یک الگوریتم مرتب سازی موازی برای اتوماتای سلولی دوبعدی

محمدرضا میبدی
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر
meybodi@ce.aut.ac.ir

شهرام گلزاری
گروه برق و کامپیوتر
دانشگاه هرمزگان
golzari@hormozgan.ac.ir

چکیده: یکی از زمینه های مهم و جالب علم کامپیوتر، مرتب سازی است و بهمین دلیل الگوریتمهای متعددی برای انجام آن ابداع شده است. در این مقاله یک الگوریتم مرتب سازی موازی برای اتوماتای سلولی دوبعدی ارایه شده است. هر سلول اتوماتا شامل یکی از عناصری است که باید مرتب شود. قوانین اتوماتای سلولی به نحوی طراحی شده اند که در پایان سطرهای فرد و ستونهای اتوماتا بصورت صعودی و سطرهای زوج بصورت نزولی و در نتیجه کل اتوماتا بصورت مارپیچی مرتب شود. الگوریتم پیشنهادی ساده و محلی بوده و ایده اصلی آن جابجایی مقادیر سلولهای همسایه می باشد.

کلمات کلیدی: اتوماتای سلولی، مرتب سازی، الگوریتمهای موازی

۱-مقدمه

مرتب سازی داده ها یکی از زمینه های مهم و جالب علم کامپیوتر می باشد. مرتب سازی داده ها نقش مهمی در حل مسایل در زمینه های مختلف از قبیل گرافها، هندسه محاسباتی و پردازش تصاویر دارد. از اینرو برای ماشینهای موازی مختلف الگوریتمهای متداولی برای مرتب سازی طراحی شده است. در ماشینهای SIMD و برای آرایه یک بعدی از پردازنده ها الگوریتم انتقال زوج و فرد مطرح شده است که n عنصر داده ای را با استفاده از n پردازنده در n مرحله مقایسه و جابجایی مرتب می نماید. علاوه بر آن می توان به الگوریتم ادغام Thompson-kung اشاره نمود [18] .

اکثر الگوریتمهای موازی گزارش شده برای مرتب سازی مش بر مبنای الگوریتمهای ادغام زوج و فرد Batcher و الگوریتم مرتب سازی Bitonic استوار می باشد. این دو الگوریتم توسط Batcher [2] و برای مرتب سازی داده ها در یک شبکه تک منظوره از پردازنده ها ارایه شده و مبنایی برای مرتب سازی در مدلهای مختلف پردازش موازی می باشد[1,16] . Thompson و Kung یک الگوریتم سریع را بر روی مش دو بعدی با اندیس گذاری مارپیچی مطرح نموده اند. الگوریتم مرتب سازی آنها که بر مبنای الگوریتم زوج و فرد می باشد نیاز به $o(n)$ مرحله مقایسه و جابجایی دارد. بر مبنای تجزیه و تحلیل آنها روش اندیس گذاری سطری برای مرتب سازی مبتنی برادغام زوج و فرد موثر نمی باشد. الگوریتم آنها را می توان بسادگی برای ماتریسهای چندبعدی نیز تعمیم داد. بعدها Kummar و Hirschberg یک الگوریتم کارا مطرح نمودند که n^2 عنصر را در یک مش دو بعدی با اندیس گذاری سطری در $o(n)$ مرحله جابجایی و در $o((\log n)^2)$ مرحله مقایسه و بر مبنای الگوریتم ادغام زوج و فرد مرتب می نماید[10].

الگوریتمهای مرتب سازی برای آرایه های تپنده نیز پیشنهاد شده است [12] . الگوریتم مرتب سازی انتخابی که در حالت ترتیبی n عنصر را با n^2 مقایسه مرتب می کند، در حالت تپنده n عنصر را با استفاده از n سلول تپنده در $2n$ مرحله (در صورت بار شدن داده ها در سلولهای تپنده) و یا در $3n$ مرحله (در صورت وارد شدن داده ها از یک طرف سلولها) مرتب می نماید. الگوریتم مرتب سازی حبابی با استفاده از آرایه های تپنده به الگوریتم انتقال زوج و فرد تبدیل شده و n عنصر را با استفاده از n سلول تپنده در n مرحله مرتب می نماید[12]. اطلاعات بیشتر در مورد الگوریتمهای مرتب سازی را می توان در [1,9,11,16] یافت.

در مورد الگوریتمهای مرتب سازی برای CA کار زیادی انجام نشده است.تنها الگوریتم گزارش شده برای CA منصوب به Gordillo و Luna می باشد که برای CA یک بعدی طراحی شده است. این الگوریتم n عنصر را با استفاده از n سلول در $2n-3$ مرحله مرتب می نماید [7] .

در این مقاله یک الگوریتم موازی برای مرتب سازی اعداد در یک CA دوبعدی با اندیس گذاری مارپیچی ارایه می شود. ایده الگوریتم پیشنهادی بسیار ساده و بر مبنای جابجایی های محلی است. ساختار مورد استفاده نیز ساده بوده و هر سلول آن دارای حافظه اندکی می باشد. تمامی سلولها دارای ساختار یکسانی بوده و بصورت همزمان و موازی در گامهای گسسته از زمان کار یکسانی را انجام می دهند. در ادامه در بخش ۲ مفاهیم مطرح در مورد اتوماتای سلولی تشریح می شود و سپس در بخش ۳ الگوریتم پیشنهادی ما با ذکر جزییاتی نظیر نحوه سازماندهی سلولها، ساختار هر سلول و نحوه پایان محاسبات شرح داده می شود و در نهایت صحت درستی الگوریتم اثبات می شود.

۲- اتوماتای سلولی(CA)

یک شبکه منظم از ماشینهای با حالت محدود را در نظر بگیرید. به هر یک از این ماشینها، سلول می گوییم. هر کدام از این سلولها طبق الگوی ثابت و یکسانی با بعضی از سلولهای مجاور خویش در ارتباط هستند. این ارتباط محلی بوده و برای تمامی سلولها یکسان می باشد. خود سلول و تمام سلولهایی که با آن در ارتباط هستند، مجموعه همسایگان سلول را تشکیل می دهند. هر سلول در هرلحظه از زمان می تواند یکی از حالتهای موجود در مجموعه حالت را اختیار نماید. مجموعه حالت برای تمامی سلولها یکسان می باشد. در هر لحظه از زمان حالت تمامی سلولها بصورت همزمان و بر اساس یک قانون تغییر می کند. این قانون خود تابعی از حالتهای همسایگان سلول بوده و بنابراین در هر لحظه از زمان، حالت بعدی هر سلول به حالت فعلی تمامی همسایگانش بستگی دارد. این شبکه از یک پیکربندی اولیه شروع بکار کرده، در هر مرحله زمانی با اعمال قانون به تمامی سلولها پیکربندی بهنگام شده و با گذشت زمان شبکه یک رفتار پیچیده و جالب از خود تولید می نماید.

با توجه به توضیحات فوق، وجه تمایز CA نسبت به دیگر شبکه های اتوماتایی عبارت است از: ساده بودن ساختار، محلی بودن ارتباطات بین سلولها، برقراری الگوی ارتباطی یکسان برای تمامی سلولها، بهنگام سازی همزمان سلولها، بهنگام سازی سلولها توسط قانون یکسان و تولید رفتارهای جالب و پیچیده از سلولها و قوانین ساده. با توجه به این خصوصیات در اکثر مستندات از CA به عنوان ساختاری موازی، محلی و یکسان نام برده شده است که براحتی می تواند رفتارهایی را که دارای این سه ویژگی هستند، شبیه سازی نماید[6,13,19,22]. امروزه CA در زمینه های متعددی از جمله تولید الگوهای تصادفی، نظریه محاسبات، مدلسازی سیستم های فیزیکی و بیولوژیکی و محاسبات کاربردی مورد استفاده قرار گرفته است[5,7,13,17,19,20,22].

یک نمونه از CA هایی که امروزه متداول می باشد، CA دو بعدی است. در این ساختار سلولها در یک شبکه دو بعدی قرار داند. یکی از نکات مهم در این CA نوع همسایگی می باشد. دو نوع همسایگی مهم در این ساختار عبارتند از همسایگی von Neumann و همسایگی Moore. در همسایگی Moore برای هر سلول هشت سلول همسایه و در همسایگی von Neumann چهار سلول همسایه در نظر گرفته می شود. در هر دو همسایگی سلول قرار گرفته در مرکز همسایگی بهنگام می شود. Packard تمامی خصوصیات CA دوبعدی را بررسی نموده است[15].

تا کنون با استفاده از روشهای مختلفی اثبات شده است که CA یک مدل محاسباتی عمومی می باشد[3,4]. ولی در عمل از این مدل اغلب برای شبیه سازی پدیده های فیزیکی پویا استفاده شده و این مدل بیش از آنکه مورد توجه دانشمندان علوم کامپیوتر قرار گیرد، توجه پژوهشگران سایر علوم مانند فیزیک و بیولوژی را به خود مشغول نموده است. آنها تکامل تدریجی پدیده های فیزیکی را با استفاده از قوانین CA بصورت گام به گام شبیه سازی می نمایند. در صورتی که طبق تعریف یک مدل محاسباتی مکانیزمی برای توصیف تکامل تدریجی پدیده ها نمی باشد، بلکه ساختاری است که برای انجام محاسبات بر روی داده های ورودی بکار می رود.

در این مقاله به CA از دید یک ماشین محاسباتی عمومی نگریسته شده و الگوریتمی برای حل یک مساله توسط CA ارایه شده است. برای حل مسایل محاسباتی به یک ساختمان داده نیاز می باشد. این ساختمان داده شامل ورودی،خروجی و یک رویه برای تبدیل ورودی به خروجی می باشد. مراحل پردازش و تبدیل ورودی به خروجی در CA توسط انتقال حالتهای CA پیاده سازی می شود. هر چند در تعریف CA استاندارد، حافظه منظور نشده است ولی اگر CA خواسته باشد نقش یک ماشین محاسبه گر عمومی را بازی کند، هر سلول اتوماتا نیاز به تعدادی حافظه برای نگهداری مقادیر ورودی و خروجی داشته و همچنین اتوماتا باید قابلیت خواندن مقادیر ورودی و قابلیت نوشتن مقادیر خروجی (از-به) حافظه را داشته باشد. تعریف ارایه شده در زیر که در این مقاله بعنوان تعریف CA در نظر گرفته شده است، اصلاح شده تعریف اتوماتای میلی [7] می باشد. در این تعریف CA یک ۷ تایی بصورت $\{Q, d, v, \Sigma, \Delta, \delta, \lambda\}$ می باشد که در آن:

Q : مجموعه حالاتی است که هر سلول می تواند اختیار کند.

d : ابعاد فضای سلولی را مشخص می نماید. اگر d=2 باشد در این صورت یک CA دو بعدی خواهیم داشت.

V : برای هر سلول x در CA ، آرایه V مشخص کننده k+1 همسایه ای می باشد که بصورت مستقیم با سلول در ارتباطند.

Σ : الفبای ورودی CA می باشد.

Δ : الفبای خروجی CA می باشد.

δ: تابع انتقال است که بفرم $\delta : (Q \times \Sigma^n)^{k+1} \rightarrow Q$ می باشد. بر اساس تابع انتقال، حالت هر سلول به حالت و مقادیر حافظه ای ورودی تمامی همسایگان این سلول در مرحله فعلی بستگی دارد. n تعداد حافظه های ورودی و خروجی هر سلول می باشد.

λ : رابطه مبدل است که زیرمجموعه متناهی از $(Q \times \Sigma^n)^{k+1} \times \Delta^n$ می باشد. این مبدل مقدار هر حافظه خروجی سلول را با توجه به حالت و مقادیر حافظه ای ورودی همسایگانش مشخص می سازد. در این جا هر سلول CA در همان حافظه هایی می نویسد که از آنها می خواند و در نتیجه $\Sigma = \Delta$ می باشد.

۳- مرتب سازی بر روی CA دوبعدی و بصورت مارپیچی

در این قسمت، یک الگوریتم موازی برای مرتب نمودن اعداد در یک CA دو بعدی $n \times n$ ارایه می شود. در این CA سلولها در یک شبکه دو بعدی ردیف شده اند و نوع همسایگی بکار گرفته شده، همسایگی von Neumann می باشد. اگر سلول قرار گرفته در سطر i و ستون j

را با $c_{i,j}$ نمایش دهیم در این صورت همسایه سمت چپ $c_{i,j-1}$، همسایه سمت راست $c_{i,j+1}$، همسایه بالا $c_{i-1,j}$ و همسایه پایین $c_{i+1,j}$ می باشد. در این الگوریتم هر سلول می تواند مقدار خود را فقط با یکی از سلولهای همسایه خود جابجا نماید. هنگامی که بیش از یک سلول همسایه تقاضای جابجایی با مقدار یک سلول را داشته باشد، تصادم پیش می اید. در این حالت محدودیتی از نوع محدودیتهای سد کننده به هر سلول نسبت داده می شود و با توجه به آن تصمیم گرفته می شود که یک سلول با کدام همسایه خود عمل جابجایی را انجام دهد.

تمامی سلولها باید بصورت همزمان به کار خویش خاتمه دهندکه این خاتمه می تواند به سه روش انجام گیرد. در روش اول از یک کانال خارجی سراسری که به تمامی سلولهای CA متصل می باشد، استفاده می شود و هر سلول در صورتی که عمل جابجایی انجام دهد مقدار پرچم مخصوصی را در این کانال برابر یک قرار می دهد. با استفاده از این کانال سراسری ارتباطی CA هنگامی متوف می شود که تمامی سلولها به حالت مناسب خویش رسیده باشند یعنی هیچ گونه جابجایی انجام نگرفته باشد و مقدار پرچم برابر صفر باشد. بدیهی است که این ایده با قوانین حاکم بر CA همخوانی ندارد. زیرا در CA شبکه ارتباطی بصورت منظم و محلی بوده و هر سلول اجازه دارد که فقط با همسایگان خویش ارتباط برقرار نماید.

در روش دوم زمان خاتمه الگوریتم، مستقل از داده های ورودی می باشد. در این روش بدترین زمان اجرای الگوریتم را بدست آورده و سپس به سلولها اجازه داده می شود که به اندازه بدترین زمان با سلولهای همسایه عمل جابجایی را انجام داده و سپس متوقف شوند. واضح است که در مورد اکثر داده های ورودی زمان اجرای الگوریتم کمتر از بدترین حالت بوده و و داده ها زودتر مرتب می شوند، بنابراین سلولها مجبورند که مقایسه های تکراری و اضافی انجام دهندکه عیب این روش می باشد. در الگوریتم پیشنهادی بدترین حالت پیچیدگی الگوریتم از مرتبه تعداد پردازنده ها یعنی $o(n^2)$ بوده و در نتیجه این روش، روش مناسبی برای خاتمه دادن به الگوریتم نمی باشد.

روش سوم بر این اساس است که شرایط خاتمه دادن به الگوریتم در هر مرحله از الگوریتم تست شود. الگوریتم هنگامی خاتمه می یابد که هیچ سلولی در CA عمل جابجایی را انجام نداده باشد و اگر حتی یکی از سلولها عمل جابجایی را انجام دهد، این موضوع را به تمامی سلولها اطلاع می دهد و همگی سلولها به اجرای الگوریتم ادامه دهند. در این روش نحوه اطلاع دادن بصورت محلی و مبتنی بر خصوصیات حاکم بر CA می باشد. پیاده سازی این روش مستلزم اضافه نمودن عناصر حافظه ای به هر سلول می باشد که در در قسمت بعد به شرح آن خواهیم پرداخت. در این مقاله برای خاتمه دادن به الگوریتم از روش سوم استفاده شده است.

۳-۱- الگوریتم پیشنهادی

اساس کار الگوریتم پیشنهادی بر جابجایی های محلی استوار است یعنی هر سلول مقدار خویش را می تواند فقط با مقدار سلولهای همسایه اش مبادله نماید. به منظور انتخاب همسایه مطلوب و جلوگیری از تصادم، هر سلول نیاز به چند پرچم برای نگهداری محدودیتهای سد کننده دارد. برای هر همسایه به یک محدودیت سد کننده نیاز است و لذا هر سلول به چهار حافظه برای نگهداری محدودیتهای سد کننده نیاز دارد.

ابتدا محدودیتهای سد کننده با مقایسه مقادیر سلولهای همسایه محاسبه شده و سپس در صورتی که دو محدودیت سدکننده متناظر مقدار یک داشته باشند، مقادیر دو سلول با هم جابجا می شود.

برای انجام جابجایی های ستونی، ابتدا هر سلول با انجام مقایسه هایی بین مقدار خود و همسایه های بالا و پایینش، مقدار محدودیتهای سد کننده بالایی و پایینی خویش را محاسبه کرده و اگر مقدار محدودیت سدکننده بالایی سلول و مقدار محدودیت سد کننده پایینی همسایه بالایی سلول هر دو یک باشد، دو سلول مقدار خویش را باهم جابجا می نمایند. برای انجام جابجایی های سطری نیز، ابتدا هر سلول با انجام مقایسه هایی بین مقدار خود و مقدار همسایه چپ و راستش، مقدار محدودیتهای سد کننده چپ و راست خویش را محاسبه نموده و اگر مقدار محدودیت سد کننده چپ سلول و مقدار محدودیت سدکننده راست همسایه سمت چپ سلول هر دو یک باشند، دو سلول مقدار خویش را با یکدیگر جابجا می نمایند. روش جابجایی مشابه حالت ستونی می باشد. این جابجایی های سطری و ستونی تا آنجا ادامه می یابد تا مجموعه عناصر در CA بصورت مارپیچی مرتب شود.

بر اساس روش اندیس گذاری مارپیچی سطرهای فرد بصورت صعودی و سطرهای زوج بصورت نزولی مرتب می شوند و لذا نحوه محاسبه محاسبه محدودیتهای سد کننده سلولهای چپ و راست سلولهای سطرهای زوج و فرد با هم فرق می کند. بهمین دلیل هر سلول باید محل خود را در CA دانسته و با توجه به آن قانون مورد نظر را اجرا نماید. برای حل این مشکل از CA هیبرید استفاده می کنیم. در CA هیبرید چند نوع سلول می توان داشت. در این جا از دو دسته سلولهای odd و even استفاده شده است. سلولهای odd در سطرهای فرد و سلولهای even در سطرهای زوج قرار می گیرند.

برای محاسبه محدودیتهای سد کننده، هر سلول مقدار خود را با سلولهای همسایه مقایسه نموده و سپس قانون یکسانی را اجرا می نماید بهمبن دلیل سلولهای مرزی به علت نداشتن یکی از همسایه ها با مشکل مواجه می شوند. برای برطرف کردن این مشکل در چهار طرف CA از تعدادی سلول اضافی استفاده می شود. در صورتی که CA دارای n^2 سلول باشد به 4n سلول اضافی نیاز است. برای اینکه در اجرای الگوریتم خللی پیش نیاید مقدار هر یک از سلولهای اضافی برابر یک عدد بسیار بزرگ و مقادیر تمام محدودیتهای سدکننده آنها برابر صفر

در نظر گرفته می شود. در این مقاله هر جا صحبت از سلول به میان می آید منظور سلولهای اصلی می باشد مگر آنکه واژه سلولهای اضافی بصورت صریح قید شده باشد.

برای خاتمه دادن به الگوریتم از روش سوم که قبلا به آن اشاره شد، استفاده می شود. یعنی الگوریتم هنگامی خاتمه می یابد که هیچ سلولی عمل جابجایی انجام ندهد. در صورتی که یک سلول مقدار خود را با مقدار سلول دیگری جابجا نماید، باید این عمل را به تمام سلولها اطلاع دهد تا از این طریق تمام سلولها حتی آنهایی که عمل جابجایی انجام نداده اند به کار خود ادامه دهند. هر سلول دارای دو وظیفه اساسی می باشد، یکی انجام عمل جابجایی و دیگری اطلاع عمل جابجایی به تمامی سلولها به منظور ادامه دادن به اجرای الگوریتم. پس الگوریتم دارای فاز مقایسه و جابجایی و فاز تست خاتمه الگوریتم می باشد که فاز مقایسه و جابجایی شامل انجام یک مرحله جابجایی های سطری و ستونی بین سلولها بوده و فاز تست خاتمه الگوریتم شامل اطلاع دادن عمل جابجایی به تمامی سلولها و تصمیم گیری در مورد ادامه الگوریتم می باشد. بعد از انجام هر فاز مقایسه و جابجایی، الگوریتم وارد فاز تست خاتمه الگوریتم می شود.

به منظور پیاده سازی فاز تست برای هر سلول یک حافظه بنام continue در نظر گرفته شده است.در هر فاز مقایسه و جابجایی اگر سلولی جابجایی انجام دهد، مقدار حافظه continue خود را برابر یک قرار می دهد و در فاز تست خاتمه الگوریتم سلول این مقدار را به حافظه continue سلولهای دیگر انتقال می دهد. در مرحله اول از فاز تست خاتمه، این مقدار یک به حافظه continue چهار همسایه سلول منتقل شده و در مرحله دوم هر یک از این همسایه ها این مقدار را به همسایه های دیگر خود منتقل نموده و بدین ترتیب در مراحل متوالی زمانی، این مقدار بصورت حرکت گسترشی لوزی شکل به تمامی سلولهای CA انتقال می یابد. این کار بوسیله OR نمودن مقادیر continue همسایه ها با یکدیگر انجام می گیرد.

بدترین حالت برای عمل گسترش هنگامی اتفاق می افتد که در یکی از گوشه های CA عمل جابجایی رخ داده باشد.در این صورت اگر تعداد سطرهای CA برابر n باشد، به اندازه 2n-3 مرحله عمل گسترش باید انجام گیرد تا مقدار continue این سلول به تمامی سلولها و در نهایت به سلول قرار گرفته شده در گوشه مقابل منتقل شود. پس از اتمام عمل گسترش مقدار continue تمامی سلولها با هم برابر است. اگر مقدار continue سلولها برابر صفر باشد، سلولها متوقف شده و در غیر این صورت دوباره وارد فاز جابجایی می شوند. در ساده ترین حالت الگوریتم در هر فاز مقایسه و جابجایی فقط یک مرحله عملیات جابجایی انجام گرفته وسپس الگوریتم وارد فاز تست خاتمه می شود. از آنجا که برای مرتب شدن داده های ورودی به حداقل n مرحله عمل جابجایی نیاز می باشد، بنابراین فاز تست خاتمه الگوریتم n بار اجرا شده و با توجه به اینکه زمان مورد نیاز برای تست خاتمه الگوریتم از مرتبه n می باشد الگوریتم دارای پیچیدگی از مرتبه $o(n^2)$ خواهد بود.

به منظور کاراتر نمودن الگوریتم ایده ذیل پیشنهاد می شود. چون فاز تست خاتمه الگوریتم 2n-3 مرحله طول می کشد، پس اگر به ازای هر 2n-3 فاز مقایسه و جابجایی یکبار خاتمه الگوریتم تست شود آنگاه فاز مقایسه و جابجایی با فاز تست خاتمه الگوریتم همزمان شده و زمان اجرای الگوریتم کاهش می یابد. به عنوان مثال پس از 2n-3 فاز مقایسه و جابجایی الگوریتم وارد فاز تست خاتمه الگوریتم شده که این فاز نیز 2n-3 مرحله طول می کشد. سپس اگر CA مرتب شده باشد الگوریتم خاتمه می یابد، در غیر این صورت دوباره وارد فاز جابجایی شده و روند فوق تا هنگام مرتب شدن CA ادامه می یابد. برای پیاده سازی این ایده لازم است تا هر سلول یک شمارنده داشته باشد و علاوه بر آن تعداد سطرها را در یکی حافظه های خود حفظ کند، بدین منظور از رجیسترهای counter و size برای هر سلول CA استفاده می شود.

در این الگوریتم با توجه به سلولهای اضافی بکار برده شده، هر دسته از سلولها برنامه یکسانی را تا نیل به هدف نهایی اجرا می نمایند. فرض بر این است که در هنگام شروع اجرای الگوریتم، عناصر ورودی در حافظه x سلولها، تعداد سطرها در حافظه size سلولها و مقدارصفر در حافظه counter و continue سلولها بار شده است. برای اجرای الگوریتم هر سلول CA دوبعدی دارای یک کلید و چهار رجیستر برای نگهداری محدودیتهای سد کننده و سه رجیستر برای فاز تست خاتمه الگوریتم می باشد. الگوریتم هنگامی خاتمه می یابد که پس از فاز تست مقدار continue تمامی سلولها برابر صفر باشد.

۳-۲-تجزیه و تحلیل الگوریتم

در این بخش صحت الگوریتم اثبات می شود. برای نشان دادن صحت الگوریتم قضیه زیر بیان می شود.

قضیه ۱ : در یک CA دوبعدی در صورتی که سطرهای فرد صعودی، سطرهای زوج نزولی و ستونها صعودی مرتب باشند، CA بصورت مارپیچی مرتب است.

برهان: طبق فرض سطرها بصورت متناوب از چپ به راست و از راست به چپ مرتبند، بنابراین برای اثبات قضیه کافی است ثابت کنیم تمام عناصر موجود در سطر i ام کوچکتر از تمام عناصر موجود در سطر i+1 می باشند. ابتدا فرض کنید سطر i ام بصورت صعودی مرتب است و سطر i+1 ام بصورت نزولی مرتب می باشد. اگر ثابت کنیم کوچکترین عضو سطر i+1 ام از بزرگترین عضو سطر i ام بزرگتر باشد می توان گفت که تمام عناصر سطر i ام از تمام عناصر سطر i+1 ام کوچکتر است. اثبات این امر نیز ساده است زیرا کوچکترین عضو سطر i+1 در سمت راستترین ستون قرار دارد و بزرگترین عضو سطر i ام نیز در سمت راستترین ستون قرار دارد و از انجا که ستونها طبق فرض بصورت صعودی مرتب هستند بنابراین کوچکترین عضو سطر i+1 ام از

بزرگترین سطر i ام بزرگتر است. اگر سطر i ام بصورت نزولی مرتب باشد به دلیل مشابه همین امر اثبات می شود. از انجا که سطرها بطور متناوب از چپ به راست و از راست به چپ مرتب می باشند و خود ستونها نیز از بالا به پایین بصورت صعودی مرتب است، بنابراین کل CA بصورت صعودی مرتب است.

۴- نتیجه گیری

در این مقاله یک الگوریتم موازی برای مرتب نمودن اعداد در یک اتوماتای سلولی دو بعدی ارایه شده است. ایده الگوریتم بسیار ساده و بر اساس جابجایی های محلی بین سلولهای همسایه می باشد.

۵- مراجع

[1] Akl, S. G. *Parallel Sorting algorithms.* Orlando, FL: Academic, 1985.

[2] Batcher, K. E. Sorting Network and Their Application. *AFIP Proc*, 32, 1968, 307-314 .

[3] Burks, W. *Essays on Cellular Automata.* Urbana, IL: University of Illinois Press, 1970.

[4] Conway, G. H., Berlekamp, E., and Guy, R. *Winning Ways for Your Mathematical Plays.* vol. 2, Academic Press, 1982.

[5] Culik, K., Hurd, L., and Yu, S. Computation Theoritic Aspects of Cellular Automata. *Physica D*, 45, 1990, 357-378.

[6] Farmer, D., Toffoli, T., and Wolfram, S. *Cellular Automata Proceeding of an Interdisciplinary Workshop.* Amsterdam, North Holland, 1984.

[7] Gordillo, L., and Luna,V. Parallel Sort on Linear Array of Cellular Automata. *IEEE Transaction on Computers*, 1994, 1904-1910.

[8] Gutowitz, A. H. *Cellular Automata.* Cambridge, MA: MIT Press, 1990.

[9] Knuth, D. E. *The Art of Computer Programming*. vol. 3, Addison-Wesley, Reading, MA, 1973.

[10] Kummar, M., and Hirschberg, D. S. An Efficient Implementation of Batcher's Odd-Even Merge Algorithms and its Application to Parallel Sorting Schemes. *IEEE Transaction on Computer,* 32, 1983, 254-264.

[11] Lakshmivarahan, S., Dhall, S. K, and Miller, L. L. Parallel Sorting Algorithms. *M. C. Yovitts, ed., Advances in Computers, Academic Press, New York*, 23, 1984, 295-354.

[12]Megson, G. M. *An Introduction to Systolic Algorithm Design.* Claradon Press Oxford, 1992.

[13] Mitchel, M. *Computation in Cellular Automata : A Selected Review.* Technical Report, Santa Fe Institute, Santa Fe, New Mexixo, 1996.

[14] Orcutt, S. *Computer Organization and Algorithms for Very High Speed Computation.* Ph. D. Thesis, Stanford University, 1984.

[15] Packard, N. Two Dimensional Cellular Automata. *Journal of Statistical Physics,* 30, 1985, 901-942.

[16] Quinn, M. J. *Parallel Computing: Theory and Practice.* McGrow-Hill, Ing, 1994.

[17] Sarkar, P. Brief History of Cellular Automata. *ACM Computing Surveys,* 32, 1, 2000;

[18] Thompson, D., and Kung, H. T. Sorting on a Mesh Connected Parallel Computer. *Communication ACM,* 20, 1977, 263-271.

[19] Toffoli, T., and Margolus, N. *Cellular Automata Machines: A New Environment for Modeling.* Cambridge, MA:MIT Press,1987.

[20] Wolfram, S. Statistical Mechanics of Cellular Automata. *Review of Modern Physics,* 55, 1983, 601-644.

[21] Wolfram, S. Computation Theory of Cellular Automata. *Communication in Mathematical Physics,* 96, 1984, 15-57.

[22] Wolfram, S. *Theory and Application of Cellular Automata.* Singapore, World Scientific, 1986.