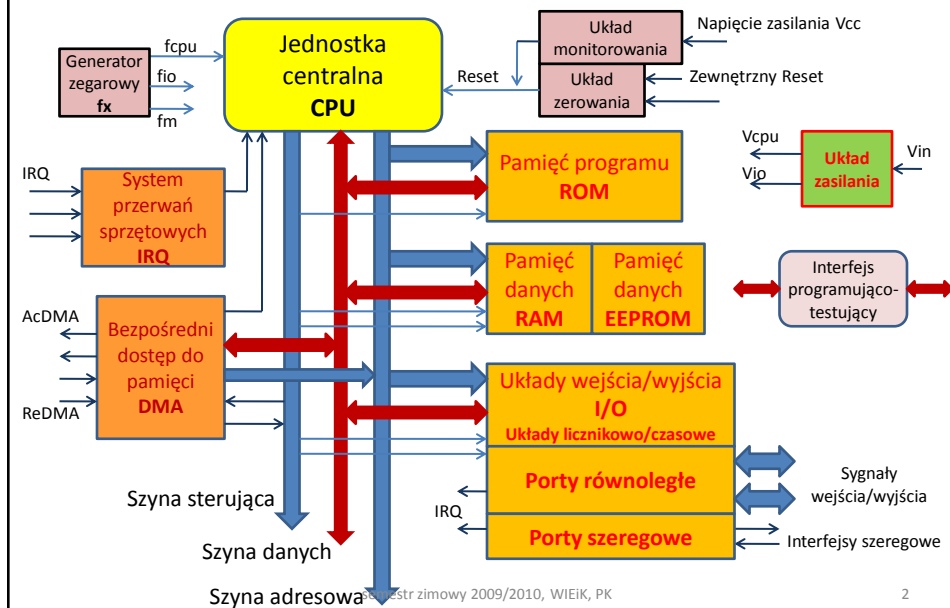


## Podstawowa budowa systemu mikroprocesorowego

semestr zimowy 2009/2010, WIEiK, PK

1

### Podstawowa struktura systemu mikroprocesorowego



2

## Pamięć programu - ROM

- **Pamięć programu** (*ang. program memory – ROM – Read Only Memory*) – pamięć do odczytu przeznaczona do przechowywania kodu programu. Pamięć nieulotna, po zaniku napięcia zasilania zawartość pozostaje.
- Podstawowe parametry pamięci ROM:
  - Pojemność (w kB i Ksłowach, 32kB, 256kB, 128k) i organizacja pamięci (8,12,14, 16, 24, 32-bitowa)
  - Czas odczytu (dziesiątki i setki ns), czas i sposób programowania (us i ms), napięcie programowania
  - Napięcie zasilania (2,7V, 3,3V 5V) i pobór prądu (mA)
  - Trwałość zapisu, ilość cykli programowania

## Rodzaje pamięć programu

- **ROM (*Read Only Memory*)** – programowanie zawartości pamięci następuje w procesie produkcyjnym i nie może być przeprowadzone przez użytkownika.
- **EPROM (*Erasable Programmable ROM*)** – pamięci z możliwością kasowania dotychczasowej zawartości promieniami ultrafioletowymi (253,7nm) i prowadzenia nowej zawartości za pomocą zewnętrznego programatora. Umieszczane są w obudowach z okienkiem kwarcowym w celu umożliwienia kasowania.
- **OTP (*One Time Programmable*)** – pamięci typu EPROM umieszczane w obudowach bez okienka kwarcowego. Dlatego możliwe jest tylko jednokrotne zaprogramowanie pamięci bez możliwości skasowania jej zawartości.
- **EEPROM (*Electrical Erasable Programmable ROM*)** – pamięci z możliwością kasowania zawartości i programowania bezpośrednio w systemie mikroprocesorowym
- **FLASH (*Bulk Erasable Non-Volatile Memory*)** – pamięci z możliwością kasowania zawartości i programowania bezpośrednio w systemie mikroprocesorowym.
- **FRAM (*Ferroelectric Random Access Memory*)** jest rodzajem pamięci RAM, w której wykorzystano efekt ferroelektryczny do zapamiętywania bitów słowa danych. Szybki zapis i odczyt jak w RAM, duża trwałość.
- **MRAM** – rodzaj jest rodzajem pamięci nieulotnej RAM wykorzystującej tunelowy efekt magnetorezystancyjny.



## Pamięć danych - RAM

- **Pamięć danych RAM** (*ang. data memory- RAM - Random Access Memory*) – pamięć do zapisu i odczytu przeznaczona do przechowywania danych obliczanych przez jednostkę centralną.
- Podstawowe parametry pamięci ROM:
  - Pojemność (w kB 128kB, 256kB,) i organizacja pamięci (1,4,8,16,32-bitowa)
  - Czas dostępu (czas odczytu i zapisu dziesiątki ns)
  - Napięcie zasilania (2,7V, 3,3V 5V) i pobór prądu (mA)
  - Wymagany cykl odświeżania dla pamięci DRAM (ms)

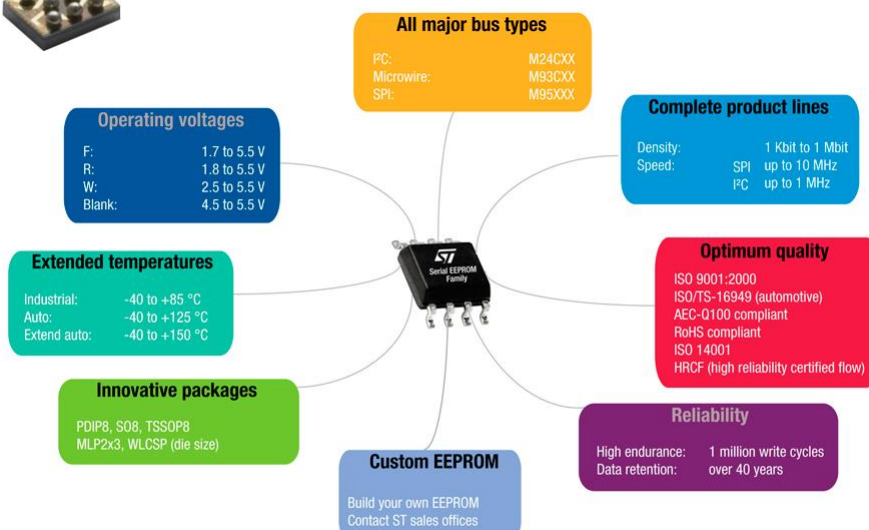
## Pamięć danych - SRAM



semestr zimowy 2009/2010, WIEIK, PK

7

## Szeregowa pamięć EEPROM



semestr zimowy 2009/2010, WIEIK, PK

8

## Rodzaje pamięć danych

- **SRAM (Static Random Access Memory)** – pamięci RAM statyczne. Są to pamięci o krótkich czasach dostępu, prostsze w obsłudze przez jednostkę centralną, ale droższe. Elementem pamięciowym jest przerzutnik D.
- **DRAM (Dynamic Random Access Memory)** – pamięci RAM dynamiczne. Są to pamięci tańsze, ale ich obsługa przez jednostkę centralną jest bardziej skomplikowana. Polega to na konieczności wykonywania w krótkich odstępach czasu określonych operacji na pamięci (tzw. odświeżaniu). W przeciwnym razie dane zawarte w pamięci dynamicznej zanikają. Elementem pamięciowym jest kondensator.
- **FRAM (Ferroelectric Random Access Memory)** jest rodzajem pamięci RAM, w której wykorzystano efekt ferroelektryczny do zapamiętywania bitów słowa danych. Oznacza to technologię wytwarzania nieulotnej pamięci RAM, czyli pamięci łączącej w sobie szybkość pracy RAM i trwałość ROM.


semestr zimowy 2009/2010, WIEIK, PK

9

- T-RAM**, short for "Thyristor RAM" or "thyristor random-access memory", is a new type of [DRAM computer memory](#) invented and developed by [T-RAM Semiconductor](#), which departs from the usual designs of memory cells, combining the strengths of the DRAM and [SRAM](#): high speed and high volume. This technology, which is based on the [NDR](#) and is called Thin-Capacitively-Coupled-Thyristor<sup>[1]</sup>, is used to create memory cells capable of very high packing densities. Due to this, this memory is highly scalable, and already has a storage density that is several times higher than found in conventional six-transistor SRAM memory. It is expected the next generation of T-RAM memory to have the same density as DRAM.
- It is assumed that this type of memory will be used in the next-generation processors by [AMD](#), produced in 32nm and 22nm <sup>[2]</sup>, replacing the previously licensed but unused [Z-RAM](#) technology.
- Z-RAM**, short for "zero capacitor RAM" is a new type of [DRAM computer memory](#) in development by [Innovative Silicon](#) based on the [floating body effect](#) of [silicon on insulator](#) (SOI) process technology. Z-RAM has been licensed by [Advanced Micro Devices](#) for possible use in future [microprocessors](#). Z-RAM offers memory access speeds similar to the standard six-transistor [SRAM](#) cell used in [cache memory](#) but uses only a single [transistor](#), therefore affording much higher packing densities.
- Twin Transistor RAM (TTRAM)** is a new type of [computer memory](#) in development by Renesas.
- TTRAM is similar to conventional one-transistor, one-capacitor [DRAM](#) in concept, but eliminates the [capacitor](#) by relying on the [floating body effect](#) inherent in a [silicon on insulator](#) (SOI) manufacturing process. This effect causes capacitance to build up between the transistors and the underlying [substrate](#), originally considered a nuisance, but here used to replace a part outright. Since a transistor created using the SOI process is somewhat smaller than a capacitor, TTRAM offers somewhat higher densities than conventional DRAM. Since prices are strongly related to density, TTRAM is theoretically less expensive. However the requirement to be built on SOI [fab](#) lines, which are currently the "leading edge", makes the cost somewhat unpredictable at this point.

semestr zimowy 2009/2010, WIEIK, PK

10



**4Mbit Ferroelectric Nonvolatile RAM**

- Organized as 256Kx16
- Configurable as 512Kx8 Using /UB, /LB
- 100 Trillion (1e14) Read/Write Cycles
- NoDelay™ Writes
- Page Mode Operation to 40MHz
- Advanced High-Reliability Ferroelectric Process

**SRAM Compatible**

- JEDEC 256Kx16 SRAM Pinout
- 55ns Access Time, 110ns Cycle Time

**Advanced Features**

- Low VDD Monitor Protects Memory against Inadvertent W
- Software Programmable Block Write Protect

**Superior to Battery-backed SRAM Modules**

- No Battery Concerns
- Monolithic Reliability

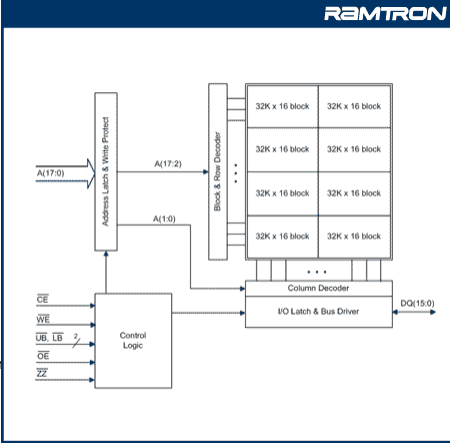
**Low Power Operation**

- 2.7V – 3.6V Power Supply
- Low Standby Current using ZZ pin
- 18 mA Active Current

**Industry Standard Configuration**

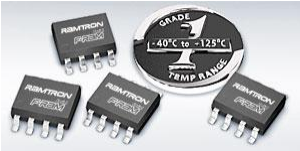
- Industrial Temperature -40° C to +85° C
- 44-pin "Green"/RoHS TSOP-II package
- True Surface Mount Solution, No Rework Steps
- Superior for Moisture, Shock, and Vibration

**Ordering Information**  
FM22L16-55-TG 55ns access, 44-pin "Green"/RoHS TSOP-II



**Pamięć FRAM**

semestr zimowy 2009/2010, WIEiK, PK 11



**16K bit Ferroelectric Nonvolatile RAM**

- Organized as 2,048 x 8 bits
- Unlimited Read/Writes
- NoDelay™ Writes
- Advanced High-Reliability Ferroelectric Process

**Very Fast Serial Peripheral Interface - SPI**

- Up to 15 MHz Frequency
- Direct hardware replacement for EEPROM
- SPI Mode 0 & 3 (CPOL, CPHA=0,0 & 1,1)

**Sophisticated Write Protection Scheme**

- Hardware Protection
- Software Protection

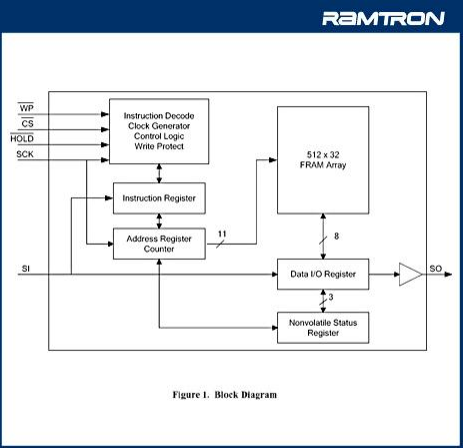
**Low Power Consumption**

- Low Voltage Operation 3.0-3.6V
- 1 µA Standby Current (+85°C)

**Industry Standard Configuration**

- Industrial Temperature -40°C to +125°C
- Grade 1 AEC-Q100 Qualified
- "Green" 8-pin SOIC

**Ordering Information**  
FM25L16-GA ("Green"/RoHS 8-pin SOIC)



**Figure 1. Block Diagram**

semestr zimowy 2009/2010, WIEiK, PK 12

## Układ wejścia-wyjścia – I/O

- **Układ wejścia-wyjścia** (*ang. input-output circuit, I/O*)
  - element systemu, służący do wymiany informacji między jednostką centralną a układami/urządzeniami zewnętrznymi.
- **Wewnętrzne układy wejścia/wyjścia mikrokontrolerów**
  - Układy cyfrowe (logiczne)
  - Układy analogowo-cyfrowe
  - Układy analogowe

semestr zimowy 2009/2010, WIEiK, PK

13

## Rodzaje układ wejścia-wyjścia

- **Układy cyfrowe (logiczne)**
  - Równoległe porty wejścia/wyjścia, pogrupowane w 4 lub 8-bitowe porty,
  - Porty wejściowe logiczne lub wejścia analogowe,
  - Szeregowe porty wejścia/wyjścia typu: UART, (*Universal Asynchronous Receiver-Transmitter*), USART (*Universal Synchronous-Asynchronous Receiver-Transmitter*), I2C (*inter-integrated circuit*), TWI (*Two Wire Interface*), SCI (*Serial Communication Interface*), SPI, (*Serial Peripheral Interface*), QSPI, CAN (*Controller Area Network*), LIN, USB, (*Universal Serial Bus*), kontroler sieci Ethernet moduły radiowe,
  - Programowalne układy czasowo-licznikowe, 8 lub 16-bitowe, liczniki z dzielnikiem wstępnym (*prescalerem*)
  - Generatory PWM, (*pulse width modulation*),
  - PCA – (*Programmable Counter Array*), matryce (*macierz*) liczników, Liczniki typu CCU (*compare capture unit*), Licznik jako zegar typu RTC (*real time clock*),
  - Programowalne struktury logiczne PLD.
- **Układy analogowo-cyfrowe**
  - Analogowy system pomiarowy - multiplexer analogowy 4, 8, 12 wejściowy, przetwornik analogowo-cyfrowy 8, 10, 12 16, 24 bitowe, mogą być dwa niezależne przetworniki A/C, przetworniki A/C 24-bitowe typu Delta-Sigma wraz programowalny filtrem cyfrowym,
  - przetwornik cyfrowo-analogowy 8-10 12-bitowe typu PWM lub klasyczny przetworniki C/A z wyjściem napięciowym, przetworniki Sigma-Delta, ( $\Sigma-\Delta$ )
  - Jeden lub dwa komparatory analogowe, monitor napięcia zasilania MCU (komparator analogowy),
- **Układy analogowe**
  - Programowalny wzmacniacz operacyjny,
  - Programowalny wzmacniacz pomiarowy (PGA) dla przetwornika A/C z wejściem pojedynczym lub różnicowym,
  - Źródło napięcia odniesienia Vref, źródło prądowe do zasilania czujników
  - Czujnik temperatury (pomiar spadku napięcia na diodzie)

semestr zimowy 2009/2010, WIEiK, PK

14

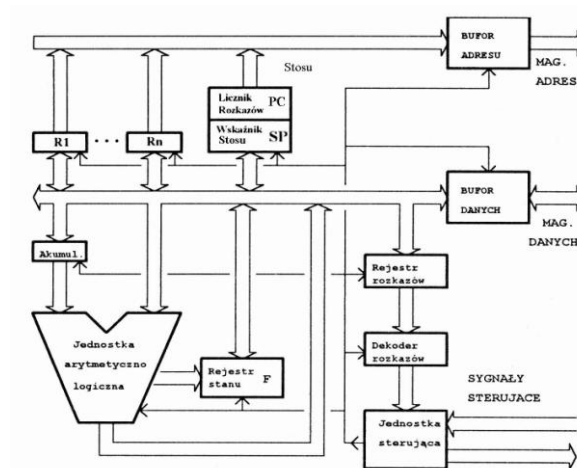
## Jednostka centralna - CPU

- Działanie CPU polega na cyklicznym wykonaniu instrukcji zawartych w programie użytkownika przechowywanym w pamięci programu ROM. Lista instrukcji jest z góry określona dla danego układu mikroprocesorowego.
- Zakłada się, że **instrukcja składa się z kodu operacji nazywanego kodem rozkazowym lub w skrócie rozkazem i argumentu lub argumentów.**
- Cykl wykonania instrukcji rozpoczyna się zawsze od wczytania do wewnętrznych rejestrów CPU kolejnego rozkazu. Gdy niezbędne do wykonania operacji są argumenty, to zawarte są one lub informacja o ich miejscu przechowywania w dalszej części instrukcji.
- W kolejnej fazie następuje pobranie tych argumentów i umieszczenie ich w odpowiednich wewnętrznych rejestrach CPU. Po tym następuje wykonanie instrukcji. Jednocześnie jest inkrementowany **licznik rozkazów wskazujący adres spod którego pobierane są instrukcje.**

semestr zimowy 2009/2010, WIEiK, PK

15

## Jednostka centralna - CPU



Schemat blokowy jednostki centralnej

semestr zimowy 2009/2010, WIEiK, PK

16



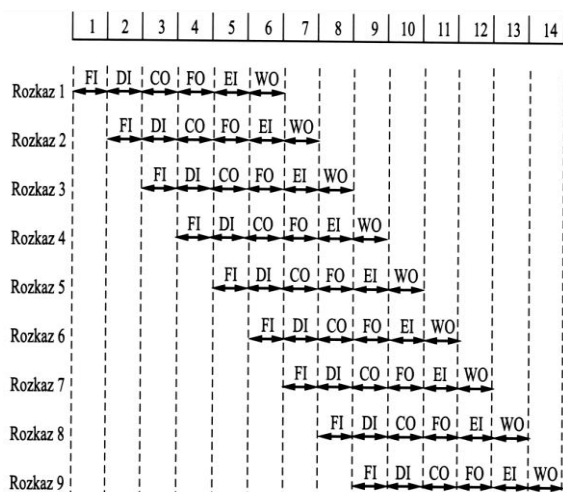
## Potokowe przetwarzanie rozkazów

- **Przetwarzanie potokowe** (*ang. pipelining*) – polega na równoczesnym przetwarzaniu grupy rozkazów, znajdujących się w różnych fazach (*ang. stage*) cyklu rozkazowego. Wyróżnia się co najmniej fazę pobierania i wykonania rozkazu.
- **Przetwarzanie superpotokowe** (*ang. hiper/superpipelining*) – zaawansowane przetwarzanie potokowe, w którym dwa rozkazy znajdują się w tej samej fazie potoku, lecz w różnych podfazach. Takie rozwiązanie jest możliwe, gdy w obrębie fazy zostaną wyodrębnione dwa nienakładające się zadania.
- Przykładowa dekompozycja rozkazu na 6 faz:
  - pobranie rozkazu (*ang. instruction fetch*) – FI,
  - dekodowanie rozkazu (*ang. instruction decoding*) – DI,
  - obliczanie adresu argumentów (*ang. operand computing*) – CO,
  - pobieranie argumentów (*ang. operand fetch*) – FO,
  - wykonanie rozkazu (*ang. instruction execution*) – EI,
  - zapisanie argumentów (*ang. operand writing*) – WO.

semestr zimowy 2009/2010, WIEiK, PK

17

## Potokowe przetwarzanie rozkazów



### Przetwarzanie potokowe 9 rozkazów

semestr zimowy 2009/2010, WIEiK, PK

18

## Pozostałe wewnętrzne elementy funkcjonalne mikrokontrolera

- system przerwain sprzętowych,
- układ do bezpośredniego dostępu do pamięci DMA,
- dodatkowy moduł obliczeniowy o zwiększonej precyzji, dzielenia, mnożenia i przesuwania
- dodatkowy układ arytmetyczno-logiczny, ALU, np. typu MAC (mnożenie i dodawanie),
- akumulator o zwiększonej liczbie bitów (dokładności),
- system nadzorowania pracę mikrokontrolera, (układy napięciowe i czasowe),
- układu do uruchamiania systemu mikroprocesorowego (debuger), JTAG
- układu do programowania pamięci programu ROM, pamięci danych typu EEPROM, pamięci konfiguracyjnej, tryb programowania szeregowy w systemie lub równoległy,
- program ładujący tzw. bootloader do programowania

## Pozostałe wewnętrzne elementy funkcjonalne mikrokontrolera

- **System dystrybucji sygnału zegarowego składa się:**
  - układu generatora sygnału zegarowego (oscylatora),
  - multiplexera do wyboru sygnału zegarowego,
  - dzielników częstotliwości sygnału zegarowego do taktowania poszczególnych fragmentów mikrokontrolera, (CPU, RAM, EEPROM), elementów układu wejścia/wyjścia (przetwornika A/C, liczników), układu zerowania,
- **Typy generatorów sygnału zegarowego:**
  - wewnętrzny układ generatora z dołączanym zewnętrznym rezonatorem kwarcowym lub rezonatorem ceramicznym, (stała częstotliwość pracy),
  - wewnętrzny układ generatora typu PLL z dołączanym zewnętrznym rezonatorem kwarcowym o częstotliwości np. 32,768KHz (możliwość ustalania częstotliwości pracy),
  - wewnętrzny układ generatora typu RC z dołączanymi zewnętrznymi elementami RC, (stała częstotliwość pracy, mała dokładność częstotliwości),
  - wewnętrzny układ generatora typu RC z możliwością ustawiania częstotliwości, (możliwość ustalania częstotliwości pracy, mała dokładność częstotliwości),
  - zewnętrzny sygnał zegarowy (możliwość synchronizacji pracy kilku układów)

## Pozostałe wewnętrzne elementy funkcjonalne mikrokontrolera

- **Do układów nadzorujących (czuwających) nad pracą mikrokontrolerów zaliczamy:**
  - Licznik nadzorujący (czuwający) (watchdog timer) Licznik czuwający może być programowalny lub o stałym podziale częstotliwości, może zliczać impulsy zegarowe lub mieć własny generator,
  - Generator czuwający (oscillator timer)
  - Układ detekcji awarii systemu, lub błędów obliczeń (dzielenie przez zero)
  - Układy monitorowania napięcia, LVD, (*Low Voltage Detector*), BOR (*Brown-out Reset*), POR (*Power on Reset*). Układy do monitorowania napięcia służą do sygnalizowania spadku napięcia zasilania mikrokontrolera poniżej wartości odniesienia. Ta funkcja może być realizowana przez wewnętrzne układy lub zewnętrzne układy.
- **Możemy wyróżnić następujące źródła zerującego (Resetu):**
  - reset po **włączeniu zasilania**,
  - reset wywołany **zewnętrznym sygnałem RESET**,
  - reset **programowy** wywołany przez **ustawienie odpowiedniego bitu**,
  - reset wywołany przez **układ watchdog**,
  - reset wywołany przez **układy nadzorujące** poprawność pracy MCU (np. od układu wykrywającego spadek napięcia zasilania LVD , BOR, POR).

## Architektura mikrokontrolerów podział ze względu na umieszczenie pamięci ROM, pamięci RAM i układów I/O (mapa pamięci)

- **Architektura Von-Neumanna**
- **Architektura Harvard**
- **Zmodyfikowana architektura harwardzka**

## Architektura systemów mikroprocesorowych – mapa pamięci

- Mapa pamięci (*memory map*) w sposób graficzny przedstawia rozmieszczenie poszczególnych pamięci w **przestrzeni adresowej CPU**. **Oprócz adresów obszarów RAM, ROM** i innych rodzajów pamięci, mapa ta podaje usytuowanie rejestrów uniwersalnych, adresów procedur obsługi przerwań, rejestrów układów we/wy (dostępne przez adresowanie pamięci RAM).
- W zależności od typu struktury mapy pamięci, procesory rdzeniowe mogą mieć następującą architekturę:
  - architekturę harwardzką,
  - zmodyfikowaną architekturę harwardzką,
  - architekturę Von-Neumanna.

## Architektura Von-Neumanna

- Architektura przedstawiona po raz pierwszy w 1945 roku przez Johna von Neumanna stworzona wspólnie z Johnem W. Mauchly'ym i Johnem Presper Eckertem.
- **Architektura Von-Neumanna cechuje się jednolitą przestrzenią adresową, w której** wszystkie pamięci, rejestry i układy we/wy są umieszczone w jednej, wspólnej przestrzeni adresowej.
- W architekturze tej zakłada się, że podział przestrzeni adresowej na pamięć programu, pamięć danych oraz obszar we/wy jest czysto umowny i zależy wyłącznie od rozmieszczenia tych elementów w obszarze adresowym podczas projektowania systemu. CPU ma jedną szynę danych wspólną dla danych i programu.
- Dzięki temu programowanie jest ułatwione, gdyż dostęp do danych, programu i urządzeń we/wy odbywa się przy użyciu zunifikowanych rozkazów wykorzystujących te same tryby adresowania. Zatem nie istnieje tu potrzeba wprowadzania specjalnych rozkazów pozwalających na przepływ danych pomiędzy pamięcią ROM i RAM. Do tego celu może być użyty typowy rozkaz adresowy. Tym samym tworzenie tablic stałych, tablicy wektorów, itp. w pamięci ROM nie stanowi problemu.

## Architektura Harwardzka

- **Architektura harwardzka opiera się na użyciu dwóch oddzielnych szyn dla danych i rozkazów**, dzięki czemu w trakcie pobierania argumentów wykonywanej właśnie instrukcji można równocześnie zacząć pobieranie następnego słowa rozkazowego (*pre-fetch*). Skracza to cykl rozkazowy i zwiększa szybkość pracy. Obszary adresowe pamięci danych i programu (wewnętrznych i czasami zewnętrznych) są rozdzielone.
- Pociąga to za sobą niejednoznaczność adresów, ponieważ pod tym samym adresem CPU widzi pamięć RAM i ROM. W tym przypadku stosuje się inne rozkazy dla pamięci programu i inne dla pamięci danych. Ponadto magistrala danych i rozkazów mają **różną szerokość (długość słowa)**

semestr zimowy 2009/2010, WIEiK, PK

25

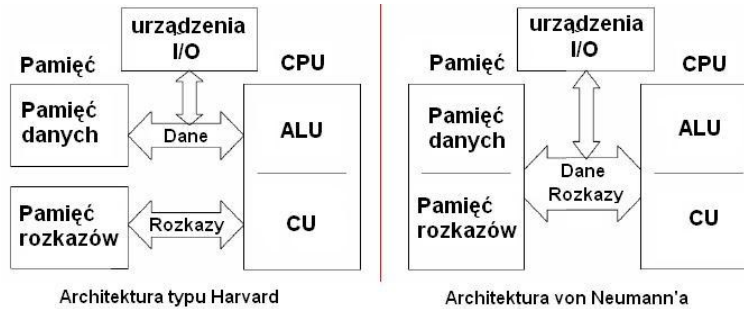
## Architektura Harwardzka

- Wadą tego rozwiązania jest utrudniony przepływ danych z pamięci programu do obszaru pamięci operacyjnej, co uniemożliwia stosowanie jednej z podstawowych technik programistycznych (*look-up tables*).
- *Innymi słowy nie jest możliwe indeksowane przesłanie* danych z pamięci ROM do RAM, co oznacza np. brak możliwości budowy tabel współczynników stałych w pamięci ROM

semestr zimowy 2009/2010, WIEiK, PK

26

## Architektura Harwardzka - Von Neumanna



semestr zimowy 2009/2010, WIEiK, PK

27

### Zmodyfikowana architektura harwardzka

- **Zmodyfikowana architektura harwardzka jest rozwiązaniem pośrednim, starającym się połączyć zalety architektury harwardzkiej i Von-Neumanna.**
- Obszary pamięci ROM i RAM są rozdzielone, ale charakteryzują się **taką samą długością słowa. Dzięki multiplekserom** i odpowiedniej organizacji magistrali pamięci ROM i RAM możliwe jest z pewnymi ograniczeniami przesyłanie stałych z pamięci ROM do rejestrów i pamięci RAM.
- Jedynym rejestrem niewidocznym jako komórka pamięci RAM jest rejestr akumulatora A. **(Mikrokontrolery z rodziny 8051)**

semestr zimowy 2009/2010, WIEiK, PK

28

## Architektura systemów mikroprocesorowych podział ze względu na listę instrukcji

Kolejny podział architektur procesorów można uzyskać korzystając z kryterium typu listy instrukcji. Pozwala ono na wyróżnienie procesorów o:

- architekturze **RISC** (*reduced instruction set computer*),
- architekturze **CISC** (*complex instruction set computer*).

## Architektura RISC (*Reduced Instruction Set Computer*)

Architektura RISC, czyli o zredukowanej liście instrukcji, odznacza się następującymi cechami:

- procesor jest zbudowany zgodnie z architekturą harwardzką,
- procesor wykorzystuje **przetwarzanie potokowe** (*pipelining*) w celu **zwiększenia szybkości** wykonywania programu,
- zbiór realizowanych instrukcji jest **ograniczony i spełnia warunki ortogonalności** (symetrii).

W przetwarzaniu potokowym CPU dysponuje pobranymi na zapas instrukcjami, które będą kierowane do współbieżnego wykonania w jej poszczególnych jednostkach wykonawczych. W procesorze tego typu zamiast prostego rejestru instrukcji stosuje się **pamięć FIFO** (*first-in first-out*), która gromadzi kolejkę instrukcji. Instrukcje pobierane z pamięci programu do kolejki w cyklu **pre-fetch** opuszczają ją w takiej samej kolejności i są kierowane do poszczególnych układów wykonawczych.

**Klasyczna architektura RISC jest stosowana w mikrokontrolerach rzadko. Najczęściej można znaleźć elementy tej architektury, ale ortogonalność instrukcji nie jest pełna.**

## Architektura RISC (*Reduced Instruction Set Computer*)

Pojęcie ortogonalności oznacza unifikację instrukcji według następujących zasad:

- każda instrukcja może operować na **dowolnym rejestrze roboczym**. **Procesor nie ma** więc wyróżnionych rejestrów, które są specjalizowane do wykonywania określonych rodzajów operacji,
- każda instrukcja może wykorzystywać **dowolny tryb adresowania argumentów**, nie ma ukrytych związków między instrukcjami (efektów ubocznych), które powodowałyby nieprzewidziane reakcje systemu w zależności od kontekstu użycia rozkazów w programie,
- kody rozkazów i formaty instrukcji są **zunifikowane**. **W szczególności wszystkie** instrukcje zajmują w pamięci programu taką samą liczbę bajtów.
- Ortogonalność zbioru instrukcji radykalnie upraszcza budowę układu sterowania, który może realizować cykl wykonania każdego rozkazu według identycznego algorytmu. Stąd prostszy układ sterowania może pracować znacznie szybciej, dlatego cykl rozkazowy ulega skróceniu.

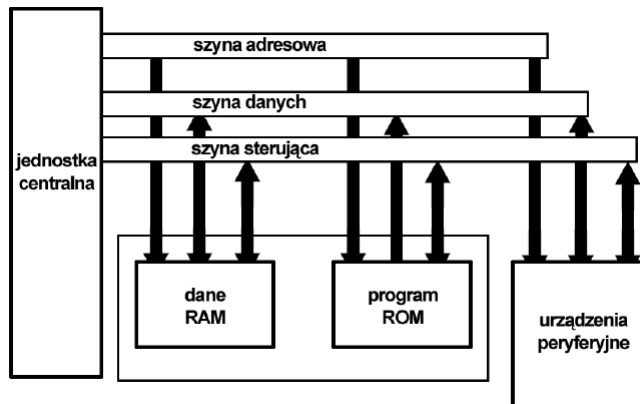
## Architektura CISC (*Complex Instruction Set Computer*)

**Architektura CISC charakteryzuje się:**

- **rozbudowaną liczbą instrukcji (często powyżej 100)**. Przeciwstawia się ją architekturze RISC. Cecha ortogonalności nie jest zachowana.
- Instrukcje są wąsko specjalizowane, współpracują na ogół tylko z określonymi rejestrami i wymagają stosowania określonych trybów adresowania.



## Struktura systemu mikroprocesorowego



### Podstawowa struktura systemu mikroprocesorowego

semestr zimowy 2009/2010, WIEiK, PK

33