Mapping Software-Defined Radio Applications onto MPSoCs

Firew Siyoum, Sander Stuijk, Marc Geilen
Department of Electrical Engineering, Eindhoven University of Technology
Map2MPSoC 2011
Radio

- Modern embedded devices support multiple wireless communication standards.
- We refer to these standards as radios.
- Smartphones, for e.g., include various radios, such as WCDMA, LTE and IEEE 802.11x.
Radio

- Modern embedded devices support multiple wireless communication standards.
- We refer to these standards as radios.
- Smartphones, for e.g., include various radios, such as WCDMA, LTE and IEEE 802.11x.

Radio Baseband Processing

- The three main stages are filters, modem and codec.
- These stages are customarily implemented as hardware blocks.

\[
\text{RF/IF} \quad \text{Filters} \quad \text{Modem} \quad \text{Codec (De)Mux} \quad \text{Higher layers} \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
\text{Control}
\]
Background

Software Defined Radio (SDR)

- Radio design is shifting from dedicated hardware blocks to software processes for better flexibility and cost efficiency.
- SDR is a radio implemented as software processes that run on a MPSoC.
Background

Software Defined Radio (SDR)

▶ Radio design is shifting from dedicated hardware blocks to software processes for better flexibility and cost efficiency.

▶ SDR is a radio implemented as software processes that run on a MPSoC.

MPSoC architectures for SDR

▶ combine homogeneous and heterogeneous multiprocessing, including GPPs, vector processors and weakly programmable accelerators.
SDR design

- Requirements:
  - satisfying temporal requirements such as latency and throughput.
  - staying within limited resource budgets, such as battery power.
SDR design

Requirements:
- satisfying temporal requirements such as latency and throughput.
- staying within limited resource budgets, such as battery power.

Challenges include:
- dynamism: data-dependent workload and system variability.
- high workload: e.g. in smartphones, a digital workload of 100GOPS within 1W power budget.
SDR design

- Requirements:
  - satisfying **temporal requirements** such as latency and throughput.
  - staying within **limited resource budgets**, such as battery power.

- Challenges include
  - **dynamism**: data-dependent workload and system variability.
  - **high workload**: e.g. in smartphones, a digital workload of 100GOPS within 1W power budget.

- One of our solutions is **variation-aware dataflow-based design flow**.
Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
  - **modeling** techniques for radios, storage, arbitration, etc
Background

Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
  - **modeling** techniques for radios, storage, arbitration, etc
  - **scheduling** techniques for scarce resources such as power and memory

Modeling

Mapping/scheduling
Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
  - **modeling** techniques for radios, storage, arbitration, etc
  - **scheduling** techniques for scarce resources such as power and memory
  - **analysis** techniques to compute buffer-sizing, latency and throughput.
Outline

What Synchronous Dataflow Scenarios are and how to analyse them

How to model radios using scenarios dynamism in Long Term Evolution (LTE) and how scenarios capture dynamism in LTE
A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.

Firing of an actor = starting execution

Replication vector: the number of rings of each actor that brings the graph back to its original state. e.g. for the above SDF, \( \bar{R} = [2, 1, 2] \)

Iteration is a set of actor rings, as specified by the replication vector.
- an iteration is marked by the production times of initial tokens, that is recorded in a time-stamp vector \( \gamma \).
Synchronous dataflow (SDF)

A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.

- Firing of an actor = starting execution
Synchronous dataflow (SDF)

- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.

- Firing of an actor = starting execution
- Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g for the above SDF, $\bar{R} = [2, 1, 2]$
Synchronous dataflow (SDF)

SDF

- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.

Firing of an actor = starting execution

Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g. for the above SDF, \( \bar{R} = [2, 1, 2] \)

Iteration

- is a set of actor firings, as specified by the repetition vector.
- an iteration is marked by the production times of initial tokens, that is recorded in a time-stamp vector \( \gamma \).
Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.
Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.
- For a scenario, behavior is mostly invariable and modeled with SDF.
Scenario-aware Dataflow (SADF)

▶ A scenario - a single mode of operation of an application.
▶ For a scenario, behavior is mostly invariable and modeled with SDF.
▶ Possible transitions can be modeled by a finite-state machine (FSM).
Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.
- For a scenario, behavior is mostly invariable and modeled with SDF.
- Possible transitions can be modeled by a finite-state machine (FSM).

FSM-SADF

- is a tuple $F = (S, f)$, consisting of a set of scenarios $S$ and a FSM $f$. 
Analysing FSM-SADF - single scenario

- Every SDF has a matrix $M$, whose dimension is $|\gamma| \times |\gamma|$.
Every SDF has a matrix $M$, whose dimension is $|\gamma| \times |\gamma|$.

\[
M = \begin{bmatrix}
2 & 3 & 2 \\
5 & 6 & 5 \\
7 & 8 & 7
\end{bmatrix}, \quad i_{xy} = \text{distance b/n tokens } y \text{ & } x \text{ tokens}
\]
Analysing FSM-SADF - single scenario

- Every SDF has a matrix $M$, whose dimension is $|\gamma| \times |\gamma|$.

  \[
  M = \begin{bmatrix}
  2 & 3 & 2 \\
  5 & 6 & 5 \\
  7 & 8 & 7
  \end{bmatrix},
  \]

  $i_{xy} =$ distance b/n tokens $y$ & $x$ tokens

- Any two consecutive iterations are related by a matrix multiplication
  in $(max, +)$ algebra, i.e. $\gamma_{k+1} = M \cdot \gamma_k$. 

\[
M = \begin{bmatrix}
  2 & 3 & 2 \\
  5 & 6 & 5 \\
  7 & 8 & 7
  \end{bmatrix},
  \]

$i_{xy} =$ distance b/n tokens $y$ & $x$ tokens
Analysing FSM-SADF - single scenario

- Every SDF has a matrix $M$, whose dimension is $|\gamma| \times |\gamma|$.

\[ M = \begin{bmatrix} 2 & 3 & 2 \\ 5 & 6 & 5 \\ 7 & 8 & 7 \end{bmatrix}, \quad i_{xy} = \text{distance b/n tokens } y \text{ & } x \text{ tokens} \]

- Any two consecutive iterations are related by a matrix multiplication in $(\text{max}, +)$ algebra, i.e. $\gamma_{k+1} = M \cdot \gamma_k$.

- For the above graph, $\gamma$ for the first 3 iterations are

\[ \gamma_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \gamma_1 = \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix}, \quad \gamma_2 = \begin{bmatrix} 10 \\ 13 \\ 15 \end{bmatrix}, \quad \gamma_3 = \begin{bmatrix} 17 \\ 20 \\ 22 \end{bmatrix} \]
Analysing FSM-SADF - single scenario

- Every SDF has a matrix $M$, whose dimension is $|\gamma| \times |\gamma|$.

\[
M = \begin{bmatrix}
2 & 3 & 2 \\
5 & 6 & 5 \\
7 & 8 & 7
\end{bmatrix}, \ i_{xy} = \text{distance b/n tokens y & x tokens}
\]

- Any two consecutive iterations are related by a matrix multiplication in $(\max, +)$ algebra, i.e. $\gamma_{k+1} = M \cdot \gamma_k$.

- For the above graph, $\gamma$ for the first 3 iterations are

\[
\gamma_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ \gamma_1 = \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix}, \ \gamma_2 = \begin{bmatrix} 10 \\ 13 \\ 15 \end{bmatrix}, \ \gamma_3 = \begin{bmatrix} 17 \\ 20 \\ 22 \end{bmatrix}
\]

- Period = 7 time-units & hence, throughput = $\frac{1}{7}$ iters. per time-unit.
Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

Conservative methods†

- linear upper-bounds to $\gamma_k$

* M. Geilen and S. Stuijk, CODES+ISSS, 2010
† M. Geilen, ACM tran. on Emb. Comp. Sys. 2010
Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

Conservative methods†

- linear upper-bounds to $\gamma_k$

- find a vector $\bar{\tau}$ and a scalar number $\lambda$ such that $\gamma_k \leq \bar{\tau} + k \cdot \lambda$

* M. Geilen and S. Stuijk, CODES+ISSS, 2010
† M. Geilen, ACM tran. on Emb. Comp. Sys. 2010
Analyzing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

Conservative methods†

- linear upper-bounds to $\gamma_k$
- find a vector $\bar{\tau}$ and a scalar number $\lambda$ such that $\gamma_k \leq \bar{\tau} + k \cdot \lambda$
- **Scenario graph**: $\max_{s \in S} \tau_s + \lambda_s$

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
† M.Geilen, ACM tran. on Emb. Comp. Sys. 2010
Analyzing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

Conservative methods†

- linear upper-bounds to $\gamma_k$
  
- find a vector $\bar{\tau}$ and a scalar number $\lambda$ such that $\gamma_k \leq \bar{\tau} + k \cdot \lambda$
  
- Scenario graph: $\max_{s \in S} \tau_s + \lambda_s$
  
- Reference schedule: MCM of the FSM where the weight of each node is $\tau_s + \lambda_s$.

* M.Geilen and S. Stuijk, CODES+ISSS, 2010

† M.Geilen, ACM tran. on Emb. Comp. Sys. 2010
LTE is a recent standard in cellular wireless communication.
Long Term Evolution (LTE)

LTE is a recent standard in cellular wireless communication.

LTE frame structure for FDD

- allocation of the grid to channels varies between sub-frames.
LTE is a recent standard in cellular wireless communication.

**Resource grid of a sub-frame**

- **Time domain (OFDM symbols)**
- **Frequency domain (sub-carriers)**

- 1 Sub-Frame (1ms)
- $2 \times N_{\text{symb}}^{DL} = 14$
- $N_{\text{RB}} = 100 \times 12$
- $N_{\text{sc}}^{RB} = 12$

- Resource grid of a sub-frame

- Resource element
- Reference signals

/* Electronic Systems Group */
LTE is a recent standard in cellular wireless communication.

LTE frame structure for FDD

- 1 Frame (10ms)
  - 1 Sub-Frame (1ms)
  - 1 Slot (0.5ms)

- 7 OFDM Symbols (short cyclic prefix)

Resource grid of a sub-frame

- Frequency domain (sub-carriers)
- Time domain (OFDM symbols)

- Allocation of the grid to channels varies between sub-frames.

/ Electronic Systems Group
FSM-SADF model of LTE

\[ S_1 \]

\[ S_2 : X(\text{dec}) = 192, R(p) = 1 \]
\[ S_3 : X(\text{dec}) = 970, R(p) = 13 \]
\[ S_4 : X(\text{dec}) = 895, R(p) = 12 \]
\[ S_5 : X(\text{dec}) = 820, R(p) = 11 \]

/Electronic Systems Group
Worst-case throughput computation

- WCT computation of the FSM-SADF model ($\times 10^{-4}$ sub-frames/time-unit)

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Static SDFG</td>
<td>Scenario graph</td>
<td>Reference schedule</td>
<td>State-space</td>
<td>MaxPlus</td>
</tr>
<tr>
<td>WCT</td>
<td>2.6</td>
<td>5.2</td>
<td>6.6</td>
<td>8.9</td>
<td>8.9</td>
</tr>
</tbody>
</table>

- Scenario-based techniques improve the static SDF result by 2 to 3.4 times more.
There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
Trade-offs between methods

- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.
Trade-offs between methods

- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.
- Methods 4 and 5 give the exact WCT, at the cost of run-time.
Trade-offs between methods

- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.
- Methods 4 and 5 give the exact WCT, at the cost of run-time.
- Method 5 has a run-time in the order of tens of seconds.

(a) Varying number of initial tokens
(b) Varying number of initial tokens
(c) Varying number of FSM states
Conclusions

- Static data ow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.
- Synchronous data ow scenarios can be used to capture dynamism in SDRs.
- Existing timing analysis techniques of SDF scenarios have very low run-time that scales well with increase in graph size.

Thank you! Questions?
Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to **pessimistic temporal analysis.**
Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to **pessimistic temporal analysis**.

- Synchronous dataflow scenarios can be used to **capture dynamism** in SDRs.
Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to **pessimistic temporal analysis**.

- Synchronous dataflow scenarios can be used to **capture dynamism** in SDRs.

- Existing timing analysis techniques of SDF scenarios have very **low run-time that scales well** with increase in graph size.
Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.
- Synchronous dataflow scenarios can be used to capture dynamism in SDRs.
- Existing timing analysis techniques of SDF scenarios have very low run-time that scales well with increase in graph size.

Thank you! Questions?