

## Pytania z PUTWIKI

1. Cykl rozkazowy procesora
2. Idea DMA
3. Odczyt w trybie FPM
4. RAID0
5. Cykl maszynowy
6. Przerwania
7. Zapis w trybie Burst
8. RAID1
9. Omów i porównaj obsługę urządzeń zewnętrznych wykorzystującą przerwania (interrupt) oraz odpytywanie (pooling).
10. Przedstaw tryby pracy łącza równoległego interfejsu IEEE-1284.
11. Na czym polega odwzorowanie sekcyjno-skojarzeniowe pamięci podręcznej.
12. Omów organizację wewnętrzną pamięci operacyjnej typu SDRAM.
13. Przedstaw ideę potokowego przetwarzania rozkazów.
14. Omów i porównaj transmisje danych metodą PIO i DMA.
15. Opisz na czym polega asynchroniczna transmisja (interfejs RS-232C)
16. Omów odwzorowanie bezpośrednie pamięci podręcznej.
17. Omów koncepcje RISC architektury procesora.
18. Przedstaw idee superskalarnego przetwarzania rozkazów.

## 1. Cykl rozkazowy procesora

W cyklu rozkazowym procesora możemy wyróżnić następujące fazy:

- Faza pobrania rozkazu, w której następuje pobranie rozkazu z podręcznej kolejki rozkazów i jeżeli zachodzi taka potrzeba, uruchomienie procedury uzupełniania kolejki
- Faza dekodowania rozkazu, w której określone są operacje do wykonania oraz obliczane adresy efektywne argumentów (jeżeli występują)
- Faza wykonania, w której pobierane są z pamięci argumenty rozkazu oraz wykonywane operacje ustalone w fazie dekodowania
- Faza zapisania wyników, w której następuje zapisanie wyniku w miejscu określonym w kodzie rozkazu oraz ustawienie znaczników rejestru statusowego

## 2. Idea DMA

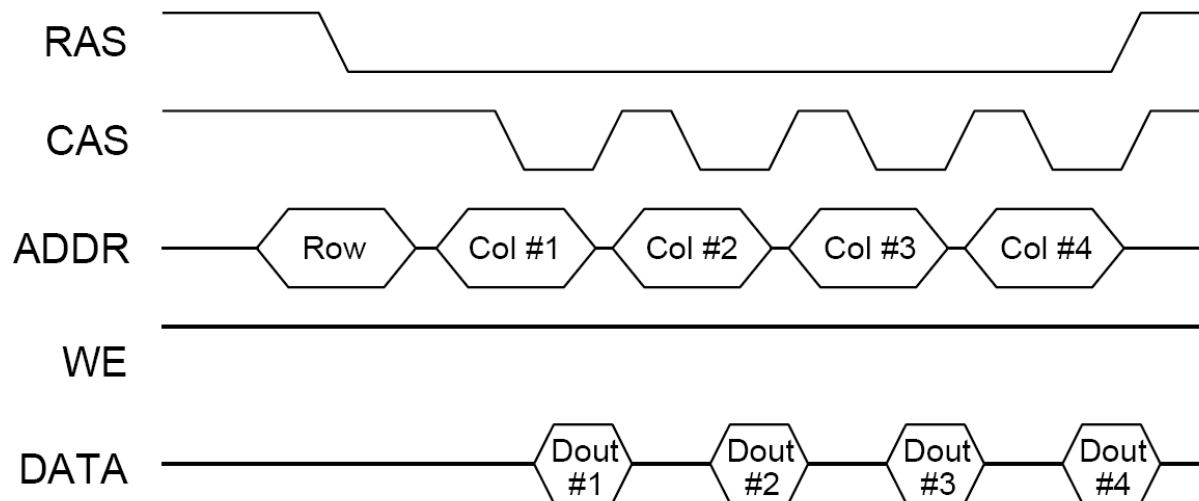
DMA (ang. Direct Memory Access - bezpośredni dostęp do pamięci) - technika, w której inne układy (np. kontroler dysku twardego, karta dźwiękowa, itd.) mogą korzystać z pamięci operacyjnej RAM lub (czasami) portów we-wy pomijając przy tym procesor główny - CPU. Wymaga to współpracy ze strony procesora, który musi zaprogramować kontroler DMA do wykonania odpowiedniego transferu, a następnie na czas przesyłania danych zwolnić magistralę systemową (przejsć w stan wysokiej impedancji). Sam transfer jest już zadaniem wyłącznie kontrolera DMA. Realizacja cykli DMA może przez urządzenie być zrzucona na specjalny układ (np. w komputerach PC) lub być realizowana samodzielnie przez urządzenie.

DMA ma za zadanie odciążyć procesor główny od samego przesyłania danych z miejsca na miejsce (np. z urządzenia wejściowego do pamięci), procesor może w tym czasie zająć się 'produktywnym' działaniem, wykonując kod programu pobrany uprzednio z pamięci RAM do pamięci cache operujący na danych w tejże pamięci zgromadzonych. Specjalizowane układy wspomagające DMA (np. te spotykane w PC) potrafią też kopiować obszary pamięci dużo szybciej niż uczyniłby to programowo procesor główny.

Istnieje specjalna procedura DMA, "Scatter-Gather" (dosł.: rozrzuć-zbierz), pozwalająca przenosić dane do wielu obszarów pamięci w pojedynczym transferze. Pod względem skutków jest to równoważne połączeniu łańcuchowemu kilku transferów, jednak jest wyraźnie szybsze.

### 3. Odczyt w trybie FPM

Najbardziej popularny rodzaj dynamicznej pamięci operacyjnej komputera. Tryb stronicowania DRAM uzyskuje dostęp do rzędu pamięci RAM bez nieustannego wyszczególniania rzędu. Sygnał RAS jest aktywny, podczas gdy sygnał CAS zmienia do odczytu sekwencję kolejnych komórek. Zmniejsza to czas dostępu i obniża wymagania co do zasilania.



## 4. RAID0 (stripping)

- dane dzielone są na tzw. paski, które rozmieszczane są cyklicznie na kolejnych dyskach macierzy
- brak redundancji danych
- duża szybkość odczytu i zapisu danych
- pojemność macierzy jest równa sumie pojemności wszystkich dysków wchodzących w jej skład

## 5. Cykl maszynowy

Cykl maszynowy to cykl, podczas którego następuje wymiana danych między procesorem a pamięcią lub układem wejścia wyjścia (odczyt albo zapis).

W każdym cyklu maszynowym następuje wystanie:

- adresu na magistralę adresową,
- danych na magistralę danych,
- sygnałów sterujących, informujących o rodzaju cyklu, na magistralę sterującą.

Układy pamięci lub wejścia - wyjścia powinny w tym czasie wykonać odpowiednie czynności - zapisać dane lub wysłać je na magistralę danych.

W zależności od rodzaju przesłania rozróżnia się cykl maszynowy:

- pobrania kodu operacji,
- odczytu i zapisu pamięci,
- odczytu i zapisu wejścia-wyjścia,
- przyjęcia, przerwania itd.

Jeden cykl maszynowy wykonywany jest w czasie jednego lub kilku (w zależności od procesora i rodzaju cyklu) cykli zegarowych.

## 6. Przerwania

Przerwanie (ang. interrupt) lub żądanie przerwania (IRQ - Interrupt ReQuest) - sygnał powodujący zmianę przepływu sterowania, niezależnie od aktualnie wykonywanego programu. Pojawienie się przerwania powoduje wstrzymanie aktualnie wykonywanego programu i wykonanie przez procesor kodu procedury obsługi przerwania (ang. interrupt handler).

Przerwania dzielą się na dwie grupy:

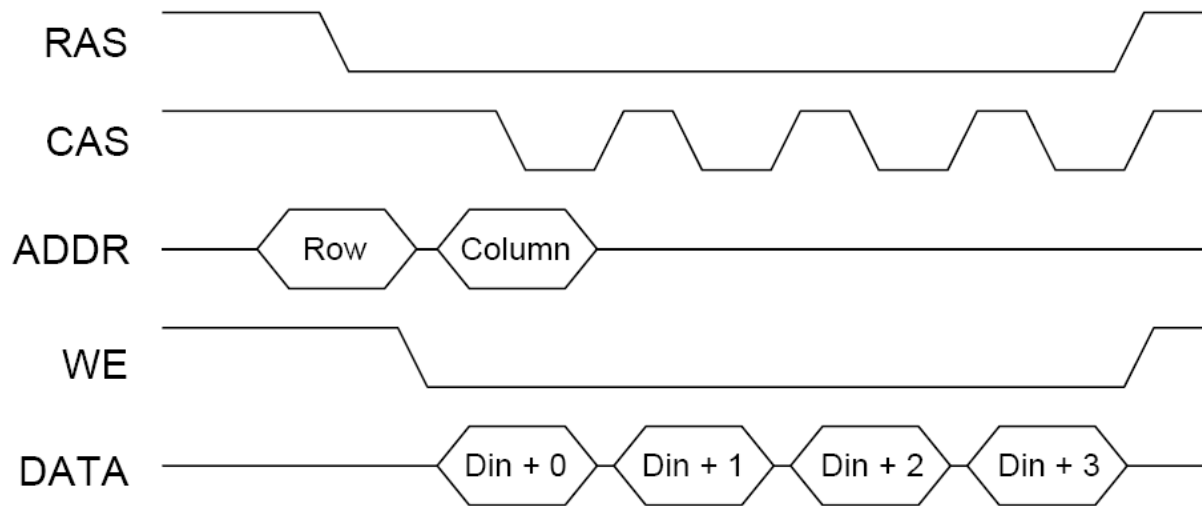
### 1. Sprzętowe:

- a. Zewnętrzne - sygnał przerwania pochodzi z zewnętrznego układu obsługującego przerwania sprzętowe; przerwania te służą do komunikacji z urządzeniami zewnętrznymi, np. z klawiaturą, napędami dysków itp.
- b. Wewnętrzne, nazywane wyjątkami (ang. exceptions) - zgłaszane przez procesor dla sygnalizowania sytuacji wyjątkowych (np. dzielenie przez zero); dzielą się na trzy grupy:
  - c. faults (niepowodzenie) - sytuacje, w których aktualnie wykonywana instrukcja powoduje błąd; gdy procesor powraca do wykonywania przerwanego kodu wykonuje następną, po tej która wywołała wyjątek, instrukcję;
  - d. traps (pułapki) - sytuacja, która nie jest błędem, jej wystąpienie ma na celu wykonanie określonego kodu; wykorzystywane przede wszystkim w debuggerach; gdy procesor powraca do wykonywania przerwanego kodu, wykonuje tę samą instrukcję która wywołała wyjątek;
  - e. aborts - błędy, których nie można naprawić.

### 2. Programowe - z kodu programu wywoływana jest procedura obsługi przerwania; najczęściej wykorzystywane do komunikacji z systemem operacyjnym, który w procedurze obsługi przerwania (np. w DOS 21h, 2fh, Windows 2fh, Linux x86 przerwanie 80h) umieszcza kod wywołujący odpowiednie funkcje systemowe w zależności od zawartości rejestrów ustawionych przez program wywołujący, lub oprogramowaniem wbudowanym jak procedury BIOS lub firmware.

## 7. Zapis w trybie Burst

Tryb pakietowy zapisu pamięci DRAM:





## 8. RAID1

- dane są paskowane i następnie odwzorowywane na dwóch odrębnych dyskach fizycznych
- średnia szybkość odczytu, mała szybkość zapisu danych
- pojemność macierzy jest równa połowie pojemności wszystkich dysków wchodzących w jej skład

## 9. Omów i porównaj obsługę urządzeń zewnętrznych wykorzystującą przerwania (interrupt) oraz odpytywanie (pooling).

*W trybie odpytywania tzw pooling-u procesor cyklicznie sprawdza stan urządzenia oczekując na spełnienie warunków koniecznych do realizacji transmisji. W przypadku tym pewna część czasu procesora tracona jest na sprawdzanie stanu urządzenia. Zdecydowanie lepszym rozwiązaniem okazują się przerwania. W tym przypadku procesor może wykonywać inne zadania i nie ma konieczności sprawdzania stanu innych urządzeń. Urządzenie może poinformować procesor o zmianie swojego trybu zgłaszając przerwanie (interrupt), co powoduje zawieszenie wykonywania bieżącego programu przez procesor, wykonanie procedury obsługi przerwania a następnie powrót do wykonywania poprzedniego programu. Zgłoszenie przerwania spowodowane może m. in. być zajściem zdarzenia zewnętrznego lub zakończeniem zadania zleconego urządzeniu przez procesor. Takie rozwiązanie zdecydowanie zwiększa wydajność procesora.*

## 10. Przedstaw tryby pracy łącza równoległego interfejsu IEEE-1284.

Specyfikacja standardu IEEE-1284 definiuje pięć trybów pracy łącza równoległego:

- **Tryb kompatybilny** - Compatibility Mode \*  
Zwany jako standardowy lub Centronics. Jednokierunkowa transmisja danych z prędkością do 200 kB/s
- **Tryb półbajtowy** - Nibble Mode \*  
dwukierunkowa transmisja danych. Do odbierania danych wykorzystuje się linie BSY, ACK, PERR i SEL połączone z magistralą danych przez multiplekser sterowany sygnałem STR
- **Tryb bajtowy PS/2** - Byte Mode \*  
dwukierunkowa transmisja danych oparta na wykorzystaniu dwukierunkowego portu danych
- **Tryb EPP** - Enhanced Parallel Port  
dwukierunkowa transmisja danych do 2 MB/s
- **Tryb ECP** - Enhanced Capabilities Port  
dwukierunkowa transmisja danych do 2 MB/s. Pozwala na przesyłanie danych z kontrolera w trybie DMA oraz korzystanie z:
  - rejestru buforowanego FIFO,
  - kompresji danych
  - definiowania i adresowania do 128 kanałów logicznych

Pierwsze trzy tryby pracy (\*) określa się także jako tryb **SPP** - Standard Parallel Port, który oprócz 8 bitowej magistrali danych wykorzystuje następujące linie sygnałowe:

- |        |                      |                         |
|--------|----------------------|-------------------------|
| 1. In  | ERR (error)          | błąd drukarki           |
| 2. Out | STR (strobe)         | sygnał strobuujący dane |
| 3. In  | ACK (Acknowledge)    | potwierdzenie           |
| 4. In  | BSY (Busy)           | zajętość                |
| 5. In  | PERR (Paper error)   | brak papieru            |
| 6. In  | SEL (Select)         | gotowość                |
| 7. Out | ALF (Auto Line Feed) | wysuw wiersza           |
| 8. Out | INI (Initialization) | inicjowanie drukarki    |
| 9. Out | DSL (Device Select)  | wybór drukarki          |

Dostęp do łącza równoległego pracującego w trybie **SPP** odbywa się za pośrednictwem 3 rejestrów:

1. danych (Data Register) - zapis odczyt danych z magistrali
2. stanu (Status Register) - odczyt stanu linii sygnałowych
3. sterującego (Control Register) - aktywowanie dwukierunkowej pracy portu danych oraz ustawianie stanu linii DSL, ALF, STR

Implementacja protokołów transmisji danych w **SPP** odbywa się programowo

Dostęp do łącza równoległego pracującego w trybie **EPP** odbywa się za pomocą rejestrów **SPP** rozszerzonych o

1. Data Register - dwukierunkowy rejestr danych
2. Address Register - dwukierunkowy rejestr adresowy

Dostęp do łącza równoległego pracującego w trybie **ECP** odbywa się za pośrednictwem rejestrów **SPP** rozszerzonych m.in. o:

1. Rejestr adresowy FIFO (ECP Address FIFO)
2. Rejestr danych FIFO (Data FIFO)
3. Rejestr sterujący ECR (Extended Control Register)

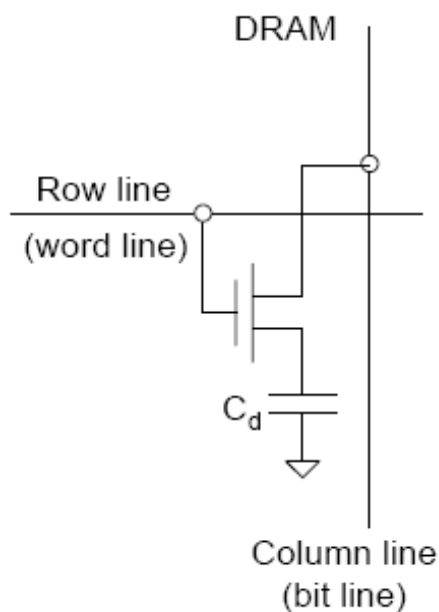
## 11. Na czym polega odwzorowanie sekcyjno-skojarzeniowe pamięci podręcznej.

Odwzorowanie sekcyjno-skojarzeniowe jest metodą, która łączy zalety odwzorowanie bezpośredniego i skojarzeniowego. Cechą charakterystyczną tego typu podejścia jest podział pamięci podręcznej na 2, 4 lub 8 sekcji o rozmiarze równym wielkości strony pamięci podręcznej.

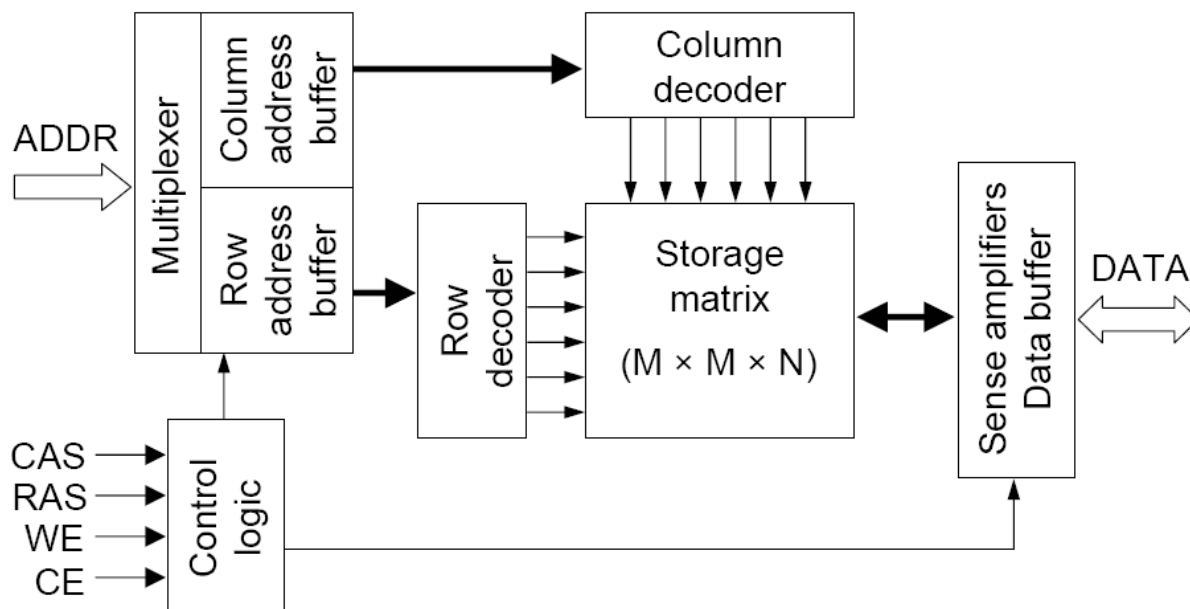
Ponieważ pamięć operacyjna odwzorowana jest na każdą sekcję w sposób bezpośredni, wyszukanie odpowiedniej informacji w pamięci podręcznej wymaga przeprowadzenia co najwyżej 8 porównań (jedno porównanie na sekcję).

## 12. Omów organizację wewnętrzną pamięci operacyjnej typu SDRAM.

Diagram komórki pamięci DRAM:



Organizacja wewnętrzna pamięci DRAM:



Komórki pamięci o dostępie swobodnym DRAM (Dynamic RAM) zbudowane są z układów przechowujących ładunek elektryczny i wymagają okresowego odświeżania ładunku w celu zachowania danych. Zbudowane są z kondensatora i tranzystora.

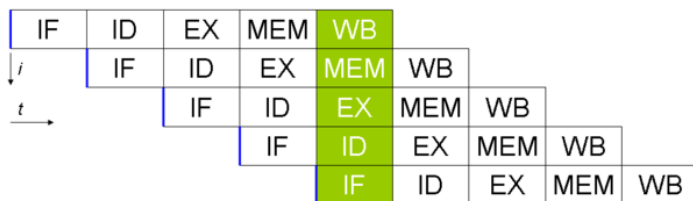
SDRAM (Synchronous DRAM) jest odmianą powyżej opisanej pamięci. Układ pamięci SDRAM zbudowany jest z niezależnych banków, które adresowane naprzemiennie przyspieszają pracę. W układ pamięci wbudowany jest licznik odświeżania i układy logiczne, które pozwalają na jego autonomiczne odświeżanie.

Cechą charakterystyczną SDRAM jest synchronizowanie wszystkich sygnałów sterujących przebiegiem zegarowym co ułatwia współpracę z magistralami systemowymi i pozwala na zwiększenie szybkości jej działania. Pamięć ta jest optymalizowana do realizacji przesyłania pakietowego.

### 13. Przedstaw ideę potokowego przetwarzania rozkazów.

Potokowość (ang. *pipelining*) – technika budowy procesorów polegająca na podziale logiki procesora odpowiedzialnej za proces wykonywania programu (instrukcji procesora) na specjalizowane grupy w taki sposób, aby każda z grup wykonywała część pracy związanej z wykonaniem rozkazu. Grupy te są połączone sekwencyjnie – potok (ang. *pipe*) – i wykonują pracę równocześnie, pobierając dane od poprzedniego elementu w sekwencji. W każdej z tych grup rozkaz jest na innym stadium wykonania. Można to porównać do taśmy produkcyjnej. W uproszczeniu, potok wykonania instrukcji procesora może wyglądać następująco: Uproszczony schemat potokowości. Części rozkazów oznaczone na zielono wykonywane są równocześnie. Uproszczony schemat potokowości. Części rozkazów oznaczone na zielono wykonywane są równocześnie.

- Pobranie instrukcji z pamięci - ang. *instruction fetch (IF)*
- Zdekodowanie instrukcji - ang. *instruction decode (ID)*
- Wykonanie instrukcji - ang. *execute (EX)*
- Dostęp do pamięci - ang. *memory access (MEM)*
- Zapisanie wyników działania instrukcji - ang. *store; write back (WB)*



W powyższym 5-stopniowym potoku, przejście przez wszystkie stopnie potoku (wykonanie jednej instrukcji) zabiera co najmniej 5 cykli zegarowych. Jednak ze względu na jednoczesną pracę wszystkich stopni potoku, jednocześnie wykonywanych jest 5 rozkazów procesora, każdy w innym stadium wykonania. Oznacza to, że taki procesor w każdym cyklu zegara rozpoczyna i kończy wykonanie jednej instrukcji i statystycznie wykonuje rozkaz w jednym cyklu zegara. Każdy ze stopni potoku wykonuje mniej pracy w porównaniu do pojedynczej logiki, dzięki czemu może wykonać ją szybciej - z większą częstotliwością - tak więc dodatkowe zwiększenie liczby stopni umożliwia osiągnięcie coraz wyższych częstotliwości pracy.

Podstawowym mankamentem techniki potoku są rozkazy skoku, powodujące w najgorszym wypadku potrzebę przeczyszczenia całego potoku i wycofania rozkazów, które następowały zaraz po instrukcji skoku i rozpoczęcie zapełniania potoku od początku od adresu, do którego następował skok. Taki rozkaz skoku może powodować ogromne opóźnienia w wykonywaniu programu - tym większe, im większa jest długość potoku. Dodatkowo szacuje się, że dla modelu programowego x86 taki skok występuje co kilkanaście rozkazów. Z tego powodu niektóre architektury programowe (np. SPARC) zakładały zawsze wykonanie jednego lub większej ilości rozkazów następujących po rozkazie skoku, tzw. skok opóźniony. Stosuje się także skomplikowane metody predykcji skoku lub metody programowania bez użycia skoków.

– Krótkie:

W cyklu rozkazowym procesora można wyróżnić cztery fazy:

- Pobrania rozkazu - **Fetch**, w której zachodzi pobranie rozkazu z podręcznej kolejki rozkazów Prefetch i jeżeli istnieje taka potrzeba uzupełnienie tej kolejki
- Dekodowania rozkazu - **Decode**, w której określane są operacje do wykonania oraz obliczane adresy efektywne argumentów o ile istnieją
- Wykonania - **Execute** - z pamięci pobierane są argumenty rozkazów i wykonywane są operacje ustalone w fazie Decode
- Zapisania wyników - **Retire**, w której następuje zapisanie wyników w miejscu określonym w kodzie rozkazu oraz ustawienie znaczników rejestru statusowego

Każda instrukcja przechodzi przez powyższe fazy co przyspiesza wykonywanie. Sposób taki można porównać przez analogię do taśmy produkcyjnej. Dobra organizacja pracy i podział na różne fazy przyspiesza produkcję.



#### 14. Omów i porównaj transmisję danych metodą PIO i DMA.

Przesyłanie danych między urządzeniem peryferyjnym a pamięcią operacyjną może odbywać się w trybie programowanego wejścia-wyjścia PIO (Programmed Input/Output) lub w trybie bezpośredniego dostępu do pamięci DMA (Direct Memory Access). W trybie programowanego wejścia-wyjścia całym procesem przesyłania danych zajmuje się procesor. W trybie bezpośredniego dostępu do pamięci procesor przechodzi w stan wysokiej impedancji, a na czas transmisji danych kontrolę nad magistralami komputera przejmuje układ DMA.

*Tutaj powinno się znaleźć porównanie tych metod. Jak dobrze pamiętam z wykładu, chodziło o ilość cykli zegarowych, które te układy potrzebowały do przesłania jednego słowa. Jeśli mnie pamięć nie myli, to było coś, że DMA potrzebuje 2 cykli zegarowych, a PIO 3 lub 4 (wykonanie rozkazu MOV).*

## 15. Opisz na czym polega asynchroniczna transmisja (interfejs RS-232C)

Asynchroniczny: każdy bajt jest przesyłany niezależnie, i jest poprzedzony bitem START (stan 0), po którym są przesyłane bity danych od 0 do 7 (lub mniej: stosuje się bajt od 5 do 8 bitów), po nich opcjonalnie bit parzystości (do wyboru: tak, by łączna ilość jedynek w danych i tym bicie była parzysta (Even Parity), albo nieparzysta (Odd Parity), albo by miał określoną wartość 0 albo 1 (Stick Parity) - łącznie 4 możliwości), i na koniec bit (lub bity) STOP (stan 1; dla słowa 5-bitowego 1 lub 1.5 bitu, dla dłuższych 1 lub 2; jest to gwarantowany odstęp przed bitem START następnego bajtu, może on jednak być dowolnie długi); bity mają czas trwania określony przez stronę wysyłającą, strona odbierająca odmierza czas od zbocza 1->0 na początku bitu start i próbuje stan w połowie długości bitu; wykrycie wartości '1' w połowie bitu START jest interpretowane jako "fałszywy start"; wykrycie wystąpienia '0' pół odstępu czasu po rozpoczęciu bitu STOP jest interpretowane jako "błąd ramki" (framing error).

Inaczej:

Dane przesyłane za pośrednictwem asynchronicznego łącza szeregowego zorganizowane są w ramki SDU (Serial Data Unit) składające się z:

1. Znacznika początku ramki (bit startu o wartości 0)
2. 5, 6, 7 lub 8 bitów danych
3. Opcjonalnie: bit parzystości
4. Znacznika końca ramki (1, 1½, 2 bitów stopu o wartości 1)

Aby nawiązać połączenie, oba urządzenia muszą mieć ustawione takie same parametry (częstotliwość i format ramki). Przekształcaniem danych z postaci równoległej na szeregową i odwrotnie zajmuje się UART (Universal Asynchronous Receiver-Transmitter). W zależności od typu UART transmisja może odbywać się z maksymalną prędkością 56 kb/s lub 115200 kb/s w dwóch trybach: jednokierunkowym lub dwukierunkowym.

Do reprezentowania stanów logicznych RS-232C wykorzystuje napięcia z zakresu od -15 do +15V (logiczne 0: od +3 do +15; logiczna 1: od -3 do -15). Maksymalna odległość między nadajnikiem i odbiornikiem nie przekracza 200m. Obsługa łącza szeregowego może być realizowana przez zgłoszenie przerwania przez układ UART lub okresowe testowanie rejestru stanu transmisji układu UART przez procesor.

## 16. Omów odwzorowanie bezpośrednio pamięci podręcznej.

Odwzorowanie bezpośrednio polega na takim sposobie odwzorowania pamięci operacyjnej na pamięć podręczną, że  $n$ -ty wiersz każdej strony pamięci operacyjnej może zostać odwzorowany tylko w  $n$ -tym wierszu pamięci podręcznej.

Zaletą tej metody jest duża szybkość wyszukiwania informacji (tylko jedna operacja porównania) oraz prosta konstrukcja układu.

Ponieważ każdy wiersz pamięci operacyjnej ma stałą lokalizację w pamięci podręcznej, prowadzi to do ciągłej wymiany zawartości pamięci podręcznej, a tym samym do obniżenia wydajności systemu.

## 17. Omów koncepcje RISC architektury procesora.

- nieduży zestaw stosunkowo prostych rozkazów o stałej długości, pozwalający na znaczące uproszczenie konstrukcji jednostki sterującej i układu dekodera instrukcji oraz na zwiększenie szybkości przetwarzania rozkazów
- wielopoziomowa pamięć podręczna usprawniająca pobieranie zwiększonej w stosunku do procesorów CISC liczby rozkazów
- duża liczba rejestrów wewnętrznych umożliwiająca optymalizację odwołań do argumentów (zwłaszcza lokalnych)
- optymalizacja potoków wykonawczych z uwagi na duży udział w kodzie rozkazów rozgałęzień warunkowych oraz wywoływań i powrotów z podprogramów

## 18. Przedstaw idee superskalarnego przetwarzania rozkazów.

Technika przetwarzania superskalarnego polega na zastosowaniu kilku działających równolegle potoków przetwarzających dane skalarne, co pozwala na wykonywanie więcej niż jednego rozkazu w jednym cyklu zegara.

Podstawową trudnością w realizacji przetwarzania superskalarnego jest wzajemna współzależność rozkazów wykonywanych w poszczególnych potokach, wynikająca m.in. z rozgałęzień oraz uzależnień i konfliktów w dostępie do zasobów.