مجله مهندسی مکانیک مدرس، فروردین 1396، دوره 17، شماره 1، ص‌ص 67-74

ماهنامه علمی پژوهشی

مهندسی مکانیک مدرس

mme.modares.ac.ir

ارزیابی قابلیت سخت‌افزار FPGA به‌عنوان راهکاری جدید در تسریع محاسبات عددی حل مساله لاپلاس

عباس ابراهیمی[1*]، محمد زندسلیمی[2]

1- استادیار، مهندسی هوافضا، دانشگاه صنعتی شریف، تهران
2- دکتری تخصصی، مهندسی هوافضا، دانشگاه صنعتی شریف، تهران
* تهران، صندوق پستی 11155-1639، ebrahimi_a@sharif.ir

اطلاعات مقاله

مقاله پژوهشی کامل
دریافت: 05 شهریور 1395
پذیرش: 25 آذر 1395
ارائه در سایت: 15 دی 1395

کلید واژگان:
سخت‌افزار قابل برنامه‌ریزی
محاسبات عددی
دینامیک سیالات محاسباتی
زبان توصیف سخت‌افزار
تسریع محاسبات عددی

چکیده

امروزه کاهش زمان محاسبات عددی از چالش‌های اصلی در شبیه‌سازی مسایل واقعی سیالاتی به‌شمار می‌رود. حل عددی معادلات دیفرانسیل در بستر سخت‌افزارهایی مانند سی.پی.یو و جی.پی.یو. براساس پیاده‌سازی نرم‌افزاری هستند. در این مقاله قابلیت سخت‌افزار اف.پی.جی.ای به‌عنوان راه‌کاری جدید برای حل عددی معادلات دیفرانسیل در دینامیک سیالات محاسباتی ارزیابی شده است. سخت‌افزاری اف.پی.جی.ای می‌تواند حل معادلات را از حالت سری خارج کرده و موازی‌سازی آن‌ها را براساس پیاده‌سازی سخت‌افزاری انجام دهد. اف.پی.جی.ای یک مدار تجمعی از بلوک‌های منطقی است و طراحی آن به شکلی انجام شده است که ساختار آن (عملکرد بلوک‌های منطقی و اتصالات و سیم‌کشی بین آن‌ها) بعد از تولید سخت‌افزار، قابل پیکربندی دوباره است. به این ترتیب می‌توان مدارهای بسیار پیچیده را با استفاده از برنامه‌ریزی این سخت‌افزار با اهداف مختلف تولید کرد. برای قابلیت‌سنجی این سخت‌افزار، حل عددی معادله لاپلاس به‌عنوان نمونه‌ای از مسایل سیالاتی روی یک اف.پی.جی.ای خاص پیاده‌سازی شده و نتایج حل عددی آن با نتایج گرفته‌شده از برنامه‌ی کامپیوتری روی سی.پی.یو مقایسه شده است. نتایج محاسبات حاکی از تسریع حدود چهار برابر برای دقت مضاعف و تسریع تا 20 برابر برای دقت تکی در مقایسه با حل روی سخت‌افزار سی.پی.یو است به‌گونه‌ای که دقت حل نیز کاهش پیدا نکرده است. برای تعیین درستی نتایج از نظر زمان و دقت، از حل‌های عددی و تحلیلی مختلفی استفاده شده است.

Evaluation of FPGA hardware as a new approach for accelerating the numerical solution of the Laplace problem

Abbas Ebrahimi*, Mohammad Zandsalimy

Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran
* P.O.B. 11155-1639, Tehran, Iran, ebrahimi_a@sharif.ir

ARTICLE INFORMATION

Original Research Paper
Received 26 August 2016
Accepted 15 December 2016
Available Online 04 January 2017

Keywords:
FPGA
CFD
Reconfigurable Hardware
Hardware Definition Language
Accelerating Numerical Solutions

ABSTRACT

Nowadays one of the main challenges facing fluid dynamics simulations is the long duration of numerical calculations. Mathematical operations in numerical solution of differential equations using traditional hardware such as CPU, are done in a series of orderly calculations and therefore take a lot of time to complete. A new solution procedure for numerical calculations is presented using FPGA (Field Programmable Gate Arrays) hardware, which will enable parallel processing inside the hardware. The main goal of this research is to use FPGAs instead of CPUs for numerical solution of the Laplace equation and therefore, accelerate its solution. FPGA is an integrated circuit containing a number of logic blocks. The architecture of this hardware can be reprogrammed and configured after manufacturing. So, it is possible to design and implement complex circuits for various applications using an FPGA. In the present research, first, the ability of FPGAs in mathematical operations on floating point numbers is studied. Then, the Laplace problem is implemented and solved numerically on a specific FPGA hardware using different mesh size and numerical methods. The time duration and precision results of the calculations are compared to the results from a CPU. The calculation procedure on the FPGA is up to twenty times faster than a conventional CPU, with the same data precision. Several numerical and analytical solutions are used to validate the results.

1- مقدمه

دینامیک سیالات محاسباتی (سی.اف.دی[1]) نقشی اساسی در بخشی از مراحل طراحی و بهینه‌سازی بسیاری از مسایل سیالاتی صنعتی و مهندسی هوافضا دارد. همچنین از سی.اف.دی برای مطالعه مفاهیم فیزیکی جریان و

[1] Computational Fluid Dynamics (CFD)

Please cite this article using:

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

A. Ebrahimi, M. Zandsalimy, Evaluation of FPGA hardware as a new approach for accelerating the numerical solution of the Laplace problem, *Modares Mechanical Engineering*, Vol. 17, No. 1, pp. 67-74, 2017 (in Persian)

[Downloaded from mme.modares.ac.ir on 2024-04-26]

[DOR: 20.1001.1.10275940.1396.17.1.13.3]

شبیه‌سازی آن‌ها مانند لایه‌مرزی، امواج ضربه‌ای[1]، آشفتگی[2] جریان و غیره استفاده می‌شود. با این‌حال هزینه‌های محاسباتی بالا باعث شده که نتوان از این دانش به‌عنوان ابزاری که کل فاز طراحی برعهده آن باشد، استفاده کرد. هزینه‌های محاسباتی بالا از حل معادلات دیفرانسیل حاکم بر حرکت سیال هستند ناشی می‌شود. این معادلات از قوانین فیزیکی بقای جرم، مومنتوم (نویر-استوکس[3]) و انرژی مشتق می‌شوند و به‌جز در موارد خاص و ساده‌شده دارای حل تحلیلی نیستند. در بسیاری از مسایل واقعی، حل عددی این معادلات به شبکه‌های محاسباتی ریز و یا تعداد تکرارهای حل زیاد نیاز دارد، بنابراین حتی با وجود کامپیوترهای بسیار قوی نیز زمان موردنیاز برای حل افزایش پیدا می‌کند. به‌همین دلیل محققین علوم مهندسی تلاش زیادی برای تسریع حل الگوریتم‌های حل معادلات دیفرانسیل حاکم بر جریان انجام داده‌اند. نتایج پژوهش‌های انجام شده در این زمینه به محاسبات موازی[4]، استفاده از تراشه‌های چند هسته‌ای، محاسبات با استفاده از تراشه‌های ویژه مانند جی.پی.یو[5] [1] و غیره منجر شده است.

سی.پی.یو و جی.پی.یو به‌عنوان شناخته‌شده‌ترین پردازشگرهای مورد استفاده، سخت‌افزارهایی هستند که قابلیت تغییر در ساختار آنها پس از ساخت وجود ندارد، به این معنی که این سخت‌افزارها تنها کاری را انجام می‌دهند که در زمان طراحی مدنظر بوده و دیگر توانایی تغییر در پیکربندی آن وجود ندارد.

برخی تراشه‌ها مانند اف.پی.جی.ای[6] قابل‌پیکربندی هستند و می‌توان ساختار این سخت‌افزارها را در حین اجرای برنامه تغییر داد. در بررسی‌های اخیر [2]، به‌نظر می‌رسد استفاده از تراشه اف.پی.جی.ای که براساس پیاده‌سازی سخت‌افزاری معادلات است برای تسریع محاسبات بسیار مناسب باشد. طبیعت موازی‌سازی محاسبات عددی در سطح سیستم یعنی استفاده از چند واحد سخت‌افزاری و در سطح سخت‌افزار یعنی پیکربندی یک واحد سخت‌افزاری، هر دو در مورد اف.پی.جی.ای قابل پیاده‌سازی است. امروزه با افزایش فرکانس سیگنال زمانی اف.پی.جی.ای‌ها، می‌توانند به‌عنوان پردازشگرهایی با قابلیت انعطاف بالا و به تنهایی استفاده شوند. در گذشته بیشتر تحقیقاتی که در زمینه‌ی محاسبات ریاضی با استفاده از اف.پی.جی.ای انجام شده، مربوط‌به کاربردهایی مانند ذخیره ماتریس‌ها و ضرب آن‌ها، تبدیل فوریه عددی، پردازش سیگنال و کاربردهای دیجیتال دیگر بوده است.

در زمینه سی.اف.دی فعالیت‌های پژوهشی اندکی وجود دارد که از جمله آن‌ها، هاوسر در [3] المان‌های اصلی یک حل عددی مانند ماتریس‌ها، بردارها و عملیات ریاضی روی آن‌ها از جمله ضرب داخلی، ضرب ماتریسی و غیره را روی سخت‌افزار اف.پی.جی.ای پیاده‌سازی کرد و سپس از این خواص برای ساخت یک حلگر جریان یک‌بعدی تراکم‌پذیر صریح استفاده کرد.

سطوح موازی‌سازی ممکن در محاسبات روی اف.پی.جی.ای، شامل موازی‌سازی در پایین‌ترین سطح محاسبات (عملیات اصلی ریاضی)، موازی‌سازی محاسبات توابع پیچیده‌تر و نحوه شبکه‌بندی میدان و بهینه‌سازی الگوریتم حل توسط نونز در [4] شرح داده شده است.

سانو و همکاران [5] با برنامه‌ریزی یک سخت‌افزار اف.پی.جی.ای به‌صورت واحدهایی که قابلیت انجام ضرب و جمع را دارند، توانستند مساله دوبعدی حفره را در یک شبکه 24 در 24 حل کنند و در فرکانس کارکرد 60 مگاهرتز به سرعت نهایی حل حدود 7 برابر سرعت حل در یک سی.پی.یو 3.2 گیگاهرتزی رسیدند.

اندرس و همکاران در [6]، توسعه یک الگوریتم برای افزایش سرعت محاسبات در اف.پی.جی.ای را امکان‌سنجی کرده و سپس روشی کاربردی برای حل معادلات شبه-یک‌بعدی اویلر در مساله لوله موج ضربه‌ای ارائه کردند. همچنین در [7] فرآیندهای مختلف حل معادلات رانس[7] از نظر زمان مورد نیاز بررسی شده و سپس با بهینه‌سازی کد، موازی کردن محاسبات و استفاده از سخت‌افزارهای جدید مانند جی.پی.یو و اف.پی.جی.ای، حل با سرعت بالاتری انجام شد.

در [8] معادلات اویلر به روش حجم محدود با استفاده از یک زبان سطح بالا (زبان برنامه‌نویسی سی[8]) بر روی اف.پی.جی.ای پیاده‌سازی شده و تا حدود سیزده برابر، افزایش سرعت مشاهده شده است. در [9] اثر استفاده از اف.پی.جی.ای در افزایش سرعت محاسبات یک مساله سی.اف.دی خطی که به شکل تکراری حل می‌شد، بررسی شده‌است سپس با افزایش تعداد اف.پی.جی.ای‌هایی که هم‌زمان مورد استفاده قرار گرفته‌اند، مشاهده شد که سرعت حل به‌صورت خطی افزایش می‌یابد. همچنین لیو و همکاران [10] در زمینه‌ی احتراق، حل عددی جریان یک‌بعدی احتراقی داخل لوله‌های موتور بنزینی با استفاده از اف.پی.جی.ای را انجام دادند.

هدف پژوهش حاضر، ارزیابی قابلیت محاسباتی سخت‌افزار اف.پی.جی.ای و پیچیدگی‌های برنامه‌نویسی آن برای افزایش سرعت حل عددی معادلات دیفرانسیل حاکم بر مسایل سیالاتی است. به این منظور، حل معادله لاپلاس به‌عنوان یک مساله نمونه دینامیک سیالات محاسباتی انتخاب شده‌است و روش‌های برنامه‌نویسی مختلفی برای بهبود عملکرد محاسبات و تسریع حل روی یک اف.پی.جی.ای از نوع زینک-7020 پیاده‌سازی شده و نتایج زمانی حل و دقت آن با حل روی سی.پی.یو مقایسه شده‌است.

2- بستر سخت‌افزاری با قابلیت تغییر در ساختار

سخت‌افزارهایی با قابلیت تغییر در پیکربندی آن‌ها[9] مدارهای مجتمعی هستند که از تعدادی گیت منطقی و اجزای دیگری ساخته شده‌اند. عملکرد گیت‌ها و روابط و کلیدهای بین اجزای داخل این سخت‌افزار قابل تغییر است و می‌توان پیکربندی و ساختار این سخت‌افزار را در حین اجرای برنامه تغییر داد. مفهوم محاسبات قابل پیکربندی در دهه 1960 زمانی که جرالد استرین پیشنهاد ساخت کامپیوتری با یک پردازشگر استاندارد و آرایه‌ای از سخت‌افزار قابل پیکربندی را داد، به‌وجود آمد [11]. پردازشگر اصلی قرار بود که رفتار سخت‌افزار قابل‌پیکربندی را کنترل کند و خود آن سخت‌افزار برای انجام عملی مانند انطباق یک الگو یا پردازش تصویر، برنامه‌ریزی شود. به این ترتیب بعد از انجام عمل موردنظر، پردازشگر می‌توانست سخت‌افزار قابل‌پیکربندی را برای فعالیت دیگری برنامه‌ریزی کند. با چنین ترکیبی از ابزارها، انعطاف‌پذیری نرم‌افزار و سرعت سخت‌افزار ترکیب می‌شدند. همچنین در اواخر دهه 1980 آزمایشی توسط استیو کسلمن [12] مطرح شد تا کامپیوتری طراحی شود که ششصد هزار گیت قابل برنامه‌ریزی درون آن وجود داشته باشد و برای کارکرد از آن‌ها کمک بگیرد که این سیستم در سال 1992 به نام او ثبت شد. ازنظر تجاری نیز، شرکت آلترا[10] که در سال 1983 بنا نهاده شد، یک سال بعد، اولین تراشه منطقی قابل برنامه‌نویسی مجدد را

[1] Shock Waves
[2] Turbulence
[3] Navier-Stokes
[4] Parallel Computing
[5] Graphical Processing Unit (GPU)
[6] Field Programmable Gate Array (FPGA)
[7] Reynolds-Averaged Navier–Stokes equation (RANS)
[8] C Programming
[9] Reconfigurable Hardware
[10] Altera

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

ساخت. نام این دستگاه ای.پی-300[1] بود که یک پنجره کوارتزی روی آن تعبیه شده بود و با تابش نور ماوراء بنفش به آن ساختارش پاک می‌شد و برای برنامه‌نویسی جدید آماده می‌شد.

امروزه چنین سخت‌افزارهایی دارای اجزای مختلفی هستند که ازجمله مهم‌ترین آن‌ها سلول حافظه[2] است. این اجزای حافظه‌دار به‌عنوان جداول مراجعه[3] استفاده می‌شوند تا به‌کمک آن‌ها کلیدهای کنترلی اتصالات بین اجزای مختلف، تنظیم و کنترل شود. برنامه‌ای که عملکرد هر گیت منطقی و حالت کلید را تعیین می‌کند، یک ساختار[4] نامیده می‌شود. فرق اساسی این سخت‌افزارها در مقایسه با میکروپردازشگر[5]های معمولی، توانایی تغییر در مسیر انتقال اطلاعات و کنترل این فرآیند است. علاوه‌بر آن می‌توان در هنگام اجرای برنامه، ساختار سخت‌افزار را با بارگذاری یک طراحی جدید، تغییر داد. یکی از معروف‌ترین انواع سخت‌افزار با قابلیت تغییر در پیکربندی آن با نام اف.پی.جی.ای موجود است. ساختار اف.پی.جی.ای معمولا با استفاده از زبان توصیف سخت‌افزاری[6] مانند وی.اچ.دی.ال[7] و وریلاگ[8] پیکربندی می‌شود. این معماری امکان پردازش و حل مسایل مختلف ازجمله معادلات دیفرانسیل را به‌صورت سخت‌افزاری فراهم می‌کند. سی.پی.یو به‌عنوان شناخته شده‌ترین پردازشگر مورد استفاده، سخت‌افزاری است که قابلیت تغییر در ساختار آن پس از ساختش وجود ندارد.

3- اتصال اف.پی.جی.ای و سی.پی.یو

سخت‌افزار قابل پیکربندی (اف.پی.جی.ای) معمولا در انجام عملیات به‌خصوصی مانند انجام حلقه‌های تکرار، بهینه عمل نمی‌کند و برای اجرای برنامه در بهینه‌ترین حالت ممکن بهتر است در کنار آن یک میکروپردازشگر قرارداده شود تا در مواردی که اف.پی.جی.ای بازدهی خوبی ندارد به‌کمک آن بیاید. به این ترتیب بازدهی کل سیستم به شکل قابل توجهی بالا خواهد رفت. در سیستمی که حاوی اف.پی.جی.ای و میکروپردازشگر است، روش‌های مختلفی برای اتصال آن‌ها به هم وجود دارد. باتوجه به "شکل 1" می‌توان الگوهای زیر را برای برقراری ارتباط بین اف.پی.جی.ای و سی.پی.یو پیشنهاد کرد (در این شکل اف.پی.جی.ای‌ها با رنگ خاکستری مشخص شده‌اند).

الگوی اول: سخت‌افزار قابل‌پیکربندی را می‌توان به تنهایی برای ایجاد واحدهای عملیاتی درون یک پردازشگر میزبانی کرد و به این شکل یک محیط برنامه‌نویسی سنتی به‌وجود می‌آید که درون آن یک سری عملیات با قابلیت تغییر در طول زمان وجود دارد. واحد قابل‌پیکربندی در این‌جا به‌عنوان واحد تابع از پردازشگر عمل می‌کند و در ورودی و خروجی‌های آن رجیسترهایی برای کنترل ورود و خروج داده قرار دارد.

الگوی دوم: ممکن است یک واحد قابل‌پیکربندی به‌عنوان پردازشگر مورد استفاده قرار بگیرد. در کل واحد قابل پیکربندی در این حالت بزرگ‌تر از حالت اول است و قابلیت انجام محاسبات را بدون نظارت پردازشگر اصلی دارد. پردازشگر اصلی واحد قابل‌پیکربندی را مقداردهی می‌کند و اطلاعات لازم برای شروع به کار را به آن می‌دهد و یا نشانی قسمتی از حافظه که این اطلاعات در آن ذخیره شده‌اند را به واحد قابل‌پیکربندی صادر می‌کند.

الگوی سوم: ممکن است واحد قابل برنامه‌ریزی به‌عنوان یک پردازشگر از یک سیستم چندپردازشگری عمل کند. اطلاعات کش[9] پردازشگر اصلی برای واحد قابل برنامه‌نویسی، قابل رویت نیست. به‌همین دلیل ارتباط بین این اجزا (مانند ارتباط برای انتقال اطلاعات پیکربندی، اطلاعات ورودی و نتایج) با سرعت پایین‌تر صورت می‌گیرد. این ساختار از این نظر که واحد قابل-پیکربندی توانایی انجام محاسبات روی مقادیر زیاد اطلاعات را به‌صورت هم‌زمان دارد، مناسب است.

الگوی چهارم: ارتباط سخت‌افزار قابل‌پیکربندی با یک پردازشگر خارجی و مستقل است. در این ساختار انتقال اطلاعات بین این دو جزء به ندرت صورت می‌گیرد (حتی اگر وجود داشته باشد). این مدل مشابه حالتی است که ایستگاه‌های مختلف، پردازش اطلاعات را برای مدت بسیار طولانی انجام می‌دهند، بدون این که انتقال اطلاعات لازم باشد.

هر کدام از این ساختارها مزایا و معایبی دارند. هرچه مکان قرارگیری واحد قابل‌برنامه‌ریزی به پردازشگر نزدیک‌تر باشد، برنامه‌های مختلف می‌توانند به تعداد دفعات بیشتر با آن ارتباط برقرار کنند و این به‌دلیل زمان کوتاه برای انتقال اطلاعات است. در این حالت سخت‌افزار بدون دخالت پردازشگر نمی‌تواند برای مدت طولانی به انجام عملیات بپردازد و معمولا مقدار سخت‌افزار در دسترس نیز محدود است. فاصله ارتباط طولانی‌تر به معنی سطح موازی‌سازی بالاتر است و به‌همین سبب سرعت و تعداد دفعات ارتباط کاهش خواهد یافت. در کاربردهایی که به دفعات ارتباط زیاد و با سرعت بالا احتیاج است، چنین ساختاری احتمالا موجب کاهش زمان اجرای برنامه نخواهد شد. در پژوهش حاضر اف.پی.جی.ای و سی.پی.یو دو واحد جدا از هم هستند که با ارتباطات بسیار سریع و بدون واسطه به‌هم متصل شده‌اند، یعنی الگوی دوم برای اتصال دو سخت‌افزار استفاده شده است.

4- ارتباطات داخلی اف.پی.جی.ای

نحوه سیم‌کشی و اتصالات بین بلوک‌های منطقی درون خود سخت‌افزار قابل‌پیکربندی نیز اهمیت زیادی دارد. ساختار این اتصالات به تراکم و تعداد بلوک‌های منطقی درون سخت‌افزار بستگی دارد. با توجه به "شکل 2"، دو روش اصلی برای تعیین ساختار سیم‌کشی‌های محلی و کلی در یک سخت‌افزار قابل‌پیکربندی وجود دارد. یکی از این روش‌ها مربوط‌به استفاده از اتصالات کوتاه[10] می‌باشد که در آن، یک سری سیم کوتاه برای انتقال اطلاعات محلی استفاده می‌شوند. می‌توان این اتصالات را توسط جعبه‌های سوییچ به‌یکدیگر متصل کرد تا به‌عنوان سیم‌های طولانی‌تر مورد استفاده قرارگیرند. سیم‌های طولانی‌تر را می‌توان برای اتصال محلی و کلی استفاده کرد. اتصال سلسله مراتبی[11] به این شکل انجام می‌شود که بعد از اتصال

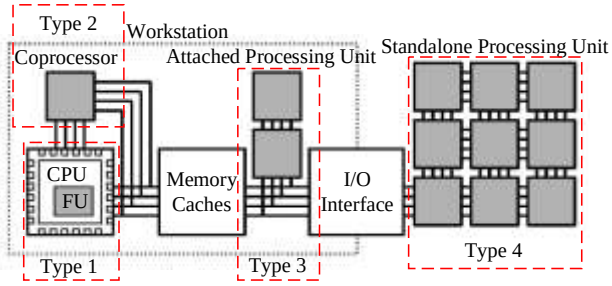

Fig. 1 FPGA and CPU connection configurations [13]

شکل 1 پیکربندی‌های مختلف برای اتصال اف.پی.جی.ای و سی.پی.یو [13]

[1] EP300
[2] Memory Cell
[3] Lookup Table
[4] Configuration
[5] Microprocessor
[6] Hardware Description Language
[7] VHDL
[8] VERILOG
[9] Cache
[10] Segmented Routing
[11] Hierarchical Routing

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

اجزای داخلی گروه‌های کوچک توسط سیم‌های کوتاه، برای اتصال این گروه‌ها از سیم‌های طولانی‌تر در مرز این گروه‌ها استفاده می‌شود. معمولا اتصال سلسله مراتبی را برای حالتی که قرار است بیشتر ارتباطات بین اجزای داخلی گروه‌ها و قسمت کمتری از ارتباطات بین گروه‌ها باشد، بهینه‌سازی می‌کنند.

5- سخت‌افزار مورد استفاده و روش پیکربندی

سخت‌افزار اف.پی.جی.ای مورد استفاده در این پژوهش از نوع اف.پی.جی.ای کنار سی.پی.یو است، به این معنی که داخل تراشه اصلی پردازشی هر دو سخت‌افزار قابل‌برنامه‌ریزی (اف.پی.جی.ای) و پردازشگر (سی.پی.یو) وجود دارند. تراشه اصلی مورد استفاده در این پژوهش مدل زینک-7020[1] از خانواده زینک-7000 ساخت شرکت زایلینکس[2] است. تراشه‌های این خانواده از نوع 28 نانومتر مقیاس‌پذیر سری هفتم اف.پی.جی.ای‌های ساخت شرکت زایلینکس هستند. هر تراشه برای کاربردهای خاص طراحی و بهینه‌سازی شده است. تراشه‌های سری زد-7010، 7015 و 7020 براساس تکنولوژی منطق قابل‌پیکربندی آرتیکس[3] از شرکت زایلینکس ساخته شده‌اند و در کاربردهای با حجم محاسبات بالا و مصرف توان کم، بهینه هستند. تراشه‌های سری زد-7030، 7035، 7045 و 7100 براساس تکنولوژی منطق قابل‌پیکربندی کینتکس[4] از شرکت زایلینکس ساخته شده‌اند و در کاربردهای با نیاز عملکردی بالاتر و گذردهی خروجی و ورودی زیاد مناسب هستند. تراشه موردنظر به‌تنهایی کاربرد خاصی نخواهد داشت، به این دلیل که ارتباطات خارجی آن برای ریختن اطلاعات پیکربندی و استخراج نتایج، هنوز قابل استفاده نیستند. بهترین راه حل ممکن برای انجام این کار، نصب تراشه بر روی یک بورد که دارای ارتباطات خارجی استاندارد می‌باشد، است. بورد مورد استفاده در این پژوهش با نام زد-ترن[5] شناخته شده و ساخت شرکت ام.وای.آی.آر[6] است. بر روی این بورد ارتباطات خارجی یو.اس.بی[7]، شبکه، حافظه فلش و غیره تعبیه شده است. تصویری از این بورد و اجزای آن در "شکل 3" نشان داده شده است.

سی.پی.یو مورد استفاده برای مقایسه نتایج، مدل کور.آی7-کیو740[8]، ساخت شرکت اینتل با بیشینه فرکانس عملکرد 1.73 گیگاهرتز است. این سی.پی.یو دارای چهار هسته پردازشی فیزیکی است. هر کدام از این هسته‌های فیزیکی می‌توانند به‌صورت هم‌زمان دو عملیات جداگانه را انجام دهند، پس در کل هشت هسته پردازشی مجازی دارد. بیشینه توان محاسبات این پردازشگر برابر 13.84 گیگافلاپ (یعنی 13.84 میلیارد عملیات اعشاری در هر ثانیه!) است. البته باید توجه داشت که این مقدار اسمی بیشینه توان محاسباتی سی.پی.یو است و در عمل هیچ وقت امکان دست‌یابی به این عدد امکان‌پذیر نیست.

هرچند عملکرد اف.پی.جی.ای‌ها برای کاهش زمان اجرای برنامه‌ها ثابت شده است ولی انجام محاسبات با سخت‌افزارهایی که قابلیت تغییر در ساختار آن‌ها بعد از ساختشان وجود دارد (مانند اف.پی.جی.ای) نیازمند افرادی با تخصص بالا است که با استفاده از زبان توصیف سخت‌افزار، اف.پی.جی.ای را برنامه‌ریزی کنند. کدنویسی با زبان توصیف سخت‌افزار حتی برای افراد متخصص، نیازمند زمان زیاد است و این هدف پژوهش که کاهش زمان حل

[1] Zynq-7020
[2] Xilinx
[3] Artix
[4] Kintex
[5] Z-turn
[6] MYiR
[7] USB
[8] Core i7 Q740

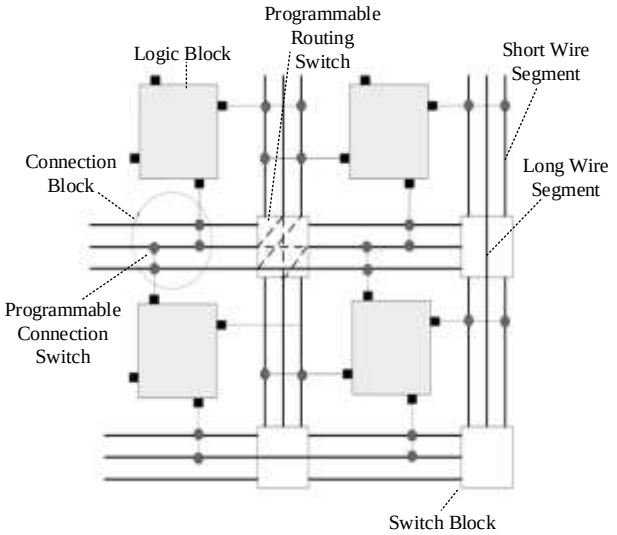

Fig. 2 Interconnection architecture inside an FPGA [14]

شکل 2 شماتیک سیم‌کشی درون یک اف.پی.جی.ای [14]

Fig. 3 z-turn board by MYiR

شکل 3 بورد زد-ترن ساخت شرکت ام.وای.آی.آر

بود را زیر سوال می‌برد. به این ترتیب محیط طراحی و توسعه مناسبی برای طراحی نرم‌افزار قابل اجرا روی سخت‌افزار لازم است. چنین محیطی می‌تواند یک نرم‌افزار کمکی برای فرآیند طراحی دستی مدار و یا یک سیستم کامل و اتوماتیک طراحی مدار باشد. در هر صورت طراحی نرم‌افزار برای چنین سخت‌افزاری نیاز به دانش بالا از سخت‌افزار و همچنین صرف زمان زیادی دارد. خوشبختانه این مشکل توسط زبان‌های توصیف سخت‌افزار سطح بالا (اچ.ال.ال[9])، که عموما براساس زبان برنامه‌نویسی سی ساخته می‌شوند، کم‌رنگ‌تر می‌شود. اچ.ال.ال‌ها زمان توسعه نرم‌افزار را کاهش می‌دهند اما برنامه نوشته شده با این زبان‌ها برای اجرای بهتر روی سخت‌افزار نیازمند بهینه‌سازی دستی می‌باشد. خانواده جدید اف.پی.جی.ای‌ها استفاده از اعداد اعشاری و انجام اعمال ریاضی روی آن‌ها را ساده‌تر می‌کنند و به این ترتیب برای انجام اعمال ریاضی مناسب هستند.

مراحل مختلفی در فرآیند طراحی یک مدار وجود دارد. مشخص کردن مدار[10]، فرآیند تعریف توابعی است که قرار است روی سخت‌افزار قابل‌پیکربندی قرار بگیرند. این کار می‌تواند با نوشتن یک برنامه به زبان سی که تابعیت الگوریتم را توضیح می‌دهد، انجام شود. از طرف دیگر ممکن است

[9] High Level Language (HLL)
[10] Circuit Specification

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

همین کار با مشخص کردن ورودی و خروجی‌ها و عملیات یک‌یک بلوک‌های منطقی درون سخت‌افزار انجام شود که دشوارتر از حالت قبل است. ساخت مدار با استفاده از واحدهای از پیش ساخته‌شده مانند جمع‌کننده‌ها و ضرب-کننده‌ها نیز روشی است که از نظر دشوار بودن، بین دو روش بالا قرارمی‌گیرد.

فرآیند اصلی استفاده شده برای طراحی سخت‌افزار در این پژوهش مخصوص تراشه‌های زینک-7000 است که توسط شرکت زایلینکس پیشنهاد شده است. سه نرم‌افزار اصلی در طول این فرآیند استفاده می‌شوند که هر سه توسط مهندسین شرکت زایلینکس و مخصوص محصولات همین شرکت، توسعه داده شده‌اند. این سه نرم‌افزار شامل ویوادو اچ.ال.اس[1]، ویوادو[2] و زایلینکس اس.دی.کی[3] هستند. فرآیند طراحی در این‌جا با استفاده از زبان توصیف سخت‌افزار سطح بالا انجام می‌شود. طراحی سخت‌افزار با بهره‌گیری از یک سری بلوک‌های از پیش ساخته شده با نام آی.پی[4] و برقراری اتصالات بین آن‌ها به‌صورت گرافیکی در نرم‌افزار ویوادو انجام می‌شود. نمونه ساختار طراحی شده برای حل مساله لاپلاس در "شکل 4" نمایش داده شده است.

6- زمان‌سنج با دقت بالا

عملیات اندازه‌گیری عملکرد یک سیستم محاسباتی شامل اندازه‌گیری زمان پاسخ‌دهی، توان عملیاتی، دیرکرد و همچنین زمان حل الگوریتم می‌شود. هرکدام از این عملیات شامل اندازه‌گیری اعمال انجام شده در فاصله زمانی شروع تا پایان آن‌ها می‌شوند. زمان‌سنج‌ها در یک سیستم محاسباتی دو نوع هستند؛ ساعت‌های مطلق و ساعت‌های اختلافی. ساعت‌های مطلق به زمان‌سنج‌هایی گفته می‌شود که زمان دقیق در روز را نمایش می‌دهند و وابسته به زمان‌سنج‌های خارجی هستند. ساعت‌های اختلافی، بازه زمانی را اندازه‌گیری می‌کنند و عموما براساس منابع خارجی کار نمی‌کنند. کیو.پی.سی[5] یک نمونه ساعت اختلافی در سیستم عامل ویندوز[6] است که با هیچ زمان‌سنج خارجی هم‌زمان‌سازی نمی‌شود و برای اندازه‌گیری بازه‌های زمانی استفاده می‌شود. کیو.پی.سی در اصل یک شمارنده سخت‌افزاری است که از سه بخش اصلی تشکیل می‌شود؛ یک منبع ارتعاشی، یک شمارنده تعداد ارتعاش‌ها و وسیله‌ای برای بازیابی مقادیر شمارنده. خصوصیات هرکدام از این سه بخش، تراکم، دقت، صحت و پایداری کیو.پی.سی را تعیین می‌کنند.

در پژوهش حاضر برای محاسبه زمان حل مساله روی سی.پی.یو از زمان‌سنج دقیق کیو.پی.سی استفاده می‌شود. دقت این زمان‌سنج تا یک میکروثانیه است. برای محاسبه زمان انجام حل روی سخت‌افزار قابل پیکربندی از زمان‌سنج دقیق‌تری که روی اف.پی.جی.ای مورد استفاده در این پژوهش تعبیه شده است، استفاده می‌شود. این زمان‌سنج به‌صورت یک آی.پی داخل بسته نرم‌افزاری ویوادو قابل استفاده است. این آی.پی با نام اکسی.تایمر[7] شناخته می‌شود و ارتباطات آن با سایر آی.پی‌ها در "شکل 4" مشخص شده است. دقت محاسبه زمان این آی.پی تا دو نانو ثانیه است.

7- معادله لاپلاس

باتوجه به کاربرد معادله لاپلاس در جریان پتانسیل و انتقال حرارت پایای دوبعدی، این معادله به‌عنوان نمونه‌ای از مسایل دینامیک سیالات محاسباتی برای قابلیت‌سنجی سخت‌افزار اف.پی.جی.ای انتخاب شده‌است. روش‌های حل عددی متفاوتی برای این معادله وجود دارد و از آن‌جا که حل تحلیلی نیز برای آن وجود دارد، دقت نتایج را می‌توان اعتبارسنجی کرد. معادله لاپلاس به‌صورت رابطه (1) نوشته می‌شود.

$$\nabla^2\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0 \tag{1}$$

شرایط مرزی "شکل 5" را برای این مساله در نظر بگیرید. طبق [15] با استفاده از روش جداسازی متغیرها، حل تحلیلی و دقیق این مساله به‌صورت رابطه (2) نوشته می‌شود.

$$\psi(x,y) = \frac{4}{\pi^2}\sum_{n=0}^{\infty}(-1)^n \frac{\sin\big((2n+1)\pi x\big)\sinh\big((2n+1)\pi(1-y)\big)}{(2n+1)^2\sinh\big((2n+1)\pi\big)} \tag{2}$$

برای انجام حل عددی این مساله از روش نقطه‌ای گوس-سایدل[8] استفاده می‌شود. طبق [16] معادله (1) بعد از گسسته‌سازی با این روش به رابطه (3) منجر می‌شود که برای حل عددی مساله لاپلاس استفاده خواهد شد.

$$\psi_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)}\Big(\psi_{i-1,j}^{k+1} + \psi_{i+1,j}^{k} + \beta^2\big(\psi_{i,j-1}^{k+1} + \psi_{i,j+1}^{k}\big)\Big) \tag{3}$$

در رابطه (3)، k گام زمانی حل، پسوندهای i و j نماینده نقطه‌ای از شبکه عددی و β نشان‌دهنده نسبت اندازه شبکه در راستای x به راستای y است.

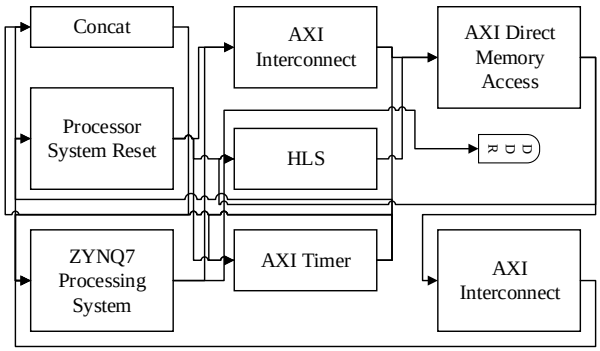

Fig. 4 Block design to solve Laplace problem

شکل 4 طرح بلوکی برای حل مساله لاپلاس

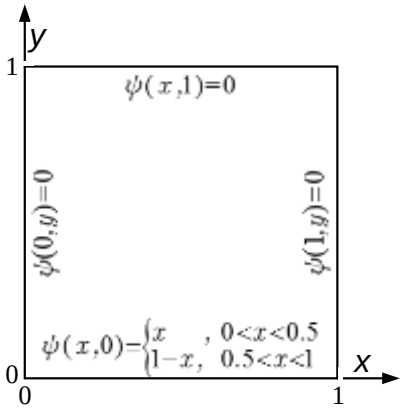

Fig. 5 Boundary conditions for Laplace problem

شکل 5 شرایط مرزی معاله لاپلاس

[1] Vivado HLS
[2] Vivado
[3] Xilinx SDK
[4] Intellectual Property
[5] Query Performance counter (QPC)
[6] Windows
[7] AXI Timer
[8] Point Gauss-Seidel

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

8- نتایج حل عددی

بعد از طی مراحل لازم برای طراحی سخت‌افزار و اعمال آن روی اف.پی.جی.ای و سی.پی.یو، حل عددی روی دو شبکه محاسباتی 51 در 51 و 101 در 101 انجام شده است. حل با استفاده از روش نقطه‌ای گوس- سایدل انجام می‌شود که یک روش حل صریح است. به این معنی که در هر تکرار حل و انجام محاسبات معادله (3)، اطلاعات نقاط میدان محاسباتی به‌صورت جداگانه و تک‌تک محاسبه می‌شوند. اختلاف اصلی روش حل نقطه‌ای گوس-سایدل و روش حل صریح ژاکوبی در اطلاعات استفاده شده برای انجام محاسبات یک نقطه از شبکه‌ی محاسباتی است. برای انجام محاسبات یک نقطه مجهول در شبکه محاسباتی با استفاده از روش صریح ژاکوبی باید اطلاعات 4 نقطه اطراف آن در اختیار باشد که تمامی این اطلاعات از گام حل قبلی استفاده می‌شوند. اما در روش نقطه‌ای گوس-سایدل همان‌گونه که از معادله (3) مشخص است، محاسبات هر نقطه مجهول در شبکه محاسباتی با استفاده از اطلاعات دو نقطه از گام حل قبلی و اطلاعات دو نقطه از گام حل کنونی انجام می‌شود. در پژوهش حاضر، محاسبات موردنظر به‌صورت ردیف به ردیف و با شروع از ضلع پایینی میدان محاسباتی انجام می‌شود. نمونه نقاط استفاده شده در معادله (3)، در "شکل 6" مشخص شده است. کانتور ψ در حل معادله (3) با دقت مضاعف و اندازه شبکه $\pi/100$، بعد از 10000 گام حل در "شکل 7" آمده است. در "شکل 8" نیز کانتور ψ برای حل تحلیلی این مساله نمایش داده شده است. در جدول 1 نرم خطای بین داده‌های حل عددی و حل تحلیلی (از معادله 2) آورده شده است. مقادیر کم نرم خطای موجود در این جدول نشان‌دهنده صحت حل عددی است. معیار همگرایی این مساله، نرم خطای بین داده‌های حل تحلیلی و حل عددی است. برای بررسی دقیق‌تر اختلاف دو حل عددی و تحلیلی مساله لاپلاس، مقادیر تابع جریان برروی دو خط $y = 0.2$ و $y = 0.5$ از هر دو حل عددی و تحلیلی در "شکل 9" مقایسه شده‌اند. همان‌گونه که از این شکل مشخص است دو حل با هم انطباق بسیار خوبی داشته و این نشان‌دهنده صحت حل عددی است.

در "شکل 10" نمودار زمان حل مساله (محور عمودی سمت چپ) و میزان تسریع محاسبات (محور عمودی سمت راست) با استفاده از سی.پی.یو و اف.پی.جی.ای در دقت‌های مختلف برحسب تعداد نقاط شبکه محاسباتی آمده است. مشاهده می‌شود که در یکی از حالت‌ها تا 20 برابر نیز افزایش سرعت به‌دست آمده است. از آن‌جایی که برای به‌دست آوردن جواب عددی مناسب و دقیق، لازم است که حل به تعداد گام‌های زیادی پیش برود (بیشتر از 1000 گام حل)، حل کامل این مساله روی سخت‌افزار قابل‌پیکربندی، به‌علت وجود سه حلقه تکرار منطقی تودرتو در برنامه کامپیوتری آن، بهینه نخواهد بود. بنابراین بهتر است فقط افزایش سرعت حل یک گام حل روی اف.پی.جی.ای بررسی شود. البته درصورتی که سخت‌افزار قوی‌تری از نظر قدرت و فرکانس پردازش در اختیار باشد، می‌توان مساله را به‌صورت کامل روی آن حل کرد.

نمودار زمان حل مساله لاپلاس روی اف.پی.جی.ای در مقابل تعداد نقاط شبکه عددی در "شکل 11" آمده است. همان‌گونه که از این شکل مشخص است، زمان حل روی سخت‌افزار قابل‌پیکربندی با تعداد نقاط شبکه رابطه مستقیم و خطی دارد. شیب این خط برابر یک بوده و به این ترتیب رابطه خطی دو پارامتر ثابت می‌شود.

در پژوهش‌های قبلی که در زمینه تسریع محاسبات عددی با استفاده از اف.پی.جی.ای انجام شده‌اند، یکی از مسائل مهم مورد بررسی حل مساله لاپلاس بوده است. در این بخش نتایج زمانی به‌دست آمده از حل مساله لاپلاس با نتایج سایر مراجع مقایسه شده است. در [4] حل یک نقطه از شبکه عددی این مساله روی هر دو سخت‌افزار سی.پی.یو و اف.پی.جی.ای انجام‌شده است. سخت‌افزارهای استفاده شده توسط [4] با سخت‌افزارهای مورد استفاده در پژوهش حاضر متفاوت هستند. به‌همین دلیل لازم است پارامتری عمومی برای مقایسه نتایج انتخاب شود. متغیری که می‌تواند توان محاسباتی را صرف‌نظر از سخت‌افزار استفاده شده مشخص کند، تعداد دوره‌های فرکانس زمانی (کلاک[1]) لازم برای حل مساله است. تعداد کلاک لازم برای تکمیل فرآیند حل، یک پارامتر بی‌بعد است و از این نظر نیز نسبت‌به سایر متغیرهای بابعد برتری دارد. در جدول 2 نتایج زمانی و تعداد کلاک در حل یک نقطه از شبکه عددی مساله لاپلاس آمده است. تمامی آزمایش‌ها با دقت اعشاری معمولی انجام شده است. پژوهش دیگری نیز در این زمینه در [9] انجام شده است که از نتایج آن نیز برای مقایسه ارایه شده است.

9- نتیجه‌گیری

در پژوهش حاضر راهکار جدیدی برای انجام محاسبات عددی با استفاده از یک سیستم الکترونیکی پیشنهاد شده است. حل عددی مساله لاپلاس به‌عنوان محاسبات نمونه روی سخت‌افزار قابل پیکربندی پیشنهادی این

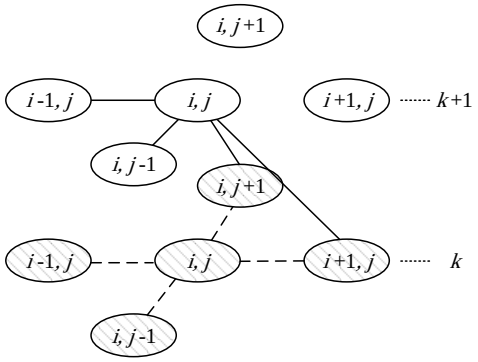

Fig. 6 Stencil of the point Gauss-Seidel method

شکل 6 الگوی روش نقطه‌ای گوس-سایدل

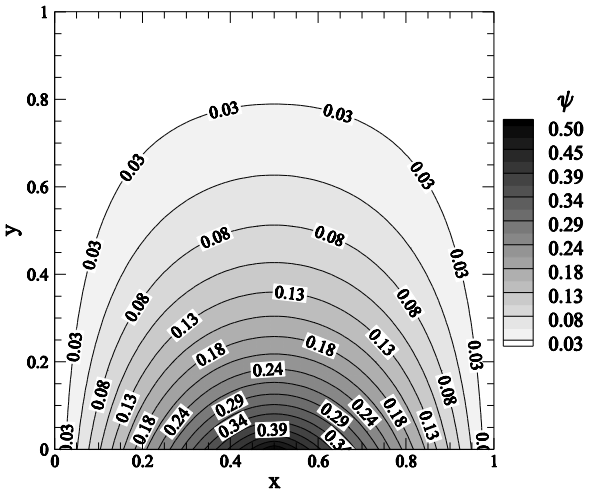

Fig. 7 ψ contour in Laplace numerical solution

شکل 7 کانتور ψ در حل عددی مساله لاپلاس

[1] Clock

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

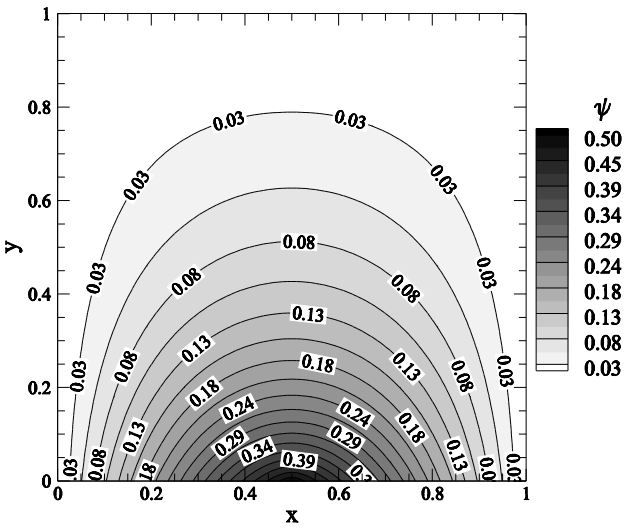

Fig. 8 ψ contour in Laplace exact solution

شکل 8 کانتور ψ در حل تحلیلی مساله لاپلاس

جدول 1 نرم خطای حل تحلیلی و عددی مساله لاپلاس

Table 1 Exact and numerical Laplace solution error norm

اندازه شبکه	گام حل	نرم اول خطا	نرم دوم خطا	نرم بی‌نهایت خطا
$\frac{\pi}{50}$	1000	0.08705	0.09807	0.15752
	5000	0.00502	0.00731	0.03524
	10000	0.00502	0.00731	0.03524
$\frac{\pi}{100}$	1000	0.38357	0.43655	0.77434
	5000	0.03569	0.06008	0.09373
	10000	0.00505	0.00625	0.01801

پژوهش انجام شده است و نتایج حاصل از آن از نظر دقت و سرعت انجام محاسبات با نتایج حاصل از یک سیستم کامپیوتری سنتی مقایسه شده است. هدف اصلی از پژوهش حاضر تلاش برای تسریع و کاهش زمان محاسبات عددی به‌صورت سخت‌افزاری است. حل مساله لاپلاس با استفاده از اف.پی.جی.ای تا 20 برابر سریع‌تر از حل روی یک سی.پی.یو 73.1 گیگاهرتزی انجام شده است. به جز تغییر سخت‌افزار محاسباتی مورد استفاده از

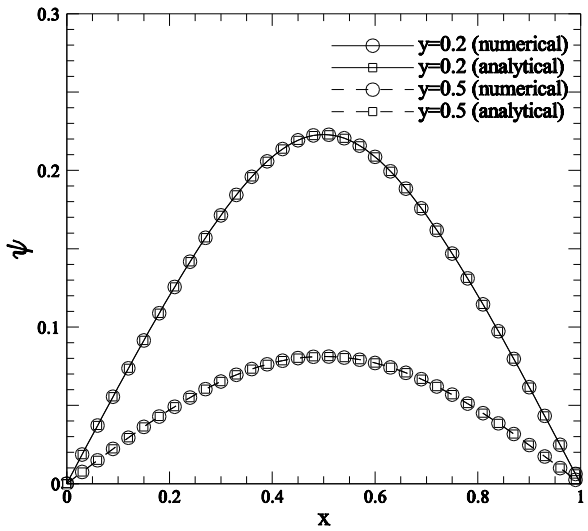

Fig. 9 Analytical vs. numerical solution of ψ on two horizontal lines

شکل 9 مقایسه مقدار ψ از حل‌های عددی و تحلیلی بر روی دو خط افقی

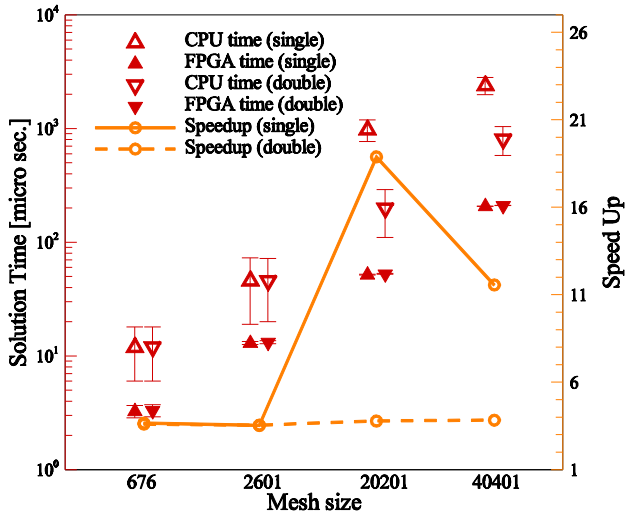

Fig. 10 Time duration and speedup for Laplace solution

شکل 10 نمودار زمان و تسریع حل مساله لاپلاس

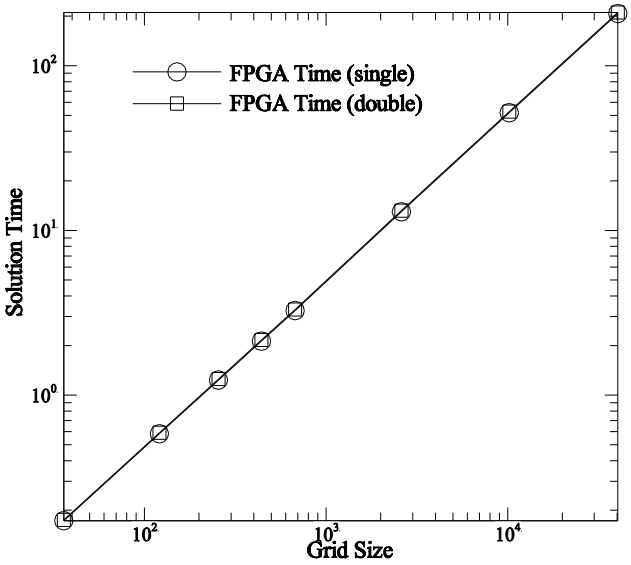

Fig. 11 Solution time versus grid size

شکل 11 زمان حل در مقابل تعداد نقاط شبکه

جدول 2 نتایج زمانی حل یک نقطه از شبکه معادله لاپلاس

Table 2 Single node solution time duration in Laplace problem

پژوهش	سخت‌افزار	فرکانس پردازش [MHz]	زمان حل [ns]	تعداد کلاک
مرجع [4]	CPU	3000	43	130
	FPGA	822.37	1.22	1
مرجع [9]	CPU	2930	89.11	261.1
	FPGA	133	7.5	1
پژوهش حاضر	CPU	1730	77.18	472.32
	FPGA	250	4.83	1.2071

سی.پی.یو به اف.پی.جی.ای، روش‌های الگوریتمی و بهینه‌سازی‌های نرم‌افزاری زیادی برای رسیدن به این هدف استفاده شدند. یکی از دلایل عدم افزایش سرعت محاسبات به اندازه زیاد، عدم کاهش دقت حل با استفاده از اف.پی.جی.ای نسبت به سی.پی.یو بوده است. درصورتی که دقت‌های پایین‌تر برای اعداد در حل روی اف.پی.جی.ای انتخاب شود، باید به این نکته توجه

[Downloaded from mme.modares.ac.ir on 2024-04-26]
[DOR: 20.1001.1.10275940.1396.17.1.13.3]

کرد که میزان اختلاف نتایج نهایی به‌دست آمده از اف.پی.جی.ای با حل سی.پی.یو به اندازه قابل قبولی باشد. در این حالت زمان حل به‌علت نیاز کمتر به محاسبات منطقی کاهش پیدا خواهد کرد.

10- مراجع

[1] J. D. Owens, M. Houston, D. Luebke, S. Green, J. Stone, J. Phillips, GPU computing, *Proceedings of the IEEE*, Vol. 96, No. 5, pp. 879-899, 2008.

[2] M. S. AbuTalip, T. Akamine, M. Hatto, H. Amano, Y. Osana, N. Fujita, Adaptive flux calculation scheme in advection term computation using partial reconfiguration, *International Journal of Networking and Computing*, Vol. 3, No. 2, pp. 289-306, 2013.

[3] H. Thomas, A flow solver for a reconfigurable FPGA-based hypercomputer, *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, January 10-13, 2005.

[4] R. Nunez, J. Gonzalez, and J. Camberos, Large-scale numerical solution of partial differential equations with reconfigurable computing, *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida, USA, June 25-28, 2007.

[5] K. Sano, T. Iizuka, S. Yamamoto, Systolic architecture for computational fluid dynamics on FPGAs, *15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM '07)*, Napa, California, USA, April 23-25, 2007.

[6] E. Andrés, C. Carreras, G. Caffarena, M. del C. Molina, O. Nieto-Taladriz, F. Palacios, A methodology for CFD acceleration through reconfigurable hardware, *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, January 7-10, 2008.

[7] E. Andres, M. Widhalm, A. Caloto, Achieving high speed CFD simulations: optimization, parallelization, and FPGA acceleration for the unstructured DLR TAU code, *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, USA, January 5-8, 2009.

[8] D. Sanchez-Roman, G. Sutter, S. Lopez-Buedo, I. Gonzalez, F. J. Gomez-Arribas, J. Aracil, An euler solver accelerator in FPGA for computational fluid dynamics applications, *VII Southern Conference on Programmable Logic (SPL)*, Cordoba, Argentina, April 13-15, 2011.

[9] K. Sano, Y. Hatsuda, S. Yamamoto, Performance evaluation of FPGA-based custom accelerators for iterative linear-equation solvers, *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, USA, June 27-30, 2011.

[10] I. Liu, E. A. Lee, M. Viele, G. Wang, H. Andrade, A heterogeneous architecture for evaluating real-time one-dimensional computational fluid dynamics on FPGAs, *IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 125–132, 2012.

[11] G. Estrin, Reconfigurable computer origins: the UCLA fixed-plus-variable (F+V) structure computer, *IEEE Annals of the History of Computing*, Vol. 24, No. 4, pp. 3–9, 2002.

[12] *History of FPGAs at the Wayback Machine*, Accessed on 12 April 2007; https://archive.org/web/.

[13] M. B. Gokhale and P. S. Graham, *Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays*, Springer Science & Business Media, First Edition, pp. 100-231, Springer, New York, USA, 2005.

[14] P. Singh, S. K. Vishvakarma, Device/Circuit/Architectural techniques for ultra-low power FPGA design, *Microelectronics and Solid State Electronics*, Vol. 2, No. A, pp. 1–15, 2013.

[15] Peter J. Olver, *Introduction to Partial Differential Equations*, Princeton university press, First Edition, pp. 236-256, Princeton university press, New Jersey, USA, 2014.

[16] K. A. Hoffmann, S. T. Chiang, *Computational Fluid Dynamics*, Engineering Education System, Fourth Edition, pp. 301-352, the University of Michigan, Michigan, USA, 2000.

[Downloaded from mme.modares.ac.ir on 2024-04-26]

[DOR: 20.1001.1.10275940.1396.17.1.13.3]