Two-Level Controlled Parallel Reconfigurable Architecture

Takanobu Baba and Kanemitsu Ootsu,
Utsunomiya University, Japan

Outline

- Review of microprogramming and two-level microprogramming scheme
- Introduction of two-level microprogrammed multiprocessor architecture
- Proposal of two-level control structure for fine-grained reconfiguration
- Conclusion

Our motivation

- To use old wineskins

  - *microprogramming* -

  to store new wine

  - *reconfigurable parallel architecture* -

Review of microprogramming and two-level microprogramming scheme

Microprogramming revisited

- Invented by M.V. Wilkes in 1951
  - as a means for systematically designing computer's control part
- Utilization of the technology
  - bridging the gap between an instruction set definition and the underlying hardware
  - the use of semiconductor memory as control storage lead to *dynamic microprogramming*
- Types of microinstruction set
  - *vertical type* vs *horizontal type*
  - *two-level microprogramming* as the combination

Features of micro-instruction set level architecture

- Basically, one-machine cycle execution
- Decoded signals -*micro-orders*- directly control RTL operations -*micro-operations* –
- Single micro-instruction controls not only usual ALU & memory operations but also sequence of micro-instructions

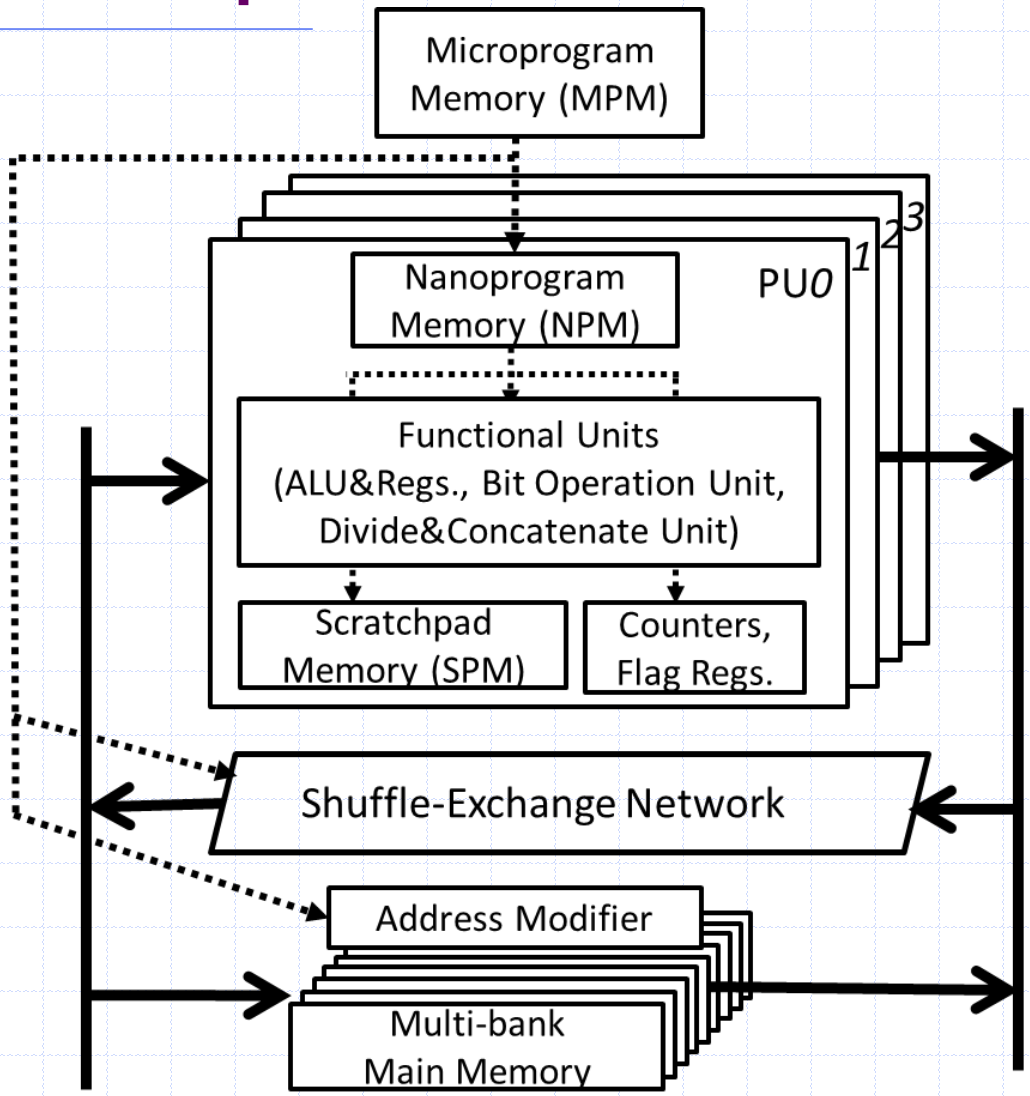
Inheritance of the technologies

- RISC computers
  - policy of simple operations
- VLIW architecture
  - long microinstruction of horizontal type
- Compiler optimizations
  - local and global microprogram optimization technologies

To me, they seem to raise micro-instruction set level technologies to machine instruction set one

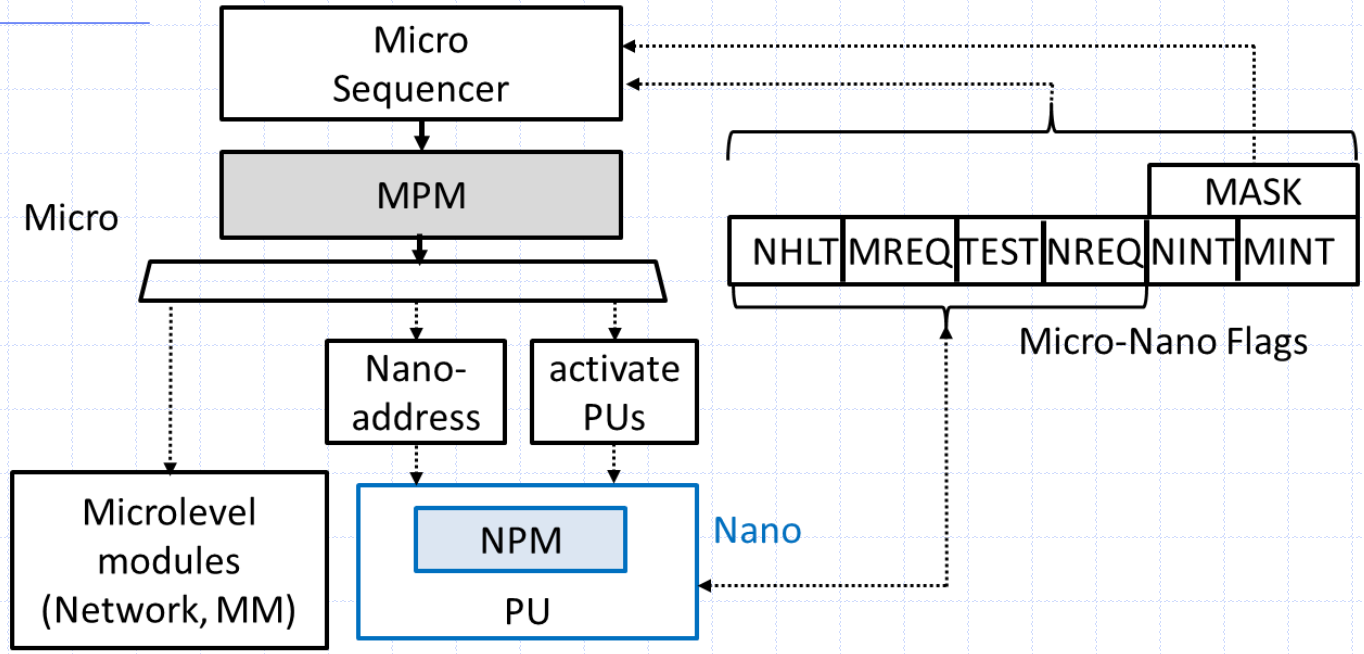
Introduction of two-level microprogrammed multiprocessor architecture -MUNAP-

Two-level microprogrammed multiprocessor architecture –MUNAP-



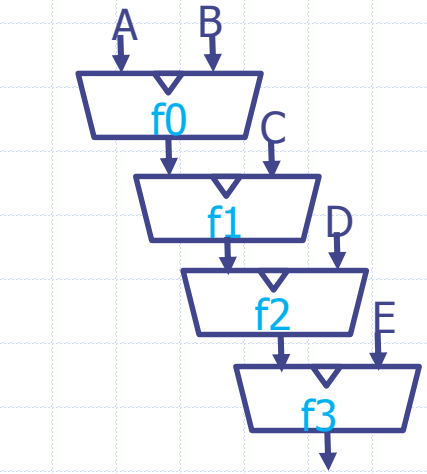
- Adoption of two-level control structure coupled with the multiprocessor parallelism
- MUlti-NAnoProgram machine

Micro-nano interaction mechanism

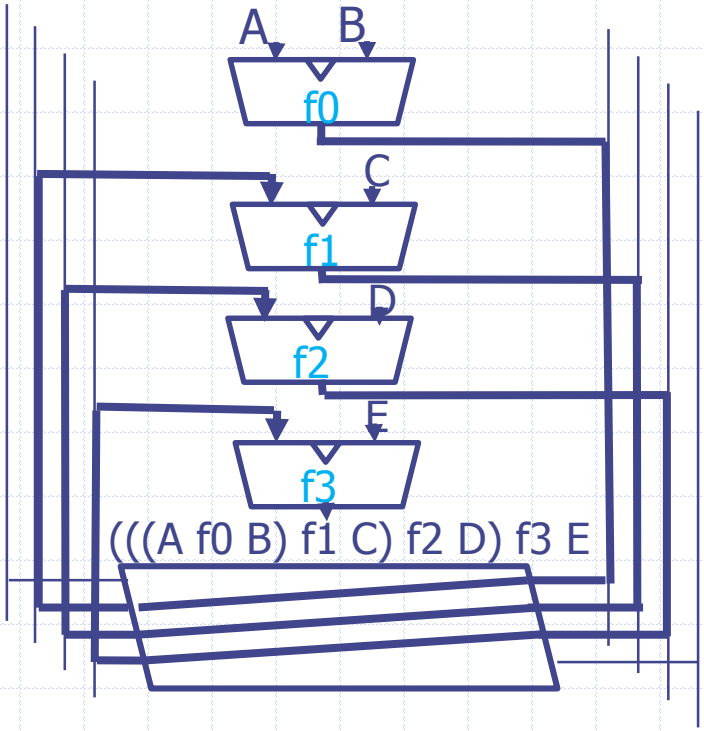


(i) activation of nanos by micro, and notifying the termination of nano to micro;
(ii) transferring the test results from nano to micro; and
(iii) requesting from one-level to another

Micro-level reconfiguration



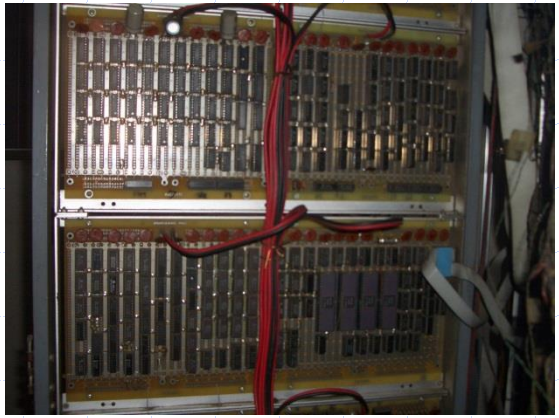
(a) Logical view of four-stage 16-bit cascading

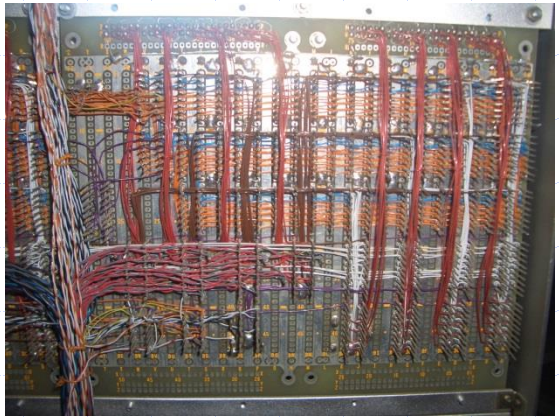


(b) Physical data flow for (a)

- f0 through f3 indicate functions of PU0 through PU3, respectively
- other configurations are possible by using the same HW

MUNAP prototype (1978~)





Proposal of two-level control structure for fine-grained reconfiguration

FPGA accelerators and dynamically microprogrammable machines

- Reconfigurable fabric for heavy workloads
  - e.g., Catapult‥Bing's ranking engine by 8 FPGAs
- Standing position
  - between the HW implementation (ASIC) and the SW implementation
  - ASIC - realizes high performance execution, but lacks the reconfigurability and fast development round
  - SW - hard to meet heavy workloads
- Both *bridge the gap* between these two extreme implementations by utilizing *RTL parallelism*

Major challenge of FPGA development

- Requirement for extensive hand coding using RTL language and manual tuning of the designed configurations
- Need to repeat the processes of description by HDL, generation of the configuration data, and downloading it to FPGA

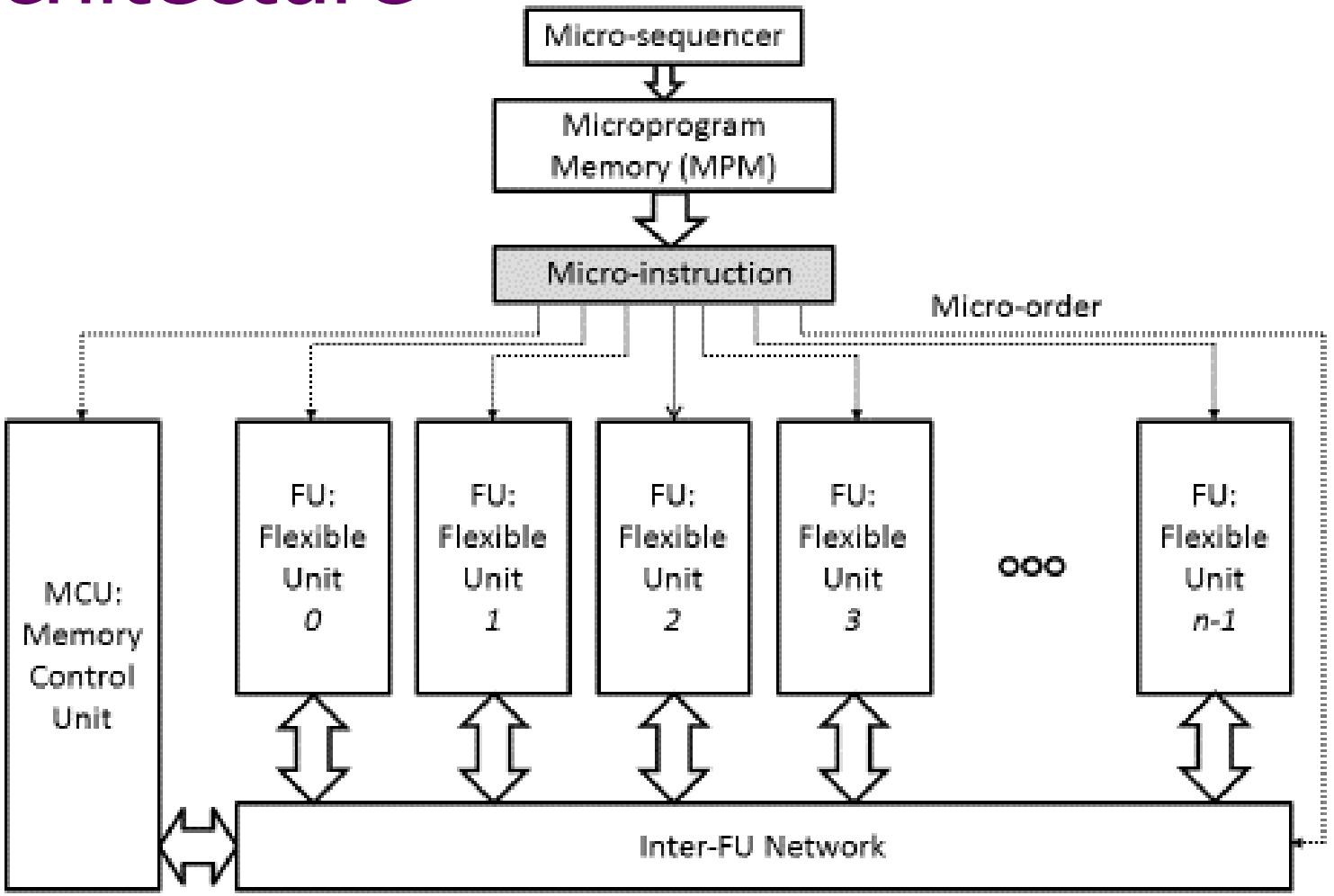
Our solution is ...

to isolate the programming view from the detailed implementation view

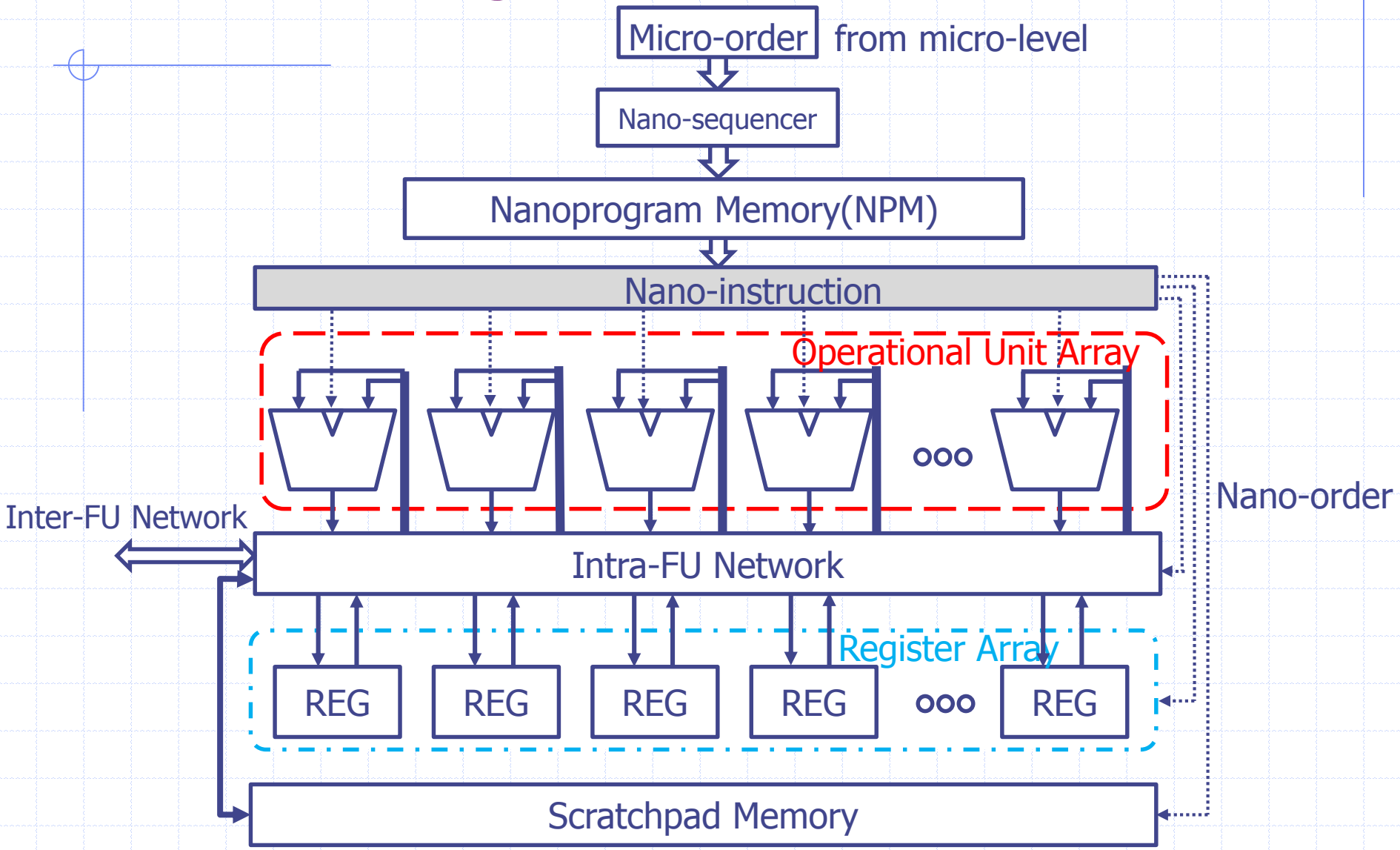
- A new parallel reconfigurable architecture that utilizes the two levels of control
- The *single stream microprogram* specifies multi-nanoprograms and manages their execution
- The *multi-nanoprograms* control a lot of fine-grained operational units in parallel

⇒ Our scheme will ease the development of applications without sacrificing the parallelism !

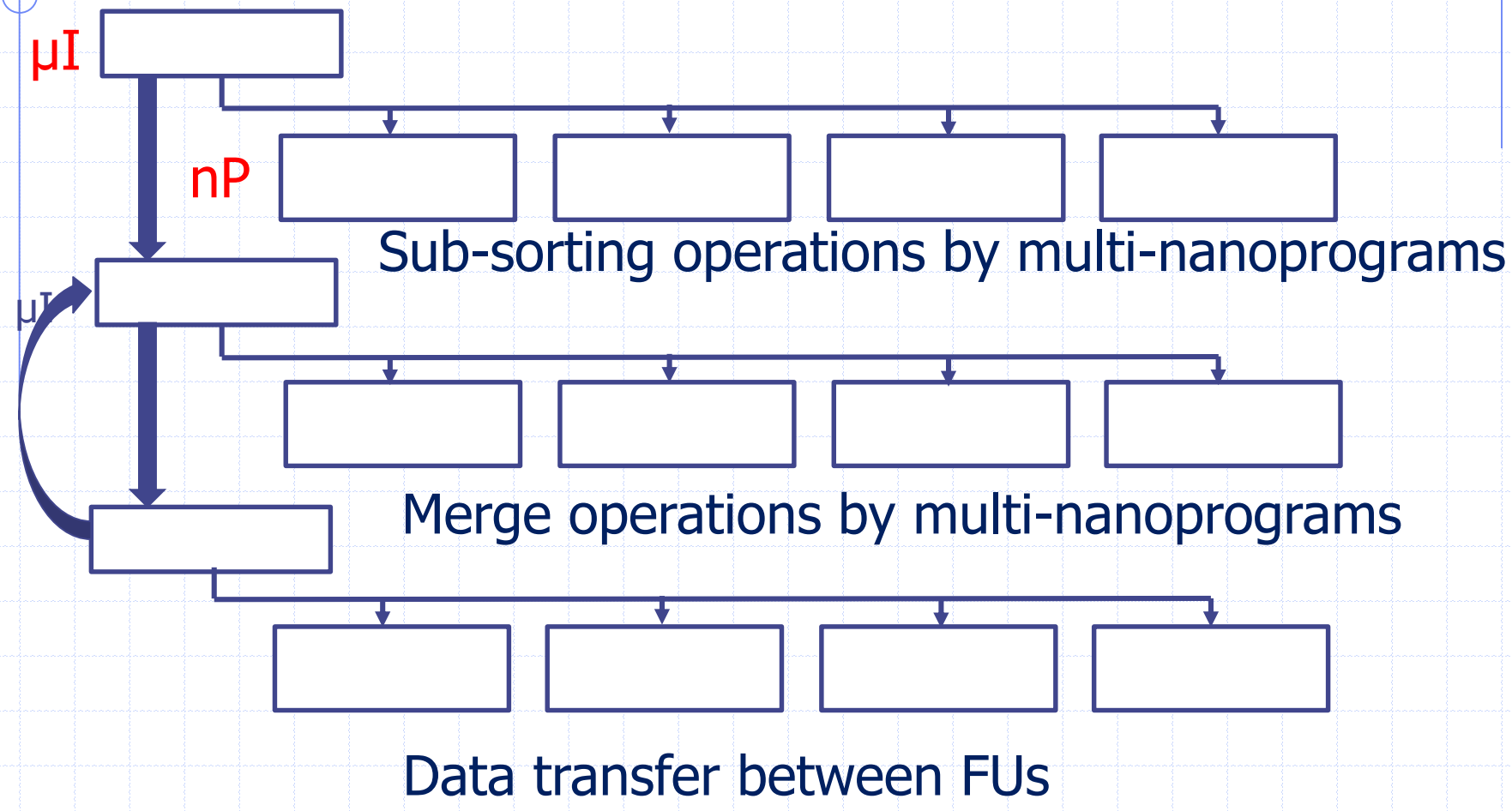
Top view of parallel reconfigurable architecture



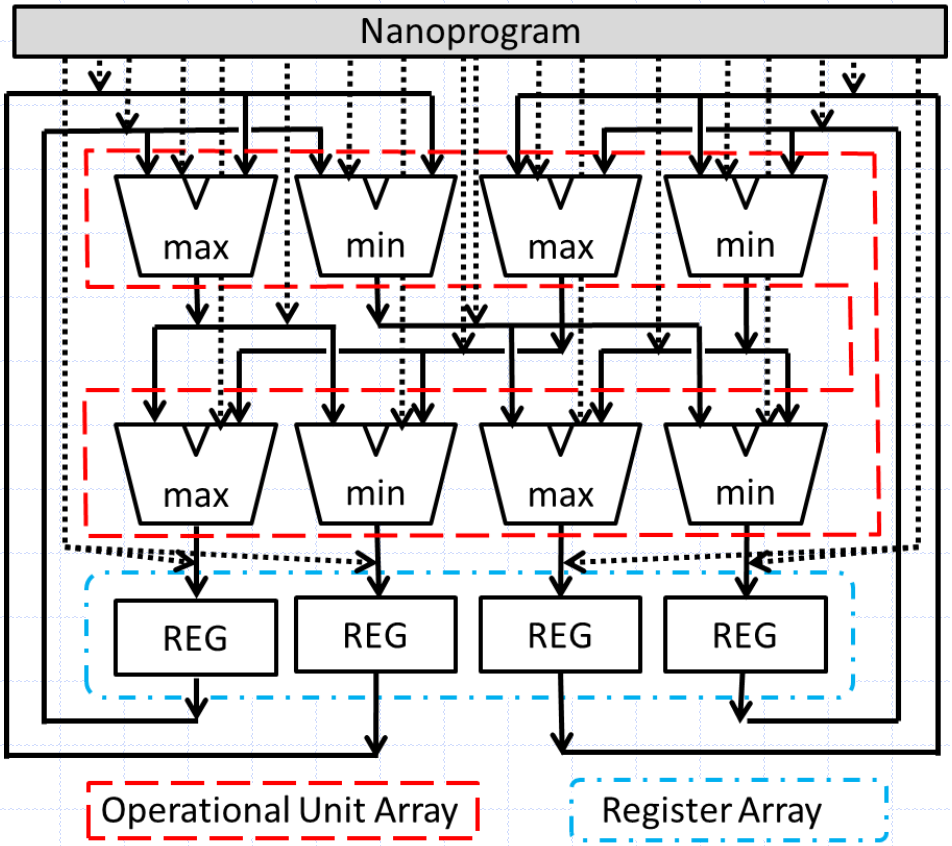
Internal organization of FUs



Our image of using the two-level microprogram – example sorting



Nanoprogram controlled sub-sorting operation



Advantages

- Our scheme isolates the programming view from the detailed implementation view and allows the flexibility of the underlying HW
- Microprogram description and its translation will be much better than the HDL description and the logic compilation
- Our scheme replaces many complex sequence control circuits of FPGA design with nanoprograms of horizontal type

Conclusion

- Based on our experience, we proposed a new parallel reconfigurable architecture that utilizes the two levels of control
- Following the original motivation of Wilkes, our scheme isolates the programming view from the detailed implementation view and allows the flexibility of the underlying HW
- Thus, it will be much better for the application developers than the use of an HDL and logic compiler scheme.

Thank you!