Probabilistic shaping

Práctica Profesional Supervisada

Dipre Ivan
Viglianco Kevin H.

Supervisior: Dr. Ariel L. Pola

18 de mayo de 2020

Fundación
FULGOR

Tabla de Contenidos

Tabla de Contenidos

Sección 1

Introducción

Introducción

Objetivos y alcances

- Se desea modificar la probabilidad de ocurrencia de cada símbolo en una transmisión digital, cambiando así la forma de su constelación. Esto se conoce como 'Probabilistic Shaping'.
- Para esto se utilizará el método 'Distribution Matching', el cual permite transformar una secuencia de entrada, con distribución uniforme, a una secuencia de salida con una distribución diferente.
- Se simulará e implementará el algoritmo en una FPGA y se analizarán los resultados obtenidos.

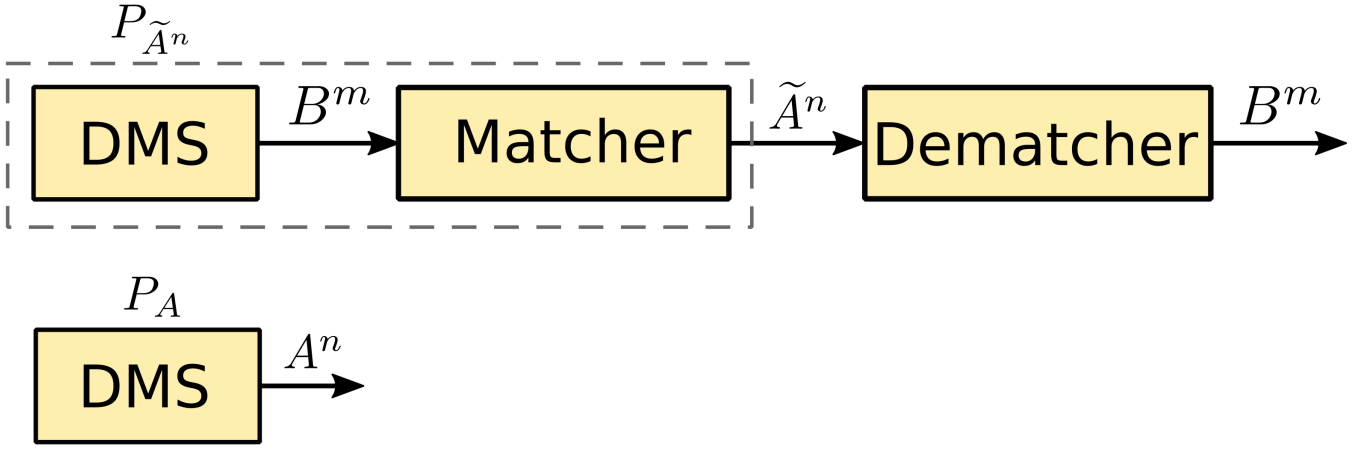


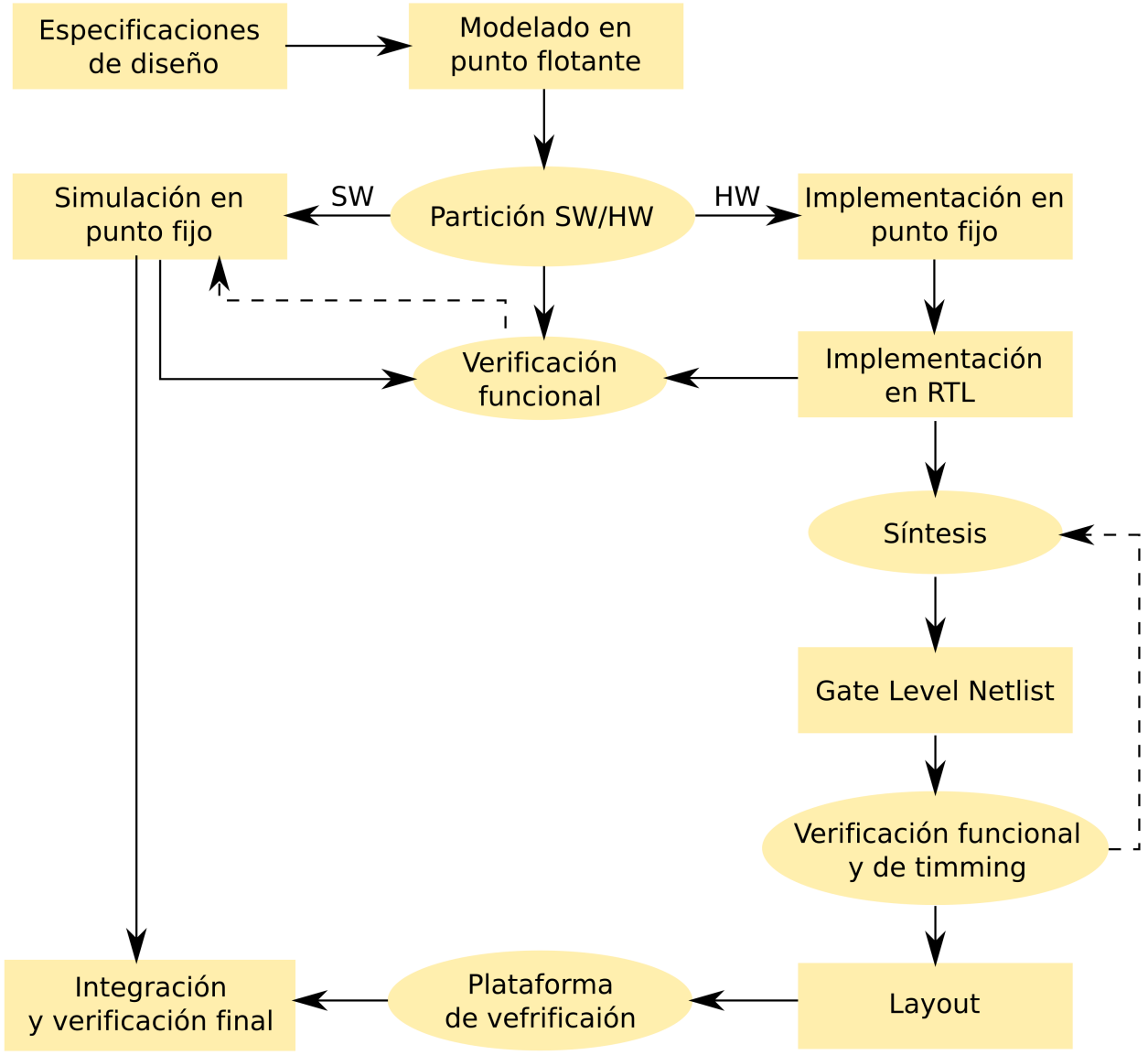
Tabla de Contenidos

Sección 2: Desarrollo

Investigación

Flujo de trabajo

Desarrollo: Investigación



Especificaciones y tipos de algoritmos

Desarrollo: Investigación

Especificaciones de diseño

- A partir de un DMS con distribución $P_{(x=0)} = P_{(x=1)} = 0{,}5$ obtener una distribución con una probabilidad de ceros de $P_{(x=0)} = 0{,}75$.
- La secuencia de entrada al codificador deberá ser de 4 bits.
- Implementar el mismo con una modulación QAM-16.

Tipos de algoritmos

- Según la longitud de la secuencia de datos:
  - Longitud variable a longitud variable (v2v).
  - Longitud variable a longitud fija, o viceversa (v2f o f2v).
  - Longitud fija a longitud fija (f2f).
- Según el modo de calculo:
  - Online, es decir, en tiempo real.
  - Offline, es decir, de forma precalculada.

Selección del algoritmo a utilizar

Desarrollo: Investigación

Algoritmo a utilizar

- Se utilizará el algoritmo 'Constant Composition Distribution Matching' propuesto en [**?**].
- El mismo trabaja con longitudes fijas (f2f) y el calculo es de tipo 'online'.

Características del algoritmo

- Utiliza 'Arithmetic Coding'.
- Asocia un intervalo a cada una de las posibles entradas y salidas en base a su probabilidad
- Realiza un mapeo de cada intervalo de entrada a un único intervalo de salida
- Todas las secuencias de salida tienen una composición constante de '0' y '1'.

Análisis de la longitud de palabra de salida

Desarrollo: Investigación

Longitud de palabra de salida

- Cantidad de bits de entrada
- La probabilidad $P_{(x=1)}$ deseada

Condiciones necesarias

- Se debe cumplir que:

$$\binom{X}{Y} \geq 2^k$$

- Donde:
  - X: Longitud de secuencia de salida.
  - Y: Cantidad de bits de salida en '1'.
  - k: Longitud de secuencia entrada.
- Además se debe mantener la relación:

$$\frac{X}{Y} = \frac{1}{P_{(x=1)}}$$

Análisis de la longitud de palabra de salida

Longitud de salida las para especificaciones de diseño

- Para $k = 4$ y $P_{(x=1)} = 0{,}25$ tendremos:

$$\frac{X}{Y} = \frac{1}{P_{(x=1)}} = 4 \implies X = 4 * Y$$

- A su vez:

$$\binom{X}{Y} \geq 16 \implies X = 8 \implies \frac{X-k}{k} = 100\,\%$$

Ejemplos para otras especificaciones

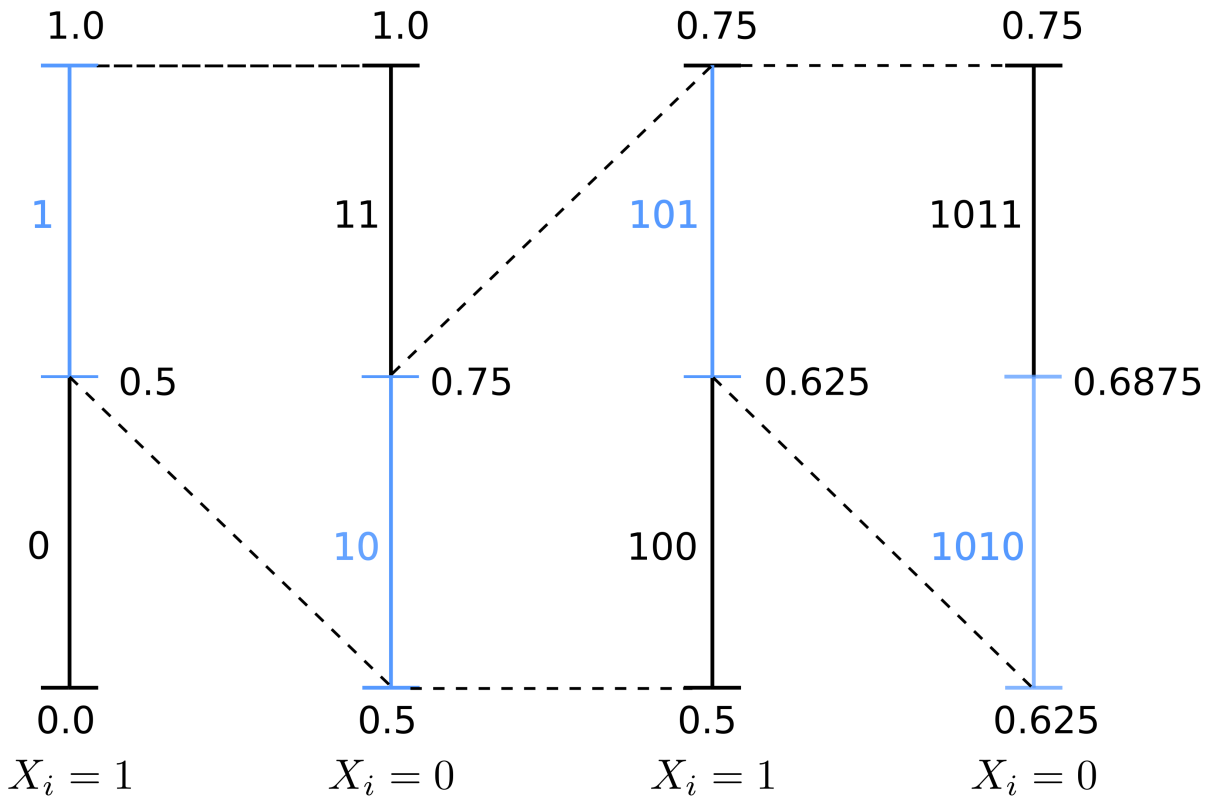
- $k = 16$ y $P_{(x=1)} = 0{,}25 \implies \binom{24}{6} \geq 2^{16} \implies \frac{X-k}{k} = 50\,\%$
- $k = 4$ y $P_{(x=1)} = 0{,}285 \implies \binom{7}{2} \geq 2^{4} \implies \frac{X-k}{k} = 75\,\%$

Calculo de intervalo de entrada del codificador

Desarrollo: Investigación

Ejemplo

- $Xi_{cod} = 1010$ y $P_{(x=0)} = 0{,}75$:

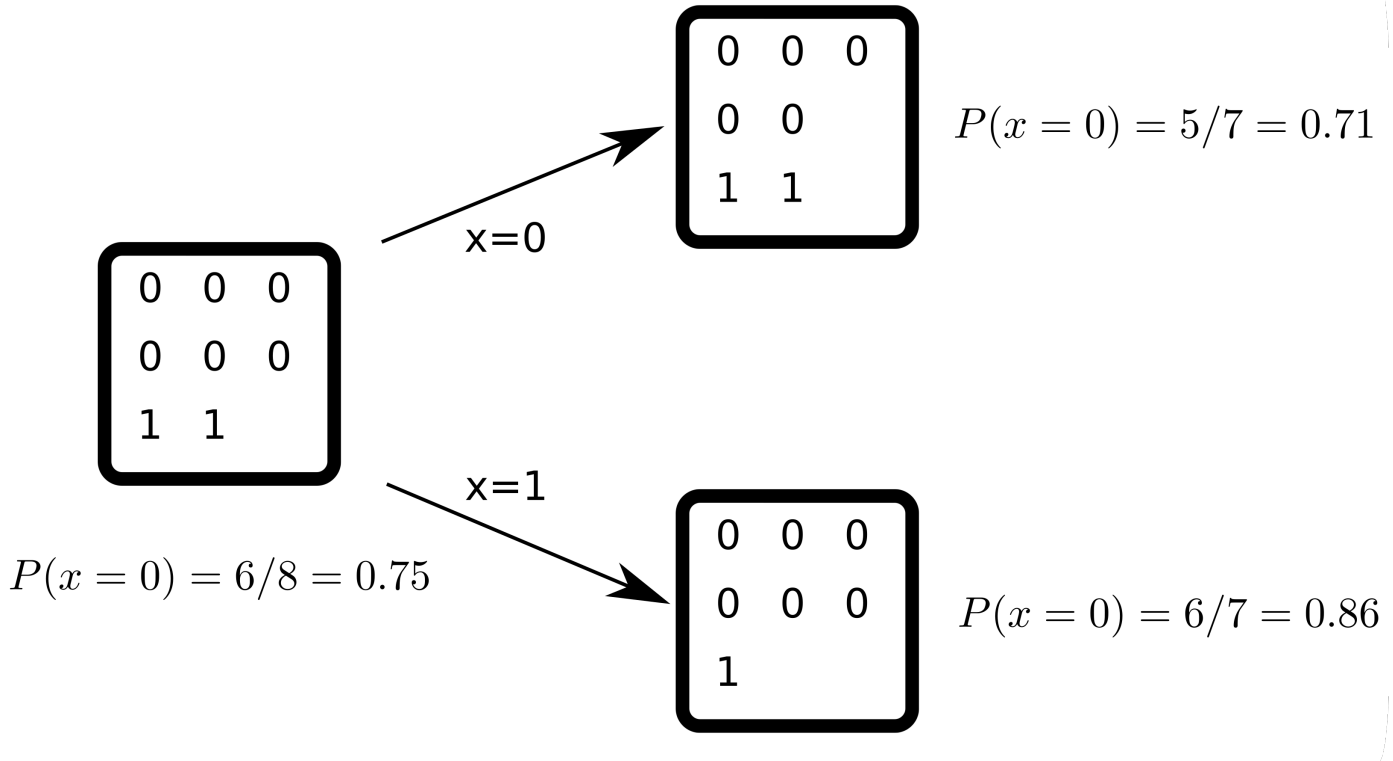


Intervalo de salida del bloque codificador

Desarrollo: Investigación

Concepto de 'Bolsa de bits'

- Propuesto en [**?**, Sec. 4].
- En cada iteración se elimina un bit, obteniendo una nueva probabilidad. $P_{(x=0)}$



Calculo de Intervalo de salida del codificador

Desarrollo: Investigación

Concepto de 'Escalado'

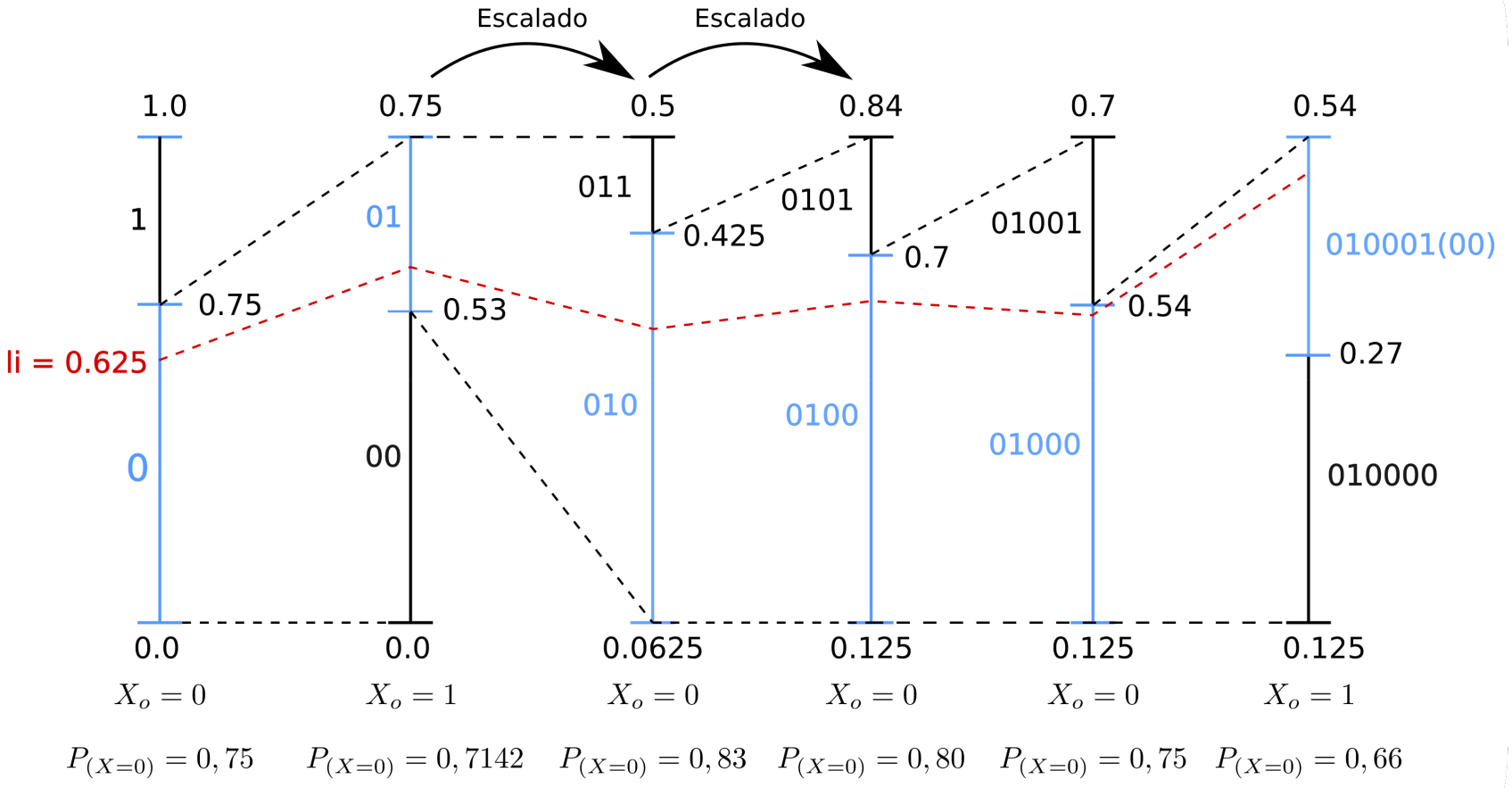
- Propuesto en [**?**, Sec. 4].
- Permite aumentar los límites de los subintervalos cuando estos cumplen con las siguientes condiciones:
  - Si $u_s \leq 0{,}5$:
    - $u'_s = 2u_s$
    - $l'_s = 2l_s$
    - $l'_i = 2l_i$
  - Si $l_s > 0{,}5$:
    - $u'_s = 2(u_s - 0{,}5)$
    - $l'_s = 2(l_s - 0{,}5)$
    - $l'_i = 2(l_i - 0{,}5)$
- Esto permitirá optimizar la cantidad de bits en la implementación.

Calculo de intervalo de salida del bloque codificador

Desarrollo: Investigación

Ejemplo

- $Xi_{cod} = 1010$ y $P_{(x=0)} = 0{,}75$:

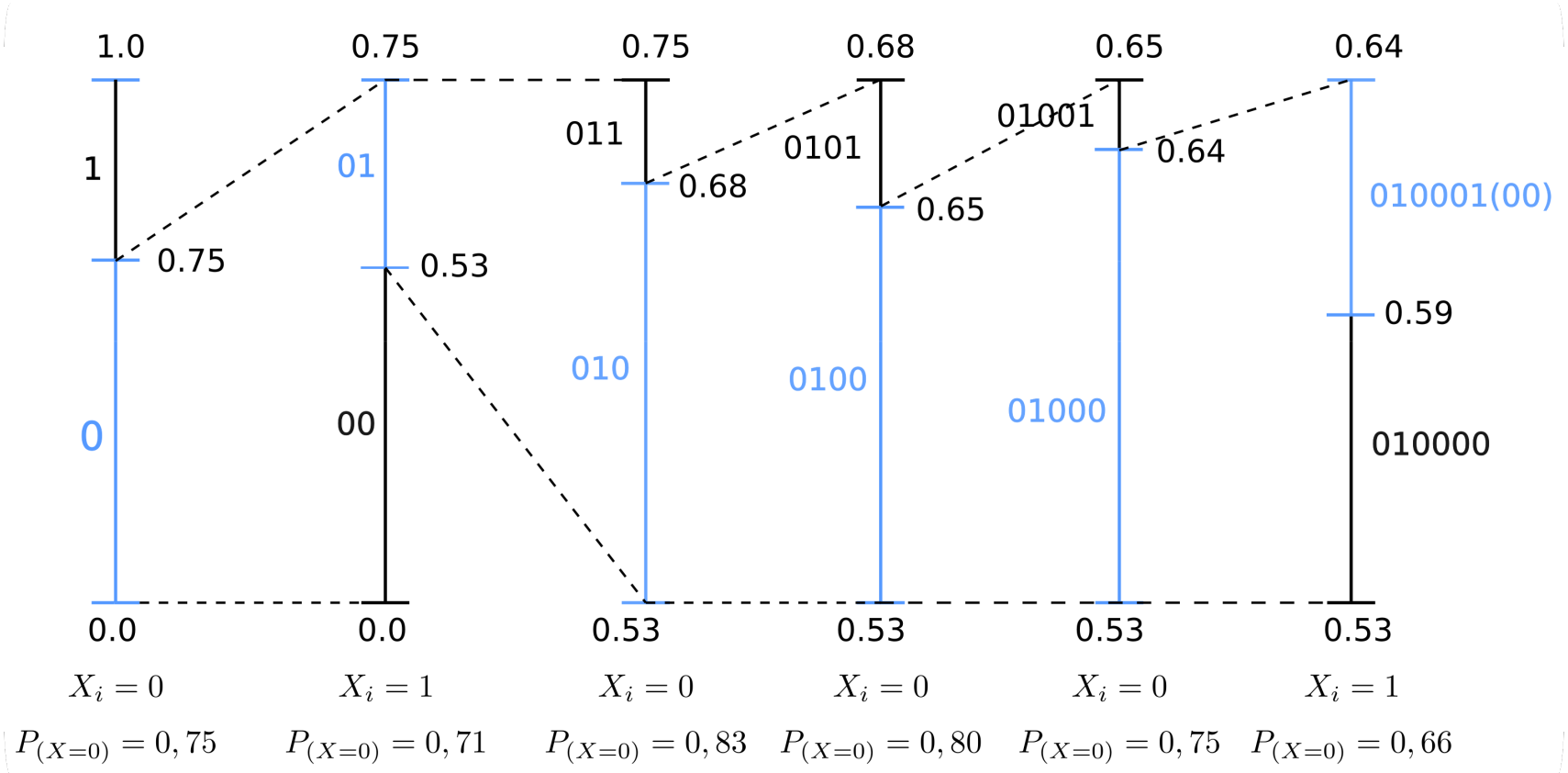


Calculo de intervalo de entrada del decodificador

Desarrollo: Investigación

Ejemplo

- $Xi_{cod} = 01000100$ y $P_{(x=0)} = 0{,}75$:



Calculo de intervalo de salida del decodificador

Desarrollo: Investigación

Ejemplo

- $Xi_{cod} = 01000100$ y $P_{(x=0)} = 0{,}75$:

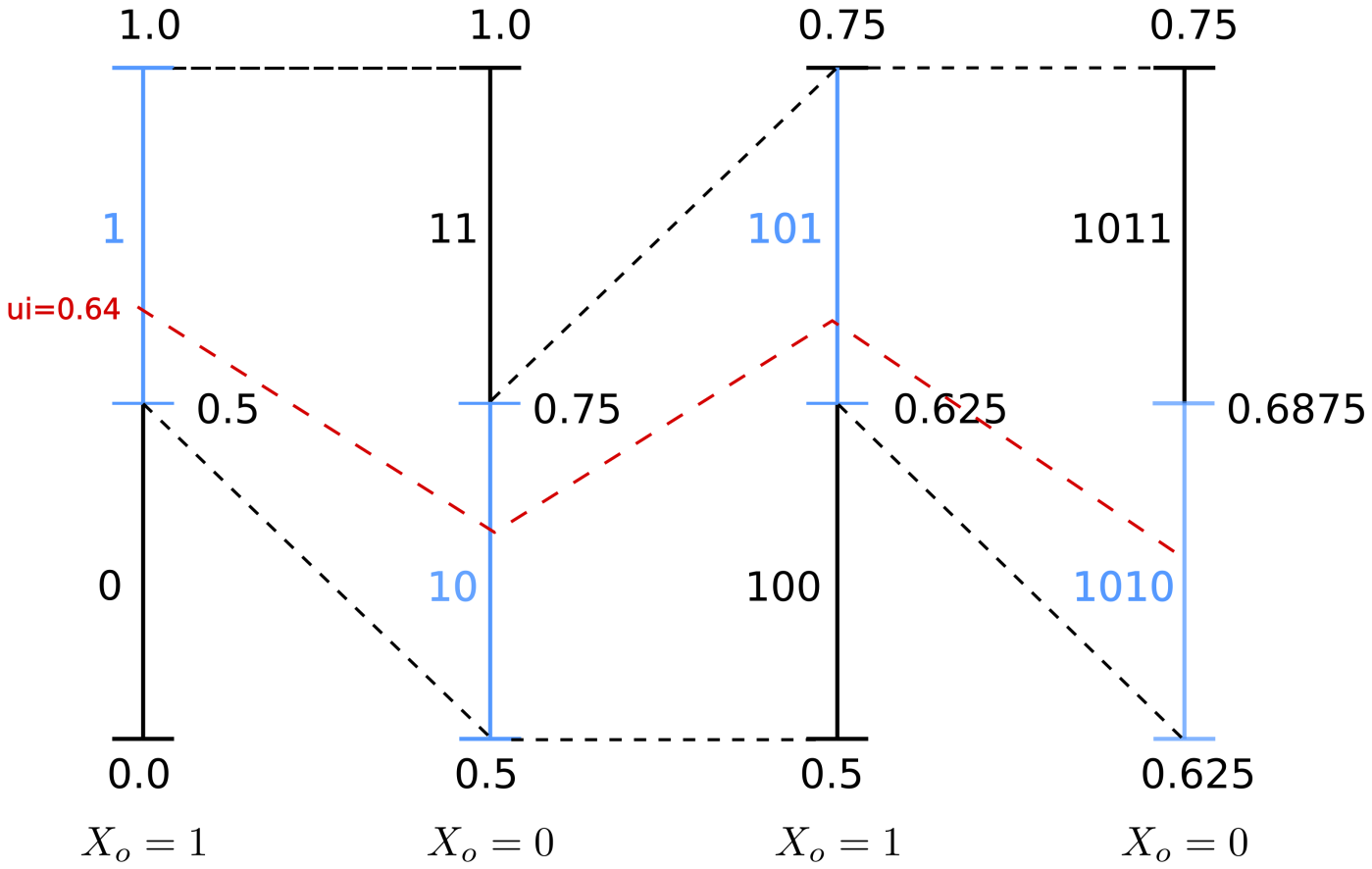


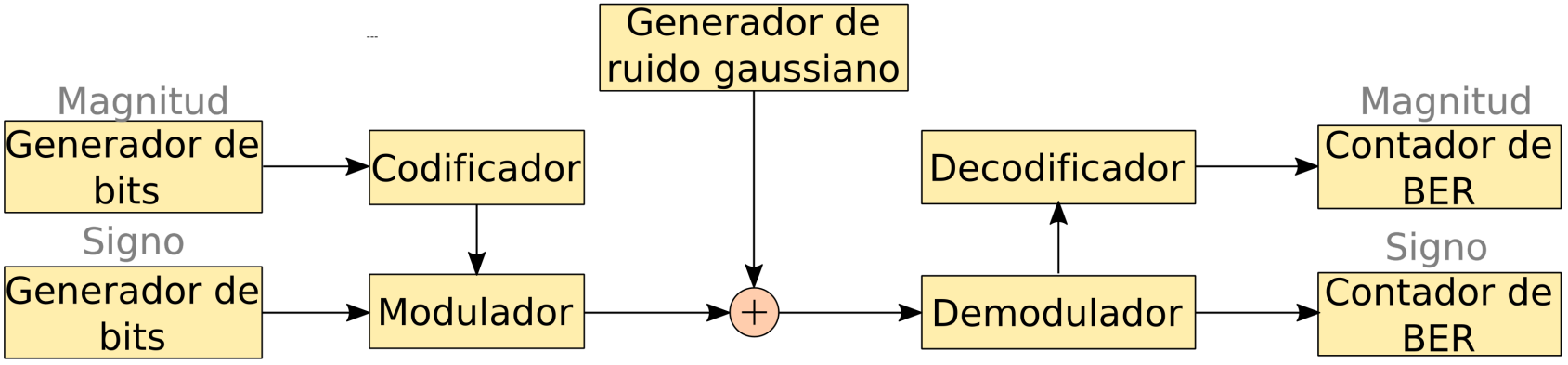
Tabla de Contenidos

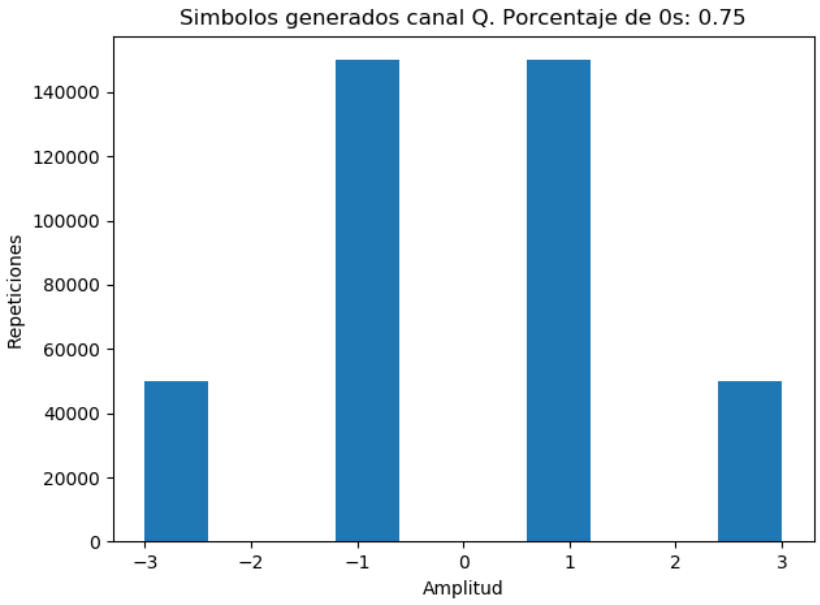
Sección 2: Desarrollo

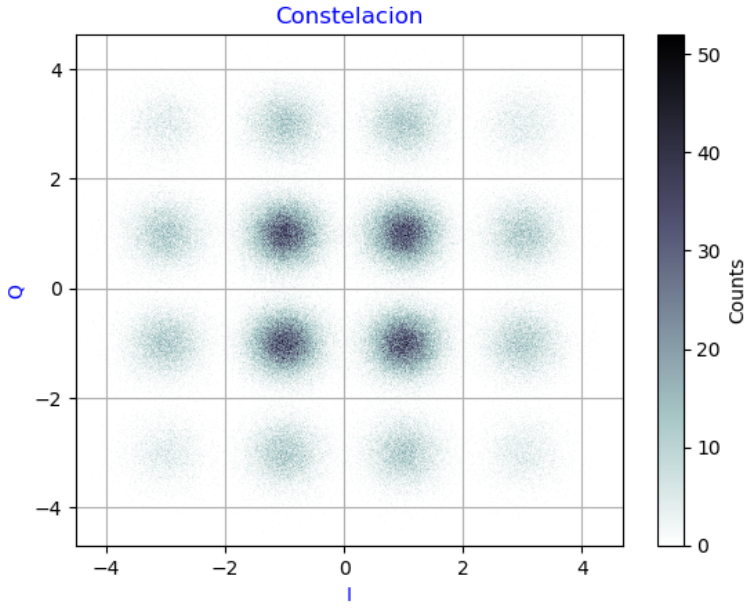
Modelo en punto flotante

Esquema de simulación

Desarrollo: Modelo en punto flotante



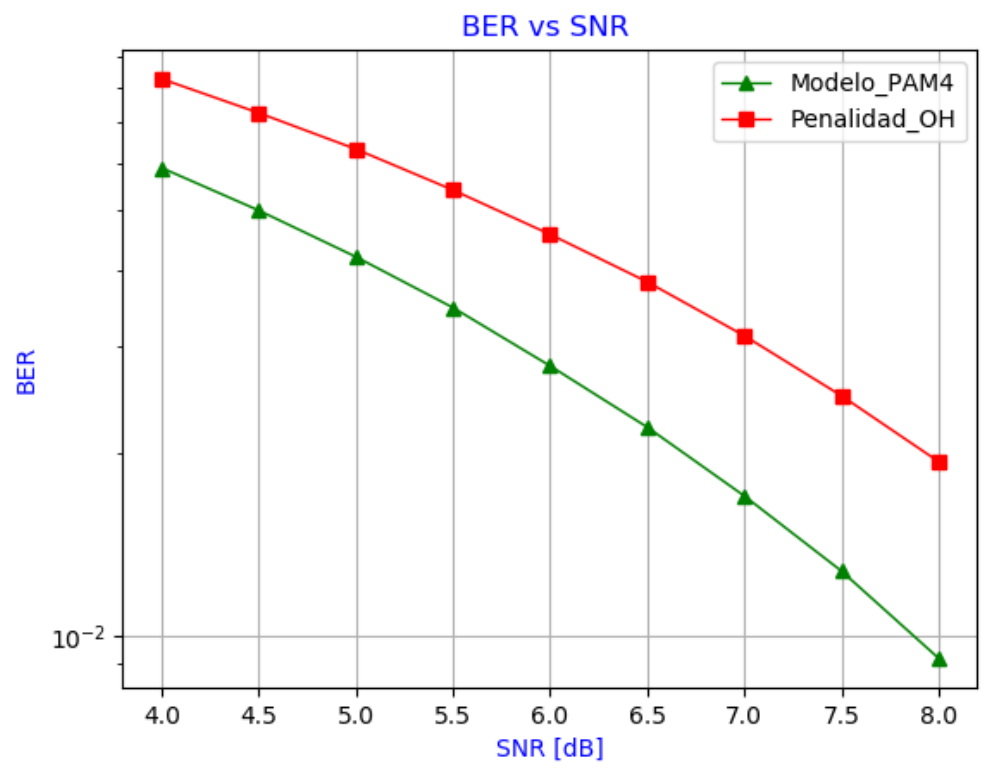




Penalidad por redundancia

Desarrollo: Modelo en punto flotante

- De los 16 bits transmitidos solo 12 son de información.
- Se produce un desplazamiento en la curva dado por la ecuación $10.log(16/12) \approx 1,25\,dB$
- Esto se conoce como 'Overhead' (OH).



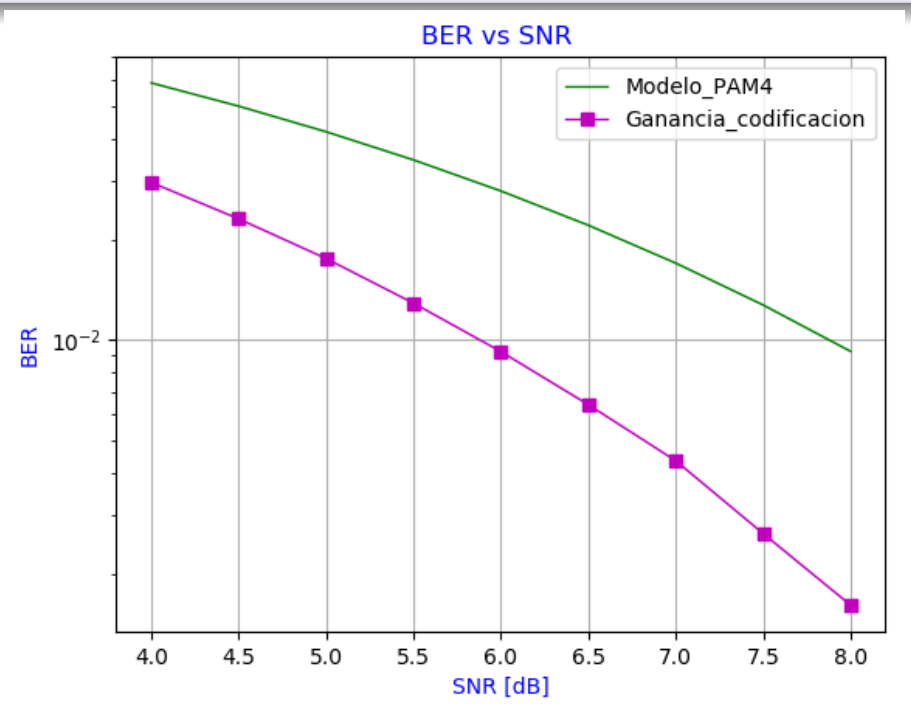
Ganancia por potencia

Desarrollo: Modelo en punto flotante

- La potencia transmitida está dada por:

$$P_{tx} = P_{(s=1)}(1)^2 + P_{(s=3)}(3)^2$$

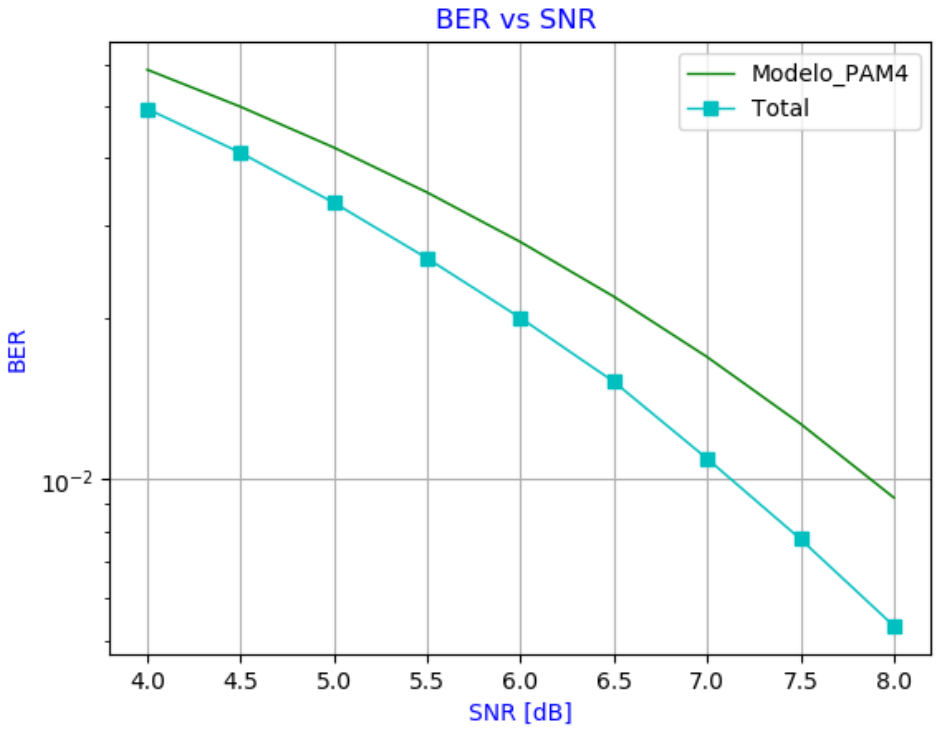
- Para $P_{(s=3)} = 0{,}25$ se obtiene una reducción de potencia del 40 % respecto a $P_{(s=3)} = 0{,}50$.

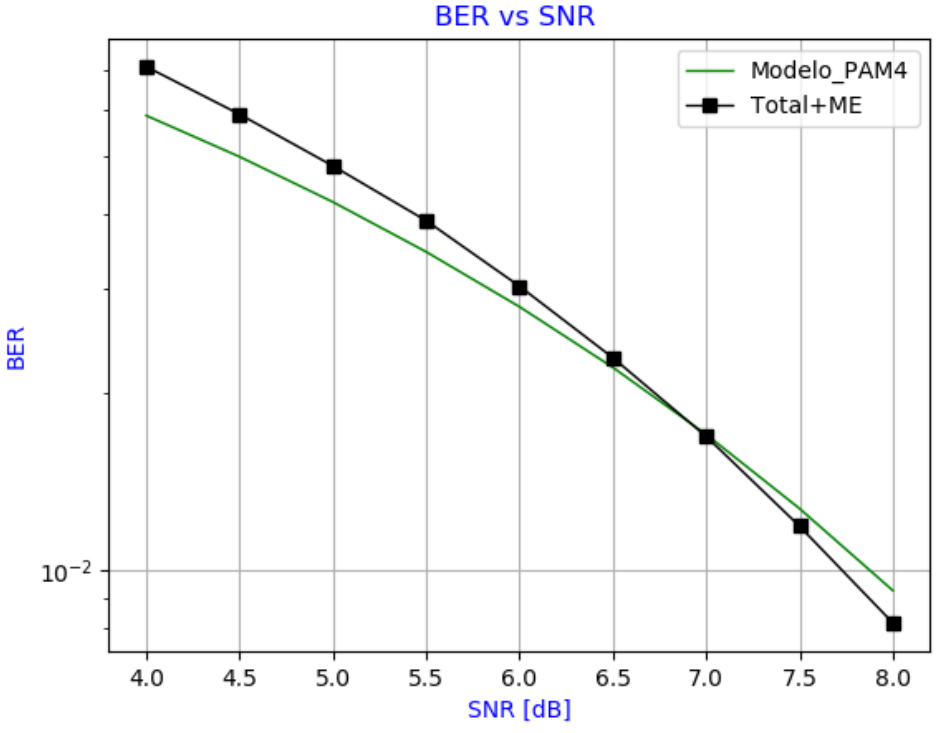


Multiplicación de errores

Desarrollo: Modelo en punto flotante

- Un error de un bit transmitido puede verse reflejado en hasta 4 bits a la salida del decodificador.
- Considerando este efecto, el sistema presenta un mejor desempeño que una modulación PAM-4 para una SNR mayor a 7dB.

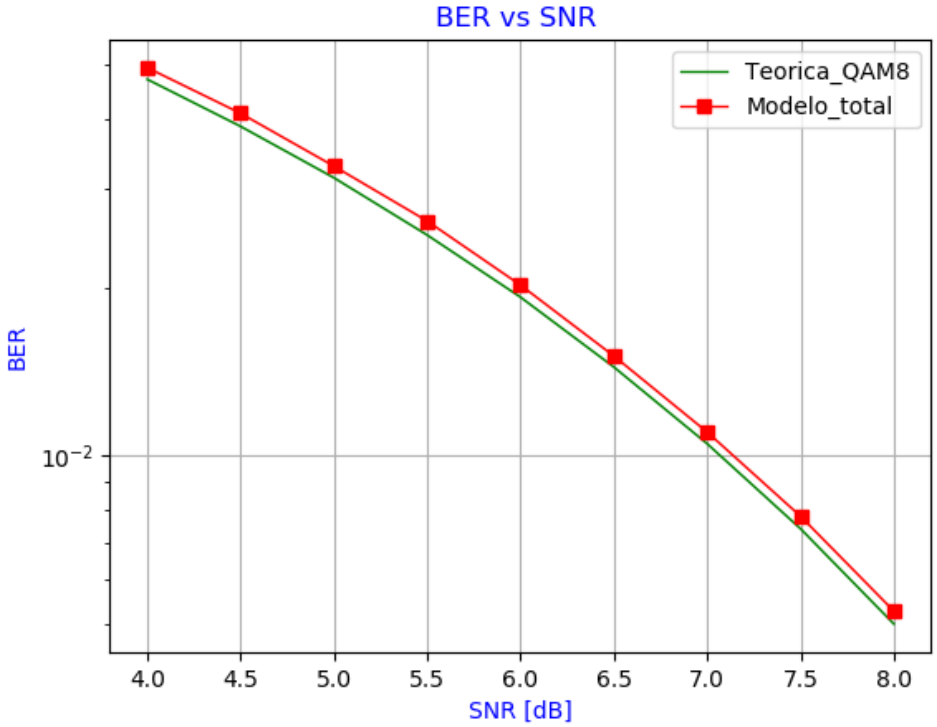


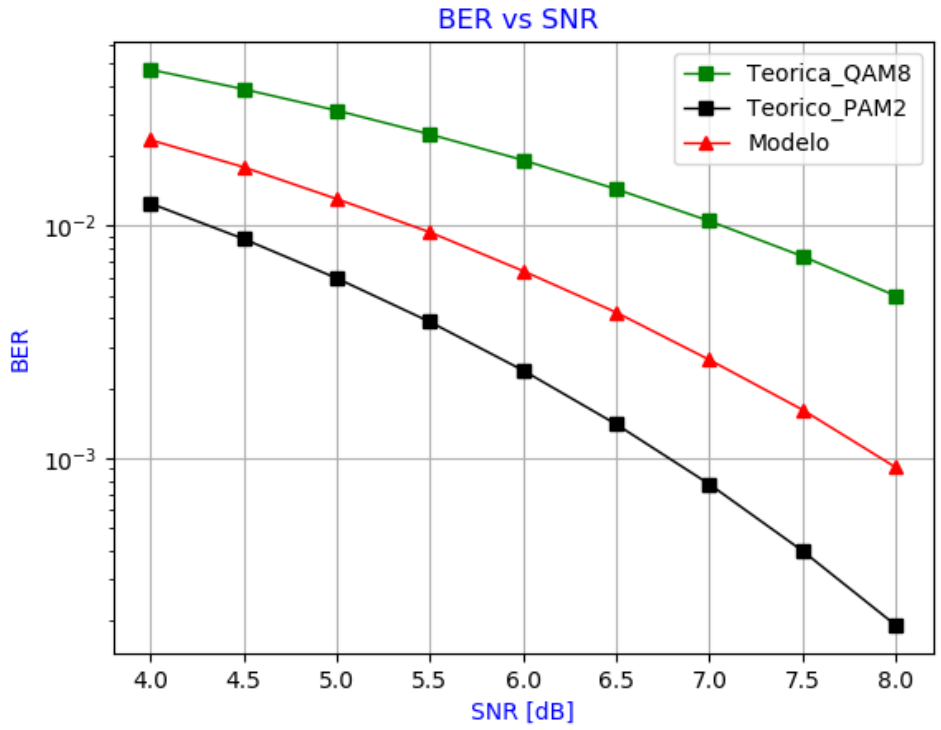


Modelo final

Desarrollo: Modelo en punto flotante

- Comparación con QAM8.
- Secuencia de entrada de 100 bits.
- Probabilidad de $P_{(s=3)} = 0{,}12$.



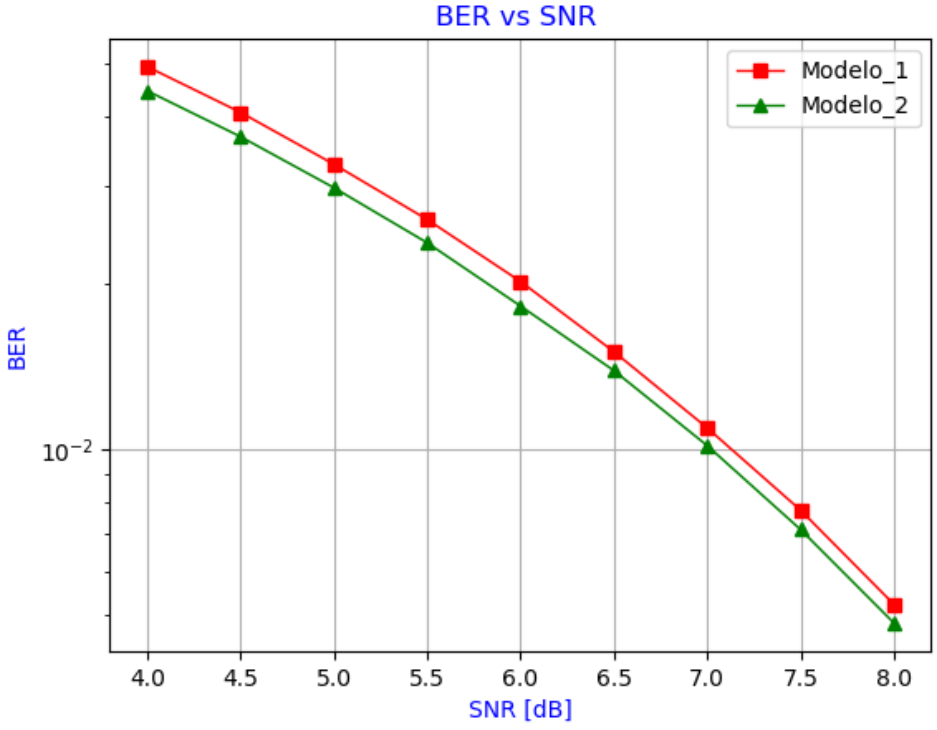


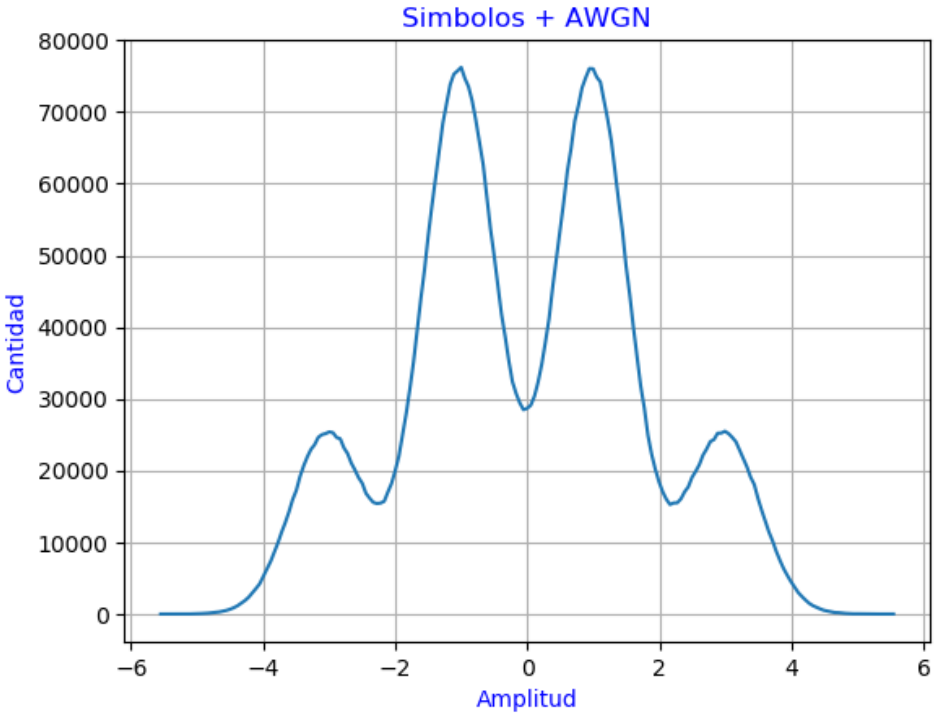
Umbral de decisión

Desarrollo: Modelo en punto flotante

Umbral de decisión

- Mejora en el desempeño.
- La diferencia se reduce a medida que la SNR aumenta.





Umbral de decisión

Desarrollo: Modelo en punto flotante

Umbral de decisión

- Está dado por:

$$P_{(s=1)} * exp(\frac{-(\tau - 1)}{2\sigma^2}) = P_{(s=3)} * exp(\frac{-(\tau - 3)}{2\sigma^2})$$

- Donde:

τ : Umbral de decisión.
σ^2 : Varianza del ruido.

- Para $P_{(s=1)} = 0{,}75$:

$$\tau = 2 + 0{,}55 * \sigma^2$$

Tabla de Contenidos

Sección 2: Desarrollo

Modelo en punto fijo

Consideraciones

Desarrollo: Modelo en punto fijo

Calculo de intervalos

- El calculo del intervalo de entrada se realiza a la mitad de frecuencia que el de salida en el bloque codificador, de forma inversa en el decodificador.
- El cálculo del intervalo de entrada y salida del codificador se realiza de manera recursiva y secuencial respectivamente.
- El cálculo del intervalo de entrada y salida del decodificador se realiza de forma recursiva.

Cuantizacion de variables

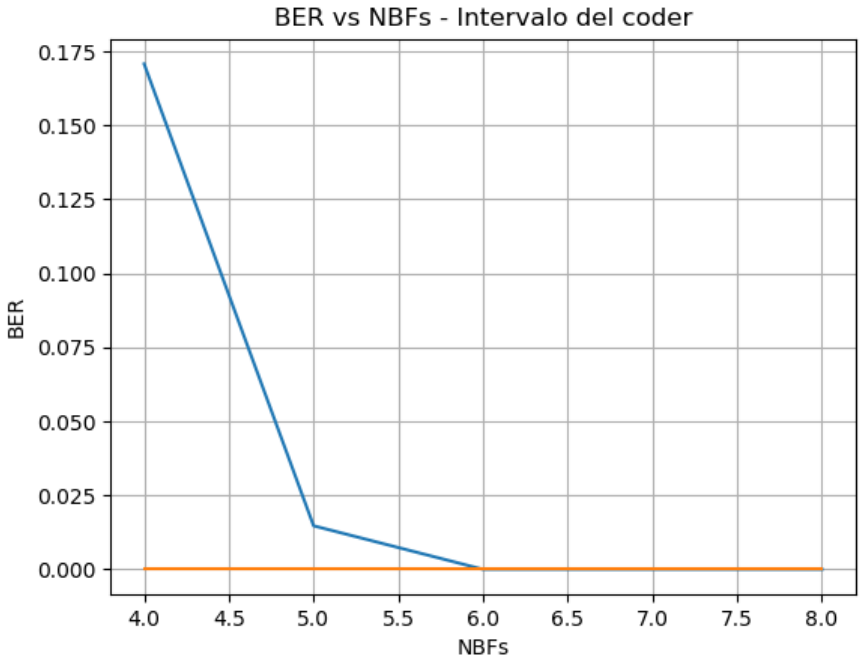
Se consideran tres grupos de variables:

- Intervalos del codificador.
- Probabilidades.
- Intervalos del decodificador.

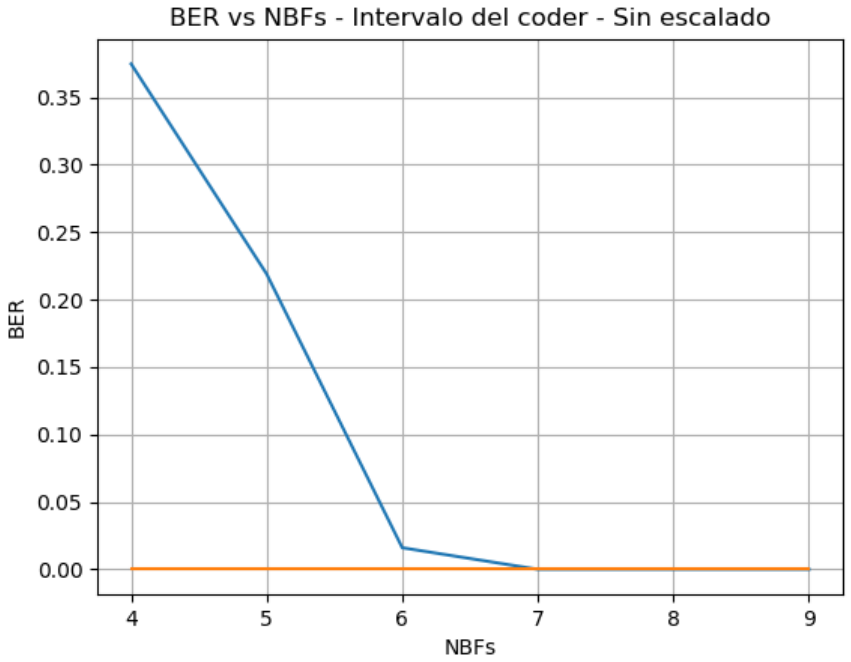
Resoluciones de intervalos del codificador

Desarrollo: Modelo en punto fijo

- Se cuantizan únicamente los limites de los intervalos del codificador.
- La resolución elegida para este grupo de variables será U(7,6).



Con escalado

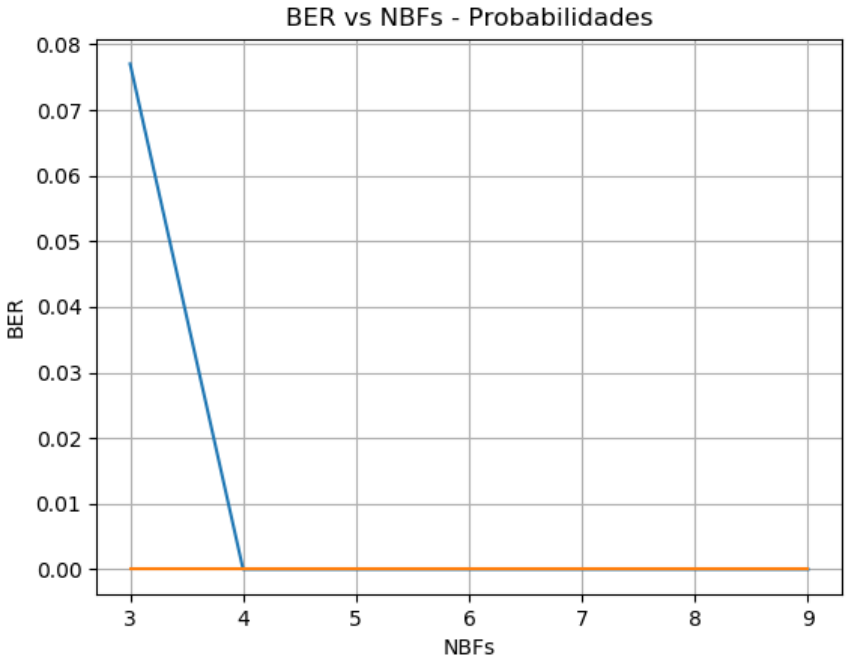


Sin escalado

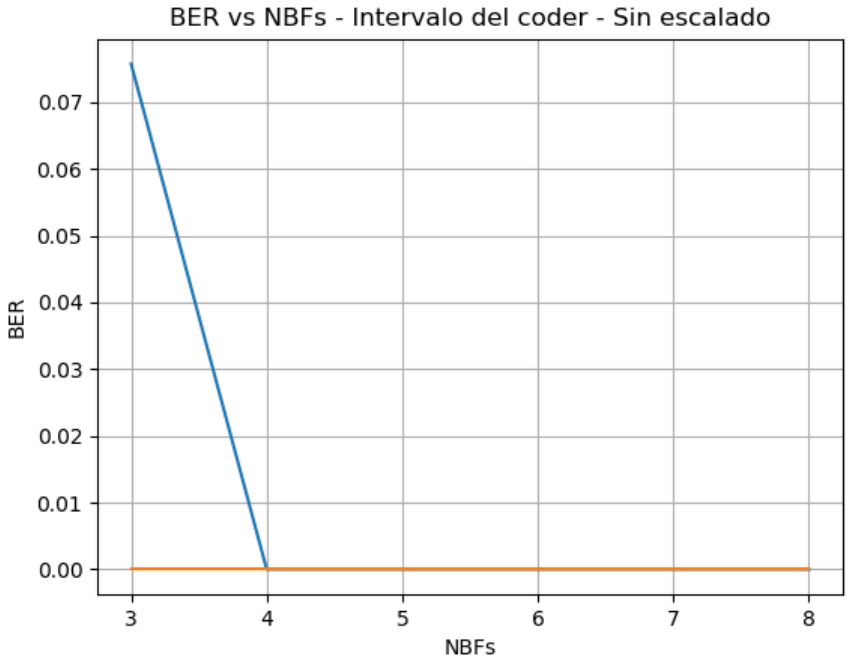
Resolución de probabilidades

Desarrollo: Modelo en punto fijo

- La resolución de los intervalos del codificador se mantiene constante.
- La resolución elegida para este grupo de variables será U(5,4).



Con escalado

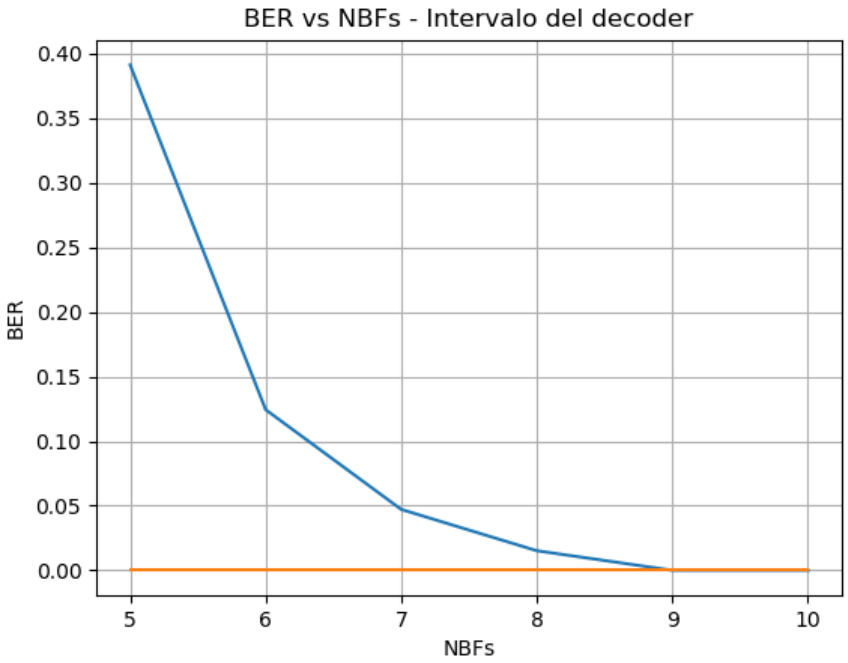


Sin escalado

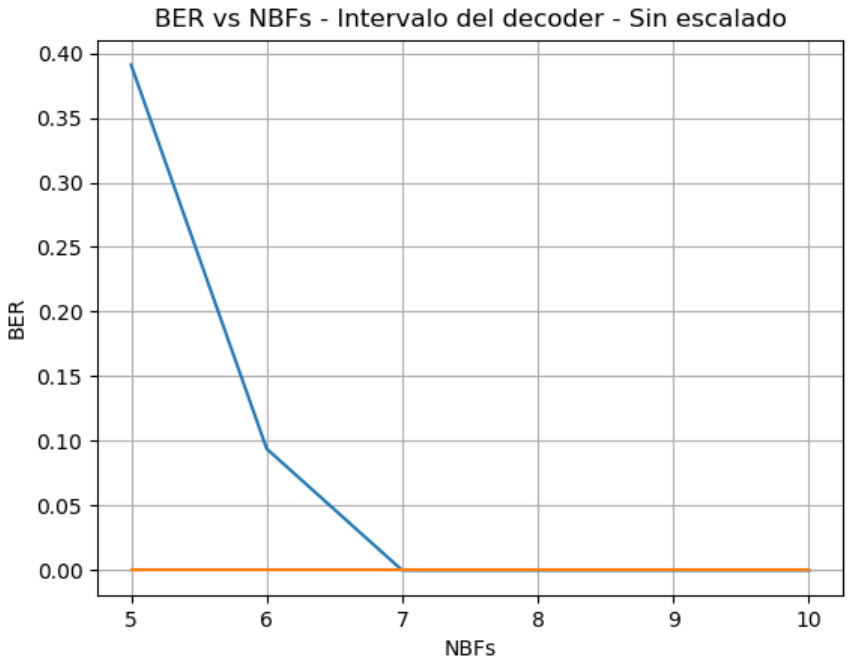
Resolución de intervalos del decodificador

Desarrollo: Modelo en punto fijo

- Se mantiene constante la resolución de los intervalos del codificador y las variables de probabilidad.
- La resolución elegida para este grupo es U(10,9).



Con escalado



Sin escalado

Consideraciones finales

Desarrollo: Modelo en punto fijo

Retardos en la salida del codificador

- Se deben incluir de tal forma que no afecte el sincronismo de datos.
- Se agregó una cantidad de retardos en el canal de comunicación para que esto no sea un problema.

Ejemplo

La secuencia [1,0,0,0] se codifica a [0,1,0,0,0,0,1,0], si el canal presenta un retardo, la secuencia será [1,0,0,0,1,0,0,0] y decodificacion resultara erronea.

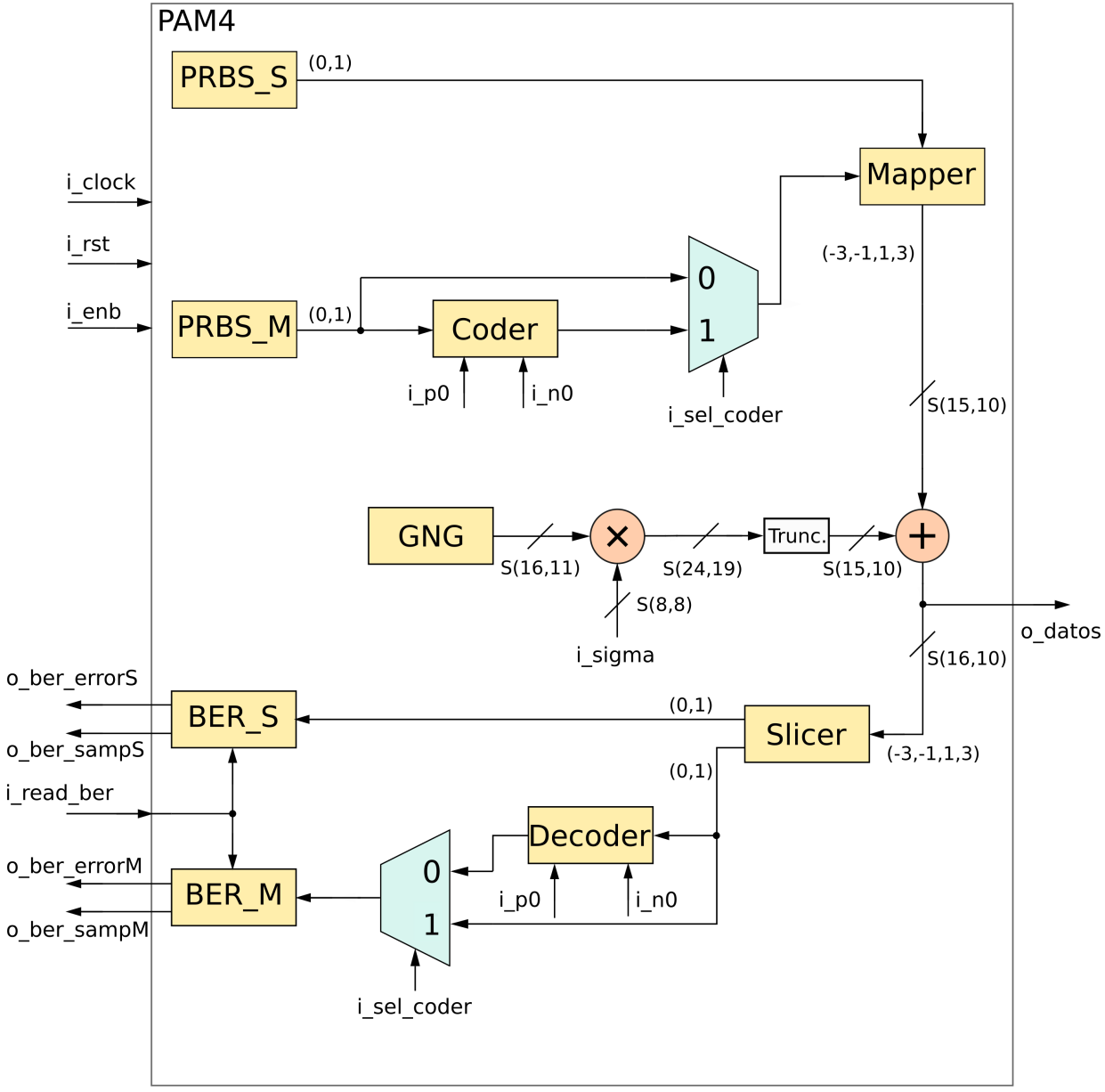
Tabla de Contenidos

Sección 2: Desarrollo

Implementación

Bloque PAM-4

Desarrollo: Implementación

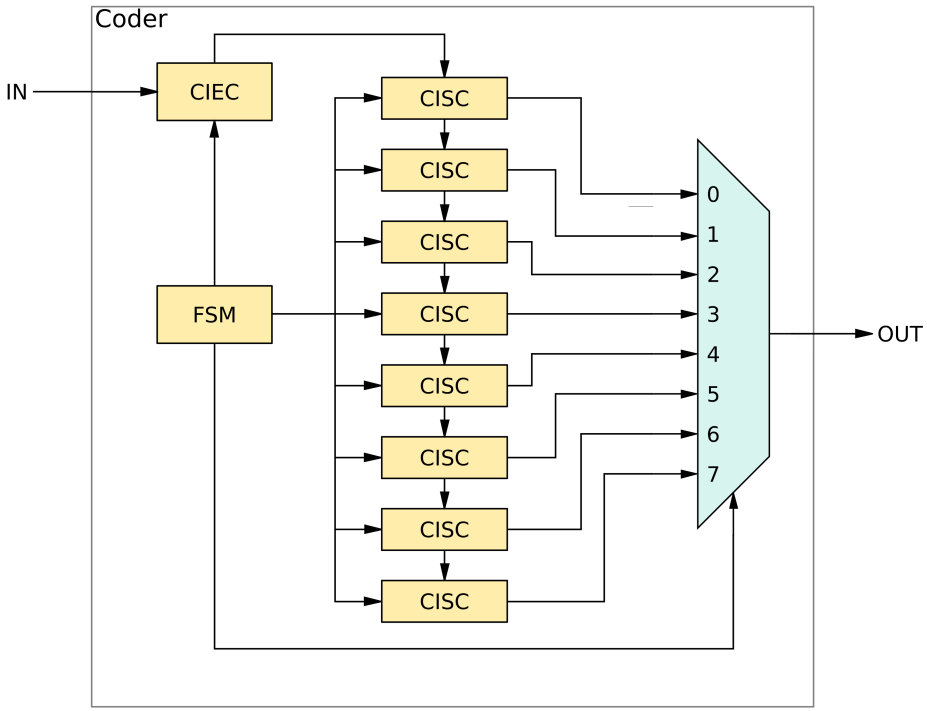


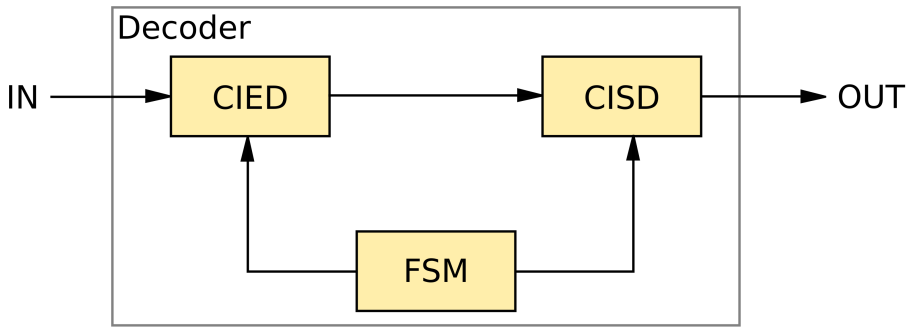
Bloque codificador y decodificador

Desarrollo: Implementación

Estructura interna

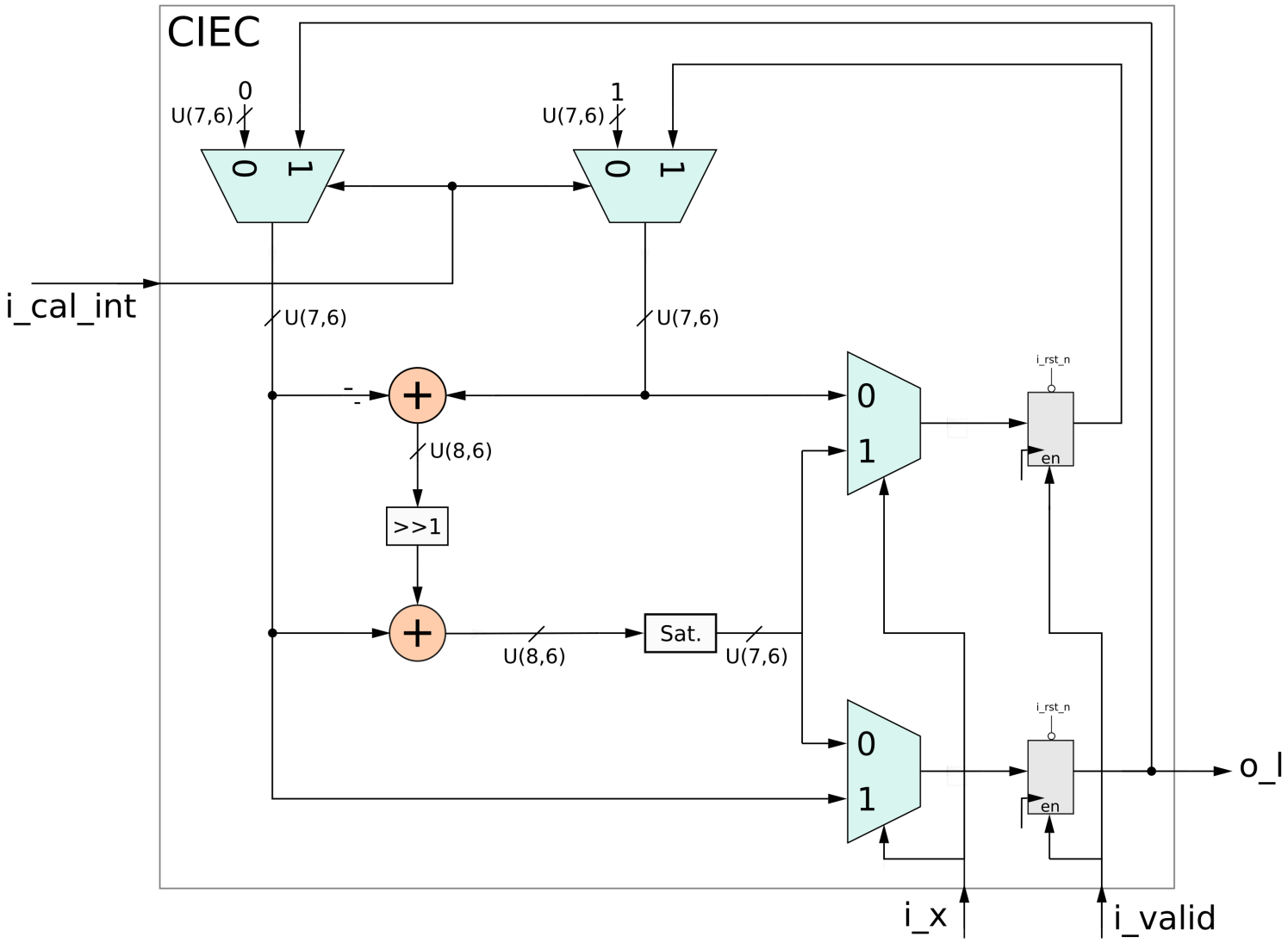
El bloque CISC se ejecuta de forma secuencial, mientras que los bloques CIEC, CIED y CISD se ejecutan de forma recursiva.





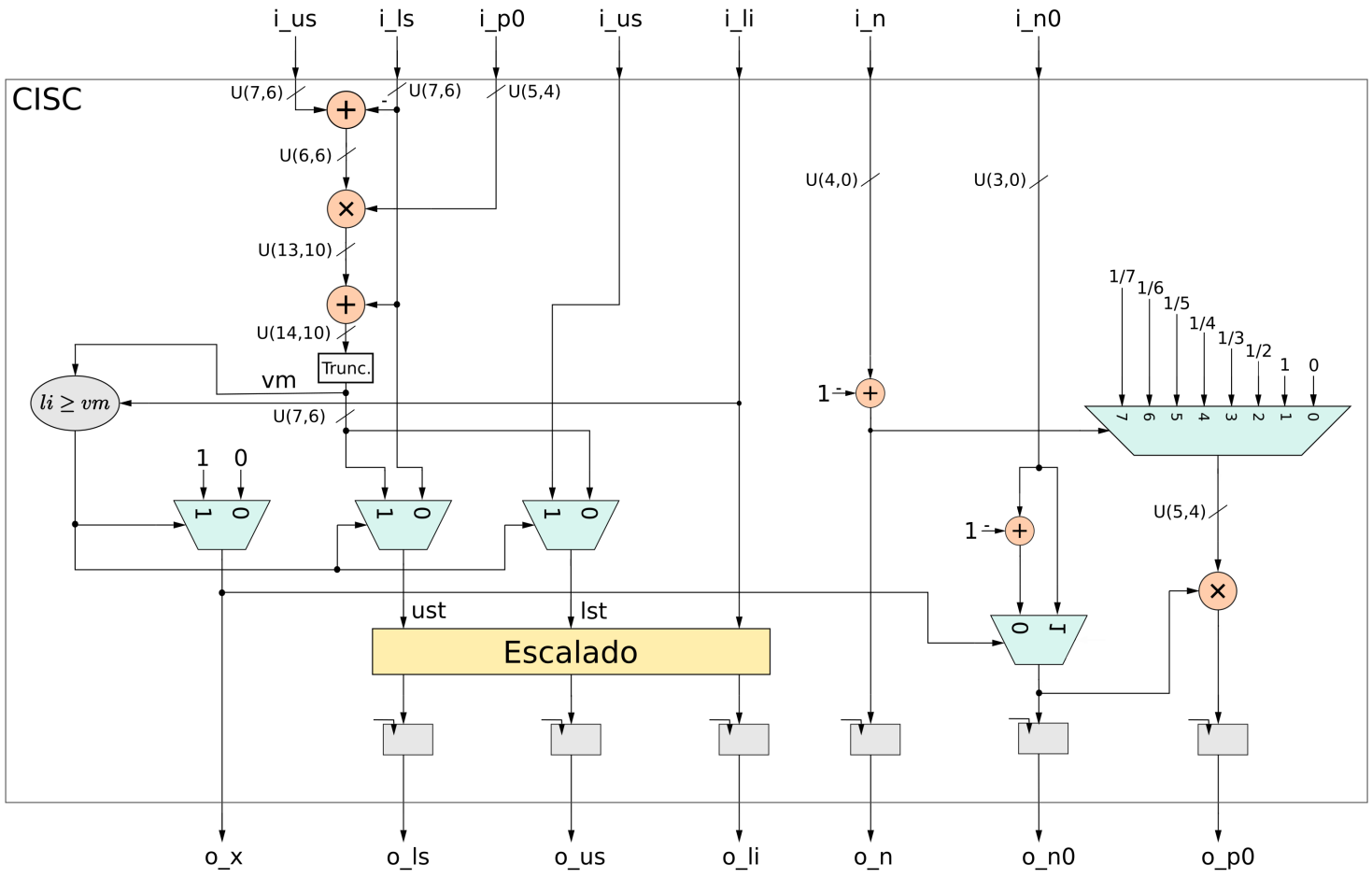
Estructura interna del CIEC

Desarrollo: Implementación



Estructura interna del CISC

Desarrollo: Implementación

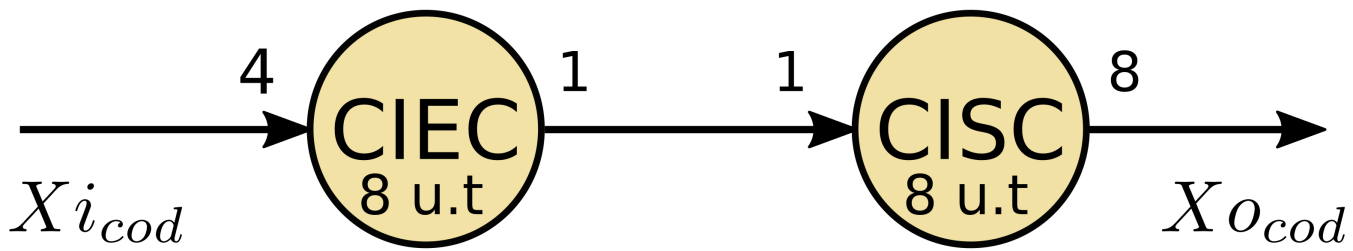


Bloque codificador y decodificador

Desarrollo: Implementación

Flujo de datos

- Los bloques CIEC requiere de 4 u.t. pero es habilitado la mitad de los ciclos, por lo que requieren 8 u.t.
- El bloque CISC se ejecuta cada ciclo, por lo que se requieren 8 u.t.
- Esto implica que se generan 8 bits de salida cada 16 unidades de tiempo.



Bloque codificador y decodificador

Desarrollo: Implementación

Ejecución en paralelo

- Para evitar un tiempo de procesamiento total de 16 u.t. se corren ambos procesos en paralelo.
- Esto implica que se generan 8 bits de salida cada 8 u.t.

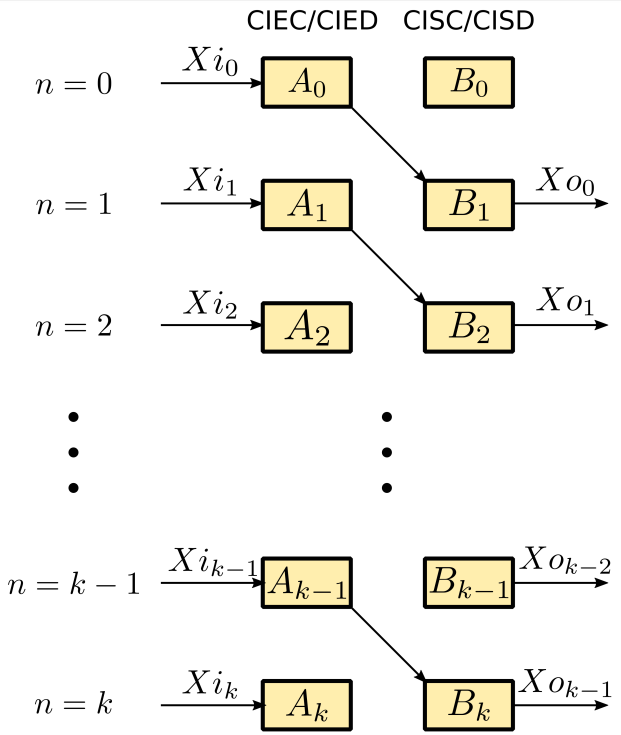


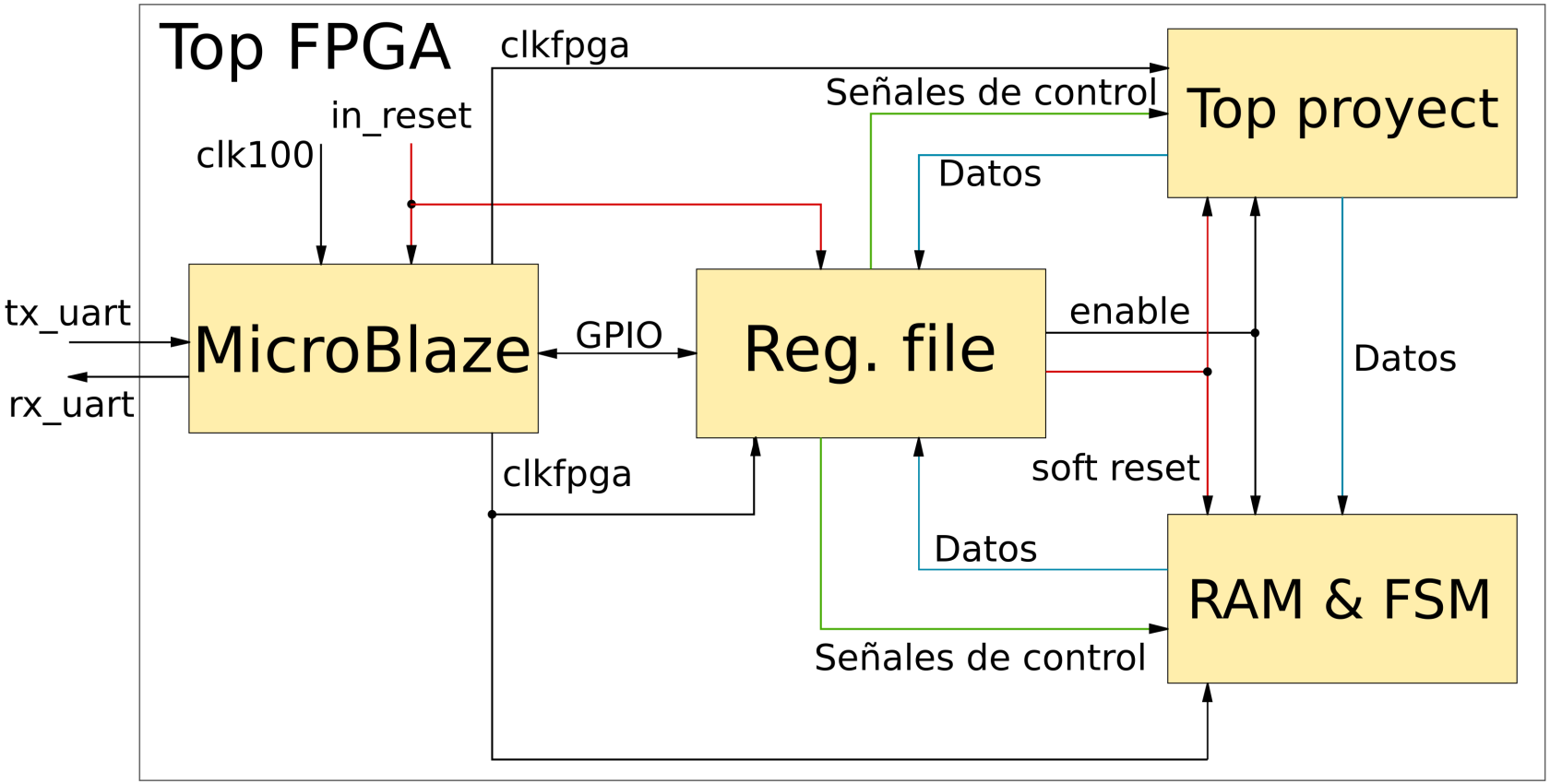
Tabla de Contenidos

Sección 2: Desarrollo

Verificación

Plataforma de verificación

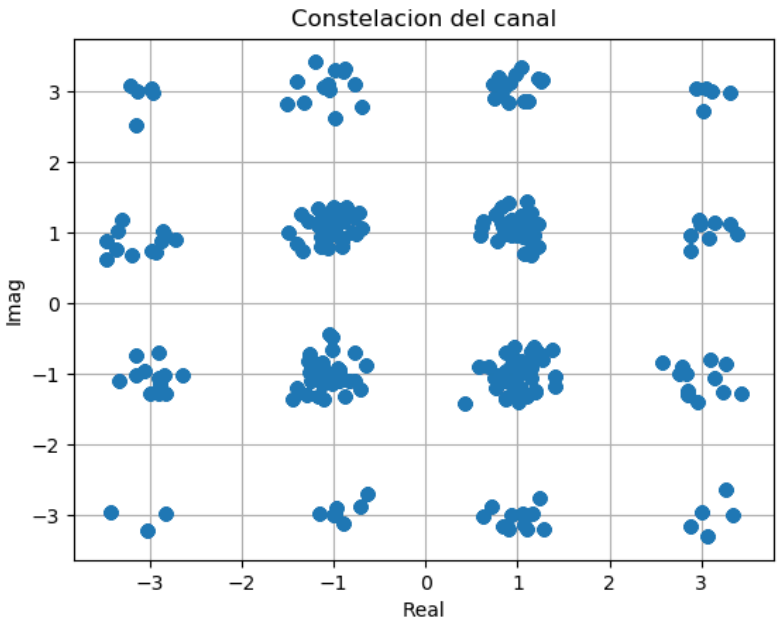
Desarrollo: Verificación

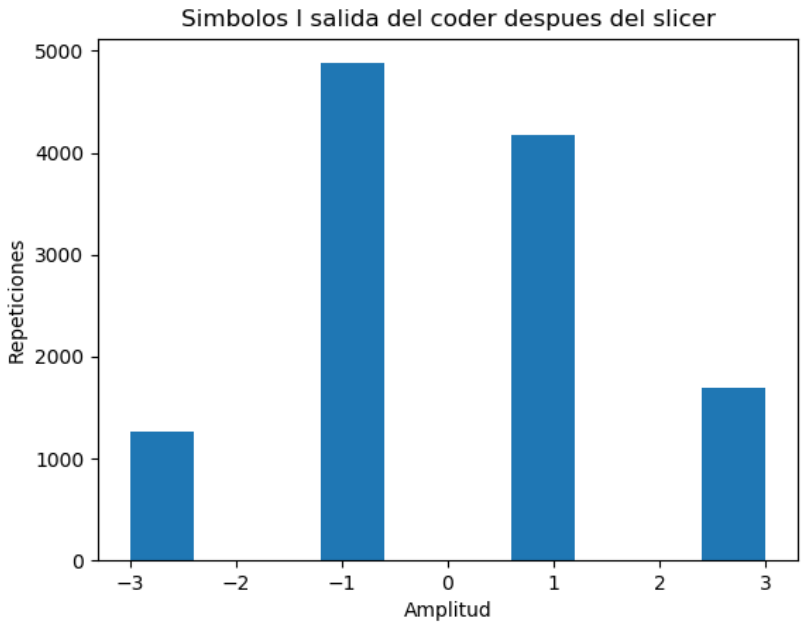


Resultados obtenidos

Desarrollo: Verificación

Símbolos codificados, $P_{(x=0)} = 0{,}75$

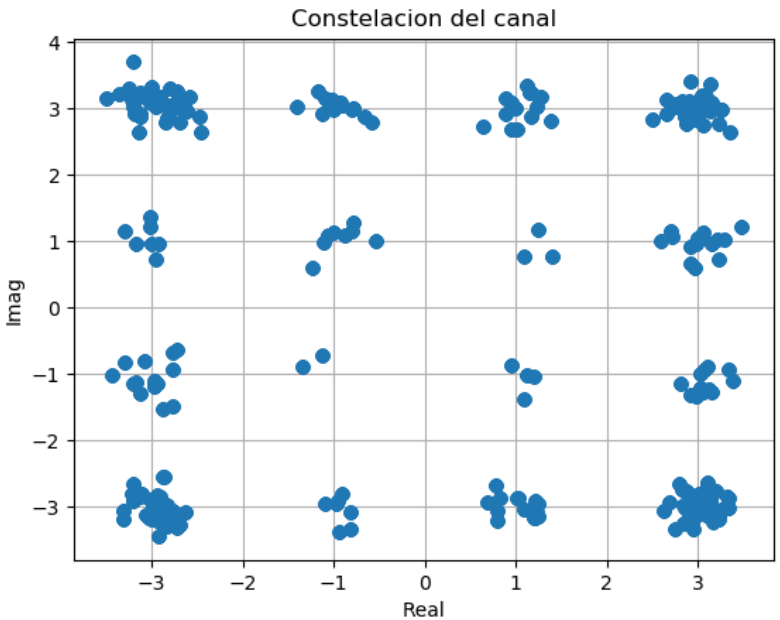




Resultados obtenidos

Desarrollo: Verificación

Símbolos codificados, $P_{(x=0)} = 0{,}25$



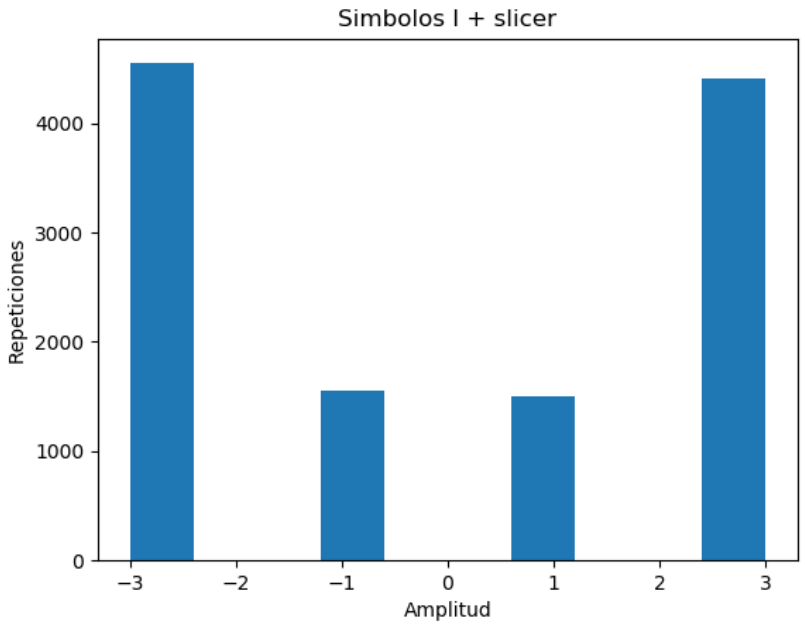


Tabla de Contenidos

Sección 3

Conclusiones

Conclusiones

Técnica 'probabilistic shaping'

- Permite disminuir la tasa de error total del sistema.
- Agrega una determinada cantidad de bits de redundancia.
- Se vuelve mas eficiente a medida que la longitud de entrada aumenta.

Técnica 'constant distribution matching'

- Se debe utilizar técnica de escalado.
- La actualización de la nueva probabilidad implica realizar una división.
- Fácil adaptación a nuevas distribuciones de probabilidad.

Implementación

- El calculo de los intervalos, se puede realizar de manera recursiva o secuencial.
- El sistema implementado es poco practico para una longitud de entrada de 4 bits.

Bibliografía

Preguntas

Muchas Gracias!