CLAIMS

WHAT IS CLAIMED IS:

1. An information-processing apparatus comprising:

a vector processing unit having vector registers designed to store vector components; and

an in-memory compute unit, or IMC unit, having a crossbar array structure connected to the vector registers and adapted to store weights, the IMC unit preferably integrated in the vector processing unit,

wherein the information-processing apparatus is configured to perform:

vector transit operations to store input vector components of input vectors in the vector registers, wherein each of the input vectors is a vector of multiple elements corresponding to an array of number;

vector feed operations to feed said input vector components as input to the IMC unit, from the vector registers, for the IMC unit to perform matrix-vector multiplications, or MVMs, based on the weights and the input vector components, to obtain output vectors, and

vector readout operations to write output vector components of the output vectors in the vector registers, wherein each of the output vectors is a vector of multiple elements corresponding to an array of numbers.

2. The information-processing apparatus according to claim **1**, wherein the information-processing apparatus is further configured to perform:

weight transit operations to store the weights in the vector registers; and

weight feed operations to feed the weights to the IMC unit from the vector registers, so as to store the weights across the crossbar array structure,

wherein

the apparatus is preferably configured to perform the weight transit operations and the weight feed operations step wise, in accordance with a memory capacity of the vector registers and/or throughput capabilities of the vector registers.

3. The information-processing apparatus according to claim **1** or **2**, wherein the crossbar array structure includes

$N \times M$ cells comprising respective memory systems, each designed to store K weights, $K \geq 2$, whereby the crossbar array structure includes $N \times M$ memory systems adapted to store K sets of $N \times M$ weights, and

a selection circuit configured to enable $N \times M$ weights as active weights by selecting, for each of the memory systems, a weight from its K weights and setting the selected weight as an active weight.

4. The information-processing apparatus according to claim **3**, wherein the information-processing apparatus is further configured to

prefetch q sets of $N \times M$ weights, while the IMC unit performs MVMs based on the $N \times M$ weights that are currently active, so as to store the prefetched weights in the $N \times M$ memory systems, in place of q sets of $N \times M$ weights that are currently not enabled as active weights, where $1 \leq q \leq K - 1$.

5. The information-processing apparatus according to claim **4**, wherein the selection circuit includes

$N \times M$ multiplexers, each connected to each of the K memory elements of a respective one of the $N \times M$ memory systems, as well as

selection control lines, which are connected to each of the multiplexers, to allow any one of the K weights of each of the memory systems to be selected and set as an active weight, in operation.

6. The information-processing apparatus according to any one of claims **1** to **5**, wherein

the information-processing apparatus further comprises an accumulation circuit interfaced with the vector registers, and

the accumulation circuit is configured to accumulate output vector components of output vectors obtained from the IMC unit, in operation.

7. The information-processing apparatus according to any of claims **1** to **6**, wherein

the vector registers are designed to store a given number of numerical values, wherein said given number is larger than or equal to a smallest value of N and M.

8. The information-processing apparatus according to any of claims **1** to **7**, wherein

the information-processing apparatus further includes a controller, which preferably forms part of the IMC unit,

the vector processing unit includes a vector instruction decoder connected to said controller,

the vector instruction decoder is configured to decode vector instructions and transmit the decoded vector instructions to the controller, and

the controller is configured to orchestrate the vector feed operations, the MVMs, and the vector readout operations, based on the decoded vector instructions transmitted from the vector instruction decoder,

and wherein, preferably,

the information-processing apparatus is further configured to store the weights across the crossbar array structure,

said vector instructions form part of a vector instruction set designed in accordance with a base instruction set architecture augmented with two types of vector instructions, and

the two types of vector instructions are designed to respectively cause to store the weights and perform the MVMs.

9. The information-processing apparatus according to claim **8**, wherein the information-processing apparatus further comprises:

a main processing unit configured to execute non-vector instructions, wherein the main processing unit is interfaced with the vector processing unit to forward the vector instructions to the vector instruction decoder.

10. The information-processing apparatus according to claim **9**, wherein

the information-processing apparatus further comprises a memory interfaced with each of the main processing unit and the vector processing unit,

the main processing unit further comprises an instruction decoder configured to fetch instructions from the memory, decode the fetched instructions, and accordingly forward the vector instructions to the vector processing unit, and

the vector processing unit further comprises a vector load and store unit configured to perform the vector transit operations and the vector readout operations, by transferring corresponding values between the memory and the vector registers.

11. An information-processing system comprising one or more information-processing apparatuses according to any of claims **1** to **10**.

12. A method of operating an in-memory compute unit, or IMC unit, to perform matrix-vector multiplications, or MVMs, wherein the IMC unit has a crossbar array structure storing weights and is connected to vector registers of a vector processing unit, and wherein the method comprises:

performing a vector transit operation to store input vector components of an input vector in the vector registers, wherein the input vector is a vector of multiple elements corresponding to an array of numbers;

performing a vector feed operation to feed said input vector components as input to the IMC unit, from the vector registers,

operating the IMC unit to perform a matrix-vector multiplication, or MVM, based on the weights stored and the input vector components fed to the IMC unit, to obtain an output vector, and

performing a vector readout operation to write output vector components of the output vector in the vector registers, wherein the output vector is a vector of multiple elements corresponding to an array of number.

13. The method according to claim **12**, wherein the method further comprises, prior to performing the vector transit operation and the vector feed operation, performing:

weight transit operations to store the weights in the vector registers; and

weight feed operations to feed the weights to the IMC unit, so as to store the weights across the crossbar array structure.

14. The method according to claim **12** or **13**, wherein

the crossbar array structure includes $N \times M$ cells comprising respective memory systems, each designed to store K weights, $K \geq 2$, whereby the crossbar array structure includes $N \times M$ memory systems adapted to store K sets of $N \times M$ weights, and

the method further comprises, prior to performing said MVM, enabling said $N \times M$ weights as active weights by selecting, for each of the memory systems, a weight from its K weights and setting the selected weight as an active weight,

and wherein, preferably, the method further comprises, while the IMC unit performs MVMs based on the currently active weights, prefetching q sets of $N \times M$ weights to store the

prefetched weights in the $N \times M$ memory systems, in place of q sets of $N \times M$ weights that are currently not enabled as active weights, where $1 \leq q \leq K - 1$.

15. The method according to any one of claims **12** to **14**, wherein

the method further comprises, at the vector processing unit, decoding vector instructions and executing the decoded instructions to orchestrate the weight transit operations, the weight feed operations, the vector transit operations, the vector feed operations, and the vector readout operations,

the vector instructions preferably form part of a vector instruction set designed in accordance with a basis instruction set architecture augmented with two types of vector instructions, the latter designed to cause to respectively cause to perform said weight feed operations and said MVMs, and,

preferably, the method further comprises, prior to decoding the vector instructions at the vector processing unit, sending the vector instructions from a main processing unit interfaced with the vector processing unit.

16. The method according to any one of claims **12** to **15**, wherein the method further comprises:

fetching the input vector components from a memory interfaced with the vector processing unit, prior to performing the vector transit operation to store the input vector components of this input vector in the vector registers; and

transferring the output vector components from the vector registers to the memory, after performing said vector readout operation.