Применение микропроцессора NM6403 для эмуляции нейронных сетей

П.А. Шевченко, Д.В.Фомин, В.М.Черников, П.Е.Виксне
НТЦ “Модуль”, Россия, 125190, Москва, 4-я ул. Восьмого марта, 3,
тел. +7-095-152-9335, факс +7-095-152-4661,
E-mail: fomin@module.vympel.msk.ru

1. ВВЕДЕНИЕ

В настоящее время рядом ведущих микроэлектронных фирм выпускаются процессоры, ориентированные на выполнение операций по эмуляции различных нейронных сетей. Существуют оптоэлектронные, аналоговые, цифровые и гибридные нейропроцессоры [1]. Наиболее точными и универсальными благодаря своей программируемости являются цифровые нейропроцессоры [2]. Одним из важнейших критериев универсальности цифрового нейропроцессора является его способность обрабатывать входные данные различных разрядностей. Существует ряд нейропроцессоров, выполняющих операции над данными, разрядность которых может быть программно задана в диапазоне от 1 до 16 [3, 4]. Как правило, такие процессоры обрабатывают одно, восьми и шестнадцатиразрядные данные. Исключение составляет процессор Lneuro фирмы Philips [5], работающий с входными данными любой разрядности от 1 до 8. Однако в данном процессоре используется последовательный способ обработки данных, что является одной из основных причин его относительно низкой производительности.

Предлагаемый новый 64-разрядный нейропроцессор NM6403 является высокопроизводительным вычислителем, в котором аппаратно поддерживаются такие операции, как умножение матрицы на матрицу или матрицы на вектор, сложение векторов, вычисление функций насыщения для элементов векторов и другие операции векторной арифметики [6]. Данный процессор производит за один такт обработку векторов, каждый из которых представляет собой 64-разрядное слово, в котором упакованы целочисленные данные. Причем разрядность каждого элемента вектора задается программно и может принимать любое значение в диапазоне от 1 до 64. С уменьшением разрядности данных увеличивается их количество в каждом векторе и тем самым повышается производительность процессора. Процессор имеет аппаратные средства для построения на его основе многопроцессорных систем. Он может использоваться в различных системах цифровой обработки сигналов. Однако основное назначение процессора NM6403 - это эмуляция нейронных сетей.

2. МОДЕЛЬ НЕЙРОННОЙ СЕТИ

На рис.1 представлена модель слоя нейронной сети, эмулируемой процессором NM6403. В общем случае один слой нейронной сети имеет N нейронных входов и состоит из M нейронов. При этом m-м нейрон выполняет взвешенное суммирование N данных X_1, X_2,…, X_N, подаваемых на соответствующие нейронные входы, с учетом смещения V_m данного нейрона:

$$Y_m = V_m + \sum_{n=1}^{N} X_n \times W_{n,m} ,$$

где $W_{n,m}$ - весовой коэффициент n-го входа в m-м нейроне (n=1,2,…,N; m=1,2,…,M). Затем m-й нейрон вычисляет функцию насыщения F_{Q_m} от результата взвешенного суммирования Y_m:

$$OUT_m = F_{Q_m}(Y_m) ,$$

где Q_m - параметр функции насыщения, вычисляемой для m-го нейрона, равный числу значащих битов в результате R_m. Общий вид функций насыщения, реализуемых нейропроцессором, представлен на рис.2. Все входные данные, весовые коэффициенты, пороговые значения и результаты представляются в дополнительном коде.

Процессор NM6403 является универсальным средством для эмуляции различных нейронных сетей. При его использовании пользователь может программно задавать следующие параметры нейронной сети: число слоев, число нейронов и нейронных входов в каждом слое, разрядность данных на каждом нейронном входе, разрядность

каждого весового коэффициента, разрядность выходного значения каждого нейрона, параметр функции насыщения для каждого нейрона.

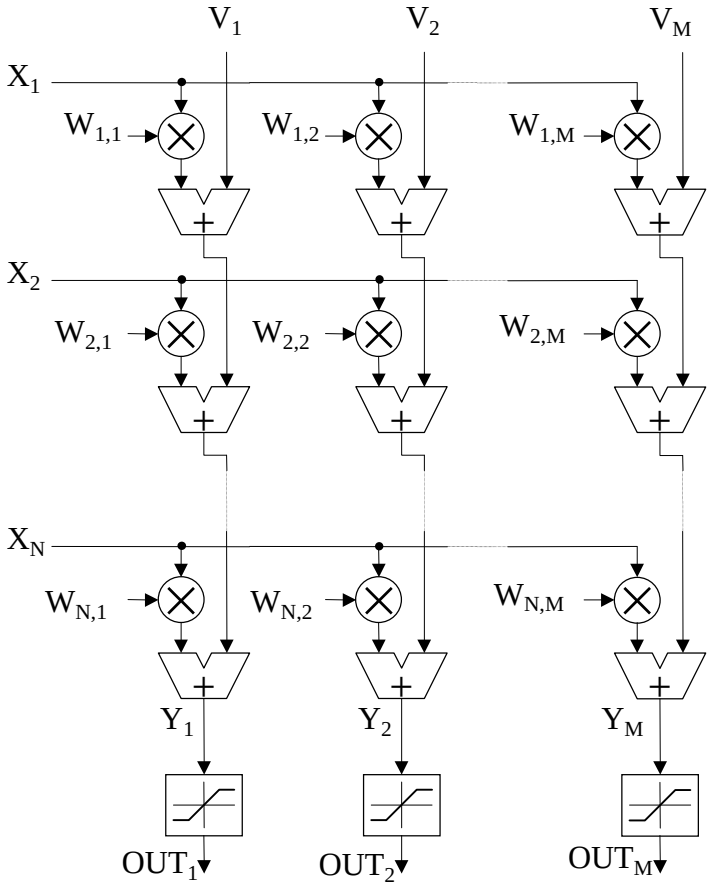

Рис. 1. Модель слоя нейронной сети, эмулируемой нейропроцессором

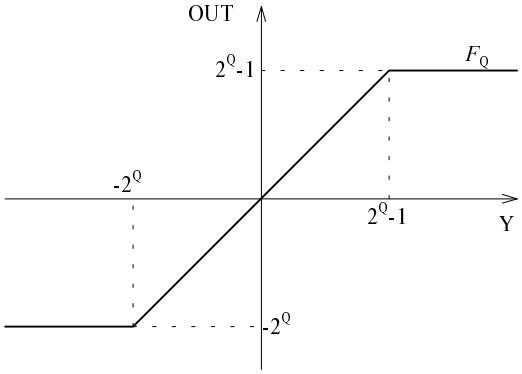

Рис. 2. Общий вид функций насыщения, вычисляемых нейропроцессором.

3. ОБЩИЙ ПОДХОД К ЭМУЛЯЦИИ НЕЙРОННЫХ СЕТЕЙ

Один процессор NM6403 позволяет эмулировать нейронную сеть практически неограниченных размеров. Эмуляция нейронной сети осуществляется послойно (последовательно слой за слоем).

Каждый слой нейронной сети разбивается на последовательно обрабатываемые фрагменты. Данное разбиение осуществляется следующим образом. Множество нейронных входов слоя разбивается на группы входов так, чтобы суммарная разрядность данных, подаваемых на каждую группу входов, была равна разрядности нейропроцессора - 64. Множество нейронов слоя разбивается на группы нейронов так, чтобы суммарная разрядность результатов взвешенного суммирования для каждой группы нейронов также была равна 64. При таком разбиении входов и выходов весь слой нейронной сети разбивается на фрагменты двух типов, имеющих различное функциональное назначение. Фрагменты первого типа образуют матричную структуру, выполняющую все операции взвешенного суммирования. Причем каждому фрагменту первого типа соответствует одна из групп нейронных входов и одна из групп нейронных выходов. Каждый такой фрагмент выполняет взвешенное суммирование всех данных, подаваемых на соответствующую группу входов, для всех нейронов соответствующей группы нейронов. Каждый фрагмент второго типа формирует выходные значения для всех нейронов, входящих в состав одной из групп нейронов, путем вычисления функции насыщения от результатов взвешенного суммирования.

Рис.1 можно использовать в качестве иллюстрации описанного выше принципа разбиения слоя нейронной сети на фрагменты. Для этого необходимо представить, что каждый блок, приведенный на рис.1, выполняет операции над 64-разрядными векторами упакованных данных, а все входы и межблочные соединения служат для пересылки этих векторов. При этом на рис.1 каждому фрагменту первого типа соответствует пара устройств, выполняющих умножение и сложение, а каждому фрагменту второго типа соответствует одно устройство вычисления функций насыщения.

Процесс эмуляции слоя нейронной сети на нейропроцессоре NM6403 состоит из последовательно выполняемых макроопераций, каждая из которых эмулирует один фрагмент слоя. При этом число выполняемых макроопераций равно числу фрагментов в слое. Обработка данных, подаваемых на нейронные входы, ведется нейропроцессором в пакетном режиме - по T наборов входных данных в каждом пакете. При выполнении каждой макрооперации по эмуляции фрагмента слоя нейронной сети последовательно выполняются вычисления для каждого набора входных данных. Поэтому за один проход всех фрагментов слоя нейропроцессор вычисляет T наборов выходных значений нейронов, каждый из которых соответствует одному из наборов входных данных. При обработке одного пакета входных данных значения весовых коэффициентов и смещений нейронов не изменяются. Величина T задается программно и может принимать любое целочисленное значение в диапазоне от 1 до 32.

4. ФОРМАТЫ ДАННЫХ, ОБРАБАТЫВАЕМЫХ В ПРОЦЕССЕ ЭМУЛЯЦИИ НЕЙРОННЫХ СЕТЕЙ

Пакет данных, подаваемых на n-ю группу нейронных входов слоя нейронной сети (n = 1,2,…,N), представляет собой вектор $\mathbf{X}_n = \begin{pmatrix} \mathbf{X}_n^1 \\ \mathbf{X}_n^2 \\ \vdots \\ \mathbf{X}_n^T \end{pmatrix}$, t-м элементом которого (t=1,2,…,T) является вектор $\mathbf{X}_n^t = (X_{n,1}^t \quad X_{n,2}^t \quad \cdots \quad X_{n,N_n}^t)$, ν-м элементом $X_{n,\nu}^t$ которого (ν = 1,2,…, N_n) является данное из t-го набора входных данных, подаваемое на ν-й вход n-й группы нейронных входов слоя.

Вектор $\mathbf{X}_n^t$ представляет собой 64-разрядное слово, в котором упаковано N_n целочисленных данных. При этом младшие разряды вектора $\mathbf{X}_n^t$ являются разрядами первого данного $X_{n,1}^t$, далее следуют разряды второго данного $X_{n,2}^t$ и т.д. Старшие разряды вектора $\mathbf{X}_n^t$ являются разрядами N_n-го данного X_{n,N_n}^t. Разрядность каждого данного, упакованного в векторе $\mathbf{X}_n^t$, может принимать любое четное значение в диапазоне от 2 до 64. Как следствие, количество данных N_n в векторе $\mathbf{X}_n^t$ может принимать любое целочисленное значение от 1 до 32.

Единственное ограничение заключается в том, что суммарная разрядность всех данных, упакованных в одном векторе $\mathbf{X}_n^t$, должна быть равна разрядности нейропроцессора - 64.

При эмуляции фрагмента первого типа, осуществляющего взвешенное суммирование n-й группы входных данных для m-й группы нейронов (n=1,2,…,N; m=1,2,…,M), нейропроцессор выполняет следующую операцию для каждого t-го набора входных данных (t=1,2,…,T):

$$\mathbf{Y}_{n,m}^t = \mathbf{Y}_{n-1,m}^t + \mathbf{X}_n^t \times \mathbf{W}_{n,m}, \qquad (1)$$

где $\mathbf{W}_{n,m} = \begin{pmatrix} W_{n,m,1,1} & W_{n,m,1,2} & \cdots & W_{n,m,1,M_m} \\ W_{n,m,2,1} & W_{n,m,2,2} & \cdots & W_{n,m,2,M_m} \\ \vdots & \vdots & & \vdots \\ W_{n,m,N_n,1} & W_{n,m,N_n,2} & \cdots & W_{n,m,N_n,M_m} \end{pmatrix}$ - матрица, элементом $W_{n,m,\nu,\mu}$ которой является весовой коэффициент ν-го входа n-й группы нейронных входов для μ-го нейрона в m-й группе нейронов ($\nu=1,2,\ldots,N_n$; $\mu=1,2,\ldots,M_m$);

$\mathbf{Y}_{n,m}^t = (Y_{n,m,1}^t \quad Y_{n,m,2}^t \quad \cdots \quad Y_{n,m,M_m}^t)$ - вектор, μ-м элементом $Y_{n,m,\mu}^t$ которого является результат взвешенного суммирования данных, подаваемых на входы n первых групп входов, для μ-го нейрона в m-й группе нейронов ($\mu=1,2,\ldots,M_m$):

$$Y_{n,m,\mu}^t = Y_{n-1,m,\mu}^t + \sum_{\nu=1}^{N_n} X_{n,\nu} \times W_{n,m,\nu,\mu}.$$

Причем в качестве вектора $\mathbf{Y}_{0,m}^t$, обрабатываемого первым фрагментом m-й группы нейронов, используется вектор $\mathbf{V}_m = (V_{m,1} \quad V_{m,2} \quad \cdots \quad V_{m,M_m})$, μ-м элементом $V_{m,\mu}$ которого является смещение μ-го нейрона в m-й группе нейронов ($\mu=1,2,\ldots,M_m$).

64-разрядные векторы $\mathbf{Y}_{1,m}^t$, $\mathbf{Y}_{2,m}^t$, … , $\mathbf{Y}_{N,m}^t$ и $\mathbf{V}_m$ имеют одинаковый формат - они содержат одинаковое число M_m упакованных в них данных, разрядность каждого из которых фиксирована и может принимать любое целочисленное значение от N_{min} до 64. Значение минимальной разрядности каждого элемента в этих векторах зависит от суммарного числа нейронных входов в слое и вытекает из необходимости избегать арифметического переполнения при сложении чисел с одним значащим разрядом:

$$N_{min} = 1 + \left\lceil \log_2 \left(\sum_{n=1}^{N} N_n \right) \right\rceil.$$

Параметр M_m может принимать значения в диапазоне от 1 до $\lfloor 64 / N_{min} \rfloor$.

ν-я строка ($\nu=1,2,\ldots,N_n$) матрицы $\mathbf{W}_{n,m}$ представляет собой вектор весовых коэффициентов $\mathbf{W}_{n,m,\nu} = (W_{n,m,\nu,1} \quad W_{n,m,\nu,2} \quad \cdots \quad W_{n,m,\nu,M_m})$, имеющий такой же формат, что и векторы $\mathbf{Y}_{1,m}^t$, $\mathbf{Y}_{2,m}^t$, … , $\mathbf{Y}_{N,m}^t$ и $\mathbf{V}_m$.

Выполнение операции (1) над всеми векторами, входящими в состав пакета $\mathbf{X}_n$, задается одной командой “**vsum**” нейропроцессора. Данная команда выполняется нейропроцессором за T тактов, причем в t-м такте операция (1) выполняется над векторами $\mathbf{X}_n^t$ и $\mathbf{Y}_{n-1,m}^t$ (t=1,2,…,T).

Настройка аппаратных средств нейропроцессора на обработку векторов данных требуемой разрядности осуществляется путем программной загрузки следующих управляющих слов в соответствующие конфигурационные регистры нейропроцессора:

- XB_n - слово, задающие границы элементов в векторах $\mathbf{X}_n^t$;
- YB_m - слово, задающее границы элементов в векторах $\mathbf{Y}_{1,m}^t$, $\mathbf{Y}_{2,m}^t$, … , $\mathbf{Y}_{N,m}^t$, $\mathbf{V}_m$, $\mathbf{W}_{n,m,1}$, $\mathbf{W}_{n,m,2}$, … , $\mathbf{W}_{n,m,N_n}$ (t=1,2,…,T).

При эмуляции фрагмента второго типа для m-й группы нейронов (m=1,2,…,M) процессор формирует пакет векторов $\mathbf{OUT}_m = \begin{pmatrix} \mathbf{OUT}_m^1 \\ \mathbf{OUT}_m^2 \\ \vdots \\ \mathbf{OUT}_m^T \end{pmatrix}$, t-м элементом которого (t=1,2,…,T) является вектор $\mathbf{OUT}_m^t = (\mathrm{OUT}_{m,1}^t \quad \mathrm{OUT}_{m,2}^t \quad \cdots \quad \mathrm{OUT}_{m,M_m}^t)$, μ-м элементом $\mathrm{OUT}_{m,\mu}^t$ которого является результат вычисления функции насыщения $F_{Q_{m,\mu}}$ над μ-м элементом $\mathrm{Y}_{N,m,\mu}^t$ вектора $\mathbf{Y}_{N,m}^t$ (μ=1,2,…,M_m):

$$\mathrm{OUT}_{m,\mu}^t = F_{Q_{m,\mu}}(\mathrm{Y}_{N,m,\mu}^t)\,.$$

Вычисление функций насыщения для всех элементов всех векторов, входящих в состав пакета $\mathbf{OUT}_m$, задается одной командой “**activate”** нейропроцессора. Данная команда выполняется нейропроцессором за T тактов. При этом в t-м такте формируется вектор $\mathbf{OUT}_m^t$ (t=1,2,…,T). имеющий точно такой же формат, что и описанные выше векторы $\mathbf{Y}_{1,m}^t$, $\mathbf{Y}_{2,m}^t$, … , $\mathbf{Y}_{N,m}^t$ и $\mathbf{V}_m$.

Настройка аппаратных средств нейропроцессора на выполнение функций насыщения с требуемыми параметрами над векторами требуемого формата осуществляется путем программной загрузки управляющего слова Q_m в соответствующий конфигурационный регистр нейропроцессора.

5. ПРИМЕР ПРОГРАММЫ ЭМУЛЯЦИИ СЛОЯ НЕЙРОННОЙ СЕТИ

Весь процесс эмуляции слоя нейронной сети на одном процессоре NM6403 можно представить в виде M последовательно выполняемых процедур, каждая из которых осуществляет эмуляцию одной группы нейронов и состоит из N+1 последовательно выполняемых макроопераций, каждая из которых эмулирует один фрагмент слоя нейронной сети. Причем n-я макрооперация данной процедуры осуществляет эмуляцию фрагмента первого типа, выполняющего взвешенное суммирование данных, подаваемых на n-ю группу нейронных входов, с накоплением результата (n=1,2,…,N). Последняя макрооперация процедуры эмулирует фрагмент второго типа, выполняющий вычисление функций насыщения от результатов взвешенного суммирования для соответствующей группы нейронов.

На рис. 3 приведен пример процедуры эмуляции m-й группы нейронов для T наборов входных данных (m=2,3,…,M). Процедура эмуляции первой группы имеет ряд особенностей на этапе эмуляции первого фрагмента (рис.4), что связано с необходимостью загрузки ряда исходных данных, общих для всех процедур, и отсутствием результатов эмуляции предыдущей группы нейронов. Остальные фрагменты первой группы нейронов эмулируются так же, как и в остальных группах нейронов. Примеры написаны на языке ассемблера нейропроцессора NM6403. С целью повышения наглядности процедуры описаны без использования циклов, переходов и других операторов, управляющих потоком команд. Каждая строка процедуры содержит одну ассемблерную команду и комментарий к ней, начинающийся символами “//”.

В рассматриваемых процедурах используются следующие регистры и блоки внутренней памяти нейропроцессора:

- **WOPER** - блок памяти для хранения матрицы весовых коэффициентов, используемой в операциях взвешенного суммирования;
- **WBUF** - блок памяти для накопления векторов, составляющих матрицу весовых коэффициентов;
- **WFIFO** - двухпортовое FIFO, предназначенное для синхронизации процессов чтения векторов весовых коэффициентов из внешней памяти и накопления матрицы весовых коэффициентов в WBUF;
- **AFIFO** - двухпортовое FIFO, выполняющее функции аккумулятора результатов при выполнении операций над пакетами векторов упакованных данных;
- **RAM** - однопортовая память магазинного типа, предназначенная для оперативного хранения пакета векторов упакованных данных;
- **AR0**, **AR1**, **AR4** - адресные регистры;
- **VR** - регистр для хранения вектора смещений;
- **F2CR** - регистр для хранения параметров функций насыщения, вычисляемых для векторов упакованных данных.

- **SB1** и **SB2** - регистры для хранения управляющего слова, задающего формат векторов упакованных данных, над которыми выполняется операция взвешенного суммирования;
- **NB1** и **NB2** - регистры для хранения управляющего слова, задающего формат векторов результатов при выполнении операций над векторами упакованных данных.

```
//******* Эмуляция 1-го фрагмента m-й группы нейронов *******
nb1 = [YB_m];      // Запись управляющего слова YB_m в регистр NB1
sb1 = [XB_1];      // Запись управляющего слова XB_1 в регистр SB1
rep N_1 wfifo = [ar4++], ftw;
                   // Загрузка матрицы W_{1,m} из внешней памяти в WFIFO
                   // и ее последующая пересылка из WFIFO в WBUF
wtw;               // Пересылка W_{1,m} из WBUF в WOPER
ar0 = X_2;         // Загрузка начального адреса пакета векторов X_2 в регистр AR0
vr = [V_m];        // Загрузка вектора V_m в регистр VR
rep T [ar1++] = afifo with vsum, ram, vr;
                   // Пересылка OUT_{m-1} из AFIFO во внешнюю память,
                   // вычисление вектора Y^t_{1,m} = X^t_1 × W_{1,m} + V_m
                   // (вектор X^t_1 считывается из внешней памяти)
                   // и его запись в AFIFO (t=1,2,…,T)
//******* Эмуляция 2-го фрагмента m-й группы нейронов *******
sb1 = [XB_2];      // Запись управляющего слова XB_2 в регистр SB1
rep N_2 wfifo = [ar4++], ftw;
                   // Загрузка матрицы W_{2,m} из внешней памяти в WFIFO
                   // и ее последующая пересылка из WFIFO в WBUF
wtw;               // Пересылка W_{2,m} из WBUF в WOPER
rep T data = [ar0++] with vsum, data, afifo;
                   // Вычисление вектора Y^t_{2,m} = X^t_2 × W_{2,m} + Y^t_{1,m}
                   //(вектор X^t_2 считывается из внешней памяти)
                   // и его запись в AFIFO (t=1,2,…,32)
⋮
//******* Эмуляция N-го фрагмента m-й группы нейронов *******
sb1 = [XB_N];      // Запись управляющего слова XB_N в регистр SB1
rep N_N wfifo = [ar4++], ftw;
                   // Загрузка матрицы W_{N,m} из внешней памяти в WFIFO
                   // и ее последующая пересылка из WFIFO в WBUF
wtw;               // Пересылка матрицы W_{N,m} из WBUF в WOPER
rep T data = [ar0++] with vsum, data, afifo;
                   // Вычисление вектора Y^t_{N,m} = X^t_N × W_{N,m} + Y^t_{N−1,m}
                   //(вектор X^t_N считывается из внешней памяти)
                   // и его запись в AFIFO (t=1,2,…,32)
//******* Эмуляция (N+1)-го фрагмента m-й группы нейронов *******
f2cr = [Q_m];      // Запись управляющего слова Q_m в регистр F2CR
rep T vsum, 0, activate afifo;
                   // Вычисление вектора OUT^t_m = F_{Q_m}(Y^t_{N,m}),
                   // записываемый в AFIFO (t=1,2,…,32)
```

Рис.3. Процедура эмуляции m-й группы нейронов (m=2,3,…,M)

//******* Эмуляция 1-го фрагмента 1-й группы нейронов *******

ar4 = $\mathbf{W}_{1,1}$; // Загрузка начального адреса матрицы $\mathbf{W}_{1,1}$ в регистр AR4
nb1 = [YB_1]; // Запись управляющего слова YB_1 в регистр NB1
sb1 = [XB_1]; // Запись управляющего слова XB_1 в регистр SB1
rep N_1 wfifo = [ar4++], ftw;
// Загрузка матрицы $\mathbf{W}_{1,1}$ из внешней памяти в WFIFO
// и ее последующая пересылка из WFIFO в WBUF
wtw; // Пересылка $\mathbf{W}_{1,1}$ из WBUF в WOPER
ar1 = $\mathbf{OUT}_1$; // Загрузка начального адреса пакета векторов $\mathbf{OUT}_1$ в регистр AR1
ar0 = $\mathbf{X}_1$; // Загрузка начального адреса пакета векторов $\mathbf{X}_1$ в регистр AR0
vr = [$\mathbf{V}_1$]; // Загрузка вектора $\mathbf{V}_1$ в регистр VR
rep T data, ram = [ar0++] with vsum ,data, vr;
// пересылка вектора $\mathbf{X}_1^t$ из внешней памяти в RAM,
// вычисление вектора $\mathbf{Y}_{1,1}^t = \mathbf{X}_1^t \times \mathbf{W}_{1,1} + \mathbf{V}_1$
// и его запись в AFIFO (t=1,2,…,32)

Рис.4. Процедура эмуляции первого фрагмента первой группы нейронов

Причем содержимое регистров SB1 и NB1 используется для управления процессом накопления матрицы весовых коэффициентов в WBUF, а содержимое регистров SB2 и NB2 - для управления процессом выполнения операций над векторами упакованных данных. Наличие в нейропроцессоре WBUF, SB1 и NB1 обеспечивает одновременное выполнение двух процессов: выполнения взвешенного суммирования текущего пакета векторов и накопления матрицы весовых коэффициентов для обработки следующего пакета векторов.

Система команд нейропроцессора содержит команды двух типов: скалярные или векторные.

Скалярные команды являются обычными для RISC-процессоров командами и задают однократное действие. В представленном примере используются скалярные команды двух видов:

- **<регистр> = <имя переменной>;** - запись адреса переменной в указанный регистр;
- **<регистр> = [<имя переменной>];** - пересылка переменной из внешней памяти в указанный регистр.

Векторные команды предназначены для выполнения операций над пакетами векторов упакованных данных. Ассемблерное представление векторной команды заканчивается символом “**;**” и может содержать несколько операторов, перечисленных через запятую или начинающихся со служебного слова “**with**”. Каждый оператор векторной команды задает выполнение одной операции. Все операции, задаваемые одной командой, выполняются одновременно.

В векторной команде префикс “**rep <K>**” означает, что каждый оператор данной команды будет последовательно выполняться K раз. То есть по своему действию векторная команда эквивалентна параметрическому циклу, тело которого состоит из одной команды, а число повторений равно K.

В примерах, представленных на рис.3 и 4, используются следующие операторы векторных команд:

- **wfifo = [ar<i>++]** - пересылка вектора весовых коэффициентов из внешней памяти, адресуемой через регистр ARi с использованием автоинкрементного метода адресации, в WFIFO;
- **ftw** - пересылка матрицы весовых коэффициентов из WFIFO в WBUF (операция выполняется за 32 такта независимо от значения параметра K в префиксе “rep <K>”);
- **wtw** - одновременная пересылка матрицы весовых коэффициентов из WBUF в WOPER, содержимого регистра SB1 в регистр SB2 и регистра NB1 в регистр NB2 (операция выполняется за один такт);
- **data = [ar<i>++]** - чтение вектора упакованных данных из внешней памяти, адресуемой через регистр ARi с использованием автоинкрементного метода адресации;
- **data, ram = [ar<i>++]** - чтение вектора упакованных данных из внешней памяти, адресуемой через регистр ARi с использованием автоинкрементного метода адресации, и его запись в RAM;
- **[ar<i>++] = afifo** - пересылка вектора упакованных данных из AFIFO во внешнюю память, адресуемую через регистр ARi с использованием автоинкрементного метода адресации;

- **vsum, ram, vr** - сложение вектора, хранящегося в регистре VR, с произведением вектора, считываемого из RAM, на матрицу весов, хранящуюся в WOPER, и запись сформированного вектора результатов в AFIFO;
- **vsum, data, afifo** - сложение вектора, считываемого из AFIFO, с произведением вектора, считываемого из внешней памяти с помощью одного из описанных выше операторов, на матрицу весов, хранящуюся в WOPER, и запись вектора результатов в AFIFO;
- **vsum, 0, activate afifo** - вычисления функций насыщения для элементов вектора, считанного из AFIFO, и запись вектора результатов в AFIFO.

Конвейер выполнения команд в нейропроцессоре NM6403 позволяет начать выполнение очередной скалярной или векторной команды не дожидаясь окончания выполнения предыдущей векторной команды. Это возможно в тех случаях, когда ресурсы нейропроцессора, необходимые для выполнения очередной команды, не заняты предыдущими векторными командами. В общем случае нейропроцессор позволяет одновременно выполнять до пяти векторных команд и одну скалярную. В примере, приведенном на рис.3, все команды, охваченные одной скобкой “{”, расположенной слева от комманд, выполняются на фоне предшествующей им векторной команды.

При работе с быстродействующей внешней памятью, позволяющей выполнять циклы чтения и записи за один такт, каждая скалярная команда выполняется за один такт, каждая векторная команда, содержащая оператор “ftw”, - за 32 такта (независимо от наличия или отсутствия в ней префикса “rep <K>”), а каждая векторная команда, не содержащая оператор “ftw”, - за K тактов. В общем случае эмуляция слоя нейронной сети для T наборов входных данных может быть выполнена приблизительно за $34 \times (N+1) \times M \times \lceil T/32 \rceil$ тактов. При этом в векторных командах, выполняющих взвешенное суммирование, следует использовать параметр K, равный 32. При меньших значениях K вычислительные ресурсы нейропроцессора будут простаивать во время загрузки очередной матрицы весовых коэффициентов в WBUF, что приведет к увеличению длительности процесса эмуляции.

К сожалению, в данной статье не представляется возможным проиллюстрировать способность нейропроцессора выполнять коммутацию элементов в векторах упакованных данных. При эмуляции нейронных сетей данная способность нейропроцессора позволяет выполнить более плотную упаковку результатов в векторах, формируемых на выходах каждого слоя нейронной сети. Данная операция может выполняться одновременно с вычислением функций насыщения и обеспечивает уменьшение числа выходных векторов.

6. ПОСТРОЕНИЕ МНОГОПРОЦЕССОРНЫХ НЕЙРОННЫХ ВЫЧИСЛИТЕЛЕЙ НА БАЗЕ ПРОЦЕССОРА NM6403

Нейропроцессор NM6403 имеет следующие аппаратные средства, предназначенные для построения многопроцессорных систем на его основе:

- два байтовых коммуникационных порта CP0 и CP1, каждый из которых позволяет осуществлять обмен информацией между нейропроцессором и его абонентом, имеющим такой же порт, со скоростью 20 Мбайт в секунду;
- два программируемых интерфейса с 64-разрядными внешними шинами (локальной и глобальной), каждый из которых поддерживает три различных мультипроцессорных конфигурации внешней шины. Подключение к одной шине нескольких нейропроцессоров позволяет им обмениваться информацией через общую память, расположенную на этой же шине. Причем, подключение к одной шине двух нейропроцессоров осуществляется без использования дополнительной управляющей аппаратуры. Скорость обмена данными через общую шину может достигать 400 Мбайт в секунду.

На рис.5 и 6 приведены примеры построения на базе NM6403 многопроцессорных систем, имеющих линейную (кольцевую) и матричную (тороидальную) структуры. При использовании коммуникационных портов для обмена информацией между двумя нейропроцессорами CP0 одного нейропроцессора должен быть подключен к CP1 другого нейропроцессора. При обмене информацией через общую память внешняя шина, соединяющая два нейропроцессора, должна быть локальной для одного нейропроцессора и глобальной для другого нейропроцессора.

Необходимо отметить, что функционально и электрически коммуникационные порты нейропроцессора NM6403 полностью совместимы с коммуникационными портами ЦПС TMS320C4x фирмы Texas Instruments [7]. Различные модификации данного ЦПС имеют от четырех до шести коммуникационных портов, что позволяет использовать этот процессор в качестве мощного коммутирующего элемента при создании на базе NM6403 мультипроцессорных нейронных систем сложных конфигураций.

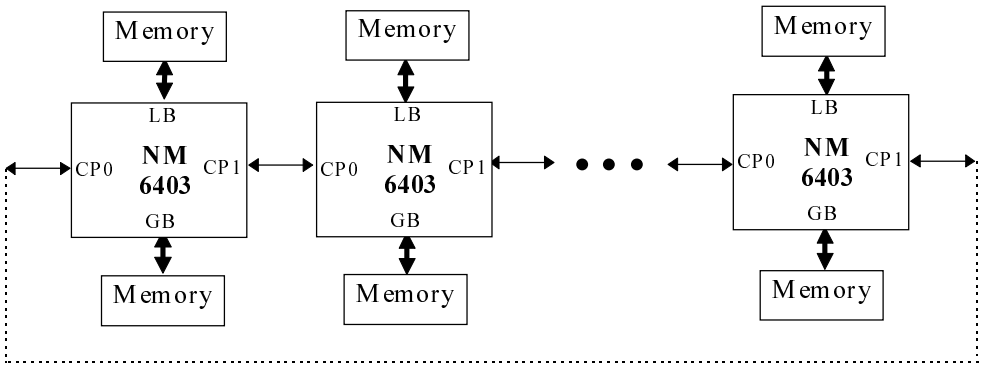

Рис. 5. Линейная (кольцевая) многопроцессорная структура
на базе процессора NM6403.

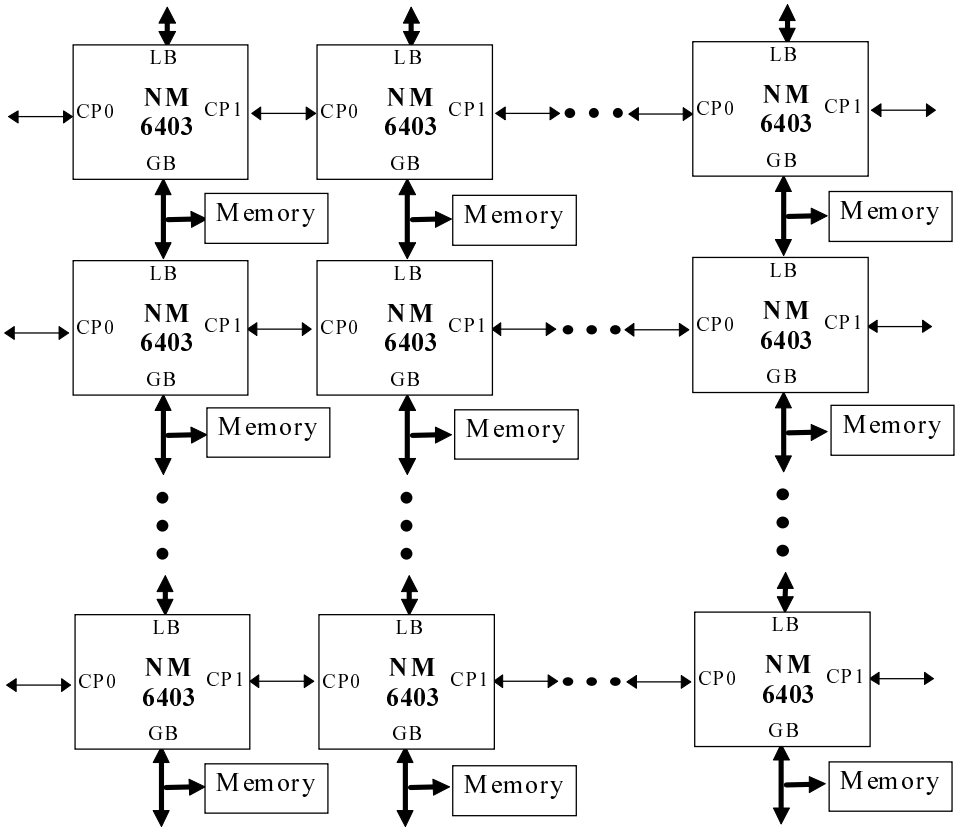

Рис. 6. Матричная многопроцессорная структура на базе процессора NM6403.

Система, состоящая из K нейропроцессоров NM6403 будет выполнять эмуляцию нейронной сети в K раз быстрее, чем один нейропроцессор. В предельном случае каждый фрагмент каждого слоя нейронной сети может эмулироваться отдельным нейропроцессором.

7. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ НЕЙРОПРОЦЕССОРА

Основной операцией, выполняемой при эмуляции нейронных сетей, является взвешенное суммирование. Поэтому производительность нейропроцессоров обычно оценивается по значению параметра “количество соединений в секунду” (CPS) [8]. Под CPS авторы подразумевают количество операций умножения с накоплением, выполняемых за одну секунду. В общем случае максимальное значение CPS для нейропроцессора NM6403 зависит от разрядности входных данных N_X и разрядности весовых коэффициентов N_W:

$$CPS = \left\lfloor \frac{64}{N_X} \right\rfloor \times \left\lfloor \frac{64}{N_X + N_W + \left\lceil \log_2 \frac{64}{N_X} \right\rceil} \right\rfloor \times f \qquad \text{при } N_X, >1 \text{ и } N_W > 1,$$

$$CPS = \left\lfloor \frac{64}{N_X} \right\rfloor \times \left\lfloor \frac{64}{N_X + \left\lceil \log_2 \frac{64}{N_X} \right\rceil} \right\rfloor \times f \qquad \text{при } N_X, > 1 \text{ и } N_W = 1,$$

$$CPS = 32 \times \left\lfloor \frac{64}{N_W + 5} \right\rfloor \times f \qquad \text{при } N_X = 1 \text{ и } N_W > 1,$$

$$CPS = 1024 \times f \qquad \text{при } N_X = N_W = 1,$$

где $f = 50*10^6$ Гц - тактовая частота процессора.

Результаты вычисления CPS нейропроцессора NM6403 для ряда наиболее распространенных значений N_X и N_W представлены на рис.7 в виде гистограмм и свидетельствуют о его высокой производительности. Для сравнения в табл.1 приведены основные параметры известных нейропроцессоров, детальный анализ которых дается в работе [8].

Таблица 1. Производительность цифровых нейрочипов.

Фирма, Тип	*Конфигурация*	*CPS* [1)]	*CPSPW* [2)]	*CPPS* [3)]	*CUPS* [4)]	*Patterns/s*
Nuralogix, NLX-420	32-16, 8-bit mode	10M	20k	640M	na	20k
Hecht-Nielson, 100 NAP	4-chips, 2M wts, 16bit Mantissa	250M	125	256G	64M	na
Hitachi, WSI	576 neuron Hopfield	138M	3,7	9,9G	na	na
Inova, N64000	64-64-1, 8bit mode	871M	3.4k, 128k wts	55.7G	220M	100k
IBM, ZISC036	64 8bit element input vectors	na	na	na	na	250k
MCE, MT19003	4-4-1, 32MHz	32M	32M	6.8G	na	140k
Micro Devices, MD-1220	8-8	8.9M	1.1M	142M	na	139k
Nestor/Intel, Ni1000	256 5bit element input vectors	na	na	na	na	40k
Philips, Lneuro-1	1-chip, 8bit mode	26M	26k	1.6G	32M	na
Siemens, MA-16	1-chip, 25MHz	400M	15M	103G	na	40k
RC Module, NM6403	**8bit mode, 50MHz**	**1200M**	**150M**	**76.8G**	**na**	**na**

Примечания:

1) CPS - количество соединений в секунду.

2) CPSPW = CPS/Nw Nw - количество синапсов в нейроне.

3) CPPS - количество соединений примитивов в секунду
CPPS = CPS*Bw*Bs, где Bw, Bs - разрядность весов и синапсов.

4) CUPS - Connection-Update-Per-Second.

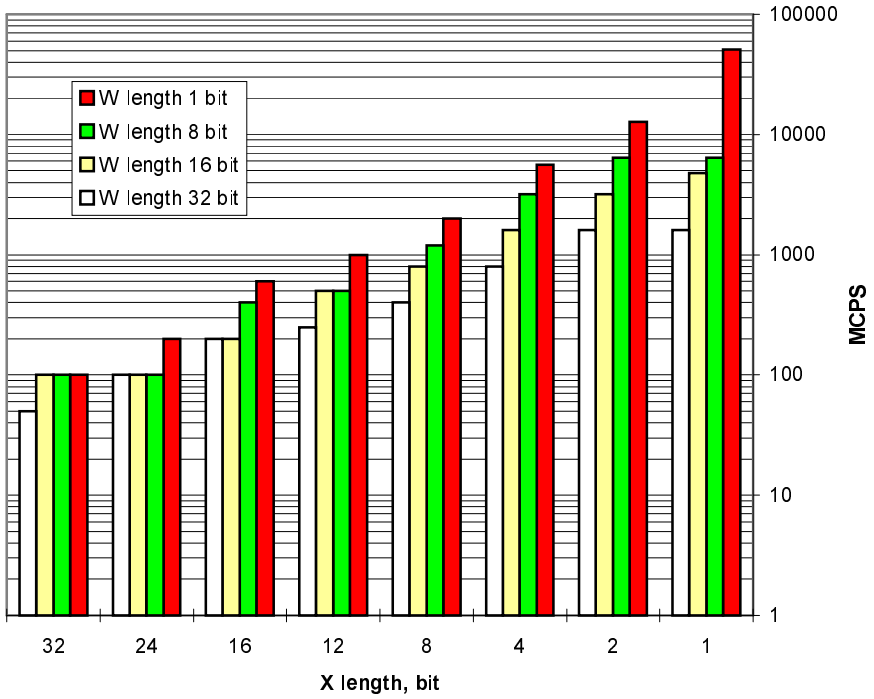

Рис.7. Пиковая производительность нейропроцессора в зависимости от разрядности входных данных и весовых коэффициентов

8. ЗАКЛЮЧЕНИЕ

Новый нейропроцессор NM6403 обладает уникальным свойством работать с данными переменной разрядности и способен увеличивать свою производительность с уменьшением разрядности операндов. Это даёт возможность программисту находить оптимальное соотношение между точностью вычислений и производительностью, которая на частоте 50 МГц может меняться в диапазоне от 50 MCPS (32-разрядные веса и входные данные) до 51.2 GCPS (однобитовые веса и входные данные).

Высокая производительность NM6403 и наличие развитого интерфейса позволяют использовать данный нейропроцессор для нейроускорителей в персональных ЭВМ, а также как базовый элемент при построении больших нейрокомпьтерных параллельных вычислительных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Jan N.H. Heemskerk, “Neurocomputers for Brain-Style Processing, Design, Implementation and Application”, PhD thesis, Unit of Experimental and Psychology Leiden University, The Netherlands, 1995.
2. P. Ienne, G. Kuhn, “Digital Systems for Neural Networks”, In P.Papamichalis and R.Kerwin, editors, *Digital Signal Processing Technology*, volume CR57 of *Critical Reviews Series*, p. 314-45, SPIE Optical Engineering, Orlando, Fla., 1995.
3. NLX420 Data Sheet, June 1992, Neurologix, Inc., 800 Charcot Av., Suite 112, San Jose, Ca. USA.
4. D. Hammerstrom, “A VLSI Architecture for High Performance, Low Cost, On-chip Learning”, Proc. Int. Joint Conf. On Neural Networks IJCNN`90, June 1990, vol.II, pp.537-544, San Diego, Ca, USA.

5. N.Mauduit, M.Duration, J.Gobert, “Lneuro 1.0: A Piece of Hardware LEGO for Building Neural Network Systems”, IEEE Trans. On Neural Networks, vol.3, no. 3, pp. 414-422, May 1992.
6. П.Е.Виксне, Д.В.Фомин, В.М.Черников, “Однокристальный цифровой нейропроцессор с переменной разрядностью операндов”, *Известия Вузов, Приборостроение*, 1996, т.39, №7, с.13-21.
7. TMS 320C4x User’s Guide, August 1993, Texas Instruments Inc., USA.
8. C. S. Lindsey, Th. Lindblad, “Survey of Neural Network Hardware”, Proc. SPIE Vol. 2492, p. 1194-1205, Applications and Science of Artificial Neural Networks, Steven K. Rogers; Dennis W. Ruck; Eds. , March 1995.