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

Ayudantía 11 2S23

ZYBOZ-Entity y Architecture con VHDL

Ayudante: Agustín Kamke akamke@uc.cl
Prof. Dr.-Ing. Félix Rojas - felix.rojas@uc.cl

---

1. Objetivo de la Ayudantía

Integración y Control de Periféricos BOOSTXL-EDUMKII mediante Comunicación SPI con la Zybo

- Utilizar la Pantalla.
- Utilizar el ADC.
- Profundizar en los drivers que manejan los periféricos.
- Utilizar las funciones que nos entregan los drivers asociadas a los periféricos

2. Actividades Previas a la Ayudantía

Se realiza la activación de la pantalla y el convertidor analógico a digital (ADC) mediante el empleo de los bloques AXI-SPI proporcionados por Vivado. Además, se instancia el procesador Zynq para gestionar estos bloques. A lo largo del video, se detalla el proceso paso a paso, destacando la interacción entre los componentes mencionados y proporcionando información adicional sobre la configuración de esta integración.

- Analizar el pinout de la tarjeta BOOSTXL-EDUMKII
- Implementar los bloques AXI-SPI y AXI-GPIO en Vivado
- Agregar los archivos de c en Vitis, los cuales se entregan en el repositorio de github.

Es importante configurar los bloques de la forma en que lo hace el video, de esta forma se pueden utilizar sin problemas.

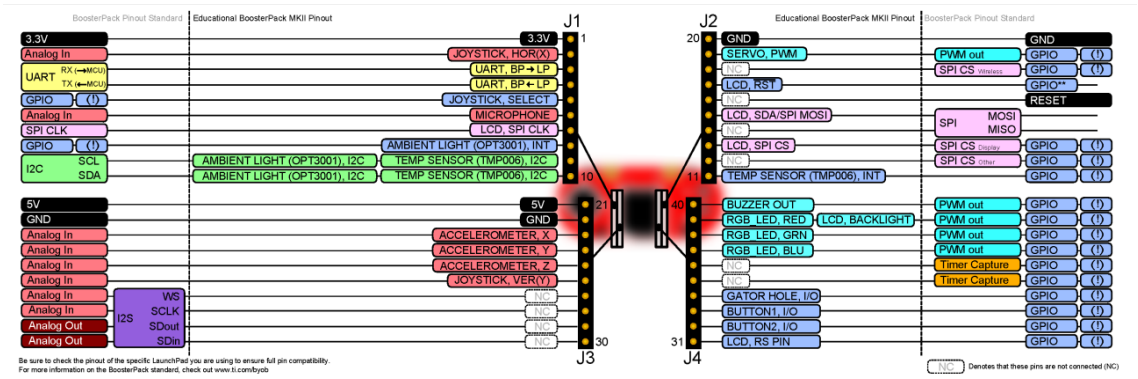

Figura 1: Pinout BOOSTXL-EDUMKII

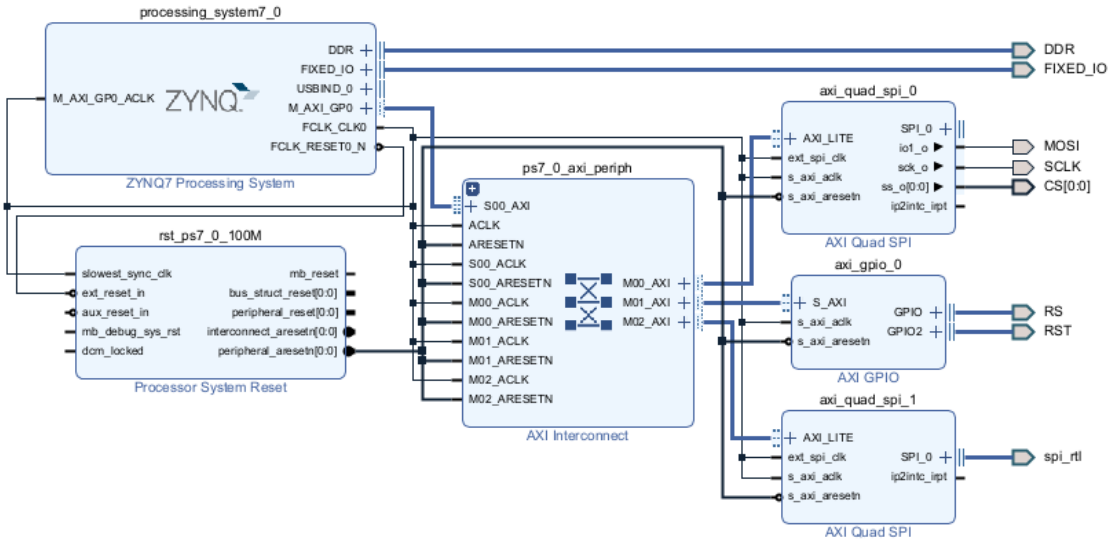

Figura 2: Diagrama de bloques en Vivado.

3. Actividades Durante la Ayudantía

Dentro de los drivers de la pantalla se entregan varias funciones que puede utilizar el usuario, estas se entregan especificamente en LCD GUI.h, el ejercicio de la ayudantía es a esta interfaz inicial hacerle modificaciones de modo que quede totalmente renovada, es decir diseñar una visualización de variables en la pantalla de forma interactiva A continuación se entrega una lista y una breve descripción de las funciones que entregan los drivers de la pantalla.

```
void GUI_DrawPoint(POINT Xpoint, POINT Ypoint, COLOR Color,
DOT_PIXEL Dot_Pixel, DOT_STYLE Dot_FillWay);
void GUI_DrawLine(POINT Xstart, POINT Ystart, POINT Xend, POINT Yend,
COLOR Color, LINE_STYLE Line_Style, DOT_PIXEL Dot_Pixel);
```

```
void GUI_DrawRectangle(POINT Xstart , POINT Ystart , POINT Xend ,
POINT Yend , COLOR Color , DRAW_FILL Filled , DOT_PIXEL Dot_Pixel );
void GUI_DrawCircle(POINT X_Center , POINT Y_Center , LENGTH Radius ,
COLOR Color , DRAW_FILL Draw_Fill , DOT_PIXEL Dot_Pixel );
void GUI_Disbitmap(POINT Xpoint , POINT Ypoint , const unsigned char *pBmp,
POINT Width , POINT Height );
void GUI_DisChar(POINT Xpoint , POINT Ypoint , const char Acsii_Char ,
sFONT* Font , COLOR Color_Background , COLOR Color_Foreground );
void GUI_DisString_EN(POINT Xstart , POINT Ystart , const char * pString ,
sFONT* Font , COLOR Color_Background , COLOR Color_Foreground );
void GUI_DisNum(POINT Xpoint , POINT Ypoint , int32_t Nummber,
sFONT* Font , COLOR Color_Background , COLOR Color_Foreground );
```

Descripción de Funciones

- `GUI_DrawPoint`: Dibuja un píxel en las coordenadas especificadas con un color dado. Puede dibujar un píxel sólido o uno más grande según el valor de `Dot_Pixel` y `Dot_Style`.
- `GUI_DrawLine`: Dibuja una línea entre dos puntos dados. El estilo de línea y el grosor del píxel pueden ser ajustados mediante los parámetros `Line_Style` y `Dot_Pixel`. También es posible especificar un patrón de línea discontinua.
- `GUI_DrawRectangle`: Dibuja un rectángulo en la pantalla, con la posibilidad de especificar si se debe rellenar o no. Los bordes del rectángulo tienen un grosor determinado por `Dot_Pixel`.
- `GUI_DrawCircle`: Dibuja un círculo en la pantalla con el centro, radio y estilo especificados. Puede ser un círculo sólido o uno vacío, y el grosor del píxel puede ser ajustado.
- `GUI_Disbitmap`: Muestra una imagen bitmap en las coordenadas especificadas. La imagen está definida por el puntero `pBmp`, con el ancho y alto dados por `Width` y `Height`, respectivamente.
- `GUI_DisChar`: Muestra un carácter ASCII en las coordenadas dadas. Los colores de fondo y primer plano se pueden personalizar.
- `GUI_DisString_EN`: Muestra una cadena de caracteres en inglés en la pantalla. Similar a `GUI_DisChar`, pero puede procesar cadenas completas.
- `GUI_DisNum`: Muestra un número en las coordenadas especificadas. Convierte el número a una cadena y luego utiliza `GUI_DisString_EN` para mostrarlo en la pantalla.