



AN AREA-EFFICIENT PARTIALLY RECONFIGURABLE CROSSBAR SWITCH WITH LOW RECONFIGURATION DELAY

Hoo Chin Hau and Akash Kumar

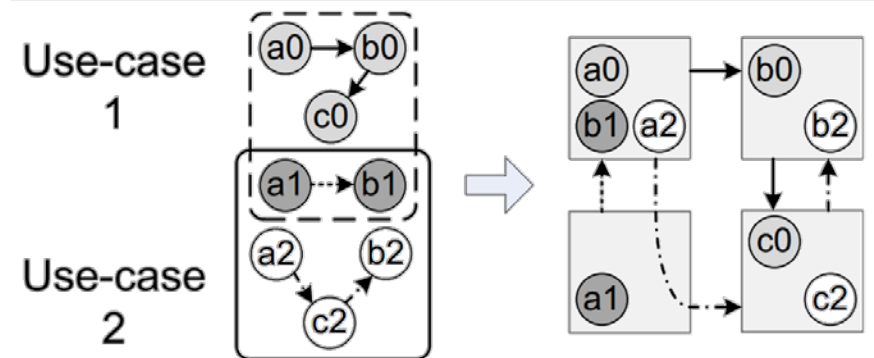
1

MOTIVATION

- Multiprocessors on the rise (MPSoCs)
 - Single processor systems unavailable to meet application performance requirements
 - Multiple processors + Memory + IO devices
 - Bottleneck shifts from computation to communication
- Networks-on-Chip (NoCs) proposed
 - Bus is not scalable

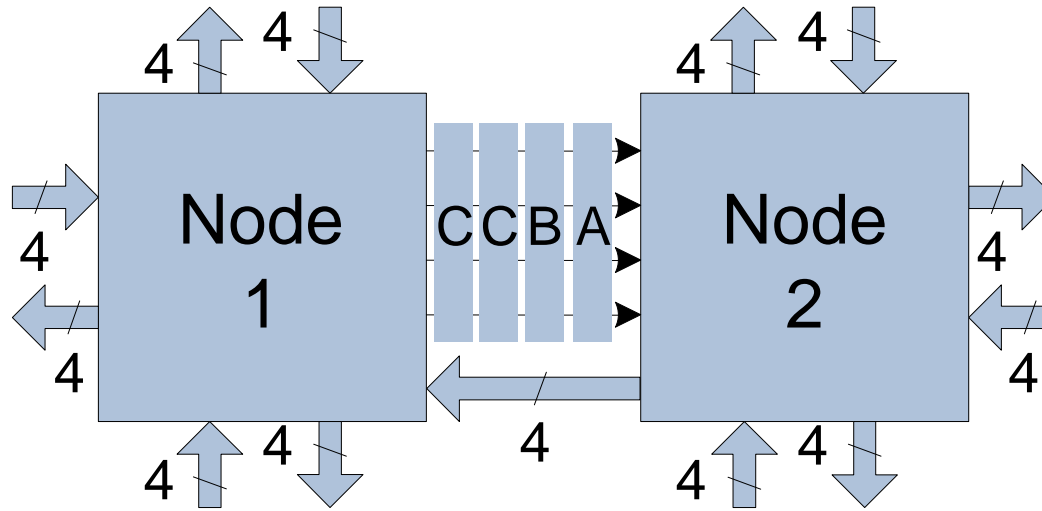
MOTIVATION

- MPSoCs need to handle multiple use cases – combinations of multiple applications
- Use cases have different connectivity requirements
- NoCs have to be dynamically reconfigurable and provide guaranteed throughput: TDM vs SDM

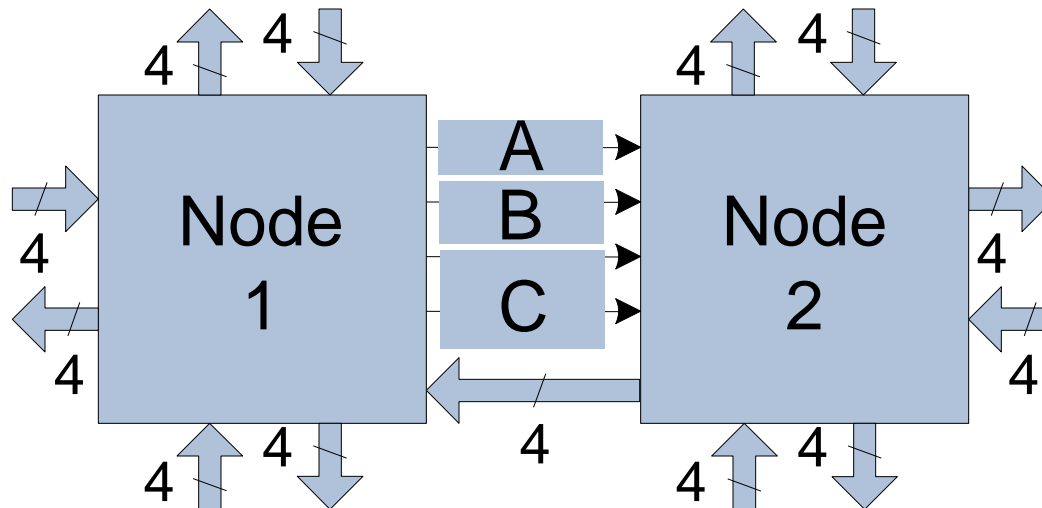


TDM vs SDM

TDM



SDM



MOTIVATION

- Two approaches to dynamic reconfiguration of NoC
 - Adding reconfiguration logic
 - Incurs area overhead
 - Partial reconfiguration (PR)
 - High reconfiguration delay
 - Requires large storage space
 - Caters to predefined use cases only

CONTRIBUTIONS

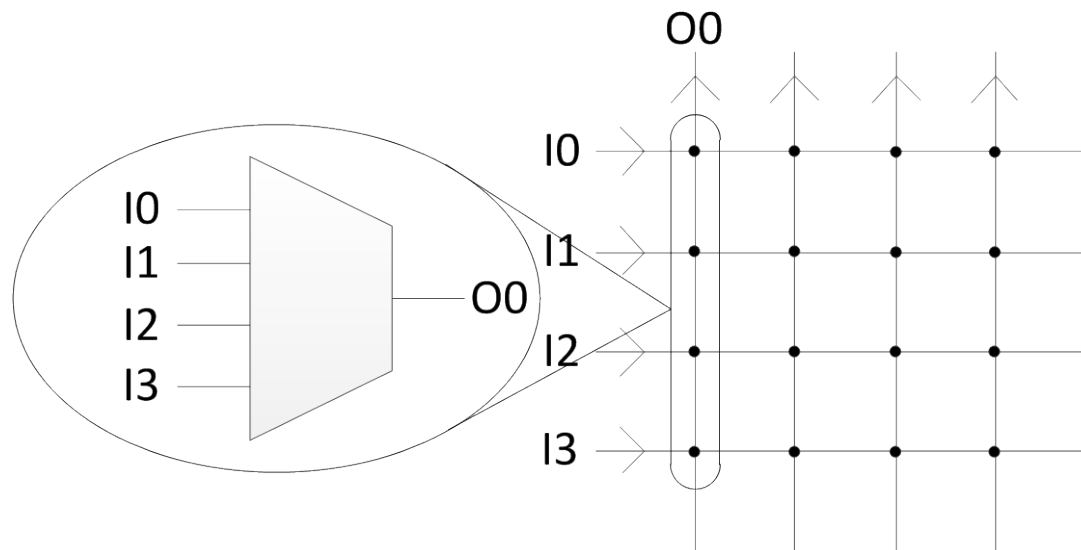
- Problem lies with the design of crossbar switch
- A novel partially reconfigurable crossbar switch design is proposed
 - 84% area saving!
 - 78% reconfiguration delay reduction!
 - Runtime bit-stream generation
 - Glitch-free reconfiguration

The slide features a dark blue background with a decorative left side consisting of vertical stripes in various shades of blue and white. Several blue circles of different sizes are scattered on the left side, with one large circle at the top and several smaller ones below it.

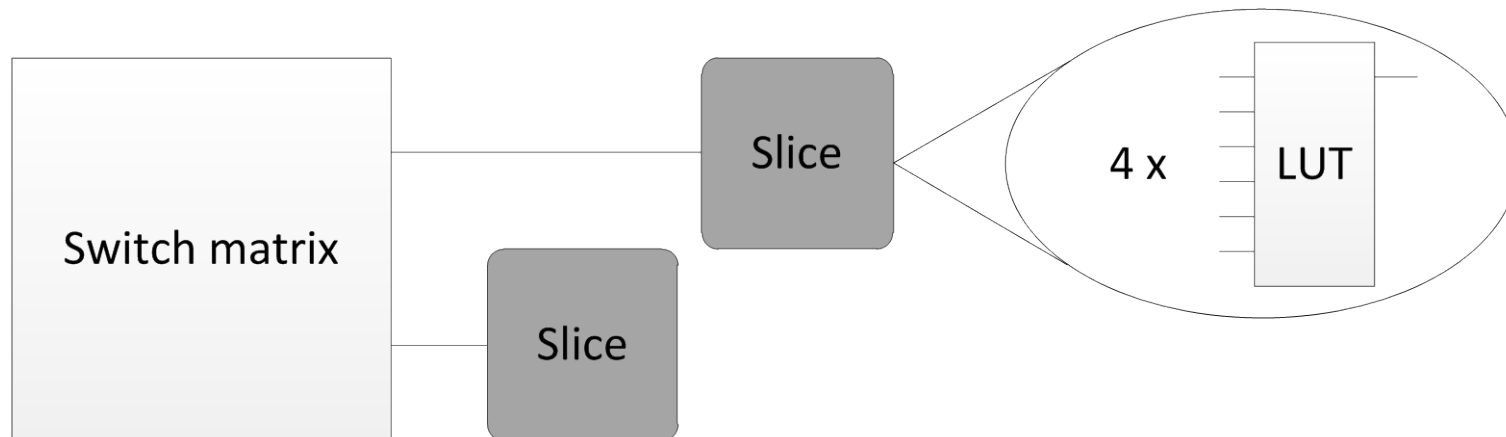
CROSSBAR SWITCH ARCHITECTURE

7

MULTIPLEXER AS BUILDING BLOCK OF CROSSBAR SWITCH



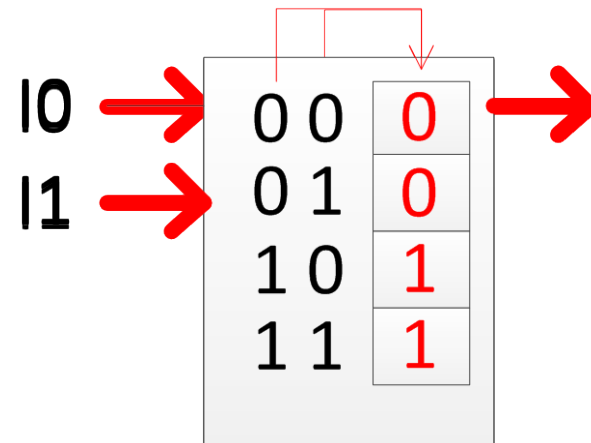
A CONFIGURABLE LOGIC BLOCK (CLB)



LUT AS MULTIPLEXER

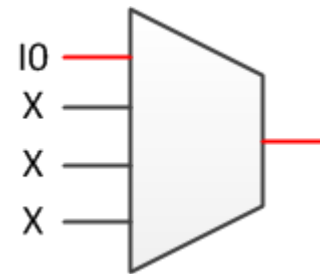
- Conventional LUT multiplexer
 - Requires $\log_2 N$ selector pins for N data inputs
- PR based LUT multiplexer
 - Requires no dedicated selector pins
 - Selection is done by changing LUT content through PR
 - Allows larger multiplexer to be built with the same number of LUTs

LUT AS MULTIPLEXER

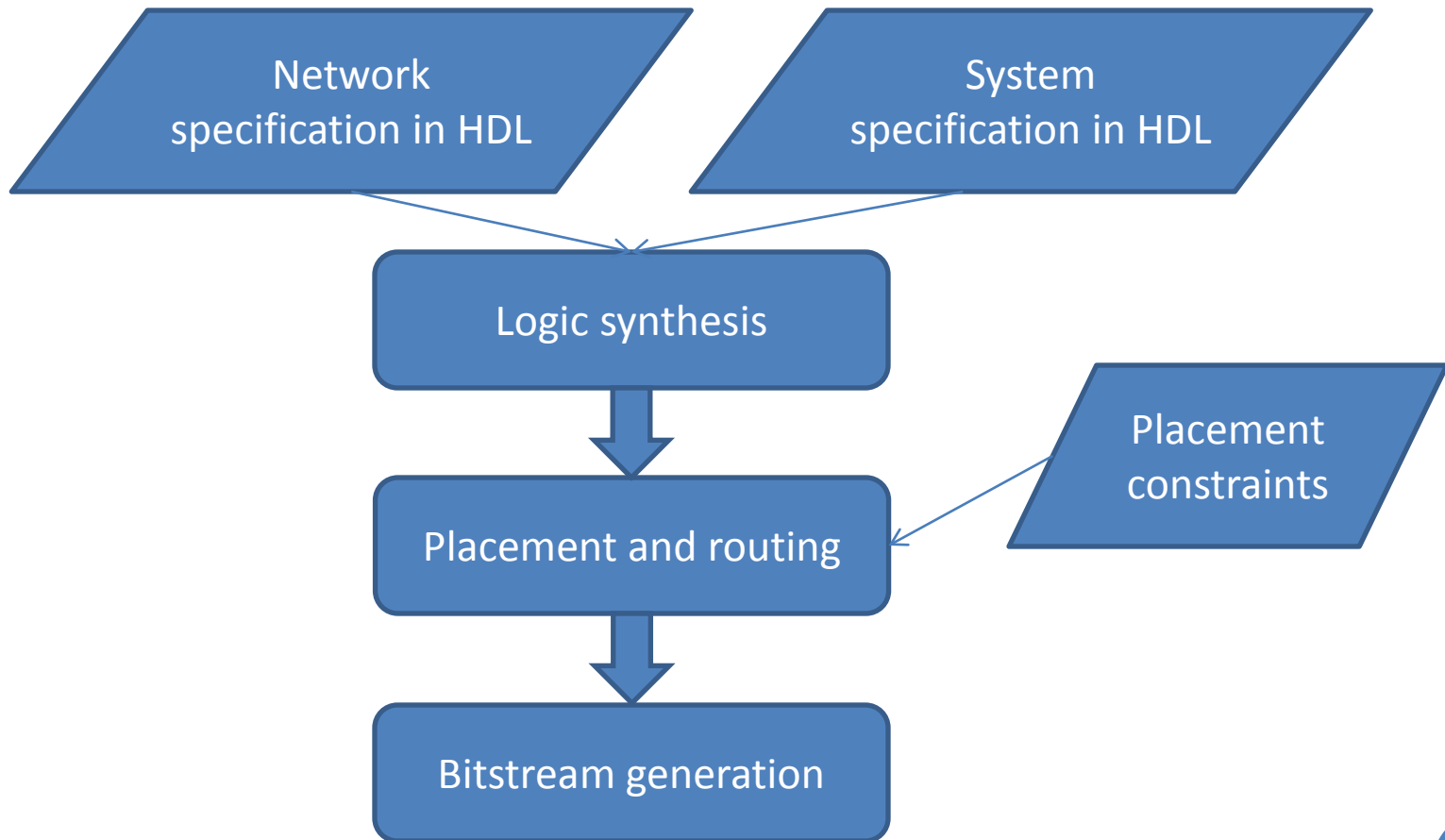


SCALABLE APPROACH

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

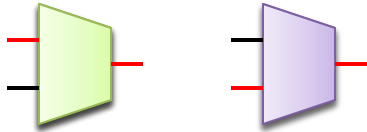


DESIGN FLOW

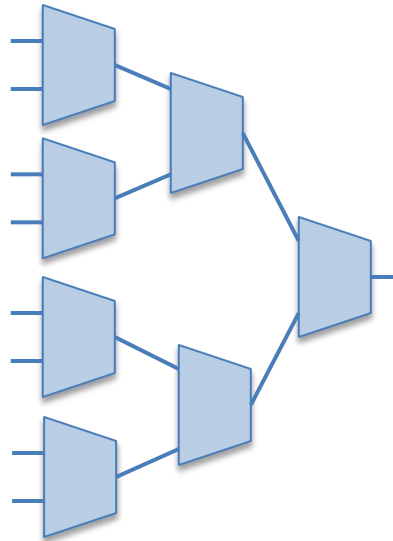


RUNTIME BITSTREAM GENERATION & RECONFIGURATION

LUT configuration database



ICAP

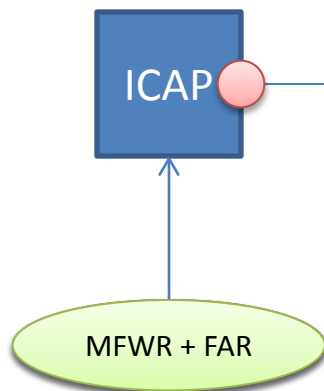
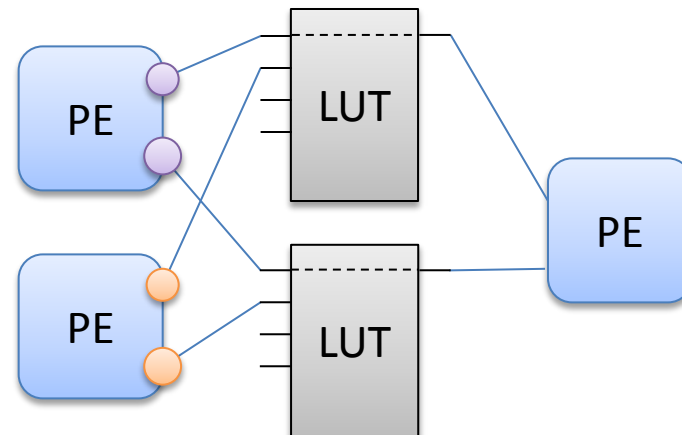
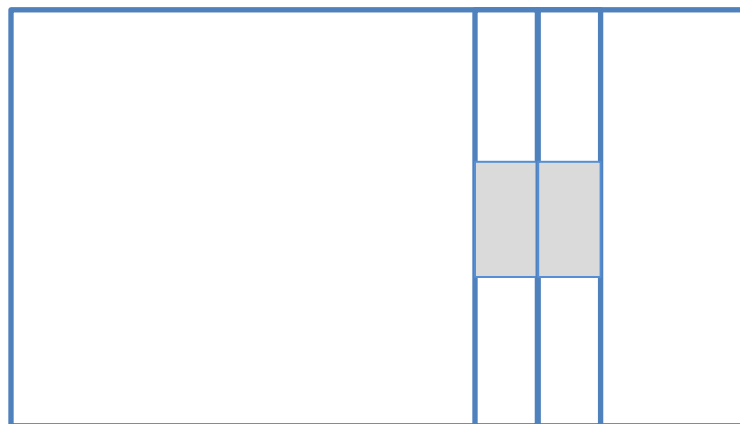


MINIMIZING RECONFIGURATION DELAY

- Apply AREA_GROUP constraint to limit the placement of LUT to the minimum number of CLB columns required
- Identify the frames that are responsible for LUT content
 - 8 out of 36 frames are required
 - 78% speedup!
- Use the Multiple Frame Write (MFW) command of ICAP

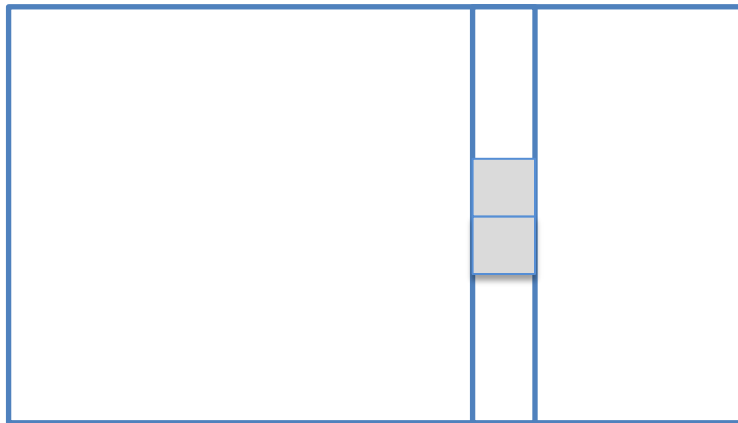
MULTIPLE FRAME WRITE (MFW)

2 LUTs configured **identically** by transferring the configuration data only **once**!



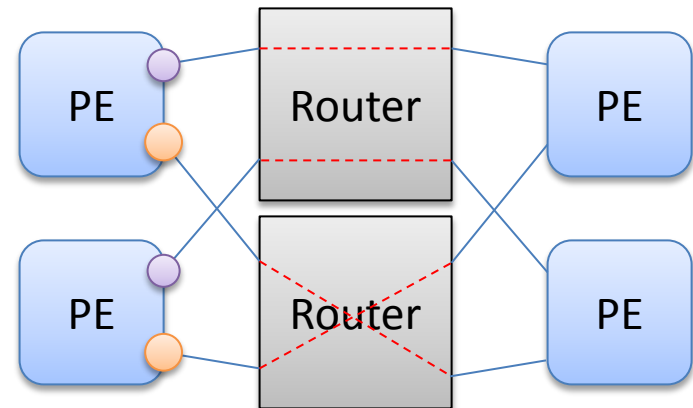
GLITCH-FREE CONNECTION ESTABLISHMENT

Configuration of first router is unchanged



ICAP

Operation of first router is unaffected

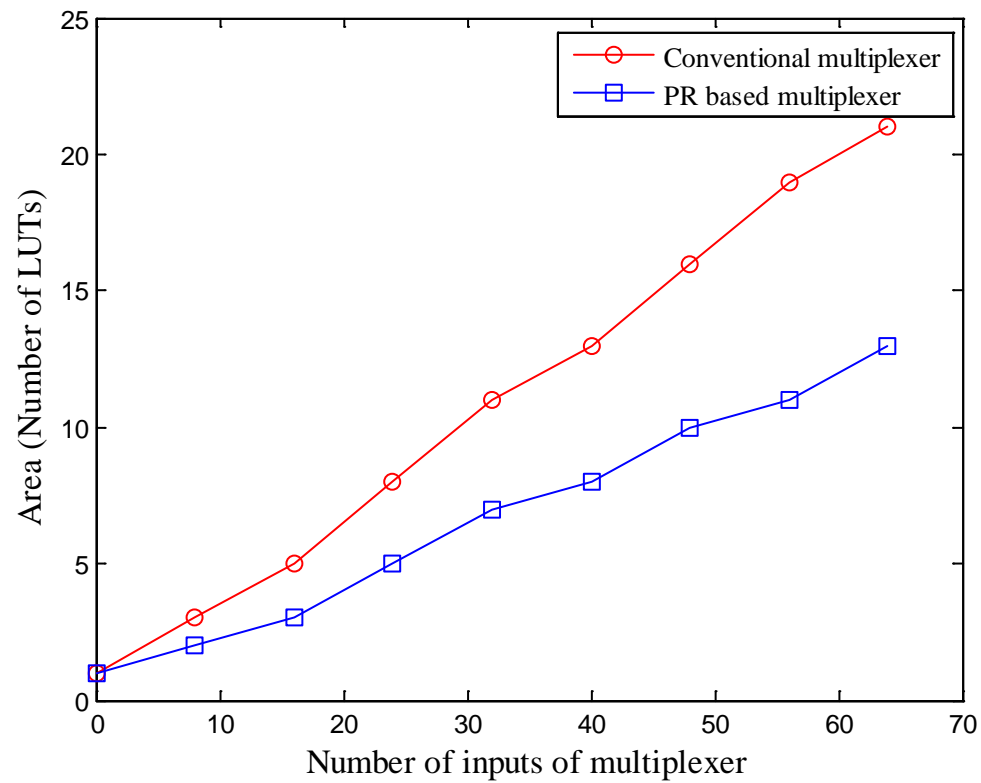


The slide features a dark blue background with a vertical decorative element on the left consisting of several thin, light blue stripes of varying widths. To the right of these stripes are several blue circles of different sizes, arranged in a vertical line. The largest circle is at the top, followed by a smaller one, then a medium-sized one containing the number '18', and finally two more smaller circles at the bottom.

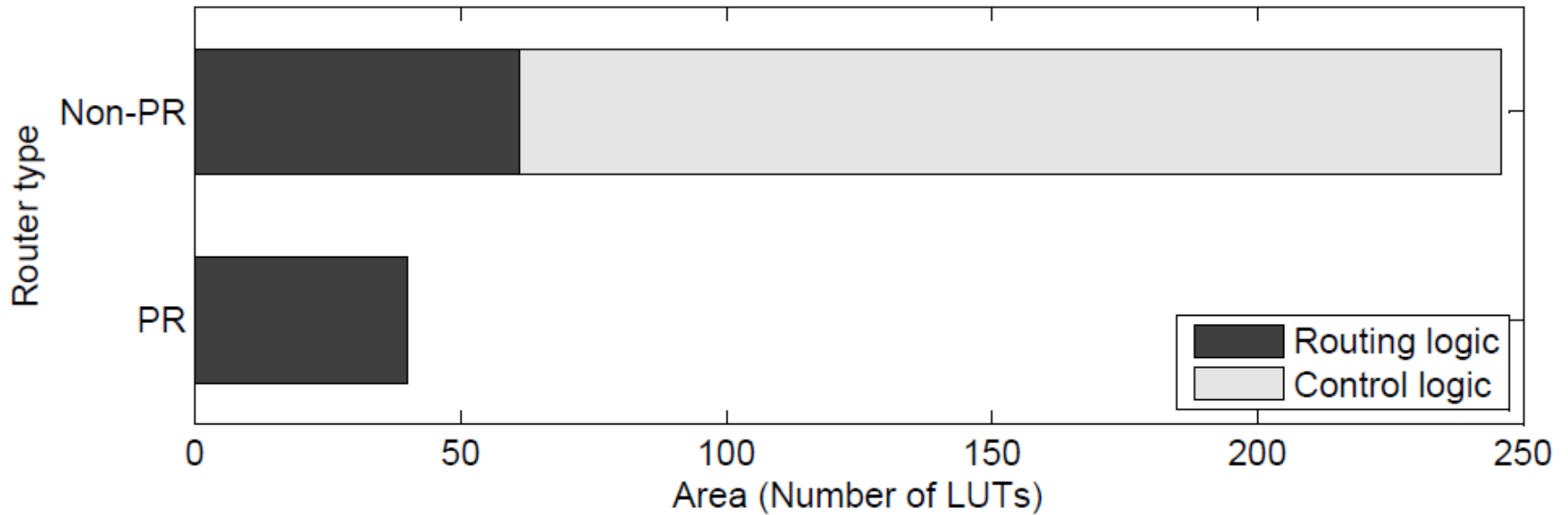
RESULTS AND ANALYSIS

18

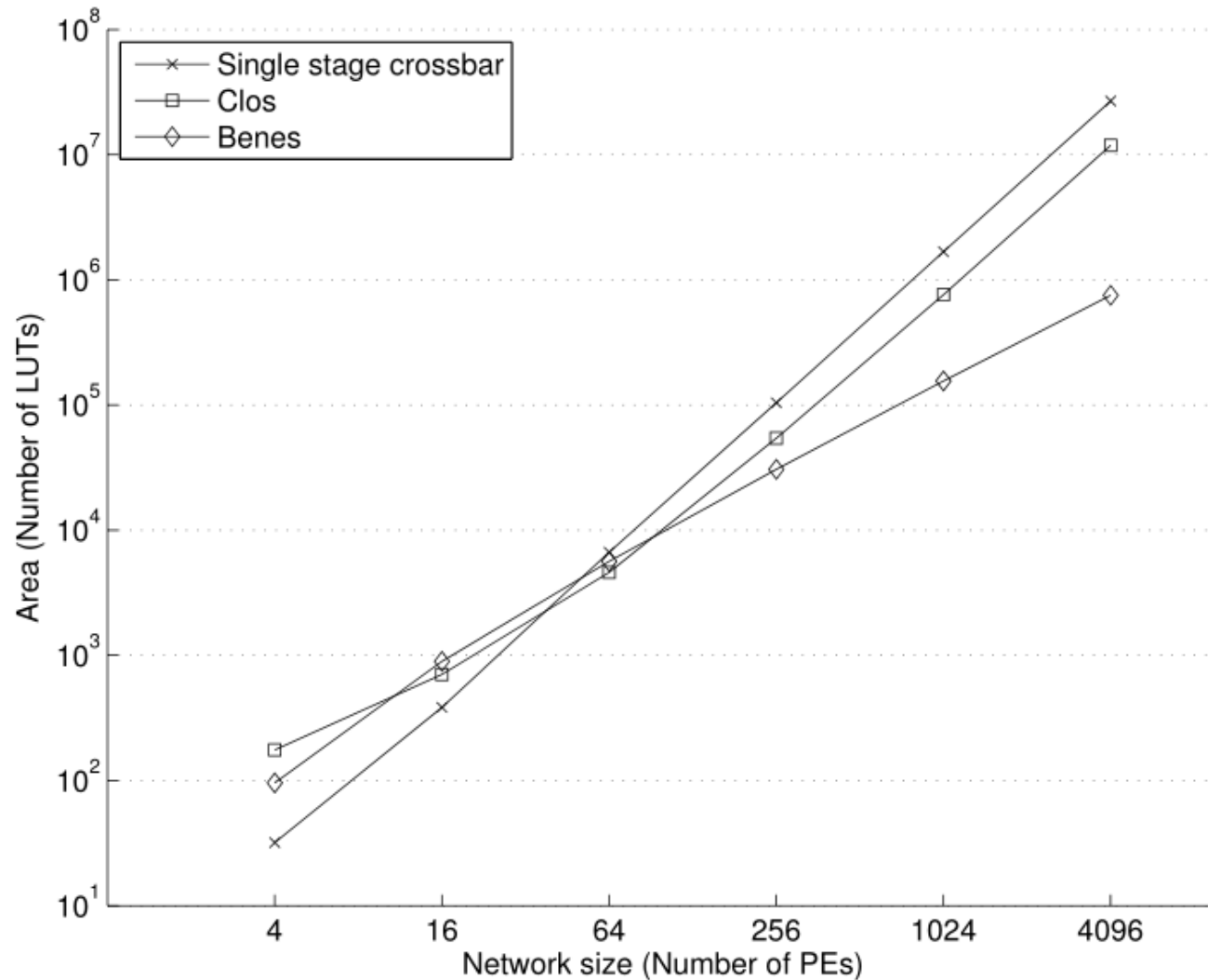
AREA SAVING OF PR ROUTER – SINGLE LINK



AREA SAVINGS OF PR ROUTER



AREA REQUIREMENT OF VARIOUS NETWORK TOPOLOGIES



CONCLUSIONS

- A novel partially reconfigurable crossbar switch design has been presented
 - 84% area saving!
 - 78% reconfiguration delay reduction!
 - Runtime bitstream generation
 - Glitch-free reconfiguration



QUESTIONS?

23