Architektura komputera typu PC z procesorem IA-32

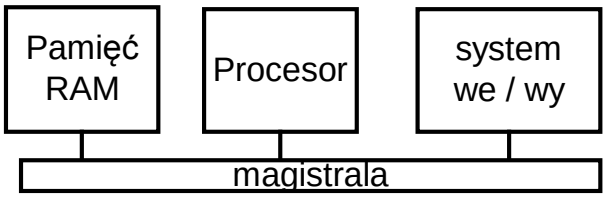
1. Ogólna struktura systemu jednoprocesorowego

Już systemy jednoprocesorowe mogą być środowiskiem, w którym wykonywane jest wiele programów współbieżnych.

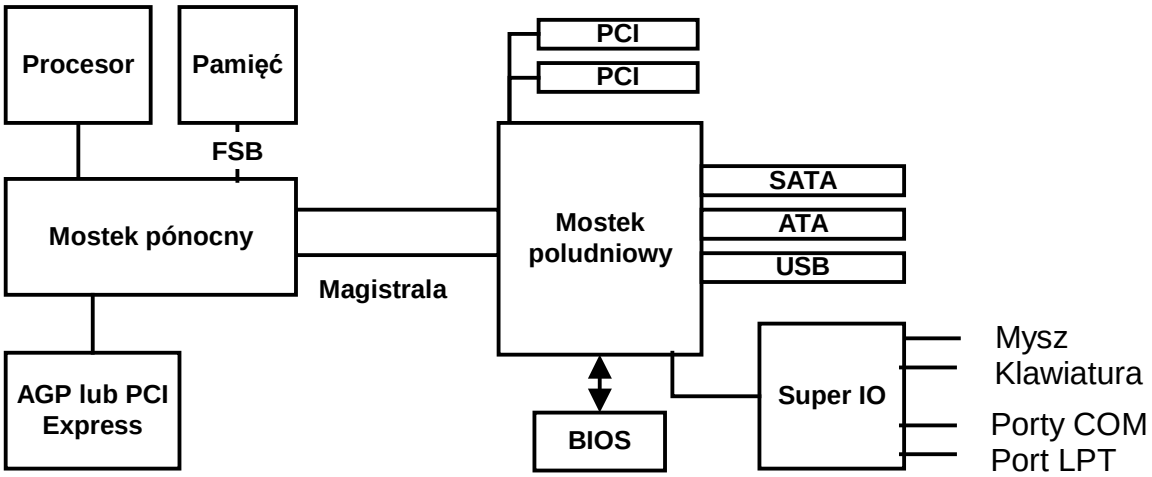
Omówione na przykładzie komputera PC i architektury IA-32 - (80386, 80486 i Pentium).

Aby program mógł się wykonywać i komunikować z otoczeniem niezbędny jest:

- procesor
- pamięć
- system wejścia wyjścia.



Rys. 1-1 Struktura systemu jednoprocesorowego



Rys. 1-2 Architektura komputera PC

Szyny	Szyny mostka północnego	Szyny mostka południowego
Rodzaje	FSB, RAM, AGP, PCI Express X16, CSA	ISA, PCI, PCI Express, USB, ATA, SCSI, FireWire
Łączą	CPU, RAM, Wideo, Ethernet	Wszystkie inne urządzenia.
Częstotliwość zegara	66 - 1066 MHz	Typowo 10-33 MHz.
Przepustowość	> 3 GB/sek	Typowo 20-500 MB/sec. na magistralę

Tabela 1-1 Charakterystyka szyn komputera PC

Procesor

Procesor wykonując zawarty w pamięci operacyjnej program i przekształca zawarte tam dane. Procesory rodziny IA-32 posiadają trzy tryby pracy:

- tryb rzeczywisty,
- tryb chroniony
- tryb zarządzania systemem.

Właściwym trybem normalnej pracy systemu jest tryb chroniony.

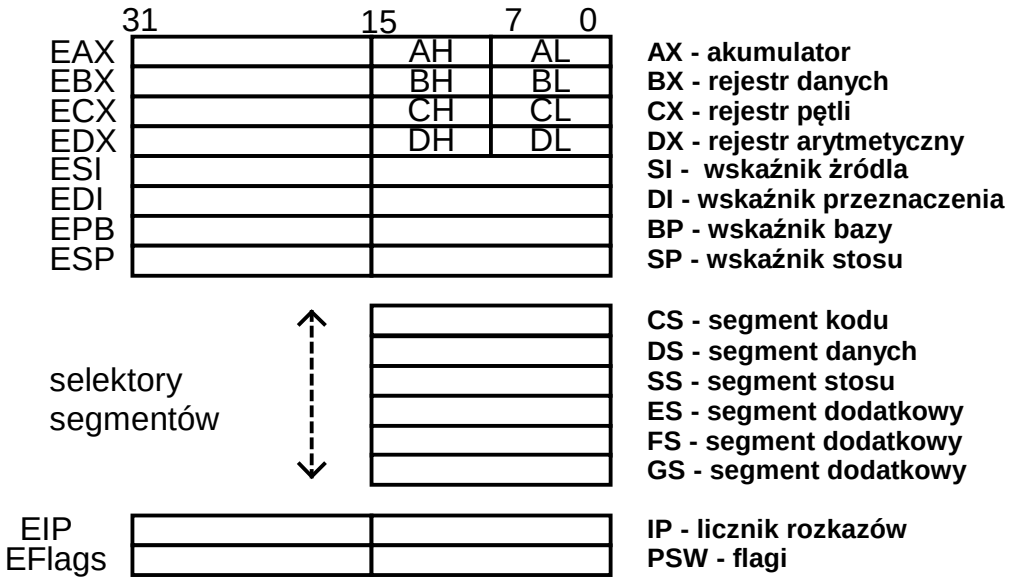
Najważniejsze mechanizmy sprzętowe wspierające wieloprogramowość:

Wsparcie ochrony programów i systemu operacyjnego:

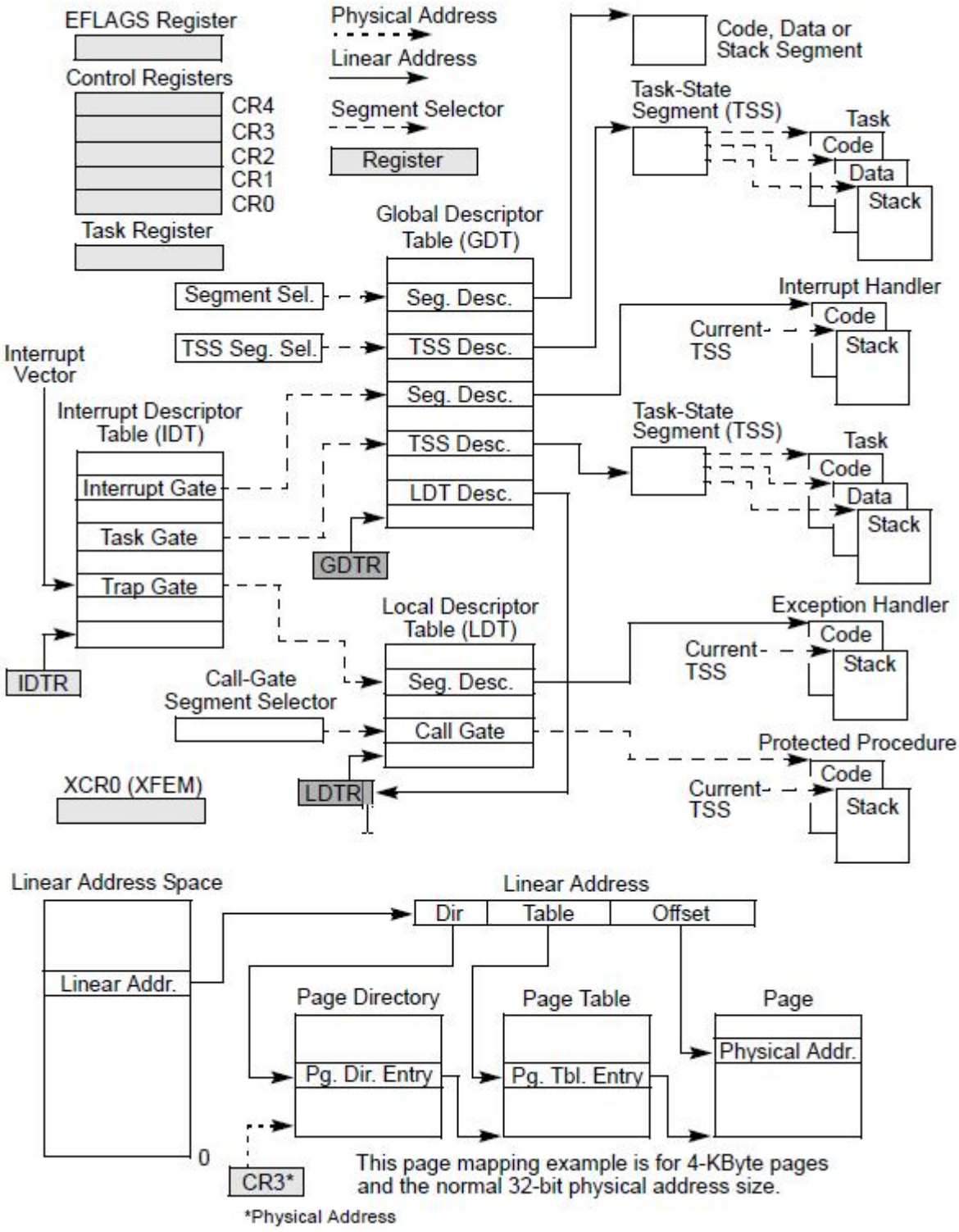
- segmentacja
- stronicowanie pamięci
- poziomy ochrony procesora.

Wsparcie przełączania procesów i współbieżnego we/wy:

- system przerwań i wyjątków
- autonomiczny system wejścia wyjścia



Rys. 1-3 Rejestry procesora w architekturze IA-32 dla trybu chronionego



Rysunek z Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 System Programming Guide

2. Ochrona pamięci

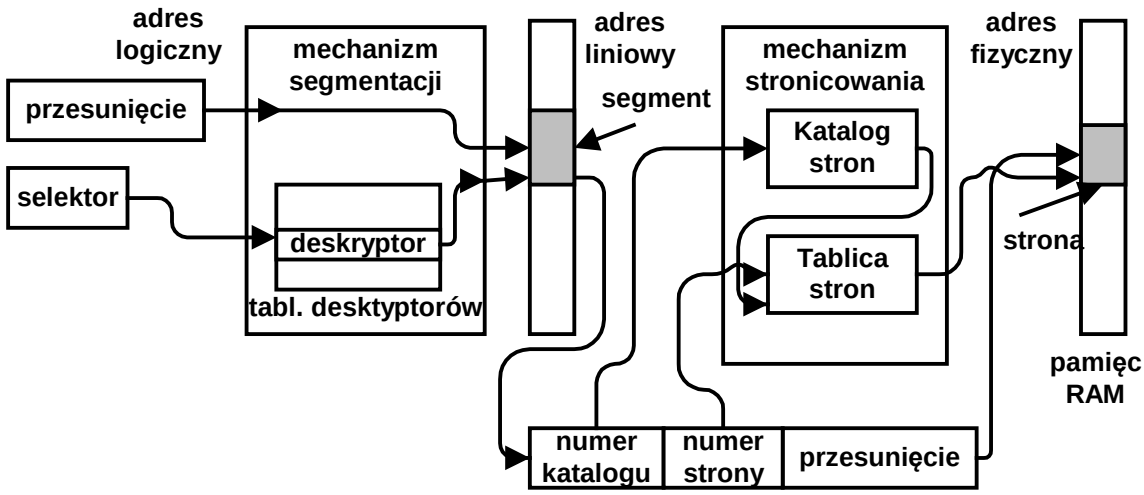
Pamięć operacyjna - jednowymiarowa tablica bajtów.
Maksymalny wymiar pamięci dla procesorów 32 bitowych - 4 GB.

Procesory mogą wykonywać wiele procesów w trybie podziału czasu procesora.

Możliwa jest sytuacja gdy wystąpi błąd w programie – na skutek tego proces dokonuje modyfikacji danych należących do innego procesu lub systemu operacyjnego.

Mechanizmem stosowanym do wzajemnego odizolowania procesów jest mechanizm segmentacji. Umożliwia on również relokację.

W architekturze IA-32 dostępna jest tak segmentacja jak i stronicowanie - mechanizm stronicowania może być wyłączony.



Rys. 2-1 Mechanizm zarządzania pamięcią w procesorach IA-32

2.1. Segmentacja

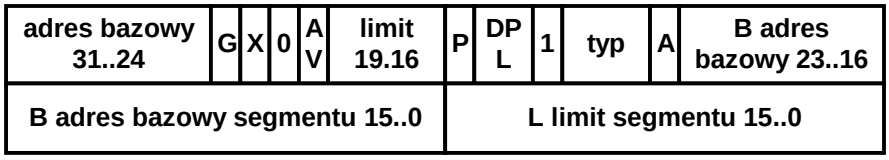
Segmentacja - mechanizmem sprzętowym polegającym na podziale pamięci operacyjnej na ciągłe bloki nazywane segmentami.
W procesorach IA-32 pamięć logiczna jest dwuwymiarowa. Adres składa się z:

- selektora (ang. *selektor*) - określa segment pamięci
- przesunięcia (ang. *offset*) - wyznacza adres wewnątrz segmentu.

Każdy segment charakteryzuje się takimi parametrami

- początek bloku,
- wielkość
- atrybuty

Parametry segmentu przechowywane są w 8 bajtowym rekordzie nazywanym deskryptorem segmentu.



Rys. 2-1 Deskryptor segmentu

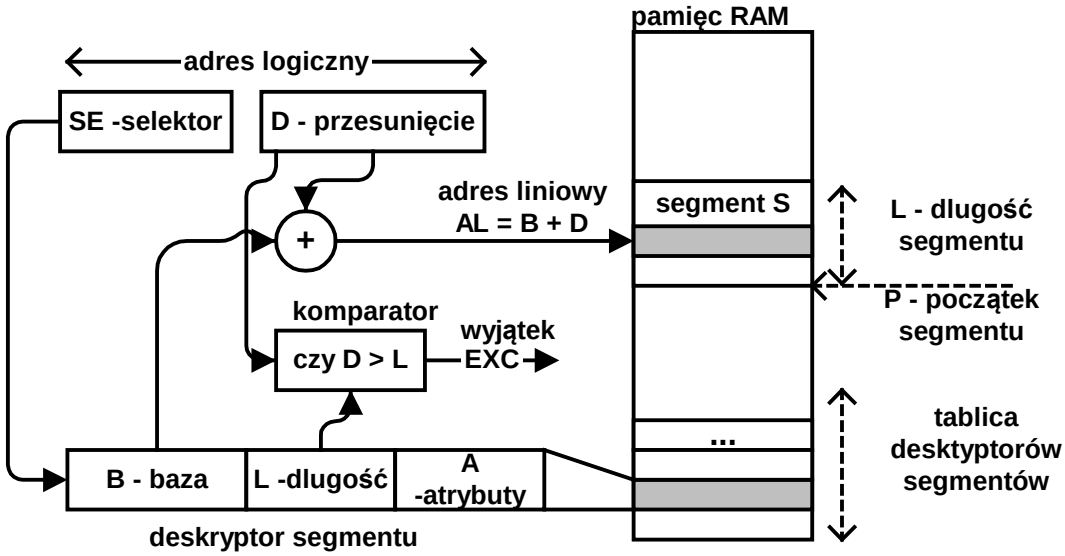
B - adres bazowy segmentu
L - długość segmentu
G - sposób interpretacji limitu segmentu (0 – bajty, 1 – strony 4KB),
DPL - poziom uprzywilejowania segmentu,
P - bit obecności segmentu (używany w pamięci wirtualnej),
AV - nie używany,
A - mówi czy deskryptor jest używany.

Deskryptory są przechowywane w dwóch rodzajach tablic:

- globalnej tablicy deskryptorów GDT (ang. *Global Descriptor Table*)
- lokalnej tablicy deskryptorów LDT (ang. *Local Descriptor Table*).

W systemie istnieje:

- jedna tablica GDT - opisuje segmenty widoczne dla wszystkich procesów
- wiele lokalnych tablic deskryptorów LDT (ang. *Local Descriptor Table*), opisujących prywatne segmenty procesów



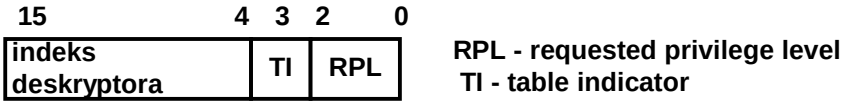
Rys. 2-2 Uproszczony schemat mechanizmu segmentacji

Adres logiczny składa się z:

- selektora segmentu SE
- przesunięcia D.

Funkcje selektora pełni jeden z rejestrów segmentowych:

- dla kodu selektorem jest rejestr DS.,
- dla danych rejestr DS.,
- dla stosu SS (BP).



Rys. 2-3 Zawartość rejestru selektora segmentu

Selektor zawiera:

- indeks deskryptora - położenie segmentu znajdującego się w tablicy deskryptorów,
- TI - określa o którą tablicę chodzi (0 – GDT, 1 - LDT)
- RPL - żądany poziom uprzywilejowania – określa poziom uprzywilejowania procesu.

Adres liniowy - suma pobieranego z pola adresowego rozkazu przesunięcia D i adresu początku segmentu B pobieranego z deskryptora.
Komparator sprawdza czy przesunięcie D nie wykracza poza długość segmentu L zapisanego w deskryptorze. Gdy tak się zdarzy generowany jest wyjątek EXC który powoduje wywołanie systemu operacyjnego. System operacyjny podejmuje decyzję, co zrobić z naruszającym przydzielony segment procesem.

Adres liniowy może być poddany przetwarzaniu przez mechanizm stronicowania.

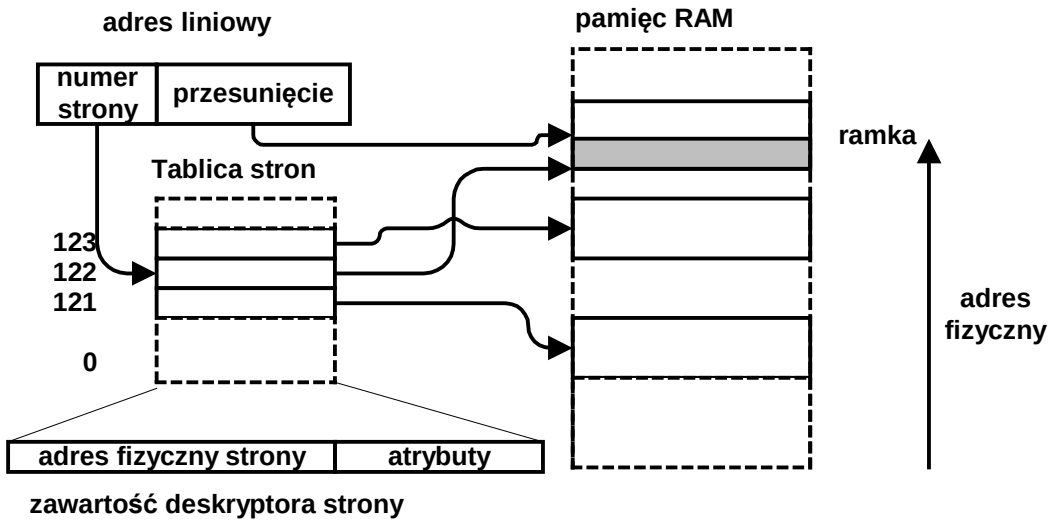
2.2. Stronicowanie

Mechanizm stronicowania rozwiązuje problem wzajemnej ochrony pamięci procesów i systemu operacyjnego przed nadpisaniem. Nie rozwiązuje jednak problemu:

- Fragmentacji pamięci – przydzielane i zwalniane segmenty powodują poszatkowanie pamięci, trudno znaleźć duży spójny obszar. Jest to tak zwana fragmentacja zewnętrzna.
- Problemu braku pamięci fizycznej.

Problemy te mogą być rozwiązane przez stronicowanie. Mechanizm ten polega na podziale pamięci liniowej na obszary o jednakowej wielkości zwane stronami (ang. *pages*). Wielkość strony to obecnie od 4 KB do 4MB.

W celu odróżnienia stron z obrazem procesu od stron pamięci fizycznej, strony fizyczne nazywa się ramkami (ang. frames). Pamięć procesu tworzą kolejne strony, natomiast ramki mogą być porozrzucane gdziekolwiek w pamięci. Ich fizyczna lokalizacja zawarta jest w tablicy stron.



Rys. 2-2 Schemat stronicowania

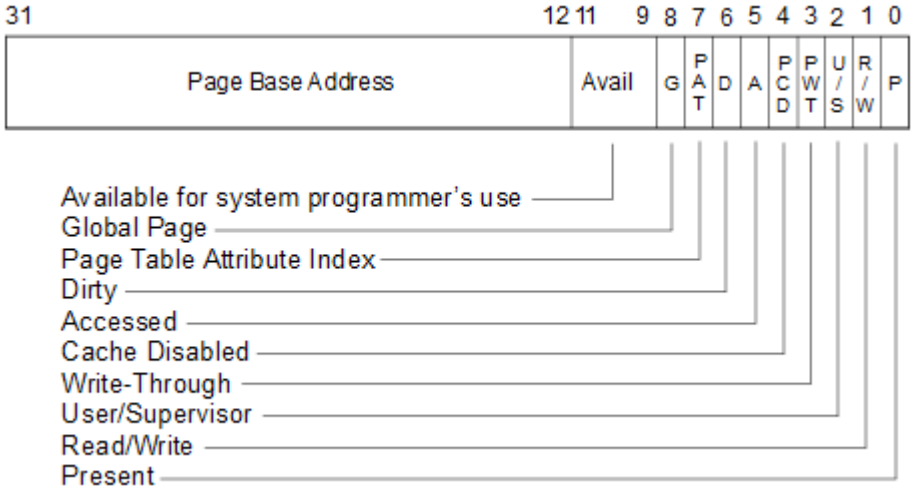
Na powyższym rysunku proces zawiera strony 121,122,123 które w adresie liniowym są po kolei, natomiast w pamięci fizycznej mogą być umieszczone gdziekolwiek.

Tablica stron

Tablica stron składa się z 32 bitowych słów, każde z nich opisuje pojedynczą stronę.

Opis strony zawiera:

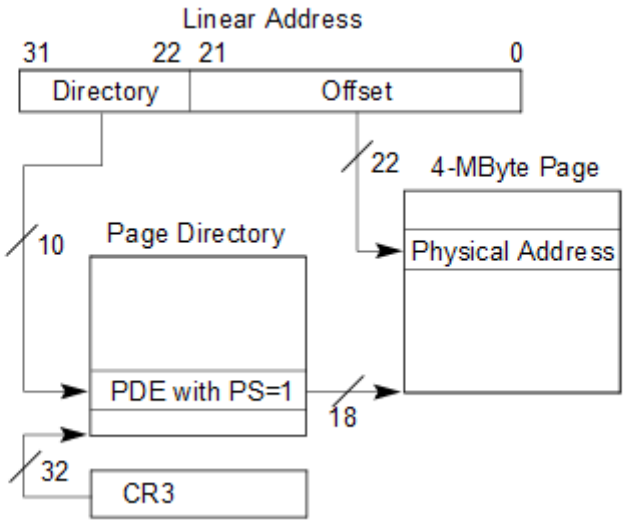
- Adres fizyczny strony
- Atrybuty strony: bit obecności, bit dostępu, możliwość zapisu, przywileje administratora, czy był zapis.



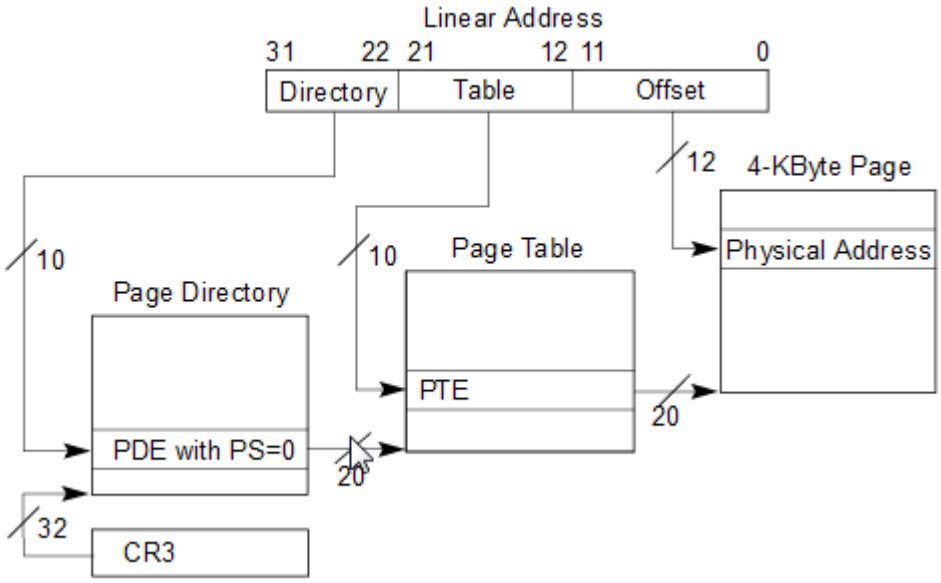
Rys. 2-3 Zawartość deskryptora strony (przykład)

W architekturze IA-32 stosuje się dwa schematy stronicowania:

- Stronicowanie jednopoziomowe (strona ma 4MB)
- Stronicowania dwupoziomowe (strona 4 KB)



Rys. 2-4 Stronicowanie jednopoziomowe, strona 4 MB



Rys. 2-5 Stronicowanie dwupoziomowe, strona 4 KB

Stronicowanie pozwala na rozwiązanie problemu fragmentacji pamięci kosztem dodatkowego obszaru pamięci potrzebnego na tablicę stron. Stronicowanie 2 poziomowe pozwala na oszczędność tej pamięci.

Co tracimy:

- Zwiększony czas dostępu do komórki pamięci (adresacja pośrednia)
- Tablica stron zajmuje pamięć
- Fragmentacja wewnętrzna

2.3. Porównanie

Segmentacja	Ochrona obszarów pamięci używanych przez procesy przed dostępem przez inny proces
	Zapewnienie przemieszczalności programów
Stronicowanie	Realizacja pamięci wirtualnej większej niż fizyczna
	Rozwiązanie problemu fragmentacji
	Ochrona obszarów pamięci

Tab. 2-1 Mechanizmy sprzętowe zarządzania pamięcią

3. Ochrona procesora

Aby system operacyjny mógł wykonywać swe funkcje powinien mieć on dostęp do wszystkich istotnych zasobów procesora, mechanizmu zarządzania pamięcią, kontrolera przerwań i kontrolerów wejścia wyjścia.

Procesy aplikacyjne nie mogą mieć dostępu do tego typu zasobów, gdyż czy to na skutek błędów czy intencjonalnie mogłyby zdestabilizować pracę systemu.

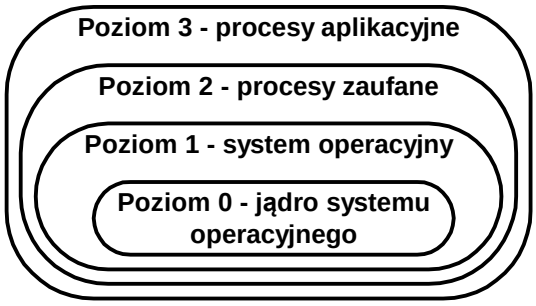
We współczesnych mikroprocesorach wprowadza się dwa (lub więcej) tryby pracy procesora:

- tryb użytkownika (ang. *User Mode*)
- tryb systemowy (ang. *System Mode*).

- Tryb systemowy - proces może wykonywać wszystkie instrukcje procesora, sięgać do wszystkich obszarów pamięci i przestrzeni wejścia wyjścia.
- Tryb użytkownika - nie jest dozwolony dostęp do rejestrów: związanych z zarządzaniem pamięcią, obsługą przerwań zarządzaniem pracą procesora.

3.1. Poziomy uprzywilejowania

W mikroprocesorach o architekturze IA-32 ochrona procesora oparta jest o koncepcję poziomów ochrony (ang. *Privilege Level*).



Rys. 3-1 Poziomy ochrony w mikroprocesorach Intel

Wykonywany proces pobiera instrukcje z segmentu wskazywanego przez rejestr CS. Rejestru CS nie da się przeładować inaczej niż przez instrukcję ICALL (IRET) lub przerwanie INT (IRET).
Używa on danych wskazywanych przez DS lub stosu wskazywanego przez SS.

Wykonywany w danej chwili proces charakteryzuje się aktualnym poziomem ochrony CPL (ang. *Current Privilege Level*) uzyskiwanym z dwóch najmłodszych bitów rejestru CS.

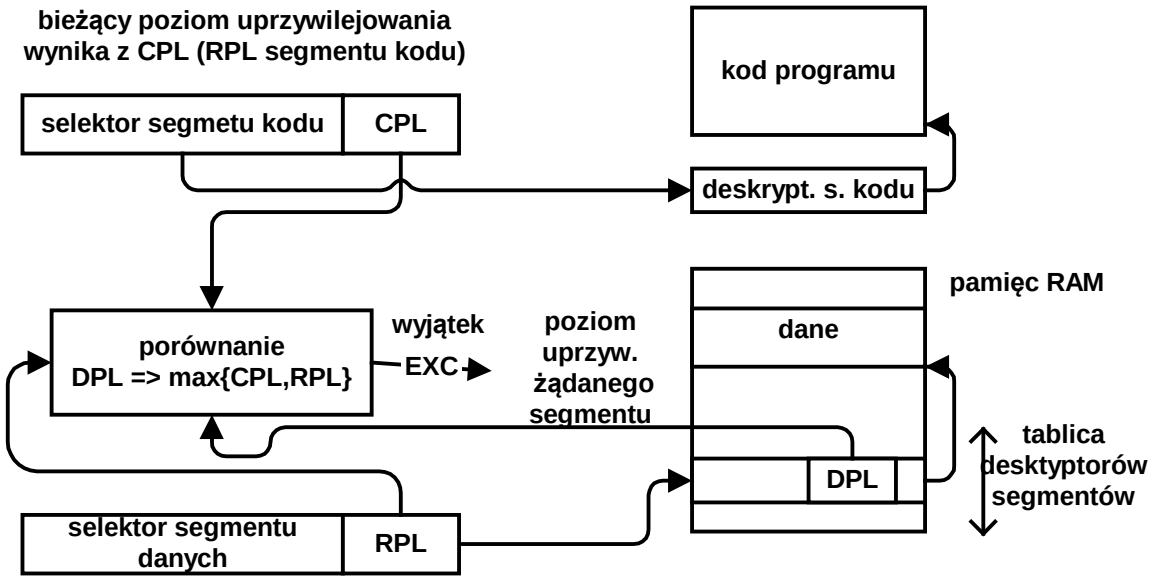
Operacja dostępu do danych używa jakiegoś selektora (np. DS.) który zawiera pole RPL (ang. *Requested Privilege Level)*, są to 2 najmłodsze bity.

Każdy segment pamięci (w tym danych) ma przyporządkowany poziom ochrony zapamiętany w polu DPL (ang. *Descriptor Privilege Level*) deskryptora segmentu do którego odnosi się żądanie.

Procesor ocenia prawa dostępu do segmentu poprzez porównanie obecnego poziomu uprzywilejowania CPL (wynika on z pola CPL), pola RPL selektora segmentu operandu i poziomu uprzywilejowania jaki ma deskryptor segmentu do którego przyznany ma być dostęp. Wartość CPL równa jest RPL z rejestru CS (selektor segmentu kodu).

$$DPL \Rightarrow \max\{CPL,RPL\}$$

- Przyznawany jest dostęp do danych o tym samym lub niższym poziomie ochrony (mniej ważnych).
- Dopuszczane jest wywoływanie procedur o tym samym lub wyższym poziomie ochrony (bardziej godnych zaufania)



Rys. 3-2 Kontrola dostępu prosesu do żądanego segmentu danych

3.2. Instrukcje systemowe

Wśród wszystkich instrukcji procesora wyróżnia się instrukcje zarezerwowane dla systemu. Są to:

- instrukcje dostępu do rejestrów systemowych,
- ładowania tablic deskryptorów,
- zatrzymywanie procesora,
- zmiany niektórych flag.

Instrukcje systemowe mogą być wykonywanie tylko w trybie uprzywilejowania CPL = 0. Gdy CPL procesu wykonującego instrukcję systemową jest różny od zera zostanie wygenerowany wyjątek.

3.3. Ochrona wejścia – wyjścia

- pole IOPL
- bitmapy

Pole IOPL

Pozwolenie na wykonywanie operacji wejścia wyjścia zależy od zawartości pola IOPL (ang. *Input Output Privilege Level*) wartość 0-3 zapisanego w rejestrze flag procesora. Gdy poziom uprzywilejowania CPL procesu bieżącego jest niższy od zawartości pola IOPL i następuje próba wykonania instrukcji wejścia wyjścia generowany jest wyjątek.

Bitmapy

Dla każdego zadania określona jest bitmapa zezwoleń dostępu (ang. *I/O Permission Bitmap*). Określa ona który adres z 64 KB przestrzeni wejścia wyjścia może być przez bieżący proces użyty.

Poziomy ochrony	Kontrola dostępu do danych zawartych w innych niż bieżący segmentach
	Kontrola wywoływania procedur zawartych w innych niż bieżący segmentach
Instrukcje systemowe	Zabezpieczenie istotnych funkcji procesora jak zarządzanie pamięcią, pamięcią, zatrzymywanie. Funkcje dostępne tylko dla procesów wykonywanych z poziomu jądra systemu operacyjnego (CLP= 0).
Pole IOPL w rejestrze znaczników	Określenie poziomu uprzywilejowania w którym mogą być wykonywane instrukcje wejścia wyjścia.
Bitmapy zezwoleń dostępu we / wy	Określenie adresów portów z przestrzeni wejścia wyjścia które mogą być użyte przez proces.

Tab. 3-1 Zestawienie mechanizmów ochrony architektury IA-32

4. Obsługa przerwań i wyjątków

Sekwencja wykonywanych rozkazów określona jest poprzez program. Może być ona jednak zmieniana na skutek zewnętrznego zdarzenia zwanego przerwaniem, lub wewnętrznego zwanego wyjątkiem.

Przerwanie - reakcja na asynchroniczne zdarzenie powstałe na zewnątrz procesora.

Wyjątek - powstaje przez detekcję przez procesor nienormalnego stanu wewnętrznego.

W procesorze Intel 386 wyróżnia się dwa źródła przerwań i dwa źródła wyjątków.

Przerwania:

- Przerwania maskowane które sygnalizowane są procesorowi poprzez sygnał na nóżce INTR. Odpowiadają one przerwaniom zgłaszanym przez urządzenia zewnętrzne.
- Przerwania niemaskowalne (ang. *Non-maskable Interrupt*) które sygnalizowane są procesorowi poprzez sygnał na nóżce NMI.

Wyjątki :

- Wyjątki wykryte przez procesor które dzielimy na błędy (ang. *fault*), pułapki (ang. *trap*) i zaniechania (ang. *abort*).
- Wyjątki wygenerowane programowo określane są jako pułapki i nazywane też przerwaniami programowymi.

4.1. Tablica deskryptorów przerwań

Procesor łączy z każdym przerwaniem i wyjątkiem pewien numer identyfikacyjny z zakresu 0 do 255 nazywany numerem przerwania.

- Dla części wyjątków i przerwań numer identyfikacyjny jest z góry zdefiniowany. Przerwania 0-31 są zarezerwowane dla wyjątków.
- Dla przerwań maskowalnych nadawany jest przez zewnętrzny układ programowalnego kontrolera przerwań.

.

Powiązanie numeru identyfikacyjnego wyjątku czy przerwania z procedurą jego obsługi następuje przez tablicę deskryptorów przerwań IDT (ang. *Interrupt Descriptor Table*) nazywaną też tablicą wektorów przerwań. Położenie tej tablicy zawarte jest w rejestrze systemowym IDTR (ang. *Interrupt Descriptor Table Register*)

Tablica IDT zawierać może trzy rodzaje deskryptorów:

- bramkę przerwania (ang. *Interrupt Gate*),
- bramkę pułapki (ang. *Trap Gate*)
- bramkę zadania (ang. *Task Gate*).

przesunięcie 31..16	**P**	**DPL**	**0001**	**T/I**	**000**	**nie używane**
selektor	**przesunięcie 15..0**					

Rys. 4-1 Format deskryptora bramki przerwania i pułapki

Gdy bit T/I jest ustawiony to deskryptor odnosi się do bramki pułapki a gdy nie do bramki przerwania.

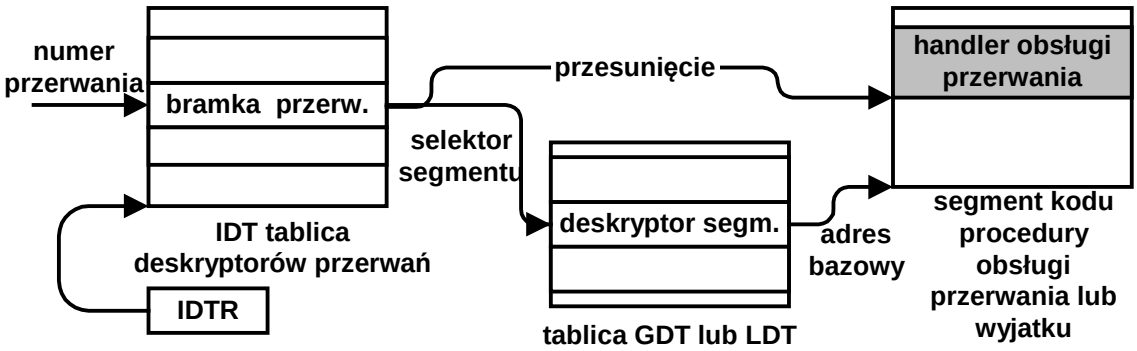
Wektor	Opis	Typ
0	Dzielenie przez 0	Błąd
1	Zarezerwowane	Błąd, pułapka
2	Przerwanie NMI	Przerwanie
3	Punkt wstrzymania INT3	Pułapka
4	Nadmiar	Pułapka
...	...	
8	Podwójny błąd	Zaniechanie
10	Błędny segment stanu zadania TSS	Błąd
11	Segment nieobecny	Błąd
12	Błąd segmentu stosu	Błąd
13	Ogólny błąd ochrony	Błąd
14	Błąd strony	Błąd
...	...	
20-31	Zarezerwowane	
32-255	Przerwanie zewnętrzne	Przerwanie

Tab. 4-1 Niektóre przerwania i wyjątki trybu chronionego dla architektury IA-32

4.2. Przebieg obsługi przerwania lub wyjątku

Wystąpienie przerwania lub wyjątku, któremu odpowiada umieszczona w tablicy IDT bramka przerwania lub wyjątku powoduje działanie podobne do wywołania procedury wykonywanej w kontekście zadania bieżącego.

Umieszczony w deskryptorze selektor wskazuje na segment w którym zawarta jest procedura obsługi wyjątku lub przerwania. Przesunięcie wskazuje na adres procedury która ma się wykonać.Po zakończeniu handlera następuje powrót do przerwanego zadania.



Rys. 4-2 Wywołanie procedury obsługi przerwania lub wyjątku

Gdy numer przerwania wskazuje na deskryptor IDT, który jest bramką zadania to obsługa przerwania podobna jest do wywołania zadania, na które wskazuje umieszczony w deskryptorze selektor.

4.3. Obsługa wyjątków

Wyjątki powodowane są przez wykonywany właśnie program. Architektura IA-32 wyróżnia następujące źródła wyjątków:

- błąd w programie,
- przerwania programowe
- wyjątek powodowany przez błąd sprzętowy.

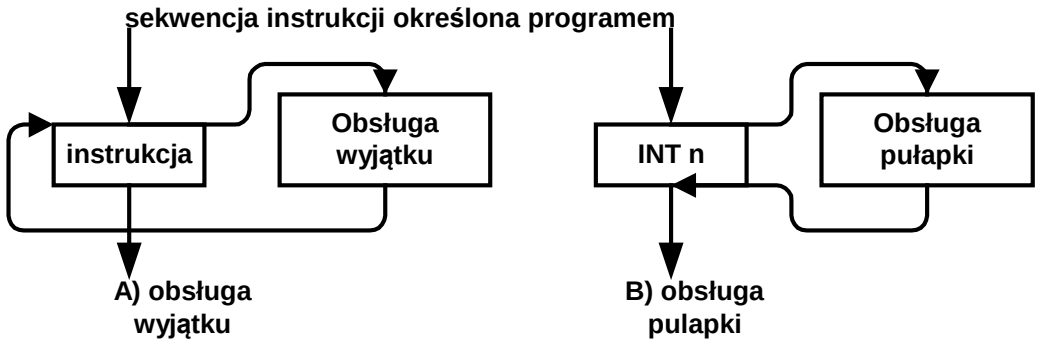
Wyjątki dzielimy na:

- błędy ,
- pułapki,
- zaniechania.

Błąd wykrywany jest zanim instrukcja go powodująca zostanie wykonana. Procedura obsługi błędu może poprawić błąd i program może być kontynuowany. Po zakończeniu procedury obsługi wyjątku, sterowanie wraca do instrukcji która go spowodowała.

Przerwanie programowe INT n (pułapka) powoduje zgłoszenie wyjątku zaraz po wykonaniu instrukcji INT. Po zakończeniu procedury jego obsługi sterowanie przekazywane jest do instrukcji następnej.

Zaniechania zgłaszane są gdy nie da się precyzyjnie zlokalizować położenia błędu i program nie może być kontynuowany (błędy sprzętowe i błędy w tablicach systemowych).



Rys. 4-3 Obsługa wyjątku i przerwania programowego (pułapki)

4.4. Przerwania sprzętowe

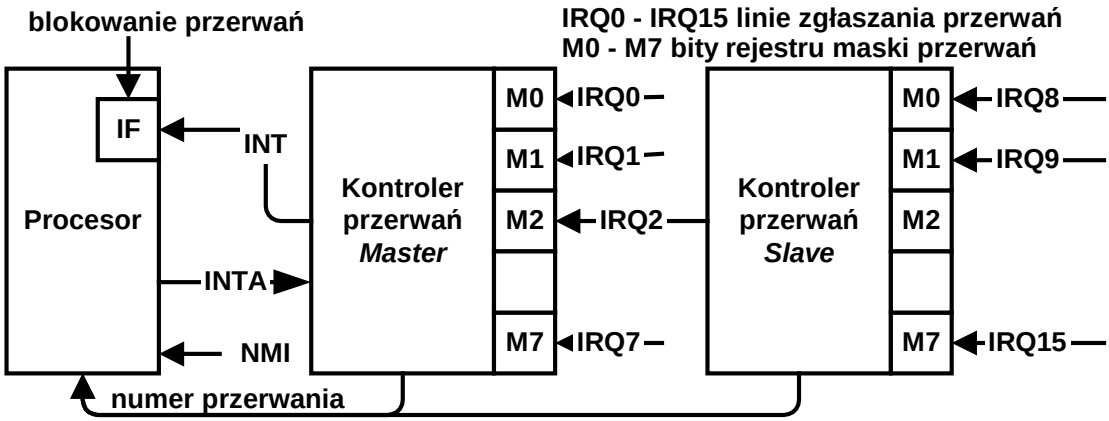
Przerwania sprzętowe dzielimy na:

- maskowalne
- niemaskowalne.

W komputerze występuje konieczność obsługi wielu przerwań, podczas gdy procesor zazwyczaj zawiera tyko jedno wejście przerywające. Występuje konieczność użycia urządzenia nazywanego kontrolerem przerwań (ang. *Interrupt Controller*).

Funkcje kontrolera przerwań:

- Rozstrzyganie konfliktu w przypadku wystąpienia wielu przerwań.
- Tworzenia powiązania pomiędzy nóżkami układu na których pojawiają się przerwania a numerami identyfikacyjnymi przerwań.



Rys. 4-4 Kontroler główny i podrzędny w komputerze PC

Kontroler przerwań posiada dwa rejestry umieszczone w przestrzeni wejścia wyjścia:

- rejestr sterujący CR (*ang. Control Register*)
- rejestr maski przerwań M (*ang. Interrupt Mask Register)*.

Rejestr sterujący CR służy do programowania kontrolera a rejestr maski M umożliwia indywidualne blokowanie poszczególnych linii zgłaszania przerwań.

Obsługa przerwania:

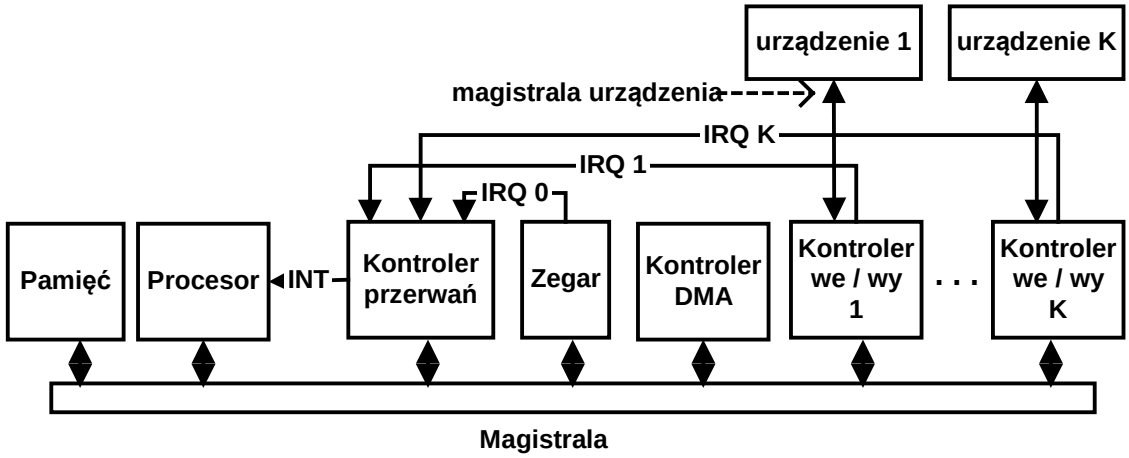
- Urządzenie sygnalizuje przerwanie poprzez wystawienie sygnału na linii zgłaszania IRQ_i.
- Kontroler przerwań decyduje czy przerwanie *i* może być przyjęte. Gdy jest ono zamaskowane nie będzie przyjęte. Gdy właśnie obsługiwane jest przerwanie o tym samym lub wyższym priorytecie to zgłoszenie musi poczekać do zakończenia obsługi bieżącego przerwania. Gdy przerwanie może być przyjęte kontroler wystawia do procesora sygnał INTR.
- Po zakończeniu bieżącego cyklu rozkazowego procesor sprawdza stan nóżki INTR. Gdy flaga IF jest ustawiona przerwanie będzie przyjęte. Procesor potwierdza przerwanie sygnałem INTA w odpowiedzi na co kontroler przesyła numer przerwania *n* = *baza* + *i*.
- Procesor sprawdza zawartość wektora *n* tablicy IDT. W zależności od jej zawartości uruchamia zadanie obsługi przerwania lub procedurę obsługi przerwania.
- W ramach obsługi przerwania testowane są rejestry urządzeń skojarzonych z danym przerwaniem. Znajdowana jest przyczyna przerwania i wykonywana jest jego obsługa.
- Po zakończeniu obsługi przerwania procesor wysyła informację o tym do kontrolera w postaci polecenie EOI (ang. *End Of Interrupt*).
- Przywracany jest kontekst przerwanego procesu.

Typ przerwania	Funkcja	Własności
Przerwania maskowalne	Obsługa zdarzeń zewnętrznych	Asynchroniczne
Przerwania niemaskowalne	Obsługa zdarzeń awaryjnych	Asynchroniczne
Wyjątki	Obsługa błędów programów	Synchroniczne
Pułapki	Wywołania systemu, programy uruchomieniowe operacyjnego	Synchroniczne

Tab. 4-2 Rodzaje przerwań w mikroprocesorach architektury IA-32

5. Kontrolery wejścia wyjścia

System wejścia wyjścia zapewnia dostęp do urządzeń zewnętrznych. Zapewnia on także komunikację z użytkownikiem. System wejścia wyjścia składa się z urządzeń i obsługujących je kontrolerów podłączonych do magistrali komputera.



Rys. 5-1 Uproszczony schemat komputera jednoprocesorowego.

Kontroler wejścia wyjścia pełni rolę pośrednika pomiędzy urządzeniem, procesorem i pamięcią operacyjną:

- z jednej strony dołączony jest on zwykle do magistrali komputera (np. magistrali PCI)
- z drugiej strony do magistrali urządzenia (np. magistrali USB czy SCSI)

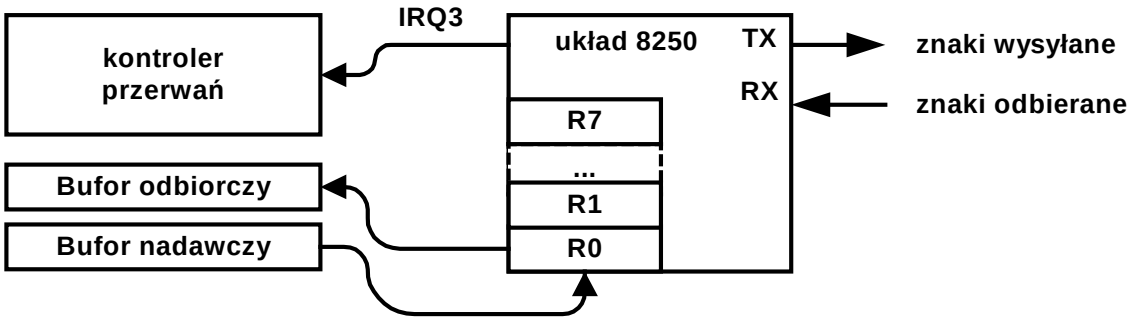
Kontroler składa się z kilku rejestrów umieszczonych w przestrzeni wejścia wyjścia. Zwykle jest to:

- rejestr wejściowy,
- wyjściowy,
- statusowy,
- danych.

Przykład - kontroler transmisji szeregowej zgodnego z układem 8250

Większość stosowanych obecnie kontrolerów wejścia wyjścia może sygnalizować przerwaniem takie zdarzenia jak:

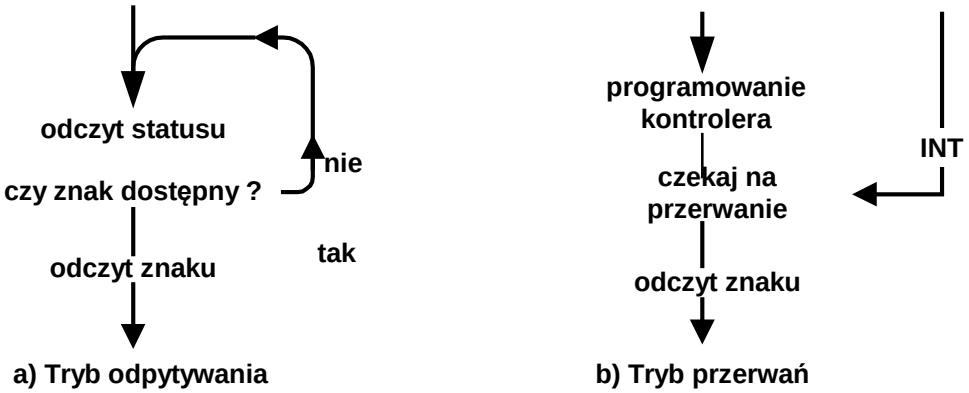
- pojawienie się nowych danych do odczytu,
- zakończenie wysyłania danych zapisywanych,
- zmiana statusu,
- wystąpienie błędu.



Rys. 5-2 Kontroler transmisji szeregowej

Odczyt znaku z kontrolera możliwy jest w dwóch trybach:

- trybie odpytywania,
- trybie przerwań.



Rys. 5-3 Odczyt znaku z kontrolera wejścia wyjścia

Tryb pracy	Procesor	Szybkość	Wykorzystanie sprzętu	Wsparcie sprzętowe	Wsparcie programowe
Odpytywanie	Zajęty	Mała	Małe	-	-
Tryb przerwań	Wolny	Średnia	Średnie	Przerwania	Procesy
Transmisja blokowa	Wolny	Duża	Duże	Przerwania, układ DMA	Procesy

Tab. 5-1 Porównanie metod transmisji danych pomiędzy pamięcią a kontrolerem urządzenia wejścia wyjścia

6. Literatura

[1] Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 System Programming Guide, Intel
http://download.intel.com/products/processor/manual/325384.pdf