

SDF3: dataflow analysis and mapping

Sander Stuijk, Marc Geilen, Twan Basten, Bart Theelen

Department of Electrical Engineering Electronic Systems

4 Embedded streaming systems



Application trends Dynamism Concurrency





Predictability

The timing behavior of an application can be guaranteed independent of other applications running in the system







- Model abstract from dynamic behavior
- Many efficient design-time analysis algorithms available
- Low implementation overhead
- SDF-based design approach may lead to resource over-allocation

7 Scenario-aware dataflow



- Dynamic behavior captured in scenarios
- Applications have relatively static behavior inside a scenario
- Trade-off between number of scenarios, run-time analysis techniques, and implementation efficiency

Scenario-aware dataflow – scenario switching



- Similar to SDF, an iteration is a set of actor firings that have no net effect on the token distribution
- Different variants of SADF can model different scenario switching behavior

Scenario-aware dataflow – scenario switching

- FSM-based SADF
 - Scenario executed for complete iteration
 - Each scenario corresponds to an SDF graph
 - FSM specifies possible scenario sequences





11 Analyzing SADF graphs

- Analysis techniques
 - Throughput
 - Latency
 - Buffer requirements
- Techniques based on (max,+)-algebra
- Assumption
 - Relevant implementation aspects must be modeled in the graph
- Example: scenario aware dataflow graph with a static structure
 - Execution times vary with scenarios a and b



¹² SDF³: SDF For Free



- Key features
 - Open-source GPL licensed software
 - Separation of analysis, transformation, and implementation techniques
 - Additional MPSoC platforms can be added with minimal effort

13 Summary

- SADF Model-of-Computation
 - Scenarios capture dynamic (application) behavior
 - Provides many analysis techniques
 - Provides implementation trajectory
- Dataflow graph model captures
 - Application behavior
 - Timing impact of platform resources
- Use of single MoC enables model-based design of predictable systems
- Analysis and implementation techniques implemented in SDF³ tool kit

www.es.ele.tue.nl/sdf3



Hands-on session

Sander Stuijk

Department of Electrical Engineering Electronic Systems

15 Hands-on session

TU/e

Archive with all files be downloaded from

www.es.ele.tue.nl/~sander/tutorials/sdf3-2012/sdf3-tutorial.zip

- USB sticks with files are also available
- Instructions
 - Unpack archive to location of your own choice
 - Open the file index.html in a web browser
 - Select option 'hands-on exercises' from top menu bar



Novel embedded systems, such as smart phones, have to execute multiple streaming applications concurrently. A user may, for example, use a mobile phone to watch a video that is being decoded using an MPEG-4 decoder while an MP3 decoder is used to decode the accompanying audio. The applications may use an Internet connection that requires a software-defined radio protocol to download the required bit streams