Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
February 2015, 22(1): 72–76
www.sciencedirect.com/science/journal/10058885
The Journal of China Universities of Posts and Telecommunications
http://jcupt.xsw.bupt.cn

Fast division-free parallel structure for convolution perfectly matched layer in finite difference time domain method

Bai Bing[1], Niu Zhongqi[1] (✉), Niu Yi[1], Wei Bing[2], Zhao Gang[1]

1. School of Electronic Engineering, Xidian University, Xi'an, 710071, China
2. School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, 710071, China

Abstract

Parallel acceleration of convolution perfectly matched layer (CPML) algorithm suffers from massive division operation which is widely accepted as one of the most expensive operations for the equipment such as graphic processing unit (GPU), field programmable gate array (FPGA) etc. In pursuit of higher efficiency and lower power consumption, this article revisited the CPML theory and proposed a new fast division-free parallel CPML structure. By optimally rearranging the CPML inner iteration process, all the division operators can be eliminated and replaced by recalculating the related field updating coefficients offline. Experiments show that the proposed division-free structure can save more than 50% arithmetic instructions and 25% execution time of the traditional parallel CPML structure without any accuracy loss.

Keywords division elimination, convolution perfectly matched layer, finit difference time domain, parallel computing, graphic processing unit

1 Introduction

As an important module of the finite difference time domain (FDTD) theory, absorbing boundary condition (ABC) technique, has been studied for years [1]. A remarkable milestone of the ABC history is the introduction of perfectly matched layer (PML) strategy proposed by Berenger in 1994 [2]. In Ref. [2], the author introduced a hypothetical lossy medium which guarantees that all the plane waves with arbitrary polarization, incidence and frequency are matched at the absorbing boundary.

The PML-based ABC technique significantly improves both the performance and generality of the classical analytically derived ABC techniques. To be specific, PML-based technique gains 30 dB on average than classic analytical ABC techniques in the absorbing performance which approaches the performance in anechoic chamber [1]. Based on Berenger's work, a series of PML techniques were proposed. Sacks et al. and Gedney proposed a set of uniaxial PML (UPML) techniques which derive the relationship of PML and Maxwell equations [3–4]. Kuzuoglu et al. proposed a complex frequency-shifted PML (CFS-PML) technique to improve the evanescent wave absorbing performance of previous PML techniques [5]. By far CFS-PML is widely accepted as the most efficient tool for ABC task. However, the CFS-PML technique has to launch massive convolution operators thus suffering from heavy computational complexity. Roden et al. revealed that the CFS-PML can be accelerated by recursive convolution and proposed a fast algorithms called convolution PML (CPML) [1,6].

In recent years, with rapid development of GPU technology, massively parallel computing based on GPU becomes an important technique in computational electromagnetics. Many researchers proposed a lot of remarkable works on the parallel implementation of FDTD systems [7–9] and the most time-consuming module CPML [10–12].

One of the main bottlenecks of the current GPU-CPML techniques is the massive high-precision (IEEE Std 754–2008 compatible) division operations. In the current GPU architecture, the IEEE Std 754–2008 compatible division operation is implemented by software. Taking the

Received date: 26-10-2014
Corresponding author: Niu Zhongqi, E-mail: zhqniu@xidian.edu.cn
DOI: 10.1016/S1005-8885(15)60627-6

Fermi architecture as an example, a division operator has to launch at least 6 types of other arithmetical instructions listed in Table 1, besides the unpredictable branching instructions. This implies that the computational complexity of division operator is at least 15 times than addition, multiplication or fused-multiply-add (FMA) operation. Therefore, the elimination of division operators becomes a challenge to improve the computational efficiency and power consumption.

Table 1 Minimum computation burden of division operator

Instruction	Times	Execution unit	Throughput*[1]	Instruction explanation
MUFU.RCP	1	SFU*[2]	4	reciprocal operation
FMA	8	CUDA*[3] core	32	fused multiply and add
LOP32I AND	3	CUDA core	32	32bit integer AND
LOP32I OR	1	CUDA core	32	32bit integer OR
IADD32I	1	CUDA core	32	32bit integer add
ISETP	1	CUDA core	16	integer set predicate

*[1] Throughput is considered in number of operations per clock cycle per multiprocessor, in compute capability 2.0 hardware
*[2] Special function unit (SFU)
*[3] Compute unified device architecture (CUDA)

As derived in Sect. 2, the CPML algorithm can be roughly divided into two steps: 1) the calculation of recursive convolution terms (RCT); 2) the CPML field update. All the division operations of CPML algorithm exist in the first step to update the RCT. An optimal iteration arranging strategy is derived to reassign the inner iteration process of RCT calculation. The division operators in step 1 can be combined and shifted to the update of field updating coefficients in step 2. Considering that the field updating coefficients in step 2 can be calculated offline, the strategy avoids launching any division instruction in the online CPML calculation. With the above mentioned optimal iteration arranging strategy. A new fast division-free structure of the GPU-CPML is proposed which can save more than 50% arithmetic instructions and 25% execution time of the traditional parallel CPML structure without any accuracy loss.

The remaining of this article is organized as follows. Sect. 2 introduces our division elimination technique and the division-free CPML (DF-CPML) structure. In Sect. 3, numerical experiments are conducted evaluate the performance of the DF-CPML structure. Sect. 4 highlights the strengths of DF-CPML structure and summarizes the whole paper.

2 Division elimination derivation and DF-CPML structure

Before introduction of our fast DF-CPML structure, it's of advantage to keep in memory of the CPML theory and the current iteration implementation. Without loss of generality, the PML equations can be expressed as complex stretched-coordinate formulations [1,13] to terminate FDTD computation space occupied by lossy media. The stretched-coordinate can be characterized by six factors, three for electric filed components and three for magnetic field components. Considering that all the electric field components have similar expressions and the magnetic field components are in simple duality forms of the electric field components. For simplicity reasons the electric field component E_z is taken as an example. According to PML equation, E_z can be formulated as

$$\mathrm{j}\omega\varepsilon_z E_z + \sigma_z^e E_z = \frac{1}{S_{ex}}\frac{\partial H_y}{\partial x} - \frac{1}{S_{ey}}\frac{\partial H_x}{\partial y} \tag{1}$$

where H_x and H_y are x-component and y-component of magnetic field, respectively. S_{ex} and S_{ey} are stretched coordinate factors, ε_z and σ_z^e are the permittivity and the conductivity of lossy media respectively. To simplify the representation, a second subscript $u=\{x, y, z\}$ is used to denote the related direction. For example, the stretched coordinate factors S_{eu} can be written as [5]:

$$S_{eu} = \kappa_{eu} + \frac{\sigma_{peu}}{\alpha_{eu} + \mathrm{j}\omega\varepsilon_0} \tag{2}$$

where σ_{peu} is the conductivity of PML medium, κ_{eu} and α_{eu} are introduced to deal with evanescent waves and to optimize the absorption of frequency spectra.

Transforming the frequency domain expression Eq. (1) into time domain, the FDTD field updating process can be written as the following form:

$$E_z^{n+1}(i,j,k) = E_z^{n+1}(i,j,k)\Big|_{\mathrm{FDTD}} + \Delta E_z^{n+1}(i,j,k)\Big|_{\mathrm{CPML}} \tag{3}$$

where the first term $E_z^{n+1}(i,j,k)\Big|_{\mathrm{FDTD}}$ is the regular FDTD iteration result which has been well-studied [10–12]. Our investigation focuses on the acceleration of the second term $\Delta E_z^{n+1}(i,j,k)\Big|_{\mathrm{CPML}}$ which denotes the recursive convolution result of CPML. According to [6], $\Delta E_z^{n+1}(i,j,k)\Big|_{\mathrm{CPML}}$ can be calculated by:

$$\Delta E_z^{n+1}(i,j,k)\Big|_{\mathrm{CPML}} = C_b(\psi_{hyx}^{n+1/2}(i,j,k) - \psi_{hxy}^{n+1/2}(i,j,k)) \tag{4}$$