

Actividad 1

Introducción

En esta serie de actividades se plantean una serie de objetivos de desarrollo de sistemas alrededor del sistema de buses Wishbone en orden de funcionalidad y complejidad crecientes. Asumiremos que se dispone de una o más cpu y de una o más memorias que se desea conectar entre sí. En este caso se trata únicamente de un procesador y una memoria.

Objetivo

En esta actividad el objeto es hacer una conexión punto a punto entre un procesador y una memoria usando el bus Wishbone. Supondremos que la memoria es capaz de comunicarse con el procesador sin estados de espera, es decir puede responder a las peticiones del procesador en un sólo ciclo.

La interfaz de el procesador es la siguiente:

```
cpu(input wire clk, reset, output wire rd, wr, output wire [15:0] dir, inout wire [7:0] data);
```

Y la definición de la memoria es esta:

```
module memdata(input wire clk,
               input wire cs,           //señal de habilitación del módulo
               input wire we,           //señal de habilitación de escritura
               input wire oe,           //señal de habilitación de salida
               input wire [11:0] dir,    //direcciones
               inout wire [7:0] data);   //datos bidireccional

    reg [7:0] mem[0:(12 * 1024)-1]; //memoria de 12K palabras de 8 bits de ancho

    always @(posedge clk)
        if (cs && we) mem[dir] <= data; //Escritura síncrona

    assign data = cs ? (oe ? mem[dir] : 8'bZ) : 8'bZ;
    //we oe = 00: data en Z, 01: lectura, 10: escritura, 11: caso ambiguo
endmodule
```

Los módulos a desarrollar serían la interfaz de cada elemento (la de master que nos conecta con la cpu y la de slave con la memoria) y un módulo que representa la red de interconexión INTERCON fusionado con la generación de reloj y reset SYSCON. Todo ello acompañado de un fichero con las plantillas de las señales wishbone y su uso y cualquier cuestión que se quiera añadir.

Consideraciones

- Aunque el bus de datos sea bidireccional, puede ser más conveniente desdoblar ese bus en dos buses de un solo sentido en las definiciones de cpu y memdata, adaptando lo que sea necesario en las definiciones de los módulos. Así sería más parecido a lo que se tiene en el Wishbone
- La memoria no llena el espacio direccionable del procesador, pero se puede asumir que no hay nada más colgado al procesador y por tanto no hay que hacer una decodificación de las direcciones para activar la memoria (simplemente unir los buses de direcciones dejando al aire los bits sobrantes)
- No importa si dejamos de usar gran cantidad de señales, que en este caso no serían necesarias, simplemente omitirlas o si es obligatorio incluirlas según la especificación dejarlas fijas en el estado correcto.
- Acompañar al código fuente de las plantillas de señales necesarias explicando lo que se ha hecho con cada una.

- El procesador puede ser la cpu real que hemos diseñado si se soporta un bus de direcciones, una adaptación con un bus de direcciones más pequeño o una simulación de cpu sencilla que simplemente emita direcciones y lea o escriba en ellas con las señales de control adecuadas.
- Opcionalmente, si hemos usado nuestra cpu o si hemos hecho una cpu simulada, escribir un programa/hacer cambios en el diseño para permitir hacer un test de memoria simple.
- Opcionalmente, considerar (teóricamente) qué habría que añadir en el diseño de nuestra cpu y en su conexión con la interfaz master para conseguir que si no recibimos en ACK en la interfaz de master la cpu quede en espera hasta acabar el ciclo de bus.