

*To my tutor, sensei and friend Jose Jaramillo who always
believed in me and gave me a hand when I needed it.*

*To Orlando Bedoya, my best friend ever who always supported
me in difficult times and who encouraged me to follow my dreams
no matter what. For you I am here now.*

*To Santiago Restrepo, David Jimenez, Orlando Savogal and
Fabián Muñoz, who were always unconditional and became an
active part of this research.*

*To my friend René Gomez, who help me in getting one of the
most incredible opportunities in my life.*

*To all the members of Sirius and all of the people that helped
directly or indirectly in this work, Thank You!*

Índice

1	Definición de la Investigación	8
1.1	Resumen	8
1.2	Planteamiento del Problema	9
1.3	Justificación	9
1.4	Revisión Bibliográfica	10
1.5	Objetivos	11
1.5.1	Objetivo General	11
1.5.2	Objetivos Específicos	11
1.6	Hipótesis	12
1.7	Metodología	12
1.7.1	Etapa 1	12
1.7.2	Etapa 2	13
1.8	Resultados	13
2	Fundamentación Teórica	14
2.1	Introduction	14
2.2	Ecuaciones de Maxwell	14
3	Caracterización de Anillos Resonadores	15
3.1	Introducción	15
3.2	Teoría de Acoplamiento	16
3.2.1	Modelo Teórico de Anillo Resonador	16
3.2.2	Relación entre Coeficientes de Acoplamiento	20
3.2.3	Filtro Notch	22
3.2.4	Parámetros	23
3.3	Simulaciones	25
3.3.1	Metodología	26
3.3.2	Diseño del Resonador	26
3.3.3	Lumerical	28
3.3.4	Meep	33
3.3.5	Conclusiones y Trabajos Futuros	40
4	Redes de Interconexión	42

4.1	PhoenixSim	42
4.1.1	Dispositivos de Interconexión	43
4.1.2	Benchmarks	47
4.2	Simulaciones	49
4.2.1	Metodología	50
4.2.2	Resultados	52
4.2.3	Conclusiones y Trabajos Futuros	64
A	Scripts	66
A.1	Meep	66
A.1.1	Filtro Notch	66
A.1.2	Filtro AddDrop	67
A.2	Lumerical	70
A.2.1	Filtro Notch	70
A.2.2	Filtro AddDrop	71

Índice de figuras

3.1	Modelo de un Anillo Resonador	17
3.2	Contribución Onda 1 Vuelta	18
3.3	Contribución Onda 2 Vueltas	19
3.4	Modos Resonantes. Fuente[5]	23
3.5	Índice de grupo. Fuente[17]	27
3.6	SetUp del filtro Notch en Lumerical FDTD	30
3.7	Transmitancia simulación 3D FDTD Filtro Notch	30
3.8	SetUp del filtro AddDrop en Lumerical FDTD	31
3.9	Transmitancia en el $puerto_d$ de valores teóricos, simulación MODE y FDTD del Filtro AddDrop	32
3.10	Transmitancia en el $puerto_i$ de valores teóricos, simulación MODE y FDTD del Filtro AddDrop	32
3.11	Transmitancia simulación 3D FDTD Filtro AddDrop	33
3.12	Geometría Filtro Notch.	36
3.13	Transmitancia Filtro Notch	38
3.14	Filtro Notch 3d vista [X,Y], [Y,Z] y [X,Z] respectivamente. (a)(b)(c) Material Dieléctrico. (d)(e)(g) Campo eléctrico E_z	39
3.15	Transmitancia Filtro Notch 3D	40
4.1	Vista General PhoenixSim. Fuente [7]	43
4.2	Caracterización: Guía de Onda.	44
4.3	Caracterización: Cruce de Guía de Onda.	46
4.4	(a). Cambio de estado de resonancia on-off al aplicar un voltaje Fuente [16]. (b). Caracterización: Anillo Resonador.	47
4.5	Patrones sintéticos. Fuente [15]	48
4.6	Comparación Consumo de Energía, PkgSize: 512B	59
4.7	Comparación Latencia, PkgSize: 512B	60
4.8	Comparación Consumo de Energía, PkgSize: 65kB	62
4.9	Comparación Latencia, PkgSize: 65kB	63

Índice de tablas

3.1	Equipo para simulación en Lumerical. Codename: Amelie	25
3.2	Equipo para simulación en Meep. Codename: Arwen	25
3.3	Configuración deseada para el filtro	27
3.4	Parámetros	35
4.1	Caracterización Guía de Onda.	44
4.2	Caracterización Guía de Onda.	45
4.3	Equipo para simulación en PhoenixSim. Codename: mateubun- tuSSD	50
4.4	Parámetros de guía de onda. [31] [13]	51
4.5	Parámetros de guía de onda doblada. [31]	51
4.6	Parámetros de cruce de guía de onda. [6]	51
4.7	Parámetros de anillos resonadores.	52
4.8	Datos Simulación App: Neighbor, PkgSize: 512B	53
4.9	Datos Simulación App: Random, PkgSize: 512B	53
4.10	Datos Simulación App: BitReverse, PkgSize: 512B	54
4.11	Datos Simulación App: HotSpot, PkgSize: 512B	54
4.12	Datos Simulación App: Tornado, PkgSize: 512B	55
4.13	Datos Simulación App: Random, PkgSize: 65536B	55
4.14	Datos Simulación App: Neighbor, PkgSize: 65536B	56
4.15	Datos Simulación App: BitReverse, PkgSize: 65536B	56
4.16	Datos Simulación App: HotSpot, PkgSize: 65536B	57
4.17	Datos Simulación App: Tornado, PkgSize: 65536B	57
4.18	Estadísticos para Diferencia de Medias. Fuente: [14]	58
4.19	My caption here	58
4.20	Datos promedio Consumo de Energía (J), PkgSize: 512B	59
4.21	Datos promedio Latencia (μs), PkgSize: 512B	59
4.22	Resultados Prueba Hipótesis para la variable Consumo de En- energía, PkgSize: 512B	60
4.23	Resultados Prueba Hipótesis para la variable Latencia, PkgSize: 512B	61
4.24	Datos promedio Consumo de Energía (J), PkgSize: 65kB	61
4.25	Datos promedio Latencia (μs), PkgSize: 65kB	62

4.26	Resultados Prueba Hipótesis para la variable Consumo de Energía, PkgSize: 65kB	63
4.27	Resultados Prueba Hipótesis para la variable Latencia, PkgSize: 65kB	64

Índice de Scripts

3.1	Parámetros Filtro Notch.	34
3.2	Geometría y Materiales Filtro Notch.	36
4.1	Implementación de la guía de onda en PhoenixSim.	45
A.1	Filtro Notch (Fuente Gaussiana) en Meep.	66
A.2	Filtro AddDrop (Fuente Gaussiana) en Meep.	67
A.3	Generar gráficas de Transmitancia”	69
A.4	Shell script de invocación con OpenMPI	70
A.5	Generar la estructura de anillo 3d FDTD	70
A.6	Obtener datos de la simulación 3d FDTD	71
A.7	Generar gráficas de Transmitancia teórica MODE FDTD	71

Capítulo 1

Definición de la Investigación

1.1 Resumen

Las tendencias actuales de los procesadores muestran que en un periodo corto de tiempo, alcanzaremos los cientos de núcleos en un solo chip. A medida que la cantidad de estos núcleos aumenta, los requerimientos de ancho de banda de las redes de interconexión que permiten la comunicación interna entre estos y hacia la memoria, se incrementa. A medida que se incrementa el ancho de banda en sistemas de interconexión electrónicos, la latencia y la disipación de poder, se ven impactadas considerablemente, por lo que dicha solución se vuelve no viable.

La fusión del campo de la fotónica con la nanotecnología denominado nanofotónica, ha permitido el desarrollo de dispositivos basados en silicio de altas prestaciones y bajo consumo, ya que pueden ser producidos usando las técnicas ya existentes de manufacturación de semiconductores, y gracias a que el silicio actualmente es utilizado como el componente base en la mayoría de circuitos, es posible crear dispositivos híbridos en los cuales se integran componentes tanto electrónicos como ópticos en un solo microchip. Tomando en cuenta estos últimos avances, las redes de interconexión ópticas en un chip o NoCs nano ópticas sobre silicio, ya ha sido conceptualizada, permitiendo superar las actuales limitaciones de su equivalente electrónico en los chips de multiprocesamiento o CMP.

1.2 Planteamiento del Problema

Las redes de interconexión electrónicas, tienen un impacto directo en la limitación de potencia, ancho de banda y latencia de los chips multiprocesadores (CMPs) actuales. Estas limitaciones sumadas a la inhabilidad de escalar eficientemente a cientos de núcleos, la presenta como una solución poco viable a largo plazo.

1.3 Justificación

Para lograr escalar eficientemente los sistemas de multiprocesamiento, las tecnologías actuales deben superar grandes dificultades, tanto en la complejidad adicional que representa la programación en paralelo, las limitaciones de ancho de banda y en la minimización de la cantidad de potencia que es desperdiciada en el sistema de comunicación. En esta última en especial, se evidencia el límite al que están llegando las tecnologías electrónicas actuales, al consumir casi la mitad [19] de la potencia dinámica de todo el sistema, haciendo de vital importancia la exploración de otras alternativas.

El empleo de la tecnología de punta que usa la nanofotónica sobre silicio, se presenta como una solución prometedora a este problema, ya que no solo permite el flujo de grandes cantidades de datos en una misma línea de transmisión, sino que tiene una facilidad de integración con la microelectrónica actual manteniendo al mismo tiempo los bajos costes de fabricación, gracias a que se pueden reutilizar las técnicas de manufactura de semiconductores tradicionales. Es por esto que el tema es foco de investigación no solo de grandes compañías microelectrónicas (Intel, IBM, HP, etc) sino también de la comunidad académica (Columbia, Purdue, MIT entre otras), quienes han logrado demostrar experimentalmente las piezas fundamentales que permitirían su materialización en un futuro no muy lejano.

Debido a que los dispositivos ópticos son fundamentalmente diferentes de las tecnologías de interconexión electrónicas convencionales, se requieren nuevas metodologías y herramientas que permitan a los arquitectos de computadores, evaluar el rendimiento holístico del sistema a través de mediciones precisas en los aspectos de capacidad de procesamiento, consumo de potencia y escalabilidad. Todo esto, incorporando de forma acertada las diferencias físicas en cuanto al comportamiento en la fotónica [9].

Por otro lado, y aunque de forma relativamente reciente, también se ha destacado la importancia del fortalecimiento y aplicación de este tipo de tecnologías convergentes en el contexto nacional. En 2004, la nanotecnología fue recono-

cida por Colciencias como una de las 8 áreas estratégicas del conocimiento para impulsar el desarrollo competitivo del país. En el documento "Visión Colombia II Centenario: 2019" se confirma la continuidad del apoyo institucional y financiero a las iniciativas relacionadas con dichas áreas. Finalmente, es mencionada también dentro del "Plan Nacional de Desarrollo Científico, Tecnológico y de Innovación 2007-2019" de Colciencias y el Departamento Nacional de Planeación, como un área perteneciente a la "Nueva Ciencia" y "Tercera Revolución Industrial" en donde se debe disminuir la brecha tecnológica con respecto a otros países.

1.4 Revisión Bibliográfica

Antecedentes

La naturaleza creativa inherente al ser humano, le ha impulsado a través de los tiempos hacia la búsqueda y análisis de información. El desarrollo de las tecnologías orientadas no solo a la manipulación de dichos datos, sino a la facilidad en el intercambio de los mismos, se han convertido en parte indispensable de la sociedad actual, la cual exige cada vez, mayores capacidades y mejores tiempos de respuesta en los mismos. Para satisfacer estos requisitos, y como consecuencia directa, la industria de los microprocesadores ha realizado constantes avances desde los años 70, en donde era más que suficiente una unidad de procesamiento (single-core). A partir de allí, se ha llegado a los sistemas actuales, en donde varios núcleos de una misma unidad (multi-core) realizan el trabajo. La tendencia de dicha industria, en un futuro a mediano-largo plazo, es el incremento de estos núcleos al orden de cientos o miles, en donde su cantidad tiene una proyección de crecimiento en un factor de 2x cada 18 meses [19], dando surgimiento a una nueva ley similar a la ley de Moore para los transistores.

Sin embargo, y para preservar las ventajas esperadas al tener cada vez más núcleos de procesamiento, se deben no solo mantener los bajos costos de fabricación, sino superar tres grandes dificultades [2] relacionadas con: el aumento en la complejidad por la programación en paralelo, el limitado ancho de banda hacia los dispositivos de entrada/salida y la disipación de poder de la red de interconexión entre núcleos, siendo esta la responsable del consumo de alrededor del 50% [4] del poder total del chip.

Los grandes avances sobre la nueva tecnología de nanofotónica sobre silicio, han demostrado los componentes necesarios [7] para la creación de redes que permitan conectar componentes tanto dentro como fuera del chip, no solo aprovechando las altas velocidades de transmisión y bajo consumo de potencia, sino también conservando el uso de las técnicas ya existentes de manufacturación de semi-

conductores. Esto facilita la integración con la tecnología actual, manteniendo viable los costes de fabricación a nivel masivo, comparado con los sistemas ópticos convencionales.

No obstante, el uso de nanofotónica sobre silicio aplicado a las redes de interconexión debe seguir nuevas metodologías y herramientas que permitan un diseño preciso y acertado en un tiempo apropiado [8]. Dentro de las herramientas disponibles, se destacan las siguientes:

El proyecto PhoenixSim [9] (Photonic and Electronic Network Integration and eXecution Simulator) de la Universidad de Columbia, ha capturado las características físicas los componentes fotónicos y electrónicos usados en este tipo de redes, sobre un software de simulación que soporta la interacción de decenas de núcleos. Esto permite no solo la simulación y comparación de redes de interconexión en ambos dominios, sino también la creación de sistemas híbridos que aprovechen las ventajas que ofrece cada uno.

Las plataformas de prototipado BEE3 [11], desarrolladas en la Universidad de Stanford, permiten la completa emulación de CMPs usando las redes de interconexión eléctrica convencionales. En este sistema, Watts [?] propone la implementación de módulos de lógica que presenten un comportamiento equivalente a los dispositivos de nanofotónica sobre silicio, permitiendo así alcanzar tiempos de simulación que permitan una evaluación dinámica de múltiples arquitecturas NoCs.

1.5 Objetivos

1.5.1 Objetivo General

Simular y evaluar redes de interconexión nanofotónicas híbridas sobre silicio para chips multiprocesadores.

1.5.2 Objetivos Específicos

- Caracterizar los componentes básicos de los dispositivos que conforman la capa de interconexión en las redes nanofotónicas sobre silicio.
- Comprender los parámetros que gobiernan, a nivel físico, los dispositivos de una red de interconexión nanofotónica para su manipulación conciente en un nivel de abstracción superior o de sistema.

- Explorar las topologías de interconexión más usadas en los CMPs electrónicos actuales.
- Simular cada una de las topologías en los dominios electrónico y nanofotónico híbrido sobre silicio.
- Evaluar y comparar las redes en términos de latencia y potencia.

1.6 Hipótesis

¿Son las redes de interconexión nanofotónicas híbridas para las topologías Mesh y Torus más eficientes en términos de consumo de potencia y latencia que sus pares electrónicos?

1.7 Metodología

Este proyecto se planteó como una investigación aplicada donde se buscaba encontrar soluciones del estado del arte en el área de arquitectura de computadores para la problemática del consumo de recursos de las redes de interconexión electrónicas actuales en los chips multiprocesadores.

Resultó entonces conveniente la utilización de una metodología estructural, en la cual se identificaron 2 grandes etapas:

La primera corresponde a la visualización del problema desde el punto de vista del comportamiento físico de los dispositivos que componen la red de interconexión nanofotónica. La segunda tiene un enfoque con un nivel de abstracción superior, extrayendo sólo algunos parámetros de la primera etapa como base para un análisis holístico de rendimiento, latencia y potencia del sistema.

La importancia de realizar ambas etapas radica en el conocimiento adquirido en la primera permite una manipulación conciente de los parámetros que son insumo de la segunda.

1.7.1 Etapa 1

- Fase 1: Base Teórica En esta etapa se hizo un estudio de los fundamentos teóricos tanto físicos como matemáticos que rigen el comportamiento

de los dispositivos que soportan los componentes que forman una red de interconexión nanofotónica sobre silicio.

- Fase 2: Caracterización de los componentes pasivos y activos de las redes nanofotónicas. Es necesario analizar el funcionamiento de los siguientes componentes nanofotónicos sobre silicio: guías de onda recta, guías de onda dobladas, cruces de guía de onda y anillos resonadores.

Finalmente, se ejecutarán las simulaciones correspondientes a 2 de los elementos activos más relevantes que emplean anillos resonadores: filtro Notch y filtro AddDrop.

En esta etapa, se usarán 2 softwares de simulación que correrán sobre dos servidores especializados para este propósito.

1.7.2 Etapa 2

- Fase 3: Análisis de Topologías de Interconexión Se analizarán las topologías de interconexión electrónicas actuales que se desean evaluar en el alcance de este proyecto: Mesh y Torus en el dominio de los CMPs.

Una vez obtenida la información de las topologías en el dominio electrónico, se procederá a simular estas topologías híbridas y se evaluará su rendimiento midiendo los resultados en términos de potencia y latencia.

1.8 Resultados

Debido a la estructura metodológica sobre la que se desarrolló esta investigación, los resultados se presentarán en los 3 capítulos siguientes, correspondientes a los resultados obtenidos en cada una de las fases mencionadas.

Capítulo 2

Fundamentación Teórica

2.1 Introduction

La parte inicial de la investigación consistió en la búsqueda y apropiación de las leyes físicas y la formulación matemática que rige el comportamiento de las ondas electromagnéticas y su propagación en un medio.

2.2 Ecuaciones de Maxwell

EQUACIONES DE MAXWELL EN FORMA DIFERENCIAL:

$$\begin{aligned}\nabla \cdot \mathbf{D} &= \rho \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} &= \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}\end{aligned}$$

Capítulo 3

Caracterización de Anillos Resonadores

3.1 Introducción

Como se vio en el capítulo 2, el anillo resonador es el componente principal de los dispositivos activos usados en Silicon Photonics. Por esta razón se analizó la teoría detrás de su funcionamiento (sección 3.2) y se simuló su comportamiento (3.3) en dos situaciones particulares:

- Filtro Notch (sección 3.2.3) el cuál actúa como un inhibidor o atenuador de frecuencias puntuales con un factor de calidad alto. Este elemento es la base de los filtros y parte de los dispositivos de modulación y de detección.
- Filtro AddDrop el cual permite cambiar de una guía a otra un conjunto de frecuencias específicas. Los switches y routers -elementos esenciales dentro de una red de interconexión de Silicon Photonics- están formados por la unión de varios filtros AddDrop tuneables.

3.2 Teoría de Acoplamiento

3.2.1 Modelo Teórico de Anillo Resonador

En la figura 3.1 se muestra el esquema del caso genérico de un anillo resonador con 2 regiones de acoplamiento representadas por las líneas punteadas. Por simplicidad, el modelo asume que no hay pérdidas por acoplamiento (3.11a) y se ignoran los efectos de reflexión dentro de la guía (sólo se asumen ondas en el sentido de la propagación). El modelamiento teórico es muy similar al resonador Fabry-Perot descrito en [30].

Cada región del anillo tiene asociados unos parámetros principales que definen su comportamiento. Estos son:

- Coeficientes de acoplamiento $(\kappa_1, \kappa_1', \kappa_2, \kappa_2')$,
- Coeficientes de transmisión (t_1, t_1', t_2, t_2') (en la sección 3.2.2 se verá la relación que existe entre los 8 coeficientes).
- Constante de propagación β del modo circulante.

$$\beta = \frac{2\pi}{\lambda} n_{eff} \quad (3.1)$$

Valor complejo donde $\Re\{\beta\}$ es la constante de fase y $\Im\{\beta\}$ representa las pérdidas por propagación dentro del anillo.

- Radio r y perímetro ($L = 2\pi r$) del anillo.

Para que el dispositivo entre en resonancia, el desfase de la onda después de un viaje completo al redor del anillo debe ser un múltiplo entero de 2π

$$\beta L = 2\pi M \quad (3.2)$$

Donde M es llamado el número de modo (Figura 3.4).

La potencia de la onda que se ve en el $puerto_t$ está dada por la porción de la onda incidente que atraviesa la guía más las $N \rightarrow \infty$ contribuciones que se dan por la otra parte de la onda que se acopló en el anillo (ec. 3.3). Cada una de las contribuciones depende del número de viajes completos que realice la onda acoplada antes de volver a salir a la guía superior.

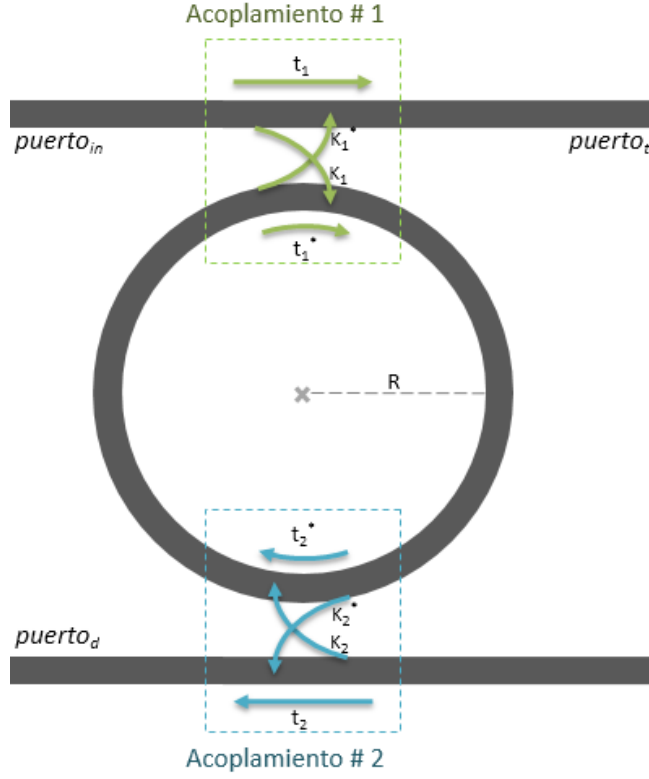


Figura 3.1: Modelo de un Anillo Resonador

$$E_t = E_i t_1 + \text{Contrib}_{M=1L} + \text{Contrib}_{M=2L} + \dots + \text{Contrib}_{M=\infty L} \quad (3.3)$$

- Contribución después de una vuelta: $\text{Contrib}_{M=1L}$

El fasor escalar $\alpha e^{-j\beta L}$ contiene la información sobre la amplitud de la atenuación debido a efectos de dispersión y a la curvatura de la onda ($\alpha = 1$ si no hay pérdida) y la fase de la onda que ha recorrido una distancia L . Por lo tanto, al dar una vuelta ($1L$), la propagación de la onda queda expresada como $\alpha e^{-j\beta L}$.

En su recorrido, la onda que da una vuelta completa (Figura 3.2) pasa por 3 regiones de interés. En la primera región (acoplamiento 1) una porción (dada por el coeficiente de acoplamiento κ_1) entra desde la guía recta hacia el anillo. En la segunda región (acoplamiento 2), una porción (proporcional al coeficiente de transmisión t_2) continúa su viaje al interior del anillo. Finalmente, en la tercera región (acoplamiento 1) sólo una parte de la onda (coeficiente de acoplamiento

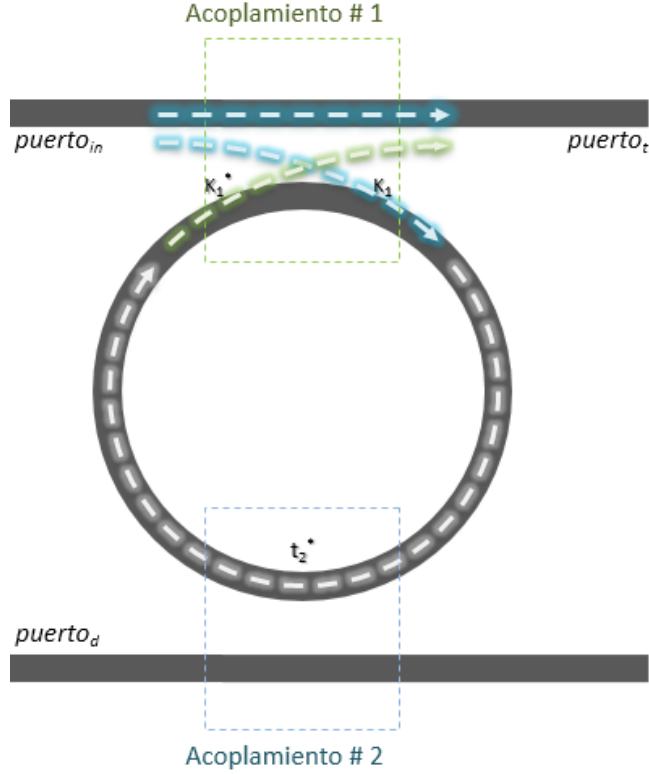


Figura 3.2: Contribución Onda 1 Vuelta

κ_1') vuelve a la guía original para salir por el *puerto_t*.

Teniendo en cuenta cada una de estas atenuaciones más el fasor que expresa la propagación de la onda, se llega a (3.4).

$$Contrib_{M=1L} = E_i \alpha e^{-j\beta L} \kappa_1 t_2' \kappa_1' \quad (3.4)$$

- Contribución después de dos vueltas: $Contrib_{M=2L}$

Se analizará la parte de la onda que no se reintegró a la guía recta tras la primera vuelta y que da otra vuelta antes de volver a la guía recta para salir por el *puerto_t* (Figura 3.3). La propagación de la onda tras 2 vueltas completas ($2L$), está dada por $\alpha^2 e^{-j\beta 2L}$. La onda atraviesa 2 nuevas regiones (aparte de las 3 regiones mencionadas en la sección anterior) por cada nueva vuelta que deba dar.

La primera es la región de acoplamiento 1 (en una proporción dada por t'_1) para seguir su trayectoria dentro del anillo. La segunda es la región de acoplamiento 2, la cual debe atravesar (según el factor de transmisión t_2).

Estas nuevas atenuaciones se ven reflejadas en (3.5). El término $\alpha^2 e^{-2j\beta L}$ se expresó como $\alpha e^{-j\beta L} \alpha e^{-j\beta L}$ para facilitar su generalización posterior.

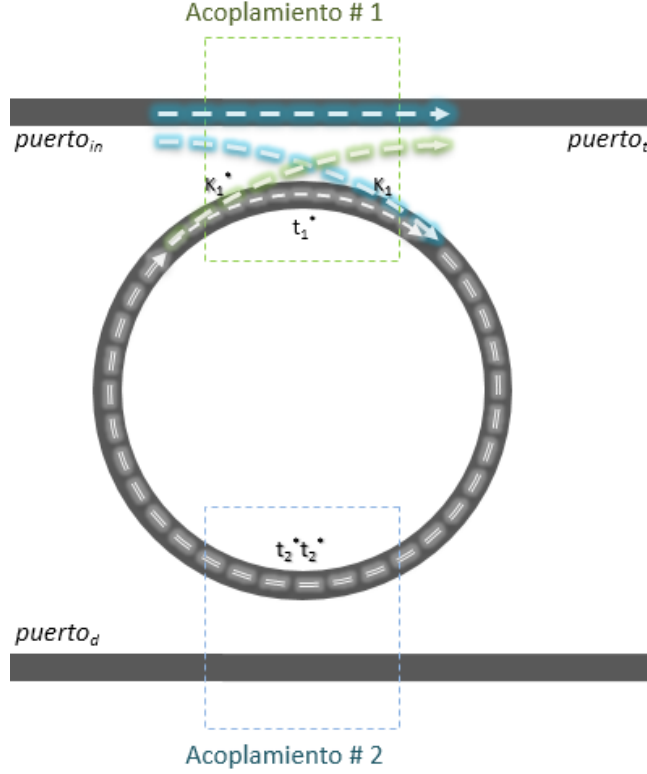


Figura 3.3: Contribución Onda 2 Vueltas

$$Contrib_{M=2L} = E_i \alpha e^{-j\beta L} \kappa_1 t'_2 \kappa'_1 (\alpha e^{-j\beta L} t'_1 t'_2) \quad (3.5)$$

- Contribución después de N vueltas: $Contrib_N$

Por cada vuelta adicional antes de acoplarse, se deben tener en cuenta los coeficientes de transmisión en estas 2 regiones más el desfase y la atenuación de la onda en cada vuelta (3.6).

$$Contrib_N = E_i \alpha e^{-j\beta L} \kappa_1 t'_2 \kappa'_1 (\alpha e^{-j\beta L} t'_1 t'_2)^{N-1} \quad (3.6)$$

Sustituyendo la expresión para cada una de las contribuciones en (3.3) y reorganizando, se llega a (3.7).

$$E_t = E_i \{ t_1 + \kappa_1 \kappa_1' t_2' \alpha e^{-j\beta L} [1 + (t_1' t_2' \alpha e^{-j\beta L})^1 + (t_1' t_2' \alpha e^{-j\beta L})^2 + \dots] \} \quad (3.7)$$

Al ser una serie geométrica infinita, su solución está dada por (3.8).

$$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}, \text{ si } |r| < 1 \quad (3.8)$$

Sea $a = \kappa_1 \kappa_1' t_2' \alpha e^{-j\beta L}$ y $r = t_1' t_2' \alpha e^{-j\beta L}$. Por lo tanto:

$$E_t = E_i \left\{ t_1 + \frac{\kappa_1 \kappa_1' t_2' \alpha e^{-j\beta L}}{1 - t_1' t_2' \alpha e^{-j\beta L}} \right\} \quad (3.9a)$$

$$\frac{E_t}{E_i} = \frac{t_1 + (\kappa_1 \kappa_1' - t_1 t_1') t_2' \alpha e^{-j\beta L}}{1 - t_1' t_2' \alpha e^{-j\beta L}} \quad (3.9b)$$

El cálculo de la potencia transmitida en el *puerto*_d (3.10) sigue una lógica similar.

$$\frac{E_d}{E_i} = \frac{\kappa_1 \kappa_2' \alpha e^{-j\beta \frac{L}{2}}}{1 - t_1' t_2' \alpha e^{-j\beta L}} \quad (3.10)$$

3.2.2 Relación entre Coeficientes de Acoplamiento

Como se explica en [32], los 4 coeficientes de transmisión más los 4 coeficientes de acoplamiento no son independientes entre si, sino que están relacionados por los principios fundamentales de reciprocidad, conservación de la energía y T-simetría. Adicionalmente, como se mencionó en el apartado anterior, el sistema asume que no hay pérdidas por inserción (3.11a).

$$|t_1|^2 + |\kappa_1|^2 = 1 \quad (3.11a)$$

$$t_1 t_1' - \kappa_1 \kappa_1' = -1 \quad (3.11b)$$

$$t_i = |t_i|e^{j\phi_{t_i}} \quad (3.11c)$$

Por conveniencia, en [32] se define el sistema del acoplamiento 1 mediante una matriz Hermitiana (3.12) con determinante -1. Esto permite relacionar facilmente el coeficiente de transmisión dentro del anillo t'_1 en términos de la conjugada compleja del coeficiente de transmisión t_1 .

$$\begin{bmatrix} E_s \\ E_{sk} \end{bmatrix} = \begin{bmatrix} t_1 & k_1^* \\ k_1 & -t_1^* \end{bmatrix} \begin{bmatrix} E_i \\ E_{ik} \end{bmatrix} \quad (3.12)$$

Por comparación directa de (3.12) con el sistema inicial, se ve que $t'_1 = -t_1^*$ y $t'_2 = -t_2^*$. Reemplazando estas equivalencias y (3.11b) en (3.9b) se encuentra la expresión para la amplitud normalizada en el *puerto* (3.13).

$$\frac{E_t}{E_i} = \frac{t_1 - t_2^* \alpha e^{-j\beta L}}{1 - t_1^* t_2^* \alpha e^{-j\beta L}} \quad (3.13)$$

Cuya función de transmisión (3.14) obtenida al multiplicar por la correspondiente conjugada compleja (recordar que $|\chi|^2 = \chi\chi^*$) es [24]:

$$T_t = \left| \frac{E_t}{E_i} \right|^2 = \frac{\alpha^2 |t_2|^2 + |t_1|^2 - 2\alpha |t_1| |t_2| \cos(\theta + \phi_{t_1} + \phi_{t_2})}{1 + \alpha^2 |t_1|^2 |t_2|^2 - 2\alpha |t_1| |t_2| \cos(\theta + \phi_{t_1} + \phi_{t_2})} \quad (3.14)$$

Donde $\theta = \beta L = \frac{2\pi n_{eff}}{\lambda} L$. Para el *puerto*_d, se calcula la amplitud normalizada a partir de (3.10) como:

$$\frac{E_d}{E_i} = \frac{\kappa_1 \kappa_2^* \alpha e^{-j\beta \frac{L}{2}}}{1 - t_1^* t_2^* \alpha e^{-j\beta L}} \quad (3.15)$$

De (3.10) y (3.11a) se obtiene también su función de transmisión [1]:

$$T_d = \frac{\alpha^2 (1 - |t_1|)^2 (1 - |t_2|)^2}{1 + \alpha^2 |t_1|^2 |t_2|^2 - 2\alpha |t_1| |t_2| \cos(\theta + \phi_{t_1} + \phi_{t_2})} \quad (3.16)$$

3.2.3 Filtro Notch

Un caso especial de (3.13) se da cuando la onda es transmitida completamente dentro del anillo resonador en la región de acoplamiento 2 [32]. En este caso $t_2^* = 1$ y $k_2^* = 0$ por lo que la salida en el *puerto_d* es cero y en el *puerto_t* es (3.17a). El sub-índice de t_1 no es necesario ya que sólo hay un acoplamiento.

$$\frac{E_t}{E_i} = \frac{t - \alpha e^{-j\beta L}}{1 - t^* \alpha e^{-j\beta L}} \quad (3.17a)$$

$$T_t = \frac{\alpha^2 + |t|^2 - 2\alpha|t|\cos(\theta + \phi_t)}{1 + \alpha^2|t|^2 - 2\alpha|t|\cos(\theta + \phi_t)} \quad (3.17b)$$

Acoplo Crítico

Se observa que ocurre una situación especial cuando la suma de los desfases que sufre la onda en su viaje completo al rededor del anillo es un múltiplo entero de 2π (3.18). Esta condición es llamada condición de resonancia (Figura 3.4).

$$\theta + \phi_t = 2\pi * M \quad (3.18)$$

Bajo esta condición, la ecuación de transmisión en el *puerto_t* queda:

$$T_t = \frac{(\alpha - |t|)^2}{(1 - \alpha|t|)^2} \quad (3.19)$$

A partir de (3.19) se ve que cuando $\alpha = |t| = \sqrt{1 - |\kappa|^2}$ la transmitancia en el *puerto_t* es cero. Es decir que cuando las pérdidas en la región de acoplamiento son iguales a las pérdidas en el anillo, se llega a la condición llamada Acoplo Crítico, donde la potencia de salida se anula.

Visto desde el punto de vista de [5] el fenómeno se produce porque la longitud de onda que cumple la condición se acopla, sufre un desfase de $\frac{\pi}{2}$, es decir $\kappa = i|\kappa|$. Luego de completar una vuelta completa sufre un desfase de 2π y cuando se vuelve a acoplar a la guía recta es desfasada nuevamente $\frac{\pi}{2}$. Es decir, que cuando vuelve a la guía inicial, se suma en contrafase en el punto de acoplamiento de la guía, anulándola.

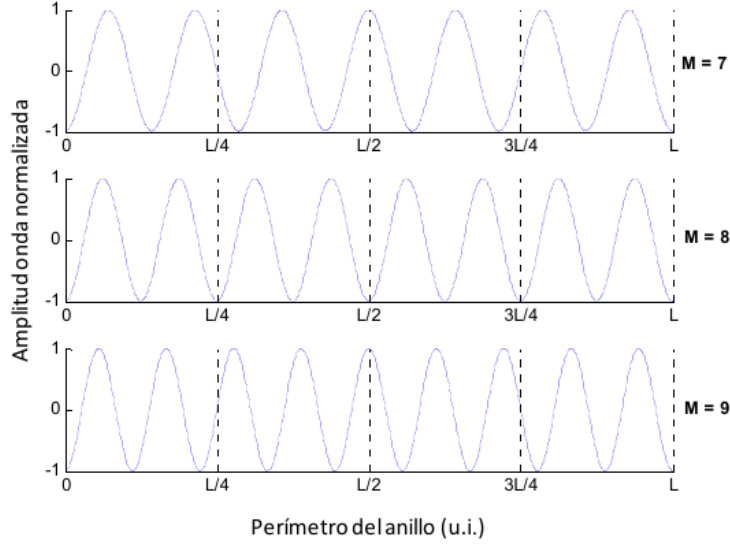


Figura 3.4: Modos Resonantes. Fuente[5]

3.2.4 Parámetros

Existen diferentes parámetros que son relevantes a la hora de describir el rendimiento de un anillo resonador. A continuación se realizará la formulación matemática de cada uno acompañado de una breve descripción.

Los parámetros para el notch filter se obtienen sustituyendo t_2 por 1, como se explicó en la sección 3.2.3.

Free Spectral Range (FSR)

El FSR o rango libre de espectro, es la separación que existe entre dos longitudes de onda que resuenan dentro del anillo. De (3.2) y (3.1) se tiene que:

$$M = \frac{n_{eff(\lambda_m)} L}{\lambda_m} \quad (3.20)$$

$$M + 1 = \frac{n_{eff(\lambda_{m+1})} L}{\lambda_{m+1}} \quad (3.21)$$

Para un rango amplio de longitudes de onda β no es realmente constante debido

a la dispersión dentro de la guía por lo que en vez de n_{eff} se usa en índice de grupo (3.22) [24].

$$n_g = n_{eff} - \lambda \frac{\partial n_{eff}}{\partial \lambda} \quad (3.22)$$

El FSR, dado por $\Delta\lambda = \lambda_{m+1} - \lambda_m$ se halla restando (3.21) y (3.20). También se asume que $n_{eff(\lambda_m)} \approx n_{eff(\lambda_{m+1})} \approx M$ y que $\lambda_m \lambda_{m+1} \approx \lambda^2$, por lo tanto:

$$FSR = \Delta\lambda \approx \frac{\lambda^2}{n_g L} \quad (3.23)$$

De (3.23) se puede observar que el FSR es inversamente proporcional a el radio del resonador. Para un anillo con un radio grande el distanciamiento entre las longitudes de onda que resuenan en él es menor. Esto indica su capacidad para manejar una densidad de espectro alta y son principalmente usados como switches de banda ancha.

Por el otro lado, un anillo pequeño tendrá un mayor espaciamento entre los modos lo que significa que es más selectivo [7] y por eso son muy usados como filtros y moduladores.

Full Width Half Maximum

La anchura a media altura representa el ancho de la resonancia medido en nanómetros ($\Delta\lambda_{FWHM}$) cuando ésta decae 3dB o mitad de la potencia y su expresión para una cavidad Fabry-Perot [24][30] es:

$$\Delta\lambda_{FWHM} = 2\delta\lambda = \frac{\lambda_0^2}{Ln_{eff}\pi} \left(\frac{1 - \alpha^2|t_1|^2|t_2|^2}{\alpha|t_1||t_2|} \right) \quad (3.24)$$

Quality Factor (Q)

El factor de calidad es una relación entre la longitud de onda que resuena en el anillo y el ancho de la resonancia en la mitad de la potencia.

$$Q = \frac{\lambda_0}{\Delta\lambda_{FWHM}} = \pi \frac{n_{eff}L}{\lambda_0} \frac{\alpha|t_1||t_2|}{1 - \alpha^2|t_1|^2|t_2|^2} \quad (3.25)$$

Finesse (F)

La finura F permite medir la estrechez de las resonancias [5], donde entre más estrecha, el anillo es más selectivo.

$$F = \frac{FSR}{\Delta\lambda_{FWHM}} = \frac{\Delta\lambda}{2\delta\lambda} = \pi \frac{\alpha|t_1||t_2|}{1 - \alpha|t_1|^2|t_2|^2} \quad (3.26)$$

En (3.26) se aprecia que entre menos pérdidas existan en el anillo, la finura del anillo es más elevada.

3.3 Simulaciones

Las simulaciones se realizaron sobre las herramientas Meep y Lumerical en 2 máquinas con las siguientes características:

Procesador intel core i7 980X. Frecuencia de Reloj OC 4.5 GHz.
6 núcleos - 12 hilos.
24 GB de Memoria RAM a bus de 1600MHz.
GPU nVidia GTX580.
2 discos duros de estado solido de 128 GB en RAID 0.

Tabla 3.1: Equipo para simulación en Lumerical. Codename: Amelie

Procesador intel core i7 Ivy Bridge, 3770k. Frecuencia de reloj OC 4.4 GHz
4 núcleos - 8 hilos.
16 GB de RAM a bus de 1600 MHZ
GPU ATI 7950. ASUS DirectCU II
128 GB de disco duro de estado sólido sata3

Tabla 3.2: Equipo para simulación en Meep. Codename: Arwen

3.3.1 Metodología

Para la realización de esta etapa de la investigación, se planteó realizar la simulación de la onda electromagética cuando interactúa con una estructura en anillo con el fin de conocer los parámetros que afectan directamente su comportamiento.

Caso de Estudio

Para el caso de prueba el diseño propuesto se eligieron dos filtros: Notch y AddDrop. Estos componentes son de extrema importancia ya que son los ladrillos de construcción de estructuras más complejas que se emplean como filtros, switches banda ancha y detectores.

Las simulaciones se realizaron mediante la técnica de FDTD en 2 plataformas diferentes. La primera es una herramienta de software libre llamada MEEP en la cual se codifica un script que contiene la información de la estructura a simular. La segunda es una herramienta comercial llamada Lumerical-FDTD, que cuenta con una interfaz gráfica de usuario que sirve para controlar el diseño y la ejecución de la simulación, pero a su vez también cuenta con soporte para scripts con una sintaxis similar a la de matlab (ver 3.3.3 y 3.3.4).

Una vez ejecutadas las simulaciones, se obtuvieron las gráficas de potencia vs longitud de onda y se procedió a comparar los resultados.

3.3.2 Diseño del Resonador

Para el diseño del anillo, se siguieron los lineamientos descritos en [17] tanto para el filtro AddDrop como para el filtro Notch. El primero consiste en el modelo explicado en la sección 3.2.1 con la salvedad que la región de Acoplamiento 1 es idéntica a la región de Acoplamiento 2, por lo tanto $\kappa = \kappa_1 = \kappa_2$, $\kappa^* = \kappa_1^* = \kappa_2^*$, $t = t_1 = t_2$ y $t^* = t_1^* = t_2^*$. Adicionalmente, se asume que no hay pérdidas ($\alpha = 1$) y se seleccionan las características dadas en la tabla 3.3.

Parámetro	Valor
Rango de frecuencias	1500nm a 1600nm
λ_0	1550nm
Espaciamiento de canales	200GHz ó 1.6nm a 1550nm
FSR	3200GHz ó 25.6nm a 1550nm ó 16 canales
$\Delta\lambda_{FWHM}$	100GHz ó 0.8nm
Q	$\frac{1550nm}{0.8nm} \approx 2000$ (3.25)

Tabla 3.3: Configuración deseada para el filtro

Índice de grupo n_g

Se calcula el índice de grupo n_g mediante la herramienta MODE de lumérica (sección 3.3.3) para encontrar los valores propios en un rango de frecuencias de interés. En la gráfica 3.5 se ve el perfil del primer modo a la izquierda y la

gráfica de longitud de onda vs índice de grupo a la izquierda. A partir de ésta, se visualiza que para $\lambda_0 = 1550nm$ se tiene $n_g \approx 4.8$.

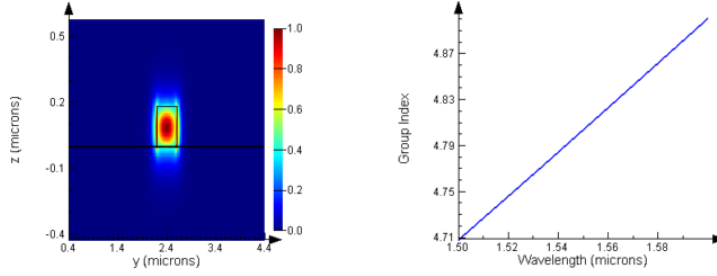


Figura 3.5: Índice de grupo. Fuente[17]

Perímetro del anillo

Despejando L partir de (3.23) y sustituyendo los parámetros (Tabla 3.3), se tiene:

$$L = \frac{\lambda_0^2}{n_g FSR} = \frac{1550e-9}{4.8 \times 25.6e-9} = 19.5\mu m$$

$$R = \frac{L}{2\pi} = 3.1\mu m$$

Coefficiente de transmisión y acoplamiento

Al despejar $|t|$ de (3.25) se tiene:

$$|t| = \sqrt{\left(\frac{n_g L \pi}{2Q\lambda}\right)^2 + 1} - \frac{n_g L \pi}{2Q\lambda} \quad (3.27)$$

Y remplazando la configuración dada en la tabla 3.3:

$$|t| = \sqrt{\left(\frac{4.819.5e-6\pi}{22000 \cdot 1550e-9}\right)^2 + 1} - \frac{4.819.5e-6\pi}{22000\lambda} \approx 0.954$$

El valor de κ se obtiene despejando (3.11a):

$$|\kappa| = \sqrt{1 - |t|^2} \approx 0.301 \quad (3.28)$$

Transmitancia

De (3.11a) y (3.16) y teniendo en cuenta las consideraciones de la tabla 3.3 se tiene:

$$T_d = \frac{|\kappa^4|}{|1 - t^2 e^{j\beta L}|^2} \quad (3.29)$$

$$T_t = \frac{|t - t e^{-j\beta L}|^2}{|1 - t^2 e^{j\beta L}|^2} \quad (3.30)$$

3.3.3 Lumerical

Lumerical FDTD Solutions es un solucionador 3D de las ecuaciones de Maxwell, capaz de analizar la interacción de la radiación UV, visible, IR y con estructuras complejas que emplean características de longitud de onda de escala. FDTD Solutions es capaz de tomar en cuenta de forma precisa la dispersión en un material dentro de intervalos de longitudes de onda, lo que permite al usuario final para calcular eficientemente la respuesta dispositivo a través de grandes rangos de ancho de banda.

Posee un motor computacional altamente optimizado capaz de explotar sistemas de varios núcleos de computación en todo, desde computadoras portátiles hasta clusters de computación de alto rendimiento, y un framework de optimización para acelerar la generación de dispositivos nanofotónicos optimizados [18].

Posee las siguientes características

- Simulación de geometrías arbitrarias en 2D, 3D.
- Computación Concurrente en varios equipos y cómputo paralelo en sistemas multicore y multiNode
- Framework de optimización.
- Capacidad de modelamiento de materiales con características dispersivas,

de ganancia no lineal y anisotrópicos.

- Potente lenguaje de scripts cuya sintaxis es similar a la de matlab.

Resultados

Sobre esta plataforma, se ejecutaron las simulaciones del resonador descrito en la sección para las configuraciones de los filtros Notch y AddDrop.

Filtro Notch

La configuración del filtro notch se realizó en 3 dimensiones. Se empleó un paralelepípedo como el material base en Silica y la estructura básica del resonador de anillo en Silicio que viene incluida en los componentes ópticos del software. Para eliminar la guía del *puerto_d*, se realizó una ligera modificación al script que general la estructura (ver apéndice A.2.1).

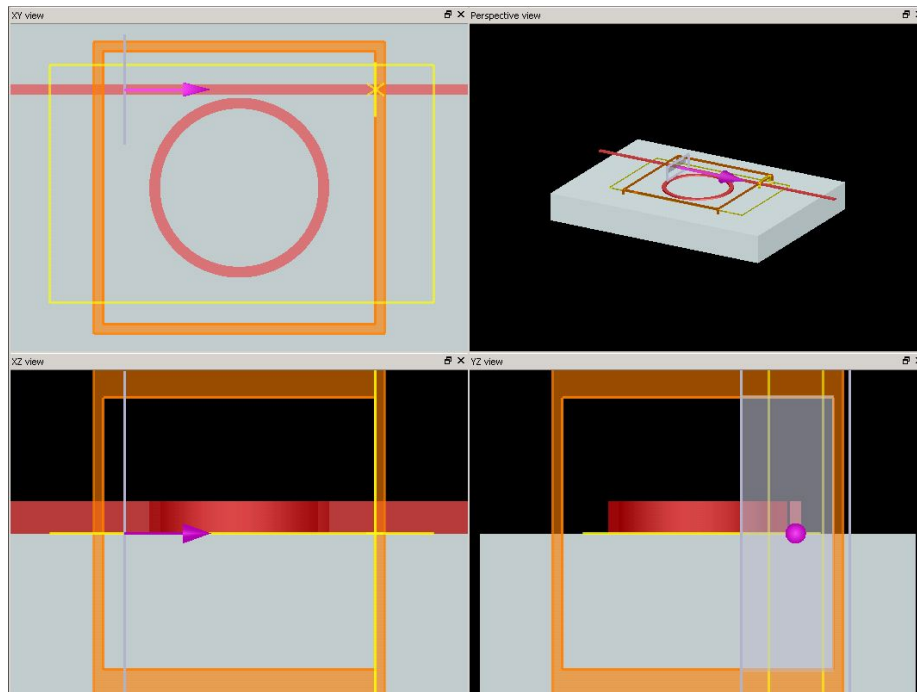


Figura 3.6: SetUp del filtro Notch en Lumerical FDTD

Se obtuvo el valor de la transmitancia, cuyo valor teórico está dado por la ecuación 3.30, para 500 valores de longitud de onda entre 1500 y 1600 nm como se ve en la siguiente figura:

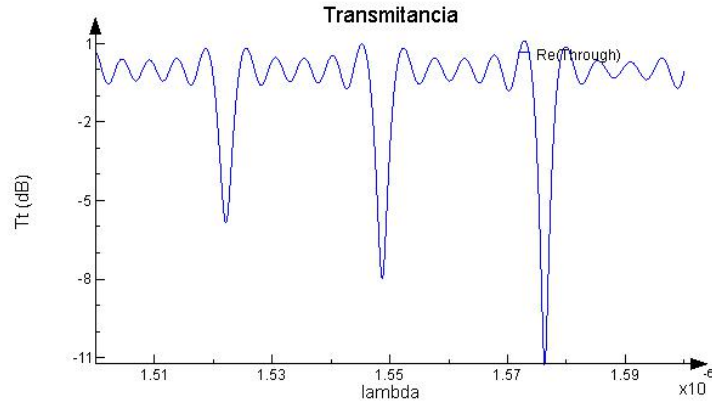


Figura 3.7: Transmitancia simulación 3D FDTD Filtro Notch

Filtro AddDrop

La configuración del filtro AddDrop se realizó de forma similar al filtro Notch simplemente adicionando la guía inferior sobre la que se encuentra el *puerto_d*.

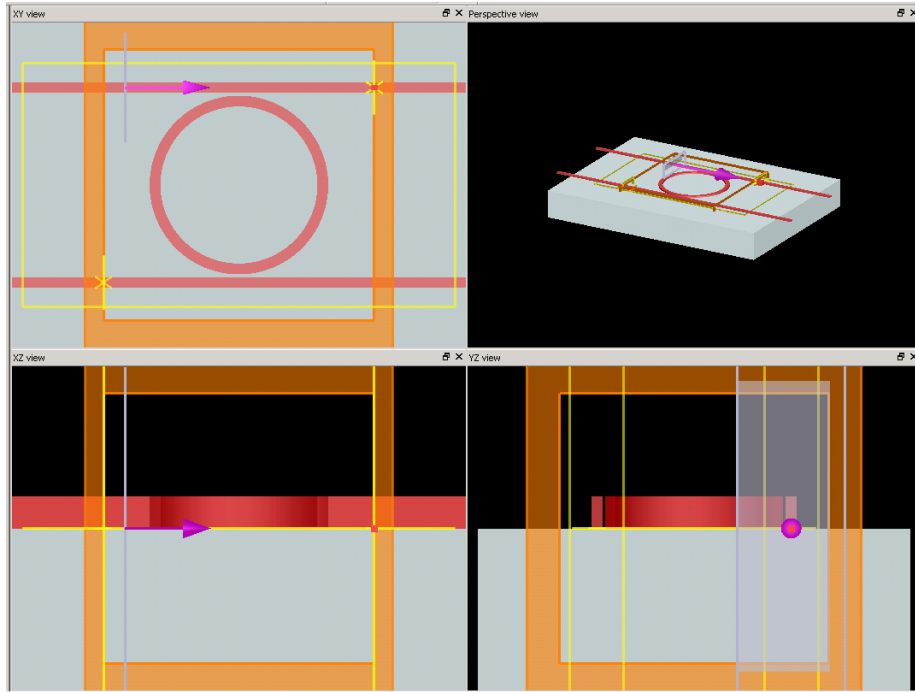


Figura 3.8: SetUp del filtro AddDrop en Lumerical FDTD

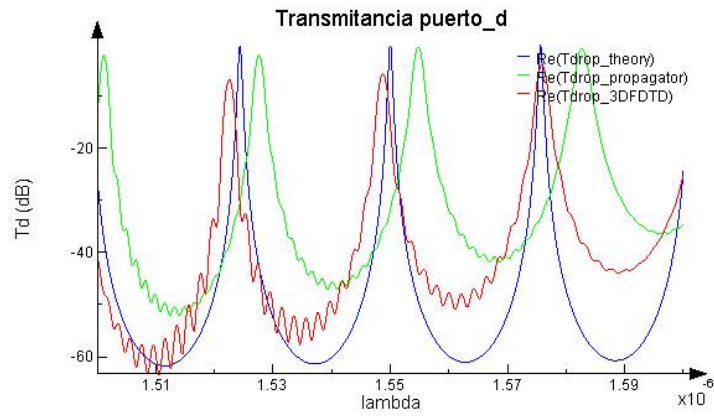


Figura 3.9: Transmitancia en el $puerto_d$ de valores teóricos, simulación MODE y FDTD del Filtro AddDrop

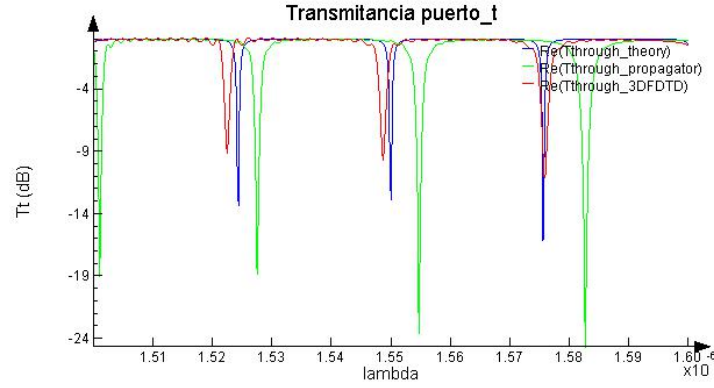


Figura 3.10: Transmitancia en el puerto_t de valores teóricos, simulación MODE y FDTD del Filtro AddDrop

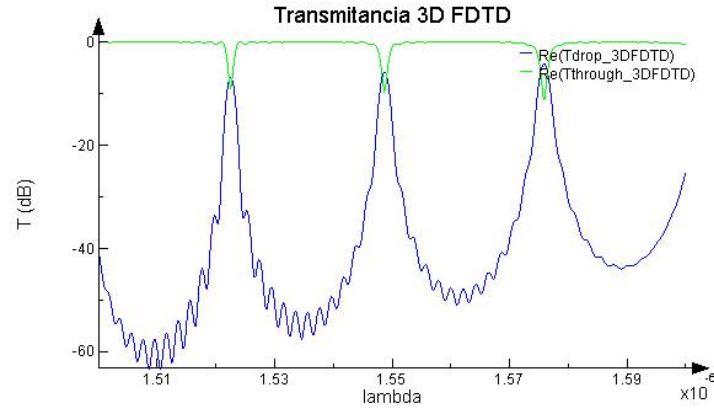


Figura 3.11: Transmitancia simulación 3D FDTD Filtro AddDrop

3.3.4 Meep

Meep es un software desarrollado en el Massachusetts Institute of Technology - MIT que implementa el método FDTD. Su nombre es un acrónimo que proviene de las iniciales de “MIT Electromagnetic Equation Propagation” y cuenta con las siguientes características [23]:

- Simulación 1D, 2D, 3D

- Soporte MPI
- Fronteras Absorción PML
- Scripts en Scheme (*.ctl)
- Scripts en C++
- Análisis de flujos, frec, energía ...
- Licencia GNU/GPL

Filtro Notch

El código está dividido en 5 secciones en cada una de las cuales se define:

1. Parámetros de la simulación.
2. Materiales y geometría a simular.
3. Fuente de onda electromagnética.
4. Puntos de medición de flujo de energía.
5. Tiempos y salidas de la simulación.

A continuación, se explicará el objetivo y el contenido de cada subsección sólo del script para el filtro Notch. Para ver el script completo 2D y 3D de los 2 filtros, ver el apéndice A.1.

Parámetros

Script 3.1: Parámetros Filtro Notch.

```

1 ;Units are measured in microns [a=luc]
2 (define-param odir "./out-")
3 (define-param w 0.4) ; width of waveguide
4 (define-param r 2.9) ; inner radius of ring
5 (define-param gap 0.1) ; gap between ring and wg
6 (define-param pad 4) ; padding between wg and edge of PML
7 (define-param dpml 2) ; thickness of PML
8 (define-param wavecen 1.550)
9 (define-param wavewid 0.050)
10 (define-param freqcen (/ 1 wavecen))

```

```

11 (define-param freq_width (-
12     (/ 1 (- wavecen waveid))
13     (/ 1 (+ wavecen waveid))))
14 (define-param flux_points 100)
15
16 (define sxy (* 2 (+ r w pad dpml))) ; cell size
17 (define si_mat (make dielectric (index 3.476558)))

```

En esta sección de código (Script 3.1) se especifican los parámetros necesarios para la ejecución de los programas. Se usaron 2 tipos diferentes de instrucciones: *define* y *define-param*. La primera instrucción, de la forma (*define* < variable > < expresión >), es nativa de Scheme y permite ejecutar una expresión dada por medio de una variable.

Por el contrario, *define-param*, está definida en una librería de extensión llamada LibCtl y permite que la asociación de la variable a la expresión sea modificada desde la línea de comandos desde la que se invoca el programa permitiendo tener un control flexible para las simulaciones.

Los parámetros usados en la simulación del filtro están especificados en la Tabla 3.4.

Parámetro	Descripción	Valor x Defecto
w	Ancho de la guía de onda	4 nm
r	Radio interno del anillo resonador	2.9 μm
gap	Espacio entre la guía de onda y el anillo	1 nm
$dpml$	Ancho de la capa PML	2 μm
$wavecen$	Ancho de banda central de la fuente	1550 nm
$waveid$	Ancho del pulso de la fuente	50 nm
$freqcen$	Frecuencia central de la fuente	$\frac{1}{wavecen}$
$freqwidth$	Ancho del pulso de la fuente (en frecuencia)	$\frac{1}{wavecen-waveid} - \frac{1}{wavecen+waveid}$

Tabla 3.4: Parámetros

Materiales y Geometría

En esta sección se emplearon algunos de los parámetros definidos en la tabla 3.4 para construir la geometría del filtro como se ve en la figura 3.14.

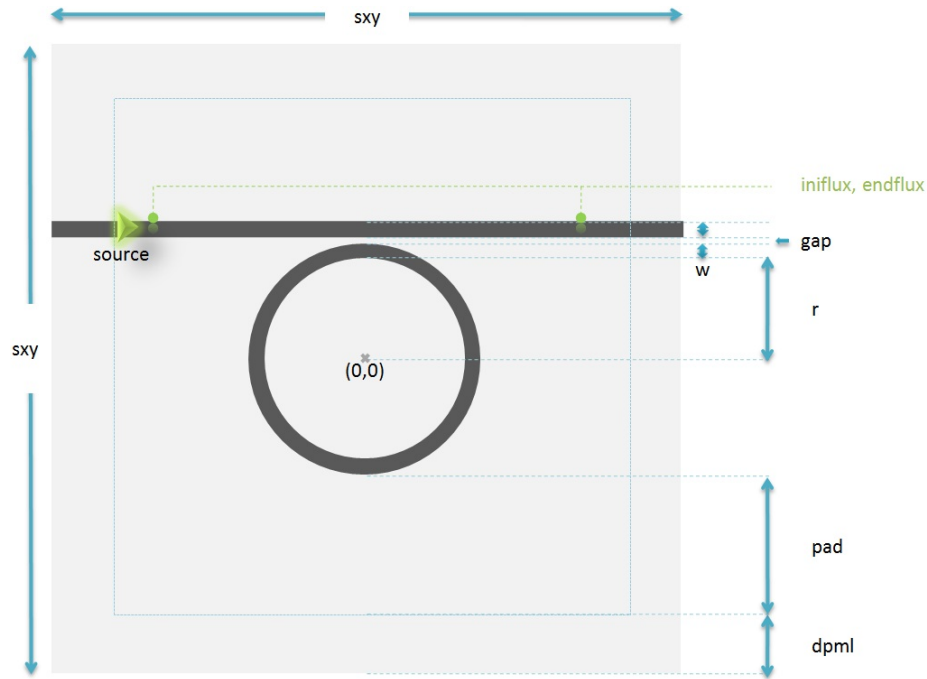


Figura 3.12: Geometría Filtro Notch.

Como se aconseja en [22] el tamaño del látice a simular se calcula de forma dinámica a partir de los parámetros del radio, ancho de la guía de onda, espacio de holgura y el borde PML (Script 3.2).

Script 3.2: Geometría y Materiales Filtro Notch.

```

1 (define si_mat (make dielectric (index 3.476558)))
2 (set! default-material air)
3
4 (set! geometry-lattice (make lattice (size sxy sxy no-size)))
5 (set! geometry (list
6   (make cylinder
7     (center 0 0) (height infinity) (radius (+ r w))
8     (material si_mat))
9   (make cylinder
10    (center 0 0) (height infinity) (radius r)
11    (material air))
12   (make block
13     (center 0 (+ r w gap (/ w 2))) (size sxy w infinity)
14     (material si_mat))))
15
16 (set-param! resolution 35)

```

Silicon Photonics utiliza, como su nombre lo indica, silicio como medio para la propagación de ondas electromagnéticas en el espectro C-Band . Por lo tanto, para la longitud de onda de 1500 nm, el índice de refracción del silicio corresponde a 3.4765 [3]. Adicionalmente, al ser una simulación en 2D, el material que rodea la guía recta y circular es el aire cuyo índice de refracción es 1. La base de dióxido de silicio SiO₂ sobre la cual está montada la guía, sólo se tomó en cuenta para la simulación 3D (sección 3.3.3).

La guía de onda se especifica como un rectángulo de Si, mientras que la estructura del anillo se construye a partir de la superposición de un cilindro externo de silicio y un cilindro interno de aire. Las dos estructuras están separadas en su punto más cercano por una distancia definida por el parámetro $gap = 100nm$.

Finalmente, se indicó una resolución de

Fuente de Onda Electromagnética

Como se mencionó anteriormente, una de las aplicaciones de FDTD es el cálculo de la transmisión o la dispersión del espectro de estructuras tales como cavidades resonantes en respuesta a un estímulo de entrada. Una forma de obtener estos datos es calcular el flujo transmitido para cada frecuencia ω de forma separada. Una forma más rápida es calcular la respuesta a un conjunto amplio de espectros usando un sólo cálculo de la transformada de Fourier usando un pulso gaussiano de banda ancha [21].

En este trabajo se simularon estas 2 formas (cada una en un script diferente) tanto para el filtro Notch, como para el filtro AddDrop.

Resultados

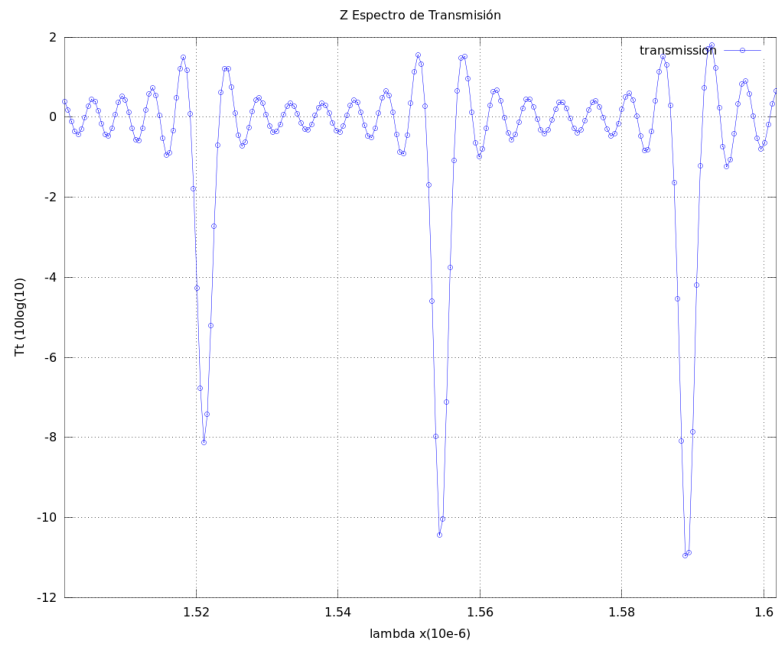


Figura 3.13: Transmitancia Filtro Notch

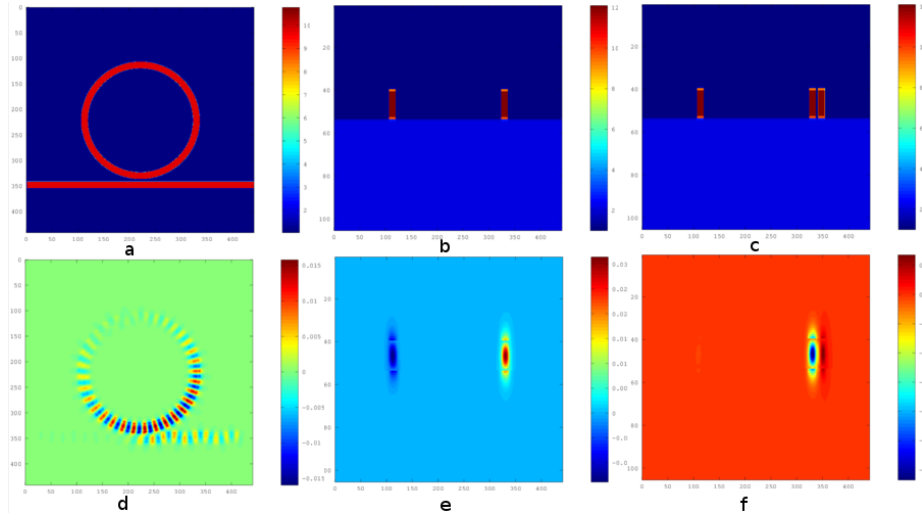


Figura 3.14: Filtro Notch 3d vista $[X,Y]$, $[Y,Z]$ y $[X,Z]$ respectivamente. (a)(b)(c) Material Dieléctrico. (d)(e)(g) Campo eléctrico E_z

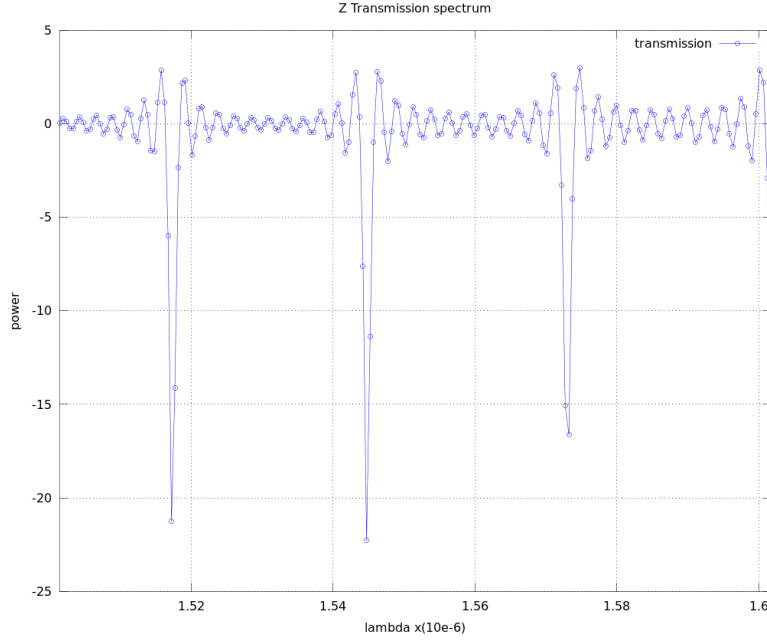


Figura 3.15: Transmitancia Filtro Notch 3D

3.3.5 Conclusiones y Trabajos Futuros

- Las frecuencias de resonancia del anillo pueden ser manipuladas mediante modificaciones en el diámetro del anillo, el gap o el índice de refracción. Este último es de gran importancia por ser la base del funcionamiento algunos dispositivos moduladores en Silicon Photonics donde se aplica un campo magnético, generando un efecto plasmónico que altera el índice y por lo tanto los modos de resonancia.
- Debido a que se asumió que el anillo no presentaba pérdidas ($\alpha = 1$) en el cálculo teórico de la transmitancia tanto del filtro Notch como del filtro AddDrop, se pueden apreciar diferencias en las condiciones de resonancia de la teoría con respecto a lo obtenido en la práctica.
- Las diferencias en el FSR obtenido en las simulaciones realizadas en Meep con respecto a las realizadas en Lumerical, pueden ser explicadas debido a la diferencia en las dimensiones de ambas simulaciones. Mientras que en Lumerical se realizó una simulación 3D de ambos filtros, en Meep la

mayoría fueron realizadas en 2D a diferencia del último caso. Este escenario permitió corroborar esta teoría ya que la simulación Meep 3D del filtro Notch fue muy similar a la simulación del mismo filtro en la plataforma Lumerical.

- Como trabajo futuro se plantea realizar nuevas simulaciones para los dos filtros modificando el diámetro del anillo para visualizar la alteración en el FSR y su uso como filtro selectivo o filtro de banda ancha.
- Se propone también, para un diámetro fijo, realizar simulaciones alterando el índice de refracción del anillo para simular el efecto plasmónico que sucede dentro del modulador y que hace que se desplacen las frecuencias de resonancia de un estado On a Off.

Capítulo 4

Redes de Interconexión

4.1 PhoenixSim

El nombre de PhoenixSim proviene de *Photonic and Electronic Network Integration* ó integración fotónica y electrónica de redes. Esta herramienta permite modelar y analizar el rendimiento de sistemas multiprocesador que usan redes electrónicas, fotónicas o híbridas. PhoenixSim captura las características físicas y las métricas de los diferentes dispositivos de interconexión nanofotónicos y elementos de red que no poseen un equivalente electrónico [9].

Está desarrollado sobre la herramienta OMNeT++ para simulación de eventos discretos [29] sobre la cuál se montó una librería de dispositivos fotónicos y electrónicos altamente parametrizable.

Para el desarrollo de redes de interconexión sobre PhoenixSim, se recomienda seguir las siguientes etapas metodológicas [7] mostradas en la Figura 4.1:

- Especificar los dispositivos con los que se construirá la red.
- Seleccionar la aplicación o aplicaciones que se ejecutarán sobre la red.
- Modelar la aquitectura de la red
- Analizar el rendimiento a nivel de sistema.
- Caracterizar la red a nivel físico.

- Refinamiento iterativo de los pasos anteriores (parámetros y diseño).

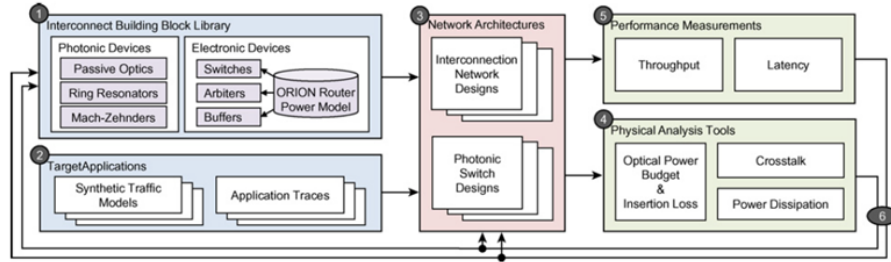


Figura 4.1: Vista General PhoenixSim. Fuente [7]

4.1.1 Dispositivos de Interconexión

Electrónicos

Router Electrónico

El router electrónico es usado tanto en las redes completamente electrónicas como en las fotónicas. En éstas últimas su uso se da a través del router híbrido, el cual lo necesita para controlar el camino por el cual serán enviados los datos en el plano fotónico.

- Puerto de Entrada (InPort)
- Entramado de comunicación (Crossbar)
- Controlador (Arbiter)

Fotónicos

PhoenixSim abstrae las características físicas más importantes de los elementos fotónicos (capítulos 2 y 3) sin perder la precisión de sus resultados y sin necesidad de realizar una simulación FDTD completa.

Al usar sólo las características más relevantes, se pueden emplear como componentes de las diferentes redes al mismo tiempo que se mantiene el rendimiento de la simulación a nivel de sistema. Existen 2 categorías: activos y pasivos. La

diferencia radica en que el comportamiento de los dispositivos pasivos o estáticos no cambia durante la ejecución de la simulación.

Pasivos

- Guía de onda

Representa el cable óptico con el que se conectan los demás dispositivos fotónicos. Presentan pérdidas por inserción debido a la atenuación que experimenta la onda al viajar a través de la estructura. Otros factores influyen son las dimensiones de la guía, las propiedades de los materiales usados y las técnicas de fabricación.

Este dispositivo está caracterizado por los 3 parámetros mostrados en las tablas 4.1 y 4.4.

Descripción	Nombre Parámetro	Abv.
Longitud	<i>Length</i>	L_{wg}
Latencia	<i>LatencyRate_Line</i>	t_{wg}
Pérdidas por Propagación	<i>PropagationLoss</i>	α_{wg}

Tabla 4.1: Caracterización Guía de Onda.

En PhoenixSim la guía de onda es modelada como un sistema con 2 puertos de entrada y 2 puertos de salida. Como se ve en la figura 4.2, este sistema posee una tabla de enrutamiento $[1, 0]$ donde el índice representa el puerto de entrada y el valor indica el puerto por el que sale la señal. Es decir, la señal óptica que ingresa por el puerto 0 sale por el puerto 1 y viceversa. [7]

$$Latencia_{wg} = \begin{bmatrix} - & L_{wg} t_{wg} \\ L_{wg} t_{wg} & - \end{bmatrix} \quad Pérdida_{wg} = \begin{bmatrix} \infty & L_{wg} \alpha_{wg} \\ L_{wg} \alpha_{wg} & \infty \end{bmatrix}$$

$$Tabla\ de\ Enrutamiento_{wg} = [1 \quad 0]$$



Figura 4.2: Caracterización: Guía de Onda.

Finalmente, para dar una idea de la implementación de este componente, se extrajo la siguiente sección de código del archivo *line.cc* de PhoenixSim.

Script 4.1: Implementación de la guía de onda en PhoenixSim.

```

1 double Line::GetLatency(int indexIn, int indexOut)
2 {
3     return Length_Line*LatencyRate_Line;
4 }
5
6 double Line::GetPropagationLoss(int indexIn, int
    indexOut, double wavelength)
7 {
8     if(indexIn != indexOut)
9     {
10        return Length_Line*PropagationLoss;
11    }
12    else
13    {
14        return MAX_INSERTION_LOSS;
15    }
16 }
17
18 void Line::SetRoutingTable()
19 {
20     routingTable[0] = 1;
21     routingTable[1] = 0;
22 }

```

- Guía de onda doblada

Este tipo de estructura es requerido con el fin de cambiar la trayectoria de una señal en un ángulo no mayor a 90 grados. Presentan pérdidas por inserción adicionales a las de la guía de onda recta (*BendingLoss*) debido a los dobles [7]. Están modelados por los parámetros de la tabla 4.2 y 4.5.

Descripción	Nombre Parámetro
Ángulo	<i>Angle_Bend</i>
Latencia	<i>Latency_Bend</i>
Pérdidas por Propagación	<i>PropagationLoss</i>
Pérdidas por doblez	<i>BendingLoss</i>

Tabla 4.2: Caracterización Guía de Onda.

- Cruce de guías de onda

Los cruces en una topología de red se presentan debido a la naturaleza planar o 2D de la plataforma tecnológica sobre la cual están construidas.

Esta situación se da cada vez que 2 guías de onda recta se intersectan presentando tanto pérdidas por inserción como diafonía (*crosstalk*) [9]. Este efecto es minimizado gracias a técnicas de fabricación donde se amplía y se emplea un doble borde en el área de cruce [6]. Este fenómeno no se presenta en las interconexiones electrónicas ya que generaría un corto circuito.

Este dispositivo está modelado como un sistema de 4 entradas y 4 salidas como se ve en la figura 4.4 y por los parámetros de la tabla 4.6.

Tabla de Enrutamiento $o_{wg} = [2 \ 3 \ 0 \ 1]$



Figura 4.3: Caracterización: Cruce de Guía de Onda.

- Acoplador

Un acoplador es una interfaz óptica entre los dominios dentro y fuera del chip. Es decir, es capaz de transferir una onda de un medio de propagación (como una fibra óptica) hacia otro (como una guía de onda) y es prácticamente transparente a la distancia de propagación [7].

Se modela como un dispositivo de 2 puertos con un sólo parámetro para demarcar las pérdidas por inserción.

Activos

Estos dispositivos están basados en anillos resonadores (Capítulo 3).

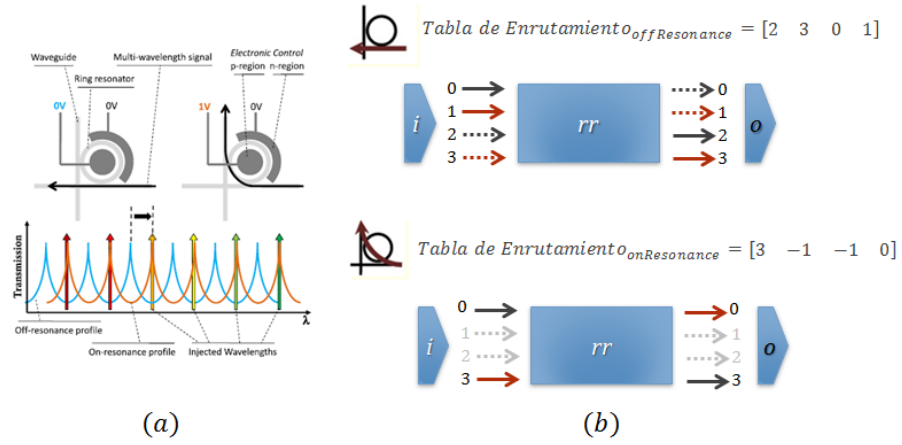


Figura 4.4: (a). Cambio de estado de resonancia on-off al aplicar un voltaje Fuente [16]. (b). Caracterización: Anillo Resonador.

- Filtros
- Switches de banda ancha
- Moduladores
- Detectores

4.1.2 Benchmarks

Modelos de Patrones Sintéticos

Random

Ejecuta un tráfico aleatorio sobre la red. Cada core selecciona un core destino aleatoriamente mediante una distribución uniforme. Una vez enviado, espera un tiempo dado por el parámetro *appParam1* para enviar un nuevo mensaje [20].

Neighbor

Realiza el envío de mensajes de cada core a sus vecinos [15].

BitReverse

Patrón de envío de mensajes diseñado para realizar pruebas de estrés sobre topologías NoC 2D. Cada core envía un mensaje a una dirección que corresponde a su dirección inversa a nivel de bits. Este tipo de comunicación, los mensajes deben viajar trayectos largos que atraviezan la red [15].

Tornado

Otro patrón de envío de mensajes que realiza pruebas de estrés sobre Mesh 2D al hacer que cada core se comuniquen con el vecino de su vecino. Es una versión competitiva del patrón Neighbor [15].

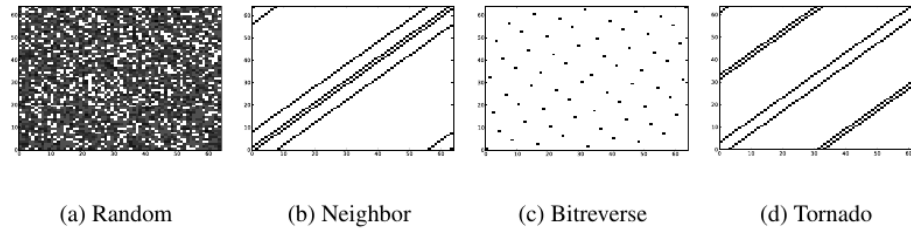


Figura 4.5: Patrones sintéticos. Fuente [15]

All2All

Permite probar que todas las rutas de comunicación entre los nodos sean accesibles. Esto permite evaluar el peor caso de pérdida por inserción en la red fotónica [20].

Pruebas de Memoria DRAM

Las pruebas sobre la memoria están divididas en 3 según la característica que se desee analizar:

- One2One comunica un procesador y un módulo de memoria DRAM. Esto permite probar la latencia de los accesos a memoria cuando no hay carga en la red (zero-load latency).
- One2All comunica un procesador hacia todos los módulos de memoria DRAM, con lo que se puede determinar la accesibilidad de estos.
- All2One prueba el mecanismo de contención al generar un patrón de comunicación de estrés donde todos los procesadores acceden a un solo módulo de memoria.

Modelos de Aplicaciones Reales

Transformada rápida de Fourier (FFT)

Está dividido en las etapas del algoritmo de Cooley-Turkey. La etapa principal es la que toma el mayor tiempo, Los tiempos de cada etapa deben ser pasados como parámetros de la simulación. Los valores por defecto incluidos en el *textbed* de PhoenixSim fueron obtenidos de la caracterización dada por Frigo y Johnson [12] usando 8 núcleos de procesamiento.

Tipos de Redes

Existe 3 tipos diferentes de redes implementadas en PhoenixSim: electrónicas basadas en créditos, fotónicas híbridas conmutadas o con arbitraje basado en longitud de onda como la empleada en [16].

A continuación se explica el tipo de red que será empleado en esta fase de la investigación:

Fotónicas Híbridas Conmutadas

El plano de comunicación entre los cores está compuesta por una subred de control electrónica y la subred de datos fotónica -es por esto que recibe el nombre de red híbrida. La subred electrónica se encarga de establecer un camino entre el origen y el destino de la comunicación mediante negociaciones entre los routers involucrados en el camino del mensaje. Una vez se recibe un mensaje de confirmación de que el camino puede ser asignado, este plano de control se encarga también de generar las señales eléctricas que configuran los dispositivos fotónicos para que establezcan el camino por el cual viajarán los datos sobre la red fotónica.

4.2 Simulaciones

Las simulaciones se realizaron sobre PhoenixSim en una máquina con las siguientes características:

Procesador intel core i7 Ivy Bridge, 3770k. Frecuencia de reloj OC 4.4 GHz 4 núcleos - 8 hilos. 16 GB de RAM a bus de 1600 MHZ GPU ATI Radeon 5830 128 GB de disco duro de estado sólido sata3 2 TB de disco duro 7200rpm
--

Tabla 4.3: Equipo para simulación en PhoenixSim. Codename: mateubuntuSSD

4.2.1 Metodología

En esta fase de la investigación, se planteó realizar la simulación de una red de interconexión electrónica y nanofotónica híbrida para realizar la comparación del desempeño en términos de latencia y consumo de energía en 5x2 diferentes escenarios.

Para cada escenario se obtuvo el tamaño de la muestra requerido para obtener un nivel de confianza del 95%. Luego se procedió a definir la comparación para cada variable a medir, mediante la prueba de hipótesis para comparación de medias.

Caso de Estudio

La red consta de 2 planos independientes: procesamiento y red de interconexión fotónica híbrida conmutada.

El plano de procesamiento está compuesto por 64 cores en un arreglo de 8x8.

Como se indica en [10], una red de interconexión está definida por 3 características principales: la topología empleada, el algoritmo de control de flujo y el algoritmo de ruteo. La red de interconexión simulada en este experimento consta de una topología Mesh con concentración de nodos, de un algoritmo de control de flujo llamado *Bubble Flow Control*[25][20] y de un algoritmo de enrutamiento basado en un esquema direccionamiento jerárquico basado en direcciones de la forma *NET.PROC* como se explica en [20].

Debido a la topología concentrada que se seleccionó, la red de interconexión está compuesta por 4x4 elementos, donde cada nodo terminal da servicio al número dado por el parámetro de concentración, que para esta simulación se estableció en 4.

Sobre cada red se ejecutaron las primeras 5 aplicaciones sintéticas mencionadas

en 4.1.2 donde el promedio del tiempo de generación de mensajes se escogió de 1 milisegundo. Adicionalmente, para cada red y para cada aplicación, se decidió realizar la simulación con 2 tamaños de mensajes (pequeño: 512Bytes y grande:65kBites) entre núcleos, siguiendo la metodología empleada en [15].

Algunos de los parámetros de dispositivos fotónicos empleados en la simulación se muestran en las tablas 4.5 a 4.7.

Parámetro	Valor	Unidad
LatencyRate_Line	1.14e-14	s/ μm
PropagationLoss	1.5e-4	dB/ μm

Tabla 4.4: Parámetros de guía de onda. [31] [13]

Parámetro	Valor	Unidad
Angle_Bend	90	grados
Latency_Bend	.06e-12	s
BendingLoss	.005	dB

Tabla 4.5: Parámetros de guía de onda doblada. [31]

Parámetro	Valor	Unidad
CrossingLoss	.15	dB
Crosstalk_Cross	40	dB

Tabla 4.6: Parámetros de cruce de guía de onda. [6]

Parámetro	Valor	Unidad
RingOn_ER_2x2	20	dB
RingOff_ER_2x2	25	dB
CrossDelay_2x2	4.35e-12	s
BarDelay_2x2	1.25e-12	s
RingOn_ER_1x2	20	dB
RingOff_ER_1x2	25	dB
ThroughDelay_1x2	1.00e-12	s
DropDelay_1x2	4.10e-12	s
RingThrough_ER_Filter1x2	25	dB
RingDrop_ER_Filter1x2	20	dB
ThroughDelay_Filter1x2	0.14e-12	s
DropDelay_Filter1x2	0.34e-12	s
RingOn_ER_1x2NX	20	dB
RingOff_ER_1x2NX	25	dB
ThroughDelay_1x2NX	0.7e-12	s
DropDelay_1x2NX	3.8e-12	s
RingThrough_ER_DetectorFilter	25	dB
RingDrop_ER_DetectorFilter	20	dB

Tabla 4.7: Parámetros de anillos resonadores.

4.2.2 Resultados

Para obtener el tamaño de la muestra necesario para una confiabilidad del 95% se empleó la siguiente ecuación:

$$n = \frac{z^2 s^2}{B^2} \quad (4.1)$$

Donde n es el tamaño de la muestra y s la desviación estándar. El nivel de confianza dado por z se estableció a un valor de 1.96 que corresponde a un 95% de confianza. Por último el parámetro B que indica la precisión, se estableció según la variable a medir. Para el consumo de energía, este valor fluctúa entre 0.1% y 0.8%, mientras que para la latencia se toma en el orden de los nanosegundos.

En las tablas de 4.8 a 4.17 se muestran los valores obtenidos en cada simulación para las variables de consumo de energía y latencia y el valor n obtenido en cada caso:

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0670822	0.00795576	0.0691798	0.0792249
5	0.0671684	0.00795162	0.0683049	0.0779179
4	0.0670545	0.00797981	0.0695929	0.0772176
2	0.0670536	0.00795009	0.0700826	0.0790122
10	0.0670933	0.00796371	0.0681599	0.0778642
8	0.0670919	0.00795244	0.0691727	0.0840341
stdev	4.19633E-05	1.13364E-05	0.000739481	0.002480388
mean	0.06709065	0.007958905	0.069082133	0.079211817
z	1.96	1.96	1.96	1.96
p	0.0015	0.0015	0.03	0.03
B	0.000100636	1.19384E-05	0.002072464	0.002376355
n	0.667950099	3.463963085	0.48909428	4.185322852

Tabla 4.8: Datos Simulación App: Neighbor, PkgSize: 512B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0672651	0.0079509	0.0865564	0.127786
5	0.0670529	0.0079602	0.0894102	0.132535
4	0.0670557	0.00796118	0.087135	0.132628
2	0.0671679	0.00796183	0.0870683	0.132774
10	0.0671439	0.00794885	0.0892598	0.133439
stdev	8.81681E-05	6.20164E-06	0.001342664	0.002289528
mean	0.0671371	0.007956592	0.08788594	0.1318324
z	1.96	1.96	1.96	1.96
p	0.0012	0.0012	0.03	0.03
B	8.05645E-05	9.54791E-06	0.002636578	0.003954972
n	4.600953054	1.620723506	0.996242993	1.287411017

Tabla 4.9: Datos Simulación App: Random, PkgSize: 512B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μs)	Latency (μs)
6	0.0671366	0.00795344	0.116493	0.197834
5	0.0671364	0.00796231	0.115827	0.20161
4	0.0671169	0.00795815	0.115762	0.198954
2	0.0670515	0.00795665	0.116925	0.202891
10	0.0672859	0.00795283	0.11856	0.205117
8	0.0671018	0.0079642	0.11602	0.201612
stdev	7.88881E-05	4.61203E-06	0.001057811	0.002638301
mean	0.067138183	0.00795793	0.116597833	0.201336333
z	1.96	1.96	1.96	1.96
p	0.0012	0.0012	0.03	0.03
B	8.05658E-05	9.54952E-06	0.003497935	0.00604009
n	3.68326535	0.896053751	0.351321474	0.732949518

Tabla 4.10: Datos Simulación App: BitReverse, PkgSize: 512B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μs)	Latency (μs)
6	0.0671167	0.00797317	0.0898645	0.131713
5	0.0670928	0.00797042	0.0994328	0.153952
4	0.067287	0.00795941	0.0989025	0.15703
2	0.0671297	0.00794928	0.0796521	0.111226
10	0.0671326	0.00796399	0.097888	0.158137
8	0.0670367	0.00795338	0.0888971	0.133294
7	0.0673762	0.00798954	0.090947	0.132685
stdev	0.000119549	1.36029E-05	0.007129387	0.017359803
mean	0.067167386	0.007965599	0.092226286	0.139719571
z	1.96	1.96	1.96	1.96
p	0.0015	0.0015	0.095	0.095
B	0.000100751	1.19484E-05	0.008761497	0.013273359
n	5.408864344	4.979186199	2.543664501	6.571129653

Tabla 4.11: Datos Simulación App: HotSpot, PkgSize: 512B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μs)	Latency (μs)
6	0.0671112	0.00794923	0.0736731	0.0837278
5	0.0670484	0.00796436	0.0702405	0.0840095
4	0.0671063	0.00796425	0.0707252	0.0889056
2	0.067085	0.00795086	0.0715833	0.0927564
10	0.0670837	0.0079514	0.0709125	0.08952
8	0.0670981	0.00794945	0.0704214	0.0836357
7	0.0670697	0.00795191	0.0692727	0.0849632
stdev	2.19061E-05	6.77157E-06	0.001382019	0.003603739
mean	0.067086057	0.007954494	0.070975529	0.086788314
z	1.96	1.96	1.96	1.96
p	0.001	0.001	0.035	0.035
B	6.70861E-05	7.95449E-06	0.002484144	0.003037591
n	0.409615903	2.783977714	1.189014215	5.40704914

Tabla 4.12: Datos Simulación App: Tornado, PkgSize: 512B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μs)	Latency (μs)
6	0.0673445	0.00799389	1.51514	0.51567
5	0.0671345	0.00799492	1.51138	0.528793
4	0.0671509	0.00799738	1.50786	0.522043
2	0.0672506	0.00800407	1.50719	0.534181
stdev	9.75049E-05	4.57715E-06	0.003659849	0.008048858
mean	0.067220125	0.007997565	1.5103925	0.52517175
z	1.96	1.96	1.96	1.96
p	0.0015	0.0015	0.03	0.03
B	0.00010083	1.19963E-05	0.045311775	0.015755153
n	3.59239214	0.559247838	0.025062029	1.002618005

Tabla 4.13: Datos Simulación App: Random, PkgSize: 65536B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0671338	0.00797883	1.39792	0.328061
5	0.0672169	0.0080181	1.39465	0.322767
4	0.0671059	0.00797706	1.40607	0.312189
2		0.00798905		0.323677
10		0.00800833		0.337259
8		0.00798299		0.311023
stdev	5.77423E-05	1.69321E-05	0.00588121	0.009881777
mean	0.0671522	0.007992393	1.399546667	0.322496
z	1.96	1.96	1.96	1.96
p	0.0017	0.0017	0.03	0.03
B	0.000114159	1.35871E-05	0.0419864	0.00967488
n	0.982837116	5.966009944	0.075375164	4.007661738

Tabla 4.14: Datos Simulación App: Neighbor, PkgSize: 65536B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0672625	0.00799928	1.55446	0.611643
5	0.0672618	0.00801075	1.55022	0.605117
4	0.0672407	0.00799567	1.55219	0.610232
2	0.0671883	0.00800271	1.55416	0.615228
stdev	3.48504E-05	6.44184E-06	0.001968627	0.004191135
mean	0.067238325	0.008002103	1.5527575	0.610555
z	1.96	1.96	1.96	1.96
p	0.001	0.001	0.007	0.007
B	6.72383E-05	8.0021E-06	0.010869303	0.004273885
n	1.032032731	2.489566182	0.126018875	3.69428032

Tabla 4.15: Datos Simulación App: BitReverse, PkgSize: 65536B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0671994	0.007995	1.56549	0.52534
5	0.0671903	0.00800166	1.59177	0.550099
4	0.067383	0.00803194	1.59952	0.571698
2	0.067201	0.00799349	1.56815	0.502018
10	0.0672379	0.00800522	1.59104	0.554626
8	0.0671213	0.00800309	1.58954	0.526473
7	0.0674615	0.00801929	1.57188	0.525672
stdev	0.000120675	1.38231E-05	0.013577873	0.023394081
mean	0.067256343	0.008007099	1.582484286	0.536560857
z	1.96	1.96	1.96	1.96
p	0.0014	0.0014	0.035	0.035
B	9.41589E-05	1.12099E-05	0.05538695	0.01877963
n	6.30995691	5.841412563	0.230866389	5.96142471

Tabla 4.16: Datos Simulación App: HotSpot, PkgSize: 65536B

	E	P	E	P
Random-Seed	All Energy (J)	All Energy (J)	Latency (μ s)	Latency (μ s)
6	0.0671693	0.00798253	1.44903	0.373905
5	0.0671009	0.00799408	1.42758	0.404685
4	0.0671575	0.00800383	1.44217	0.377759
2	0.067091	0.00798413	1.43348	0.384779
10	0.0671379	0.00799219	1.43278	0.398339
8	0.0671538	0.00798319	1.43929	0.388029
7	0.0670885	0.00798815	1.43149	0.390422
stdev	3.41615E-05	7.66216E-06	0.007354642	0.010836262
mean	0.067128414	0.007989729	1.436545714	0.388274
z	1.96	1.96	1.96	1.96
p	0.0008	0.0008	0.025	0.025
B	5.37027E-05	6.39178E-06	0.035913643	0.00970685
n	1.554509071	5.520398642	0.161107709	4.787563253

Tabla 4.17: Datos Simulación App: Tornado, PkgSize: 65536B

Una vez ejecutadas las simulaciones, se realizaron pruebas hipótesis de diferencia de medias de la tabla 4.18 para validar o no la hipótesis de que la red de interconexión nanofotónicas híbridas con topología P-Mesh es mejor que su equivalente electrónica E-Mesh. Las variables que se utilizaron como indicador de un mejor sistema fueron el consumo de energía y la latencia del sistema.

Caso	Hipótesis	Estadístico de Prueba	Criterio de Rechazo
$\sigma_1^2 = \sigma_2^2$	$H_0 : \mu_1 = \mu_2$ $H_A : \mu_1 > \mu_2$	$S_p = \sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}}$ $t_0 = \frac{\hat{x}_1 - \hat{x}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$ $v = n_1 + n_2 - 2$	$t_0 > t_{\alpha, v}$
$\sigma_1^2 \neq \sigma_2^2$	$H_0 : \mu_1 = \mu_2$ $H_A : \mu_1 > \mu_2$	$t_0 = \frac{\hat{x}_1 - \hat{x}_2}{S_p \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$ $v = \frac{(\frac{S_1^2}{n_1} - \frac{S_2^2}{n_2})}{(\frac{S_1^2}{n_1})^2 + (\frac{S_2^2}{n_2})^2}$	$t_0 > t_{\alpha, v}$

Tabla 4.18: Estadísticos para Diferencia de Medias. Fuente: [14]

Según el n obtenido para cada caso en las tablas 4.8 a 4.17, se procedió a realizar el análisis de las 5 aplicaciones en 2 subgrupos según el tamaño de los mensajes enviados en cada momento. Adicionalmente, en cada caso, se hizo un analisis de varianza [27] para saber cual de las 2 fórmulas de 4.18 utilizar.

Como el nivel de confianza realizado para la prueba de hipótesis fue del 0.95, se tiene un valor de $\alpha = 0.05$. Por lo tanto sólo las pruebas que presenten un valor inferior a alfa en el valor P se podrá decir con una confiabilidad del 95% que la red nanofotónica híbrida es mejor en alguna de las 2 variables que la electrónica.

La prueba de hipótesis y la alterna para las 2 variables de medición se puede ver en la tabla 4.19. En ambos casos, se busca rechazar la H_0 , es decir confirmar que la red nanofotónica híbrida se comporta mejor en términos de consumo de energía o lantencia que la electrónica.

H_0	H_a	Variable
$CE_E = CE_P$	$CE_E > CE_P$	Consumo de Energía (CE)
$L_E = L_P$	$L_E > L_P$	Latencia (L)

Tabla 4.19: My caption here

Paquetes Pequeños (512B)

En las tablas 4.20 y 4.21 se sintetizan las medias del consumo de energía y de latencia para los mensajes pequeños.

	Electrónica	Híbrida Fotónica
Neighbor	0.06709065	0.007958905
Random	0.0671371	0.007956592
BitReverse	0.067138183	0.00795793
HotSpot	0.067167386	0.007965599
Tornado	0.067086057	0.007954494

Tabla 4.20: Datos promedio Consumo de Energía (J), PkgSize: 512B

	Electrónica	Híbrida Fotónica
Neighbor	0.069082133	0.079211817
Random	0.08788594	0.1318324
BitReverse	0.116597833	0.201336333
HotSpot	0.092226286	0.139719571
Tornado	0.070975529	0.086788314

Tabla 4.21: Datos promedio Latencia (μs), PkgSize: 512B

Se puede observar gráficamente que el consumo de energía de la red nanofotónica híbrida es mucho menor que el de la red netamente electrónica. Sin embargo, la variable de latencia muestra que para este tipo de mensajes la red fotónica híbrida se comporta peor que la red electrónica.

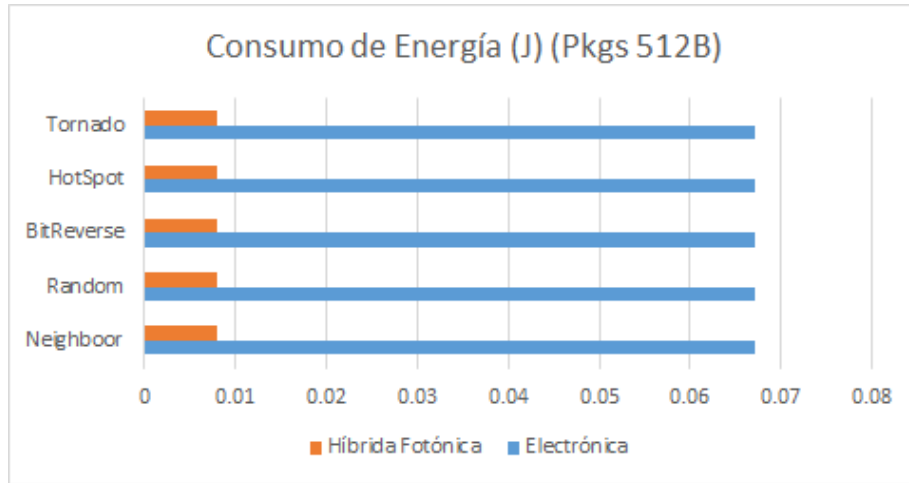


Figura 4.6: Comparación Consumo de Energía, PkgSize: 512B

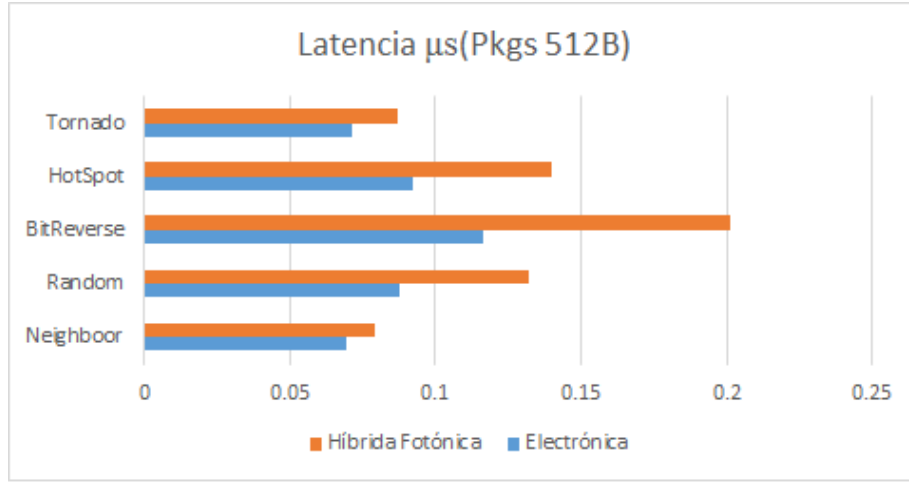


Figura 4.7: Comparación Latencia, PkgSize: 512B

Los resultados de las pruebas de hipótesis se presentan a continuación:

	Valor P	Rechaza H_0 ?
Neighbor	7.29052E-32	TRUE
Random	2.21789E-23	TRUE
BitReverse	2.85143E-29	TRUE
HotSpot	1.42176E-32	TRUE
Tornado	3.3077E-41	TRUE

Tabla 4.22: Resultados Prueba Hipótesis para la variable Consumo de Energía, PkgSize: 512B

En la tabla 4.22 se encontró que en todos los casos se presentó una respuesta positiva. Esto se puede concluir debido a que el valor P para en todos los resultados fue mayor a 0.05 y por lo tanto se puede concluir que la red nanofotónica híbrida sí es mejor en términos de consumo de potencia que la red electrónica analizada para tamaños de mensajes pequeños.

	Valor P	Rechaza H_0 ?
Neighbor	0.99999883	FALSE
Random	1	FALSE
BitReverse	1	FALSE
HotSpot	0.999989	FALSE
Tornado	0.9999999	FALSE

Tabla 4.23: Resultados Prueba Hipótesis para la variable Latencia, PkgSize: 512B

En la tabla 4.23 se puede confirmar lo observado gráficamente para la variable de latencia. En los paquetes de mensajes pequeños, no hay evidencias suficientes para afirmar que la red fotónica es mejor que la electrónica. Es más, en todos los casos se comportó peor.

Este resultado se puede entender desde el punto de vista de la red híbrida, ya que al necesitar un subplano de control electrónico para hacer el *setup* del camino antes de enviar los datos por el subplano fotónico. Cuando los mensajes son pequeños, el sobre costo de realizar este trabajo sobrepasa en gran medida el beneficio de tener una red de transmisión fotónica [26] y por lo tanto no logra el rendimiento esperado.

Tamaño de Paquetes Grande (65kB)

En las tablas 4.24 y 4.25 se sintetizan las medias del consumo de energía y de latencia para los mensajes grandes (65kB).

	Electrónica	Híbrida Fotónica
Neighbor	0.0671522	0.007992393
Random	0.067220125	0.007997565
BitReverse	0.067238325	0.008002103
HotSpot	0.067256343	0.008007099
Tornado	0.067128414	0.007989729

Tabla 4.24: Datos promedio Consumo de Energía (J), PkgSize: 65kB

	Electrónica	Híbrida Fotónica
Neighbor	1.399546667	0.322496
Random	1.5103925	0.52517175
BitReverse	1.5527575	0.610555
HotSpot	1.582484286	0.536560857
Tornado	1.436545714	0.388274

Tabla 4.25: Datos promedio Latencia (μs), PkgSize: 65kB

Nuevamente se puede observar gráficamente que el consumo de energía de la red nanofotónica híbrida es mucho menor que el de la red netamente electrónica. Sin embargo, en este caso, la variable de latencia SI muestra que para este tipo de mensajes la red fotónica híbrida se comporta mucho mejor que la red electrónica.

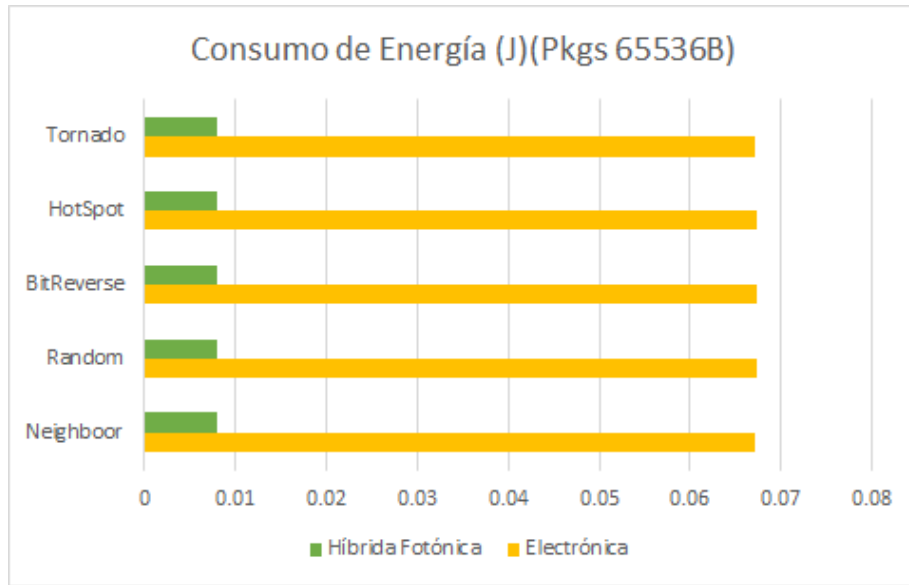


Figura 4.8: Comparación Consumo de Energía, PkgSize: 65kB

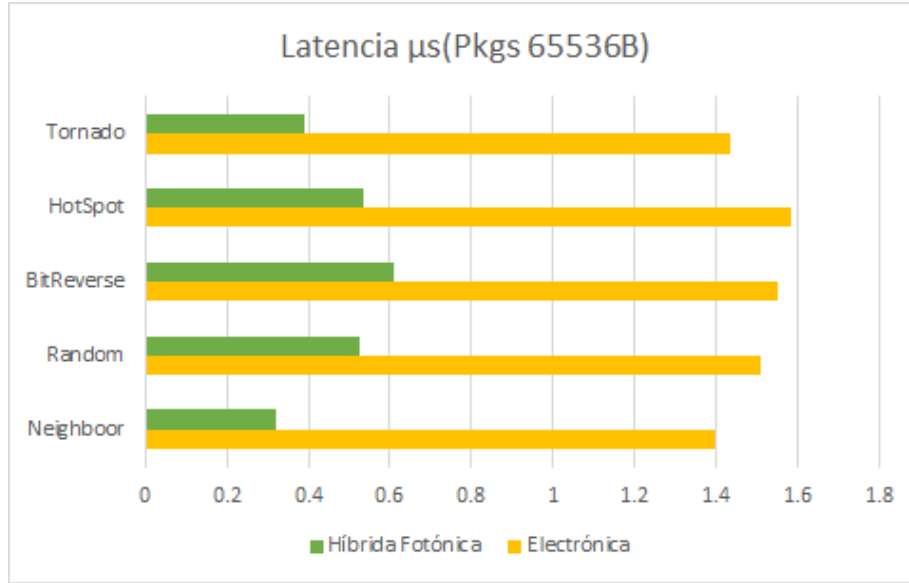


Figura 4.9: Comparación Latencia, PkgSize: 65kB

	Valor P	Rechaza H_0 ?
Neighbor	2.427440e-22 TRUE	TRUE
Random	1.05729E-17	TRUE
BitReverse	2.41872E-20	TRUE
HotSpot	1.57746E-32	TRUE
Tornado	5.3047E-39	TRUE

Tabla 4.26: Resultados Prueba Hipótesis para la variable Consumo de Energía, PkgSize: 65kB

En la tabla 4.26 se puede observar en todos los casos una respuesta positiva debido a que el valor P para todos los resultados fue mayor a 0.05. Por lo tanto, también se puede concluir que la red nanofotónica híbrida sí es mejor en términos de consumo de potencia que la red electrónica analizada para tamaños de mensajes grandes.

	Valor P	Rechaza H_0 ?
Neighbor	7.08709E-14	TRUE
Random	5.37024E-10	TRUE
BitReverse	3.01947E-11	TRUE
HotSpot	2.99509E-16	TRUE
Tornado	8.35414E-21	TRUE

Tabla 4.27: Resultados Prueba Hipótesis para la variable Latencia, PkgSize: 65kB

En la tabla 4.27 también se encontró en todos los casos una respuesta positiva. Es decir que para tamaños de mensajes grandes, el sobre costo de la subred de control es insignificante en cuanto a las mejoras obtenidas en latencia por el subplano fotónico de transmisión de datos.

4.2.3 Conclusiones y Trabajos Futuros

- Para tamaños de mensajes grandes, la red de interconexión nanofotónica híbrida C-Mesh se comporta mejor en términos de consumo de energía y latencia que la red electrónica C-Mesh.
- Para tamaños de mensajes grandes, la red de interconexión nanofotónica híbrida C-Mesh se comporta mejor únicamente en términos de consumo de energía que la red electrónica C-Mesh.
- En los paquetes de mensajes pequeños, no hay evidencias suficientes para afirmar que la red fotónica es mejor que la electrónica en la variable de latencia debido a que el sobre costo de la subred de control electrónica realizar sobrepasa en gran medida el beneficio de tener una red de transmisión fotónica y por lo tanto no logra el rendimiento esperado.
- Se plantea como trabajos futuros, evaluar otras topologías en la red de interconexión, así como ampliar la prueba ejecutando también el subgrupo de aplicaciones científicas.
- Debido a los resultados obtenidos para mensajes pequeños en términos de latencia, sería ideal probar otros tipos de redes implementadas diferentes a las híbridas conmutadas, es decir, que no tengan un subplano de control electrónico. Por ejemplo la TDM arbitrada por longitud de onda descrito en [16] o [28].
- Fortalecer los benchmark con los que se analiza el rendimiento de las

redes electrónicas y nanofotónicas en PhoenixSim, abarcando un espectro más amplio de aplicaciones que involucren áreas como Machine Learning, Software de Base de Datos, Computación Gráfica y Juegos.

Apéndice A

Scripts

A.1 Meep

A.1.1 Filtro Notch

Script A.1: Filtro Notch (Fuente Gaussiana) en Meep.

```
1 ;-----Parameters-----
2 ;Units are measured in microns [a=luc]
3 (define-param odir "./out-")
4 (define-param w 0.4) ; width of waveguide
5 (define-param r 2.9) ; inner radius of ring
6 (define-param gap 0.1) ; gap between ring and wg
7 (define-param pad 4) ; padding between waveguide and edge of
  PML
8 (define-param dpml 2) ; thickness of PML
9 (define-param wavecen 1.550)
10 (define-param wavewid 0.050)
11 (define-param freqcen (/ 1 wavecen))
12 (define-param freq_width (- (/ 1 (- wavecen wavewid)) (/ 1 (+
  wavecen wavewid))))
13 (define-param flux_points 200)
14
15 (define sxy (* 2 (+ r w pad dpml))) ; cell size
16 (define si_mat (make dielectric (index 3.476558)))
17
18 ;-----Geometry-----
19 (set! default-material air)
20
21 (set! geometry-lattice (make lattice (size sxy sxy no-size)))
22 (set! geometry (list
23   (make cylinder
```

```

24         (center 0 0) (height infinity) (radius (+ r w))
25         (material si_mat))
26     (make cylinder
27         (center 0 0) (height infinity) (radius r)
28         (material air))
29     (make block
30         (center 0 (+ r w gap (/ w 2))) (size sxy w infinity)
31         (material si_mat)))
32
33 (set-param! resolution 35)
34 (set! pml-layers (list (make pml (thickness dpml))))
35
36 ;-----Sources-----
37 (set! sources (list
38     (make source
39         (src (make continuous-src (frequency freqcen) ))
40         (component Ez)
41         (size 0 w)
42         (center (+ dpml (* -0.5 sxy)) (+ r w gap (/ w 2)) ) )
43 ))
44
45 ;-----Fluxes-----
46 ;transmitted flux
47 (define iniflux
48     (add-flux freqcen freq_width flux_points
49         (make flux-region
50             (center (+ 1 dpml (* -0.5 sxy)) (+ r w gap (/ w 2)))
51             (size 0 (* 2 w)) )))
52 ;reflected flux
53 (define endflux
54     (add-flux freqcen freq_width flux_points
55         (make flux-region
56             (center (- (* 0.5 sxy) dpml 1) (+ r w gap (/ w 2)))
57             (size 0 (* 2 w)) )))
58
59 ;-----Simulation-----
60 (use-output-directory (string-append odir
61     (get-filename-prefix)))
62 (run-until 3000
63     (at-beginning output-epsilon)
64     (at-end output-efield-z)
65     (at-every 40 (output-png Ez "-S3 -Zc dkbluered -a yarg -A
66         $EPS")))
67 )
68
69 (display-fluxes iniflux endflux)

```

A.1.2 Filtro AddDrop

Script A.2: Filtro AddDrop (Fuente Gaussiana) en Meep.

```

1 ;-----Parameters-----
2 ;Units are measured in microns [a=luc]

```

```

3 (define-param odir "./out-")
4 (define-param w 0.4) ; width of waveguide
5 (define-param r 2.9) ; inner radius of ring
6 (define-param gap 0.1) ; gap between ring and wg
7 (define-param pad 4) ; padding between waveguide and edge of
      PML
8 (define-param dpml 2) ; thickness of PML
9 (define-param wavecen 1.550)
10 (define-param wavewid 0.050)
11 (define-param freqcen (/ 1 wavecen))
12 (define-param freq_width (- (/ 1 (- wavecen wavewid)) (/ 1 (+
      wavecen wavewid))))
13 (define-param flux_points 100)
14
15 (define sxy (* 2 (+ r w pad dpml))) ; cell size
16
17 (define fill_mat air);
18 (define sio2_mat (make dielectric (index 1.44427)))
19 (define si_mat (make dielectric (index 3.476558)))
20
21 (set! default-material fill_mat)
22
23 ;-----Geometry-----
24 (set! geometry-lattice (make lattice (size sxy sxy no-size)))
25 (set! geometry (list
26   ;Outer ring
27   (make cylinder
28     (center 0 0) (height infinity) (radius (+ r w))
29     (material si_mat))
30   ;Inner ring
31   (make cylinder
32     (center 0 0) (height infinity) (radius r)
33     (material fill_mat))
34   ;Upper waveguide
35   (make block
36     (center 0 (+ r w gap (/ w 2))) (size sxy w infinity)
37     (material si_mat))
38   ;Bottom waveguide
39   (make block
40     (center 0 (* -1 (+ r w gap (/ w 2)))) (size sxy w
41       infinity)
42     (material si_mat))
43 ))
44 (set-param! resolution 80)
45 (set! pml-layers (list (make pml (thickness dpml))))
46
47 ;-----Sources-----
48 (define theta (/ pi 4))
49 (define kx (* freqcen (sin theta)))
50 (set! k-point (vector3 kx 0 0))
51 (set! ensure-periodicity true)
52 (define (my-amp-func p)
53   (exp (* 0+2i pi kx (vector3-x p))))
54
55 (set! sources (list
56   (make source

```

```

57     (src (make gaussian-src (frequency freqcen) (fwidth (* 3
58         freq_width)) ))
59     (component Ez)
60     (size 0 w)
61     ; (amp-func my-amp-func)
62     (center (+ dpml (* -0.5 sxy)) (+ r w gap (/ w 2)) ) )
63 ))
64 ;-----Fluxes-----
65 ;in flux
66 (define influx
67     (add-flux freqcen freq_width flux_points
68         (make flux-region
69             (center (+ 1 dpml (* -0.5 sxy)) (+ r w gap (/ w 2)))
70             (size 0 (* 2 w)) )))
71 ;through flux
72 (define thflux
73     (add-flux freqcen freq_width flux_points
74         (make flux-region
75             (center (- (* 0.5 sxy) dpml 1) (+ r w gap (/ w 2)))
76             (size 0 (* 2 w)) )))
77 ;drop flux
78 (define drflux
79     (add-flux freqcen freq_width flux_points
80         (make flux-region
81             (center (+ 1 dpml (* -0.5 sxy)) (* -1 (+ r w gap (/ w
82                 2))))
83             (size 0 (* 2 w)) )))
84 ;-----Simulation-----
85 (use-output-directory (string-append odir
86     (get-filename-prefix)))
87 (run-sources+
88     (stop-when-fields-decayed 50 Ez (vector3 (- (* 0.5 sxy)
89         dpml 1) (+ r w gap (/ w 2))) 1e-3)
90     (at-beginning output-epsilon)
91     (at-end output-efield-z)
92     ; (at-every 40 (output-png Ez "-S3 -Zc dkbluered -a yarg -A
93         $EPS"))
94 )
95 (display-fluxes influx thflux drflux)

```

Script A.3: Generar gráficas de Transmitancia"

```

1  clear ; close all; clc
2
3  arg_list = argv();
4  if nargin < 1
5      printf('Filename not found in args, using default
6          [power.dat]\n');
7      fname = 'power.dat';
8  else
9      fname = arg_list{1};
10 endif

```

```

11  printf('Filename: %s\n', fname);
12
13  power = dlmread(fname, ',');
14
15  x = 1./(power(:,2));
16  yt = 10*log(power(:,4)./power(:,3));
17  yd = 10*log(power(:,5)./power(:,3));
18
19  plot(x,yt,"-bo", x, yd, "-rx");
20  %hold on;
21  %hold off;
22  grid;
23  xlabel('lambda x(10e-6)');
24  ylabel('power');
25  title('Z Transmission spectrum');
26  legend("transmission");
27  set(gca(), "xlim", [min(x),max(x)]);
28  print -dpng graph.png

```

Script A.4: Shell script de invocación con OpenMPI

```

1  #!/bin/bash
2
3  odir=~/ZMeepOut/
4  mkdir $odir > /dev/null 2>&1
5  fname='gausrc_flux_adddrop'
6  rm $odir$fname-power.dat > /dev/null 2>&1
7
8  time ../../zmpimeep $fname $l
9  grep flux1: >> $odir$fname-power.dat $odir$fname.log
10 octave graph_adddrop.m $odir$fname-power.dat > /dev/null 2>&1
11 mv graph.png $fname-graph.png > /dev/null 2>&1

```

A.2 Lumerical

A.2.1 Filtro Notch

Script A.5: Generar la estructura de anillo 3d FDTD

```

1  deleteall;
2
3  addrect;
4  set("x",0);
5  set("x span",lx);
6  set("y span",base_width);
7  set("y",radius+gap+base_width);
8  #copy(0,-2*(radius+gap+base_width),0);
9
10 copy;

```

```

11 set("y",radius);
12 set("x span",Lc);
13
14 copy;
15 set("y",-radius);
16
17 addring;
18 set("theta start",-90);
19 set("theta stop",90);
20 set("inner radius",radius-base_width/2);
21 set("outer radius",radius+base_width/2);
22 set("x",Lc/2);
23
24 copy;
25 set("theta start",90);
26 set("theta stop",270);
27 set("x",-Lc/2);
28
29
30 selectall;
31 set("material",material);
32 if(material=="<Object defined dielectric>") {
33     set("index",index);
34 }
35 set("z min",0);
36 set("z max",height);

```

A.2.2 Filtro AddDrop

Script A.6: Obtener datos de la simulación 3d FDTD

```

1 #####
2 # file: ring_resonator_fDTD.lsf
3 #
4 # Description: This file calculates the results of
5 #             ring_resonator.fsp and saves them to and
6 #             ldf file. The final results are plotted
7 #             using ring_resonator.lsf
8 #
9 # Copyright 2011, Lumerical Solutions, Inc.
10 #####
11
12
13 Tdrop_3DFDTD = -transmission("drop");
14 Tt_3DFDTD = transmission("through");
15 lambda_3DFDTD = c/getdata("drop","f");
16
17 savedata("fDTD_results.ldf",Tdrop_3DFDTD,Tt_3DFDTD,lambda_3DFDTD);

```

Script A.7: Generar gráficas de Transmitancia teórica MODE FDTD

```

1 #####

```

```

2 # file: ring_resonator.lsf
3 #
4 # Description: Send transmission trough ring resonator to
5 # Visualizer. Transmission data comes from 3
6 # different sources: an .ldf with 3D FDTD data,
7 # a MODE Solutions propagator simulation and an
8 # analytical formula
9 #
10 #
11 # Copyright 2012, Lumerical Solutions, Inc.
12 #####
13
14 # FDTD results
15 loaddata("fddt_results.ldf");
16
17 #####
18 # Results from Propagator
19 # ring resonator design parameters
20 lambda0 = 1.55e-6; # center wavelength
21 FSR = 3200e9*lambda0^2/c; # free spectral range in m
22 Q = 2000; # Q
23
24 # get data from propagator simulation;
25 T = getresult("drop", 'T');
26 Tdrop_propagator = -T.T;
27
28 #get data from through simulation
29 Tt = getresult("through", 'T');
30 Tt_propagator = Tt.T;
31
32 lambda = c/T.f;
33 #####
34
35 #####
36 # Calculation of theory
37 tau_11 = sqrt((lambda*pi/FSR/Q/2)^2+1)-(lambda*pi/FSR/Q/2);
38 tau_12 = sqrt(1-abs(tau_11)^2);
39
40 # offset the phase for a particular peak wavelength
41 peak_lambda = 1550e-9;
42 phase_offset = -2i*pi*peak_lambda/FSR;
43
44 Tdrop_theory =
45     abs(tau_12)^4/abs(1-tau_11^2*exp(1i*2*pi*lambda/FSR+phase_offset))^2;
46 Tt_theory =
47     abs(tau_11-tau_11*exp(1i*2*pi*lambda/FSR+phase_offset))^2/abs(1-tau_11^2*exp(1i*2*pi*la
48 #####
49 # Apply to 10xlog10
50 Tdrop_theory = 10*log(Tdrop_theory);
51 Tdrop_propagator = 10*log(Tdrop_propagator);
52 Tdrop_3DFDTD = 10*log(Tdrop_3DFDTD);
53
54 Tt_theory = 10*log(Tt_theory);
55 Tt_propagator = 10*log(Tt_propagator);
56 Tt_3DFDTD = 10*log(Tt_3DFDTD);

```

```

57 #####
58
59 # create dataset
60 T_result = matrixdataset("T_result");
61 T_result.addparameter("lambda", lambda);
62 T_result.addattribute("Tdrop_theory", Tdrop_theory);
63 T_result.addattribute("Tdrop_propagator", Tdrop_propagator);
64 T_result.addattribute("Tdrop_3DFDTD", Tdrop_3DFDTD);
65
66 visualize(T_result);
67
68 T_result_t = matrixdataset("T_result_t");
69 T_result_t.addparameter("lambda", lambda);
70 T_result_t.addattribute("Tthrough_theory", Tt_theory);
71 T_result_t.addattribute("Tthrough_propagator", Tt_propagator);
72 T_result_t.addattribute("Tthrough_3DFDTD", Tt_3DFDTD);
73
74 visualize(T_result_t);
75
76 T_result_fddt = matrixdataset("T_result_3dFDTD");
77 T_result_fddt.addparameter("lambda", lambda);
78 T_result_fddt.addattribute("Tdrop_3DFDTD", Tdrop_3DFDTD);
79 T_result_fddt.addattribute("Tthrough_3DFDTD", Tt_3DFDTD);
80
81 visualize(T_result_fddt);

```

Bibliografía

- [1] Nabeil Abduljallil Abubaker Abujnah. Numerical modelling of optical micro-cavity ring resonators for wdm networks. 2012.
- [2] Krste Asanovic, Ras Bodik, Jim Demmel, John Kubiatoiwicz, Kurt Keutzer, Edward Lee, George Nacula, Dave Patterson, Koushik Sen, John Shalf, John Wawrzynek, and Kathy Yelick. The Landscape of Parallel Computing Research : A View from Berkeley. *Electrical Engineering and Computer Sciences University of California at Berkeley Technical Report No. UCB/EECS-2006-183*, page 54, 2007.
- [3] M. Bass, C. DeCusatis, J. Enoch, V. Lakshminarayanan, G. Li, C. MacDonald, V. Mahajan, and E. Van Stryland. *Handbook of Optics, Third Edition Volume IV: Optical Properties of Materials, Nonlinear Optics, Quantum Optics (set)*. Handbook of Optics. McGraw-Hill Education, 2009.
- [4] Keren Bergman, L P Carloni, J A Kash, and Yurii A Vlasov. On-Chip Photonic Communication for High-Performance Multi-Core Processors. *Proceedings of the Eleventh Annual Workshop on High Performance Embedded Computing HPEC*, pages 2–3, 2007.
- [5] Javier Blasco Solbes. Desarrollo de aplicaciones basadas en la guía nanofotónica de silicio tipo ranura. 2011.
- [6] Wim Bogaerts, Pieter Dumon, Dries Van Thourhout, and Roel Baets. Low-loss, low-cross-talk crossings for silicon-on-insulator nanophotonic waveguides. *Optics letters*, 32(19):2801–2803, 2007.
- [7] Johnnie Chan, Gilbert Hendry, Keren Bergman, and Luca P Carloni. Physical-Layer Modeling and System-Level Design of Chip-Scale Photonic Interconnection Networks. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(10):1507–1520, 2011.
- [8] Johnnie Chan, Gilbert Hendry, Aleksandr Biberman, and Keren Bergman. Tools and methodologies for designing energy-efficient photonic networks-on-chip for highperformance chip multiprocessors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 3605–3608. IEEE, May 2010.

- [9] Johnnie Chan, Gilbert Hendry, Aleksandr Biberman, Keren Bergman, and Luca P Carloni. PhoenixSim: a simulator for physical-layer analysis of chip-scale photonic interconnection networks. pages 691–696, March 2010.
- [10] William James Dally and Brian Patrick Towles. *Principles and practices of interconnection networks*. Morgan Kaufmann, 2004.
- [11] John D Davis, Charles P Thacker, Chen Chang, C Thacker, and J Davis. Bee3: Revitalizing computer architecture research. *Microsoft Research*, 2009.
- [12] M Frigo and Steven G Johnson. The benchFFT, 2013.
- [13] M Gnan, S Thoms, DS Macintyre, RM De La Rue, and M Sorel. Fabrication of low-loss photonic wires in silicon-on-insulator using hydrogen silsesquioxane electron-beam resist. *Electronics Letters*, 44(2):115–116, 2008.
- [14] Humberto Gutiérrez Pulido and Ramón de la Vara Salazar. Análisis y diseño de experimentos. *Editorial McGraw Hill, México*, 2003.
- [15] Gilbert Hendry, Shoaib Kamil, Aleksandr Biberman, Johnnie Chan, Benjamin G Lee, Marghoob Mohiyuddin, Ankit Jain, Keren Bergman, Luca P Carloni, John Kubiawicz, et al. Analysis of photonic networks for a chip multiprocessor using scientific applications. In *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, pages 104–113. IEEE, 2009.
- [16] Gilbert Hendry, Eric Robinson, Vitaliy Gleyzer, Johnnie Chan, Luca P Carloni, Nadya Bliss, and Keren Bergman. Time-division-multiplexed arbitration in silicon nanophotonic networks-on-chip for high-performance chip multiprocessors. *Journal of Parallel and Distributed Computing*, 71(5):641–650, 2011.
- [17] Lumerical. MODE Solutions, Getting Started. Technical report, 2009.
- [18] Inc Lumerical Solutions,. Lumerical FDTD Solutions.
- [19] Nir Magen, Avinoam Kolodny, Uri Weiser, and Nachum Shamir. Interconnect-power dissipation in a microprocessor. In *Proceedings of the 2004 international workshop on System level interconnect prediction - SLIP '04*, page 7, New York, New York, USA, February 2004. ACM Press.
- [20] Phoenixsim User Manual, Gilbert Hendry, and Johnnie Chan. PhoenixSim 1.0 User Manual. pages 0–60.
- [21] MIT Massachusetts Institute of Technology. Meep Introduction - AbInitio. http://ab-initio.mit.edu/wiki/index.php/Meep_Introduction. [Online; accessed Dec-2012].

- [22] MIT Massachusetts Institute of Technology. Meep Tutorial - AbInitio. http://ab-initio.mit.edu/wiki/index.php/Meep_tutorial. [Online; accessed Dec-2012].
- [23] Ardavan F Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, JD Joannopoulos, and Steven G Johnson. Meep: A flexible free-software package for electromagnetic simulations by the fdtd method. *Computer Physics Communications*, 181(3):687–702, 2010.
- [24] George T Paloczi. *Polymer integrated optics: device architectures and fabrication methods*. 2005.
- [25] Valentin Puente, Ramn Beivide, JA Gregorio, JM Prellezo, Jose Duato, and Cruz Izu. Adaptive bubble router: a design to improve performance in torus networks. In *Parallel Processing, 1999. Proceedings. 1999 International Conference on*, pages 58–67. IEEE, 1999.
- [26] Assaf Shacham, Keren Bergman, and Luca P Carloni. Photonic networks-on-chip for future generations of chip multiprocessors. *Computers, IEEE Transactions on*, 57(9):1246–1260, 2008.
- [27] Álvaro TREJOS. *Texto guia-inferencia estadística*. Universidad tecnologica de Pereira., 2004.
- [28] Dana Vantrease, Robert Schreiber, Matteo Monchiero, Moray McLaren, Norman P Jouppi, Marco Fiorentino, Al Davis, Nathan Binkert, Raymond G Beausoleil, and Jung Ho Ahn. Corona: System implications of emerging nanophotonic technology. *ACM SIGARCH Computer Architecture News*, 36(3):153–164, 2008.
- [29] Andras Varga et al. Omnet++ discrete event simulation system. 2005.
- [30] Joseph Thomas Verdeyen. *Laser electronics*. 1989.
- [31] Fengnian Xia, Lidija Sekaric, and Yurii Vlasov. Ultracompact optical buffers on a silicon chip. *Nature Photonics*, 1(1):65–71, 2006.
- [32] Amnon Yariv and Pochi Yeh. *Photonics: Optical Electronics in Modern Communications (The Oxford Series in Electrical and Computer Engineering)*. Oxford University Press, Inc., 2006.