

*Krzysztof Szafran*

# Systemy Operacyjne

## Literatura podstawowa:

- *A. Silberschatz, P. Galvin*: Podstawy Systemów Operacyjnych, wydanie trzecie, zmienione, rozszerzone, WNT 2000,

## SYSTEM KOMPUTEROWY

### SYSTEM OPERACYJNY

- Eksploatuje zasoby sprzętowe, w tym co najmniej jeden procesor.
- Zapewnia użytkownikowi zbiór usług systemowych.
- Zarządza w imieniu użytkowników pamięcią pomocniczą i urządzeniami wejścia/wyjścia.

### PODSTAWOWE ELEMENTY

- Procesor(y).
- Pamięć główna.
- Moduły wejścia/wyjścia:
  - urządzenia pamięci pomocniczej,
  - wyposażenie komunikacyjne,
  - terminale.
- Szyna systemowa (Bus) łącząca pozostałe elementy systemu.

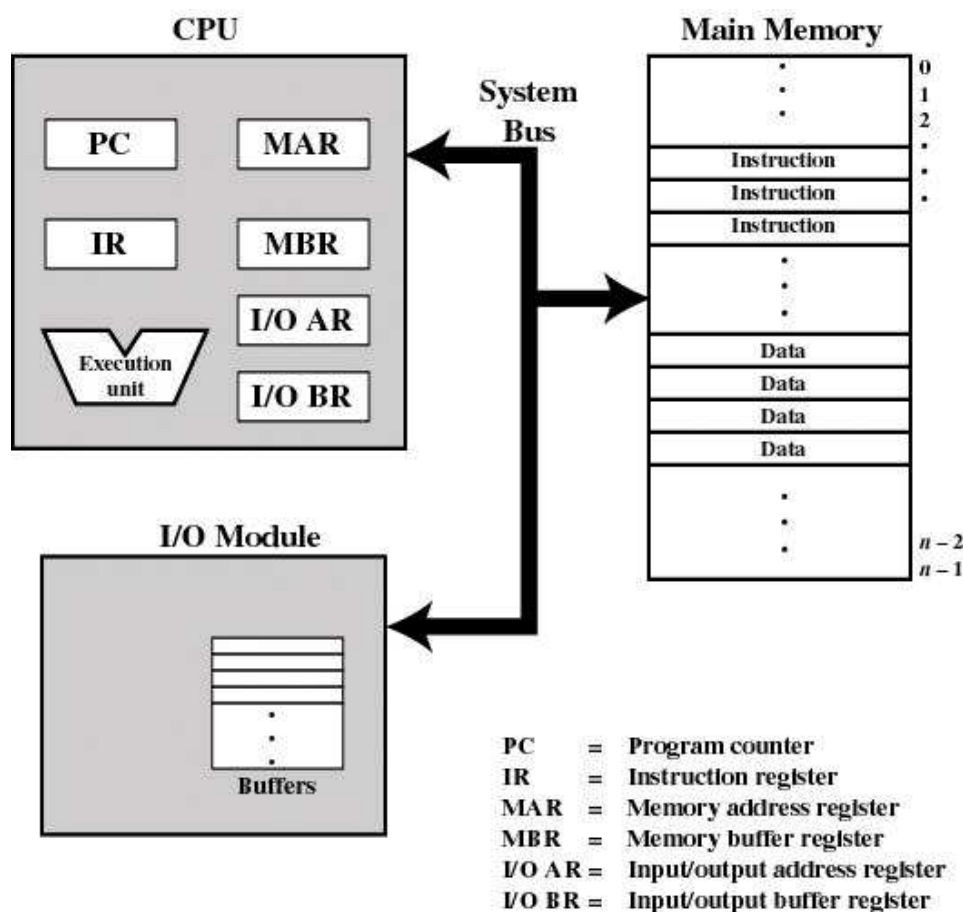


Figure 1.1 Computer Components: Top-Level View

## REJESTRY PROCESORA

- Dostępne dla użytkownika — pozwalają programiście/kompilatorowi na optymalizację odwołań do pamięci głównej.
- Rejestry kontrolne i rejestry stanu — używane przez procesor do kontroli jego pracy i przez system operacyjny do kontroli wykonywanych programów.

## REJESTRY DOSTĘPNE DLA UŻYTKOWNIKA

Dostępne dla wszystkich programów — aplikacji i programów systemowych z poziomu języka maszynowego i/lub assemblera.

Podstawowe typy:

- rejestry danych,
- rejestry adresowe,
  - indeksowy — umożliwia dodawanie indeksu do wartości bazowej w celu uzyskania adresu ostatecznego,
  - wskaźnik segmentu — przy segmentacji pamięci odwołanie realizowane jest jako adres segmentu i offset,
  - wskaźnik stosu — wskazuje wierzchołek stosu.

## REJESTRY KONTROLNE I STANU

Częściowo dostępne dla użytkownika zawierają kody kontrolne (flagi) ustawiane przez procesor w wyniku wykonania operacji.

Typowe przykłady:

- licznik rozkazów (PC, Program Counter) — zawiera adres kolejnej instrukcji do wykonania,
- IR — zawiera ostatnio pobraną instrukcję,
- słowo stanu programu (PSW):
  - kody stanu — mogą być sprawdzane przez program ale nie zmieniane, np.: wynik dodatni, wynik ujemny, zero,
  - informacja o zamaskowaniu przerwań,
  - tryb pracy procesora (zwykły/uprzywilejowany)

- wskaźnik stosu systemowego,
- rejestr przerw,
- inne.

## WYKONANIE INSTRUKCJI

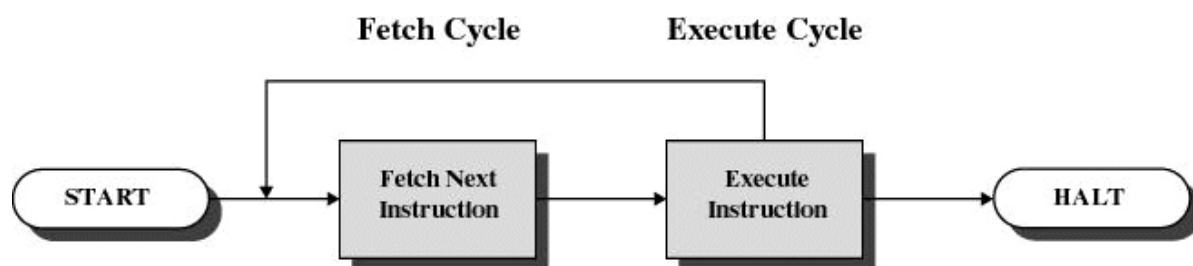


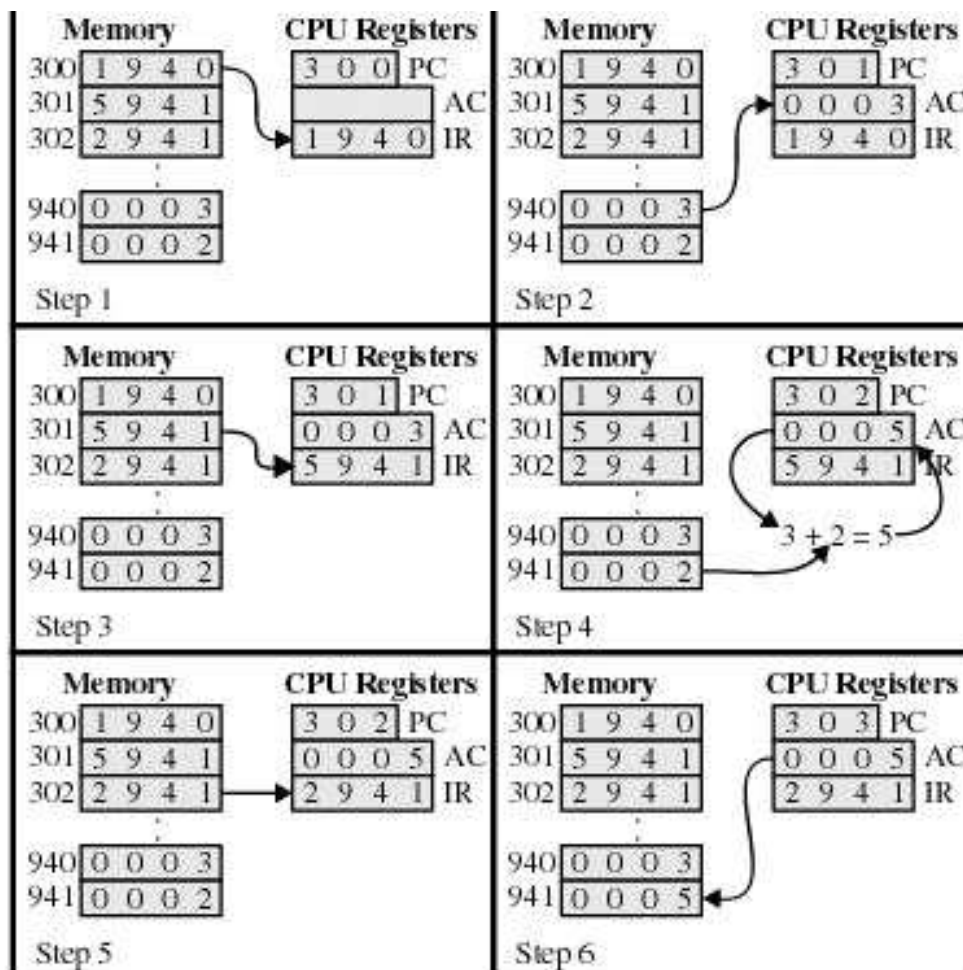
Figure 1.2 Basic Instruction Cycle

1. Instrukcja jest pobierana z pamięci, z miejsca wskazywanego przez PC i umieszczana w rejestrze instrukcji (IR).
2. Licznik rozkazów jest modyfikowany tak, aby wskazywał następną instrukcję programu.
3. Instrukcja jest wykonywana.

Podstawowe typy instrukcji to:

- transmisja pamięć-procesor (i odwrotnie),
- transmisja procesor-bufor we/wy (i odwrotnie),
- przetwarzanie danych — operacje logiczne i arytmetyczne,
- kontrola przepływu sterowania.

Poniższy fragment kodu realizuje (w trzech krokach) operację dodania zawartości adresu 940 do zawartości adresu 941 i umieszczenie wyniku pod adresem 941.



**Figure 1.4 Example of Program Execution**  
(contents of memory and registers in hexadecimal)

## PRZERWANIA

- Przerwanie normalnej kolejności wykonywania programu.
- Zwiększenie efektywności przetwarzania.

- umożliwienie procesorowi wykonywania innych instrukcji w czasie realizacji operacji wejścia/wyjścia.
- Zawieszenie programu przez zdarzenie zewnętrzne dla niego i postępowanie w taki sposób, aby możliwe było wznowienie wykonania zawieszanego programu.

Podstawowe klasy przerwań:

- Programowe:
  - przekroczenie zakresu (overflow),
  - dzielenie przez zero,
  - wykonanie nielegalnej instrukcji,
  - odwołanie się do adresu spoza przestrzeni adresowej procesu.
- Zegarowe.
- Wejścia/wyjścia.
- Awarie sprzętu.

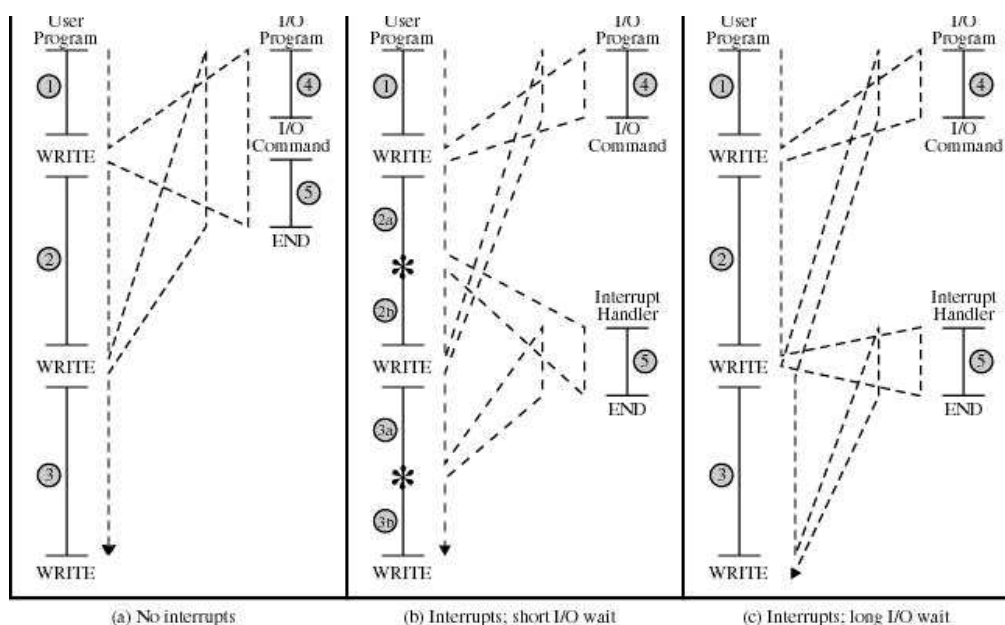


Figure 1.5 Program Flow of Control Without and With Interrupts

Handler przerwań — część systemu, która odpowiada za identyfikację rodzaju przerwania i podjęcie odpowiedniej akcji. Obecnie najczęściej sprzęt realizuje identyfikację przerwania (rejestr przerwań) i przekazuje sterowanie do odpowiedniej procedury obsługi, oddzielnej dla każdego typu przerwania.

Zmodyfikowany cykl realizacji instrukcji:

- Po zakończeniu wykonywania bieżącej instrukcji, jeśli przerwania nie są zamaskowane, następuje sprawdzenie, czy nie wystąpiło przerwanie.
- Jeśli tak, przechodzimy do obsługi przerwania.
- Jeśli nie, pobieramy kolejną instrukcję.

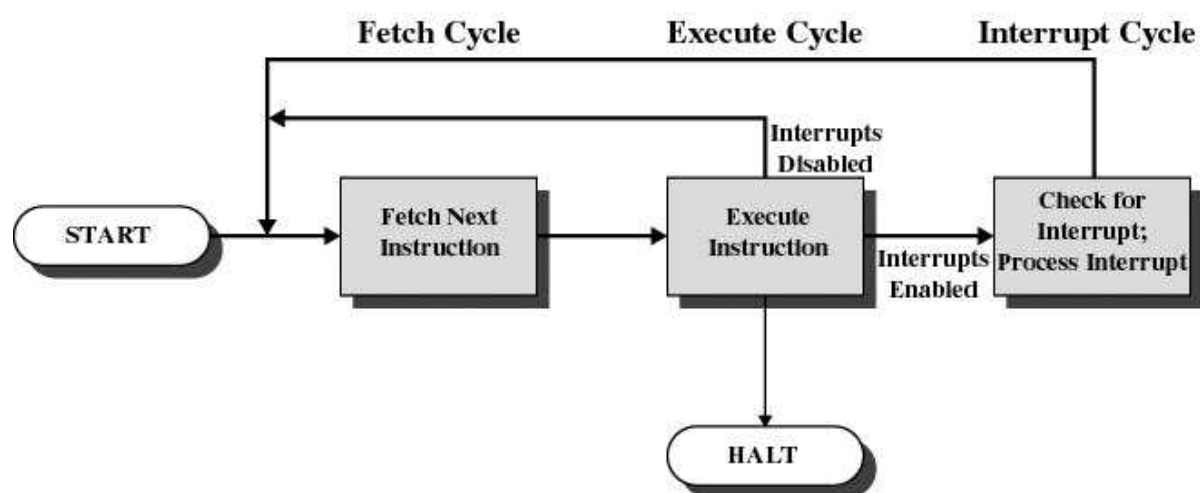


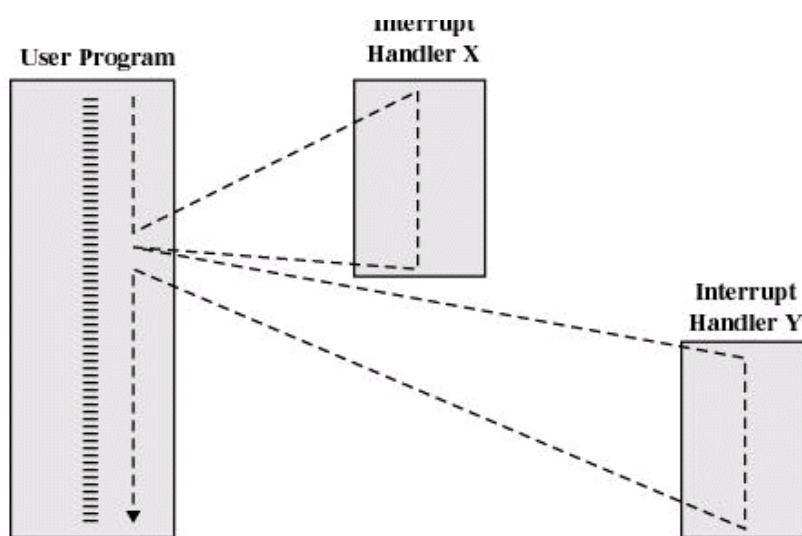
Figure 1.7 Instruction Cycle with Interrupts

## WIELOKROTNE PRZERWANIA

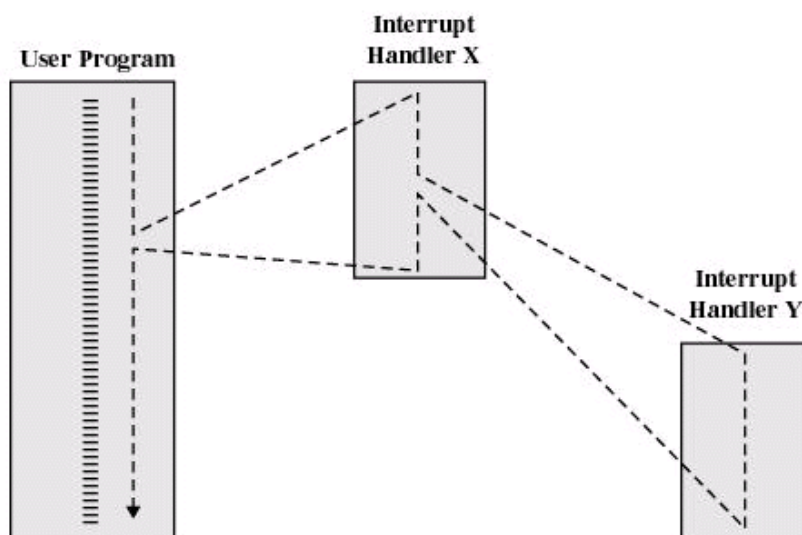
Jak postępować z przerwaniem, które pojawiają się w czasie obsługi innego przerwania ?

- Ignorować pojawianie się przerwań w trakcie obsługi przerwania.

- Zbierać informacje o dodatkowych przerwaniach i po zakończeniu obsługi jednego sprawdzić, czy inne przerwanie nie czeka na obsłużenie.
- Priorytety przerw z możliwością przerywania obsługi przerwa w przypadku pojawienia się ważniejszego.



(a) Sequential Interrupt processing



(b) Nested Interrupt processing

Figure 1.12 Transfer of Control with Multiple Interrupts



## **OBSŁUGA PRZERWANIA**

1. Urządzenie wysyła sygnał przerwania
2. Procesor kończy bieżącą instrukcję.
3. Procesor wykrywa przerwanie.
4. Zapamiętuje bieżący kontekst obliczeń (stos systemowy).
5. Do licznika rozkazów wstawia adres początkowy procedury obsługi przerwania.
6. Realizacja obsługi przerwania.
7. Zdjęcie ze stosu kontekstu przerwanego procesu (wraz z zawartością PC).

## **WIELOPROGRAMOWOŚĆ**

- Procesor realizuje jednocześnie więcej niż jeden program.
- Kolejność realizacji zależy od względnych priorytetów i ewentualnego oczekiwania na we/wy.
- Po zakończeniu obsługi przerwania sterowanie może zostać przekazane do innego programu niż poprzednio wykonywany.

## HIERARCHIA PAMIĘCI

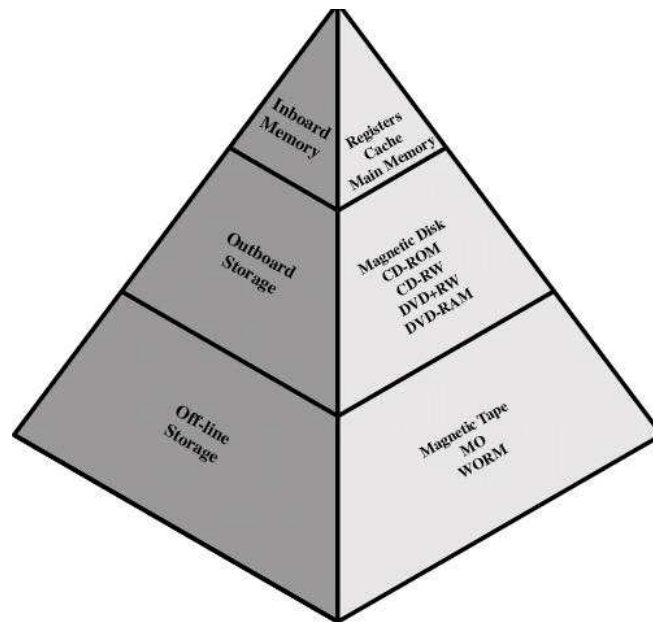


Figure 1.14 The Memory Hierarchy

Patrząc od góry do dołu:

- Malejący koszt jednego bitu.
- Wzrastająca pojemność
- Rosnący czas dostępu.
- Malejąca częstość odwołań procesora.

Średni czas dostępu zależy w pamięci wielopoziomowej — współczynnik trafień.

## PAMIĘĆ PODRĘCZNA DYSKU

- Część pamięci głównej przeznaczona jest na bufor czasowo przechowujące porcje (bloki) danych z dysku.
- Zamiast czytać z wolnego dysku odwołujemy się do szybkiego bufora.

- Operacje zapisu na dysku są grupowane.

## **PODRĘCZNA PAMIĘĆ PROCESORA**

- Niewidoczna dla systemu operacyjnego.
- Przyspiesza dostęp do pamięci (procesor jest szybszy niż pamięć).
- Zawiera część zawartości pamięci.
- Proces najpierw odwołuje się do pamięci podręcznej.
- Jeśli cache nie zawiera potrzebnych danych, kopiowany jest blok zawierający odpowiedni fragment.

## **CHARAKTERYSTYKA PAMIĘCI PODRĘCZNEJ**

Dla funkcjonowania pamięci podręcznej istotne są (między innymi):

- Wielkość pamięci podręcznej — mała wyraźnie pogarsza efektywność.
- Wielkość blok.
- Funkcja mapująca — określa adres, gdzie blok zostanie zapisany.
- Algorytm wymiany — które bloki usuwać w razie potrzeby, LRU.
- Polityka zapisu — jak często i kiedy uaktualniać pamięć główną (kompromis między liczbą operacji a aktualnością zawartości).

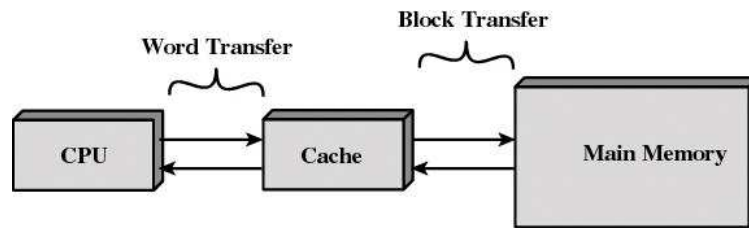


Figure 1.16 Cache and Main Memory

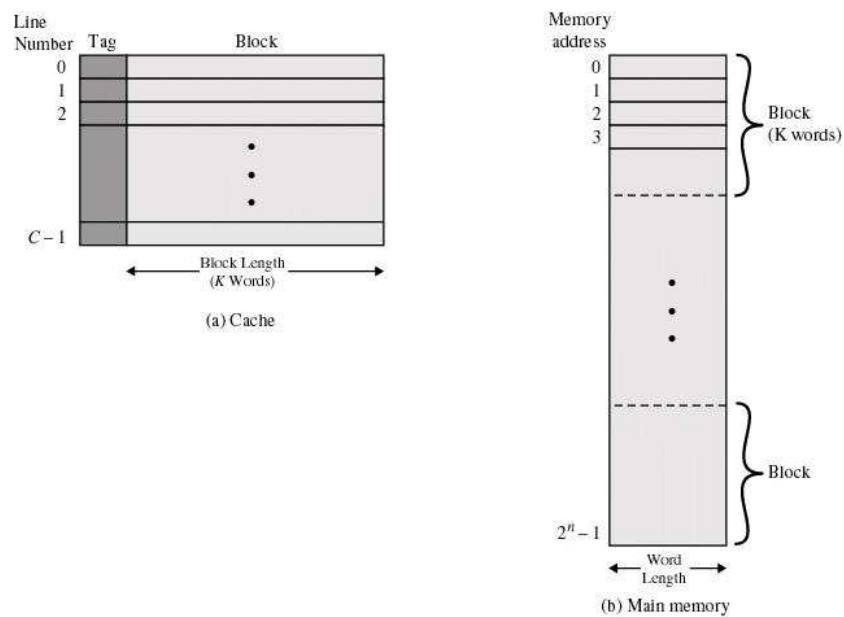


Figure 1.17 Cache/Main-Memory Structure

## TECHNIKI REALIZACJI WE/WY

### PROGRAMOWANE WE/WY

- Operację wykonuje moduł we/wy, nie procesor.
- Ustawia odpowiednie bity w rejestrze stanu we/wy.
- Nie ma przerwań.
- Procesor sprawdza odpowiednie bity aż do zakończenia operacji.

## WE/WY STEROWANE PRZERWANIAMI

- Przerwanie sygnalizuje procesorowi, że dane są w buforze.
- W czasie pobierania danych do bufora procesor może robić coś innego.
- Angażuje procesor, który musi zlecić transmisję każdego słowa.

## DMA — BEZPOŚREDNI DOSTĘP

- Dane transmitowane są blokami bezpośrednio do/z pamięci (specjalny układ DMA).
- Przerwanie pojawia się w momencie zakończenia transmisji.
- Procesor jest zaangażowany tylko na początku i końcu operacji.

