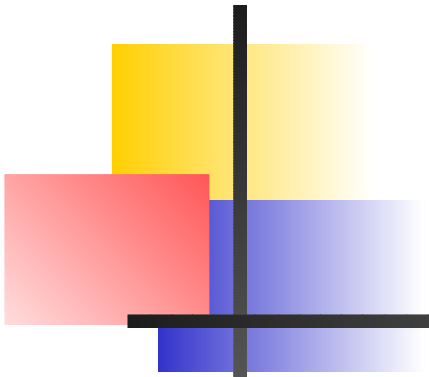


A Self-Optimising Simulator For A Coarse-Grained Reconfigurable Array

Jack Whitham and Neil Audsley

jack@cs.york.ac.uk

Real Time Systems Group

Department of Computer Science

University of York

Overview

- MCGREP project summary
- Simulator wanted!
- Other simulators
- Design & Implementation
- Evaluation
- Applications
- Conclusion

RTS York

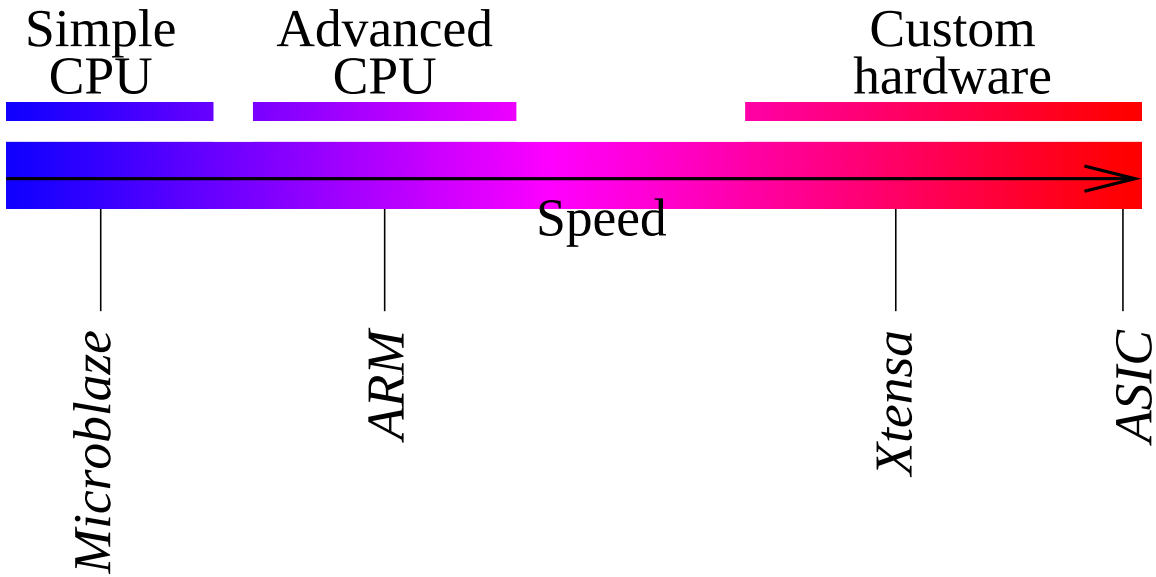
Real-Time Embedded Systems

Special challenges for Real-time Embedded (RTE) systems:

1. High Performance.
2. Easy WCET Analysis.
3. General Purpose.

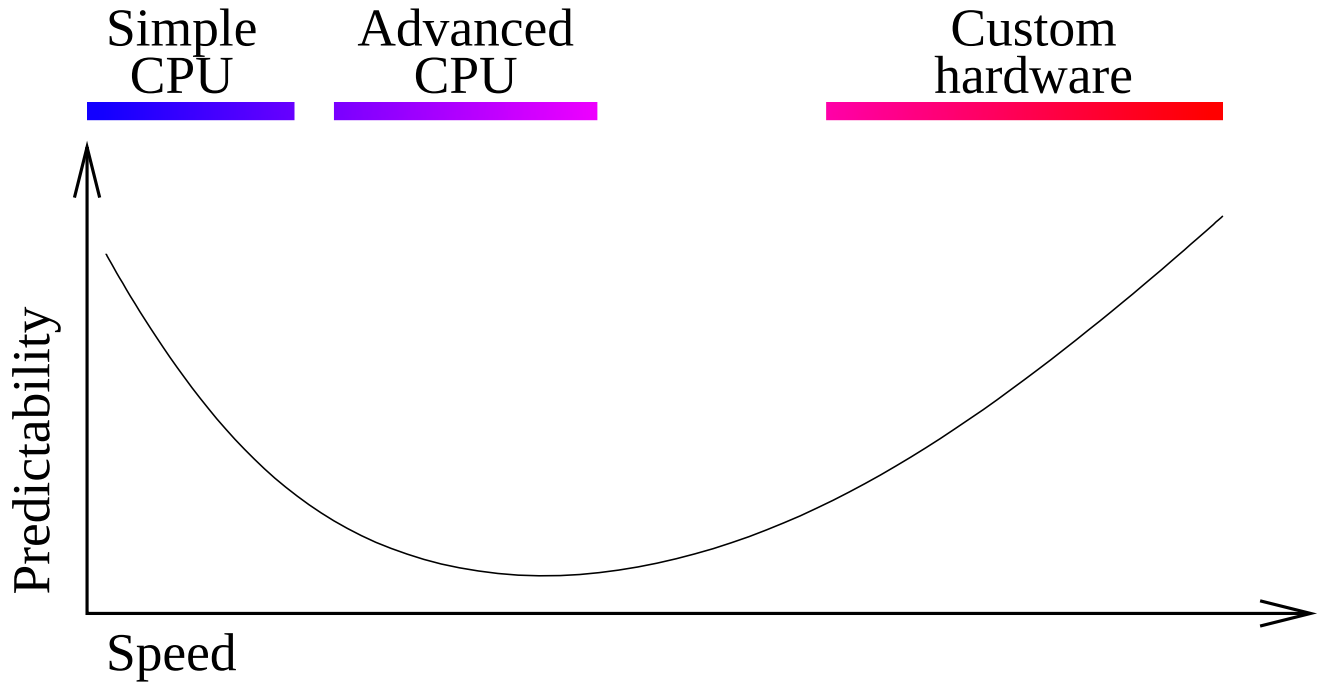
RTE Requirements

(1): High Performance



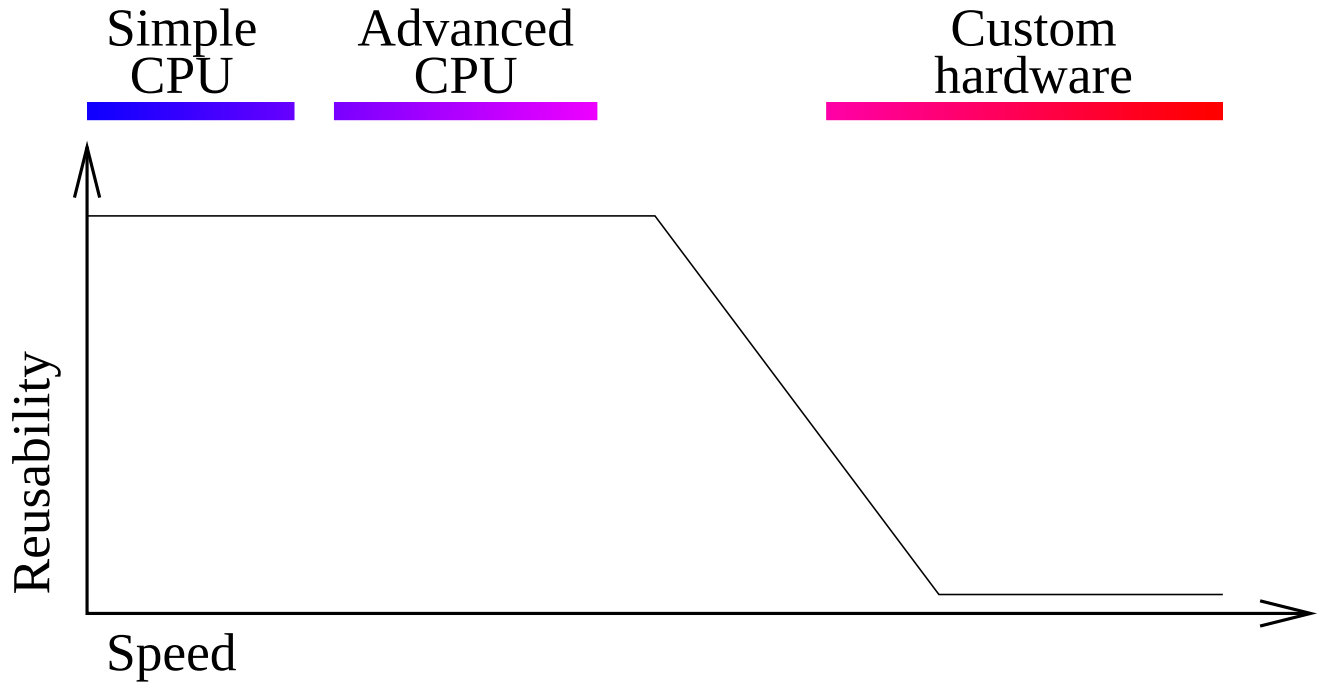
RTE Requirements

(2): Easy WCET Analysis



RTE Requirements

(3): General Purpose



RTS York

General Purpose?

But embedded systems are fixed purpose!

- Or are they?

General Purpose?

But embedded systems are fixed purpose!

- Or are they?

General purpose features also useful for:

- Bug fixes and extensions,
- Reuse of old designs,
- Run-time adaptation.

MCGREP

Microcoded **c**oarse-**g**rained **re**configurable **p**rocessor: software controlled CPU

MCGREP

Microcoded **c**oarse-**g**rained **re**configurable **p**rocessor: software controlled CPU

Aims of MCGREP:

1. **Speed** and **predictability**, approaching custom hardware.

RTS York

MCGREP

Microcoded **c**oarse-**g**rained **re**configurable **p**rocessor: software controlled CPU

Aims of MCGREP:

1. **Speed** and **predictability**, approaching custom hardware.
2. General purpose **reusability**, like a simple CPU.

RTS York

Architecture

- Both **Coarse-grained Array** and **CPU**.

Architecture

- Both **Coarse-grained Array** and **CPU**.
- Many reprogrammable processors.

Architecture

- Both **Coarse-grained Array** and **CPU**.
- Many reprogrammable processors.
- Each can run programs from external memory and from **internal microcode**.

RTS York

Architecture

- Both **Coarse-grained Array** and **CPU**.
- Many reprogrammable processors.
- Each can run programs from external memory and from **internal microcode**.
- Microcode is used for **hotspot** execution.

MCGREP Paradigm

- Applications execute from external memory, **except** when hotspots are reached.

MCGREP Paradigm

- Applications execute from external memory, **except** when hotspots are reached.
- Software **binary translation** is used to retarget **hotspots** for the MCGREP array.

MCGREP Paradigm

- Applications execute from external memory, **except** when hotspots are reached.
- Software **binary translation** is used to retarget **hotspots** for the MCGREP array.
- Hotspot tasks are distributed to all or part of the reconfigurable array.

RTS York

MCGREP Paradigm

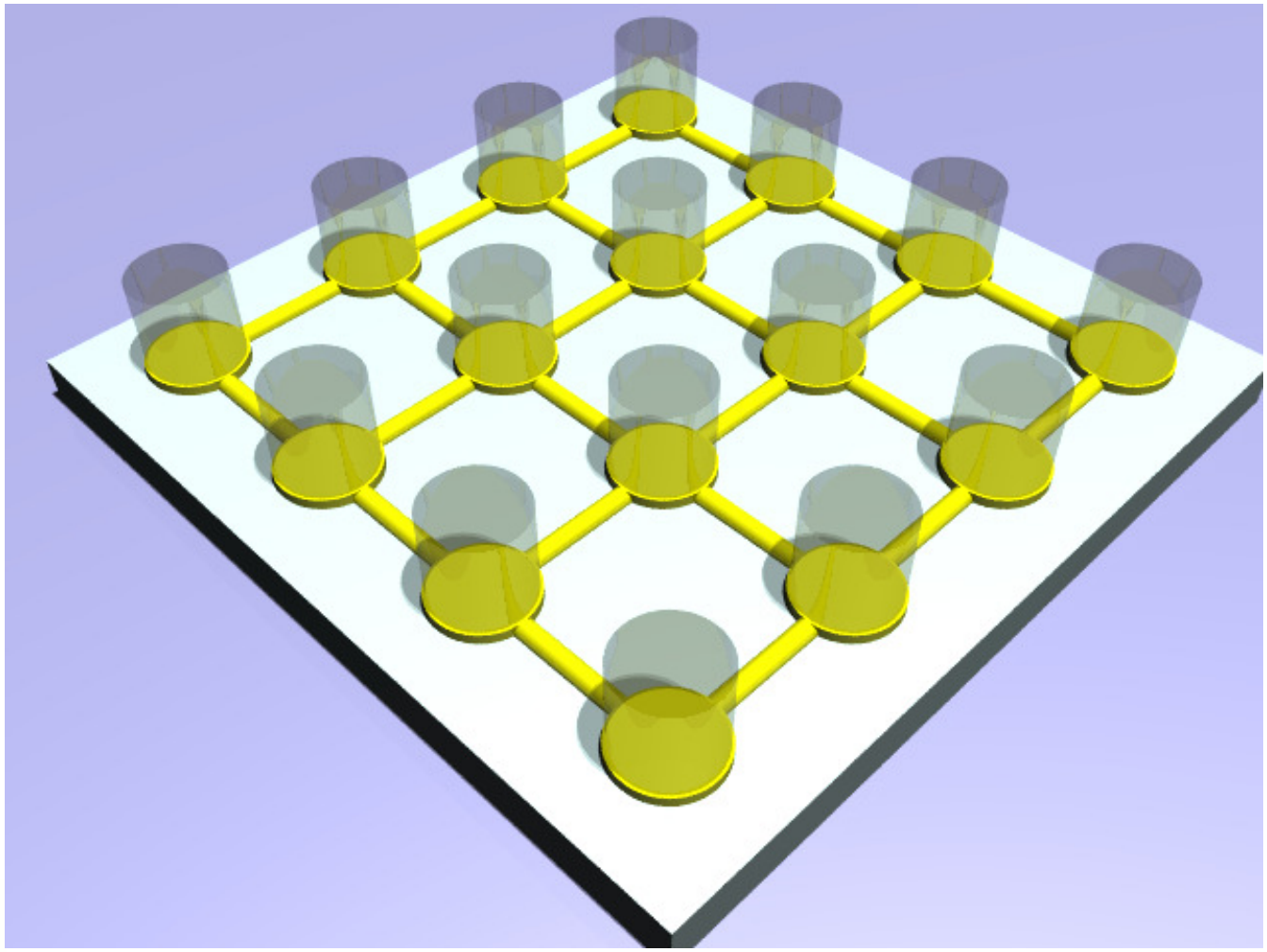
- Applications execute from external memory, **except** when hotspots are reached.
- Software **binary translation** is used to retarget **hotspots** for the MCGREP array.
- Hotspot tasks are distributed to all or part of the reconfigurable array.
- ILP exploited as multiple processors may be used to execute each task.

RTS York

MCGREP Paradigm

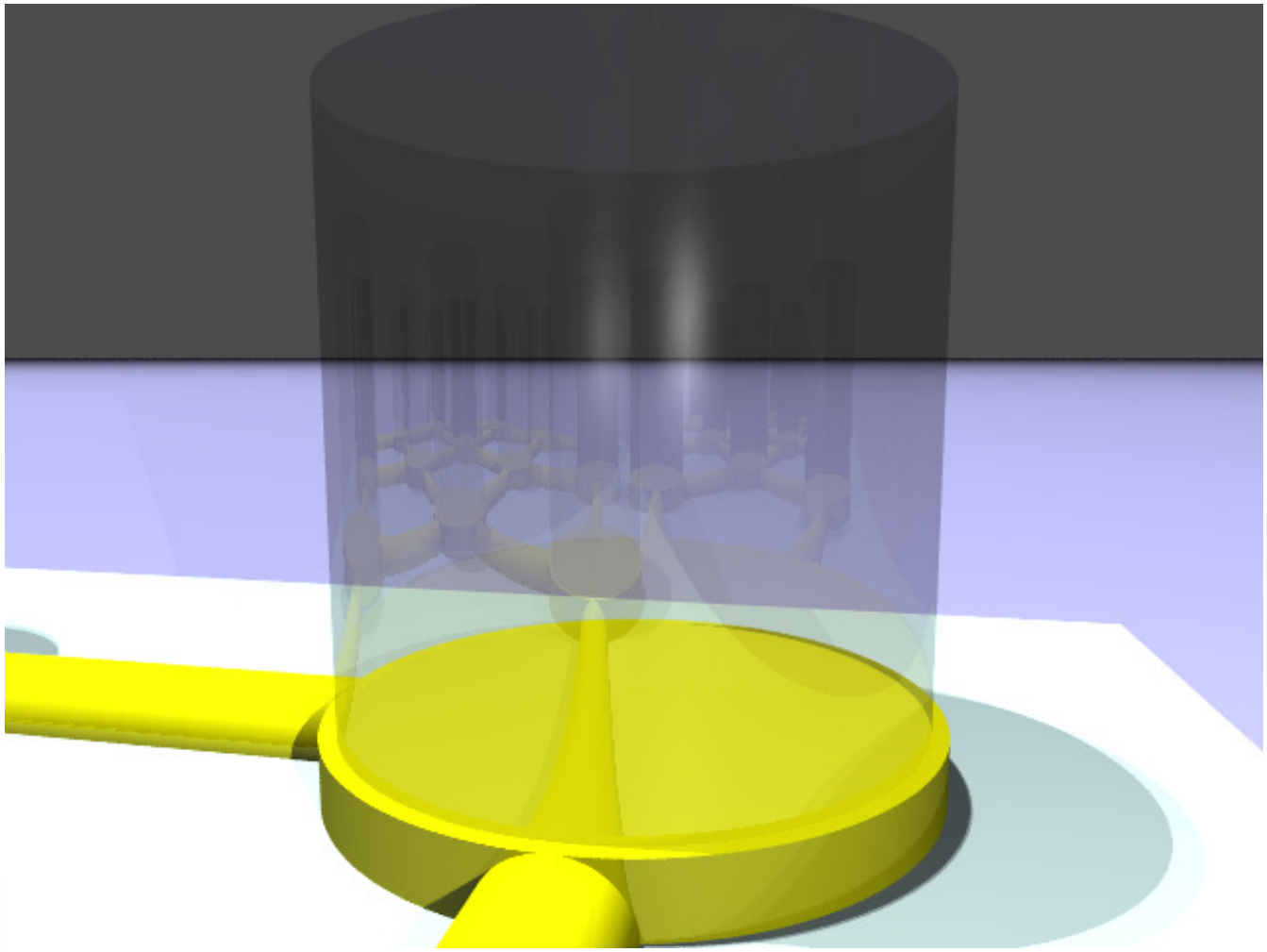
- Applications execute from external memory, **except** when hotspots are reached.
- Software **binary translation** is used to retarget **hotspots** for the MCGREP array.
- Hotspot tasks are distributed to all or part of the reconfigurable array.
- ILP exploited as multiple processors may be used to execute each task.
- Applications are speeded up, predictability and reusability are retained.

Distributing Hotspots (1)



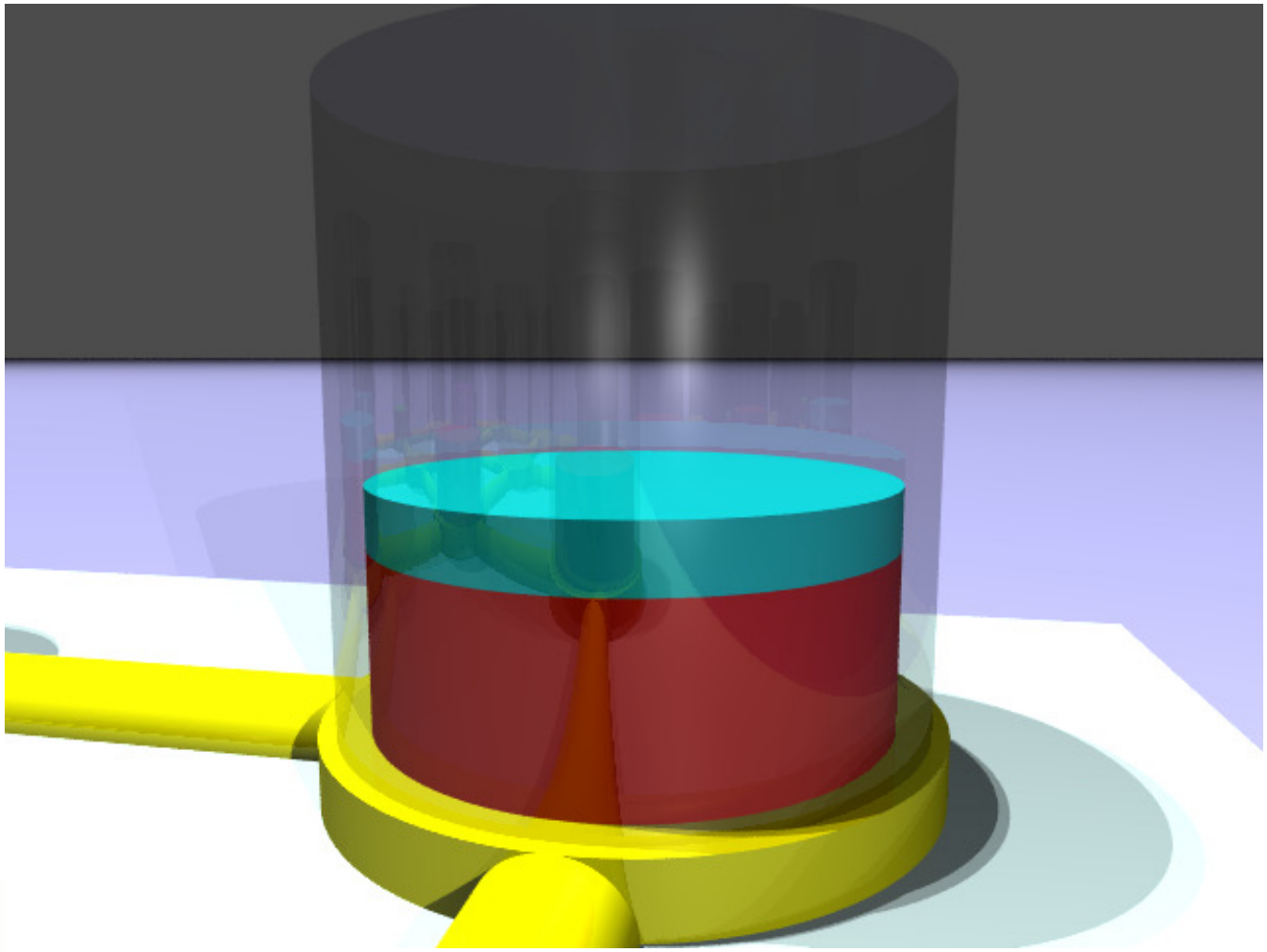
RTS York

Distributing Hotspots (2)

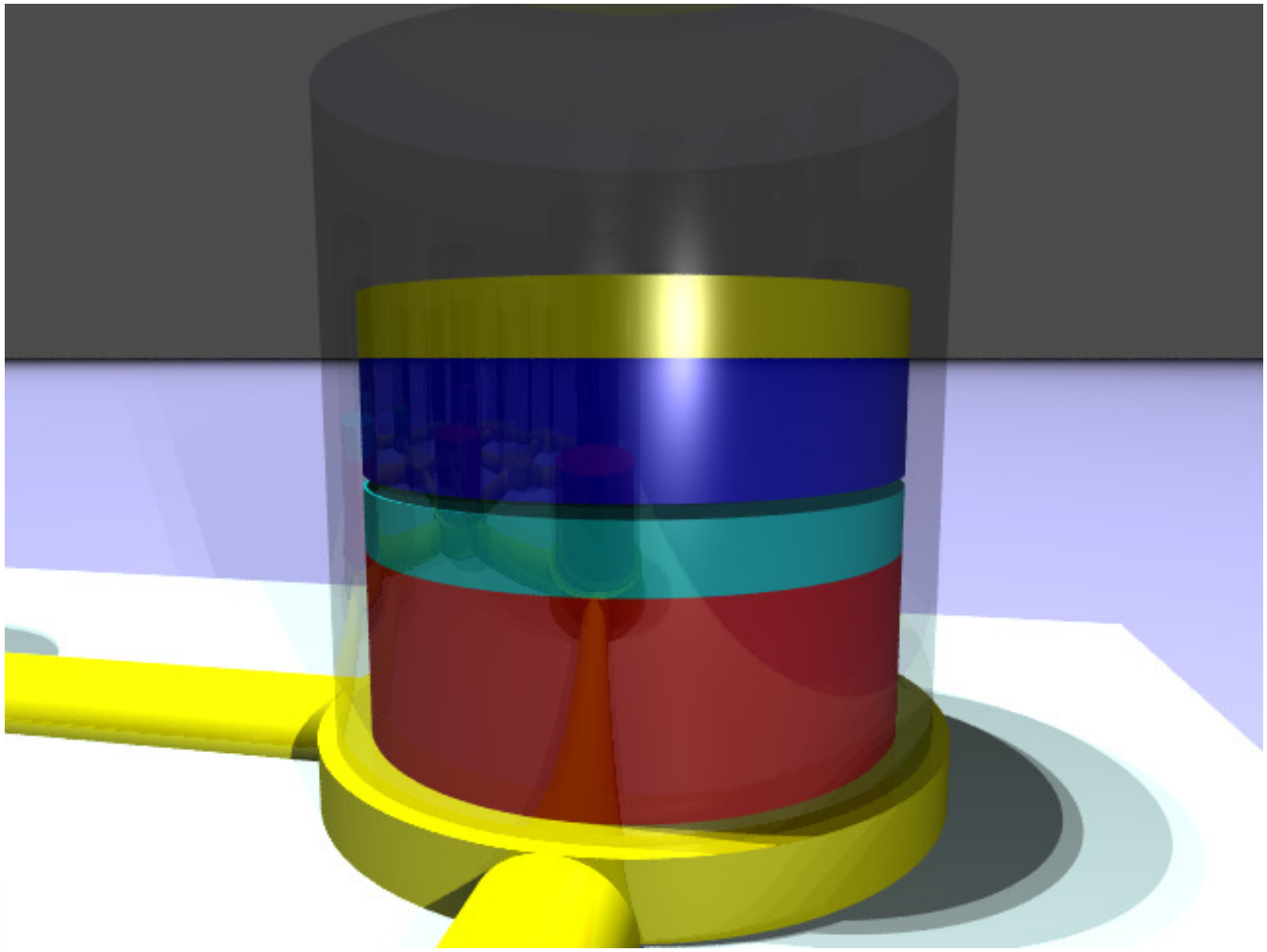


RTS York

Distributing Hotspots (3)

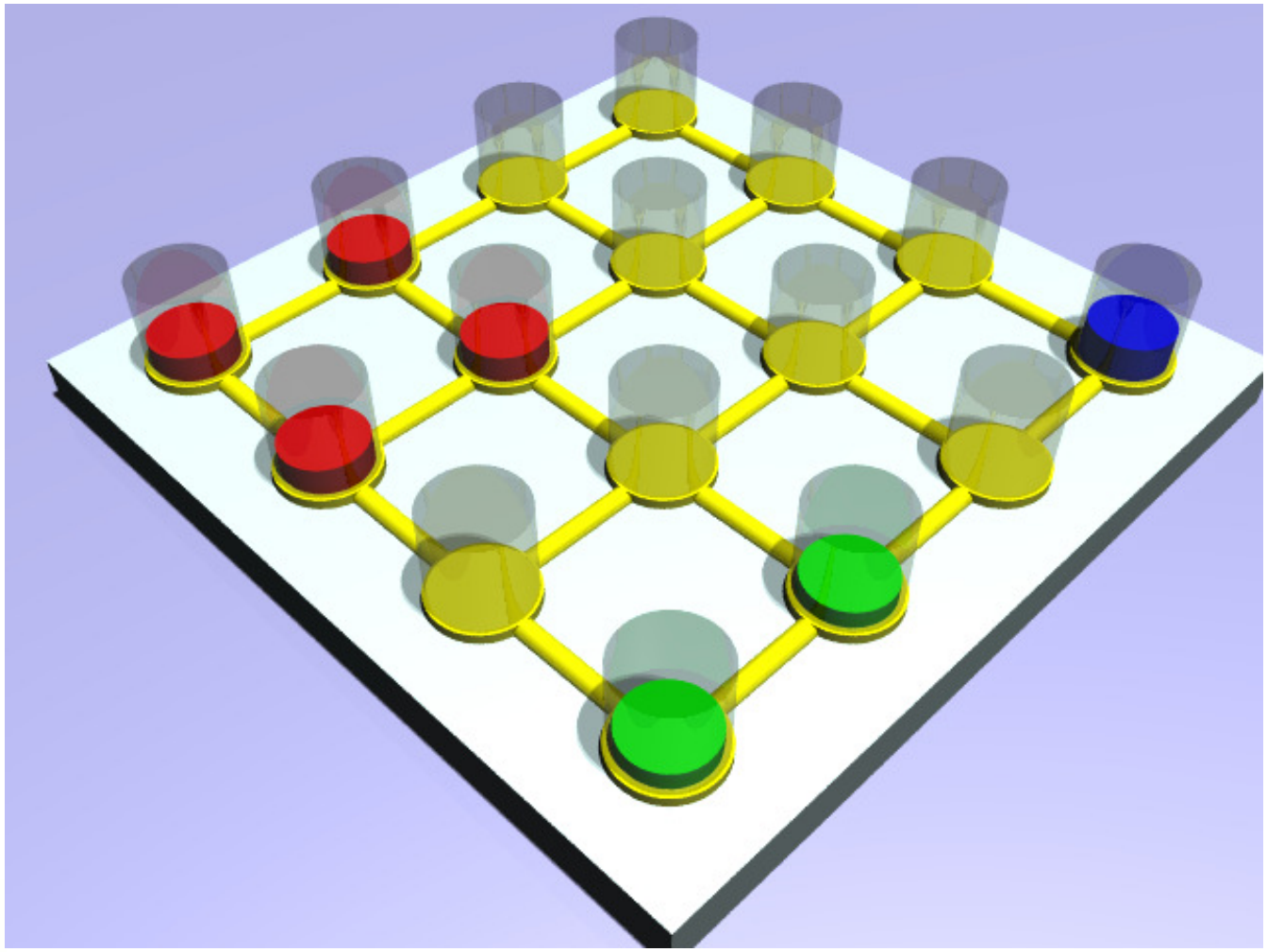


Distributing Hotspots (4)

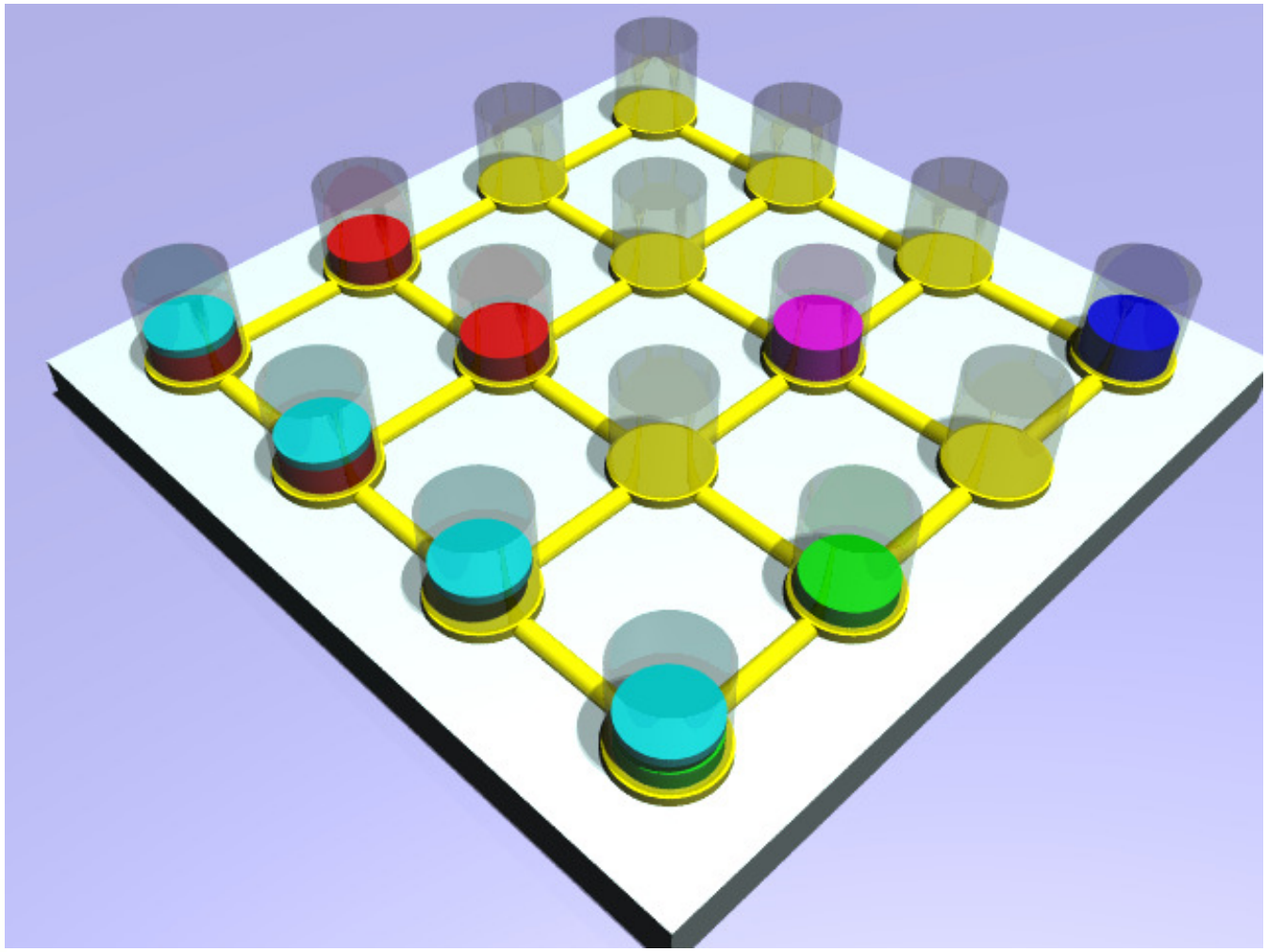


RTS York

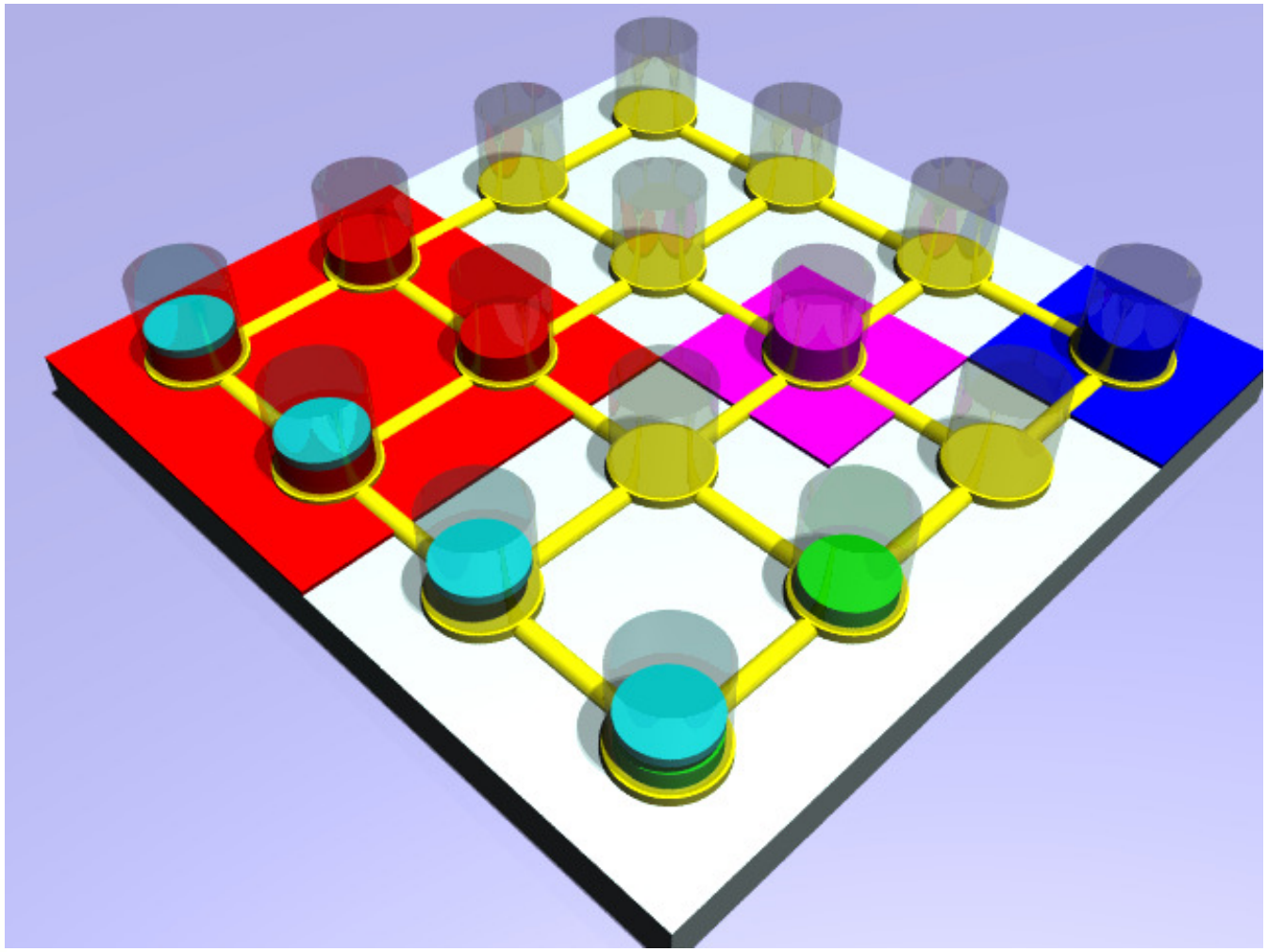
Distributing Hotspots (5)



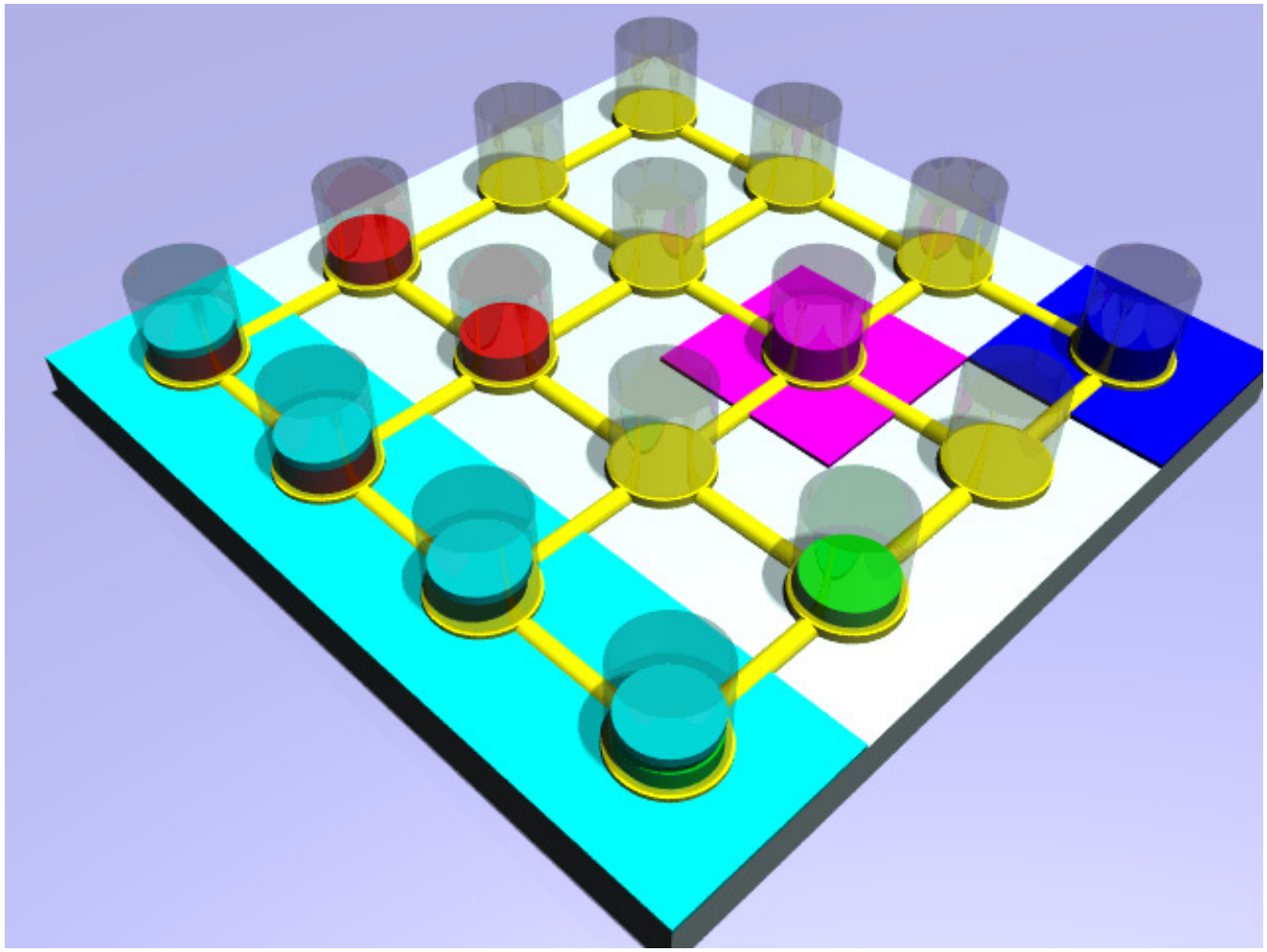
Distributing Hotspots (6)



Distributing Hotspots (7)

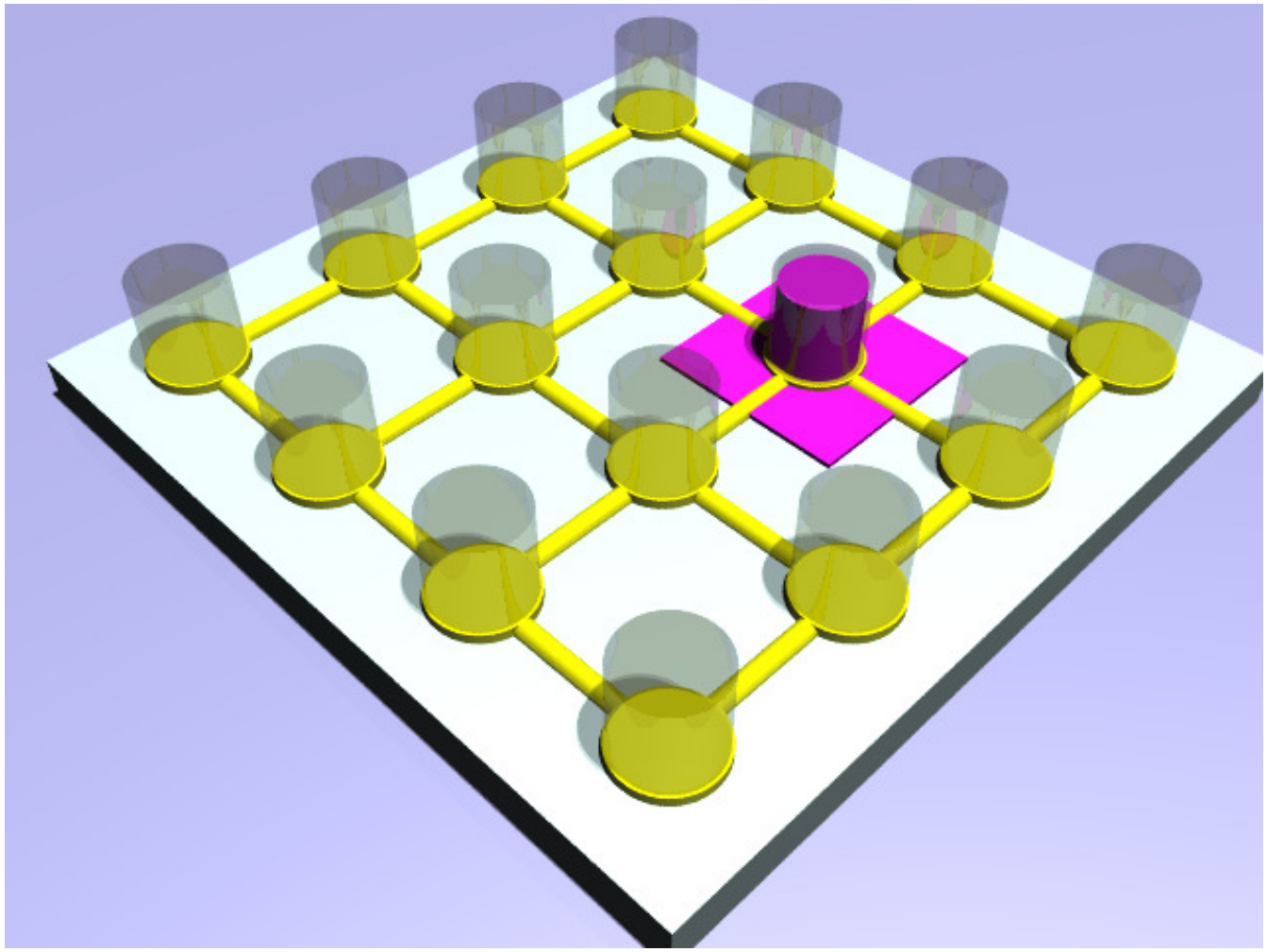


Distributing Hotspots (8)



RTS York

Distributing Hotspots (9)



Simulation

Simulation is

- The process of modelling the behaviour of one system using another.

Simulation

Simulation is

- The process of modelling the behaviour of one system using another.

Simulation allows experimentation with a new design while only parts of the design are complete.

MCGREP Needs A Simulator

A simulator is required:

1. for **experiments** using real programs.

MCGREP Needs A Simulator

A simulator is required:

1. for **experiments** using real programs.
2. for **hardware verification**.

MCGREP Needs A Simulator

A simulator is required:

1. for **experiments** using real programs.
2. for **hardware verification**.
3. for **debugging** MCGREP tools.

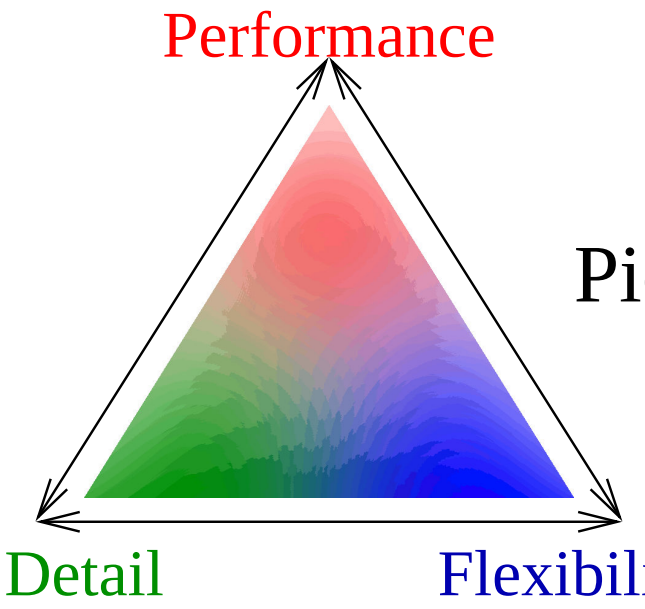
Two Classes of Simulator

There are two types of architectural simulator:

1. Functional (captures function but not timing),
2. Performance (captures function and timing).

“Tao of Simulation”

The tradeoffs in simulator design:



Pick two

Example

Simplescalar:

Name	Speed	Detail	Description
`sim-fast`	Most	Least	Functional, no checks
`sim-safe`			Functional, with checks
`sim-uop`			Functional, μop
`sim-outorder`	Least	Most	Performance

MCGREP Simulator

Requirements are sorted into priority order:

1. **Detail** - Architectural accuracy is essential.

MCGREP Simulator

Requirements are sorted into priority order:

1. **Detail** - Architectural accuracy is essential.
2. **Flexibility** - Important for experimentation and debugging, which require extensible software.

RTS York

MCGREP Simulator

Requirements are sorted into priority order:

1. **Detail** - Architectural accuracy is essential.
2. **Flexibility** - Important for experimentation and debugging, which require extensible software.
3. **Performance** - Speed is important for testing real applications.

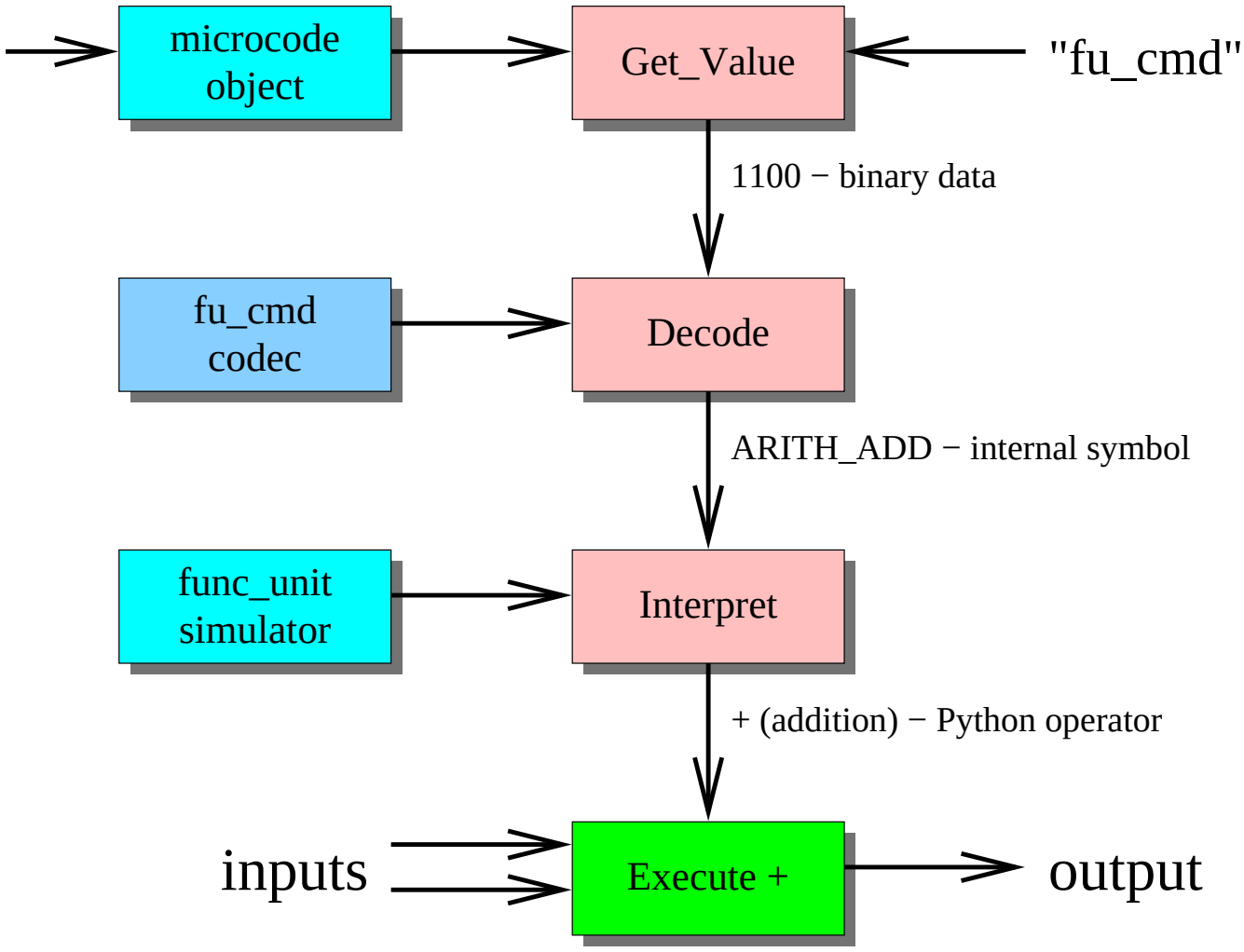
RTS York

Simulator 1

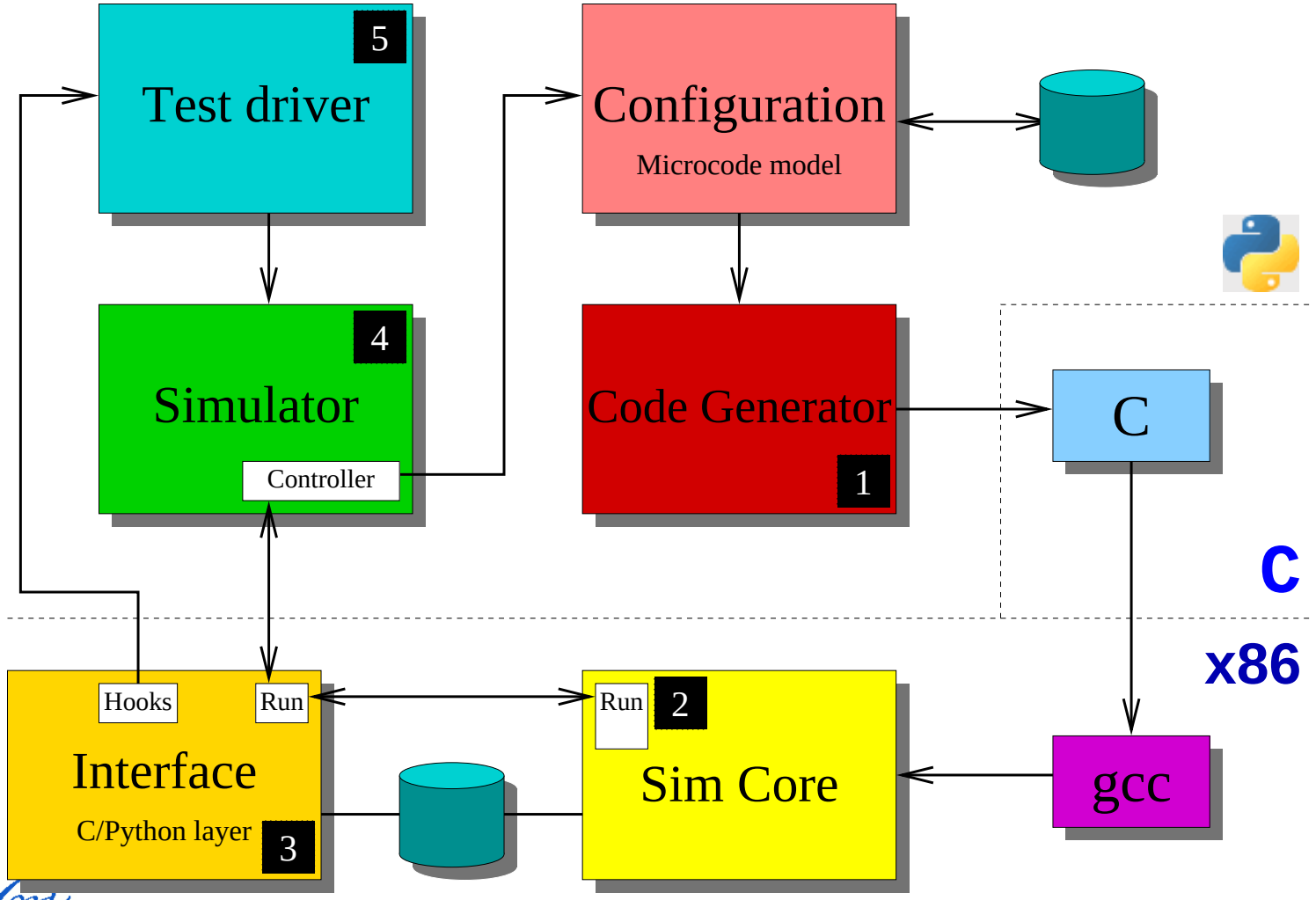
First simulator:

1. Written in Python,
2. Approx. 1000 instructions per second,
3. Detailed and flexible, but very poor performance,
4. Impossible to test real applications!

Poor Performance



New Simulator



RTS York

Flexibility

- First simulator was extended by subclassing.

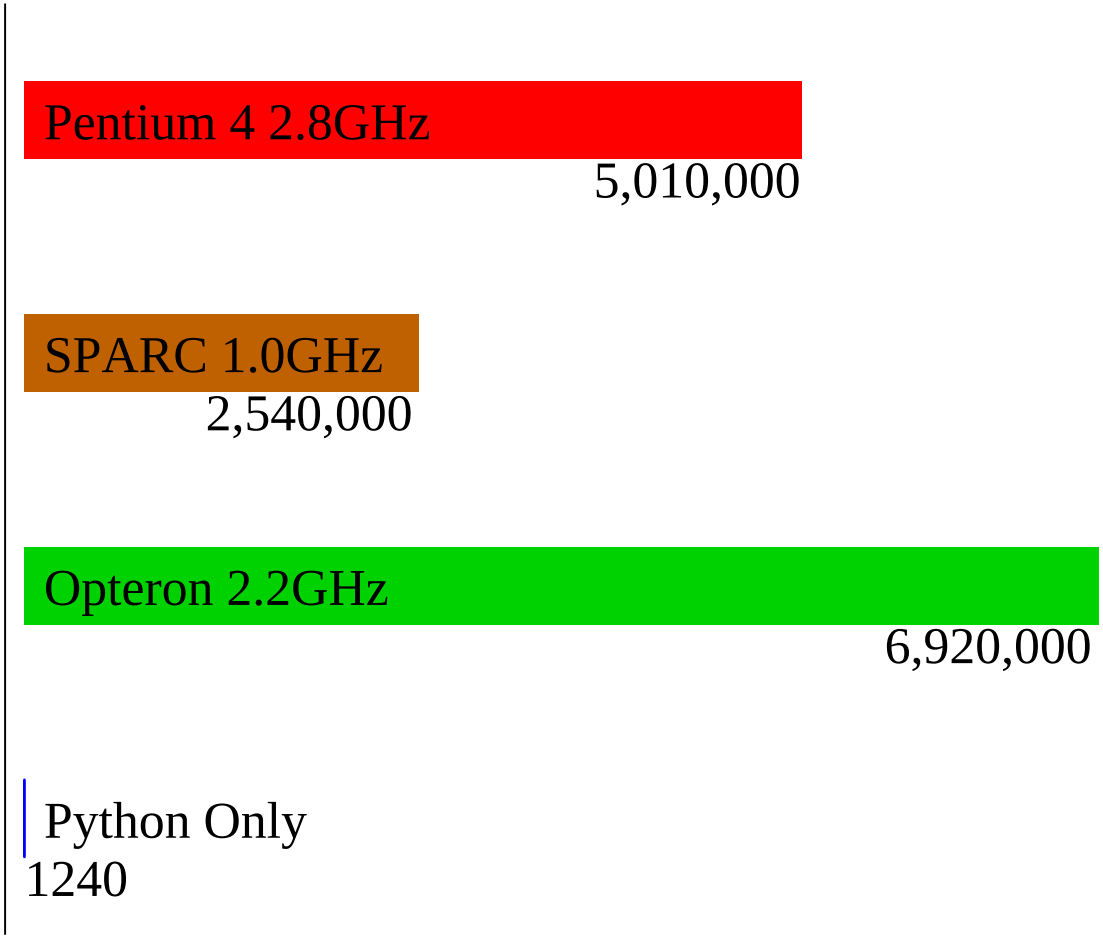
Flexibility

- First simulator was extended by subclassing.
- New simulator is extended by **hook functions**.

Flexibility

- First simulator was extended by subclassing.
- New simulator is extended by **hook functions**.
- This is sufficient for current experiments and tests.

Performance



(Figures in instructions per second)

Test Drivers

Driver 1: Tests this simulator by comparison with another.

- Used OpenRISC simulator.

Test Drivers

Driver 1: Tests this simulator by comparison with another.

- Used OpenRISC simulator.
- A trace was generated for a number of test programs.

Test Drivers

Driver 1: Tests this simulator by comparison with another.

- Used OpenRISC simulator.
- A trace was generated for a number of test programs.
- Test programs execute on MCGREP. On each dispatch, the trace test driver compares trace data against the register file.

RTS York

Test Drivers

Driver 1: Tests this simulator by comparison with another.

- Used OpenRISC simulator.
- A trace was generated for a number of test programs.
- Test programs execute on MCGREP. On each dispatch, the trace test driver compares trace data against the register file.
- Functional equivalence demonstrated!

Test Drivers

Driver 2: Test hotspot operation.

- Hotspots are executed by microcode...

Test Drivers

Driver 2: Test hotspot operation.

- Hotspots are executed by microcode...
- Microcode execution should be equivalent to “normal” execution.

Test Drivers

Driver 2: Test hotspot operation.

- Hotspots are executed by microcode...
- Microcode execution should be equivalent to “normal” execution.
- The microcode test driver compares results of two types of execution.

Test Drivers

Driver 2: Test hotspot operation.

- Hotspots are executed by microcode...
- Microcode execution should be equivalent to “normal” execution.
- The microcode test driver compares results of two types of execution.
- Functional equivalence demonstrated!

Applications

The simulator has been used for:

- Experiments on MCGREP.

Applications

The simulator has been used for:

- Experiments on MCGREP.
- Online compilation.

Applications

The simulator has been used for:

- Experiments on MCGREP.
- Online compilation.
- Debugging MCGREP.

Applications

The simulator has been used for:

- Experiments on MCGREP.
- Online compilation.
- Debugging MCGREP.
- And hardware development.

Conclusion

The simulator has enabled:

- New types of experiment (*e.g.* online compilation).

Conclusion

The simulator has enabled:

- New types of experiment (*e.g.* online compilation).
- Unit testing of MCGREP hardware.

Conclusion

The simulator has enabled:

- New types of experiment (*e.g.* online compilation).
- Unit testing of MCGREP hardware.
- Debugging.

It has also improved performance by a factor of 1000.

RTS York

Questions?

jack@cs.york.ac.uk

Real Time Systems Group
Department of Computer Science
University of York