Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Eléctrica
IEE2463 Sistema electrónicos programables

Ayudantia 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

- Ejercitar protocolo AXI.
- Introducir creación de IPCores con puerto AXI.
- Manejar LED RGB ZyboZ7.
- Generación de señal PWM.

2. Actividades Previas a la Ayudantía

En el vídeo asociado a nuestra ayudantía 05 se creó un IPCore con puerto AXI esclavo RGB_{Driver}, el cual recibe 4 constantes de comparación desde un AXI Traffic Generator para ser comparada con un contador diente de sierra y de esta manera, a través del ciclo de trabajo de la señal PWM producto de la comparación, variar la intensidad lumínica del led RGB. Se cargan los archivos $.COE$ en el IPCore ATG para la correcta transacción de datos, y se guardan en 4 registros esclavos del IPCore RGB_{Driver}. Se valida el protocolo por medio de la incorporación de un bloque ILA, logrando visualizar las principales señales de control e información utilizadas para una transacción del tipo AXI lite.

- Cree una IPCore con puerto AXI desde la herramienta $tools \rightarrow CreateAndPackageNewIP$ en la barra superior de Vivado.
- Se abrirá un proyecto nuevo asociado al IPCore, incluya la fuente de menor jerarquía el archivo $.VHD$ `RGB_driver_v1_0_S00_AXI`, en la fuente de mayor jerarquía incluya `RGB_driver_v1_0`. Finalmente, empaquete el IPCore e incorpórelo desde el diagrama de bloques $add\ IP$.

- Incorpore un AXI Traffic Generator, configurado en Test Mode, protocolo AXI lite. Añada los archivo $.COE$ de control, dirección, datos y mascara disponibles en el repositorio.
- Incorpore un bloque ILA $MonitorType$ AXI para medir y validar la correcta transacción de datos a través de AXI entre el ATG y el RGB_{driver}.
- Finalmente, cree el HDL Wrapper y genere el Bitstream.

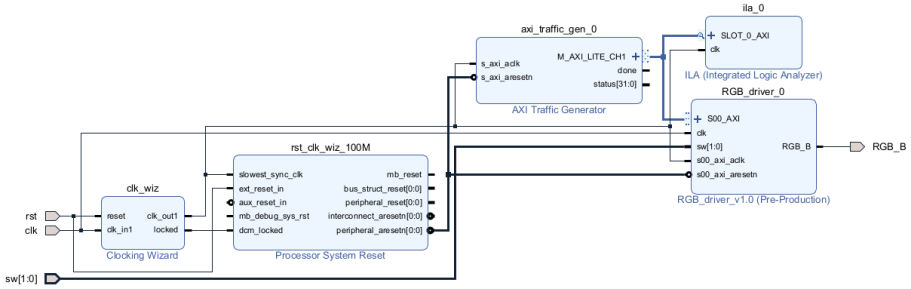

Figura 1: *Block Design* proyecto ayudantía 05 Vivado.

3. Actividades Durante la Ayudantía

Durante la ayudantía deberá modificar el IPCore para generar distintos colores (4 en especifico, no puede ser ni rojo, ni azul ni verde por si solo), para esto considere más salidas en el RGB_{driver}, específicamente, el LED rojo y el LED verde. Cambie las intensidades de color de cada LED, para esto considere enviar más datos desde el ATG en función de los colores que desea generar y utilice más registros esclavos donde almacenarlos.

- Modifique el IPCore, defina las salidas de cada LED, debe generar 4 colores distintos.
- Modifique sus archivos $.COE$ para generar los niveles de intensidad lumínica de cada LED para un color en especifico.
- Cree un código tipo $Look - Up$ Table que reciba de entrada los switches, y genere el color correspondiente.

A modo de ejemplo;

Color Chart	R	G	B	Color Name
	0	0	0	Black
	255	255	255	White
	224	224	224	Light Gray
	128	128	128	Gray
	64	64	64	Dark Gray
	255	0	0	Red
	255	96	208	Pink
	160	32	255	Purple
	80	208	255	Light Blue
	0	32	255	Blue
	96	255	128	Yellow-Green
	0	192	0	Green
	255	224	32	Yellow
	255	160	16	Orange
	160	128	96	Brown
	255	208	160	Pale Pink

Figura 2: Tabla valores RGB.

Suponga que quiere generar el color morado, teniendo en consideración que el máximo de luminosidad en cada LED es de 255 (Ciclo de trabajo unitario). Necesitará aplicar, según la Fig. 2, un ciclo de trabajo $\alpha_{rojo} = \frac{160}{255}$, $\alpha_{verde} = \frac{96}{255}$ y $\alpha_{azul} = \frac{255}{255}$ en cada LED correspondiente. Tenga en cuenta la amplitud del contador que estamos utilizando para mandar el dato a través de AXI al IPCore que generará el ciclo de trabajo necesario para el color que tiene que generar. Definiendo las intensidades que necesita, deberá a través de los Switches generar un color.

También le puede ser útil;

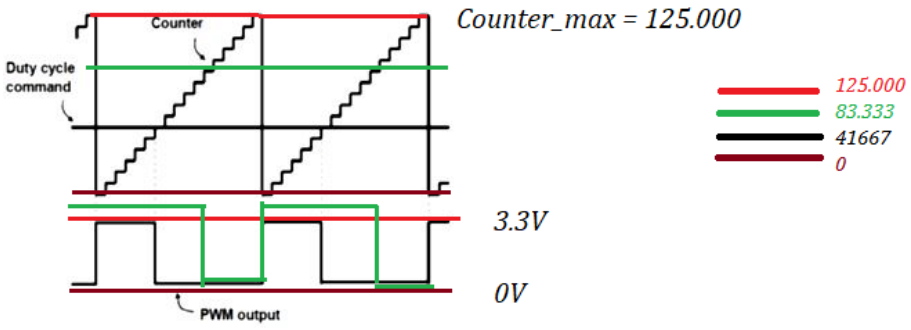

Figura 3: PWM.

El Ciclo de Trabajo se considera como;

$$\alpha = \frac{t_{on}}{T}$$

Dónde t_{on} es el tiempo que pasa la señal en estado HIGH (3.3V) y T es el periodo de PWM.