Technische Universität Chemnitz
Fakultät für Informatik
Professur Technische Informatik
Prof. Dr. Wolfram Hardt

HW/SW Codesign II

Interface Synthesis

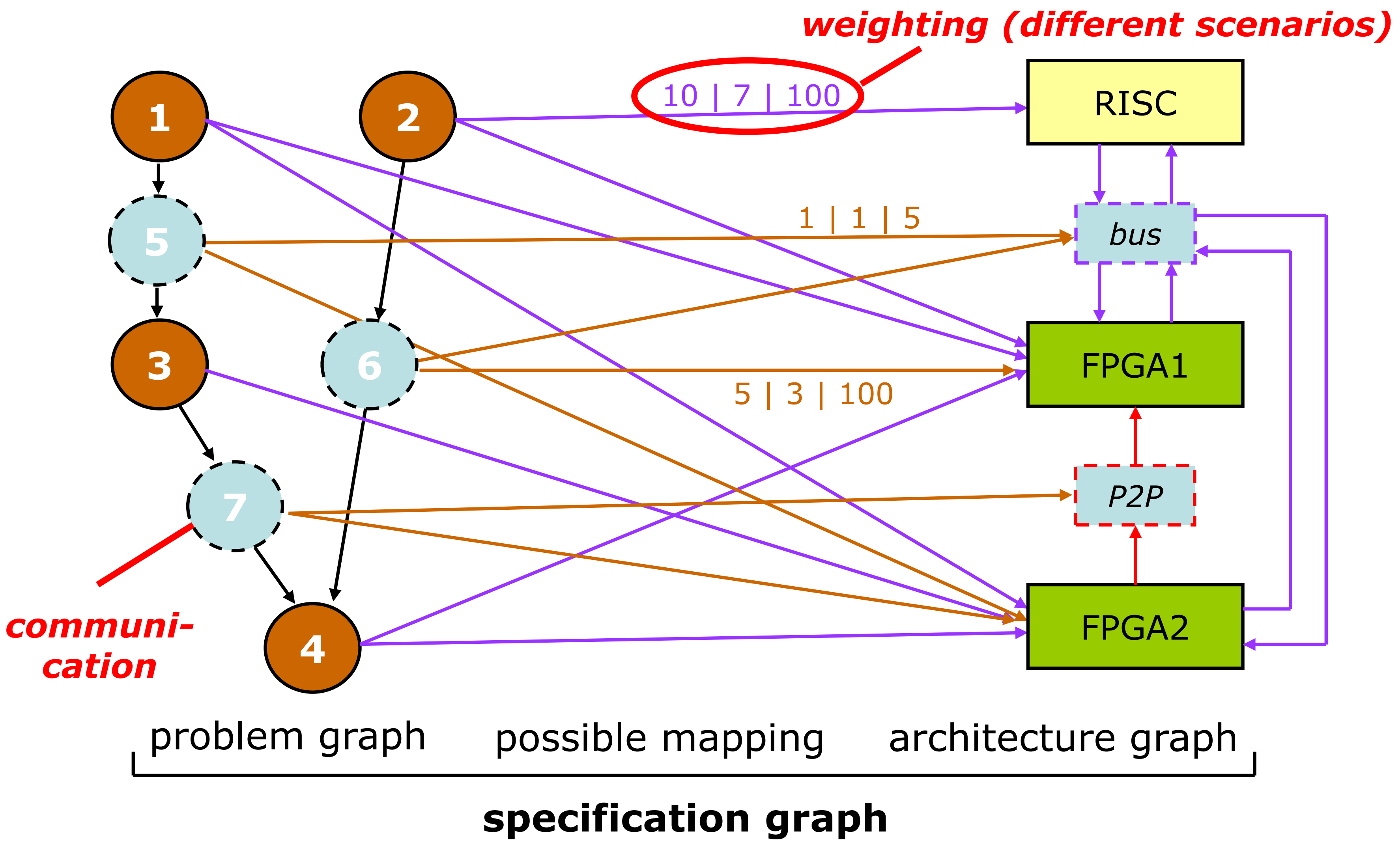
Prof. Dr. Wolfram Hardt
Dipl.-Inf. Michael Nagler

Sommersemester 2015

Contents

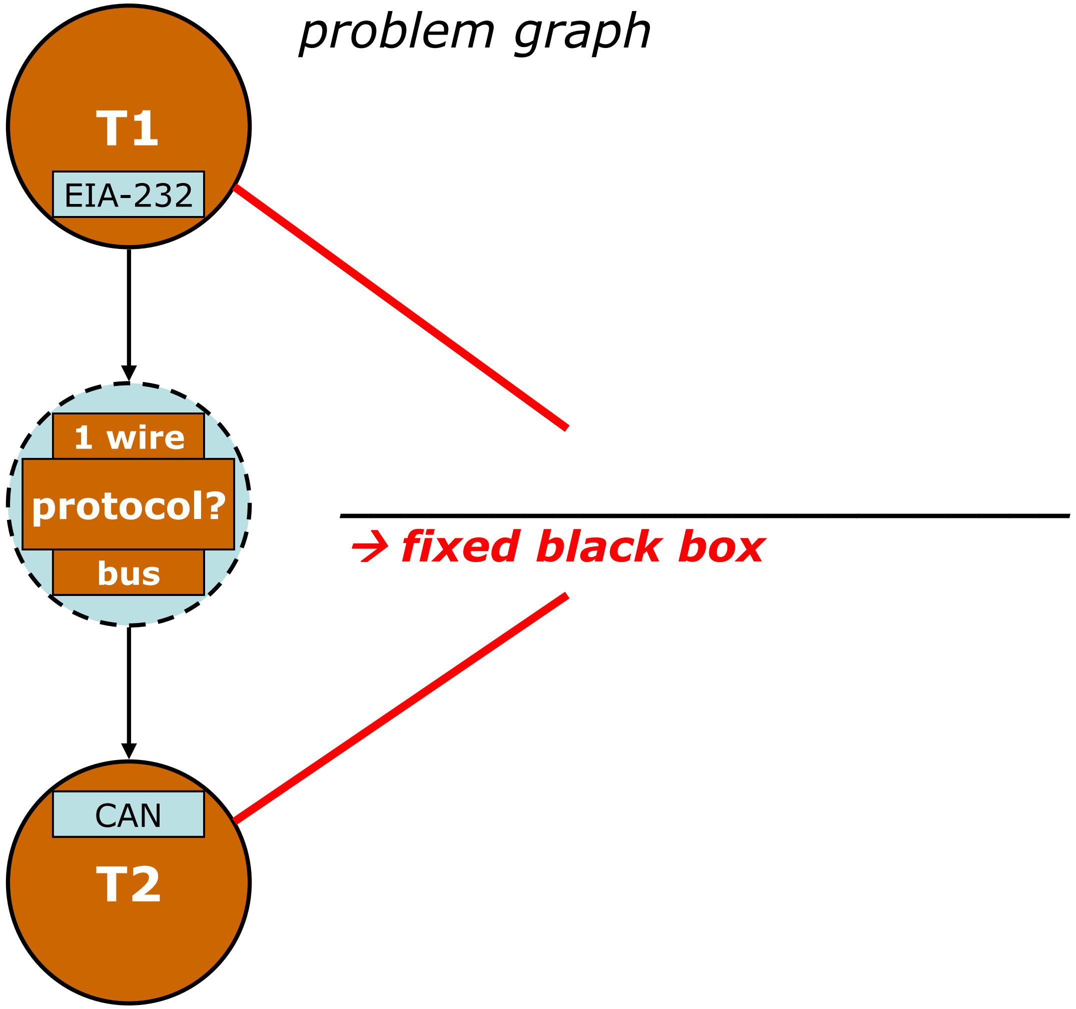
- $\boxed{\text{Introduction}}$
- Interface Block (IFB)

Remember (HSC I)



Example

→ **How to generate the implementation of communication between two tasks with different but known interfaces automatically?**



Interface Incompatibilities

- interfaces may be incompatible by
  - general
    - bus/P2P (e.g. EIA-232 ←→ USB)
  - _______________ definition
    - protocol (e.g. serial, asynchronous: EIA-232 ←→ CAN)
    - bus widths / data types (e.g. serial EIA-232 ←→ parallel PCI)
    - different bandwidth / latencies

always important
→ this lecture

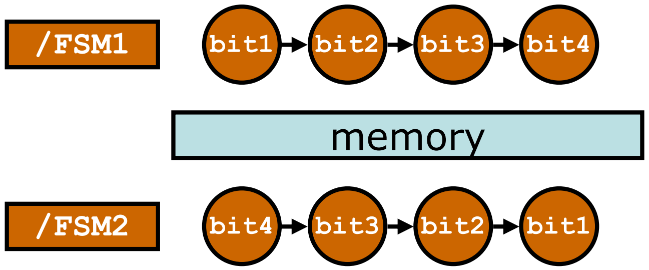
  - _______________ definition
    - different synchronisation (e.g. asynchronous EIA-232 ←→ synchronous I²C)
    - different timings (clock, …)
    - different voltages/currents
  - _______________ definition
    - pin assignment
    - dimensions

important, if tasks mapped to different hardware entities (e.g. FPGA and MC)

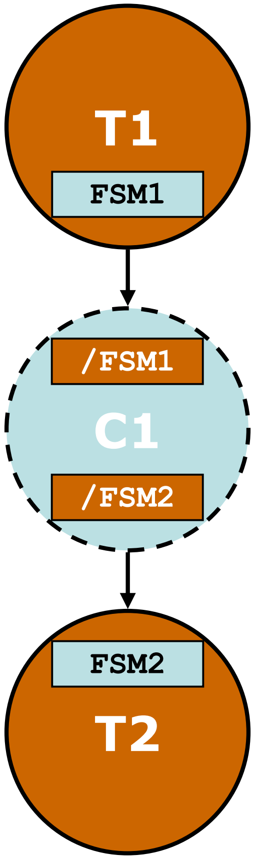
Idea: Finite State Machines

- protocol behaviour defined as finite state machine
  → use outputs of **/FSM1** as inputs for **/FSM2**?

- problem
  - **FSM1** = little endian, **FSM2** = big endian



  - necessary to save bit 1 to 4 before starting **/FSM2**
  - **but:** ______________________________
    ____



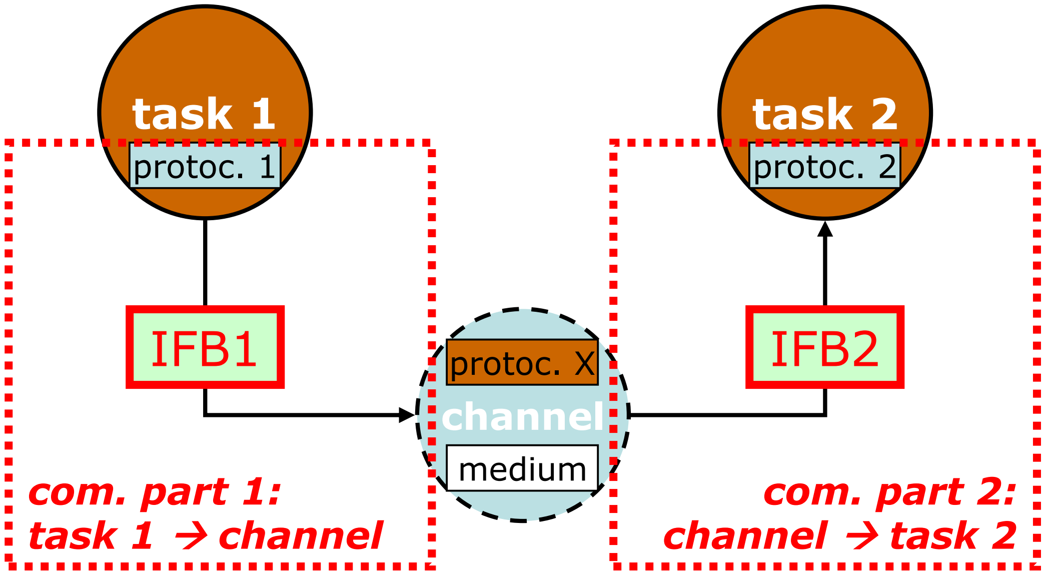
→ **generic and automatically synthesisable** solution preferable

Contents

- Introduction
- $\boxed{\text{Interface Block (IFB)}}$

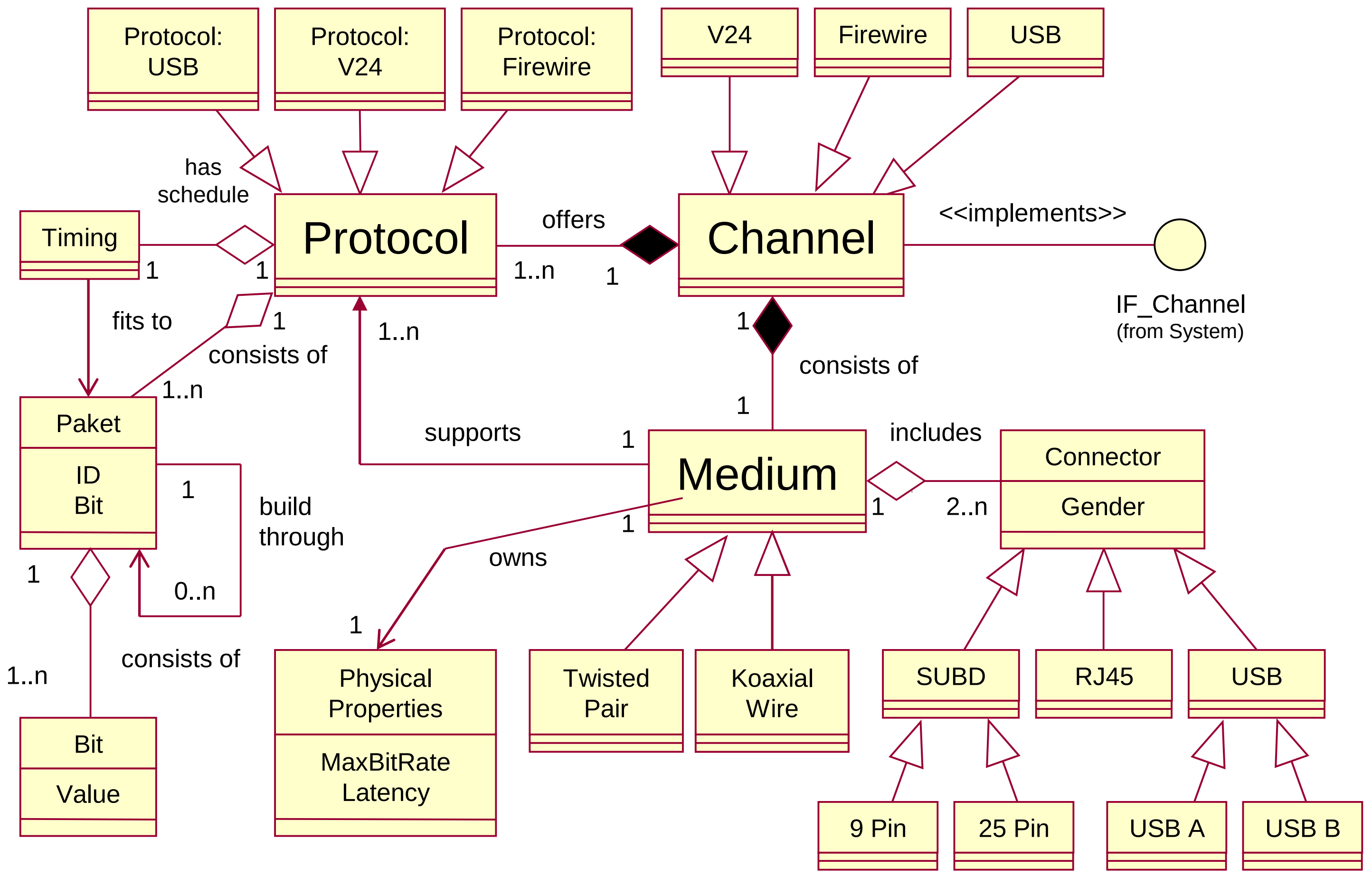
Abstract

- divide communication in two parts

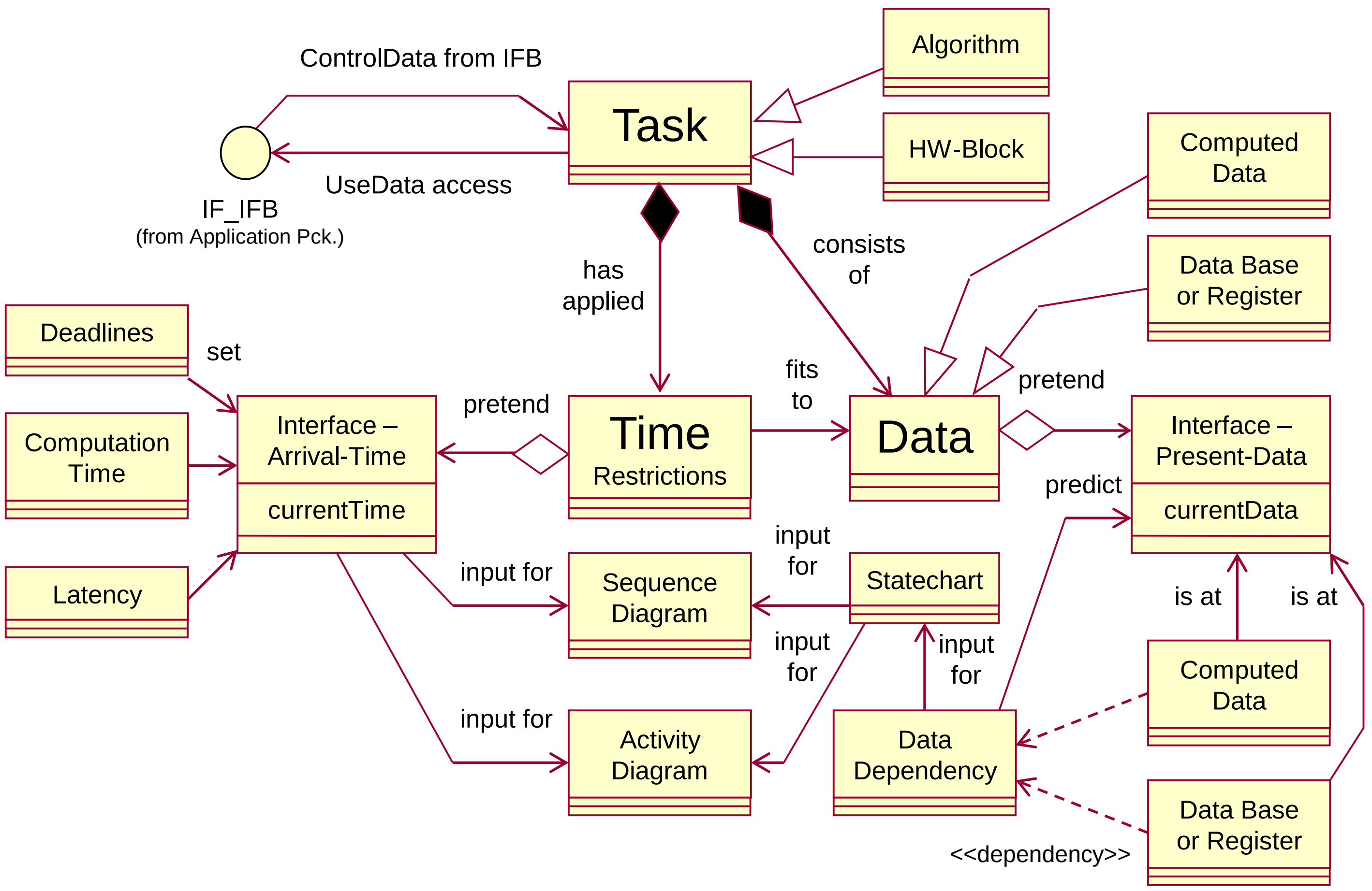


- introduce ________________________________ to translate between
  - protocol 1 (task 1) and protocol X (channel)
  - protocol X (channel) and protocol 2 (task 2)

Channel Modelling by UML



Task Modelling by UML

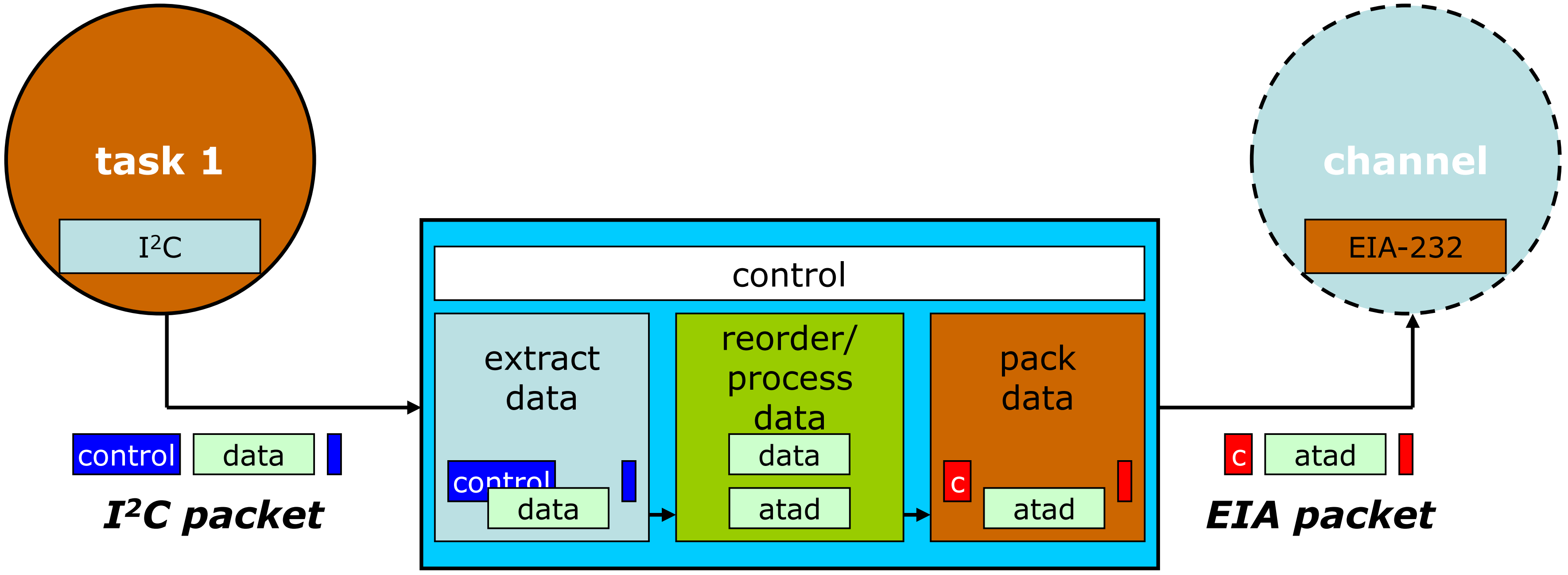


Interface Block – Idea (I)

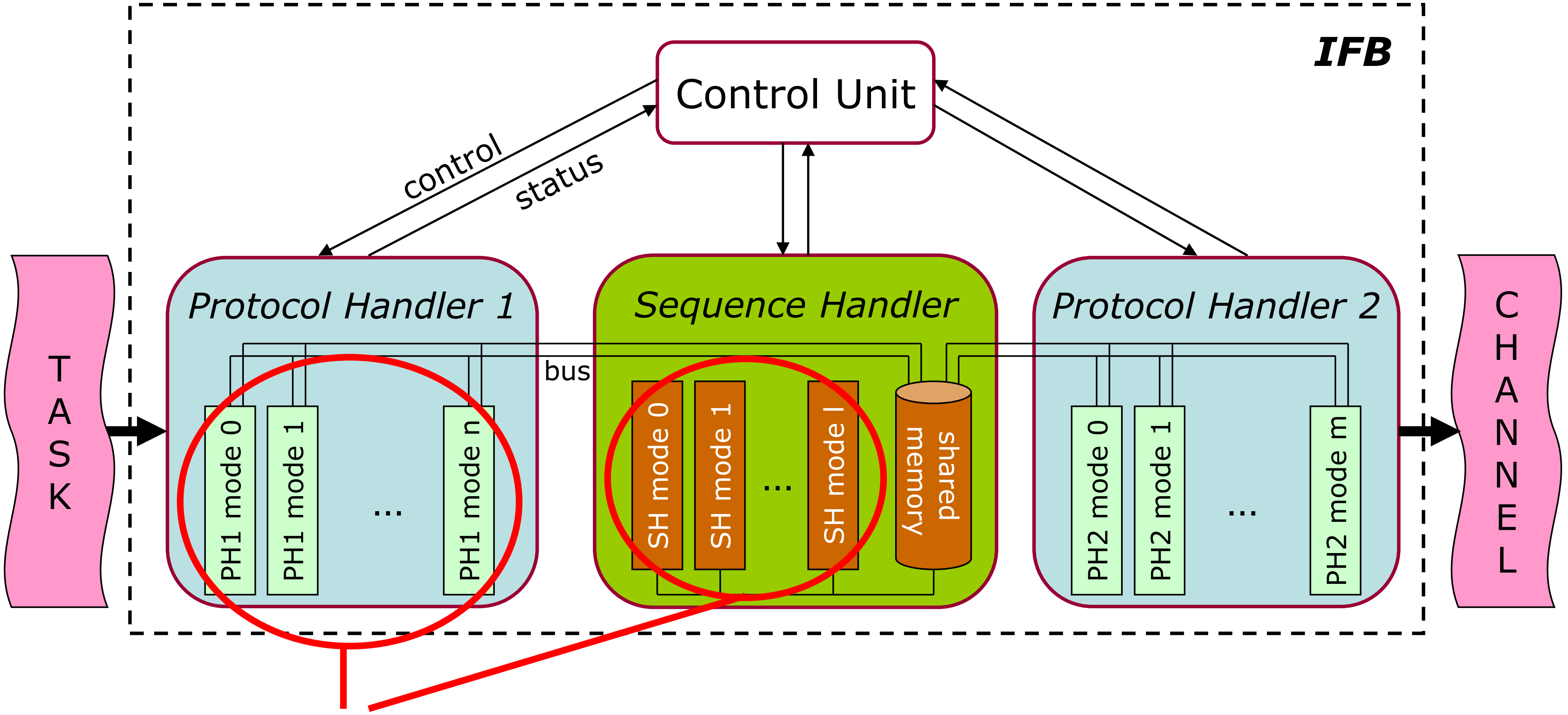
- simplify the problem by using hierarchy
  - no direct protocol conversion
  - protocol: ________________________________
  - data defines the information, not depending on used protocol
  - control information depends on the used protocol
- introduction of transformation sequence (hierarchy)
  - extract data from protocol 1
  - build new data structure
  - generate control information of protocol 2
- definition of transformation sequence (hierarchy)
  - architecture – template

→ important simplification of design method → ________________

Interface Block – Idea (II)



Interface Block – Structural Template



only one mode active at one point of time

Protocol Handler (PH)

- decoding of _______________ protocol
  - check if frame is correct (completeness, timing, CRC, …)
  - remove control data and extract user data
- encoding of _______________ protocol
  - generate control data (address, CRC, …) and put user data in frame
- control of medium accesses
  - allows access to physical interface (task and channel)
  - arbitration method has to be implemented
- transmit user data to Sequence Handler

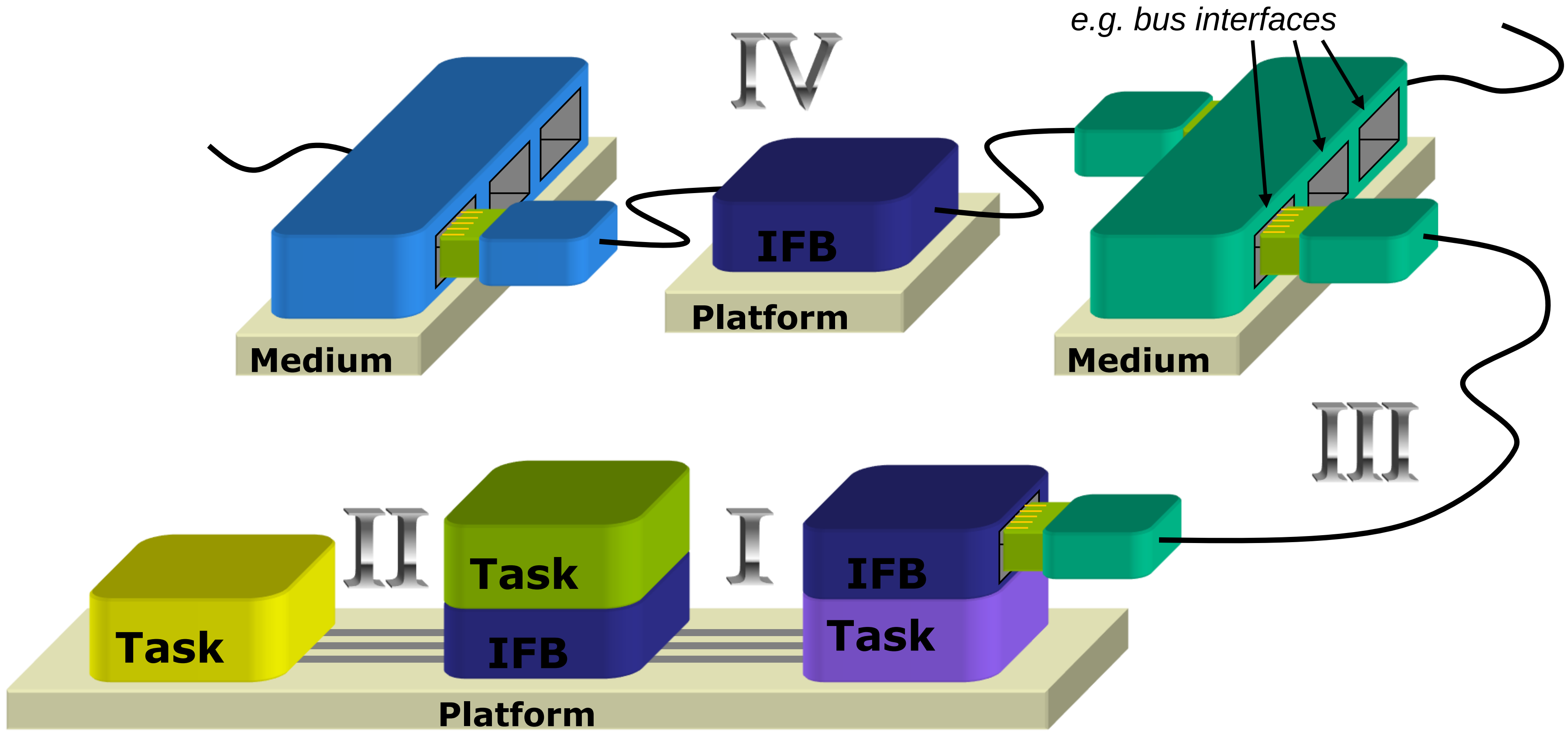
Sequence Handler (SH)

- receive user data from connected PHs
  - store in shared memory
- process data
  - ____________________________________ – if necessary
  - reorder data (big endian ←→ little endian)
  - create new packets/frames (size, order, …)
- select target PH
- send processed data to target PH

Control Unit (CU)

- controls all communication within the IFB
  - control of SH, PH and - if necessary - the task
  - _________________________________ (state machines)
  - feedback by status signals
- control of bus access PH-SH
  - access routines can be controlled directly
  - monitoring, watchdog (passive)
- supports realtime behaviour
  - implementation of time basis
  - realization of global schedule

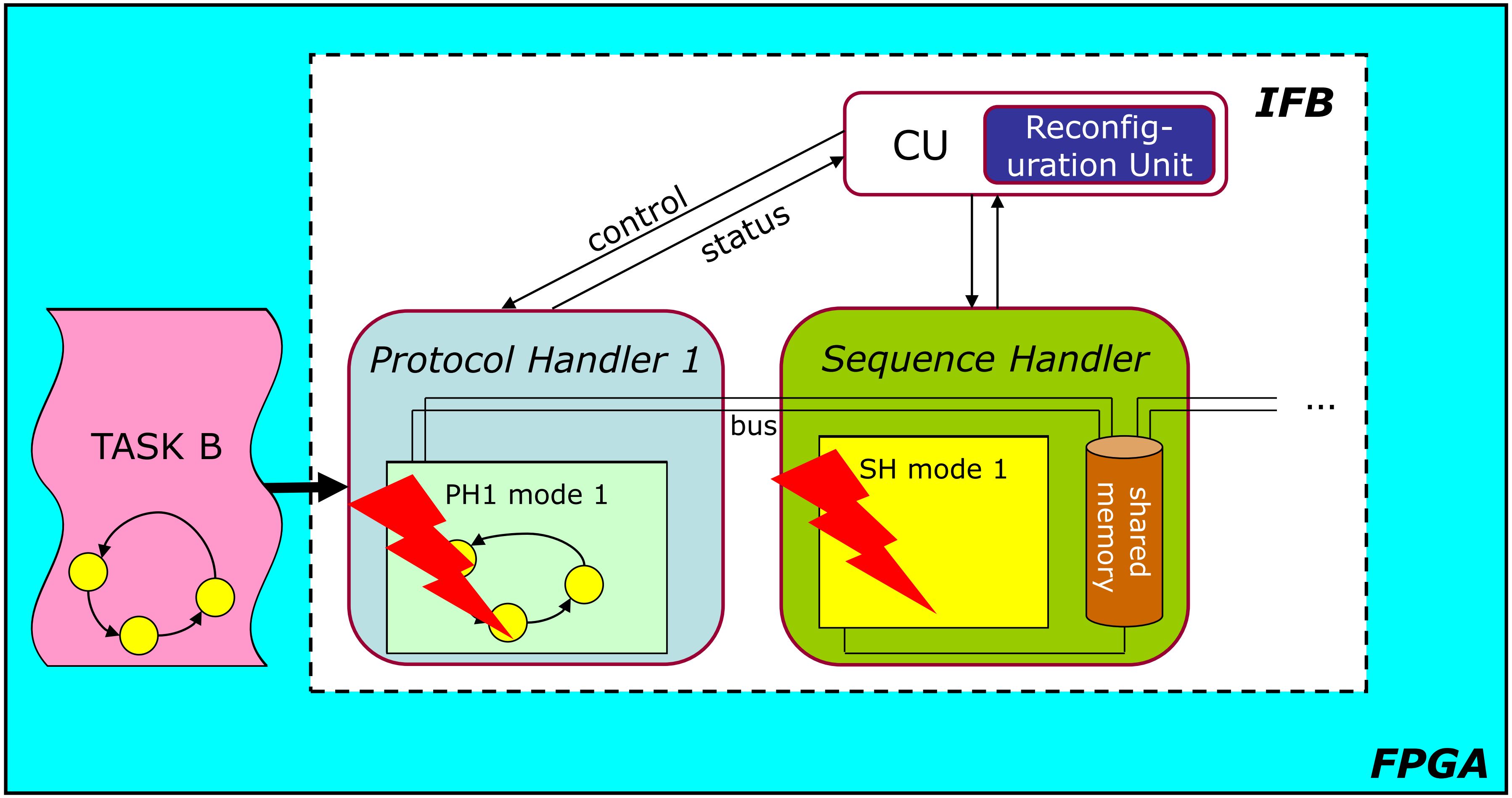
Further Potentials (I)



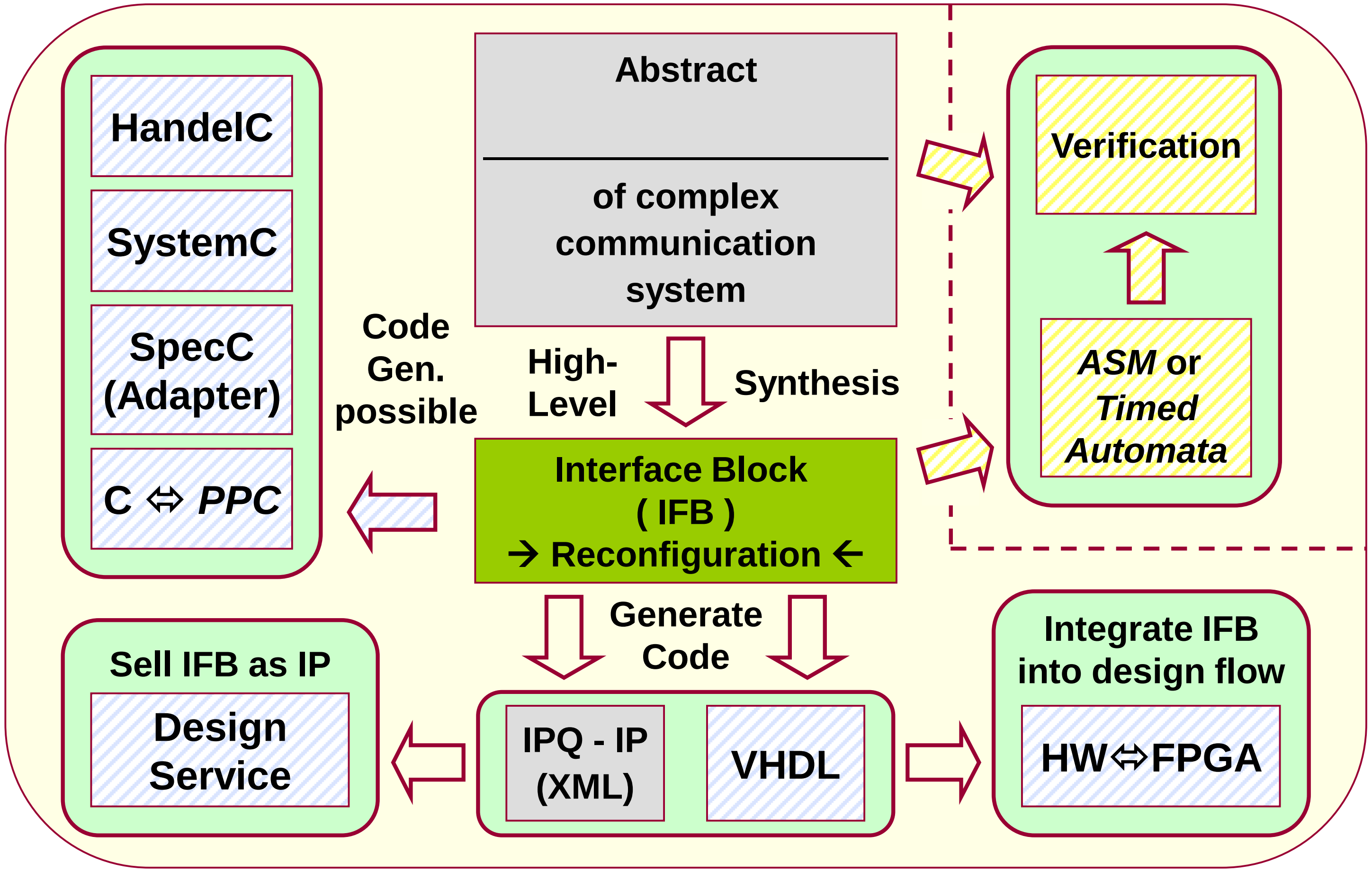
I. task adaptation (without channel)
II. ______________________________
III. task – medium access
IV. media gateway

Further Potentials (II)

- reconfiguration – ___________________________

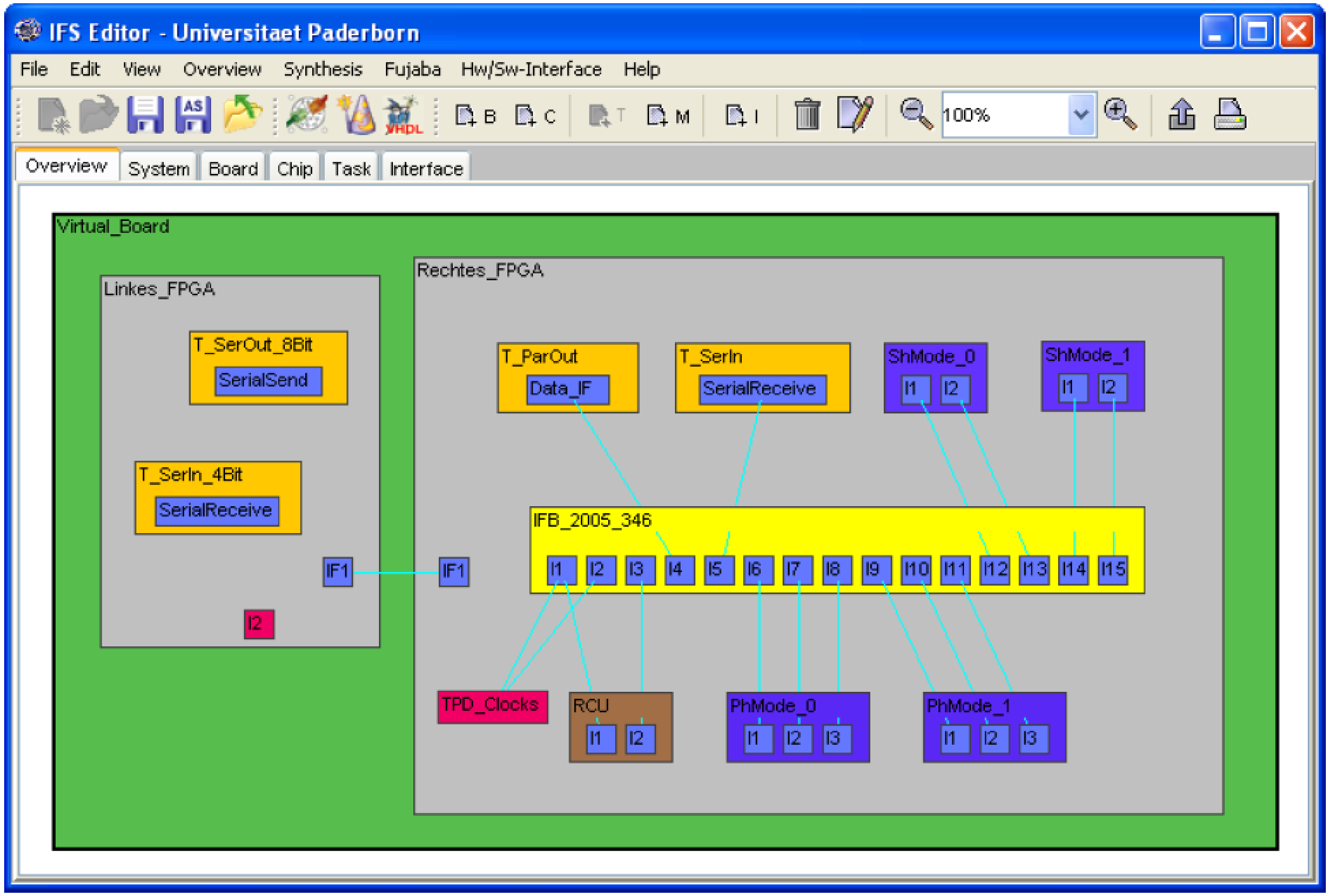


Usage in Design Process



Synthesis Tool: IFS Editor

- software to generate IFB (VHDL)

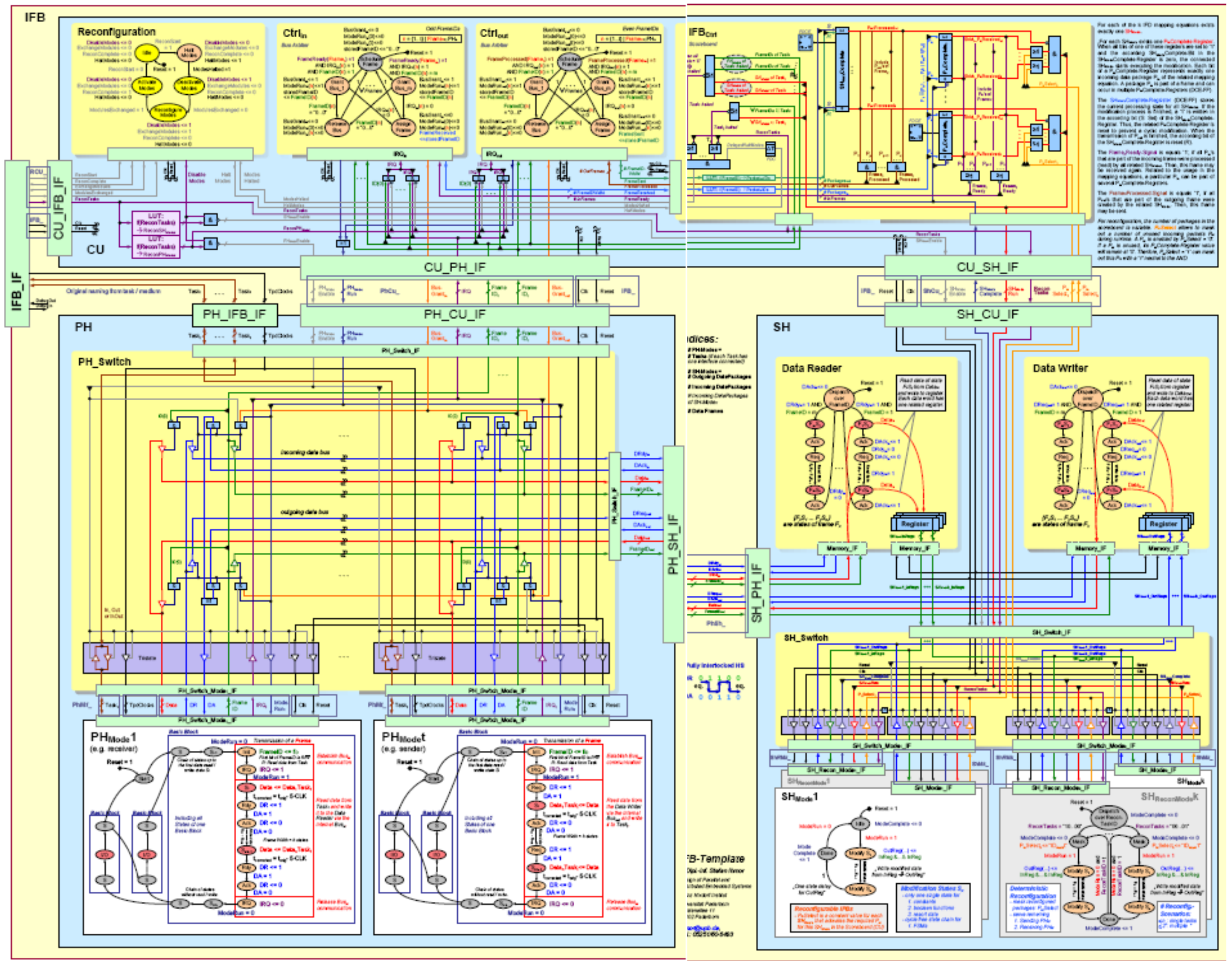


IFS Editor: Synthesis

- input
  - interface description of (at least) two communication partners
    - functional
      - protocol (finite state machine)
      - timing
    - mechanical
      - ports, pins, register
    - target platform description
      - available FPGA resources
      - clock networks, …

- output
  - ______________________________
    - VHDL
    - UCF (constraint file for VHDL synthesis tool)

Result – Very Complex


IFB
Reconfiguration
Ctrl_in
Ctrl_out
IFB_Ctrl
CU
CU_IFB_IF
IFB_IF
CU_PH_IF
CU_SH_IF
PH_IFB_IF
PH_CU_IF
SH_CU_IF
PH
PH_Switch
PH_SH_IF
SH_PH_IF
SH
Data Reader
Data Writer
SH_Switch
PH_Mode 1
PH_Mode t
SH_Mode 1
SH_Recon Mode k
IFB-Template