

Tarea Verilog

IE-503 Estructuras de Computadoras Digitales I

II-2013

Prof. Lochi Yu

Fecha límite de entrega: Viernes 6 de Diciembre, 11:59pm

Se debe de entregar: Un archivo .pdf con:

1. Programa en ensamblador que se utilizó para probar la instrucción.
2. Diagrama de temporización.
3. Código de verilog donde está el error (si lo hubo).
4. Código de verilog corregido (si hubo error).
5. Diagrama de temporización correcto (si antes había salido con errores).

No se permiten entregas tardías.

Objetivo:

Usando los archivos .v entregados, detectar cuáles instrucciones de la CPUCR se comportan de una forma distinta a la vista en clase. Deben revisar solamente instrucciones, no otros comportamientos dependientes de eventos externos como interrupciones, DMA, wait, etc. Quedan excluidos por ende las instrucciones relacionadas con interrupciones).

Deben imprimir todos los microprogramas utilizados para verificar el funcionamiento correcto de cada instrucción. Para todo error detectado se deberá mostrar:

1. la forma de onda anómala (40% de la nota de ese error)
2. la prueba que se está corriendo (20%)
3. indicar en qué módulo de verilog se encuentra el problema (30%)
4. indicar cómo corregirlo (10%)

Si un error no fue detectado tendrá un 0 en esa prueba. Se tomarán todos puntajes de los errores, y se promediarán y será la nota de la tarea.

Instrucciones:

Antes de poder ingresar los programas de prueba, tienen que montar el testbench. Éste está en el archivo testfinal.v.

- Dentro de este módulo, no existe todavía ningún módulo. Hay que instanciar los módulos de reloj, cpucr y memoria.
- Para poder instanciarlos, hay que conectarlos por medio de nodos, que serán o “wire”s o “reg”s. Los wires son nodos pasivos, los cuales no se le puede fijar el valor (el valor se los va a fijar algún reg dentro de algún módulo). Los regs son activos, osea, nosotros fijaremos sus valores.
- El primer paso es mirar dentro de cada uno de los tres módulos y hacer un listado de señales, ver cuáles son salidas y cuáles son entradas.
- Desde el archivo testfinal.v, podemos poner cualquier nombre, no es necesario que sea igual a las señales internas de cada módulo, aunque por claridad, debería ser similar.
 1. Instanciar todos los regs:
 - `reg nombre_señal; //señal de 1 bit`
 - `reg [7:0] bus; //señal de 8 bits`

2. Instanciar todos los wires
3. Instanciar el reloj, memoria y cpucr, colocando las señales EN EL MISMO ORDEN que los recibe el módulo (ver dentro de cada módulo) o de la forma indicada en el laboratorio que se llevó a cabo el día Martes 12 de Noviembre.
4. En el bloque de “initial”, luego de la instrucción “\$dumpvars” y antes del “\$finish”, se colocarán las señales de inicialización.
 - cualquier señal puede ser acompañada por un retardo de tiempo:
- #10 a=1; //significa que luego de esperar 10 unidades de tiempo, a tendrá un valor de 1

En el archivo de memoria es donde se ingresarán los microprogramas.

Simulación:

En las computadoras del LICIT, abrir una línea de comandos, en el mismo directorio de los archivos .v, correr:

```
iverilog -o salida.o CPUCR_8bits_alterada.v testfinal.v memfinal.v clock.v
```

Esto compilará los archivos de verilog. Si hubiera un error en la definición de los módulos aparecerá aquí.

```
vvp salida.o
```

Esto ejecutará el sistema y grabará en un archivo test.vcd la salida.

```
gtkwave test.vcd
```

Desplegará las ondas resultantes. Deben marcar el módulo en la izquierda y arrastrar la señal a la columna de la derecha. Hay que ajustar el zoom para ver bien las ondas.

Advertencia: cada grupo deberá diseñar sus propios microprogramas. Cualquier sospecha de plagio será denunciada a la Dirección de la Escuela.