VLSI Micro-Architectures for High-Radix Crossbar Schedulers

Giorgos Passas, Manolis Katevenis, Dionisios Pnevmatikatos

presentation by Giorgos Passas

FORTH & Univ. of Crete, Greece

Motivation

-- Academic Curiosity

* How large crossbars can grow?

Outline

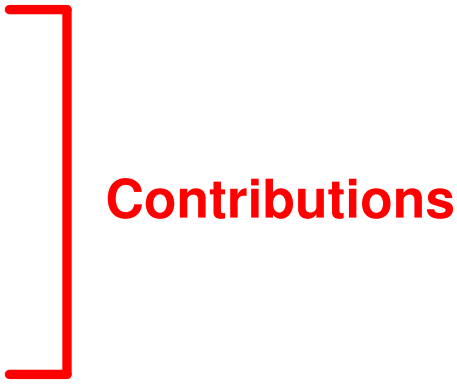
1) Background
* Crossbar datapath

2) Scheduling System
* High-level operation

3) iSLIP Overview
* Algorithm & Circuit

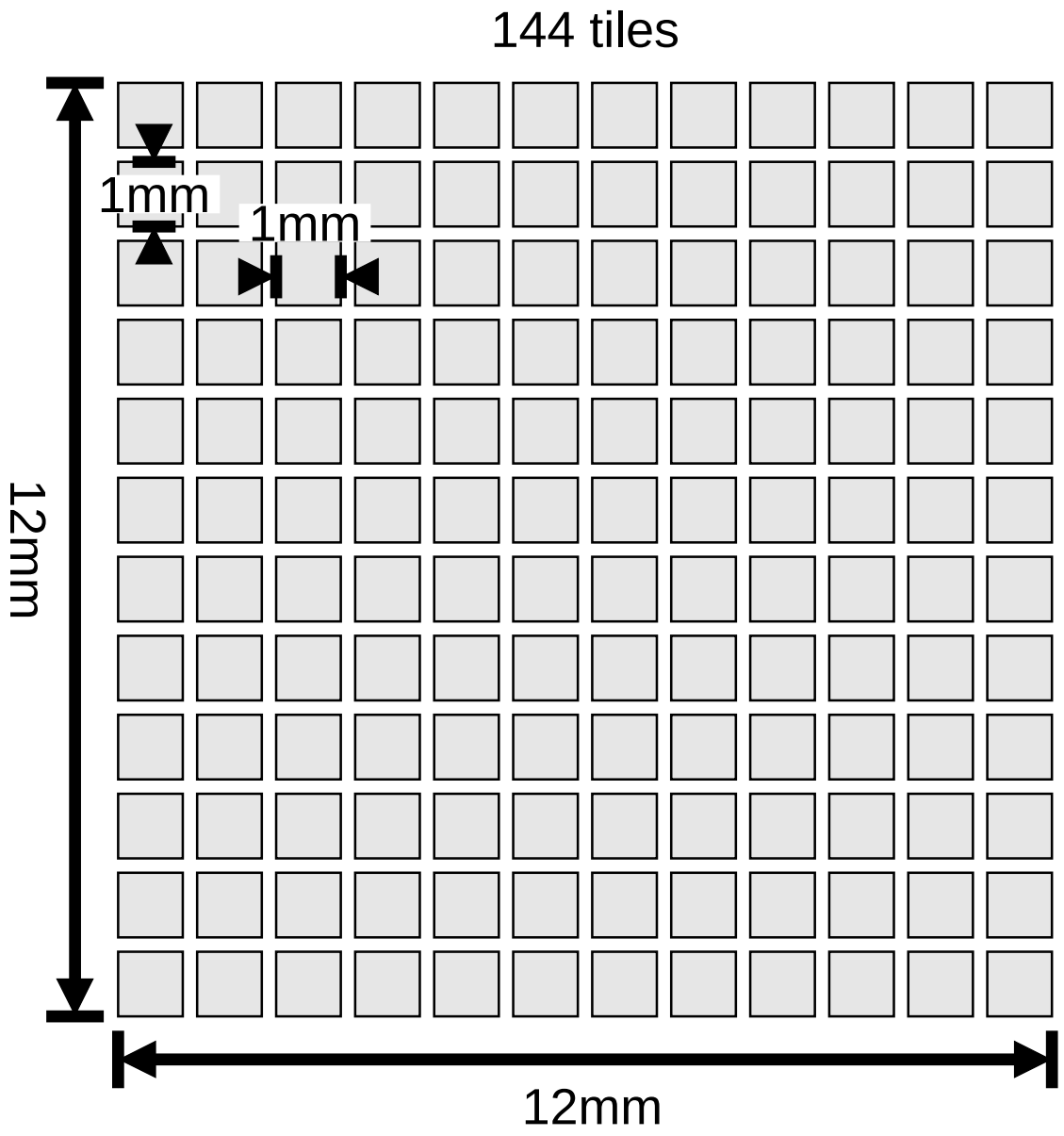
4) iSLIP Area Cost Analysis
* Wiring limitations at high radices

5) Cross Micro-Architecture
* Overcoming the wiring limitations



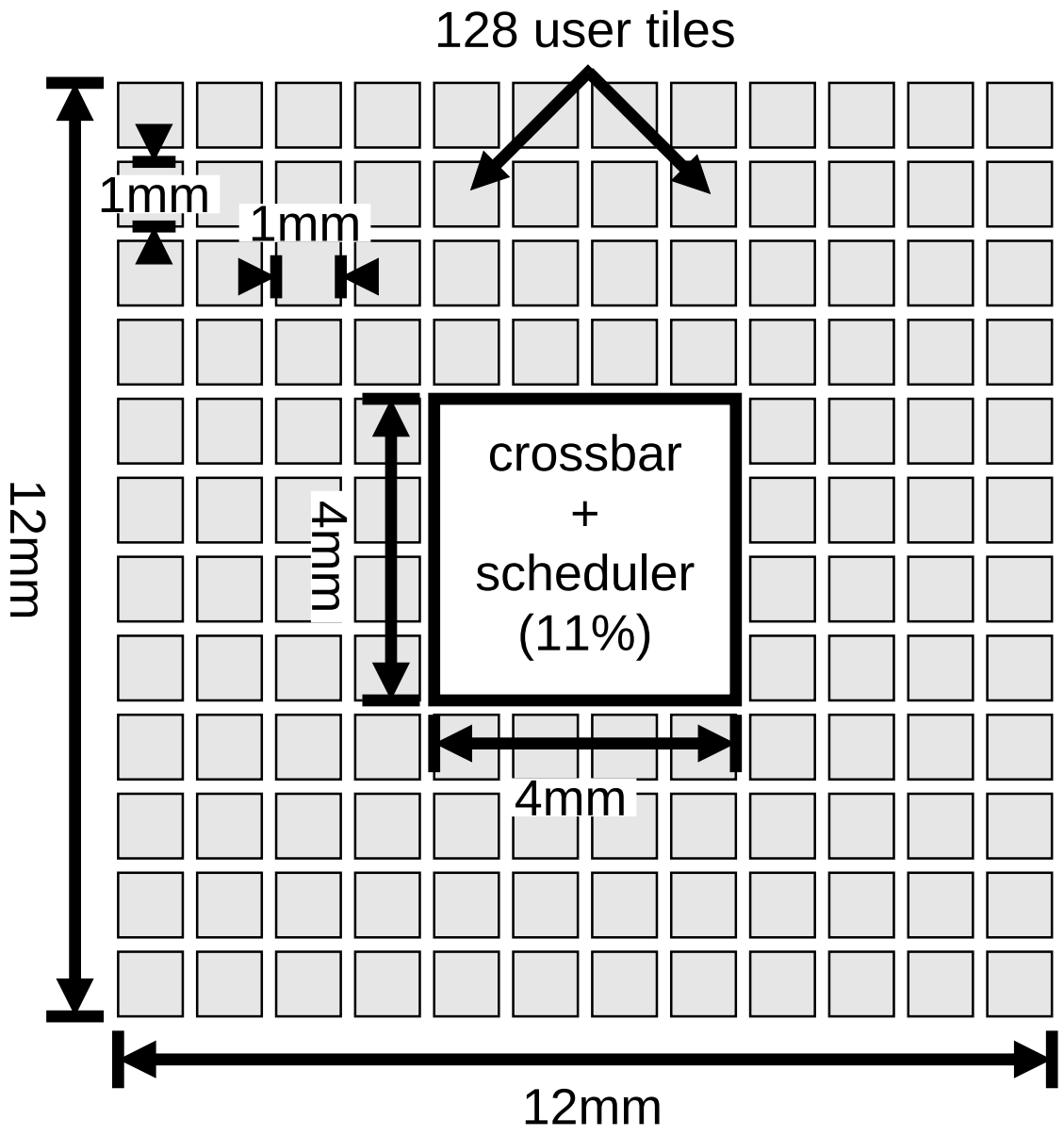
Background: Chip Floorplan

90nm CMOS, 9 metal layers (M1-M9)



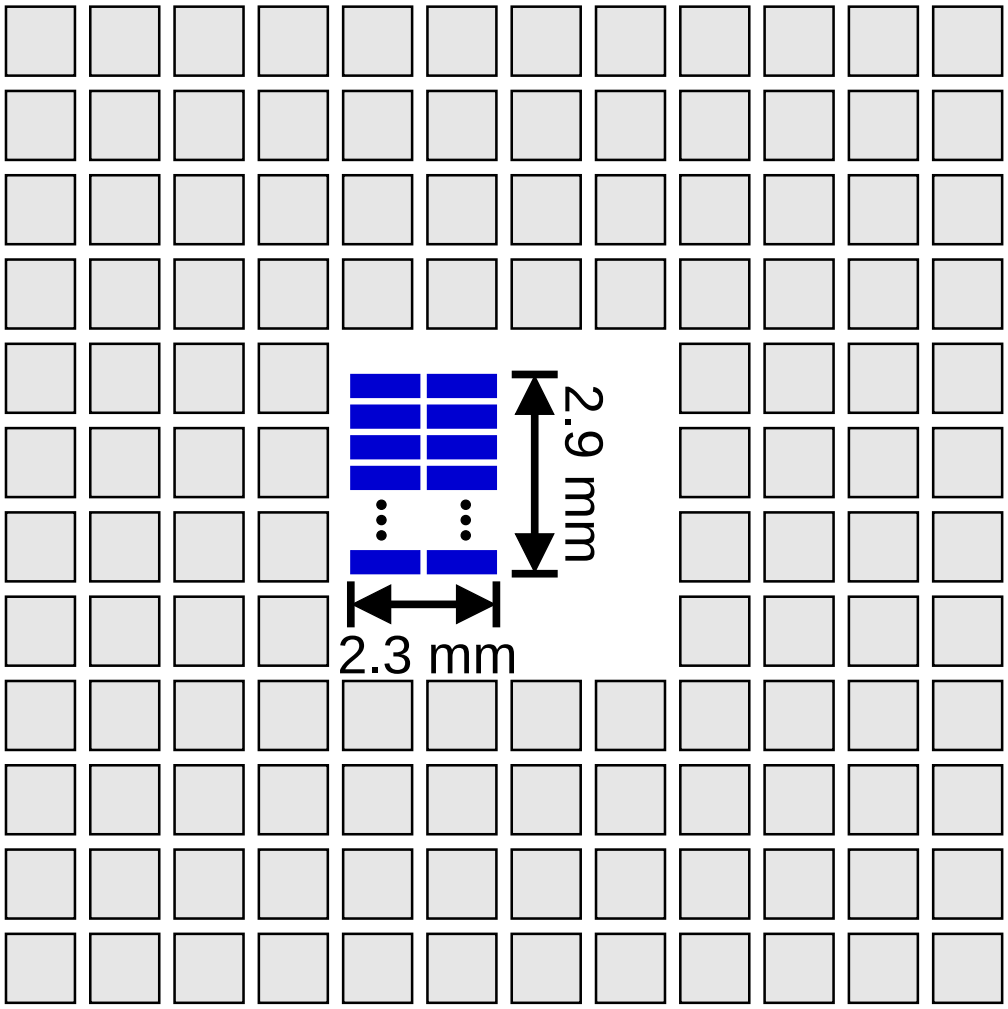
Background: Chip Floorplan

90nm CMOS, 9 metal layers (M1-M9)



Background: Crossbar Datapath

Passas, Katevenis, Pnevmatikatos: NOCS 2010



-- 128 ports

-- 32bit

-- 6.6mm^2

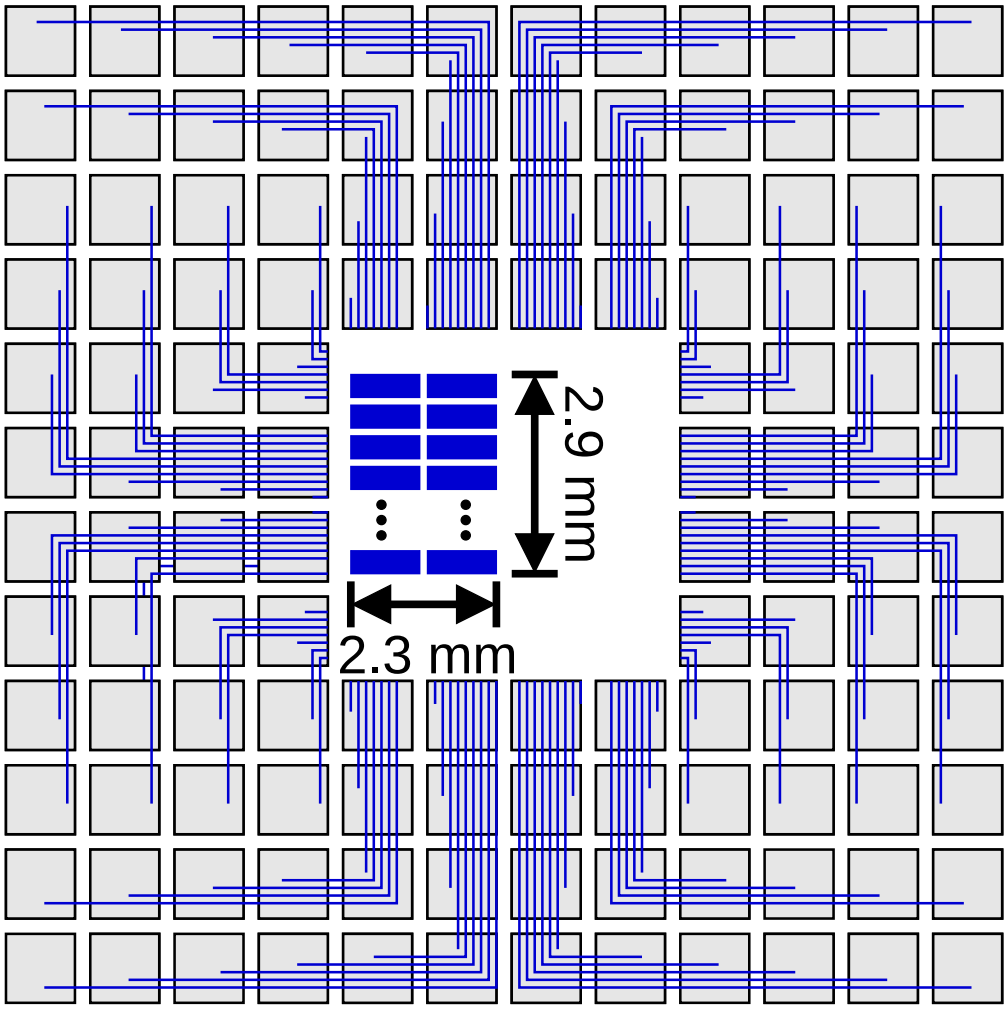
-- 3 pipe stages

-- 750MHz
- 48Gb/s ports

-- 3.5Watts
- xput=1, toggle=0.5

Background: Crossbar Datapath

Passas, Katevenis, Pnevmatikatos: NOCS 2010



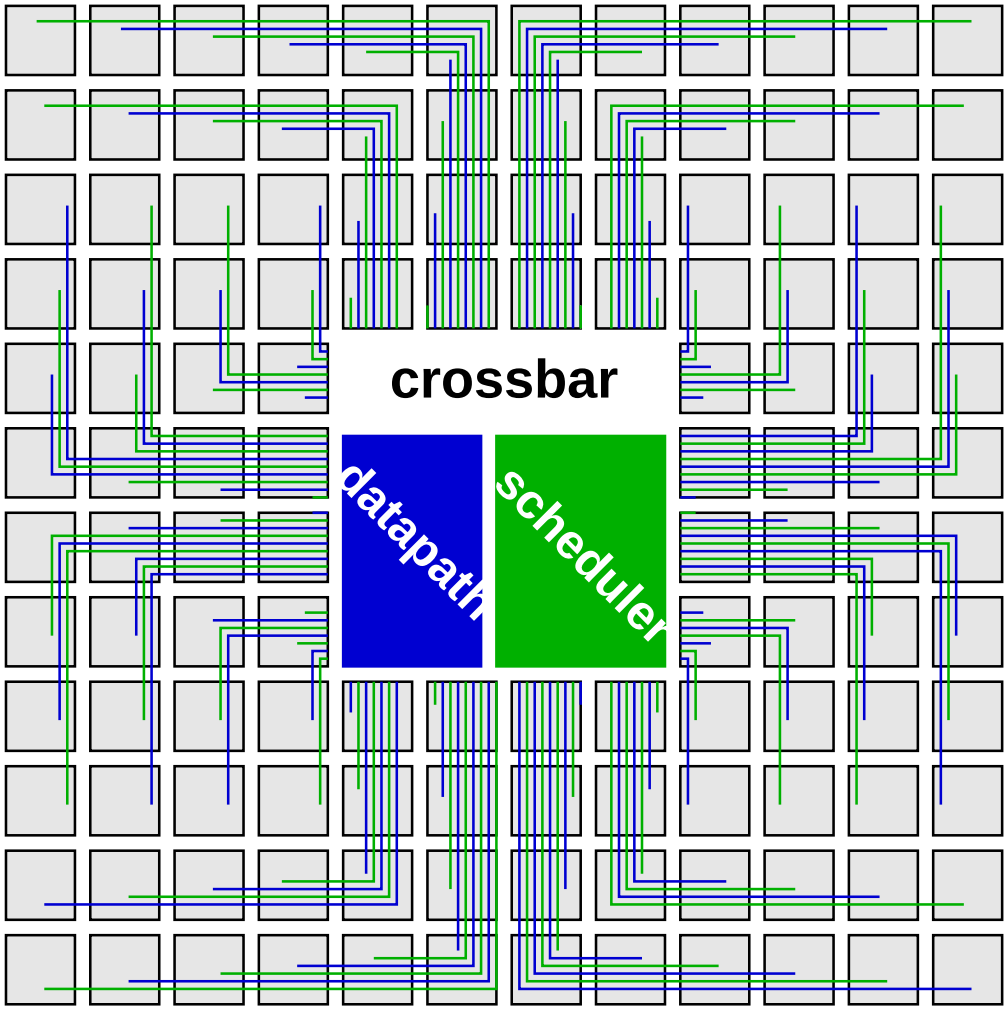
-- M5-M6 over the tiles

-- 2 pipe stages

-- 0.7mm^2 repeaters, FF

-- 1Watt

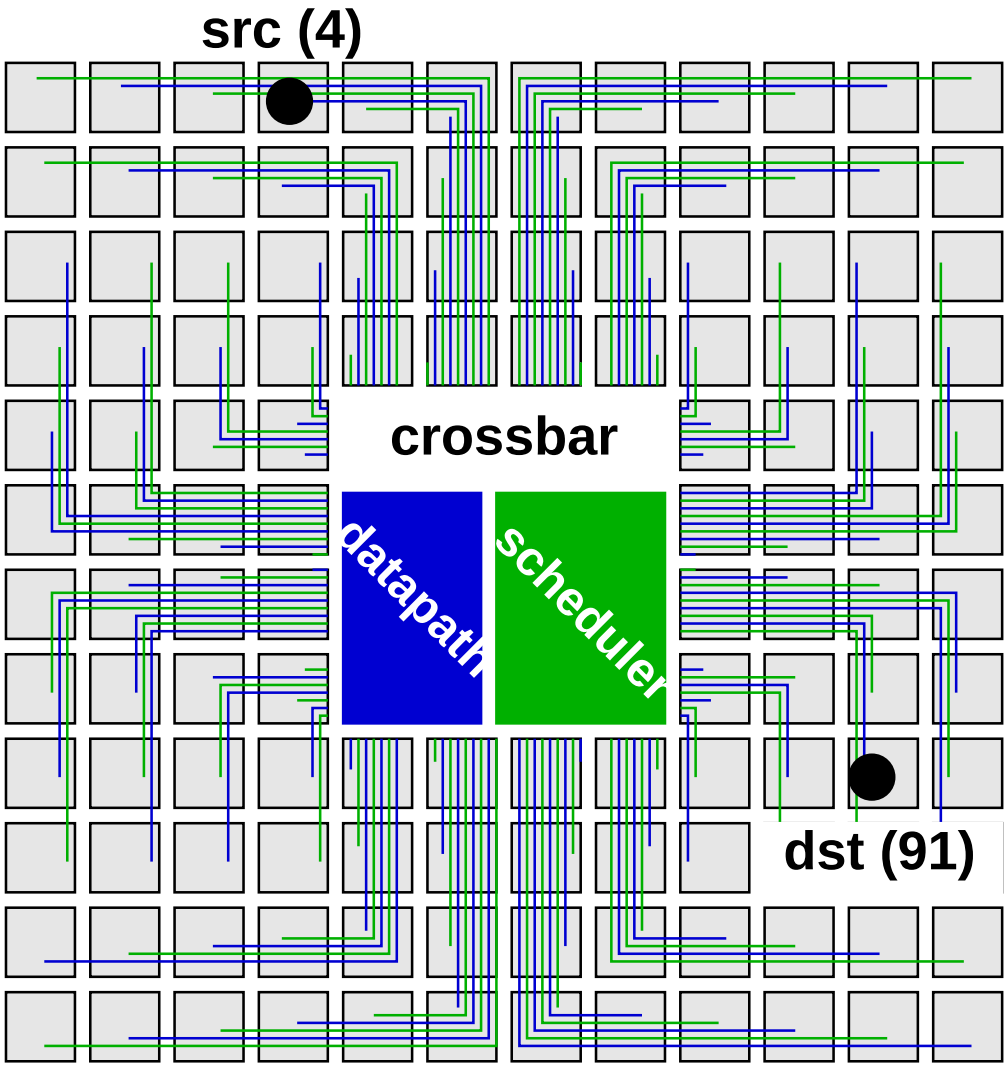
Scheduling System: Floorplan



-- 9mm^2 for scheduler

-- scheduler IO parallel to xbar

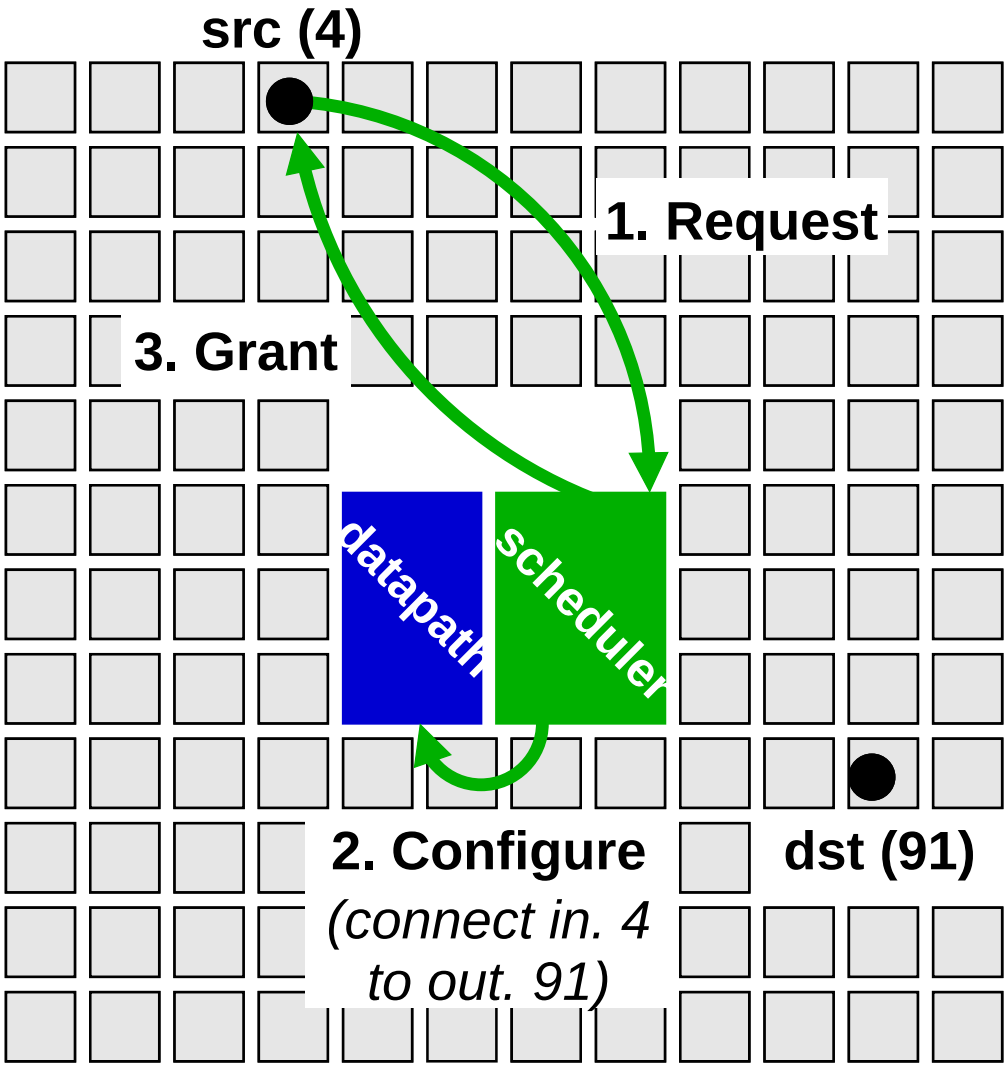
Scheduling System: Operation



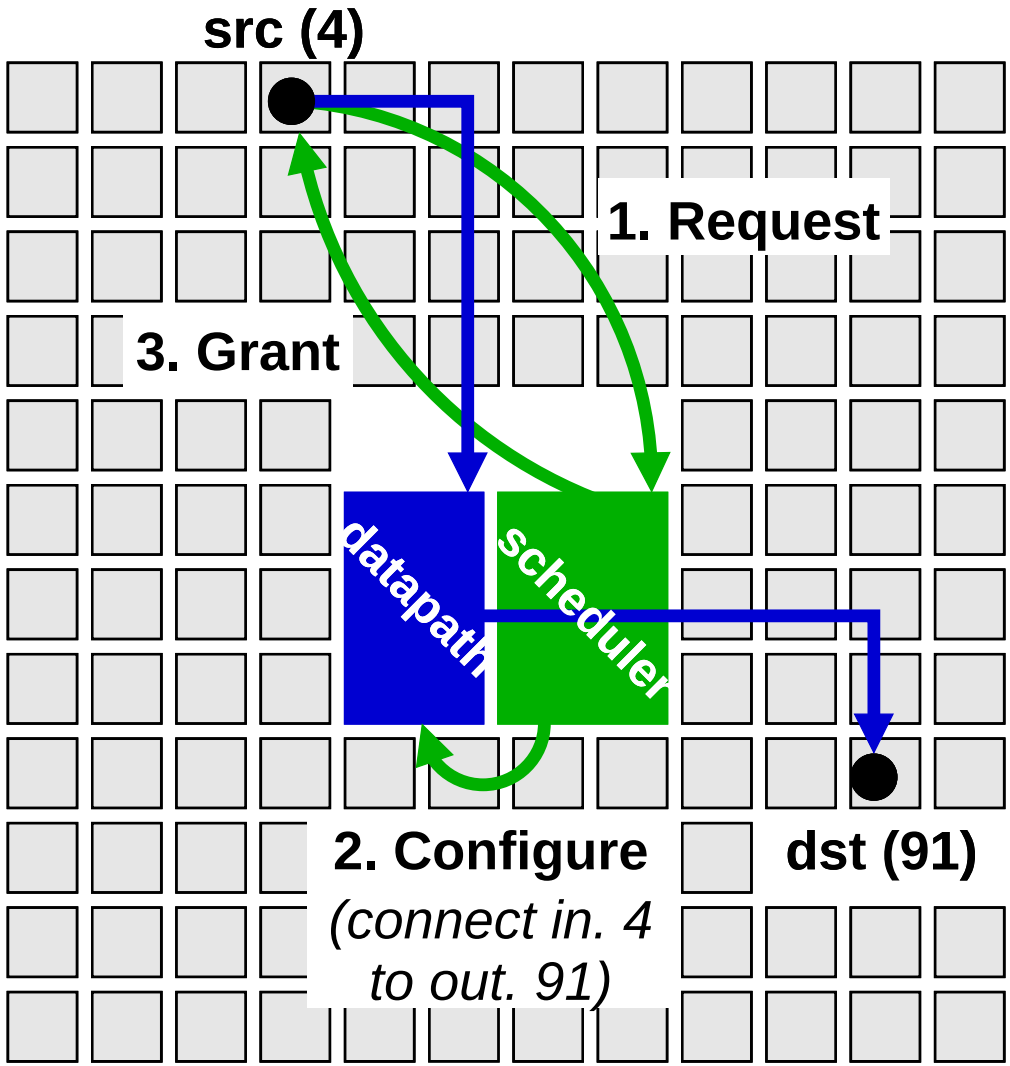
-- src connects to xbar in 4

-- dst connects to xbar out 91

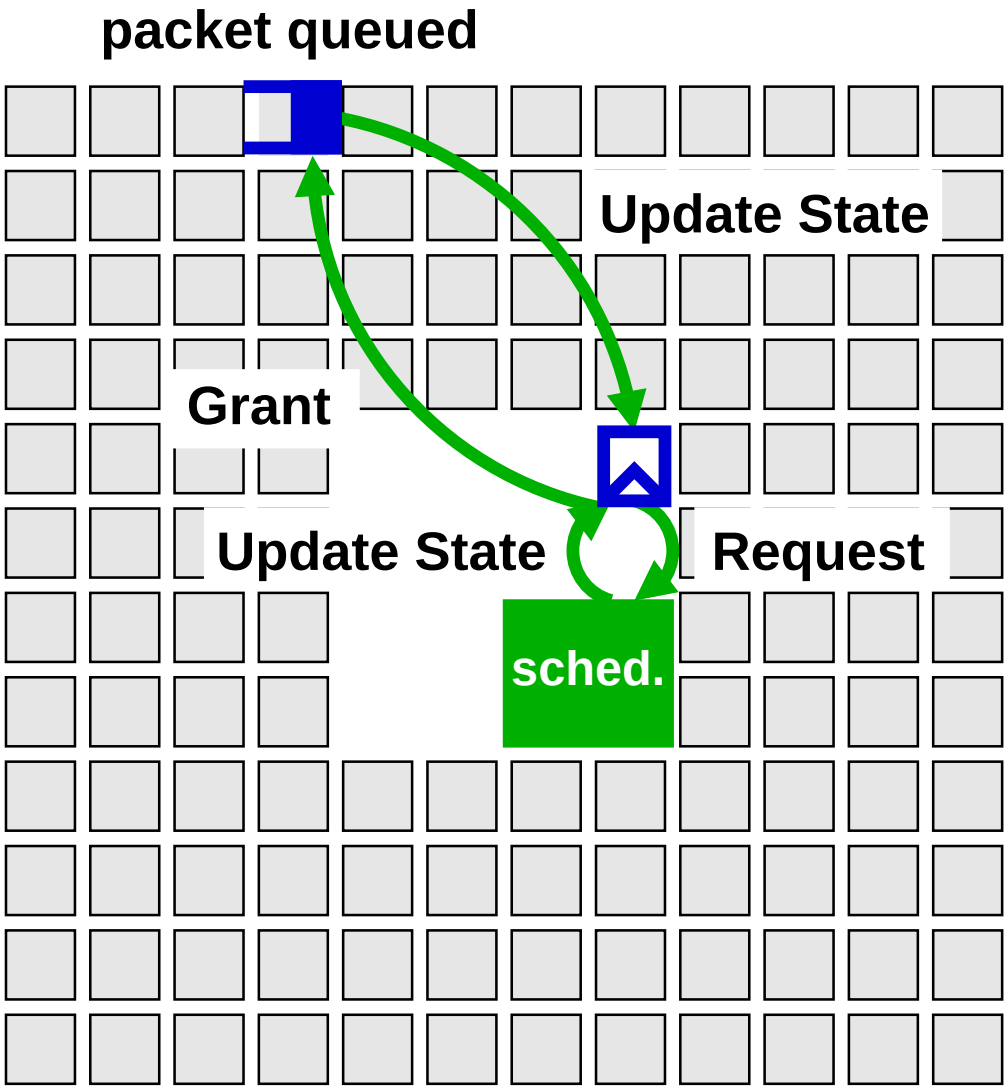
Scheduling System: Operation



Scheduling System: Operation



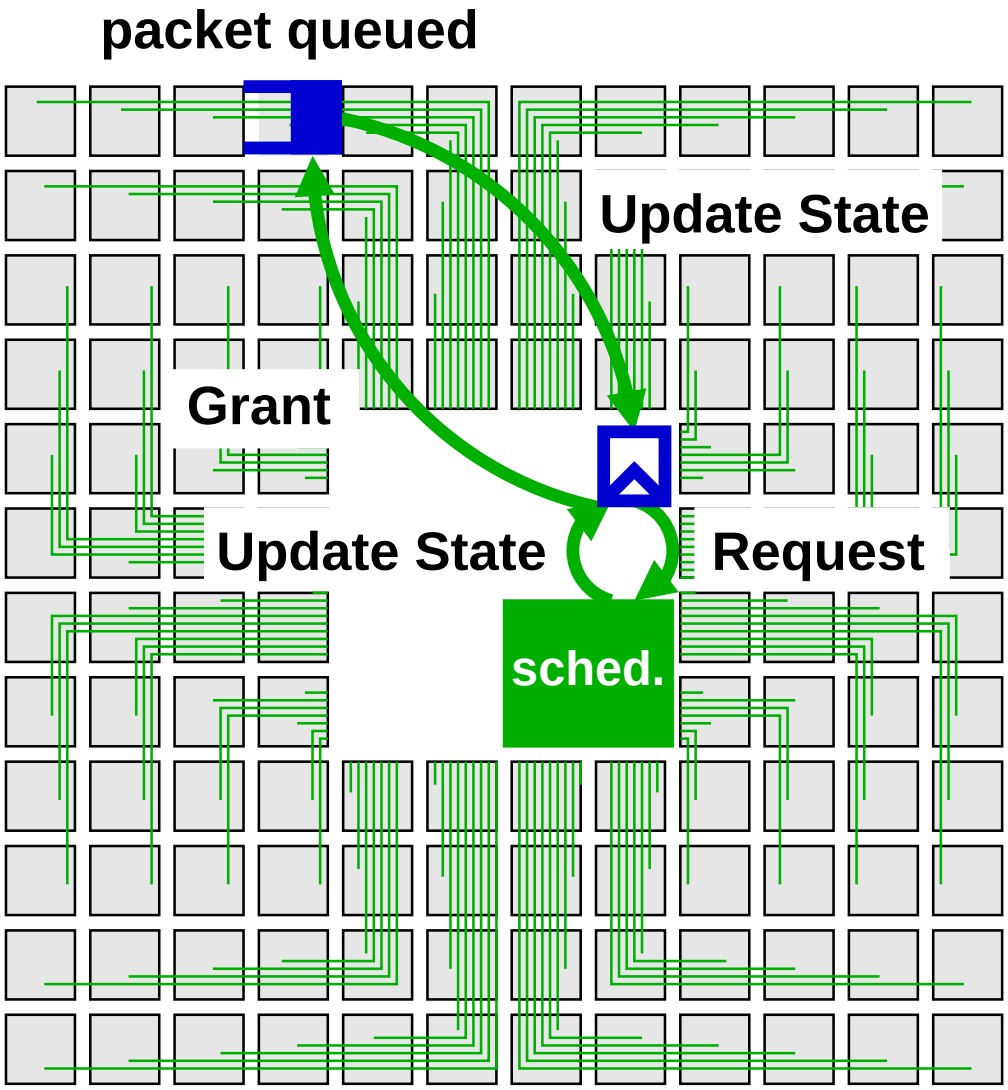
Scheduling System: Operation



-- queues @ tiles

-- queue state @ scheduler

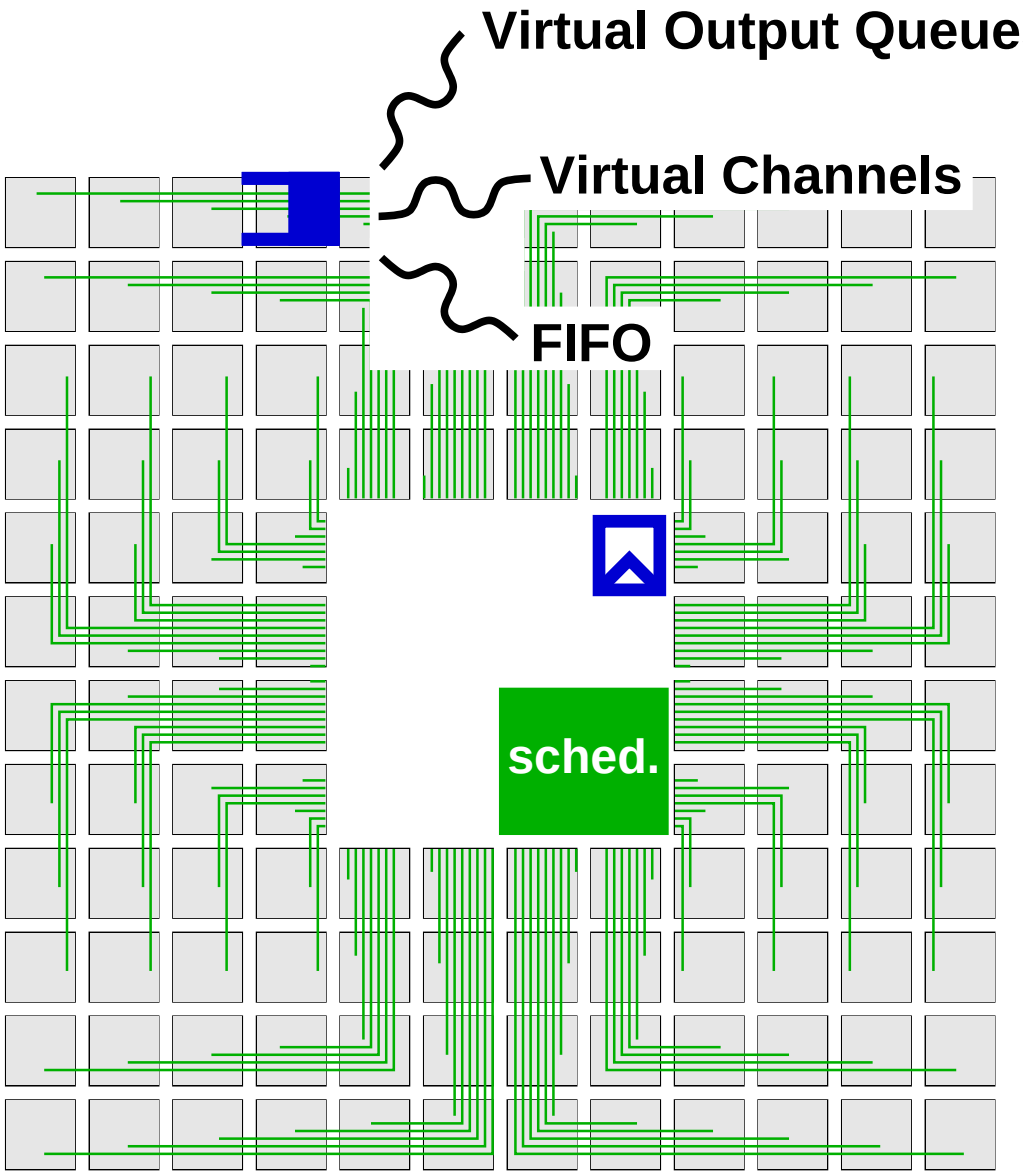
Scheduling System: Operation



-- 7bits / tile / direction

→ cheap intreconnect

Scheduler Design Depends on Queueing Strategy

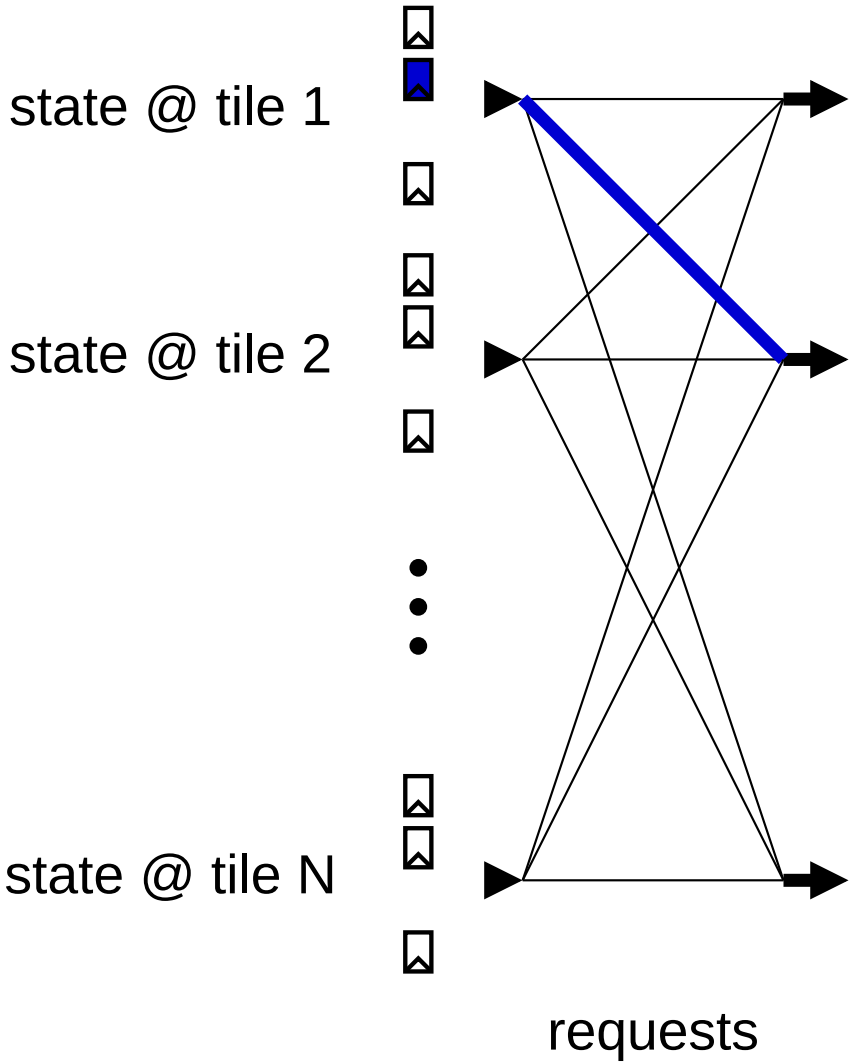


-- this talk: VOQs/iSLIP

-- see paper for FIFO/VCs

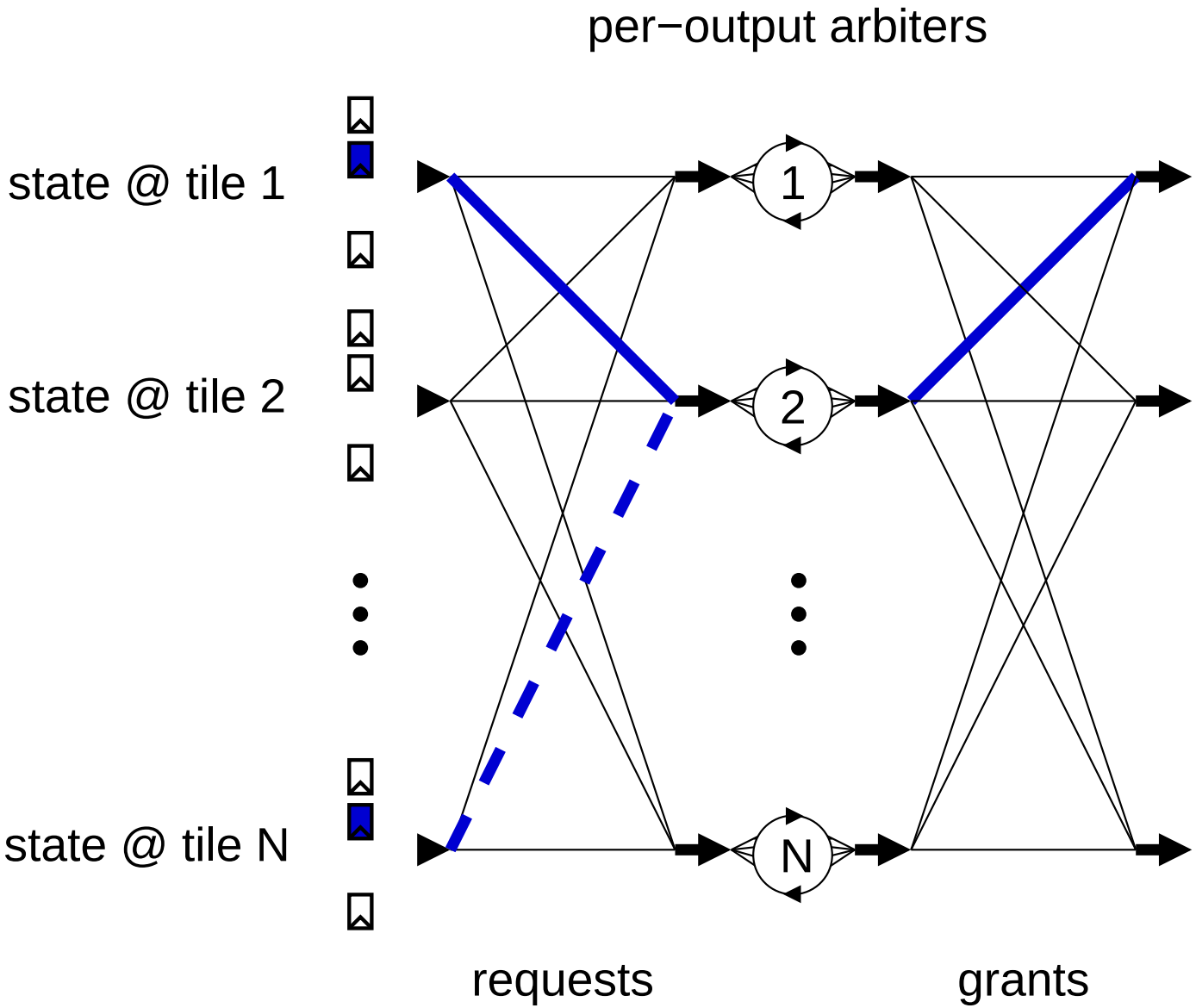
The iSLIP Algorithm (Mckeown, 1999)

case study: N = 128



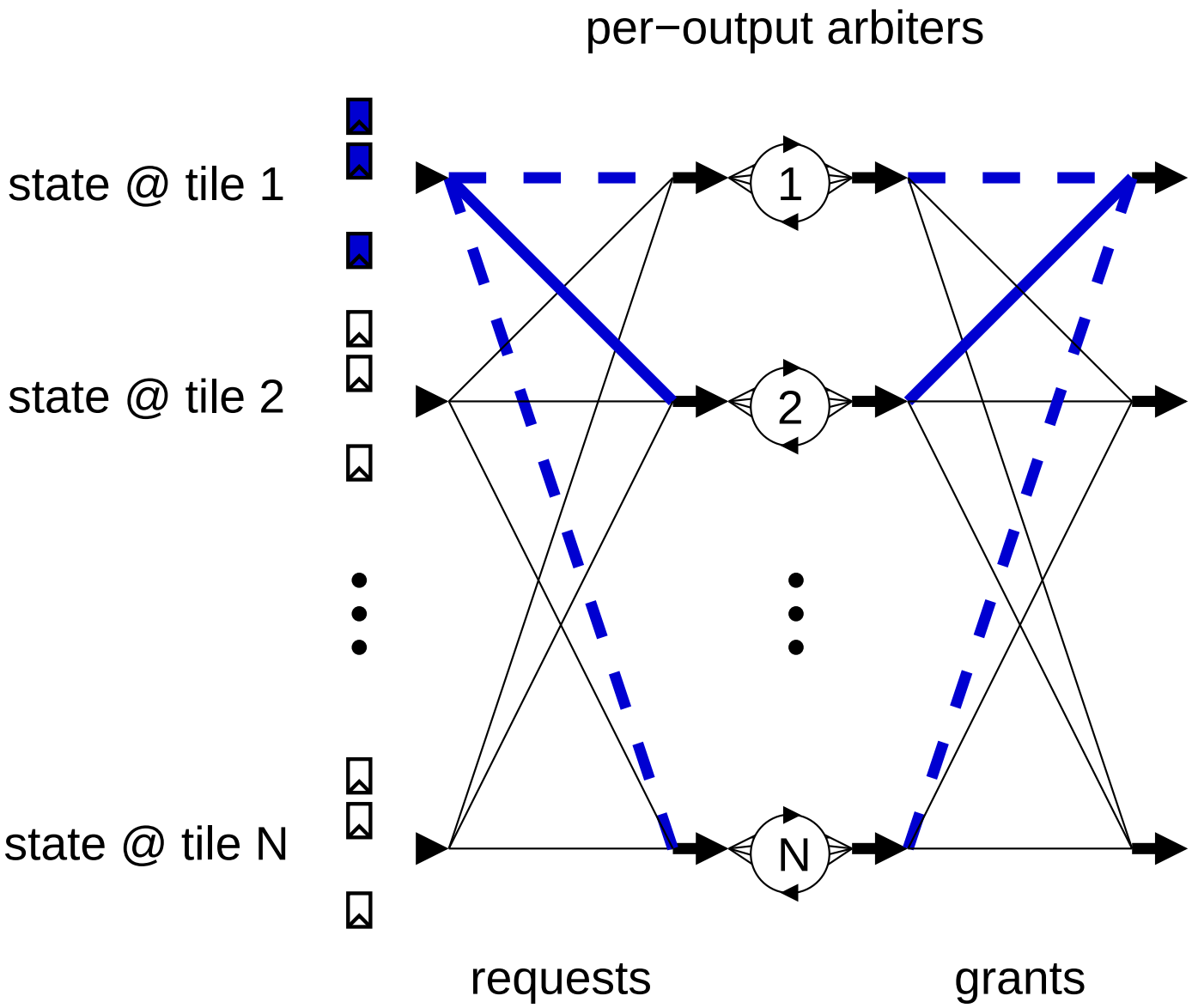
The iSLIP Algorithm (Mckeown, 1999)

case study: N = 128



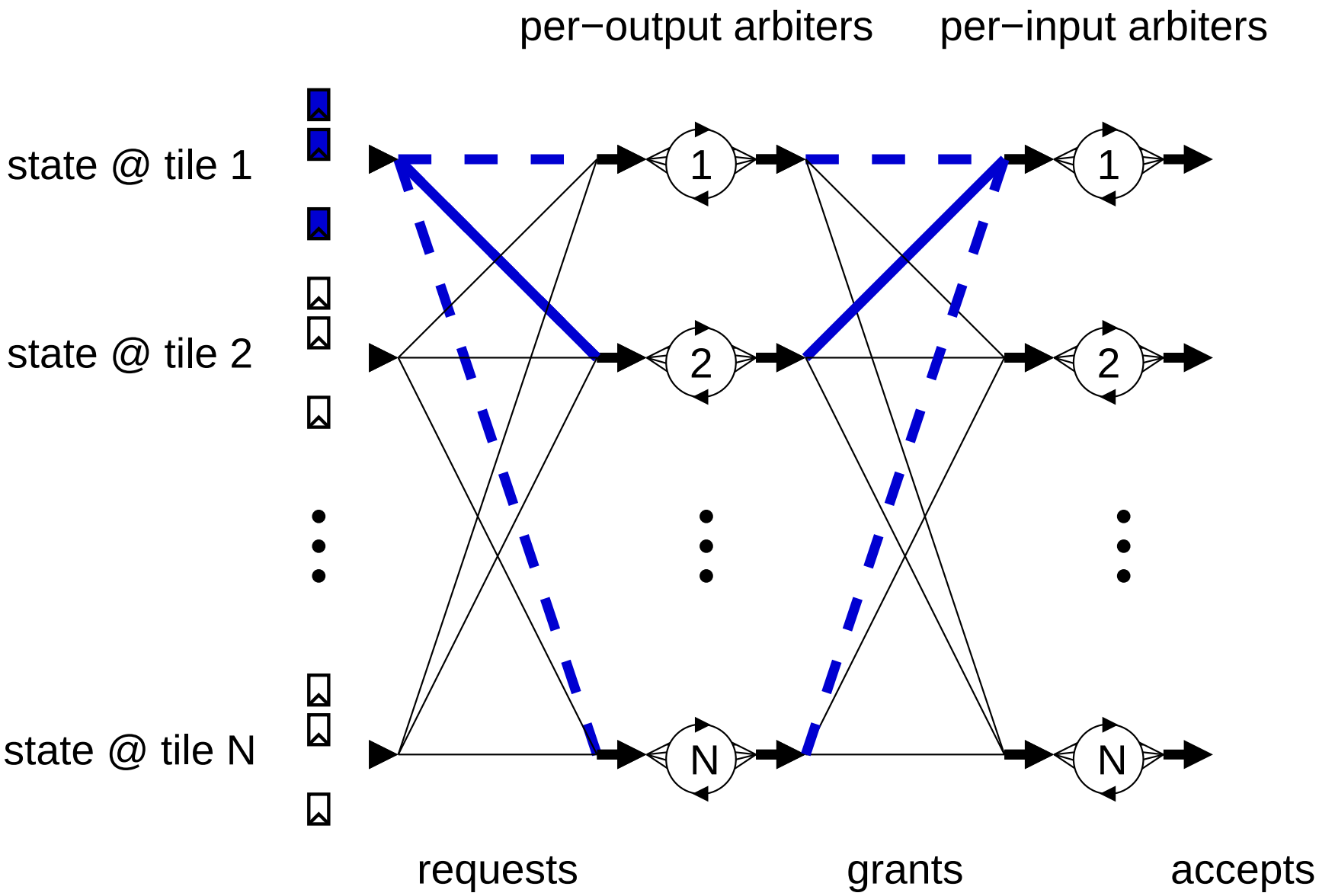
The iSLIP Algorithm (Mckeown, 1999)

case study: N = 128



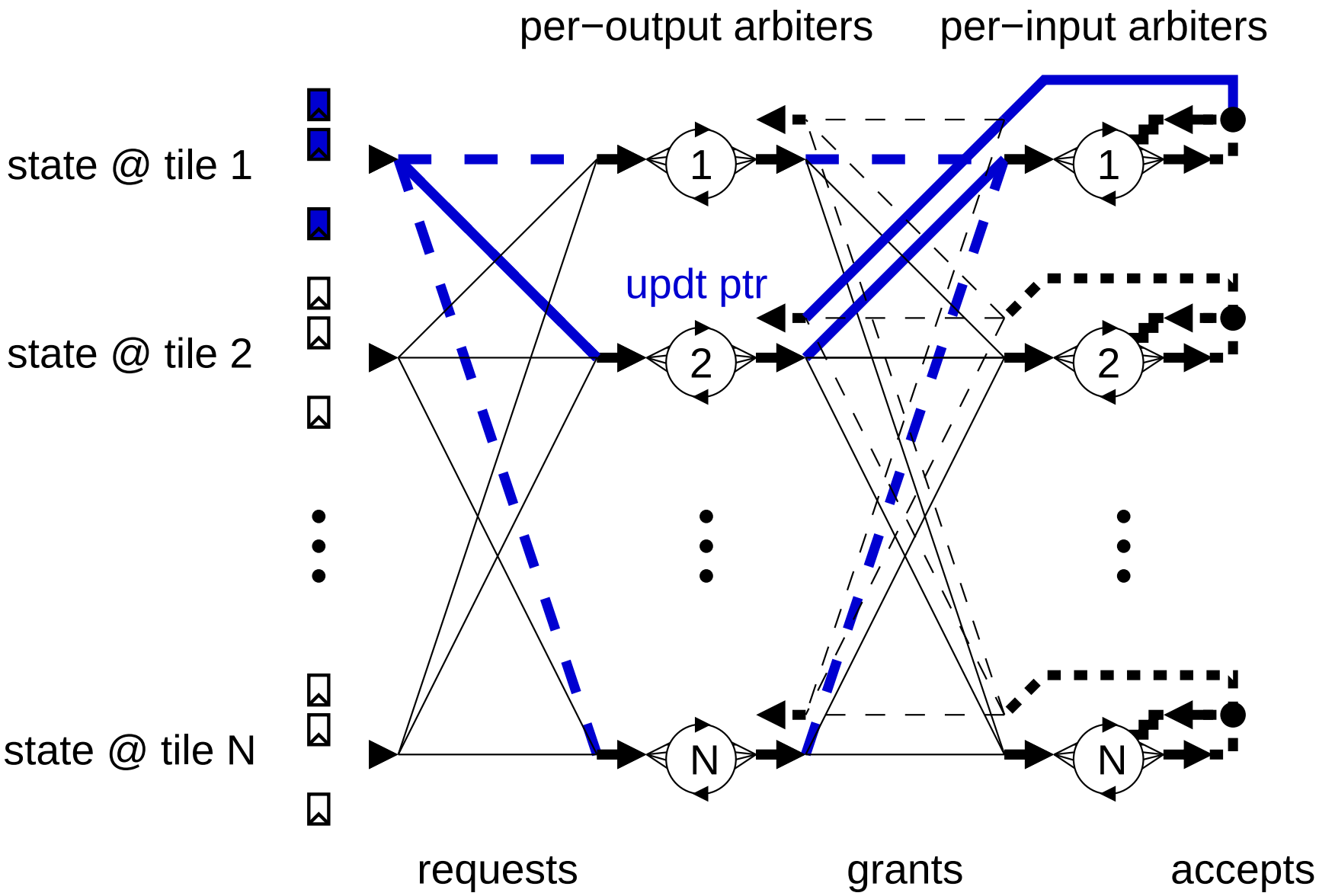
The iSLIP Algorithm (Mckeown, 1999)

case study: N = 128



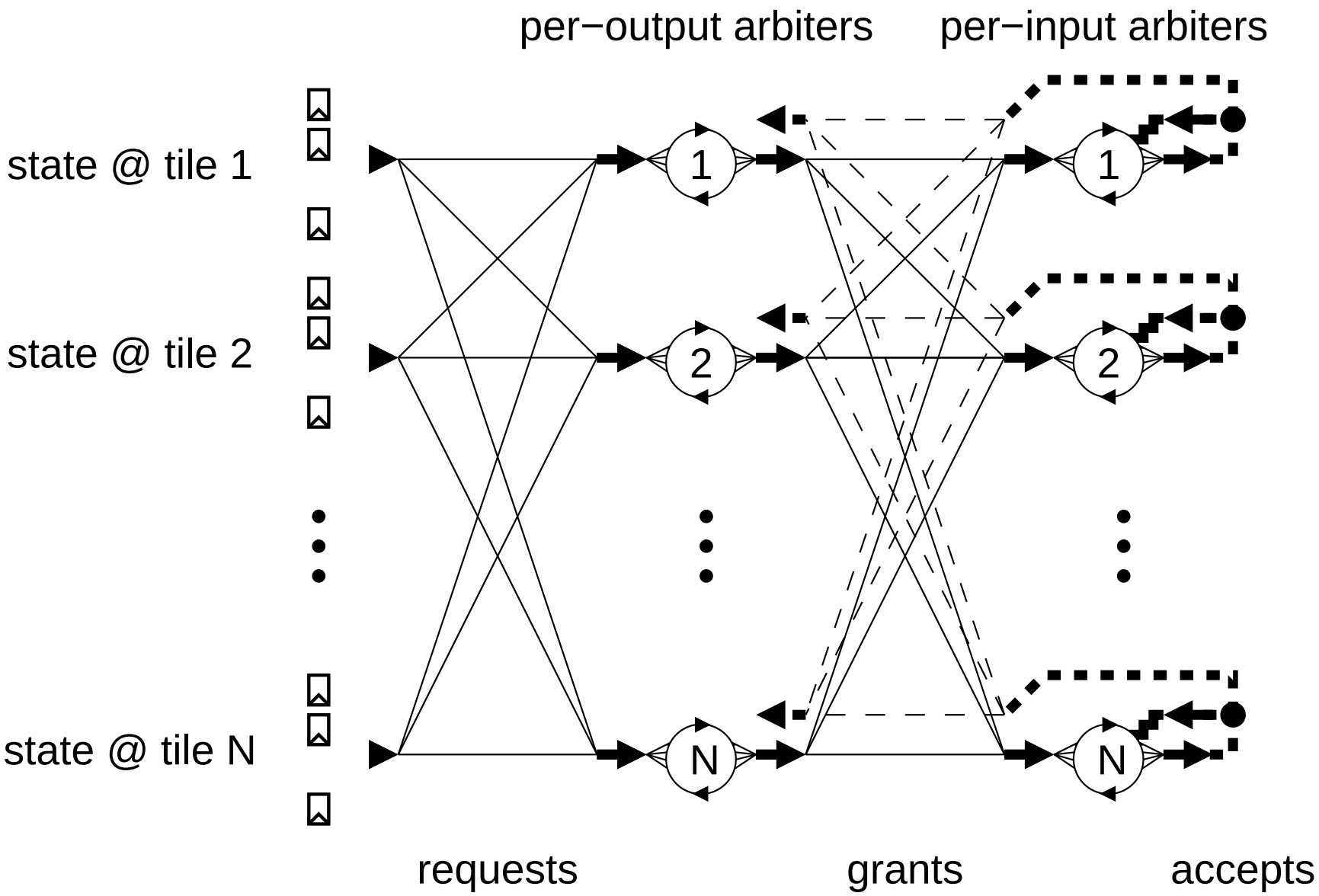
The iSLIP Algorithm (Mckeown, 1999)

case study: N = 128

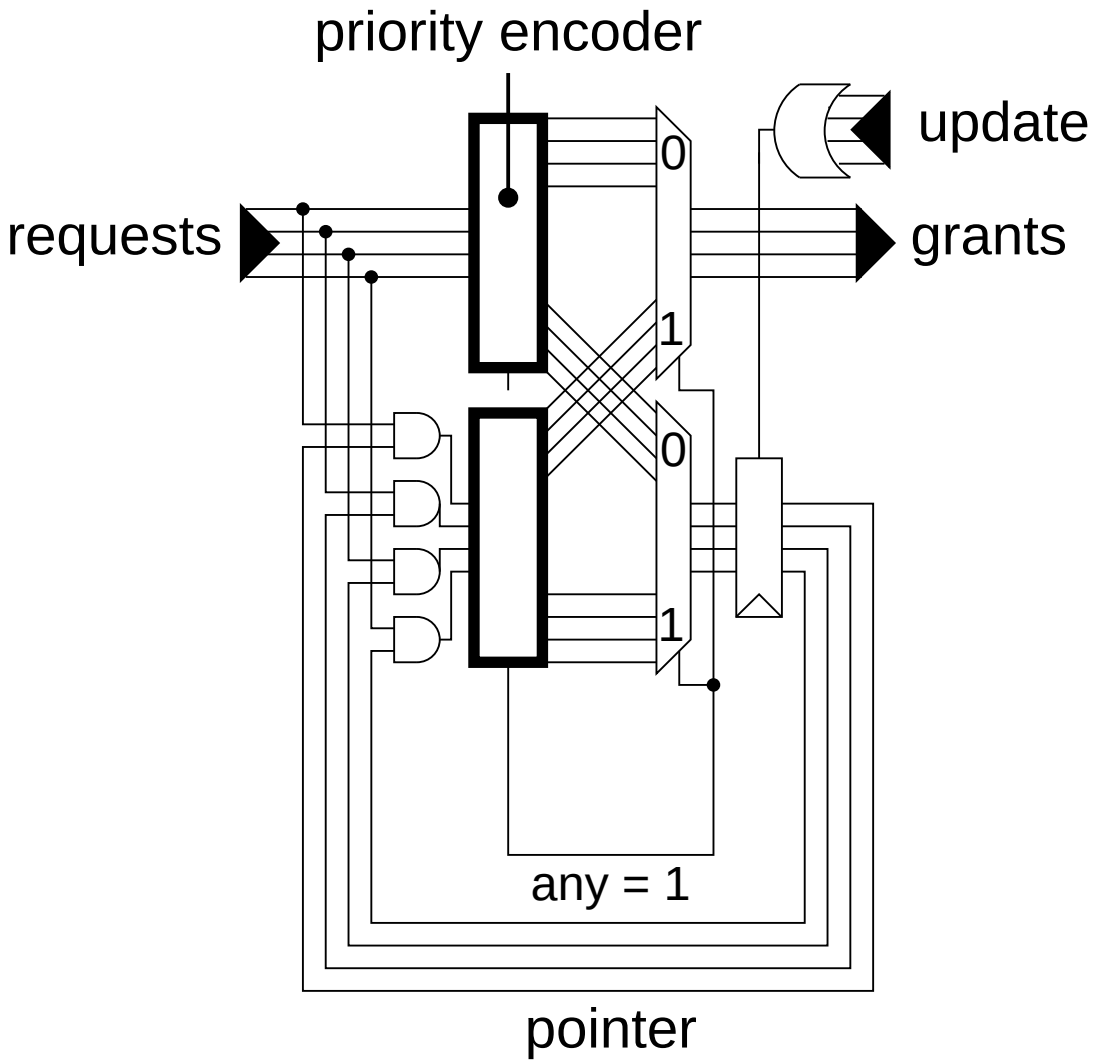


The iSLIP Algorithm (Mckeown, 1999)

2N arbiters, $3N^2$ p2p links

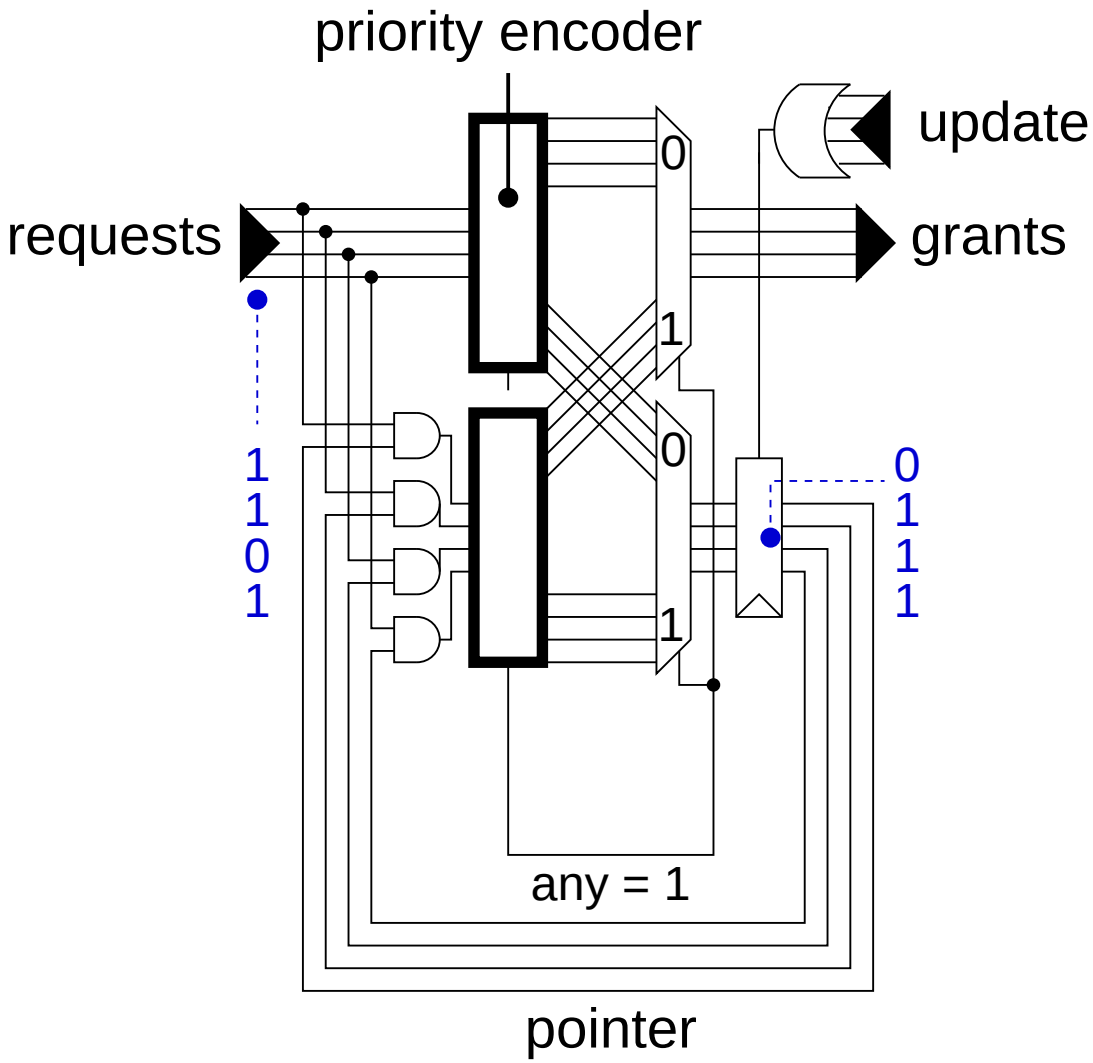


The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



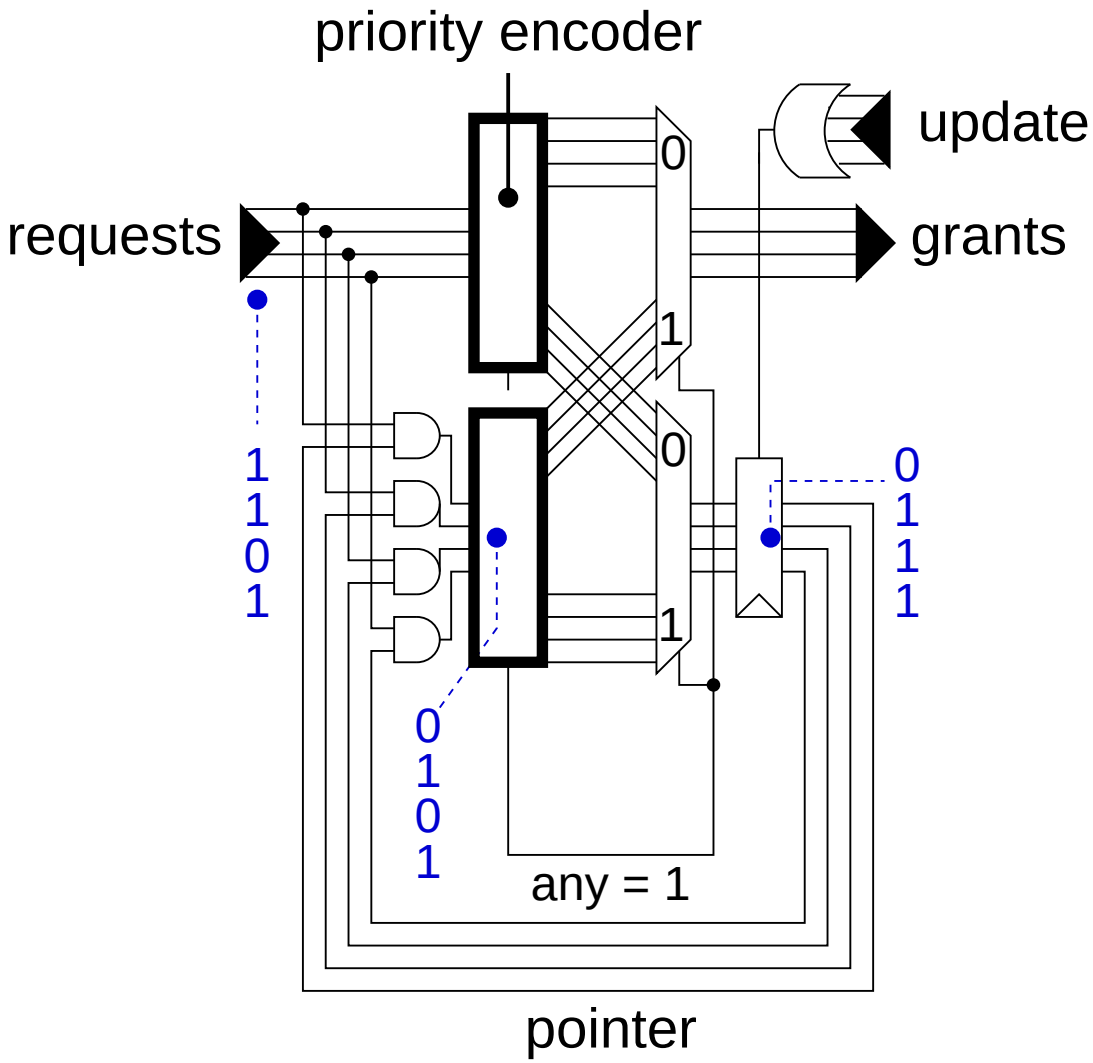
example N = 4

The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



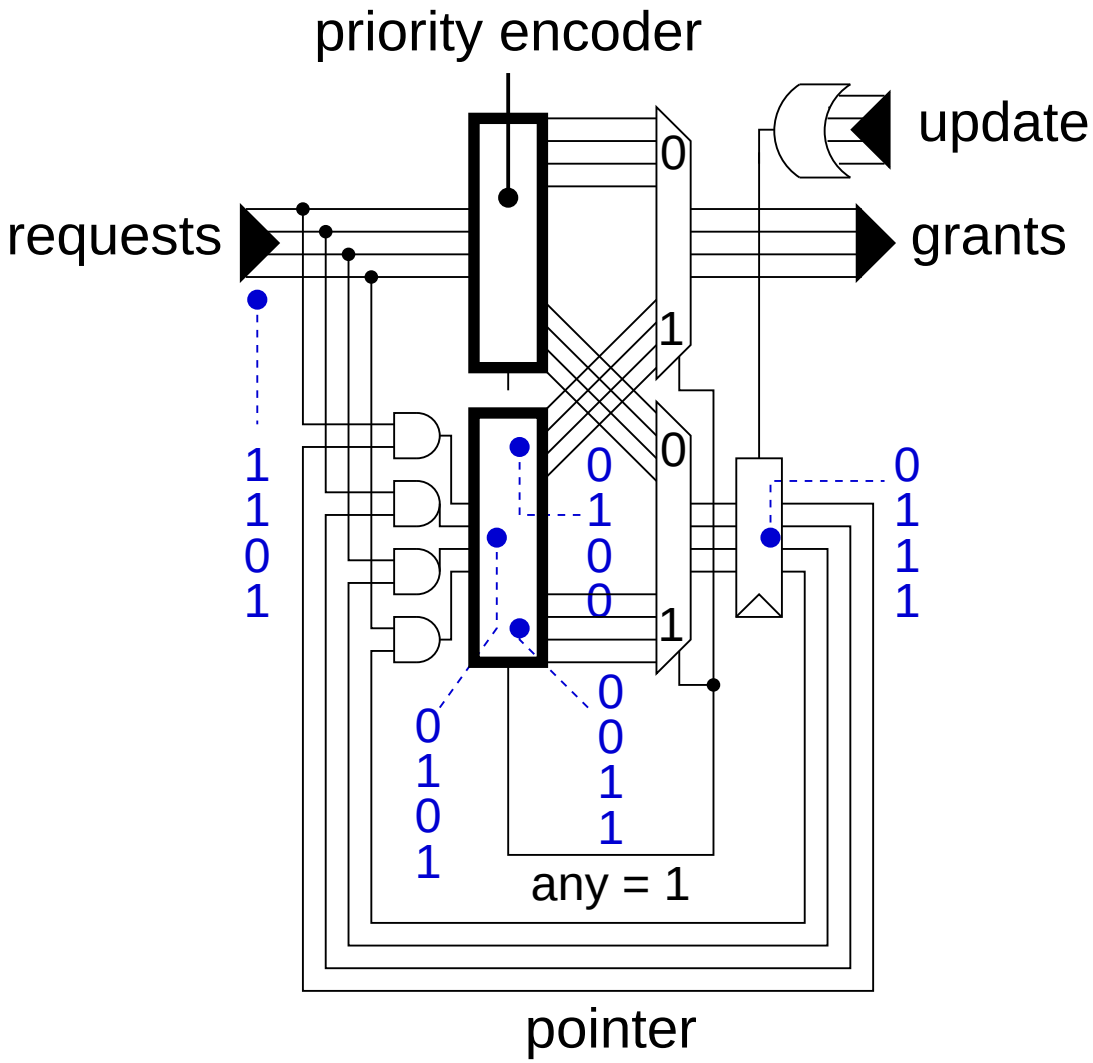
example N = 4

The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



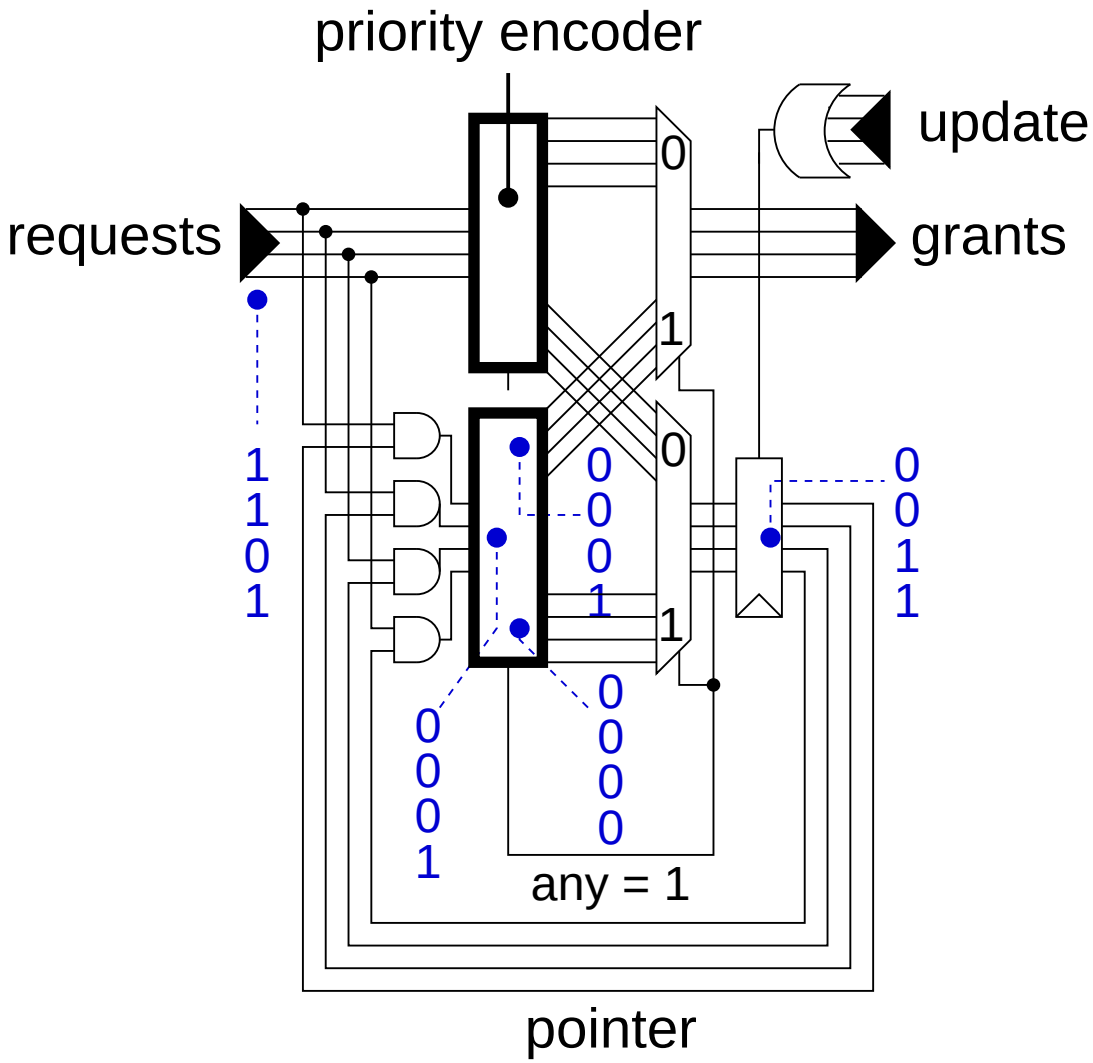
example N = 4

The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



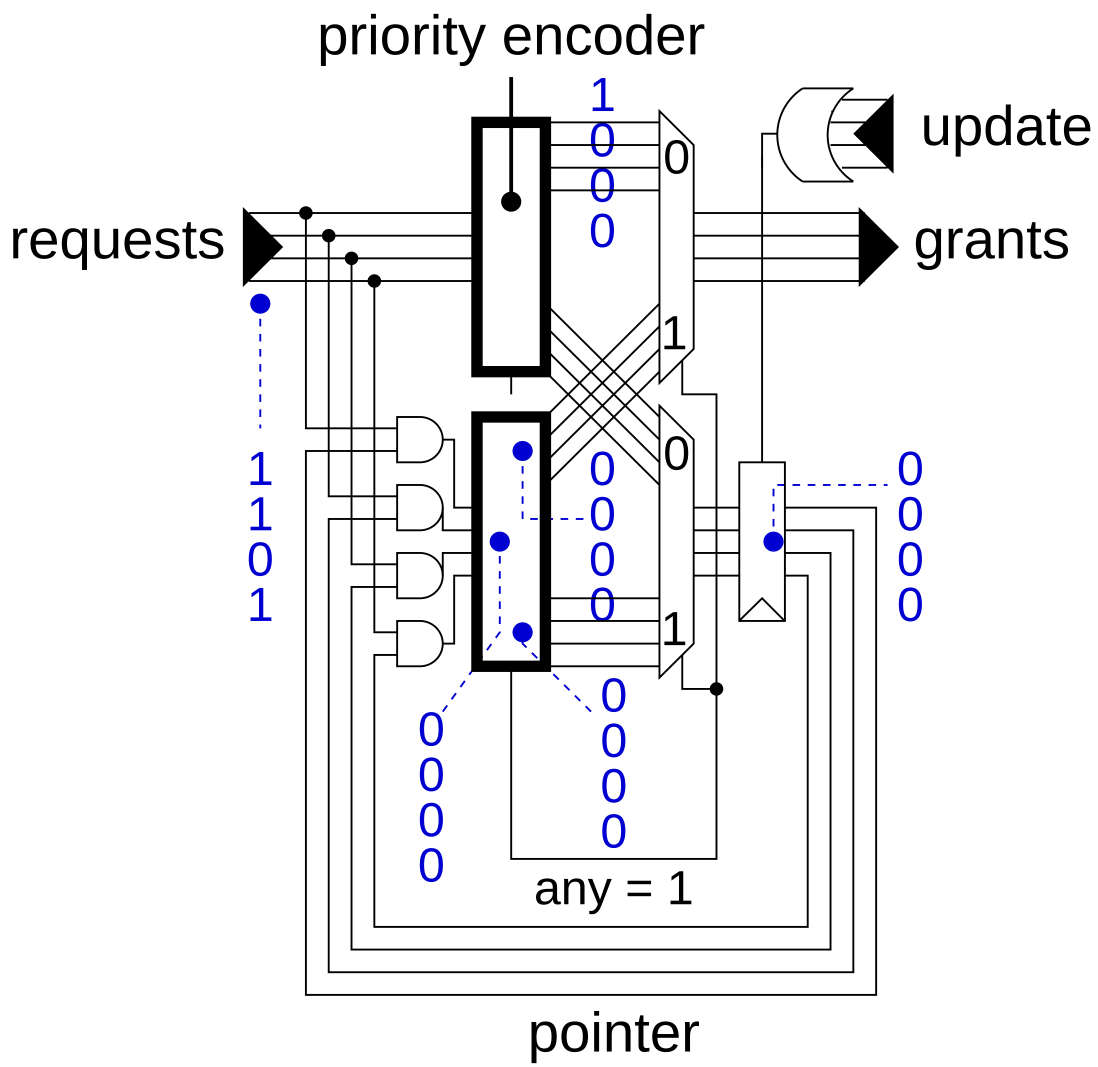
example N = 4

The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



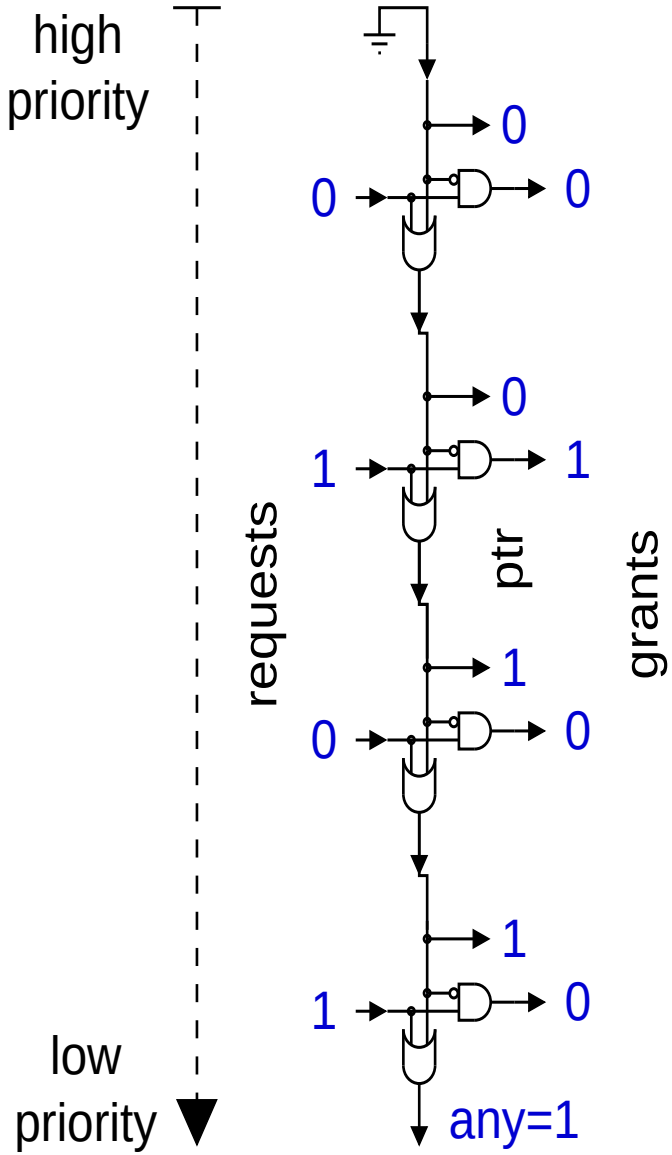
example N = 4

The iSLIP Circuit: Arbiter (Gupta/Mckeown, 1999)



example N = 4

iSLIP Area Cost Analysis: Priority Encoder

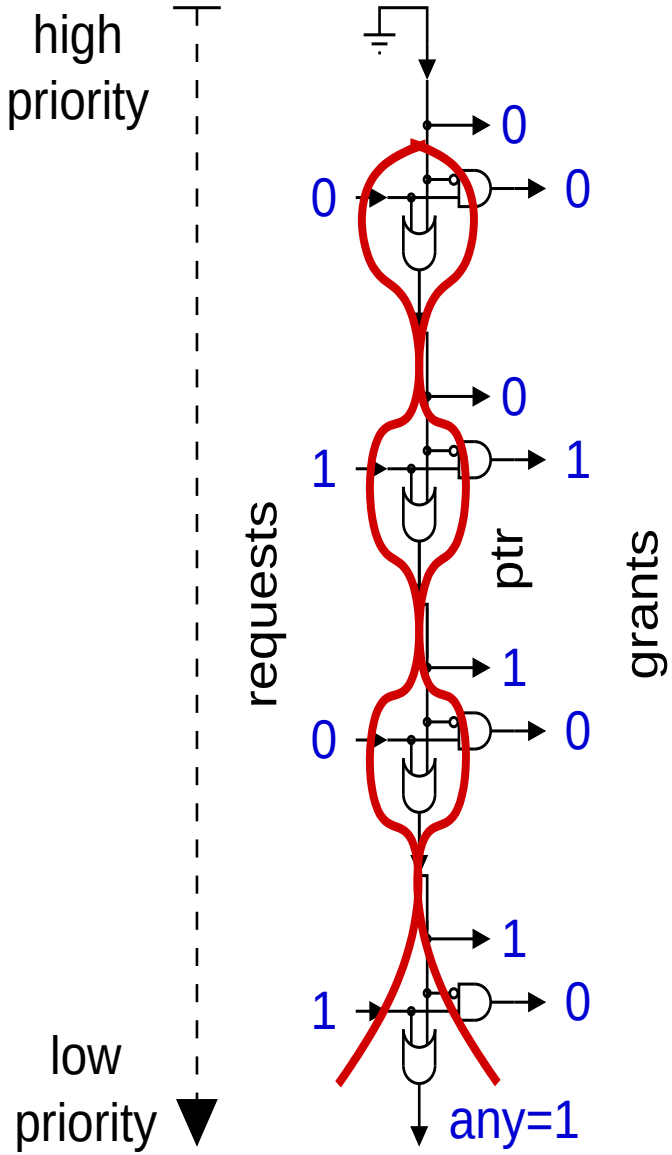


example N = 4

-- O(N) gate area

-- O(N) wiring area

iSLIP Area Cost Analysis: Priority Encoder



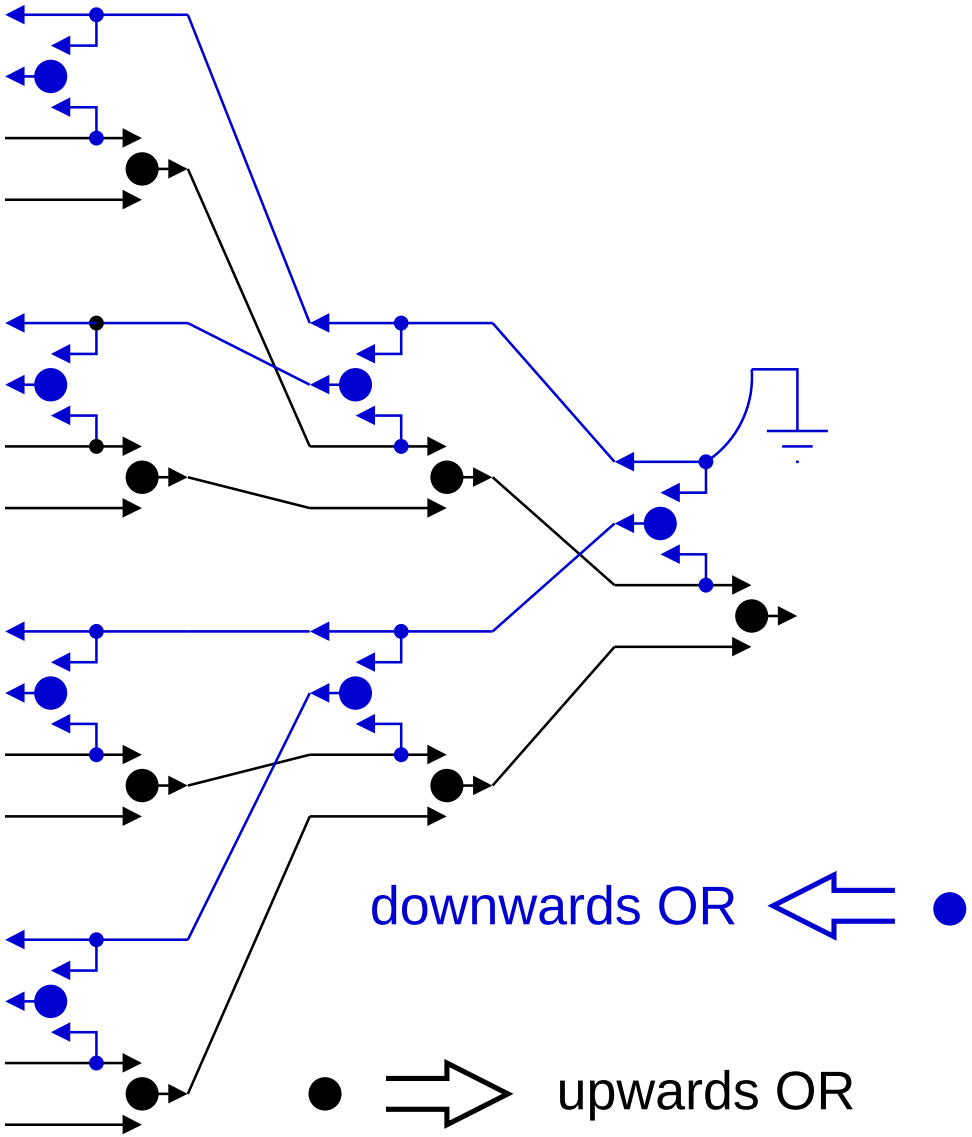
example N = 4

-- O(N) gate area

-- O(N) wiring area

...too slow

iSLIP Area Cost Analysis: Priority Encoder



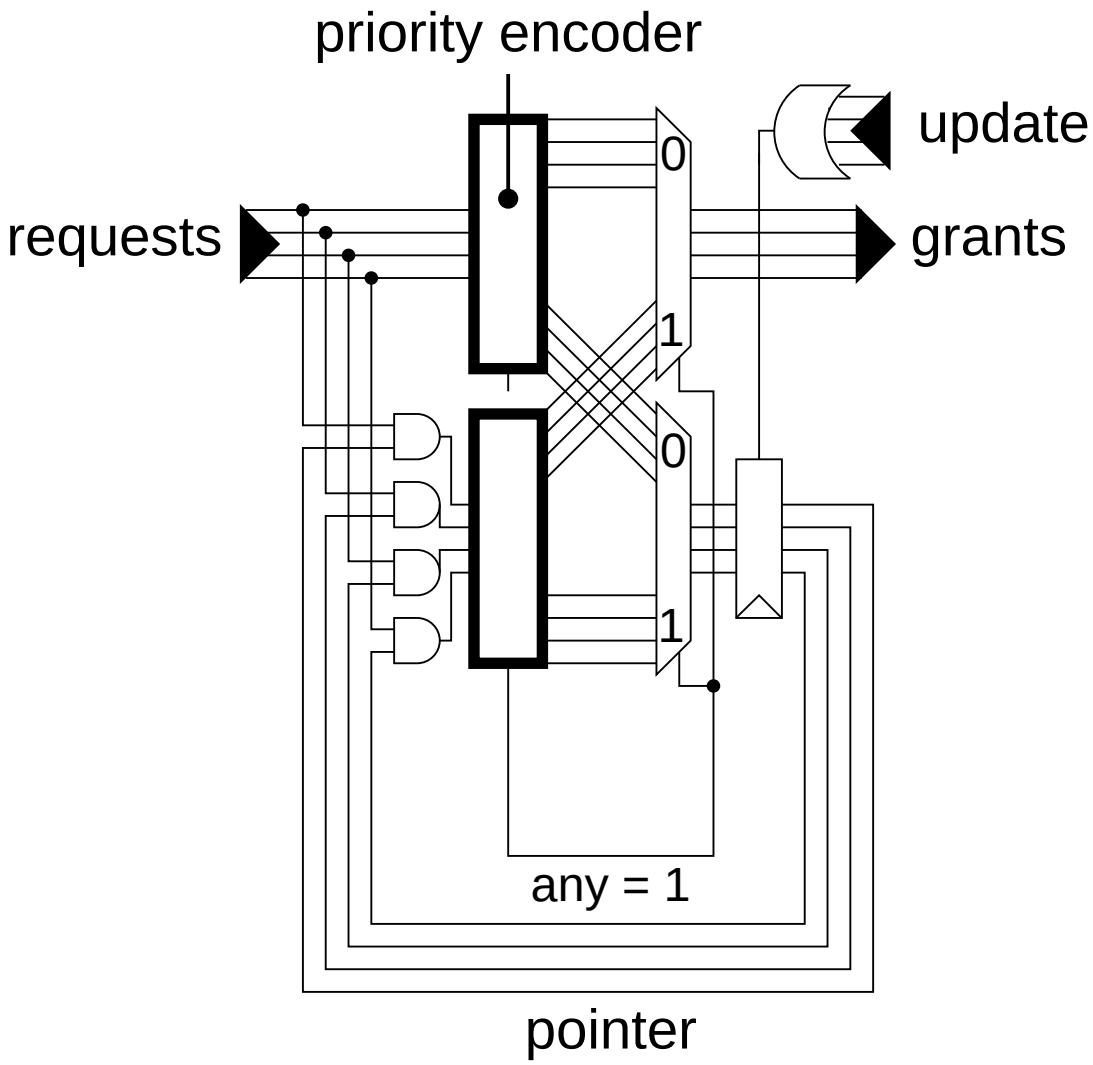
-- O(N) gate area

-- O(NlogN) wiring area

Parallel-Prefix Network (Brent & Kung, 1982)

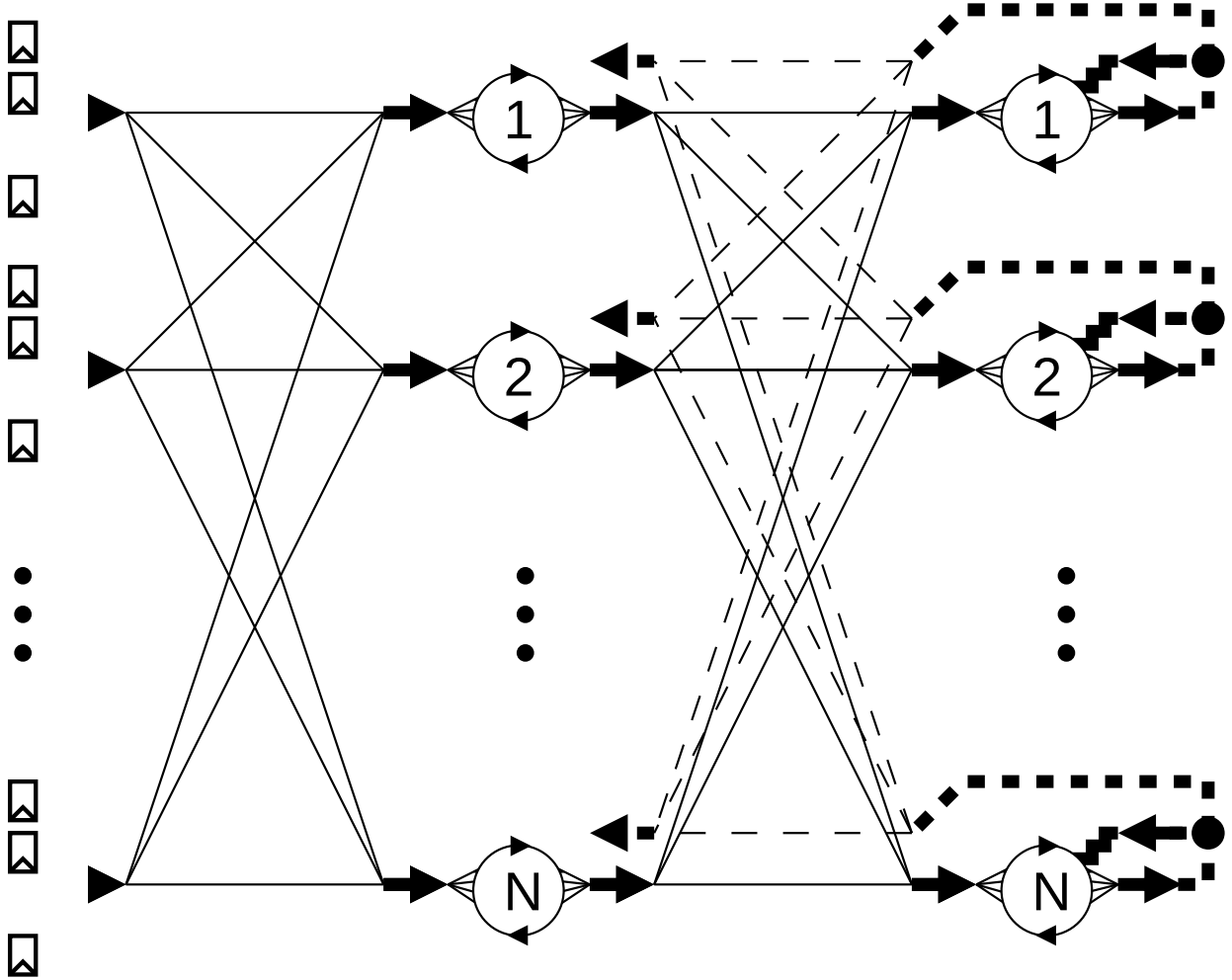
iSLIP Area Cost Analysis: Arbiter

O(N) gates, O(NlogN) wiring



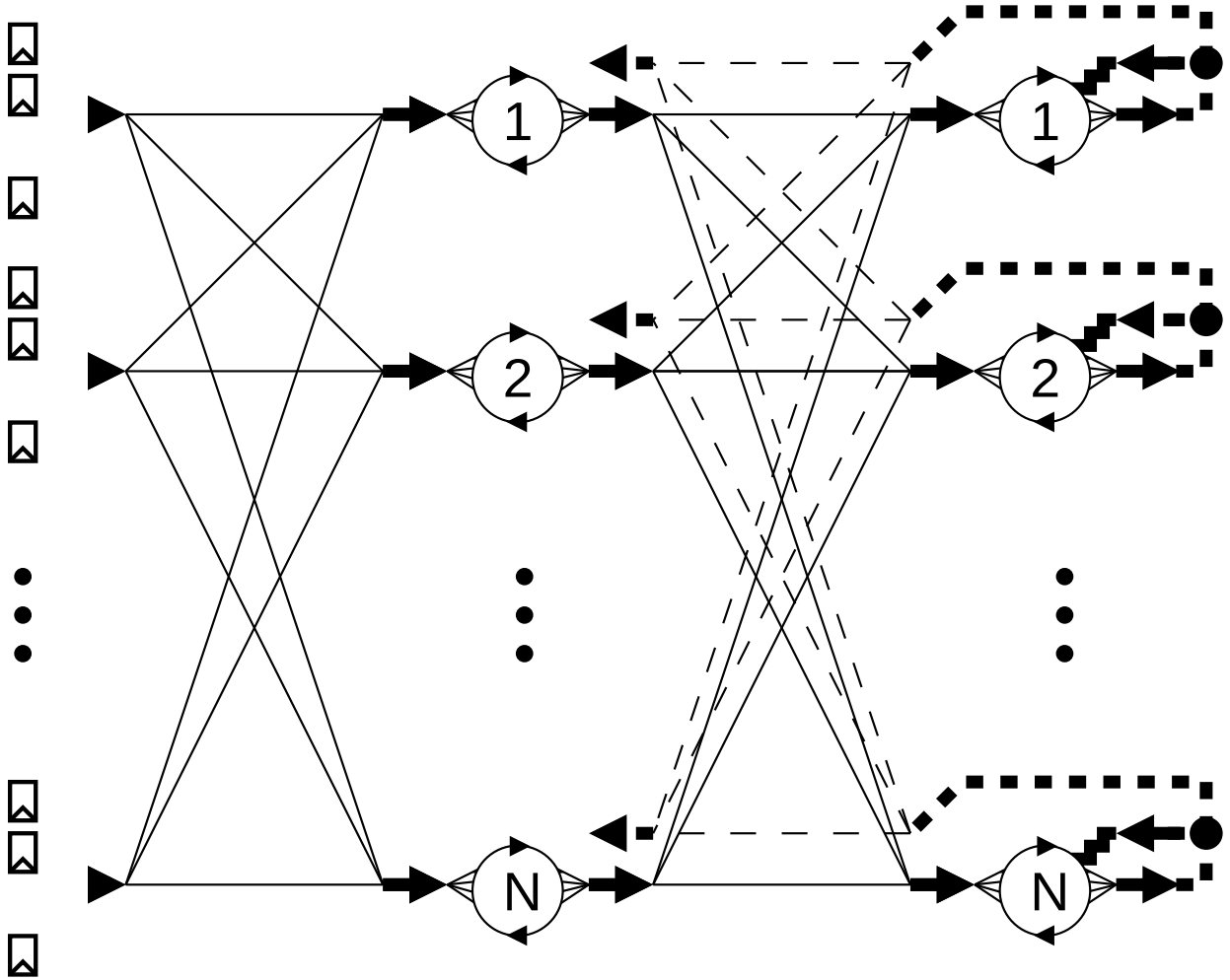
iSLIP Area Cost Analysis: All Arbiters

$O(N^2)$ gates, $O(N^2 \log N)$ wiring



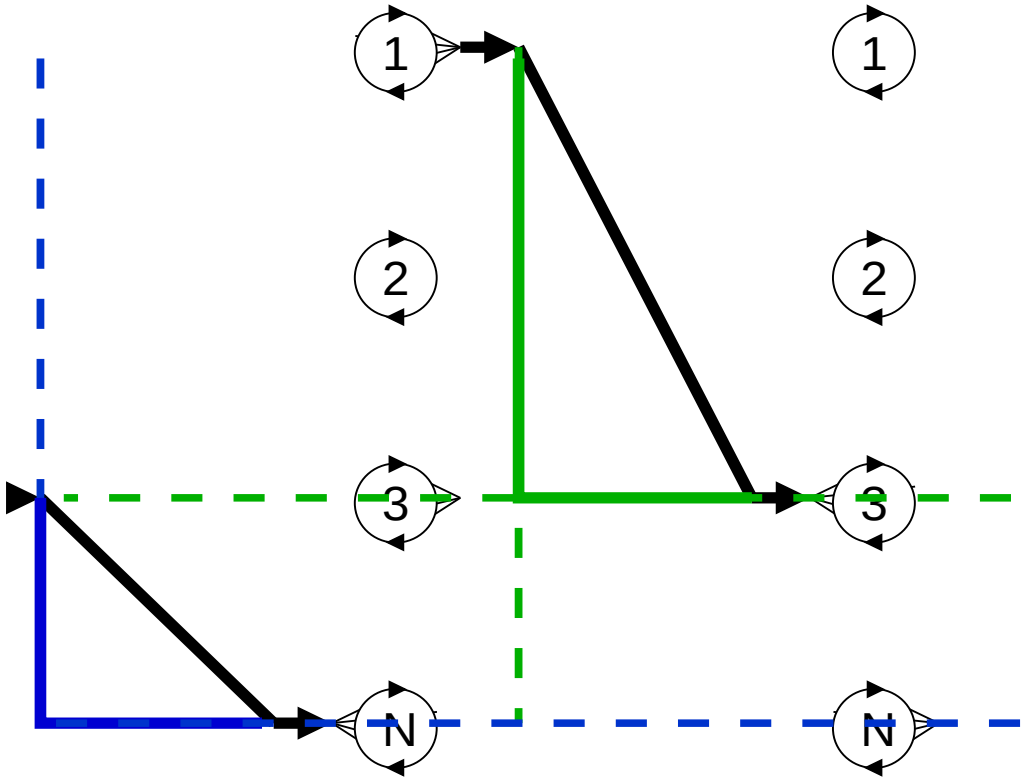
iSLIP Area Cost Analysis: All Arbiters

$O(N^2)$ gates, $O(N^2 \log N)$ wiring



...cost of p2p links?

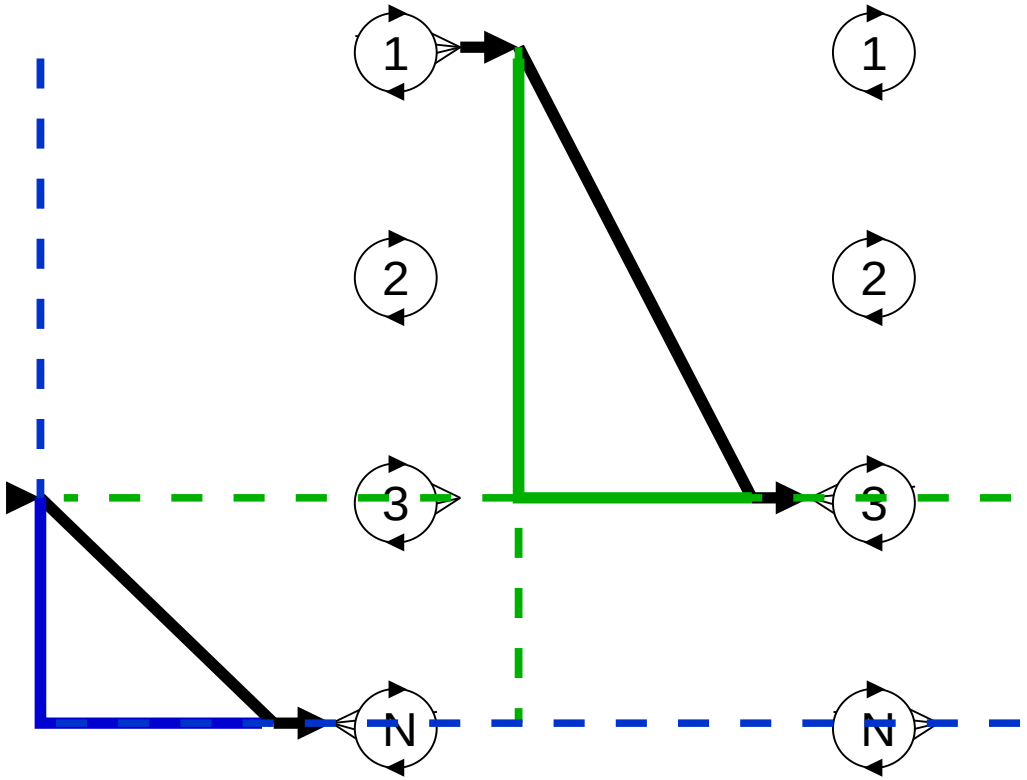
iSLIP Area Cost Analysis: P2P Links



p2p link area = ($3N^2$ R / L / 2) x ($3N^2$ R / L / 2)

$3N^2$ = # p2p links,
R = avg routing pitch,
L = # metal layers / dimension,
metal tracks half utilized on avg

iSLIP Area Cost Analysis: P2P Links



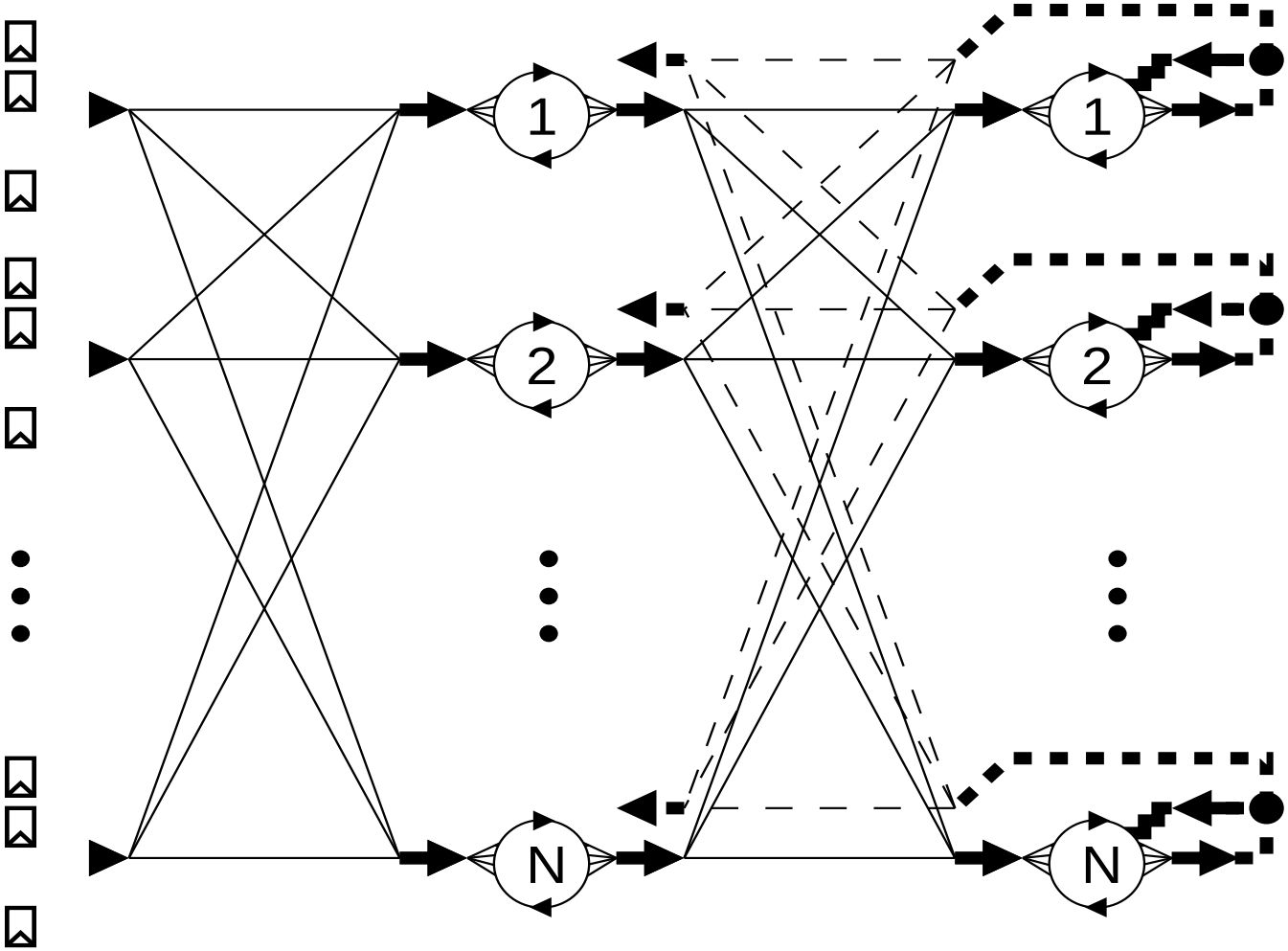
p2p link area = ($3N^2$ R / L / 2) x ($3N^2$ R / L / 2) → **O(N^4)**

$3N^2$ = # p2p links,
R = avg routing pitch,
L = # metal layers / dimension,
metal tracks half utilized on avg

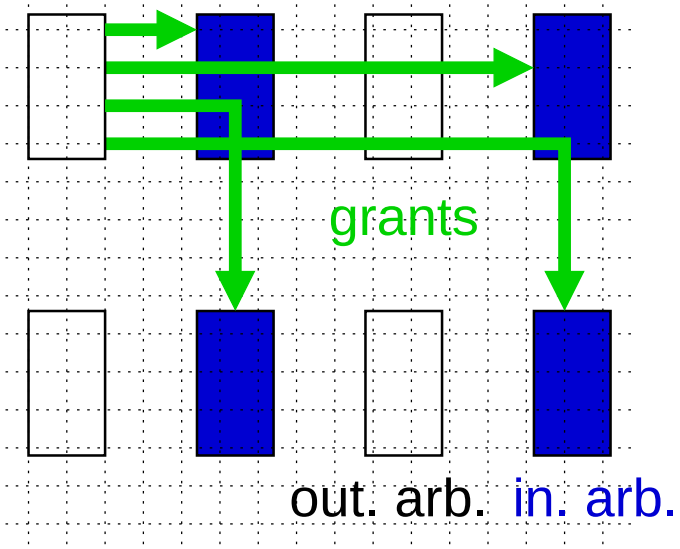
iSLIP Area Cost Analysis: Summary

$O(N^2)$ gates, $O(N^2 \log N)$ wiring for arbiters

$O(N^4)$ for p2p links



iSLIP Area Cost Analysis: Experimentation



example N = 4

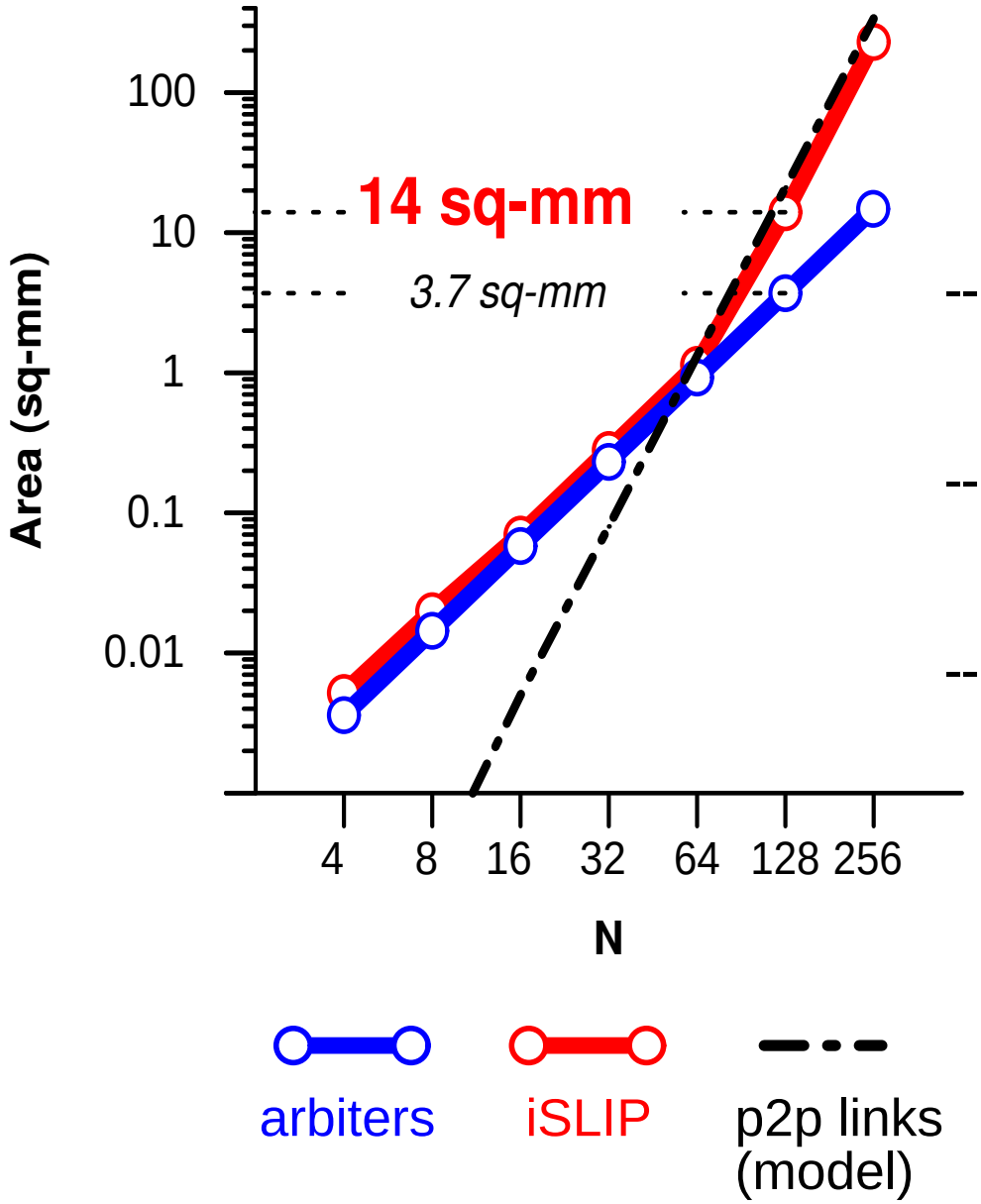
-- Hierarchical PnR for N in [4:256]

-- Arbiter block
- optimized for speed
- M1-M3
- area utilization ~ 100%

-- Top level (arb. blocks + p2p links)
- 2D array of arbiter blocks
- M4-M9 over arbiter blocks
- M1-M9 bet arbiter blocks

-- Define min arbiter spacing
for routable layouts (no DRVs)

iSLIP Area Cost Analysis: Experimentation



-- iSLIP gate limited for N < 64

-- iSLIP wire limited for N > 64

-- N=128: 70% of area for p2p links

Main Idea:

The traditional architecture
localizes the internal wires of the arbiters
and globalizes their p2p links.

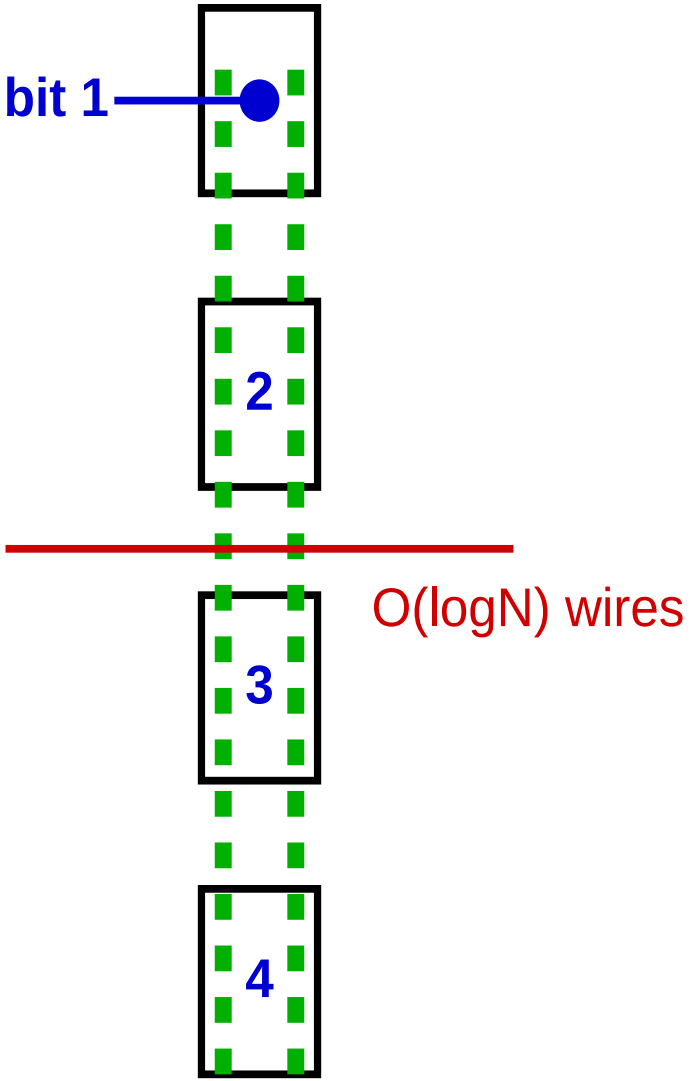
Since internal arbiter wiring is much cheaper than p2p links,
this is a handicap.

The locality of wires has to be inverted!

How?

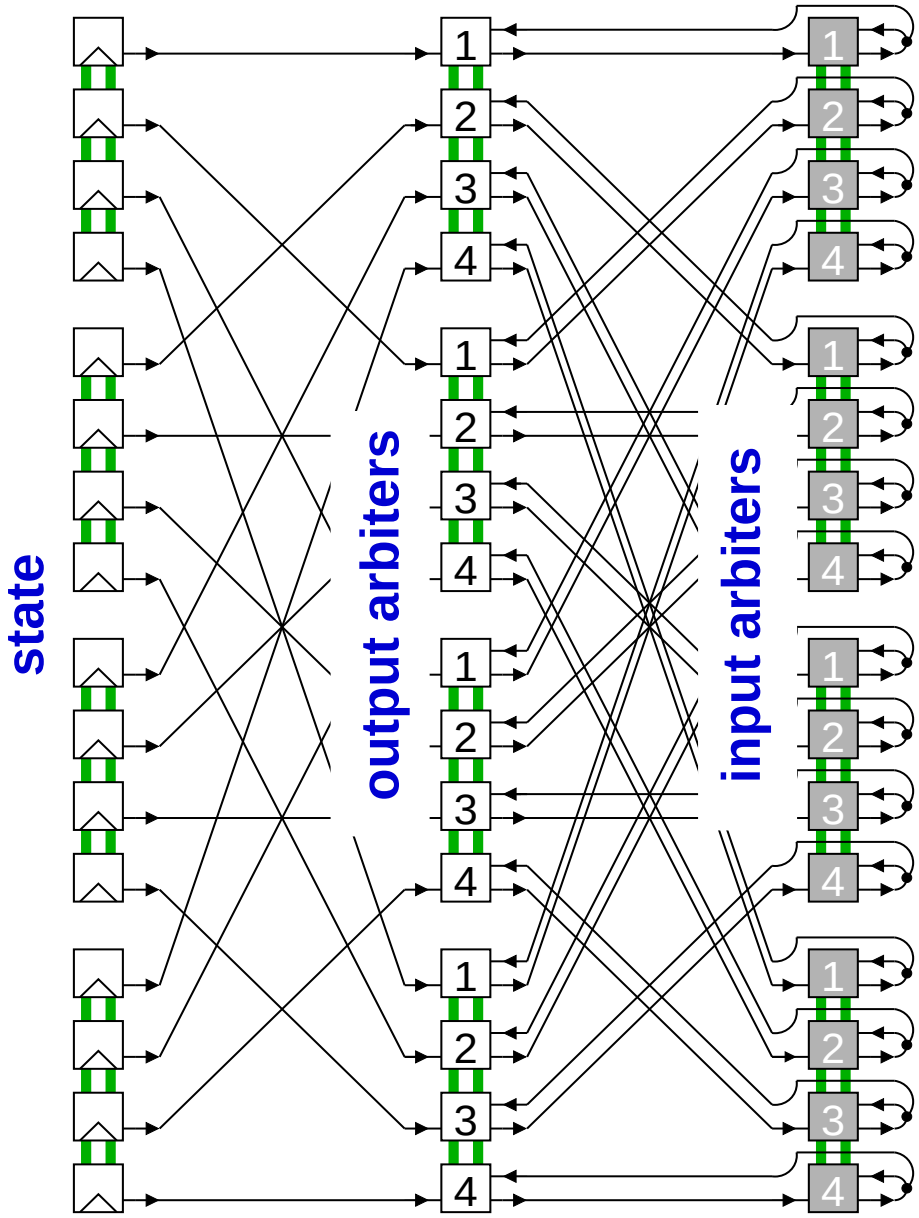
Cross Micro-Architecture

Cross Architecture: Arbiter Slicing



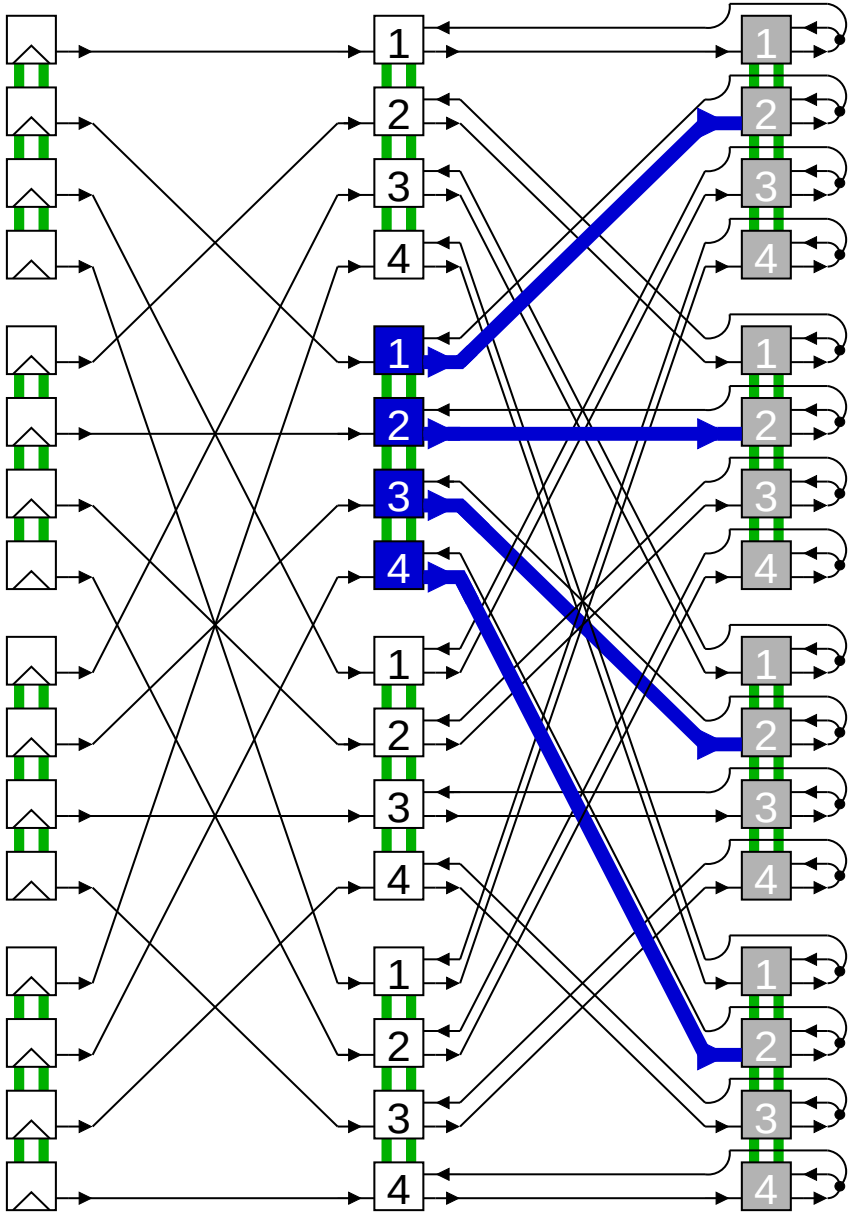
example N = 4

Cross Architecture: Scheduler Slicing



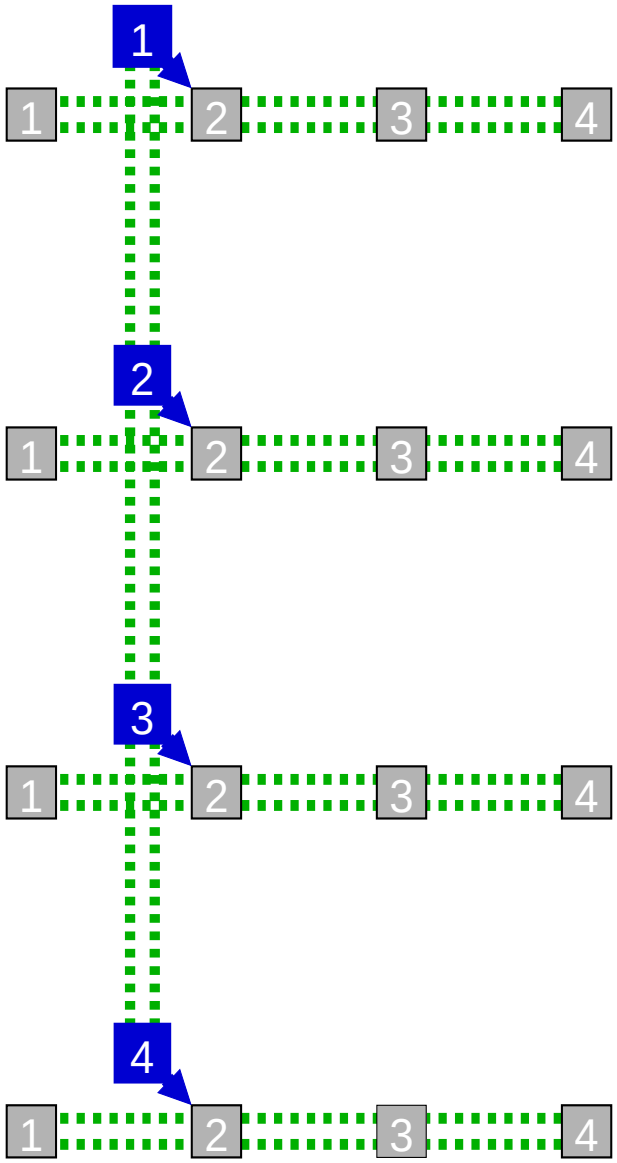
example N = 4

Cross Architecture: Scheduler Slicing



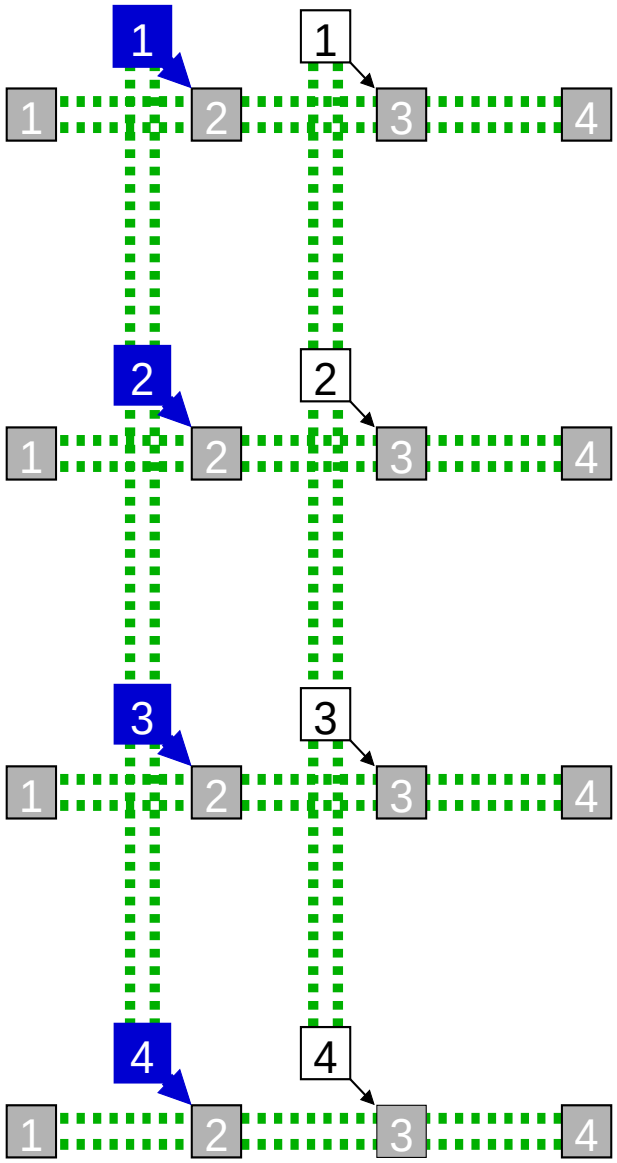
grant links

Cross Architecture: Arbiter Interleaving



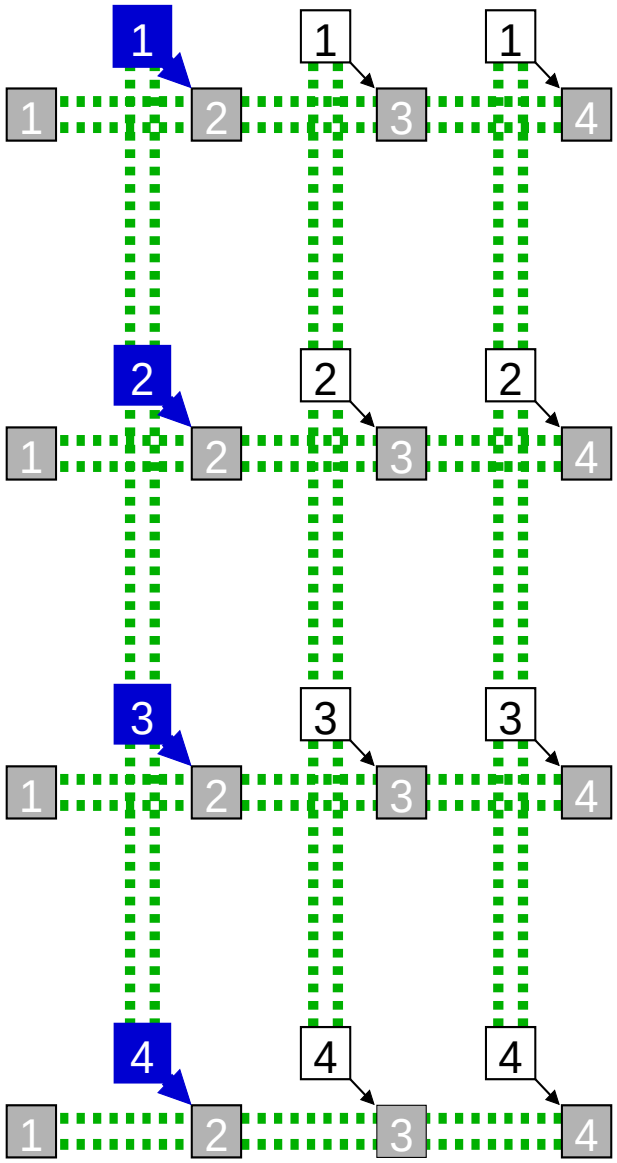
localizing the grant links

Cross Architecture: Arbiter Interleaving



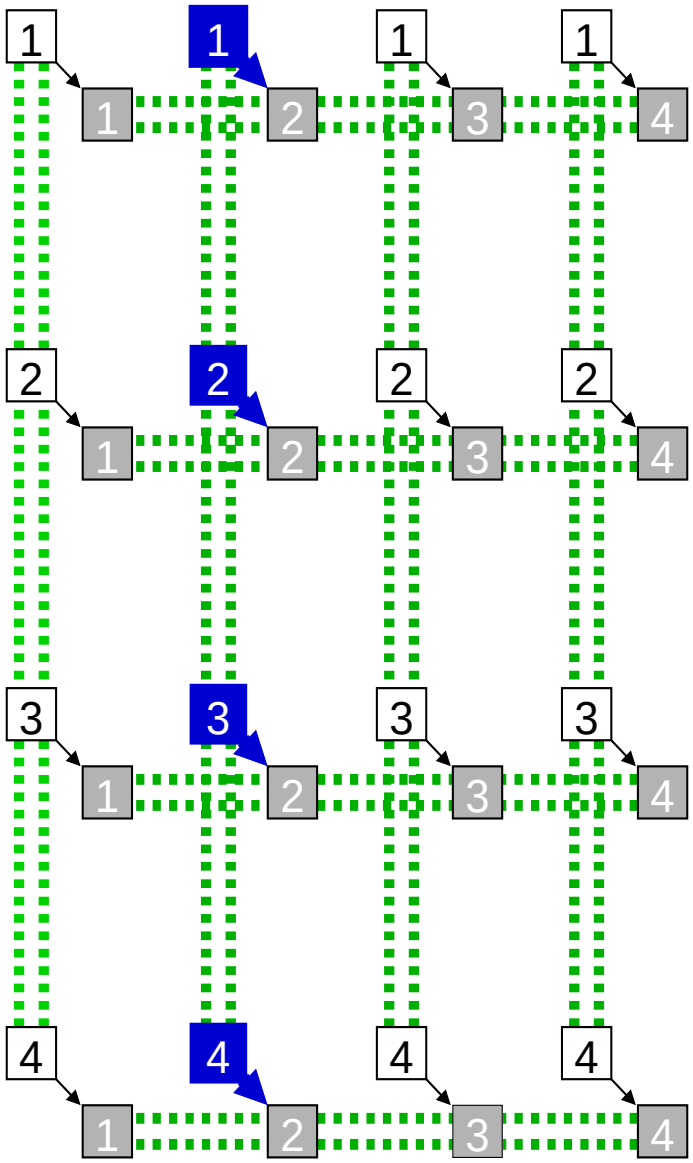
localizing the grant links

Cross Architecture: Arbiter Interleaving



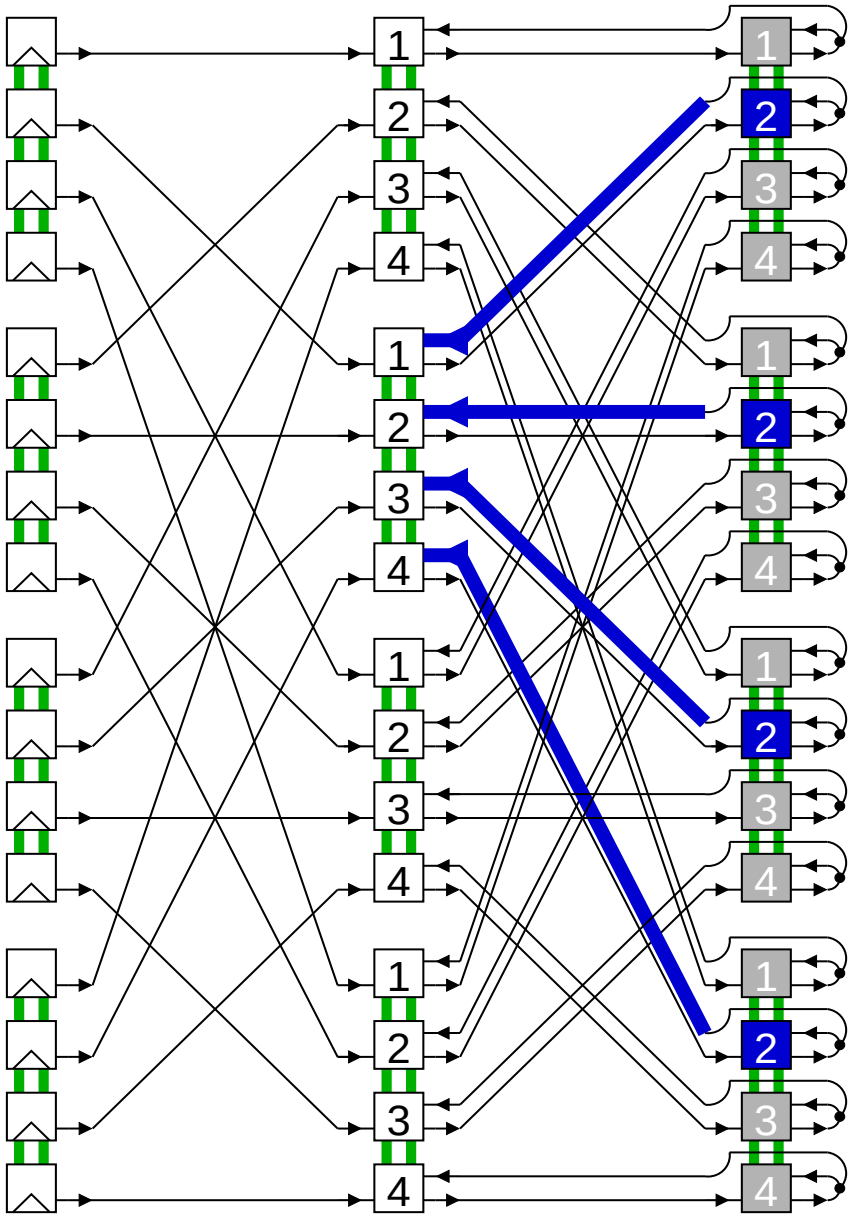
localizing the grant links

Cross Architecture: Arbiter Interleaving



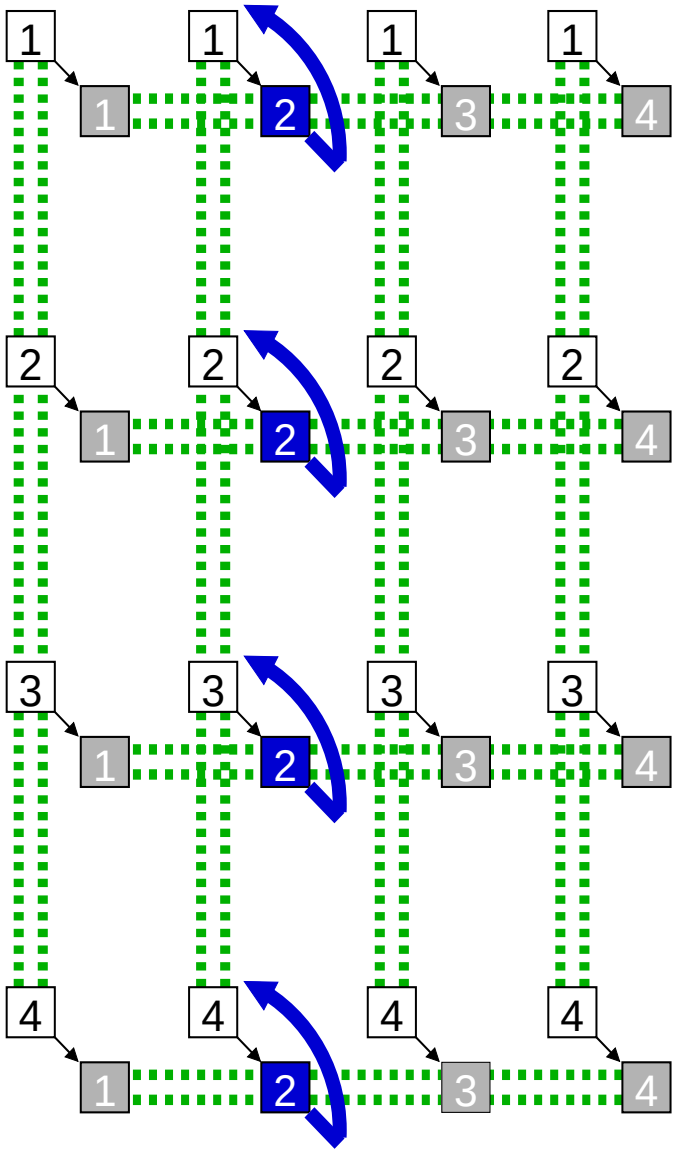
localizing the grant links

Cross Architecture: Scheduler Slicing



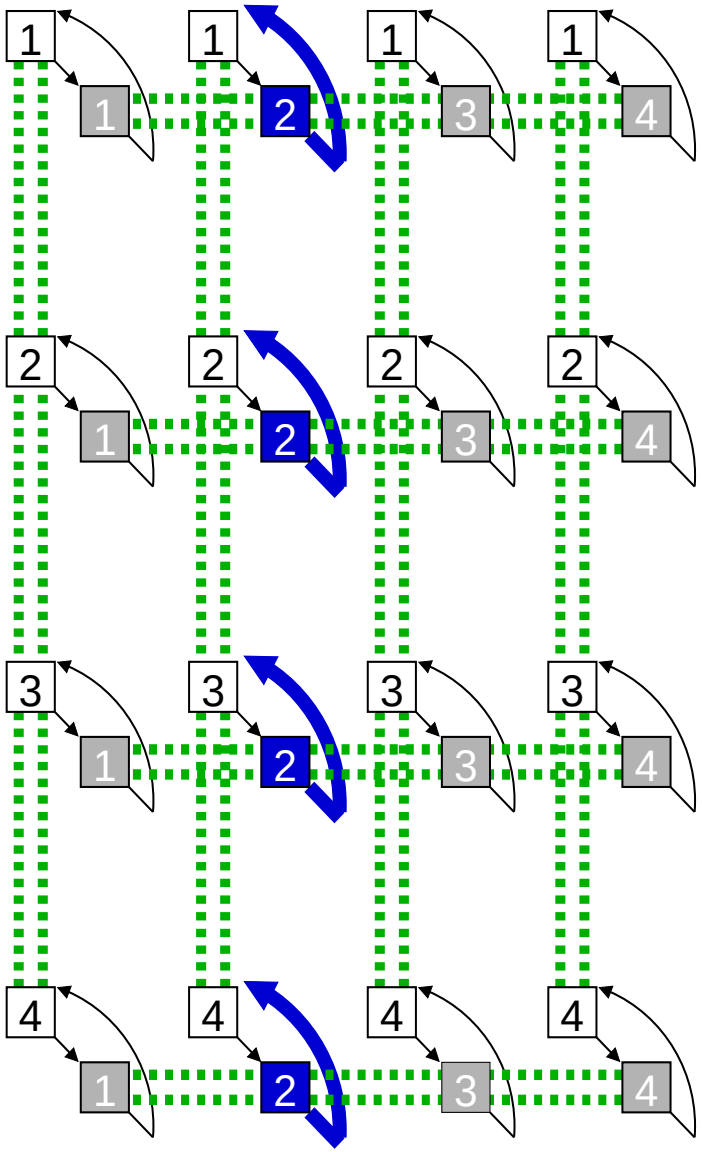
accept links

Cross Architecture: Arbiter Interleaving



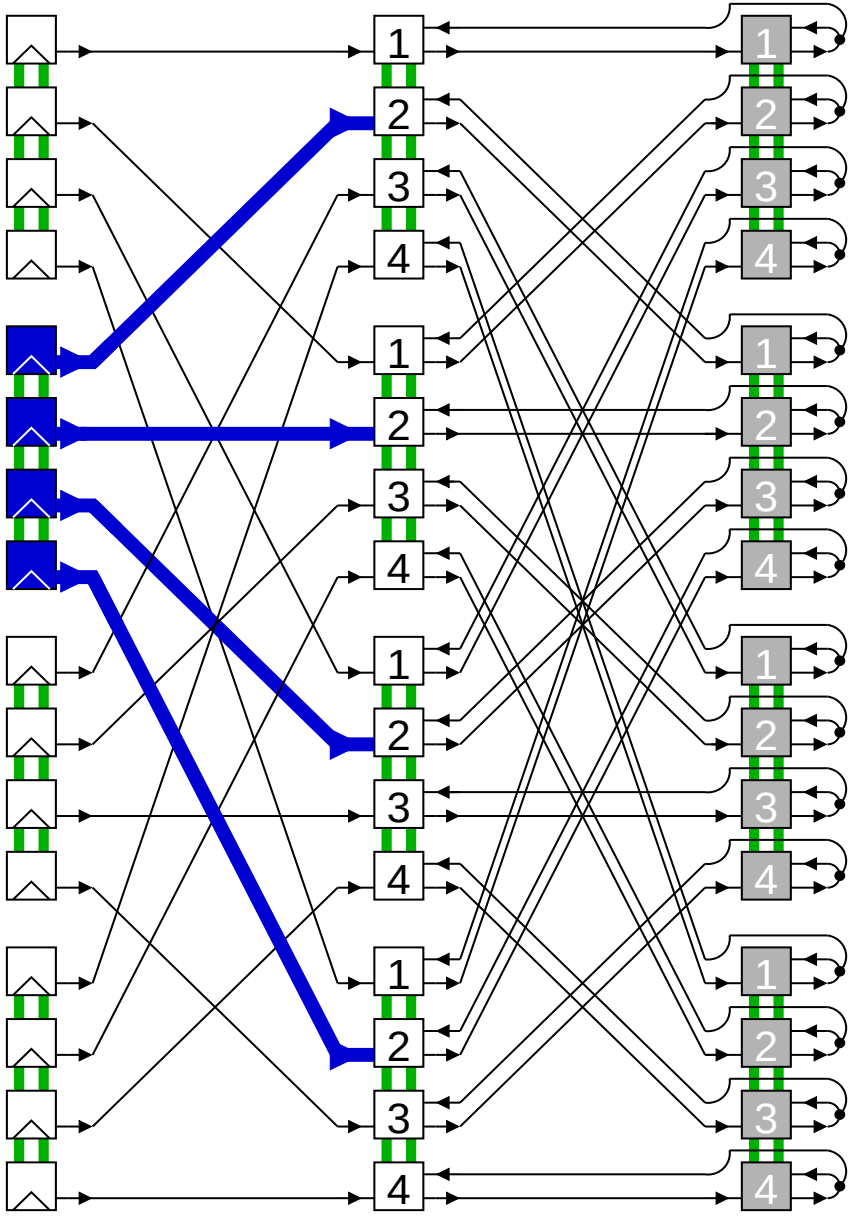
accept links localized by symmetry

Cross Archtiecture: Arbiter Interleaving



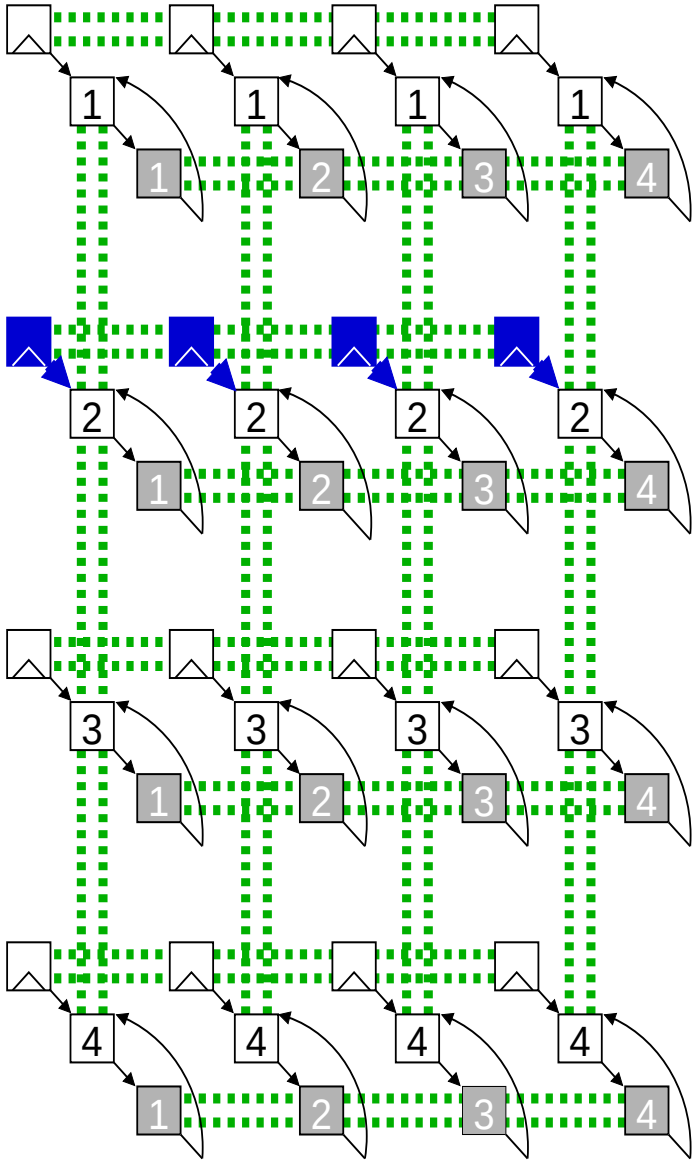
accept links localized by symmetry

Cross Architecture: Scheduler Slicing



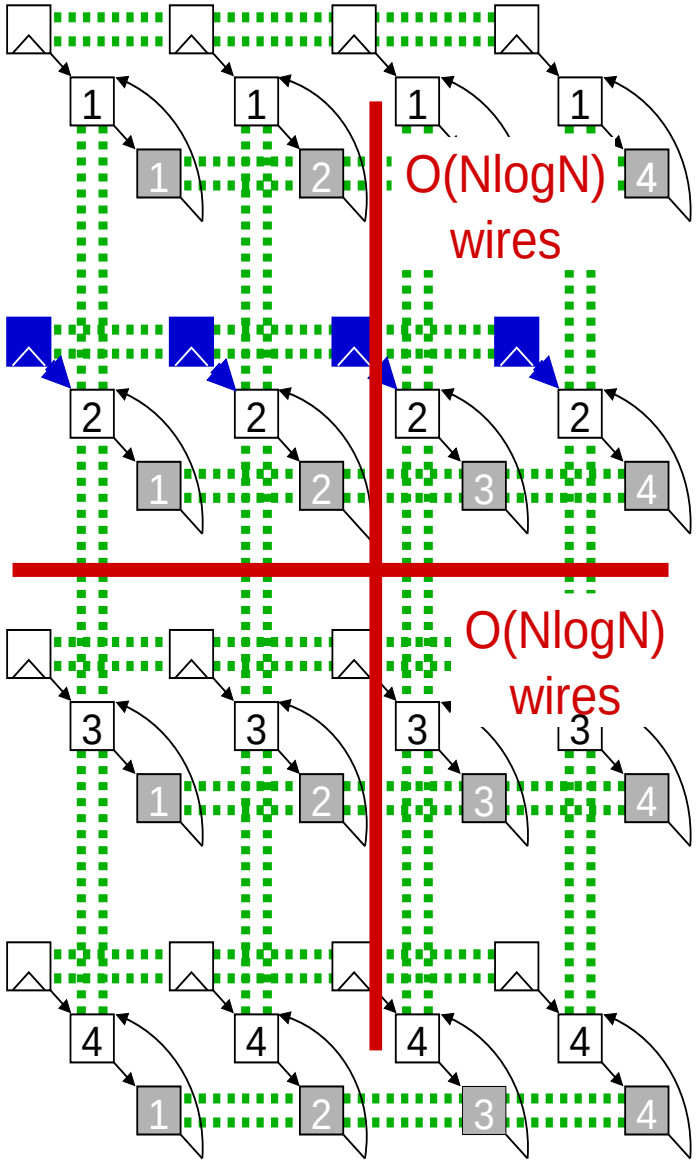
request links

Cross Architecture: Crosspoint State



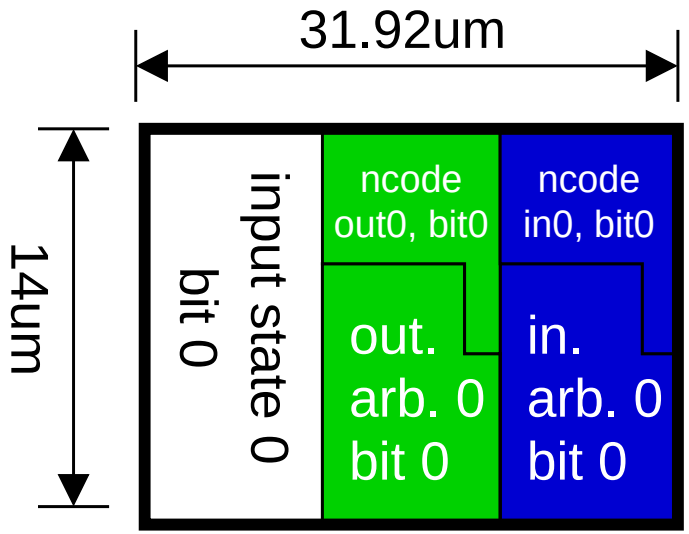
localizing the request links

Cross Architecture: Wiring Complexity



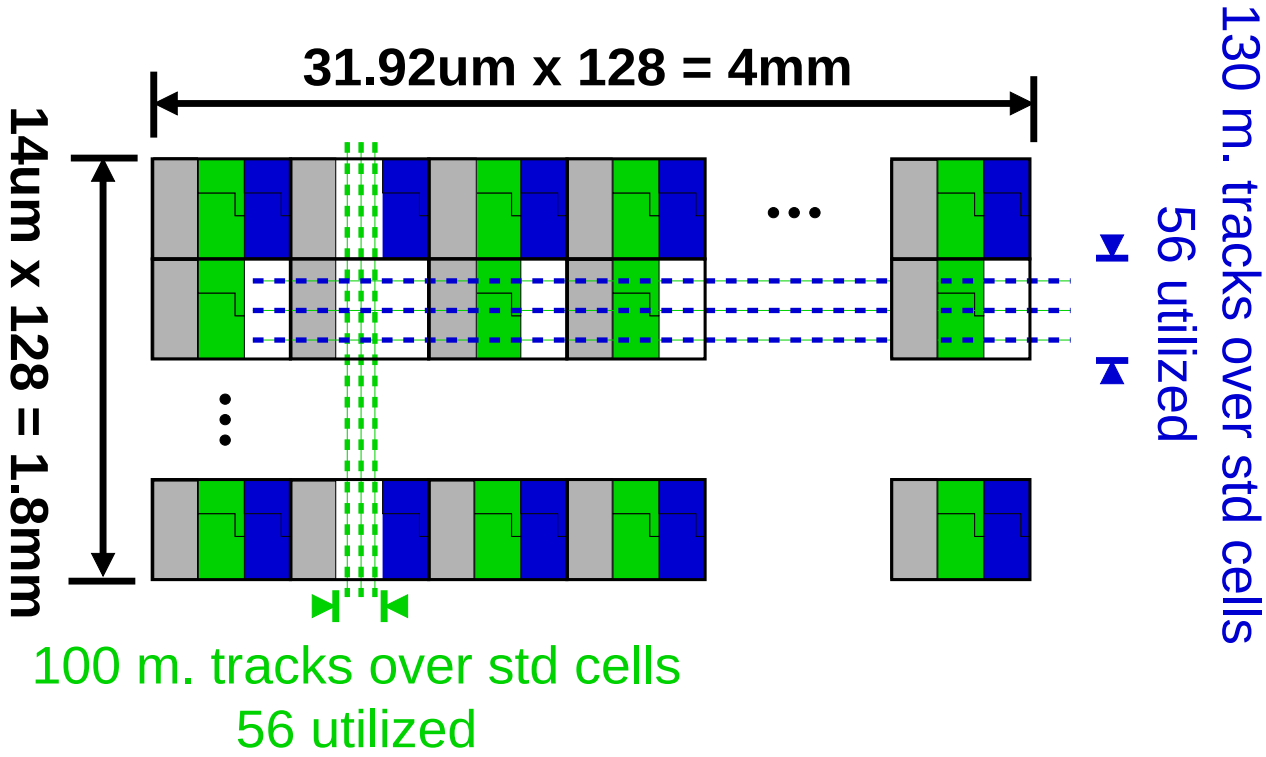
$O(N^2 \log^2 N)$ wiring complexity

Cross Architecture: radix-128 90nm CMOS Layout



Cross Architecture: 90nm CMOS Layout

7.2mm^2, 100% utilization, 2x smaller than the traditional



8 trees per arbiter: 5 lookahead + 2 fanout

Conclusion

-- High-radix crossbar NoCs are feasible even in 90nm CMOS

- NOCS 2010: datapath

- this paper: control optimization

-- Future work:

- energy efficiency

- fault tolerance