https://hao-ai-lab.github.io/dsc204a-f25/

DSC 204A: Scalable Data Systems
Fall 2025

Staff
Instructor: Hao Zhang
TAs: Mingjia Huo, Yuxuan Zhang

@haozhangml @haoailab
haozhang@ucsd.edu

Where We Are

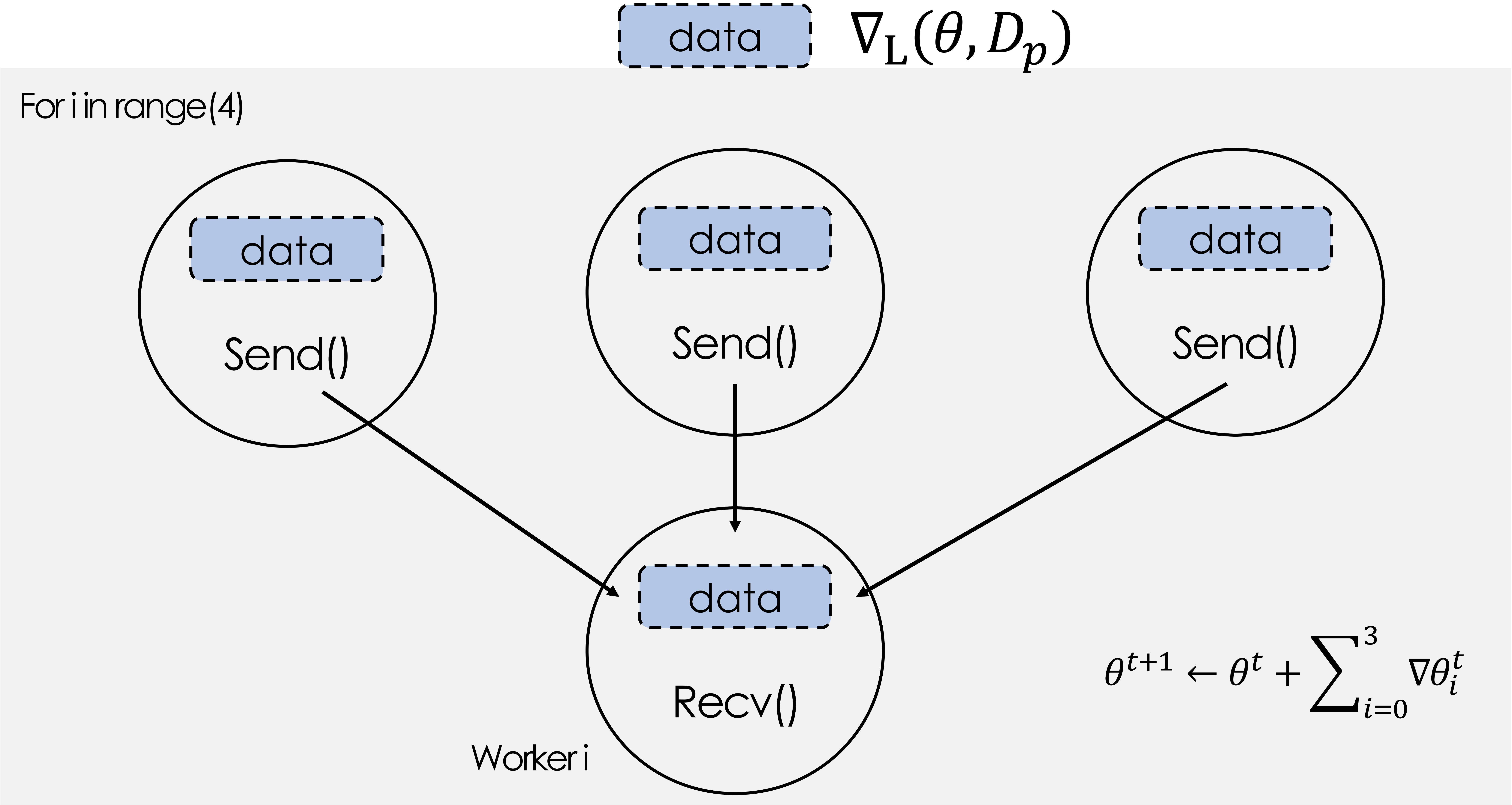
Machine Learning Systems

Big Data

Cloud — 2000 - 2016

Foundations of Data Systems — 1980 - 2000

Problem: We need **All-**Reduce



Program This? Will be in PA2!

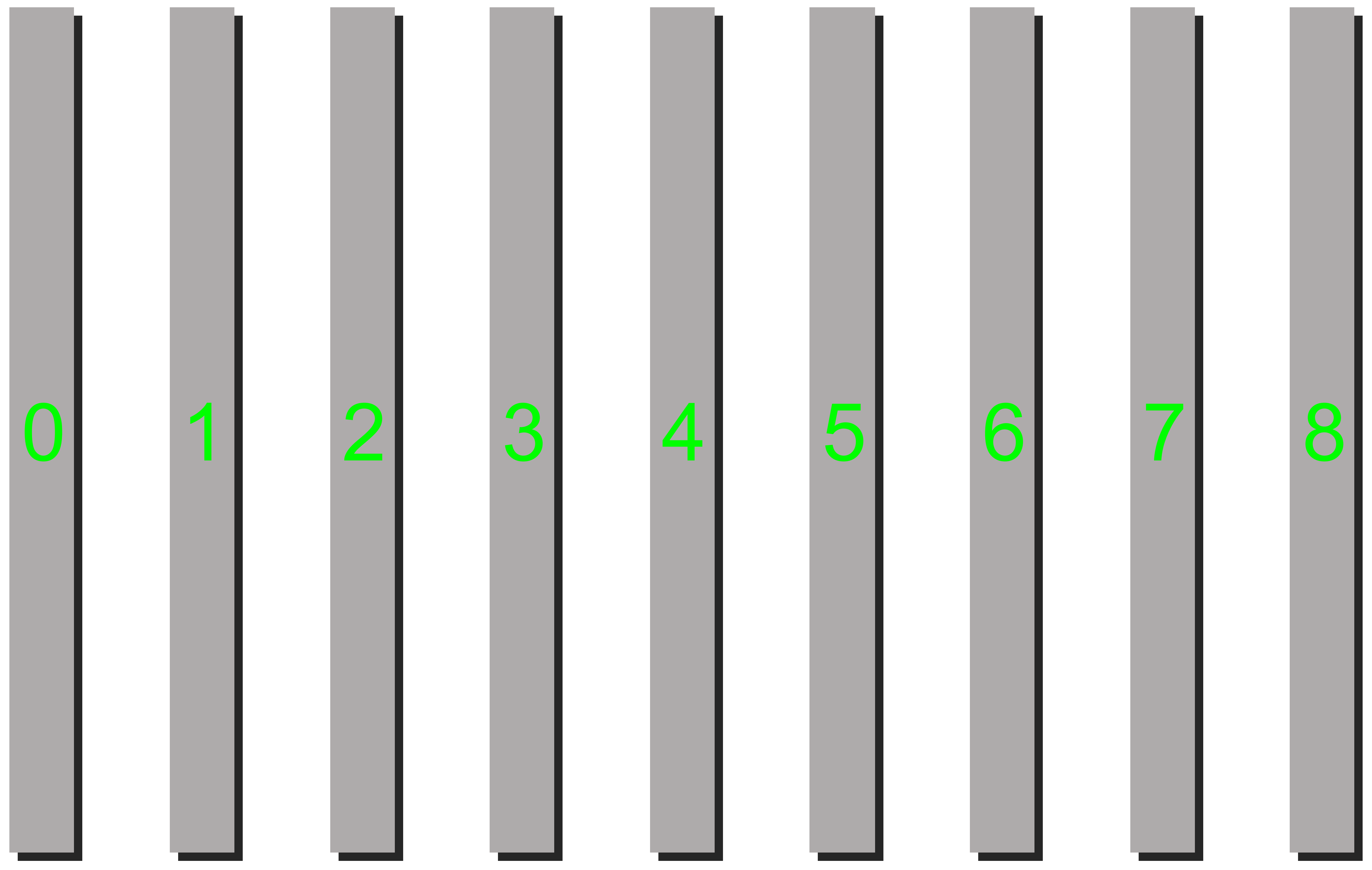
Performance

- Message size over networks:
  - Sum: 3N
  - Send Sum back: 3N
  - = 6N
- Can we do better?
  - Hint: we cannot do better than 3N

Why Collective Communication?

- Programming Convenience
  - Use a set of well-defined communication primitives to express complex communication patterns
- Unification and Performance
  - Since they are well defined and well structured, we can optimize them to the extreme
- ML Systems 🧡 Collective communication

Make it Formal

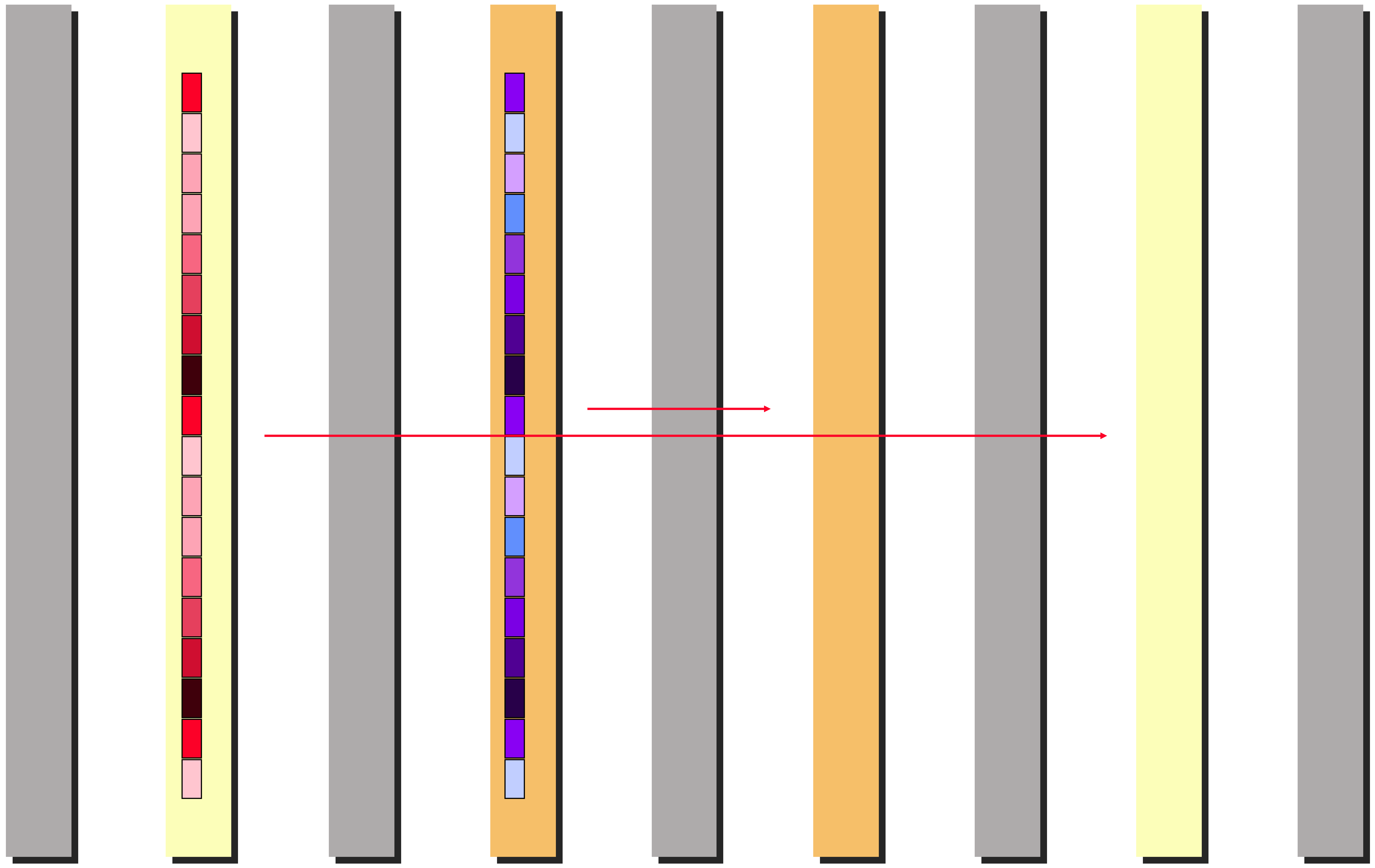


- A 1D **Mesh** of workers (or devices, or nodes)

Model of Parallel Computation

- a node can send directly to any other node (maybe not true)
- a node can simultaneously receive and send
- cost of communication
  - sending a message of length *n* between any two nodes

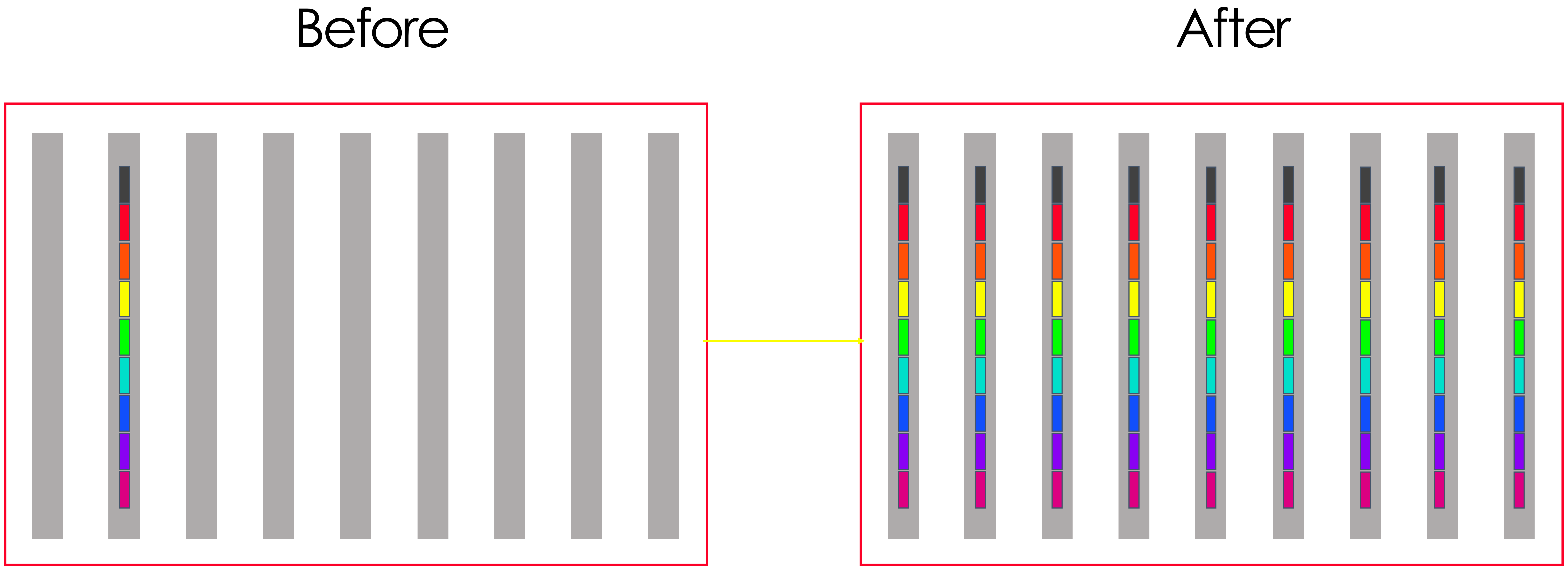
$$\alpha + n\,\beta$$



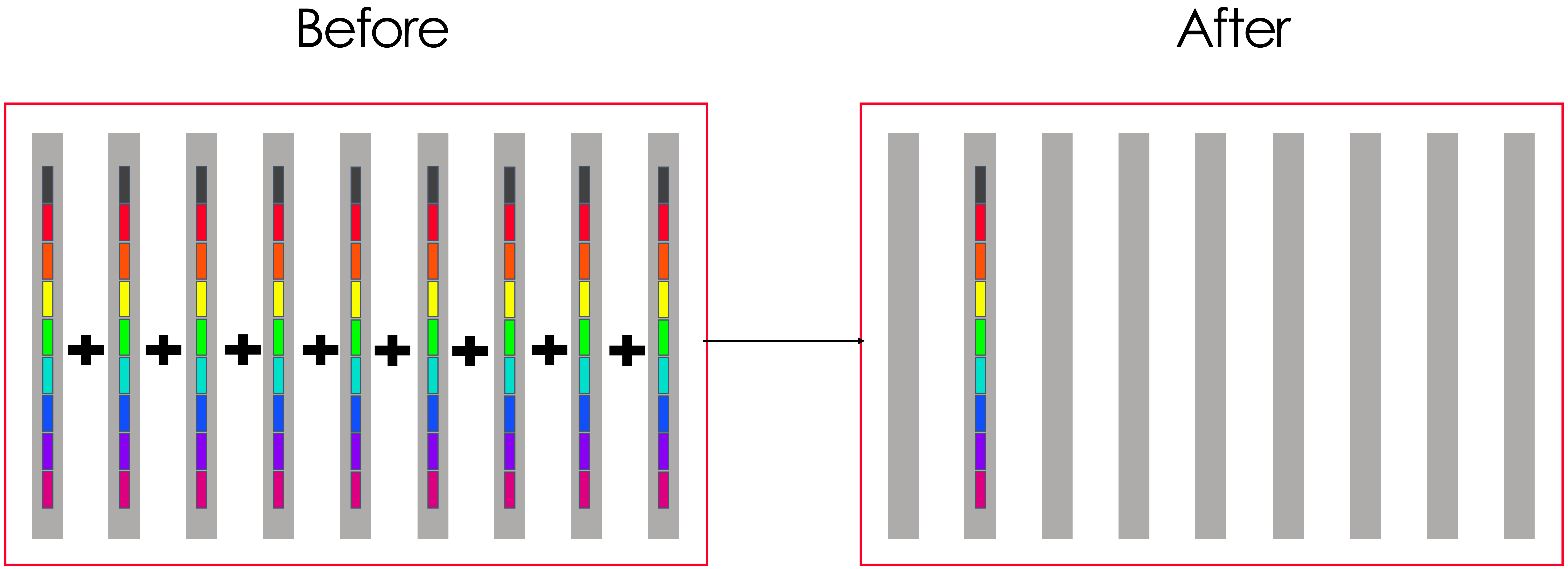
Collective Communications

- Broadcast
- Reduce(-to-one)
- Scatter
- Gather
- Allgather
- Reduce-scatter
- Allreduce
- All-2-All

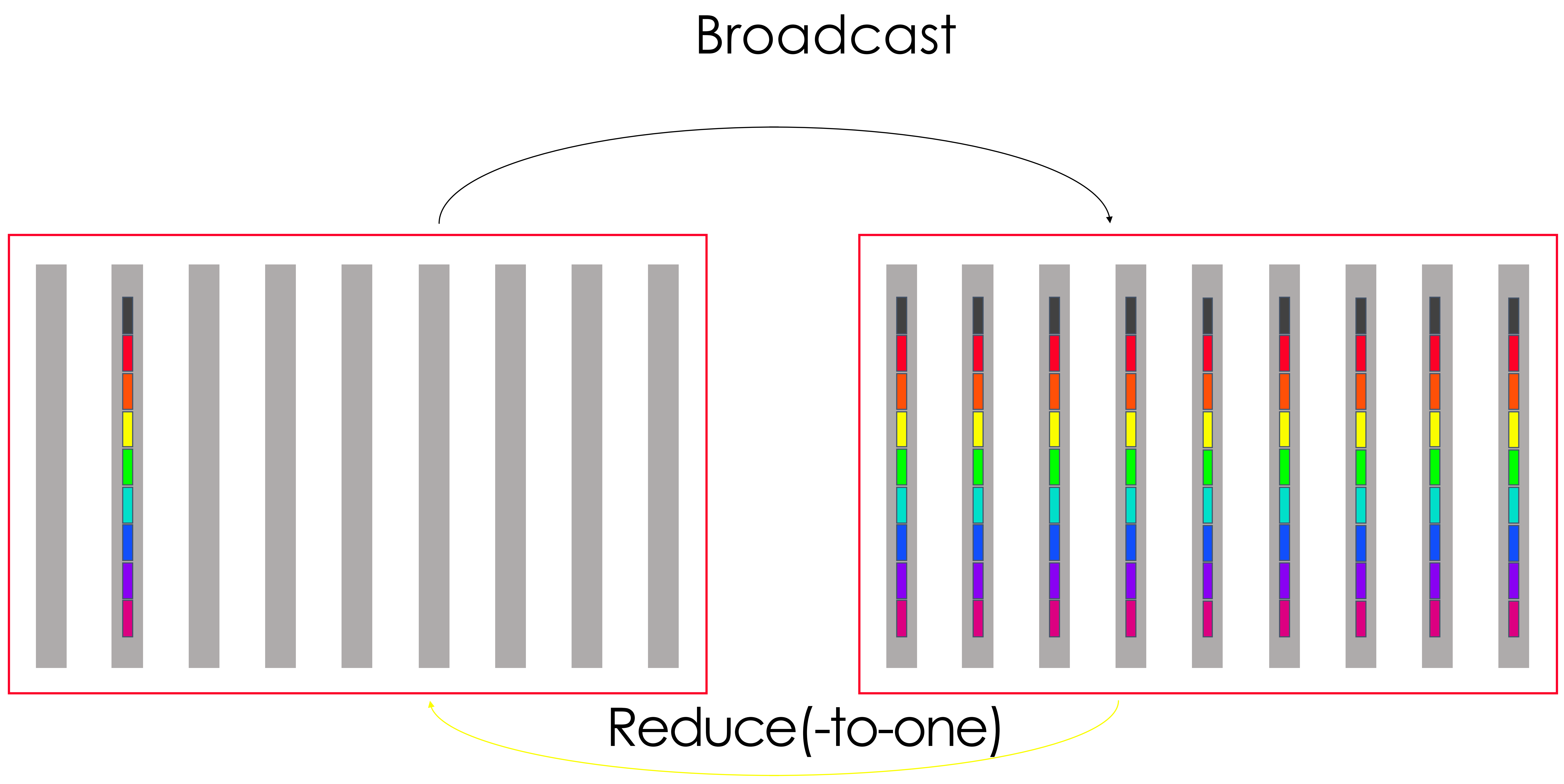
Broadcast



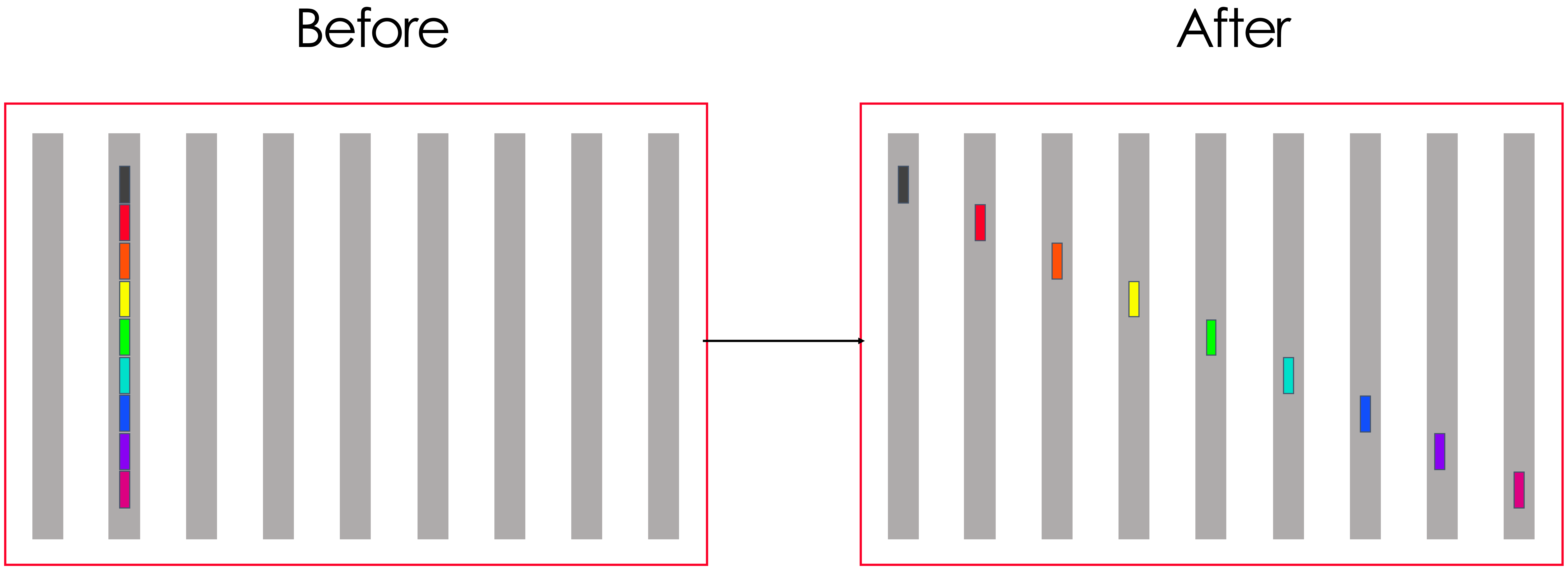
Reduce(-to-one)



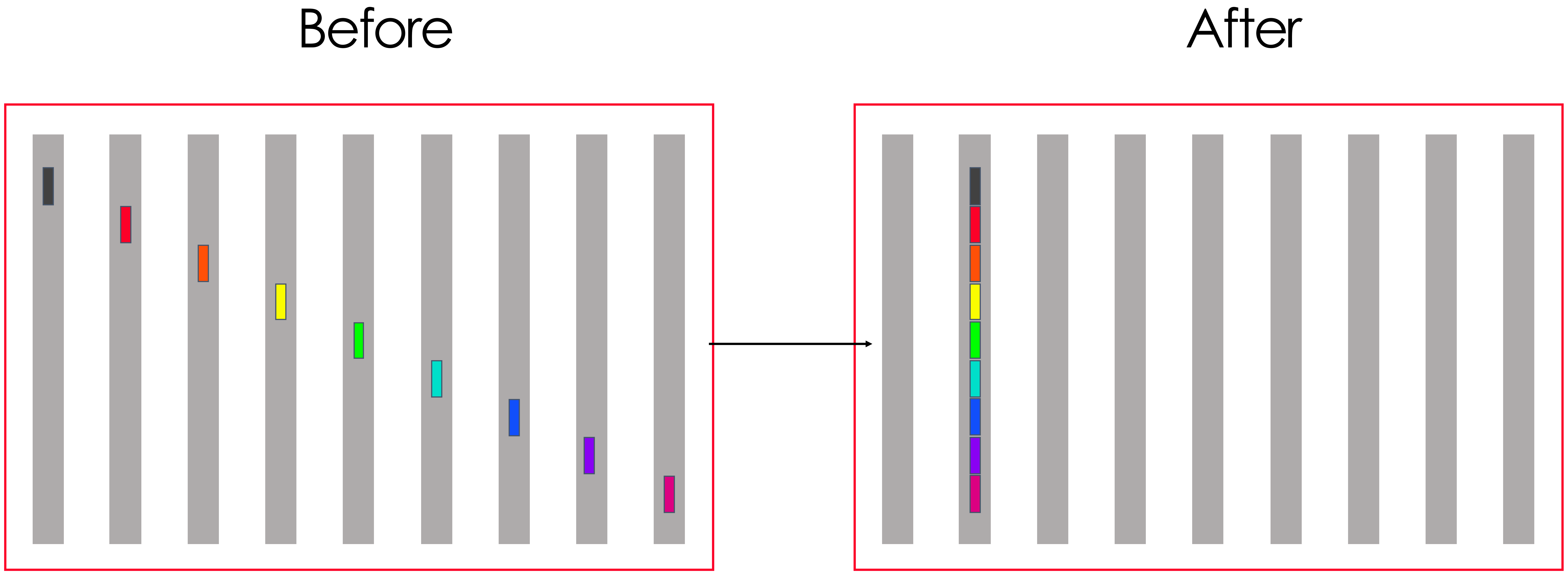
Broadcast/Reduce(-to-one)



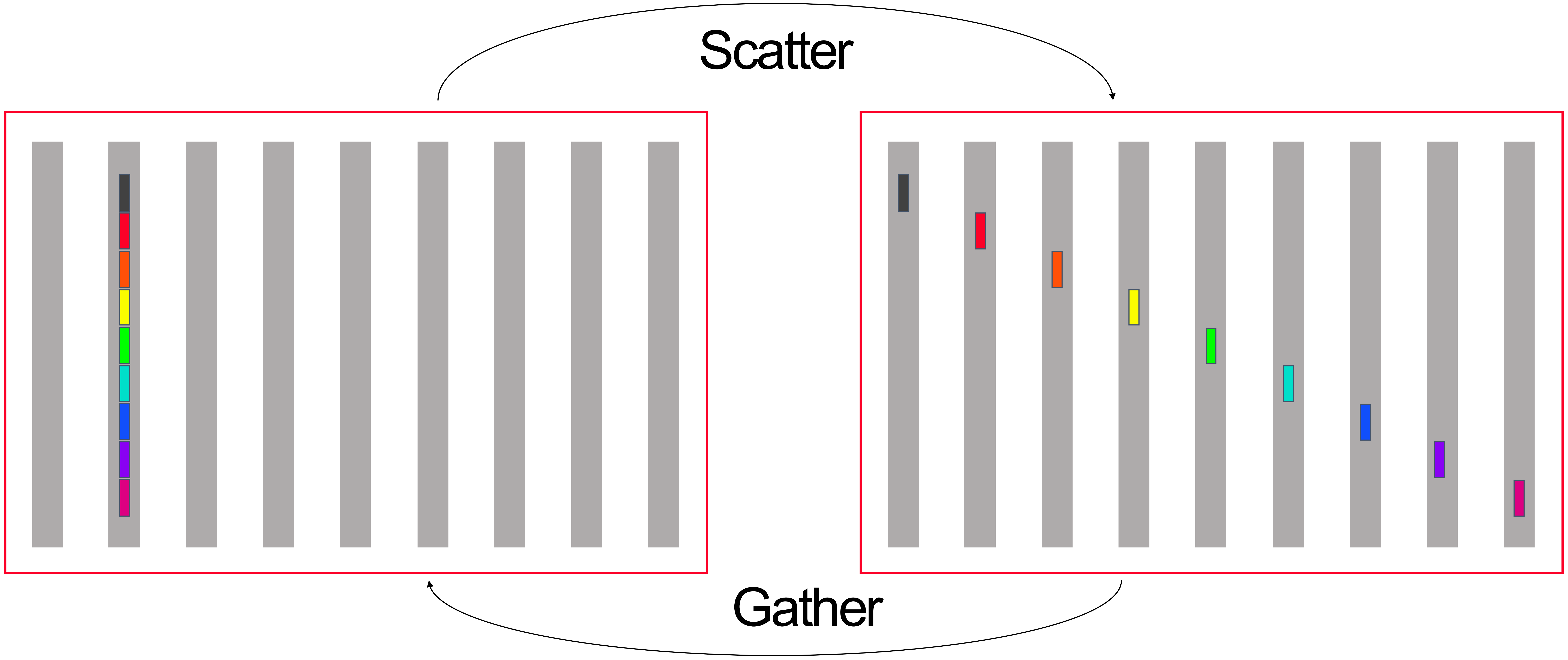
Scatter



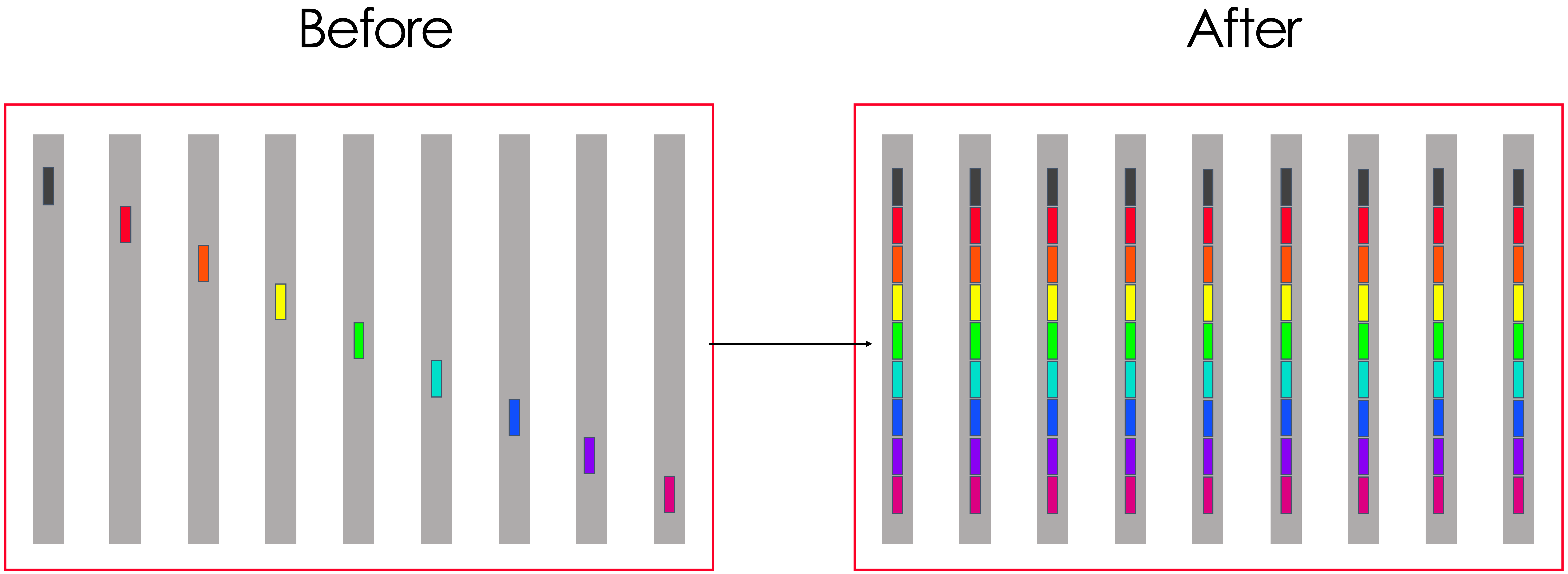
Gather



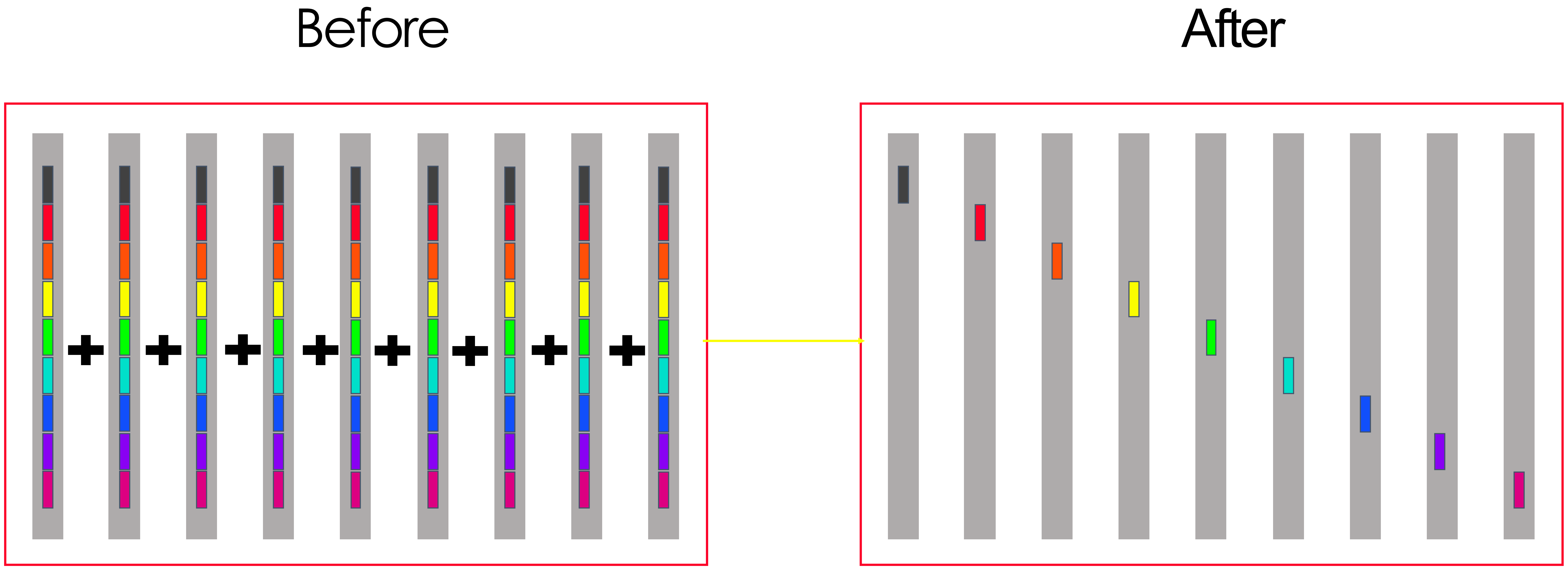
Scatter/Gather



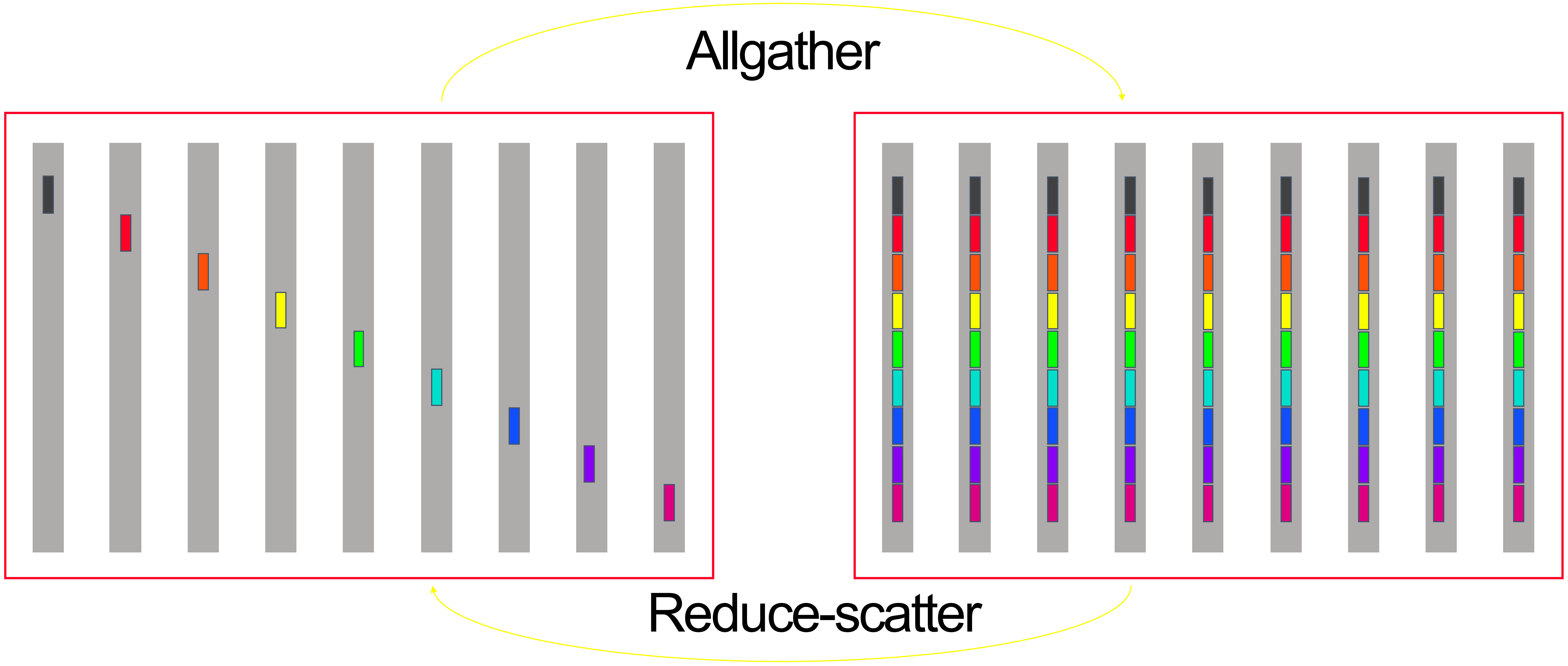
Allgather



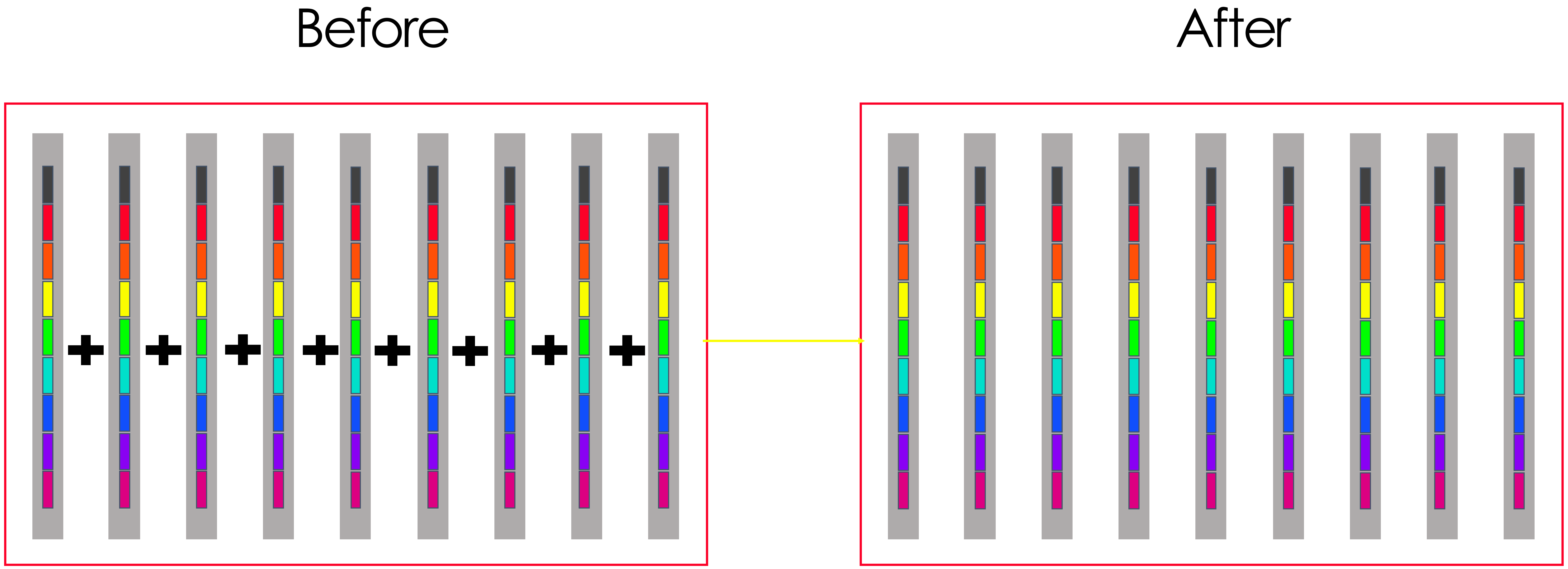
Reduce-scatter



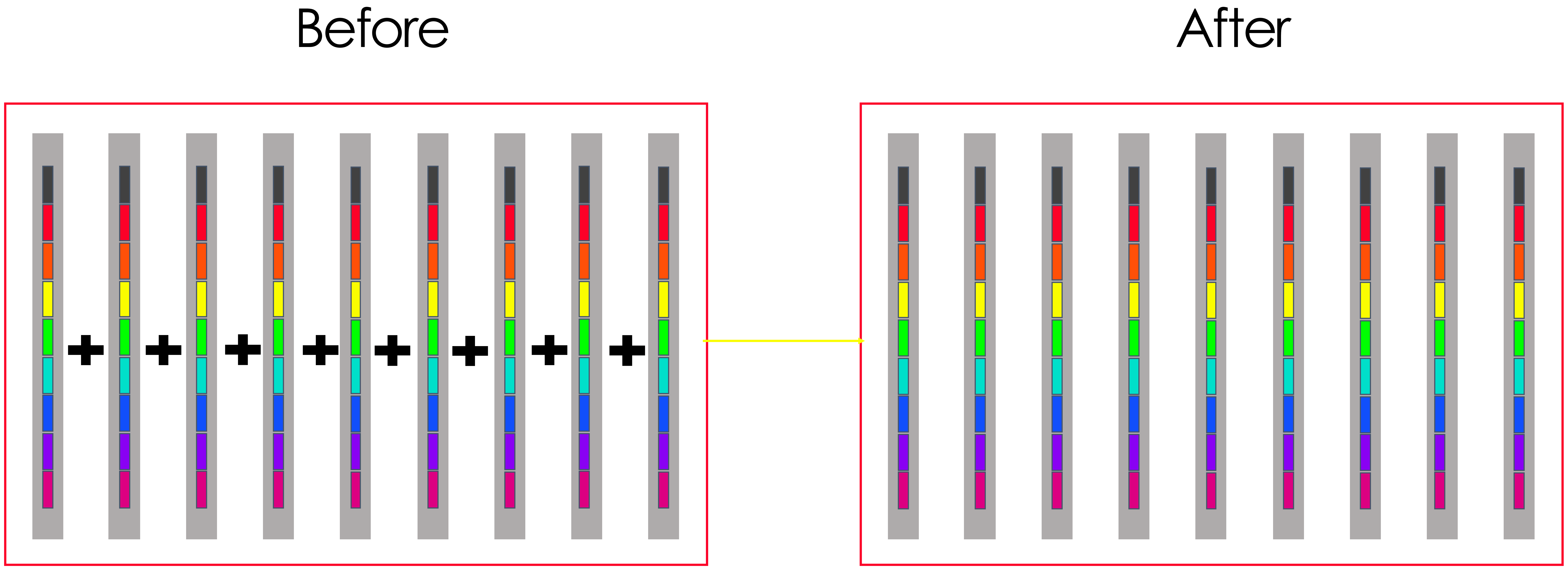
Allgather/Reduce-scatter



Allreduce



All2All



Some Facts

- Collective is much more expensive than P2P
  - Collective can be assembled using many P2P
  - Collective is cheaper than realizing collective using P2P (we'll see)
- Collective is highly optimized in the past 20 years
  - Look out for "X"CCL libraries
    - NCCL, MCCL, OneCCL, UCCL
- Collective is not fault-tolerant
  - A major sources of faults in ML systems

Communication Model: $\alpha\beta$ model

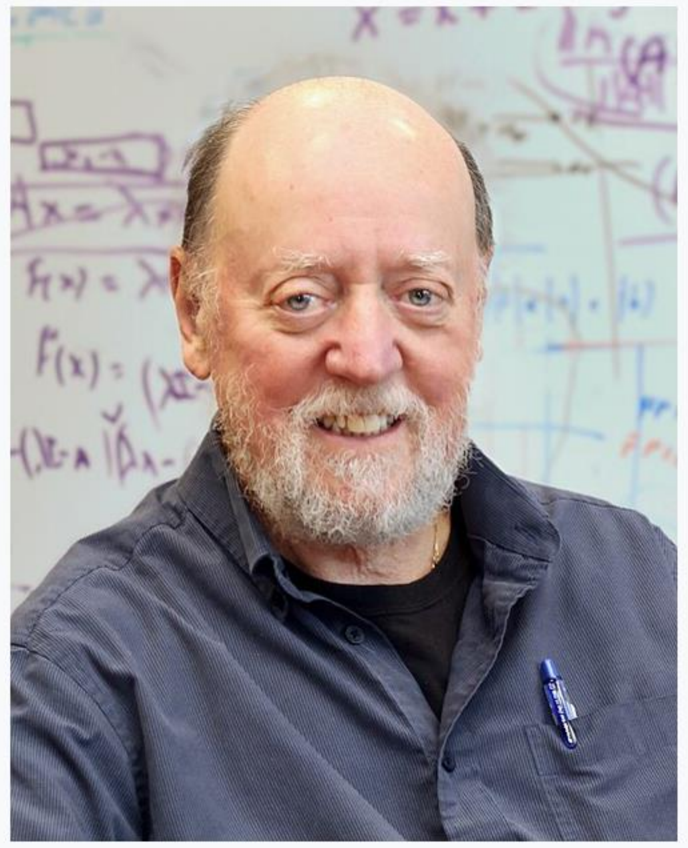
Communication Model: $\alpha + n\,\beta, \beta = \frac{1}{B}$

- Small Message size ($n \to 0$): α dominates, emphasize latency
- Large Message Size ($n \to +\infty$): $n\,\beta$ dominate, emphasize bandwidth utilization

Two Family of Mainstream Algorithms/Implementations

- Small message: Minimum Spanning Tree algorithm
  - Emphasize **low latency**

- Large Message: Ring algorithm
  - Emphasize **bandwidth utilization**



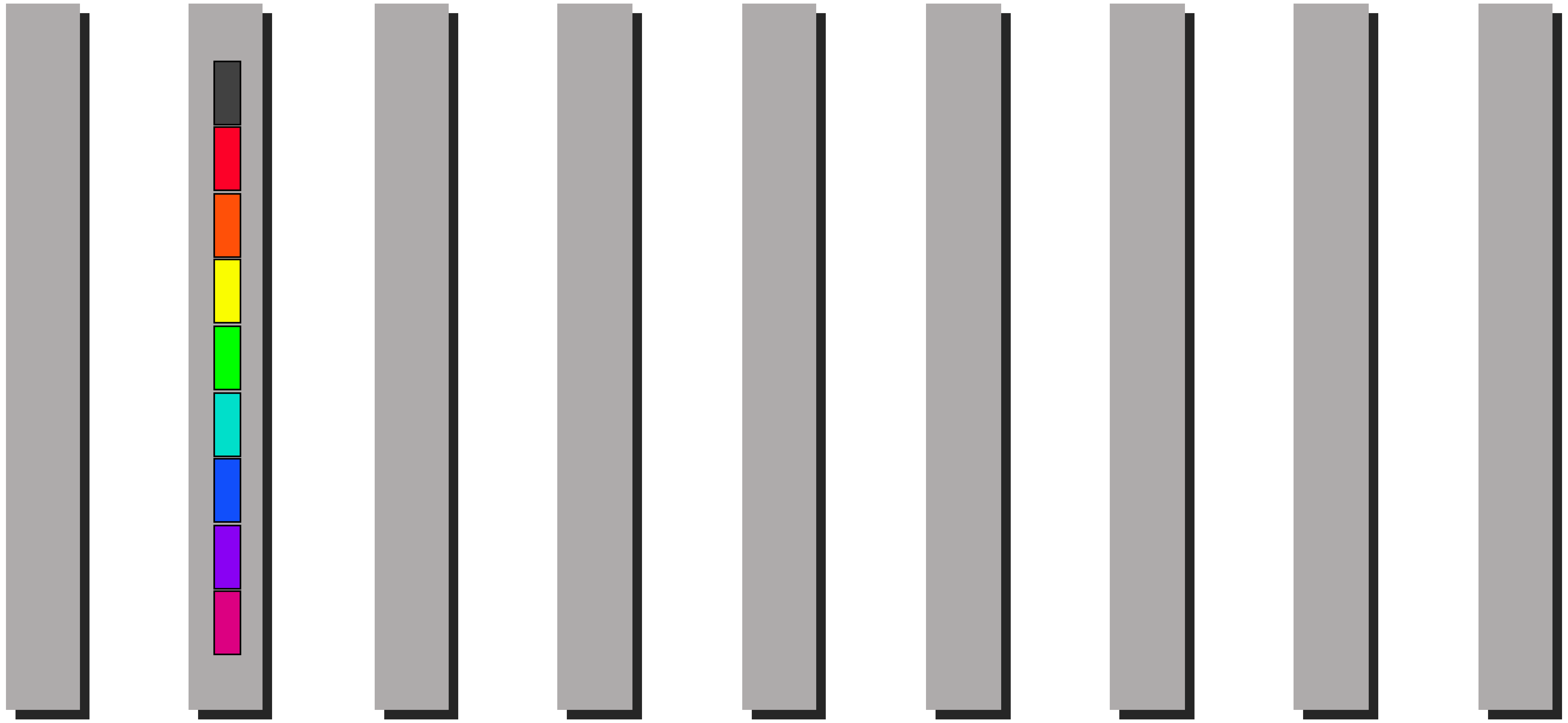
Dongarra in 2022

- There are 50+ different algorithms developed in the past 50 years by a community called “High-performance computing”
  - 2021 Turing award

General principles: Low Latency

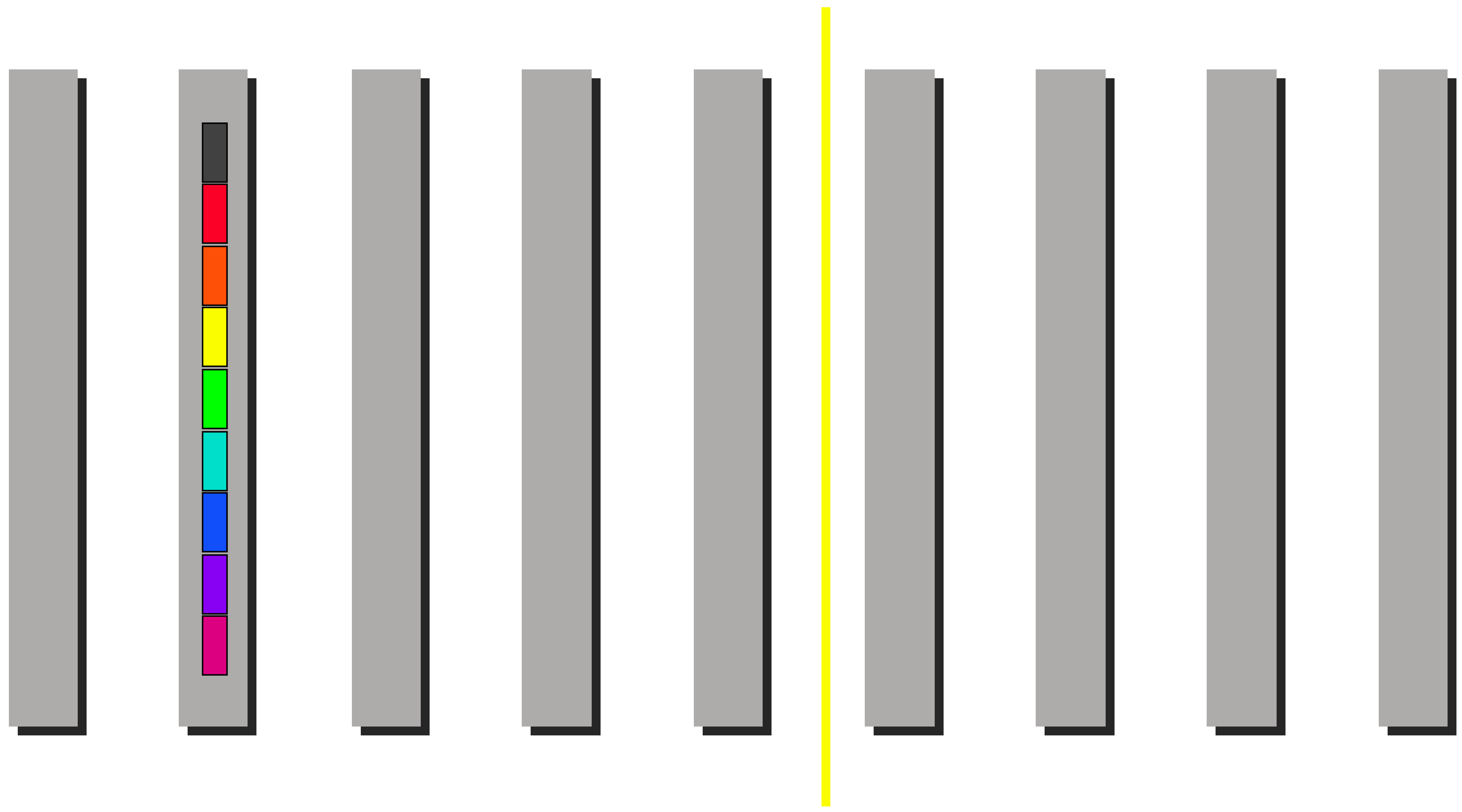
- Minimize the number of rounds needed for communication
- Minimal-spanning tree algorithm

General principles



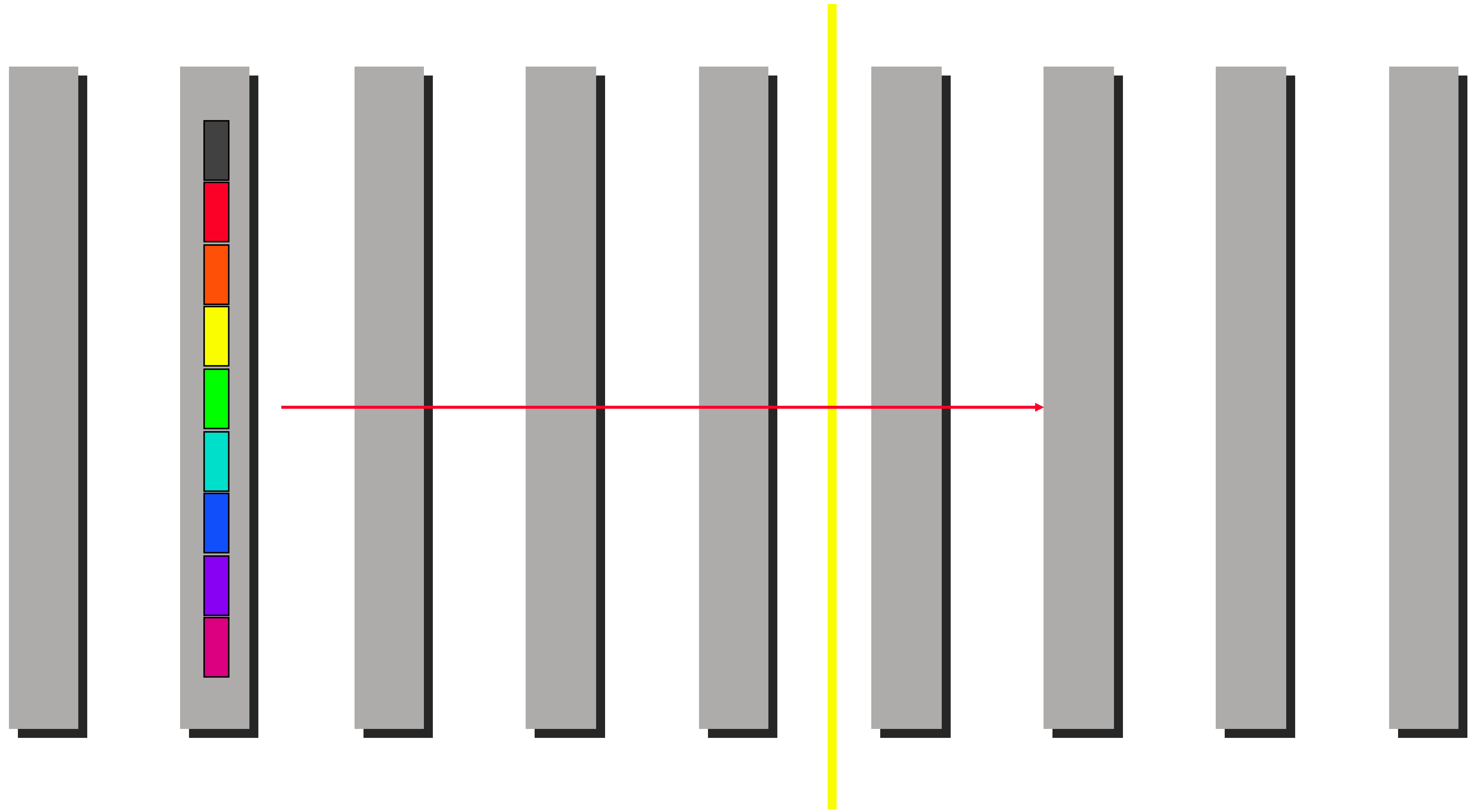
- message starts on one processor

General principles



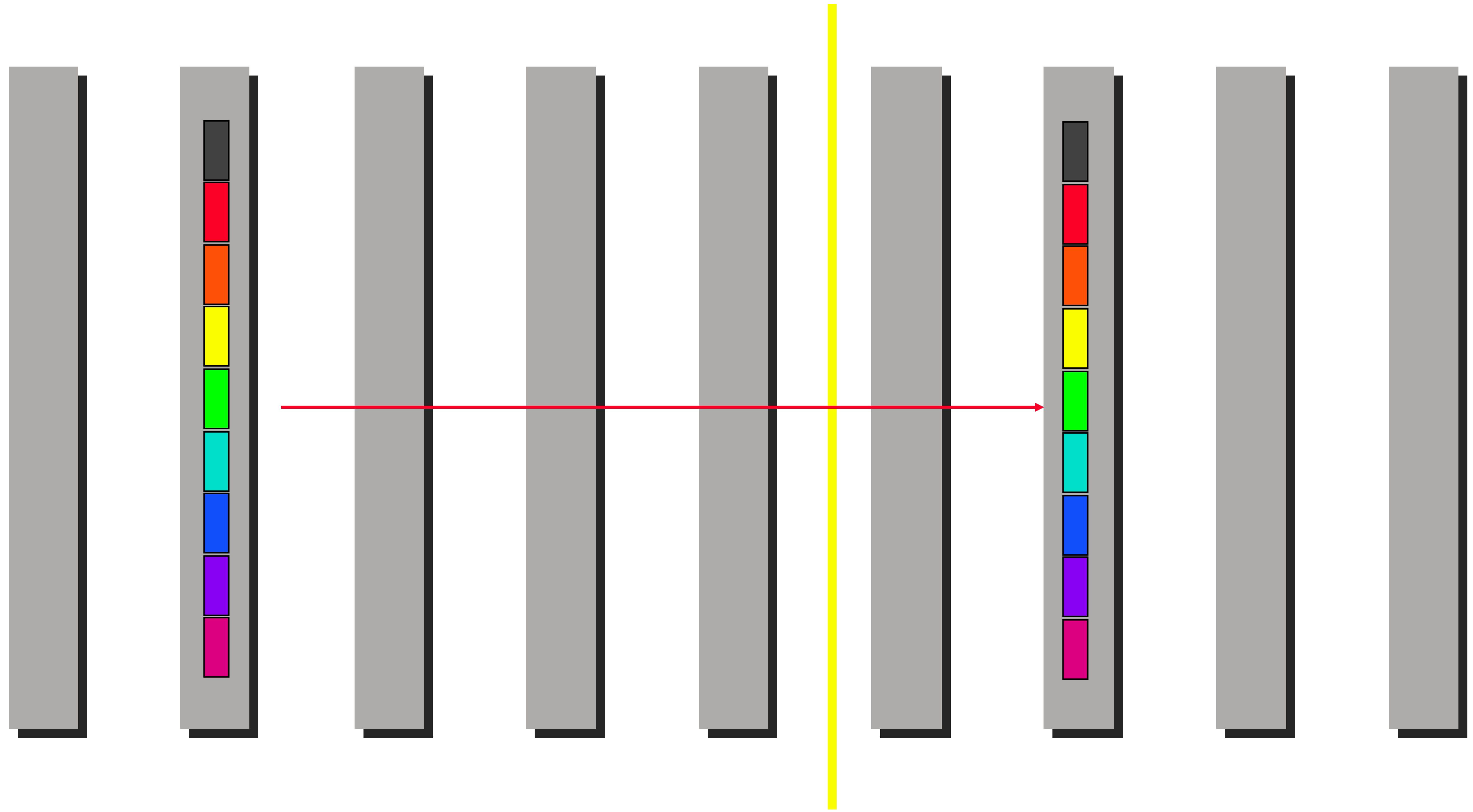
- divide logical linear array in half

General principles



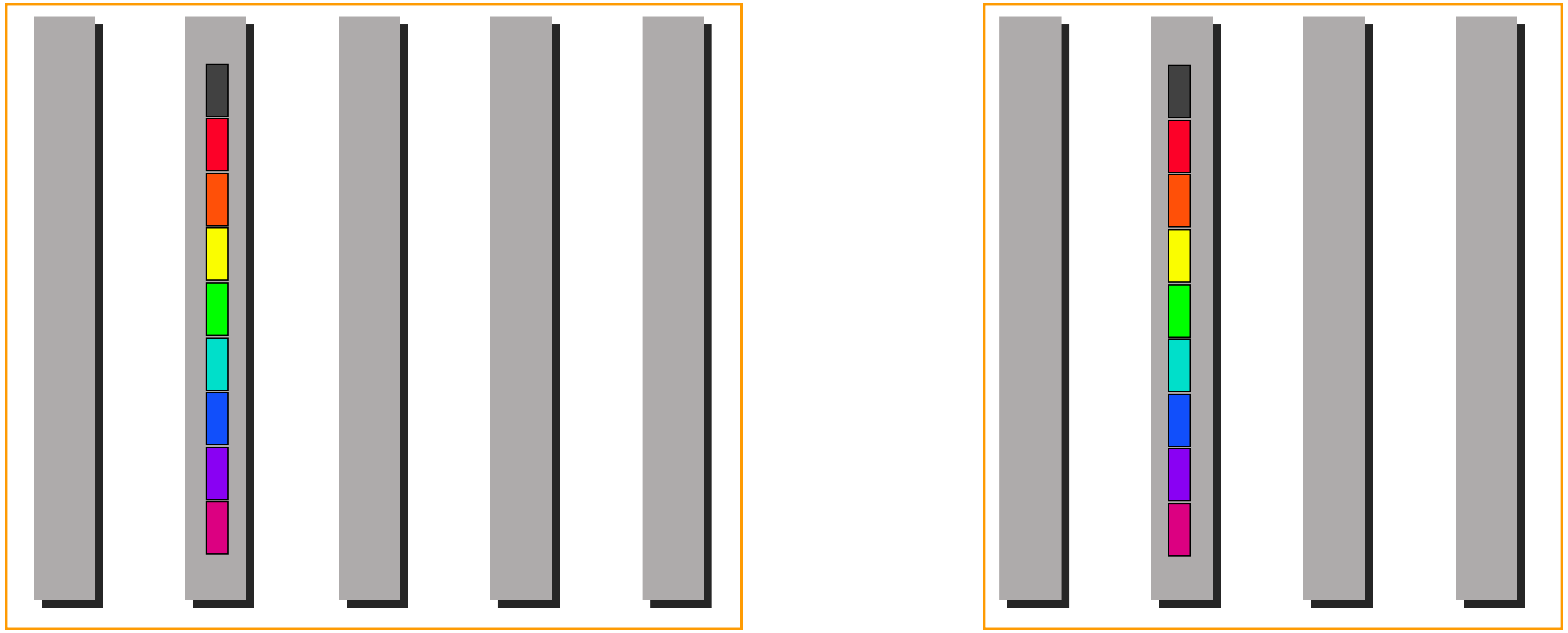
- send message to the half of the network that does not contain the current node (root) that holds the message

General principles



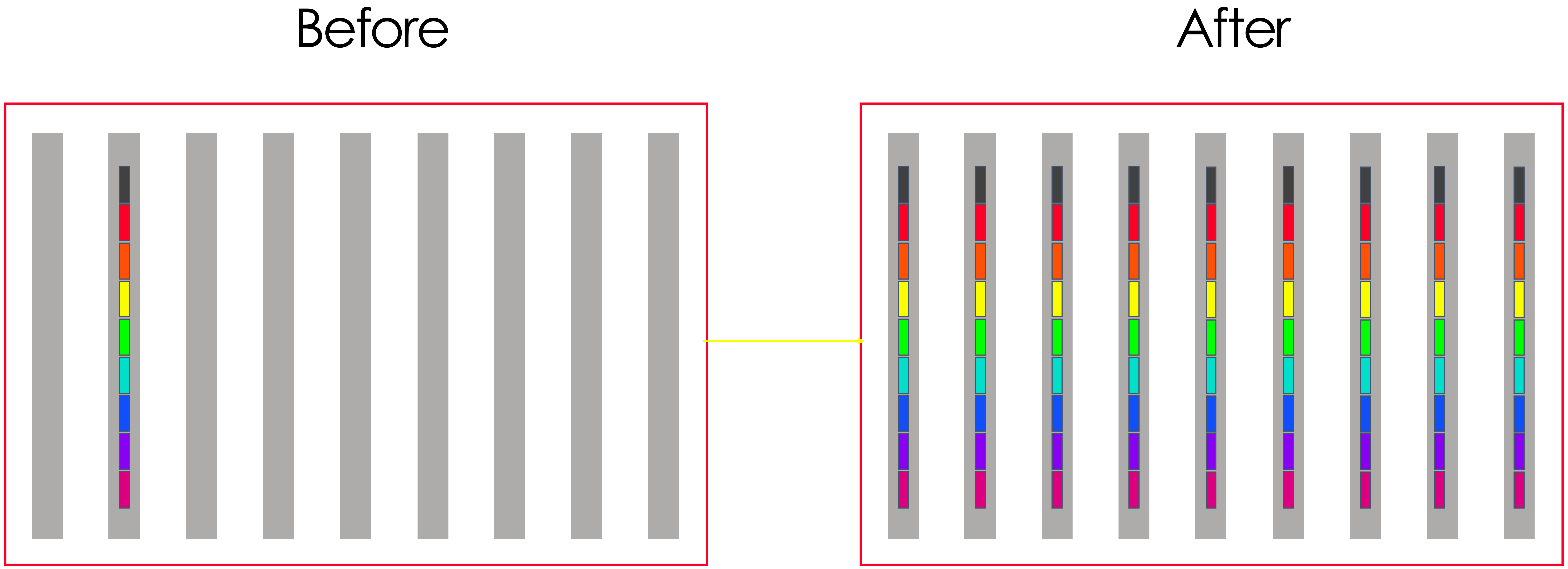
- send message to the half of the network that does not contain the current node (root) that holds the message

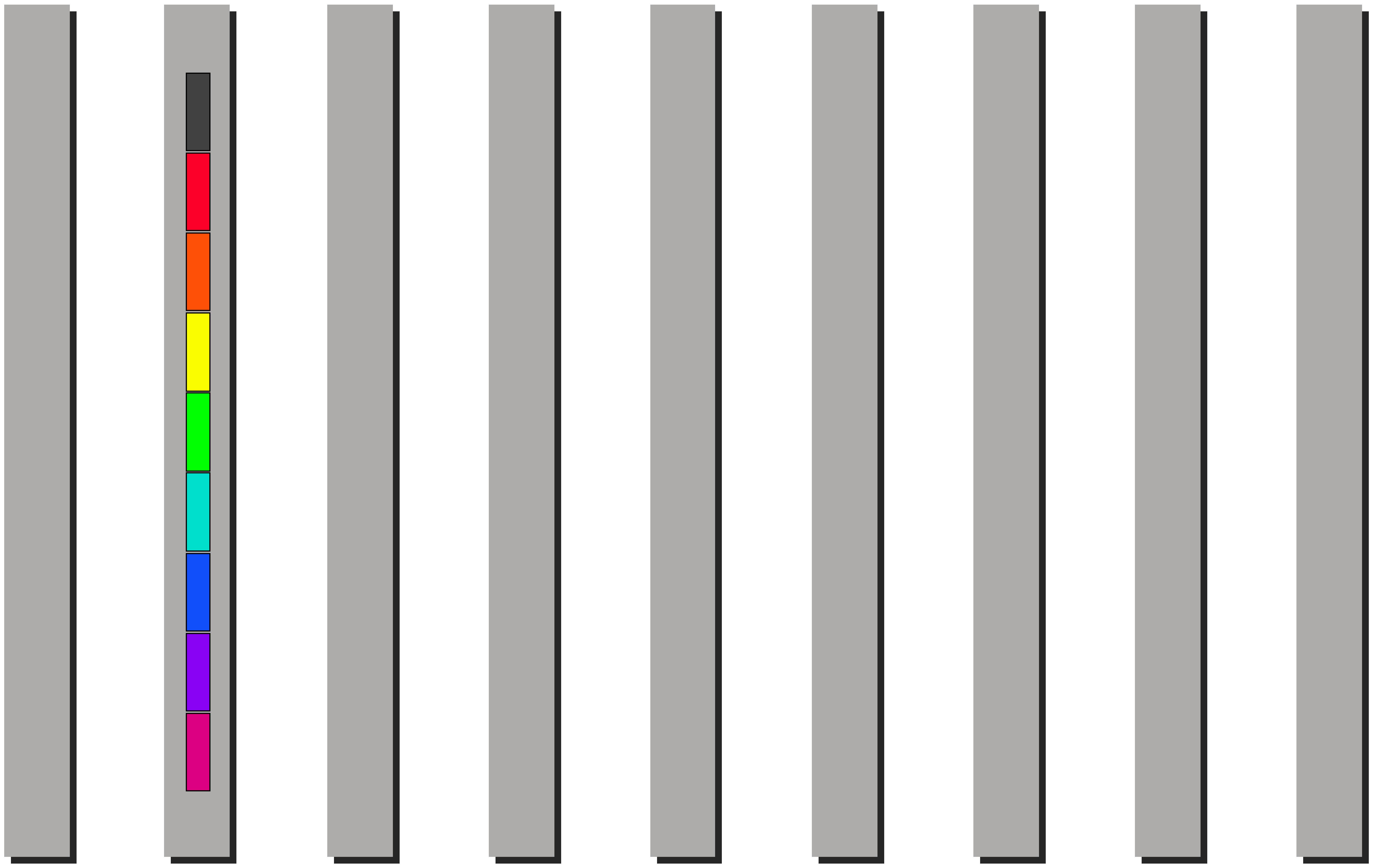
General principles

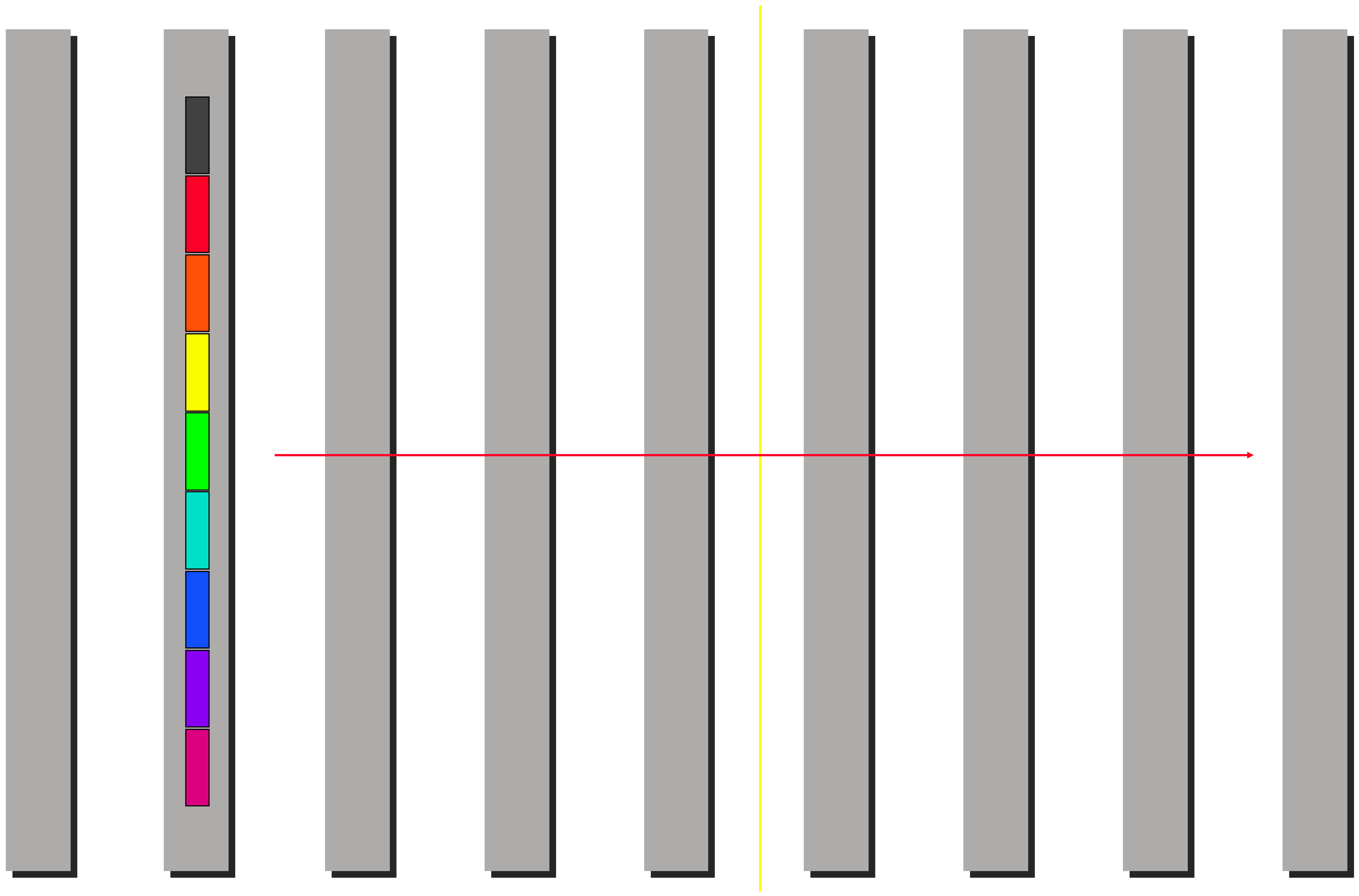


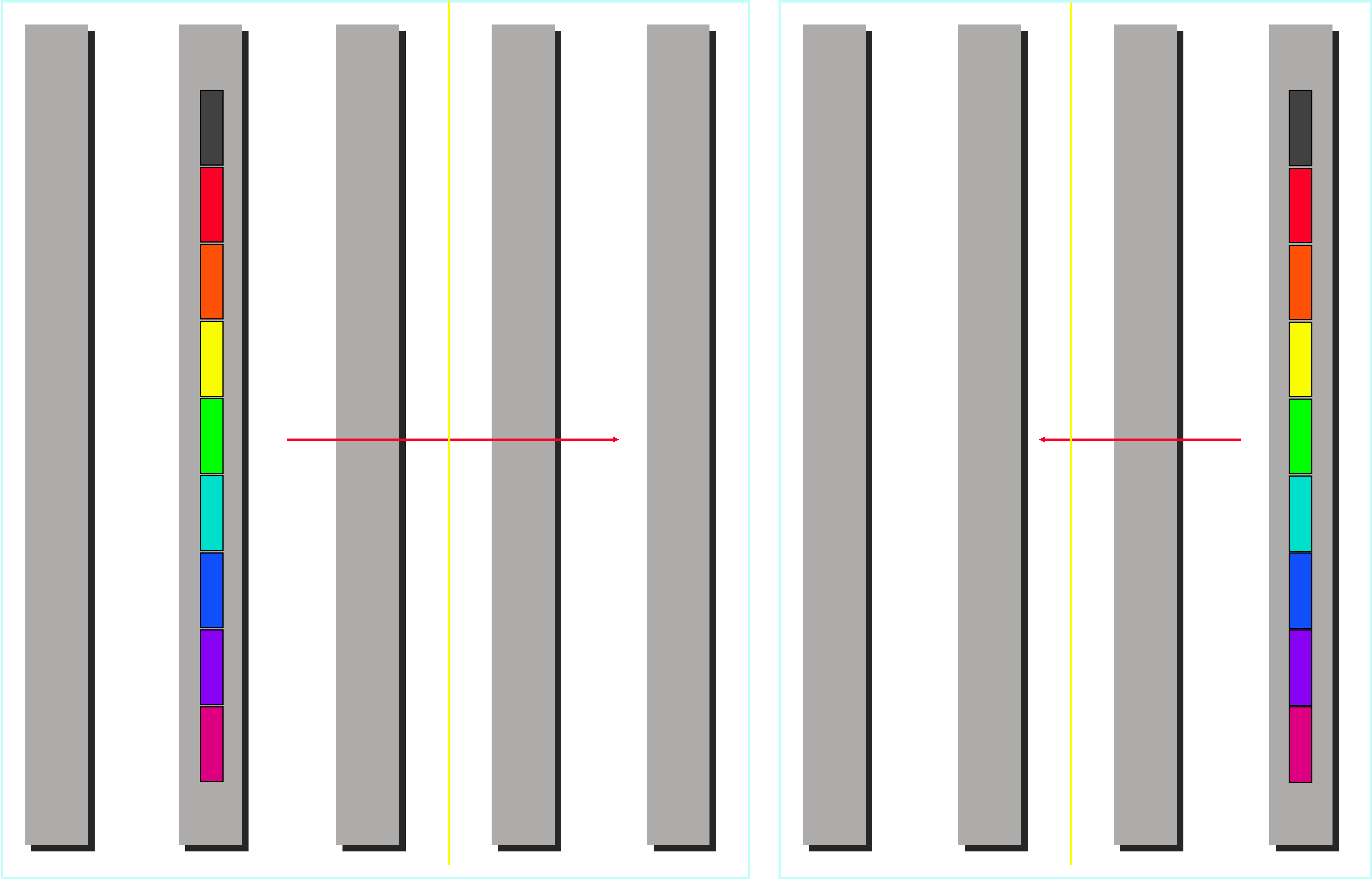
- continue recursively in each of the two halves

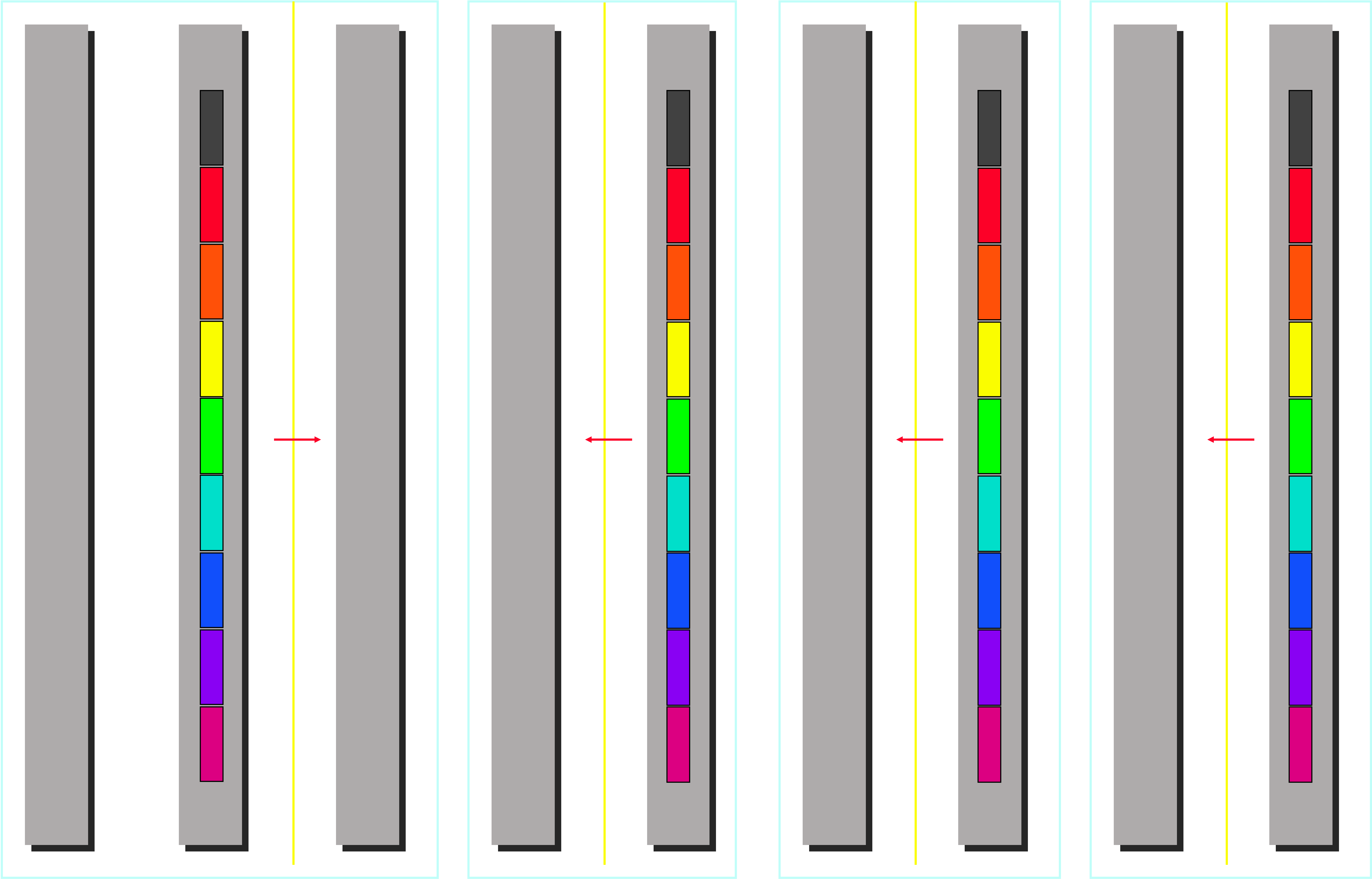
Broadcast

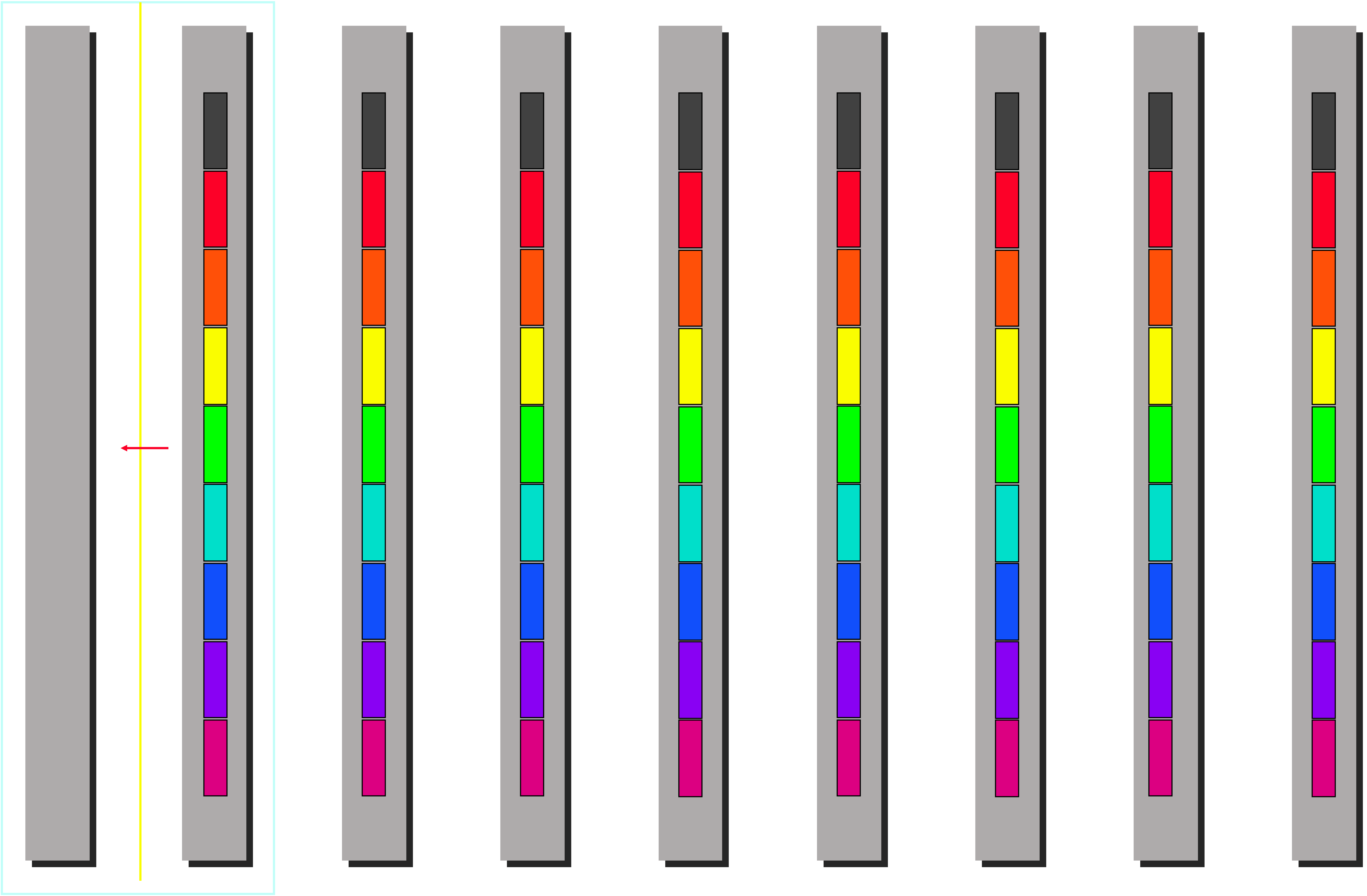


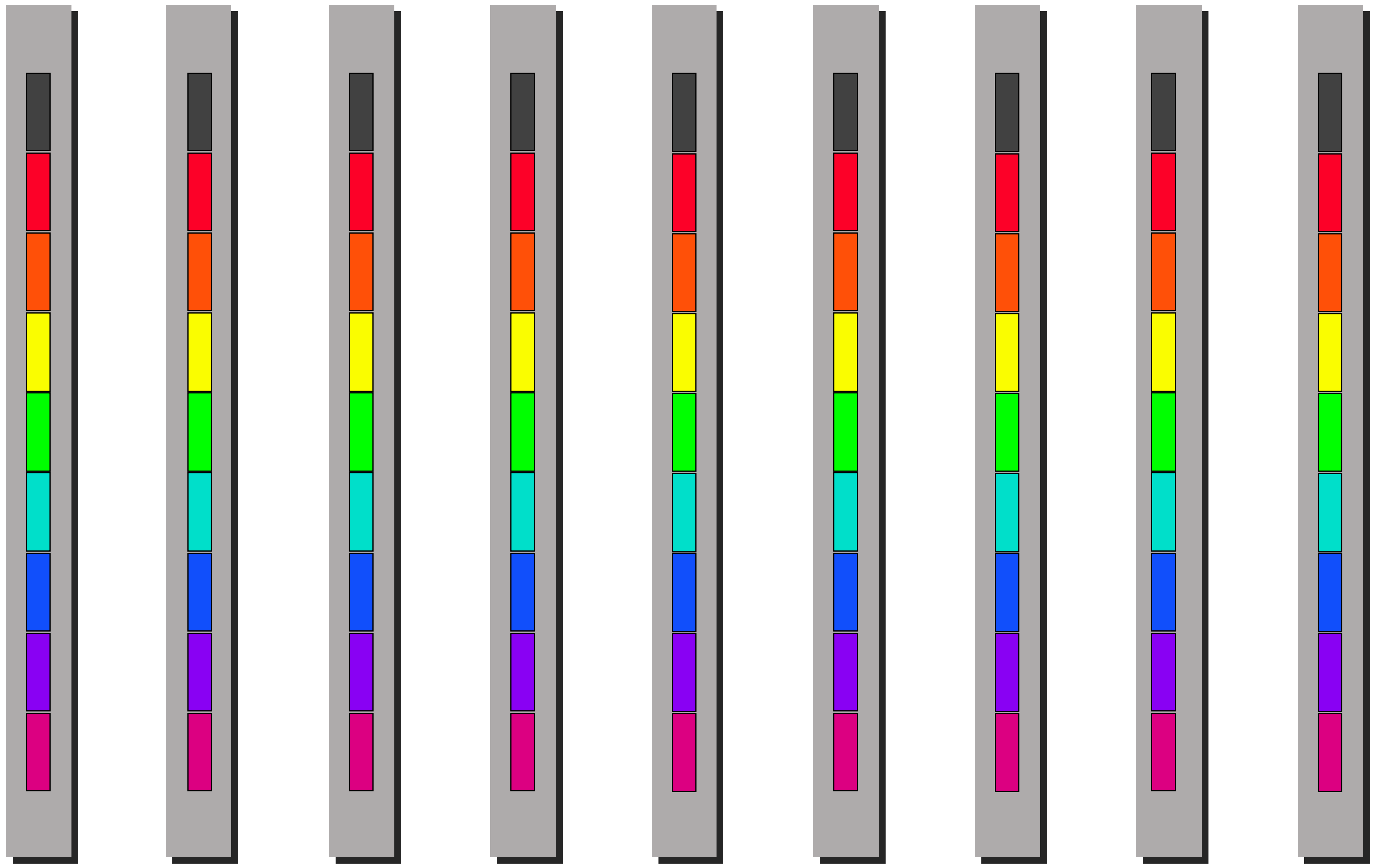








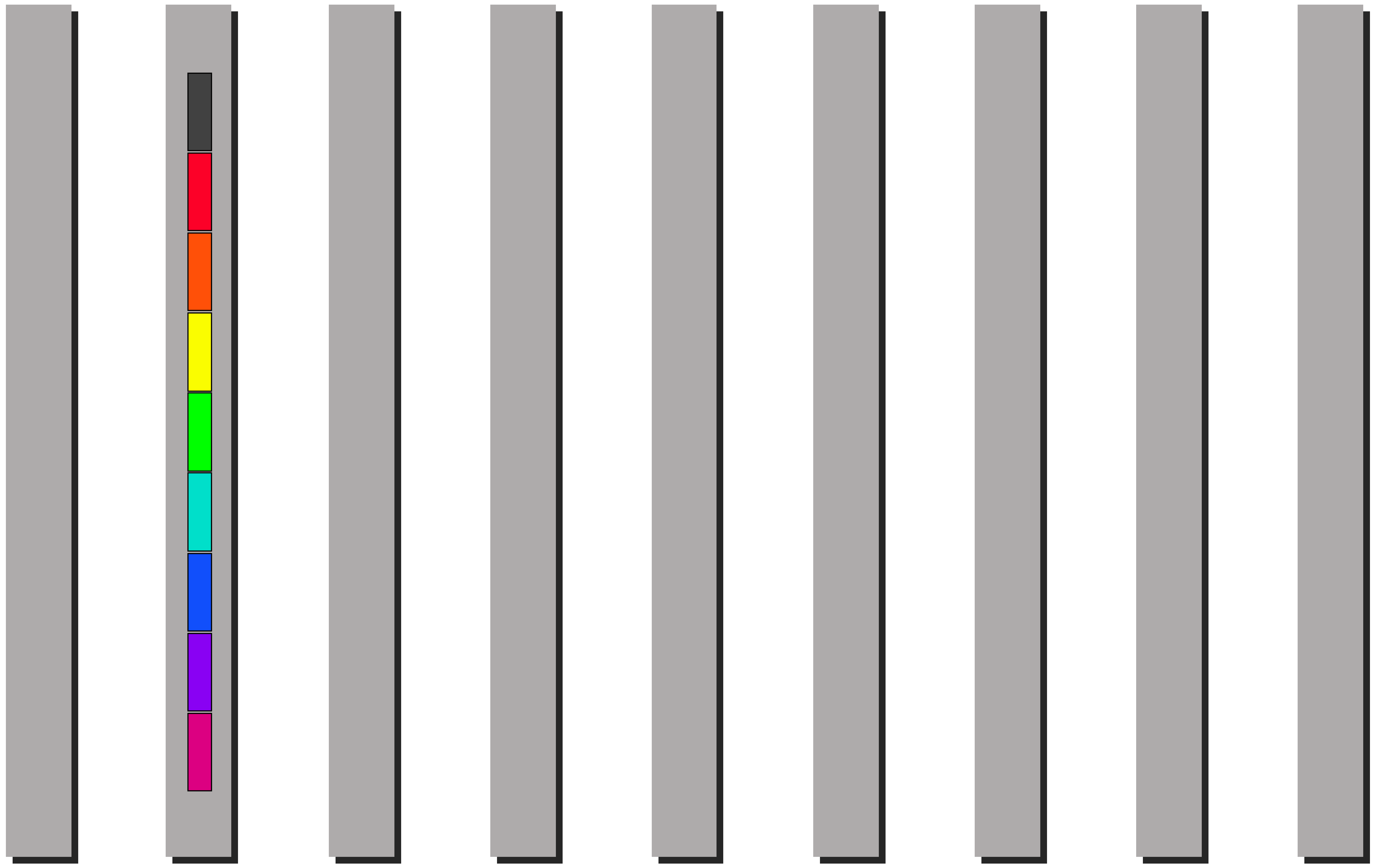


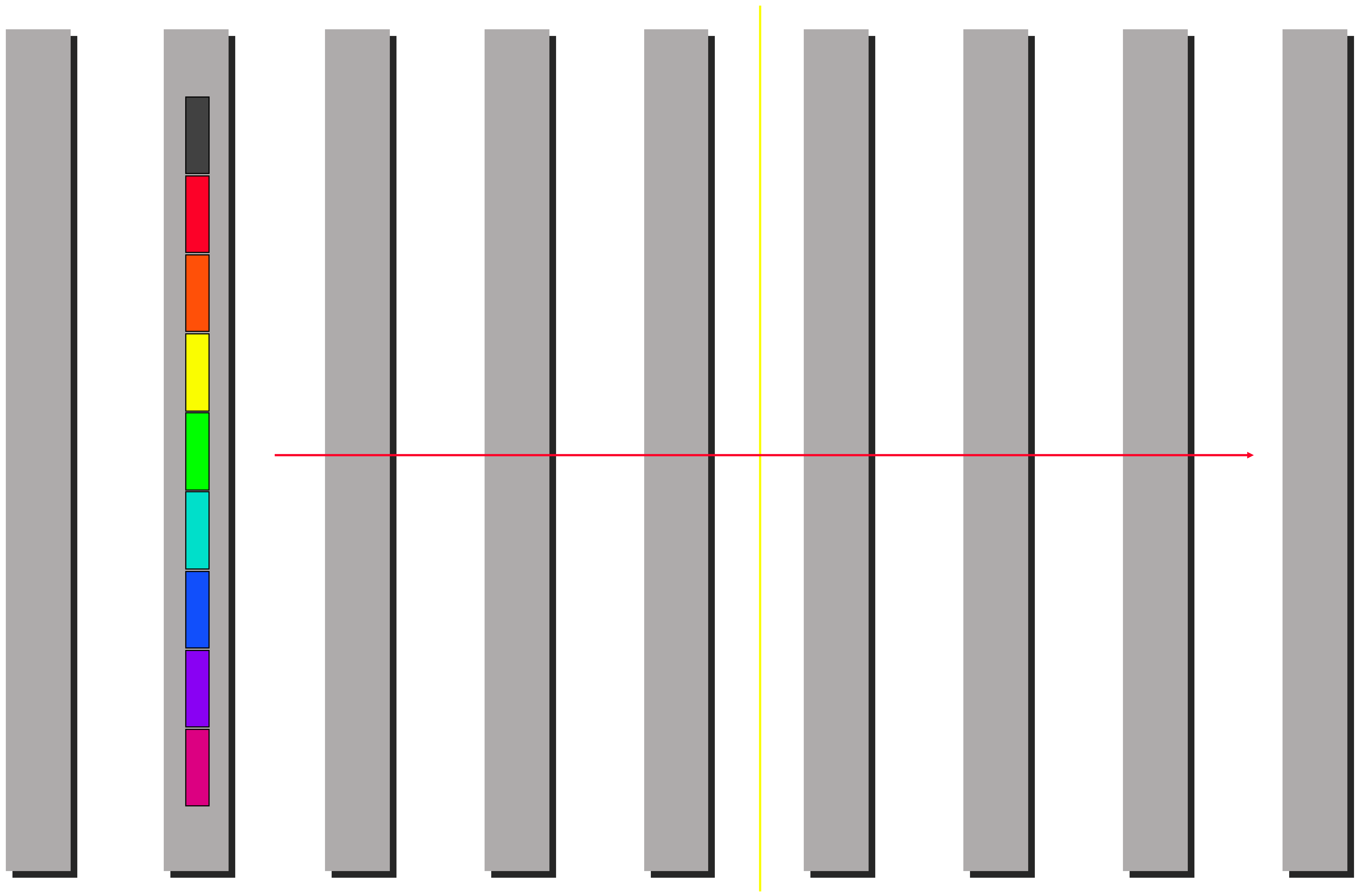


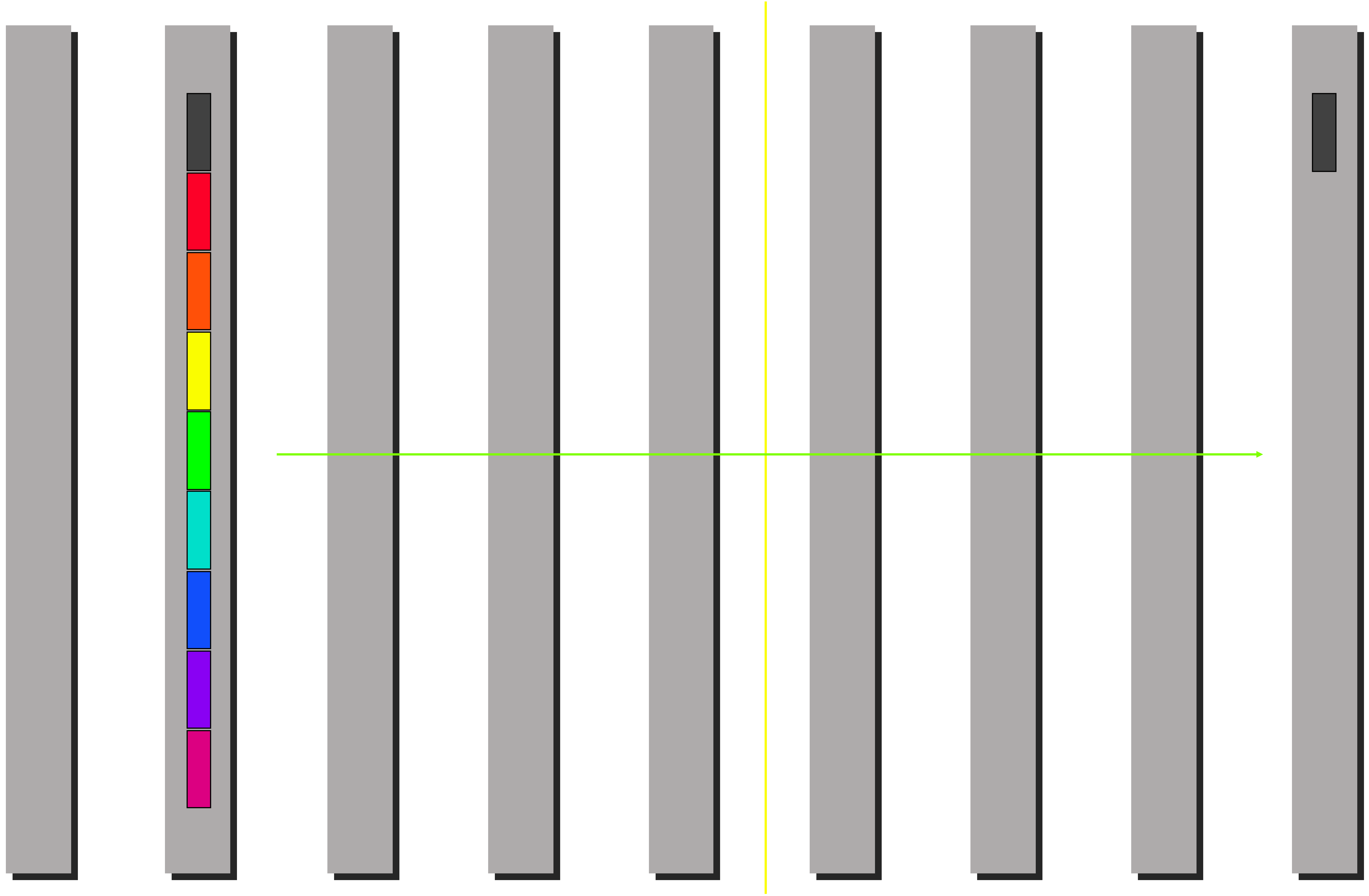
Let us view this more closely

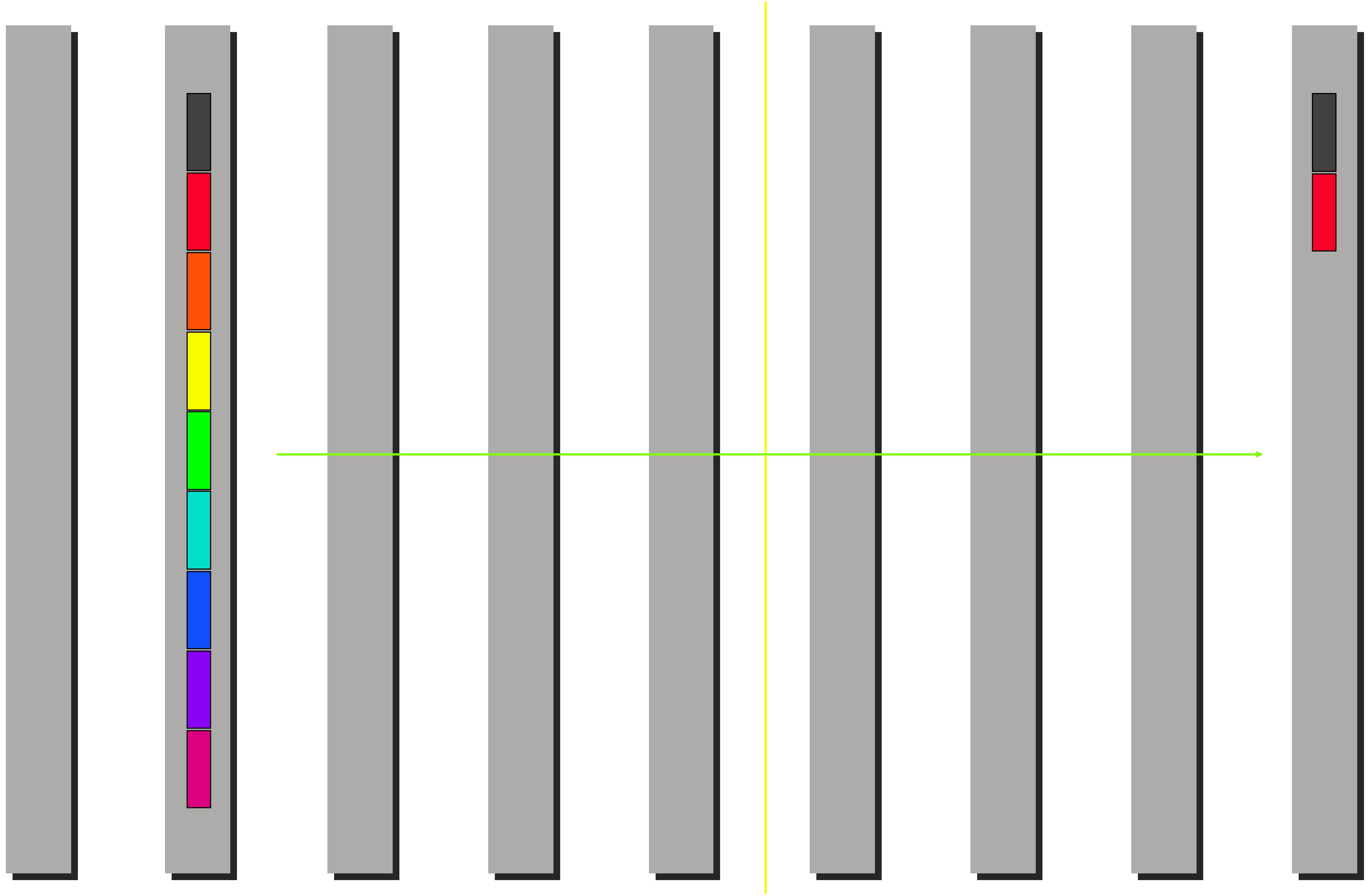
- Red arrows indicate startup of communication (leading to latency, α)

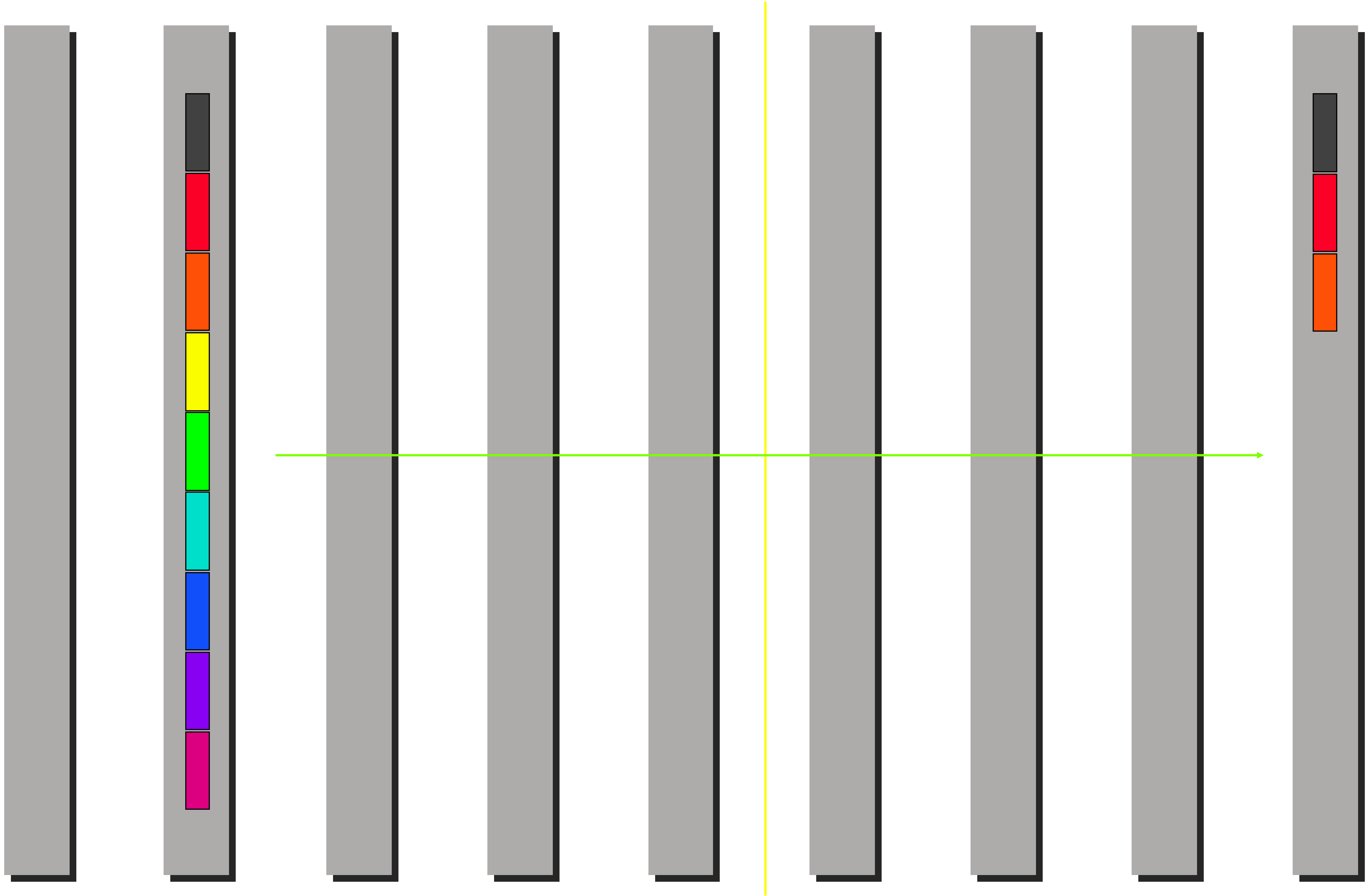
- Green arrows indicate packets in transit (leading to a bandwidth related cost proportional to β and the length of the packet

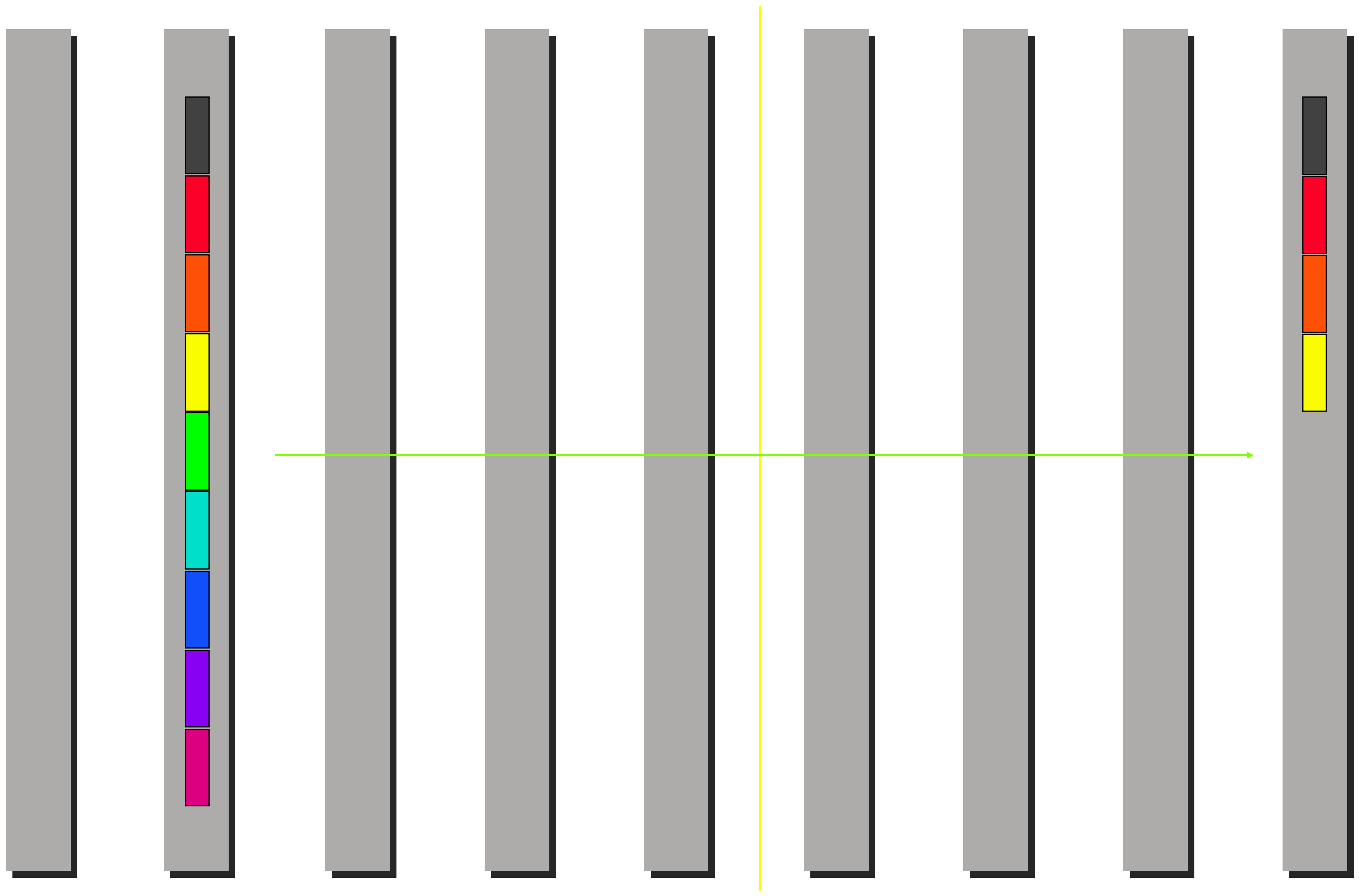


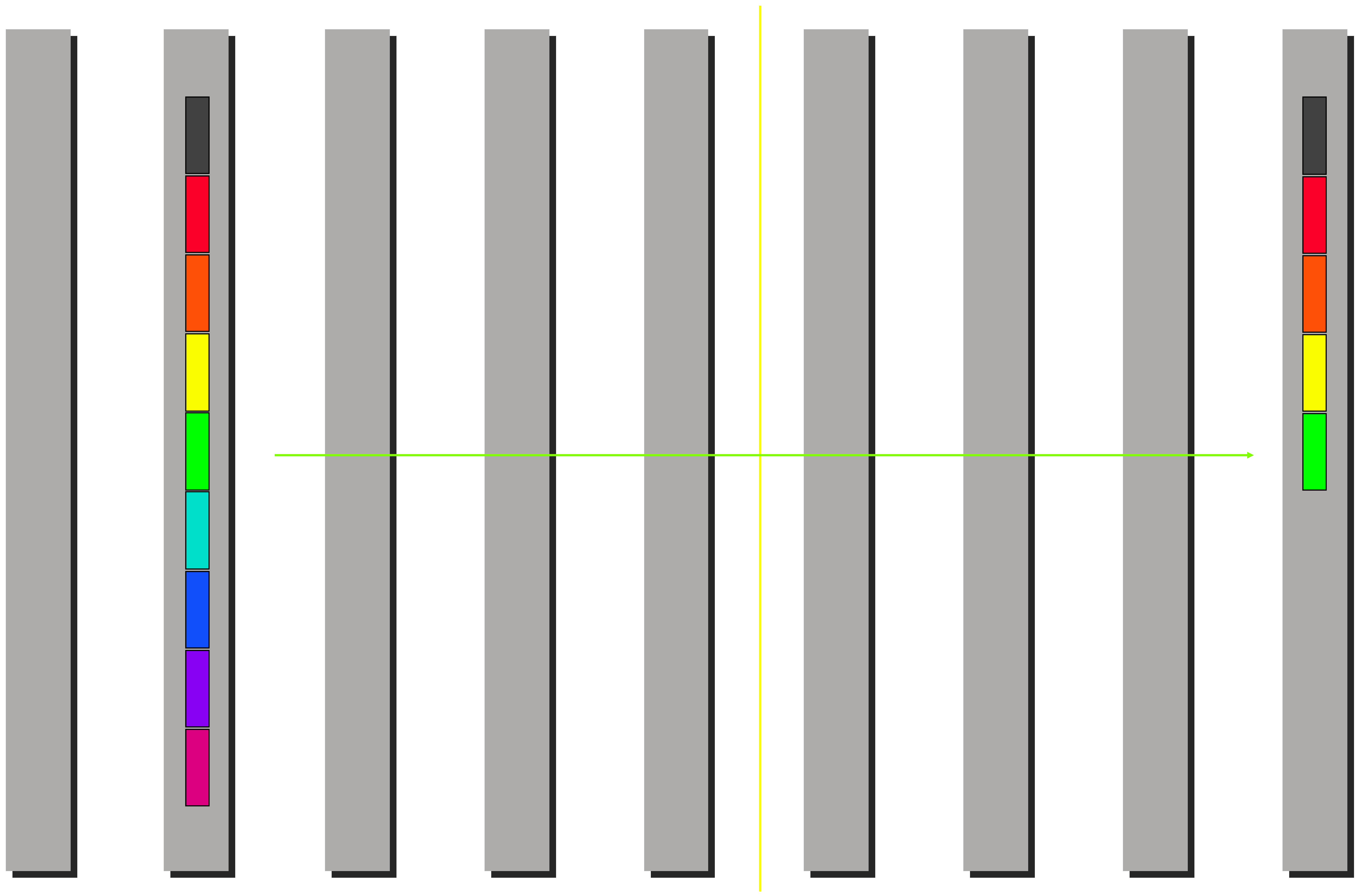


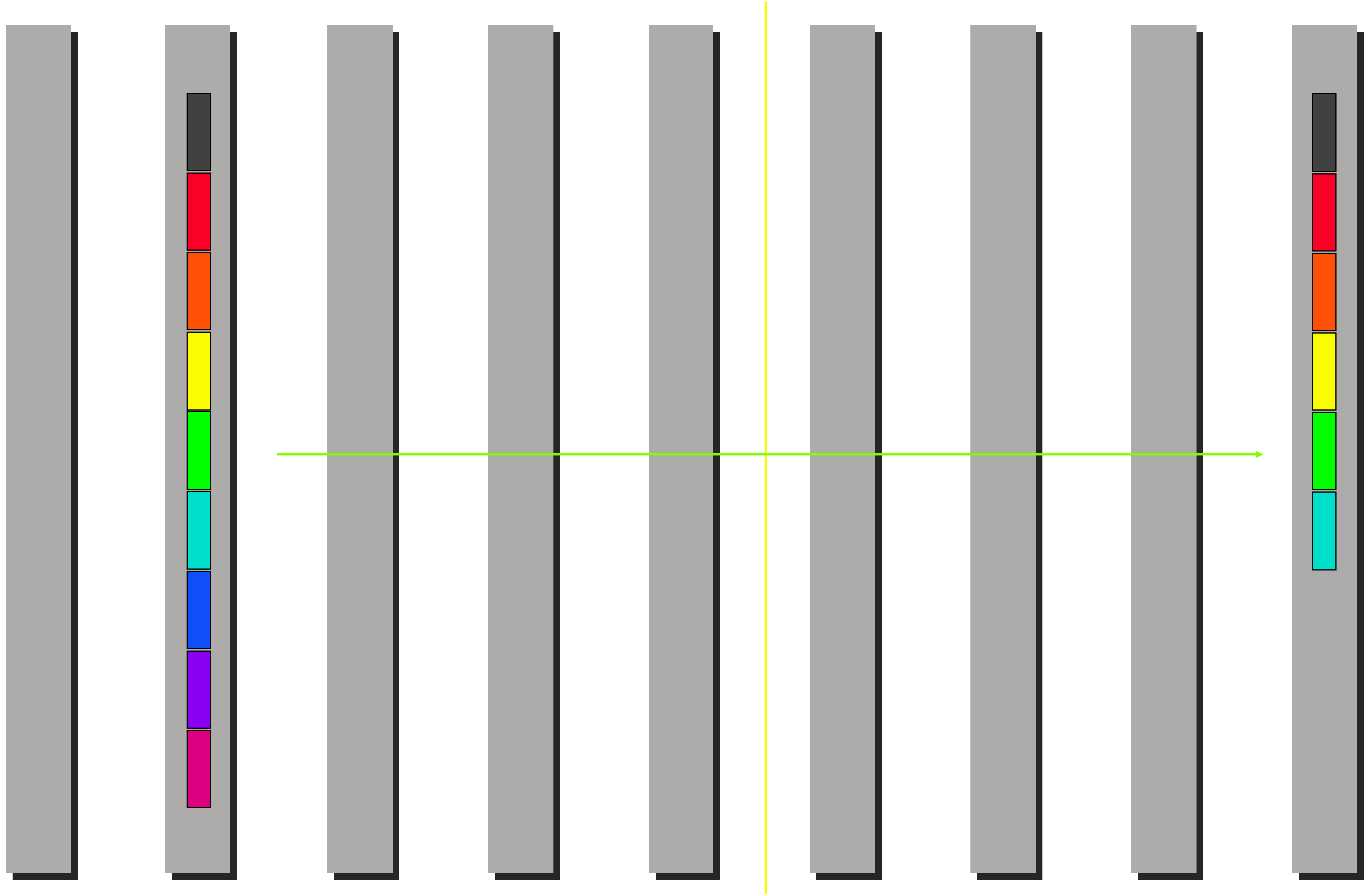


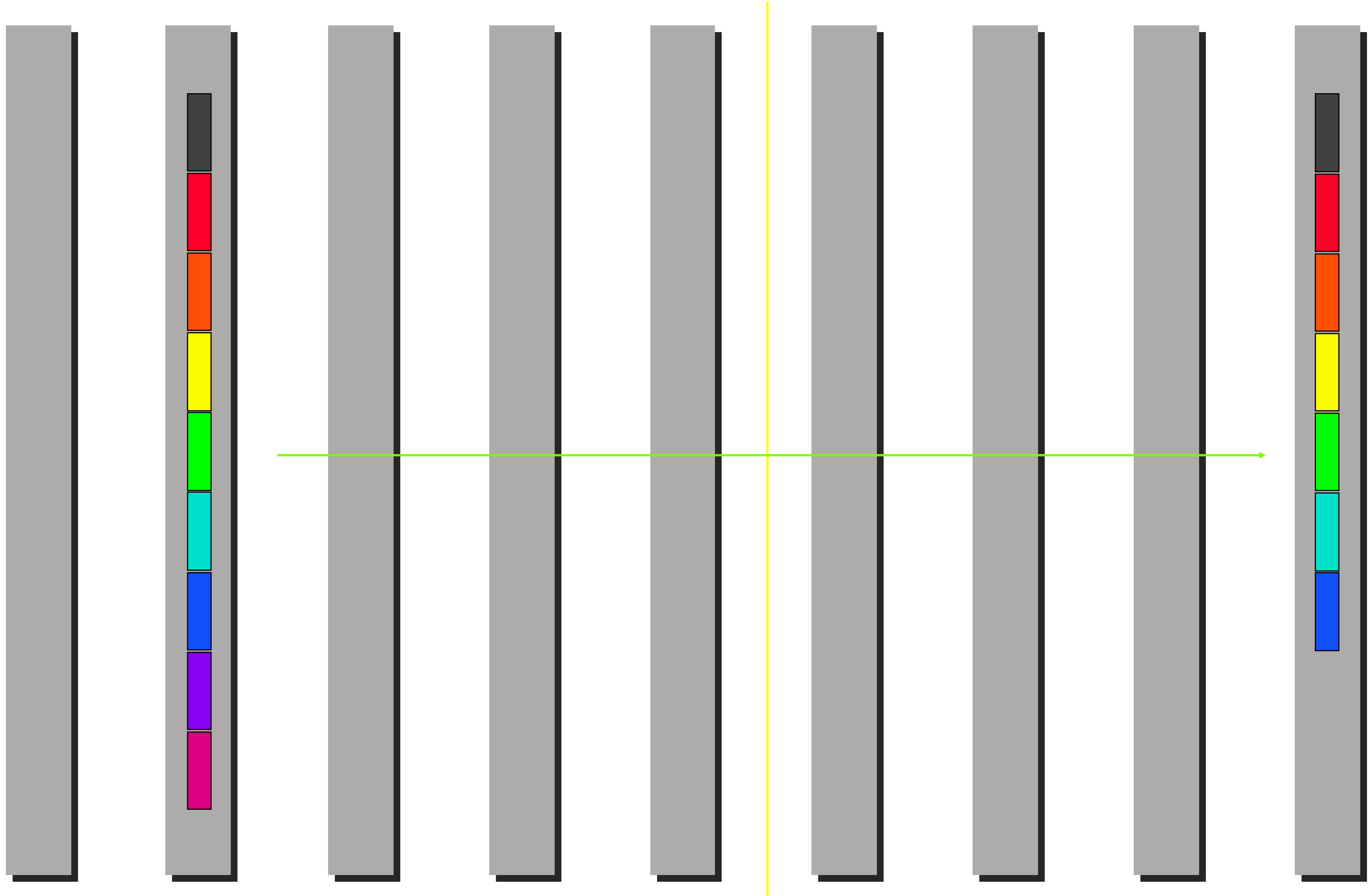


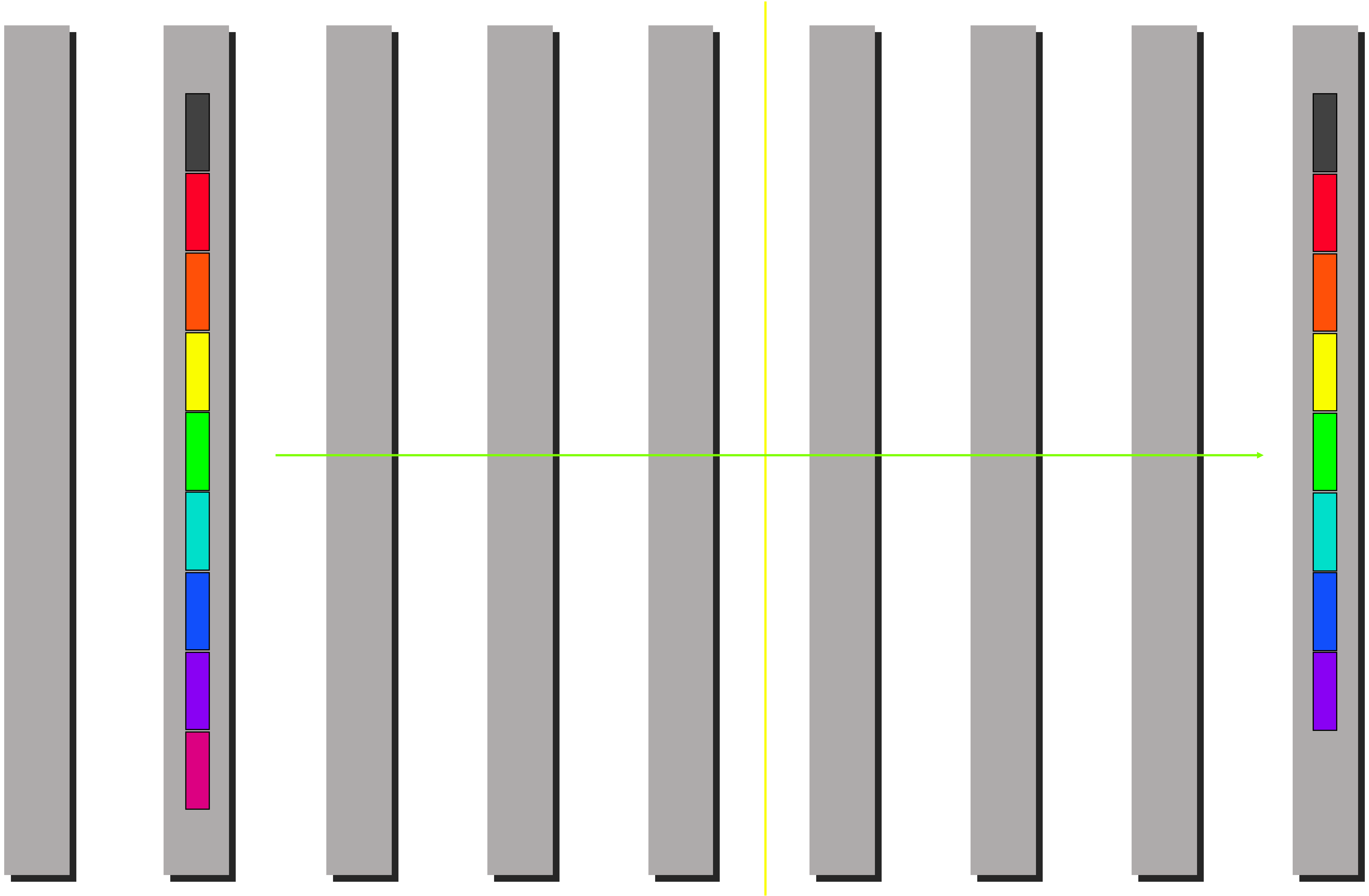


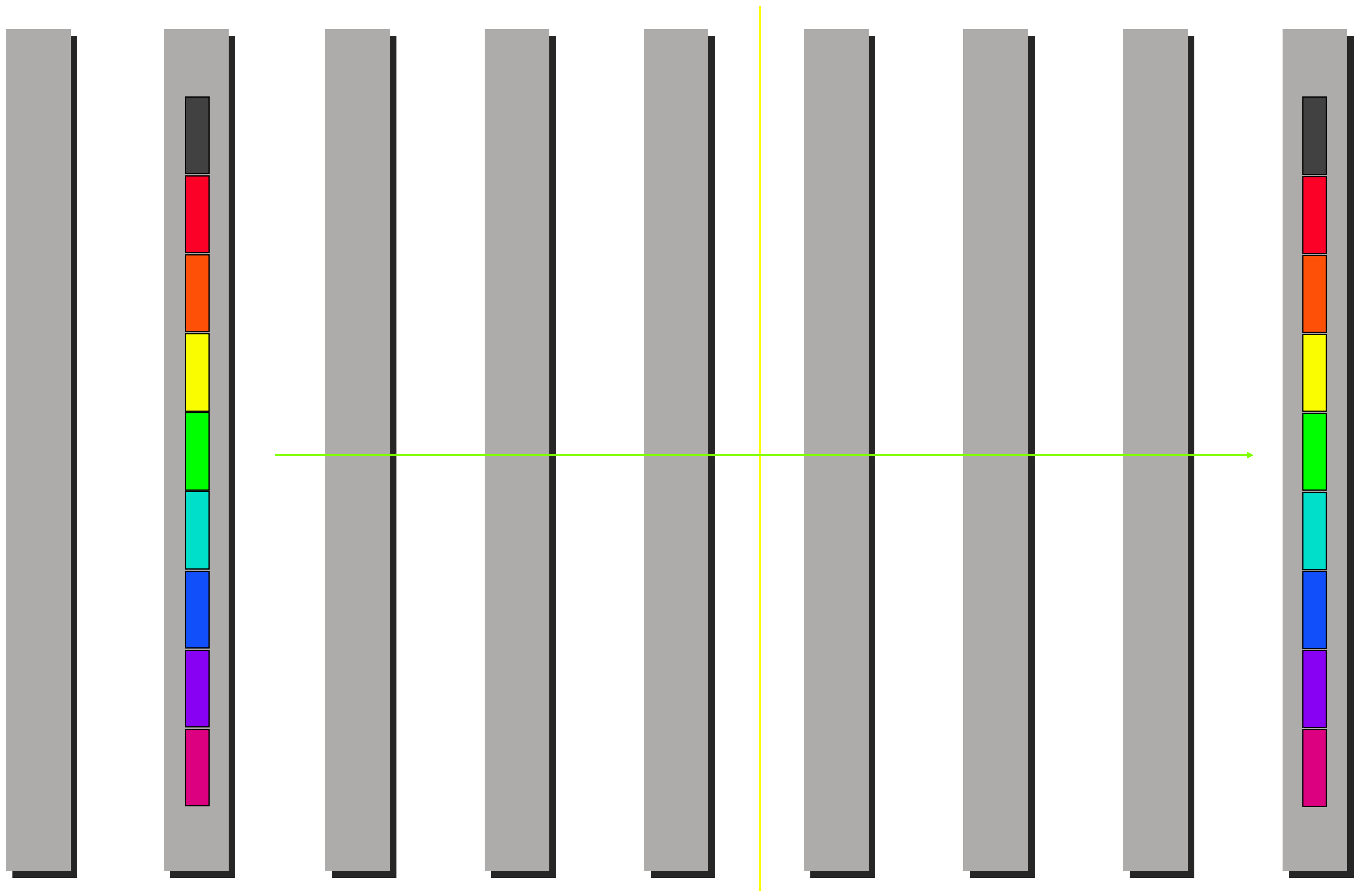


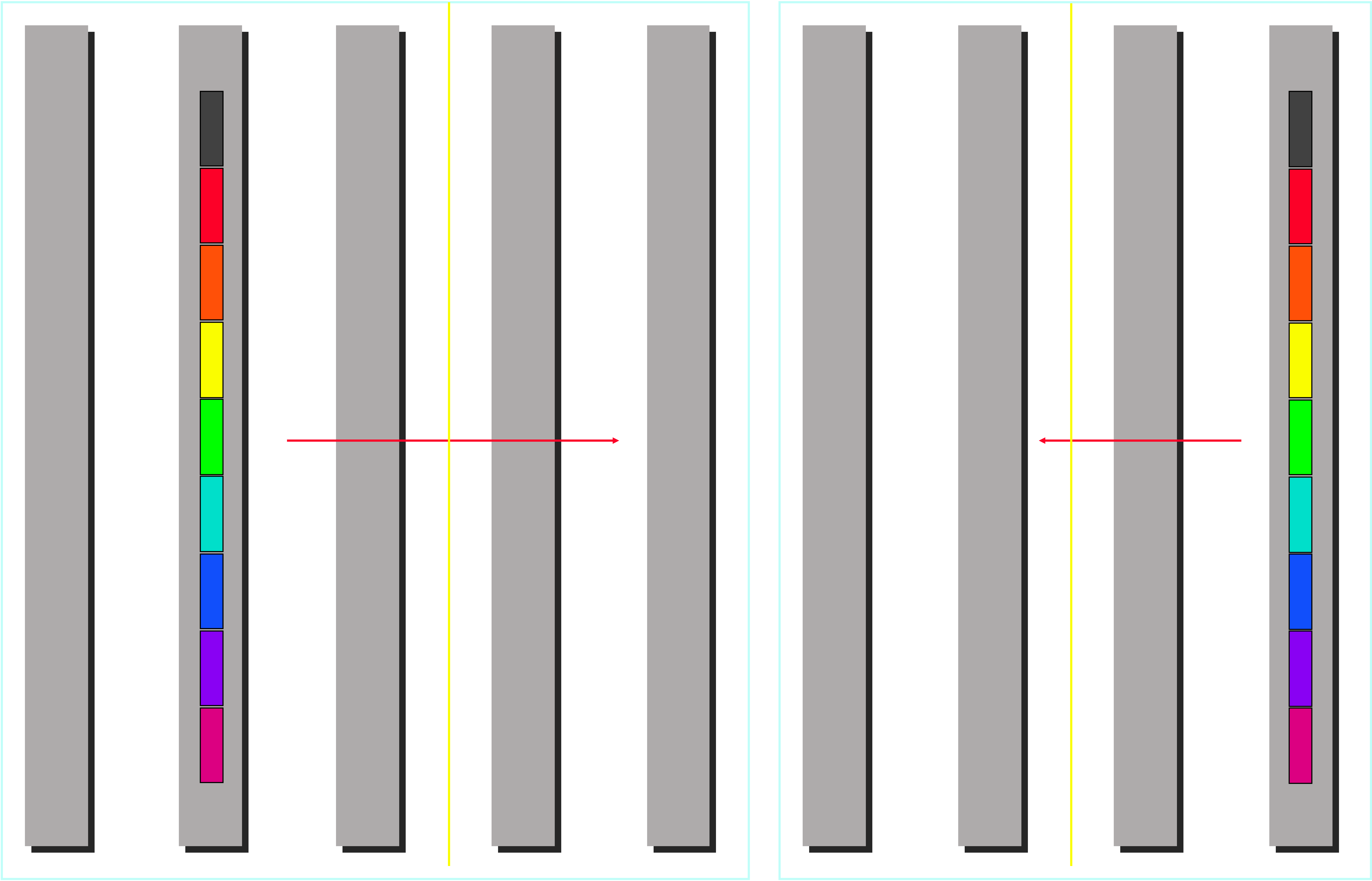


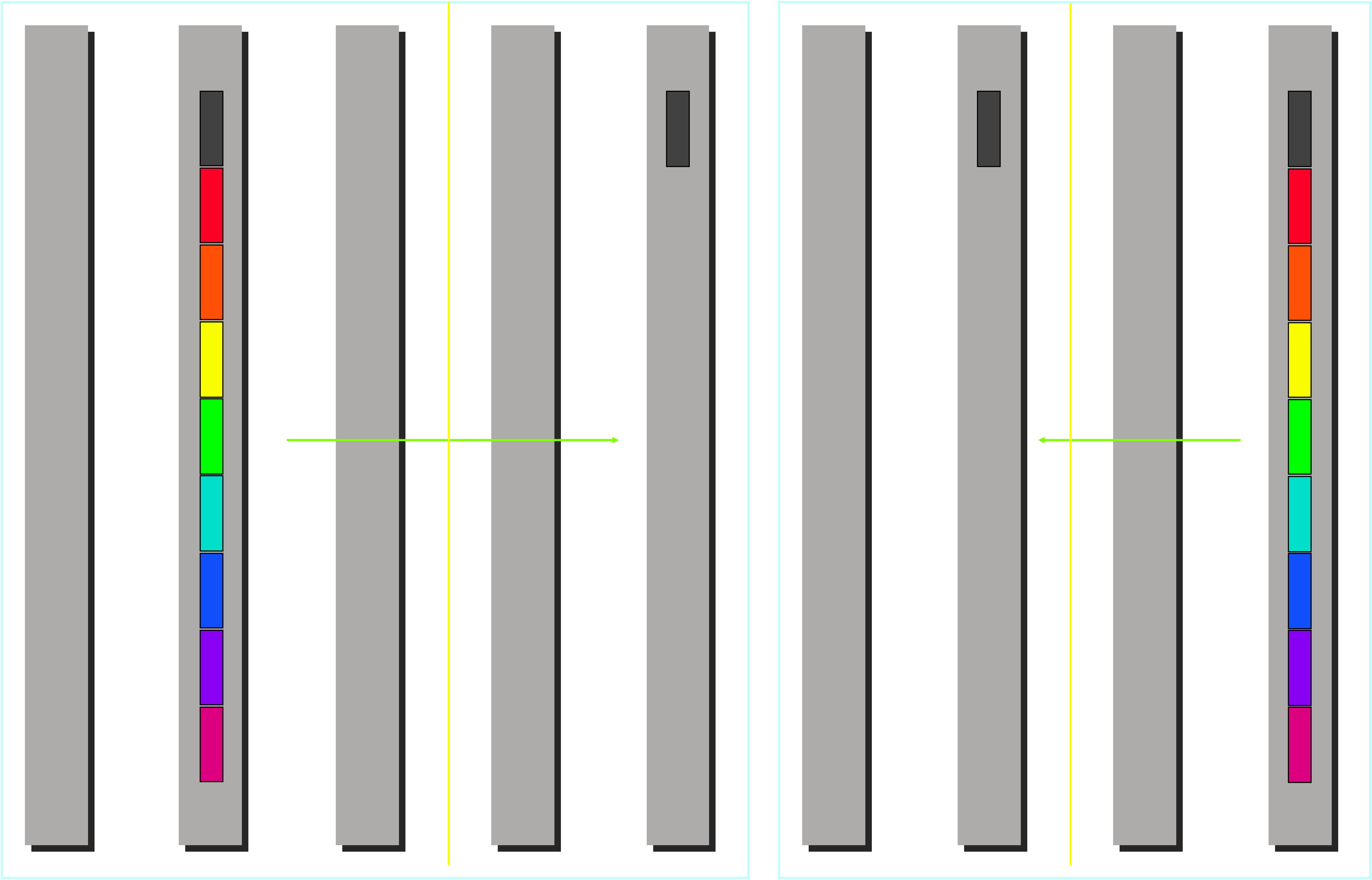


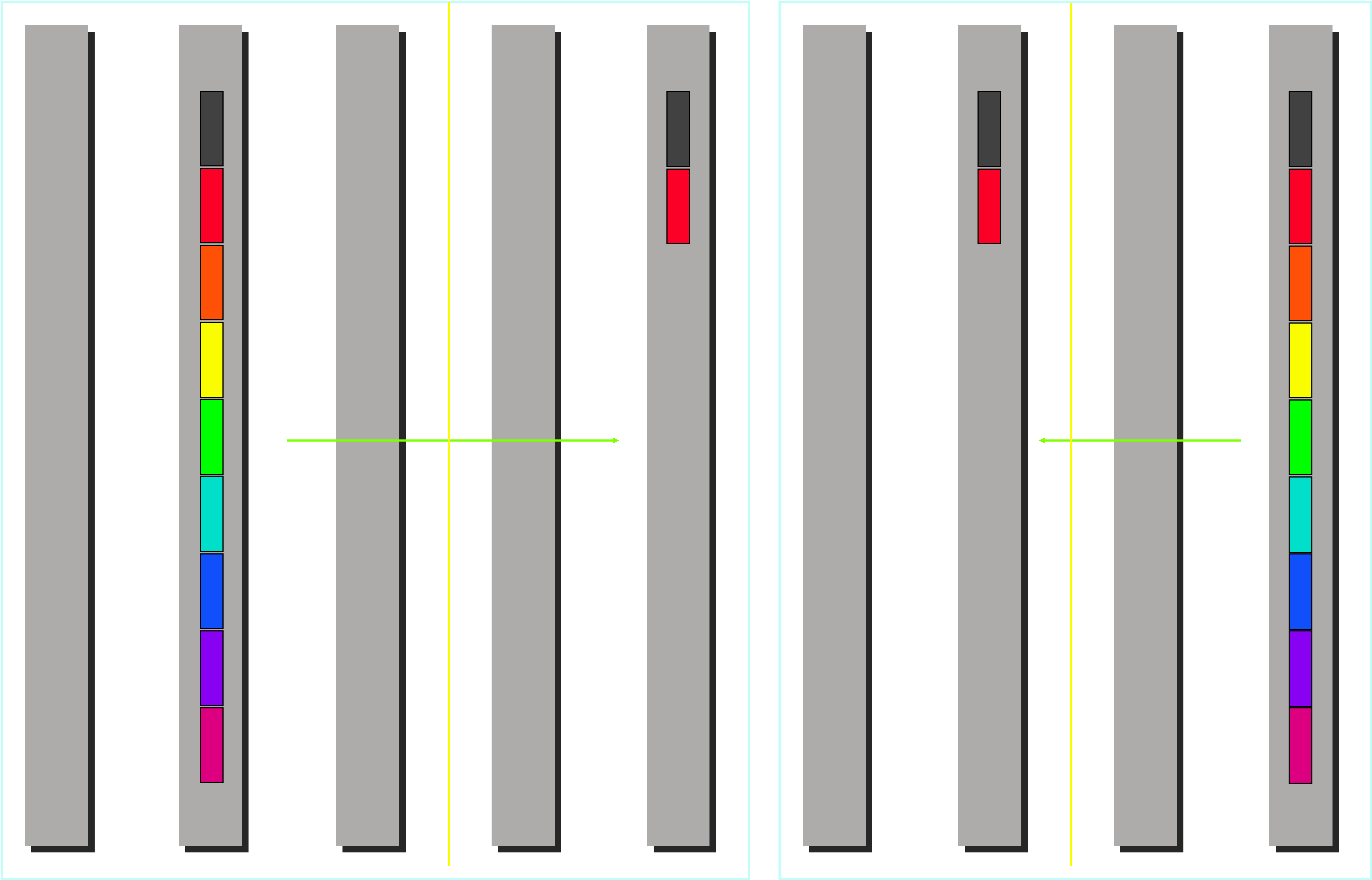


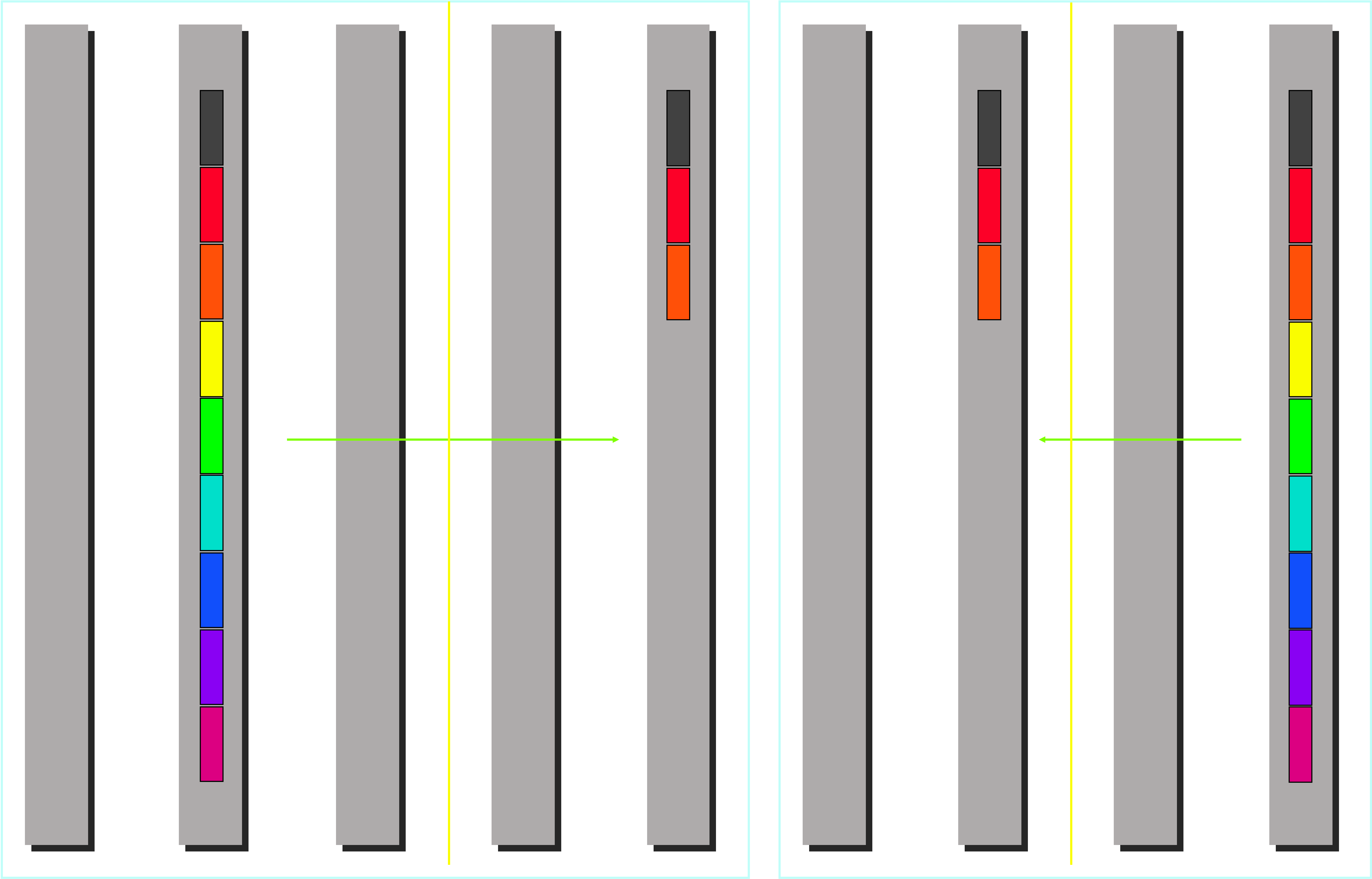


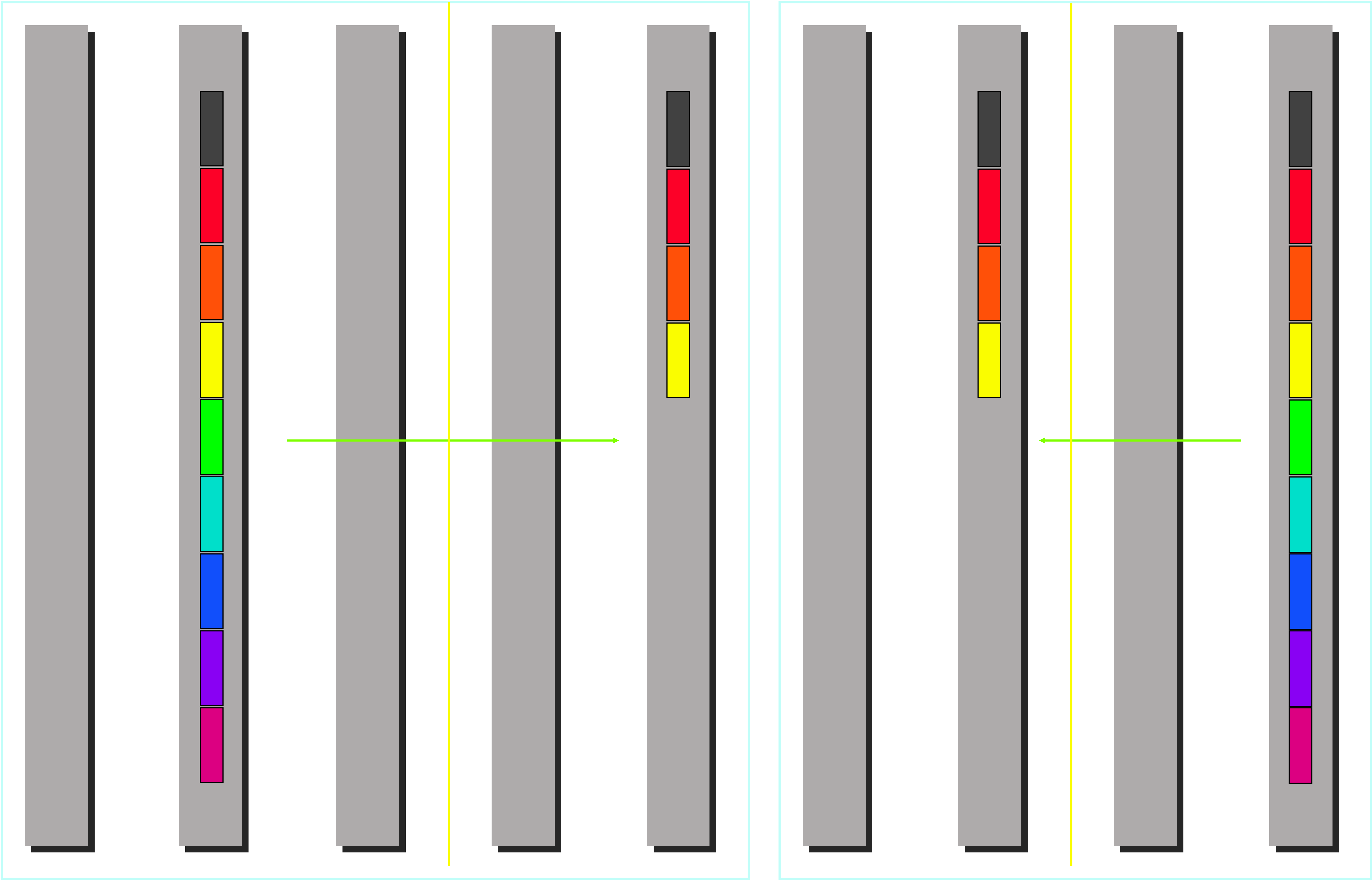


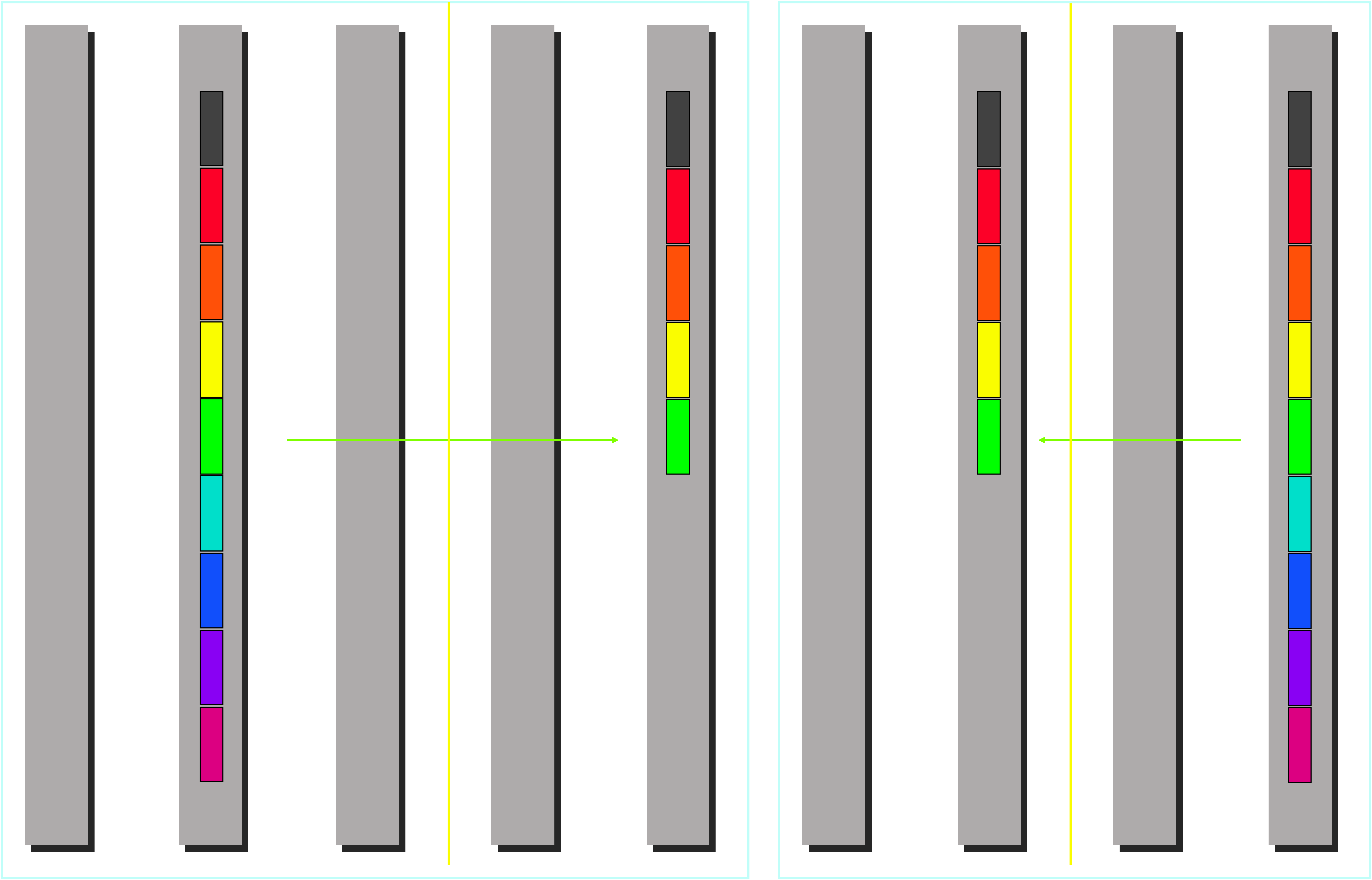


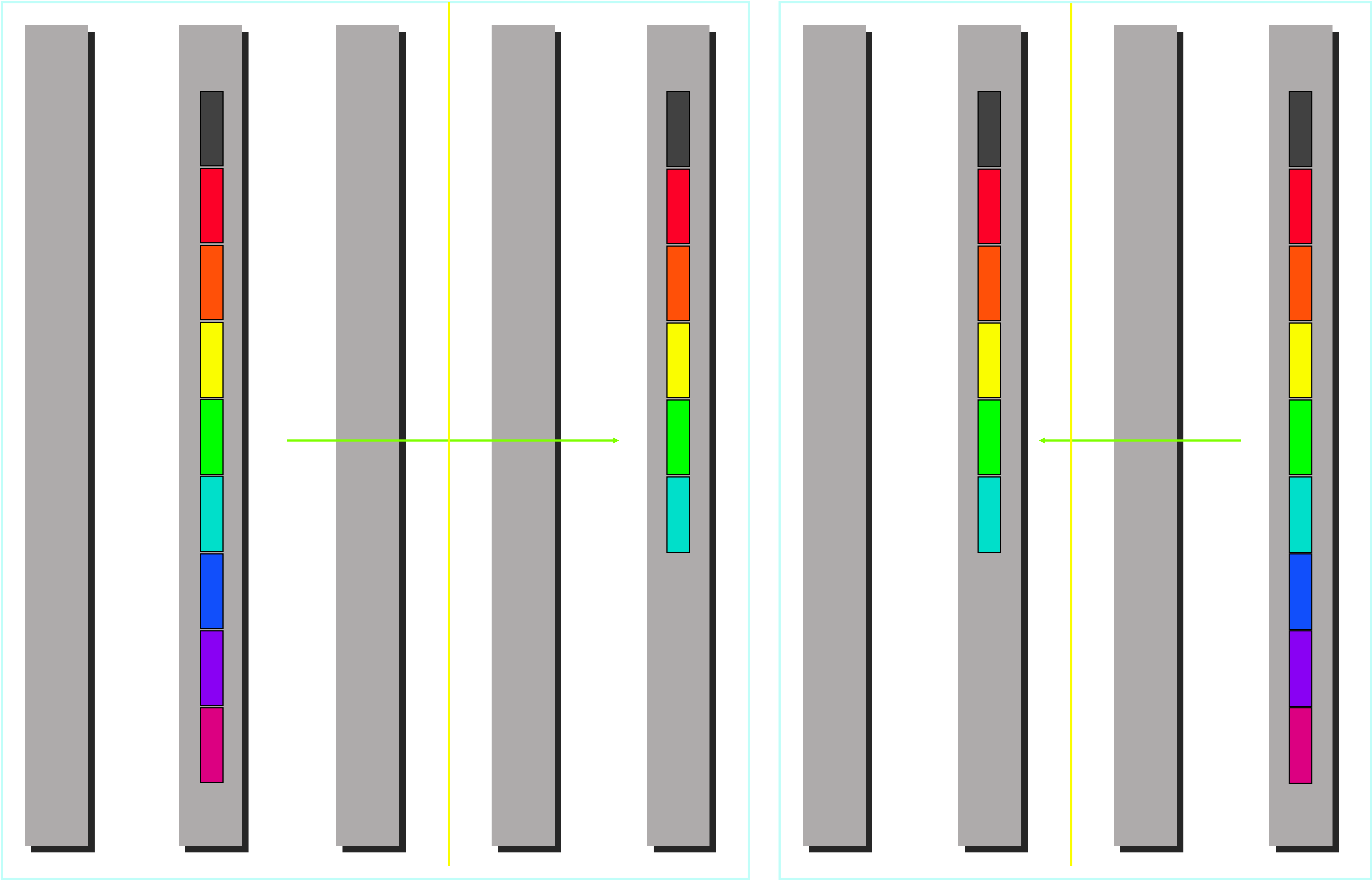


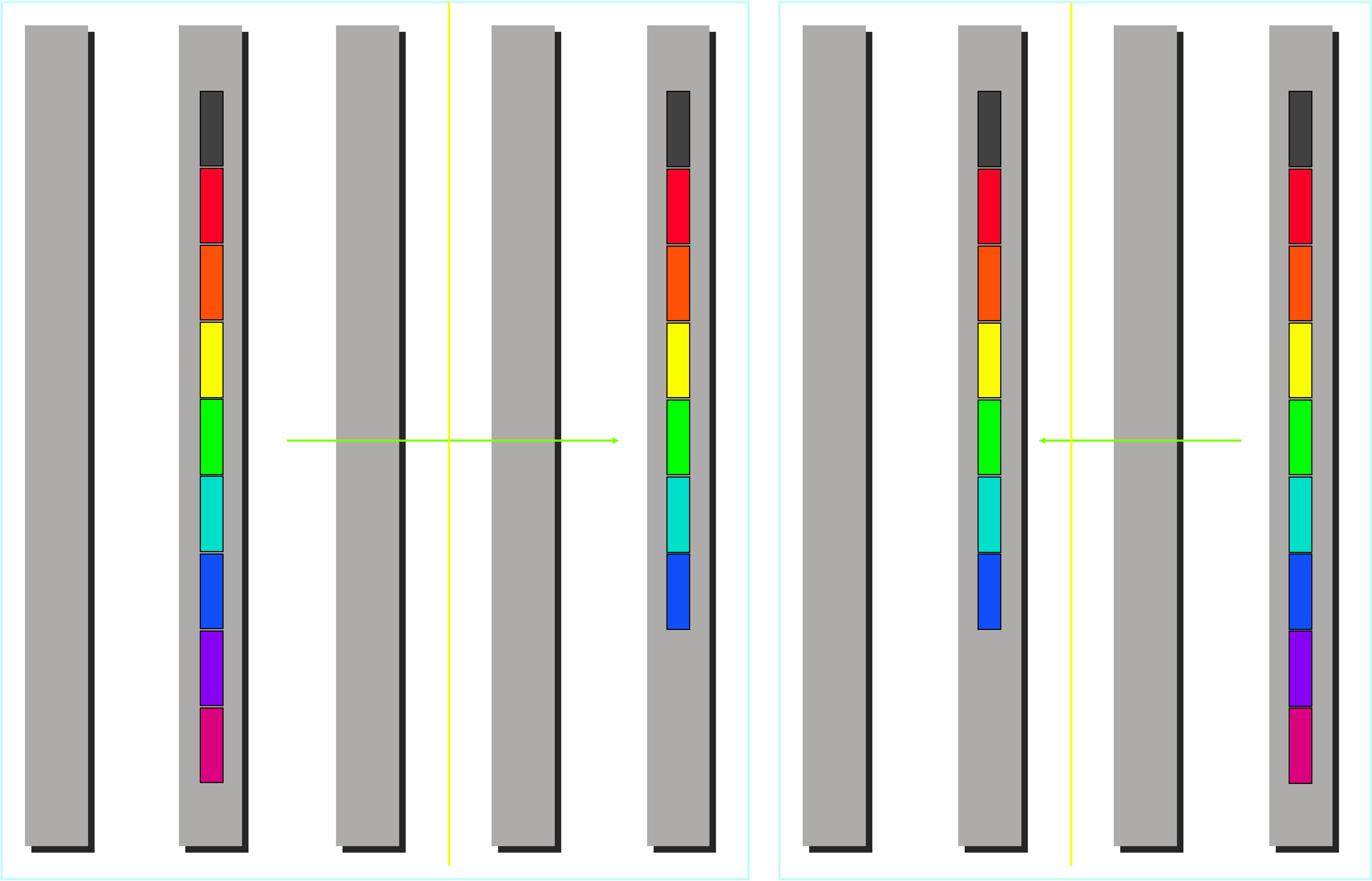


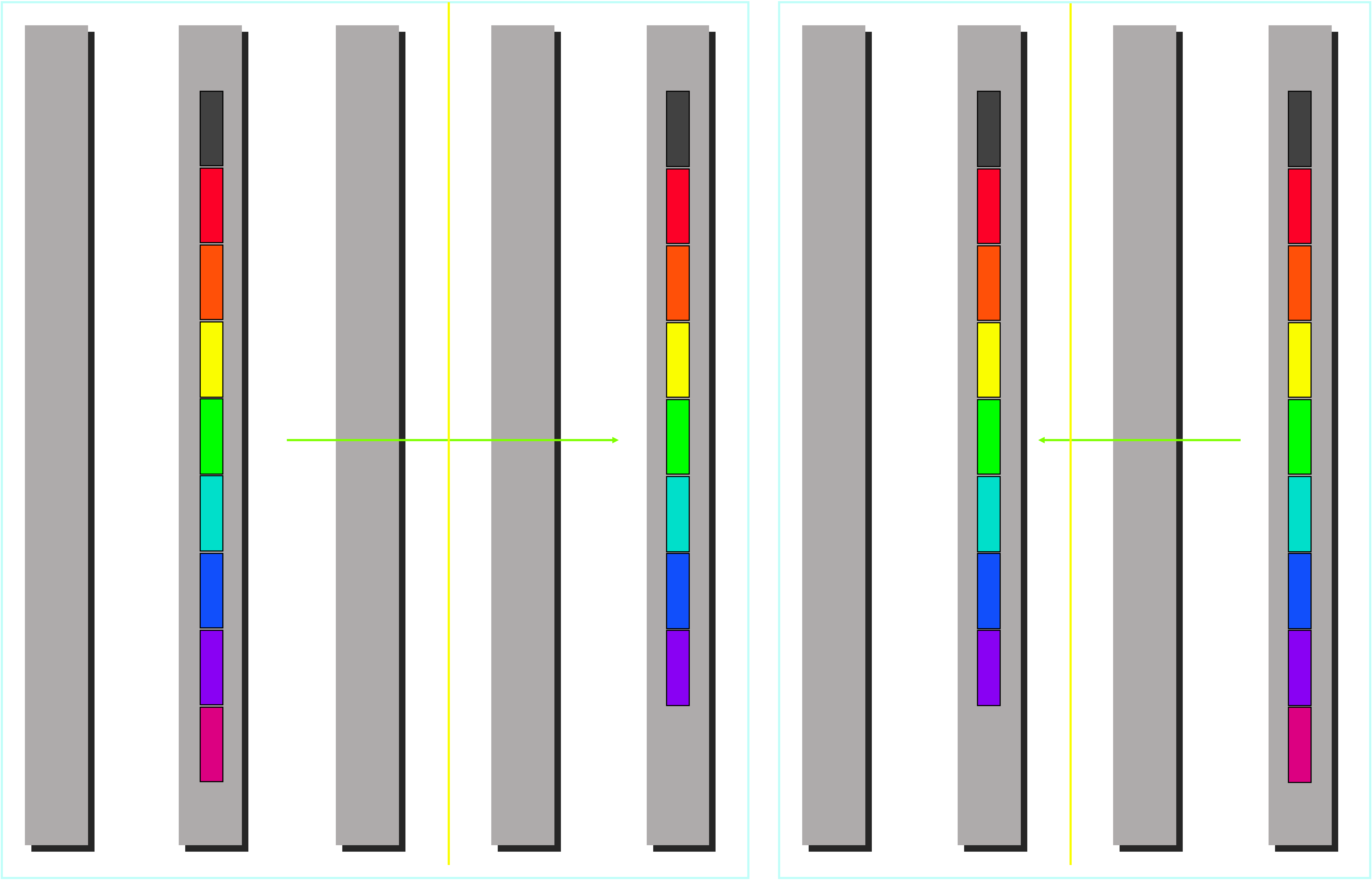


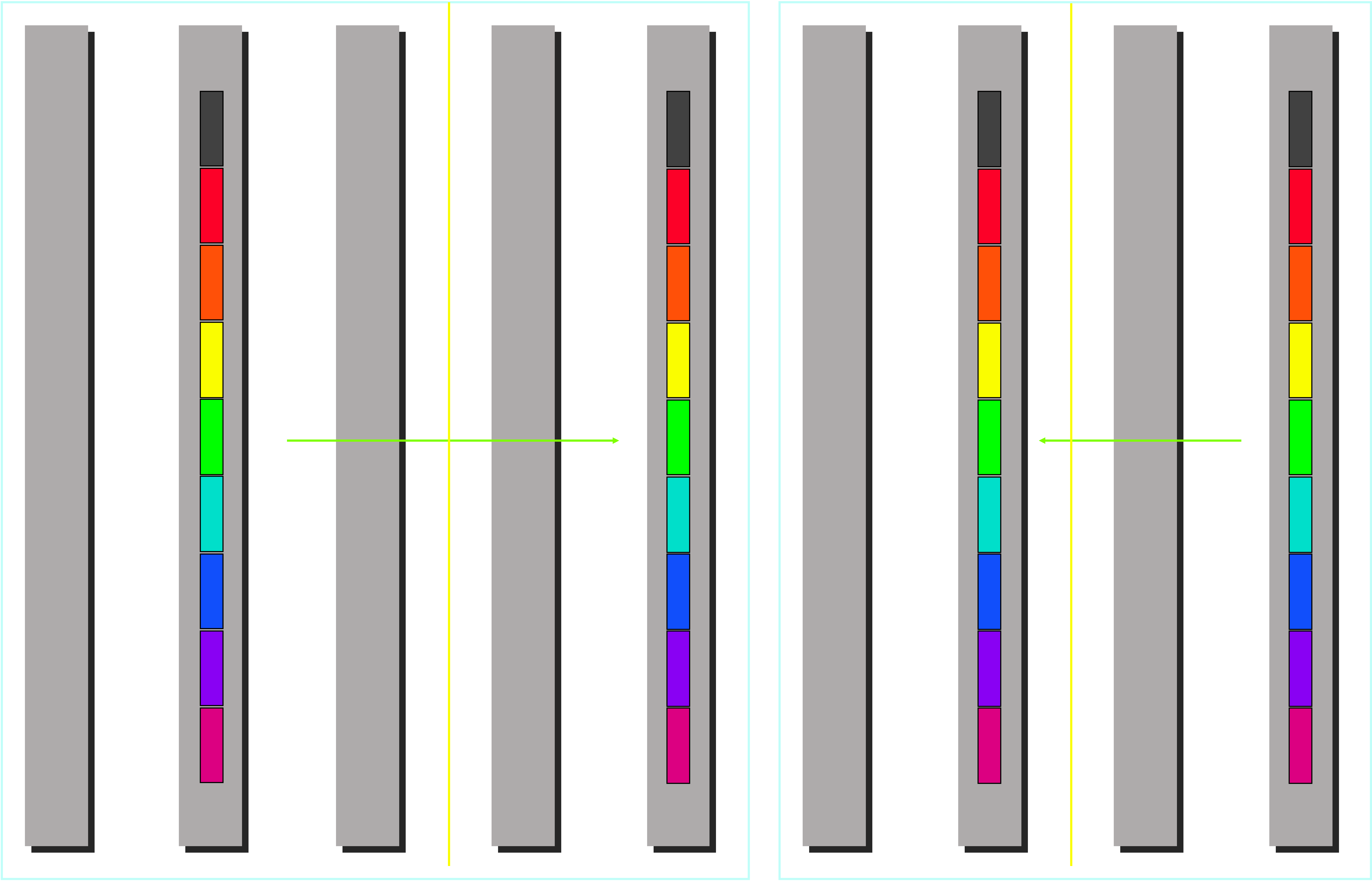


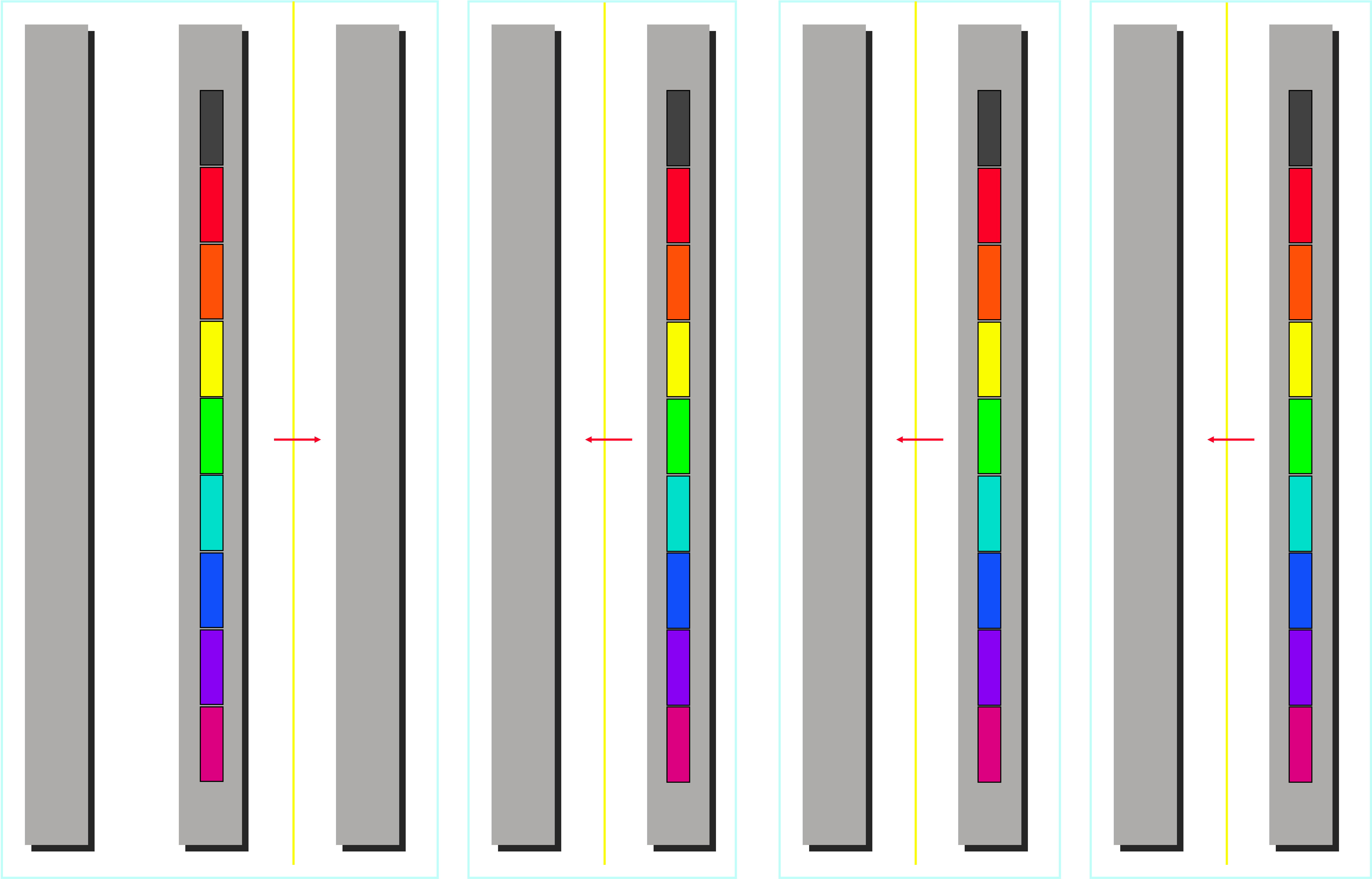


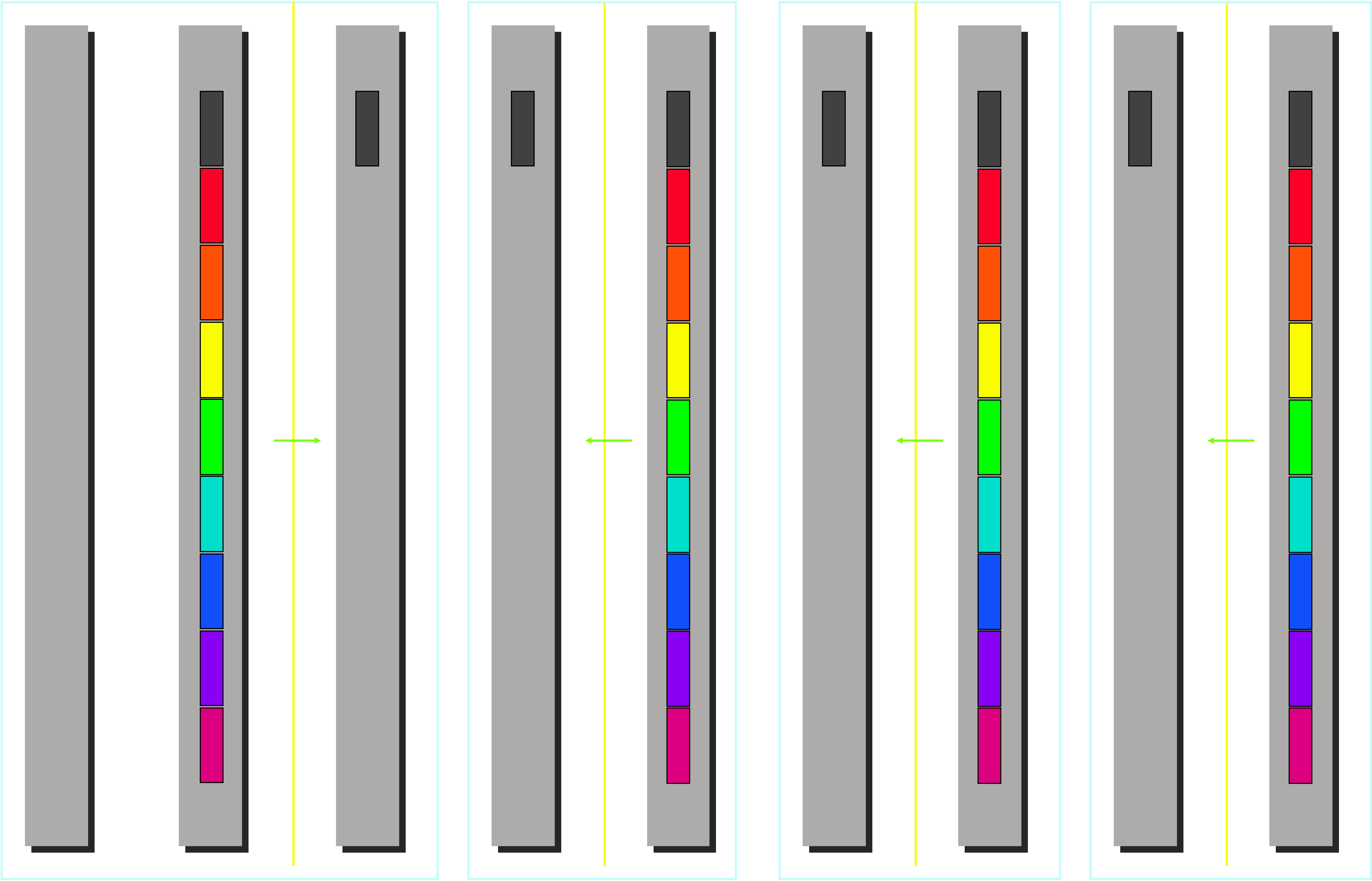


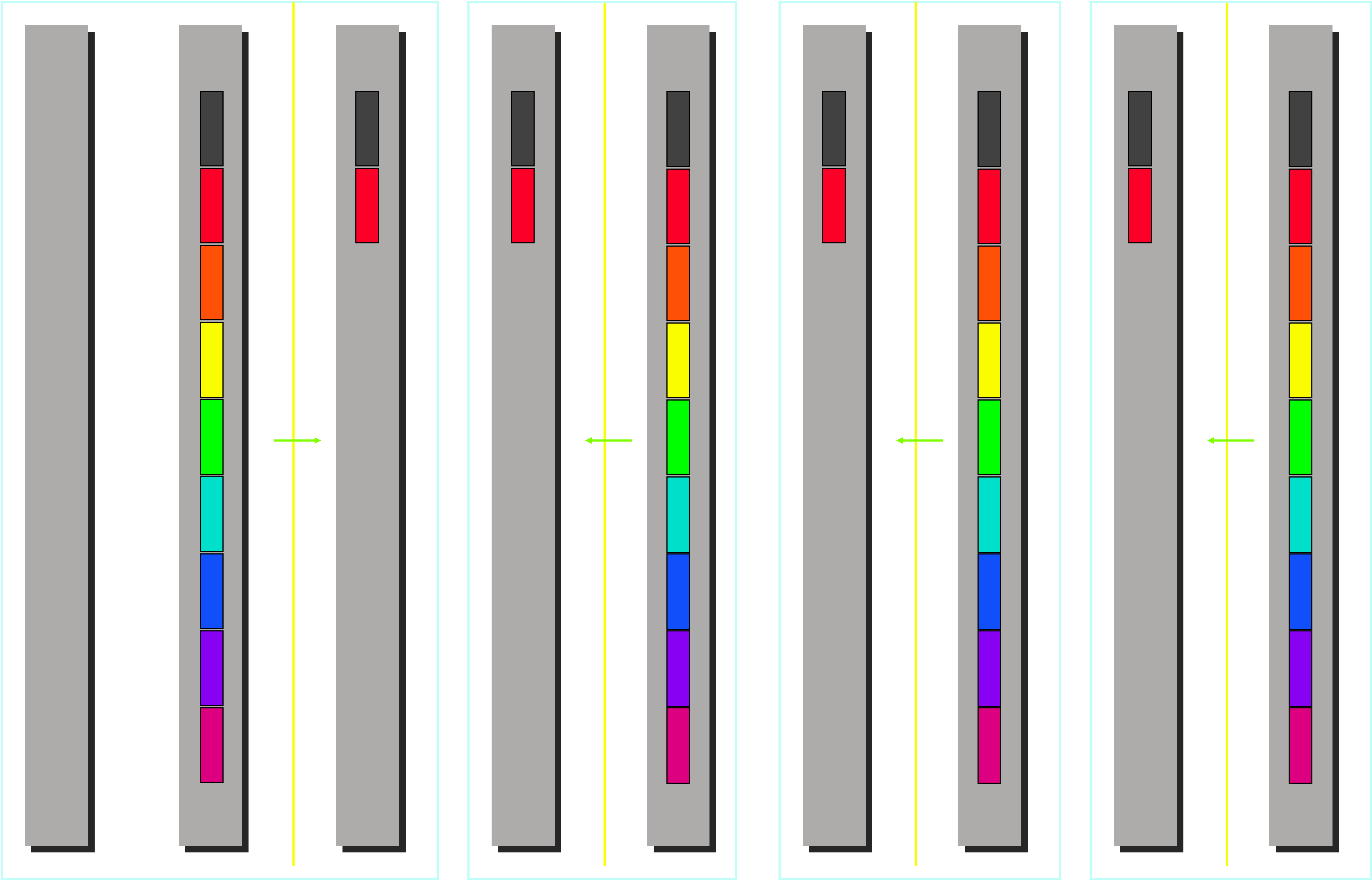


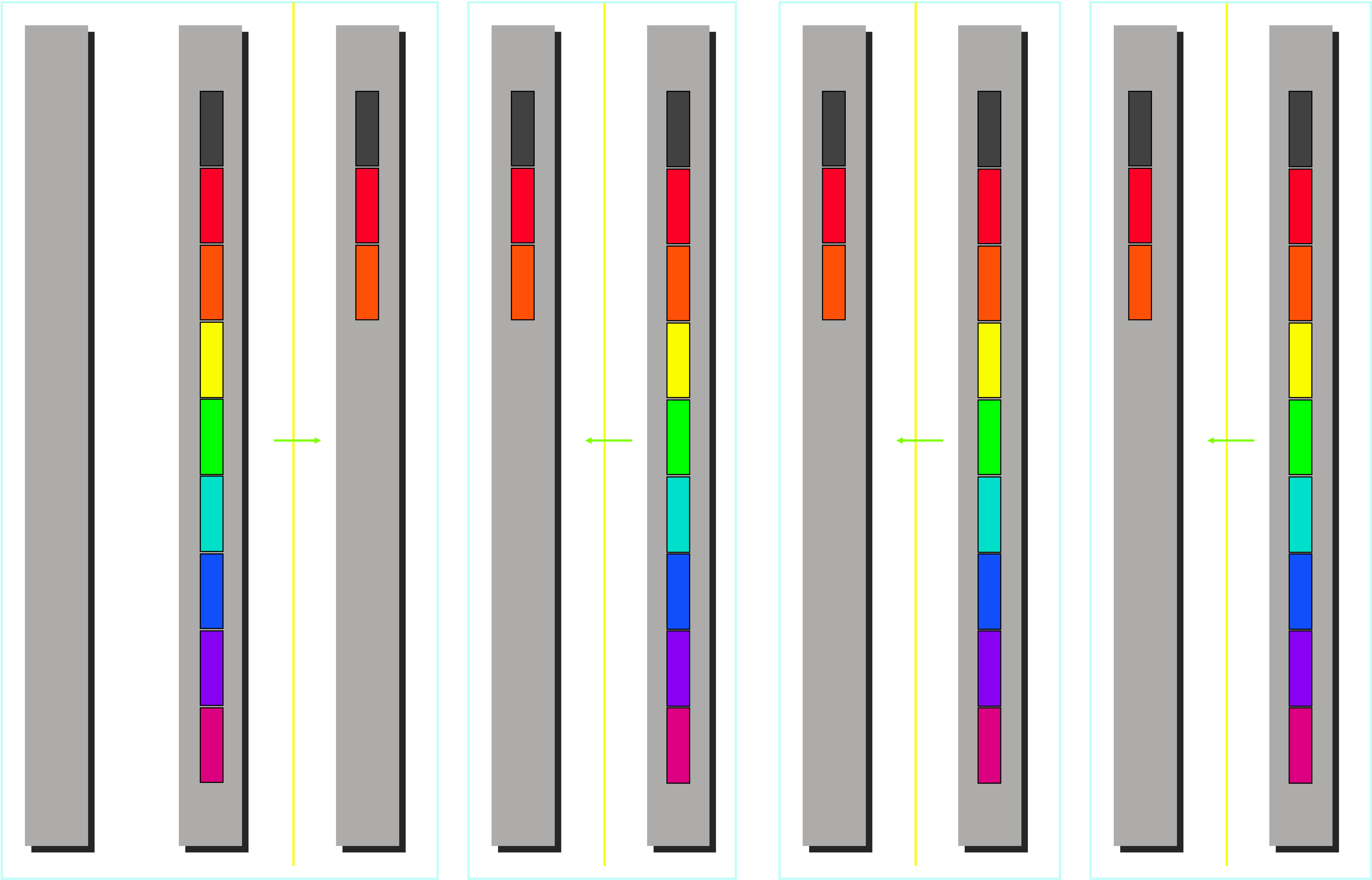


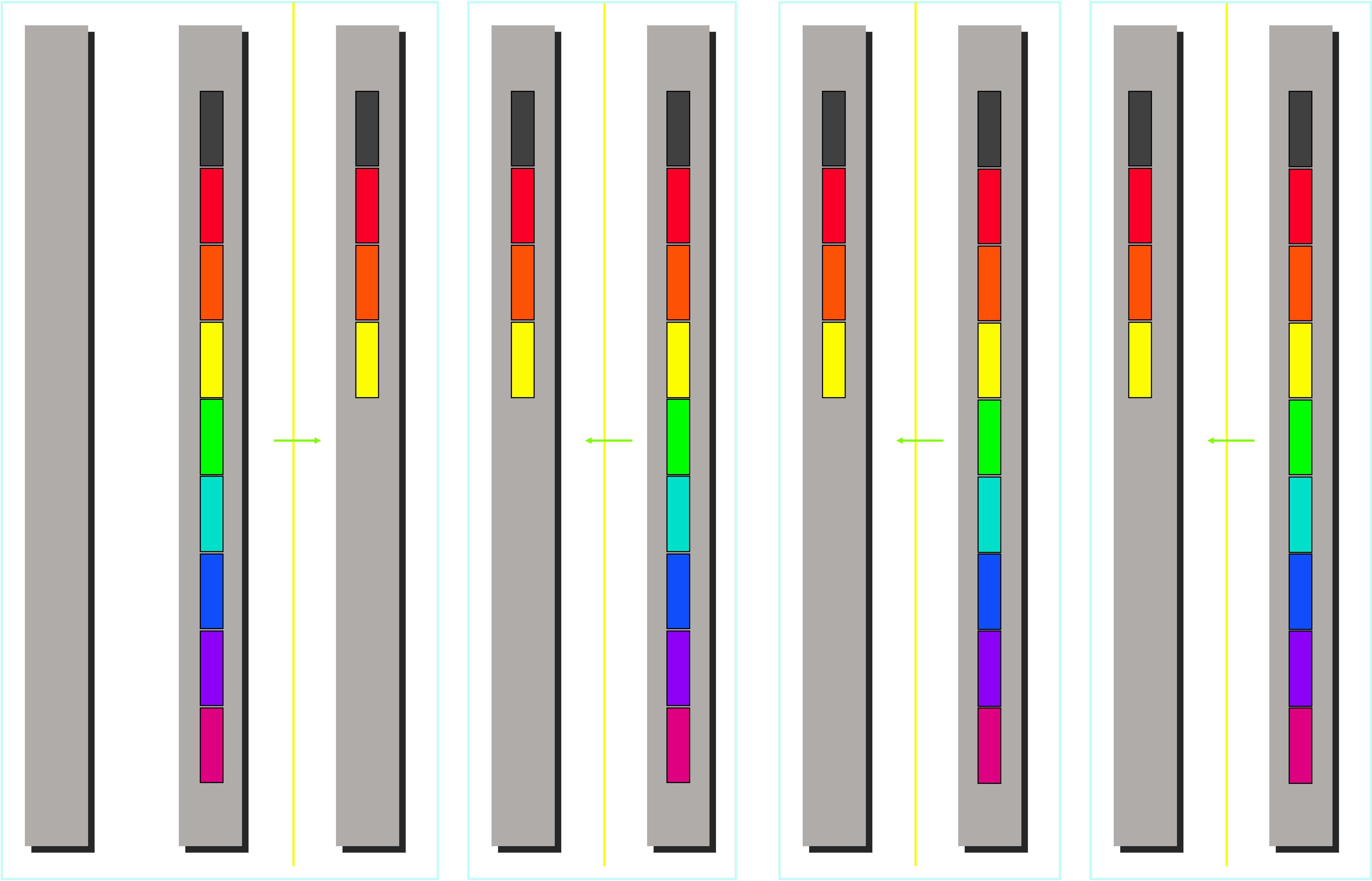


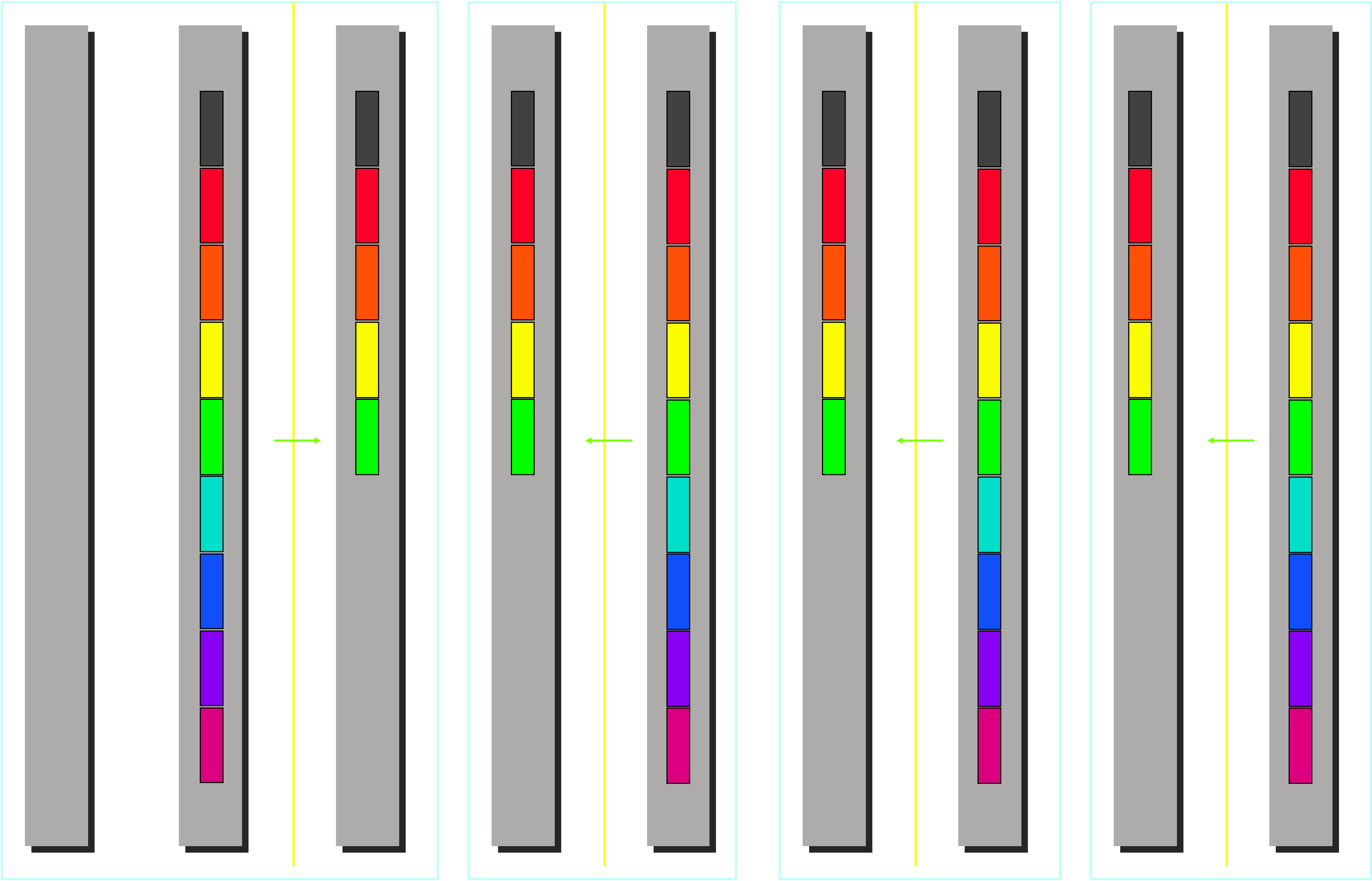


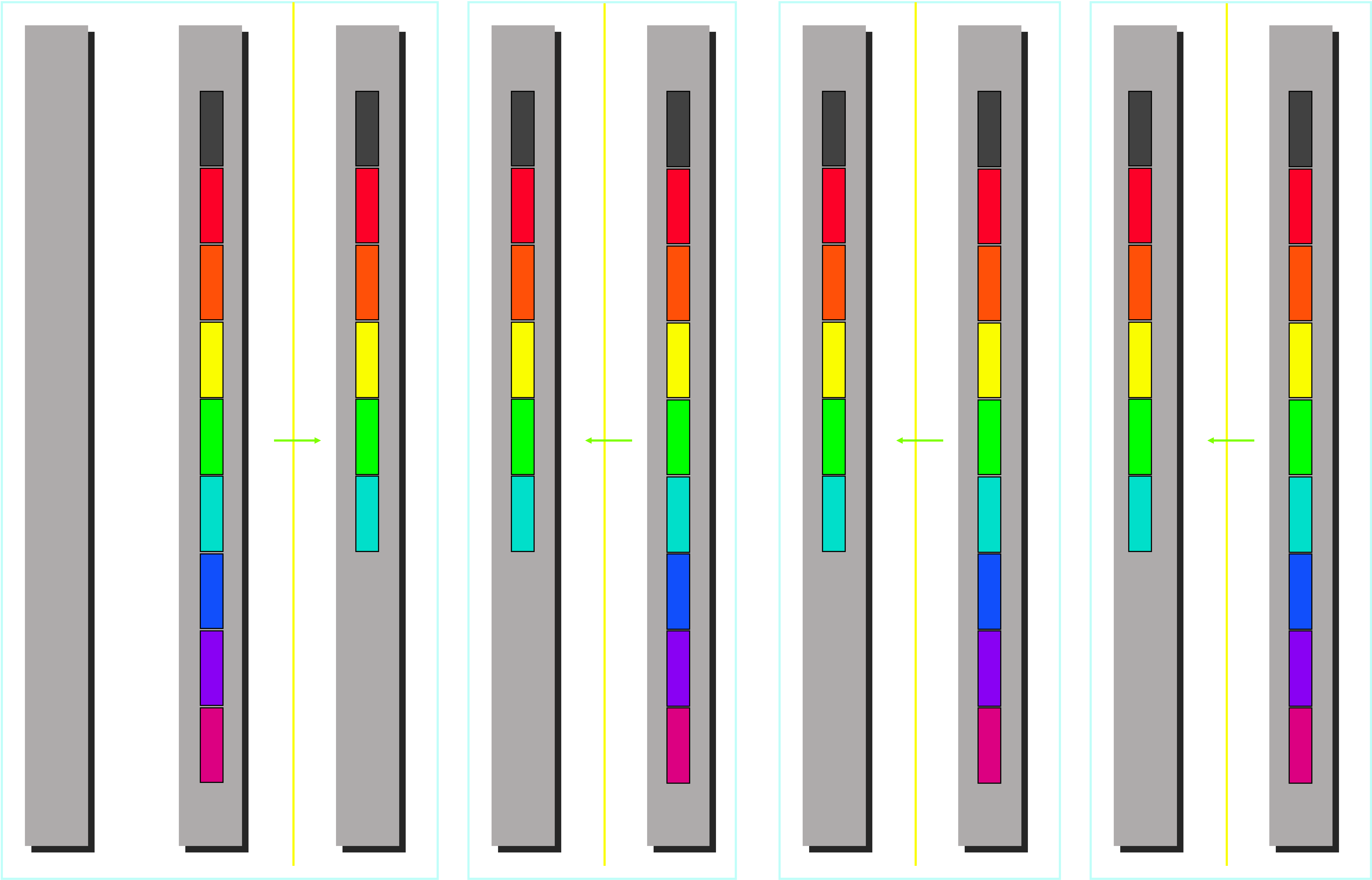


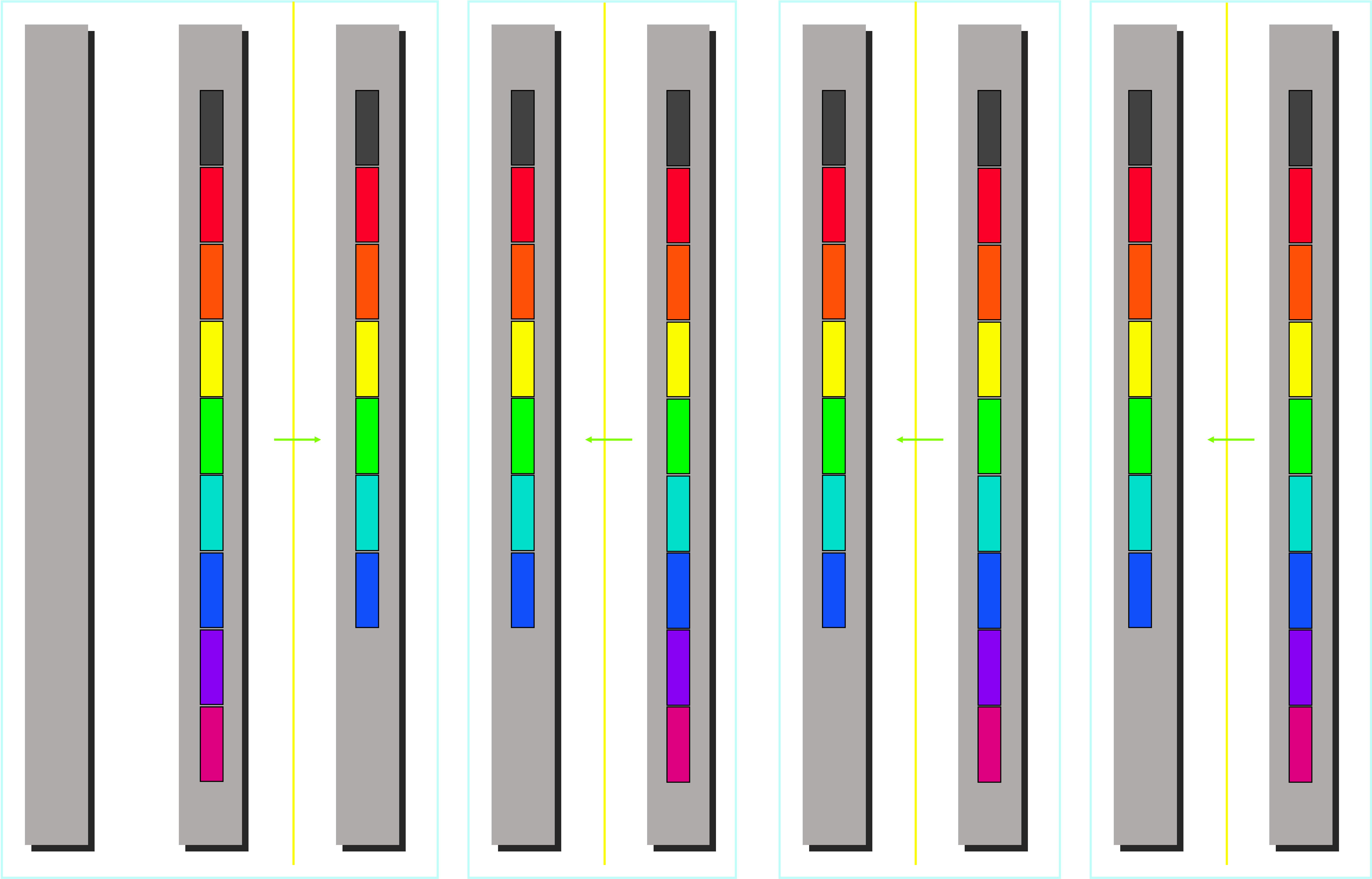


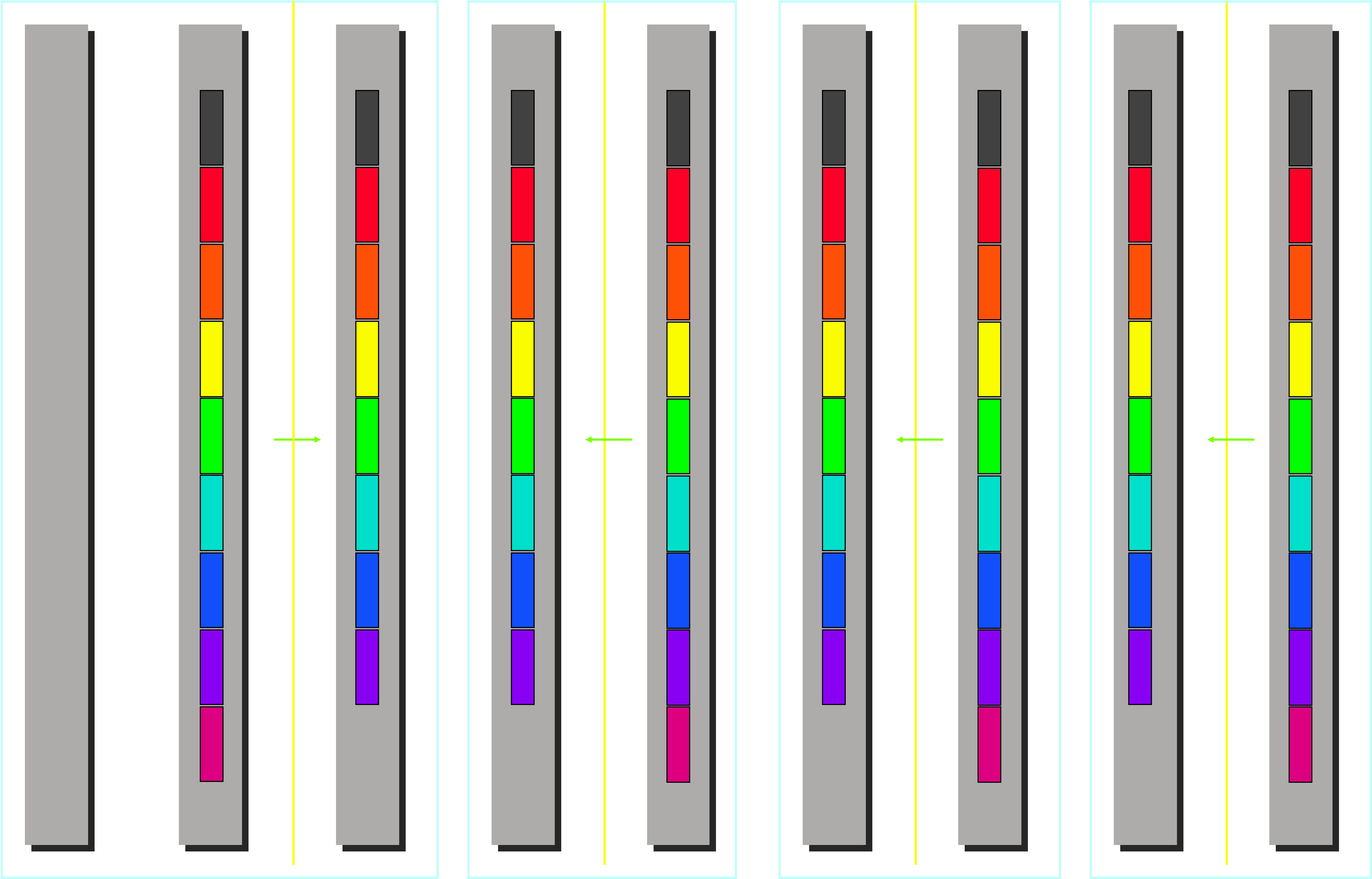


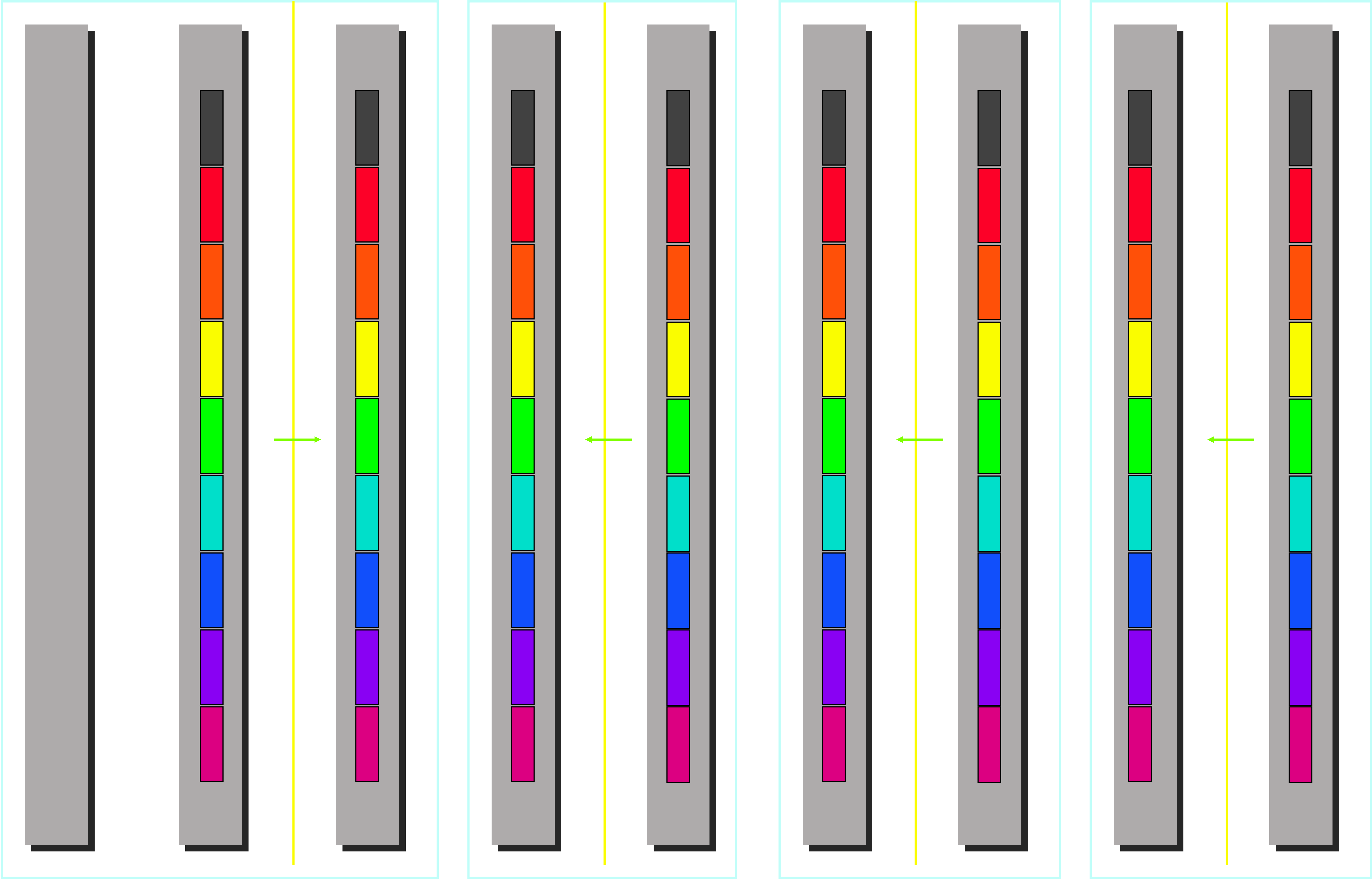


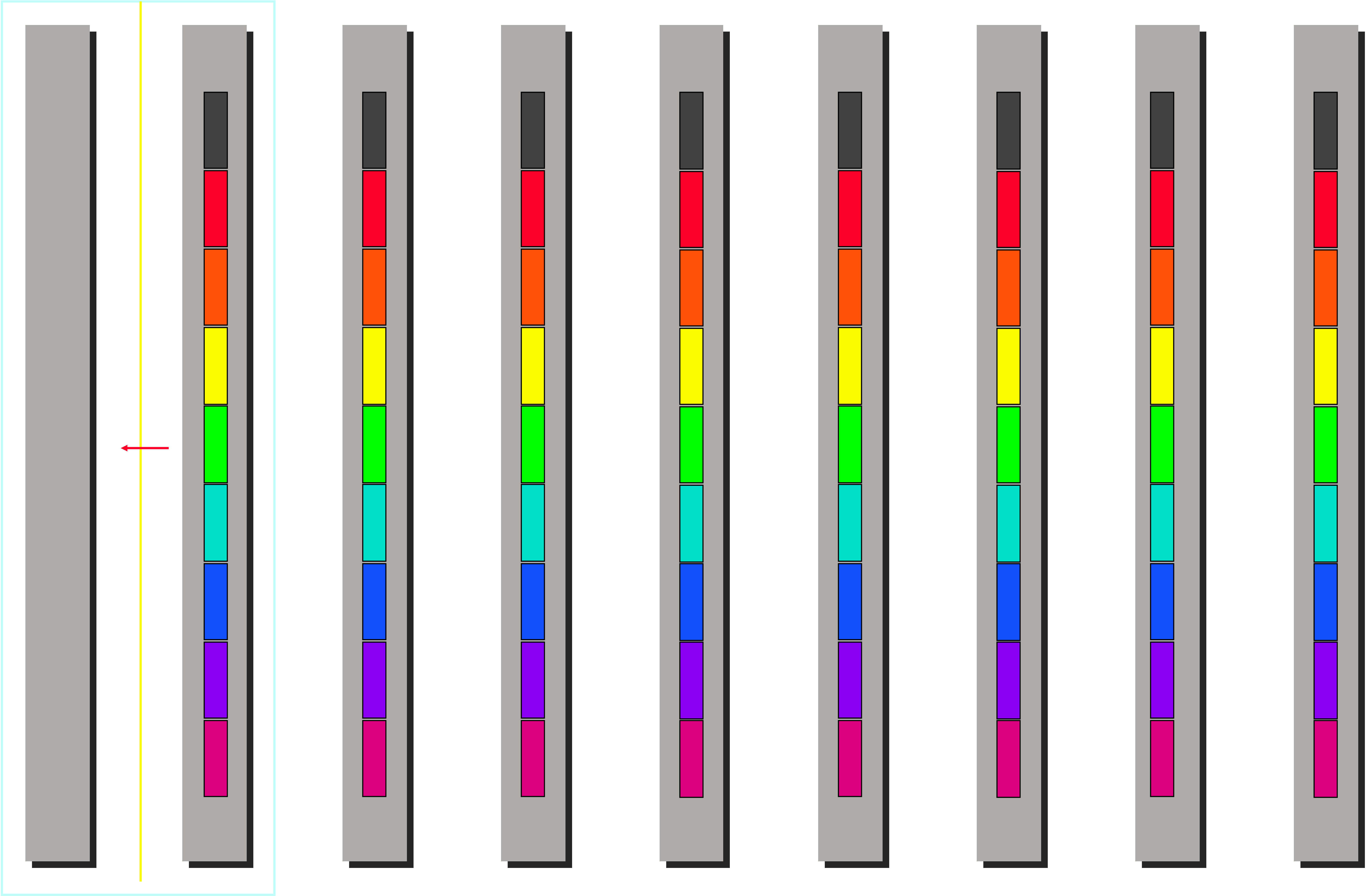


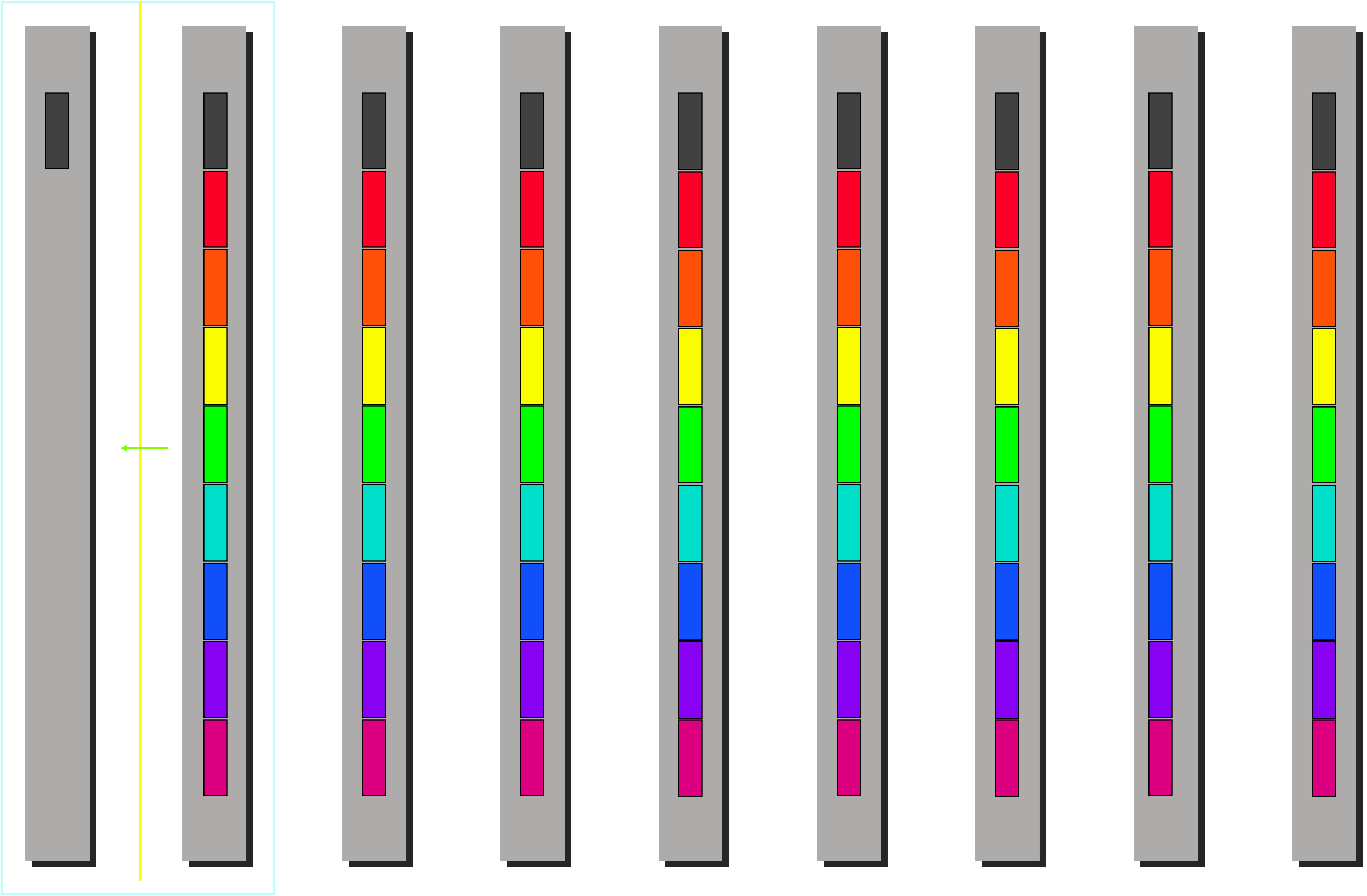


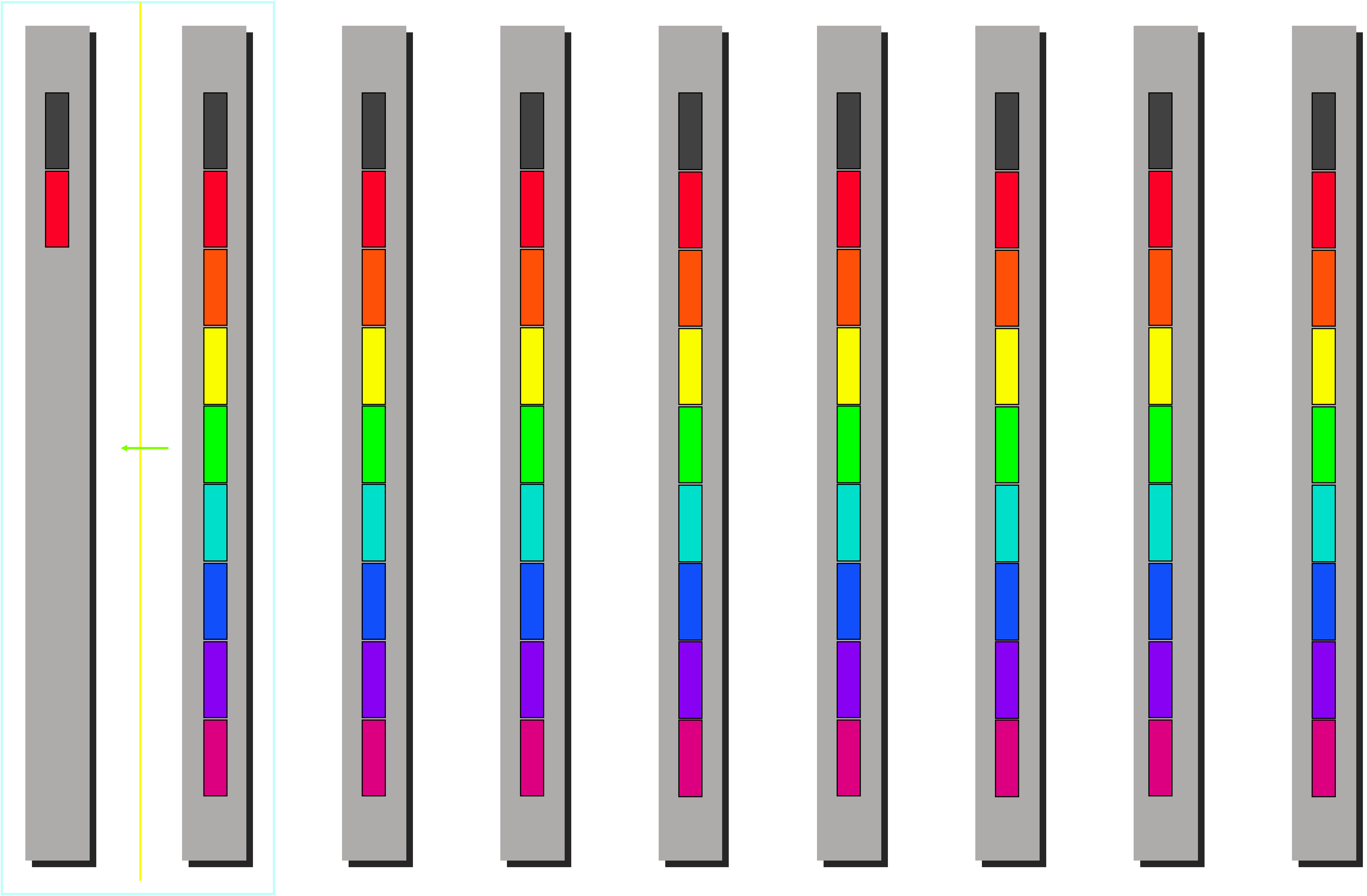


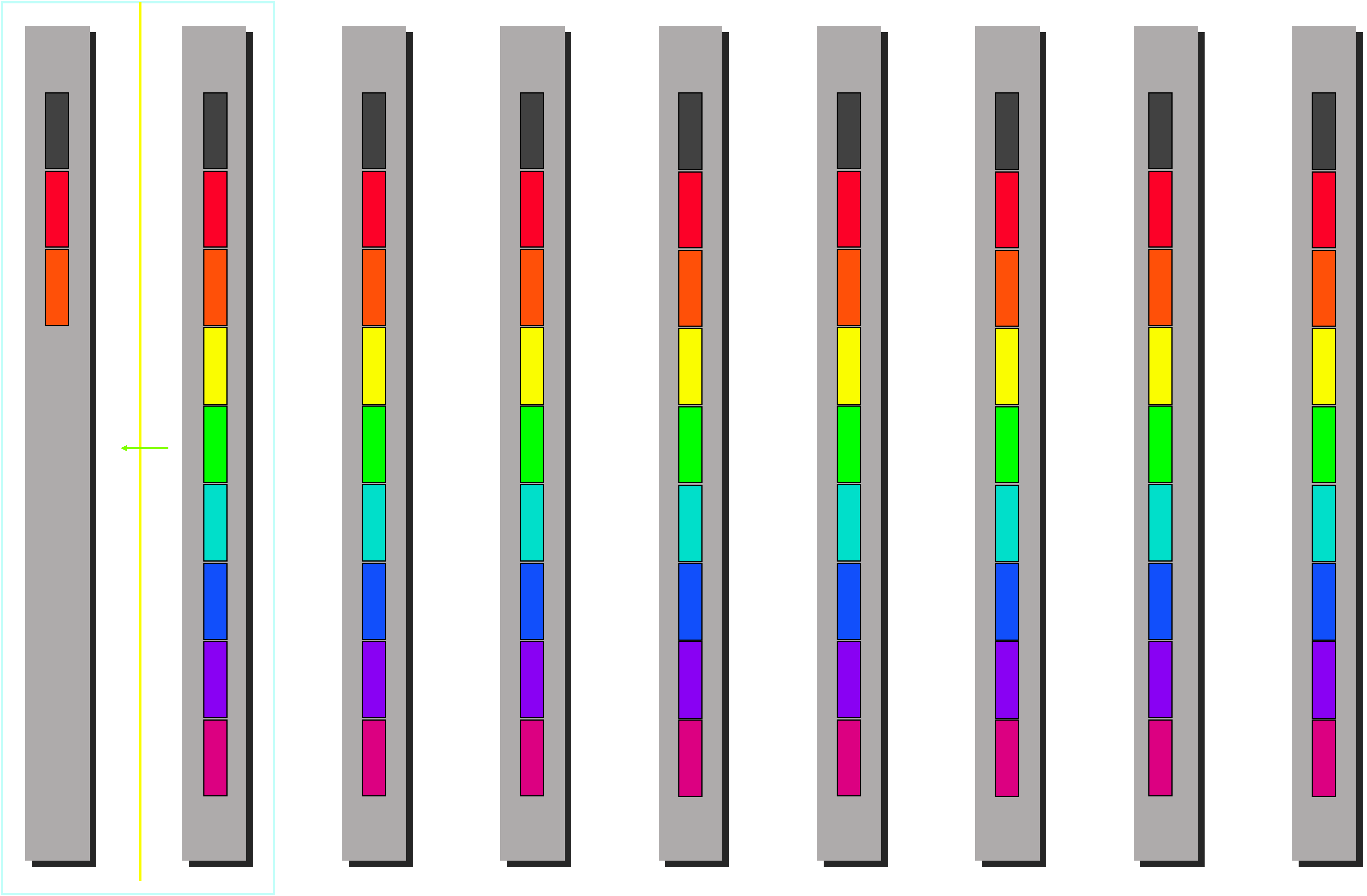


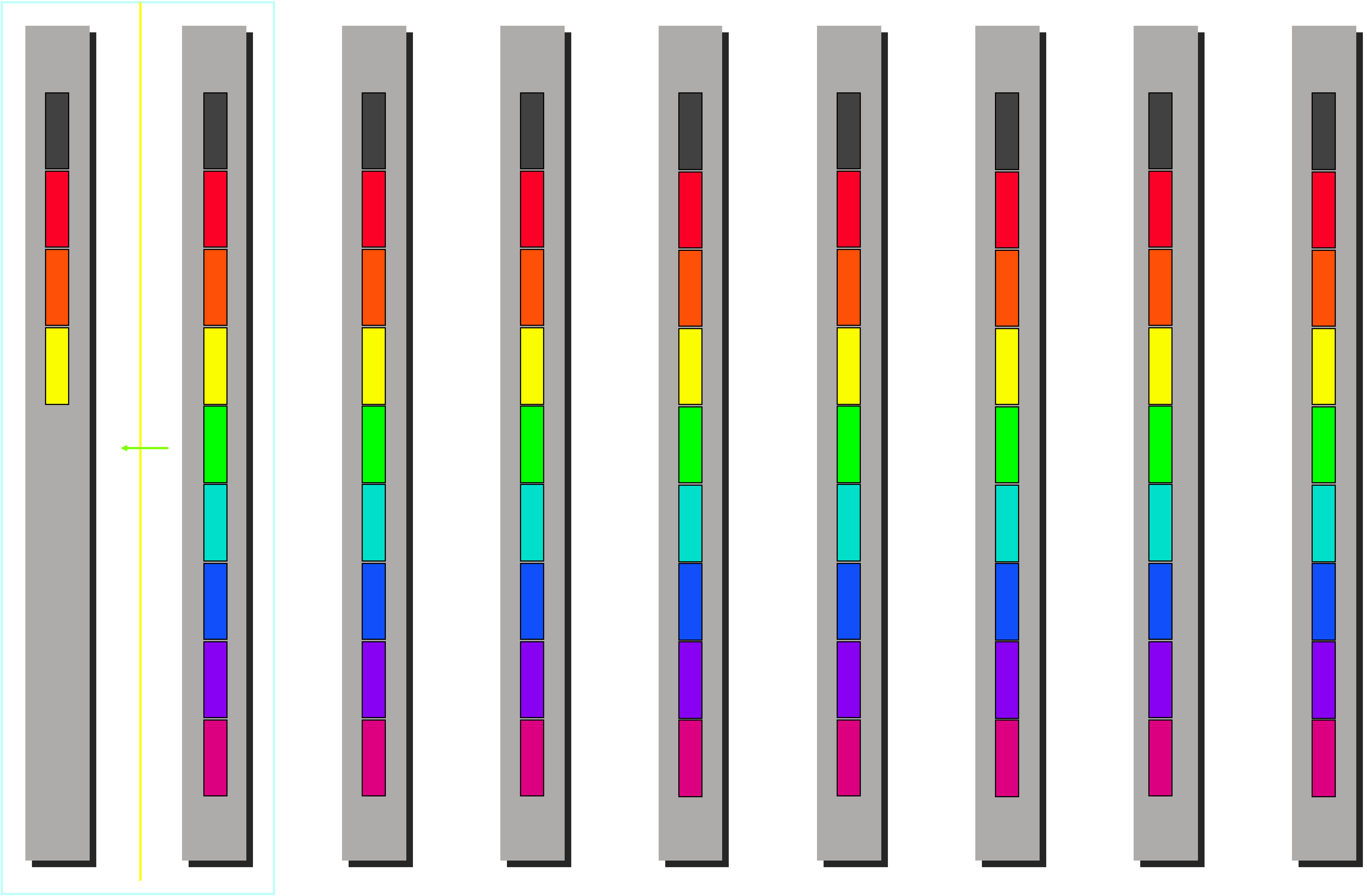


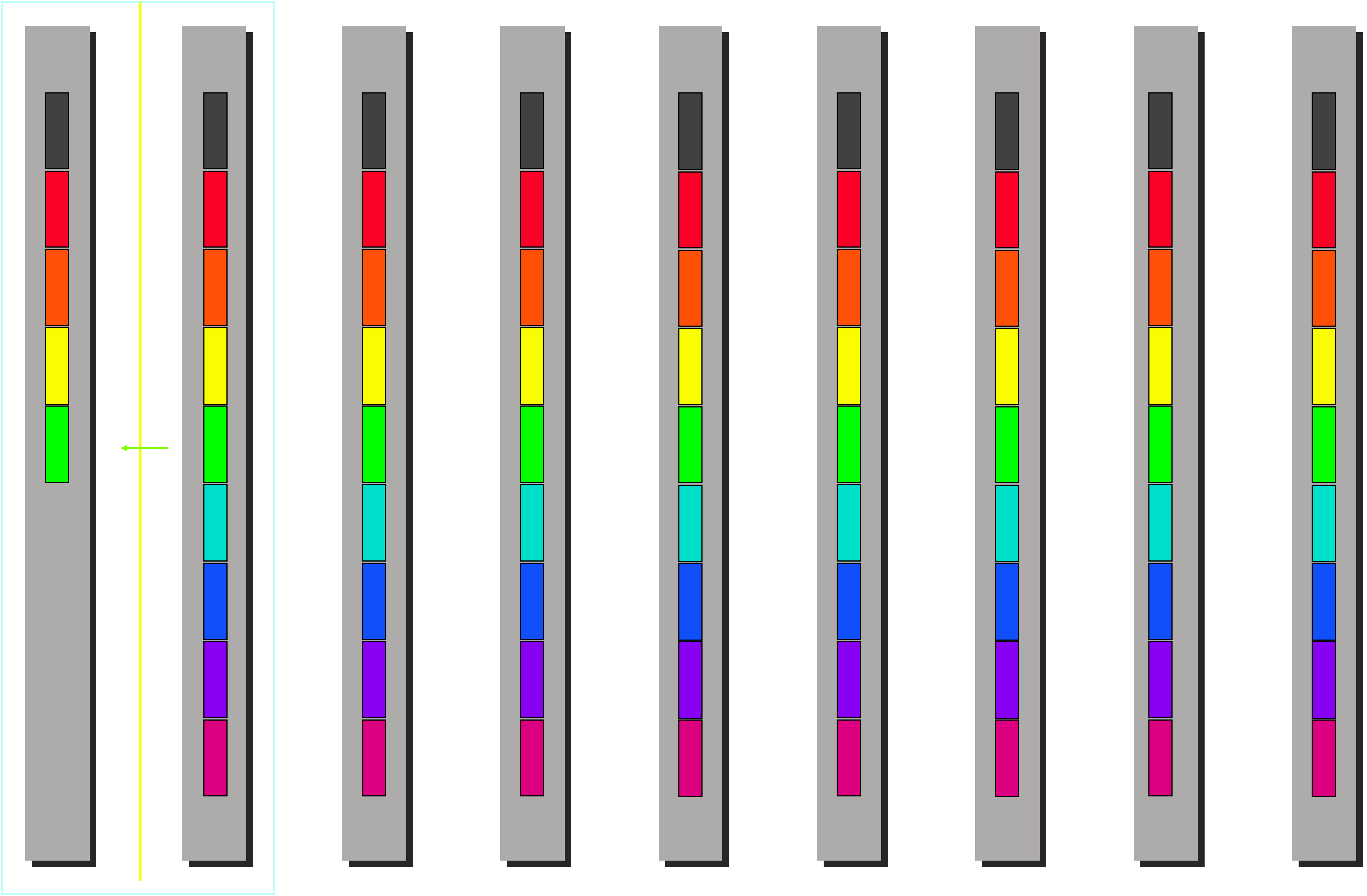


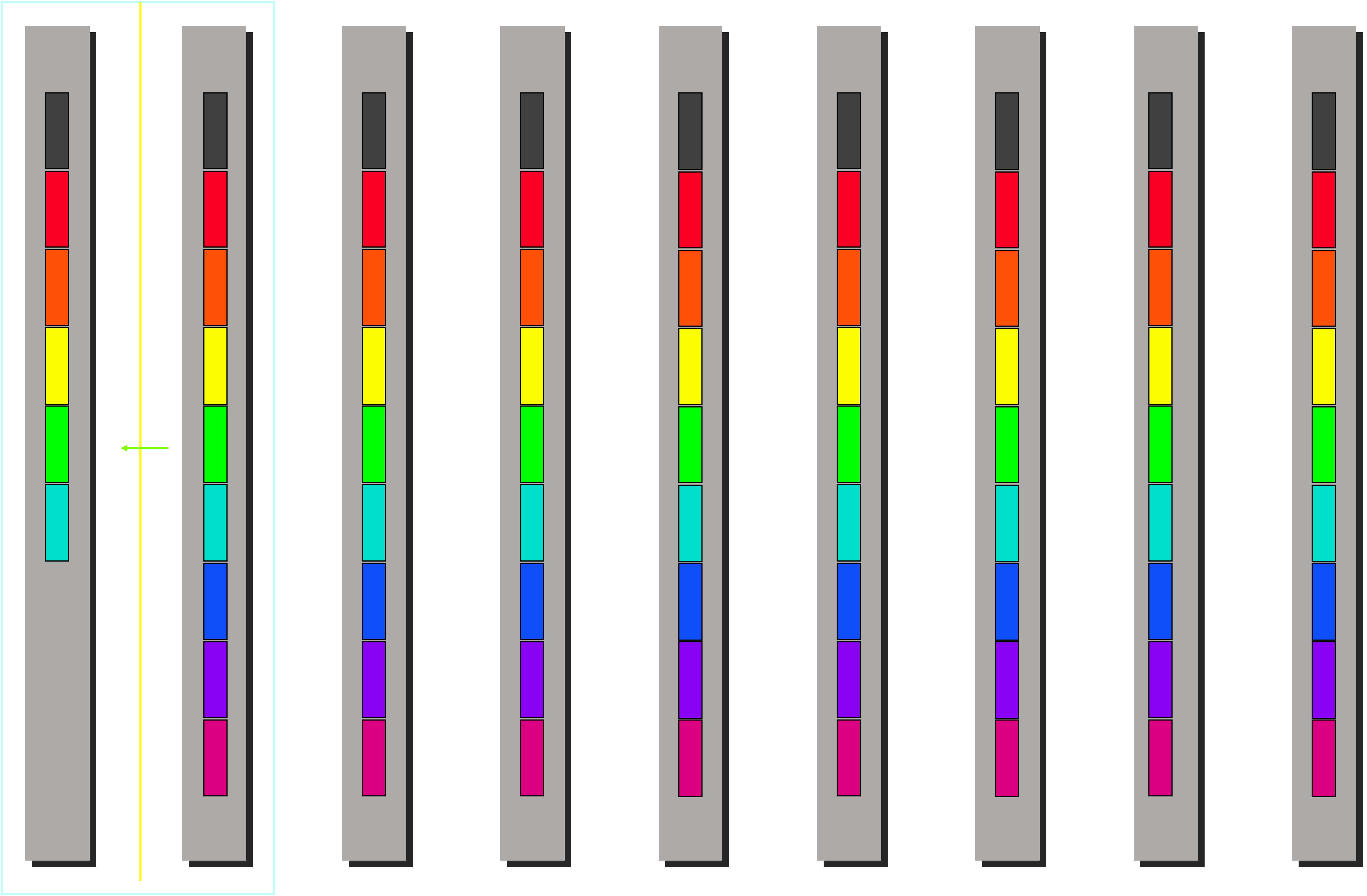


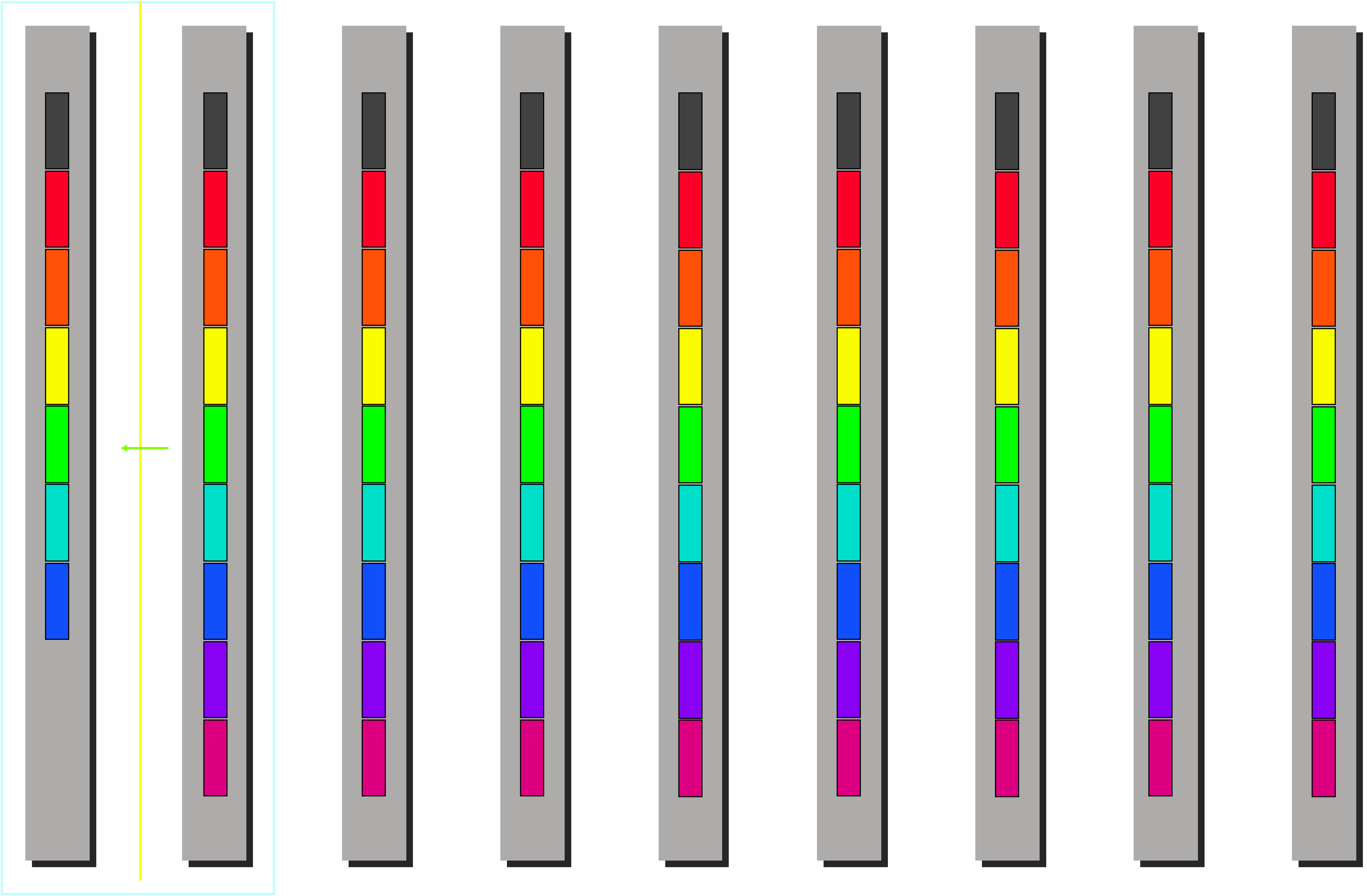


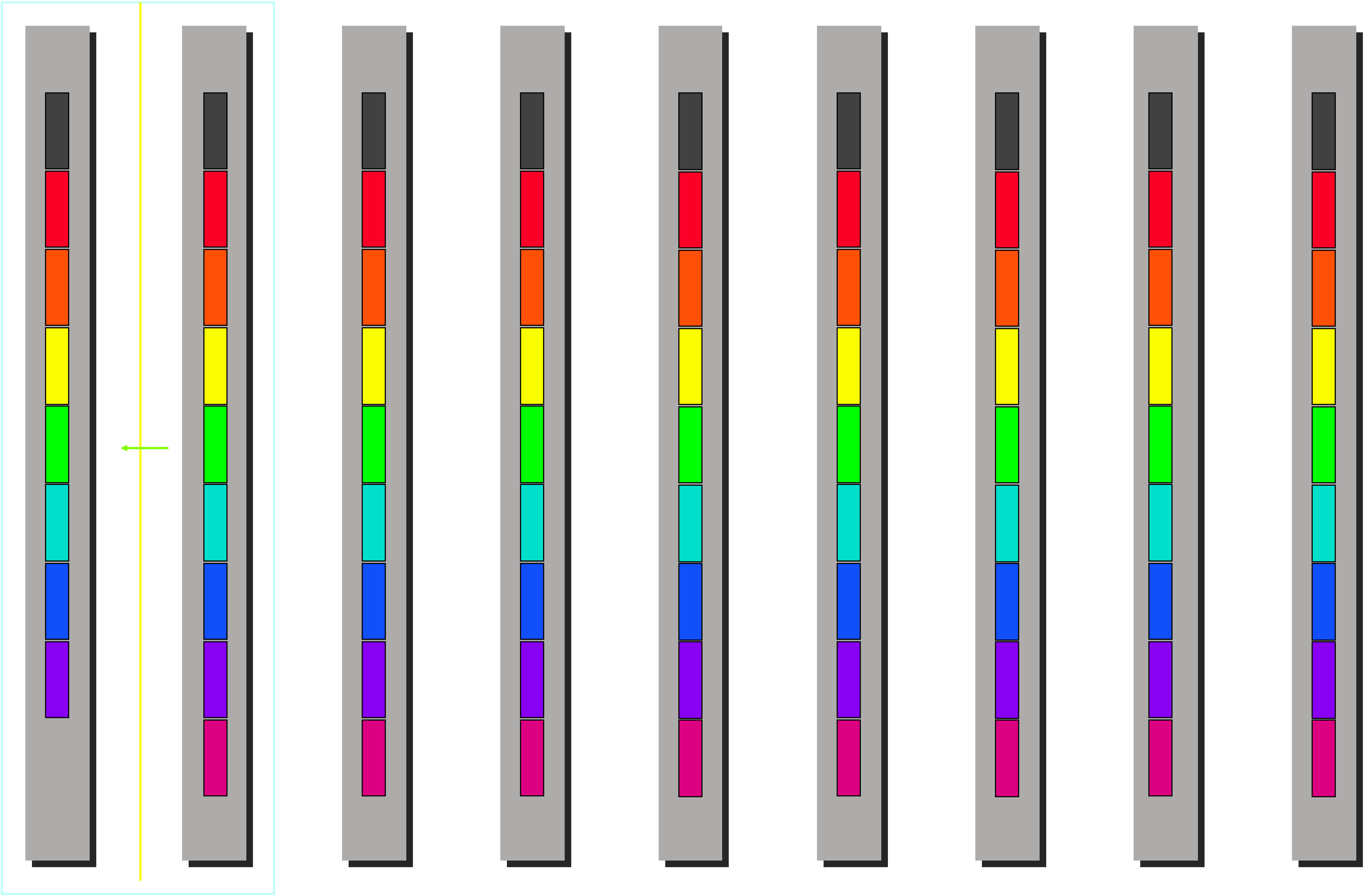


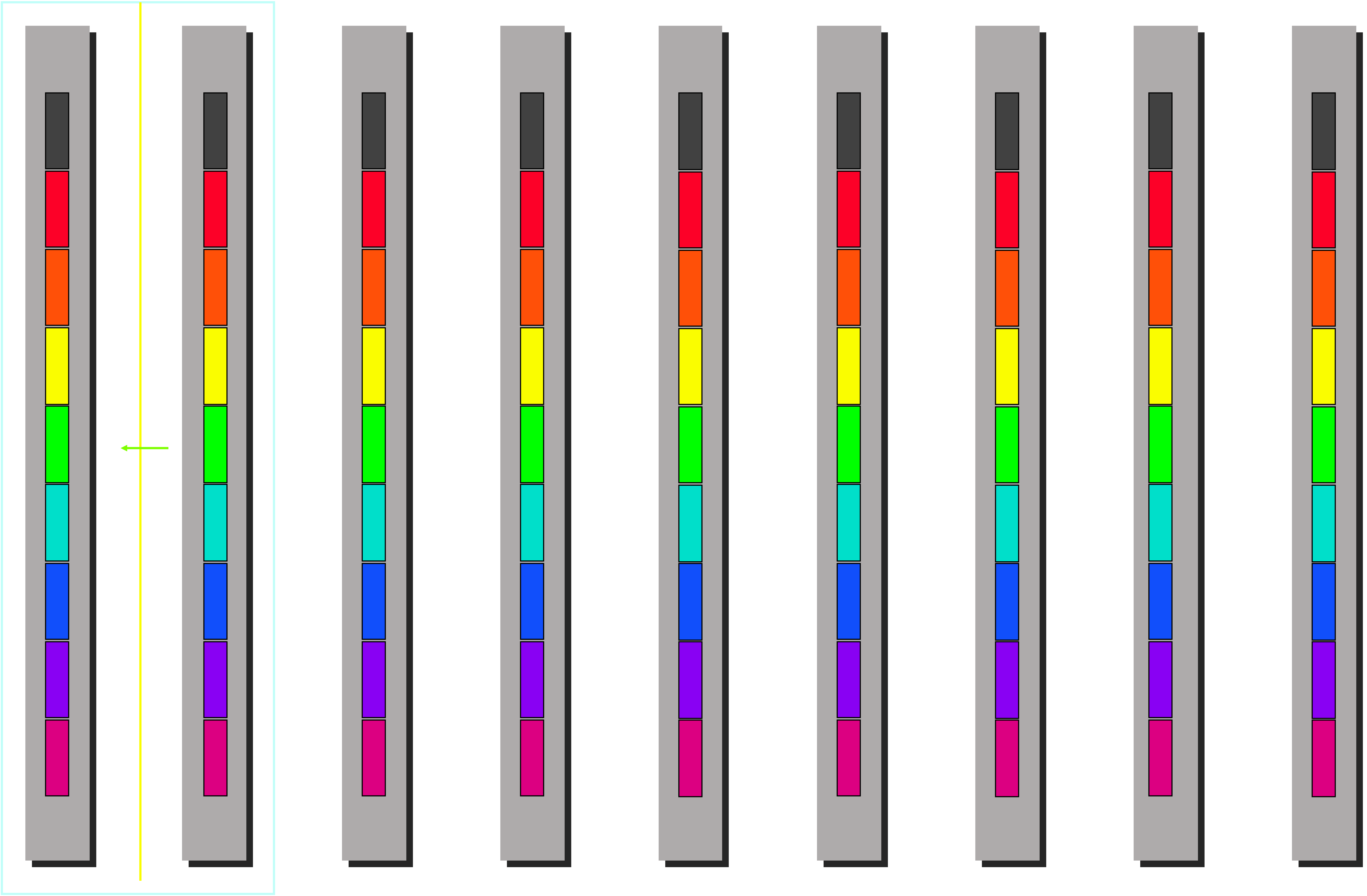


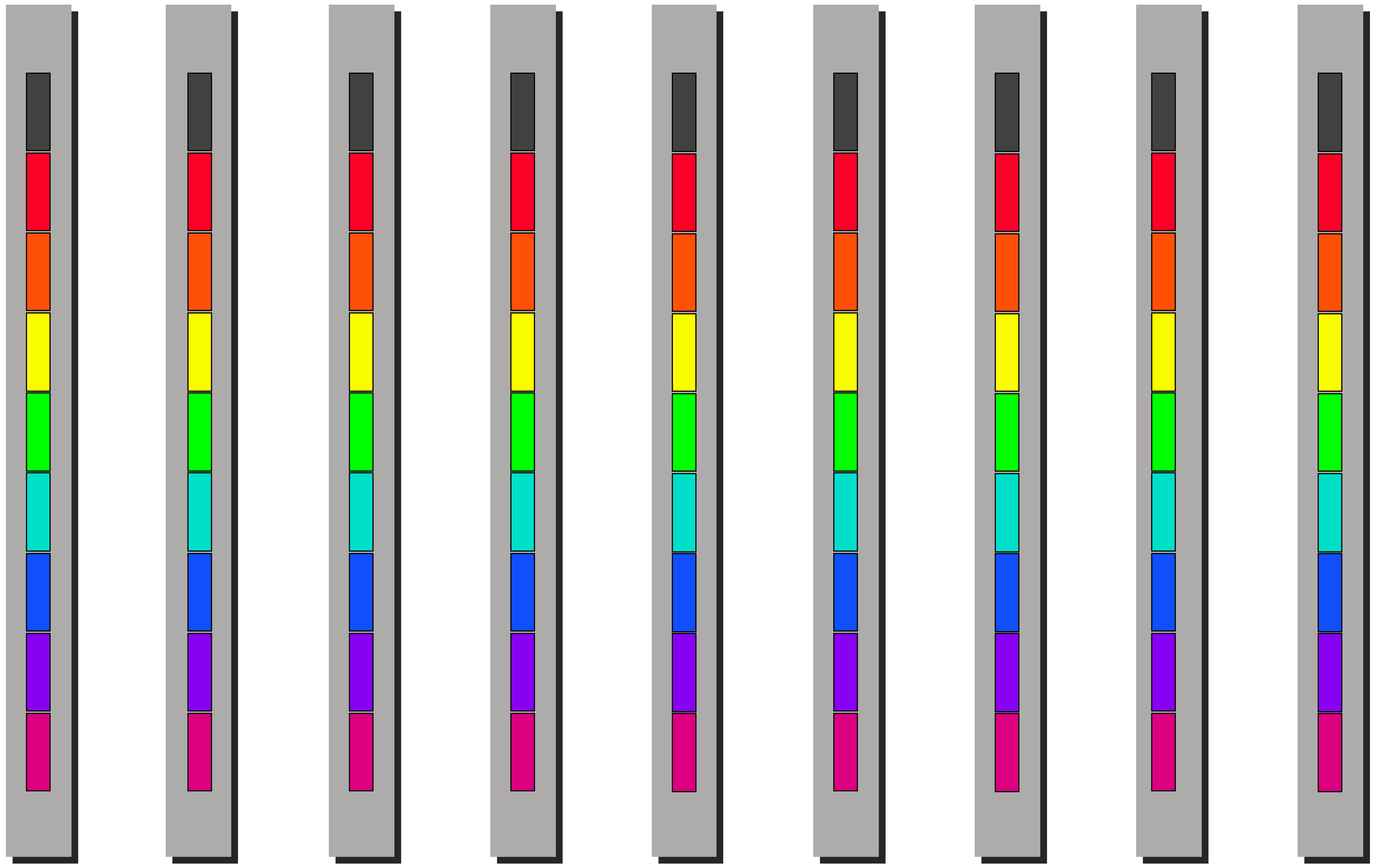












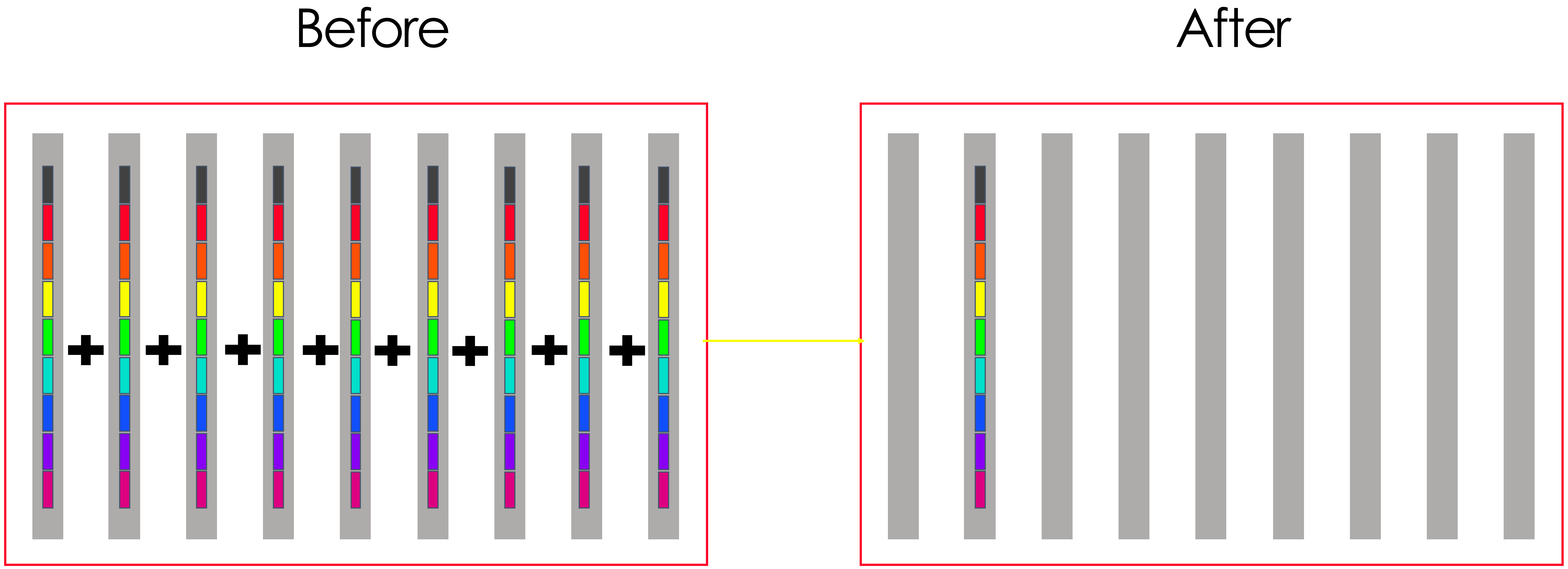
Cost of minimum spanning tree broadcast

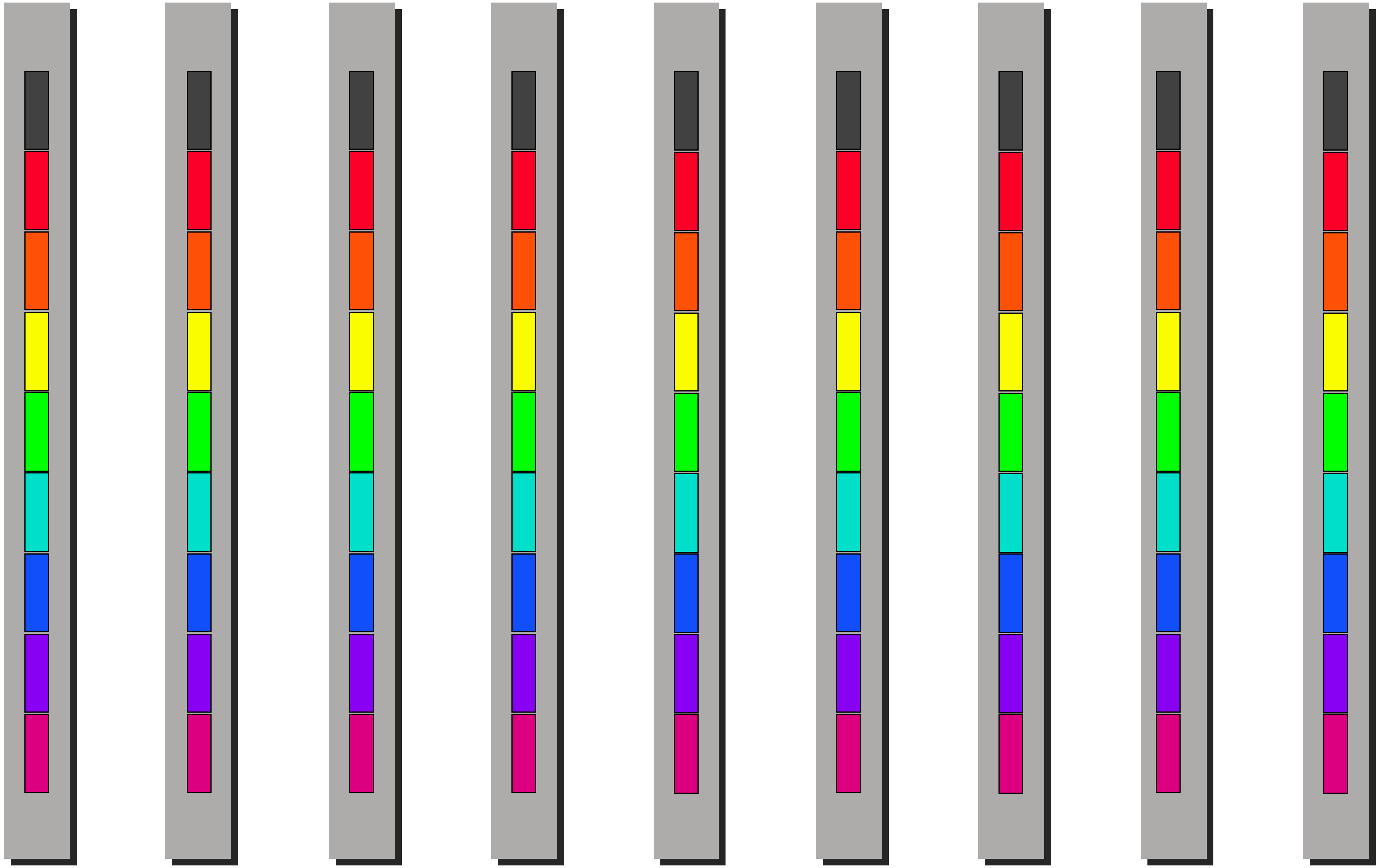
$$\boxed{\lceil log(p) \rceil}\ \boxed{(\alpha + n\beta)}$$

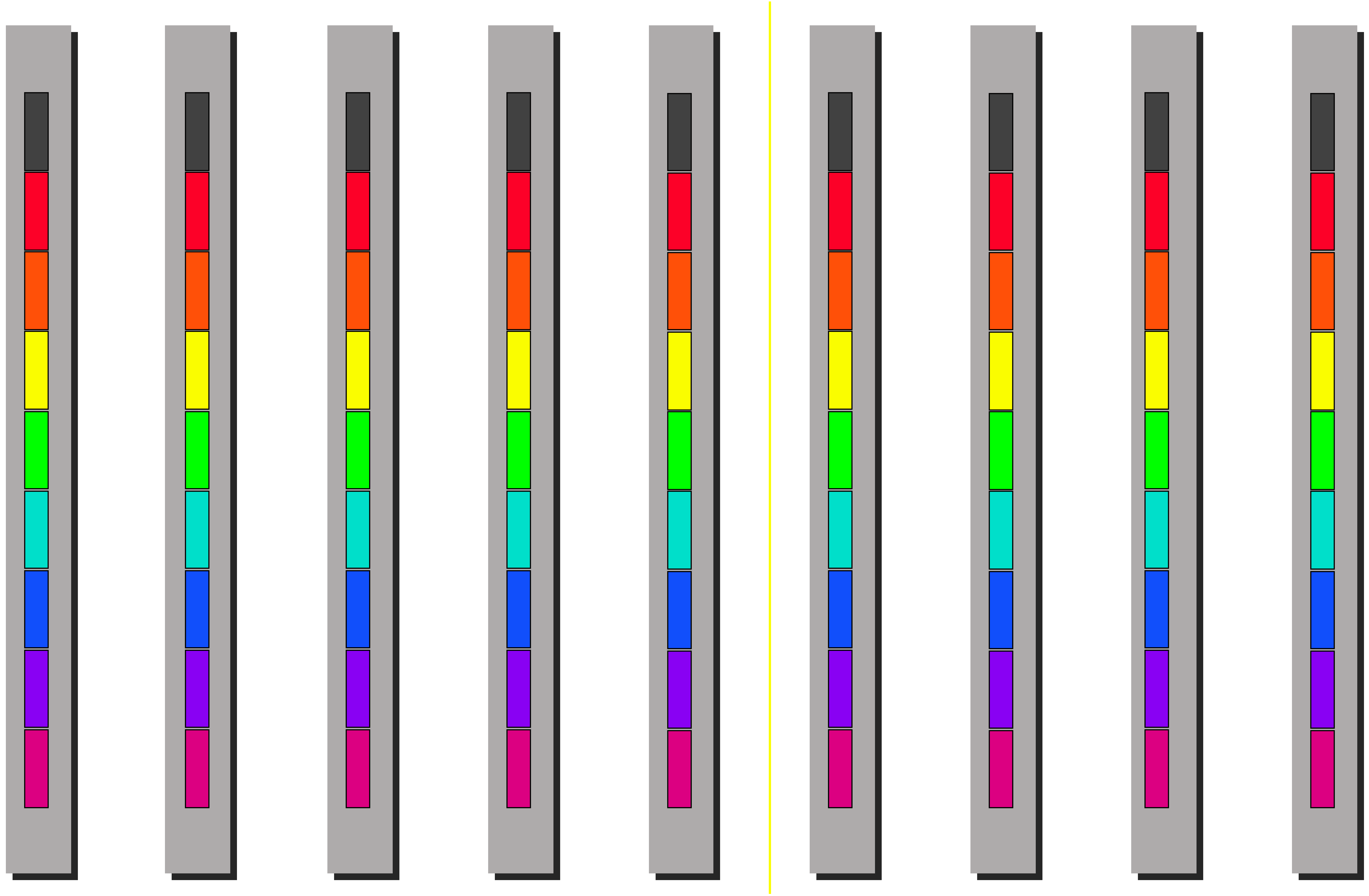
number of steps

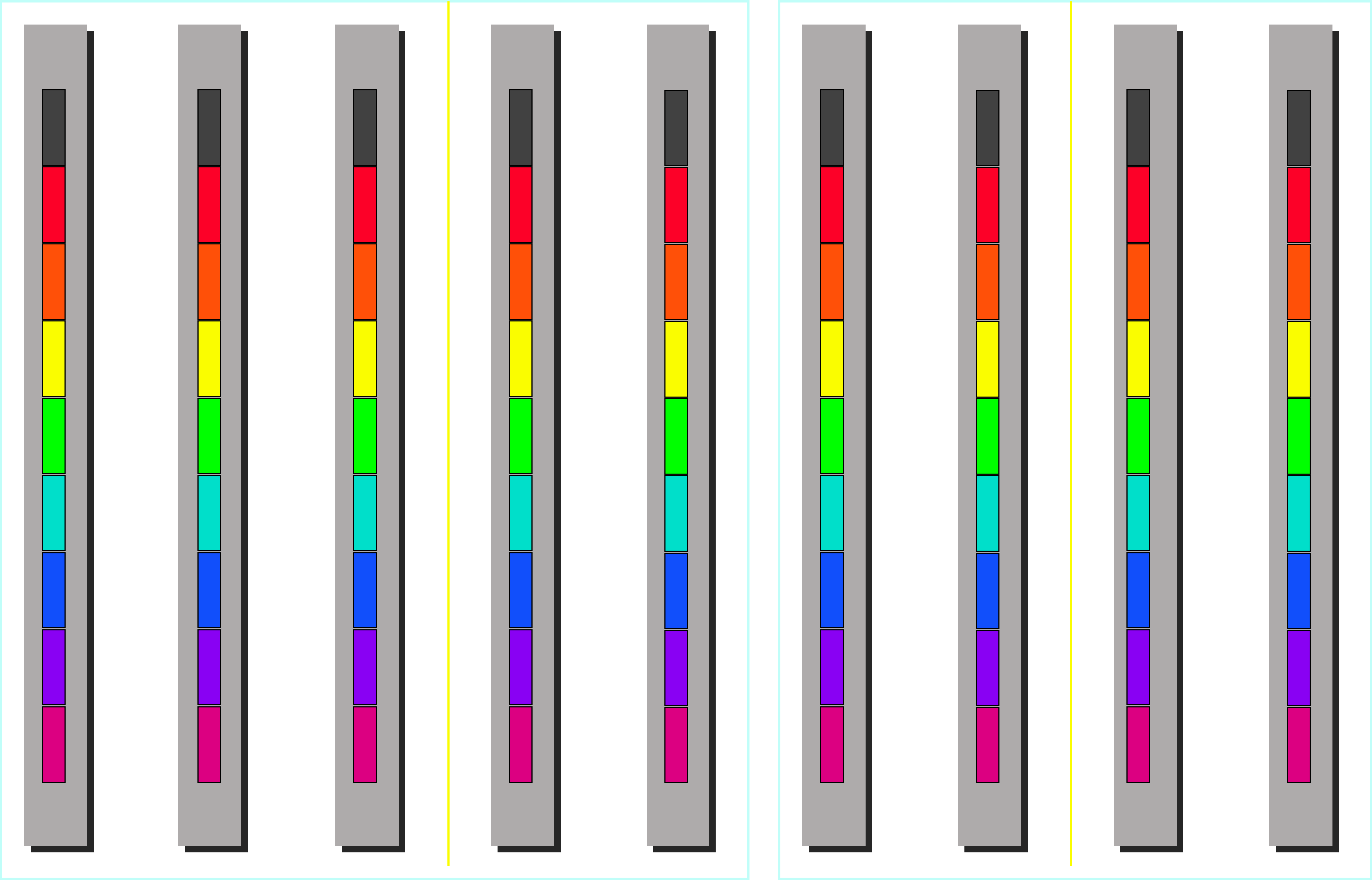
cost per steps

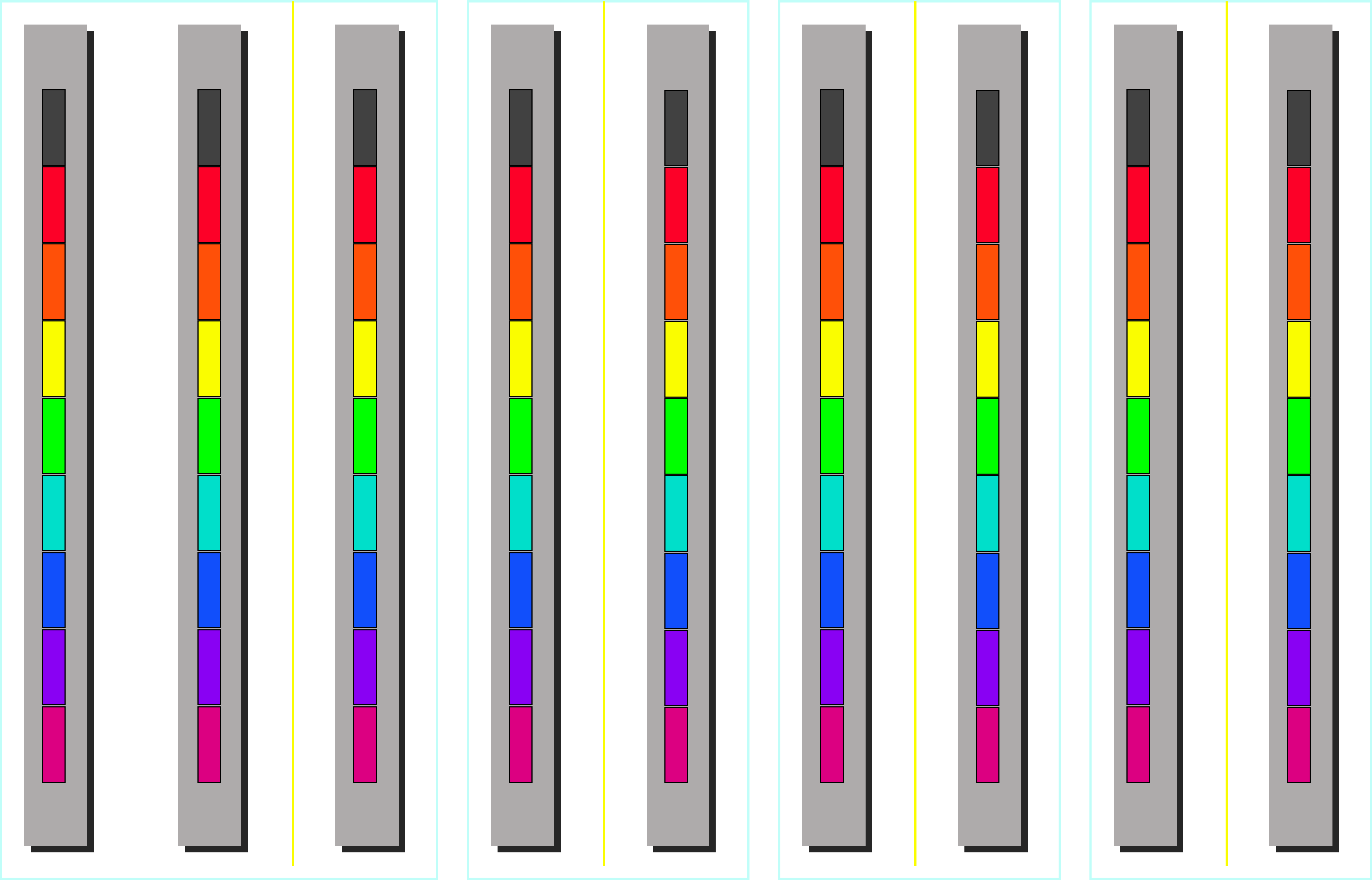
Reduce(-to-one)

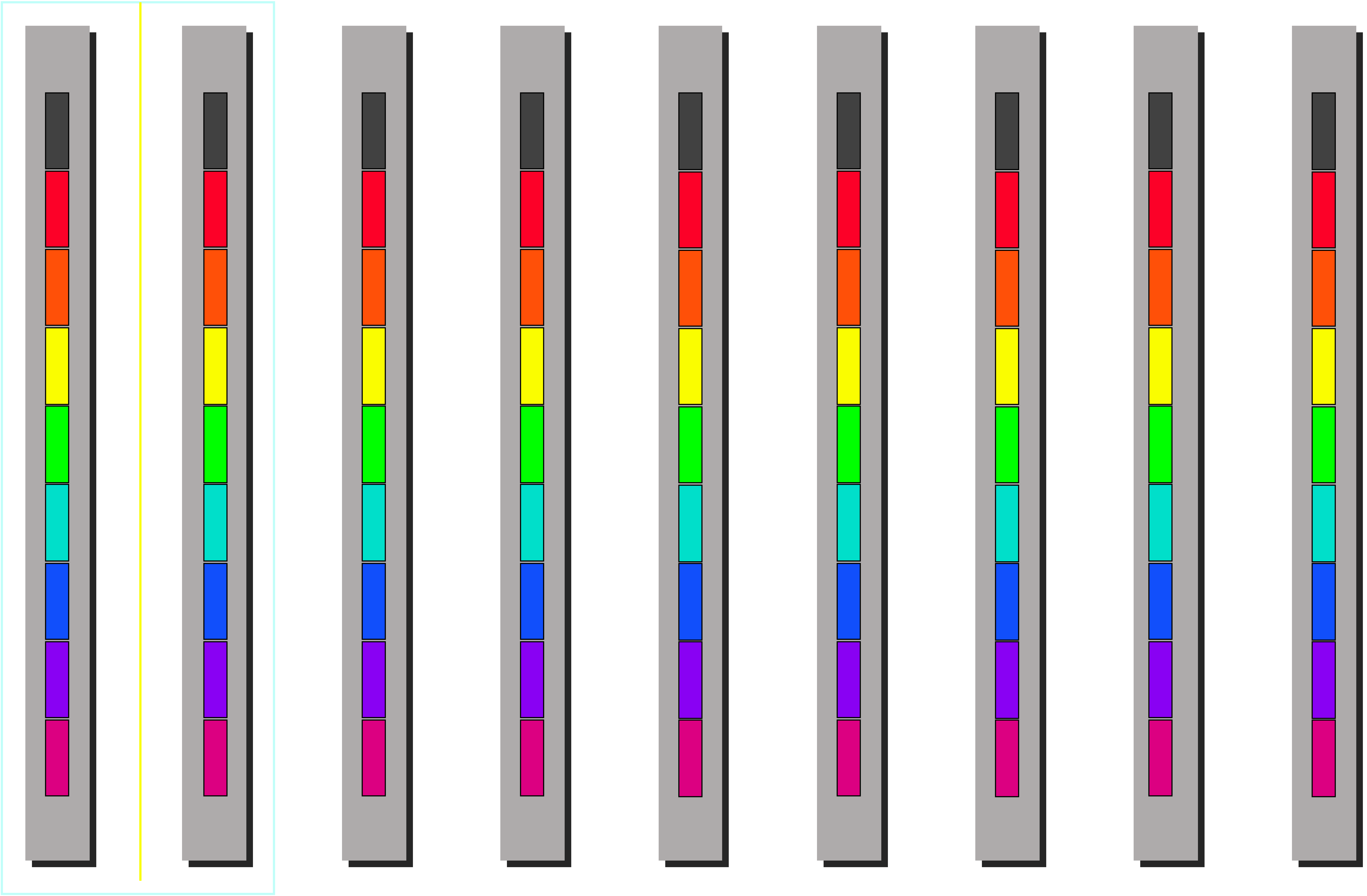


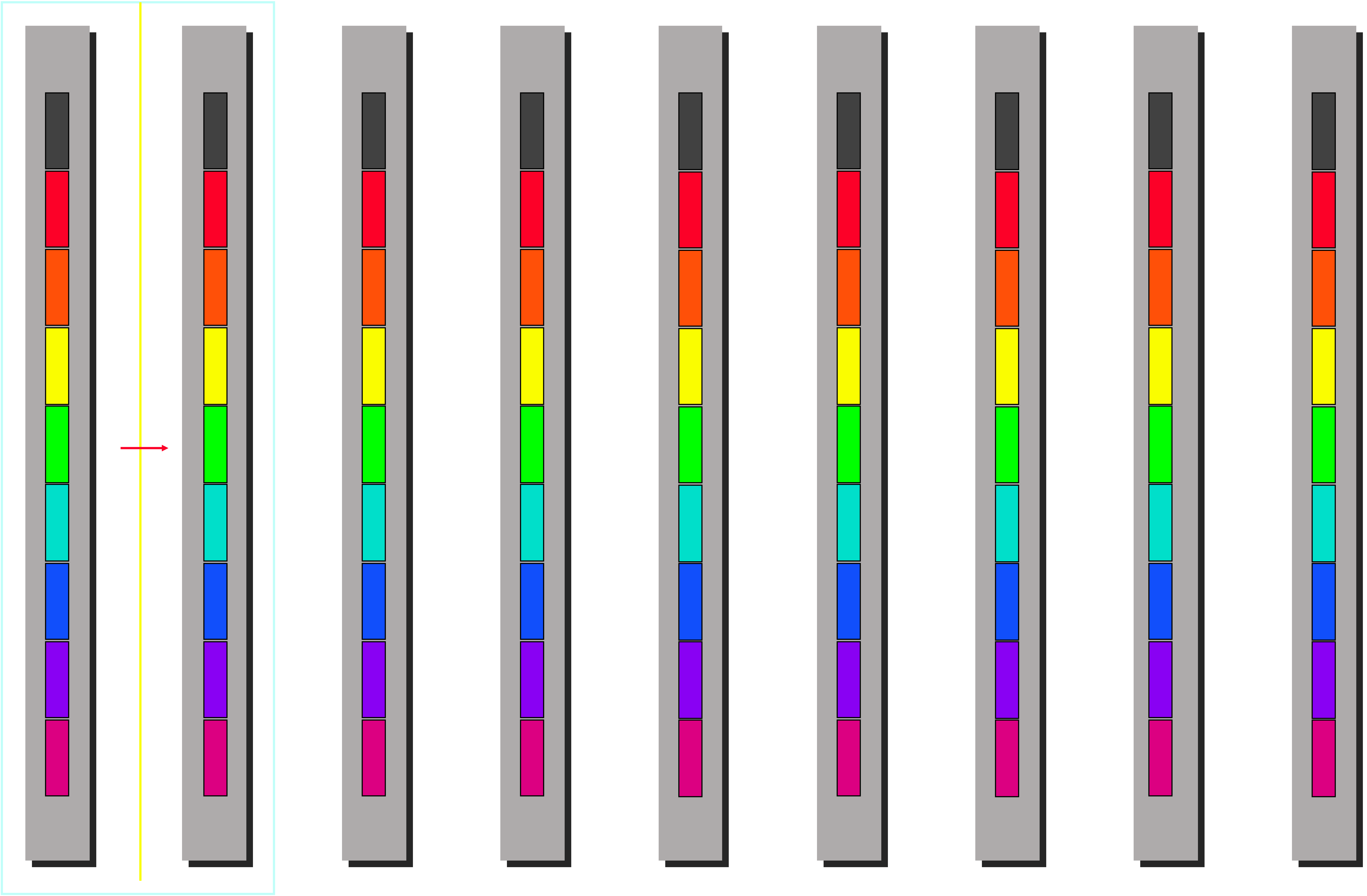


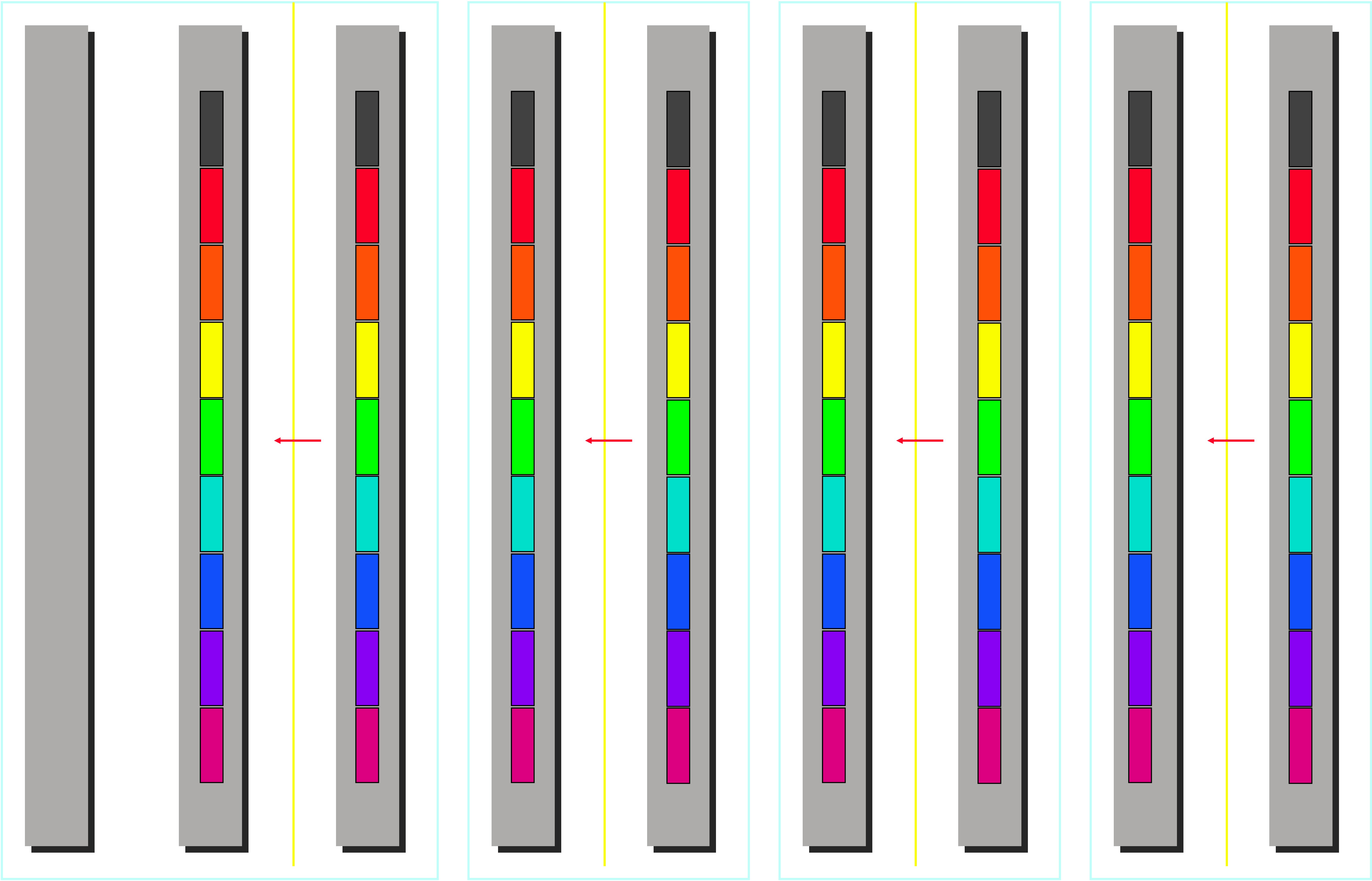


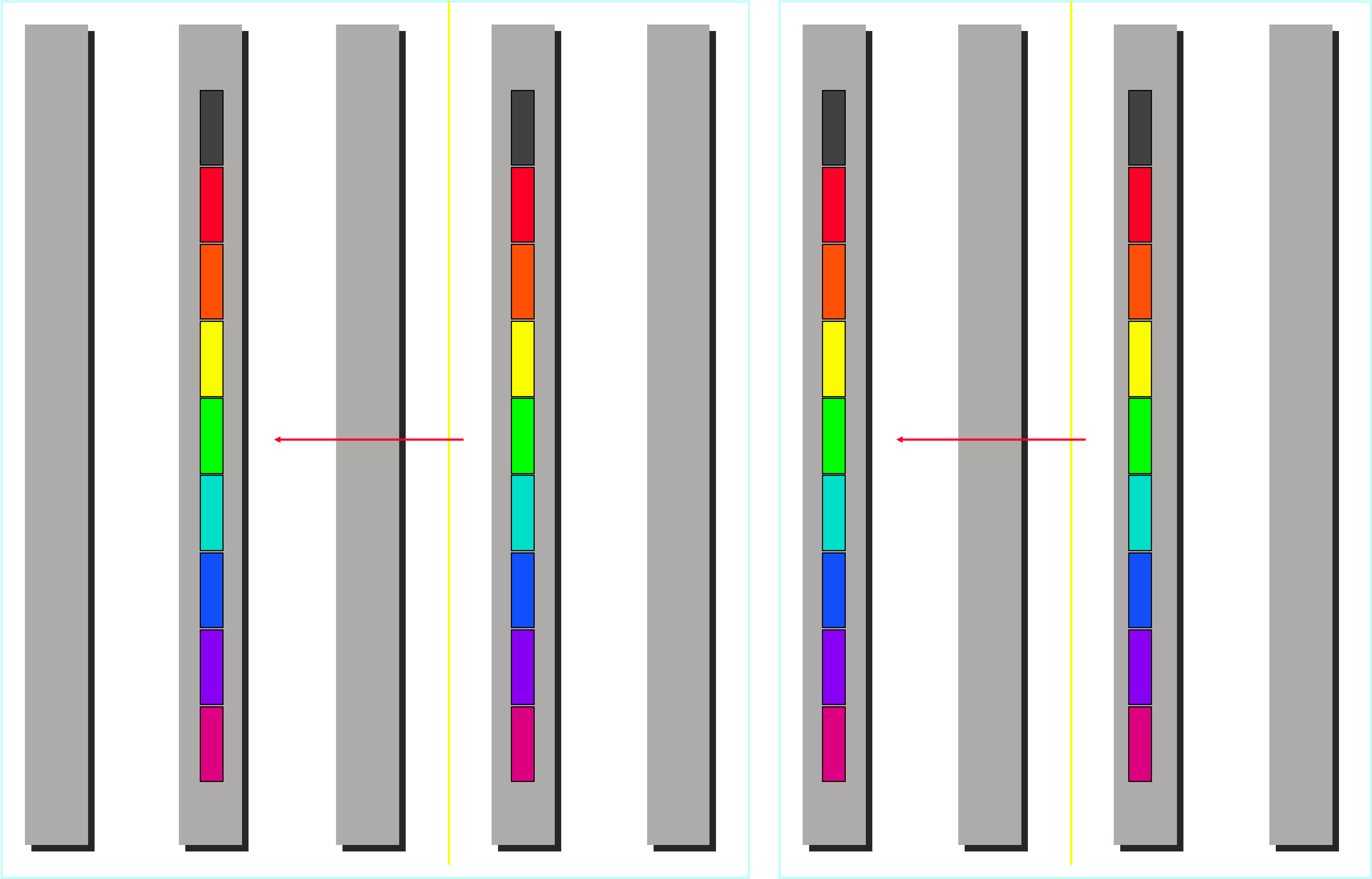


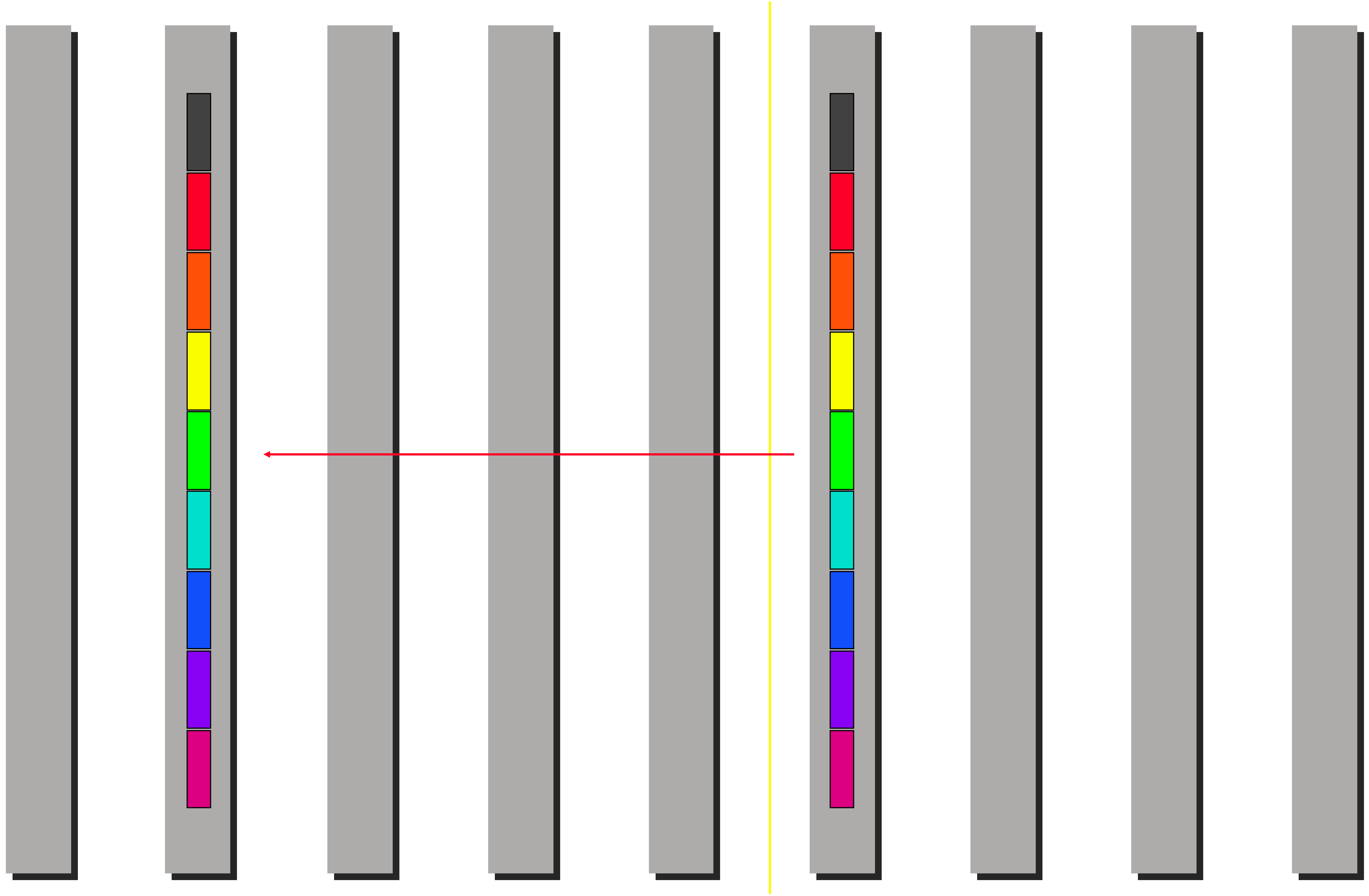








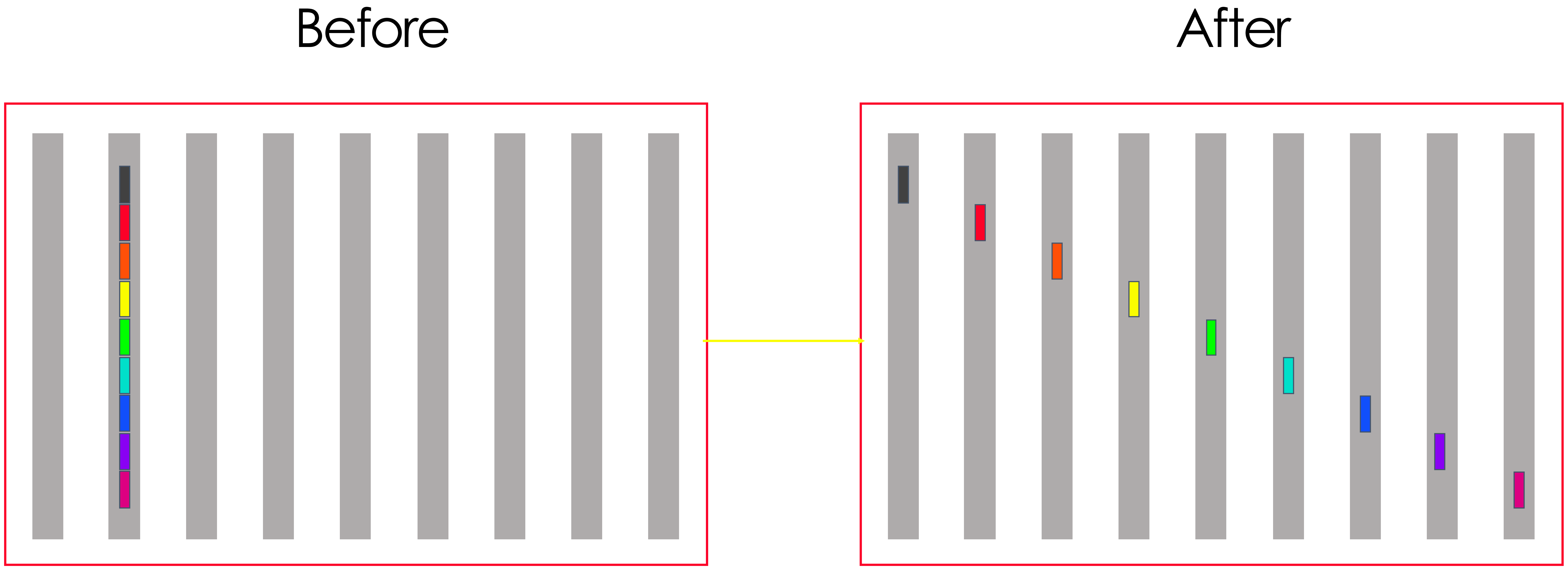


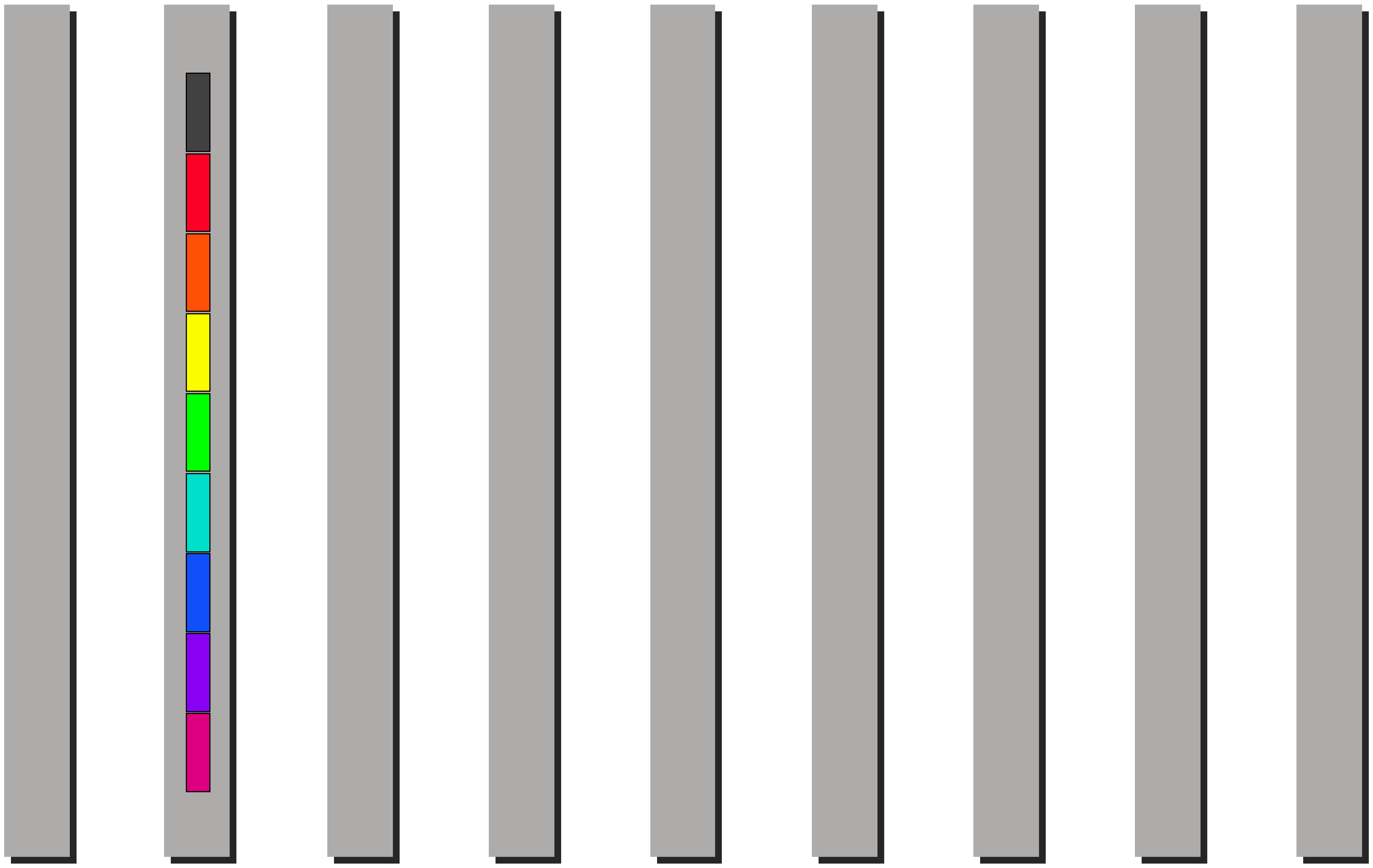


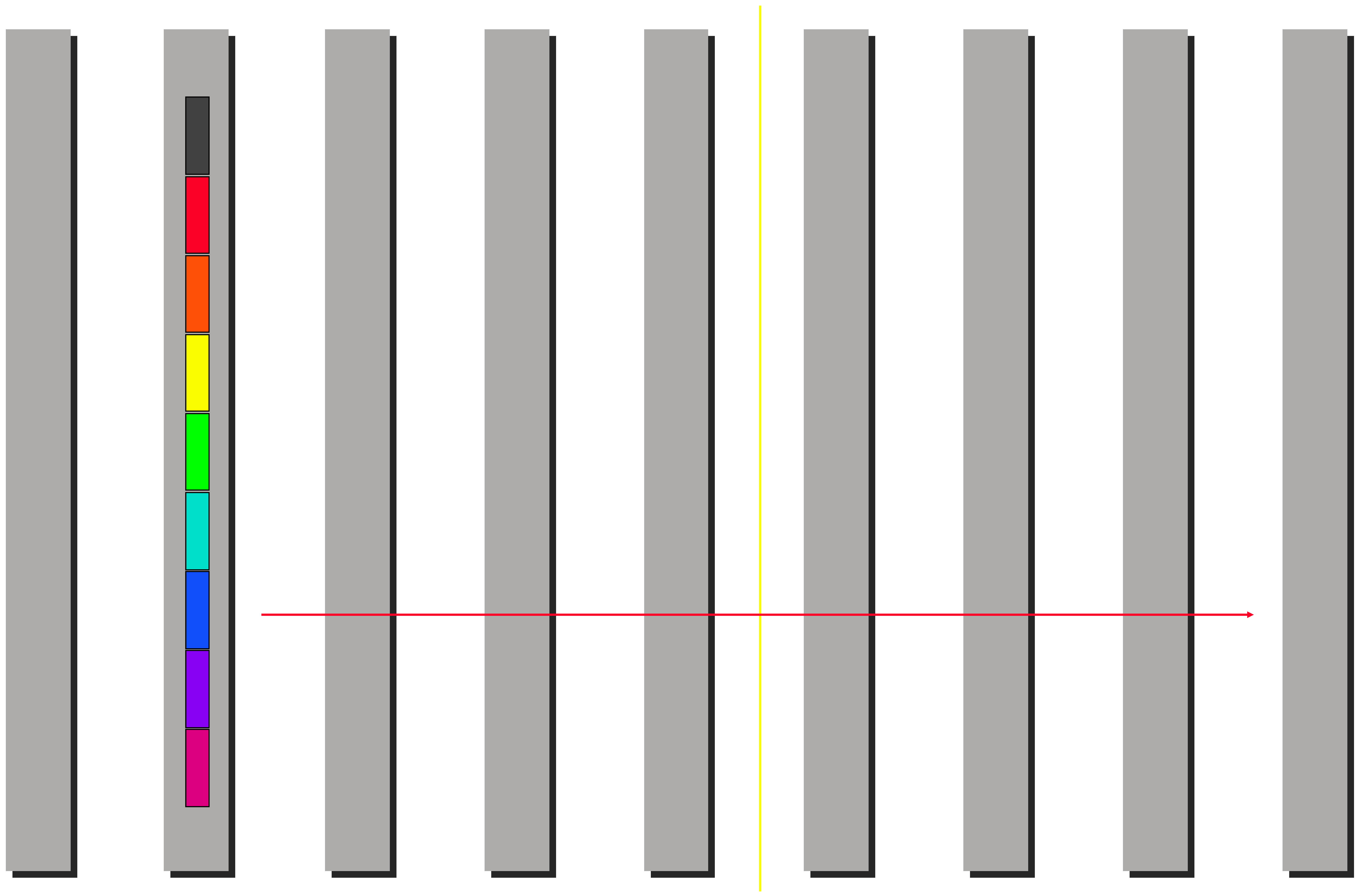
Cost of minimum spanning tree reduce(-to-one)

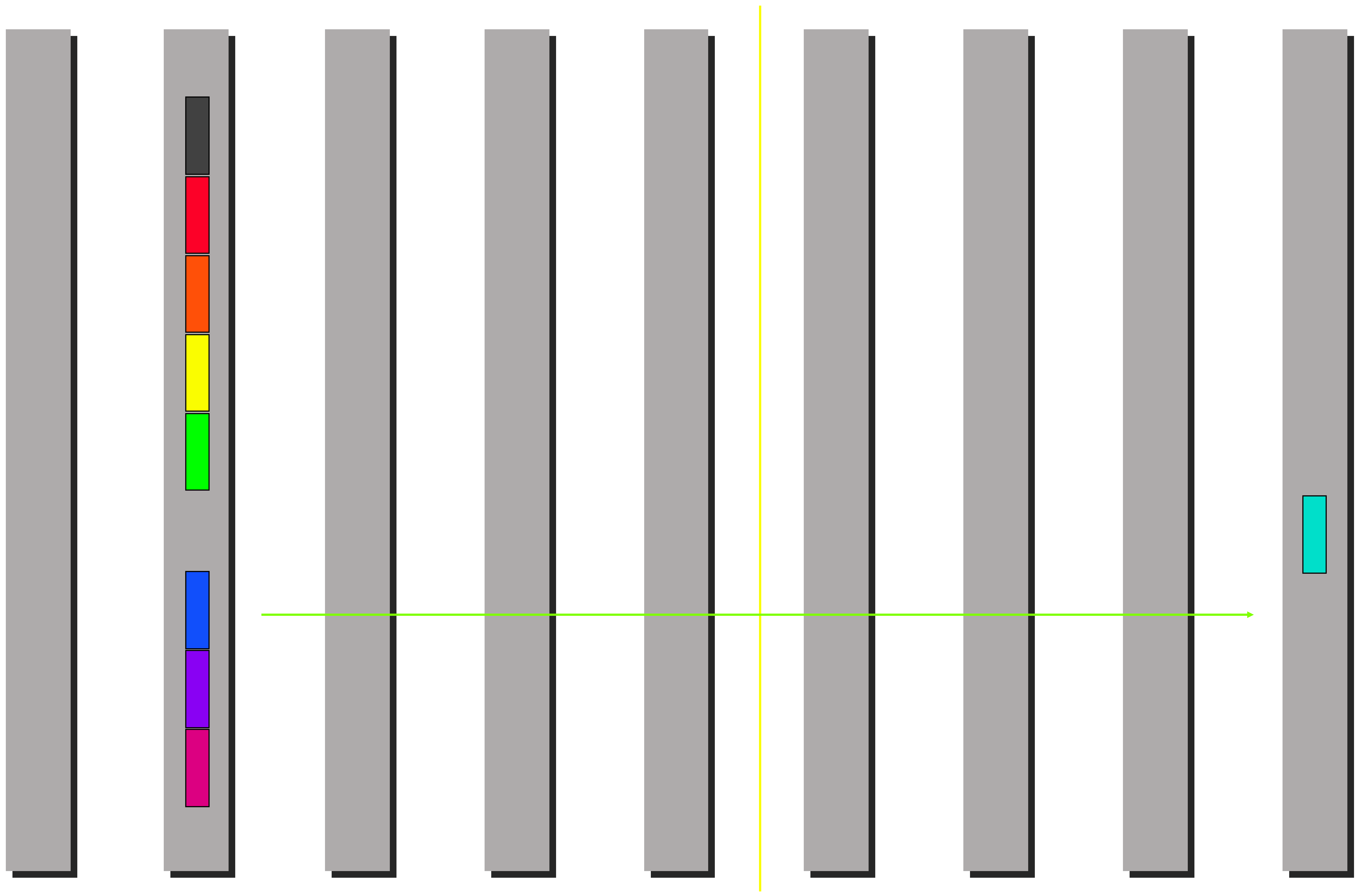
$$\lceil log(p) \rceil \ (\alpha + n\beta + n\gamma)$$

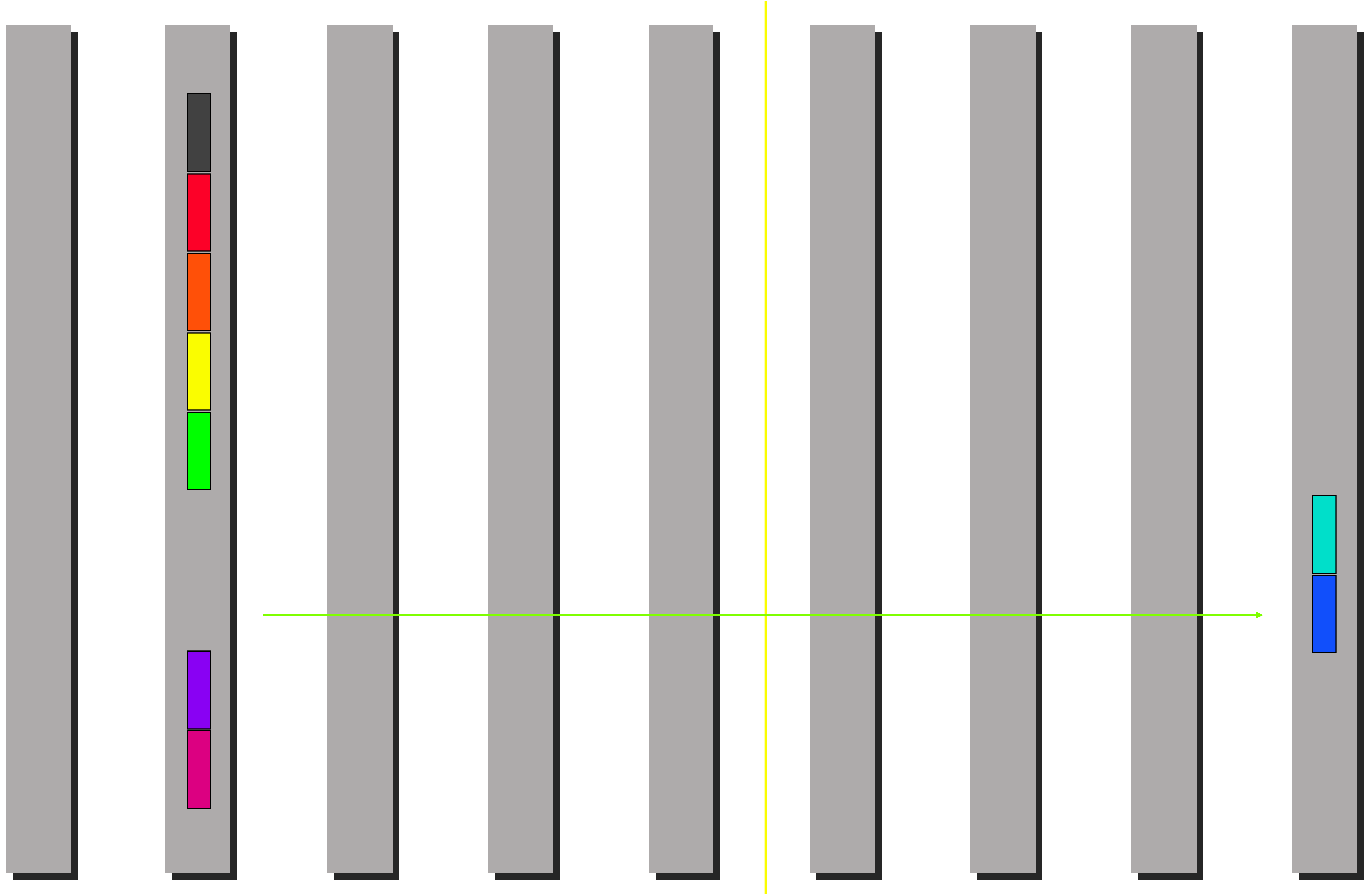
Scatter

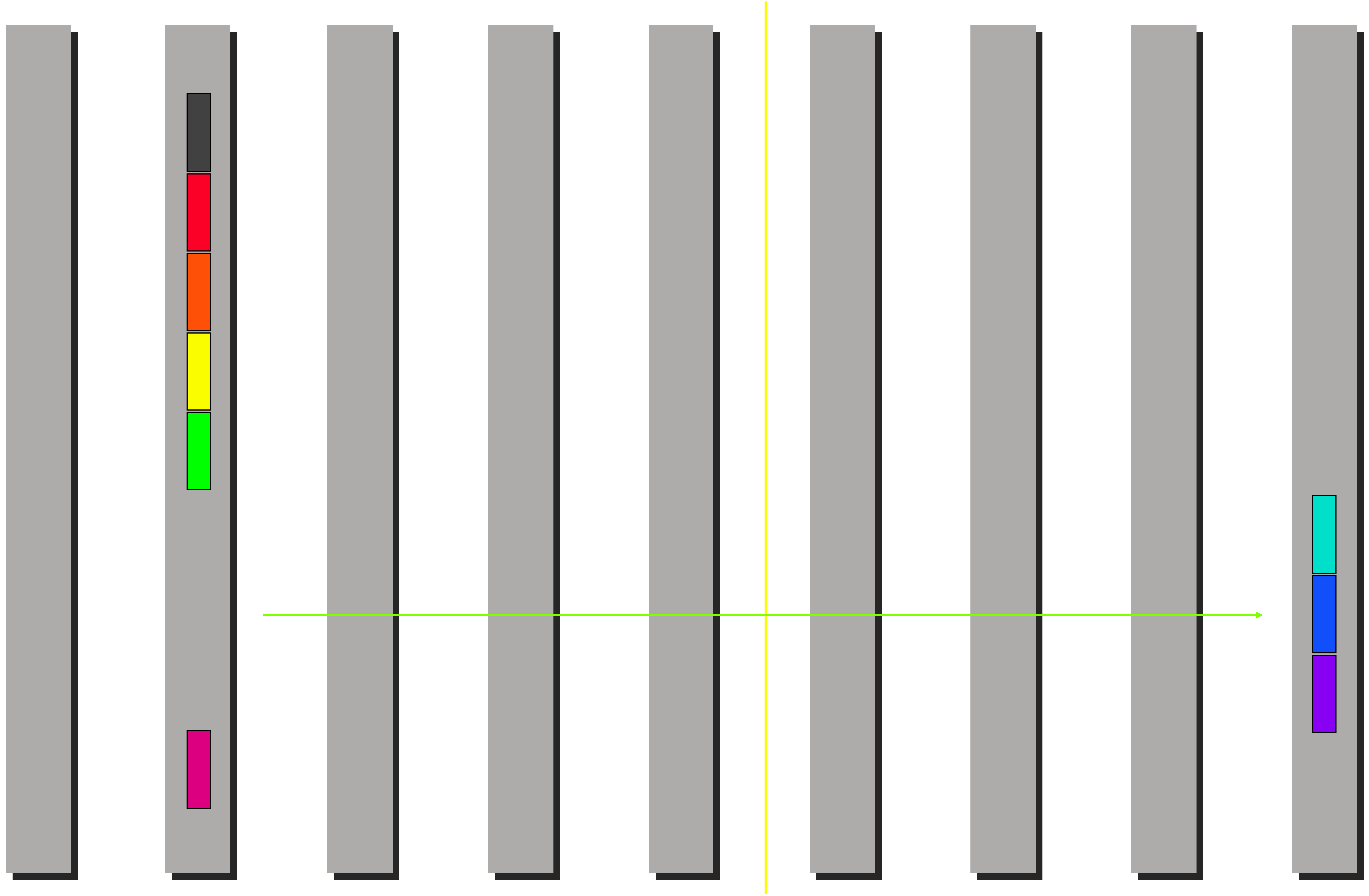


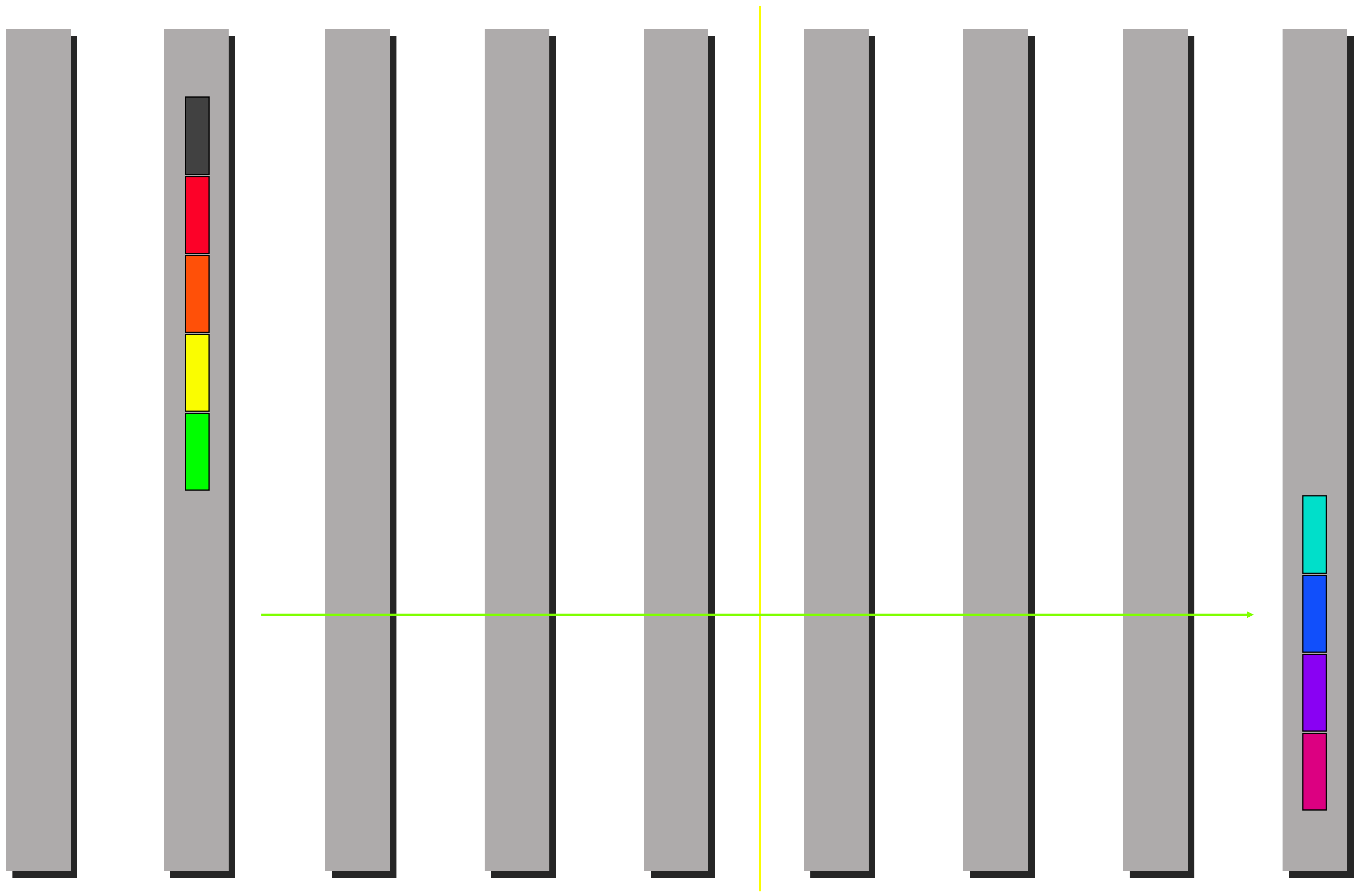


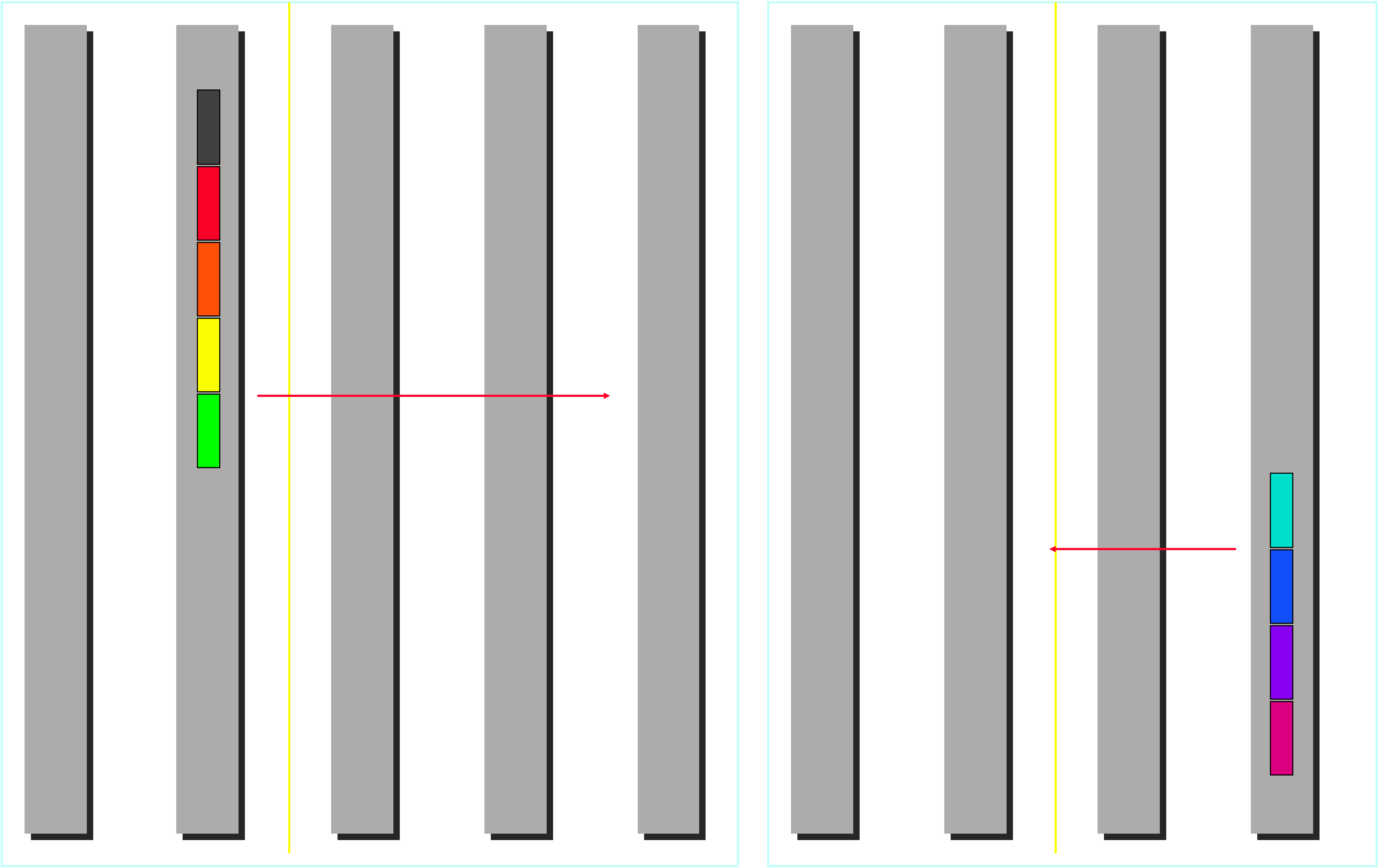


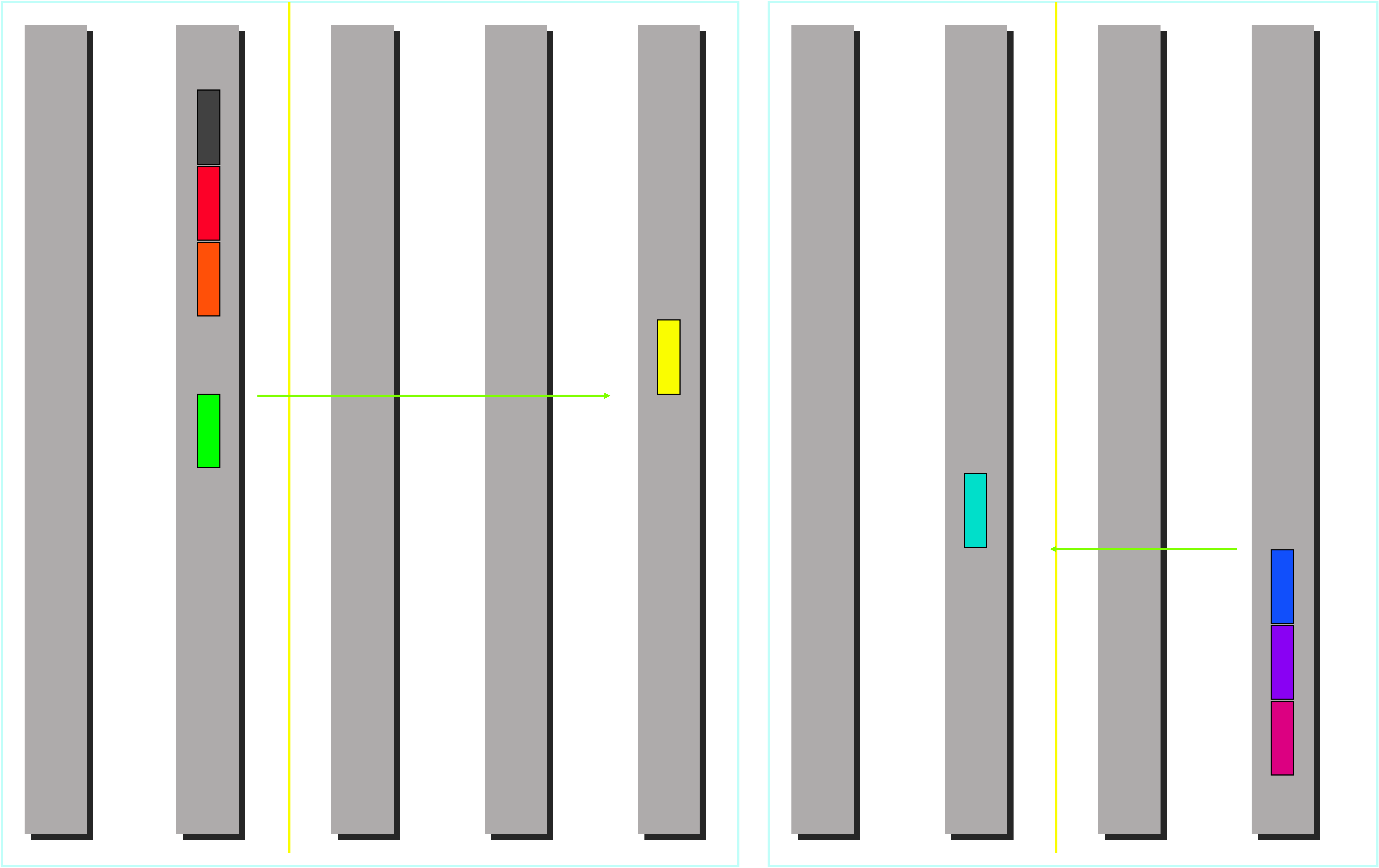


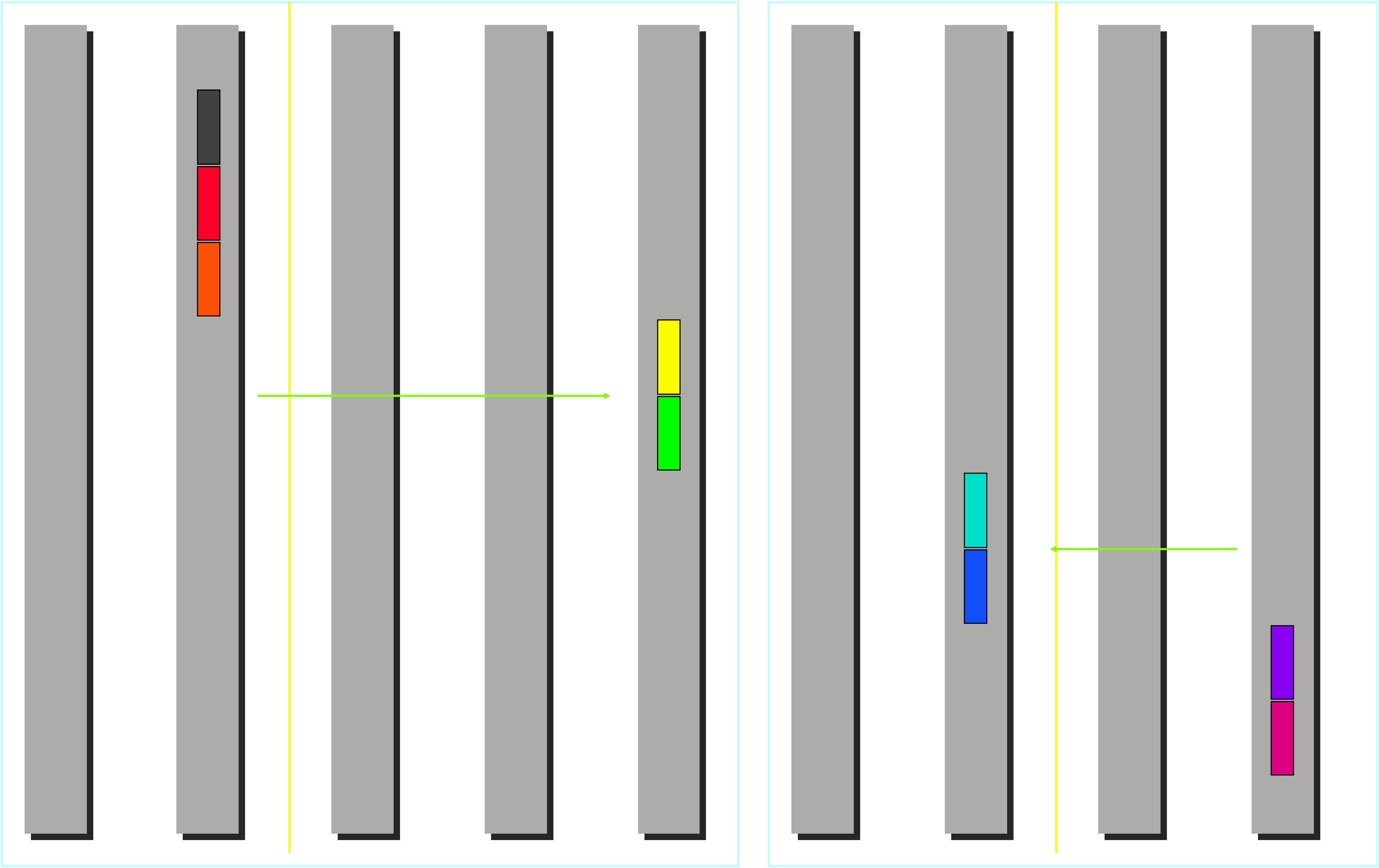


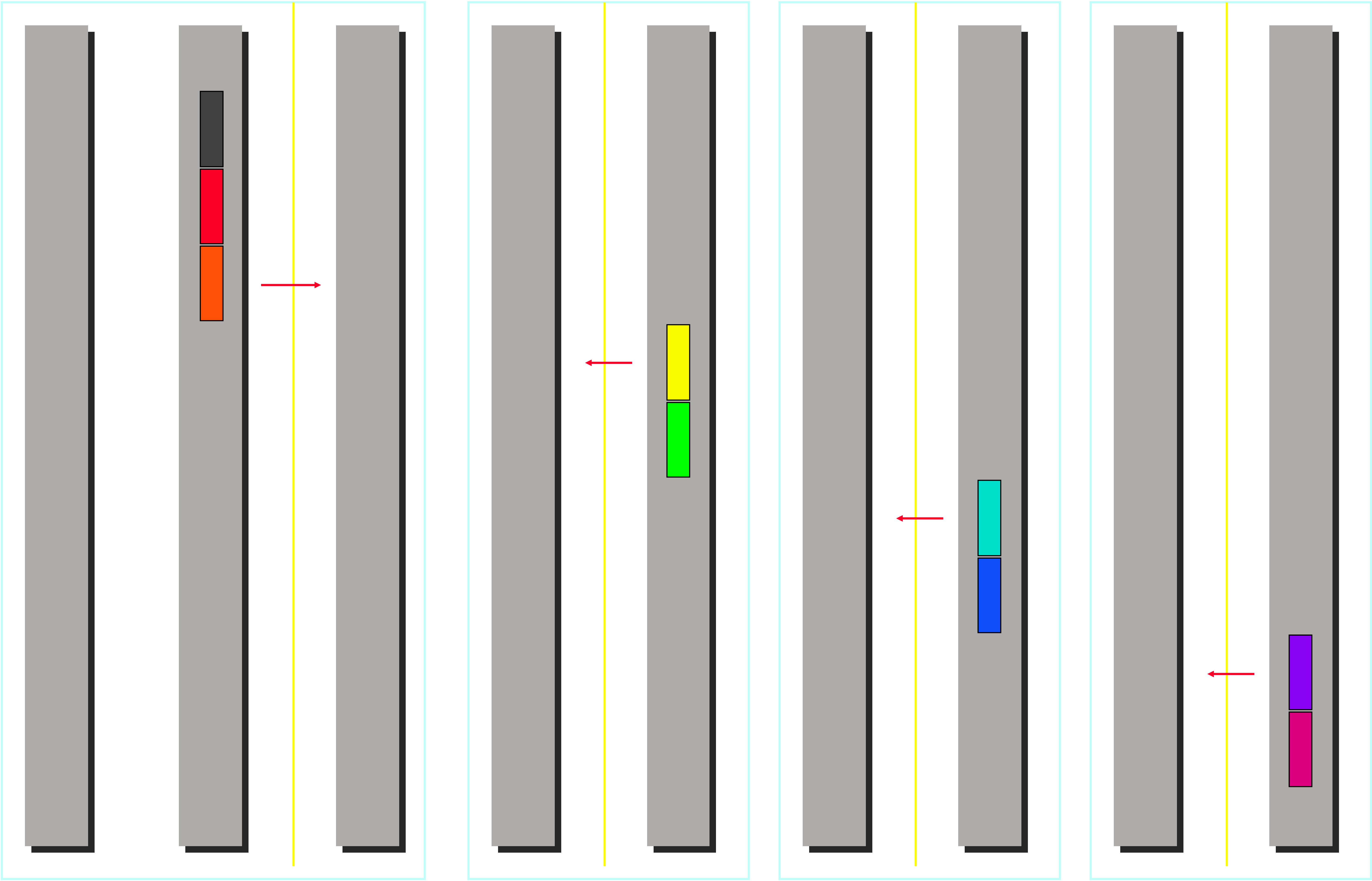


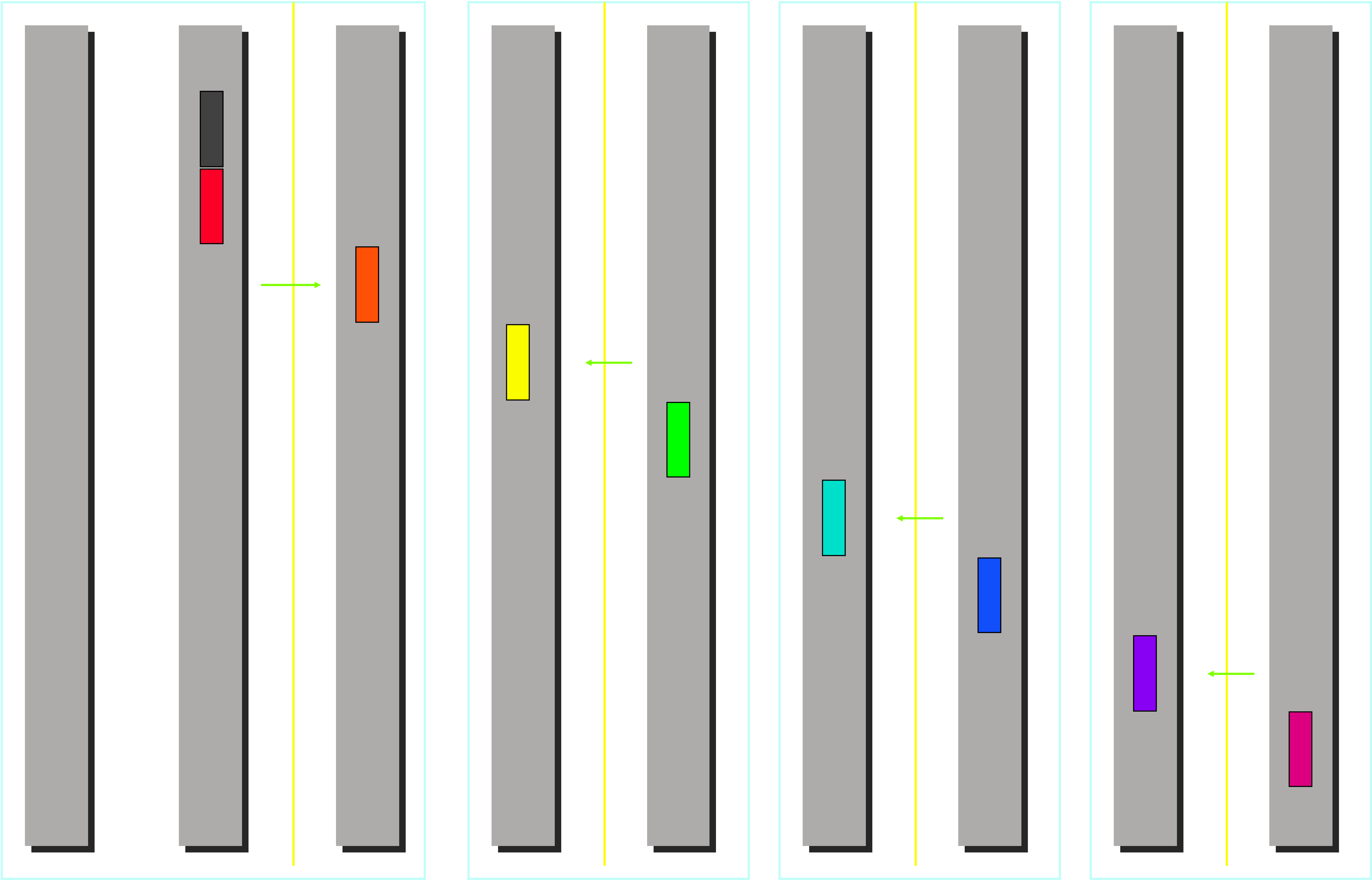


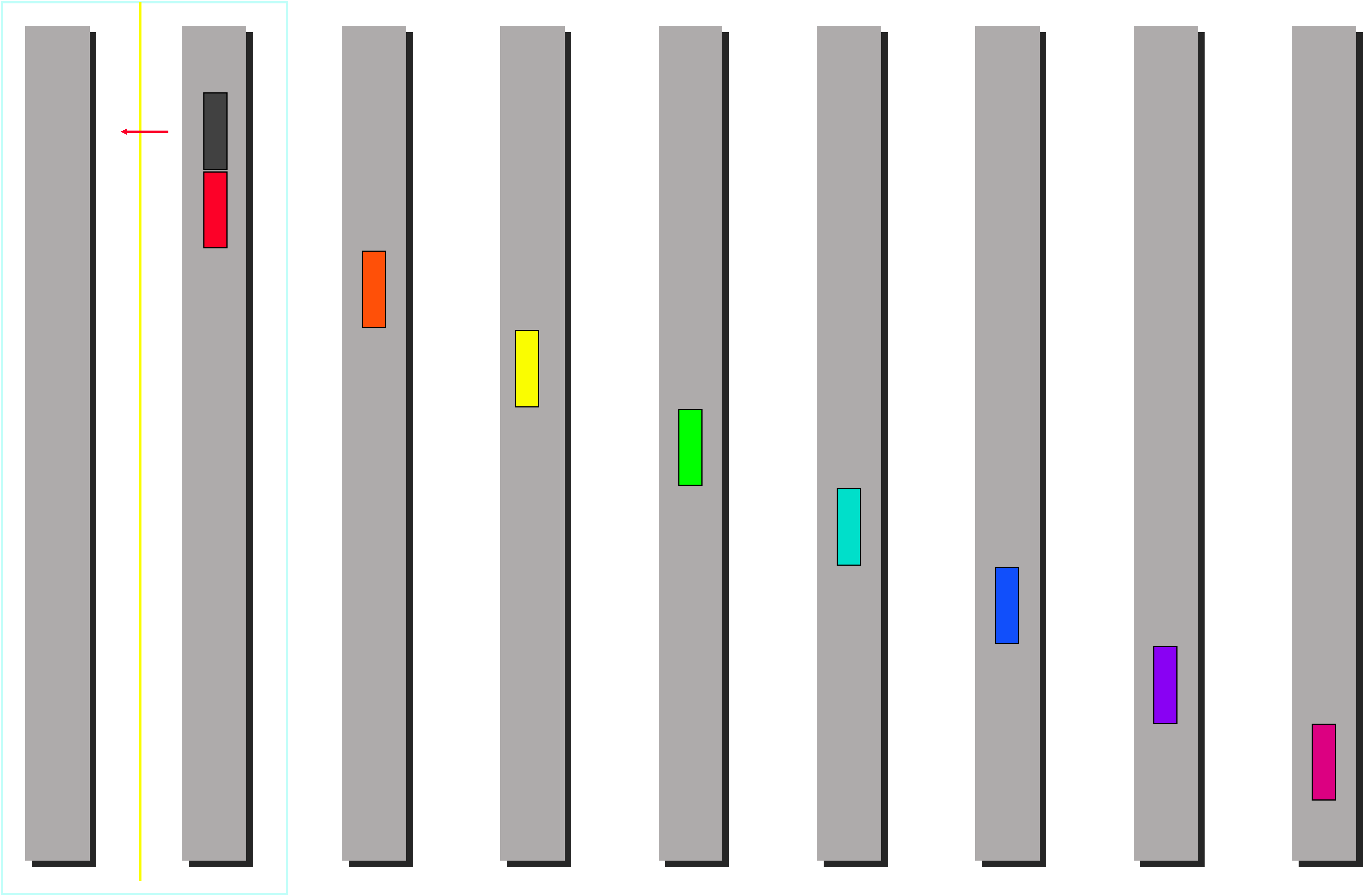


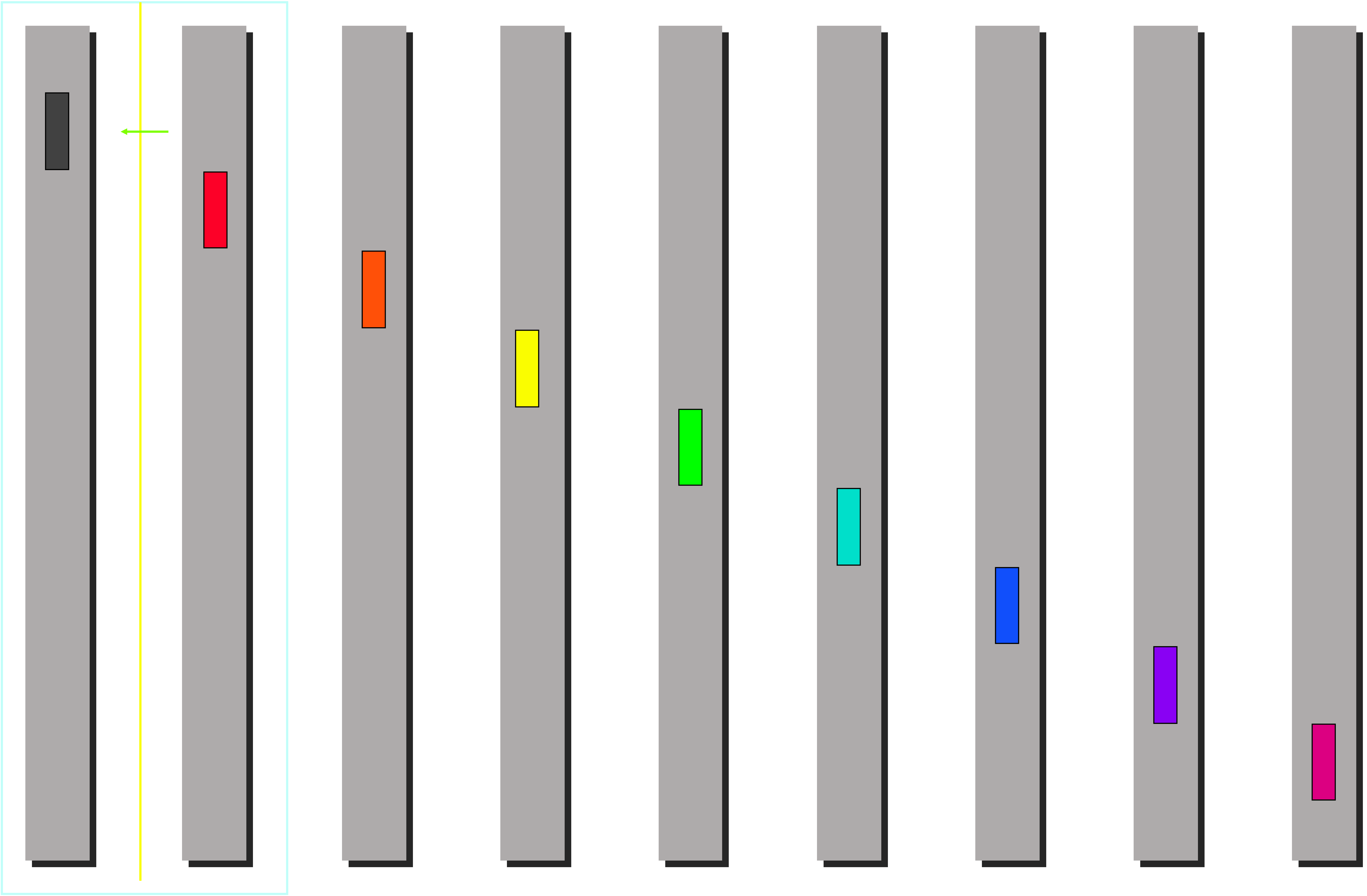


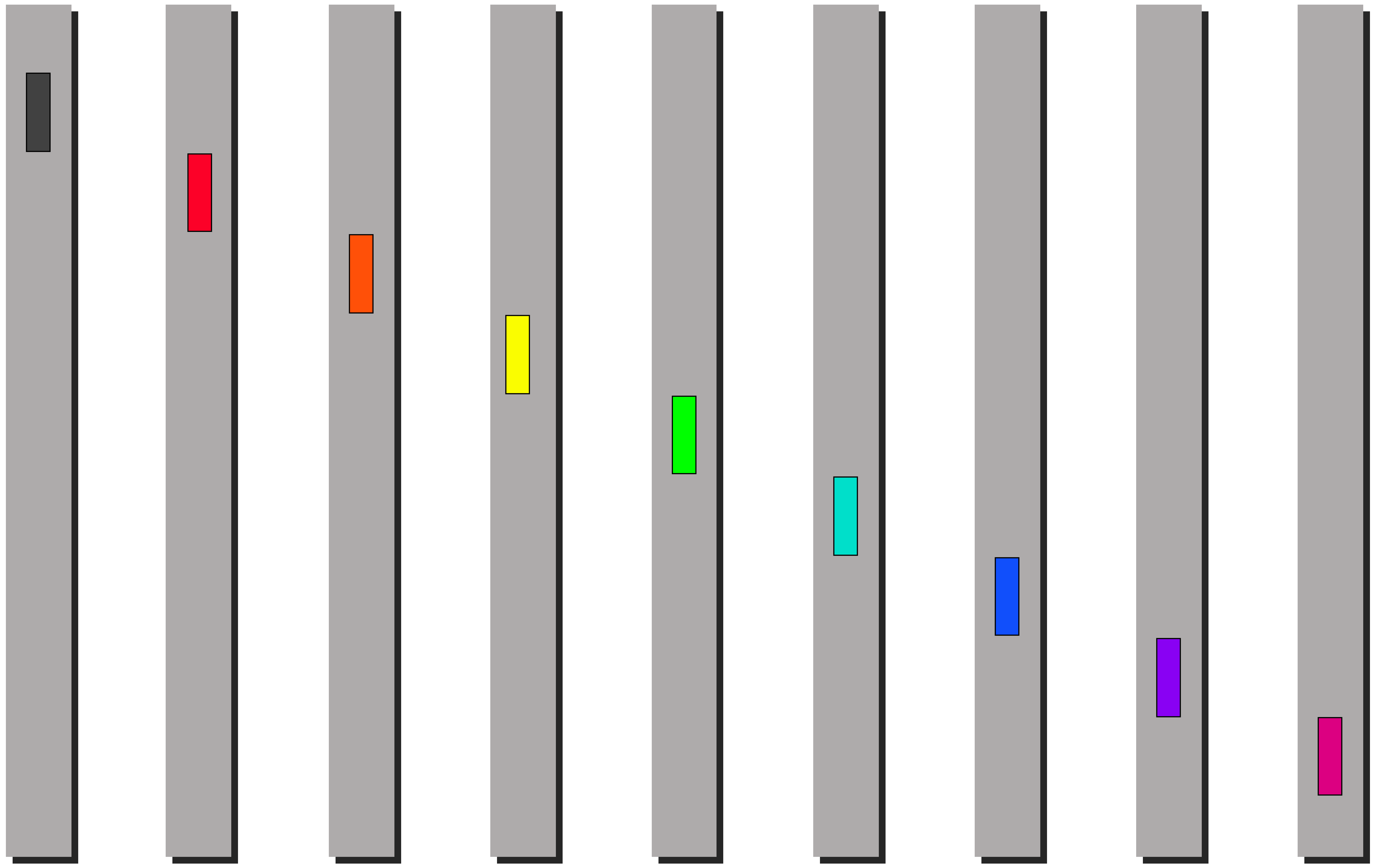












Cost of minimum spanning tree scatter

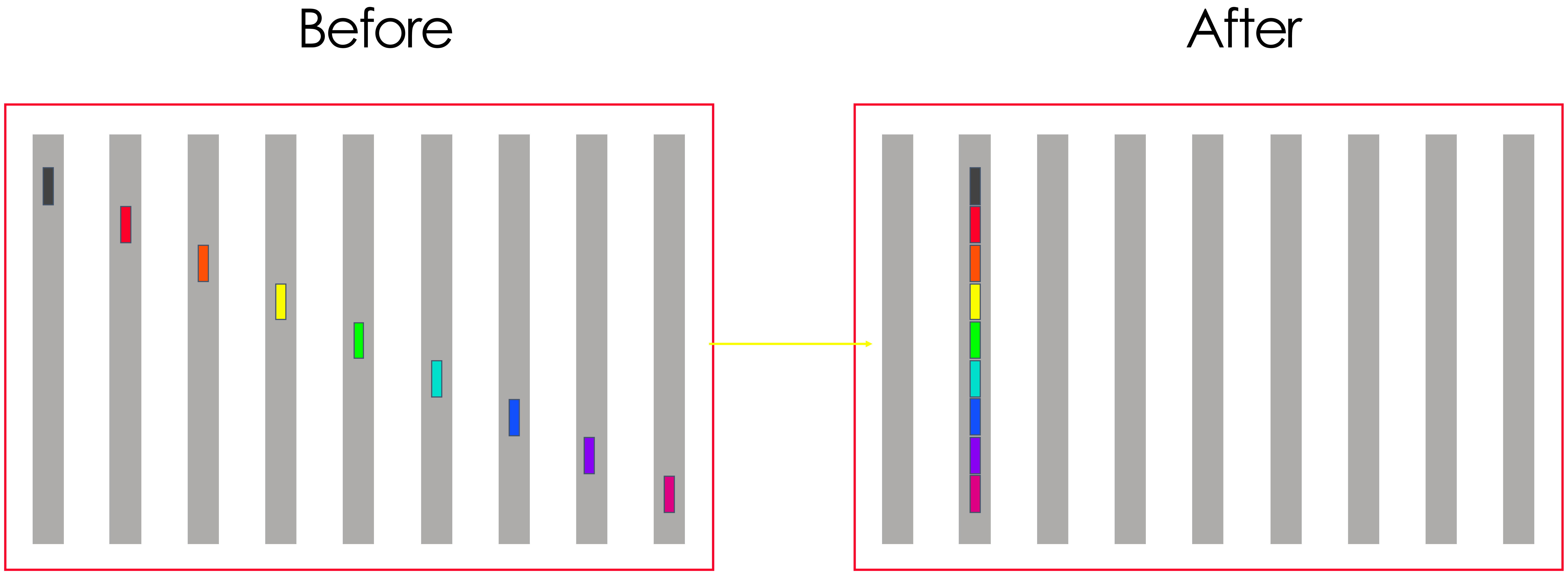
- Assumption: power of two number of nodes

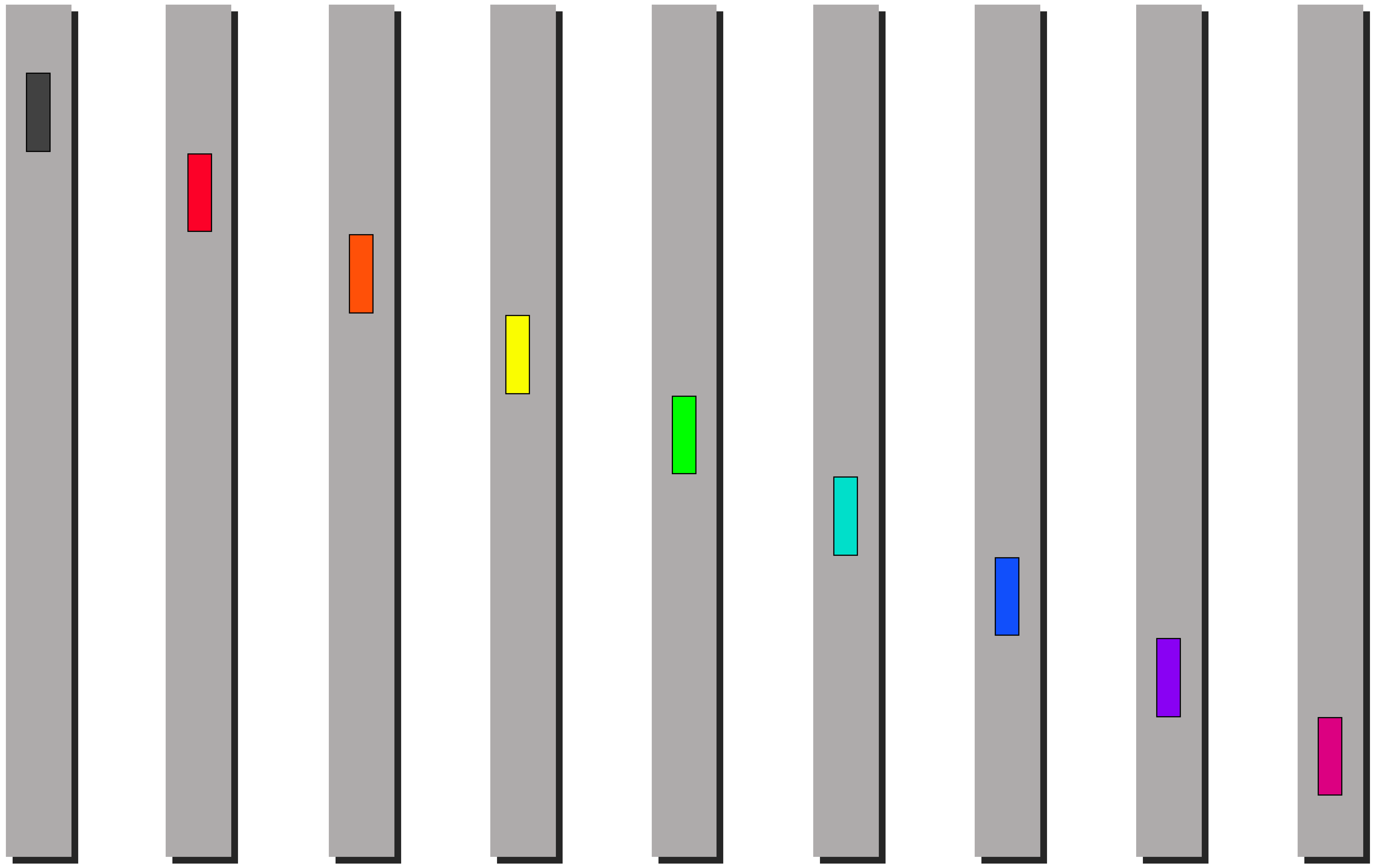
$$\sum_{k=1}^{log(p)} \left(\alpha + \frac{n}{2^k} \beta \right)$$

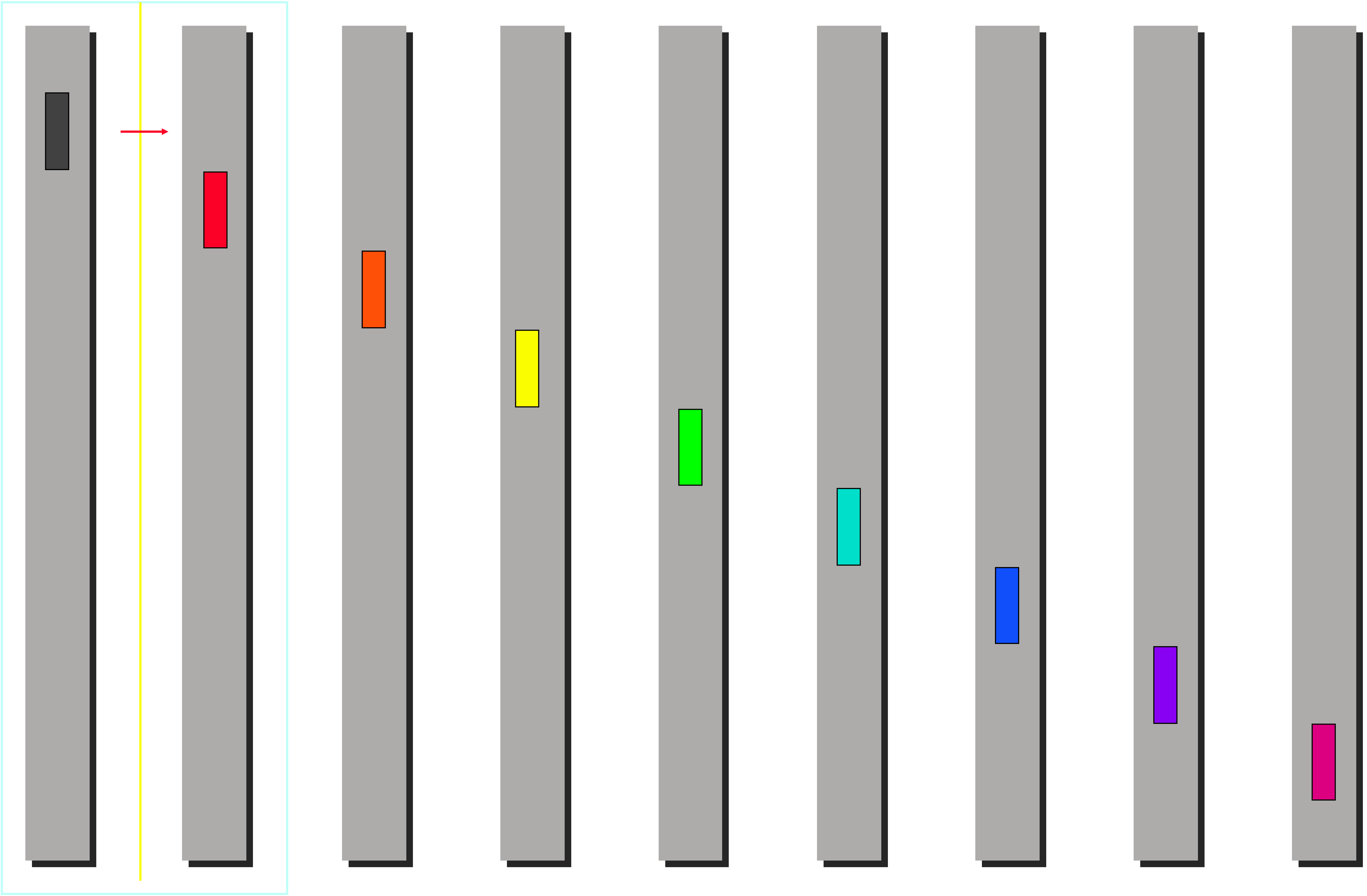
$$=$$

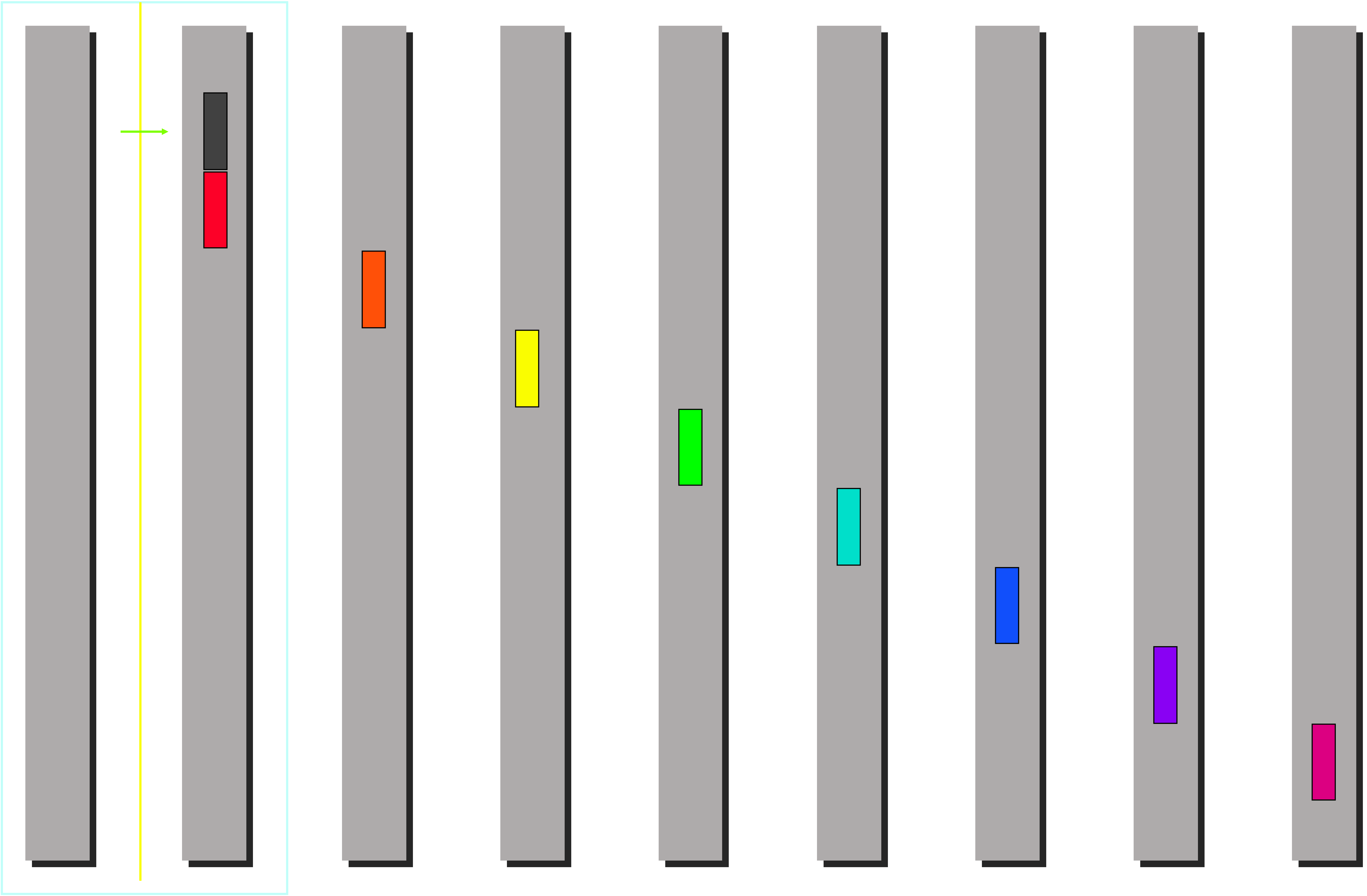
$$log(p) \ \alpha + \frac{p-1}{p} n\beta$$

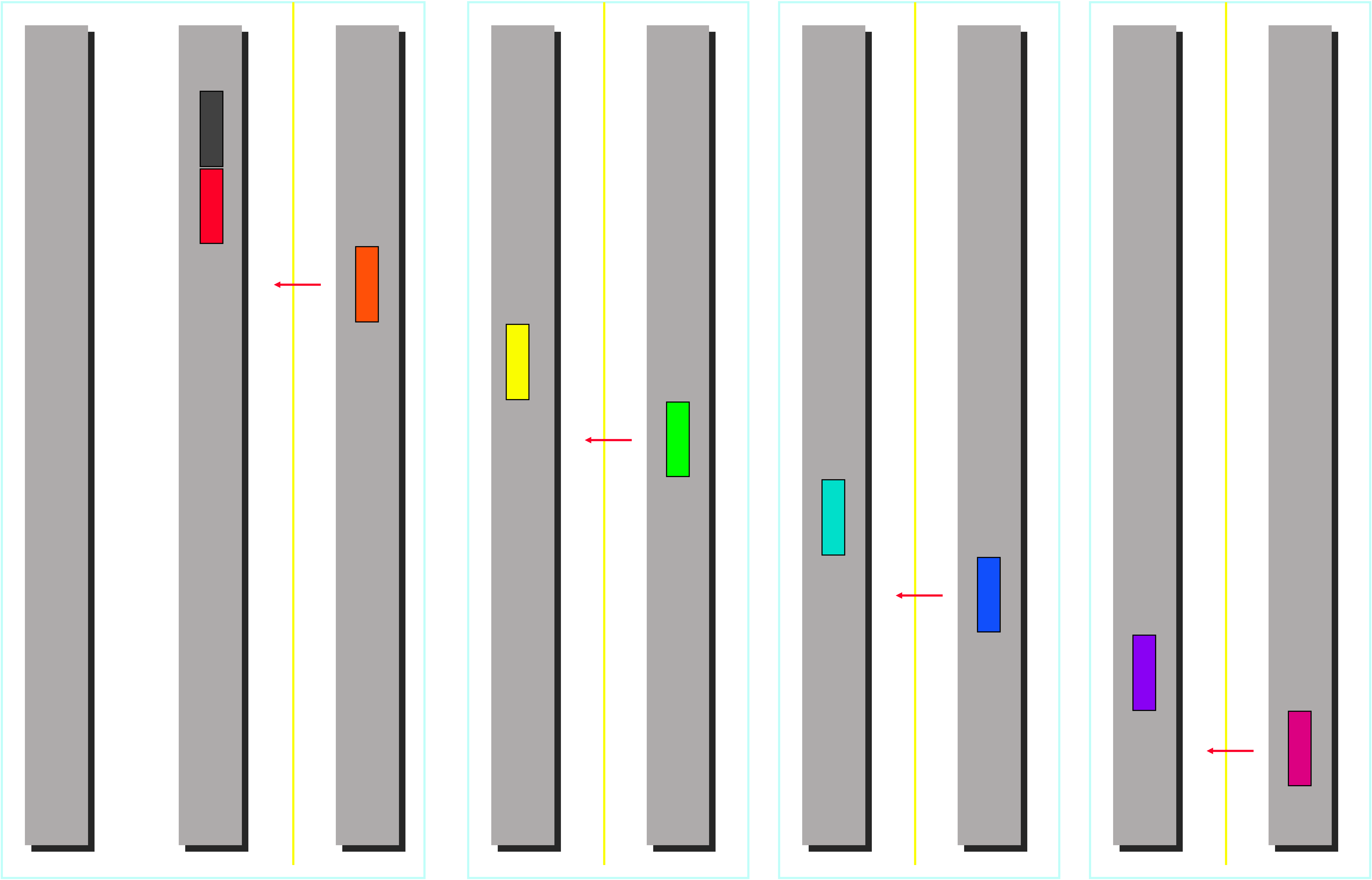
Gather

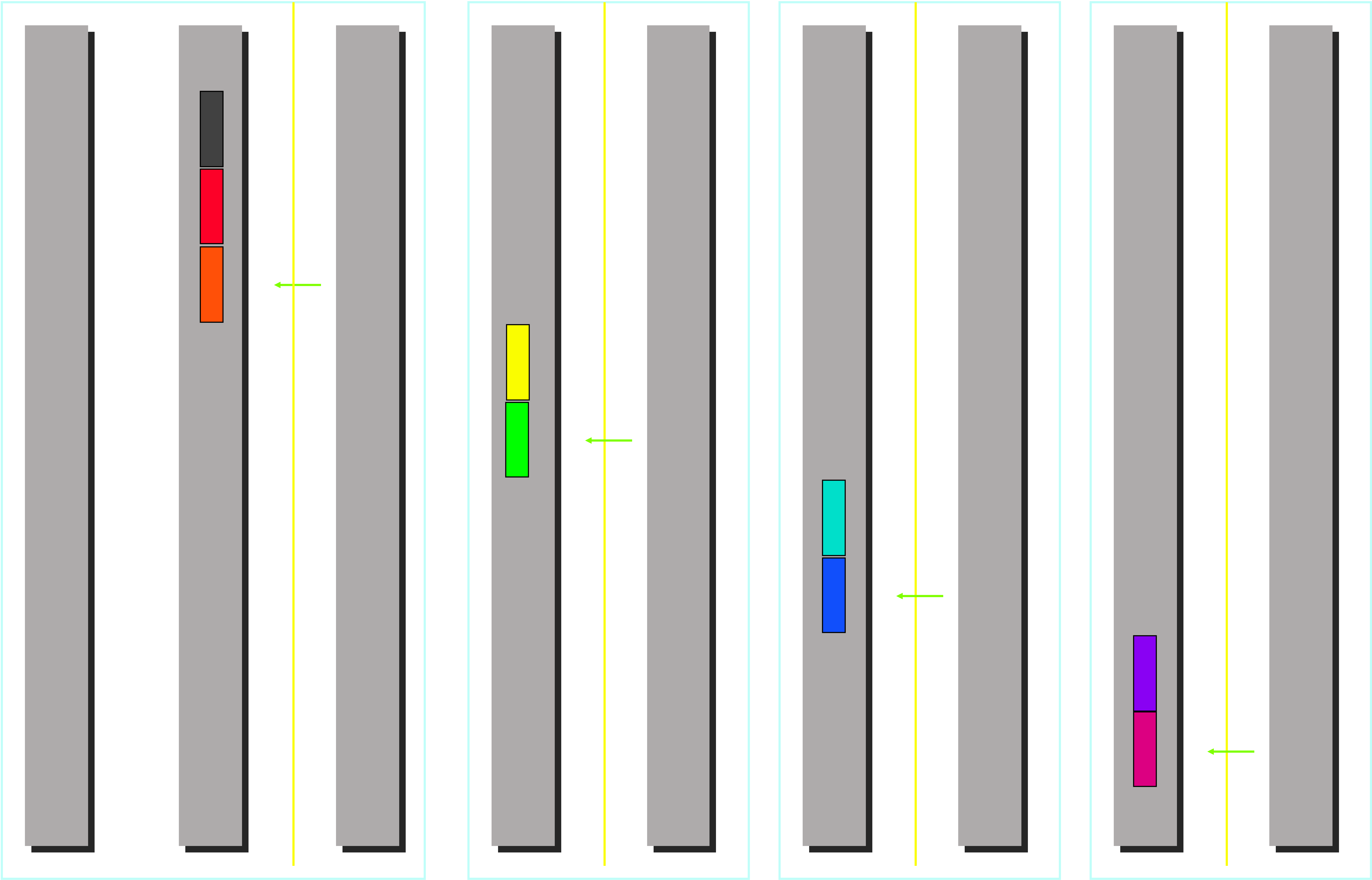


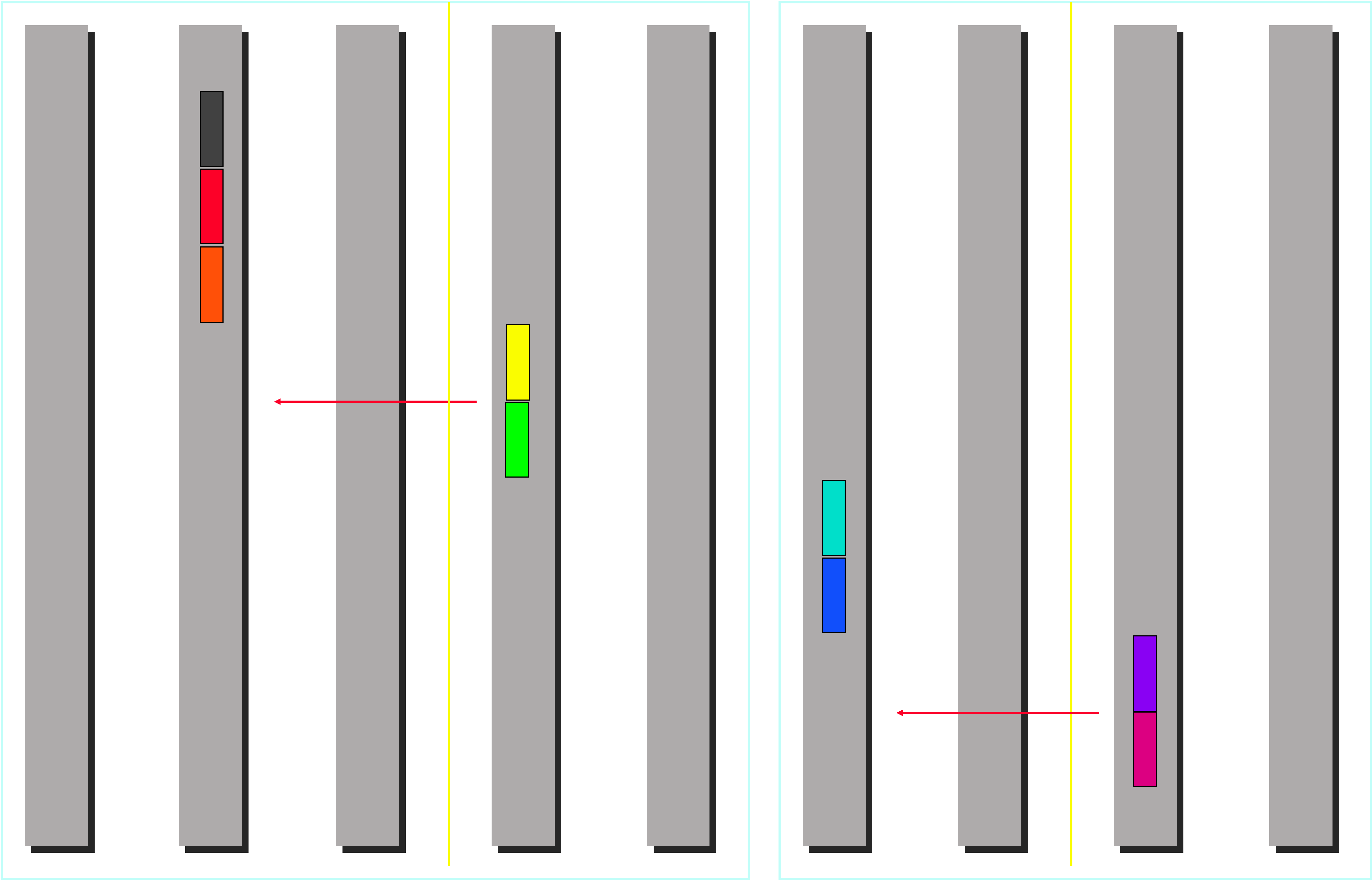


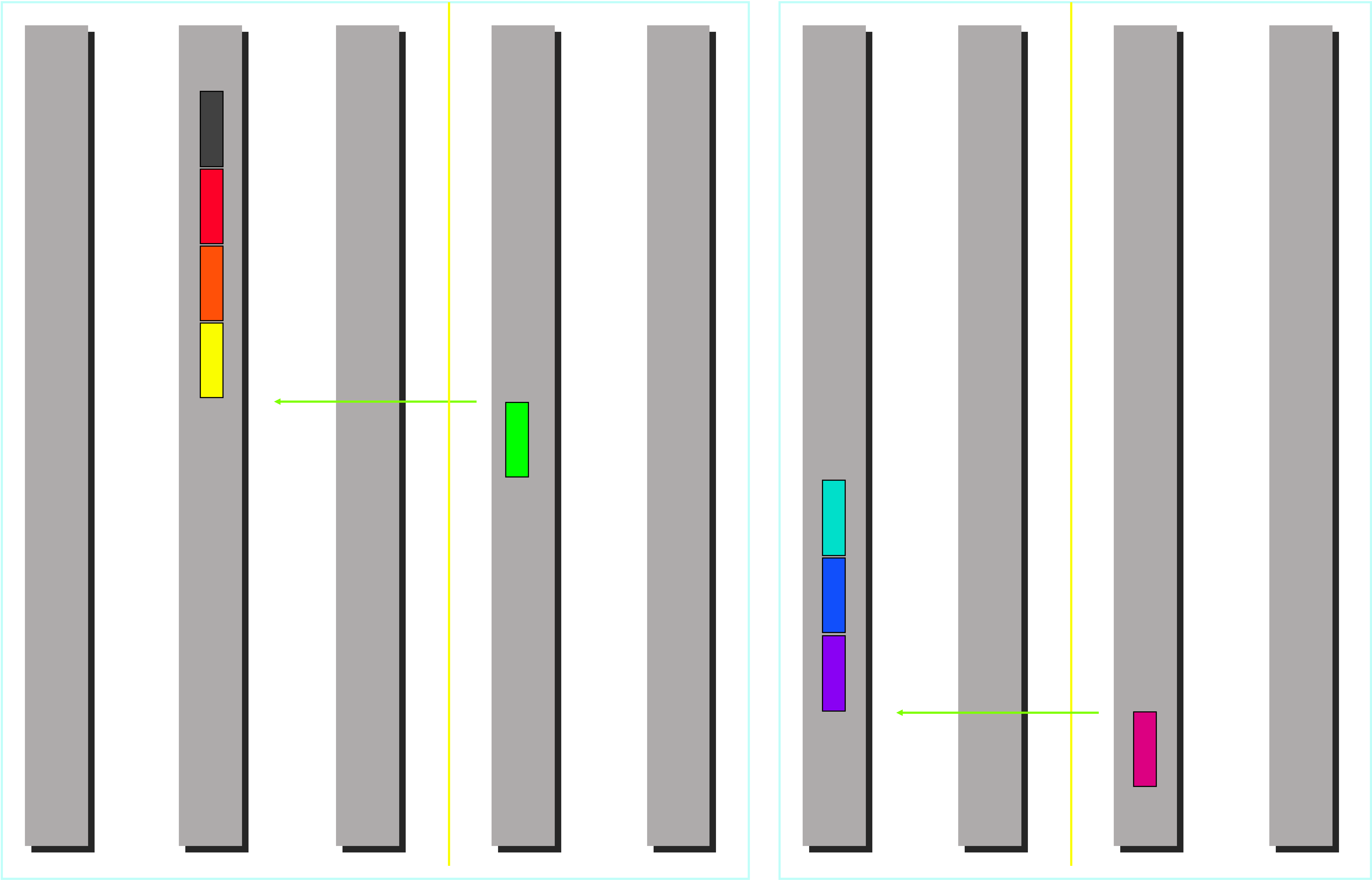


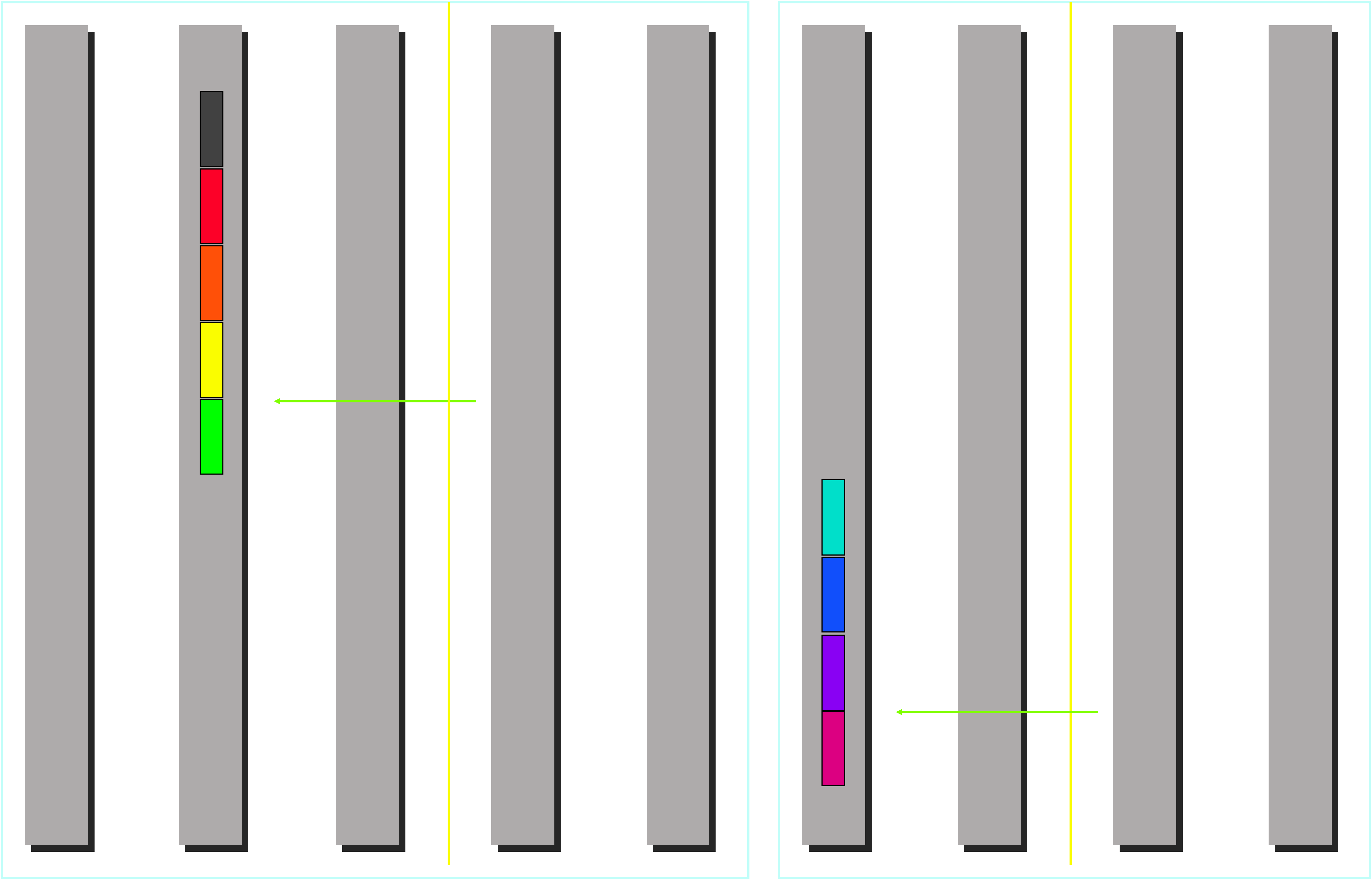


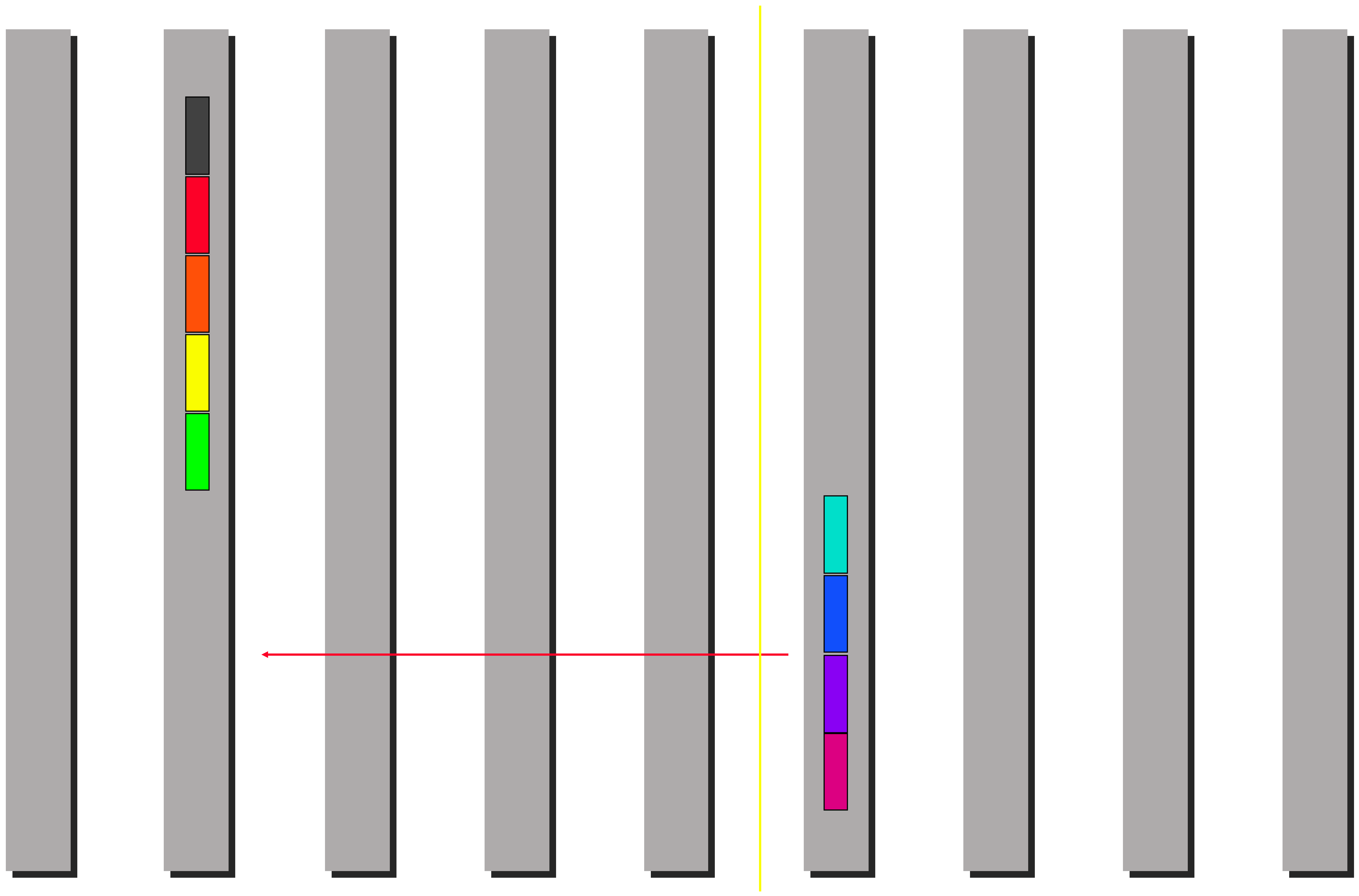


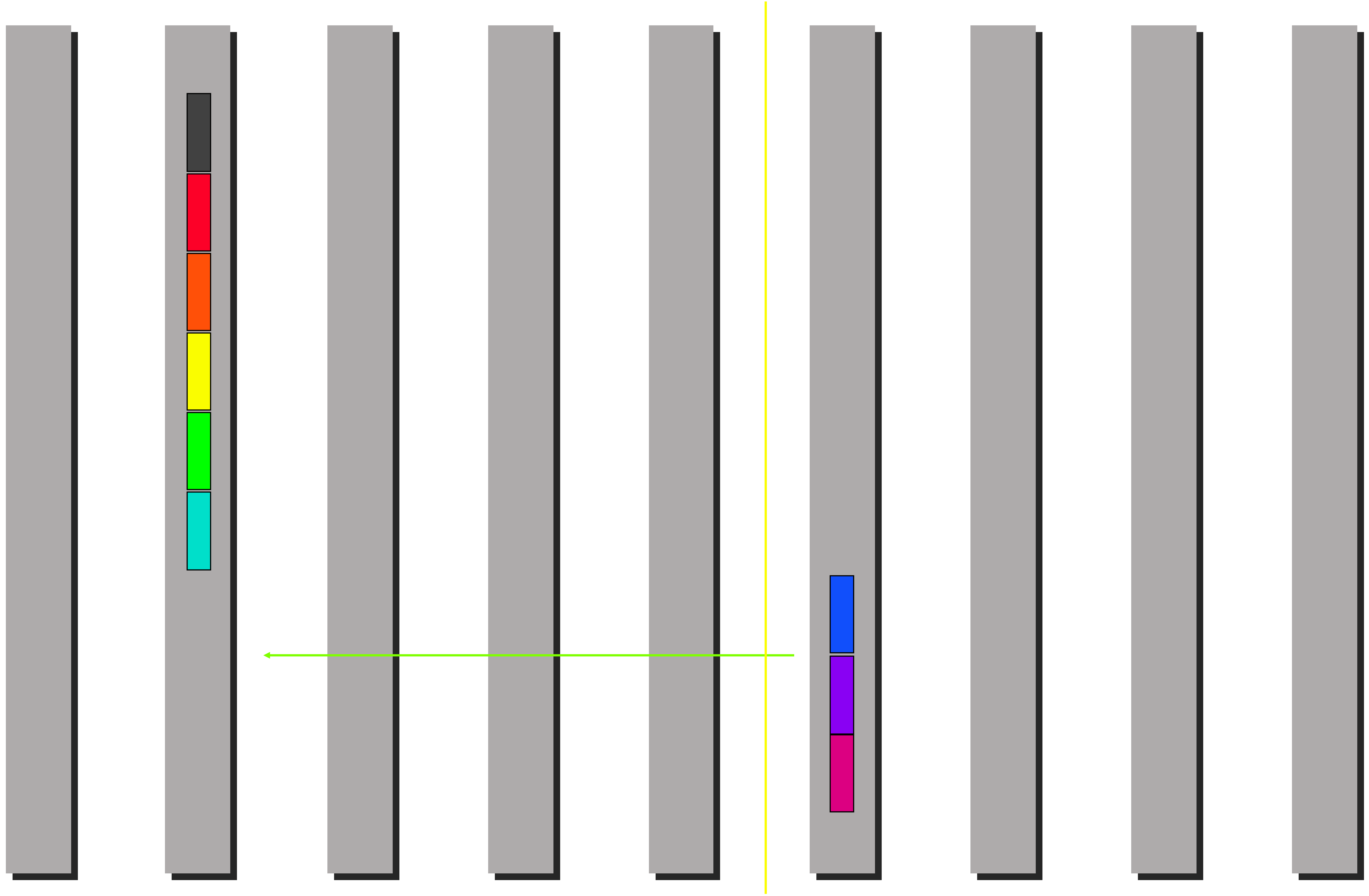


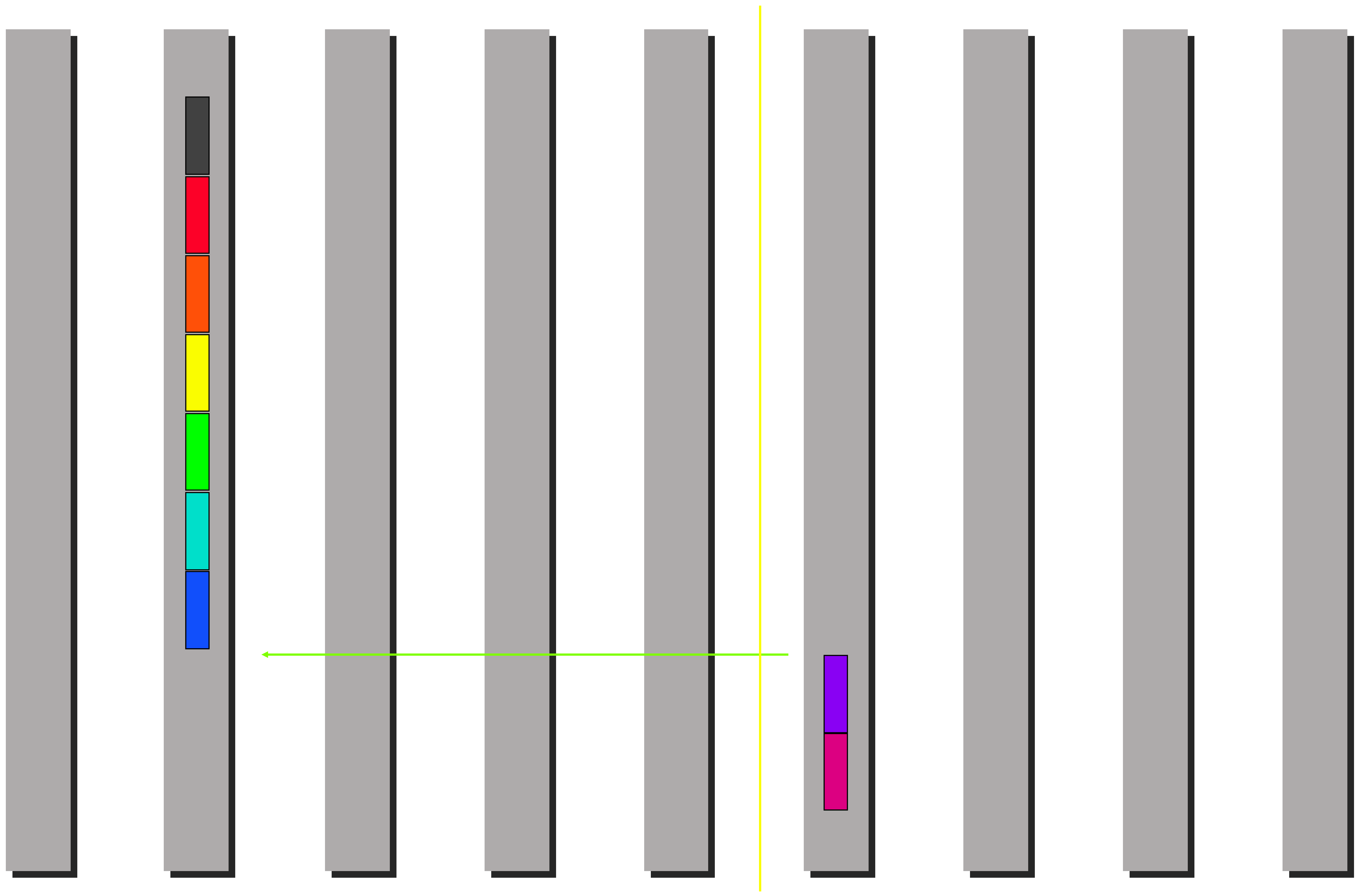


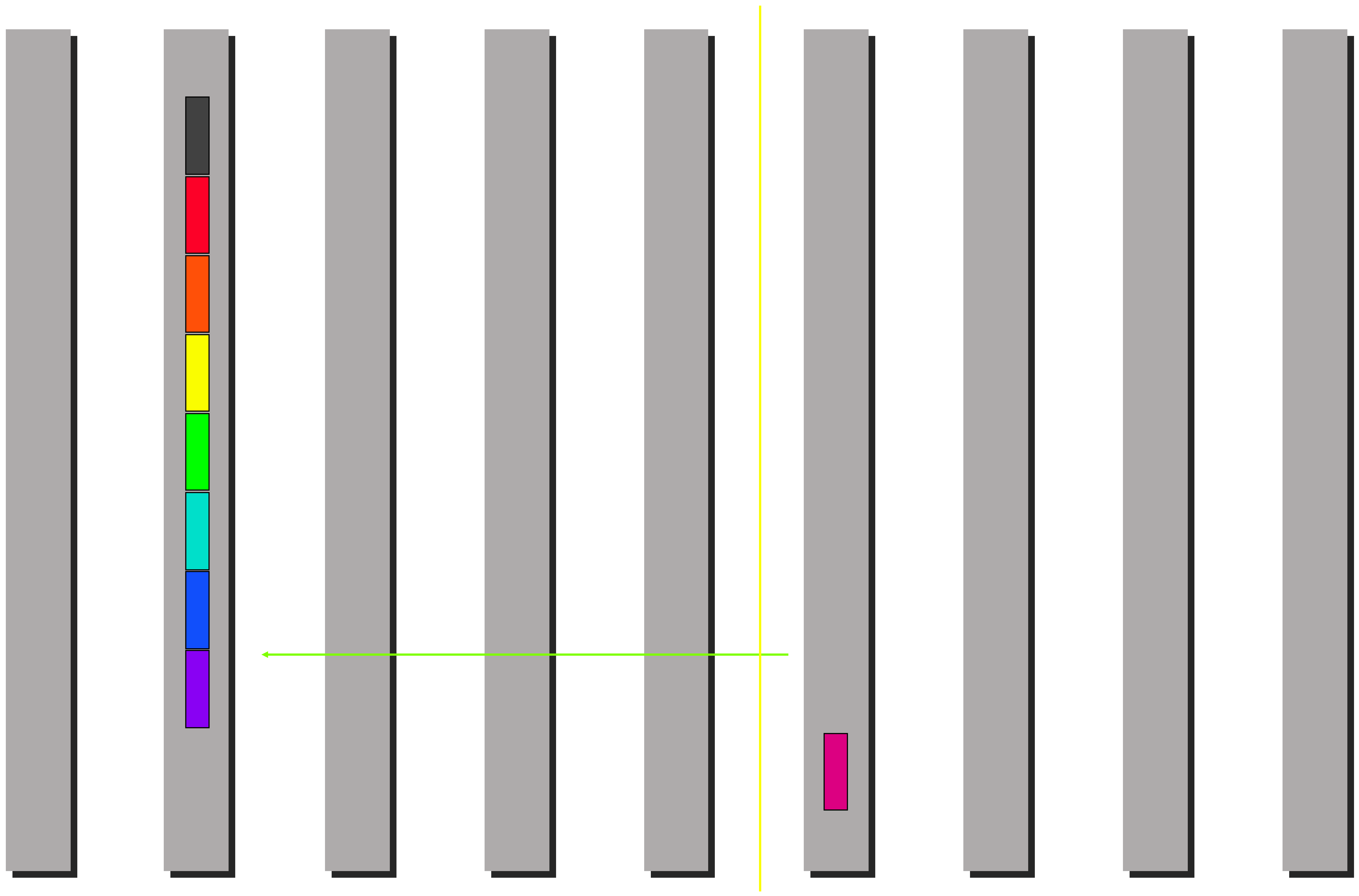


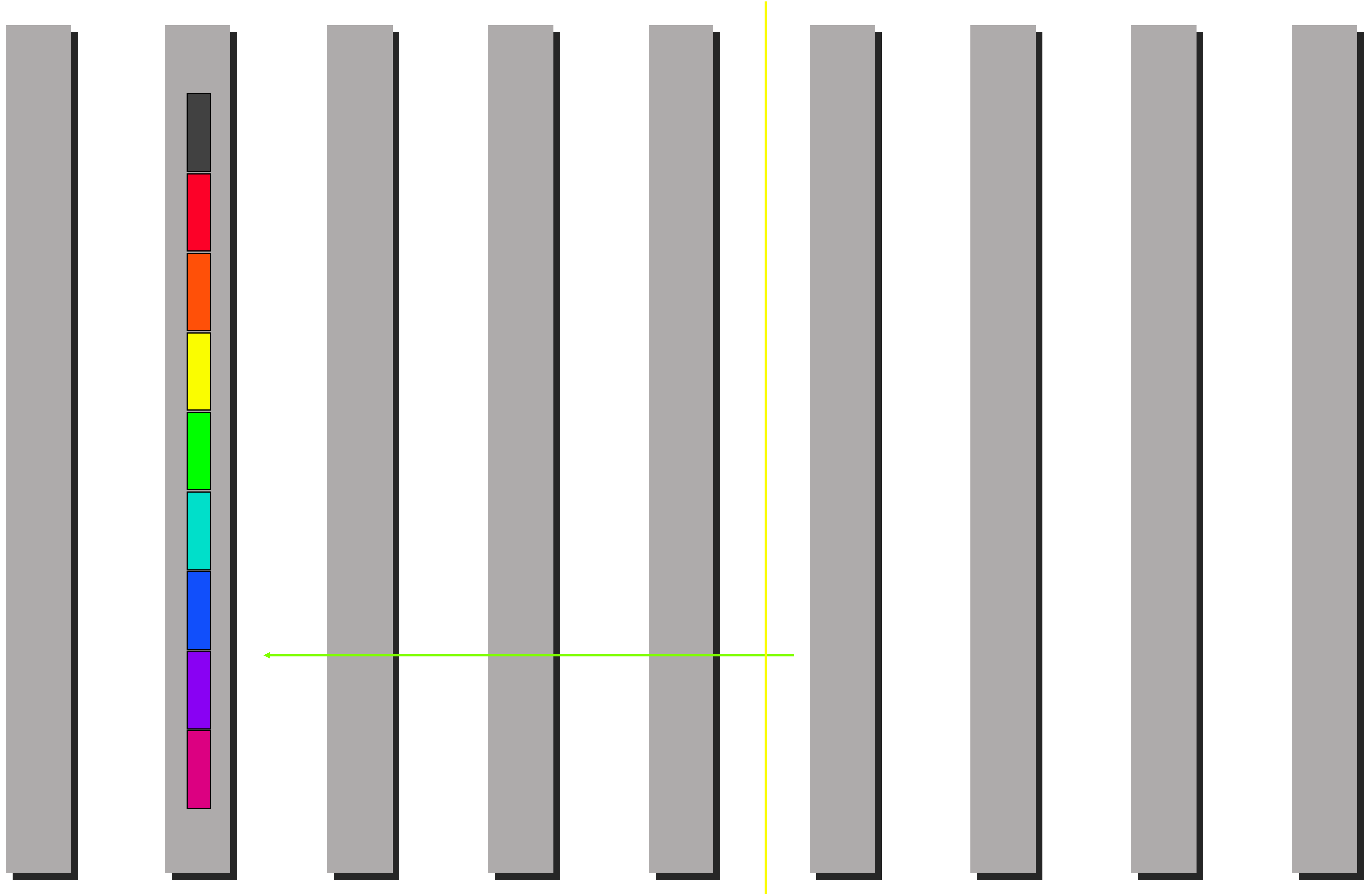


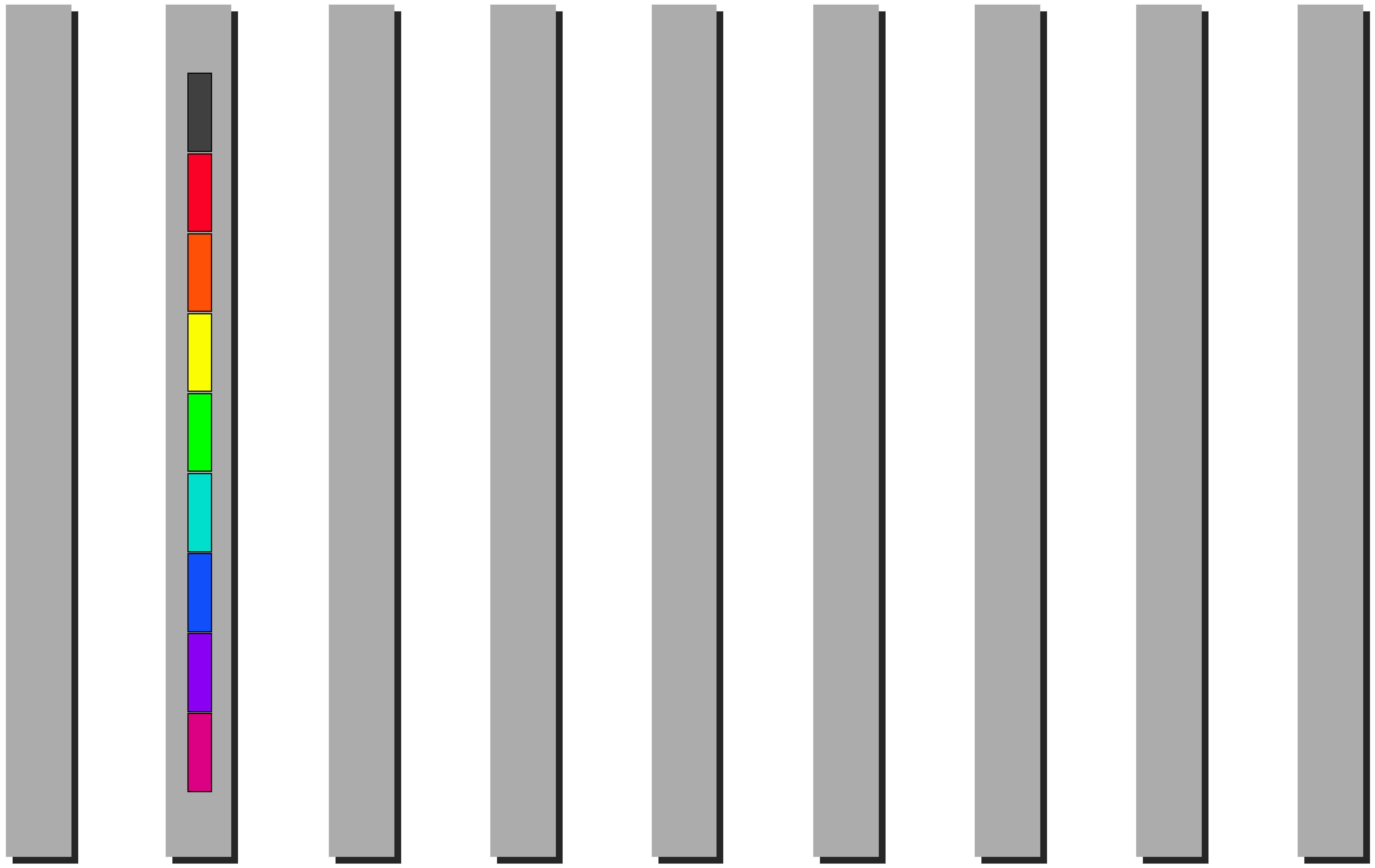












Cost of minimum spanning tree gather

- Assumption: power of two number of nodes

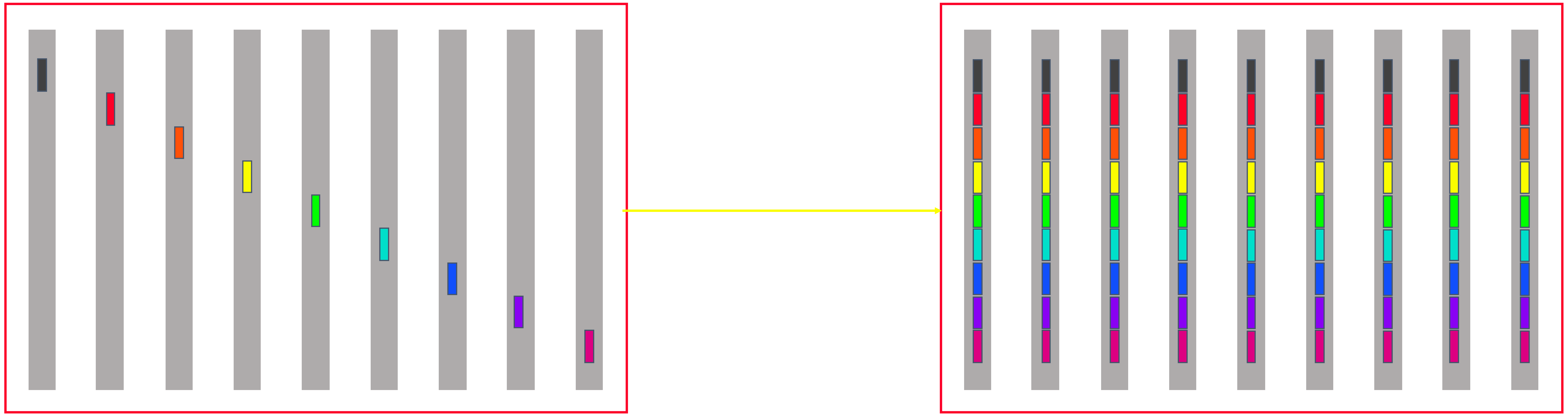
$$\sum_{k=1}^{log(p)} \left(\alpha + \frac{n}{2^k} \beta \right)$$

$$=$$

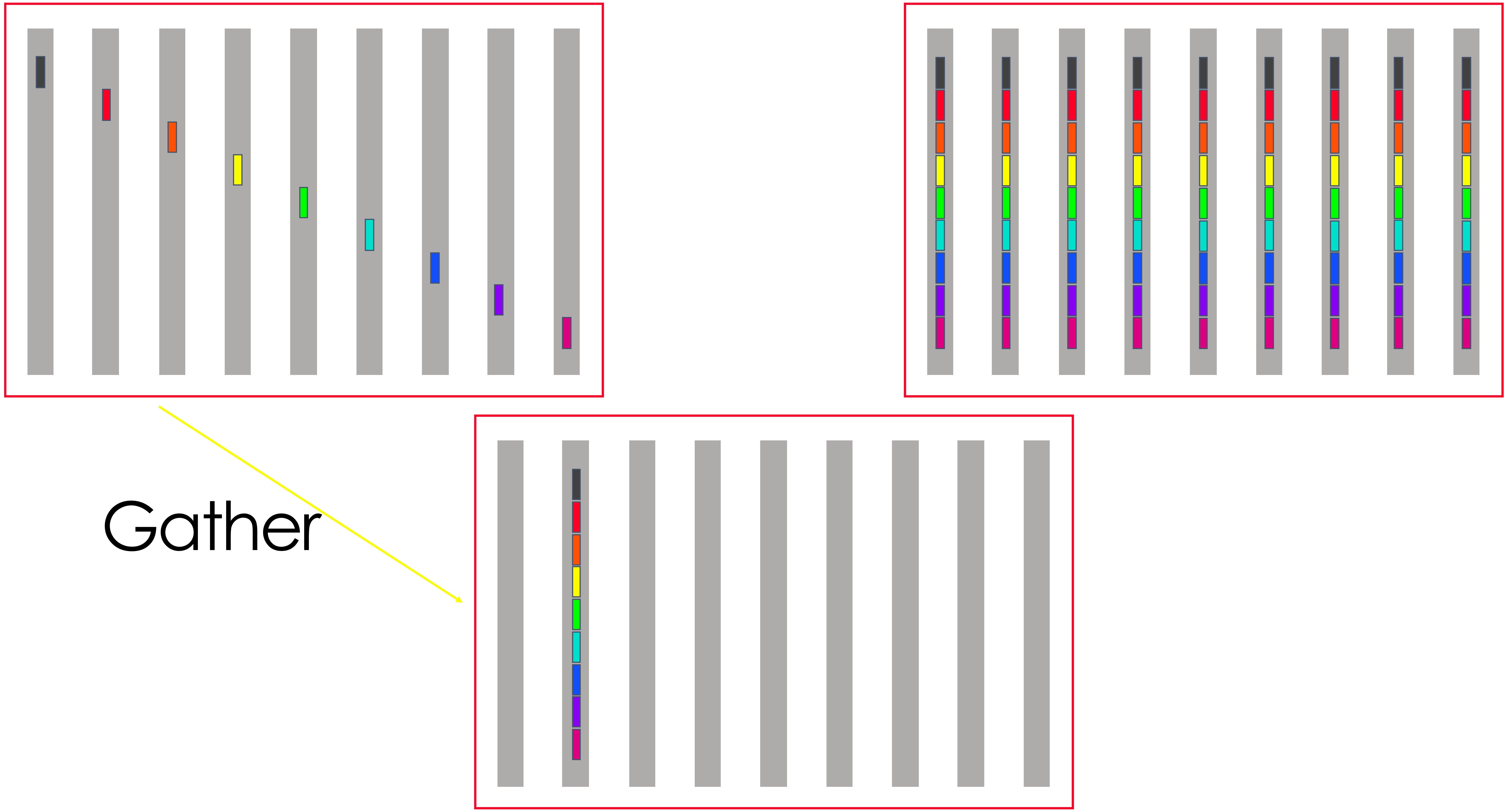
$$log(p) \alpha + \frac{p-1}{p} n\beta$$

Using the building blocks

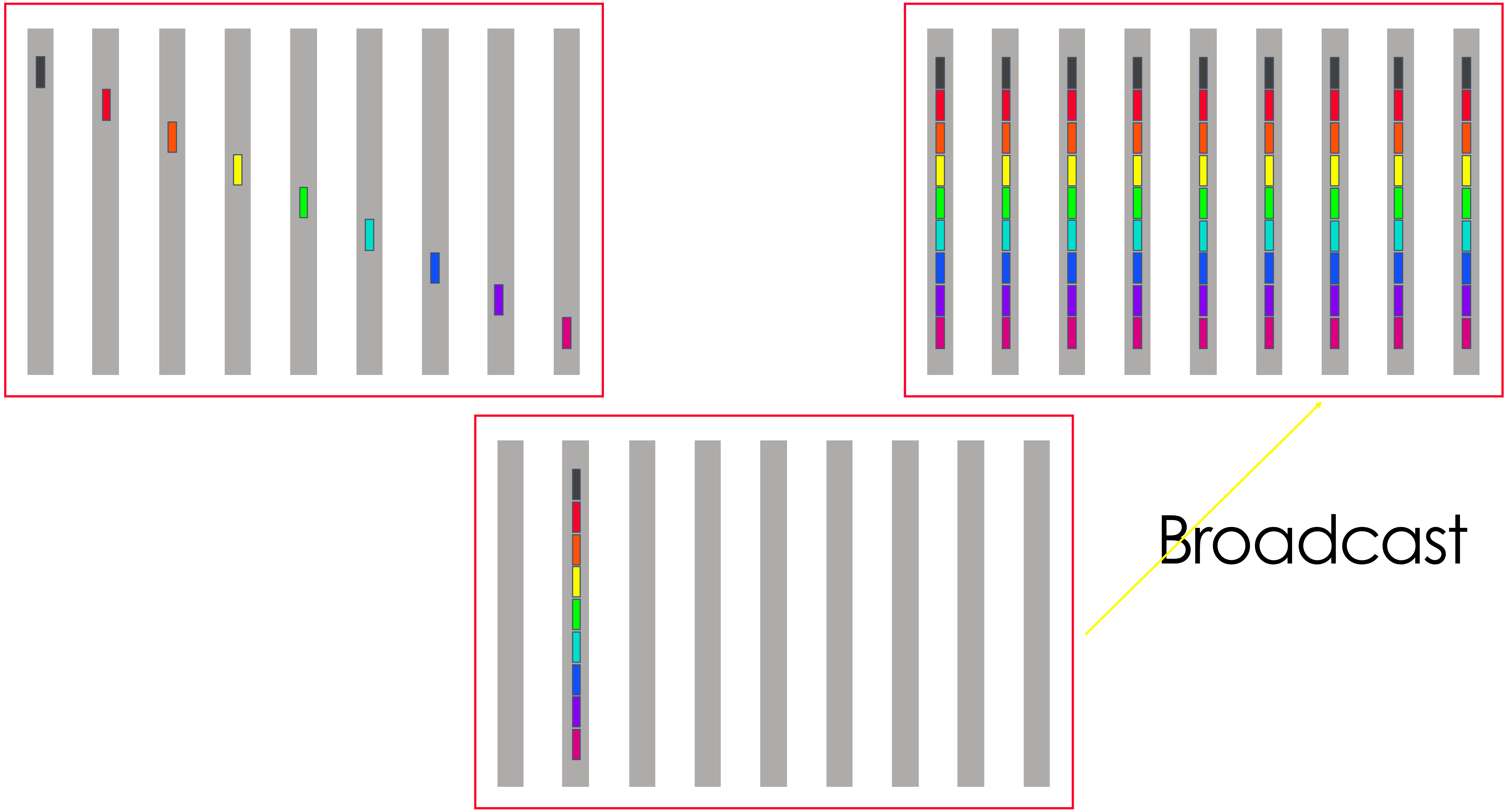
Allgather



Allgather



Allgather (short vector)



Cost of gather/broadcast allgather

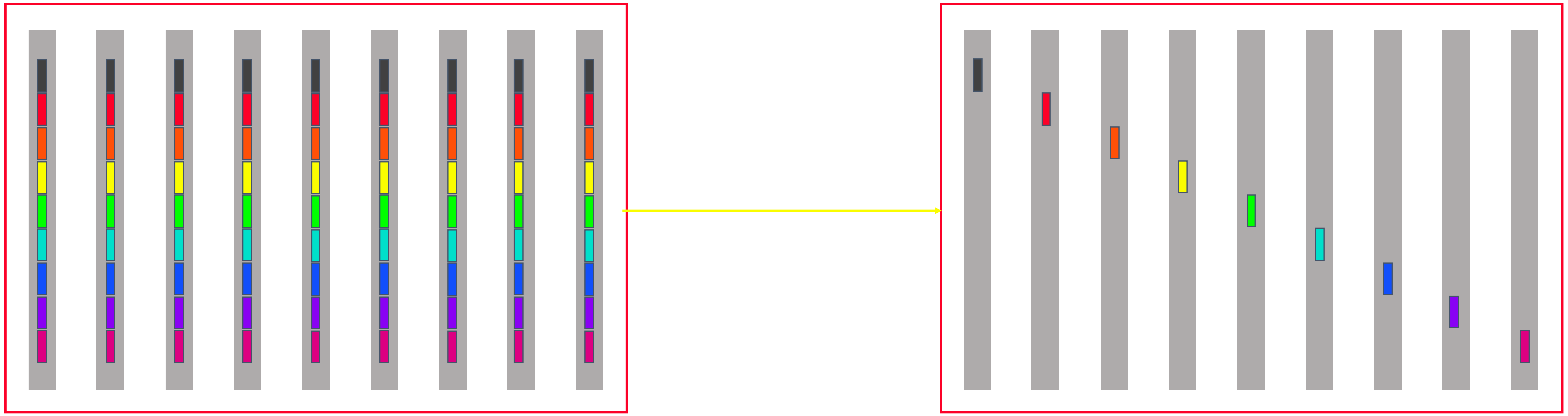
- Assumption: power of two number of nodes

gather $$\log(p)\alpha + \frac{p-1}{p} n\beta$$

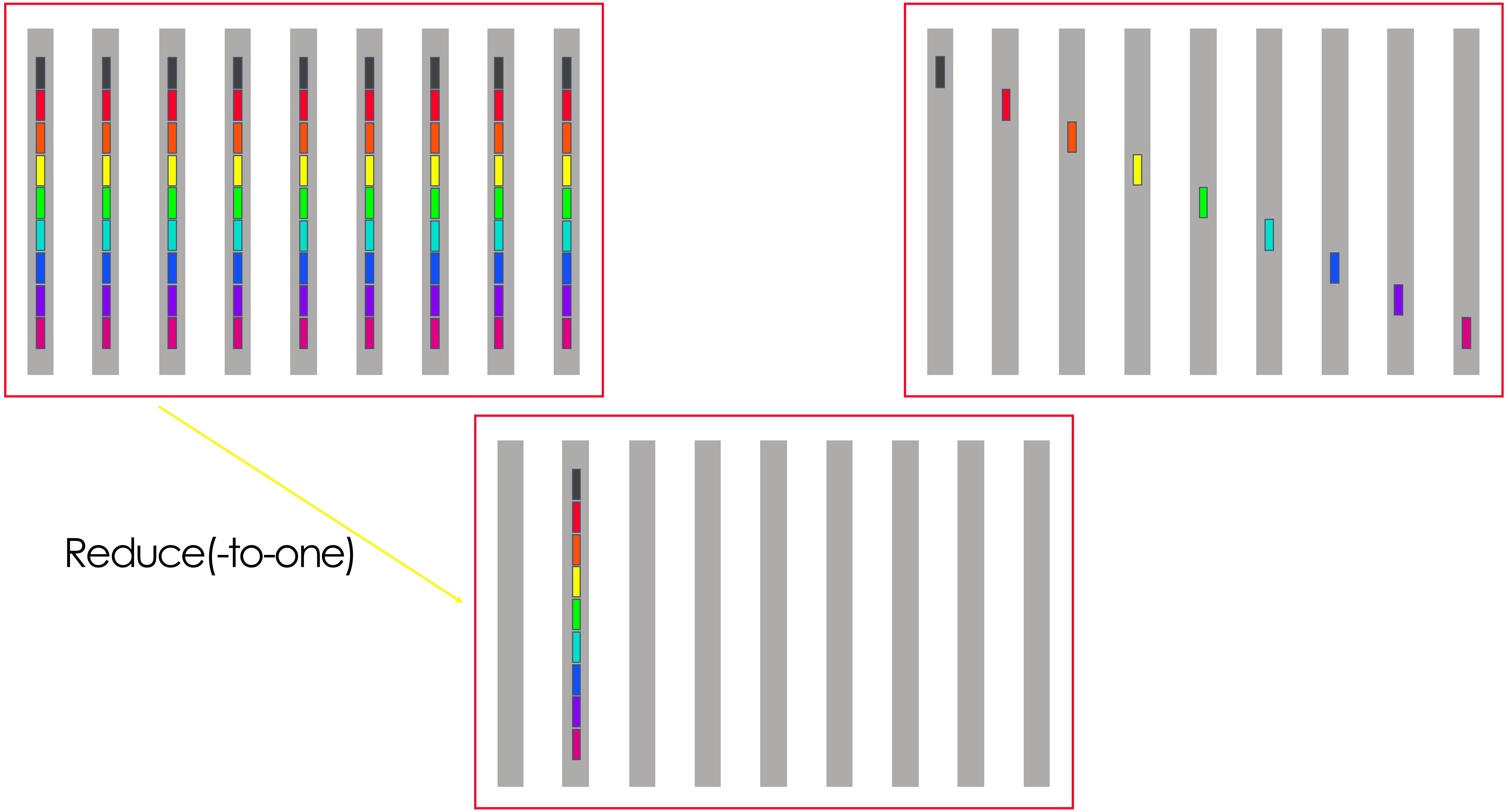
broadcast $$\log(p)(\alpha + n\beta)$$

$$2\log(p)\alpha + \left(\frac{p-1}{p} + \log(p)\right) n\beta$$

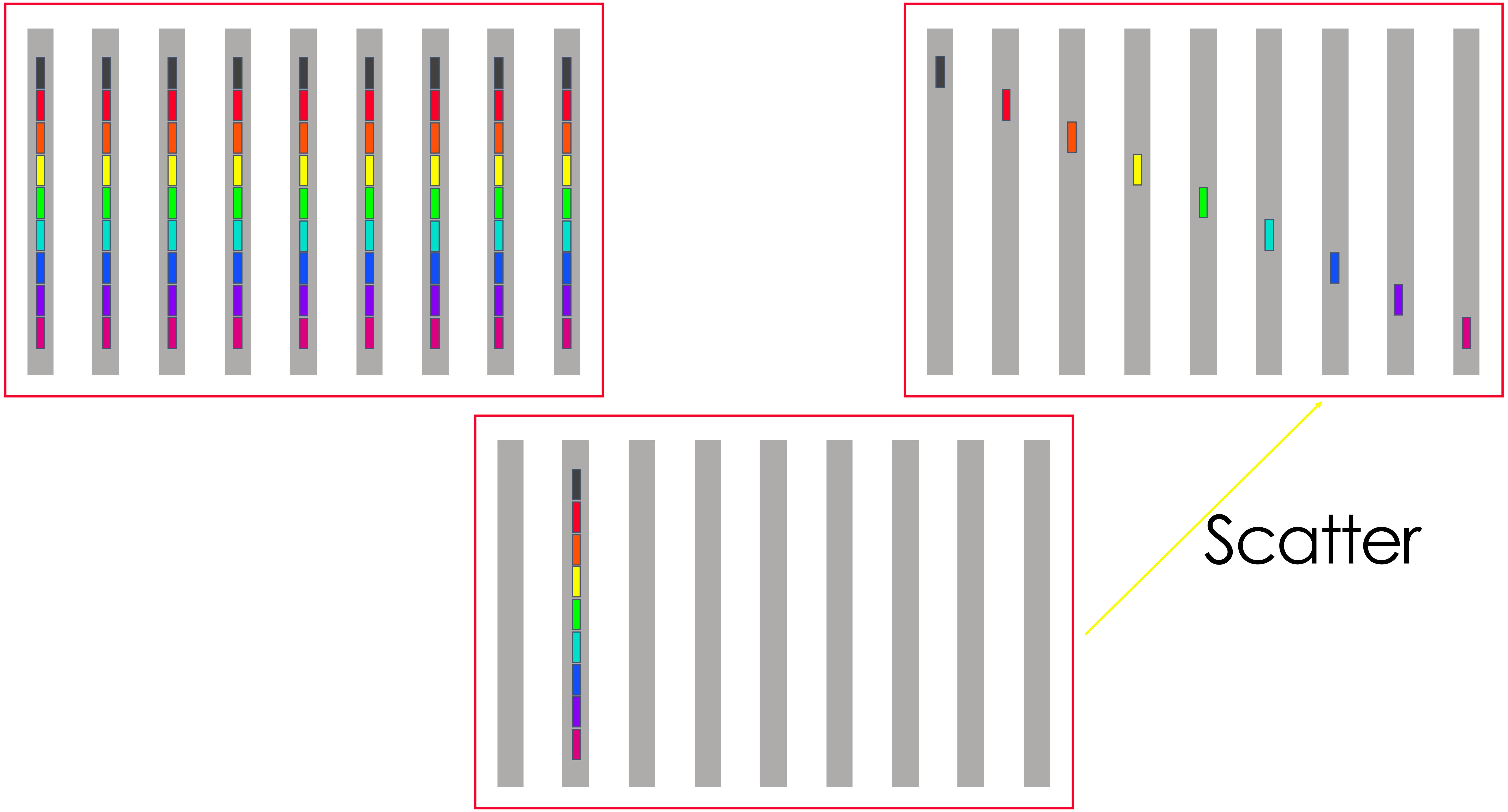
Reduce-scatter (small message)



Reduce-scatter (short vector)



Reduce-scatter (short vector)



Cost of Reduce(-to-one)/scatter Reduce-scatter

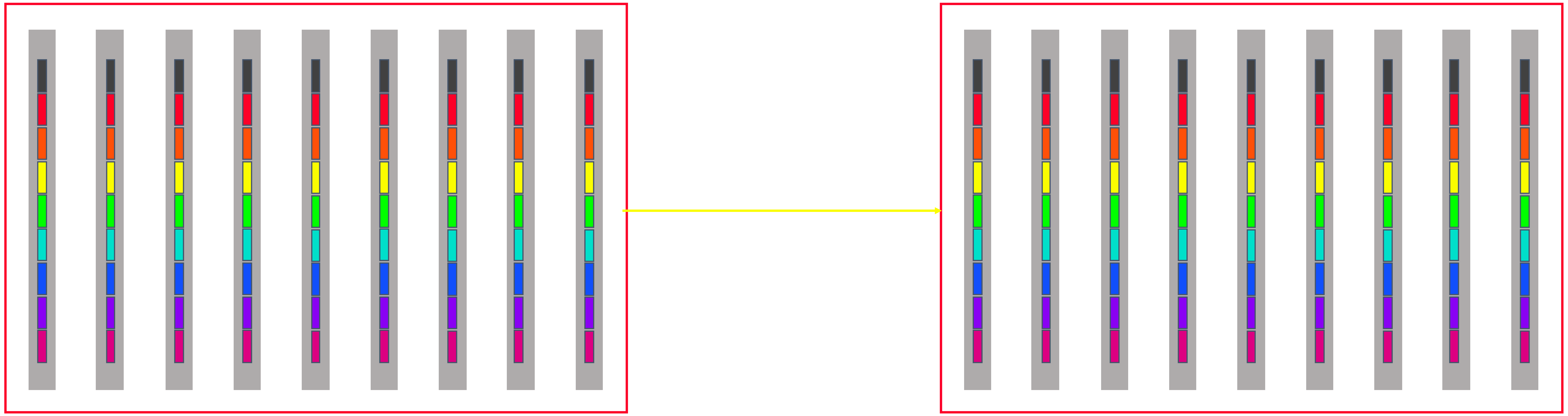
- Assumption: power of two number of nodes

Reduce(-to-one) $$log(p)(\alpha + n\beta + n\gamma)$$

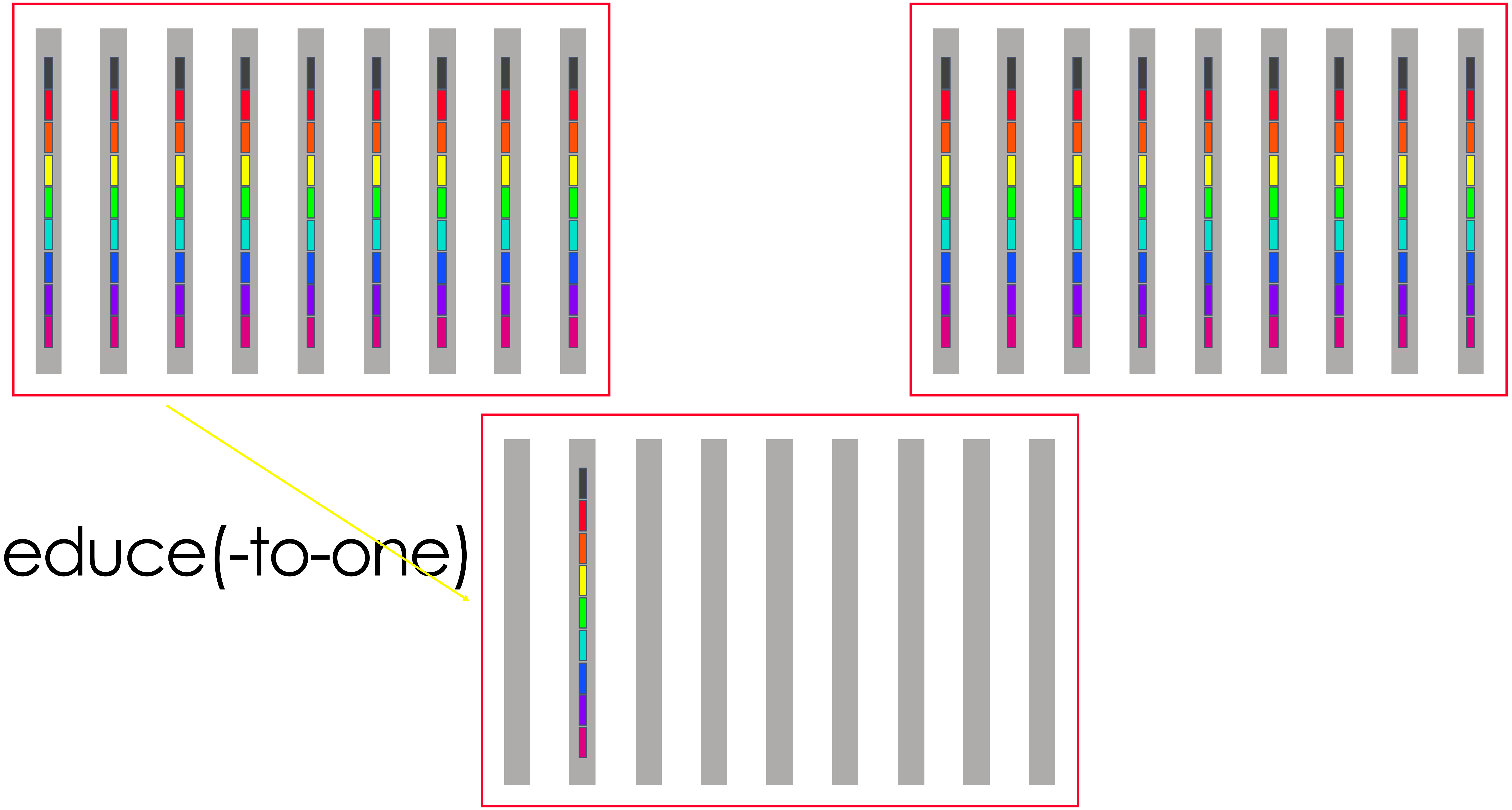
scatter $$log(p)\alpha + \frac{p-1}{p} n\beta$$

$$2log(p)\alpha + \left(\frac{p-1}{p} + log(p)\right) n\beta + log(p) n\gamma$$

Allreduce (Latency-optimized)

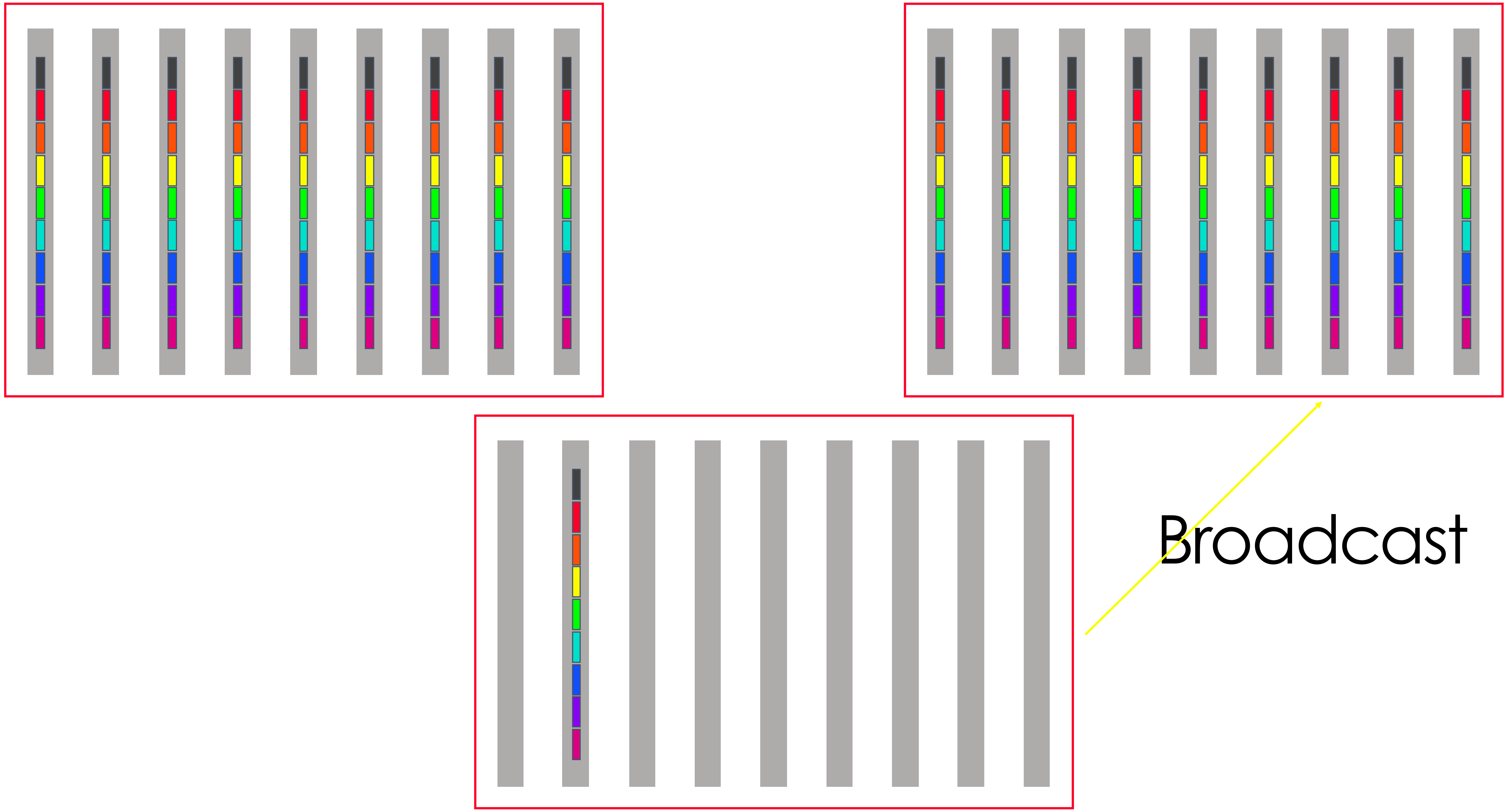


Allreduce (Latency-optimized)



Reduce(-to-one)

Allreduce
(short vector)



Cost of reduce(-to-one)/broadcast Allreduce

- Assumption: power of two number of nodes

Reduce(-to-one) $log(p)(\alpha + n\beta + n\gamma)$

broadcast $log(p)(\alpha + n\beta)$

$$2log(p)\alpha + 2log(p)n\beta + log(p)n\gamma$$

Recap

Reduce(-to-one)

$$log(p)(\alpha + n\beta + n\gamma)$$

Scatter

$$log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$log(p)\alpha + \frac{p-1}{p} n\beta$$

Broadcast

$$log(p)(\alpha + n\beta)$$

Reduce-scatter

Allreduce

Allgather

Recap

Reduce(-to-one)

$log(p)(\alpha + n\beta + n\gamma)$

Scatter

$log(p)\cancel{\alpha + \frac{p-1}{p} n\beta}$

Gather

$log(p)\cancel{\alpha + \frac{p-1}{p} n\beta}$

Broadcast

$log(p)(\alpha + n\beta)$

Reduce-scatter

$2log(p)\alpha + log(p)n(\beta + \cancel{\gamma)} \cancel{+ \frac{p-1}{p} n\beta}$

Allreduce

Allgather

Recap

Reduce(-to-one)

$$\log(p)(\alpha + n\beta + n\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Broadcast

$$\log(p)(\alpha + n\beta)$$

Reduce-scatter

$$2\log(p)\alpha + \log(p)n(\beta + \gamma) + \frac{p-1}{p} n\beta$$

Allreduce

$$2\log(p)\alpha + \log(p)n(2\beta + \gamma)$$

Allgather

$$2\log(p)\alpha + \log(p)n\beta + \frac{p-1}{p} n\beta$$

Recap

Reduce(-to-one)

$log(p)(\alpha + n\beta + n\gamma)$

Scatter

$log(p)\alpha + \frac{p-1}{p} n\beta$

Gather

$log(p)\alpha + \frac{p-1}{p} n\beta$

Broadcast

$log(p)(\alpha + n\beta)$

Reduce-scatter

$2log(p)\alpha + log(p)n(\beta + \gamma) + \frac{p-1}{p} n\beta$

Allreduce

$2log(p)\alpha + log(p)n(2\beta + \gamma)$

Allgather

Recap

Reduce(-to-one)

$$\log(p)(\alpha + n\beta + n\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p}n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p}n\beta$$

Broadcast

$$\log(p)(\alpha + n\beta)$$

Reduce-scatter

$$2\log(p)\alpha + \log(p)n(\beta + \gamma) + \frac{p-1}{p}n\beta$$

Allreduce

$$2\log(p)\alpha + \log(p)n(2\beta + \gamma)$$

Allgather

$$2\log(p)\alpha + \log(p)n\beta + \frac{p-1}{p}n\beta$$

Summary of MST algorithms

- Small message: Minimum Spanning Tree algorithm
  - Emphasize **low latency**
- **Can we do better?**
- Problem of Minimum Spanning Tree Algorithm?
  - It prioritize latency rather than bandwidth
  - Hence: Some links are idle
- Next: Large message size algorithm

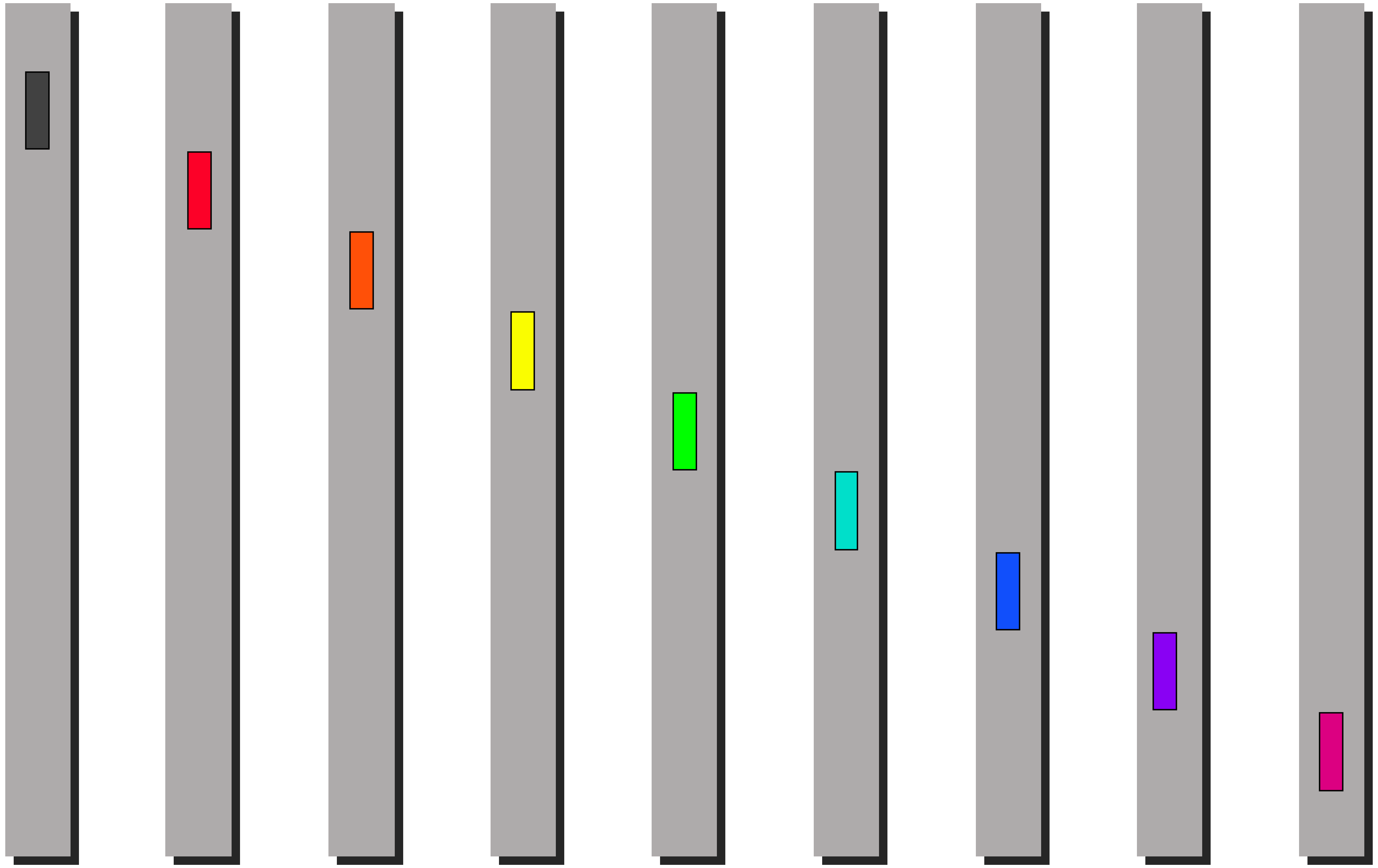
Large Message

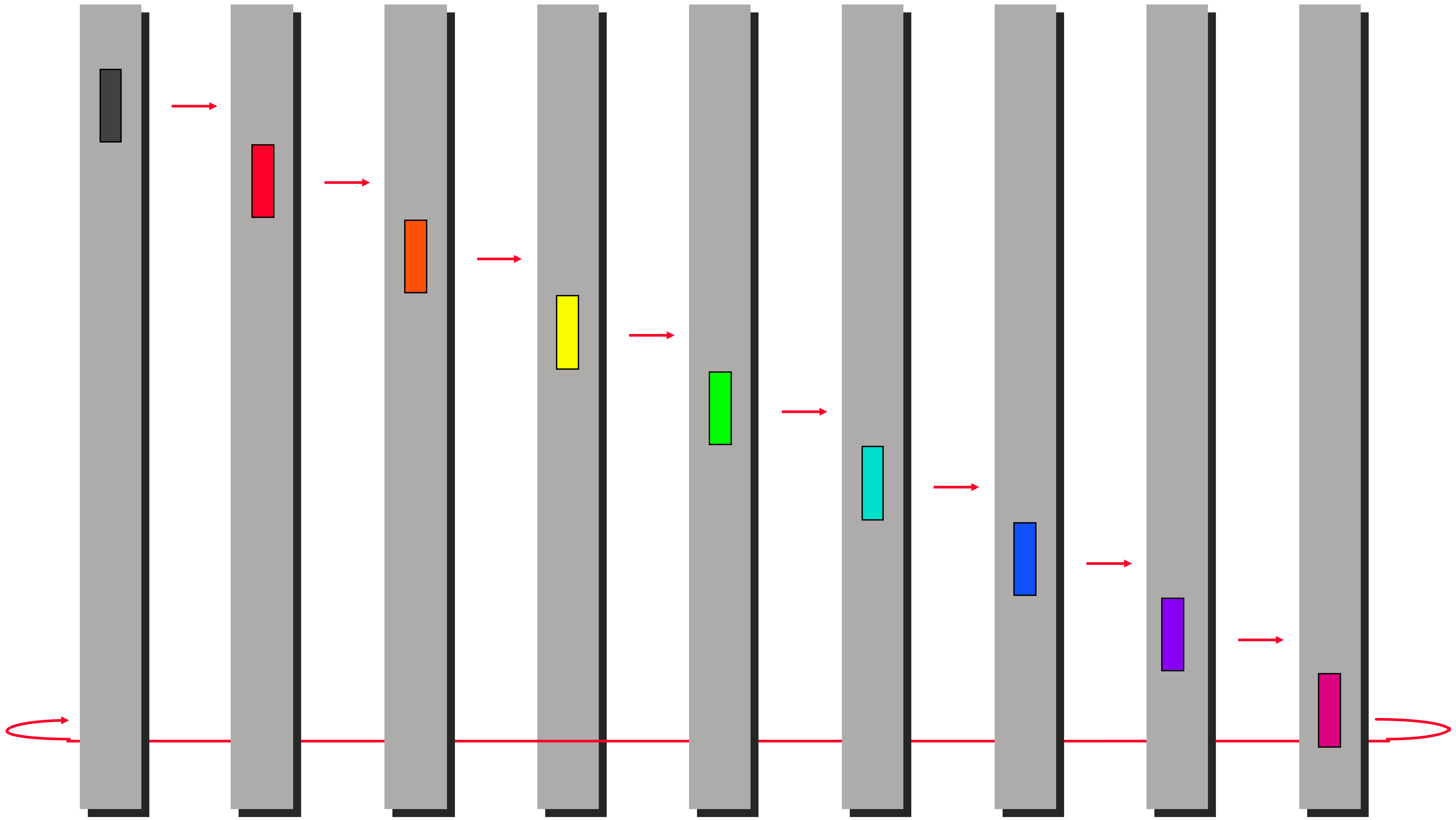
Communication Model: $\alpha + n\,\beta, \beta = \frac{1}{B}$

- The second term dominates – we want to minimize the second term
  - We want to utilize the bandwidth as much as possible

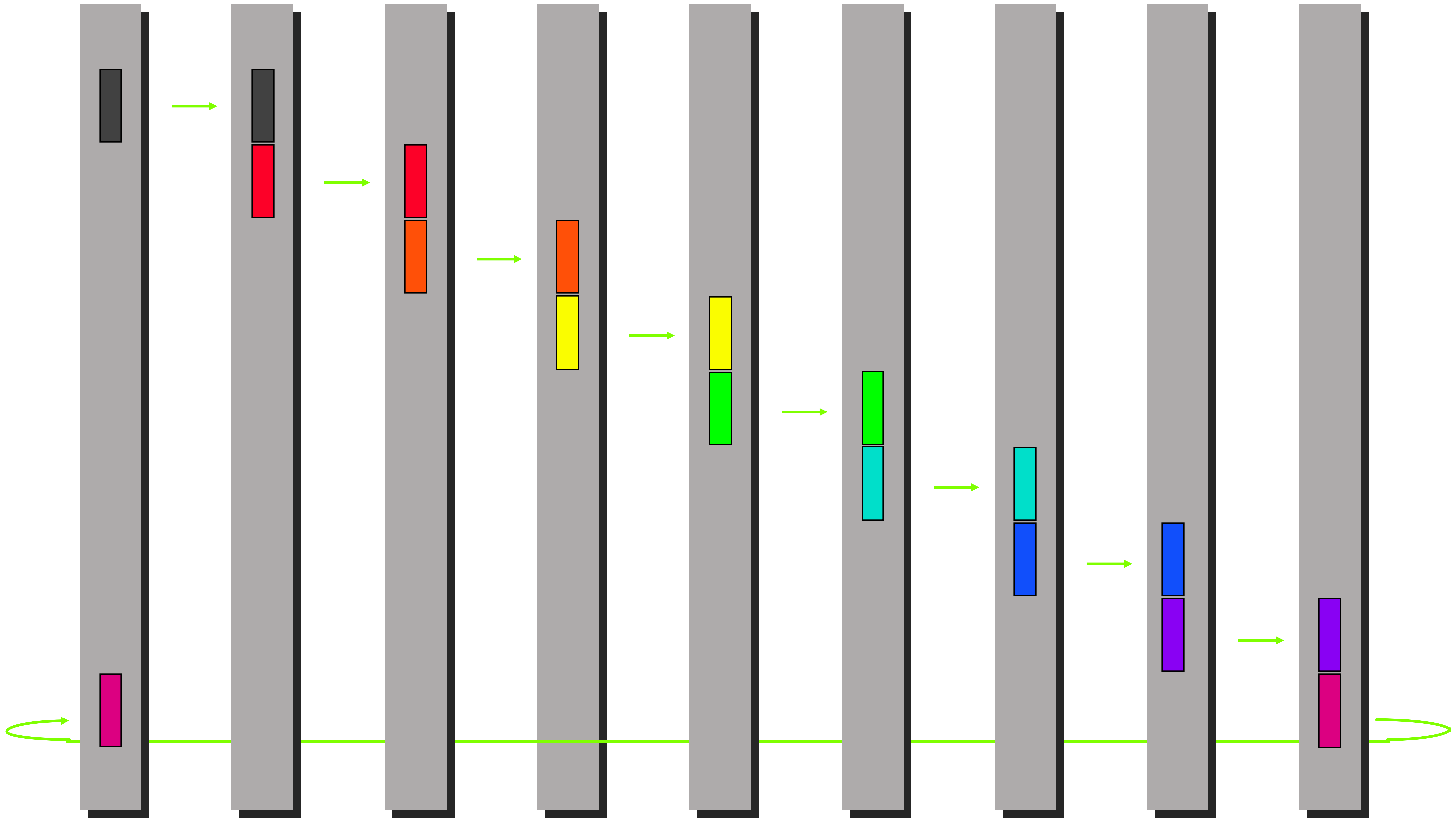
General principles

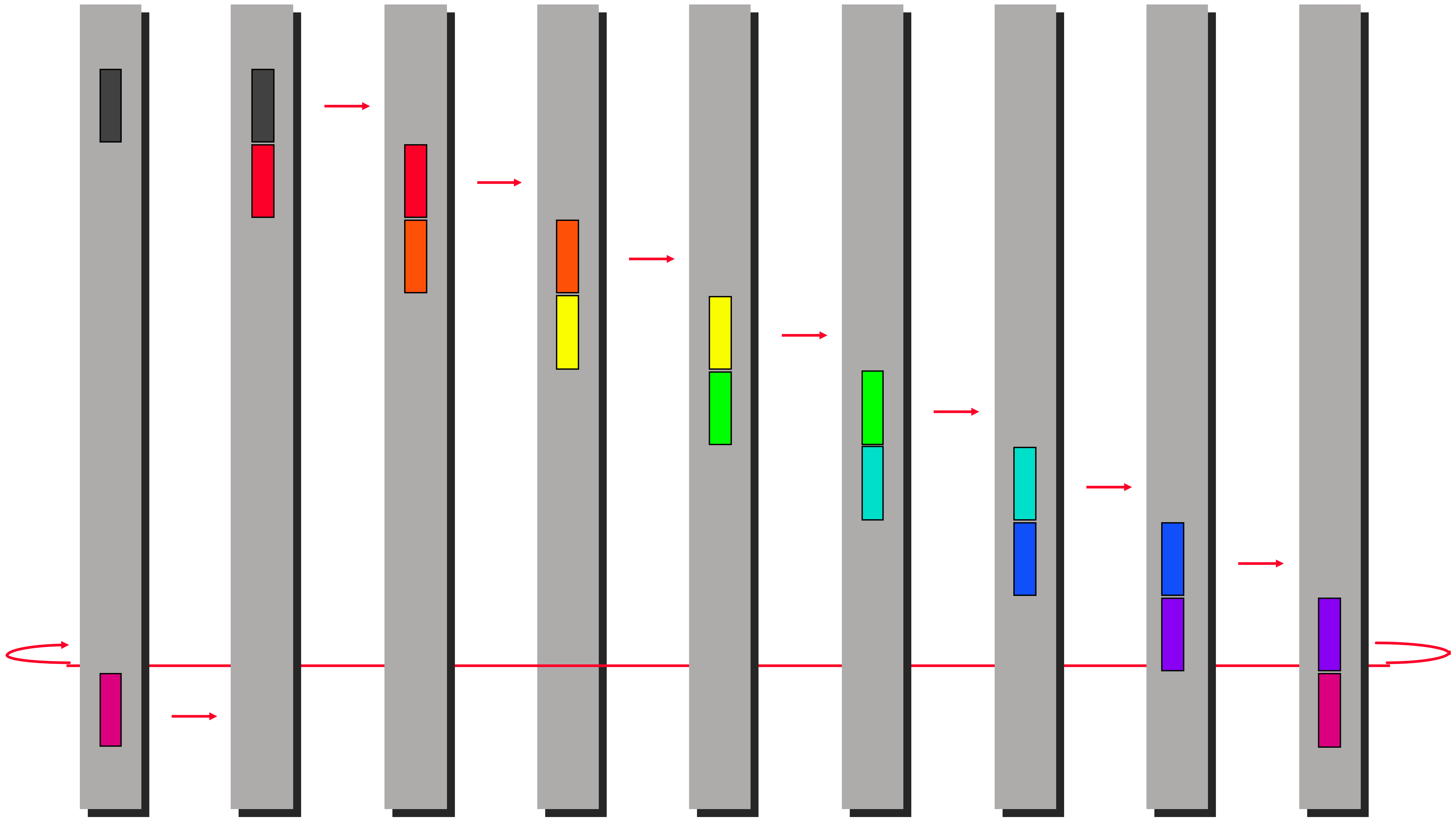
- Use all the links between every two nodes
- A logical ring can be embedded in a physical linear array with worm-hole routing, since the “wrap-around” message doesn’t conflict

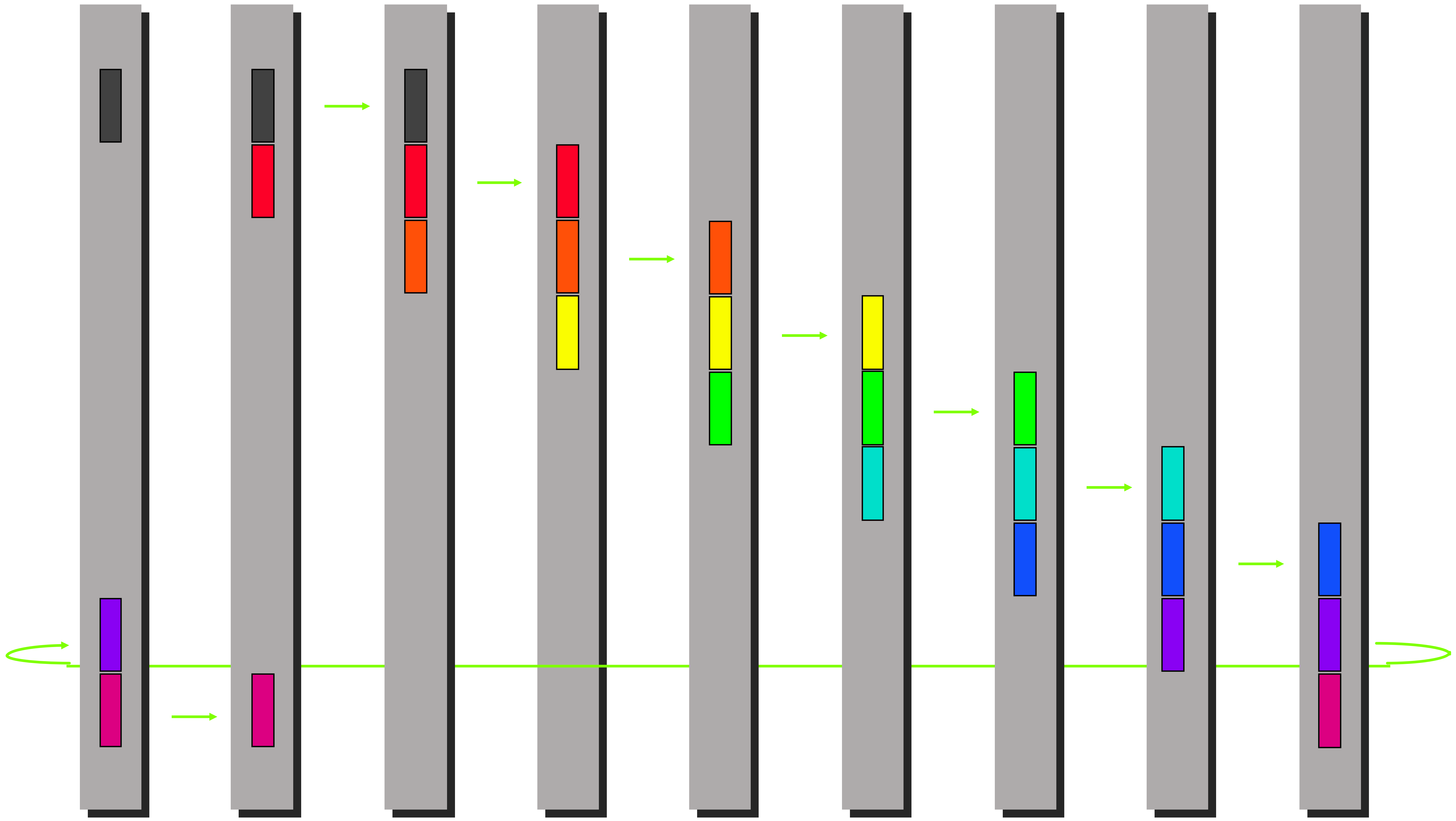




- A logical ring can be embedded in a physical linear array with worm-hole routing, since the “wrap-around” message doesn’t conflict





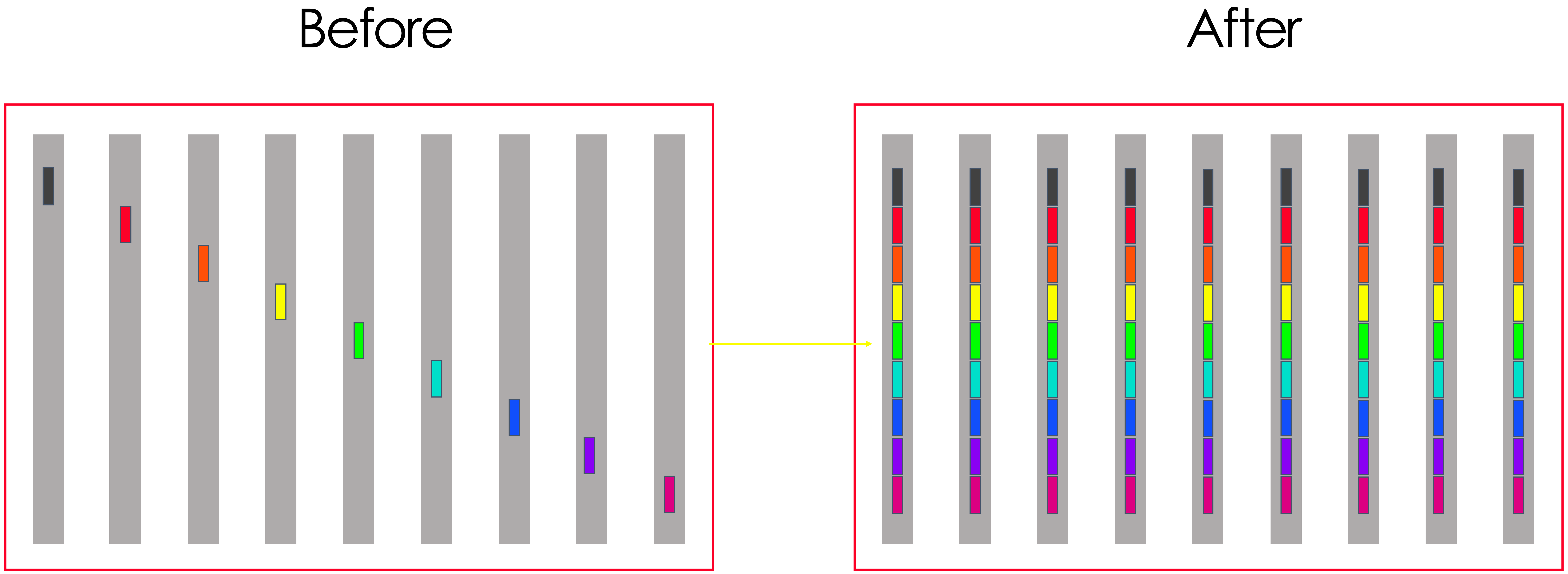


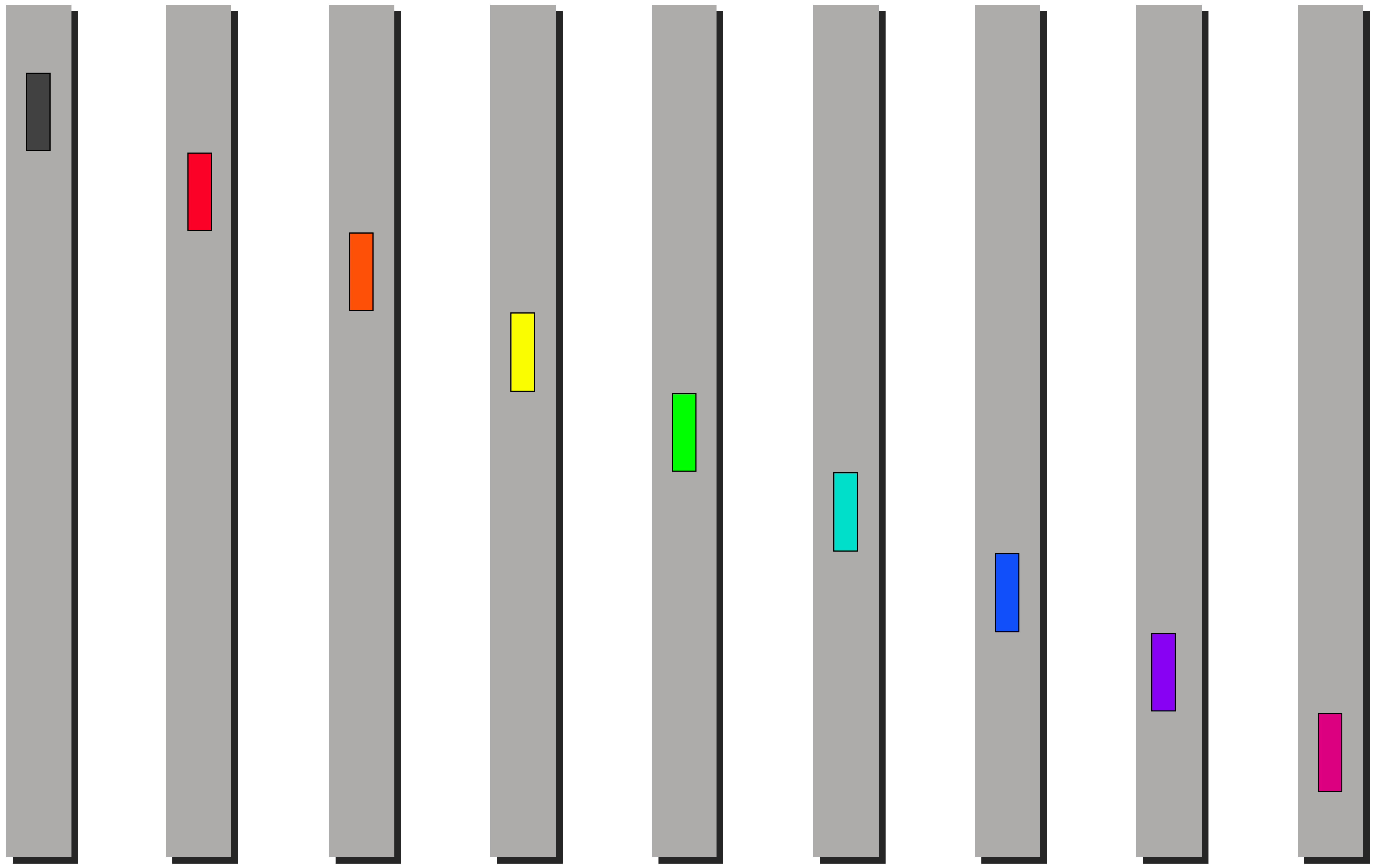
General principles

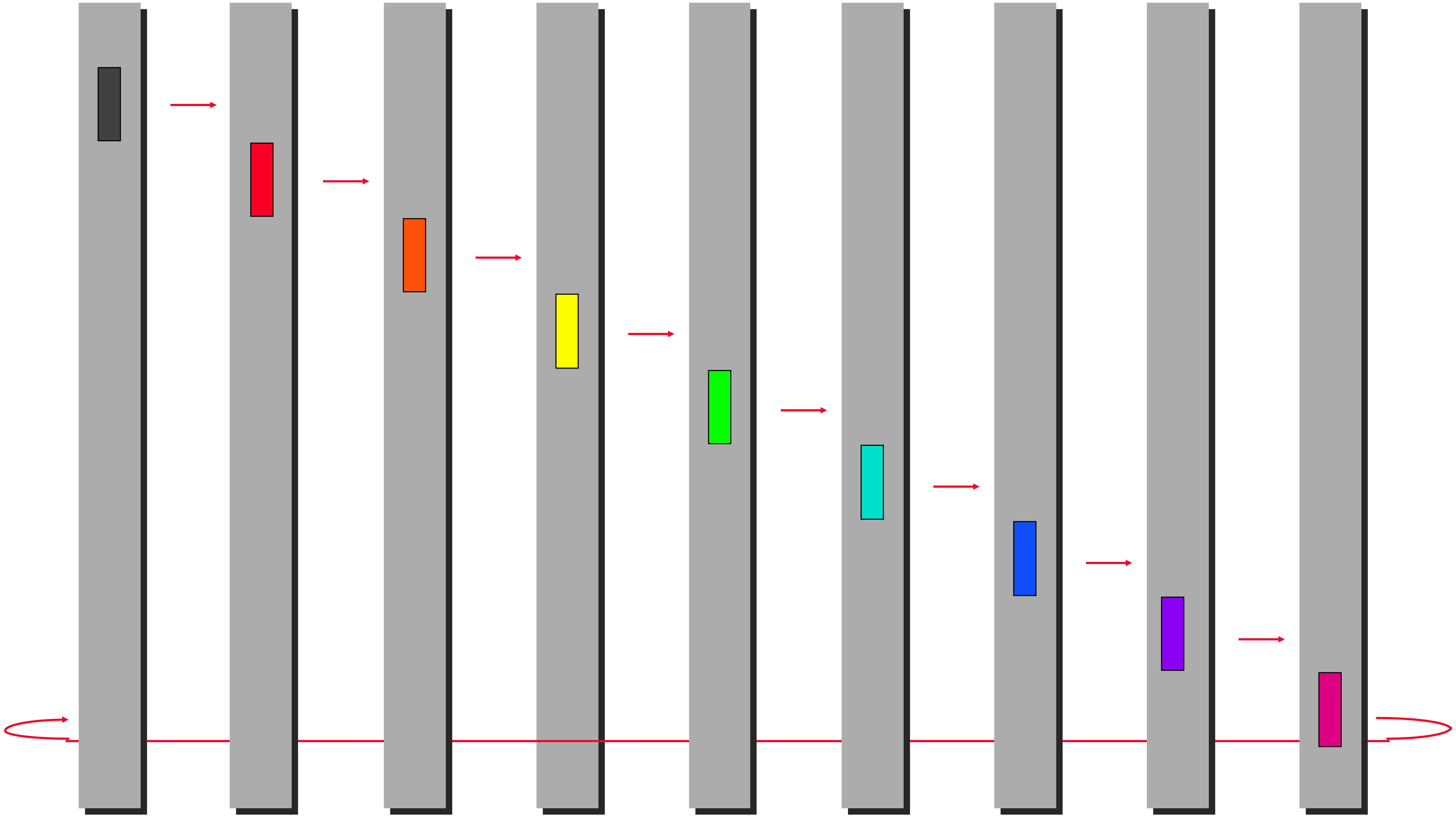
Ring algorithm has the following advantages

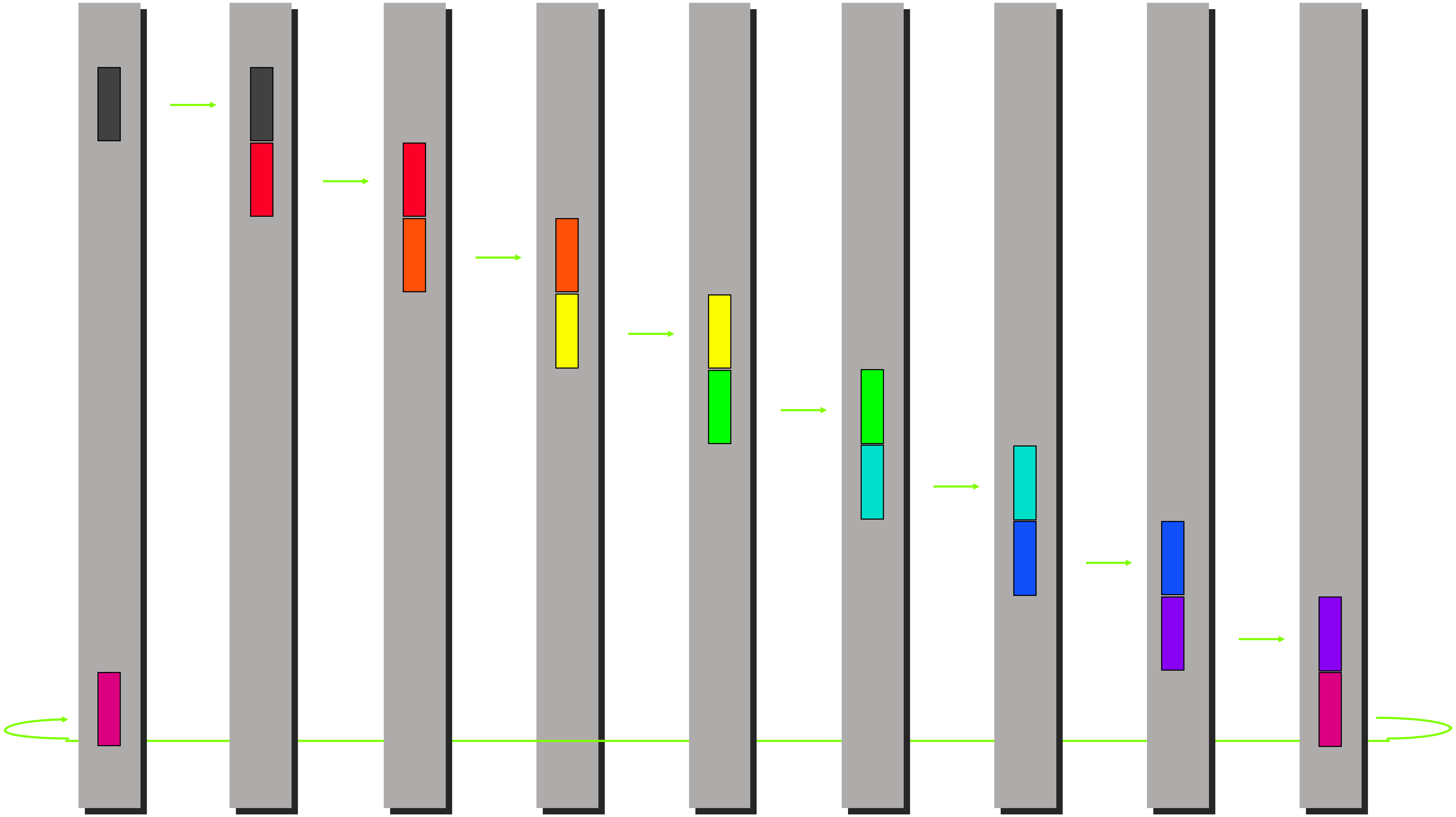
- Fully utilize the bandwidth (bandwidth optimal)
- implementation for arbitrary numbers of node

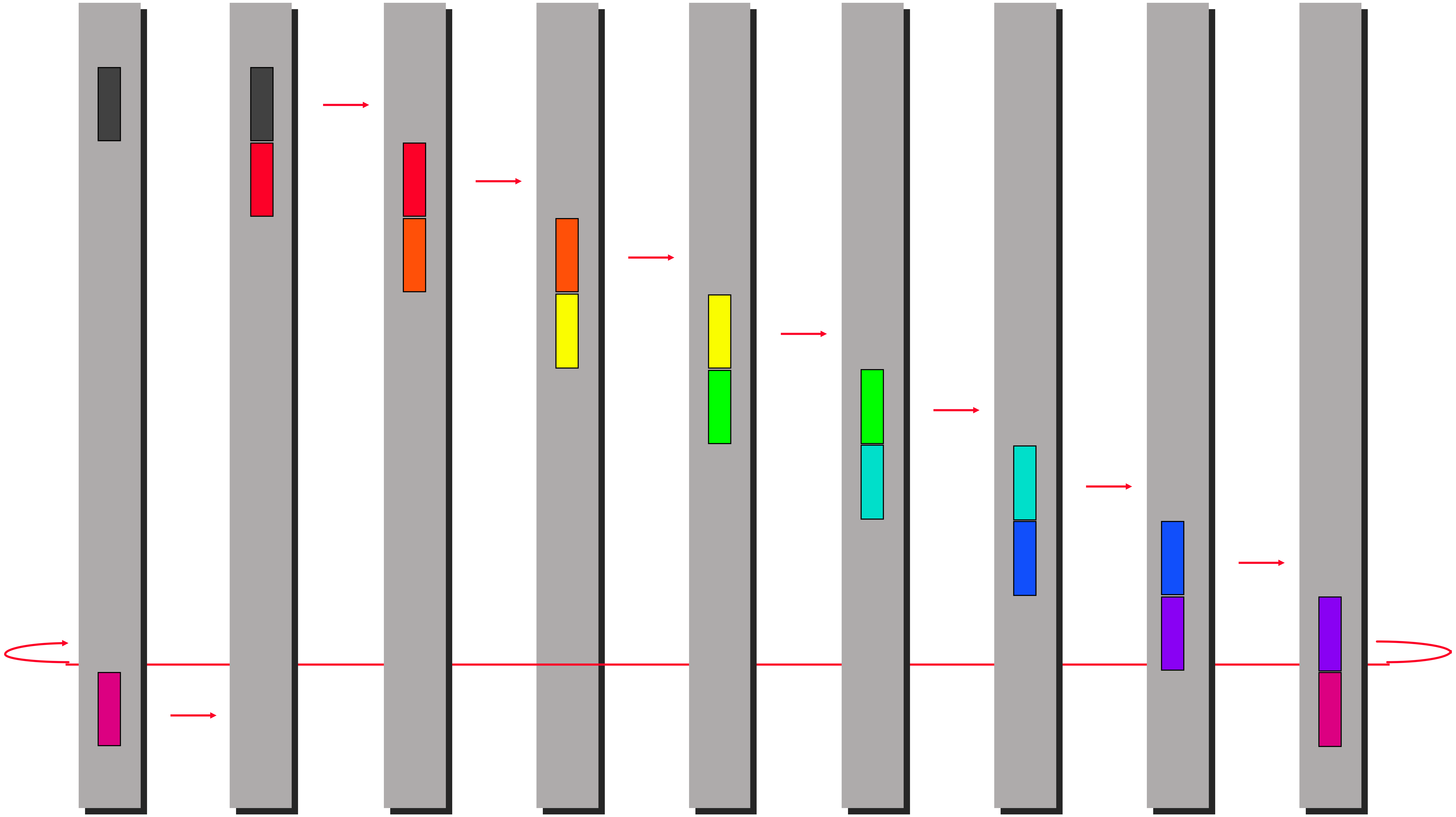
Allgather

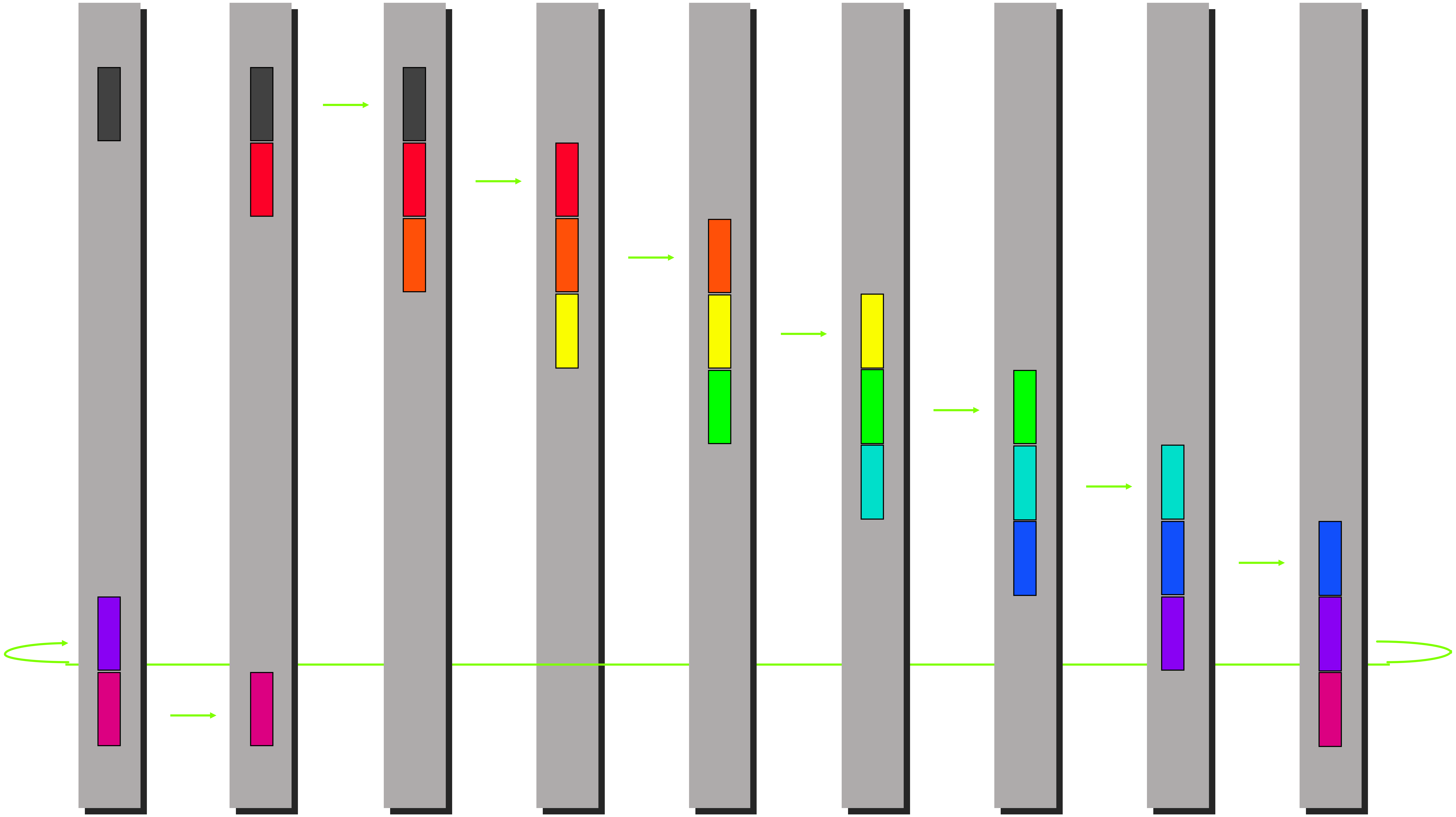


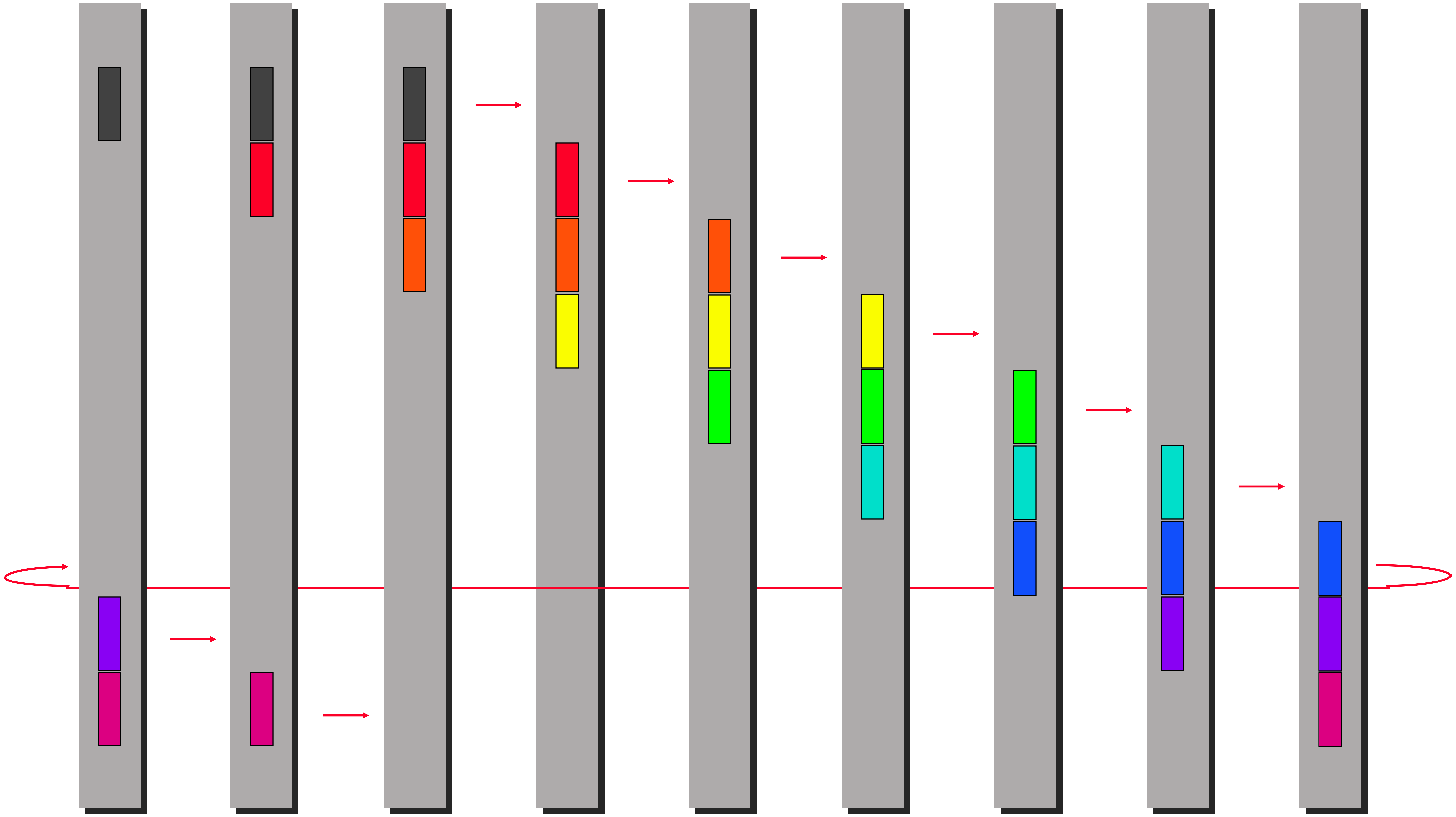


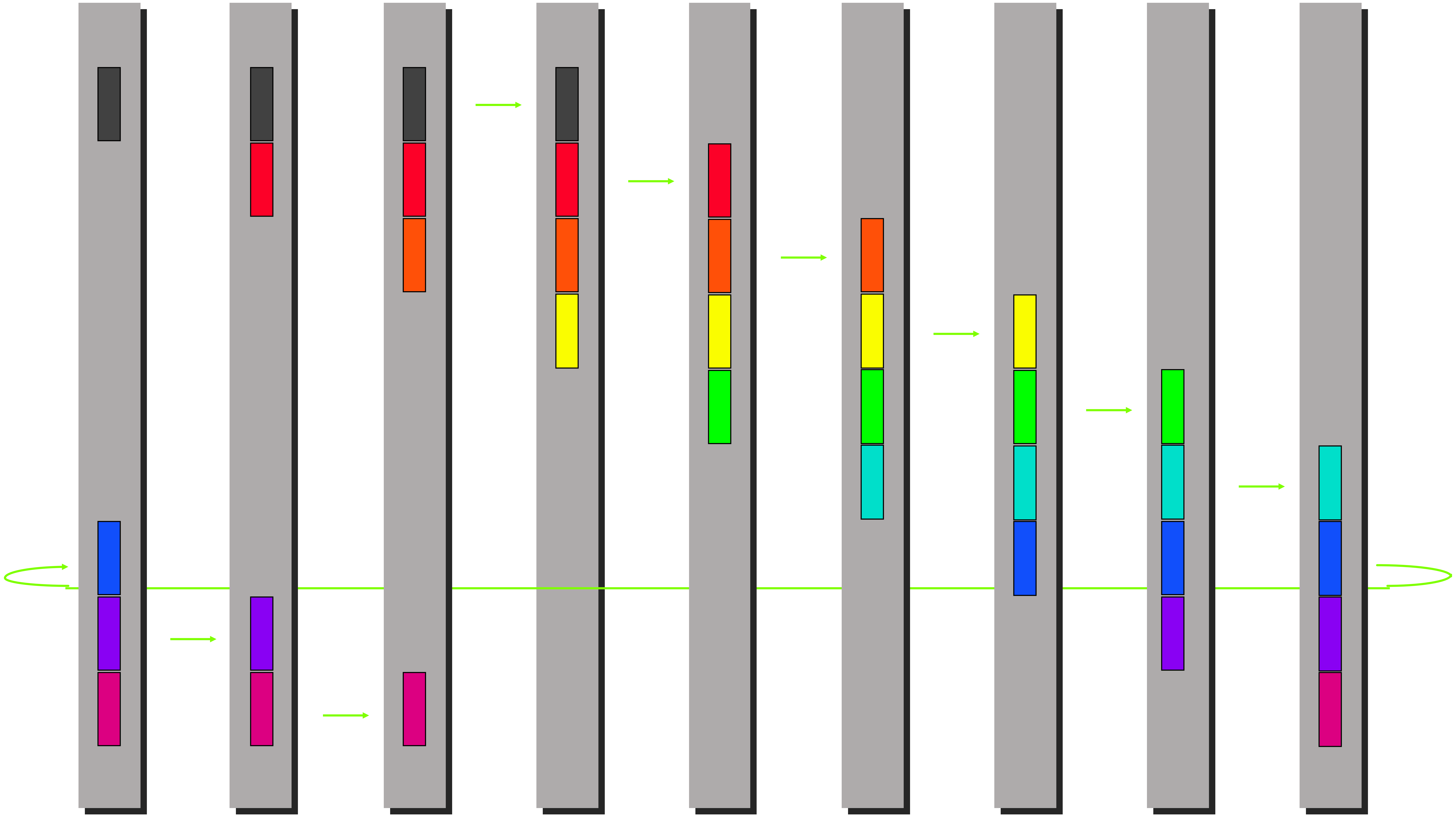


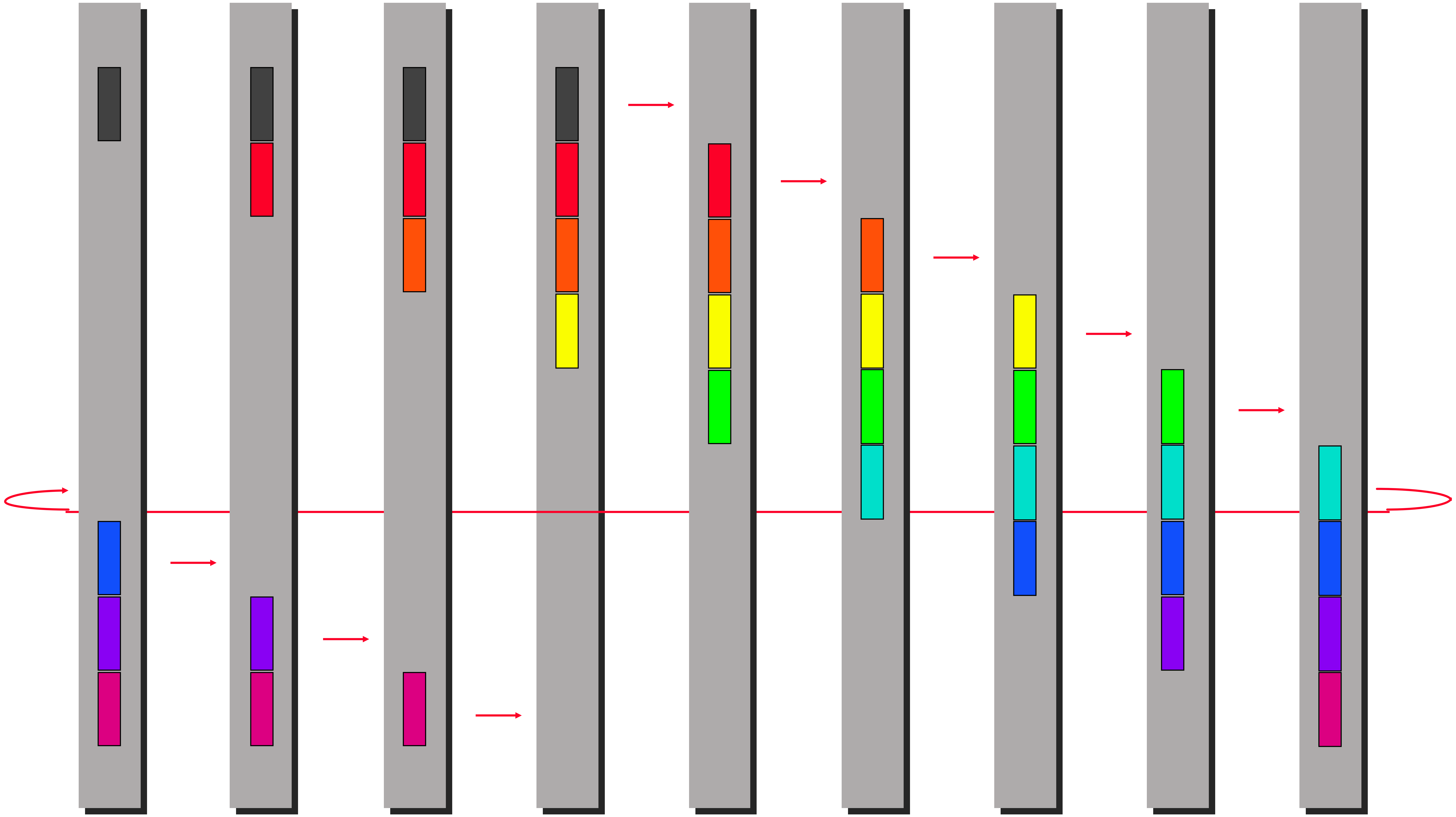


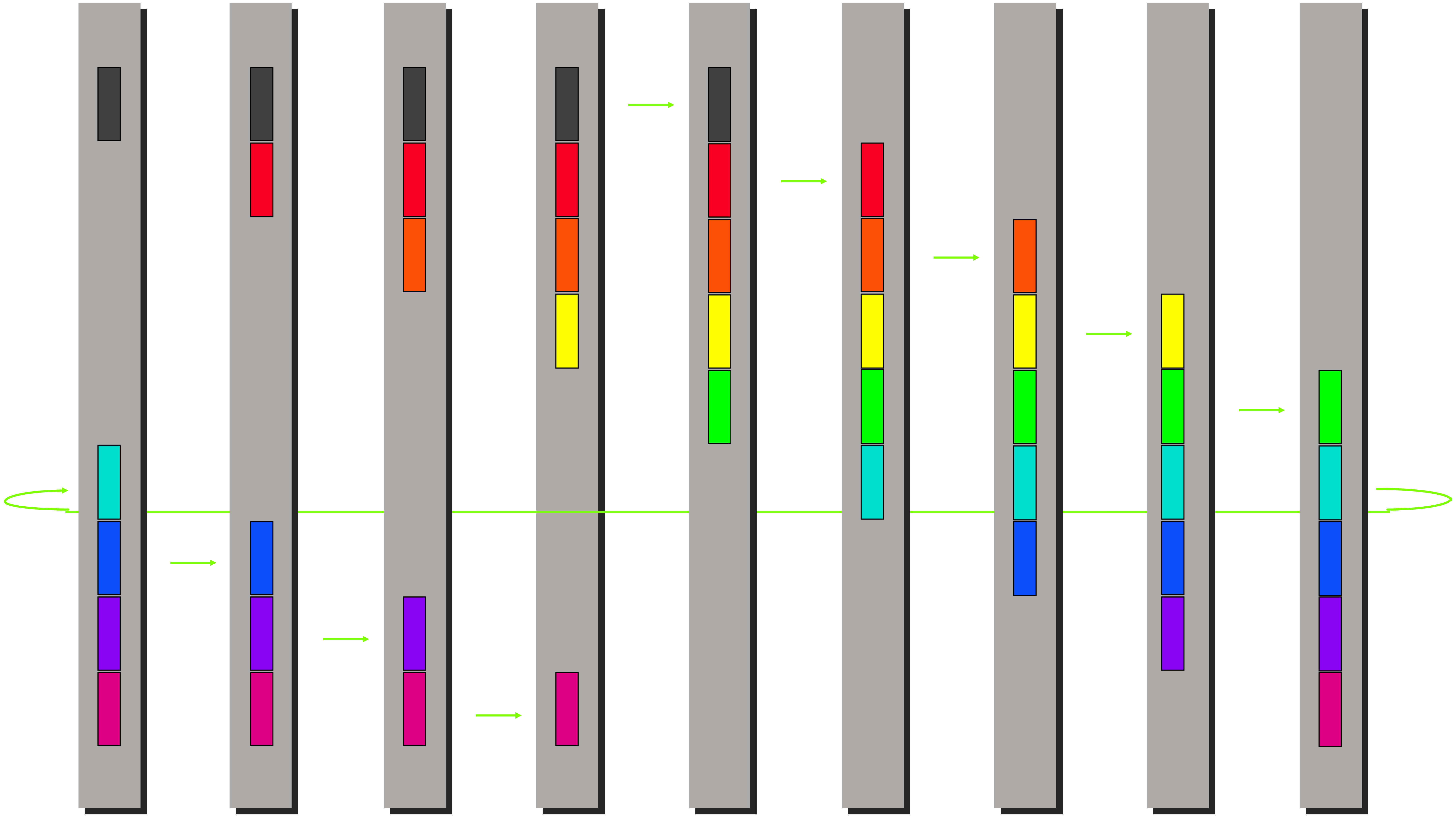


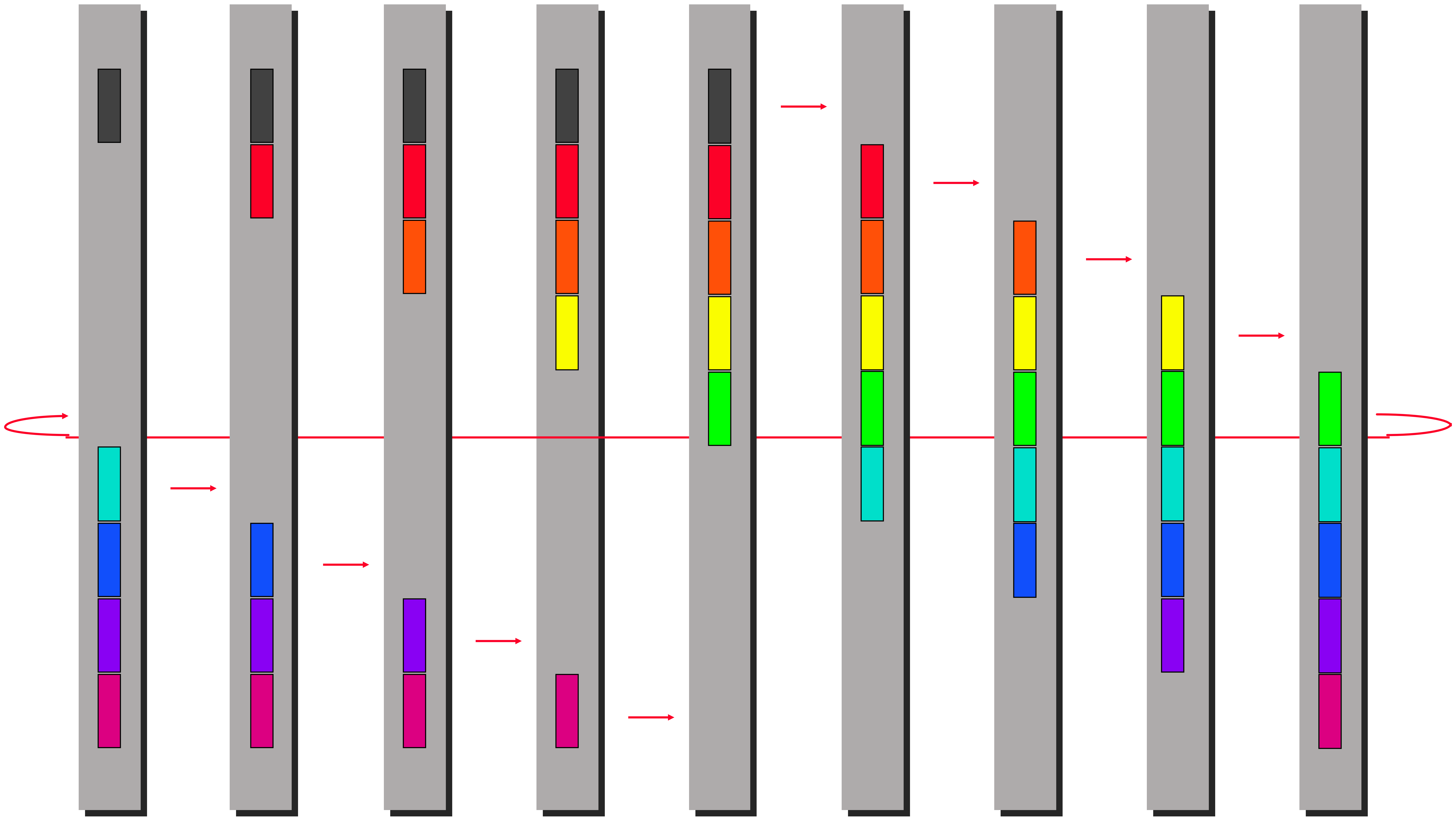


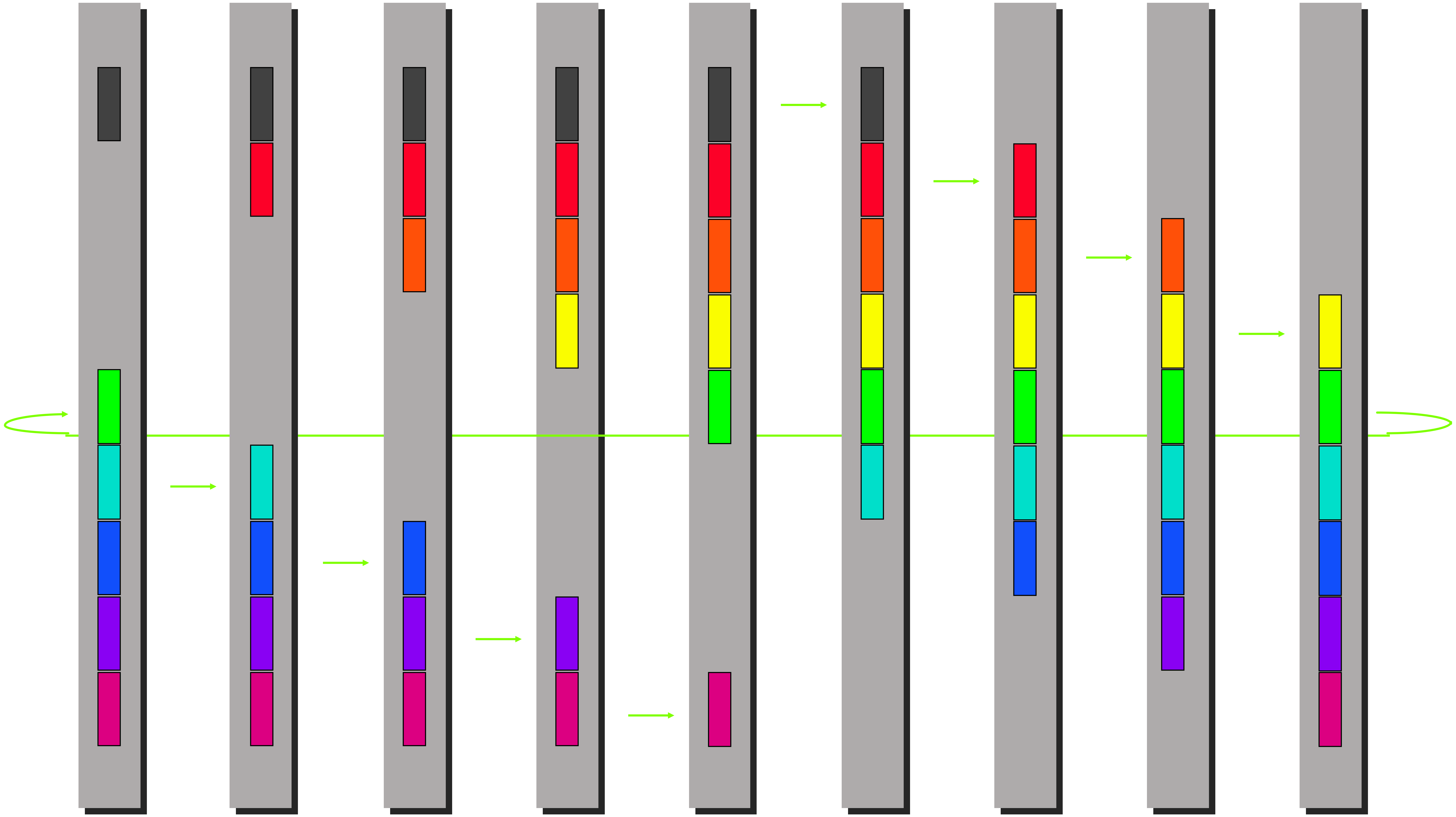


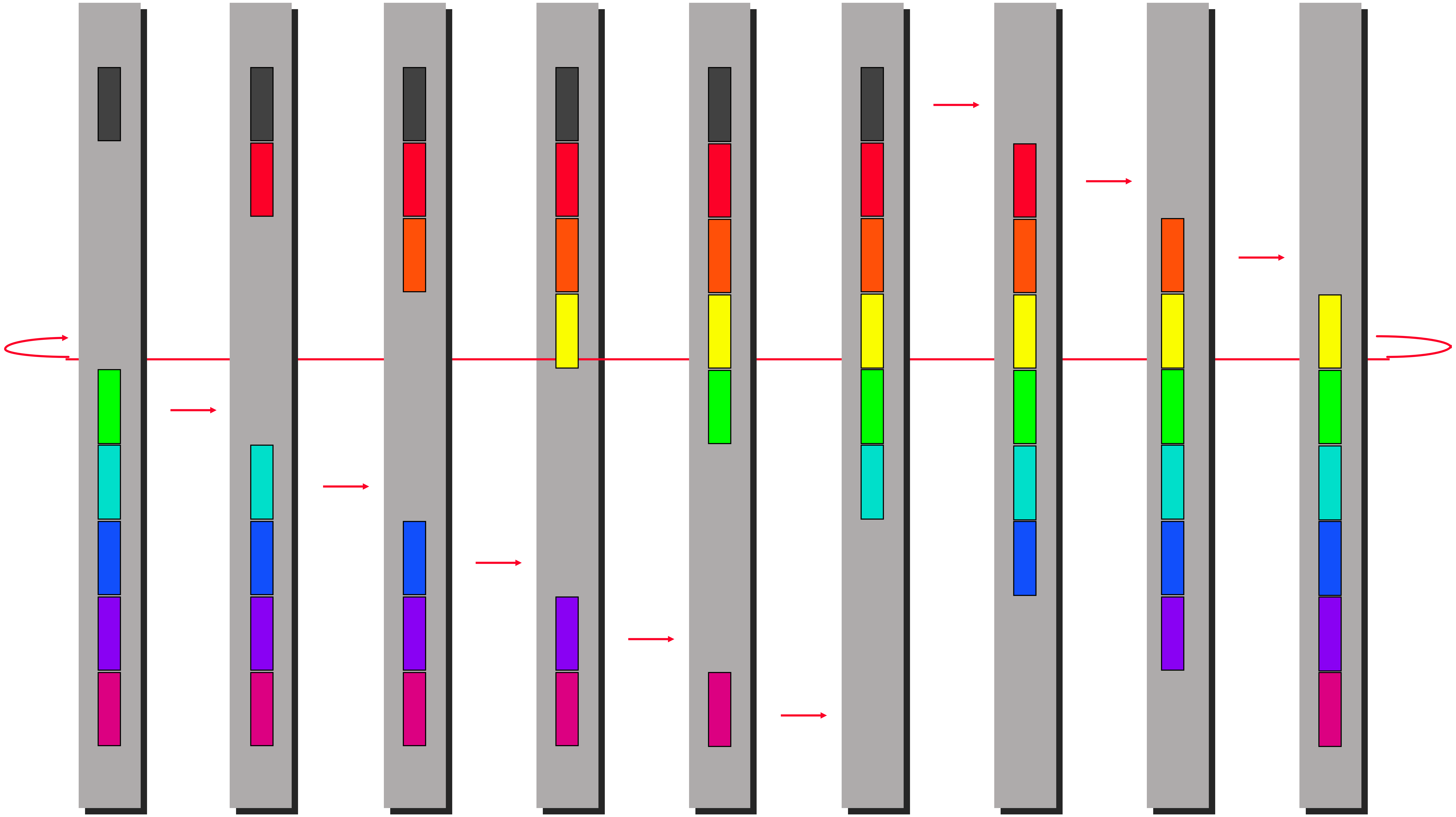


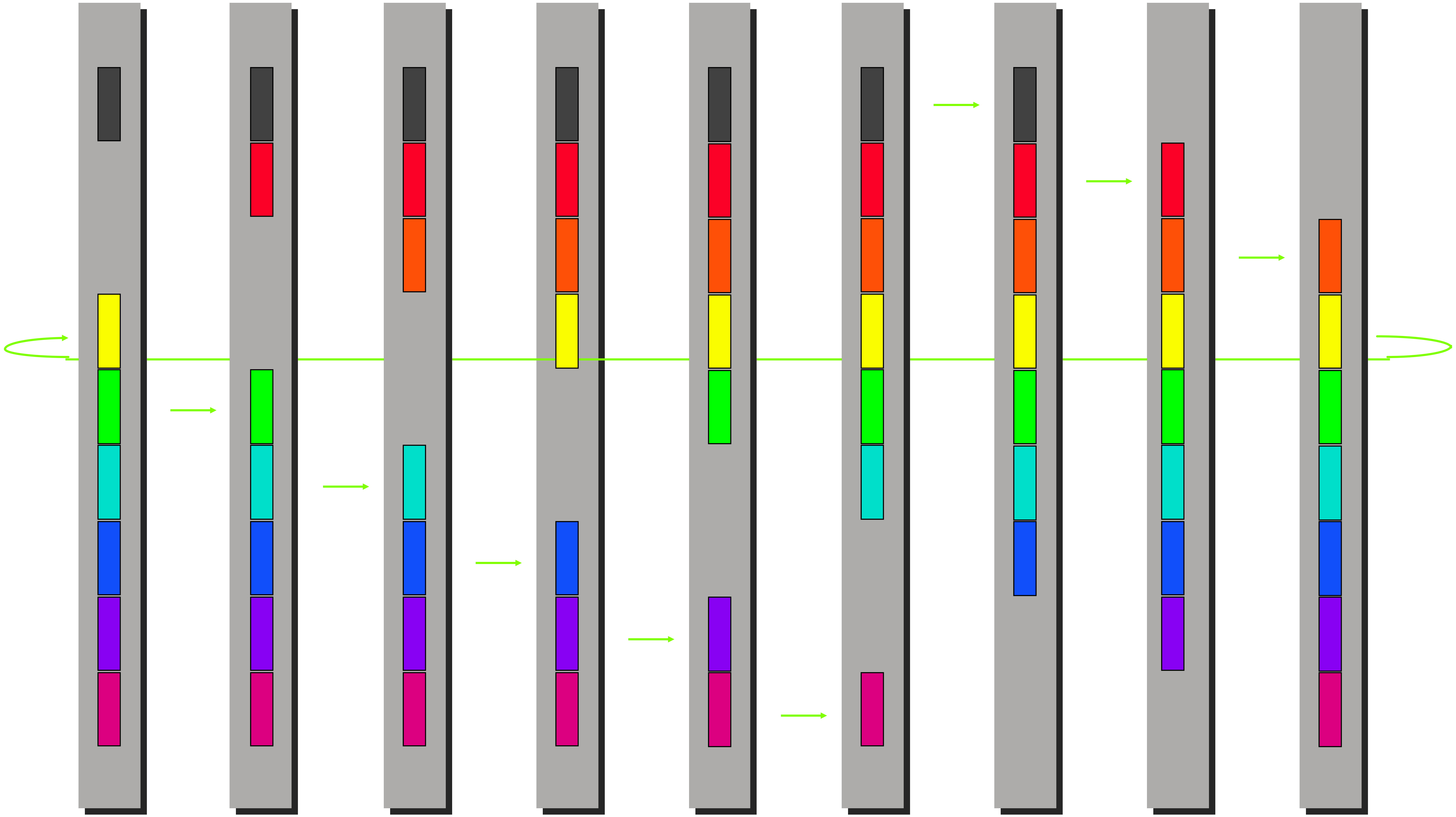


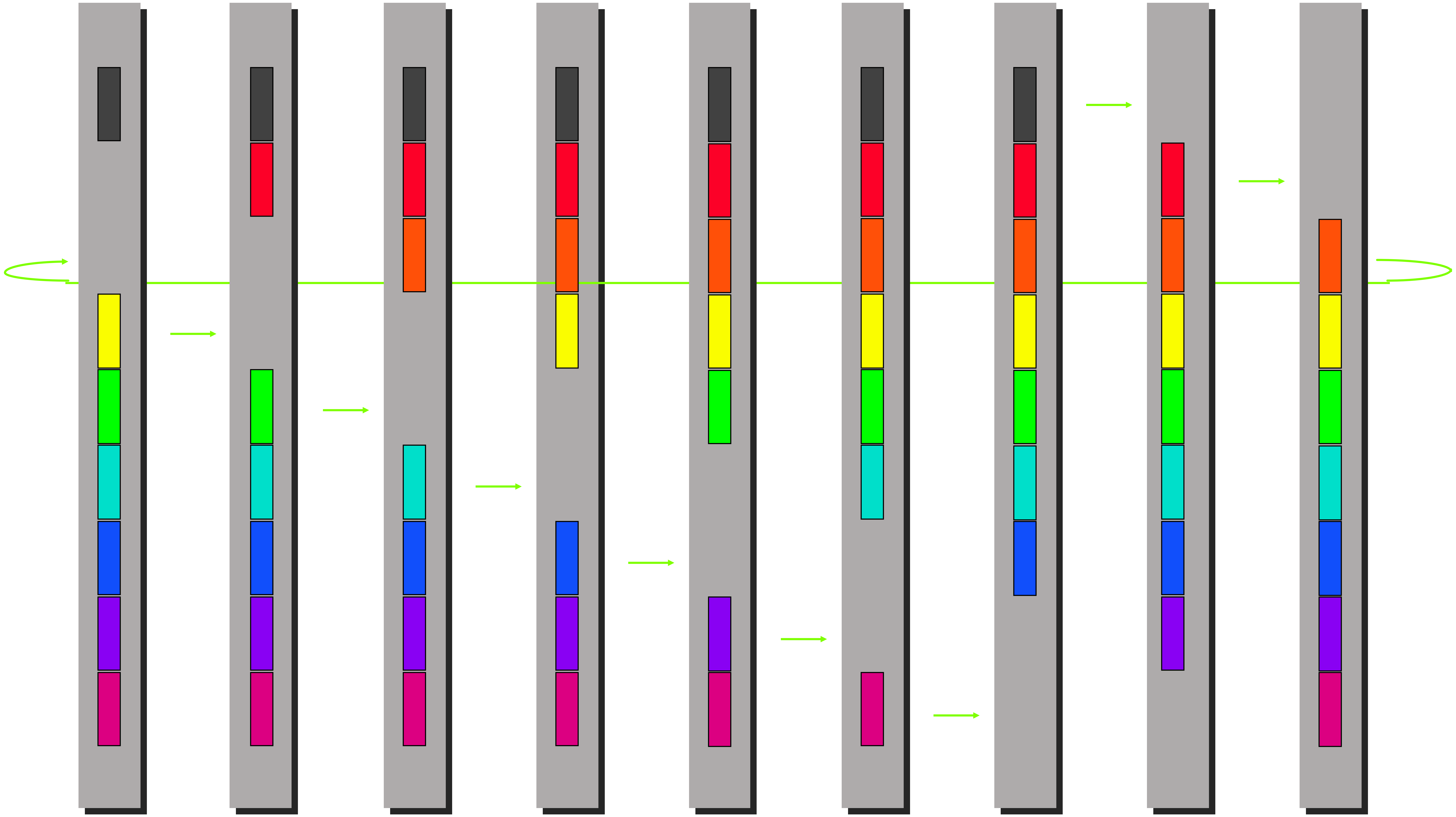


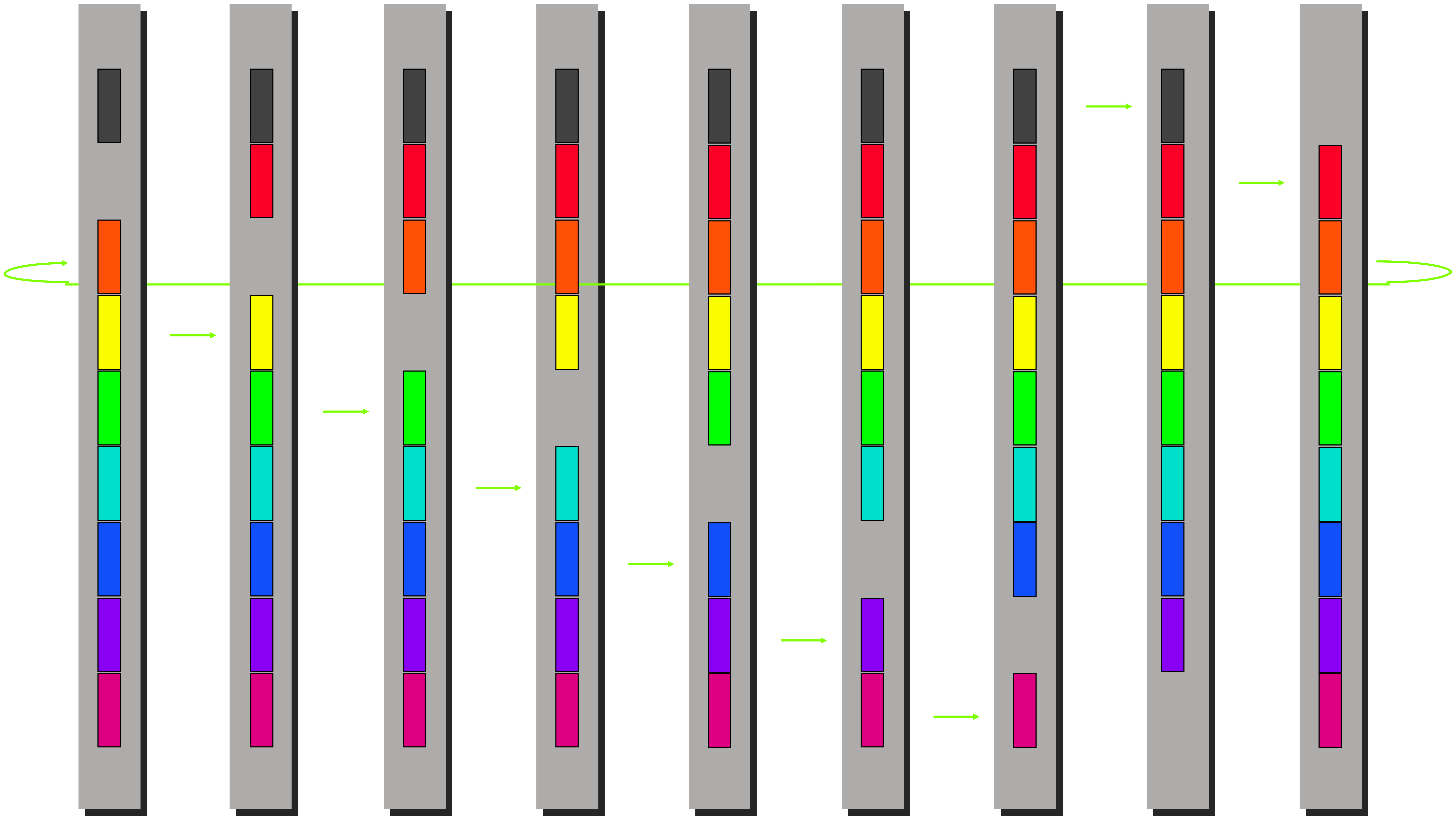


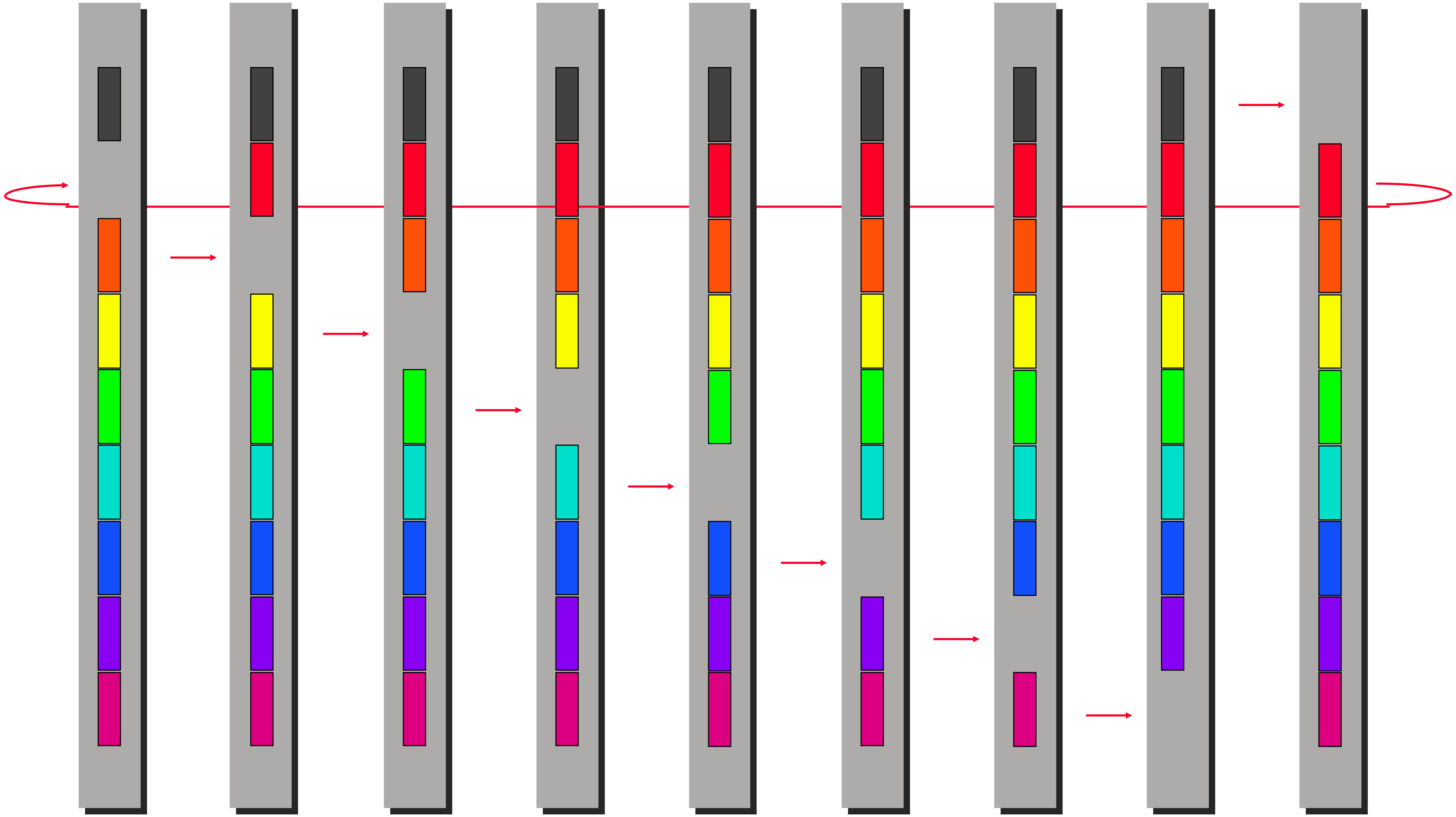


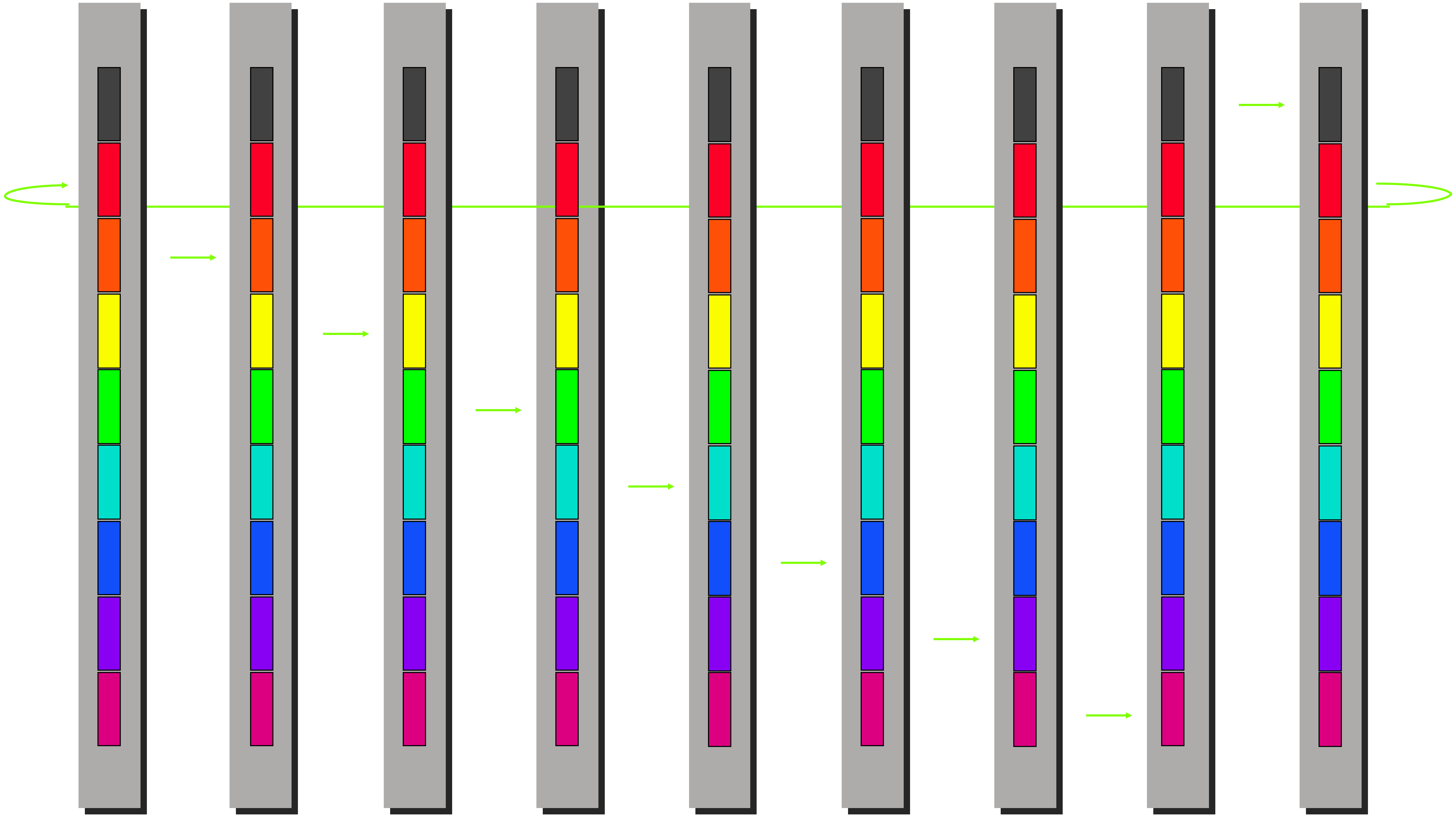


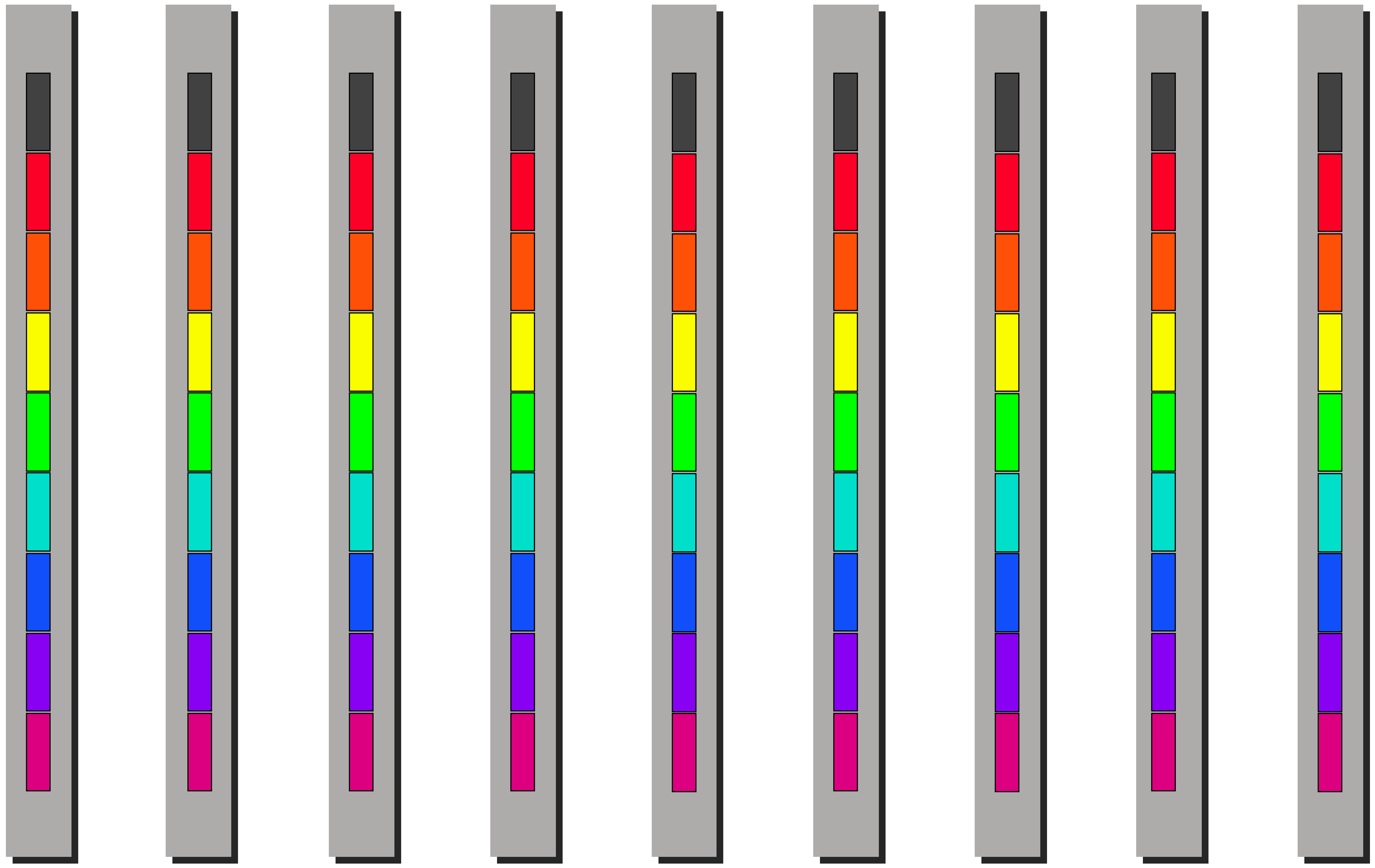












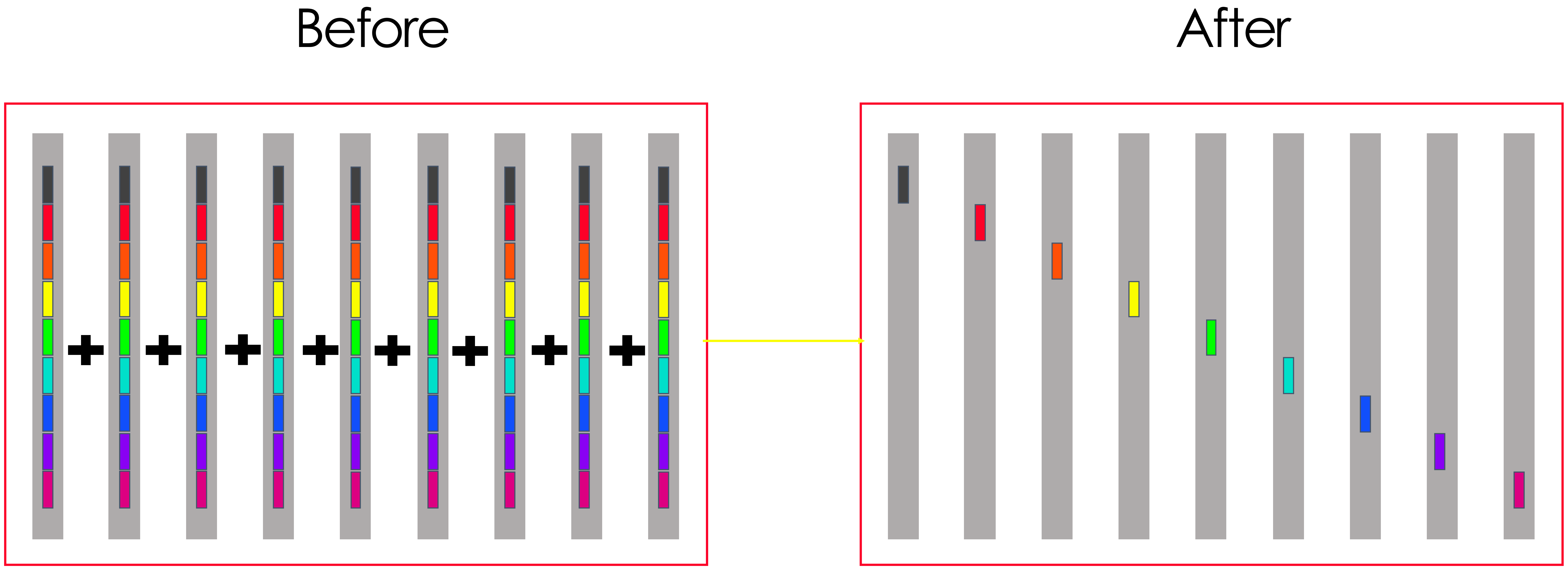
Cost of bucket Allgather

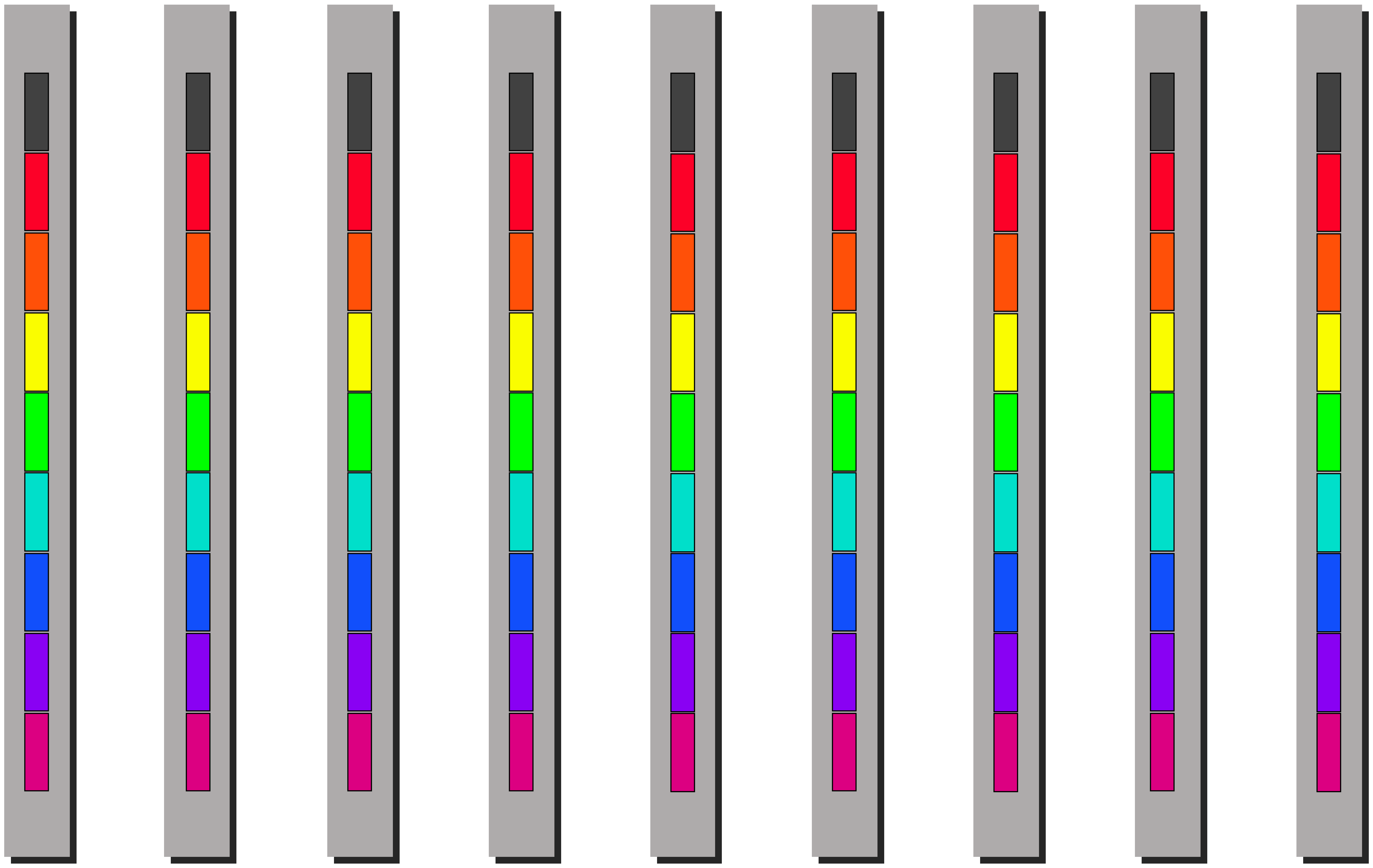
$$\underbrace{(p-1)}_{\text{number of steps}}\underbrace{\left(\alpha+\frac{n}{p}\beta\right)}_{\text{cost per steps}}$$

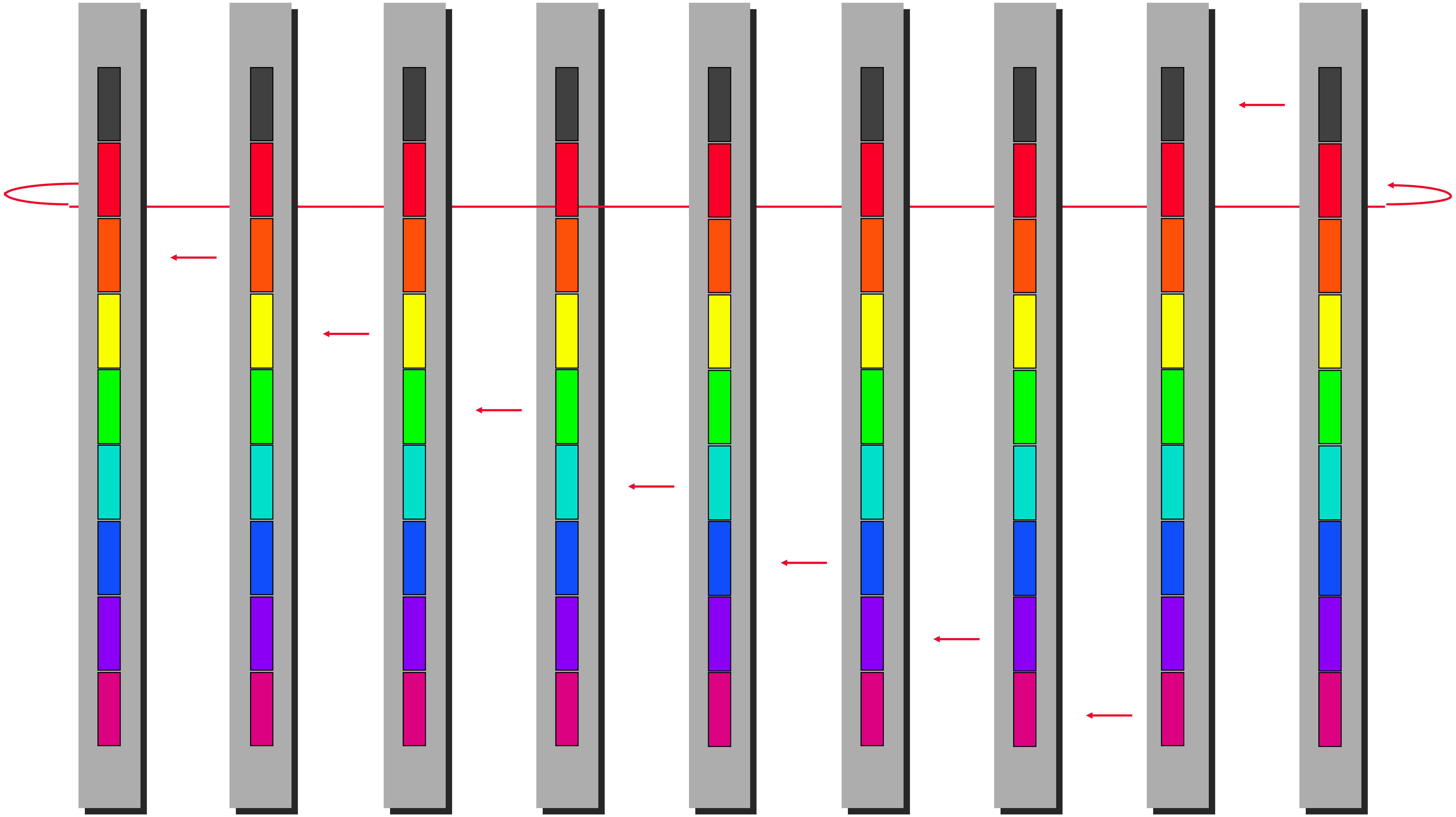
$$=$$

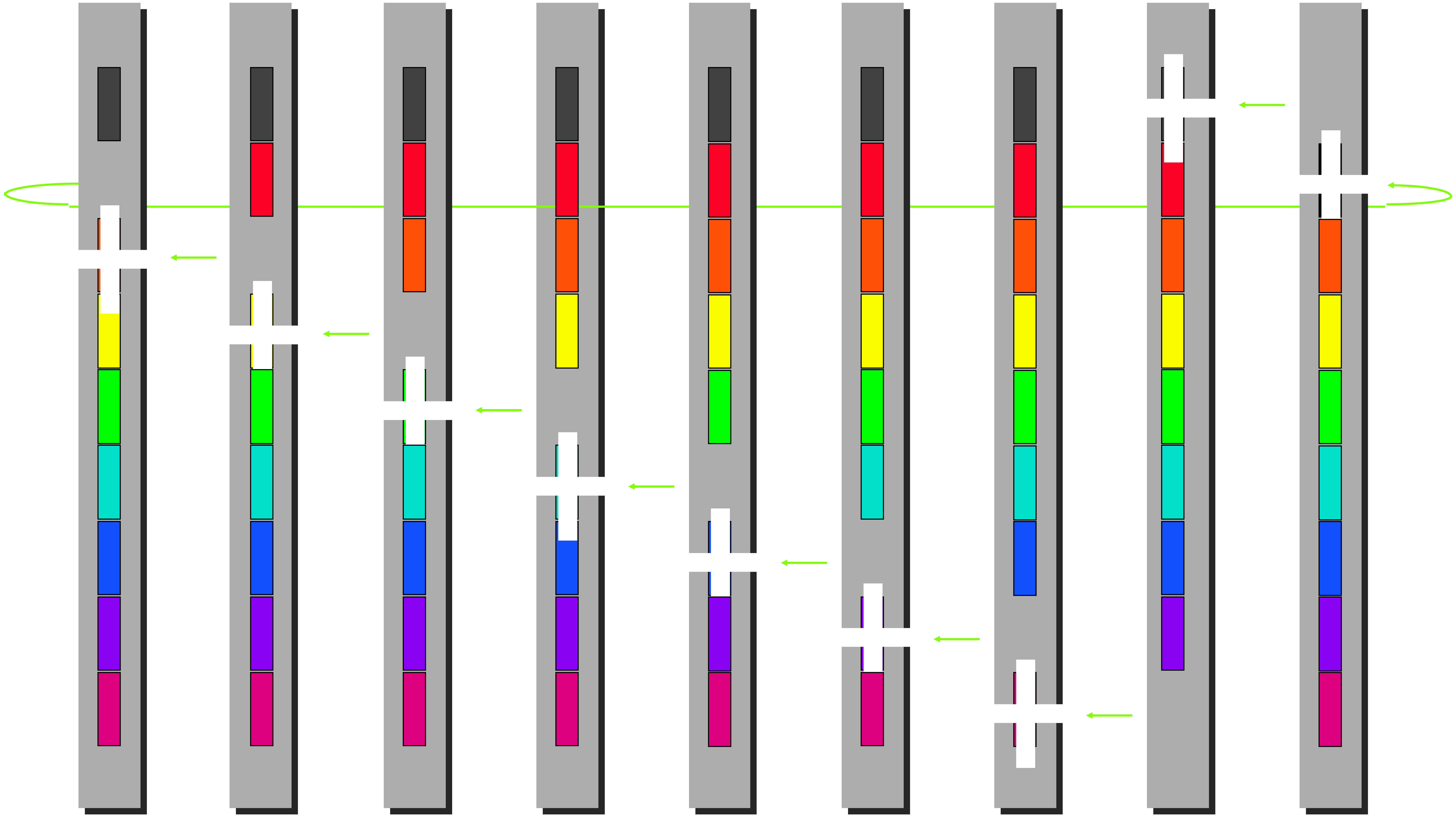
$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

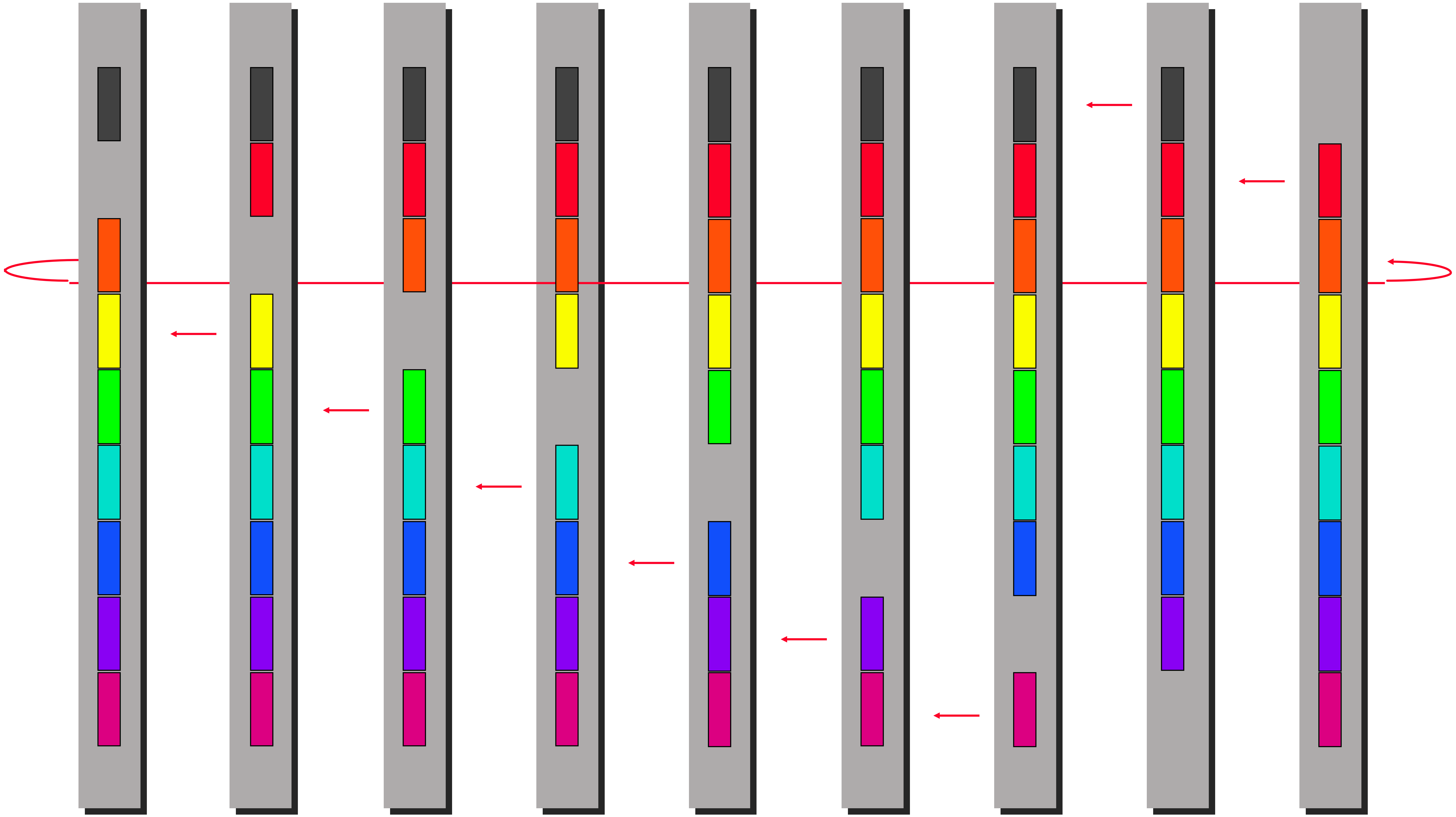
Reduce-scatter

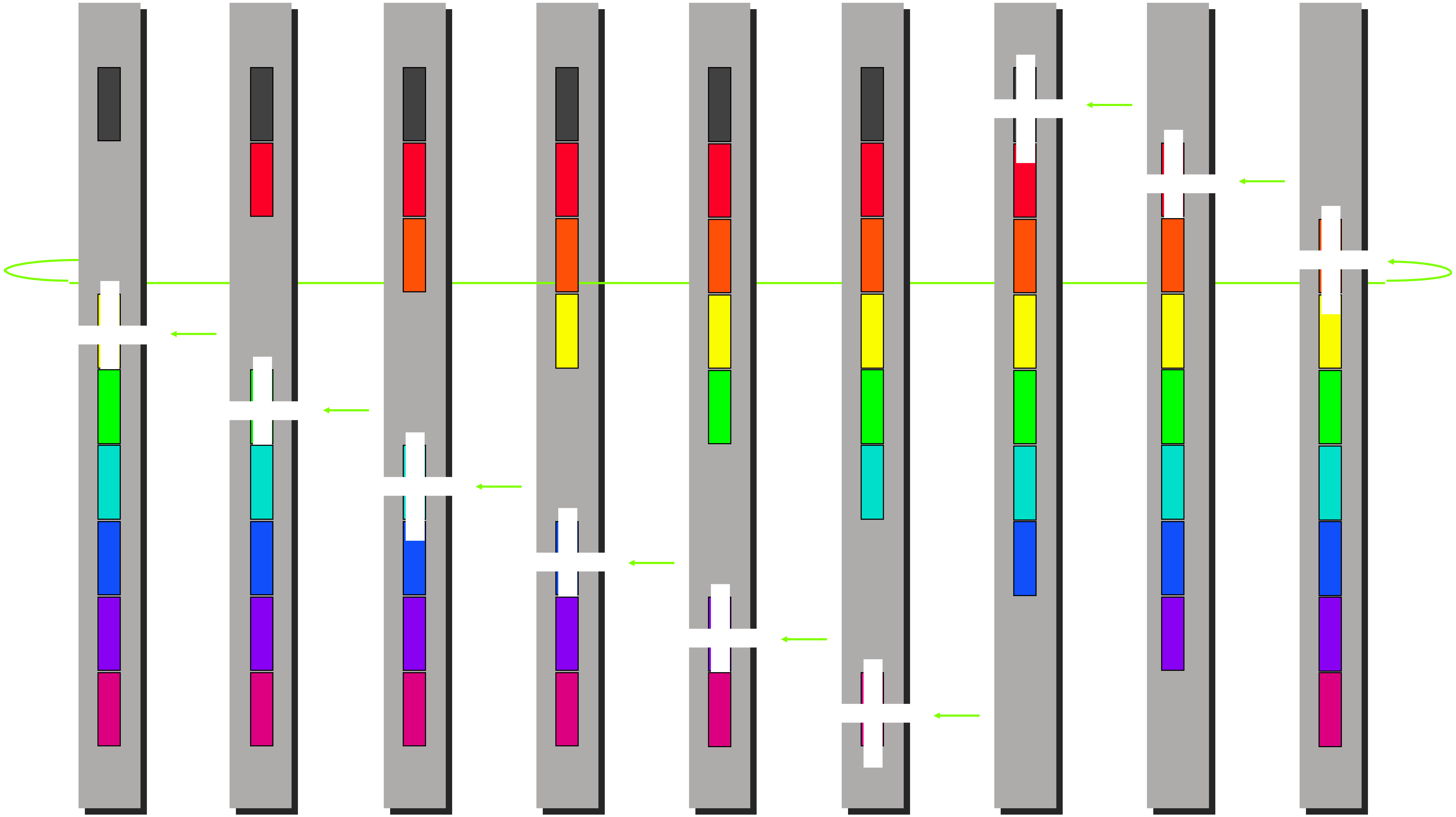


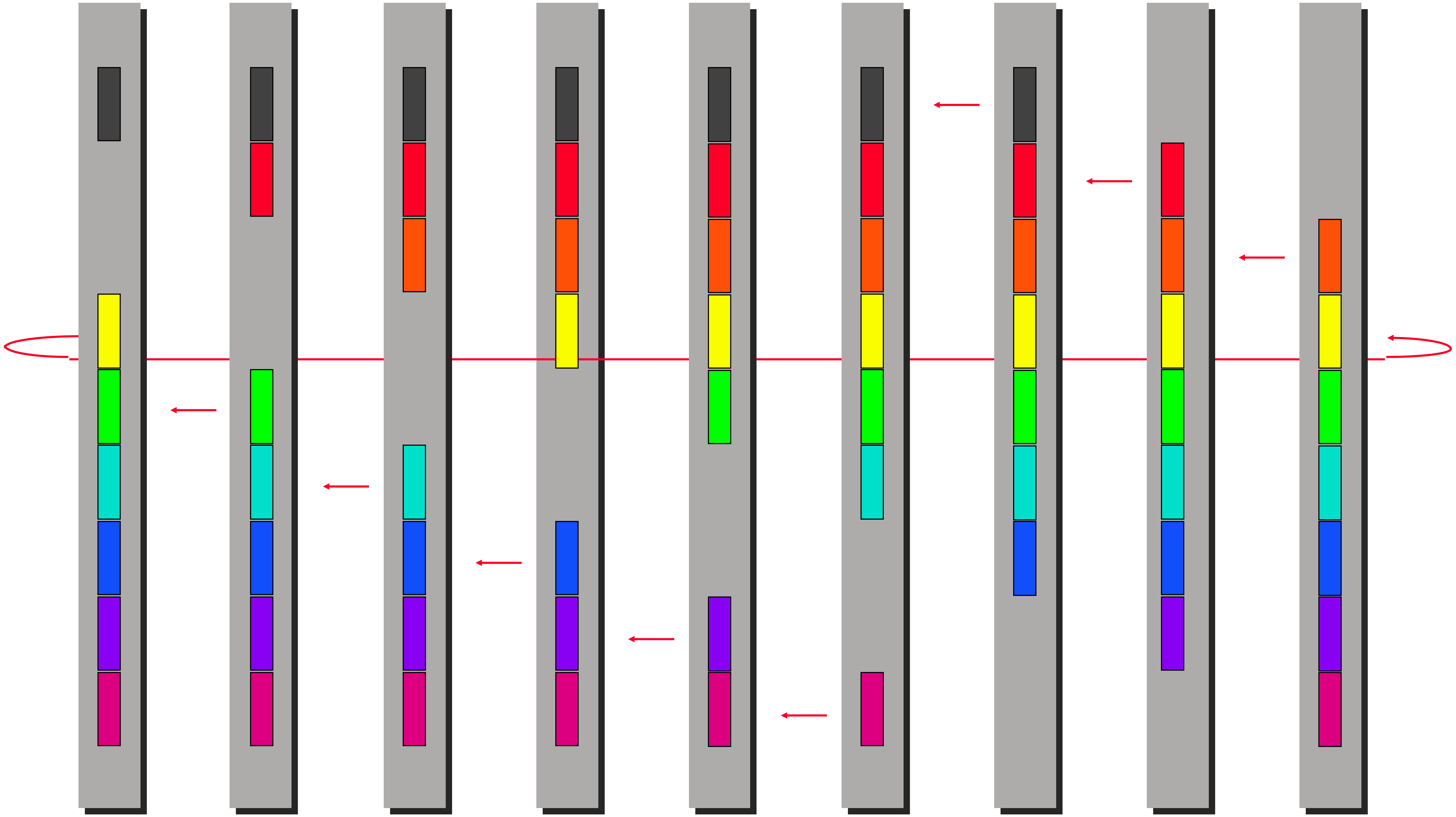


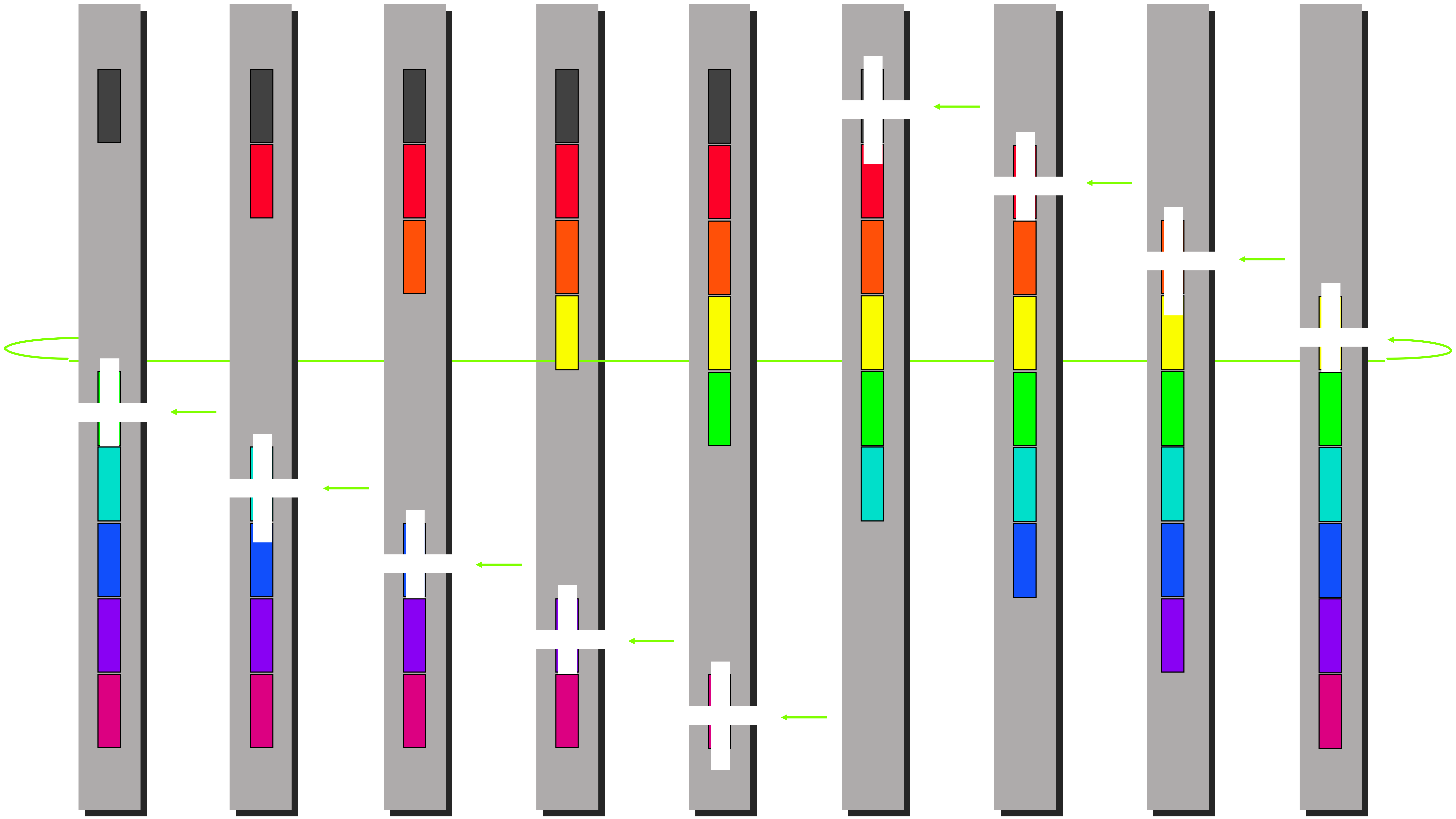


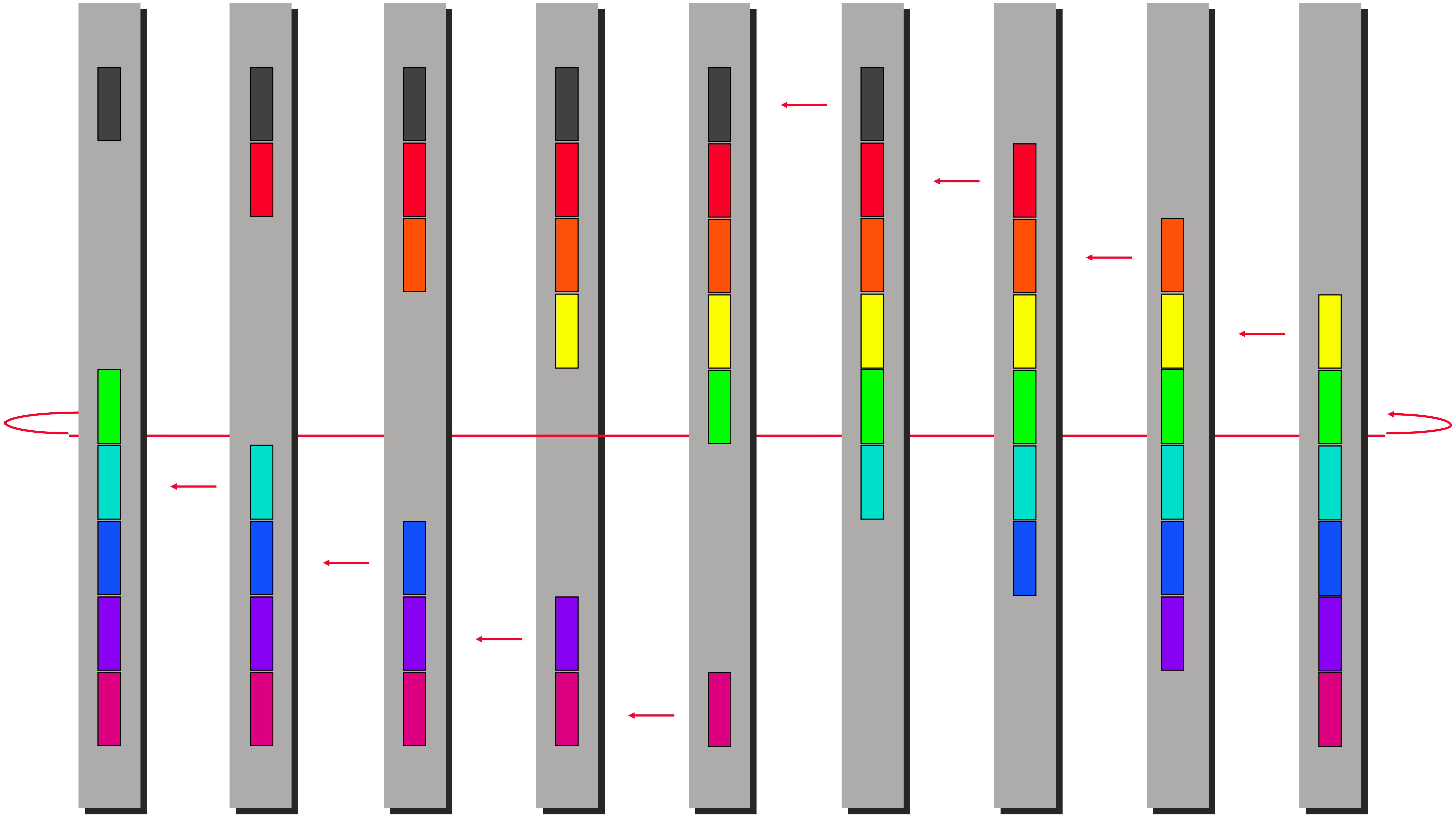


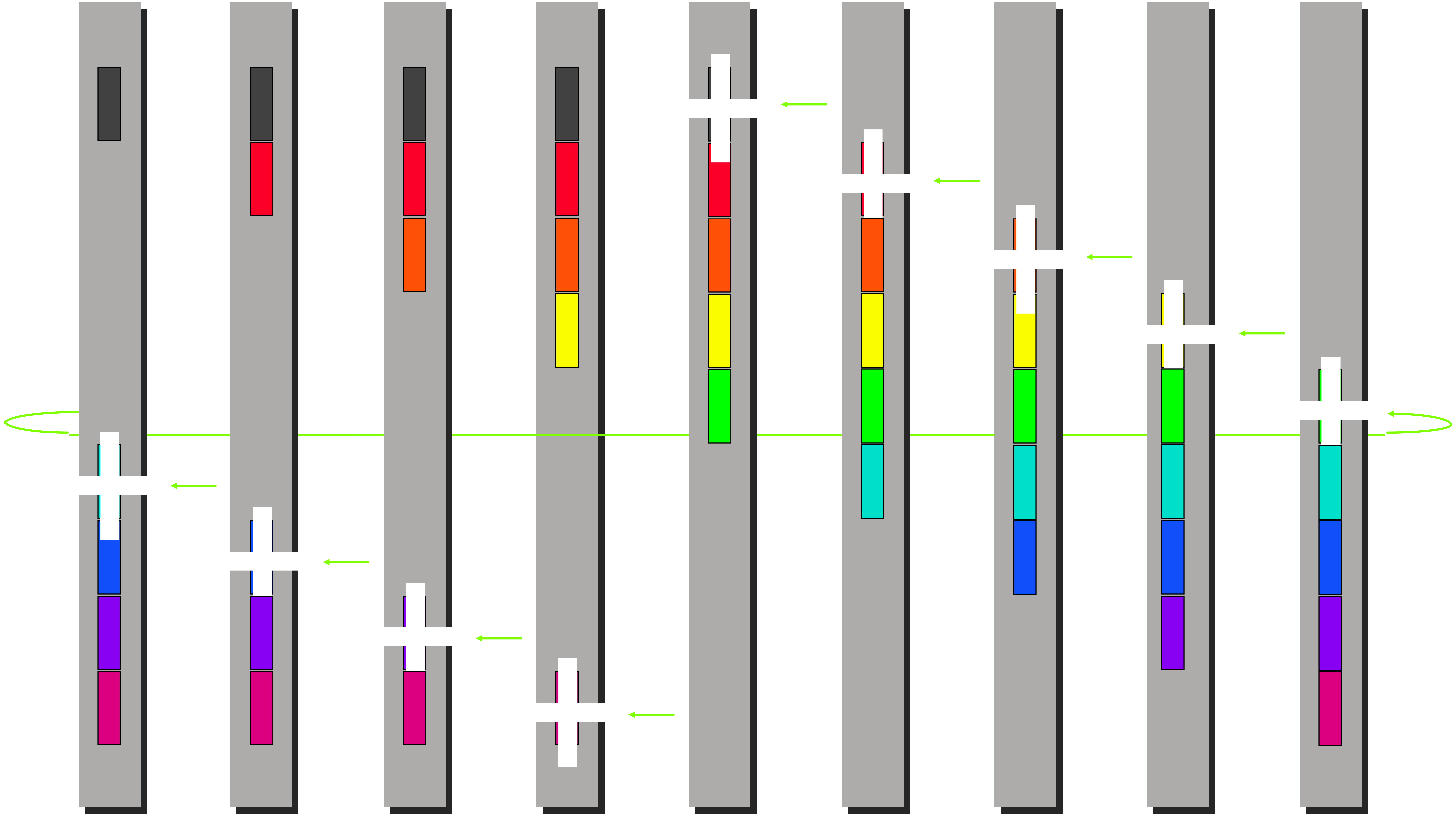


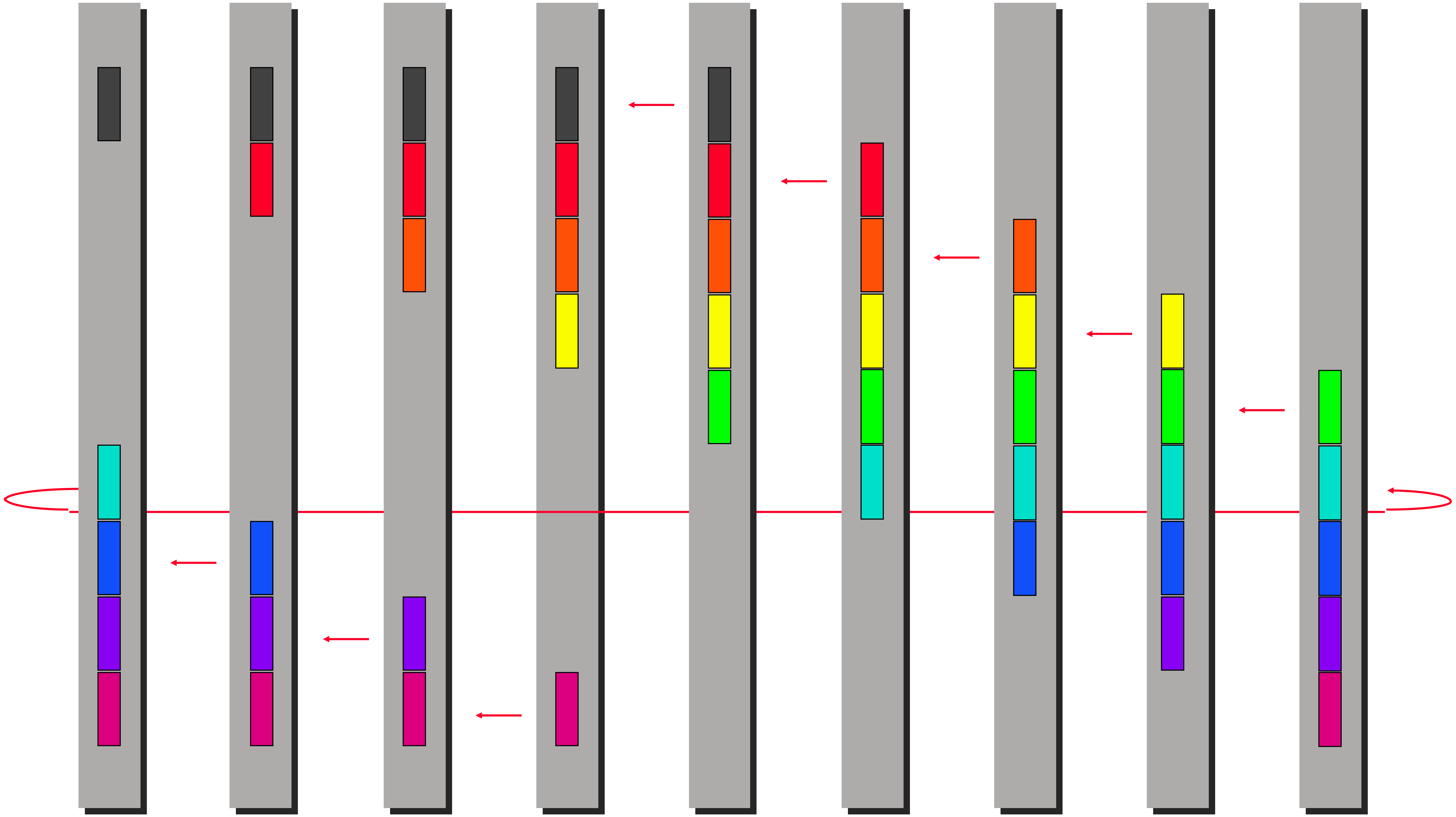


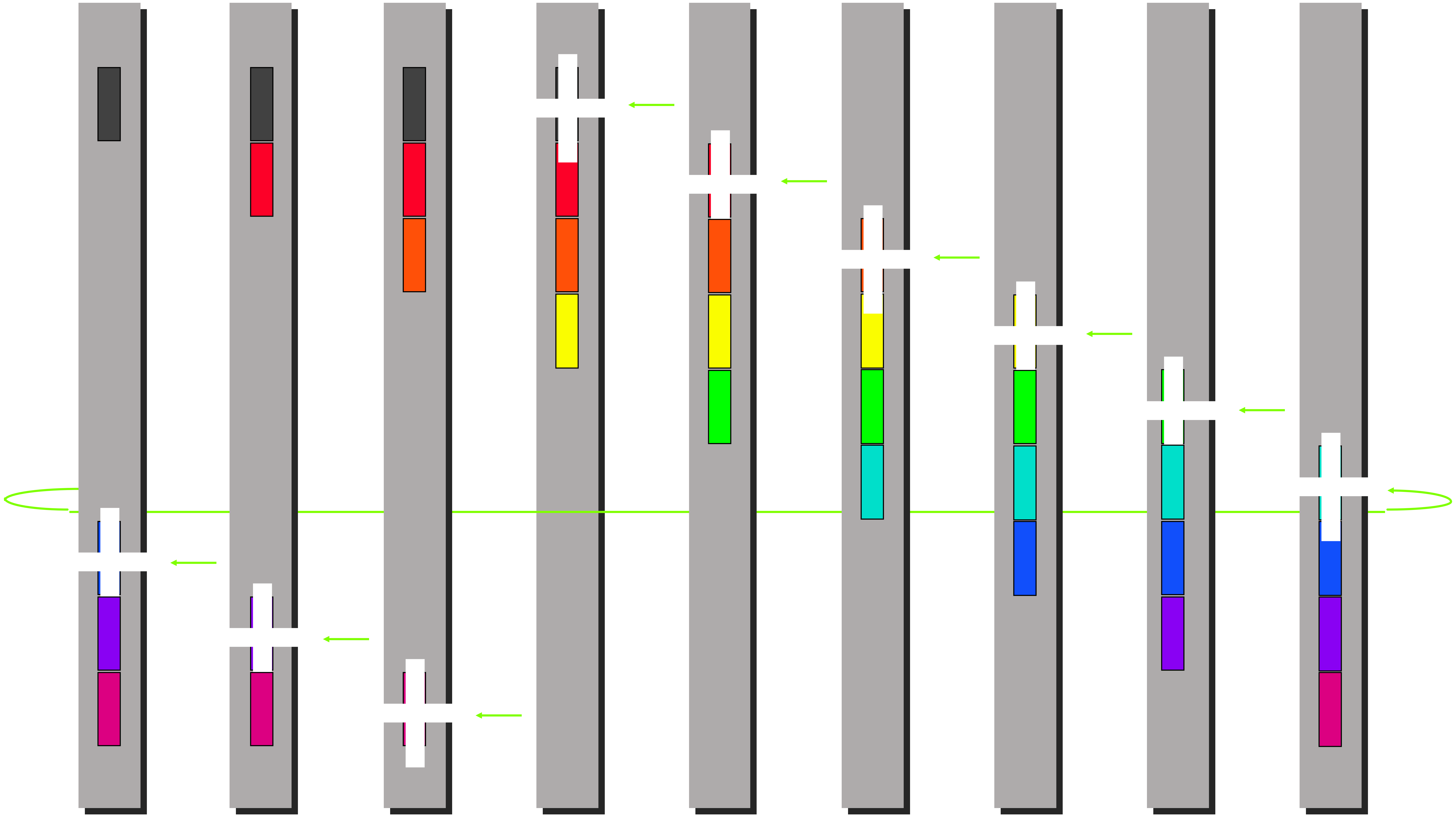


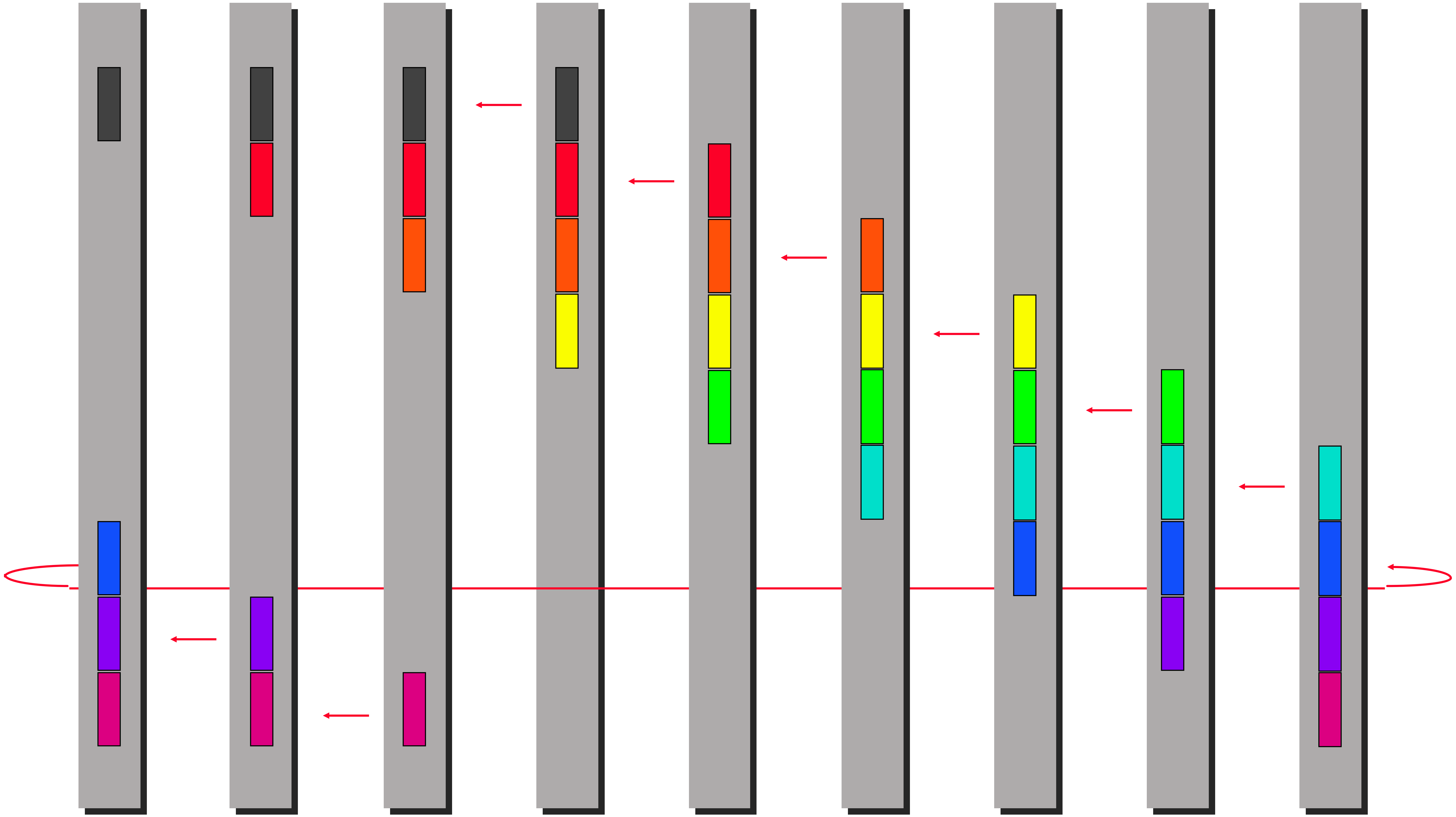


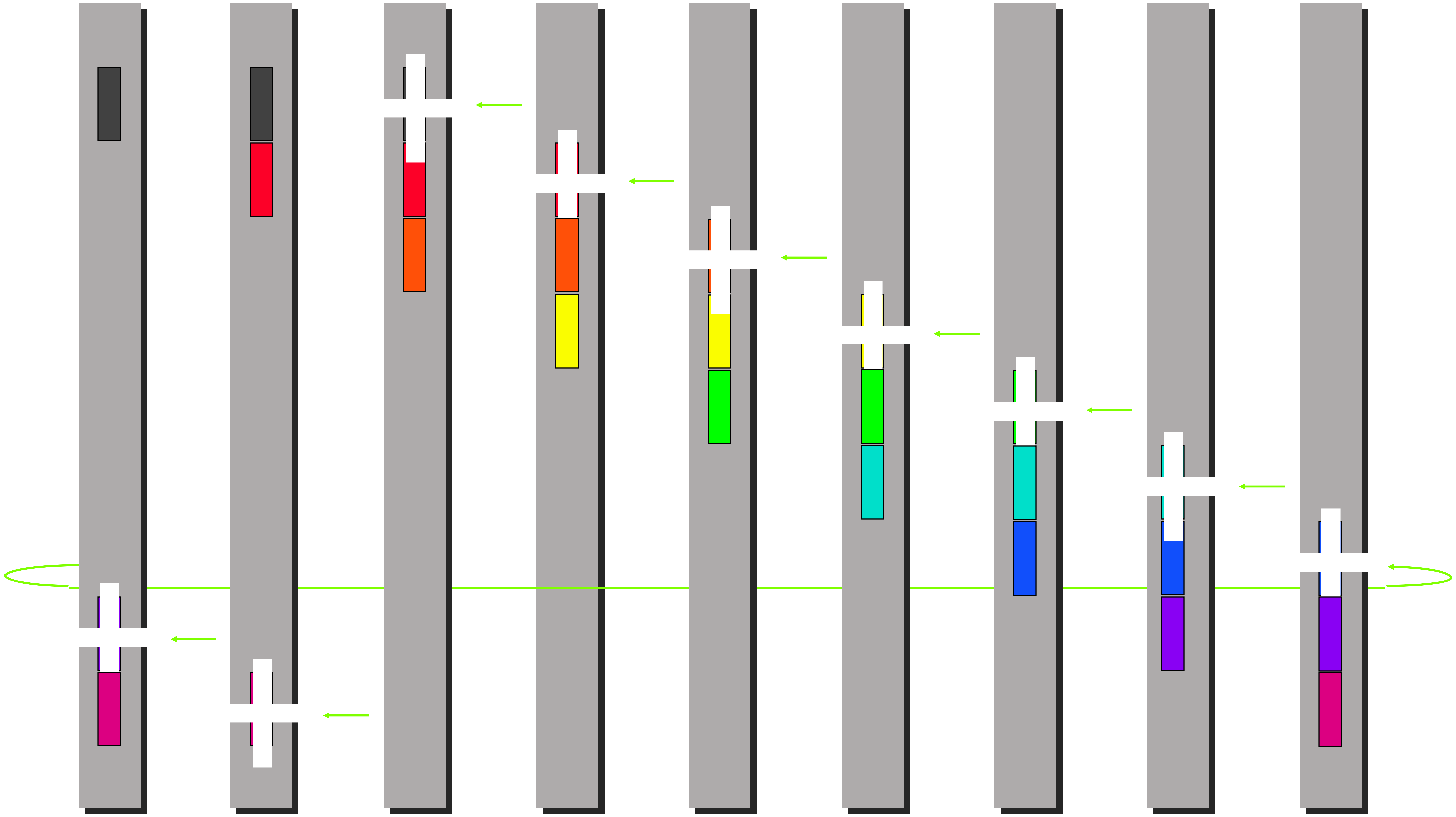


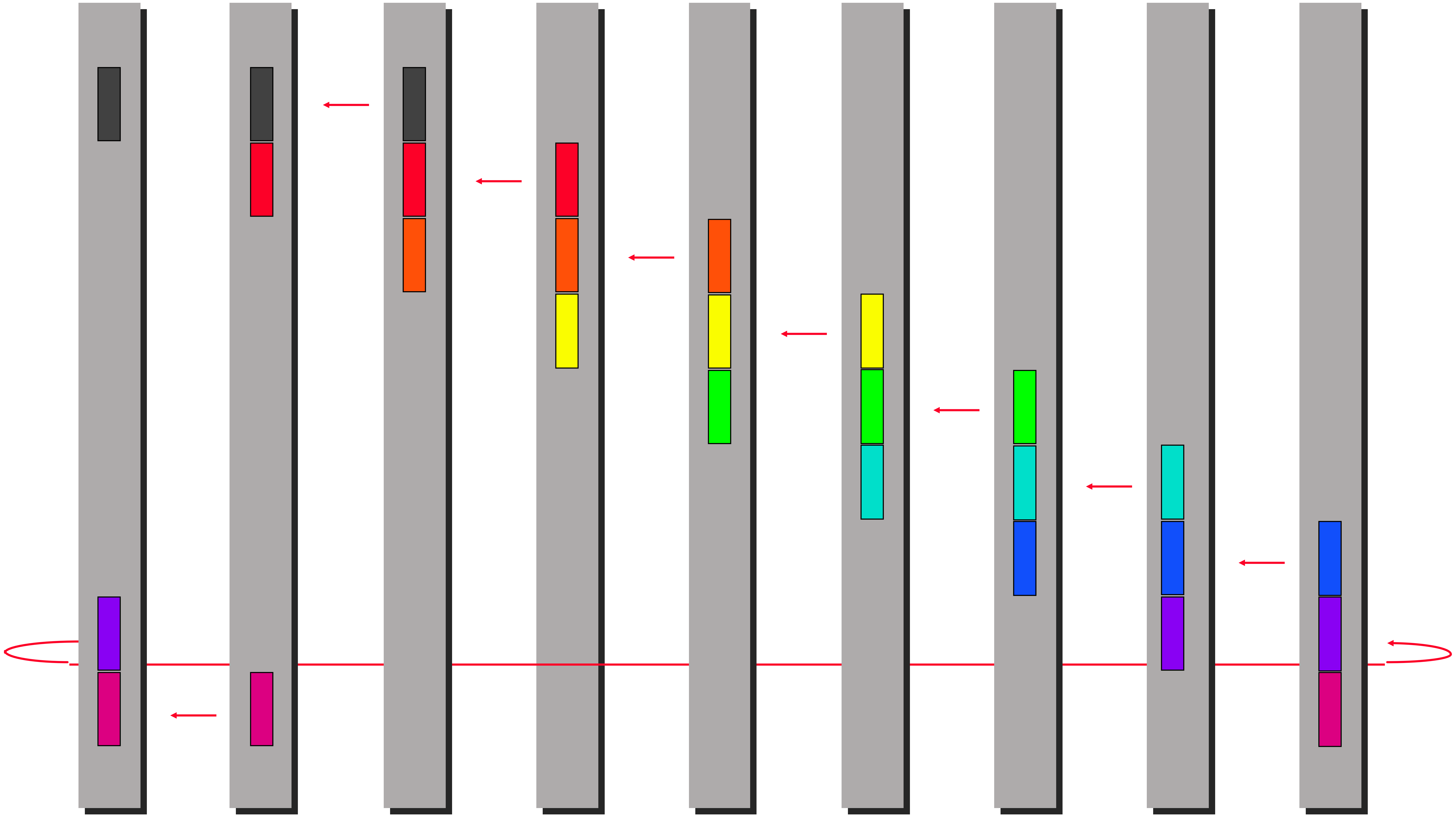


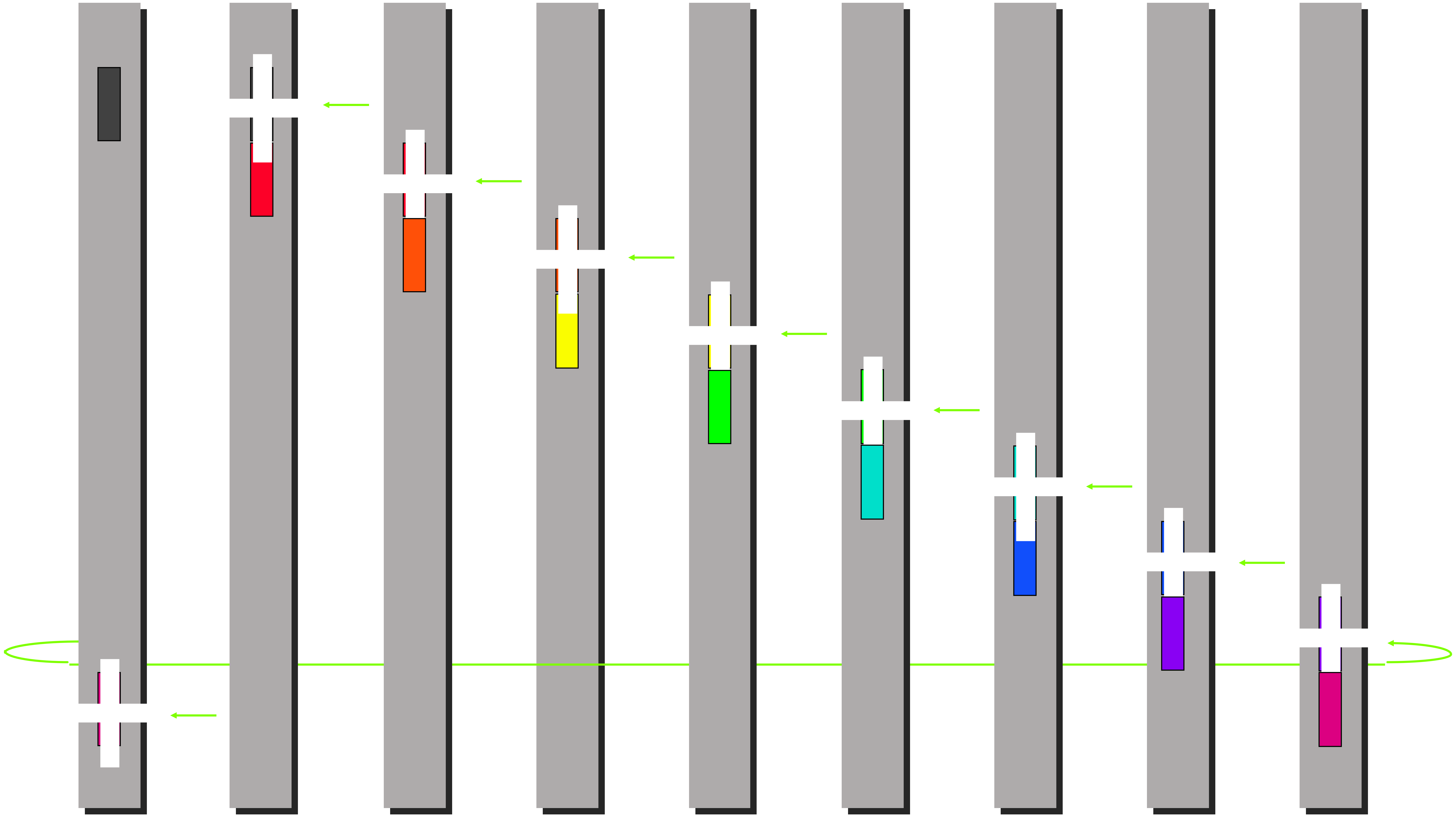


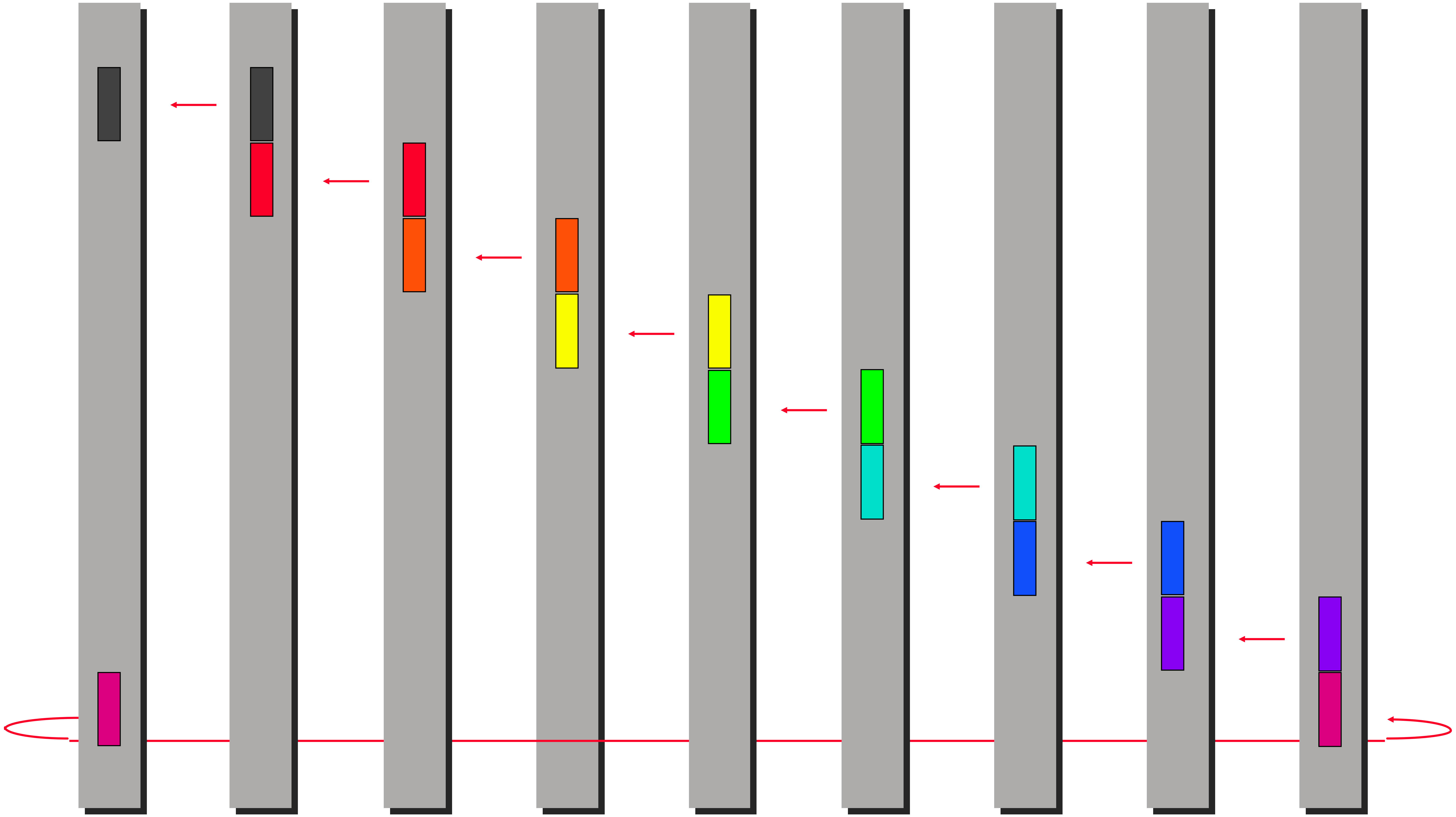


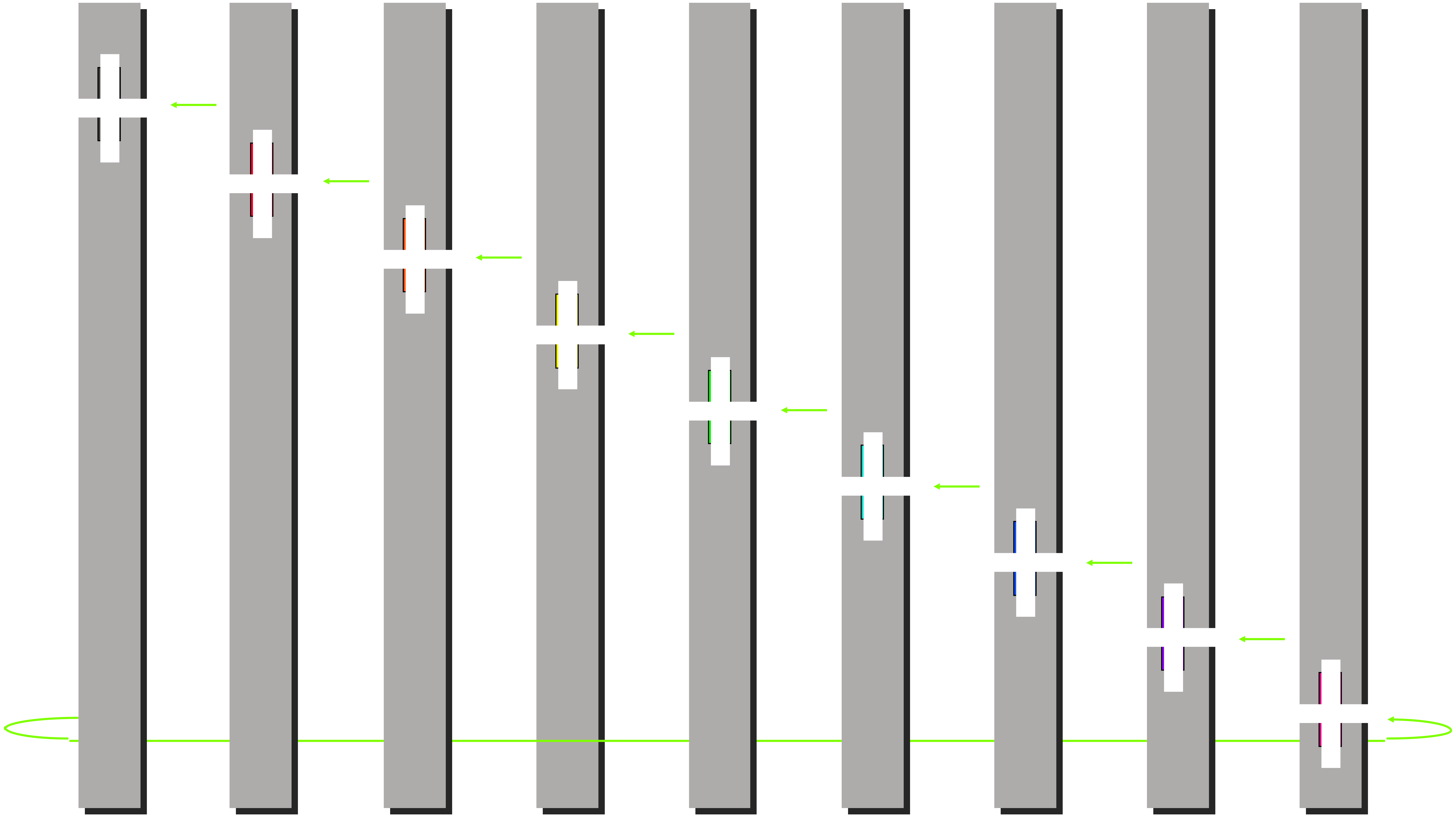


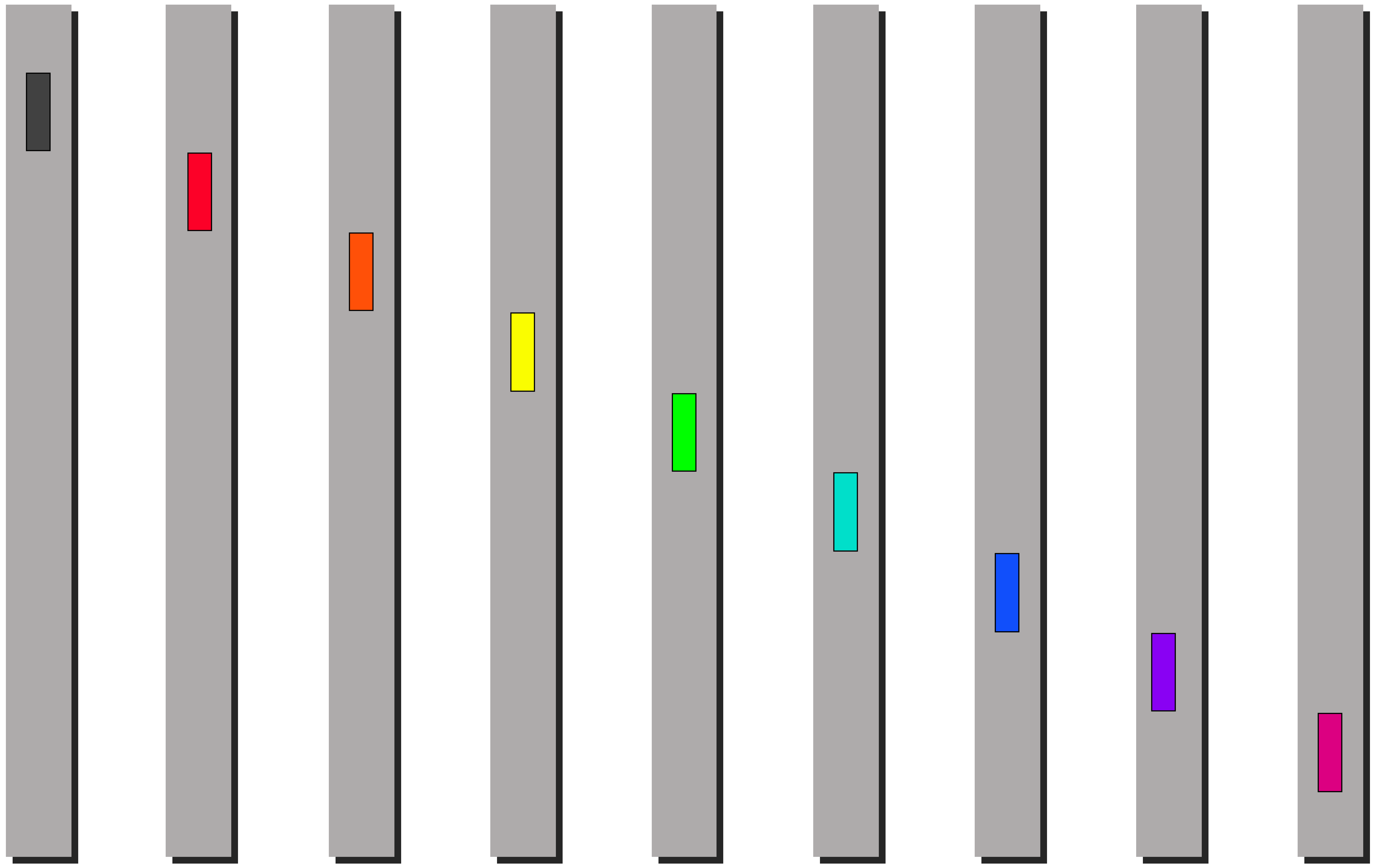












Cost

$$\boxed{(p-1)}\boxed{\left(\alpha+\frac{n}{p}\beta+\frac{n}{p}\gamma\right)}$$

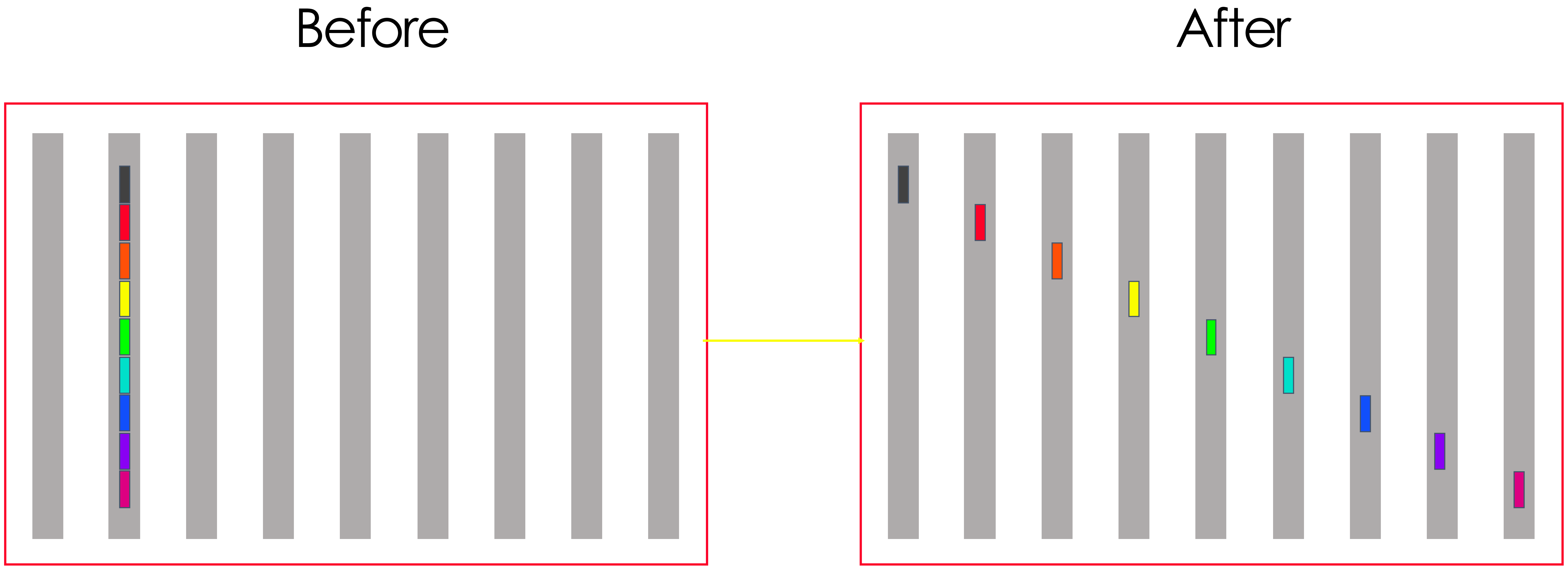
number of steps

cost per steps

$$= (p-1)\alpha+\frac{p-1}{p}n\beta+\frac{p-1}{p}n\gamma$$

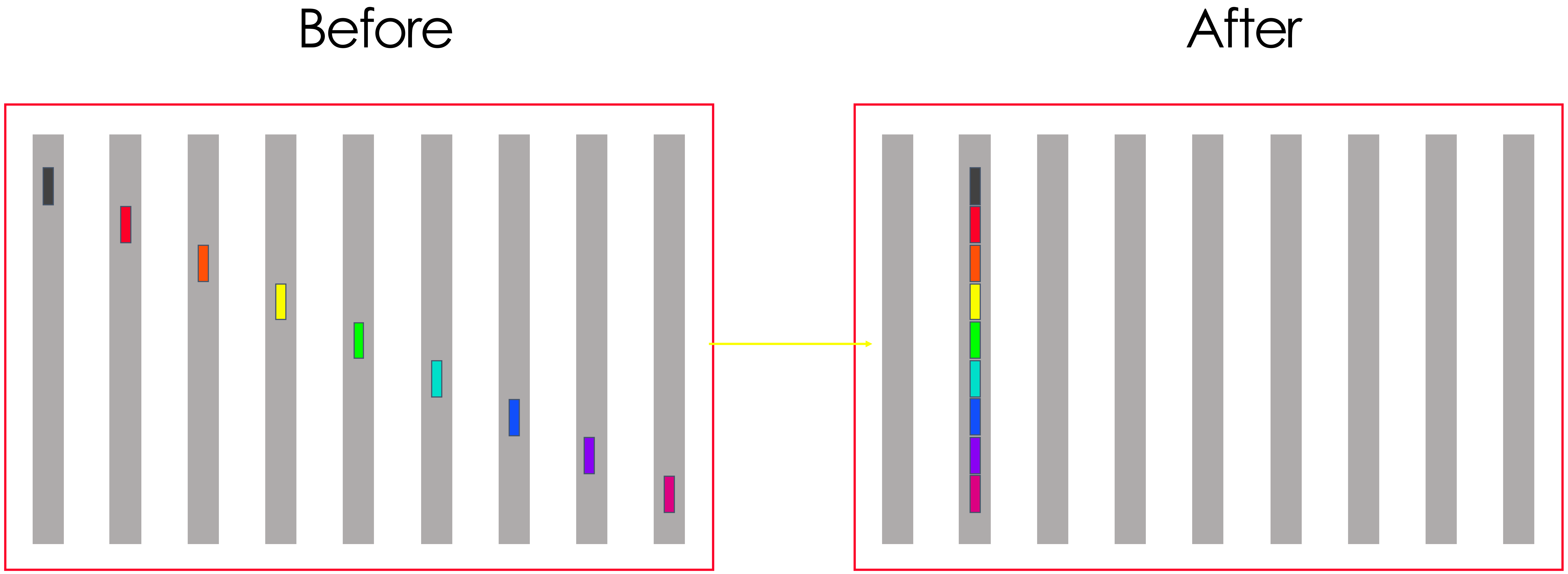
Scatter: Can Ring Be Better?

Notice: Scatter as implemented before using MST was optimal in Bandwidth as well (How to Prove?)



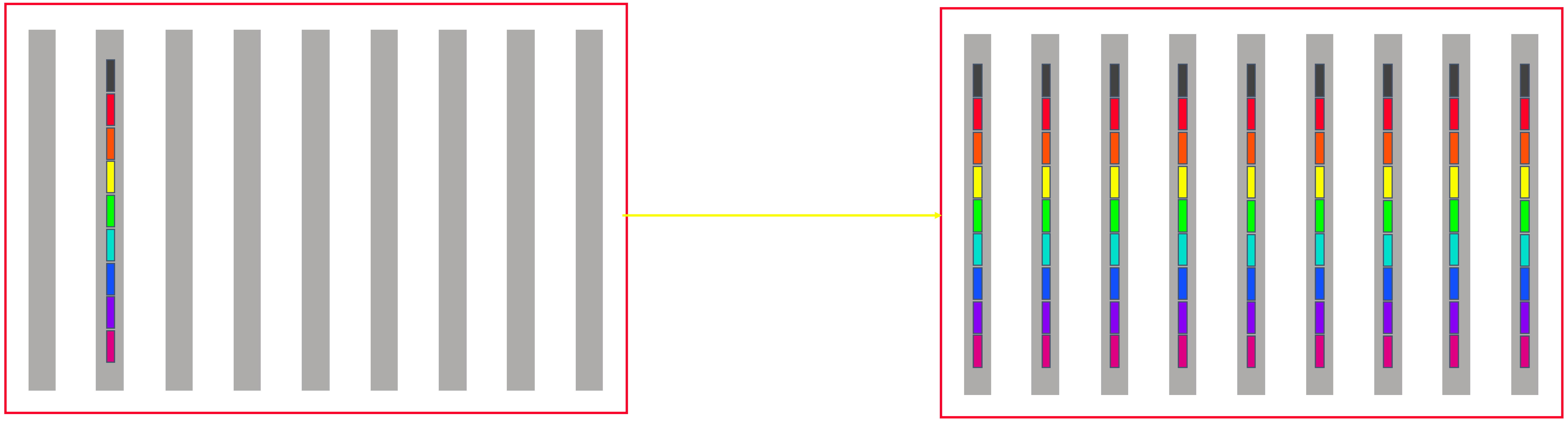
Gather

Notice: Gather as implemented before using MST was optimal in bandwidth as well (how to prove?)

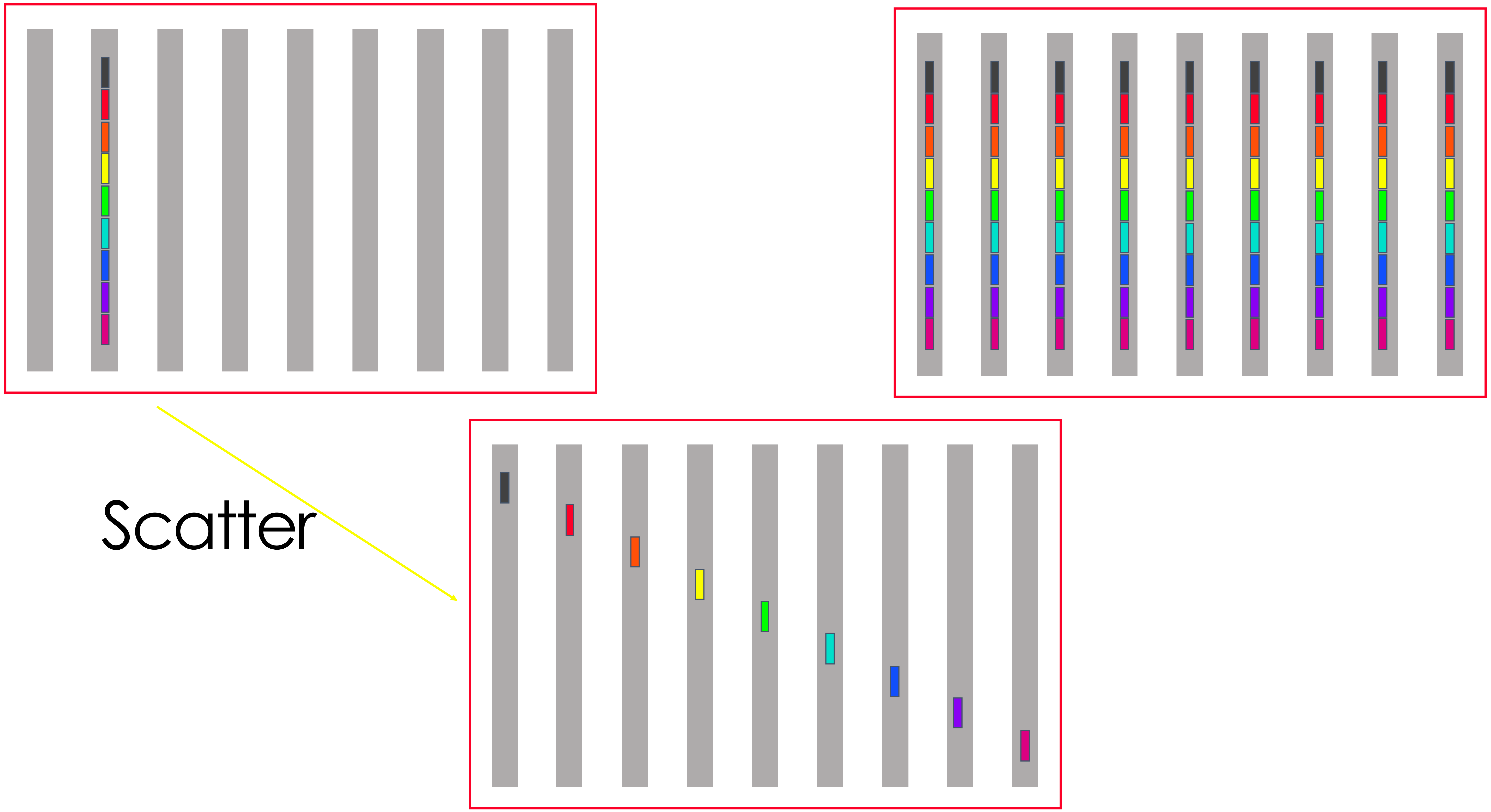


Using the building blocks

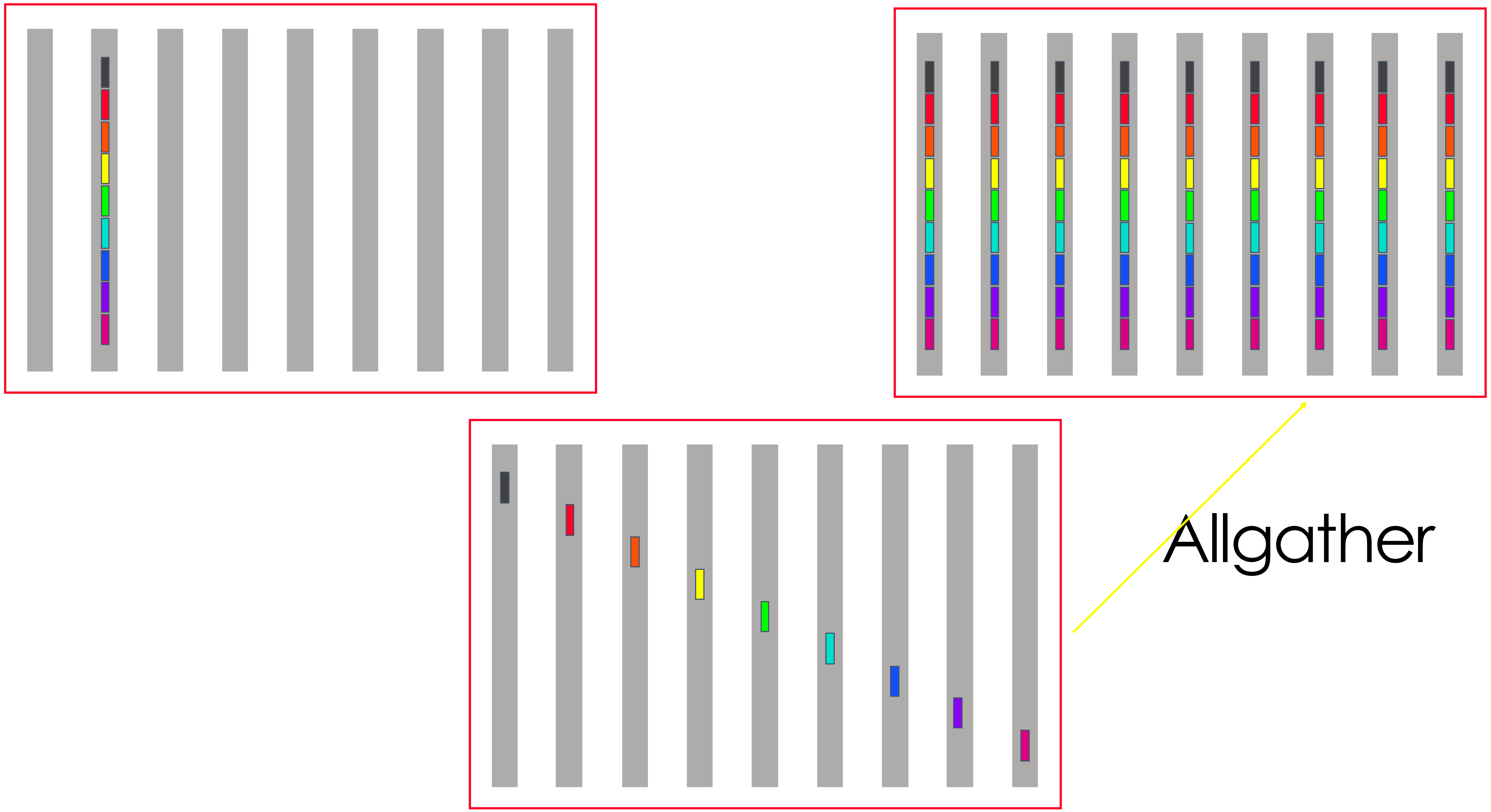
Broadcast (Large Message)



Broadcast (long vector)



Broadcast (long vector)



Cost of scatter/allgather broadcast

- Assumption: power of two number of nodes

scatter $\qquad \log(p)\alpha + \frac{p-1}{p} n\beta$

allgather $\qquad (p-1)\alpha + \frac{p-1}{p} n\beta$

$$(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta$$

Cost of scatter/allgather broadcast

- Assumption: power of two number of nodes

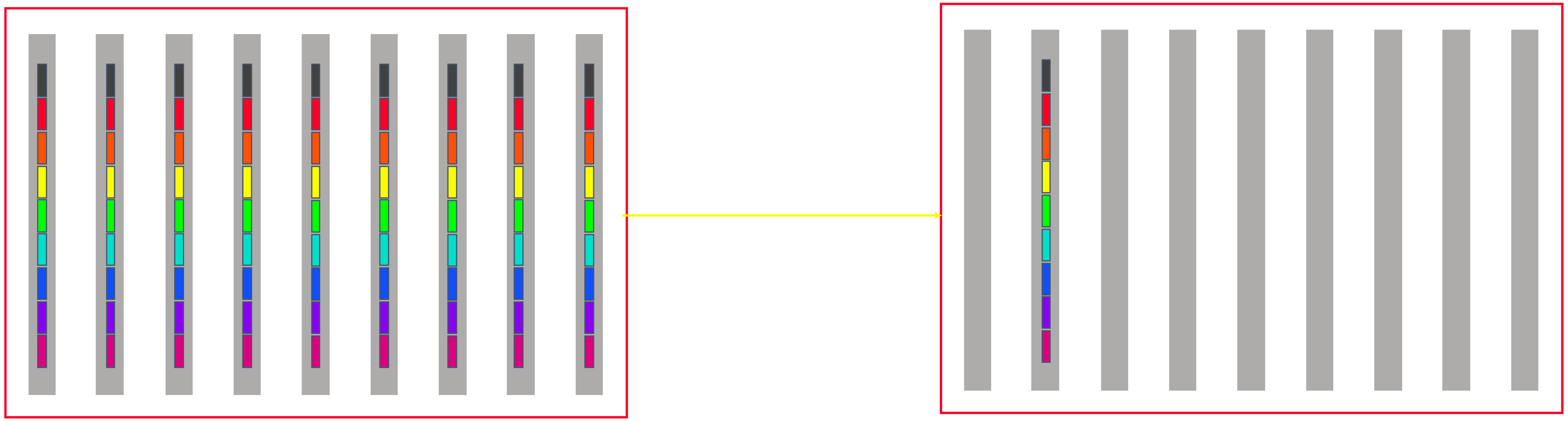
scatter
$$\log(p)\alpha + \frac{p-1}{p}n\beta$$

allgather
$$(p-1)\alpha + \frac{p-1}{p}n\beta$$

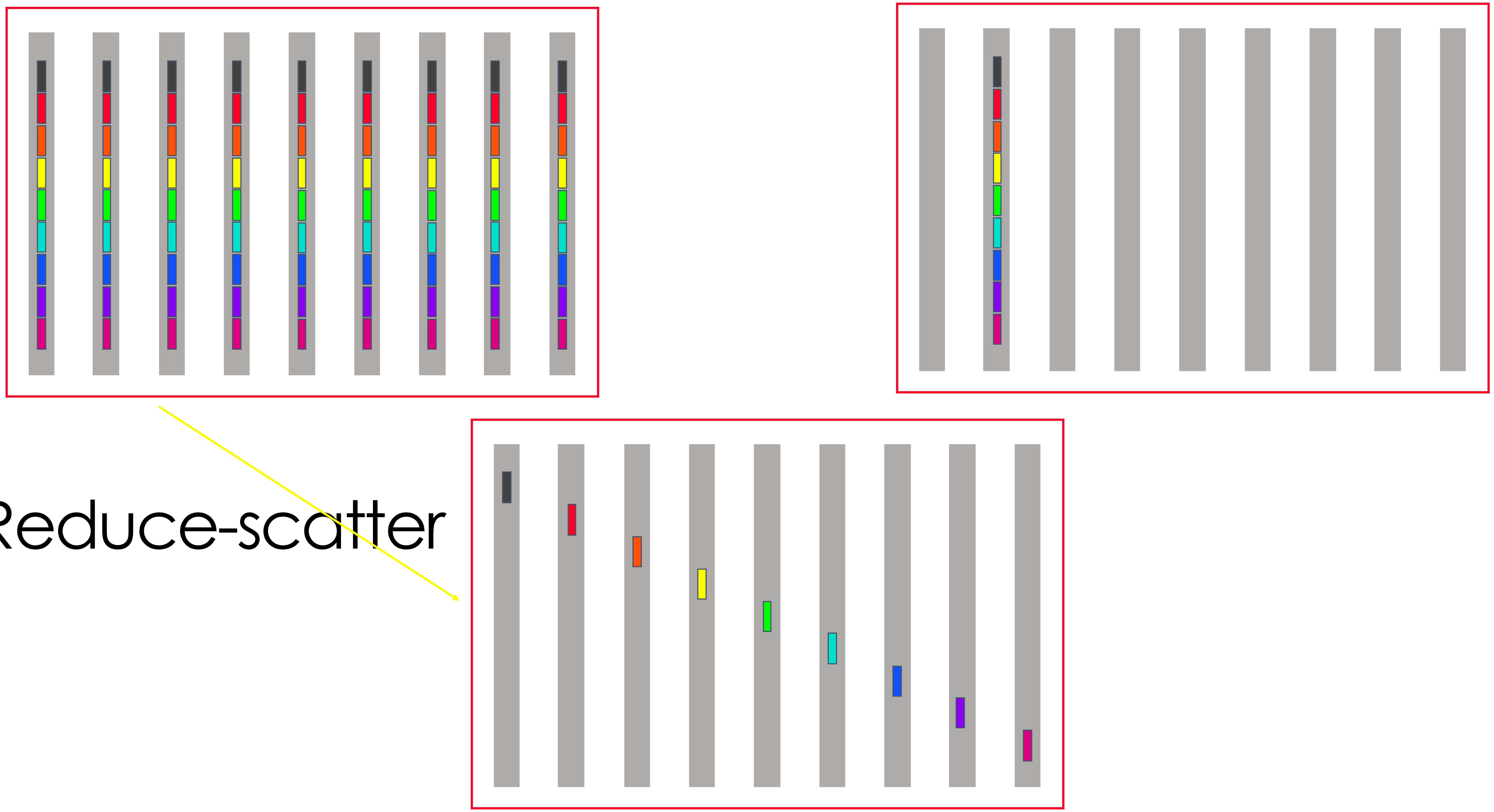
$$(\log(p) + p - 1)\alpha + 2\frac{p-1}{p}n\beta$$

Vs. MST broadcast: $\lceil \log(p) \rceil (\alpha + n\beta)$

Reduce(-to-one) (long vector)

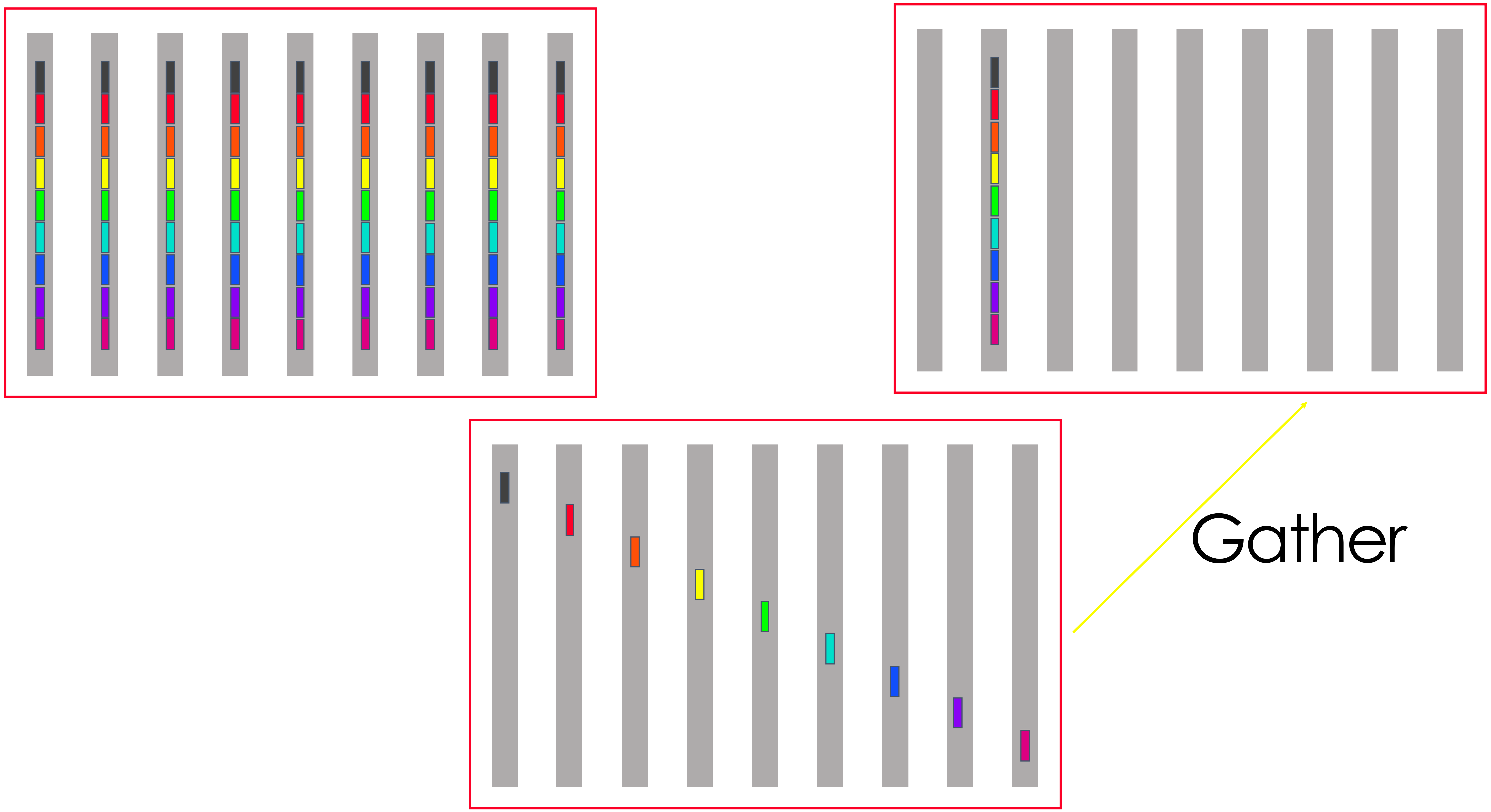


Reduce (long vector)



Reduce-scatter

Combine-to-one (long vector)



Cost of Reduce-scatter/Gather Reduce(-to-one)

- Assumption: power of two number of nodes

Reduce-scatter $$(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

gather $$\log(p)\alpha + \frac{p-1}{p} n\beta$$

$$(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

Cost of Reduce-scatter/Gather Reduce(-to-one)

- Assumption: power of two number of nodes

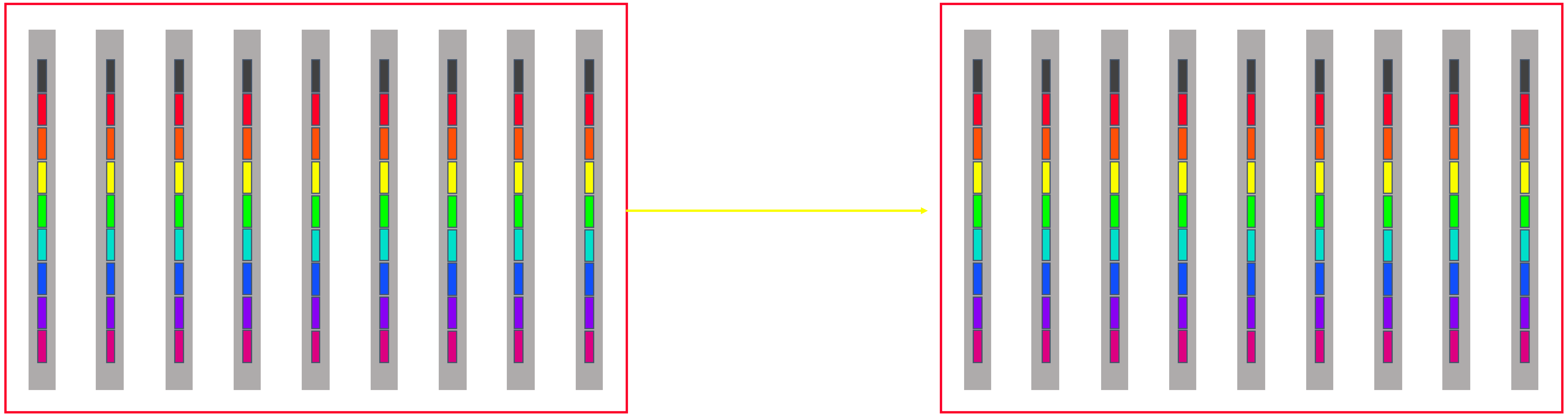
Reduce-scatter $$(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

gather $$\log(p)\alpha + \frac{p-1}{p} n\beta$$

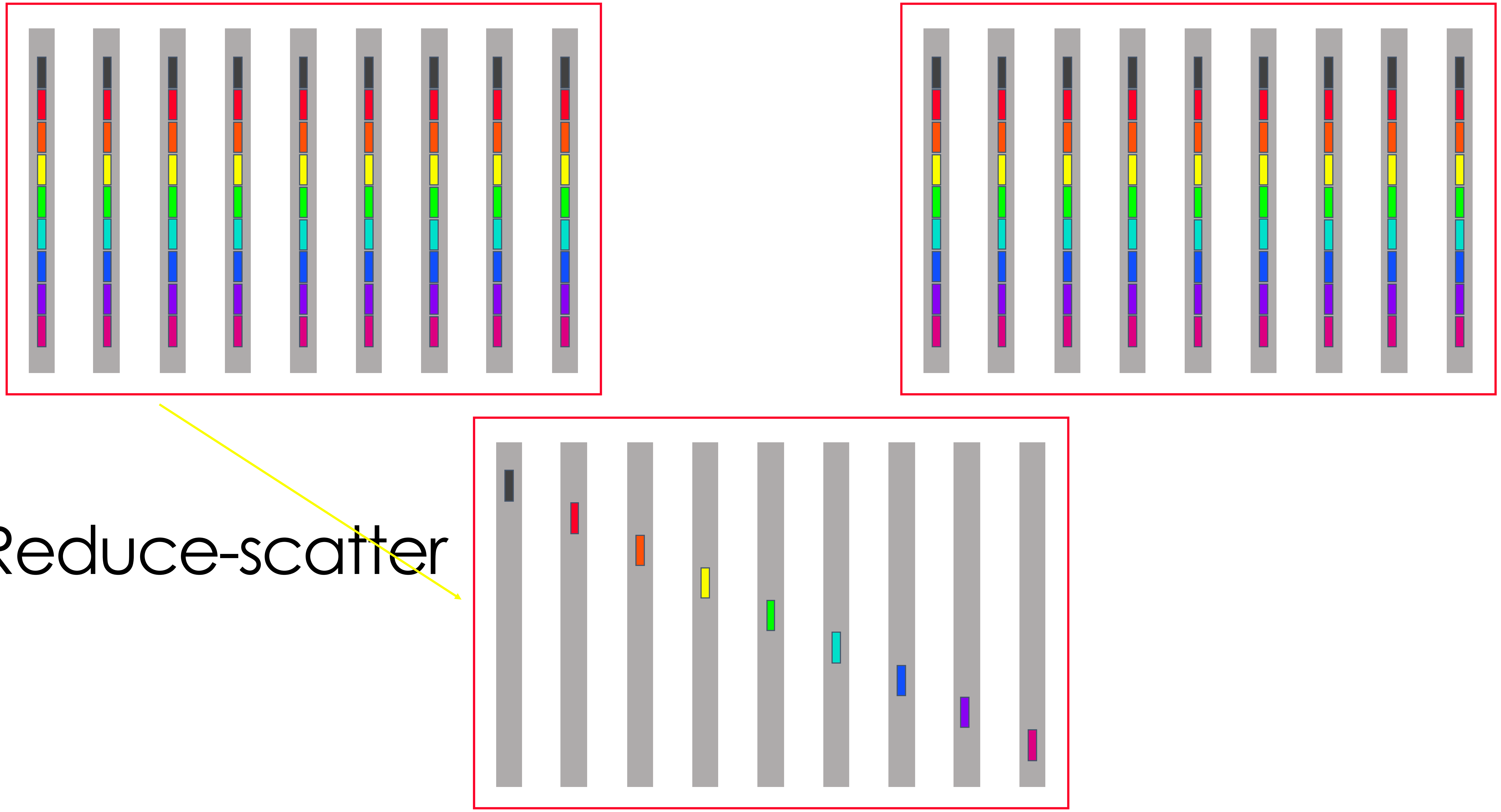
$$(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

Vs. MST reduce: $$\lceil \log(p) \rceil \; (\alpha + n\beta + n\gamma)$$

Allreduce (Large Message)

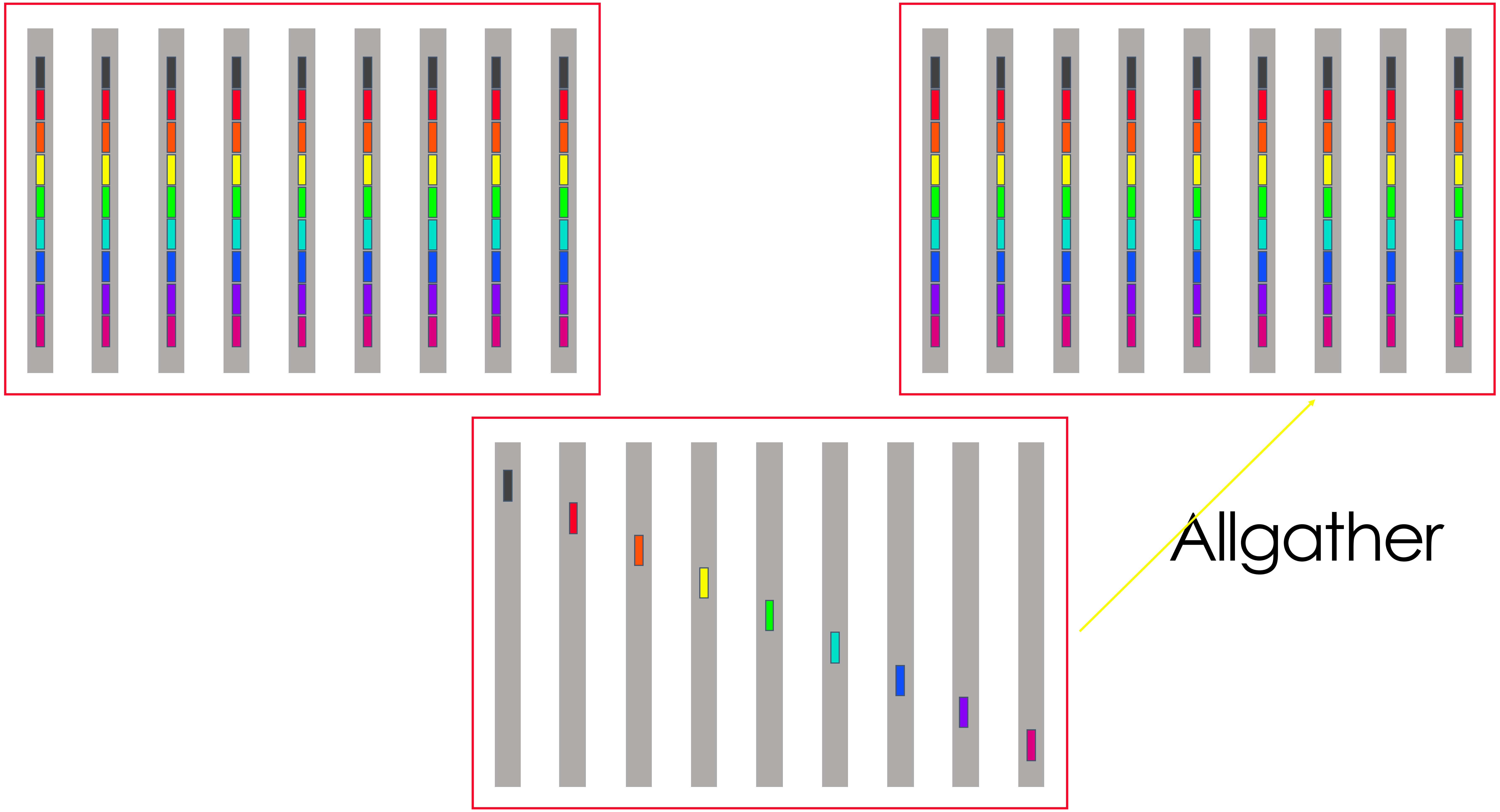


Allreduce (Large Message)



Reduce-scatter

Allreduce
(long vector)



Cost of Reduce-scatter/Allgather Allreduce

- Assumption: power of two number of nodes

Reduce-scatter $$(p-1)\alpha + \frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma$$

Allgather $$(p-1)\alpha + \frac{p-1}{p}n\beta$$

$$2(p-1)\alpha + 2\frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma$$

Cost of Reduce-scatter/Allgather Allreduce

- Assumption: power of two number of nodes

Reduce-scatter $$(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

Allgather $$(p-1)\alpha + \frac{p-1}{p} n\beta$$

$$2(p-1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$$

Vs. Reduce-broadcast allreduce

$$2log(p)\alpha + 2log(p)n\beta + log(p)n\gamma$$

Recap

Reduce-scatter

$$(p-1)\alpha + \frac{p-1}{p} n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Allgather

$$(p-1)\alpha + \frac{p-1}{p} n\beta$$

Reduce(-to-one)

Allreduce

Broadcast

Recap

Reduce-scatter

$$(p-1)\alpha + \frac{p-1}{p} n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Allgather

$$(p-1)\alpha + \frac{p-1}{p} n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha + \frac{p-1}{p} n(2\beta+\gamma)$$

Allreduce

Broadcast

Recap

Reduce-scatter

$$(p-1)\alpha + \frac{p-1}{p} n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Allgather

$$(p-1)\alpha + \frac{p-1}{p} n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha + \frac{p-1}{p} n(2\beta+\gamma)$$

Allreduce

$$2(p-1)\alpha + \frac{p-1}{p} n(2\beta+\gamma)$$

Broadcast

$$(\log(p)+p-1)\alpha + 2\frac{p-1}{p} n\beta$$

Recap

Reduce-scatter

$$(p-1)\alpha + \frac{p-1}{p} n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Allgather

$$(p-1)\alpha + \frac{p-1}{p} n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha + \frac{p-1}{p} n(2\beta+\gamma)$$

Allreduce

$$2(p-1)\alpha + \frac{p-1}{p} n(2\beta+\gamma)$$

Broadcast

Recap

Reduce-scatter

$$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Gather

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Allgather

$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Allreduce

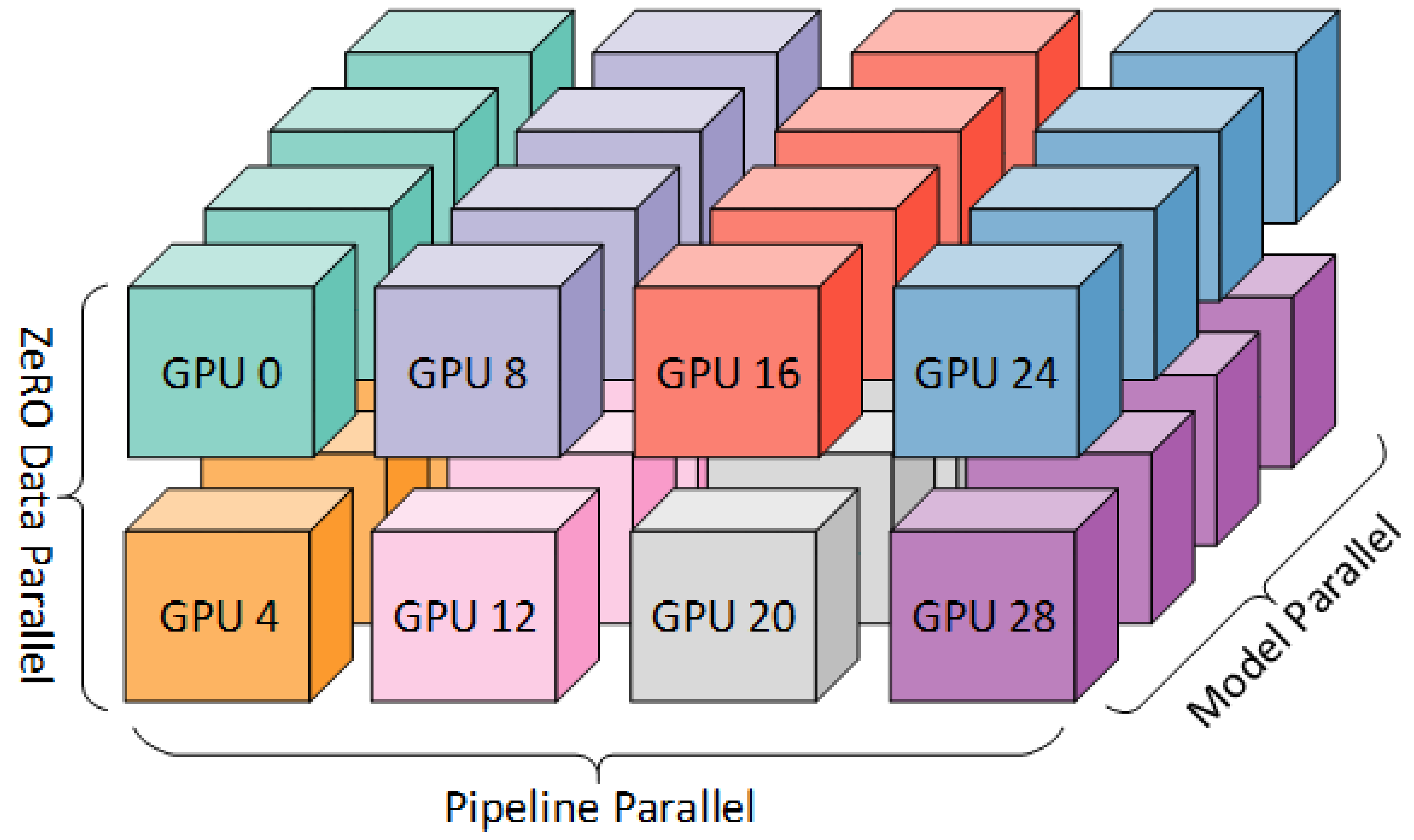
$$2(p-1)\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Broadcast

$$(\log(p)+p-1)\alpha+2\frac{p-1}{p}n\beta$$

A More Complicate Case

- Real Cluster to train ChatGPT:
  - If using GPU: 2D Mesh
  - If using TPU: 3D Mesh, see figure below



Example: 2D Broadcast

- Idea: Use 1D to compose 2

Example: 2D Broadcast

- Idea: Use 1D to compose 2
- Option 1:
  - **MST broadcast in column**

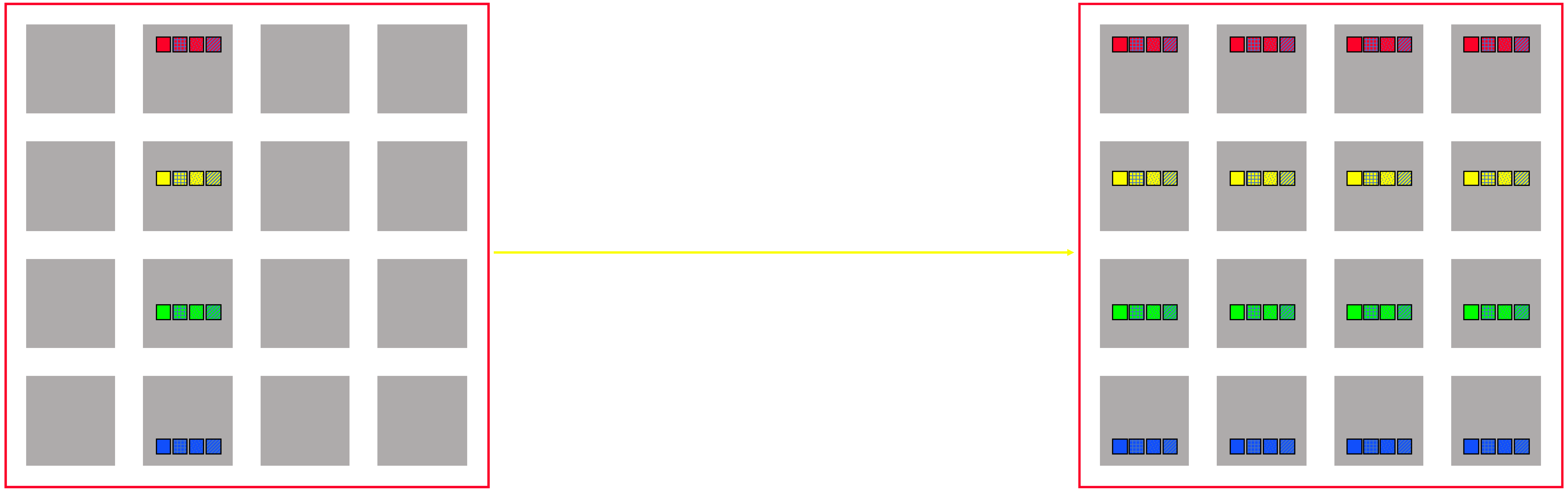
Example: 2D Broadcast

- Option 1:
  - MST broadcast in column
  - **MST broadcast in rows**

Example: 2D Broadcast

- Option 2:
  - Scatter in column

Example: 2D Broadcast



- Option 2:
  - Scatter in column
  - MST broadcast in rows

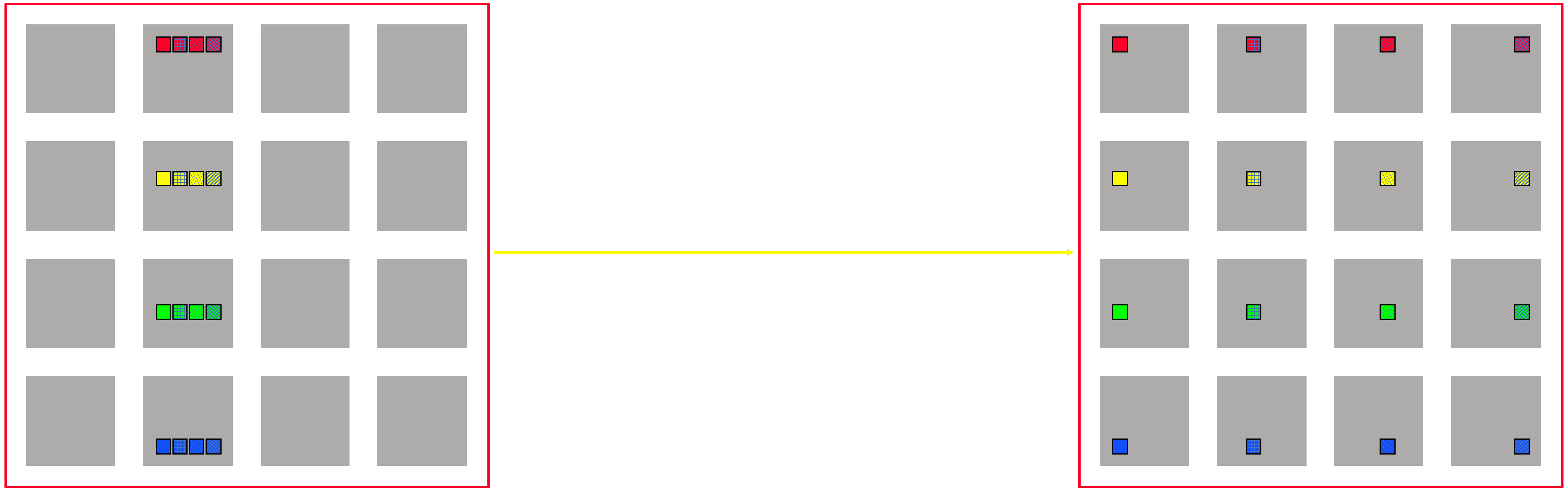
Example: 2D Broadcast

- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

Example: 2D Broadcast



- Option 3:
  - Scatter in column
  - Scatter in rows

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

Cost comparison

- **Option 1:**
  - **MST broadcast in column**
  - **MST broadcast in rows**
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(c)\alpha + log(c)n\beta$$

$$\frac{log(r)\alpha + log(r)n\beta}{log(p)\alpha + log(p)n\beta}$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- **Option 2:**
  - **Scatter in column**
  - **MST broadcast in rows**
  - **Allgather in columns**
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(c)\alpha + \frac{c-1}{c}n\beta$$

$$log(r)\alpha + log(r)\frac{n}{c}\beta$$

$$(c-1)\alpha + \frac{c-1}{c}n\beta$$

$$(log(p) + c - 1)\alpha + \left(2\frac{c-1+log(r)}{c}\right)n\beta$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- **Option 3:**
  - **Scatter in column**
  - **Scatter in rows**
  - **Allgather in rows**
  - **Allgather in columns**

$$\log(c)\alpha + \frac{c-1}{c} n\beta$$

$$\log(r)\alpha + \frac{r-1}{r}\frac{n}{c}\beta$$

$$(r-1)\alpha + \frac{r-1}{r}\frac{n}{c}\beta$$

$$(c-1)\alpha + \frac{c-1}{c} n\beta$$

$$(\log(p) + r + c - 2)\alpha + 2\frac{p-1}{p} n\beta$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(p)\alpha + log(p)n\beta$$

$$(log(p) + c - 1)\alpha + \left(2\frac{c - 1 + log(r)}{c}\right)n\beta$$

$$(log(p) + r + c - 2)\alpha + 2\frac{p-1}{p}n\beta$$

Summary and Question

- MST -> when alpha dominates
- Ring -> when n*beta dominates
- 2D can be composed using 1D, 3D can be composed using 2D, …
- Latency / Bandwidth trade-offs

Recap

- Q1: Which collective primitive maps to the distributed SGD gradient synchronization step?
- Q2: How many messages do we need to transfer over the network for a single iteration of GPT-3 SGD update assuming 8-gpu parallelism?
- Q3: For Q2, assuming 1D mesh, should we use MST or Ring?

Collective Pros

- A set of structured / well-defined communication primitives
- Extremely well-optimized
- Beautiful math, easy to analyze, and easy to understand its performance

Collective Cons

- Lack of Fault Tolerance
  - What if one node (in the ring) is dead?
- Requires Homogeneity
  - What if one node computes slower than all other nodes?
  - What if one link has lower bandwidth than the other node?

Real Cluster:

- Need Fault tolerance
- Heterogeneous hardware setup

Next Topics

- This week 2 classes: Data base + Cloud Storage
  - Delta from previous year offering:
    - we skip a substantial part of relational database
    - spend more time one networking, HPC, and ML

- Next week: Parallelism and Big Data processing
  - We will come back to study how we address the problem of Collectives