به نام خدا

طراحی کامپیوتری سیستم‌های دیجیتال - زمستان ۱۴۰۳

پروژه امتیازی: **طراحی و پیاده‌سازی آرایه واحد پردازشی (Processing Element) شتاب‌دهنده[1] Eyeriss**

طراحان: مهدی محمدی نسب - محمد حسین نیکخواه

هدف پروژه :

در این تمرین قصد داریم طراحی آرایه ای از واحدهای پردازش‌کننده شتاب‌دهنده Eyeriss را، که در تمرین ۵ پیاده‌سازی شد، پیاده سازی کنیم. بنابراین در پیاده‌سازی این تمرین، ماژول پیاده‌سازی شده در تمرین قبلی مورد نیاز است.

مقدمه:

پیش‌تر با شبکه‌های عصبی پیچشی، و شتاب‌دهنده Eyeriss آشنا شده‌اید. در ادامه، جزئیات مربوط روش متصل کردن این واحدهای پردازشی شرح داده می‌شود.

معماری Eyeriss:

ساختار کلی Eyeriss، به صورت یک آرایه 12*14 از واحدهای پردازشی است، که از طریق ساختاری به نام شبکه روی تراشه[2] به بافر سراسری[3] یا واحدهای همسایه متصل می‌شوند.

هر کدام از واحدهای پردازش‌کننده، توانایی انجام کانولوشن یک بعدی را دارد. به عبارتی دیگر، این واحد با دریافت یک ماتریس تک سطری فیلتر، و یک ماتریس تک سطری ورودی، نتیجه کانولوشن را در چند مرحله ارائه می‌دهد. کانولوشن دو بعدی و سه بعدی را می‌توان مجموعه‌ای از کانولوشن‌های یک بعدی در نظر گرفت. به همین دلیل، با داشتن یک واحد پردازشی، توانایی انجام این کانولوشن ها نیز میسر می‌باشد. اما برای افزایش سرعت محاسبات، و کاهش بار پردازشی، از مجموعه‌ای از این واحدها به صورت موازی برای انجام عملیات کلی در شتاب‌دهنده Eyeriss استفاده شده است.

[1] Accelerator
[2] Network-On-Chip
[3] Global Buffer

آرایه واحدهای پردازشی (PE):

با ساختار یک واحد پردازشی Eyeriss در تمرین‌های قبل آشنا شده و آن را پیاده‌سازی کردید. در ادامه قرار است آن‌ها را در یک بعد به هم متصل کنید، به طوری که برای انجام عملیات یک کانولوشن با هم در یک زمانبندی مشخص در ارتباط باشند.

پیاده سازی:

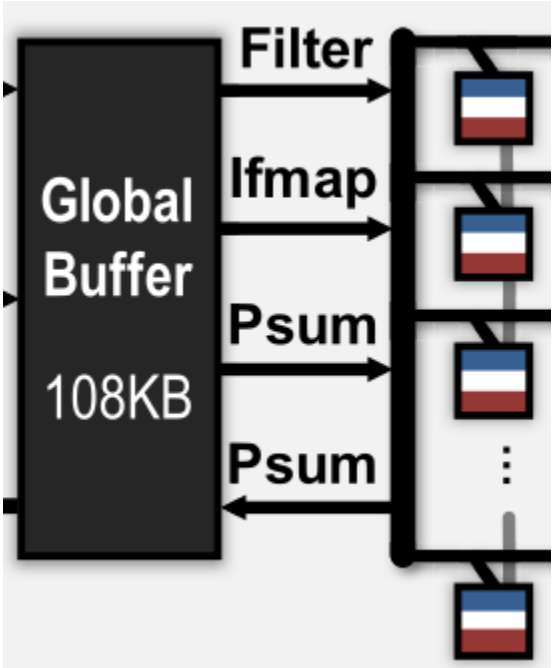
فرض کنید قرار است n واحد پردازشی مانند شکل زیر به یکدیگر متصل باشند. این واحدها عملیات کانولوشن را برای ورودی‌های و فیلترهای مختلف به صورت همزمان و مستقل انجام داده و سپس در نهایت خروجی‌های خود را با هم جمع می‌کنند. ورودی‌های دریافتی و خروجی نهایی از و به یک Global Buffer منتقل خواهند شد.

ابتدا با استفاده از ماژول SRAM خود یک Global Buffer با حجم 32KB بسازید (البته این مقدار را پارامتری در نظر بگیرید). فرض کنید در 16KB ابتدایی، ورودی ها و فیلترها قرار دارند (می‌توانید از طریق یک فایل یا از طریق تست بنچ در ابتدای کار حافظه را پر کنید). سپس ماژولی طراحی کنید که با دریافت پارامتر N، این واحدهای پردازشی را از طریق بافرهای psum ورودی و خروجی به هم متصل کند. نحوه عملکرد آن‌ها به صورت زیر است:

- همه واحدها ابتدا در حالت اول (صفر) هستند؛ یعنی قرار است عملیات کانولوشن انجام دهند (توضیح حالت صفر و یک psum، مراجعه به تمرین ۵).
- ابتدا از طریق حافظه Global Buffer، همه فیلترهای مربوط به هر PE را به ترتیب به بافر ورودی فیلتر هر PE انتقال دهید تا در scratch pad فیلتر هر یک قرار گیرند.
- سپس باید عنصر اول مربوط به ورودی ifmap هر PE به ترتیب منتقل شود؛ یعنی اگر طول ورودی‌ها L باشد، ابتدا یک عنصر از این L عنصر به هر یک از آن‌ها باید داده شود. پس از آن، PEها باید شروع به انجام عملیات کانولوشن کنند. توجه شود که تنها وقتی که همه PE ها اولین عنصر ورودی خود را دریافت کردند، کار خود را با هم شروع می‌کنند.
- بنابراین هر یک از این عناصر پردازشی، با دریافت یک سطر از ifmapها و filterهای متفاوت، عملیات خود را شروع و ادامه می‌دهند. در این قسمت شرایط همه واحدها یکسان است؛ یعنی طول فیلتر و

ورودی همه به یک اندازه بوده و همچنین در یک mod یکسان از کانولوشن قرار دارند (از سه mod توضیح داده شده در تمرین ۵).

- پس از اتمام محاسبه کانولوشن همه PEها (که باید همزمان با هم تمام شود)، همه آن‌ها در حالت دوم psum (یک) قرار می‌گیرند. در این حالت، ورودی psum اولین PE از پایین صفر است و پس از جمع آن با psum scratchpad خود، نتایج جمع را از طریق بافر خروجی به PE بالایی خود منتقل می‌کند. بقیه PEها نیز نتیجه جمع PE قبلی خود را از ورودی دریافت کرده و با psumهای خود جمع می‌کنند. در نهایت، از بالاترین واحد پردازشی، خروجی نهایی خارج می‌شود. خروجی‌ها در 16KB انتهایی حافظه Global Buffer ذخیره خواهند شد.



سایر نکات

- - انجام این تمرین به صورت گروه های دونفره خواهد بود:

1. `controller` و `datapath` طراحی شده خود را با زبان verilog مدل سازی و در Modelsim شبیه‌سازی کرده و در موعد معین در داخل سایت بارگذاری کنید.

- برای تمرین لازم است فایل های `Testbench` و `HDL` خود را مطابق توضیح داده شده در `trunk` در `subdirectory` های `trunk/doc` بارگذاری کنید. همچنین اطمینان حاصل کنید که با اجرای `trunk/sim/sim_top.tcl` تست پنج شما اجرا می شود. برای اجرای این اسکریپت می توانید از دستور زیر در Modelsim استفاده کنید:

>> do <sim_file>

- فایل ها و گزارش خود را تا قبل از موعد تحویل هر فاز، با نام CAD_HW6<SID>.zip به ترتیب در محل های مربوطه در صفحه درس آپلود کنید.

- برای آزمودن کد خودتان در این تمرین تست بنچ‌های مربوطه را خود شما طراحی و پیاده سازی می کنید، اما با توجه به اینکه تمارین بعدی درس مبتنی بر ادامه دادن این تمرین هستند حتما در طراحی و پیاده‌سازی خود به قابلیت مقاومت در برابر تغییر و پارامتری بودن ورودی‌ها و خروجی‌ها توجه لازم را داشته باشید.

- نام گذاری صحیح متغیرها، تمیزی کد و توضیحات و پارامتری بودن ورودی‌های ماژول‌ها می تواند تا حدودی کاستی‌های کد را در بخش‌های دیگر جبران کند.

- - هدف این تمرین یادگیری شماست! در صورت کشف تقلب، مطابق با قوانین درس برخورد خواهد شد.

موفق باشید