**Explicación del Diagrama General**

1. **One-hot counter**

**Será un contador One-Hot. Se le asignará un valor de 12’b0000\_0000\_0001 tras la escritura del FT. Posteriormente, tras cada ciclo de reloj, el 1 será desplazado hacia la izquierda.**

**Tras llegar la señal de ch\_end, se le reasignará el valor 12’b0000\_0000\_0001. Cada bit corresponderá con la señal de latch\_en de cada uno de los latch incluidos en el bloque.**

**En caso de la llegada de un rd\_end, su valor cambiará a 0, hasta la llegada del siguiente FT.**

1. **Register Block**

**Contará con 12 registros de 12 bits. Permitirá guardar hasta los 12 canales, para poder restarle al valor siguiente dicho valor de bloque anterior y realizar la codificación basada en el diferencial. El enable de cada uno de los registros se corresponde con un bit diferente del contador one-hot, por lo que solo uno estará activado a la vez**

1. **Comparador**

**Con un valor umbral superior y un valor umbral inferior, decidiremos si realizar una codificación huffman o no.**

**Se asume que el valor de umbral inferior será siempre negativo, mientras que el superior será siempre positivo.**

**El resultado de la comparación será despreciado si tenemos habilitado el wr\_ft**

1. **LUT Huffman**

**Consistirá básicamente en un bloque case con las asignaciones huffman correspondientes. Si no se encuentra dentro del rango de compresión da un valor por defecto.**

**Se añadirá un 1 extra que indicando que está codificado. Todos los códigos de huffman (con el 1 que indica la codificación incluido) se encuentran invertidos.**

1. **LUT N Bits**

**Al igual que el bloque anterior consiste en un case que indicaría el número de bits de cada codificación Huffman (teniendo en cuenta el bit extra que indica si se encuentra o no codificado).**

1. **Contador 5 bits**

**Lleva la cuenta de donde debemos colocar el nuevo valoren el shift register; mandando la señal de load cuando se encuentre lleno la parte baja de 16 bits.**

**Para poder ‘colocar’ los nuevos valores en el shift register, se utilizará un decodificador que nos devuelve un solo 1 en la posición indicada por este contador. De esta manera podremos multiplicar el dato introducido por N y sumarlo al propio shift register. (No se ha utilizado directamente un contador one-hot, puesto que como el desplazamiento es variable nos veríamos obligados a utilizar un barrel shifter.)**

1. **Shift Register**

**Con un bloque de 32 bits, para asegurar que nunca haya overfloat. La parte baja de 16 bits es la que se cargará a la interfaz de GbE, al igual que generará la señal de enable. Solo cargará los valores cuando las señales de wr\_en y rd\_en indiquen que son válidos.**

**Conexiones para integrar el módulo.**

**data\_out: Se corresponde con el flujo de datos de 16 bits.**

**wr\_ft: señal que indica la existencia de un FT en el data\_out, así como indica el comienzo de la transmisión de un evento.**

**wr\_en: enmascara data\_en, eligiendo solo aquellos valores que queremos transmitir.**

**data\_en: indica que el dato que se encuentra en data\_out es válido.**

**ch\_end: señal que indica el final de la retrasmisión de un ciclo de canales (hasta 12).**

**rd\_end: señal que indica la finalización de un evento completo.**

**on: señal principal, debe estar activa para que el módulo se encuentre dispuesto a esperar un FT y comenzar la transmisión. Si tras la llegada de un rd\_end, on se encuentra activo, automáticamente comenzará la espera de un FT.**

**DFdata\_wren: Señal que indica la validez de los datos en DFdata\_out para ser enviados**

**DFdata\_out: datos de salida codificados y concatenados de 16 bits con MSB DFdata\_out [15] y LSB [0] .**