Politechnika Wrocławska

Architektura Systemów Komputerowych

Wykład 5

Dr inż. Radosław Michalski
Katedra Inteligencji Obliczeniowej, Wydział Informatyki i Zarządzania
Politechnika Wrocławska

Wersja 1.1, wiosna 2018

Źródła i licencja

Najbardziej aktualna wersja tego wykładu znajduje się tu:
https://github.com/rmhere/lecture-comp-arch-org

Zawartość tego wykładu

Architektura zestawu instrukcji

Architektura - rozważania

Jak działa procesor?

Architektura zestawu instrukcji

Wprowadzenie

Architektura zestawu instrukcji komputera (ang. *instruction set architecture*) definiuje komponenty i operacje widoczne z punktu widzenia programisty. Ta specyfikacja to zatem interfejs pomiędzy sprzętem a programistą, gdyż do pracy z daną architekturą nic więcej wiedzieć nie musimy.

Architektura zestawu instrukcji

Co definiuje ISA?

- organizacja pamięci
  - przestrzeń adresowa - ile lokalizacji pamięci można adresować
  - rozmiar słowa - ile bitów można odczytać z adresu
- rejestry
  - ile
  - jakiego rozmiaru
  - jakiego przeznaczenia
  - jak z nich korzystać
- zbiór instrukcji
  - kody operacji / opcodes (operation selection codes)
  - typy danych (bajt czy słowo)
  - adresowanie

Architektura zestawu instrukcji

Zbiór instrukcji - c.d.

- liczba operacji
- rodzaje operacji
- liczba argumentów
- lokalizacja argumentów
- typ argumentów
- sposób podawania argumentów
- format instrukcji
- liczba formatów instrukcji

Architektura zestawu instrukcji

Operacje

- obsługa danych i interakcja z pamięcią
  - ustawianie rejestru na stałą wartość
  - kopiowanie danych z pamięci do rejestrów i na odwrót
  - odczyt i zapis danych z urządzeń zewnętrznych
- operacje artytmetyczne i logiczne
  - dodawanie, odejmowanie, mnożenie i dzielenie
  - inkrementacja, dekrementacja
  - operacje logiczne (AND, OR, NOT etc.)
  - porównania
- operacje sterowania przepływem
  - rozgałęzienia (ang. *branching*)
  - wywołania (ang. *calling*)

Architektura zestawu instrukcji

Typowa instrukcja

$$C = A + B$$

(argument docelowy) = (argument źródłowy) (operacja) (argument źródłowy)

Architektura zestawu instrukcji

Argumenty

- gdzie są przechowywane
  - rejestry, pamięć, stos, akumulator
- ile argumentów
  - 0, 1, 2, 3
- odwołania do argumentów
  - bezpośrednie, niebezpośrednie, stała
- typy i rozmiary argumentów
  - byte, int, float, double, string, vector

Architektura zestawu instrukcji

Lokalizacja argumentów - stos

Stos

- argumenty znajdują się na stosie (domniemane)
- zalety: prosta ewaluacja wyrażeń (odwrotna notacja polska), krótsze insrukcje
- wady: brak losowego dostępu do stosu

C = A + B

- PUSH A
- PUSH B
- ADD
- POP C

Architektura zestawu instrukcji

Lokalizacja argumentów - akumulator

Akumulator

- akumulator jest rejestrem, który pełni rolę argumentu i zapisuje wyniki
- zalety: krótkie instrukcje
- wady: duża wymiana z pamięcią, ponieważ akumulator jest tymczasowy

C = A + B

- LOAD A
- ADD B
- STORE C

Architektura zestawu instrukcji

Lokalizacja argumentów - rejestry ogólnego przeznaczenia

Rejestry ogólnego przeznaczenia

- ang. *general purpose registers* - GPRs
- wszystkie argumenty podane są wprost, jako nazwy rejestrów lub lokalizacje pamięci
- zalety: łatwa generacja kodu, możliwość dłuższego przechowywania danych
- wady: dłuższe instrukcje

C = A + B

- LOAD R1, A
- ADD R2, R1, B
- STORE R2, C

Architektura zestawu instrukcji

GPR w praktyce

Ile argumentów może być adresowanych z pamięci?

- **rejestr - rejestr**: ADD R3, R1, R2
- **rejestr - pamięć**: ADD R1, A
- **pamięć - pamięć**: ADD C, A, B

Architektura zestawu instrukcji

Podawanie argumentów

Bezpośrednie:

- argumenty w rejestrach
- add R1, R4, R3

Poprzez stałą:

- argumenty podawane wprost
- add R1, R2, 5

Baza (przesunięcie):

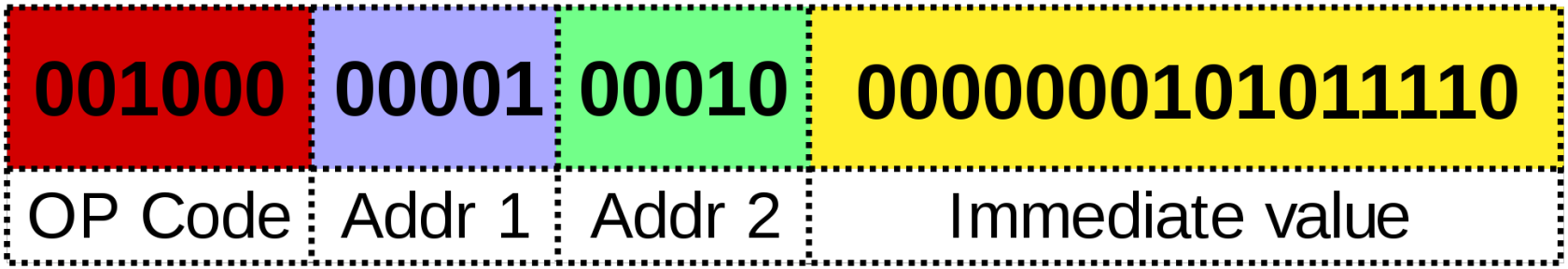
- adres argumentu jest obliczany na podstawie stałej i wartości w rejestrze
- add R3, R2, 100(R1)

Są także inne tryby, ale zwróć uwagę, że nie każda ISA implementuje wszystkie.

Architektura zestawu instrukcji

Format instrukcji

MIPS32 Add Immediate Instruction



Equivalent mnemonic: **addi $r1, $r2, 350**

Architektura zestawu instrukcji

Instrukcje - rozważania

- stała długość
  - potrzeba więcej instrukcji
  - szybsze, bez dekodowania
  - Alpha, ARM, MIPS, PowerPC
- zmienna długość
  - większa elastyczność
  - wolniejsze, wymagają dekodowania
  - VAX, Intel 80x86
- model mieszany
  - IBM 360/70, ARM

Architektura zestawu instrukcji

Źródła i polecane materiały

- M. Martin, *"Introduction to Computer Architecture"*, University of Pennsylvania, PA, USA (materiały do kursu)
- H. Jiang, *"Computer Architecture"*, University of Nebraska-Lincoln, NE, USA (materiały do kursu)
- E. MacDonald, *"Microprocessors"*, University of Texas at El Paso, TX, USA (materiały do kursu)

Architektura - rozważania

Wprowadzneie

- front-endem procesora jest jego ISA
- szczegóły w jaki sposób instrukcja jest wykonywana nie muszą być jawne
- wiele implementacji pojedynczej ISA
- przewaga konkurencyjna uzyskiwana poprzez sposób wykonywania instrukcji
  - szybkość
  - zużycie energii
  - niezawodność

Architektura - rozważania

Rodzaje procesorów

Rodzaje procesorów:

- procesory ogólnego przeznaczenia
- procesory specjalizowane, koprocesory:
  - GPU - graphics processing unit (w kartach graficznych)
  - offload TCP (w kartach sieciowych/płytach głównych)
  - akceleratory kryptograficzne

Architektura - rozważania

Optymalizacja zużycia energii

W jaki sposób zmniejszyć zużycie energii?

- redukcja napięcia
- zmniejszenie częstotliwości
- clock gating
- redukcja przełączania

Architektura - rozważania

Od watów do pikowatów

Ile energii zużywa procesor?

- Intel® Core™ i7-6800K - 140W
- Intel® Core™ i7-4600M - 37W
- ARM® Cortex® A9 - 1.9W
- M^3 - 900pW

Ile energii zużywa Twój procesor?

- sprawdź specyfikację procesora swojego komputera PC, laptopa, telefonu komórkowego, czy tabletu.

Architektura - rozważania

Złożoność ISA

Poszukiwania kompromisu:

- więcej złożonych instrukcji
- proste instrukcje (np. stała długość)
- więcej cykli zegara / mniej cykli zegara

Jednak czy złożoność ma swoje dolne/górne granice?

Architektura - rozważania

Wprowadzenie do architektury RISC

Akronimy:

- CISC - complex instruction set computing
- RISC - reduced instruction set computing

Do pojawienia się RISC, typowym podejściem było CISC:

- złożone instrukcje
- więcej cykli zegara do realizacji operacji

Architektura - rozważania

Źródła i polecane materiały

- IBM Icons of Progress, *"RISC Architecture"*, IBM (artykuł)
- J.C. Scott, But How Do It Know? - The Basic Principles of Computers for Everyone, 2009 (książka i powiązane filmy)

Jak działa procesor?

Wprowadzenie

Materiał wideo

In One Lesson - from Curiosity to Clarity
How a CPU Works

Slajd końcowy

Pytania? Komentarze?

Jeśli masz pomysł jak poprawić lub wzbogacić te wykłady,
proszę zgłoś to jako issue w tym repozytorium:

https://github.com/rmhere/lecture-comp-arch-org