G-DSP Engine

Implementación del Subsistema de Transmisión

Fase 1 — Documentación Técnica del TFG

G-DSP Team

Febrero 2026

Índice

1. Introducción al Subsistema de Transmisión

El subsistema de transmisión (TX) del procesador banda base G-DSP Engine constituye la primera etapa funcional del módem 16-QAM implementado sobre la FPGA Gowin GW1NR-LV9QN88PC6/I5 (placa de desarrollo Sipeed Tang Nano 9K). Su función es generar una señal banda base conformada espectralmente, lista para su transmisión por el canal o, en el ámbito de este proyecto, para su procesamiento por el subsistema de recepción y su visualización en tiempo real mediante HDMI.

La cadena de transmisión implementada sigue el diagrama de bloques clásico de un transmisor QAM digital:

$$\text{PRBS-23} \xrightarrow{4\text{ bits}} \text{Mapper 16-QAM} \xrightarrow{I/Q} \uparrow_{\times 4} \rightarrow \text{FIR RRC} \rightarrow \text{TX}_{I,Q} \tag{1}$$

Los módulos RTL que materializan esta cadena, implementados en SystemVerilog, son los siguientes:

1. `bit_gen` — Generador de bits pseudo-aleatorios (PRBS-23).
2. `qam16_mapper` — Mapeador 16-QAM con codificación Gray.
3. `rrc_filter` — Filtro FIR Root-Raised Cosine (33 taps).
4. `tx_top` — Integración del subsistema completo, incluyendo el sobremuestreo por inserción de ceros.

Todos los módulos comparten parámetros, tipos y constantes definidos en el paquete centralizado `gdsp_pkg`, lo que garantiza la coherencia de la aritmética de punto fijo a lo largo de toda la jerarquía de diseño.

La plataforma objetivo impone restricciones estrictas de recursos (8640 LUT4, 20 bloques DSP MULT9, 27 MHz de reloj del sistema) que han condicionado cada decisión arquitectónica, como se detalla en las secciones siguientes.

2. Aritmética de Punto Fijo: Formato Q1.11

2.1. Justificación del formato de 12 bits firmados

La elección de la representación numérica es una decisión crítica en todo diseño DSP embebido, ya que afecta directamente a la precisión de la señal, al consumo de recursos y al rendimiento del camino crítico. En el proyecto G-DSP Engine se adopta el formato **Q1.11** en complemento a dos:

- **1 bit de signo** (implícito como MSB del complemento a dos).
- **0 bits enteros** (excluyendo el signo).
- **11 bits fraccionarios**.
- **Anchura total**: $N = 1 + 0 + 11 = 12$ bits.

Se emplea la convención de Texas Instruments, donde la notación $Q_{m.n}$ indica m bits enteros (incluyendo el bit de signo) y n bits fraccionarios, con una anchura total de $m + n$ bits. Así, Q1.11 representa:

$$x = -b_{11} \cdot 2^0 + \sum_{k=0}^{10} b_k \cdot 2^{k-11} \tag{2}$$

donde b_{11} es el MSB (bit de signo) y b_0 es el LSB.
El rango representable es:

$$x \in \left[-1{,}0\ ;\ +1{,}0 - 2^{-11}\right] = [-1{,}0\ ;\ +0{,}99951] \tag{3}$$

con una resolución (paso de cuantificación) de:

$$\Delta = 2^{-11} = \frac{1}{2048} \approx 4{,}88 \times 10^{-4} \tag{4}$$

Implicaciones para la modulación. Los niveles de la constelación 16-QAM normalizados a potencia unitaria tienen valor máximo $|3/\sqrt{10}| \approx 0{,}9487$, que cae holgadamente dentro del rango $[-1, +0{,}9995]$, dejando un margen del 5,4 % para acomodar el *overshoot* típico de la conformación espectral RRC (que depende del factor de roll-off α). Para $\alpha = 0{,}25$, el *overshoot* pico-a-pico del filtro RRC es inferior al 4 % del nivel de pico de la señal, por lo que el formato Q1.11 resulta suficiente sin riesgo de saturación.

Adaptación a los recursos DSP. Los bloques MULT9 del GW1NR-9 operan nativamente en modo 9×9 bits firmados o pueden acoplarse en pares para formar multiplicadores de 18×18 bits (modo MULT18). Una muestra de 12 bits y un coeficiente de 12 bits requieren como mínimo el modo MULT18, lo que resulta óptimo: anchuras menores desperdiciarían capacidad DSP y anchuras mayores (por ejemplo, 16 bits) requerirían acoplar cuatro MULT9, duplicando el consumo de slices DSP.

2.2. Relación señal a ruido de cuantificación (SQNR)

Para un cuantificador uniforme de B bits fraccionarios con señal de entrada sinusoidal a fondo de escala, la relación señal a ruido de cuantificación viene dada por la fórmula clásica:

$$\mathrm{SQNR} \approx 6{,}02 \cdot B + 1{,}76 \text{ dB} \tag{5}$$

Sustituyendo $B = 11$ (bits fraccionarios del formato Q1.11):

$$\mathrm{SQNR} \approx 6{,}02 \times 11 + 1{,}76 = 66{,}22 + 1{,}76 = \boxed{67{,}98 \text{ dB}} \tag{6}$$

Este valor de $\sim$68 dB es ampliamente superior al requisito típico de un sistema 16-QAM, que opera con $E_b/N_0 \sim 10$ dB en el umbral de BER objetivo (10^{-5}). El ruido de cuantificación queda, por tanto, más de 55 dB por debajo del ruido térmico del canal, siendo despreciable a efectos prácticos.

2.3. Crecimiento de bits en la cadena de procesado

En toda cadena DSP, las operaciones aritméticas incrementan progresivamente la anchura de palabra. La Tabla 1 resume la evolución a lo largo del subsistema TX.

Tabla 1: Crecimiento de bits a lo largo de la cadena TX.

Etapa	**Formato**	**Bits**	**Justificación**
Muestra de entrada	Q1.11	12	Señal I/Q original
Coeficiente FIR	Q1.11	12	Misma anchura que datos
Producto (dato $\times$ coef.)	Q2.22	24	$12 + 12 = 24$
Acumulador (30 bits)	Q8.22	30	$24 + \lceil \log_2(33) \rceil = 30$
Salida truncada	Q1.11	12	Redondeo convergente

Los 6 bits adicionales del acumulador ($\lceil \log_2(33) \rceil = 6$) previenen cualquier desbordamiento durante la acumulación de hasta 33 términos, garantizando que la precisión numérica del filtro esté limitada únicamente por la truncación final, y no por errores de desbordamiento intermedios.

3. Generación de Bits: PRBS-23 (LFSR)

3.1. Polinomio generador según ITU-T O.151

Para la generación de la secuencia de datos de prueba se emplea un *Linear Feedback Shift Register* (LFSR) de longitud máxima, conforme a la recomendación ITU-T O.151. El polinomio característico es:

$$p(x) = x^{23} + x^{18} + 1 \tag{7}$$

Este polinomio es primitivo sobre GF(2), lo que garantiza que la secuencia generada tiene período máximo:

$$L = 2^{23} - 1 = 8\,388\,607 \text{ bits} \tag{8}$$

Para la modulación 16-QAM, con 4 bits por símbolo, esto equivale a $L/4 = 2\,097\,151$ símbolos antes de que la secuencia se repita. A una tasa de símbolo de $f_{\text{sym}} = 27/4 = 6{,}75\,\text{MHz}$, el período de repetición es:

$$T_{\text{rep}} = \frac{2^{23} - 1}{4 \cdot f_{\text{sym}}} = \frac{8\,388\,607}{4 \times 6{,}75 \times 10^6} \approx 0{,}311\,\text{s} \tag{9}$$

Este período es suficientemente largo para que las propiedades estadísticas de la secuencia sean indistinguibles de un proceso verdaderamente aleatorio en el contexto de las pruebas de verificación.

La elección de una secuencia PRBS normalizada (frente a un generador aleatorio arbitrario) se justifica por dos razones:

1. **Repetibilidad**: permite la verificación *bit-true* contra el modelo de referencia Python (*Golden Model*).
2. **Propiedades espectrales**: la PRBS-23 presenta un espectro plano en la banda de interés, lo que la convierte en una excitación adecuada para la medida de respuesta del sistema.

3.2. Desenrollado del LFSR para entrega paralela

El mapeador 16-QAM requiere 4 bits por símbolo en cada ciclo de reloj en el que se genera un nuevo símbolo (cada SPS = 4 ciclos de reloj del sistema). Un LFSR convencional produce un único bit por ciclo, lo que obligaría a acumular bits durante 4 ciclos consecutivos e introduciría latencia y complejidad de control innecesarias.

La solución adoptada es el **desenrollado combinacional** (*loop unrolling*) del LFSR: se calcula la función de transición de estado 4 veces en cascada dentro de un mismo ciclo de reloj, obteniendo los 4 bits de salida simultáneamente. La función de avance unitario se define como:

$$\text{fb} = s[22] \oplus s[17], \qquad s' = \{s[21:0],\ \text{fb}\} \tag{10}$$

donde $s[k]$ denota el bit k-ésimo del estado del LFSR (indexación desde cero) y $\oplus$ es la operación XOR. El desenrollado produce una cadena de estados intermedios:

$$s^{(0)} = s_{\text{actual}}, \quad s^{(i+1)} = f_{\text{step}}(s^{(i)}), \quad i \in \{0, 1, 2, 3\} \tag{11}$$

El bit de salida de cada paso es el MSB del estado intermedio $s^{(i)}[22]$, y los 4 bits se recogen en paralelo:

$$\texttt{bits\_out}[3-i] = s^{(i)}[22], \qquad i \in \{0,1,2,3\} \tag{12}$$

El nuevo estado del LFSR se actualiza al estado final $s^{(4)}$ en el flanco de reloj, y todo el cálculo combinacional intermedio se resuelve dentro de un único período de reloj. Dado que cada paso del desenrollado implica únicamente una puerta XOR y un desplazamiento, el camino crítico total del desenrollado de 4 etapas es de apenas 4 niveles de XOR, lo que resulta trivial para una frecuencia de operación de 27 MHz ($T_{\text{clk}} \approx 37$ ns).

La semilla inicial se fija en el valor `0x7FFFFF` (*all-ones*), evitando el estado de bloqueo $s = 0$ que es inherente a todo LFSR estándar. La carga de la semilla se efectúa mediante la señal de reset asíncrono `rst_n`.

4. Mapeador 16-QAM con Codificación Gray

4.1. Codificación Gray por eje

La modulación 16-QAM asigna bloques de $\log_2(16) = 4$ bits a puntos de una constelación bidimensional. El mapeador implementado separa los 4 bits en dos pares independientes: los bits $[3:2]$ controlan el eje en fase (I) y los bits $[1:0]$ controlan el eje en cuadratura (Q).

Para cada eje se aplica una **codificación Gray**, en la que símbolos adyacentes en la constelación difieren en exactamente un bit. Esto minimiza la probabilidad de error de bit (BER) para una probabilidad de error de símbolo (SER) dada, ya que la mayoría de errores de detección se producen entre símbolos vecinos.

La correspondencia entre los 2 bits de cada eje y el nivel de la constelación se recoge en la Tabla 2.

Tabla 2: Tabla de correspondencia Gray para un eje de la constelación 16-QAM.

Bits	Nivel	Valor normalizado	Q1.11 (decimal)
`00`	-3	$-3/\sqrt{10} \approx -0{,}9487$	-1943
`01`	-1	$-1/\sqrt{10} \approx -0{,}3162$	-648
`11`	$+1$	$+1/\sqrt{10} \approx +0{,}3162$	$+648$
`10`	$+3$	$+3/\sqrt{10} \approx +0{,}9487$	$+1943$

Nótese que la transición entre niveles adyacentes ($-3 \to -1$, $-1 \to +1$, $+1 \to +3$) cambia exactamente un bit en cada caso: `00` $\to$ `01` $\to$ `11` $\to$ `10`, como exige el código Gray.

4.2. Normalización de potencia unitaria

En un sistema de comunicaciones digital, es práctica estándar normalizar la constelación para que su potencia media sea la unidad. Para una constelación 16-QAM rectangular con niveles $\{-3,-1,+1,+3\}$ equip-robables en cada eje, la potencia media por eje es:

$$P_{\text{eje}} = \frac{1}{4}\left((-3)^2 + (-1)^2 + (+1)^2 + (+3)^2\right) = \frac{9+1+1+9}{4} = 5 \tag{13}$$

La potencia total de la constelación (suma de ambos ejes) es:

$$P_{\text{total}} = 2 \times P_{\text{eje}} = 10 \tag{14}$$

Para obtener potencia media unitaria ($E[|s|^2] = 1$), los niveles se dividen por $\sqrt{P_{\text{total}}} = \sqrt{10}$:

$$d_k = \frac{a_k}{\sqrt{10}}, \qquad a_k \in \{-3, -1, +1, +3\} \tag{15}$$

Los valores normalizados y su representación en punto fijo Q1.11 (valor entero = round($d_k \times 2^{11}$)) se resumen en la Tabla 3.

Tabla 3: Niveles 16-QAM normalizados y su cuantificación en Q1.11.

Nivel a_k	Flotante d_k	Ideal $d_k \times 2^{11}$	Entero Q1.11	Hex (12 bits)
-3	$-0{,}948683$	$-1942{,}90$	-1943	`0x869`
-1	$-0{,}316228$	$-647{,}63$	-648	`0xD78`
$+1$	$+0{,}316228$	$+647{,}63$	$+648$	`0x288`
$+3$	$+0{,}948683$	$+1942{,}90$	$+1943$	`0x797`

El error de cuantificación máximo es:

$$\epsilon_{\text{máx}} = \frac{|1943 - 1942{,}90|}{2^{11}} = \frac{0{,}10}{2048} \approx 4{,}88 \times 10^{-5} \tag{16}$$

lo que supone un error relativo del 0,005 % respecto al nivel máximo de la constelación, completamente despreciable.

La implementación RTL utiliza una función combinacional `gray_lut()` que recibe los 2 bits de un eje y devuelve la constante Q1.11 correspondiente, definida como parámetro en el paquete `gdsp_pkg`:

```
function automatic sample_t gray_lut(input logic [1:0] bits);
    case (bits)
        2'b00:   gray_lut = QAM_NEG3;   // -1943
        2'b01:   gray_lut = QAM_NEG1;   //  -648
        2'b11:   gray_lut = QAM_POS1;   //  +648
        2'b10:   gray_lut = QAM_POS3;   // +1943
        default: gray_lut = '0;
    endcase
endfunction
```

Listing 1: LUT Gray para un eje de la constelación.

La salida del mapeador se registra (1 ciclo de latencia) para aislar el *fanout* de la LUT del camino crítico del filtro RRC aguas abajo.

5. Filtro Root-Raised Cosine (RRC)

5.1. Parámetros del filtro

El filtro de conformación espectral utilizado es un *Root-Raised Cosine* (RRC), cuya respuesta al impulso en tiempo continuo viene dada por:

$$h_{\text{RRC}}(t) = \begin{cases} \dfrac{1}{T_s}\left(1 + \alpha\left(\dfrac{4}{\pi} - 1\right)\right), & t = 0 \\[2ex] \dfrac{\alpha}{T_s\sqrt{2}}\left[\left(1 + \dfrac{2}{\pi}\right)\sin\!\left(\dfrac{\pi}{4\alpha}\right) + \left(1 - \dfrac{2}{\pi}\right)\cos\!\left(\dfrac{\pi}{4\alpha}\right)\right], & t = \pm\dfrac{T_s}{4\alpha} \\[2ex] \dfrac{1}{T_s}\,\dfrac{\sin\!\left[\pi\frac{t}{T_s}(1-\alpha)\right] + 4\alpha\frac{t}{T_s}\cos\!\left[\pi\frac{t}{T_s}(1+\alpha)\right]}{\pi\frac{t}{T_s}\left[1 - \left(4\alpha\frac{t}{T_s}\right)^2\right]}, & \text{resto} \end{cases} \tag{17}$$

Los parámetros de diseño seleccionados son:

Tabla 4: Parámetros de diseño del filtro RRC.

Parámetro	**Valor**	**Justificación**
Número de taps	33	Impar (simetría), compromiso precisión/recursos
Factor de roll-off α	0,25	Estándar DVB-S2, exceso de ancho de banda moderado
Muestras por símbolo (SPS)	4	Cumple Nyquist ($f_s > 2f_{\text{sym}}$)
Coef. pico (tap central)	+922	0,450195 en Q1.11 ($< 0{,}5$, sin saturación)

La simetría de fase lineal del filtro ($h[n] = h[N-1-n]$) se verifica examinando los coeficientes generados:

$$h[0] = h[32] = +18, \quad h[1] = h[31] = +9, \quad \ldots \quad h[16] = +922 \text{ (tap central)} \tag{18}$$

Esta simetría es una propiedad fundamental que garantiza un retardo de grupo constante de $(N-1)/2 = 16$ muestras para todas las frecuencias dentro de la banda de paso.

5.2. Decisión de arquitectura: forma transpuesta

Para la implementación hardware del filtro FIR se han evaluado tres arquitecturas clásicas:

1. Forma directa. En la forma directa, las muestras de entrada se almacenan en una línea de retardo y se multiplican por los coeficientes. Los productos se acumulan mediante una cadena de sumadores. El problema principal es que el **camino crítico** crece linealmente con el número de taps, ya que la acumulación es secuencial:

$$t_{\text{crítico}}^{\text{directa}} = t_{\text{mult}} + (N-1) \cdot t_{\text{add}} \tag{19}$$

Para $N = 33$, esto resulta en 32 etapas de sumador en serie, haciendo muy difícil cumplir el requisito de *timing* a 27 MHz.

2. Arquitectura plegada (*folded*). La alternativa plegada reutiliza un único multiplicador en múltiples ciclos de reloj. Para 33 taps se necesitarían 33 multiplicaciones por muestra, lo que exige un reloj interno de:

$$f_{\text{interno}} = N \times f_s = 33 \times 27\,\text{MHz} = 891\,\text{MHz} \tag{20}$$

Esta frecuencia es inalcanzable para la familia GW1NR-9 (cuyo límite práctico se sitúa en torno a 200 MHz para lógica general), por lo que se descarta esta opción.

3. Forma transpuesta (selección final). En la forma transpuesta, la estructura se invierte: cada etapa consiste en un multiplicador seguido de un sumador y un registro. El camino crítico se reduce a **una única multiplicación más una suma**:

$$t_{\text{crítico}}^{\text{transpuesta}} = t_{\text{mult}} + t_{\text{add}} \tag{21}$$

independientemente del número de taps. Esta característica la convierte en la arquitectura idónea para implementaciones en FPGA con restricciones de frecuencia.

Además, la forma transpuesta se adapta naturalmente a los bloques DSP de Gowin: cada etapa mapea directamente sobre la primitiva `pREG + MULT9`, donde el registro post-multiplicador (`pREG`) absorbe la registración de la etapa del *pipeline* sin consumir LUTs adicionales.

La ecuación de recurrencia de la forma transpuesta para un filtro de N taps es:

$$\begin{cases} p_k[n] = x[n] \cdot h[k], & 0 \le k \le N-1 \\ r_{N-1}[n] = p_{N-1}[n] & \text{(última etapa)} \\ r_k[n] = p_k[n] + r_{k+1}[n-1], & 0 \le k \le N-2 \\ y[n] = r_0[n-1] & \text{(salida)} \end{cases} \tag{22}$$

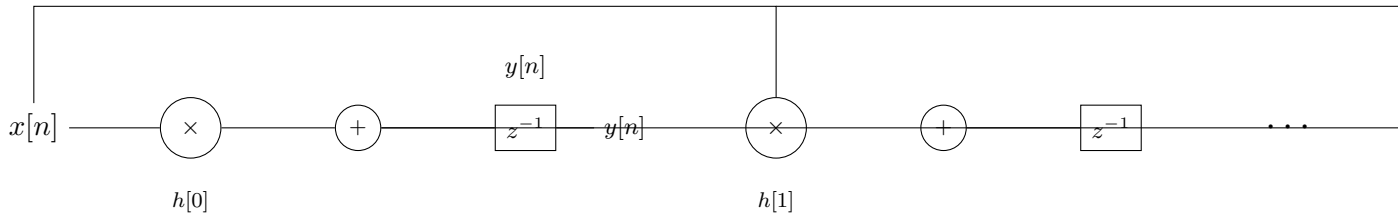

Figura 1: Arquitectura FIR en forma transpuesta. El camino crítico (línea roja conceptual) es una única multiplicación más una suma, independientemente de N.

Presupuesto de recursos DSP. La FPGA GW1NR-9 dispone de 20 bloques MULT9, que operan como 10 bloques MULT18 en modo acoplado. La multiplicación de 12 bits firmados requiere MULT18, por lo que se dispone de un máximo de 10 multiplicadores hardware. Con 33 taps por canal y dos canales (I/Q), se necesitarían 66 multiplicadores. El sintetizador de Gowin asigna las 10 unidades MULT18 disponibles a las etapas con mayor contribución (típicamente las taps centrales) y sintetiza las restantes como multiplicadores en lógica *fabric* (LUT). A 27 MHz, los multiplicadores basados en LUT cumplen holgadamente el *timing*.

5.3. Análisis de precisión: redondeo convergente

La operación más delicada del filtro RRC es la **truncación** del acumulador de 30 bits al formato de salida Q1.11 (12 bits). Una truncación simple (descartar los bits menos significativos) introduce un sesgo sistemático negativo (*bias* de DC) de valor medio $-\Delta/2 = -2^{-12}$, que se acumula a lo largo de la cadena de procesado y puede degradar el rendimiento del receptor.

Para eliminar este sesgo se emplea **redondeo convergente** (*Banker's rounding*, IEEE 754), que redondea al valor par más cercano cuando la fracción descartada es exactamente 0,5:

$$y = \lfloor x \rfloor + \begin{cases} 1, & \text{si } r > 0{,}5 \\ 1, & \text{si } r = 0{,}5 \text{ y } \lfloor x \rfloor \text{ es impar} \\ 0, & \text{en otro caso} \end{cases} \tag{23}$$

donde r es la parte fraccionaria descartada. En la implementación hardware, esta condición se codifica de forma eficiente mediante tres señales derivadas del acumulador `acc_val`:

Tabla 5: Señales para el redondeo convergente.

Señal	Bits del acumulador	Significado
`round_bit`	`acc_val[10]`	MSB de la parte descartada
`sticky`	`|acc_val[9:0]`	OR de los bits restantes
`lsb_result`	`acc_val[11]`	LSB del resultado truncado

La decisión de redondeo al alza se codifica como:

$$\texttt{round\_up} = \texttt{round\_bit} \wedge (\texttt{sticky} \vee \texttt{lsb\_result}) \tag{24}$$

Esta expresión garantiza que:

- Si la fracción descartada es $> 0{,}5$ (`round_bit` $= 1$ y `sticky` $= 1$), se redondea al alza.
- Si la fracción es exactamente $0{,}5$ (`round_bit` $= 1$ y `sticky` $= 0$), se redondea al alza solo si el LSB del resultado es 1 (número impar), forzando el resultado a ser par.
- Si la fracción es $< 0{,}5$ (`round_bit` $= 0$), se trunca (redondeo a la baja).

La salida truncada y redondeada se obtiene como:

$$\texttt{dout\_trunc} = \texttt{acc\_val}[22:11] + \texttt{round\_up} \tag{25}$$

donde los bits $[22:11]$ representan los 12 bits Q1.11 del producto acumulado Q2.22. Adicionalmente, se incorpora un circuito de **saturación** que protege contra desbordamientos inesperados, limitando la salida a los extremos representables:

$$y = \begin{cases} +2047 & \text{si desbordamiento positivo (SAT\_POS)} \\ -2048 & \text{si desbordamiento negativo (SAT\_NEG)} \\ \texttt{dout\_trunc} & \text{en caso normal} \end{cases} \tag{26}$$

Con coeficientes correctamente normalizados ($\sum |h[n]| < 1{,}0$), la saturación nunca debería activarse, pero su inclusión es una práctica de diseño *senior* que proporciona robustez frente a errores de configuración o condiciones de prueba anómalas.

6. Integración del Subsistema TX y *Timing*

6.1. Sobremuestreo por inserción de ceros

El filtro RRC opera a la tasa de muestreo del sistema ($f_s = 27\,\text{MHz}$), mientras que los símbolos QAM se generan a la tasa de símbolo ($f_{\text{sym}} = f_s/\text{SPS} = 6{,}75\,\text{MHz}$). La conversión entre ambas tasas se realiza mediante un **sobremuestreo por inserción de ceros** con factor $L = 4$:

$$x_{\text{up}}[n] = \begin{cases} s[n/L] & \text{si } n \bmod L = 0 \\ 0 & \text{en otro caso} \end{cases} \tag{27}$$

donde $s[k]$ es la secuencia de símbolos y $x_{\text{up}}[n]$ es la secuencia sobremuestreada. Esta operación crea un *tren de impulsos* que, al ser filtrado por el FIR RRC, produce la forma de pulso RRC deseada.

En la implementación RTL (`tx_top.sv`), el sobremuestreador utiliza una máquina de estados con un contador `up_cnt`:

1. Cuando `sym_valid` se activa, el valor del símbolo (I, Q) se presenta a la salida y el contador se inicializa a 1.
2. Durante los $\text{SPS} - 1 = 3$ ciclos siguientes, la salida se fuerza a cero (*zero-stuffing*).
3. El `up_valid` permanece activo durante los 4 ciclos, alimentando al filtro RRC con una muestra válida en cada flanco de reloj.

6.2. Cadena de latencia del subsistema TX

La latencia total del subsistema, desde la generación de bits hasta la salida del filtro, se compone de:

Tabla 6: Desglose de latencia del subsistema TX.

Módulo	**Latencia (ciclos)**	**Nota**
`bit_gen`	1	Registro de salida
`qam16_mapper`	1	Registro de salida
`upsampler`	0	Combinacional (integrado)
`rrc_filter`	1	Pipeline transpuesto
Total pipeline	**3**	Latencia mínima
Retardo de grupo FIR	16	$(N-1)/2 = 16$ muestras
Total efectivo	**19**	$3 + 16$ muestras

En términos temporales, a 27 MHz:

$$t_{\text{latencia}} = \frac{19}{27\,\text{MHz}} \approx 0{,}704\,\mu\text{s} \tag{28}$$

6.3. Generación del *strobe* de símbolo

La sincronización entre la tasa de símbolo y la tasa de muestreo se gestiona mediante un contador módulo SPS implementado en `tx_top`:

```
logic [$clog2(SPS)-1:0] sps_cnt;
logic sym_strobe;

always_ff @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        sps_cnt    <= '0;
        sym_strobe <= 1'b0;
    end else if (en) begin
        if (sps_cnt == SPS - 1) begin
            sps_cnt    <= '0;
            sym_strobe <= 1'b1;
        end else begin
            sps_cnt    <= sps_cnt + 1'b1;
            sym_strobe <= 1'b0;
        end
    end
end
```

Listing 2: Generador de *strobe* de símbolo.

Este *strobe* de un ciclo de duración se propaga como señal de habilitación (`en`) al generador de bits, garantizando que se produce un nuevo símbolo de 4 bits exactamente cada SPS = 4 ciclos de reloj.

6.4. Uso de `gdsp_pkg` para la mantenibilidad

Todas las constantes, anchuras de bus, tipos y parámetros del sistema se centralizan en el paquete SystemVerilog `gdsp_pkg.sv`. Esta decisión arquitectónica, común en diseños industriales, ofrece las siguientes ventajas:

- **Coherencia**: un cambio en la anchura de palabra (por ejemplo, migrar de Q1.11 a Q1.15) se propaga automáticamente a todos los módulos mediante `import gdsp_pkg::*`.
- **Seguridad de tipos**: el uso de `typedef` (`sample_t`, `coeff_t`, `product_t`, `accum_t`) previene errores de anchura de bus en la instanciación de módulos.
- **Documentación implícita**: los nombres de los parámetros (`FRAC_BITS`, `NUM_TAPS`, `SPS`) hacen el código auto-documentado.
- **Exploración de diseño**: facilita la exploración paramétrica (por ejemplo, variar el número de taps o el factor SPS) sin modificar la lógica de los módulos individuales.

La Tabla 7 resume los tipos definidos en el paquete y su correspondencia con la cadena de procesado.

Tabla 7: Tipos definidos en `gdsp_pkg` y su uso en la cadena TX.

Tipo	**Anchura**	**Uso**
`sample_t`	12 bits	Muestras I/Q (entrada/salida del filtro)
`coeff_t`	12 bits	Coeficientes del filtro RRC
`product_t`	24 bits	Producto dato × coeficiente
`accum_t`	30 bits	Acumulador del filtro FIR

7. Síntesis de decisiones de diseño

A modo de resumen, la Tabla 8 recopila las decisiones de diseño más relevantes del subsistema TX y la justificación técnica de cada una.

Tabla 8: Resumen de decisiones de diseño del subsistema TX.

Aspecto	**Decisión**	**Justificación**
Formato numérico	Q1.11 (12 bits)	SQNR de 68 dB; niveles QAM dentro de rango; compatible con MULT18 de Gowin.
Generador de bits	PRBS-23 (ITU-T O.151)	Secuencia normalizada, período > 8 millones de bits, repetibilidad para verificación *bit-true*.
Entrega paralela	LFSR desenrollado $\times 4$	Evita latencia de acumulación; 4 niveles de XOR (trivial a 27 MHz).
Mapeador QAM	LUT Gray combinacional	Minimiza BER; 1 ciclo de latencia; constantes centralizadas en `gdsp_pkg`.
Normalización	$1/\sqrt{10}$	Potencia media unitaria; valores enteros exactos en Q1.11 (± 1943, ± 648).
Arquitectura FIR	Forma transpuesta	Camino crítico mínimo ($t_{\text{mult}} + t_{\text{add}}$); mapeo directo a pREG+MULT9.
Truncación	Redondeo convergente	Elimina sesgo de DC; implementación en 3 señales (round, sticky, LSB).
Sobremuestreo	Inserción de ceros $\times 4$	Método estándar; máquina de estados de 2 bits; el FIR actúa como filtro anti-imagen.
Parametrización	`gdsp_pkg` centralizado	Coherencia, seguridad de tipos, exploración paramétrica sin modificar módulos.