CAD and Routing Architecture for Interposer-Based Multi-FPGA Systems

Andre Hahn Pereira[1] Vaughn Betz[2]

[1]University of São Paulo

[2]University of Toronto

February 27, 2014
22nd ACM/SIGDA International Symposium on Field-Programmable Gate Arrays

Introduction

Architecture Models

CAD Enhancements

Architecture Results

Conclusion and Future Work

Interposer view

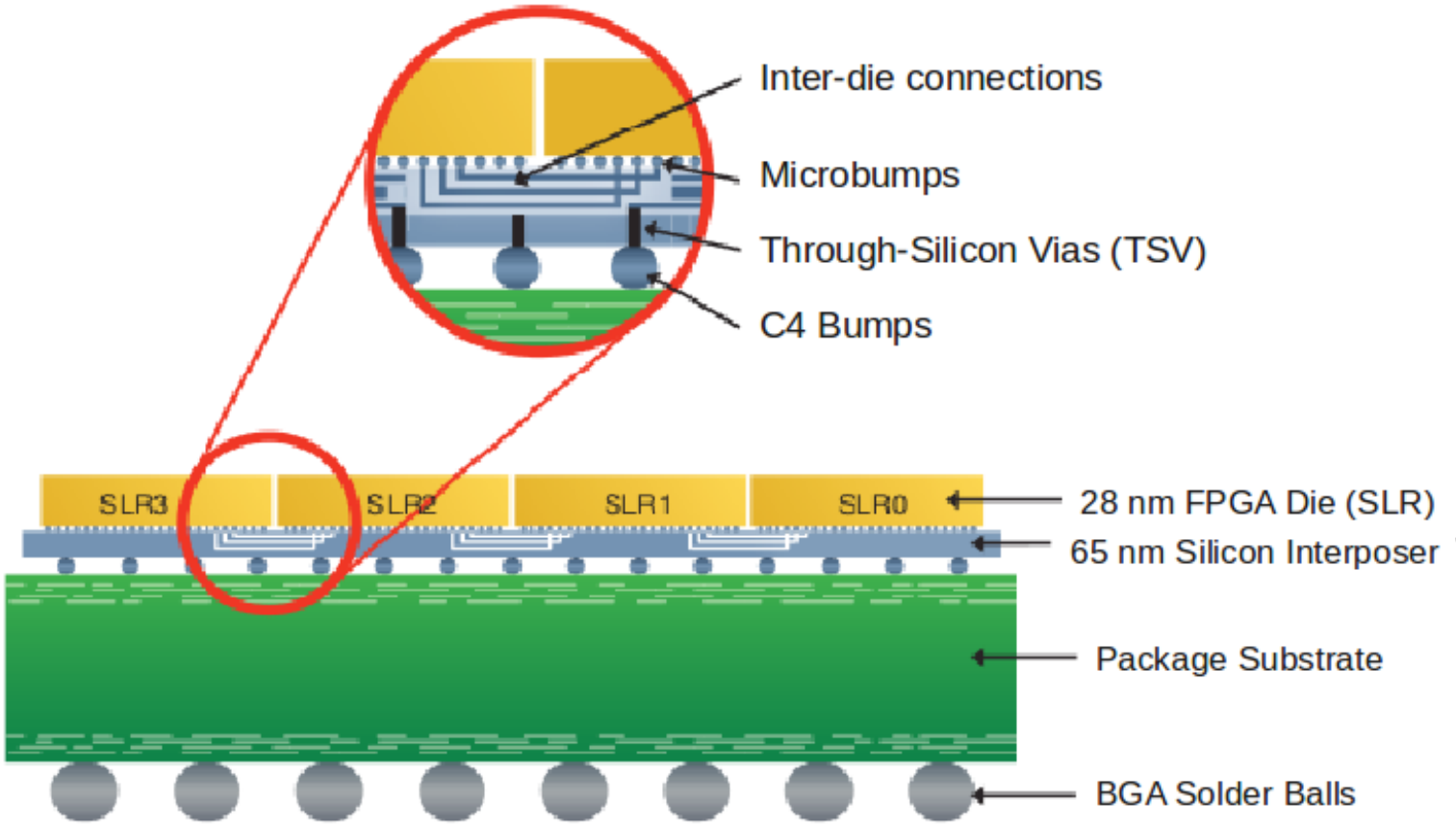


Figure: From Chaware et al (2012)

Interposer view

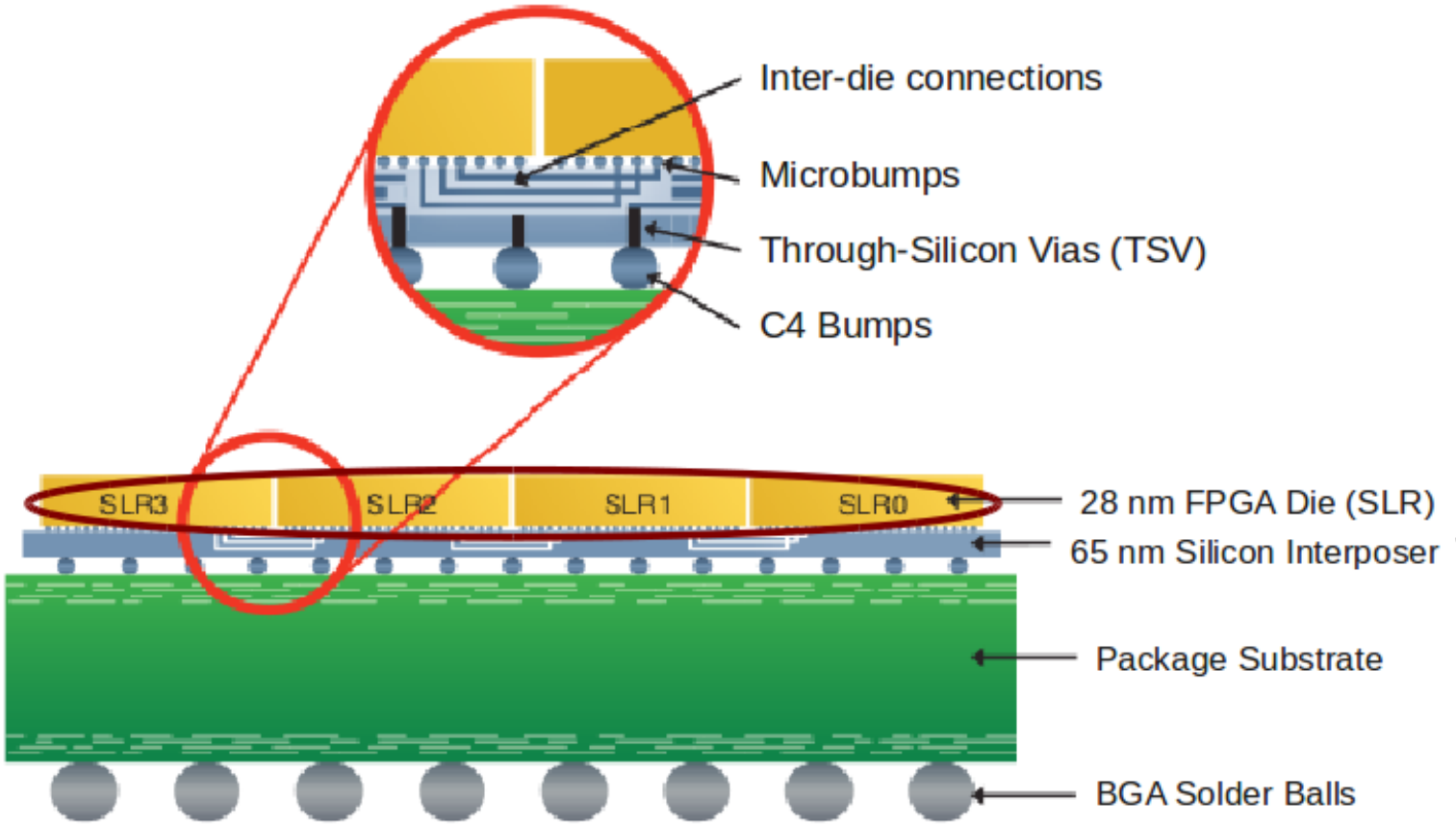


Figure: From Chaware et al (2012)

Interposer view

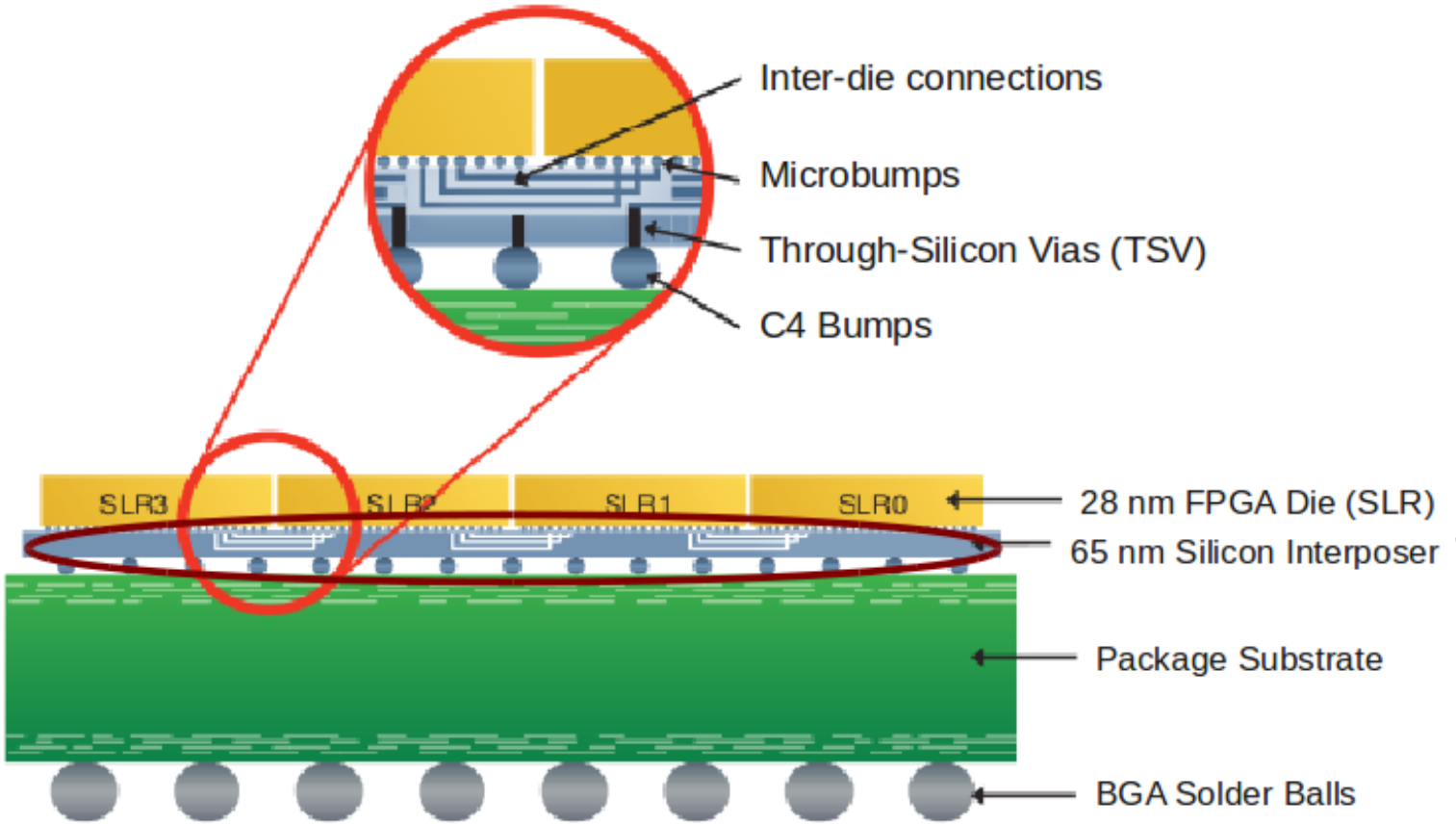


Figure: From Chaware et al (2012)

Interposer view

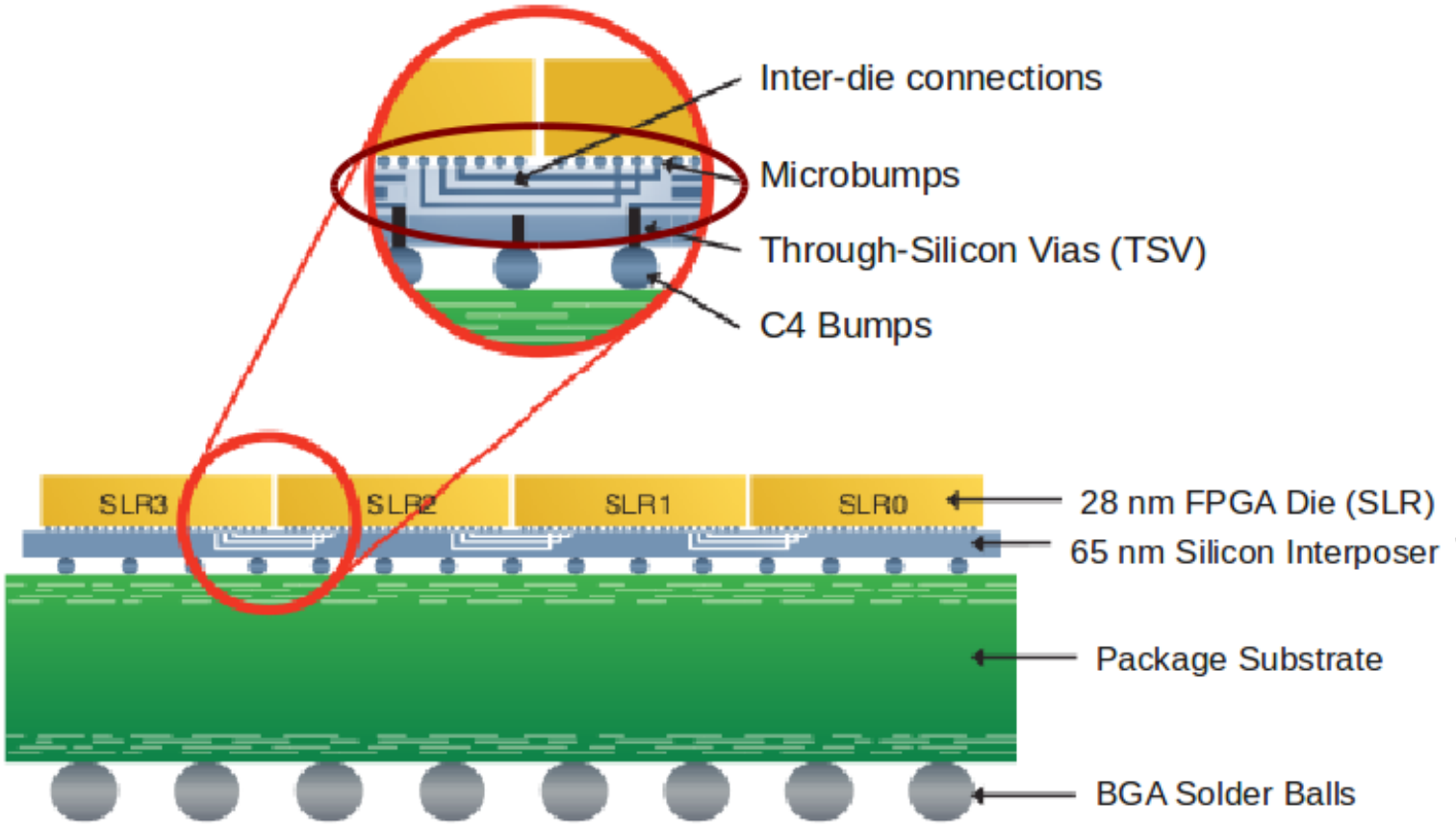


Figure: From Chaware et al (2012)

Interposer view

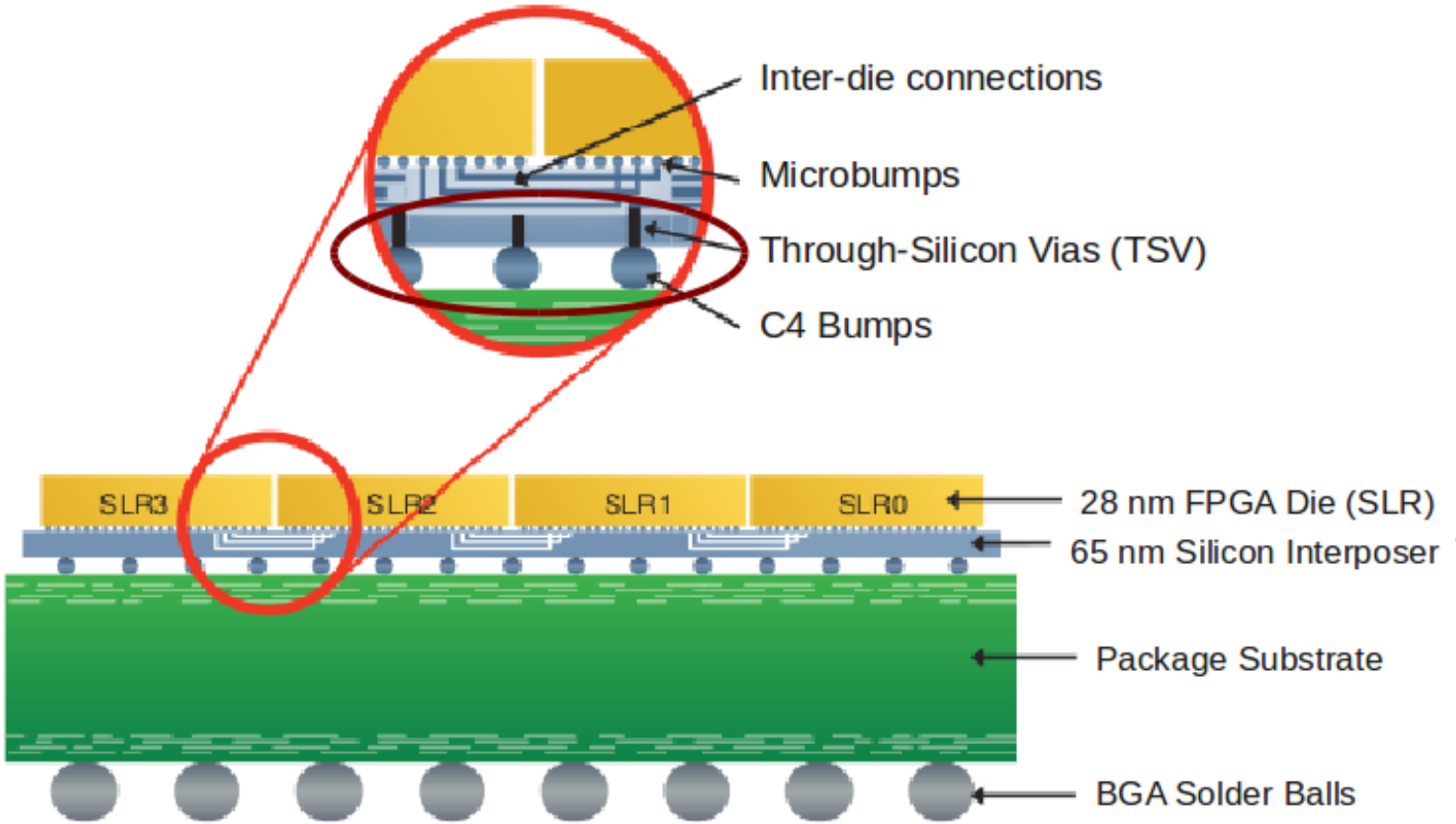


Figure: From Chaware et al (2012)

Motivations

- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice

Motivations

- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice
  - Largest 2.5D FPGA at 28nm (XC7V2000T): 1954k LEs

Motivations

- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice
  - Largest 2.5D FPGA at 28nm (XC7V2000T): 1954k LEs
  - Largest standard FPGA at 28nm (XCE7VX980T): 979k LEs

Motivations

- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice
  - Largest 2.5D FPGA at 28nm (XC7V2000T): 1954k LEs
  - Largest standard FPGA at 28nm (XCE7VX980T): 979k LEs
  - Largest 2.5D FPGA at 20nm (XCVU440): 4407k LEs

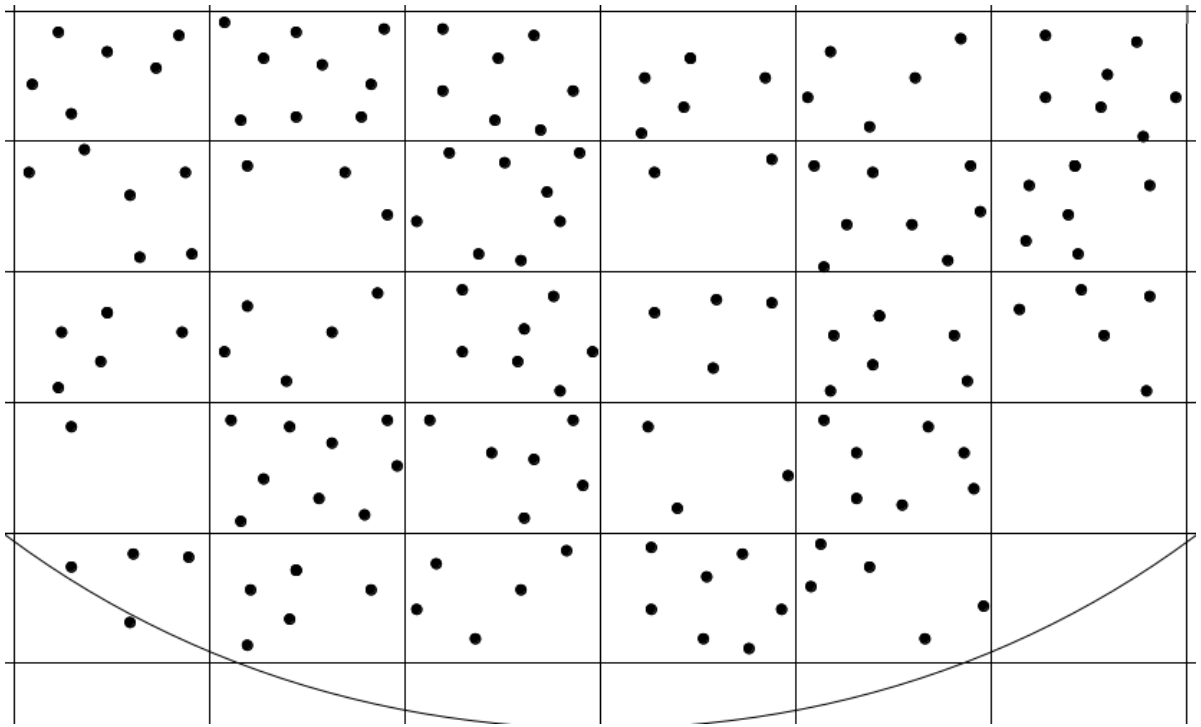
Motivations

- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice
  - Largest 2.5D FPGA at 28nm (XC7V2000T): 1954k LEs
  - Largest standard FPGA at 28nm (XCE7VX980T): 979k LEs
  - Largest 2.5D FPGA at 20nm (XCVU440): 4407k LEs
- Enables integration of heterogeneous dice

Motivations

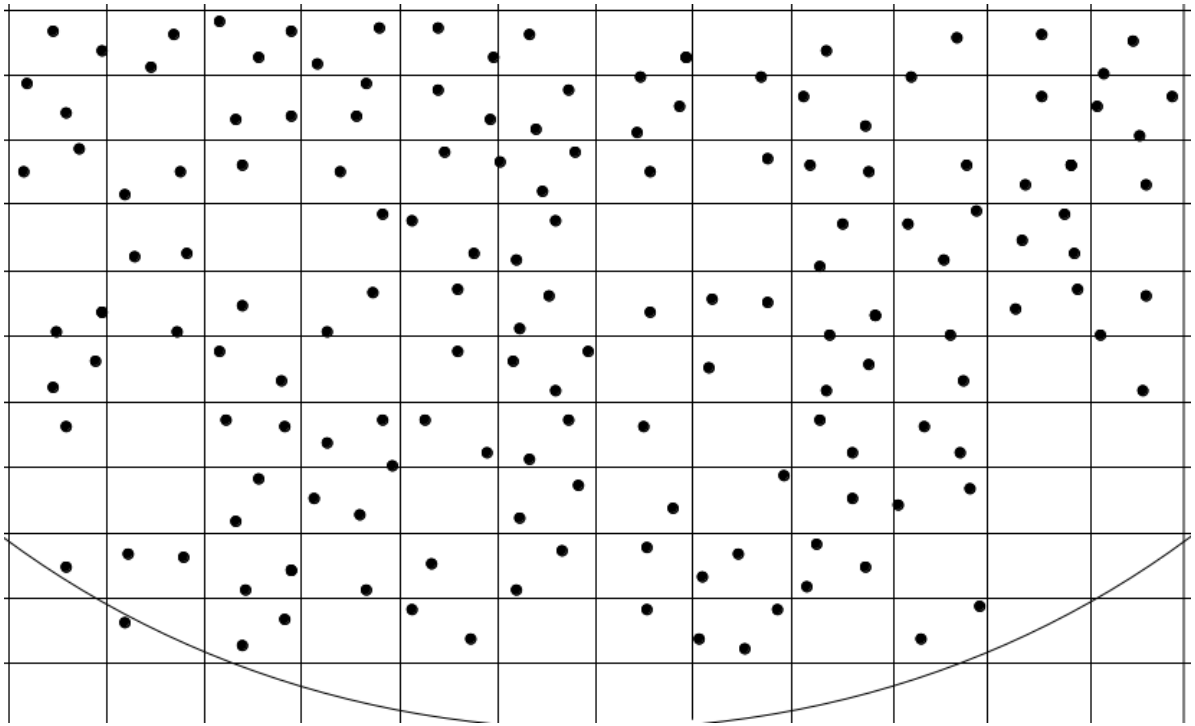
- Enables the creation of large FPGAs, beyond manufacturing limit, composed of smaller dice
  - Largest 2.5D FPGA at 28nm (XC7V2000T): 1954k LEs
  - Largest standard FPGA at 28nm (XCE7VX980T): 979k LEs
  - Largest 2.5D FPGA at 20nm (XCVU440): 4407k LEs
- Enables integration of heterogeneous dice
- Better yield $\rightarrow$ lower cost

Die Yield example



- For a defect density of $1/cm^2$ and $6cm^2$ dice, on a 12 inch wafer:
- Die yield: 0.25%, avg working dice: 0.3
- Average working systems: 0.3

Die Yield example



- For a defect density of $1/cm^2$ and $1.5cm^2$ dice, on a 12 inch wafer:
- Die yield: 22%, avg working dice: 107
- Average working systems: 26.75

Interposer challenges

- Reduced connectivity:

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance
- Virtex-7 XC7V2000T:

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance
- Virtex-7 XC7V2000T:
  - 280×210 total vertical wires

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance
- Virtex-7 XC7V2000T:
  - 280×210 total vertical wires
  - 280×48 wires crossing the interposer

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance
- Virtex-7 XC7V2000T:
  - 280×210 total vertical wires
  - 280×48 wires crossing the interposer
  - $\sim$23% of the wires cross the interposer

Interposer challenges

- Reduced connectivity:
  - Connection to interposer made by microbumps
- Increased delay:
  - Increased wirelength and microbump capacitance
- Virtex-7 XC7V2000T:
  - 280×210 total vertical wires
  - 280×48 wires crossing the interposer
  - $\sim$23% of the wires cross the interposer
  - $\sim$1ns extra delay to cross

Goals

- CAD flow for interposer-based FPGAs

Goals

- CAD flow for interposer-based FPGAs
- Limited connectivity between dice: impact on circuit routability?

Goals

- CAD flow for interposer-based FPGAs
- Limited connectivity between dice: impact on circuit routability?
- Interposer delay: impact on circuit speed?

Introduction

Architecture Models

CAD Enhancements

Architecture Results

Conclusion and Future Work

Models overview

- Used VTR flow as the base

Models overview

- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs

Models overview

- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs
- Required changes to placement, routing and timing analysis

Models overview

- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs
- Required changes to placement, routing and timing analysis
- Parameters added:

Models overview

- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs
- Required changes to placement, routing and timing analysis
- Parameters added:
  - Number of cuts

Models overview

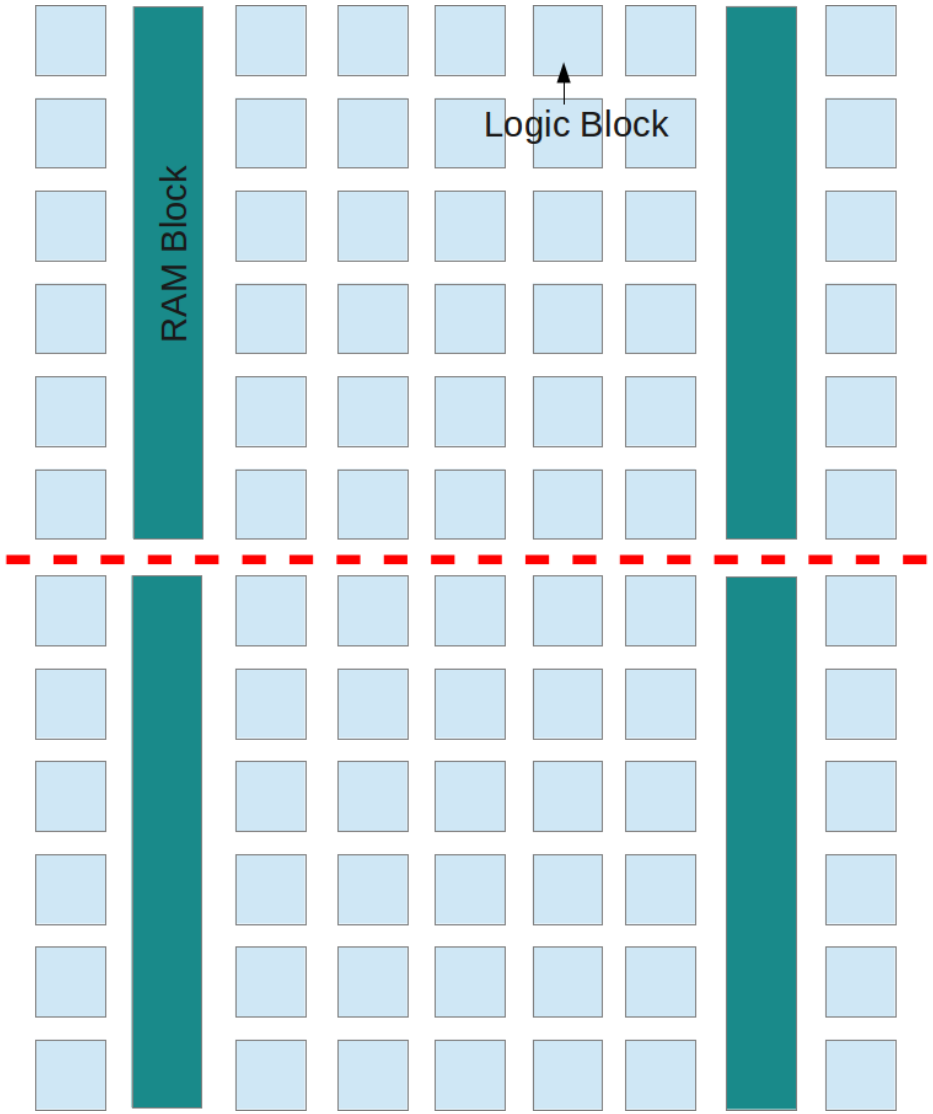
- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs
- Required changes to placement, routing and timing analysis
- Parameters added:
  - Number of cuts
  - % wires cut

Models overview

- Used VTR flow as the base
- Modified VPR to target/model interposer-based FPGAs
- Required changes to placement, routing and timing analysis
- Parameters added:
  - Number of cuts
  - % wires cut
  - Delay added by interposer

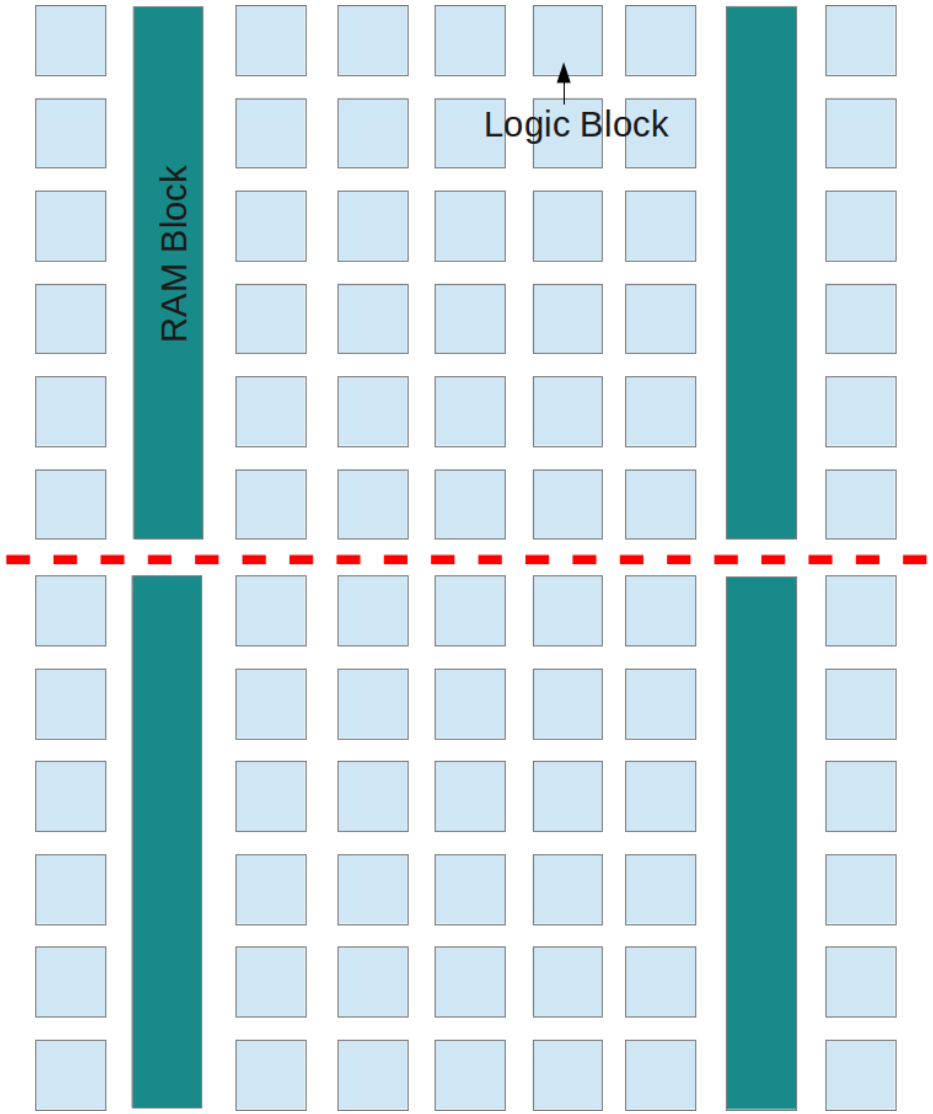
Number of cuts

- Models the number of dice on the FPGA



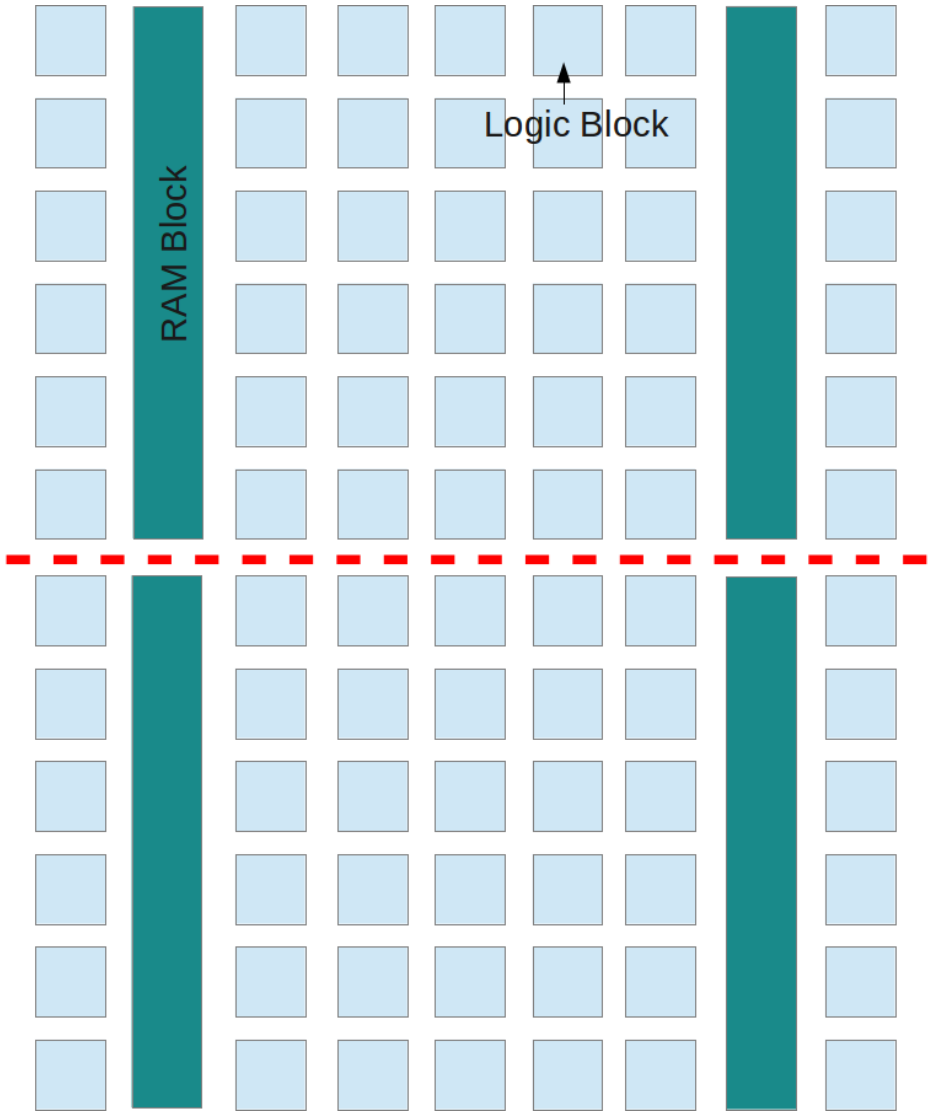
Number of cuts

- Models the number of dice on the FPGA
- Cuts are equally spaced vertically

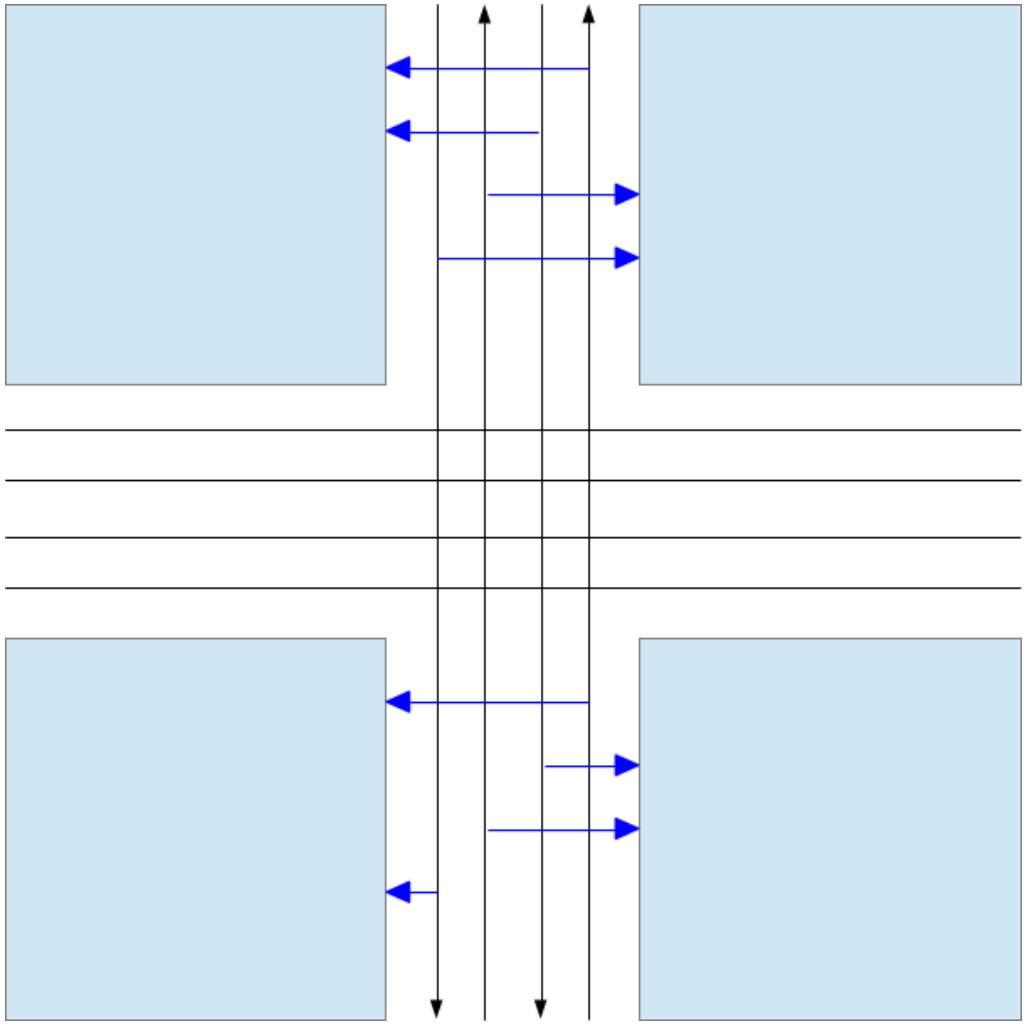


Number of cuts

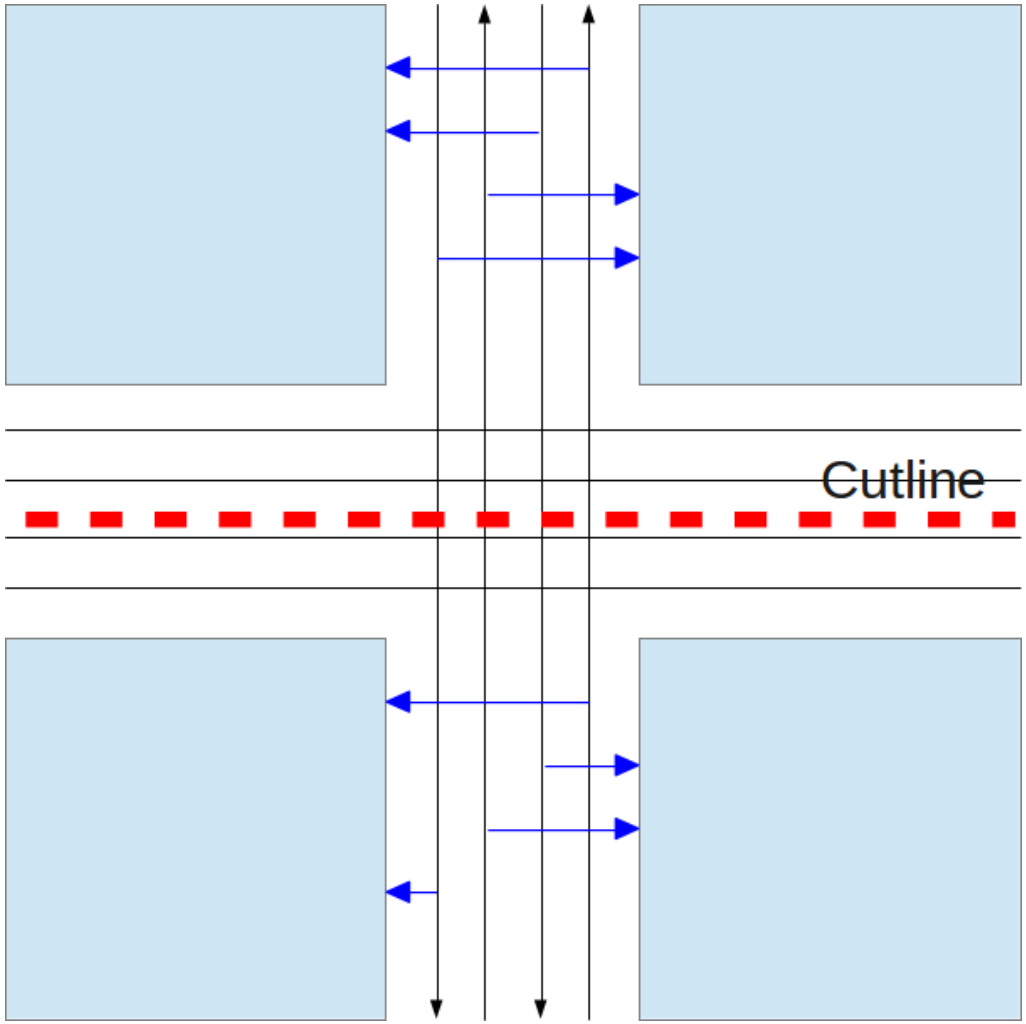
- Models the number of dice on the FPGA
- Cuts are equally spaced vertically
- 1 cut $\rightarrow$ 2 dice, 3 cuts $\rightarrow$ 4 dice



% wires cut & Increased delay

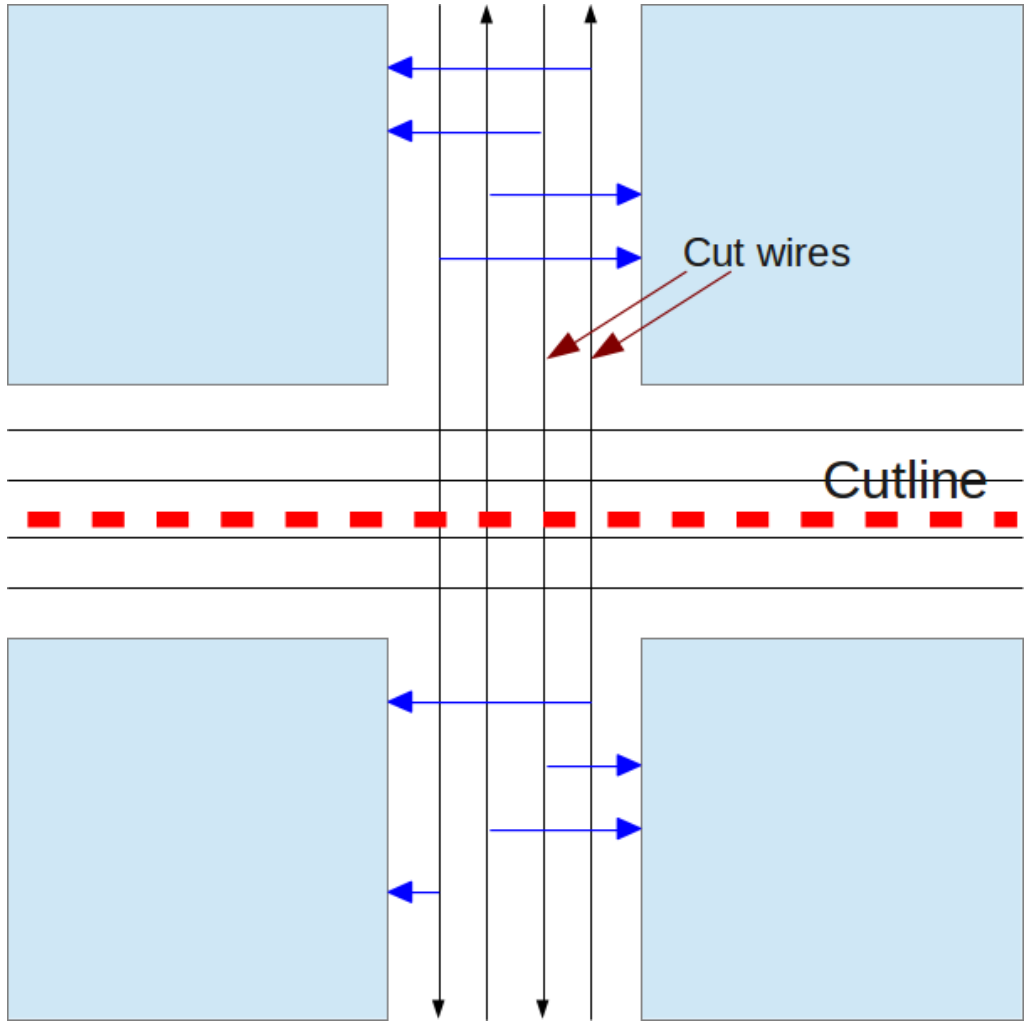


% wires cut & Increased delay



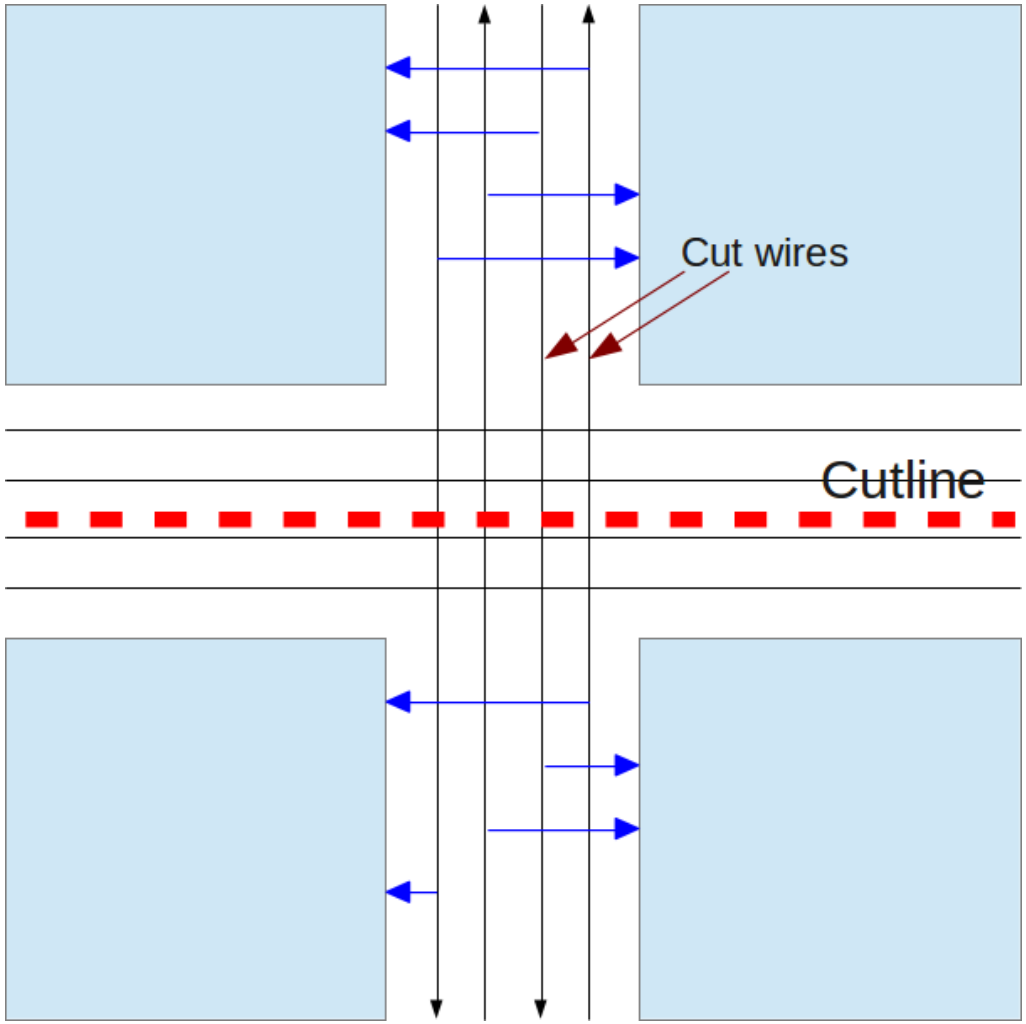
% wires cut & Increased delay

- % wires cut:
  - Models reduced connectivity between dice



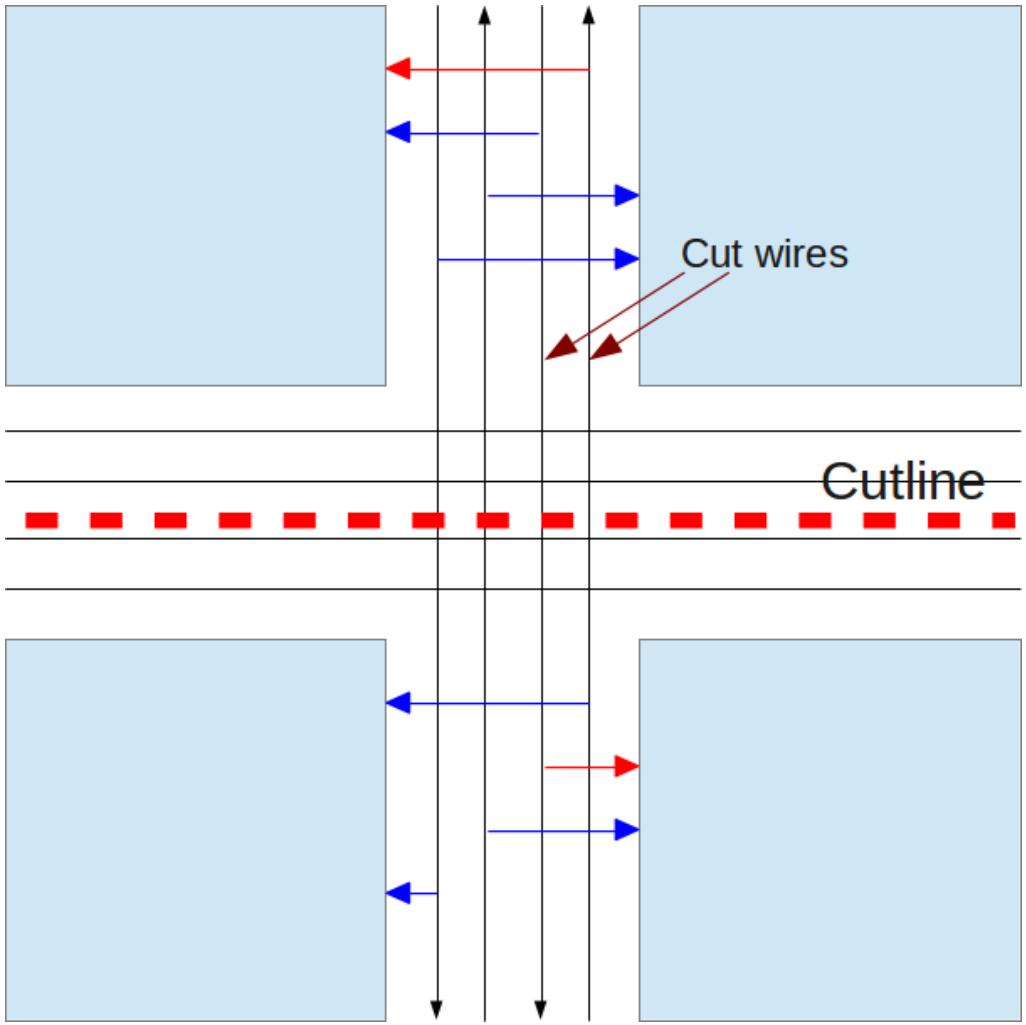
% wires cut & Increased delay

- % wires cut:
  - Models reduced connectivity between dice
  - % of wires which are removed between dice



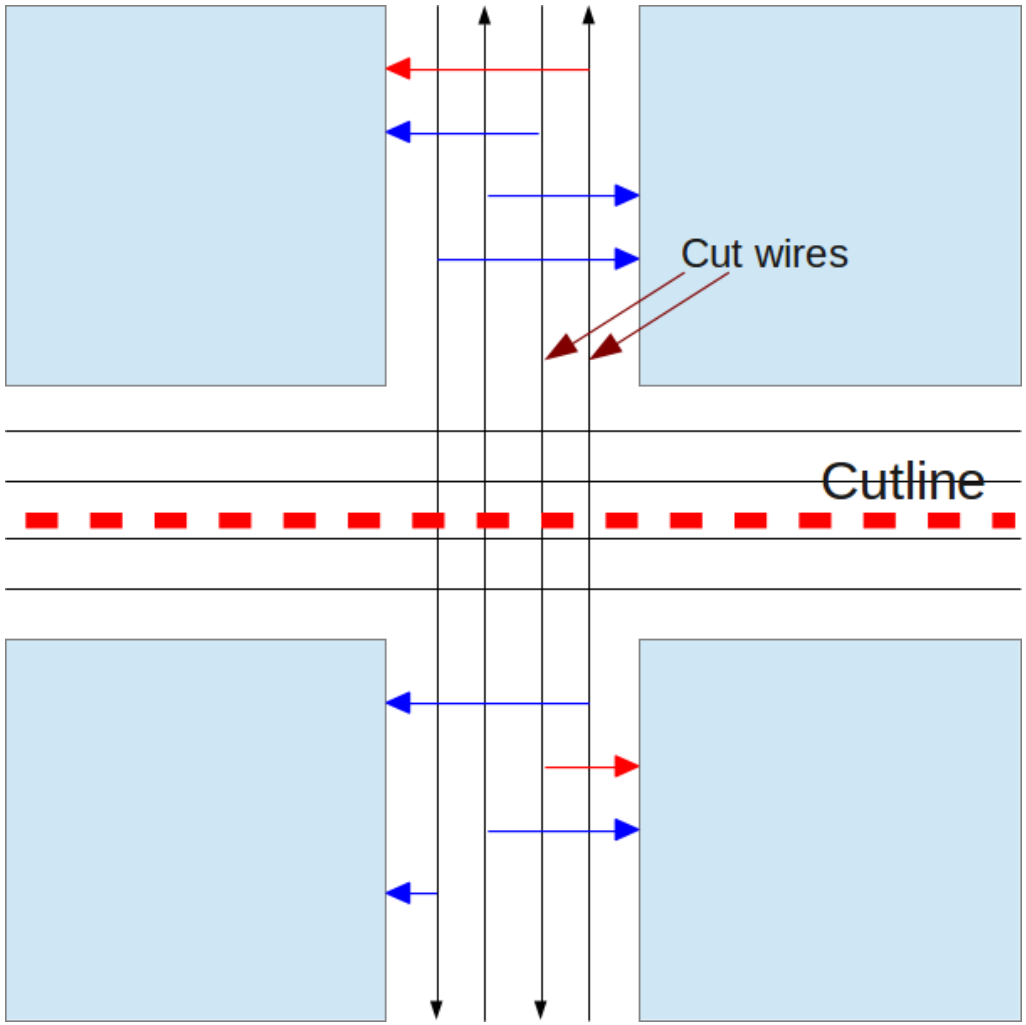
% wires cut & Increased delay

- % wires cut:
  - Models reduced connectivity between dice
  - % of wires which are removed between dice



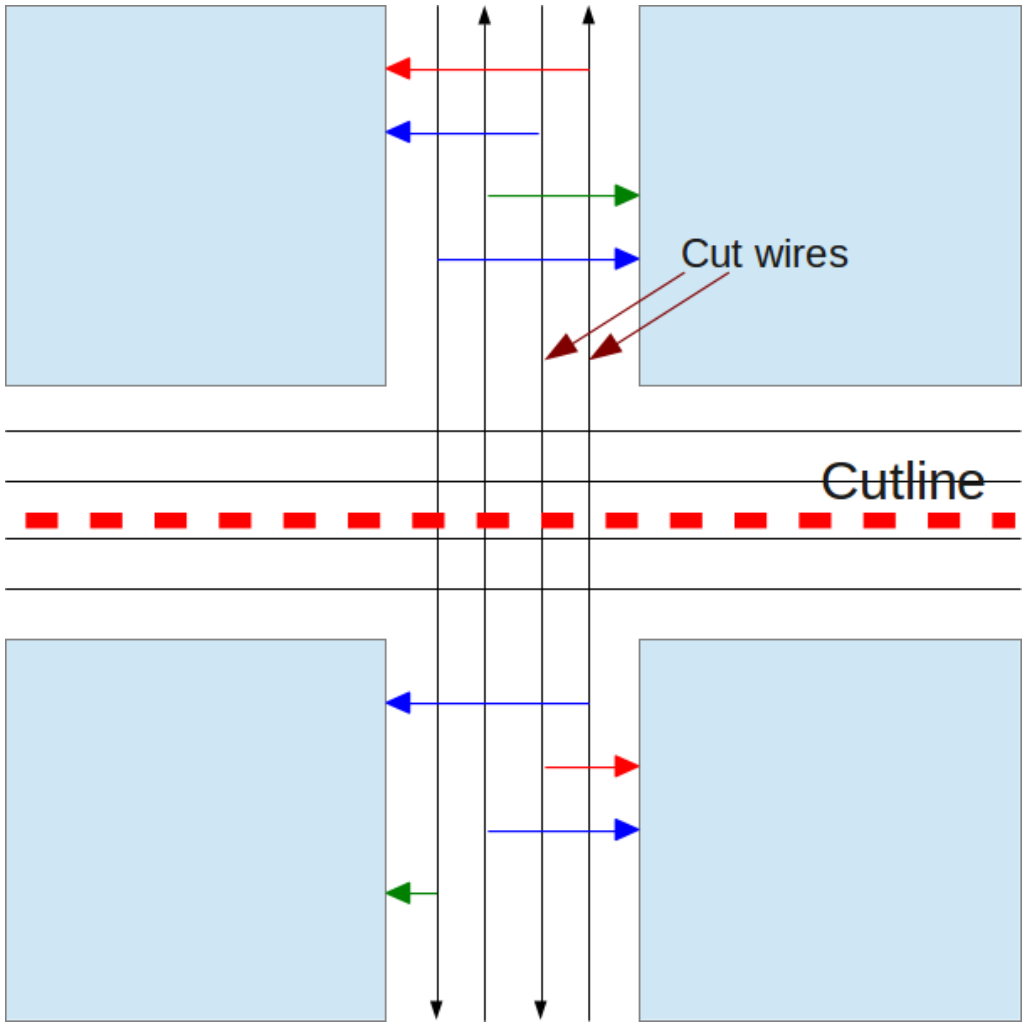
% wires cut & Increased delay

- % wires cut:
  - Models reduced connectivity between dice
  - % of wires which are removed between dice
- Increased delay:
  - Models larger delay to cross the interposer



% wires cut & Increased delay

- % wires cut:
  - Models reduced connectivity between dice
  - % of wires which are removed between dice
- Increased delay:
  - Models larger delay to cross the interposer
  - A reasonable value is $1ns$



Introduction

Architecture Models

CAD Enhancements

Architecture Results

Conclusion and Future Work

Required changes

- Changed routing resource graph

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges
  - Now router automatically adapts to the interposer

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges
  - Now router automatically adapts to the interposer
  - Timing analysis also adapts automatically

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges
  - Now router automatically adapts to the interposer
  - Timing analysis also adapts automatically
- Placer needs to change to:

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges
  - Now router automatically adapts to the interposer
  - Timing analysis also adapts automatically
- Placer needs to change to:
  - Reduce nets crossing interposer

Required changes

- Changed routing resource graph
  - Changed delays of appropriate edges
  - Removed some edges
  - Now router automatically adapts to the interposer
  - Timing analysis also adapts automatically
- Placer needs to change to:
  - Reduce nets crossing interposer
  - Minimize critical path crossings of interposer

Placement: Timing cost

- $Timing\_Cost = \sum_{\forall i,j \subset circuit} delay(\Delta x_{ij}, \Delta y_{ij}) \times criticality(i, j)$

Placement: Timing cost

- $Timing\_Cost = \sum_{\forall i,j \subset circuit} delay(\Delta x_{ij}, \Delta y_{ij}) \times criticality(i, j)$

Placement: Timing cost

- $Timing\_Cost = \sum_{\forall i,j \subset circuit} delay(\Delta x_{ij}, \Delta y_{ij}) \times criticality(i,j)$

Placement: Timing cost

- $Timing\_Cost = \sum_{\forall i,j \subset circuit} delay(\Delta x_{ij}, \Delta y_{ij}) \times criticality(i,j)$
- Need to update estimate of best case routing delay

Placement: Timing cost

- $Timing\_Cost = \sum_{\forall i,j \subset circuit} delay(\Delta x_{ij}, \Delta y_{ij}) \times criticality(i,j)$
- Need to update estimate of best case routing delay
- $extra\_delay(i,j) = times\_crossed(i,j) \times delay\_increase$

Placement: Wiring cost

- $wiring\_cost_{orig} = \sum_{n=1}^{N_{nets}} q(n) \times [\frac{bb_x(n)}{avg\_chanx\_W(n)} + \frac{bb_y(n)}{avg\_chany\_W(n)}]$

Placement: Wiring cost

- $wiring\_cost_{orig} = \sum_{n=1}^{N_{nets}} q(n) \times [\frac{bb_x(n)}{avg\_chanx\_W(n)} + \frac{bb_y(n)}{avg\_chany\_W(n)}]$

Placement: Wiring cost

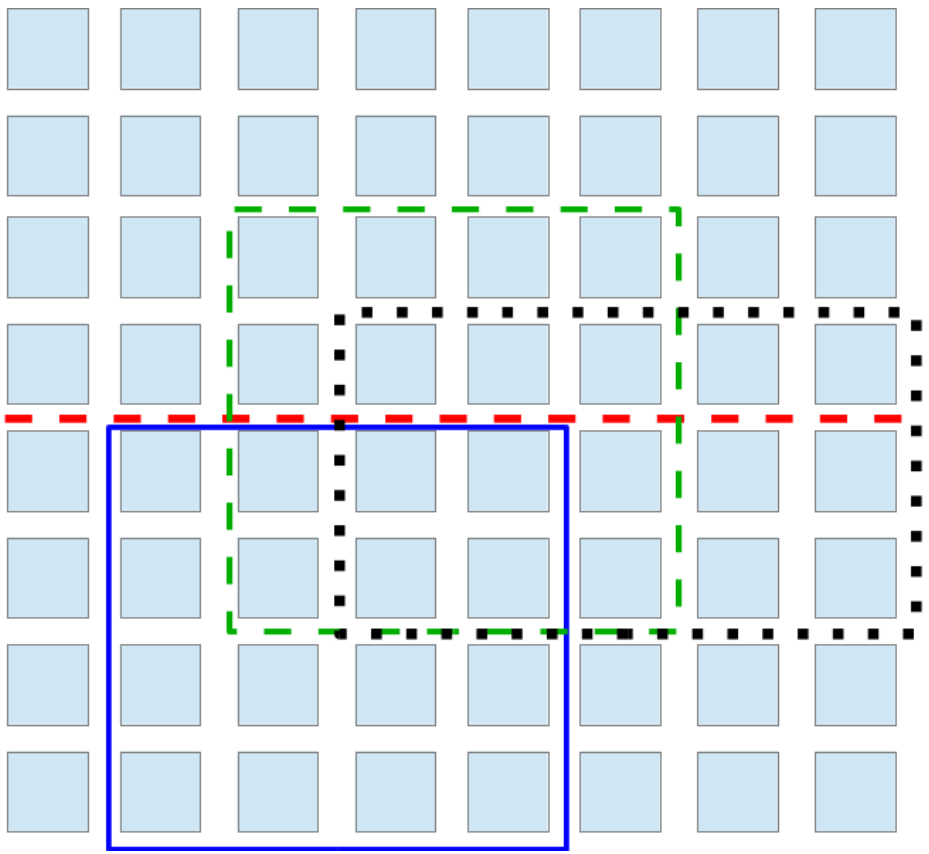
- $wiring\_cost_{orig} = \sum_{n=1}^{N_{nets}} q(n) \times [\frac{bb_x(n)}{avg\_chanx\_W(n)} + \frac{bb_y(n)}{avg\_chany\_W(n)}]$

Placement: Wiring cost

- $wiring\_cost_{orig} = \sum_{n=1}^{N_{nets}} q(n) \times [\frac{bb_x(n)}{avg\_chanx\_W(n)} + \frac{bb_y(n)}{avg\_chany\_W(n)}]$
- $wiring\_cost = wiring\_cost_{orig} + cut\_cost$

Tested wiring costs

- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times times\_crossed(n)$

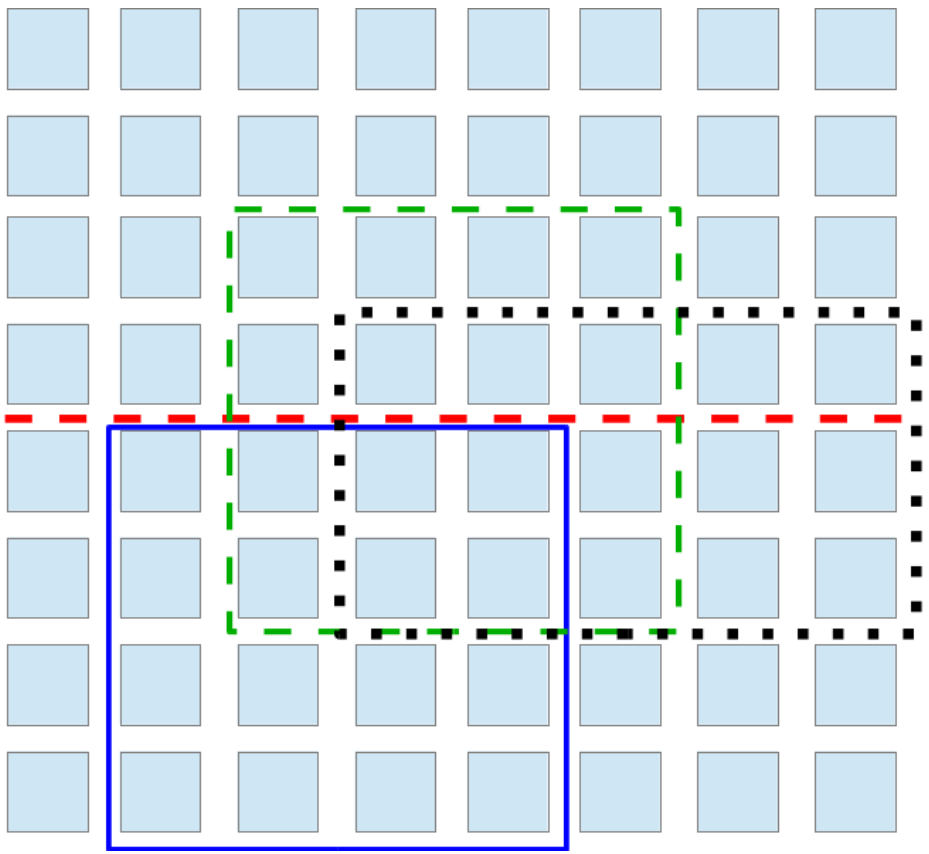


Cut costs:

- Green: 1
- Black: 1
- Blue: 0

Tested wiring costs

- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times times\_crossed(n)$
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(n)$

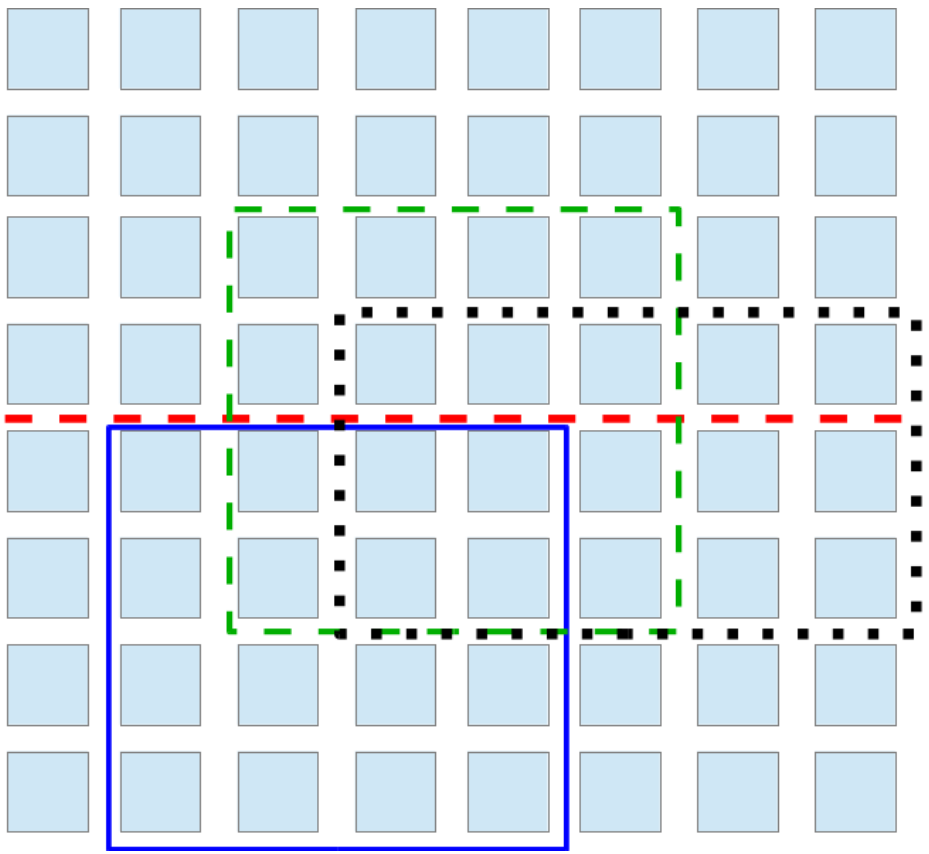


Cut costs:

- Green: 4
- Black: 3
- Blue: 4

Tested wiring costs

- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times times\_crossed(n)$
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(n)$
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(i) \times times\_crossed(n)$

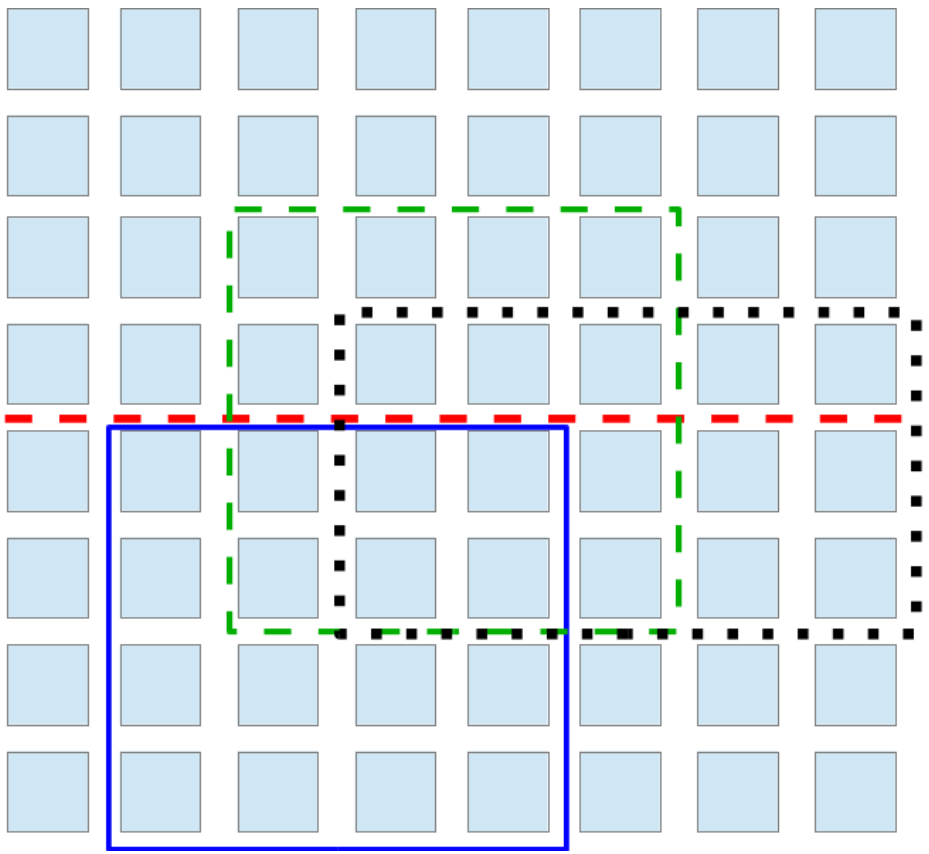


Cut costs:

- Green: 4
- Black: 3
- Blue: 0

Tested wiring costs

- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times times\_crossed(n)$
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(n)$
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(i) \times times\_crossed(n)$
  - Smoother cost function guides gradual progress



Cut costs:

- Green: 4
- Black: 3
- Blue: 0

Experiment conditions

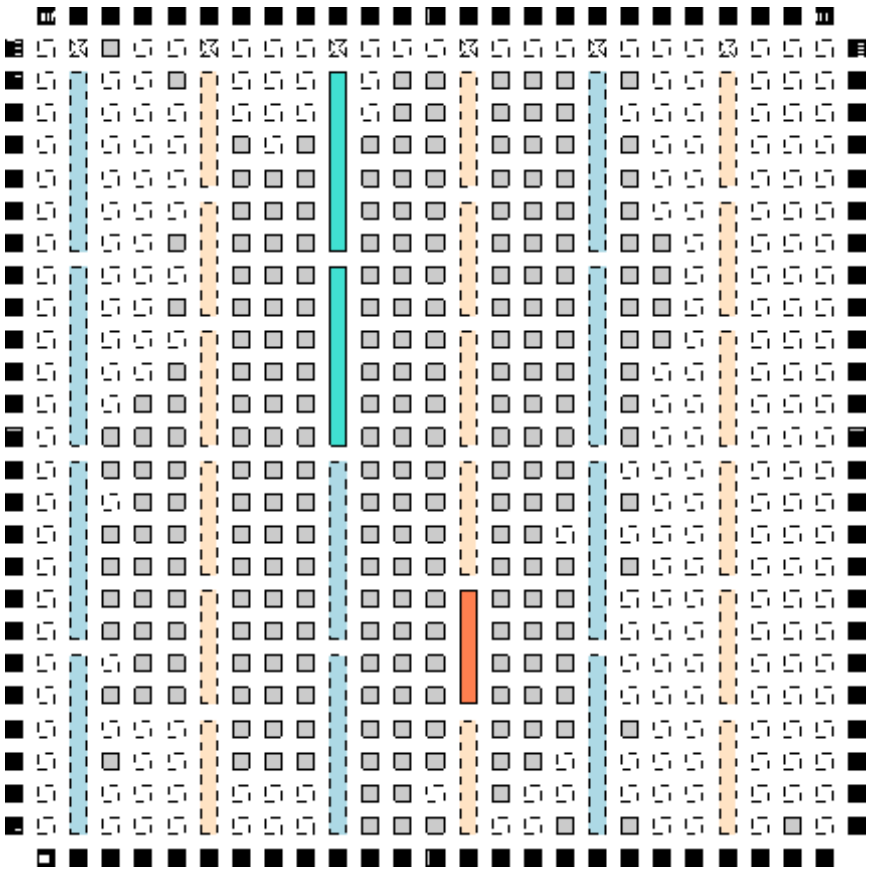
- Architecture file from VTR 7.0,
  40nm area and delay

Experiment conditions

- Architecture file from VTR 7.0, 40nm area and delay
- Unidirectional wires with length 4

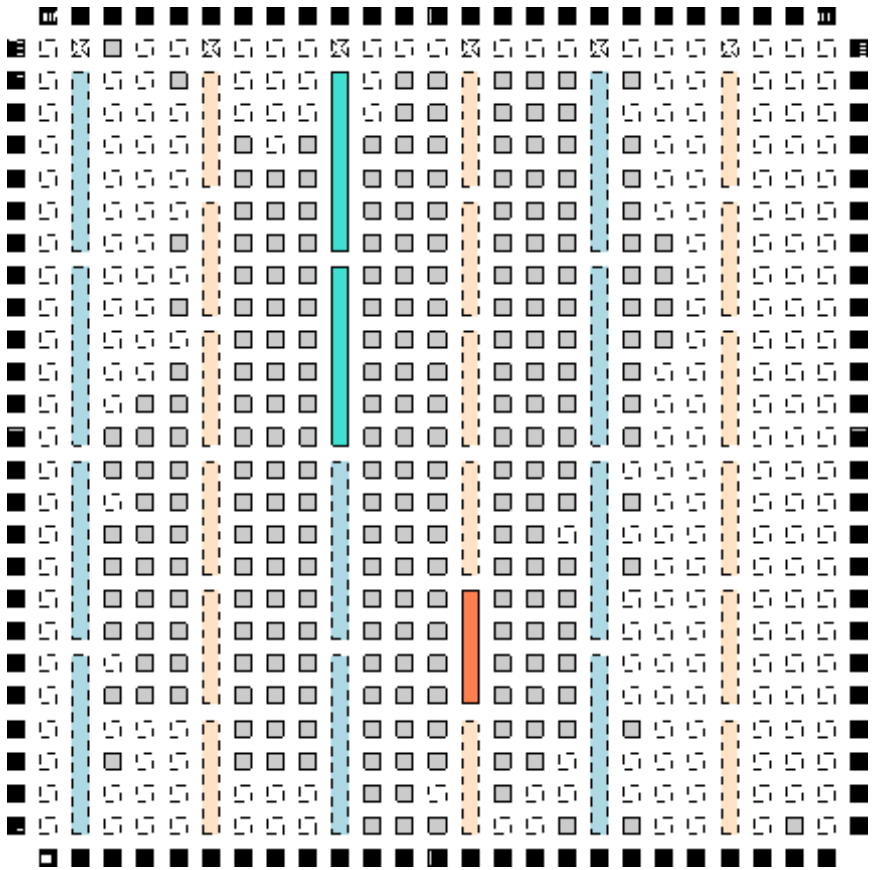
Experiment conditions

- Architecture file from VTR 7.0, 40nm area and delay
- Unidirectional wires with length 4
- 10 fracturable 6-LUTs per logic block



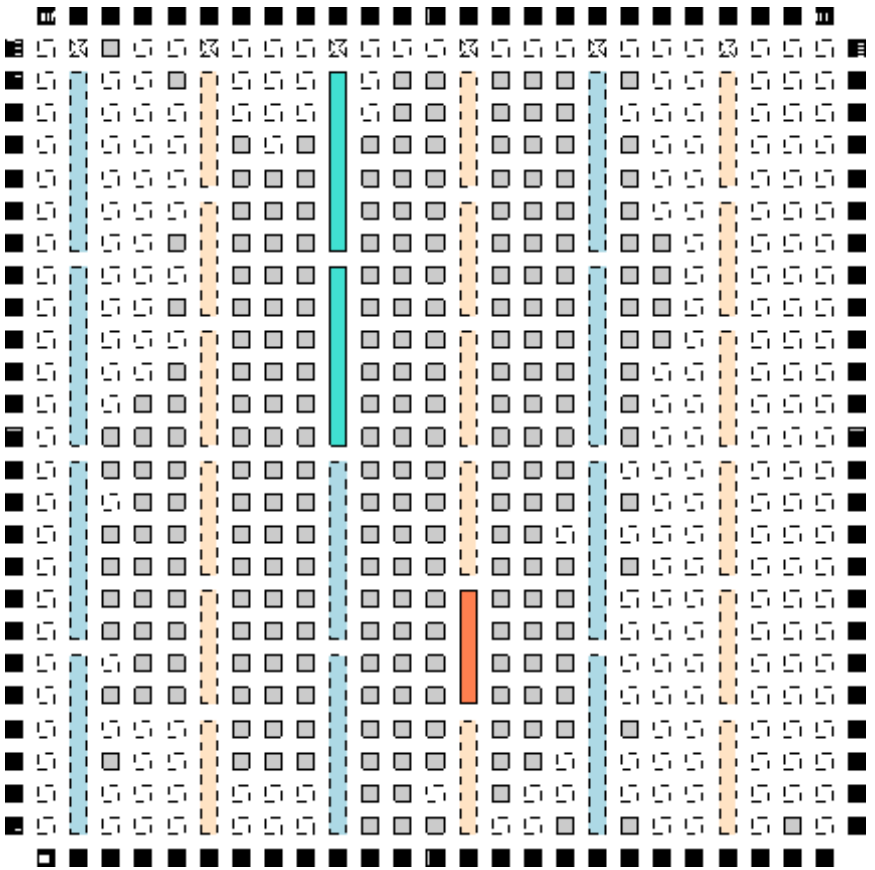
Experiment conditions

- Architecture file from VTR 7.0, 40nm area and delay
- Unidirectional wires with length 4
- 10 fracturable 6-LUTs per logic block
- 32kb RAM blocks, reconfigurable DSP blocks



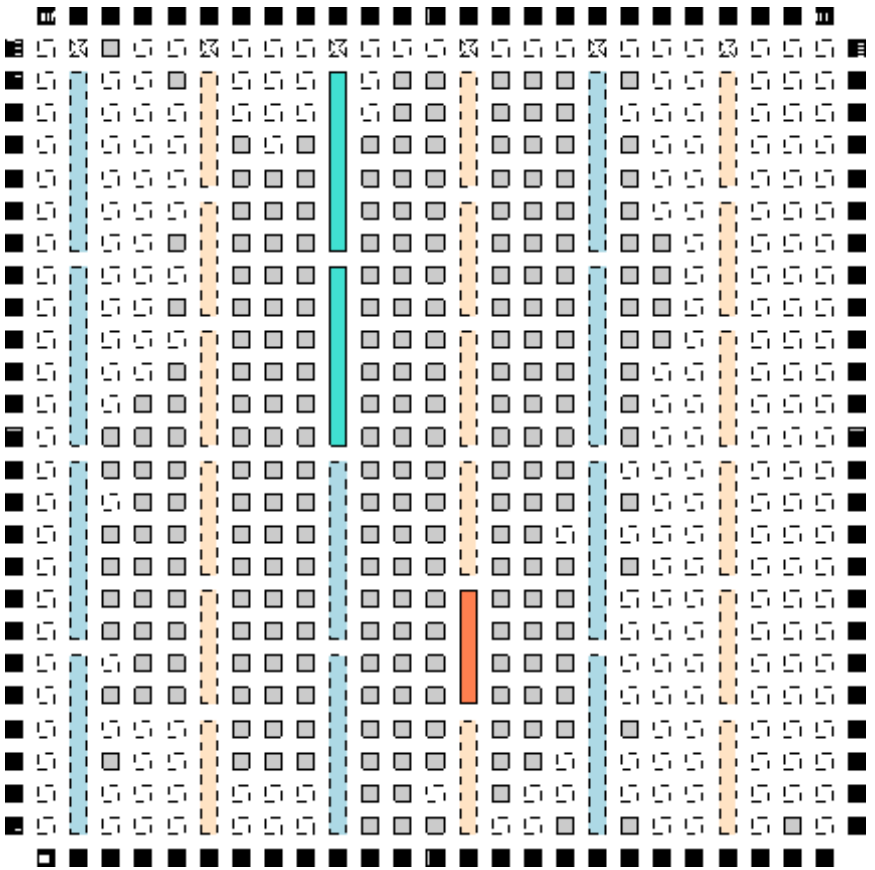
Experiment conditions

- Architecture file from VTR 7.0, 40nm area and delay
- Unidirectional wires with length 4
- 10 fracturable 6-LUTs per logic block
- 32kb RAM blocks, reconfigurable DSP blocks
- Experiments ran with the eight largest circuits from VTR, ranging from 9.1k to 153k primitives



Experiment conditions

- Architecture file from VTR 7.0, 40nm area and delay
- Unidirectional wires with length 4
- 10 fracturable 6-LUTs per logic block
- 32kb RAM blocks, reconfigurable DSP blocks
- Experiments ran with the eight largest circuits from VTR, ranging from 9.1k to 153k primitives
- Results are the geometric mean over all circuits



Enhancements results

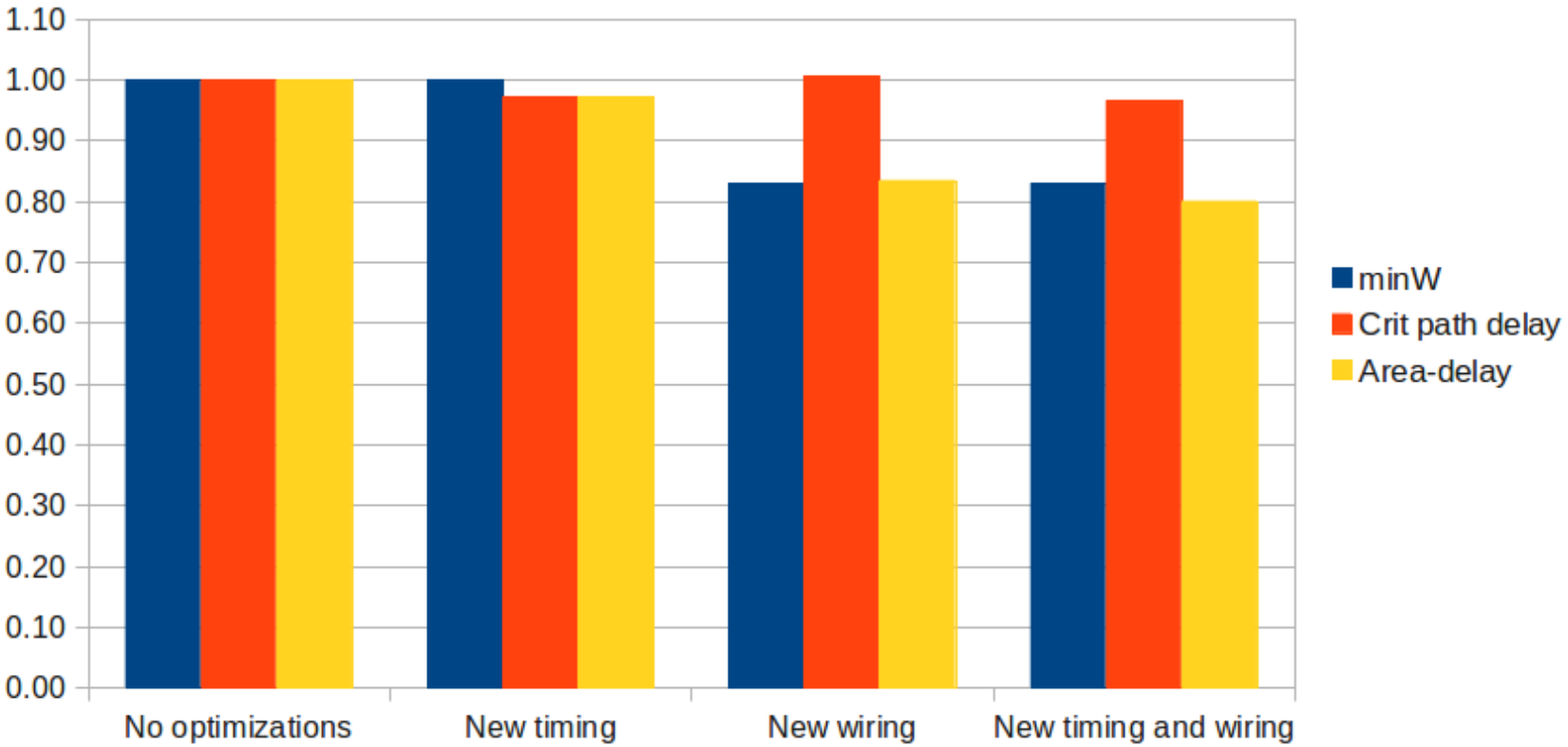
- Best placement routability cost term:

Enhancements results

- Best placement routability cost term:
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(i) \times times\_crossed(n)$

Enhancements results

- Best placement routability cost term:
- $cut\_cost = \sum_{n=1}^{N_{nets}} C' \times bbHeight(i) \times times\_crossed(n)$



Introduction

Architecture Models

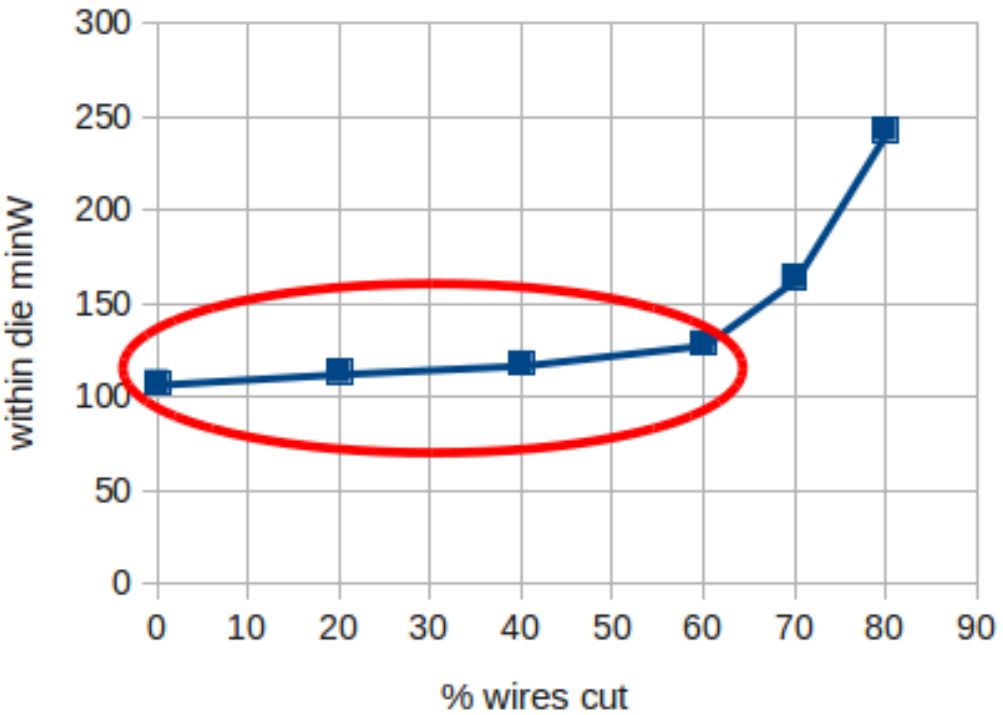
CAD Enhancements

Architecture Results

Conclusion and Future Work

Routability vs. Interposer Wiring Bandwidth

Minimum channel width vs % wires cut

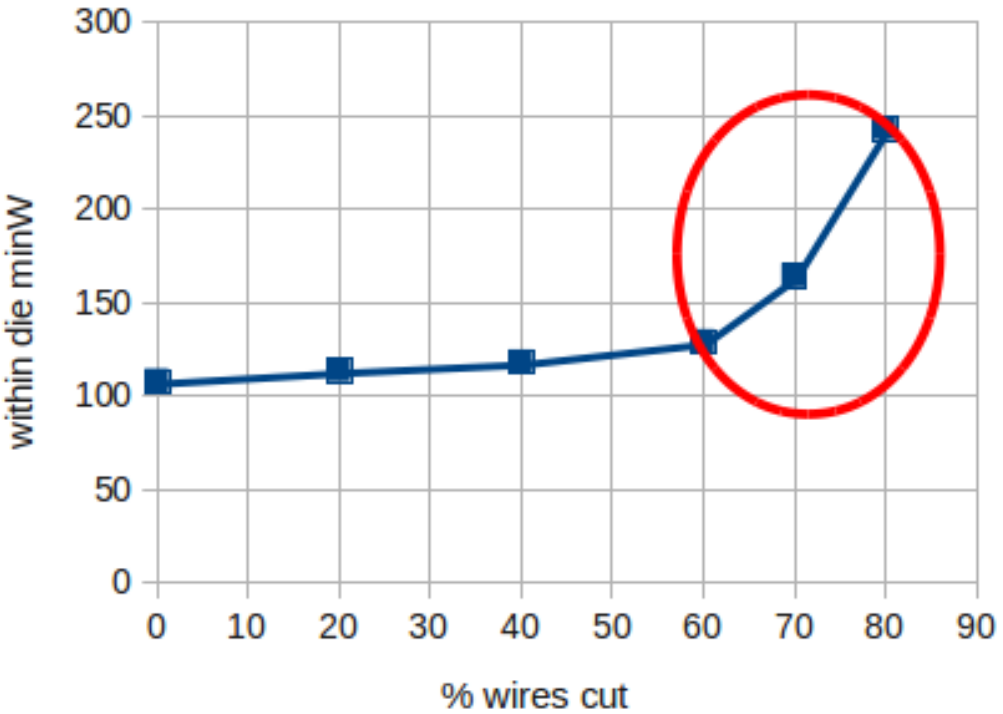


Delay increase = 1ns and number of cuts = 3
(4 dice)

- Cutting up to 60% of wires at interposer: modest impact on routability

Routability vs. Interposer Wiring Bandwidth

Minimum channel width vs % wires cut

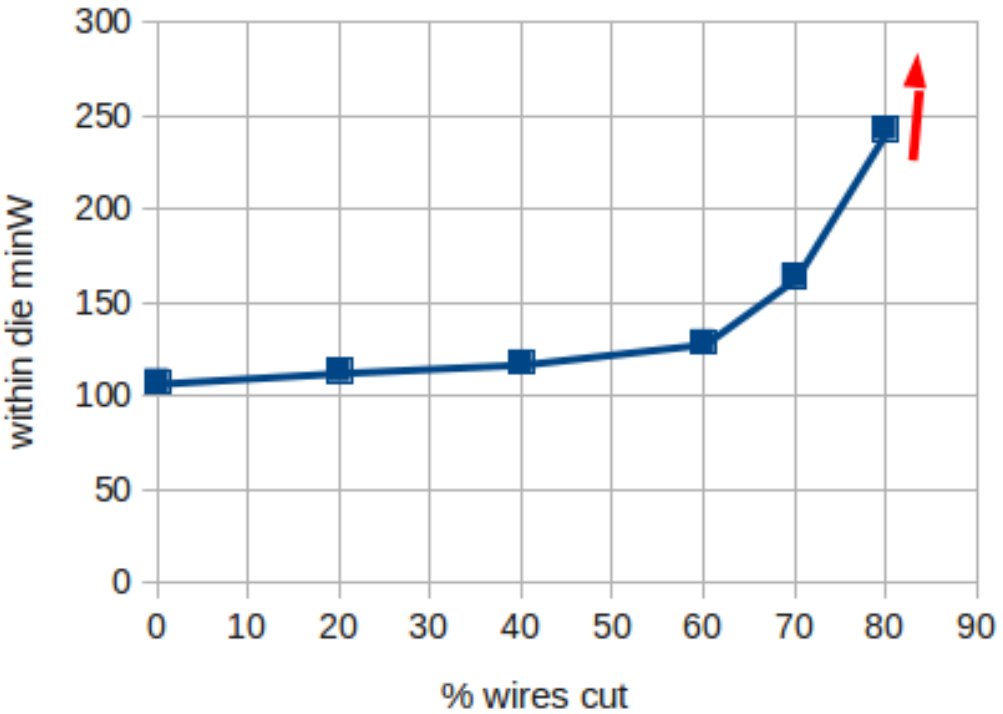


- Rapid degradation after 60% wires cut, limited by interposer bandwidth

Delay increase = 1ns and number of cuts = 3 (4 dice)

Routability vs. Interposer Wiring Bandwidth

Minimum channel width vs % wires cut

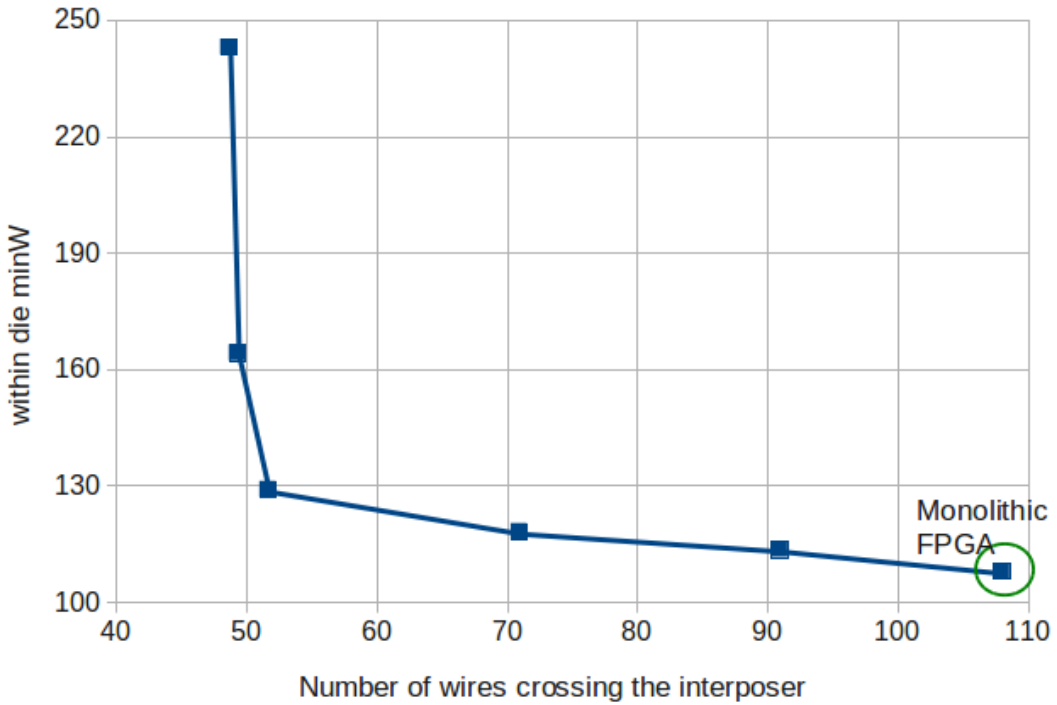


Delay increase = 1ns and number of cuts = 3 (4 dice)

- Very poor routability after 80% wires cut, routability dominated by interposer bandwidth

Routability vs. Interposer Wiring Bandwidth

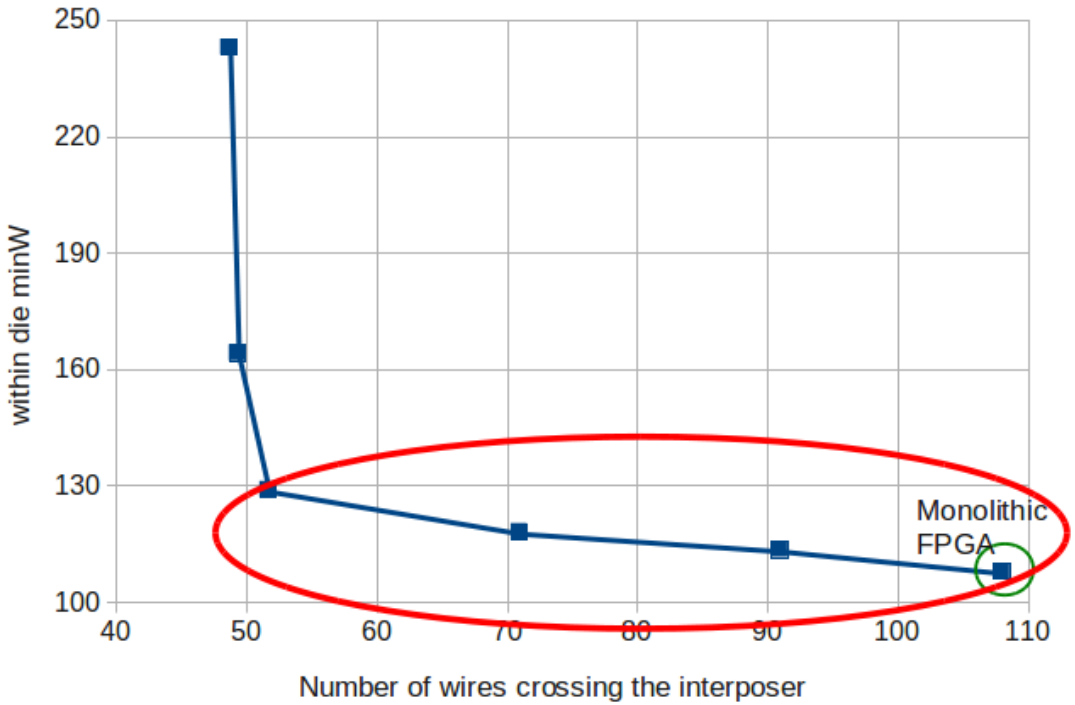
Minimum channel width vs geometric mean of the number of wires crossing the interposer



Delay increase = 1ns and number of cuts = 3 (4 dice)

Routability vs. Interposer Wiring Bandwidth

Minimum channel width vs geometric mean of the number of wires crossing the interposer

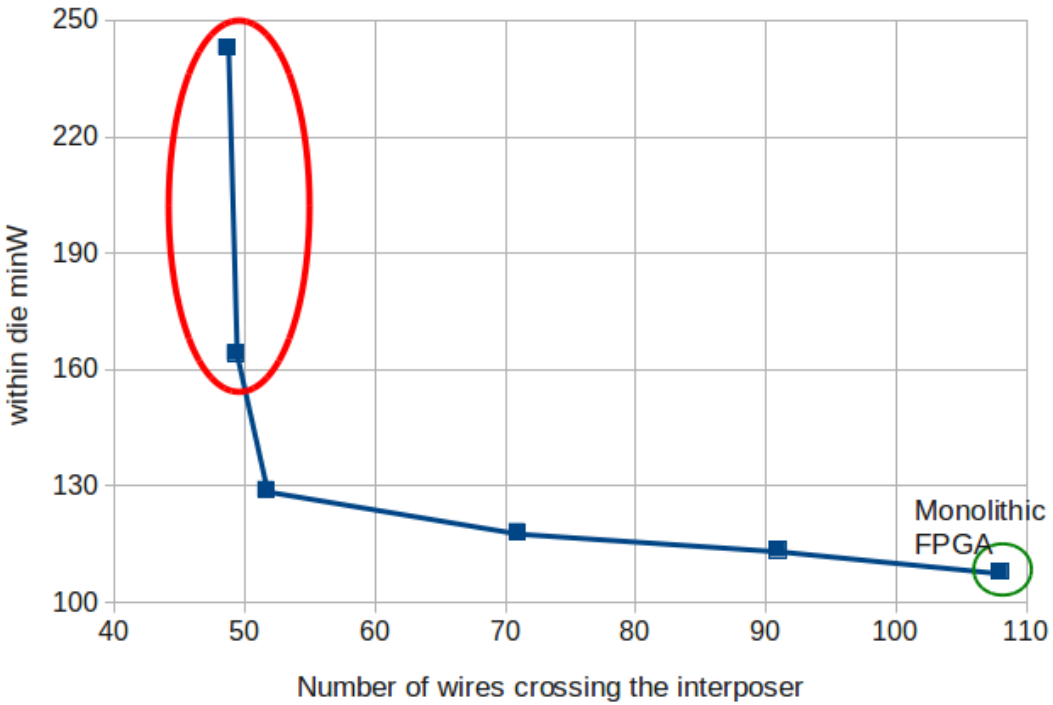


Delay increase = 1ns and number of cuts = 3 (4 dice)

- Gentle impact on minW

Routability vs. Interposer Wiring Bandwidth

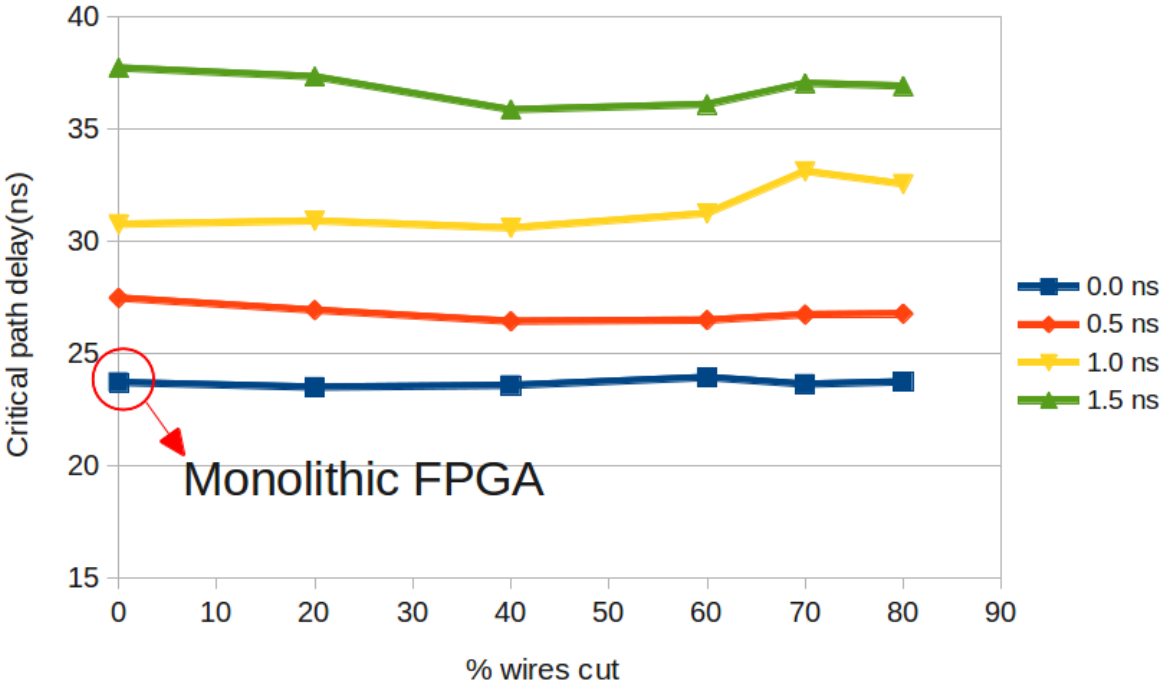
Minimum channel width vs geometric mean of the number of wires crossing the interposer



Delay increase = 1ns and number of cuts = 3 (4 dice)

- Interposer wires dominate within-die minW

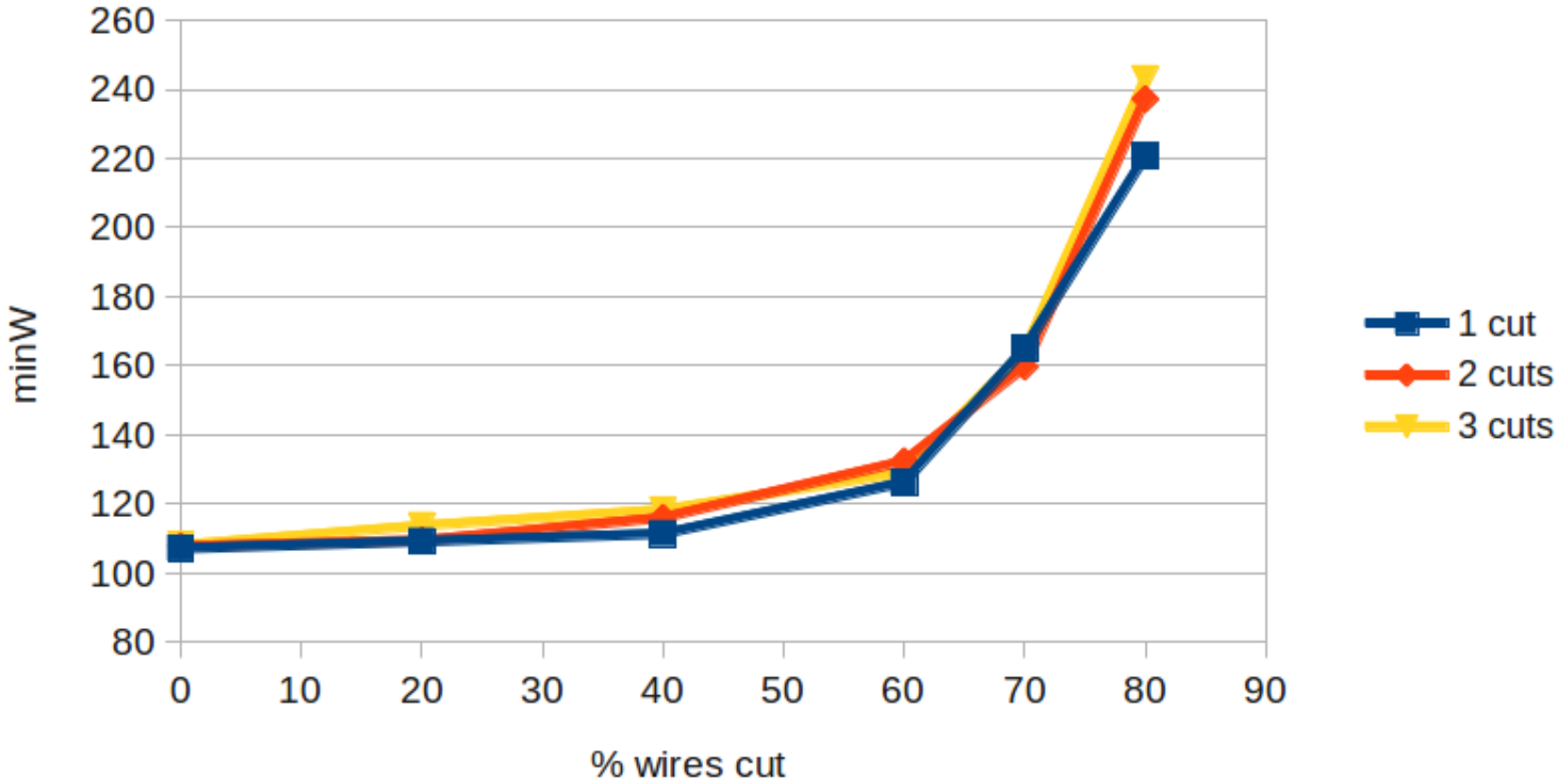
Circuit speed vs interposer delay



Number of cuts = 3 (4 dice)

- Interposer delay has a large impact on speed
- Critical path crosses interposer multiple times

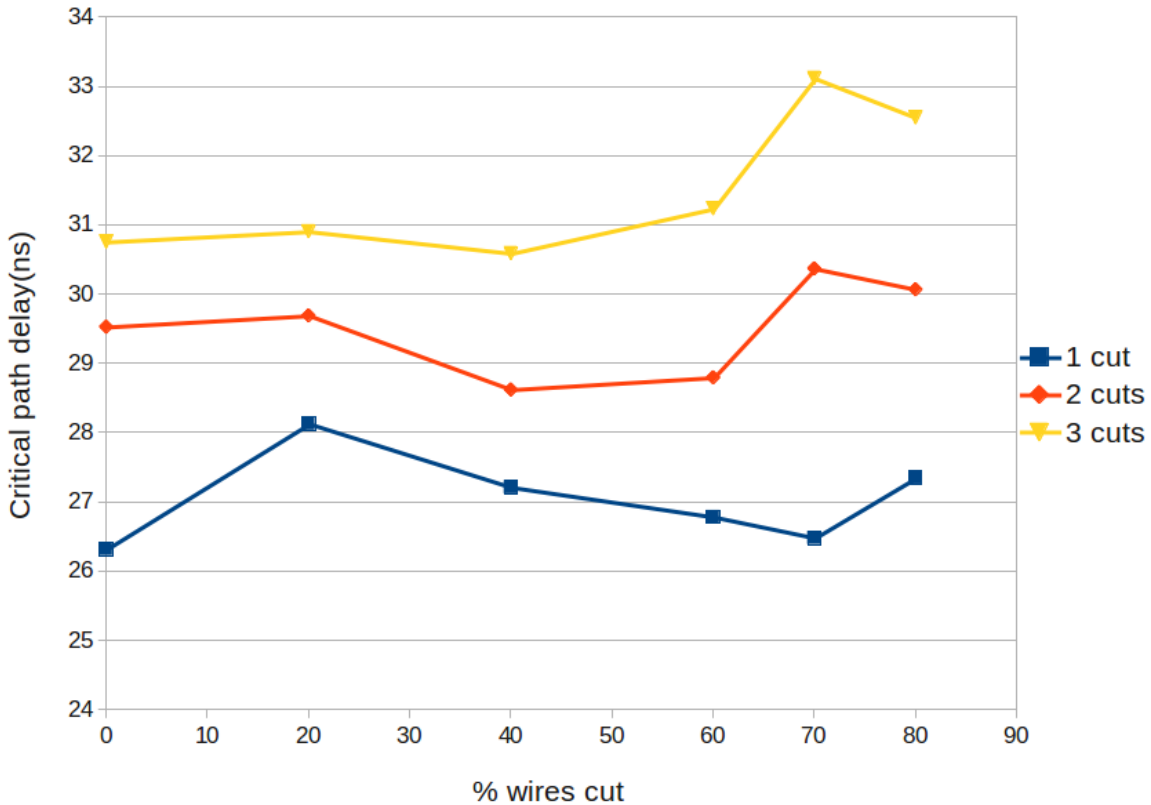
Impact of number of dice



Increased delay = 1ns

- Number of dice has little impact on routability

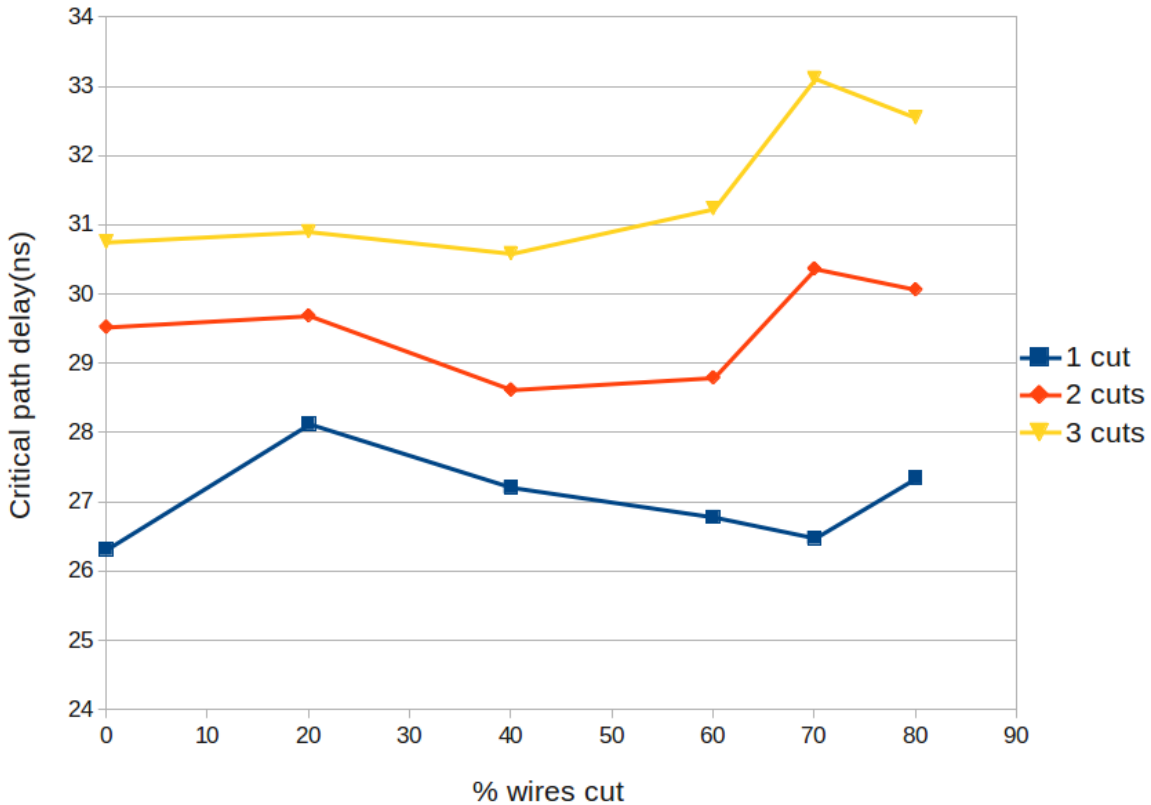
Impact of number of dice



Increased delay = 1ns

- Number of interposer wires has little effect on circuit speed

Impact of number of dice



Increased delay = 1ns

- Number of interposer wires has little effect on circuit speed
- Number of dice has a significant impact on speed

Introduction

Architecture Models

CAD Enhancements

Architecture Results

Conclusion and Future Work

Conclusion

- New cost function improves area-delay by 20%

Conclusion

- New cost function improves area-delay by 20%
- No drastic impact when interposer provides only 40% of intra-die routing capacity

Conclusion

- New cost function improves area-delay by 20%
- No drastic impact when interposer provides only 40% of intra-die routing capacity
- Routability is dominated by interposer below 40% of within-die routing capacity

Conclusion

- New cost function improves area-delay by 20%
- No drastic impact when interposer provides only 40% of intra-die routing capacity
- Routability is dominated by interposer below 40% of within-die routing capacity
- Critical path strongly impacted by interposer delay

Conclusion

- New cost function improves area-delay by 20%
- No drastic impact when interposer provides only 40% of intra-die routing capacity
- Routability is dominated by interposer below 40% of within-die routing capacity
- Critical path strongly impacted by interposer delay
- Circuit delay affected by number of dice, but not by routing capacity

Future work

Interposer wires are scarce...

Future work

Interposer wires are scarce...

- Investigate different switch structures for wires crossing the interposer

Future work

Interposer wires are scarce...

- Investigate different switch structures for wires crossing the interposer
- Investigate alternative CAD flows, such as adding a partitioning step before placement

Future work

Interposer wires are scarce...

- Investigate different switch structures for wires crossing the interposer
- Investigate alternative CAD flows, such as adding a partitioning step before placement

Bigger circuits (Titan benchmarks)

Thank you!
Questions?

andre.hahn@usp.br

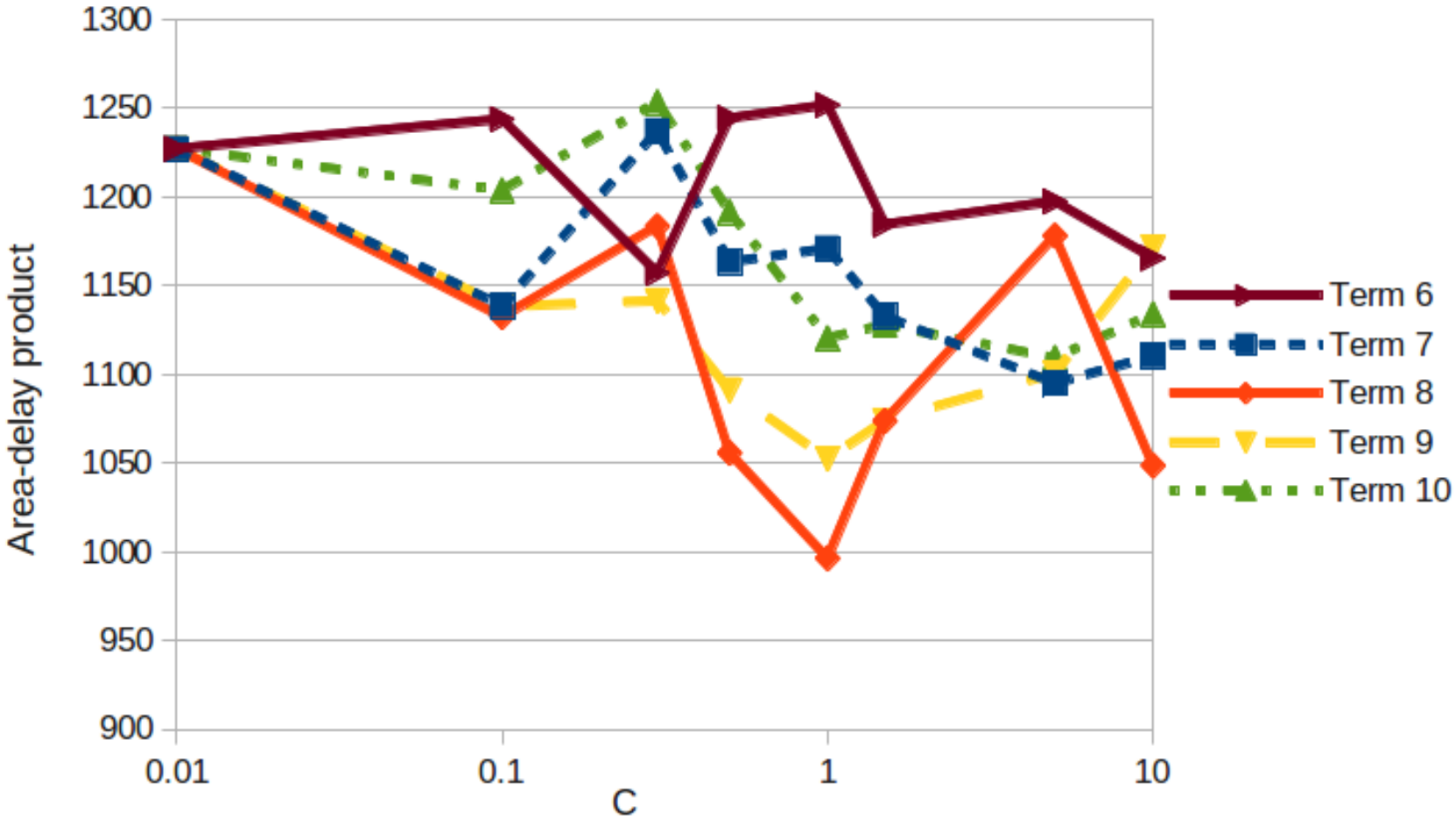
Extra slides

Placer routability cost enhancement results

Term	minW	crit_path(ns)	Area-delay
None	124.6	9.8	1227
# of crossings only	122.9	9.4	1157
Height only	112.7	9.8	1109
Crossings + height	107.3	9.3	996

Best performance for each term.

Constant sweep



Circuit characteristics

Circuit	# in	# out	#6-LUTs	#FFs	#Mults	#Mem
bgm	257	32	30089	5362	11	0
LU8PEEng	114	102	21954	6630	8	9
LU32PEEng	114	102	75530	20898	32	9
mcml	36	33	99700	53736	30	10
mkDelayWorker32B	511	553	5580	2491	0	9
stereovision0	157	197	11462	13405	0	0
stereovision1	133	197	11462	13405	152	0
stereovision2	149	182	29849	18416	564	0