

Neuroprocessor NeuroMatrix NM6403 architecture overview

P.A. Chevtchenko^a, D.V. Fomine^a, V.M. Tchernikov^a, and P.E. Vixne^a

^aRC Module, 3 Eight March 4th Street, Box 166, Moscow, 125190, Russia,
tel. +7-095-152-9335, fax. +7-095-152-4661,
E-mail: fomin@module.vympel.msk.ru

ABSTRACT

The paper represents an architecture overview of the NeuroMatrix NM6403 neuroprocessor designed for 32-bit and 64-bit data processing. The paper includes brief description of the neuroprocessor pinout, structure and functional units. The neuroprocessor comprise original RISC core, vector coprocessor (VCP) and some peripheral units. RISC core provides general control functions, 32-bit program and data address generation, 32-bit arithmetic, logic and shift operations. The main neuroprocessor operational unit is VCP, applied for variable bit-length vector data arithmetic, logic and saturation operations. The base VCP operation is matrix by vector multiplication with accumulation. Each data vector is a 64-bit word of packed data word. It is formed by set of variable bit length operands with user defined bit length in a range from 1 to 64 bits. Neuroprocessor includes two external 64-bit buses. The programmable memory interface units allow to use static or dynamic memory having wide range of time parameters without external controller. The neuroprocessor support shared memory mode for each of the external buses. With conjunction of two byte width communication ports this one makes it easy to design multiprocessor systems. Also this paper represents addressing modes, instruction set, supported interrupts. The neuroprocessor is designed using CMOS 0.5 μ m technology, power supply voltage is 3.3V, clock rate is 50MHz with one instruction per clock cycle performance.

Keywords: neural network, neuroprocessor, processor architecture, RISC-core, vector arithmetic, vector coprocessor.

1. INTRODUCTION

Neural network emulation are very time consuming tasks for ordinary sequential computers and the need for fast evaluation of large networks becomes more and more acute. The motivation for the architecture described in this paper was to develop hardware suitable to emulate a variety of neural network types by providing high multiplication rates in dedicated scalar and vector matrix on chip processing engine. The NM6403 neuroprocessor is a high performance microprocessor with static super scalar architecture [1]. The neuroprocessor high performance are achieved not by usage of super technology, but by state-of-the-art microprocessor architecture. Special attention was paid to making the device as cascadable as possible so that network of any structure and size can be build with several processors.

2. NEUROPROCESSOR STRUCTURE

Neuroprocessor is intended for processing of 32-bit scalar data and variable bit length vector data packed into 64-bit data words. The second data type will be named below as packed vector data. The neuroprocessor block diagram is shown in Fig. 1.

Neuroprocessor comprises the following functional units:

RISC CORE - the main functional unit that is intended for arithmetical and logical computations and shift operations over 32-bit data, 32-bit address calculations of data and instructions for memory access and for neuroprocessor activity control.

VCP - vector coprocessor that performs arithmetical and logical operations over 64-bit packed words of variable bit length vector data.

LMI and **GMI** - two identical programmable external memory interface units for 64-bit external buses each connected to the external memory with amount of up to 2^{31} 32-bit words. Neuroprocessor supports 32-bit and 64-bit memory access. In the second case two sequential memory words are accessed. Memory addressing is provided by page mode using the same 15-bit address bus in the time division mode for page address and address into the page. Page address appear at address bus while new page access only. Each programmable memory interface unit supports up to two memory banks without additional

external controller. Memory banks connected to one programmable memory interface may be of various types and may have different sizes and time parameters. Also each programmable memory interface supports three shared memory modes for easy design of multiprocessor systems.

CP0 and **CP1** - two identical communication ports each of which provides point-to-point data transfer via bi-directional byte width link between neuroprocessor and external device. The ports are intended for design of neuroprocessor based high-performance multiprocessor systems. Communication ports are compatible with those of TMS320C4x [2]. Each communication port comprises DMA unit that allows 64-bit data transfer between the port and the external memory connected to the global and (or) local buses.

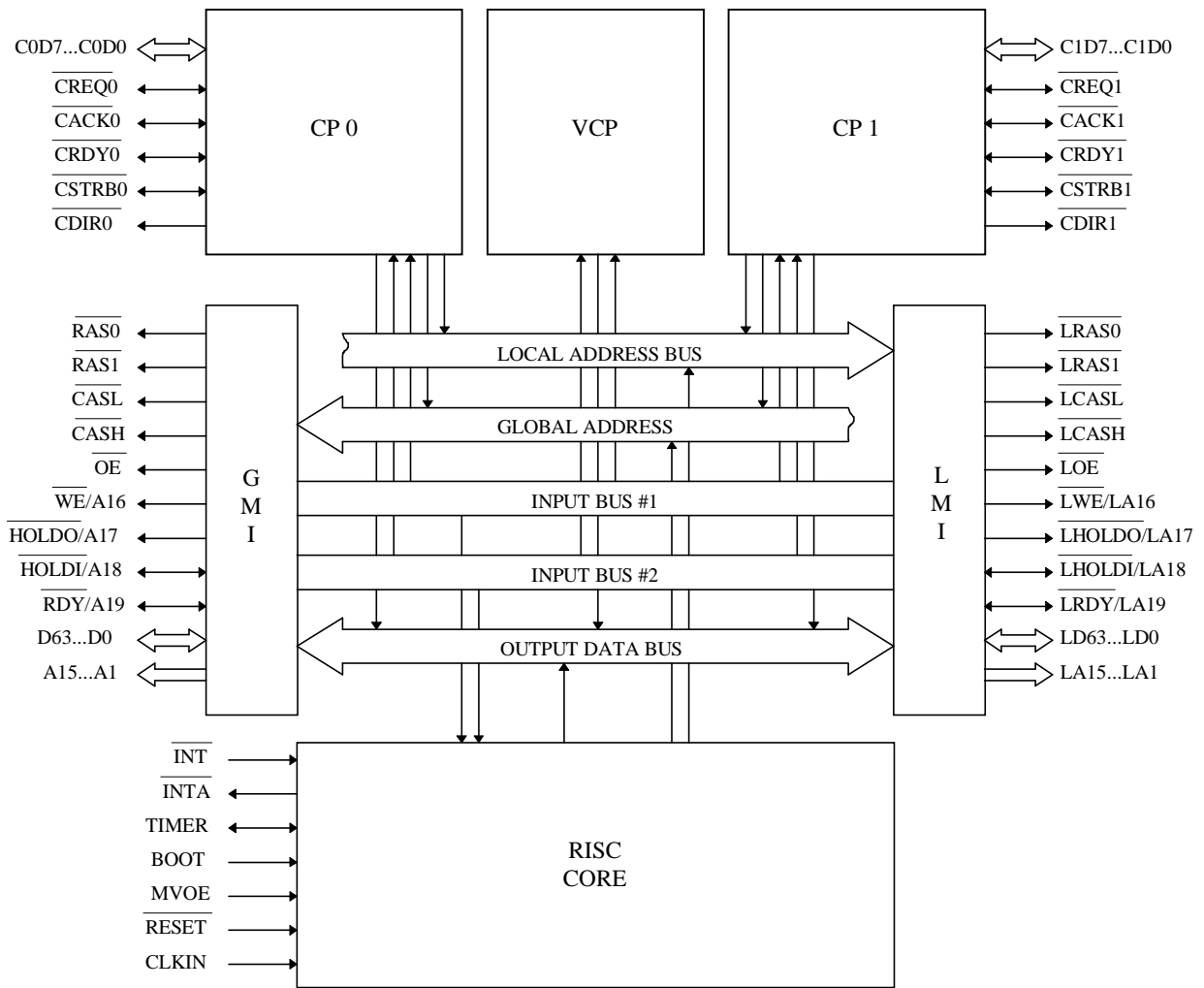


Figure 1. Neuroprocessor Block Diagram.

Neuroprocessor comprises five internal buses for rapid data exchange between the functional units:

LOCAL ADDRESS BUS and **GLOBAL ADDRESS BUS** are used to transfer instruction addresses generated by RISC core and data addresses generated by RISC core in program mode and by communication port in DMA mode to the respective programmable memory interface for memory access.

OUTPUT DATA BUS is used to transfer data that have to be written to external memory from RISC core, vector coprocessor and communication ports to the respective programmable memory interface.

INPUT BUS #1 and **INPUT BUS #2** are used to transfer instructions and data fetched from global or local external memory by the respective programmable memory interface unit to any of the main neuroprocessor functional units. The bus **INPUT BUS #2** is exclusively occupied by scalar data and **INPUT BUS #1** is exclusively occupied by vector data in the normal mode. Data transfers via DMA mode and instruction transfers can use any available input bus.

Data move from one scalar register to another and immediate constant load from instruction buffer to scalar register are provided by means of programmable memory interface units using internal buses **OUTPUT DATA BUS** and **INPUT BUS #2**.

Internal buses **INPUT BUS #1**, **INPUT BUS #2** and **OUTPUT DATA BUS** are 64-bit wide. 32-bit or 64-bit transfer operation/cycle via these buses is possible. To increase efficiency of data moving between 32-bit scalar registers and between 32-bit registers and memory these registers are grouped into 64-bit register pairs. Also neuroprocessor contains some 64-bit control registers. Thus it is possible to say that neuroprocessor supports 64-bit transfer operations of scalar data.

Instruction fetch is performed by 64-bit words. Each word is a one 64-bit instruction or two 32-bit instructions.

Neuroprocessor supports 32-bit internal address. The total memory range of the neuroprocessor is 4Giga 32-bit words. This address space is divided into two equal parts: local and global (see Fig. 2). If address MSB is equal to zero then there is an access to local memory, if address MSB is equal to one then there is an access to global memory. The address LSB is used for access to 32-bit data. If address LSB is equal to zero the data from low memory word (bits 31-0) are used, if address LSB is equal to one the data from high memory word (bits 63-32) are used. While accessing 64-bit data or fetching an instruction the internal address LSB is ignored.

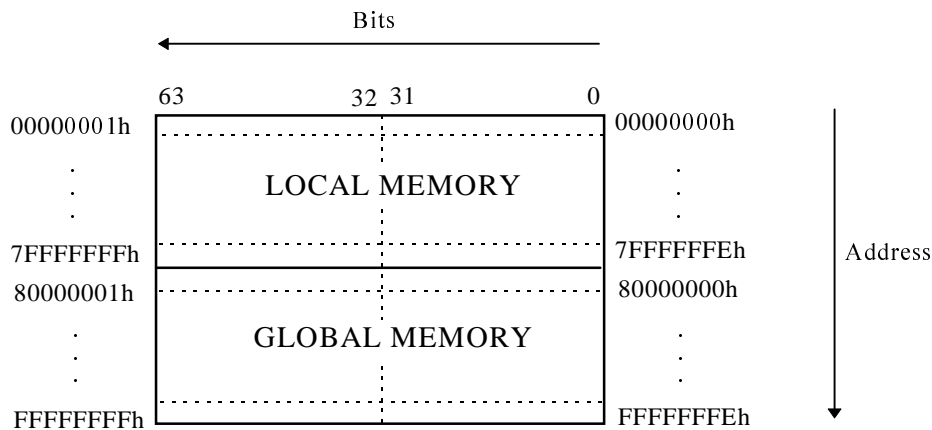


Figure 2. Memory Map

2.1. Neuroprocessor RISC Core

The neuroprocessor RISC core block diagram is shown in Fig. 3. It comprises four functional units and internal buses that intended for data exchange between RISC core units. All internal units and buses of RISC core are 32-bit wide.

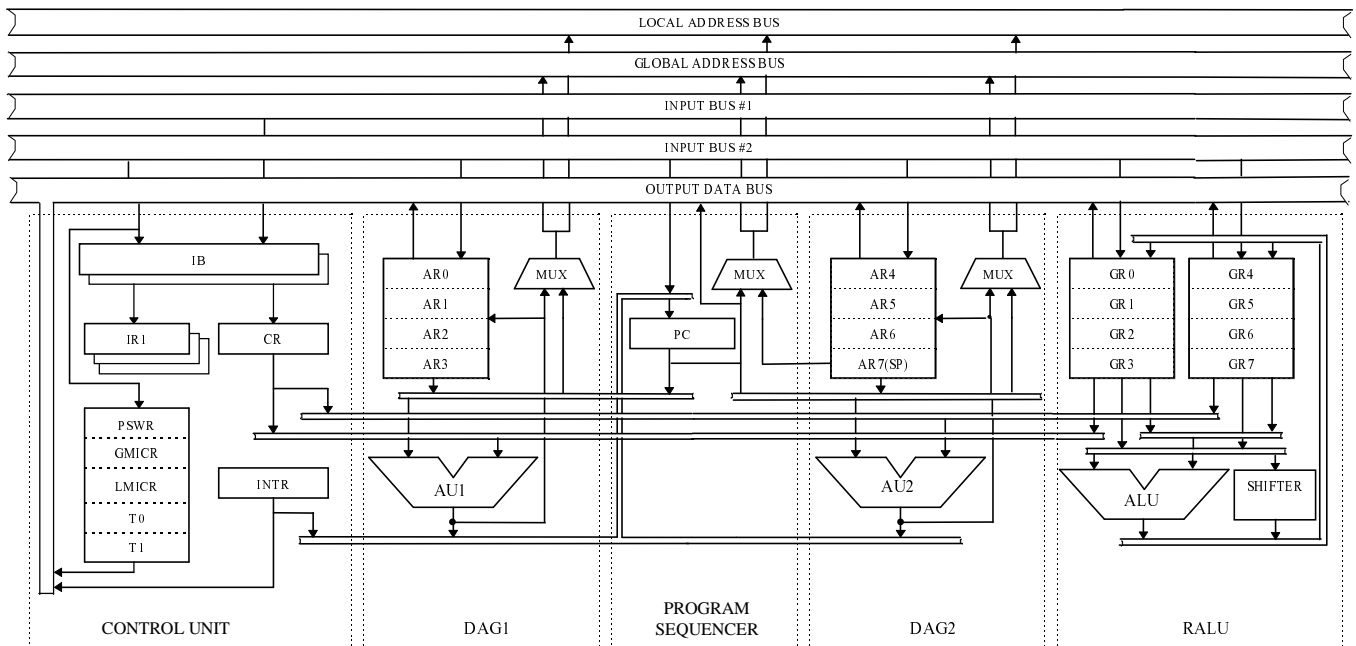


Figure 3. RISC core Block Diagram

2.1.1. Register and Arithmetic/Logic Unit

Register and arithmetic/logic unit (RALU) is intended for storage of up to eight 32-bit scalar data and for execution of shift operations and single-operands and double-operands arithmetical and logical operations over them. By execution of operations RALU forms some flags: zero result, negative result, carry and arithmetic overflow that can be fixed into program state word register for following reference by conditional branch instructions. The data stored in RALU also can be used as address or address displacement for some memory access or control instructions.

RALU comprises the following subblocks:

GR0, ... , GR7 - general purpose registers that are united into a register file. This register file has two input ports respectively connected to the neuroprocessor *INPUT BUS #2* and to the ALU or shifter result bus and three output ports respectively connected to *OUTPUT DATA BUS* and to the first and the second ALU operand buses. Also the registers GR0, ... , GR3 have an output connected to the second operand bus of AU1. That allows to use data stored in that registers when instruction or data address are formed at DAG1 or when modification of address registers occurs at DAG1. Registers GR4, ... , GR7 have an output connected to the second operand bus of AU2. That allows to use data stored in that registers when instruction or data addresses are formed at DAG2 or when modification of address registers occurs at DAG2.

ALU - arithmetic/logic unit that is able to provide one of sixteen logical or eleven arithmetical operations using data from one or two general purpose registers per one clock cycle. Arithmetical operations are conducted over two's complement 32-bit data.

SHIFTER - barrel shifter unit, that is able to provide left or right arithmetical or logical shift or rotate at any bit count for data placed from any general purpose register to the first ALU operand bus per one clock cycle.

2.1.2. Primary data address generator

Primary data address generator (DAG1) is intended for data address generation and instruction address generation for branch instructions. Also DAG1 provides storage and modification of up to four 32-bit data addresses, branch addresses or address displacements.

DAG1 comprises the following subblocks:

AR0, ... , AR3 - address registers that are united into a register file that has two input ports respectively connected to neuroprocessor *INPUT BUS #2* and to the AU1 outputs and two output ports respectively connected to neuroprocessor *OUTPUT DATA BUS* and to the first input operand bus of AU1.

AU1 - the first arithmetical unit to perform single operand or double operand arithmetical address calculations or modification of one of DAG1 address registers. Data from AR0, ... , AR3 or from PC can be used as the first operand and data from GR0 - GR3 or 32-bit immediate constant can be used as the second operand.

MUX - address multiplexor to control address outputting from AU1 output or from the first operand bus of AU1 to neuroprocessor internal address bus depending on the addressing mode. If MSB address value at multiplexor output is equal to 0, the address from DAG1 is placed to *LOCAL ADDRESS BUS*, otherwise the address from DAG1 is placed to *GLOBAL ADDRESS BUS*.

2.1.3. Secondary data address generator

The secondary data address generator DAG2 is structurally and functionally very similar to DAG1. The main specific feature of DAG2 is that one of its address registers AR7 also can be used as a stack pointer SP for interrupt processing and subroutine handling instructions.

2.1.4. Program Sequencer

PROGRAM SEQUENCER is intended for forming the address of the next 64-bit instruction or the address of the next pair of 32-bit instructions at linear program blocks when the next instruction address is defined from the current instruction address by an increment operation. Also PROGRAM SEQUENCER is used to form interrupt vector address and to modify stack pointer during call instructions.

PROGRAM SEQUENCER comprises the following subblocks:

PC - program counter that intended for storage of the address of a currently fetched 64-bit instruction or a pair of 32-bit instructions and calculating of the next instruction or instruction pair address by post-incrementing by 2. By branch instructions the new address data feed PC from AU1 or AU2 output. When interrupt is processed, address data is taken from register INTR and when return from subroutine or return from interrupt operation occurs, address data is taken from neuroprocessor *INPUT BUS #2*. PC output is connected to neuroprocessor *OUTPUT DATA BUS* that makes it possible to read the value from PC to user program.

MUX - address multiplexor that controls address outputting from PC incremented by 2 or from stack pointer AR7(SP) to neuroprocessor internal address bus. If MSB address value at multiplexor output is equal to 0 the address from PROGRAM SEQUENCER is placed to *LOCAL ADDRESS BUS*, otherwise the address from DAG2 is placed to *GLOBAL ADDRESS BUS*.

2.1.5. Control Unit

CONTROL UNIT - performs the initial analysis and decoding of instructions that are fetched from external memory. This unit forms control signals to all neuroprocessor functional units during pipeline instruction executing. It processes interrupt requests and performs request arbitration of the neuroprocessor resources such as external and internal buses for other neuroprocessor functional units. Also it controls external $\overline{\text{INTA}}$ and TIMER signals.

CONTROL UNIT comprises the following subblocks:

IB - instruction buffer that provides store and preliminary analysis up to four 32-bit or two 64-bit instructions, fetched from external memory.

IR1, ... , IR6 - pipeline stages instruction registers

CR – register for storage of the immediate constant of a current instruction

GMICR - global memory interface control register. This register defines global bus configuration, global memory banks boundaries, page size for each global memory bank, operational mode (sync./async.), memory type and time parameters for memory access cycles.

LMICR - local memory interface control register. This register is identical to GMICR, but it performs control under LMI.

T0, T1 –programmable timers. They are used to form interrupt signals and signals at external neuroprocessor output **TIMER** in a programmable time range. Operation mode for each timer (single or periodical) is specified by a corresponding bits of **PSWR** register. Also, if either **LMICR** or **GMICR** specifies usage of dynamic memory, then the timer **T1** will be used as dynamic memory refresh period counter.

PSWR - program state word register. This register is used for storage of the last **RALU** operation flags, interrupt mask and information of control timers **T0, T1** and external pin **TIMER** operational mode, communication ports initialization modes, input and output channel stop flags for communication ports **CP0** and **CP1**, **AFIFO** and **WFIFO** empty flags, shared memory control flags. This register is combined with the program counter **PC** and this register pair is stored at system stack when instruction call or interrupt occurs.

INTR - interrupt and DMA requests register. This register is used for storage of all requests for DMA from communication ports **CP0** and **CP1**, and all interrupt requests before they are processed. Additionally this register contains information about **AFIFO** and **WFIFO** state (full/empty), communication ports state (receive/tranceive), the amount of 64-bit words contained in **AFIFO** and **RAM**, and external bus ownership in shared memory mode. Register **INTR** are connected to neuroprocessor **OUTPUT DATA BUS** and available for read only.

All registers **PSWR, GMICR, LMICR, T0** and **T1** are available for user program for reading and writing. Their inputs are connected to neuroprocessor **INPUT BUS #2** and outputs are connected to neuroprocessor **OUTPUT DATA BUS**. Also there is hardware support for mask based setting or resetting of any bit or bits of **PSWR** by single instruction.

Neuroprocessor supports one external interrupt and 9 internal interrupts:

- two timer interrupts;
- arithmetic overflow during the operation at RISC core interrupt;
- illegal vector instruction interrupt;
- four interrupts concerned with communication port input or output operation finished;
- trace interrupt.

2.2. Vector Coprocessor

VCP is designed to operate with variable bit length data from 1 to 64 bit wide, stored into 64-bit packed data. **VCP** block diagram is represented in Fig. 4.

VCP comprises the following functional units:

OU - operational unit that performs arithmetic and logic operations under 64-bit packed data $\mathbf{X}=\{X_K \dots X_1\}$ and $\mathbf{Y}=\{Y_1 \dots Y_1\}$ taken from the respective **X** and **Y** inputs of **OU** and weight coefficients matrix **WOPER**, that feed the $\mathbf{W}_1, \dots, \mathbf{W}_J$ inputs as **J** 64-bit words of the packed weight coefficients $\mathbf{W}_1=\{W_{11} \dots W_{11}\}, \dots, \mathbf{W}_J=\{W_{J1} \dots W_{J1}\}$. Each operation result is formed as 64-bit packed data word $\mathbf{R}=\{R_1 \dots R_1\}$. The **I** value is to be in the range from 1 up to 64 depending on **NB2** register value (**I** is equal to the number of ones in **NB2**), and **J** value is to be in the range from 1 to 32 depending on **SB2** register value (**J** is equal to the number of ones in **SB2**). The **K** value depends on the executed vector operation: **K=I** - for arithmetic operations, **K=J** - for multiply-accumulate operations. Operation type depends on the instruction code. Operations are executed at pipeline with one operation per cycle throughput rate. **OU** does not generate any condition flags.

RCS - one bit right cyclic shifter. Depending on the instruction code, 64-bit data that feed **X** operand of **OU** may be processed by one bit right cyclic shifter **RCS**. This shift is performed at one clock cycle over packed 64-bit data as a whole.

SU1, SU2 - saturation units that are used for saturation of 64-bit packed words for both **X** and **Y** operands. For each unit **SU1** and **SU2** there are respective programmable control registers.

F1CR, F2CR - saturation control registers (**F1CR** for **SU1** and **F2CR** for **SU2**), whose value defines the number and the bit length of data at packed 64-bit data word and the absolute value of the saturation threshold.

SWITCH 3→2 - switch that selects two vector operands **X** and **Y** from three possible data sources. Depending on the instruction code each vector operand is all zeros operand or is selected from one of the following data sources:

- data from vector **RAM** via **VECTOR DATA BUS**;
- data from **AFIFO** via feedback bus of **VCP**;
- data from external memory.

WBUF, WOPER - weight coefficient memories. There are two memory matrixes WBUF and WOPER with the size of 32*64 bit each. These memories store weight matrix **W** as a J 64-bit words of packed weight coefficients: $W_1=\{W_{11} \dots W_{1j}\}, \dots, W_j=\{W_{j1} \dots W_{jj}\}$. WOPER stores weight matrix that is used in OU multiply-accumulate operations. WOPER outputs are directly connected to OU inputs and WOPER inputs are connected to WBUF outputs. This architecture allows to load data from WBUF into WOPER per one clock cycle according to LOAD instruction. Simultaneously with the WBUF to WOPER load the vector control registers SB2 and NB2 are loaded from respective SB1 and NB1.

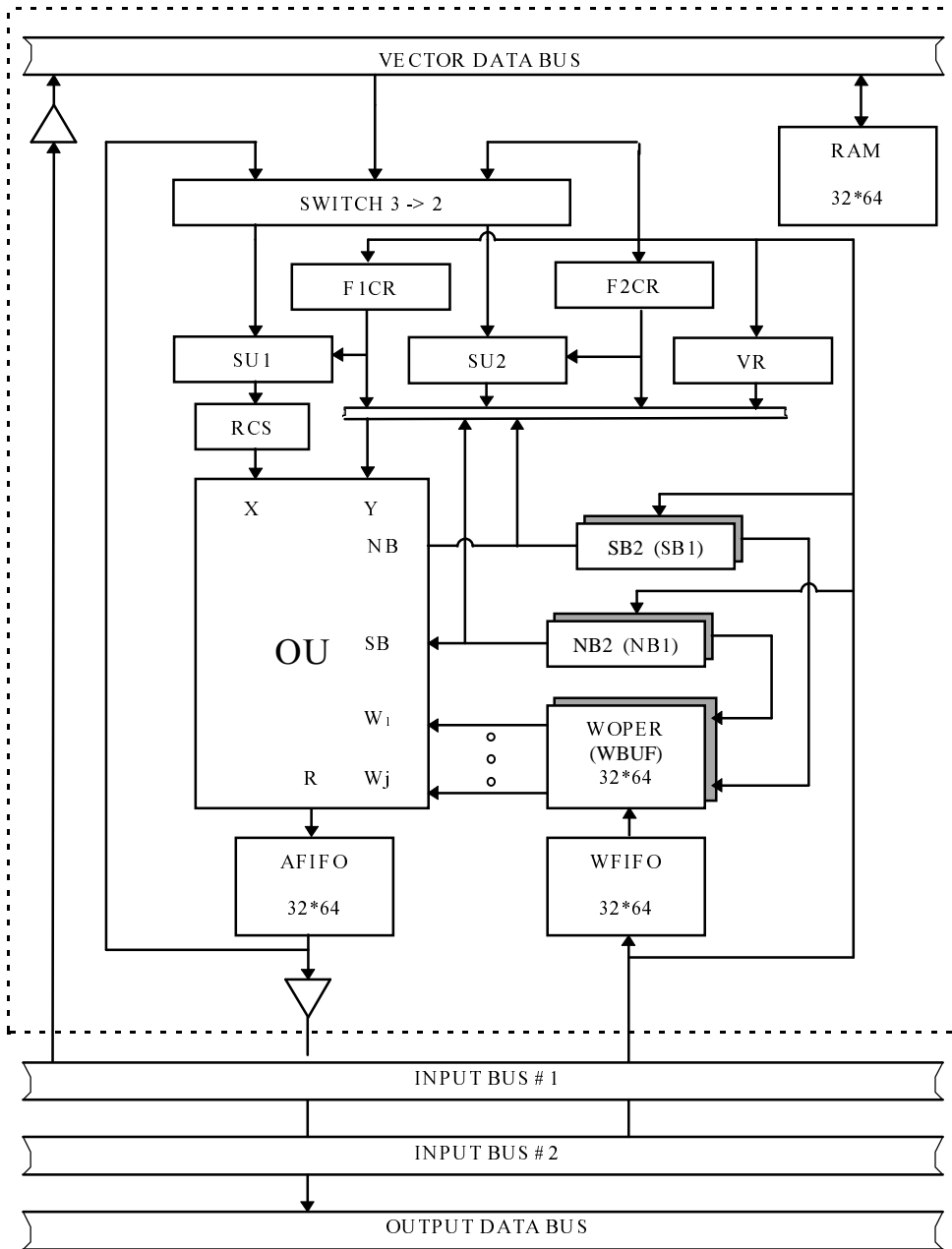


Figure 4. Vector Coprocessor (VCP) Block Diagram

WBUF is used to load new weight matrix from WFIFO while OU operates with the previous weight matrix values stored in WOPER. Weight load procedure to WBUF is initiated by single instruction and processed during 32 clock cycles. The

load is represented as sequential writing of J 64-bit packed weight coefficients into WBUF from WFIFO. The J is defined by SB1 value (J is equal to the number of ones in SB1). At a load mode WBUF works as a stack. The first loaded packed word will be $\mathbf{W}_1 = \{W_{11} \dots W_{1J}\}$, and the last loaded packed word will be $\mathbf{W}_J = \{W_{J1} \dots W_{JJ}\}$. Number of data and their bit length at packed 64-bit word are defined by NB1.

WFIFO - weight coefficient FIFO. Dual port WFIFO has the size of 32*64 bit and is used as a buffer during weight coefficients loading from external memory to WBUF. Read and write operations for WFIFO are processed under 64-bit packed weight coefficient words. The internal data bus *INPUT BUS #2* is used to load data into WFIFO.

AFIFO - accumulator FIFO. Dual port AFIFO has the size of 32*64 bit and is used at VCP as an accumulator for last vector operation result in packed 64-bit word format.

RAM - vector register that is a single port memory with the size of 32*64 bit connected to VCP *VECTOR DATA BUS*. Vector register RAM is similar to ordinary FIFO with the exception that data at vector register RAM may be read many times.

VR - bias register, that stores 64-bit bias word. There are some instructions that can use that register instead of Y operand in multiply-accumulate operation.

2.3. Programmable External Memory Interfaces

Neuroprocessor has two similar programmable external global and local memory interfaces. Each interface represents the following key features:

- separate 88 pins configuration, each with its own 64-bit data bus, up to 19 bit address bus that allow to output 30-bit address in a multiplexed mode;
- 64 bit data and 32 bit data access support (in the last case LSB of internal 32-bit address is used to address 32-bit data at 64-bit data word);
- one clock cycle for read/write memory access;
- shared memory arbitration signals;
- programmable bank and page mapping with up to two memory banks of different memory type support (SRAM, Flash ROM, DRAM, EDO DRAM);
- page mode support independently for any of memory bank;
- selectable wait states (can be programmed or external ready signal can be used);
- refresh cycles for DRAM and EDO DRAM hardware support (with programmable time parameters);
- interface signals set allows to work with different memory types without external memory controller.

Neuroprocessor supports both the single processor mode and the multiprocessor mode for any external bus. If external memory is shared between two neuroprocessors then bus arbitration can be processed without external memory controller.

An example of the bus configuration type 1 is shown in Fig. 5. The key feature of this configuration is that the access to memory banks (MEMORY BANK1, MEMORY BANK2) can be provided by only one neuroprocessor NP1 or NP2 at the same time.

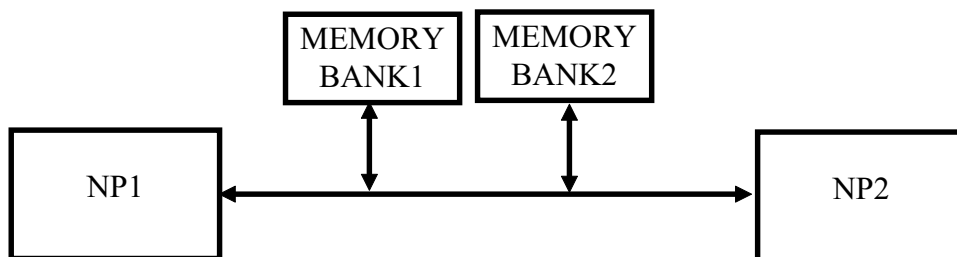


Figure 5. Bus Configuration Type 1 (Common Bank 0, Common Bank 1)

An example of the bus configuration type 2 is shown in Fig. 6. At this configuration each neuroprocessor has one own memory bank that is inaccessible from other neuroprocessor (NP1 has own MEMORY BANK1 and NP2 has own

MEMORY BANK2) and one memory bank (MEMORY BANK3) is common for both neuroprocessors. Neuroprocessors can gain access to this memory bank one by one through external buffers BUFFER1 and BUFFER2.

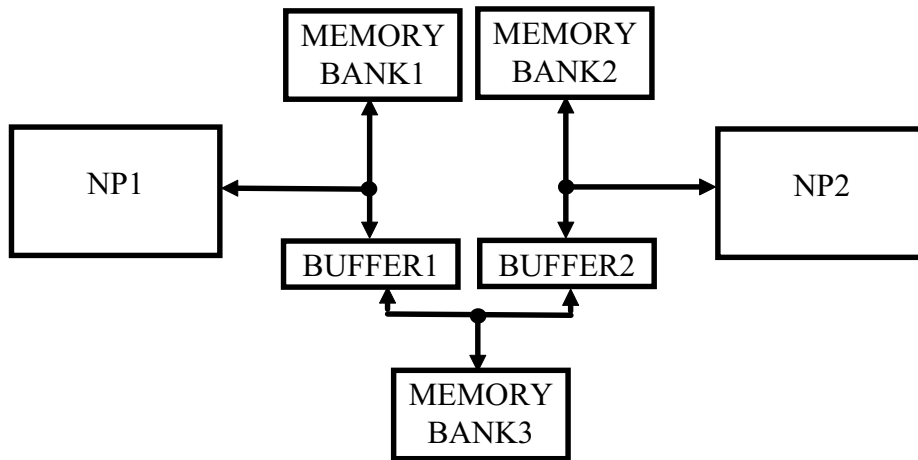


Figure 6. Bus Configuration Type 2 (Own Bank 0, Common Bank 1)

An example of bus configuration type 3 is shown in Fig. 7. At this configuration each neuroprocessor has one own memory bank (NP1 own MEMORY BANK1 and NP2 own MEMORY BANK2) that is accessible from other neuroprocessor as second not own memory bank through external BUFFER.

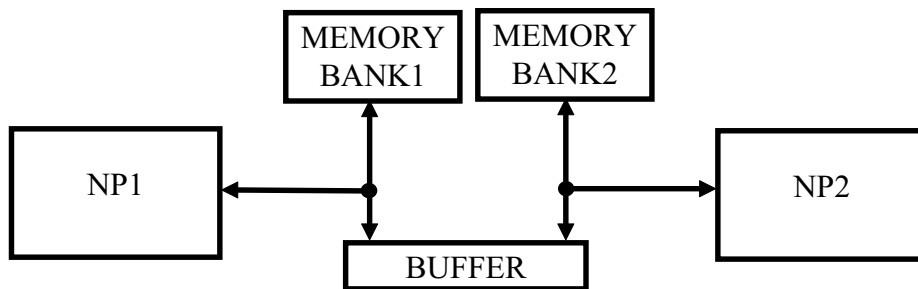


Figure 7. Bus Configuration Type 3 (Own Bank 0, not Own Bank 1)

2.5. I/O Communication Ports

Neuroprocessor has two identical high-speed communication ports: port 0 and port 1, each of which provides a point-to-point bi-directional communication interface to another neuroprocessor or some external peripherals. Fig. 8 represents an internal architecture of a single communication port.

Each port CP_x (x = 0,1) contains the following components:

CPI_x - communication port interface control unit which arbitrates between neuroprocessor and another device which has possession of the communication data bus at any given time;

MUX - multiplexor for selection of the address source for the communication port DMA request;

OCC_x - output channel counter that counts the number of 64-bit words during tranceiving via communication port;

OC_{Ax} - output channel address register that defines DMA address during tranceiving via communication port;

OCD_R - output channel data register;

ICC_x - input channel counter that counts the number of 64-bit words during receiving via communication port;
ICA_x - input channel address register that define DMA address during receiving via communication port;
ICDR_x - input channel data register.

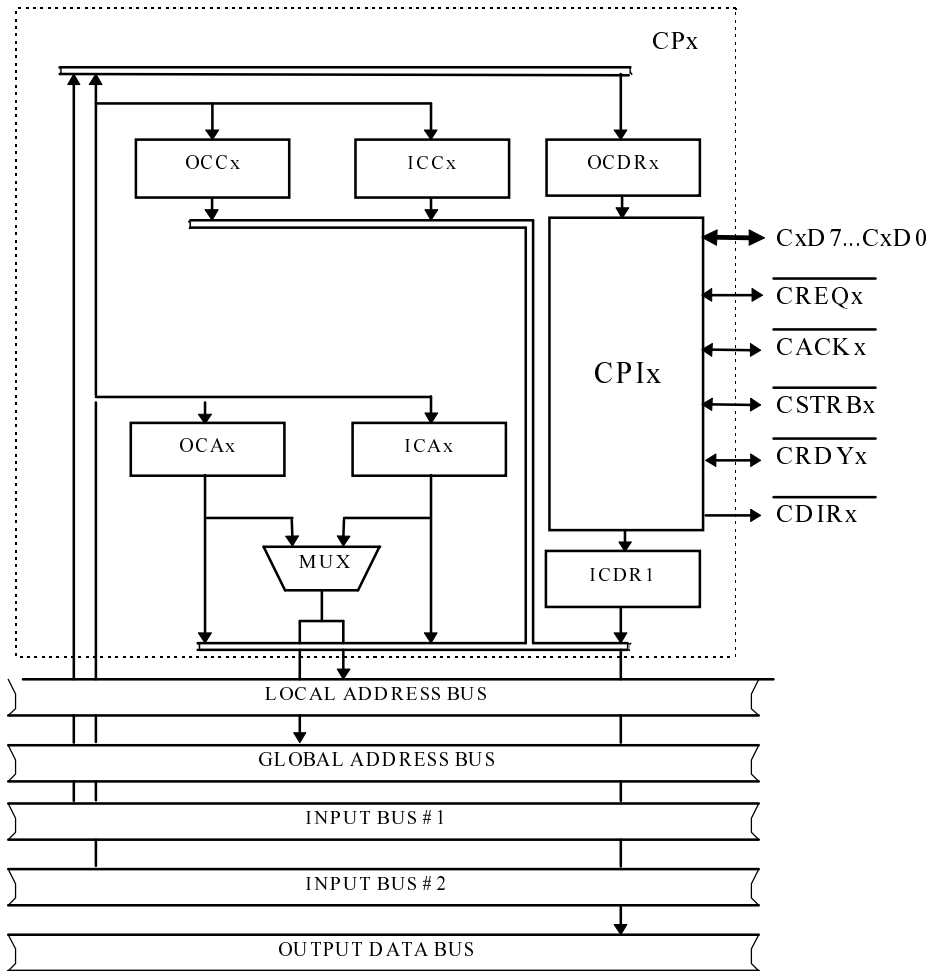


Figure 8. Communication Port CP_x Block Diagram (x = 0, 1)

The communication port interface consists of the following bi-directional data and control lines:

$\overline{\text{CREQ}}_x$ - communication port token request;

$\overline{\text{CACK}}_x$ - communication port token acknowledge to relinquish ownership of the communication port data bus upon receiving $\overline{\text{CREQ}}_x$ from another processor;

$\overline{\text{CSTRB}}_x$ - communication port strobe. A sending neuroprocessor activates this signal to indicate that it has placed a valid data on the communication port data bus;

$\overline{\text{CRDY}}_x$ - communication port ready. A receiving neuroprocessor activates this signal to indicate that it has received a data byte via the communication port data bus;

C_xD(7-0) - communication port data bus. This bus carries data bi-directionally between two neuroprocessors or between a neuroprocessor and some other device.

There are also output signals $\overline{\text{CDIRx}}$, that indicate transfer direction with high level when input and low level when output.

3. BRIEF INSTRUCTION SET DESCRIPTION

The neuroprocessor instruction set comprises two generic instruction types: 32-bit and 64-bit scalar data processing instructions (scalar instructions) and 32-bit vector data processing instructions (vector instructions).

The common format of scalar instructions is represented at Fig. 9. Each scalar instruction simultaneously defines two scalar operations:

- the 16 low significant bits of instruction from 0th to 15th specify the OPER field which defines execution one of shift or arithmetic/logic operation over general purpose registers GR0, ..., GR7;
- the bits from 16th to 30th specify the MOVE field that controls execution one of the data or constant move operation, address register modification operation or control instruction.

The neuroprocessor performs data move operations of: “register-register”, “memory-register“, “constant-register“ types. The neuroprocessor performs four control operation types: jump, call to subroutine, return from subroutine, return from interrupt.

The neuroprocessor instructions that use 32-bit immediate constant, address or address displacement are 64-bit wide. The immediate constant is placed into instruction from 32nd to 63rd bits.

Under non-stalls condition the scalar instructions are executed in pipeline with one instruction per clock cycle.

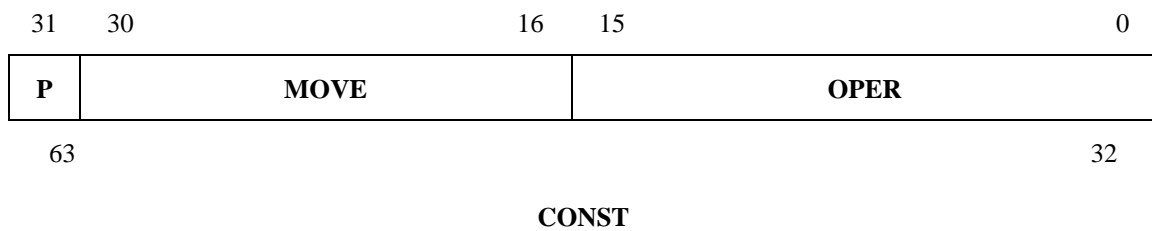


Figure 9. Scalar Instruction Code

The common format of vector instructions is represented at Fig. 10. Each vector instruction simultaneously defines up to four vector instructions:

- 0th bit of vector instruction specifies the reloading operation from WBUF to WOPER.
- the vector instruction bits from 1st to 12th specify the VOPER field which defines arithmetic or logic operation executing over vectors of packed data.
- the 18th bit of vector instruction specifies the weight coefficients reloading operation from WFIFO to WBUF.
- the vector instruction bits from 19th to 30th specify the VMOVE field which defines “read from external memory” or “write to external memory“ for the vectors of packed data.

The vector instruction bits from 13th to 17th specifies the COUNT field which defines the value of repeat counter for vector instruction (from 1 to 32 times). That mean that vector instruction is equivalent to parametrical cycle. The body of this cycle consist of only single instruction and repeat count value is equal to the value stored at COUNT field. Each iteration comprises single execution of all operations specified into instruction. The exeption is the operation of reload the weight coefficients from WBUF to WOPER. This operation always executed at the vector instruction execution last cycle. The second exeption is the operation of reload the weight coefficients from WFIFO and WBUF. This operation always is executed 32 clock cycles independently of COUNT field value.

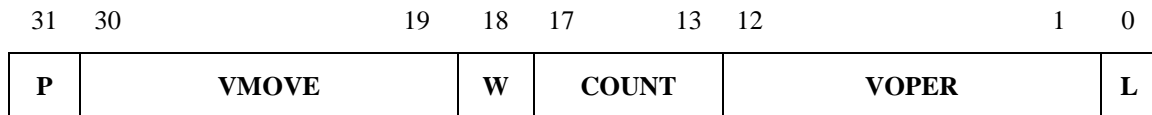


Figure 10. Vector Instruction Code

The 31th bit of both scalar or vector instruction allows to start this instruction execution in parallel when previous vector instruction not finish yet. It is possible when the neuroprocessor resources that is necessary for next vector instruction execution are not used by previous ones.

4. CONCLUSION

NeuroMatrix NM6403 is fixed point 64-bit DSP designed for neural nets. It provides a programmable operand width and offers scaleable performance from 50 MCPS (32-bit inputs and weights) up to 51.2 GCPS (1-bit inputs and weights) with 50 MHz clock frequency. The flexible operand and ability to scale performance let designers trade off precision and efficiency to suit their applications. NM6403 is produced using CMOS 0.5 μm, power supply voltage is 3.0-3.6V, power dissipation is less than 1 W at 50 MHz.

Due to its flexibility, the neuroprocessor can be used for neural nets accelerators, signal processing systems and for vector-matrix calculations. The communication ports and shared memory support allows ease building of multiprocessor systems with any configuration.

REFERENCES

1. Vixne P.E., Fomine D.V., Tchernikov V.M. "A VLSI Digital Neural Processor with Variable Word Length of Operands", *PRIBOROSTROENIE, Hard-Software systems for neural calculation support*, thematic issue, 1996, v.39, 17, pp.13-21.
2. TMS 320C4x User's Guide, August 1993, Texas Instruments Inc., USA.