




Trasmisor de señal.

Fernández-Palacios Serrano, Jaime.
González Bonet, Santiago de Jesús.
Linares Serrano, Pablo.



INDICE

1 BLOQUES.

1.1 PPDU.

1.2 CONVOLUCIONADOR.

1.3 SCRAMBLER.

1.4 INTERLEAVER.

1.5 MAPPER.

1.6 IFFT.

1.7 PUERTO SERIE.

1.8 Shift.

2 COMENTARIOS SOBRE RESULTADOS.

3 RECURSOS DE LA FPGA.

4 WARNINGS.

5 DIAGRAMA DE BOLAS.

PPDU

El Bloque de PPDU trata de la parte externa al sistema porque que guarda la memoria con los datos a enviar. Estos datos están internamente guardados. Cuando pulsamos y soltamos el botón de activación, vamos cogiendo los datos de la memoria, y se lo pasamos al Convolucionador cuando éste nos indique que está libre.

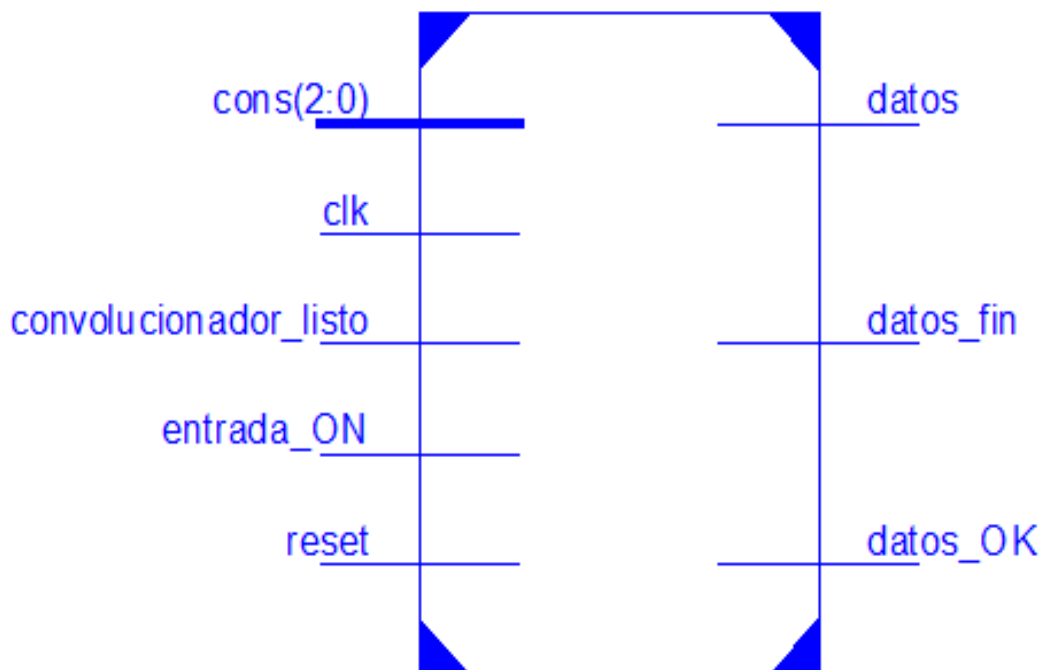
Todo el envío se transmite recorriendo la tabla, incrementando tanto la posición del bit como la dirección. Una vez que termina de transmitir todos los datos termina el envío y enviamos el bit de finalización.

Pines de entrada:

- **clk**: Señal de reloj del sistema. (Se repite en cada bloque)
- **reset**: Reset asíncrono del sistema. (Se repite en cada bloque)
- **entrada_ON**: Botón que da comienzo a la transmisión. Todo comienza una vez que pulsamos y soltamos.
- **convolucionador_listo**: Señal procedente del convolucionador que nos indica que éste ha tomado el dato que tenemos ha la salida. Solo podrá activarse si datos_OK está a 1.
- **cons**: Señal de 3 bits que nos indica la constelación del sistema. Procederá de 3 switches.

Pines de salida:

- **datos**: Bit que enviamos al bloque del sistema que contiene la información que enviamos sobre el carácter.
- **datos_fin**: Bit que indica el final de la transmisión de datos.
- **datos_OK**: Bit de señalización que indica que el bit puesto en datos es correcto.



CONVOLUCIONADOR

El convolucionador se encarga de combinar los bits de entrada que vamos recibiendo a través de unos registros y combinarlos en dos salidas diferentes.

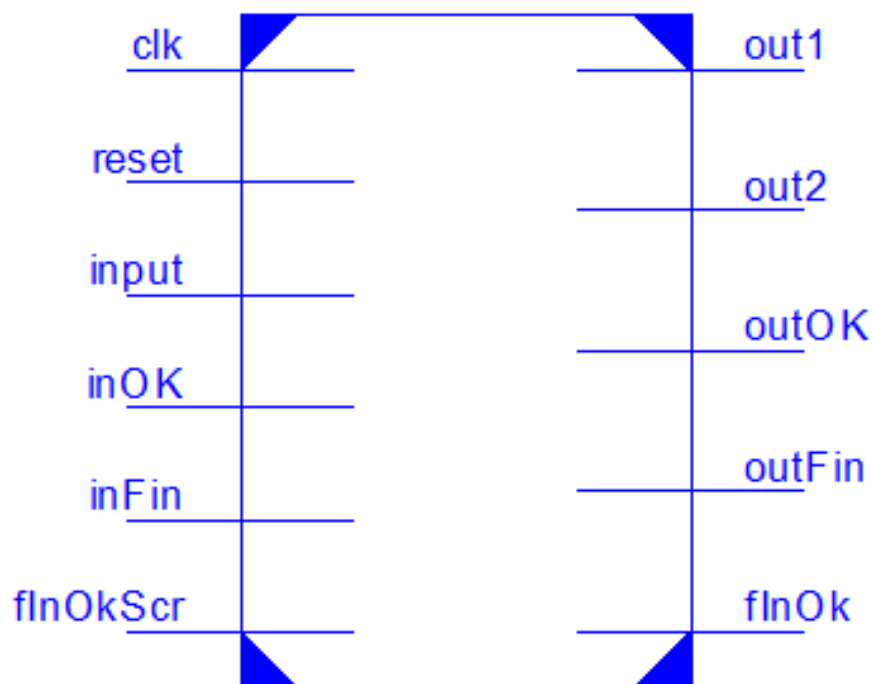
Cuenta con un registro de 6 bits en el sistema, que inicialmente vamos rellenando uno a uno y desplazando los datos. Una vez que terminamos de transmitir, vamos rellenando los registros de ceros del mismo modo que anteriormente, uno a uno y desplazando el dígito. Los datos los vamos introduciendo por el registro más significativo.

Pines de entrada

- **inFin:** Señal que indica la finalización de datos recibidos.
- **input:** Bit que recibimos desde la PPDU.
- **inOK:** Señal que nos indica que está activado el envío del bloque anterior.
- **flnOkScr:** Señal que indica que el Scrambler ha tomado el bit que hemos puesto en la out1 y out2.

Pines de salida

- **outFin:** Señal que indica el fin de transmisión de datos.
- **outOK:** Bit que indica que está activo el envío de datos. Esta señal se usa también para indicar que el convolucionador está listo y se envía a la PPDU.
- **out1:** Bit generado en la primera salida por la combinación de los registros.
- **out2:** Bit generado en la segunda salida por la combinación de los registros
- **flnOk:** Señal de feedback, que confirma a la PPDU que hemos tomado el dato proporcionado.



SCRAMBLER

Este bloque se encarga de aleatorizar el flujo de bits. Evitamos que se nos acumulen muchos unos o ceros seguidos.

Aquí tenemos un primer estado inicial en el que estamos esperando a que tengamos tanto el interleaver preparado como que hayamos recibido un dato. Luego lo combinamos con un valor del último registro interno, desplazamos el registro, y como nuevo valor, hacemos una combinación del cuarto y el último registro.

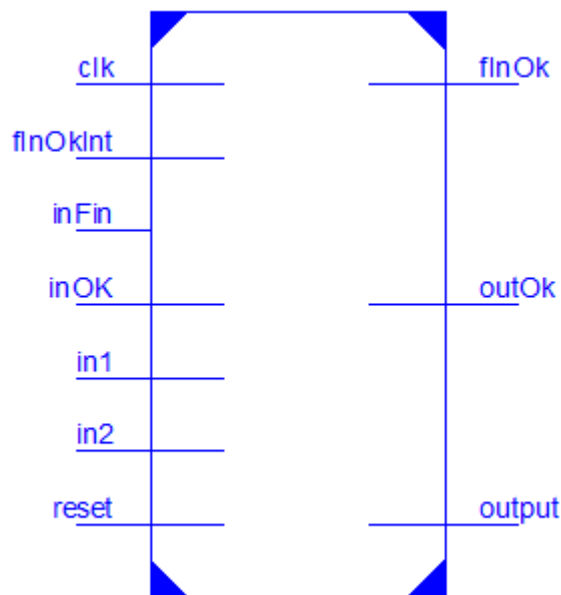
Y así repetimos este proceso primero con la primera entrada, y después con la última, hasta que lo hayamos hecho con todos los datos.

Pines de entrada

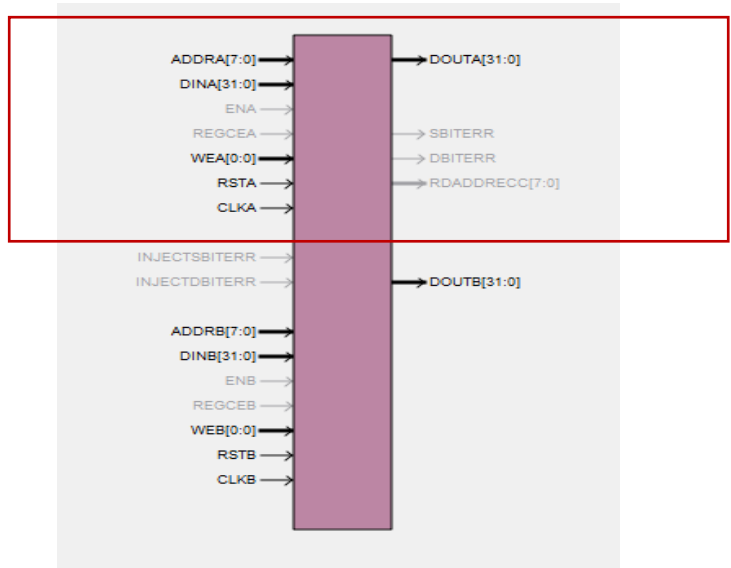
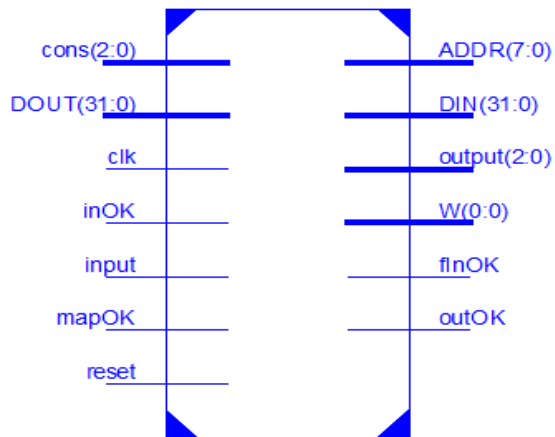
- **in1**: Bit de la primera entrada al que vamos a hacer lectura.
- **in2**: Bit de la segunda entrada al que vamos a hacer lectura.
- **inOK**: Señal que nos indica que está activado el envío del bloque anterior.
- **inFin**: Señal que indica la finalización de datos recibidos.
- **flnOkInt**: Señal que indica que el interleaver ha tomado el dato que hemos colocado en nuestra salida "output".

Pines de salida

- **output**: Dato que estamos mandando al interleaver.
- **outOk**: Bit que indica que está activo el envío de datos.
- **flnOk**: Indicación enviada al convolucionador para indicar que hemos tomado el dato.



INTERLEAVER



Este bloque barajará los datos escribiéndolos en una memoria por columnas y entregándolos al mapper leyendo por filas.

Tiene dos estados principales: 'recibiendo', en el que recibe datos del scrambler y los escribe en la memoria y 'pasando' en el que lee los datos de la memoria y los entrega al mapper. Es interesante comentar que en el estado pasando entrega 1, 2 o 3 bits por ciclo al mapper, dependiendo de la constelación. De esta manera, el mapper recibe los bits de una portadora cada ciclo, de modo que no requiere de una memoria a su salida.

Pines de entrada

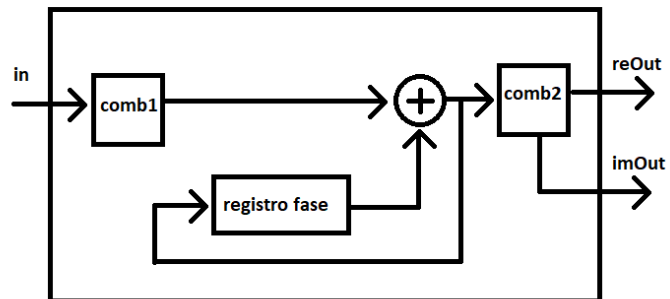
- **cons:** Señal de 3 bits que nos indica la constelación del sistema.
- **DOUT:** Es la señal de salida de la memoria.
- **inOK:** Señal que indica que el bit a la entrada procedente del Scrambler es correcto.
- **input:** Es por donde llega el flujo de bits procedente del scrambler.
- **mapOK:** Señal que indica que el mapper ha cogido los datos, y que tomar nuevos datos del bloque anterior.
- **finSerie:** Señal de conexión con el mapper, que le indica cuando el mapper y el puerto serie están libre, y así puede comenzar la carga de datos.

Pines de salida

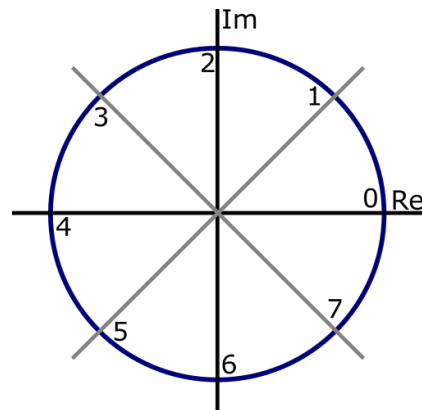
- **ADDR:** Señal de entrada de direcciones a la memoria.
- **DIN:** Señal de entrada de datos a la memoria.
- **output:** Dato en forma de vector de 3 bits que estamos mandando al mapper.
- **W:** Señal de activación de escritura en la memoria.
- **finOK:** Señal de feedback, que confirma al scrambler que hemos tomado el dato proporcionado.
- **outOK:** Bit que indica que está activo el envío de datos.

MAPPER

El mapper insertará las muestras complejas en la ifft. Para ello tendrá 4 estados principales: 'reposo', en el que esperará a que el interleaver baraje los bits de un símbolo y ponga una muestra válida en su salida. 'Ceros1', en el que inyectará a la ifft la primera ristra de ceros y la referencia. 'Datos', en el que inyectará las portadoras que contienen datos. Por último, 'Ceros2', en el que inyecta la segunda ristra de ceros y espera a que se envíe el símbolo por el puesto serie.



En el estado 'datos' el mapper se comportará como en las figuras adyacentes. 'Comb1' se corresponde con un circuito combinacional que transforma los bits provenientes del interleaver en el desfase que se debe aplicar a cada muestra con respecto a la anterior (que se conserva en registro fase). El registro fase se inicializará a 4 al comienzo de cada símbolo (ver la 2ª figura). 'comb2' se corresponde con un circuito combinacional que transformará las unidades de fase (2ª figura) en muestras complejas de 32 bits (16 para la parte real y 16 para la parte imaginaria).



Se entregará una muestra compleja en cada ciclo, para optimizar el uso de memorias.

En los estados 'ceros1' y 'ceros2' el comportamiento es mucho más sencillo, ya que sólo se inyectan ceros en la ifft (y la referencia en la última muestra de ceros1).

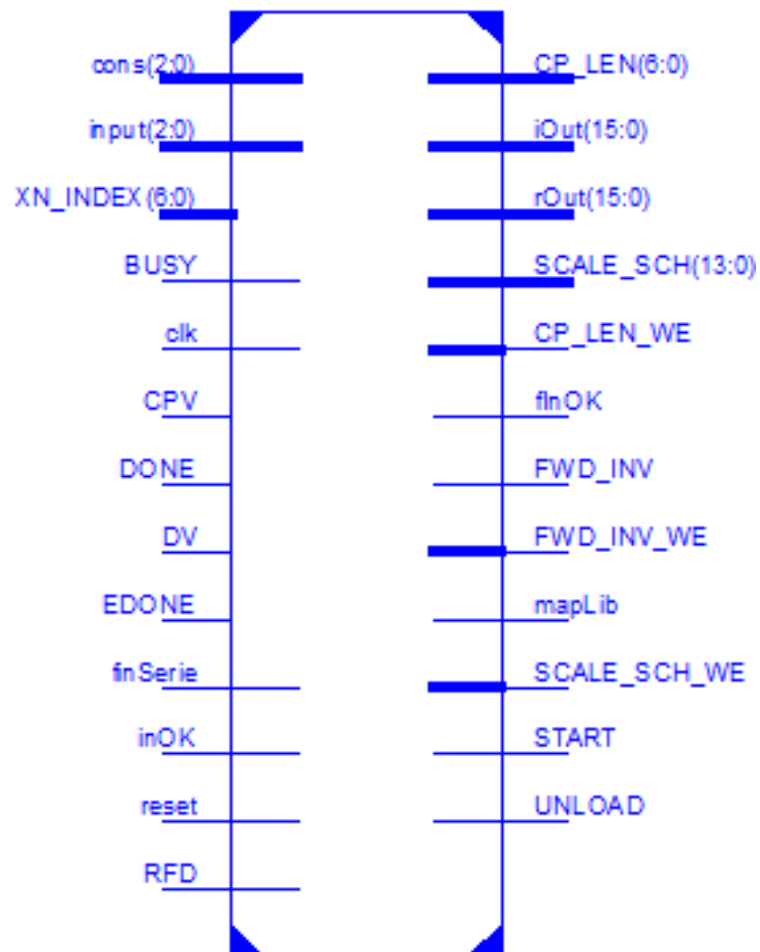
Pines de entrada (Las señales que no aparecen es porque no se usan en el sistema, excepto clk y reset)

- **cons:** Señal de 3 bits que nos indica la constelación del sistema.
- **input:** : Es por donde llega el flujo de bits procedente del interleaver.
- **BUSY:** Señal procedente de la ifft que indica que se están procesando los datos.
- **inOK:** Señal que indica que el bit a la entrada procedente del interleaver es correcto.
- **RFD:** Entrada procedente de la ifft que indica que está en espera de las muestras.

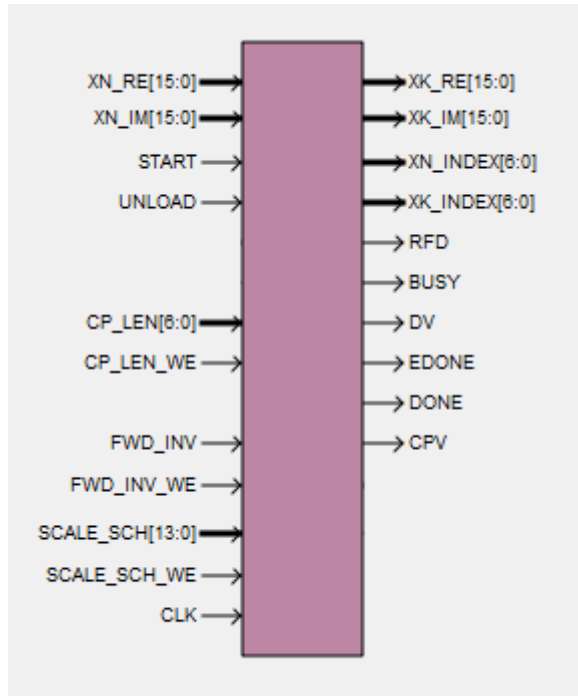
Pines de salida

- **CP_LEN:** Valor que indica la longitud del prefijo cíclico.
- **iOut:** Flujo de datos procedente de la parte imaginaria del proceso.
- **rOut:** Flujo de datos procedente de la parte real del proceso.
- **SCALE_SCH:** Valor de escala de la ifft.
- **CP_LEN_WE:** Señal que indica a la ifft el momento en el que tomamos dato de CP_LEN.
- **finOK:** Señal de feedback, que confirma al interleaver que hemos tomado el dato proporcionado

- **FWD_INV**: Señal que indica si el flujo de datos es natural o invertido.
- **FWD_INV_WE**: Señal que indica a la ifft el momento en el que tomamos dato de FWD_INV.
- **SCALE_SCH_WE**: Señal que indica a la ifft el momento en el que tomamos dato de SCALE_SCH.
- **START**: Indicación a la ifft del proceso de comienzo de carga de datos.
- **UNLOAD**: Indicación a la ifft del proceso de descarga de datos.
- **mapLib**: Señal de conexión al interleaver, que le indica cuando el mapper y el puerto serie están libre, y así puede comenzar la carga de datos.



IFFT



Configuración del Core:

Implementation: Radix2 lite. Porque esta opción necesita menos recursos.

Transform length: 128 muestras y 16 bits/muestra en cada parte real e imaginaria

Target Clock Frequency: 50MHz es la frecuencia para la IFFT, siendo ésta lo necesariamente rápida para poder cumplir con todas las especificaciones de la transmisión de símbolos. Marca el clk del sistema completo.

Data format: fixed point

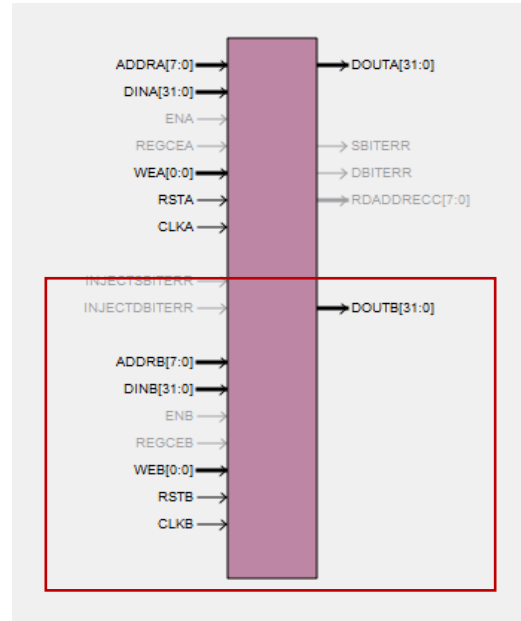
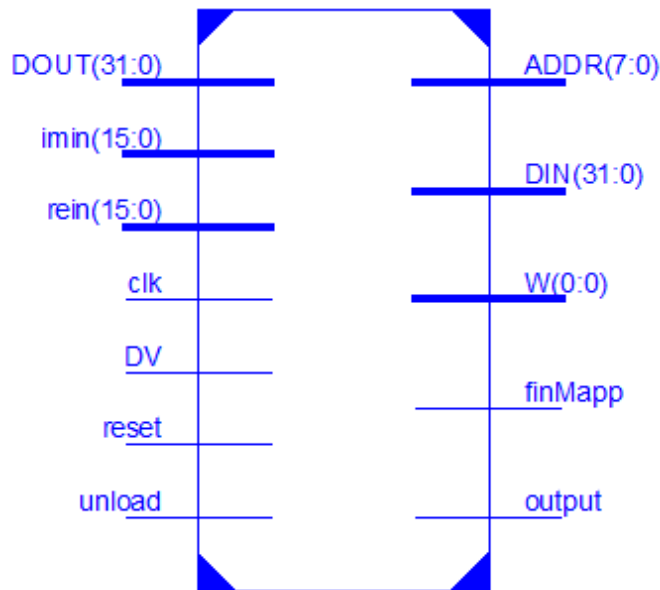
Scaling options: Scaled con lo que tenemos menos Recursos usados

Input data timing: sin offset

Output ordering: Natural order

Rounding modes: Truncation al ocupar menos recursos

PUERTO SERIE



Serializa la salida del shift (siguiente página), tras almacenar los datos en la memoria.

Pines de entrada

- **DOUT:** Salida de datos de la memoria.
- **imin:** Valor imaginario proveniente de la ifft.
- **rein:** Valor real proveniente de la ifft.
- **unload:** Señal proveniente del mapper que indica el comienzo del proceso de descarga de datos de la ifft, y carga de datos en la memoria
- **DV:** Proveniente de la ifft, que nos indica que los datos reales e imaginarios recibidos son válidos.

Pines de salida

- **ADDR:** Señal de entrada de direcciones a la memoria.
- **DIN:** Señal de entrada de datos a la memoria.
- **W:** Señal de activación de escritura en la memoria.
- **output:** Salida serializada global del sistema.
- **finMapp:** Señal que indica al mapper que el puerto serie está libre.

SHIFT

Este bloque se ha añadido al proyecto después de realizar la presentación, tras los comentarios recibidos en ella por parte de los profesores de la asignatura.

Como no hemos realizado un shift a la entrada de la ifft, hemos obtenido como resultado que una de cada dos muestras a la salida del sistema resultaba tener el signo contrario al esperado. Para solucionarlo hemos añadido un pequeño bloque combinacional que realiza el complemento A2 de las muestras con índice impar (tener en cuenta que la 1ª muestra es la N.º 0) a la salida de la ifft. De esta manera obtenemos la salida correcta.

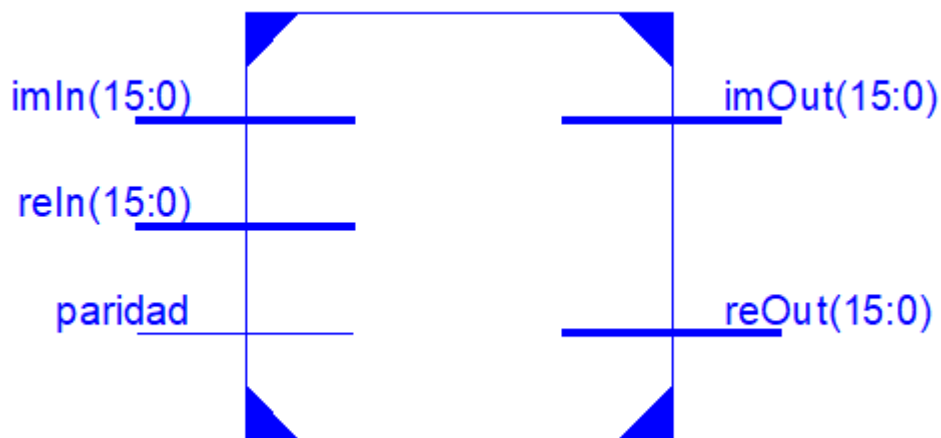
De este bloque no se incluye diagrama de bolas porque es puramente combinacional, por lo que no tiene estados.

Pines de entrada

- **imIn:** Entrada de las muestras de la parte real provenientes de la ifft.
- **reIn:** Entrada de las muestras de la parte real provenientes de la ifft.
- **paridad:** Bit menos significativo de la salida `xk_index` de la ifft, nos indicará si la muestra en cuestión es par o impar.

Pines de salida

- **imOut:** Salida de la parte imaginaria tras realizar el shift.
- **reOut:** Salida de la parte imaginaria tras realizar el shift.



COMENTARIOS SOBRE RESULTADOS

Tras la simulación del circuito, se ha volcado la salida del sistema en ficheros que se adjuntan en la entrega del trabajo realizado.

Si se comparan con los resultados del modelo realizado en Matlab, se obtiene que el máximo error cometido en las muestras es del orden de 10^{-4} (como se puede observar en la figura) siendo la media de un orden de magnitud menor.

```
Trial>> [ES,SS,IS,MS,FS,PS] = transmitir(input4,2);
Trial>> max(abs(PS.'-salidaQpskRe-li*salidaQpskIm))

ans =

    1.1856e-04

Trial>> [ES,SS,IS,MS,FS,PS] = transmitir(input4,3);
Trial>> max(abs(PS.'-salida8pskRe-li*salida8pskIm))

ans =

    1.2487e-04

Trial>> [ES,SS,IS,MS,FS,PS] = transmitir(input4,1);
Trial>> max(abs(PS.'-salidaBpskRe-li*salidaBpskIm))

ans =

    1.2548e-04
```

Los ficheros obtenidos tras la realización del trabajo se han cargado en un fichero .m, para poder compararlo fácilmente.

RECURSOS OCUPADOS DE LA FPGA

El sistema completo y la PPDU no cabrían en la FPGA a utilizar, como se puede observar en la siguiente tabla. Además del porcentaje de ocupación, se obtiene un warning indicando que los recursos requeridos para la ifft y las memorias no son compatibles.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	992	960	103%
Number of Slice Flip Flops	882	1920	45%
Number of 4 input LUTs	1230	1920	64%
Number of bonded IOBs	7	83	8%
Number of BRAMs	4	4	100%
Number of MULT18X18SIOs	2	4	50%
Number of GCLKs	1	24	4%

Sin embargo, el transmisor realizado si que cabe en la FPGA si la PPDU se implementase aparte, como se observa en la siguiente tabla. Los ficheros .bit y .ucf entregados se corresponde con la implementación únicamente del transmisor sin la PPDU.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	916	960	95%
Number of Slice Flip Flops	822	1920	42%
Number of 4 input LUTs	1097	1920	57%
Number of bonded IOBs	10	83	12%
Number of BRAMs	3	4	75%
Number of MULT18X18SIOs	2	4	50%
Number of GCLKs	1	24	4%

WARNINGS

Aparecen en los ip core en el proceso de síntesis:

En cada bloque ip (memoria del interleaver/puerto serie, memoria de la ppdu e ifft) nos alerta de la instanciación de una caja negra. No afectan al funcionamiento del sistema.

Aparecen en el mapper en el proceso de síntesis:

"Input <EDONE> is never used. This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved."

"Input <CPV> is never used. This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved."

"Input <DV> is never used. This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved."

"Input <DONE> is never used. This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved."

"Input <XN_INDEX> is never used. This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved."

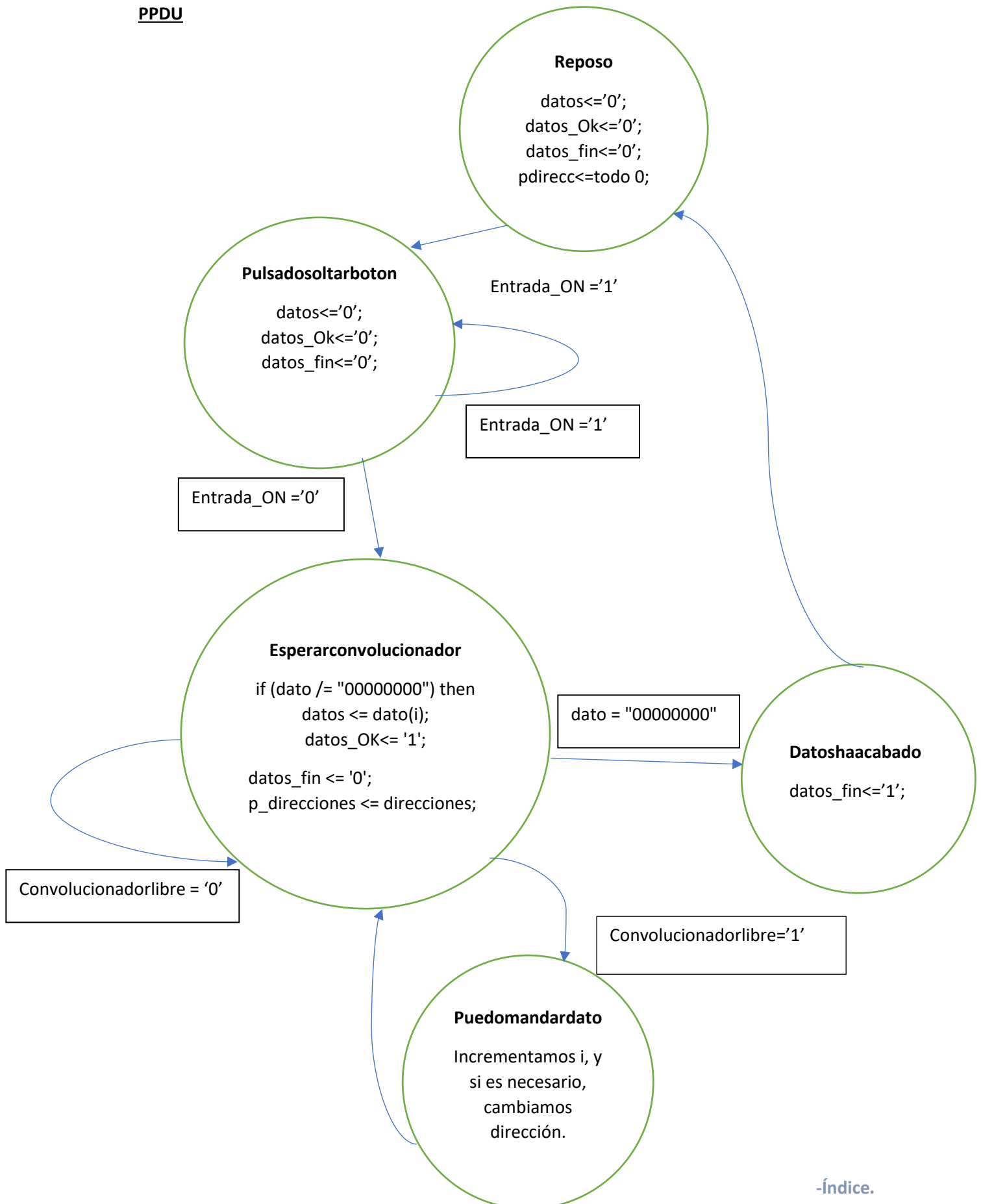
Al realizar el mapper, hicimos que toda la señalización de la ifft se dirigiese al mapper. Sin embargo, no requerimos todas las señales para el correcto funcionamiento de éste, por lo que permanecen sin utilizarse. Por ello nos sale un warning por cada señal no empleada.

Aparecen en la ifft en el proceso de implementación:

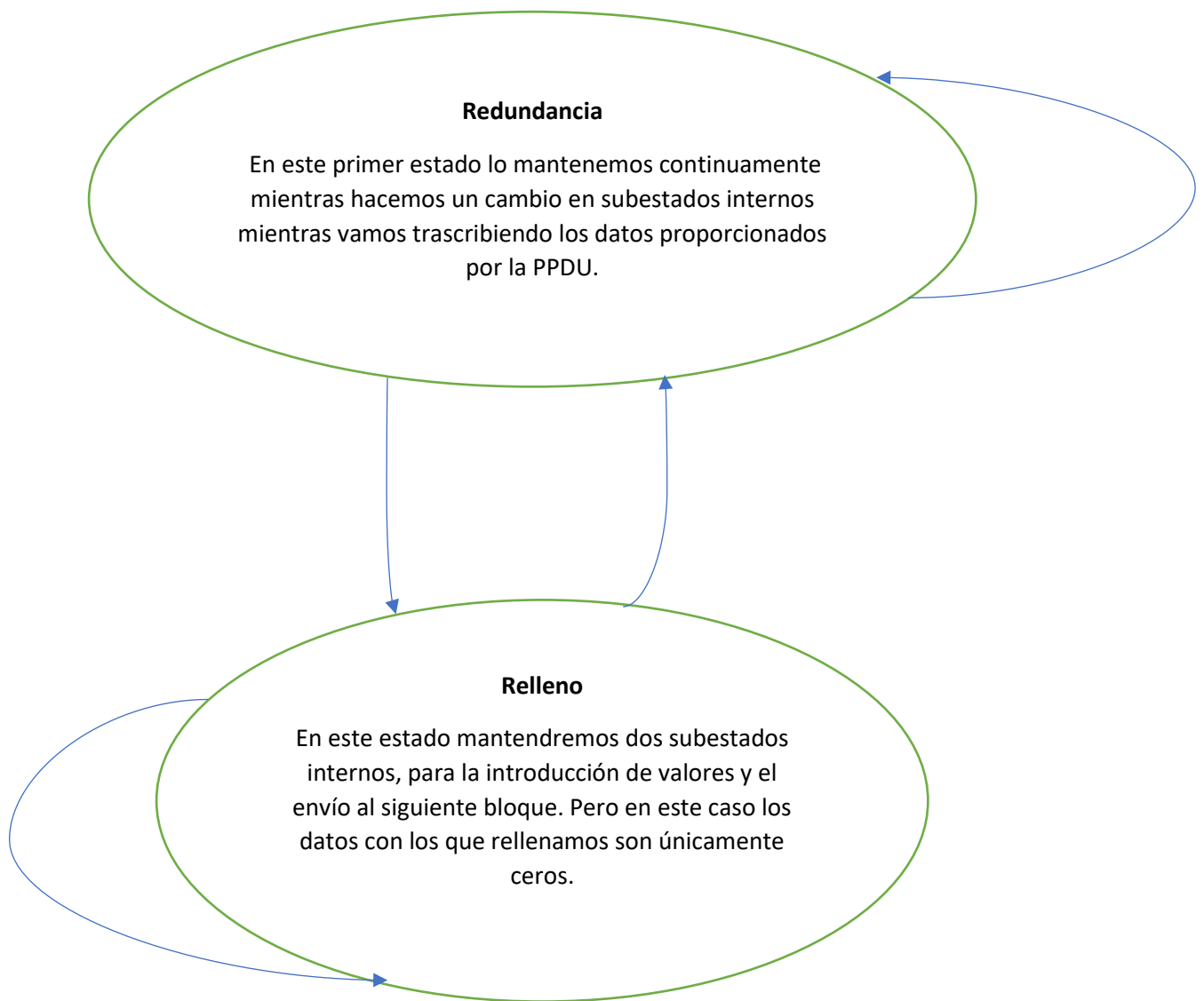
No incluimos una lista, porque son muy numerosos. Además se deben a la configuración empleada por la ifft, por lo que no afectan a nuestra implementación.

DIAGRAMAS DE BOLAS

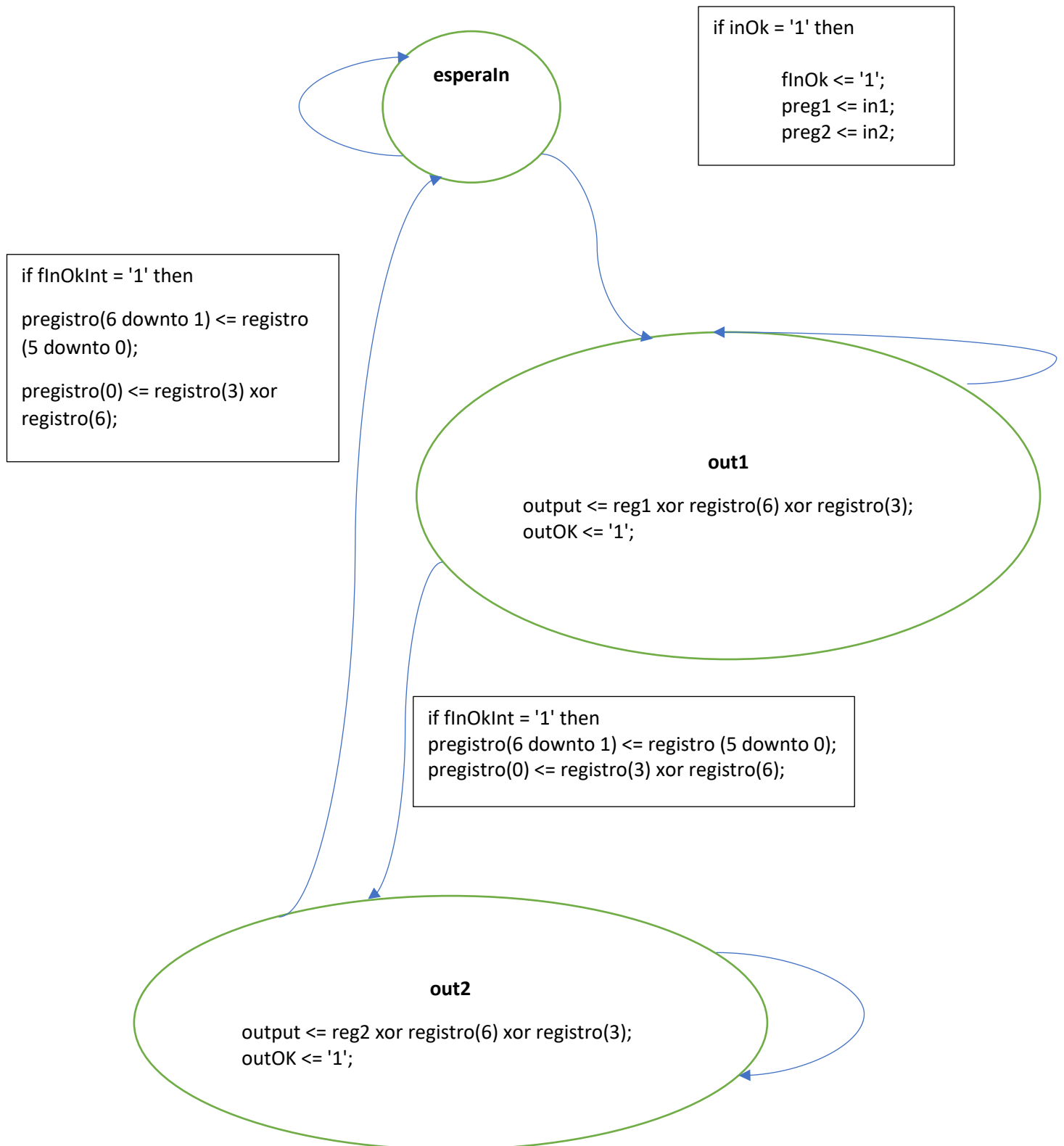
PPDU



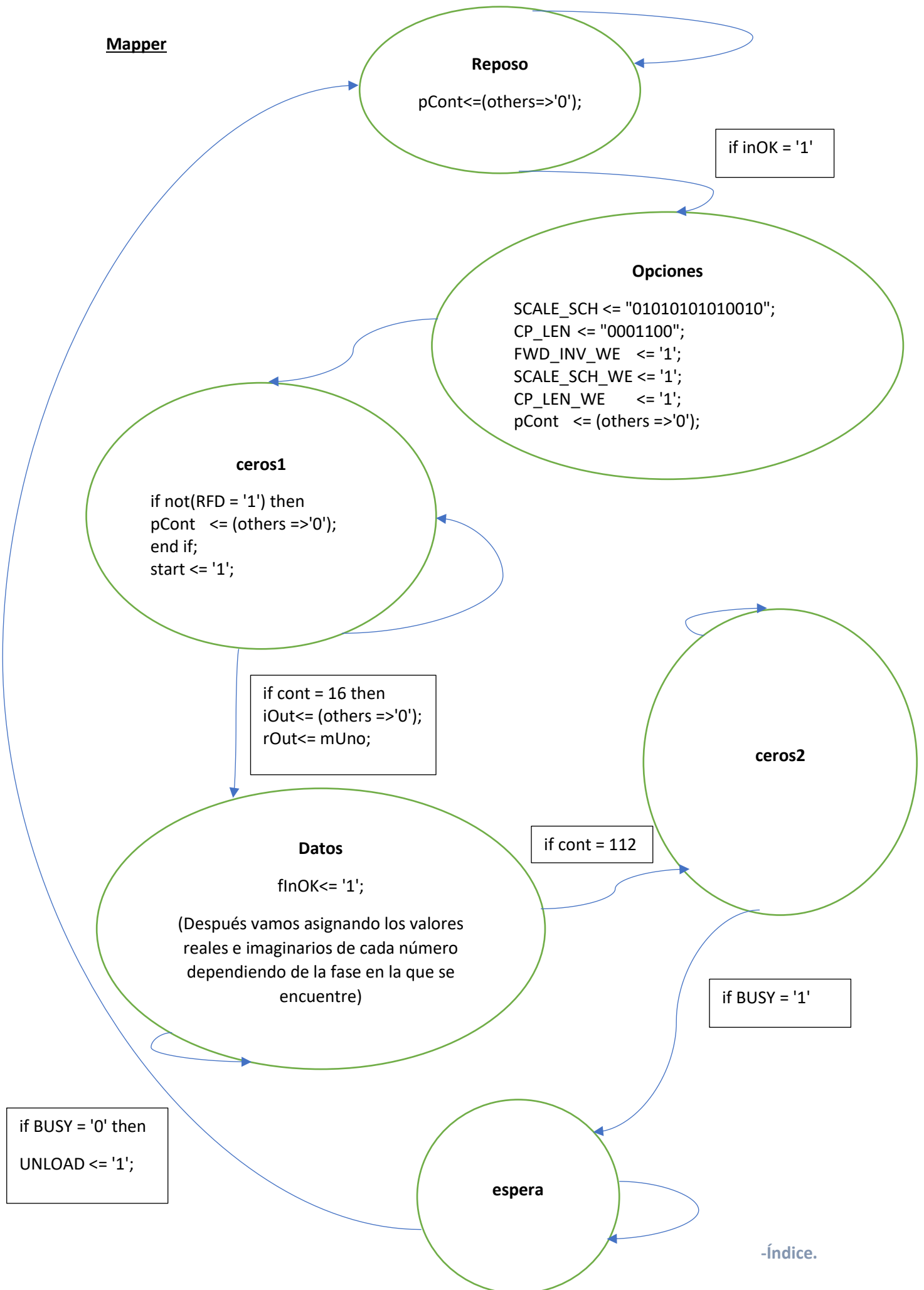
Convolucionador



Scrambler

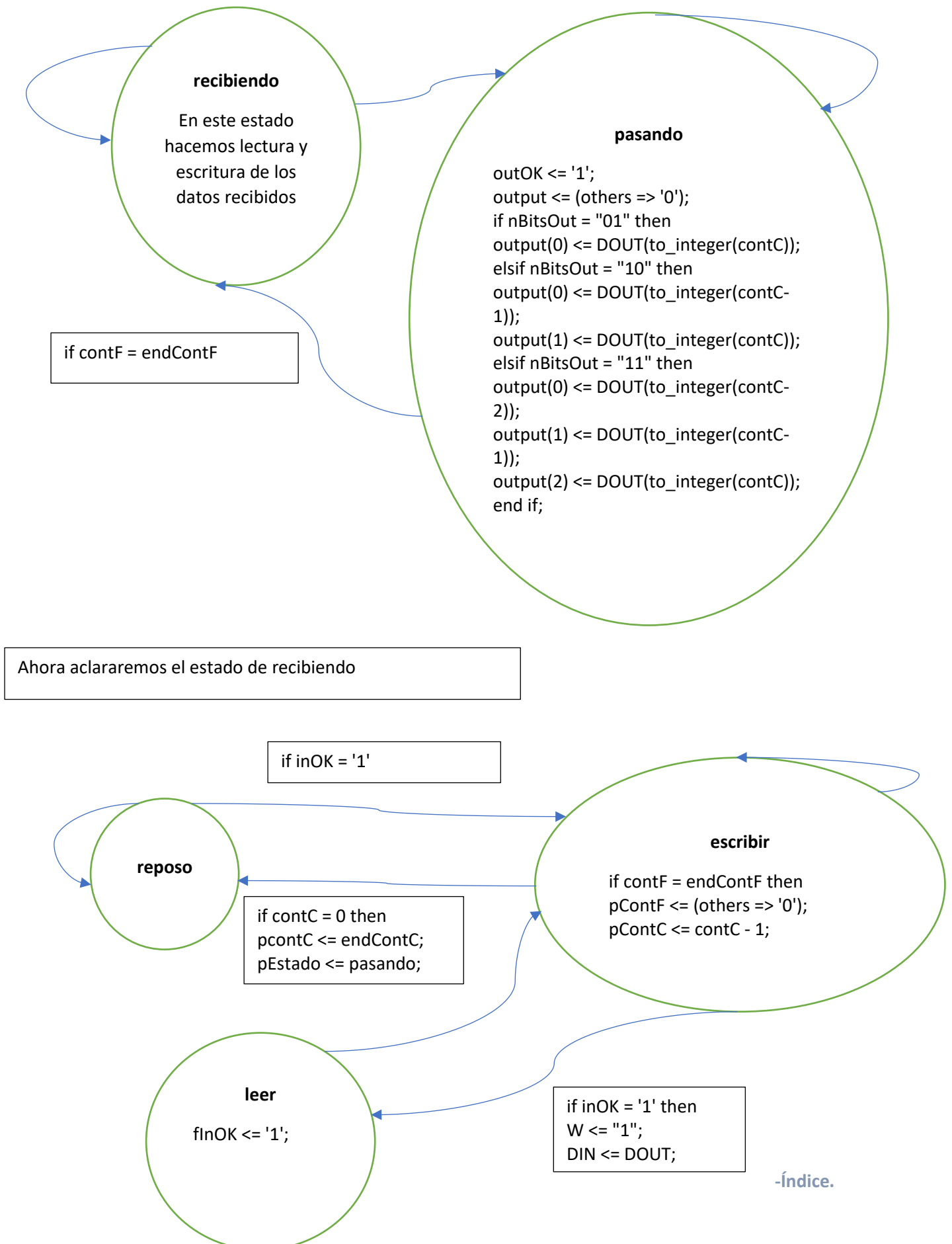


Mapper



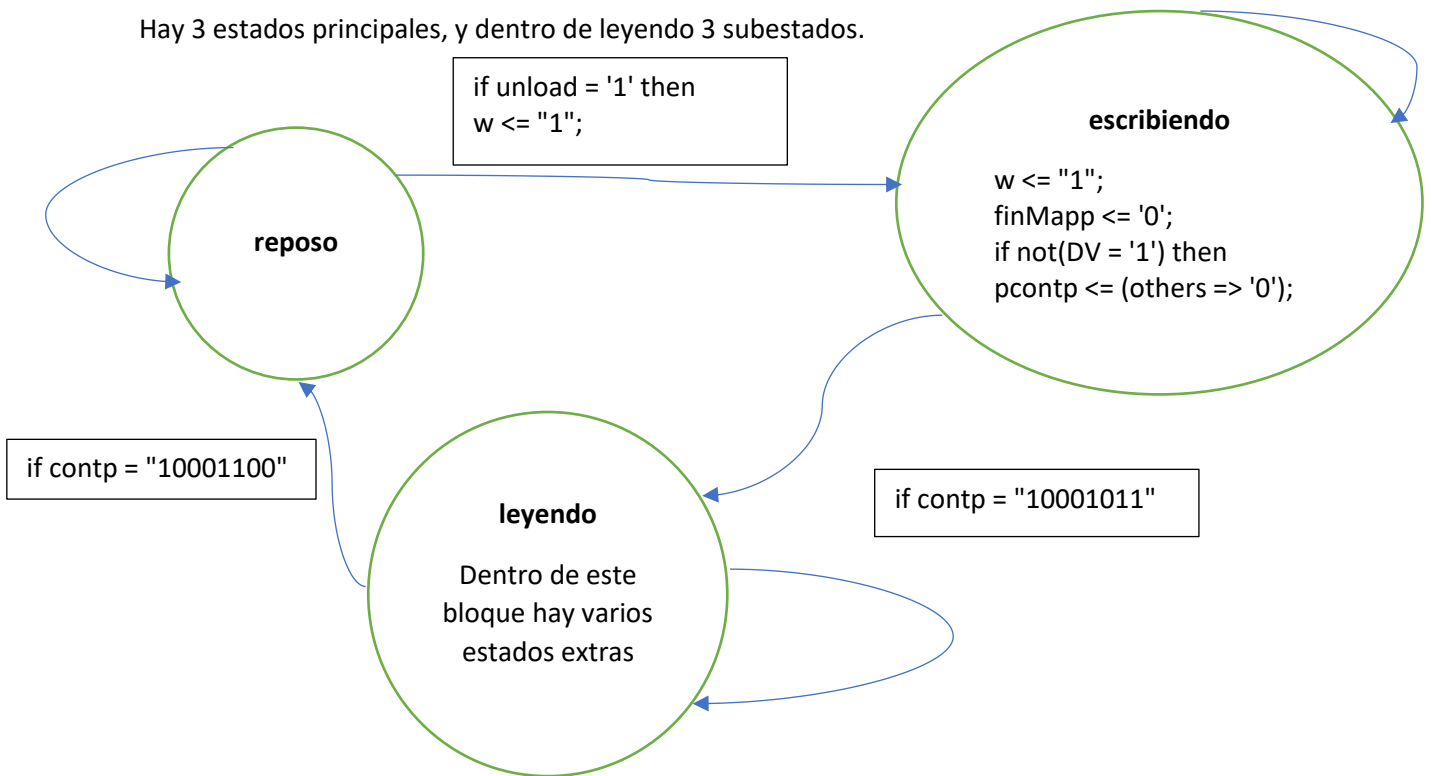
Interleaver

Hay dos estados principales, y dentro de recibiendo se está generando otros estados.



Puerto serie

Hay 3 estados principales, y dentro de leyendo 3 subestados.



Ahora aclararemos el estado de **leyendo**

