

سیستم های چند پردازنده (Multi Processor Systems)

مهدی حریری
mehdi_hariri@yahoo.com

علیرضا صدیقیان
ali1984_1@yahoo.com

دانشگاه آزاد اسلامی واحد زنجان، دانشکده مهندسی کامپیوتر
تلفن: ۴۲۵۰۰۱۱

چکیده: سیستم های چند پردازنده سیستمهایی با بیش از یک پردازنده هستند که امروزه با توجه به تنزل قیمت سخت افزار استفاده از آنها به شکل صعودی مرسوم تر می شود. از آنجایی که یک سری دستورات خاص در ارگان های خاص بایستی به شکل بلادرنگ به اجرا درآیند بنابراین استفاده از این سیستم ها پررنگ تر می شود. تا کنون دو ساختار کلی برای این سیستم ها پیشنهاد شده است. که هر کدام اگر چه قابل پیاده سازی است ولی نقص هایی نیز دارد. از آنجایی که توپولوژی گراف بهترین توپولوژی برای این سیستم ها است. مادر این مقاله با استفاده از الگوریتم های مسیریابی و هماهنگ سازی نودها در گراف با ترکیب این دو ساختار به یک ساختار جدید در طراحی رسیده ایم.

کلمات کلیدی: چند پردازنده ها، حافظه توزیع شده، حافظه مشترک، واحد کنترل کننده بایس و الگوریتم دایکسترا.

۱- مقدمه

در این مقاله سیستم های چند پردازنده مورد بررسی قرار می گیرند. از آنجایی که این سیستم ها روز به روز به کاربردهایشان افزوده می شود بنابراین بایستی به شکل دقیقی مورد تحلیل قرار گیرند. در بخش نخست این مقاله به اجمال به تحلیل این سیستم ها می پردازیم در بخش دوم مشخصات سیستمهای چند پردازنده ای را مورد بررسی قرار می دهیم، در بخش سوم به چگونگی ارتباط بین پردازنده ها اشاره میکنیم، در بخش چهارم ایده جدیدی را که در مورد ارتباط بین پردازنده ها ارائه داده ایم بیان می کنیم و در بخش ۵ درباره دآوری و حکمیت بین درخواست های پردازنده ها مطالبی را بیان می کنیم.

۲- مشخصات سیستم های چند پردازنده ای (Multi Processors)

یک سیستم چند پردازنده دارای دو یا چند پردازنده با حافظه و دستگاههای ورودی خروجی است. کلمه پردازنده در سیستم های چند پردازنده ای ممکن است به CPU یا پردازنده های ورودی خروجی اطلاق شود. چند پردازنده ها به عنوان سیستم های دارای چندین رشته دستورات و داده ها و با معماری طبقه بندی MIMD (Multi instruction multi data) طبقه بندی می شوند. بدلیل ارزانی و اندازه کوچک ریز پردازنده ها و امکان اتصال چند صد ریز پردازنده در یک سیستم بزرگ، امروزه طراحی سیستم های چند پردازنده ای با چندین

ریز پردازنده با کارایی بالا فراهم شده است. سیستم های چند پردازنده بر حسب تشکیلات حافظه نیز دسته بندی می شوند :

دسته اول، سیستم های چند پردازنده با حافظه مشترک (Tightly coupled) می باشند که در آنها یک حافظه مشترک برای کل سیستم وجود دارد و اطلاعات حافظه می تواند بطور مشترک توسط پردازنده ها استفاده شود. البته در این حالت پردازنده می تواند حافظه محلی خود را نیز داشته باشد و در حالت کلی در این سیستم ها هر پردازنده دارای حافظه نهان مختص خود نیز می باشد.

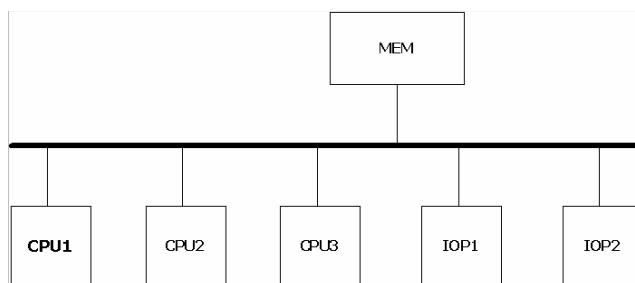
دسته دوم، سیستم های چند پردازنده با حافظه توزیع شده (Loosely coupled) می باشد. در این سیستم ها پروسورها از طریق سیستم های سوییچینگ اطلاعات را بین خود مبادله می کنند. ارسال اطلاعات از پردازنده ای به پردازنده دیگر به صورت بسته اطلاعات است که بسته اطلاعات (Information Packet) شامل آدرس پروسور خود اطلاعات و مقدار کدهای خطایابی است. بسته اطلاعات بستگی به نوع سیستم مخابراتی به اولین پردازنده نزدیک و یا پردازنده بخصوص می رود. این سیستم موقعی استفاده می شود که تبادل اطلاعات بین پردازنده ها کم باشد. در غیر این صورت سیستم های چند پردازنده ای با حافظه مشترک استفاده می شوند.

۳- چگونگی ارتباط و اتصال پردازنده ها

ارتباط بین قسمت های مختلف در سیستم های چند پردازنده ای تابع تعداد پردازنده ها و نوع ارتباطات آنها ... می باشد. و اصولاً راههای مختلفی برای ایجاد یک شبکه ارتباطی بین سیستم های چند پردازنده ای وجود دارد که بعضی او آنها عبارتند از :

۳-۱- گذرگاه مشترک با اشتراک زمانی (Time Shared Common Bus)

در این سیستم یک تعداد پردازنده از طریق یک گذرگاه مشترک با حافظه ارتباط برقرار می کنند. شکل (۱) این ساختار را نشان می دهد. که از سه پردازنده مرکزی و دو پردازنده ورودی خروجی با یک حافظه مشترک تشکیل شده است. در این سیستم در هر لحظه فقط یک پردازنده می تواند با حافظه مشترک یا پردازنده دیگر ارتباط برقرار کند. در این سیستم موقعی که پردازنده بخواهد ارتباطی با سایر قسمتهای سیستم داشته باشد ابتدا بایستی وضعیت گذرگاه را بررسی کند.



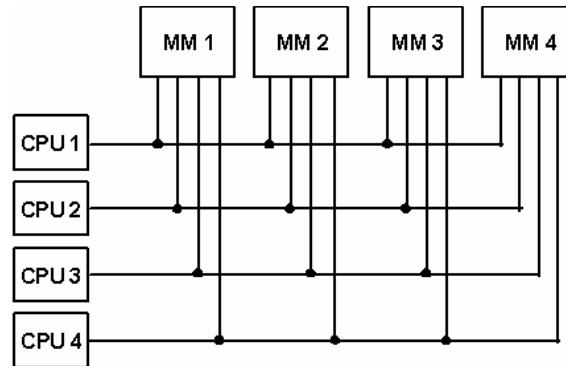
شکل (۱) گذرگاه مشترک با اشتراک زمانی

در صورتیکه گذرگاه آزاد بود پردازنده می تواند آدرس پردازنده مقصد را برای شروع انتقال اطلاعات ارسال نماید. پردازنده دریافت کننده آدرس نیز آدرس خود را تشخیص می دهد و جوابی به ارسال کننده آدرس مبنی بر آماده بودن جهت مبادله اطلاعات ارسال می کند.

در این سیستم ها چون یک باس بین همه پردازنده ها مشترک است اگر دو یا چند پروسسور بخواهند همزمان با هم ارتباط برقرار کنند ممکن است حین نقل و انتقال اطلاعات پردازنده ها برخورد پیش بیاید. برای حل مشکل یک واحد کنترل کننده گذرگاه در سیستم قرار داده می شود که گذرگاه را کنترل کرده و اولویت لازم را بین واحدهای مختلف برقرار کند.

۳-۲- سیستم حافظه های چندپورتی یا چنددرگاهی (Multiport Memory)

در سیستم حافظه های چندپورتی یک باس مجزا بین هر واحد حافظه و CPU وجود دارد.

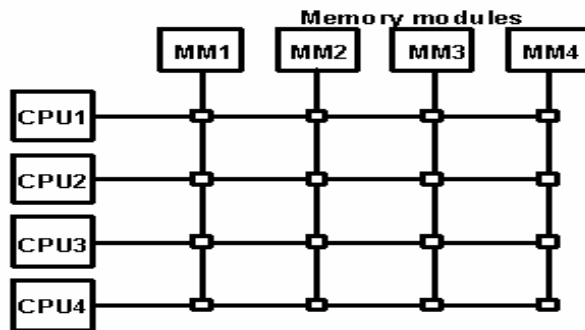


شکل ۲) سیستم حافظه های چند پورتی

در شکل فوق هر واحد حافظه دارای چهار پورت یا چهار درگاه است که هر پورت با یک باس ارتباط دارد. بدیهی است که در این حالت هر واحد حافظه می بایستی دارای مدارهای منطقی لازم باشد بطوریکه در هر لحظه یکی از پورتها را با حافظه مرتبط سازد. برای جلوگیری از برخورد و تداخل اطلاعات در موقع دسترسی به حافظه هر واحد حافظه برای پورت های خود دارای اولویت می باشد. این سیستم به علت داشتن گذرگاه های زیاد بین حافظه و پردازنده دارای نرخ تبادل اطلاعات بالایی می باشد ولی اشکال آن این است که نیاز به مدار کنترل گران قیمتی برای حافظه و همچنین مقدار زیادی کابل و اتصالات جهت ارتباطات دارد. بنابراین این ساختار حافظه چند پورتی در سیستم های چند پردازنده که دارای تعداد کمی پردازنده باشد قابل استفاده است.

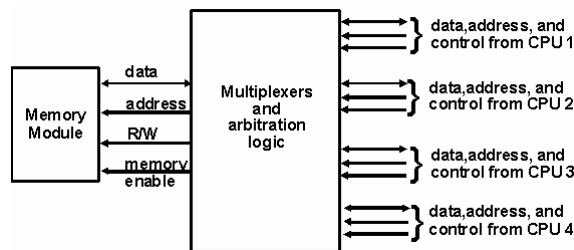
۳-۳- سویچ های متقاطع یا سویچ کرس بار (Crossbar Switches)

این سیستم چند پردازنده ای دارای یک سری سویچ هایی در محل تقاطع گذرگاههای پردازنده ها و گذرگاههای حافظه ها می باشد. در شکل ۳ نمونه ای از این سیستم ها را مشاهده می کنید.



شکل ۳) سیستم سوئیچ های متقاطع

در شکل ۳ مربع کوچک در محل تقاطع باس ها یک سوئیچ می باشد که ارتباط یک پردازنده با یک واحد حافظه را برقرار می کند. هر سوئیچ دارای مدارهای منطقی است که بتواند این ارتباط را برقرار نماید به این ترتیب که سوئیچ آدرسی را که بر روی گذرگاه گذارده شده است آزمایش می نماید که ببیند آیا آدرس واحد حافظه متصل شده به آن است یا نه. اگر آدرس مربوط به آن واحد حافظه بود ارتباط را برقرار می نماید. علاوه بر این چون این سوئیچ دارای چندین ورودی از پردازنده های مختلف است بنابراین یک اولویت بندی از قبل برای سوئیچ ها تعیین شده است. به طوریکه اگر دو پردازنده همزمان تقاضای ارتباط با یک واحد حافظه را بنمایند ابتدا ارتباط را به پردازنده ای که دارای اولویت بالاتری است می دهد و سپس برای پردازنده دوم ارتباط برقرار می کند. شکل ۴ ساختار ارتباطی یک سوئیچ متقاطع، را که که چهار cpu را می تواند به یک واحد حافظه ارتباط دهد، نشان می دهد.



شکل ۴) شمای یک سوئیچ متقاطع

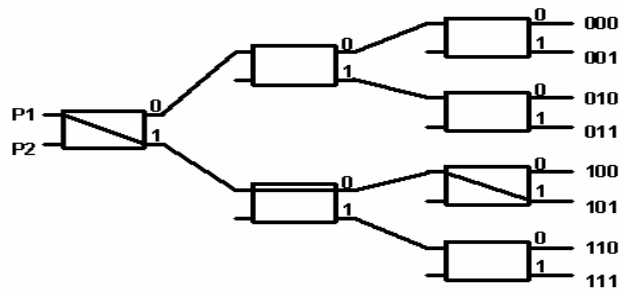
۳-۴- سوئیچ چند طبقه ای (Multistage switch Network)

سوئیچ چند طبقه سوئیچی است که هر واحد بنیادی آن دارای دو ورودی و دو خروجی می باشد. علاوه بر این سوئیچ دارای قابلیت تصمیم گیری برای جلوگیری برخورد اطلاعات است یعنی اگر هر دو ورودی A, B در یک زمان بخواهند به یکی از خروجی ها متصل شوند یکی از آنها را به خروجی متصل خواهد نمود. در شکل ۵ یک سوئیچ ۲*۲ نمایش داده شده است.



شکل ۵) طرز کار سوئیچ تبدیلی ۲*۲

با بکار بردن سویچ 2×2 می توان یک شبکه سویچ چندطبقه ساخت بطوریکه بتوان ارتباط بین یک تعداد از واحدهای منبع و مقصد را برقرار و کنترل نمود. برای روشن شدن مطلب درخت باینری زیر را در نظر بگیرید:

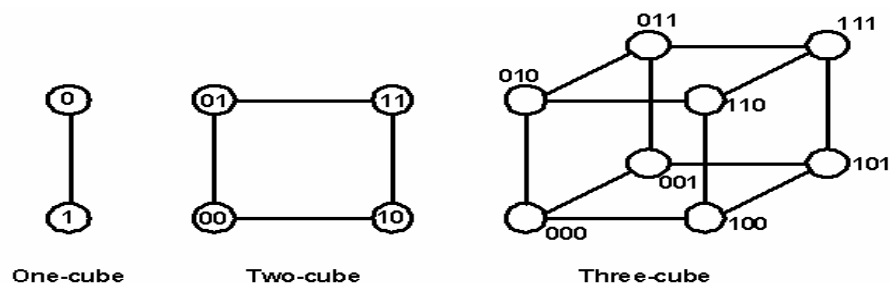


شکل ۶ درخت باینری با سویچ های 2×2

در این ساختار هر یک از پروسسورها $P1, P2$ می توانند از طریق سویچهای مذکور با ۸ واحد حافظه به شماره های باینری ۰۰۰ تا ۱۱۱ ارتباط برقرار کنند. اصولاً مسیر بین یک منبع به یک مقصد از روی شماره باینری مسیر مقصد تعیین می شود.

۳-۵- اتصال فوق مکعبی (Hypercube System)

در سیستم های چند پردازنده ای که اتصال آنها به صورت فوق مکعبی است هر پردازنده به صورت گره ای از مکعب است. البته در اینجا منظور از پردازنده فقط CPU نیست بلکه حافظه محلی و وسایل ورودی و خروجی را نیز می باشد. در ساختار فوق مکعبی N بعدی 2^N پردازنده می توانند در گوشه های مکعب قرار گیرند و هر پردازنده دارای آدرسی برابر با یکی از 2^N ترکیب باینری است. در این روش هر پروسسور با پروسسور همسایه خود در ارتباط است. شکل ۷ آرایش های مختلف این اتصال را نشان می دهد.



شکل ۷ ساختار فوق مکعبی برای $n=1,2,3$

نمونه هایی از این ساختار در کامپیوترهای اینتل مورد استفاده قرار گرفته است.

۴- ایده پیشنهاد شده در ارتباط و اتصال پردازنده ها

۴-۱- الگوریتم دایکسترا (Dijkstra):

این الگوریتم کوتاهترین مسیر را در یک گراف وزن دار ، با شروع از یک گره خاص تا گره های دیگر مشخص می کند. وزن مربوط به یالها می تواند هر پارامتر قراردادی باشد. که در سیستم های چند پردازنده ، این وزن می تواند عدم مشغول بودن پردازنده های مسیر در نظر گرفته شود. به این صورت که اگر در مسیر که می پیماییم به پردازنده ای برسیم که در حال پردازش باشد (بیکار نباشد) این مسیر بینهایت تصور می شود ، پس قبل از رسیدن به آن مسیر را عوض میکنیم یا فرض می کنیم که یال مربوط به آن پردازنده در گراف وجود ندارد. پیچیدگی زمانی الگوریتم دایکسترا برابر $O(n^2)$ می باشد که n تعداد گره های گراف در نظر گرفته می شود. n برای سیستم های چند پردازنده می تواند تنها کسری از تعداد واقعی پردازنده ها باشد.

$$(۲) \quad \text{تعداد ابعاد گراف} / \text{تعداد واقعی پردازنده ها} = n$$

۴-۲- طرح پیشنهادی (ترکیب حافظه مشترک و حافظه توزیع شده)

همان گونه که اشاره شد سیستم های چند پردازنده از لحاظ دسترسی به حافظه به دو دسته کلی تقسیم می شوند:

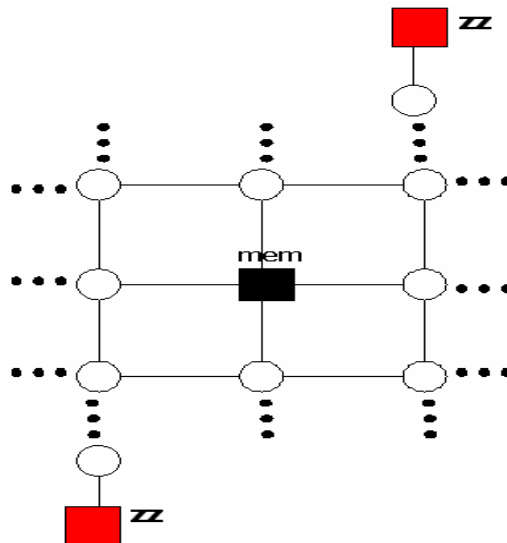
(۱) حافظه مشترک Tightly Coupled

(۲) حافظه توزیع شده Loosely Couple

اما هر دو دسته فوق دارای یک سری معایب کلی می باشند. به عنوان مثال در دسته اول پروسسورها نمی توانند همزمان به حافظه دسترسی داشته باشند. همچنین در دسته دوم اگر بخواهیم داده ای را در اختیار تمامی پردازنده ها قرار دهیم بایستی در تمامی حافظه های محلی مربوطه بنویسیم . که این کار بسیار زمان بر و هزینه بر است . در این مقاله با ترکیب دو دسته ، یک دسته میانی ایجاد نموده ایم که ضمن کم رنگ کردن معایب دو دسته ، مزایای دو دسته را یک جا جمع نماییم. شکل ۸ تا حدی گویای این مطلب می باشد. در این شکل هر کدام از دایره ها که نمایشگر پردازنده می باشند ، دارای یک حافظه محلی می باشند .

همچنین برای هر $(2i+1)*(2i+1)$ پردازنده یک حافظه مرکزی مشترک در نظر گرفته ایم . (ا بستگی مستقیم به تعداد پردازنده ها دارد). علاوه بر آن در دو سوی هر بعد یک سری پردازنده های خاصی تعبیه شده که نقش مهمی را ایفا می کنند. هر گاه ما بخواهیم داده ای را در بین پردازنده ها به اشتراک بگذاریم ، این داده را در حافظه ZZ قرار میدهیم و پردازنده خاص آن را با استفاده از الگوریتم دایکسترا (*dijkstra*) در حافظه های مرکزی قرار می دهد و پردازنده های کناری حافظه ، با مراجعه به آن به داده مورد نظر دسترسی پیدا می کنند. به این ترتیب پردازنده ها علاوه بر داشتن حافظه محلی خود ، دارای حافظه مشترک نیز خواهند بود.

$$(۱) \quad \begin{matrix} (2i+1)*(2i+1) \\ (i=1,2,3,...) \end{matrix}$$

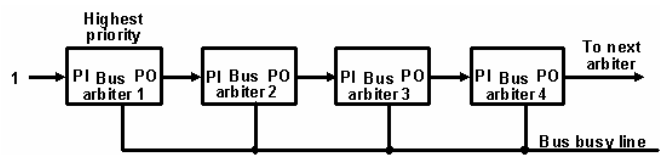


شکل ۸) نمایش ترکیبی حافظه مشترک و حافظه توزیع شده

نکته دیگری که در مورد این ساختار قابل ذکر می باشد این است که ما می توانیم برای بعد های مختلف از آ های متمایز استفاده کنیم که این قابلیت می تواند نرخ انعطاف پذیری ساختار را تا حد بالایی ، افزایش دهد.

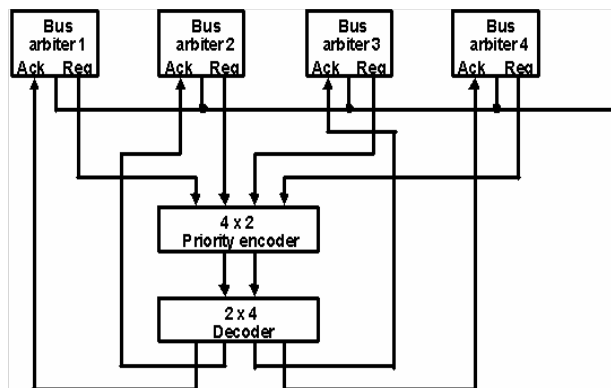
۵) حکمیت و داوری

در حکمیت و داوری ایستا بین درخواست های پردازنده ها بایستی اولییتی بین آنها در نظر گرفته شود. این داوری می تواند به دو حالت کلی سری و موازی انجام گیرد. در حالت سری ، اولییتی پردازنده ای که به گذرگاه سیستم متصل می شود ، بر طبق موقعیت آن در سیستم زنجیره ای است. یعنی اولین دستگاه سمت چپ دارای بالاترین اولویت و به ترتیب دست راستی آنها ، دارای اولویت کمتر می باشد. موقعی که چند دستگاه با هم تقاضای در اختیار گرفتن گذرگاه را می کنند ، گذرگاه سیستم در اختیار دستگاهی که اولویت بالاتری است قرار می گیرد. در شکل ۹ حالت کلی حکمیت و داوری ایستای سری نمایش داده می شود. که اولویت پراسسورها نقش مهمی را در پاسخ دهی به درخواست شان ایفا می کند



شکل ۹) داوری و انتخاب باس موازی

در تکنیک داوری انتخاب گذرگاه به صورت موازی یک انکدر با اولویت و یک دیکدر مطابق شکل ۱۰ بکار می رود. در این روش ، هر واحد داوری و انتخاب گذرگاه دارای یک خروجی تقاضا کننده (Req) گذرگاه و یک خط ورودی قبولی باس Ack می باشد. هر موقع که پردازنده ای بخواهد گذرگاه سیستم را در اختیار بگیرد ، مدار داوری و انتخاب گذرگاه آن خط خروجی تقاضای گذرگاه را فعال می کند و در صورتی که قبولی باس را دریافت کرد ، کنترل گذرگاه را در اختیار می گیرد.



شکل ۱۰ داوری و انتخاب سری

۵) نتیجه گیری

با توجه به ساختار ارایه شده در این مقاله ، و همچنین ماهیت بسیار نزدیک نخ ها (threads) در پردازشهای درون (process) سیستم عامل این ساختار پیشنهادی برای الگوریتم های همروند و چند نخی (multi thread) می تواند بسیار کارساز و بهینه باشد.و همچنین می توان به این نکته اشاره کرد که در این ساختار می توان انسجام داده ها در cache را تا حد بسیار بالایی ارتقا داد.که می تواند به عنوان تحقیق در آینده مورد نظر قرار گیرد. همچنین در این ساختار این امکان نیز وجود دارد که می توان در هر بعد ، پروسسور ها را با هم همزمان (synchron) کرد.

۶) سپاسگذاری

لازم میدانیم از دست اندر کاران این کنفرانس که برای تشکیل این مجمع علمی و گردآوری ایده های جدید زحمات بسیاری را متقبل شده اند تشکر و سپاسگزاری داشته باشیم .امیدواریم که این مجامع باعث بالا رفتن سطح علمی کشورمان گردد.

۷) مراجع

- [1] Patterson,David A,Hennesy,John L,"computer organization and disign",Morgan kaufman publishers,inc,1994
- [2] Prof.jerry Breecher,"Multiprocessors Shared Memory Architectures",CSCI 240,FALL 2003
- [3] Robert D.Blumofe ,"Thread Scheduling FOR Multiprogrammed Multiprocessors",