Zbigniew S. Szewczak
Podstawy Systemów Operacyjnych

Wykład 3

Struktura systemu komputerowego

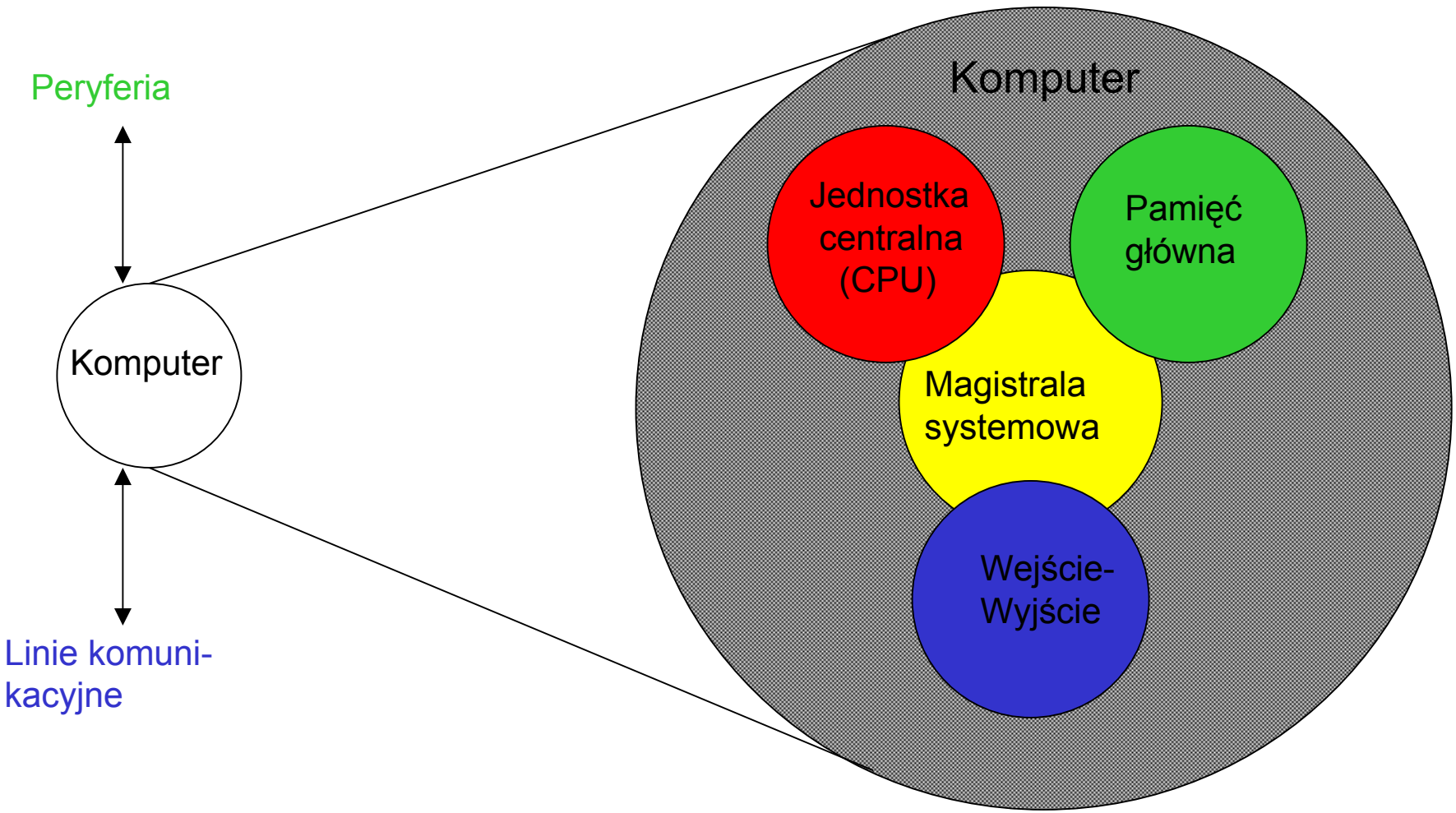
Toruń, 2011

terminu egzaminu

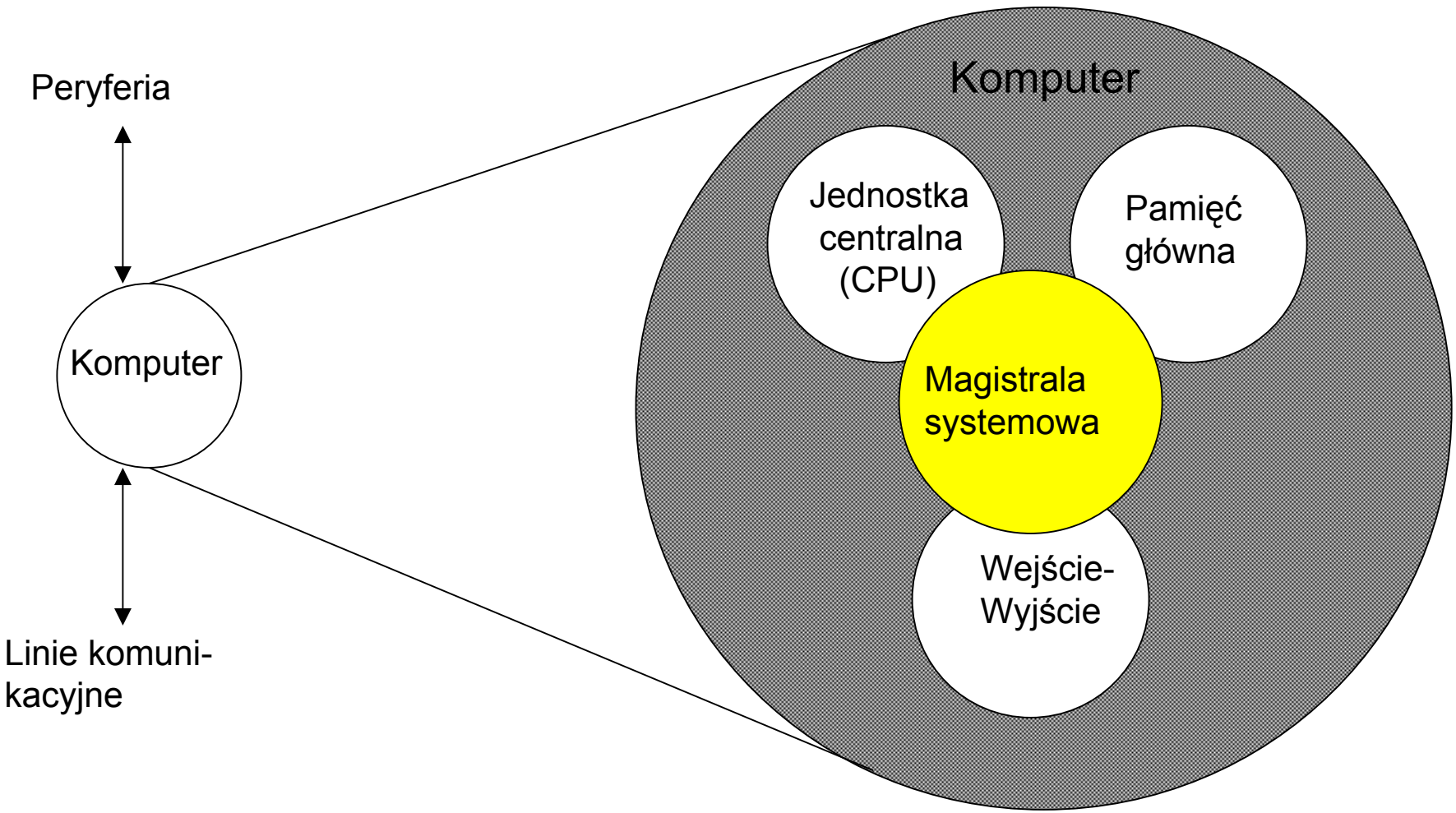
- I termin
- wtorek, 14.06.11, g.10 (AULA)

- II termin
- poniedziałek, 19.09.11, g.10 (S9)

Struktura komputera



Struktura komputera



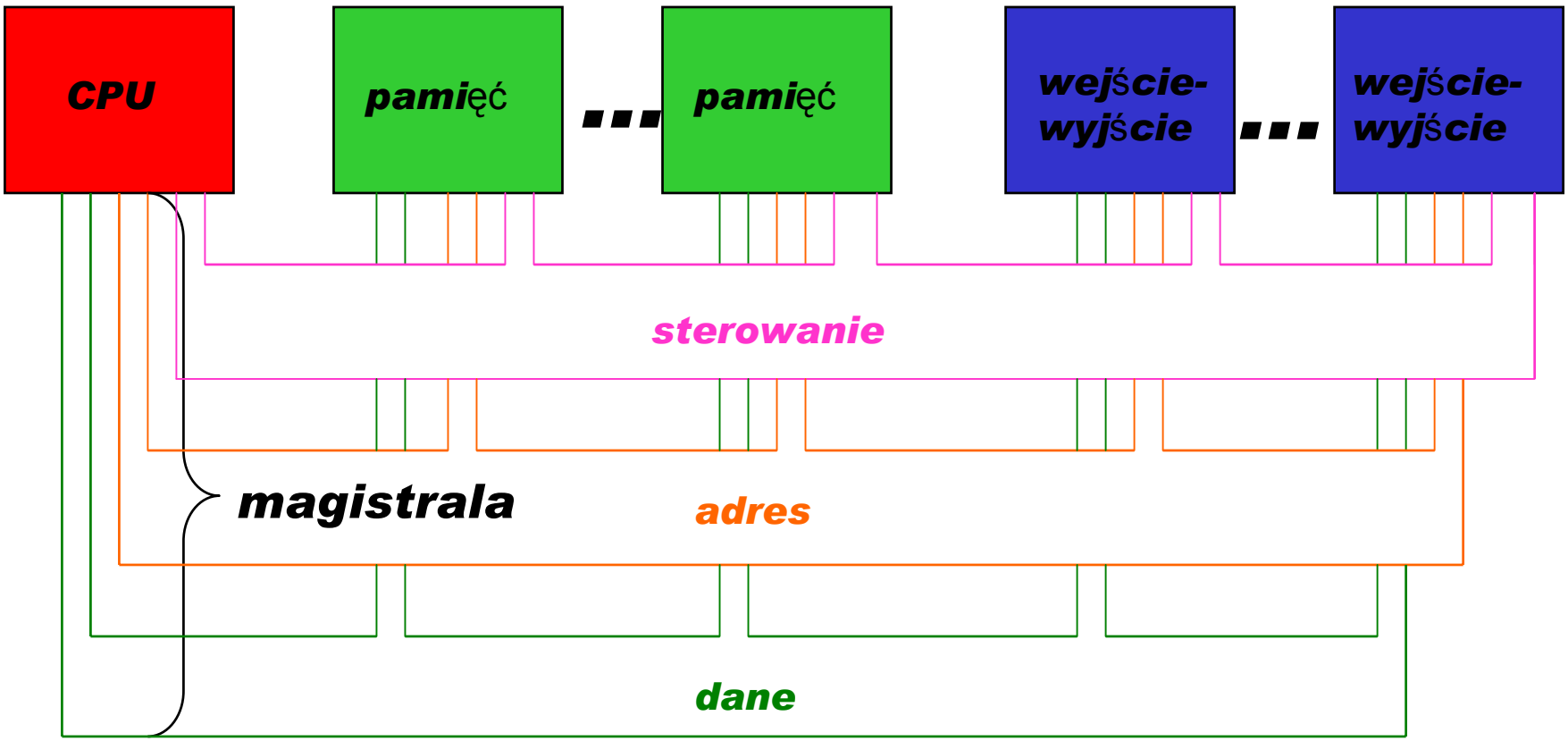
Magistrala systemowa

- System bus (ang.)
- Droga zapewniająca komunikację między (modułami): procesorem, pamięcią i wejściem-wyjściem
- Tryb rozgłaszania (ang. broadcast)
- 50 do 100 oddzielnych linii pogrupowanych w 8, 16, 32 linii zwanych szynami
  - linie danych - do przenoszenia danych
    - np. szyna danych 8b a rozkaz 16b: 2 x transfer z pamięci
  - linie adresowe - do określenia adresu danych
  - linie sterowania - do sterowania liniami
  - linie zasilania

Magistrala systemowa (c.d.)

- Arbitraż (ang. bus arbitration)
  - sterownik magistrali (arbiter) jest odpowiedzialny za sposób wykorzystania magistrali
    - określa które urządzenie jest nadrzędne: procesor czy moduł we/wy
- Taktowanie
  - synchroniczne
    - linia zegarowa transmituje sekwencje 1-0 zwane cyklem magistrali
    - np. 200MHz zegar ma cykl długości 5ns (=$5*10^{-9}$s)
    - przesunięcie względne sygnałów w magistrali (ang. bus skew) oznacza niejednakowe przybycie sygnału do celu na różnych liniach
  - asynchroniczne
    - wystąpienie zdarzenia jest zależne jedynie od zdarzenia poprzedzającego
- http://en.wikipedia.org/wiki/System_bus
- http://en.wikipedia.org/wiki/Bus_%28computing%29

Schemat magistrali



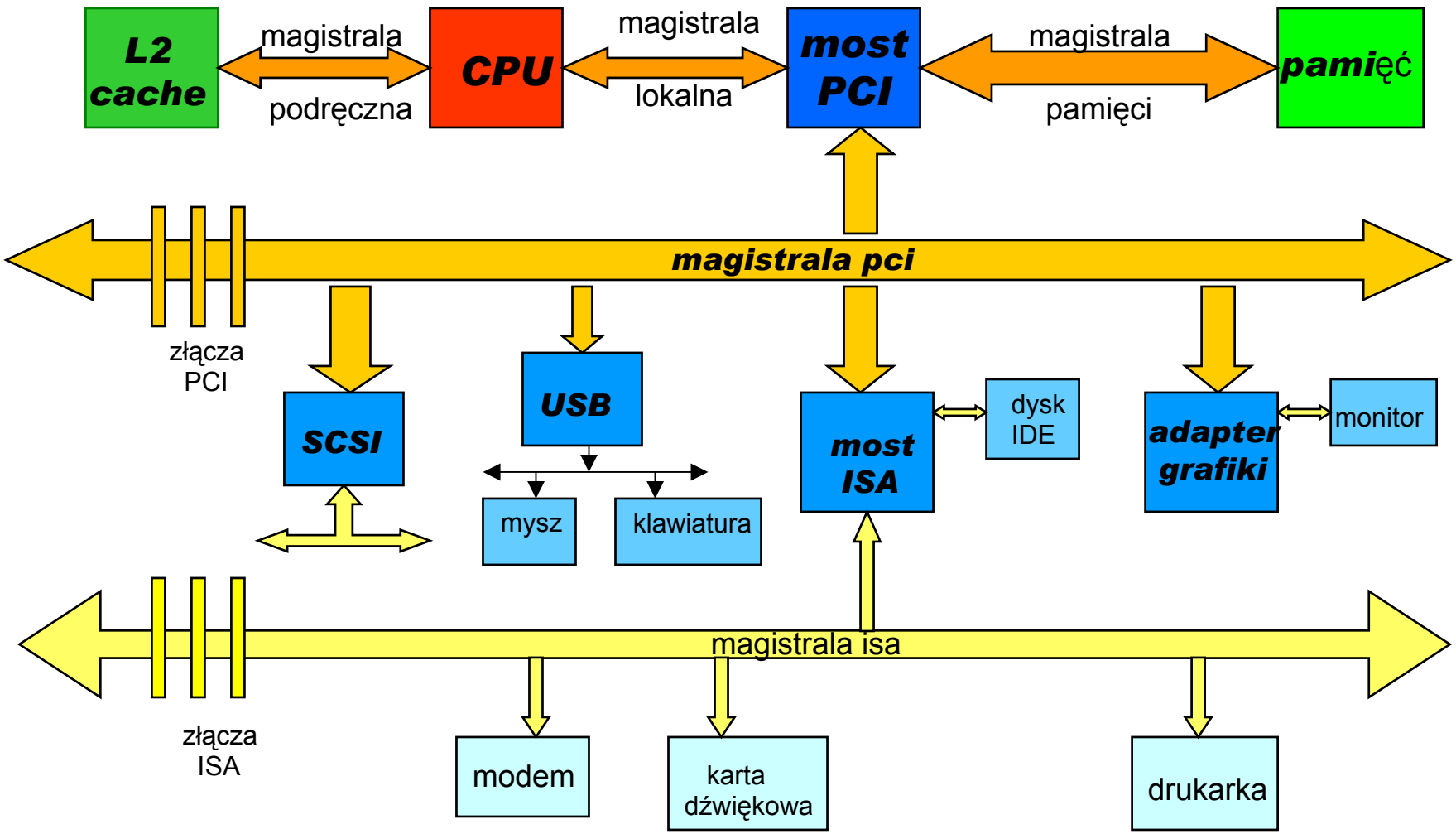
Struktury wielomagistralowe

- Duża ilość modułów opóźnia działanie magistrali
  - można zwiększać szybkość magistrali lub jej szerokość (z 32b na 64b)
  - jednak szybkość modułów (grafika, video) wzrasta bardziej
- Struktury wielomagistralowe o określonej hierarchii
  - lokalna magistrala: procesor <->pamięć podręczna
  - magistrala systemowa: we/wy komunikując się z pamięcią główną poprzez interfejs szyny rozszerzenia (ang. expansion bus) nie ma wpływu na działanie procesora
  - szyna rozszerzenia buforuje dane między magistralą systemową a sterownikami we/wy dołączonymi do szyny rozszerzenia (ang. expansion bus)

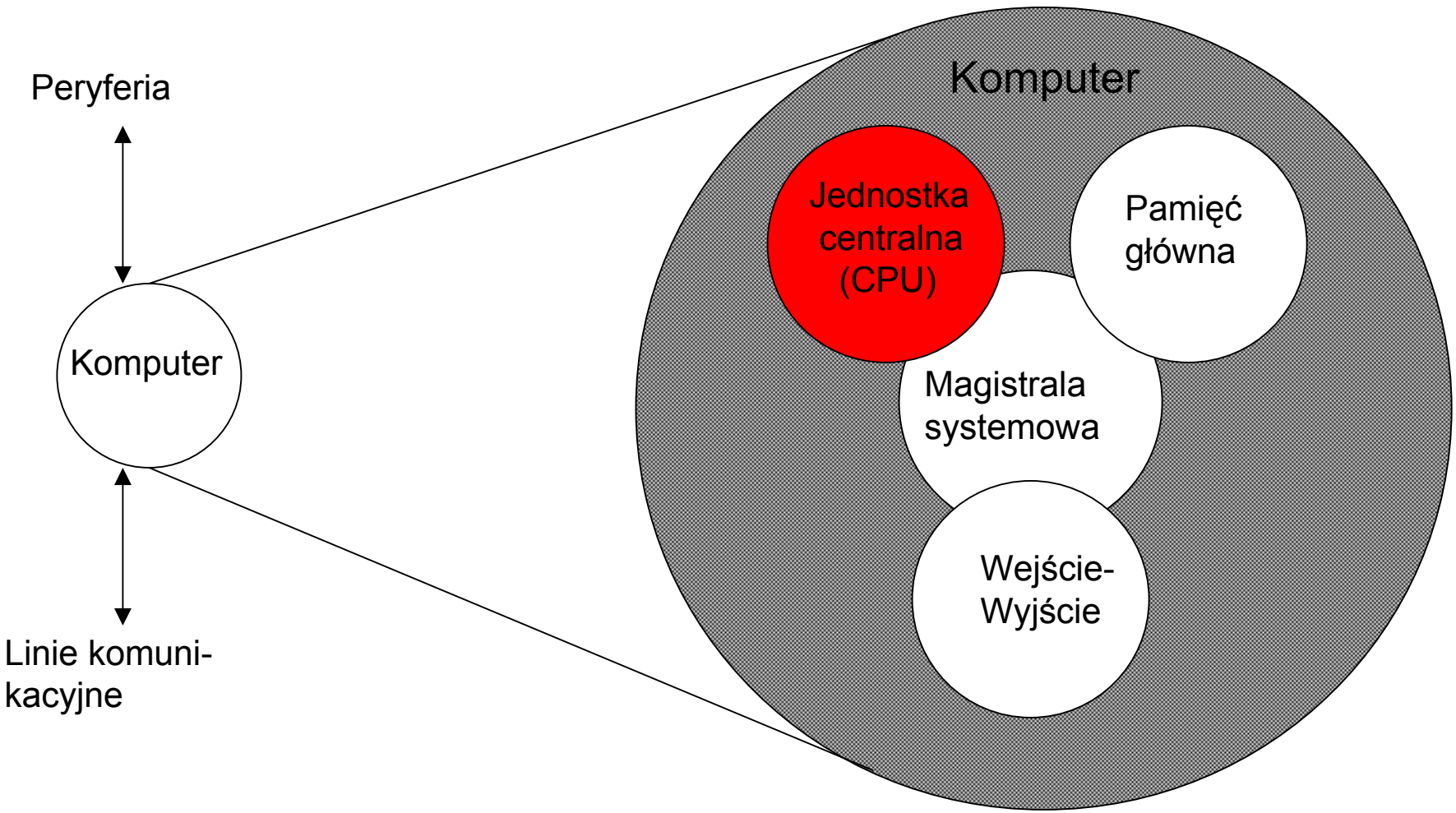
Przykład Pentium

- cache(L2), local bus, memory bus
  - PCI bridge chip: local, memory, szyna PCI
- ISA (ang. Industry Standard Architecture)
  - I/O bus: EIDE (ang. Enhanced Integrated Drive Electronics), ATA (ang. Advanced Technology Attachment) - dyski 16,67MBps
- PCI (ang. Peripheral Component Interconnect)
  - SCSI (ang. Small Computer System Interface)
    - szybkie we/wy do 160MBps (dyski,..)
  - USB (ang. Universal Serial Bus)
    - wolne urządzenia we/wy do 1.5MBps, (mysz, skaner,..)
  - IEEE 1394 (FireWire) - do 50MBps (multimedia)
  - 1024*768*3B = 2,25MB; 2.25MB*30 ekranów na s = 67,5MBps; 2 przesłania z dysku do pamięci i na ekran: razem 135MBps
  - sygnalizacja równoległa
- AGP (ang. Accelerated Graphics Port) – grafika

Struktura magistralowa Pentium



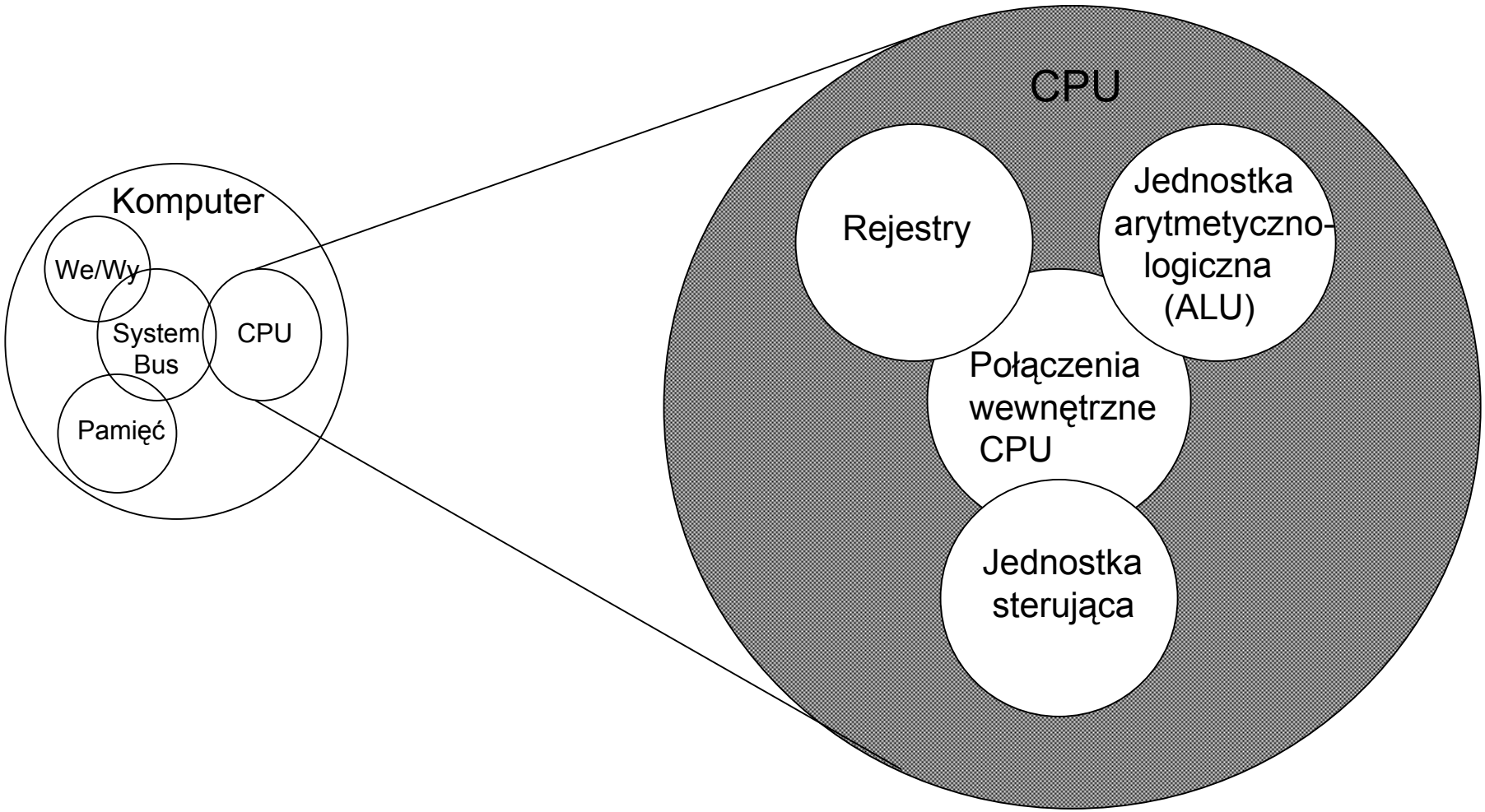
Struktura komputera



Funkcje jednostki centralnej

- CPU = procesor = jednostka centralna
- Pobieranie rozkazów z pamięci
- Interpretowanie rozkazów
- Pobieranie danych (z pamięci lub we/wy)
- Przechowywanie danych w pamięci
- Przetwarzanie danych - wykonywanie rozkazów
- Zapisywanie wyników (do pamięci lub na we/wy)

Schemat jednostki centralnej



Rejestry procesora

- licznik programu (PC) - adres rozkazu do pobrania
- rejestr rozkazu (IR) - kod rozkazu
- rejestr adresowy pamięci (MAR) - adres lokacji
- rejestr buforowy pamięci (MBR) - dane do/z pamięci

- rejestry PSW (ang. program status word) - słowo stanu programu, informacje o stanie

PSW (bity, flagi)

- znak - bit znaku ostatniej operacji
- zero - wynik operacji = zero
- przeniesienie (ang. carry) - przeniesienie w wielokrotnej precyzji
- równość (ang. equal) - wynik porównania logicznego
- przepełnienie - (ang. overflow)
- zezwolenie/blokowanie przerwań
- nadzorca - tryb systemu lub tryb użytkownika

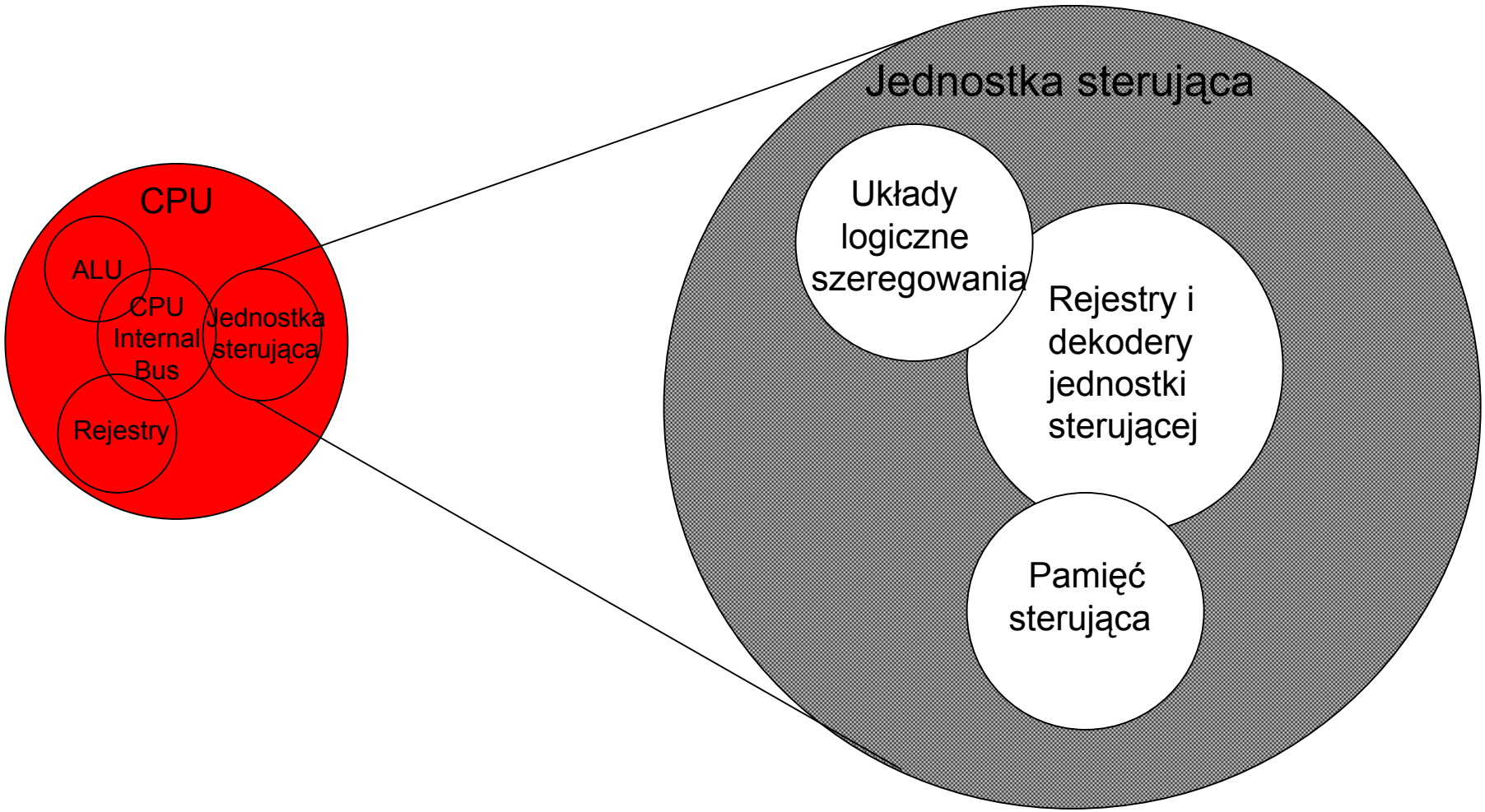
Procesory CISC i RISC

- CISC (ang. complex instruction set computer) - procesor o złożonej liście rozkazów
  - złożona lista rozkazów = łatwiejsze programowanie
  - trudne do zbudowania
  - rozkazy maszynowe realizowane są przez mikrozkazy
  - IBM360/370, VAX, Pentium (Intel), C25xx (router Cisco): M68360 (Motorola), 20MHz
- RISC (ang. reduced instruction set computer) - procesor o zredukowanej liście rozkazów
  - prosta lista rozkazów = trudniejsze programowanie
  - łatwe do zbudowania ale programy są dłuższe (ale pamięć tanieje)
  - SPARC, HP-PA, PowerPC, Athlon, C4500: MIPS R4600,100MHz
- http://bwrc.eecs.berkeley.edu/CIC/
- http://www.baznet.freeserve.co.uk/index.htm
- http://developer.intel.com/design/pentium/manuals/

Zegar

- Zegar - układ wysyłający regularne impulsy o stałej szerokości i częstotliwości
  - http://en.wikipedia.org/wiki/Computer_clock
- Umożliwia kontrolę relacji czasowych w CPU
- Odstęp między dwoma impusami to cykl zegara (ang. clock cycle time)
  - zwykle 1MHz-3GHz
  - za precyzję zegara odpowiada oscylator (rezonator) kwarcowy
  - http://en.wikipedia.org/wiki/Clock_rate
- W jednym cyklu zegara może być wykonywanych wiele rozkazów
  - w podcyklach wyznaczonych przez przesunięte sygnały wtórne zegara
- Zegar może być użyty do generownia ciągów liczb (pseudo)losowych
  - http://en.wikipedia.org/wiki/Random_number_generator
- Zegar może być celem ataku z zewnątrz (ang. timing attack)
  - http://en.wikipedia.org/wiki/Clock_drift
- Sygnał z zegara może przybyć o różnych czasach do różnych celów
  - http://en.wikipedia.org/wiki/Clock_skew

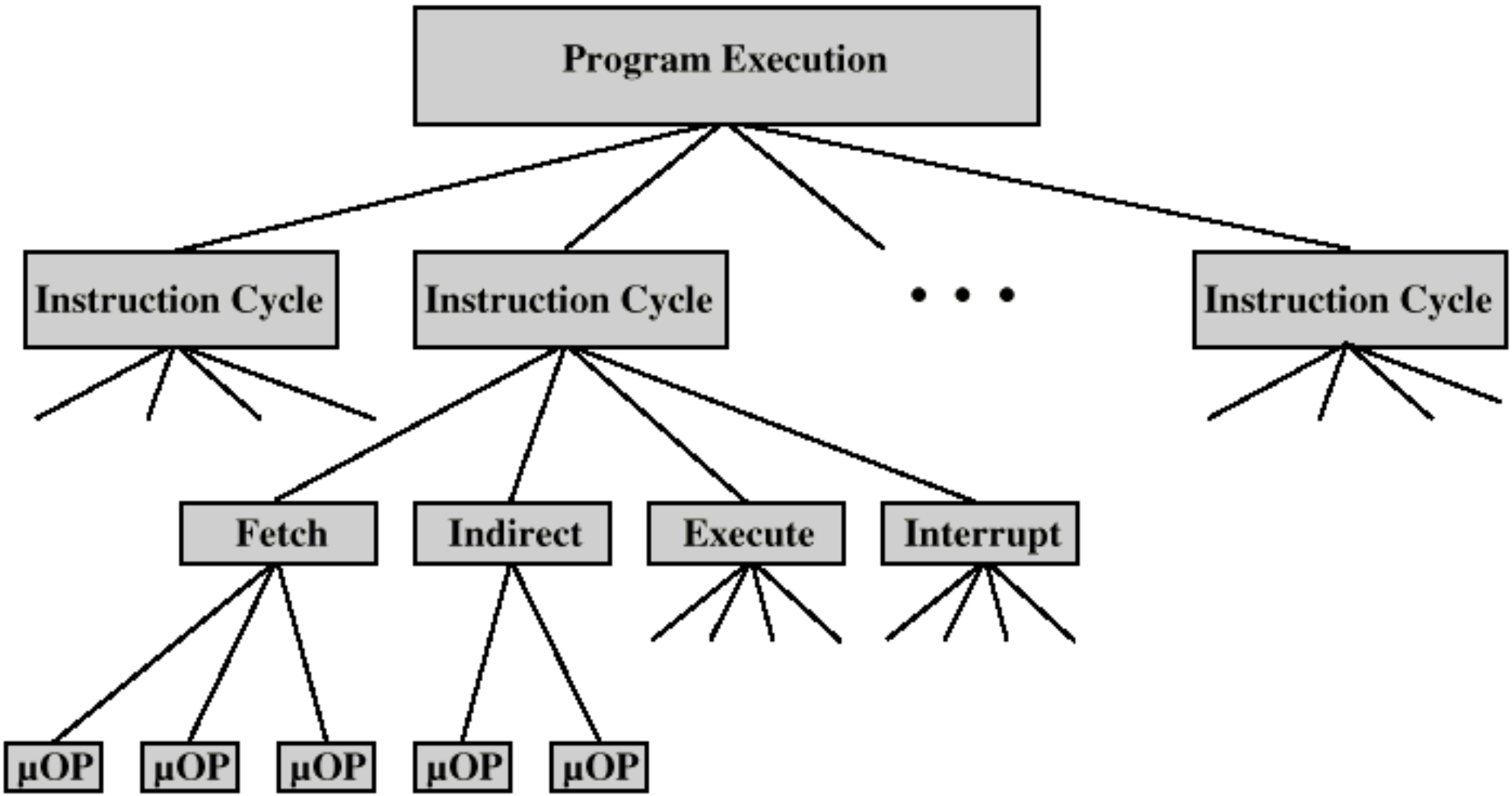
Jednostka sterująca



Mikrooperacje

- Komputer wykonuje programy
- Cykl pobierania i wykonania
- Każdy cykl jest złożony z mniejszych jednostek
  - dalsza dekompozycja rozkazu maszynowego
- Mikrooperacje (ang. micro-operations)
  - każdy z podcykli (np. pobrania, adresowania pośredniego, wykonania, przerwania) obejmuje jedną lub więcej mikrooperacji
- Mikrooperacje są elementarnymi operacjami (ang. atomic operation) wykonywanymi przez CPU
- Czas cyklu procesora – czas potrzebny do wykonania najkrótszej mikrooperacji
- Mikroprogramowanie - koncepcja zaproponowana przez M. V. Wilkes'a (1951)
  - zastosowanie: IBM System/360 (1964)

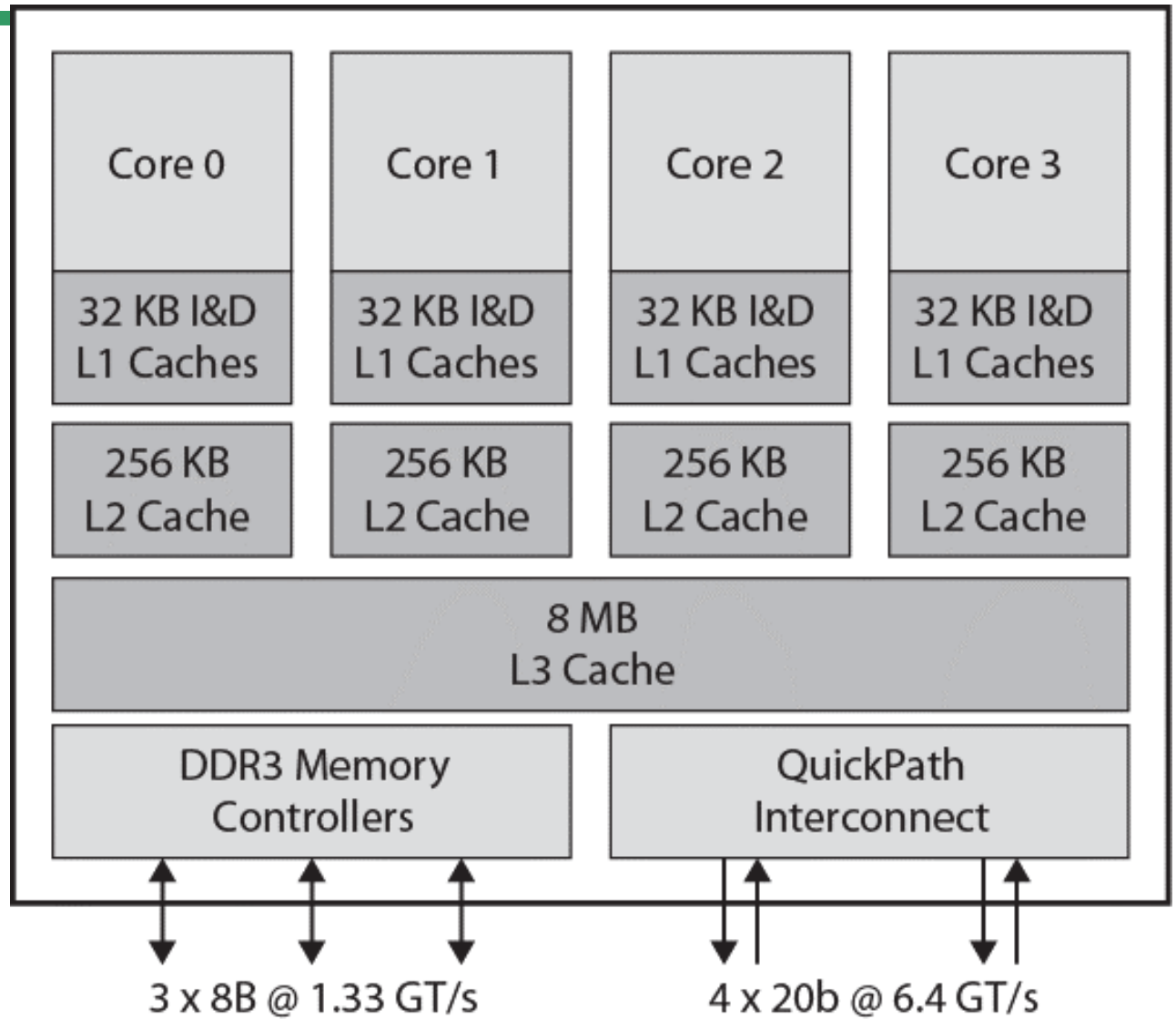
μOP- mikrooperacje



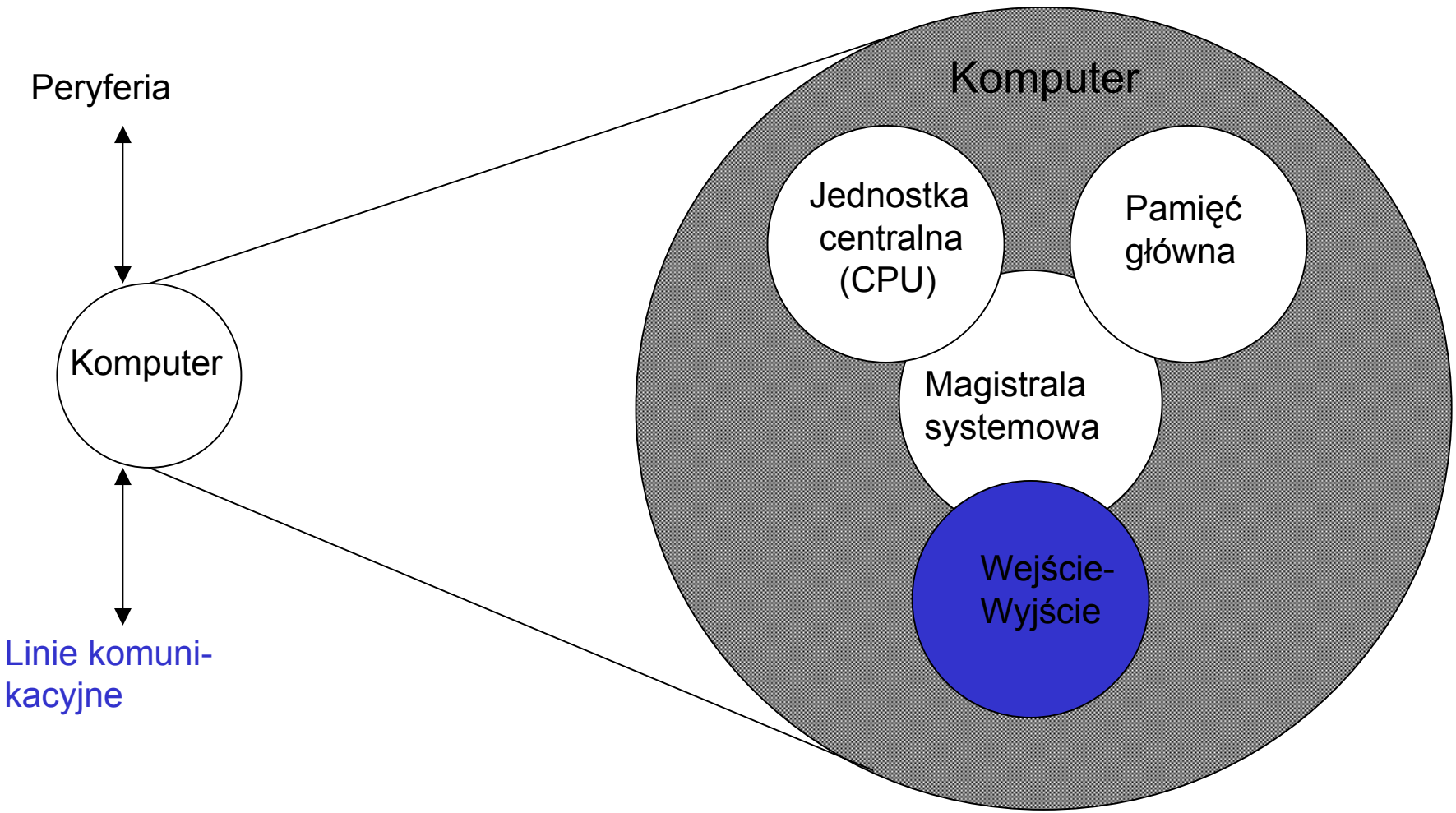
Intel Core i7

- Listopad 2008
- Cztery procesory x86 SMT
- Dedykowany L2, dzielony L3 cache (8MB)
- Spekulatywne wypełnianie cache'a (ang. speculative pre-fetch for caches) które mogą być wkrótce użyte
- Kontroler pamięci DDR3 na chipie
  - trzy kanały szerokości 8 B (szerkość szyny 192b) dające 32GB/s
  - pozwala na wyeliminowanie FSB (ang. front side bus)
- QuickPath Interconnection (QPI)
  - łącze point-to-point do komunikowania się pomiędzy rdzeniami (zapewnia zgodność cache'a)
  - 6.4G transferów per second, 16b per transfer
  - komunikacja dwukierunkowa (ang. bidirectional)
  - Przepustowość = 2*2*6.4GB=25.6GB/s

Intel Core i7 Diagram



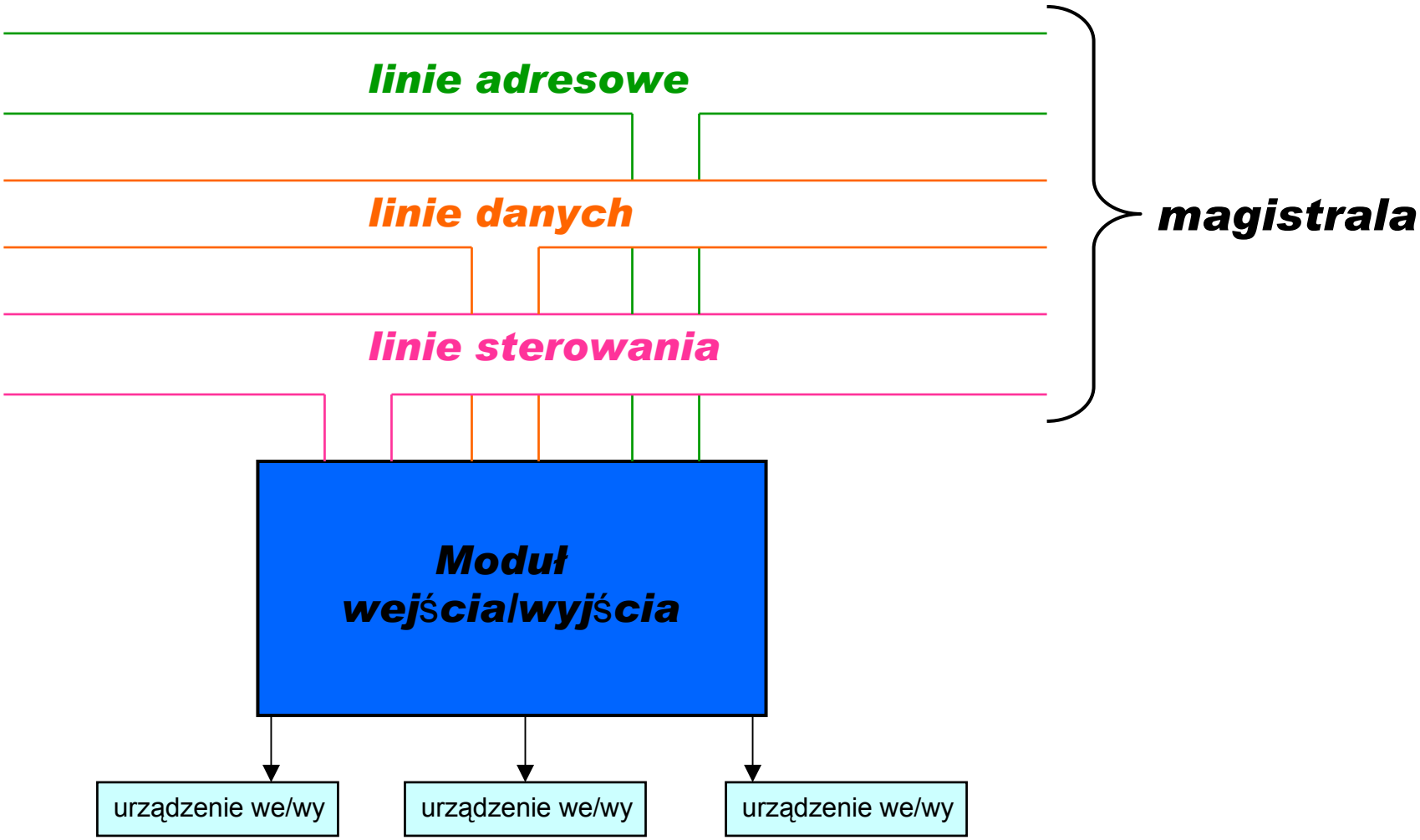
Struktura komputera



Wejście/Wyjście

- Dlaczego nie łączy się urządzeń peryferyjnych bezpośrednio z magistralą systemową?
  - wielka różnorodność urządzeń we/wy
  - przesyłanie różnej ilości danych
    - z różną prędkością
    - w różnym formacie
  - urządzenia we/wy za wolne dla procesora i pamięci
- Potrzeba modułu we/wy
  - interfejs z procesorem i pamięcia przez magistralę
  - interfejs z urządzeniami peryferyjnymi

Model modułu we/wy



Kategorie wejścia/wyjścia

- Rodzaje urządzeń we/wy
  - przeznaczone do odczytu przez człowieka (np. monitor ekranowy, wydruk, dźwięk)
  - przeznaczone do odczytu przez maszynę (np. dyski magnetyczne, taśmy, czujniki w robotach)
  - komunikacyjne (np. modem, karta sieciowa)

Funkcje modułu we/wy

- Sterowanie i taktowanie (zegar)
- Komunikacja z procesorem
- Komunikacja z urządzeniem
- Buforowanie danych
- Wykrywanie błędów

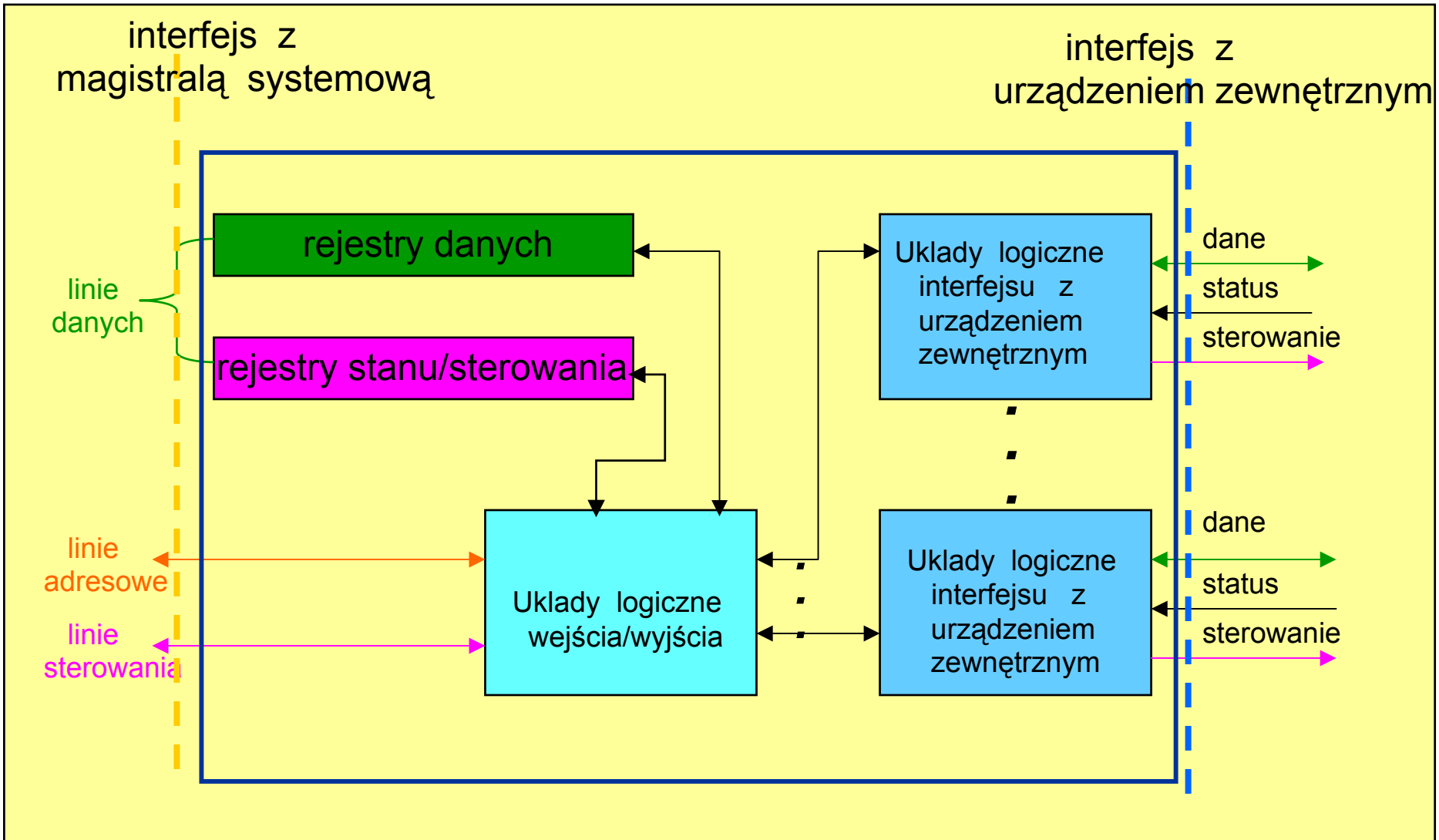
Funkcje modułu we/wy (c.d.)

- Sterowanie i taktowanie (zegar), np.:
  - procesor żąda statusu urządzenia we/wy
  - moduł we/wy udziela odpowiedzi
  - jeśli urządzenie jest gotowe procesor wydaje rozkaz
  - moduł we/wy otrzymuje dane z urządzenia
  - dane są przenoszone z modułu we/wy do procesora
- Komunikacja z procesorem
  - dekodowanie rozkazu przez moduł we/wy
  - przesyłanie danych poprzez magistralę
  - przesyłanie informacji o stanie modułu we/wy
  - rozpoznawanie adresu modułu we/wy

Funkcje modułu we/wy

- Komunikacja z urządzeniem
  - sygnały sterujące z modułu we/wy (np. ustaw głowicę dysku)
  - sygnały stanu do modułu we/wy (np. ready, not-ready)
  - dane do/z modułu we/wy
  - dane specyficzne dla urządzenia do/z otoczenia
- Buforowanie danych
  - dane są buforowane w module we/wy i wysyłane do urządzenia z prędkością stosowaną dla tego urządzenia
- Wykrywanie błędów, np.:
  - 7-bitowy kod ASCII, ósmy bit ustawiany w zależności od tego czy jest parzysta liczba jedynek czy nie

Struktura modułu we/wy



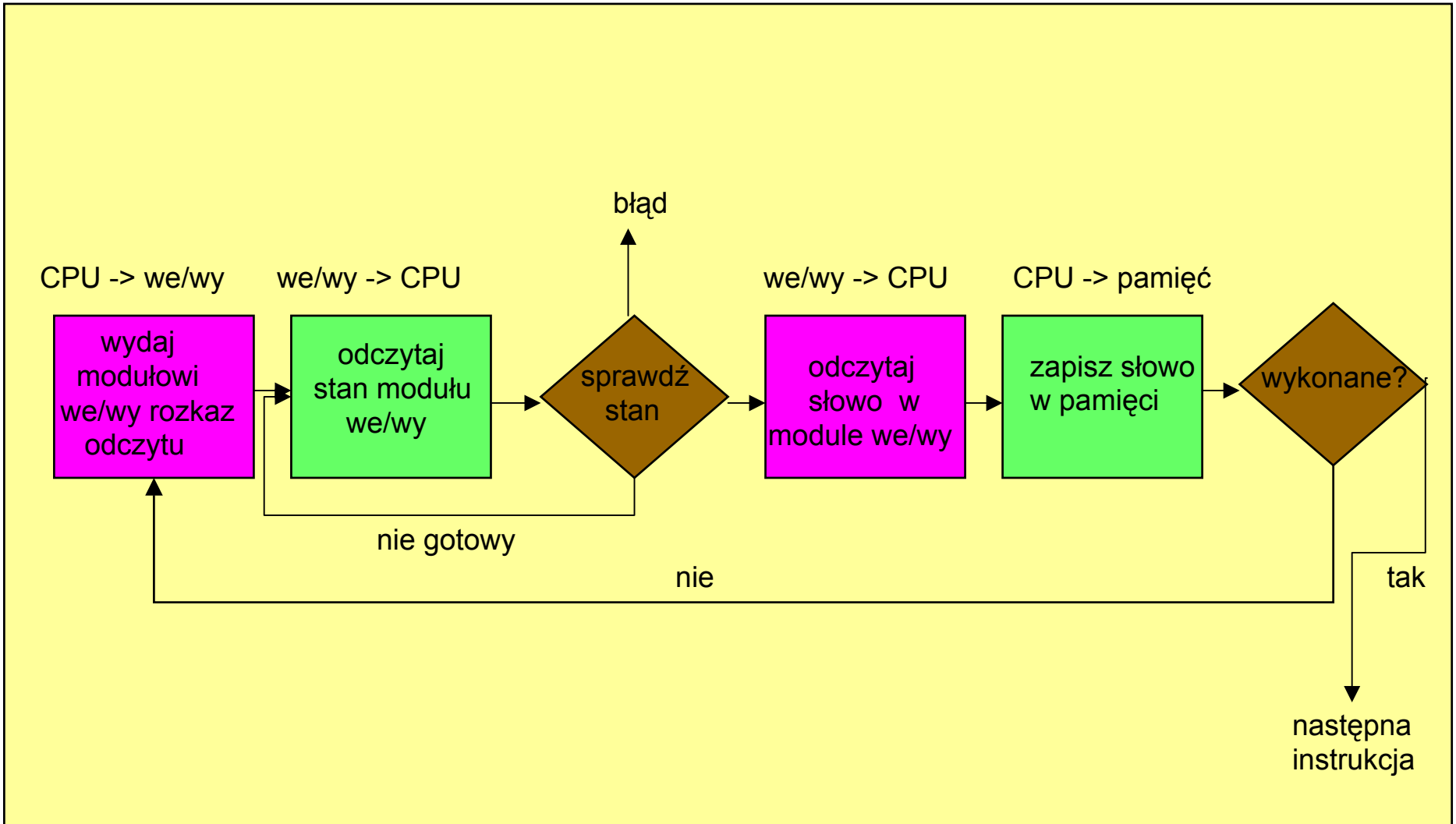
Rodzaje modułów we/wy

- Sterownik wejścia/wyjścia (urządzenia) - prymitywny, wymaga szczegółowego sterowania
  - mikrokomputery, minikomputery
- Kanał (procesor) wejścia/wyjścia - przejmuje większość obciążenia we/wy CPU, wysoki priorytet dojścia do procesora
  - superkomputery, duże instalacje

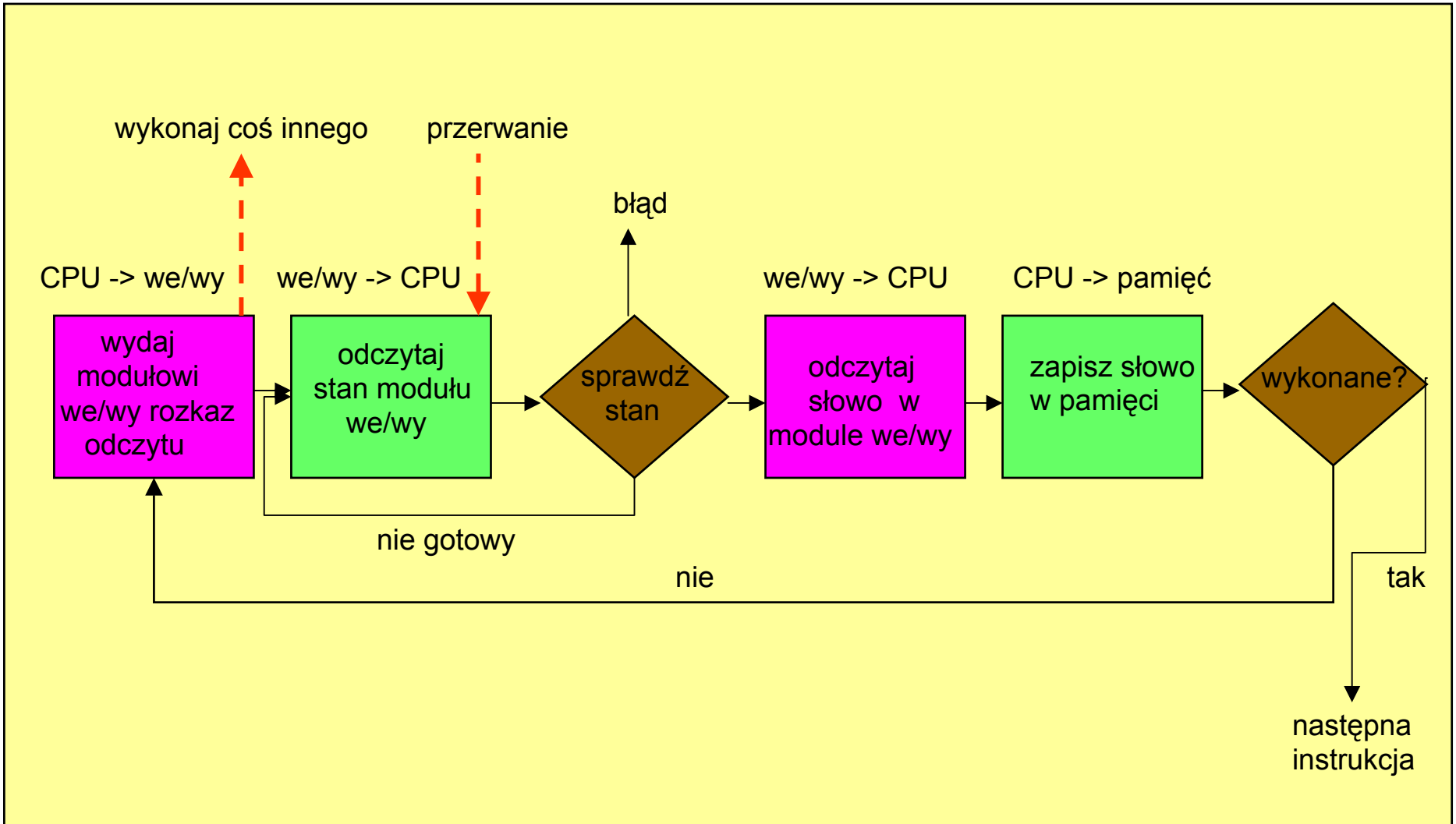
Sposoby realizacji we/wy

- programowane - dane są wymieniane między procesorem a modułem we/wy, procesor czeka na zakończenie operacji we/wy
- sterowane przerwaniami - procesor wydaje operację we/wy i wykonuje dalsze rozkazy do momentu zakończenia operacji we/wy (przerwanie we/wy)
- bezpośredni dostęp do pamięci (ang. direct memory access - DMA) - moduł we/wy wymienia dane bezpośrednio z pamięcią bez udziału procesora

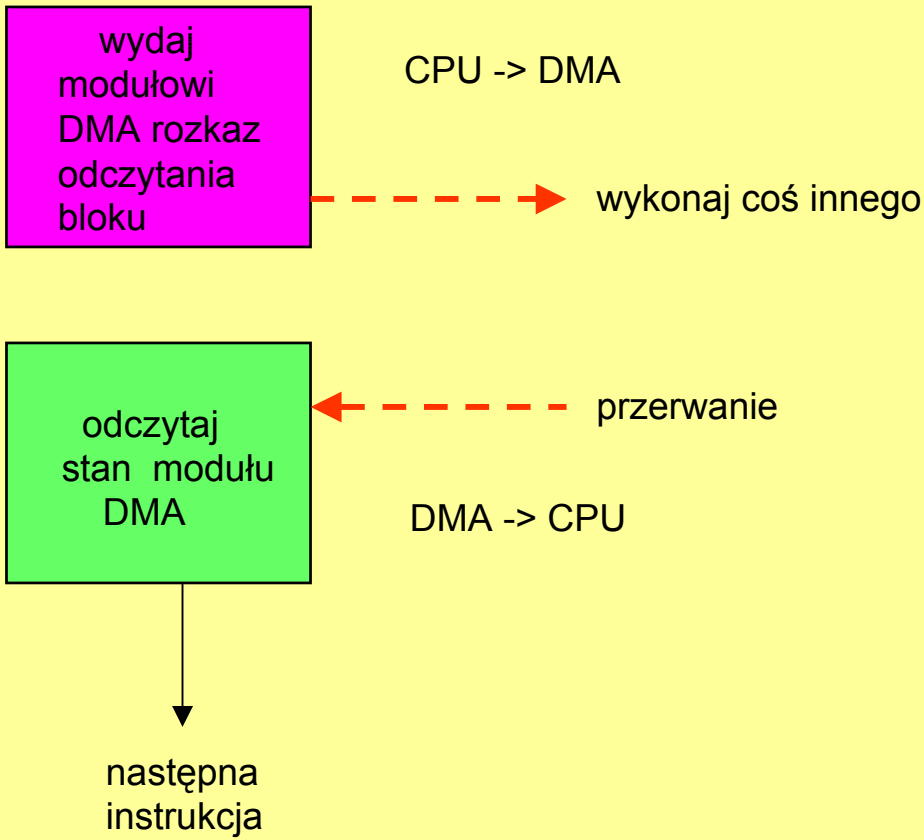
We/wy progamowane



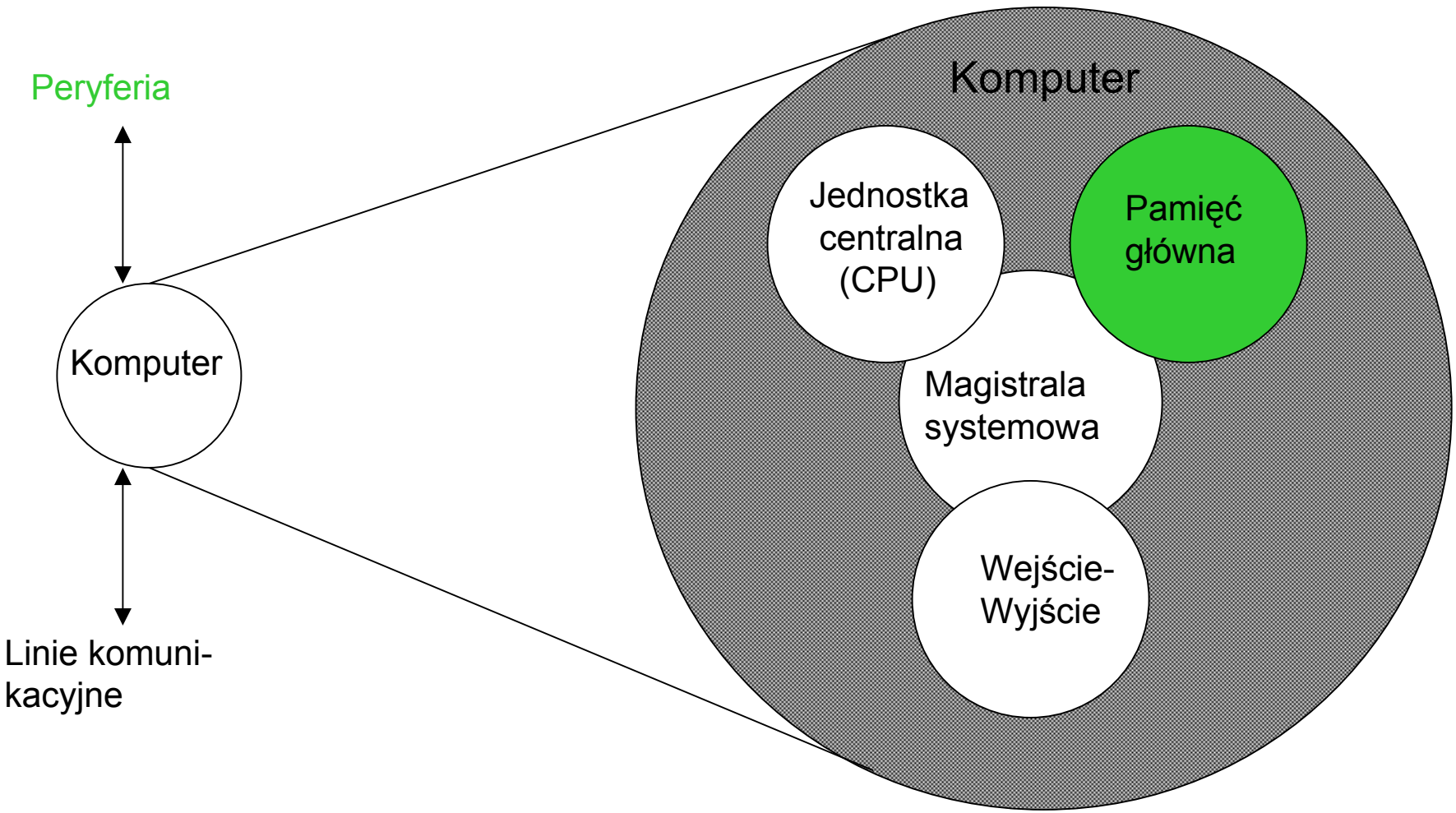
We/wy sterowane przerwaniami



Bezpośredni dostęp do pamięci



Struktura komputera



Pamięć - lokalizacja, miara

- CPU
- wewnętrzna (główna)
- zewnętrzna (pomocnicza)

- komórka pamięci – najmniejsza adresowalna jednostka pamięci – zwykle jeden bajt (=8b)
  - Burroughs B1700 – 1b; IBM 1130 – 16b; Honeywell 6180 – 36b
- słowo – zgrupowane komórki
  - długość słowa - 8, 16, 32, 64 bitów (b)
- pojemność pamięci mierzona w ilości słów (lub w bajtach)
  - CRAY C90 ma słowo 64b a liczba całkowita jest reprezentowana przez 46b

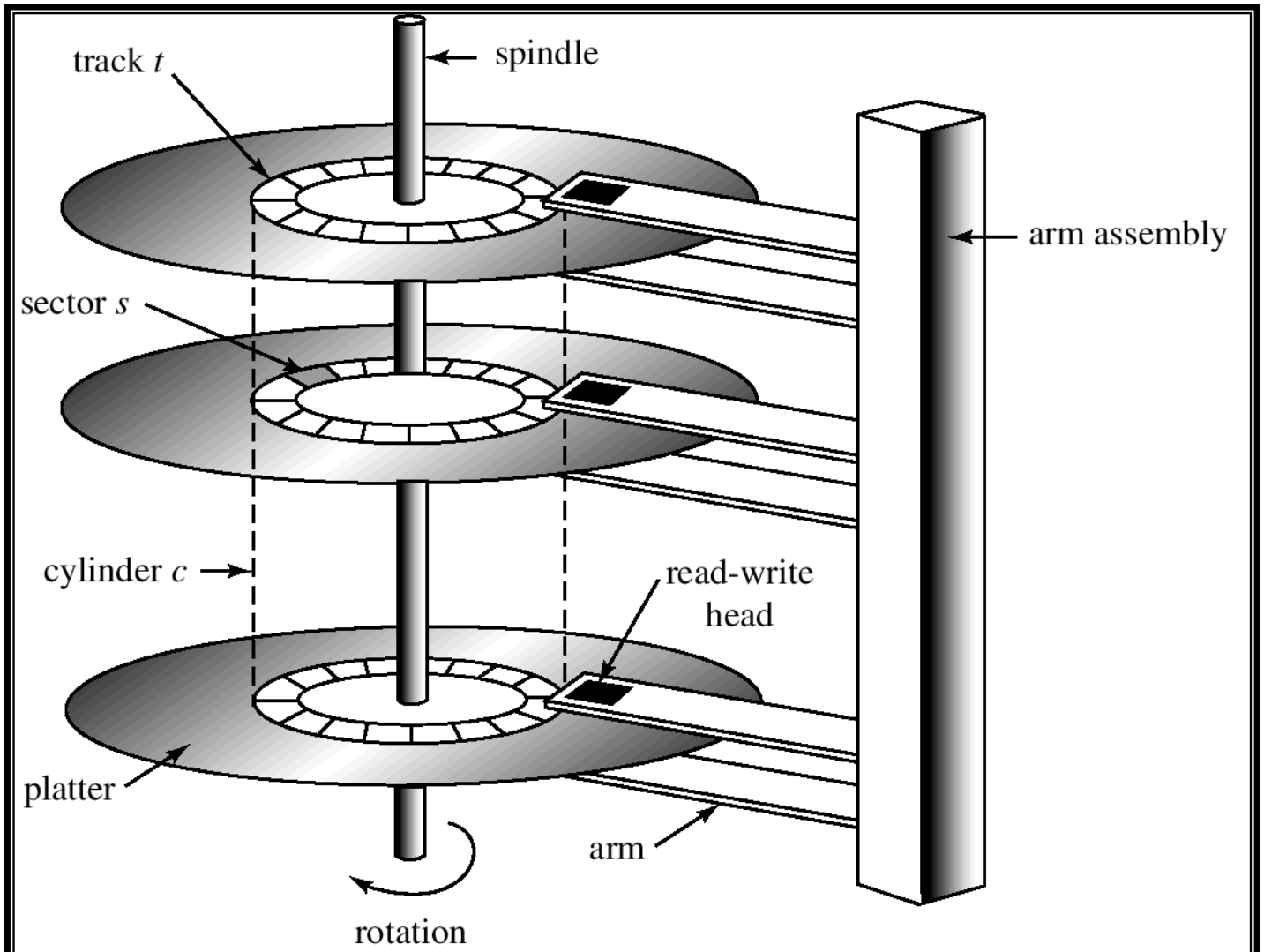
Pamięć - jednostka transferu

- pamięć wewnętrzna
  - zwykle szerokość szyny danych
- pamięć zewnętrzna
  - zwykle bloki o wiele większe niż słowo
- jednostka adresowalna
  - najmniejsza ilość danych dająca się jednoznacznie zaadresować (zwykle słowo ale może być to klaster na dysku)
  - N = ilość adresowalnych jednostek; A = długość adresu
  - zależność: N = 2^A

Pamięć - metody dostępu (1)

- dostęp sekwencyjny (ang. sequential)
  - dostęp liniowy blok po bloku wprzód lub wtył
  - czas dostępu zależy od pozycji bloku względem pozycji bieżącej
  - np. taśmy
- dostęp bezpośredni (ang. direct)
  - każdy blok ma unikalny adres
  - czas dostępu realizowany przez skok do najbliższego otoczenia i sekwencyjne przeszukiwanie
  - np. dysk magnetyczny

Mechanizm dysku



Pamięć - metody dostępu (2)

- dostęp swobodny (ang. random)
  - każda adresowalna lokacja w pamięci ma unikatowy, fizycznie wbudowany mechanizm adresowania
  - czas dostępu nie zależy od poprzednich operacji i jest stały
  - np. RAM
- dostęp skojarzeniowy (ang. associative)
  - dane są lokalizowane raczej na podstawie porównania z ich zawartością niż na podstawie adresu
  - czas dostępu nie zależy od poprzednich operacji i jest stały
  - np. pamięć podręczna (ang. cache)

Pamięć - wydajność (definicje)

- czas dostępu
  - dostęp swobodny : czas niezbędny do zrealizowania operacji
  - dostęp nieswobodny : czas potrzebny na ustawienie głowicy
- czas_cyklu pamięci
  - czas dostępu + czas, który musi upłynąć aby mógła nastąpić kolejna operacja (zapis, odczyt)
- szybkość transferu
  - dostęp swobodny : 1/czas_cyklu
  - dostęp nieswobodny : $T_n = T_a + n/r$
    - T_n - śr. czas operacji na n-bitach; T_a = śr. czas dostępu
    - r - szybkość transferu w b/sek (bps)

Pamięć - rodzaje fizyczne

- półprzewodnikowa (ang. semiconductor)
  - RAM
- magnetyczna
  - dysk & taśma
- magneto-optyczna
  - CD & DVD

Pamięć - własności fizyczne

- zanikanie, rozpad (ang. decay)
- ulotność (ang. volatility)
- wymazywalność (ang. erasable)
- zasilanie do utrzymania zawartości

Pamięć - organizacja

- Naturalny porządek bitów w słowie
  - Czy numerujemy od 0 czy od 1?
  - Czy numerujemy bity od lewej do prawej czy odwrotnie?
  - Czy numerujemy bajty w słowie od lewej do prawej czy odwrotnie?
- Nie zawsze oczywisty
  - Big-Endian, Little-Endian
    - problem transmisji jeśli jeden komputer jest grubokońcowy a drugi cienkokońcowy

Hierarchia pamięci

- Jak osiągnąć efektywność ze względu na koszt?
- rejestry
  - w CPU
- pamięć wewnętrzna lub główna
  - pamięć podręczna (cache) - może być wiele poziomów
  - pamięć główna - “RAM”
- pamięć zewnętrzna
  - dyskowa pamięć podręczna
  - pamięć dyskowa
  - pamięć taśmowa lub dysk optyczny

Hierarchia pamięci (c.d.)

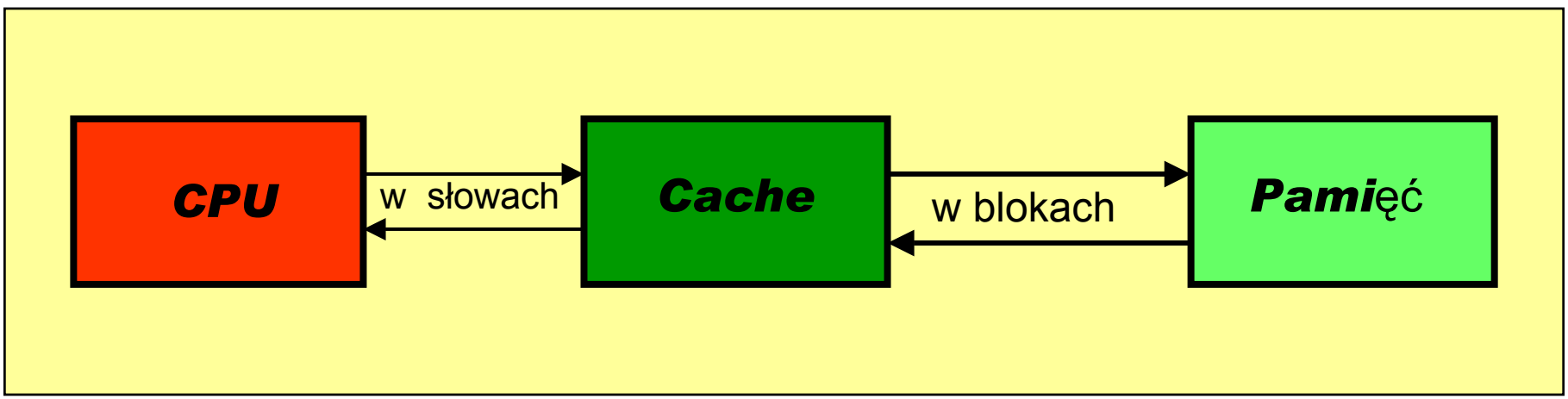
- od góry do dołu:
  - malejący koszt na bit
  - rosnąca pojemność
  - rosnący czas dostępu
  - malejąca częstotliwość dostępu do pamięci przez procesor
- Metody programowe hierarchizacji z wykorzystaniem bufora dyskowego
  - grupowanie danych przez transfery większych ilości danych
  - czytanie danych z bufora a nie z dysku

Zasada lokalności odniesień

- zasada lokalności odniesień (ang. locality of reference) oznacza, że w czasie wykonania programu odwołania do danych i rozkazów mają tendencję do gromadzenia się
- przyczyna: programy zwykle zawierają tablice deklaracji zmiennych oraz stałych i wiele pętli iteracyjnych i podprogramów
  - lokalność przestrzenna – grupowanie odniesień do tych samych miejsc w pamięci
    - skłonność do sekwencyjnego sięgania po rozkazy lub dane (np. tablica)
  - lokalność czasowa - skłonność do odwołań do ostatnio wykorzystywanych miejsc w pamięci (np. pętla iteracyjna)
- operacje na tablicach - dostęp do zgrupowanych słów
- wykorzystanie zasady lokalności odniesień pozwala na zmniejszenie częstotliwości dostępu
- przykład: pamięć podręczna (ang. cache, fr. cacher) procesora

Pamięć podręczna a główna

- wydajność CPU a szybkość dostępu do pamięci
- mała (od 1kB do 512kB) szybka pamięć (droga)
- typowy współczynnik czasu dostępu 5:1
- wykorzystanie zasady lokalności odniesień



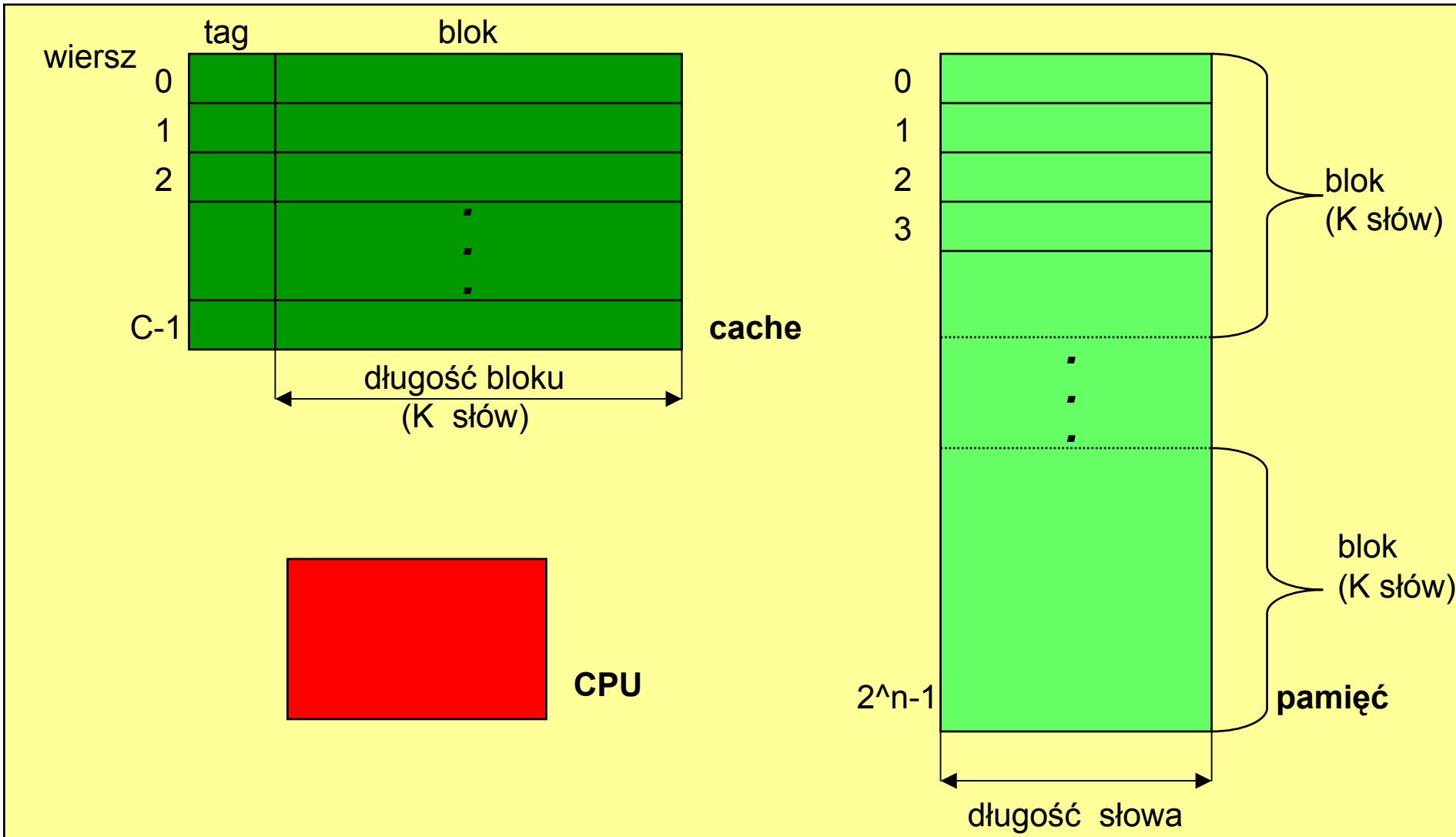
Pamięć podręczna - działanie

- Cache zawiera fragment pamięci głównej
- Procesor sprawdza czy aktualnie potrzebne do wykonania rozkazu słowo z pamięci jest w cache'u
  - jeśli nie, to blok pamięci o ustalonej liczbie K słów zawierający potrzebne słowo jest ściągnięty do pamięci podręcznej
- Cache zawiera znaczniki identyfikujące bloki pamięci głównej
- Zrealizowana po raz pierwszy w 1969 na komputerze IBM S/360 model 85

Pamięć podręczna (c.d.)

- rozmiar cache'u
  - mały cache poprawia wydajność
- rozmiar bloku (od 4B do 128B)
  - większy blok: większe prawdopodobieństwo, że słowo jest w cache'u (zasada lokalności)
  - zbyt duży blok: większe prawdopodobieństwo, że słowo jest w usuniętym bloku
- funkcja mapująca (ang. mapping function)
  - określa lokalizację bloku w cachu'u
  - przykład: 2b tag(znacznik), 6b adres
    - 01: 010000, 010001, 010010, 010011, 010100, 010101, 010110, 010111, 011000, 011001, 011010, 011011, 011100, 011101, 011110, 011111

Struktura pamięci podręcznej i głównej



Mapowanie bezpośrednie

- Mapowanie bezpośrednie (ang. direct mapping)
- Każdy blok w pamięci głównej jest odwzorowywany na tylko jeden możliwy wiersz (ang. line) pamięci
  - tzn. jeśli blok jest w cache'u to tylko w ściśle określonym miejscu
- Adres jest dzielony na dwie części
  - najmniej znaczących w-bitów identyfikuje jednoznacznie słowo (ang. word) lub bajt w pamięci
  - najbardziej znaczących s-bitów określa jeden z 2^s bloków pamięci
    - najbardziej znaczące bity są dzielone na pole wiersza złożone z r bitów oraz znacznik w postaci s-r bitów (najbardziej znacząca część)

Wiersz	Przypisane bloki w pamięci głównej
0	0, 2^r, 2*2^r, 3*2^r,....., 2^s - 2^r
1	1, 2^r + 1, 2*2^r + 1, 3*2^r + 1,…, 2^s - 2^r +1
2^r -1	2^r - 1, 2*2^r - 1, 3*2^r - 1, 4*2^r - 1,....., 2^s - 1

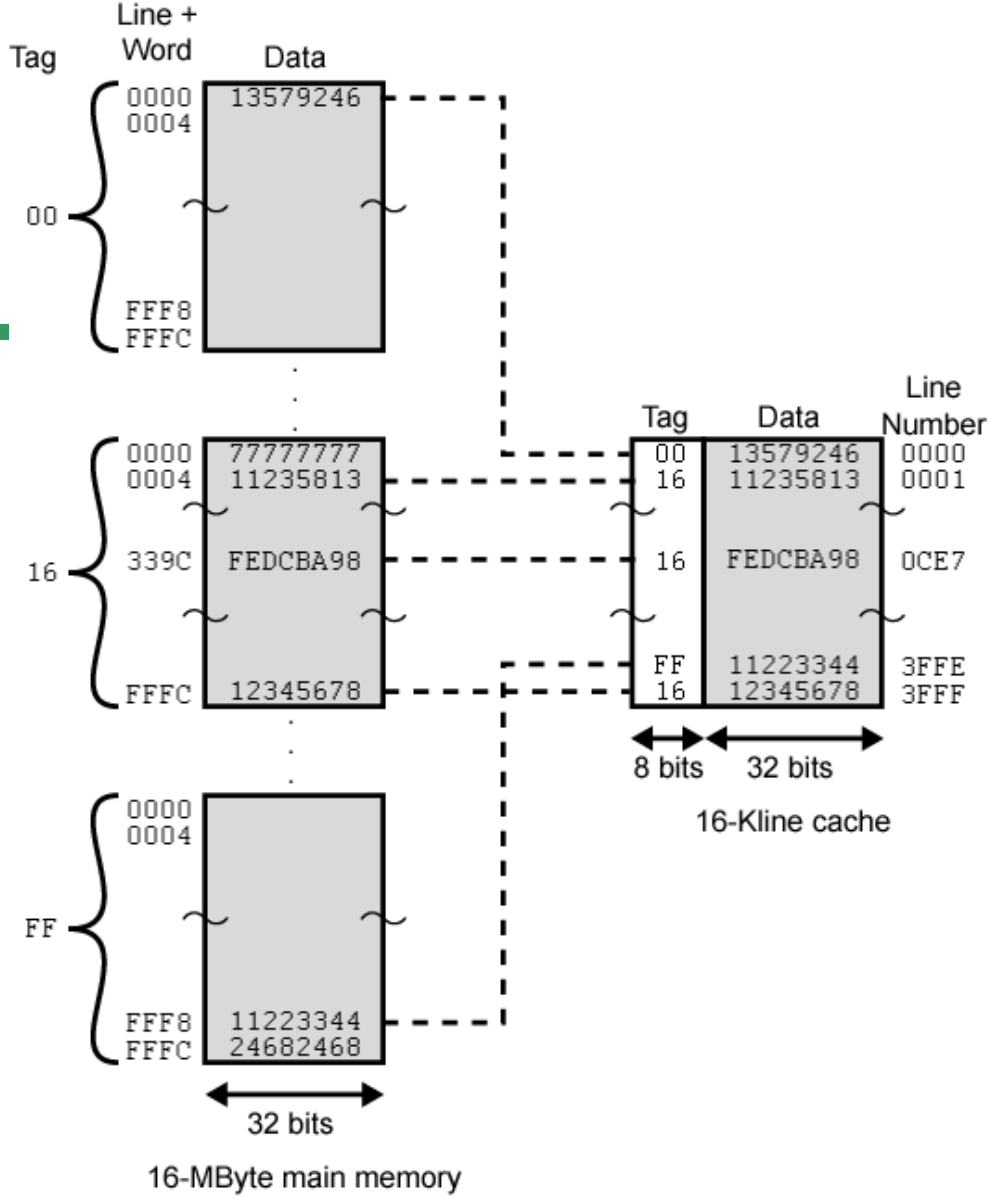
Struktura adresu (mapowanie bezpośrednie)

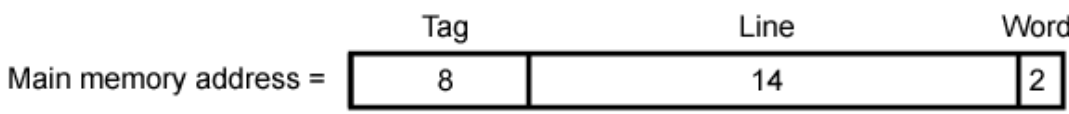
Znacznik s-r	Wiersz r	Słowo w
8	14	2

- Adres s+w=24 bitowy: 2^24 = 16MB
- Rozmiar cache'a: 64kB
  - cache mieści się w pamięci 16MB/64kB = 2^(s-r)=2^8 razy
- Rozmiar bloku w cache'u – 4B
  - w=2 bitowy identyfikator słowa (jeden z 4 bajtów)
- s=22 bitowy identyfikator bloku: 16MB/4B = 4M = 2^22
  - 8 bitowy znacznik (=s-r=22-14)
  - r=14 bitowy wiersz: funkcja mapująca modulo m=2^r=2^14

Przykład 1

Mapowanie bezpośrednie





Mapowanie bezpośrednie

- Długość adresu = (s + w) bitów
- Liczba adresowalnych jednostek = 2^(s+w) słów lub bajtów
- Rozmiar bloku = rozmiar wiersza = 2^w słów lub bajtów
- Liczba wierszy w cache'u = m = 2^r
- Liczba bloków w pamięci głównej = 2^(s+ w)/2^w = 2^s
- Rozmiar znacznika = (s – r) bitów

Mapowanie bezpośrednie za & przeciw

- Proste
- Tanie

- Dla danego bloku istnieje stała lokalizacja w cache'u
  - Co się stanie jeśli w krótkiej pętli są odwołania do 2 bloków pamięci mapowanych na ten sam wiersz?

Mapowanie skojarzeniowe

- mapowanie skojarzeniowe (ang. associative mapping)
- Blok pamięci może zostać załadowany do dowolnego wiersza w cache'u
- Adres dzielimy na dwie części: znacznik i słowo
  - znacznik jednoznacznie określa blok pamięci
- Aby stwierdzić czy blok jest w cache'u trzeba zbadać zgodność adresu ze znacznikiem każdego wiersza

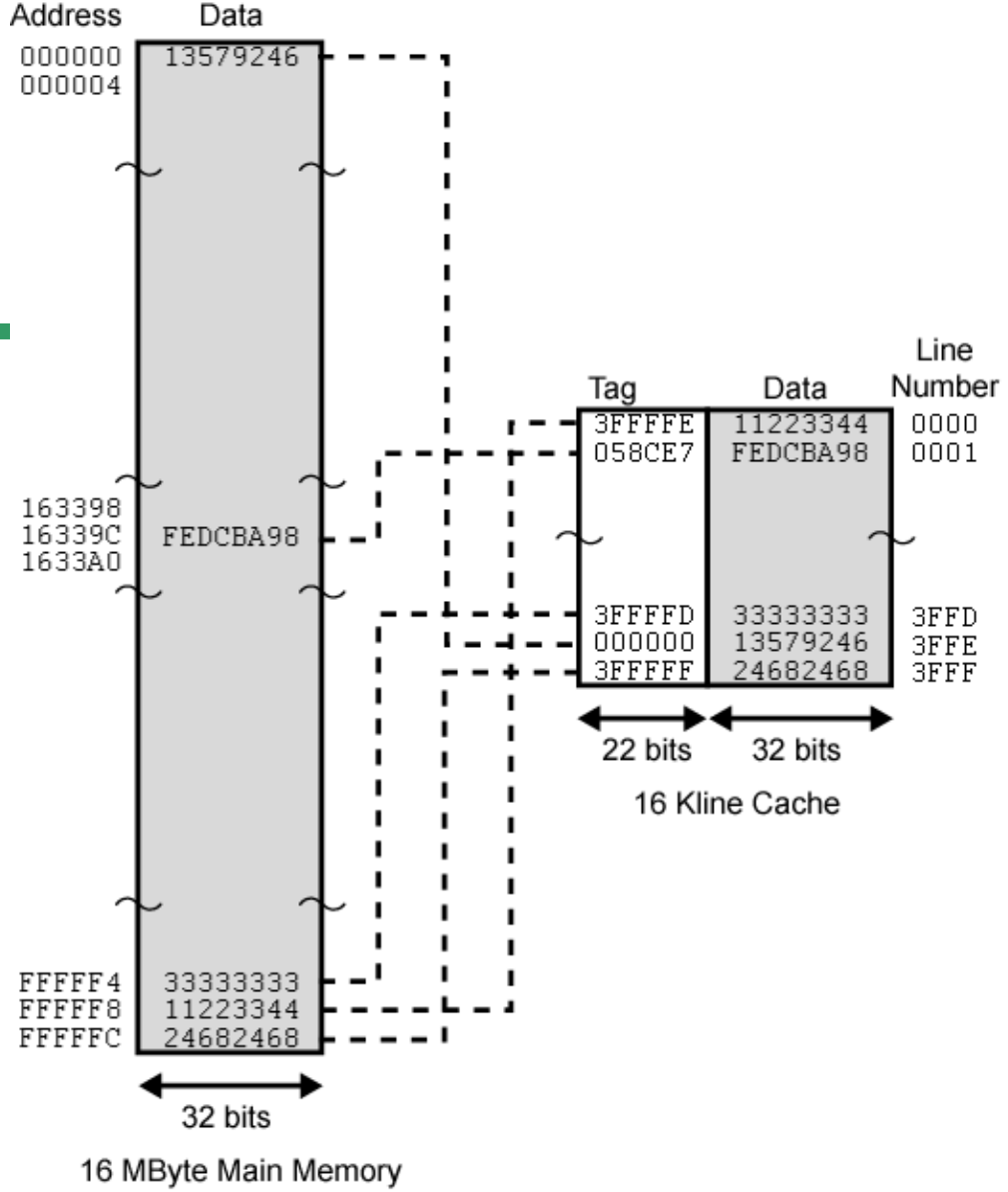
- Kosztowana metoda zwłaszcza gdy rozmiar cache'a jest duży
  - konieczność równoległego badania znaczników wszystkich wierszy w pamięci podręcznej (złożone układy)

Struktura adresu (mapowanie skojarzeniowe)

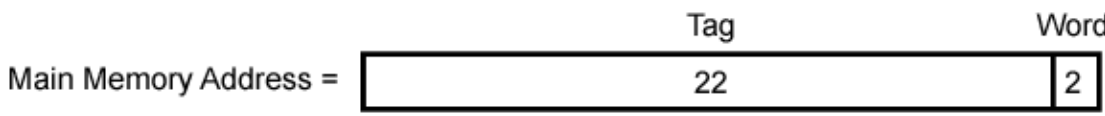
- 22 bitowy znacznik przechowywany wraz z każdym 32 bitowym (=4*8B) blokiem danych
- Porównanie pola znacznika adresu ze znacznikiem w cache'u dla stwierdzenia czy blok jest w cache'u
- Najmniej znaczące 2 bity adresu wyznaczają 16 bitowe słowo z 32 bitowego bloku danych
- np.

Adres	Znacznik	Dane	Wiersz
FFFFFC	3FFFFF	24682468	3FFF

Przykład 2



Mapowanie skojarzeniowe



Mapowanie skojarzeniowe

- Długość adresu = (s + w) bitów
- Liczba jednostek adresowalnych = 2^(s+w) słów lub bajtów
- Rozmiar bloku = rozmiar wiersza = 2^w słów lub bajtów
- Liczba bloków w pamięci = 2^(s+w)/2^w = 2^s
- Liczba wierszy w pamięci podręcznej = nieokreślona
- Rozmiar znacznika= s bitów

Mapowanie sekcyjno-skojarzeniowe

- k-drożne mapowanie sekcyjno-skojarzeniowe (ang. k-way set associative mapping)
- Cache jest dzielony na v sekcji (ang. sets)
- Każda sekcja składa się z k wierszy
- Dany blok B może zostać odwzorowany na dowolny wiersz jakiejś sekcji i
  - np. jeśli sekcja ma 2 wiersze
    - 2 sposoby mapowania (mapowanie dwudrożne)
    - blok może zostać umieszczony w jednym z dwu wierszy w jednej sekcji
  - np. jeśli adres sekcji jest 13 bitowy
    - określa się numer bloku w pamięci modulo 2^{13}
    - bloki 000000, 008000, 018000,...., FF8000 są mapowane na tę samą sekcję 0

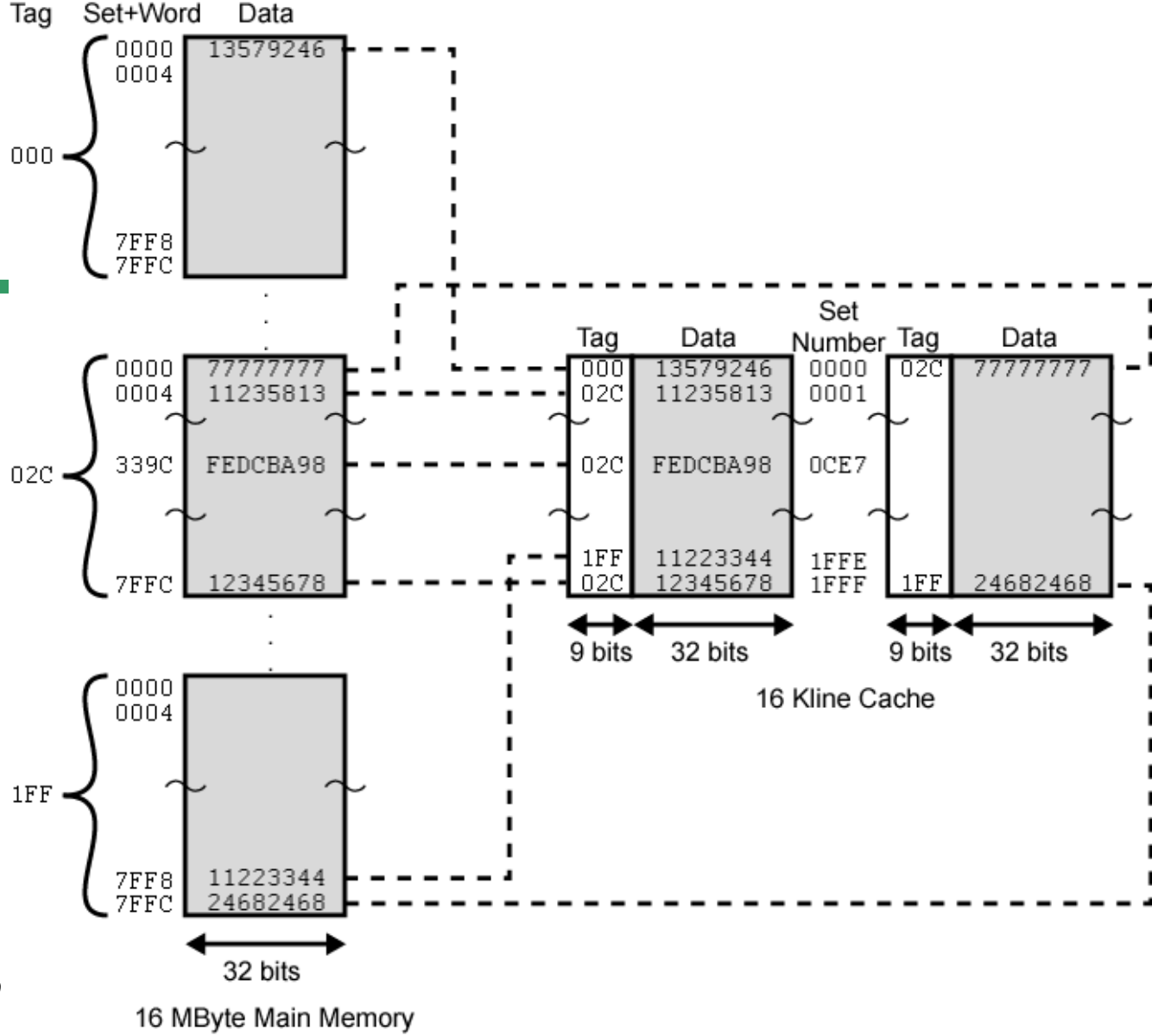
Struktura adresu (mapowanie sekcyjno-skojarzeniowe)

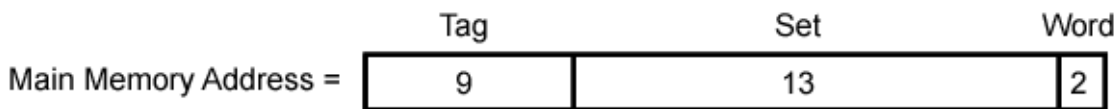
- mapowanie dwudrożne
- Za pomocą pola sekcji w adresie wybiera się sekcję w cache'u
- Następnie porównuje się pole znacznika aby określić czy blok jest cache'u
- Np.

Adres	Znacznik	Dane	Numer sekcji
167FFC	02C	12345678	1FFF
FFFFFC	1FF	24682468	1FFF

Przykład 3

Dwudrożne mapowanie sekcyjno-skojarzeniowe





k-drożne mapowanie sekcyjno-skojarzeniowe

- Długość adresu = (s + w) bitów
- Liczba jednostek adresowalnych = 2^(s+w) bajtów lub słów
- Rozmiar bloku = rozmiar wiersza = 2^w bajtów lub słów
- Liczba sekcji = v = 2^d
- Liczba bloków w pamięci = 2^(s+w)/2^w = 2^s
- Liczba wierszy w sekcji = k
- Liczba wierszy w cache'u = k*v = k * 2^d
- Rozmiar znacznika = (s – d) bitów

Pamięć podręczna (c.d.)

- algorytm zastępowania - określa, który blok należy zastąpić
  - mapowanie bezpośrednie - nie ma wyboru
  - mapowanie skojarzeniowe i sekcyjno-skojarzeniowe
    - least-recently-used (LRU) - zastąp najdawniej użyty
    - first-in-first-out (FIFO) - zastąp najstarszy
    - least-frequently-used (LFU) - zastąp najmniej używany
    - random - zastąp na „chybił-trafił” (tylko do testów porównawczych)
- polityka zapisywania (ang. write policy) - zmieniony blok w cache’u musi zostać zapisywany w pamięci głównej
  - zapis jednoczesny (ang. write through)- za każdym razem kiedy blok w cache’u jest zmieniany jest też zmieniany w pamięci
  - zapis opóźniony (ang. write back) - gdy jest blok jest zmieniany w cache’u ustawiany jest bit UPDATE skojarzony z danym wierszem i jeśli blok jest zastępowany to następuje aktualizacja pamięci
    - minimalizuje ilość operacji zapisywania (ok. 15% operacji dostępu do pamięci to zapis) ale pamięć główna może być nieaktualna (np. w połączeniu z DMA)

Pamięć podręczna - Intel

- 80386 – brak,
- 80486 - 8kB cache (16B wiersz, 4-drożne mapowanie sekcyjno-skojarzeniowe)
- Pentium - dwa L1 cache: 8kB - dane i 8kB - rozkazy
- Pentium III – L3 poza chipem
- Pentium 4
  - L1 cache: 8kB
    - 4-drożne mapowanie sekcyjno-skojarzeniowe
    - wiersz: 64B
  - L2 cache: 256kB
    - wsparcie dla L1 cache'a
    - 8-drożne mapowanie sekcyjno-skojarzeniowe
    - wiersz: 128B
  - L3 cache na chipie

Pamięć półprzewodnikowa

- RAM (ang. Random Access Memory)
  - nazwa myląca bowiem wszystkie pamięci półprzewodnikowe mają dostęp swobodny
  - odczyt/zapis
  - ulotna
  - sposób wymazywania: elektryczny
  - statyczna (ang. static) lub dynamiczna (ang. dynamic)
- DIMM (ang. dual in-line memory module) rodzaj pamięci RAM dla PC

Pamięć dynamiczna (DRAM)

- bity zapamiętane jak ładunki w kondensatorach
- rozładowywanie
- okresowe “odświeżanie ładunku”
- prosta budowa
- mała
- tania
- wolna
- asynchroniczna
- zastosowanie: pamięć główna

Przykład działania DRAM

- 16 Mb DRAM - 4 bity czytane i pisane jednocześnie
- 4M 4-bitowych słów
- tablica: 2048 wierszy i 2048 kolumn
- linie adresowe A0-A10 (2^11=2048)
- 11 sygnałów określa adres wiersza lub kolumny
  - zmniejszenie liczby pinów
  - dodatkowy sygnał
    - wyboru wiersza RAS (ang. row address select)
    - wyboru kolumny CAS (ang. column address select)
- DRAM zablokowany podczas odświeżania
  - wszystkie wiersze są okresowo odświeżane
- MUX pozwala na 4-krotne zwiększenie pamięci (jednoczesne użycie 4 układów)

Inne pamięci DRAM

- EDRAM (ang. Enhanced DRAM)- wzbogacona o pamięć podręczną statyczną RAM (16b)
- CDRAM (ang. Cache DRAM)- zawiera pamięć podręczną o 2Kb statycznej RAM
- SDRAM (ang. Synchronous DRAM) - pamięć synchroniczna, transmisja do/z CPU pod kontrolą zewnętrznego zegara systemowego; bardzo wydajna przy dużych transferach (np. multimedia)
  - DDR (ang. double-data-rate SDRAM) - ok. 3GBps
  - DDR2 (ang. quad-data-rate), XDR (ang. octal-data-rate)
  - http://www.simmtester.com/page/news/showpubnews.asp?num=109
- RDRAM (ang. Rambus DRAM) - do 3.5GBps
  - http://www.rambus.com/
  - http://www.reed-electronics.com/electronicnews/article/CA501992.html

Pamięć statyczna (SRAM)

- bity przechowywane za pomocą przerzutników
- nie wymaga odświeżania
- bardziej złożona budowa
- większa
- droższa
- szybsza
- zastosowanie: pamięć podręczna
- przykład: CMOS (ang. complementary metal-oxide-semiconductor) - zasilanie: bateria

Pamięć stała (ROM)

- ROM - ang. read-only memory
- trwały wzór danych
- zastosowanie:
  - mikroprogramowanie
  - tablice funkcji
  - programy systemowe (BIOS, ang. Basic Input Output System - niskopoziomowe we/wy)
  - podprogramy biblioteczne dla często wywoływanych funkcji

Rodzaje pamięci ROM (1)

- zapisywana w trakcie produkcji
  - bardzo droga dla małych serii
- pamięć programowalna (PROM, ang. programmable)
  - do zapisu (tylko raz) wymagane jest specjalne urządzenie
- pamięć głównie do odczytu (ang. read-mostly memory)
  - optycznie wymazywalna programowalna pamięć stała (ang. Erasable Programmable (EPROM))
    - wymazywanie : naświetlanie promieniowaniem ultrafioletowym z układu znajdującego się w obudowie

Rodzaje pamięci ROM (2)

- pamięć głównie do odczytu (c.d.)
  - elektrycznie wymazywalna programowalna pamięć stała (ang. Electrically Erasable Programmable (EEPROM))
    - zapisywane są tylko bajty zaadresowane
    - zapisywanie trwa dłużej niż odczyt (kilka mikrosekund/B)
    - mniej gęsto upakowana niż EPROM
  - pamięć błyskawiczna (ang. Flash memory, Flash EEPROM)
    - wymazywanie elektryczne
    - nie umożliwia wymazywania na poziomie bajtu
    - szybsza niż EPROM
    - tańsza niż EEPROM
    - zastosowanie: BIOS, Cisco IOS, PenDrive= USB flash memory drive (ok. 8GB)
      - http://en.wikipedia.org/wiki/Flash_memory

Rodzaje pamięci półprzewodnikowych

<table>
<tr><th>rodzaj pamięci</th><th>kategoria</th><th>wymazywanie</th><th>sposób zapisu</th><th>ulotność</th></tr>
<tr><td>Pamięć o dostępie swobodnym (RAM)</td><td>odczyt-zapis</td><td>elektrycznie, na poziomie bajtu</td><td>elektryczny</td><td>ulotna</td></tr>
<tr><td>Pamięć stała (ROM)</td><td rowspan="2">tylko odczyt</td><td rowspan="2">niemożliwe</td><td>maski</td><td rowspan="5">nieulotna</td></tr>
<tr><td>Programowalna pamięć stała (PROM)</td><td rowspan="4">elektryczny</td></tr>
<tr><td>Wymazywalna PROM (EPROM)</td><td rowspan="4">głównie odczyt</td><td>światłem UV, na poziomie mikroukładu</td></tr>
<tr><td>Pamięć błyskawiczna (flash)</td><td>elektrycznie, na poziomie bloku</td></tr>
<tr><td>Elektrycznie wymazywalna PROM (EEPROM)</td><td>elektrycznie, na poziomie bajtu</td></tr>
</table>

Literatura

- W. Stallings - Architektura i organizacja systemu komputerowego, WNT, 2000, 2003, 2004
  - http://williamstallings.com/COA5e.html
  - http://WilliamStallings.com/COA6e.html
  - http://williamstallings.com/COA/COA7e.html
- A. S. Tanenbaum, Strukturalna organizacja systemów komputerowych, Helion, 2006
- Wikipedia