FACULTY OF ENGINEERING AND ARCHITECTURE

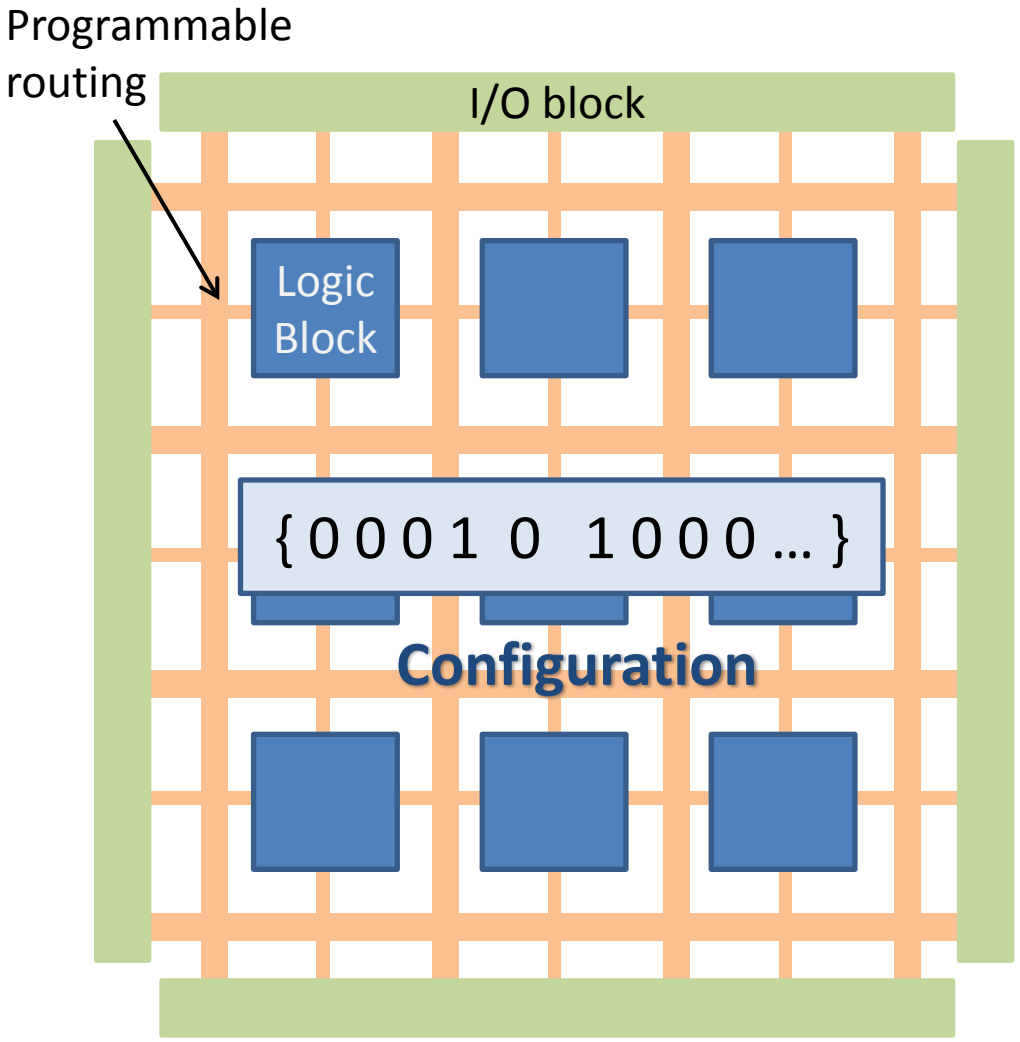
Maximizing Routing Resource Reuse in a Reconfiguration-aware Connection Router for FPGAs

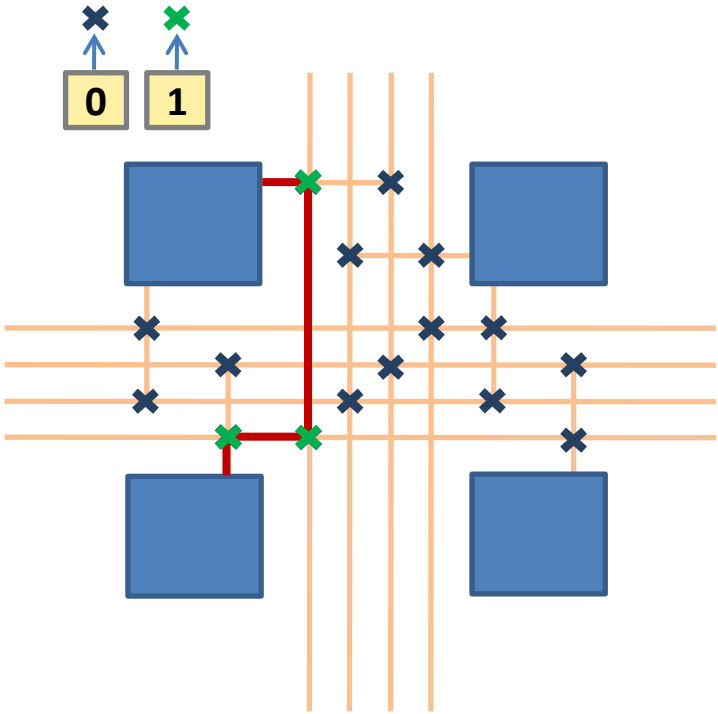
Elias Vansteenkiste

Karel Bruneel and Dirk Stroobandt

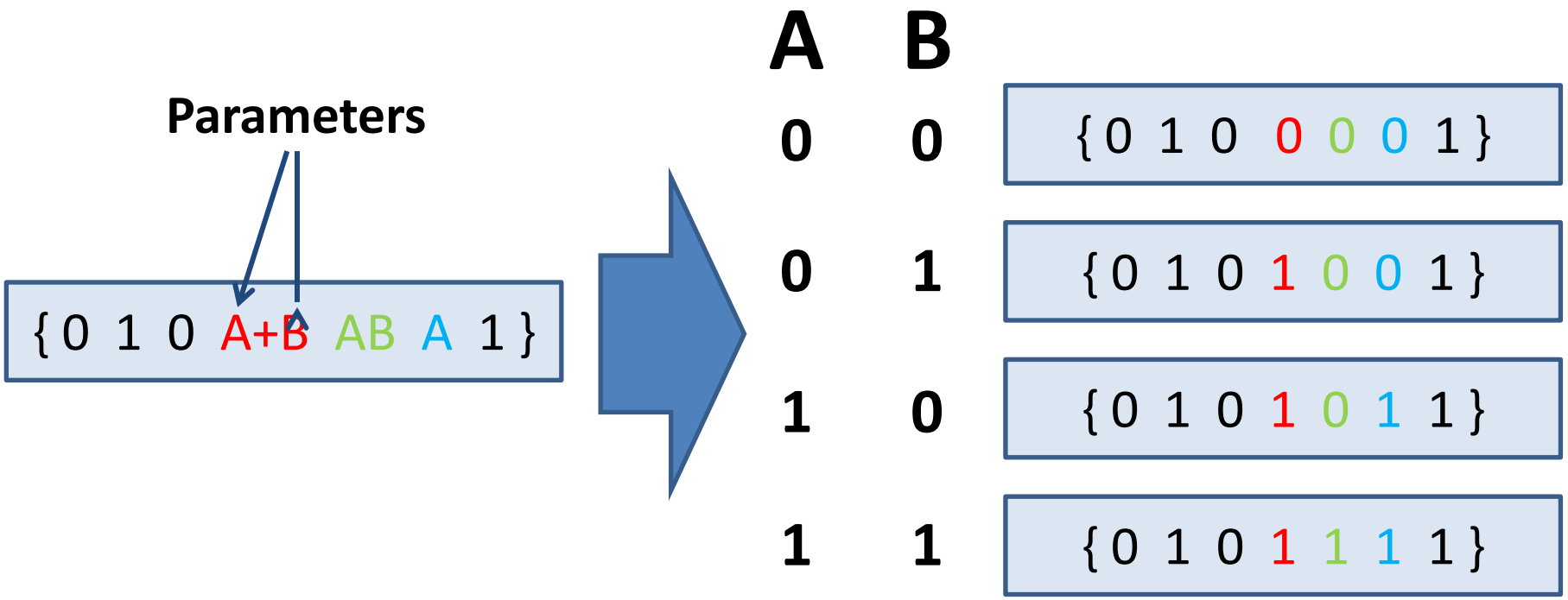
Elias.Vansteenkiste@UGent.be

FPGA configuration





Parameterized Configuration



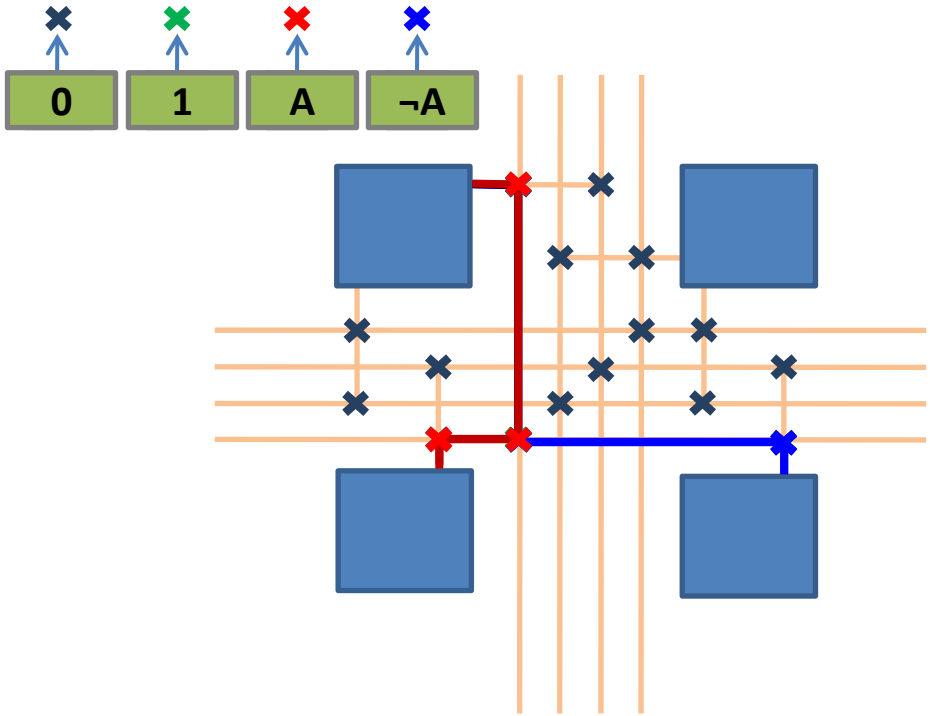
* K. Bruneel and D. Stroobandt, "Automatic Generation of Run-time Parameterizable Configurations," FPL 2008.

Applications

- Dynamic Circuit Specialization
  - Circuit optimization (smaller, faster, …) using run-time reconfiguration
- Circuit Specialization
  - Hard coded settings of devices

Very fast generation of specialized configurations

What's new in this work



Tuneable Connections (TCON)

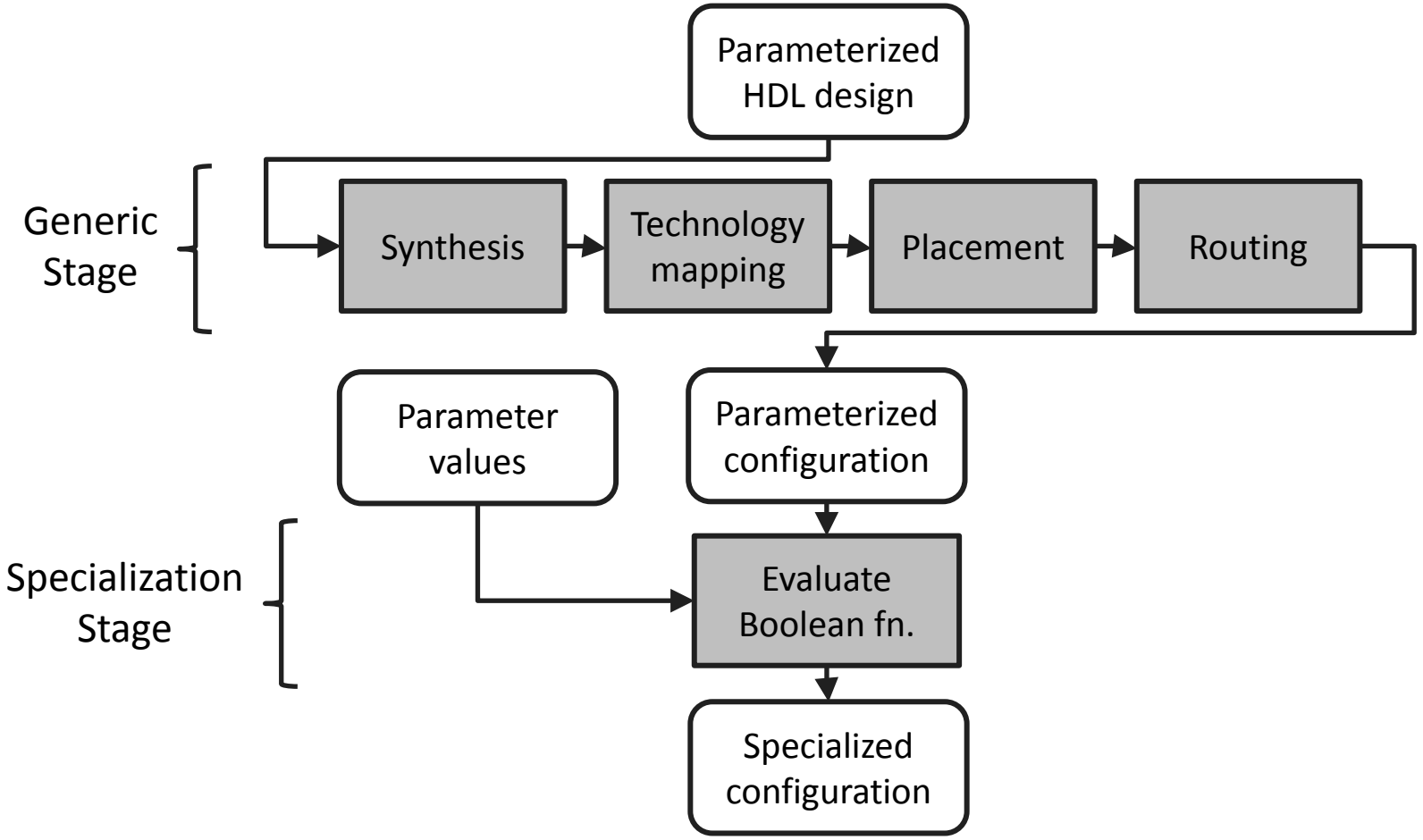
Static Connections

Routing TCONs,
while maximizing the reuse of routing resources

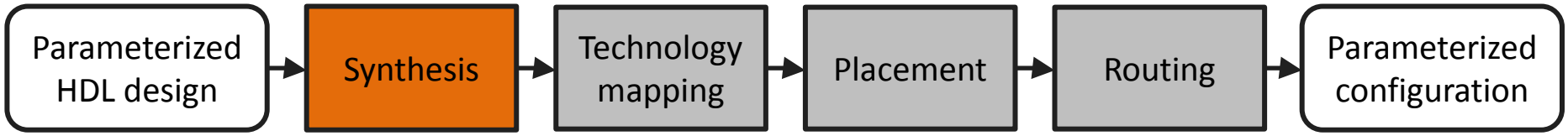
Outline

- FPGA configuration
- Applications
- What's new in this work
- Toolflow
- Router
- Experimental Case Study
- Conclusion and Future work

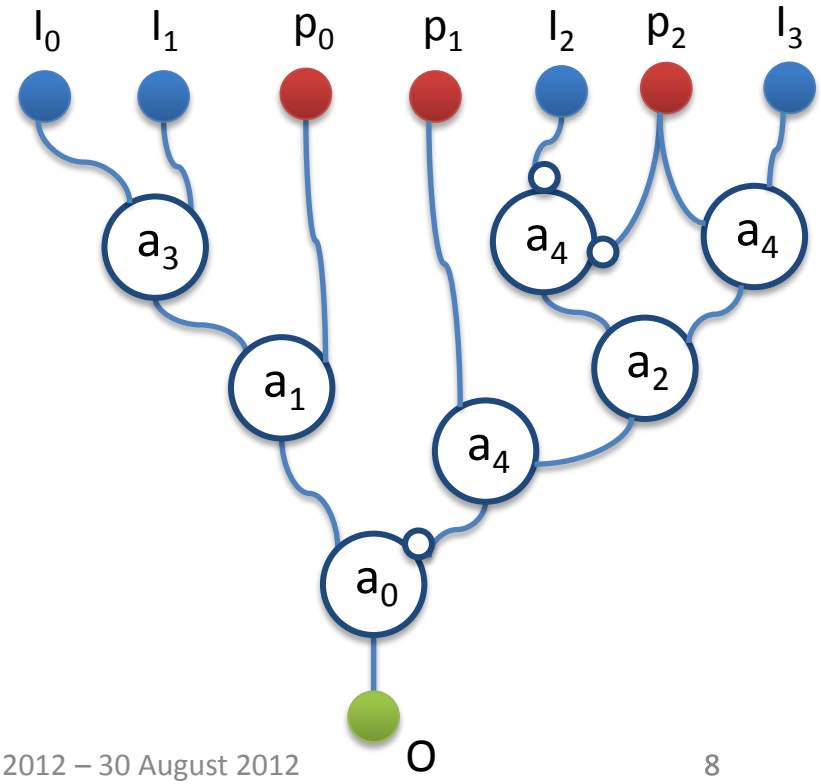
Toolflow


Parameterized HDL design
Generic Stage
Synthesis
Technology mapping
Placement
Routing
Parameter values
Parameterized configuration
Specialization Stage
Evaluate Boolean fn.
Specialized configuration

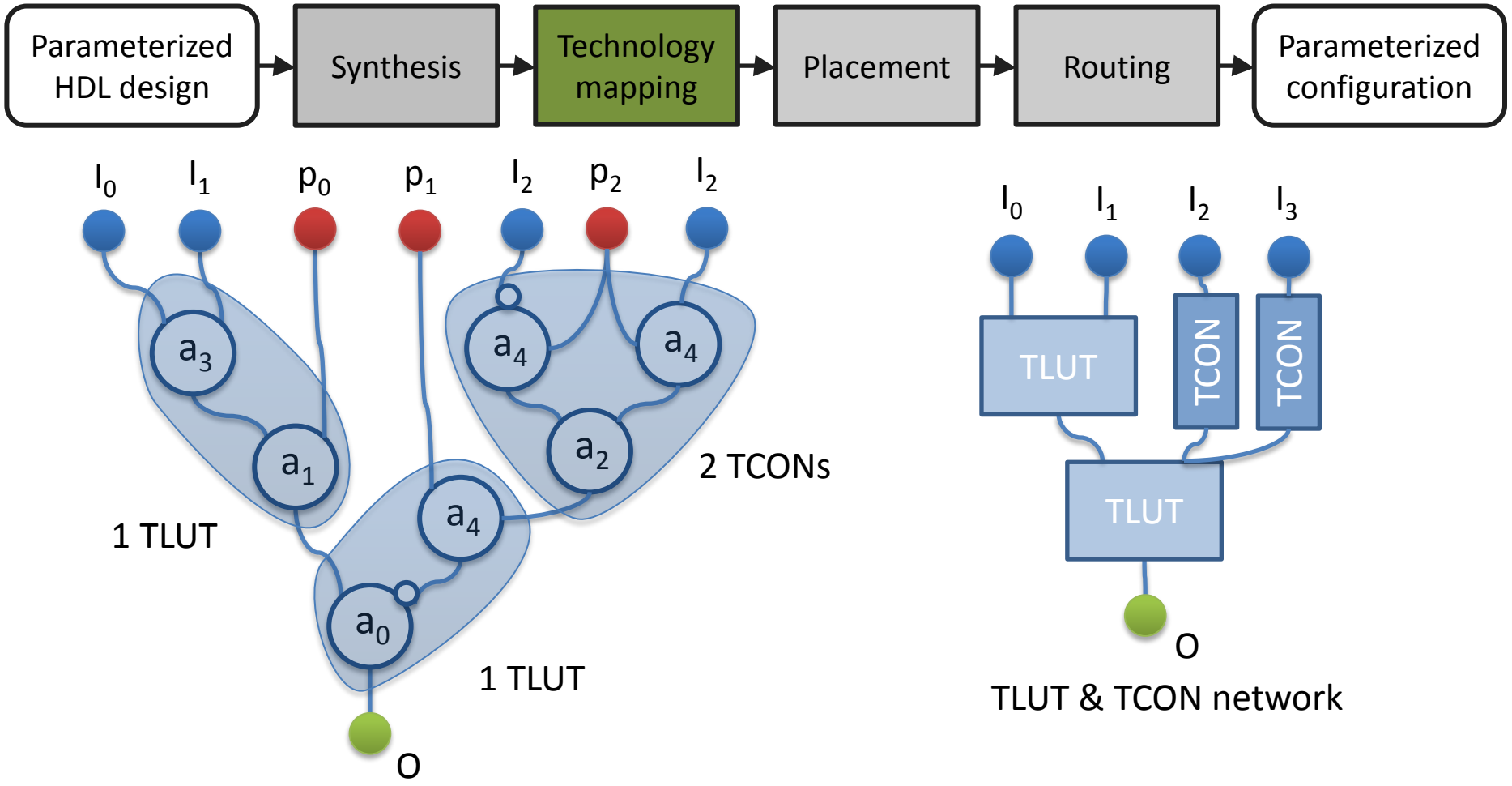
Generic Stage of the Tool Flow



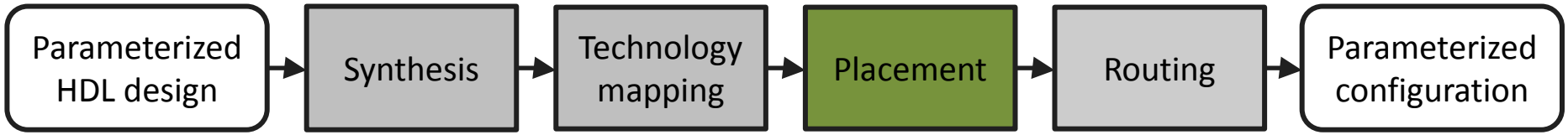

```
entity multiplexer is
port(
  --BEGIN PARAM
  sel : in  std_logic_vector(2 downto 0);
  --END PARAM
  in  : in  std_logic_vector(3 downto 0);
  out : out std_logic
);
end multiplexer;

architecture behavior of multiplexer is
begin
  out <= in(conv_integer(sel));
end behavior;
```

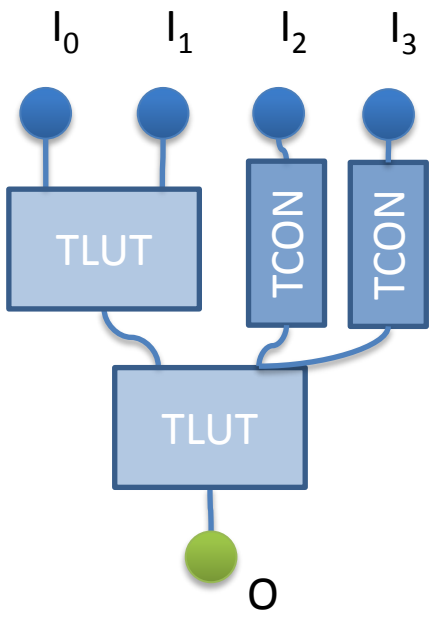


Generic Stage of the Tool Flow

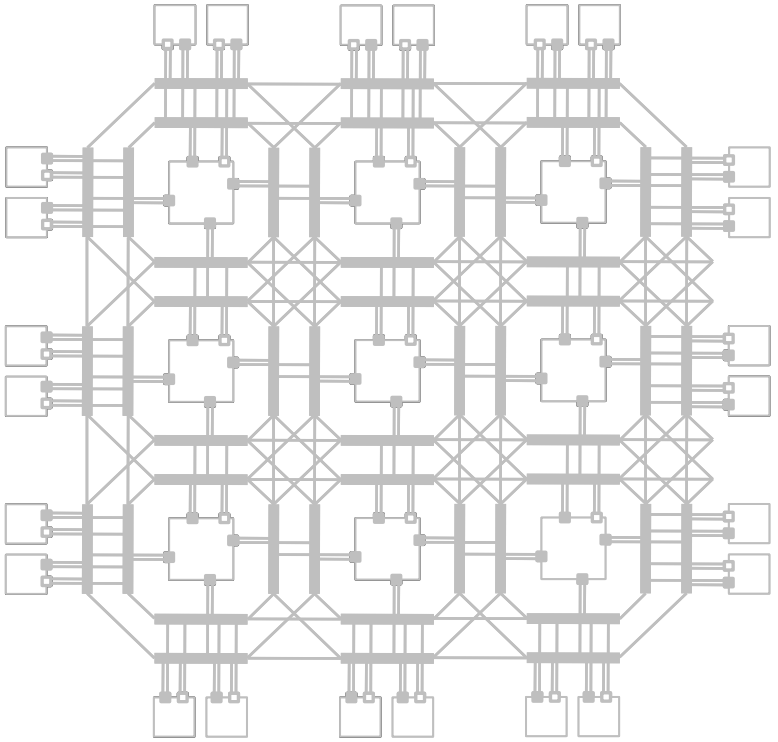


Generic Stage of the Tool Flow

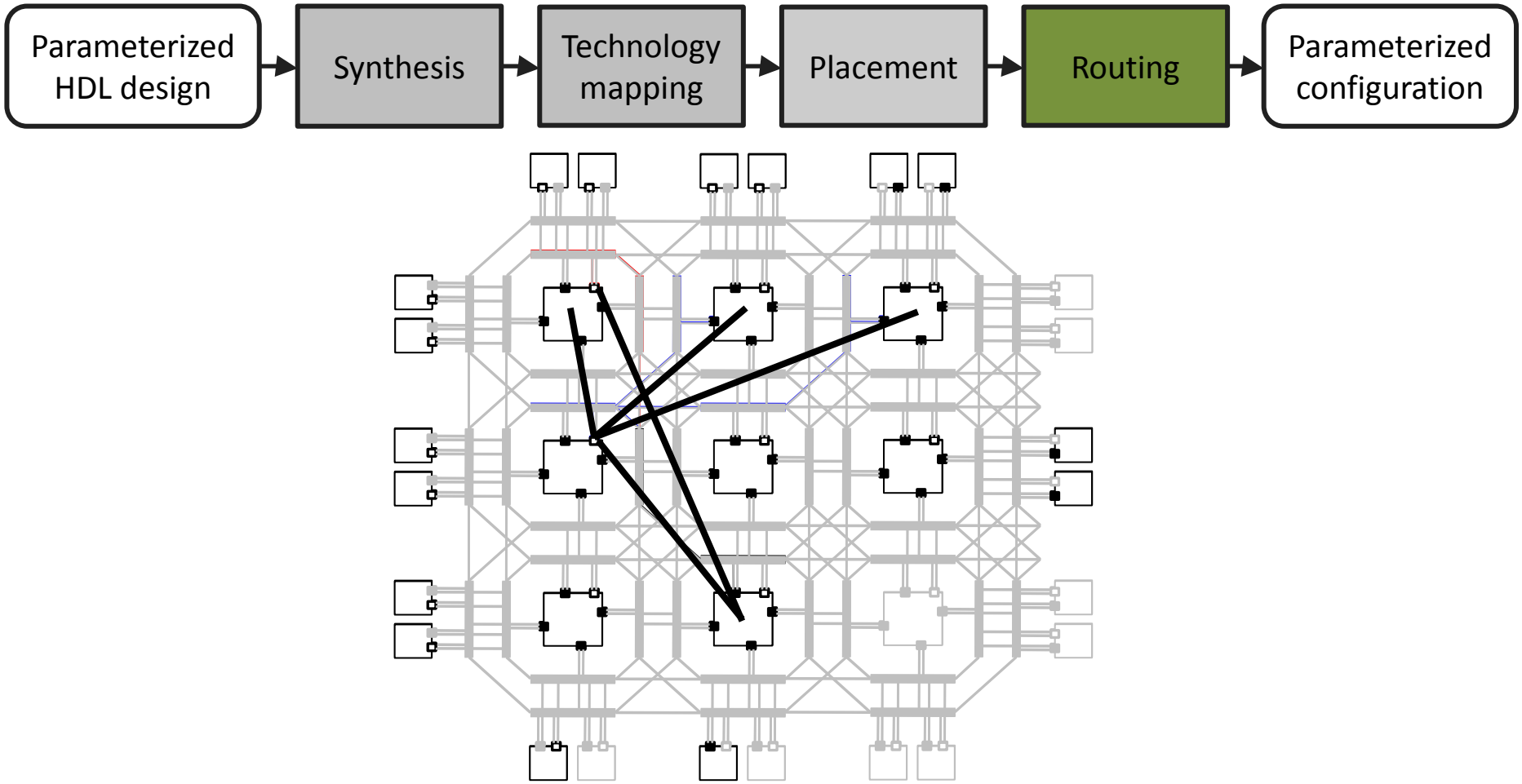




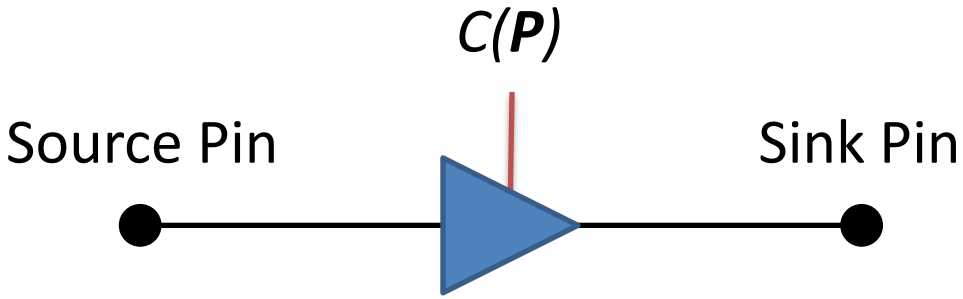
TLUT & TCON network



Generic Stage of the Tool Flow



TCON - Tunable Connection

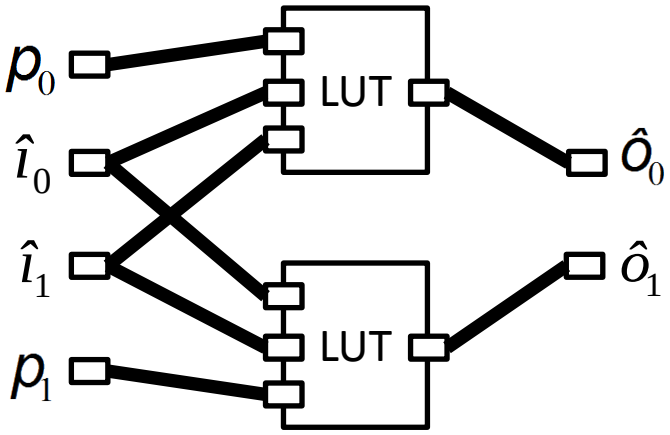


*C(**P**)* - Connection condition, Boolean function of ***P***
If *C(**P**)* is true, then connection is active

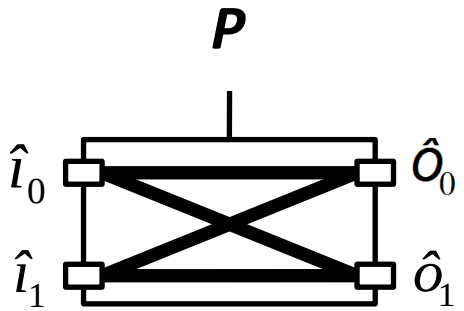
- Generated by Technology Mapping
- Endpoints are fixed after Placement
- Router reserves the wires to realize the connection, when *C(**P**)* is true

Example: 2 x 2 Crossbar

- Conventional implementation:
  - 2 LUTs
  - 6 connections



- TCON implementation:
  - 0 LUTs
  - 4 TCONs



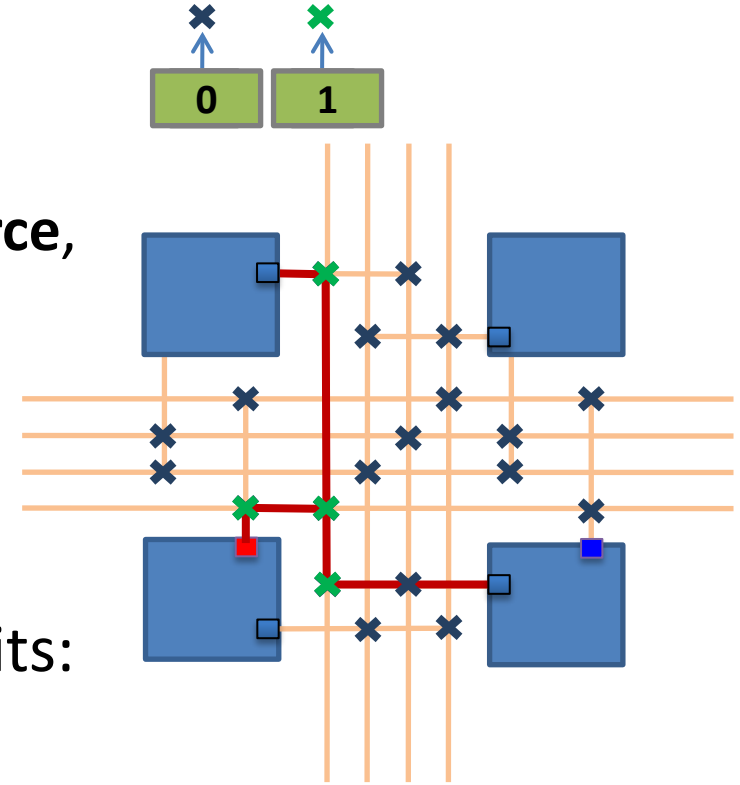
Router: Problems

Goal: Realize TCONs, minimizing the necessary resources?`

Problem 1:

Circuits with static connections:

- Connections ζ_1 and ζ_2 **with same source**,
  -> carry the same signal
  -> may share resources
  -> bundled in a net



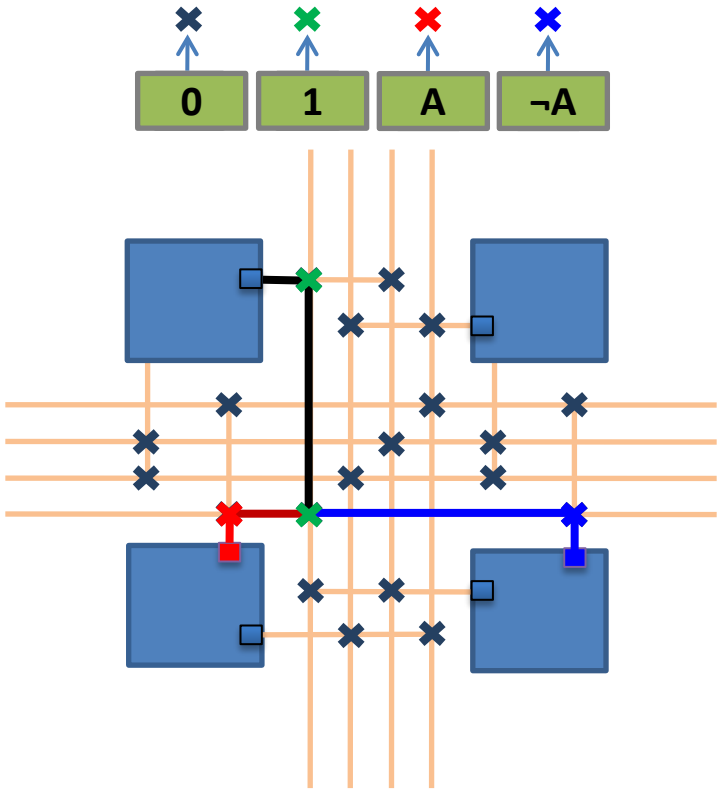
Conventional Solution for static circuits:

Pathfinder

Router: Problems

Circuits with TCONs:

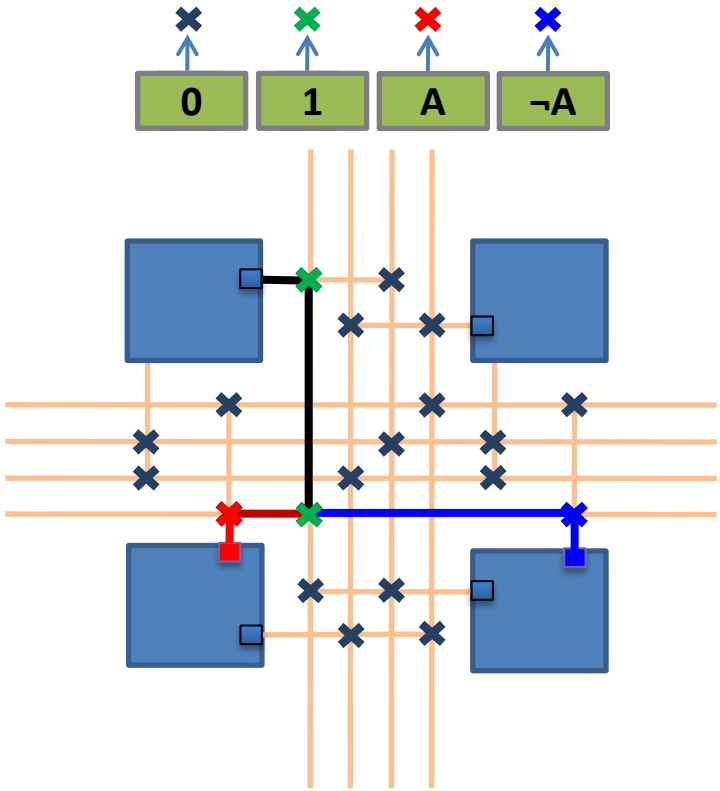
- TCONs ζ_1 and ζ_2 **with same source**, carry same signal, may share resources
- TCONs ζ_1 and ζ_2 **not active at same time,** may share resources



Pathfinder **not applicable**

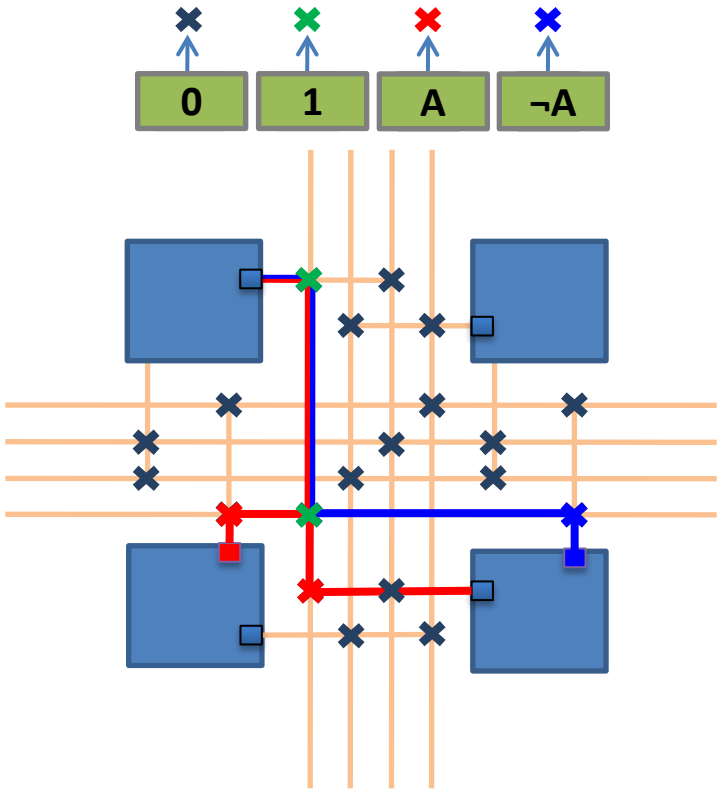
Bundle Connections

- Problem 2:
  Comparing Connection Conditions in the kernel
  Computationally Not Feasible
- Bundle connections that may share routing resources before routing



Router: Problems

- Problem 3:
  Resource Sharing Stimulation
- Conventional routing:
  Zero cost stimulation
- Not possible in case of TCONs:
  Two resource sharing possibilities
- Solution:
  Both types of legal resource sharing are allowed
  Only the largest bundle is stimulated



Experimental Case Study: Clos Networks

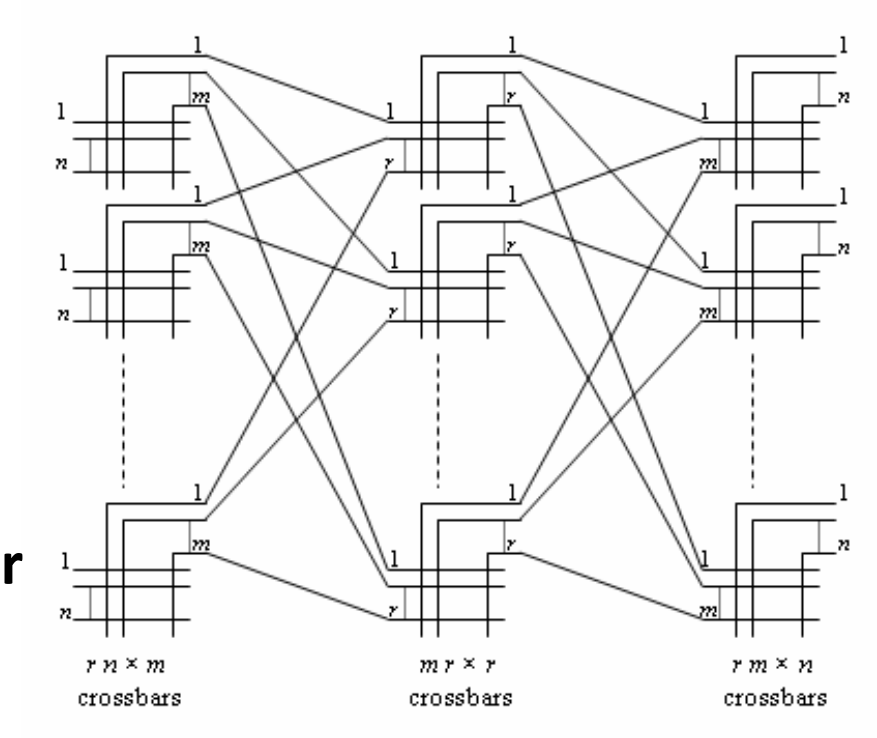
Multistage Circuit Switching Network

Three stages

- Ingress
- Middle
- Egress

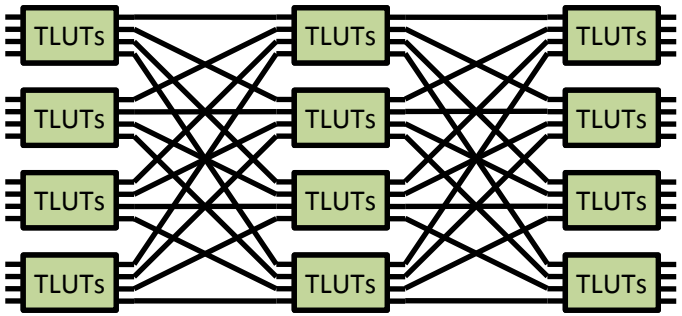
Each stage built up of crossbars

Advantage:
the number of crosspoints **fewer** compared to one large crossbar

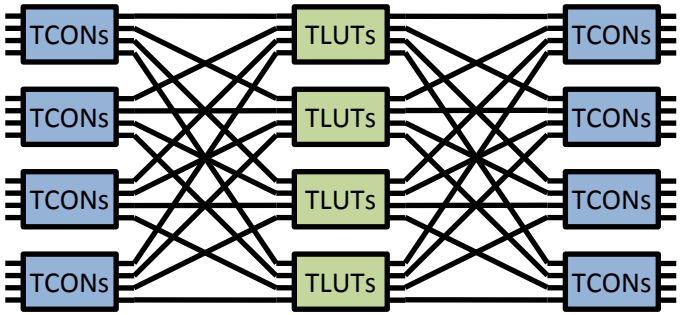


Clos Network Implementations

- Stages are built up of 4 x 4 Crossbars
- 3 Sizes:
  - 16 x 16
  - 64 x 64
  - 256 x 256
- 3 types of implementation:
  - Conventional
  - TLUT
  - TCON (Combined)

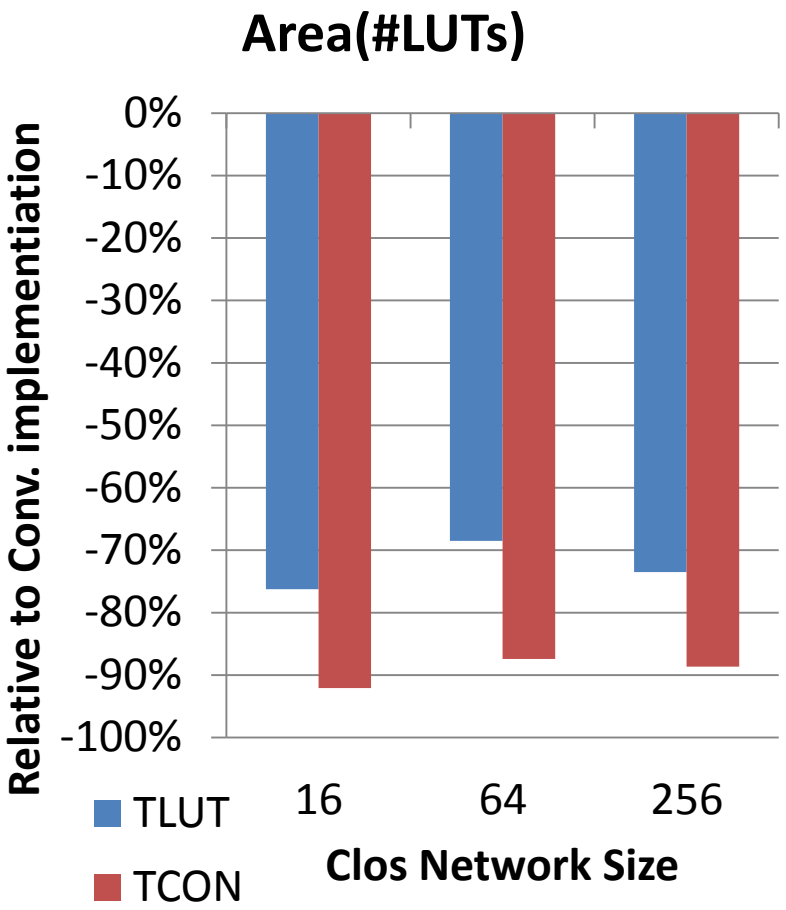


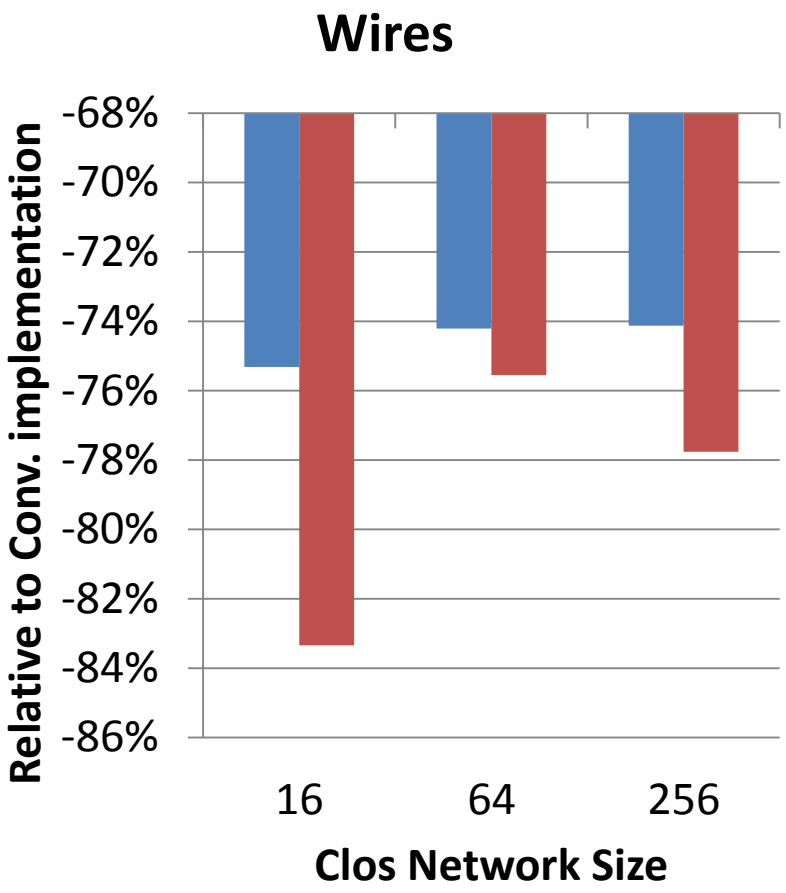
TLUT implementation



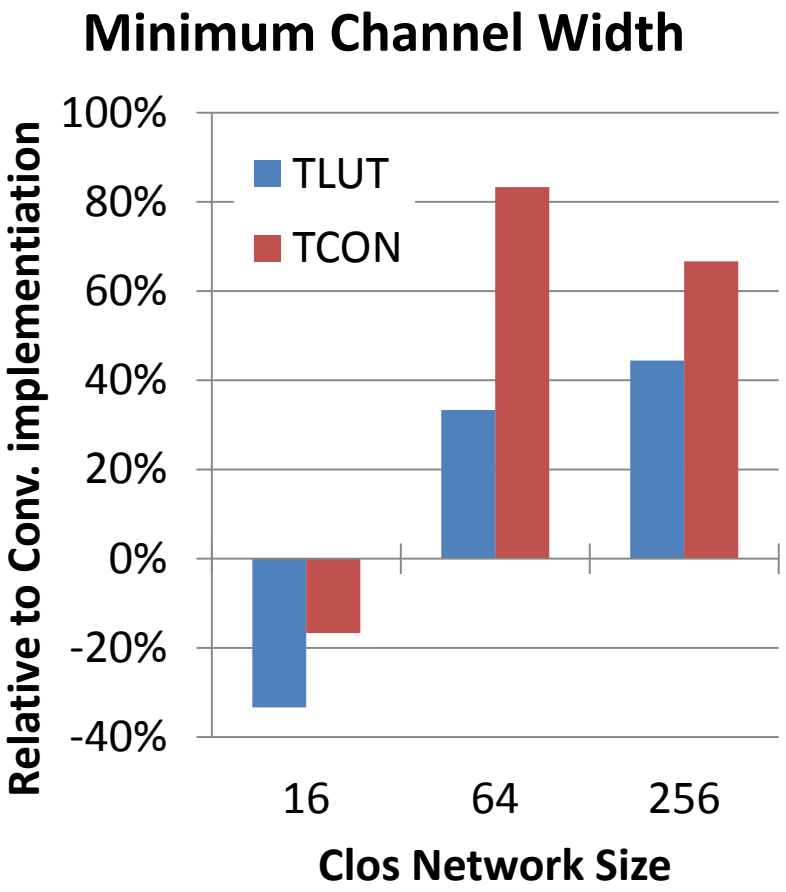
TCON implementation

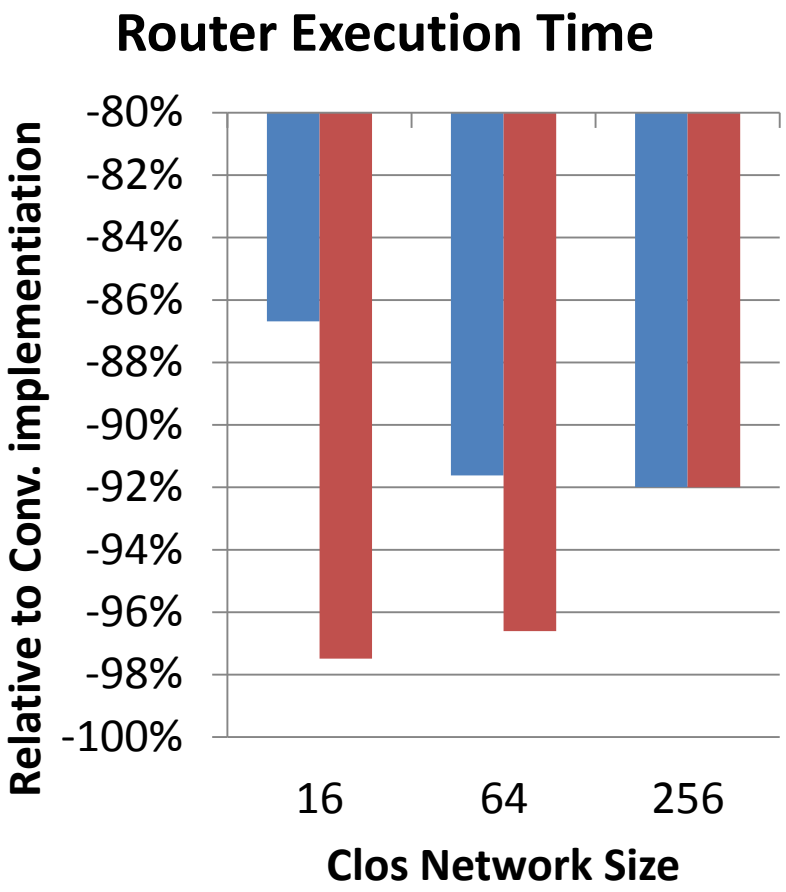
Area


Area(#LUTs)
Relative to Conv. implementation
0%
-10%
-20%
-30%
-40%
-50%
-60%
-70%
-80%
-90%
-100%
16
64
256
Clos Network Size
TLUT
TCON


Wires
Relative to Conv. implementation
-68%
-70%
-72%
-74%
-76%
-78%
-80%
-82%
-84%
-86%
16
64
256
Clos Network Size

Minimum Channel Width and Compilation time





Conclusion

- Routing algorithm for parameterized interconnections (Tunable Connections)
- Saves area (LUTs and wires) on FPGA
- Reduces logic depth
- Reduces execution time

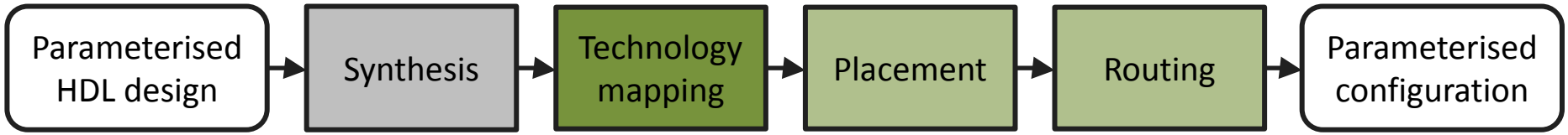
Future Work

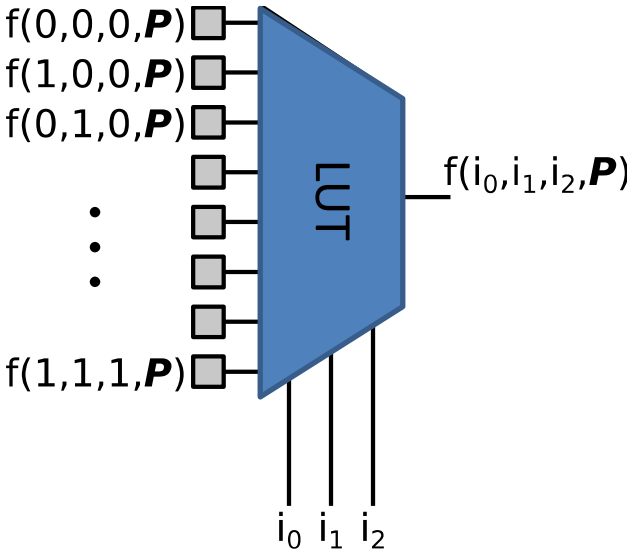
- Pack and Place algorithms
- Demonstrate Dynamic specialization of the FPGA's interconnect network on contemporary FPGAs
  - Support for Heteregenuous FPGAs: Xilinx's Virtex-V
  - Complex routing architectures
  - Fully elaborated examples
  - Rapid Smith

Acknowledgement

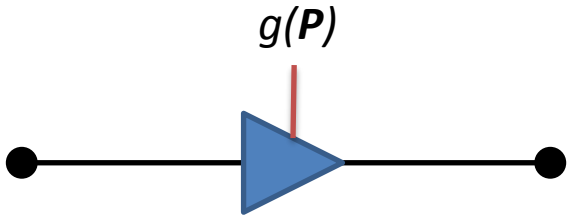
- Faster European project
- Phd. Grant, IWT,
  Flemish Agency for Innovation through Science and Technics

Toolflow





Tuneable LUT
TLUT



Tuneable Connection
TCON

Router: Algorithm

while congestedResourcesExist():
    for each bundle β in tunable circuit **do**:
        for each connection ζ in β **do**:
            ζ.ripUpRouting()
            ζ.path = Dijkstra(ζ.source, ζ.sink)

    ζ.resources.updateCongestionCost()
    allResources.updateHistoryCost()