

Towards Runtime Adaptivity by using Models of Computation for Real-Time Embedded Systems Design

D. S. Loubach ¹ E. G. O. Nóbrega ¹ I. Sander ² I. Söderquist ³
O. Saotome ⁴

¹University of Campinas - UNICAMP

²KTH Royal Institute of Technology

³Aeronautics, Saab AB

⁴Aeronautics Institute of Technology - ITA

9th Aerospace
Technology
Congress
2016

LOUBACH,
D.S. *et al*

About Denis
Loubach

Introduction

Background

Case Study

Specification Model
SoC Architecture
Implementation
Model

Results

Static Data
Runtime Data

Conclusion

Ack.

Q&A

- 1 About Denis Loubach
- 2 Introduction
- 3 Background
- 4 Case Study
- 5 Results
- 6 Conclusion
- 7 Acknowledgments
- 8 Q&A

Assistant Professor at UNICAMP, since 2014

Background

- PhD in Electronics and Computer Engineering, ITA, 2012
Brazilian Aeronautics Institute of Technology - ITA
- MSc (ITA, 2007) and BSc in Computer Engineering (UMC, 2004)

Industry experience as development engineer (*e.g.*, Embraer 2011-2014)

Research Interests:

- Reconfigurable computing
- Embedded systems
- Real time systems
- Real time operating systems
- Model-based design

Assistant Professor at UNICAMP, since 2014

Background

- PhD in Electronics and Computer Engineering, ITA, 2012
Brazilian Aeronautics Institute of Technology - ITA
- MSc (ITA, 2007) and BSc in Computer Engineering (UMC, 2004)

Industry experience as development engineer (*e.g.*, Embraer 2011-2014)

Research Interests:

- Reconfigurable computing
- Embedded systems
- Real time systems
- Real time operating systems
- Model-based design

Assistant Professor at UNICAMP, since 2014

Background

- PhD in Electronics and Computer Engineering, ITA, 2012
Brazilian Aeronautics Institute of Technology - ITA
- MSc (ITA, 2007) and BSc in Computer Engineering (UMC, 2004)

Industry experience as development engineer (e.g., Embraer 2011-2014)

Research Interests:

- Reconfigurable computing
- Embedded systems
- Real time systems
- Real time operating systems
- Model-based design

Assistant Professor at UNICAMP, since 2014

Background

- PhD in Electronics and Computer Engineering, ITA, 2012
Brazilian Aeronautics Institute of Technology - ITA
- MSc (ITA, 2007) and BSc in Computer Engineering (UMC, 2004)

Industry experience as development engineer (*e.g.*, Embraer 2011-2014)

Research Interests:

- Reconfigurable computing
- Embedded systems
- Real time systems
- Real time operating systems
- Model-based design

Motivation: aeronautics current trend towards the second generation of **integrated modular avionics** (IMA2G), a.k.a. distributed IMA

Our paper shows that **runtime adaptivity** is feasible to be performed

Case study implementation on both software and hardware, using a **heterogeneous SoC** (uC + FPGA)

To capture and deal with the complexity of adaptivity, we modeled a case study using the theory of **formal models of computation** (MoC), mainly the synchronous MoC

Motivation: aeronautics current trend towards the second generation of **integrated modular avionics** (IMA2G), a.k.a. distributed IMA

Our paper shows that **runtime adaptivity** is feasible to be performed

Case study implementation on both software and hardware, using a **heterogeneous SoC** (uC + FPGA)

To capture and deal with the complexity of adaptivity, we modeled a case study using the theory of **formal models of computation** (MoC), mainly the synchronous MoC

Motivation: aeronautics current trend towards the second generation of **integrated modular avionics** (IMA2G), a.k.a. distributed IMA

Our paper shows that **runtime adaptivity** is feasible to be performed

Case study implementation on both software and hardware, using a **heterogeneous SoC** (uC + FPGA)

To capture and deal with the complexity of adaptivity, we modeled a case study using the theory of **formal models of computation** (MoC), mainly the synchronous MoC

Motivation: aeronautics current trend towards the second generation of **integrated modular avionics** (IMA2G), a.k.a. distributed IMA

Our paper shows that **runtime adaptivity** is feasible to be performed

Case study implementation on both software and hardware, using a **heterogeneous SoC** (uC + FPGA)

To capture and deal with the complexity of adaptivity, we modeled a case study using the theory of **formal models of computation** (MoC), mainly the synchronous MoC

Main concepts:

- Model of Computation (MoC)
- Event, Signal, Process
- Synchronous MoC
- ForSyDe
- Real-time embedded systems
- Adaptivity
 - Full vs. Partial reconfiguration
- Modeling of adaptivity
 - Signals carrying functions
 - Adaptive process

Main concepts:

- Model of Computation (MoC)
- Event, Signal, Process
- Synchronous MoC
- ForSyDe
- Real-time embedded systems
- Adaptivity
 - Full vs. Partial reconfiguration
- Modeling of adaptivity
 - Signals carrying functions
 - Adaptive process

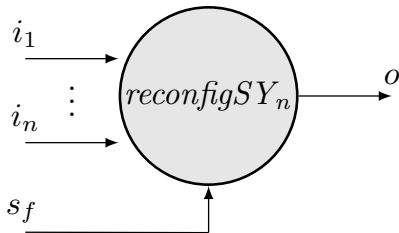


Figure: Process constructor for *reconfigSY* adaptive process

$$ap_f = reconfigSY_n$$

where:

$$o = ap_f(s_f, (i_1, \dots, i_n))$$

$$o[k] = s_f[k](i_1[k], \dots, i_n[k])$$

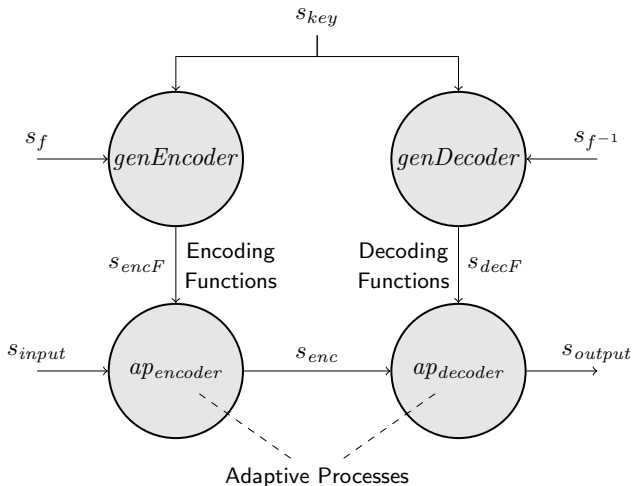


Figure: Specification model

Processes definitions:

$$\mathit{genEncoder} : s_f[k](s_{key}[k]) = s_{encF}[k] \quad (1)$$

$$\mathit{apEncoder} : s_{encF}[k](s_{input}[k]) = s_{enc}[k] \quad (2)$$

$$\mathit{genDecoder} : s_{f-1}[k](s_{key}[k]) = s_{decF}[k] \quad (3)$$

$$\mathit{apdecoder} : s_{decF}[k](s_{enc}[k]) = s_{output}[k] \quad (4)$$

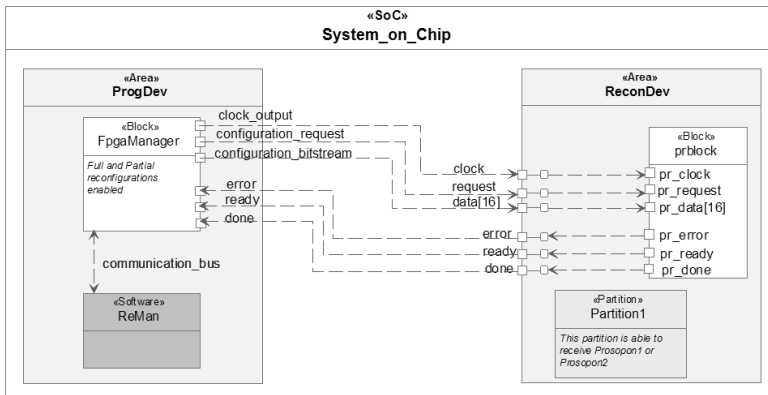


Figure: SoC with the ProgDev and ReconDev areas overview. It also contains the ReMan software application inside the ProgDev

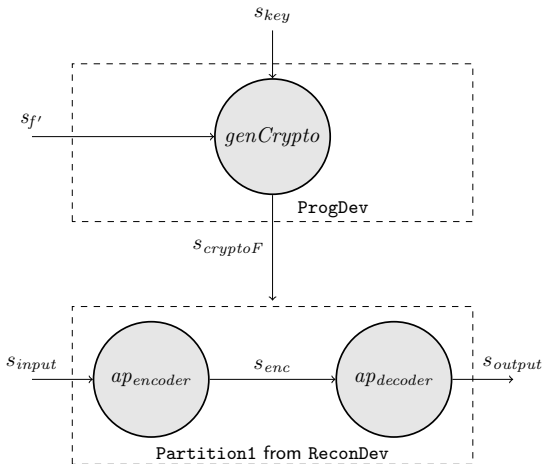


Figure: Encoder and decoder implementation model showing both SoC areas: ProgDev and ReconDev with one partition named Partition1

Signals containing the encoding and decoding functions:

$$s_f = \langle (enc_{AES}), (enc_{DES}), \dots \rangle$$

$$s_f^{-1} = \langle (dec_{AES}), (dec_{DES}), \dots \rangle$$

s_f and s_f^{-1} signals turned in $s_{f'}$ (for simplicity):

$$s_{f'} = \langle (f_{AES}), (f_{DES}), \dots \rangle \quad (5)$$

where:

f_{AES} stands for AES cryptography (encoder/decoder) function;

f_{DES} for the DES cryptography (encoder/decoder) function

Signals containing the encoding and decoding functions:

$$s_f = \langle (enc_{AES}), (enc_{DES}), \dots \rangle$$

$$s_f^{-1} = \langle (dec_{AES}), (dec_{DES}), \dots \rangle$$

s_f and s_f^{-1} signals turned in $s_{f'}$ (for simplicity):

$$s_{f'} = \langle (f_{AES}), (f_{DES}), \dots \rangle \quad (5)$$

where:

f_{AES} stands for AES cryptography (encoder/decoder) function;

f_{DES} for the DES cryptography (encoder/decoder) function

Table: Full reconfiguration static data

FR	Measure	Number
	ALM	2,896
	Bytes	7,007,204

Table: *Prosopons* and the number of ALM needed to implement them

<i>Prosopon</i>	ALM
<i>prosopon_{AES}</i>	2,849
<i>prosopon_{DES}</i>	1,019

Table: *Prosopons* bitstream modes and sizes. (*)Data given in Bytes

Bitstream mode	<i>prosopon</i> _{AES} [*]	<i>prosopon</i> _{DES} [*]
Scrub	1,873,272	1,873,272
And/or	3,081,512	3,000,628

- ReconDev clock: 50 MHz (20 nano seconds period)
- ProgDev clock: 925 MHz

Table: Functions and sub functions runtime clock cycles and computation time

Main function	Sub function	Clock cycles	Time [ns]
f_{AES}	encode	63	1,260
f_{AES}	decode	63	1,260
f_{DES}	encode	18	360
f_{DES}	decode	19	380

Partial and full reconfigurations

Table: Partial reconfiguration measured times. (*)Time is given in mili seconds [ms]

Bitstream mode	<i>prosopon_{AES}</i> *	<i>prosopon_{DES}</i> *
Scrub	7.76	7.76
And/or	12.7	12.4

Table: Full reconfiguration with no data compression measured time. (*)Time is given in mili seconds [ms]

Bitstream	Time*
FR	29.6

We introduce in this paper a **runtime adaptivity case study** using **formal models of computation** (MoC) for real-time embedded systems design

We modeled an encoder/decoder system in a high-level of abstraction (*system specification*) using ForSyDe

Next, we refined that model and manually transformed it into an *implementation specification* to be designed in software/hardware

We introduce in this paper a **runtime adaptivity case study** using **formal models of computation** (MoC) for real-time embedded systems design

We modeled an encoder/decoder system in a high-level of abstraction (*system specification*) using ForSyDe

Next, we refined that model and manually transformed it into an *implementation specification* to be designed in software/hardware

We introduce in this paper a **runtime adaptivity case study** using **formal models of computation** (MoC) for real-time embedded systems design

We modeled an encoder/decoder system in a high-level of abstraction (*system specification*) using ForSyDe

Next, we refined that model and manually transformed it into an *implementation specification* to be designed in software/hardware

Results showed that one partial reconfiguration takes less time to complete than a full reconfiguration

This enables the possibility to **runtime reconfiguration using partial reconfiguration techniques**, then leading to runtime adaptivity feasibility

Results showed that one partial reconfiguration takes less time to complete than a full reconfiguration

This enables the possibility to **runtime reconfiguration using partial reconfiguration techniques**, then leading to runtime adaptivity feasibility

This research work is supported by:

- Swedish-Brazilian Research and Innovation Centre (CISB) – Project CISB ID 68-2015-A
- Regular Research Awards grant #2014/24855-8 São Paulo Research Foundation - FAPESP
- Altera University Program

Thank you for your attention!

Denis Loubach

Assistant Professor, UNICAMP

Email: dloubach@fem.unicamp.br

Academic page: <http://www.fem.unicamp.br/~dloubach/>

Lab page: <http://www.fem.unicamp.br/~acceslab/>