Języki asemblerowe

WYKŁAD 2

Dr Krzysztof Balicki

Komponenty systemu komputerowego

- Trzy główne komponenty:
  - Jednostka centralna (CPU) / Procesor;
  - Pamięć;
  - Urządzenia wejścia/wyjścia.

- Komponenty połączone są za pomocą magistrali systemowych.

Komponenty systemu komputerowego

- Szerokość magistrali adresowej wyznacza maksymalną pojemność pamięci, którą może zaadresować procesor.
- Szerokość magistrali danych wyznacza rozmiar danych przesyłanych pomiędzy procesorem , pamięcią i urządzeniami wejścia/wyjścia.
- Przykładowe magistrale:
  - Procesor Pentium: 32 linie adresowe, 64 linie danych
  - Procesor Itanium: 64 linie adresowe, 64 linie danych

Adresy w instrukcjach procesora

- Liczba operandów instrukcji procesora:
  - dwa operandy (instrukcje binarne),
  - jeden operand (instrukcje unarne).
- Liczba rezultatów operacji:
  - jeden rezultat
  - dwa rezultaty (np. operacja dzielenia: iloraz i reszta z dzielenia).
- W przypadku instrukcji binarnych produkujących jeden rezultat w instrukcji podawane są trzy adresy: dwa adresy operandów i jeden adres rezultatu.

Adresy w instrukcjach procesora

- Maszyny trójadresowe (np. procesor MIPS):

instrukcja adresPrzezn adresZrodla1 adresZrodla2

  – instrukcja wykonywana jest na wartościach określonych przez adresy *adresZrodla1* i *adresZrodla2*, miejsce wyniku wykonania instrukcji wyznacza *adresPrzezn*

Adresy w instrukcjach procesora

- Maszyny trójadresowe (np. procesor MIPS) - przykład:
- Wyrażenie: A = B + C * D - E + F + A

Instrukcja	***Obliczenia***
mult T,C,D	T = C*D
add T,T,B	T = B + C*D
sub T,T,E	T = B + C*D - E
add T,T,F	T = B + C*D - E + F
add A,A,T	A = B + C*D - E + F + A

Adresy w instrukcjach procesora

- Maszyny dwuadresowe (np. procesor Pentium):

instrukcja adresPrzezn adresZrodla

  – instrukcja wykonywana jest na wartościach określonych przez adresy *adresPrzezn* i *adresZrodla*, miejsce wyniku wykonania instrukcji wyznacza *adresPrzezn*

Adresy w instrukcjach procesora

- Maszyny dwuadresowe (np. procesor Pentium) - przykład:
- Wyrażenie: A = B + C * D - E + F + A

Instrukcja	***Obliczenia***
load T,C	T = C
mult T,D	T = C*D
add T,B	T = B + C*D
sub T,E	T = B + C*D - E
add T,F	T = B + C*D - E + F
add A,T	A = B + C*D - E + F + A

Adresy w instrukcjach procesora

- Maszyny jednoadresowe

instrukcja adresZrodla

  – instrukcja wykonywana jest na akumulatorze i na wartości określonej przez adres *adresZrodla*, wynik wykonania instrukcji przechowywany jest w akumulatorze

Adresy w instrukcjach procesora

- Maszyny zeroadresowe

instrukcja

  – instrukcja wykonywana jest na wartościach umieszczonych w domyślnej lokalizacji (na stosie), wynik wykonywania instrukcji umieszczany jest również na stosie.

Adresy w instrukcjach procesora

- Maszyny zeroadresowe

instrukcja

  – instrukcja wykonywana jest na wartościach umieszczonych w domyślnej lokalizacji (na stosie), wynik wykonywania instrukcji umieszczany jest również na stosie.

Adresy w instrukcjach procesora

- Architektura *load/store*
  - instrukcje wykonywane są wyłącznie na wartościach umieszczonych w rejestrach wewnętrznych procesora,
  - tylko instrukcje *load* oraz *store* przenoszą dane pomiędzy rejestrami procesora i pamięcią:

load *rejestr* *adresZrodla*

store *adresPrzezn* *rejestr*

Adresy w instrukcjach procesora

- Architektura *load/store* - przykład:
- Wyrażenie: A = B + C * D - E + F + A

Instrukcja	***Obliczenia***
load R1,B	ładowanie B
load R2,C	ładowanie C
load R3,D	ładowanie D
load R4,E	ładowanie E
load R5,F	ładowanie F
load R6,A	ładowanie A

Adresy w instrukcjach procesora

- Architektura *load/store* - przykład (cd.):

Instrukcja	***Obliczenia***
mult R2,R2,R3	R2 = C*D
add R2,R2,R1	R2 = B + C*D
sub R2,R2,R4	R2 = B + C*D - E
add R2,R2,R5	R2 = B + C*D - E + F
add R2,R2,R6	R2 = B + C*D - E + F + A
store A,R2	zapis wyniku

Adresy w instrukcjach procesora

- Architektura *load/store*
  - architektura *load/store* redukuje rozmiar instrukcji, np.:
    - magistrala adresowa: 32 bity,
    - → instrukcja z trzema adresami pamięci: 104 bity (kod operacji: 8 bitów, adresy operandów: 3 × 32 bity),
    - → instrukcja operująca tylko na rejestrach: 23 bity (kod operacji: 8 bitów, kody rejestrów: 3 × 5 bitów).

Rejestry procesora

- Rejestry przechowują dane, instrukcje lub informacje o stanie procesora.

- Klasyfikacja rejestrów ze względu na strukturę i sposób użycia:
  - Rejestry ogólnego przeznaczenia,
  - Rejestry specjalnego przeznaczenia:
    - dostępne dla programisty,
    - dostępne wyłącznie dla systemu.

Rejestry procesora

- Liczba adresów w instrukcjach procesora determinuje liczbę rejestrów procesora:
  - maszyny trój- i dwuadresowe nie potrzebują wewnętrznych rejestrów, jednak dla poprawy wydajności kilka takich rejestrów one posiadają,
  - procesory w architekturze *load/store* posiadają większą liczbę rejestrów.

Obsługa pamięci

- Proces odczytu pamięci:
  1. Ustawienie adresu odczytywanej komórki pamięci na magistrali adresowej.
  2. Aktywacja sygnału odczytu pamięci na magistrali sterującej.
  3. Oczekiwanie na dane z zaadresowanej komórki pamięci na magistrali danych.
  4. Odczyt danych z magistrali danych.
  5. Dezaktywacja sygnału odczytu pamięci na magistrali sterującej.

Obsługa pamięci

- Przykładowo, procesor Pentium potrzebuje trzy cykle maszynowe do realizacji procesu odczytu pamięci:
  - cykl 1: kroki 1 i 2,
  - cykl 2: krok 3,
  - cykl 3: kroki 4 i 5.

Obsługa pamięci

- Proces zapisu pamięci:
  - ustawienie adresu zapisywanej komórki pamięci na magistrali adresowej
  - wysłanie danych zapisywanych w pamięci na magistralę danych
  - aktywacja sygnału zapisu pamięci na magistrali sterującej
  - oczekiwanie na zapisanie przez pamięć danych w zaadresowanej komórce pamięci
  - dezaktywacja sygnału zapisu pamięci na magistrali sterującej

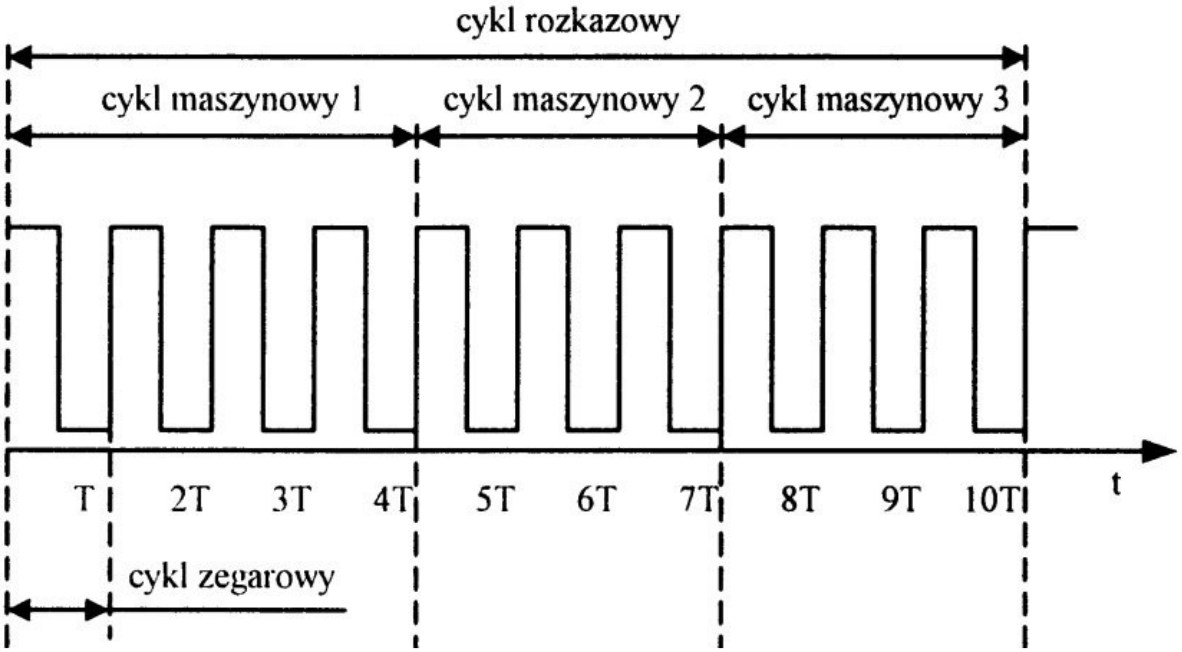
Obsługa pamięci

- Przykładowo, procesor Pentium potrzebuje trzy cykle maszynowe do realizacji procesu zapisu pamięci:
  - cykl 1: kroki 1 i 3,
  - cykl 2: krok 2,
  - cykl 3: krok 4 i 5.

Przechowywanie wielobajtowych danych

- Porządek *Little-endian*: w pierwszej komórce przechowywany jest najmniej znaczący bajt (LSB), w ostatniej komórce - najbardziej znaczący bajt (MSB).
- Porządek *Big-endian*: w pierwszej komórce przechowywany jest najbardziej znaczący bajt (MSB), w ostatniej komórce - najmniej znaczący bajt (LSB).

Cykl rozkazowy procesora



Cykl maszynowy obejmuje kilka cykli zegarowych i może mieć różne długości zależnie od rodzajów wykonywanych operacji.
Cykl rozkazowy obejmuje od jednego do kilku cykli maszynowych, zależnie od rodzaju rozkazu.

Zbiór instrukcji procesora

- ISA (*Instruction Set Architecture*) - zbiór instrukcji (rozkazów) procesora definiuje logikę procesora specyfikując, które instrukcje (rozkazy) mogą być wykonywane przez procesor.

Procesory

- CISC (*Complex Instruction Set Computer*) - procesory o złożonej liście rozkazów.

- RISC (*Reduced Instruction Set Computer*) - procesory o zredukowanej liście rozkazów.

Procesory CISC

- Złożona lista rozkazów.
- Złożone rozkazy.
- Zmienna długość instrukcji (rozkazów).
- Złożone typy danych.
- Wiele trybów adresowania.
- Mała liczba rejestrów wewnętrznych.

Procesory CISC

Procesory CISC

- Przykłady:
  - VAX
  - Pentium
    - *generalnie procesory rodziny x86 (pod względem mikroarchitektury niektóre procesory mają jednak wiele cech procesora RISC)*

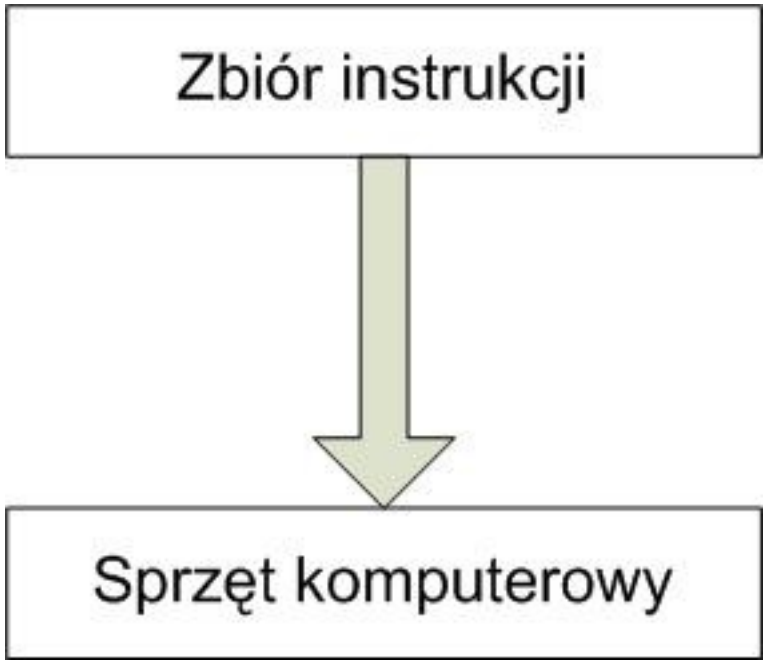
Procesory RISC

- Zredukowana lista rozkazów.
- Proste rozkazy wykonywane w jednym cyklu maszynowy.
- Stała długość instrukcji (rozkazów) - prosty format instrukcji.
- Proste typy danych.
- Proste tryby adresowania (adresowanie rejestrów, tylko operacje *load* i *store* wykorzystują adresowanie pamięci).
- Wykorzystywana architektura *load/store*.

Procesory RISC

- Duża liczba rejestrów wewnętrznych.
- Większość procesorów RISC wykorzystuje architekturę Harvard (oddzielna pamięć danych, oddzielna pamięć programu).

Procesory RISC



Procesory RISC

- Przykłady:
  - PowerPC
  - MIPS
  - Itanium
  - Atmel AVR
  - AMD 29000
  - Motorola M88000

Porównanie procesorów CISC i RISC

- Przykład instrukcji procesora CISC:

add [R2], R3

- Przykład instrukcji procesora RISC:

load R4, [R2]

add R4, R4, R3

store [R2], R4

Porównanie procesorów CISC i RISC

Typ procesora	CISC		RISC
Cecha	VAX 11/780	Intel 486	MIPS R4000
Liczba instrukcji	303	235	94
Liczba trybów adresowania	22	11	1
Rozmiar instrukcji [B]	2–57	1–12	4
Liczba rejestrów ogólnego przeznaczenia	16	8	32

Architektury mieszane

- Współczesne procesory zgodne z x86 przetwarzają rozkazy procesora x86 na proste mikropolecenia pracujące wg idei RISC.