



THE IMPLEMENTATION OF THE NEURAL NETWORK FOR THE CLASSIFICATION PROBLEM

Ulyana Lisovik, Oleksandr Lipchanskiy

Kharkiv National University of Radioelectronics,
Department of computing automation and design,
Lenin Prospect 14, Kharkiv 61726, Ukraine

Abstract: *The example of NN realization is considered. Also description of all its design stages from NN function model description to its timing and hardware characteristics estimation is considered. NN structural model is presented in VHDL code. Through SynplifyPro 7.0 package from Synplicity® the system synthesis with the orientation on Virtex-II XC2V6000 family is made out. The estimation of the optimality of the synthesized NN model utilization is accomplished. NN structures are shown; hardware costs are taken to the table.*

INTRODUCTION

Hardware realization of artificial neural networks (NN) aims at efficiently accelerating the processing speed of neural network specific tasks. Currently the new NN designs demands decreasing in hardware costs and terms of development, moreover them need to have capability for further modification during exploitation.

Using of Field programmable gate arrays (FPGAs) can reduce the concept-to-product time and the cost of making modifications. Recent FPGA have computational resources on the order of hundreds of thousands or millions of gate equivalents, can be reprogrammed indefinitely and have in-circuit and partial reconfiguration possibilities.

The main stages of NN design have been carried through using the task of creation such NN, which can forecast a possibility of delay in adolescent sexual development. In the given paper results of medicogenetic checkup of adolescents with sexual development delay diagnosis and without this diagnosis have been used as the learning sample.

The NN have been created by using NeuroPro 0.25 software. Since now most of NN hardware realizations do not allow on-chip learning, therefore NN representation in a high-level hardware description language (VHDL) use weight coefficient values, obtained during NN training using the NeuroPro v 0.25 software.

During training in NeuroPro v 0.25 software the non-linear sigmoid function has been used as neuron activation function and back-propagation algorithm

as learning algorithm. It is advisable to use this algorithm for training fully connected feed-forward networks. The learning rule is based on gradient descent.

NN PARAMETERS

The NN consists of three layers. The input layer has four neurons fed concurrently by information from the test sample. Each neuron has a different number of synapses, depending on the number of characters received by the neuron. The second hidden layer consists of three neurons, which process data from the first layer. Each neuron from the second layer has four input lines, which present neurons' synapses, and the bias signal. The third layer is a summing layer, which made of two elements that sum outputs from the previous layer.

The weights matrix is described as one-dimensional massive. Massive elements are weight coefficients of each neuron's synapse. Each NN's neuron has its own weights matrix.

IMPLEMENTATION PROBLEMS

The NN model must be designed in a such way that a further synthesis to the logic gates should be possible. And even if VHDL representation has correct behavioral simulation results, it doesn't means that it will be synthesized correctly and the end realization will be optimal.

So the main problems of NN hardware implementation were:

- representation of floating point numbers (weights' values have size of changing from -1 to 1);
- implementation of the non-linear activation function;

The arithmetical operations on floating point numbers are more sophisticated than operations on integer numbers, therefore special-purpose schemes for operating on floating point numbers are more complicated, works slowly and occupy a lot of chip resources. Also numbers of real type are not synthesized. Therefore in the paper we passed to the integer representation of floating point numbers.

It is a problem to implement the non-linear activation function in a digital hardware. Some activation functions, for example, such as sigmoid function, need to be modified to simplify its designing in hardware. In the given case the sigmoid function has been replaced by piecewise-linear function.

DEVELOPED DEVICE

The developed device consists of the controller and NN. The control block works on 20 MHz system rate and sends necessary signals for NN controlling. It synchronizes each layer functioning and reports when the next test sample can be supplied to NNs' inputs.

NN consists of neuron components, which organized in layers. Each neuron component contains certain number of synapses, one adder and activation function. The bus that responsible for input signals feeding and another input buses have eight bit width, because the device doesn't demand of great precision.

While working on the design it was clarified that synthesized scheme completely depends on the method of describing device functioning. All device parts were designed and functionally validated in VHDL, to be synthesized and integrated as standard cells by a placement and routing software tool.

For the simulation of the NN functionality the Aldec's Active-HDL 5.0. was used. The NN behavioral and post-synthesis simulation results are same that signifies about correct NN synthesis stage.

CONCLUIONS

In the paper the estimation of obtained NN optimality have been given. In the NN design the main characteristics of the activation function and requirements for FPGA's devices are taken into account.

Using the described in the paper design methodology allows to decrease terms of NN realization and optimize on the FPGA and logic gates resources.



АПАРАТНА РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖІ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ

Уляна Лісовик, Олександр Ліпчанський

Харківський національний університет радіоелектроніки,
Кафедра автоматизації та проектування обчислювальної техніки,
пр. Леніна, 14, Харків 61726, Україна

Резюме: В статті приведена методологія проектування нейромереж (НМ) для вирішення задачі класифікації. За допомогою пакета NeuroPro V 0.25 проведено навчання НМ. Алгоритм функціонування НМ описано мовою VHDL з урахуванням оптимальних параметрів НМ. Розроблена модель НМ, перевірена правильність функціонування на тестових прикладах. За допомогою пакета SynplifyPro 7.0 від Synplicity® зроблено синтез пристрою з орієнтацією на мікросхему сімейства Virtex-II XC2V6000. Приведені структури НМ та апаратурні витрати.

Ключові Слова: Нейромережа, FPGA, Прогнозування, Вибірка даних, Алгоритм зворотного поширення, Функціональна модель, Синапс, Активаційна функція, Кусково-лінійна функція, Навчальна вибірка

1. АКТУАЛЬНІСТЬ І ПОСТАНОВКА ЗАДАЧІ ПРОЕКТУВАННЯ

Питання зменшення вартості і термінів розробки, зниження апаратних витрат при схемній реалізації НМ, а також можливість подальшої модифікації у процесі експлуатації, є актуальною проблемою проектування штучних НМ.

Схемна реалізація НМ забезпечує максимально можливу швидкодію при вирішенні задачі класифікації. Це досягається за рахунок переваги принципу паралелізму у функціонуванні елементів НМ.

Використання сучасних перепрограмувальних користувачем базових матричних кристалів (FPGA) для реалізації НМ можна вважати альтернативою стосовно тільки програмного або апаратного варіантів. Сучасні FPGA мають значний обсяг ресурсів – до 10 млн. системних вентилів на кристал, високу продуктивність із системними частотами до 420 Мгц, а також можливість реконфігурації кристала безпосередньо в процесі функціонування [1].

В останнє десятиліття НМ знайшли широке застосування в медичній інформатиці для розв'язання різних задач класифікації й прогнозування завдяки їхній здатності

ефективно будувати нелінійні регресійні залежності [2]. На прикладі задачі створення НМ, здатної прогнозувати можливість затримки статевого розвитку підлітків, були проведені головні етапи розробки НМ. Для рішення задачі класифікації та прогнозування НМ повинна бути навчена на вибірці даних, утримуючих «знання» про прогнозований об'єкт. У даній роботі як навчальна вибірка використовувалися результати медико-генетичного обстеження підлітків з діагнозом – затримка статевого розвитку та підлітків з нормальним процесом статевого розвитку.

Основними об'єктами такого обстеження є малюнок шкіри долонь людини [3]. Для медико-генетичного обстеження використовувалася методика аналізу дерматогліфічних відбитків по Cummins [3] з обліком уніфікованої Лондонської класифікації по дерматогліфіці. Факт, що у хворих із затримкою статевого розвитку (ЗСР) існує специфічний дерматогліфічний малюнок, був установлений с. н. с. Інституту охорони здоров'я дітей і підлітків кандидатом біологічних наук Багацкой Н.В. і опублікований нею в ряді статей [4].

Для прогнозування ЗСР була застосована методика навчання комп'ютерних НМ у середовищі програми NeuroPro v 0.25, створеної співробітниками групи "НейроКомп" лабораторії

моделювання нерівноважних систем ОЦК СВ РАН м. Красноярськ [5,6].

У ході даної роботи за допомогою пакета NeuroPro 0.25 була побудована НМ. Після з'ясування оптимальної архітектури НМ, кількості нейронних елементів і вагових коефіцієнтів кожного з нейронів, НМ була навчена згідно алгоритму зворотного поширення. Далі була перевірена правильність її функціонування на тестових прикладах.

На даний час більшість апаратурних реалізацій НМ не дозволяє здійснювати навчання мережі в реальному часі (on-chip learning). Тому при описі моделі НМ мовою VHDL були враховані значення вагових коефіцієнтів, отримані при навчанні за допомогою пакета NeuroPro v 0.25.

Стаття містить приклад реалізації НМ і опис етапів її розробки, починаючи від функціонального опису моделі НМ до оцінки її часових і апаратурних характеристик. Приведено оцінку оптимальності використання моделі НМ, отриманої після опису використаного нейроалгоритма, у коді VHDL. Модель описана з урахуванням характерних особливостей активаційної функції і вимог, пропонує до пристроїв для реалізації на FPGA.

2. РОЗРОБКА ФУНКЦІОНАЛЬНОЇ МОДЕЛІ НМ

2.1 Навчальна і тестова вибірки

У роботі використані результати медико-генетичного обстеження підлітків з діагнозом – ЗСР (270 чоловік) і підлітків з нормальним процесом статевого розвитку (90 чол), що розглядаються як два вихідних класи на множинах 0 – «здорові», 1 – «передбачувані хворі». Інформація сконцентрована в 28 стовпцях (по кількості дерматогліфіческих маркерів) і 360 рядках електронної таблиці формату *.dbf. У кожному рядку знаходиться інформація про підлітка, отримана з історії хвороби. Стовпці містять дерматогліфічески маркери, що свідчать про можливий ЗСР. У ході роботи дані були розділені на тестову й навчальну вибірку.

Кількість векторів ознак, що складають тестову і навчальну вибірку, представлені в таблиці 1.

Таблиця 1

	Навчальна вибірка, кількість	Тестова вибірка, кількість
Чол. зі ЗСР	220	50
Чол. без ЗСР	80	10

2.2 Вибір величини характеристики нейронів

Як активаційну функцію штучного нейрона використовувалася нелінійна сигмоїдна функція:

$$f(x)=x/(C+|x|),$$

де x - вихід суматора нейрона, а константа C – характеристика сигмоїди. З формули видно, що величина сигналу завжди зменшується, причому, чим більше характеристика нейрона, тим істотніше зменшення сигналу [2].

Ефективно починати процес навчання НМ з невеликим значенням C , а при успішному навчанні генерувати і навчати нові мережі, збільшуючи цей параметр. Рекомендуємо діапазон значень характеристики від 0,1 до 0,8 [7]. У даній роботі ми використовували мережу з характеристикою $C = 0,1$.

2.3 Нормування вхідних сигналів

У результаті проведених [2] експериментів було встановлено, що найбільш універсальна і швидко навчаєма архітектура повнозв'язної сигмоїдної НМ оптимально працює при значеннях вхідних сигналів у діапазоні [-1;1]. При розширенні цього діапазону відбувається спочатку практично непомітне, плавне, а потім швидко наростаюче зниження якості навчання, при різних діапазонах прикладів навчальної вибірки.

У роботі автоматичне нормування вхідних сигналів на діапазон [-1;1] реалізовано в середовищі NeuroPro v 0.25.

2.4 Вибір архітектури мережі і числа штучних нейронів

Число нейронів у шарах може бути різним і не залежати від числа вхідних і вихідних сигналів мережі. Цей параметр визначає сумарна кількість синаптичних зв'язків НМ.

Оптимальна кількість нейронів багато в чому залежить від поставленої задачі. Більша кількість нейронів підвищує гарантію успішного навчання, але збільшує розмір НМ, а також збільшує час її спрацювання при тесті і завантаження в пам'ять. У загальному випадку, число нейронів дорівнює числу навчальних параметрів. Пропонують використовувати кількість нейронів рівне сумі числа навчальних параметрів і кількості класів у задачі [5].

Пакет NeuroPro v 0.25 дозволяє будувати повнозв'язні багат шарові мережі без зворотного зв'язку (feedforward networks) і визначати оптимальне число нейронів у мережі.

2.5 Навчання

Для навчання застосовано наявний в арсеналі NeuroPro v 0.25 алгоритм зворотного поширення (back-propagation)[5]. Цей алгоритм доцільно використовувати для навчання повнозв'язних мереж прямого поширення. Оптимізація проводиться по методу сполучених градієнтів[7].

Кількість циклів навчання, також як і час, що вимагається для повного навчання, залежать від таких факторів: величини навчальної вибірки, кількості вхідних параметрів, виду задачі, типу й параметрів НМ і навіть від випадкового розкладу ваг синапсів при ініціалізації мережі.

Швидкість навчання НМ багато в чому залежить від необхідної точності рішення задачі. Так, не слід вимагати від НМ точності, що перевищує точність виміру даних, або точності, що багаторазово перевершує задовольняючу. Швидкість навчання мережі також залежить і від алгоритмів навчання. У даній роботі були реалізовані градієнтні методи навчання, для яких градієнт функції оцінки по параметрах, що набудовуються, обчислюється за принципом подвійності [5].

2.6 Структурна модель НМ

У результаті розробки мережі була обрана структура з двох схованих шарів, у кожному по чотири нейрони. Однак після спрощення стало очевидним, що один нейрон у другому шарі надлишковий. Також після спрощення були вилучені незначущі вхідні поля, виключення яких майже не погіршить точність рішення задачі мережею. Параметри НМ після спрощення в таблиці 2.

Таблиця 2

Параметр	Мережа
Число шарів	2
Число вхідних полів	21
Число штучних нейронів у схованих шарах	4-3
Крутість сігмоїди	0,1

Структура мережі зображена на малюнку 1.

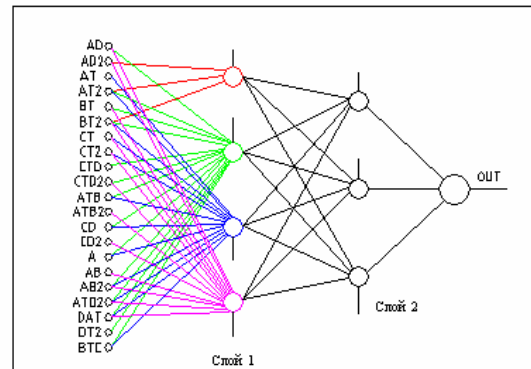


Рис. 1 - Архітектура НМ

3. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ НМ НА ПЛМ

Для реалізації отриманої структурної моделі НМ в апаратурі вона була представлена в кодах мови VHDL. Отримана мовою VHDL модель містить функціональну модель НМ і складений «тестбенч» із необхідними тестами для перевірки функціонування до синтезу, після синтезу і після розміщення логіки на чіпі. Наступним етапом модель була синтезована в цифровій логіці для наступної імплементації на таких цифрових пристроях як FPGA.

Для реалізації даної НМ був обраний чіп Xilinx Virtex II FPGA. Він має убудовані блоки перемножувачів 18x18, спеціальну логіку прискореного переносу для високошвидкісних арифметичних операцій, часткове реконфігурування, гнучкі логічні ресурси: до 122 880 тригерів, до 122 880 16-ти розрядних зсувових регістрів на базі LUT, підтримка багатовходових перемножувачів і логічних функцій. Зроблений за технологією 0.15-мкм із 8-шаровою металізацією і 0.12-мкм швидкодіючими транзисторами. Часткове реконфігурування дозволяє робити дуже швидкі наростаючі зміни властиві алгоритмові навчання.

4. ПРИКЛАД РЕАЛІЗАЦІЇ НМ З ОЦІНКОЮ АПАРАТУРНИХ І ЧАСОВИХ ВИТРАТ

Основна задача наступного етапу проектування - описати модель НМ стилем, розрахованим на подальший синтез у схемну логіку. Навіть якщо код VHDL компілюється і правильно моделюється, не обов'язково, що він буде правильно синтезуватися. І навіть якщо код VHDL правильно синтезується, то результуюча реалізація може бути неефективною.

Кожен синтезатор має власний функціональний пакет для операцій, звичайно використовуваних в апаратурних моделях. IEEE надає стандартний синтезуючий пакет, що включає функції для арифметичних операцій над *bit_vector* і *std_logic* векторами. Пакет *numeric_bit* визначає арифметичні операції *bit_vector* векторів. Пакет містить два класи визначених масивів для подачі беззнакових і знакових бітових значень:

```
type unsigned is array (natural range <>) of bit;
type signed is array (natural range <>) of bit;
```

Числа зі знаками представлені двоїчним підсумовуючим кодом. Пакет містить версії арифметичних операцій, операцій відносини, логічних і операцій зсуву, що перезавантажуються, так само як і функції перетворення з типу в тип. Пакет *numeric_std* визначає аналогічні операції для *std_logic* векторів.

VHDL-код проекту містить функції, яких немає в синтезуючих бібліотеках. З цієї причини ми замінили *std_logic* на тип *unsigned* або *signed*, щоб було можна використовувати визначені для цих типів оператори, що перезавантажуються, “+” і “*”. У проекті ми також замінили *bit* логіку на тип *std_logic*.

У програмі були використані пакети бібліотеки IEEE:

```
IEEE.std_logic_1164.all;
ieee.numeric_std.all;
ieee.std_logic_unsigned.all;
ieee.std_logic_signed.all;
```

Були також створені власні пакети (пакет *global_matrixs* містить опис матриці ваг першого нейрона першого шару), що містять опис типу сигналів ваг синапсів нейрона. Надалі можна користуватися цими пакетами, увівши наступний рядок в розділі опису використовуваних бібліотек:

```
use global_matrixs.all;
```

4.1 Проблеми реалізації

Основні проблеми при реалізації НМ в апаратурі:

- представлення чисел із рухомою комою;
- реалізація нелінійної активаційної функції нейрона;

Робота з числами з рухомою комою є значною проблемою, тому що арифметичні операції для чисел з рухомою комою набагато складніші операцій для цілих чисел. До того ж, спеціалізовані схеми для операцій над числами з рухомою комою набагато складніші, працюють повільніше і займають велику площу кристала.

Доцільним рішенням для удосконалення продуктивності проекту і зменшення апаратурних витрат є конвертування чисел з рухомою комою в числа типу *integer*. Звичайно, це накладає деякі обмеження на точність, але в даному окремому випадку гарні результати були досягнуті.

Діапазон зміни ваг (-1;1) - число з рухомою комою; описується мовою VHDL як число у форматі *real*. Числа формату *real* не синтезовані, тому необхідно перейти до представлення ваг у цілочисельному форматі. Для цього було зроблене наступне:

Сигнал, що представляє ваговий коефіцієнт синапса нейрона, має тип *unsigned* (ціле без знака) і розрядність вісім біт. Створюємо двоїчне представлення дробової частини вагового коефіцієнта таким чином, що якщо, наприклад, ваговий коефіцієнт 2-го синапса першого нейрона $W_{1,2} = 0.5$ у двоїчній системі представлений як 0.1, то будемо його записувати як 1000 0000.

У програмі, що реалізує НМ, на вхідні сигнали X_i подаються ознаки вже в нормалізованому виді. Тобто сигнали X_i теж знаходяться в діапазоні (-1;1) і повинні бути представлені типом *unsigned*.

Результат множення $X*W$ буде менше одиниці. У моделі нейрона це синапс, значення сигналу якого повинно підсумовуватися зі значеннями сигналів інших синапсів нейрона. Для того, щоб результат множення був в зручному для нас виді і згодом міг бути використаний у підсумовуванні, зсуваємо добуток Qp_i на 8 розрядів праворуч, а потім беремо молодші 8 бітів. Таким чином, ми округляємо отриманий нами добуток до 8-ми біт:

```
Qout_frac:= Qp3 Srl n;
Q_frac := Qout_frac(n-1 downto 0);
```

Інша проблема представлення арифметичних операцій у цифровій апаратурі полягає в реалізації активаційної функції нейрона. Деякі активаційні функції, наприклад, такі як сигмоїдна (використовувана при створенні моделі нейрона за допомогою пакета NeuroPro 0.5), мають потребу в модифікаціях для того, щоб спростити її для проектування в апаратурі. У нашому випадку, сигмоїдна функція була заміщена кусочно-лінійною функцією (рис. 2).

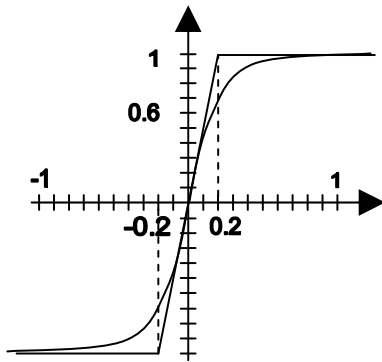


Рис. 2 - Активаційна функція нейронів

Таким чином, активаційна функція може бути описана мовою VHDL в такий спосіб:

```

if ADD_frac2 <= COMP then
    Q_B:="11111111";
else Q1:= ADDITION*ADD_f;
    Q1:=Q1 sll 3;
    Q_B:= Q1(10 downto 3);
end if;

```

Тобто, якщо значення на виході суматора лежить у діапазоні від $-0,2$ до $0,2$, то Y буде обчислюватися по функції $y=5x$, в іншому випадку функція дорівнює -1 або 1 (1 - якщо вихід суматора позитивний).

4.2 Архітектура НМ при описі її мовою VHDL

НМ складається з трьох шарів. Вхідний шар має чотири нейрони, на які паралельно надходить інформація зі тестової вибірки. Кожен нейрон має різну кількість синапсів – різні вхідні сигнали (ознаки) надходять на кожному з нейронів вхідного шару. Перший нейрон має три синапси, другий – 14, третій – 12, четвертий – 13 (див. мал. 1).

Другий схований шар складається з 3-х нейронів, що обробляють дані від нейронів першого шару. Кожен з нейронів другого шару має 4 вхідні шини, що представляють синапси нейрона і сигнал bias – сигнал зсуву.

Третій шар – шар підсумовування складається з двох елементів, що підсумовують виходи від 3-х нейронів попереднього шару.

Матриця ваг описується як одномірний масив, елементами якого є вагові коефіцієнти кожного із синапсів нейрона. Матриця ваг окрема для кожного з нейронів НМ.

При описі моделі НМ мовою VHDL були створені наступні компоненти: чотири нейрони першого шару, що містять опис у чотирьох

окремих файлах - *sample3.vhdl*, *neuron11_2.vhdl*, *neuron11_3.vhdl*, *neuron11_4.vhdl*; один компонент визначає нейрони другого шару (вони однакові по архітектурі) у файлі *neuron21_comp.vhdl* і підсумовуючий елемент підсумовуючого шару у файлі *sum_component.vhdl*. Другий шар складається з трьох компонентів нейрона другого шару, що підсумовує шар складається з двох підсумовуючих елементів і схеми порівняння, тому що на вихідний тригер надходить результат із того підсумовуючого елемента, чий вихід більше.

Кожний з компонентів, що представляють нейрон, містить у собі визначену кількість синапсів, один суматор і граничну функцію. Шина, що відповідає за подачу вхідних сигналів, а також інші вхідні шини мають розмірність 8 біт (7 downto 0), тому що велика точність не потрібна.

Синапс у моделі нейрона реалізовано у виді операції множення $X_{ij} * W_{ij}$,

де X_{ij} - вхідна ознака, що надходить на j -тий синапс i -го нейрона,

W_{ij} - ваговий коефіцієнт j -ого синапса i -го нейрона.

Підсумовування оперує з числами у форматі *signed*. Для цього, перед тим як підсумовувати, необхідно перетворити результат множення з формату *unsigned* у *signed* з урахуванням знака добутку.

В ході роботи над проектом було з'ясовано, що синтезована схема цілком залежить від способу опису функціонування пристрою. Опис НМ і контролера було зроблено на структурному рівні на мові VHDL, і в результаті, після синтезу одержані такі елементи логіки як регістри, суматори, тригери й інші логічні блоки, що були зіставлені до відповідних об'єктів структури.

Для функціонального моделювання використано пакет фірми Aldec Active-HDL 5.0. Результати симуляції НМ до синтезу і після збігаються, що означає правильність синтезу схеми НМ.

4.3 Опис синтезованої моделі

Синтез остаточних VHDL файлів був здійснений з використанням пакета SynplifyPro 7.0 від Synplicity® і орієнтований на Virtex-II сімейство ПЛІС з архітектурою FPGA. Була використана XC2V6000 мікросхема сімейства

Virtex-II. Вона містить 6 млн. системних вентилів 144 блоку перемножувачів і блокову пам'ять обсягом 2592 Кбіт.

Розроблений пристрій складається з контролера і самої НМ. Блок управління працює на системній частоті 20 MHz і надає необхідні сигнали управління. Контролер керує роботою НМ. Він синхронізує роботу кожного шару і повідомляє, коли можна подавати на входи наступний тестовий вектор.

Схема НМ, синтезована за допомогою пакета SymplifyPro 7.0., містить чотири компоненти нейрона, один компонент схованого шару й один компонент шару підсумовування. Кожний з нейронів має визначену кількість логічних вентилів, що реалізують функцію нейронного елемента.

Після синтезу була здійснена перевірка правильності функціонування пристрою. Таке моделювання демонструє успішність етапу синтезу.

Симуляція після етапу розміщення і розведення необхідна для впевненості, що проект буде працювати з необхідною швидкістю, приймаючи в увагу час завантаження логічних вентилів і затримку поширення сигналу в лініях.

В таблиці 3 наведені ресурси, використані в FPGA:

Таблиця 3

Число CLBs	3471 (3%)
Число bonded IOBs	201 (2 %)
Число глобальних буферів	1/8 (12%)
Сума використаних вентилів	3948
Мінімальний період	25.405 ns
Максимальна частота	39.362 MHz
Максимальна затримка на лінії	1.148 ns
Середня затримка зв'язку	3.494 ns
Середня разфазіровка синхронізуючих імпульсів	0.248 ns
Максимальна затримка контакту	1.148 ns
Середня затримка зв'язку на гірших зв'язках	9.184 ns

5. ВИСНОВКИ

Приведена вище методологія проектування штучних НМ орієнтована на апаратне вирішення задачі класифікації. Особливість даної статті полягає у визначенні й аналізі основних структурних і функціональних особливостей НМ, що можуть впливати на можливість їх

автоматизованого проектування відповідно до сучасної методики синтезу пристроїв на FPGA. Описано припустимі способи обліку особливостей НМ у процесі синтезу.

НМ реалізована на базі мікросхеми XC2V6000 сімейства Virtex-II. Для опису функціонування НМ використана мова VHDL. Оцінено апаратні й часові характеристики апаратної реалізації НМ.

Використання описаної методики проектування дозволяє досягти скорочення термінів реалізації НМ при раціональному розподілу площі кристала FPGA для НМ, а також оптимізувати схемну реалізацію на етапі синтезу НМ.

СПИСОК ЛІТЕРАТУРИ

1. Кнышев Д.А., Кузелин М.О. ПЛИС фирмы "Xilinx": описание структуры основных семейств. – М.: Издательский дом «Додэка-XXI», 2001.
2. Haykin S. Neural Networks, A Comprehensive Foundation. New York: Macmillan College Publishing Company. – 1994. 678 p.
3. Гусева И.С. Морфогенез и генетика гребешковой кожи.- Минск: Беларусь, 1986. 158 с.
4. Багацкая Н.В. Анализ пальцевой и ладонной дерматоглифики при задержке полового созревания у мальчиков подростков.// Проблемы екологічної та медичної генетики і клінічної імунології. -2000. – в.2(28).С.37-42
5. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск.: "Наука", 1996.276 с.
6. Россиев Д.А., Головенкин С.Е., Шульман В.А., Матюшин Г.В. Прогнозирование осложненного инфаркта миокарда нейронными сетями. // Нейроинформатика и ее приложения
7. Горбань А.Н. Обучение нейронных сетей. М.: изд. СССР - США СП "ParaGraph", 1990. 160 с.

Лісовик Уляна Ігорівна, аспірант кафедри автоматизації та проектування обчислювальної техніки ХНУРЕ. Наукові інтереси: технічна та медична діагностика з використанням НС. Хоббі: туризм, музика, іноземна мова.

Ліпчанський Олександр Іванович, докторант кафедри автоматизації та проектування обчислювальної техніки ХНУРЕ. Наукові інтереси: технічна та медична діагностика