**SEÑALES DE ENTRADA**

**Clock\_100Hz:** Señal de reloj que sirve para coordinar las acciones de los bloques de nuestro circuito.

**reset:** Obliga al sistema a entrar en el estado T1, no importando el estado anterior en el que se hubiere encontrado.

**start:** Una vez estemos en el estado T1 (luego de presionar reset) permite pasar al estado de activación de nuestro circuito (estado T3), donde se podrá elegir entre jugar o mostrar las 10 mejores puntuaciones.

**jugar:** Permite pasar al estado T4 que será el principio de la secuencia de estados que se encargarán de la parte del juego en sí de nuestro sistema.

**puntuaciones:** Muestra las 10 mejores puntuaciones guardadas (a intervalos de 3 segundos) en la memoria del sistema.

**dado1:** Activará el proceso para obtener el valor del dado 1.

**dado2:** Activará el proceso para obtener el valor del dado 2.

**tcont3:** Señal que le indicará a nuestra MSS cuando el bloque CONTADOR\_3\_SEGUNDOS ha completado el conteo de 3 segundos, permitiendo así el paso al siguiente estado (utilizado para controlar un tiempo de espera de 3 segundos desde que se revela el valor del dado hasta que es el turno del siguiente jugador).

**dcont3:** Señal que le indicará a nuestra MSS cuando el bloque CONTADOR\_3\_SEGUNDOS ha completado el conteo de 3 segundos, permitiendo así el paso al siguiente estado (utilizado para controlar un tiempo de espera de 3 segundos desde que se presiona la respectiva botonera de dado del turno del jugador correspondiente, hasta cuando finalmente pasa este tiempo, y así muestre el valor de dicho dado en el display de 7 segmentos).

**gcont5:** Señal que le indicará a nuestra MSS cuando el bloque CONTADOR\_5\_SEGUNDOS ha completado el conteo de 5 segundos, permitiendo así el paso al siguiente estado (esta señal está encargada de controlar los 5 segundos que tienen que pasar mientras se va mostrando la puntuación que alcanzó el jugador ganador, el número de jugador que ganó la partida, y el parpadeo de los 20 LEDs del tablero de juego).

**spevent:** Señal que sirve para avisar al sistema digital cuándo ha finalizado el proceso de mostrar las 10 mejores puntuaciones y, de esta manera, enviar al sistema al estado T3 (estado de activación).

**finj1:** Señal que sirve para indicar al sistema si el jugador 1 ha ganado la partida (la cual se activará cuando el valor del registro de casillas del jugador 1 haya llegado a 20), para así enviar al sistema al estado de fin de juego, donde se mostrará durante 5 segundos la puntuación obtenida por el jugador 1 en los displays de 7 segmentos, el número 1 en el otro display de 7 segmentos, y mientras parpadean todos los LEDs del tablero.

**finj2:** Señal que sirve para indicar al sistema si el jugador 2 ha ganado la partida (la cual se activará cuando el valor del registro de casillas del jugador 2 haya llegado a 20), para así enviar al sistema al estado de fin de juego, donde se mostrará durante 5 segundos la puntuación obtenida por el jugador 2 en los displays de 7 segmentos, el número 2 en el otro display de 7 segmentos, y mientras parpadean todos los LEDs del tablero.

**dadodif:** Señal que sirve para indicar si el valor de los dos primeros lanzamientos de dados (los que determinan el orden de juego de los jugadores) son iguales, si es así, el sistema regresará al estado T5, que es el estado donde se espera la activación de la botonera de dado1 para volver a repetir el proceso, y así sucesivamente hasta que el valor del dado 1 y dado 2 sean diferentes, cuando esto se cumpla, el sistema avanzará al estado T13, que es el primero de la secuencia de estados del juego en sí (esperando que el jugador 1 presione la botonera del dado correspondiente).

**sf1:** Señal que servirá para indicarle a nuestra MSS que el registro de casillas y puntos del jugador 1 se ha reiniciado (valores a cero), con lo cual nuestro controlador pasará del estado TF1 al estado T3 (estado de activación).

**sf2:** Señal que servirá para indicarle a nuestra MSS que el registro de casillas y puntos del jugador 2 se ha reiniciado (valores a cero), con lo cual nuestro controlador pasará del estado TF2 al estado T3 (estado de activación).

**SEÑALES DE SALIDA**

**ledj1:** Señal que encenderá el LED que indicará que el turno actual le corresponde al jugador 1.

**ledj2:** Señal que encenderá el LED que indicará que el turno actual le corresponde al jugador 2.

**sdcont1:** Activa el bloque contador que se encargará de contar 3 segundos luego de que se haya presionado la botonera del dado 1 (en su respectivo estado de juego), para después enviar una señal de finalización de conteo y así mostrar el valor del dado.

**sdcont2:** Activa el bloque contador que se encargará de contar 3 segundos luego de que se haya presionado la botonera del dado 2 (en su respectivo estado de juego), para después enviar una señal de finalización de conteo y así mostrar el valor del dado.

**stcont:** Activa el bloque contador que se encargará de contar 3 segundos, para después enviar una señal de finalización de conteo (utilizado para controlar un tiempo de espera de 3 segundos desde que se revela el valor del dado hasta que es el turno del siguiente jugador).

**d7s1:** Señal que provocará que el mux cambie su salida para así mostrar el valor del dado 1 en el display de 7 segmentos.

**d7s2:** Señal que provocará que el mux cambie su salida para así mostrar el valor del dado 2 en el display de 7 segmentos.

**punt1:** Señal que provocará que el mux cambie su salida para así mostrar el valor de la puntuación obtenida por el jugador 1 (durante 5 segundos después de que el jugador 1 haya alcanzado la casilla 20), cuando éste ha ganado la partida.

**punt2:** Señal que provocará que el mux cambie su salida para así mostrar el valor de la puntuación obtenida por el jugador 2 (durante 5 segundos después de que el jugador 2 haya alcanzado la casilla 20), cuando éste ha ganado la partida.

**tab1:** Señal que provocará que el mux cambie su salida para así encender la cantidad de LEDs del tablero acorde al valor de casillas que haya avanzado el jugador 1 (se activará 3 segundos después de presionar la botonera del dado en su respectivo turno).

**tab2:** Señal que provocará que el mux cambie su salida para así encender la cantidad de LEDs del tablero acorde al valor de casillas que haya avanzado el jugador 2 (se activará 3 segundos después de presionar la botonera del dado en su respectivo turno).

**vdado1:** Señal cuya activación (instante en el que su valor cambia de 0 a 1) activará a su vez, valga la redundancia, el bloque de registro del jugador 1 para que haga los respectivos cálculos internos y así determine los valores nuevos de casillas avanzadas y puntuación alcanzado.

**vdado2:** Señal cuya activación (instante en el que su valor cambia de 0 a 1) activará a su vez, valga la redundancia, el bloque de registro del jugador 2 para que haga los respectivos cálculos internos y así determine los valores nuevos de casillas avanzadas y puntuación alcanzado.

**fin:** Señal que le indicará a los bloques de nuestro sistema digital que el juego ha terminado, volviendo así los valores de los registros de casillas y puntuaciones de cada jugador a cero, “reiniciando” el juego (la diferencia con el reset además es que los valores de puntuaciones guardadas no se ven afectadas por esta señal).

**d7sgana1:** Señal que provocará que el mux cambie su salida para así mostrar en el display de 7 segmentos el valor “1” durante 5 segundos (indicando que el jugador 1 es el ganador). Esta señal se activará una vez que el jugador 1 haya ganado la partida.

**d7sgana2:** Señal que provocará que el mux cambie su salida para así mostrar en el display de 7 segmentos el valor “2” durante 5 segundos (indicando que el jugador 2 es el ganador). Esta señal se activará una vez que el jugador 2 haya ganado la partida.

**enpcont:** Señal que activará el proceso dentro del bloque RAM\_Punt que mostrará las 10 mejores puntuaciones a intervalos de 10 segundos (activa el proceso de contadores para que se cumpla este cometido).

**endadodif:** Señal que activará el bloque COMPARADOR\_DADO que será el que hará la comparación del valor del dado 1 con el del dado 2, teniendo como salida el indicador que permitirá que nuestra MSS continúe al proceso del juego si estos valores son distintos o que vuelva al estado T5 si son iguales, que es donde se espera presionar la botonera de dado 1 y después presionar la del dado 2 para obtener nuevos valores de dados, y así sucesivamente hasta que éstos sean diferentes.

**par:** Señal que, en conjunto con la activación de tab1 o tab2, provocará que el mux cambie su salida para así mostrar el parpadeo de los 20 LEDs.

**res:** Señal que provocará que todos los flip flops de nuestro sistema se reinicien a un valor de cero (esta señal se activará cuando nuestro sistema entre al estado T1, es decir, cuando presionemos la botonera reset).

**pdado1:** Señal que será el enable del bloque CONTADOR\_DADO para el dado 1, trabajando de la siguiente manera: Durante el tiempo que la señal esté activada (tiempo que dure el estado siguiente al estado donde presionamos la botonera del dado 1, en otras palabras el tiempo en el que la botonera pasa siendo presionada), el contador estará contando a una velocidad de reloj mayor que la del circuito en sí (siendo la señal clock para este bloque de 1MHz, y la señal de reloj para todo lo demás del circuito de 100 Hz), siendo esta velocidad tan alta, el valor obtenido cuando soltemos la botonera (o sea pase al siguiente estado nuestro MSS) será prácticamente aleatorio, y es de esta forma como se obtiene el valor random del dado 1.

**pdado2:** Señal que será el enable del bloque CONTADOR\_DADO para el dado 2, trabajando de la siguiente manera: Durante el tiempo que la señal esté activada (tiempo que dure el estado siguiente al estado donde presionamos la botonera del dado 2, en otras palabras el tiempo en el que la botonera pasa siendo presionada), el contador estará contando a una velocidad de reloj mayor que la del circuito en sí (siendo la señal clock para este bloque de 1MHz, y la señal de reloj para todo lo demás del circuito de 100 Hz), siendo esta velocidad tan alta, el valor obtenido cuando soltemos la botonera (o sea pase al siguiente estado nuestro MSS) será prácticamente aleatorio, y es de esta forma como se obtiene el valor random del dado 2.