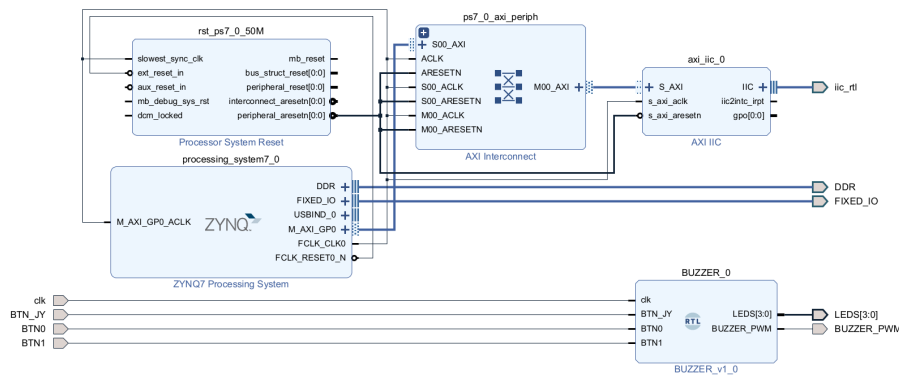


Figura 1: *Block Design* proyecto ayudantía 10 Vivado.

3. Actividades Durante la Ayudantía

Durante la ayudantía deberá modificar el IPCore para generar la PWM asociada al BUZZER. Deberá configurar este pequeño parlante para que alerte al usuario en caso de que la temperatura que marque el sensor comience a subir. Considere una alarma de frecuencia y no de amplitud, es decir, si la temperatura del dispositivo está llegando a valores límites (definido por usted), el tono del parlante deberá ser agudo (Alta frecuencia). Para realizar esto, deberá primero establecer un rango de temperatura de medición, haga pruebas y determine la temperatura mínima que marca el sensor y la máxima cuando acerque su dedo. Una vez que tiene el rango de temperaturas, deberá establecer un rango de frecuencias apreciables.

Debe notar que para realizar esto, debe habilitar un puerto AXI en su IPCore Buzzer, para enviar desde el *PS* la frecuencia deseado dada la Temperatura y de esta forma ajustar la frecuencia de la PWM.

- Modifique el IPCore Buzzer, incorpore un puerto AXI.

- Modifique el código al interior del IPCore para que varíe su frecuencia dada la temperatura que marque el sensor.
- Puede crear una función *Alertar()* en Vitis que reciba como argumento la temperatura, la asocie a una frecuencia definida por usted, y envíe a través de AXI al IPCore Buzzer la información necesaria para ajustar su PWM.

A modo de ejemplo;

Suponga que su rango de temperatura es el siguiente:

- $T_{min} : 18C$
- $T_{max} : 26C$

Establezca una frecuencia mínima, para los $18C$, de 200Hz, y una frecuencia máxima para los $26C$ de 4000Hz. Luego, haga una relación lineal entre T y frecuencia para los valores intermedios.

Importante: Mantenga siempre el ciclo de trabajo en un 50 %