Arhitecturi Paralele
Pipeline

Dr. Ing. Cristian Chilipirea – cristian.chilipirea@gmail.ro

Address Space Qualifiers

- __global
  - Accesibil de toate work-item-urile.
- __local
  - Accesibil doar de work-item-urile unui work-group.
- __private
  - Accesibil de un singur work-item.
- __constant

Synchronization

- void **barrier**(cl_mem_fence_flags *flags*)
- void **work_group_barrier**(cl_mem_fence_flags *flags*)
  - Toate work-item-urile unui work-group trebuie să aștepte la barieră pentru a trece mai departe.
  - Dacă bariera în if toate să intre pe aceeași ramură a if-ului.
  - Dacă bariera în for toate să facă același număr de iterații ale for-ului.

Built in Functions

- uint **get_work_dim**()
  - Cu câte dimensiuni a fost rulat kernel-ul.
- size_t **get_global_size**(uint *dimindx*)
  - Mărimea globală pentru dimensiunea dimindx.
- size_t **get_global_id**(uint *dimindx*)
  - Locația în dimensiunea dimindx.
- size_t **get_local_size**(uint *dimindx*)
  - Mărimea work-group-ului pe dimensiunea dimindx.
- size_t **get_local_id**(uint *dimindx*)
  - Locația în work-group pe dimensiunea dimindx.

Synchronization

- No mutex
- No semaphore
- Yes atomics... Tons of atomics.
  - atomic_add
  - atomic_sub
  - atomic_xchg
  - atomic_inc
  - atomic_min
  - atomic_and

OpenCL Hardware vs Software

Work Item (Thread) executes on a core

Work Group executes on an **SM**

GTX 1080 (Pascal) **Streaming Multiprocessor** (Compute Unit)

Detalii Work Group

- **Work Group**
  - Conține mai mulți work items (threads)
  - Toate work items dintr-un work group împart Shared memory (__local)
  - Barieră din kernel există doar la nivel de work group
    - **Toate** work items din work group trebuie să apeleze bariera
  - Atomicele pot funcționa la nivel de work group
  - Work items din același work group pot rula în **lock step**
- Unul (sau mai mulți – depinde de implementare) work group pe **SM**
- **MAX** un **SM** pe work group

Barieră globală

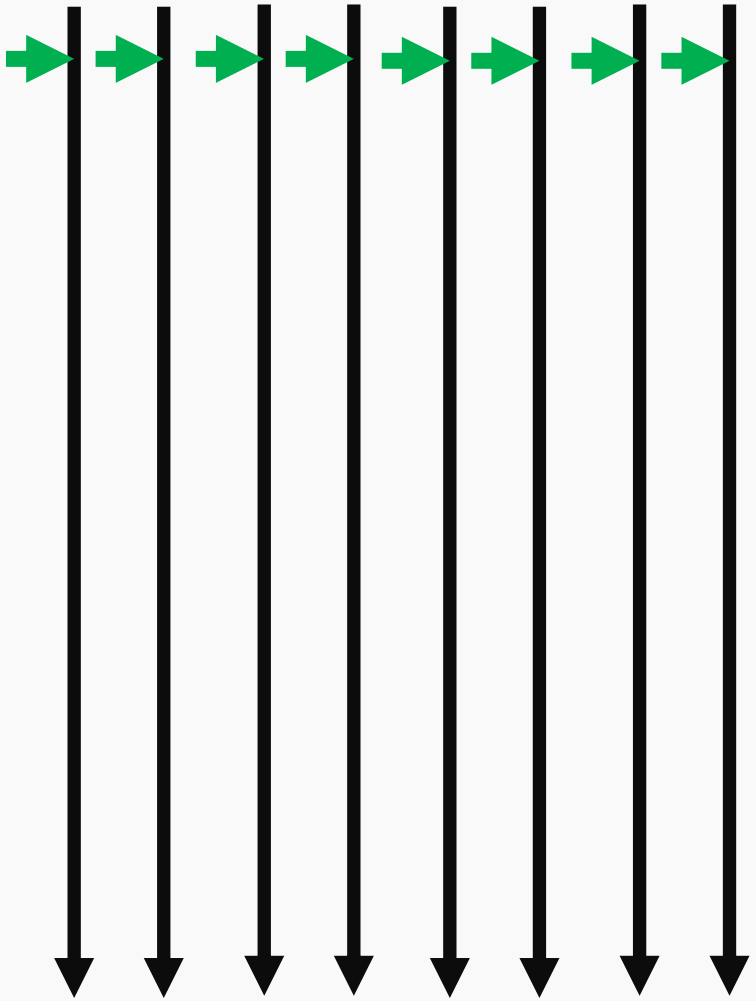
- Default toate comenzile introduse într-un queue sunt executate în ordine și se așteaptă terminarea uneia ca să înceapă alta.
- Dacă avem nevoie de barieră globală, singura metodă e să separăm codul la acea barieră și să executăm porțiunea de după barieră printr-un nou apel clEnqueueNDRangeKernel (fie că e același kernel cu alți parametri sau altul)

OpenCL Hardware vs Software

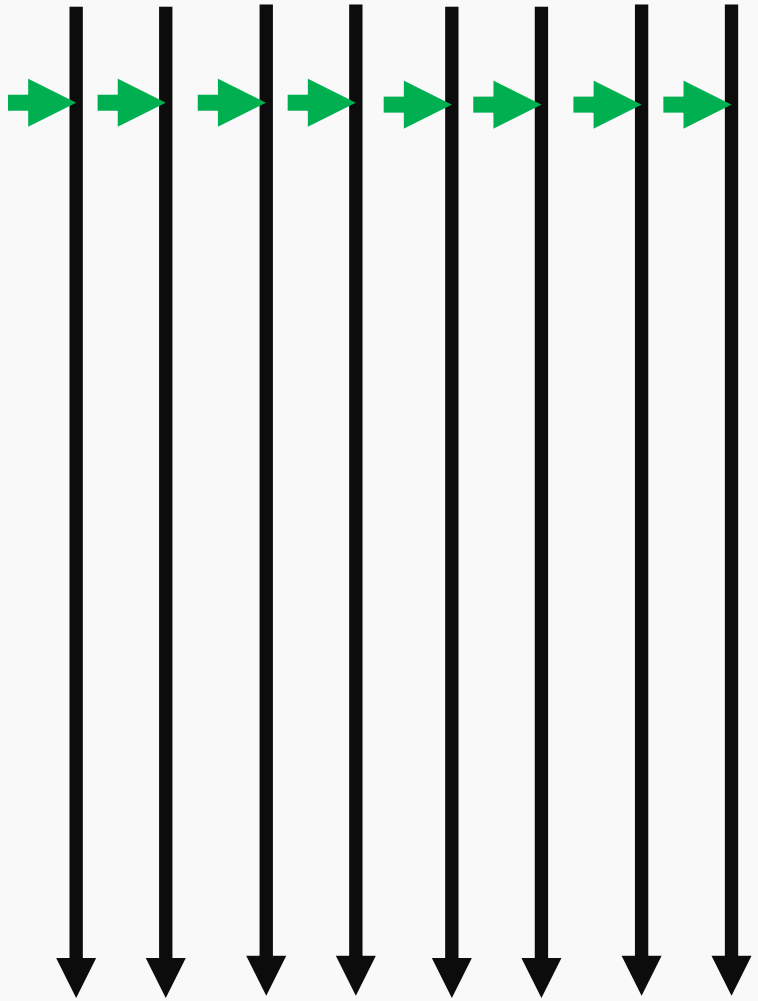
- GTX 1080
  - CUDA cores: 2560
  - Max Compute Units (raportat de OpenCL): 20
  - CUDA cores per SM: 128
  - Max Work Group Size (raportat de OpenCL): 1024
- i7-6800k
  - Cores: 6 (12 hyper-threaded)
  - Max Compute Units (raportat de OpenCL): 12
  - AVX2: operații pe vectori de 8 int
  - Max Work Group Size (raportat de OpenCL): 8192

Max Work Group Size poate să fie afectat de mărimea kernel-ului.

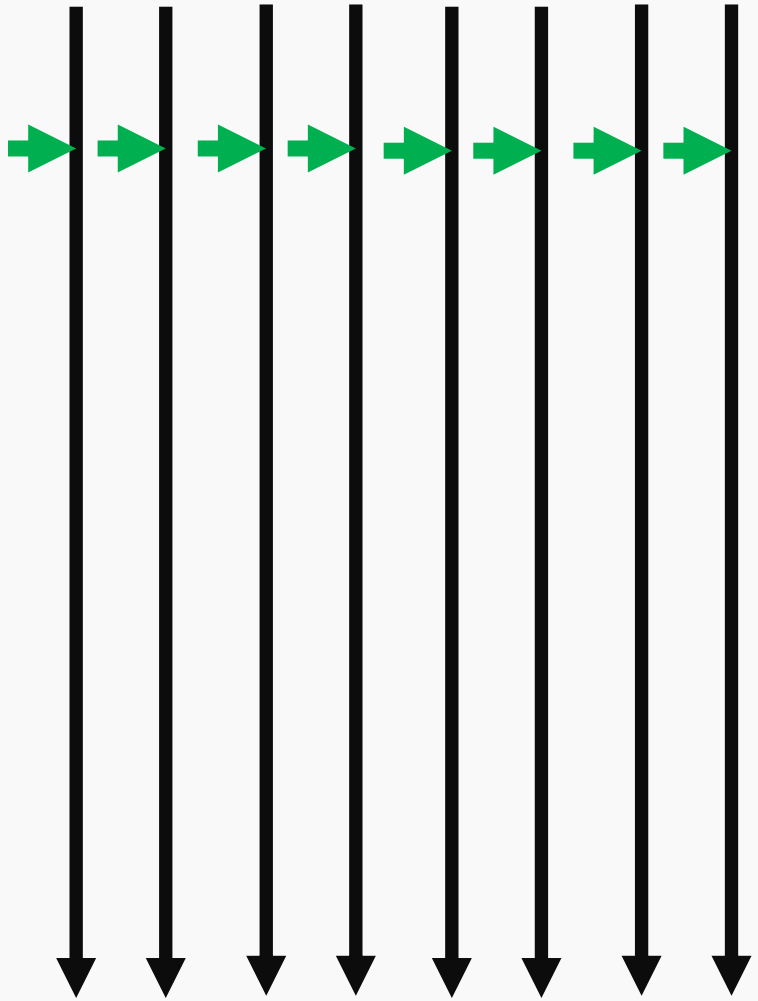
Execuție lock-step



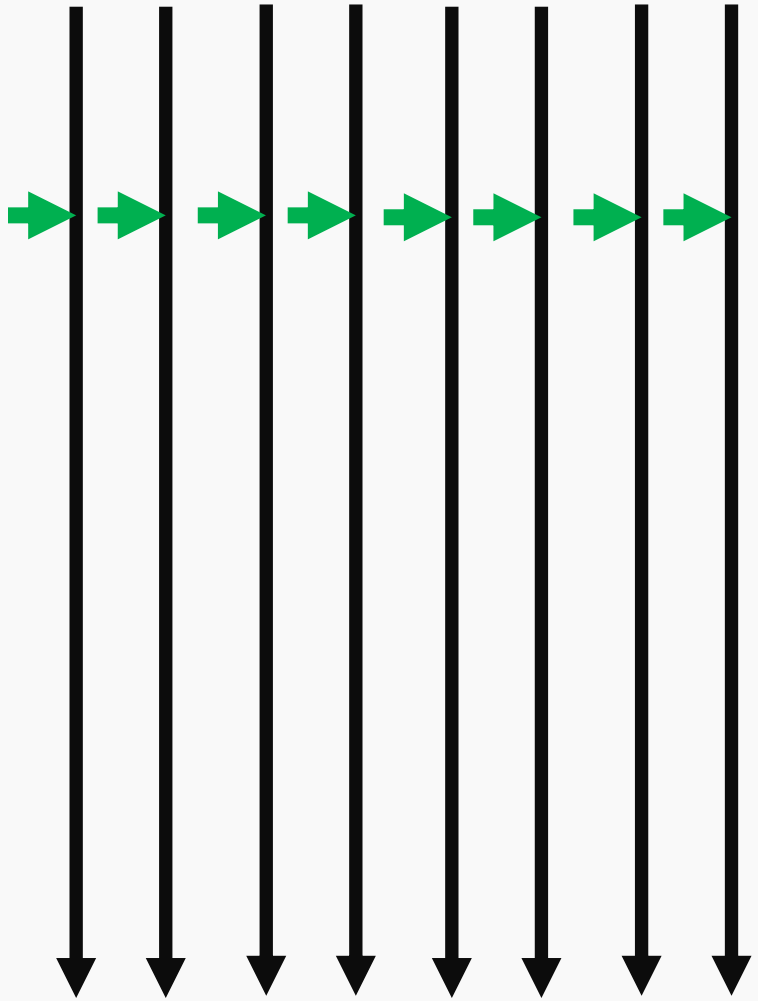
Execuție lock-step



Execuție lock-step

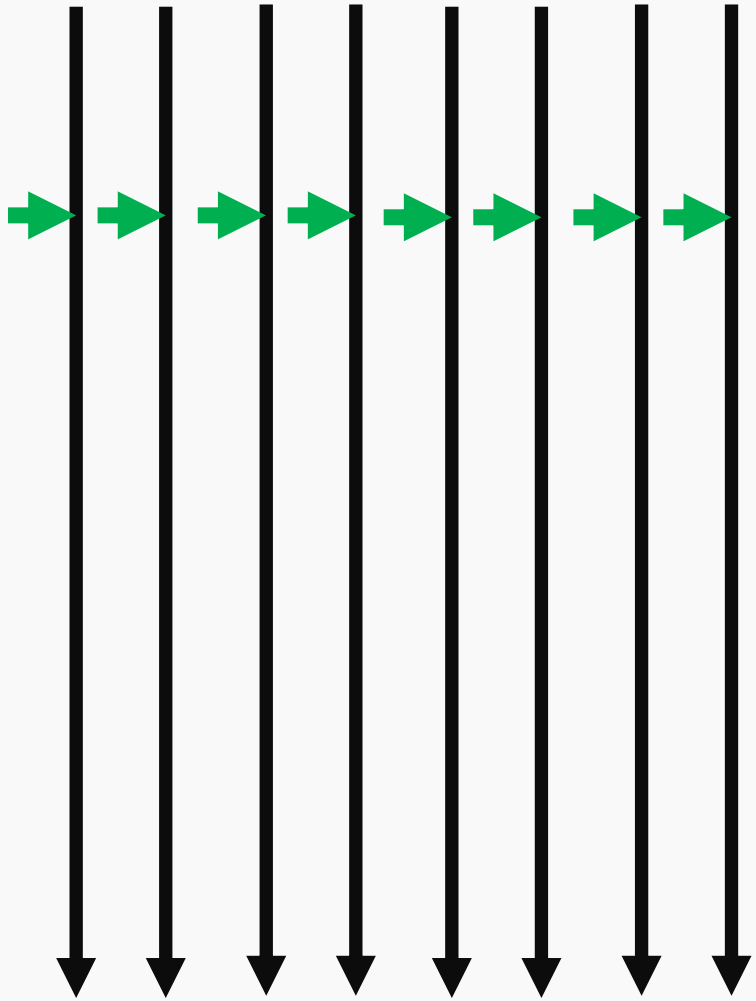


Execuție lock-step



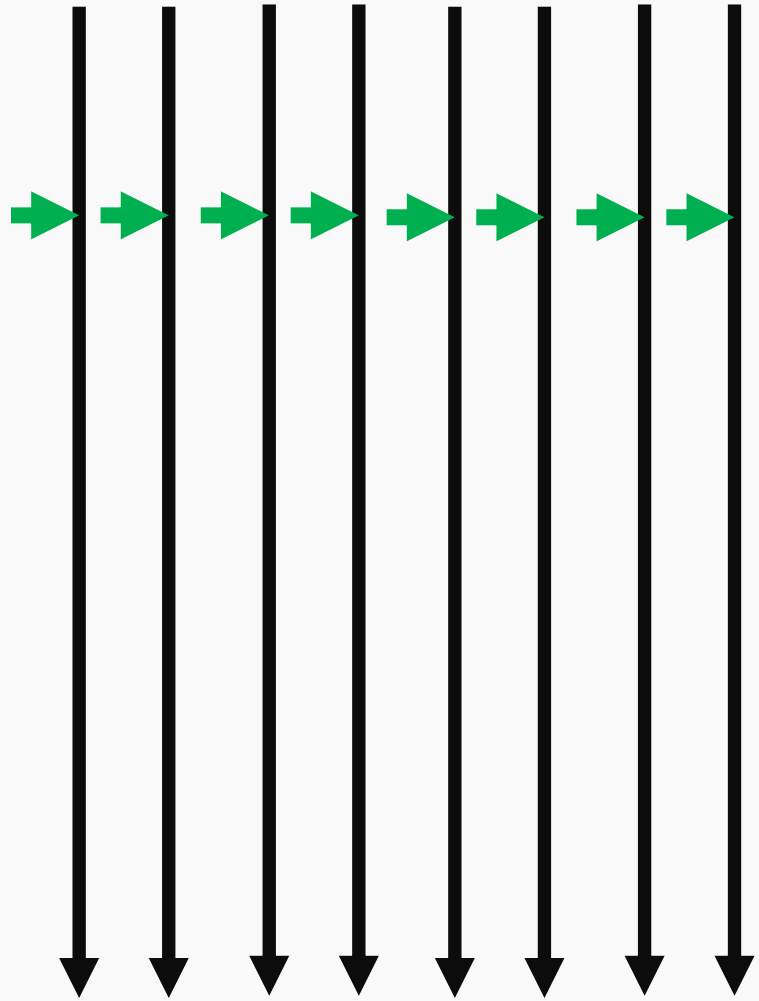
Execuție lock-step

Dar la if?



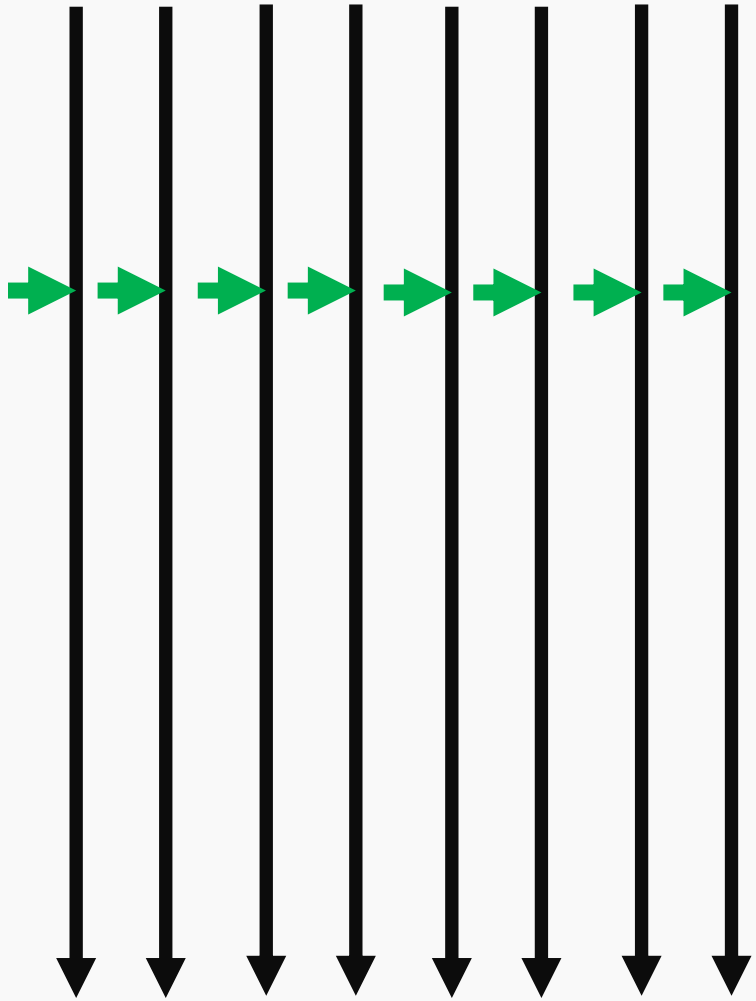
Execuție lock-step

```
if(thread_id%i==1)
        do_something();
else
        do_somethingelse();
```



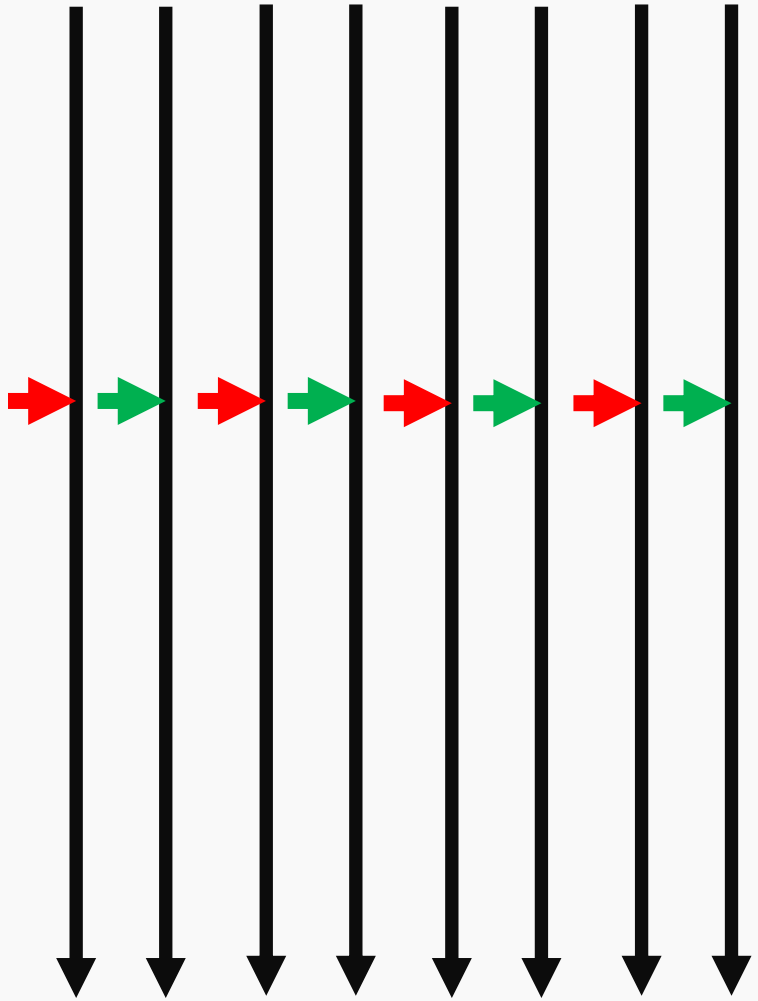
Execuție lock-step

```
if(thread_id%i==1)
        do_something();
else
        do_somethingelse();
```



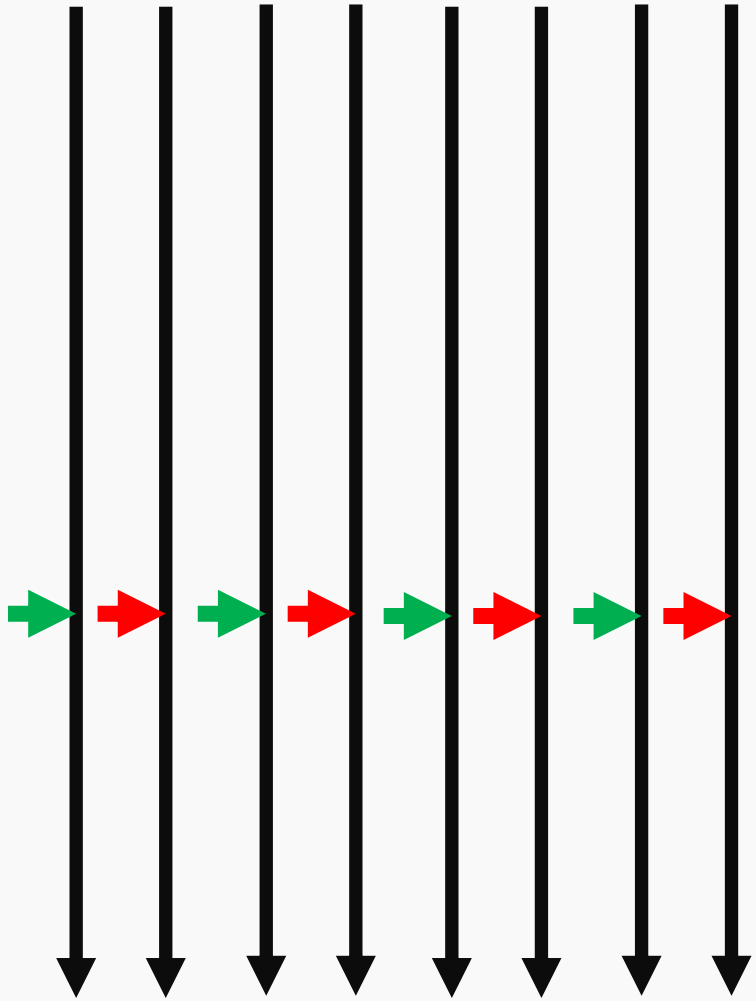
Execuție lock-step

```
if(thread_id%i==1)
        do_something();
else
        do_somethingelse();
```



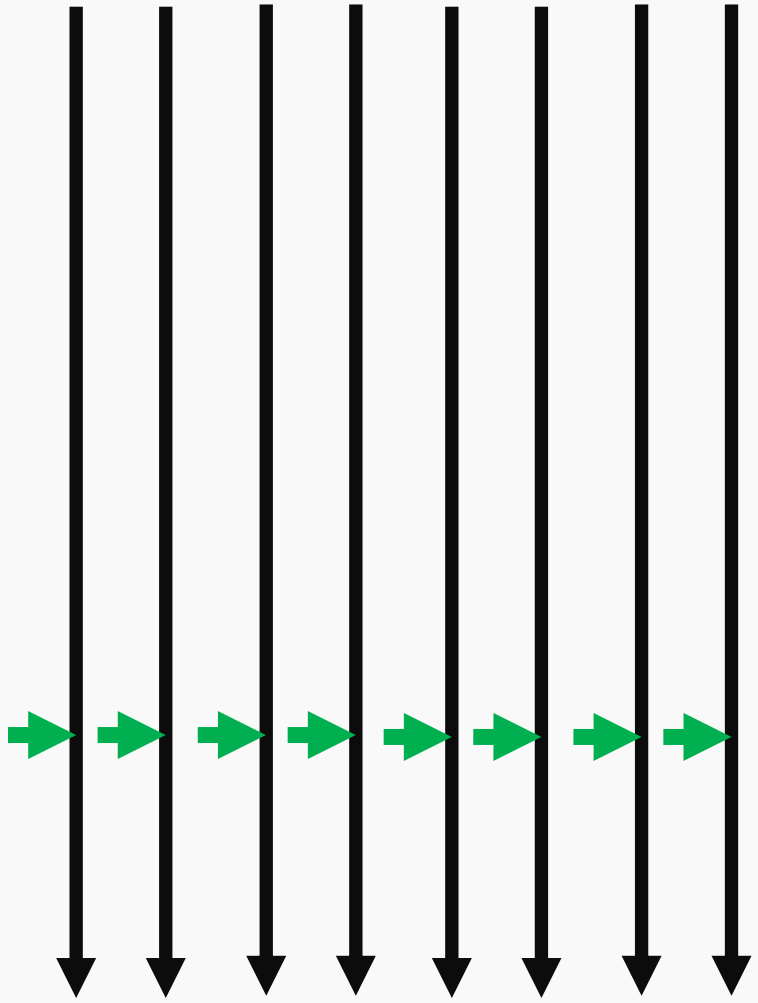
Execuție lock-step

```
if(thread_id%i==1)
        do_something();
else
        do_somethingelse();
```



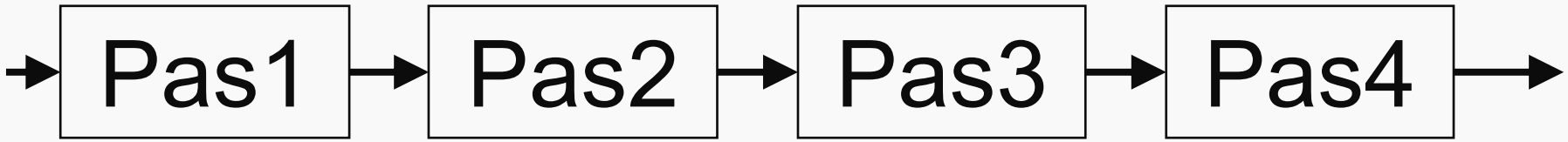
Execuție lock-step

```
if(thread_id%i==1)
        do_something();
else
        do_somethingelse();
```



Pipeline

- Pipeline de instrucțiuni CPU
- Pipeline grafic (randare, antialiasting)
- Diferiți algoritmi

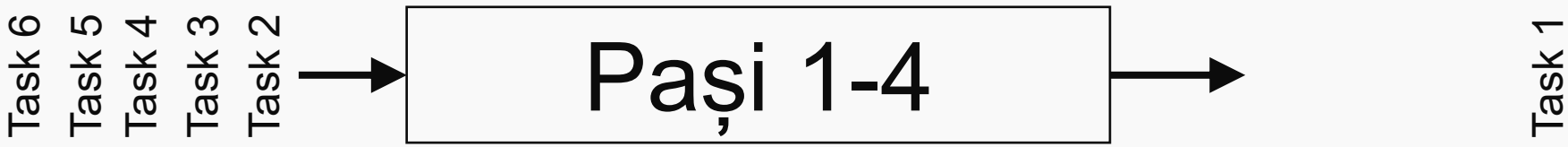


Un **pas** poate fi un:

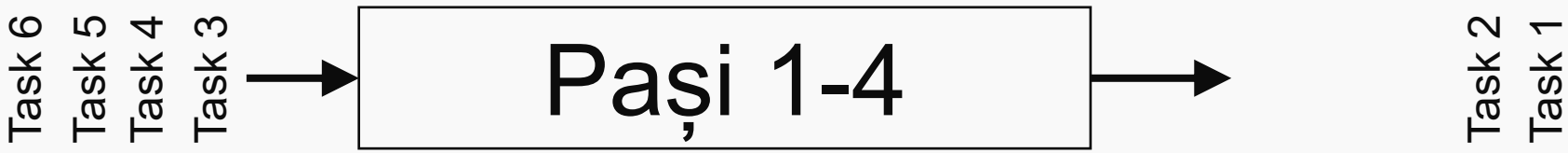
- thread
- proces
- element hardware

Fără Pipeline

Fără Pipeline



Fără Pipeline



Fără Pipeline

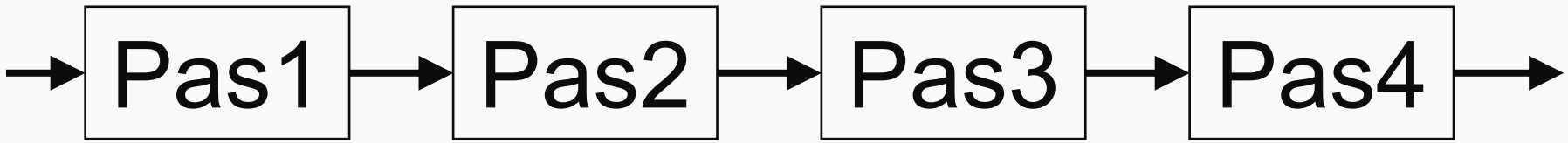
Fără Pipeline

Fără Pipeline

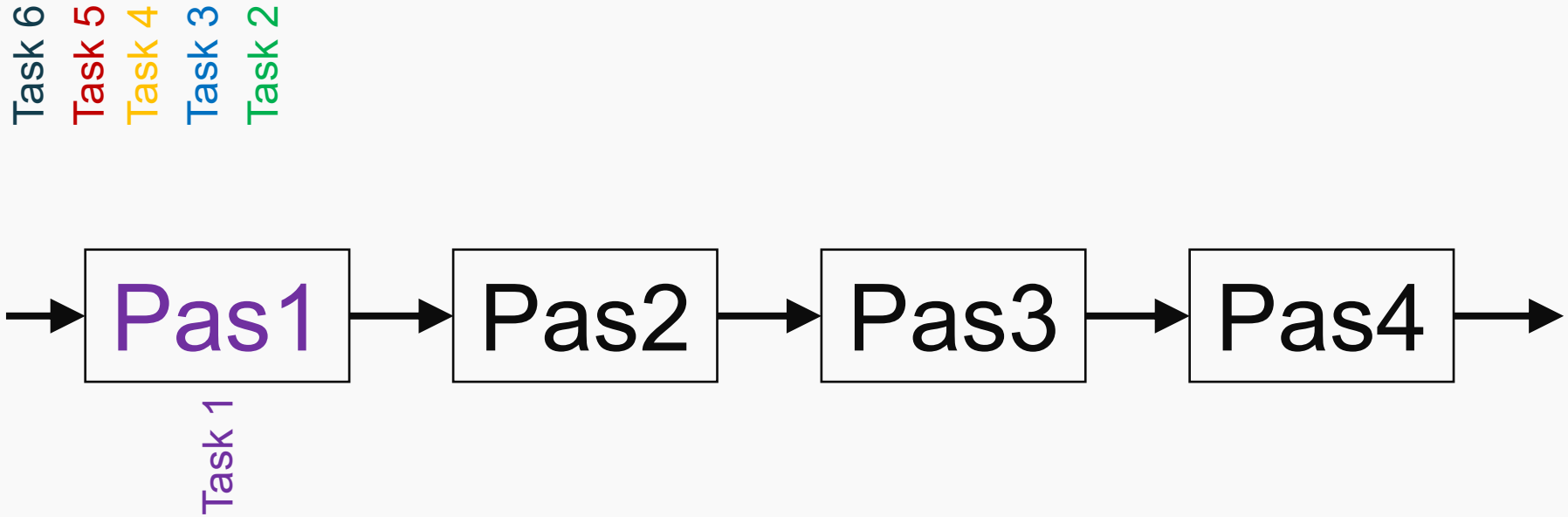
$$total\_execution\_time = task\_execution\_time * number\_of\_tasks$$

Pipeline

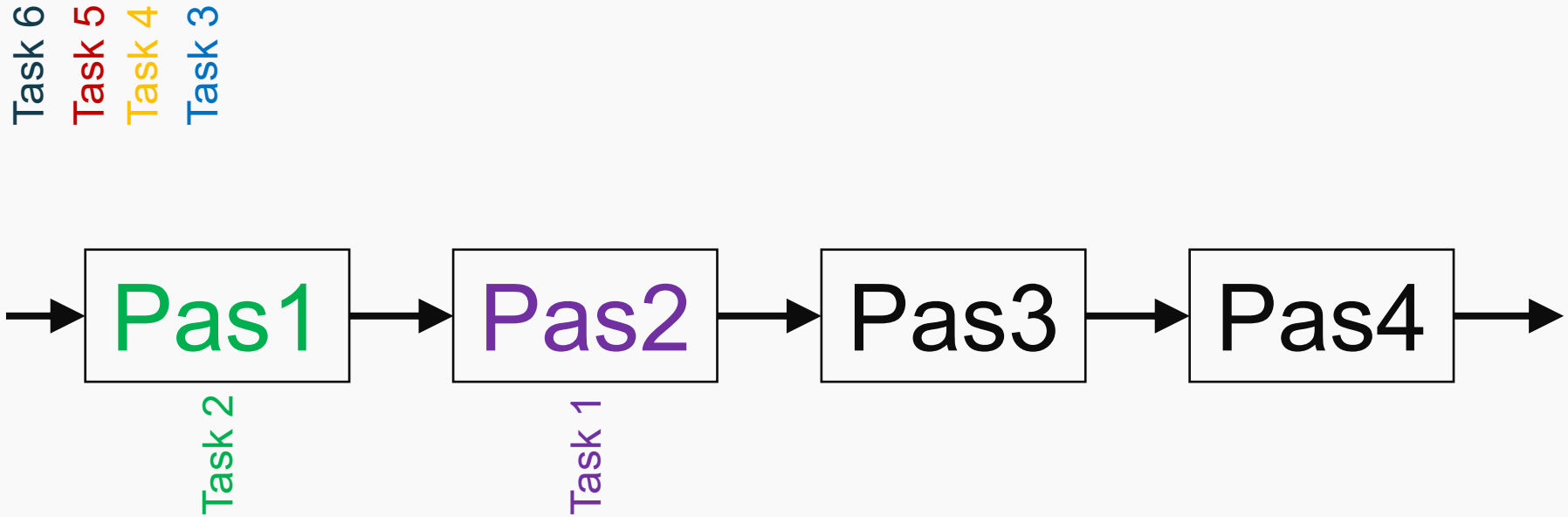
Task 6 Task 5 Task 4 Task 3 Task 2 Task 1



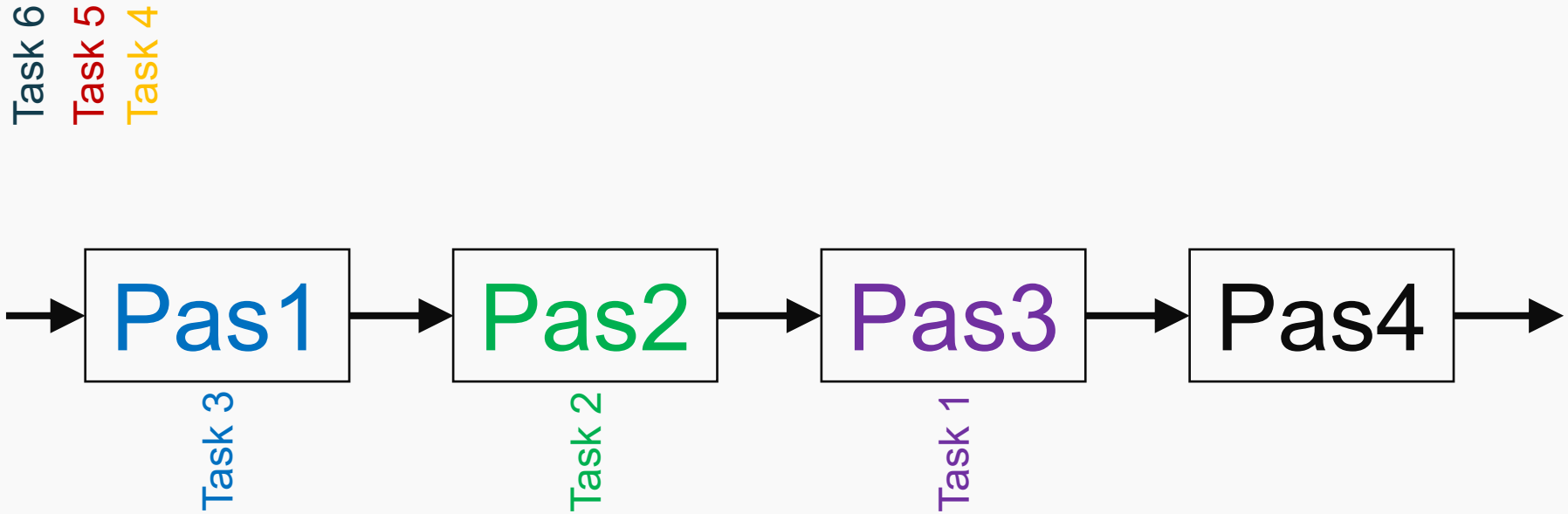
Pipeline



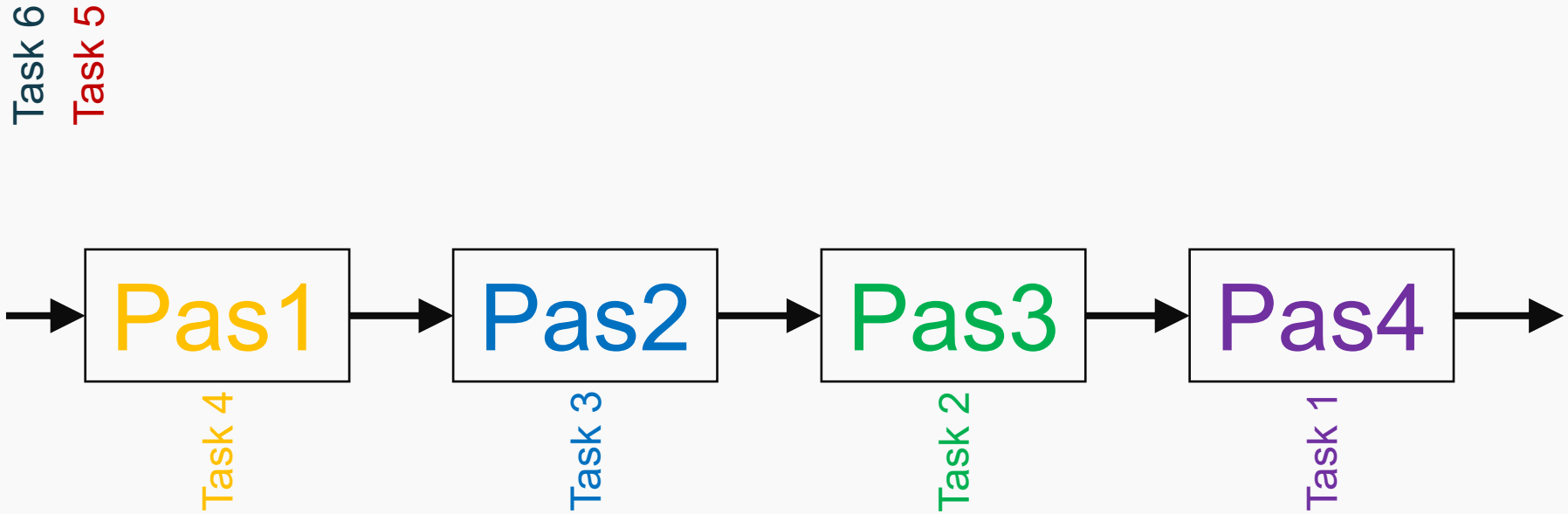
Pipeline



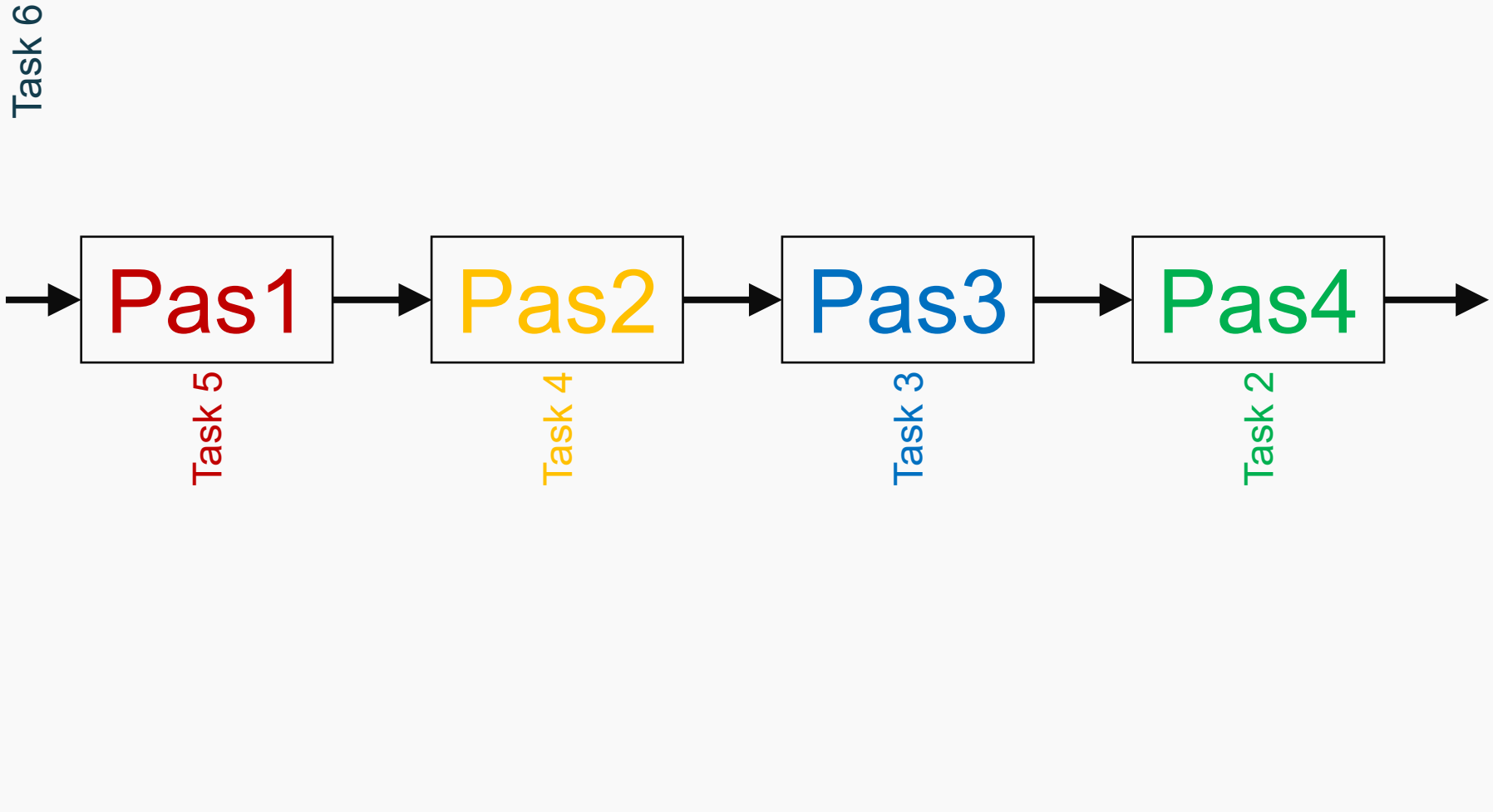
Pipeline



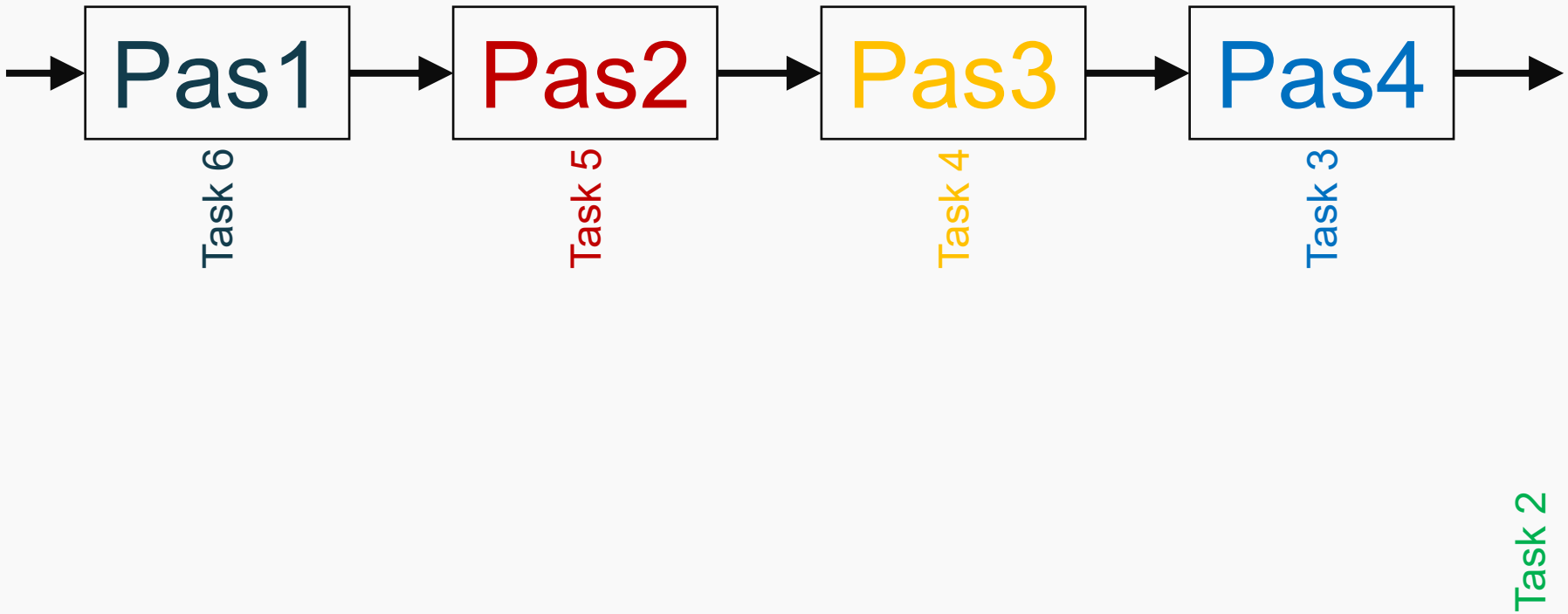
Pipeline



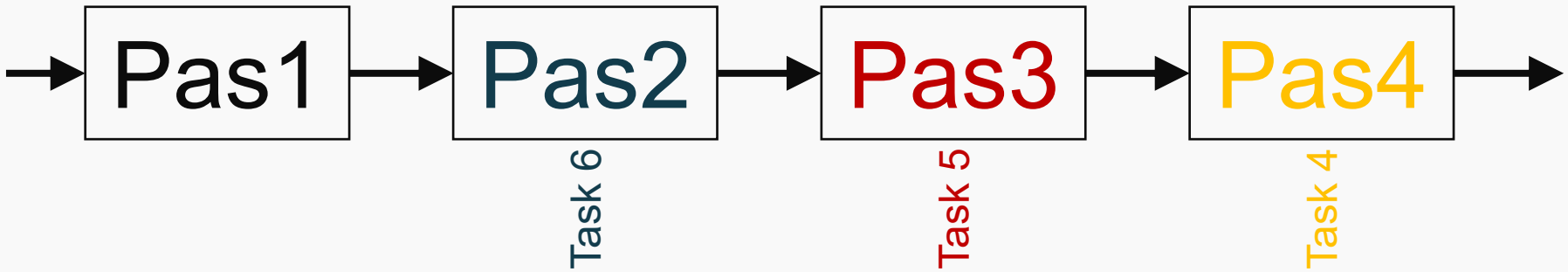
Pipeline



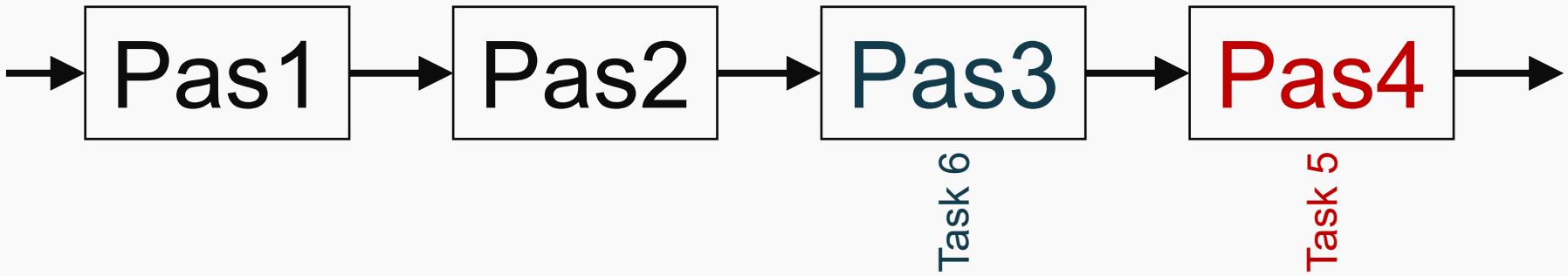
Pipeline



Pipeline

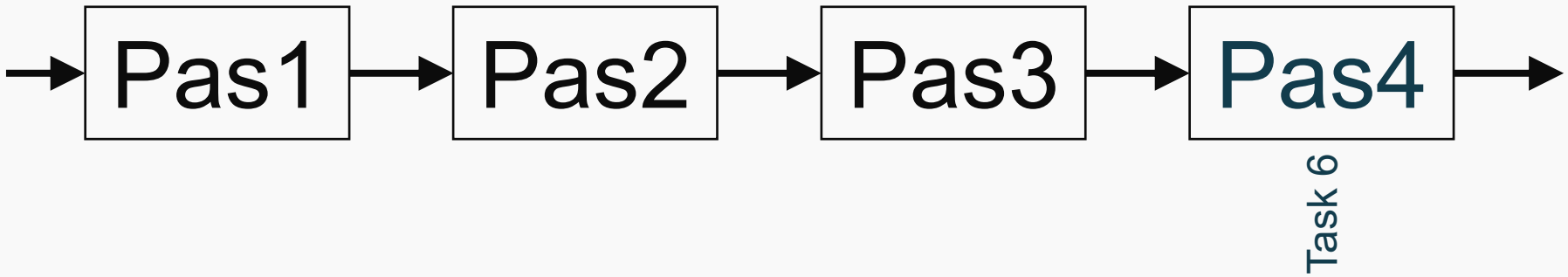


Pipeline



Pipeline

Ideal: $step\_execution\_time = \frac{task\_execution\_time}{number\_of\_steps}$

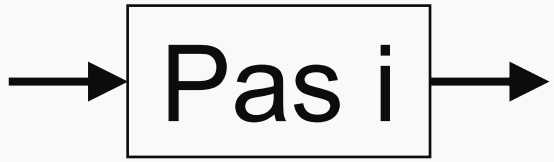


După avem mai mult decât $number\_of\_steps$ tasks timpul devine:

$total\_execution\_time = number\_of\_tasks * step\_execution\_time$

Un task se termină la fiecare “step tick”

Pipeline – ghid programare



Inițializare

for(un număr de pași) {

    primește date de la Pas(i-1)

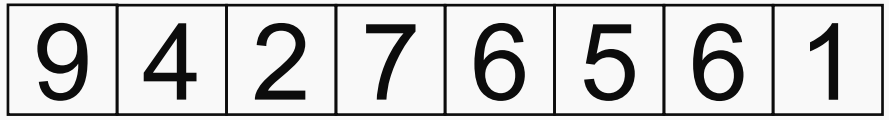
    procesează

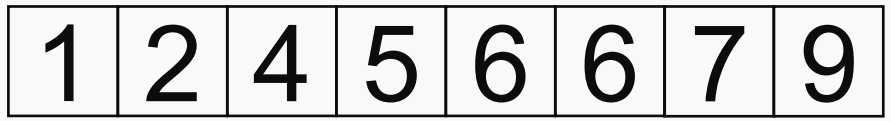
    trimite date la Pas(i+1)

}

Finalizare

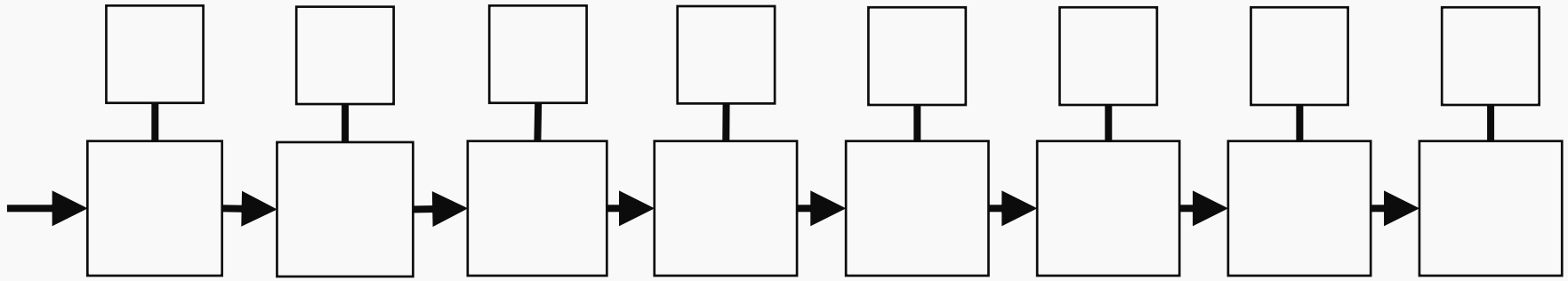
Sortare cu pipeline





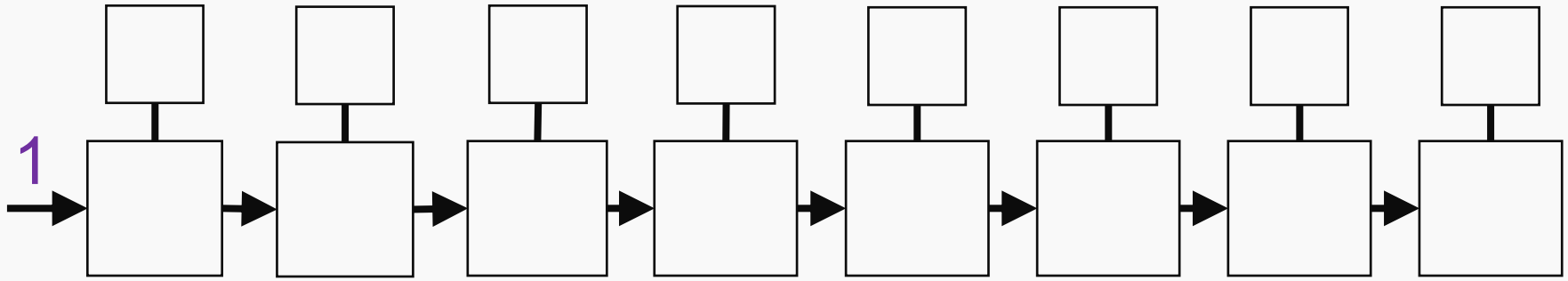
Sortare cu pipeline

9	4	2	7	6	5	6	1



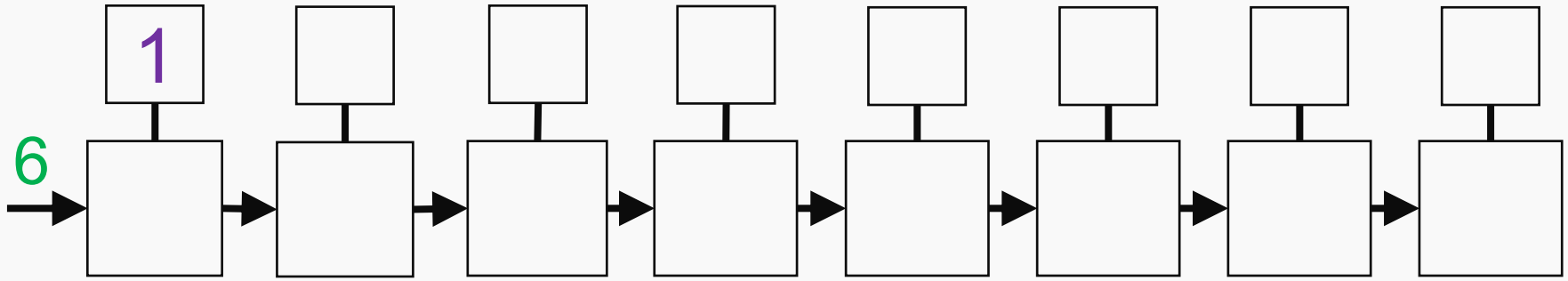
Sortare cu pipeline

9	4	2	7	6	5	6



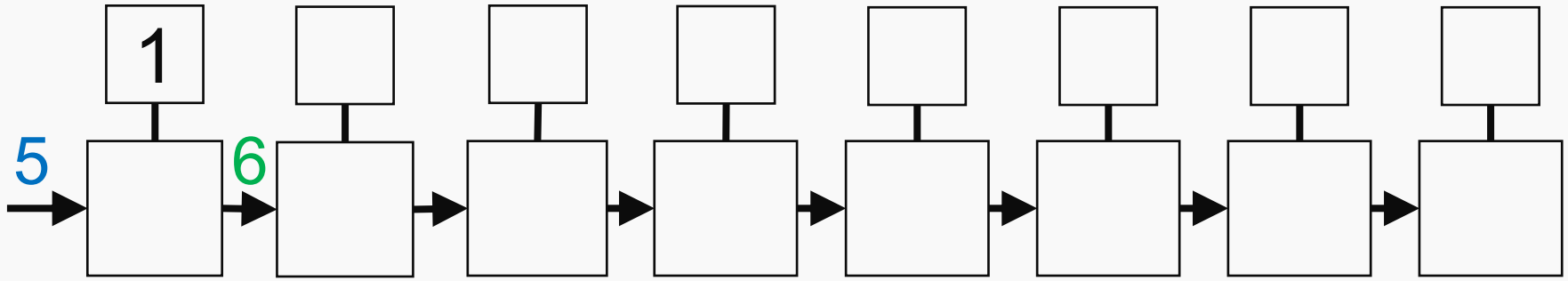
Sorting with pipeline

9	4	2	7	6	5



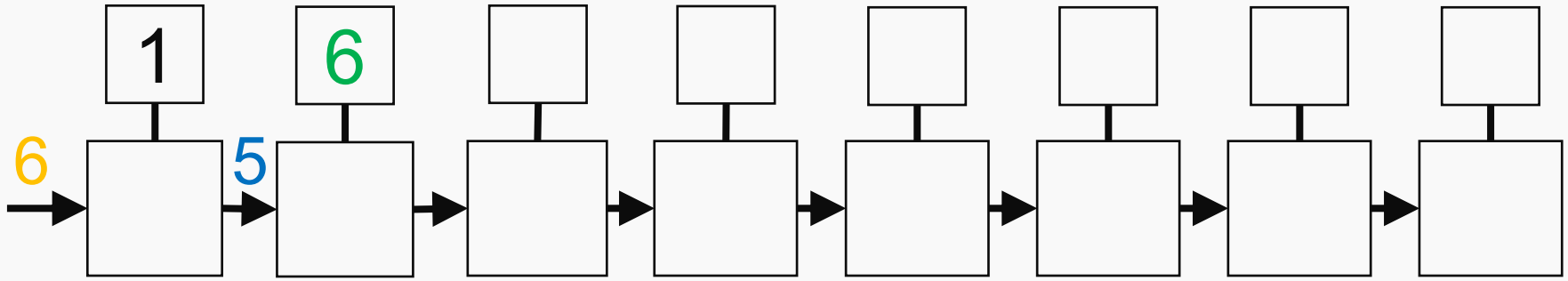
Sortare cu pipeline

9	4	2	7	6

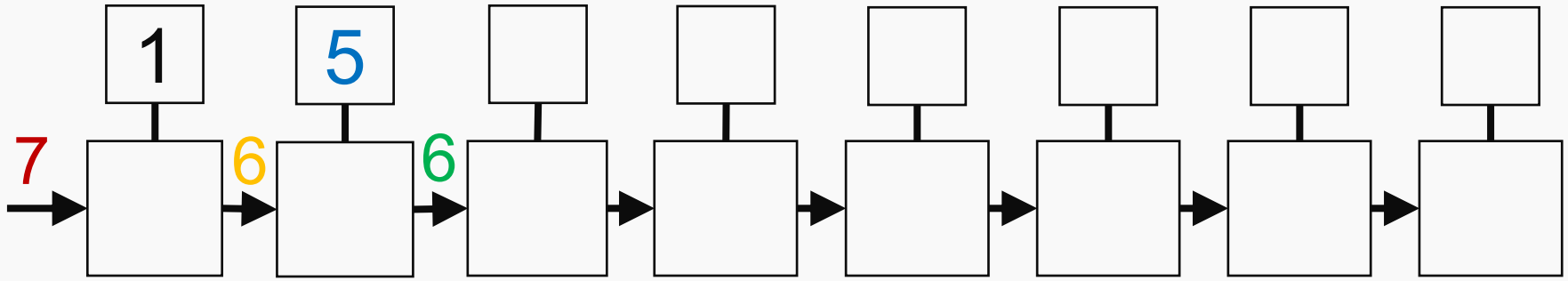


Sortare cu pipeline

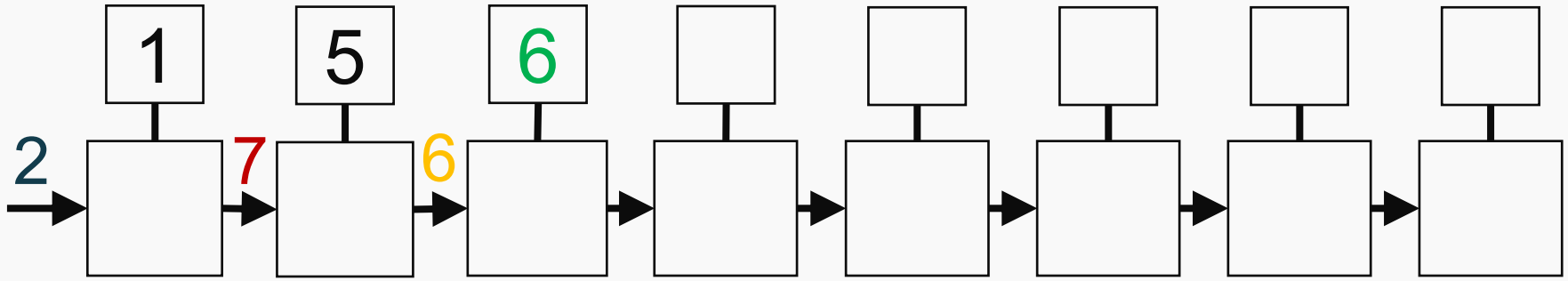
9	4	2	7



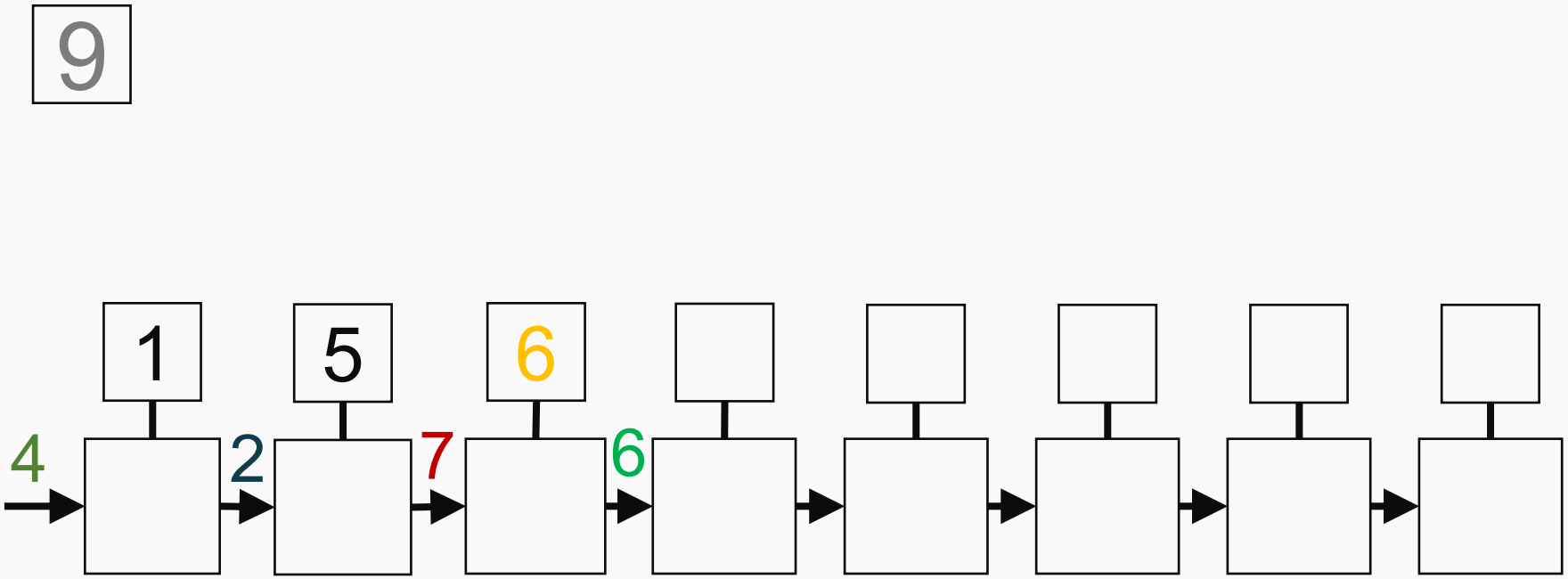
Sortare cu pipeline



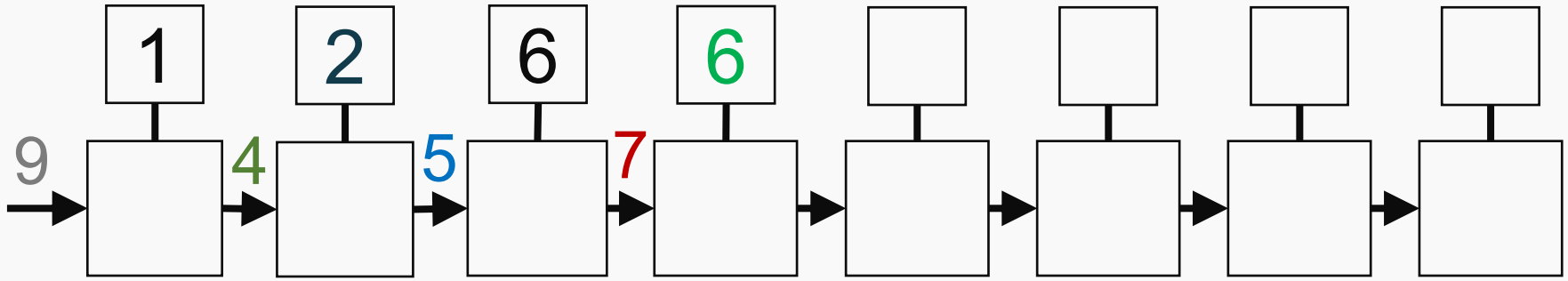
Sortare cu pipeline



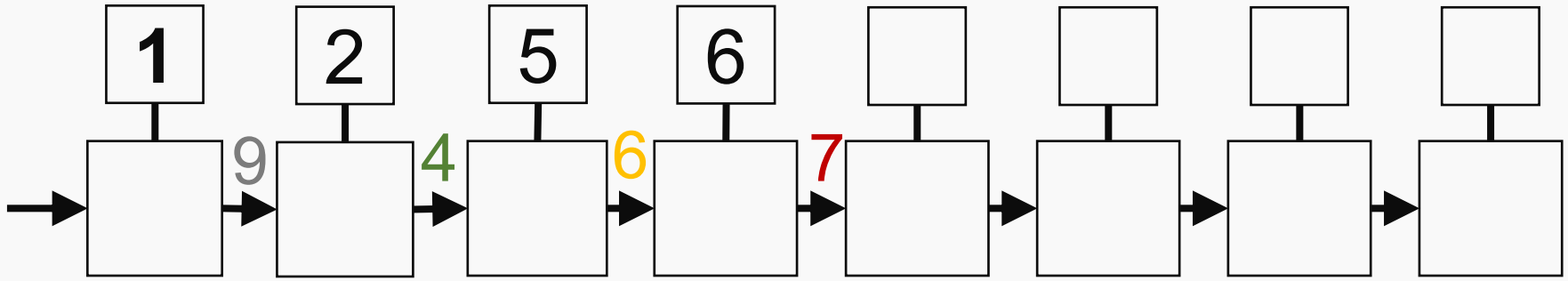
Sortare cu pipeline



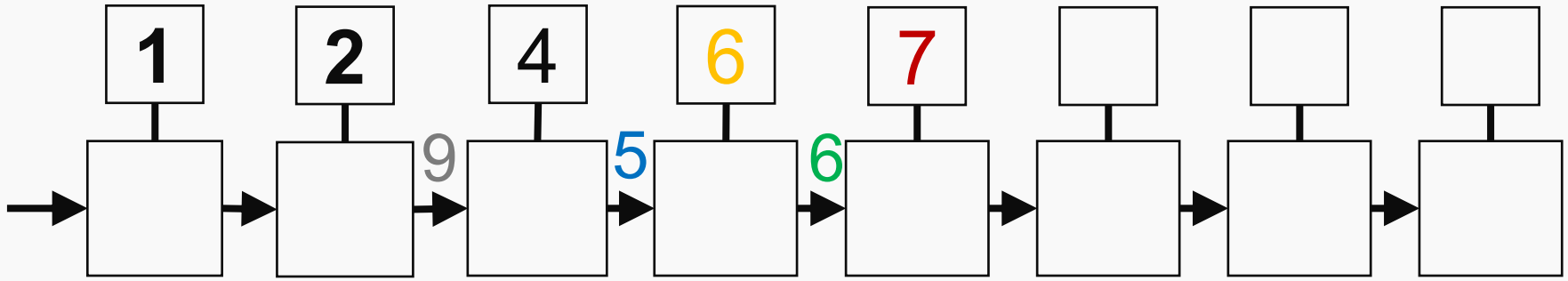
Sortare cu pipeline



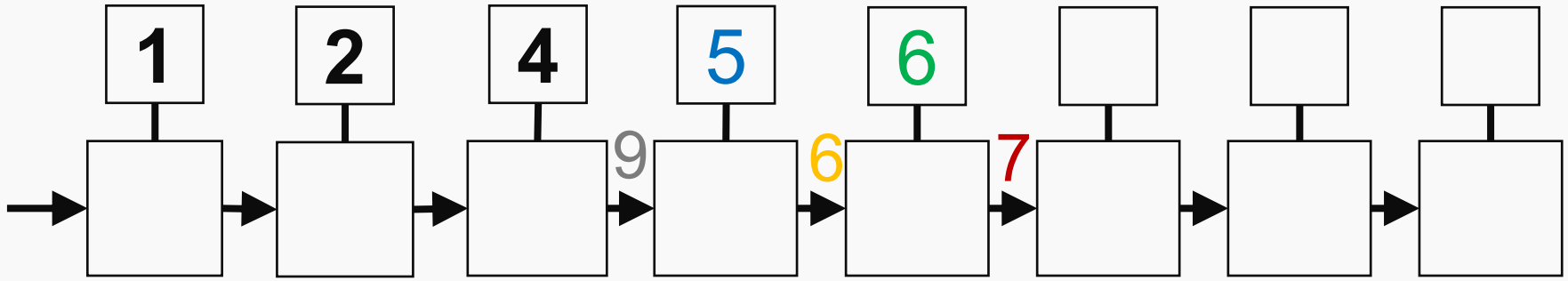
Sortare cu pipeline



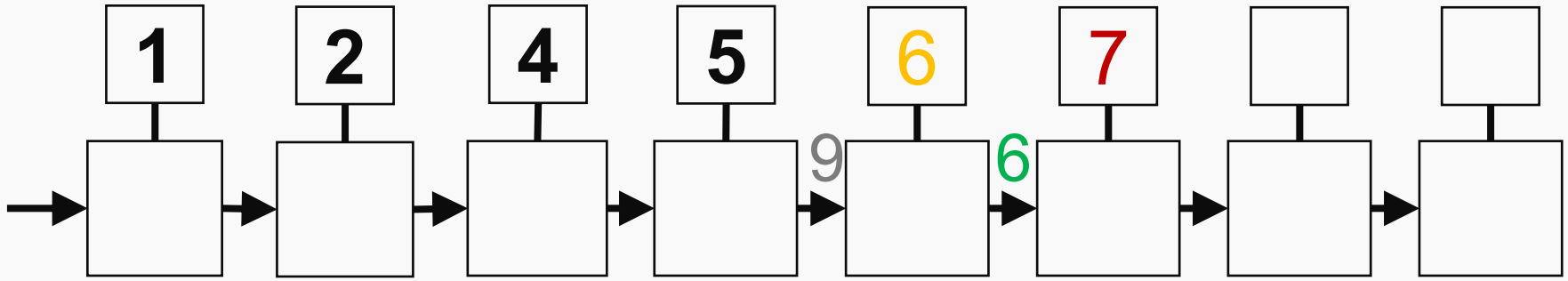
Sortare cu pipeline



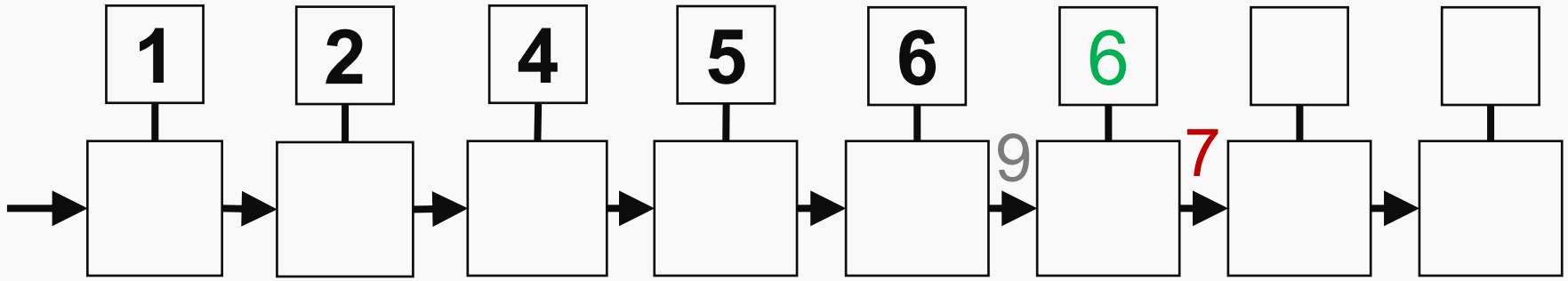
Sortare cu pipeline



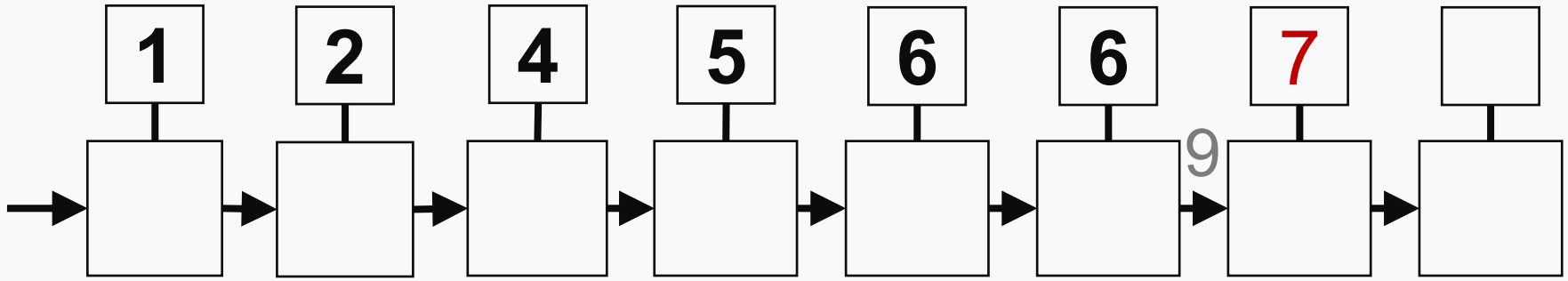
Sortare cu pipeline



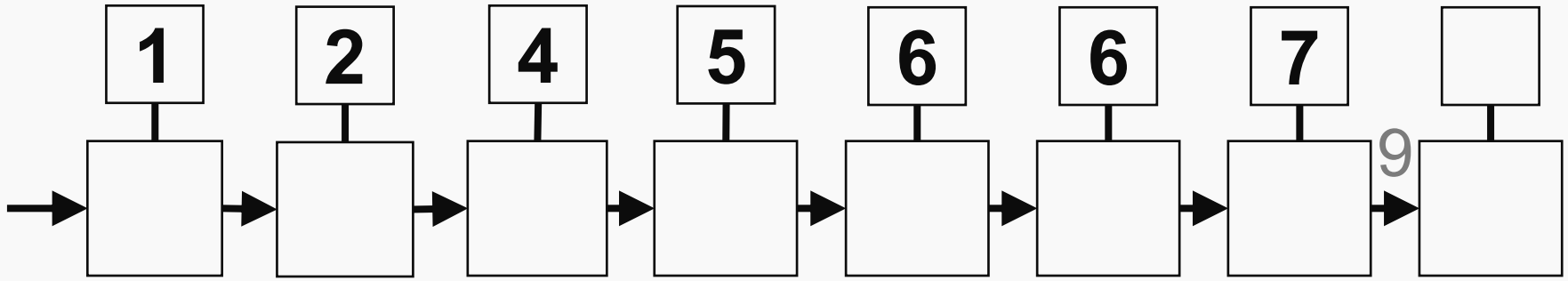
Sortare cu pipeline



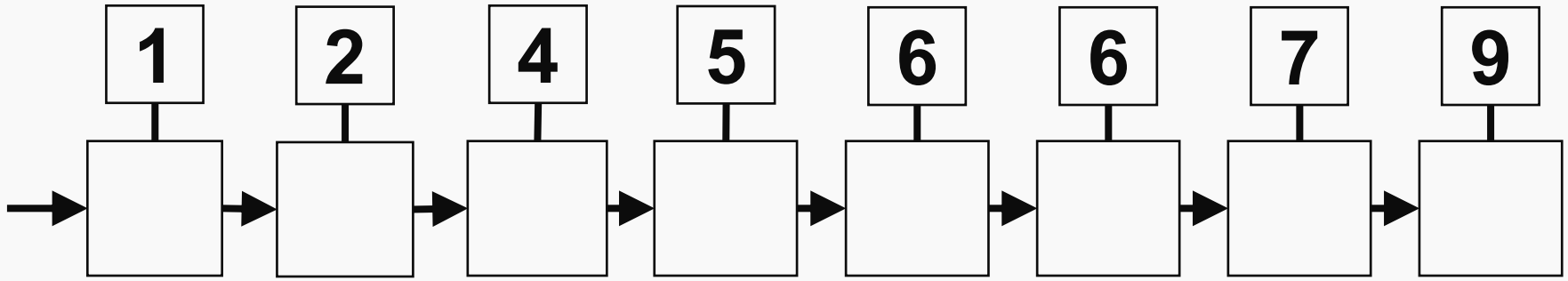
Sortare cu pipeline



Sortare cu pipeline



Sortare cu pipeline



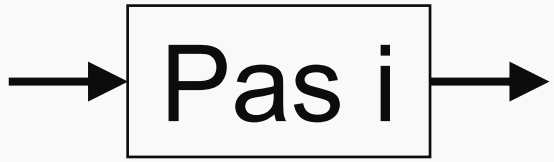
Sortare cu pipeline - complexitate

$$O(N)$$

pentru P=N

Dar comunicația e foarte lentă

Pipeline – ghid programare



Inițializare

for(un număr de pași) {

    primește date de la Pas(i-1)

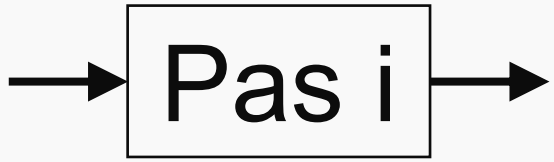
    procesează

    trimite date la Pas(i+1)

}

Finalizare

Sortare cu Pipeline – ghid programare



noop;

for(fiecare face cu o operație mai puțin decât pasul precedent) {

primește număr de la Pas(i-1)

ține local numărul minim între cel primit sau cel avut

trimite numărul mai mare la Pas(i+1)

}

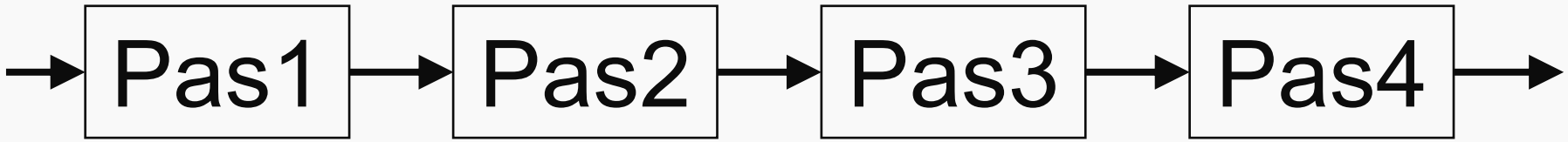
Scrie numărul la poziția i

Calcul unui polinom cu pipeline

$$P(x) = \sum_{i=0}^{n} a_i x^i = a_0 x^0 + a_1 x^1 + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n, n \geq 0$$

Se dorește să calculăm valoarea lui P pentru diverși x.
Use case: desenarea graficului aferent polinomului.

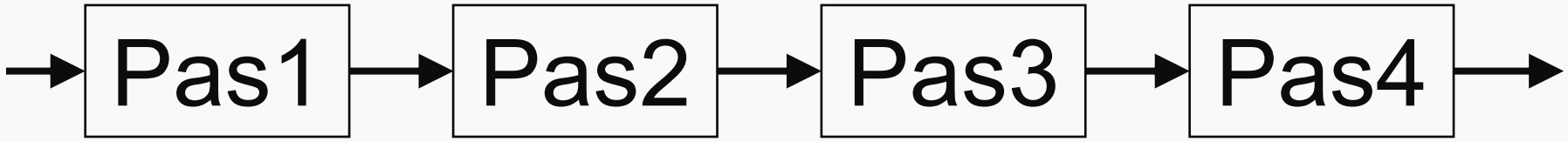
Pipeline



Calcul unui polinom cu pipeline

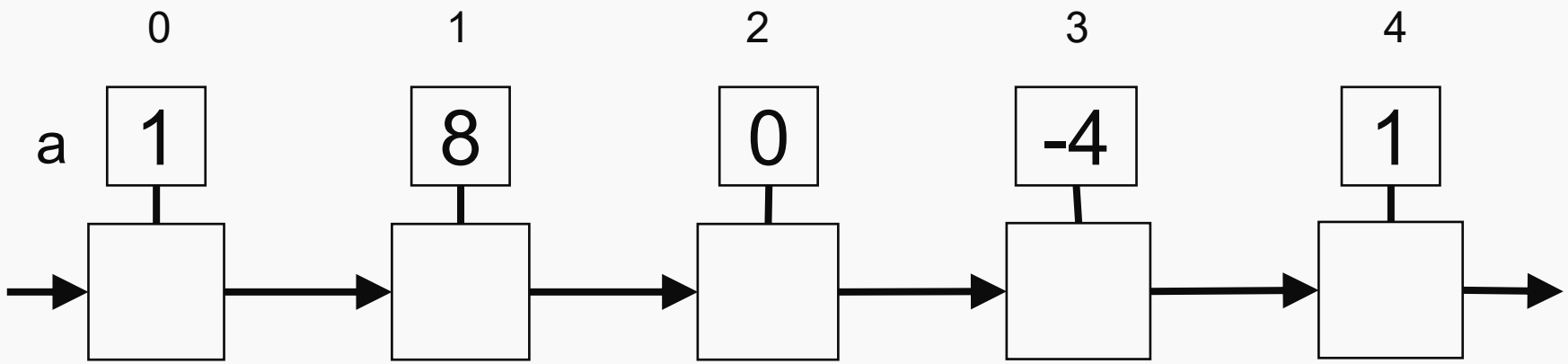
$$P(x) = 1 + 8x + (-4)x^3 + x^4$$

	0	1	2	3	4
a	1	8	0	-4	1



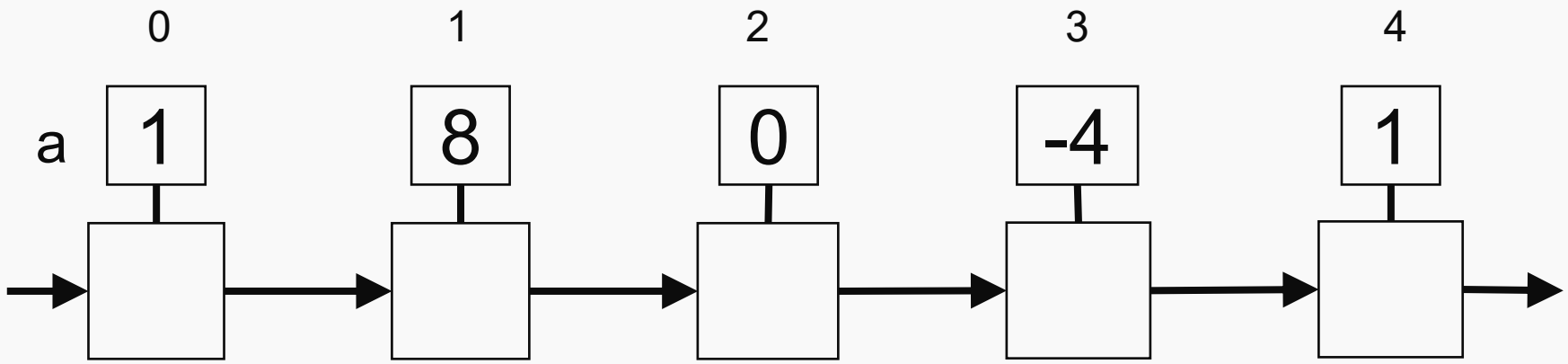
Calcul unui polinom cu pipeline

$$P(x) = 1 + 8x + (-4)x^3 + x^4$$



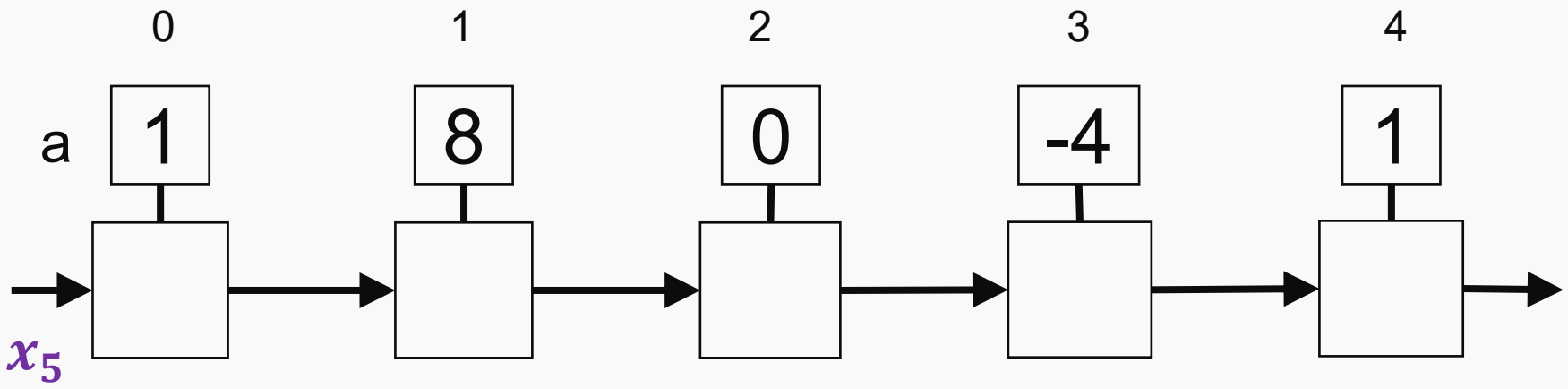
Calcul unui polinom cu pipeline

x_1 x_2 x_3 x_4 x_5

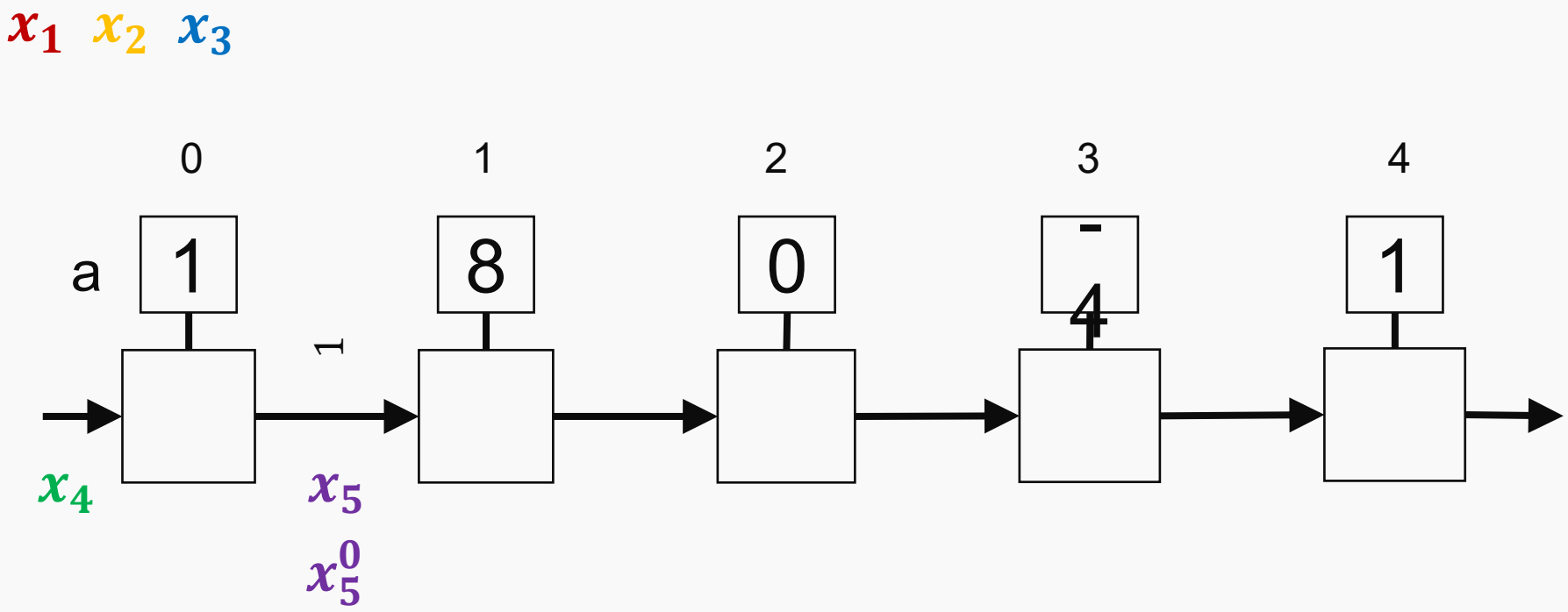


Calcul unui polinom cu pipeline

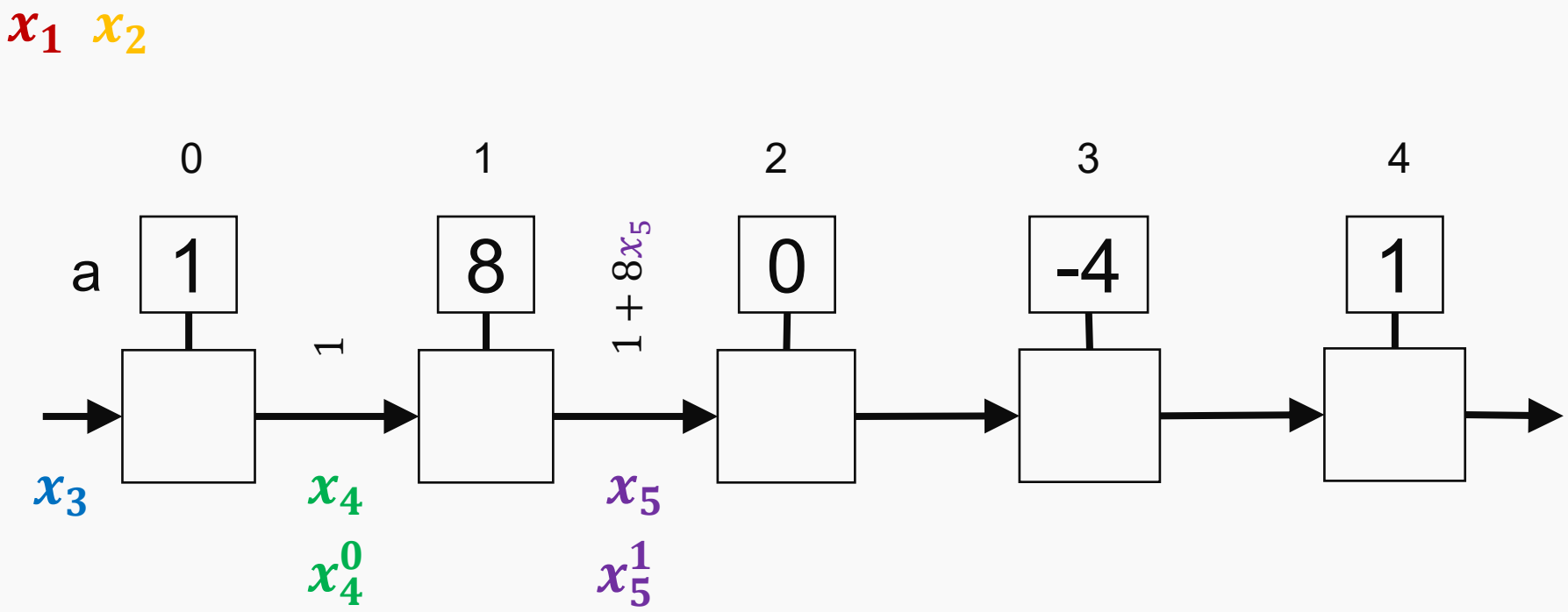
x_1 x_2 x_3 x_4



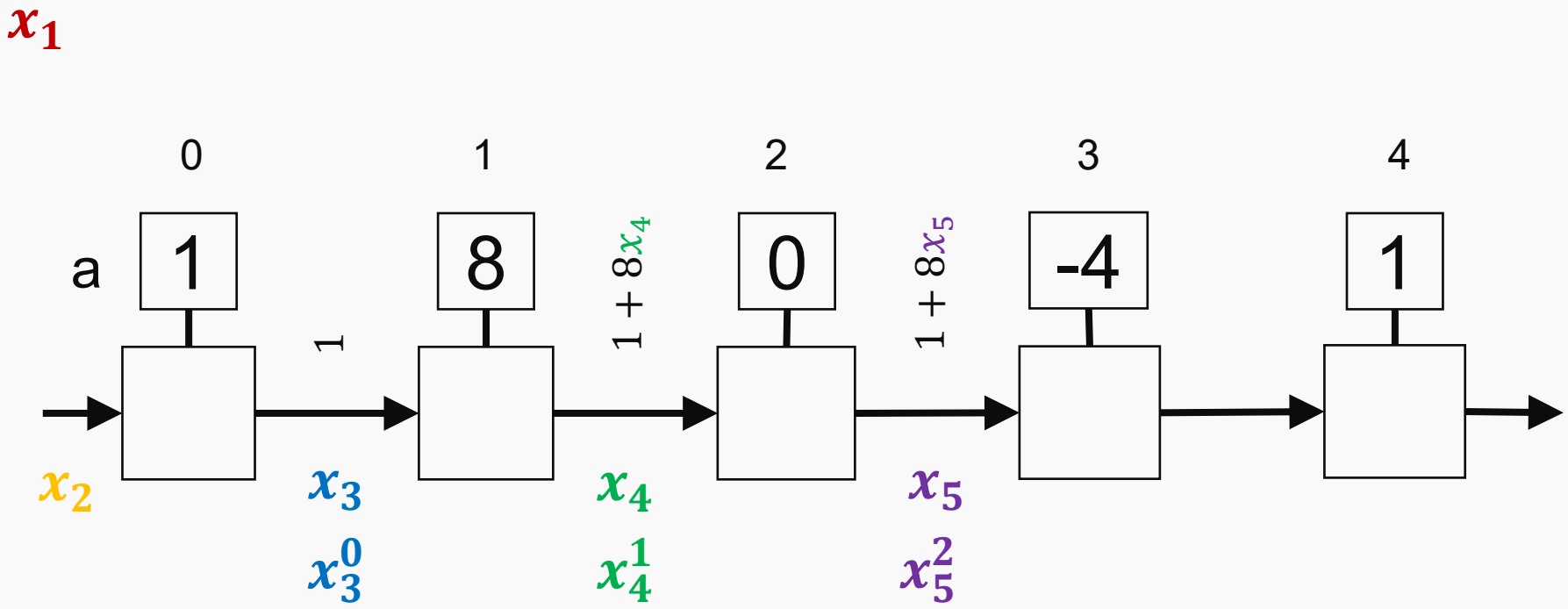
Calcul unui polinom cu pipeline



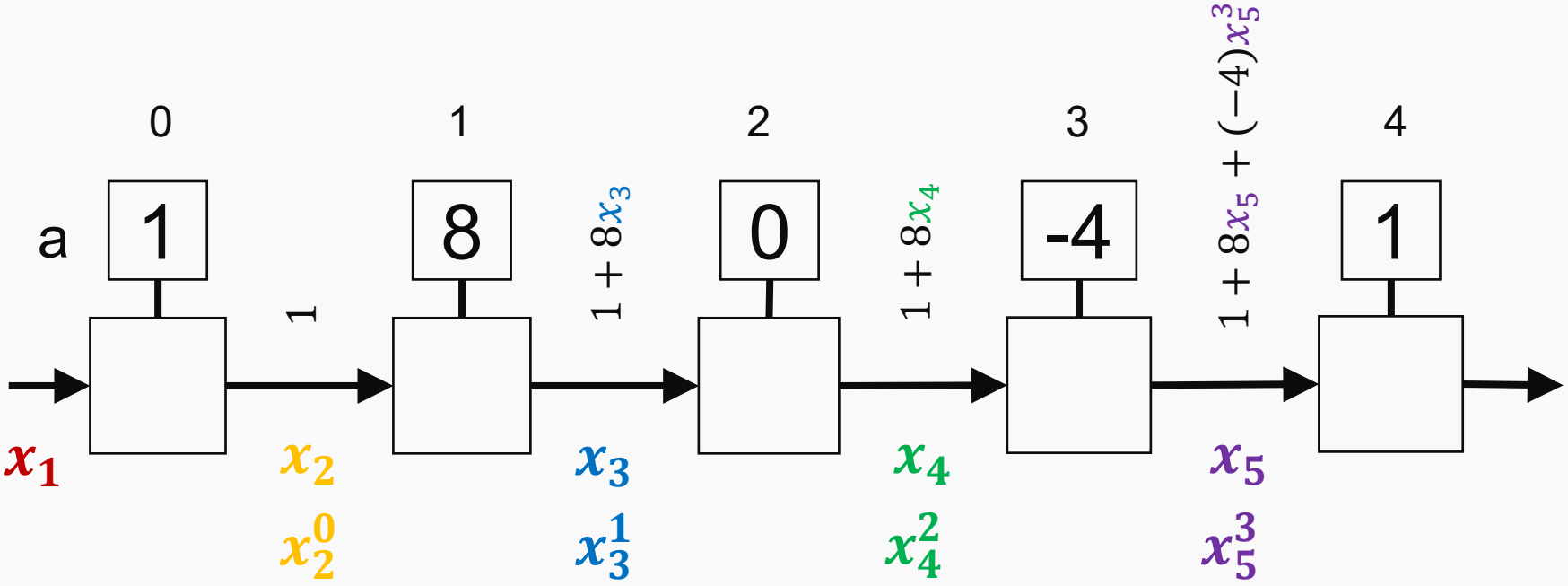
Calcul unui polinom cu pipeline



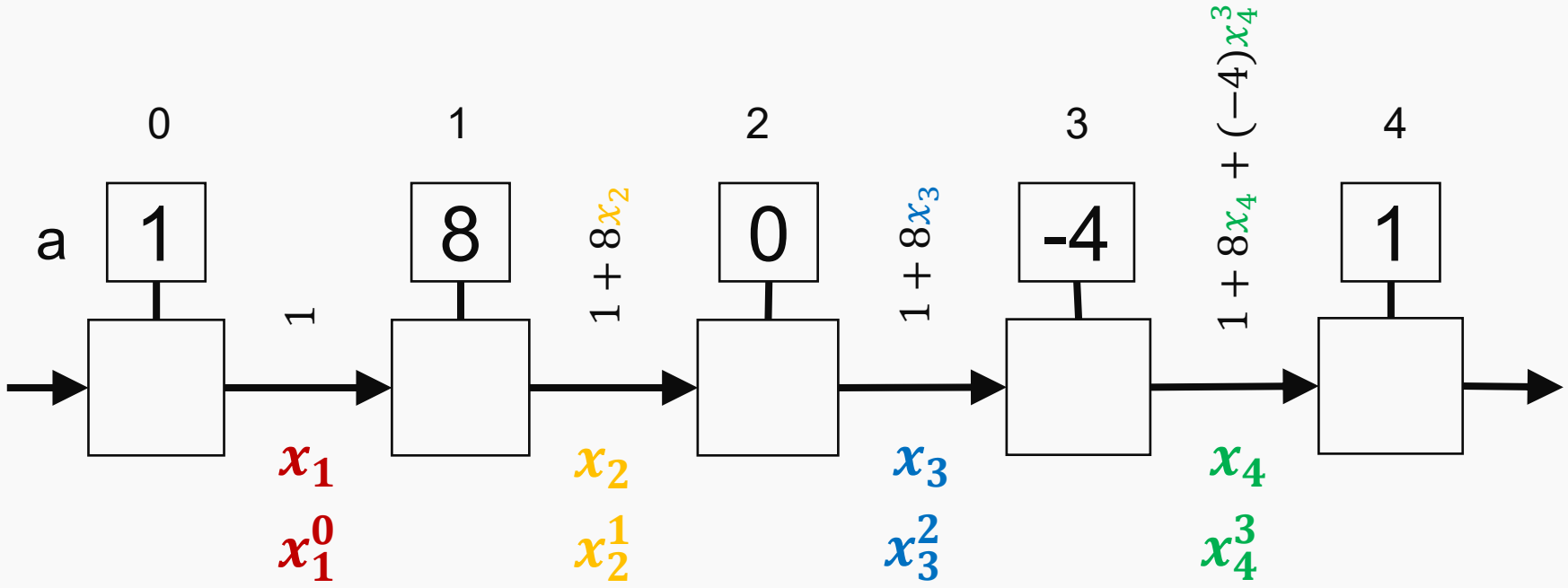
Calcul unui polinom cu pipeline



Calcul unui polinom cu pipeline

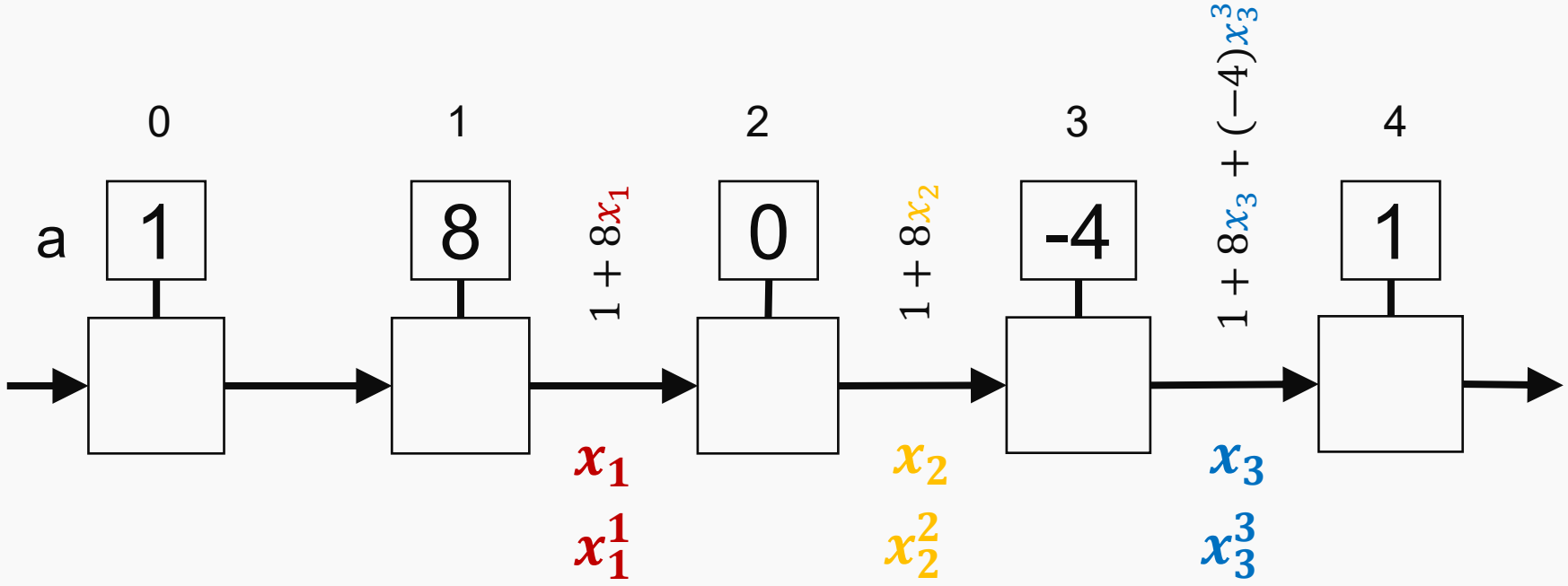


Calcul unui polinom cu pipeline



$$1+8x_5+(-4)x_5^3+x_5^4$$

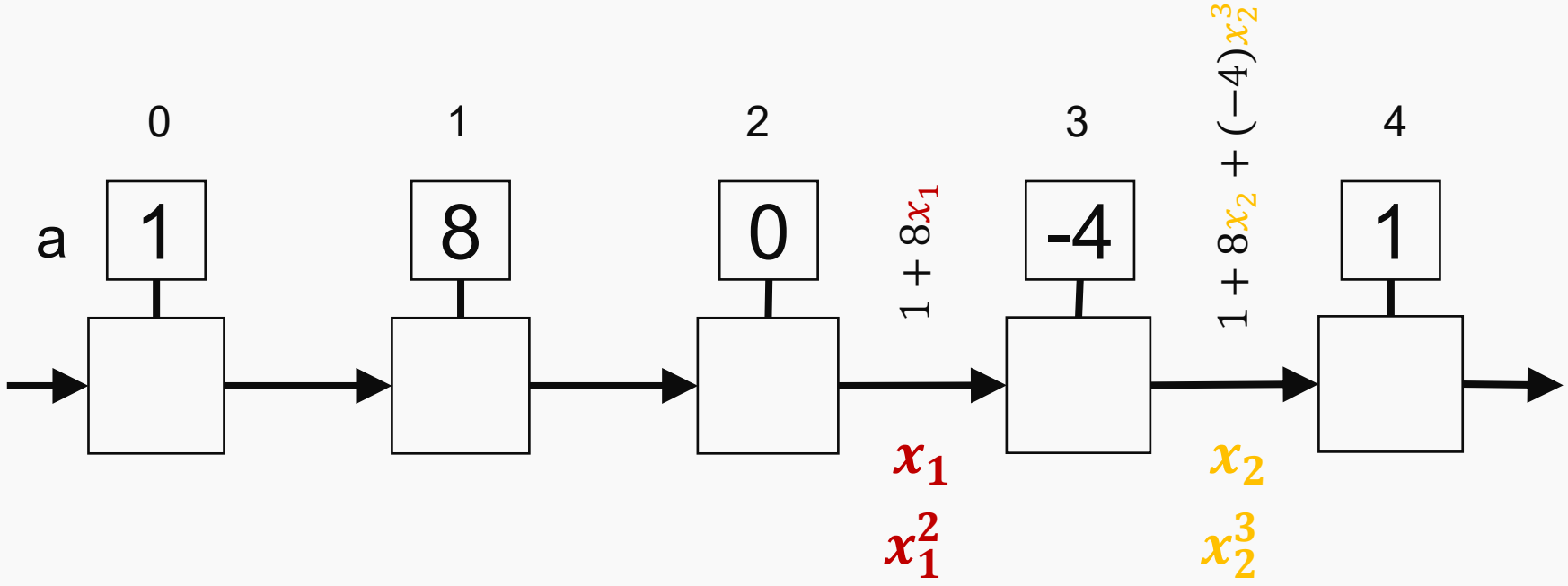
Calcul unui polinom cu pipeline



$$1+8x_4+(-4)x_4^3+x_4^4$$

$$1+8x_5+(-4)x_5^3+x_5^4$$

Calcul unui polinom cu pipeline

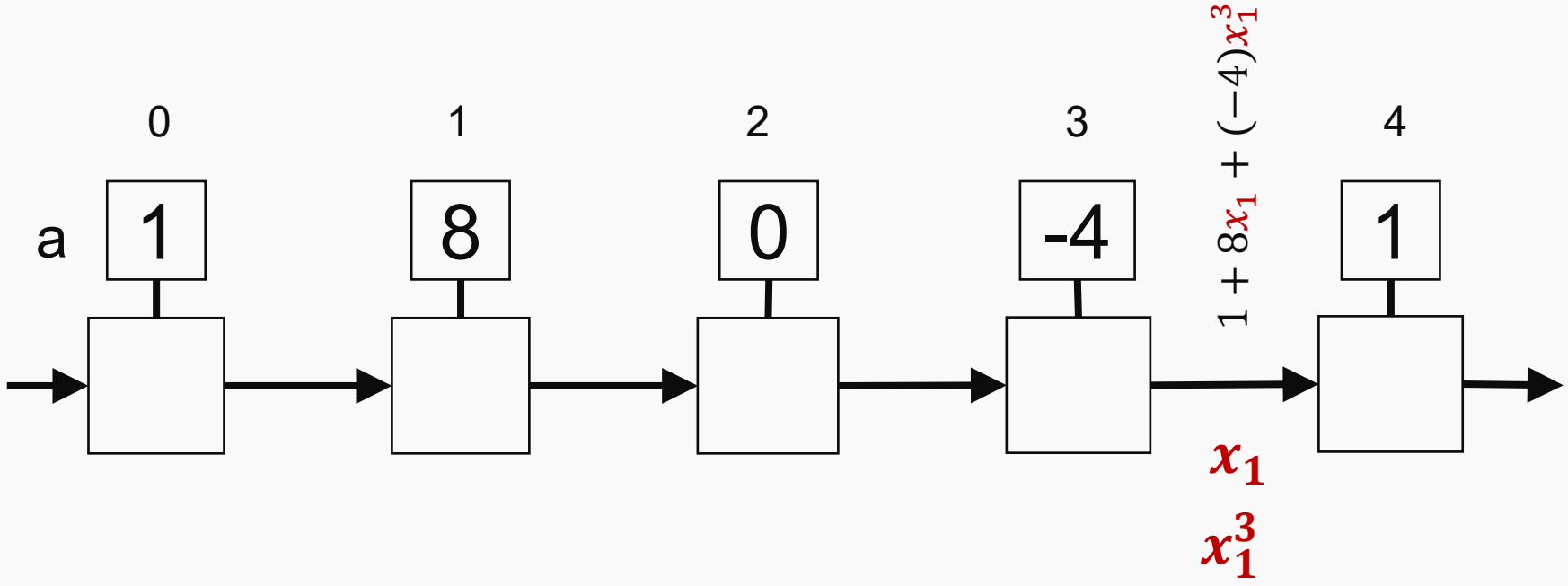


$$1+8x_3+(-4)x_3^3+x_3^4$$

$$1+8x_4+(-4)x_4^3+x_4^4$$

$$1+8x_5+(-4)x_5^3+x_5^4$$

Calcul unui polinom cu pipeline



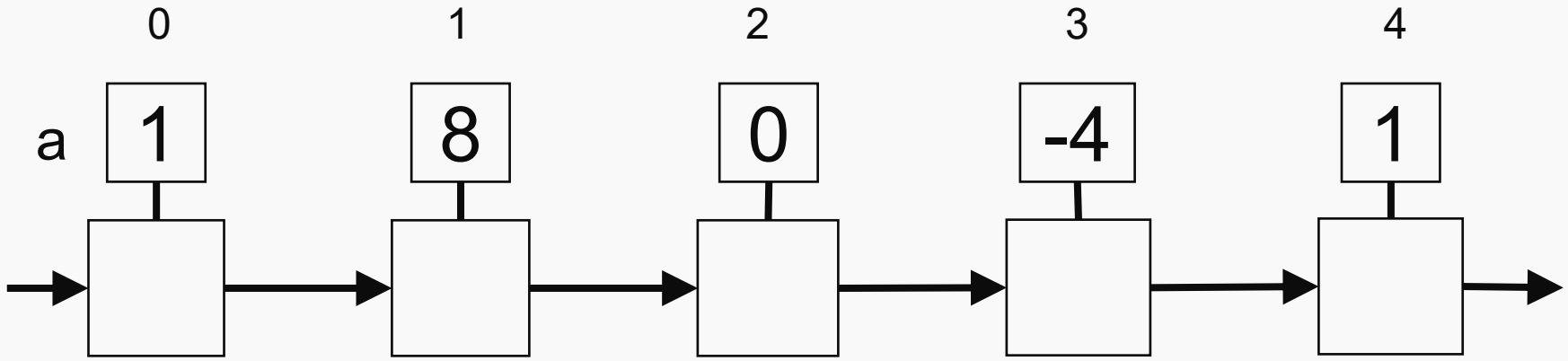
$1 + 8x_2 + (-4)x_2^3 + x_2^4$

$1 + 8x_3 + (-4)x_3^3 + x_3^4$

$1 + 8x_4 + (-4)x_4^3 + x_4^4$

$1 + 8x_5 + (-4)x_5^3 + x_5^4$

Calcul unui polinom cu pipeline



$$1 + 8x_1 + (-4)x_1^3 + x_1^4$$

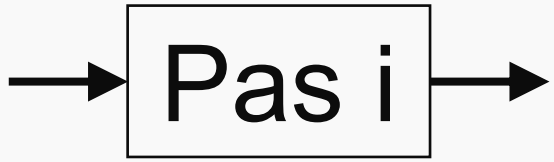
$$1 + 8x_2 + (-4)x_2^3 + x_2^4$$

$$1 + 8x_3 + (-4)x_3^3 + x_3^4$$

$$1 + 8x_4 + (-4)x_4^3 + x_4^4$$

$$1 + 8x_5 + (-4)x_5^3 + x_5^4$$

Pipeline – ghid programare



Inițializare

for(un număr de pași) {

    primește date de la Pas(i-1)

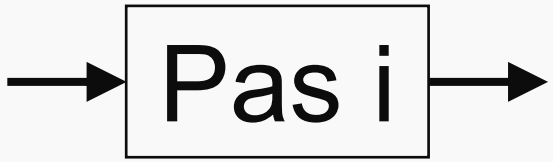
    procesează

    trimite date la Pas(i+1)

}

Finalizare

Pipeline – ghid programare



Primește coeficientul potrivit pasului

for(un număr de pași egal cu numărul de valori) {

primește de la Pas(i-1): polinom parțial calculat

valoarea originală

valoare originală ^ (i-1)

Calculează (valoarea originală ^ i) și adaugă la polinom parțial calculat produsul dintre coeficient și aceasta.

trimite spre Pas(i+1): noul polinom parțial calculat, valoarea originală și valoare originală ^ i)

}