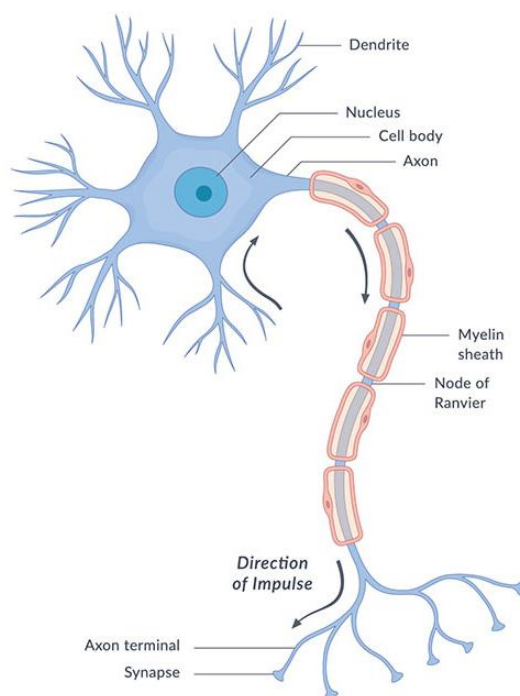




LIF Spiking Neuron RTL Design

توضیح مقدماتی از نحوه کار نورون‌ها

شبکه عصبی بدن انسان، شامل میلیاردها نورون^۱ است. تعامل این رقم شگرف از نورون‌ها در یک شبکه منجر به شکل‌گیری توانایی درک مسائل، تحلیل و تصمیم‌گیری در انسان‌ها شده است. نورون‌ها یا همان سلول‌های عصبی در این شبکه مسئول پردازش و انتقال پیام‌های الکتریکی در بدن انسان هستند. نورون‌ها شامل سه بخش اصلی هستند. اولین و مهمترین بخش جسم سلولی (سوما)^۲ است. این بخش شامل هسته سلولی است و تصمیمات مهم سلولی اعم از سنتز پروتئین‌ها در این قسمت گرفته می‌شود. دندریت‌ها^۳ بخش تاثیرگذار دیگری هستند که مانند شاخه‌های کوچکی از سلول بیرون‌زده‌اند و از طریق آنها سیگنال الکتریکی سایر نورون‌ها دریافت می‌شود. به علاوه هر نورون شامل یک آکسون^۴ است. این بخش یک رشته‌ی بلند است که پیام را از جسم سلولی به سلول‌های عصبی دیگر منتقل می‌کند. در انتهای آکسون، چندین پایانه وجود دارد که از طریق کانال‌هایی موسوم به سیناپس^۵ به دندریت سلول‌های دیگر متصل می‌شوند.



شکل ۱: تصویر مربوط به یک نورون شبکه عصبی

- ^۱ Neuron
- ^۲ Soma
- ^۳ Dendrites
- ^۴ Axon
- ^۵ Synapse



نحوه‌ی انتقال سیگنال‌های الکتریکی توسط نورون‌ها به این شکل است که با دریافت سیگنال توسط بخشی از دندریت‌ها، بنا به شدت این سیگنال الکتریکی که توسط ویژگی‌های شیمیایی کانال سیناپسی مربوطه تعیین می‌گردد، ولتاژ غشای سلولی^۶ افزایش پیدا می‌کند. ولتاژ غشای سلولی، به اختلاف پتانسیل بین داخل جسم سلولی و خارج آن گفته می‌شود که به دلیل اختلاف تراکم یون‌ها در داخل و خارج رخ می‌دهد. در حالت استراحت، این ولتاژ در حدود -0.07 ولت است. با افزایش ولتاژ غشای سلولی، در صورت گذر این ولتاژ از حد آستانه، سیگنال الکتریکی تولید می‌شود که این سیگنال از طریق آکسون منتقل می‌گردد. پس از اسپایک، ولتاژ غشایی به حدود -0.08 ولت کاهش یافته و سپس به ولتاژ استراحت باز می‌گردد. پس از تولید اسپایک، سیگنال الکتریکی توسط آکسون به پایانه‌ها منتقل شده و با آزاد شدن پیام‌رسان‌های شیمیایی^۷ در کانال‌های سیناپسی، این سیگنال الکتریکی به نورون‌های دیگر منتقل می‌شود.

مدل نوروני (LIF) Leaky Integrate-and-Fire

یکی از اهداف دیرینه‌ی دانشمندان، پیاده‌سازی سیستمی با الهام از سیستم عصبی انسان است. این سیستم مانند مغز انسان باید در زمان محدود و با مصرف توان کم قادر به ارائه عملکردی مشابه باشد. به همین دلیل، پژوهشگران تلاش کردند تا بتوانند عملکرد سلول‌های عصبی انسان را به کمک روابط ریاضی مدل کنند. در اواسط دهه‌ی ۱۹۵۰ میلادی، آلن هاجکین^۸ و اندرو هاکسلی^۹، توانستند به دستاوردی تاریخی در حوزه‌ی علوم اعصاب دست پیدا کنند و به کمک تعدادی معادله دیفرانسیل پیچیده، عملکرد مدل نورون عصبی ماهی مرکب را توصیف نمایند. اما آیا این مدل نوروני، می‌تواند پایه مناسبی جهت ساخت یک سیستم بزرگ، متشکل از تعداد بسیار زیادی نورون بدل شود؟ پاسخ این سوال با توجه به سخت‌افزار موجود در جهان احتمالاً خیر است؛ زیرا اجرای محاسبات مربوط به تعداد زیادی نورون از نوع هاجکین-هاکسلی نه تنها به لحاظ زمان اجرا، بلکه از نظر مصرف توان بسیار پرهزینه است. به همین دلیل، پژوهشگران به سمت یافتن مدل‌های نورونی رفتند که به لحاظ زیستی از نورون حقیقی دورتر باشند اما به لحاظ پیچیدگی محاسبات ساده‌تر باشند. این موضوع باعث می‌شود با ارائه ساختار مناسب جهت اجرای محاسبات این مدل‌های نورونی، محاسبات به صورت بی‌درنگ^{۱۰} صورت پذیرد و همچنین سیستم، توان مصرفی پایینی داشته باشد. یکی از معروف‌ترین و پرکاربردترین این مدل‌های نورونی، مدل Leaky Integrate-and-Fire (LIF) است. در ادامه به توضیح روابط ریاضی بیان‌کننده‌ی این مدل می‌پردازیم.

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{rest}) + R_m I(t) \quad (۱)$$

$$V(t) \geq V_{th} \rightarrow \begin{cases} V(t) = V_{rest} \\ \text{Fires Spike} \end{cases} \quad (۲)$$

^۶ Membrane Potential

^۷ Neuro Transmitter

^۸ Alan Hodgkin

^۹ Andrew Huxley

^{۱۰} Real-Time



رابطه‌ی (۱) بیان‌کننده ولتاژ غشایی یک نورون LIF در طول زمان است. این رابطه در فضای پیوسته مطرح شده است. در این رابطه، $I(t)$ مجموع جریان‌هایی است که از طریق دندریت‌ها به نورون وارد می‌گردد. در این مدل‌سازی، برای دندریت‌های مربوط به نورون، یک مقاومت (R_m) در نظر گرفته شده است و همچنین جسم سلولی با یک خازن مدل شده است. این خازن، با هدف مدل‌سازی نشی ولتاژ غشایی در نورون گذاشته شده است و مدار معادل را به یک مدار RC تبدیل کرده است. اما تفاوت اصلی در شرطی است که برای ایجاد اسپایک وجود دارد. زمانی که ولتاژ از حد آستانه (V_{th}) فراتر رود، مقدار ولتاژ غشایی به مقدار ولتاژ حالت استراحت (V_{rest}) تقلیل پیدا می‌کند. به علاوه، در این لحظه در خروجی این نورون، اسپایک تولید می‌شود که به نورون‌های بعدی منتقل می‌شود. همین رابطه را می‌توان به کمک روش اویلر، در فضای گسسته نیز نوشت که با ذات سیستم‌های دیجیتال سازگاری بیشتری دارد.

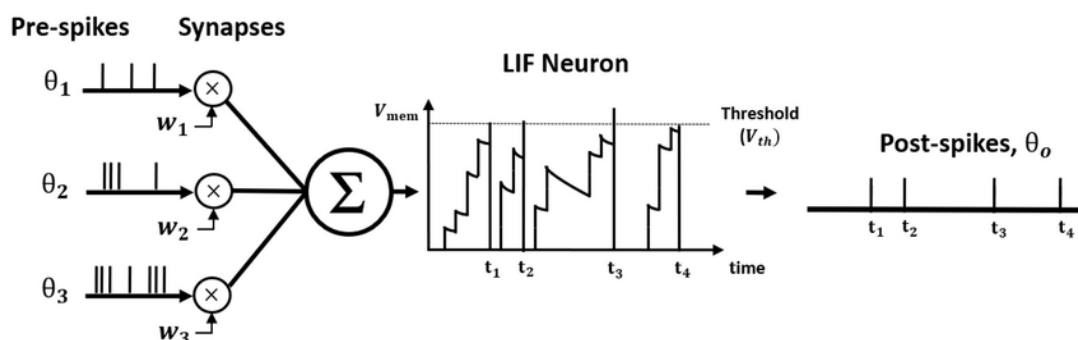
$$\frac{V[n+1]-V[n]}{\Delta t} = \frac{-(V[n]-V_{rest})+R_m I[n]}{\tau_m} \rightarrow V[n+1] = V[n] \left(1 - \frac{\Delta t}{\tau_m}\right) + \frac{\Delta t}{\tau_m} (V_{rest} + R_m I[n]) \xrightarrow{\alpha=\frac{\Delta t}{\tau_m}, \beta=\alpha R_m}$$

$$V[n+1] = V[n](1 - \alpha) + \alpha V_{rest} + \beta I[n] \rightarrow \boxed{V[n+1] = V[n](1 - \alpha) + \alpha V_{rest} + I[n]} \quad (۳)$$

$$\boxed{\text{if } V[n+1] \geq V_{th} \rightarrow \begin{cases} V[n+1] = V_{rest} \\ S[n+1] = 1 \end{cases}} \quad (۴)$$

$$\boxed{I_k[n] = \sum_{j=0}^n W_{jk} S_j} \quad (۵)$$

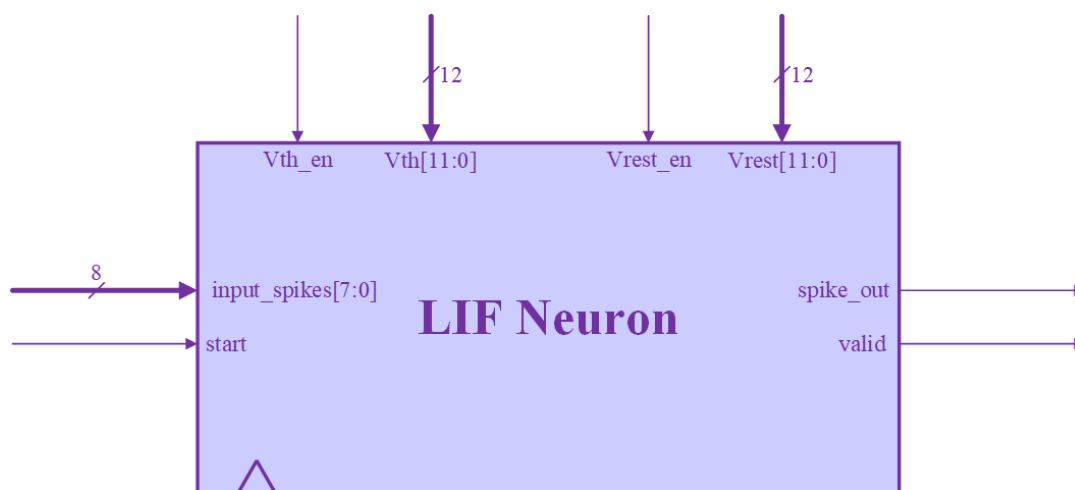
برای بیان رابطه به صورت گسسته، ما گام زمانی را به صورت Δt تعریف می‌نماییم. در بیان گسسته، اسپایک‌ها به منزله‌ی رویداد هستند؛ به این معنی که در هر گام زمانی یا نورون k ام اسپایک تولید کرده است، که در این صورت $S_k[n] = 1$ یا اینکه اسپایک تولید نکرده است که $S_k[n] = 0$ است. رابطه‌ی (۵) رابطه‌ی مهمی را برای یافتن جریان ورودی به نورون از طریق سیناپس‌ها بیان می‌کند. همانطور که در شکل ۲ نشان داده شده است، تعدادی نورون پیش‌سیناپسی که در شکل با θ نشان داده شده‌اند از طریق سیناپس‌هایی به نورون مدنظر ما متصل شده‌اند. همانطور که پیشتر اشاره شد، هر کدام از این سیناپس‌ها بنا به ویژگی‌های شیمیایی که دارا هستند، تاثیرگذاری متفاوتی دارند که این اثرگذاری با وزن (W_i) که یک عدد است، مدل می‌شود. به این صورت که در هر گام زمانی، برای محاسبه‌ی جریان ورودی به نورون، باید مجموع وزن‌های سیناپس‌هایی را محاسبه کنیم، که نورون پیش‌سیناپسی متصل به آنها در گام زمانی قبلی اسپایک تولید کرده‌اند. برای مثال اگر θ_1 و θ_3 در گام زمانی دوم ($n = 2$) اسپایک تولید کرده باشند، می‌توان گفت $I[3] = w_1 + w_3$. در رابطه‌ی (۳)، مقدار $\alpha = \frac{\Delta t}{\tau_m}$ و مقدار $\beta = \alpha R_m$ در نظر گرفته می‌شود اما باتوجه به نحوه تولید $I[n]$ ، می‌توانیم از مقدار β صرف نظر کنیم. به این صورت که می‌توانیم مقدار تک تک وزن‌ها را در β ضرب کنیم تا دیگر مجبور نباشیم مجموع آنها را در β ضرب نماییم. در انتها رابطه‌ی (۴)، شرایط تولید اسپایک را مانند آن چیزی که در حالت گسسته دیده بودیم، بررسی می‌کند.



شکل ۲: نحوه محاسبه جریان $I[n]$ ، بروزرسانی ولتاژ $V[n]$ و تولید اسپایک در هر گام زمانی

شرح مسئله

همانطور که اشاره شد، هدف نهایی از ارائه مدل‌های نورونی با ساختاری مشابه نورون‌های زیستی، این است که بتوانیم در زمان کم و با مصرف توان پایین، عملکردی مشابه سیستم عصبی انسان ارائه کنیم. برای رسیدن به این هدف، یکی از روش‌ها، طراحی سخت‌افزار اختصاصی برای این شبکه‌هاست. در این تمرین کامپیوتری، قصد داریم طراحی در سطح رجیستر^{۱۱} ارائه کنیم که توانایی انجام محاسبات مربوط به تنها یک نورون LIF را داشته باشد. در واقع، طراحی ارائه شده توسط شما باید بتواند محاسبات بیان شده در رابطه (۳)، (۴) و (۵) را انجام دهد و در انتها بعد از محاسبه مقدار جدید پتانسیل غشایی، اسپایک خروجی را نیز در صورت وقوع تولید کند. شکل ۳ نمای کلی طراحی مدنظر را نشان می‌دهد.



شکل ۳: نمای کلی ماژول LIF و سیگنال‌های ورودی و خروجی آن

^{۱۱} Register Transfer Level (RTL)



سیستم عددی

سیستم عددی در نظر گرفته شده برای این پروژه ممیز ثابت (Fixed Point) است. به این صورت که وزن‌ها، پتانسیل آستانه، پتانسیل حالت استراحت، پتانسیل غشای سلولی و تمام مقادیر میانی، همگی اعدادی با علامت هستند که ۱۲ بیت دارند. ۴ بیت سمت چپ آنها مقدار صحیح و ۸ بیت سمت راست مقدار اعشاری را مشخص می‌کنند.

ورودی‌ها و خروجی‌ها

مهمترین ورودی این ماژول، $input_spikes$ است. در این پروژه، فرض شده است که نورون LIF مدنظر، به ۸ نورون پیش‌سیناپسی از طریق سیناپس‌هایی با وزن مشخص متصل شده است. به این دلیل یک سیگنال با عرض ۸ بیت در نظر گرفته شده که هر بیت آن نشان دهنده این است که در گام زمانی قبلی، نورون متناظر با این بیت اسپایک تولید کرده است یا خیر. برای مثال اگر $input_spikes = 01000011$ باشد، یعنی در گام زمانی قبل، نورون صفرم، اول و ششم متصل به نورون LIF مدنظر، اسپایک تولید کرده‌اند. حال برای اینکه یک پروتکل ارتباطی برای این سیگنال مهم تعریف شده باشد که به وسیله آن مشخص شود در چه زمانی مقدار قرار گرفته روی ورودی $input_spikes$ معتبر است، یک سیگنال تحت عنوان $start$ تعریف کرده‌ایم. ماژول LIF هرگاه یک پالس کامل (یک شدن سیگنال در یک لبه‌ی کلاک و صفر شدن آن در یکی از لبه‌های بعدی کلاک) روی سیگنال $start$ دیده شود، مقدار $input_spikes$ ثبت شده و انجام عملیات آغاز می‌شود.

برای اینکه این واحد LIF بتواند با مقادیر متفاوتی از V_{th} و V_{rest} کار کند و تنها به مقادیری خاص محدود نشود، دو ورودی مجزا برای این دو مقدار قرار دادیم. برای گرفتن مقادیر V_{th} و V_{rest} پروتکل $handshaking$ در نظر گرفته نشده است. تنها کفایت مقدار مدنظر روی ورودی قرار بگیرد و سیگنال $enable$ متناظر فعال شود تا در لبه‌ی کلاک، داده در ماژول LIF رجیستر شود. هر زمان مقدار V_{rest} تغییر کند، مقدار پتانسیل غشایی نیز باید برابر با آن قرار بگیرد.

برای تولیدی اسپایک خروجی، یک سیگنال $spike_out$ قرار دادیم که پس از اتمام محاسبات، در صورت وقوع اسپایک مقدار ۱ و در غیر این صورت مقدار ۰ را می‌گیرد. به علاوه، برای اینکه مشخص کنیم چه زمانی مقدار خروجی معتبر است، سیگنال $valid$ را قرار داده‌ایم که پس از انجام محاسبات و قرار گرفتن اسپایک در خروجی، یک می‌شود و تا زمانی که اسپایک‌های ورودی جدید نرسیده‌اند، یک باقی می‌ماند.

نکات طراحی

همانطور که اشاره شد، یکی از اهداف مهم در راستای ارائه سیستم، مصرف پایین توان ساختار ارائه شده است. لذا نمی‌توانیم تعداد زیادی واحد محاسباتی نظیر ضرب‌کننده، جمع‌کننده و مقایسه‌کننده در طراحی خود داشته باشیم. به همین دلیل، محدودیت‌های زیر، به سیستم اعمال می‌گردد:

۱. در مدل نورونی که آن را پیاده‌سازی می‌نمایید، مقدار α برابر ۰.۲۵ در نظر گرفته شود.



۲. در طراحی خود مجاز هستید تنها از یک ALU استفاده کنید که این واحد قادر به انجام عملیات جمع، تفریق، مقایسه و شیفت به راست است. این ماژول، ۲ بیت کنترلی دارد که مشخص می‌کند کدام عملیات مدنظر است.
۳. در طراحی خود نمی‌توانید از هیچ واحد ضرب‌کننده‌ای استفاده کنید.
۴. برای ذخیره کردن وزن‌ها، یک Read-Only Memory (ROM) در نظر بگیرید تعداد خانه‌های آن ۸ و عرض بیت آن ۱۲ بیت باشد تا بتواند وزن‌های هر ۸ سیناپس را ذخیره کند. برای مقداردهی این ROM در وریلاگ، از دستور `$readmemb` استفاده نمایید و اطلاعات موجود در فایل **weights.mif** را در ROM قرار دهید.
۵. محدودیتی در قبال تعداد رجیسترها یا ماژول‌های دیگر نظیر MUX وجود ندارد.
۶. برای پیاده‌سازی ماژول‌های محاسباتی نظیر جمع‌کننده در وریلاگ، می‌توانید از اپراتور + استفاده کنید و نیازی به پیاده‌سازی نوع خاصی از آنها نیست.

نکات کدنویسی

۱. کدنویسی حتما باید به صورت *modular* انجام شود.
۲. اسم‌گذاری سیگنال‌های میانی ترجیحا منطبق بر طراحی ارائه شده باشد.
۳. هنگام نوشتن کنترلر، حتما از مدل Huffman پیروی نمایید.

موارد تحویلی

۱. ارائه شماتیک مسی‌ر داده مربوط به ماژول LIF (نکات طراحی رعایت شود) (۳۰٪).
۲. ارائه ماشین حالت کنترلر مربوط به ماژول LIF (نکات طراحی رعایت شود) (۳۰٪).
۳. ارائه کد وریلاگ کامل هر دو بخش مسی‌ر داده و کنترلر (نکات کدنویسی رعایت شود و کد تمیز باشد) (۲۵٪).
۴. برای صحت‌سنجی مدار، یک فایل برای شما آماده شده است که در آن ۲۰ داده مختلف به صورت متوالی مشخص شده‌اند که باید در یک **testbench** به ماژول طراحی شده داده شوند. به علاوه، به ازای هر ورودی، مشخص شده است آیا اسپایک رخ می‌دهد یا خیر و همچنین پتانسیل غشایی مربوط به نورون پس از انجام عملیات مربوط به این ورودی چقدر است. بعد از صحت‌سنجی، مانند شکل ۲، نمودار ولتاژ غشایی و همین‌طور اسپایک‌های تولید شده در طول زمان رسم شود. (نیاز نیست این کار اتوماتیک انجام شود؛ کفایت داده‌ها را به صورت دستی از **testbench** استخراج نموده و نمودار آنها را در اکسل یا به هر صورت دیگری رسم نمایید) (۱۵٪).