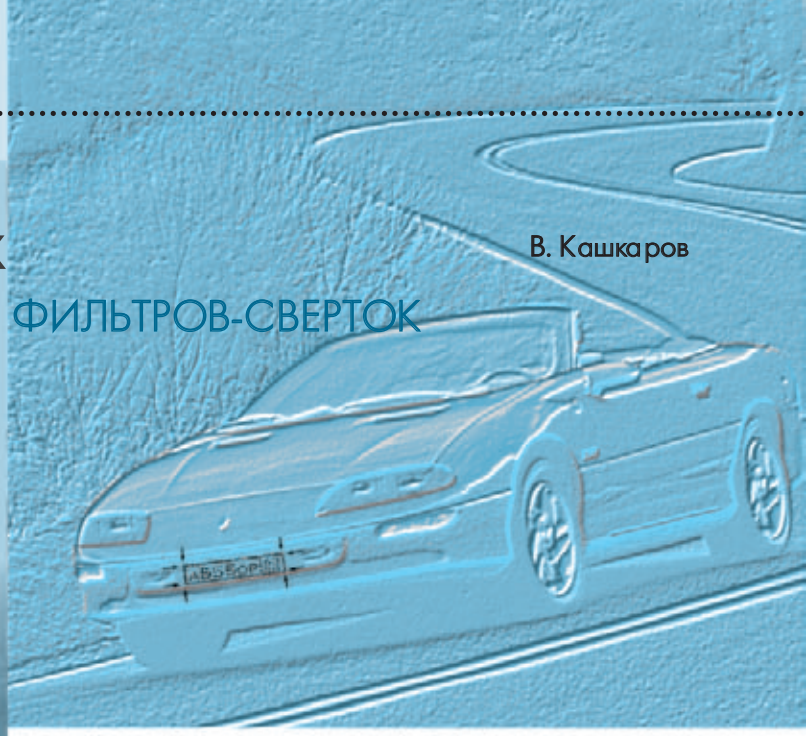
Ядро NeuroMatrix

ЭФФЕКТИВНАЯ РЕАЛИЗАЦИЯ ФИЛЬТРОВ-СВЕРТОК

В. Кашкаров



Мы уже рассказывали об отечественном процессорном ядре NeuroMatrix Core (NMC), разработанном НТЦ "Модуль" для обработки изображений и сигналов, а также для аппаратной поддержки искусственных нейронных сетей [1–3]. На сей раз один из ведущих специалистов НТЦ "Модуль" описывает реализацию фильтров-сверток на базе ядра NMC. Данные фильтры являются основополагающим элементом алгоритмов обработки изображений и используются в основном для выделения границ объектов и перепадов яркости. Эффективная реализация таких фильтров особенно важна для различных систем распознавания, работающих в реальном времени. Специфика архитектуры NMC как нельзя лучше подходит для решения задач данного класса.

АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ NEUROMATRIX CORE

Напомним, ядро NMC состоит из оригинального 32-битного RISC-ядра и 64-битного векторного сопроцессора VCP. В отличие от других современных DSP и микропроцессоров, таких как Pentium MMX (Intel), AltiVec PowerPC G4 (Motorola) и TigerSHARC (Analog Devices), NMC позволяет выполнять арифметические и логические операции, операции насыщения над векторами и матрицами *различной разрядности.* На основе NMC НТЦ "Модуль" разработал процессор цифровой обработки сигналов NM6403, выполняющий операции над числами с фиксированной точкой. Процессор производится по 0,5-мкм КМОП-технологии на фабрике фирмы Samsung Electronics.

Одно из главных достоинств NMC – это возможность обрабатывать 64-битные данные, разделенные на элементы различной разрядности. Например, одновременно обрабатываются восемь 8-битных элементов, либо шесть 10-битных и один 4-битный, либо два 32- битных элемента и т.д. Программист сам может выбирать между точностью и производительностью вычислений. Наивысшая производительность – 14400 MMACs (миллионов операций умножения с накоплением, MAC – multiplication and accumulation) в секунду – достигается при 1-битных операндах на частоте 50 МГц. При 32-битных операндах и 64-битном результате точность вычислений максимальна, но производительность составит 50 MMACs в секунду.

Основной элемент векторного сопроцессора VCP – устройство умножения, которое выглядит, как матричный умножитель (рис. 1). Оно содержит 1-битные ячейки памяти, окруженные логическими элементами. Эти ячейки можно объединять в несколько макроячеек с помощью двух программируемых 64-битных регистров: DB (data boundary) и MB (MAC boundary), которые определяют границы соответственно между строками и столбцами устройства умножения. Каждая макроячейка выполняет умножение входных данных на записанный в нее заранее весовой коэффициент W_i. Результаты умножения аккумулируются вдоль столбцов в соответствии с формулой $Y_i = U_i + \sum_j X_i W_{ij}$. Все вычисления устройства умножения

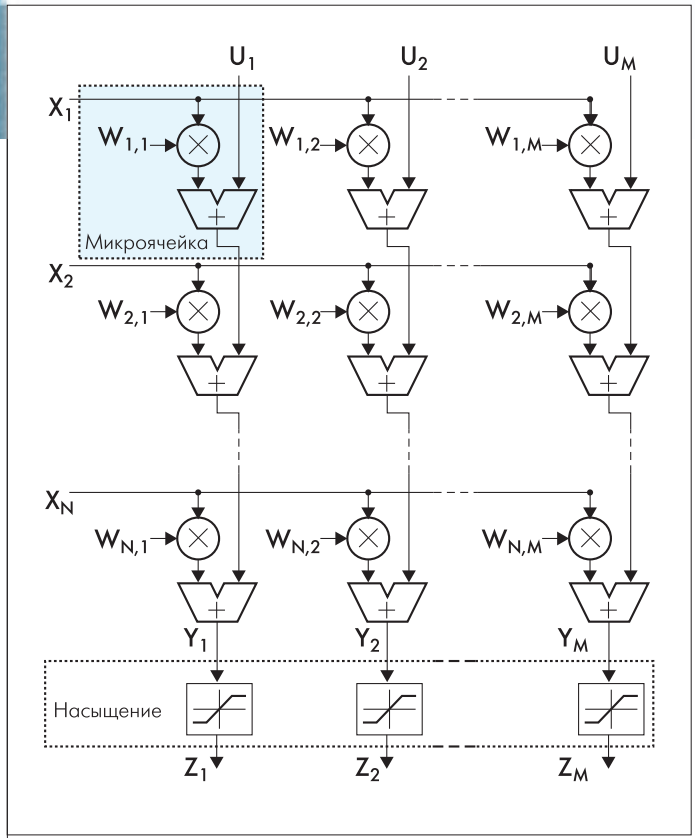


Рис. 1. Узел умножения NMC

0	0	0	0	-1	0	0	0	0
0	0	-1	-1	-2	-1	-1	0	0
0	-1	-2	-3	-4	-3	-2	-1	0
0	-1	-3	-7	-8	-7	-3	-1	0
-1	-2	-4	-8	136	-8	-4	-2	-1
0	-1	-3	-7	-8	-7	-3	-1	0
0	-1	-2	-3	-4	-3	-2	-1	0
0	0	-1	-1	-2	-1	-1	0	0
0	0	0	0	-1	0	0	0	0

Рис. 2. Пример маски 9x9

выполняются за один процессорный такт. Запись весовых коэффициентов в макроячейки (запись матрицы весовых коэффициентов) происходит за 32 такта. Однако кроме основной матрицы весовых коэффициентов в NMC есть и теневая, которая может заполняться также за 32 такта, причем в фоновом режиме. А переключение с основной на теневую матрицу происходит за 1 такт.

ФИЛЬТР-СВЕРТКА

Алгоритм свертки лежит в основе огромного количества различных фильтров, таких как, например, определение границ объектов, фильтры перепадов яркости, сглаживание изображения и т.п. Принцип фильтра-свертки сводится к наложению на окрестность каждого пиксела маски размером $n \cdot n$, где n – нечетное число. Маска содержит коэффициенты m_{ij}, которые подбираются так, чтобы их сумма была близка к нулю (рис. 2). Все пикселы, находящиеся в области $n \cdot n$, умножаются на соответствующие коэффициенты, суммируются, и результат присваивается центральному пикселу окрестности. Таким образом, для маски 9·9 новое значение пиксела в точке (x,y) определяется формулой $d(x, y) = \sum_{i,j=-4}^{4} s(x+i, y+j)\, m_{i+5\, j+5}$, где $s(x,y)$ – исходное значение пиксела. При маске 9·9 для кажого пиксела необходима 81 операция умножения с накоплением.

Рис. 3. Пример обработки изображения фильтром-сверткой с маской 9x9

В результате применения фильтра-свертки при подобранных соответствующим образом коэффициентах возрастает контрастность изображения. Применив к обработанному изображению пороговую функцию, можно выделить контуры объектов, что крайне важно для ряда задач, например распознавания образов (рис. 3).

РЕАЛИЗАЦИЯ ФИЛЬТРОВ-СВЕРТОК НА NMC

Существует несколько путей реализации фильтров-сверток на NMC. Их выбор зависит от структуры и размера маски, разрядности весовых коэффициентов и входных данных. Рассмотрим обработку 8-битных входных данных как наиболее популярного формата цифровой видеоинформации.

Сразу оговоримся, что предлагаемый метод работает, если промежуточные результаты подсчета свертки 9·9 – не более 16 бит. Это допустимо, так как большинство масок имеют положительные и отрицательные коэффициенты, сумма которых колеблется около нуля. Алгоритм предусматривает, что исходные данные располагаются в памяти в виде одномерного массива, строка за строкой. Очевидно, что свертка не применима к пикселам из первых и последних четырех строк и столбцов. Однако специфика NMC такова, что проще просчитать значения свертки для всех пикселов, а затем отбросить лишние данные.

В NMC имеется буфер на 32 64-битных слова, позволяющий хранить и многократно использовать результаты вычислений без обращения к внешней памяти. Этот буфер организован по принципу FIFO (first input – first output) и называется afifo (с накоплением). Размер afifo определяет подход к обработке. Поскольку разрядность промежуточных результатов – 16 бит, в одном слове

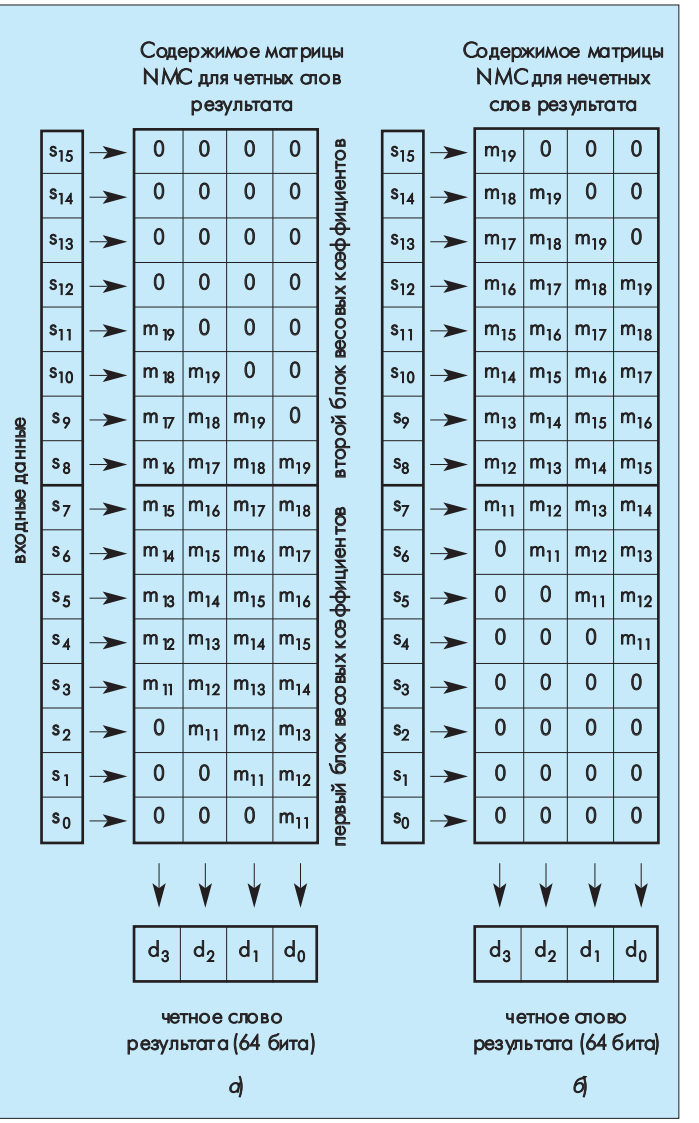


Рис. 4. Матрица весовых коэффициентов NMC для одной строки маски 9x9

результата сумматора можно записать значения четырех результатов свертки. В буфер afifo помещается 32·4=128 результатов. Таким образом, обработка производится для групп из 128 пикселов. Сначала вычисляется свертка для одной строки маски для 128 пикселов, затем – для другой, и так – для всех девяти строк. При этом результаты постепенно накапливаются в afifo. Наконец, значения свертки 128 пикселов переписываются во внешнюю память и начинается расчет для следующих 128 пикселов.

Во внешней памяти изображение представляет собой одномерный массив. Отдельно хранятся указатели на начала строк изображения и их длина. В умножитель поступают первые восемь 8-разрядных значений пикселов s_0–s_7. В матрицу к этому моменту загружают 8-разрядные весовые коэффициенты первой строки маски m_{11}–m_{18}, сгруппированные, как показано на рис. 4а, – первый блок коэффициентов. В результате перемножения получатся частичные

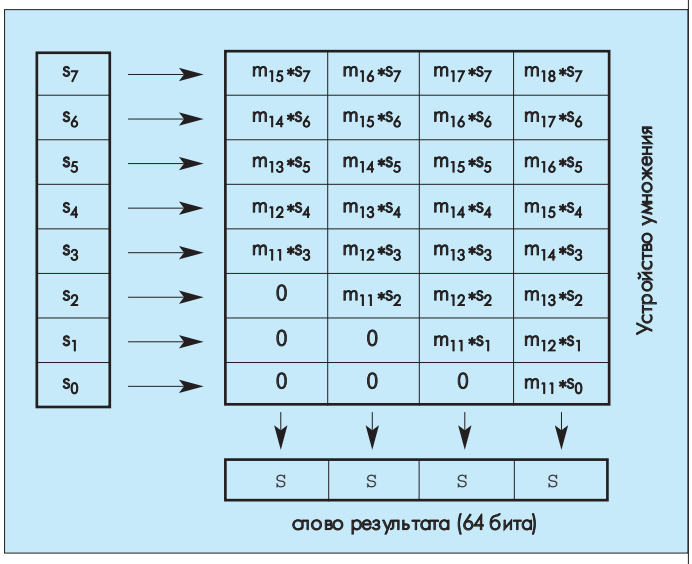


Рис. 5. Обработка данных в устройстве умножения

значения первой строки свертки для четырех пикселов d_0–d_3 (рис. 5). Чтобы закончить вычисление для первой строки маски, необходимо значения пикселов s_8–s_{11} перемножить на соответствующие весовые коэффициенты m_{16}–m_{19}, сгруппированные во второй блок весовых коэффициентов. Однако, поскольку NMC не позволяет постоянно менять местами основную и теневую матрицы весов – это можно сделать лишь один раз, – все вычисления проводятся сначала для первого блока весовых коэффициентов, затем – для второго. Таким образом, загружается первое 64-битное слово входных данных (s_0–s_7), вычисляются фрагменты свертки d_0–d_3, результат попадает в буфер afifo. Затем загружается второе слово (s_8–s_{15}) входных данных, и с тем же набором весовых коэффициентов вычисляются частичные значения для пикселов – внимание! – d_8–d_{11}. (Результаты сгруппированы по четыре в слово результата сумматора – всего 64 бита. Вычисления проводятся для "четных" слов – 0, 2, 4 и т.д., чтобы не изменять матрицу весовых коэффициентов, – рис. 6.) Результат опять попадает в afifo. Данный процесс повторяется 32 раза – пока не заполнится afifo.

В NMC за один такт одновременно происходит загрузка слова входных данных из внешней памяти, производятся все вычисления в устройстве перемножения с накоплением и запись 64-битного слова результата в afifo. Запись коэффициентов в теневую матрицу занимает 32 такта. Таким образом, пока заполняется afifo, в теневую матрицу в фоновом режиме записывается второй блок весовых коэффициентов.

После вычисления 128 промежуточных результатов теневая матрица за один такт переписывается в рабочую. Затем все вычисления повторяются для тех же входных данных, но сдвинутых на одно 64-битное слово (не s_0–s_{255}, а s_8–s_{263}), а результат умножения с накоплением добавляется к старому содержимому afifo. После выполнения описанных действий в afifo накапливаются 128 результатов свертки для первой строки маски. Пока в матричном умножителе ведутся вычисления, в теневой матрице формируется первый блок весовых коэффициентов для второй строки (m_{21}–m_{28}). Затем все действия повторяются, как было описано, но для входных значений s_{0+L}–s_{255+L}, где L – длина строки изображения в пикселах. Так повторяется для всех девяти строк маски. В итоге в afifo накапливаются полностью просчитанные значения свертки для 128 пикселов, сгруппированные в слова по четыре: d_0–d_3, d_8–d_{11},..., d_{8i}–d_{8i+3}, где $i=\overline{0, 120}$ – "четные слова результата". При этом данные, циркулируя в afifo, совершают 18 проходов.

Точно так же вычисляются и "нечетные" слова результатов – d_4–d_7, d_{12}–d_{15},..., d_{8i+4}–d_{8i+7}. Для них используются такие же блоки весовых коэффициентов, но сдвинутые вверх на четыре строки (рис. 4б).

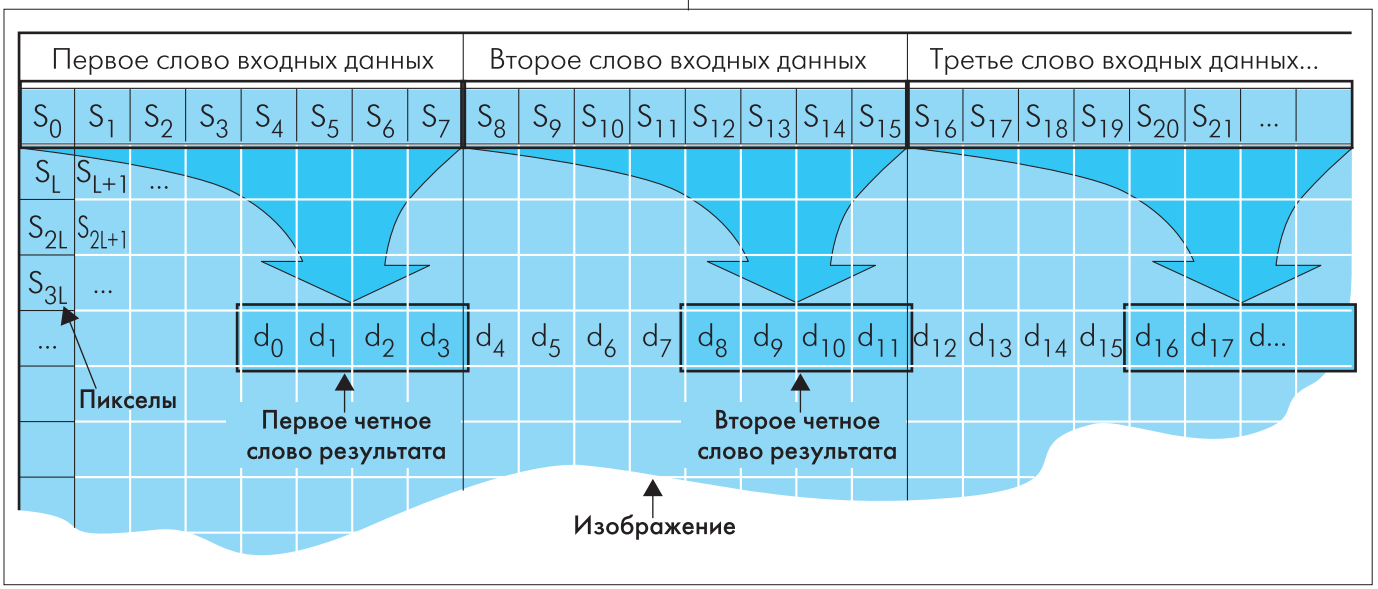


Рис. 6. Вычисление "четных" слов результата для одной строки маски

Для доступа к внешней памяти у NMC есть две независимые шины. Это позволяет ядру максимально увеличить производительность, поскольку в процессе вычислений одна шина используется для загрузки входных данных, а другая – для загрузки весовых коэффициентов. Поскольку промежуточные результаты накапливаются во внутреннем буфере, выходной поток в 18 раз ниже входного.

Несмотря на всю сложность изложенного алгоритма, его легко описать на языке ассемблера NMC. Например, SIMD команда

rep 32 data = [ar0++] with vsum, data, afifo;

загружает из внешней памяти 32 64-битных слова, посылает их на устройство умножения VCP, производит умножение с накоплением и добавляет результаты этого этапа вычислений к результатам предыдущих вычислений, хранящихся в буфере afifo. Заметим, что программы достаточно компактны: для большинства задач достаточно 40–60 строк кода [4].

Помимо матричного умножителя, в состав NMC входят еще несколько вычислительных узлов, и среди них – устройства маскирования и активации. Они располагаются в тракте обработки данных и, в зависимости от модификаторов в ассемблерной инструкции, либо пропускают данные без изменений, либо подвергают их обработке.

Устройство активации используют для операции отсечения по порогу, когда значения, превышающие заданный порог яркости, заменяются максимально возможными, а значения, лежащие ниже этого порога, – нулями. Например, если установлен порог 128, все меньшие значения после активации станут равны 0, а большие или равные – 255. После того, как изображение бинаризовано пороговой функцией (либо все нули, либо – единицы), объем обрабатываемой информации можно сократить за счет перехода от 16-разрядного представления пикселов к двухразрядному (00 или 01). Два бита – минимальная разрядность входных данных при обработке на матричном умножителе.

Устройство маскирования позволяет присваивать элементам вектора те или иные значения в зависимости от вектора маски. Оно реализует логическую функцию Z=(X and MASK) or (Y and not MASK) над каждым элементом исходных векторов *X* и *Y*. В результате элемент z_i примет значение x_i, если соответствующий элемент маски m_i=1. Если m_i=0, то z_i=y_i. Если в качестве маски использовать вектор входных данных видеоизображения, обработанный устройством активации, то с помощью маскирования пикселам можно присваивать те или иные заданные значения, а не только максимальное и нулевое.

Отметим, что на языке ассемблера NMC процедуры активации и маскирования описываются крайне просто. Так, команда

rep 32 with not activate afifo;

выполняет операцию активации над содержимым afifo. Порог задается в регистре

f1cr = 80808080h.

А команда

rep 32 data = [ar1] with mask afifo, data, ram;

производит маскирование, используя содержимое afifo в качестве маски.

Испытания на реальных приложениях показывают, что NMC обрабатывает изображения размером 512·512 8-битных пикселов фильтром-сверткой 9·9 за $1{,}4 \cdot 10^6$ тактов. При этом производится около 21 ММАС. Следовательно, в данной задаче NMC тратит примерно 5,4 такта на пиксел. Это означает, что NMC выполняет 14 MAC за один процессорный такт, что соответствует 28 скалярным операциям. Сравнение NMC со специализированным для обработки видеоизображений сигнальным мультипроцессором TMS 320C8x (MVP) фирмы Texas Instruments [5] также говорит в пользу отечественного процессорного ядра (см. таблицу). Реальная (не пиковая) производительность NMC на свертке 9x9 достигает более 3000 MIPS при частоте 100 МГц.

Производительность NMC и TMS 320C80 для различных размеров масок-сверток, тактов/пиксел

Размер маски	NMC	TMS 320C80 (ADSP*4)
3·3	1,8	2,1
5·5	2,6	7,3
7·7	4,3	–
9·9	5,4	–

РЕЗЮМЕ

Таким образом, NMC представляет новый класс процессорных ядер, предназначенных для обработки видео- и аудиосигналов, а также для аппаратной поддержки искусственных нейронных сетей [6]. Его способность обрабатывать элементы различной разрядности и масштабировать производительность позволяет разработчикам самим выбирать соотношение точности и эффективности, необходимое для выполнения конкретных задач.

Следует особо подчеркнуть, что процессор NM6403 допущен Министерством обороны РФ к применению в вооружениях и военной технике. В соответствии с отраслевым стандартом ему присвоено условное обозначение Л1879ВМ1.

В настоящее время НТЦ "Модуль" проводит разработку следующей версии процессора (рабочее название – NM6404). Этот процессор будет доступен в конце 2001 года. А в перспективе – новый процессор NM1281. NM6404 в отличие от NM6403 планируется производить по 0,25-мкм КМОП-технологии в корпусе типа HQFP (более удобен для пайки, чем BGA, в котором поставляется NM6403), рабочая частота увеличится с 40 до 100 МГц, появится внутренняя память (2 Мбита) и тестовый порт JTAG [2]. NM1281 будет характеризоваться более глубокими усовершенствованиями, основное из которых – увеличение разрядности векторных данных до 128 бит. Но главное, NM6404 и NM1281 обеспечат преемственность прикладного программного обеспечения, разработанного для NM6403. Это значит, что уже сейчас с помощью инструментальных средств для NM6403 можно разрабатывать прикладное ПО для будущих проектов, ориентируясь на применение NM6404 и NM1281.

ЛИТЕРАТУРА

1. **Peter Clarke.** Neural-Emulator IC Promises Scalability. – EETimes 1998, April 27, Issue 1004.
2. **Шахнович И.** Отечественный процессор цифровой обработки сигналов NM6403 – чудо свершилось. – ЭЛЕКТРОНИКА: НТБ, 1999, № 2.
3. **Patent 2131145 Russian Federation.** Neuroprocessor, Device for Saturation Functions Calculation, Computing Device and Adder. – RC Module; June 16,1998.
4. NeuroMatrixT NM6403. Описание языка ассемблера. – НТЦ "Модуль", 30002-01 35 02 А, 1999.
5. Implementation of an Image Processing Library for the TMS320C8X (MVP). – Texas Instruments Europe, BPRA059, July 1997.
6. **Chevtchenko P.A., Fomine D.V., Tchernikov V.M., Vixne P.E.** Using of Microprocessor NM6403 for Neural Net Emulation. – SPIE Proceedings, Vol. 3728, 1999.