Automated Extraction of Scenario Sequences from Disciplined Dataflow Networks

Firew Siyoum¹, Marc Geilen¹, Johan Eker², Carl von Platen³ and Henk Corporaal¹

¹Eindhoven University of Technology, The Netherlands
²Ericsson, Lund, Sweden
³ARM, Lund, Sweden

MEMOCODE 2013, Portland, Oregon
Embedded Streaming Applications

- Modern multimedia and wireless systems are stream processing applications.
- E.g. audio and video encoders/decoders, WLAN IEEE 802.11x, WCDMA and LTE.
Modern multimedia and wireless systems are stream processing applications.

E.g. audio and video encoders/decoders, WLAN IEEE 802.11x, WCDMA and LTE.

Design-time Analysis

Analysing deadlock-freedom and buffer boundedness is crucial.
They also have real-time temporal constraints such as throughput.

Figure: LTE frame structure
Applications are partitioned into a set of tasks that run in parallel on a MPSoC.

**Communicating tasks**

- Tasks are data-dependent and state-dependent.

### Task A
- **InputPort**: i, j
- **OutputPort**: o

- **Firing 1**: `readFromPort(i)`
  - `doSomething()`

- **Firing 2**: `readFromPort(j)`
  - `doAnotherThing()`
  - `writeToPort(o)`

### Task B
- **InputPort**: i
- **OutputPort**: o

- **Firing 1**: `if bitand(i, 1) != 0`
  - `readFromPort(i)`
  - `doSomething()`
  - `writeToPort(o)`

- **Firing 2**: `if bitand(i, 2) != 0`
  - `readFromPort(i)`
  - `doAnotherThing()`
  - `writeToPort(o)`
Synchronous Dataflow (SDF)

- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.

\[ A = \{ x, y, z \} \]
\[ C = \{ c_{xx}, c_{xy}, c_{yz}, c_{zy} \} \]
\[ X(x) = 2, X(y) = 3, X(z) = 2 \]

- Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g. \([x, y, z] = [2, 1, 2] \)
Synchronous Dataflow (SDF)

- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.

\[ A = \{ x, y, z \} \]
\[ C = \{ c_{xx}, c_{xy}, c_{yz}, c_{zy} \} \]
\[ X(x) = 2, \ X(y) = 3, \ X(z) = 2 \]

- Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g. \([x, y, z] = [2, 1, 2]\)

Limitation of SDF

- is not expressive enough to capture dynamic behavior.
Scenario-aware Dataflow (SADF)

- A **scenario** is a single mode of operation and is modeled by a SDF.
- A **FSM** encodes the possible sequences of scenarios.

- A more accurate analysis than SDF. e.g. > 60% in resource saving
Scenario-aware Dataflow (SADF)

- A **scenario** is a single mode of operation and is modeled by a SDF.
- A **FSM** encodes the possible sequences of scenarios.

![Scenario diagrams](image)

- A more accurate analysis than SDF. e.g. > 60% in resource saving

**Limitation of FSM-SADF**

- Model construction is challenged by large number of scenarios.
- Manual construction is time-consuming and error-prone.
We propose

Disciplined Dataflow Network

- is a network of actors that communicate through FIFO connections.
- is defines certain construction rules for parallel dataflow programs.
We propose

Disciplined Dataflow Network

- is a network of actors that communicate through FIFO connections.
- is defines certain construction rules for parallel dataflow programs.

Objectives of DDN

1. automatically test if a dataflow program complies with these rules
2. automatically extract a FSM-based SADF model if it complies
We propose

Disciplined Dataflow Network

▶ defines rules that restrict state and data dependencies of actors.
▶ is constructed from two types of actors: detectors and kernels.

Detectors are controllers and kernels are followers.
▶ A detector can control multiple kernels.
▶ A kernel is controlled by at most one detector.
▶ A detector and the set of kernels it controls form a SSR.
Kernel Actor

- has a set of firings and each firing has a **DDN firing rule**.
  - DDN firing rules are integer constraints.
  - DDN firing rules are **complete**.
- Completeness guarantees termination-free execution.

```plaintext
actor ExampleKernel()
    InputPort i, j
    OutputPort o

    Firing 1: if (i > 2 and j < 5) //firing rule 1
doSomething ...

    Firing 2: if (i <= 2 and 5 <= j < 10) //firing rule 2
doAnotherThing ...

    Firing 3: if (j >= 10) //firing rule 3
doNothing ...

end
```
Firing rules do not have to be DDN firing rules.
Output tokens are of type integer and take values from a finite set.
Each firing produces a set of control tokens.

```java
actor ExampleDetector()
    InputPort i,
    OutputPort o

    Firing 1: if foo(i) > 0 // produce control token 0
doSomething ...
    writeToOutputPort(0)

    Firing 2: if i > 0 // produce control tokens 1 to 5
doAnotherThing ...
x := pick from 1 to 5
    writeToOutputPort(x)
end
```
Scenario configuration

- A single firing of a detector defines a scenario of the SSR.
- Thus, a tuple of firings, one from each detector of the DDN, determines a scenario of the whole network.
- such a tuple is called a scenario configuration.
Scenario Sequence Extraction

Configuration-space Exploration

- Extracting configurations using the Cartesian product is pessimistic.
- We extract scenario configurations more accurately from local FSMs of detectors, through configuration-space exploration.
Extracting configurations using the Cartesian product is pessimistic.

We extract scenario configurations more accurately from local FSMs of detectors, through configuration-space exploration.

Complete Space
Scenario Sequence Extraction

Configuration-space Exploration

- Extracting configurations using the Cartesian product is pessimistic.
- We extract scenario configurations more accurately from local FSMs of detectors, through configuration-space exploration.

Complete Space

Simplified Space
Extraction framework

- The approach is demonstrated for the CAL actor language.
- CAL is employed by the ISO/IEC standardization for the Reconfigurable Video Coding (RVC) MPEG standard.
WLAN 802.11a Baseband Processing

Extracted WLAN Scenario

Extracted FSM

Psi(q0) = s1
Psi(q1) = s2
Psi(q2) = s3
Psi(q3) = s4
RVC-MPEG4 SP Video Decoder
Case-study

RVC-MPEG4 SP Video Decoder

Extracted Scenario
Case-study

Analysis

- Deadlock-freedom and boundedness: each scenario sequence
- Timing Analysis: we mapped the extracted scenarios onto a multi-core platform model and analysed the worst-case throughputs.
Disciplined Dataflow Networks

- is an analysable programming model for dynamic streaming applications.
  1. guarantees a scenario-based analysis model can always be constructed.
  2. guarantees the process of constructing the model can be automated.
Conclusions

Disciplined Dataflow Networks

- is an analysable programming model for dynamic streaming applications.
  1. guarantees a scenario-based analysis model can always be constructed.
  2. guarantees the process of constructing the model can be automated.

- Caltoopia - http://www.github.com/caltoopia
- SDF3 - http://www.es.ele.tue.nl/sdf3
- Analysis - Worst-case throughput analysis of dynamic streaming applications, in proc. CODES+ISSS 2012
Disciplined Dataflow Networks

- is an analysable programming model for dynamic streaming applications.
  1. guarantees a scenario-based analysis model can always be constructed.
  2. guarantees the process of constructing the model can be automated.

- Caltoopia - http://www.github.com/caltoopia
- SDF3 - http://www.es.ele.tue.nl/sdf3
- Analysis - Worst-case throughput analysis of dynamic streaming applications, in proc. CODES+ISSS 2012

Thank you!

Questions?