



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Diseño de algoritmos de simulación de filtros de alto
rendimiento para
fotónica programable

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

AUTOR/A: Montalvo López, Román

Tutor/a: Capmany Francoy, José

Cotutor/a externo: PEREZ LOPEZ, DANIEL

CURSO ACADÉMICO: 2022/2023

RESUMEN

La fotónica integrada de microondas es un campo que permite generar, procesar y dirigir señales ópticas mediante circuitos integrados. Esta tecnología ha supuesto un gran avance en diversos campos relacionados con las telecomunicaciones, como pueden ser los sistemas de radio sobre fibra, radar o comunicaciones espaciales.

Una gran hito en este campo es la fotónica programable, la cual permite generar una gran cantidad de circuitos fotónicos mediante la configuración por software de una malla, la cual está formada por varias celdas unitarias programables (PUCs). Esto permite una gran flexibilidad y potencia disponiendo de un único circuito integrado.

A pesar de la potencia y versatilidad de esta tecnología, se necesita acceder de un conjunto de filtros de alto rendimiento (HPBs) que permitan, de manera complementaria, generar configuraciones más complejas que habiliten el acceso a operaciones de filtrado y procesado más avanzadas que las disponibles únicamente empleando la malla programable. Estos conceptos forman la base de la tecnología FPPGA (Field Programmable Photonic Gate Arrays) que promete revolucionar los campos de las telecomunicaciones, el cloud computing y el procesamiento de datos.

El objetivo de este trabajo es diseñar algoritmos que permitan simular la respuesta espectral de filtros ubicados en el conjunto de HPBs de las FPPGA de iPronics e ilustrar las utilidades de estos algoritmos en el desarrollo del producto y en la experiencia del usuario de FPPGA.

Palabras clave: Fotónica de microondas, Fotónica integrada, Fotónica programable, Simulador de software, Filtros.

ABSTRACT

Integrated microwave photonics is a field that enables optical signals to be generated, processed and steered by integrated circuits. This technology has been a breakthrough in several fields related to telecommunications, such as radio over fibre, radar or space communications.

A major milestone in this field is programmable photonics, which makes it possible to generate a large number of photonic circuits through the software configuration of a mesh, which is made up of several programmable unit cells (PUCs). This allows for great flexibility and power with a single integrated circuit.

Despite the power and versatility of this technology, access is needed to a set of High Performance Blocks (HPBs) that allow, in a complementary way, to generate more complex configurations that enable access to more advanced filtering and processing operations than those available only using the programmable grid. These concepts form the basis of Field Programmable Photonic Gate Arrays (FPPGA) technology, which promises to revolutionise the fields of telecommunications, cloud computing and data processing.

The aim of this work is to design algorithms to simulate the spectral response of filters located in the HPB array of iPronics FPPGAs and to illustrate the utilities of these algorithms in product development and in the FPPGA user experience.

Keywords: Microwave Photonics, Integrated Photonics, Programmable Photonics, Software Simulator, Filters.

RESUM

La fotónica integrada de microones és un camp que permet generar, processar i dirigir senyals òptics mitjançant circuits integrats. Aquesta tecnologia ha suposat un gran avanç en diversos camps relacionats amb les telecomunicacions, com poden ser els sistemes de ràdio sobre fibra, radar o comunicacions espacials.

Una gran fita en aquest camp és la fotónica programable, la qual permet generar una gran quantitat de circuits fotònics mitjançant la configuració per programari d'una malla, la qual està formada per diverses cel·les unitàries programables (PUCs). Això permet una gran flexibilitat i potència disposant d'un únic circuit integrat.

Malgrat la potència i versatilitat d'aquesta tecnologia, es necessita accedir d'un conjunt de filtres d'alt rendiment (HPBs) que permeten, de manera complementària, generar configuracions més complexes que habiliten l'accés a operacions de filtrat i processament més avançades que les disponibles únicament emprant la malla programable. Aquests conceptes formen la base de la tecnologia FPPGA (Field Programmable Photonic Gate Arrays) que promet revolucionar els camps de les telecomunicacions, el cloud*computing i el processament de dades.

L'objectiu d'aquest treball és dissenyar algorismes que permeten simular la resposta espectral de filtres situats en el conjunt de HPBs de les FPPGA de iPronics i il·lustrar les utilitats d'aquests algorismes en el desenvolupament del producte i en l'experiència de l'usuari de FPPGA.

Paraules clau: Fotónica de microones, Fotónica integrada, Fotónica programable, Simulador de software, Filtres.

ÍNDICE

RESUMEN	I
ABSTRACT	II
RESUM	III
ÍNDICE DE TABLAS	IX
ÍNDICE DE FIGURAS	XII
1. INTRODUCCIÓN	1
1.1. Motivación	2
1.2. Resumen de la metodología empleada	2
1.3. Objetivos	3
1.3.1. Objetivos generales	3
1.3.2. Objetivos de Desarrollo Sostenible	4
1.4. Relación con las asignaturas	4
1.5. Estructura del trabajo	5
2. Revisión del estado del arte	7
2.1. Introducción a la Fotónica	7
2.1.1. Fotónica Integrada en Silicio	7
2.1.2. Fotónica de microondas	9
2.1.3. Fotónica Programable	10
2.2. Field Programmable Photonic Gate Array o FPPGA	12
2.2.1. High-Performance filters	13
2.2.1.1. ROADM	14

3. Elementos fundamentales para el desarrollo del simulador	15
3.1. Modelado matricial de componentes fotónicas	15
3.1.1. Matriz de Scattering	15
3.1.2. Matriz de Transferencia	16
3.2. Elementos principales	18
3.2.1. Unidad básica programable	18
3.2.2. Phase shifters	19
3.2.3. Guías de onda	19
3.3. Filtro a simular: ROADM	19
4. Desarrollo y resultados de los algoritmos	23
4.1. Herramientas utilizadas	23
4.2. Implementación en software de elementos fundamentales	24
4.2.1. Atributos y métodos PUCs	24
4.2.2. Atributos y métodos Phase Shifters	25
4.3. Diseño del interferómetro Mach-Zehnder dual	25
4.3.1. Resultados obtenidos	28
4.4. Diseño del ROADM de 4 canales	31
4.4.1. Resultados obtenidos y configuraciones del ROADM	34
4.5. Capacidades y configuraciones del ROADM	35
4.5.1. Demultiplexación de 2 canales	36
4.5.2. Demultiplexación de 3 canales	38
4.5.3. Multiplexado de 4 canales	40
4.5.4. Cambio de asignación de canales a cada puerto	40
5. Conclusiones y futuras líneas de trabajo	43
5.1. Conclusiones	43
5.2. Propuestas de trabajo futuro	44
A. Anexo I: Códigos empleados para mostrar las respuestas espectrales	47

ÍNDICE DE TABLAS

4.2.1.Atributos fundamentales de la implementación en software de la PUC	24
4.2.2.Métodos fundamentales de la implementación en software de la PUC	25
4.2.3.Atributos fundamentales de la implementación en software de un phase shifter	25
4.2.4.Métodos fundamentales de la implementación en software de un phase shifter	25
4.3.1.Atributos fundamentales de la implementación en software de un MZI dual	26
4.3.2.Métodos fundamentales de la implementación en software de un MZI dual	27
4.4.1.Atributos fundamentales de la implementación en software de un ROADM de 4 canales	32
4.4.2.Métodos fundamentales de la implementación en software de un MZI dual	33
4.4.3.Fases aplicadas para la configuración de demultiplexación de 4 canales	35
4.5.1.Fases aplicadas para la configuración de demultiplexación de 2 canales	37
4.5.2.Fases aplicadas para la configuración de demultiplexación de 3 canales	39

ÍNDICE DE FIGURAS

2.1.1. Proceso de fabricación de un circuito integrado. Fuente [5]	8
2.1.2. Espectro electromagnético. Permite ubicar las bandas de microondas y fotónica	9
2.1.3. Diagrama de bloques de un sistema de RF y su versión con fotónica	10
2.1.4. Esquema y comparación de la evolución de la fotónica y la electrónica. Fuente [8]	10
2.1.5. Tipos de estructuras para mallas fotónicas: a) rectangular, b) hexagonal y c) triangular. Fuente [8]	11
2.2.1. Software de la FPPGA, iPronics Smartlight	12
2.2.2. Estructura del procesador fotónico	13
2.2.3. Diagrama que muestra la ubicación de los los filtros de alto rendimiento en el procesador. Fuente [13]	13
3.1.1. Sistema óptico de dos puertos representado como caja negra. Fuente: Elaboración propia	16
3.1.2. Matriz de transferencia de dos elementos en cascada. T_i indica la matriz de transferencia asociada a un dispositivo. Fuente: Elaboración propia.	17
3.2.1. Esquema de la PUC basada en un MZI. Fuente: Elaboración propia	18
3.3.1. Ilustración de una red metro en anillo que emplea ROADM. Fuente [6].	20
3.3.2. Esquema del funcionamiento de un ROADM. Este ROADM tiene 7 canales, está sacando de la señal la λ_1 y añadiendo λ_3 . Fuente: Elaboración propia	20
3.3.3. Esquema del MZI Dual y respuesta espectral. Fuente: [12]	20
3.3.4. Esquema de bloques del ROADM de iPronics. Fuente: Elaboración propia	21
4.3.1. Esquema del MZI dual a modo de sistema de PUCs y Phase shifters interconectados	26
4.3.2. Esquema de equivalencia de MZI dual a conjunto de elementos en cascada, y su representación en etapas	27
4.3.3. Resultados del simulador de MZI dual	29

4.3.4.Comparativa de los simuladores de MZI duales con distintas diferencias de longitud	29
4.4.1.Estructura de etapas para el simulador del ROADM	31
4.4.2.Ejemplo de matriz que simula la etapa 1 del ROADM	32
4.4.3.Respuesta del ROADM por pares en configuración de demultiplexado de 4 canales	34
4.4.4.Respuesta del ROADM total en configuración de demultiplexado de 4 canales . .	35
4.5.1.Respuesta del ROADM por pares en configuración de demultiplexado de 2 canales	36
4.5.2.Respuesta del ROADM en el output en configuración de demultiplexado de 2 canales	36
4.5.3.Respuesta del ROADM total en configuración de demultiplexado de 2 canales . .	37
4.5.4.Respuesta del ROADM por pares en configuración de demultiplexado de 3 canales	38
4.5.5.Respuesta del ROADM en el output en configuración de demultiplexado de 3 canales	38
4.5.6.Respuesta del ROADM total en configuración de demultiplexado de 3 canales . .	39
4.5.7.Respuesta del ROADM total en configuración de multiplexado de 4 canales . . .	40
4.5.8.Configuración inicial asignación de canales en el ROADM	40
4.5.9.Primer cambio de asignación de canales en el ROADM	41
4.5.10.Segundo cambio de asignación de canales en el ROADM	41

Capítulo 1

INTRODUCCIÓN

Desde su origen en el Siglo XX, tras el desarrollo de la teoría sobre la emisión estimulada de luz desarrollada por Albert Einstein, que derivó en la invención del láser en 1960, la fotónica integrada ha surgido como un campo prometedor que apunta a revolucionar el panorama tecnológico actual. Debido al incremento de la demanda de tráfico de datos más rápidos y seguros, la fotónica destaca gracias a sus elevados anchos de banda, baja latencia y elevadas velocidades de transmisión en comparación con alternativas tradicionales. Estos aspectos la vuelven una tecnología ideal en áreas como las telecomunicaciones, el sensado o la computación.

Gracias a la capacidad de manipular fotones a una escala compacta, para transmitir y modificar señales ópticas a través de un circuito, la fotónica integrada permite la alta integración de múltiples componentes fotónicos en un único chip, permitiendo una eficiencia y funcionalidad mejoradas acompañada de una reducción en los costes de fabricación.

Un paradigma prometedor dentro de la fotónica es la fotónica programable. Esta rama de la fotónica consigue, a partir de un conjunto de elementos fotónicos básicos organizados de manera inteligente, otorgar cierto grado de reconfigurabilidad al chip para obtener una mayor flexibilidad y adaptabilidad, en comparación con los circuitos fotónicos diseñados para una única aplicación o ASPIC (Application Specific Photonic Integrated Circuits). Esta capacidad de reconfiguración permite controlar dinámicamente la propagación de la luz a través del chip, para optimizarlo y ajustarlo a las necesidades de cada usuario.

Investigadores de la Universitat Politècnica de València han desarrollado el primer procesador fotónico programable de propósito general, llamado FPPGA (Field Programmable Photonic Gate Array), y han desarrollado una empresa spin-off de la misma universidad en torno a esta idea para su comercialización, llamada iPronics. Esta tecnología combina la potencia de la fotónica programable con un preciso sistema de electrónica de control, encargado de enviar las señales adecuadas a las componentes fotónicas para poder reconfigurarlas. Gracias a su software nativo, llamado iPronics Smartlight, la FPPGA ofrece una gran capacidad de programación fotónica, desde programación sencilla a alto nivel hasta configuración individual de las corrientes aplicadas a cada componentes del circuito, además de incluir opciones de simulación previas a la implementación del circuito programado en el propio hardware. Esto convierte a la FPPGA en una herramienta potente, versátil y manejable tanto para profesionales de la fotónica como usuarios con conocimientos básicos del campo.

Ya que la fotónica programable es un campo emergente en la época actual, los simuladores toman un papel fundamental para un rápido desarrollo y avance de esta tecnología. Al igual que en electrónica, un simulador permite visualizar y entender el comportamiento de un circuito que

todavía no ha sido diseñado a través de un programa de software que implemente el algoritmo de simulación. Esto implica una gran ventaja para el desarrollo de chips porque permite reducir la cantidad de prototipos diseñados para hacer pruebas, y conlleva a una gran reducción de tiempo y dinero debido al elevado coste de fabricar en esta tecnología hoy en día.

Debido a la complejidad de los filtros de alto rendimiento incorporados en la FPPGA de iPronics se ha decidido desarrollar algoritmos de simulación que permitan estudiar la respuesta de uno de estos filtros programables, el ROADM, además de evaluar posibles configuraciones de dicho filtro utilizando el simulador desarrollado, y así poder mostrar posibles aplicaciones de esta tecnología.

1.1. Motivación

Este trabajo se realiza a modo de Trabajo de Final de Grado para el grado en Ingeniería de Tecnologías y Servicios de Telecomunicación de la Universitat Politècnica de València. Surge de la colaboración con la empresa iPronics, Programmable Photonics, una empresa spin-off de la Universitat Politècnica de València que ha desarrollado el primer procesador fotónico de propósito general, y es referente a nivel internacional en el campo de la fotónica programable. Su producto principal, combina hardware electrónico y fotónico con una plataforma de software llamada Smartlight, que permite reconfigurarlo mediante programación. La gran variedad de tecnologías y campos que se aplican a esta tecnología permitirá desarrollar y emplear los conocimientos adquiridos en el grado de Ingeniería de Tecnologías y Servicios de Telecomunicación.

En este documento se desarrolla una serie de algoritmos de simulación de filtros fotónicos que se ubicarán en los próximos modelos de FPPGA de iPronics, en concreto de un ROADM ubicado en la sección de filtros de alto rendimiento del chip y las componentes que lo integran. Un ROADM es un filtro avanzado que está tomando gran relevancia en el campo de las redes ópticas pasivas basadas en multiplexación por longitud de onda (WDM-PON) para sus aplicaciones en redes metro y backhaul 5G. Esto permitirá poder simular el comportamiento de la estructura y probar distintas configuraciones del filtro en la FPPGA, además de contribuir al desarrollo de métodos de caracterización y calibración de las componentes del filtro en hardware.

En este contexto, la motivación principal para realizar este trabajo nace de la posibilidad de combinar los conocimientos adquiridos durante los estudios de grado con la innovación tecnológica de iPronics para mejorar el campo de las telecomunicaciones, la cuántica y la computación, a la vez que se realiza investigación y desarrollo en un campo novedoso y con mucho potencial, dentro de una empresa tecnológica con perfiles internacionales y de la mano de profesionales altamente capacitados del sector.

Una segunda motivación es demostrar que el valor que aporta un ingeniero de telecomunicaciones comprende una gran variedad de aspectos, y que la formación que aporta la universidad permite crear perfiles tecnológicos multidisciplinares que pueden generar un gran impacto en multitud de campos, como puede ser el de la fotónica.

1.2. Resumen de la metodología empleada

Para desarrollar los algoritmos explicados en el trabajo se ha empleado la metodología de trabajo propia del departamento de software de la empresa, y ha sido desarrollado durante el periodo de prácticas y trabajo en esta empresa en el curso 2022-2023.

La metodología empleada consistió principalmente en un periodo de aprendizaje en programación y metodologías ágiles de desarrollo de software, entre los que se encuentra el desarrollo de los conocimientos del lenguaje de programación Python y de la interfaz de software de las FPPGA, un periodo de aprendizaje del hardware de la empresa, los elementos que lo conforman y de desarrollo de mis conocimientos en comunicaciones ópticas y fotónica, y finalmente la propia integración en el flujo de trabajo del departamento de software. Esto llevó a un desarrollo continuo y constante de las tareas asociadas a este Trabajo de Final de Grado aplicando la metodología de desarrollo de software Scrum, un método utilizado por la mayor parte de las grandes compañías tecnológicas para un desarrollo eficaz y sostenible de código a largo plazo.

1. Trabajo previo:

- Desarrollo de conocimientos en el lenguaje de programación Python y en metodologías ágiles de desarrollo de software.
- Introducción a la interfaz de desarrollo de software para fotónica programable, iPro-nics Smartlight.
- Introducción al dispositivo de la empresa, su funcionamiento, elementos principales y sus campos de aplicación.
- Revisión bibliográfica y desarrollo de conocimientos en fotónica integrada, fotónica de microondas y fotónica programable, centrando el enfoque en las unidades básicas programables que componen la FPPGA y su comportamiento.

2. Metodología de desarrollo:

- Revisión bibliográfica y estudio del filtro ROADM como tecnología, sus componentes y su implementación en el hardware de la empresa.
- Establecimiento de los objetivos y el alcance del Trabajo de Final de Grado.
- Desglose de las tareas de desarrollo de software para adaptarse a la metodología ágil de desarrollo del departamento.
- Desarrollo del algoritmo de simulación, presentación de sus resultados y revisión continua por parte de los integrantes del departamento.

3. Escritura de la memoria y revisión del trabajo realizado.

1.3. Objetivos

En esta sección servirá para enunciar los objetivos que se pretenden cumplir con el desarrollo del Trabajo de Final de Grado.

1.3.1. Objetivos generales

Los objetivos principales de este trabajo, que servirán como referencia a las conclusiones enunciadas en el capítulo 5, son los siguientes.

- Presentar la importancia de la fotónica integrada como tecnología clave para el desarrollo tecnológico actual, destacando el papel de la fotónica integrada programable y de la FPPGA como su exponente principal.

- Analizar cada uno de los conceptos teóricos y componentes que intervendrán en el desarrollo del simulador del filtro ROADM, resaltando la importancia de este elemento para las telecomunicaciones en la actualidad
- Elaborar un algoritmo de simulación eficaz y eficiente para el ROADM que se ubicará en la FPPGA de iPronics en próximos modelos.
- Obtener y evaluar los resultados de la simulación del filtro, estableciendo distintas configuraciones aplicables al filtro y mostrando los resultados de dichas aplicaciones.
- Emplear y combinar los conocimientos adquiridos a lo largo del grado en distintas áreas como pueden ser las comunicaciones ópticas, el diseño y análisis de filtros o la programación.
- Aprender sobre el trabajo en una empresa tecnológica puntera valenciana de la mano de profesionales del sector, en un campo con elevado potencial que ponga a prueba los conocimientos y habilidades adquiridas.
- Mostrar las conclusiones obtenidas durante el desarrollo del trabajo y presentar posibles líneas de investigación relacionadas con los resultados del trabajo.

1.3.2. Objetivos de Desarrollo Sostenible

La fotónica tiene una gran relación con los Objetivos de Desarrollo Sostenible (ODS), ya que forma parte de las tecnologías clave habilitadoras, conocidas comúnmente como *key enabling technologies* o KETs, establecidas por la Unión Europea. Las KETs son un conjunto de tecnologías esenciales para fomentar la innovación, la competitividad y la sostenibilidad en el panorama tecnológico e industrial europeo. Principalmente se incide en los siguientes ODS:

- ODS 8: Trabajo decente y crecimiento económico: La Comunidad Valenciana es un foco de desarrollo de fotónica muy importante en España. Mediante la creación de empresas y tecnologías fotónicas se está generando una gran cantidad de empleos de calidad en tecnologías punteras para perfiles universitarios cualificados, dentro de las fronteras del país.
- ODS 9: Industria, Innovación e Infraestructura: Las empresas de fotónica es campo con una gran proyección a futuro y que afecta a sectores de importancia como las telecomunicaciones o la computación. Desarrollar innovación en este campo es importante para impulsar la competitividad valenciana y española en sectores tecnológicos punteros como puede ser la fotónica.
- ODS 13: Acción por el clima: La fotónica como tecnología se está introduciendo a un ritmo vertiginoso en el campo de la computación y las comunicaciones, para el desarrollo de procesadores como la FPPGA. Al tener un consumo energético inferior al de sus equivalentes electrónicos, la fotónica es una alternativa más sostenible que permite reducir elevadamente el consumo energético derivado de su uso.

1.4. Relación con las asignaturas

Este trabajo sirve como culminación de los estudios del grado de Ingeniería de Tecnologías y Servicios de Telecomunicación, por ello es importante mencionar las asignaturas que tienen

relación con el trabajo.

Fundamentos de Comunicaciones Ópticas. Estudio de aspectos y elementos fundamentales de las comunicaciones ópticas, así como introducción a la fotónica.

Señales y sistemas. Introducción al modelado de sistemas mediante funciones de transferencia, así como el modelado de sistemas complejos a partir de componentes elementales unidas en serie, en paralelo y en cascada.

Lineas de transmisión. Desarrollo de conocimientos en circuitos de microondas. Se aprendieron los conceptos de matriz de dispersión o scattering y matriz de transferencia para modelar el comportamiento de componentes de microondas. Este modelado también se aplica a elementos fotónicos.

Tecnologías y Sistemas en Redes de Acceso. Estudio de las redes ópticas pasivas, destacando las tecnologías de redes WDM-PON para FTTH, redes metro y backhaul 5G, donde el filtro fotónico a simular tiene grandes aplicaciones.

Programación. Aprendizaje en lenguaje de programación Java y programación orientada a objetos. Estos conocimientos se han podido extrapolar al lenguaje Python, que es la principal herramienta empleada para el desarrollo del trabajo.

1.5. Estructura del trabajo

Esta sección resume la estructura en la que se desarrolla el documento.

El **Capítulo 1** sirve como introducción al trabajo y a su campo de aplicación, los objetivos establecidos, la metodología empleada para su desarrollo y la estructuración del documento.

El **Capítulo 2** sirve como revisión del estado del arte de la fotónica integrada, centrándose principalmente en la fotónica programable. También se desarrolla la FPPGA como ejemplo de éxito de primer procesador fotónico programable de propósito general y los elementos que la componen, centrando el enfoque en la sección de filtros de alto rendimiento donde se localiza el ROADM a simular.

El **Capítulo 3** presenta los elementos principales que contribuirán al desarrollo del algoritmo de simulación del filtro, que son los conceptos de matriz de scattering y de transferencia, empleados con frecuencia para el modelado de componentes fotónicas y de microondas, los elementos básicos programables que se utilizan en la FPPGA, y el ROADM implementado en el dispositivo.

El **Capítulo 4** muestra todo el desarrollo de los algoritmos de simulación de los filtros y las herramientas de software empleadas para su desarrollo. También se muestran los resultados de los simuladores y distintas configuraciones aplicables al ROADM para mostrar sus capacidades.

El **Capítulo 5** resume las conclusiones obtenidas durante el trabajo y describe propuestas futuras de trabajo en este campo.

Capítulo 2

Revisión del estado del arte

Este capítulo servirá como introducción a los conceptos principales sobre los que se fundamenta este Trabajo de Final de Grado. En primer lugar se introducirá el campo de la fotónica integrada, destacando principalmente el papel de la fotónica integrada programable. Posteriormente se presentará la FPPGA como tecnología principal dentro del campo de la fotónica programable, así como una descripción de sus elementos principales.

2.1. Introducción a la Fotónica

La fotónica es la ciencia que estudia la generación, detección y propagación de la luz, así como su interacción con la materia [9]. Albert Einstein y Niels Bohr fueron los que establecieron sus fundamentos teóricos, pero no fue hasta el 1960 cuando se desarrollaron los primeros semiconductores de luz simultáneamente en laboratorios del MIT y los laboratorios Bell, dando nacimiento al concepto de láser. A partir de este momento el campo de la fotónica fue avanzando paralelamente al de la electrónica, introduciéndose en una gran cantidad de campos como las telecomunicaciones, la ingeniería cuántica o la computación neuromórfica. Actualmente es considerada por muchos una tecnología fundamental para el desarrollo y la innovación en el siglo XXI [4].

El campo de la fotónica ha ido desarrollando distintas tendencias en cuanto a materiales de fabricación, características y propósito de los circuitos. A continuación se verán las tres tendencias principales que en conjunto sustentan la tecnología sobre la que se ha desarrollado este Trabajo de Final de Grado, que es el *Field Programmable Phonic Gate Array* o FPPGA.

2.1.1. Fotónica Integrada en Silicio

La evolución de la fotónica ha avanzado de manera paralela al desarrollo de la electrónica, siendo esta última un campo totalmente consolidado a día de hoy. Sin embargo, estas tecnologías no han surgido para competir entre ellas, sino para combinarse y emplearse simultáneamente en todo tipo de sistemas, con el objetivo de aprovechar todas las ventajas y potencial que ambas tiene que ofrecer. Debido a la similitud en cuanto a su desarrollo, la fotónica se ha beneficiado de ventajas similares a las que experimentó el campo de la electrónica durante su expansión, entre las que destaca la integración de circuitos en un solo chip.

Un circuito integrado es un circuito eléctrico cuyas componentes, como resistencias y transistores, se disponen a lo largo de una lámina de material semiconductor. Gracias a la invención del transistor en 1947 y el desarrollo del primer circuito integrado en 1958 por la compañía Texas Instruments, esta tecnología ha sido el motor de la sociedad tecnológica que conocemos hoy en día, y el campo de la fotónica ha sabido aprovecharse de ello.

La principal tecnología actual de fabricación de circuitos integrados es la tecnología de semiconductor complementario de óxido metálico, conocida como CMOS. Los sistemas de fabricación centrados en esta tecnología se caracterizan por una alta capacidad de integración que permite incluir una enorme cantidad de componentes en una única oblea de silicio, realizando una producción e integración muy elevada y eficiente [5].

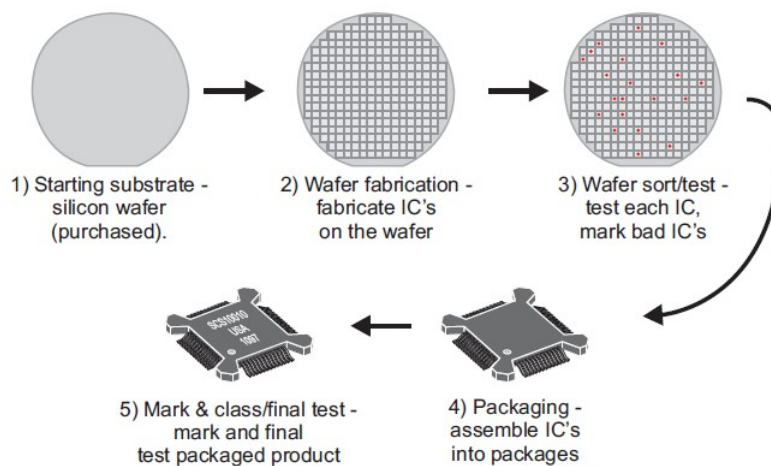


Figura 2.1.1: Proceso de fabricación de un circuito integrado. Fuente [5]

La eficacia de este proceso suscitó en aquellos años la posibilidad de compatibilizar la fotónica con los procesos de fabricación de tecnología CMOS, para poder disponer de procesos de fabricación que permitiesen a la fotónica disponer de la misma capacidad de miniaturización y producción en masa con bajo coste que la electrónica [3].

Gracias al conocimiento del silicio como un material excelente capaz de confinar y guiar la luz a escala submicrométrica, destaca el campo de la Fotónica Integrada de Silicio como principal tendencia para el desarrollo de circuitos fotónicos. La tecnología de fotónica de silicio, también conocida como SOI (Silicon on Insulator) es una tecnología que fabrica componentes en una capa de silicio de un grosor de alrededor de 200 nm ubicada encima de una capa aislante de entre 1 y 3 micras [2]. La principal ventaja que presenta esta tecnología es la compatibilidad con la tecnología de fabricación CMOS y la capacidad de integración de componentes electrónicas y fotónicas, lo que supone un gran potencial para la integración monolítica de componentes electrónicos y fotónicos en un mismo chip.

2.1.2. Fotónica de microondas

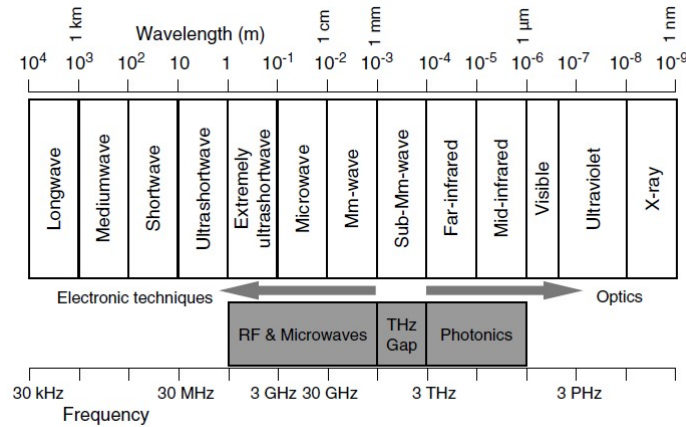


Figura 2.1.2: Espectro electromagnético. Permite ubicar las bandas de microondas y fotónica

Uno de los campos en la que la fotónica ha tenido una gran recepción ha sido la ingeniería de microondas, rama que tiene su mayor impacto en el ámbito de las telecomunicaciones.

A inicios de los años 90 comenzó la integración de las tecnologías ópticas y de microondas, motivadas por una necesidad cada vez mayor de disponer de procesamiento de señal de alta velocidad y alta frecuencia, principalmente en las telecomunicaciones, los sistemas radar o las comunicaciones inalámbricas. Una serie de innovaciones tecnológicas sucedieron simultáneamente que permitieron acelerar el desarrollo e interés en este campo, entre las cuales destaca el desarrollo de la fibra óptica aplicada a las telecomunicaciones, y el nacimiento de la fotónica integrada.

Respecto a la fibra óptica, este mecanismo de guiado de luz permitía incidir en las necesidades crecientes de altas velocidades y ancho de banda, superando a su vez los problemas de los sistemas de comunicación convencionales y permitiendo una transmisión de datos de alta velocidad adecuada para la sociedad de la información. Para adaptarse a las prestaciones que esta tecnología tiene que ofrecer, era necesario integrar las tecnologías previamente empleadas en comunicaciones con componentes capaces de procesar las señales ópticas que viajan a través de la fibra. La fotónica integrada resultó ser el elemento perfecto para realizar esta labor, ya que su compatibilidad con la tecnología CMOS y sus prestaciones permitían una fabricación extensa y aplicación directa en estos sistemas.

De la combinación de estas tecnologías surgió el campo de la fotónica de microondas, la cual se define como la ciencia que combina la ingeniería de microondas y la fotónica para generar, procesar, transmitir y detectar señales de microondas y ópticas empleando ambas tecnologías. Incluye el estudio de los dispositivos fotónicos capaces de procesar las señales de microondas y la aplicación de componentes y técnicas a los sistemas de microondas [10], y surge para integrar los sistemas de comunicaciones ópticas en las telecomunicaciones y sacar el máximo partido a esta tecnología como puede verse en el ejemplo de la figura 2.1.3.

El surgimiento de este campo sirvió para destacar la importancia que tiene la fotónica para las comunicaciones, siendo este sector uno de los principales campos de aplicación de esta tecnología. Es por esto que una gran cantidad de ingenieros de telecomunicación se dedican a día de hoy a investigar e innovar en este campo.

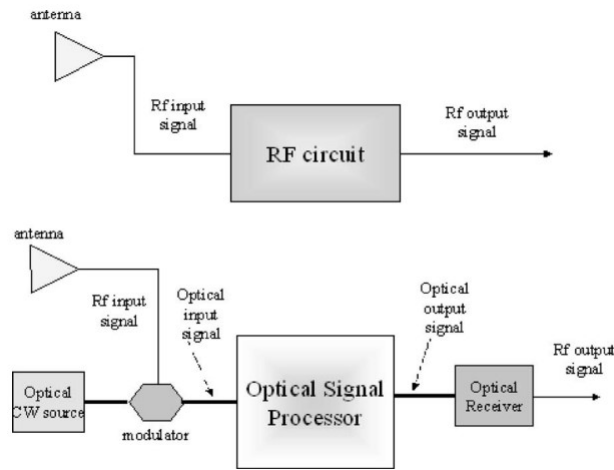


Figura 2.1.3: Diagrama de bloques de un sistema de RF y su versión con fotónica

2.1.3. Fotónica Programable

Los sistemas reconfigurables son sistemas con características definibles mediante software, de manera que un hardware puede ser afinado, modificado o alterado de acuerdo a los propósitos de sus usuarios [7]. Estos sistemas son comúnmente denominados como "soft hardware".

El máximo paradigma de la reconfigurabilidad en la electrónica es la FPGA (Field Programmable Gate Array), que permite disponer de una gran versatilidad integrada en un único sistema programable, empleando lenguajes de descripción de hardware como Verilog o VHDL. Estos dispositivos han tenido una gran recepción debido a las ventajas en miniaturización expuestas por la ley de Moore, la reducción en los costes de fabricación y la cantidad de circuitos configurables en la plataforma.

Ya que el panorama tecnológico actual es altamente cambiante, los sistemas reconfigurables surgen como una alternativa muy prometedora respecto a los dispositivos de aplicación específica o ASIC (*Application Specific Integrated Circuits*) y, aunque en un inicio fueron concebidas para convivir con los ASIC, hoy en día los dispositivos programables los han reemplazado en la mayor parte de las aplicaciones [11].

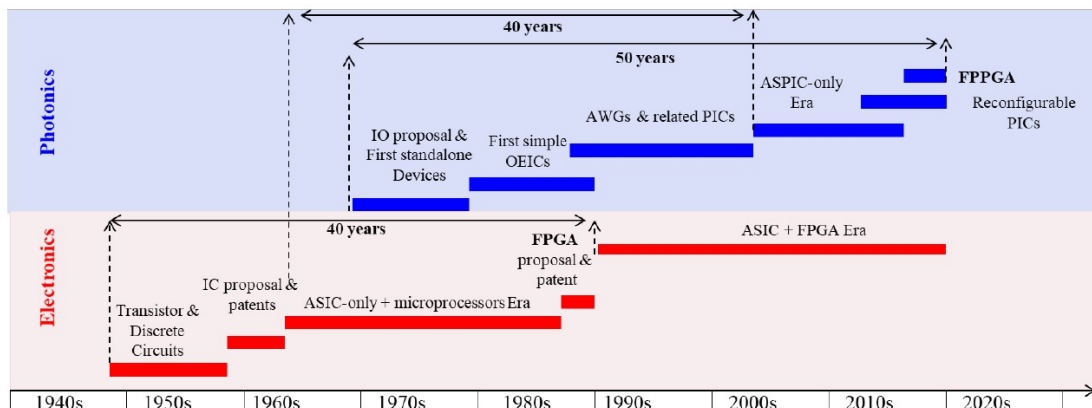


Figura 2.1.4: Esquema y comparación de la evolución de la fotónica y la electrónica. Fuente [8]

Como se puede observar en 2.1.4, históricamente la fotónica ha tenido un desarrollo similar a la electrónica. En los últimos años distintos centros de investigación han comenzado el desarrollo de la fotónica integrada programable o PIP (Programmable Integrated Photonics), que es un nuevo paradigma que pretende diseñar una plataforma óptica reconfigurable mediante software, con multitud de funcionalidades para diversas aplicaciones. Mediante señales eléctricas externas, generadas por equipos de electrónica de control, los chips programables son capaces de redirigir la luz que los atraviesa para realizar operaciones de filtrado, multiplexación, enrutamiento, además de ser capaces de adaptarse a las no idealidades producidas por condiciones ambientales o defectos de fábrica [2].

A día de hoy existen muchos circuitos fotónicos que contienen algún elemento programable, pero el más característico es la malla fotónica. Las mallas fotónicas están formadas por un conjunto de pequeñas unidades programables, llamadas TBUs (Tuneable Basic Unit) que, organizadas y conectadas de forma inteligente, permiten configurar estructuras fotónicas como anillos resonantes o interferómetros que permiten realizar operaciones de procesamiento y filtrado de señal, multiplexado y enrutamiento. Ejemplos de estas mallas se pueden ver en la figura 2.1.5.

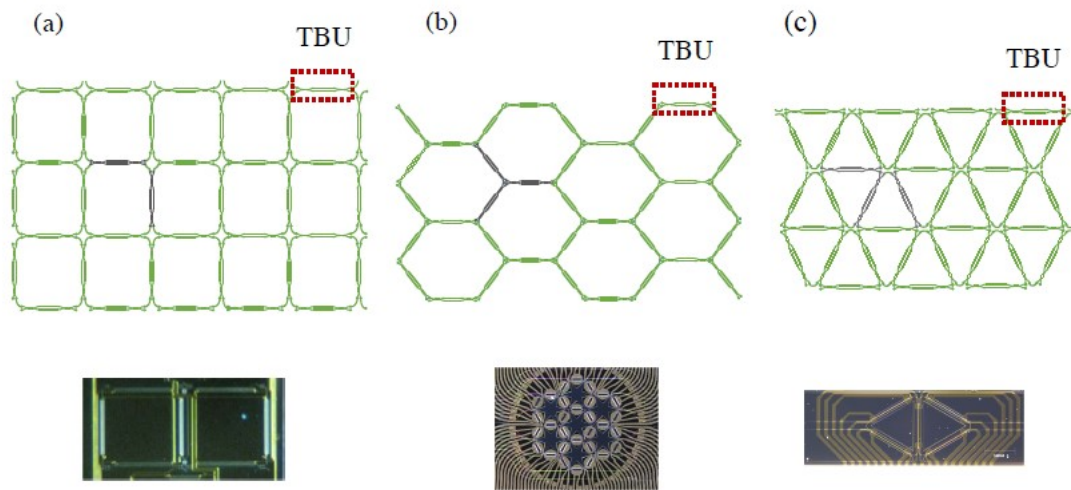


Figura 2.1.5: Tipos de estructuras para mallas fotónicas: a) rectangular, b) hexagonal y c) triangular. Fuente [8]

A día de hoy todavía no está claro qué estructura de malla fotónica es la óptima para maximizar la capacidad de los circuitos, pero actualmente predominan tres modelos: las mallas rectangulares, las mallas triangulares y las mallas hexagonales. En particular las mallas rectangulares han demostrado ser de gran utilidad para computación, y las mallas hexagonales han surgido como modelos de gran versatilidad para comunicaciones y procesamiento de señal, pudiendo adoptar también esquemas similares a la malla rectangular si se configura adecuadamente.

Debido al gran potencial de las mallas fotónicas programables para generar procesadores de alto rendimiento, muchas compañías actuales han dedicado a investigar y desarrollar tecnologías en este campo. De las más destacadas se encuentra *iPronics*, *Programmable Photonics* con su principal tecnología, la FPPGA.

2.2. Field Programmable Photonic Gate Array o FPPGA

Tras el éxito de las FPGA como tecnología, y la aparición de la fotónica programable, en las mentes de muchos investigadores surgió la idea de generar un equivalente fotónico de las FPGAs que no solo consistiese en una malla fotónica programable, sino en un conjunto más sofisticado de elementos que combinaran electrónica y fotónica para generar una tecnología versátil y potente. Entre estas mentes figuran los investigadores del Photonics Research Lab de la UPV que lograron desarrollar el equivalente fotónico de las FPGAs, llamado FPPGA (Field Programmable Photonic Gate Array) como primer procesador fotónico de propósito general. Esto les permitió ir un paso más allá y fundar la compañía spin-off de la UPV, iPronics Programmable Photonics.

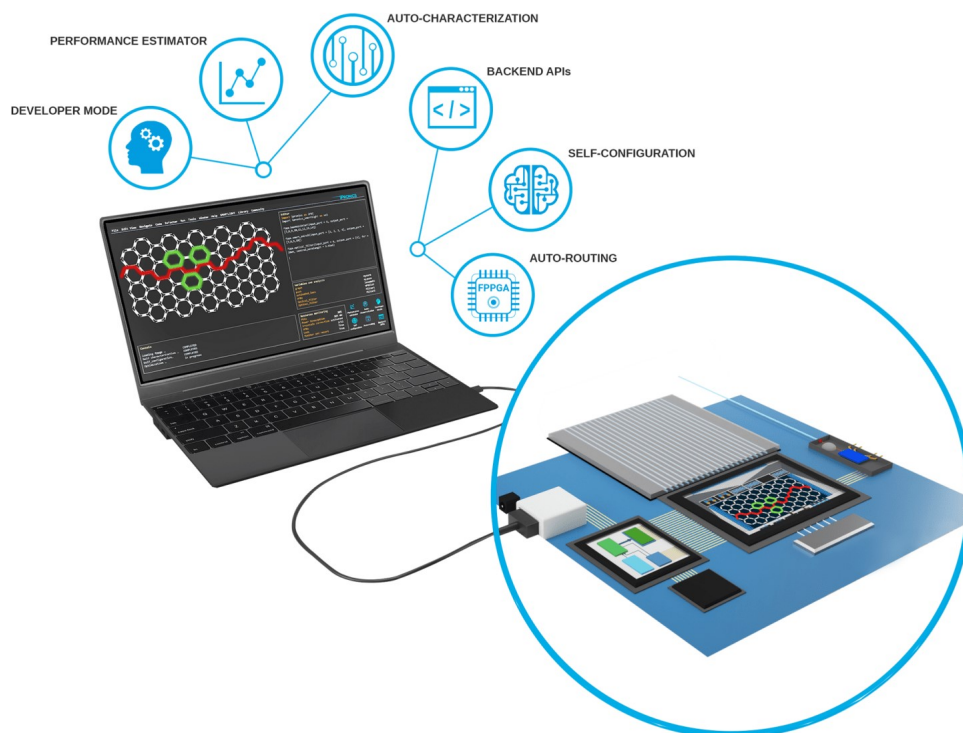


Figura 2.2.1: Software de la FPPGA, iPronics Smartlight

La FPPGA es una tecnología que combina un hardware fotónico controlado mediante equipos electrónicos y un software que orquesta todas las operaciones llamado iPronics Smartlight. Este software basado en Python permite que el usuario pueda configurar una gran cantidad de circuitos distintos de manera sencilla y rápida. Además posee una modalidad de simulación que permite experimentar con diseños de circuitos y comprobar su respuesta antes de configurarlo en el propio hardware. Esta funcionalidad de simulación es muy popular en el diseño de sistemas y circuitos electrónicos y fotónicos, ya que permite realizar pruebas y análisis de los circuitos evitando problemas como los costes temporales de ejecución o los efectos negativos inducidos en el hardware por ruido o interferencia que se producen. Todo esto sin la necesidad de reconfigurar el circuito en la malla fotónica.

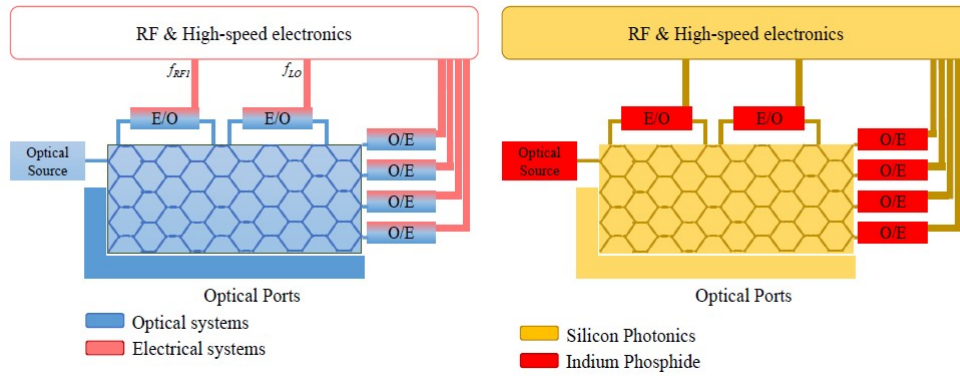


Figura 2.2.2: Estructura del procesador fotónico

Un esquema general del procesador de la FPPGA se muestra en la figura 2.2.2. El tipo de procesador que incorpora es una malla fotónica hexagonal, debido al enfoque de propósito general de la tecnología, la cual es accesible a través de multitud de puertos que la rodean. Esta malla está complementada por componentes eléctricos que permiten generar y controlar todas las corrientes que se aplican a cada una de las componentes programables, y por otras componentes fotónicas como fuentes ópticas, fotodetectores, moduladores o amplificadores.

2.2.1. High-Performance filters

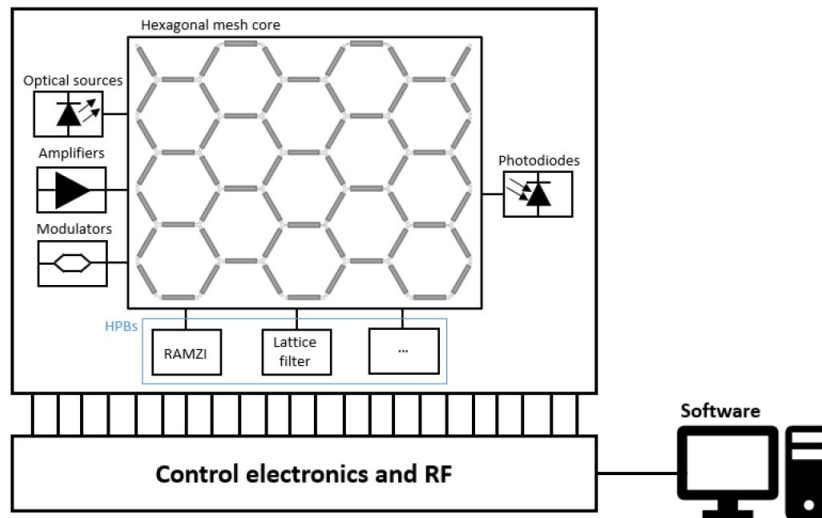


Figura 2.2.3: Diagrama que muestra la ubicación de los los filtros de alto rendimiento en el procesador. Fuente [13]

De manera adicional, con el objetivo de maximizar la versatilidad de la tecnología, se ha incorporado una sección a la malla programable, la cual es accesible por la parte inferior como se muestra en la figura 2.2.3. Esta sección contiene un conjunto de filtros de alto rendimiento o HPBs (High-Performance Blocks) que permite acceder a operaciones de filtrado mucho más potentes que las disponibles únicamente empleando la malla fotónica. En la tecnología FPPGA actual se dispone de un conjunto de HPBs entre los que se comprenden:

- **RAMZI:** Un RAMZI (Ring-assisted Mach-Zehnder Interferometer) es un filtro fotónico que, como indica su nombre, consiste en un MZI al que se le acopla un anillo resonante. Un MZI o Interferómetro Mach-Zehnder es un dispositivo que permite cambiar la fase entre dos haces de luz, y realizar distintos tipos de interferencia entre ellos. Un anillo resonante son estructuras que permiten generar una interferencia destructiva a la frecuencia de interés cuando se cumple una condición denominada acoplo crítico. El RAMZI permite acceder a opciones de filtrado y modulado mucho más complejas que empleando solo un MZI. El HPB de RAMZI consta de 4 RAMZIs conectados en cascada.
- **CROW:** Un CROW (Coupled Resonator Optical Waveguide) es una estructura formada por un conjunto de anillos resonantes acoplados entre ellos en paralelo. Estas guías tienen mucho potencial para actuar como líneas de retardo, para procesamiento de señal y sensado, debido al alto grado de confinamiento de la luz en este campo.
- **Lattice Filters:** Son un filtro multietapa que se elabora concatenando MZIs en cascada. Al conectarlos de esta forma se consigue mejorar la respuesta interferométrica que se tenía empleando solo un MZI y acceder a otras configuraciones más complejas. Existen dos tipos de Lattice Filters en los HPBs de iPronics.

Al igual que la malla fotónica, estos filtros están compuestos de los mismos elementos básicos programables y disponen de sus respectivos simuladores en software que permiten simular su respuesta espectral, sin necesidad de ejecutar el hardware disponible, siendo la respuesta espectral el comportamiento que tiene el filtro para un rango determinado de frecuencias o longitudes de onda. Además, debido a la complejidad de estos elementos en comparación a las configuraciones de la malla, se dispone de una herramienta que permite obtener las respuestas deseadas de los filtros optimizando los parámetros de las componentes programables que los forman. Estas componentes se explicarán en el capítulo 3 de la memoria.

2.2.1.1. ROADM

En las nuevas versiones de los chips fotónicos que están en producción se ha decidido incorporar un nuevo elemento a la suite de HPBs. Este nuevo elemento es el ROADM (Reconfigurable Optical Add-Drop Multiplexer), un filtro sofisticado cuyo uso está creciendo en la actualidad, gracias al interés en las tecnologías de multiplexación por división de longitud de onda (WDM) en sistemas de redes ópticas pasivas, aplicadas a sistemas como las redes metro y el 5G.

Antes de proceder a su incorporación en el chip es necesario evaluar las prestaciones del ROADM que se pretende incorporar, y desarrollar métodos para poder calibrar y caracterizar las componentes del ROADM una vez salido de fábrica. Por todo esto es necesario disponer de un simulador que permita evaluar el comportamiento y la respuesta del filtro, para evaluar su potencial y diseñar los procedimientos de calibración y caracterización, sin disponer del ROADM físico, además de para ser incorporado posteriormente en el conjunto de simuladores que ofrece iPronics Smartlight.

Capítulo 3

Elementos fundamentales para el desarrollo del simulador

En este capítulo se presentan los elementos principales que intervienen en el desarrollo de los algoritmos de simulación de filtros de alto rendimiento. Se introducen los conceptos de ROADM, como filtro de referencia para el simulador, los elementos que lo componen como guías de onda, phase shifters y PUCs, y las matrices de dispersión y transferencia, que servirán para modelar matemáticamente la respuesta del filtro en simulación.

3.1. Modelado matricial de componentes fotónicas

La mayor parte de los circuitos fotónicos programables, como el que se trata en este Trabajo de Final de Grado, están compuestos por elementos fundamentales que pueden ser descritos por un conjunto de ecuaciones. Estas ecuaciones representan el comportamiento de un rayo de luz que interactúa con un circuito o sistema fotónico [9], y se pueden establecer en forma de matrices. Estas ecuaciones en forma matricial van a ser la herramienta principal para elaborar el algoritmo de simulación del ROADM.

Los dos tipos principales de matrices que modelan la respuesta de un circuito fotónico son la matriz de dispersión o *scattering*, y las matrices de transferencia. Al igual que en la ingeniería de microondas, estas matrices son poderosas herramientas que permiten simular y diseñar dispositivos fotónicos. Poder simular estos circuitos fotónicos permite a los profesionales entender y predecir el comportamiento de sistemas complejos, además de colaborar en la optimización y desarrollo de dichos circuitos sin requerir de los amplios costes temporales y económicos de otras alternativas.

3.1.1. Matriz de Scattering

Dado un sistema óptico de dos puertos como el que se ve en la figura 3.1.1, se definen las amplitudes de las ondas incidentes y reflejadas al sistema como a_1 , a_2 , b_1 y b_2 , que se relacionan con el campo eléctrico y magnético incidente en cada puerto como se ve en la ecuación a continuación.

$$\mathbf{E}_i = \sqrt{\frac{2}{C_i}} \cdot [a_i e^{-j\beta z} + b_i e^{-j\beta z}] \hat{y} \quad (3.1.1)$$

$$\mathbf{H}_i = \sqrt{2C_i} \cdot [a_i e^{-j\beta z} - b_i e^{-j\beta z}] \hat{x} \quad (3.1.2)$$

Donde C_i se relaciona con el índice de refracción del medio y β es la constante de propagación.

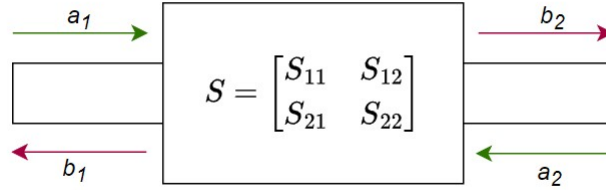


Figura 3.1.1: Sistema óptico de dos puertos representado como caja negra. Fuente: Elaboración propia

En esta situación la matriz de Scattering se utiliza para caracterizar la transmisión y reflexión de un dispositivo óptico, relacionando las ondas incidentes y reflejadas en cada puerto de forma que:

$$b_1 = S_{11}a_1 + S_{12}a_2 \quad (3.1.3)$$

$$b_2 = S_{21}a_1 + S_{22}a_2 \quad (3.1.4)$$

Agrupando las ecuaciones 3.1.9 en forma vectorial y matricial:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}; \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (3.1.5)$$

$$(3.1.6)$$

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \quad (3.1.7)$$

De forma que en conjunto:

$$\mathbf{a} = \mathbf{S} \cdot \mathbf{b} \quad (3.1.8)$$

Esta descripción matricial se puede extender a sistemas ópticos de $N \times N$ puertos, de forma que es capaz de describir a la perfección los elementos que se desean simular. Sin embargo este formalismo dificulta mucho la obtención de una matriz que describa un conjunto de elementos conectados en cascada, y por eso se debe emplear otro tipo matriz que, elaborada a partir de la matriz de scattering, permita realizar esta tarea.

3.1.2. Matriz de Transferencia

La matriz de transferencia surge como herramienta complementaria a la matriz de scattering para el análisis y simulación de la luz que se propaga por elementos dispuestos en cascada, como se ve en la figura 3.1.2.

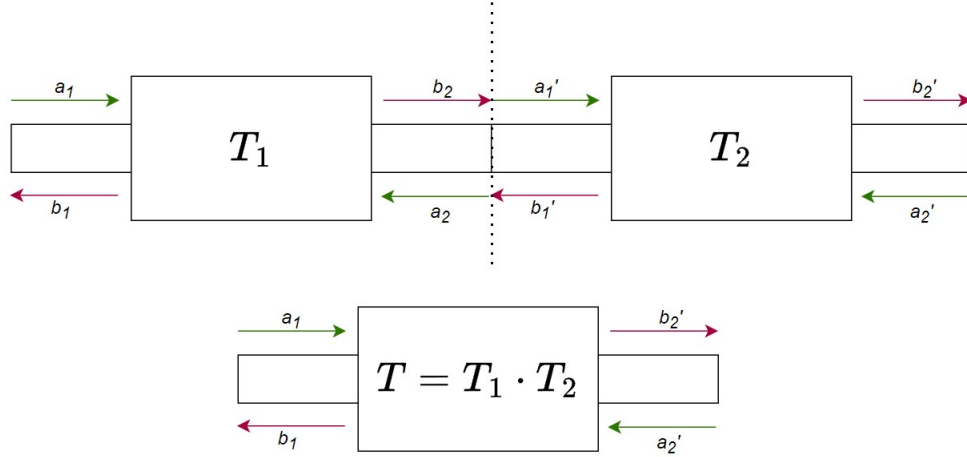


Figura 3.1.2: Matriz de transferencia de dos elementos en cascada. T_i indica la matriz de transferencia asociada a un dispositivo. Fuente: Elaboración propia.

A diferencia de la matriz de scattering, en este modelado matricial lo que se relaciona son la señal que se propaga hacia la entrada del dispositivo y la señal que se propaga hacia la salida, como puede verse en las siguientes expresiones:

$$b_2 = T_{11}a_1 + T_{12}b_1 \quad (3.1.9)$$

$$a_2 = T_{21}a_1 + T_{22}b_1 \quad (3.1.10)$$

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \quad (3.1.11)$$

De forma que todo el conjunto resultaría:

$$\begin{bmatrix} b_2 \\ a_2 \end{bmatrix} = \mathbf{T} \begin{bmatrix} b_1 \\ a_1 \end{bmatrix} \quad (3.1.12)$$

$$(3.1.13)$$

Este formalismo surge para afrontar los problemas de predicción, simulación y diseño de sistemas fotónicos más complejos, debido que es más sencillo de aplicar que la matriz de scattering para un conjunto de elementos conectados [1], y se puede aplicar a componentes de distinta naturaleza siempre y cuando se puedan modelar mediante una matriz de transferencia o scattering. Esto último se debe a que existe una relación directa entre la matriz de scattering y de transferencia de un elemento. Para un sistema de dos puertos la transformación sigue la expresión:

$$T_{11} = -\frac{(S_{11}S_{22} - S_{12}S_{21})}{S_{1S}}; T_{12} = \frac{S_{22}}{S_{21}} \quad (3.1.14)$$

$$T_{21} = -\frac{S_{11}}{S_{21}}; T_{22} = \frac{1}{S_{21}} \quad (3.1.15)$$

Estas expresiones serán de gran importancia para el simulador, ya que permitirá obtener las matrices de transferencia a partir de las matrices de scattering de los elementos fundamentales que componen el filtro, y disponer de la posibilidad de simular múltiples elementos en cascada. Estos elementos se definen en la siguiente sección.

3.2. Elementos principales

Todos los elementos que forman parte del procesador fotónico de iPrionics están compuestos por unas unidades fundamentales. Estas en combinación son capaces de generar distintas arquitecturas de filtros y mallas programables y otorgar un máximo grado de reconfigurabilidad y flexibilidad a dichas estructuras.

3.2.1. Unidad básica programable

La unidad básica programable, denominada PUC por sus siglas en inglés *Programmable Unit Cell*, es la componente principal del procesador fotónico programable. Este elemento posee dos entradas y dos salidas que serán referidas como los *brazos* de la PUC, por los cuales pueden introducirse y extraerse las señales ópticas. Esta PUC está compuesta por un conjunto de acopladores direccionales de 3dB a la entrada y la salida que permiten distribuir las señales introducidas entre los dos brazos, dando mitad de la potencia de la señal a cada brazo.

Para poder realizar las operaciones adecuadas con estas PUCs se dispone de un phase shifter en cada brazo, que es un elemento que permitirá modificar la fase de las señales que lo atraviesan para realizar las operaciones requeridas. Las unidades básicas programables pueden presentar distintas arquitecturas, pero esta configuración en concreto está basada en un Interferómetro Mach-Zehnder (MZI). Este es un dispositivo que permite variar las fases entre dos haces de luz, y realizar interferencias con ellas, que ha sido clave en el campo de la óptica, la fotónica y la cuántica.

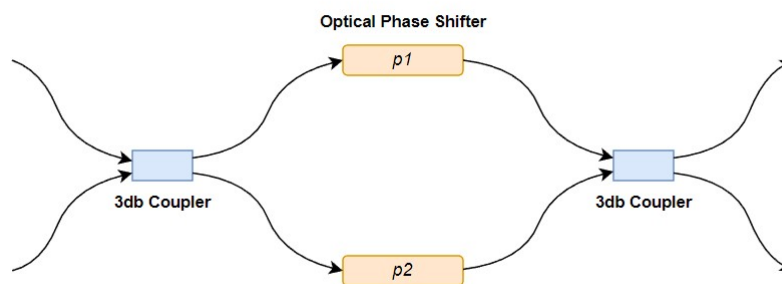


Figura 3.2.1: Esquema de la PUC basada en un MZI. Fuente: Elaboración propia

El comportamiento de la PUC se describirá mediante su matriz de transferencia, que sigue la siguiente expresión:

$$h_{puc} = -je^{-j\Delta} \cdot \begin{bmatrix} \sin(\delta) & \cos(\delta) \\ \cos(\delta) & -\sin(\delta) \end{bmatrix} \cdot \gamma e^{-j\omega\tau} \quad (3.2.1)$$

Donde $\Delta = \frac{\phi_1 + \phi_2 + \alpha}{2}$, $\delta = \frac{\phi_1 - \phi_2 + \alpha}{2}$, γ es el factor de pérdidas de la PUC y $e^{-j\omega\tau}$ es el factor de propagación. ϕ_1 y ϕ_2 son las fases que aplican los dos phase shifters de cada brazo de la PUC al aplicarles corriente, y α es la fase pasiva de los phase shifters. Los phase shifters por lo general deberían generar el mismo desfase al aplicarles la misma corriente, pero debido a defectos de fabricación esta fase no es la misma en cada phase shifter de cada brazo, por lo que α representa esa diferencia de fase entre ambos elementos.

3.2.2. Phase shifters

Los phase shifters son elementos fotónicos a los cuales, tras aplicarles una corriente específica, producen un desfase ϕ en la señal óptica que lo atraviesa. A diferencia que en la malla, en la región donde se ubican los HPBs existen, además de PUCs interconectadas entre ellas, unos phase shifters que actúan solos, fuera de una PUC. La razón de ubicar dichos phase shifters entre PUCs es que permite desplazar en el espectro la respuesta generada por los filtros, para ubicarlas en la región del espectro más conveniente sin alterar la morfología de la respuesta, y conservar parámetros importantes como puede ser el *rango espectral libre* o FSR (Free Spectral Range) o la relación de extinción o ER (Extinction Ratio). El FSR es el espacio que hay entre dos máximos sucesivos de la respuesta espectral y el ER la diferencia que hay entre el valor máximo y mínimo de potencia óptica.

Su respuesta se modela como un simple desfase del tipo $e^{-j\phi}$ como se ha podido ver en la expresión anterior de la PUC.

3.2.3. Guías de onda

Las PUCs y los phase shifters estarán conectados mediante guías de silicio en el circuito fotónico integrado. A lo largo de estas guías la señal principalmente sufrirá un desfase y unas pérdidas al recorrerlas. A la hora de modelarlas, los filtros que se van a simular suelen tener una estructura similar, pero cuentan con una diferencia de longitud en uno de los brazos, que es la que genera el efecto de interferencia y forma a la respuesta del filtro. Por esto guías también se modelarán como una fase que permanecerá siempre constante, a diferencia de las fases de los phase shifters que variarán según la corriente que se les aplique.

3.3. Filtro a simular: ROADM

Un ROADM (Reconfigurable Optical Add-Drop Multiplexer) es un filtro complejo y reconfigurable que permite multiplexar y demultiplexar distintos rangos de longitudes de onda. Es el filtro del que se va a realizar el simulador en software y estará ubicado en la suite de HPBs de la FPPGA de iPronics. Un esquema de su funcionamiento se puede ver en la figura 3.3.2.

Este elemento se ubica en las redes de fibra óptica para permitir una gestión de tráfico eficiente, sin la necesidad de intervenir manualmente en la infraestructura de la red. Se desarrollaron con el objetivo de adaptarse a la demanda creciente de redes ópticas escalables que cumplieren con las necesidades de tráfico de la sociedad actual, habilitando el desarrollo de redes ópticas de multiplexado en longitud de onda (WDM) como una alternativa prometedora a las redes ópticas de multiplexado en tiempo (TDM) para sistemas de red metro o red core para 5G entre otros campos, como puede verse en la figura 3.3.1 [6].

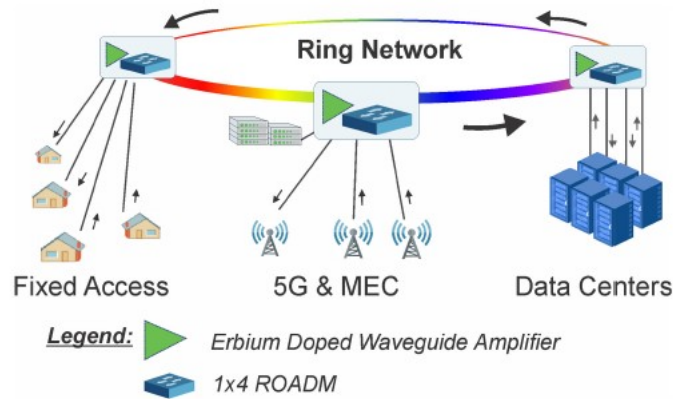


Figura 3.3.1: Ilustración de una red metro en anillo que emplea ROADM. Fuente [6].

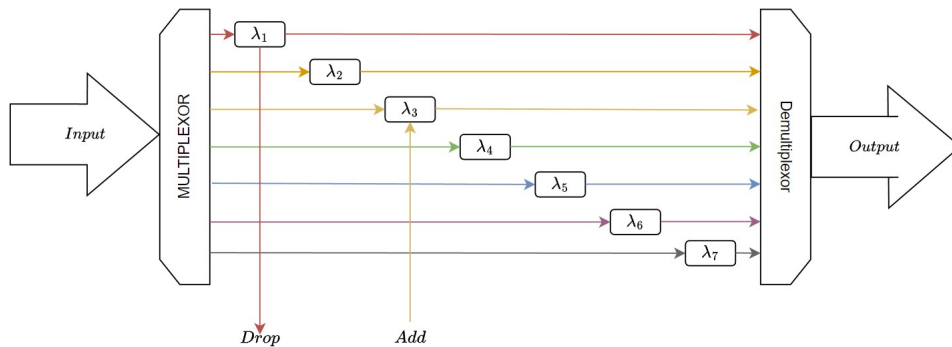


Figura 3.3.2: Esquema del funcionamiento de un ROADM. Este ROADM tiene 7 canales, está sacando de la señal la λ_1 y añadiendo λ_3 . Fuente: Elaboración propia

Este filtro se puede realizar empleando distintas arquitecturas. La empleada en el procesador fotónico de iPrronics está basada en una combinación de interferómetros Mach-Zehnder especiales llamados interferómetros Mach-Zehnder duales (MZI duales) [12]. Estos MZI duales consisten en dos estructuras de tipo MZI combinadas que ofrecen una respuesta interferométrica un poco más compleja que un MZI al uso, como se puede ver en la figura 3.3.3 extraída del artículo de referencia [12].

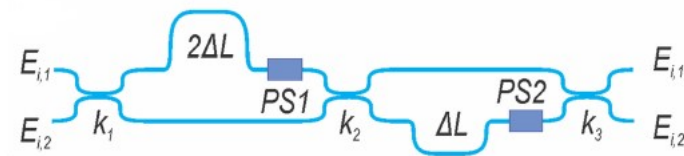


Figura 3.3.3: Esquema del MZI Dual y respuesta espectral. Fuente: [12]

Este elemento posee unas propiedades únicas que permitirán realizar un ROADM empleando las componentes básicas vistas anteriormente. Para las zonas con acopladores k_i de la imagen se emplearán PUCs, que permitirán ajustar fácilmente el acoplo necesario para obtener la respuesta deseada. El acoplo es un fenómeno que permite que una señal óptica, que discurre a través de una guía, se transfiera a otra guía muy cercana si se dan las condiciones. En concreto la k_i indica

la proporción de luz que se propagará de un brazo hacia el otro, por lo que sus valores oscilan entre 0, cuando no hay acoplo, y 1, cuando la luz pasa totalmente hacia el otro brazo. Este valor es uno de los parámetros programables de la estructura.

Existirán dos tipos de MZI duales, cuya diferencia residirá en la diferencia de longitud entre los brazos ΔL para obtener un FSR más ancho o estrecho en el espectro. Los parámetros de longitud, fase y acoplo se especificarán en apartados posteriores.

Los MZI duales se unirán en cascada para configurar el ROADM ubicado en los HPBs de la FPPGA de iPronics, formando la estructura de la imagen 3.3.4 de la que se va a extraer la respuesta espectral, aplicando el método de matrices de transferencia. Este ROADM en particular es capaz de multiplexar y demultiplexar cuatro canales distintos, que podrán entrar por los puertos ADD y salir por los puertos DROP de la estructura, o incorporarse a la señal principal que entra por el puerto Input, y sale por el puerto Output.

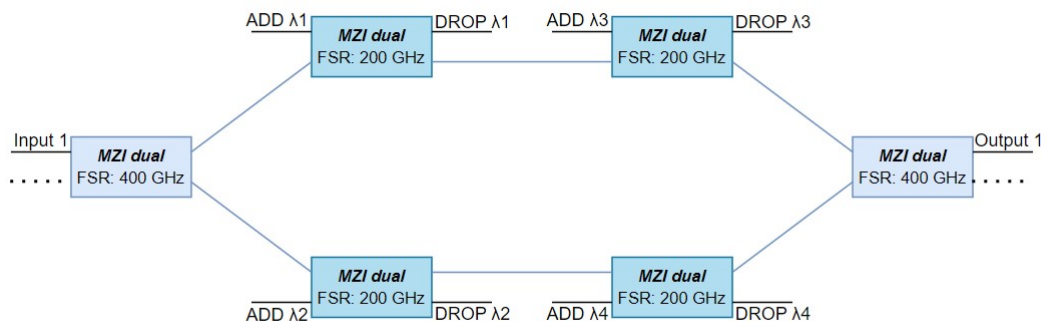


Figura 3.3.4: Esquema de bloques del ROADM de iPronics. Fuente: Elaboración propia

Capítulo 4

Desarrollo y resultados de los algoritmos

En este capítulo se presenta el desarrollo del algoritmo de simulación desarrollado para uno de los filtros incluidos en la suite de HPBs de la FPPGA, el ROADM de 4 canales. Se empleará como base los conceptos teóricos mostrados en el capítulo anterior y la plataforma de software iPronics Smartlight basada en Python.

El contenido de esta sección se desarrolla de la siguiente manera: Inicialmente se explicarán las unidades básicas que componen el ROADM, que son las PUCs, los phase shifters y las guías de silicio que los unen, y cómo se modelan en el software de iPronics. Tras esto se muestra como se conectan los distintos elementos para configurar un MZI dual, y simular su respuesta espectral. Finalmente, se desarrollará la configuración de los MZI duales para simular la función de transferencia de un ROADM de 4 canales, mostrando además como en función de los parámetros programables de las PUCs y los phase shifters se pueden generar distintas configuraciones de multiplexado, demultiplexado y enrutamiento.

4.1. Herramientas utilizadas

Para este proyecto se ha empleado como herramienta principal Python, un lenguaje de programación orientado a objetos de alto nivel altamente utilizado hoy en día por su facilidad de aprendizaje y versatilidad en una gran variedad de campos, como las telecomunicaciones, el análisis de datos o el desarrollo web. La principal razón de su uso, además de lo mencionado anteriormente, es por la necesidad de utilizar el paquete de software nativo de las FPPGAs de iPronics, llamado Smartlight, el cual contiene todos los elementos necesarios para implementar los algoritmos de simulación de filtros fotónicos, y permitir su configuración desde un nivel alto de abstracción, solo necesitando unos conocimientos fundamentales de fotónica.

Por lo tanto, los paquetes empleados de Python en su versión 3.8.0 para el desarrollo del proyecto son:

- **iPronics Smartlight:** Permite disponer de información sobre los elementos fundamentales que componen el ROADM que se va a configurar y la capacidad de abstracción suficiente para facilitar su configuración. De este paquete se van a extraer la mayor parte de elementos y funciones que se van a emplear para el desarrollo de los algoritmos. Se necesita una versión

superior a la 1.6.0.

- **Numpy:** Paquete fundamental de Python para aplicaciones científicas y de ingeniería. En este caso particular permite realizar una gran variedad de operaciones matemáticas matriciales que habilitarán un cálculo y procesamiento eficaz de las matrices de transferencia de los elementos que componen el ROADM. La versión ha de ser superior a la 1.21.5.
- **Matplotlib:** Uno de los paquetes de Python más empleados para la visualización de datos. Permitirá observar los resultados obtenidos tras computar las respuestas espectrales de los elementos que se van a utilizar. La versión ha de ser superior a la 3.5.0.

4.2. Implementación en software de elementos fundamentales

Como se ha visto en capítulos anteriores, los elementos fundamentales para elaborar el simulador del ROADM son las PUCs, los phase shifters y las guías de onda. En Smartlight se pueden encontrar las distintas clases que definen estos elementos, cuyos aspectos fundamentales se enuncian a continuación. Para este algoritmo, la fase inducida al recorrer la guía de onda no se encuentra definida mediante una clase, sino que se calcula mediante una función que depende de la longitud de la guía, el índice de grupo y las longitudes de onda que se van a simular.

4.2.1. Atributos y métodos PUCs

A continuación se muestran los parámetros fundamentales que definen una PUC. Aunque las PUCs de iPronics poseen más atributos y características que las mostradas, solo se nombrarán los atributos básicos y los que tengan un impacto directo en la implementación del algoritmo de simulación.

Atributos	Descripción
<i>Nombre</i>	Parámetro asociado al identificador de la PUC que se está simulando, para poder asociar la PUC simulada a la PUC real de hardware.
<i>Posición y rotación</i>	Para poder mantener la correspondencia con los elementos reales de hardware se establecen estos atributos mediante los parámetros self.pos y self.rot entre otros.
<i>Tecnología</i>	Mediante el parámetro self.technology podemos establecer la tecnología de elaboración de la PUC y poder obtener sus características asociadas como pérdidas de inserción, índice de grupo, etc. En este caso la tecnología será Silicon Photonics.
<i>Fases aplicadas</i>	Parámetro con cada una de las fases aplicadas a los phase shifters de cada brazo de las PUCs.

Tabla 4.2.1: Atributos fundamentales de la implementación en software de la PUC

Donde el parámetro de fases aplicadas corresponde a las fases que se aplicarán a las PUCs siguiendo la ecuación de la PUC 3.2.1 vista en el capítulo anterior, y cuyos valores se distribuyen en un rango de 0 a 2π radianes.

A continuación se muestran los métodos fundamentales empleados para simular el comportamiento de las PUCs en software:

Método	Descripción
<i>compute_response()</i>	Método fundamental para obtener la respuesta espectral de una PUC.
<i>set_coupling_factor_phase()</i>	Método fundamental para establecer el factor de acoplo, así como la fase cruzada de los dos brazos de la PUC.
<i>set_wavelengths()</i>	Método para establecer el número de longitudes de onda que se van a emplear para computar la respuesta espectral de la PUC. Requiere un rango discreto de longitudes de onda y su espaciado. Las unidades de las longitudes de onda serán los nanómetros.

Tabla 4.2.2: Métodos fundamentales de la implementación en software de la PUC

El método *compute_response()* permitirá calcular la matriz de scattering de la PUC, para después obtener la matriz de transferencia, que es la expresión matricial deseada para el algoritmo de simulación.

4.2.2. Atributos y métodos Phase Shifters

Al igual que con las PUCs anteriores abajo se muestran los atributos y métodos más importantes en la implementación de software de los phase shifters que se utilizarán para elaborar el ROADM:

Atributo	Descripción
<i>Fase pasiva</i>	Fase que induce el phase shifter por defecto en radianes
<i>Fase aplicada</i>	Fase que debe aplicar el phase shifter en radianes.
<i>Tecnología</i>	Mediante el parámetro <i>self.technology</i> podemos establecer la tecnología de elaboración de la PUC y poder obtener sus características asociadas, como pérdidas de inserción, índice de grupo, etc. En este caso la tecnología será Silicon Photonics.

Tabla 4.2.3: Atributos fundamentales de la implementación en software de un phase shifter

A continuación, los métodos principales para la simulación de los phase shifters:

Método	Descripción
<i>compute_response()</i>	Método fundamental para obtener la respuesta espectral de un phase shifter.
<i>set_phase()</i>	Método fundamental para generar en el phase shifter el desfase deseado.
<i>self.set_wavelengths()</i>	Método para establecer el número de longitudes de onda que se van a emplear para computar la respuesta espectral del phase shifter. Requiere un rango discreto de longitudes de onda y su espaciado. Las unidades de las longitudes de onda serán los nanómetros.

Tabla 4.2.4: Métodos fundamentales de la implementación en software de un phase shifter

4.3. Diseño del interferómetro Mach-Zehnder dual

Una vez explorados los elementos básicos y sus particularidades se explicará la elaboración del primer algoritmo de simulación para el interferómetro Mach-Zehnder dual, que es el primer paso

requerido para diseñar el algoritmo de simulación del ROADM.

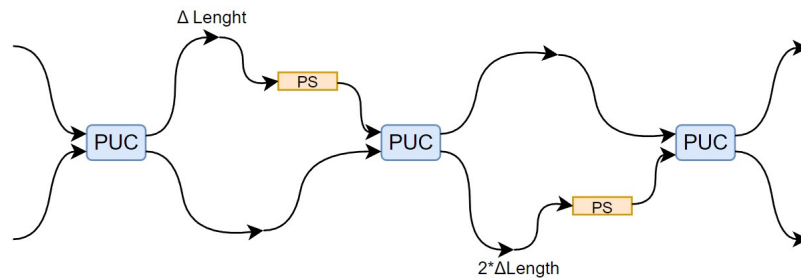


Figura 4.3.1: Esquema del MZI dual a modo de sistema de PUCs y Phase shifters interconectados

Para enfocarlo obtendremos su respuesta espectral empleando el sistema de arriba, que imita el comportamiento MZI dual empleando las componentes propias de la FPPGA, que son las PUCs y los phase shifters. Cabe destacar que la diferencia en la longitud de los brazos, que aparece como ΔLength y $2 \cdot \Delta\text{Length}$, se indica así porque será un elemento **parametrizable** del simulador que se va a desarrollar. Esto no quiere decir que en el hardware real se puedan modificar dichas longitudes. Si se recuerda la estructura del ROADM comentada en el apartado anterior se puede apreciar que está formada por 2 tipos de MZI duales con FSRs distintos. La diferencia en los FSR de ambos elementos radica en ese valor de longitud que aparece indicado, y por ello para poder simular ambos elementos con el mismo código se dejará esa longitud como un parámetro de entrada del simulador.

Previamente al cálculo de la respuesta espectral se necesita crear una código que represente en software el MZI dual. Este código en programación se denomina **clase**, y es un código que contiene tanto los atributos como las funciones del MZI dual, al igual que se tenía con las PUCs y los PhaseShifters. Esto se necesita para poder ser instanciado posteriormente al elaborar el simulador del ROADM. Los atributos y métodos principales de la clase del MZI dual son los siguientes:

Atributo	Descripción
<i>Delta Length</i>	Diferencia de longitud entre los primeros brazos del MZI dual. Se empleará este valor y su doble para simular los filtros con el FSR adecuado.
<i>PUCs</i>	Todas las PUCs que componen la estructura.
<i>Phase Shifters</i>	Todos los Phase Shifters que componen la estructura.
<i>Technology</i>	Tecnología con la que se elabora el filtro. Es necesario para poder pasar el valor a las PUCs y Phase Shifters. En este caso es Silicon Photonics
<i>Wavelengths</i>	Rango discreto de longitudes de onda en las que se calculará la respuesta espectral del filtro.

Tabla 4.3.1: Atributos fundamentales de la implementación en software de un MZI dual

Método	Descripción
<i>compute_response()</i>	Calcula la matriz de transferencia de la estructura.
<i>set_wavelengths()</i>	Establece en los elementos del MZI dual el rango de longitudes de onda en el que se va a calcular la respuesta espectral.
<i>get_radians()</i>	Permite obtener el desfase aplicado por una diferencia de longitud ΔLength cualquiera entre los brazos del MZI.

Tabla 4.3.2: Métodos fundamentales de la implementación en software de un MZI dual

Para una implementación eficiente del código se interpretará la estructura como un conjunto de etapas individuales conectadas en cascada, y para obtener la respuesta espectral de la estructura se calcularán las matrices de transferencia de cada una de las etapas por separado, y se multiplicarán entre ellas de una en una y de izquierda a derecha.

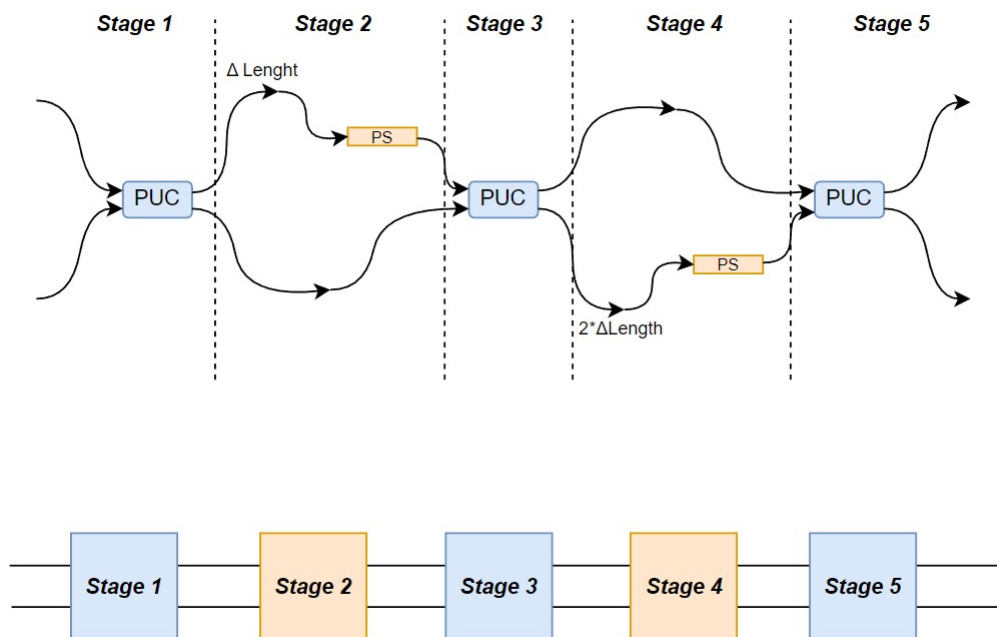


Figura 4.3.2: Esquema de equivalencia de MZI dual a conjunto de elementos en cascada, y su representación en etapas

Se ejecutará el algoritmo para simular el comportamiento del filtro mediante el método *compute_response()* siguiendo el estándar de la compañía para las funciones de cálculo de la respuesta espectral. El pseudo código que desarrolla el funcionamiento de este método, y que permitiría obtener la respuesta espectral de dicha estructura sería el siguiente:

Algoritmo 1: Algoritmo de desarrollo del método `compute_response()` del MZI dual

Input : PUCs, Phase Shifters, $\Delta Length$, Technology, Wavelengths**Output:** Respuesta espectral del MZI dual, o matriz de transferencia para cada una de las longitudes de onda establecidas

```
ComputeResponse(PUC1, Wavelengths)
ComputeResponse(PUC2, Wavelengths)
ComputeResponse(PUC3, Wavelengths)

ComputeResponse(PhaseShifter1, Wavelengths)
ComputeResponse(PhaseShifter2, Wavelengths)

GetRadians( $\Delta Length$ , Technology)
GetRadians( $2\Delta Length$ , Technology)

StageTransferMatrix(PUC1)
StageTransferMatrix(PhaseShifter1, Radians of  $\Delta Length$ )
StageTransferMatrix(PUC2)
StageTransferMatrix(PhaseShifter1, Radians of  $2\Delta Length$ )
StageTransferMatrix(PUC3)

CascadedMatrixEinsum(All Stages)
```

Donde **ComputeResponse** obtiene las matrices de transferencia de cada una de las PUCs y los phase shifters utilizados, **GetRadians** obtiene el desfase aplicado por la diferencia de longitud entre los brazos del MZI, **StageTransferMatrix** obtiene la función de transferencia de cada una de las etapas y finalmente **CascadeMatrixEinsum** realiza el producto de cada una de las matrices de transferencia de las etapas aplicando un método de Numpy llamado Einsum, que deriva de Einstein Summation. Este método permite realizar un cálculo muy rápido de un producto matricial mediante una computadora empleando la notación de Einstein para operar con matrices.

4.3.1. Resultados obtenidos

Una vez desarrollado el simulador del MZI dual se probará el rendimiento del algoritmo comparándolo con los resultados mostrados en el artículo de referencia [12]. Las gráficas que aparecen a continuación están generadas por el código que aparece en el Anexo I de este trabajo empleando el paquete matplotlib, y estableciendo las longitudes adecuadas a la estructura de los filtros en hardware:

La figura 4.3.3 muestra la respuesta del filtro para todas las combinaciones de puertos de entrada y salida. Por ejemplo, arriba muestra la respuesta del filtro entre entrar por el brazo superior, que es el puerto 0, y salir por el brazo superior o el inferior. La entrada y salida por el mismo brazo se denomina **bar**, y entrar y salir por brazos opuestos se denomina **cross**. Esto aparece en la leyenda de las gráficas.

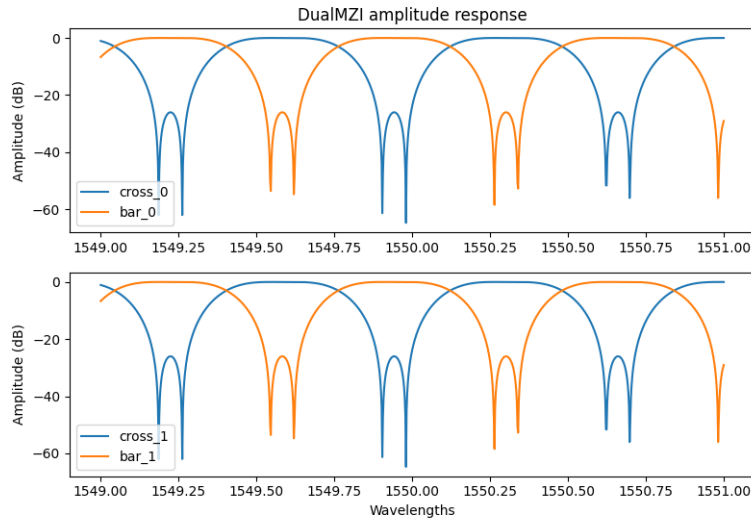


Figura 4.3.3: Resultados del simulador de MZI dual

Para poder apreciar el efecto de las distintas diferencias de longitud $\Delta Length$ que se van a aplicar, abajo se muestra una gráfica comparativa de los 2 tipos de MZI dual que se van a emplear para el desarrollo del ROADM, y cómo empleando las particularidades de esta estructura se pueden eliminar ciertas bandas de paso.

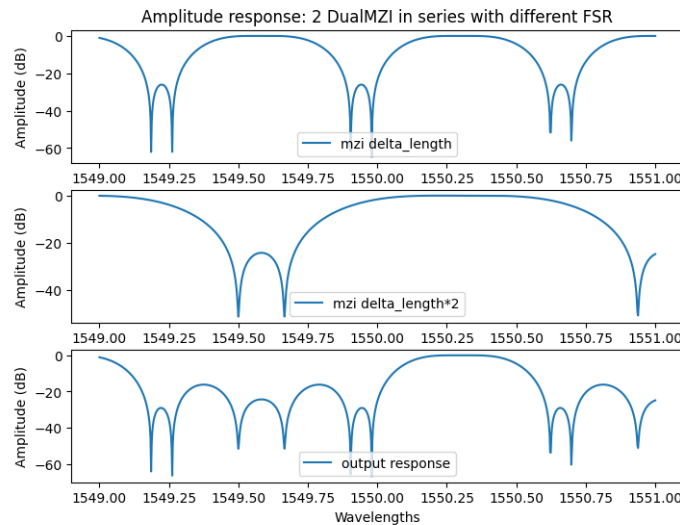


Figura 4.3.4: Comparativa de los simuladores de MZI duales con distintas diferencias de longitud

Antes de proceder al diseño del ROADM de 4 canales se presenta esta última figura, donde se han simulado 2 MZI duales, cada uno con un parámetro $\Delta \cdot Length$ distinto, para posteriormente unirlos en cascada. Con esto se puede apreciar el principio de funcionamiento de este conjunto de estructuras. El MZI con el FSR más grande (figura del medio) permite dejar paso a los canales de interés y se aprovecha la zona de baja atenuación para anular las bandas que no se desea que pasen. Para la otra salida del MZI dual que no está contemplada en esta imagen sucedería el mismo proceso, pero con las bandas de frecuencias que ahora están siendo atenuadas. También

cabe destacar que gracias a la presencia de los phase shifters en los brazos del MZI dual esta respuesta espectral puede desplazarse para ajustarse a las bandas de interés, modificando la fase de ambos brazos. Para que se mantengan las propiedades de la estructura, las fases aplicadas en los phase shifters de cada brazo deben seguir la siguiente expresión:

$$\phi_{PS2} = \pm \frac{1}{2} * \phi_{PS1} + 2\pi n \quad (4.3.1)$$

Donde ϕ_{PS2} es la fase que hay que aplicar al segundo phase shifter y ϕ_{PS1} la fase que hay que aplicar al primero.

4.4. Diseño del ROADM de 4 canales

Tras comprobar el correcto funcionamiento del simulador de software del MZI dual, se procederá con el diseño del algoritmo de simulación del ROADM, la configuración de los elementos que lo componen para obtener las respuestas deseadas y los resultados obtenidos de la simulación de la estructura.

Para el diseño del simulador del ROADM de cuatro canales se va a seguir la estela del filtro anterior. Se planteará una estructura inicial compuesta de los elementos más sencillos y se tratará de organizarlos de manera inteligente para que el proceso de cómputo sea lo más rápido y eficiente posible.

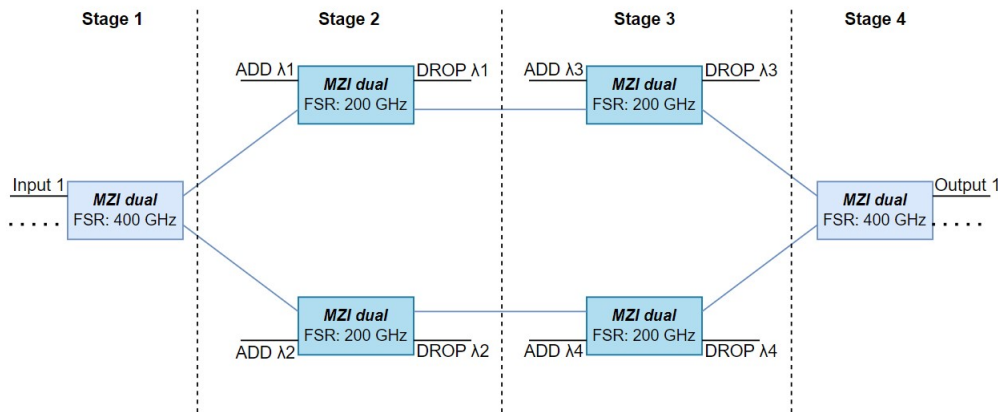


Figura 4.4.1: Estructura de etapas para el simulador del ROADM

Este caso resulta más complicado que el anterior, ya que para el MZI dual cada etapa poseía no solo el mismo número de entradas y salidas, sino que además todas las entradas y salidas de cada etapa se conectaban a la siguiente, por lo que resultaba más sencillo generar la matriz de transferencia para una etapa. En este caso no es así, por lo que previamente al cálculo de la matriz de transferencia se realizó una reestructuración de la ubicación de los MZI duales, para que al definir las matrices de transferencia asociadas a cada etapa se pudiera realizar el producto en condiciones. Cabe destacar que aunque esta reestructuración pueda resultar tediosa para el desarrollador, porque requiere la definición manual de las matrices de cada etapa y el diseño de una estructura equivalente que siga el comportamiento del filtro, es preferible al cálculo individual de cada una de las combinaciones de entrada y salida del ROADM, que incrementaría el coste computacional y el tiempo de ejecución en comparación con el primer enfoque. Esto es justo lo que se desea evitar al realizar un simulador, cuya ventaja radica en poder experimentar resultados similares a los obtenidos en hardware sin tener que recurrir al dispositivo físico y a los elevados tiempos de ejecución.

```

stage_1 = np.array(
    [
        [one, zero, zero, zero, zero, zero],
        [zero, dm1u[0, 0, :], dm1u[0, 1, :], zero, zero, zero],
        [zero, dm1u[1, 0, :], dm1u[1, 1, :], zero, zero, zero],
        [zero, zero, zero, dm1d[0, 0, :], dm1d[0, 1, :], zero],
        [zero, zero, zero, dm1d[1, 0, :], dm1d[1, 1, :], zero],
        [zero, zero, zero, zero, zero, one],
    ]
)

```

Figura 4.4.2: Ejemplo de matriz que simula la etapa 1 del ROADM

En lugar de indicar como se ha realizado la reestructuración de cada etapa para facilitar el producto de las matrices de transferencia debido a su extensión, arriba podemos ver un ejemplo de matriz de transferencia asociada a la primera etapa del ROADM. Las matrices están dispuestas siguiendo la nomenclatura algebraica de vectores columna. Los valores **dm1u** y **dm1d** corresponden a las respuestas espectrales de los MZI duales de la primera etapa de la parte superior e inferior. Los parámetros **one** y **zero** son vectores que contienen ceros o unos para todo el rango de longitudes de onda establecido para simulación. Esto es importante porque aunque la forma que poseen las respuestas del método `compute_response()` es matricial, en realidad son tensores que realizan el cálculo de la matriz de transferencia para todos los inputs y outputs de la estructura, y para todas las longitudes de onda establecidas en simulación.

Este proceso se realizará de forma equivalente para las tres etapas restantes, y se añadirá un par de matrices más al inicio y al final de ceros y unos para reorganizar las entradas y salidas de la estructura. Estas dos matrices sirven para que el usuario tenga un orden homogéneo y sencillo de entradas y salidas, y que estas se adapten a la estructura del ROADM para que no se altere su configuración

Tras el proceso de diseño se desarrolla la instanciación del simulador de software del ROADM. Esta presentará los siguientes atributos y métodos fundamentales conforme al estándar del resto de elementos:

Métodos	Descripción
<i>Delta Length</i>	Diferencia de longitud entre los primeros brazos de los MZI duales que componen el ROADM. Será el valor establecido en hardware y servirá para instanciar los MZI duales con los valores adecuados.
<i>MZI dual 1</i>	Primer tipo de MZI de la estructura, con un FSR de 400 GHz.
<i>MZI dual 2</i>	Segundo tipo de MZI de la estructura, con un FSR de 200 GHz.
<i>Technology</i>	Tecnología con la que se elabora el filtro. Es necesario para poder pasar el valor a los MZI duales y de ahí a las PUCs y phase shifters. En este caso es Silicon Photonics.
<i>Wavelengths</i>	Rango discreto de longitudes de onda en las que se calculará la respuesta espectral del filtro.

Tabla 4.4.1: Atributos fundamentales de la implementación en software de un ROADM de 4 canales

Métodos	Descripción
<code>compute_response()</code>	Calcula la matriz de transferencia de la estructura.
<code>set_wavelengths()</code>	Establece en los elementos del MZI dual el rango de longitudes de onda en el que se va a calcular la respuesta espectral.
<code>get_radians()</code>	Permite obtener el desfase aplicado por una diferencia de longitud entre los brazos del MZI.

Tabla 4.4.2: Métodos fundamentales de la implementación en software de un MZI dual

Al igual que con el MZI dual, ya que el método fundamental del simulador de software es el método `compute_response()`, abajo se muestra el pseudocódigo que modela el algoritmo de simulación de la respuesta espectral:

Algoritmo 2: Algoritmo de desarrollo del método `compute response` para el ROADM de 4 longitudes de onda

```

Input  : DualMZIType1, DualMZIType2, ΔLength, Technology
Output: Matriz de transferencia del ROADM de 4 longitudes de onda

/* Obtener todas las respuestas espectrales de los MZI duales          */
ComputeResponse(DualMZIType1 No1)
ComputeResponse(DualMZIType1 No2)
ComputeResponse(DualMZIType2 No1)
ComputeResponse(DualMZIType2 No2)
ComputeResponse(DualMZIType2 No3)
ComputeResponse(DualMZIType2 No4)

/* Generar las matrices de transferencia de cada etapa y matrices de
   reestructuración                                                    */
InputRearrangeTransferMatrix()
StageTransferMatrix(DualMZIType1 No1)
StageTransferMatrix(DualMZIType2 No1,DualMZIType2 No2)
StageTransferMatrix(DualMZIType2 No3,DualMZIType2 No4)
StageTransferMatrix(DualMZIType1 No2)
OutputRearrangeTransferMatrix()

/* Obtener matriz de transferencia de todas las etapas                */
CascadedMatrixEinsum(All Stages)
    
```

Donde **ComputeResponse** permite obtener las matrices de transferencia de cada uno de los MZI duales que se han utilizado, **StageTransferMatrix** obtiene la función de transferencia de cada una de las etapas, **InputRearrangeTransferMatrix** y **OutputRearrangeTransferMatrix** reorganiza las entradas y salidas del ROADM y finalmente **CascadeMatrixEinsum** realiza el producto de cada una de las matrices de transferencia de las etapas mediante el método `einsum` de `numpy` al igual que con el MZI dual.

4.4.1. Resultados obtenidos y configuraciones del ROADM

Tras el diseño del simulador del ROADM se elaboró un código para poder ejecutar la simulación, configurar los elementos del ROADM y probar diversas configuraciones, el cual se ubica en el Apéndice A de la memoria.

Para ejecutar la simulación, en primer lugar se instancia el simulador del ROADM que se ha diseñado, con las diferencias de longitud de los brazos establecidas en hardware, que son $\Delta L_1 = 872,1 \mu\text{m}$ y $\Delta L_2 = 436 \mu\text{m}$. Estas longitudes se aplicarán a los distintos MZI duales que componen el ROADM. Tras esto se establece el rango de longitudes de onda que se va a simular entre 1549 nm y 1551 nm, empleando 2001 valores equiespaciados dentro de ese rango, y se configuran las PUCs de los MZI duales estableciendo unos factores de acoplo $k_1 = 0.9248$, $k_2 = 0.280$ y $k_3 = 0.5$ siguiendo lo indicado en [12]. Tras esto el ROADM ya estaría configurado para su funcionamiento. Estos factores de acoplo no deben modificarse, ya que la estructura dejaría de comportarse como un ROADM. Para cambiar entre las distintas opciones de multiplexado y demultiplexado que se verán a continuación solo faltaría modificar los phase shifters de cada MZI dual con los valores adecuados. Para exponer el resultado se ha configurado una opción de demultiplexado de 4 longitudes de onda. Se introducirá una señal por el puerto **Input 1** y se analizarán las salidas por los puertos **DROP λ_1** , **DROP λ_2** , **DROP λ_3** y **DROP λ_4** , siguiendo la nomenclatura establecida en las imágenes previas, por pares y en conjunto.

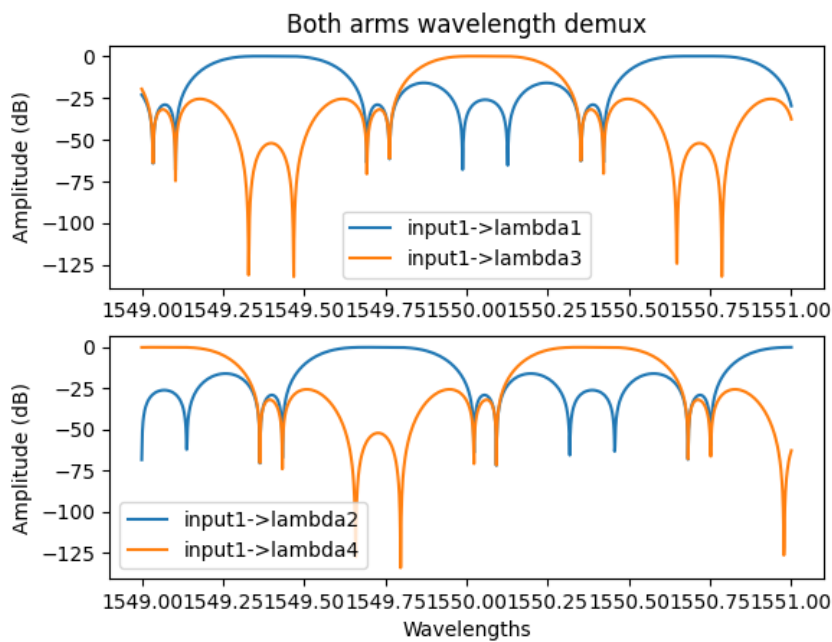


Figura 4.4.3: Respuesta del ROADM por pares en configuración de demultiplexado de 4 canales

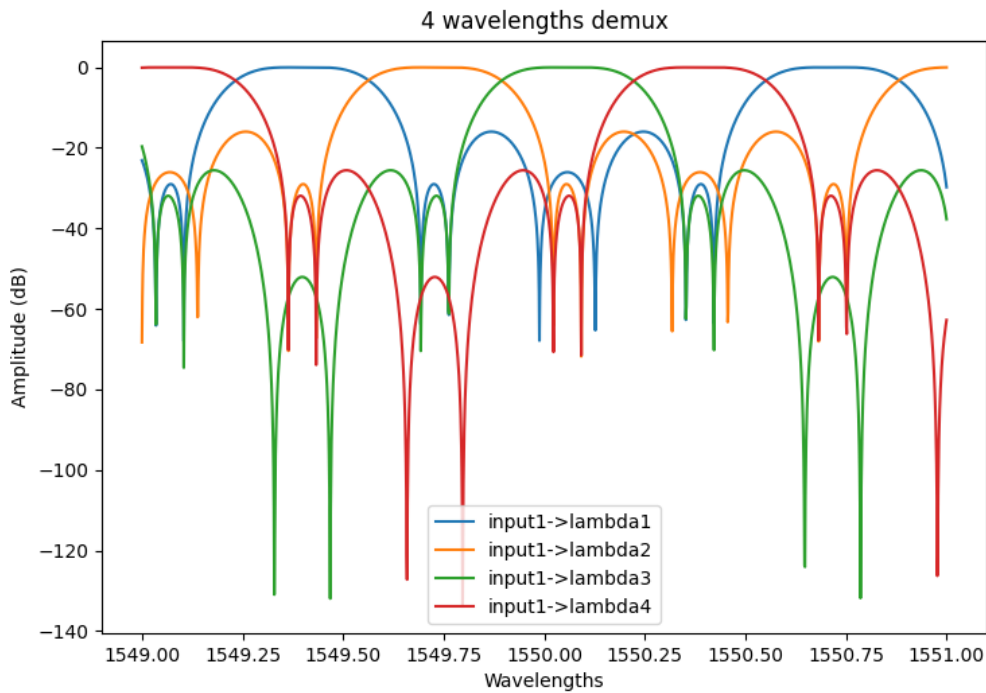


Figura 4.4.4: Respuesta del ROADM total en configuración de demultiplexado de 4 canales

Se puede ver en las imágenes superiores (4.5.7 Y 4.4.3) que el simulador del ROADM se comporta de la manera ilustrada por el artículo de referencia [12], separando la señal de entrada en 4 canales que salen por cada uno de los puertos establecidos.

Para generar este comportamiento en la estructura se han empleado los siguientes valores de fase para los phase shifters de cada MZI dual:

Elemento	Fase Phase Shifter 1	Fase Phase Shifter 2
<i>MZI dual 1</i>	0 radianes	0 radianes
<i>MZI dual 2 superior</i>	π radianes	$\frac{3}{2}\pi$ radianes
<i>MZI dual 3 superior</i>	π radianes	$\frac{3}{2}\pi$ radianes
<i>MZI dual 2 inferior</i>	0 radianes	0 radianes
<i>MZI dual 3 inferior</i>	0 radianes	π radianes
MZI dual 4	0 radianes	0 radianes

Tabla 4.4.3: Fases aplicadas para la configuración de demultiplexación de 4 canales

4.5. Capacidades y configuraciones del ROADM

La principal ventaja de este ROADM frente a otras alternativas es su gran capacidad de reconfigurabilidad, que permite disponer de una gran variedad de funcionalidades realizando ligeras modificaciones en las fases aplicadas en los phase shifters. A continuación se mostrarán varios ejemplos de configuraciones de demultiplexado, adicionales a la anterior, para 2 y 3 canales, junto con la posibilidad de multiplexado y de reasignación variable de los canales a los puertos, que habilita la posibilidad de enrutamiento. Los valores de fase empleados en cada configuración

se han obtenido de manera experimental haciendo pruebas con el simulador y manteniendo la relación de fase enunciada en el apartado del MZI dual mediante la ecuación 4.3.1, reduciendo el tiempo empleado en la ejecución y las pruebas que habría requerido configurarlo y ejecutarlo disponiendo de un dispositivo físico. Para analizar las gráficas hay que fijarse en la leyenda, que indica a qué par de puertos está asociada la respuesta espectral. La nomenclatura es la misma empleada en el esquema del ROADM que se ve en la figura 3.3.4. Las señales que no son demultiplexadas hacia los puertos DROP saldrán por el puerto Output 1 del ROADM, y en la configuración de multiplexado se incluirán las señales por los puertos ADD y se combinarán con la señal de entrada por el puerto Input 1 para salir por el puerto Output 1.

4.5.1. Demultiplexación de 2 canales

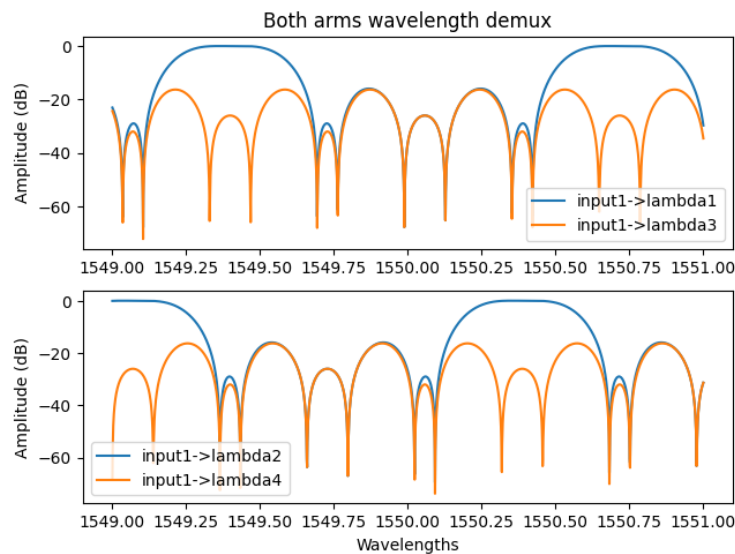


Figura 4.5.1: Respuesta del ROADM por pares en configuración de demultiplexado de 2 canales

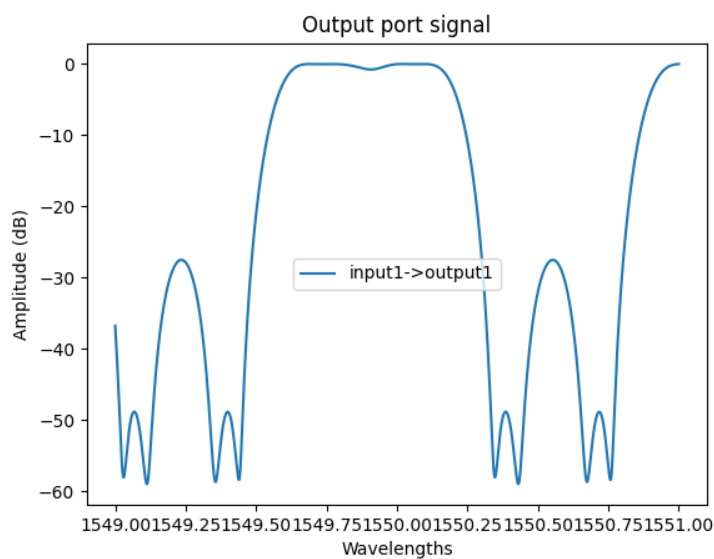


Figura 4.5.2: Respuesta del ROADM en el output en configuración de demultiplexado de 2 canales

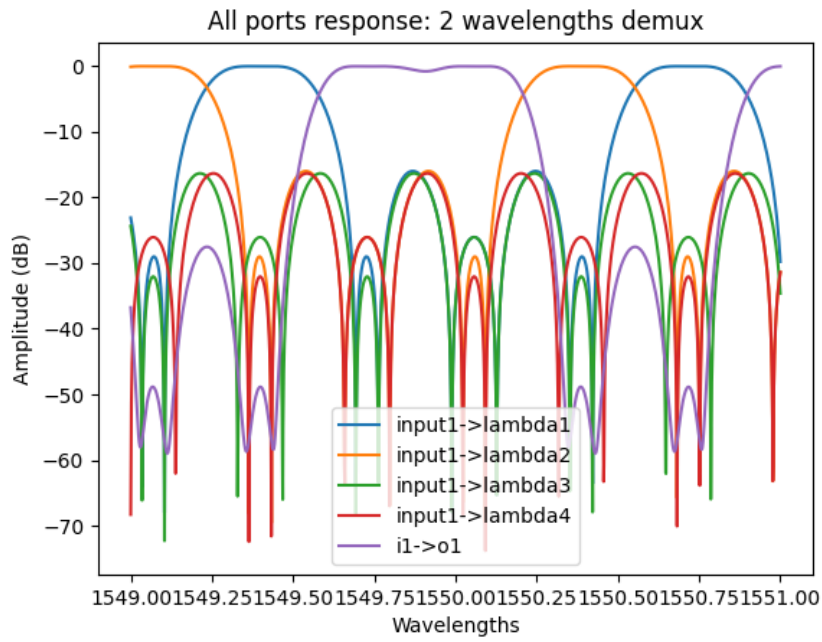


Figura 4.5.3: Respuesta del ROADM total en configuración de demultiplexado de 2 canales

Elemento	Fase Phase Shifter 1	Fase Phase Shifter 2
<i>MZI dual 1</i>	0 radianes	0 radianes
<i>MZI dual 2 superior</i>	π radianes	$\frac{3}{2}\pi$ radianes
<i>MZI dual 3 superior</i>	π radianes	$\frac{3}{2}\pi$ radianes
<i>MZI dual 2 inferior</i>	0 radianes	π radianes
<i>MZI dual 3 inferior</i>	0 radianes	π radianes
<i>MZI dual 4</i>	0 radianes	0 radianes

Tabla 4.5.1: Fases aplicadas para la configuración de demultiplexación de 2 canales

4.5.2. Demultiplexacion de 3 canales

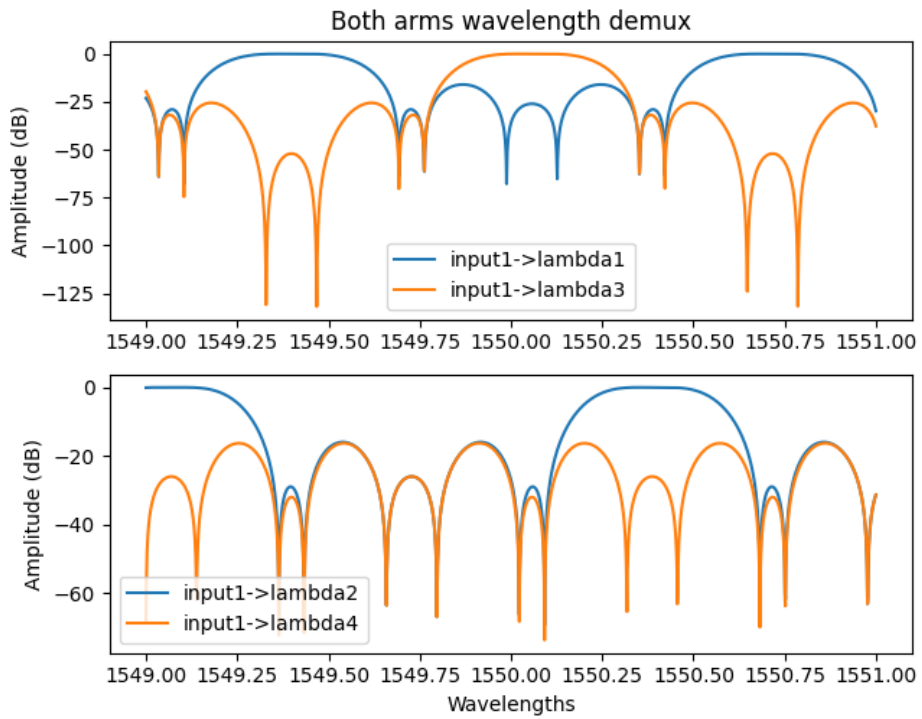


Figura 4.5.4: Respuesta del ROADM por pares en configuración de demultiplexado de 3 canales

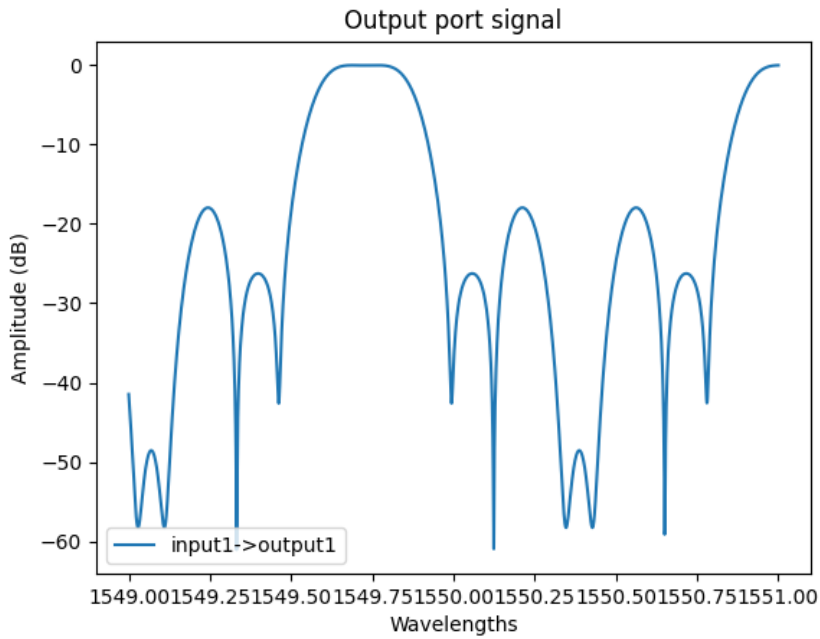


Figura 4.5.5: Respuesta del ROADM en el output en configuración de demultiplexado de 3 canales

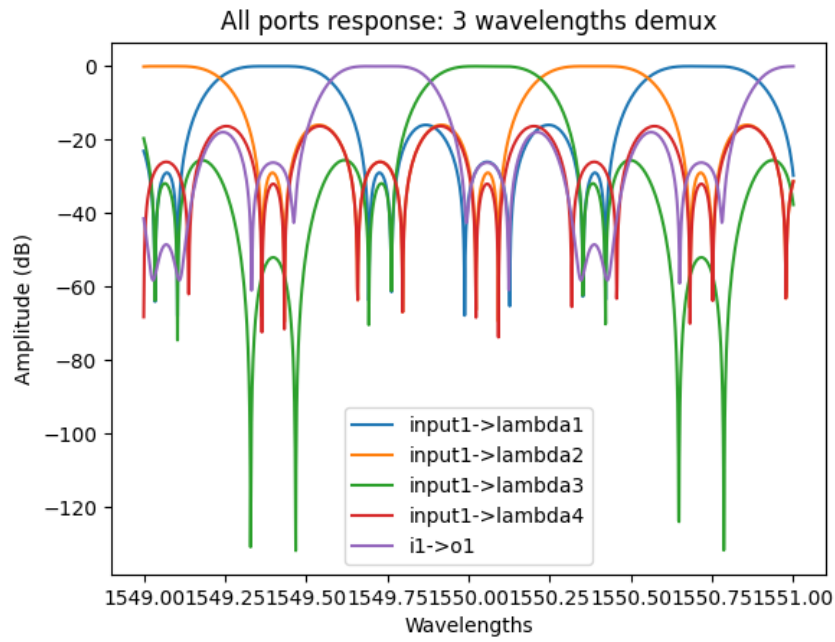


Figura 4.5.6: Respuesta del ROADM total en configuración de demultiplexado de 3 canales

Elemento	Fase Phase Shifter 1	Fase Phase Shifter 2
<i>MZI dual 1</i>	0 radianes	0 radianes
<i>MZI dual 2 superior</i>	π radianes	$\frac{3}{2}\pi$ radianes
<i>MZI dual 3 superior</i>	π radianes	$\frac{1}{2}\pi$ radianes
<i>MZI dual 2 inferior</i>	0 radianes	π radianes
<i>MZI dual 3 inferior</i>	0 radianes	π radianes
<i>MZI dual 4</i>	0 radianes	0 radianes

Tabla 4.5.2: Fases aplicadas para la configuración de demultiplexación de 3 canales

4.5.3. Multiplexado de 4 canales

Ahora las respuestas visualizadas son entrando desde los puertos **ADD** hacia el puerto **Output1**

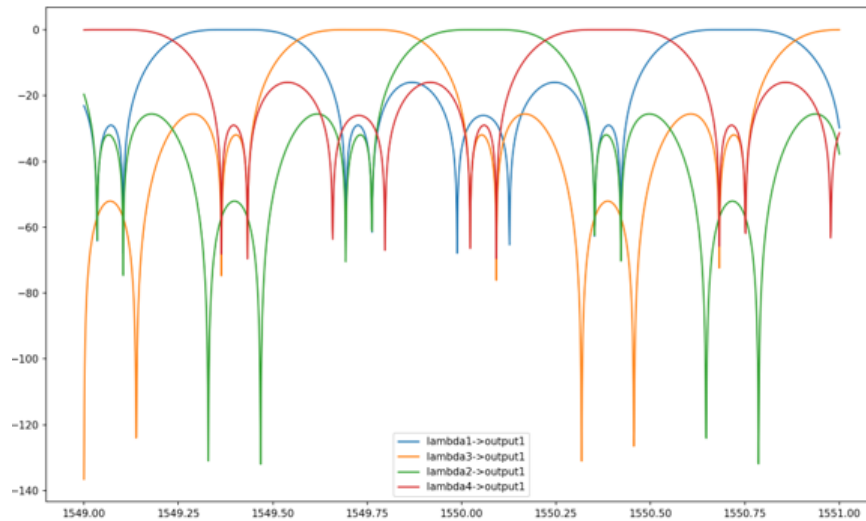


Figura 4.5.7: Respuesta del ROADM total en configuración de multiplexado de 4 canales

4.5.4. Cambio de asignación de canales a cada puerto

También es posible modificar los parámetros del ROADM para reorganizar los canales asignados a cada puerto ADD o DROP del circuito. Abajo se muestran tres respuestas de demultiplexado de 4 canales, cada una con una asignación de canales a los puertos ADD distintas. Esto se puede ver fijándose en los colores de las gráficas y en la leyenda asociada.

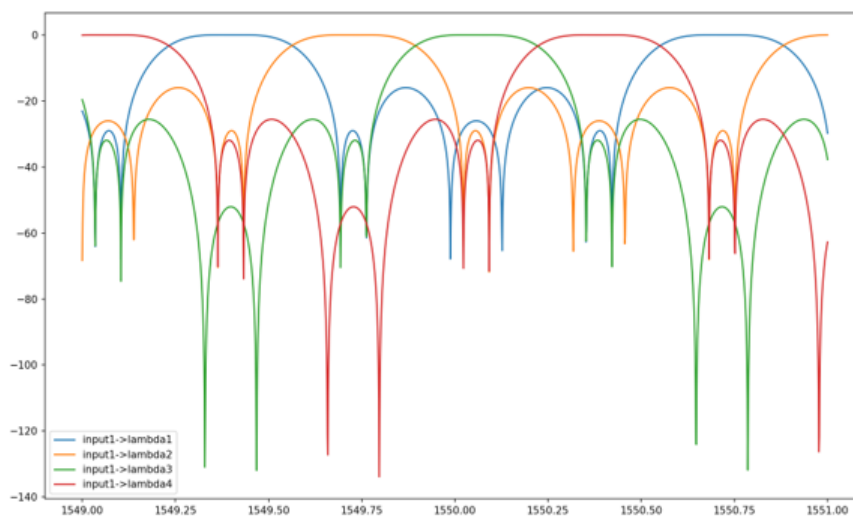


Figura 4.5.8: Configuración inicial asignación de canales en el ROADM

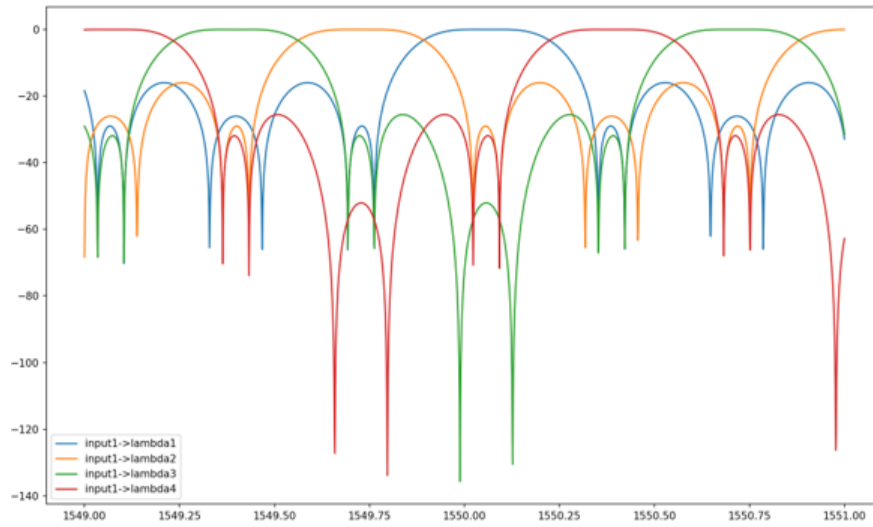


Figura 4.5.9: Primer cambio de asignación de canales en el ROADM

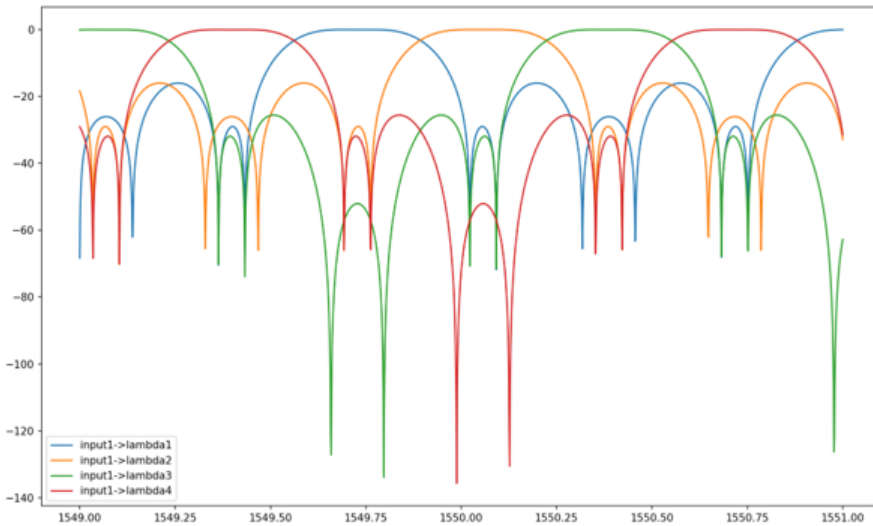


Figura 4.5.10: Segundo cambio de asignación de canales en el ROADM

Capítulo 5

Conclusiones y futuras líneas de trabajo

En esta sección se presentan las conclusiones obtenidas a partir de la preparación de este Trabajo de Final de Grado, además de resumir todos los conocimientos adquiridos durante su desarrollo. También se describen distintas propuestas de trabajo futuro en la línea de la fotónica programable y las FPPGAs.

5.1. Conclusiones

La fotónica integrada ha surgido como una solución tecnológica de altas prestaciones que destaca gracias a su alta velocidad y bajo consumo, y que es capaz de complementar y superar a la electrónica en aquellos sectores donde esta flaquea. En concreto la tecnología fotónica programable es un campo multidisciplinar en el que intervienen gran cantidad de sectores y multitud de áreas y tecnologías, como las telecomunicaciones, la electrónica, la programación, la ingeniería de microondas o la misma fotónica. Esto lo hace ideal para perfiles profesionales de ingenieros flexibles e instruidos en gran variedad de tecnologías, como es el Ingeniero de Telecomunicaciones.

Durante el desarrollo de este trabajo se ha comenzado aprendiendo sobre la fotónica en sus distintas vertientes y el funcionamiento de la FPPGA destacando los siguientes aspectos:

- El potencial de la fotónica como tecnología, y sus vertientes programable y de microondas
- Las características de la tecnología FPPGA y los filtros incluidos en su región de HPBs
- La importancia de los simuladores en el campo de la electrónica y la fotónica para diseñar y analizar dispositivos

En la segunda parte se han desarrollado los fundamentos teóricos en los que se basa el algoritmo de simulación propuesto para el ROADM de los nuevos chips de la FPPGA y los elementos principales:

- Desarrollo e integración de conceptos como las matrices de scattering y transferencia que permiten caracterizar circuitos fotónicos.

- Desarrollo de las unidades básicas programables y componentes que forman las mallas y filtros fotónicos incluidas en la FPPGA.
- Presentación del ROADM a simular, sus características y la importancia del filtro en el panorama tecnológico actual.

Con todos estos conocimientos, en la última sección se ha desarrollado el procedimiento para obtener los algoritmos de simulación del ROADM y sus estructuras básicas, además de presentar los resultados obtenidos de los simuladores:

- Modelaje de las unidades básicas programables y cálculo de la respuesta espectral.
- Estudio del MZI dual como componente del ROADM y desarrollo de su algoritmo de simulación.
- Estudio del ROADM y reestructuración de sus componentes para desarrollar su algoritmo de simulación.
- Presentación de los resultados y configuraciones atendiendo a las posibles aplicaciones del filtro.

En resumen, durante este trabajo se han adquirido conocimientos en una gran variedad de áreas tecnológicas como la fotónica integrada programable y se han potenciado las habilidades tecnológicas como la programación en Python, que ha demostrado la versatilidad de este lenguaje para ser empleado en multitud de campos. También se ha experimentado la dinámica de trabajo y el funcionamiento de una empresa tecnológica puntera como es iPronics, obteniendo conocimientos en multitud de ámbitos de la mano de profesionales del sector, y desarrollando mis habilidades como futuro profesional en el campo de la investigación y el desarrollo.

Por último, y lo más importante, durante este trabajo se ha conseguido diseñar algoritmos de simulación de filtros para fotónica programable de manera eficiente, lo que permitirá simular el ROADM y sus componentes en hardware empleando el Kit de Desarrollo de Software de iPronics, *iPronics Smartlight*. Esto habilitará a los usuarios la posibilidad de experimentar y desarrollar distintas configuraciones de los filtros sin emplear el hardware directamente y a los desarrolladores poder elaborar métodos y procedimientos para caracterizar y calibrar las componentes que forman el filtro ubicado en el propio hardware. Con todo esto se han logrado cumplir con los objetivos establecidos a inicios del trabajo.

5.2. Propuestas de trabajo futuro

A pesar del potencial y la versatilidad de las FPPGAs, actualmente se trata de una tecnología en desarrollo que puede ser mejorada en muchos aspectos de áreas muy distintas.

En la línea de lo mencionado anteriormente, una línea importante de trabajo futuro será, empleando el simulador desarrollado en este trabajo, diseñar los métodos de calibración y caracterización de las componentes que integran el ROADM físico, y probar dichos resultados en el hardware de iPronics. También sería adecuado realizar un análisis de tipo benchmark del potencial de los filtros ubicados en la región de HPBs en combinación con estructuras generadas en la malla fotónica para aplicaciones actuales en las telecomunicaciones, la computación fotónica y neuromórfica o la cuántica.

Por otro lado, centrándose más en la parte de hardware, ya que la fotónica está aumentando su influencia en el sector de la computación, sería adecuado desarrollar una sección similar a los HPBs que ubicar en el chip, o en una versión centrada en computación, que permita acceder a características más adecuadas para computación fotónica, o la reciente computación neuromórfica que está ganando mucha popularidad en la actualidad.

Por último, otra línea interesante sería desarrollar adaptaciones en la malla fotónica para ser compatible con paradigmas de computación cuántica. Una de las empresas más relevantes de computación cuántica en la actualidad es Xanadu, empresa canadiense que emplea un procesador fotónico programable para ello, y cuya malla fotónica presenta configuración rectangular. Dada la posibilidad de adaptar la malla hexagonal de la FPPGA a esquemas rectangulares, sería muy interesante evaluar la capacidad de este dispositivo para incluir este campo en sus posibles aplicaciones. Esto podría realizarse incluyendo una sección similar a los HPBs que se centre en generar el tipo de señales empleadas en computación cuántica, como pueden ser los pares de fotones entrelazados, e introducirlas en la malla fotónica.

Apéndice A

Anexo I: Códigos empleados para mostrar las respuestas espectrales

```
1 """
2 Simulation of a Dual Mach Zehnder Interleaver
3
4 In this script is checked that the spectral response of the DualMZI is
5 correct and behaves properly when changing the delta_length. Also, two
6 instances of DualMZI will be declared to check if the spectral response
7 of both devices in series is proper to implement the ROADM
8 """
9
10 import matplotlib.pyplot as plt
11 import numpy as np
12 from iPrronics.HighPerformanceBlocks.dual_mzi_interleaver import
13     ↪ DualMziInterleaver
14
15 length = 800e-6 # 800um
16 test_dual = DualMziInterleaver(length)
17
18 wavelengths = np.linspace(1549, 1551, 1001) * 1e-9
19 test_dual.set_wavelengths(wavelengths)
20
21 test_dual.pucs[0].set_coupling_factor_phase(0.9248, 0)
22 test_dual.pucs[1].set_coupling_factor_phase(0.280, 0)
23 test_dual.pucs[2].set_coupling_factor_phase(0.5, 0)
24
25
26 H = test_dual.compute_response()
27
28 plt.figure(1)
29
30 amplitude_response_db_bar_0 = 20 * np.log10(abs(H[0, 0, :]))
31 amplitude_response_db_cross_0 = 20 * np.log10(abs(H[0, 1, :]))
32 amplitude_response_db_bar_1 = 20 * np.log10(abs(H[1, 1, :]))
33 amplitude_response_db_cross_1 = 20 * np.log10(abs(H[1, 0, :]))
34 plt.subplot(211)
35 plt.plot(wavelengths / 1e-9, amplitude_response_db_cross_0, label="cross_0")
36 plt.plot(wavelengths / 1e-9, amplitude_response_db_bar_0, label="bar_0")
37 plt.ylabel("Amplitude (dB)")
38 plt.title("DualMZI amplitude response")
39 plt.legend()
```

```

40 plt.subplot(212)
41 plt.plot(wavelengths / 1e-9, amplitude_response_db_cross_1, label="cross_1")
42 plt.plot(wavelengths / 1e-9, amplitude_response_db_bar_1, label="bar_1")
43 plt.xlabel("Wavelengths")
44 plt.ylabel("Amplitude (dB)")
45 plt.legend()
46 plt.show()
47
48 length = 400e-6
49 test_dual_2 = DualMziInterleaver(length)
50
51 wavelengths = np.linspace(1549, 1551, 1001) * 1e-9
52
53 test_dual_2.set_wavelengths(wavelengths)
54
55 test_dual_2.pucs[0].set_coupling_factor_phase(0.9248, 0)
56 test_dual_2.pucs[1].set_coupling_factor_phase(0.290, 0)
57 test_dual_2.pucs[2].set_coupling_factor_phase(0.5, 0)
58
59
60 H_2 = test_dual_2.compute_response()
61
62 output_response_1 = 20 * np.log10(abs(H[0, 1, :] * H_2[1, 1, :]))
63
64
65 plt.figure(2)
66 amplitude_response_db_bar_1_2 = 20 * np.log10(abs(H_2[1, 1, :]))
67 plt.subplot(311)
68 plt.plot(wavelengths / 1e-9, amplitude_response_db_cross_1, label="mzi
        ↪ delta_length")
69 plt.ylabel("Amplitude (dB)")
70 plt.legend()
71 plt.title("Amplitude response: 2 DualMZI in series with different FSR")
72 plt.subplot(312)
73 plt.plot(wavelengths / 1e-9, amplitude_response_db_bar_1_2, label="mzi
        ↪ delta_length*2")
74 plt.ylabel("Amplitude (dB)")
75 plt.legend()
76 plt.subplot(313)
77 plt.plot(wavelengths / 1e-9, output_response_1, label="output response")
78 plt.xlabel("Wavelengths")
79 plt.ylabel("Amplitude (dB)")
80 plt.legend()
81 plt.show()

```

Código A.0.1: Código utilizado para generar las gráficas 4.3.3 y 4.3.4 que visualizan las respuestas espectrales de los MZI duales

```

1
2 """
3 Simulation of a Reconfigurable Optical Add-Drop Multiplexer (ROADM)
4
5 In this script is checked that the spectral response of the ROADM is
6 correct and behaves properly when changing the delta_length. The default
7 parameters of the ROADM generate a 4 wavelength demux. For other
8 functionalities, the phase shift of the different DualMZI should
9 be changed
10 """
11
12 import matplotlib.pyplot as plt
13 import numpy as np
14 from iPrronics.HighPerformanceBlocks.roadm import ROADM_4_channels
15
16 delta_length = 436e-6
17 test_ROADM = ROADM_4_channels(delta_length)
18
19 wavelengths = np.linspace(1549, 1551, 2001) * 1e-9
20
21 test_ROADM.set_wavelengths(wavelengths)
22
23 # ROADM settings
24
25 # 1. Coupling factors
26 for k in range(4):
27     test_ROADM.dualMZI_delta[k].pucs[0].set_coupling_factor_phase(0.9248, 0)
28     test_ROADM.dualMZI_delta[k].pucs[1].set_coupling_factor_phase(0.280, 0)
29     test_ROADM.dualMZI_delta[k].pucs[2].set_coupling_factor_phase(0.5, 0)
30
31 for k in range(2):
32     test_ROADM.dualMZI_delta_two[k].pucs[0].set_coupling_factor_phase(0.9248, 0)
33     test_ROADM.dualMZI_delta_two[k].pucs[1].set_coupling_factor_phase(0.280, 0)
34     test_ROADM.dualMZI_delta_two[k].pucs[2].set_coupling_factor_phase(0.5, 0)
35
36 # 2. Phase shifters
37 test_ROADM.dualMZI_delta_two[0].phase_shifters[0].set_phase(0)
38 test_ROADM.dualMZI_delta_two[0].phase_shifters[1].set_phase(0)
39
40 test_ROADM.dualMZI_delta[0].phase_shifters[0].set_phase(np.pi)
41 test_ROADM.dualMZI_delta[0].phase_shifters[1].set_phase(np.pi / 2 * 3)
42
43 test_ROADM.dualMZI_delta[1].phase_shifters[0].set_phase(np.pi)
44 test_ROADM.dualMZI_delta[1].phase_shifters[1].set_phase(np.pi / 2)
45
46 test_ROADM.dualMZI_delta[2].phase_shifters[0].set_phase(0)
47 test_ROADM.dualMZI_delta[2].phase_shifters[1].set_phase(0)
48
49 test_ROADM.dualMZI_delta[3].phase_shifters[0].set_phase(0)
50 test_ROADM.dualMZI_delta[3].phase_shifters[1].set_phase(np.pi)
51
52
53 # Compute response
54 H = test_ROADM.compute_response()
55
56 amplitude_response_db_i1_o1 = 20 * np.log10(abs(H[0, 0, :]))
57 amplitude_response_db_i1_o2 = 20 * np.log10(abs(H[0, 1, :]))
58 amplitude_response_db_i1_lambda1 = 20 * np.log10(abs(H[0, 2, :]))
59 amplitude_response_db_i1_lambda2 = 20 * np.log10(abs(H[0, 3, :]))
60 amplitude_response_db_i1_lambda3 = 20 * np.log10(abs(H[0, 4, :]))
61 amplitude_response_db_i1_lambda4 = 20 * np.log10(abs(H[0, 5, :]))

```

```

62
63 amplitude_response_db_i2_o1 = 20 * np.log10(abs(H[1, 0, :]))
64 amplitude_response_db_lambda1_o1 = 20 * np.log10(abs(H[2, 0, :]))
65 amplitude_response_db_lambda2_o1 = 20 * np.log10(abs(H[3, 0, :]))
66 amplitude_response_db_lambda3_o1 = 20 * np.log10(abs(H[4, 0, :]))
67 amplitude_response_db_lambda4_o1 = 20 * np.log10(abs(H[5, 0, :]))
68
69
70 # Response visualization
71
72 plt.figure(1)
73
74 plt.subplot(211)
75 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda1,
76          ↪ label="input1->lambda1")
77 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda3,
78          ↪ label="input1->lambda3")
79 plt.title("Both arms wavelength demux")
80 plt.ylabel("Amplitude (dB)")
81 plt.legend()
82
83 plt.subplot(212)
84 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda2,
85          ↪ label="input1->lambda2")
86 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda4,
87          ↪ label="input1->lambda4")
88 plt.xlabel("Wavelengths")
89 plt.ylabel("Amplitude (dB)")
90 plt.legend()
91 plt.show()
92
93 plt.figure(2)
94 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda1,
95          ↪ label="input1->lambda1")
96 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda2,
97          ↪ label="input1->lambda2")
98 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda3,
99          ↪ label="input1->lambda3")
100 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_lambda4,
101          ↪ label="input1->lambda4")
102
103 plt.title("4 wavelengths demux")
104 plt.xlabel("Wavelengths")
105 plt.ylabel("Amplitude (dB)")
106 plt.legend()
107 plt.show()
108
109 plt.figure(3)
110 plt.plot(wavelengths / 1e-9, amplitude_response_db_i1_o1,
111          ↪ label="input1->output1")
112
113 plt.title("Through port signal")
114 plt.xlabel("Wavelengths")
115 plt.ylabel("Amplitude (dB)")
116 plt.legend()
117 plt.show()
118
119 plt.figure(4)

```

```
112
113 plt.plot(wavelengths / 1e-9, amplitud_response_db_i1_lambda1,
           ↪ label="input1->lambda1")
114 plt.plot(wavelengths / 1e-9, amplitud_response_db_i1_lambda2,
           ↪ label="input1->lambda2")
115 plt.plot(wavelengths / 1e-9, amplitud_response_db_i1_lambda3,
           ↪ label="input1->lambda3")
116 plt.plot(wavelengths / 1e-9, amplitud_response_db_i1_lambda4,
           ↪ label="input1->lambda4")
117 plt.plot(wavelengths / 1e-9, amplitud_response_db_i1_o1, label="i1->o1")
118
119 plt.title("All ports response: 4 wavelengths demux")
120 plt.xlabel("Wavelengths")
121 plt.ylabel("Amplitude (dB)")
122 plt.legend()
123 plt.show()
```

Código A.0.2: Código utilizado para generar las gráficas 4.3.3 y 4.3.4 que visualizan las respuestas espectrales de las distintas configuraciones del ROADM

BIBLIOGRAFÍA

- [1] J. Capmany y M. A. Muriel. «A New Transfer Matrix Formalism for the Analysis of Fiber Ring Resonators: Compound Coupled Structures for FDMA Demultiplexing». En: *Journal of Lightwave Technology* 8.12 (1990).
- [2] J. Capmany y D. Pérez. *Programmable Integrated Photonics*. Oxford University Press, 2020.
- [3] P. Cheben, R. Soref, D. Lockwood y G. Reed. «The Achievements and Challenges of Silicon Photonics». En: *Advances in Optical Technologies* 1.1 (2008).
- [4] E. Commission. *Digital Strategy, Photonics*. 2020.
- [5] S. W. Jones. *Introduction to Integrated Circuit Technology. Fifth edition*. 2012.
- [6] I. P. Kaminow, T. Li y A. E. Willner. *Optical Fiber Telecommunications V. Volume B*. Elsevier Inc., 2008.
- [7] J. Lyke, C. G. Christodolou, G. A. Vera y A. H. Edwards. «An Introduction to Reconfigurable Systems». En: *IEEE* ().
- [8] D. Pérez, I. Gasulla, P. D. Mahapatra y J. Capmany. «Principles, fundamentals and applications of Programmable Integrated Photonics». En: 30 (2015).
- [9] S. Saleh y M. Teich. *Fundamentals of Photonics. Third Edition*. Wiley Series in Pure y Applied Optics.
- [10] A. J. Seeds y K. J. Williams. «Microwave Photonics». En: *Journal of Lightwave Technology* 24.12 (2006).
- [11] S. M. Trimberger. «Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology». En: 30 (2015).
- [12] C. Vagionas et al. «Lossless 1×4 Silicon Photonic ROADM Based on a Monolithic Integrated Erbium Doped Waveguide Amplifier on a Si₃N₄ Platform». En: *Journal of Lightwave Technology* 40.5 ().
- [13] M. G. Zubilanga. «Experimental demonstration and benchmarking of Microwave Photonic applications in a Field Programmable Photonic Gate Array». En: (2022).