

El objetivo de este boletín es familiarizar al alumno con la simulación de circuitos digitales combinacionales mediante simulación lógica. Para estudiar las características de conmutación típica de circuitos digitales nos fijaremos en la implementación, mediante elementos de librería, de funciones de conmutación combinacionales. Se hará énfasis en el análisis lógico-temporal, analizando el efecto de los parámetros configurables de los modelos que implementan, azares y ambigüedades. El boletín se compone de 4 ejercicios.

- 1.- Dibuje los esquemáticos de la Fig.1 conteniendo los elementos de librería que se indican. Deduzca analíticamente la función combinacional que implementan previamente a realizar las simulaciones.
 - 1.a) Obtenga usando PSPICE la tabla de verdad de los circuitos de la Fig. 1. Emplee para ello estímulos tipo DigClock de baja frecuencia (periodos múltiplos de 1 μ s).
 - 1.b) Repita el apartado anterior, únicamente para la puerta AND de la Fig 1.a, fijando los puertos de entrada a los valores lógicos alto (HI) y bajo (LO).
 - 1.c) Conecte la puerta triestado 74125 a la salida de una puerta AND 7408 (Fig 2.a) para generar un bloque cuya salida esté habilitada por la señal de control de la puerta triestado. Caracterice la nueva puerta, analizando el caso de la Fig 2.b. Considere retrasos típicos. Discuta la utilidad genérica de las puertas con salida triestado.

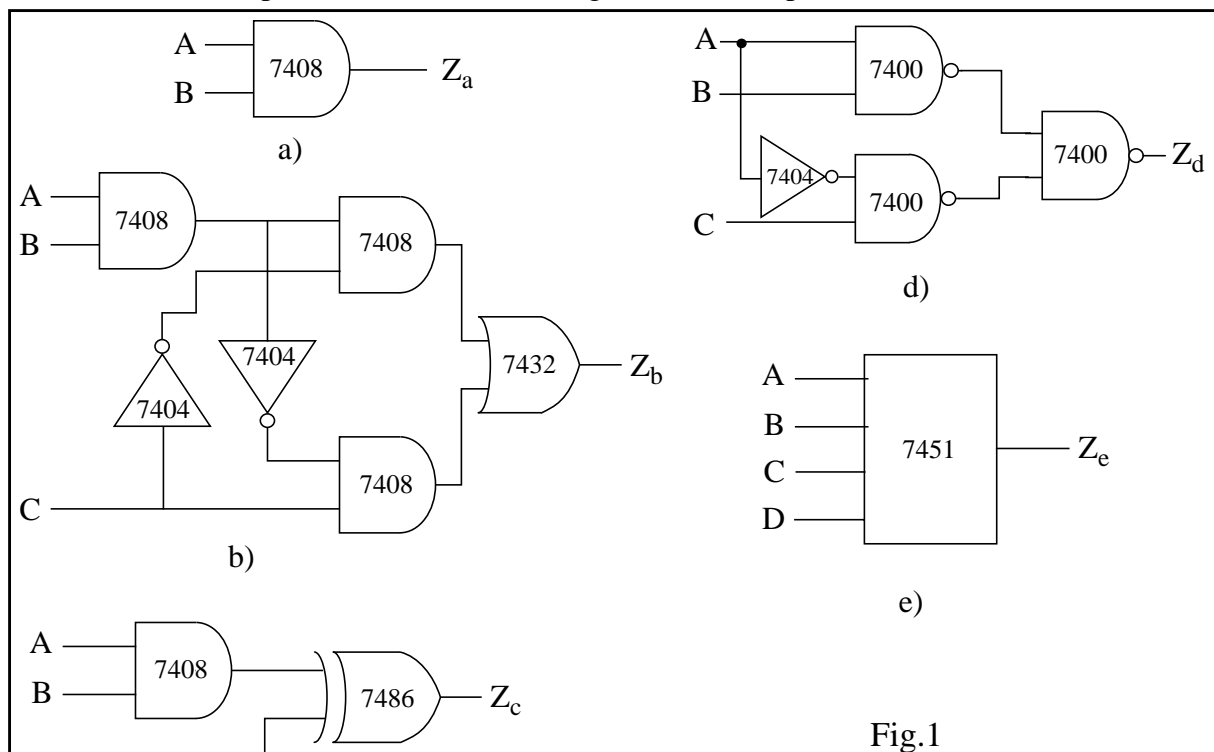
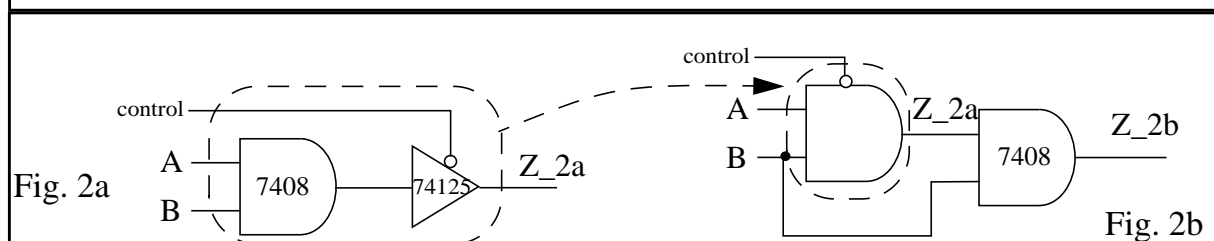
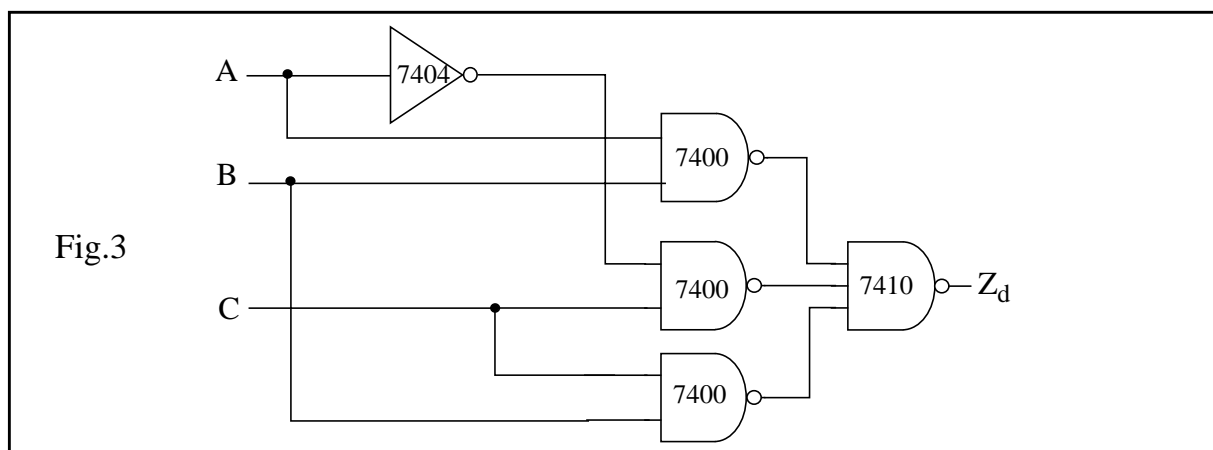


Fig.1



- 2.- En este problema trataremos el estudio mediante simulación de azares dinámicos (provocados por cambios simultáneos o cercanos en dos o más entradas) en circuitos combinacionales.
- 2.a) Compare para el circuito de la Fig. 1c los tiempos de retraso en los casos típico, mínimo y máximo (para no cambiar el parámetro MNTYMXDLY de todas las puertas, basta con modificar el parámetro en *Analysis* \Rightarrow *Setup* \Rightarrow *Digital Setup* \Rightarrow *Minimum, Typical o Maximum*). Discuta las anomalías que se observan en la salida en el caso en que los patrones de entrada recorran todos los valores posibles y conmuten simultáneamente (estos análisis se puede realizar seleccionando en *Analysis* \Rightarrow *Setup* \Rightarrow *Digital Setup* \Rightarrow *Worst-Case (Min/Max)*). Estudie la presencia de **regiones de ambigüedad** (diferencias en la respuesta entre el caso de retraso mínimo y máximo, equivalente a estados F y R) así como **azares de convergencia**, que ocurren cuando dos o más señales con regiones de ambigüedad que se solapan en el tiempo confluyen en un mismo punto de un circuito y provocan un intervalo de ambigüedad en la salida del circuito. Analice también los **azares de ambigüedad acumulativa**, que ocurre cuando las señales se propagan a través de niveles de puertas. Es importante hacer notar que cuando el límite de ambigüedad de subida (R) se solapa con el de ambigüedad de bajada (F), se crea una región de indeterminación (X), con avisos por parte del simulador.
- 2.b) Modifique la temporización de los patrones de entrada para que se eliminen los azares de convergencia en los circuitos de las Fig 1.c y 1.d (asimilado al concepto de azares dinámicos en circuitos combinacionales).
- 3.- En este problema se tratará de distinguir los azares estáticos (provocados por cambios en una entrada, manteniendo las otras constantes) de los dinámicos estudiados en el problema anterior. Considere para ello el circuito de la Fig 1.d.
- 3.a) Simule todas las posibles combinaciones de entrada en el caso típico y caracterice las anomalías en el funcionamiento. Vuelva a simular el circuito considerando las entradas B y C colocadas a nivel alto (HI), mientras que la entrada A cambia su valor. Caracterice el **azar estático** para tiempos mínimo, típico y máximo, determinando la causa del azar.
- 3.b) Compruebe que el circuito de la Fig. 3 genera la misma función combinacional que el de la Fig 1.d. Compruebe asimismo que es una implementación libre de azares estáticos. Determine analíticamente el motivo.



4.- El circuito de la Fig. 4a es un sumador completo de un bit, con generación de suma (S) y acarreo (CO).

4.a) Obtenga la tabla de verdad del circuito de la Fig. 4a. Caracterícelo temporalmente, obteniendo los valores del retraso en los casos mínimo, típico y máximo.

4.b) Construya un símbolo para el sumador de un bit como el que se muestra en la Fig. 4b y que pueda ser empleado en el diseño de esquemáticos en niveles de jerarquía superior. La generación del símbolo puede realizarse desde la ventana del esquemático activando el menú *File => Symbolize*. Previamente se han de colocar en el esquemático los puertos de entrada y de salida con los elementos IF_IN y IF_OUT, respectivamente. Se le da el nombre *sumador* al símbolo y se incluye en una librería de símbolos propios, que llamaremos *propia.slb*. El símbolo ya está creado y guardado en dicha librería. Para poder editarlo y cambiarle la forma, ejecutaremos *File => Edit Library*; una vez aquí, estamos en una ventana de edición de símbolos. Para acceder a la librería de símbolos que hemos creado seleccionamos *File => Open => propia.slb*. Para editar el símbolo del sumador que hemos generado, seleccionamos *Part => Get => sumador* y aparece el símbolo generado por defecto. Para modificar los límites del símbolo (línea discontinua), es necesario seleccionar *Graphics => BBox* y redibujar el límite desde la esquina superior izquierda. Se puede modificar ahora la posición y características de los pines, chequear el esquemático con *Edit => Check Schematic*, o asociar otros esquemáticos al mismo símbolo con *Edit => Set Schematic*.

4.c) Utilizando este símbolo, construya un sumador de cuatro bits (Fig.4c) y compruebe su funcionamiento. Para poder incluir el símbolo sumador que hemos generado en un nuevo esquemático, se debe añadir la librería propia al conjunto de librerías accesibles al usuario. Para ello es necesario seleccionarla con *Options => Editor Configuration => Library Settings* y añadirla localmente al esquemático que vayamos a generar. Emplee notación de buses para el interfaz de entrada-salida. Para generar estímulos en las entradas es necesario emplear un estímulo por cada sublínea del bus, ya que generar un único estímulo para todo el bus no es posible con la versión de evaluación de Microsim. Simule las sumas mostradas a continuación, verificando el funcionamiento del sumador, considerando estímulos del tipo HI y LO para los buses A y B. Es necesario colocar el pin IF_OUT con el nombre S[3-0] en la salida.

A + B: 0000+0000; 0000+1111; 1111+0000; 1111+1111

Para comprobar el funcionamiento dinámico a la vez que se verifica el diseño, emplee estímulos DigStim variantes en el tiempo (para los bits del bus A) y estímulos tipo HI y LO para los bits del bus B. Realice varios análisis transitorios, considerando retrasos típicos, para cubrir las siguientes sumas:

A conmutando desde 0000 a 1111, siendo alternativamente B=0000, 0110, 1111.

Verifique el funcionamiento lógico, el paso por valores de suma espúreos (debidos a los azares de convergencia cumulativa) y la variación en el tiempo de propagación para cada suma. Para ello mida el tiempo máximo, mínimo y promedio, para la suma en el caso en que A conmute desde 0000 a 1111, siendo B=1111.

