Computing Dedekind Numbers

Patrick De Causmaecker
Lennart Van Hirtum
Jens Goemaere

Richard Dedekind, 1897

ÜBER

ZERLEGUNGEN VON ZAHLEN

DURCH IHRE

GRÖSSTEN GEMEINSAMEN THEILER.

Von

R. DEDEKIND.

Die Anzahl der in dieser Gruppe $\mathfrak{P}$ enthaltenen Elemente scheint mit der Anzahl n der gegebenen Elemente (1) sehr rasch zu wachsen; sie ist $= 18$ im Falle $n = 3$, und (wenn ich nicht irre) $= 166$ im Falle $n = 4$; einen allgemeinen Ausdruck für diese Anzahl zu finden, habe ich noch nicht versucht. Dagegen leuchtet ein, dass die Elemente von $\mathfrak{P}$, d. h. die vollständigen Producte $\mathfrak{p}$ sich nach der Anzahl der

Randolph Church, 1940

NUMERICAL ANALYSIS OF CERTAIN FREE DISTRIBUTIVE STRUCTURES

By Randolph Church

Consider the set Σ_n of all formal cross-cuts and unions[1] of n symbols $A_1, A_2, \cdots, A_n$. Disjoint classes which exhaust Σ_n can be formed with respect to an equivalence introduced according to the axioms of a distributive

Dedekind[3] gave the order of Δ_n for $n \leqq 4$. The purpose of this paper is to present an analysis of $N(\Delta_n)$, $n \leqq 5$. The analysis depends on the notion of conjugate elements. Let X_1 and X_2 be two elements of Δ_n, written as the cross-

$n = 2$

r \ h	1	2	N.
2	1		1
1		1	2
0	1		1
	2	1	4

$n = 3$

r \ h	1	3	N.
6	1		1
5		1	3
4		1	3
3	1	1	4
2		1	3
1		1	3
0	1		1
	3	5	18

$n = 4$

r \ h	1	3	4	6	12	N.
14	1					1
13			1			4
12				1		6
11			1	1		10
10	1				1	13
9				1	1	18
8		1	1		1	19
7			3		1	24
6		1	1		1	19
5				1	1	18
4	1				1	13
3			1	1		10
2				1		6
1			1			4
0	1					1
	4	2	9	6	7	166

$n = 5$

r \ h	1	5	10	12	15	20	30	60	120	N.
30	1									1
29		1								5
28			1							10
27			2							20
26		1					1			35
25	1						2			61
24			2		1			1		95
23			1		1	2	1	1		155
22			1		1	2	1	2		215
21		1	2		1		1	4		310
20		1	1	1			2	5		387
19		1	1		1	1	2	6		470
18			1			2	4	4	1	530
17					2	2	3	5	1	580
16		1	1			1	3	6	1	605
15	1	2	2			1	3	6	1	621
14		1	1			1	3	6	1	605
13					2	2	3	5	1	580
12			1			2	4	4	1	530
11		1	1		1	1	2	6		470
10		1	1	1			2	5		387
9		1	2		1		1	4		310
8			1		1	2	1	2		215
7			1		1	2	1	1		155
6			2		1			1		95
5	1						2			61
4		1					1			35
3			2							20
2			1							10
1		1								5
0	1									1
	5	14	28	2	14	21	43	74	7	7579

Morgan Ward, 1946

Morgan Ward, Note on the order of free distributive lattices Bulletin of the American Mathematical Society, Vol. 52, No. 5 (1946), p. 423.

Eric Weisstein's World of Mathematics, Antichain

135. Morgan Ward: *Note on the order of the free distributive lattice.*

If r_n denotes the order of the free distributive lattice on n elements, and if we set $\log_2 r_n$ equal to $2^n\phi(n)$, then for large n, $1/n^{1/2} < \phi(n) < 1/4$ so that $\log_2 \log_2 r_n \sim n$. Computational evidence and combinatorial arguments suggest that $n^{1/2}\phi(n) \to \infty$, but the exact order of $\phi(n)$ is unknown. Incidentally the value of r_6 was computed. It is 7,828352. The method of computation devised easily verified Randolph Church's value 7579 for r_5 (Duke Math. J. vol. 6 (1940) pp. 732–734) but is not powerful enough to evaluate r_7 without prohibitive labor. (Received March 22, 1946.)

Randolph Church, 1965

Six Hundred Twenty-Sixth Meeting
Massachusetts Institute of Technology
Cambridge, Massachusetts
October 30, 1965

Notices

OF THE

AMERICAN MATHEMATICAL SOCIETY

65T-447. RANDOLPH CHURCH, U. S. Naval Postgraduate School, Monterey, California 93940. Enumeration by rank of the elements of the free distributive lattice with seven generators.

A method has been devised by which it has been feasible to calculate, using a high speed digital computer [CDC 1604] as time on it was available, the number of isotone functions from B [the Boolean lattice having two elements, 0 and 1] to FDL(6). The method of enumeration yields a table in which the entry in the ith row and jth column, [illegible], is the number of these isotone functions for which the

Doug Wiedemann, 1991

Order **8**: 5–6, 1991.

A Computation of the Eighth Dedekind Number

DOUG WIEDEMANN*
Thinking Machines Corporation, 245 First Street, Cambridge, Massachusetts 02142, U.S.A.

$$d_8 = \sum_{R \in R_6, T \in D_6} \gamma(R)\eta(R \cap T)\eta(R^* \cap T^*).$$

of R_6, since the value of η only depends on the symmetric class. The entire computation used about 200 hours of time on one Cray-2 processor. A very short test run reproduced the previously known value of d_7.

The number obtained for d_8 was 56, 130, 437, 228, 687, 557, 907, 788. This is slightly larger than the estimate of about 5.43×10^{22} obtained by substituting $n = 8$ into Korshunov's formula [4]. Another, very minor check on our result is that d_n is even whenever n is even [2, p. 63].

Lennart Van Hirtum, 2023

$$286386577668298411128469151667598498812366.$$

$$d(n+3) = \sum_{y \in D(n)} \sum_{a,b,c \in [\bot, y]} \bot(a \wedge b \wedge c) \cdot \top(a \vee b) \cdot \top(a \vee c) \cdot \top(b \vee c)$$

Christian Jäkel, 2023

$$d(n+4) = \sum_{[I] \in \mathrm{Int}(\mathbb{D}_n)/_{\equiv}} \#[I] \cdot \sum_{a,b \in I} \mathrm{Tr}(\gamma_{ab}^2).$$

$$\gamma(e,d) = \sum_{f \in I} \alpha(e,f)\beta(f,d).$$

Parallelizing with MPI in Java to find the ninth Dedekind Number

Pieter-Jan Hoedt

At last it is worth mentioning De Causmaecker introduced a new formula

$$|\mathcal{A}_{n+3}| = \sum_{\alpha,\beta,\gamma \leq \rho \in \mathcal{A}_n} |[\bot, \alpha \wedge \beta \wedge \gamma]| \cdot |[\alpha \vee \beta, \rho]| \cdot |[\gamma \vee \beta, \rho]| \cdot |[\alpha \vee \gamma, \rho]|$$

Christian Jäkel, 2023

The only thing left to say is that we run the algorithm on Nvidia A100 GPUs. 5311 GPU hours and 4257682565 matrix multiplications later, we got the following value for the ninth Dedekind number:

$$286386577668298411128469151667598498812366.$$

Lennart Van Hirtum, 2023

$$D(n+2) = \sum_{\alpha,\beta \in D_n} |[\bot, \alpha]| 2^{C_{\alpha,\beta}} |[\beta, \top]|$$

$$\sum_{\alpha \in R_n} |[\bot, \alpha]| D_\alpha \sum_{\substack{\beta \in R_n \\ \exists \delta \simeq \beta : \alpha \leq \delta}} |[\beta, \top]| \frac{D_\beta}{n!} \sum_{\substack{\gamma \in Permut_\beta \\ \alpha \leq \gamma}} 2^{C_{\alpha,\gamma}}$$

$C_{\alpha,\beta}$: P-coefficient = number of connected components of a graph.

- Vertices sets in β
- Arcs: intersection of the sets is not below α

-> Master thesis -> FPGA -> Noctua 2…

A000372 as a simple table

n	a(n)
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788
9	286386577668298411128469151667598498812366

Sort: relevance | references | number | modified | created   Format: long | short | data

A000372 **Dedekind numbers** or **Dedekind**'s problem: **number** of monotone Boolean functions of n variables, **number** of antichains of subsets of an n-set, **number** of elements in a free distributive lattice on n generators, **number** of Sperner families. +40 82
(Formerly M0817 N0309)

2, 3, 6, 20, 168, 7581, 7828354, 2414682040998, 56130437228687557907788, 286386577668298411128469151667598498812366 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,1

COMMENTS
A monotone Boolean function is an increasing functions from P(S), the set of subsets of S, to {0,1}.
The count of antichains includes the empty antichain which contains no subsets and the antichain consisting of only the empty set.
a(n) is also equal to the **number** of upsets of an n-set S. A set U of subsets of S is an upset if whenever A is in U and B is a superset of A then B is in U. - W. Edwin Clark, Nov 06 2003
Also the **number** of simple games with n players in minimal winning form. - Fabián Riquelme, May 29 2011
The unlabeled case is A003182. - Gus Wiseman, Feb 20 2019
From Amiram Eldar, May 28 2021 and Michel Marcus, Apr 07 2023: (Start)
The terms were first calculated by:
a(0)-a(4) - **Dedekind** (1897)
a(5) - Church (1940)
a(6) - Ward (1946)
a(7) - Church (1965, verified by Berman and Kohler, 1976)
a(8) - Wiedemann (1991)
a(9) - Jäkel (2023)
a(9) - independently computed by Lennart Van Hirtum, Patrick De Causmaecker, Jens Goemaere, Tobias Kenter, Heinrich Riebler, Michael Lass, and Christian Plessl (2023)
(End)

A Computation of the 9th Dedekind Number using FPGA Supercomputing

Lennart Van Hirtum

Patrick De Causmaecker, Jens Goemaere, PC2

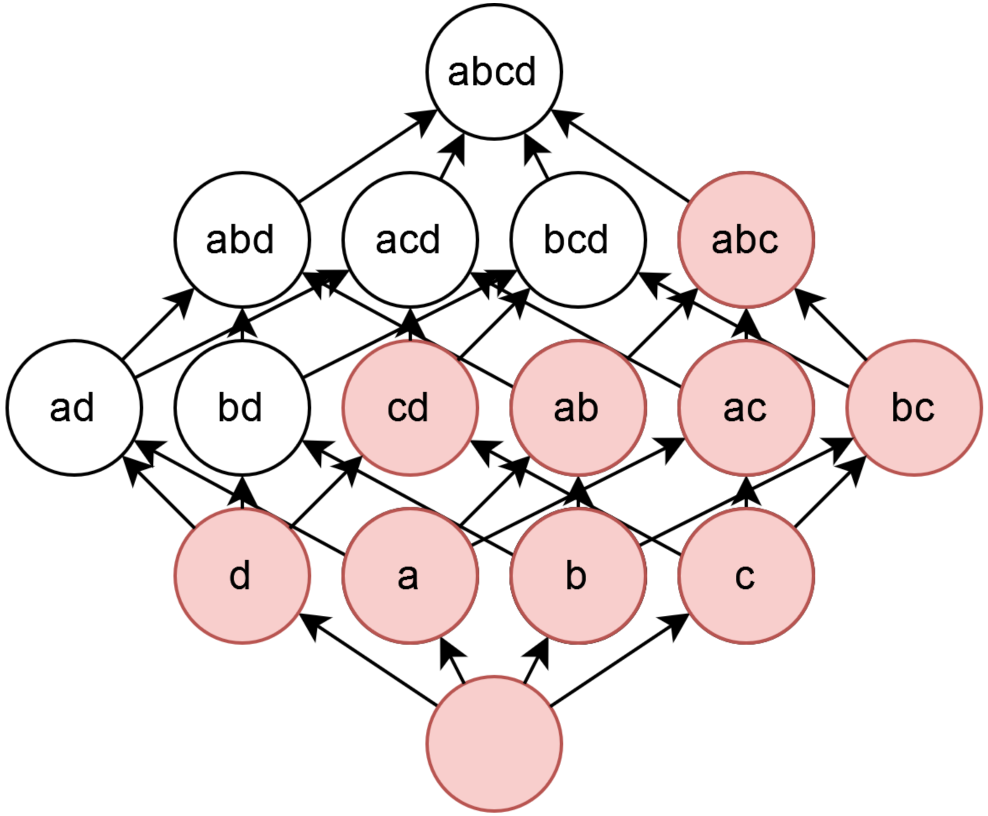
Paderborn University, Germany
Paderborn Center for Parallel Computing

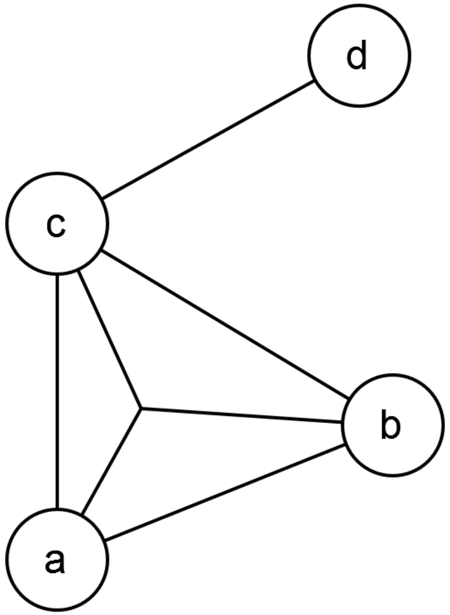
KU Leuven Arenberg, 6 July 2023

Paderborn Center for Parallel Computing

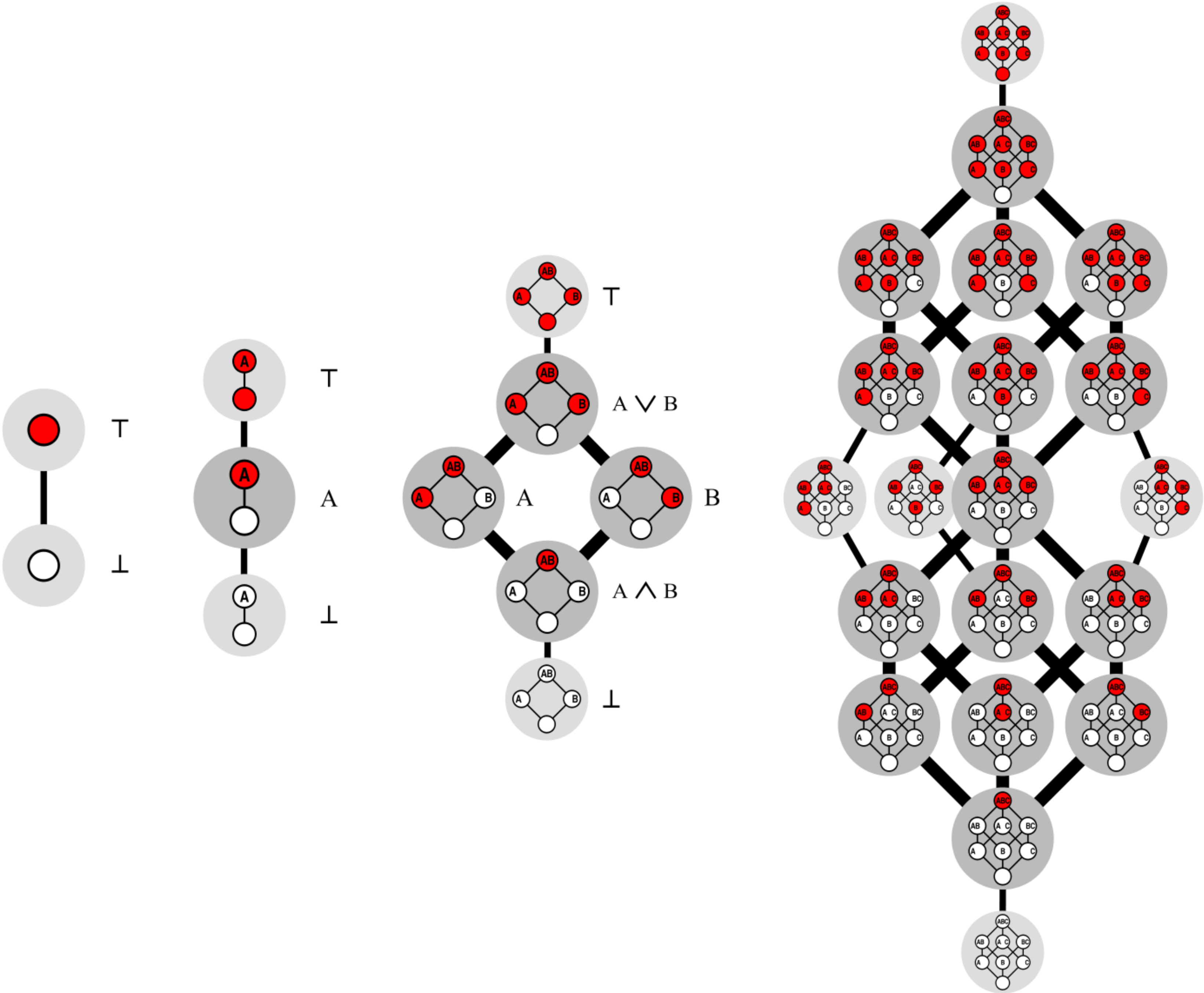
D(0) =	2	Dedekind (1897)
D(1) =	3	Dedekind (1897)
D(2) =	6	Dedekind (1897)
D(3) =	20	Dedekind (1897)
D(4) =	168	Dedekind (1897)
D(5) =	7581	Church (1940)
D(6) =	7828354	Ward (1946)
D(7) =	2414682040998	Church (1965)
D(8) =	56130437228687557907788	Wiedemann (1991)
D(9) =	286386577668298411128469151667598498812366	(2023)

Monotone Boolean Functions





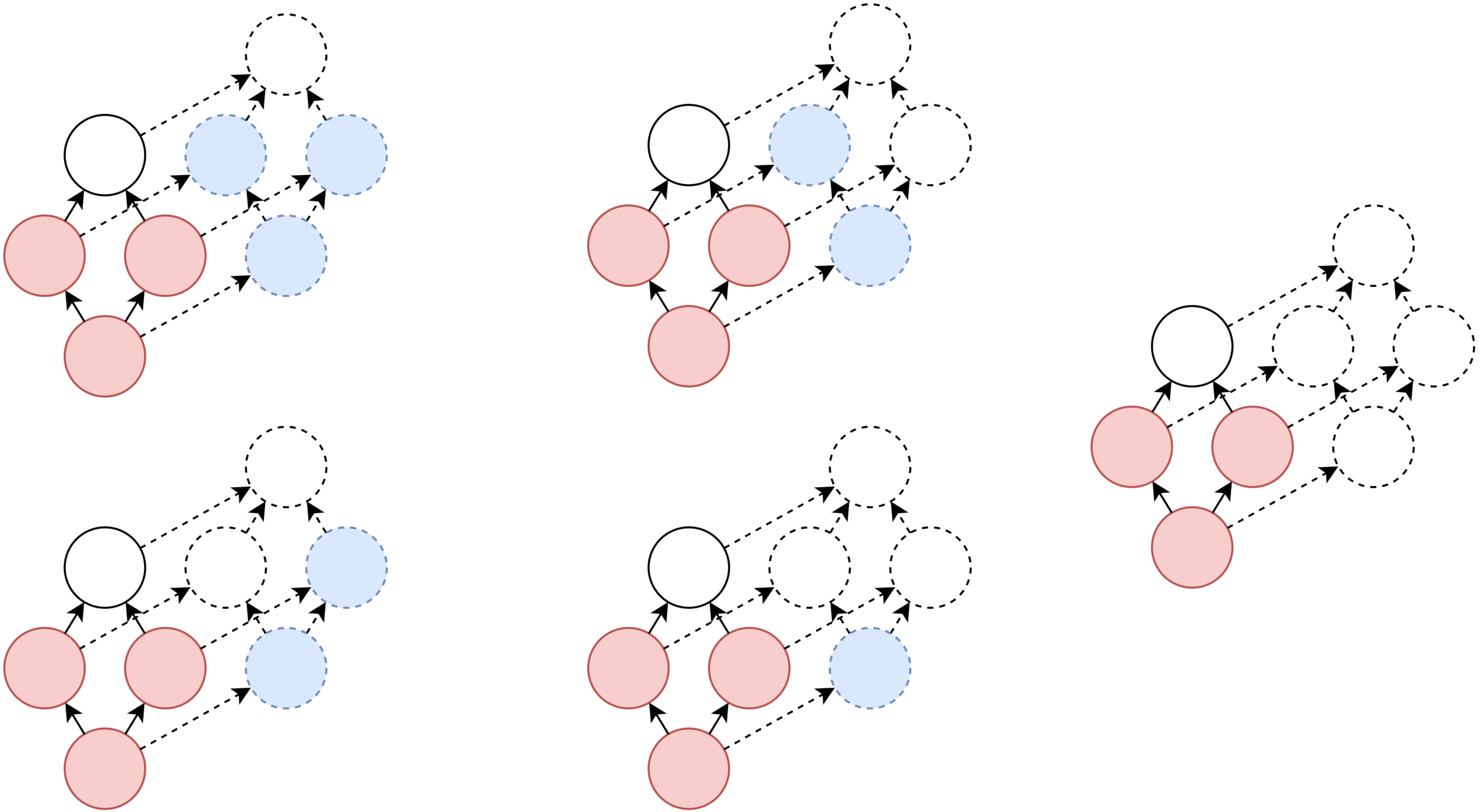
abcd	bcd	acd	cd	abd	bd	ad	d	abc	bc	ac	c	ab	b	a	


⊤
⊥
⊤
A
⊥
⊤
A ∨ B
A
B
A ∧ B
⊥

Jumping Formulas

$$D(n+1) = \sum_{\alpha \in A_n} |[\bot, \alpha]|$$

Core Idea



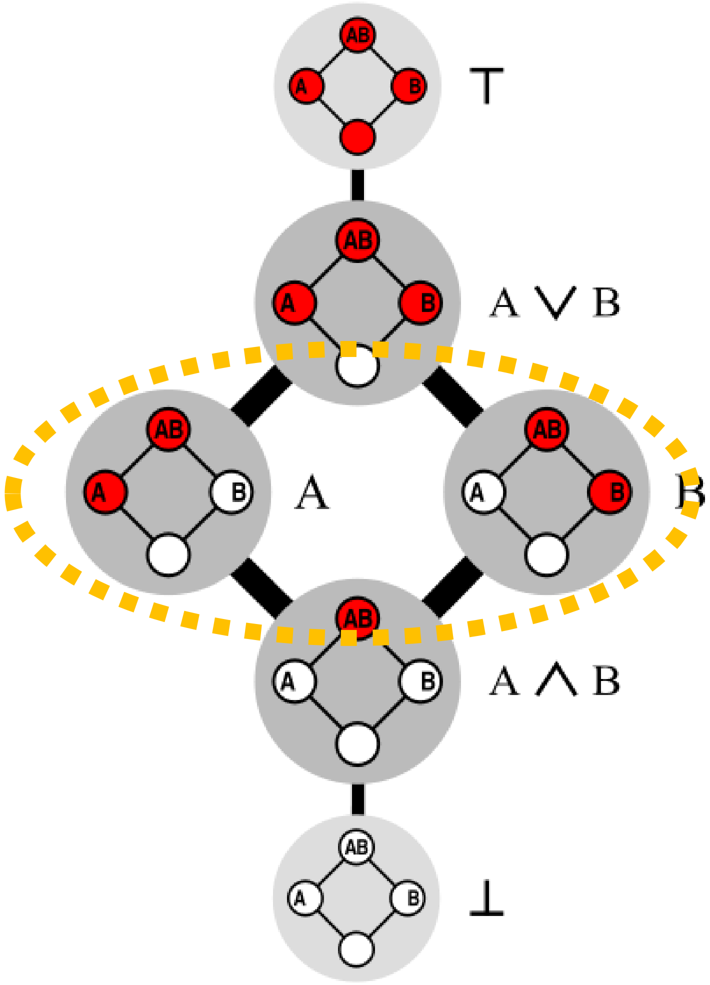
1-Jump

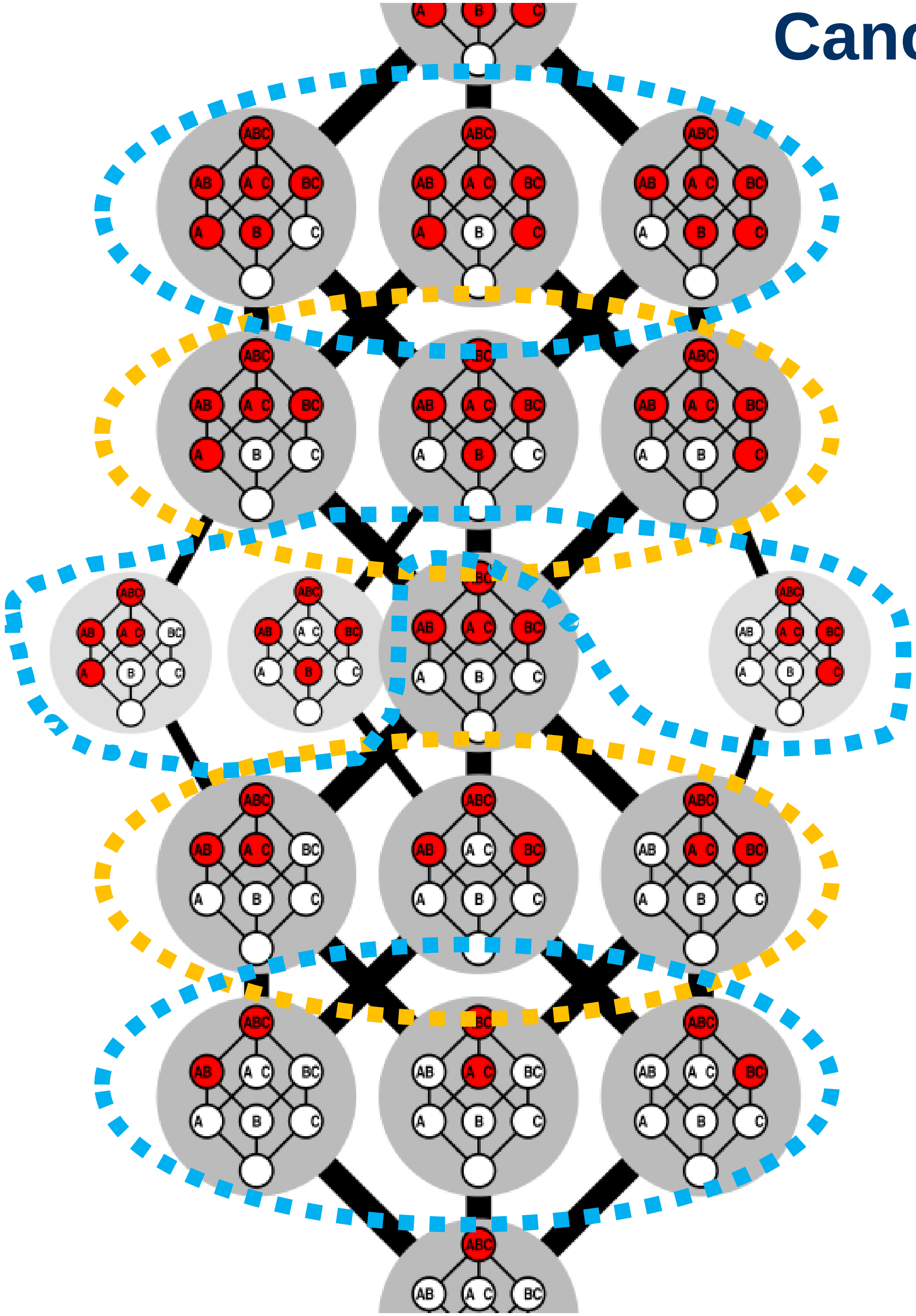
$$D(n+1) = \sum_{\alpha \in A_n} |[\bot, \alpha]|$$

$$\Downarrow$$

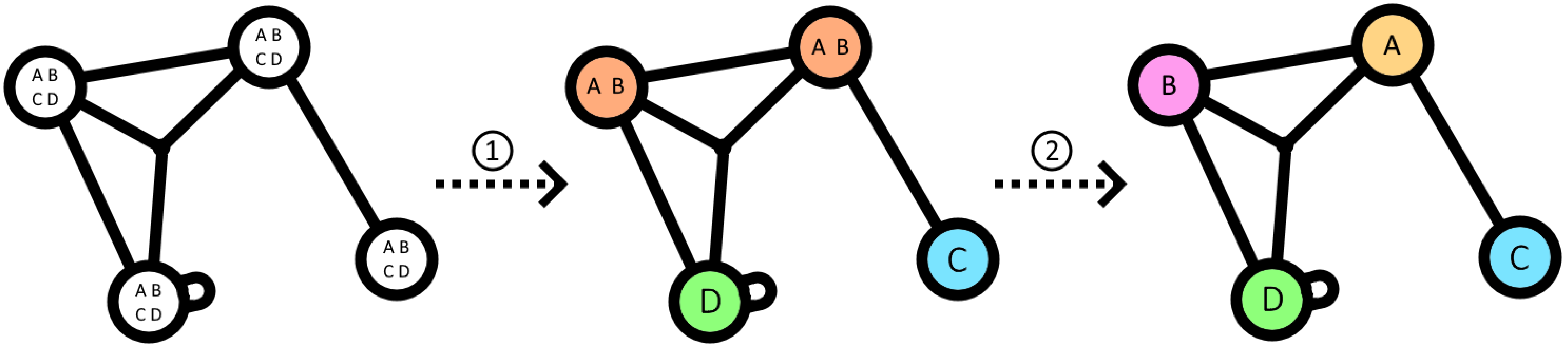
$$D(n+1) = \sum_{\alpha \in R_n} D_\alpha |[\bot, \alpha]|$$

Canonization





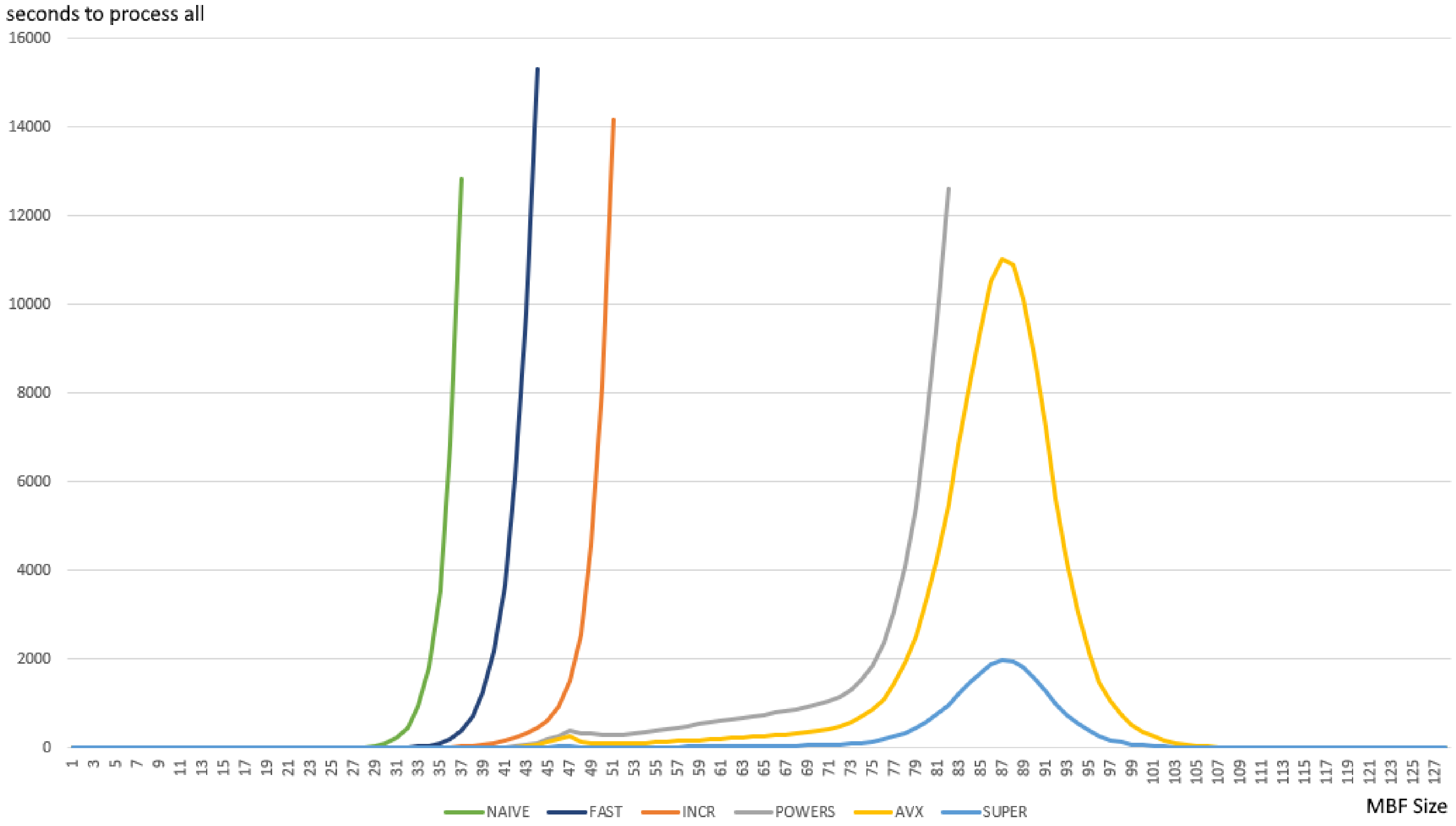
Canonization



R(0) =	2	
R(1) =	3	
R(2) =	5	
R(3) =	10	
R(4) =	30	
R(5) =	210	
R(6) =	16353	
R(7) =	490013148	Yusan (2012)
R(8) =	1392195548889993358	Pawelski (2021)
R(9) =	789204635842035040527740846300252680 (Paw 2023)	

P-Coëfficient Formula

$$D(n+2) = \sum_{\substack{\alpha,\beta \in A_n \\ \alpha \leq \beta}} |[\bot, \alpha]| 2^{C_{\alpha,\beta}} |[\beta, \top]|$$



seconds to process all
16000
14000
12000
10000
8000
6000
4000
2000
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99 101 103 105 107 109 111 113 115 117 119 121 123 125 127
MBF Size
NAIVE
FAST
INCR
POWERS
AVX
SUPER

Expanded P-Coëfficient Formula

$$D(n+2) = \sum_{\substack{\alpha,\beta \in A_n \\ \alpha \leq \beta}} |[\bot, \alpha]| 2^{C_{\alpha,\beta}} |[\beta, \top]|$$

$$D(n+2) = \sum_{\alpha \in R_n} |[\bot, \alpha]| D_\alpha \sum_{\substack{\beta \in R_n \\ \exists \delta \simeq \beta : \alpha \leq \delta}} |[\beta, \top]| \frac{D_\beta}{n!} \sum_{\substack{\gamma \in Permut_\beta \\ \alpha \leq \gamma}} 2^{C_{\alpha,\gamma}}$$

$$|[\bot,\alpha]|2^{C_{\alpha,\beta}}|[\beta,\top]| = |[\overline{\alpha},\top]|2^{C_{\overline{\beta},\overline{\alpha}}}|[\bot,\overline{\beta}]|$$

Expanded P-Coëfficient Formula

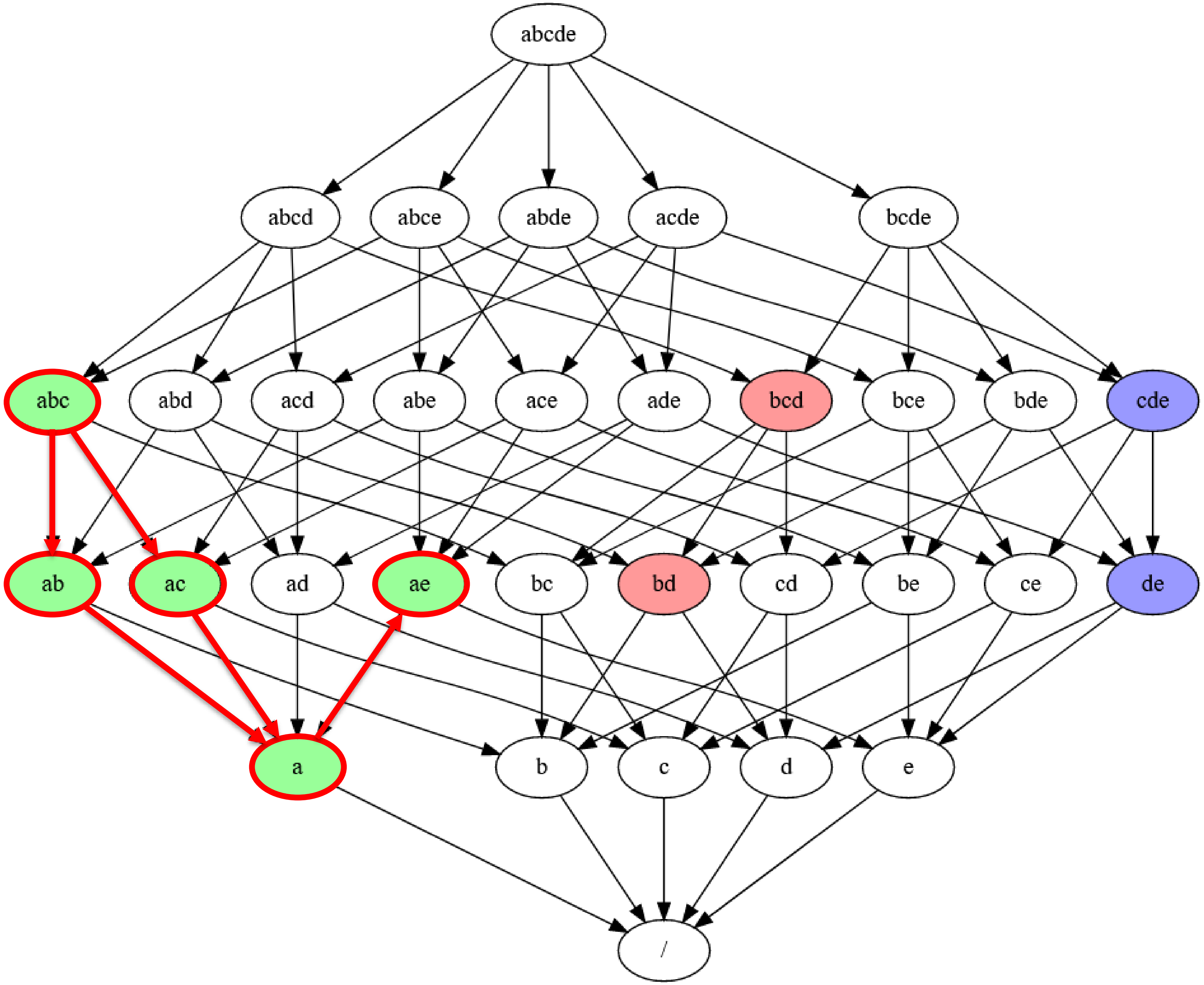
$$D(n+2) = \sum_{\alpha \in R_n} |[\bot, \alpha]| D_\alpha \sum_{\substack{\beta \in R_n \\ \exists \delta \simeq \beta : \alpha \leq \delta}} |[\beta, \top]| \frac{D_\beta}{n!} \sum_{\substack{\gamma \in Permut_\beta \\ \alpha \leq \gamma}} 2^{C_{\alpha,\gamma}}$$

490M *45M *5040

$5.57 * 10^{18}$ $C_{\alpha,\gamma}$ values in total!

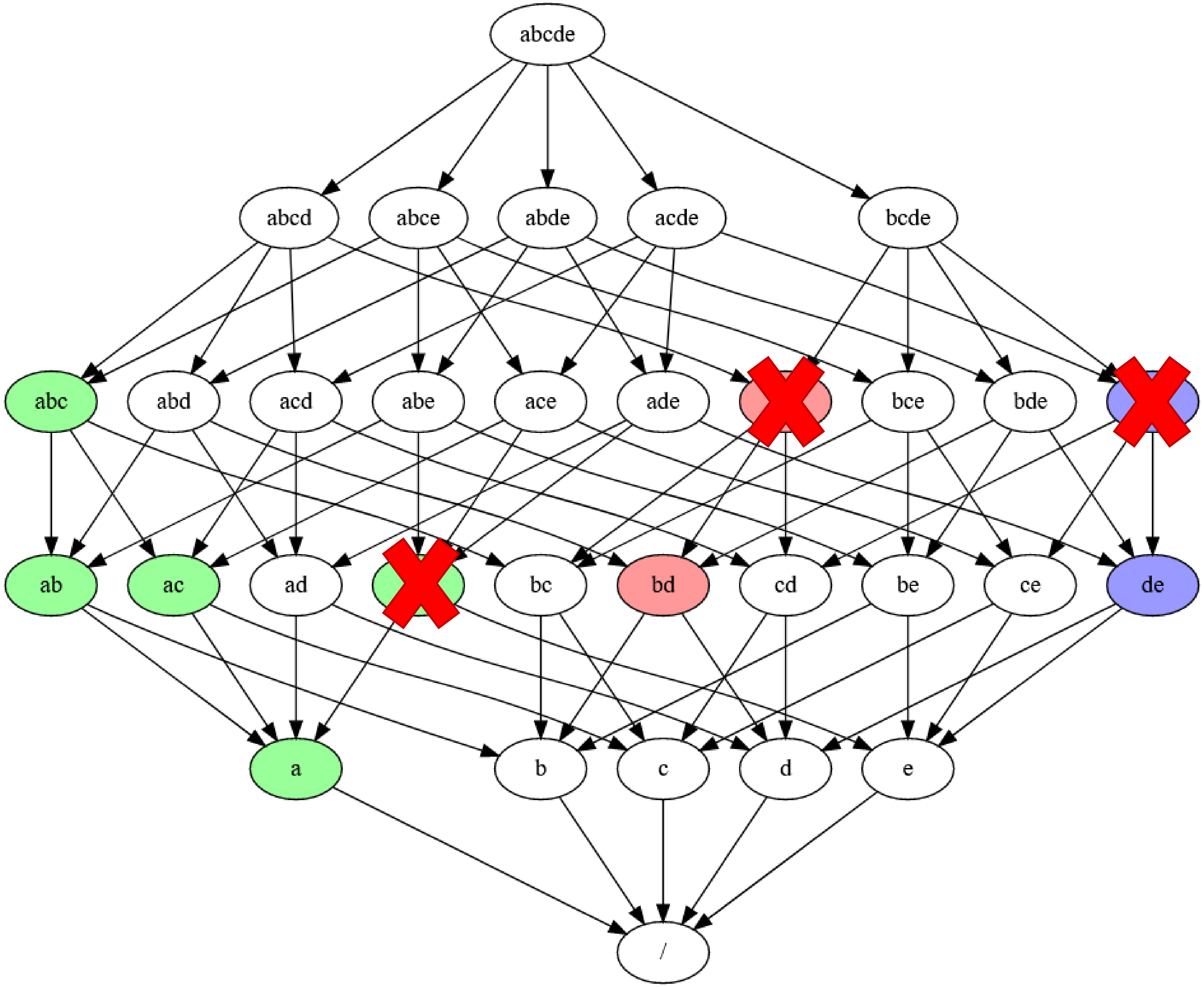
FloodFill Algorithm

FloodFill Algorithm



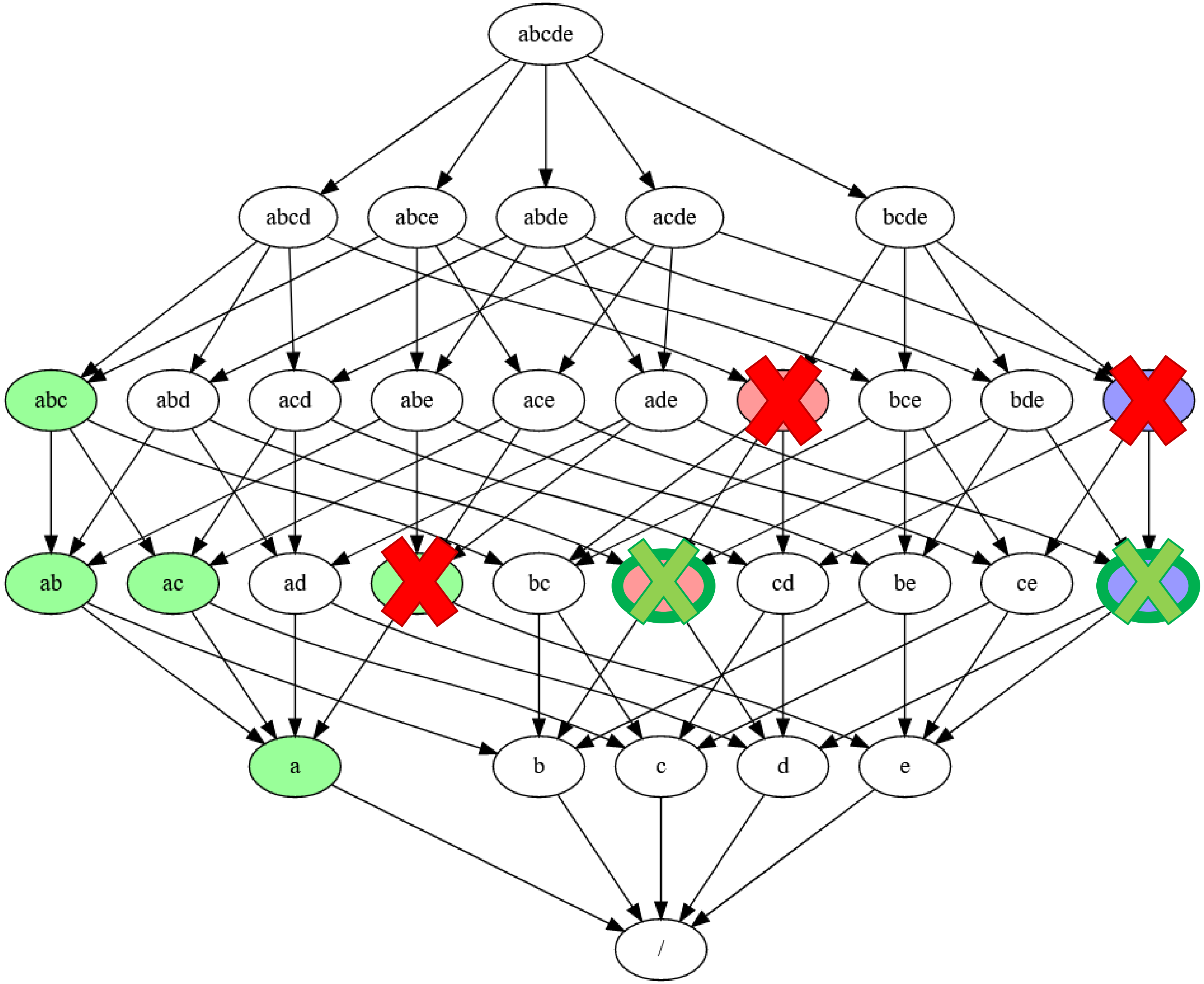
$$C_{\alpha,\beta} = 3$$

Leaf Elimination



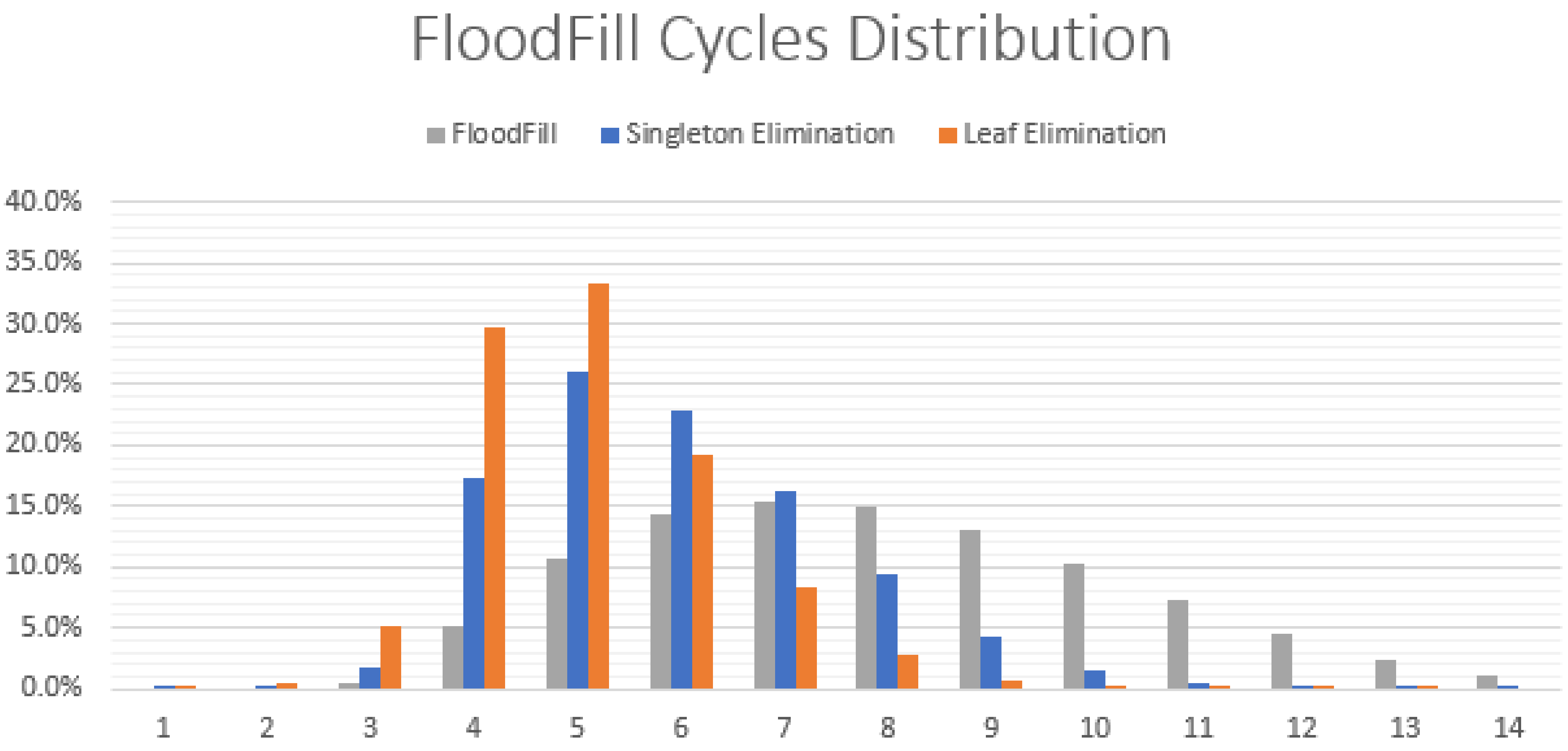
- LE Up
- LE Down

Singleton Elimination



FloodFill Algorithm

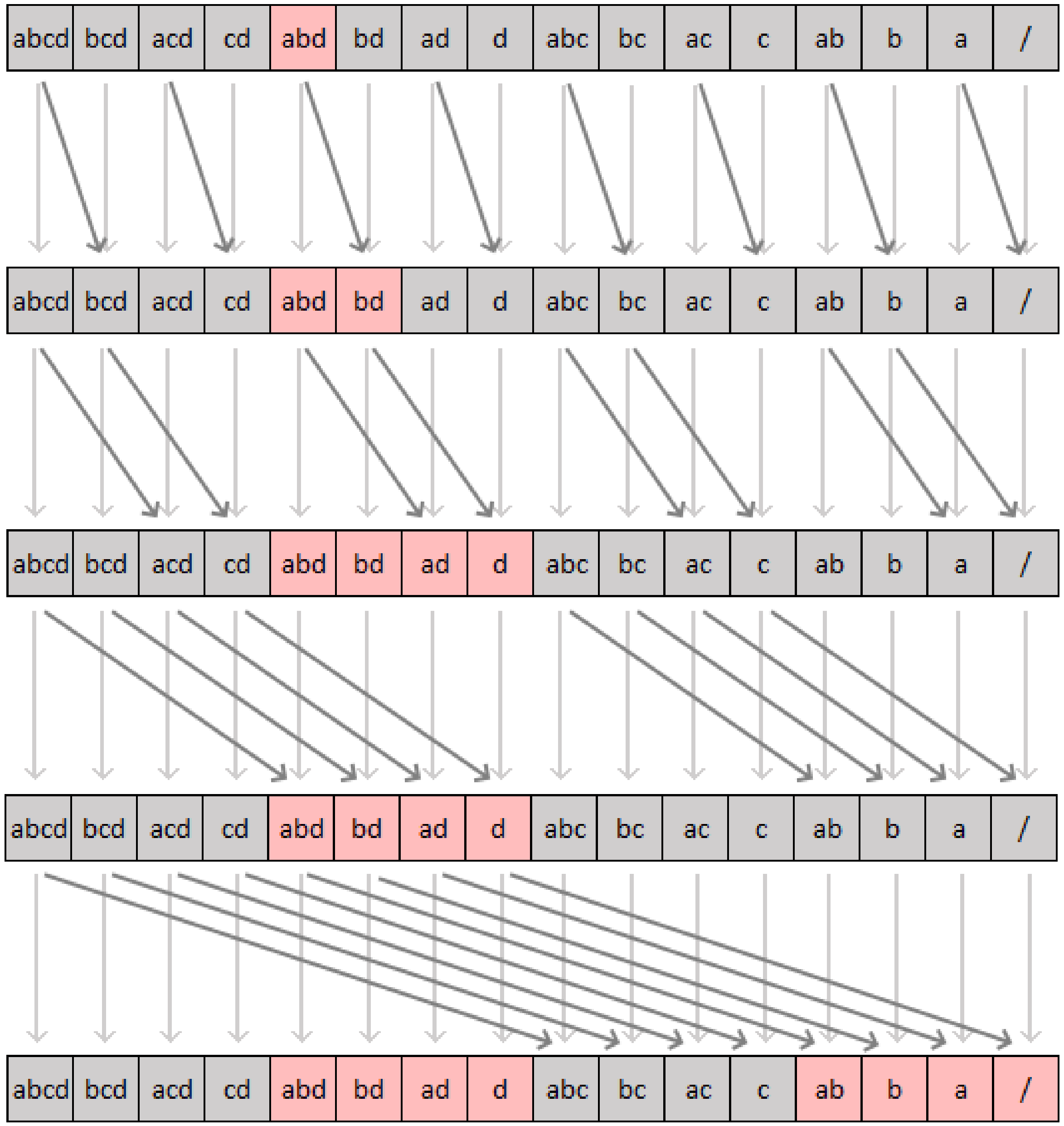
```
def countConnected(MBF α, MBF γ):
  BF graph = α & ¬γ
  graph = eliminateLeafesUp(graph)
  graph, int count = eliminateSingletons(graph)
  while graph not empty:
    BF seed = firstNode(graph)
    do:
      BF seedUp = monotonizeUp(seed) & graph
      seed = monotonizeDown(seedUp) & graph
    while seedUp != seed
    graph = graph & ¬ seed
    count++
  return count
```

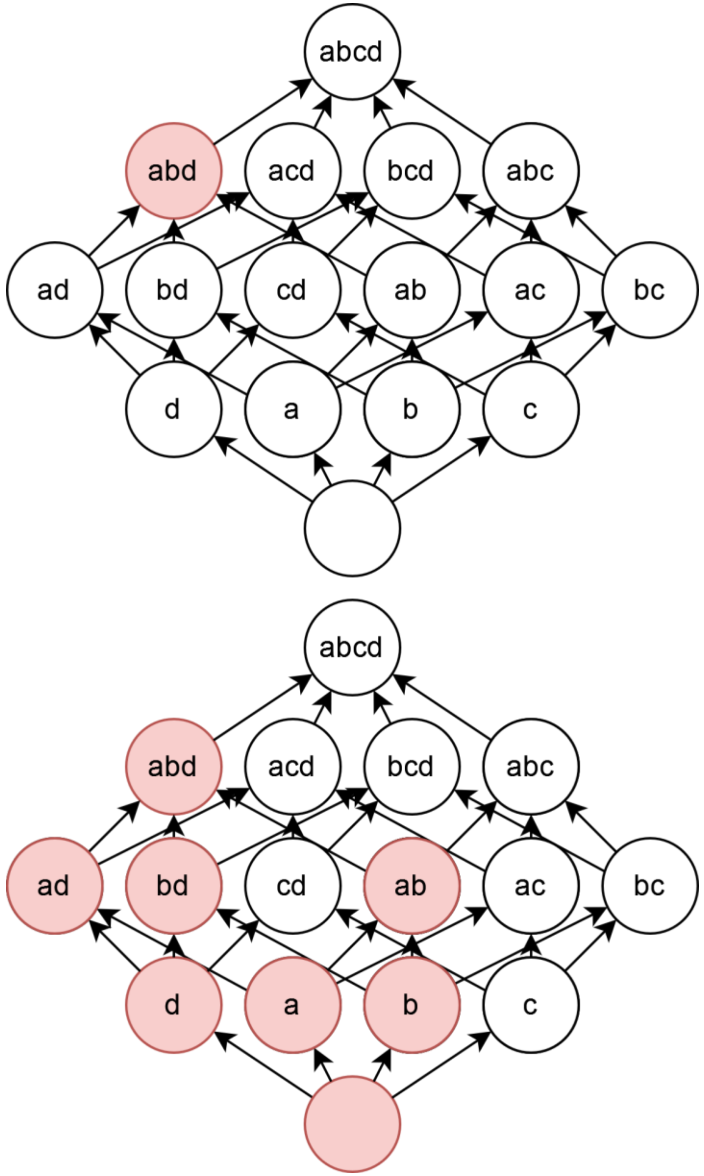


FloodFill Cycles Distribution
FloodFill
Singleton Elimination
Leaf Elimination
40.0%
35.0%
30.0%
25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
1
2
3
4
5
6
7
8
9
10
11
12
13
14

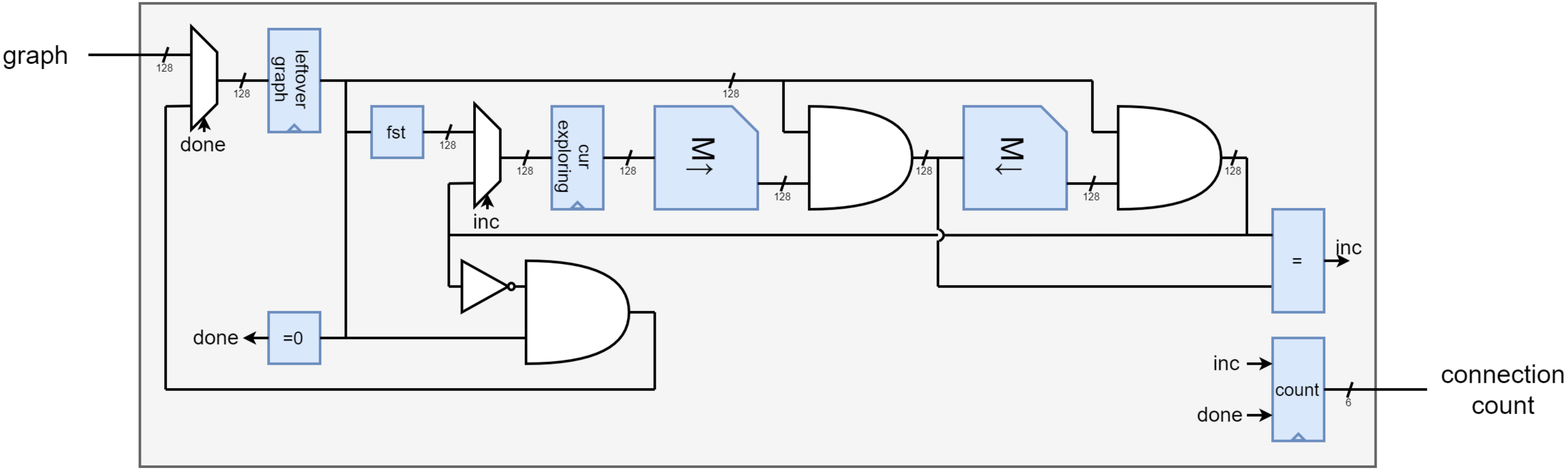
Hardware Implementation

Monotonization

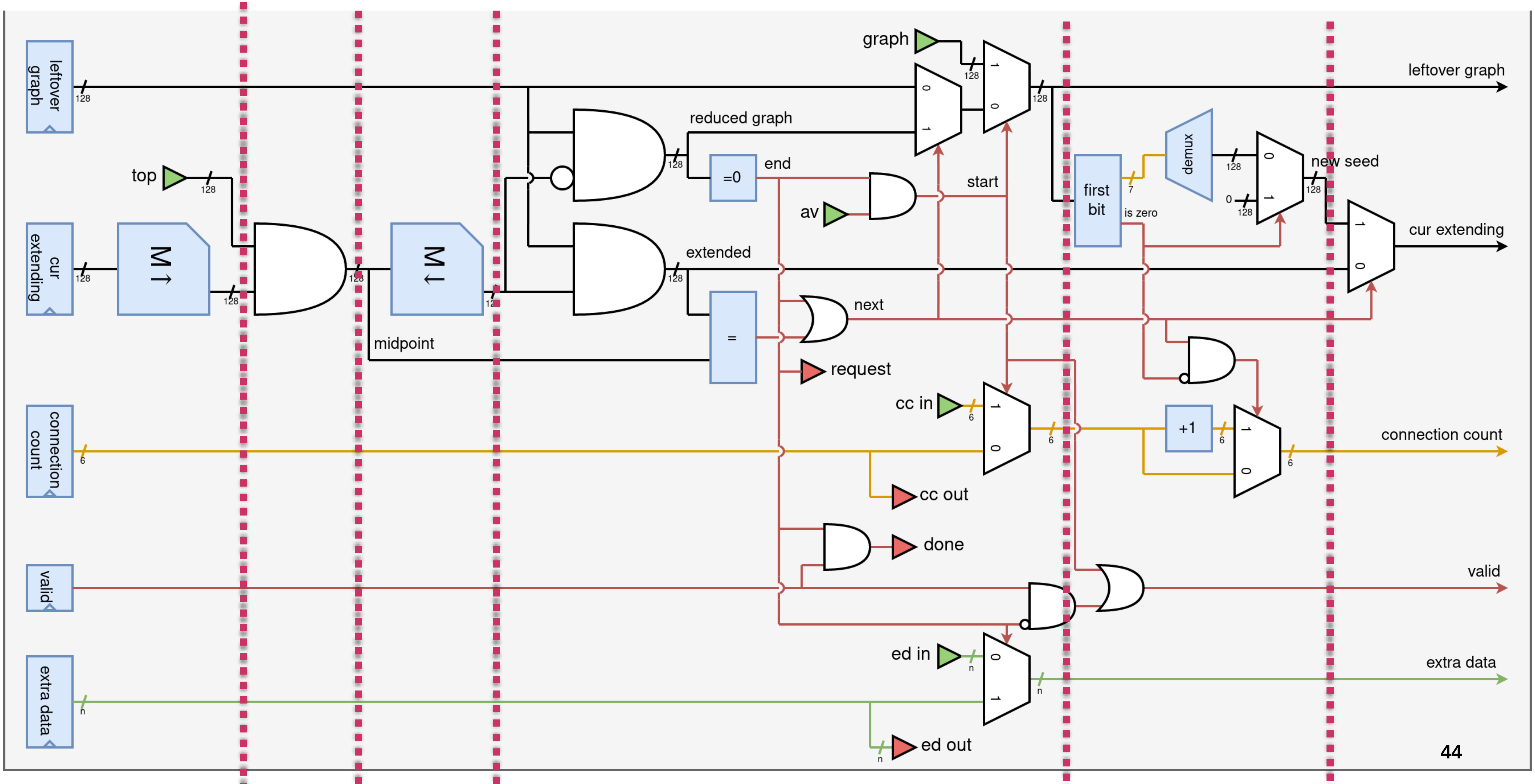




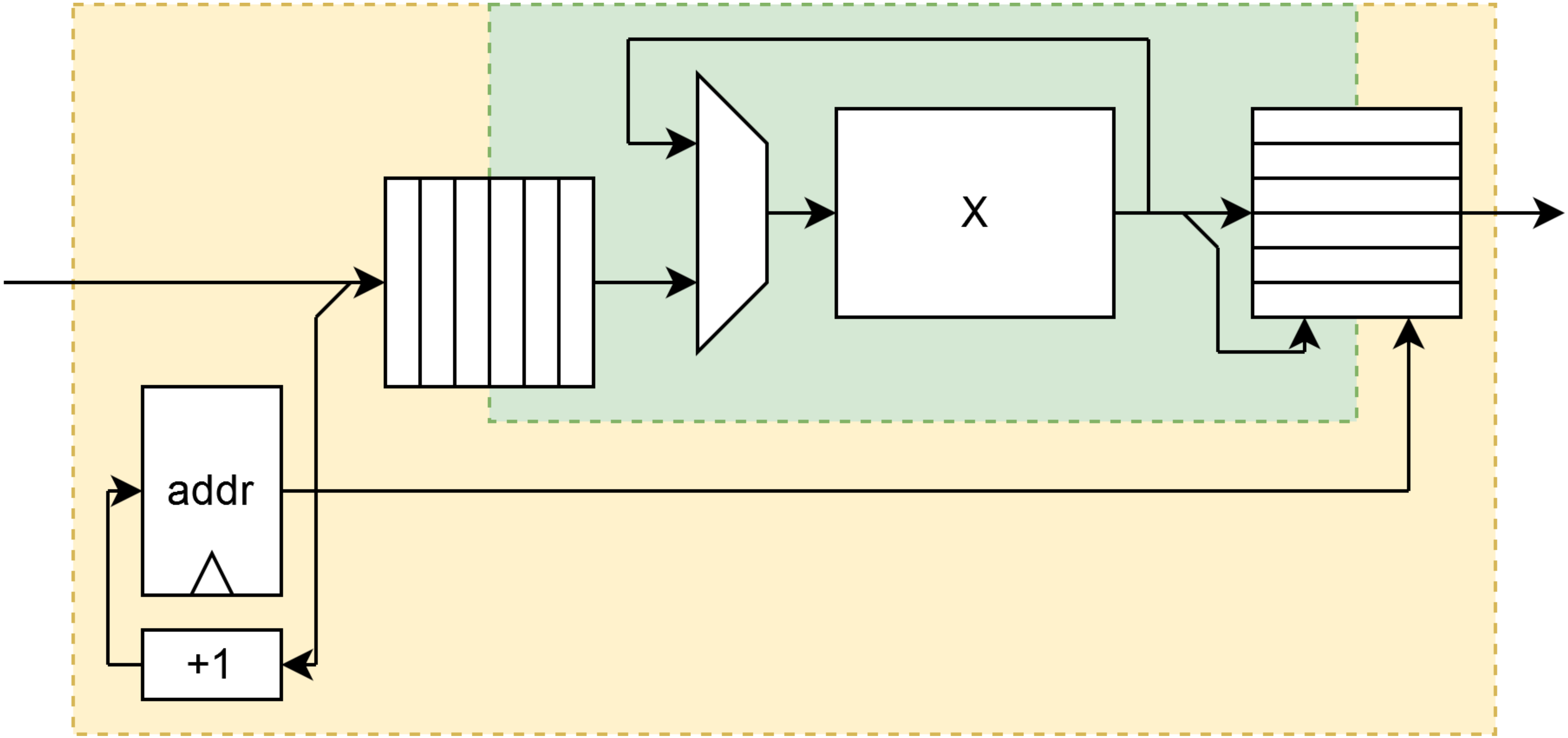
Count Connected Core



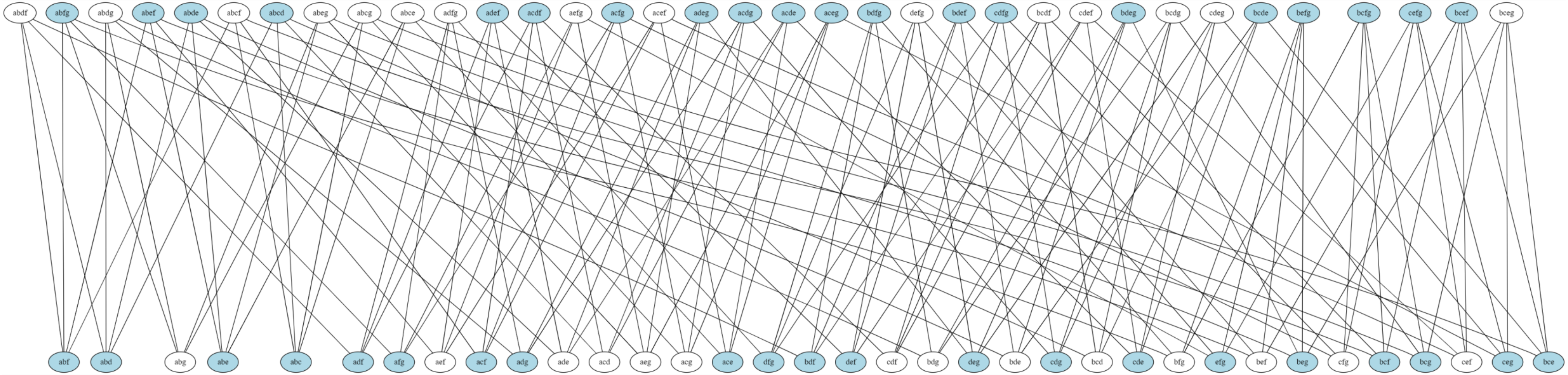
Pipelined


leftover graph
cur extending
connection count
valid
extra data
top
graph
reduced graph
end
start
av
extended
next
midpoint
request
cc in
cc out
done
ed in
ed out
first bit
is zero
demux
new seed
leftover graph
cur extending
connection count
valid
extra data
=0
=
+1
128
6
n
7

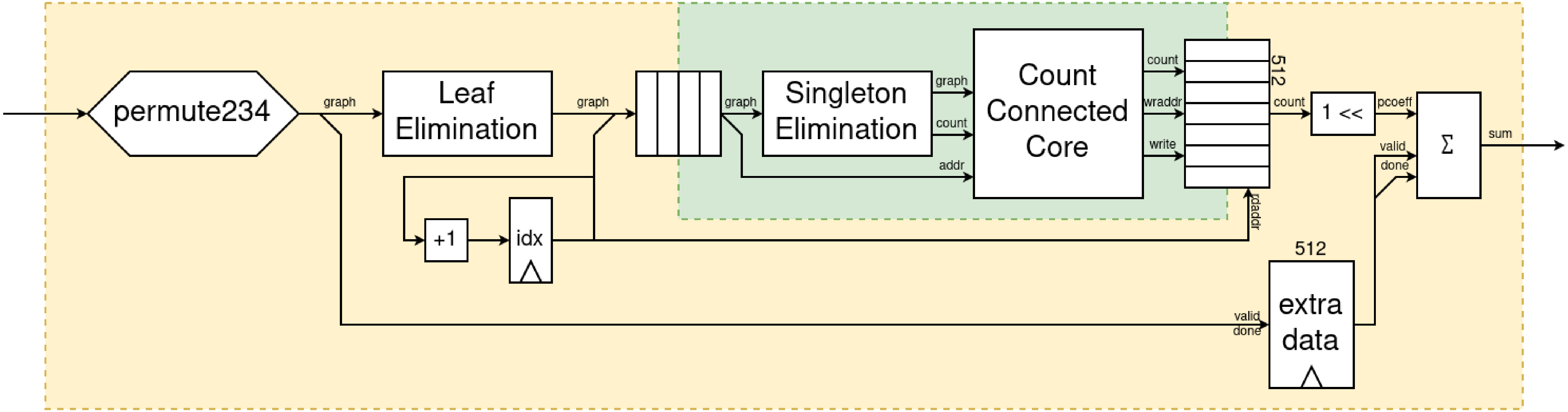
Loop implementation

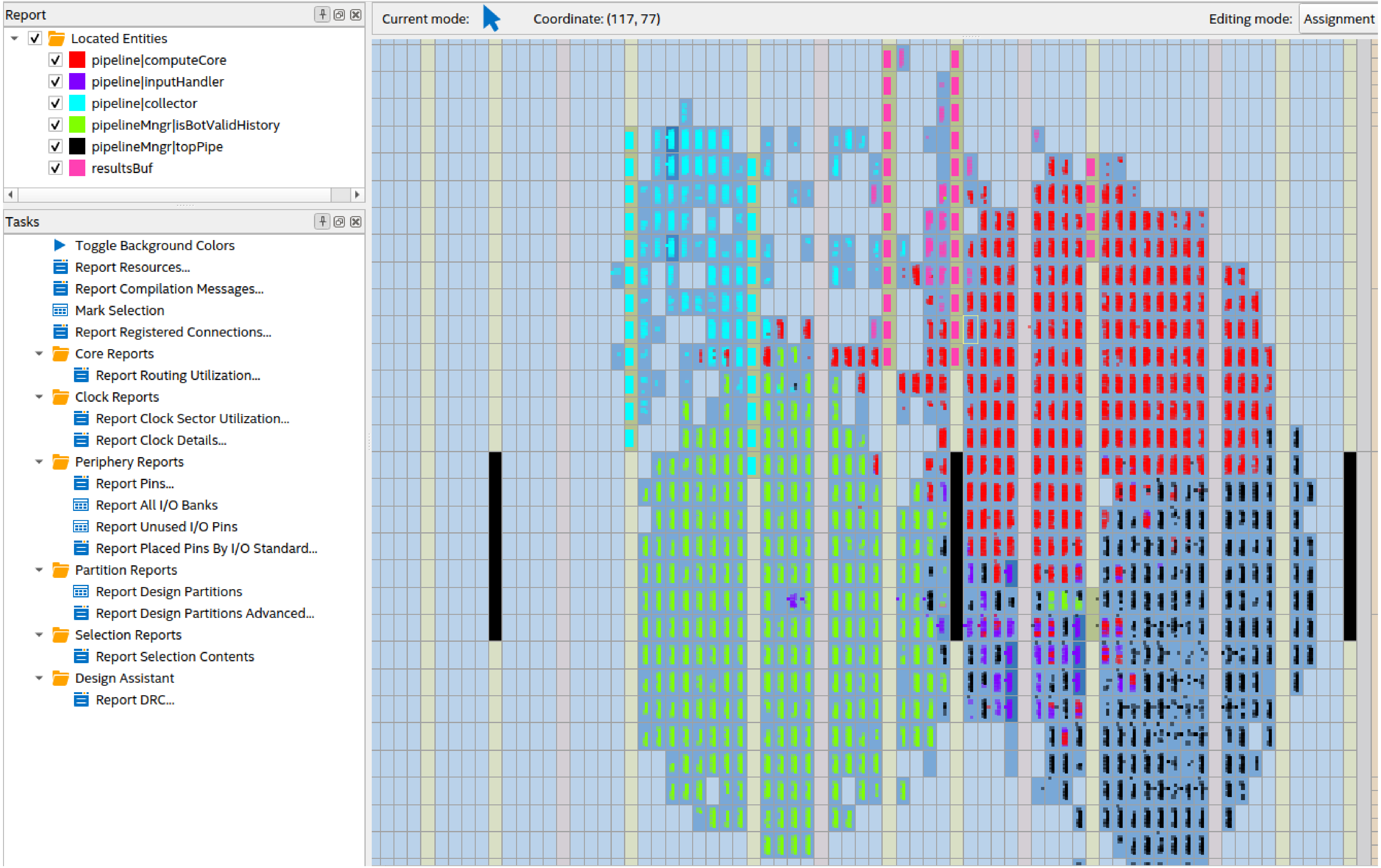


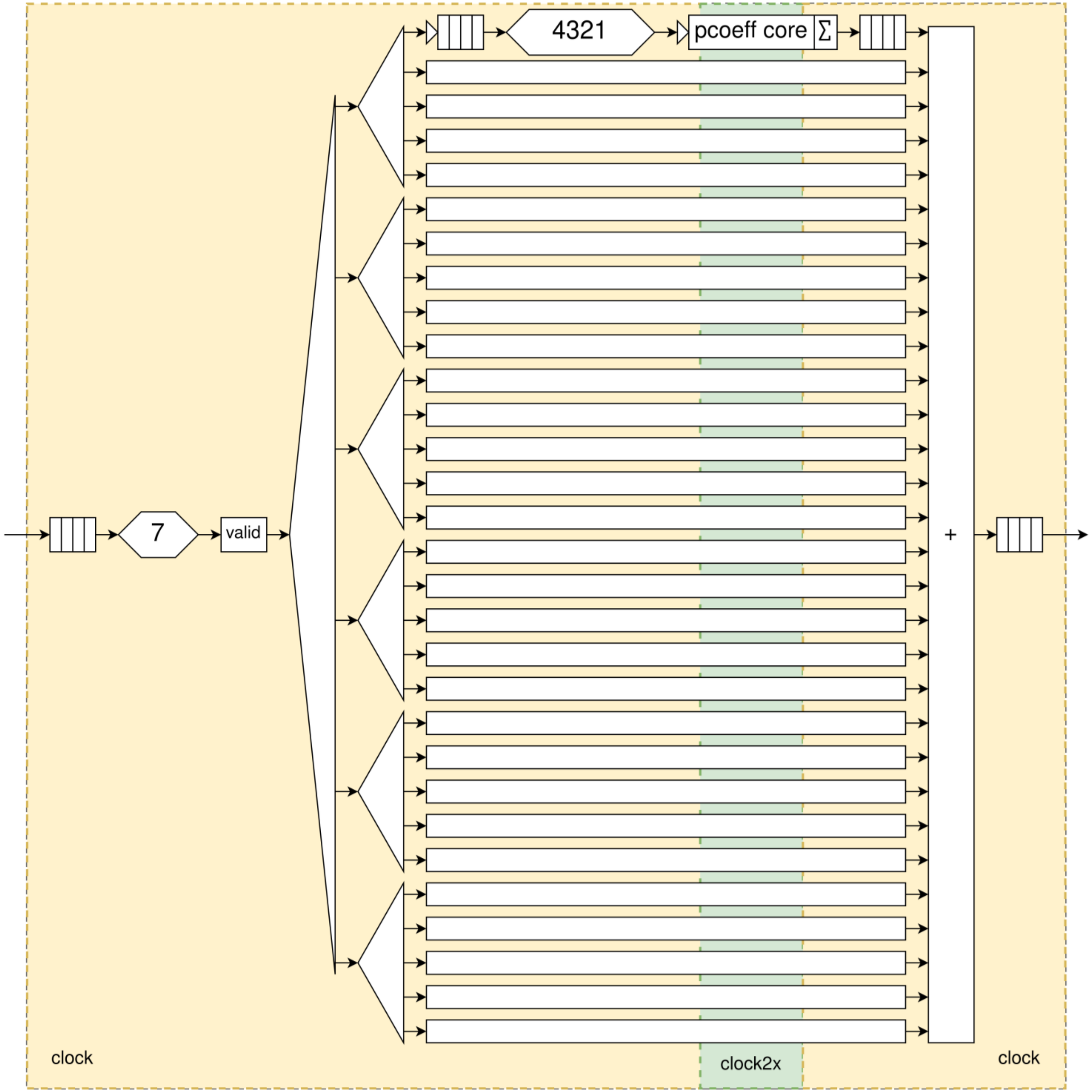
Worst Case


abdf
abfg
abdg
abef
abde
abcf
abcd
abeg
abcg
abce
adfg
adef
acdf
aefg
acfg
acef
adeg
acdg
acde
aceg
bdfg
defg
bdef
cdfg
bcdf
cdef
bdeg
bcdg
cdeg
bcde
befg
bcfg
cefg
bcef
bceg
abf
abd
abg
abe
abc
adf
afg
aef
acf
adg
ade
acd
aeg
acg
ace
dfg
bdf
def
cdf
bdg
deg
bde
cdg
bcd
cde
bfg
efg
bef
beg
cfg
bcf
bcg
cef
ceg
bce

Processing Module




Report
Located Entities
pipeline|computeCore
pipeline|inputHandler
pipeline|collector
pipelineMngr|isBotValidHistory
pipelineMngr|topPipe
resultsBuf
Tasks
Toggle Background Colors
Report Resources...
Report Compilation Messages...
Mark Selection
Report Registered Connections...
Core Reports
Report Routing Utilization...
Clock Reports
Report Clock Sector Utilization...
Report Clock Details...
Periphery Reports
Report Pins...
Report All I/O Banks
Report Unused I/O Pins
Report Placed Pins By I/O Standard...
Partition Reports
Report Design Partitions
Report Design Partitions Advanced...
Selection Reports
Report Selection Contents
Design Assistant
Report DRC...
Current mode:
Coordinate: (117, 77)
Editing mode:
Assignment

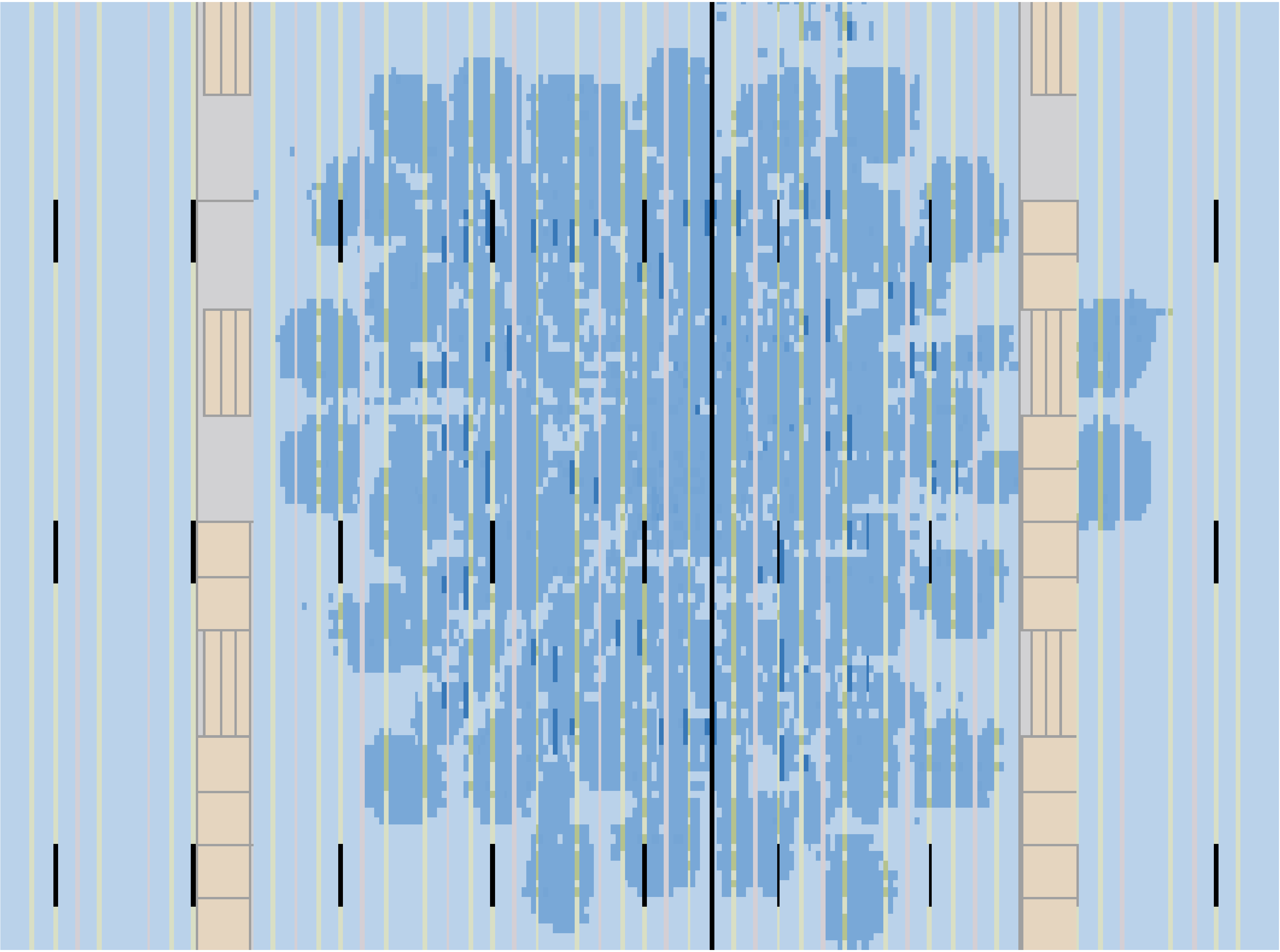


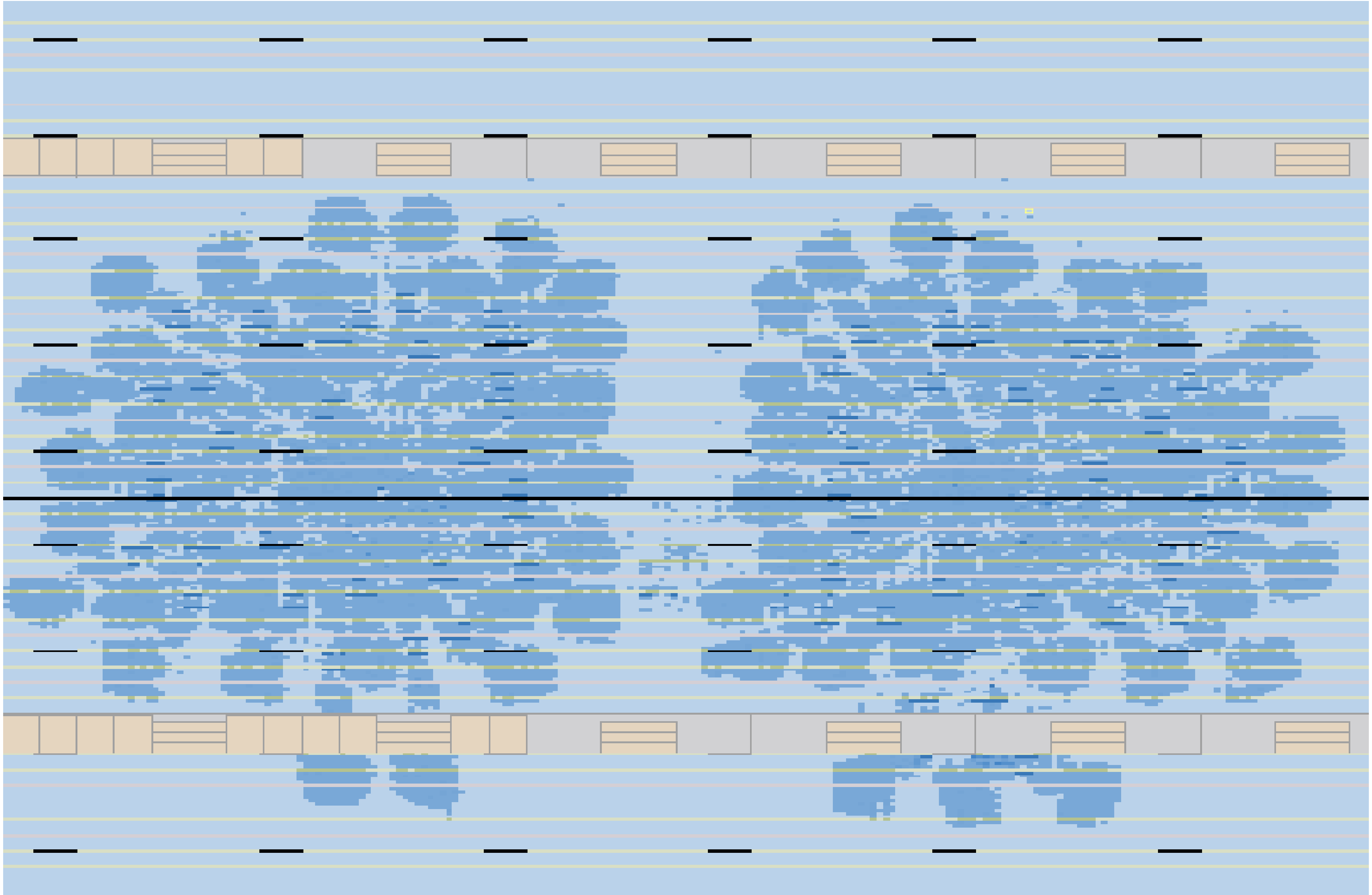
5040 Permutation splitter

$$\sum_{\substack{\gamma \in Permut_\beta \\ \alpha \leq \gamma}} 2^{C_{\alpha,\gamma}}$$

$7! = 5040$

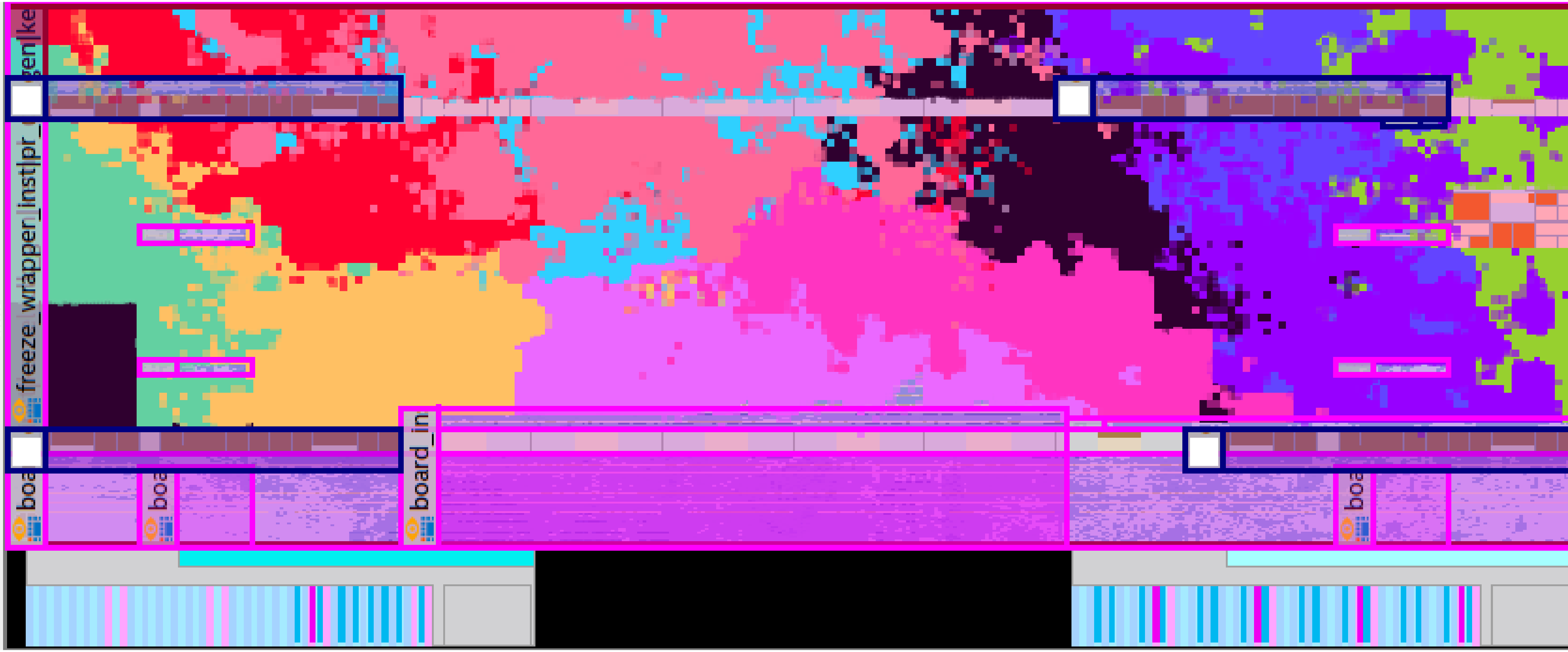
permutations





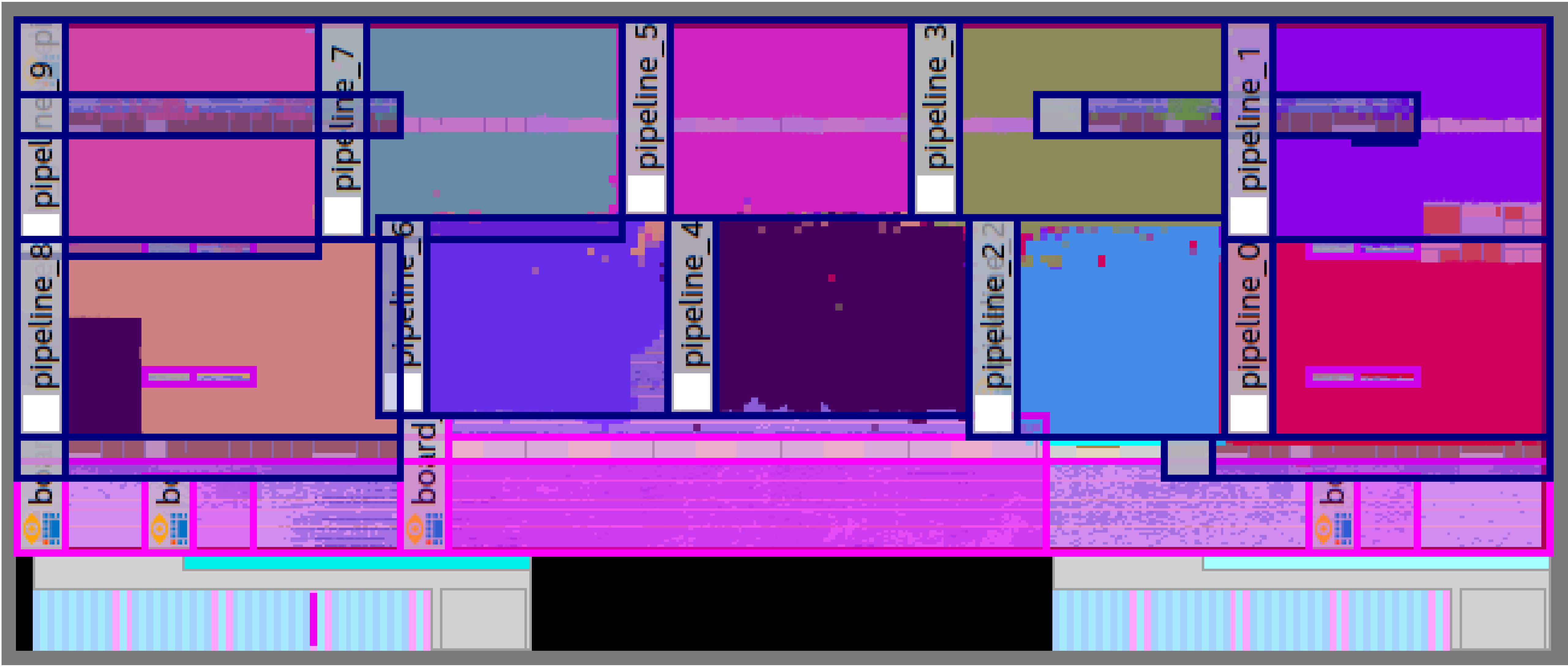
6 pipelines

10 pipelines



Terrible performance!

Adding logic lock regions



450 MHz! 92% Logic Density 300 Mbots/s

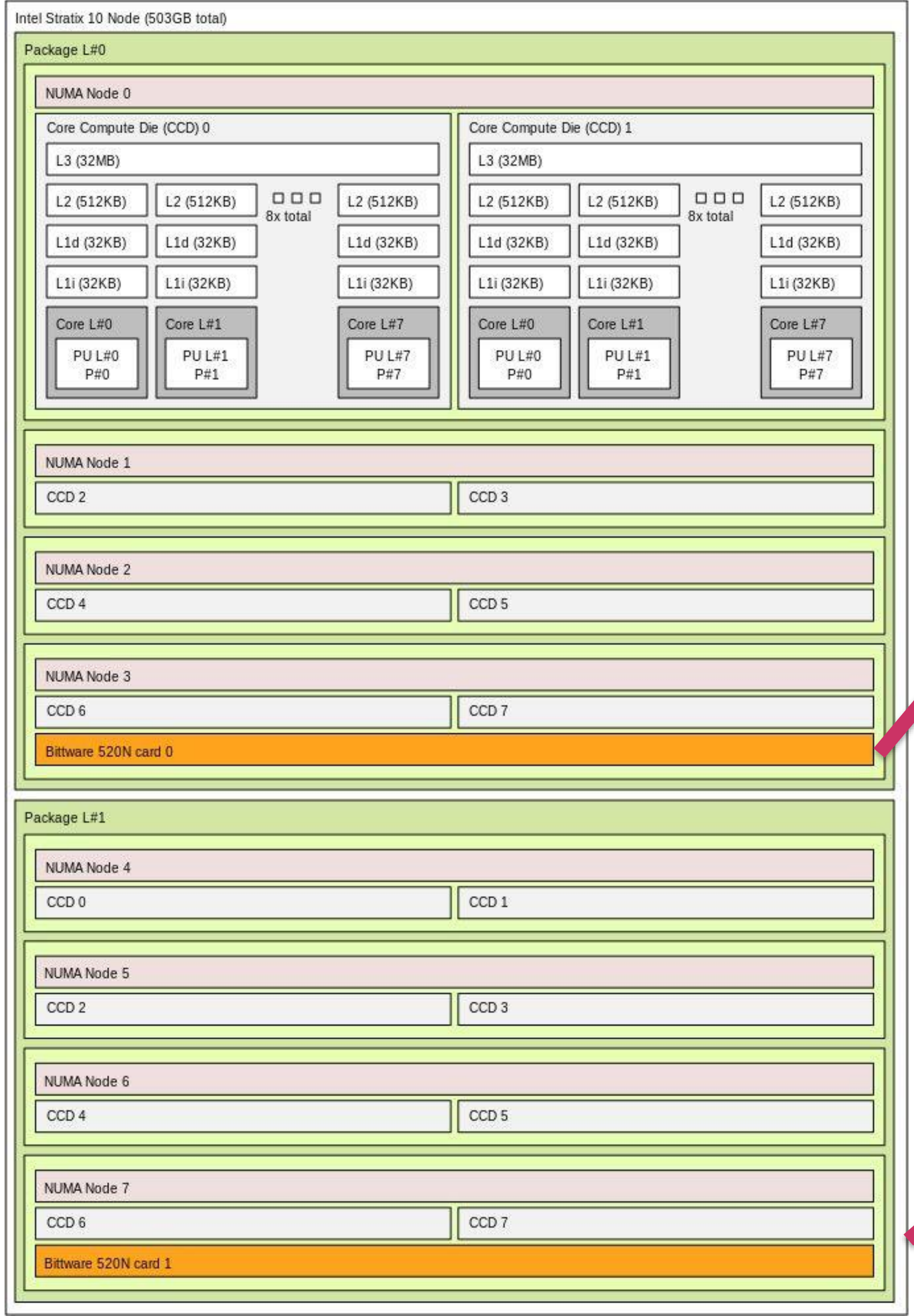
64 Cores
~0.6M bots/sec

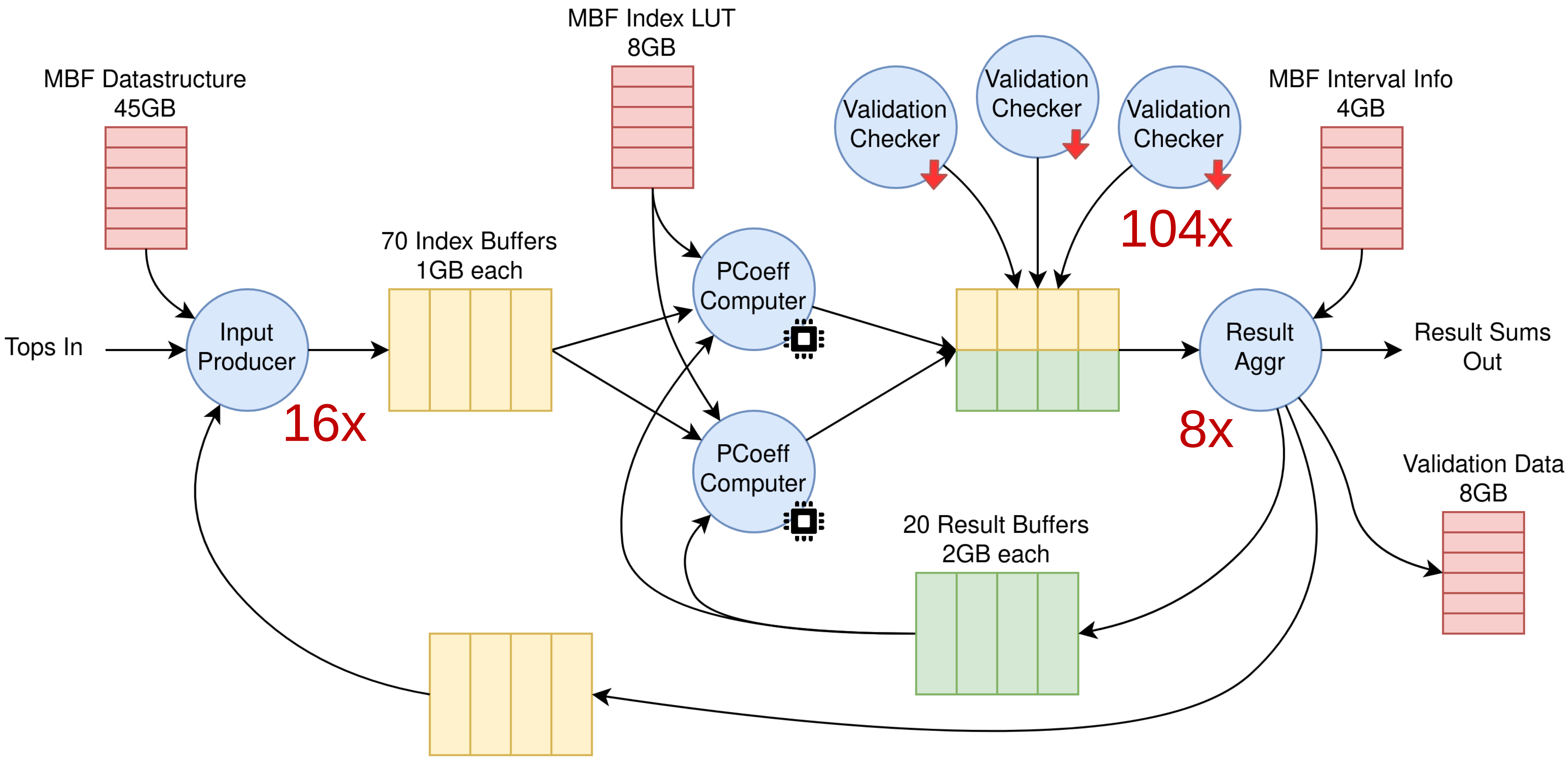
300 P-Coëff Cores
300M bots/sec

500x
Faster!

Supercomputing

Noctua 2 FPGA Node




MBF Datastructure
45GB
MBF Index LUT
8GB
Validation Checker
Validation Checker
Validation Checker
MBF Interval Info
4GB
70 Index Buffers
1GB each
104x
Tops In
Input Producer
PCoeff Computer
Result Aggr
Result Sums Out
16x
8x
PCoeff Computer
Validation Data
8GB
20 Result Buffers
2GB each

Pcoëfficiënt Recap

$$D(n+2) = \sum_{\alpha \in R_n} |[\bot, \alpha]| D_\alpha \sum_{\substack{\beta \in R_n \\ \exists \delta \simeq \beta : \alpha \leq \delta}} |[\beta, \top]| \frac{D_\beta}{n!} \sum_{\substack{\gamma \in Permut_\beta \\ \alpha \leq \gamma}} 2^{C_{\alpha,\gamma}}$$

5.8 α per second
490 million in total

Computation on Noctua 2

- 15’000 jobs
- 33’000 tops/job
- 100 mins / job
- 16 FPGA servers

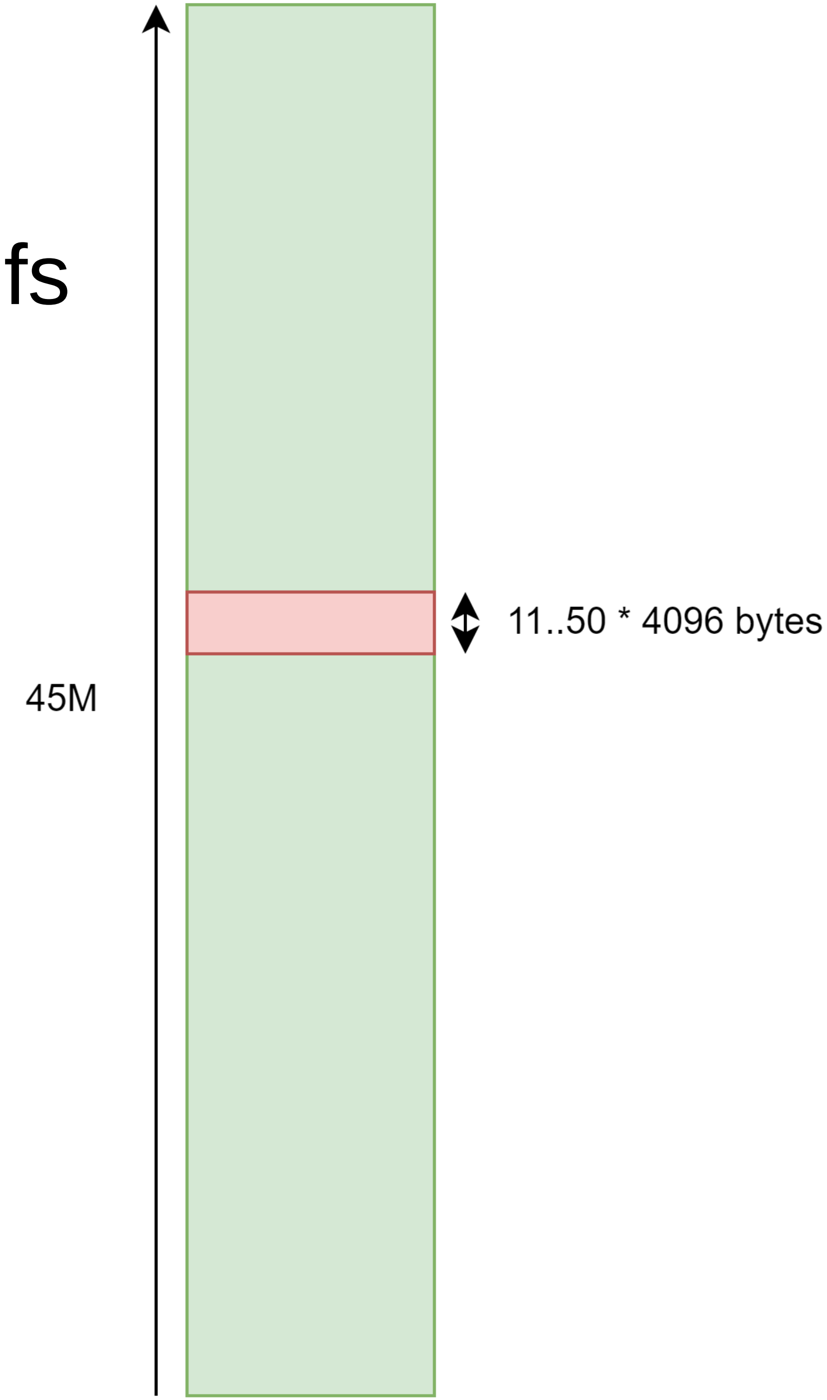
Computation on Noctua 2

- After 4 months on Noctua 2
- 47’000 FPGA hours in total

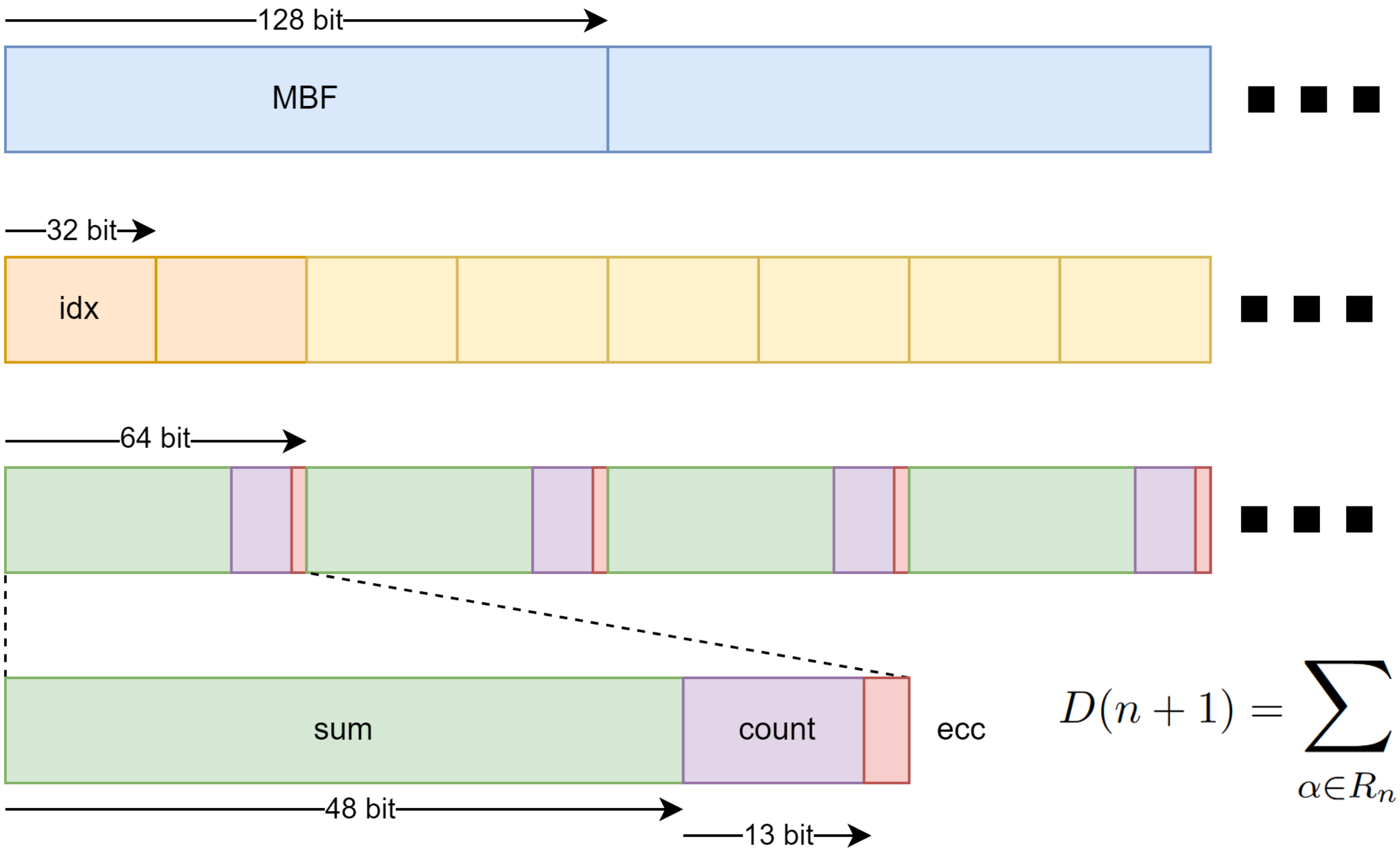
March 8th 2023, at 5pm:

286386577668298411128469151667598498812366

- Large error blocks in result bufs
- Aligned to 4096 bytes
- “Forgot to copy”?
- Fixed now



Errors



$$\sum_{\substack{\gamma\in Permut_\beta \\ \alpha\le\gamma}} 2^{C_{\alpha,\gamma}} \qquad \sum_{\substack{\gamma\in Permut_\beta \\ \alpha\le\gamma}} 1$$

$$D(n+1) = \sum_{\alpha\in R_n} D_\alpha \sum_{\substack{\beta\in R_n \\ \exists\delta\simeq\beta:\alpha\le\delta}} \frac{D_\beta}{n!} \sum_{\substack{\gamma\in Permut_\beta \\ \alpha\le\gamma}} 1$$

Lucky Checksum!

Mathematics

Mathematicians calculate 42-digit number after decades of trying

Dedekind numbers describe the number of ways sets of logical operations can be combined, and are fiendishly difficult to calculate, with only eight known since 1991 - and now mathematicians have calculated the ninth in the series

By Alex Wilkins

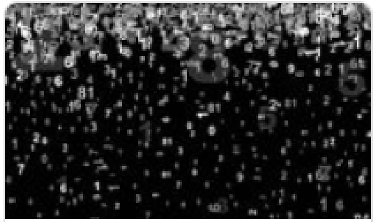
3 July 2023

https://phys.org › news › 2023-06-ninth-dedekind-scientists-long-known-problem.html

Ninth Dedekind number discovered: Scientists solve long-known pro...

26 Jun 2023 · June 26, 2023 Editors' notes Ninth **Dedekind number** discovered: Scientists solve long-known problem in mathematics by Universität Paderborn Credit: Unsplash/CC0 Public Domain Making history with...

Recent News



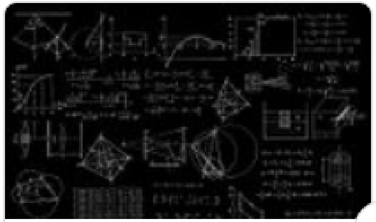
Making History With 42 Digits: Scientists Solve Long-Known Problem In Mathematics

NDTV | 17h

Elusive Ninth Dedekind Number Discovered: Unlocking a Decades-Old Mystery of Mathematics

Scientists from the Universities of Paderborn and Leuven solve long-known problem in mathematics. Making histo...

SciTech Daily | 2d



Ninth Dedekind number discovered: Scientists solve long-known problem in mathematics

Phys.org | 10d

More News →

Are these links helpful? Yes No

https://scitechdaily.com › elusive-ninth-dedekind-number-discovered-unlocking-a-decades-old-...

Elusive Ninth Dedekind Number Discovered: Unlocking a Decades-O...

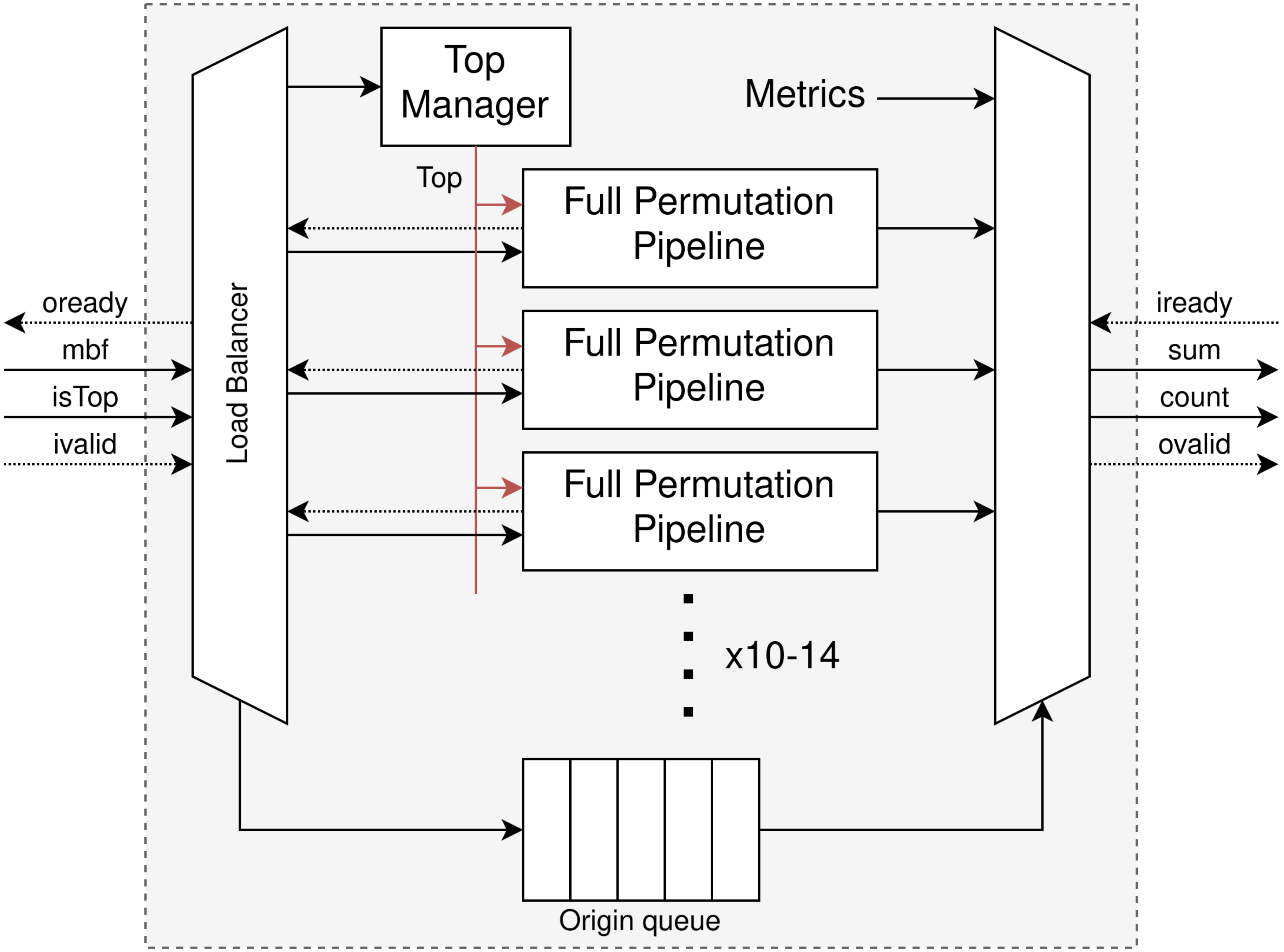
2 days ago · The exact **number**, previously thought uncomputable due to its size, is 286386577668298411128469151667598498812366. Scientists from the Universities of Paderborn and Leuven solve long-known problem in mathematics.

https://interestingengineering.com › innovation › supercomputer-ninth-dedekind-number-probl...

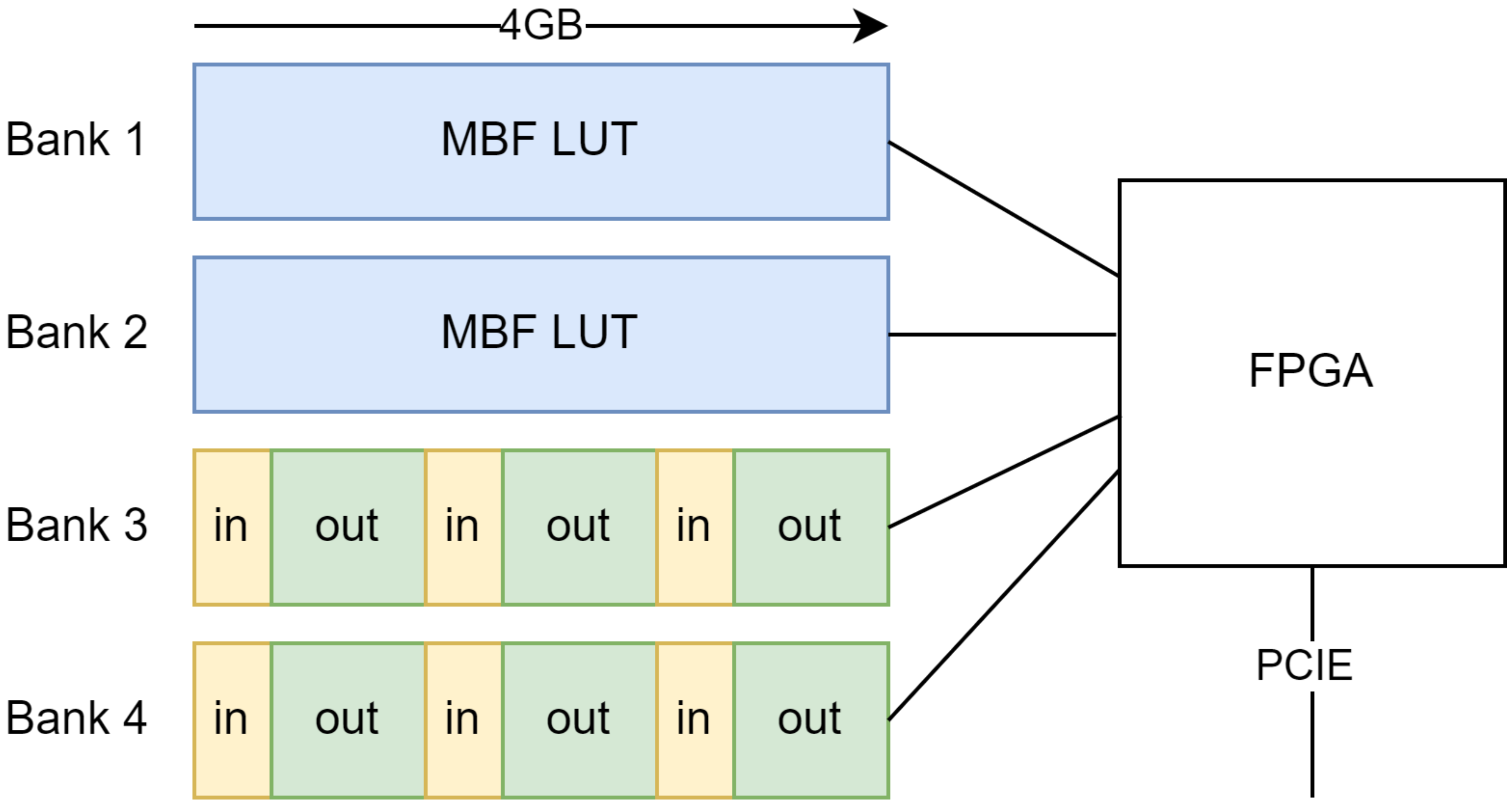
Decades-old math problem solved: The ninth Dedekind number reve...

27 Jun 2023 · **Dedekind numbers** are a rapidly growing series of integers. They are closely related to monotone functions, which are mathematical functions that take a binary input (0 or 1) and produce a binary...

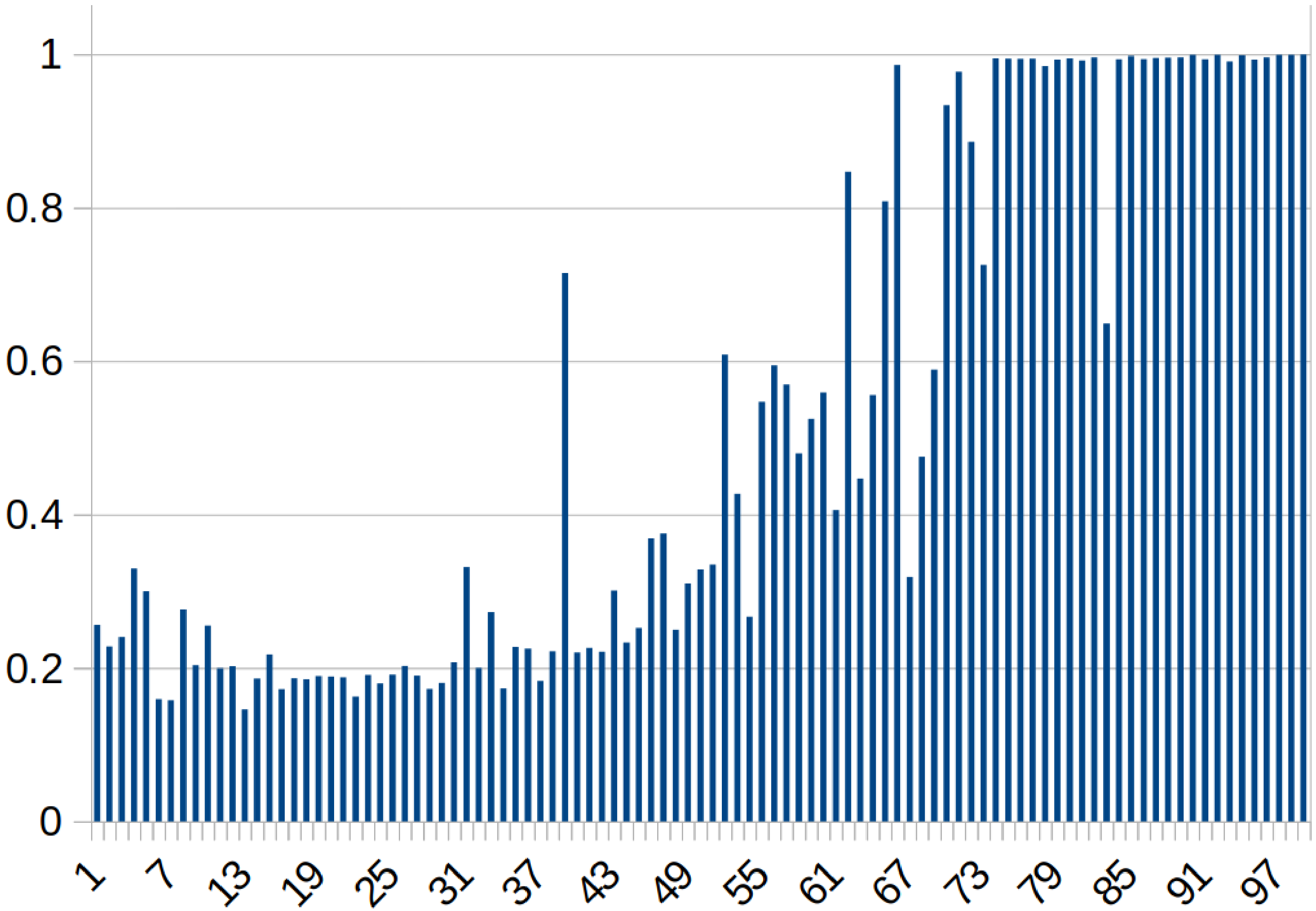
Appendix


Top Manager
Metrics
Top
Full Permutation Pipeline
Full Permutation Pipeline
Full Permutation Pipeline
x10-14
Load Balancer
oready
mbf
isTop
ivalid
iready
sum
count
ovalid
Origin queue

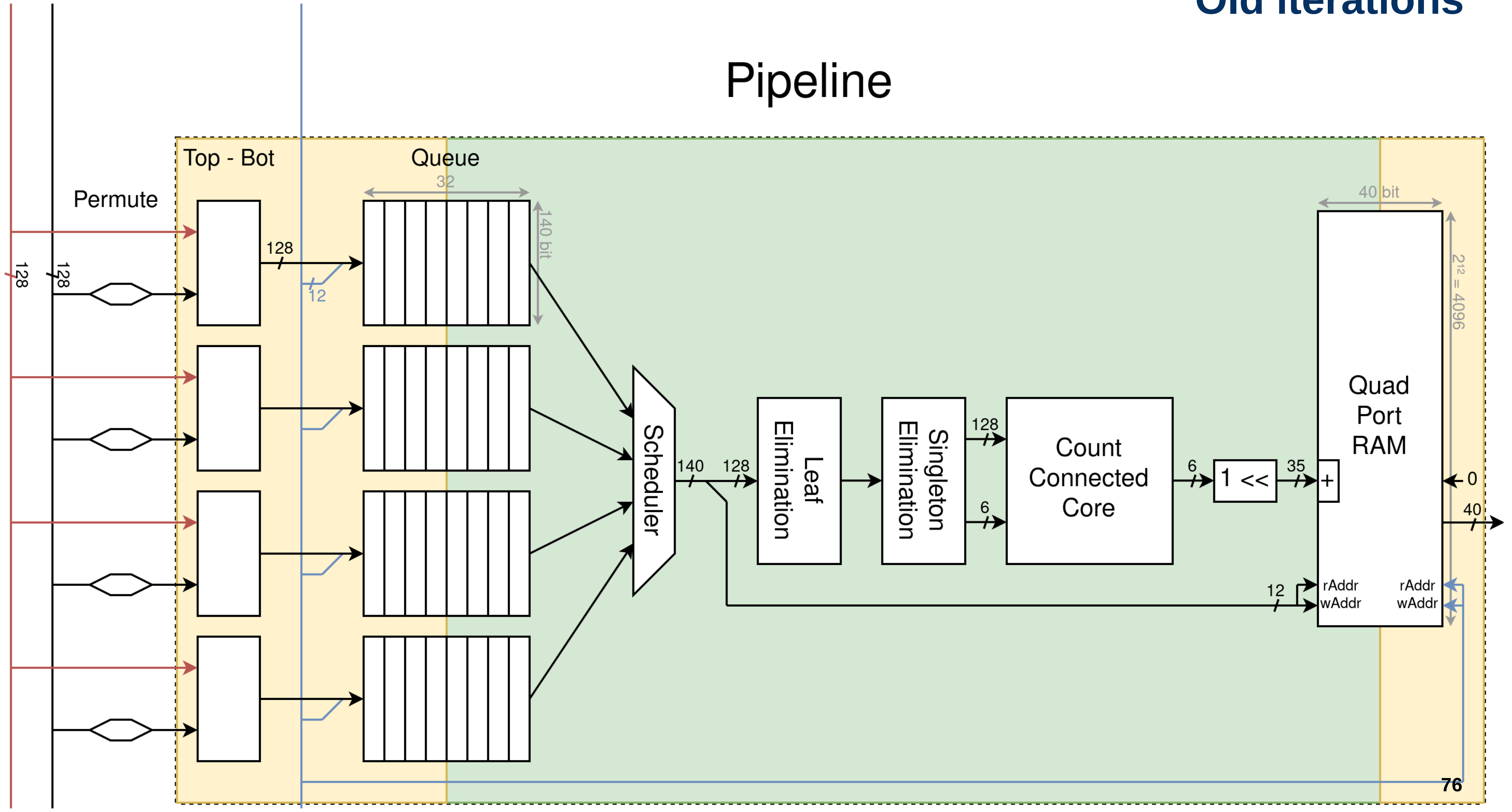
FPGA Card Memory



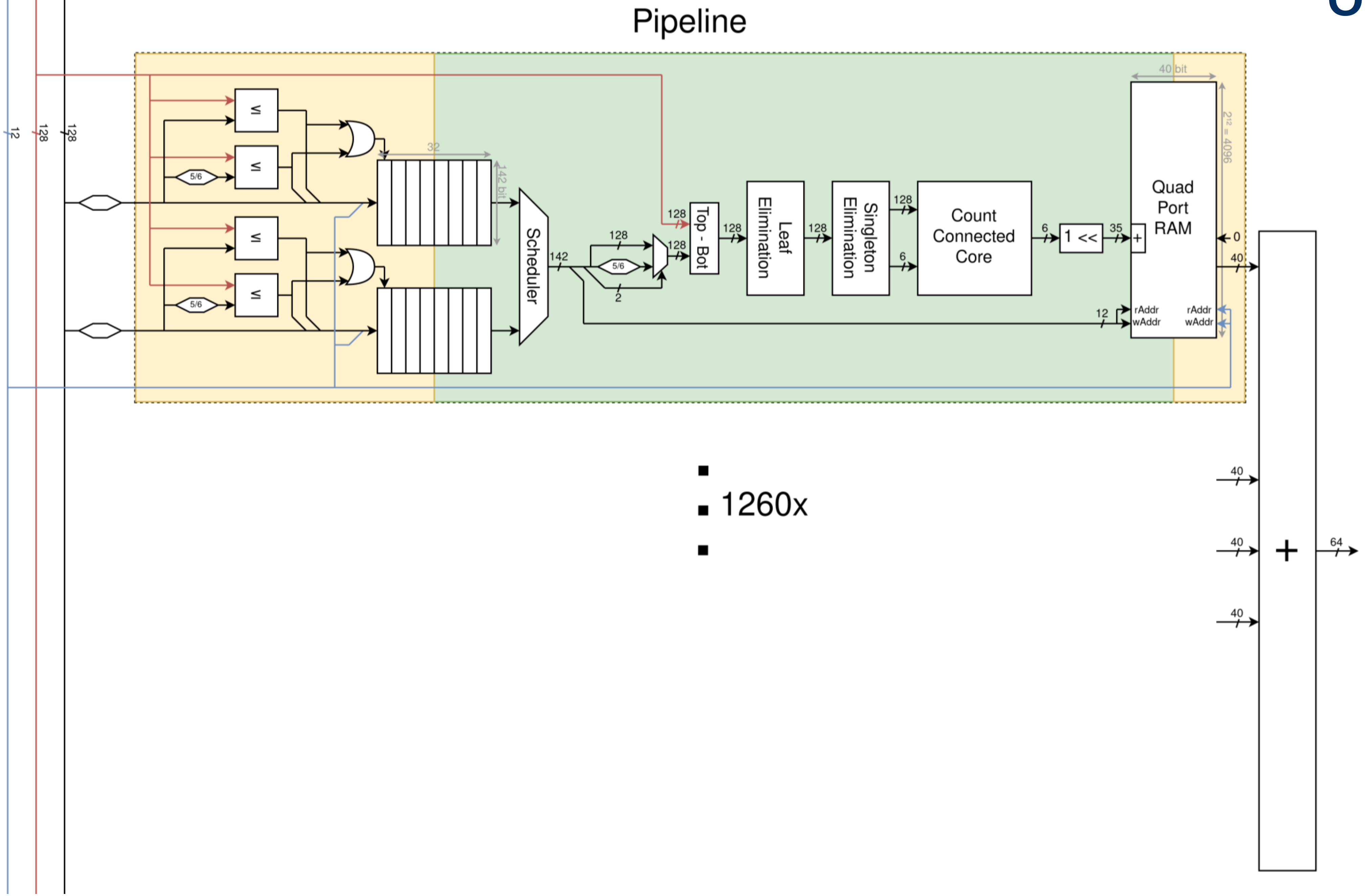
Efficiency Distribution



Pipeline


Top - Bot
Queue
Permute
32
140 bit
128
128
128
12
Scheduler
140
128
Leaf Elimination
Singleton Elimination
128
6
Count Connected Core
6
1 <<
35
+
Quad Port RAM
40 bit
2^12 = 4096
0
40
12
rAddr
wAddr
rAddr
wAddr

Old iterations



-
- 1260x
-