

Tema 1: Introducción al diseño digital de sistemas de comunicaciones

Hipólito Guzmán Miranda
Departamento de Ingeniería Electrónica
Universidad de Sevilla
hguzman@us.es

Contexto docente

BT01: Componentes y sistemas digitales para comunicaciones

- Tema 1: Introducción al diseño digital de sistemas de comunicaciones
- Tema 2: Arquitectura de FPGAs avanzadas

Conocimientos previos requeridos:

- Electrónica digital básica
- Fundamentos de procesamiento de señal

Objetivos de aprendizaje

- Conocer el papel de las comunicaciones digitales en el actual contexto tecnológico de las Tecnologías de la Información y Comunicaciones
- Conocer el diseño a nivel de grandes bloques del trabajo de curso
- Saber plantear el diseño hardware de un bloque de procesamiento de señal de acuerdo a criterios de área y tiempo de procesamiento

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Analógico vs. digital

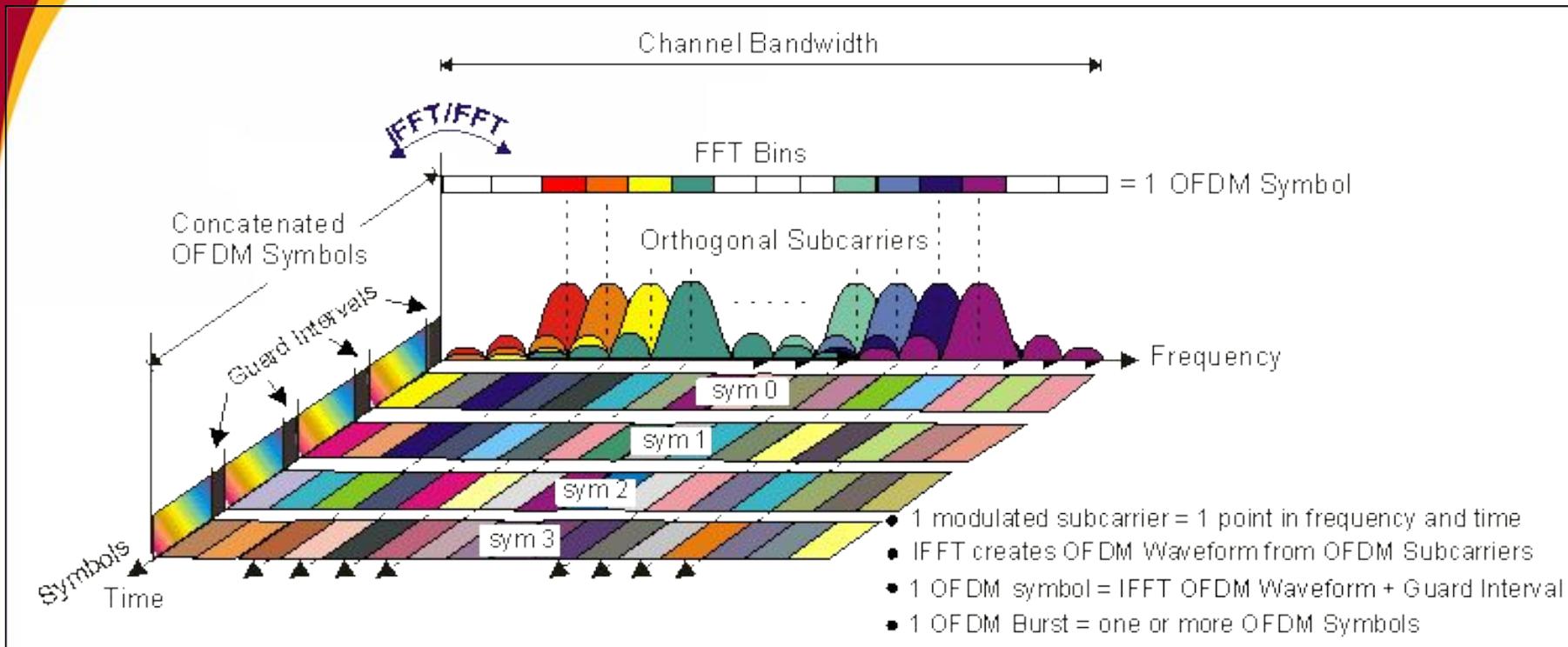
- Saturación del espectro radiológico
- Ruido y distorsión en recepción no corregibles
- Necesidad de elevados niveles en recepción
- Transmisión de datos muy limitada
- Mayor aprovechamiento del ancho de banda (permite mayor número de canales)
- Corrección de datos recibidos
- Menor potencia de transmisión
- Recepción portátil y en movimiento
- Servicios adicionales gracias a la transmisión de datos

Influencia de 50+ años de Ley de Moore en los circuitos de comunicaciones

- Digitalización de los transceptores
- Procesado digital en banda base y frecuencia intermedia
 - RF sigue teniendo procesado analógico
- Modulación multi-portadora
 - OFDM Presente en múltiples estándares
- Software-Defined Radio
 - También aplicable a FPGA (HDL-defined radio?)

OFDM

Orthogonal Frequency-Division Multiplexing



Frequency-Time Representative of an OFDM signal

Fuente: Keysight

OFDM

Ventajas

- Alta eficiencia espectral
- Mayor robustez a interferencia entre símbolos
- Eficiencia en entornos multitrayecto
- Realización hardware eficiente mediante iFFT / FFT
- Permite redes de frecuencia única (reutilización de frecuencia)

Inconvenientes

- Pérdida de eficiencia espectral por la inserción del intervalo de guarda o el prefijo cíclico
- Alta sensibilidad a errores de sincronización
- Sensibilidad al ruido de fase (mitigable en parte utilizando PSK diferencial)
- Alto nivel de Peak-to-Average Power Ratio (PAPR), lo que dificulta el diseño de los amplificadores RF

OFDM

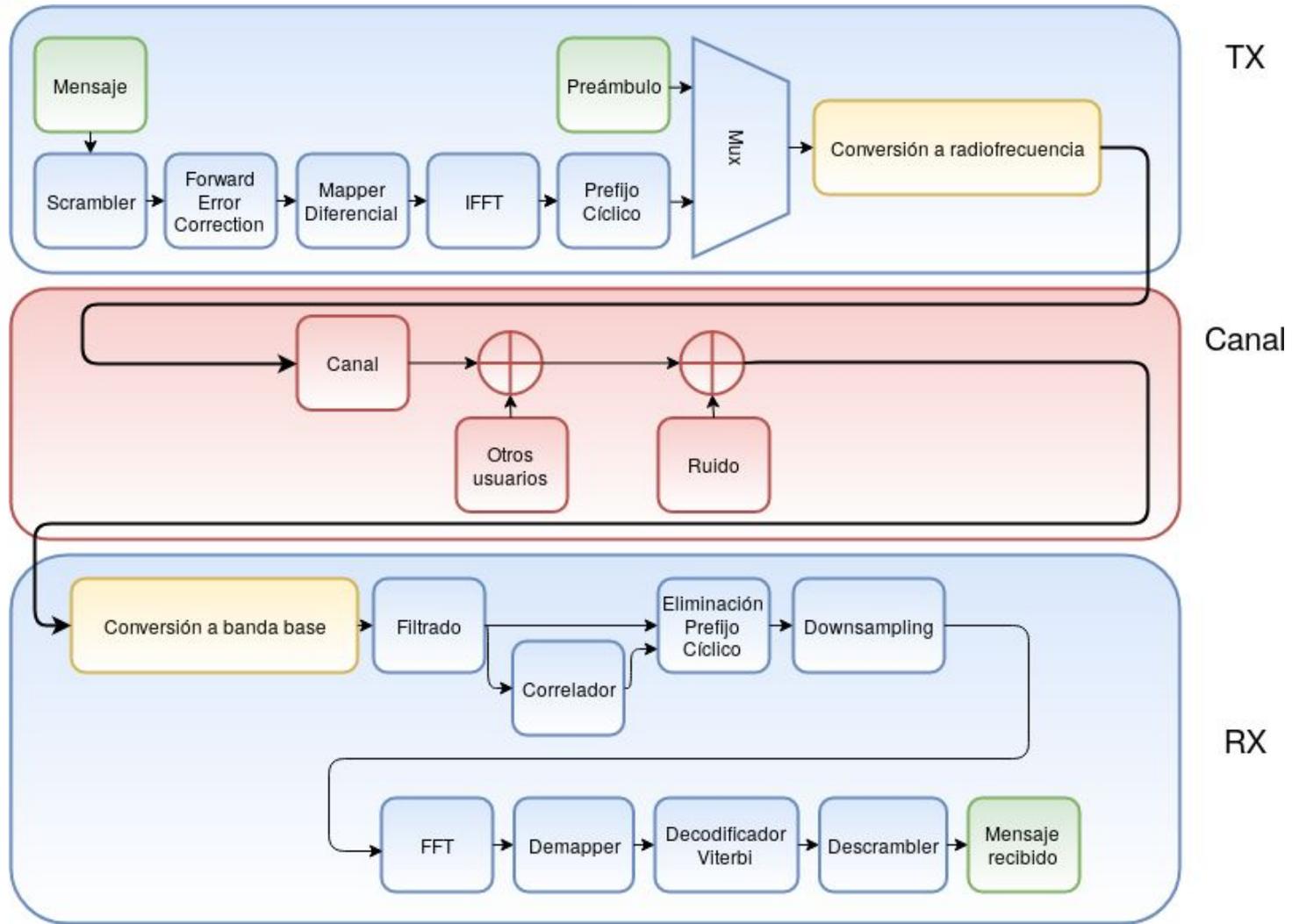
Algunos estándares que utilizan OFDM:



Contenido

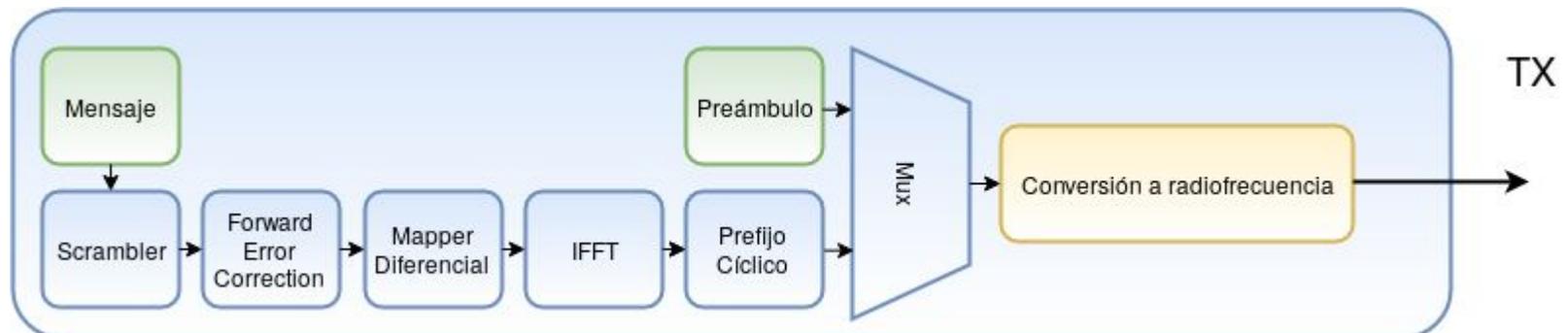
- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Trabajo de curso



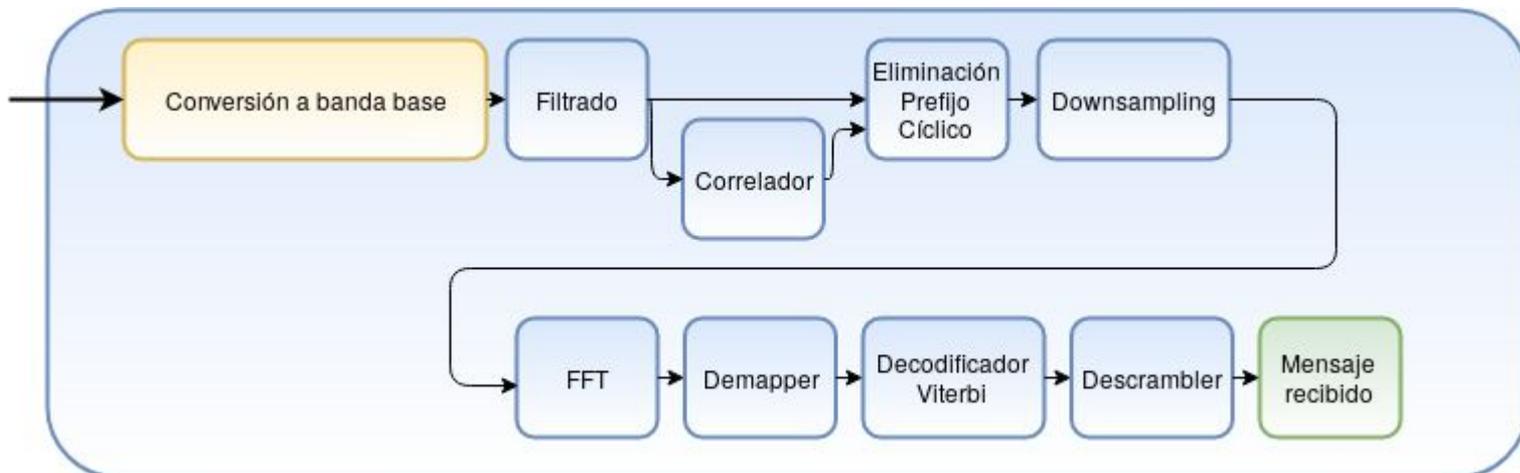
Transmisor

- Scrambler: evitar secuencias largas de 1s y 0s
- FEC: insertar código de corrección de errores
- Mapper: convertir bits a portadoras
- iFFT: convertir portadoras a dominio del tiempo
- Prefijo Cíclico: evitar la interferencia entre símbolos (ISI: Inter-Symbol Interference)
- Preámbulo: señalar el principio del símbolo para poder sincronizarnos en el receptor



Receptor

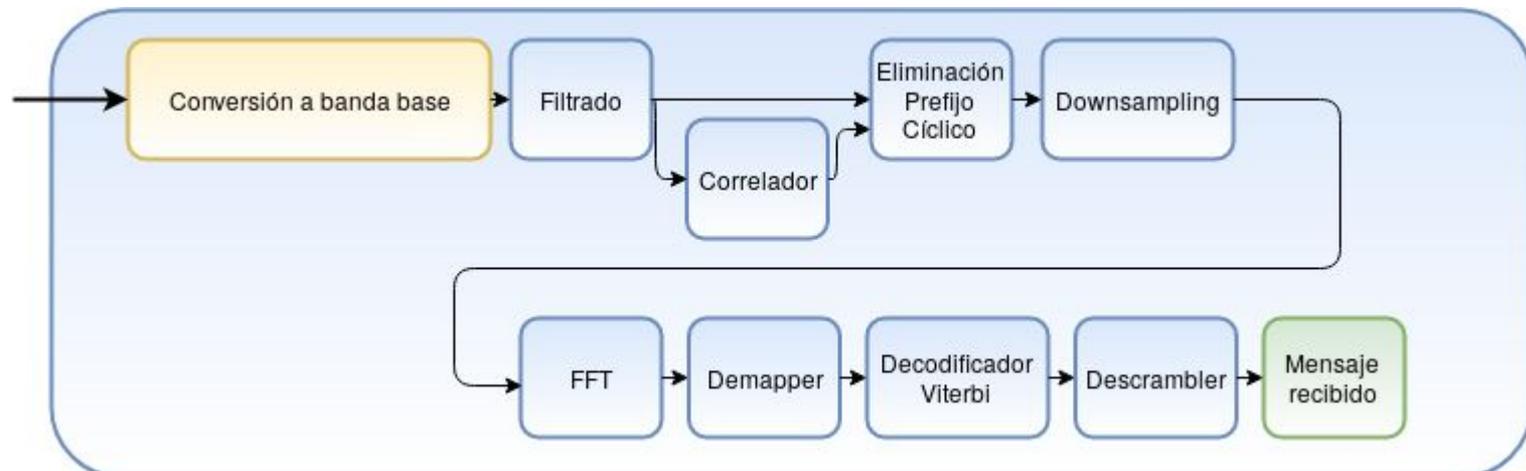
- Filtrado: eliminar ruido fuera de banda para evitar aliasing tras el downsampling
- Correlador: detección del preámbulo (sincronización)
- Eliminación prefijo cíclico: eliminar las primeras muestras
- Downsampling: reducir la tasa de datos



RX

Receptor

- FFT: Conversión tiempo a frecuencia
- Demapper: conversión portadoras -> bits
- Decodificador Viterbi: corrección de errores
- Descrambler: deshacer la pseudo-aleatorización introducida por el scrambler



RX

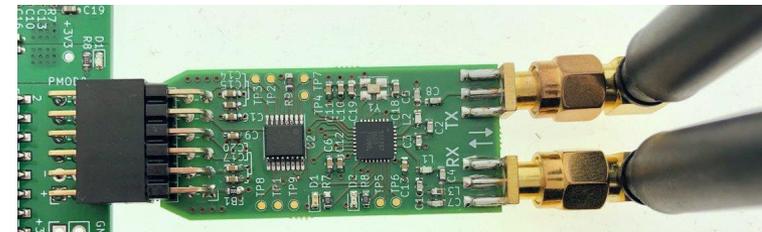
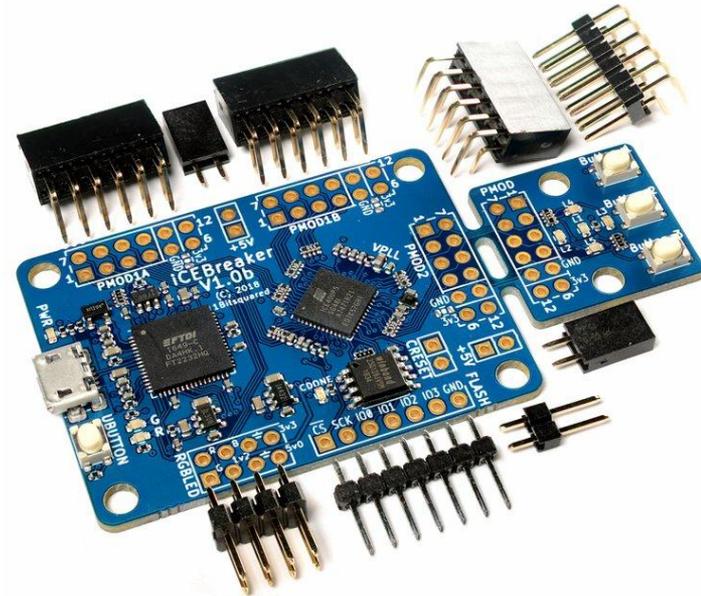
Trabajo de curso

Procesado en banda base: tarjeta IceBreaker (FPGA Lattice iCE40UP5k)

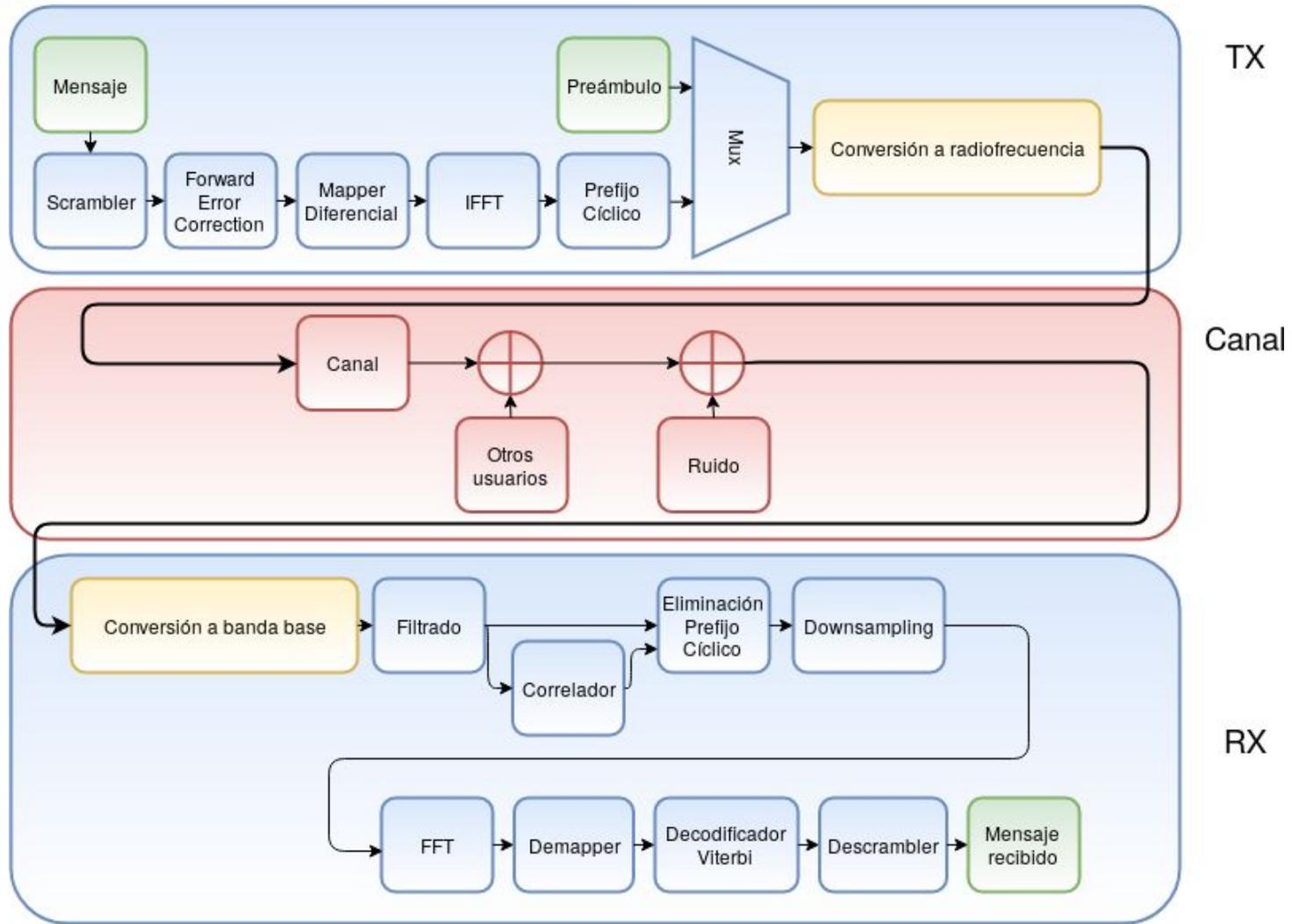
- 5280 celdas lógicas
- 8 bloques DSP
- Más de 1 mbit de memoria interna

Conversión a radiofrecuencia y banda base: SDR PMOD SX1257

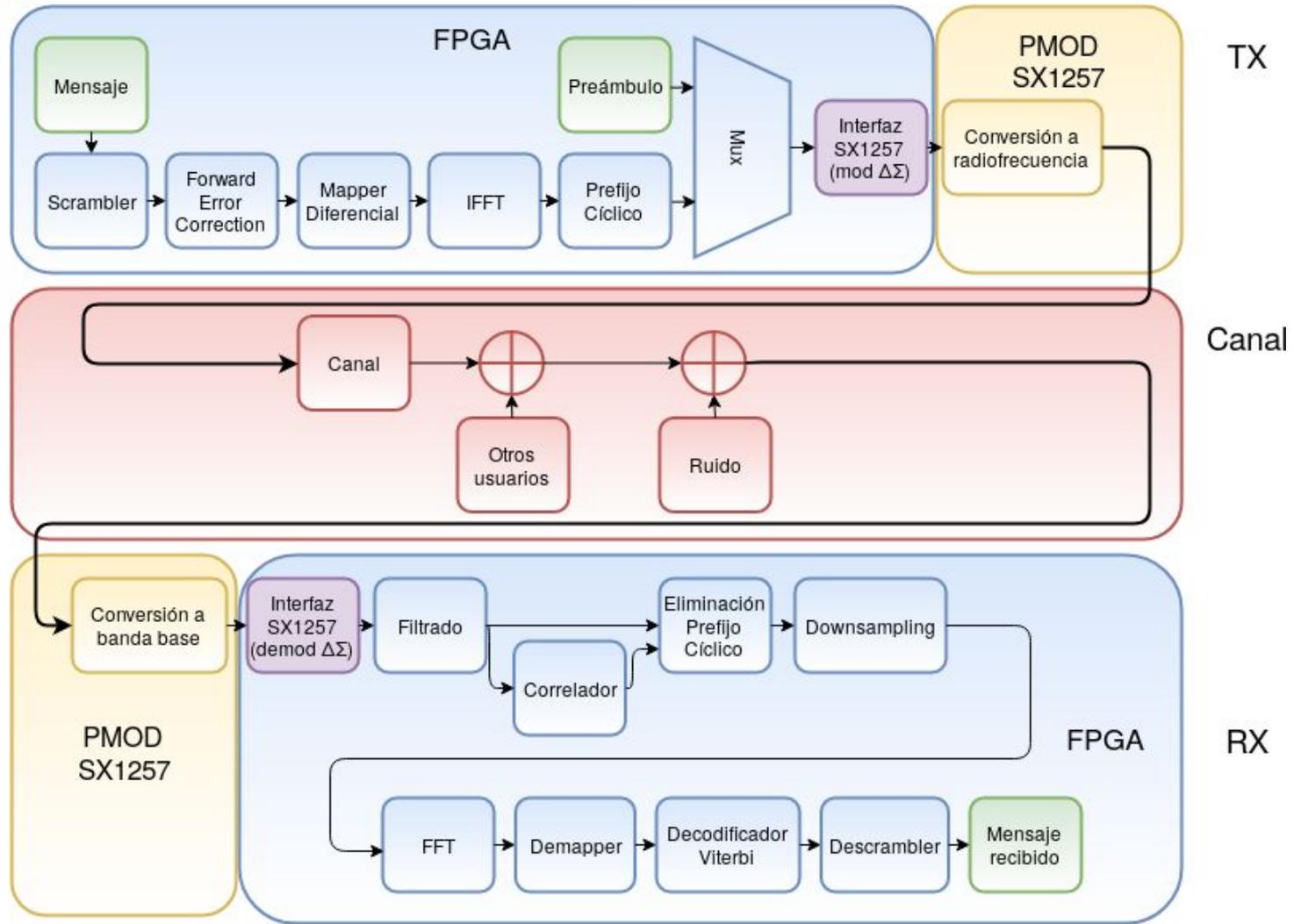
- Transceptor zero-IF full-duplex
- Banda ISM (Industrial, Scientific and Medical), 862-960 MHz, sin licencia
- Interfaz 1-bit I, 1-bit Q, modulado $\Sigma\Delta$
- Configuración por SPI



Trabajo de curso



Trabajo de curso

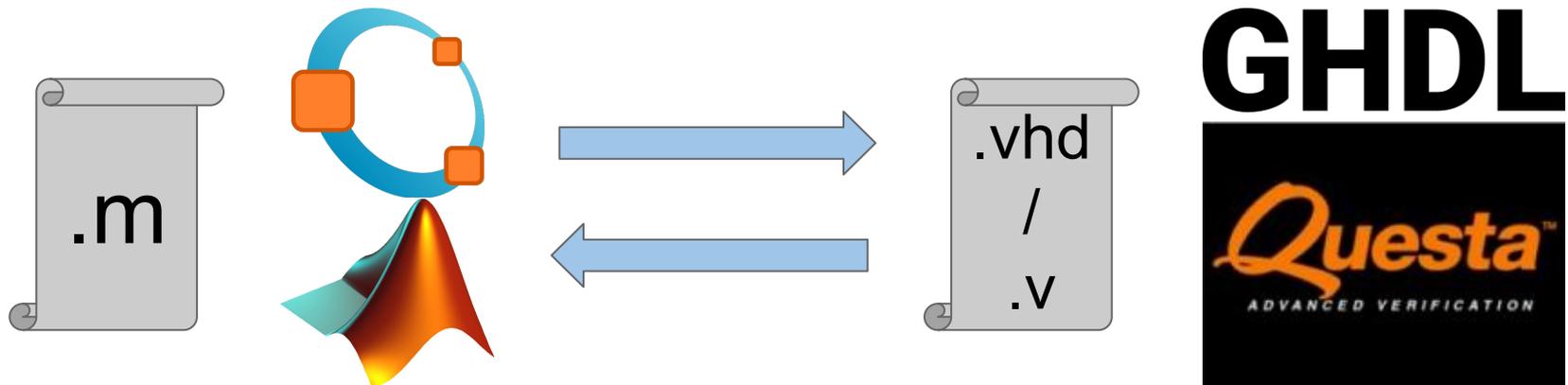


Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Metodología

1. Especificaciones
2. Modelo matlab/octave
3. Cuantización / anchura de los buses
4. Implementación HDL
5. Verificación
6. Optimización (iterando sobre 4-5)



Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Definición

Latencia: tiempo entre que entra el dato y sale procesado

(tarda M ciclos en cruzar el pipeline completo)

Throughput: cada cuántos ciclos sale un dato nuevo

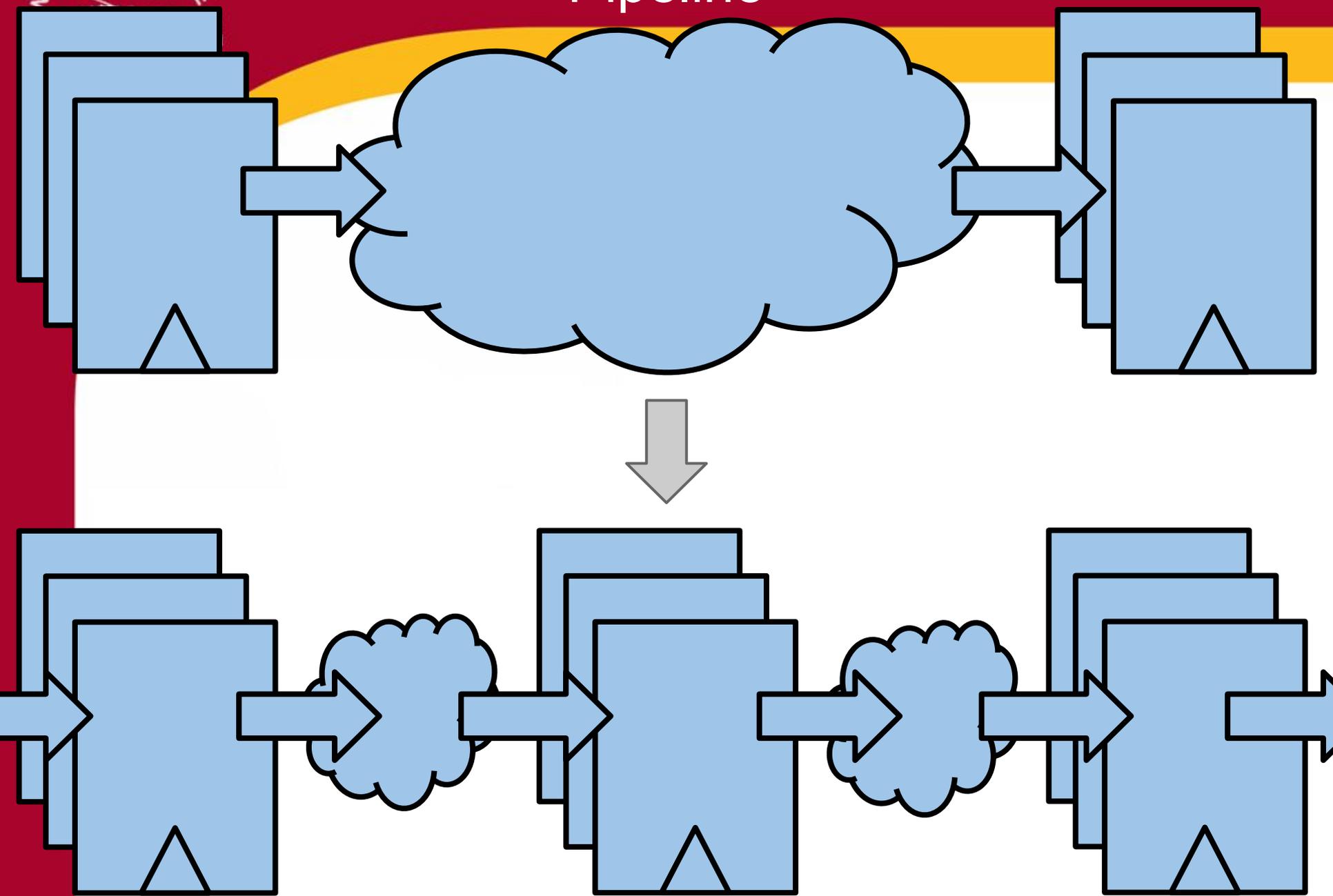
(un dato cada N ciclos)

Se pueden medir en ciclos o en tiempo

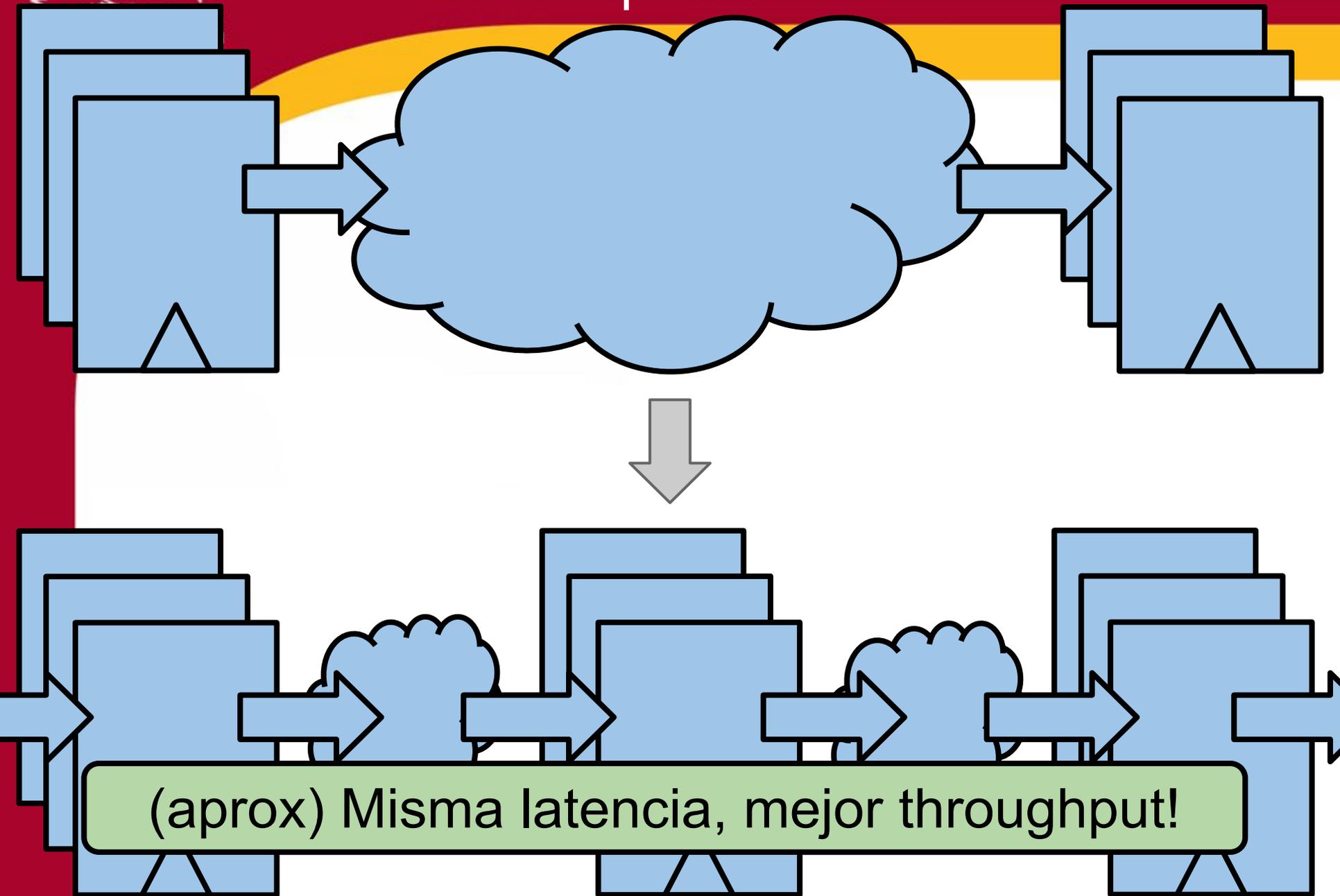
Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- **Pipeline**
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Pipeline



Pipeline



(aprox) Misma latencia, mejor throughput!

Técnica muy utilizada en diseño digital

Por ejemplo en microprocesadores,
si para cada instrucción tenemos que hacer:

- Fetch
- Decode
- Execute
- Write-back

Clock cycle

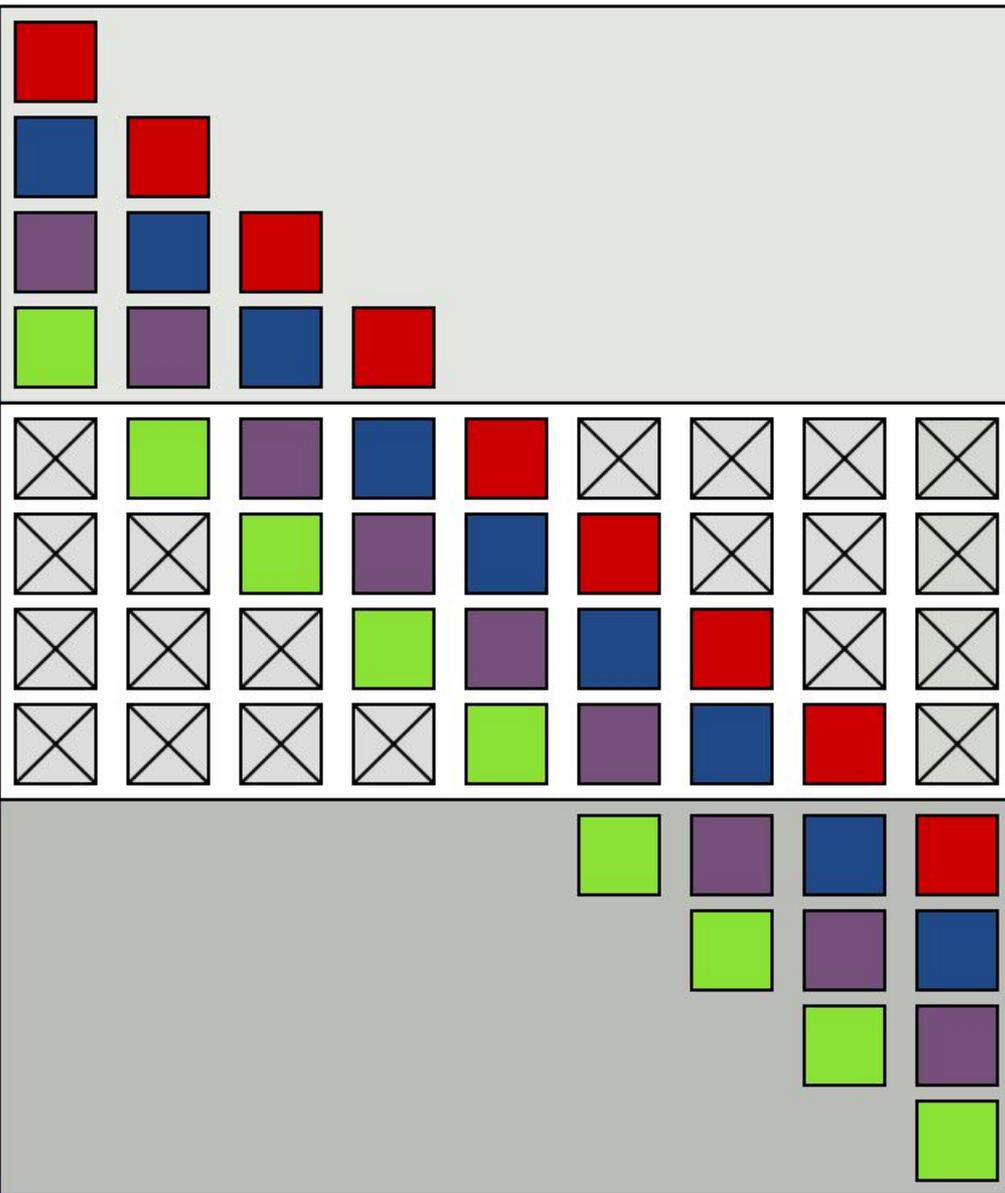
0 1 2 3 4 5 6 7 8

Waiting instructions

Pipeline

- Stage 1: Fetch
- Stage 2: Decode
- Stage 3: Execute
- Stage 4: Write-back

Completed instructions



Procesado de señal

- Dividir el sistema en etapas
- Cada etapa procesa un dato
- Cuando el pipeline está lleno, todas las etapas están procesando un dato a la vez
- La latencia se mantiene o incluso empeora
- El throughput mejora
- En sistemas de comunicaciones el throughput es más crítico
 - Añadir varios ciclos de latencia a cientos de MHz no suele tener efectos perceptibles

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Cómo enfocarla? Tiempo-área es un tradeoff

Tiempo:

- Pipeline
- Paralelizar operaciones
- Cuidado con los I/Os
 - Insertar FFs de sincronización para evitar problemas de timing

Área:

- Aprovechar primitivas de la FPGA
- Reutilizar recursos
 - Procesado secuencial
- Cuidado con reset globales

También configurar opciones relevantes en el sintetizador!

Tradeoff

- El pipeline suele ser interesante en cualquier caso, ya queramos optimizar en área o tiempo
- Añadiendo recursos normalmente podemos mejorar los tiempos
- Empeorando los tiempos normalmente podemos aprovechar mejor (reutilizar) los recursos

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesado de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

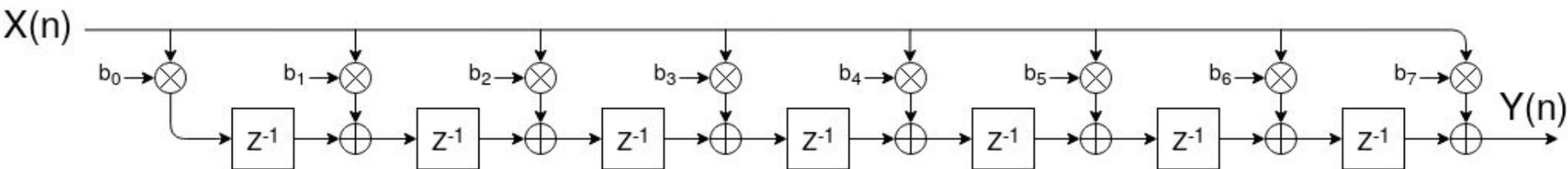
Uso del mismo recurso secuencialmente

- Alto ahorro en área
- Coste en tiempo
- Útil para reducir uso de multiplicadores o bloques DSP
- Muy recomendable en operaciones complejas (divisores)

Filtro FIR

Ej. 8 etapas:
$$y(n) = \sum_{k=0}^7 (x_{n-k} \cdot b_k)$$

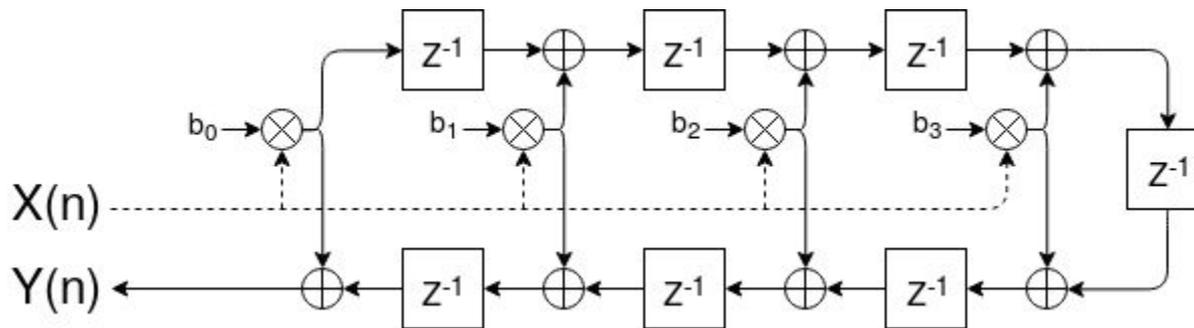
Implementación típica (8 multiplicadores)



Filtro FIR

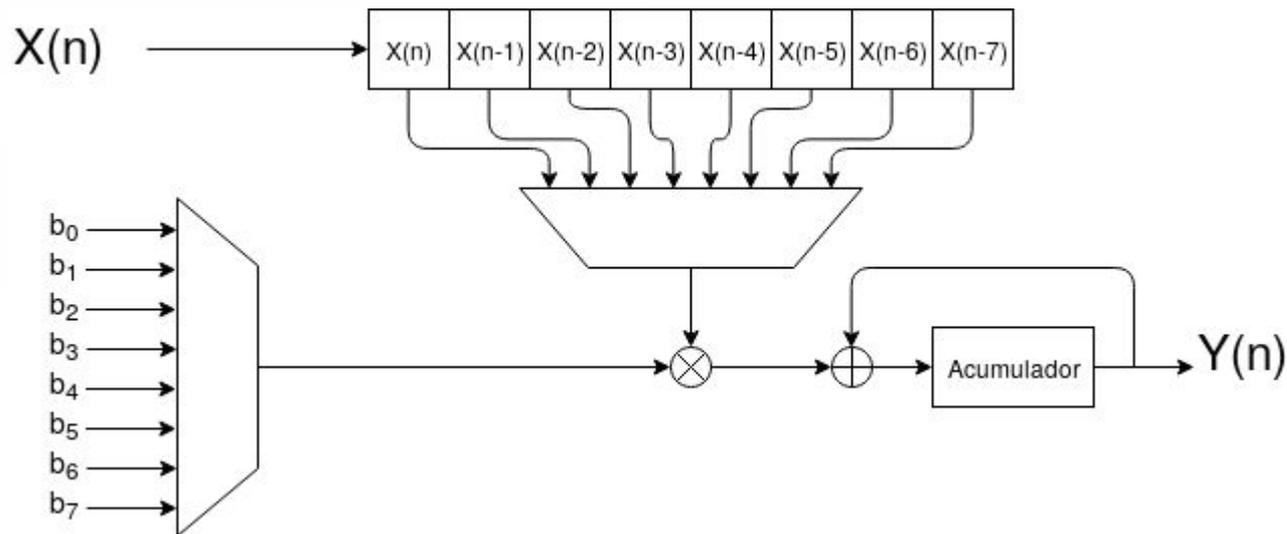
Ej. 8 etapas:
$$y(n) = \sum_{k=0}^7 (x_{n-k} \cdot b_k)$$

Si los coeficientes son simétricos ($b_k = b_{7-k}$):
Implementación plegada (4 multiplicadores)



Filtro FIR

Implementación secuencial



Coeficientes y señal pueden almacenarse en registros o memorias!

D. Gisselquist, [A better filter implementation for slower signals](#)

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Conclusiones

- Comunicaciones digitales ubicuas y muy relevantes en la actualidad
- OFDM de gran interés por el aprovechamiento del espectro
- Trabajo de curso compuesto de bloques de procesado digital de señal
- Metodología de trabajo octave \leftrightarrow HDL
- Pipeline suele ser de interés en procesado de señal
- Podemos diseñar para optimizar en área o tiempo
 - Optimización puede ser iterando / mejorando el diseño

Contenido

- Tendencias en comunicaciones digitales
- Descripción del sistema a implementar
- Diseño de bloques digitales de procesamiento de señal
- Throughput y latencia
- Pipeline
- Optimización en tiempo y área
- Compartición de recursos
- Conclusiones
- Bibliografía

Bibliografía

- Engels, M., Wireless OFDM Systems: How to make them work? Springer, 2002
- Oria Oria, A. C., “Sistemas OFDM: DVB-H y DVB-SH”, en Contribución a los algoritmos de Estimación de canal y Cancelación de ICI en sistemas DVB-H y DVB-SH, Tesis Doctoral. Director: Vicente Baena Lecuyer, dic. de 2009, cap. 3, págs. 51-78
- Bahai, A. R. S., Saltzberg, B. R., Multi-Carrier Digital Communications: Theory and Applications of OFDM. Kluwer Academic Publishers, 2002
- Schiff, M., Introduction to Communication Systems Simulation. Artech House, 2006
- Viswanathan, M., Simulation of Digital Communication Systems using Matlab. Amazon, 2013.
- Woods, R., McAllister, J., Lightbody, G., Yi, Y., FPGA-based Implementation of Signal Processing Systems. Wiley, 2017

Resultados de aprendizaje

- En comunicaciones digitales actuales, ¿qué parte del procesado se hace con electrónica analógica, y qué parte con electrónica digital? ¿Por qué?
- ¿Por qué a los fabricantes de microprocesadores modernos les interesa incrementar el número de etapas de sus pipelines?
- Ventajas e inconvenientes de OFDM
- Bloques funcionales del trabajo de curso: ¿qué papel cumple cada uno?
- ¿Por qué decimos que área y tiempo de procesado son un tradeoff en diseño digital?