

تحويل در روز جمعه مورخ ۹۸/۰۱/۳۰ ساعت ۲۳:۵۵ از طریق سایت درس

#### نکاتی در رابطه با پروژه درس

هدف از تعریف پروژه آشنایی دانشجویان با کاربرد FPGA در پیاده‌سازی شبکه‌های عصبی و یادگیری عمیق است. شبکه‌های عصبی در سیستم‌های تشخیص متن، پردازش زبان طبیعی، تشخیص صدا و تصویر کاربردهای گسترده‌ای دارند. در فاز اول پروژه دانشجویان باید مدارهایی که در طراحی شبکه‌ی عصبی جهت طبقه‌بندی پردازش متن کاربرد دارند را پیاده‌سازی نمایند.

- در هر یک از مازول‌ها سعی کنید مدار را به صورت کاملاً ترکیبی (موازی و بدون نیاز به کلاک) طراحی نمایید.
- در صورتی که به صورت سریالی طراحی شده است، نیاز است تا یک مدار کنترل کننده برای آن طراحی شود.
- برای هر یک از مدارهای طراحی شده، فایل‌هایی با نام input.txt و golden.txt داده خواهد شد. یک فایل محیط آزمون (test bench) طراحی نمایید که اطلاعات ورودی را از فایل input.txt بخواند و نتایج شبیه‌سازی را در فایلی با نام output.txt بنویسد. جهت مقایسه می‌توانید نتایج را با فایل golden.txt مقایسه کنید.

نحوه‌ی تحويل به صورت غیرحضوری و از طریق سامانه درس خواهد بود. یک گزارش ۲ الی ۳ صفحه‌ای در رابطه با نحوه‌ی پیاده‌سازی، گزارش منابع استفاده شده و شماتیک مدار پس از ستنز نوشته و همراه با فایل‌های مدارهای طراحی شده در موعد مقرر ارسال نمایید.

چنانچه ابهامی در زمینه پروژه دارید، می‌توانید اشکالات خود را از طریق پست الکترونیکی زیر با موضوع DA.2019 رفع نمایید.

[ali.mohammadpour@aut.ac.ir](mailto:ali.mohammadpour@aut.ac.ir)

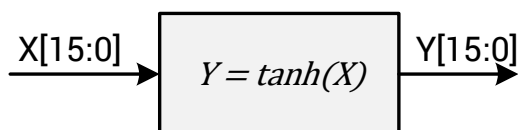
محمدپور

موفق و پیروز باشید!

## تحويل در روز جمعه مورخ ۹۸/۰۱/۳۰ ساعت ۲۳:۵۵ از طریق سایت درس

**ماژول ۰)** یک فایل Package بسازید و انواع داده‌هایی که در طراحی مورد استفاده قرار خواهد گرفت را شناسایی کرده و در آن تعریف نمایید. سپس برای طراحی سایر قسمت‌ها استفاده کنید.

**ماژول ۱)** با استفاده از IP Core های موجود در ابزار Xilinx Vivado مداری طراحی نمایید که عدد ممیز ثابت ۱۶ بیتی  $x$  را به عنوان ورودی گرفته و حاصل  $y = \tanh(x)$  را به صورت عدد ممیز ثابت ۱۶ بیتی به دست آورد. با مراجعه به فایل راهنمای CORDIC IP Core و مطالعه‌ی آن نوع داده و محدودیت بازه‌ها را در نظر بگیرید.

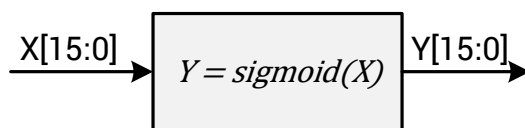


شکل ۱: مدار  $Y = \tanh(X)$

**ماژول ۲)** با استفاده از مدار طراحی شده (ماژول ۱) مداری طراحی نمایید که آرایه‌ی دو بعدی  $m \times n$  از اعداد ممیز ثابت ۱۶ بیتی را به عنوان ورودی گرفته و برای هر درایه‌ی آن حاصل  $y_{ij} = \tanh(x_{ij})$  را محاسبه کند. (می‌توانید از توصیف رفتاری استفاده کنید).

**ماژول ۳)** با استفاده از IP Core های موجود در ابزار Xilinx Vivado مداری طراحی نمایید که عدد ممیز ثابت ۱۶ بیتی  $x$  را به عنوان ورودی گرفته و حاصل  $y = \text{sigmoid}(x) = \frac{1}{1+e^{-x}}$  را به صورت عدد ممیز ثابت ۱۶ بیتی به دست آورد. جهت پیاده‌سازی تابع sigmoid می‌توانید از رابطه‌ی زیر استفاده کنید.

$$\text{sigmoid}(x) = \frac{1}{2} + \frac{1}{2} \times \tanh\left(\frac{x}{2}\right)$$



شکل ۲: مدار  $Y = \text{sigmoid}(X)$

**ماژول ۴)** با استفاده از مدار طراحی شده برای محاسبه sigmoid (ماژول ۳)، مداری طراحی نمایید که آرایه‌ی دو بعدی  $m \times n$  از اعداد ممیز ثابت ۱۶ بیتی را به عنوان ورودی گرفته و برای هر درایه‌ی آن حاصل  $\text{sigmoid}(x_{ij})$  را محاسبه کند.

$$y_{ij} = \text{sigmoid}(x_{ij}) = \frac{1}{1 + e^{-x_{ij}}}$$

**ماژول ۵)** ماژولی طراحی نمایید که دو ماتریس با ابعاد  $m \times p$  و  $p \times n$  از اعداد ممیز ثابت ۱۶ بیتی را به عنوان ورودی گرفته و حاصل ضرب برداری آن‌ها را به صورت یک ماتریس  $m \times n$  با طول داده‌ی ۳۲ بیتی محاسبه کند.

تحويل در روز جمعه مورخ ۹۸/۰۱/۳۰ ساعت ۲۳:۵۵ از طریق سایت درس

**ماژول ۶)** ماژولی طراحی نمایید که دو ماتریس با ابعاد  $m \times n$  از اعداد ممیز ثابت ۱۶ بیتی را به عنوان ورودی گرفته و درایه‌ها را نظیر به نظیر در هم ضرب کند. (هر درایه‌ی ماتریس خروجی، ۳۲ بیتی ممیز ثابت است.)

**ماژول ۷)** با استفاده از IP Core های موجود در ابزار Xilinx Vivado ماژولی طراحی نمایید که آرایه‌ی یک بعدی به طول  $n$  از عدد ممیز ثابت ۳۲ بیتی  $x_i$  را به عنوان ورودی گرفته و حاصل  $y_i = softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$  را به صورت عدد ممیز ثابت ۳۲ بیتی به دست آورد.

### توضیحات

- در سامانه‌ی درس، راهنمای استفاده از هسته‌های آماده جهت پیاده‌سازی توابع پیچیده مانند محاسبات نمایی، مثلثاتی، هذلولوی<sup>۱</sup> و تقسیم اعشاری قرار داده شده است.
- جهت انجام محاسبات نمایی و مثلثاتی و هذلولوی، می‌توان از هسته‌ی مالکیت معنوی<sup>۲</sup> CORDIC استفاده کنید.
- جهت انجام محاسبات تقسیم ممیز ثابت می‌توان از هسته‌ی مالکیت معنوی Divider و یا Floating-Point استفاده کرد.

<sup>۱</sup> Hyperbolic

<sup>۲</sup> Intellectual Property Core