Egy. J. Pure & Appl. Sci. 2023; 61(1):19-27

Egyptian Journal of Pure and Applied Science

An efficient FDTD algorithm based on FPGA

R. Masoud [1], M. S. Kamel [2], M. H. Abdelrazek [1], M. Hameed [3, 4], S. Obayya [5, 6], Ashraf S. E. Yahia [1]

[1] *Physics Department, Faculty of Science, Ain Shams University, Cairo, Egypt.*
[2] *National center for research and radiation technology, Atomic Energy Authority, Cairo, Egypt.*
[3] *Mathematics and Engineering Physics Department, Faculty of Engineering, Mansoura University, Mansoura, Egypt.*
[4] *Centre for Photonics and Smart Materials, Zewail City of Science and Technology, Giza 12588, Egypt.*
[5] *Centre for Photonics and Smart Materials, Zewail City of Science and Technology, Giza 12588, Egypt.*
[6] *Electronics and communication Engineering, Faculty of Engineering, Mansoura University, Mansoura, Egypt.*

ARTICLE INFO

Received 25 September 2022
Accepted 18 December 2022

Keywords

FPGA,
VHDL,
FDTD,
Photonic crystal,
Hardware accelerator.

Correspondence

Rehab Masoud

E-mail

rehabhemdan@sci.asu.edu.eg

ABSTRACT

Designing and modelling of photonic devices using PC simulators take a long time to get the optimum device structures. where a huge number of iterative simulations are required. So, high-speed computers are needed. A design for an FPGA-based finite difference time domain (FDTD) simulator is proposed to accelerate the simulation process where the FDTD method is a remarkably effective computational electromagnetic technique for modelling electromagnetic space. The proposed technique is tested using the analysis of two-dimensional photonic crystal bend structure. The proposed FPGA simulator is 130x faster than the MATLAB program implemented in a PC with a 2.6 GHz processor and 40G RAM. Where, the relative error between FPGA proposed simulator and the MATLAB program was 7.52%. In this study memory architecture, parallelism, pipelining and fixed-point arithmetic have been studied and optimized. Implementing this structure will significantly improve computational speed, allowing it to be used in a wide range of other computational electromagnetics domains.

1. Introduction

Recently, photonics attract great attention in several applications such as solar cell [1], optical communication [2], sensors [3] and etc. As the photonic devices fabrication is expensive, computational numerical techniques are needed to design and optimize the optical device at an early stage with cost savings.

Time-domain (TD) modelling techniques could study computational electromagnetic propagation in linear and nonlinear waveguides. The popular finite difference time domain method (FDTD) is considered a simple simulation method to solve Maxwell's equations with wideband frequency response in a single simulation run [4].

Although, the FDTD method is well-defined and precise, current computer system technology limits the simulation speed. However, simulation run time may take several hours, weeks, months, or longer according simulation time and memory constraints. Slow computation of the FDTD algorithm is due to the nested loops that iterate over the three dimensions space and time. To reduce the simulation time, high performance computer clusters (HPCs) frequently used to speed up the simulation process on the expense of the cost. Therefore, a new approach is needed to increase the speed of FDTD algorithm in a relatively inexpensive and practical way. Recently, several FDTD method accelerators have been proposed [5, 13].

In this regard, different technologies have been used to implement these accelerators such as field-programmable gate array (FPGA) and graphics processing unit (GPU). In this context, Ryan et al. [5] proposed a one-dimensional FDTD accelerator based on pipelined bit-serial circuits and applied on the FPGA. This accelerator was designed for one dimensional resonator. using forward model of the buried object detection. This work was 24x faster than C code that was executed on a 3.0GHz PC. Furthermore, a 3D FDTD architecture was implemented by James et al. [7].

This algorithm was designed for air-filled cavity with speed up factor equals to 8.55 times over a 2.0-GHz PC using C language program. This enhancement of calculation speed was due to improvement of the special nodes handling, data storage and the computation engine. In the other hand, Hasitha et al. [8] used the OpenCL framework [9] to design the 2-D FDTD FPGA-based accelerator. This work achieved speed-up 13 times faster than CPU and 4 times than GPU. Furthermore, to cover wide range of applications Hasitha et al. [10] use stencil computations and introduced an optimization methodology to find the optimal architecture for any OpenCL compatible FPGA board.

The speed up reach 14.1 and 5.1 times than CPU and GPU respectively. In [11], Fujita et al. proposed a FDTD/FIT (finite integration technique) implementation of FPGA based printed-circuit board (PCB) using architecture of parallel memory-access. This work concerned with simulation of microwave phenomena and it make speed up of 7.6 times over the PC (2.4-GHz CPU).

However, in [12] author removed memory access architecture which increased the speed by 50 times higher than GPU performance. After that, these authors [13] duplicated the speed of the computations by modifying his previous architecture in [12] to use all FPGA hardware resources. Conversely, the GPU can be used to enhance the numerical technique over the CPU. Such as in our previous work [14], where the beam propagation method was accelerated by 50x over the CPU.

Numerical simulation of photonic industrial products has been widely introduced in the recent years to reduce fabrication time and cost. However, their simulation time often takes several hours. Thus, the objective of this paper is to design an accelerator for the FDTD method that is applied on the FPGA. In this work, we used highly parallel algorithms to solve differential equations, including finite difference techniques. However, the algorithms are memory-constrained, which makes it challenging to apply them effectively to CPUs or GPUs. Study the implementation of the Finite Time Domain (FDTD) method for solving Maxwell's equations on an FPGA-based data-flow computer. We base our work on actual problems from the field of computational photonics.

The rest of the paper is organized as follows; we begin with a brief discussion about the Finite-Difference Time-Domain (FDTD) Method. Next, we explain the accelerated FPGA architecture for FDTD accelerator. Then, the proposed FPGA-based FDTD hardware design was discussed. Finally, we presented our simulation results and comparing them with current PC-based FDTD solutions.

2. The Finite-Difference Time-Domain (FDTD) Method

FDTD is an accurate, powerful and flexible method to solve electromagnetic problems. The FDTD was introduced by K. Yee method in 1966 [15] since it can solve Maxwell's equations on any scale with practically any environment [16], these equations in differential form can be written [17] as:

$$\nabla \times E = -\frac{\partial B}{\partial t} - M \qquad (1)$$

$$\nabla \times H = \frac{\partial D}{\partial t} + J \qquad (2)$$

$$\nabla \cdot D = \rho_e \qquad (3)$$

$$\nabla \cdot B = \rho_m \qquad (4)$$

Where:

E	Electric field strength vector
D	Electric displacement vector
H	Magnetic field strength vector
B	Magnetic flux density vector
J	Electric current density vector
M	Magnetic current density vector
ρ_e	Electric charge density
ρ_m	Magnetic charge density

Yee's algorithm discretized the Maxwell's equations in space and time interval with identical grids. The space was determined by the central difference approximation. Then, the electric field components and time partial derivatives approximation were calculated at integer time steps. However, the magnetic field components were calculated at half-integer time steps as shown [17] in Fig.1.

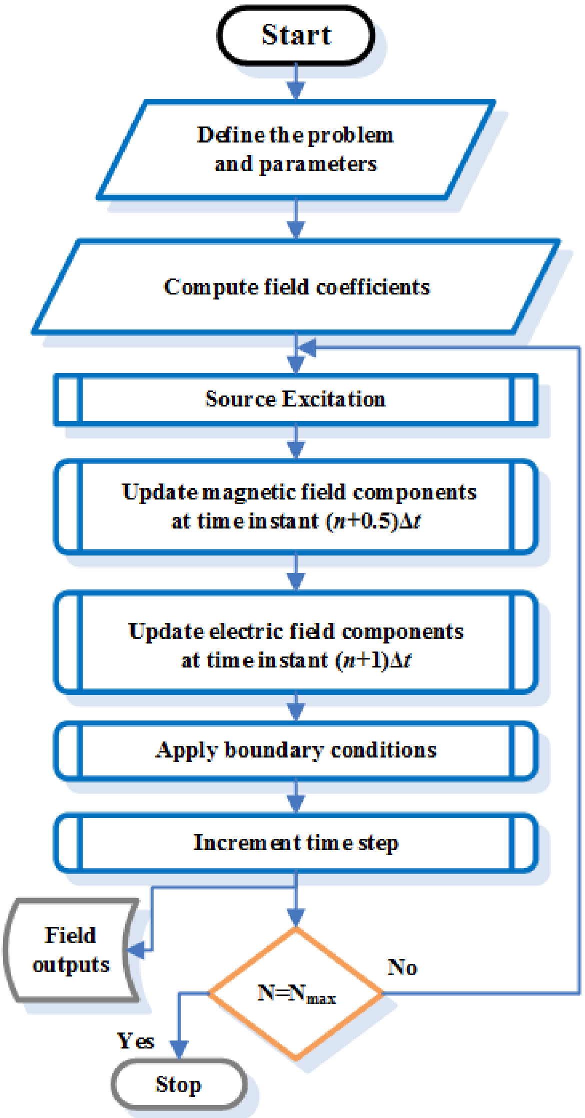

Start
Define the problem and parameters
Compute field coefficients
Source Excitation
Update magnetic field components at time instant $(n+0.5)\Delta t$
Update electric field components at time instant $(n+1)\Delta t$
Apply boundary conditions
Increment time step
Field outputs
$N=N_{max}$
No
Yes
Stop

Fig.1 Flow Diagram of FDTD Method

3. The accelerated FPGA architecture for FDTD

Fig. 2 illustrates the top-level block diagram of the proposed technique. It is composed of a set of on-board DDRAM, a Peripheral Component Interconnect (PCI) and an FPGA chip. In this design, the input data is read from the PCI interface or on-board memories. Further, the computing process is executed inside the FPGA-chip and the results are written back to on-board memories. The FPGA chip design was built using deep pipelines with maximum number of possible parallelism. The parallelism of the design could be controlled by the FPGA chip size, clock speed, on-board and I/O ports.

As shown in Fig. 1 the electric and magnetic fields are computed in series steps, where, The source was injected to structure then the electric and magnetic field component are updated in each step with respect to the field coefficients and appropriate boundary conditions. On the hand, in the proposed technique the fields are computed in two phases. The first one is computed on PC (offline). In the second phase, the electric and magnetic update equations are computed on the FPGA. In the first (offline) phase, the multiplication factor matrices of H and E fields are calculated. These calculations are computed once regardless the number of time steps.

This approach reduced the processing time of the proposed design and saved the FPGA resources. In the previous work [11], each multiplication factors were calculated using FPGA for each cell. Thus, the processing time and memory access were increased. In the second phase, the electric and magnetic fields are computed in a parallel approach. Thus, in the proposed technique, the dependency problem of the electric and magnetic fields is solved as the following manner, we firstly computed one row of the electric field component then we computed in a parallel approach both the remaining electric and magnetic field matrices. Therefore, the computation of the FDTD is increased up to double as shown in Fig. 3.

Also, the cells are computed in the same iteration in parallel as shown in Fig.3 where, there is no data dependency for the computations in the same time step. Thus, the matrix is divided to M parts and we started the calculations of update equations for each part at the same time from down-to-up and from left-to-right as shown in Fig.4. Therefore, the speed of calculations will be increased by M factor. Finally, the update equations of the fields are computed in pipeline approach using the proposed computational modules. Therefore, the FDTD was accelerated by 2*M factor.

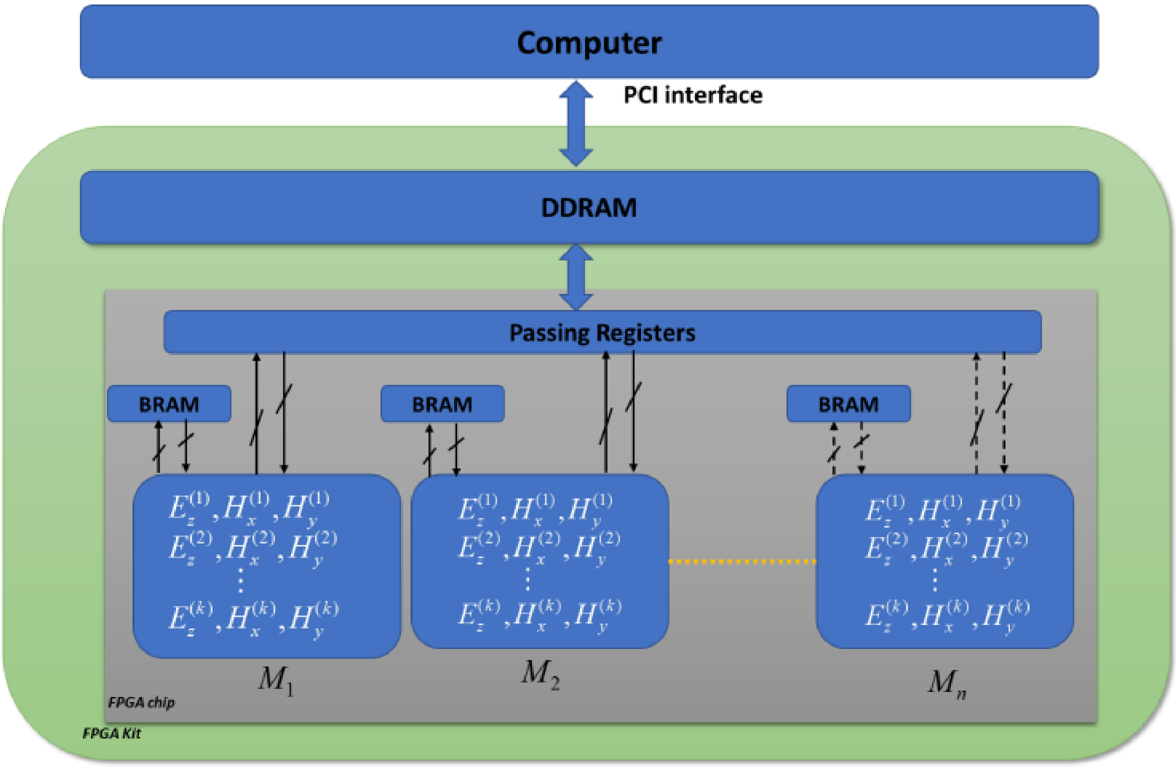

Computer
PCI interface
DDRAM
Passing Registers
BRAM
BRAM
BRAM
$E_z^{(1)}, H_x^{(1)}, H_y^{(1)}$
$E_z^{(2)}, H_x^{(2)}, H_y^{(2)}$
$E_z^{(k)}, H_x^{(k)}, H_y^{(k)}$
M_1
M_2
M_n
FPGA chip
FPGA Kit

Fig. 1 Simple structure diagram of computing board

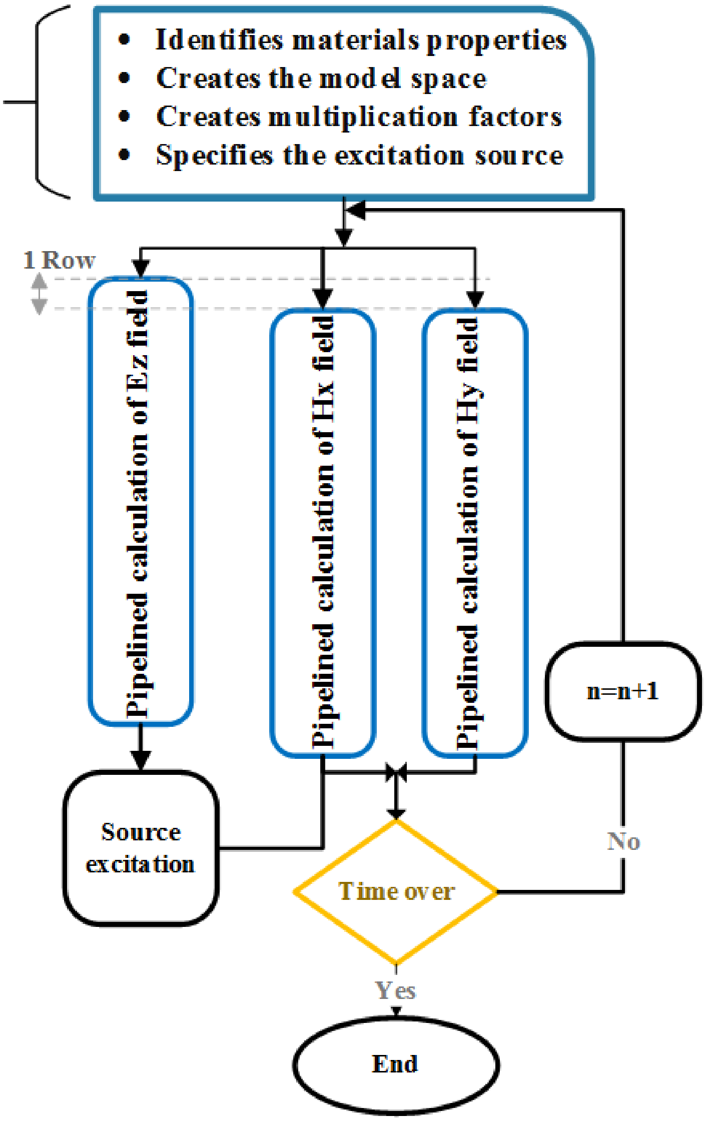

Identifies materials properties
Creates the model space
Creates multiplication factors
Specifies the excitation source
1 Row
Pipelined calculation of Ez field
Pipelined calculation of Hx field
Pipelined calculation of Hy field
n=n+1
Source excitation
No
Time over
Yes
End

Fig. 2 Structural diagram of the paralleled 2D FDTD algorithm

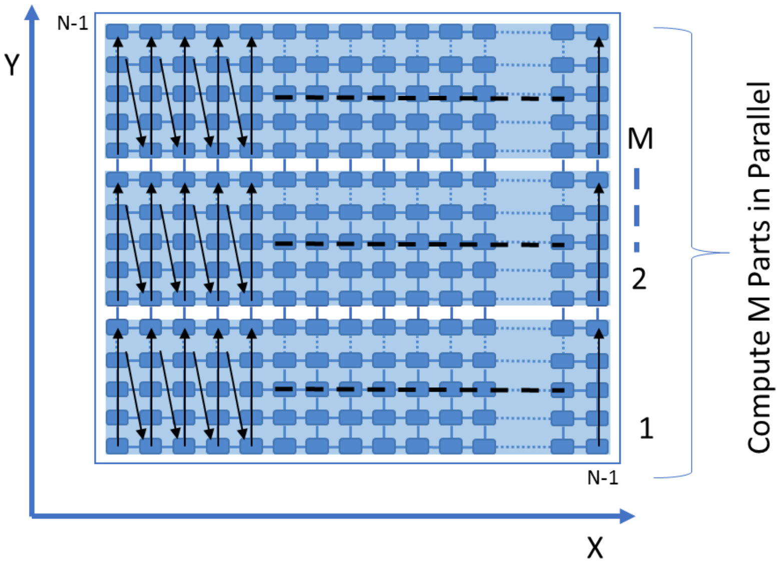


Fig. 4 Compute M parts of matrix in parallel

All calculations of the proposed FDTD algorithm are represented based on two's complement fixed-point format as shown in Fig. 5. This representation increases the speed, reduces the design area and simplifies the arithmetic operations of our design than the double precision floating point. The N-bit signed fixed-point data representation is computed as illustrated equation (5) [18]:

$$X_{fix} = -S2^{a} + \sum_{i=0}^{a-1} 2^{i} A_{i} + \frac{1}{2^{b}} \sum_{i=0}^{b-1} 2^{i} F_{i} \qquad (5)$$

Where:

A Integer part.
F Fraction part.
S Sign of the number.
a Length of integer part.
b Length of fraction part.

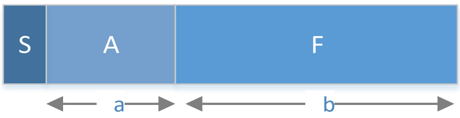


Fig. 5 Fixed-point representation

4. The proposed FPGA-based FDTD hardware design

The proposed hardware design is based on the structural diagram of the paralleled 2D FDTD algorithm presented in Fig.6. In this work, the FPGA hardware device is Xilinx Vertex 7 XC7v2000T. Its specification are 1,954,560 logic cells, 305,400 slices, 46,512 Kb of on chip Block RAM ,1200 I/O pins and Fixed 200 MHz LVDS oscillator [19]. This family is selected due to its high operation frequency, size and I/O ports. The data flow diagram of the hardware design is shown in Fig. 6. The electric and magnetic field matrices are stored in on-board memories while multiplication factor and source signal matrices are stored in Block RAM in order to reduce data transfer to FPGA chip. The processed 2D matrices in the FDTD are transferred to one-dimensional matrix in order to simplify the processing approach.

The values of each field are saved in two memories, one for reading the previous time step values while the other is used to store the current step values. During the data processing in each time step, the two memories of each field will be swapped. To accelerate the proposed technique, reading and saving data from the different memories were handled concurrently with the field calculations.

5. Results and discussion

In order to check the effectiveness of our proposed technique. Firstly, 2D photonic crystal (PC) design was simulated based on the proposed FPGA -FDTD based technique. The structure of the studied photonics crystal device was 7×7 periodic PCs square structures made up silicon with width b = 0.15 (μm) and the separation between horizontal and vertical neighbouring structures is a = 0.45 (μm). In this study, the computational domain is 4.65(μm) × 4.65(μm) as shown in Fig.7.

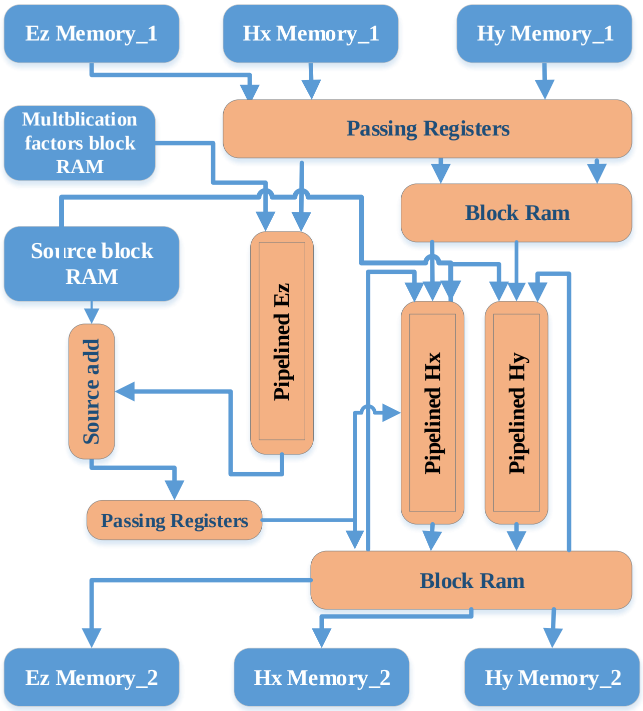


Fig.6 Data flow diagram of the hardware design

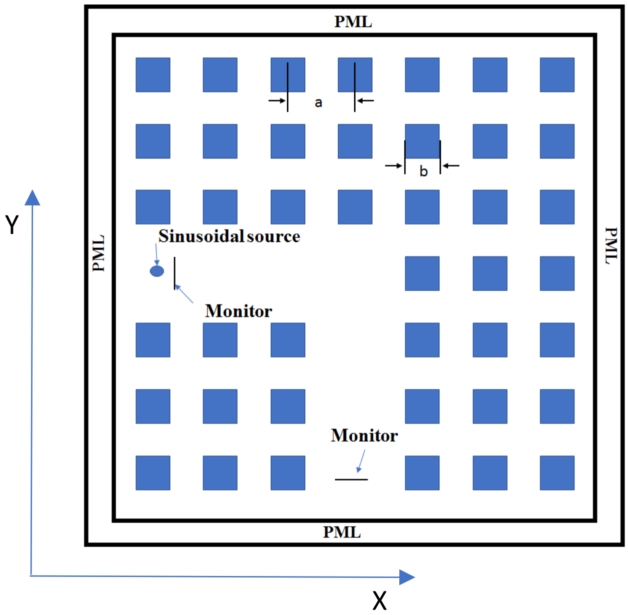
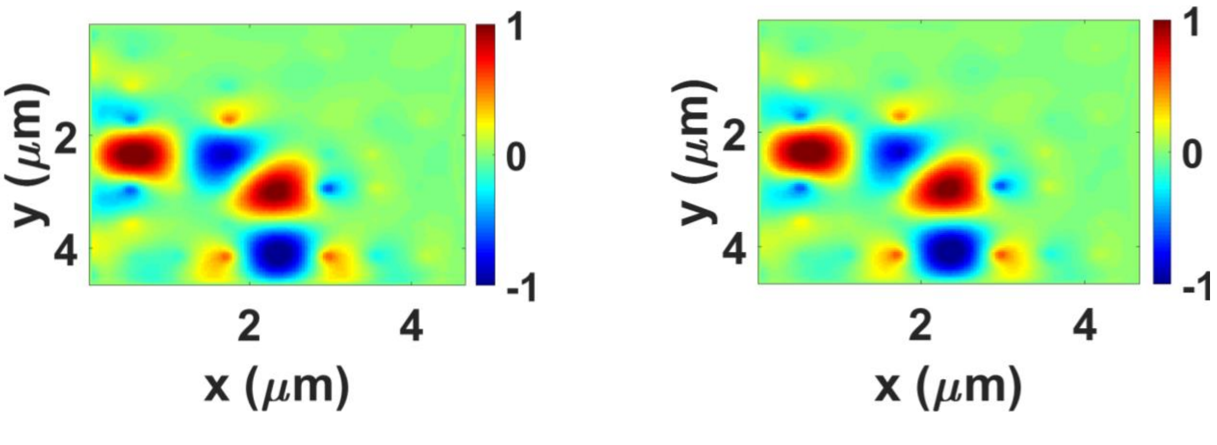


Fig. 7 Numerical model of center 90-degree bent waveguide

The proposed structure is bounded by perfectly matched layer (PML) in x and y-directions and excited by a sinusoidal source with 124×124 total grid space. Such a PC could guide the light through a 90-degree bent defined by removing 4 silicon rods starting from left to center and 3 from center to bottom. Figs. 8 (a) and (b) show the electric field distributions calculated by proposed technique (FDTD algorithm based on VHDL) and by MATLAB software, respectively at λ=1.55μm. It can be seen from these figures that a good agreement is archived between the electric field distribution calculated by MATLAB and VHDL.

One of the most performance parameters in the photonic crystal calculation are S parameters. In order to check the validity of the proposed technique the S parameters of the studied photonic crystal device was calculated. Two-time domain monitors are placed on the structure after the source and at the end of guided wave to calculate S_{11} and S_{21} as shown in Fig.7. Then, the results were compared with the results obtained from MATLAB program solving the same problem on PC. Table 1 illustrates the calculated S parameters values. The relative error in between FPGA and MATLAB is in range of10^{-4}. A good agreement is obtained between the S parameters calculations obtained by MATLAB and VHDL.



(a)

(b)

Fig. 8 Simulated electric field distributions at 300th time step by (a) MATLAB and (b) VHDL at λ=1.55μm.

Next, we will study the effect of the different word lengths (32, 40 and 48 bits) on average absolute and relative errors in electric field calculations where these values in a suitable rang for the studied devices. It is worth mentioning that the optimal word length can be determined according to the studied problem. The average absolute and relative errors in electric field over 1000-time step are illustrated in Table 2. It may be seen from Table 2 that as the word length is increased the relative and average absolute errors are decreased. This is attributed to increasing the precision. On the other hand, the speed up is decreased with increased the word length. This is mainly due to increase calculation time with high precision. It is interesting to note that, the computed values are acceptable in a wide range of the photonic field simulation problems.

It is important to show that, the simulation speed of the proposed technique using FPGA (Vetex 7) is faster than the software implementation using the MATLAB software by factor up to 130 (PC-8 core-2.6 GHz processor - 40G RAM). The speed up factors are illustrated in Table 2. The speedup factors were inversely proportional to the word length due to the reduction in the parallel segments.

The hardware size for the different word lengths are illustrated in Table 3. The average absolute error was in order of $(10^{-5} - 10^{-9})$ where its values depended on word size. This numerical error is a result of two primary factors: first, the calculations error during reduction of the word size multiplication step in the proposed FDTD technique.

Second, the MATLAB software is a double-precision (64 bit) language whereas the proposed technique implemented using 32, 40 or 48 fixed point. However, the speed of simulation based on fixed point is faster than the floating point in the studied problems. If the precision of the simulated problem is important, it can be implemented on FPGA but the speed of the simulation will be reduced. Thus, according to the simulated problem and acceptable range of accuracy, we can select the word length. Finally, the typical computational time for the simulated design increases as the area of such simulated model increases since the value of computed sections M in Fig.4 is limited, minimizing the parallelism of our model.

6. Conclusion

An efficient FPGA based FDTD technique is proposed to accelerate the computations speed of the 2D photonic crystal. To verify the results of the proposed technique, A structure of a photonic crystal was excited using a sinusoidal source and calculated results of FPGA for this crystal were compared with the of MATAB software. The relative error of the s parameter was 2.1046e-04. Different word lengths for the proposed technique calculations using FPGA were used to acquire the best one. Where, the word lengths were 48, and 32 bit the speedup factors were 62, and 130 respectively. While, the relative errors were 5.732×10^{-9}, and 1.022×10^{-5} respectively. Thus, a low error requires a wide word length. As a future work, more complex structures of photonic crystals will be simulated.

Table 1. Hardware size for the different word lengths at λ = 1.55 µm

	MATLAB	VHDL	Relative error
S_{11}	0.13788756	0.13785854	2.1046e-04
S_{21}	0.84384835	0.84358473	3.1240e-04

Table 2. Average absolute, relative errors and speed up of electric field calculation over 1000-time step with different word length

Word width	E_z Average absolute error	E_z Relative error	Speed up
32 bits	1.022×10^{-5}	0.0752	130.2724
40 bits	4.776×10^{-7}	0.00684	64.8693
48 bits	5.732×10^{-9}	8.211×10^{-5}	62.0109

Table 3. Hardware size for the different word lengths

	32 bit	40 bit	48 bit
Slice Registers	506	602	698
Slice LUTs	1613	2454	3757
Logic	1437	2238	3501
Memory	176	216	256
Ram	176	216	256

7. References

1. **M. Hussein, K. R. Mahmoud, M. F. O. Hameed, and S. S. A. Obayya (Nov. 2017).** Optimal design of vertical silicon nanowires solar cell using hybrid optimization algorithm, *J. Photonics Energy*, vol. **8**, no. **2**, pp. **1–14**.
2. **A. M. Ghanim *et al.* (2016).** Highly directive hybrid Yagi-Uda nanoantenna for radition emission enhancement, vol. **0655**, no. **c**, pp. **1–8**.
3. **A. S. Saadeldin, M. F. O. Hameed, S. Member, E. M. A. Elkaramany, S. S. A. Obayya, and S. Member (2019).** Highly Sensitive Terahertz Metamaterial Sensor, IEEE Sens. J., vol. **19**, no. **18**, pp. **7993–7999**.
4. Lumerical: High-Performance Photonic Simulation Software. [Online]. Available: lumerical.com
5. **R. N. Schneider, L. E. Turner, and M. M. Okoniewski (2002).** Application of FPGA technology to accelerate the Finite-Difference Time-Domain (FDTD) method. ACM/SIGDA Int. Symp. F. Program. Gate Arrays - FPGA, pp. **97–105.**
6. **W. Chen, P. Kosmas, M. Leeser, and C. Rappaport (2004).** An FPGA Implementation of the Two-Dimensional Finite-Difference Time-Domain (FDTD) Algorithm Categories and Subject Descriptors, *ACM Trans. Des. Autom. Electron. Syst.*, pp. **213–222.**
7. **F. T. Algorithm, J. P. Durbano, F. E. Ortiz, J. R. Humphrey, M. S. Mirotznik, and D. W. Prather (2003).** Hardware Implementation of a Three-Dimensional, vol. **2**, pp. **54–57.**

8. **H. M. Waidyasooriya and M. Hariyama (2016).** FPGA-based deep-pipelined architecture for FDTD acceleration using OpenCL, 2016 IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. ICIS 2016 - Proc.
9. Altera SDK for OpenCL. [Online]. Available: https://www.altera.com/products/%0Adesign-software/embedded-software-developers/opencl/overview.html.
10. **H. M. Waidyasooriya, Y. Takei, S. Tatsumi, and M. Hariyama (2017).** OpenCL-based FPGA-platform for stencil computation and its optimization methodology, *IEEE Trans. Parallel Distrib. Syst.*, vol. **28**, no. **5**, pp. **1390–1402**.
11. **Y. Fujita and H. Kawaguchi (2009).** Full-custom PCB implementation of the FDTD/FIT dedicated computer IEEE Trans. Magn., vol. **45**, no. **3**, pp. **1100–1103.**
12. **H. Kawaguchi and S. S. Matsuoka (2015).** Conceptual design of 3-D FDTD dedicated computer with dataflow architecture for high performance microwave simulation, IEEE Trans. Magn., vol. **51**, no. **3**, pp. **2–5.**
13. **H. Kawaguchi (2016).** Improved Architecture of FDTD Dataflow Machine for Higher Performance Electromagnetic Wave Simulation**,** IEEE Trans. Magn., vol. **52**, no. **3**, pp. **3–6.**
14. **Shaaban, A., Sayed, M., Hameed, M. F. O., Saleh, H. I., Gomaa, L. R., Du, Y. C., & Obayya, S. S. A. (2019).** Fast parallel beam propagation method based on multi-core and many-core architectures. Optik**, 180, 484-491.**
15. **Kunz, Karl S., and Raymond J. Luebbers (1993).** The finite difference time domain method for electromagnetics. CRC press.
16. **Lyu, Xing-long, Tiexiang Li, Tsung-ming Huang, Jia-wei Lin, Wen-wei Lin, and Sheng Wang (2021).** FAME: Fast Algorithms for Maxwell's Equations for Three-dimensional Photonic Crystals. ACM Transactions on Mathematical Software (TOMS) **47**, no. **3: 1-24**.
17. **A. Z. Elsherbeni and V. Demir (2016).** The finite-difference time-domain method for electromagnetics with MATLAB® simulations: ACES series, 2nd edition.
18. **P. D. Cem Ünsalan and P. D. Bora Tar (2017).** Digital System Design with FPGA: Implementation Using Verilog and VHDL. New York: McGraw-Hill Education.
19. **M. Capability (2016).** 7 Series FPGAs Overview Summary of 7 Series FPGA Features Table 1 : 7 Series Families Comparison Spartan-7 FPGA Feature Summary, vol. **180**, pp. **1–18.**