

Memoria Proyecto Electrónica

Gonzalo Fernández Suárez

Condiciones: Tener activo el reloj en todo momento. Alta frecuencia de reloj para evitar fallos (Funcionamiento óptimo a 256 Hz).

1. Circuito Dado

He construido este circuito a partir de un **contador** conectado a **dos registros en serie**.

Al pulsar el botón "Tirar Dado" se guarda el valor del contador en los registros. Primero en el primero. Al pulsar por segunda vez, se guarda el valor de la primera tirada en el segundo registro y el de la nueva tirada (segunda) en el primero.

Así ya tenemos almacenadas las dos tiradas.

2. Circuito Acumulador de Puntuaciones

He implementado al Circuito Dado un **contador de las tiradas**. Si el número de tiradas es 0 no ocurre nada, el dado está en el aire (Está corriendo el contador).

Si el número de tiradas es igual a 2 **se guardan los valores** de ambas en el **registro 74198** y **se suman** para obtener la puntuación.

Si el número de tiradas es **múltiplo de 3**, se hace un **reset total** de los contadores y los registros. Vuelve a empezar el ciclo.

A la hora de hacer la suma podría haber utilizado el sumador que trae por defecto logisim obteniendo como salida del registro dos números de 4 Bits con el Bit más significativo siempre en 0, sin embargo, he preferido hacer un **sub-circuito que obtiene dos números de 3 Bits** (Los que salen del registro 74198) y devuelve la **suma completa** de estos en un **número de 4 Bits**. Simplemente por limpieza y gusto personal.

Este subcircuito lo he generado a partir de su tabla de verdad. Para verla: Click derecho sobre el circuito *SumCompleto2N3Bits* -> Analizar circuito -> Pestaña Tabla

3. Circuito Visualizador del Estado del Juego

He implementado túneles para mayor limpieza del circuito. También he reubicado algunos botones. Si el **circuito se apaga** se **resetean** todos los contadores y registros. Si alguno de los jugadores hace un **múltiplo de 3 tiradas** también se **resetea** el circuito, se vuelve a empezar la partida. Los **dados no funcionan** con el juego **apagado**.

He generado un sub-circuito *CircEstadoJuego* (Etiqueta "ESTADO") para gestionar el estado del juego. Esta es su tabla de verdad:

CircEstadoJuego

Inputs			Output
OnOff (0/1)(Pin de Inicio del Juego)	SumCont0 (Suma de contadores de tiradas == 0)	Pts0 (PuntosJ1 == 0 && PuntosJ2 == 0)	E (Indicador de estado Previo)
0	X	X	0000 (0)(Apagado y Reset)
1	1	1	1101 (d)
1	0	1	0001 (1)(Tirada de alguien)
1	0	0	0010 (2)(Hay ganador)
1	1	0	0010 (2)(Este caso es imposible que se dé)

Como se puede observar este sub-circuito devuelve **“0”** si **esta apagado** el juego, **“d”** si está **funcionando el dado**, **“1”** si ha habido alguna **tirada** y **“2”** si los jugadores ya **han hecho sus dos tiradas**.

Si E es igual a **“0”** o **“d”** simplemente **se envía el valor de E al Hex Digit Display**. Si E es igual a **“2”** se comparan las puntuaciones de los jugadores y se envía **“a”, “b” o “e”** en función de quien es el **ganador**. Si no se da ninguna de estas opciones se enviará al display la **mayor de las últimas tiradas** de los jugadores (No se especifica en el enunciado). Estas condiciones **activan o desactivan los transistores** que informan al Hex Digit Display sobre lo que debe mostrar.

Existe la remota posibilidad de que **ambos jugadores puntúen 0**, si en las dos tiradas obtienen 0. **Es casi imposible** que ocurra si se respetan los 256 Hz de reloj, pero de todas formas **lo he contemplado** en mi circuito. El **Caso Imposible** se da si los contadores de tirada están a 2 y la suma de puntuaciones es 0, el circuito tendería a pensar que quedan tiradas (Tabla de arriba) pero está configurado para que contemple este caso y **muestre el empate 0-0**.

El **Display del Estado del Juego** es el que tiene el **fondo negro y la cifra en verde**. El comparador de puntuaciones para ver el ganador es sin signo, siempre son puntuaciones positivas.

4. Circuito de Control del Estado del Juego

El Circuito de Control se encuentra a la **derecha del todo**.

El **apagado/encendido** ha pasado de ser un Pin a un **Registro** que guarda 1 cuando el Circuito de control marca 1 y se resetea al pulsar el botón *Full Reset*. Si el contador del Control del Juego **llega a 0** directamente **carga 1** en el contador, porque en el enunciado no se contemplaba el 0.

En este circuito he implementado **Bit Extenders** y un **Multiplexor** para gestionar el envío de las tiradas al Display en el orden requerido.

En cuanto al resto, el circuito hace exactamente lo descrito en el enunciado.