

Práctica 5. Diseño de un Controlador para Pantalla Táctil.

Antonio María Franqués García (77791084R), grupo A2A.

Tarea única

En esta práctica se implementa el control táctil de la pantalla de la FPGA junto al uso del teclado; conectado también a ésta. Concretamente se simula el estado del cielo a lo largo del día. Un punto cuadrado de ancho configurable y de color amarillo, que representa El Sol, aparece en la zona de la pantalla que estemos tocando, mientras tanto el fondo, representando el cielo, se ilumina o se oscurece según la altura a la que se encuentre éste.

Para mayor comodidad, se permite mediante el uso de los cursores del teclado, el control de dicho punto. Correspondiendo los cursores izquierdo y derecho con distintos valores de acimut (misma cantidad de luz solar) y los cursores arriba y abajo con distintos valores de altura (mayor o menor luminosidad solar).

Comentarios y dificultades encontradas durante la realización:

Debido a que hice una estructura del tipo máquina de estados para la lectura del punto tocado de la pantalla ahora me ha sido más fácil unir el control por teclado y la visualización por pantalla.

El estado cero, o de reposo, de la captura del punto de la pantalla es el único que permite el control por teclado, dando así preferencia al táctil.

El primer paso al empezar la práctica fue partir del proyecto visualiza.qar facilitado por los profesores en la práctica 3 ya que así dispuse del esquemático que contiene la señal de código de tecla pulsada y la señal de control que notifica dicha pulsación. El objetivo fue crear el símbolo que contuviese el táctil y la salida por pantalla para así añadirlo y conectarlo al resto de señales de dicho esquemático.

Tuve que crear una señal que se invirtiera cada vez que pulsaba una tecla para poder comprobar que la señal de control saliente del bloque del teclado funcionaba. Así pues me pareció una buena idea asociar dicha señal a un LED de la FPGA que invierte su estado a cada pulsación. Aunque ahora ya he comprobado que todo funciona he dejado dicha señal asociada a dicho LED para comprobar al inicio de cada uso que la conexión entre el teclado y la pantalla es correcta. La señal de control saliente del bloque del teclado la descubrí mirando el código de dicho bloque. El motivo por el cual la necesito es que yo no quiero mover el punto indefinidamente hasta que se detecta otra combinación de tecla sino que quiero mover una posición cada vez que se produce la interrupción de teclado.

Para poder restringir el dibujo del punto rectangular que representa El Sol sólo en la zona de la pantalla indicada pensé en sumar a la “fila” inicial el valor de “oX_COORD” que el táctil indicara, sin embargo “oX_COORD” decrece a medida que “fila” crece, así pues debí crear una señal auxiliar “oXaux_COORD” que fuera “oX_COORD” invertida, de esa forma si yo indico que quiero el punto en la “fila” cero le corresponderá sumarle “oXaux_COORD” cero en vez de FFF que es el valor que facilita el control táctil al tocar esa zona y que haría pintar el punto en el otro lado. “oY_COORD” no hizo falta invertirla ya que crece acorde con “columna”.

Otro aspecto importante es que “oXaux_COORD” sólo se fija e invierte los 9 bits de mayor peso de “oX_COORD” en vez de los 12, ya que debo ajustar que si quiero el punto en el último valor de “columna” no le sume FFF (que es el valor que correspondería a esa zona si hubiera invertido los 12 bits de “oX_COORD”) y se pase del área de la pantalla, sino que le sume el valor máximo que hace que se quede dentro. Así pues, siendo 35 y 515 los valores límite para

“fila” y 216 y 1016 los valores límite para “columna”, no puedo hacer “fila” $35 + \$FFF = 4130$ sino que como máximo puedo hacer $35 + (1\ 1111\ 1111) = 546$ (coger un bit menos haría que nunca le pudiese sumar lo suficiente como para situar el punto en el extremo de la pantalla). Para “columna” no puedo hacer $216 + \$FFF = 4311$ ya que es mayor que 1016, sin embargo sí que $216 + (11\ 1111\ 1111) = 1239$ se queda muy cerca del valor límite. Definitivamente debí sumarle al eje de “fila” los 9 bits invertidos de mayor peso que indicamos con “oXaux_COORD” y los 10 bits sin invertir de mayor peso que indicamos con “oYaux_COORD” al eje de “columna”.

Hacer los incrementos de posición del punto en pantalla fue relativamente fácil una vez creadas las señales auxiliares “oXaux_COORD” y “oYaux_COORD”, ya que tan sólo hay que sumarles una cantidad de posiciones en caso de que se haya producido una interrupción de teclado y la tecla pulsada se corresponda con las que están permitidas.

Comentarios sobre la simulación: Se observa una vez transferido el código a la FPGA que el punto rectangular sigue el movimiento del dedo sobre la pantalla, que el color de fondo varía según la posición vertical del punto y que finalmente podemos controlar el punto mediante los cursores del teclado. Al mismo tiempo observamos la inversión del estado del LED a cada interrupción del teclado y el reset de la pantalla al pulsar el botón KEY0.

Observando tal funcionamiento podemos decir que todo funciona según lo esperado.

Autoevaluación:

- Estimo que la dedicación a esta tarea ha sido de 8 horas, incluyendo las del horario de clase de prácticas. Aunque al principio no tuve muy claro por dónde empezar y como estructurar los bloques, a las dos horas de haber empezado ya había cogido soltura. El hecho de que pude quedarme 6 horas seguidas en clase ese día y con la ayuda de los profesores hizo que tan sólo tuviera que dedicarle otras dos horas de otro día para poderla terminar. En mi opinión es una cantidad de tiempo adecuada. No quise dedicar mucho tiempo en curar la estética del programa sino en englobar un funcionamiento conjunto de las anteriores prácticas.
- La dedicación a la realización de la tarea ha sido total ya que la he realizado de forma individual.