

بررسی عملکرد و بهبود نورون مصنوعی و پیادهسازی آن برروی FPGA

بنیامین کاکاوند/ گروه الکترونیک/ واحد شهرقدس/ دانشگاه آز اد اسلامی/تهران– ایر ان/ b.kakavand@gmail.com
m.mahdavi@qodsiau.ac.ir مژده مهدوی/ گروه الکترونیک/ واحد شهر قدس/ دانشگاه آزاد اسلامی/ تهران– ایران/ m.mahdavi
m.zare@gmail.com مهدی زارع/ گروه الکترونیک/ واحد شهر قدس/ دانشگاه آراد اسلامی/ تهران– ایران/ m.zare@gmail.com

چکیدہ

در این مقاله با هدف شبیهسازی رفتار تک نورون، به پیادهسازی سختافزاری یک نورون زیستی طبق مدل Hodgkin-Huxley پرداخته شده است که این مدل بیشترین شباهت را به نورون واقعی دارد و پیادهسازی آن بسیار دشوار است، ولی به کمک استفاده از قابلیت پیکربندی دوباره و اجرای دستورات بهصورت موازی، طراحی سختافزاری سیستم نورونی بهینهشده، مبتنی بر FPGA انجام شده است. برای اینمنظور در پیادهسازی تک نورون، از ایدههای سخت افزاری گوناگونی، بهره گرفته ایم تا بهترین نوع از نورون کامل را بر اساس سرعت بالا، حجمکم، دقت بالا و قابلیت توسعه پذیری، طراحی کنیم. برخی از ایدهها در طراحی و پیادهسازی عبار تناو انجام محاسبات در یعیتال با بیت محدود، سادهسازی توابع ریاضی و تبدیل عملیات ممیز شناور به ممیز ثابت، که علی غم بهینه بودن و کاهش سختافزار مصرفی، از ویژگی بیشینه موازی سازی و دقت کافی در تولید سیگنال خروجی نیز برخوردار باشد. مدل ارایه شده در این مقاله برای

کلمات کلیدی: نورون زیستی، باز پیکربندی، مدل Hodgkin-Huxley، کاهش پیچیدگی.

Evaluation, improvement and implementation of an artificial neuron on FPGA

--- Benyamin Kakavand/ Department of Electronics, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran./ b.kakavand@gmail.com

Mojdeh Mahdavi/ Department of Electronics, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran/ m.mahdavi@qodsiau.ac.ir

Mahdi Zare/ Department of Electronics, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran/ m.zare@gmail.com

ABSTRACT

In this paper, simulation the behavior of a single neuron, and the implementation of the hardware of a biochemical neuron according to the Hodgkin-Huxley model is investigated, that is closely resembles the real neuron, and its implementation is very difficult, But with the help of re-configuration and implementation on parallel, the FPGA-based hardware design of the optimized neuronal system was performed. For this purpose, we have been using a variety of hardware ideas for the implementation of single-neurons to design the best type of complete neuron based on high speed, compactness, high precision and expandability. Some of the ideas in design and implementation include doing limited digital calculations, simplifying mathematical functions, and converting floating-point operations to fixed-point, which have the optimization and reduction of consumable hardware, have the maximum parallelism and the precision of the production of the output signals. The model presented in this paper is very important for many scientists in the field of cognitive neuroscience.

Keywords: Bio-neuron, reconfiguration, Hodgkin-Huxley model, reduced complexity.



۱ – مقدمه

نورون ها سلول های تحریکی و مهاری هستند که سیگنال ها را از نورون های دیگر دریافت نموده، داده ها را پردازش کرده و آنها را به كمك يك فرآيند پيچيده الكتروشيميايي منتقل مي كنند. سازماندهي مغز و فعل و انفعالات بین نورونها بر اساس تابعیت نورونها و محل آناتومیک آنهـا تغییر میکند. در یک شبکـه نورونی هزاران نورون به صورت موازی با یکدیگر در ارتباط هستند[۱]. در دهه کنونی، شبیهسازی رفتار نورون، در یک شبکه نورونی بزرگ به یک علاقه خاص تبدیل شده است؛ به طوری که در سال های اخیر از شبکه های نورونی در زمینههای مختلف پزشکی مانند بررسی اثر بیماری اتیسم و یا تاثیر آن بر سیستم درک انسان استفاده شده است [۴–۱]. تعامل پویا بین نورونها در عملکرد شبکه نورونی بسیار مهم است. چنین تعاملی در سیستم با ماهیت سریال، قابل شبیهسازی و پیادهسازی نیست، بنابراین مدلسازی موازی، برای تعاملات پویای نورونها ضروری بوده و پیادهسازی یک سیستم نورونی زیستی در یک بستر موازی بهمنظور دستیابی به یک شبکه با تعداد نورونهای زیاد و زمان اجرای کم قابل اجرا است [۵]. ویژگی اجرای موازی نورونها در یک شبکه نورونی بر روی نرمافزار ناممکن است[۶]. از بین روشهای پیادهسازی سختافزار، پیادهسازی دیجیتال ترجیح داده می شود زیـرا ویژگیهایی چون سیگنال به نویـز کمتر، تکرارپذیری راحت تر، انعطاف پذیری زیاد و قابلیت تست راحت تر نسبت به يپادەسازى آنالـوگ را داراست[۵]. بەعلاوە، زمان توليد نمونە اوليە و هزینه مهندسی طراحی دیجیتال کمتر است[۷]. پیادهسازی به کمک پردازنده های خاص منظوره به دلیل ویژگی اجرای ترتیبی آن روش مناسبی برای مدل کردن رفتار موازی نورون ها در یک شبکه نورونی نیست. پیادهسازیهای مبتنی بر ASIC علیرغم سرعت مناسب اجرا، تـوان مصرفي پاييـن و هزينه توليد نهايي انـدک، قابليت پيکربندي دوباره را ندارد [۸]. طرحهای مبتنی بر FPGA دارای مزایای قابلیت پیکربندی دوباره و پردازش موازی است و بدین جهت، سختافزار بهینه برای پیادهسازی شبکه نورونی به شمار مرود [۵]. از جمله مزایای پیادهسازی بر روی FPGA این مزیت است که کد طراحی شده با HDL، مختص یک نوعFPGA خاص نیست و به راحتی قابل انتقال از یک نوع تراشه به نوع دیگر است[۹].

هر نورون از یک هسته و بدنه نورون تشکیل شده است. نورونها اطلاعات را از طریق شاخههایی بهنام دندریت دریافت کرده و آنها را به بدنه منتقل مي كنند. اين اطلاعات از آكسون به پايانه آكسون رسیده و به نورونهای دیگر منتقل می شود. به این ترتیب، نورونها اطلاعات را تا فاصله های دور منتقل می کنند [۱۰]. فعالیت الکتریکی نورونها از طريق نقل و انتقال يونها صورت مي گيرد [١٠]. اگر نورون توسط یک جریان خارجی و یا توسط نورونهای دیگر تحریک شود، تعادل یون ها به هم ریخته و جریان های یونی از داخل به خارج غشاء و برعکس برقرار می شود. در اثر این تحریک، پتانسیل نورون به یک مقدار مثبت افزایش پیدا میکند و بعد از مدت کوتاهی به مقدار حالت استراحت خود بر می گردد. در این حالت گویند که نورون آتش کرده است و یا اسپایک زده است، شکل (۱) نمودار آتش کردن نورون را در اثر تحریک نشان میدهد. در طراحی نورون هر چه به سمت مدلهای انتزاعیی پیش برویم، بهرهوری محاسباتی افزایش می یابد، نمونه هایی از مدل انتزاعی در مراجع [۱۴-۱۱] مورد استفاده قـرار گرفته است. اگرچه این نوع شبکهها در کاربردهای خاص، چون تشخيص الكوو . . . بسيار مورد استفاده قرار مى گيرد، ولى نمى توان از آن انتظار رفتار یک نورون زیستی را داشت.



شکل ۱: تغییرات سطح ولتاژ نورون در یک اسپایک بر حسب زمان[۱۰]

مدل نورونی H-H کامل ترین مدل نورونی زیستی است و رفتارهای یک نورون واقعی را از خود نشان میدهد. بنابراین میتوان از آن بهعنوان مدل مناسب برای شبیهسازی رفتارهای مغز استفاده کرد. از آن جمله می توان به مقاله [۱۵] اشاره کرد که در آن نویسنده، شبکه نورونیی از نورون های مدل Hodgkin-Huxley (H-H) را با استفاده از ASIC پیادہ سازی کردہ است، اگرچہ پیادہ سازی خاص منظورہ می تواند مزایای خود را داشته باشد، ولی در این حالت بعد از ساخت، امکان تغییر سیستم وجود ندارد. در [۱۶]، گرس و همکارانش نورون مدل H-H را بر روی FPGA پیادهسازی کردهاند. برای پیادهسازی این مدل، ابتدا مدل را در محیط Simulink شبیه سازی کرده و توسط مولد سیستم مدل را به کد قابل انتقال بر روی FPGA تبدیل کردهاند. اگرچه در این روش، پیادهسازی راحت تر و دارای زمان کمی است، اما واضح است که سیستمی که با روش مبتنی بر HDL طراحی شود، انعطاف پذیری بیشتری در بهبود طراحی و پیادهسازی بر روی انواع تراشههای FPGA خواهد داشت. پورهاج و تنگ در [۱۷] به پیادهسازی نورون مدل H-H پرداختهاند. آنها برای پیادهسازی این مدل، از همان مرحله نخست به تبدیل و گسسته سازی معادلات مربوطه با استفاده از روش اویلر پرداختند. اگرچه این نوع پیادهسازی مصرف منابع سختافزاری پایینی دارد و دارای سرعت اجرایی بالا است، ولی دقت کمی دارد در حالی که در کاربردهای پزشکی دقت مهمترین معیار است. در [۲۰-۱۸] یک شبکه نورونی زیستی بر مبنای نورونهای مدل H-H پیادهسازی شدهاست. در این پیادهسازیها از چندین مدار مجتمع آنالوگ که محاسبات مربوط به هر نورون در این مدارهای مجتمع انجام می شود، استفاده شده است. در این سیستمها، به دلیل پیادهسازی نورون با مدارهای مجتمع، امکان تغییر مدل و یا تغییر اتصالات بعد از ساخت وجود ندارد. گریسیا و همکارانش در [۲۱] شبکه نورونی پیادهسازی شده در [۲۰] را برای بررسی رفتارهای مختلف نورونها در شبکه نورونی استفاده کردهاند. آنها بهمنظور کم کردن سختافزار مصرفی، ثابت زمانی موجود در معادلات H-H را بهصورت یک مقدار ثابت فرض کردند. اگرچه این کار باعث کم شدن فضای مصرفی می شود، ولی در مصارفی که نیاز به دانستن تغییرات این ثابت زمانی باشد، پاسخ گو نیست. مدل Izhikevich را می توان بین مدل های نورون زیستی و انتزاعی با سخت افرار بالا و دقت متوسط قرار داد، که در مقالات [۲۴-۲۲] به پیادهسازی این مدل پرداخته شده است. از بین تمامی مدل های ارایه شده برای نورون، مدل H-H بهعنوان یک مدل مناسب و کامل برای شبیهسازی رفتار نورون انتخاب شده است [۲۱–۱۶]. تاکنون تمایل کمتری به پیادهسازی مدل H-H به دلیل پیچیدگی پیادهسازی و فضای مصرفی



علمی – ترویجی نجمن مهندسین برق و الکترونیک ایران – شاخه خراسان ۱۹۹۷ ینجم/ شعمار ۱۱۵ (زمستان ۱۹۹۷

فصلنامه

زیاد آن وجود داشت و است [۲۵]. قابلیت پیکربندی دوباره و اجرای موازی برای پیادهسازی شبکههای نورونی در انتخاب سختافزار مناسب بسیار حائز اهمیت است. در بین بسترهای موجود برای طراحی دیجیتال، FPGA هر دو ویژگی را دارد. بنابراین بهعنوان بستر مناسب برای این پژوهش انتخاب شده است. از مدل تک نورون بهبود یافته در این مقاله میتوان در راستای ساخت یک بستر سختافزاری مناسب برای یک نورون با کانالهای ورودی و خروجی فراوان استفاده کرد که با الهام از ساختار مغز بتواند قابلیت های مغز نظیر حجم کوچک، سرعت پردازش زیاد، مصرف توان کم و قابلیت تشخیص و مدل نورون مذکور بیان میشود. در بخش سوم الگوریتم پیشنهادی برای بهینه سازی هر کدام از اجزای موردنیاز توضیح داده میشود، در بخرش چهارم نتایج شبیه سازی هر کدام از اجزای موردنیاز وضیح داده میشود، در در انتها نیز نتیجه مقاله آمده است.

H-H -مدل نورون-۲

همان گونه که در بخش قبل اشاره شد، مدل H-H برای نورون توسط Hodgkin-Huxley پـس از آزمایشات بسیار ارایـه گردید که با این توصیـف، توانستند جایزه نوبـل را از آن خود کنند. مهمترین ویژگی این مـدل، الهام گرفتن از ساختار بیولوژیکی نورون واقعی است. غشا نورون مانند یک سد، یونهای داخل نورون را از مایع خارجی جدا می کنـد. این عمل می تواند توسط یک خازن در مدارهای الکتریکی جایگزیـن شود. بنابراین یک خازن با ظرفیـت C می تواند نقش غشا سلول را برای مدل سازی بازی کند. مدل نورونی استفاده شده در این پیادهسازی، مـدل مرتبه چهار است. دو نوع نورون تحریکی و مهاری در شبکـه وجود دارد که ساختار مدل نورون برای هر دو نوع نورون تحریکی و مهاری یکسان است. مدار شکل (۲) برای مدل سازی روابط نورون توسط هاجیکن و هاکسلی معرفی شده است [۲۶].



شکل ۲: مدار پیشنهادی توسط هاجکین و هاکسلی در یک اسپایک [۲۶]

با توجه به [۱۰] و [۲۶،۲۷] رابطه بین پتانسیل غشاء و جریانهای یونی و جریان تحریکی در معادله (۱) نشان داده شده است.

$$\frac{CdV}{dt} = I - I_{Na} - I_K - I_L \tag{1}$$

در عبارت (۱) I_{Na} (۱) نشان دهنده جریان یونیی مربوط به سدیم است. این جریان تحت تاثیر دو متغیر دریچه فعال ساز (m) وغیر فعال ساز (h) است. فعال ساز این دریچه از سه پره تشکیل شده است که با توان سه در m اثر می گذارد و سرعت باز شدن این دریچه بیشتر از سرعت بسته شدن دریچه توسط h است. رابطه (۲) نشان دهنده این جریان یونی است.

$$I_{Na} = \overline{g}_{Na} m^3 h (V - V_{Na}) \tag{(Y)}$$

جریان پتاسیمی ناشی از حرکت یون های پتاسیم در رابطه (۳) نشان

فصل ناعه محمد مرجو . علمی- ترویجی انجعن مهندسین برق والکترونیک ایران-شاخه خراسان

نجمن مهندسین برق و الکترونیک ایران-شاخه خراسان ۴۴ سیال پنجم/شماره۱۱/ زمسیتان ۱۳۹۷

$$I_K = \overline{g}_K n^4 (V - V_K) \tag{7}$$

در مدل H-H جریان های ناشی از سایر یون ها، از جمله یون کلسیم به صورت یک جریان نشتی (IL) نشان داده می شود. این جریان در رابطه (۴) نشان داده شده است.

$$I_L = g_L (V - V_L) \tag{(f)}$$

مقادیر پارامترهای مربوط به دریچههای کانالهای یونی با استفاده از معادلات (۵) تا (۱۳) بدست میآید[۲۷]. روابط (۵) تا (۷) نشان دهنده معادلات مربوط به بارامت m است.

$$\alpha_m = \frac{0.182(V-35)}{1-e^{\frac{V+35}{9}}}$$
(Δ)

$$\alpha_h = 0.25e^{\left(-\frac{V+90}{12}\right)} \tag{($)}$$

$$\alpha_N = \frac{0.02(V-25)}{1-e^{-(\frac{V-25}{9})}}$$
(Y)

روابط (۸) تــا (۱۰) نشاندهنده معادله مربوط به دریچه غیرفعالساز کانــال یون سدیم است. با صرفنظر کردن از دینامیکh ، روابطH-H از مرتبه چهار به مرتبه دو کاهش یافته است.

$$\beta_h = \frac{0.25e^{\frac{V+62}{6}}}{e^{\frac{V+90}{12}}} \tag{(A)}$$

$$\beta_m = \frac{0.125(-(V+35))}{1-e^{\frac{V+35}{2}}}$$
(9)

$$\beta_N = \frac{0.002 \left(-(V-25)\right)}{1-e^{\frac{V-25}{9}}} \tag{(1.)}$$

روابط مربوط به پارامتر کانال پتاسیمی در روابط (۱۱) تا (۱۳) نشان داده شده است.

$$\frac{dn}{dt} = \alpha_n (1-n) - \beta_n n \tag{11}$$

$$\frac{dh}{dt} = \alpha_h (1-h) - \beta_h h \tag{11}$$

$$\frac{dm}{dt} = \alpha_m \left(1 - m\right) - \beta_m h \tag{(11)}$$

پارامترهای ۲،۷۰C و g به ترتیب نشاندهنده چگالی خازنی، ولتاژ، زمان، جریان تحریکی و چگالی رسانایی است. واحدهای استفاده شده برای این پارامترها برابر است با:

 μ A/cm ² ms/sm², ms ·mV· μ ·F/Cm ²

مقادیـر پارامترهای مورد استفاده در معـادلات بیان شده، برابر است بـا C=1 ،g Na =100 ،V Na = 50، g K =80 ، V K = -100 و g L = 1.0 و VL = -67 و [۲۸،۲۷] است.

۲-۱- معماری پیشــنهادی برای پیادهسازی تک نورون براساس مدل H-H

برای پیادهسازی تک نورون معماری شکل (۳) مورد استفاده قرار گرفته است. هر کدام از بلوکهای موجود در این معماری، نشاندهنده

معادلات مربوط به تک نورون هستند که در روابط (۱) تا (۱۳) آورده شده است. كنترل ارتباط بين اين بلوكها، زمان انتقال ورودي به هـ كدام، زمان آماده شدن خروجي و تمامي فعاليتهاي مربوط به سیستم کنترل به طوری که از درستی انجام عملیات اطمینان حاصل شود، توسط کنترلر انجام می شود، که ماشین حالت آن در شکل (۴) نشان داده شده است. در ابتدا تمام بلوكها ريست ميشوند و كنترلر به مرحله بعدی، S1 تغییر حالت میدهد. در حالتS1 کلاک مربوط به بلوکهای m, n, hو x که x شامل پارامترهای m, n, h می شود $lpha_x$ فعال شده و این مدارها کار خود را شروع می کنند و تا زمانی که خروجی آنها آماده شود، کنترلر در این حالت باقی میماند. بعد از أمـاده شدن خروجي ايـن بلوكها، كنترلر با أمـدن كلاك به حالت S2 تغییر حالت میدهد. در حالت S2 کلاک بلوکهای dx/dt که x شامل m, n, h می شود، فعال شده و کنترلر تا زمانی که خروجی I_{L} ، آمـاده نشـده، در این حالت باقی میمانـد. در $\mathrm{S3}$ کلاک جریان، ا و I_K و I_K، فعال شده و با آمادهشدن خروجی آن ها، کنترلر به حالت بعد يعنى S4، تغيير حالت مىدهد. در S4 ، كلاك dv / dt فعال شده و با آماده شدن خروجی، کنترلر به حالت Finish رفته و پاسخ یک مرتبه اجرای نورون آماده می شود. کنترلر شکل (۵) که ارتباط بین بلوک نورونی و انتگرال گیرها را ممکن میسازد، از حالت S_{Init} کار خود را شروع می کند و با آمدن کلاک به حالت S1 تغییر حالت میدهد. در حالت s1 کلاک انتگرال گیر فعال می شود که با مقادیر اولیــه مقداردهی شود. در حالت ${
m S}_{
m Back}$ که حالـت بازگشت برای تکرار عملیات نیز است، کلاک انتگرال گیر غیرفعال شده، سپس کنترلر به حالت 3 میرود و تا زمانی که خروجی بلوک نورونی آماده شود، در این حالت باقی میماند.



شکل ۳: معماری پیشنهادی برای پیادهسازی تک نورون

با آماده شدن خروجی بلـوک نورونی، کنترلر به حالتS4 منتقل شده و کلاک انتگرالگیر فعال می شود که داده های جدید را دریافت کند. شکـل(۶) بلـوک دیاگرام مـدار پیاده سازی شده بـرای معادله _m مرا نشـان می دهد. با توجه به بلوک دیاگرام شکل (۳) و شکل (۶)، برای پیاده سـازی تک نورون و معادلات آن باید در ابتـدا تمام توابع مورد نیـاز، جمع/ تفریق، ضرب/ تقسیم، تابـع نمایی، توان و انتگرالگیر، با تعداد بیت مناسب پیاده سازی شوند، به طوری که نتیجه نهایی از دقت مناسـب برخوردار باشد و از طرفی مساحت زیادی را اشغال نکند. در

ایسن زمینه ایدههای مختلفی در پیادهسازی توابع در این مقاله به کار گرفته شده است، که در ادامه توضیحات مربوط به آنها شرح داده خواهد شد.



شكل ۴: كنترلر پیشنهادی برای كنترل عملیات بین بلوكها



شکل ۵: کنترلر طراحی شده برای تک نورون



 \mathbf{a}_{m} شکل ۶: بلوک دیاگرام مدار پیادہ
سازی شدہ برای

۳-الگوريتم پيشنهادى

در پیادهسازی سختافزاری باید نحوه نمایش دادهها و تعداد بیتهای لازم برای نمایش انتخاب شود. برای تصمیم گیری در مورد تعداد بیت مناسب برای پیادهسازی بهینه با دقت بالا، مراحل زیر به ترتیب انجام شد:

- شبیهسـازی تک نورون توسـطSimulink به ازای بازه پیوستهای از ورودیها و بدسـت آوردن بازه تغییر تک تک متغیرها، و در نهایت تعییـن تعداد بیت مورد نیاز برای قسمـت اعشار با استفاده از بازه



علمی – ترویجی انجمن مهندسین برق و الکترونیک ایران – شاخه خراسان سال پنجم/شمار ۱۱۵/زمستان ۱۳۹۷

تغيير متغيرها.

- محـدود کردن تعداد ارقام اعشار تمامی پارامترها و عملیات موجود در کد تک نورون شبیهسازی شده در Simulink، تغییر تعداد ارقام اعشـار و بررسی تغییرات حاصـل در نتیجه نهایی و فرکانس آتش کـردن نورونها و بدسـت آوردن تعداد بیتهای مـورد نیاز برای نمایش ارقام در پیادهسازی دیجیتال.
- از جملـه توابعی کـه در پیادهسازی آنهـا از روشهای محاسبات سختافـزاری برای بهبود پیادهسازی استفـاده شده است می توان بـه تابع نمایی اشاره کرد که محـدوده ورودی آن با صرف حداقل سختافزار افزایش مییابـد. همچنین تابع توان که در پیادهسازی آن عـلاوه بر داشتـن دقت کافی، تعداد ماژولهـای مورد استفاده کاهش یافـت. معیارهای انتخاب روش پیادهسـازی توابع عبارتند از: استفـاده از توابع موردنظر در چند بخش مدار، تعداد ارقام مورد استفاده در عملیات، تعداد کلاک پالسهایی که میتوان برای تولید خروجی بعـد از اعمال ورودی منتظر ماند، اولویت سرعت یا حجم سخت افزاری و بازه تغییرات ورودی. در ادامه به بررسی پیادهسازی توابع و ایدههای ارایه شده می پردازیم.

۳-۱- پیادہسازی دیجیتال تابع نمایی

با توجه به روابط (۵) و (۶)، تابع نمایی از جمله توابع مهم و پر کاربرد در پیادهسازی تـک نورون اسـت و دقت خروجـی آن تاثیر زیادی در دقـت خروجی نهایـی دارد. برای پیادهسازی ایـن تابع روشهای مختلفی چـون پیادهسازی با روش اویلر، پیادهسازی بر اساس جدول و پیادهسازی با استفاده از الگوریتم CORDIC وجـود دارد. روش ماویلـر بهدلیل مصـرف سختافزار زیاد و دقت کـم در خروجی روش مناسبـی نیست. ایده به کـار رفته در این مقالـه، تلفیقی از دو روش یافتـن بـه پیادهسازی بهینـه، سرعت اجرای زیاد و دقت بالا است. پیادهسازی بـر اساس جدول و الگوریتـم CORDIC بهمنظور دست یافتـن بـه پیادهسازی بهینـه، سرعت اجرای زیاد و دقـت بالا است. پیادهسازی بولیک، Sin و ... کاربـرد دارد. اساس کار این الگوریتم بـا استفاده از جمع و شیفتهای پیاپی است که تا رسیدن به جواب بـا استفاده از جمع و شیفتهای پیاپی است که تا رسیدن به جواب دقـت کافی تولید می شود [۲۹]. روابط (۱۴) تا (۲۰) نشاندهنده این الگوریتم است.

$$x[j+1] = x[j] - \sigma y^{2-j} y[i]$$
(14)

$$v[i+1] = v[i] + \sigma^{2-j} x[i] \tag{10}$$

$$z[i+1] = z[i] - \sigma_i^{\tan^{-1}(2^{-j})}$$
(19)

$$z[0] = \theta \tag{1Y}$$

$$x[0] = y[0] = \frac{1}{k} \tag{1}$$

$$k_{h} = 0.82816$$
 (19)

$$x[f] = y[f] = e^{\theta} \tag{(Y \cdot)}$$

همان طور که از روابط (۱۴) تا (۲۰) پیداست، این الگوریتم با استفاده از شیفترجیستر و جمع کننده قابل پیادهسازی است. برای پیادهسازی بخـش ¹- tan ، بـا توجه به محـدوده j و مقـدار ¹- tan ، مقادیر آن از قبـل محاسبه و ذخیـره شدند. در شکـل (۷) معمـاری به کار رفته بـرای پیادهسازی این الگوریتم نشان داده شـده است. بیشینه مقدار ورودی برای کار کـرد صحیح این الگوریتم برابر است با 1.1817 = $\boldsymbol{\theta}$ بهطوری کـه بـه ازای مقادیر بزرگتر از آن خروجـی درستی بدست نمی آید[۲۰].

از این رو، بنا به درستی روابط (۲۱) و (۲۲) ایده دیگری که از آن

فصل نامه عصر رو علمی- ترویجی

نجمن مهندسین برق و الکترونیک ایران-شاخه خراسان ۴۶ سال پنجم/شماره۱۱/ زمستان ۱۳۹۷

استفاده شد، جـدا کردن بخش صحیح از بخش اعشـار ورودی تابع نمایـی، محاسبه جداگانه هر بخــش و در نهایت حاصلضرب دو پاسخ بدست آمده برای محاسبه پاسخ تابع نمایی است.



شکل ۷: بلوک دیاگرام پیشنهادی برای پیادهسازی الگوریتم CORDIC

$$\exp(\theta) = \exp(m).\exp(n) \tag{(1)}$$

$$\theta = m + n \tag{(YY)}$$

معماری پیشنهادی در شکل (۸) نشان داده شده است. با استفاده از این پیادهسازی، پاسخ دقیق تابع نمایی، در زمان کم و با مصرف کمینه سختافزار بدست آمد.



شکل ۸: بلوک دیاگرام پیشنهادی برای پیادهسازی تابع نمایی

۲-۲- پیادەسازی دیجیتال تابع توان

با توجه به روابط (۲) و (۳) برای پیادهسازی جریانها نیاز به محاسبه توان سه و توان چهار است. برای پیادهسازی توان، معماری شکل (۹) در نظر گرفته شده است. به اینصورت تعداد ضرب کنندهها کاهش یافته و به سرعت بالا و دقت مناسب دست یافتیم.



شکل ۹: بلوک دیاگرام محاسبهگر توان

۳-۳-پیادہسازی دیجیتال تابع انتگرالگیر

با توجه به شکل (۳) در بخش قبل، ابتدا مقادیر n،V،h،m ، باید با مقادیر اولیه مقداردهی شوند و بعد از نخستین اجرا، خروجی انتگرال dv/dt بدست آید. برای پیادهسازی انتگرال گیر از ساده شده روش اویلر با گامهای زمانی بسیار کوچک که از زمان آماده شدن پاسخ نورون بدست آمده، استفاده شد. رابطه استفاده شده برای پیادهسازی انتگرال گیر در عبارت (۲۳) نشان داده شده است.

$$x(t + \Delta t) = x(t) + \Delta t \dot{x}(t) \tag{(17)}$$

با توجه به رابطه (۲۳) پیادهسازی انتگرال گیر با استفاده از مدار شکل (۱۰) قابل انجام است.



شکل ۱۰: بلوک دیاگرام معماری پیشنهادی برای پیادهسازی انتگرالگیر

۴-نتایج شبیهسازی و پیکربندی محیط آزمایش

برای ارزیابی درستی خروجی نتایج مربوط به شبیهسازی پیادهسازی دیجیتال با شبیهسازی انجام شده با کمک نرمافزار MATLAB و در محیط Simulink مقایسه شده است. برای پیادهسازی بخشهای دیجیتال از نرمافزارXilinx ISE بهره گرفته شده است. کلیه شبیه سازی های انجام شده نیز در نرمافزار Modelsim صورت گرفته است. برای شروع پیادهسازی و بررسی مرحله به مرحله درستی آن، بایـد یک مـدل پایه وجود داشتـه باشد، که بتـوان مقادیر دادههای پیادهسازی شده را با مقادیر آن سنجید. به این منظور تک نورون مـدل H-H در Simulink شبیهسازی شد. پس از شبیهسازی سطح بالای انجام شده، با اعمال ورودیهای مختلف به این نورون و بررسی تاثیر تغییرات این ورودی، از درستی عملکرد و شبیهسازی اطمینان حاصل شد. شکل (۱۱) نمودار سطح ولتاژ غشاء نورون شبیهسازی شده را به ازای جریان ورودی ثابت $I_{ext} = 0.5 \, A$ نشان میدهد. نم ودار تغییرات مقادیر دریچه های n،m و h که سرعت تغییرات ایــن دریچهها را نشان میدهد، به ترتیب در شکل (۱۲)، شکل (۱۳) و شکل (۱۴) آورده شده است.



شکل ۱۱: نمودار سطح ولتاژ غشا نورون به ازای جریان خارجی ۵۰۰٫۵۳۸-با بررسی شکل (۱۱)، شکل (۱۲)، شکل (۱۳) و شکل (۱۴) معلوم می شود که با اعمال جریان ورودی، ابتدا در یچه سا سرعت زیاد فعال شده و باعث افزایش ولتاژ غشاء می شود. سرعت باز شدن این در یچه بیشتر از تغییارات در یچه های n و h است. بنابراین قبل از این که در یچه های n و h بتوانند مانع تغییارات ناشی از در یچه m شوند، ولتاژ غشاء نورون به حد آستانه رسیده و نورون آتش می کند. سپس با بیشتر شدن مقار n و کمتر شدن مقار ا

کاهـش مییابد و این عملیات تازمانی که جریان اعمالی وجود داشته باشـد، تکرار می شود، نمـودار سطح ولتاژ غشاء نـورون بدون حضور جریان تحریکی در شکـل(۱۵) نشان داده شده است. با توجه به این شکـل، در زمانهای اولیه شبیهسازی سطح ولتاژ غشاء نورون بهدلیل مقادیر اولیه و بـرای ایجاد تعادل بین آنها، تغییر کرده، ولی با ثابت شدن مقادیـر پارامترها، ولتاژ غشاء نورون نیز ثابت می ماند. در شکل (۱۶) نمـودار سطح ولتاژ غشاء نورون به ازای جریان تحریکی خارجی ثابـت A =0.7m و ولتاژ اولیـه کهها- = ۷ و متغیر دریچه اولیه n ثابـت ۵ = نشـان داده شده است. به ازای جریان تحریکی اعمال شده، نورون به صورت تکرار شونده آتش می کند. بنابراین درستی دادههای تک نورون ثابت می شود.



شکل ۱۲: نمودار تغییرات متغیر دریچه m برحسب زمان به ازای جریان خارجی I=0.5mA



شکل ۱۳: نمودار تغییرات متغیر دریچه n برحسب زمان به ازای جریان خارجی I=0.5mA



شکل ۱۴: نمودار تغییرات متغیر دریچه h برحسب زمان به ازای جریان خارجی I=0.5mA

در طول مراحل پیادهسازی، هر کدام از توابع تشکیل دهنده تک نورون، بهطور جداگانه و به ازای بازههای ورودی مختلف تست شدند و مقادیرشان با مقادیر شبیهسازی شده مقایسه شد. در شکل(۱۷) شبیهسازی نورون پیشنهادی با نرمافزار Modelsim آمده است و درستی عملکرد آن بررسی شده است.



علمی – ترویجی انجمن مهندسین برق و الکترونیک ایران – شاخه خراسان سال ینجم/شمار ۱۱۵/زمستان ۱۳۹۷



شکل ۱۵: نمودار سطح ولتاژ غشا نورون بدون حضور جریان تحریکی



شکل ۱۶: نمودار سطح ولتاژ غشا نورون در حضور جریان تحریکی ثابت



شکل ۱۷: بخشی از شبیهسازی اسپایکهای نورون پیشنهادی در نرمافزار Modelsim

۵– گزارش پیادەسـازی و ارزیابی عملکرد سیستم برحسب زمان

در این بخش درستی سیستم پیادهسازی شده با استفاده از دادههای معتبر بدست آمده از شبیهسازی بخش قبل، مورد بررسی قرار گرفته و به تحلیل عملکرد سیستم، تحلیل زمانی و سختافزار مصرفی پرداخته شده است. در جدول (۱) نتایج سنتز مدل پیشنهادی و منبع سختافزاری قابل دسترس در تراشه موردنظر آمده است.

دول ۱: گزارش سنتز مدل پیشنهادی	÷
--------------------------------	---

	HH(ARTIX-7)		
	used	available	
FF	1700	160000	
RAM	700byte	13Mbyte	
LUT	5400	215000	
DSP BLUK	4	740	
MAX SPEED	250MHZ		

در محاسبات Matlab برای بهروزرسانی گامها از مدت زمانی که برابر است با ۷ میکروثانیه استفاده می شود اما در سخت افزار موجود بیشترین کلاک برابر با ۴ نانوثانیه اندازه گیری گردید بدون اینکه مدار از مقدار زمانی که دارد، تخطی کند. در Matlab تمامی محاسبات با



نجمن مهندسین برق و الکترونیک ایران-شاخه خراسان ۲۸۸ سال پنجم/شماره۱۱۰ زمستان ۱۳۹۷

سیستم ممیز شناور و با دقت 64 Bit double 64 انجام شده در حالی که برای دقت مورد نظر می که ۱۲ بیت صحیح و ۲۰ بیت اعشار میباشد. در جدول (۲) مقایسه زمانی و دقت در نرمافزار مطلب و تراشه ARTIX-7 را مشاهده می کنید. جدول ۲: مقایسه زمانی Artix و Matlab

	· · ·	
	Artix-7	Matlab
Speed	4 ns	7 μs
Accuracy	fix point	floating point

در شکل (۱۸)نمودار سطح ولتاژ غشاء تک نورون به ازای جریان ورودی I = 0.7m A و مقدار اولیه n = 0.01 نشان داده شده است. نمودار قرمز رنگ نشان دهنده نتایج حاصل از شبیه سازی است. همان نمودار آبی رنگ نشان دهنده نتایج حاصل از پیاده سازی است. همان طور که در این نمودار مشخص است، هیچ تغییری در فرکانس آتش کردن نورون که از جمله خصوصیات مهم در رفتار نورون است، دیده نمی شود.

با توجه به شکل (۱۸) تنها اختلاف اندک در دو نمودار است که اًن هم از خطای دیجیتالسازی ناشی می شود و قابل چشم پوشی است. هر چند به علت محدودیت های فضایی مربوط به تراشه مورد نظر، امکان موازیسازی کامل در سیستم وجود ندارد ولی در طراحی دیجیتال موردنظر برای پیاده سازی سیستم نورونی، حداکثر موازی سازی به کار گرفته شد. این موضوع سبب شده است که با توجـه به فرکانس کاری سیستم دیجیتال پیاده سازی شده، سرعت عملکرد آن بسیار بیشتر از سرعت اجرای آن بر روی نرمافزار باشد. به طوریکه زمان اجرایی سیستم نزدیک به زمان واقعی است، در حالی که زمان شبیهسازی سیستم با استفاده از MATLAB بر روی پردازنده Intel Core is 2.8GHz برابر ۱۷۳۶/۸۱ ثانیه است. در مشابهترین مقاله، تـک نورونـی بـا روش Izhikevich [۳۱] پیادەسازی شدە است، با توجه به اینکه محاسبات مدل H-H بسیار پیچیدهتر از سایر مدل های نورونی برای دستیابی به دقت بالا میباشد اما نسبت به مـدل IZH کـه از محاسبات بسیار سادهای برخـوردار است، سرعت محاسبات که ملاک مهمی در عملکرد نورون است، بیشتر است.



شکل ۱۸: نمودار سطح ولتاژ غشا نورون در حضور جریان تحریکی خارجی I=0.7mA و n=0.01 (نتایج حاصل از نرمافزار Matlab (قرمز)، نتایج حاصل از پیادہسازی (آبی))

همان طور کـه در جـدول (۳) مشاهده میشـود، استفـاده از تراشه Artix-7 بهدلیـل برخـورداری از بلوکهـای DSP، بالابردن سرعت محاسبات و کاهش سایر منابع سختافزاری، مناسبترین گزینه برای پیادهسازی این طرح بـوده است که در معماریهای جدید نیز به آن تاکید شده است [۳۲،۳۳]. FPGA implementation of a neural network for character recognition," Advances in Neural Networks Symposium, pp. 1357– 1365, China, 2006.

- [14]S. Sahin, Y. Becerikli, and S. Yazici," Neural network implementation in hardware using FPGAs," in Neural Information Processing, pp. 1105–1112, China, 2016.
- [15]Q. Zou, Y. Bornat, J. Tomas, S. Renaud, and A. Destexhe," Real-time simulations of networks of Hodgkin-Huxley neurons using analog circuits," presented at the Neurocomput, vol.69, pp 1137–1140, 2006.
- [16]E. L. Graas, E. A. Brown, and R. H. Lee," An FPGA-based approach to high-speed simulation of conductance-based neuron models," Neuroinformatics, vol. 2, pp. 417-435, 2004.
- [17]P. Pourhaj and D. H.-Y. Teng, "FPGA based pipelined architecture for action potential simulation in biological neural systems, "Conference on Electrical and Computer Engineering, pp. 1–4, Canada, 2010.
- [18]Y. Zhang, J. Nunez-Yanez, J. McGeehan, E. Regan, and S. Kelly," A biophysically accurate floating point somatic neuro processor ",Conference on Field Programmable Logic and Applications, pp. 26–31, Prague, 2009.
- [19]S. Saïghi, J. Tomas, Y. Bornat, B. Belhadj, O. Malot, and S. Renaud, "Real-time multi-board architecture for analog spiking neural networks ", Proceedings of 2010 IEEE International Symposium on Circuits, pp. 1939–1942, France, 2010.
- [20]S. Saïghi, T. Levi, B. Belhadj, O. Malot, and J. Tomas, "Hardware system for biologically realistic, plastic, and real-time spiking neural network simulations, "IEEE International Joint Conference on Neural Networks, pp. 1-7, Spain, 2010.
- [21]F. Grassia, T. Lévi, J. Tomas, S. Renaud, and S. Saïghi, " A neuromimetic spiking neural network for simulating cortical circuits." IEEE Information Sciences, pp. 1–6. USA, 2011.
- circuits, "IEEE Information Sciences, pp. 1–6, USA, 2011. [22]M. Mokhtar, D. Halliday, and A. Tyrrell," Hippocampus-inspired spiking neural network on FPGA", Evolvable Systems: From Biology to Hardware, pp. 362–371, Prague, 2008.
- [23]K. L. Rice, M. A. Bhuiyan, T. M. Taha, C. N. Vutsinas, and M. C. Smith," FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition, "IEEE in International Conference on Reconfigurable Computing and FPGAs, pp. 451–456, Mexico, 2009.
- [24]A. Cassidy and A. G. Andreou," Dynamical digital silicon neurons", IEEE Biomedical Circuits and Systems Conference, pp. 289–292, USA, 2008.
- [25]E. M. Izhikevich, "Which model to use for cortical spiking neurons?," IEEE Transactions on Neural Networks, vol. 15, pp. 1063–1070, 2004.
- [26]A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," The Journal of physiology, vol. 117, pp. 500–544, 1952.
- [27]C. Börgers, S. Epstein, and N. J. Kopell, "Background gamma rhythmicity and attention in cortical local circuits: a computational study," Proceedings of the National Academy of Sciences of the United States of America, vol. 102, pp. 7002-7007, 2005.
- [28]G. B. Ermentrout and N. Kopell, "Fine structure of neural spiking and synchronization in the presence of conduction delays," Proceedings of the National Academy of Science., vol. 95, pp. 1259–1264, 1998.
- [29]M. D. Ercegovac and T. Lang, Digital arithmetic, Morgan Kaufmann publishes, 2003.
- [30]D. R. Llamocca-Obregón and C. P. Agurto-Ríos, "A fixedpoint implementation of the expanded hyperbolic CORDIC algorithm," Latin American applied research, vol. 37, pp. 83–91, 2007.
- [31]H. Soleimani, A. Ahmadi. And M. Bavandpour, "Biologically Inspired Spiking Neurons," IEEE Transactions on Circuits and Systems I: Regular Papers., vol. 59, pp. 2991 - 3004, 2012.
- [32]QY. Sun, QX. Wu, X. Wang, L Hou; "A spiking neural network for extraction of features in color opponent visual pathway and FPGA implementation" Int. J. Neurocomputing, vol 228 ,pp 119-132, 2017.
- [33]L Wan, Y Luo," Efficient neuron architecture for FPGA based spiking neural network" IEEE Signals and Systems, pp. 1-6, UK, 2016.

جدول ۳: مقایسه سختافزاری و سرعت

	IZH(VIRTIX-II)		HH(A	RTIX-7)
	used	available	used	available
بلوک DSP			4	740
سرعت ماکزیمم	241MHZ		250	MHZ

۶-نتيجەگيرى

در ایت مقاله به پیاده سازی یک سیست م نورونی زیستی مدل H-H مبتنی بر FPGA پرداخته شد. در مقاله حاضر در پیاده سازی نورون از روش ساده سازی محاسباتی بهره برده و با ایده های دیگر در طراحی، دقت محاسبات را افزایش دادیم. همچنین برای غلبه بر محدودیت های عملکرد نرمافزاری، طراحی سخت افزار با سرعت و با عملکرد بالا انجام شد. بدین منظور نیز ایده های مختلفی در طول مراحل پیاده سازی در نظر گرفته شد که با به کار گرفتن آنها، تک نورون پیاده سازی شده از جمله ویژگی های دیگر این طرح، مقیاس پذیر و قابل توسعه بودن آن برای تعداد بیشتری از نورون ها است. از مدل بهبودیافته در این مقاله می توان برای ساخت یک بستر مناسب که با الهام از ساختار مغرز بتواند قابلیت های مغز نظیر حجم کوچک، سرعت پردازش زیاد، مصرف توان کم و قابلیت تشخیص و استنتاج چشمگیری را در خود داشته باشد، استفاده نمود.

مراجع

- [1]E. Kandel, J. Schwartz, and T. Jessell, Principles of Neural Science, 4nd edition., McGraw-Hill Medical, 2000.
- [2]J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," Neurocomputing. Mag., vol. 74, no. 1, pp. 239-255, 2010.
- [3]M. A. Cavu, C. Karakuzu, S. Sahin, and M. Yakut, "Neural network training based on FPGA with floating point number format and it's performance," Neural computing & applications. Mag., vol. 6, no. 1, pp. 195-202, 2011.
 [4]Y. Heo and H. Song, "Circuit modeling and implementation of
- [4]Y. Heo and H. Song, "Circuit modeling and implementation of a biological neuron using a negative resistor for neuron chip," BioChip J, vol. 92, no. 2, pp. 17-24, 2012.
 [5]A. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neu-
- [5]A. Muthuramalingam, S. Himavathi, and E. Srinivasan, "Neural network implementation using FPGA: issues and application," International journal of information technology, vol. 4, no. 2, pp. 86-92, 2008.
- [6]F. J. Pelayo, E. Ros, X. Arreguit, and A. Prieto, "VLSI Implementation of a Neural Model Using Spikes," Analog Integrated Circuits and Signal Processing, vol. 13, no. 1, pp. 111-121, 1997.
- [7]L. Gatet, H. Tap-Béteille, and F. Bony, "Comparison between analog and digital neural network implementations for rangefinding applications," Neural Networks, IEEE Transactions, vol. 20, no. 3, pp. 460-470, 2009.
- [8]L. Wanhammar, DSP Integrated Circuits, 2nd edition., Academic Press, 1999.
- [9]G. Li, V. Talebi, A. Yoonessi, and C. L. Baker Jr, "A FPGA realtime model of single and multiple visual cortex neurons," J. Neurosci. Methods, vol. 193, no. 1, pp. 62-66, 2010.
- [10]E. M. Izhikevich, Dynamical systems in neuroscience: the geometry of excitability and bursting, 1st ed. MIT press, 2006.
- [11]J. A. Bailey, R. Wilcock, P. R. Wilson "Behavioral simulation and synthesis of biological neuron systems using synthesizable VHDL," Neurocomputing. vol. 74, pp. 2392-2406, 2011.
- [12]M. Bahoura and C. W. Park, "FPGA-implementation of dynamic time delay neural network for power amplifier behavioral modeling," Analog Integrated Circuits and Signal Processing, vol. 73, no. 3, pp. 819-828, 2012.
- [13]F. Khan, M. Uppal, W. C. Song, M. J. Kang, and A. Mirza,"



فصلنامه