MultiClet is a technical term used to define processor core, processor with multicellular architecture, and multicellular processor based device. For this reason the term was chosen to be the company's name.

Main advantages of multicellular architecture

- **It enhances performance and at the same time reduces energy consumption**
- **There is no need for compliated controlling units such as brunch prediction, out-of-sequence command execution etc., therefore it reduces considerably the size of the chip.**

<u>**Liveness**</u>

- **Fault Tolerance (continuous program running in case of hardware environment break-down - cells failure)**
- **Dynamic reconfiguration (in the process of task performance 1,2,3 or 4 cells can be involved, while uninvolved cells can perform other tasks)**

<u>**Super security of information**</u>

- **Natural virus immunity**
- **Cryptographic capabilities**
- **Anti-hacking features**

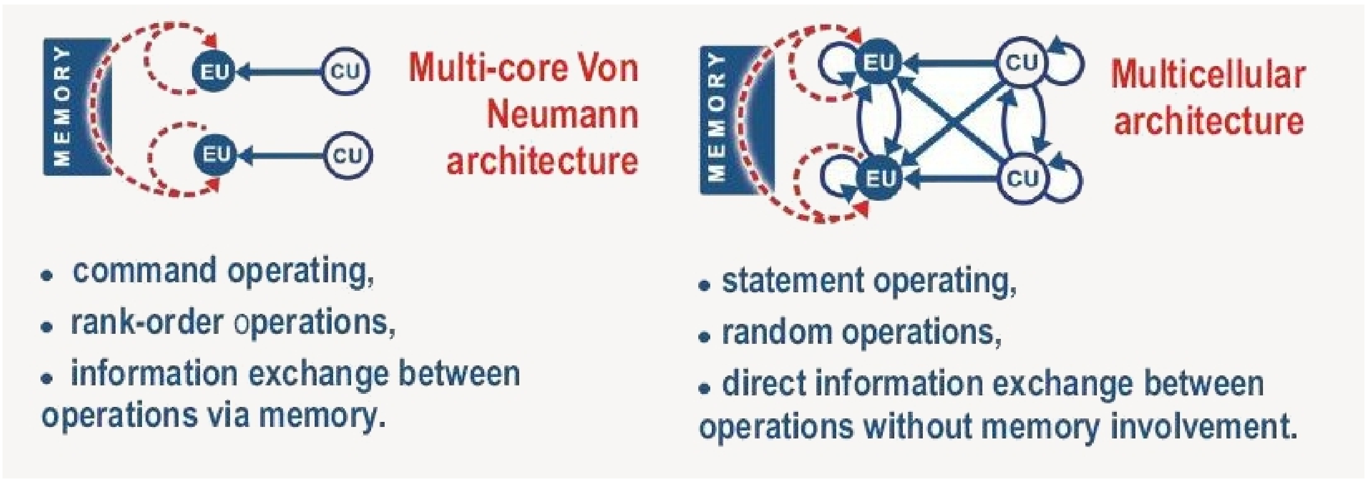
Multicellular architecture features

- **"Natural" implementation of parallelism (without solving the problem of parallelizing)**
- **Program execution without recompilation for any quantity of cells**
- **Effective settlement of any class of problems (commutation environment doesn't limit intercellular data exchange)**

- **Significant reduction of memory accesses during execution of task**

The multicellular architecture differs from the von Neumann model by the direct indication of informational connections between operations and consequently any requirement for the ordered arrangement of the operation description in the program is annulled.

It is different from the well known non-von Neumann architectures by means of sequential fetching which realizes imperative programming languages as well as by dynamically generated tags, but not instruction addresses of indicating informational connections. Any instruction is executed at the “data readiness” and “its output users’ readiness”.


MEMORY
EU
CU
EU
CU
Multi-core Von Neumann architecture
MEMORY
EU
CU
EU
CU
Multicellular architecture
• command operating,
• rank-order operations,
• information exchange between operations via memory.
• statement operating,
• random operations,
• direct information exchange between operations without memory involvement.

The cell instruction set is based upon some intermediate presentation of a compiled program after the syntax analysis (triads) and actually it is a sort of hardware realization of input programming language. It minimizes the labour costs to create compilers due to the fact that the units of machine-oriented optimization and paralleling disappear as well as the instruction generating unit dramatically decreases. The notion "assembler programming" disappears as the processor language is not visible and thus it is “not programmable”. The software becomes really hardware-independent.

Comparison with analogs:
Download the file in pdf format

For more information, please see:

General technical information