

# Architektura systemów komputerowych

---

Mariusz Wiśniewski

---

Politechnika Świętokrzyska w Kielcach  
Katedra Informatyki

# Przetwarzanie potokowe I



## Plan wykładu

1. Praca potokowa.
2. Projekt  $\mu P$  – koncepcja potoku:
  - 2.1. model ścieżki danych
  - 2.2. rejestry w potoku,
  - 2.3. wykonanie instrukcji,
  - 2.3. program w potoku.

## Cele

Poznanie pojęcia pracy potokowej. Poznanie techniki projektowania procesora z przetwarzaniem potokowym.

# Praca potokowa





## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

Czynności w produkcji samochodów:

1 godzina



1 godzina



1 godzina



1 godzina

Wydajność produkcyjna:

- 1 samochód w ciągu 4 godzin,
- 6 samochodów na dzień,
- 180 samochodów na miesiąc,
- 2160 samochodów na rok.

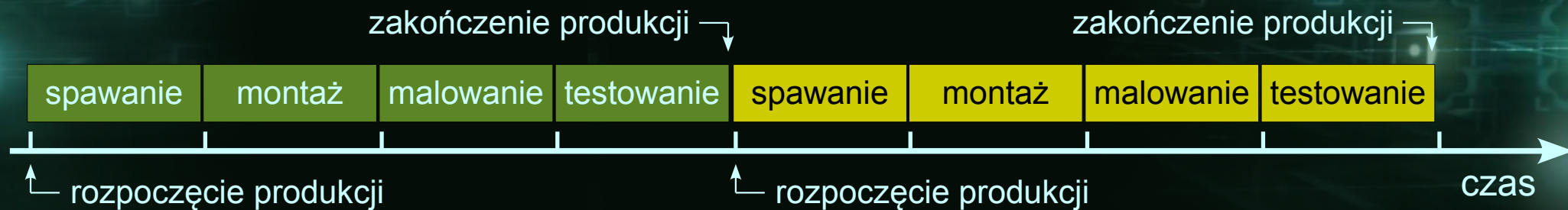
## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

Czynności w produkcji samochodów:

Wydajność produkcyjna:

- 1 samochód w ciągu 4 godzin,
- 6 samochodów na dzień,
- 180 samochodów na miesiąc,
- 2160 samochodów na rok.





## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

Taśma produkcyjna samochodów:

dostawa części

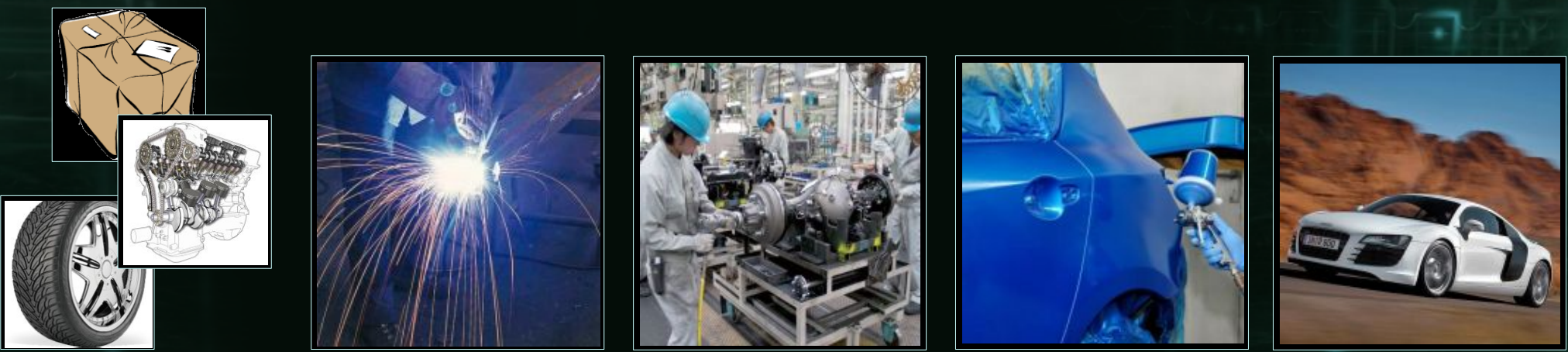
spawanie  
1 godzinamontaż  
1 godzinamalowanie  
1 godzinatestowanie  
1 godzinaWydajność taśmy produkcyjnej:

- pierwszy samochód jest składany w ciągu 4-ch godzin ← **opóźnienie potoku**
- następne samochody powstają co 1 godzinę,
  - 21 samochodów powstanie pierwszego dnia produkcji,
  - 24 samochody powstaną każdego następnego dnia,
- w miesiącu fabryka wyprodukuje 717 samochodów,
- rocznie powstanie 8604 samochody.

# Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

## Taśma produkcyjna samochodów:



dostawa części → spawanie → montaż → malowanie → testowanie





## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Praca potokowa to cykl połączonych ze sobą operacji, w których wyniki poprzedniej stają się argumentami dla następnej.

Pojęcia:

- rodzaje potoków
- specyfika pracy potoku
- implementacje

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

– wstęp

Praca potokowa to cykl połączonych ze sobą operacji, w których wyniki poprzedniej stają się argumentami dla następnej.

– wydajność przetwarzania

Pojęcia:

– potok instrukcji

- rodzaje potoków
- specyfika pracy potoku
- implementacje

W odniesieniu do komputerów pojęcie potoku powiązane jest z:

- procesorem – **potok instrukcyjny** – wymaga podziału pracy CPU na etapy, w których wynik wykonania rozkazu zostaje skompletowany po wykonaniu wszystkich etapów, związanych z daną instrukcją,
- układem **GPU** – spotykane w większości procesorów graficznych, składający się z wielu bloków funkcjonalnych lub całych **procesorów specjalizowanych**. Potok graficzny jest zoptymalizowany pod kontem wykonywania operacji graficznych, np.: operacji teksturowania, manipulacji światłami oraz kolorami, odwzorowania perspektywy itp.,
- oprogramowaniem – jest funkcją systemu operacyjnego, umożliwiając komunikację (wymianę danych) między uruchomionymi aplikacjami. Przykładem jest **potok nazwany** lub **nienazwany** systemu UNIX/Linux.

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Praca potokowa to cykl połączonych ze sobą operacji, w których wyniki poprzedniej stają się argumentami dla następnej.

Pojęcia:

- rodzaje potoków
- specyfika pracy potoku
- implementacje

W odniesieniu do komputerów pojęcie potoku powiązane jest z:

- procesorem – potok instrukcyjny – wymaga podziału pracy CPU na etapy, w których wynik wykonania rozkazu zostaje skompletowany po wykonaniu wszystkich etapów, związanych z daną instrukcją,
- układem GPU – spotykane w większości procesorów graficznych, składający się z wielu bloków funkcjonalnych lub całych procesorów specjalizowanych. Potok graficzny jest zoptymalizowany pod kątem wykonywania operacji graficznych, np.: operacji teksturowania, manipulacji światłami oraz kolorami, odwzorowania perspektywy itp.,
- oprogramowaniem – jest funkcją systemu operacyjnego, umożliwiając komunikację (wymianę danych) między uruchomionymi aplikacjami. Przykładem jest potok nazwany lub nienazwany systemu UNIX/Linux.

Zastosowanie potoku nie powoduje zmniejszenia czasu potrzebnego na wykonanie czynności, natomiast zwiększa znacznie **wydajność systemu**.

Jednocześnie obecność potoku powoduje:

- powstawanie **opóźnień w propagacji sygnałów** (wyników),
- zwiększone zapotrzebowanie na komponenty służące do budowy systemu, takie jak: jednostki wykonawcze, moduły pamiętające, bufory i magistrale, itp..



## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Praca potokowa to cykl połączonych ze sobą operacji, w których wyniki poprzedniej stają się argumentami dla następnej.

Pojęcia:

- rodzaje potoków
- specyfika pracy potoku
- implementacje

W odniesieniu do komputerów pojęcie potoku powiązane jest z:

- procesorem – potok instrukcyjny – wymaga podziału pracy CPU na etapy, w których wynik wykonania rozkazu zostaje skompletowany po wykonaniu wszystkich etapów, związanych z daną instrukcją,
- układem GPU – spotykane w większości procesorów graficznych, składający się z wielu bloków funkcjonalnych lub całych procesorów specjalizowanych. Potok graficzny jest zoptymalizowany pod kątem wykonywania operacji graficznych, np.: operacji teksturowania, manipulacji światłami oraz kolorami, odwzorowania perspektywy itp.,
- oprogramowaniem – jest funkcją systemu operacyjnego, umożliwiając komunikację (wymianę danych) między uruchomionymi aplikacjami. Przykładem jest potok nazwany lub nienazwany systemu UNIX/Linux.

Istnieją trzy główne metody implementacji potoku:

- potok rejestrowy **synchroniczny**,
- potok rejestrowy **asynchroniczny**,
- potok niebuforowany.

Dla rozwiązań niebuforowanych proces sterowania jest tak zaprojektowany, że przekazywanie danych następuje tylko podczas ich stabilności. W każdym przypadku **szybkość pracy potoku zależy od jego najwolniejszego stopnia**.

Zastosowanie potoku nie powoduje zmniejszenia czasu potrzebnego na wykonanie czynności, natomiast zwiększa znacznie wydajność systemu.

Jednocześnie obecność potoku powoduje:

- powstawanie opóźnień w propagacji sygnałów (wyników),
- zwiększone zapotrzebowanie na komponenty służące do budowy systemu, takie jak: jednostki wykonawcze, moduły pamiętające, bufora i magistrale, itp..

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Wydajność systemu można przedstawić jako liczbę produkcji w jednostce czasu. Wydajność jest zależna architektury danego systemu.

Przypadki:

- architektura liniowa
- architektura potokowa

## Praca potokowa

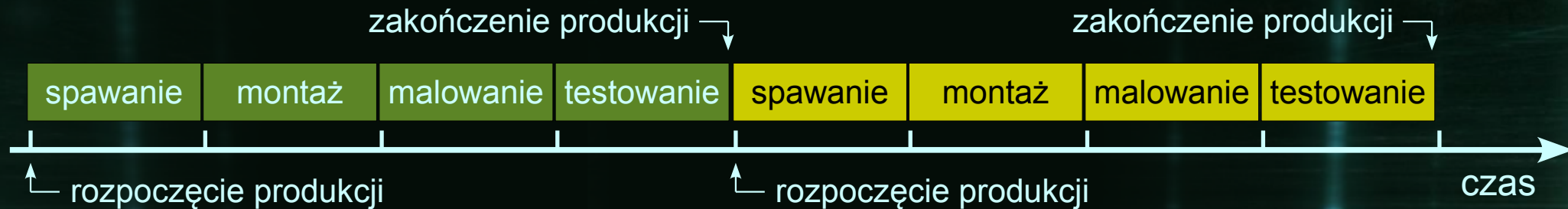
wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Wydajność systemu można przedstawić jako liczbę produkcji w jednostce czasu. Wydajność jest zależna architektury danego systemu.

Przypadki:

- architektura liniowa
- architektura potokowa



- czas wykonania jednego zadania =  $n$  jednostek czasu,
- czas wykonania jednego podzadania  $\sim 1$  jednostka czasu,
- wydajność systemu =  $\frac{1}{n}$  zadań w jednostce czasu.



## Praca potokowa

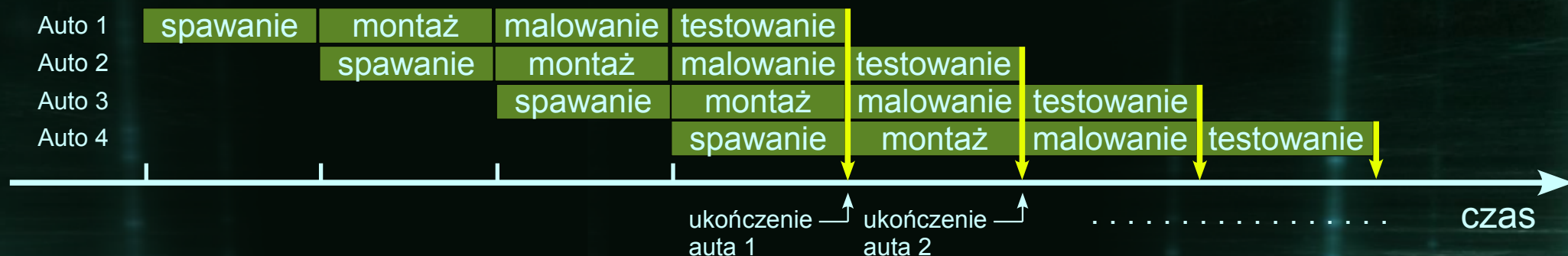
wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Wydajność systemu można przedstawić jako liczbę produkcji w jednostce czasu. Wydajność jest zależna architektury danego systemu.

Przypadki:

- architektura liniowa
- architektura potokowa



- czas wykonania pierwszego zadania =  $n$  jednostek czasu (opóźnienie),
- liczba zadań wykonanych w czasie  $T = T - n + 1$

- wydajność systemu =  $\frac{1 - (n - 1)}{T}$  zadań w jednostce czasu.

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Wydajność systemu można przedstawić jako liczbę produkcji w jednostce czasu. Wydajność jest zależna architektury danego systemu.

Przypadki:

- architektura liniowa
- architektura potokowa

Współczynnik zwiększenia wydajności:

$$\frac{\text{wydajność architektury potokowej}}{\text{wydajność architektury liniowej}} = \frac{\frac{1 - (n - 1)}{T}}{\frac{1}{n}} = n - \frac{n(n - 1)}{T} \rightarrow n \text{ dla } T \rightarrow \infty$$

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Potok instrukcji zwiększa wydajność procesora w zakresie liczby wykonywanych instrukcji w jednostce czasu.

Pojęcia:

- koncepcja
- stopnie potoku



## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Potok instrukcji zwiększa wydajność procesora w zakresie liczby wykonywanych instrukcji w jednostce czasu.

Pojęcia:

- koncepcja
- stopnie potoku

W procesorach najpowszechniej stosuje się potok instrukcji, którego etapy – nazywane **stopniami**, odpowiadają **fazom cyklu rozkazowego CPU**.

Procesor jest urządzeniem synchronicznym, rozpoczynającym każdą z czynności zgodnie z **taktami zegara systemowego**. Zazwyczaj koncepcja potoku zakłada rozpoczynanie akcji w każdym jego stopniu w tym samym momencie, kiedy zegar systemowy zmienia stan, generując kolejny takt. Architektura potoku zależy od różnych czynników, w ogólności:

- od modelu ISA procesora,
- od architektury CPU,
- uwzględnia ograniczenia zasobów wykonawczych procesora.

Nowoczesne procesory posiadają mocno rozbudowane potoki instrukcyjne – złożone z ponad 20 stopni, które realizują również **zadania optymalizacyjne** pracy potoku.

## Praca potokowa

wprowadzenie do zagadnienia przetwarzania potokowego

- wstęp
- wydajność przetwarzania
- potok instrukcji

Potok instrukcji zwiększa wydajność procesora w zakresie liczby wykonywanych instrukcji w jednostce czasu.

Pojęcia:

- koncepcja
- stopnie potoku

W procesorach najpowszechniej stosuje się potok instrukcji, którego etapy – nazywane stopniami, odpowiadają fazom cyklu rozkazowego CPU.

W podstawowej konfiguracji potok powinien zawierać następujące stopnie:

- **pobranie rozkazu:**
  - adres w RAM określa rejestr PC/IP,
  - pobranie rozkazu (**opcodu** i ewentualnie argumentu),
- **dekodowanie rozkazu** i pobranie wartości rejestrów:
  - wybranie rejestrów biorących udział w operacjach,
  - wskazanie dodatkowych argumentów,
- **pobranie argumentów** z pamięci (opcjonalnie):
  - obliczenie **adresu efektywnego** argumentu w RAM,
  - pobranie argumentu do rejestru tymczasowego,
- **wykonanie rozkazu:**
  - wykonanie obliczeń lub przekształceń,
  - wykonanie operacji na pamięci RAM,
  - wyznaczenie adresu następnej instrukcji,
- **zapisanie wyników:**
  - zapisanie do rejestrów procesora wyników obliczeń,
  - zapisanie adresu następnego rozkazu do rejestru PC/IP.

Procesor jest urządzeniem synchronicznym, rozpoczynającym każdą z czynności zgodnie z taktami zegara systemowego. Zazwyczaj koncepcja potoku zakłada rozpoczynanie akcji w każdym jego stopniu w tym samym momencie, kiedy zegar systemowy zmienia stan, generując kolejny takt. Architektura potoku zależy od różnych czynników, w ogólności:

- od modelu ISA procesora,
- od architektury CPU,
- uwzględnia ograniczenia zasobów wykonawczych procesora.

Nowoczesne procesory posiadają mocno rozbudowane potoki instrukcyjne – złożone z ponad 20 stopni, które realizują również zadania optymalizacyjne pracy potoku.

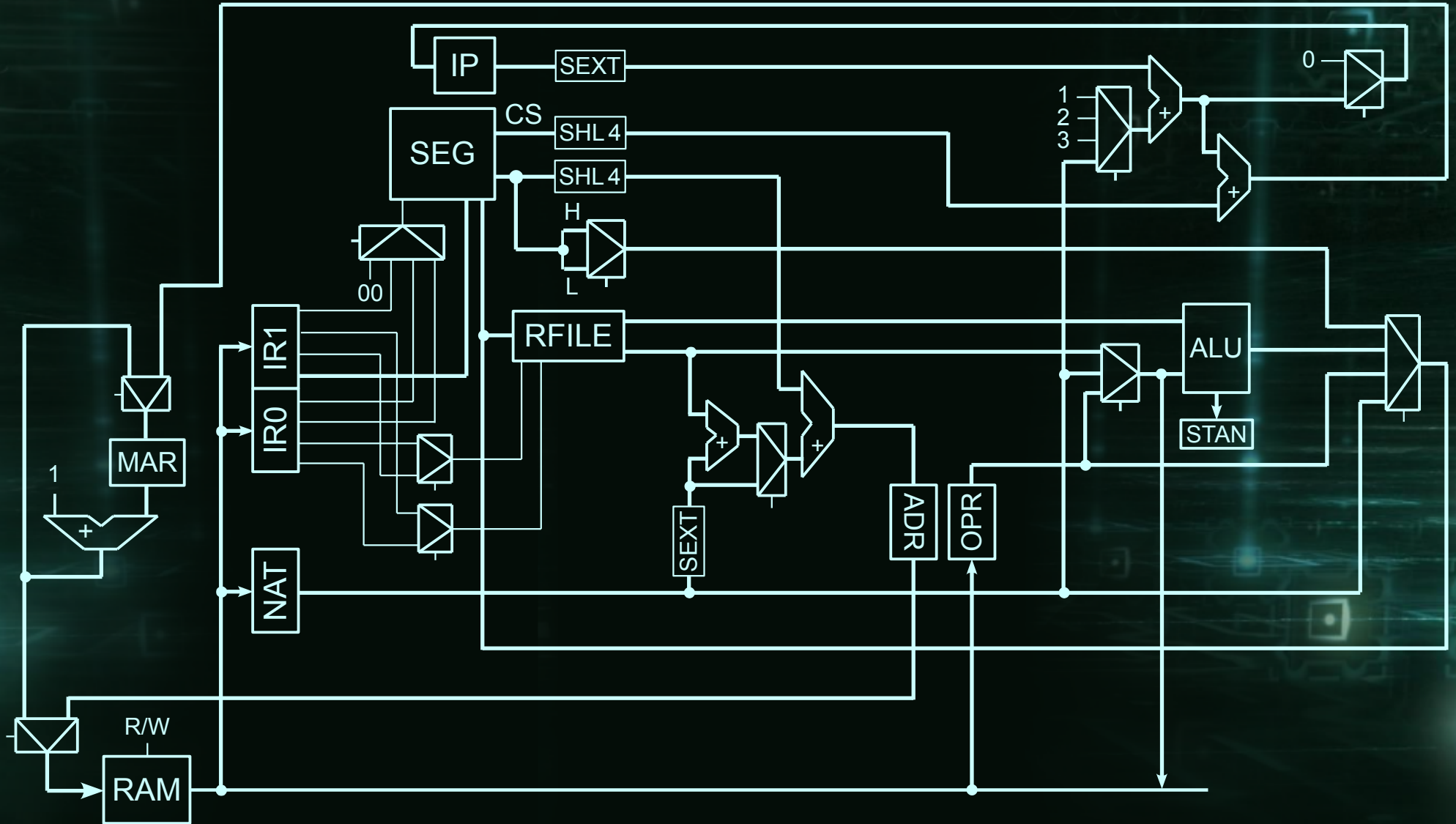
# Projekt $\mu P$ – koncepcja potoku: model ścieżki danych



# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU

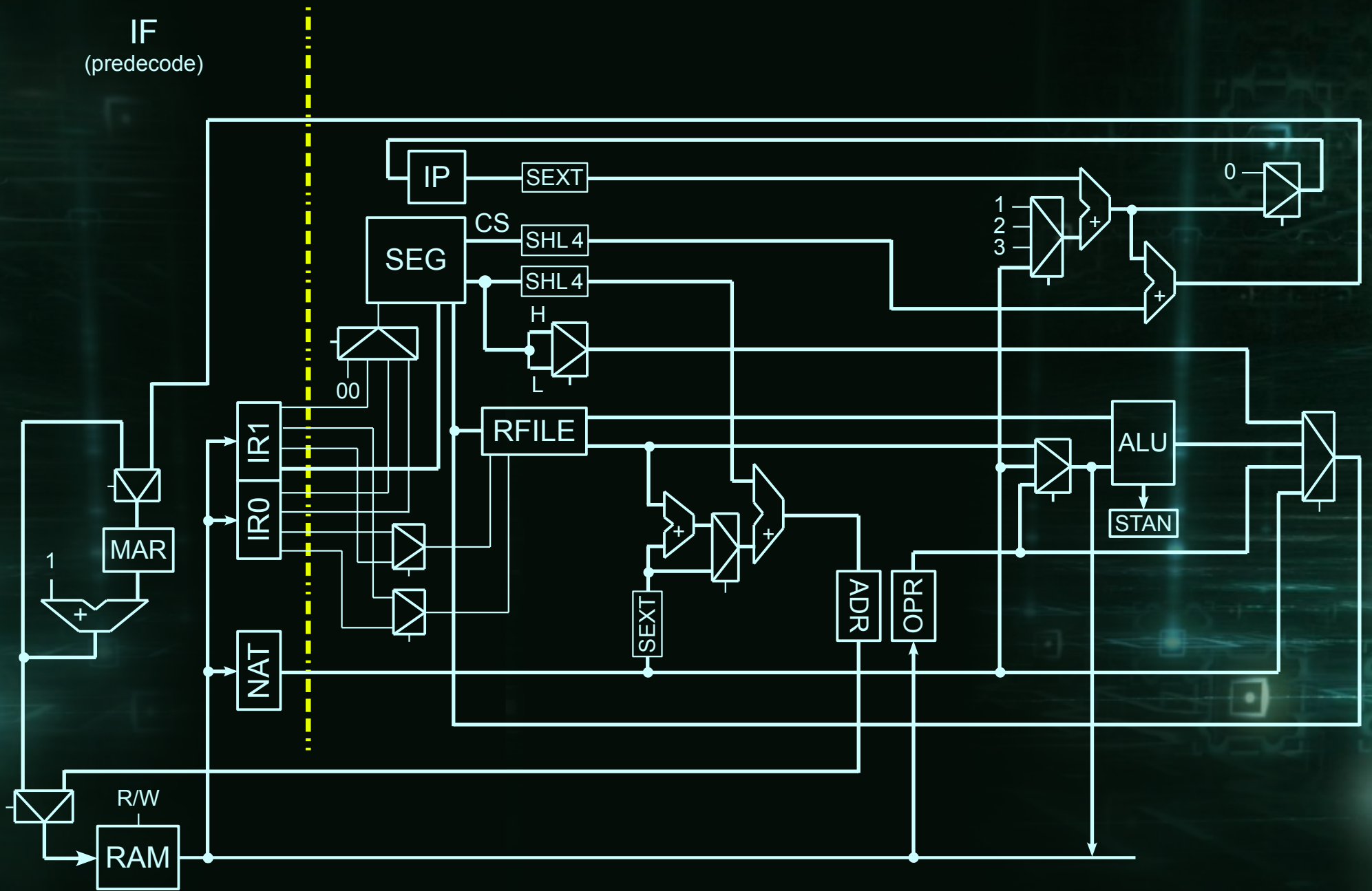
## Schemat blokowy ścieżki danych





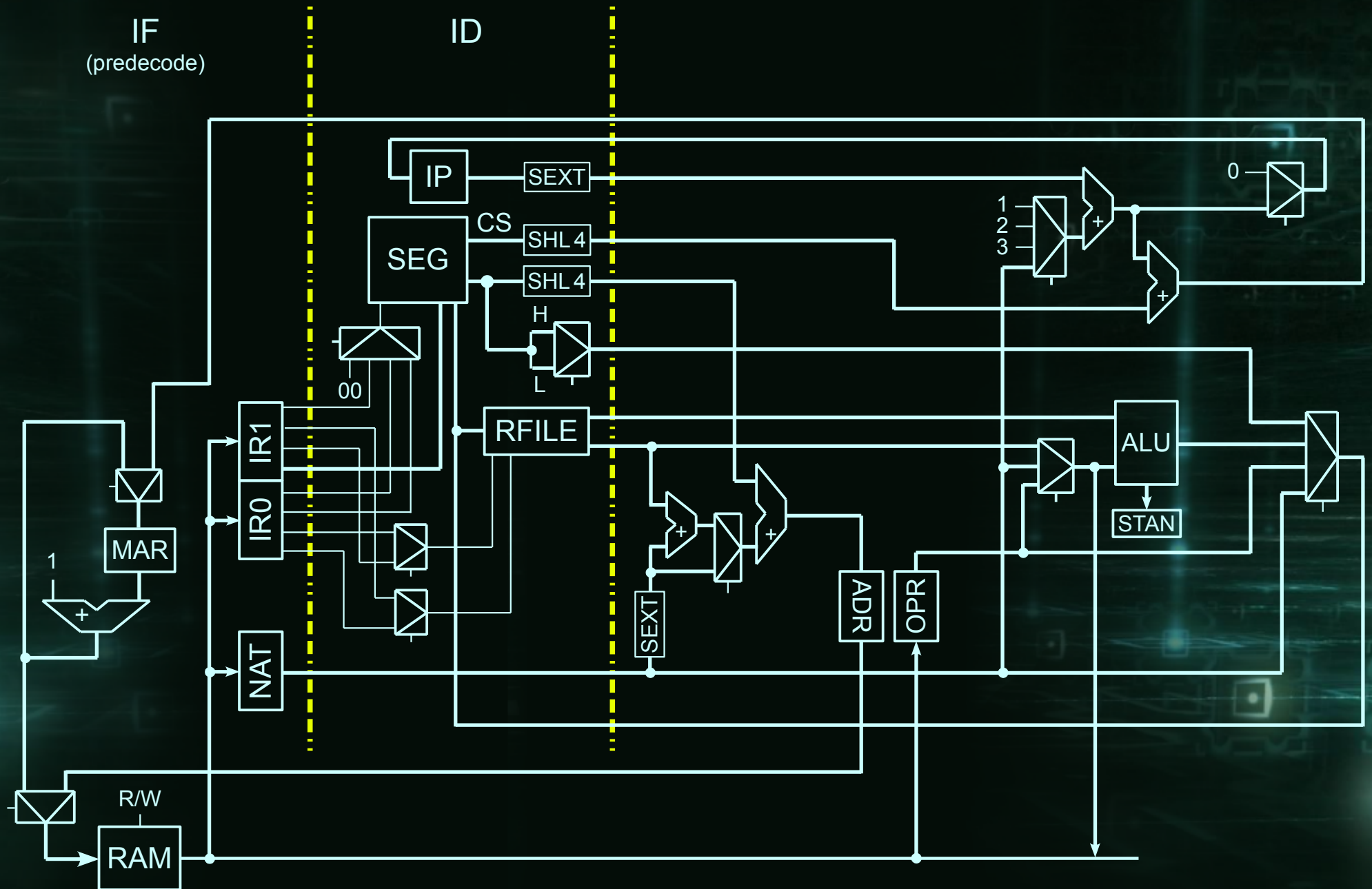
# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



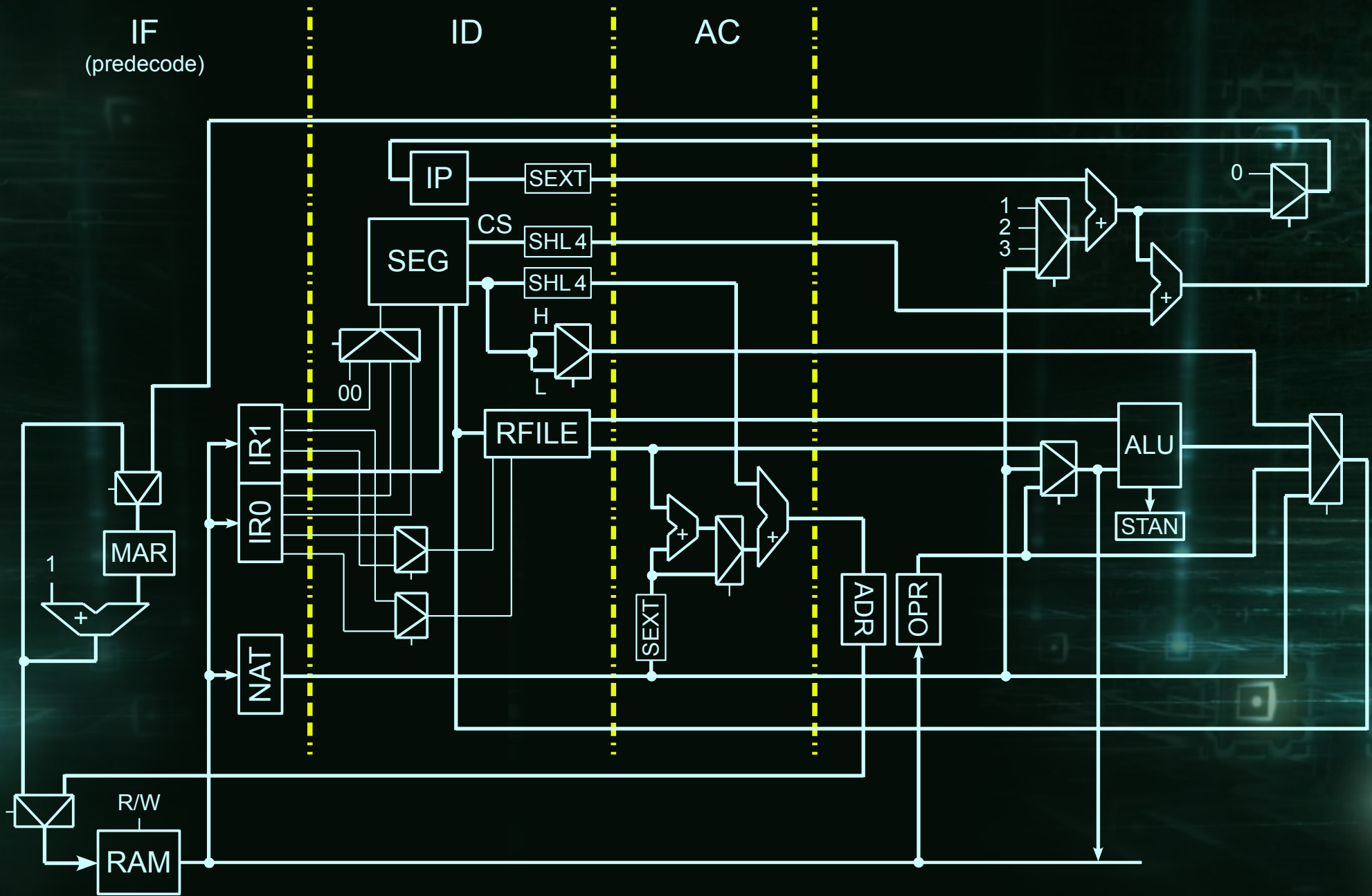
# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



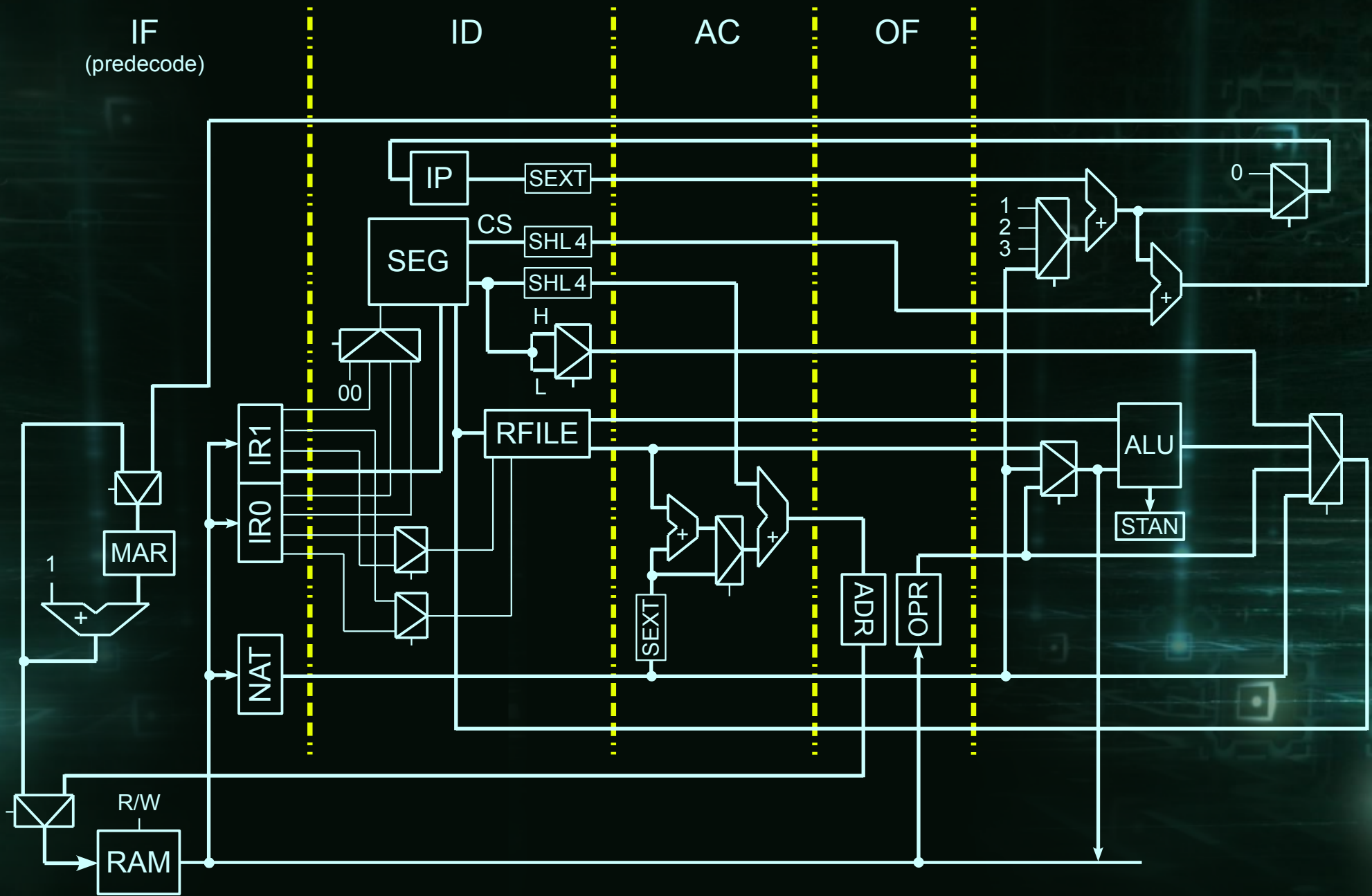
# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



# Projekt $\mu P$ – model ścieżki danych

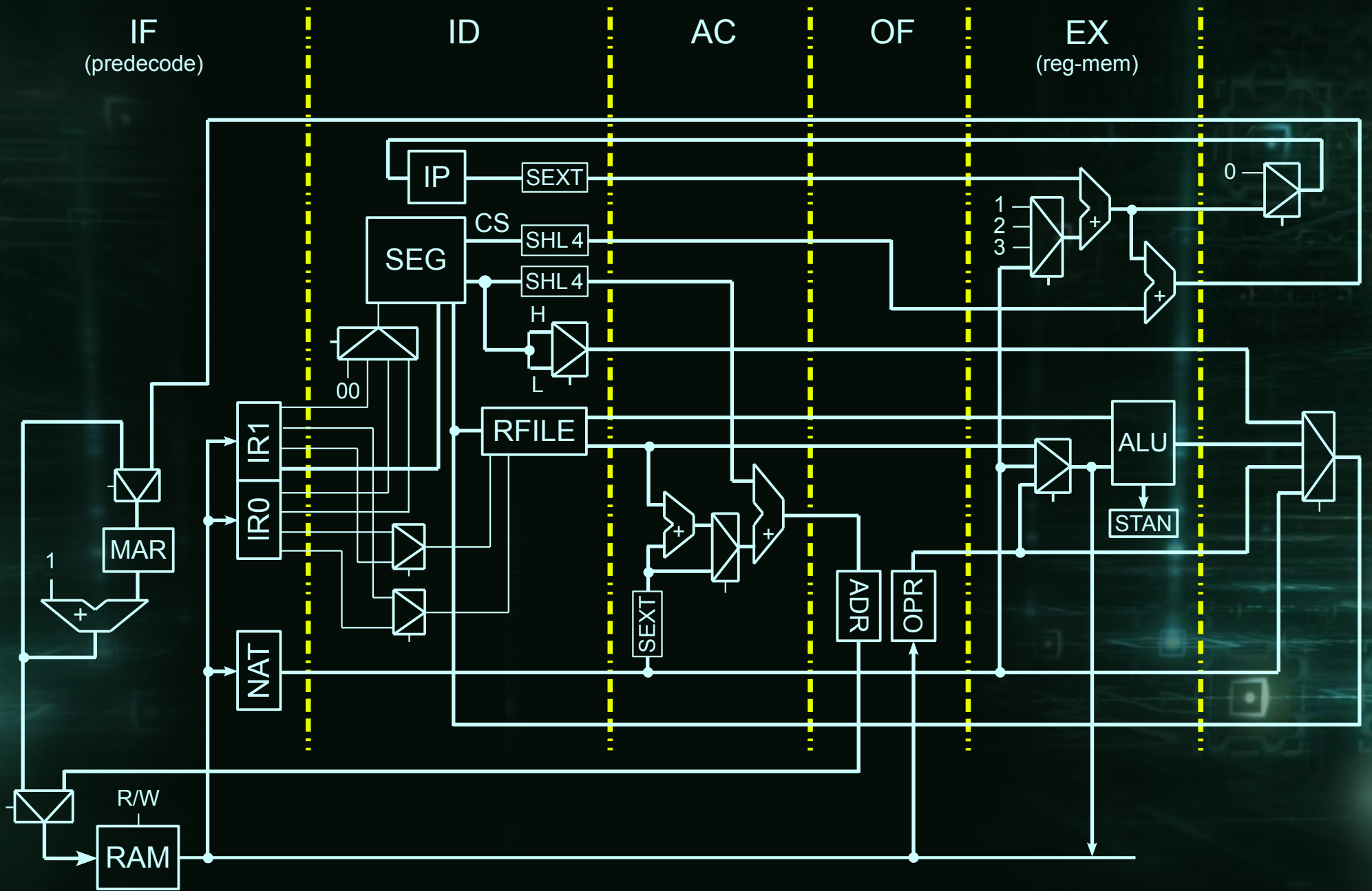
architektura potokowa projektowanego na wykładzie CPU





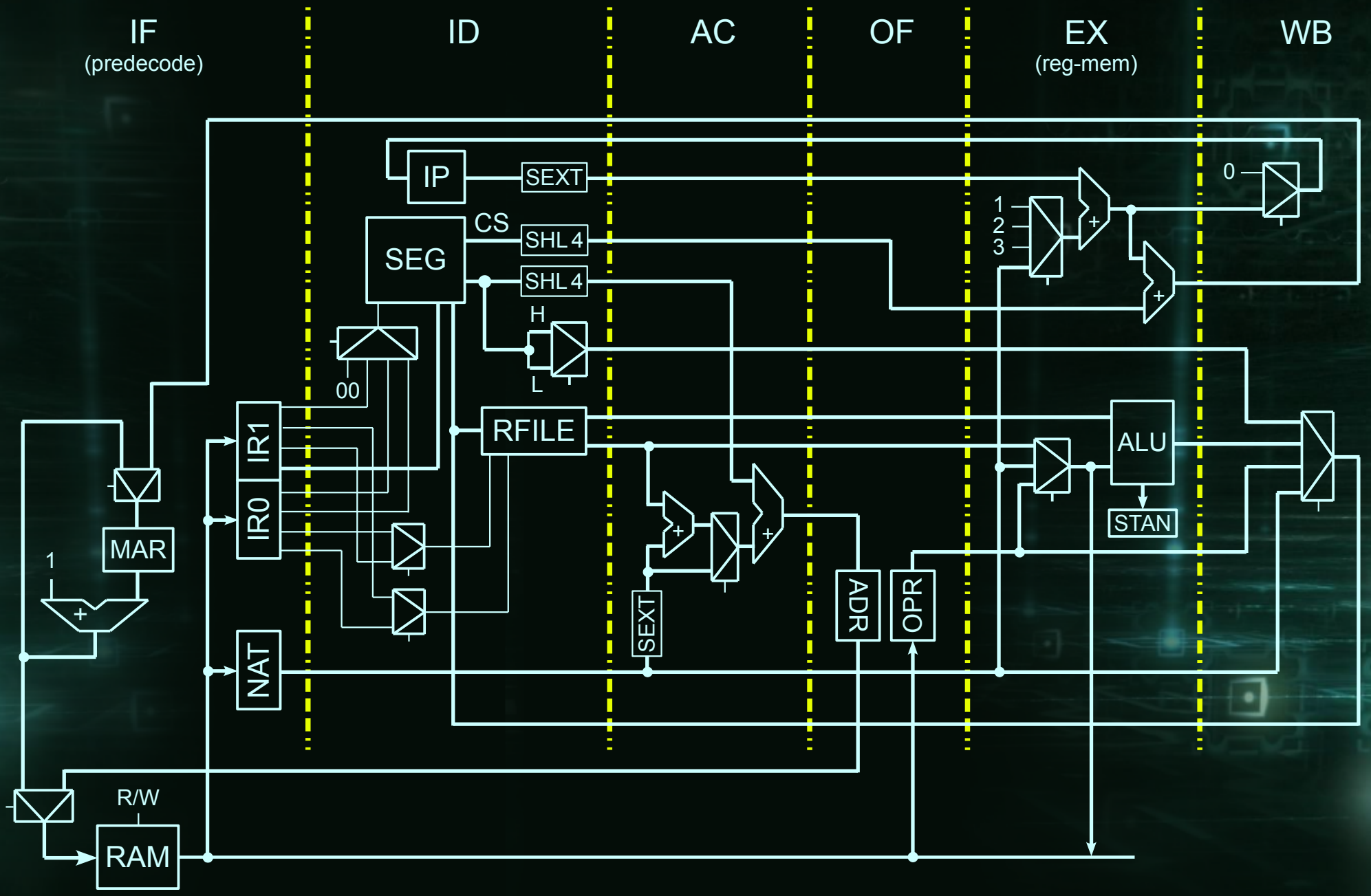
# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



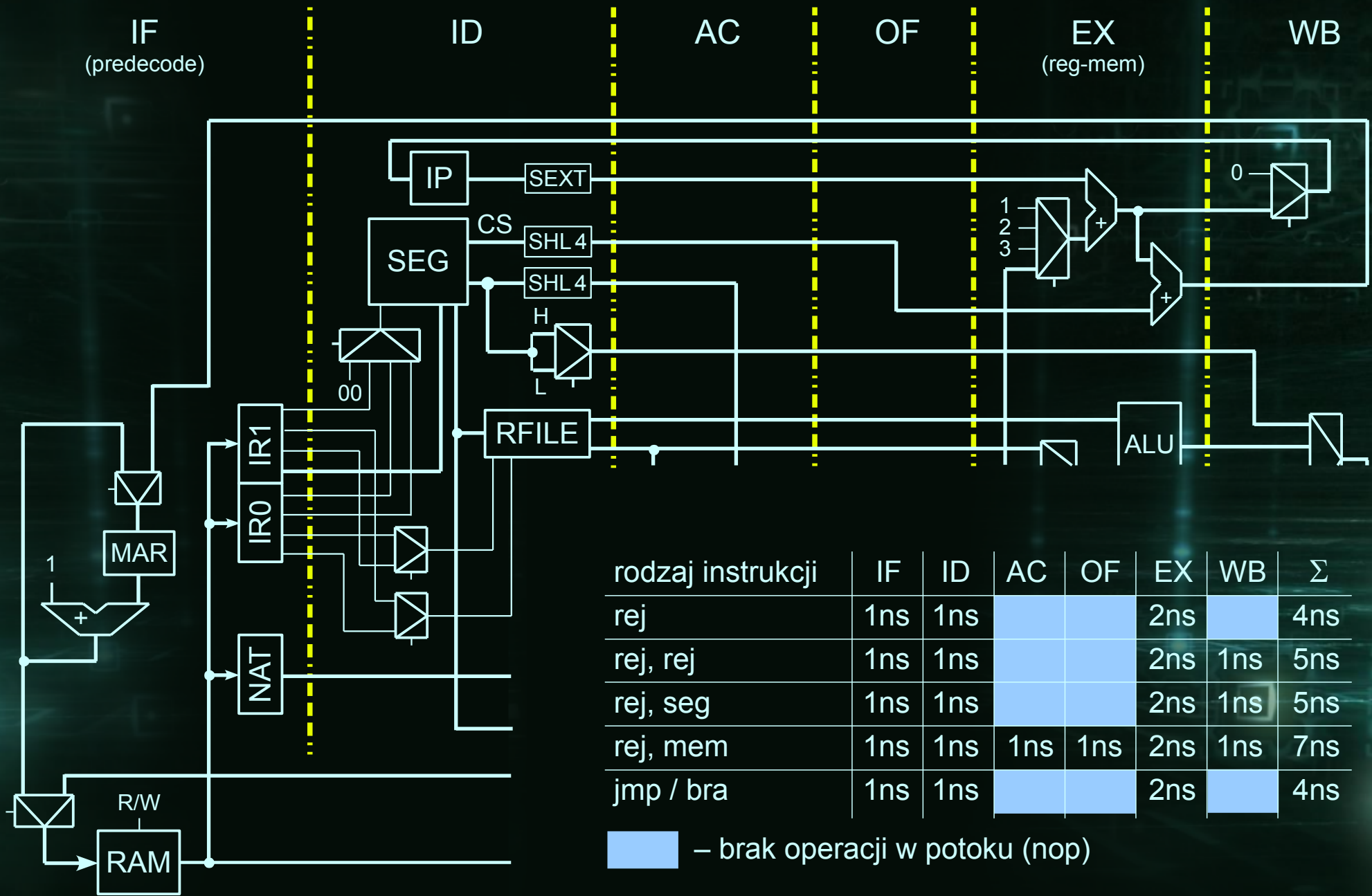
# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



# Projekt $\mu P$ – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU

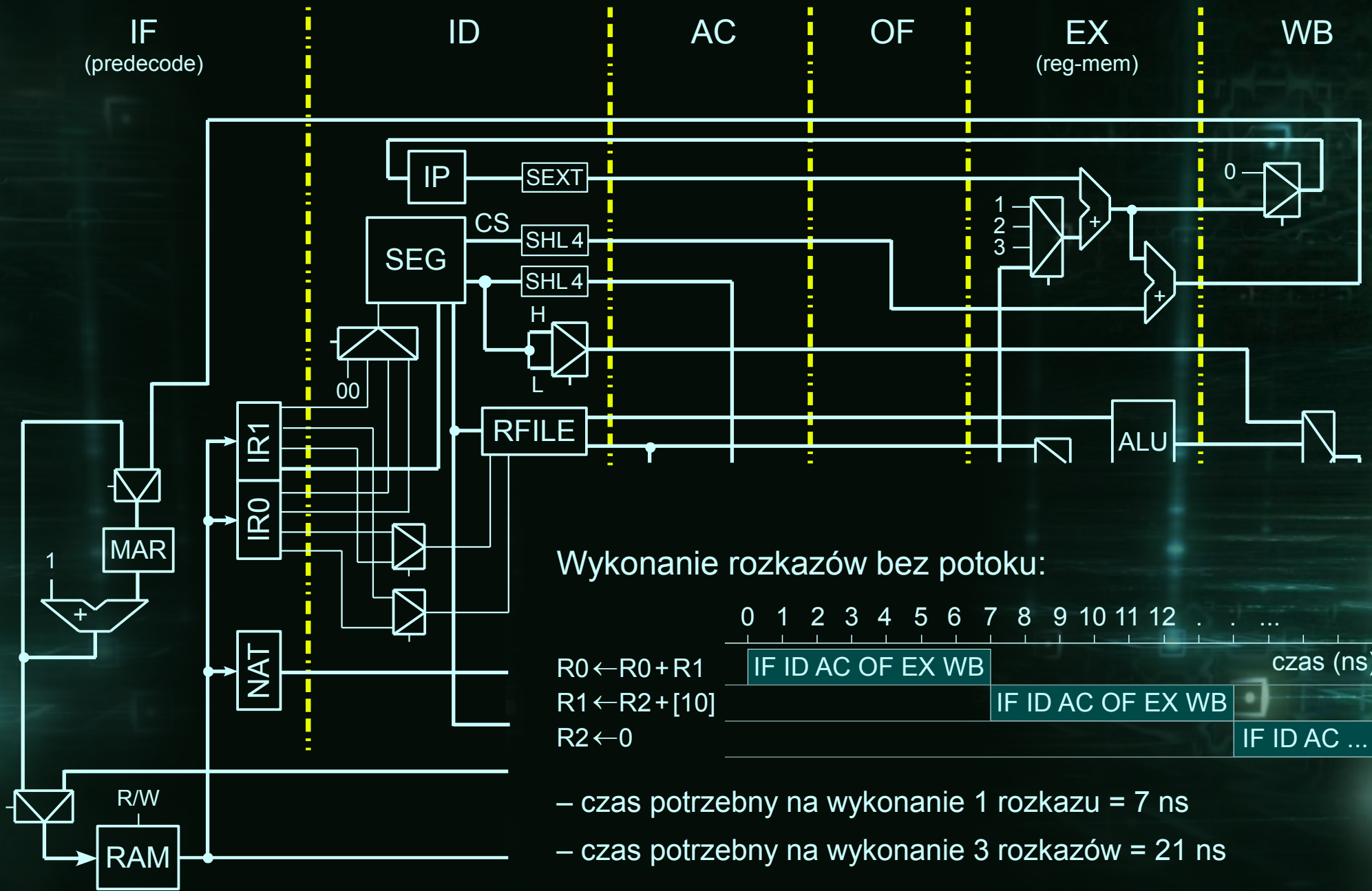


rodzaj instrukcji	IF	ID	AC	OF	EX	WB	$\Sigma$
rej	1ns	1ns			2ns		4ns
rej, rej	1ns	1ns			2ns	1ns	5ns
rej, seg	1ns	1ns			2ns	1ns	5ns
rej, mem	1ns	1ns	1ns	1ns	2ns	1ns	7ns
jmp / bra	1ns	1ns			2ns		4ns

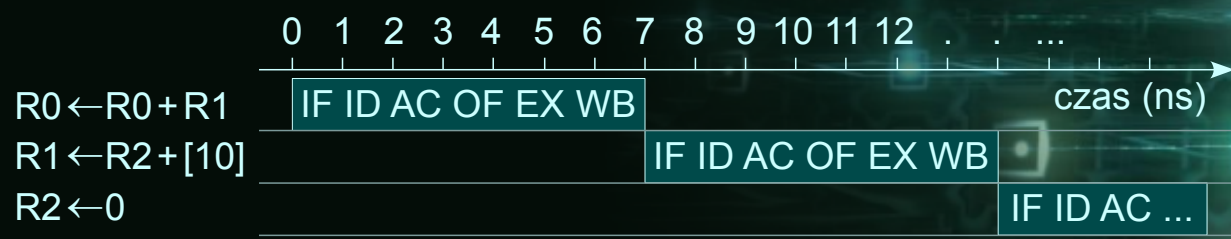
– brak operacji w potoku (nop)

Projekt  $\mu P$  – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



Wykonanie rozkazów bez potoku:

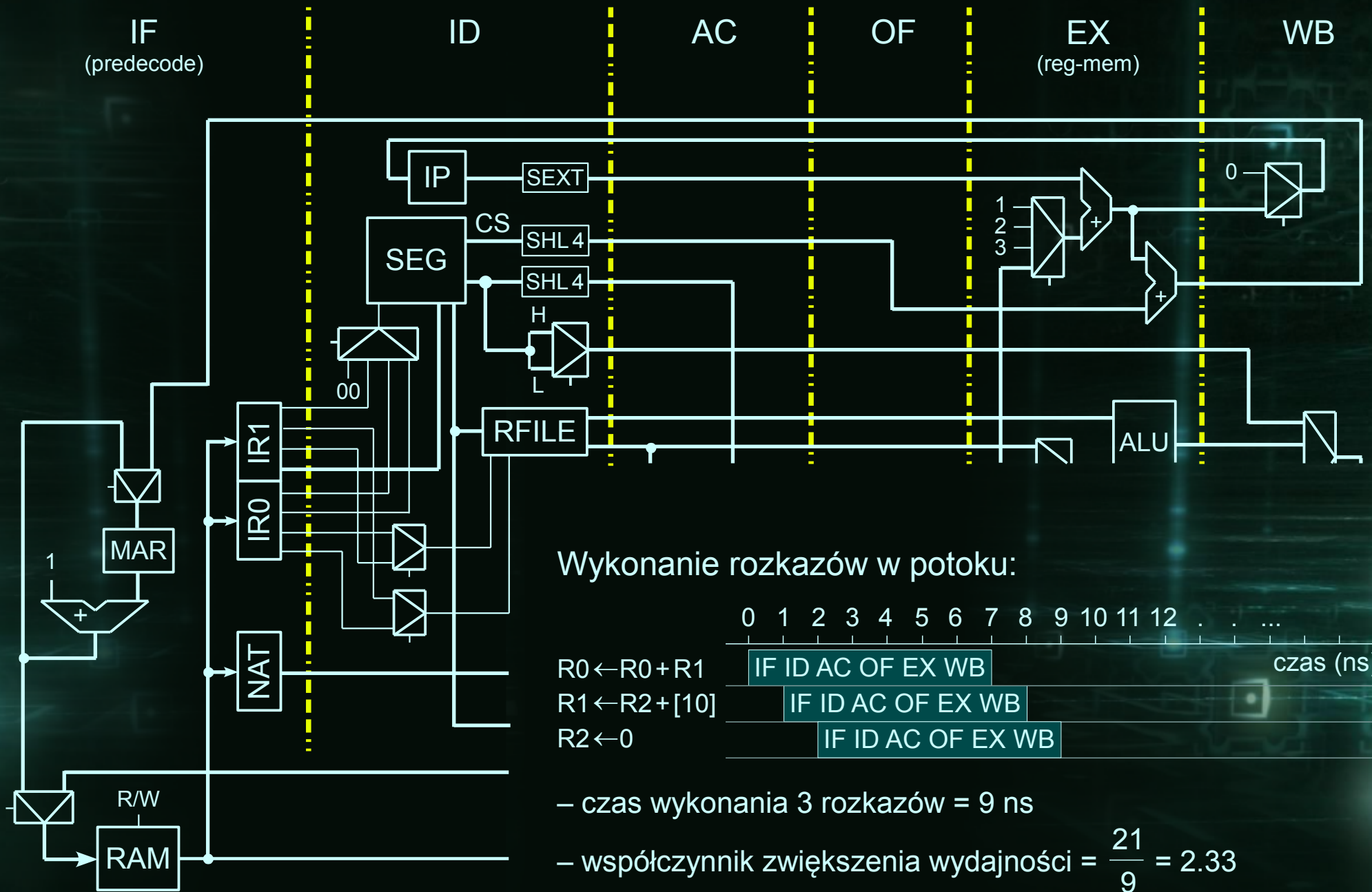


- czas potrzebny na wykonanie 1 rozkazu = 7 ns
- czas potrzebny na wykonanie 3 rozkazów = 21 ns



Projekt  $\mu P$  – model ścieżki danych

architektura potokowa projektowanego na wykładzie CPU



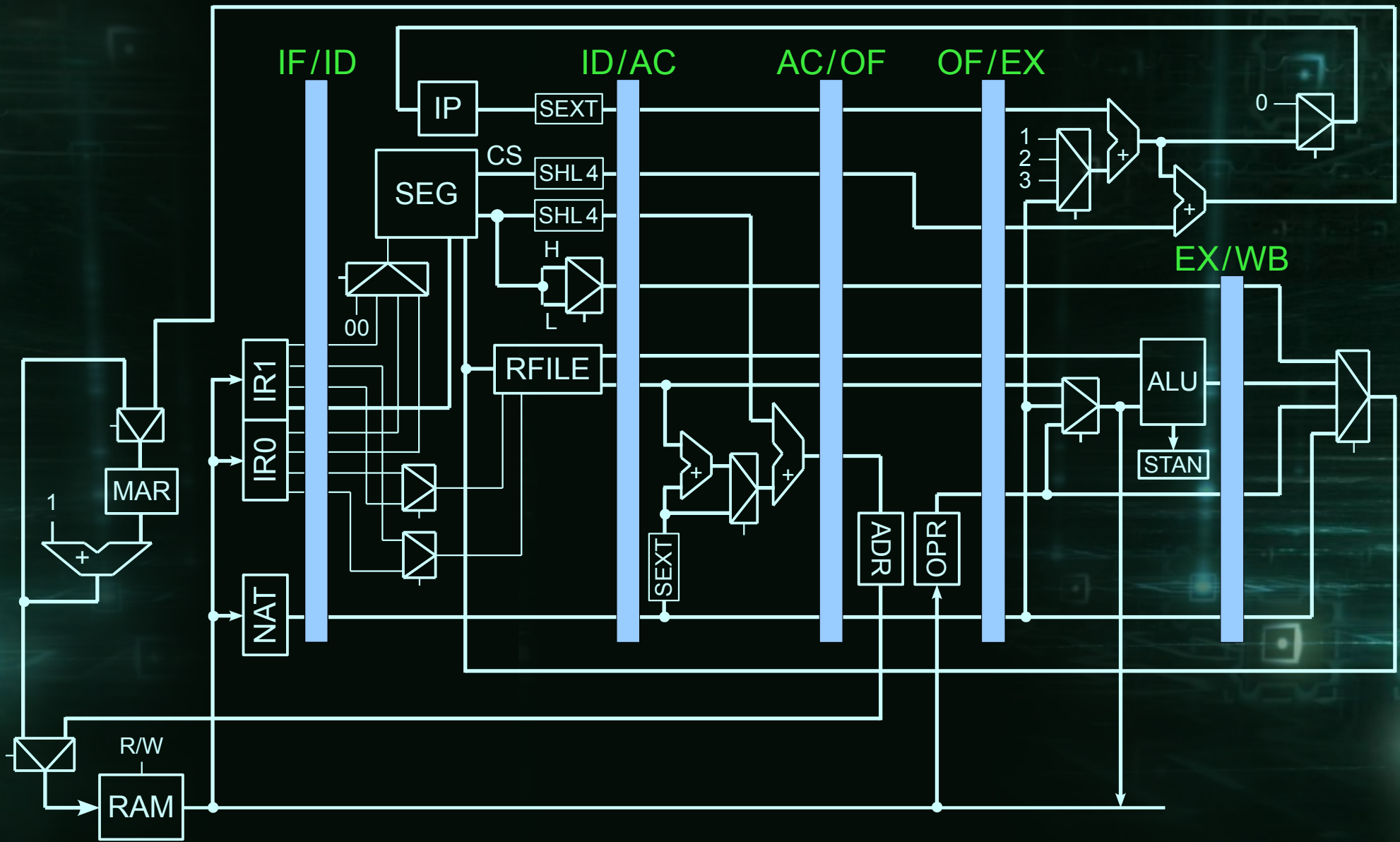
# Projekt $\mu P$ – koncepcja potoku: rejstry w potoku



# Projekt $\mu P$ – rejestry w potoku

rodzaj i rozmieszczenie rejestrów dodatkowych w potoku

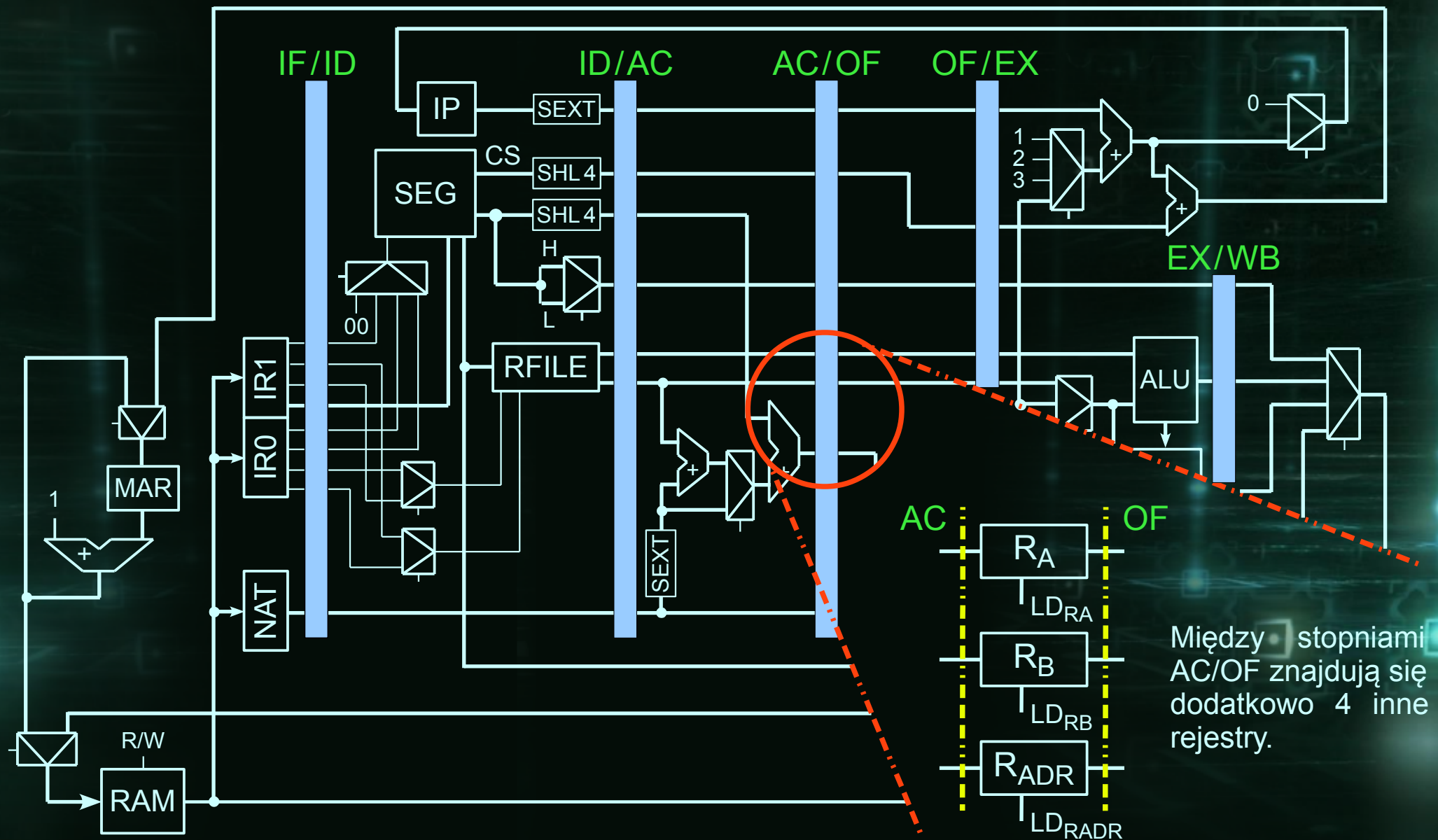
## Schemat ścieżki danych z rejestrami potoku



Projekt  $\mu P$  – rejestry w potoku

rodzaj i rozmieszczenie rejestrów dodatkowych w potoku

## Schemat ścieżki danych z rejestrami potoku





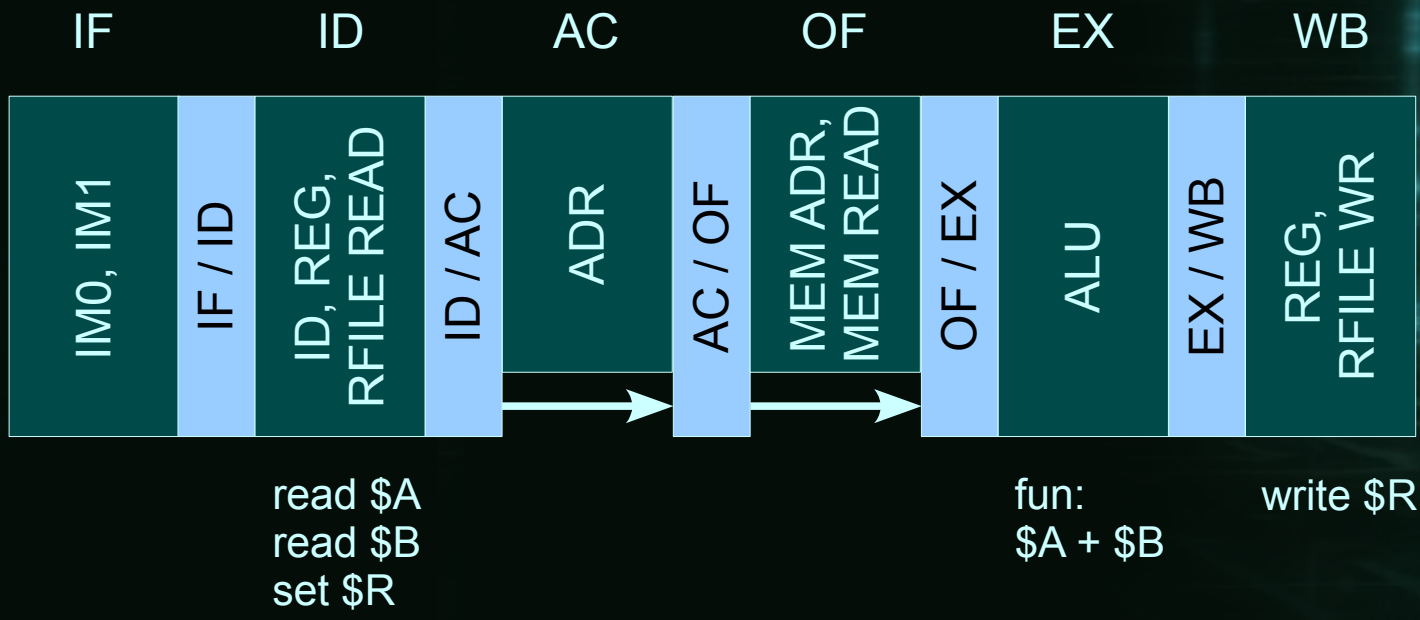
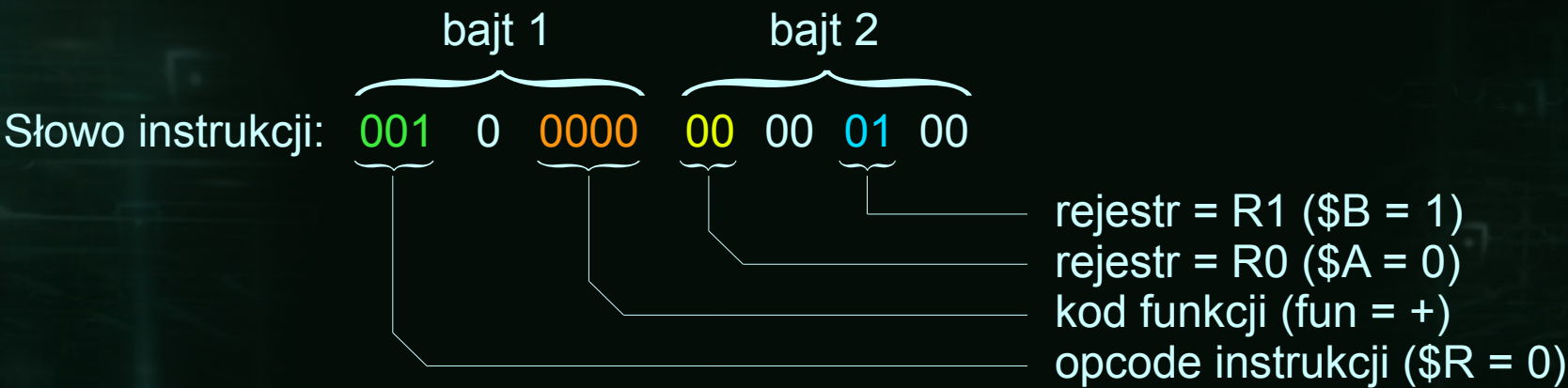
# Projekt $\mu P$ – koncepcja potoku: wykonanie instrukcji



# Projekt $\mu P$ – wykonanie instrukcji

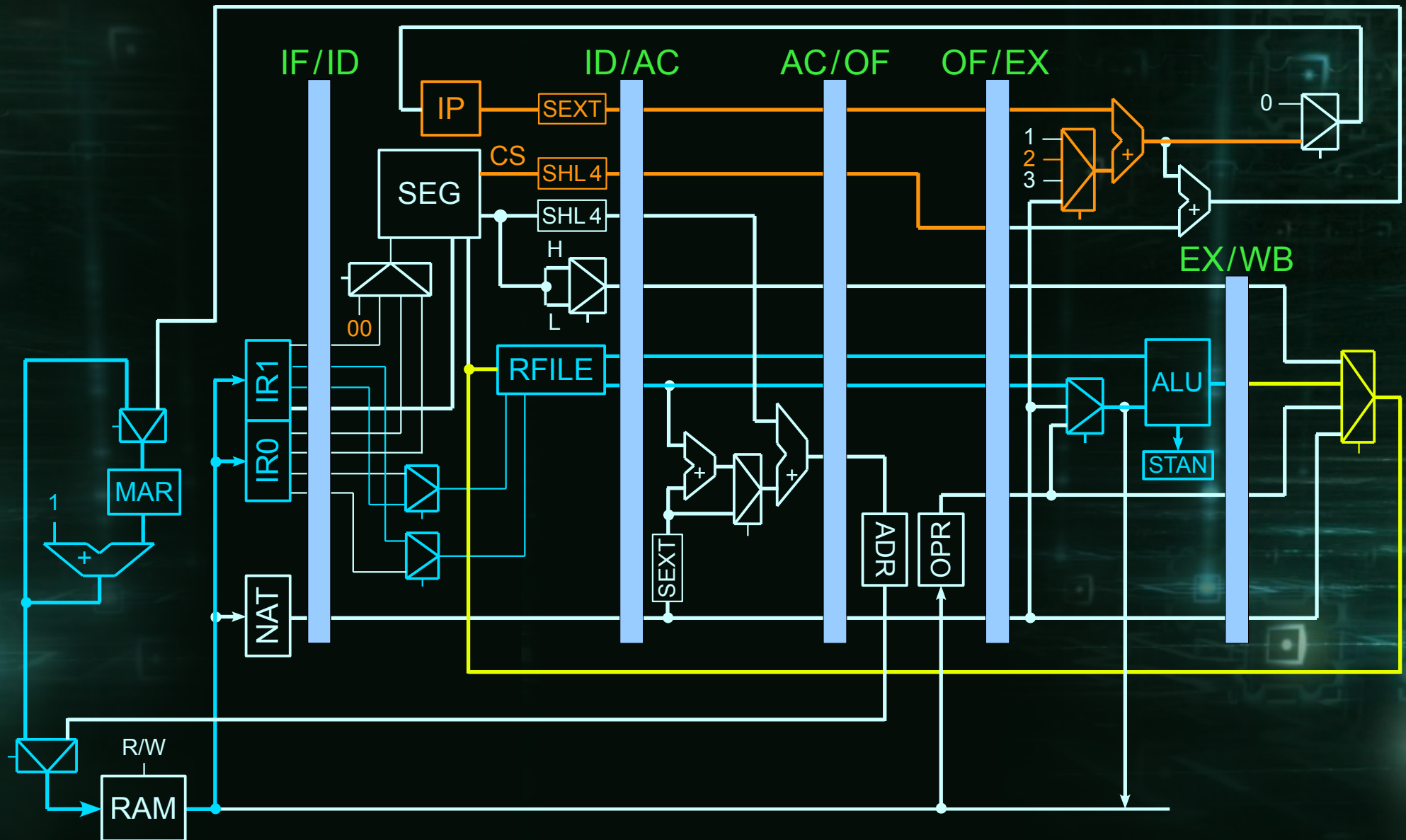
działanie potoku podczas wykonania instrukcji procesora

Rozkaz:  $R0 \leftarrow R0 + R1$



Projekt  $\mu P$  – wykonanie instrukcji

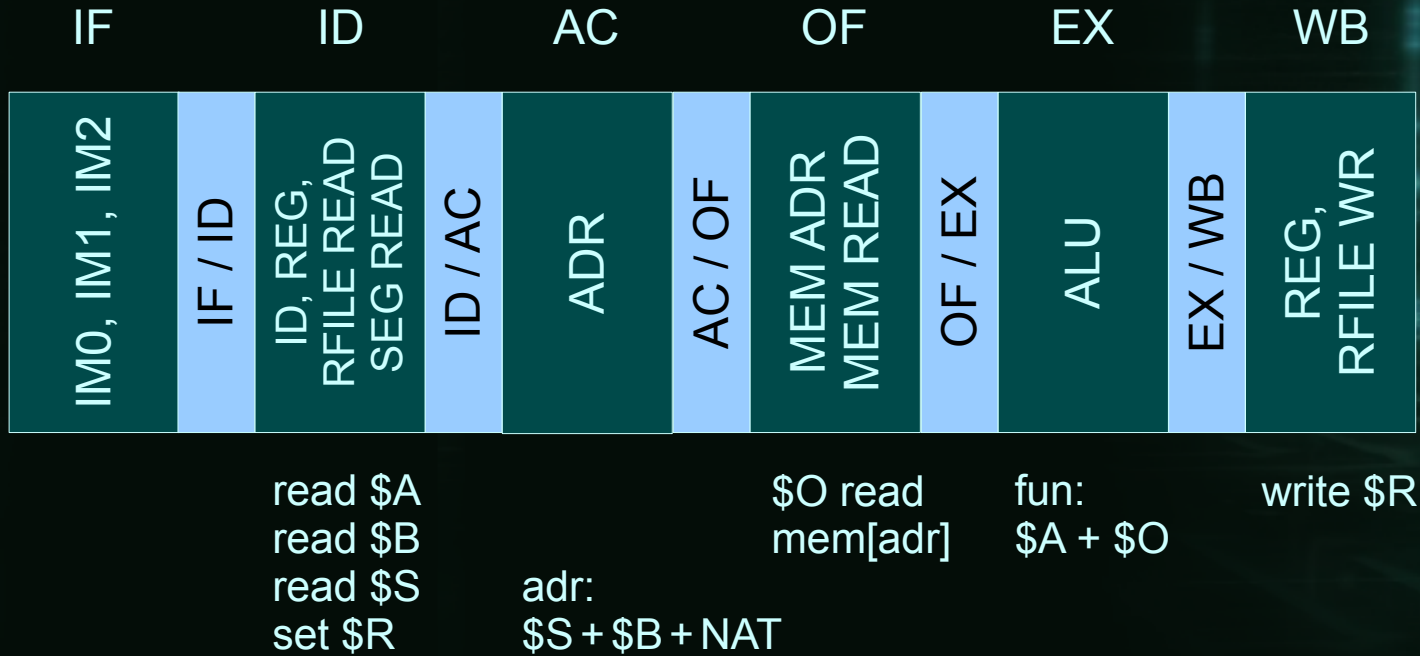
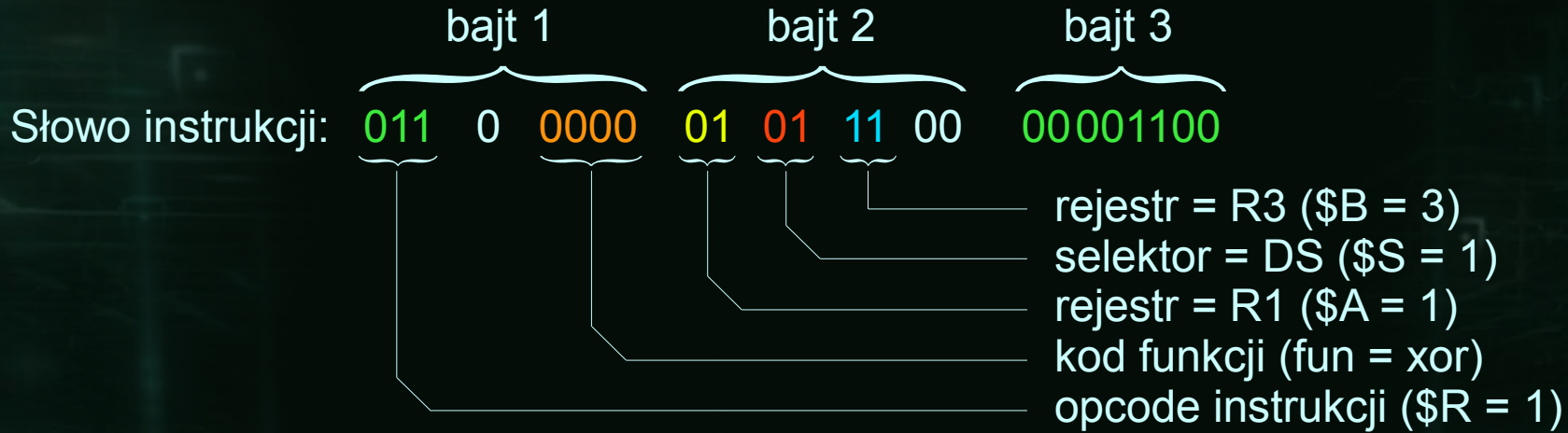
działanie potoku podczas wykonania instrukcji procesora

Rozkaz:  $R0 \leftarrow R0 + R1$ 

# Projekt $\mu P$ – wykonanie instrukcji

działanie potoku podczas wykonania instrukcji procesora

Rozkaz:  $R1 \leftarrow R1 \text{ xor } DS:[R3 + 12]$

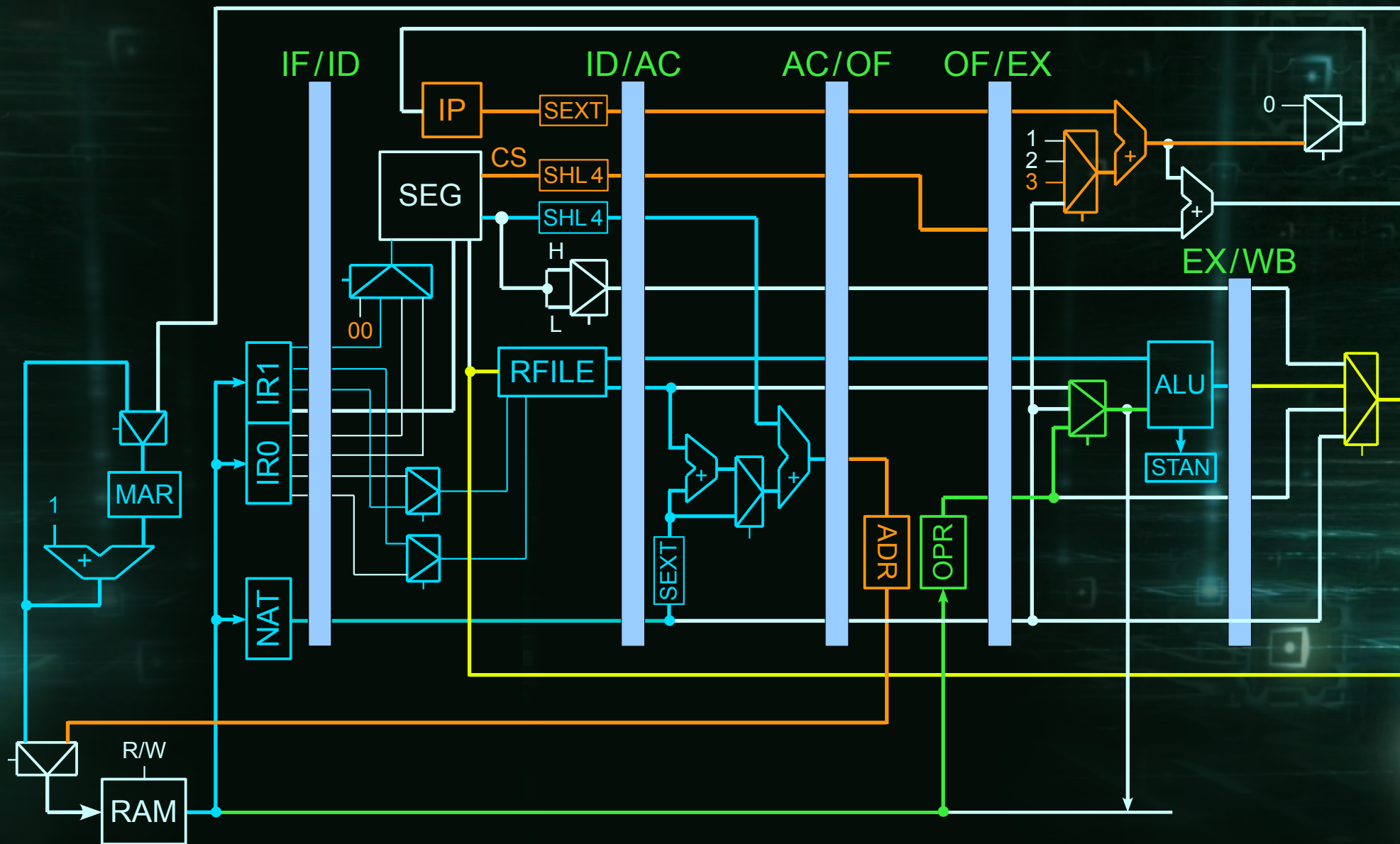




# Projekt $\mu P$ – wykonanie instrukcji

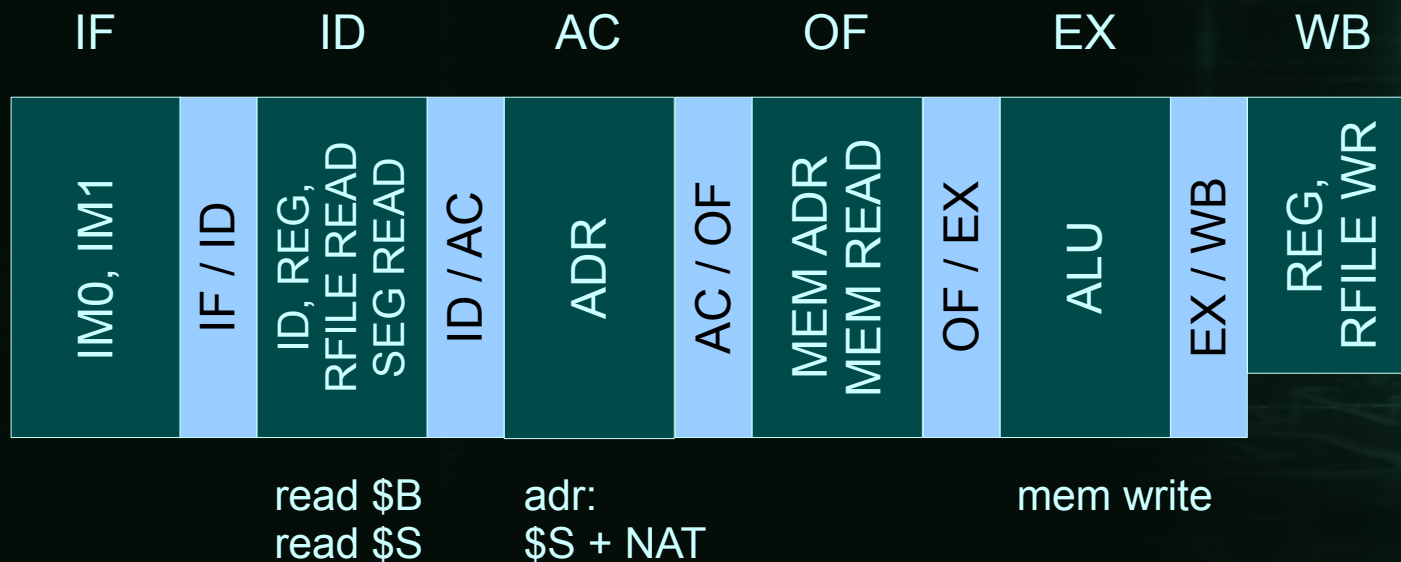
działanie potoku podczas wykonania instrukcji procesora

Rozkaz:  $R1 \leftarrow R1 \text{ xor } DS:[R3 + 12]$



Projekt  $\mu P$  – wykonanie instrukcji

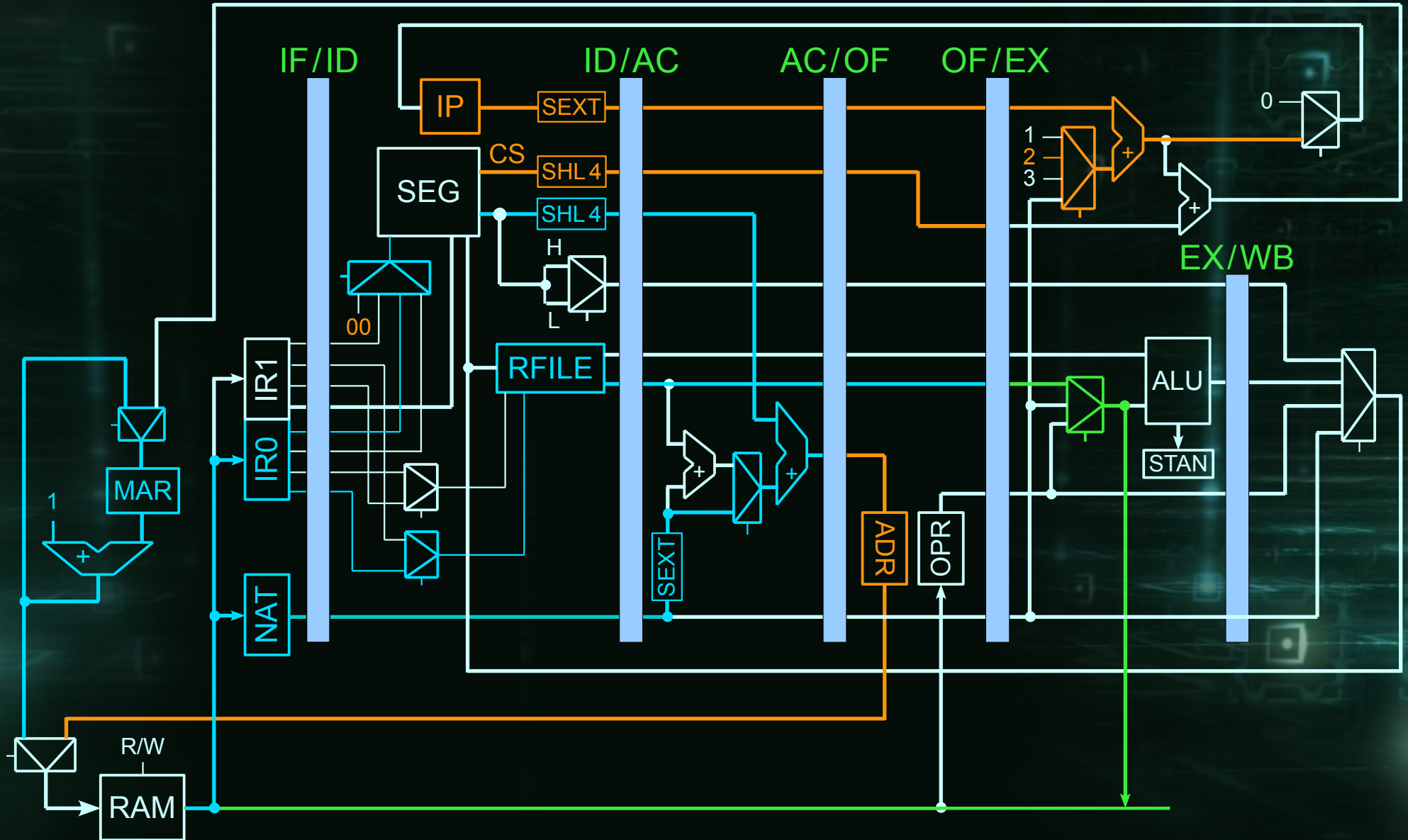
działanie potoku podczas wykonania instrukcji procesora

Rozkaz:  $DS:[102] \leftarrow R2$ 

# Projekt $\mu P$ – wykonanie instrukcji

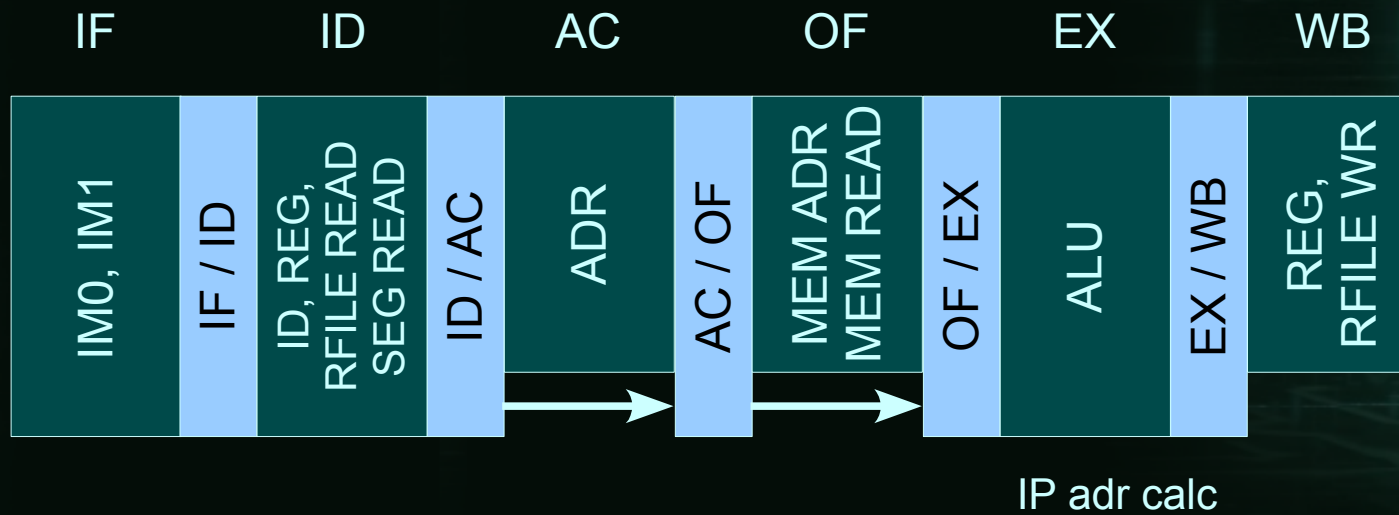
działanie potoku podczas wykonania instrukcji procesora

Rozkaz: DS:[102]  $\leftarrow$  R2



Projekt  $\mu P$  – wykonanie instrukcji

działanie potoku podczas wykonania instrukcji procesora

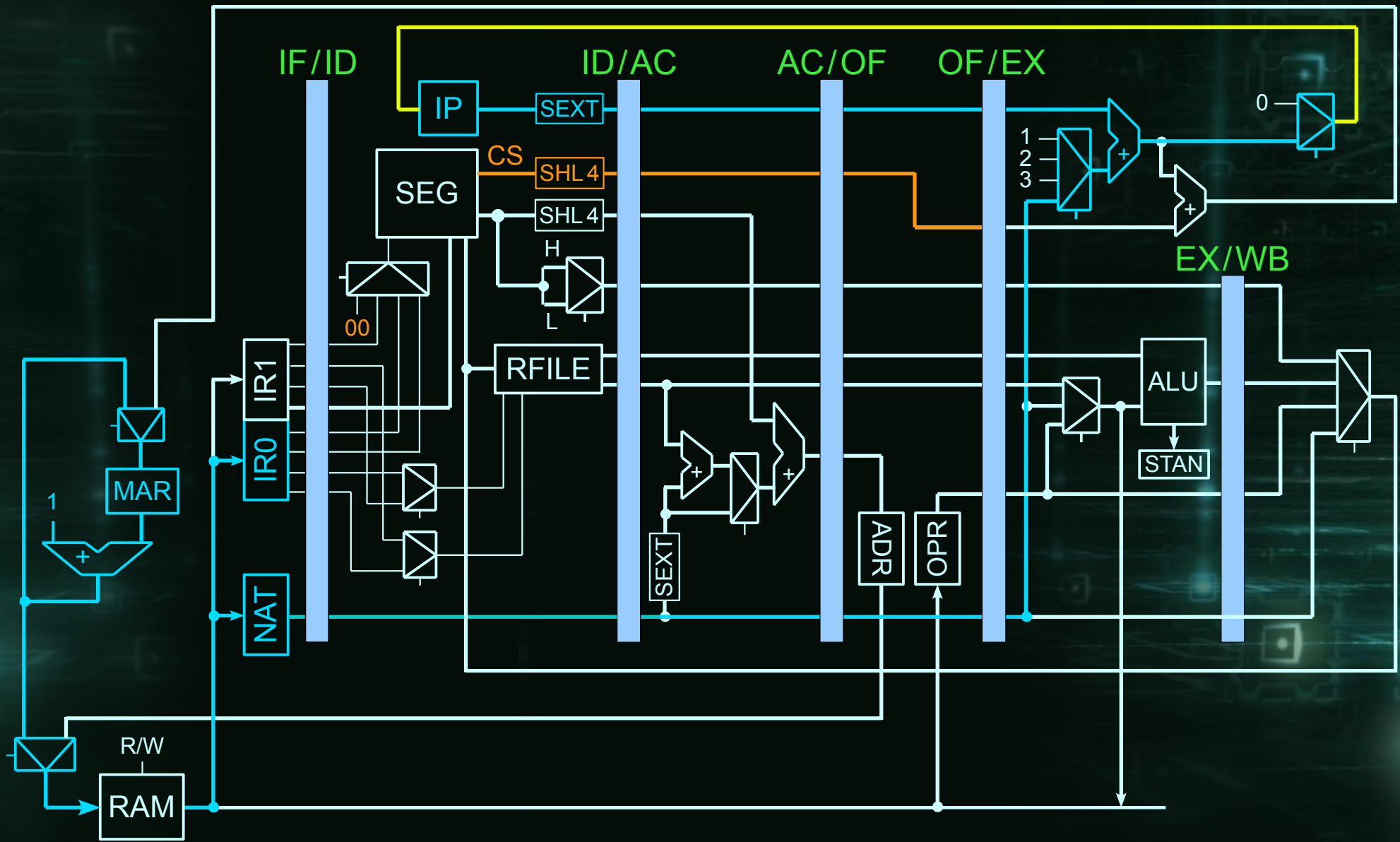
Rozkaz: ja CS:[IP+56]



# Projekt $\mu P$ – wykonanie instrukcji

działanie potoku podczas wykonania instrukcji procesora

Rozkaz: ja CS:[IP+56]



# Projekt $\mu P$ – koncepcja potoku: program w potoku



Projekt  $\mu P$  – program w potoku

praca procesora z wypełnionym potokiem

Program:R0  $\leftarrow$  DS:[100]R1  $\leftarrow$  R1 - R0

ja +3

DS:[R2+0]  $\leftarrow$  R3R2  $\leftarrow$  32

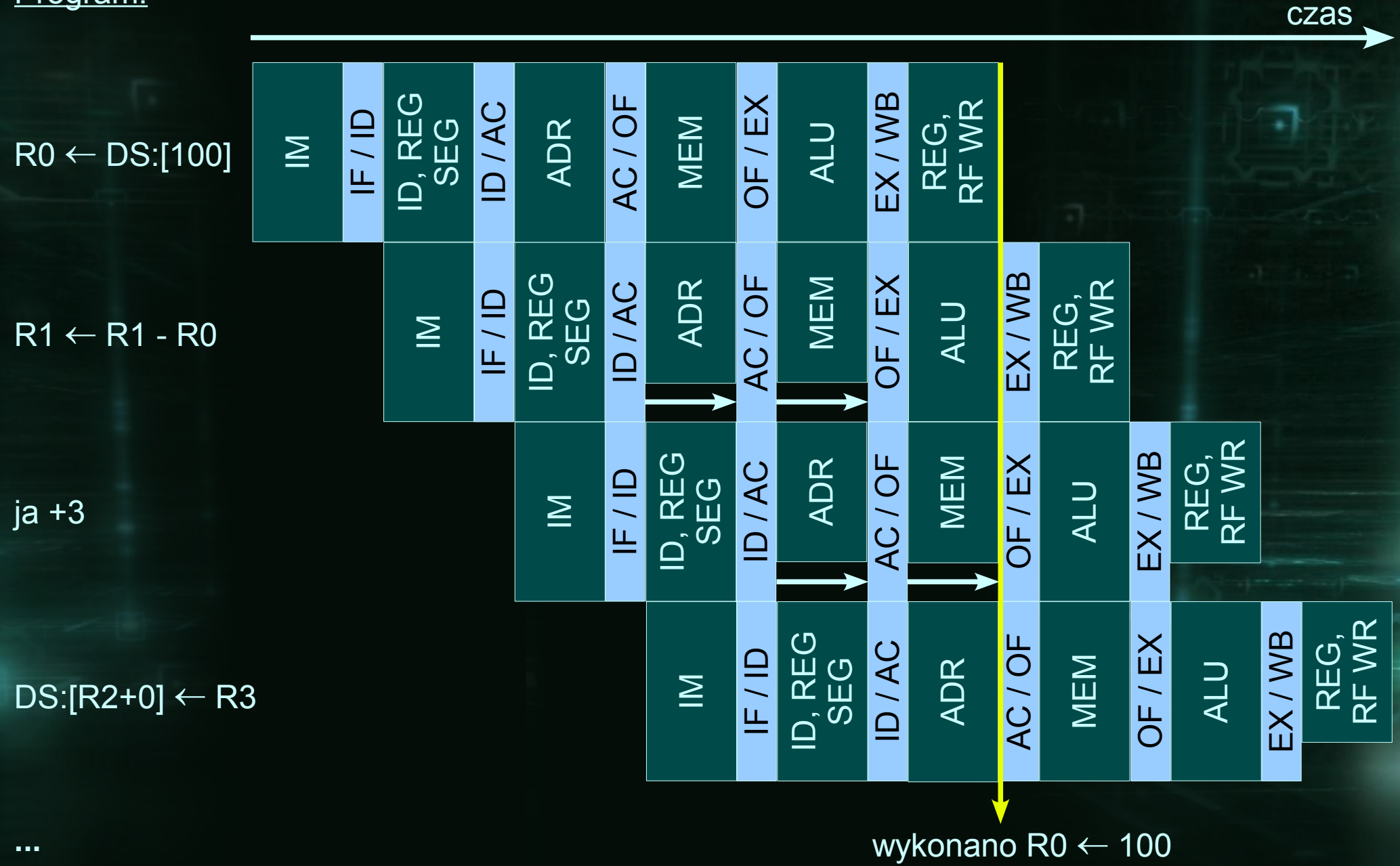
jmp 250

...

# Projekt $\mu P$ – program w potoku

praca procesora z wypełnionym potokiem

Program:





# Projekt $\mu P$ – program w potoku

praca procesora z wypełnionym potokiem

Program:

jmp 250

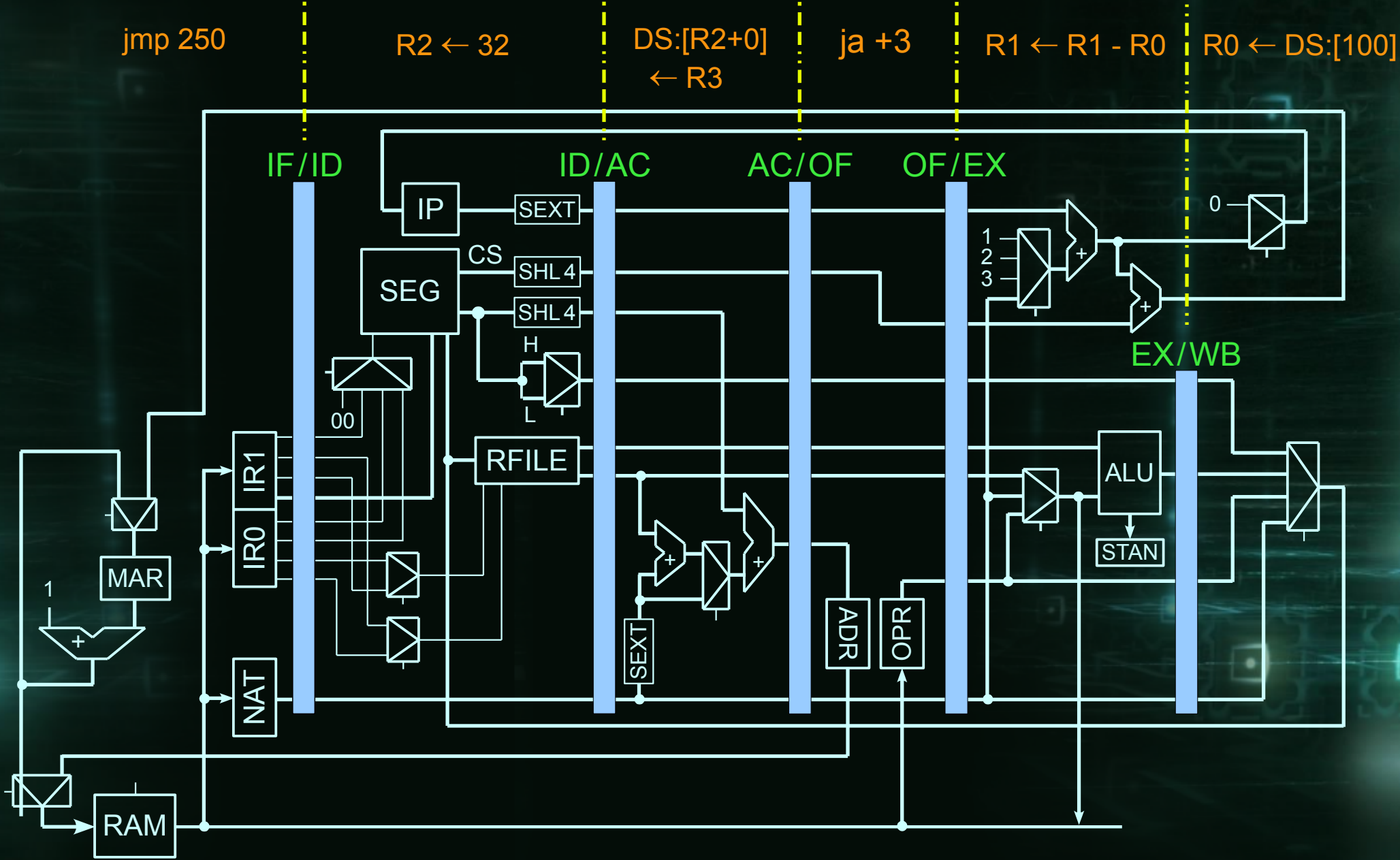
$R2 \leftarrow 32$

$DS:[R2+0] \leftarrow R3$

ja +3

$R1 \leftarrow R1 - R0$

$R0 \leftarrow DS:[100]$



Koniec wykładu