Adaptive Hardware in Autonomous and Evolvable Embedded Systems

Stephan Kubisch, Ronald Hecht, Dirk Timmermann
stephan.kubisch@uni-rostock.de

University of Rostock,
Institute of Applied Microelectronics and Computer Engineering

February 15th, embedded world 2006, Nuremberg

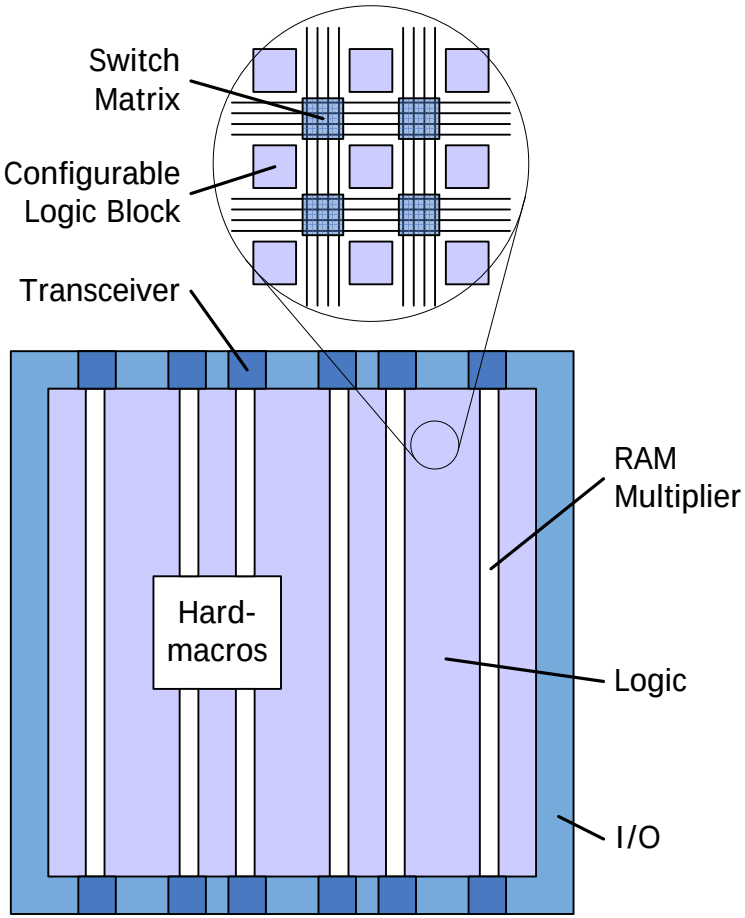
Outline

Outline

Current Trends

Embedded Systems

- ...play major role in a broad range of applications
- using FPGAs is very popular
- FPGAs are/become the embedded system



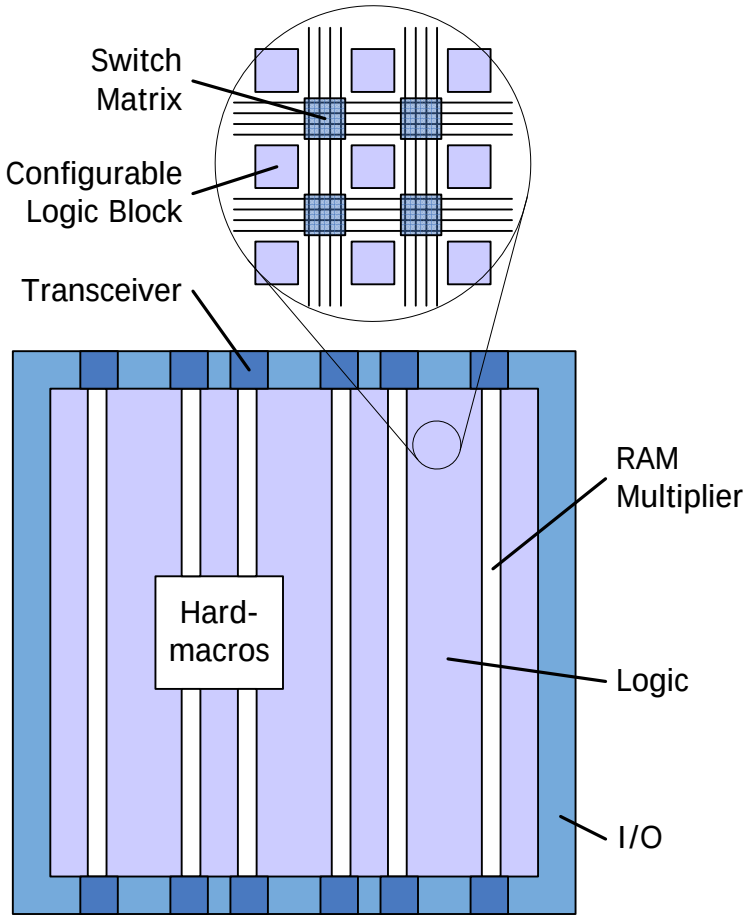
Current Trends

Embedded Systems

- ...play major role in a broad range of applications
- using FPGAs is very popular
- FPGAs are/become the embedded system

Growing Functional Spectrum

- “Products must do more, cost less, and be quickly available!“
- to meet high in-field demands
- manage complexity/simplify the development

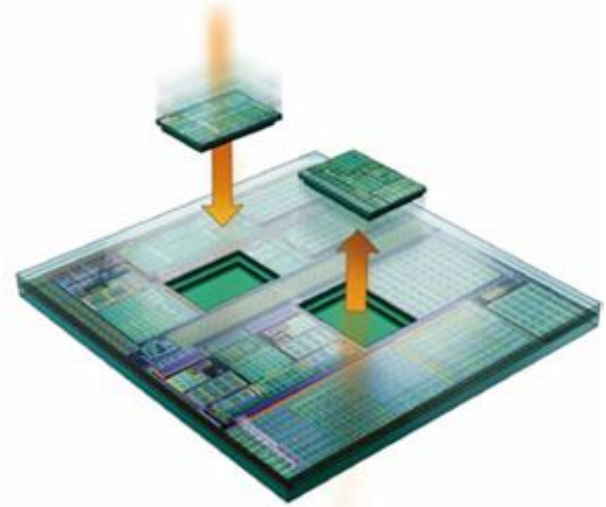


Current Trends

Partial & Dynamic Reconfiguration (PDR)

partial only a distinct part of the FPGA's resources is (re)configured

dynamic @ runtime! remaining design modules are not touched



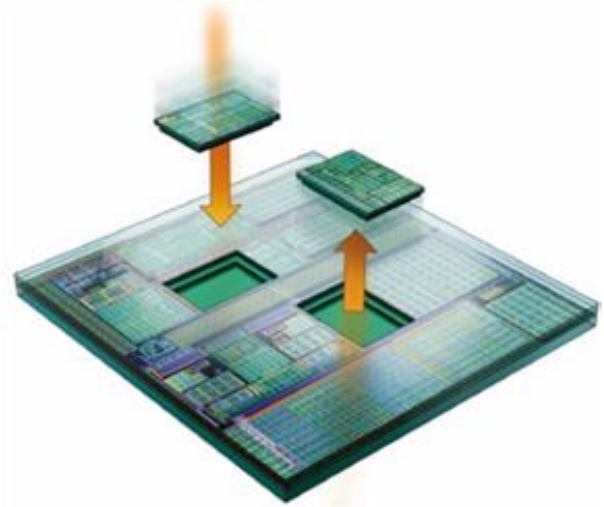
Current Trends

Partial & Dynamic Reconfiguration (PDR)

partial only a distinct part of the FPGA's resources is (re)configured

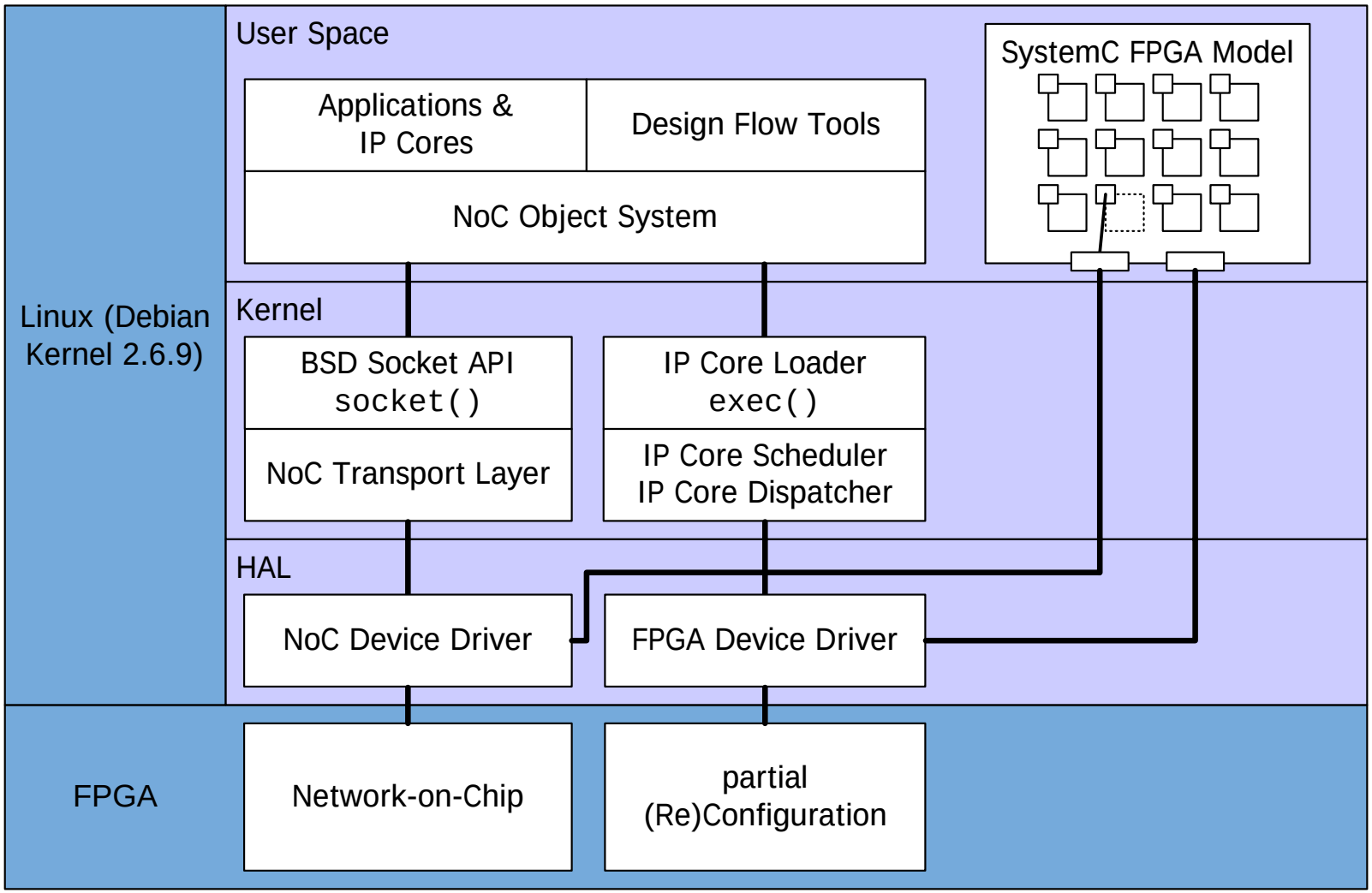
dynamic @ runtime! remaining design modules are not touched

- increased flexibility
- in-field updates & bug-fixes
- beneficial for applications allowing no interrupts

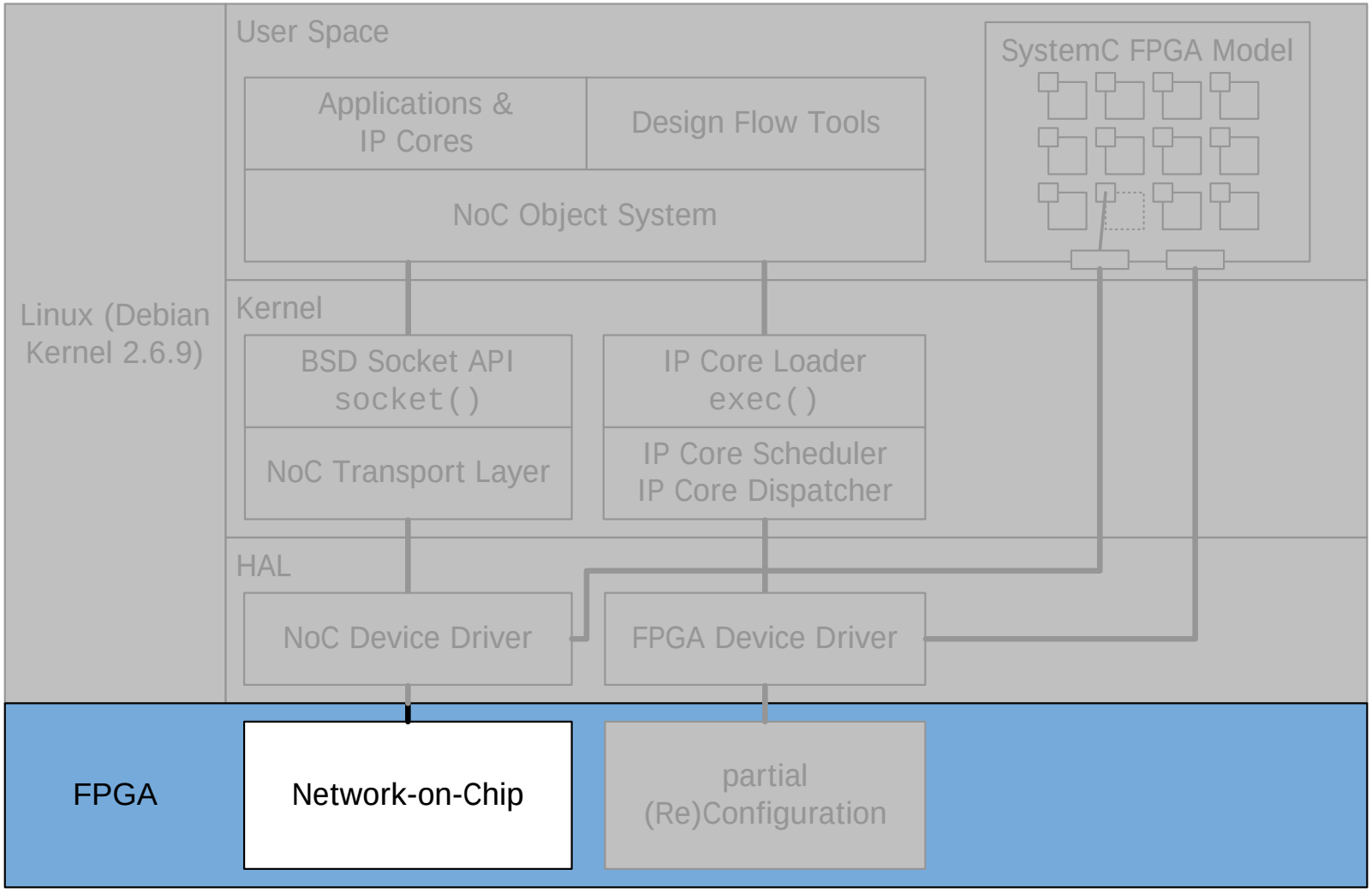


Outline

The Systems Architecture



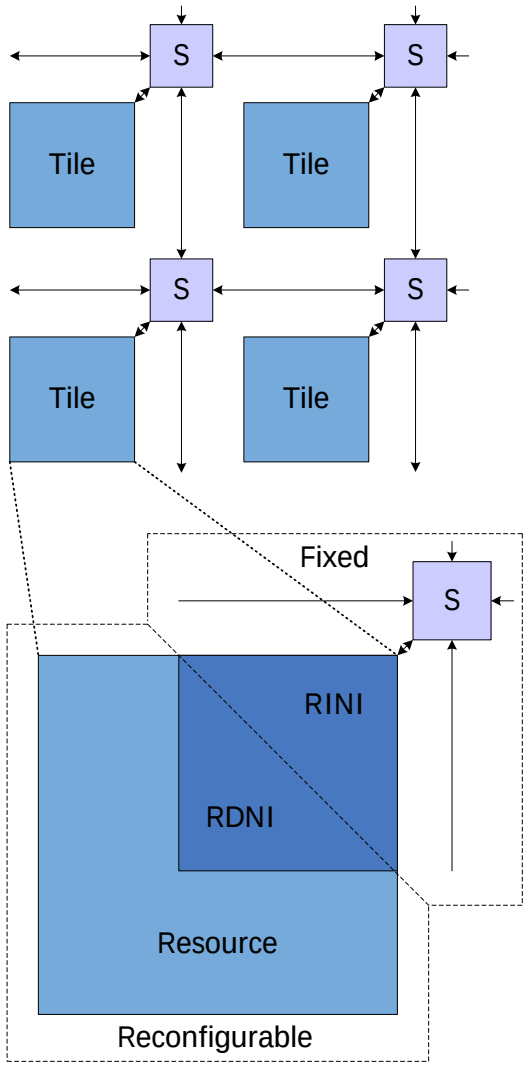
The Network-on-Chip (NoC)



The NoC Based Communication Grid

NoC Building Blocks

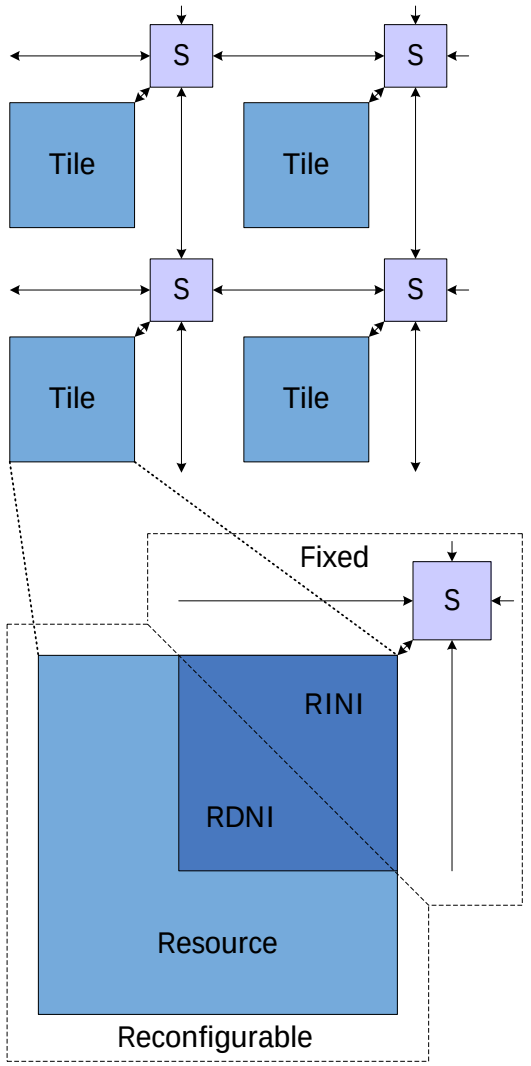
- Tiles containing hardmacros and/or user defined logic



The NoC Based Communication Grid

NoC Building Blocks

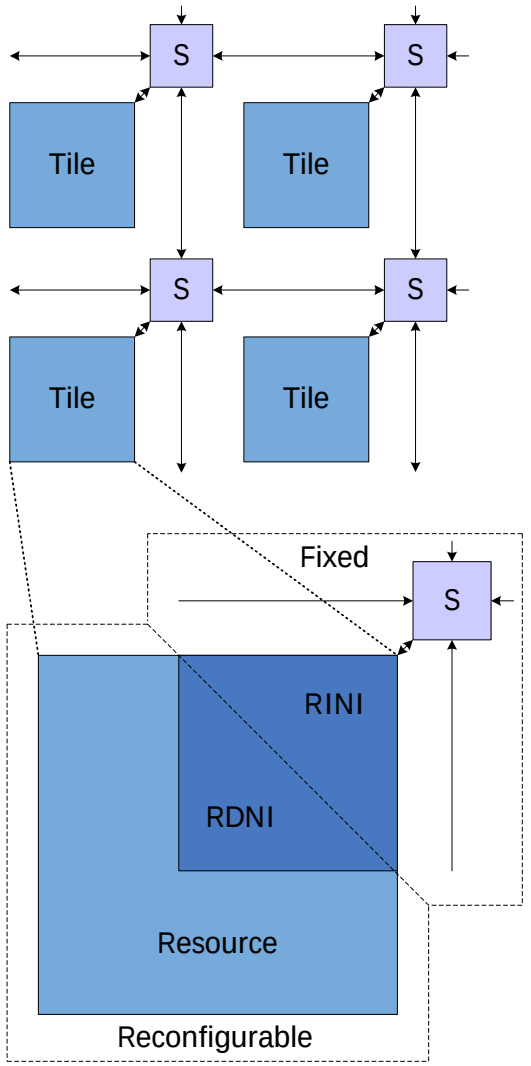
- Tiles containing hardmacros and/or user defined logic
- Resource Network Interface (RNI)
  - RDNI
  - RINI



The NoC Based Communication Grid

NoC Building Blocks

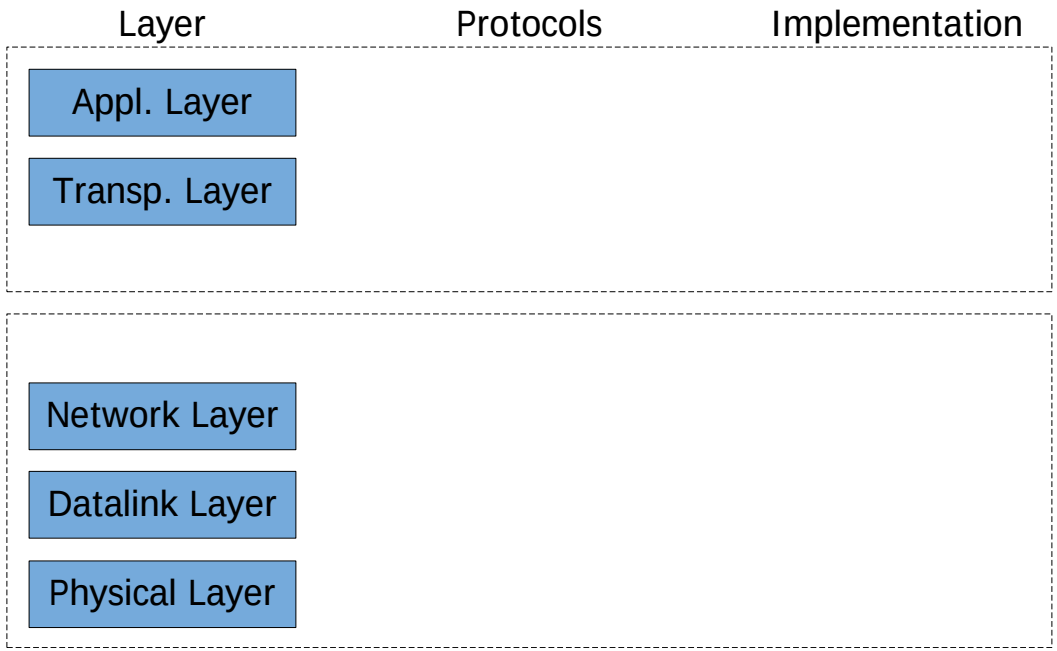
- Tiles containing hardmacros and/or user defined logic
- Resource Network Interface (RNI)
  - RDNI
  - RINI
- Switches (S) to direct the traffic
- Wires connecting the switches



The NoC Protocol Stack

Layered Stack

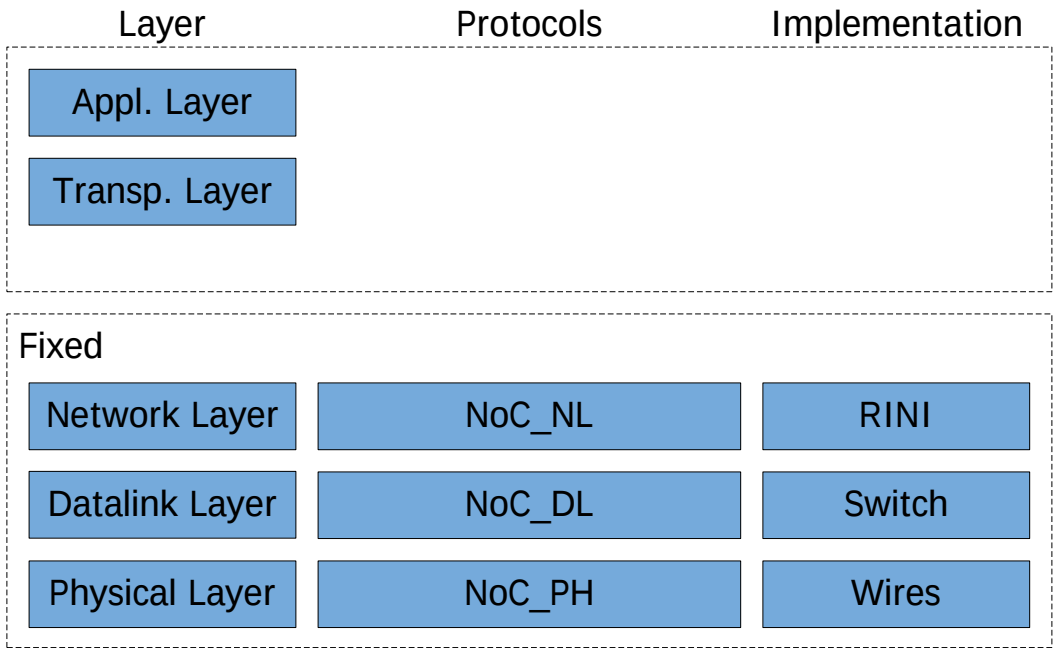
- similar to the layered OSI model



The NoC Protocol Stack

Layered Stack

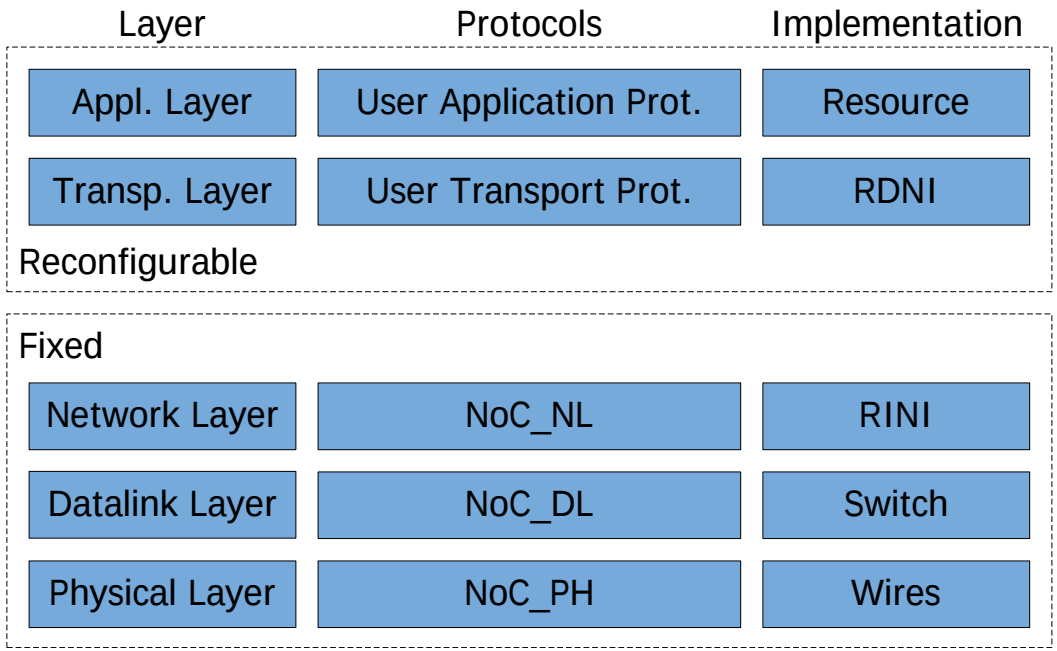
- similar to the layered OSI model
- fixed part for the lower layer protocols



The NoC Protocol Stack

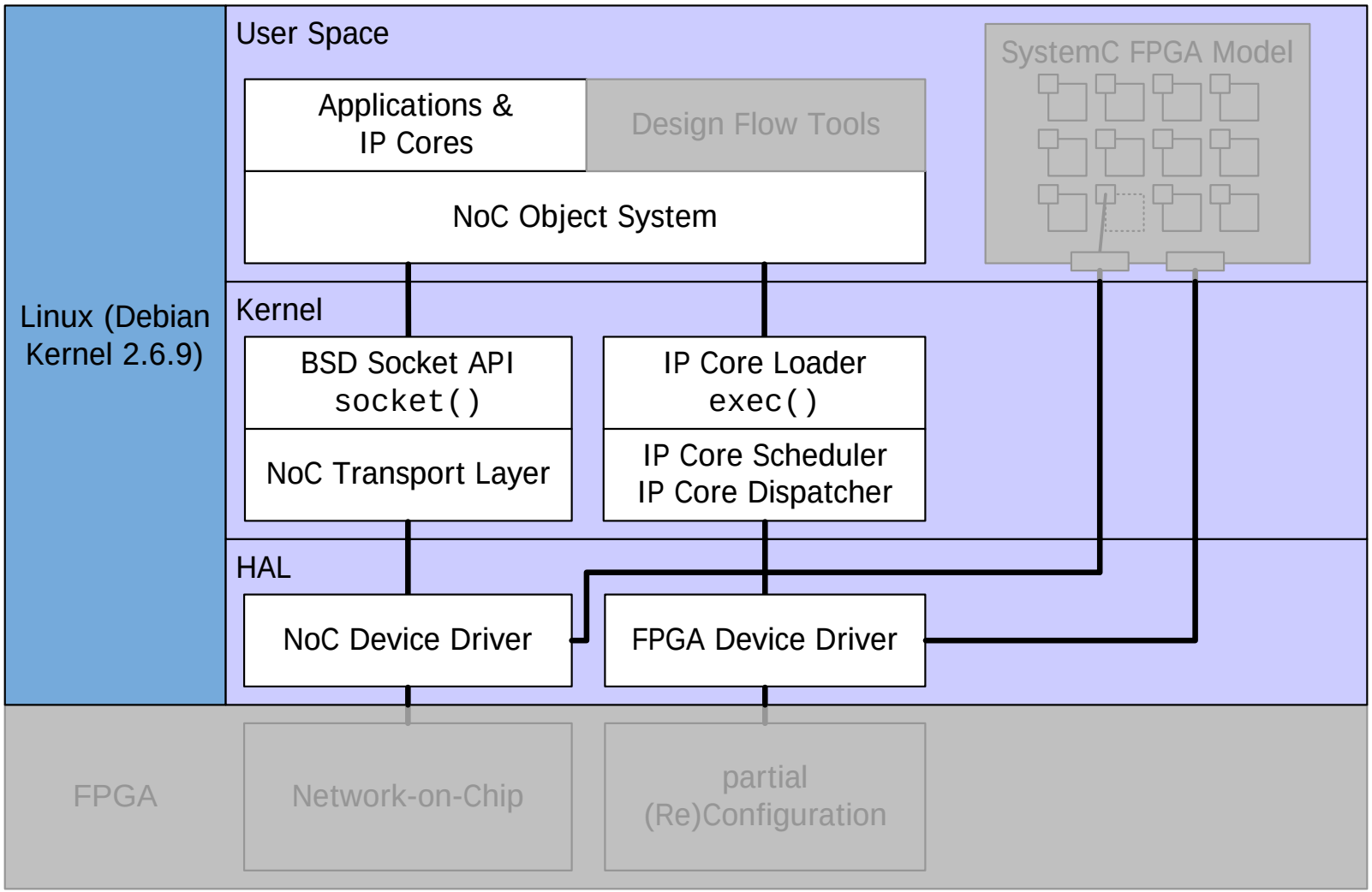
Layered Stack

- similar to the layered OSI model
- reconfigurable part for user defined protocols
- fixed part for the lower layer protocols



Outline

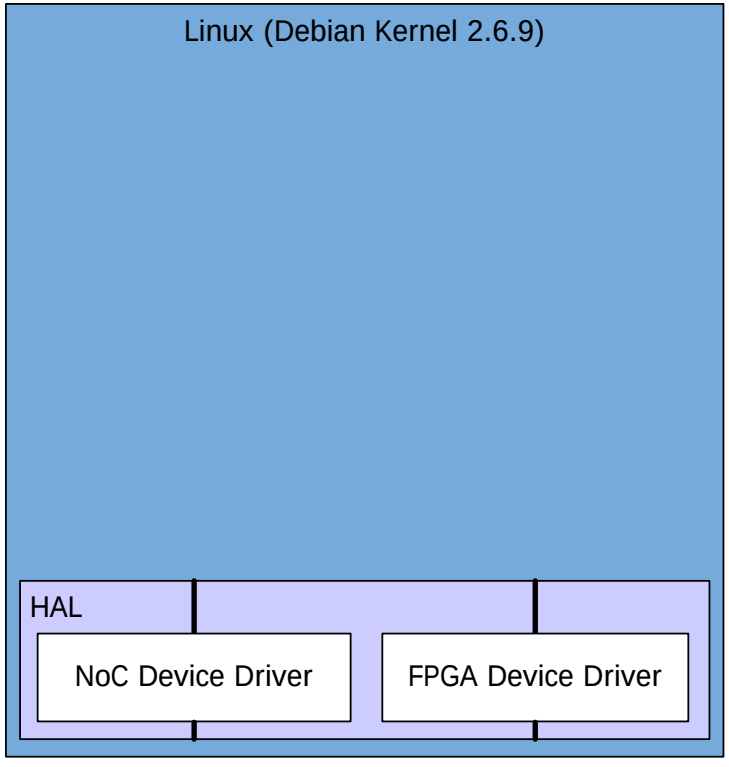
The Operating System



Operating System support for PDR

Linux Modifications

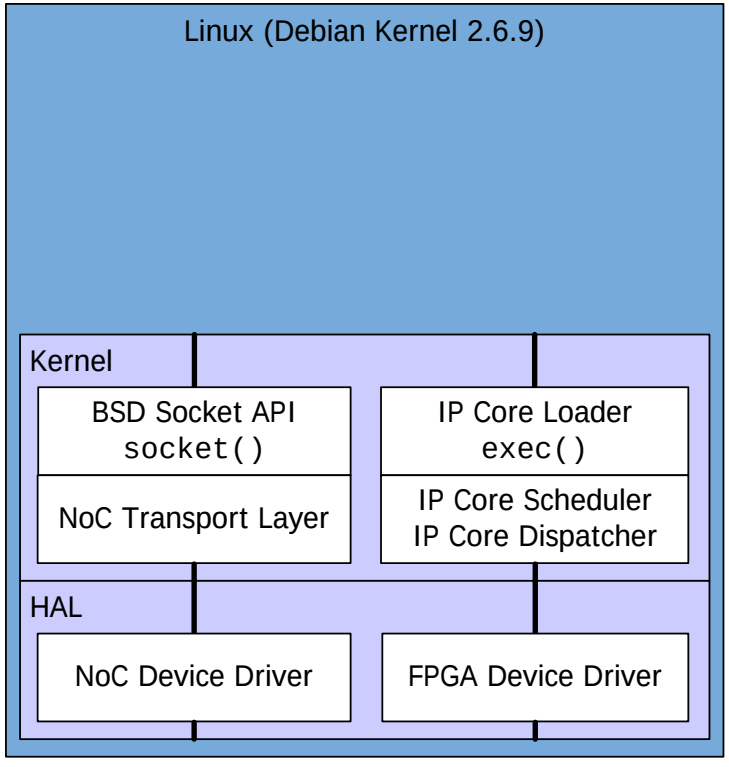
- Runs on embedded microprocessor
- Hardware Abstraction Layer (HAL) contains device drivers



Operating System support for PDR

Linux Modifications

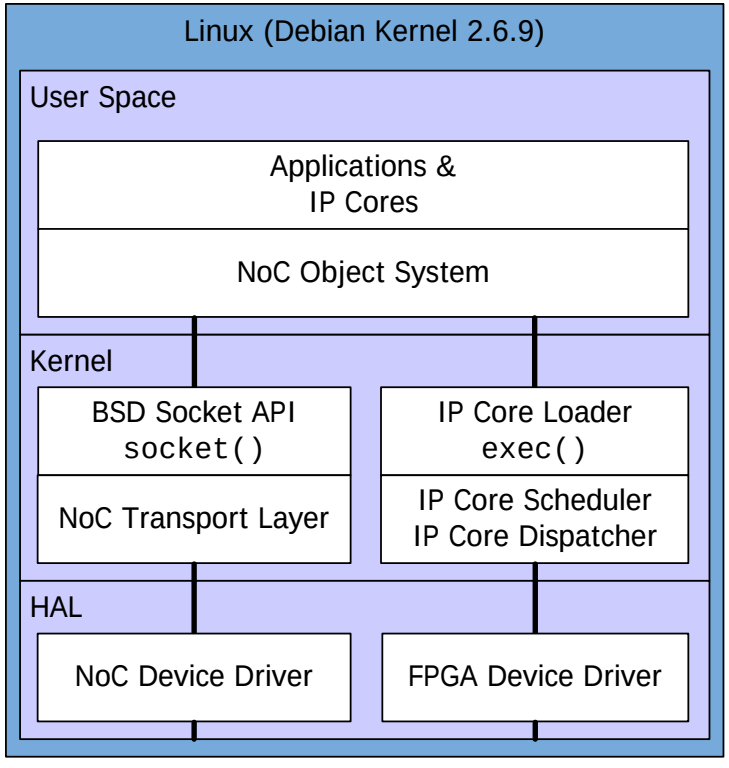
- Runs on embedded microprocessor
- Kernel with mechanisms for IP Core handling
- Hardware Abstraction Layer (HAL) contains device drivers



Operating System support for PDR

Linux Modifications

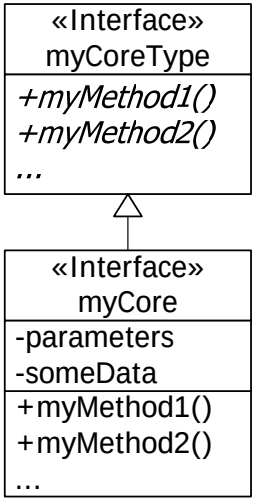
- Runs on embedded microprocessor
- User Space contains library of IP cores and applications
- Kernel with mechanisms for IP Core handling
- Hardware Abstraction Layer (HAL) contains device drivers



Operating System support for PDR

Adapted Mechanisms

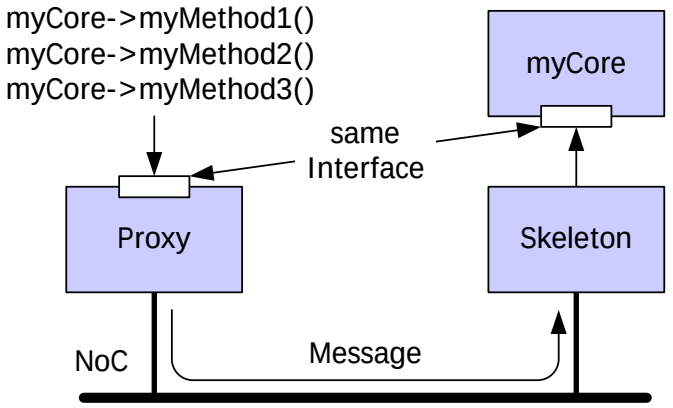
- "Everything is an Object!"
  → object oriented programming



Operating System support for PDR

Adapted Mechanisms

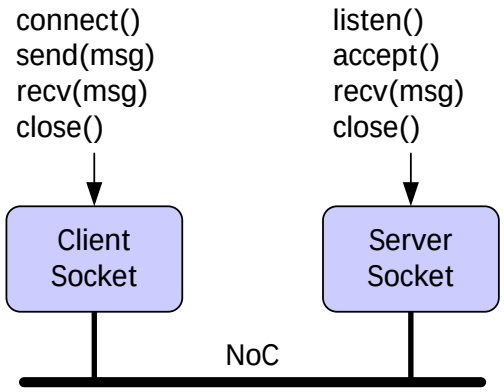
- "Everything is an Object!"
  $\longrightarrow$ object oriented programming
- NoC $\approx$ distributed object systems
  - Remote Method Invocation (RMI)



Operating System support for PDR

Adapted Mechanisms

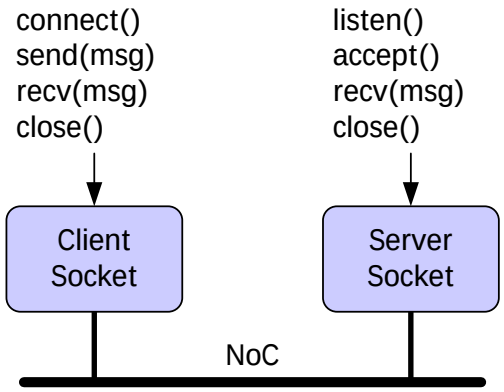
- "Everything is an Object!"
  $\longrightarrow$ object oriented programming
- NoC $\approx$ distributed object systems
  - Remote Method Invocation (RMI)
- BSD socket API



Operating System support for PDR

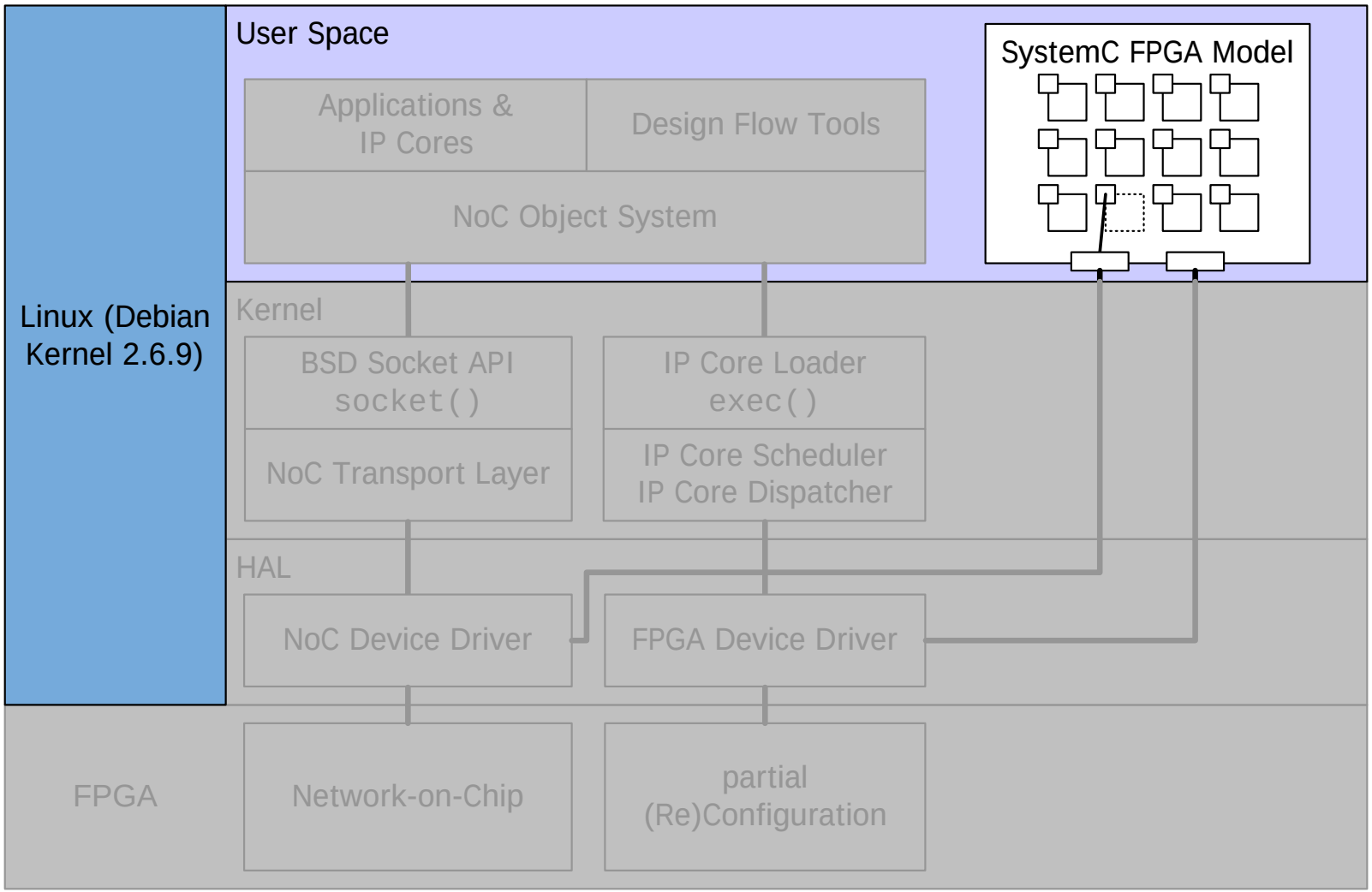
Adapted Mechanisms

- "Everything is an Object!"
  $\rightarrow$ object oriented programming
- NoC $\approx$ distributed object systems
  - Remote Method Invocation (RMI)
- BSD socket API
- Linux device drivers
  - TUN/TAP driver
  - char device driver
- system is more graspable and clear



Outline

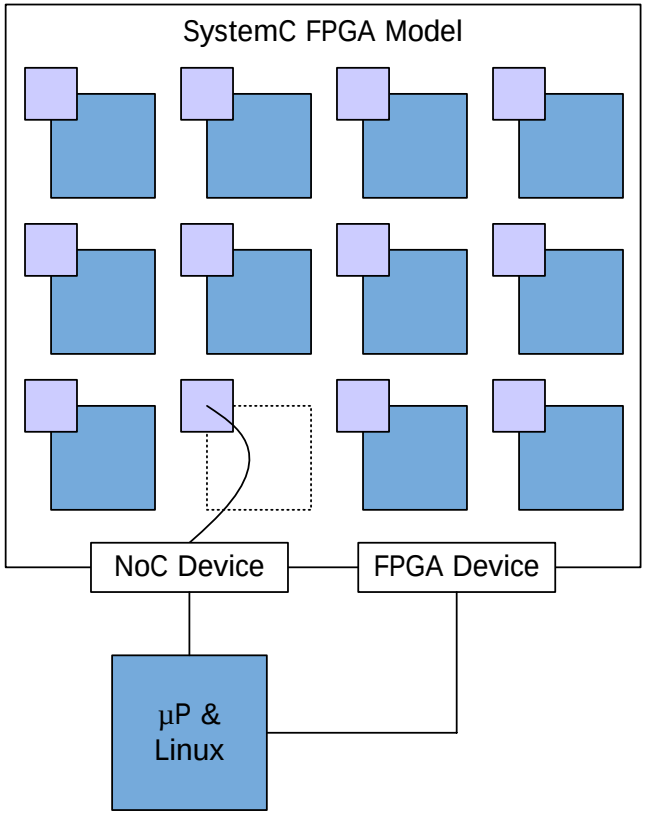
The Integrated System Model



The Integrated System Model

What is it?

- functional model of NoC & FPGA
- SystemC, channel concept
- virtual NoC & FPGA, virtual hardware



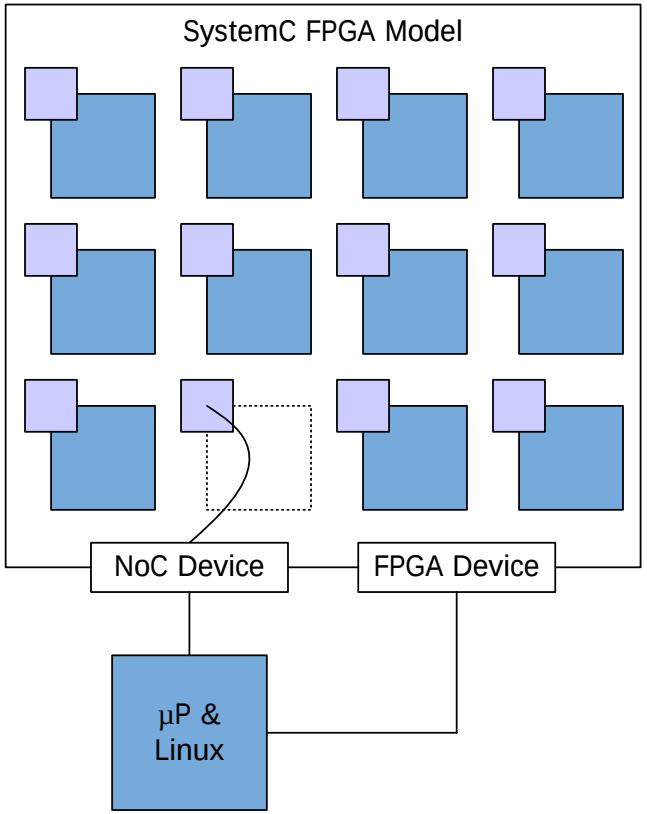
The Integrated System Model

What is it?

- functional model of NoC & FPGA
- SystemC, channel concept
- virtual NoC & FPGA, virtual hardware

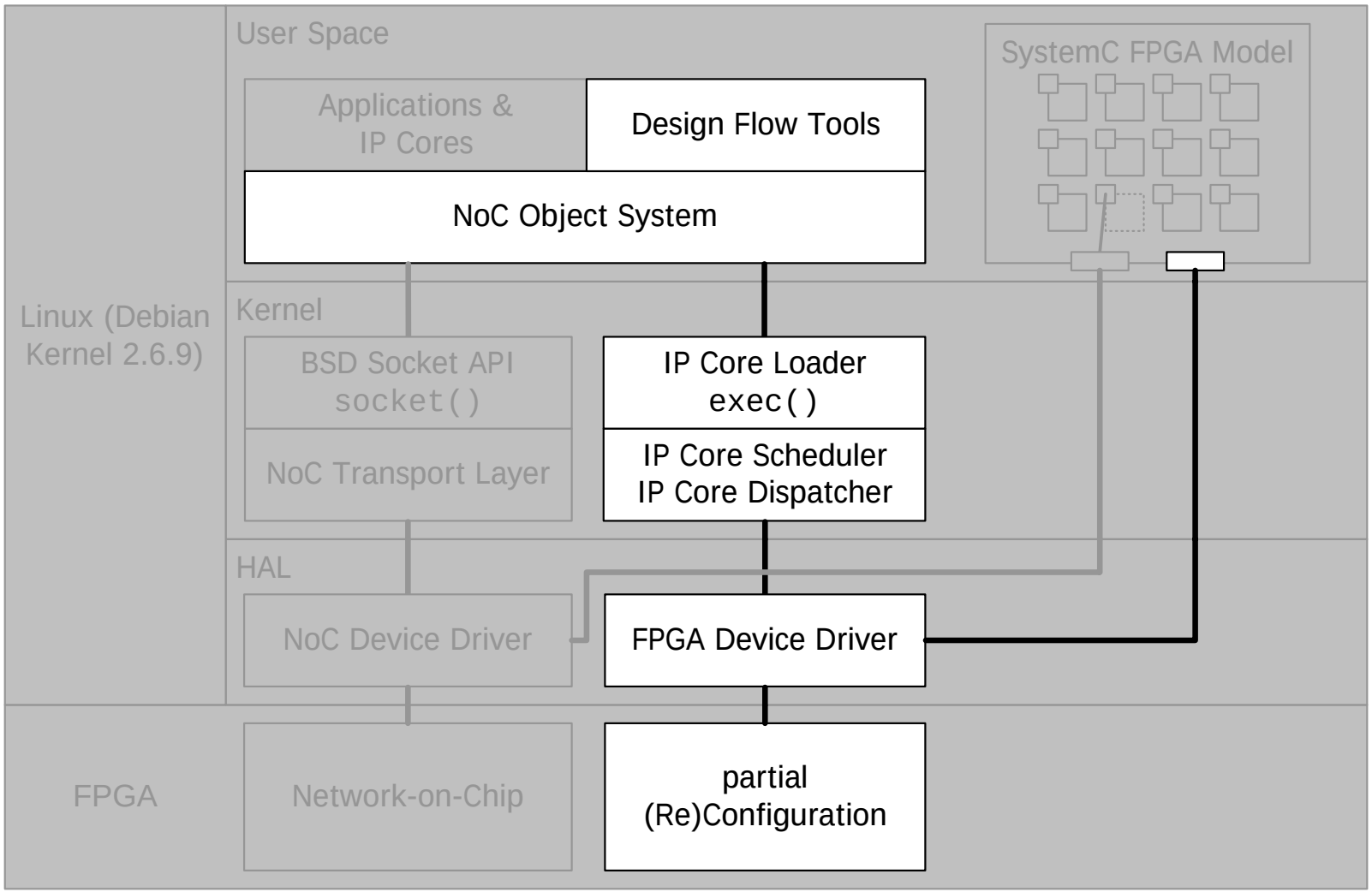
Feasible for...

- simulator during development
- on-board simulation
- container for software executables



Outline

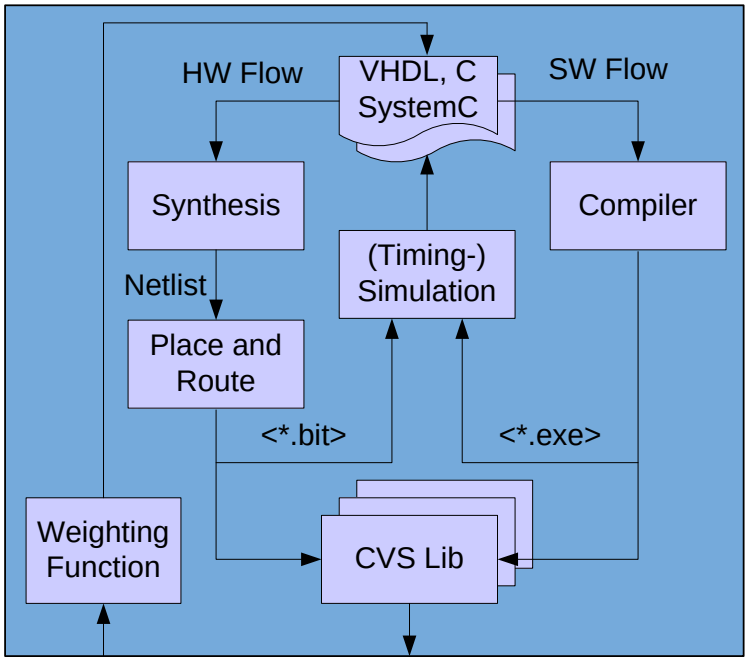
The Design Flow-on-Chip



Integrating a Design Flow on Chip (DFoC)?

DFoC Components

- software flow, e.g., C++
- hardware flow, e.g., VHDL
- IP core library
  - similar to CVS
- feedback loop
  - with weighting function
  - specified vs. current properties
  - stimulates modifications



Outline

Dealing with Complexity

What is complex?

- Functional Spectrum↑
- Design Complexity↑

Dealing with Complexity

What is complex?

- Functional Spectrum↑
- Design Complexity↑
- Hardware, e.g., TMR
  - wiring↑, energy↑, weight↑

How to simplify matters?

- Virtual Hardware
  - similar to virtual memory
  - hardware re-use
- Flexibility instead of Redundancy

Dealing with Complexity

What is complex?

- Functional Spectrum↑
- Design Complexity↑
- Hardware, e.g., TMR
  - wiring↑, energy↑, weight↑
- Software
  - much lines of code
  - written for non-standard hardware

How to simplify matters?

- Virtual Hardware
  - similar to virtual memory
  - hardware re-use
- Flexibility instead of Redundancy
- Object Orientation
- Fixed Development Steps

Outline

Enhanced Autonomy

Autonomy

- = self governance
- reach missions objectives without human intervention
- e.g., DARPA G.C. vehicles

Enhanced Autonomy

Autonomy

- = self governance
- reach missions objectives without human intervention
- e.g., DARPA G.C. vehicles

Autonomicity

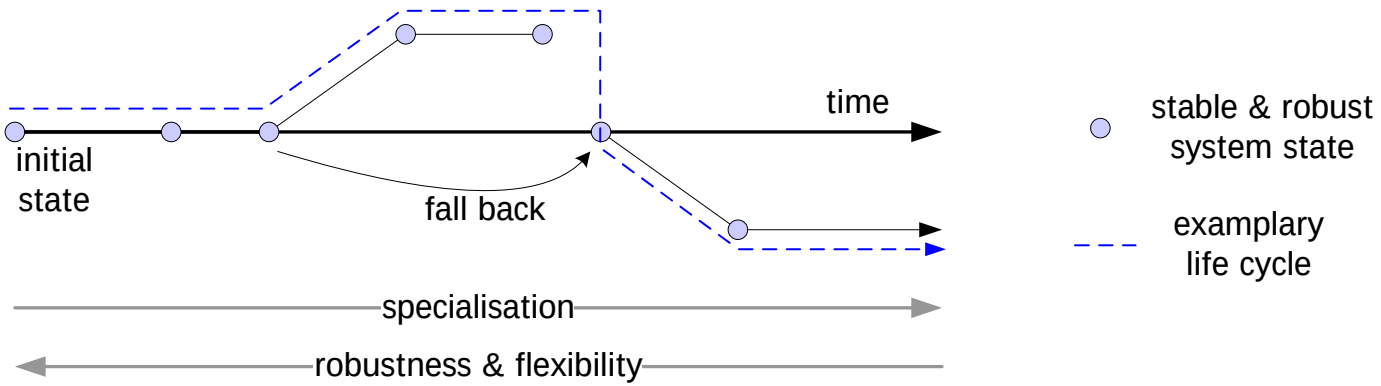
- autonomy + self-x attributes = self management
- goal: improve/not downgrade within functional scope
- e.g., future space missions

Outline

The Progress over Time

Specialisation vs. Flexibility

- specialisation ↑ but flexibility & robustness ↓
- CVS-library
  - every system state/change is minuted
  - branch levels indicate important changes
  - fall back capability in case of failure



Outline

:-(

Technology

- adequate sized NoCs are not realisable on current FPGAs
  - NoC simulated as SystemC model
  - independent from technology

:-(

Technology

- adequate sized NoCs are not realisable on current FPGAs
  - NoC simulated as SystemC model
  - independent from technology

Tools

- tool support still under development
- adequate design flow with small memory footprint needed

:-(

Technology

- adequate sized NoCs are not realisable on current FPGAs
  - NoC simulated as SystemC model
  - independent from technology

Tools

- tool support still under development
- adequate design flow with small memory footprint needed

Non-deterministic Behaviour

- packet based communication is asynchronous!
- autonomy is non-deterministic!
- Does it fit into space missions?

Outline

Summary

Wrap up

- NoC & FPGA based system architecture
- based on approved and well-known principles
- DFoC & PDR = new symbiosis of FPGA & embedded system

We can benefit more from an FPGA's flexibility, even when deployed in the field.

"Wishes" for the (far) Future

- ↑ tool support for PDR
- hard-wired NoCs in FPGAs
- ...
- ~~exploring~~ knowing Mars
- 3D games incl. specific graphic accelerator core

Thank You!