

Review of:

Building Run-Time Reconfigurable Systems from Tiles

•Paper by:

- Gareth Lee (The University of Western Australia)
- George Milne (The University of Western Australia)

•Published in:

- Field Programmable Logic and Applications (FPL03)
- Lisbon/Portugal
- September 2003

•Review by:

- Phillip H. Jones

Outline

- **Motivation**
- Approach
- Framework
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- Related Work
- Conclusions

Motivation

- For several years commercial FPGAs have had the capability to perform run-time partial reconfiguration, however very little work has been done to create Applications/Systems to take advantage of this functionality.
- Providing a framework for system/application designers to utilize, may increase the attention given to developing run-time reconfigurable systems/applications.

- Motivation
- **Approach**
- Framework
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- Related Work
- Conclusions

Approach

- Concepts from Object oriented programming are used to:
 - Develop a “Tile” based design methodology
 - Help explain new concepts that are presented in this “Tile” based methodology

- Motivation
- Approach
- **Framework**
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- Related Work
- Conclusions

Framework

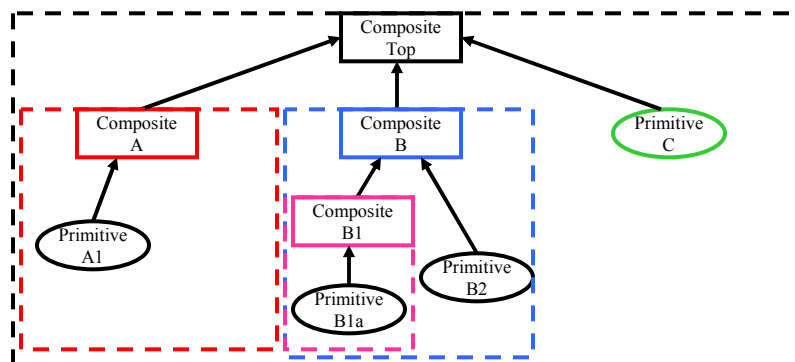
- Relevant Component Definitions
 - Abstract Tile: An object which contains the following
 - Set of signal ports, mapped to edges of a rectangular area
 - A plain English specification of its behavior
 - A unique name
 - Concrete Tile: A specific configuration of an Abstract Tile.
 - An Abstract Tile can have Multiple concrete that map to it
 - Primitive Tile: A Tile which contains the physical description of a circuit.
 - An HDL description
 - Parameterizable core
 - Composite Tile: A Tile made up of one or more Primitive Tiles and possibly other Composite Tiles.
 - Controlling Object: Software object associated with each Abstract Tile.
 - responsible for managing the configuration of it's assigned Tile

CS6814: Fall 2003

Washington University in St. Louis

Framework

Example of a Tile based application layout



CS6814: Fall 2003

Washington University in St. Louis

- Motivation
- Approach
- Framework
- **Run-Time Reconfiguration Method**
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- Related Work
- Conclusions

Run-time Reconfiguration Method

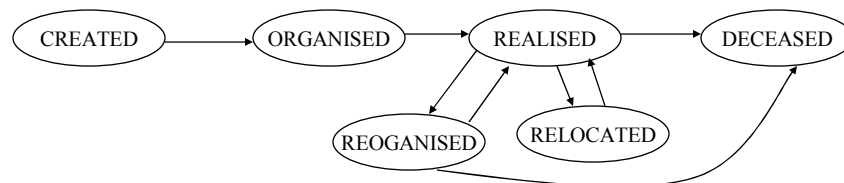
- Tile State Model
 - Each controlling object walks its corresponding Tile through the following States:
 - **CREATED:**
 - Initial State of all Tiles
 - Control object must obtain configuration parameters before transitioning its Title to the ORGANIZED State
 - **ORGANIZED:**
 - Appropriately structure (connect) the sub-tiles that make up the current tile
 - **REALISED:**
 - Each Tile generates a physical mapping (bit-stream for the case of an FPGA)

Run-time Reconfiguration Method

- **Tile State Model (cont.)**
 - Only during Reconfiguration may a Tile visit the following States:
 - **REORGANIZED:**
 - Substitute a functionally equivalent sub-tile organization for Tile under configuration.
 - **RELOCATED:**
 - Move a sub-tile to a physically different location
 - » Make room for other sub-tiles during reorganization
 - » Resource Management (e.g. Garbage collection)
 - **DECEASED:**
 - Tile is no longer in use by the design. (i.e. permanently inactive)

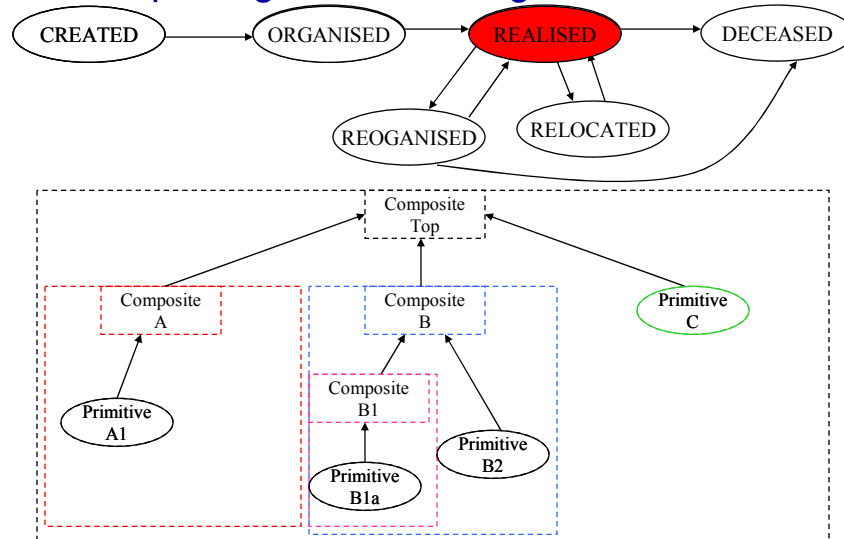
Run-time Reconfiguration Method

Relationship Between States



Run-time Reconfiguration Method

Example High Level Configuration Process



CS6814: Fall 2003

Washington University in St. Louis

- Motivation
- Approach
- Framework
- Run-Time Reconfiguration Method
- **Example Application (Regular Expressions)**
- Example Resource Management Strategy
- Related Work
- Conclusions

CS6814: Fall 2003

Washington University in St. Louis

Example Application (Regular Expressions)

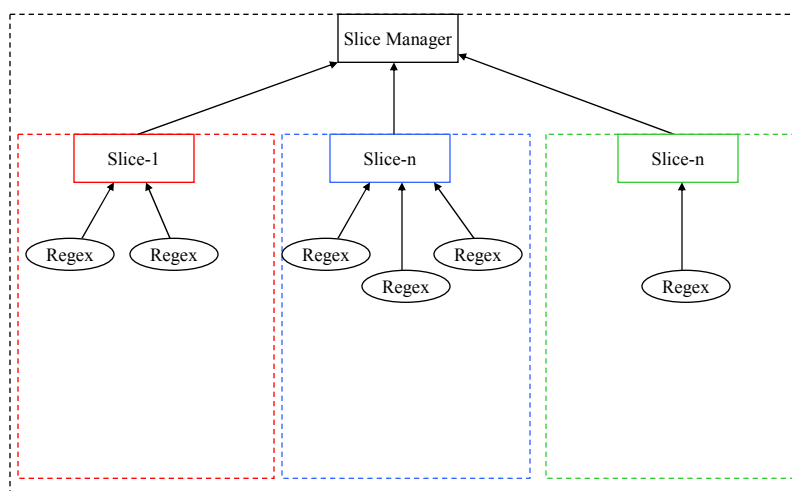
- **Regular Expression Application**
 - Match user define regular expressions within a streaming text input.
- **Mapping to Tile Methodology**
 - Each regular expression is implemented as a State machine and is encapsulated into a primitive tile (Regex)
 - Multiple primitive tiles are group into an Abstract Tile (Slice)
 - All Slice Tiles are sub-tiles of the a root Abstract Tile called “Slice Manager”

CS6814: Fall 2003

Washington University in St. Louis

Example Application (Regular Expressions)

Tile Structure

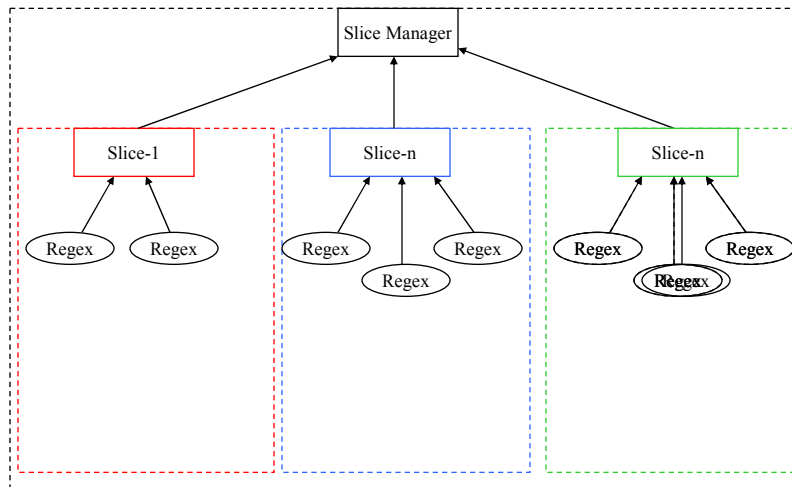


CS6814: Fall 2003

Washington University in St. Louis

Example Application (Regular Expressions)

High level Reconfiguration to add more Regular Expressions



CS6814: Fall 2003

Washington University in St. Louis

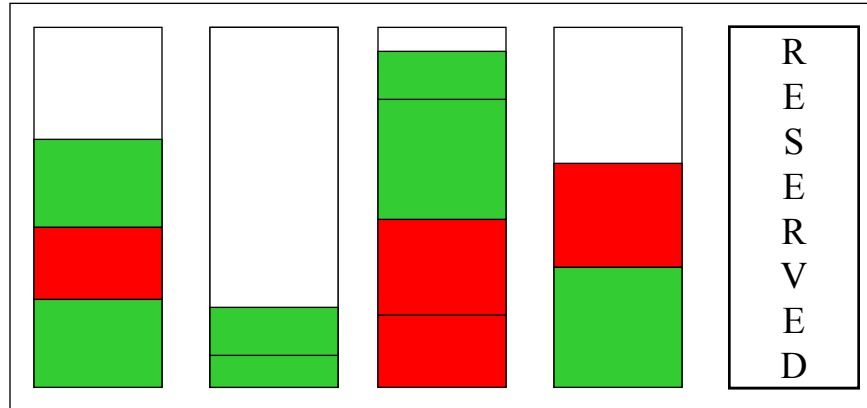
- Motivation
- Approach
- Framework
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- **Example Resource Management Strategy**
- Related Work
- Conclusions

CS6814: Fall 2003

Washington University in St. Louis

Example Resource Management Strategy

Incremental Incrementally-Compaction Garbage Collection



CS6814: Fall 2003

 Washington University in St. Louis

- Motivation
- Approach
- Framework
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- **Related Work**
- Conclusions

CS6814: Fall 2003

 Washington University in St. Louis

Related Work

- “A Methodology for design of run-time reconfigurable systems”
 - More detailed explanation of Titles and how to use them to design **run-time reconfigurable** systems

- Motivation
- Approach
- Framework
- Run-Time Reconfiguration Method
- Example Application (Regular Expressions)
- Example Resource Management Strategy
- Related Work
- **Conclusions**

Conclusions

- A methodology was presented to take advantage of partial reconfiguration capabilities supplied by modern FPGAs
 - Re-Introduces the concept of Tiles
 - Assigning a software control object to each Abstract tile.
 - Gives a novel example of how Resource management could be accomplished in an FPGA while using real-time reconfiguration