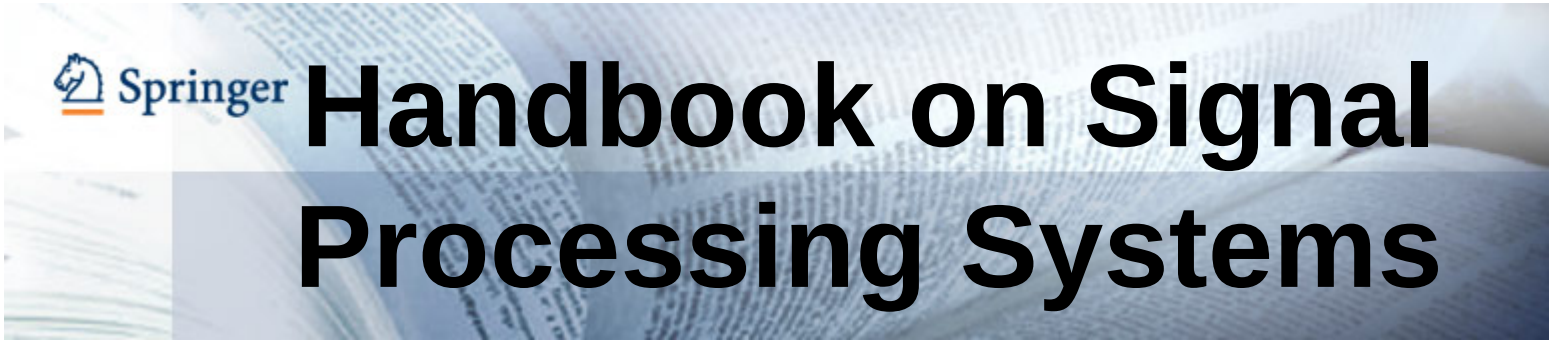


Book chapter on

Intermediate Representations for Simulation and Implementation

To be published 2010

Organization of the book

- The handbook is organized in four parts covering
  - Applications driving the *state-of-the-art* methods for design and implementation
    - Managing editor Shuvra Bhattacharyya
  - HW architectures for implementation
    - Managing editor Jarmo Takala
      - with contr. from O. Gustafsson and L. Wanhammar, D. Lieu
  - Programming, Compilers & Simulation tools
    - Managing editor Rainer Leupers
      - with contr. from C. Kessler, J. Bengtsson
  - Models of computation and associated design tools and methodologies
    - Managing editor Ed Deprettere

Preliminary Book Outline

- **Applications**
  - e.g. Medical Image Processing, OFDM Modems, Low-power WSN Platforms, DSP for Control Systems, Video Compression, ...
- **Architectures**
  - FPGA, Low-power DSP Techniques, Coarse-grained Reconfigurable DSP, GPP DSP, Multicore SOC, Application-specific Accelerators for Communication Systems...
- **Programming and Simulation Tools**
  - C Compilers and Optimization, Compilers for VLIW DSPs, MPSoC Compilers, DSP Instruction Set Simulation, MPSoC Virtualization, *Intermediate Representations for Simulation and Implementation...*
- **Design Methods**
  - Signal-flow and data-flow Graphs, Systolic and Wavefront Arrays for DSP, Dataflow MoC's (SDF, CSDF, PN...), Heterogeneous MoC, Methods and Tools for Mapping Process Networks on MP-Systems...

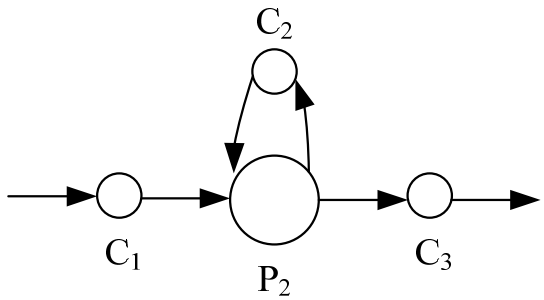
Intermediate Representations for Simulation and Implementation

- To be used for what?
  - Development of simulation, design and implementation tools for (embedded) DSP systems
  - Focus on
    - mapping to parallel processors (Multi/Manycores)
    - different representation levels of non functional properties (timing) and scheduling strategies
- Contents of the book chapter
  - Four representative types of parallel IRs
    - Untimed intermediate representations
      - *System Property Intervals and FunState*
    - Timed intermediate representations
      - *Job Configuration Networks and Timed Configuration Graphs*

System Property Intervals

[F. Cieslok and J. Teich]

- Implemented using Process Networks
- Flat graph representation
- Mixed data and control flow
- Intervals [min, max] are used to specify e.g.
  - I/O data rates (for each channel)
  - computation latency
- Execution mode change by
  - communication of mode tags
  - activation functions
- Representation of scheduling strategies is not possible



Cyclo-static execution of P2

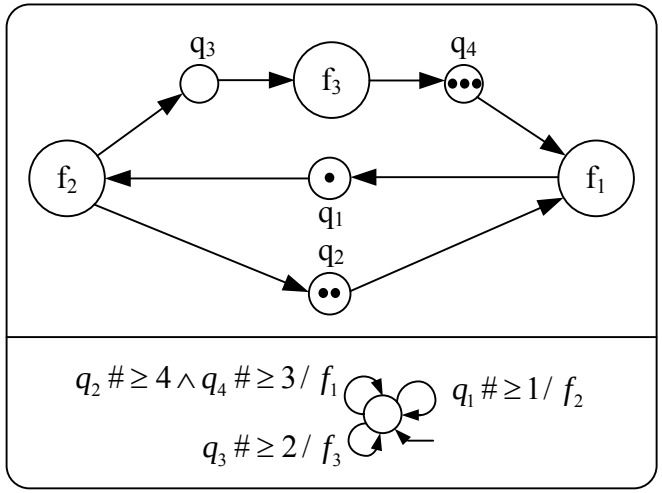
$C_2.tag = 1 \rightarrow C_2.tag = 2$

$C_2.tag = 2 \rightarrow C_2.tag = 1$

FunState

[L. Thiele et al.]

- *Fun*ctions driven by *State* machines
- Dataflow and controlflow is separated
- Hierarchical representations
- Function execution controlled by network global state machines
  - conditions and actions
  - functions are not autonomous
- Complex representation
- Not possible to implement using dataflow MoC



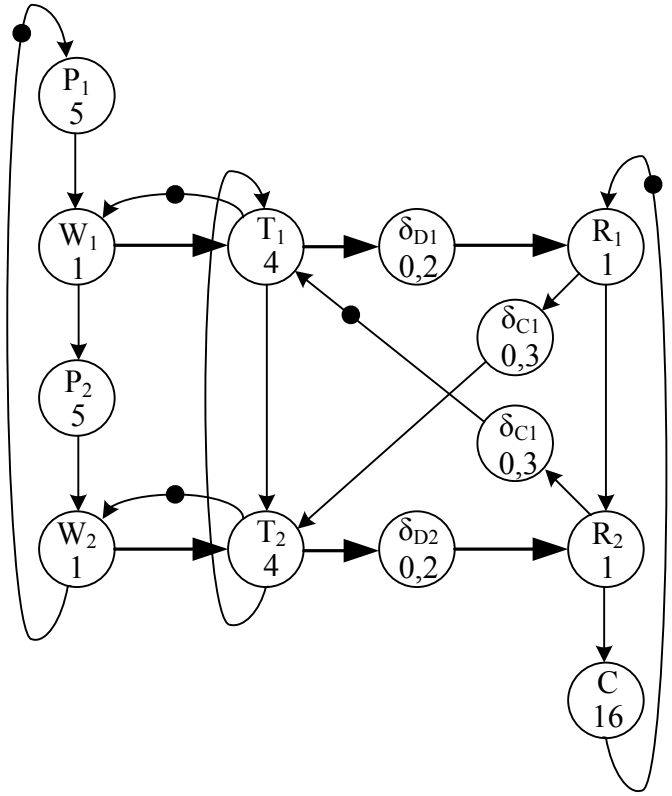
Upper part: a network of functions

Lower part: a state machine (global)

Job Configuration Networks

[P. Poplavko et al.]

- Flat graph representation
- Based on Interprocess Communication Graphs (IPC)
  - Implementable using HSDF
  - amenable for timing analysis
- Input is SDF programs
  - mapping to IR is started from HSDF program
- Expansion to HSDF can result in very large graphs

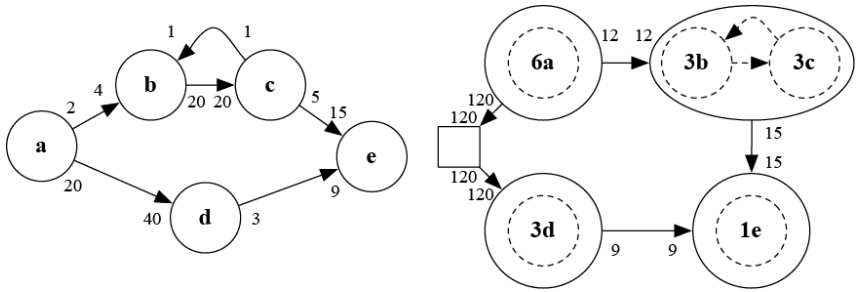


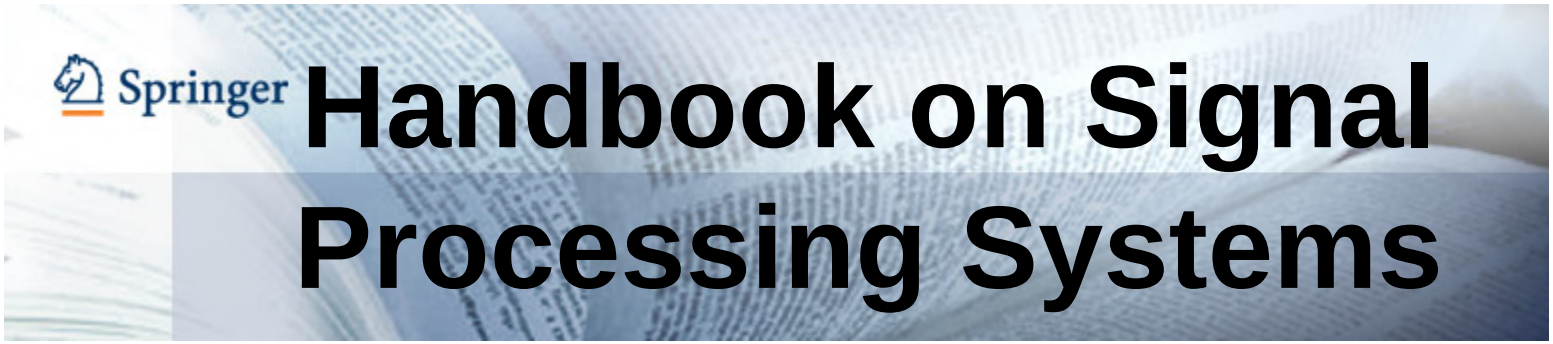
Producer - Consumer example

Timed Configuration Graphs

[J. Bengtsson and B. Svensson]

- Similar to Job Configuration Graphs
- Currently support for
  - repr. of SDF programs
  - constant latencies
  - hierarchical graphs
- Both input program and IR of program is multi-rate SDF
- Coupled to a machine model (manycores)





Book chapter on

Intermediate Representations for Simulation and Implementation

To be published 2010