



UNIVERSIDAD COMPLUTENSE DE MADRID

SISTEMAS EMPOTRADOS DISTRIBUIDOS

PONG:

Diseño e implementación sobre
FPGA y ARM

Autores:

Luis María Costero Valero
Jesús Javier Doménech Arellano

Mayo 2016

Índice

1. Introducción	2
1.1. El juego	2
1.2. Descripción del proyecto	2
2. Diseño	4
2.1. Componentes del sistema	4
2.2. Comportamiento del sistema	4
2.3. Comunicación entre componentes	5
3. Implementación	8
3.1. FPGA	8
3.1.1. Gestión de los relojes	9
3.2. Maletines ARM	10
4. Conclusiones	12

Índice de figuras

1. Imagen del juego original.	2
2. Descripción general del sistema.	3
3. Diagrama de Actividad del sistema (sin VGA).	5
4. Diagrama de secuencia entre los distintos componentes.	7
5. Diagrama de componentes implementados en la FPGA.	8
6. Distintos relojes utilizados.	9
7. FSM describiendo el comportamiento de los maletines.	10

1. Introducción

1.1. El juego

El videojuego *Pong* es uno de los videojuegos más conocidos por todos, publicado por Atari en 1972. *Pong* consiste en un juego para dos personas, en el que el objetivo es hacer rebotar a una pelota para que el oponente no pueda golpearla y pierda.

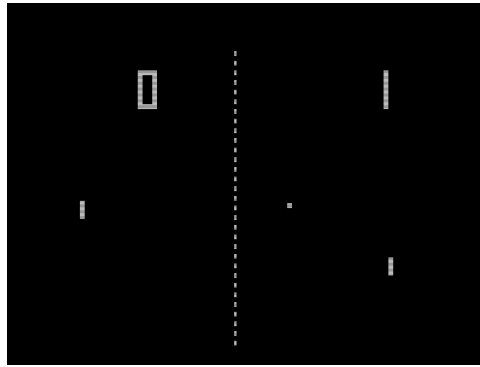


Figura 1: Imagen del juego original.

Para controlar la pelota, cada jugador dispone de una raqueta situada en uno de los extremos de la pantalla, la cual únicamente puede desplazarse en vertical. Cuando la pelota golpea una de estas raquetas, ésta cambia el sentido horizontal de movimiento, dirigiéndose al otro jugador. Según en que parte de la raqueta golpee la bola, ésta también modifica su componente de movimiento vertical. De igual manera, cuando la pelota golpea el borde superior o inferior del juego, también se modifica su movimiento. Se considera que un jugador pierde cuando no es capaz de golpear la pelota y ésta sale del juego por su lado de la pantalla.

1.2. Descripción del proyecto

El trabajo realizado consiste en el diseño e implementación del juego *Pong* sobre una arquitectura distribuida formada por una FPGA Spartan 3 [1], y dos maletines ARM S3C44B0X [2], comunicados entre ellos mediante UARTs.

La FPGA es el componente encargado de la lógica del juego. En él se realizan todos los cálculos de posiciones y choques de la pelota. También se lleva control de las puntuaciones. Además, sobre la FPGA se han implementado dos módulos adicionales, uno encargado de la comunicación con los maletines mediante UARTs, y otro para mostrar el juego en un monitor VGA.

Los maletines ARM son utilizados para leer los movimientos de los jugadores (mediante pulsadores y teclado matricial) y comunicárselo a la FPGA mediante una UART. Además, por motivos de diseño de la FPGA (sólo posee un puerto UART), los maletines están diseñados para ser conectados en serie y retransmitir los mensajes de un maletín al

siguiente, hasta llegar a la FPGA. La puntuación del juego también es mostrada por los *display* 8-segmentos de la placa.

En la figura figura 2 se puede ver una descripción general del sistema.

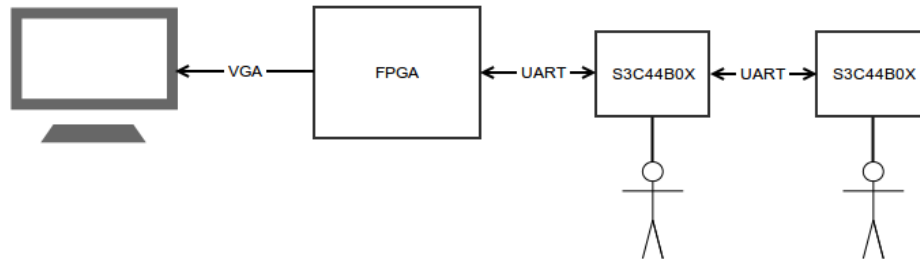


Figura 2: Descripción general del sistema.

La memoria está organizada de la siguiente forma: en la sección 2 se muestra el modelado realizado por el sistema, y diversos diagramas que describen el comportamiento del mismo. La sección 3 muestra los aspectos más relevantes en la implementación. Y para finalizar, la sección 4 muestra unas conclusiones y trabajo futuro a realizar si se desea seguir con el proyecto.

2. Diseño

2.1. Componentes del sistema

A la hora de realizar el sistema, se ha decidido utilizar los siguientes componentes:

- *FPGA*, que como se ha comentado en la sección anterior, es la encargada de realizar los cálculos del juego, interpretar las órdenes de los usuarios, y mostrar el juego mediante un monitor VGA.
- *Maletines ARM*, encargados de leer las órdenes de los jugadores (mediante pulsaciones en el teclado matricial o los botones de la placa) y transmitirlos a la FPGA mediante una UART. Como la FPGA utilizada únicamente posee una UART, los maletines se pueden conectar en serie y transmitir los mensajes de uno hacia el siguiente hasta que lleguen a la FPGA (ver sección 3.2). Además, la FPGA transmite la información de puntuación a los maletines, los cuales la muestran mediante sus *Display 8-Segmentos*.
- *Display 8-Segmentos*, utilizado para mostrar la información de puntuación.
- *Pulsadores y teclado matricial*, para que los jugadores indiquen sus órdenes de movimiento al sistema.
- *Monitor VGA*, utilizado para mostrar el juego.

A continuación se muestra tanto el comportamiento global del sistema, como la comunicación entre los distintos componentes. En la sección siguiente se muestra con más detalle el funcionamiento del componente FPGA (apartado 3.1) y de los maletines ARM (apartado 3.2).

2.2. Comportamiento del sistema

El diseño elegido hace que el núcleo del sistema sea la FPGA, encargada del control del juego, mientras que los maletines ARM actúan de periféricos, modificando el estado del juego.

Esto hace que el comportamiento principal del sistema, resumido en la figura 3, consista en un bucle principal, siguiendo los pulsos del reloj de juego, donde se va calculando el estado actual del juego, y se reciban modificaciones externas de manera distribuida que sirven para modificar los distintos parámetros que componen el escenario, en este caso la posición de las raquetas de cada jugador.

Por otro lado, la FPGA notifica a los maletines el estado del juego en lo relativo a las puntuaciones. Esta forma de abstraer el control de los jugadores de la lógica del juego nos permitiría por ejemplo, sustituir el control por un teclado PS2, o cualquier otro periférico, como pudiera ser un puerto de red.

El sistema diseñado e implementado en los maletines es más sencillo (explicado en detalle en la sección 3.2). Ambos maletines actúan de igual manera, independientemente del orden en el que se conecten. Los maletines envían las pulsaciones del teclado por una de las UARTs, y en caso de recibir información por la otra, esta se trata como si viniera de otro maletín, modificándose y reenviándose hacia la FPGA. Este método permite conectar en serie cualquier cantidad de maletines, permitiendo tener un juego de N jugadores si fuese necesario¹. Además, desde el punto de vista de la FPGA se trata de un único maletín o teclado en el que juegan todos los jugadores de manera simultánea.

Este sistema podría mostrar la debilidad de un posible retardo en las acciones realizadas por el jugador del maletín más lejano pero esto se compensa tal y como se explica en la sección 3.1.1 al llevar diferentes relojes en cada parte del sistema; la recepción de comandos es mucho más rápida que el reloj de movimiento del juego así que por mucho que llegase un poco más tarde, llegaría antes del siguiente ciclo de reloj de movimiento.

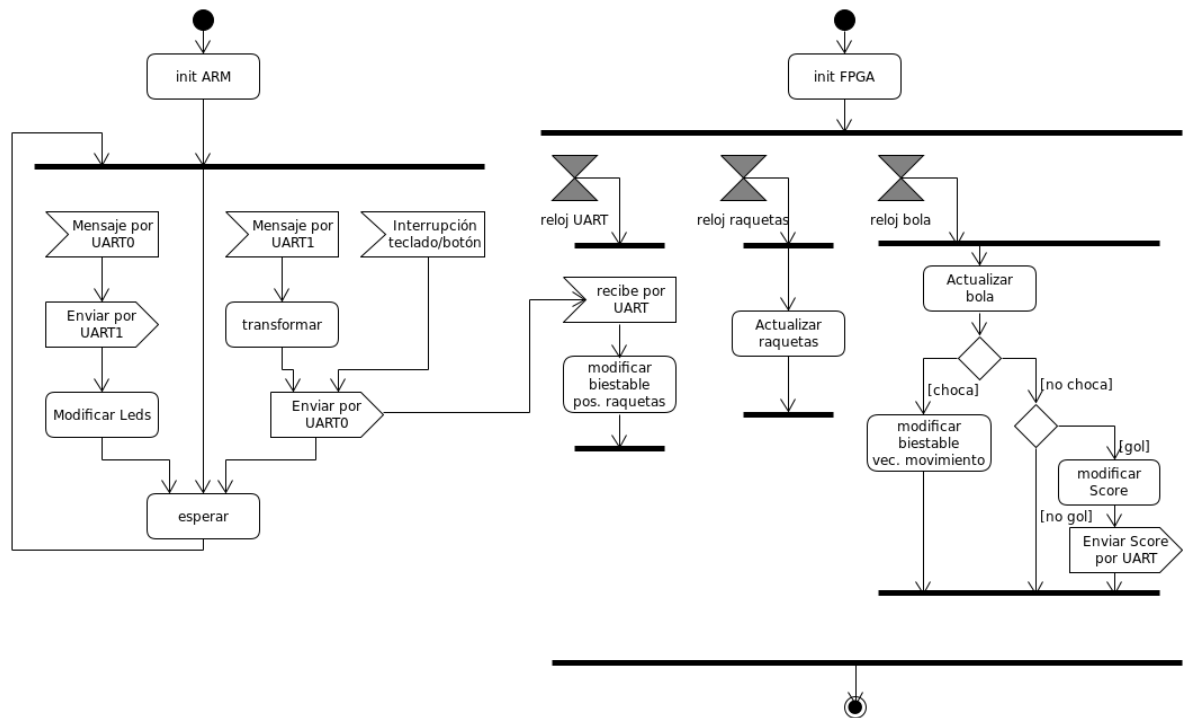


Figura 3: Diagrama de Actividad del sistema (sin VGA).

2.3. Comunicación entre componentes

Como se ha mencionado anteriormente, el sistema está controlado por una FPGA, la cual ejecuta varios bucles de manera paralela para modificar el estado del juego y pintar en el monitor VGA los elementos del juego. El resto de componentes (maletín, display

¹En la última sección 4 pueden verse detalles de como podría convertirse en un juego de cuatro jugadores

8-Segmentos, pulsadores, ...) sirven para modificar el estado del juego mediante comunicaciones con la FPGA, que permiten que esta calcule las nuevas posiciones. Según qué componentes interactúen entre ellos, la comunicación usada es diferente:

- *FPGA – VGA*: Comunicación unidireccional utilizando el conector VGA. Se utilizan 9 líneas para los colores (3 para cada color), y 2 líneas adicionales para la sincronización horizontal y vertical respectivamente. (Más información se puede encontrar en [3]).
- *Pulsadores – Maletín*: Detección de pulsaciones mediante interrupciones en la placa.
- *Maletín – Maletín*: Comunicación mediante UARTs. El diseño utilizado permite conectar varios maletines en serie, por lo que el comportamiento general de los maletines es recibir la información por una UART, procesarla y reenviarla por la otra UART. Si la información es recibida por la UART0, se trata de otro maletín que envía información a la FPGA, mientras que si se recibe por la UART1, es la FPGA la que envía información a los maletines.
- *FPGA – Maletín*: Comunicación mediante UART. Como la FPGA únicamente dispone de un puerto UART, la conexión de los maletines se hace en serie, siendo los maletines intermedios encargados de reenviar la información en ambos sentidos (hacia la FPGA o hacia el resto de maletines).

La figura 4 muestra de manera resumida la comunicación entre los distintos componentes. Como se puede observar, la FPGA posee dos grandes bucles² encargados de calcular el estado del juego, y de pintar en el monitor. Cuando un jugador pulsa un botón, la información es transmitida a la FPGA a través de los maletines, modificando el cálculo del estado del juego. En el caso de que un jugador falle, la FPGA envía información hacia los maletines para informar de la nueva puntuación.

²Más información sobre los bucles de la FPGA, y de manera especial sobre los distintos relojes utilizados, se puede encontrar en 3.1.1.

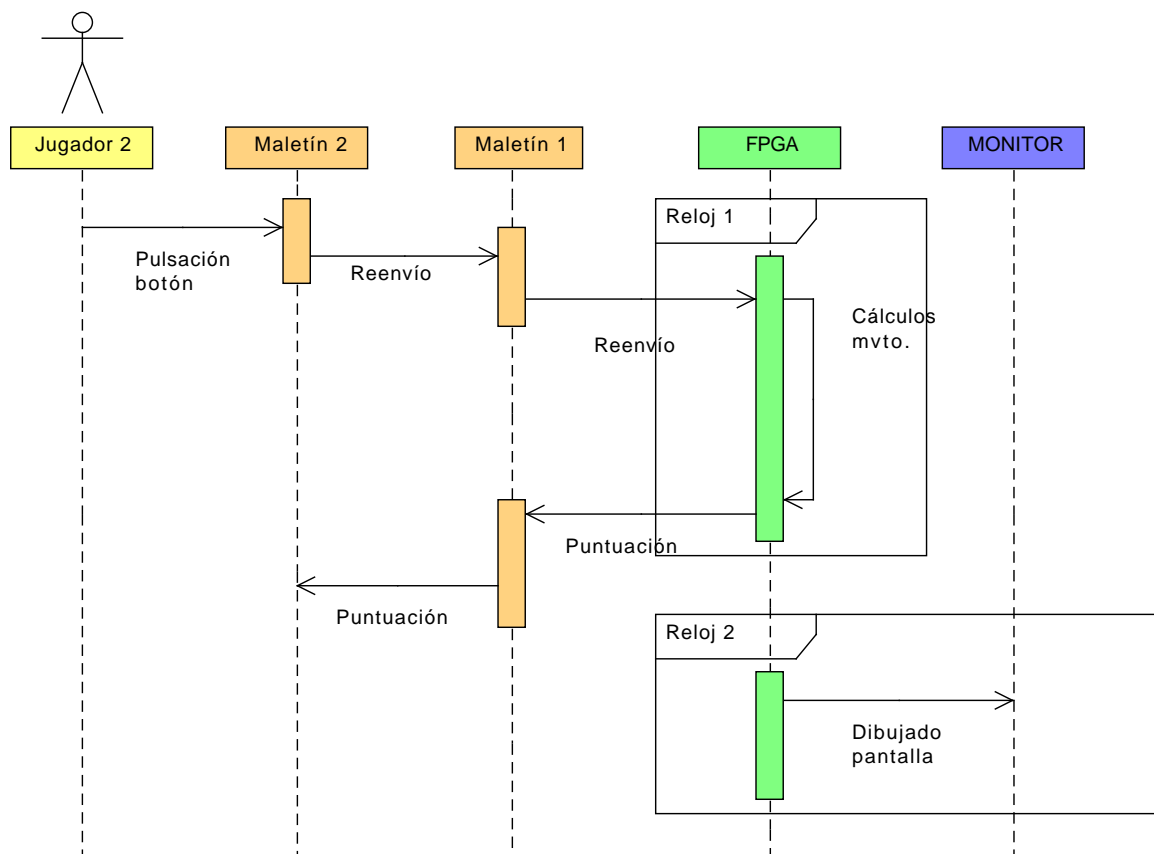


Figura 4: Diagrama de secuencia entre los distintos componentes.

3. Implementación

3.1. FPGA

Para la implementación del juego sobre la FPGA, se ha decidido implementar distintos módulos, conectados entre ellos, que permiten calcular el estado del juego. La relación entre los distintos módulos se puede ver en la figura 5. Además, para que el juego pueda funcionar a una velocidad adecuada, pero se pueda comunicar con los maletines, y además pintar en el monitor VGA, cada módulo está controlado por su propio reloj, como se explica en la sección 3.1.1. y se muestra en la figura 6.

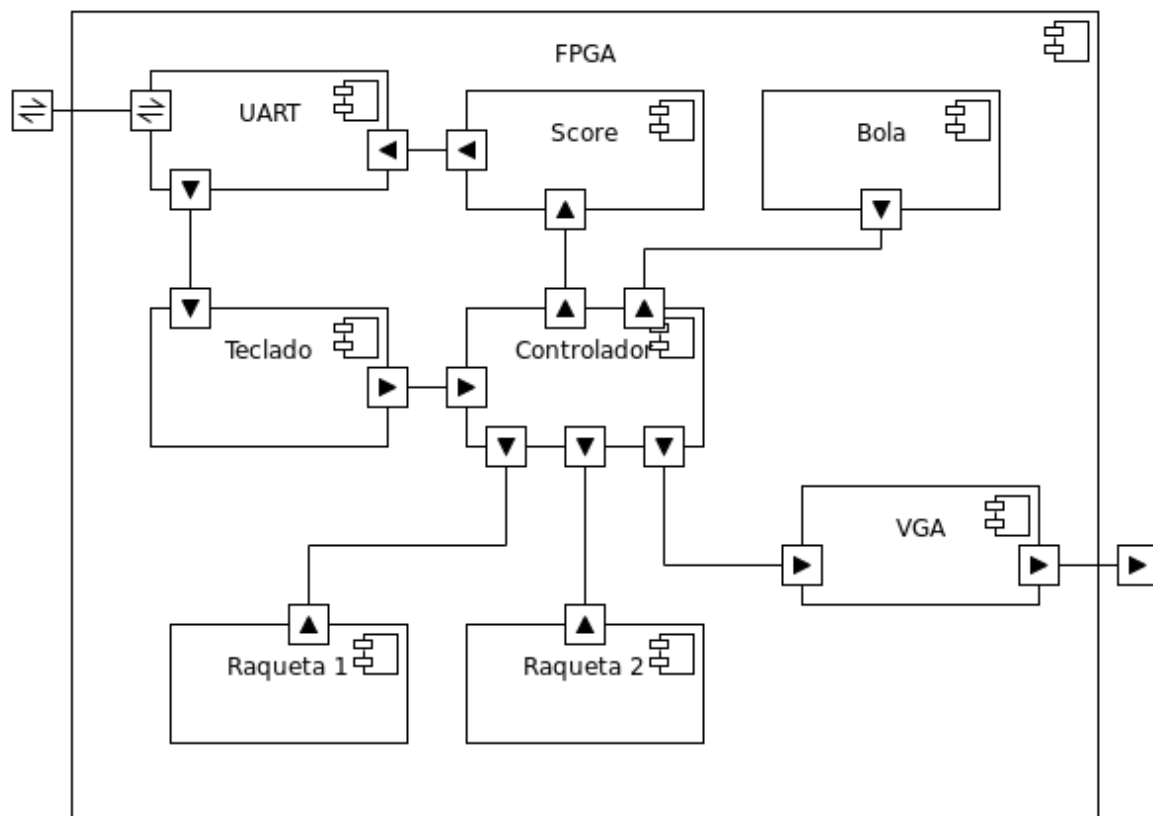


Figura 5: Diagrama de componentes implementados en la FPGA.

Como se muestra en la figura 5, los módulos de los que está compuesta la implementación sobre la FPGA son:

- *Módulo UART*: módulo encargado de leer y enviar información a través de la UART de la FPGA. Posee señales de control para indicar cuándo un dato se ha recibido, o cuando está ocupada enviando información.
- *Módulo teclado*: traduce la información recibida por la UART a comandos internos entendidos por el juego. Este diseño nos permite desacoplar el módulo teclado de la UART y cambiarlo por otros módulos que interpreten otros dispositivos de entrada, como puede ser un teclado PS2.

- *Módulo raquetas:* contienen la información de la posición de las raquetas. Las teclas leídas del módulo teclado modifican la posición de este módulo. Este módulo se encuentra instanciado varias veces, una por cada raqueta del juego.
- *Módulo bola:* calcula en cada pulso del reloj del juego la nueva posición de la pelota. Además, recibe la información de las palas para calcular los rebotes con ellas, o si un jugador gana el partido actual. Si es el caso, el módulo informa al módulo controlador para que anote e informe del cambio de puntuación.
- *Módulo puntuación:* encargado de guardar la información de los partidos. Cuando un jugador consigue un tanto, se comunica con el módulo de la UART para enviar la información a los maletines.
- *Módulo VGA:* a partir de la información del módulo bola y de los módulos de raquetas, dibuja en un monitor VGA el juego.
- *Módulo controlador:* encargado de llevar la máquina de estados principal del juego y comunicar los diferentes módulos entre sí.

3.1.1. Gestión de los relojes

Aunque el diseño y ejecución del juego sobre la FPGA tiene muchas ventajas (la mayoría relativas a la facilidad de realizar cálculos en paralelo), es necesario sincronizar cada uno de los módulos entre ellos. Además, el diseño elegido para la implementación hace que cada módulo utilice un reloj distinto. Los distintos relojes utilizados se muestran en la figura 6, donde cada color refleja el reloj utilizado en los módulos que se indican.

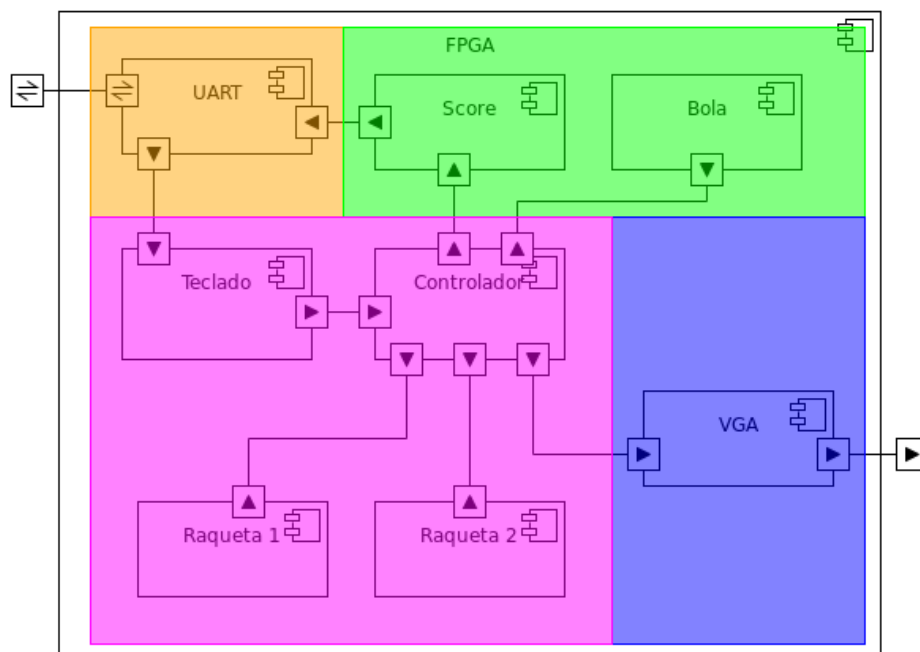


Figura 6: Distintos relojes utilizados.

Más concretamente se ha decidido utilizar cuatro relojes sobre la FPGA, cada uno con un propósito distinto:

- *Reloj principal (rosa)*: reloj principal de la placa. Proporciona un pulso cada 100 ns. Es utilizado para leer los datos procesados en tiempo real por la UART e informar a las raquetas de su nueva posición. Se ha decidido utilizar un reloj rápido ya que se consigue reaccionar de manera más rápida a las pulsaciones del jugador. Para no perder datos con el módulo de la UART (al tener distintos relojes), la información es almacenada en un biestable hasta que ésta es leída.
- *Reloj UART (naranja)*: se trata del mismo reloj utilizado para la transmisión con la UART en los maletines ARM.
- *Reloj de juego (verde)*: utilizado para calcular los movimientos de la pelota. Se trata de una división del reloj principal del juego, que al ser más lento permite visualizar los movimientos por la pantalla y hacer que el jugador sea consciente de lo que ocurre y le dé tiempo a reaccionar.
- *Reloj VGA (azul)*: necesario para transmitir por el protocolo VGA. Según este protocolo, el reloj debe ir a 25MHz (un cuarto del reloj principal de la placa).

3.2. Maletines ARM

Como se ha dicho anteriormente, la FPGA utilizada solamente posee una conexión UART, por lo que el diseño elegido para los maletines es conectarlos en serie. Además, todos los maletines ejecutan el mismo código, encargado de transmitir la información de una UART hacia la otra. La figura 7 muestra el comportamiento de los maletines.

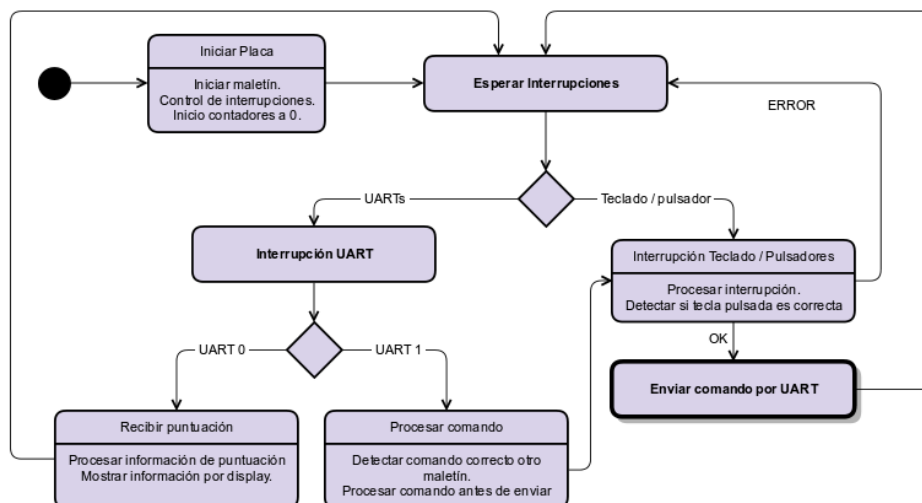


Figura 7: FSM describiendo el comportamiento de los maletines.

Al iniciar las placas, lo primero ejecutado es la inicialización de la misma, las conexiones UARTs, activación de interrupciones y marcar los contadores de puntuación a 0. A continuación se entra en un bucle que va a controlar el juego. Si se recibe una interrupción de

un pulsador o del teclado matricial, esta pulsación se procesa y se envía por la UART1 al siguiente elemento de la cadena (que puede ser otro maletín o la FPGA). Si se detecta la interrupción asociada a un puerto UART, pueden distinguirse dos casos:

- Interrupción en UART0. Indica que la información recibida proviene de otro maletín. En este caso, se procesa la información para indicar que no la ha generado el maletín, y se envía por la UART1 al siguiente elemento de la cadena (que va a ser la FPGA en el caso de tener únicamente dos maletines).
- Interrupción en UART1. Indica que la información recibida proviene o bien de la FPGA, o bien la está retransmitiendo el otro maletín. La información es procesada (información relativa a la puntuación), y mostrada por los display 8-Segmentos.

4. Conclusiones

En lo expuesto en esta memoria y el trabajo realizado creemos que se ha realizado con éxito el juego *Pong* utilizando maletines ARM y una FPGA comunicados mediante UARTs y visualizado en un monitor VGA.

El trabajo realizado supera nuestras expectativas iniciales de conectar sólo dos maletines al añadir la FPGA y un monitor VGA.

Somos conscientes de algunas deficiencias en el sistema de comunicación que, aún siendo funcional, podrían mejorar. Se deja para un posible trabajo futuro junto con la idea de añadir más jugadores.

Esta última idea no debería ser complicada, ya que el sistema de propagación de comandos y puntuaciones en los maletines no tiene número máximo establecido, las limitaciones se nos plantean en la lógica del juego que está pensado para máximo dos jugadores, pero sustituyendo los muros superior e inferior por raquetas de los nuevos jugadores, creemos viable realizar la ampliación a cuatro jugadores.

Queremos por último mostrar nuestra satisfacción con el trabajo realizado ya que consideramos haber conseguido integrar en un proyecto lo aprendido en diferentes asignaturas tanto del máster como del grado.

Referencias

- [1] XESS Corp. - X Engineering Software Systems Corporation, *XSA-3S1000 Board V1.0 User Manual*, 1.0 ed., 2001.
- [2] Samsung Electronics, *S3C44B0X RISC Micropocessor*.
- [3] Xilinx, *Spartan-3 FPGA Starter Kit Board User Guide*, v1.2 ed., June 20 2008. Págs. 22-27.