



Co-funded by the
Erasmus+ Programme
of the European Union

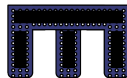
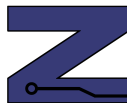
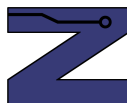
*"Teaching online electronics,
microcontrollers and programming in
Higher Education"*
Erasmus+ Programme
2020-1-PL01-KA226-HE-095653

PODSTAWY SYSTEMÓW MIKROPROCESOROWYCH: WPROWADZENIE DO MIKROKONTROLERÓW

dr inż. Dariusz Tefelski

dariusz.tefelski@pw.edu.pl

Pokój 225GF



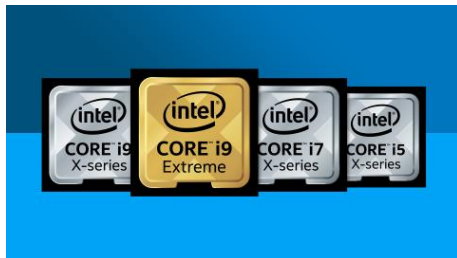
NA JEDNYM KOŃCU SKALI...



Źródło obrazka: <https://www.morele.net/>.

Parametry:

- procesor: Intel Core i9 z serii X,
- liczba rdzeni: 10-18,
- częstotliwość: 3.0-4.6 GHz,
- moc: 165 W.



Źródło obrazka: <https://www.intel.pl>.

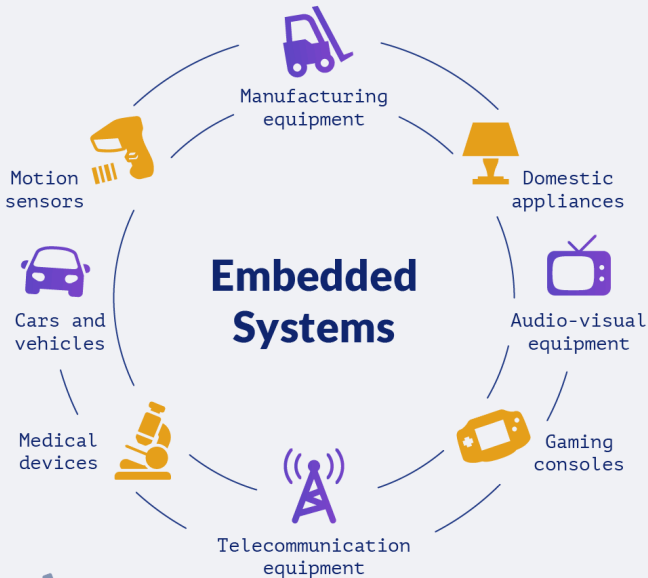
NA DRUGIM KOŃCU SKALI



Systemy wbudowane
(*Embedded Devices*)

Moc: $\sim$ mW

Źródła obrazków: <https://www.apple.com>, <https://centrumstrazaka.pl/>,
<https://exsmart.pl/>, <https://kenys.pl>.



- **Urządzenia osobistego użytku:** telefon komórkowy, tablet, zegarek, dyktafon, kalkulator.
- **Komponenty komputerowe:** myszka, klawiatura, joystick, router, fax, karta dźwiękowa, ładowarka baterii.
- **Urządzenia domowe:** monitoring domu i posesji, zamek do drzwi, automatyka do rolet czy światel, budzik, termostat, pompa ciepła, klimatyzator, rekuperacja, fotowoltaika, sprzęt fitness, lodówka, pralka, zmywarka, kuchenka mikrofalowa.
- **Zabawki/rozrywka:** konsole do gier, samochody, lalki, roboty, drony.

- Procesor:
 - 2X szybkość co każde 1.5 roku;
 - **100X w ostatniej dekadzie.**
- Pamięć:
 - 2X pojemność co każde 2 lata;
 - **64X w ostatniej dekadzie;**
 - koszt jednego bitu: maleje o 25% na rok.
- Dysk:
 - pojemność > 2X każdego roku;
 - **250X w ostatniej dekadzie;**
 - koszt jednego bitu: maleje o 50% na rok.

"Gdyby rozwój samochodów odzwierciedlał rozwój komputerów, Rolls Royce kosztowałby dziś 100 dolarów, pokonywałby milion kilometrów na jednym litrze paliwa i eksplodował raz do roku zabijając wszystkich wewnątrz."

Triumph of the Nerds - Robert X. Cringely

Zmierzch ery PC?



Po co mam się uczyć PSM?

- Bo to jest fajne! I będzie coraz fajniejszy!
- Komputery PC odpowiadają za sprzedaż mniej niż 1% wszystkich procesorów. **Systemy wbudowane za ponad 99%.**



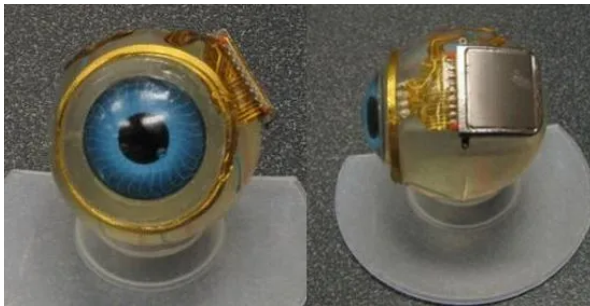
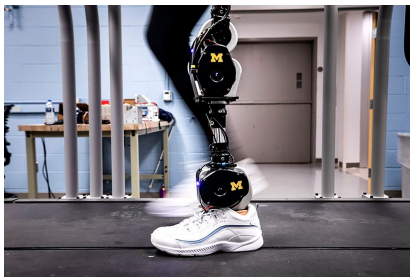
Bionika:

Czujniki w lateksowych palcach nieprzerwanie rejestrują temperaturę, electroniczny interfejs w sztucznej kończynie stymuluje zakończenia nerwów w ramieniu, które przekazują tę informację do mózgu. Wart 3,000\$ system pozwala czuć ciśnienie i ciężar. Po raz pierwszy od wypadku w 1986 roku ten strażak może podnieść szklankę bez zgniecenia jej i nie pozwalając jej się wyslizgnąć.

One Digital Day

Nie musisz mieć dyplomu lekarza, by pomagać ludziom.

Po co mam się uczyć PSM?



Źródła obrazków: <https://circuitdigest.com>, <https://thenewstack.io>, <https://www.pcworld.pl>

- ❶ Wprowadzenie
- ❷ Mikrokontrolery
 - Architektura wewnętrzna
 - Zasada działania
- ❸ Programowanie mikrokontrolerów
- ❹ Komunikacja zewnętrzna i standardy komunikacyjne
 - Interfejsy szeregowo
 - Interfejsy równoległe
- ❺ Zastosowania z przykładami
- ❻ Mikrokontrolery w zaawansowanych systemach.

❶ Materiały z platformy Moodle: <http://study.engined.eu/>

❷ www.google.pl

❸ Książki:

- Mirosław Kardaś, *Mikrokontrolery AVR, Język C*, Atmel (2011).
- Tomasz Francuz, *Język C dla mikrokontrolerów AVR*, Helion (2011).
- Filip Sala, Marzena Sala-Tefelska, *Wprowadzenie do mikrokontrolerów AVR. Od elektroniki do programowania*, Helion.
- Ryszard Pełka, *Mikrokontrolery. Architektura, programowanie, zastosowania*, WKŁ.
- Jarosław Doliński, *Mikrokontrolery AVR w praktyce*, BTC.
- Rafał Baranowski, *Mikrokontrolery AVR ATmega w praktyce*, BTC.
- Piotr Gałka, Paweł Gałka, *Podstawy programowania mikrokontrolera 8051*, Mikom.
- James M. Sibigtroth, *Zrozumieć małe mikrokontrolery*, BTC.
- A. Pawluczuk, *Sztuka programowania mikrokontrolerów AVR, podstawy*, BTC.
- Tomasz Jabłoński, *Mikrokontrolery PIC16F8x w praktyce*, BTC.
- Jacek Bogusz, *Lokalne interfejsy szeregowo w systemach mikroprocesorowych*, BTC.

- ❶ Ocena końcowa jest równa średniej ważonej ocen z laboratorium i wykładu z wagami odpowiednio: $\frac{2}{3}$ i $\frac{1}{3}$ (przy czym obydwie oceny muszą być pozytywne).
- ❷ Zaliczenie wykładu:
 - Ocena z wykładu jest wystawiana na podstawie kolokwium za 12 pkt.

Ocena	Suma punktów
3.0	[6;7)
3.5	[7;8.5)
4.0	[8.5;9.5)
4.5	[9.5;11)
5.0	[11;12]

- Pod koniec każdego tematu będą uruchamiane quizy na platformie Moodle: <http://study.engined.eu/> . Podejście do quizów jest dobrowolne.
- Benefity za wykonywanie quiz-ów na platformie Moodle:
 - (A) Nie więcej niż 5% najlepszych studentów uzyskuje ocenę 5.0 za część wykładową. W takim przypadku są zwolnieni z kolokwium.
 - (B) W przypadku uzyskania minimum 70% punktów z quizów, ocena za część wykładową jest podnoszona o 0.5 stopnia.
 - (C) W przypadku uzyskania [50-70)% punktów z quizów, punktacja za część wykładową jest zwiększana o 0.75 pkt.

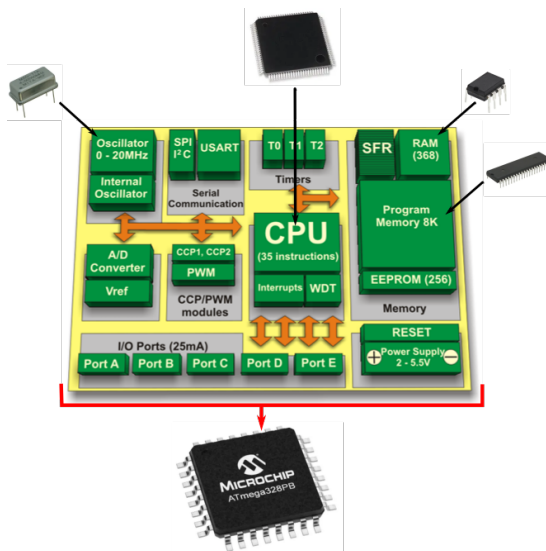
- 8 Ocena z laboratorium jest wystawiana w następujący sposób:
- 3.0 - w przypadku zaliczenia wszystkich ćwiczeń laboratoryjnych oraz miniprojektu,
 - +0.5 stopnia - w przypadku wykonania projektu końcowego,
 - +0.5 stopnia - w przypadku wykonania dokumentacji technicznej do projektu końcowego,
 - +0.5 stopnia - w przypadku zaliczenia kursu on-line z języka C na platformie Moodle: <http://study.engined.eu/> . Przez zaliczenie rozumiemy ukończenie wszystkich tematów z ramach kursu i uzyskanie minimum 50% punktów. Kurs można zrealizować samodzielnie w dowolnym momencie trwania semestru. Jednak zalecam wykonanie go na początku semestru.

MIKROKONTROLER I MIKROPROCESOR

- **Mikroprocesor** to cyfrowy układ scalony, którego działanie jest sterowane pobieranymi z zewnątrz rozkazami. Ciąg rozkazów sterujących pracą mikroprocesora stanowi jego program.
- **Mikrokontroler** to układ scalony, w którego strukturze zawarto mikroprocesor oraz pewien zestaw elementów zewnętrznych (peryferyjnych).

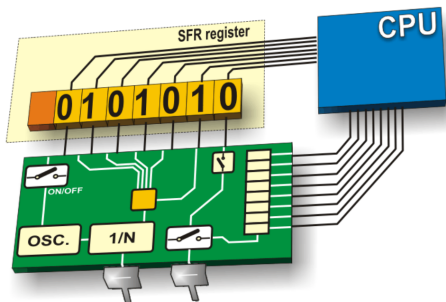
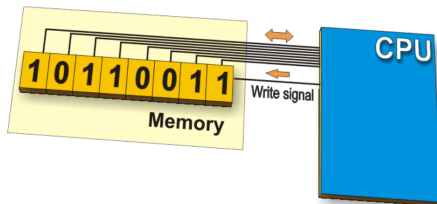
	Mikroprocesor	Mikrokontroler
Peryferia zewnętrzne	RAM, ROM, porty I/O, liczniki, wiele innych	Pamięć (opcjonalnie)
Aplikacje	PC, laptopy, tablety	Systemy wbudowane, sterowniki, motoryzacja itd.
Szybkość zegara	4M-4G	32K-20M
Zużycie prądu	5 - 150 W	100 μ W - 2 W
Przykład	Motorola 68000, Intel x86	Intel 8051, Atmel AVR

ARCHITEKTURA MIKROKONTROLERA



Źródło: <https://www.electronicshub.org/pic-microcontroller-architecture/>, <https://pl.rs-online.com>,
<https://www.gmelectronic.com>, <https://pl.farnell.com>.

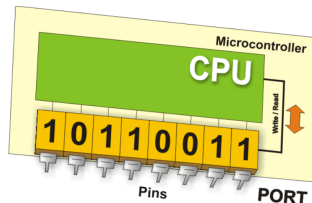
- **Rejestr** - komórka pamięci, układ elektroniczny, który potrafi zapamiętać stan jednego bajta.



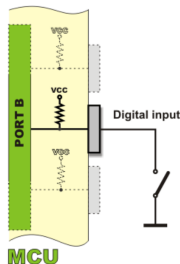
- Rejestr specjalny (Special Function Register - SFR)
Oprócz rejestrów, które nie mają określonej z góry funkcji, każdy mikrokontroler posiada pewną liczbę rejestrów, których funkcja została zaprojektowana odgórnie przez producenta. Poszczególne bity tych rejestrów są dosłownie połączone z wewnętrznymi obwodami mikrokontrolera.

PORTY WEJŚCIA/WYJŚCIA

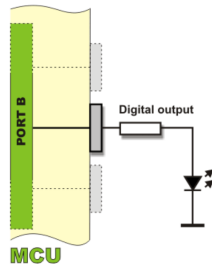
- Porty to rejestry połączone do pinów (nóżek) mikrokontrolera.
- Za pomocą odpowiedniego rejestru wybiera się rolę pinów (wejście/wyjście).
- Porty mogą posiadać rezystory podciągające (czasem również kontrolowane przez odpowiedni rejestr).



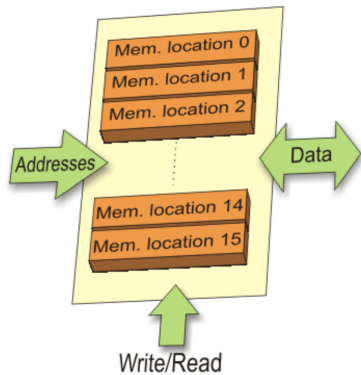
Pin with pull-up resistor



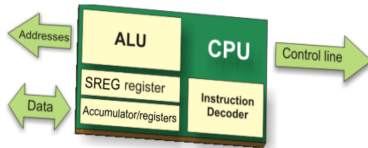
Pin without pull-up resistor



- **Pamięć** - część mikrokontrolera umożliwiająca przechowywanie danych i kodu programu.
- **Pamięci nieulotne** (zawartość pamięci nie ulega skasowaniu w momencie wyłączenia zasilania):
 - ROM (*Read Only Memory*)- pamięć tylko do odczytu,
 - EPROM (*Erasable Programmable Read Only Memory*),
 - EEPROM (*Electrically Erasable Programmable Read Only Memory*),
 - Flash.
- **Pamięci ulotne** (zawartość pamięci ulega skasowaniu w momencie wyłączenia zasilania):
 - SRAM - statyczne, droższe, mniej pakowne, szybsze;
 - DRAM - dynamiczne, tańsze, bardziej pojemne, wolniejsze;

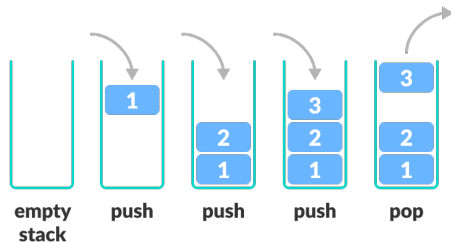


- **Centralna jednostka obliczeniowa** (Central Processing Unit - CPU) to serce mikrokontrolera, monitoruje i kontroluje wszystkie procesy wewnątrz mikrokontrolera.



- **Dekoder instrukcji** - na podstawie zawartości rejestru instrukcji, generuje odpowiednie sygnały sterujące dla automatu realizującego funkcje procesora. Zbiór instrukcji jest odmienny dla każdej rodziny mikrokontrolerów.
- **Jednostka arytmetyczno-logiczna** (Arithmetical Logical Unit ALU) wykonuje wszystkie matematyczne i logiczne operacje na danych.
- Wynik operacji jest umieszczany w dowolnym rejestrze (w starszych mikrokontrolerach tylko jeden taki rejestr - akumulator). Może być wiele ALU; ALU może zawierać FPU (Floating-Point Unit) do obliczeń zmiennoprzecinkowych.
- **Rejestr SREG** - jest rejestrem statusowym zawierającym informacje o rezultacie ostatnio wykonanej przez ALU operacji arytmetycznej (znak, przeniesienie, przepełnienie do dwóch, itd..).

- **Stos** (stack) rodzaj pamięci danych (logiczny).
- Pobieranie danych ze stosu musi być wykonywane w kolejności odwrotnej niż ich umieszczanie na stosie. Tak więc w danej chwili możliwy jest dostęp do jednego tylko - ostatniego elementu stosu.

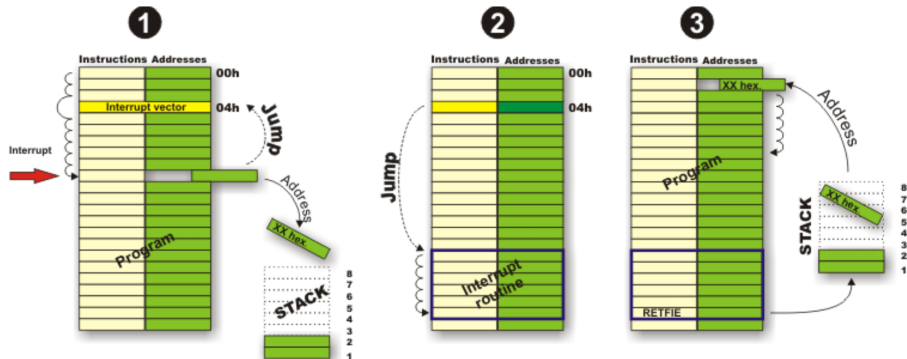


Źródło: <https://medium.com/@bereza95/data-structures-stack-first-in-last-out-8faf242aa6c2>

- Podstawowym zastosowaniem stosu jest zapamiętywanie adresów powrotu podczas wywoływania procedur. Stos wykorzystywany jest też jako rodzaj podręcznej pamięci do chwilowego przechowywania danych.
- Przy małych pojemnościach pamięci SRAM należy zwracać baczną uwagę, by odkładanie danych na stos (np. przy wykonywaniu wielokrotnie zagnieżdżonych skoków) nie powodowało jego rozszerzenia na obszar zmiennych programowych.

PRZERWANIE (INTERRUPT)

- Idea przerwania w mikrokontrolerach polega na tym, że w odpowiedzi na określony sygnał (sygnał przerwania) mikrokontroler zawiesza chwilowo wykonywanie programu głównego i wykonuje specjalną procedurę określaną mianem **procedury obsługi przerwania**. Po zakończeniu tej procedury mikrokontroler wraca do wykonywania programu głównego, począwszy od miejsca, w którym zostało ono zawieszone.

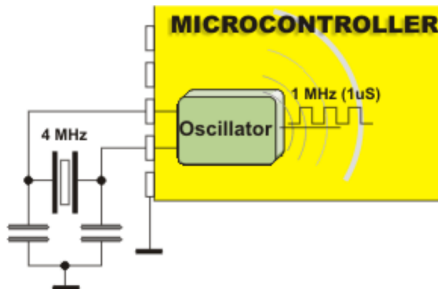


PRZERWANIA (INTERRUPT)

- Podstawową funkcją przerwań jest umożliwienie szybkiego reagowania na zdarzenie zewnętrzne.
- Przerwania nie zawsze mają postać sygnałów zewnętrznych - mogą być one generowane także przez układy wewnętrzne, np. liczniki, łącza szeregowo.
- Stosowanie przerwań umożliwia maksymalne wykorzystanie możliwości przetwarzania danych, jakie daje mikrokontroler.

Zegar

- Serce procesora wyznaczające rytm jego działania.
- Każdy takt zegara - cykl zegarowy.
- Do niedawna procesory wymagały kilku cykli zegarowych do wykonania jednej instrukcji.
- Obecnie procesor może wykonać więcej niż jedną instrukcję na cykl (*pipelining*).



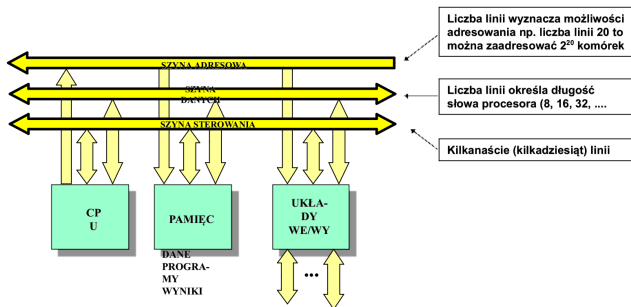
Magistrala (bus) - szyna komunikacyjna (zbiór linii, drucików). Zawarte w magistralach linie można podzielić na trzy grupy funkcjonalne:

- 1 linie danych,
- 2 linie adresów,
- 3 linie sterowania.

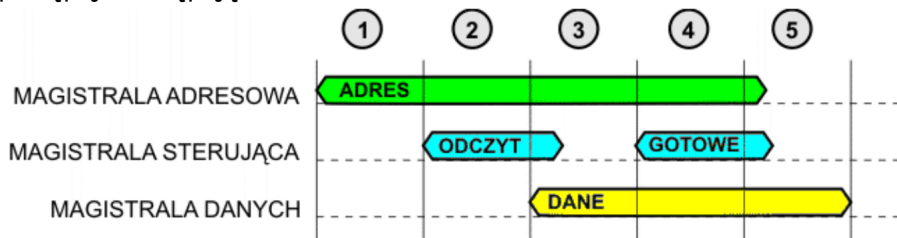


- **Linie danych (data bus)** są ścieżkami służącymi do przenoszenia danych między modułami systemu. Szyna danych składa się typowo z 8, 16 lub 32 oddzielnych linii, przy czym liczba linii określa szerokość tej szyny. Ponieważ w danym momencie każda linia może przenosić tylko 1 bit, z liczby linii wynika, ile bitów można jednocześnie przenosić. Szerokość szyny danych jest kluczowym czynnikiem określającym wydajność całego systemu. Jeśli na przykład szyna danych ma szerokość 8 bitów, a każdy rozkaz ma długość 16 bitów, to procesor musi łączyć się z modułem pamięci dwukrotnie w czasie każdego cyklu rozkazu.

- **Linie adresowe** są wykorzystywane do określania źródła lub miejsca przeznaczenia danych przesyłanych magistralą. Jeśli na przykład procesor ma zamiar odczytać słowo (8, 16 lub 32 bity) danych z pamięci, umieszcza adres potrzebnego słowa na linii adresowej. Szerokość szyny adresowej determinuje maksymalną możliwą pojemność pamięci systemu. Ponadto linie adresowe są również używane do adresowania portów wejścia-wyjścia.
- **Linie sterowania** używa się do sterowania dostępem do linii danych i linii adresowych, a także do sterowania ich wykorzystaniem. Ponieważ linie danych i adresowe służą wszystkim zespołom, musi istnieć sposób sterowania ich używaniem. Sygnały sterujące przekazywane między modułami systemu zawierają zarówno rozkazy, jak i informacje regulujące czas (taktujące). Sygnały czasowe określają ważność danych i adresów. Sygnały rozkazów precyzują operacje, które mają być przeprowadzone.

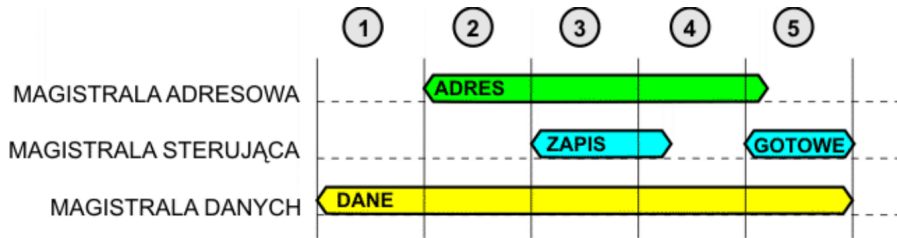


Gdy procesor chce odczytać dane z pamięci lub z urządzenia wejścia, to postępuje następująco:



- 1 Umieszcza na magistrali adresowej adres komórki pamięci lub numer urządzenia we/wy.
- 2 Na magistrali sterującej umieszcza informację, że chce odczytać dane z pamięci lub z urządzenia we/wy.
- 3 Pamięć lub urządzenie wejścia umieszcza dane na magistrali danych.
- 4 Pamięć lub urządzenie wejścia umieszcza na magistrali sterującej informację, że dane są gotowe do odczytu z magistrali danych.
- 5 Procesor odczytuje dane z magistrali danych.

Podobnie wygląda zapis do pamięci lub do urządzenia wyjścia:



- 1 Procesor umieszcza dane na magistrali danych.
- 2 Procesor umieszcza adres komórki pamięci lub numer urządzenia wyjścia na magistrali adresowej.
- 3 Procesor umieszcza na magistrali sterującej informację, że chce zapisać dane do pamięci lub urządzenia we/wy.
- 4 Pamięć lub urządzenie wyjścia odczytują dane z magistrali danych. Pamięć umieszcza dane pod adresem odczytanym z magistrali adresowej. Dla urządzeń wyjścia adres określa numer urządzenia, dla którego są przeznaczone dane.
- 5 Pamięć lub urządzenie wyjścia umieszczają na magistrali sterującej potwierdzenie odebrania danych.

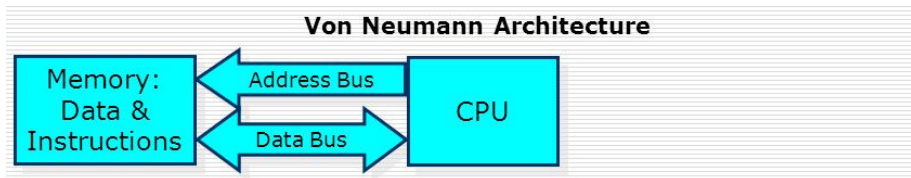
ARCHITEKTURA VON-NEUMANN-A A HARVARD-A



Źródło: https://www.researchgate.net/publication/331160423_Von-Neumann_Architecture_Vs_Harvard_Architecture

ARCHITEKTURA VON-NEUMANN-A

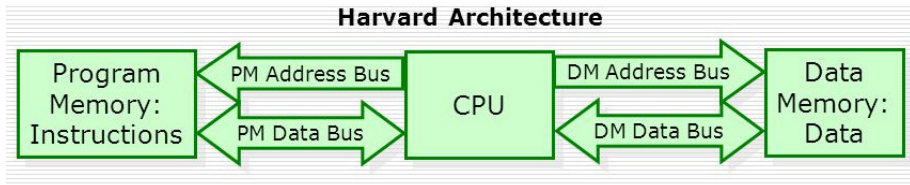
- Składa się z jednej szyny danych, która jest wspólna dla danych i programu.
- Umowny podział pamięci na dane i program.
- Zaleta: prostota pisania programu. Za pomocą tych samych komend można uzyskać dostęp zarówno do danych programu jak i rozkazów.
- Wada: powolna realizacja programu.



Źródło: Veton Kepuska, <https://slideplayer.com/slide/4892266/>

ARCHITEKTURA HARVARD-A

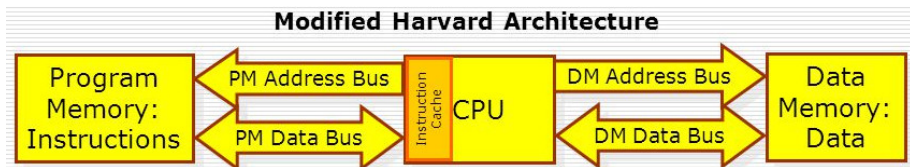
- Składa się z dwóch szyn: danych i rozkazów.
- Zaleta: możliwe równoczesne wykonywanie rozkazów i pobieranie danych.
- Wady: szyna danych i rozkazów mają różne szerokości. Utrudniony przepływ danych pomiędzy pamięciami.



Źródło: Veton Kepuska, <https://slideplayer.com/slide/4892266/>

ZMODYFIKOWANA ARCHITEKTURA HARVARD-A

- Nadal mamy rozdzielenie pamięci programu od pamięci danych lecz mają one taką samą długość słowa.



Źródło: Veton Kepuska, <https://slideplayer.com/slide/4892266/>

ARCHITEKTURA PROCESORA WEDŁUG LISTY ROZKAZÓW

RISC (REDUCED INSTRUCTION SET COMPUTER):

- Oparty o architekturę Harvard-a.
- Wykorzystuje przetwarzanie potokowe aby zwiększyć szybkość programu.
- Zredukowana liczba rozkazów (instrukcji) - nawet poniżej 30!
- Większość rozkazów wykonywana w jednym cyklu procesora.
- Rozkazy proste lub bardzo proste (+, -, kopiowanie).
- Bardzo mała liczba trybów adresowania.
- Ograniczony dostęp do pamięci.
- Szybkie działanie dekodera rozkazów.
- Skompilowane programy są dłuższe.
- Użycie: ARM, PA-RISC, Power Architecture, Alpha, AVR, ARC.

CISC (COMPLEX INSTRUCTION SET COMPUTER):

- Duża liczba rozkazów (instrukcji) (ponad 200).
- Niektóre rozkazy potrzebują dużej liczby cykli procesora do wykonania.
- Występowanie złożonych, specjalistycznych rozkazów.
- Duża liczba trybów adresowania.
- Powolne działanie dekodera rozkazów.
- Krótkie skompilowane programy.
- Użycie: VAX, Motorola 68000, System/360, AMD oraz Intel x86 CPUs.

O LICZBACH - DLA PRZYPOMNIENIA...

- **Bit** - cyfra w układzie binarnym (podstawowa jednostka informacji).
- **Bajt** - słowo, liczba złożona z 8 bitów.

Przeliczanie liczb:

- z systemu dwójkowego na dziesiętny:
np. $1100 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 + 0 + 0 = 12$
- z systemu dwójkowego na szesnastkowy:

np. **1101** **0010**

$$0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$$

$$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$$

W systemie szesnastkowym: A=10, B=11, C=12, D=13, E=14, F=15, zatem:

1101 **0010** = 0x**D2**, alternatywne zapisy: 0xd2, D2h.

O LICZBACH - DLA PRZYPOMNIENIA...

- z systemu dziesiętnego na dowolny :

(A) liczba 21 z dziesiętnego na czwórkowy:

Iteracja	Operacja	Reszta	Aktualna liczba
1	$21 // 4 = 5$	1	1
2	$5 // 4 = 1$	1	11
3	$1 // 4 = 0$	1	111

czyli 21 jest równe 111 w systemie czwórkowym. Inny zapis: 111_4 .
Indeks informuje o podstawie.

(B) liczba 21 z dziesiętnego na piątkowy:

Iteracja	Operacja	Reszta	Aktualna liczba
1	$21 // 5 = 4$	1	1
2	$4 // 5 = 0$	4	41

czyli 21 to 41_5 .

O LICZBACH - DLA PRZYPOMNIENIA...

- z dowolnego systemu liczbowego na dziesiętny:

$$(A) \text{ } 120_3 = 1 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0 = 9 + 6 + 0 = 15$$

$$(B) \text{ } 33_9 = 3 \cdot 9^1 + 3 \cdot 9^0 = 27 + 3 = 30$$

$$(C) \text{ } 1h_{20} = 1 \cdot 20^1 + 17 \cdot 20^0 = 37, \text{ bo } a=10, \dots, h=17.$$

Dziękuję za uwagę.



Pierwotna wersja wykładu jest dziełem:
dr hab. Piotra Fronczaka, mgr inż. Marcina Zaręby.