

Laboratorio 1

Organización del Computador
Original 2014, Eduardo Sánchez

Objetivos

- Desarrollar código a base de componentes simples en lenguaje VHDL, que describan pequeños circuitos secuenciales y combinacionales vistos en el teórico y el práctico.
- Utilizar la herramienta GHDL para analizar y “compilar” el código VHDL.
- Desarrollar y/o entender código VHDL (*testbench*) que testee el código del circuito.
- Mediante el uso de *gtkwave*, analizar las formas de onda y testear el resultado.
- Aprender a reutilizar código VHDL mediante componentes.

Tarea

La tarea a desarrollar en este laboratorio consta de la implementación de una máquina de estados (*fsm*) de una expendedora de gaseosa. Para ello deberán realizar:

1. El ejercicio en formato papel (determinando el diagrama de estados, la tabla de transiciones, la ecuación y la implementación en compuertas/PLA/ROM/Etc.)
2. La implementación del mismo en lenguaje VHDL. Para estandarizar, el archivo con la entidad principal se deben llamarlo *fsm.vhdl*, y el *testbench* deben llamarlo *fsm_tb.vhdl*.

Ejercicio pedido:

Diseñar la máquina de estados de una expendedora de gaseosa.

Descripción del funcionamiento y particularidades:

- Cuando el cliente ingresa el importe adecuado (\$2) por la ranura de monedas y aprieta el botón “b” en la máquina, se entrega la gaseosa; y vuelve a esperar nuevos pedidos.
- Solo acepta monedas de \$1 y de \$0,50.
- Solo se expende un tipo de gaseosas.
- La máquina no da vuelto.
- Existe un boton “c”, que permite cancelar la compra y devolver el dinero (esto último, lo hace automáticamente y ustedes no deberían preocuparse, solo deberían poner la máquina a esperar nuevos pedidos).
- Si se pide entregar la gaseosa mediante el botón “b”, y no se ingresó el monto correcto, la máquina de estados no se ve afectada.
- El cliente puede pasarse del monto de la gaseosa y al presionar el botón “b”, perder el dinero excedente.
- Al ser una única ranura de monedas, no pueden ingresarse los dos tipos de monedas al mismo tiempo.
- Puede eliminar estados inalcanzables siempre y cuando los especifique. Por ejemplo: “el cliente no puede presionar dos botones a la vez”, o “el cliente no puede presionar un botón e ingresar una moneda”.

Para lograr hablar todos el mismo “idioma” en cuanto a entradas y salidas; fijamos la comunicación al exterior de las entidades de más alto nivel (*fsm.vhdl*); por lo que se pide

Laboratorio 1

Organización del Computador
Original 2014, Eduardo Sánchez

que todas tengan la siguiente interfaz:

- Entradas: p , m , b , c , clk y rst de tipo *std_logic*.
- Salida: *release* de tipo *std_logic*.
- Las señales para los estados a utilizar: " sN " para los estados actuales y " sN_prima " para el nuevo estado (ejemplo: en caso de necesitar 4 bits de estado, serán " $s3,s2,s1$ y $s0$ ", donde 3 es el bit más significativo).
- Si se ingresa una moneda de "\$1", entonces " p " vale "1".
- Si se ingresa una moneda de "\$0,50", entonces " m " vale "1".
- Cuando se presiona el botón de gaseosa, " b " vale "1".
- Cuando se presiona el botón de cancelar, " c " vale "1".
- Cuando no se ingresa ninguna moneda ni se presiona algún botón, la máquina de estados debería permanecer en el estado en que se encuentre.
- "*release*" es una señal de salida que cuando vale "1" libera la gaseosa y se la entrega al cliente. Se mantiene en "1" hasta la transición al estado inicial.

En caso de minimizar las ecuaciones con algún *solver* automático, tenga en cuenta las posiciones (índices) en la tabla de los estados eliminados. **No** son condiciones sin cuidado, la máquina debe regresar al estado inicial.

Requisitos del código a entregar

- Las entregas serán a través del mail: entregas.odc2013@gmail.com. Deberán crear un archivo comprimido (tar, zip, rar, etc.) explicitando el número de laboratorio y sus integrantes (ApellidoNombre): ej: "lab1_FraireJuan_SanchezEduardo_FerreyraPablo.tar.gz".
- Se deben utilizar los mismos nombres de los componentes de alto nivel pedidos.
- Junto con el código, se deberá entregar un pequeño informe en el cual se explique la estructuración del código (entidades de mayor nivel), decisiones de diseño tomadas (si las hubiere), dificultades con las que se encontraron y cómo las resolvieron, *testbenches*, etc. junto con el ejercicio en formato "papel" resuelto en el que se basaron para implementar la solución. Sin el informe, el laboratorio **NO** se considera completo.
- El formato de dicho informe queda a elección de ustedes, siempre y cuando sea un formato libre.
- El trabajo es grupal (Max. 3 personas). Todos los integrantes del grupo deberán ser capaces de explicar el código presentado (en una fecha posterior a la entrega).
- **No** está permitido compartir código entre grupos.

Entrega de Laboratorio: **25 de abril.**