

## 9.1 Podstawowe struktury procesora

### 1. Wprowadzenie

Architektura komputerów, a dokładniej architektura systemów komputerowych, to inaczej organizacja systemów komputerowych, choć czasami czyni się pomiędzy tymi pojęciami rozróżnienie. Architektura to według Stallingsa spojrzenie na konstrukcję systemu cyfrowego od strony programisty, a organizacja to bardziej punkt widzenia konstruktora systemu.

Czasami wygodniej jest mówić o architekturze czy organizacji systemu cyfrowego zamiast o architekturze komputerów. Pojęcie systemu cyfrowego jest bowiem nieco ogólniejsze od pojęcia systemu komputerowego.

Dla opisu architektury systemu cyfrowego posługujemy się na ogół schematami czyli rysunkami utrzymanymi w pewnej konwencji, które odzwierciedlają wzajemne połączenia i relacje pomiędzy fragmentami systemu cyfrowego. Z reguły też uzupełniamy taki schemat opisem słownym lub opisem sformalizowanym działania systemu.

**Uwaga:** Często też stosujemy do opisu architektury systemów cyfrowych specjalne języki formalne nazywane językami opisu sprzętu lub językami HDL (od ang. Hardware Description Language). Typowymi językami tego typu są VHDL, Verilog czy Abel. Języki HDL szczególnie użyteczne są w programach CAD (ang. Computer Aided Design) wspomagających projektowanie systemów cyfrowych. Programy takie umożliwiają symulację i weryfikację sposobu działania systemu cyfrowego na podstawie wprowadzonego opisu.

### 2. Architektura von Neumanna

Większość współczesnych komputerów (a dokładniej procesorów) ma architekturę stanowiącą pewną modyfikację podstawowej koncepcji von Neumanna. Procesor o *architekturze von Nuemanna* (mówimy także *organizacja von Nuemanna* lub *maszyna von Nuemanna* - ang. *von Neumann machine*) organizacja składa się z:

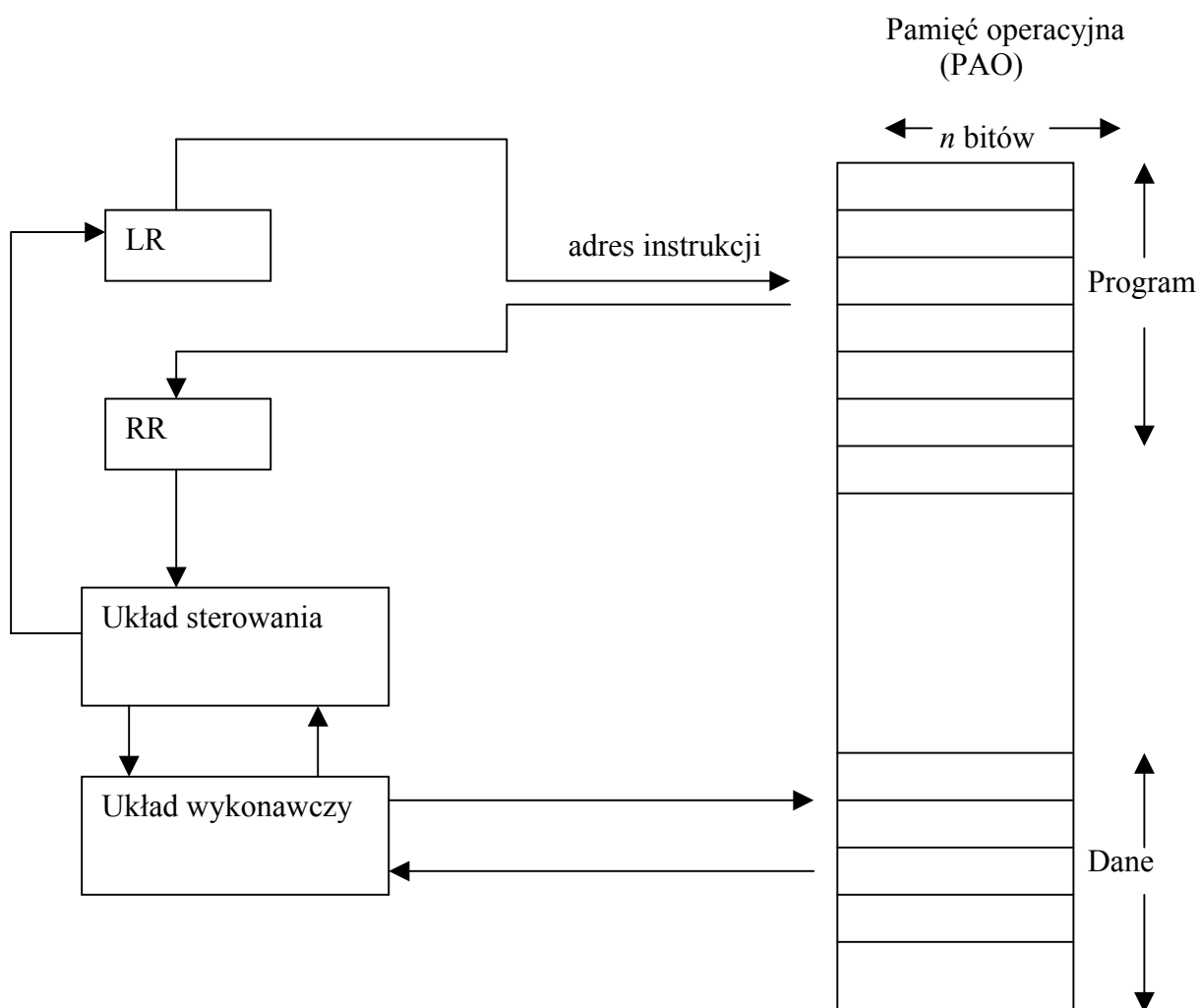
- pamięci operacyjnej PAO (składającej się z komórek o długości  $n$ )
- licznika rozkazów LR (ang. *Program Counter* lub *Program Pointer*, stosowane skróty to PC lub PP)
- rejestru rozkazów RR (ang. *Instruction Register*, stosowany skrót to IR)
- części wykonawczej (część wykonawcza realizuje rozkazy m.in. rozkazy arytmetyczne i logiczne np. rozkazy dodawania, odejmowania itd)
- części sterującej (interpretującej rozkazy)

Podstawowym pojęciem jest *rozkaz, instrukcja* (ang. *instruction*) czyli elementarne polecenie. Lista wszystkich elementarnych poleceń nazywa się *listą rozkazów* (ang. *instruction set*) lub *listą instrukcji*. Pojęć rozkaz i instrukcja będziemy w dalszym ciągu używali wymiennie. Czasami mówimy instrukcja maszynowa lub rozkaz maszynowy dla podkreślenia, że jest to instrukcja w powyższym sensie i dla odróżnienia jej od instrukcji w języku wysokiego poziomu.. Elementarne polecenie jest zapisywane w postaci słowa binarnego i po przesłaniu do rejestru rozkazów RR jest interpretowane i wykonywane.

Ciąg rozkazów („sensowny ciąg rozkazów”, „ciąg rozkazów realizujący pewien algorytm”) nazywamy *programem*. Podstawowym celem działania procesora jest wykonanie programu zapisanego w pamięci na danych również zapisanych w tej samej pamięci. Krok po kroku wykonujemy instrukcje stanowiące program.

Procesor działa w *cyklach rozkazowych*. W jednym cyklu rozkazowym procesor wykonuje jedną instrukcję. Rozkazy są pobierane do wykonania po kolei zgodnie z adresem wskazywanym przez licznik rozkazów LR. Na początku cyklu rozkazowego zawartość LR jest zwiększana o 1, tzn. dokonujemy podstawienia  $LR := LR + 1$ , po czym z komórki pamięci o adresie zawartym w LR pobierana jest kolejna instrukcja programu. Jednak dopuszczalna jest zmiana kolejności wykonywania instrukcji tzw. rozkazem skoku. Rozkaz skoku sprowadza się do załadowania rejestru LR nową wartością.

Patrząc na słowo zawarte w komórce PAO w zasadzie nie możemy powiedzieć, czy jest to rozkaz czy słowo danych.



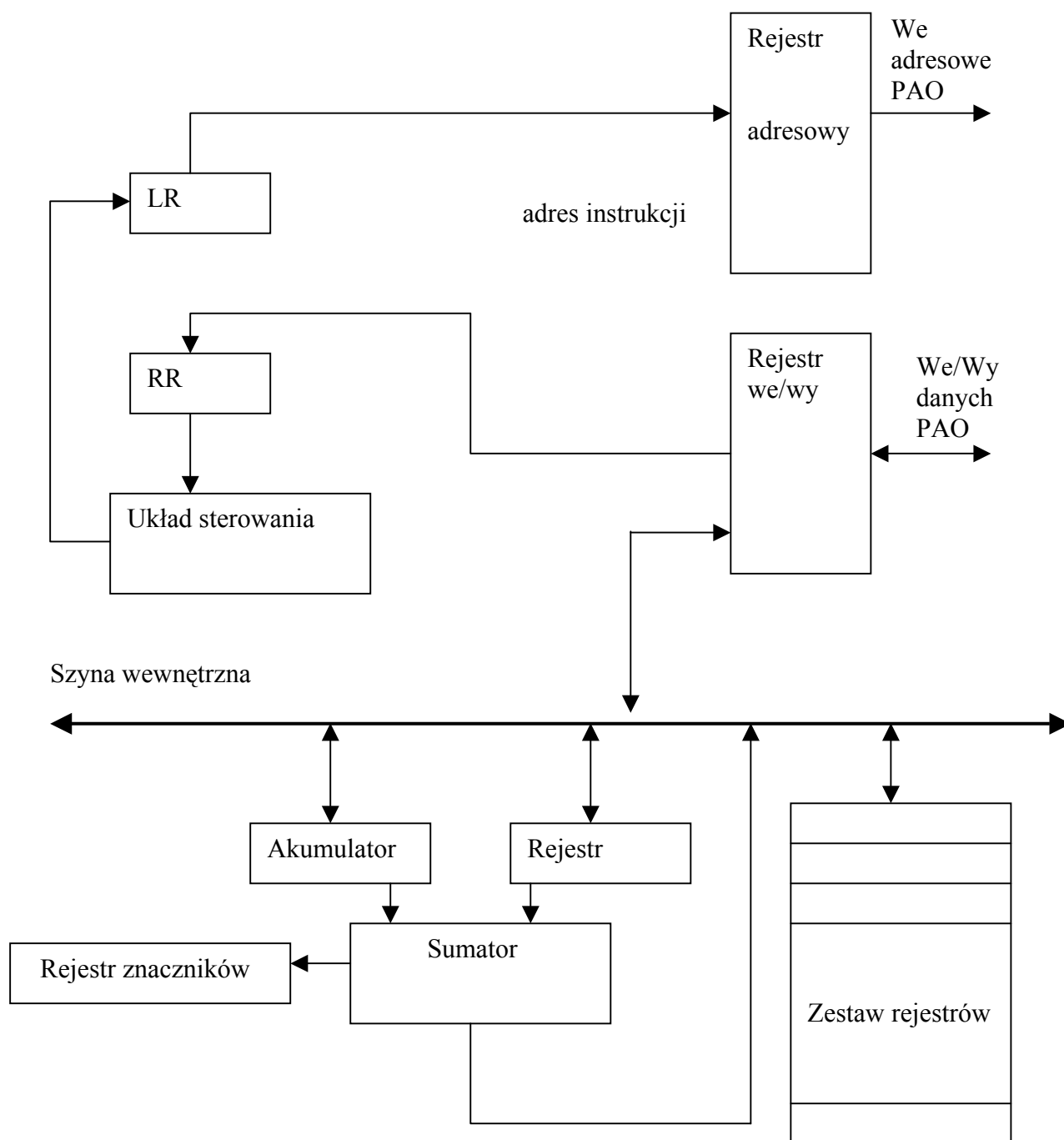
**Rys.1. Architektura von Neumanna; program to ciąg instrukcji**

Architektura von Neumanna ma 3 charakterystyczne cechy

- Dane i rozkazy są pamiętane w tej samej pamięci umożliwiającej zapis i odczyt z dowolnej komórki pamięci

- Wykonanie rozkazów programu następuje kolejno rozkaz po rozkazie (poza sytuacją gdy następuje tzw. skok).
- Patrząc na słowo binarne zawarte w komórce pamięci nie wiemy czy to jest rozkaz czy dane.

Ważnym pojęciem w architekturze komputerów jest *słowo maszynowe* (ang. *machine word*). Jest to typowa dla danego procesora (mikroprocesora) struktura słowa, którym operujemy w systemie. Jeśli słowo maszynowe jest 8-bitowe mówimy, że procesor, mikroprocesor czy system komputerowy jest 8-bitowy, jeśli 32-bitowy, to 32-bitowy itd. Typowe długości słowa maszynowego to 8, 16, 32 i 64 bity.

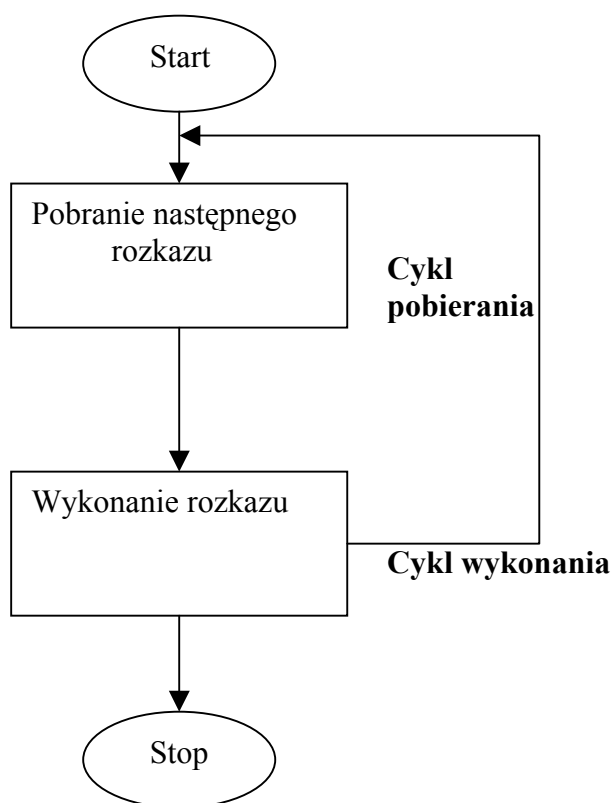


Rys.2. Najprostsza struktura procesora; koncepcja minimum procesora to RR, LR, układ sterowania, zestaw rejestrów, sumator i rejestr stanu

Według przyjętej w opisie architektury von Neumanna definicji, *procesor* to licznik rozkazów LR, rejestr rozkazów RR, układ sterowania, układ wykonawczy i pamięć operacyjna. Często jednak zwłaszcza w systemach mikroprocesorowych nie zaliczamy pamięci operacyjnej do procesora. Procesor wykonany w postaci układu scalonego nazywamy *mikroprocesorem*. Schemat blokowy najprostszego procesora pokazany jest na rys.2. Na ogół procesor jest jednak znacznie bardziej skomplikowany, ponieważ zależy nam na tym by wydajność obliczeniowa, czyli jak mówimy moc obliczeniowa procesora, była jak największa.

Czasami procesor (zwłaszcza w dużych systemach komputerowych) nazywamy *jednostką centralną* lub *CPU* (ang. *Central Processor Unit*).

Z fizycznego punktu widzenia cykl rozkazowy to ciąg skończony taktów zegara, w których procesor wykonuje daną instrukcję. Instrukcje zależnie od ich stopnia skomplikowania mogą potrzebować różnej liczby taktów zegara na realizację. Podstawowy schemat cyklu rozkazowego z podziałem na cykl pobierania i cykl wykonania rozkazu pokazany jest na rys.3. Cykl rozkazowy dzieli się na ogół na tzw. *cykle maszynowe* (ang. *machine cycle*) zawierające pojedyncze odwołania do pamięci.



Rys. 3, Podstawowy cykl rozkazowy

*Pamięć operacyjna* (ang. *operational memory*) lub *pamięć główna* (ang. *main memory*) to pamięć RAM w której przechowywany jest wykonywany program i dane.

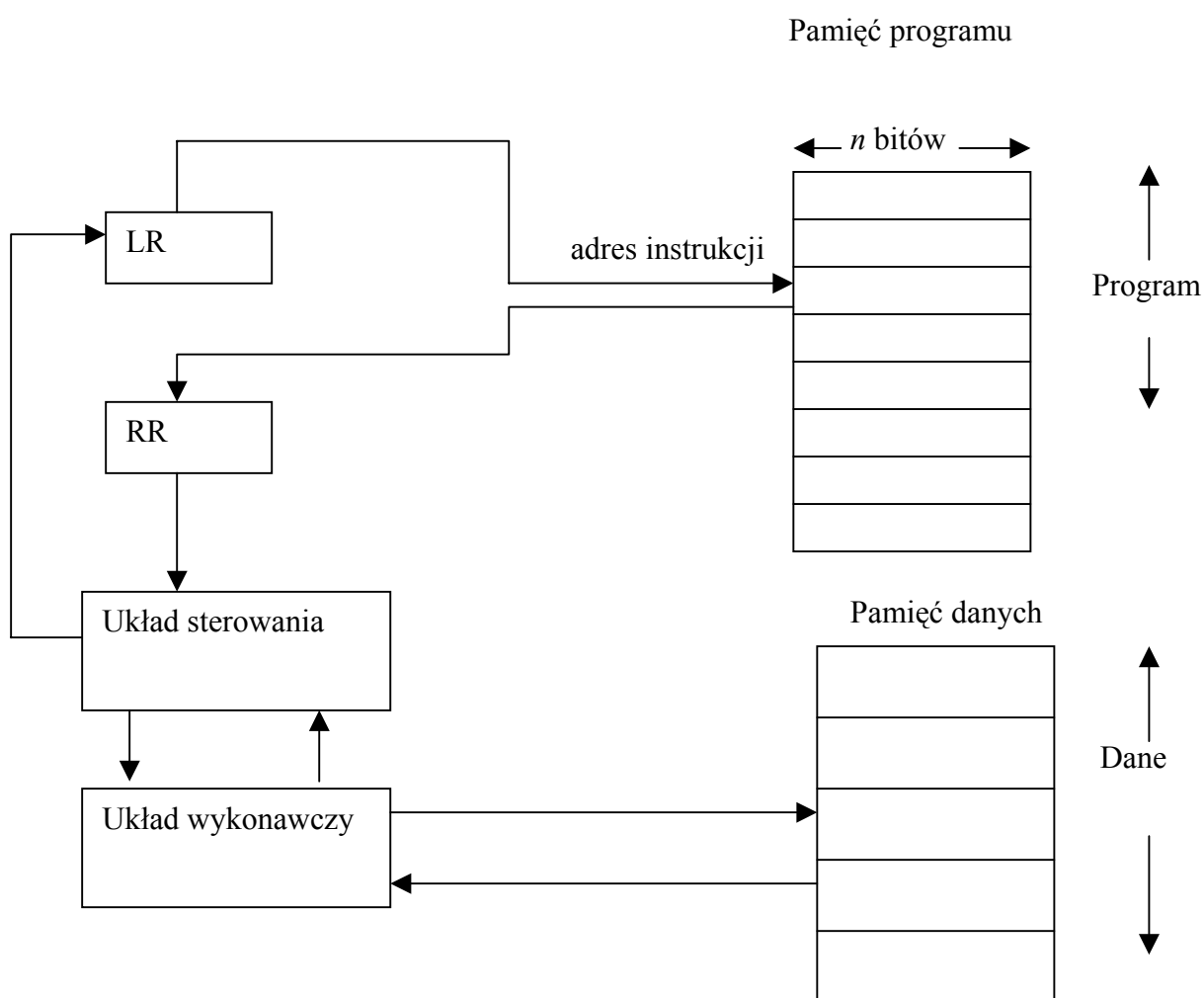
*Język maszynowy* (ang. *machine language*) to zbiór wszystkich słów binarnych interpretowanych jako rozkazy procesora. Programowanie w takim języku, czyli pisanie programów, jest bardzo niewygodne. Żeby ułatwić programiście operowanie instrukcjami stosuje się symboliczny zapis rozkazów.

**Mnemonic** (ang. *mnemonic*) to symboliczny zapis rozkazu. Np. Typowym mnemonikiem dla rozkazu przesłania zawartości rejestru B do rejestru akumulatora A jest: MOV A, B

**Asembler** (ang. *assembler*) to język programowania, w którym każdej instrukcji odpowiada dokładnie jedna instrukcja maszynowa. Asemblerem nazywamy też program tłumaczący programy zapisane w tym języku na język maszynowy.

### 3. Architektura Harvard

Architektura Harvard polega na rozdzieleniu pamięci programu i pamięci danych na dwie niezależnie pracujące pamięci. Takie rozwiązanie umożliwia równoczesne pobieranie rozkazu i przesyłanie argumentów lub wyników wykonania rozkazu, co umożliwia podwyższenie szybkości działania komputera.



Rys.4. Architektura typu Harvard; pamięć danych i pamięć programu są rozdzielone; zaleta: dostęp do nich możemy uzyskiwać równocześnie