

Reporte Final Silicon Photonics

Tatiana López Guevara

March 14, 2013

*Agradecimientos especiales a JSensei y a mis amigos de Sirius
quienes me apoyaron durante todo el proceso ...*

Índice

1	Definición de la Investigación	7
1.1	Resumen	7
1.2	Planteamiento del Problema	8
1.3	Justificación	8
1.4	Revisión Bibliográfica	9
1.5	Objetivos	11
1.5.1	Objetivo General	11
1.5.2	Objetivos Específicos	11
1.6	Hipótesis	11
1.7	Metodología	11
1.8	Resultados	12
2	Fundamentación Teórica	13
2.1	Introduction	13
2.2	Ecuaciones de Maxwell	13
3	Caracterización de Anillos Resonadores	14
3.1	Introducción	14
3.2	Teoría de Acoplamiento	15
3.2.1	Modelo Teórico de Anillo Resonador	15
3.2.2	Relación entre Coeficientes de Acoplamiento	19
3.2.3	Filtro Notch	21
3.2.4	Parámetros	22
3.3	Simulaciones	24
3.3.1	Diseño del Resonador	24
3.3.2	Lumerical	27
3.3.3	Meep	30
4	Redes de Interconexión	36
4.1	PhoenixSim	36
4.1.1	Librería de Dispositivos de Interconexión	36
4.1.2	Benchmarks	37
4.1.3	Herramientas de Medición y Análisis	37

4.2	Simulaciones	37
5	Conclusiones y Trabajos Futuros	38
A	Scripts	39
A.1	Meep	39
A.1.1	Filtro Notch	39
A.1.2	Filtro AddDrop	40
A.2	Lumerical	43
A.2.1	Filtro Notch	43
A.2.2	Filtro AddDrop	44

Índice de figuras

3.1	Modelo de un Anillo Resonador	16
3.2	Contribución Onda 1 Vuelta	17
3.3	Contribución Onda 2 Vueltas	18
3.4	Modos Resonantes. Fuente[5]	22
3.5	Índice de grupo. Fuente[11]	25
3.6	SetUp del filtro Notch en Lumerical FDTD	27
3.7	Transmitancia simulación 3D FDTD Filtro Notch	28
3.8	SetUp del filtro AddDrop en Lumerical FDTD	28
3.9	Transmitancia en el $puerto_d$ de valores teóricos, simulación MODE y FDTD del Filtro AddDrop	29
3.10	Transmitancia en el $puerto_i$ de valores teóricos, simulación MODE y FDTD del Filtro AddDrop	29
3.11	Transmitancia simulación 3D FDTD Filtro AddDrop	30
3.12	Geometría Filtro Notch.	33
3.13	Transmitancia Filtro Notch	35

Índice de tablas

3.1	Configuración deseada para el filtro	25
3.2	Parámetros	32

Índice de Scripts

3.1	Parámetros Filtro Notch.	31
3.2	Geometría y Materiales Filtro Notch.	33
A.1	Filtro Notch (Fuente Gaussiana) en Meep.	39
A.2	Filtro AddDrop (Fuente Gaussiana) en Meep.	40
A.3	Generar gráficas de Transmitancia”	42
A.4	Shell script de invocación con OpenMPI	43
A.5	Generar la estructura de anillo 3d FDTD	43
A.6	Obtener datos de la simulación 3d FDTD	44
A.7	Generar gráficas de Transmitancia teórica MODE FDTD	44

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.

En este proyecto se pretende explorar y evaluar diferentes diseños de arqui-

tecturas que permitan analizar el impacto de las NoCs ópticas en las futuras generaciones de CMPs, ya que son de vital importancia para lograr el aumento en el rendimiento manteniendo al mismo tiempo la eficiencia en el consumo de potencia.

Al finalizar, se tendrán los análisis comparativos en términos de ancho de banda, potencia y latencia de algunas de las redes de interconexión más conocidas sobre diferentes conjuntos de pruebas. La aplicación de este conocimiento científico orientado al área de arquitectura de computadores y a la nanotecnología, permitirá la creación de nuevos diseños que aporten a la solución de los problemas de rendimiento vs potencia, apuntando a mejorar la competitividad nacional mediante la generación de producción intelectual en dicho aspecto.

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 [12] 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, Pur-

due, MIT entre otras), quienes han logrado demostrar experimentalmente las piezas fundamentales que permitirían su materialización en un futuro no muy lejano.

Para lograr este objetivo, se debe permitir primero a los arquitectos de computadores, a través de la simulación, explorar de forma ágil nuevos diseños de arquitecturas de interconexión y evaluar el rendimiento holístico a través de mediciones precisas en los aspectos de capacidad de procesamiento, consumo de potencia y escalabilidad.

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 reconocida 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 [12], dando surgimiento a una nueva ley similar a la ley de Moore para los transistores.

in 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 [6] 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 semiconductores. 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 [10], 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 las redes nanofotónicas sobre silicio.
- 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 plantea como una investigación aplicada donde se busca 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.

Resulta entonces conveniente la utilización de una metodología estructural, dividiendo la investigación en tres fases que dan solución a la problemática propuesta:

- Fase 1: Base Teórica En esta etapa se hace 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.

- 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.

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:

- Anillo 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 [16].

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 rededor 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.

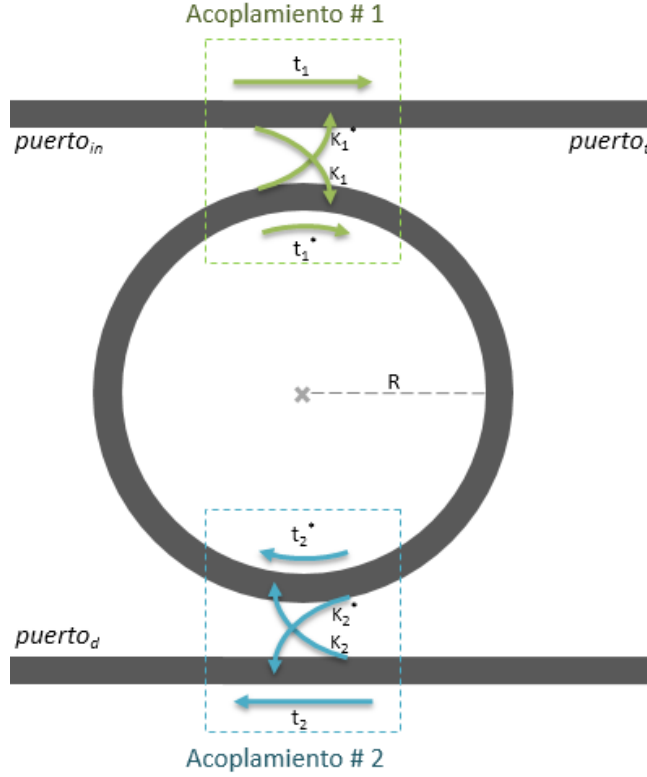


Figura 3.1: Modelo de un Anillo Resonador

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

- 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 completo, 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

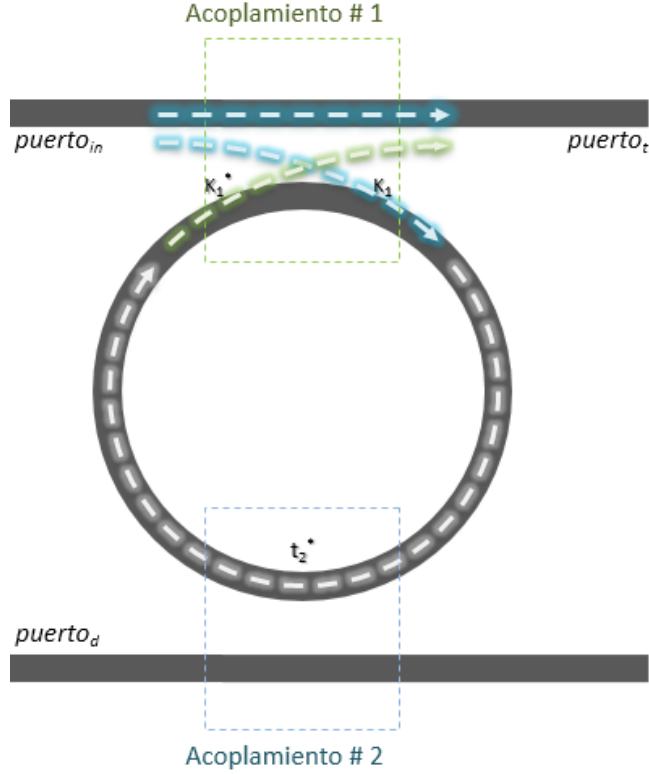


Figura 3.2: Contribución Onda 1 Vuelta

acoplamiento 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.

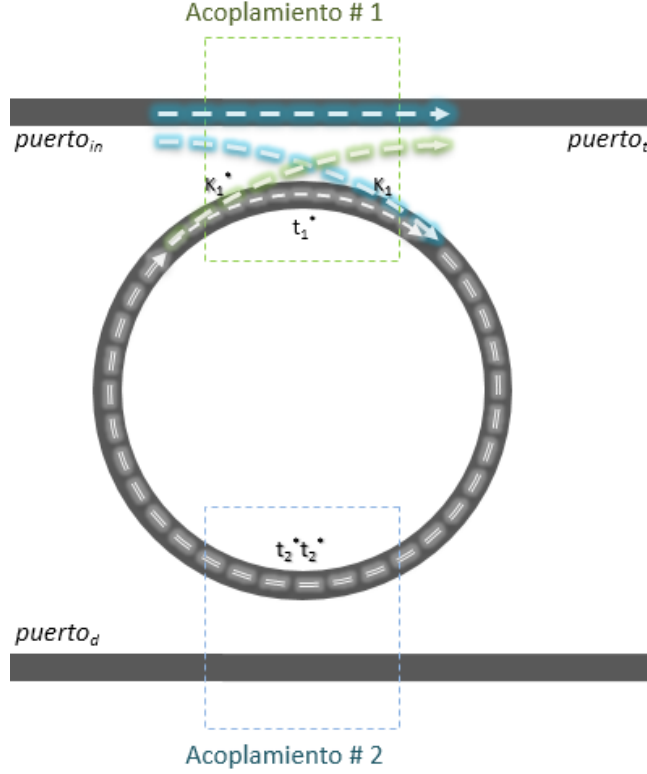


Figura 3.3: Contribución Onda 2 Vueltas

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

- 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 [17], 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 [17] 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 [15]:

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

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 [17]. 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.

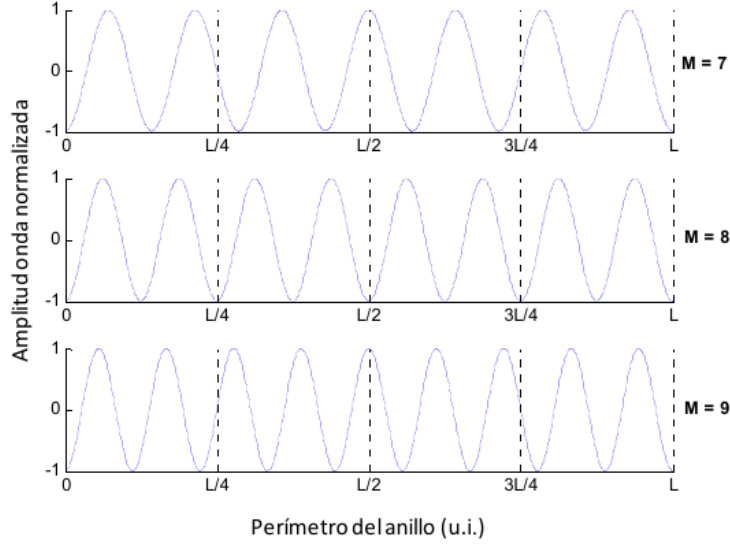


Figura 3.4: Modos Resonantes. Fuente[5]

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) [15].

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

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 espaciamiento 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 [15][16] 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.

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

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 3.1: Configuración deseada para el filtro

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:

3.3.1 Diseño del Resonador

Para el diseño del simulador, se siguieron los lineamientos descritos en [11] 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 3.1.

Índice de grupo n_g

Se calcula el índice de grupo n_g mediante la herramienta MODE de lumérical (sección 3.3.2) 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 $n_g \approx 4.8$.

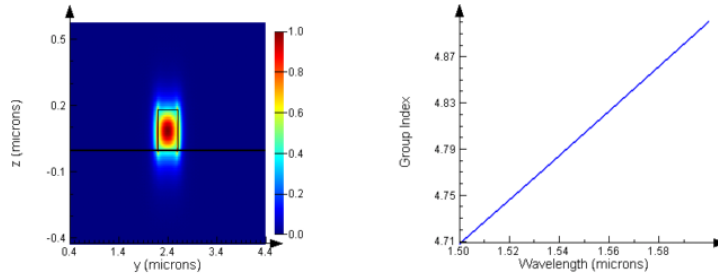


Figura 3.5: Índice de grupo. Fuente[11]

Perímetro del anillo

Despejando L partir de (3.23) y sustituyendo los parámetros (Tabla 3.1), 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 3.1:

$$|t| = \sqrt{\left(\frac{4.819.5e - 6\pi}{220001550e - 9}\right)^2 + 1} - \frac{4.819.5e - 6\pi}{22000\lambda} \approx 0.954$$

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 3.1 se tiene:

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

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

3.3.2 Lumerical

Filtro Notch

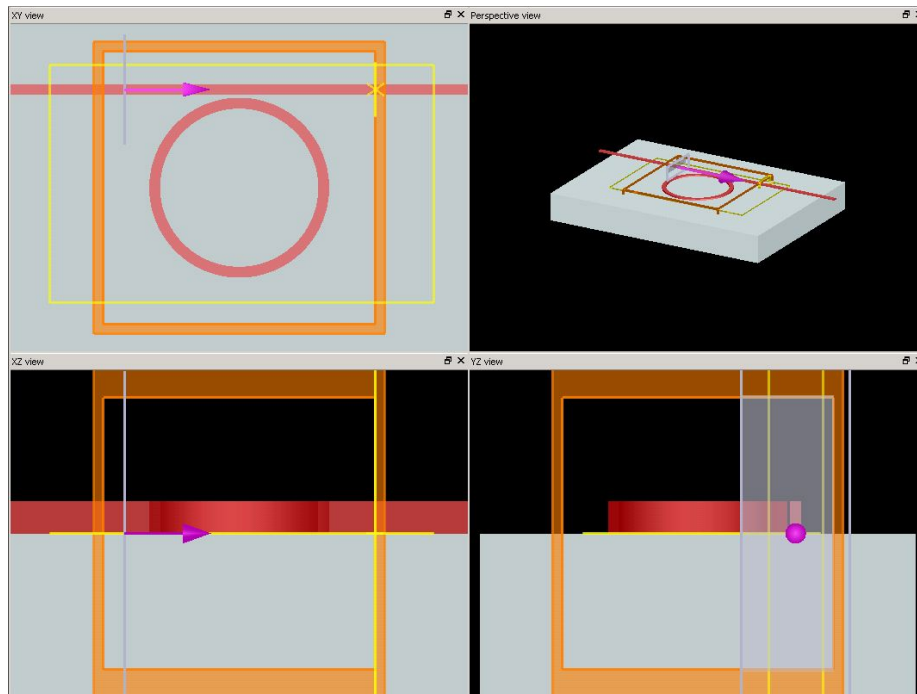


Figura 3.6: SetUp del filtro Notch en Lumerical FDTD

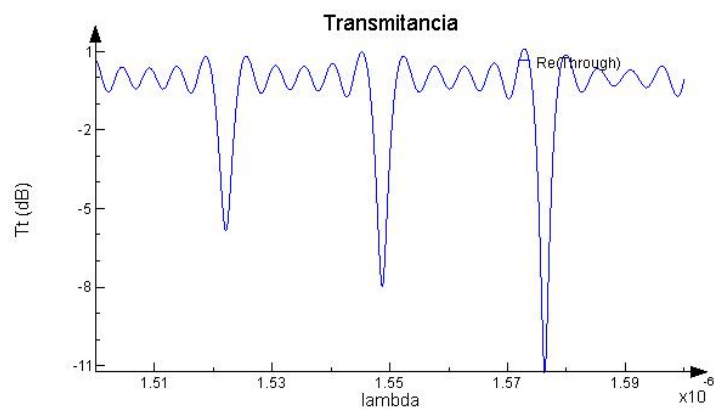


Figura 3.7: Transmitancia simulación 3D FDTD Filtro Notch

Filtro AddDrop

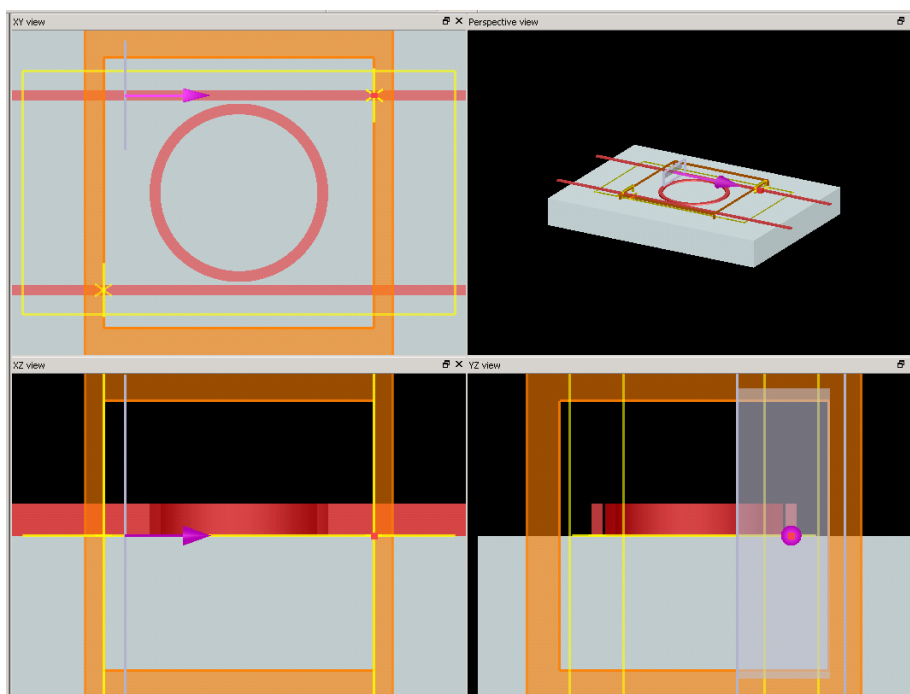


Figura 3.8: SetUp del filtro AddDrop en Lumerical FDTD

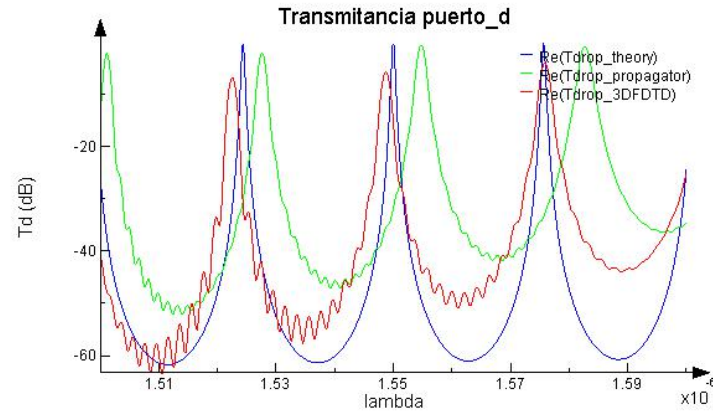


Figura 3.9: Transmitancia en el puerto_d de valores teóricos, simulación MODE y FDTD del Filtro AddDrop

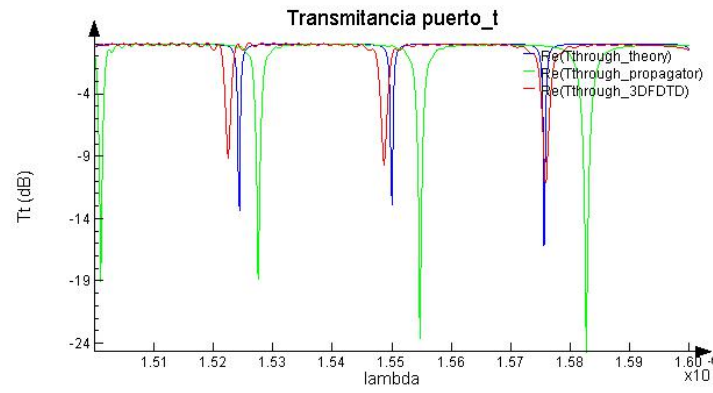


Figura 3.10: Transmitancia en el puerto_t de valores teóricos, simulación MODE y FDTD del Filtro AddDrop

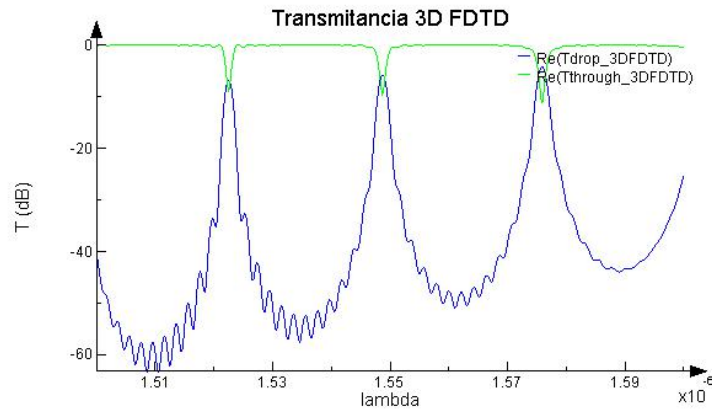


Figura 3.11: Transmitancia simulación 3D FDTD Filtro AddDrop

3.3.3 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:

- 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 wavewid))
13     (/ 1 (+ wavecen wavewid))))
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 3.2.

Parámetro	Descripción	Valor x Defecto
w	Ancho de la guía de onda	4 nm
r	Radio interno del anillo resonador	$2.9 \mu\text{m}$
gap	Espacio entre la guía de onda y el anillo	1 nm
$dpml$	Ancho de la capa PML	$2 \mu\text{m}$
$wavecen$	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 3.2: Parámetros

Materiales y Geometría

En esta sección se emplearon algunos de los parámetros definidos en la tabla 3.2 para construir la geometría del filtro como se ve en la figura 3.12.

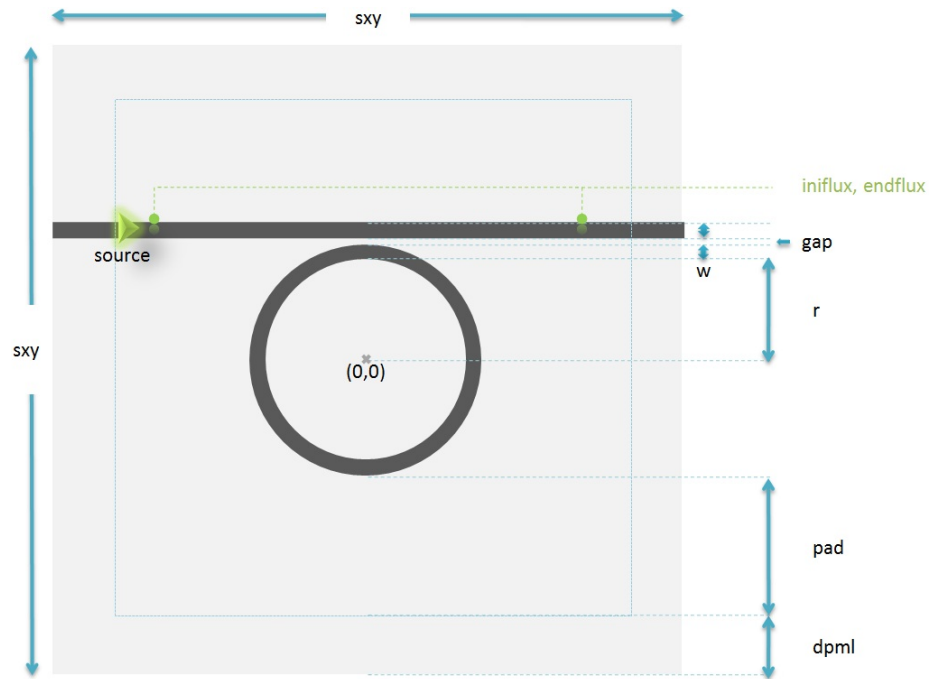


Figura 3.12: Geometría Filtro Notch.

Como se aconseja en [14] 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 3.3.2).

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 [13].

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

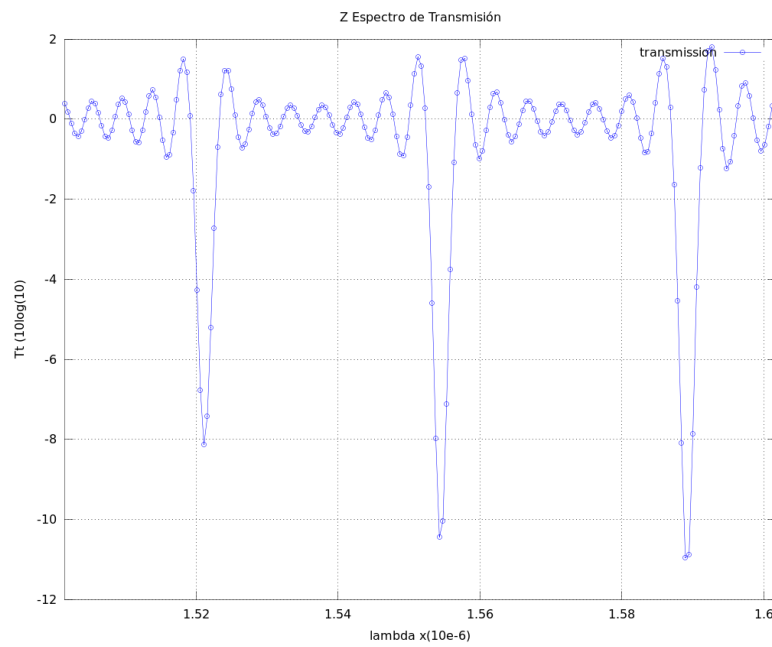


Figura 3.13: Transmitancia Filtro Notch

Capítulo 4

Redes de Interconexión

4.1 PhoenixSim

4.1.1 Librería de Dispositivos de Interconexión

Electrónicos

Fotónicos

- Pasivos
- Activos

4.1.2 Benchmarks

Modelos de Patrones Sintéticos

Modelos de Aplicaciones Reales

4.1.3 Herramientas de Medición y Análisis

4.2 Simulaciones

Capítulo 5

Conclusiones y Trabajos Futuros

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 Kubiataowicz, 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] Johnnie Chan, Gilbert Hendry, Keren Bergman, and Luca P Carloni. Physical-Layer Modeling and System-Level Design of Chip-Scale Photonic Interconnection Networks, 2011.
- [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] John D Davis, Charles P Thacker, Chen Chang, C Thacker, and J Davis. Bee3: Revitalizing computer architecture research. *Microsoft Research*, 2009.
- [11] Lumerical. MODE Solutions, Getting Started. Technical report, 2009.
- [12] 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.
- [13] MIT Massachusetts Institute of Technology. Meep Introduction - AbInitio.
- [14] MIT Massachusetts Institute of Technology. Meep Tutorial - AbInitio.
- [15] George T Paloczi. *Polymer integrated optics: device architectures and fabrication methods*. 2005.
- [16] Joseph Thomas Verdeyen. Laser electronics. 1989.
- [17] Amnon Yariv and Pochi Yeh. *Photonics: Optical Electronics in Modern Communications (The Oxford Series in Electrical and Computer Engineering)*. Oxford University Press, Inc., 2006.