

Tarea #1, parte A

Ejercicios Circuitos Digitales

Uso software para simulación "Digital", 2do Semestre - 2023

Profesor: PhD. Nerio Andres Montoya Giraldo
namontoy@unal.edu.co

Consideraciones Generales

A continuación se presentan una serie de ejercicios para ser resueltos con el software que se ha utilizado en clase en las 2 primeras semanas del curso (Digital).

Tenga en cuenta las condiciones que se plantean en la descripción de cada punto y resuelva los ejercicios solo apegándose a las condiciones planteadas.

La tarea será revisada en la semana 6 del curso, ya sea el día jueves o viernes de dicha semana, dependiendo si es el grupo 01 o 02.

NOTA: Esta tarea puede ser resuelta en grupos, pero cada estudiante es responsable de forma individual por la presentación de la misma. Por favor, realice los ejercicios usted mismo, no copie y pegue de su compañero, es usted quien se está afectando en su proceso de aprendizaje.

NOTA 2: Puede agregar comentarios en cuadros de texto de forma que sea más fácil entender qué está haciendo.

NOTA 3: Recuerde que estamos para ayudarle, pero no para resolver por usted los problemas planteados

Ejercicios:

1. Utilizando dos TriState buffers, crear un multiplexor de 2 entradas y 1 salida. (5 puntos)
2. El elemento “dual” de un multiplexor se conoce como **Demultiplexor** (Demux) y su funcionamiento es completamente el “espejo” de un Mux: redirige una sola entrada a muchas posibles salidas (solo una a la vez). Utilizando compuertas lógicas de máximo 4 entradas, por favor crear un Demux de 1 entrada y 4 salidas. (10 puntos)
3. **Extensión del modelo simple de un puerto de uso general:** (35 puntos)

Descripción del ejercicio:

En clase estudiamos un modelo muy simple de un puerto GPIO, en el cual con 3 registros se podía simular el comportamiento básico de entrada y salida de un pin genérico (en la clase se simuló un puerto con 4 pines, utilizando tres registros de 4-bit cada uno). El presente ejercicio consiste en “expandir” las funcionalidades del puerto descrito en la clase, de forma que cada pin podrá tener dos posibles registros de entrada (ambos diferentes) y dos posibles registros de salida (ambos diferentes), para un total de 4 funciones diferentes en un mismo pin.

NOTA: Este ejercicio asume que el estudiante ya ha replicado la simulación mostrada en clase, debido a que es desde esta simulación funcionando de donde se debe partir para desarrollar este ejercicio (la imagen del circuito se encuentra disponible en los recursos montados en la página web del curso).

Para desarrollar este ejercicio, por favor tenga presente que un pin es la interfaz que utiliza un microcontrolador para interactuar con el medio que lo rodea. Los pines de propósito general más simples son aquellos que solo pueden ser configurados como entradas o salidas digitales (unos y ceros), mientras que los pines más complejos pueden realizar otras funciones, como entradas análogas o ser conectados a otras herramientas internas del microcontrolador, como puertos de comunicación.

Para el caso de un puerto un más complejo, es necesario por lo menos la existencia de 5 registros relacionados con la configuración y uso de cada pin:

-
- Un primer registro que maneja el comportamiento de entrada o salida del pin, también conocido como registro de “modo de funcionamiento”. En el caso simple (explicado en clase) en este registro solo se seleccionan dos opciones: Entrada o salida; mientras que en el caso que se desea desarrollar se desean seleccionar cuatro modos: **Entrada simple, entrada análoga, salida simple, salida alternativa**. Observe que en este caso, es necesario tener dos bits por cada pin (para seleccionar 4 opciones) por lo cual el registro del ejemplo de clase debe ser ampliado a un registro de 8 bits (2 bit para configurar el modo de trabajo de cada uno de los 4 pines del puerto del ejemplo). Por favor utilice bits contiguos para cada pin, por ejemplo: Bits 0 y 1 para el Pin0, bits 2 y 3 para el Pin1 y así sucesivamente.
 - Es necesario tener dos registros de entrada, uno que almacena el valor de la entrada simple (llamado Input Data Register) y otro idéntico (llamado ADC Input Register), que almacenará el estado de la entrada que supondremos análoga (no será una entrada análoga real, ya que en Digital no es posible simularla).
 - Son necesarios además dos registros de salidas, uno llamado Output Data Register, y uno llamado Alternate Output Register, los cuales se encargarán de controlar la salida del pin, que a su vez dependerá de cuál será la salida activa (Output o Alternate).

Además de los registros descritos anteriormente, es necesario ampliar un poco más las conexiones internas del pin, de forma que los 4 canales (2 de entrada y 2 de salida) puedan ser direccionados hacia/desde el pin. Para esto, es necesario:

- Agregar dos multiplexores 2:1 creados con elementos TriState (punto 1 de la tarea), un Mux en dirección de entrada y el otro Mux en dirección de salida.
- El pin que controla la selección del canal activo en **ambos** Mux debe ser controlado por un solo bit del registro “modo de funcionamiento”.
- El otro bit del registro “modo de funcionamiento” debe controlar el hardware ya simulado, que habilita si el pin es una entrada o salida.
- Recuerde que ambos bits deben ser preferiblemente contiguos.

Después de tener todo configurado adecuadamente, los valores cargados en el registro “modo” (dos bits por pin) activarán el pin X en uno de sus 4 posibles modos de funcionamiento.