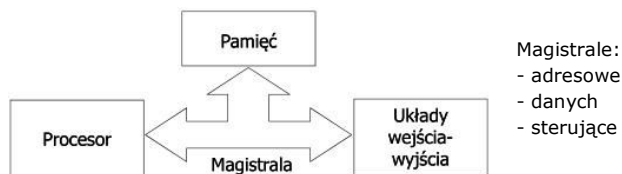


1. Koncepcja architektury komputera:



Magistrale pracują **synchronicznie**, sygnały pojawiają się w ściśle określonych momentach, wyznaczonych przez **cykl zegarowy**.

Przykład **cyklu maszynowego** (pobranie danej z pamięci):

1. Procesor wysyła na magistralę adresową adres komórki pamięci
2. Pamięć dekoduje adres, po czym umieszcza zawartość komórki na magistralę danych
3. Jeśli pamięć jest wolniejsza od procka i nie może wysłać danych w odpowiednim czasie, informuje o tym procesor (sygnał WAIT) poprzez magistralę sterującą
4. Procesor wydłuża czas oczekiwania o cykl zegarowy i w określonym momencie pobiera dane z magistrali danych

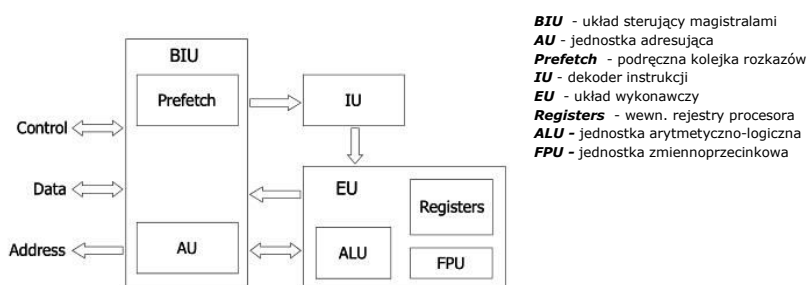
Cykle maszynowe są wielokrotnością cyklu zegarowego.

Komunikacja procesora z urządzeniami IO odbywa się w postaci analogicznych cykli maszynowych.

Komunikacja IO z pamięcią odbywa się w podobny sposób w przypadku transmisji za pomocą **kanalu bezpośredniego dostępu do pamięci**.

Cykl rozkazowy - składa się z kilku cykli maszynowych, w których występuje wykonanie pełnego rozkazu procesora

2. Ogólna zasada działania procesora



BIU - układ sterujący magistralami
AU - jednostka adresująca
Prefetch - podręczna kolejka rozkazów
IU - dekodery instrukcji
EU - układ wykonawczy
Registers - wewn. rejestry procesora
ALU - jednostka arytmetyczno-logiczna
FPU - jednostka zmiennoprzecinkowa

Zadaniem **BIU** jest komunikacja procesora ze światem zewnętrznym.

AU ma wytworzyć na magistrali adresowej adresy z danymi lub rozkazami.

Pobrane rozkazy umieszczane są w **Prefetch'u**, której uzupełnianie jest automatyczne. **IU** dekoduje rozkazy pobrane z Prefetch'a.

Rozkodowane instrukcje trafiają do **EU**, który w zależności od rodzaju rozkazu wydaje odpowiednie polecenia do AU oraz ALU/FPU.

Wykonywane rozkazy mogą dotyczyć manipulacji na **rejestrach** wewnętrznych procesora lub na komórkach pamięci operacyjnej.

ALU zajmuje się bezpośrednimi operacjami na danych (dodawanie/odejmowanie itd.)

W przypadku rozkazów na danych zmiennoprzecinkowych ich przetwarzaniem zajmuje się **FPU**.

W każdej komórce pamięci mogą znajdować się dane lub rozkazy.

W celu zwiększenia efektywności procesor posiada rejestry (szybka pamięć | RAM - wolna)

Wykonanie operacji na pamięci powoduje przesłanie zawartości do rejestru, wykonania operacji i odesłania wyniku do pamięci.

Rejestry:

a) **ogólnego przeznaczenia** (np. AX, BX, CX, DX) - przechowywanie danych:

AX - akumulator (op. arytmetyczne i logiczne) | **BX** - adresowanie pamięci | **CX** - licznik w pętach | **DX** - transfer danych z portów IO

b) **segmentowe** (np. CS, DS, SS) - przechowywanie:

CS - adresu kodu do wykonania | **DS** - adresu danych | **SS** - adresu stosu

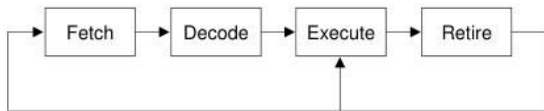
c) **statusowe i sterujące** (np. FLAGS, IP) - kontrola pracy:

FLAGS - posiada flagi takie jak: **SIGN FLAG, ZERO FLAG, CARRY FLAG, OVERFLOW**

IP - wskaźnik rozkazów (przemieszczenie wykonywanej instrukcji względem początku segmentu kodu)

O tym, że wartość komórki w pamięci zostanie zinterpretowana jako rozkaz (a nie dana) decyduje fakt jej zaadresowania przez IP. W cyklu rozkazowym można wyróżnić 4 fazy:

- **pobranie rozkazu (fetch)** - pobranie rozkazu z Prefetch'a i (jeśli trzeba) uruchomienie procedury uzupełniania kolejki
- **dekodowanie rozkazu (decode)** - określenie operacji do wykonania oraz obliczenie efektywnych adresów argumentów (jeżeli występują)
- **wykonanie (execute)** - pobranie argumentów rozkazu z pamięci oraz wykonanie operacji ustalonych w fazie dekodowania
- **zapisanie wyników (retire, write-back)** - zapisanie wyniku w określonym w kodzie rozkazu miejscu oraz ustawienie znaczników rejestru statusowego



3. Obsługa urządzeń zewnętrznych

Urządzenie zewnętrzne komunikuje się z układem IO za pośrednictwem interfejsu, tworząc wspólnie **urządzenie peryferyjne**.

Funkcje jakie realizuje układ IO:

- komunikacja z urządzeniem zewnętrznym, polegająca na **transferze danych, wysyłanie rozkazów sterujących i odczytywanie informacji o jego stanie**
- komunikacja z procesorem, obejmująca **dekodowanie adresu układu, wymianę danych, przesyłanie rozkazów, informowanie o stanie układu**
- buforowanie i formatowanie danych
- wykrywanie i informowanie o błędach

Przestrzeń adresowa układów IO:

- **odzworowana w pamięci** - układy IO i komórki pamięci **współdzielą tę samą** przestrzeń adresową
- **izolowana** - układy IO i komórki pamięci korzystają z **niezależnych** przestrzeni

Typowe układy IO:

- szeregową i równoległą transmisję danych
- timery
- zegary czasu rzeczywistego
- układy ADC i DAC

Sterowanie urządzeniami zewnętrznymi polega na wpisaniu do odpowiedniego portu określonej wartości. Transfer danych między urządzeniem peryferyjnym a pamięcią może odbywać się w trybie PIO lub w trybie DMA.

PIO (programowane wejście/wyjście) - transferem danych zajmuje się procesor

DMA (bezpośredni dostęp do pamięci) - procesor przechodzi w **stan wysokiej impedancji**, a na czas transmisji kontrolę nad magistralami przejmuje układ DMA

Urządzenia mogą informować procesor o zmianie stanu za pomocą **przerwań**.

Zgłoszenie przerwania powoduje:

- przerwanie bieżącego programu
- wykonanie obsługi przerwania
- powrót do poprzedniej czynności.

Alternatywną metodą sprawdzenia stanu urządzenia jest cykliczne odpytywanie przez procesor w oczekiwaniu na spełnienie warunków koniecznych do realizacji transmisji (**polling**).

4. Podstawowe własności i klasyfikacje pamięci

Podstawowe własności pamięci:

- pojemność** - wyrażana w bajtach lub słowach
- jednostka transferu** - liczba bitów jednocześnie odczytywanych z pamięci lub do niej zapisywanych
- sposób dostępu do danych**
- wydajność**
- technologia wykonania**

Podział pamięci ze względu na **sposób dostępu**:

- sekwencyjny** - dostęp jest możliwy w określonej sekwencji liniowej (np. pamięć taśmowa)
- bezpośredni** - możliwy jest bezpośredni dostęp do bloku danych, natomiast dostęp do danej w bloku odbywa się sekwencyjnie (np. pamięć dyskowa)
- swobodny** - poszczególne dane mają unikatowy adres i dostęp do nich nie zależy od sekwencji poprzednich operacji dostępu (np. pamięć operacyjna)
- skojarzeniowy** - rodzaj swobodnego, w którym adresowanie danych odbywa się na zasadzie porównywania wybranych bitów słowa z zawartością pamięci (np. pamięć podręczna)

Wydajność pamięci określana jest m.in. przez:

- czas dostępu** - czas od doprowadzenia adresu do momentu udostępnienia danych
- czas cyklu pamięci** - czas między dwiema operacjami dostępu do pamięci
- szybkość przesyłania** - szybkość z jaką dane mogą być odczytywane/zapisywane

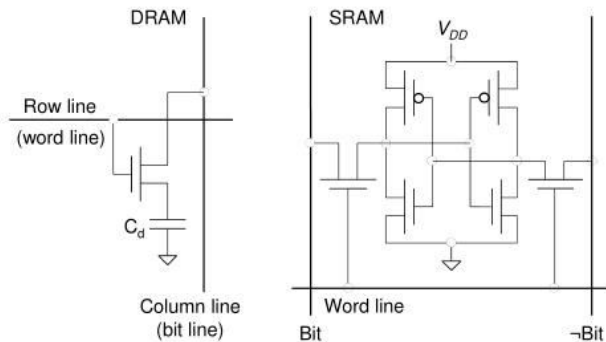
W zależności od **technologii wykonania** mamy pamięci:

- a) **półprzewodnikowe** - RAM, NVRAM, ROM, PROM, EPROM, EEPROM, FLASH
- b) **optyczne**
- c) **magnetyczne**

SRAM (Static RAM) - komórki zbudowane są z dwustanowych przerzutników

DRAM (Dynamic RAM) - komórki zbudowane są z układów przechowujących ładunek (kondensator + tranzystor) i wymagających okresowego odświeżania ładunku w celu zachowania danych

Komórki pamięci RAM:

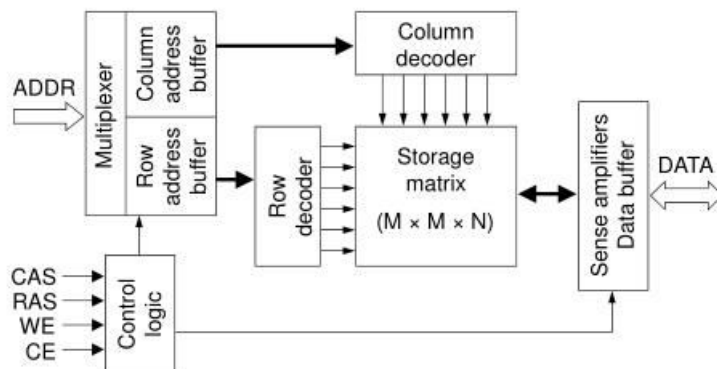


5. Pamięć operacyjna

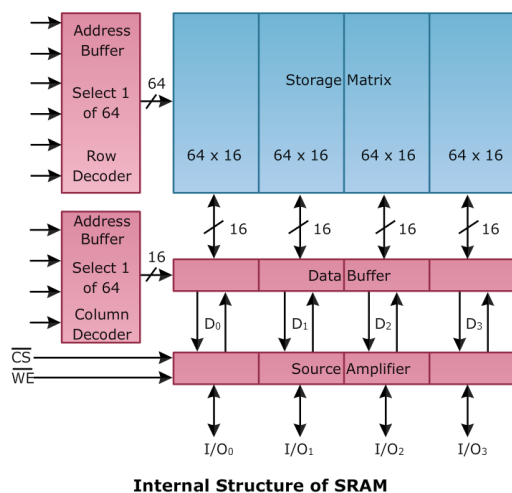
Linie sygnałowe pamięci DRAM:

- a) **adresy słowa:** $A_0 - A_{m-1}$ (maksymalnie 2^{2^m} słów do zaadresowania)
- b) **dane słowa:** $D_0 - D_{n-1}$ (n linii danych)
- c) **CE (Chip Enable)**
- d) **WE (Write Enable)**
- e) **OE (Output Enable - zezwolenie na odczyt)**
- f) **RAS:** wybór adresu wiersza
- g) **CAS:** wybór adresu kolumny

Organizacja wewnętrzna pamięci DRAM:



Organizacja wewnętrzna pamięci SRAM:



W celu prawidłowego odświeżania komórek pamięci, konieczne jest cykliczne pobudzenie (adresowanie) poszczególnych wierszy matrycy pamięci z częstotliwością nie mniejszą niż tzw. **czas cyklu odświeżania**. Odświeżanie pamięci realizowane jest albo przez **kontroler pamięci**, albo przez **integralną część** samego układu pamięci.

Istnieją cztery podstawowe sposoby odświeżania DRAM:

- a) **sygnałem RAS**
- b) **CAS przed RAS**
- c) **ukryte**
- d) **autoodświeżanie**

SDRAM - różni się od typowego DRAM sposobem sterowania matrycą pamięci oraz techniką dostępu. Charakterystyczne dla tego typu jest **synchronizowanie** wszystkich sygnałów sterujących specjalnym zegarem (CK), co ułatwia współpracę z magistralami komputera. SDRAM jest optymalizowana do przesyłania pakietowego.

SDRAM zbudowany jest z niezależnych od siebie banków (najczęściej **2 lub 4**), które adresowane są naprzemiennie (przyspieszenie dostępu). Wbudowany licznik odświeżania i układy logiczne pozwalają na autoodświeżanie. Przed użyciem SDRAM trzeba zaprogramować, tj. wpisać do specjalnego rejestru (Mode Register) informacji konfiguracyjnych.

DDR (double data rate) - w każdym taktie zegara przekazują **dwa słowa danych**

Do takiego układu przekazywane są **dwa przebiegi zegarowe (CLK/~CLK)**, których przecięcie wyznacza moment kluczowania. DDR wykorzystuje **dwukierunkowy sygnał strobowy (DQS)**, który przy odczycie generowany jest przez układ pamięci, a przy zapisie przez kontroler pamięci. Dane odczytywane są w środku impulsu DQS, a zapisywane na zboczach tego impulsu.

RDRAM (Rambus DRAM) - do komunikacji wykorzystują dedykowaną magistralę RAMBUS przesyłającą dane w trybie pakietowym. Pojedynczy kanał magistrali pozwala na dołączenie **max. 32** układów pamięci. Taktowanie magistrali (400 MHz), szerokość 16 bitów danych (podzielone na dwie 8-bitowe szyny: DQA i DQB). Transmisja danych z szybkością **1.6 GB/s**. Informacja w RDRAM przechowywana jest w typowych blokach SDRAM zorganizowanych w niezależne banki (zwykle **4-32**) o szerokości najczęściej **128 bitów**. Przesyłanie danych wykorzystując bardzo szybkie (de)multiplexery. Do synchronizacji wykorzystuje dwie pary komplementarnych sygnałów zegarowych **CTM (Clock to Master)** i **CFM (Clock from Master)**. Dane odczytywane są w trakcie narastającego i opadającego zbocza CTM, a zapisywane w trakcie narastającego i opadającego zbocza CFM. RDRAM posiada 8-bitową magistralę adresową: 3 bity na adres wiersza i 5 bitów na adres kolumny. Przed użyciem wymaga zaprogramowania (wpisania odpowiednich informacji konfiguracyjnych do rejestrów sterujących).

Z uwagi na **możliwość przekłamań** układy pamięci i kontrolery wyposaża się w mechanizmy, które wykrywają (i naprawiają) błędy. Najprostszym mechanizmem wykrywającym jest **kontrola parzystości**. Jej realizacja wymaga zapamiętania dodatkowego bitu, którego wartość jest dobierana tak, aby liczba jedynek w słowie danych była parzysta (XOR). Pozwala to na wykrycie dowolnej nieparzystej liczby błędnych bitów danych.

Bardziej zaawansowane są **kody korekcyjne (ECC)** bazujące głównie na **kodowaniu Hamminga**. Najprostsza wersja tego kodu polega na generowaniu bitów parzystości dla odpowiednio dobranych grup bitów danych.

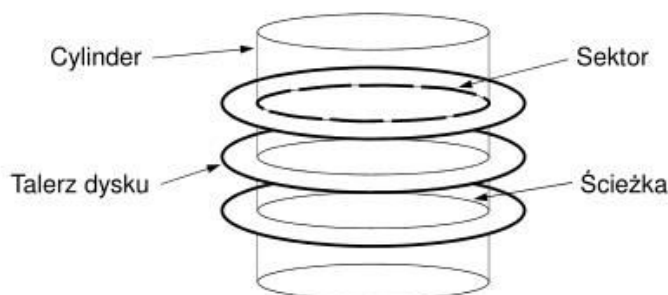
6. Pamięć dyskowa

Nośnikiem jest krążek z tworzywa pokrytego warstwą magnetyczną. W trakcie zapisu przez głowicę płynie prąd, tworząc pole magnetyczne namagnesowujące nośnik. W trakcie odczytu nośnik wytwarza zmienne pole magnetyczne wzbudzające w głowicy prąd elektryczny. Ruch obrotowy dysku i pozycjonowanie głowicy pozwala na dostęp do dowolnego sektora danych.

Aby w prawidłowy sposób odtworzyć zapisane dane, informacja użyteczna musi zostać "wzbogacona" o informację synchronizującą wynikającą z przyjętego **sposobu kodowania** (FM, MFM, RLL(2,7)).

Mechanizm głowicy może być **kontaktowy**, mieć **ustaloną odległość** od nośnika lub odległość ta **może być zmienna** w wyniku oddziaływań aerodynamicznych.

Dane organizowane są w **cylindry** będące zbiorem koncentrycznych okręgów o tym samym promieniu, zwanych **ścieżkami**, które dzielone są na **sektory**.



Początek każdej ścieżki oznaczony jest:

- a) **ciągami bajtów synchronizujących (SYNC)**
- b) **znacznikiem adresowym indeksu (IAM)**

W sektorze można wyszczególnić:

- a) **ciąg bajtów synchronizujących (SYNC)**
- b) **znacznik adresowy (IDAM)**
- c) **identyfikator sektora (ID)**
- d) **znacznik adresowy danych (DAM)**
- e) **blok danych sektora**
- f) **poła kodu ECC/CRC**

Znacznik początku ścieżki oraz sektory rozdzielone są od siebie przerwami (**GAP**).

Adres fizyczny sektora na dysku (**CHS**) określają parametry:

- **numer cylindra**
- **numer głowicy**
- **numer sektora**

Czas dostępu do danych zależy od:

- **czasu przeszukiwania** (pozycjonowanie głowicy nad ścieżką)
- **czasu opóźnienia obrotowego** (przemieszczenie głowicy nad sektor)

Gęstość zapisu zależy od:

- **szerokości ścieżki**
- **sposobu kodowania danych**

RAID - macierz dysków, które widoczne są przez OS jako jeden dysk logiczny

Dane zapisywane w macierzy rozpraszane są na wiele dysków. W praktyce stosuje się różne poziomy RAID, oznaczane **0-5**.

RAID 0 (stripping):

- a) dane dzielone na "paski", rozmieszczane cyklicznie na kolejnych dyskach
- b) brak redundancji danych
- c) duża szybkość odczytu i zapisu
- d) pojemność macierzy równa sumie pojemności dysków

RAID 1 (mirroring):

- a) dane są paskowane i następnie odwzorowywane na dwóch odrębnych dyskach
- b) średnia szybkość odczytu, mała zapisu
- c) pojemność macierzy równa połowie pojemności dysków

RAID 2:

- a) dane paskowane (o rozmiarze mniejszym niż RAID 0/1) i rozmieszczane cyklicznie na kolejnych dyskach macierzy
- b) na pozostałych dyskach (redundancyjnych) umieszczone są informacje korekcyjne, utworzone w oparciu o kod Hamminga na podstawie odpowiednich bitów z każdego dysku
- c) liczba dysków redundancyjnych proporcjonalna do logarytmu liczby dysków danych
- d) duża szybkość odczytu i zapisu (równoległy dostęp)
- e) bardzo wysoki koszt

RAID 3:

- a) dane paskowane (jak w RAID 2)
- b) na JEDNYM dysku redundancyjnym są informacje korekcyjne utworzone w oparciu o bity parzystości wyliczone dla odpowiednich bitów z każdego dysku
- c) duża szybkość odczytu i zapisu (równoległy dostęp)

RAID 4:

- a) dane paskowane (paski o dużym rozmiarze) i rozmieszczane cyklicznie na kolejnych dyskach macierzy
- b) na JEDNYM dysku redundancyjnym umieszczone są paski informacji korekcyjnych utworzone w oparciu o bity parzystości, wyliczone dla odpowiednich pasków z każdego dysku
- c) dyski działają niezależnie od siebie, co pozwala na obsługę wielu żądań IO równolegle
- d) średnia szybkość odczytu, mała zapisu

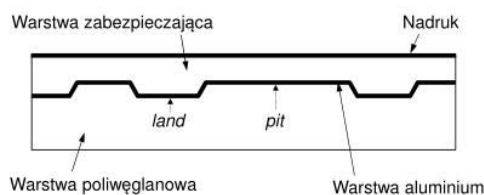
RAID 5:

- a) organizacja zbliżona do RAID 4
- b) paski parzystości są rozproszone na wszystkie dyski w macierzy
- c) średnia szybkość odczytu/zapisu, ale lepsza niż w RAID 1

7. Pamięć optyczna

W dyskach **CD-ROM** nośnikiem jest przezroczysta warstwa poliwęglanowa z wytłoczonymi wgłębieniami, pokryta cienką warstwą aluminium i zabezpieczona od góry lakierem. Promień lasera odbijając się od warstwy aluminium w obszarze zwanym landem ulega mniejszemu rozproszeniu niż odbijając się w obszarze pit. Po odbiciu promień lasera pada przez półprzepuszczalny pryzmat i układ soczewek na światłoczułe detektory, których zadaniem jest:

- a) **przetworzenie zmian natężenia światła na sygnał elektryczny**
- b) **wytworzenie sygnałów sterujących układem ogniskowania wiązki**
- c) **wytworzenie sygnałów sterujących układem pozycjonowania wiązki**



Podczas odczytu promień lasera przemieszcza się nad spiralną ścieżką z szybkością zależną od prędkości wirowania płyty. W praktyce stosuje się dwie metody odczytu danych:

- a) **ze stałą prędkością liniową (CLV)**
- b) **ze stałą prędkością kątową (CAV)**

Informacje na płycie kodowane są metodą **EFM (Eight to Fourteen Modulation)**, w której każdy bajt zastępowany jest ciągiem 14 bitów tak, aby dwie kolejne jedyńki były rozdzielone nie mniej niż dwoma i nie więcej niż dziesięcioma zerami (2 pity na bajt).

Dane zorganizowane są fizycznie w ramki o strukturze:

- ciąg 27 bitów synchronizujących
- bajt sterujący
- dwa 16-bajtowe bloki (12 bajtów danych | 4 bajty kontrolne)

Grupa połączonych 98 ramek tworzy logiczny sektor na który składa się:

- ciąg 12 bajtów synchronizujących
- 4-bajtowy nagłówek
- blok 2048 bajtów danych
- blok 288 bajtów kontrolno-korekcyjnych

Pojedynczy sektor zawiera 2048 bajtów danych zapisane w trybie 1 (korekcja błędów) lub 2336 bajtów w trybie 2 (brak korekcji). Bloki bajtów kontrolno-korekcyjnych połączone z odpowiednim rozlokowaniem kolejnych ramek umożliwiają korekcję błędów metodą **CIRC** która pozwala na skorygowanie ciągu około 450 błędnych bajtów.

Typowa płyta CD-ROM o średnicy 12 cm może pomieścić 650-700 MB.

Płyty **CD-R** (jednokrotnego zapisu) oraz **CD-RW** (wielokrotnego zapisu).

Warstwa poliwęglanowa pokryta specjalną substancją, która mętnieje (zmienia postać z krystalicznej w amorficzną) pod wpływem naświetlania laserem o zwiększonej mocy (powoduje zmniejszenie natężenia wiązki odbitej w trakcie odczytu). Substancja w CD-RW może zmienić się z powrotem w postać krystaliczną.

Dzięki zastosowaniu lasera o mniejszej długości fali (gęstsze upakowanie danych) oraz dwuwarstwowej strukturze nośnika **DVD** (pierwsza półprzezroczysta) uzyskano możliwość zapisania większej ilości informacji niż na CD. W trakcie odczytu wiązka lasera jest ogniskowana na pierwszej lub drugiej warstwie. Specyfikacja płyt:

- a) jednostronne jednowarstwowe - 4.7 GB
- b) jednostronne dwuwarstwowe - 8.5 GB
- c) dwustronne jednowarstwowe - 9.4 GB
- d) dwustronne dwuwarstwowe - 17 GB

Format danych jest podobny do formatu na CD, ale został ulepszony mechanizm korekcji.

8. Interfejs IDE (EIDE)

Zadaniem interfejsu **EIDE (rozszerzenie IDE)** jest podłączenie do komputera takich urządzeń zewnętrznych jak dyski twarde, napędy (magneto)optyczne. Cechą urządzeń sterowanych przez ten interfejs jest to, że **zintegrowano** w nich **mechanizm napędu** wraz z **kontrolerem** odpowiedzialnym m.in. za: pozycjonowanie głowic, kodowanie/dekodowanie sygnału, kontrolę i korekcję błędów, translację adresów, przesyłanie danych.

Standardy IDE (AT-BUS):

- a) ATA-1 (Advanced Technology Attachment):
 - transmisja w trybie PIO
 - do 4 MB/s
 - obsługa max. 2 dysków twardych
- b) ATA-2:
 - transmisja w trybach PIO i DMA
 - do 16.6 MB/s
- c) ATA-3:
 - system autodiagnostyki S.M.A.R.T.
- d) ATA/ATAPI-4:
 - transmisja w trybach PIO, DMA i Ultra-DMA
 - do 33 MB/s
 - interfejs logiczny ATAPI pozwalający na obsługę np. napędów CD
 - obsługa max. 4 urządzeń zewnętrznych
- e) ATA/ATAPI-5:
 - do 100 MB/s
- f) ATA/ATAPI-6:
 - do 133 MB/s
 - 48-bitowy model adresowania LBA
 - funkcje AAM

Magistrala IDE połączona jest z magistralą systemową przez kontroler IDE, którego zadaniem jest dekodowanie adresu oraz buforowanie sygnałów na magistralach. IDE pozwala na dołączenie dwóch urządzeń do kontrolera: Master i Slave, wykorzystując 40, 50 lub 80-żyłowy przewód o długości do 46 cm. W przypadku EIDE wykorzystuje się dwa kanały IDE: 0 (primary) i 1 (secondary), dzięki czemu jest możliwe podłączenie czterech urządzeń. Kanały kontrolera działają niezależnie, do komunikacji z systemem wykorzystują oddzielne przerwania i adresy IO rejestrów kontrolno-sterujących.

Specyfikacja ATA/ATAPI pozwala na obsługę urządzeń z podstawowymi lub zaawansowanymi funkcjami zarządzania poborem mocy. Cztery tryby pracy urządzenia: gotowości, jałowy, czuwania, uśpienia. W trybie zaawansowanym 254 poziomów poboru energii. Urządzenia IDE z protokołem ATAPI mogą implementować:

- system zabezpieczeń chroniący dostęp dwoma hasłami (admin / user)
- system autodiagnostyki S.M.A.R.T. do kontroli krytycznych parametrów dysku
- system zarządzania parametrami akustycznymi (szybkość a poziom hałasu)

9. Interfejs Serial ATA (SATA)

SATA służy do podłączenia urządzeń takich jak przy EIDE. Cechy SATA:

- transfer z szybkością od 1.5 Gb/s (SATA 1)
- programowa zgodność z ATA/ATAPI
- niższy pobór mocy (niższe napięcie magistrali)
- topologia gwiazdy (urządzenie połączone z kontrolerem własnym przewodem)
- zwiększona odporność na błędy
- możliwość zmiany konfiguracji przyłączonych urządzeń w trakcie pracy

SATA wykorzystuje 7-żyłowy przewód połączeniowy (cztery linie sygnałowe, trzy linie masy) o długości mniejszej niż metr. Przesyłane dane zorganizowane są w 32-bitowe słowa, kodowane regułą 8b/10b (NRZ), co zapobiega jednostronnej polaryzacji kabla. Wymiana między kontrolerem i urządzeniem odbywa się za pomocą:

- symboli
- ramek FIS

Symbol jest pojedynczym słowem, którego zadaniem jest przesłanie:

- informacji o stanie magistrali i urządzenia
- sygnatury oznaczającej początek (SOF) lub koniec (EOF) przesyłanej ramki
- znaku synchronizującego (SYNC) w stanie spoczynkowym magistrali

Ramka FIS ma zmienną długość i jest zbudowana z:

- nagłówka o długości 1 słowa
- bloku danych użytkowych do 2048 słów
- kodu CRC o długości 1 słowa

Ramki FIS służą do przesyłania danych użytkowych, diagnostycznych i rozkazów programujących rejestry trybu pracy SRB. SATA posiada symetryczny protokół transmisji danych według którego każda odebrana ramka wymaga przesłania zwrotnie symbolu potwierdzającego odbiór ramki (ROK) lub błąd transmisji (RERR). Transfer może odbywać się w trybie PIO lub DMA. Rozwój SATA:

- a) SATA I:
 - transfer 1.5 Gb/s
- b) SATA II:
 - transfer 3 Gb/s
 - kolejkovanie zadań NCQ
 - możliwość przyłączenia wielu urządzeń do jednego portu dzięki powielaczom portów
 - redundancja połączeń dzięki selektorom portów
- c) SATA III:
 - transfer 6 Gb/s
 - ulepszony NCQ
 - tryb izochorycznego transferu danych
- d) eSATA:
 - 3 Gb/s dla urządzeń zewnętrznych do 2m
- e) xSATA:
 - eSATA na odległość do 8m

10. Interfejs SCSI

SCSI powstał do obsługi urządzeń zewnętrznych przez minikomputery i wydajne stacje robocze. Magistrala została opracowana jako szybka, uniwersalna, elastyczna i niezawodna magistrala równoległa obsługująca wiele urządzeń przyłączonych do jednego kontrolera montowanych wewnątrz lub zewnątrz obudowy. Standardy SCSI:

- a) **SCSI-1**:
 - magistrala o szerokości 8 bitów
 - adresowanie do 8 urządzeń
 - transmisja do 5 MB/s
 - przewód do 6m
- b) **SCSI-2**:
 - magistrala o szerokości 16/32 bitów
 - adresowanie do 16 urządzeń
 - transmisja do 10, 20 lub 40 MB/s
 - przewód do 25m

c) **Ultra3 (160) SCSI / Ultra4 (320) SCSI:**

- transfer do 160 / 320 MB/s
- pełna kontrola integralności danych kodami CRC (Ultra 4)
- obsługa protokołu pakietowego (Ultra 4)
- adresowanie do 32 urządzeń
- protokół diagnostyczny jakości łącza
- przyspieszenie arbitrażu w dostępie do magistrali

d) **Ultra 640 SCSI:**

- transfer do 640 MB/s

Każde urządzenie podłączone do magistrali ma przydzielony unikalny identyfikator określający adres i priorytet. Najczęściej najwyższy priorytet ma kontroler SCSI. Urządzenia przyłączone są równolegle, przy czym końce magistrali muszą być wyposażone w **terminatory**, mostki rezystorowe zapewniające dopasowanie falowe. Szerokość magistrali:

- 50 linii sygnałowych dla 8-bitowej magistrali danych
- 68 linii sygnałowych dla 16-bitowej magistrali danych

Magistrala SCSI pozwala na niezależną komunikację między różnymi parami urządzeń do niej podłączonych (kanały logiczne).

Fazy pracy:

- a) **wolnej magistrali** - żadne urządzenie nie korzysta z magistrali
- b) **arbitrażu** - dokonywany jest wybór urządzenia, które przejmie kontrolę nad magistralą
- c) **selekcji** - urządzenie inicjujące dokonuje wyboru urządzenia docelowego
- d) **rozkazowa** - przesyłane są rozkazy do urządzenia docelowego
- e) **reselekcji** - nawiązanie połączenia przez urządzenie docelowe z inicjującym
- f) **przesyłania danych** - przesył danych między urządzeniami
- g) **przesyłania raportu** - przesył informacji o sposobie zakończenia wykonanej operacji
- h) **przesyłania wiadomości** - przesył informacji pomocniczych
- i) **zgłaszania przesłania wiadomości** - urządzenie inicjujące zgłasza żądanie wysłania wiadomości
- j) **zerowania** - zerowanie wszystkich urządzeń na magistrali

Przesył danych w trybie **(a)synchronicznym**. Rozkazy SCSI przesyłane w blokach CDB o długości 6, 10 lub 12 bajtów zawierających:

- kod operacji
- parametry rozkazu
- bajt kontrolny

11. Interfejs IEEE-1284

Przesył danych za pośrednictwem łącza **równoległego** (drukarki/skanery).

Pięć trybów pracy łącza:

- kompatybilny
- półbajtowy
- bajtowy PS/2
- EPP
- ECP

Pierwsze trzy często określa się także jako tryb SPP. Oprócz 8-bitowej magistrali danych występują też linie sygnałowe:

- **wejściowy błąd drukarki (ERR)**
- **wyjściowy strobujący dane (STR)**
- **wejściowy potwierdzenia (ACK)**
- **wejściowy zajętości (BSY)**
- **wejściowy braku papieru (PERR)**
- **wejściowy gotowości (SEL)**
- **wyjściowy wysuwu wiersza (ALF)**
- **wyjściowy zerowania drukarki (INI)**
- **wyjściowy wyboru drukarki (DSL)**

Tryb kompatybilny - jednokierunkowa transmisja do 200 kB/s

Tryb (pół)bajtowy - dwukierunkowa transmisja

Dostęp do łącza w trybie SPP odbywa się za pośrednictwem:

- a) **rejestr danych** - umożliwia zapis/odczyt z magistrali danych interfejsu
- b) **rejestr stanu** - pozwala na odczyt linii BSY, ACK, PERR, SEL, ERR
- c) **rejestr sterującego** - pozwala na aktywację dwukierunkowej pracy portu danych oraz na ustawianie stanu linii DSL, INI, ALF, STR

Protokoły transmisji danych realizowane są programowo.

Tryb **EPP** pozwala na dwukierunkową transmisję do 2 MB/s.

Protokół wymiany danych wykorzystuje linie sygnałowe:

- **kierunku transmisji (WRITE / STR)**
- **wyzwolenia przerwania (INTERRUPT / ACK)**
- **zakończenia cyklu EPP (WAIT / BSY)**

- **strobujący dane (DATA STROBE / ALF)**
- **strobujący adres (ADDR STROBE / DSL)**
- **zerowania (RESET / INI)**

Dostęp do łącza w trybie EPP odbywa się za pośrednictwem rejestrów SPP plus:

- a) **dwukierunkowy rejestr danych**
- b) **dwukierunkowy rejestr adresowy**

Wytworzenie właściwych sekwencji sygnałów sterujących związanych z realizacją protokołu transmisji danych realizowane jest automatycznie przez kontroler łącza równoległego. Transmisję danych/adresów realizuje się przez bezpośrednie wpisanie do odpowiednich rejestrów EPP.

Tryb **ECP** pozwala na dwukierunkową transmisję do 2 MB/s.

Protokół wymiany danych wykorzystuje linie sygnałowe:

- **zegarowe (HostCLK / STR)**
- **zegarowe (PeriphCLK / ACK)**
- **sterujące przesyłaniem danych/rozkazów (PeriphACK / BSY)**
- **sterujące przesyłaniem danych/rozkazów (HostACK / ALF)**
- **zmiany kierunku transmisji (RevREQ / INI .. AckREV / PERR)**

Tryb ECP pozwala na korzystanie z:

- a) przesyłania danych z kontrolera w trybie DMA
- b) dodatkowego rejestru buforowanego FIFO
- c) kompresji przesyłanych danych algorytmem RLE
- d) definiowania i adresowania do 128 kanałów logicznych

Dostęp do łącza w trybie ECP odbywa się za pośrednictwem rejestrów SPP plus:

- a) **rejestr adresowy FIFO**
- b) **rejestry danych FIFO**
- c) **rejestr sterujący ECR**

Wytworzenie właściwych sekwencji sygnałów sterujących związanych z realizacją protokołu transmisji danych realizowane jest automatycznie przez kontroler łącza równoległego.

IEEE-1284 korzysta z poziomów logicznych **TTL**, a długość kabla nie powinna przekraczać 5m. Gniazdo przyłączeniowe może mieć 25 pinów (DB-25) lub 36 pinów (Centronics).

12. Interfejs RS-232C

Standard przesyłania informacji za pośrednictwem **asynchronicznego** łącza **szeregowego**. Dane organizowane są w ramki **SDU** składające się z:

- a) znacznika początku ramki (bit startu - 0)
- b) 5-8 bitów danych
- c) opcjonalnego bitu kontroli parzystości
- d) znacznika końca ramki (1-2 bity stopu - 1)

Oba urządzenia muszą mieć ustawione takie same parametry transmisji (częstotliwość taktowania i format ramki). Odbiornik próbuje stan linii odbiorczej z częstotliwością wielokrotnie wyższą od częstotliwości przesyłu danych, aby zsynchronizować się z nadajnikiem. Przekształcaniem danych z równoległej na szeregową i odwrotnie zajmuje się układ **UART**. Transmisja może odbywać się w trybie:

- a) jednokierunkowym (simpleks)
- b) dwukierunkowym ((pół)dupleks)

Standard definiuje protokół przesyłania danych między terminalem DTE a modemem DCE wykorzystujący linie sygnałowe:

- a) **dane wysyłane (TxD)**
- b) **dane odbierane (RxD)**
- c) **żądanie rozpoczęcia nadawania (RTS)**
- d) **zgłoszenie gotowości odbioru danych z terminala (CTS)**
- e) **gotowość modemu na przyjęcie danych (DSR)**
- f) **gotowość terminala (DTR)**
- g) **odbiór fali nośnej (DCD)**
- h) **wywołanie abonenta (RI)**

Para sygnałów RTS/CTS odpowiada za sterowaniem przepływem danych oraz kierunkiem transmisji w trybie półdupleksowym. Para sygnałów DTR/DSR odpowiada za utrzymanie nawiązanego połączenia.

RS-232C wykorzystuje napięcia -15 do +15 V ('0' - odpowiada +3 do +15V, '1' - odpowiada -3 do -15V). Typowe gniazdo ma 25 (DB-25) lub 9 pinów (DB-9). Max. odległość między urządzeniami < 200m. Obsługa łącza szeregowego może być realizowana przez:

- a) **przerwania zgłaszane przez UART**
- b) **okresowe testowanie przez procesor rejestru stanu transmisji UART**

13. Magistrala USB

USB - łączy szeregowo przeznaczone do podłączania urządzeń peryferyjnych. Cechy magistrali:

- a) prędkość: 1.5 Mb/s | 12 Mb/s | 480 Mb/s | 4.8 Gb/s
- b) obsługa przez kontroler do 127 urządzeń USB
- c) topologia gwiazdy (max. 7 poziomów) realizowana przez koncentratory USB
- d) max. długość kabla 5m
- e) możliwość zmiany konfiguracji urządzeń w trakcie pracy (Hot-Plugging)
- f) automatyczny przydział adresów do urządzeń (Auto-ID)
- g) możliwość zasilania podłączonych urządzeń kablem USB (do 500mA dla USB 2.0, 900mA dla USB 3.0)

Dane kodowane metodą **NRZI** przesyłane parą przewodów sygnałowych D+ i D- w sposób różnicowy (0-3.6V). Kabel połączeniowy to czteropłytowa skrętka (USB 3.0 - ośmiopłytowa) zakończona złączami standardowymi lub miniaturowymi. Wymiana informacji odbywa się w pakietach przez wirtualne kanały przesyłowe tworzone na żądanie przez kontroler USB. Każdy kanał może wykorzystywać jeden z czterech trybów transmisji:

- a) **asynchroniczny nie gwarantujący określone przepustowości (Bulk Transfer)** - obsługa podstawowych urządzeń zewnętrznych (np. drukarka, skaner)
- b) **synchroniczny o gwarantowanej szerokości pasma (Isochronous Transfer)** - obsługa urządzeń przesyłających duże porcje danych z określoną prędkością
- c) **natychmiastowy (Interrupt Transfer)** - obsługa urządzeń asynchronicznych przesyłających niewielkie porcje (np. myszka, klawiatura)
- d) **sterujący (Control Transfer)** - konfiguracja nowo podłączonych urządzeń

Przesyłane dane transmitowane są w kolejności od najmłodszego do najstarszego bitu. Każdy pakiet rozpoczyna się 8 lub 32-bitową **sekwencją synchronizującą (SYNC)** po której przekazywany jest 8-bitowy **identyfikator pakietu (PID)** określający m.in. rodzaj pakietu. Specyfikacja USB definiuje pakiety typu:

- a) **żeton (Token)** - określa adres urządzenia USB (7 bitów ADDR + 4 bity ENDP), kierunek przepływu danych oraz testowanie aktywności urządzenia
- b) **pakiet danych (Data)** - przesyłanie danych w blokach do 1024 bajtów zabezpieczonych 16-bitowym kodem CRC
- c) **pakiet potwierdzający (Handshake)** - przekazanie raportu o przesłanych pakietach lub informacja o aktywności urządzenia
- d) **pakiet specjalny (Special)** - sterowanie przepływem danych w magistrali, do której podłączone są urządzenia o różnej prędkości transmisji

14. Magistrala FireWire

Odmiana łączy **szeregowo** do podłączania urządzeń peryferyjnych (głównie audiowizualnych). Cechy FireWire:

- a) **transmisja asynchroniczna do 100, 200 lub 400 Mb/s**
- b) **negocjowana minimalna gwarantowana przepustowość pasma**
- c) **automatyczny przydział adresów do urządzeń (Auto-ID)**
- d) **Hot-Plugging**
- e) **automatyczne dopasowanie końców magistrali impedancją falową**
- f) **topologia drzewa złożona z kaskadowo połączonych łańcuchów węzłów spajanych pomostami**
- g) **max. głębokość pojedynczego drzewa - 16**
- h) **16-bitowe adresowanie węzłów (pomost pozwala na dołączenie do 63 węzłów)**
- i) **10-bitowe adresowanie pomostów**
- j) **max. długość kabla 4.5m**
- k) **możliwość zasilania urządzeń izolowanych (max. 1.5A)**

Sześciorzędowy kabel zakończony jednym z dwóch typów gniazd: 4 (1394a) lub 6 (1394) pinowym. Pierwsza para skrętek służy do przesyłania strumienia danych, druga do impulsów synchronizujących. Wymiana danych poprzez kanały wirtualne do transmisji **asynchronicznej** lub z **gwarantowaną szerokością pasma**. Najnowsza wersja IEEE-1394b:

- a) **transmisja: 400, 800, 1600, 3200 Mb/s**
- b) **kodowanie danych 8/10, co pozwala na korzystanie z obu par skrętek sygnałowych do transmisji danych równocześnie**
- c) **max. długość kabla 100m**
- d) **możliwość realizacji połączeń na bazie innych mediów (np. światłowód)**
- e) **nowe typy kabli połączeniowych i złącz**
- f) **optymalizacja protokołu transmisji**

15. Architektura procesora

Od lat 60-tych zasada **CISC** cechująca się tendencją rozszerzania listy rozkazów o nowe rozkazy, realizujące coraz bardziej złożone zadania. Lista rozkazów obejmowała nawet **kilkaset rozkazów** cechujących się zmienną długością i czasem wykonania od **jednego do kilkudziesięciu** taktów zegara. Na pojedynczy rozkaz składa się sekwencja **kilku do kilkunastu mikrooperacji**. Analiza kodu doprowadziła do wniosków:

- a) **za 80% czasu wykonywania odpowiada 20% instrukcji (80/20)**
- b) **dominująca grupa rozkazów to instrukcje przypisania, warunkowe, wywołanie i powroty z podprogramów**
- c) **większość argumentów to zmienne skalarne**

Powyższe obserwacje doprowadziły do powstania koncepcji o **zredukowanej liczbie rozkazów RISC**. Kluczowe elementy koncepcji RISC:

- a) **nieduży zestaw prostych rozkazów o stałej długości (uproszczenie konstrukcji jednostki sterującej i układu dekodera instrukcji)**
- b) **wielopoziomowa pamięć podręczna usprawniająca pobieranie większej liczby rozkazów**
- c) **duża liczba rejestrów wewnętrznych do optymalizacji odwołań do argumentów**
- d) **optymalizacja potoków wykonawczych z uwagi na duży udział rozkazów rozgałęzień warunkowych oraz wywołań/powrotów z podprogramów**

Procesory oparte na RISC cechują się wspólnymi własnościami:

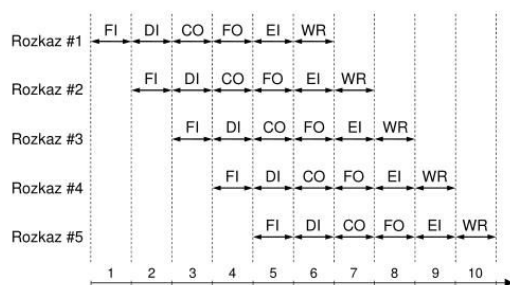
- a) **wykonanie większości rozkazów w jednym cyklu zegara**
- b) **prosty format i stała długość rozkazu**
- c) **jawne określanie rejestrów jako argumentów w kodzie rozkazu**
- d) **mała liczba prostych trybów adresowania**

Wiele obecnych procesorów łączy w sobie cechy obu architektur (np. Pentium 4)

16. Przetwarzanie potokowe

Przetwarzanie pojedynczego rozkazu można dekomponować na szereg faz, które realizowane są przez niezależne obwody procesora. **Przetwarzanie potokowe (Pipeline)** polega na zorganizowaniu procesu przetwarzania w sposób zbliżony **do linii montażowej w zakładzie produkcyjnym**. Potok zawiera rozkazy znajdujące się w różnym stadium przetwarzania, co prowadzi do zwiększenia szybkości ich przetwarzania. Podstawowym parametrem, który ma wpływ na efektywność jest **głębokość**. Przykładowy 6-etapowy potok pozwala na skrócenie czasu przetwarzania pięciu rozkazów z 30 do 10 cykli zegara:

- a) **pobieranie rozkazu (FI)**
- b) **dekodowanie rozkazu (DI)**
- c) **obliczanie argumentów (CO)**
- d) **pobieranie argumentów (FO)**
- e) **wykonanie rozkazu (EI)**
- f) **zapisanie wyniku (WR)**



Czynniki obniżające wydajność:

- a) **różny czas trwania poszczególnych etapów** (opóźnienia, wymuszenie stosowania dodatkowych buforów)
- b) **nie wszystkie etapy mogą być realizowane równolegle** (dostęp do zasobów)
- c) **przerwanie, błąd strony, naruszenie ochrony zaburza sekwencję przetwarzania**
- d) **rozgałęzienie warunkowe może prowadzić do unieważnienia rozkazów wewnątrz potoku**

Sposoby postępowania w przypadku rozgałęzień:

- a) **pobieranie rozkazu docelowego razem z rozkazem następującym po rozgałęzieniu**
- b) **jednoczesne przetwarzanie kilku ścieżek wykonawczych**
- c) **zapamiętywanie ostatnio pobranych rozkazów i sprawdzenie czy cel rozgałęzienia znajduje się wewnątrz tego bufora**
- d) **zmiana porządku wykonywanych rozkazów tak, aby rozgałęzienie występowało później**
- e) **przewidywanie rozgałęzień**

Najczęściej stosowanym rozwiązaniem jest **technika przewidywania biegu programu** w oparciu o bufor historii rozgałęzień **BTB**. Tablica BTB jest szybką pamięcią asocjacyjną zawierającą **do kilku tysięcy** rekordów przechowujących: **adres instrukcji rozgałęzienia, adres instrukcji docelowej, pewną liczbę bitów rejestrujących historię przetwarzania**. Efektywność przewidywania waha się od ok. 65 do 90%.

17. Przetwarzanie superskalarne

Przetwarzanie superskalarne polega na zastosowaniu kilku działających równolegle potoków przetwarzających dane skalarne, co pozwala na wykonywanie więcej niż jednego rozkazu w jednym cyklu zegara. Podstawową trudnością jest **wzajemna współzależność** rozkazów wykonywanych w poszczególnych potokach, m.in. rozgałęzienia, uzależnienia i konflikty w dostępie do zasobów. Przykładowo procesor Pentium wykorzystuje **dwa** potoki wykonawcze: U i V. Dzięki równoległemu wykonywaniu rozkazów, możliwe jest wykonywanie instrukcji całkowitoliczbowych w obu potokach w jednym cyklu rozkazowym. Potoki skojarzone z oboma kolejkami rozkazów **nie są równouprawnione**:

- a) **potok U pozwala na wykonywanie dowolnej instrukcji z zestawu rozkazów procesorów 80x86**
- b) **V na realizację jedynie instrukcji prostych, realizowanych całkowicie sprzętowo, bez udziału mikroprogramu wbudowanego w ROM procesora**

Rozkazy w potokach U i V przesuwają się **synchronicznie** przez poszczególne stopnie przetwarzania. Jeżeli rozkaz przejdzie określony stopień przetwarzania przed zakończeniem tej samej fazy w drugim potoku, to **oczekuje na drugi rozkaz**, tak aby

razem mogły przejść do kolejnej fazy. Procesor jest w stanie z każdego potoku odwoływać się do danych umieszczonych w wewnętrznej pamięci podręcznej. Uzyskanie zrównoleglenia wymaga "parowania rozkazów" tj. **łączenia ich w pary** w taki sposób, aby mogły być wykonywane równolegle przez oba potoki. Instrukcje podlegające łączeniu dzielimy na trzy grupy:

- a) **UV - instrukcje mogą być wykonywane w obu potokach**
- b) **PU - wykonywane tylko w potoku U połączone z odpowiednimi instrukcjami V**
- c) **PV - instrukcje połączone w pary z instrukcjami ze zbioru UV i PU będą wykonywane w potoku V**

Próba wykonania pewnych par instrukcji w sposób równoległy może prowadzić do pojawienia się konfliktów lub uzależnień:

- a) **Read After Read** (add bx, ax | mov cx, ax)
- b) **Read After Write** (mov ax, bx | mov [mem], ax)
- c) **Write After Read** (mov bx, ax | add ax, cx)
- d) **Write After Write** (mov ax, [mem] | mov ax, bx)

Uzależnienie **RAR** jest wynikiem konfliktu podczas próby jednoczesnego odczytu rejestru lecz można je stosunkowo łatwo usunąć poprzez **zdublowanie portu odczytu** rejestru.

Wynikiem **RAW** jest wystąpienie opóźnienia związanego z oczekiwaniem na wynik wcześniejszej operacji.

WAR i WAW wynikają z konfliktu podczas próby jednoczesnego zapisu do rejestru i można im zapobiec stosując **technikę przemianowywania rejestrów**.

Technika ta polega na wykorzystaniu puli dodatkowych rejestrów i ich dynamicznym mapowaniu do rejestrów dostępnych programowo. Po zdekodowaniu rozkazu następuje przydzielenie z puli rejestru, na którym będzie wykonywana operacja i zaktualizowanie tablicy zamienników **RAT**. Rejestr dostępny programowo może być mapowany na kilka rejestrów z puli, co pozwala na zrównoleglenie wykonywania rozkazów.

18. Dynamiczne wykonywanie instrukcji

Technika pozwalająca na wykonywanie wejściowej sekwencji instrukcji w dowolnej kolejności **w celu uniknięcia konfliktów** w dostępie do zasobów, oparta na:

- a) **zaawansowanej predykcji rozgałęzień**
- b) **dynamicznej analizie przepływu danych**
- c) **spekulatywnym wykonywaniu rozkazów**

Dynamiczna analiza w czasie rzeczywistym analizuje sekwencje instrukcji w **celu wykrycia i określenia zależności** pozwalających na ich optymalne uszeregowanie w jednostkach wykonawczych. **Spekulatywne wykonywanie instrukcji** polega na wykonywaniu instrukcji znajdujących się za jeszcze nierozstrzygniętą instrukcją warunkową przy czym wyniki są zapamiętywane w specjalnym bloku rejestrów tymczasowych. Typowa sekwencja dynamicznego wykonania instrukcji:

- a) **utworzenie przez układ MIS sekwencji mikrooperacji do wykonania**
- b) **uzupełnienie sekwencji informacją o ich oryginalnej kolejności**
- c) **przydzielenie do mikrooperacji rejestrów tymczasowych**
- d) **pobranie mikrooperacji do wykonania przez układ przydziału zasobów RS w takiej kolejności, która zapewnia optymalność**
- e) **odtworzenie przez układ RU właściwej kolejności zrealizowanych mikrooperacji i przepisanie wyników z rejestrów tymczasowych do rejestrów dostępnych programowo**

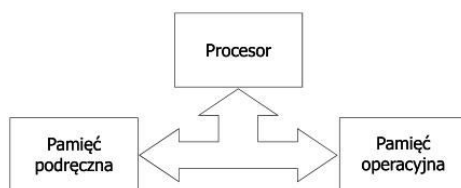
19. Pamięć podręczna

Celem jest **zwiększenie wydajności** podsystemu pamięci wynikające z zasady lokalności, w myśl której:

- a) **większość kodu przetwarzana jest w sposób sekwencyjny** (ciągły blok pamięci)
- b) **znaczna część obliczeń dotyczy przetwarzania zmiennych prostych, tablic i rekordów, położonych w pamięci blisko siebie**

Bardzo krótki czas dostępu, co pozwala na efektywne buforowanie wolniejszej pamięci RAM. System wielopoziomowy, złożony najczęściej z dwóch lub trzech kaskadowo połączonych układów pamięci statycznych o coraz większym czasie dostępu i większe pojemności. **L1** ma zwykle do kilkudziesięciu KB i służy do rozdzielonego buforowania danych i instrukcji. Rodzaje architektur systemów z pamięcią podręczną:

- **konwencjonalna (Look-Aside)**
- **szeregową (Look-Through)**
- **dwuszybową (Backside)**



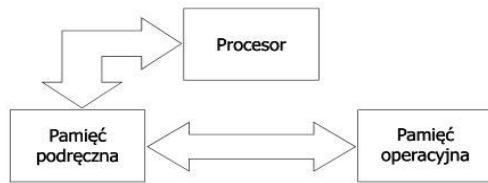
Look-Aside dołączona do systemu przez magistralę systemową i pracuje z wyznaczoną przez nią częstotliwością. Kontroler pamięci podręcznej monitoruje wszystkie odwołania procesora do pamięci, aby sprawdzić, czy żądana dana znajduje się w pamięci podręcznej. W przypadku **trafienia** procesor przerywa cykl magistrali i informuje podsystem sterujący pamięcią operacyjną aby zignorował żądanie. W przypadku **chybienia** cykle magistrali wykonują się normalnie.

Zalety Look-Aside:

- **niski koszt** (prosta konstrukcja)
- **polepszony czas dostępu do danych, które nie znajdują się w cache'u**

Wadą jest brak izolowania magistrali procesora od systemowej co prowadzi do:

- **niemożności zredukowania stopnia wykorzystania magistrali systemowej**
- **braku możliwości współbieżnego wykonywania operacji**



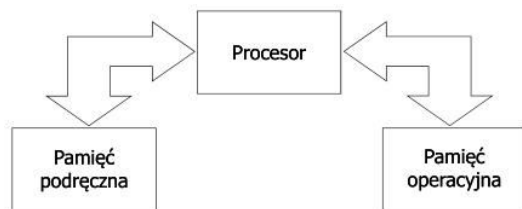
Po zainicjowaniu cyklu odczytu pamięć **Look-Through** przeszukuje swój katalog w celu rozstrzygnięcia czy zawiera kopię żądanej danej. W przypadku **trafienia** dana wysyłana jest do procesora z zerowymi cyklami oczekiwania magistrali, a magistrala systemowa pozostaje dostępna dla innych urządzeń. W przypadku **chybienia** następuje odczyt z pamięci operacyjnej (przez magistralę systemową).

Zalety:

- **zredukowanie stopnia wykorzystania magistrali systemowej**
- **możliwość współbieżnego wykonywania operacji na dwóch magistralach znajdujących się w jednym systemie**
- **wykonanie operacji zapisu w zerowych cyklach oczekiwania magistrali**
- **możliwość taktowania magistrali lokalnej inną częstotliwością**

Wady:

- **wyższy koszt** (większa złożoność układu)
- **wolniejszy odczyt danych, które nie znajdują się w cache'u**



Backside podłączona jest do procesora dedykowaną magistralą **BSB** która jest całkowicie niezależna od magistrali pamięciowej **FSB**. Pozwala to na swobodny wybór częstotliwości pracy cache. Obie magistrale mogą działać współbieżnie. Może być wkomponowana w strukturę procesora lub w pobliżu. Jeżeli dane w RAM różnią się odpowiadającym im danych w cache mamy do czynienia z utratą spójności cache. Przyczyną takiej sytuacji może być **trafienie** zapisu w pamięć podręczną lub zapisanie danych do RAMu przez urządzenie zewnętrzne.

Strategie zapewniające spójność pamięci:

- **zapis jednoczesny (Write-Through)**
- **zapis buforowany (Buffered Write-Through)**
- **zapis opóźniony (Write-Back)**

Write-Through - zapis do cache powoduje wykonanie cyklu zapisu do RAM. Implementacja tej metody jest efektywna i prosta, jednak mało wydajna.

Buffered Write-Through - kontroler pamięci podręcznej zapamiętuje realizowane operacje zapisu w specjalnym buforze, co umożliwia ich późniejsze zrealizowanie w RAM bez wpływu na pracę procesora. Brak cykli oczekiwania magistrali. Inne urządzenie wymagające dostępu do magistrali nie może przeprowadzać żadnych operacji do chwili zakończenia zapisu.

Write-Back - aktualizuje RAM tylko, gdy jest to konieczne; w przypadku gdy:

- **urządzenie zewnętrzne inicjuje operację odczytu/zapisu tych słów pamięci, zawierające przeterminowane dane**
- **słowo pamięci podręcznej, które zawiera zmodyfikowane dane, ma zostać nadpisane danymi z RAM**

W momencie zapisu/odczytu do RAMu, podsystem cache monitoruje magistralę systemową, aby sprawdzić czy są odwołania do słów oznaczonych jako zmodyfikowane. Zachowanie spójności oparte jest na protokole **MESI**, który określa stan każdego wiersza pamięci podręcznej danych jako:

- **zmodyfikowany**
- **wyłączony**
- **współdzielony**
- **nieważny**

Cache zorganizowany jest w wiersze, których szerokość = szerokości magistrali danych.

Dzielona na strony o rozmiarze równym liczbie wierszy. Informacja, które wiersze pamięci operacyjnej są odwzorowane w cache znajduje się w katalogu pamięci podręcznej **TAG-RAM**. Strategie odwzorowywania:

- **bezpośrednie**
- **skojarzeniowe**
- **sekcyjno-skojarzeniowe**

Bezpośrednie polega na sposobie odwzorowywania RAM na cache tak, że n-ty wiersz każdej strony RAMu może zostać odwzorowany tylko w n-tym wierszu cache.

Zaletą jest duża szybkość wyszukiwania informacji oraz prosta konstrukcja układu.

Ponieważ każdy wiersz RAM ma stałą lokalizację w cache, następuje do ciągłej wymiany zawartości cache, a tym samym do obniżenia wydajności systemu. (strona + dane)

Skojarzeniowe pozwala na ładowanie każdego wiersza RAM do dowolnego wiersza cache, co eliminuje największą wadę bezpośredniego. Nie występuje tu podział RAM na strony, wszelkie operacje wykonywane są na wierszach identyfikowanych przez adresy. Wadą jest konieczność przeszukania całego katalogu TAG-RAM aby zbadać, czy żądany wiersz RAM został odwzorowany w cache. (adres + dane)

Sekcyjno-skojarzeniowe charakteryzuje się podziałem cache na 2, 4, 8 sekcji o rozmiarze wielkości strony RAM. Ponieważ RAM odwzorowywany jest na każdą sekcję w sposób bezpośredni, wyszukiwanie odpowiedniej informacji w cache wymaga przeprowadzenia co najwyżej 8 porównań. (strona + dane + nr sekcji)

Wspólną cechą (2 i 3) jest konieczność wyboru tego wiersza cache, który zostanie zastąpiony przez nowo ładowany wiersz. W tym celu kontroler cache implementuje jeden z algorytmów:

- a) **FIFO** - wybiera wiersz najdłużej pozostający w cache
- b) **LFU** - wybiera najrzadziej używany wiersz cache
- c) **LRU** - wybiera najmniej ostatnio używany wiersz cache

20. Przegląd architektury IA-32

Rozwinięcie rodziny 80x86. Cechuje się wsteczną zgodnością programową. Typowi przedstawiciele: Pentium 4. Intel Pentium 4 HT:

- liczba tranzystorów 55 mln
- częstotliwość 3.06 GHz
- cache: 12 uops (ETC), 8 KB (L1), 512 KB (L2)
- technologia Hyper Threading

Podstawowe środowisko wykonawcze procesorów IA-32 tworzy:

a) rejestry podstawowe:

- osiem 32-bitowych rejestrów ogólnego przeznaczenia (E(A-D)X, EBP, ESI, EDI, ESP), których młodsze 16-bitów jest odwzorowanych na rejestry 16-bitowe procesorów 8086 i 80286 ((A-D)X, BP, SI, DI, SP), dodatkowo (A-D)X są mapowane na pary 8-bitowe AH, AL itd.

- sześć 16-bitowych rejestrów segmentowych (CS, DS, ES, FS, GS, SS)

- jeden 32-bitowy rejestr statusowy (EFLAGS)

- jeden 32-bitowy rejestr wskaźnika rozkazów (EIP)

b) rejestry jednostki zmienno-przecinkowej:

- osiem 80-bitowych rejestrów danych w postaci stosu ST0-ST7

- jeden 16-bitowy rejestr sterujący

- jeden 16-bitowy rejestr stanu

- jeden 16-bitowy rejestr stanu słowa wyróżników

- jeden 48-bitowy rejestr wskaźnika rozkazu

- jeden 48-bitowy rejestr wskaźnika argumentu

c) osiem 64-bitowych rejestrów technologii MMX, współdzielonych z rejestrami danych FPU

d) rejestry technologii SSE i SSE2:

- osiem 128-bitowych rejestrów danych XMM0 - XMM7

- jeden 32-bitowy rejestr kontrolno-statusowy MXCSR

e) rejestry pomocnicze, m.in.:

- rejestry sterujące pracą procesora CR0-CR4

- podsystem zarządzania pamięcią (GDTR, LDTR, IDTR, TR)

- uruchomieniowe (DR0-DR7)

- zaawansowanego kontrolera przerwań APIC (ISR, TMR, IRR, ESR)

- wspomagające testowanie procesora (TR3-TR7)

Procesor może pracować w trybie:

a) rzeczywistym

b) chronionym

c) zarządzania SMM

W trybie **rzeczywistym** odpowiada funkcjonalnie procesorowi 8086. 20-bitowy adres fizyczny tworzony na podstawie 16-bitowego adresu segmentu i 16-bitowego przesunięcia w segmencie ($Addr = 16 * Seg + Off$).

Tryb chroniony jest podstawowym trybem, zapewniającym ochronę niezależnych zadań wykonywanych współbieżnie w środowisku wielozadaniowym. Możliwe jest udostępnianie dla każdego z wykonywanych zadań niezależnego środowiska procesora 8086. Tryb wirtualny procesora 8086 pozwala na uruchamianie programów pisanych dla trybu rzeczywistego w środowisku wielozadaniowym, chronionym.

Tryb zarządzania SMM pozwala na realizację funkcji systemowych wysokiego poziomu, jak zarządzanie energią, czy ochrona zasobów przed nieuprawnionym dostępem. Procesor kontrolowany jest wyłącznie przez oprogramowanie systemowe zawarte w pamięci stałej w sposób transparentny dla innych programów. Zastosowaniem jest implementacja systemu zarządzania zasilaniem, który może wprowadzać procesor i wybrane urządzenia w stan uśpienia w okresie bezczynności.

20.1 IA-32: Zarządzanie pamięcią

Procesory mogą adresować max do 64 GB pamięci fizycznej. Mechanizm zarządzania pamięcią wirtualną odwzorowuje przestrzeń adresową wykonywanego programu w przestrzeń pamięci fizycznej wykorzystując model pamięci:

a) płaski

b) segmentowy

c) trybu rzeczywistego procesora 8086

W modelu **płaskim** pamięć widoczna dla programu jest ciągła i liniowa (kod, dane, stos rezydują w jednej przestrzeni adresowej). Max. rozmiar udostępnianej przestrzeni dla programu wynosi 4 GB.

W modelu **segmentowym** pamięć widoczna dla programu jest jako grupa niezależnych obszarów pamięci zwanych segmentami (kod, dane, stos w niezależnych segmentach). Program może zaadresować max. 16383 segmenty różnego typu i rozmiaru (nie większe niż 4 GB). Adres logiczny składa się z dwóch elementów: 16-bitowego selektora segmentu i 32-bitowego przesunięcia w segmencie. **Selektor** stanowi indeks do globalnej (GDT) lub lokalnej (LDT) tablicy deskryptorów opisujących poszczególne segmenty. **Deskryptor** jest strukturą przechowującą jego typ, granularność, rozmiar, lokalizację w pamięci oraz prawa dostępu. Adresacja wewnątrz segmentu jest liniowa. Selektory segmentu przechowywane są w rejestrach segmentowych. Przesunięcia wewnątrz segmentów przechowywane są najczęściej w rejestrach ogólnego przeznaczenia. Zaletą modelu segmentowego jest poprawienie **niezawodności systemu operacyjnego** dzięki:

- **odseparowaniu programów wykonywanych współbieżnie w środowisku wielozadaniowym**
- **rozdzieleniu logicznie niezależnych "fragmentów" programu**

Model **trybu rzeczywistego** odwzorowuje mechanizm adresowania pamięci procesora 8086. Wykorzystuje specjalną implementację modelu segmentowego polegającą na podziale pamięci na nakładające się i przesunięte o 16 B segmenty o rozmiarze 64 KB.

Max. rozmiar udostępnianej programowi przestrzeni wynosi 1 MB. Fizyczny adres wyznaczany jest na podstawie 16-bitowego adresu segmentu i 16-bitowego przesunięcia w segmencie.

Od procesora 80386 odwzorowanie adresu logicznego na adres fizyczny może być realizowane z udziałem **stronicowania pamięci**. Jednostka stronicowania dzieli liniową przestrzeń adresową na strony o stałym rozmiarze, które są odwzorowywane w pamięci wirtualnej złożonej z pamięci operacyjnej i dyskowej. Max. rozmiar adresowalnej w ten sposób pamięci wirtualnej wynosi 64 TB. Jeżeli strona **nie znajduje się** w pamięci operacyjnej to jej adres zostaje zapisany w rejestrze **CR2** i generowany jest wyjątek błędu strony. Wczytanie brakującej strony z dysku spoczywa na OS. W celu zwiększenia wydajności stronicowania, procesor wyposażony jest w pamięci podręczne **TLB** przechowującą zawartość ostatnio używanych tablic stron.

20.2 IA-32: Technologia MMX

Ta technologia ma na celu **zwiększenie efektywności wykonywania programów** (głównie multimedialnych), które cechują się powtarzaniem takich samych, prostych operacji na długich ciągach krótkich danych. Idea ta stanowi krok w stronę **przetwarzania równoległego SIMD** - zamiast kolejnego przetwarzania pojedynczych jednostek informacji, można te same dane przetworzyć szybciej, pobierając i wykonując rozkazy na kilku porcjach informacji naraz. Taki procesor jest wyposażony w osiem 64-bitowych rejestrów **MM0-MM7** pozwalających na przechowywanie nowych typów danych:

- **Packet Bytes (8 x 8 bitów)**
- **Packed Words (4 x 16 bitów)**
- **Packet Doublewords (2 x 32 bity)**
- **Quadword (1 x 64 bity)**

Jednoczesne wykonywanie operacji na danych spakowanych umożliwia zestaw dodatkowych 57 rozkazów wykorzystujących tzw. **arytmetykę nasycenia**. W tej arytmetyce nie występuje sygnalizacja przekroczenia zakresu lecz zastąpienie wyniku wartością maksymalną lub minimalną.

20.3 IA-32: Technologia SSE (SSE2, ...)

Technologia ta rozszerza ideę jednoczesnego przetwarzania grupy danych pojedynczym rozkazem na dane w formacie zmiennoprzecinkowym. Procesor implementujący tą technologię jest wyposażony w osiem 128-bitowych rejestrów **XMM0 - XMM7** oraz rejestr kontrolno-sterujący **MXCSR**. Zaletą jest uniezależnienie się od rejestrów zmiennoprzecinkowych FPU. Wadą jest brak zgodności z wcześniejszymi typami procesorów. Lista rozkazów zawiera 72 rozkazy. Typowym zastosowaniem jest obróbka grafiki 3D oraz pełnoekranowa kompresja MPEG-2 w czasie rzeczywistym.

21. Magistrale systemowe

Magistrala jest zbiorem pewnych linii sygnałowych umożliwiającym takie połączenie współpracujących elementów, aby każdy z nich mógł komunikować się ze wszystkimi pozostałymi. Podstawowe cechy:

- rodzaj** (specjalistyczna, multipleksowana)
- arbitraż** (centralny, rozproszony)
- koordynacja czasowa** ((a)synchroniczna)
- szerokość magistrali** (adres, dane)
- rodzaj transferu danych**

Magistrala systemowa jest komponentem **zapewniającym komunikację** między podstawowymi podzespołami komputera. W stosowanych obecnie magistralach można wyróżnić trzy grupy linii sygnałowych:

- **linie adresowe**
- **linie danych**
- **linie sterujące**

W zależności od przeznaczenia mamy magistrale:

- **pamięciowe**
- **międzyukładowe (wewnętrzne)**
- **rozszerzające (zewnętrzne)**
- **lokalne**

Zadaniem **magistrali pamięciowej** jest przesył informacji między prockiem a RAM/cache.

Magistrala międzyukładowa służy do wymiany informacji między poszczególnymi układami otoczenia procesora (chipsets)

Magistrala rozszerzająca służy do wymiany informacji między urządzeniami IO a resztą komputera. Najczęściej konstrukcja takiej magistrali musi spełniać warunki zgodności z wcześniejszymi rozwiązaniami i zapewnić poprawną współpracę z szerokim zakresem urządzeń peryferyjnych.

Magistrala lokalna jest odmianą magistrali rozszerzającej przeznaczoną do obsługi urządzeń wymagających bardzo szybkiej transmisji danych. Cechą charakterystyczną jest bezpośrednie korzystanie z sygnałów sterujących procesora oraz praca z prędkością wyznaczoną przez zegar systemowy.

Standardy magistral rozszerzających:

- **ISA**

- **MCA**

- **EISA**

- **VESA**

- **PCI:**

@ 32/64-bitowa szyna danych

@ przepustowość 133/266/533 MB/s (taktowanie 33/66 MHz)

@ odczyt i zapis w trybie pakietowym (burst)

@ obsługa do 32 urządzeń

@ możliwość współpracy do 256 magistral PCI w jednym komputerze

@ możliwość obsługi magistrali ISA

@ niezależność od typu procesora

@ wbudowany mechanizm autokonfiguracji

@ urządzenia dołączane mogą należeć do kategorii inicjujących transmisję oraz urządzeń podporządkowanych

@ umożliwia współdzielenie każdej z czterech linii zgłoszenia przerwania przez kilka urządzeń, lecz nie określa ich podłączenia do kontrolera przerw systemu

@ napięcie 3.3 / 5 V

- **AGP (rozszerzenie PCI 2.1)**

- **PCI Express:**

@ połączeniowa typu punkt-punkt

@ wykorzystuje 1-32 linii sygnałowych determinujących jej przepustowość oraz rozmiar złącza połączeniowego

@ przepustowość w zależności od wariantu wynosi 250 MB/s do 16 GB/s

@ maksymalny pobór mocy: 75 W (1.0) 150 W (2.0) 300 W (3.0)