

From UML Modelling to FPGA Implementation of Intensive Signal Processing Tasks



Sébastien LE BEUX
PhD student
lebeux@lifl.fr



Introduction



- Context
 - Software applications more and more complex
 - Hardware architecture complexity increase
- Software modelling: UML
 - High level modelling of Intensive Signal Processing applications
 - Compound and repetitive components
- FPGA configuration for ISP applications from UML
 - "Classical" configuration for elementary tasks
 - Configuration generation for repetition from high level modelling

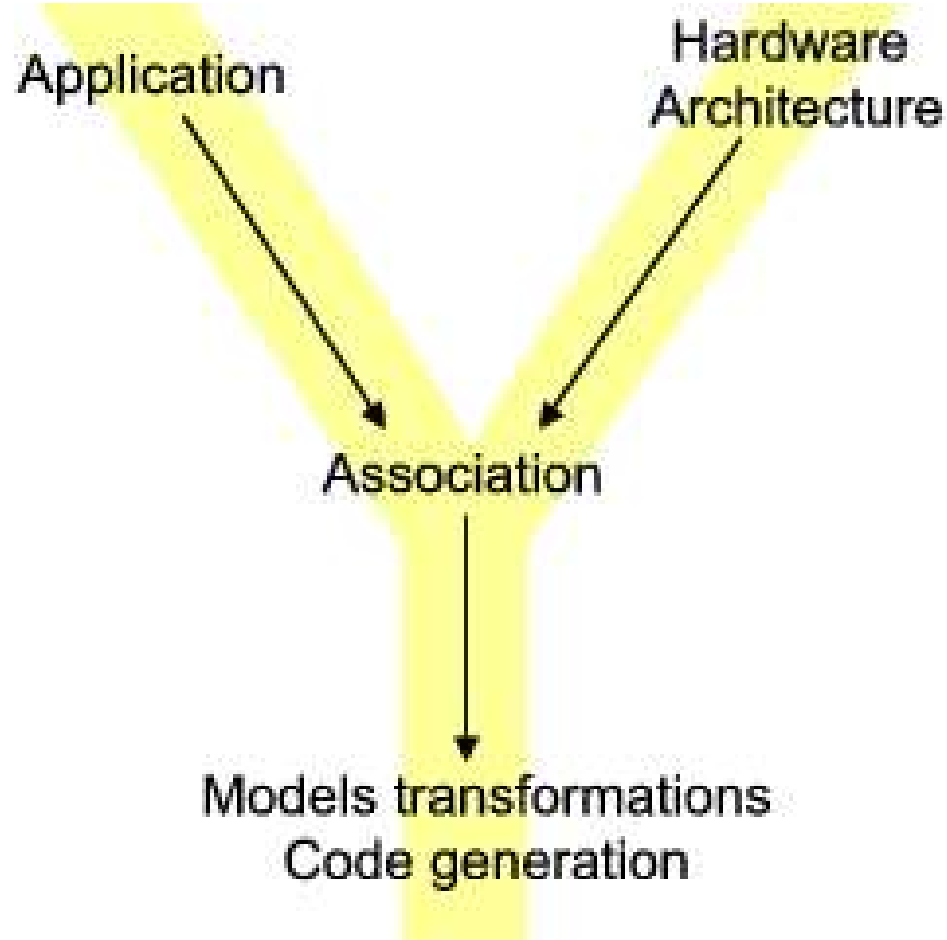
Overview



- UML modelling
- BLIF format
- Introduction of repetition in BLIF file
- Implementation of repetitive tasks on hardware
- A tool to implement applications on FPGA

- UML modelling

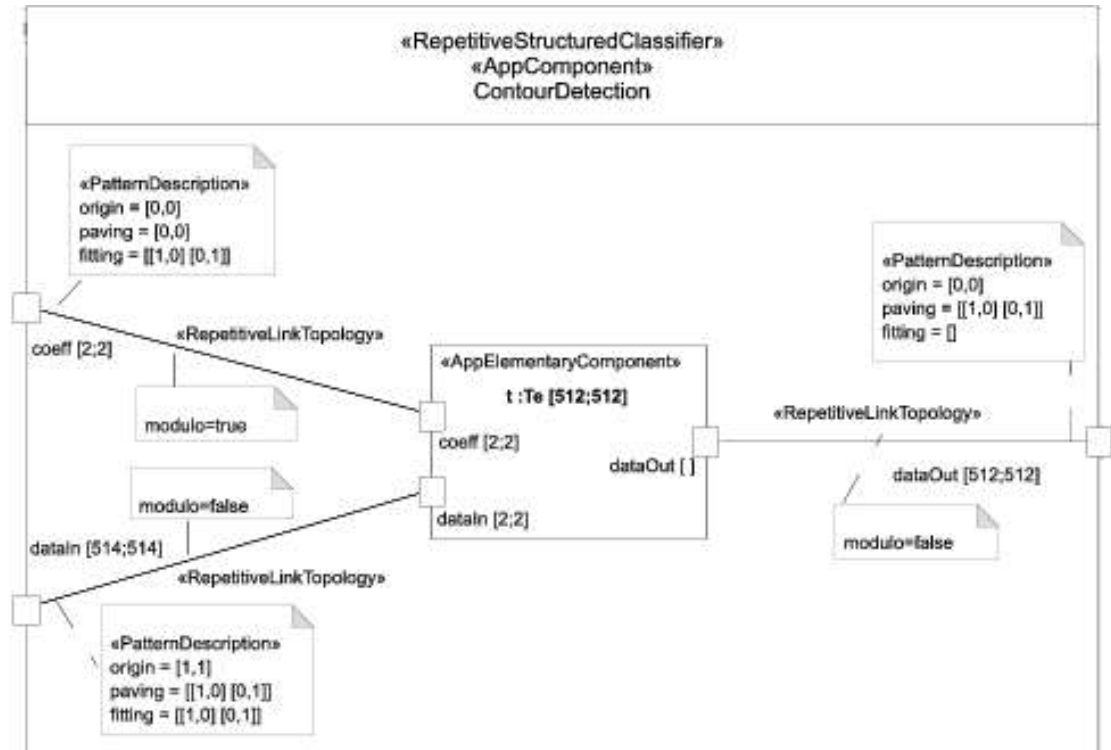
Gaspard flow



Application model



- Kind of component
 - Compound
 - Repetitive
 - Elementary
- ArrayOL repetition
 - Origin
 - Paving
 - Fitting



- BLIF format

Berkeley Logical Interchange Format



- From University of California, Berkeley
- Describe a logic-level hierarchical circuit in textual form
- Description of:
 - LUT
 - Flip flops
 - Finite state machine
 - ...
- Edif2blif, SIS...

Structure of BLIF file

```
.model t7
.inputs {List of inputs}
.outputs {List of outputs}
.clocks
.subckt t0 {List of connections}
    .model t0
    .inputs {List of inputs}
    .outputs {List of outputs}
    .clocks
    .names {True table inputs and
result}
    111- 1
    11-1 1
    ...
    .end
```

Structure of BLIF file

```
.model t7
```

```
.inputs {List of inputs}  
.outputs {List of outputs}  
.clocks  
.subckt t0 {List of connections}  
  .model t0  
  .inputs {List of inputs}  
  .outputs {List of outputs}  
  .clocks  
  .names {True table inputs and  
  result}  
  111- 1  
  11-1 1  
  ...  
.end
```



Global component t7

The diagram shows a large red rectangular box representing a global component. Inside the box, the text "Global component t7" is written in red. On the left side of the box, two horizontal red arrows point into the box, representing inputs. On the right side of the box, one horizontal red arrow points out of the box, representing an output.

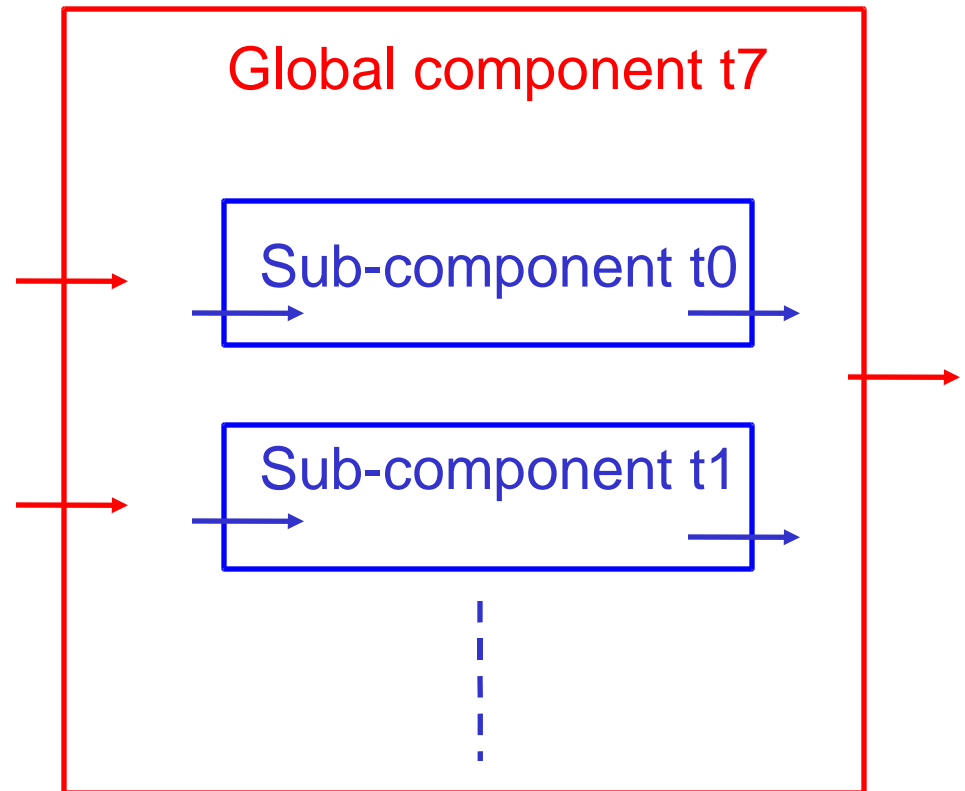
Structure of BLIF file

`.model t7`

`.inputs {List of inputs}`
`.outputs {List of outputs}`
`.clocks`
`.subckt t0 {List of connections}`

`.model t0`

`.inputs {List of inputs}`
`.outputs {List of outputs}`
`.clocks`
`.names {True table inputs and result}`
`111- 1`
`11-1 1`
`...`
`.end`



Structure of BLIF file

`.model t7`

`.inputs {List of inputs}`

`.outputs {List of outputs}`

`.clocks`

`.subckt t0 {List of connections}`

`.model t0`

`.inputs {List of inputs}`

`.outputs {List of outputs}`

`.clocks`

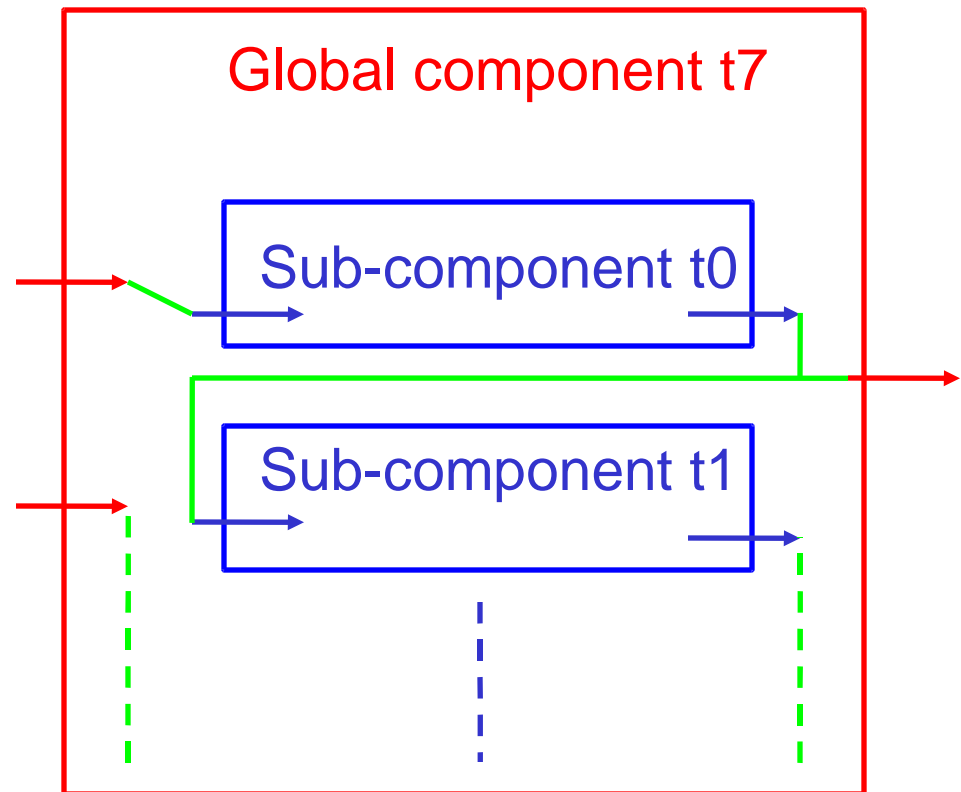
`.names {True table inputs and result}`

111- 1

11-1 1

...

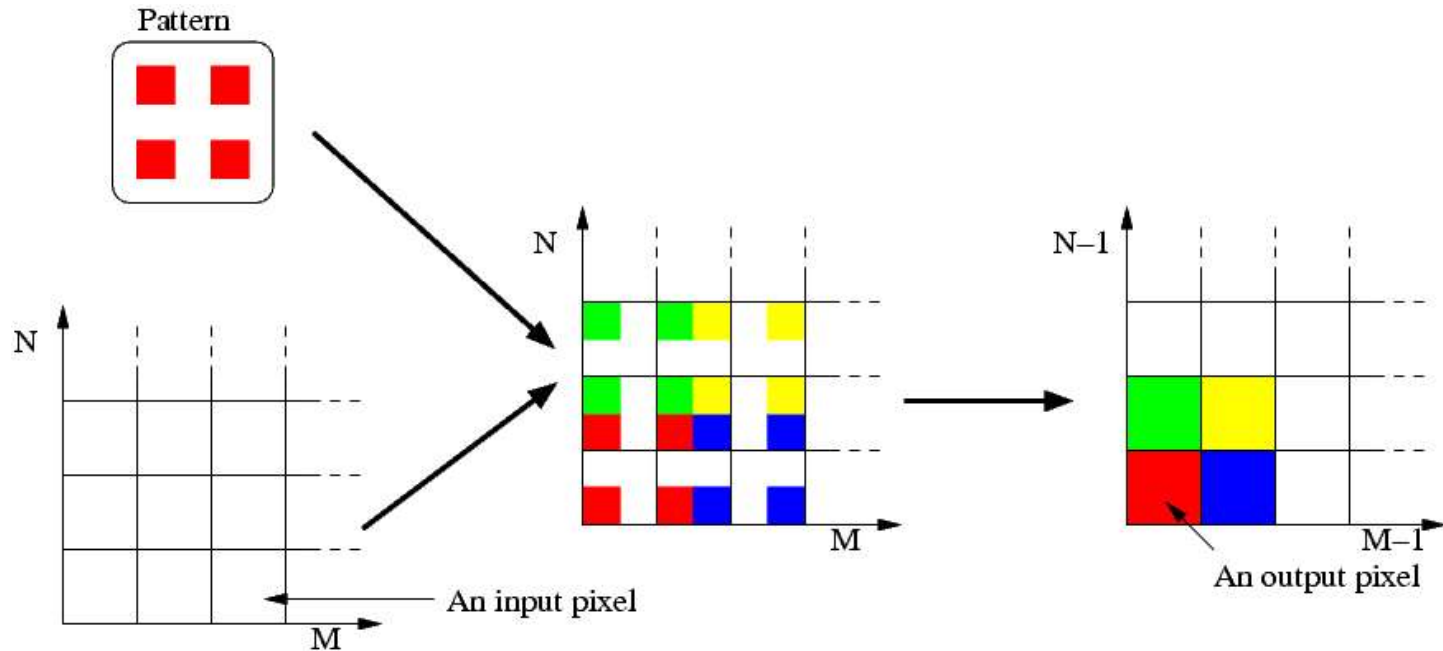
`.end`



- Introduction of repetition in BLIF file

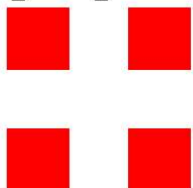

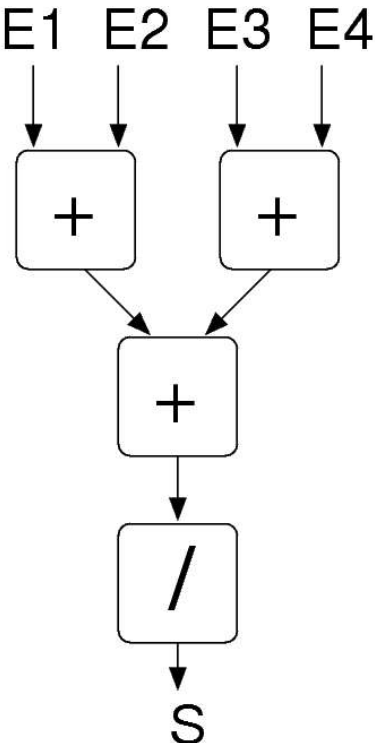
Example

- Input: picture of $M \times N$ pixels
- Convolution application on input picture
- Output: picture of $(M-1) \times (N-1)$ pixels



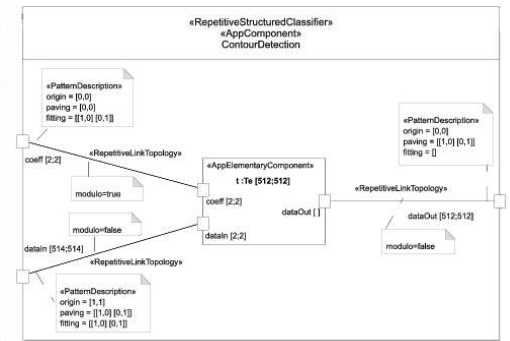
Elementary task



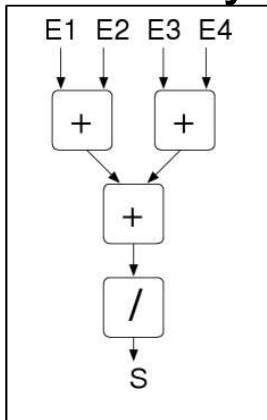
Pattern	Data flow graph	BLIF file
<p>Input pattern</p>  <p>Output pattern</p> 		<pre>.model t7 .inputs {List of inputs} .outputs {List of outputs} .clocks .subckt t0 {List of connections} .model t0 .inputs {List of inputs} .outputs {List of outputs} .clocks .names {True table inouts} 111- 1 11-1 1 end .end</pre>

Repetitive task

- UML model of the application
 - Origin
 - Paving
 - Fitting

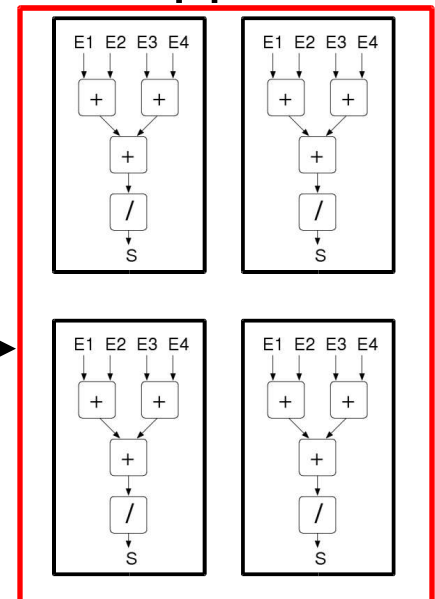


Elementary task



Code generation

Full application



But...



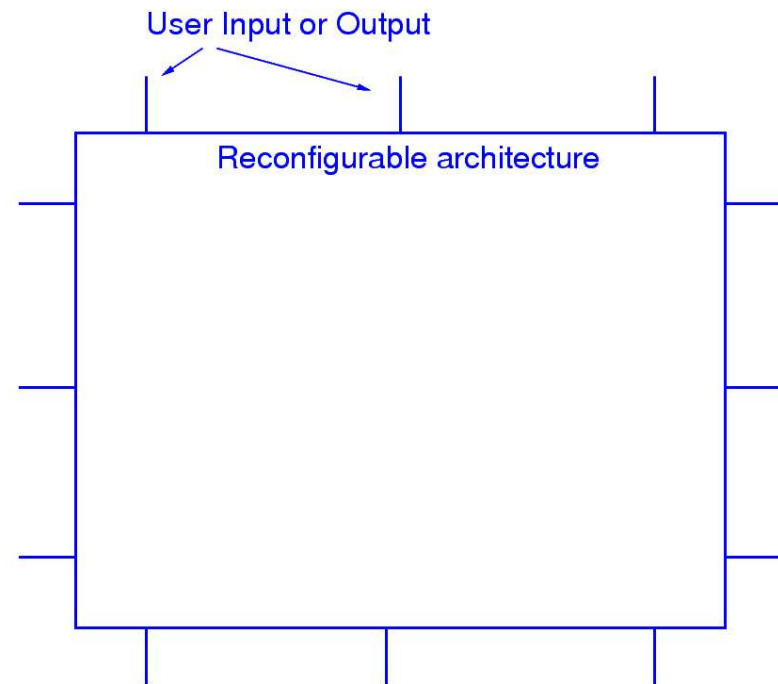
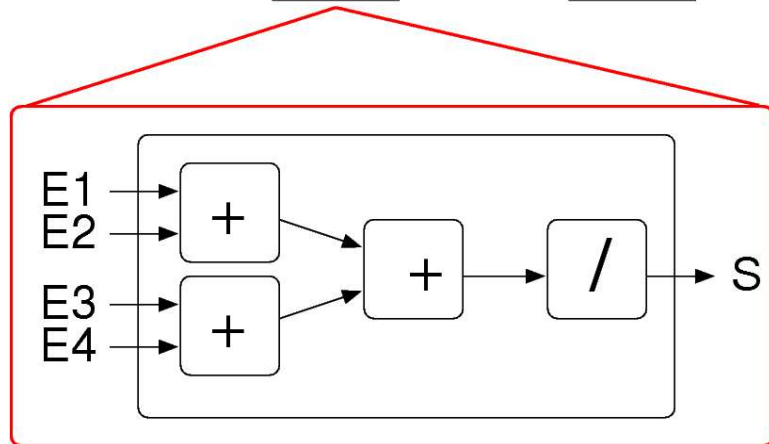
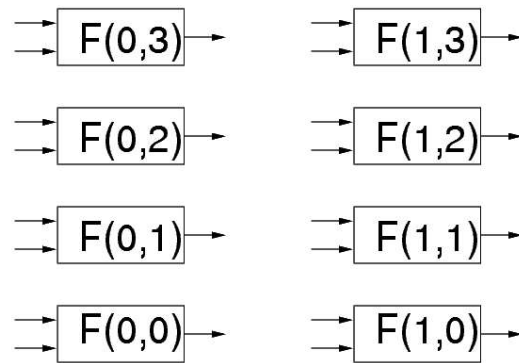
- Characteristics of reconfigurable architecture to implement?

- Implementation of repetitive tasks on hardware

Characteristics of reconfigurable architecture



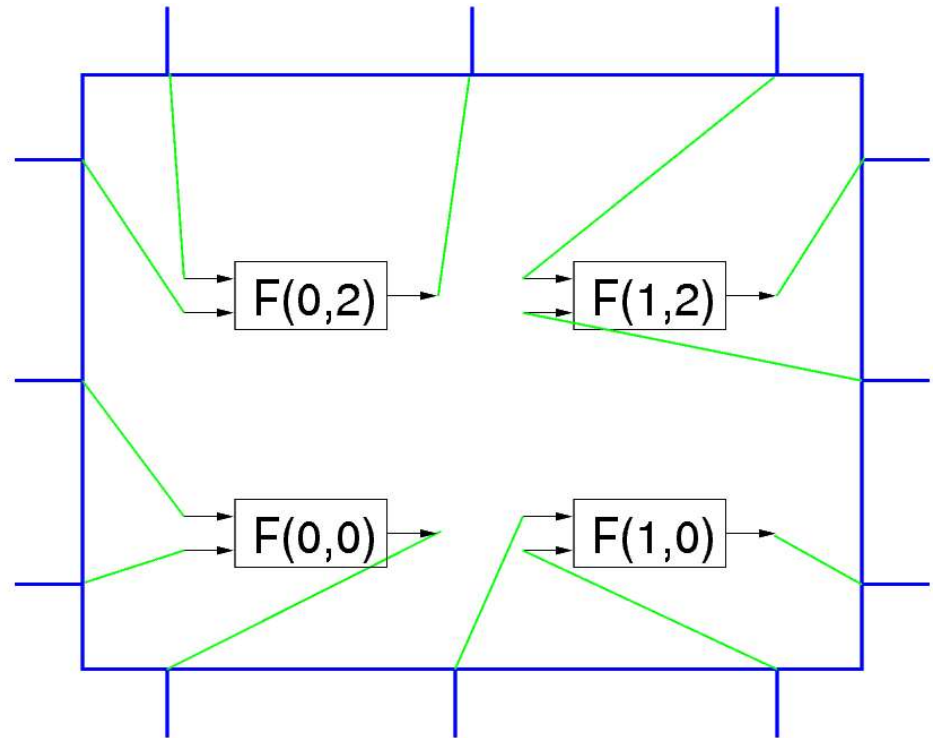
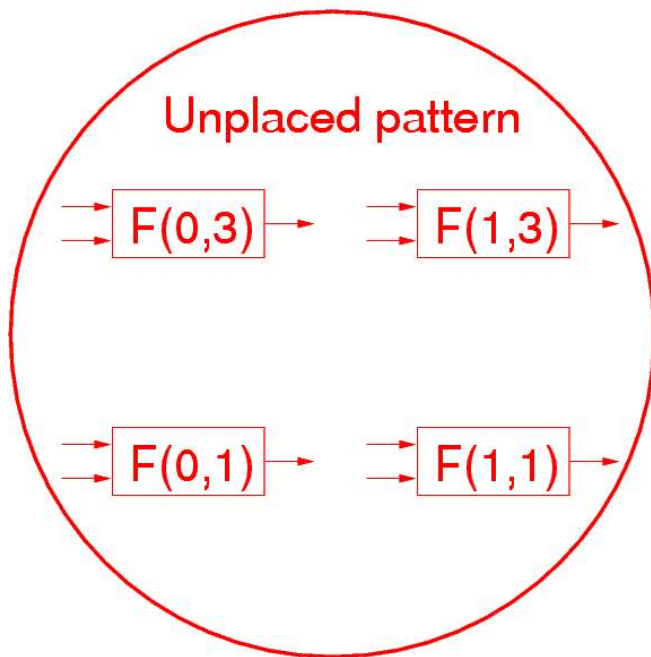
- Full application and reconfigurable architecture



Characteristics of reconfigurable architecture



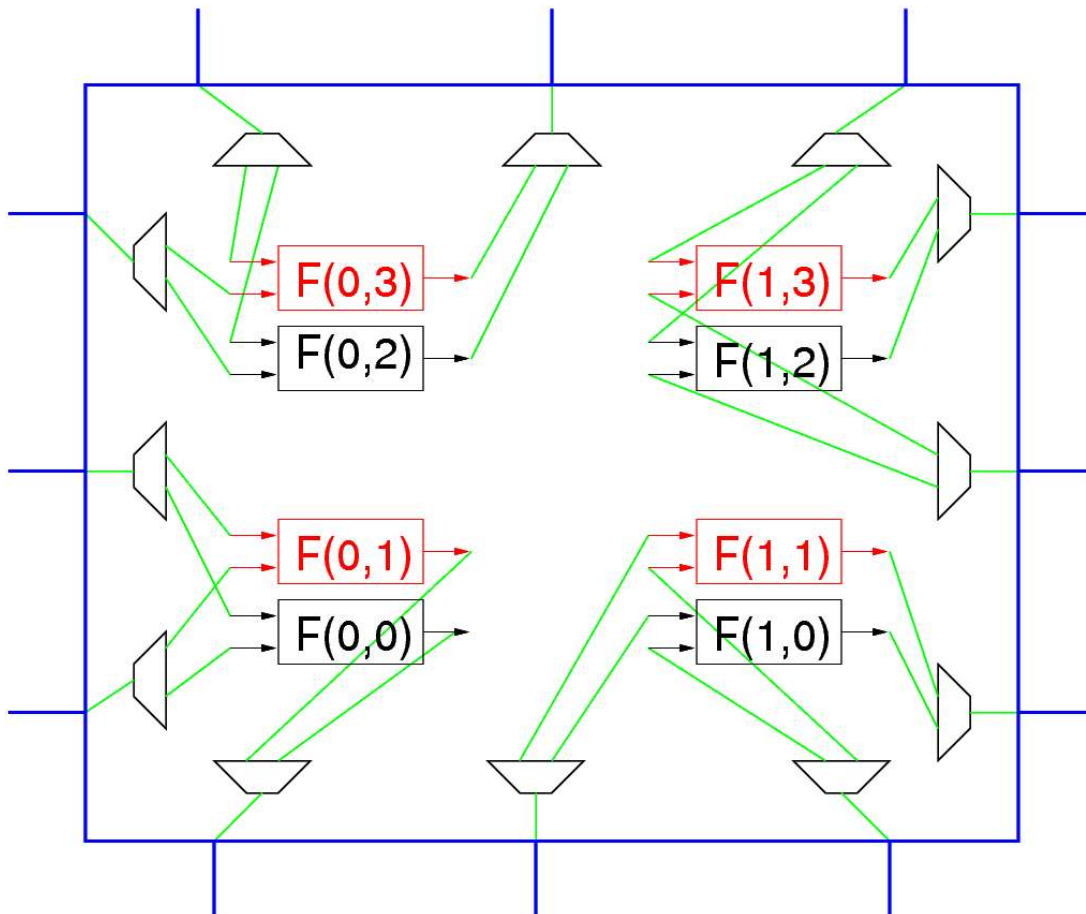
- Simple mapping of application on architecture



Characteristics of reconfigurable architecture

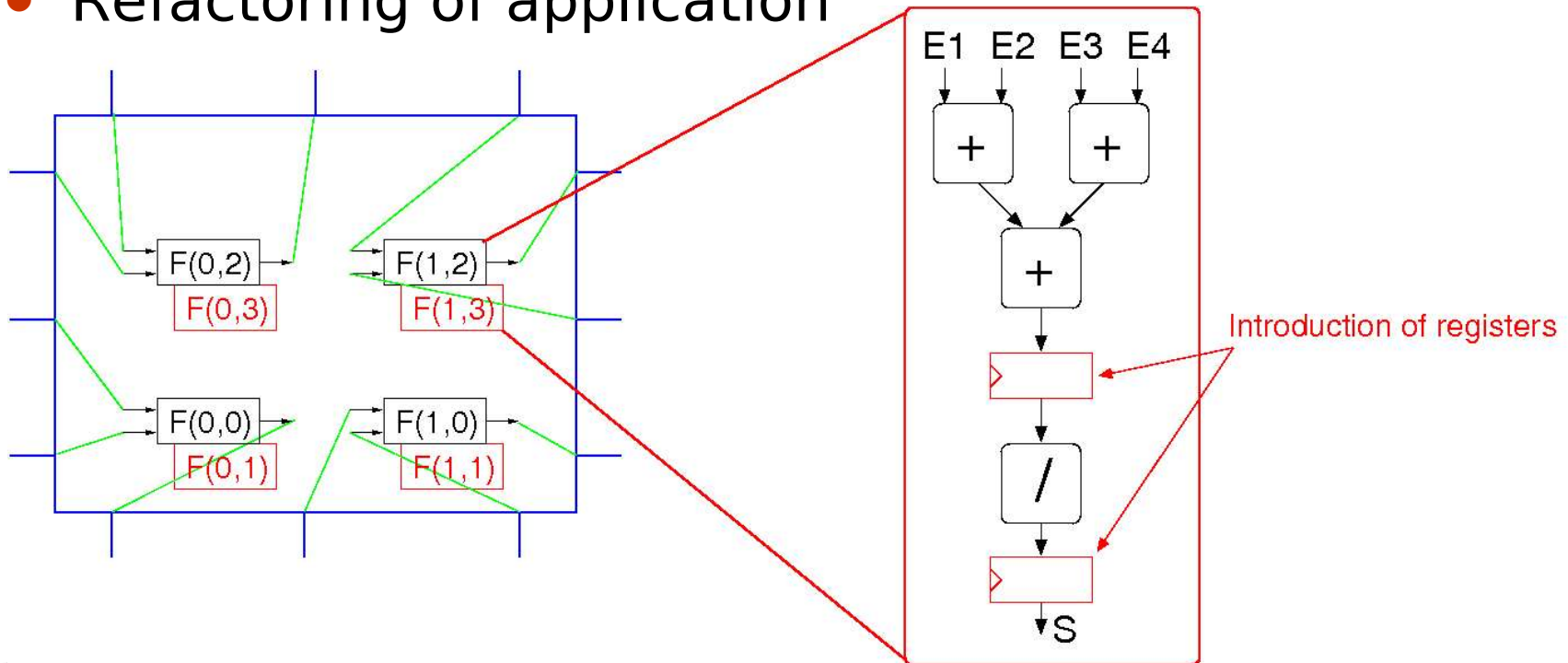


- Use of multiplexers



Pipelined application

- Limited number of dedicated pins on FPGA
- Introduction of pipelined execution
 - Registers
 - Refactoring of application



- A tool to implement application described in BLIF format

Implementation with an existing tool: Madeo



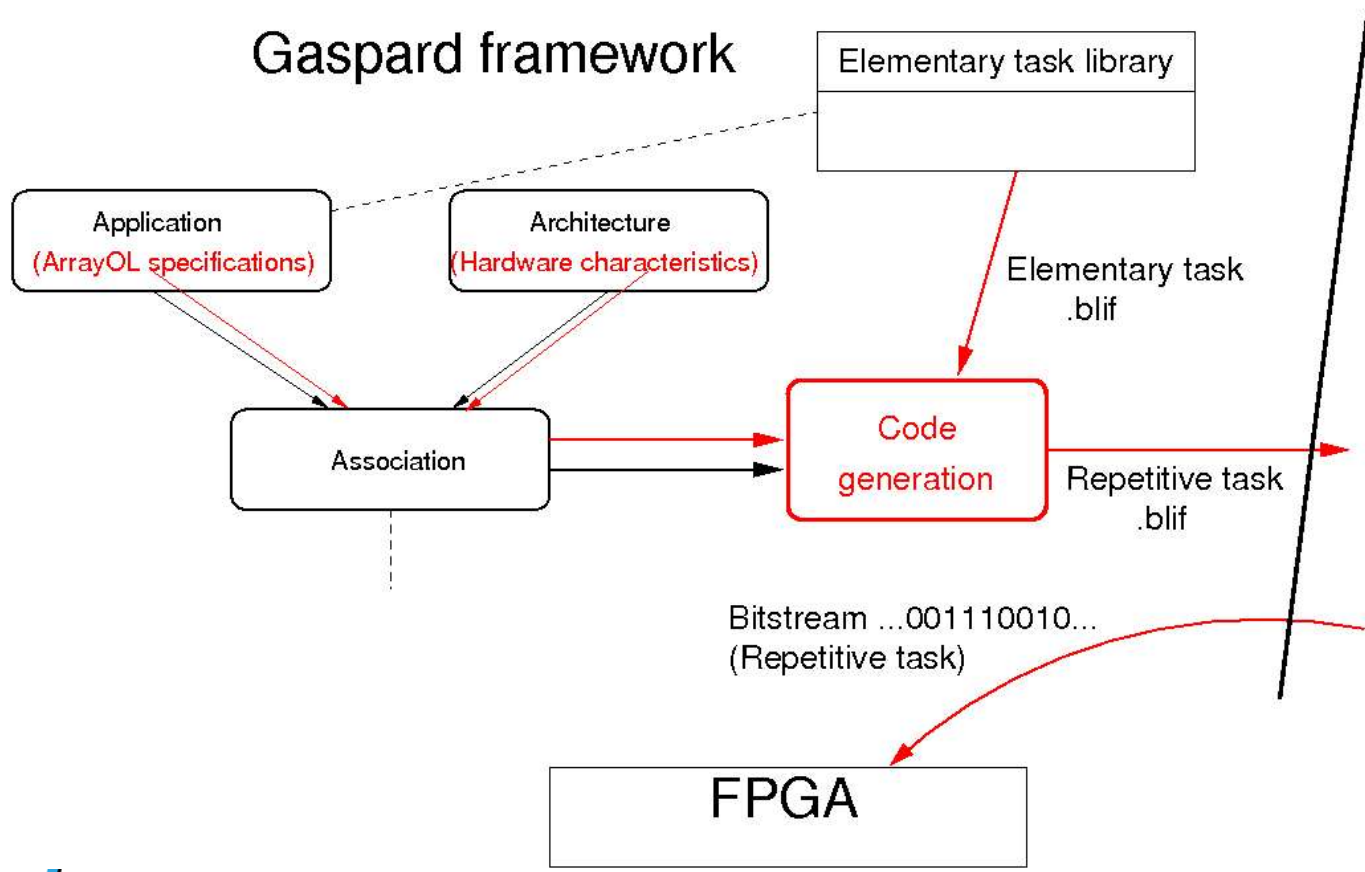
- Pros:
 - Modelling an application
 - Modelling a reconfigurable architecture
 - Implementation of application on architecture
- Cons:
 - Low level description of architecture and application
 - Slow optimisation process
 - *Only on elementary components*
 - No repetitive task
 - *Introduced in BLIF*

Global flow of integration



Gaspard framework

Madeo framework



- Conclusion and perspectives

Conclusion



- Description of application structure in high level modelling
- Generation of code implementable on FPGA
- Take into account reconfigurable architecture characteristics
- Reduce optimisation time of applications

Perspectives



- Creation of a BLIF library of elementary components
- Complete the implementation (code generation)
- Introduction of pipeline specification