



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

عباس ابراهیمی^{1*}، محمد زندسالیمی²

1- استادیار، مهندسی هوافضا، دانشگاه صنعتی شریف، تهران

2- دکتری تخصصی، مهندسی هوافضا، دانشگاه صنعتی شریف، تهران

ebrahimi_a@sharif.ir، 11155-1639 * تهران، صندوق پستی

چکیده

امروزه کاهش زمان محاسبات عددی از چالش‌های اصلی در شبیه‌سازی مسائل واقعی سیالاتی به شمار می‌رود. حل عددی معادلات دیفرانسیل در بستر سخت‌افزارهایی مانند سی‌پی‌بی و جی‌پی‌بی، براساس پیاده‌سازی نرم‌افزاری هستند. در این مقاله قابلیت سخت‌افزاری اف‌پی‌جی‌ای به عنوان راهکاری جدید برای حل عددی معادلات دیفرانسیل در دینامیک سیالات محاسباتی ارزیابی شده است. سخت‌افزاری اف‌پی‌جی‌ای می‌تواند حل معادلات را از حالت سری خارج کرده و موازی‌سازی آن‌ها را براساس پیاده‌سازی سخت‌افزاری انجام دهد. اف‌پی‌جی‌ای 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 به عنوان نمونه‌ای از مسائل سیالاتی روی یک اف‌پی‌جی‌ای خاص پیاده‌سازی شده و نتایج حل عددی آن با نتایج گرفته شده از برنامه‌کامپیوتری روی سی‌پی‌بی 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 سی‌پی‌بی است به گونه‌ای 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.

اطلاعات مقاله

مقاله پژوهشی کامل

دریافت: 05 شهرور 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

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.

مراحل طراحی و بهینه‌سازی بسیاری از مسائل سیالاتی صنعتی و مهندسی

هوافضا دارد. همچنین از سی‌اف‌دی برای مطالعه مفاهیم فیزیکی جریان و

دینامیک سیالات محاسباتی (سی‌اف‌دی¹) نقشی اساسی در بخشی از

۱- مقدمه

¹ Computational Fluid Dynamics (CFD)

Please cite this article using:

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)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

مگاهتر به سرعت نهایی حل حدود 7 برابر سرعت حل در یک سی‌پی‌یو 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 روشهای کاربردی برای حل معادلات شبه-یکبعدی اوپلر در مساله لوله موج ضربهای ارائه کردند. همچنین در [7] فرآیندهای مختلف حل معادلات رانس⁷ از نظر زمان مورد نیاز بررسی شده و سپس با بهینه‌سازی کد، موازی کردن محاسبات و استفاده از ساخت‌افزارهای جدید مانند حی.پی.یو و اف.پی.جی.ای، حل با سرعت بالاتری انجام شد.

در [8] معادلات اوپلر به روش حجم محدود با استفاده از یک زبان سطح بالا (زبان برنامه‌نویسی سی⁸) 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 سی‌اف.دی خطی که به شکل تکراری حل می‌شد، بررسی شده است سپس با افزایش تعداد اف.پی.جی.ای‌هایی که هم‌زمان مورد استفاده قرار گرفته‌اند، مشاهده شد که سرعت حل به صورت خطی افزایش می‌یابد. همچنین لیو و همکاران [10] در زمینه 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 مسائل سیالاتی است. به این منظور، حل معادله لابلاس به عنوان یک مساله نمونه دینامیک سیالات محاسباتی انتخاب شده است و روش‌های برنامه‌نویسی مختلفی برای بهبود عملکرد محاسبات و تسريع حل روی یک اف.پی.جی.ای از نوع زینک-2020⁹ پیاده‌سازی شده و نتایج زمانی حل و دقت آن با حل روی سی‌پی.یو مقایسه شده است.

2- بستر سخت‌افزاری با قابلیت تغییر در ساختار

سخت‌افزارهایی با قابلیت تغییر در پیکربندی آن‌ها¹⁰ 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، به وجود آمد [11]. پردازشگر اصلی قرار بود که رفتار سخت‌افزار قابل پیکربندی 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 قابل پیکربندی را برای فعالیت دیگری برنامه‌ریزی کند. با چنین ترکیبی از ابزارها، انعطاف‌پذیری نرم‌افزار و سرعت سخت‌افزار ترکیب می‌شوند. همچنین در اوخر دهه 1980 آزمایشی توسط استیو کسلمن [12] مطرح شد تا کامپیوترا 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. از نظر تجاری نیز، شرکت آلترا¹¹ که در سال 1983 بنایاده شد، یک سال بعد، اولین تراشه منطقی قابل برنامه‌نویسی مجدد را

⁷ Reynolds-Averaged Navier–Stokes equation (RANS)

⁸ C Programming

⁹ Reconfigurable Hardware

¹⁰ Altera

شیوه‌سازی آن‌ها مانند لایه‌مرزی، امواج ضربهای¹، آشفتگی² جریان و غیره استفاده می‌شود. با این حال هزینه‌های محاسباتی بالا باعث شده که نتوان از این دانش به عنوان ابزاری که کل فاز طراحی بر عهده 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 (نور-استوکس³) و انرژی مشتق می‌شوند و به جز در موارد خاص و ساده شده دارای حل تحلیلی نیستند. در بسیاری از مسائل واقعی، حل عددی این معادلات به شبکه‌های محاسباتی ریز و یا تعداد تکرارهای حل زیاد دارد، بنابراین حتی با وجود کامپیوترهای بسیار قوی نیز زمان موردنیاز برای حل افزایش پیدا می‌کند. به همین دلیل محققین علوم مهندسی تلاش زیادی برای تسريع حل الگوریتم‌های حل معادلات دیفرانسیل حاکم بر جریان انجام داده‌اند. نتایج پژوهش‌های انجام شده در این زمینه به محاسبات موازی⁴، استفاده از تراشه‌های چند هسته‌ای، محاسبات با استفاده از تراشه‌های ویژه مانند جی.پی.یو⁵ [1] و غیره منجر شده است.

سی‌پی.یو و جی.پی.یو به عنوان شناخته شده‌ترین 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.ای⁶ قابل پیکربندی هستند و می‌توان ساختار این سخت‌افزارها را در حین اجرای برنامه تغییر داد. در بررسی‌های اخیر [2]، به نظر می‌رسد استفاده از تراشه اف.پی.جی.ای که براساس پیاده‌سازی سخت‌افزاری معادلات است برای تسريع 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.ای، می‌توانند به عنوان پردازشگرهایی با قابلیت انعطاف‌پالا و به تنهایی استفاده شوند. در گذشته بیشتر تحقیقاتی که در زمینه محاسبات ریاضی با استفاده از اف.پی.جی.ای انجام شده، مربوط به کاربردهایی مانند ذخیره ماتریس‌ها و ضرب آن‌ها، تبدیل فوریه عددی، پردازش سیگنال و کاربردهای دیجیتال دیگر بوده است.

در زمینه سی‌اف.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 نویز در [4] شرح داده شده است.

سانو و همکاران [5] با برنامه‌ریزی یک سخت‌افزار اف.پی.جی.ای به صورت واحدهایی که قابلیت انجام ضرب و جمع را دارند، توانستند مساله دو بعدی حفره را در یک شبکه 24 در 24 حل کنند و در فرکانس کار کرد 60

¹ Shock Waves

² Turbulence

³ Navier-Stokes

⁴ Parallel Computing

⁵ Graphical Processing Unit (GPU)

⁶ Field Programmable Gate Array (FPGA)

یک سیستم چندپردازشگری عمل کند. اطلاعات کش^۹ پردازشگر اصلی برای واحد قابل برنامه‌نویسی، قابل رویت نیست. بهمین دلیل ارتباط بین این اجزا (مانند ارتباط برای انتقال اطلاعات پیکربندی، اطلاعات ورودی و نتایج) با سرعت پایین تر صورت می‌گیرد. این ساختار از این نظر که واحد قابل پیکربندی توانایی انجام محاسبات روی مقادیر زیاد اطلاعات را به صورت همزمان دارد، مناسب است.

الگوی چهارم: ارتباط سخت‌افزار قابل پیکربندی 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 فرارگیری واحد قابل برنامه‌نویسی به پردازشگر نزدیک‌تر باشد، برنامه‌های مختلف می‌توانند به تعداد دفعات بیشتر با آن ارتباط برقرار کنند و این به دلیل زمان کوتاه برای انتقال اطلاعات است. در این حالت سخت‌افزار بدون دخالت پردازشگر نمی‌تواند برای مدت طولانی به انجام عملیات پردازد و معمولاً مقدار سخت‌افزار در دسترس نیز محدود است. فاصله ارتباط طولانی‌تر به معنی سطح موازی‌سازی بالاتر است و بهمین سبب سرعت و تعداد دفعات ارتباط کاهش خواهد یافت. در کاربردهایی که به دفعات ارتباط زیاد و با سرعت بالا احتیاج است، چنین ساختاری احتمالاً موجب کاهش زمان اجرای برنامه نخواهد شد. در پژوهش حاضر اف.پی.ای و سی.پی.بو دو واحد جدا از هم هستند که با ارتباطات بسیار سریع و بدون واسطه بهم 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 قابل پیکربندی 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 قابل پیکربندی وجود دارد. یکی از این روش‌ها مربوط به استفاده از اتصالات کوتاه^{۱۰} می‌باشد که در آن، یک سری سیم کوتاه برای انتقال اطلاعات محلی استفاده می‌شوند. می‌توان این اتصالات را توسط جعبه‌های سوییج بهیکدیگر متصل کرد تا به عنوان سیم‌های طولانی‌تر مورد استفاده قرار گیرند. سیم‌های طولانی‌تر را می‌توان برای اتصال محلی و کلی استفاده کرد. اتصال سلسله مراتبی^{۱۱} به این شکل انجام می‌شود که بعد از اتصال

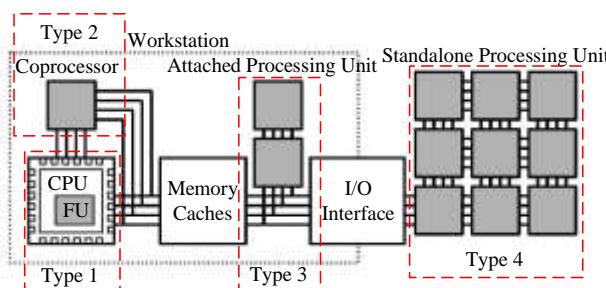


Fig. 1 FPGA and CPU connection configurations [13]

[۱۳] شکل ۱ پیکربندی‌های مختلف برای اتصال اف.پی.جی.ای و سی.پی.بو

^۹ Cache

^{۱۰} Segmented Routing

^{۱۱} Hierarchical Routing

ساخت. نام این دستگاه ای.پی.^۱-300 بود که یک پنجه کوارتزی روی آن تعییه شده بود و با تایش نور ماوراء بنفش به آن ساختارش پاک می‌شد و برای برنامه‌نویسی جدید آماده می‌شد.

امروزه چنین سخت‌افزارهایی دارای اجزای مختلفی هستند که از جمله مهم‌ترین آن‌ها سلول حافظه^۲ است. این اجزای حافظه‌دار به عنوان جداول مراجعه^۳ استفاده می‌شوند تا به کمک 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 اف.پی.ای معمولاً فرق اساسی این سخت‌افزارها در مقایسه با میکروپردازشگر^۴‌های 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 معمولاً با استفاده از زبان توصیف سخت‌افزاری^۵ مانند وی.اچ.دی.ال^۶ و وریلاغ^۷ پیکربندی می‌شود. این عماری امکان پردازش و حل مسائل مختلف از جمله معادلات دیفرانسیل را به صورت سخت‌افزاری فراهم می‌کند. سی.پی.بو به عنوان شناخته شده‌ترین پردازشگر موردن استفاده، سخت‌افزاری است که قابلیت تغییر در ساختار آن پس از ساختش وجود ندارد.

3- اتصال اف.پی.جی.ای و سی.پی.بو

سخت‌افزار قابل پیکربندی (اف.پی.جی.ای) معمولاً در انجام عملیات به خصوصی 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 به کمک آن بیاید. به این ترتیب بازدهی کل سیستم به شکل قبل توجهی بالا خواهد رفت. در سیستمی که حاوی اف.پی.جی.ای و میکروپردازشگر است، روش‌های مختلفی برای اتصال آن‌ها به هم وجود دارد. با توجه به "شکل 1" می‌توان الگوهای زیر را برای برقراری ارتباط بین اف.پی.جی.ای و سی.پی.بو پیشنهاد کرد (در این شکل اف.پی.جی.ای‌ها با رنگ خاکستری مشخص شده‌اند).

الگوی اول: سخت‌افزار قابل پیکربندی را می‌توان به تنظیمی برای ایجاد واحدهای عملیاتی درون یک پردازشگر میزبانی کرد و به این شکل یک محیط برنامه‌نویسی سنتی به وجود دارد. واحد قابل پیکربندی در اینجا به عنوان واحد تابع از پردازشگر عمل می‌کند و در ورودی و خروجی‌های آن رجیسترها یکی برای کنترل ورود و خروج داده قرار دارد.

الگوی دوم: ممکن است یک واحد قابل پیکربندی به عنوان 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 قابل پیکربندی 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 قابل پیکربندی صادر می‌کند.

الگوی سوم: ممکن است واحد قابل برنامه‌نویسی به عنوان یک پردازشگر از

^۱ EP300

^۲ Memory Cell

^۳ Lookup Table

^۴ Configuration

^۵ Microprocessor

^۶ Hardware Description Language

^۷ VHDL

^۸ VERILOG

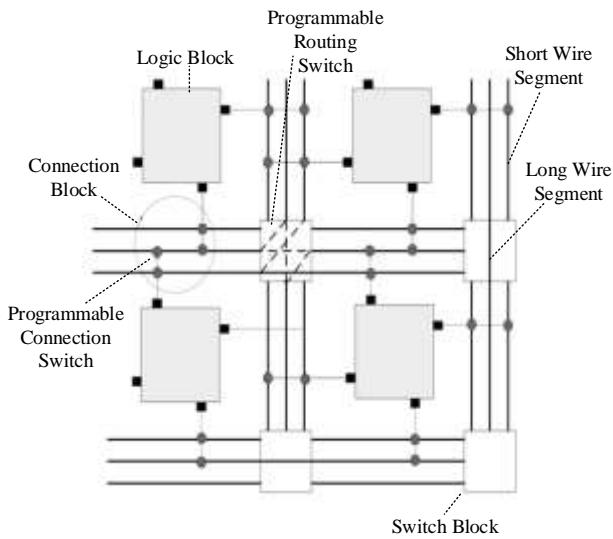


Fig. 2 Interconnection architecture inside an FPGA [14]

شکل 2 شماتیک سیم کشی 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 سخت افزار 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 سخت افزاری نیاز به دانش بالا از سخت افزار و همچنین صرف زمان زیادی دارد. خوب شناختن این مشکل توسط زبان های توصیف سخت افزار سطح بالا (اج.ال.^۹، که عموماً براساس زبان برنامه نویسی سی ساخته می‌شوند، کمرنگتر می‌شود. اج.ال.ها زمان توسعه نرم افزار را کاهش می‌دهند اما برنامه نوشته شده با این زبان ها برای اجرای پهلوی سخت افزار نیازمند بهینه سازی دستی می‌باشد. خانواده جدید اف.پی.جی.ای.ها استفاده از اعداد اعشاری و انجام اعمال ریاضی روی آن ها را ساده تر می‌کنند و به این ترتیب برای انجام اعمال ریاضی مناسب هستند.

مراحل مختلفی در فرآیند طراحی یک مدار وجود دارد. مشخص کردن مدار^{۱۰}، فرآیند تعریف توابعی است که قرار است روی سخت افزار قابل پیکربندی 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

جزای داخلی گروههای کوچک توسط سیم های کوتاه، برای اتصال این گروههای از سیم های طولانی تر در مرز این گروههای استفاده می‌شود. معمولاً اتصال سلسله مراتی را برای حالتی که قرار است بیشتر ارتباطات بین اجزای داخلی گروههای و قسمت کمتری از ارتباطات بین گروههای باشد، بهینه سازی می‌کنند.

۵- سخت افزار مورد استفاده و روش پیکربندی

سخت افزار 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 سخت افزار قابل برنامه ریزی (اف.پی.جی.ای) و پردازشگر (سی.پی.یو) وجود دارند. تراشه اصلی مورد استفاده در این پژوهش مدل زینک-^۱ 7020 از خانواده زینک-7000 ساخت شرکت زایلینکس^۲ است. تراشه های این خانواده از نوع 28 نانومتر مقیاس پذیر سری هفتم اف.پی.جی.ای های ساخت شرکت زایلینکس هستند. هر تراشه برای کاربردهای خاص طراحی و بهینه سازی شده است. تراشه های سری زد-7010، 7015 و 7020 براساس تکنولوژی منطق قابل پیکربندی آرتیکس^۳ از شرکت زایلینکس ساخته شده اند و در کاربردهای با حجم محاسبات بالا و مصرف توان کم، بهینه هستند. تراشه های سری زد-7030، 7035، 7045 و 7100 براساس تکنولوژی منطق قابل پیکربندی کینتکس^۴ از شرکت زایلینکس ساخته شده اند و در کاربردهای با نیاز عملکردی بالاتر و گذردهی خروجی و ورودی زیاد مناسب هستند. تراشه موردنظر به تنهایی 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^۵ شناخته شده و ساخت شرکت ام.وای.آی.آر^۶ است. بر روی این بورد ارتباطات خارجی یواس.پی.^۷، شبکه، حافظه فلاش و غیره تعییه 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.ای-7-کیو^۸، ساخت شرکت اینتل با بهینه سازی فرکانس عملکرد ۱.۷۳ گیگاهرتز است. این سی.پی.یو دارای چهار هسته پردازشی فیزیکی است. هر کدام از این هسته های فیزیکی می‌توانند به صورت همزمان دو عملیات جداگانه را انجام دهند، پس در کل هشت هسته پردازشی مجازی دارد. بهینه سازی توان محاسبات این پردازشگر برابر ۱۳.۸۴ گیگافلاب (یعنی ۱۳.۸۴ میلیارد عملیات اعشاری در هر ثانیه!) است. البته باید توجه داشت که این مقدار اسمی بهینه سازی توان محاسباتی سی.پی.یو است و در عمل هیچ وقت امکان دستیابی به این عدد امکان پذیر نیست.

هر چند عملکرد اف.پی.جی.ای ها برای کاهش زمان اجرای برنامه ها ثابت شده است ولی انجام محاسبات با سخت افزارهایی که قابلیت تغییر در ساختار آن ها بعد از ساختشان وجود دارد (مانند اف.پی.جی.ای) نیازمند افرادی با تخصص بالا است که با استفاده از زبان توصیف سخت افزار، اف.پی.جی.ای را برنامه ریزی کنند. کدنویسی با زبان توصیف سخت افزار حتی برای افراد متخصص، نیازمند زمان زیاد است و این هدف پژوهش که کاهش زمان حل

¹ Zynq-7020² Xilinx³ Artix⁴ Kintex⁵ Z-turn⁶ MYiR⁷ USB⁸ Core i7 Q740⁹ High Level Language (HLL)¹⁰ Circuit Specification

برای قابلیت‌سنجدی سخت‌افزار اف.پی.جی. ای انتخاب شده است. 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 لابلس به صورت رابطه (1) نوشته می‌شود.

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

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 به صورت رابطه (2) نوشته می‌شود.

$$\begin{aligned} \psi(x, y) \\ = \frac{4}{\pi^2} \sum_{n=0}^{\infty} (-1)^n & \frac{\sin((2n+1)\pi x) \sinh((2n+1)\pi(1-y))}{(2n+1)^2 \sinh((2n+1)\pi)} \end{aligned} \quad (2)$$

برای انجام حل عددی این مساله از روش نقطه‌ای گوس-سایدل^۱ استفاده می‌شود. طبق [16] معادله (1) بعد از گسته‌سازی با این روش به رابطه (3) منجر می‌شود که برای حل عددی مساله لابلس استفاده خواهد شد.

$$\begin{aligned} \psi_{i,j}^{k+1} = \frac{1}{2(1+\beta^2)} & (\psi_{i-1,j}^{k+1} + \psi_{i+1,j}^k \\ & + \beta^2(\psi_{i,j+1}^{k+1} + \psi_{i,j+1}^k)) \end{aligned} \quad (3)$$

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.

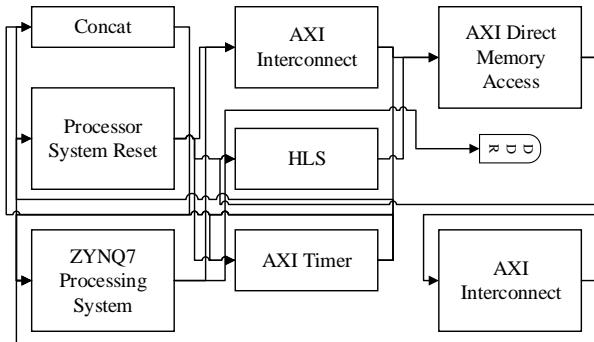


Fig. 4 Block design to solve Laplace problem

شکل 4 طرح بلوکی برای حل مساله لابلس

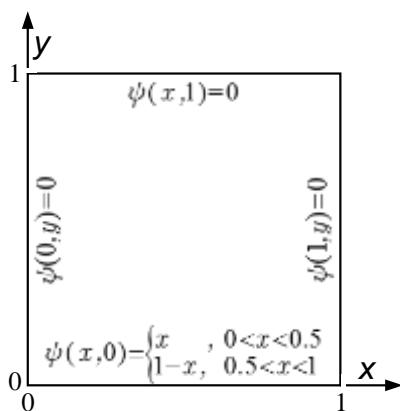


Fig. 5 Boundary conditions for Laplace problem

شکل 5 شرایط مرزی مساله لابلس

^۱ Point Gauss-Seidel

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 ساخته شده مانند جمع کننده‌ها و ضرب-کننده‌ها نیز روشنی است که از نظر دشوار بودن، بین دو روش بالا فراتر می‌گیرد.

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 اچ‌ال‌اس^۲، ویوادو^۳ و زایلینکس اس.دی.کی^۴ 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 آی‌پی^۵ و برقراری اتصالات بین آن‌ها به صورت گرافیکی در نرم‌افزار ویوادو انجام می‌شود. نمونه ساختار طراحی شده برای حل مساله لابلس در "شکل 4" نمایش داده شده است.

6- زمان سنج با دقت بالا

عملیات اندازه‌گیری عملکرد یک سیستم محاسباتی شامل اندازه‌گیری زمان پاسخ‌دهی، توان عملیاتی، دیرکرد و همچنین زمان حل الگوریتم می‌شود. هر کدام از این عملیات شامل اندازه‌گیری اعمال انجام شده در فاصله زمانی شروع تا پایان آن‌ها می‌شوند. زمان سنج‌ها در یک سیستم محاسباتی دو نوع هستند؛ ساعت‌های مطلق و ساعت‌های اختلافی. ساعت‌های مطلق به زمان سنج‌هایی گفته می‌شود که زمان دقیق در روز را نمایش می‌دهند و واپسی به زمان سنج‌های خارجی هستند. ساعت‌های اختلافی، بازه زمانی را اندازه‌گیری می‌کنند و عموماً براساس منابع خارجی کار نمی‌کنند. کیو.پی.سی^۶ یک نمونه ساعت اختلافی در سیستم عامل ویندوز^۷ است که با هیچ زمان سنج 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 بازبایی مقادیر شمارنده. خصوصیات هر کدام 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.بو از زمان سنج دقیق کیو.پی.سی استفاده می‌شود. دقت این زمان سنج تا یک میکروثانیه است. برای محاسبه زمان انجام حل روی سخت‌افزار قابل پیکربندی از زمان سنج دقیق تری که روی اف.پی.جی.ای مورد استفاده در این پژوهش تعییه شده است، استفاده می‌شود. این زمان سنج به صورت یک آی‌پی. داخل بسته نرم‌افزاری ویوادو قابل استفاده است. این آی‌پی.بو با نام "اکسی.تایمر"^۸ شناخته می‌شود و ارتباطات آن با سایر آی‌پی‌ها در "شکل 4" مشخص شده است. دقت محاسبه زمان این آی‌پی تا دو نانو ثانیه است.

7- معادله لابلس

باتوجه به کاربرد معادله لابلس در جریان پتانسیل و انتقال حرارت پایای دو بعدی، این معادله به عنوان نمونه‌ای از مسائل دینامیک سیالات محاسباتی

¹ Vivado HLS

² Vivado

³ Xilinx SDK

⁴ Intellectual Property

⁵ Query Performance counter (QPC)

⁶ Windows

⁷ AXI Timer

لایپلاس بوده است. در این بخش نتایج زمانی به دست آمده از حل مساله لایپلاس با نتایج سایر مراجع مقایسه شده است. در [4] حل یک نقطه از شبکه عددی این مساله روی هر دو ساخت‌افزار سی‌پی‌یو و اف‌پی‌جی‌ای انجام شده است. ساخت‌افزارهای استفاده شده توسط [4] با ساخت‌افزارهای مورد استفاده در پژوهش حاضر متفاوت هستند. بهمین دلیل لازم است پارامتری عمومی برای مقایسه نتایج انتخاب شود. متغیری که می‌تواند توان محاسباتی را صرف‌نظر از ساخت‌افزار استفاده شده مشخص کند، تعداد دوره‌های فرکانس زمانی (کلک^۱) لازم برای حل مساله است. تعداد کلک‌ها لازم برای تکمیل فرآیند حل، یک پارامتر بی بعد است و از این نظر نیز نسبت به سایر متغیرهای با بعد برتری دارد. در جدول 2 نتایج زمانی و تعداد کلک در حل یک نقطه از شبکه عددی مساله لایپلاس 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- نتیجه گیری

در پژوهش حاضر راهکار جدیدی برای انجام محاسبات عددی با استفاده از یک سیستم الکترونیکی پیشنهاد شده است. حل عددی مساله لایپلاس به عنوان محاسبات نمونه روی ساخت‌افزار قابل پیکربندی پیشنهادی این

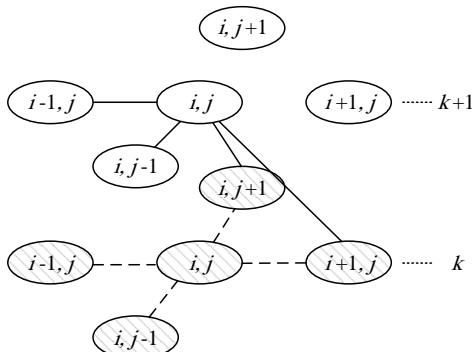


Fig. 6 Stencil of the point Gauss-Seidel method

شکل 6 الگوی روش نقطه‌ای گوس-سایدل

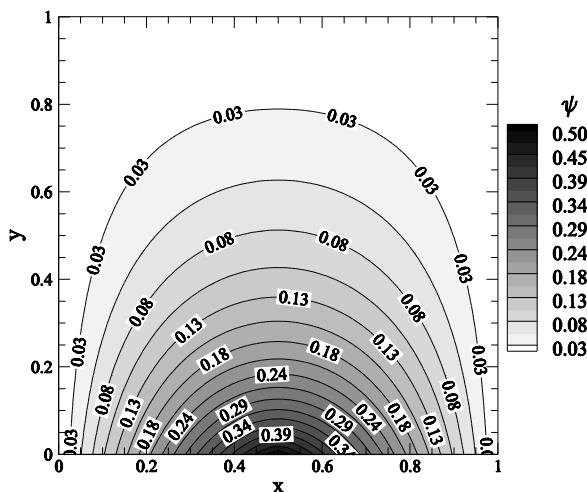


Fig. 7 ψ contour in Laplace numerical solution

شکل 7 کانتور ψ در حل عددی مساله لایپلاس

8- نتایج حل عددی

بعد از طی مراحل لازم برای طراحی سخت‌افزار و اعمال آن روی اف‌پی‌جی‌ای و سی‌پی‌یو، 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 شبکه 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 $100/\pi$ ، 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 لایپلاس، مقادیرتابع جريان برروی دو خط $y = 0.2$ و $y = 0.5$ در حل عددی و تحلیلی در "شکل 9" مقایسه شده‌اند. همان‌گونه که از این شکل مشخص است دو حل با

هم انتطاق بسیار خوبی داشته و این نشان‌دهنده صحت حل عددی است.

در "شکل 10" نمودار زمان حل مساله (محور عمودی سمت راست) با استفاده از سی‌پی‌یو و اف‌پی‌جی‌ای در دقت‌های مختلف بر حسب تعداد نقاط شبکه محاسباتی آمده است. مشاهده می‌شود که در یکی از حالت‌ها تا 20 برابر نیز افزایش سرعت به دست آمده است. از آنجایی که برای به دست 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 ساخت‌افزار قابل پیکربندی، به علت وجود سه حلقه تکرار منطقی تودرتو در برنامه کامپیوتری آن، بهینه نخواهد بود. بنابراین بهتر است فقط افزایش سرعت حل یک گام حل روی اف‌پی‌جی‌ای بررسی شود. البته در صورتی که ساخت‌افزار قوی‌تری از نظر قدرت و فرکانس پردازش در اختیار باشد، می‌توان مساله را بهصورت کامل روی آن حل کرد.

نمودار زمان حل مساله لایپلاس روی اف‌پی‌جی‌ای در مقابل تعداد نقاط شبکه عددی در "شکل 11" آمده است. همان‌گونه که از این شکل مشخص است، زمان حل روی ساخت‌افزار قابل پیکربندی با تعداد نقاط شبکه رابطه مستقیم و خطی دارد. شبیه 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 اف‌پی‌جی‌ای انجام شده‌اند، یکی از مسائل مهم مورد بررسی حل مساله

¹ Clock

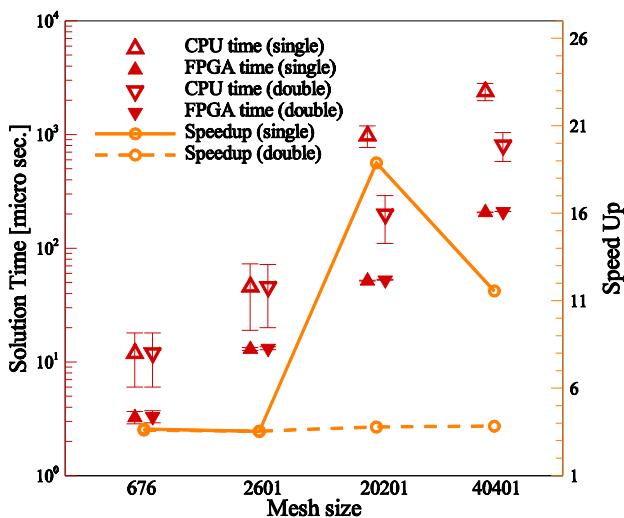


Fig. 10 Time duration and speedup for Laplace solution

شکل 10 نمودار زمان و تسريع حل مساله لابلانس

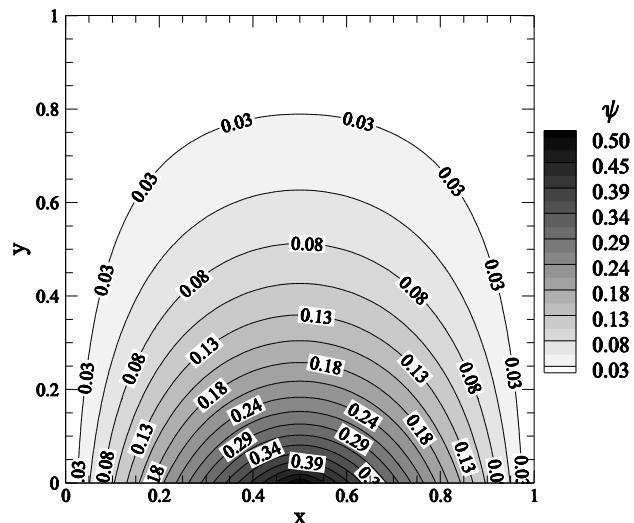
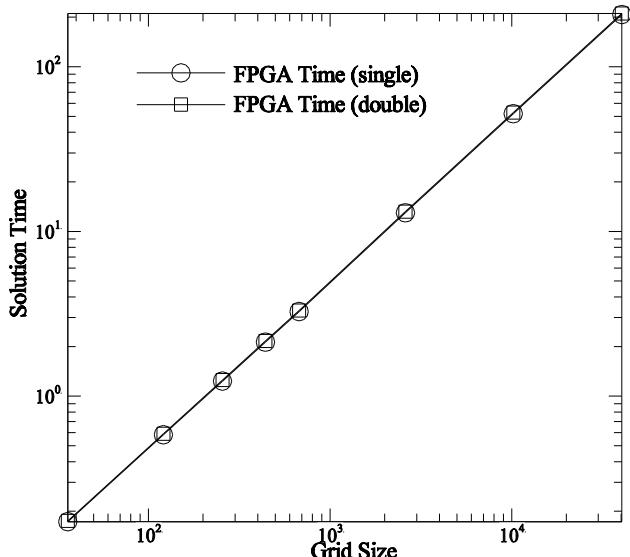
Fig. 8 ψ contour in Laplace exact solutionشکل 8 کانتور ψ در حل تحلیلی مساله لابلانس

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 لابلانس
Table 2 Single node solution time duration in Laplace problem

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

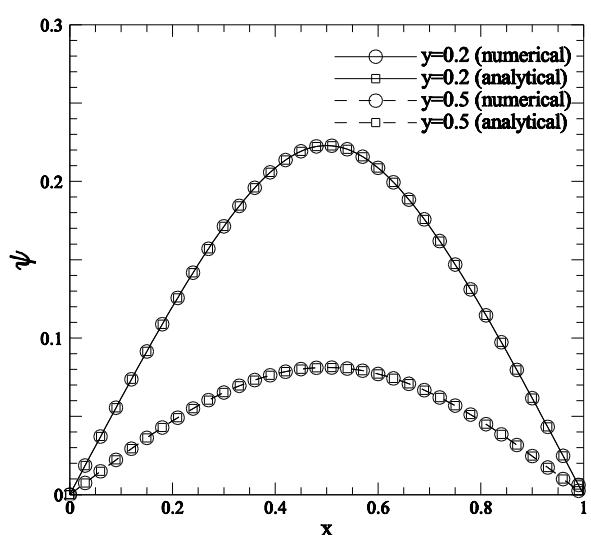
سی‌پی‌یو به اف‌پی‌جی‌ای، روش‌های الگوریتمی و بهینه‌سازی‌های نرم‌افزاری زیادی برای رسیدن به این هدف استفاده شدند. یکی از دلایل عدم افزایش سرعت محاسبات به اندازه زیاد، عدم کاهش دقت حل با استفاده از اف‌پی‌جی‌ای نسبت به سی‌پی‌یو بوده است. درصورتی که دقت‌های پایین‌تر برای اعداد در حل روی اف‌پی‌جی‌ای انتخاب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

جدول 1 نرم خطای حل تحلیلی و عددی مساله لابلانس

Table 1 Exact and numerical Laplace solution error norm

اندازه شبکه	گام حل	نم اول خطأ	نم دوم خطأ	نم بی‌نهایت خطأ
1000	π	0.08705	0.09807	0.15752
5000	50	0.00502	0.00731	0.03524
10000	100	0.00502	0.00731	0.03524
1000	π	0.38357	0.43655	0.77434
5000	50	0.03569	0.06008	0.09373
10000	100	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 کامپیوتراستی مقایسه شده است. هدف اصلی از پژوهش حاضر تلاش برای تسريع و کاهش زمان محاسبات عددی بهصورت سخت‌افزاری است. حل مساله لابلانس با استفاده از اف‌پی‌جی‌ای تا 20 برابر سریع‌تر از حل روی یک سی‌پی‌یو یا 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 linesشکل 9 مقایسه مقدار ψ از حل‌های عددی و تحلیلی بر روی دو خط افقی

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.

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

۱۰- 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