A Bus-based SoC Architecture for Flexible Module Placement on Reconfigurable FPGAs

Andreas Oetken, Stefan Wildermann, Jürgen Teich and Dirk Koch

andreas.oetken@cs.fau.de

Department of Computer Science
Hardware-Software-Co-Design
University of Erlangen-Nuremberg, Germany

August 31, 2010
Outline

- Introduction
- Architecture
- Design Flow
- Run-time Reconfiguration
- Smart Camera
- Implementation Results
- Conclusion
Introduction

BIG FPGA

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Introduction

Static Part
(e.g. µP, Memory Controller, UART, Ethernet, Bus)

Dynamic Part 1
(e.g. Videofilter, HW-Accelerator)

Dynamic Part 2
(e.g. Videofilter, HW-Accelerator)

Dynamic Part 3
(e.g. Videofilter, HW-Accelerator)

Dynamic Part 4
(e.g. Videofilter, HW-Accelerator)

EXCLUSIVE USE
Introduction
Introduction

Static Part
(e.g. µP, Memory Controller, UART, Ethernet, Bus)

Dynamic Module 1
(e.g. Videofilter, HW-Accelerator)

Dynamic Module 2
(e.g. Videofilter, HW-Accelerator)

Dynamic Module 3
(e.g. Videofilter, HW-Accelerator)

Dynamic Module 4
(e.g. Videofilter, HW-Accelerator)
Introduction

Static Part

Dynamic Module 1..4

COMMON COMMUNICATION INTERFACE
Introduction

FLEXIBLE SIZE, POSITION AND NUMBER

COMMON COMMUNICATION BUS

Static Part
Dynamic Module 1
Dynamic Module 2

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Introduction

- Run-time reconfiguration
- Dynamic position, size and number of modules
- Common bus system

→ ReBus
Basic Architecture

University of Erlangen-Nuremberg, Germany
Andreas Oetken

SoC

Static SoC part

Local Bus

Bridge

Dynamic Module 1
Dynamic Module 2
Dynamic Module 3

Reconfigurable SoC part

ReCoBus

Dynamic

Static SoC part

SoC
ReCoBus

▶ Hardmacro with a regular structure
▶ Multiple lanes to reduce latency and resource utilisation consisting of:
  ▶ Read/write signal
  ▶ Module address signal (base address)
  ▶ Connection point for n bit data
  ▶ Connection points for interrupts/bus-requests line (shared resources)
  ▶ Connection point for m bit internal address
ReCoBus

static part

z = 6

module select & read/write

data in/out

request

address

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Module Types

- Simple slave module
  - Small address range
  - Small data width (e.g. $1 \text{ lane} \times 8 \frac{\text{bit}}{\text{lane}} = 8$ bit data width)
Module Types

- Simple slave module
  - Small address range
  - Small data width (e.g. $1\text{ lane} \times \frac{8\text{ bit}}{\text{lane}} = 8\text{ bit data width}$)
- Big slave module
  - Wide address range
  - Wide data range (e.g. $4\text{ lanes} \times \frac{8\text{ bit}}{\text{lane}} = 32\text{ bit}$)
Module Types

- **Simple slave module**
  - Small address range
  - Small data width (e.g. $1 \text{ lane} \times 8 \frac{\text{bit}}{\text{lane}} = 8 \text{ bit data width}$)

- **Big slave module**
  - Wide address range
  - Wide data range (e.g. $4 \text{ lanes} \times 8 \frac{\text{bit}}{\text{lane}} = 32\text{bit}$)

- **Large master module**
  - Wide address range
  - Wide data range
  - Additional control signals (e.g. $(4 + 2) \text{ lanes} \times 8 \frac{\text{bit}}{\text{lane}} = 32\text{databits} + 16\text{controlbits}$)
Bus Bridge

- Bidirectional communication

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Bus Bridge

- Bidirectional communication
- Burst-capable
Bus Bridge

- Bidirectional communication
- Burst-capable
- Includes ReCoBus arbiter

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Bus Bridge

- Bidirectional communication
- Burst-capable
- Includes ReCoBus arbiter
- Configurable bus parameters
  - Address range
  - Data width
  - IRQ/Bus requests

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Exemplary Architecture

- Virtex 2 Pro Development Board
- Video processing SoC
- I/O streaming bar
Custom Design Flow (Design-Time)

- Routing constraints using blocker macros:
  - No static routing inside the reconfigurable area
  - No module routing outside the module area
- Exclusive CLB/Block-Ram/Multiplier usage inside the module or static area
Custom Design Flow

- static
  - reconfigurable areas
  - bus properties

- partial
  - module size
  - bus properties

toolkit

selected depending on the design type

- bus macros
- constraints
- routing blocker

mapped netlist

- routed netlist

bitfile

partial bitfile

design flow

synthesize
translate/map/place
route
generate bitstream
cut out partial module

design parameters

static

- reconfigurable areas
- bus properties

partial

- module size
- bus properties

selected depending on the design type

- bus macros
- constraints
- routing blocker

mapped netlist

- routed netlist

bitfile

partial bitfile

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Run-time Reconfiguration

1 Load
2 Store

Module Repository
Static Image

I/O Device
(Flash Card, Ethernet, Serial Connection)

Controller
(µP Software, Hardware)

Memory
(e.g. DDR)

ICAP

partial-module-bitfiles
static-bitfile

partial-modules
reconfig. areas
frame

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Run-time Reconfiguration

- Load bitfiles into memory
  - Prefetch modules
  - Static conf. Image
Run-time Reconfiguration

- Load bitfiles into memory
  - Prefetch modules
  - Static conf. Image

- Module repository

Module Repository
Static Image

partial modules
reconfig. areas
Run-time Reconfiguration

- Load bitfiles into memory
  - Prefetch modules
  - Static conf. Image
- Module repository
- Static image as virtual representation of the FPGA
Run-time Reconfiguration

- Load bitfiles into memory
  - Prefetch modules
  - Static conf. Image
- Module repository
- Static image as virtual representation of the FPGA
- Insert modules into the static image
Run-time Reconfiguration

- Load bitfiles into memory
  - Prefetch modules
  - Static conf. Image
- Module repository
- Static image as virtual representation of the FPGA
- Insert modules into the static image
- Write modified frames to the FPGA (ICAP)
Smart Camera Application

Video dataflow

Video dataflow
Smart Camera Application

- Preprocessing
  (Skin color extraction)

Video dataflow

YCbCr
Skin-color

Video dataflow
Smart Camera Application

- Preprocessing (Skin-color extraction)
- Pass extracted information to the DDR
Smart Camera Application

- Preprocessing (Skincolor extraction)
- Pass extracted information to the DDR
- Control intensive software part on PowerPC
Smart Camera Application

- Preprocessing (Skin color extraction)
- Pass extracted information to the DDR
- Control intensive software part on PowerPC
- Hardware accelerated part
Smart Camera Application

- Preprocessing (Skin color extraction)
- Pass extracted information to the DDR
- Control intensive software part on PowerPC
- Hardware accelerated part
- Visualisation and integration
# Reconfiguration Time

<table>
<thead>
<tr>
<th>module</th>
<th>slots (w/h)</th>
<th>irq/req</th>
<th>size(B)</th>
<th>load</th>
<th>write</th>
<th>connection</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>marker</td>
<td>4/1</td>
<td>n/n</td>
<td>14.120</td>
<td>15,9</td>
<td>1,9</td>
<td>0,3</td>
<td>18,3</td>
</tr>
<tr>
<td>skin</td>
<td>7/1</td>
<td>n/n</td>
<td>24.683</td>
<td>27,9</td>
<td>3,3</td>
<td>0,3</td>
<td>31,5</td>
</tr>
<tr>
<td>accel.</td>
<td>7/2</td>
<td>y/y</td>
<td>49.323</td>
<td>56</td>
<td>3,3</td>
<td>4,5</td>
<td>63,8</td>
</tr>
</tbody>
</table>

Factors of influence

- Module size (height and width) ⇒ load time
- Module width ⇒ configuration time
- Module height ⇨ configuration time (frames)
- Interrupt/Request connection/Setting base address ⇒ connection time
Future Work

▶ Current technology port (First Spartan 6 prototype)
▶ Flexible self-adaptive applications
▶ Partial μP module (R-MP-SoC)
▶ Automatic macro generation

University of Erlangen-Nuremberg, Germany
Andreas Oetken
Conclusion

- Reconfigurable SoC allowing
  - Fine grained run-time reconfiguration
  - Module relocation
  - Module reusability (predefined interface)
- Dynamic placement for run-time reconfiguration
- Image processing algorithm
Thanks for your attention!

Questions?