سومین کنفرانس بین المللی سالانه کامپیوتر انجمن کامپیوتر ایران
۲ الی ۴ دیماه ۱۳۷۶
دانشکده مهندسی کامپیوتر - دانشگاه علم و صنعت ایران

یک زبان سطح بالا برای برنامه سازی الگاریتم های تپنده[1]

آسیه سینایی خسروشاهی — کارشناس ارشد
محمدرضا میبدی — عضو هیئت علمی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر

Keyword: Systolic Algorithm, Parallel Algorithm, Parallel Programming Language

چکیده:

در این مقاله یک زبان سطح بالا برای توصیف الگاریتم های تپنده و همچنین پیاده سازی آن ارائه میگردد. این نرم افزار به عنوان ورودی یک الگوریتم تپنده را به زبان سطح بالایی که خاص اینکار طراحی شده است، دریافت میکند وسپس کد برنامه موازی متناظربا آن را به زبان اکام[2] تولید می نماید. این کد میتواند بر روی شبکه ای از ترانسپیوتر ها اجرا گردد. کاربر با استفاده ازاین نرم افزار میتواند بدون اینکه آشنایی با برنامه سازی موازی و زبان اکام داشته باشد کد اجرایی الگوریتم تپنده مورد نظر را تولید کند. همچنین برای کمک به بررسی نحوه اجرای الگوریتم امکان نمایش گرافیکی الگوریتم نیز تعبیه شده است که از این طریق کاربر میتواند جریان داده ها در پردازنده ها و حاصل اجرای عملیات در هر مرحله از الگوریتم را مشاهده نمایند.

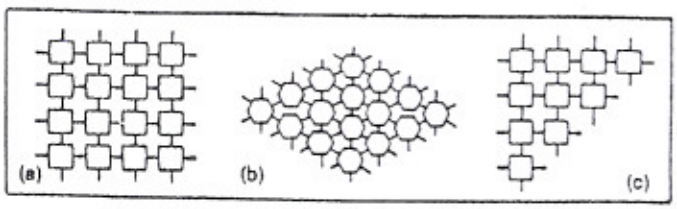
مقدمه :

بنابرتعریف، یک ساختار بسیار موازی، مجموعه ای از عناصر محاسباتی یکسان هستند که کار خاصی را انجام می دهند. ساختارهای بسیار موازی، ابتداء به عنوان یک فرم معماری برای پیاده سازی سخت افزاری برخی الگوریتم ها مثل ضرب ماتریس ها مطرح شدند. دومتدولوژی عمومی برای نگاشت محاسبات به ساختارهای سخت افزاری مطرح شده است که عبارتنداز: معماری تپنده و معماری جبهه موج[3] [۸،۱۰،۱۱] ماشین هایی که براساس این متدولوژی ها ساخته میشوند از تعداد زیادی عناصر پردازشی تشکیل می یابند (احتمالا" ناهمگن) که شبکه ای از پردازنده ها را بوجود می آورند. در معمارهای تپنده، داده ها از یک طرف شبکه وارد شده، برای مدتی روی آنها

---

1) Systolic
2) Occam
3) Wavefront

پردازش صورت میگیرد و سپس از سوی دیگرشبکه خارج میشوند. جریان داده در سیستمهای تپنده را میتوان به جریان خون از قلب به اعضاء بدن و برعکس تشبیه کرد. آرایه های تپنده از بسط مفهوم محاسبات لوله ای[1] حاصل شده اند[۲]، این بسط دووجه دارد: اول آنکه لوله ها معمولا" یک بعدی هستندو جریان داده یک سویه دارند، در حالیکه معماری های تپنده میتوانندچند بعدی وبا جریان داده چند سویه باشند. از آن گذشته در لوله ها فقط نتایج جزئی در جریان هستند در حالیکه در معماری تپنده هم نتایج جزئی حاصل از محاسبه و هم داده های ورودی بین پردازشگرها در حرکت هستند. سیستم تپنده از مجموعه ای از سلولهای به هم مرتبط تشکیل می یابد و هر سلول قادر به انجام چند عمل ساده است. ارتباط با دنیای خارج فقط از طریق سلولهای مرزی انجام میگرد. در شکل ۱ : سه نوع آرایه تپنده دو بعدی : نشان داده شده است.



آرایه های سیستولیک دو بعدی : (a) گونه R ، (b) گونه H و (c) گونه T

شکل ۱

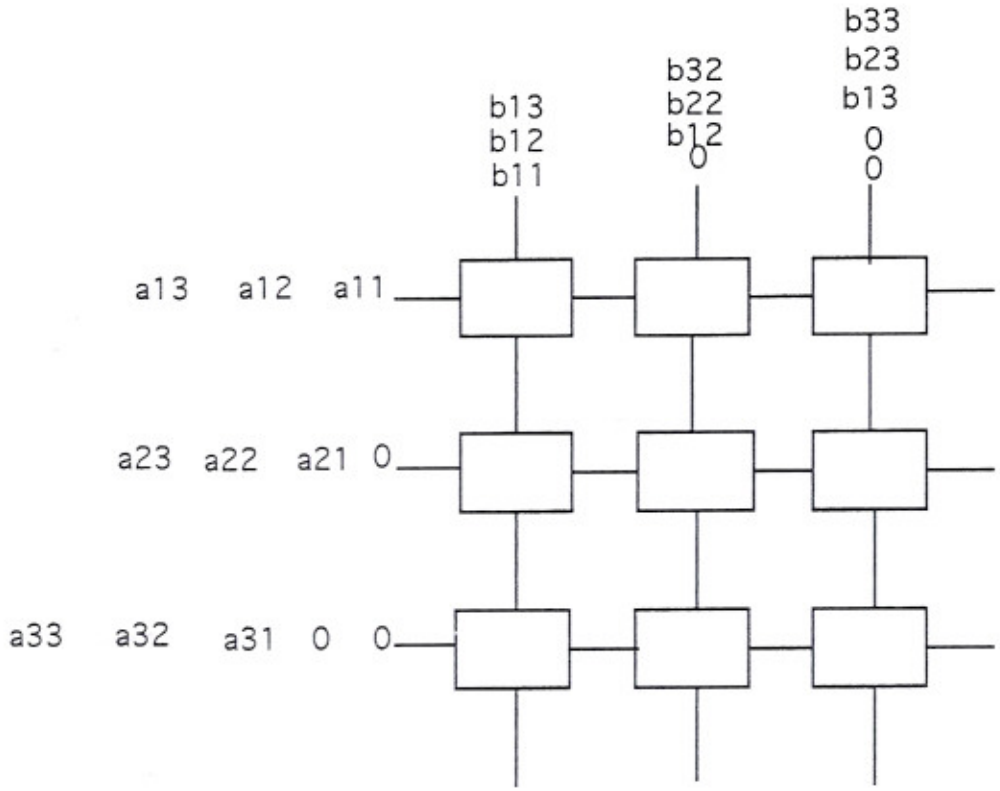
برای درک بیشتر نحوه کار سیستم های تپنده به بررسی یک الگوریتم تپنده برای محاسبه حاصلضرب دو ماتریس می پردازیم. برای ضرب دو ماتریس $k\times n$ و $m\times k$ میتوان ازیک آرایه تپنده به ابعاد $m\times n$ استفاده کرد. ساختار چنین ضرب کننده ای در شکل ۲ نشان داده شده است. برای سادگی $m=۳$، $k=۳$، $n=۴$ فرض شده است. عناصر ماتریس A از سمت چپ وارد میشوند. عناصر هر سطر بایک ضربه ساعت تأخیرنسبت به سطر قبل وارد میشوند (نقطه های سیاه در شکل این تأخیر را نشان می دهند، میتوان فرض کرد که در این ضربه های ساعت مقدار صفر وارد آرایه میشود که هیچ تأثیری در نتیجه محاسبه ندارد).عناصر ماتریس B نیز از قسمت بالا وارد آرایه میشوند. هر عنصر بعد از طی آرایه از سمت مقابل خارج میشود. هر سلول آرایه یک عنصر از ماتریس نتیجه را محاسبه میکند. متغیر محلی c_{ij} در هر عنصر پردازشی مقدار اولیه صفر دارد و بارسیدن دو عنصر ماتریس A و B محاسبه زیر انجام میشود:

$$c_{ij}=c_{ij}+a_{ik} * b_{kj}$$

بعد از انجام محاسبه، a_{ij} به پردازنده راست و b_{ij} به پردازنده پایینی منتقل میشود و سیکل بعدی عملیات آغاز میشود پس از زمان $O(n^2)$ محاسبه ماتریس نتیجه کامل میشود.

---

۱) Pipelining



شکل ۲- ضرب ماتریسی

پیاده سازی سخت افزاری الگوریتم های تپنده نیاز به تلاش مهندسی بسیار زیادی دارد که برای بسیاری از کاربردها قابل توجیه نمی باشد به همین دلیل اخیرا" توجه بسیار زیادی به پیاده سازی این الگوریتم ها در نرم افزار گردیده است. نرم افزارهای مختلفی برای تولید برنامه های تپنده تولید شده که از آن جمله میتوان SDEF و ADVIS را نام برد[۳و۹]. این نرم افزارها متکی به تشخیص بردارهای وابستگی یک الگوریتم و تعیین یک تابع برای نگاشت آن بردارها به یک فضای زمان و مکان هستند. در واقع فضای مکان تعداد سلولها و ارتباط بین سلولها در آرایه تپنده را مشخص میکند و فضای زمان مشخص میکند که در هر لحظه کدام سلولها باید فعال باشند. باتوجه به اینکه تعیین بردارهای وابستگی الگوریتم و پیدا کردن یک تابع نگاشت مناسب برای الگوریتم کار پیچیده ای است، استفاده از این نرم افزارها بسیار مشکل می باشد.

شیوه ای که در این مقاله ارائه میگردد، متکی بر تعریف سطح بالای الگوریتم تپنده است که فقط مستلزم شناخت الگوریتم تپنده میباشد و در نتیجه کاربر میتواند به شیوه ساده تری به کد موازی الگوریتم مورد نظر دست یابد، کاربر الگاریتم تپنده خودرا با استفاده از یک زبان سطح بالا توصیف میکند. این توصیف سپس به کد موازی برای اجرا برروی یک کامپیوتر موازی تبدیل میگردد.

زبانهای موازی مختلفی برای پیاده سازی الگوریتم های موازی ایجاد شده اند[۳]. باتوجه به ویژگیهای زبان اکام، کد موازی متناظر با ماشین الگورتیم های تپنده به زبان اکام تولید میشود. عمده ترین ویژگی این زبان که آن را برای پیاده سازی الگوریتم هایی که بر روی چندین پردازنده

بهم مرتبط اجرا میشوند مناسب می سازد، مفهوم کانال و پروتکل است [۴,۷]. ادامه مقاله زبان طراحی شده برای توصیف الگوریتم های تپنده را ارائه میکند.

زبان تعریف الگوریتم های تپنده

زبان تعریف الگوریتم های تپنده از شش عبارت تشکیل می یابد که برای تعیین ساختارشبکه، تعریف ورودی و خروجی های و همچنین تعیین عملیاتی که هر سلول الگوریتم تپنده اجرا میکند بکار میروند. حال به توصیف هریک از جملات زبان می پردازیم.

عبارت تعریف شبکه

```
NETWORK     TYPE     ARRAY (m)
                     MESH (m,n)
                     HEX (m,n)
```

سه نوع شبکه توسط سیستم پشتیبانی میشود که عبارتنداز: آرایه خطی با m واحد پردازشی، آرایه دو بعدی با m×n واحد پردازشی و آرایه دو بعدی شش ضلعی.

عبارت تعریف متغیرهای ثابت در هر سلول

این متغیرها به عنوان متغیر کمکی برای حفظ نتایج بینابینی محاسبات یا به عنوان مقادیر ثابت در محاسبات استفاده میشوند. برای تعریف این متغیرها باید نام، نوع و مقدار اولیه را مشخص کرد.

```
VARIABLE  id  TYPE    INT | INT16 | INT32      INITIAL (VI1,...,VI m)
                      FLOAT32 | FLOAT64
                      BOOL | BYTE
```

VIi مقدار اولیه متغیر در پردازشگر i ام است.

عبارت متغیرهای ورودی که به صورت تپنده جریان می یابند

به ازاء هر رشته ورودی به آرایه تپنده باید نوع مقادیر ورودی و جهت ورود اطلاعات به هر سلول را تعیین کرد.

```
INPUT id   TYPE     INT | INT16 | INT32      DIRECTION   WEST
                    FLOAT32 | FLOAT64                    EAST
                    BOOL | BYTE                          SOUTH
                                                         NORTH
```

عبارت متغیرهای خروجی که به صورت تپنده جریان می یابند

در صورتی که یک رشته داده به عنوان خروجی آرایه تپنده استفاده شود، توسط عبارت OUTPUT معرفی میگردد.

OUTPUT id	TYPE	INT | INT16 | INT32 FLOAT32 | FLOAT64 BOOL | BYTE	DIRECTION	WEST EAST SOUTH NORTH

عبارت دستورالعمل های پردازشی

در این بخش،دستورالعملهایی که با هر ضربه ساعت در هرسلول اجرا میگردد، مشخص میشوند. تمام متغیرهایی که در این دستورالعمل ها استفاده میشوند، باید قبلا" توسط جملات VARIABLE، INPUT یا OUTPUT معرفی شده باشند.

PROCESS { statement ; statement; }

بخش PROCESS از تعدادی دستورالعمل تشکیل می یابد که توسط ; از یکدیگر جدا میشوند و کلیه دستورالعمل ها باید داخل { } قرار گیرند.

عبارت تغذیه کننده[1]

این عبارت مشخص میکند که سلولهای مرزی آرایه در چه زمانهایی کار خواندن اطلاعات ورودی را انجام میدهند. در واقع تعدادی فرآروند ورودی در سلولهای مرزی وجود دارد که کار خواندن داده ها از محیط خارج از آرایه را انجام می دهند. ممکن است تمام فرآروندهای ورودی یکسان باشند یا در سیکل های زمانی متفاوت فعال باشند.

FEEDER	Id	ACTIVE	FOR	Process number ALL
		THEN	From Valuel to Value2 if Exp	

در سیکل هایی که فرآروند ورودی فعال نیست، مقدار صفر فرستاده میشود که تأثیری در محاسبات نخواهد داشت.

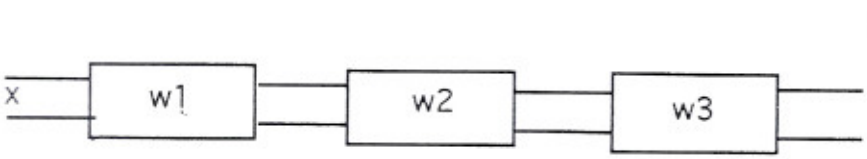
مثال: برای روشن شدن نحوه استفاده از زبان تعریف الگوریتم های تپنده، عبارات لازم برای تعریف الگوریتم تپنده کانولوشن را ذکر میکنیم. در مسأله کانولوشن یک رشته ورودی $\{x_1, x_2, \ldots x_n\}$ و مجموعه ای از $w_1, w_2, w_3, \ldots, w_n$ داریم که برای محاسبه رشته خروجی $\{y_1, y_2, \ldots y_{n+1-k}\}$ به کار می روند. y_i طبق رابطه زیر محاسبه می شود:

$$y_i = w_1 x_i + w_2 x_{i+1} + \ldots w_k x_{i+k-1}$$

ساختار آرایه تپنده این الگوریتم با فرض $k=3$ در شکل ۳ نشان داده شده است.

---

1) Feeder



شکل ۳

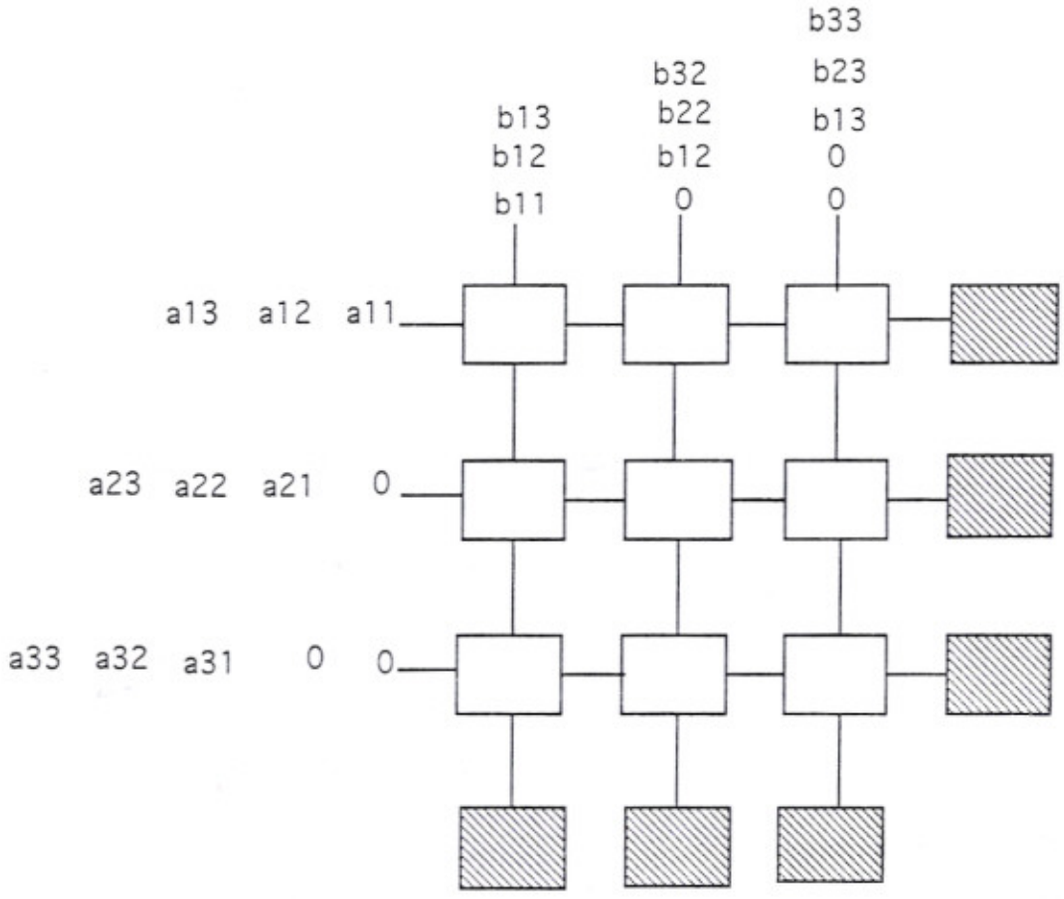
مقادیر x از چپ به راست و مقادیر y از راست به چپ حرکت می کنند. برای اینکه مقادیر x و y در لحظات مناسب به یکدیگر رسیده و الگوریتم به درستی عمل کند، مقادیر y,x باید یک سیکل در میان وارد شوند و در سیکل های میانی مقدار صفر وارد شود.الگوریتم تپنده کانولوشن به صورت زیر تعریف میشود:

```
NETWORK      TYPE   ARRAY (3);
VARIABLE  w  TYPE  INT INITIAL   (1,2,3);
INPUT     X  TYPE  INT DIRECTION WEST;
OUTPUT    Y  TYPE  INT  DIRECTION  EAST;
PROCESS      {Y=Y+w*X;};
FEEDER    X   ACTIVE  FOR  I  THEN  FROM  I TO  6;
FEEDER    Y   ACTIVE  FOR  I  THEN  FROM  3 TO  10;
```

مثال: ماشین الگوریتم ضرب ماتریسی را میتوان توسط برنامه زیر تعریف کرد:

```
NETWORK    TYPE  MESH (3,3);
VARIABLE  C   TYPE  INT INITIAL   (0,0,0,0,0,0,0,0,0);
INPUT     A  TYPE  INT DIRECTION WEST;
INPUT     B  TYPE  INT DIRECSTION   SOUTH;
PROCESS      {C=C+A*B;};
FEEDER     A   ACTIVE  FOR  I   THEN   FROM  I  TO  3;
FEEDER     A   ACTIVE  FOR  2   THEN   FROM  2  TO  4;
FEEDER     A   ACTIVE  FOR  3   THEN   FROM  3  TO  5;
FEEDER     B   ACTIVE  FOR  I   THEN   FROM  I  TO  3;
FEEDER     B   ACTIVE  FOR  2   THEN   FROM  2  TO  4;
FEEDER     B   ACTIVE  FOR  3   THEN   FROM  3  TO  5;
```

در این الگوریتم عناصر ماتریس A از سمت چپ و هر سطر با یک سیکل تأخیر نسبت به سطر قبل ورد آرایه میشوند. همچنین عناصر ماتریس B ازبالای آرایه وهرستون بایک سیکل تأخیر نسبت به ستون قبل وارد آرایه میشوند. هرعنصر از ماتریس نتیجه در متغیر C پردازنده ها ذخیره میگردد. در هر سیکل عمل محاسبه C با مقادیر B,A دریافت شده از سلولهای مجاور انجام شده و سپس مقادیر A , B به سلولهای مجاور انتقال می یابند. ساختار آرایه تپنده مورد نظر در شکل زیر نشان داده شده است.

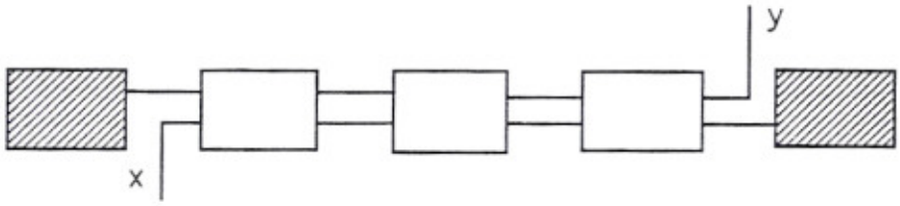


شکل ۴

پیاده سازی الگوریتم های تپنده

برای پیاده سازی الگوریتم های تپنده، یک کامپایلر طراحی شده است. این کامپایلر تعریف الگوریتم را پردازش کرده و براساس آن برنامه اکام متناظر با آن را تولید مینماید. کامپایلر طراحی شده دو بخش دارد. در بخش اول جملات فایل ورودی (تعریف الگوریتم) خوانده شده و از نظر نحوی بررسی میگردد. در صورت وجود خطا، پیغامهای خطا در فایل خروجی نوشته شده و کار کامپلر متوقف میشود. در بخش دوم، در صورتی که خطای نحوی در تعریف الگوریتم وجود نداشته باشد، جملات زبان پارس شده و کد خروجی که مجموعه ای از فرآروندها به زبان اکام است تولید میشود. برنامه تولید شده در صورت اجرا بر روی یک ترانسپیوتر، الگوریتم تپنده مورد نظر را اجرا میکند. گرامر زبان طراحی شده در [۱۳] آورده شده است.

در مرحله تولید کد، برای اینکه ساختار تمام سلولهای آرایه یکسان باشد و مجبور نباشیم تمایزی بین سلولهای مرزی و سایر سلولها قائل شویم، در هر لبه آرایه، یک ردیف سلول اضافی در نظر میگیریم که محلی برای دریافت مقادیری هستند که باید از آرایه خارج شوند. برای مثال اگر الگوریتم مورد نظر نیاز به یک آرایه خطی ۳ عنصری داشته باشد، در برنامه یک آرایه ۵ عنصری تعریف میشود.(شکل ۵)



شکل ۵

در شکل فوق این دو سلول اضافی با هاشور مشخص شده اند و در محاسبات شرکت نمی کنند. در هر سیکل، هر سلول مقدار X را از سلول سمت چپ و مقدار Y را از سلول سمت راست دریافت کرده و محاسبه را انجام میدهد. پس از آن مقدار X به سلول سمت راست و مقدار Y به سلول سمت چپ منتقل میشود، سلولهای واقع در دو لبه نیز از این قاعده مستثنی نیستند، اما مقادیر را به دو سلول اضافی میفرستند که در آنجا از بین میروند، چون به مقادیر آنها دیگر نیازی نداریم. به ازاء هر متغیری که از نوع INPUT یا OUTPUT تعریف شده است، تعدادی فرآروند ورودی ایجاد میشود که کار خواندن اطلاعات از خارج ووارد کردن اطلاعات به ارایه را به عهده دارند. در واقع با توجه به ابعاد آرایه تپنده تعریف شده و جهت جریان داده ها، به ازاءهر سلول مرزی یک فرآروند ورودی ایجاد میشود. این فرآروندها با توجه به اطلاعاتی که در تعریف الگوریتم درباره زمان ورود اطلاعات داده شده است (عبارتهای FEEDER) کنترل میشود. بدین ترتیب اطلاعات در سیکل های زمانی مناسب وارد آرایه تپنده می شوند و در محاسبات شرکت کنند.

یک فرآروند محاسباتی به نام COMPUTE نیز تولید میشود که حاوی دستورالعملهایی است که در هر مرحله توسط هر سلول بایستی اجراگردد. این فرآروند به تعداد سلولهای تپنده تکرار میشوند که در هر سیکل به طور همزمان اجرا میشوند. این فرآروند، وظیفه دارد که داده ها را از سلولهای مجاور دریافت کرده، محاسبه را انجام داده و سپس نتایج را به سلولهای مجاور انتقال دهد. به ازاء هر متغیر OUTPUT تعدادی فرآروند خروجی تولید میشود که نتایج حاصل از محاسبات را از آرایه تپنده دریافت کرده و به خارج انتقال میدهند. فرآروند MAIN و SPAWN نیز برای تعریف ساختارهای مورد نیاز (کانالها و پروتکل ها) و راه اندازی سایر فرآروندها به کار میروند. کد خروجی متناظربا ماشین الگوریتم کانولوشن در [۱۲] آمده است. برای نمایش نحوه عملکرد الگوریتم ها، اجرای الگوریتم به صورت گرافیکی نمایش داده میشود. ابتدا مقادیر متغیرها در هر سلول نمایش داده میشود، با زدن یک کلید توسط کاربرمحاسبات انجام شده و نتایج حاصله در هر سلول نمایش داده میشود. به همین ترتیب، کار در سیکل های بعدی ادامه می یابد تا اجرای الگوریتم خاتمه یابد.

جمع بندی

در این مقاله یک زبان سطح بالا برای توصیف الگاریتم های تپنده و همچنین پیاده سازی آن ارائه گردیده است. با استفاده از این زبان کاربرمیتواند فقط با شناخت الگاریتم تپنده کد موازی قابل اجرا در کامپیوتر های موازی را تولید کند. علاوه براین امکان نمایش گرافیکی اجرای الگاریتم های

تپنده نیز فراهم شده است. با استفاده از این شبیه ساز گرافیکی بررسی الگاریتم های تپنده مختلف برای حل یک مسأله خاص و مقایسه کارآئی آنها ساده تر انجام میگیرد. همچنین امکان معرفی الگاریتم های تپنده عملکرد دقیق آنها در محیط های آموزشی را امکان پذیر می سازد.

مراجع

1- A. V. Aho and J. D. Ullman - Principles of Compiler Design, Addision Wesley , 1979.
2- G. R. Desrochers, Principles of Multiprocessing, McGraw Hill International Edition , 1988.
3- B. R. Engstrom and P. R. Cappello, The SDEF systolic programming system, VLSI Systems and Computations , Editors. H. T. Kung, B. Sproull and G. Steele, Computer Science Press, 1988.
4- M. Geranit Jones, Programming in Occam 2, Perntice Hall, 1988.
5- E. V. Krishnamurthy, Parallel Processing, Addision Wessley, 1989.
6- P. Lee and Z. M. Kedem, Synthesizing Linear Array Algorithms From Nested for Loop Algorithms, IEEE Transaction on Computers, Vol. 37, No. 12, December 1988.
7- Inmos Limited Occam Reference Manual , Prentice Hall, 1988.
8- H. T. kung, Why Systolic Architectures? , Computers, Jan. 1982.
9- D. I. Moldovan, ADVIS: A Software Package for the Design of Systolic Arrays, IEEE Trans. Computer-Aided Design, Jan., 1987.
10-S. Y. Kung, VLSI Array Processor, Prentice Hall International, 1988.
11- M. R. Meybodi, R. L. Furbee, and T.Y. Kong - Simulating Highly Parallel Structures: Systolic Arrays, Technical Report , Ohio University, Computer Science Department, 1987.
12- A. Sinaee and M. R. Meybodi, Techn ical Report, Amirkabir Univ ersity, Computer Engineering Department,1997.

دانشگاه علم و صنعت ایران

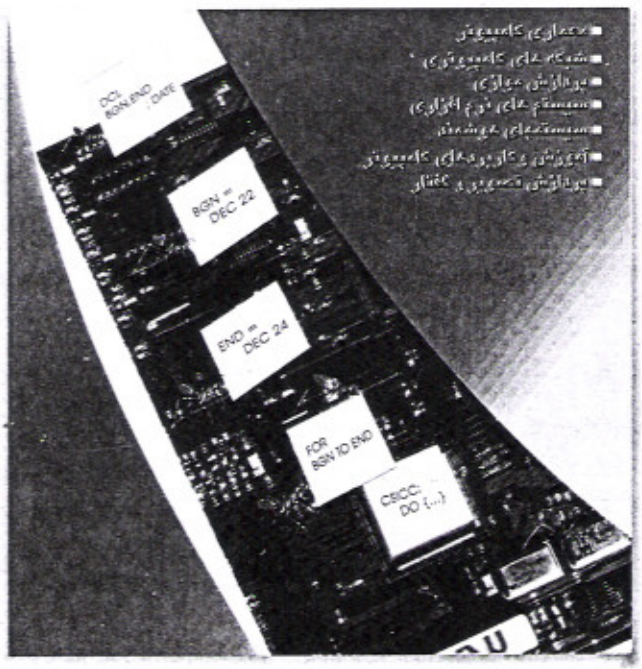
مجموعه مقالات کامپیوت

سومین کنفرانس

بین المللی سالانه

انجمن کامپیوتر ایران

دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
تهران — جمهوری اسلامی ایران



۲ الی ۴ دیماه ۶