تمرین سری چهارم

تینا صداقت ۹۳۳۱۰۴۴

۱. یک سیستم اجزای متفاوتی دارد. اجزای تشکیل دهنده یک سیستم– به‌طور نمونه -عبارت‌اند از یک یا تعدادی CPU، GPU، RAM ، HDD، SDD و. ... این اجزاء در نسل های مختلف معماری به شیوه های متفاوتی به یکدیگر متصل می شوند که در نهایت کارایی سیستم را تعیین می کنند. معماری چند نسل از سیستمهای کامپیوتری در شکلهای 1 و 2 نشان داده شده است .لطفاً از کتاب [cook12] از فصل3 ، بخش PC Architecture را مطالعه کنید .سپس با استدلال به سؤالات خواسته شده پاسخ دهید.

- سرعت اتصال حافظه های جانبی در دو معماری چه تفاوتی دارد؟

توجه داریم که همه‌ی دستگاه‌های GPU از طریق باس PCI-E به پردازنده وصل می‌شوند. فرض می‌کنیم این گذرگاه از نوع PCI-E 2.0 است که در حال حاضر سریع‌ترین باس موجود با نرخ انتقال اطلاعاتی ۵ گیگابایت بر ثانیه می‌باشد. PCI-E 3.0 نیز امروزه معرفی شده است و مقدار پهنای باند ممکن را به طور قابل توجهی افزایش داده است.

اگرچه برای دریافت داده از پردازنده، باید از طریق دستگاه northbridge و گذرگاه کم سرعت FSB (front-side bus) عبور کنیم. FSB می تواند تا نرخ کلاک ۱۶۰۰ مگاهرتز را اجرا کند اگرچه در بعضی طراحی ها به مراتب کمتر است. این فرکانس تنها یک سوم فرکانس پردازنده سریع است.

حافظه از طریق northbridge و دستگاه های جانبی از طریق northbridge chipset و southbridge chipset به پردازنده دسترسی پیدا می‌کنند. Northbridge با همه اجزای پرسرعت مثل حافظه،CPU و اتصالات باس PCI-E ارتباط برقرار می‌کند.South bridge با اجزای کم‌سرعت مثل hard disk، USB، keyboard و network connection ارتباط برقرار می‌کند. همچنین می‌توان کنترل کننده هارد دیسک را به اتصالات PCI-E وصل کرد که باعث دسترسی پرسرعت داده از طریق این سیستم می‌شود.

با ظهور معماری Nehalem، تغییرات جدید مشاهده شد. مهمترین این تغییرات، قرار دادن چیپ X58 به جای northbridge و southbridge است. این معماری دارای Quick path interconnect یا QPI است که خیلی بهتر از FSB است و کمی شبیه به AMD’s hypertransport است. اتصال QBI بسیار سریع است که می‌تواند برای ارتباط دستگاه‌های جانبی با CPU به کار رود. در سیستم Nahelem معمولی، به subsystem ها وصل می‌شود و از بین X58 ، به PCI-E متصل می‌شود. QPI در پردازنده های مدل extreme/Xeon سرعتی به اندازه ۴,۸ یا ۶,۴

GT/s دارد. در طراحی اینتل، سطوح پهنای باند مشابهی در سیستم وجود دارد. اینتل از ۳ یا ۴ کانال حافظه در بالای سیستم و از ۲ کانال حافظه در پایین سیستم استفاده می‌کند.
بعد از طراحی X58 chipset، اینتل طراحی sandy bridge را آغاز کرد. یکی از مهمترین پیشرفت‌ها، پشتیبانی از استاندارد SATA-3 بود که نرخ انتقال ۶۰۰ Mb/s را دارد که با SSD ها ترکیب می‌شود و کارایی IO برای load و ذخیره داده‌ها را بالا می‌برد.
پیشرفت دیگر در sandy bridge، معرفی دستورالعمل AVX (Advanced Vector Extensions) بود که پردازنده‌های AMD هم آن را پشتیبانی می‌کنند. AVX برای دستورات وکتوری با دقت دوبل چهار تایی (four double precision) یعنی ۲۵۶bit /۳۲byte عملکرد وکتوری را دارد. این توسعه ارزشمندی است و باعث تسریع در محاسبه‌های CPU شد.
AMD از یک کانال دو گانه حافظه استفاده می‌کند که به طور قابل توجهی باعث کاهش پهنای باند در دسترس برای اتصال host-memory در CPU می‌شود.

- آیا سرعت اتصال حافظه های جانبی اهمیت دارد؟ چرا؟

بله، چون در صورتی می‌توانیم سیستم پرسرعت داشته باشیم که گذرگاه پرسرعت برای خواندن و نوشتن داده‌ها بر روی حافظه داشته باشیم. اما این پرسرعت بودن باید برای همه اجزا برقرار باشد چون میزان تسریع، به کند ترین جزء بستگی دارد.

- برای اتصال GPU به سیستم از چه درگاهی استفاده میشود؟ ویژگیهای این درگاه چیست؟

توجه داریم که همه‌ی دستگاه‌های GPU از طریق باس PCI-E به پردازنده وصل می‌شوند. فرض می‌کنیم این گذرگاه از نوع PCI-E 2.0 است که در حال حاضر سریع‌ترین باس موجود با نرخ انتقال اطلاعاتی ۵ گیگابایت بر ثانیه می‌باشد. PCI-E 3.0 نیز امروزه معرفی شده است و مقدار پهنای باند ممکن را به طور قابل توجهی افزایش داده است.
اگرچه برای دریافت داده از پردازنده، باید از طریق دستگاه northbridge و گذرگاه کم سرعت FSB (front-side bus) عبور کنیم. FSB می تواند تا نرخ کلاک ۱۶۰۰ مگاهرتز را اجرا کند اگرچه در بعضی طراحی ها به مراتب کمتر است. این فرکانس تنها یک سوم فرکانس پردازنده سریع است.
حافظه از طریق northbridge و دستگاه های جانبی از طریق northbridge chipset و southbridge chipset به پردازنده دسترسی پیدا می‌کنند. Northbridge با همه اجزای پرسرعت مثل حافظه،CPU و اتصالات باس PCI-E ارتباط برقرار می‌کند.South bridge با اجزای کم‌سرعت مثل hard disk، USB، keyboard و network connection ارتباط برقرار می‌کند. همچنین

می‌توان کنترل کننده هارد دیسک را به اتصالات PCI-E وصل کرد که باعث دسترسی پرسرعت داده از طریق این سیستم می‌شود.

PCI-E بر مبنای یک پهنای باند تضمین شده کار می‌کند. در سیستم‌های PCI قدیمی، هر جزء می‌توانست همه‌ی پهنای باند را در اختیار بگیرد که باعث می‌شد در هر زمانی تنها یک جز پهنای باند را داشته باشد. بنابراین با اضافه کردن card بیشتر، مقدار پهنای باند کمتری برای هر Card خواهد بود. PCI-E این مشکل را با معرفی خطوط PCI-E حل کرد. تعدادی لینک های پرسرعت سریال به نام‌هایX1وX2وX4وX8وX16 با همدیگر ترکیب شده‌اند. امروزه معمولا در GPU ها حداقل از PCI-E 2.0 و X16 استفاده می‌کنند که از یک گذرگاه full duplex با نرخ انتقال داده ۵ گیگابایت بر ثانیه داریم که به معنی این است که می‌توانیم سرعت آپلود و دانلود یکسانی در زمان یکسان داشته باشیم. بنابراین می‌توانیم با یک نرخی داده به card ارسال کنیم و با همین نرخ داده از card دریافت کنیم.

- با توجه به شکل های نشان داده شده چند GPU و با چه پهنای باندی می توان در هر معماری به سیستم متصل کرد؟

به وسیله X58 و 1366 سوکت پردازنده، 36 عدد خط PCI-E در دسترس است که این بدین معناست که در X16 ، 2 عددcard و در X8 ،4 عدد card پشتیبانی می‌شود .این مسئله، تا به امروز بهترین راه حل برای پهنای باند خوب در GPU است. طراحی X58 در چیپهای lesser P55 تنها با 16 خط قابل دسترسی است، یعنی تنها به یک GPU درX16 یا دوتا در X8 نیاز داریم. با ظهور طراحی چیپI7/X58 ، اینتل طراحی sandy bridge را آغاز کرد. یکی از مهمترین پیشرفت‌ها، پشتیبانی از استاندارد SATA-3 بود که نرخ انتقال ۶۰۰ Mb/s را دارد که با SSD ها ترکیب می‌شود و کارایIO برای load و ذخیره داده‌ها را بالا می‌برد. در طراحی اینتل، سطوح پهنای باند مشابهی در سیستم وجود دارد .اینتل از 3 یا 4 کانال حافظه در بالای سیستم و از 2 کانال حافظه در پایین سیستم استفاده می‌کند AMD .تنها از یک کانال دوگانه حافظه استفاده می‌کند که به طور قابل توجهی باعث کاهش پهنای بایند در دسترس برای اتصال host-memory در CPU می‌شود. توجه داشته باشید که به هر حال، طراحی Sandybridge ، تنها 16 خط PCI-E را پشتیبانی می‌کند و پهنای باند در PCI-E از لحاظ تئوری، به 16 GB/s و در عمل به 10 GB/s محدود می‌شود .در-Sandybridge Eتعداد خطوط PCI-E برابر با 40 تاست .همان طورکه قبلا ذکر شد، تمام دستگاه های GPU از طریق گذرگاه PCI-E به سیستم متصل می‌شوند.

---

۲. اجزای اصلی پردازنده ی گرافیکی عبارت‌اند از:

I. Memory (global, constant, shared)
II. Streaming multiprocessors (SMs)
III. Streaming processors (SPs)

a. هر یک از این اجزا را شرح داده و ارتباط میان آنها را توصیف کنید.

مهمترین ویژگی قابل توجه این است که GPU در واقع آرایه‌ای از □□ هاست که هر کدام از آنها دارای □ هسته (۸ تا در G80 وGT200 ، ۳۲ الی ۴۸ تا در Fermi، بیشتر از ۸ تا در Kepler) است. این مساله، نکته اصلی است که موجب مقیاس پذیری پردازنده می‌شود. یک GPU شامل یک یا چند SM است. با اضافه کردن تعداد بیشتر SM به GPU، می‌توان تعداد بیشتری کار در زمان ثابت انجام داد یا تعداد ثابتی کار را سریعتر انجام داد(در صورت قابلیت موازی سازی Task ها)

اگر برنامه نویس یک کدی بنویسدکه استفاده از پردازنده را به N هسته محدود کند(مثلا ۲هسته)، در صورت اجرای کد روی ۴ هسته تاثیری در تسریع نخواهد داشت.

تعدادی اجزای اصلی وجود دارد که یک SM را تشکیل می‌دهند. چندین SP در هر SM وجود دارد .در این جا ۸ تا SP نشان داده شده اند، در Fermi این مقدار بین ۳۲ الی ۴۸ عدد SP و در Kepler ، تا ۱۹۲ عدد افزایش می‌یابد .در نسلهای بعدی سخت افزارها، مقدار SP ها و SM ها را نمی‌توان گفت که حتما زیاد می‌شوند(دلیلی برای آن نداریم)

هر SM به بخشی به نام register file دسترسی دارد که مانند بخشی از حافظه است که با سرعتی برابر سرعت SP کار می‌کند. بنابراین زمان انتظار در این حافظه برابر صفر است. اندازه این حافظه نسل به نسل متفاوت است و برای مرتب سازی رجیسترهای در حال استفاده و thread های در حال اجرا روی SP استفاده می‌شود.

همچنین یک حافظه اشتراکی نیز قابل دسترس برای هر SM وجود دارد. این حافظه می‌تواند به عنوان program-managed cache استفاده شود. بر خلاف Cache در CPU ، خارج کردن داده Cache بر اساس کنترل برنامه نویس وجود ندارد.

b. در GPU از حافظه GDDR استفاده می‌شود .تفاوت آن با DDR در چیست؟ چرا مانند CPU از DDR استفاده نمی‌شود؟

GDDR(Graphic Double Data Model) یکی از ورژن های حافظه DDR است که کارایی بالایی دارد. عرض گذرگاه حافظه می‌تواند تا ۵۱۲ بیت باشد که پهنای باندی ۵ تا ۱۰ برابر بیشتر از □□□ ها و تا ۱۹۰ □/□□(با سخت افزار Fermi) دارد. این پهنای باند بالا برای GPU لازم می‌باشد و دلیل اصلی استفاده از GDDRدر GPU همین موضوع است.

c. شکل ۳ درون یک □□ را نشان می‌دهد .اجزای درونی نشان داده شده را شرح دهید.

تعدادی اجزای اصلی وجود دارد که یک SM را تشکیل می‌دهند. چندین SP در هر SM وجود دارد .در این جا ۸ تا SP نشان داده شده اند، در Fermi این مقدار بین ۳۲ الی ۴۸ عدد SP و در Kepler ، تا ۱۹۲ عدد افزایش می‌یابد .در نسلهای بعدی سخت افزارها، مقدار SP ها و SM ها را نمی‌توان گفت که حتما زیاد می‌شوند(دلیلی برای آن نداریم)

هر SM دارای دو یا بیشتر واحد خاص منظوره (SPUها) است که هر کدام، دستورالعمل‌های سخت‌افزاری خاصی را اجرا می‌کنند(مثل عملیات ۲۴ بیتی پرسرعت مربوط به محاسبه سینوس و کسینوس و توان). واحدهای (double precision)،هم روی سخت‌افزارهای GT200 و Fermi قرار دارند.

d. حافظه های global ، texture و constant چه فرقی با یکدیگر دارند؟ آیا این حافظه ها از نظر فیزیکی مجزا هستند؟

هر SM، گذرگاه جداگانه‌ای برای فضای حافظه‌ی texture ، حافظه‌ی constant و حافظه‌ی global دارد. حافظه texture، یک نمای خاص از حافظه global است که برای حالتی که بین داده‌ها تعامل وجود دارد، مناسب است. (برای مثال، برای lookup table های دو بعدی یا سه بعدی). این حافظه، ویژگی خاصی مبتنی بر سخت‌افزار برای تعامل دارد. خواندن ها بیشتر موثر است مثل مد های آدرس و تعاملی که می‌تواند بدون هزینه انجام شود.
حافظه constant ، برای داده‌های فقط خواندنی استفاده می‌شود و روی سخت‌افزار cache می‌شود. حافظه constantنیز همانند حافظه texture، نمای ساده‌ای از حافظه global است. در این حافظه مقادیر ثابت و مقادیر کرنل ذخیره می‌شوند. کند است اما cache آن سایزی برابر با 8kb دارد. برای broadcast کردن استفاده می‌شود.
حافظه global کند است. و دو نوع uncached (1.0) و cached(2.0) دارد. احتیاج به خواندن ونوشتن ۱۶ بایتی سریال دارد برای انکه سریع باشد.

---

۳. مفهوم compute level (compute capability) در CUDA چیست؟

کودا چندین compute level را پشتیبانی می‌کند. سری اصلی G80 گرافیک کارت با اولین ورژن کودا ساخته شد.compute capability در سخت افزار ثابت می‌شود. برای آپدیت کردن به ورژن جدید، کاربران باید سخت افزار خود را آپدیت کنند. اگرچه به نظر می‌رسد که NVIDIA دارد آنها را مجبور به خرید کارت های بیشتر می‌کند اما در حقیقت فواید زیادی دارد. با آپدیت کرد compute capability از پلتفرم قدیمی به جدید می‌رویم و ظرفیت محاسبه را دو برابر می‌کنیم. در طی چند سالی که کودا دردسترس بوده است

می‌بینیم که توان محاسباتی بسیار پیشرفت کرده است. در مورد تفاوت های بین هر compute level در کتاب مرجع ۱ (عنوان شده در سوالات) صحبت شده‌است.

این موارد در هر compute level تفاوت دارند:

- طراحی سخت افزار
- تعداد هسته ها
- سایز کش
- دستورالعمل های محاسباتی پشتیبان شده

به طور مثال NVIDIA Tesla K20s بر پایه‌ی معماری Kepler GPU با compute capablitty 3.5 می‌باشد. وقتی با کودا برنامه نویسی می‌کنیم خیلی مهم است که درمورد ورژن های مختلف سخت افزار اطلاعات داشته‌باشیم. به طور مثال compute capability 1.x ، برای هر نخ، حافظه محلی 16KB دارد و ورژن 2.x و 3.x برای هر نخ، حافظه محلی 512KB دارد.

مرجع:

https://cvw.cac.cornell.edu/gpu/computecap

http://cuda-programming.blogspot.com/2013/01/what-is-compute-capability-in-cuda.html

---

۴. مفهوم occupancy در CUDA چیست؟ با در نظر گرفتن چه مولفه هایی محاسبه می شود؟

برای همه‌ی هسته‌های کودا دو مد profile و trace اجرا می‌شود که قسمت Occupancy experiment detail درواقع Theoretical Occupancy (یعنی اشغال تئوری) را نشان می‌دهد . حد بالای اشغال توسط پیکربندی کرنل و قابلیت‌های کودا تنظیم می‌شود. مد پروفایل، اشغال را در هنگام اجرا شدن کرنل اندازه‌گیری می‌کند ومقادیر به دست آمده را به مقدار نظری اضافه می‌کند. نمودارهای دیگر مقدار اشغال به دست آمده بر حسب SM و اینکه چطور اشغال را می‌توان با تغییر کامپایلر و پارامترهای راه اندازی تغییر داد، را نشان می‌دهند. راهنمای برنامه نویسی C در CUDA بیان می‌کند که چگونه پیاده‌سازی سخت‌افزاری یک دستگاه CUDA به گروه‌هایی از نخها که در یک بلوک wrap شده‌اند، تبدیل می‌شوند. Wrap از زمانی که نخهای آن شروع به اجرا می‌کنند تا زمانی که همه نخهای داخل wrap از کرنل خارج می‌شوند، در نظر گرفته می‌شود.

مقدار بیشینه‌ای برای تعداد wrap هایی که می‌توانند در Streaming فعال باشند، وجود دارد . Occupancy به صورت نرخ wrap های فعال در SM بر بیشینه تعداد wrap هایی که در آن SM پشتیبانی می‌شود، تعریف می‌گردد.

مقدار occupancy با این موارد محاسبه می‌شود:

- Wrap per SM: SM مقدار ماکسیمم wrap را دارد که میتوانند همزمان فعال باشند.
- Blocks per SM: SM مقدار ماکسیمم block را دارد که میتوانند همزمان فعال باشند.
- SM:Registers per SM یک مجموعه از رجیستر ها دارد که توسط همه ی نخ های فعال به اشتراک گذاشته می‌شوند.
- Shared memory per SM: SM یک مقدار ثابت از shared memory دارد که توسط همه ی نخ های فعال به اشتراک گذاشته می‌شوند.

که در مورد هرکدام از این موارد توضیحات مفصلی در لینک: https://docs.nvidia.com/gameworks/content/developertools/desktop/analysis/report/cudaexperiments/kernellevel/achievedoccupancy.htm داده شده است.

---

۵. می‌خواهیم دو بردار را به یکدیگر جمع کنیم. اگر بخواهیم هر نخ یک خروجی را تولید کند، اندیس مناسب برای بردار خروجی کدام است؟

a. i = threadIdx.x + threadIdx.y;
b. i = blockIdx.x + threadIdx.x;
c. = blockIdx.x*blockDim.x + threadIdx.x;
d. i = blockIdx.x * threadIdx.x;

نحوه محاسبه جمع دو آرایه در اسلاید ها کامل توضیح داده شده است که با توجه به آن:

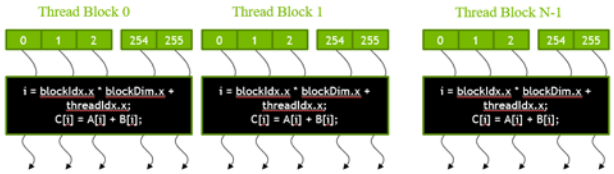
هر نخ با استفاده از اندیس‌هایی می‌فهمد مسئول پردازش چه داده‌ای است.

- blockIdx: اندیس بلوک در یک گرید
- threadIdx: اندیس نخ در یک بلوک

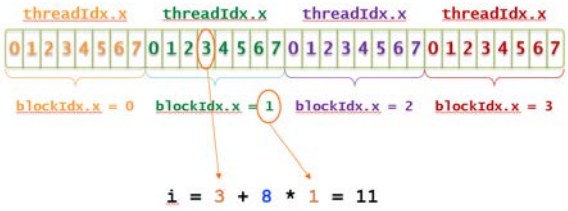
هر نخ اندیس‌هایی دارد که بر اساس آنها به داده متناظر خود دسترسی دارد و می‌تواند تصمیمات کنترلی بگیرد. نحوه محاسبه این اندیس در زیر آمده است:

i = blockIdx.x * blockDim.x + threadIdx.x;

C[i] = A[i] + B[i];



نحوه محاسبه هر اندیس از نخ ها به صورت زیر به دست می‌آید:



پس گزینه c درست است.

---

۶. برای جمع دو بردار به طول ۸۰۰۰ عنصر، هر نخ یک خروجی را تولید می کند و اندازه بلوک ۱۰۲۴ نخ می باشد .برنامه نویس kernel launch را به گونه ای تنظیم می کند که با کمترین تعداد بلوک نخ همه ی عناصر بردار پوشش داده شوند .در این شرایط چند نخ در grid وجود دارد؟

- ۸۰۰۰
- ۸۱۹۶
- ۸۱۹۲
- ۸۲۰۰

باید مقداری را انتخاب کنیم که اولا بر ۱۰۲۴ بخش پذیر باشد و دوما تعداد نخ های گرید در هر بلوک نخ ماکسیمم باشد (چون می‌خواهیم مینیمم تعداد بلوک نخ را داشته باشیم) و از ۸۰۰۰ بیشتر است.

همانطور که در گزینه ها میبینیم تنها گزینه c، این ویژگی ها را دارد.

8192/1024 = 8