Constructing A General Purpose Cellular Computer

An Explorative Approach to Nanocomputers

Markus Koskimies
Master's Thesis presentation
30.4.2010
Supervisors: Dr. Peter Antoniac, Mr. Antti Alasalmi
Reviewers: Prof. Olli Martikainen, Prof. Valentin Cristea

Contents

- Backgrounds
- Research objectives
- Research method
- Results
- Conclusions
- Further research

Backgrounds

- Find out, how future computers are programmed
- Generally, increasing number of logical switches increases the processing power of the computer → nanotechnology
- There are evidence from previous research that nanocomputers are made cellular
  - Simple processing cells
  - Connected to nearest neighbors

Research objectives

- Analyze the properties of cellular computer
- Make experiments with a cellular computer
- Analyze, how software could be build and organized with cellular computers
- Research questions
  - What kind of physical structure a cellular computer has?
  - How cellular computer is built as general purpose computer?
  - How to implement and organize operating system and software to cellular computer?

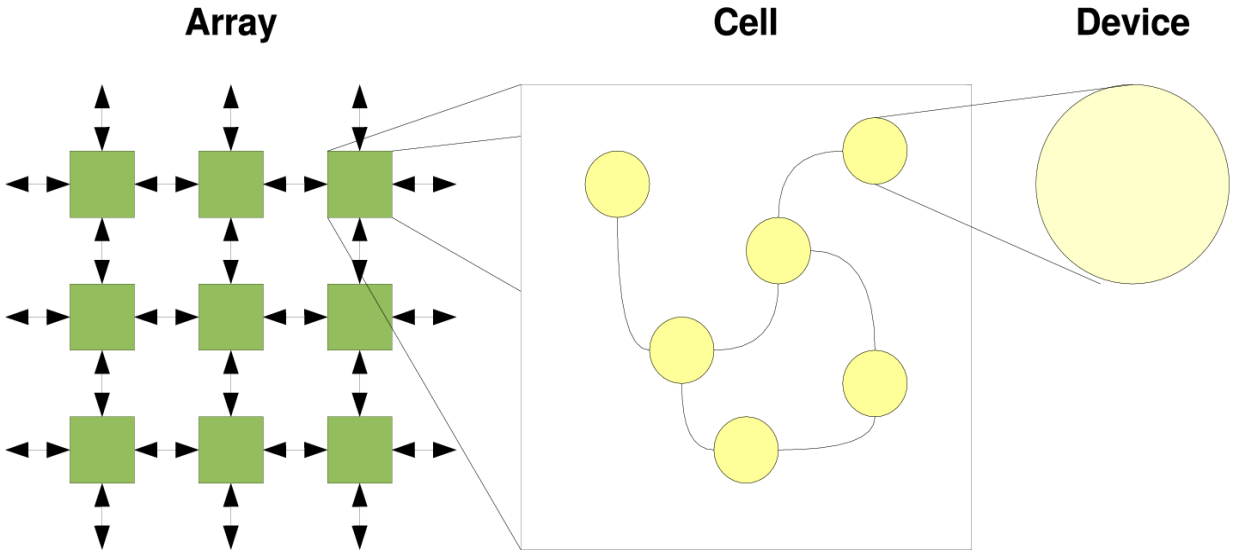
Research method

- Constructive research
  - A virtual cellular computer was built
  - Evaluated with selected experiments
- Eplorative nature
  - At the beginning, all possibilities were considered
  - By analyzing, the ones with most support with current research were selected

General Purpose Computer

- A general purpose computer has two important theoretical properties:
  - It is Turing complete
  - It is self-reconfigurable

Cellular Computer



Physical realization

- 3D technology
  - Massive amounts of devices
  - The size is not practical otherwise
- Energy consumption, heat generation
  - A big problem with massive amounts of switching devices
- Defects and faults
  - Even with nearly perfect manufacturing, large numbers of defected cells

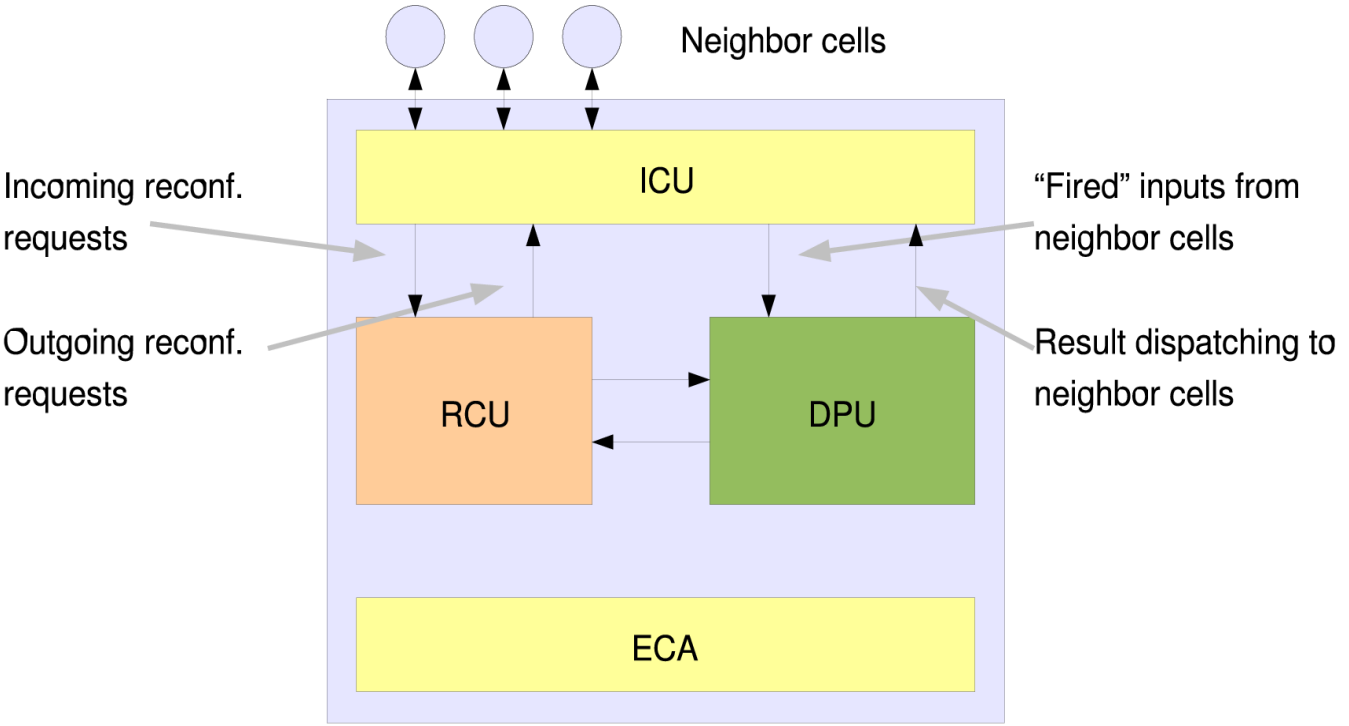
Physical realization

- To fight against the problems:
  - Asynchronicity; tolerates variations in timings, due to overheating or lack of energy
  - Self-reconfigurability; for creating configurations on the fly, not using faulty cells

Logical Cell Model

- Was created for experimenting cellular computer
- String based processing:
  - Streams of NIL terminated tokens (in this case, 5 bits)
  - Cells are state machines
  - For being able to do configurations manually

Logical Cell Model



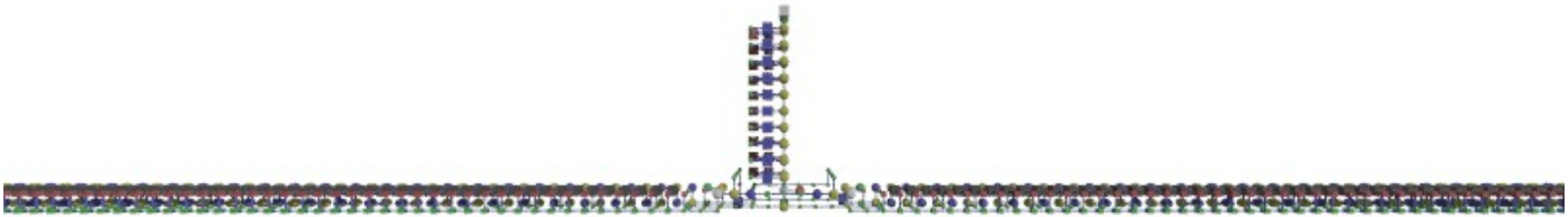
Experiments

- Three problems selected
  - A Turing machine emulation
  - A self-replicating machine
  - An algorithm to solve Eight Queens' Problem

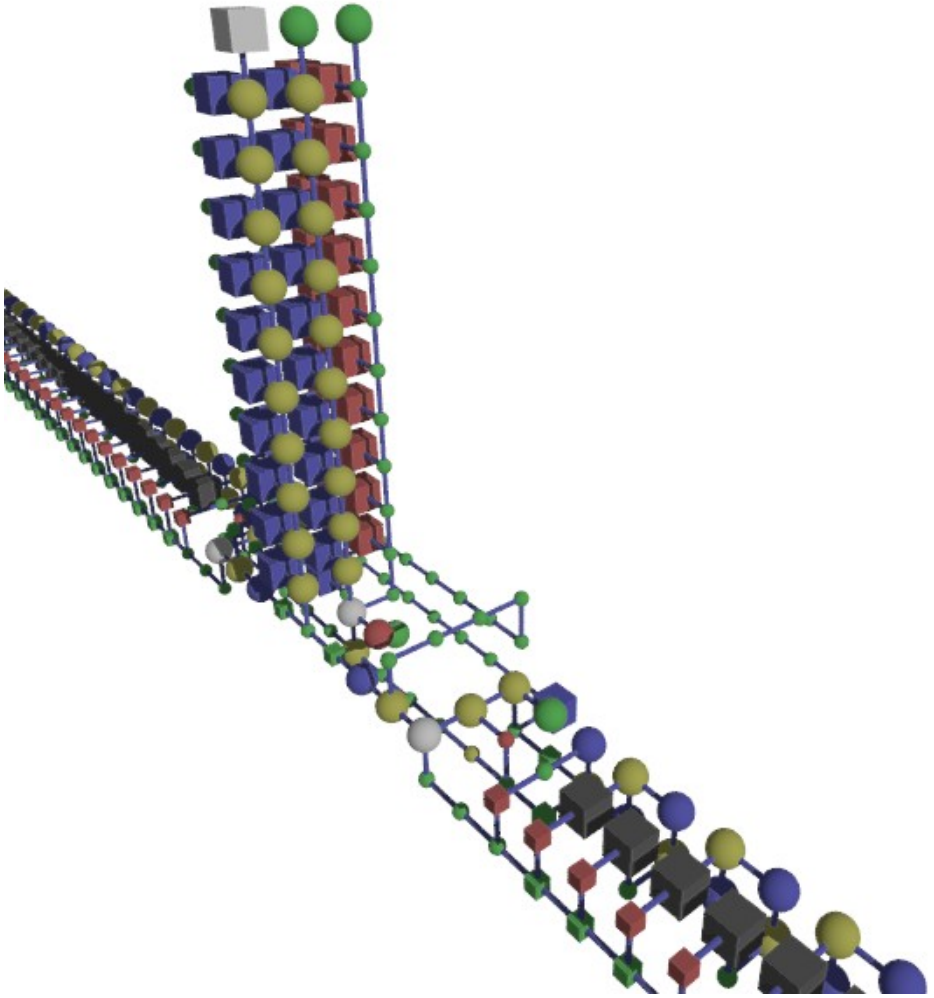
Turing Machine Emulation

- Busy Beaver 501 implemented as an example
- Two-stack approach
  - Fits well for cellular automaton

Turing machine emulation



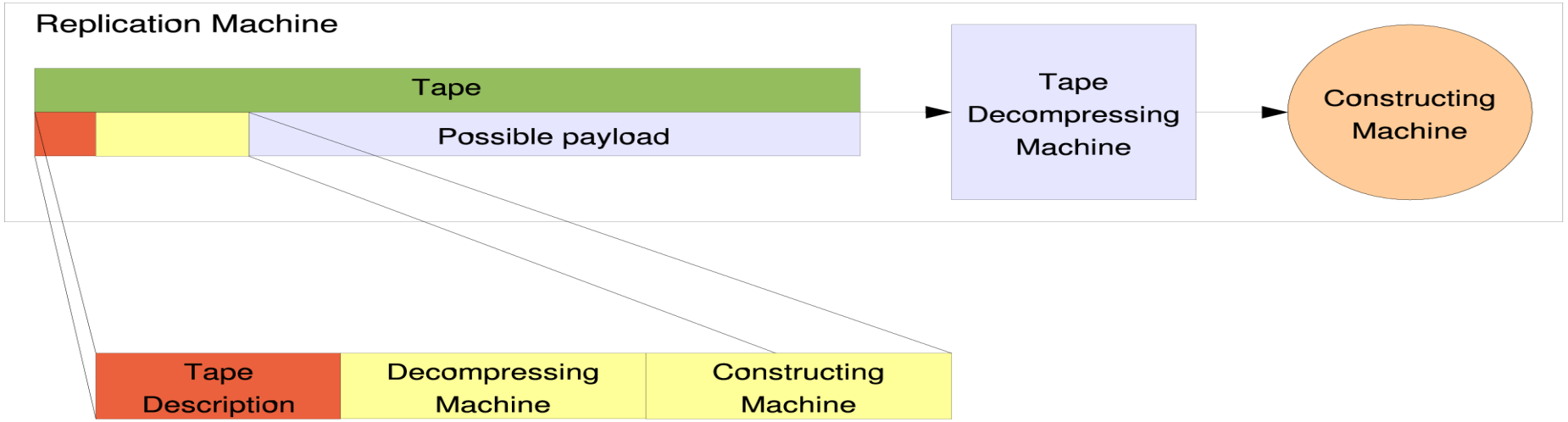
Turing machine emulation



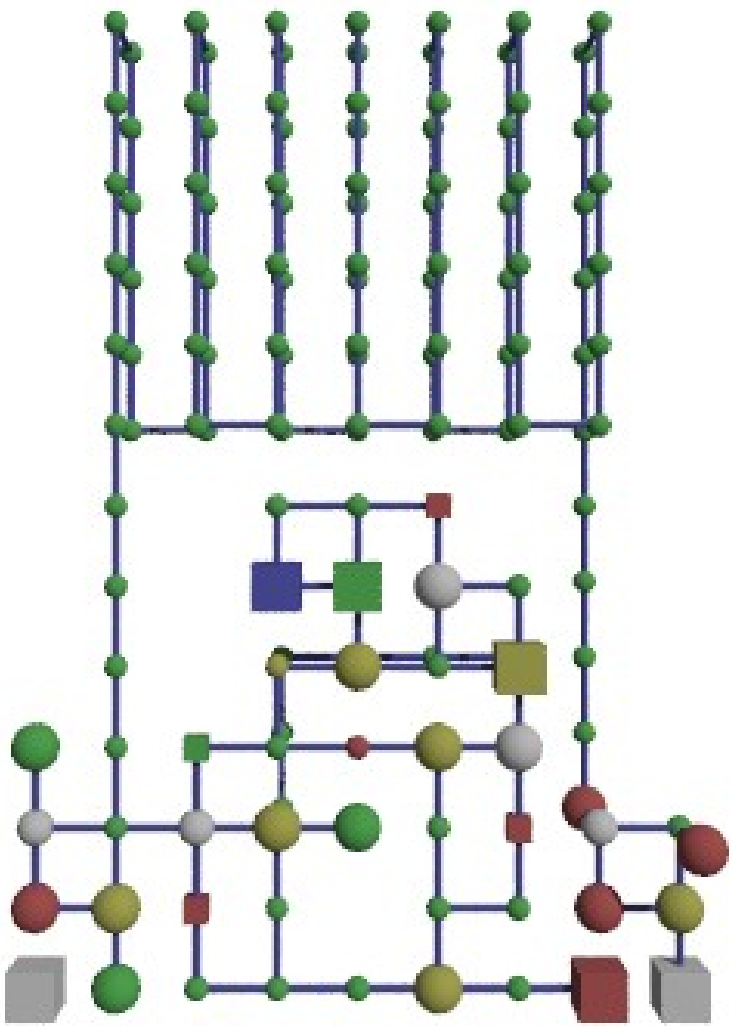
Self replicating loop

- Originally, a von Neumann processor was considered
  - Sequential processor
  - Both data and code in same memory
- Later, a simpler solution was found and implemented

Self replicating loop



Self replicating loop

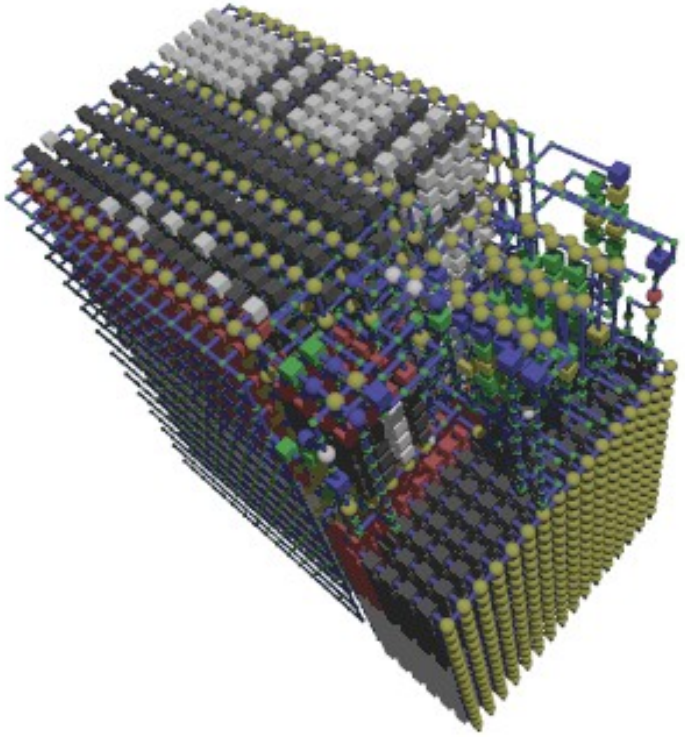


- Implemented for experimenting self-reconfigurability

Eight Queen's Problem

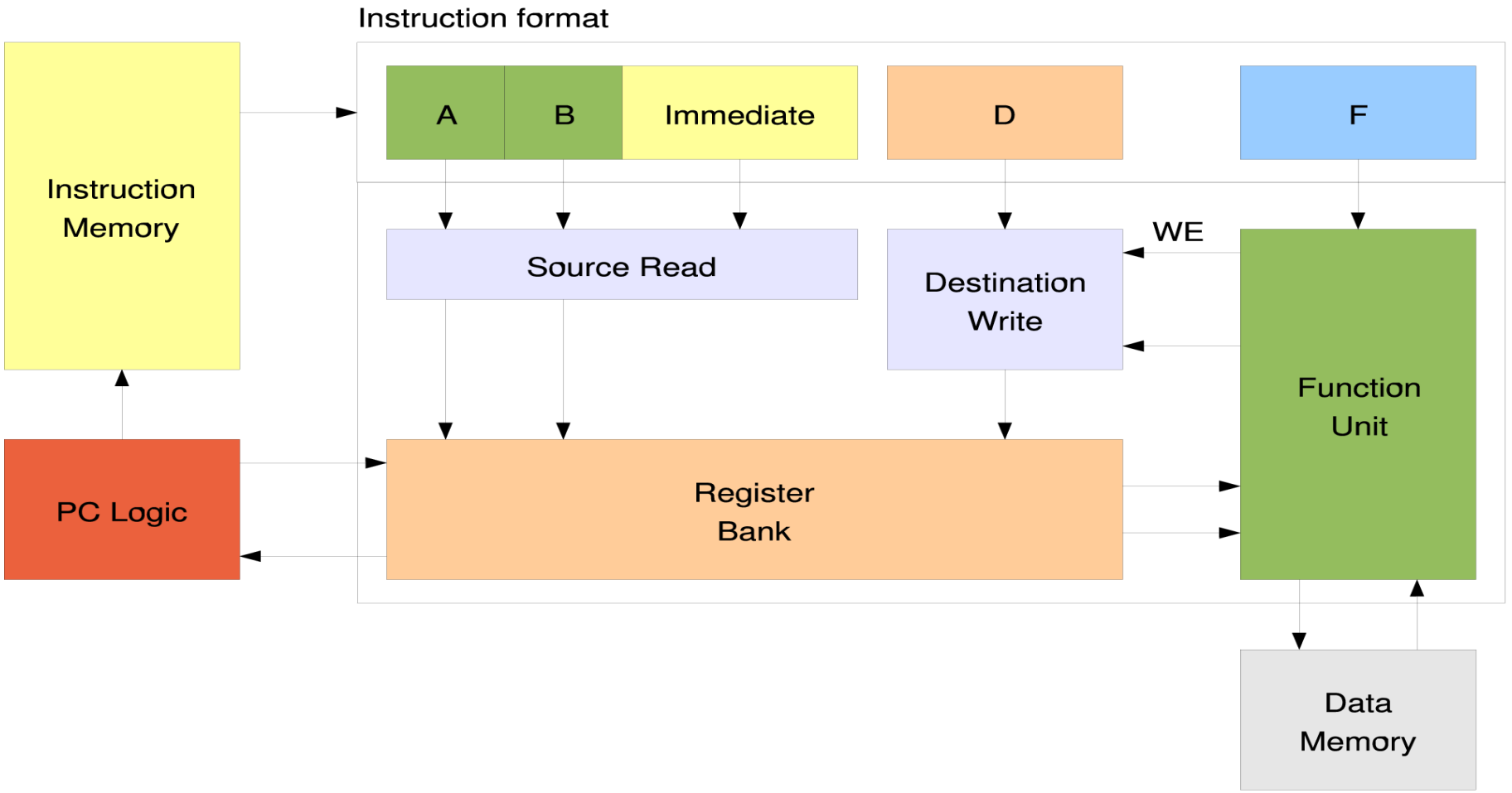
- Implemented for experimenting computational problem solving
- Two approaches
  - With (general purpose) sequential processor emulation
  - With dedicated configuration

MLI, Machine Language Interp.

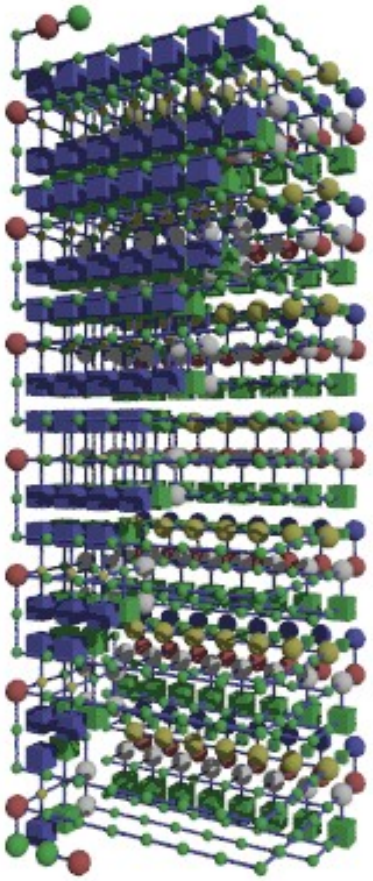


- Emulation of sequential processor
- Executing recursive algorithm to solve the problem

MLI Architecture



Dedicated solution



- Dedicated, parallel solver
- DEQPS (Dedicated Eight Queens' Problem Solver)

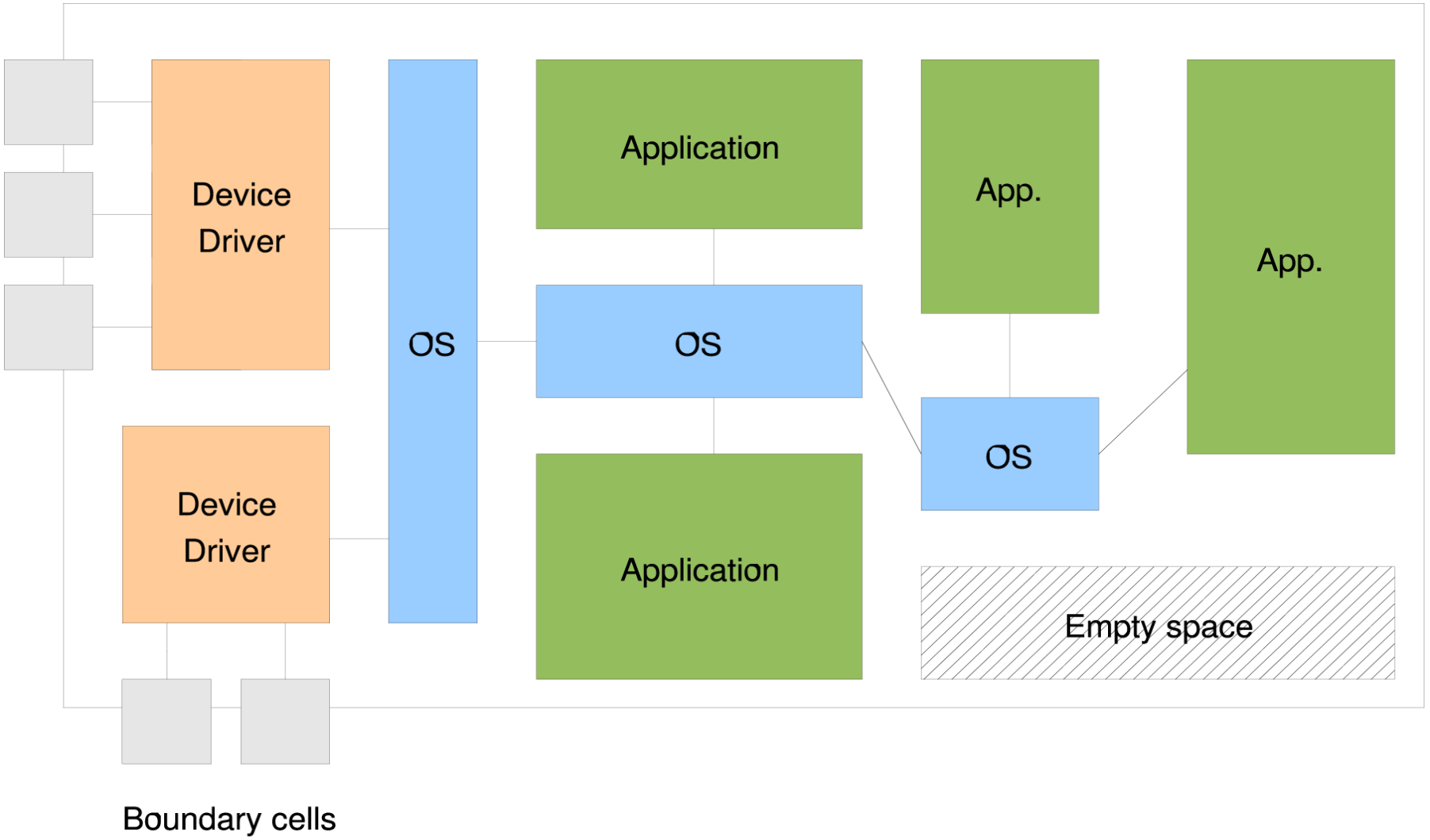
Software and Operating System

- As backgrounds, research on reconfigurable platforms was analyzed
- Combined with the results of the experiments

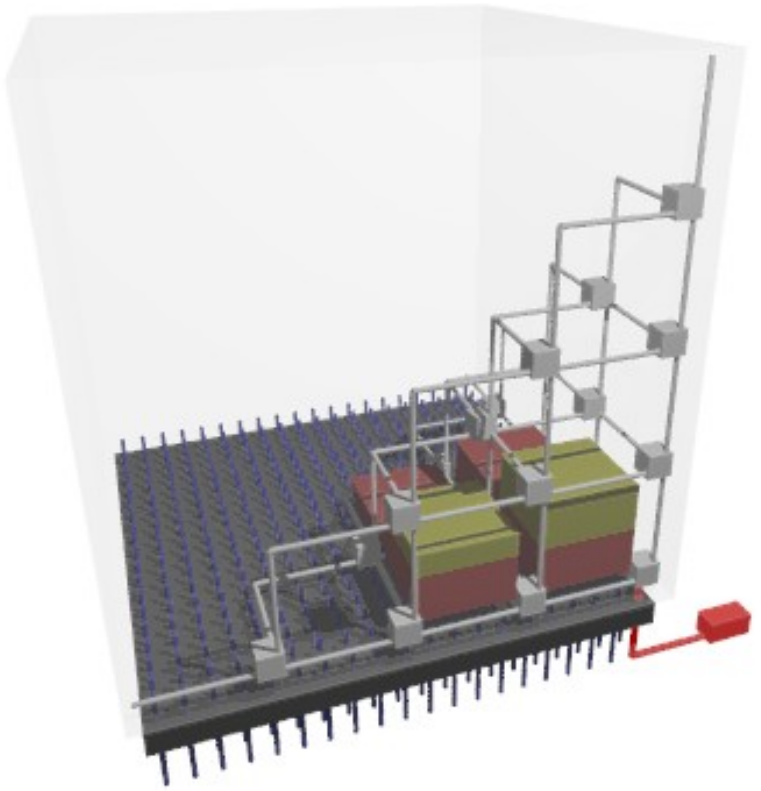
Software development

- Existing languages, tools and methodologies can be used
- Selection criteria is human productiveness
  - Compiler technology hides the low-level architecture
- Similar toolchain as with reconfigurable computers

Operating System



Operating System (3D)



Conclusions

- General purpose cellular computers are:
  - Turing Complete
  - Asynchronous
  - Self-reconfigurable
- The software development is similar to existing computers (with future advances)
- The operating system is logically the same, but internally very different from existing computers

Further research

- String based computation
- Larger constructions
- Realtime vs. asynchronicity
- Distributed computing
- Dataflow computing

Thank you!