

User name:  
**Volodymyr Matiiievskiy**

Check ID:  
**1013386398**

Check date:  
**09.01.2023 13:43:04 EET**

Check type:  
**Doc vs Internet + Library**

Report date:  
**09.01.2023 13:45:50 EET**

User ID:  
**100010994**

File name: **ЧередниченкоВ.О**

Page count: **49** Word count: **7871** Character count: **59402** File size: **2.56 MB** File ID: **1013158618**

Text modifications detected (similarity score might be affected)

## 15.5% Matches

Highest match: **13.9%** with Internet source (<https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/4676/280.pdf>)

15.5% Internet sources 13

Page 51

No Library sources found

## 0% Quotes

No quotes found

Exclusion of references is off

## 0% Exclusions

No exclusions

## Modifind

Text modifications detected. Find more details in the online report.

Replaced characters 12

Suspicious formatting 8 Pages

## ВСТУП

Розвиток теорії автоматичного управління, як і розвиток будь-якого іншого напрямку науки, характеризується ускладненням розв'язуваних завдань та підвищенням якісних показників необхідних рішень [1]. Традиційні методи управління переважно спираються на теорію лінійних систем, тоді як реальні об'єкти є, за своєю природою, нелінійними. Нейромережеві системи управління є нове, високотехнологічне напрям у теорії управління і ставляться до класу нелінійних динамічних систем [2]. Висока швидкодія за рахунок розпаралелювання вхідної інформації у поєднанні зі здатністю до навчання нейронних мереж робить цю технологію дуже привабливою для створення пристроїв керування в автоматичних системах.

В даний час основним методом реалізації нейромережевих систем управління є програмний, з використанням комп'ютерної техніки або спеціалізованих контролерів, побудованих на її основі, що значно звужує коло практичної реалізації систем управління через значну вартість таких регуляторів і робить їх практично недоцільними та недоступними для використання в простих системах управління, крім того, комп'ютерні нейромережні регулятори мають обмежену продуктивність і вимагають значних витрат часу на навчання. Рекурентність та послідовність дій процедури навчання нейромережі при реалізації на всій кількості налагоджувальних параметрів не дозволяє повністю вирішити проблему швидкодії процедури навчання нейромережевих структур у темпі з динамікою об'єкта управління.

Єдиною альтернативою цьому є розпаралелювання процедури навчання та роботи внутрішніх елементів нейромережевих структур. Такі можливості з'являються при апаратно-програмній реалізації нейромережевих структур, побудованих на нейрочіпах або програмованих логічних інтегральних структурах (ПЛІС-FPGA) [3].

Однією з проблем при апаратно-програмній реалізації нейромережових структур є реалізація штучної нейронної мережі та її частини штучного нейрона засобами FPGA.

З вищесказаного завдання створення елемента штучної нейронної мережі, саме штучного нейрона дуже актуальна.

Таким чином, **мета магістерської роботи** полягає у розробці елемента штучної нейронної мережі, а саме штучного нейрона мовою проектування пристроїв VHDL у базисі Xilinx.

**Для досягнення поставленої мети необхідно вирішити такі завдання:**

- проаналізувати пристрої, що застосовуються для моделювання нейронних мереж та штучних нейронів на підставі чого розробити рекомендації щодо апаратної реалізації, математичної моделі штучної нейронної мережі;
- провести аналіз основних етапів схемотехнічного проектування в базисі Xilinx;
- виходячи з обраного інструментального комплексу Xilinx Spartan 3E Starter Board та системи проектування провести проектування елемента штучного нейрона як частини штучної нейронної мережі;
- розглянути питання охорони праці під час проектування вбудованих систем та у процесі виробництва радіоелектронної апаратури.

## РОЗДІЛ 1. ПРИСТРОЇ ДЛЯ МОДЕЛЮВАННЯ НЕЙРОНІВ.

### РЕКОМЕНДАЦІЇ З АПАРАТНОЇ РЕАЛІЗАЦІЇ

#### 1.1. Класифікація пристроїв для моделювання нейрона

На сьогоднішній день існує багато різноманітних підходів до побудови пристроїв для моделювання нейронів (ПМН), тому постає завдання їх аналізу та класифікації [3, 4].

ПМН можна класифікувати за такими ознаками:

- Форма подання інформації (цифрова або аналогова);
- Тип елементної бази;
- характер налаштування синапсів (постійні чи змінні);
- час передачі сигналів (синхронні чи асинхронні).

Класифікація ПМН формою подання інформації показано на рис. 1.1.

За формою подання інформації всі ПМН можуть бути поділені на два основні класи (рис. 1.1):

- ПМН, у яких вхідні, вихідні сигнали і вагові коефіцієнти зв'язку представляються як цифрових кодів (вони зазвичай будуються на цифрових елементах – інверторах, логічних елементах, тригерах, регістрах, цифрових суматорах, цифрових компараторах, запам'ятовуючих пристроях та інших.). Цифрові ПМН бувають однорозрядні (наприклад, формальний нейрон) та багаторозрядні;
- ПМН, в яких вхідні, вихідні сигнали та вагові коефіцієнти зв'язку подаються у вигляді аналогових сигналів (вони зазвичай будуються на аналогових елементах – операційних підсилювачах та компараторах або на електронних компонентах: діодах, транзисторах, тиристорах, резисторах, конденсаторах та ін.).

Залежно від виду аналогового сигналу розрізняють ПМН із поданням інформації у вигляді:

- 1) рівня напруги,
- 2) величини струму,
- 3) частоти імпульсів,

4) тривалості часових інтервалів.

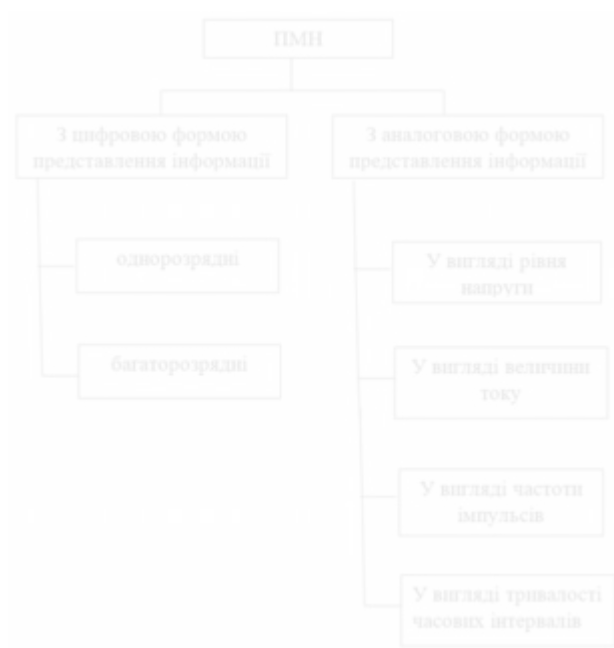


Рис.1.1. Класифікація пристроїв для моделювання нейрона за формою подання інформації

Класифікація нейронних елементів на кшталт елементної бази представлена на рис.1. 2.

Серед електронної елементної бази можна виділити три основні групи: надвеликі інтегральні схеми (СВІС), інтегральні схеми середнього ступеня інтеграції (СІС) та програмовані логічні інтегральні схеми (ПЛІС). СВІС містять до 1 мільйона елементів на кристалі, тоді як СІС лише до 1000 елементів. Основним елементом аналогових мікросхем є біполярні та польові транзистори.

ПЛІС (programmable logic device, PLD) – електронний компонент, який використовується для створення цифрових інтегральних схем. На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС визначається за допомогою програмування, а не при виготовленні.

Основні типи ПЛІС:

4

- FPGA (field-programmable gate array);
- CPLD (complex programmable logic device);
- PAL (programmable array logic);
- GAL (gate array logic).

Типи PAL та GAL мають простішу архітектуру. Альтернативою ПЛІС є:

- БМК (Uncommitted Logic Array) – базові матричні кристали, які потребують виробництва на заводі для програмування. Вони є великими інтегральними схемами, які технологічно програмуються шляхом нанесення маски сполук останнього шару металізації;
- ASIC (application-specific integrated circuit) – спеціалізовані замовні інтегральні схеми для вирішення конкретного завдання;
- Спеціалізовані процесори або мікроконтролери (повільніше, ніж ПЛІС).

Серед оптоелектронної елементної бази можна виділити чотири основні групи:

- оптоелектронні мікросхеми на основі фотодіодів та операційних підсилювачів (ФД-ОП), пристрої на основі оптично керованих транспарантів (ОКТ);
- пристрої на основі оптичних бістабільних SEED-пристроїв (Self-Electrooptic Effect Device) та пристрої на основі просторово-часових модуляторів світла (ПМС)).

Класифікація ПМН за характером настроювання синапсів:

- з фіксованими ваговими коефіцієнтами (вибираються одразу, виходячи з умов завдання);
- з ваговими коефіцієнтами, що налаштовуються (у процесі навчання відбувається настроювання ваг синаптичних зв'язків).

У ряді нейронних елементів функція активації може залежати від часу передачі імпульсу (сигналу) каналами зв'язку  $I_{ij}$ , тому час передачі сигналів моделі нейронних елементів можна розділити на синхронні та асинхронні. Синхронним нейроном називають такий нейрон, у якого час передачі  $I_{ij}$

кожного зв'язку дорівнює нулю або фіксованій постійній  $\tau$ . Асинхронним називають такий нейрон, у якого час передачі  $T_{ij}$  для кожного зв'язку між елементами є постійним і своїм.



Рис. 1. 2. Класифікація ПМН на кшталт елементної бази

### 1.2. Відомі пристрої для моделювання нейрона та їх недоліки

У патентних джерелах України знайдено біля ста різноманітних ПМН. Нижче розглянемо детально ПМН окремих типів, зображені на рис. 1.1.

З проведеного огляду літератури видно, що на території колишнього СРСР дослідженнями в ПМН займалися такі інститути, як: Інститут кібернетики ім. В. М. Глушкова АН України, Львівський державний медичний інститут, Інститут фізіології ім. І. П. Павлова, Львівський політехнічний інститут, Вінницький національний технічний університет, Одеський державний університет, Дніпропетровський державний університет та інші.

Одним із представників цифрових однорозрядних ПМН є модель формального нейрона [5], що містить логічні елементи І-НІ та інвертори з вхідними діодами Шоттки та транзисторами. В основі моделі – звичайний R-S-тригер. Цифрові логічні схеми часто використовують для побудови формального нейрона, але іноді використовують екзотичні елементи, наприклад, іонотронний транзистор (а. с. СРСР № 619933) або пристрої з циліндричними магнітними доменами - ЦМД (а. с. СРСР № 1013984).

Формальні моделі нейрона є дуже спрощеними, оскільки вхідні та вихідні сигнали у них бінарні (хоча ваги розрядів можуть бути багатозначними). Саме тому вони мало функціональні та дуже рідко використовуються на сучасному етапі для побудови нейронних мереж. Зараз перспективніші моделі нейронів з багатозначними (нескінченно значними) вхідними, вихідними сигналами та вагами синапсів [4].

Цифрові багаторозрядні ПМН – чисельніший клас. Їх типовим представником є пристрій [6] для моделювання нейрона (рис.1.3), що містить дві групи інформаційних входів  $11...1n$  та  $21...2n$ ,  $n$  блоків  $31...3n$  зміни синаптичних ваг, настановні входи  $41...4n$ , суматор 5, елементи І 6 і 7, регістри 8 і 9, логічний блок 10, керуючі входи 11 - 16, інформаційні виходи 17 - 20. У блоках  $31 ... 3n$  зміни синаптичних ваг відбувається множення вхідних сигналів  $x_{i-1}$ ,  $x_{i-1}$  на поточні значення синаптичних ваг  $y_{i1}$ , ...,  $i_i$ . В режимах градуального та формального нейронів, множення величини порога  $(-\theta)$  на 1, отримані добутки сумуються в суматорі 5 (для перших двох режимів  $P_i = \sum_{j=1}^{n-1} x_{ij} \cdot y_{ij} - \theta$ ). На виході 20 формується вихідний сигнал  $Y_{вих}$ , який буде для режиму градуального  $Y_{вих} = \max\{0; P_i\}$  і формального нейрона  $Y_{вих} = \text{sign}(\sum_{j=1}^{n-1} x_{ij} \cdot y_{ij} - \theta)$ .

Недоліком цього пристрою є недостатня швидкодія у зв'язку з послідовним принципом обробки  $n$  операндів у суматорі.

Головною перевагою багаторозрядних цифрових ПМН є висока (цифрова) точність моделювання, а головним недоліком – великі апаратні витрати. , Що складається з суматора 1, інтеграторів 2 і 3, інвертора 4, множення блоку 5, реле 6 з контактною групою 7, входів 8 і 9 і виходу. Як бачимо, основою таких ПМН є операційний підсилювач, тому вони мають



меншу точність моделювання, ніж багаторозрядні цифрові ПМН, але апаратно їх простіше.

Попри це, апаратурні витрати таких ПМН ще дозволяють будувати з їхньої основи великомасштабні нейронні мережі.

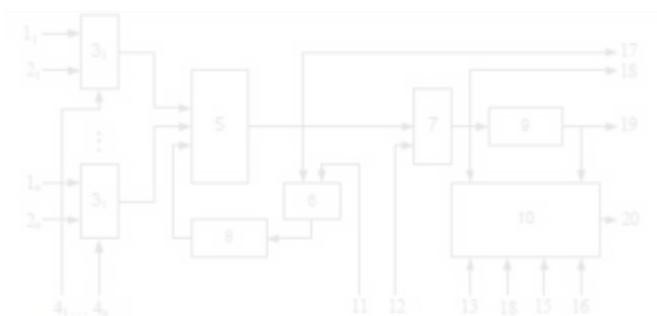


Рис. 1.3. Багаторозрядний цифровий пристрій для моделювання нейрона

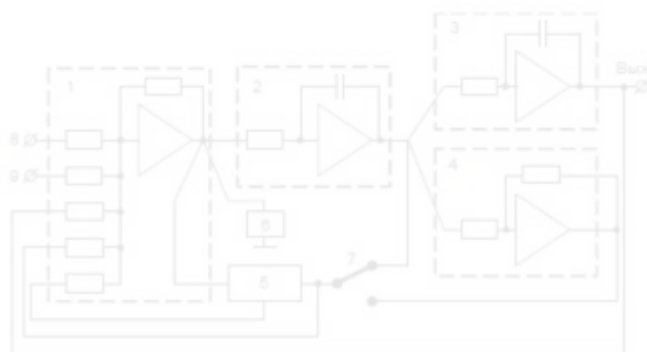


Рис.1.4. Пристрій для відтворення передавальної функції нервової клітини

Одним із представників аналогових ПМН, в яких інформація подається у вигляді величини струму, є пристрій для моделювання нейрона [8], побудований на І2Л-вентиліях і містить два колекторні інвертори-мультіплікатори струму, чотиризначні І2Л-вентилі та порогові детектори. Функціональну схему пристрою показано на рис.1.5. Вона містить групу каналів збудження 1 і гальмування 2 суматор 3, пороговий блок 4 і блок формування вихідного сигналу 5. Кожна група каналів 1 та 2 містить блоки 6

– 9 зважування вхідних сигналів та вхідні порогові блоки 10 – 13 відповідно. Група каналів збудження додатково містить суматор 14 зважених вхідних сигналів і інвертор 15.

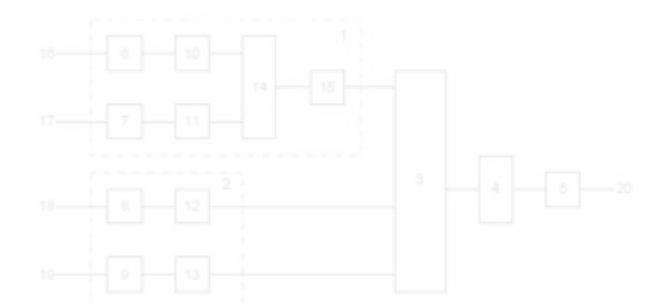


Рис. 1.5. Пристрій для моделювання нейрона на І2Л-вентилях

Найбільш численними є ПМН із частотно-імпульсною формою подання інформації. Це можна пояснити тим, що в біологічних нейронах інформація також представляється імпульсами, частота яких залежить від рівня збудження. За апаратною складністю такі ПМН можна поділити на три класи:

1) складні – з використанням цифрової елементної бази (лічильники, тригери, перетворювачі частота – напруга та напруга – код). Приклад – патент України №46470;

2) середньої складності – з використанням аналогової елементної бази (операційні підсилювачі, аналогові компаратори, перетворювачі частота – напруга та напруга – частота).

3) найпростіші – з використанням нелінійних електронних (оптоелектронних) елементів (одноперехідний транзистор [9], лавинний транзистор [10], тиристор [11], біспін-прилад [12], тригер Шмідта на МДП-транзисторах [13]).

Звісно, з погляду апаратної реалізації нейромереж з великою кількістю елементів найбільш привабливими є найпростіші ПМН.

Як приклад розглянемо ПМН [9], що містить (рис. 1.6.) одноперехідний транзистор 1, біполярний транзистор 2, резистори 3 – 6, резистор

навантаження 7, конденсатори 8 – 10, діоди 11 – 12. Активним елементом є одноперехідний транзистор 1, резистори 3, 5 і 7 забезпечують заданий режим роботи, що збуджує 13 і гальмує входи 14 містять діоди 11, 12 і ланцюги тимчасового підсумовування, які складаються з резисторів 6, 4 і конденсаторів 9, 10. - 13] є наявність електричних входів і виходів (що не дозволяє організувати велику кількість зв'язків нейронів) і низька здатність навантаження (неможливість керувати великими струмами, необхідними для живлення матриць світлодіодів або напівпровідникових лазерів при організації оптичних виходів нейрона).

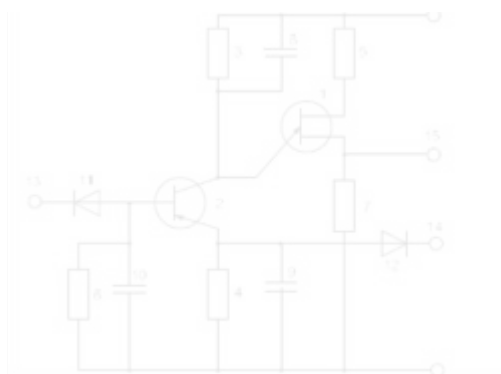


Рис. 1.6. Пристрій для моделювання нейрона на одноперехідному транзисторі

ПМН, у яких інформація подається у вигляді тривалості часових інтервалів, під час проведеного огляду не було знайдено. Проте авторами у роботі [14] запропоновано модель нейрона логіко-тимчасового типу, що працює саме з тривалістю часових інтервалів.

### 1.3. Рекомендації з апаратної реалізації нейронних мереж та шляхи удосконалення пристроїв для моделювання нейрона

З проведеного у попередньому розділі огляду типових ПМН видно, що:

- 1) цифрові однорозрядні ПМН дуже спрощено відтворюють роботу біологічного нейрона, тому доречні лише на початковому етапі дослідження штучних нейронних структур;

- 2) побудова цифрових багаторозрядних ПМН вимагає значних апаратних витрат, але вони досить точні та багатофункціональні, а тому придатні для використання при детальному вивченні та дослідженні роботи окремих біологічних нейронів або невеликих ансамблів нейронів. Огляд показав, що їхні недоліки – низька швидкодія та значна апаратна складність. Тому необхідно збільшувати швидкодію таких пристроїв, що пропонується авторами в роботах [15 – 17];
- 3) ПМН, що працюють з аналоговою інформацією, представленою напругою, струмом або частотою, зазвичай значно простіше, ніж багаторозрядні цифрові ПМН, але мають обмежені функціональні можливості та невисоку точність моделювання.

Тому, крім використання для дослідження функціонування біологічних нейронів, їх можна використовувати для побудови апаратних реалізацій нейронних мереж із великою кількістю елементів.

Безперечним є той факт, що для отримання максимуму переваг від застосування нейронних мереж для практичних завдань розпізнавання образів різного походження необхідні апаратні реалізації нейронних мереж з якомога більшою кількістю нейронних елементів. В ідеалі ця кількість має наближатися до кількості нейронів у мозку людини –  $(2...5) \cdot 10^{11}$ . Тому такі нейронні елементи повинні бути якомога простішими та придатними для виконання в інтегральному вигляді.

Відома, наприклад, структура імпульсної нейронної мережі [18] на основі оптоелектронної елементної бази просторово-безперервних оптоелектронних структур (ПНОЕС), показана на рис. 1.7. Вона складається з  $n$  вхідних нейронних елементів  $E_1 \dots E_n$ , на які надходять вхідні сигнали  $x_1 \dots x_n$ . ~~Х~~  $N$  інтернейронів  $E_1 \dots E_N$ , двох циліндричних лінз ЦЛ 1 і ЦЛ2, транспаранта  $T$  (виготовленого, наприклад, у вигляді фотопластики, коефіцієнти прозорості локальних ділянок якої відповідають значенням вагових коефіцієнтів зв'язку інтернейронів), оптично керованого

11

прозорого ОУТ із системою з  $2m$  пар смугових вертикальних електродів,  $m$  вихідних нейронних елементів  $EO_1 \dots EO_n$ , волоконно-оптичного джгута ВОЖ із фоконами  $\Phi_1$  та  $\Phi_2$ .

В ОУТ непарні пари електродів відповідають збудливим (Excitatory) зв'язкам і мають висновки  $YE_i$  і  $YE_i'$ , а парні пари електродів відповідають зв'язкам, що гальмують (Inhibitory) і мають висновки  $YI_i$  і  $YI_i'$  ( $i=1\dots m$ ). Прозорий Т і ОУТ утворюють загальну матрицю ваги зв'язків всіх нейронів ІНМ.

Головним недоліком цієї апаратної реалізації є погані конструктивно-технологічні параметри, а саме:

- через наявність таких оптичних елементів, як: циліндричні лінзи, фокони та волоконно-оптичний джгут — маса пристрою буде великою;
- через необхідність певної відстані (визначається фокусною відстанню циліндричних лінз) від масиву вхідних та інтернейронів до матриці зв'язків і від матриці зв'язків до вихідних нейронів пристрій матиме значну довжину, а значить, і об'єм;

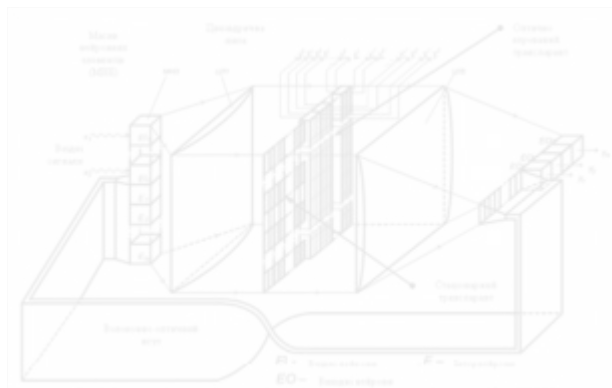


Рис. 1.7. Структура імпульсної нейронної мережі на основі оптоелектронної елементної бази ПНОЕС

- через те, що вихідний оптичний сигнал одного нейрона розширюється циліндричною лінзою на весь рядок матриці зв'язків, він повинен мати значну потужність. Це вимагає використання як світлодіодів, так і потужних напівпровідникових лазерів, що, у свою чергу, викликає необхідність потужних вихідних каскадів нейронних елементів, здатних видавати великі струми. А при великій кількості нейронів у мережі оптична потужність на один елемент матриці зв'язків все одно може виявитися недостатньою.

Таким чином, шляхи поліпшення багаторозрядних цифрових УМН полягають у підвищенні швидкодії, а аналогових УМН – у підвищенні здатності навантаження (вихідної оптичної потужності) і створенні таких УМН, які дозволяли б будувати на їх основі нейронні мережі з покращеними масо-габаритними показниками. Результати проведеного аналізу наведено як логічної структури на рис.1.8.



Рис. 1.8. Недоліки існуючих УМН та шляхи їх усунення

#### 1.4. Математична модель штучного нейрона

На рис. 1.9 представлена модель нейрона, що лежить в основі штучних нейронних мереж. До складу нейрона входять помножувачі (синапси), суматор та пристрій, що реалізує функцію активації.

Синапси виконують скалярний добуток  $n$ -вимірного вектору вхідних даних  $x = x_1, x_2, \dots, x_n$  і  $n$ -вимірного вектора вагових даних  $w = w_1, w_2, \dots, w_n$ .

Суматор виконує складання зважених входів:

$$\Sigma_j = \sum_{i=1}^n w_i x_i \quad 1.1$$

де  $w_i$  ( $i=1, 2, \dots, n$ ) - вага синапса,

$x_i$  ( $i=1, 2, \dots, n$ ) - вхідний сигнал,

$n$  - число входів нейрона,

$j$  - номер нейрона.

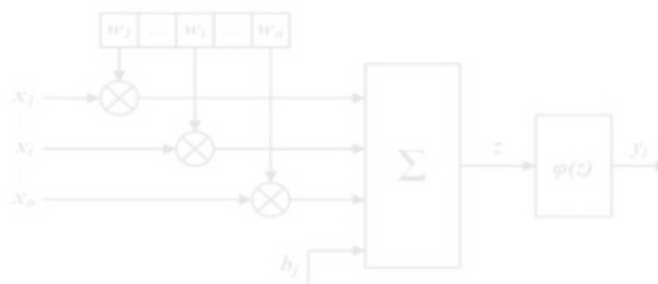


Рис. 1.9. Структура штучного нейрона

Нейрон показаний на рис. 1.9., доповнений скалярним зсувом  $b_j$ . Зміщення підсумовується з завислими входами і призводить до зсуву аргументу функції активації на величину  $b_j$ :

$$z = \Sigma_j + b_j, \quad 1.2$$

де  $b_j$  – значення зсуву.

Функція активації  $\varphi(z)$  визначає вихідний сигнал нейрона.

Сигмоїдальна функція активації має вигляд (1.3).

$$y_i = \varphi(z) = \frac{1}{1 + e^{-z}}$$

1.3

Математична модель нейрона описується системою розглянутих співвідношень

$$\begin{cases} z = \sum_j x_j + b_j, \\ \varphi(z) = \frac{1}{1 + e^{-z}}; \end{cases}$$

1.4

### 1.5. Апаратне уявлення штучного нейрона

Один з варіантів архітектурного рішення штучного нейрона на основі ПЛІС Xilinx Spartan IIIЕ розглянутий у роботі [4] та представлений на рис. 1.10.

Структура нейрона складається із різних обчислювальних підблоків. Для визначення значення функції активації використовується таблиця (LUT), що зберігається у блоці пам'яті RAM. У структурі міститься один помножувач (MUL) та один суматор (ADD). Вхідні сигнали (input\*) вводяться паралельно R0M1. CONTROL UNIT, керуючи стартовими сигналами (start), кінцевими сигналами (reset) та лічильником сигналів нейрона (num) координує функції блоків, забезпечує синхронізацію вхідних сигналів та відповідних їм ваг. Перший вхідний сигнал і відповідний йому вага зберігається R0M2, після чого блок MUL послідовно їх перемножує і заносить в R0M3. Та ж операція повторюється й інших вхідних сигналів, після чого суматор ADD підсумовує зважені входи (sum) і додає до них значення зміщення BIEST. Результат підсумовування (sum\_out) надходить у таблицю визначення значень сигмоїдальної функції LUT, після чого отримане значення зберігається в блоці OUT, що подає сигнал закінчення процесу.



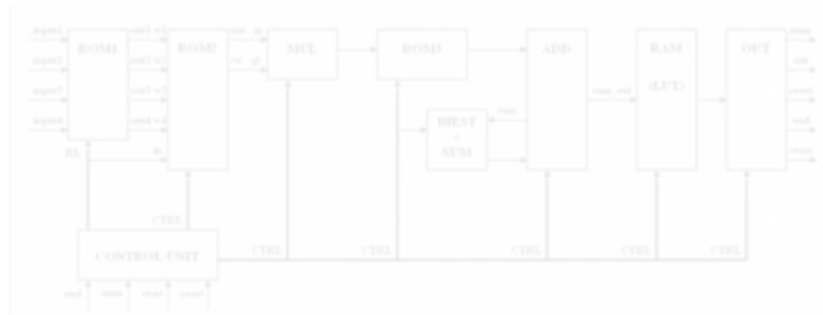


Рис.1.10. Структурна схема апаратного представлення нейрона на основі ПЛІС Spartan -3E

### 1.6. Висновки до розділу

Виходячи з проведеного аналізу, можна зробити висновок, що:

- для моделювання ПМН необхідно використовувати модель з цифровою формою подання інформації, в яких вхідні, вихідні сигнали та вагові коефіцієнти, зв'язки подаються у вигляді цифрових кодів (вони зазвичай будуються на цифрових елементах – інверторах, логічних елементах, тригерах, регістрах, цифрових суматорах, цифрових компараторах, запам'ятовуючих пристроях та ін;
- найдоцільніше як апаратну частину проекту використовувати програмовані інтегральні схеми (ПЛІС) у яких на відміну звичайних цифрових мікросхем, логіка роботи задається з допомогою програмування, а не під час виготовлення;
- за характером настроювання синапсів необхідно вибрати фіксовані вагові коефіцієнти.

## **РОЗДІЛ 2. АНАЛІЗ ОСНОВНИХ ЕТАПІВ СХЕМОТЕХНІЧНОГО ПРОЕКТУВАННЯ У БАЗИСІ XILINX**

### **2.1. Основні етапи схемотехнічного проектування цифрових пристроїв**

Процес автоматизованого схемотехнічного проектування радіoeлектронних пристроїв складається із кількох етапів (див. рис.2.1).

На першому етапі (1) складний проектований пристрій розбивається на функціонально закінчені блоки та виробляються приватні технічні завдання (ТЗ) на кожен окремий блок. ТЗ передбачає опис зовнішніх та внутрішніх параметрів: вхідних та вихідних сигналів, діапазону частот, споживаної потужності, умов експлуатації, граничних допусків на основні характеристики тощо. На цьому етапі дуже багато залежить від особистості розробника-конструктора: від його знань, інтуїції, інтелекту та кругозору.

На другому етапі (2) після формулювання ТЗ на блок, що розробляється складається його принципова електрична схема початкового (нульового) наближення. Це зазвичай робиться розробником також на підставі власного досвіду та досвіду попередніх розробок. Тут же вибираються компоненти схем, як-то: транзистори, діоди, ІМС, резистори, конденсатори, котушки індуктивності та ін, а також номінальні значення та допуски на параметри компонентів.

Далі на третьому етапі (3) вибирається система автоматизованого схемотехнічного проектування (моделювання), а в ній програма, яка найкраще підійде для аналізу даної електронної схеми та дозволить судити про відповідність ТЗ обраної схеми. Іноді у вибраному пакеті потрібна не одна, а група програм для всіх необхідних обчислень. Наприклад, аналіз по постійному струму, аналіз у часовій та частотній областях.

Потім принципова схема проектованого блоку готується щодо комп'ютерного аналізу та вводиться на згадку ПЕОМ текстовим чи графічним способом (етап 4).



Рис. 2.1. Основні етапи автоматизованого проектування електронних схем

Далі виходячи з вбудованої бібліотеки моделей компонентів автоматично складається математична модель аналізованого устрою за введеною принциповою схемою (етап 5).

На етапі 6 проводиться аналіз математичної моделі електронної схеми діалоговому режимі. Наприклад, у разі аналізу схеми аналогового пристрою передбачається виконання таких видів розрахунків:

- Розрахунок схеми по постійному струму;
- Розрахунок схеми в частотній області; наприклад, обчислення АЧХ та ФЧХ, спектральної щільності шуму;
- Розрахунок у тимчасовій області; наприклад, визначення перехідних та імпульсних характеристик, проведення спектрального аналізу.

Отримані в результаті аналізу характеристики схеми порівнюються з даними ТЗ та (або) з результатами випробувань макета (етапи 7 та 8).

На підставі цього порівняння приймається рішення про прийняття чи відхилення розглянутого варіанта проекту (етап 9). Таке рішення проводиться неформально, оскільки у деяких випадках інженерне розуміння суті справи дозволяє знехтувати деяким розбіжністю результатів комп'ютерного аналізу з ТЗ. Після ухвалення проекту розробляється технічна документація для подальшого виготовлення розробленого пристрою та проведення випробувань (етап 11).

Якщо характеристики незадовільні, то принципова схема та (або) моделі компонентів мають бути змінені (етап 10).

Цикл аналізу потім повторюється знову. Саме тут, під час проведення багатоваріантних розрахунків, комп'ютерні програми аналізу електронних схем особливо корисні: вони дозволяють автоматично протягом короткого часу провести аналіз багатьох варіантів. Модифікація схеми може проводитися також за допомогою спеціальних програм оптимізації на ПЕОМ (етап 12), у яких широко застосовуються методи оптимізації проектних рішень, заснованих на вирішенні задач математичного (лінійного та нелінійного) програмування. У цих завданнях проводиться пошук мінімуму чи максимуму певної цільової функції, яка залежить від багатьох змінних за наявності обмежень на ці змінні. При проектуванні РЕУ ця цільова функція відображає якість роботи, вартість апаратури та інші характеристики, що залежать від параметрів компонентів, оптимальні значення яких потрібно знайти в результаті виконання завдання. Обмеження формуються у вигляді системи співвідношень, що звужують допустиму область зміни параметрів компонентів при вирішенні задачі оптимізації РЕУ.

Після закінчення оптимізації можна розрахувати чутливість схеми, оцінити вплив розкиду параметрів компонентів та отримати інші важливі характеристики. Таким чином, за такого процесу проектування вирішуються завдання, пов'язані з розрахунком, аналізом та оптимізацією схемних рішень.

19

Що ж до завдання синтезу, це дуже складне завдання, її можна жорстко алгоритмізувати лише деяких окремих випадків, наприклад, існує методика класичного синтезу пасивних і активних аналогових і цифрових частотних фільтрів, класичного синтезу широкосмугових узгоджувальних пристроїв, синтезу цифрових автоматів. В інших випадках зазвичай завдання синтезу вирішується евристичним шляхом, ґрунтуючись на попередньому досвіді шляхом винахідництва.

Підкреслимо, що в процесі конструювання та розробки технології також може знадобитися корекція важливих схем, структури системи і навіть вихідних даних. Тому процес проектування не тільки багатоетапним, а й багаторазово коригованим у його виконання, тобто. процес має ітераційний характер.

Зазначимо, що автоматизоване проектування електронних схем за допомогою ПЕОМ має ряд переваг перед традиційним способом проектування "вручну" з подальшим доведенням на фізичному макеті. Розробник може використовувати можливості ПЕОМ у кількох областях. По-перше, за допомогою прикладних програм набагато легше спостерігати ефект варіювання параметрів схеми, ніж за допомогою суто експериментальних досліджень. По-друге, можна аналізувати критичні режими роботи пристрою без фізичного руйнування його компонентів. По-третє, програми аналізу дозволяють оцінити роботу схеми за найгіршого поєднання параметрів, що важко і не завжди можливо здійснити експериментально. По-четверте, програми дозволяють провести такі вимірювання на моделі електронної схеми, які важко виконати експериментально в лабораторії.

## **2.2. Поняття рівнів абстракції та ієрархії у проектуванні апаратури**

З часу першої появи цифрової мікросхеми у 1958 році розробники спостерігають експоненційне зростання числа транзисторів на інтегральній мікросхемі, що призводить до зростання проектів та їхнього стрімкого ускладнення. Для спрощення процесу проектування у проектах використовуються техніки, такі як абстракція та ієрархія. Ці техніки

20

працюють за принципом «розділяй і владарюй» і виявляються досить ефективними у рамках екстремально великих проєктів.

Функціональні можливості цифрових мікросхем можуть бути представлені за допомогою HDL різних рівнях абстракції. Найнижчим рівнем абстракції цифрових HDL є рівень транзисторних ключів, який визначає здатність описувати схему як таблиці транзисторних ключів. Вище за нього знаходиться рівень вентилів, який описує схему у вигляді таблиці з'єднань простих логічних вентилів та функцій. Обидва описані рівні абстракції можна віднести до структурного подання пристрою.

Наступний, більш складний рівень HDL визначається здатністю підтримувати логічні примітиви. Цей рівень називається функціональним чи RTL (Register Transfer Level), тобто, рівнем регістрових передач. Пристрій на цьому рівні абстракції представляється набором регістрів, пов'язаних між собою елементами комбінаційної логіки.

Найвищим рівнем абстракції вважається поведінковий, який підтримується сучасними версіями HDL і означає можливість описувати поведінку схеми, використовуючи абстрактні логічні структури, наприклад цикли і процеси. Цей рівень також передбачає використання у виразах алгоритмічних елементів, таких як суматори та помножувачі.

У рамках представлення апаратних блоків є системний рівень абстракції, що представляє набори алгоритмів у рамках великих блоків та їх з'єднання між собою. На жаль, цей підхід використовується тільки в моделюванні поведінки системи і поки що не придатний для розробки опису реального пристрою.

На малюнку 2.2 показано спрощене уявлення про функціонування апаратних блоків різних рівнях абстракції.

В даний час основним способом представлення апаратури є рівень RTL. Решта уявлень або є допоміжними, або згодом конвертуються в однозначний RTL-код.

Спочатку розглянемо поняття проекту в проектуванні цифрових пристроїв для FPGA. Проектом є набір файлів, що містять графічний опис блоків, описи блоків на мовах HDL, описи обмежень проекту, і т.д. Глобально можна сказати, що будь-який проект містить файли вихідного коду, що описують модулі та блоки проекту та ряд допоміжних файлів.

У межах сучасного підходу до проектування будь-який проект є ієрархічним, тобто. побудованим за модульним принципом (можна порівняти з матрьошкою). У проекті є деякий файл, що містить модуль верхнього рівня, який містить деяку логіку роботи і включає модулі нижчого рівня, які в свою чергу здійснюють ту ж послідовність дій-тобто. містять свою внутрішню логіку та підключають модулі нижчого рівня і т.д. На малюнку 2.3. представлено спрощену структуру проекту з урахуванням ієрархії.

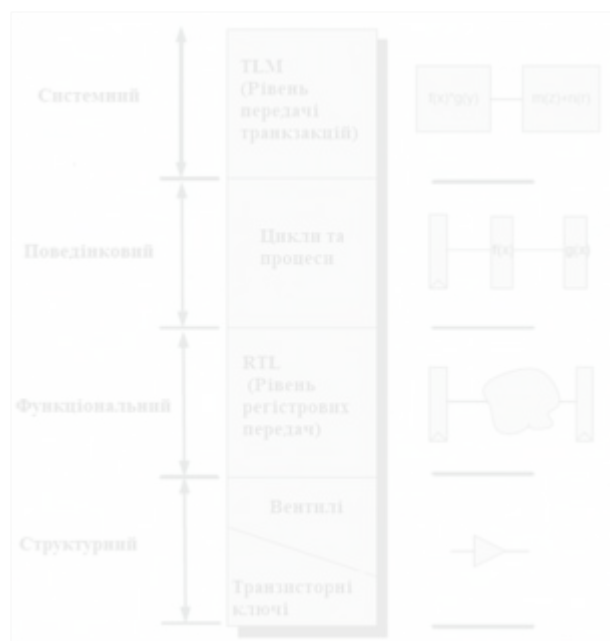


Рис. 2.2. Уявлення про функціонування апаратних блоків на різних рівнях абстракції

Спрощена схема маршруту проектування систем, заснованих на FPGA, з допомогою мов HDL лише на рівні абстракції RTL показано малюнку 2.4. У

лівій частині маршруту проектування представлені етапи перетворення текстового HDL опису (RTL code) конфігурацію пристрою рівня Celllevel, яка потім завантажується в цільову FPGA (device programming). У правій частині представлений процес валідації (перевірки правильності), який використовує тестове оточення (testbench) для перевірки, чи система задовольняє технічним вимогам (RTL verification) і вимогам продуктивності (static timing analysis).

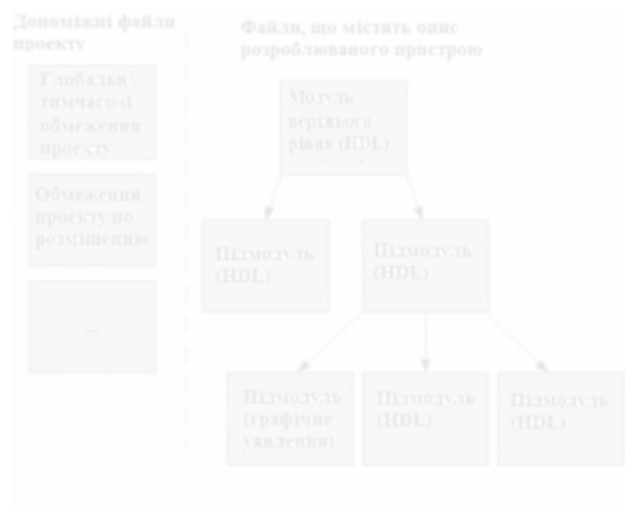


Рис. 2.3. Спрощена структура ієрархії проекту

Головні етапи у маршруті проектування:

1. Проектування системи та отримання HDL файлів. Додавання окремих файлів обмеження (constraints) визначення обмежень реалізації;
2. Створення тестового оточення (testbench) та здійснення моделювання RTL-код;
3. Виконання синтезу (synthesis) та імплементації (implementation). Процес синтезу переважно відомий як логічний синтез (logic synthesis). На цій стадії HDL файли перетворюються на списки з'єднань (netlist).

Процес імплементації складається з трьох ступенів: трансляції (translate), відображення (map), а також розміщення та розведення (place and route). Процес трансляції полягає у перекладі всіх списків з'єднань до єдиного формату та єдиного файлу, що залежить від засобів САПР. Процес, відомий



як відображення, відображає універсальні вентиля в логічні осередки, специфічні для певної цільової мікросхеми FPGA.

Процес розміщення та розведення, служить для отримання фізичного розміщення (топології) пристрою, що розробляється всередині мікросхеми FPGA. Цей процес розміщує осередки у фізичній області та визначає маршрути з'єднання різних сигналів. Статичний часовий аналіз (static timing analysis) виконується наприкінці процесу реалізації. Він визначає часові параметри з'єднань, такі як максимальну затримку розповсюдження сигналу та максимальну тактову частоту;

4. Створення та прошивка програмного файлу. На цьому етапі, згідно з остаточним поданням проекту після фізичного розміщення, створюється файл конфігурації. Цей файл служить для конфігурування логічних осередків та комутаційних матриць цільової FPGA.

Також, за потреби, можна виконати два додаткові кроки: зробити функціональне моделювання (functional simulation), яке може бути виконане після синтезу, а також тимчасове моделювання (timing simulation), яке виконується після етапу імплементації.

Функціональне моделювання використовує отриманий на етапі синтезу netlist для заміни RTL опису та перевірки коректності процесу синтезу. Тимчасове моделювання використовує отримане на етапі імплементації кінцеве уявлення пристрою, що розробляється поряд з детальними тимчасовими даними, також для здійснення перевірки на коректність.



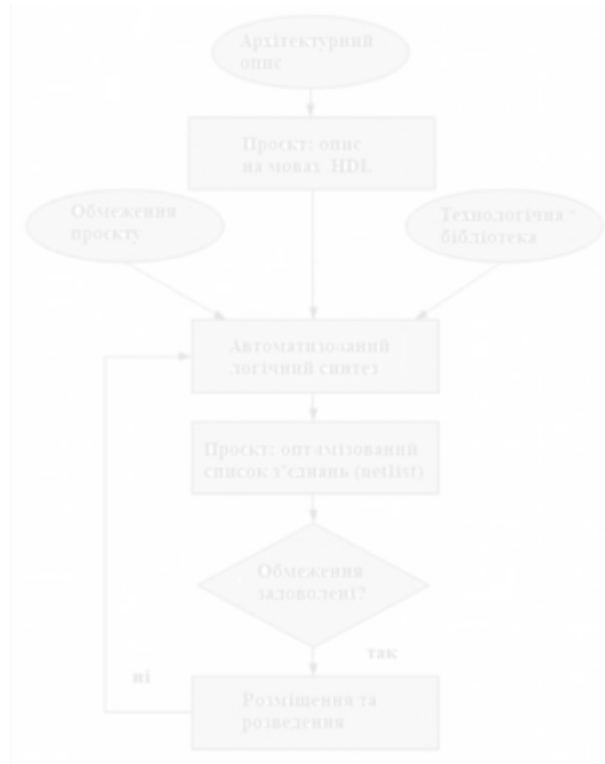
Рис. 2.4. Спрощений маршрут проектування для FPGA

### 2.3. Процес синтезу

У процесі розробки проектувальники описують поведінку апаратних блоків технологічно-незалежними високорівневими мовами HDL. Процес синтезу є процесом перекладу високорівневого опису розроблюваного пристрою в оптимізоване представлення вентильного рівня (netlist) за деяким алгоритмом, ґрунтуючись на обмеженнях проекту та обраній технологічній бібліотеці. Подібна технологічна бібліотека містить як базові логічні вентиля, так і макро-блоки, такі як суматори, мультиплексори, пам'ять та тригери. На перших етапах розвитку цифрової електроніки, розробники здійснювали логічний синтез своїми руками, аналізуючи проект і переводячи модулі, що розробляються, і блоки у вентильне уявлення, спираючись на деяку технологічну бібліотеку та обмеження проекту. На малюнку 2.5.a представлено блок-схему етапів ручного перекладу проекту до оптимізованого подання у вигляді набору вентилів.



A)



Б)

Рис.2.5. Процес логічного синтезу: а) без використання засобів автоматизованих систем проектування; б) з використанням засобів автоматизованого проектування

Нині процес логічного синтезу повністю автоматизованим. Це дозволяє розробникам зосередити свою увагу на проектуванні та використовувати у розробці високорівневі мови HDL. На малюнку 2.5.б представлено блок-схему етапів проектування з використанням автоматизованих систем проектування.

Процес імплементації є наступним кроком після процесу синтезу маршруті проектування. Для даного етапу можна виділити три характерні ступені:

- Трансляція (Translate);

- Відображення (Map);
- Розміщення та розведення (Place and Route).

У зв'язку з тим, що будь-який проект може складатися не тільки з безлічі файлів, що містять вихідний код опису модулів проекту на мовах HDL, але і містити безліч файлів, що є списками з'єднань (netlist), то потрібен певний етап, на якому буде здійснено конвертацію всіх списків з'єднань до одного формату. Цим етапом є процес трансляції. При об'єднанні всіх списків з'єднань в єдине ціле, на цьому етапі відбувається злиття з файлами, що описують обмеження проекту (design constraints). Також слід зазначити, що формат та структура результуючого файлу, отриманого на цьому етапі, є специфічною від фірми виробника програмного забезпечення.

На стадії відображення проект відображається на реально доступні ресурси, характерні для цільового кристала FPGA. У цьому процесі виявляються помилки використання недоступних чи відсутніх для цільового FPGA ресурсів.

Заключною стадією процесу імплементації проекту є стадія розміщення та розведення. На цьому етапі кошти автоматизованого проектування розміщують проект фізично. Уся логіка Проекту розміщується на конкретних фізичних структурах цільової FPGA, крім цього між логічними блоками прокладається трасування. На цьому етапі можна виявити помилки щодо переповнення цільової FPGA – тобто. виявити потреба у використанні більшої кількості ресурсів, ніж є. Також причиною помилок можуть стати ситуації, при виникненні яких, неможливість коректно поєднати деяку ділянку логіки із загальним проектом через брак з'єднань тощо.

Після завершення процесу імплементації та відсутності критичних попереджень та помилок, проект стає повністю готовим до заключного етапу, а саме до створення конфігураційного файлу для FPGA.

#### 2.4. Специфіка проектування цифрових пристроїв для FPGA з використанням коштів фірми Xilinx

Насамперед хотілося б представити огляд сімейств ПЛІС, що є в арсеналі цієї фірми. Кожна із сімейств, представлених нижче, має свою власну специфіку застосування та відповідні цільові ринки:

- Spartan - є масовими FPGA, що мають досить низьку вартість і призначені для вирішення завдань, що не потребують великих обчислювальних потужностей. На жаль, останнім представником даного сімейства є Spartan 6 і Xilinx планує завершити розвиток даного сімейства. На зміну йому прийдуть дві нові родини - Artix і Kintex;
- Artix — це сімейство орієнтоване на масові ринки недорогої продукції, має відносно низьку вартість та найнижче енергоспоживання порівняно з іншими сімействами, що мають однаковий порядковий номер і виконані за однаковою технологічною нормою. Також дане сімейство має малі габарити, що робить його в поєднанні з іншими параметрами дуже привабливим для побудови портативних пристроїв;
- Kintex - кристали цього типу мають великий обсяг пам'яті і розширені ресурси DSP, що робить це сімейство ідеальним для побудови LTE, світлодіодних і 3D цифрових відео дисплеїв, пристрій відображення для медицини та авіації;
- Virtex - являють собою найшвидші і ємніші FPGA, що містять найбагатшу периферію і найбільшу кількість додаткових блоків, таких як DSP-ядра, блокова пам'ять і т.д. Все це робить дану родину FPGA порівнянною з кристалами ASIC;
- CoolRunner – дані сімейства є реалізацією архітектури CPLD, що означає, що ці пристрої є невеликими енергонезалежними кристалами, що служать для організації невеликих, критичних до енергоспоживання цифрових схем.

Нові кристали з архітектурою FPGA від Xilinx спроектовані за технологією high-K metal gate та 28nm техпроцесу, що дозволяє при мінімальному енергоспоживанні досягти максимальної продуктивності.

Для покриття всього стандартного маршруту проектування для FPGA та CPLD фірмою Xilinx було розроблено два програмні пакети: ISE (Integrated Software Environment) та PlanAhead. Обидва ці пакети є закінчені середовища проектування зі зручним графічним інтерфейсом, що дозволяє викликати для різних етапів маршруту проектування спеціалізовані програми та утиліти. Вибір того чи іншого пакету є в основному справою смаку, але все ж таки слід враховувати, що пакет PlanAhead вважається більш професійним і має ряд більш розширених опцій.

На малюнку 2.6 представлено покриття всього маршруту проектування засобами Xilinx, а також на малюнку представлені розширення файлів проекту, створених на кожному з кроків.

Написання HDL може бути виконане як у вбудованому текстовому редакторі, так і в будь-якому іншому за бажанням розробника. На етапі синтезу можна використовувати як вбудований синтезатор, XST, і зовнішні синтезатори. Як зовнішні синтезатори, рекомендується використовувати Synplify або Precision.

- Процеси імплементації та створення конфігурації реалізовані засобами ISE у вигляді виклику набору відповідних утиліт. Протягом усіх етапів маршруту проектування користувач може контролювати правильність виконання кроків шляхом аналізу різноманітних звітів.



Рис.2.6. Покриття всього маршруту проектування для FPGA засобами Xilinx

## 2.5. Висновки до розділу

Виходячи з аналізу етапів схмотехнічного проектування цифрових пристроїв та специфіки проектування для FPGA з використанням засобів фірми Xilinx, можна зробити висновок, що автоматизоване проектування електронних схем за допомогою ПЕОМ має ряд переваг перед традиційним способом проектування:

- "вручну" з подальшим доведенням на фізичному макеті а саме за допомогою прикладних програм набагато легше спостерігати ефект варіювання параметрів схеми, ніж за допомогою суто експериментальних досліджень;
- є можливість аналізувати критичні режими роботи пристрою без фізичної руйнації його компонентів;



- програми аналізу дозволяють оцінити роботу схеми при найгіршому поєднанні параметрів, що важко і не завжди можливо здійснити експериментально;
- програми дозволяють провести такі вимірювання на моделі електронної схеми, які важко виконати експериментально в лабораторії;
- новітні кристали з архітектурою FPGA від Xilinx спроектовані за технологією high-K metal gate та 28нм техпроцесу, що дозволяє при мінімальному енергоспоживання досягти максимальної продуктивності.

## РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ МИСТЕЦЬКОГО НЕЙРОНУ ЯК ОСНОВНОГО ЄЛЕМЕНТА ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

### 3.1. Архітектура інструментального модуля Xilinx Spartan 3E Starter Board

Інструментальний комплект Spartan-3E Starter Kit призначений в першу чергу для практичного вивчення методів проектування цифрових пристроїв з апаратною реалізацією операцій та розробки вбудованих мікропроцесорних систем на основі ПЛІС передових сімейств FPGA фірми Xilinx. Унікальні функціональні можливості, технічні характеристики та конструктивне виконання інструментального модуля, що входить до цього комплекту, дозволяють не тільки виконувати налагодження проектів, що реалізуються на основі ПЛІС сімейства Spartan-3E, але й використовувати його як промисловий серійний варіант пристрою, що розробляється.

Архітектура інструментального модуля Xilinx Spartan-3E Starter Board дозволяє використовувати його для реалізації автономних систем управління, збору та обробки інформації, цифрової обробки сигналів, цифрових пристроїв з різними комп'ютерними інтерфейсами. Крім того, цей модуль можна рекомендувати для застосування в навчальних лабораторіях вузів для вивчення сучасних методів проектування цифрових пристроїв, мікропроцесорних систем з різною архітектурою, а також цифрових пристроїв обробки сигналів. Ресурси інструментального модуля Xilinx Spartan-3E Starter Board дозволяють реалізувати проекти мікропроцесорних систем, що вбудовуються, що виконуються на основі як 8-розрядних ядер сімейства PicoBlaze, так і 32-розрядних ядер сімейства MicroBlaze. Декілька типових проектів, спеціально розроблених для аналізованого інструментального комплекту, наочно демонструють не тільки можливості налагоджувальної плати, але і представляють можливі варіанти конфігурації вбудованих систем, створюваних на базі цих мікропроцесорних ядер.

До складу комплекту Spartan-3E Starter Kit входять:

- плата інструментального модуля Xilinx Spartan-3E Starter Board;

33

- мережевий адаптер з вихідною стабілізованою напругою 5 В та аксимальним струмом навантаження 2,5 А, що використовується як первинне джерело живлення інструментального модуля;
- стандартний USB-кабель Type A/Type B, який підключається до відповідного порту комп'ютера та призначений для конфігурування ПЛІС та програмування конфігураційного ППЗУ, встановлених на платі інструментального модуля Xilinx Spartan-3E Starter Board, за допомогою інтегрованої схеми завантажувального кабелю;
- комплект CD-ROM, що містить нову повнофункціональну версію системи проектування ISE™ (Integrated Software Environment/Integrated Synthesis Environment) WebPACK™, оціночні версії САПР ISE Foundation™ та засобів розробки вбудованих мікропроцесорних систем Xilinx Embedded Development Kit™ (EDK) протягом 60 днів з моменту встановлення, а також Spartan-3/3E Starter Kit Resource CD.

Інструментальний модуль Xilinx Spartan-3E Starter Board характеризується такими відмінними рисами:

- використання в якості основного компонента ПЛІС з сімейства Spartan-3E з об'ємом 500 000 системних вентилів (10 476 логічних осередків) в корпусі FG320, об'єм логічних і трасувальних ресурсів якої в поєднанні з великою кількістю користувальницьких висновків дозволяють реалізувати не тільки окремі функціональні , але систему, що розробляється в повному обсязі;
- можливість підтримки на рівні проектів стандартних інтерфейсів обчислювальних систем RS-232, PS/2 та VGA, що дозволяє підключати до інструментального модуля зовнішні пристрої з послідовним інтерфейсом, клавіатуру/мишу та дисплей відповідно;

34

- застосування у схемі модуля перетворювача рівнів RS-232, що забезпечує можливість безпосереднього підключення до послідовного порту зовнішніх пристроїв різного типу через роз'єми DB-9 (типу DTE та DCE), встановлені на платі (при реалізації універсального асинхронного приймача UART на основі ресурсів ПЛІС);
- наявність додаткових компонентів, що реалізують фізичний рівень інтерфейсу 10/100 Ethernet PHY, що дозволяє підключати інструментальний модуль через стандартний роз'єм безпосередньо до відповідної мережі при реалізації контролера Ethernet MAC у складі проекту, що завантажується в кристал сімейства Spartan-3E;
- підключення спеціальних та користувальницьких висновків ПЛІС до контактів чотирьох роз'ємів розширення, що відповідають різним стандартам, що забезпечує можливість гнучкого сполучення із зовнішніми компонентами та пристроями;
- застосування зовнішнього високошвидкісного синхронного динамічного ОЗП з подвоєною швидкістю передачі даних, виконаного у вигляді DDR SDRAM ємністю 64 Мбайт (512 Мбіт), що розширює можливості оперативної пам'яті вбудованих систем, що реалізується на основі відповідних ресурсів ПЛІС;
- використання як стандартного ППЗУ для зберігання конфігураційних даних ПЛІС Flash-пам'яті серії Platform Flash, що програмується в системі, об'ємом 4 Мбіт;
- включення до схеми інструментального модуля додаткової ПЛІС із архітектурою CPLD XC2C64A сімейства CoolRunner-II [11-14], що використовується, зокрема, для комутації конфігураційної пам'яті різного типу та управління режимами конфігурування основного кристала (серії Spartan-3E);

- інтегрована схема завантажувального кабелю, що дозволяє виконувати конфігурування всіх ПЛІС та програмування конфігураційної пам'яті, що входять до складу інструментального модуля, за допомогою стандартного кабелю, що підключається до порту USB комп'ютера;
- наявність додаткового спеціального роз'єму для підключення стандартних завантажувальних кабелів різного типу, що дозволяють виконувати операції конфігурування ПЛІС та програмування ППЗУ у різних режимах, а також зворотного зчитування конфігураційних даних через порт JTAG-інтерфейсу;
- використання у складі інструментального модуля паралельної NOR Flash-пам'яті ємністю 16 Мбайт (128 Мбіт), яка може використовуватися, зокрема, для зберігання конфігураційної послідовності даних ПЛІС сімейства Spartan-3E або програмного коду вбудованої мікропроцесорної системи, що виконується на базі 32-розрядного ядра сімейства MicroBlaze;
- наявність послідовної Flash-пам'яті об'ємом 16 Мбіт з інтерфейсом SPI (Serial Peripheral Interface), що використовується для запису конфігураційної інформації кристала сімейства Spartan-3E або програмного коду, що вбудованої мікропроцесорної системи, що виконується на базі 32-розрядного ядра сімейства MicroBlaze;
- включення до складу схеми інструментального модуля послідовного ППЗУ EEPROM, який підтримує криптографічний алгоритм Secure Hash Algorithm (SHA-1) та призначений для захисту конфігураційних даних від несанкціонованого копіювання;
- застосування двоканального аналого-цифрового перетворювача ADC (Analog-to-Digital Converter) з інтерфейсом управління SPI,

що забезпечує можливість реалізації пристроїв цифрової обробки сигналів;

- наявність чотириканального послідовного цифро-аналогового перетворювача DAC (Digital-to-Analog Converter) з 12-розрядною роздільною здатністю, керованого за допомогою інтерфейсу SPI;
- застосування кварцового генератора із частотою 50 МГц, призначеного для формування основного тактового сигналу для ПЛІС;
- наявність панелі для встановлення додаткового кварцового генератора, що використовується як альтернативне або додаткове джерело сигналу синхронізації;
- присутність на платі елементів індикації різних типів, що забезпечують можливість візуального контролю напруги живлення, процесу конфігурування кристала та функціонування системи, що розробляється;
- наявність чотирьох повзункових перемикачів, чотирьох кнопок і поворотного перемикача, поєднаного з кнопкою, які можуть використовуватися, наприклад, для ручної установки режиму роботи реалізованої системи або в процесі налагодження проекрованої системи, а також для тестування інструментального модуля та прикладного програмного забезпечення;
- використання комплексної схеми управління живленням, що виконує функції формування напруг, необхідних для живлення компонентів модуля, у тому числі для блоків введення/виводу та ядра кристала FPGA, конфігураційного ППЗП, елементів оперативної та постійної пам'яті, ЦАП та АЦП, інтегрованої схеми завантажувального кабелю;
- наявність кнопки, що забезпечує реалізацію режиму примусового завантаження послідовності конфігурації в основну ПЛІС;

- повна сумісність з усім сімейством систем проектування та програмування кристалів фірми Xilinx (ISE WebPACK та ISE Foundation) та підтримка засобами розробки вбудованих мікропроцесорних систем Xilinx EDK. Зовнішній вигляд інструментального модуля подано на рис 3.2.



Рис. 3.1. Зовнішній вигляд інструментальний модуль Xilinx Spartan-3E

Усі компоненти модуля змонтовані на друкованій платі із двостороннім розміщенням компонентів. Структурне уявлення архітектури модуля, що розглядається, показано на рис. 3.2. Основними елементами архітектури модуля Xilinx Spartan-3E Starter Board є:

- головна ПЛІС XC3S500E сімейства Spartan-3E у корпусі FG320, на основі якої реалізується проектована система;
- програмоване в системі ППЗУ серії Platform Flash XCF04S, призначене для зберігання конфігураційних даних ПЛІС XC3S500E;
- блок завантаження конфігураційних даних;
- схема управління конфігуруванням ПЛІС;
- допоміжна ПЛІС CPLD XC2C64A сімейства CoolRunner-II;
- послідовне ППЗУ EEPROM, піддер SHA-1;

- блок синхронізації, призначений для формування зовнішніх (стосовно ПЛІС) тактових сигналів;
- зовнішнє високошвидкісне ОЗП;
- вузол двоканального аналого-цифрового перетворювача (АЦП);
- вузол чотириканального цифро-аналогового перетворювача (ЦАП);
- модуль паралельної NOR Flash-пам'яті ємністю 16 Мбайт (128 Мбіт);
- модуль послідовної Flash-пам'яті об'ємом 16 Мбіт з інтерфейсом SPI;
- схема формування та контролю живильних напруг;
- блок світлодіодних індикаторів;
- дворядковий шістнадцятизначний рідкокристалічний дисплей;
- блок повзункових перемикачів;
- блок кнопочових перемикачів, з поєднаним поворотним перемикачем;
- схема перетворення рівнів сигналів інтерфейсу RS-232;
- модуль фізичного рівня інтерфейсу 10/100 Ethernet PHY;
- стандартні роз'єми інтерфейсів RS-232, PS/2, VGA та Ethernet;
- чотири роз'єми розширення.





Рис. 3.2. Структурне представлення архітектури інструментального модуля Xilinx Spartan 3E Starter Kit

Тип кристалу сімейства Spartan-3E, що використовується як основна ПЛІС розглянутого інструментального модуля, значною мірою визначає функціональні можливості останнього

Виходячи з того, що вихідна напруга інструментального модуля Xilinx Spartan#3E становить 3,3 а також для того щоб виключити увагу вплив силових схем необхідно розробити блок опторозв'язки.

### 3.2. Реалізація штучного нейрона

Використання вагів дуже важливе і пов'язане з вагами синапсів і може збільшити або зменшити сигнали, які надходять на синапс. Як уже згадувалося раніше, ми визначили 3 змінними вагами як вхідні дані, кожна вага пов'язана з дендриту, щоб глобальній нейронній мережі дозволити здатність навчання.

На першому погляді наша система матиме:

-3 входи що називають дендритами від інших нейронів.

-1 Вихід називають аксона. Коли вихід не буде використаний, доведеться генерувати імпульс для поширення його на інші нейрони через аксон.



Рис.3.3. Нейрон з 3 входами

У більш глибокому аналізі, ми повинні включати інші блоки.

Для іншої сторони також необхідно ввести сигнал годинника в системі. Всі сигнали будуть цифрові сигнали, для того, щоб сигнал синхронізував Глобальну систему.

- 3 входи називають вагами зв'язку між іншими нейронами та нашим нейроном. Кожна вага у зв'язку з кожним дендритом.

- 1 вхід годинника. Необхідний тактовий сигнал, для синхронності системи



Рис. 3.4. Нейрон із входами ваг та входом синхросигналу

Коли сплеск прибуває, має в основному дві функції, щоб генерувати потенціал дії відповідно до вхідного і порівняти, якщо додавання всіх можливих дій в цей момент часу знаходиться над порогом, в цьому випадку це буде генерувати імпульс через аксона.

Ми розглядатимемо ці дії як:

- Потенціал-генератор. Коли отриманий імпульс починає збільшувати потенціал.
- Додавання та порівняння. Додати всі можливості та порівняти з порогом.



Рис.3.5. Функції нейрона

Наступний блок постійно робить додавання всіх потенціалів та порівняння результатів з порога, визначеного 20 мВ. Якщо позитивний результат то посилає імпульс, що вказує, що нейрон збуджується. Коли додавання всіх потенціалів не перевищує поріг - вихід '0' логіка. Цей блок може бути реалізований тільки в блоці, що робить додавання та порівняння.

З іншого боку, є фактор, який ми ще не розглянули. Це вогнетривкий час. Коли нейрон збуджується протягом періоду часу, він не може генерувати інший сплеск, цей період часу - рефрактерний період і має тривалість близько 2 мс. Щоб вирішити цю проблему, ми використовували інший блок імені заборони блоку. Цей блок працює як таймер. Коли нейрон збуджується, він посилає сигнал до потенційних генераторів інгібувати та встановити будь-який рахунок. У той же час він починає відлік, поки таймер не більше 2 мс, то продовжує посилати сигнал заборони. Коли лічильник перевантажено, він вимикає сигнал заборони. Потенціал генератора здатний генерувати потенціал, коли імпульс надходить знову.

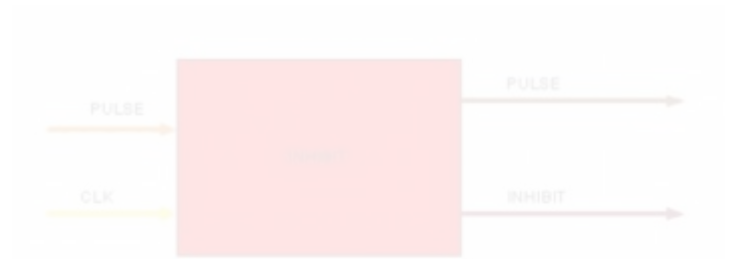


Рис. 3.6. Блок блокування

Ми повинні створити новий блок, інгібують блок, і змінювати потенційну генератор додавання його новий вхід, що дозволяє вхід, якщо цей вхід активний таймер не рахувати і вихід альфа блоку завжди буде "0" логіка.



Рис. 3.7. Генератор потенціалу із блокуванням сигналу

Для того щоб створити штучний нейрон, необхідно реалізувати окремі його частини.

### Таймер

#### Вимоги:

Нам потрібна система, що коли вона виявляє імпульс (зростання краю), починає відлік реального часу. Він також матиме дозвіл входу, якщо цей вхід активний, таймер зупинить суперника, кількість буде скинуто. Коли цей сигнал буде неактивним, таймер зможуть розраховувати знову коли отримаєте іншу наростання краю.

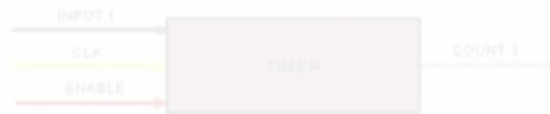


Рис.3.8. Таймер

Inputs: Input1: External input from others neurons

CLK: External input, used to synchronize the count.

Enable: Internal input, from the inhibit block.

Outputs: Count1: Internal output, to alpha block.

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
use IEEE.math_real.all;
entity counters is port(
  clk: in std_logic;
  input1: in std_logic;
  enable: in std_logic;
  count1: buffer real:=0.0
);
end counters;
architecture archicontador of counters is
begin
  process (input1, clk)
    variable pulso_rec: real :=0.0;
  begin
    if (enable = '1') then
      count1 <= 0.0;
      pulso_rec :=0.0;
    else
      if (input1'event and input1='1')
        then
          count1 <= 0.0;
          pulso_rec := 1.0;
        end if;
        if ((clk'event and clk= '1') and
          pulso_rec= 1.0) then
          count1 <= count1 + 0.1;
        end if;
      end if;
    end process;
  end archicontador;
end archicontador;

```

Поведінка блоку така. Програма перевіряє, якщо сигнал, що дозволяє, активний, в цьому випадку, він блокує систему і скидає лічильник. Інакше система починає перевіряти, якщо новий імпульс надходить по input1. У цьому випадку лічильник скидається і показує систему, яка була оброблена імпульсом. Після цього лічильник буде збільшуватися своє значення до іншого імпульсу не надходить або включити сигнал був активним знову.



Рис. 3.9. Симуляція таймера

Цей блок повинен бути в змозі використовувати результат підрахунку, отримати можливі дії сліднують за альфа функції  $F(X) = X * e^{-x}$ , де "x" в режимі реального часу, E.I., результат таймера Лічильник. В результаті необхідно також набувати значення ваги цього входу в розгляд.

- Inputs: A: Internal input, з Timer block.
- Weight1: External input associated to the input from others neurons.
- RESULT\_A: Internal input to Addition/Comparison block



Рис. 3.10. Розрахунковий блок

<pre>library IEEE; use IEEE.std_logic_1164.all; use IEEE.std_logic_arith.all; use IEEE.std_logic_unsigned.all; use IEEE.math_real.all; entity alpha is port(A:in real :=3.0;</pre>	<pre>RESULTADO: out real:=0.0); end entity alpha; architecture behavioral of alpha is begin RESULTADO &lt;= exp(-A)*A*1.0; end architecture behavioral;</pre>
--	---

Цей блок повинен бути здатний приймати результат із можливих дій з альфа-блоків, так і в кожен момент часу, щоб додати всі ці потенціали. У той же час система повинна взяти до уваги порівняння між значенням того і порога. Поріг рівня, що визначає, коли нейрон збуджується, закріплений на 20 мВ. Коли результат додавання перевищує поріг, то система вміщує вихід в один.



Рис.3.11. Блок підсумовування та порівняння

Inputs: A1: Internal input, від Alpha blok of dentrite1.

B1: Internal input, від Alpha block of dentrite2.

C1: Internal input, від Alpha block of dentrite3.

SAL: Internal output, до Inhibit block.

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
entity threshold is
port(
A1:in real :=1.0;
B1:in real :=2.0;
C1:in real :=0.0;
SAL: out std_logic);
end entity threshold;
architecture behavioral of threshold is
begin
SAL<='1' when ((A1+B1)+C1)>1.2
else '0';
end architecture behavioral;

```

Цей блок має генерувати сигнал, який керує рефрактерний період. Коли він отримує сигнал, повинен генерувати інший сигнал тривалістю де 2 мс (вогнетривкий час).

Inputs: CLK: External input, до synchronize the system.

pul\_sal: Internal input, від addition/comparison block.

Enable: Internal output, до timer blocks.

PULSE: External output, pulse generated.

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;
use IEEE.math_real.all;
begin
process (pul_sal, clk)
begin
PULSE <= pul_sal;
if (pul_sal'event and pul_sal='1') then

```

46

```
entity contador_inhib is port(
  clk: in std_logic;
  pul_sal: in std_logic;
  conta: buffer real:=0.0;
  enable: out std_logic;
  PULSE: out std_logic
);
end contador_inhib;
architecture archicontador of
  contador_inhib is
    conta <= 0.0;
    enable<= '1';
    else
      if (clk'event and clk='1') then
        conta <= conta + 0.1;
        if (conta > 0.8) then
          enable <= '0';
        end if;
      end if;
    end if;
  end if;
end process;
end archicontador;
```

Поведінка цього блоку є такою. Коли отримує наростання краю, він скидає лічильник, активами сигнал дозволу і починає відлік. Дозволяючий сигнал буде активний доти, лічильник таймера не прибуває в 2 мс. На даний момент відключається сигнал активації. Коли він отримує інший наростання краю системи будуть робити те ж саме, скидання лічильника, активний включити та запустити відлік.

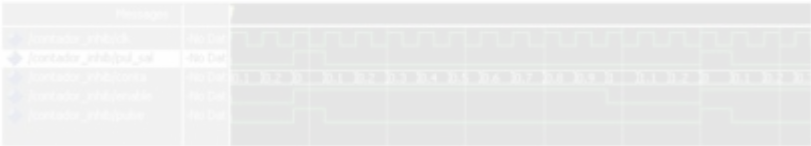


Рис.3.12. Симуляція блоку заборони

Складання



Рис.3.13. Загальна схема нейрона



### **3.3. Висновки до розділу**

Виходячи з параметрів інструментального модуля Xilinx Spartan-3E встановлено доцільність використання останнього як апаратної частини проекту, що розробляється.

Запропоновано та реалізовано мовою проектування пристроїв VHDL схема штучного нейрона, що складається з блоків блокування, генераторів потенціалу, розрахункових блоків та блоку підсумовування та порівняння.

Наведена схема дозволяє отримати електронну систему, яка відтворює поведінку біологічного нейрона.

## ВИСНОВКИ

В результаті проведених досліджень, присвячених вирішенню проблеми розробки штучного нейрона за рахунок визначення типу штучного з формою подання інформації в яких вхідні, вихідні сигнали і вагові коефіцієнти, зв'язки подаються у вигляді цифрових кодів при цьому найдоцільніше як апаратну частину проекту використовувати програмовані інтегральні схеми ( ПЛІС).

Виходячи з аналізу етапів схемотехнічного проектування цифрових пристроїв та специфіки проектування для FPGA з використанням засобів фірми Xilinx, можна зробити висновок, що необхідно використовувати системи проектування прийняті для ПЛІС фірми Xilinx а саме обраного нами інструментального модуля Xilinx Spartan-3E.

Запропоновано та реалізовано мовою проектування пристроїв VHDL схема штучного нейрона, що складається з блоків блокування, генераторів потенціалу, розрахункових блоків та блоку підсумовування та порівняння.

Наведена схема дозволяє отримати електронну систему, яка відтворює поведінку біологічного нейрона.

## Matches

Internet sources

13

1	<a href="https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/4676/280.pdf">https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/4676/280.pdf</a>	2 Sources	13.9%
2	<a href="http://hdl.handle.net/10251/31691">http://hdl.handle.net/10251/31691</a>	8 Sources	1.21%
3	<a href="https://ela.kpi.ua/bitstream/123456789/3714/1/55_23.pdf">https://ela.kpi.ua/bitstream/123456789/3714/1/55_23.pdf</a>		0.37%
4	<a href="https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/18339/Aylas_ba.pdf?isAllowed=y&amp;sequence=3">https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/18339/Aylas_ba.pdf?isAllowed=y&amp;sequence=3</a>	2 Sources	0.1%