

Hardware-accelerated Opto-Solver for FDTD and BPM Simulations

Abstract—We have developed hardware-accelerated Finite-Difference Time-Domain (FDTD) solver and Beam-Propagation-Method (BPM) solver which use Graphic Cards for parallel computation. These solvers are nearly 4 to 8 times faster than the multi-threaded solvers which use multi-threaded CPU. We employ these solvers to compute wave propagation through the Silicon waveguide coupler. Both FDTD and BPM simulations show that a fraction of mode in the active waveguide enters the adjacent waveguide in the coupled section. Also, We compare computation speed using the GPU accelerated and multi-threaded versions of the solvers with two precision - 32-bit and 64-bit previsions. GPU acceleration yields a speed-up of 8.5 and 1.82, respectively, in FDTD and BPM simulations.

Index Terms—Hardware-acceleration, FDTD solver, BPM solver, Silicon waveguide, coupler.

I. INTRODUCTION

Ever-increasing usage of AI brings forth the necessity of large data-centers. Servers in these data-centers use ultra-fast optical cables for data-transfer. Each of these servers stand to benefit from Silicon photonic modules which can be used for off-chip communications between processors, as well as for data-transceivers. These Si photonic modules are designed and optimized by performing optical simulations using Finite-Difference Time-Domain (FDTD) method [1] and Beam-Propagation-Method (BPM) [2]. These methods are known to be computationally demanding, but are also parallelizable, thereby, reducing computation time. The smaller the simulation time, the faster the iterative optimization procedure would be.

Modern hardware such as Graphics Processing Units (GPU) tend to have a large number of processing cores (called shaders). Running a computationally demanding but parallelizable application on the GPU can deliver performance gains compared to its run on a multi-threaded CPU. However, developing a software which utilizes GPU computation is a complex task and requires in-depth knowledge of Cuda libraries, in case of Nvidia GPUs.

To assist the optical simulation experts with faster runs, we, at SemiVi, have developed hardware-accelerated FDTD solver application [3] which runs on a CPU as well as on a GPU. Together with that, we have also developed hardware-accelerated BPM solver [4] for calculation of steady-state optical response of the photonic Structures. We employ both FDTD and BPM solvers to study a Silicon waveguide coupler structure.

II. THEORY

A. FDTD Solver

Maxwell's equation describe the response of a material system in the absence of electric charges to an electromagnetic excitation (with $\rho = 0$).

$$\nabla \cdot \vec{D} = 0 \quad (1a)$$

$$\nabla \cdot \vec{B} = 0 \quad (1b)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (1c)$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} \quad (1d)$$

Here, $\vec{D} = \epsilon \vec{E}$ is a displacement vector, E is an electric field, $\vec{H} = \vec{B}/\mu$ is the magnetic flux, and $\vec{B}$ is the magnetic field.

Out of Maxwell's equations, Eq. 1a and Eq. 1b are always satisfied by FDTD algorithm. Eq. 1c links spatial variation of electric field vector ($\vec{E}$) with temporal variation of magnetic flux vector ($\vec{H}$) and Eq. 1d links spatial variation of $\vec{H}$ with temporal variation of $\vec{E}$. The LHS of Eq. 1c can be expanded and its x, y, z components can be equated to those of the RHS which gives,

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t}, \quad (2a)$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t}, \quad (2b)$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t}, \quad (2c)$$

Above set of equations form $\vec{H}$ update equations. Similarly, the LHS of Eq. 1d can be expanded to get update equation for electric field. In FDTD algorithm, these two equations (Eq. 1c and Eq. 1d) are alternately solved on a special rectangular (cubic, in 3D) grid called Yee grid at every half-time step. That is, Eq. 1c is solved at time-step i while Eq. 1d is solved at time-step $i + \frac{1}{2}$ [5].

In all the simulations, we use convolutional perfectly matched layer (CPML) boundary condition along all the six faces of the cuboidal boundary surface. A CPML introduces a special region in the first $n = 10$ layers nearest to the boundary. During an $\vec{E}$ update in this special region, an additional displacements coming from convolutional functions is added to it to cancel out the reflections coming from the finite domain boundary at the surface.

B. BPM Solver

In BPM implementation, mode is set to propagate in x-direction. Splitting ∇ operator and $\vec{E}$ into transverse and longitudinal components, we get,

$$\frac{\partial^2 \vec{E}_t}{\partial x^2} + \nabla_t^2 \vec{E}_t + \nabla_t \left[\frac{\partial_t n^2}{n^2} \cdot \vec{E}_t \right] + n^2 k_0^2 \vec{E}_t = 0 \quad (3)$$

. In a waveguide, rapid variation of the transverse fields is the phase variation due to propagation along the propagation axis. One can define a Spatially Varying Envelope (SVE) of the transverse $\vec{E}_t$ as follows,

$$\vec{E}_t = \Psi_y \exp(-in_0 k_0 x) \hat{y} + \Psi_z \exp(-in_0 k_0 x) \hat{z}. \quad (4)$$

The above equation can be used to calculate $\frac{\partial^2 \vec{E}_t}{\partial x^2}$. Substituting SVE of $\vec{E}_t$, and expressions for $\frac{\partial^2 E_y}{\partial x^2}$ and $\frac{\partial^2 E_z}{\partial x^2}$ in Eq. 3, we get

$$2in_0 k_0 \frac{\partial \Psi_y}{\partial x} = \frac{\partial^2 \Psi_y}{\partial y^2} + \frac{\partial^2 \Psi_y}{\partial z^2} + \frac{\partial}{\partial y} \left[\frac{1}{n^2} \frac{\partial(n^2)}{\partial y} \Psi_y + \frac{1}{n^2} \frac{\partial(n^2)}{\partial z} \Psi_z \right] + k_0^2 (n^2 - n_0^2) \Psi_y \quad (5a)$$

$$2in_0 k_0 \frac{\partial \Psi_z}{\partial x} = \frac{\partial^2 \Psi_z}{\partial y^2} + \frac{\partial^2 \Psi_z}{\partial z^2} + \frac{\partial}{\partial z} \left[\frac{1}{n^2} \frac{\partial(n^2)}{\partial y} \Psi_y + \frac{1}{n^2} \frac{\partial(n^2)}{\partial z} \Psi_z \right] + k_0^2 (n^2 - n_0^2) \Psi_z \quad (5b)$$

Eq. 5 form the equations used for propagating the mode along x direction in a BPM solver [6].

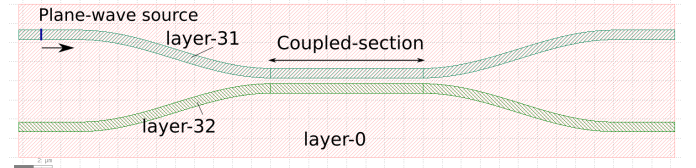
III. SIMULATION SETUP

We use the ‘Structure generator and Mesher’ [7] to create the structure and a regular cubic mesh for FDTD and BPM simulations.

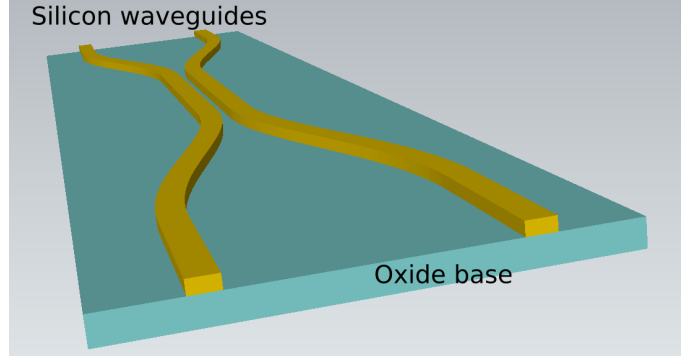
A. Device Structure

The structure consists of two Silicon waveguides which propagate along x-direction. They are coupled in the middle $8\mu\text{m}$ region. Layout of these waveguides is shown in Fig. 1(a). The structure generator takes the GDS layout file as an input and creates the waveguide structure. The structure consists of Oxide base of thickness 480nm created from ‘layer-0’ and two Si waveguides of thickness 240nm created using ‘layer-31’ and ‘layer-32’. Vacuum padding of 240nm is used along y- and z-direction, while that of $1.1\mu\text{m}$ is used along x-direction. The simulated structure is shown in Fig. 1(b).

Mesh spacing of 50nm , 20nm , and 50nm are used along x-, y- and z-directions. This spacing creates a 3D grid of $765 \times 61 \times 251$ grid-points. There are 11,712,915 vertices in this cubic grid.



(a) Layout of the coupler



(b) Coupler structure generated from layout

Fig. 1. (a) Layout of the coupler structure. The shaded red layer is ‘layer-0’, the top waveguide is ‘layer-31’, and the bottom waveguide is ‘layer-32’. (b) Structure of the two waveguides, generated by taking the layout as input.

B. Source

The Si waveguide of ‘layer-31’ is excited by a uniform plane-wave source at 200nm away from the left boundary. The source of intensity 1000W and wavelength of $1.35\mu\text{m}$, is placed in the plane perpendicular to the waveguide and is confined to the cross-section of the waveguide.

C. Boundary Conditions

1) *FDTD Solver*: Convolutional perfectly matched layer (CPML) boundary conditions (BCs) are active at all the boundaries of the simulation domain. The CPML BCs are active on the last five boundary layers parallel to ‘XZ’ and ‘XY’ planes, and on last ten boundary layers parallel to ‘YZ’ plane. All the boundary layers are located in the ‘vacuum’ padding.

2) *BPM Solver*: Perfectly Matched Layer (PML) BC is active at the boundaries parallel to ‘XY’ and ‘XZ’ planes on the nearest five boundary layers. The PML ensures that light incident on the boundary is absorbed in these layers.

IV. RESULTS AND DISCUSSION

A. FDTD Simulations

FDTD simulations are performed on the coupler structure for 500 fsec which corresponds to 8900 time-steps. At time 490 fsec , electric and magnetic fields are post-processed to calculate intensity. Spatial distribution of intensity is plotted in Fig. 2. As seen in the figure, a fraction of light from the active waveguide is transmitted to the adjacent waveguide in the coupling region. Coupling strength can be increased further by bringing the waveguide structures closer to each other.

Absolute electric field at each vertex is averaged from time 450 fsec to 490 fsec . It is plotted in Fig. 3 on linear scale.

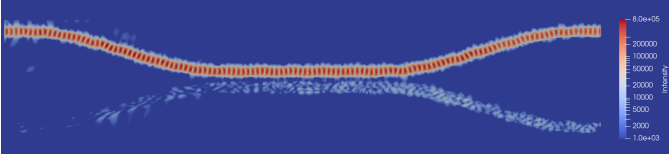


Fig. 2. In FDTD simulations, spatial distribution of intensity in the horizontal cross-section at the middle of the waveguide height (i.e. 120nm above the base-oxide). The figures show that, a tiny fraction of optical power is transferred from the illuminated waveguide to the adjacent one. Note, that a color-map with log scale is used.

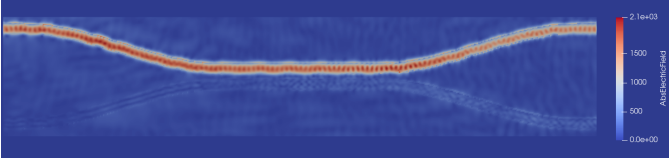


Fig. 3. In FDTD simulations, spatial distribution of abs. electric field in the horizontal cross-section at the middle of the waveguide height (i.e. 120nm above the base-oxide). The figure shows that, a small fraction of the waveguide signal enters the adjacent waveguide in the middle region.

Notice that weak coupling between the active and the adjacent waveguide causes a small field to be present in the adjacent waveguide.

B. BPM Simulations

Envelope functions Ψ_y and Ψ_z are calculated by the BPM solver starting from the source and propagating the fields along positive x-direction. These functions correspond to the amplitude of the electric fields along the waveguide in the ‘steady-state’. Absolute electric field ($\sqrt{E_x^2 + E_y^2 + E_z^2}$) is calculated as post-processing step. Spatial distribution of absolute electric field is plotted in Fig. 4. The figure shows that a small fraction of light entering in the adjacent waveguide, similar to what is observed in FDTD simulations.

C. Speed-up By Hardware-acceleration

Hardware-accelerated FDTD solver and BPM solver are used to speed up the calculations, particularly for large structures such as the coupler structure shown above. Nvidia ‘RTX-3090’ GPU is used as a hardware for the above simulations. To understand usefulness of the GPU, the simulations are also performed using a 4-threaded CPU ‘i3-2100’ without using the GPU. For both the GPU and CPU simulations, the solver software is compiled using two floating point

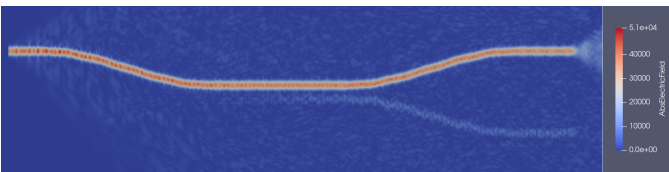


Fig. 4. In BPM simulations, spatial distribution of abs. electric field in the horizontal cross-section at the middle of the waveguide height (i.e. 120nm above the base-oxide). The figure confirms the findings of FDTD simulations.

TABLE I
COMPUTATION TIME OF FDTD SIMULATIONS WITH 64-BIT AND 32-BIT PRECISION

Hardware	64-bit FDTD	32-bit FDTD
CPU	1hr 56m 40s	1hr 30m 5s
RTX-3090	20m 50s	10m 40s
Speed-up	5.60	8.45

TABLE II
COMPUTATION TIMES OF BPM SIMULATIONS WITH 64-BIT AND 32-BIT PRECISION

Hardware	64-bit BPM	32-bit BPM
CPU	2m 2s	2m 26s
RTX-3090	1m 33s	1m 20s
Speed-up	1.33	1.82

precision, namely, 32-bit and 64-bit float. Computation times for the above FDTD simulation on the GPU and the CPU for the two precision are compared in Table I. The table shows that, for both 32-bit and 64-bit precision, the GPU delivers faster performance compared to the 4-threaded CPU. As expected, 32-bit computation takes less time compared to 64-bit computation. Also, the GPU delivers nearly 8.5 times speedup for FDTD simulations compared to the CPU. It is observed, that higher speedup is observed for longer FDTD simulations i.e. the simulations which use more time-steps.

Comparison of BPM simulation time using the GPU and the CPU is presented in Table II. The table confirms that using the GPU delivers faster simulation, but the observed speed-up is not as high as in the case of FDTD solver. The reason seems to be as follows. In the BPM solver, the GPU is used *only* for the sparse matrix system solving step, since it is the most time-consuming step. All the other computations are performed on the CPU. Perhaps, moving all the other computations (apart from matrix solving) to the GPU will deliver better speed-up.

V. CONCLUSIONS

We have presented hardware-accelerated finite-difference time-domain (FDTD) and beam-propagation-method (BPM) solvers for optical simulations. We perform FDTD and BPM simulations of Silicon waveguide coupler structure using these solvers. When only one of the waveguide carries optical signal, the other waveguide receives a fraction of the signal owing to the proximity of the two waveguides in the middle section of the coupler. Both, FDTD and BPM simulations confirm this effect. Additionally, we compare simulation times of FDTD and BPM simulations by using hardware-accelerator - GPU - and by using multi-threaded CPU. In the case of FDTD simulations, use of the GPU achieves speed-up of up to 8.5 times. On the other hand, speed-up of iPod 1.82 times is achieved when the GPU is used for the BPM simulations.

REFERENCES

- [1] J. Young, "Photonic crystal-based optical interconnect," in *IEEE Potentials*, vol. 26, no. 3, pp. 36-41, May-June 2007.
- [2] S. Makino, T. Fujisawa and K. Saitoh, "Wavefront Matching Method Based on Full-Vector Finite-Element Beam Propagation Method for Polarization Control Devices," in *Journal of Lightwave Technology*, vol. 35, no. 14, pp. 2840-2845, 15 July 2017.
- [3] *FDTD Solver User Guide*, SemiVi LLC, Switzerland, 2025.
- [4] *BPM Solver User Guide*, SemiVi LLC, Switzerland, 2025.
- [5] J. Schneider, "Understanding the Finite-Difference Time-Domain Method," www.eecs.wsu.edu/~schneidj/ufdtd, 2010.
- [6] G. Pedrola, "Beam Propagation Method for Design of Optical Waveguide Devices," October 2015, ISBN: 978-1-119-08340-5.
- [7] *Structure Generator and Mesher User Guide*, SemiVi LLC, Switzerland, 2025.