

Technika cyfrowa 2

wykład 2

Procesor - mikroprocesor - mikrokontroler

procesor - podstawowe urządzenie komputera cyfrowego, składające się z jednostki sterującej i arytmometru, w działaniu wykorzystujące rozkazy (*Leksykon Naukowo-Techniczny. WNT, Warszawa 1984*)

mikroprocesor - procesor charakteryzujący się **szczególnie małymi wymiarami** dzięki zastosowaniu elementów mikroelektronicznych, tj. układów scalonych (*Leksykon Naukowo-Techniczny. WNT, Warszawa 1984*)

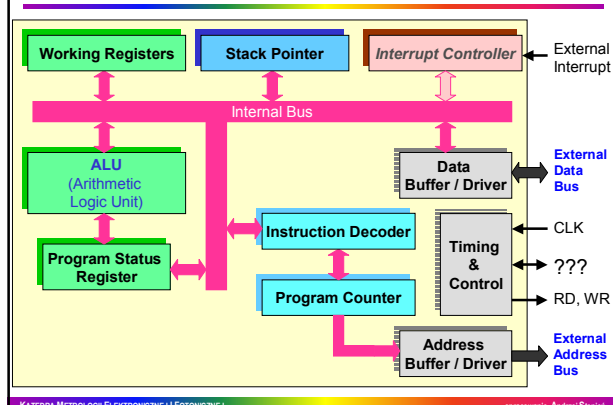
programowalny układ cyfrowy z ograniczoną listą instrukcji, służący do przetwarzania informacji w zadany przez użytkownika sposób

mikrokontroler (mikrokomputer / procesor jednoukładowy) - układ scalony zawierający:
mikroprocesor
pamięć programu (*nie zawsze*) i pamięć danych
urządzenia peryferyjne (liczniki, porty, przetworniki itp.)

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

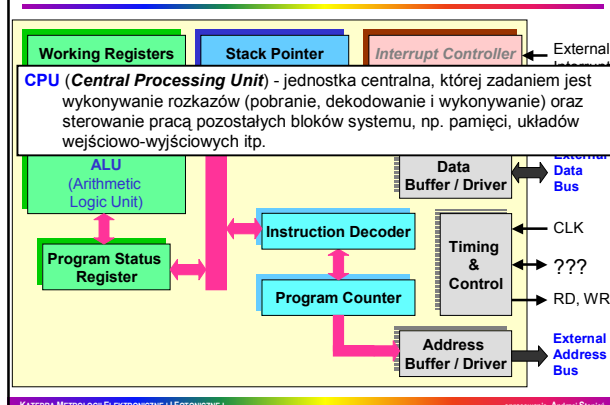
Co powinien zawierać mikroprocesor



KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

CPU

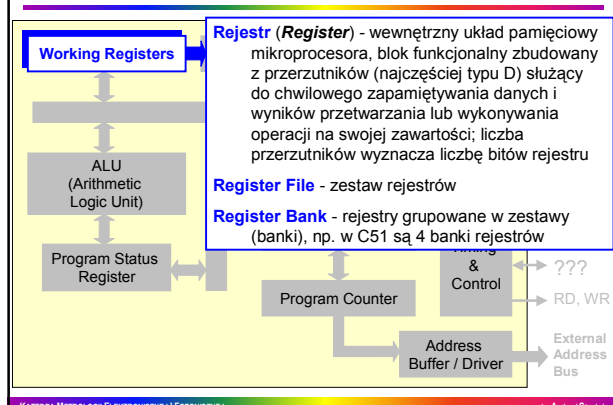


CPU (Central Processing Unit) - jednostka centralna, której zadaniem jest wykonywanie rozkazów (pobranie, dekodowanie i wykonywanie) oraz sterowanie pracą pozostałych bloków systemu, np. pamięci, układów wejściowo-wyjściowych itp.

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Register



Rejestr (Register) - wewnętrzny układ pamięciowy mikroprocesora, blok funkcjonalny zbudowany z przerzutników (najczęściej typu D) służący do chwilowego zapamiętywania danych i wyników przetwarzania lub wykonywania operacji na swojej zawartości; liczba przerzutników wyznacza liczbę bitów rejestru

Register File - zestaw rejestrów

Register Bank - rejestry grupowane w zestawy (banki), np. w C51 są 4 banki rejestrów

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

ALU



ALU (Arithmetic Logic Unit) - układ arytmetyczno-logiczny, przewidziany do wykonywania operacji arytmetycznych i logicznych, układ będący fragmentem jednostki centralnej CPU

funkcje arytmetyczne - suma, różnica, iloczyn, iloraz (INT/DIV, MOD; zmienne: całkowito-liczbowe, ułamkowe w zapisie stało- i zmiennie-przecinkowym), sumy iloczynów cząstkowych, negacja arytmetyczna (w kodzie U2)

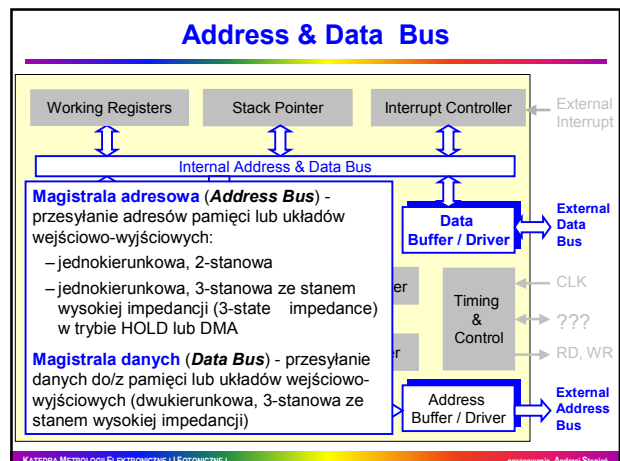
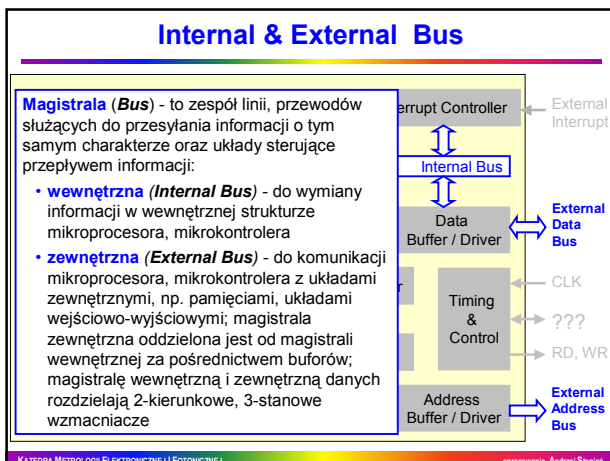
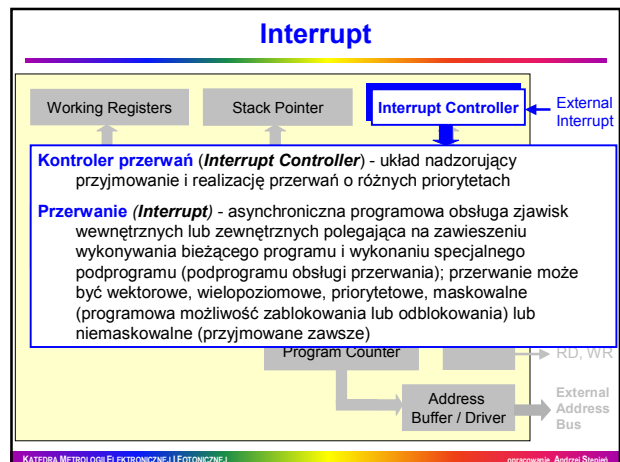
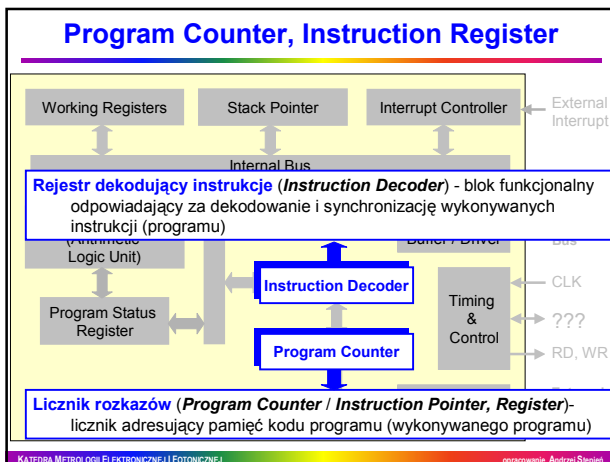
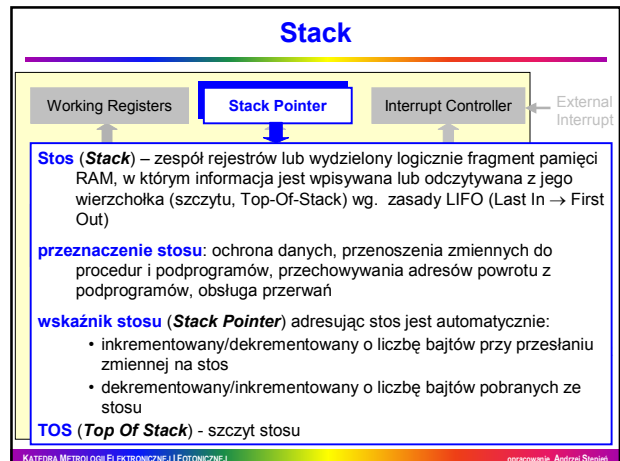
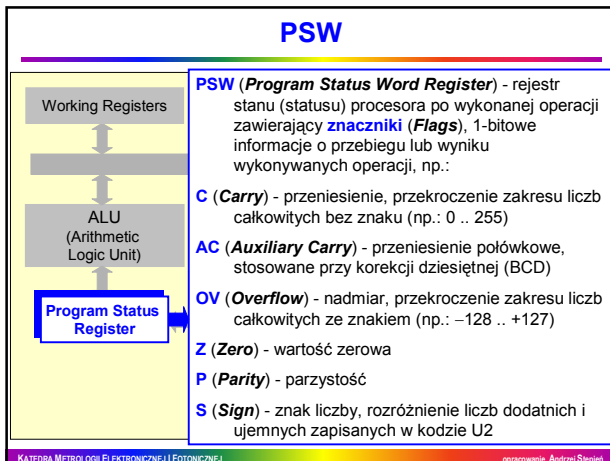
funkcje logiczne - AND, OR, NOT, XOR

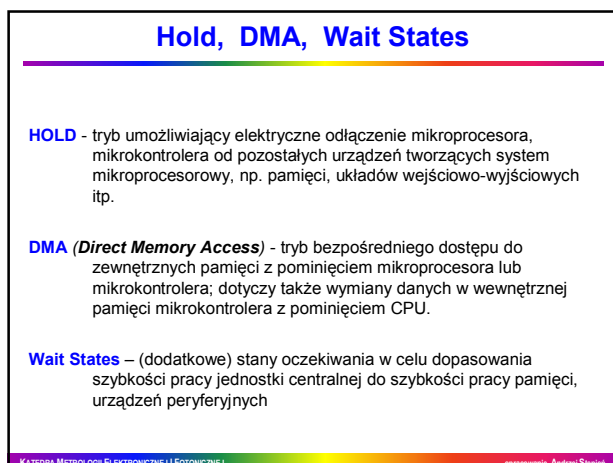
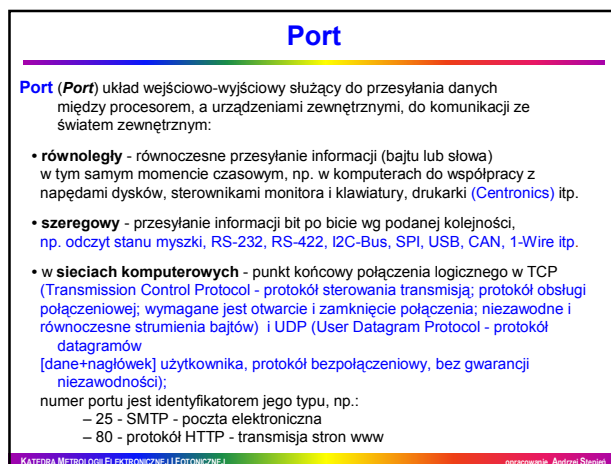
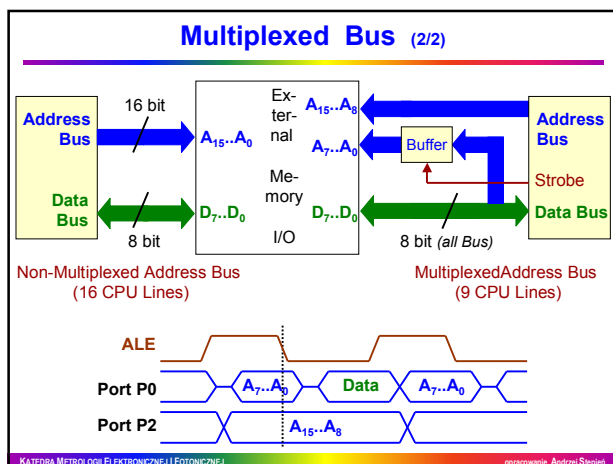
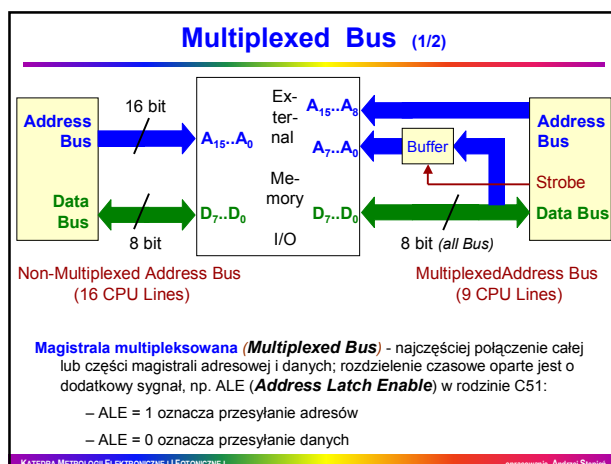
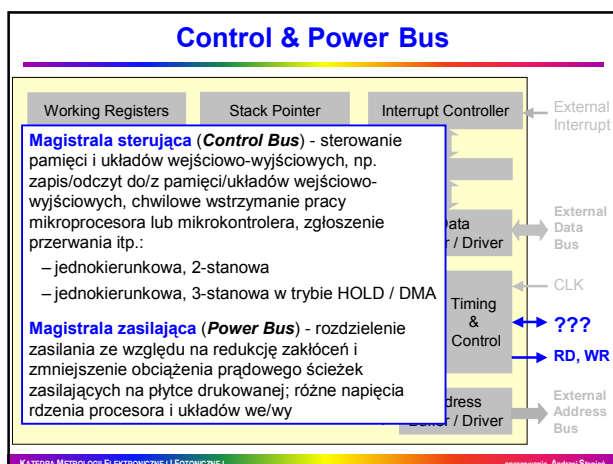
operacje bitowe (w mikrokontrolerach)

Akumulator (Accumulator), W (Work register) - wewnętrzny rejestr ALU zawierający argument lub jeden z argumentów oraz wynik wykonywanych operacji, przeważnie arytmetyczno-logicznych

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak





Alokacja programu

alokacja programu (Program Allocation) - sposób rozmieszczenia programu w pamięci mikroprocesora lub mikrokomputera:

- program **nieprzesuwalny** (absolutny) - bezwzględne adresy rozkazów ustalane na etapie pisania programu, np. start procesora, wektory przerwań, fizyczne obszary pamięci, bankowanie pamięci
- program **przesuwalny** (relokowalny) - kod programu nie zależy od miejsca zajmowanego w pamięci i może być przemieszczany bez dodatkowych przeadresowań, np. kompilatory języków wysokiego poziomu, programy biblioteczne

Oznaczenia

Bit (Binary digit) - pojedyncza, dwuwartościowa (binarna) jednostka informacji o wartości 0 lub 1

Bajt (Byte) - porcja informacji zawierająca (najczęściej) 8 bitów

Słowo (Word) - porcja informacji, którą operuje jednostka centralna, zawiera zwykle całkowitą liczbę bajtów, np. 16, 32 itp.

MSB (Most Significant Bit/Byte) - bit/bajt najbardziej znaczący

LSB (Least Significant Bit/Byte) - bit/bajt najmniej znaczący

prefix: **1 K** = 2^{10} (1024)

1 k = 10^3 (1000)

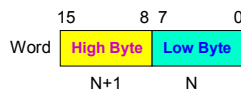
1 M = 2^{20} (1.048.576)

BIT \leftrightarrow **BYTE**: **1 Kb** = 2^{10} bitów (1024 bity)

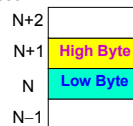
1 KB = 2^{10} bajtów (1024 bajty)

Little / Big Endian Machines

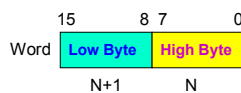
Little Endian:



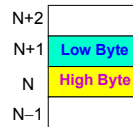
memory address:



Big Endian:



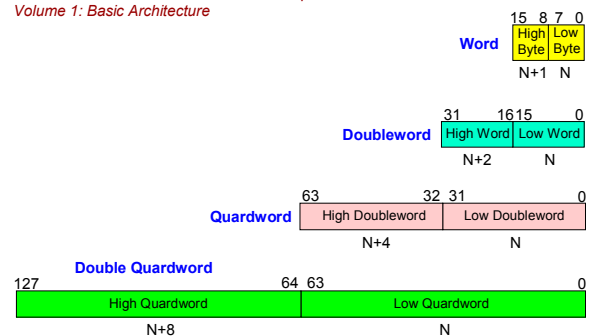
memory address:



Słowa, słowa .. wg INTEL

24547010.pdf

IA-32 Intel® Architecture. Software Developer's Manual.
Volume 1: Basic Architecture



Przetwarzanie rozkazów

Przetwarzanie rozkazów – sposób wykonania rozkazu w procesorze:

- podział** wykonywanego rozkazu na **standardowe fazy**:

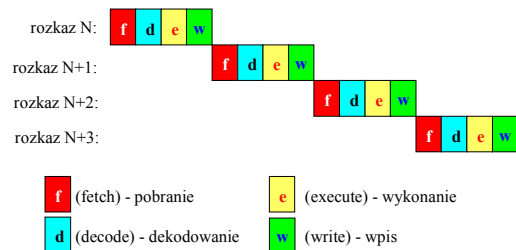
- pobierania (**fetch**) rozkazu z pamięci i umieszczenia w wewnętrznym rejestrze rozkazów lub pamięci buforującej,
- dekodowania (**decode**) rozkazu, ustalenie typu wykonywanej operacji, np. pobrania argumentów,
- wykonania (**execute**) rozkazu, np. wymaganej operacji arytmetyczno-logicznej, obliczenia adresu itp.
- zapisu (**write**) wyniku w rejestrach lub pamięci,

- w przetwarzaniu **sekwencyjnym** (skalarnym) każda faza wykonywana jest oddzielnie, niezależnie od pozostałych,

- w przetwarzaniu **potokowym**, strumieniowym (pipeline processing) poszczególne fazy różnych rozkazów są w zasadzie wykonywane równocześnie.

Sekwencyjne przetwarzanie rozkazów

Sekwencyjne (skalarne) - każda faza wykonywana jest oddzielnie, niezależnie od pozostałych.



f (fetch) - pobranie

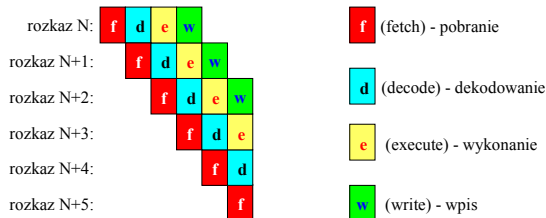
e (execute) - wykonanie

d (decode) - dekodowanie

w (write) - zapis

Potokowe przetwarzanie rozkazów

Potokowe (strumieniowe, pipeline processing) - poszczególne fazy różnych rozkazów są wykonywane równocześnie.



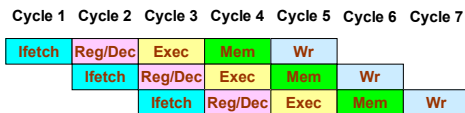
Konflikty przy przetwarzaniu potokowym

- zasobów** – ten sam zasób wykorzystywany jest przez dwie lub więcej faz równocześnie (ALU, rejestry lub pamięć):
 - każda operacja zapisu do rejestrów lub pamięci może być w konflikcie z fazą pobrania rozkazu lub argumentów
- danych** – jeśli argumentem następnego rozkazu jest wynik poprzedniego, który nie został obliczony:
 - wykonanie następnego rozkazu musi być wstrzymane z powodu niedostępności argumentu
- sterowania** – gdy wykonywany jest rozkaz skoku warunkowego zależny od wskaźników ustalanych przez poprzednie, jeszcze nie zakończone rozkazy

Przetwarzanie potokowe - LOAD (1/2)

Instrukcja typu **LOAD** złożona z 5 cykli maszynowych:

- Ifetch:** pobranie kodu instrukcji z pamięci programu
- Reg/Dec:** pobranie zawartości rejestru i dekodowanie instrukcji
- Exec:** obliczenie adresu zmiennej w pamięci danych
- Mem:** odczyt wartości zmiennej z pamięci danych
- Wr:** wpisanie wartości zmiennej do rejestru

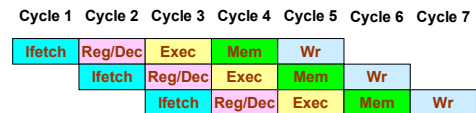


- instrukcja wykonywana w 5 niezależnych blokach funkcjonalnych
- każda instrukcja korzysta tylko 1 raz z pojedynczego bloku funkcjonalnego
- pobranie kodu następnej instrukcji możliwe po zakończeniu poprzedniego

Przetwarzanie potokowe - LOAD (2/2)

Instrukcja typu **LOAD** złożona z 5 cykli maszynowych:

- Ifetch:** pobranie kodu instrukcji z pamięci programu
- Reg/Dec:** pobranie zawartości rejestru i dekodowanie instrukcji
- Exec:** obliczenie adresu zmiennej w pamięci danych
- Mem:** odczyt wartości zmiennej z pamięci danych
- Wr:** wpisanie wartości zmiennej do rejestru

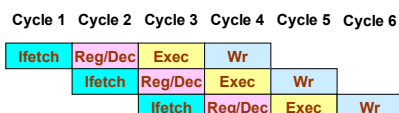


- czas trwania pojedynczej instrukcji wynosi 5 cykli maszynowych
- średni czas trwania instrukcji w programie wynosi 1 cykl maszynowy

Przetwarzanie potokowe - typ R

Instrukcja typu **R** (dostęp do rejestru) złożona z 4 cykli maszynowych:

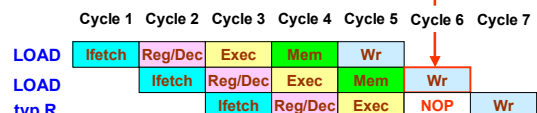
- Ifetch:** pobranie kodu instrukcji z pamięci programu
- Reg/Dec:** pobranie zawartości rejestru i dekodowanie instrukcji
- Exec:** ALU wykonuje operację na zawartości 2 rejestrów
- Wr:** wpisanie wartości zmiennej do rejestru



- typ instrukcji znany po jej zdekodowaniu (Ifetch & Reg/Dec)

Konflikt zasobów

- zasobów** – ten sam zasób wykorzystywany jest przez dwie lub więcej faz równocześnie (ALU, rejestry lub pamięć):
 - każda operacja zapisu do rejestrów lub pamięci może być w konflikcie z fazą pobrania rozkazu lub argumentów

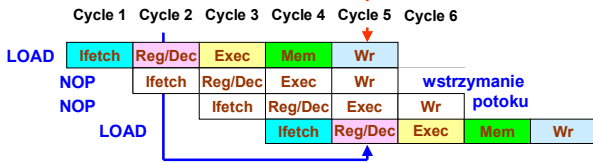


- w jednej z faz musi wystąpić dodatkowy cykl oczekiwania

Konflikt danych

- danych** – jeśli argumentem następnego rozkazu jest wynik poprzedniego, który nie został obliczony:

- wykonanie następnego rozkazu (zatrzymanie potoku) wstrzymane z powodu niedostępności argumentu

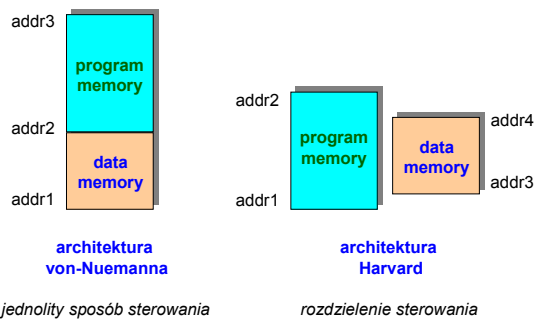


- rozwiązaniem konfliktu jest umieszczenie między rozkazami instrukcji niezależnej od argumentów, np. NOP lub innej (problem optymalizacji kompilatorów – rozkazy są wykonywane nie w takiej kolejności jak zostały napisane w programie)

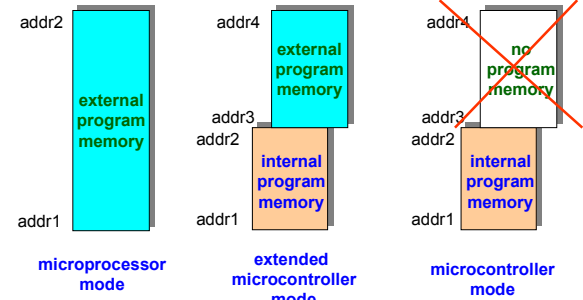
Eliminacja konfliktów

- powielenie** niektórych wewnętrznych układów i **równoległe** wykonywanie tych samych operacji prowadzi do struktury **superskalarnej**,
- czasy realizacji** poszczególnych faz mogą się różnić w zależności od typu wykonywanej operacji,
- takty zegarowe** muszą być tak dobrane, aby dłużej trwające fazy były wykonywane przez całkowitą liczbę tych taktów, np. jałowe takty oczekiwania,
- do **skrócenia czasu** wykonywanych operacji przyczynia się umieszczenie argumentów w wewnętrznych rejestrach, a nie w pamięci; procesor musi być wyposażony w dużą liczbę wewnętrznych rejestrów,
- zmniejszeniu liczby rozkazów** potrzebnych do wykonania zadania sprzyja duża długość słowa maszynowego.

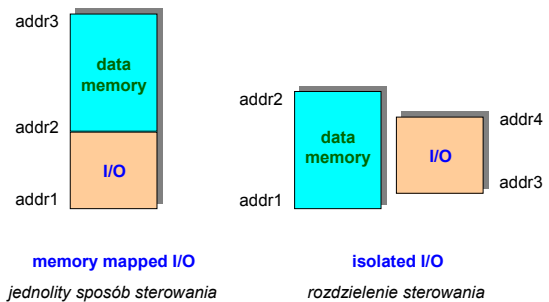
Adresowanie pamięci



Adresowanie pamięci programu



Adresowanie układów we/wy



Sposób obliczania adresu

liniowy:

adresem jest zawartość rejestru bazowego

np.

licznika rozkazów PC (Program Counter), IR (Instruction Register)

segmentowy:

adresem jest suma zawartości dwóch rejestrów: segmentowego i bazowego (liniowe adresowanie wewnątrz segmentu)

np. w komputerach PC

- rejestru segmentowego: CS = 1234h (Code Segment)

- licznika rozkazów: PC = 5678h (Program Counter)

$$\text{adres} = \text{CS} : \text{PC} = 1234h : 5678h = 1234h * 16 + 5678h = 179B8h$$

Architektura procesora typu CISC

J. Biernat: *Architektura komputerów.*

Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2001

- **CISC (Complex Instruction Set Computer)** - procesor:
 - o złożonej liście rozkazów mających różny, zmienny format
 - z małym zestawem rejestrów strukturalnych (dla użytkownika)
 - o rozbudowanych trybach adresowania
- wzrost liczby taktów zegarowych przy wzroście złożoności instrukcji
- trudności z racjonalnym wykorzystaniem zasobów procesora; przy prostych, elementarnych operacjach wykorzystanie niewielkiej części zasobów procesora
- statystyka działania systemów operacyjnych i programów użytkowych wskazuje na częste wykonywanie operacji prostych i rzadkie złożonych
- sterowanie przepływem rozkazów realizowane programowo (mikro-kody); łatwa realizacja sterowania wykonywania złożonych rozkazów wymagających różnych okresów czasu

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

What is "microcode"?

www.intel.com: XP SP2 vs. Intel Prescott

Anything sufficiently complex that has been handcrafted by **humans will contain errors**. Modern software has millions of lines of source code, and modern processors have millions of transistors, so this applies to both.

Microsoft Operating Systems arrive on CD that is built to a particular **Service Pack** level, and errors found after that was made will be fixed via patches **downloaded** from their web site.

Intel processors are manufactured to a particular stepping level, and **errors** found after that was made may be **fixed via microcode** updates sent to the processor, typically by BIOS on every system startup.

Microcode updates are not stored permanently within the processor; they vanish when the power is cut, so **every boot has to re-assert** them.

Intel® IXP1200 Network. Processor Family. Microcode Programmer's Reference Manual. INTEL Corporation, March 2002, Part Number: 278304-011

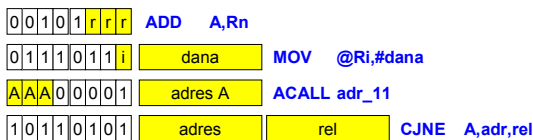
80C196KB User's Guide. INTEL Corporation, November 1990, Order Number: 270651-003

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

C51 - procesor typu CISC

- różny, zmienny format rozkazów:



- wzrost liczby taktów zegarowych przy wzroście złożoności instrukcji:
 - 1 lub 2 cykle maszynowe dla wszystkich instrukcji
 - 4 cykle maszynowe dla MUL AB oraz DIV AB (standard)
- elementarne operacje wykorzystują tylko wybrane rejestry:
 - uprzywilejowany akumulator A i znacznik przeniesienia C
 - adresowanie pośrednie tylko przez R0 i R1 (z 8 dostępnych rejestrów)

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu RISC (1/2)

Zd. Pogoda: *Mikroprocesory RISC rodziny PowerPC.*

Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice 1995

- uproszczoną strukturę mikroprocesora, o zredukowanej liście rozkazów zaproponował John Cocke (IBM Research w Yorktown, New York, 1974) wychodząc z założenia, że 20% dostępnych instrukcji zajmuje aż 80% czasu pracy procesora
- skrót **RISC (Reduced Instruction Set Computer)** zaproponował David Patterson (University of California, Berkeley, 1985)

obecnie skrót **RISC** tłumaczony jako **Rational Instruction Set Computer**
- idea RISC została wykorzystana w mikroprocesorze SPARC firmy Sun Microsystems; początek powstania tego co jest znane jako technologia MIPS, np. w komputerach Silicon Graphics, procesorze Alpha firmy DEC

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu RISC (2/2)

- prosta struktura wewnętrzna
- sterowanie przepływem rozkazów realizowane sprzętowo (brak mikro-kodów, eliminacja mikroprogramowania)
- uproszczenie trybów adresowania, zwykle jednolity format rozkazów (stała długość kodów instrukcji)
- większe możliwości wyboru uniwersalnych rejestrów mikroprocesora
- architektura typu load/store, przetwarzanie danych w oparciu o rejestry nie bezpośrednio na zawartości pamięci
- szybsze działanie → ideałem jest wykonanie każdej instrukcji w jednym cyklu maszynowym (przetwarzanie potokowe)
- ułatwienie opracowania, wytworzenia i testowania:
 - nowego mikroprocesora
 - systemu operacyjnego i programu użytkowego
 - kompilatora wysokiego poziomu
- wykonanie operacji równoważnej wymaga większej liczby rozkazów w procesorze RISC niż w procesorze CISC

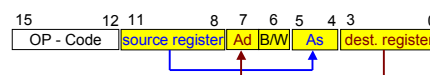
KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

MSP430 - procesor typu RISC

www.ti.com

- 27 basic instructions + 24 emulated instructions
- jednolity format rozkazów (stała długość kodu), np. dla formatu I:



- większe możliwości wyboru uniwersalnych rejestrów mikroprocesora:
 - R0 - licznik rozkazów PC
 - R1 - wskaźnik stosu SP
 - R2 - rejestr statusowy SR / generator stałych CG1
 - R3 - generator stałych CG2
 - R4 .. R15 - rejestry uniwersalne
- ortogonalne tryby adresowania - wszystkie instrukcje ze wszystkimi trybami adresowania i operandami

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Stasiak

Architektura procesora typu Bit Slice

- każdy blok funkcjonalny tworzą oddzielne układy zwane **segmentami**, np. ALU, rejestry
- istnieje możliwość wzajemnej współpracy (połączenia) segmentów w celu zbudowania procesora o zadanej długości słowa, np. dysponując procesorem 4-bitowym można zbudować procesor 8-, 16-, 24- ...-bitowy
- mikrokod procesora (*microcode*) jest pamiętany w pamięci ROM lub szybkiej pamięci RAM (dynamiczne mikroprogramowanie)
- najpopularniejszym procesorem segmentowym jest seria Am2900 (*Advanced Micro Devices*), np. **Am2901**:
 - **struktura**: 4-bitowa ALU, 16x4-bitowa pamięć RAM (rejestry), akumulator Q
 - **wykonywane operacje**: dodawanie z przeniesieniem (*addition with carry*), odejmowanie z pożyczką (*subtraction with borrow*), OR, AND, XOR i XNOR
 - **argumenty** zawarte w: RAM, akumulatorze, zewnętrznych wejściach, domyślna wartość 0,
 - **wynik operacji**: przeniesienie dla innych segmentów (*carry-out flag*), znak (*sign flag*), przekroczenie zakresu (*overflow flag*), wynik zerowy (*zero flag*)...

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Szustek

Optymalizacja, przetwarzanie danych

Optymalizacja kompilatorów:

- rozkazy wykonywane przez procesor realizują funkcje, które najczęściej powtarzają się podczas kompilacji
- lista rozkazów dostosowana do potrzeb kompilatorów, a nie własności procesora
- założenia przydatne przy projektowaniu nowych konstrukcji procesorów

Sposób przetwarzania danych - obciążenie rejestrów procesora i pamięci:

- **register-to-register** → przestrzenią roboczą są rejestry ogólnego przeznaczenia:
 - **accumulator-based** CPU, z akumulatorem związana jest większość instrukcji arytmetyczno-logicznych i wymiany danych
 - **register-based** CPU (register-to-register), rolę akumulatora pełnią rejestry ogólnego przeznaczenia
- **memory-to-memory** → bloki pamięci traktowane są jako przestrzeń roboczą, zastępująca rejestry ogólnego przeznaczenia

KATEDRA METROLOGII ELEKTRONICZNEJ I FOTONICZNEJ

opracowanie: Andrzej Szustek