

# Języki niskiego poziomu

PODSTAWY INFORMATYKI

WYKŁAD NR 9

# Języki programowania

- ▶ Języki maszynowe (kod maszynowy)
- ▶ 2GL – języki asemblera (symboliczne)
- ▶ 3GL – języki programowania ogólnych zastosowań
- ▶ 4GL – języki specjalistyczne (zorientowane dziedzinowo)

# Języki programowania

- ▶ Języki maszynowe (kod maszynowy)
  - ▶ Kody rozkazów w postaci binarnej i binarne adresy absolutne jako argumenty
- ▶ 2GL – języki asemblera (symboliczne)
- ▶ 3GL – języki programowania ogólnych zastosowań
- ▶ 4GL – języki specjalistyczne (zorientowane dziedzinowo)

# Języki programowania

- ▶ Języki maszynowe (kod maszynowy)
  - ▶ Kody rozkazów w postaci binarnej i binarne adresy absolutne jako argumenty
- ▶ 2GL – języki asemblera (symboliczne)
  - ▶ Kody maszynowe i adresy w postaci symbolicznej
- ▶ 3GL – języki programowania ogólnych zastosowań
- ▶ 4GL – języki specjalistyczne (zorientowane dziedzinowo)

# Języki programowania

- ▶ Języki maszynowe (kod maszynowy)
  - ▶ Kody rozkazów w postaci binarnej i binarne adresy absolutne jako argumenty
- ▶ 2GL – języki asemblera (symboliczne)
  - ▶ Kody maszynowe i adresy w postaci symbolicznej
- ▶ 3GL – języki programowania ogólnych zastosowań
  - ▶ Instrukcje symboliczne, na które składa się często wiele rozkazów komputera
- ▶ 4GL – języki specjalistyczne (zorientowane dziedzinowo)

# Języki programowania

- ▶ Języki maszynowe (kod maszynowy)
  - ▶ Kody rozkazów w postaci binarnej i binarne adresy absolutne jako argumenty
- ▶ 2GL – języki asemblera (symboliczne)
  - ▶ Kody maszynowe i adresy w postaci symbolicznej
- ▶ 3GL – języki programowania ogólnych zastosowań
  - ▶ Instrukcje symboliczne, na które składa się często wiele rozkazów komputera
- ▶ 4GL – języki specjalistyczne (dziedzinowe)
  - ▶ Programista określa co chce zrobić a nie jak

# Języki programowania

- ▶ **Języki maszynowe (kod maszynowy)**
- ▶ **2GL – języki asemblera (symboliczne)**
- ▶ **3GL – języki programowania ogólnych zastosowań**
- ▶ **4GL – języki specjalistyczne (zorientowane dziedzinowo)**

# Kodowanie binarne

- ▶ Program składa się z ciągu rozkazów oraz danych
- ▶ Realizacja programu rozpoczyna się od rozkazu zapisanego w komórce pamięci o adresie 0
- ▶ Stały format rozkazów

KOD

Argument

# Lista rozkazów maszyny W

Symbol	Kod	Treść
DOD	00001	$(AK) + ((AD)) \rightarrow AK$
ODE	00010	$(AK) - ((AD)) \rightarrow AK$
ŁAD	00011	$(AK) \rightarrow (AD)$
POB	00100	$((AD)) \rightarrow AK$
SOB	00101	$(AD) \rightarrow L$
SOM	00110	$(AD) \rightarrow L$ , gdy $(AK) < 0$
STP	00111	zatrzymanie pracy zegara

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:  
 $(4) \rightarrow AK$

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:
  - (4) → AK
  - (AK) + (5) → AK

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:
  - (4) → AK
  - (AK) + (5) → AK
  - (AK) → 6

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:
  - (4) → AK
  - (AK) + (5) → AK
  - (AK) → 6
  - koniec

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:

(4) → AK                    00100 00000000100

(AK) + (5) → AK

(AK) → 6

koniec

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:

(4) → AK                    00100 00000000100

(AK) + (5) → AK        00001 00000000101

(AK) → 6

koniec

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:

(4) → AK                    00100 00000000100

(AK) + (5) → AK            00001 00000000101

(AK) → 6                    00011 00000000110

koniec

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiążanie:

(4) → AK                    00100 00000000100

(AK) + (5) → AK            00001 00000000101

(AK) → 6                    00011 00000000110

Koniec                      00111 000000000000

# Przykład programu binarnego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6

- ▶ Rozwiążanie:

(4) → AK                    00100 0000000100

(AK) + (5) → AK            00001 0000000101

(AK) → 6                    00011 0000000110

Koniec                      00111 000000000000

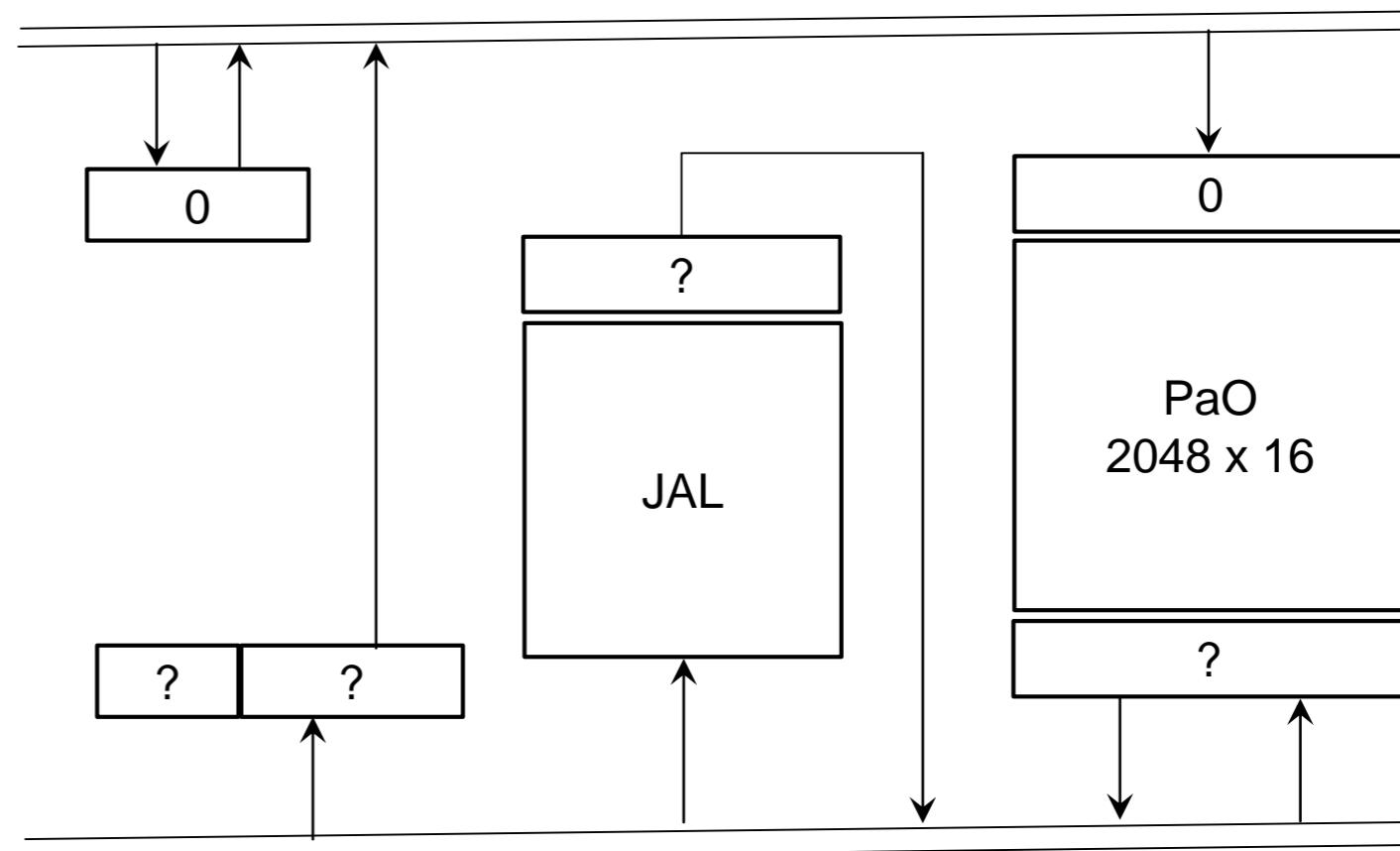
Pierwszy argument        0000 0000 0000 0111

Drugi argument            0000 0000 0001 0001

Wynik                        ?

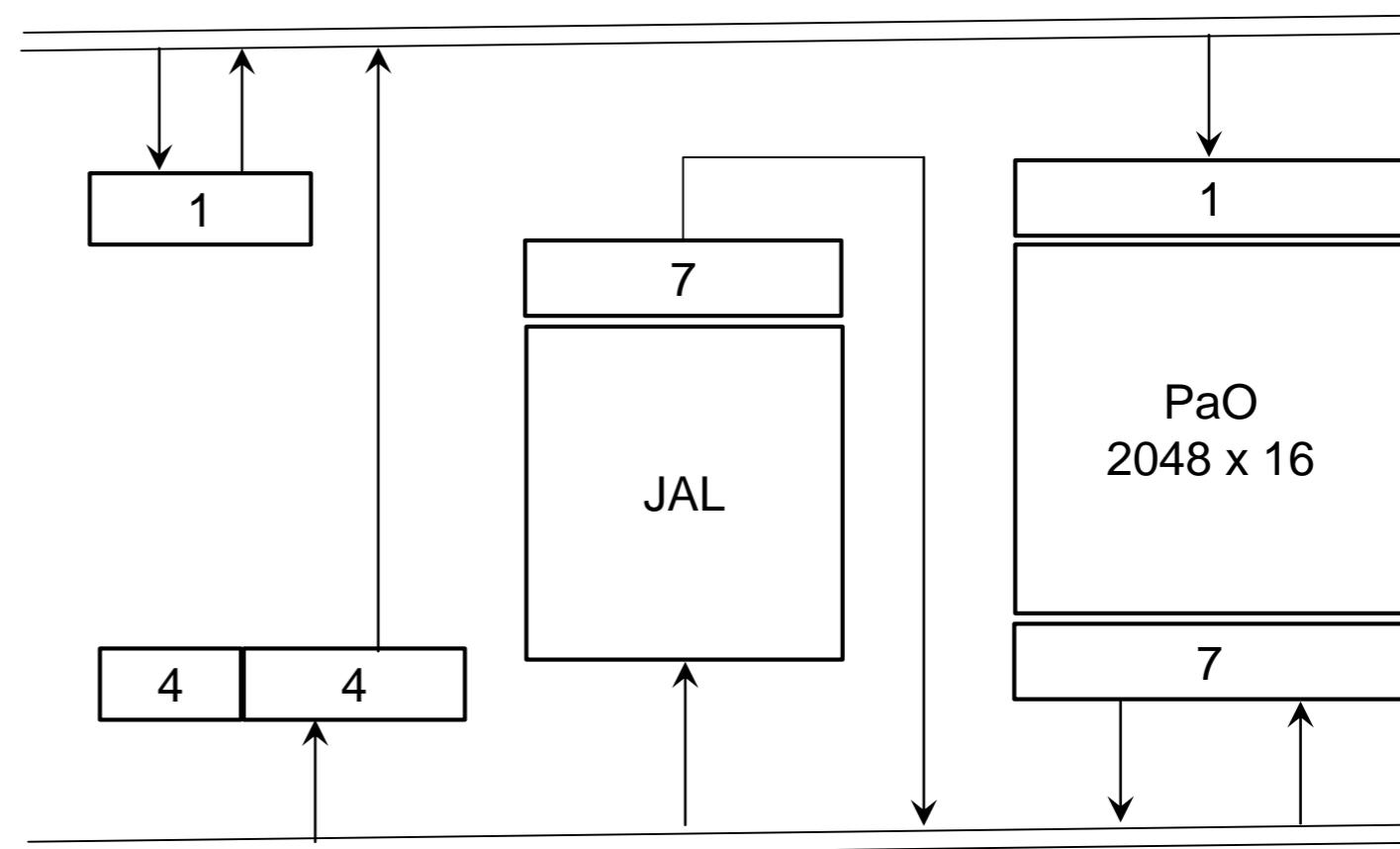
# Realizacja programu

0 00100 00000000100  
1 00001 00000000101  
2 00011 00000000110  
3 00111 000000000000  
4 0000 0000 0000 0111  
5 0000 0000 0001 0001  
6 ?  
AK ?  
L 0



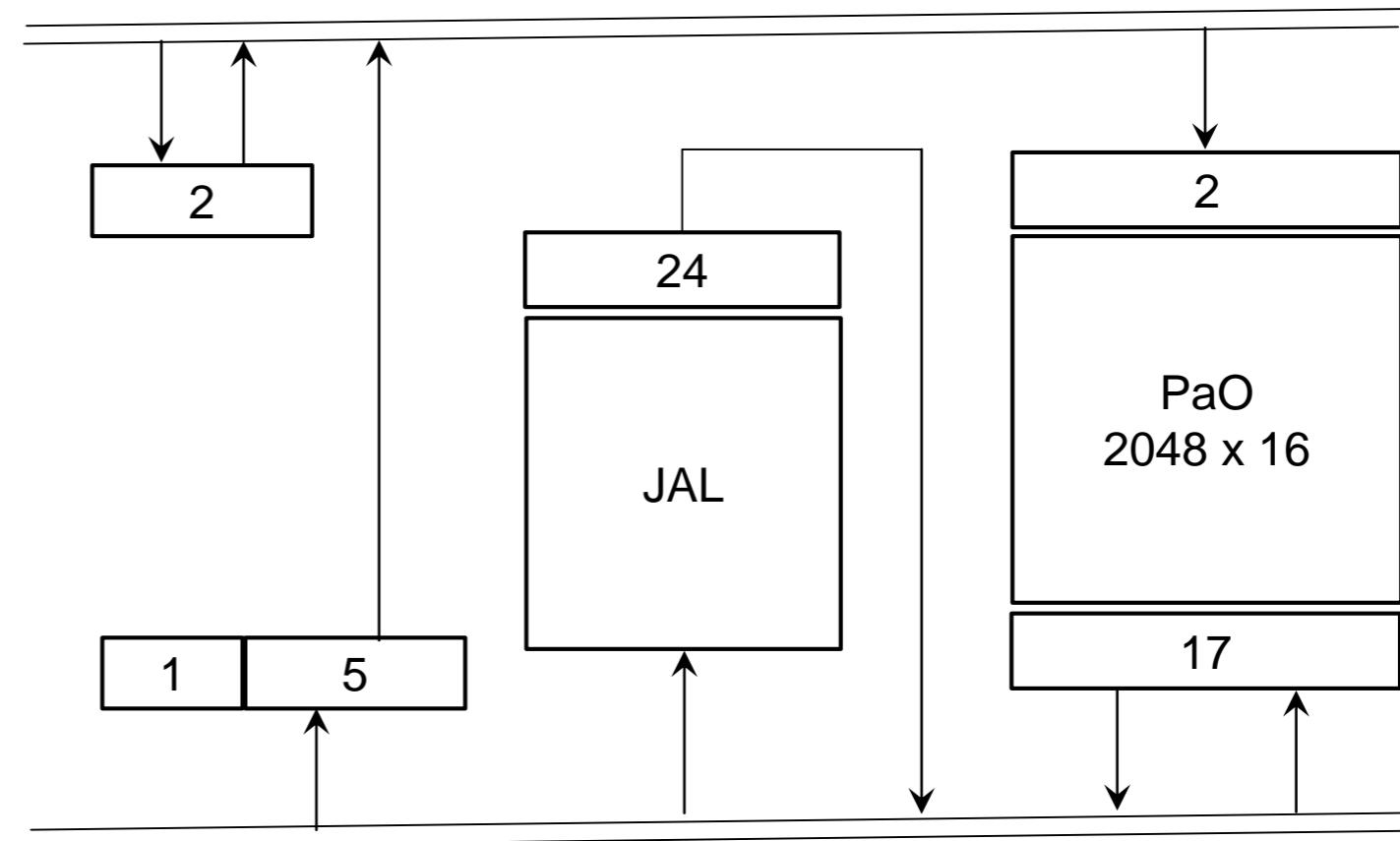
# Realizacja programu

```
0 00100 00000000100  
1 00001 00000000101  
2 00011 00000000110  
3 00111 00000000000  
4 0000 0000 0000 0111  
5 0000 0000 0001 0001  
6 ?  
AK 7  
L 1
```



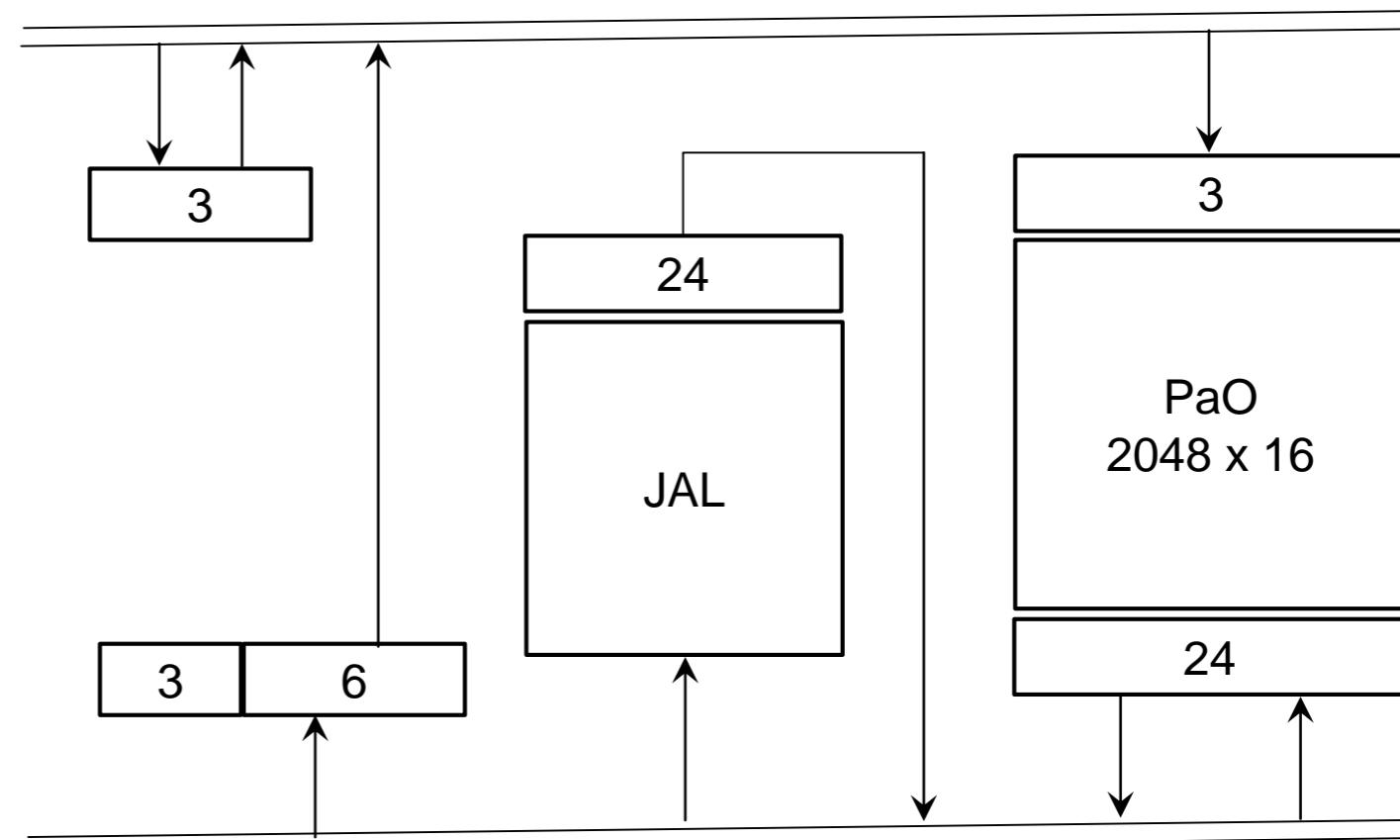
# Realizacja programu

0 00100 00000000100  
1 00001 00000000101  
2 00011 00000000110  
3 00111 000000000000  
4 0000 0000 0000 0111  
5 0000 0000 0001 0001  
6 ?  
AK 24  
L 2



# Realizacja programu

0 00100 00000000100  
1 00001 00000000101  
2 00011 00000000110  
3 00111 000000000000  
4 0000 0000 0000 0111  
5 0000 0000 0001 0001  
6 0000 0000 0001 1000  
AK 24  
L 3



# Programowanie binarne

## ► Zalety

- ▶ Program nie wymaga tłumaczenia
- ▶ Realizowany zaraz po wprowadzeniu do pamięci
- ▶ Małe wymagania pamięciowe

## ► Wady

- ▶ Operowanie na kodach binarnych rozkazów
- ▶ Konieczność wyliczania adresów komórek
- ▶ Bardzo trudna modyfikacja programu
- ▶ Rozwiążanie dedykowane konkretnemu procesorowi

# Asembler

- ▶ Zastąpienie kodów wewnętrznych procesora oraz adresów komórek pamięci w postaci binarnej specjalnymi symbolami
- ▶ Brak potrzeby wyliczania adresów komórek
- ▶ Program staje się bardziej zrozumiały
- ▶ Łatwiejsza modyfikacja
- ▶ Wciąż rozwiązywanie dedykowane konkretnemu procesorowi

# Przykład programu symbolicznego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6

- ▶ Rozwiążanie:

(4) → AK 00100 0000000100

(AK) + (5) → AK 00001 0000000101

(AK) → 6 00011 00000000110

Koniec 00111 000000000000

Pierwszy argument 0000 0000 0000 0111

Drugi argument 0000 0000 0001 0001

Wynik ?

# Przykład programu symbolicznego

- ▶ Dodaj do siebie zawartość dwóch komórek pamięci (4 i 5) i sumę zapisz do komórki 6
- ▶ Rozwiązanie:

(4) → AK	00100 00000000100	POB Arg1
(AK) + (5) → AK	00001 00000000101	DOD Arg2
(AK) → 6	00011 00000000110	ŁAD Suma
Koniec	00111 000000000000	STP
Pierwszy argument	0000 0000 0000 0111	Arg1: RST 7
Drugi argument	0000 0000 0001 0001	Arg2: RST 17
Wynik	?	Suma: RPA KON

# Przykład programu symbolicznego

- ▶ Dodaj do siebie zawartość dwóch komórek Arg1 i Arg2 sumę zapisz do komórki Suma
- ▶ Rozwiążanie w języku symbolicznym:

POB Arg1

DOD Arg2

ŁAD Suma

STP

Arg1: RST 7

Arg2: RST 17

Suma: RPA

KON

## Przykład 2 – obliczanie wartości bezwzględnej

W komórce pamięci oznaczonej etykietą **Liczba** znajduje się liczba całkowita ze znakiem, do komórki oznaczonej etykietą **Wynik** należy wpisać jej wartość bezwzględną

# Przykład 2 – obliczanie wartości bezwzględnej

POB Liczba

SOM Ujemna

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Składnia instrukcji asemblera

- ▶ Instrukcję dzielimy na 3 pola  
[<Etykieta>:] <kod> <argument(y)>
- ▶ Etykieta to nazwa zakończona dwukropkiem
- ▶ Etykieta może zostać pominięta
- ▶ Kod jest wymagany, może to być kod rozkazu lub dyrektywa (RST, RPA, KON)
- ▶ Argument jest opcjonalny (liczba i występowanie zależy od kodu rozkazu)
- ▶ Argumentem może być etykieta lub liczba dziesiętna

# Translacja z języka symbolicznego na kody binarne

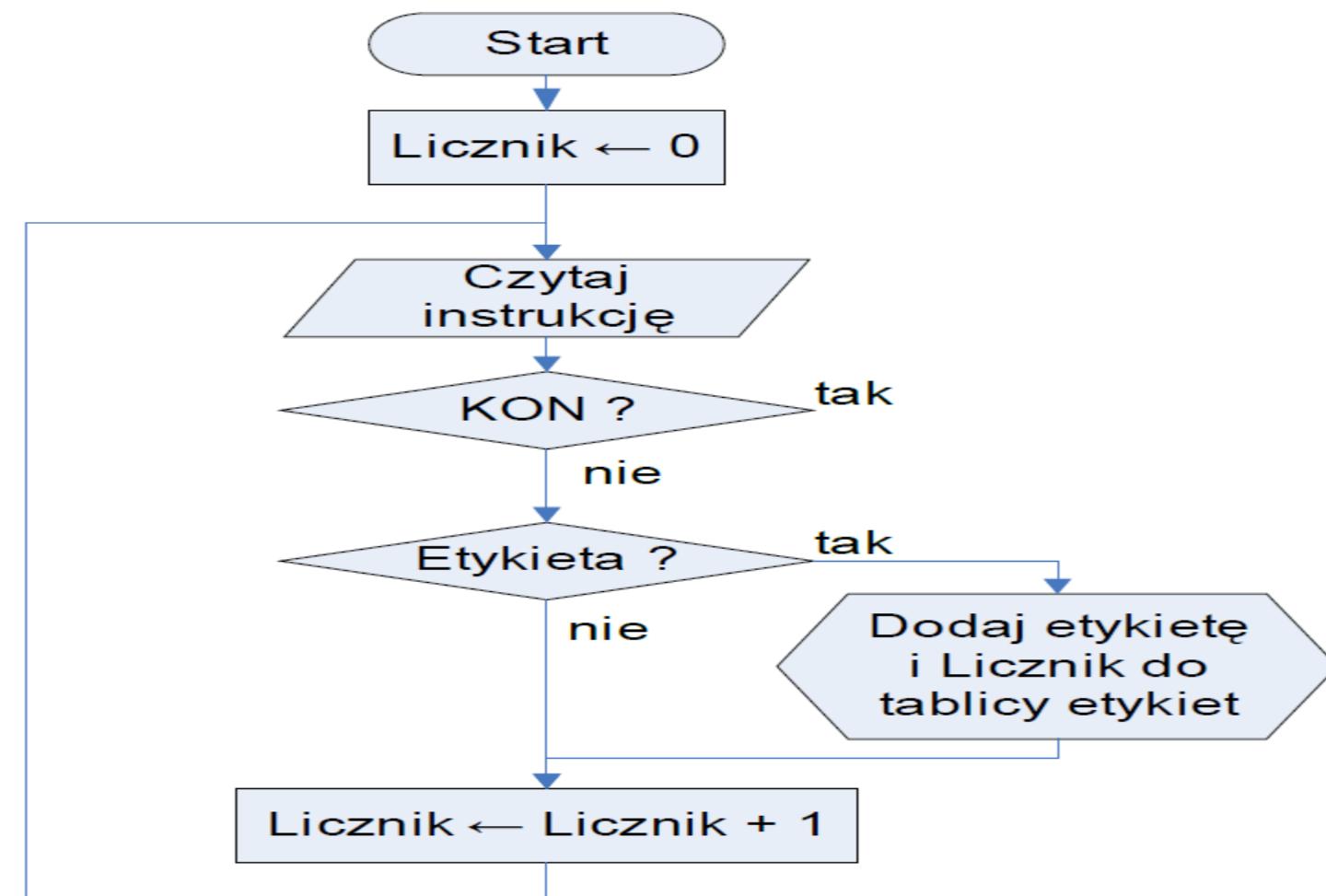
- ▶ Zadanie wykonywane przez programy tłumaczące zwane asemblerami
- ▶ Zwykle przebiega dwuetapowo
- ▶ Każdy etap (przebieg, przejście) to czytanie kodu źródłowego od początku do końca
- ▶ Pierwszy etap – utworzenie tablicy etykiet T2
- ▶ Drugi etap – właściwe tłumaczenie instrukcji na podstawie tablicy kodów T1 (niezmiennej dla każdego języka) i stworzonej wcześniej tablicy etykiet T2

# Tablica kodów rozkazów T1

- ▶ Przyporządkowuje kodom symbolicznym rozkazów odpowiednie kody binarne

Symbol	Kod	Treść
DOD	00001	$(AK) + ((AD)) \rightarrow AK$
ODE	00010	$(AK) - ((AD)) \rightarrow AK$
ŁAD	00011	$(AK) \rightarrow (AD)$
POB	00100	$((AD)) \rightarrow AK$
SOB	00101	$(AD) \rightarrow L$
SOM	00110	$(AD) \rightarrow L$ , gdy $(AK) < 0$
STP	00111	zatrzymanie pracy zegara

# Pierwszy przebieg asemblacji



# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

**ŁAD Wynik**

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

ŁAD Wynik

**STP**

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

STP

**Ujemna:** ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

STP

Ujemna: ODE Liczba

**ODE Liczba**

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

**ŁAD Wynik**

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

Liczba 8

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna

Ujemna 4

ŁAD Wynik

Liczba 8

STP

Wynik 9

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Tworzenie tablicy etykiet

POB Liczba

SOM Ujemna              Ujemna      4

ŁAD Wynik              Liczba      8

STP              Wynik      9

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

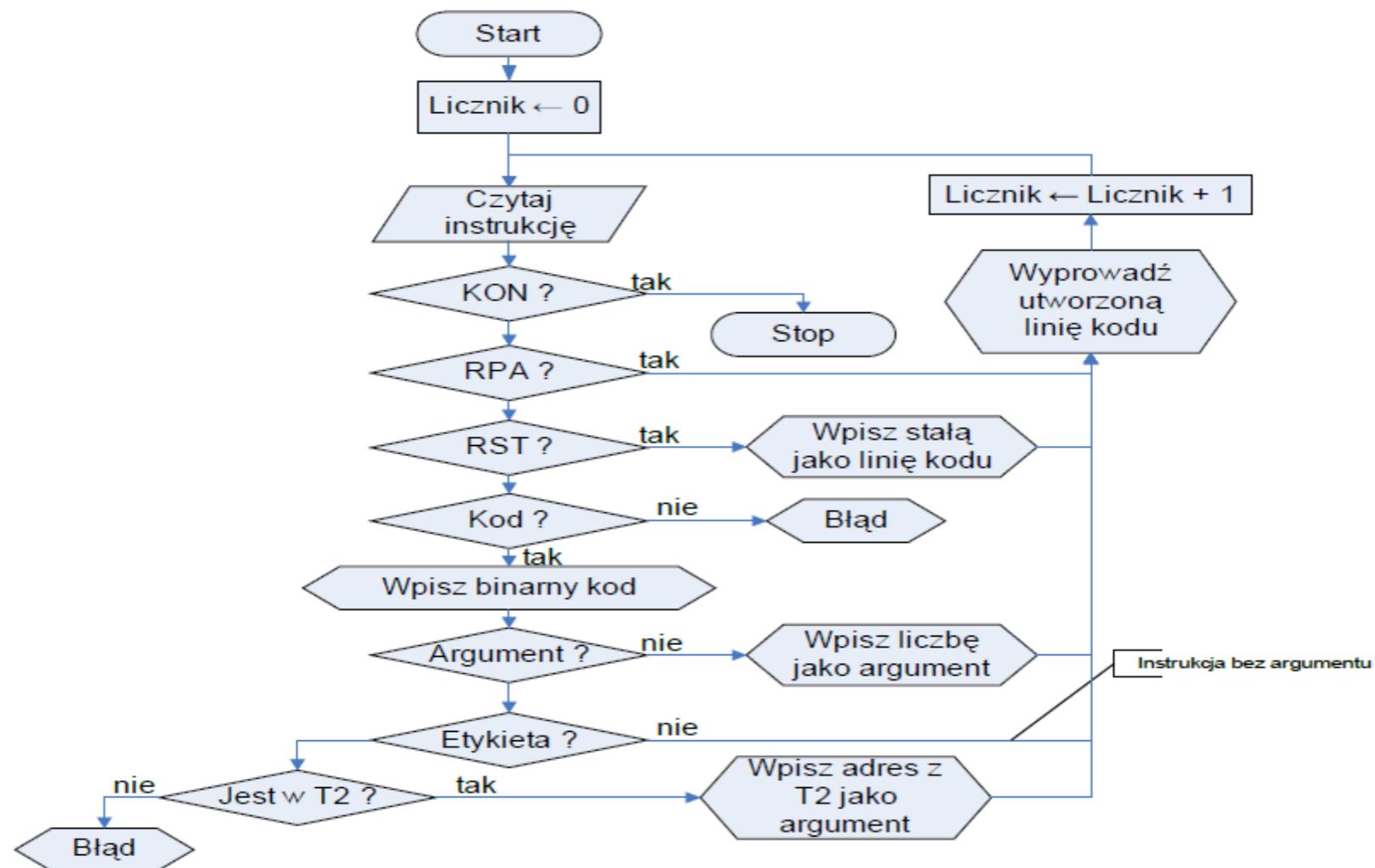
STP

Liczba: RST -21

Wynik: RPA

KON

# Drugi przebieg asemblacji



# Generacja kodu wynikowego

POB Liczba

SOM Ujemna

ŁAD Wynik

STP

Ujemna: ODE Liczba

ODE Liczba

ŁAD Wynik

STP

Liczba: RST -21

Wynik: RPA

KON

# Generacja kodu wynikowego

POB	Liczba	00100 00000001000
SOM	Ujemna	
ŁAD	Wynik	
STP		
Ujemna:	ODE	Liczba
	ODE	Liczba
	ŁAD	Wynik
	STP	
Liczba:	RST	-21
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

POB	Liczba	00100 00000001000
SOM	Ujemna	00110 00000000100
ŁAD	Wynik	
STP		
Ujemna:	ODE	Liczba
	ODE	Liczba
	ŁAD	Wynik
	STP	
Liczba:	RST	-21
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

POB	Liczba	00100 00000001000
SOM	Ujemna	00110 00000000100
ŁAD	Wynik	00011 00000001001
STP		
Ujemna:	ODE	Liczba
	ODE	Liczba
	ŁAD	Wynik
	STP	
Liczba:	RST	-21
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

POB	Liczba	00100 00000001000
SOM	Ujemna	00110 00000000100
ŁAD	Wynik	00011 00000001001
STP		00111 00000000000

Ujemna: ODE  
ODE  
ŁAD Wynik  
STP

Liczba: RST -21

Wynik: RPA  
KON

# Generacja kodu wynikowego

	POB Liczba	00100 00000001000
	SOM Ujemna	00110 00000000100
	ŁAD Wynik	00011 00000001001
	STP	00111 00000000000
Ujemna:	ODE Liczba	00010 00000001000
	ODE Liczba	
	ŁAD Wynik	
	STP	
Liczba:	RST -21	
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

	POB Liczba	00100 00000001000
	SOM Ujemna	00110 00000000100
	ŁAD Wynik	00011 00000001001
	STP	00111 00000000000
Ujemna:	ODE Liczba	00010 00000001000
	ODE Liczba	00010 00000001000
	ŁAD Wynik	
	STP	
Liczba:	RST -21	
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

	POB	Liczba	00100 00000001000
	SOM	Ujemna	00110 00000000100
	ŁAD	Wynik	00011 00000001001
	STP		00111 00000000000
Ujemna:	ODE	Liczba	00010 00000001000
	ODE	Liczba	00010 00000001000
	ŁAD	Wynik	00011 00000001001
	STP		
Liczba:	RST	-21	
Wynik:	RPA		
	KON		

# Generacja kodu wynikowego

	POB	Liczba	00100 00000001000
	SOM	Ujemna	00110 00000000100
	ŁAD	Wynik	00011 00000001001
	STP		00111 00000000000
Ujemna:	ODE	Liczba	00010 00000001000
	ODE	Liczba	00010 00000001000
	ŁAD	Wynik	00011 00000001001
	STP		00111 00000000000
Liczba:	RST	-21	
Wynik:	RPA		
	KON		

# Generacja kodu wynikowego

	POB Liczba	00100 00000001000
	SOM Ujemna	00110 00000000100
	ŁAD Wynik	00011 00000001001
	STP	00111 00000000000
Ujemna:	ODE Liczba	00010 00000001000
	ODE Liczba	00010 00000001000
	ŁAD Wynik	00011 00000001001
	STP	00111 00000000000
Liczba:	RST -21	1111 1111 1110 1011
Wynik:	RPA	
	KON	

# Generacja kodu wynikowego

	POB	Liczba	00100 00000001000
	SOM	Ujemna	00110 00000000100
	ŁAD	Wynik	00011 00000001001
	STP		00111 00000000000
Ujemna:	ODE	Liczba	00010 00000001000
	ODE	Liczba	00010 00000001000
	ŁAD	Wynik	00011 00000001001
	STP		00111 00000000000
Liczba:	RST	-21	1111 1111 1110 1011
Wynik:	RPA		0000 0000 0000 0000
	KON		

# Generacja kodu wynikowego

	POB Liczba	00100 00000001000	20 08
	SOM Ujemna	00110 00000000100	30 04
	ŁAD Wynik	00011 00000001001	18 09
	STP	00111 00000000000	38 00
Ujemna:	ODE Liczba	00010 00000001000	10 08
	ODE Liczba	00010 00000001000	10 08
	ŁAD Wynik	00011 00000001001	18 09
	STP	00111 00000000000	38 00
Liczba:	RST -21	1111 1111 1110 1011	FF EB
Wynik:	RPA	0000 0000 0000 0000	00 00
	KON		

## Przykład 3 – sumowanie elementów tablicy

- ▶ Dana jest tablica  $n$  liczb całkowitych, na której początek wskazuje etykieta *Tab*, napisać w języku assemblera maszyny W program obliczający sumę wszystkich elementów tej tablicy i zapisujący ją w zmiennej *Suma*

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. Dopóki nie przetworzono całej tablicy powtarzaj:

  3. Pobierz kolejny element tablicy
  4. Dodaj go do zmiennej  $\text{Suma}$
  5. Zmniejsz liczbę elementów pozostałych do przetworzenia

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. Dopóki nie przetworzono całej tablicy powtarzaj:

  3. Pobierz kolejny element tablicy
  4. Dodaj go do zmiennej  $\text{Suma}$

5. **Zmniejsz liczbę elementów pozostałych do przetworzenia**

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. Dopóki nie przetworzono całej tablicy powtarzaj:

  3. Pobierz kolejny element tablicy
  4. Dodaj go do zmiennej  $\text{Suma}$
  5.  $\mathbf{N} \leftarrow \mathbf{N} - 1$

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. Dopóki nie przetworzono całej tablicy powtarzaj:

  3. Pobierz kolejny element tablicy
  4. Dodaj go do zmiennej  $\text{Suma}$
  5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. **Dopóki nie przetworzono całej tablicy powtarzaj:**
3. Pobierz kolejny element tablicy
4. Dodaj go do zmiennej  $\text{Suma}$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. **Dopóki  $N > 0$  powtarzaj:**
3. Pobierz kolejny element tablicy
4. Dodaj go do zmiennej  $\text{Suma}$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $Suma \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
3. Pobierz kolejny element tablicy
4. Dodaj go do zmiennej  $Suma$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $Suma \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
3. **Pobierz kolejny element tablicy**
4. Dodaj go do zmiennej  $Suma$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $Suma \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
3. **POB Tab**
4. Dodaj go do zmiennej  $Suma$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $Suma \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
3. POB Tab
4. Dodaj go do zmiennej  $Suma$
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $Suma \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
3. POB Tab
4. **Dodaj go do zmiennej Suma**
5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1.  $\text{Suma} \leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:

  3. POB Tab
  4. **DOD Suma  
ŁAD Suma**
  5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

1. Suma  $\leftarrow 0$
2. Dopóki  $N > 0$  powtarzaj:
  3. POB Tab
  4. DOD Suma  
ŁAD Suma
  5.  $N \leftarrow N - 1$

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

Dopóki  $N > 0$  powtarzaj:

POB Tab

DOD Suma

ŁAD Suma

**N  $\leftarrow$  N - 1**

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

Dopóki N > 0 powtarzaj:

POB Tab

DOD Suma

ŁAD Suma

**POB N**

**ODE Jeden**

**ŁAD N**

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

Dopóki N > 0 powtarzaj:

POB Tab

DOD Suma

ŁAD Suma

POB N

ODE Jeden

ŁAD N

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow 0$

**Dopóki  $N > 0$  powtarzaj:**

POB Tab

DOD Suma

ŁAD Suma

POB N

ODE Jeden

ŁAD N

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

**POB N**

**ODE Jeden**

**SOM Koniec**

POB Tab

DOD Suma

ŁAD Suma

POB N

ODE Jeden

ŁAD N

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

POB N

ODE Jeden

SOM Koniec

POB Tab

DOD Suma

ŁAD Suma

**POB N**

**ODE Jeden**

**ŁAD N**

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

**Pętla: POB N**

**ODE Jeden**

**SOM Koniec**

**ŁAD N**

POB Tab

DOD Suma

ŁAD Suma

**SOB Pętla**

**Koniec:**

# Sumowanie elementów tablicy – algorytm

Suma  $\leftarrow$  0

Pętla: POB N

ODE Jeden

SOM Koniec

ŁAD N

POB Tab

DOD Suma

ŁAD Suma

SOB Pętla

Koniec:

# Sumowanie elementów tablicy – algorytm

**Suma**  $\leftarrow$  0

Pętla: POB N

ODE Jeden

SOM Koniec

ŁAD N

POB Tab

DOD Suma

ŁAD Suma

SOB Pętla

Koniec:

# Sumowanie elementów tablicy – program

Pętla: POB N

ODE Jeden

SOM Koniec

ŁAD N

POB Tab

DOD Suma

ŁAD Suma

SOB Pętla

Koniec: STP

**Suma: RST 0**

# Sumowanie elementów tablicy – program

Pętla:POB N              **N:**      **RST 5**

ODE Jeden  
SOM Koniec  
ŁAD N  
POB Tab  
DOD Suma  
ŁAD Suma  
SOB Pętla

Koniec: STP

Suma: RST 0

# Sumowanie elementów tablicy – program

Pętla: POB N

ODE Jeden

SOM Koniec

ŁAD N

POB Tab

DOD Suma

ŁAD Suma

SOB Pętla

Koniec: STP

Suma: RST 0

N: RST 5

Jak zapisać tablicę ?

# Sumowanie elementów tablicy – program

Pętla: POB N	N: RST 5
ODE Jeden	<b>Tab: RST 6</b>
SOM Koniec	<b>RST 2</b>
ŁAD N	<b>RST -4</b>
POB Tab	<b>RST 0</b>
DOD Suma	<b>RST -3</b>
ŁAD Suma	
SOB Pętla	
Koniec: STP	
Suma: RST 0	

# Sumowanie elementów tablicy – program

Pętla: POB N	N: RST 5
ODE Jeden	Tab: RST 6
SOM Koniec	RST 2
ŁAD N	RST -4
POB Tab	RST 0
DOD Suma	RST -3
ŁAD Suma	<b>Jeden: RST 1</b>
SOB Pętla	
Koniec: STP	
Suma: RST 0	

# Sumowanie elementów tablicy – program

Pętla: POB N	N:	RST 5
ODE Jeden	Tab:	RST 6
SOM Koniec		RST 2
ŁAD N		RST -4
POB Tab		RST 0
DOD Suma		RST -3
ŁAD Suma	Jeden:	RST 1
SOB Pętla		
Koniec: STP		
Suma:		RST 0

# Sumowanie elementów tablicy – program

Pętla:POB N	N:	RST 5
ODE Jeden	Tab:	RST 6
SOM Koniec		RST 2
ŁAD N		RST -4
POB Tab		RST 0
DOD Suma		RST -3
ŁAD Suma	Jeden:	RST 1
SOB Pętla		
Koniec: STP		
Suma: RST 0		

**To nie działa !**

# Sumowanie elementów tablicy – program

Pętla: POB N

ODE Jeden

SOM Koniec

ŁAD N

POB Tab

DOD Suma

ŁAD Suma

SOB Pętla

Koniec: STP

Suma: RST 0

N: RST 5

Tab: RST 6

RST 2

RST -4

RST 0

RST -3

Jeden: RST 1

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	Suma:	RST 0
	ODE Jeden	N:	RST 5
	SOM Koniec	Tab:	RST 6
	ŁAD N		RST 2
<b>Rozkaz:</b>	POB Tab		RST -4
	DOD Suma		RST 0
	ŁAD Suma		RST -3
	<b>POB Rozkaz</b>	Jeden:	RST 1
	<b>DOD Jeden</b>		
	<b>ŁAD Rozkaz</b>		
	SOB Pętla		
Koniec:	STP		

# Sumowanie elementów tablicy – program poprawny

Pętla:  
POB N  
ODE Jeden  
SOM Koniec  
ŁAD N  
  
Rozkaz:  
POB Tab  
DOD Suma  
ŁAD Suma  
POB Rozkaz  
DOD Jeden  
ŁAD Rozkaz  
SOB Pętla  
  
Koniec: STP

Suma:  
N: RST 0  
Tab: RST 5  
RST 6  
RST 2  
RST -4  
RST 0  
RST -3  
Jeden: RST 1

KOD

ADRES

# Lista rozkazów - przykład

Nazwa symboliczna	Kod
DOD	001
ODE	010
ŁAD	011
POB	100
SOB	101
SOM	110
STP	111

# Sumowanie elementów tablicy – program poprawny

Pętla: POB N  
ODE Jeden  
SOM Koniec  
ŁAD N  
  
Rozkaz: POB Tab  
DOD Suma  
ŁAD Suma  
POB Rozkaz  
DOD Jeden  
ŁAD Rozkaz  
SOB Pętla  
  
Koniec: STP

Suma: RST 0  
N: RST 5  
Tab: RST 6  
RST 2  
RST -4  
RST 0  
RST -3  
Jeden: RST 1

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0101
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	<b>POB N</b>	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0101
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	<b>ODE Jeden</b>	010 10011	N:	RST 5	0000 0101
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0101
	<b>SOM Koniec</b>	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	<b>ŁAD N</b>	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	<b>RST 6</b>	<b>0000 0110</b>
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	<b>POB Tab</b>	100 <b>01110</b>		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0000
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	<b>DOD Suma</b>	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	<b>ŁAD Suma</b>	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	<b>POB Tab</b>	<b>100 01110 (-114)</b>		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	<b>POB Rozkaz</b>	<b>100 00100</b>	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01110		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	<b>DOD Jeden</b>	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab+1	100 01111 (-113)		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	<b>ŁAD Rozkaz</b>	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	<b>SOB Pętla</b>	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	<b>POB N</b>	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	<b>ODE Jeden</b>	010 10011	N:	RST 5	0000 0100
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0100
	<b>SOM Koniec</b>	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	<b>ŁAD N</b>	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	<b>0000 0010</b>
Rozkaz:	<b>POB Tab</b>	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 0110
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	<b>DOD Suma</b>	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	<b>ŁAD Suma</b>	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	<b>POB Rozkaz</b>	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 01111		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	<b>DOD Jeden</b>	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	<b>100 10000</b>		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	<b>ŁAD Rozkaz</b>	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	<b>SOB Pętla</b>	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	<b>POB N</b>	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	<b>ODE Jeden</b>	010 10011	N:	RST 5	0000 0011
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0011
	<b>SOM Koniec</b>	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0010
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	<b>ŁAD N</b>	011 01101		RST 2	0000 0010
Rozkaz:	POB Tab	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

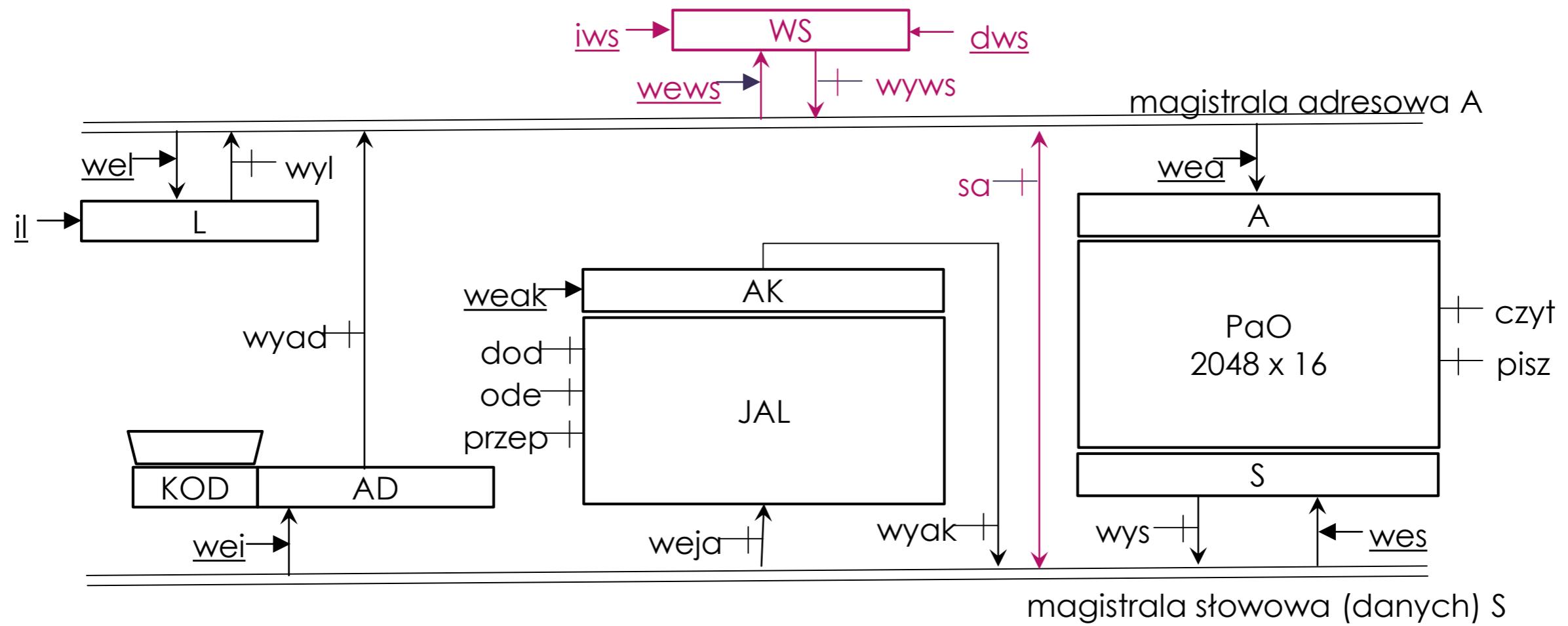
# Sumowanie elementów tablicy – program poprawny

Pętla:	POB N	100 01101	Suma:	RST 0	0000 1000
	ODE Jeden	010 10011	N:	RST 5	0000 0010
	SOM Koniec	110 01011	Tab:	RST 6	0000 0110
	ŁAD N	011 01101		RST 2	0000 0010
Rozkaz:	<b>POB Tab</b>	100 10000		RST -4	1111 1100
	DOD Suma	001 01100		RST 0	0000 0000
	ŁAD Suma	011 01100		RST -3	1111 1101
	POB Rozkaz	100 00100	Jeden:	RST 1	0000 0001
	DOD Jeden	001 10011			
	ŁAD Rozkaz	011 00100			
	SOB Pętla	101 00000			
Koniec:	STP	111 00000			

# Podprogramy

- ▶ Pozwalają na modularne pisanie programów
- ▶ Wymagają specjalnych rozkazów
- ▶ Przekazywanie parametrów:
  - ▶ Poprzez rejesty procesora
  - ▶ We wskazanym (zwykle zawartością rejestrów) obszarze pamięci
  - ▶ Z wykorzystaniem stosu

# Uzupełnienia konstrukcyjne



# Rozkazy wspierające obsługę podprogramów

- ▶ Skok do podprogramu **SDP** adres  
 $(WS) - 1 \rightarrow WS, (L) \rightarrow (WS), (AD) \rightarrow L, A$
- ▶ Powrót z podprogramu **PWR**  
 $((WS)) \rightarrow L, A, (WS) + 1 \rightarrow (WS)$
- ▶ Zapis zawartości akumulatora na stos **DNS**  
 $(WS) - 1 \rightarrow WS, (AK) \rightarrow (WS)$
- ▶ Odczyt ze stosu **PZS**  
 $((WS)) \rightarrow AK, (WS) + 1 \rightarrow (WS)$

## Przykład 4

- ▶ Dane dwie zmienne umieszczone w komórkach pamięci x i y
- ▶ Policzyć sumę ich wartości bezwzględnych
- ▶ Wynik umieścić w zmiennej z

$$z = |x| + |y|$$

# Użycie podprogramu w programie

POB X

**SDP Abs**

ŁAD Z

POB Y

**SDP Abs**

DOD Z

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

**POB X**

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

SOM Ujemna

PWR

Ujemna:

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

Pomocnicza: RPA

X: RST -2

Y: RST 3

Z: RPA

# Użycie podprogramu w programie

POB X  
**SDP Abs**  
ŁAD Z  
POB Y  
SDP Abs  
DOD Z  
ŁAD Z  
STP



Abs:

Ujemna:

Pomocnicza: RPA

X: RST -2

Y: RST 3

Z: RPA

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

**Abs:**

Ujemna:

Pomocnicza: RPA

X: RST -2

Y: RST 3

Z: RPA

**SOM Ujemna**

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR



# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

**Ujemna:**

Pomocnicza: RPA

X: RST -2

Y: RST 3

Z: RPA

SOM Ujemna

PWR

**ŁAD Pomocnicza**

ODE Pomocnicza

ODE Pomocnicza

PWR

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

**ODE Pomocnicza**

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

**ODE Pomocnicza**

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X  
SDP Abs  
ŁAD Z  
POB Y  
SDP Abs  
DOD Z  
ŁAD Z  
STP

Abs:

Ujemna:

Pomocnicza: RPA  
X: RST -2  
Y: RST 3  
Z: RPA

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

**PWR**

# Użycie podprogramu w programie

POB X

SDP Abs

**ŁAD Z**

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

**POB Y**

SDP Abs

DOD Z

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

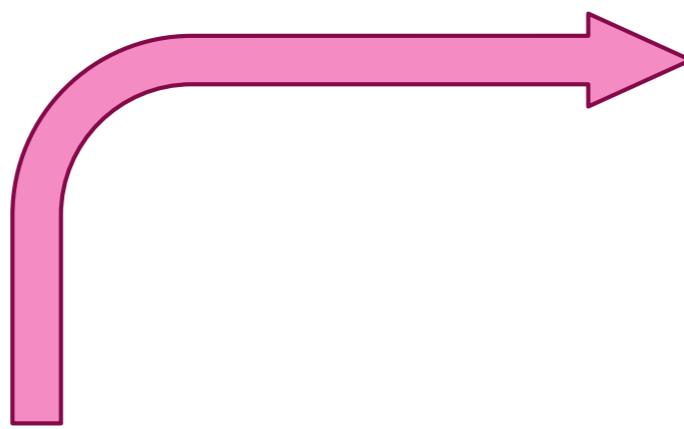
RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X  
SDP Abs  
ŁAD Z  
POB Y  
**SDP Abs**  
DOD Z  
ŁAD Z  
STP



Abs:  
Ujemna:  
Pomocnicza:  
X:  
Y:  
Z:

SOM Ujemna  
PWR  
ŁAD Pomocnicza  
ODE Pomocnicza  
ODE Pomocnicza  
PWR  
RPA  
RST -2  
RST 3  
RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

STP

**Abs:**

Ujemna:

Pomocnicza: RPA

X: RST -2

Y: RST 3

Z: RPA

**SOM Ujemna**

PWR

ŁAD Pomocnicza

ODE Pomocnicza

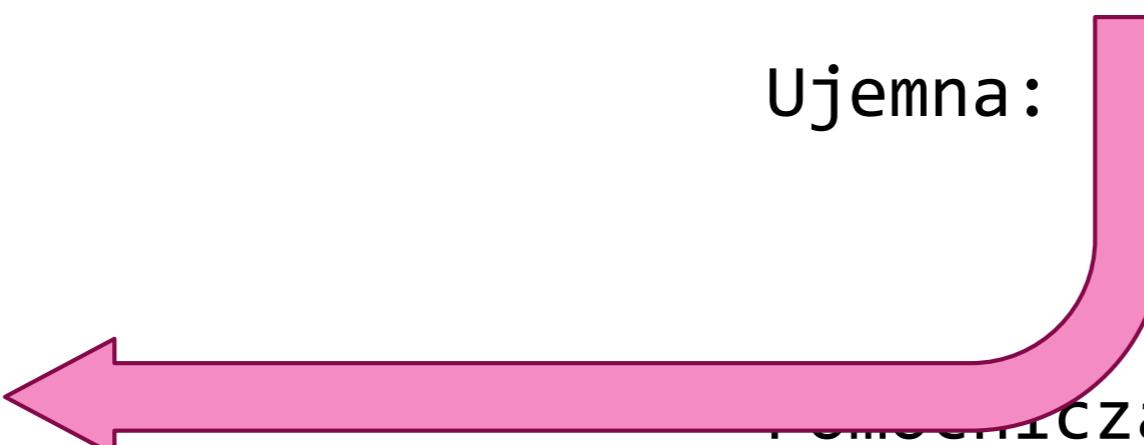
ODE Pomocnicza

PWR

# Użycie podprogramu w programie

POB X  
SDP Abs  
ŁAD Z  
POB Y  
SDP Abs  
DOD Z  
ŁAD Z  
STP

Abs:  
Ujemna:  
Pomocnicza:  
X:  
Y:  
Z:



SOM Ujemna  
**PWR**  
ŁAD Pomocnicza  
ODE Pomocnicza  
ODE Pomocnicza  
PWR  
RPA  
RST -2  
RST 3  
RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

**DOD Z**

ŁAD Z

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

**ŁAD Z**

STP

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Użycie podprogramu w programie

POB X

SDP Abs

ŁAD Z

POB Y

SDP Abs

DOD Z

ŁAD Z

**STP**

Abs:

Ujemna:

Pomocnicza:

X:

Y:

Z:

SOM Ujemna

PWR

ŁAD Pomocnicza

ODE Pomocnicza

ODE Pomocnicza

PWR

RPA

RST -2

RST 3

RPA

# Przekazywanie parametrów przez stos

- ▶ Przed wywołaniem podprogramu należy zapisać jego aktualne parametry na stosie w ustalonej kolejności
- ▶ W końcowej części wywołania podprogramu należy parametry usunąć ze stosu
  - ▶ Parametry usuwa podprogram wywoływany
  - ▶ Po zakończeniu podprogramu program wywołujący usuwa parametry

# Przekazywanie parametrów przez stos – przykład

```
POB X
DNS
SDP Abs
ŁAD Z
POB Y
DNS
SDP Abs
DOD Z
ŁAD Z
STP
X: RST -2
Y: RST 3
Z: RPA
```

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	<b>POB X</b>	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

POB X	Abs:	PZS
<b>DNS</b>		ŁAD ślad
SDP Abs		PZS
ŁAD Z		ŁAD parametr
POB Y		SOM Ujemna
DNS	Ujemna:	SOB Powrót
SDP Abs		ODE Parametr
DOD Z		ODE Parametr
ŁAD Z	Powrót:	ŁAD Parametr
STP		POB ślad
X: RST -2		DNS
Y: RST 3		POB Parametr
Z: RPA	ślad:	PWR
		-2
	Parametr:	RST 3
		RPA

# Przekazywanie parametrów przez stos – przykład

X:	POB X	Abs:	PZS	
	DNS		ŁAD ślad	
	<b>SDP Abs</b>		PZS	
	ŁAD Z		ŁAD parametr	
	POB Y		SOM Ujemna	
	DNS	Ujemna:	SOB Powrót	
	SDP Abs		ODE Parametr	
	DOD Z		ODE Parametr	
	ŁAD Z	Powrót:	ŁAD Parametr	
	STP		POB ślad	
	X: RST -2		DNS	
	Y: RST 3		POB Parametr	Ślad (3)
	Z: RPA	ślad:	PWR	-2
			RST 3	
		Parametr:	RPA	

# Przekazywanie parametrów przez stos – przykład

	POB X	<b>Abs:</b>	<b>PZS</b>	
	DNS		ŁAD ślad	
	SDP Abs		PZS	
	ŁAD Z		ŁAD parametr	
	POB Y		SOM Ujemna	
	DNS	<b>Ujemna:</b>	SOB Powrót	
	SDP Abs		ODE Parametr	
	DOD Z		ODE Parametr	
	ŁAD Z	<b>Powrót:</b>	ŁAD Parametr	
	STP		POB ślad	
X:	RST -2		DNS	
Y:	RST 3		POB Parametr	Ślad (3)
Z:	RPA	<b>ślad:</b>	PWR	-2
			RST 3	
		<b>Parametr:</b>	RPA	

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS	Ujemna:	SOB Powrót
	SDP Abs		ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			-2
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		<b>PZS</b>
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			-2
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		<b>ŁAD parametr</b>
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		<b>SOM Ujemna</b>
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

POB X	Abs:	PZS
DNS		ŁAD ślad
SDP Abs		PZS
ŁAD Z		ŁAD parametr
POB Y		SOM Ujemna
DNS	<b>Ujemna:</b>	SOB Powrót
SDP Abs		<b>ODE Parametr</b>
DOD Z		ODE Parametr
ŁAD Z	<b>Powrót:</b>	ŁAD Parametr
STP		POB ślad
X: RST -2		DNS
Y: RST 3		POB Parametr
Z: RPA	<b>ślad:</b>	PWR
		RST 3
	<b>Parametr:</b>	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		<b>ODE Parametr</b>
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z		<b>ŁAD Parametr</b>
	STP	Powrót:	POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z		ŁAD Parametr
	STP	Powrót:	POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

POB X	Abs:	PZS
DNS		ŁAD ślad
SDP Abs		PZS
ŁAD Z		ŁAD parametr
POB Y		SOM Ujemna
DNS	Ujemna:	SOB Powrót
SDP Abs		ODE Parametr
DOD Z		ODE Parametr
ŁAD Z	Powrót:	ŁAD Parametr
STP		POB ślad
X: RST -2		<b>DNS</b>
Y: RST 3		POB Parametr
Z: RPA	ślad:	PWR
		RST 3
	Parametr:	RPA

Ślad (3)

# Przekazywanie parametrów przez stos – przykład

POB X	Abs:	PZS
DNS		ŁAD ślad
SDP Abs		PZS
ŁAD Z		ŁAD parametr
POB Y		SOM Ujemna
DNS	Ujemna:	SOB Powrót
SDP Abs		ODE Parametr
DOD Z		ODE Parametr
ŁAD Z	Powrót:	ŁAD Parametr
STP		POB ślad
X: RST -2		DNS
Y: RST 3		<b>POB Parametr</b>
Z: RPA	ślad:	PWR
		RST 3
	Parametr:	RPA

Ślad (3)

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	ŁAD Z		ŁAD parametr
	POB Y		SOM Ujemna
	DNS	Ujemna:	SOB Powrót
	SDP Abs		ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	<b>PWR</b>
		Parametr:	RST 3
			Ślad (3)
		Parametr:	RPA

# Przekazywanie parametrów przez stos – przykład

	POB X	Abs:	PZS
	DNS		ŁAD ślad
	SDP Abs		PZS
	<b>ŁAD Z</b>		ŁAD parametr
	POB Y		SOM Ujemna
	DNS		SOB Powrót
	SDP Abs	Ujemna:	ODE Parametr
	DOD Z		ODE Parametr
	ŁAD Z	Powrót:	ŁAD Parametr
	STP		POB ślad
X:	RST -2		DNS
Y:	RST 3		POB Parametr
Z:	RPA	ślad:	PWR
			RST 3
		Parametr:	RPA

# Podsumowanie

- ▶ Klasyfikacja języków programowania
- ▶ Przykłady zapisu prostych programów w języku binarnym
- ▶ Język asemblera i przykłady programów
- ▶ Składnia instrukcji asemblera
- ▶ Algorytm pracy asemblera
- ▶ Przetwarzanie tablic
- ▶ Podprogramy i przekazywanie parametrów