

VLIW/SIMD NeuroMatrix® Core

Dmitri Fomine^a, Vladimir Tchernikov^a, Pavel Vixne^a and Pavel Chevtchenko^a

^aResearch Center MODULE, 3 Eight March 4th Street, Box 166, Moscow, 125190, Russia, tel. +7-095-152-9335, fax. +7-095-152-3168, e-mail: dfomine@module.ru

ABSTRACT

The paper represents architecture of the NeuroMatrix® Core (NMC) designed for image processing, signal processing and neural networks support [1,2]. The paper includes brief description of the core structure and instruction set. The NMC comprises an original 32-bit VLIW RISC processor and a 64-bit SIMD Vector co-processor (VCP). In contrast to other modern general purpose DSP and microprocessors: Texas Instruments c64xx, Intel Pentium MMX, Motorola AltiVec PowerPC G4 and Analog Devices TigerSHARC, the core performs variable bit-length vector/matrix arithmetic, logic and saturation operations. The base VCP operation is matrix by vector multiplication. The first DSP based on NMC is NM6403 supports shared memory mode for two 64-bit external data buses. Two byte-width communication ports simplify the multiprocessor system design. The NM6403 DSP has been designed by RC "Module" (Moscow), using SAMSUNG 0.5µm standard cell CMOS technology. The peak performance up to 14.400 MMACs (million multiplication and accumulations) has been achieved at 50MHz clock rate, 3.3V operating voltage and PBGA256 package.

1. INTRODUCTION

NMC is a high performance DSP core with elements of VLIW and SIMD architectures [3]. NeuroMatrix® architecture has a native support for 1-, 2-, 3- up to 64-bit data processing. Each of these data types is critical to the standard DSP algorithms, image processing, voice compression and next generation of wireless protocols. The flexible operands and ability to scale performance

let designers trade off precision and performance to suit their applications.

2. NMC STRUCTURE

NMC is intended for processing of 32-bit scalar data and variable bit length vector data packed into 64-bit data words. The block diagram is depicted in Fig. 1.

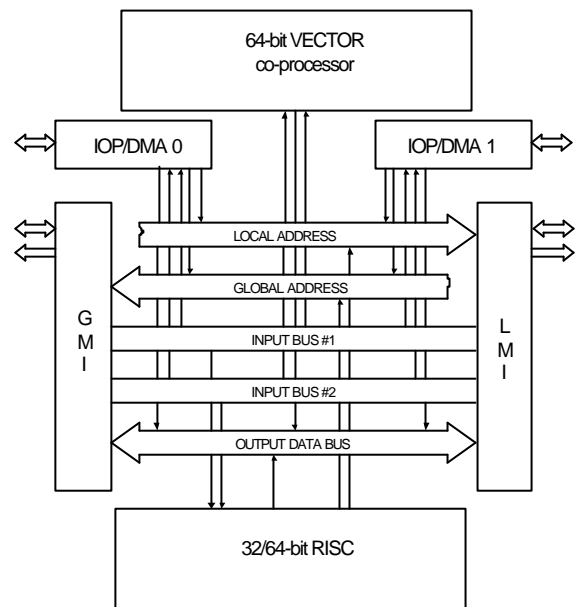


Fig. 1. Processor Block Diagram

NMC is comprised of the following functional units:

32-bit RISC processor - performs scalar arithmetical/logical and shift operations with 32-bit data and general control functions.

64-bit Vector Co-Processor (VCP) - performs arithmetical and logical operations with 64-bit vectors - packed words of variable bit length vector data.

LMI and **GMI** - two identical 64-bit programmable local and global interfaces to on-chip memory. The interfaces support SSRAM. Each interface supports up to two memory banks and can function in a "shared-memory" mode. The core supports 32-bit internal address. The total external memory range is 4Giga 32-bit words. This address space is divided into two equal parts: local and global (see Fig. 2).

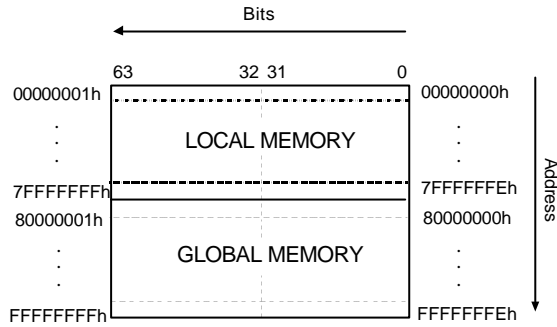


Fig. 2. Memory Map

IOP/DMA0 and **IOP/DMA1** - two IO ports provide bi-directional data transfer. Each IO port includes DMA unit that allows 64-bit data transfer between the port and the on-chip memory connected to the global and (or) local buses.

NMC has five buses for rapid data exchange between the functional units. Instruction fetch is performed by 64-bit words. Each word is a one 64-bit instruction or two 32-bit instructions.

2.1. RISC processor

The processor is a 5-stage pipelined 32-bit RISC with the original instruction set. It operates with 32- and 64-bit wide instructions (usually two operations are executed by each instruction). All internal units of the RISC are 32-bit wide.

2.2. Vector Coprocessor

The NeuroMatrix® architecture [4] provides the unique flexibility of choice of the desired level of performance and precision for 2-D MAC procedure:

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

According to application requirements, you can select the necessary length of operands and precision of products. The number of multiplications/accumulations depends on the length and number of operands. The highest performance - 14.400 MMACs is achieved with one-bit length operands at 50MHz clock rate. There is a possibility to increase the precision of calculation using any operand length up to 32-bit. In this case, the performance is 50 MMACs with a 64-bit result. The VCP includes an active matrix which looks like an array multiplier (Fig. 3).

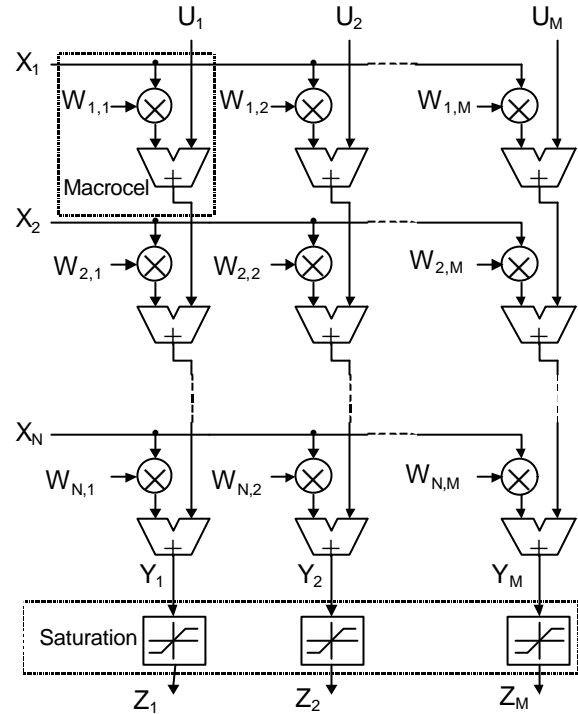


Fig. 3. Active Matrix

The structure comprises cells that include a 1-bit memory (flip-flop) surrounded by several logical elements. The software designer can combine the cells into several macrocells using two 64-bit programmable registers: DB - data boundary and MB - MAC boundary. These registers define the borders between rows and columns with macrocells. Each macrocell performs the

multiplication of variable input words using preloaded coefficients (W_i) and accumulates the result from the macrocells in the column above it. The columns simultaneously calculate the results in one processor cycle. The example of VCP configuration for 8-bit data (X_i) and coefficients (W_{ij}) processing is shown in Fig. 4. This results in a peak performance of 1.200 MMACs by parallel execution of 24 multiplication/accumulations with 21-bit results in one 20-nsec processor cycle.

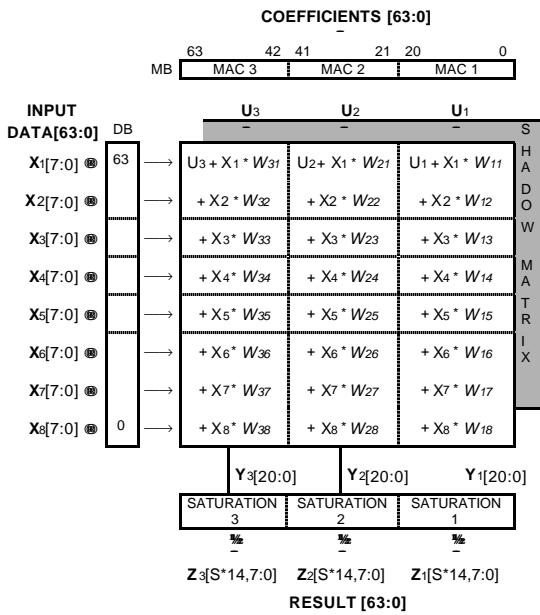


Fig. 4. NeuroMatrix® Engine

There is another interesting feature of using the active matrix. If the coefficients have binary values - "all 1" or "all 0", the matrix becomes a powerful switch (router). The remapping of bit positions of 64-bit input data word is performed in one processor cycle.

The number of multiplications/accumulations depends on the length and number of words packaged into a 64-bit block. The engine configuration can change dynamically during the calculations. You can start the application with maximum precision and minimum performance and then dynamically increase the performance by reducing the data-word lengths (Fig. 5).

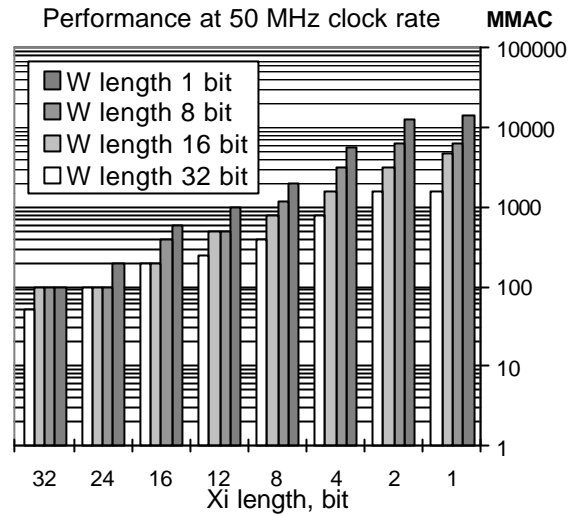


Fig. 5. Speed/Precision Trade-Off

The active matrix offers outstanding performance with "Boolean" arithmetic. So, 1-bit by 1-bit "Boolean" multiplication delivers 50.000+ MOPS at 50MHz clock rate.

To load new coefficients to the active matrix, 32 clock cycles are needed. To avoid the delays due to the coefficients refresh, the shadow matrix is used. The new coefficients are loaded to the shadow matrix in a background mode and then copy to the active matrix in one clock cycle.

To avoid arithmetic overflow, the saturation function (Fig. 6) with user-programmable saturation boundaries is used:

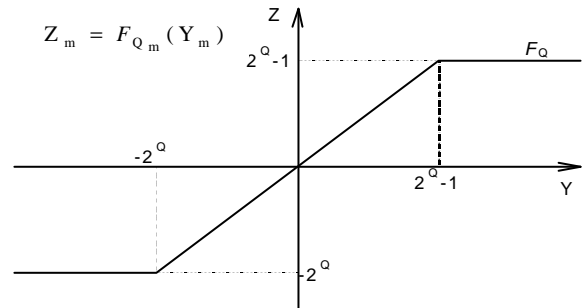


Fig. 6. Saturation Function

The saturation function reduces the number of significant bits of MAC products.

3. INSTRUCTION SET

The NeuroMatrix® instruction set is divided into two major types: Scalar Instructions (Fig. 7)

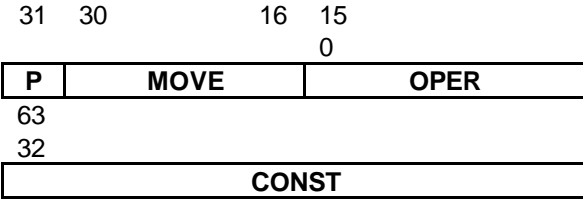


Fig. 7. Scalar Instruction Code

and Vector Instructions (Fig. 8).

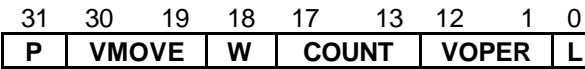


Fig. 8. Vector Instruction Code

The scalar instruction is RISC-like instructions which also support conditional branch, call, and return instructions. The NMC supports immediate with 32-bit addressing, base, indexed, and relative addressing. The vector instructions have special field to define the number (from 1 up to 32) of repeats of their execution. This solution allows to support short hardware loops and essentially increases the code density.

The assembly code of inner loop of the 3x3 Convolution Filter can be written as:

```

nbl = 80008000h;
sb = 02020202h;
<L>
rep 24 wfifo = [ar6++], ftw;
WTW_REG(gr2);
rep 32 data = [ar0++],ftw with vsum
, data, 0;
WTW_REG(gr2);
rep 32 data = [ar1++],ftw with vsum
, data, afifo;
WTW_REG(gr2);
rep 32 data = [ar2++] with vsum ,
data, afifo;
if > delayed goto L with gr7--;
ar6 = gr6;
rep 32 [ar4++gr4] = afifo;

```

In the first two lines the active matrix of 4 columns 16-bits each and 8 rows 8-bits each is initialized. In the third line the weights load to the shadow matrix. The next line copies the weights from the shadow matrix to the active one. Then calculation of sums of 3 elements in the first line of 3x3 mask is performed. The sixth line copies new weights from the shadow matrix to the active one. Then we calculate the sums of 3 elements in the second line of 3x3 mask and add them to the previous ones. The eighth line, copies new weights from the shadow to the active matrix. Then calculation of sums of 3 elements in the third line of 3x3 mask and their addition to the previous ones is performed. Finally, the tenth line contains a delayed conditional jump. The term "delayed" means that two more lines are executed before the jump itself occurs. The first of the delayed commands restores pointer to weights array in external memory, the second one stores the result of calculations to the memory.

Some benchmark comparisons between NMC and TI's C80 is shown in Table 1. The competitor's data are taken from [5].

Table 1. Convolution Filters

Convolution Filter	NM6403 Cycles/pixel	TMS320C80 Cycles/pixel
3x3	1.8	2.1
5x5	2.6	7.3
7x7	4.3	n/a
9x9	5.7	n/a

It is important to note that the number of cycles for NMC increases as a linear function.

The processor uses vector instructions for efficient work with packets of up to 32 64-bit data word. This type of instruction is the best for such operations as matrix-matrix, matrix-vector, or vector-vector multiplication, vector-vector addition/subtraction with saturation of results, block moving and so on.

The technique of using NeuroMatrix® Core for neural networks acceleration can be found in [6].

4. PIPELINE

The pipeline of NMC consists of a few sub-pipelines. Any scalar or vector instruction can use one or more sub-pipelines at a time. If there are free sub-pipelines, the next instruction can be fetched, decoded (one instruction per cycle) and executed. Using multicycle vector instructions up to four vector instructions and one scalar instruction may be executed simultaneously. Synchronization of different sub-pipelines is supported by special mechanism of locks. If some instruction needs a resource that is busy, a lock is formed and this instruction is stalled. If it uses a few sub-pipelines, all of these sub-pipelines are stalled, but other sub-pipelines may continue their work. So despite sequential fetch and decoding of instruction usage of vector instructions permits to achieve performance of superscalar processors. Out-of-order execution is also supported. High level of hardware utilization is achieved due to using the same apparatus both for scalar and vector instructions with small amount of control logic (about 5% of 80,000 total usable equivalent gates).

5. APPLICATIONS AND BENCHMARKS

Due to its flexibility and small number of eq. gates, NMC has a broad range of applications:

- digital signal processors (FFT, DFT, WHT);
- image processing (Convolution Filters, IDCT);
- neural net and vector/matrix accelerators;
- telecommunication chips (MPEG-4);
- embedded DSPs;
- basic block for building large super parallel SoC.

The results of running the standard DSP functions are shown in Table 2. The benchmarks were written in NMC assembler and compiled using NeuroMatrix® Software Development Kit. To run the benchmarks the NM1 PCI board was used.

The parameters of the functions are: Sobel Transform - frame size: 384×288 bytes, FFT 256-points - 32-bit data, Walsh Hadamard Transform (WHT) - 21 step, initial data - 5-bit.

Table 2. NM6403 Benchmarks

	Pentium II, 300 MHz	Pentium MMX, 200 MHz	TMS320C40, 50 MHz	NM6403, 40 MHz
Sobel (frs/sec)	n/a	21	6.8	68
FFT (usec)	200	n/a	464	102
WHT (sec)	2.58	2.80	n/a	0.45

6. SUMMARY

NMC is a new class of fixed point DSP cores. Its key element is NeuroMatrix® Engine that provides a programmable operand width and offers scalable performance from 1 MAC (32-bit inputs and coefficients) up to 288 MAC (1-bit inputs and coefficients) in one processor cycle. Due to its scalability, the size and power of NeuroMatrix® Engine can be changed very fast and easily, that is especially important in today's time-to-market demands.

7. REFERENCES

1. Peter Clarke, "Neural-emulator IC promises scalability", *EETimes* 1998, April 27, Issue 1004, pp. 37-38.
2. Peter Clarke, "Pact eyes multimedia, telecom", *EETimes* 1999, October 25, http://www.eet.com/story/core_competency/OEG_19991025S0005
3. Markus Levy, "1999 DSP-architecture Directory", *EDN Access* 1999, April 15, pp. 67-68, 102.
4. *Patent 2131145 Russian Federation*, "Processor, device for saturation functions calculation, computing device and adder", RC Module, June 16, 1998.
5. *Texas Instruments Europe*, "Implementation of an image processing library for the TMS320C8X (MVP)", BPRA059, July 1997.
6. P.A. Chevtchenko, D.V. Fomine, V.M. Tchernikov, P.E. Vixne, "Using of microprocessor NM6403 for neural net emulation", *SPIE Proceedings Vol. 3728*, 1999, pp.242-252.