



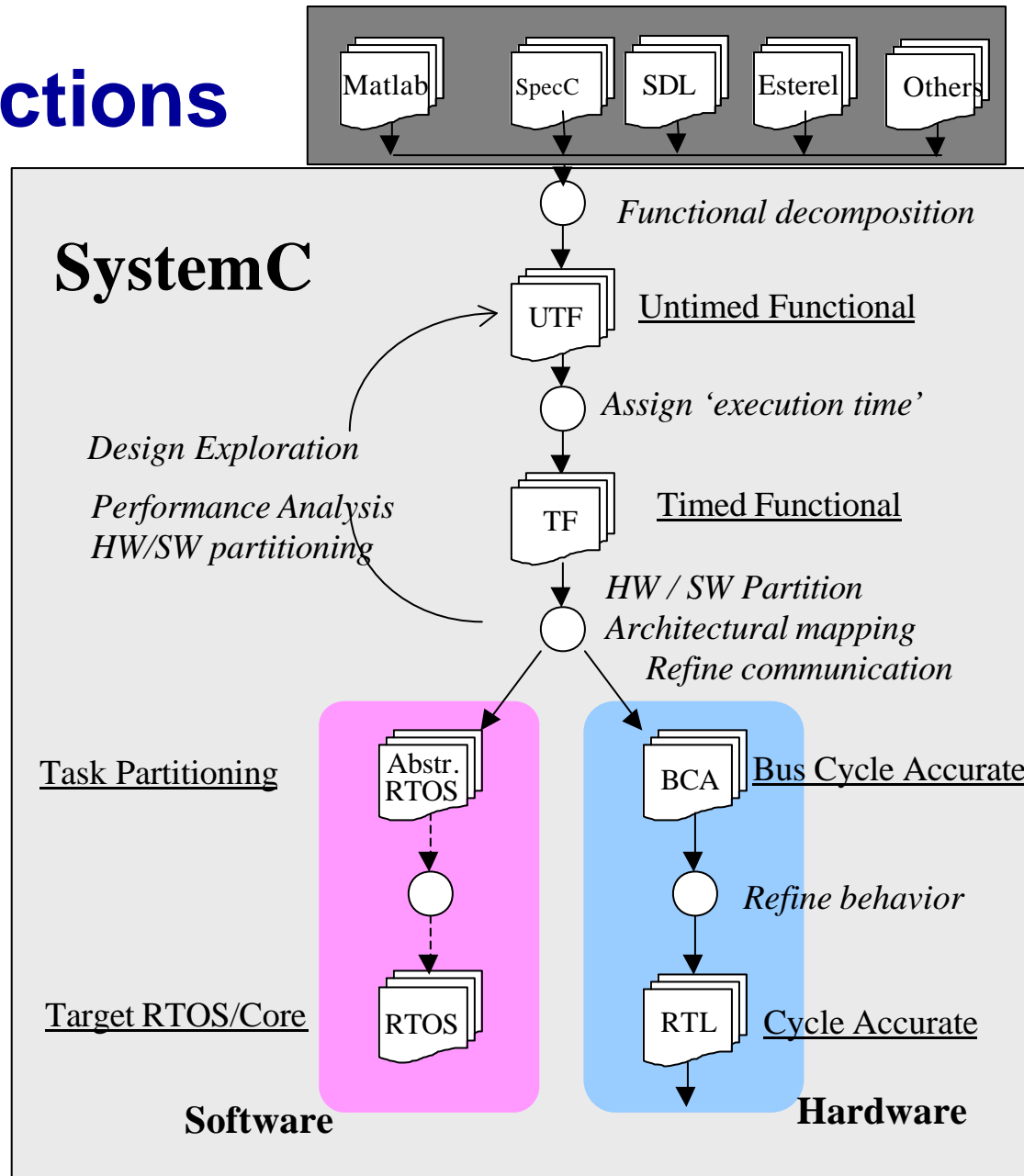
# **System design capabilities in v1.2beta**

**Dündar Dumlugöl  
Director of Engineering, CoWare**

# Master-Slave communication library

- Targets systems with bus protocol communication
  - SoC's with cores, DSP's, peripherals & custom HW
- Enables automatic synthesis of bus interfaces
- Complete path from Functional to RTL

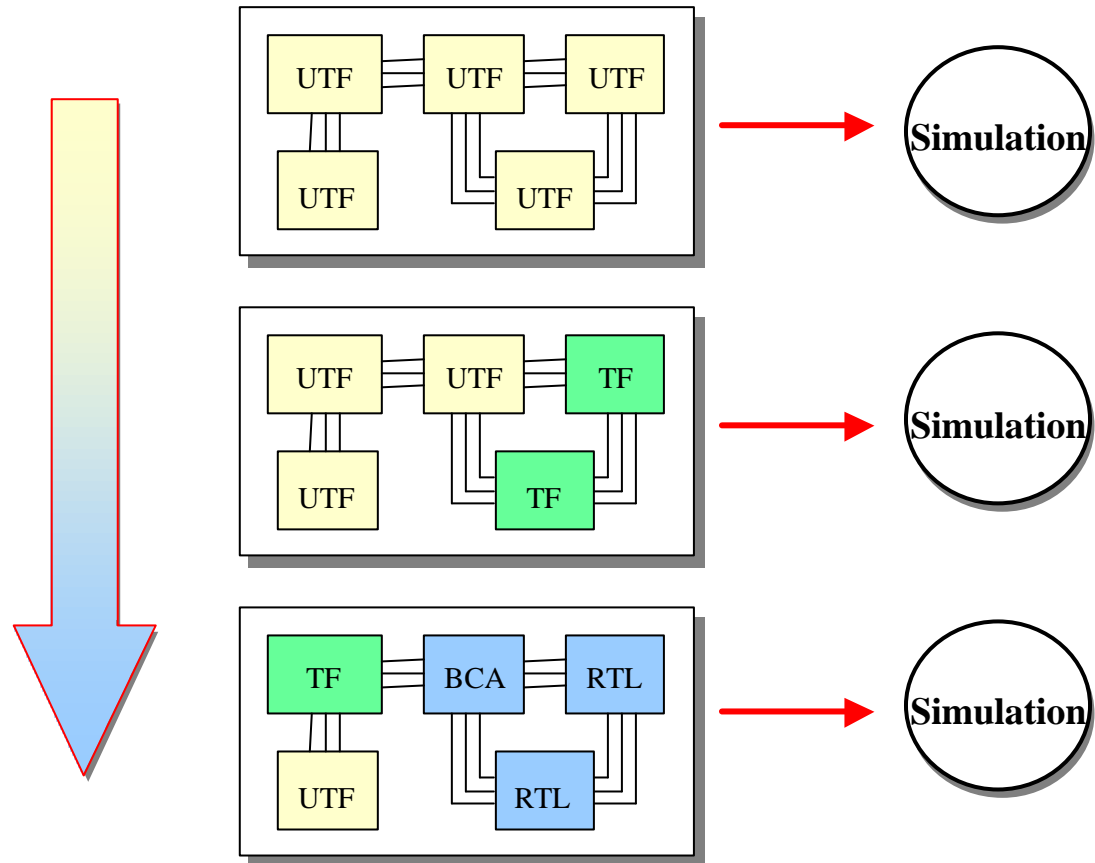
# Abstractions



# Gradual Refinement of the Design

Key to the methodology is that a design may be refined in a gradual step-wise fashion, rather than in one giant step... it need not be "all or nothing".

*Details added to portions of the system.*

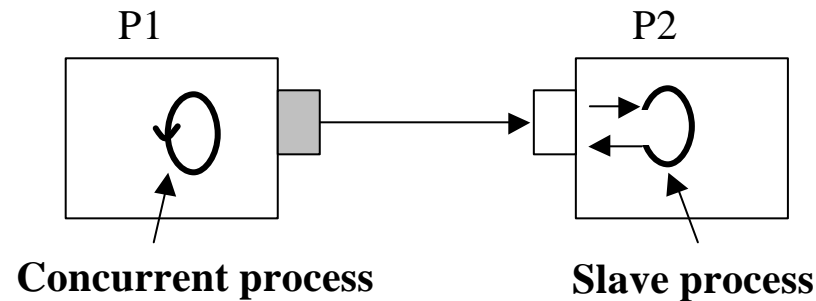


# Key modeling paradigms

- **Abstract functional communication**
  - Point-to-point & multi-point sequential channel
  - Abstract model for bus communication
- **Concurrent communication & synchronization**
- **Cycle accurate bus protocol communication**

# Point-to-Point abstract communication

Equivalent to function call but without function pointer

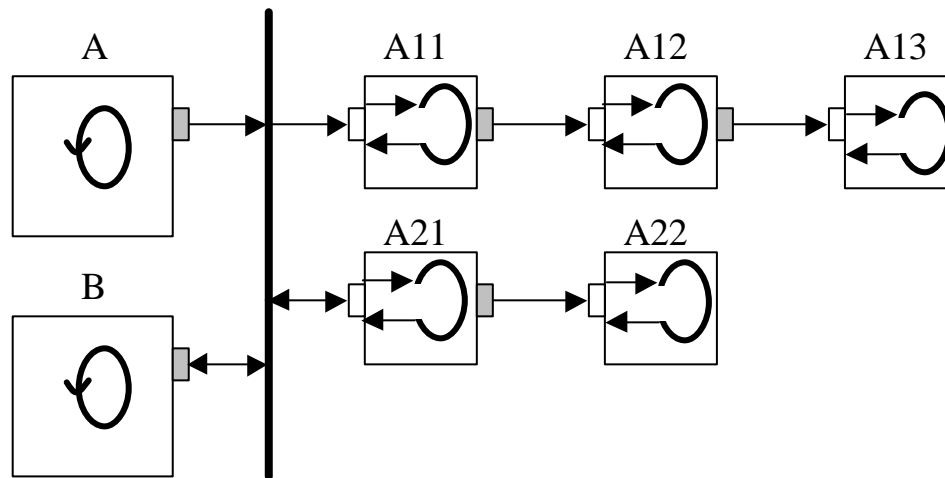


Structure is key for re-use

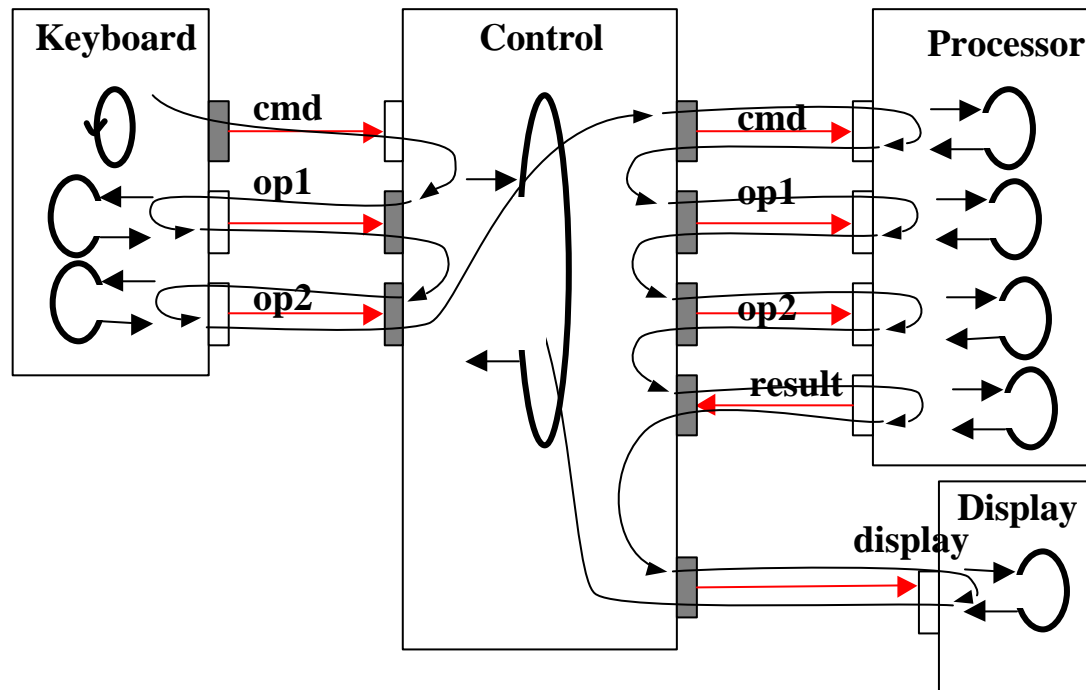
# Specify a communication through ...

- Initiator (master / slave ports)
- Direction of data transfer (in-, out-, inout ports)
- Data type
- Index (=address)
- Channel / transaction configuration
  - Type of communication (e.g.blocking with time out)
  - Return status (pass/ fail ...)
- Bus protocol at the cycle accurate level

# Multi-point abstract communication

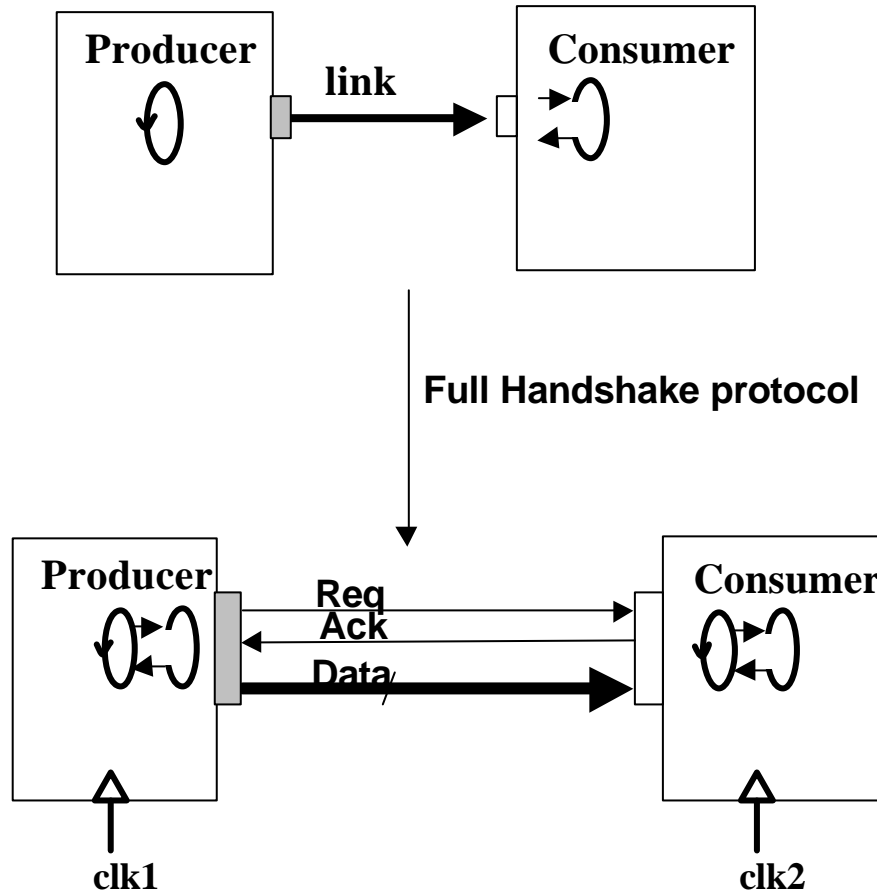


# Example: sequential execution thread in a system





# Example: full-handshake protocol



# Dynamic thread library (prototype)

- For future SW flow
  - Enables fork & join functionality
  - Foundation for future RTOS
- Create thread at run time
- Start another thread with process id
- Stop thread (self)
- Kill another thread
- Get parent id

# Benefits of master-slave library

- HW/SW co-design from Functional to RTL
- Enables automatic synthesis of bus interfaces
- Enables fast IP embedding & HW-SW partitioning
- Orders of magnitude faster than RTL verification