

# Sprzętowe wspomaganie pamięci wirtualnej

Stanisław Skonieczny

6 grudnia 2002

## Spis treści

<b>1 Intel</b>	<b>2</b>
1.1 Tryby pracy procesora . . . . .	2
1.2 Adresowanie liniowe . . . . .	2
1.3 Adresowanie rzeczywiste . . . . .	2
1.4 Segmentacja . . . . .	3
1.5 Stronicowanie . . . . .	4
<b>2 PowerPC</b>	<b>5</b>
<b>3 MIPS32</b>	<b>5</b>
3.1 Segmenty . . . . .	6
3.2 TBL . . . . .	6
<b>4 Alpha</b>	<b>7</b>

# 1 Intel

## 1.1 Tryby pracy procesora

Począwszy od architektury 386, Intel wspiera trzy tryby pracy procesora:

- **Tryb chroniony**

Podstawowy tryb pracy procesora.

Model pamięci zależy tu od systemu.

Dozwolone są: segmentacja, stronicowanie, adresowanie liniowe, adresowanie rzeczywiste.

Możliwe jest nawet stosowanie różnych modeli dla różnych procesorów.

- **Tryb rzeczywisty**

Zapewnia środowisko procesora 8086 z niewielkimi zmianami (takimi jak przełączanie się do innych trybów).

Adresowanie rzeczywiste.

- **Tryb systemowy**

Jest to tryb, do którego proces przełącza się, by wykonać swoje własne zadania (np. zarządzanie energią).

Po ich wykonaniu powraca do poprzedniego stanu.

W tym trybie procesor działa na oddzielnej przestrzeni adresowej (SMRAM), adresowanej podobnie jak w trybie rzeczywistym.

## 1.2 Adresowanie liniowe

Najprostsze rozwiązanie.

Adres logiczny równy jest adresowi fizycznemu.

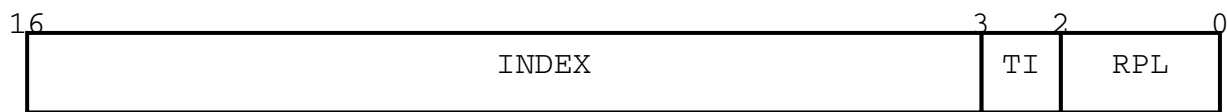
Wszystkie rejestry segmentów ustawione są na zero.

## 1.3 Adresowanie rzeczywiste

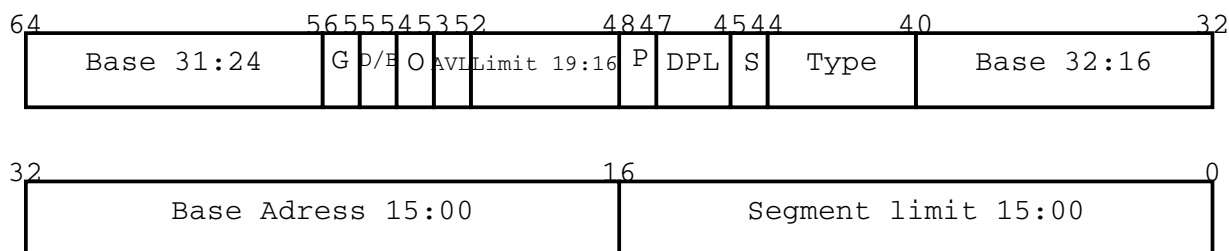
Jest to podstawowy sposób adresowania w architekturach starszych niż 386. Do pamięci odwołujemy się używając dwóch 16-bitowych rejestrów: segmentu i offsetu. Adres fizyczny uzyskujemy przez przesunięcie z lewo rejestru segmentu o 4 bity i dodanie offsetu.

Przebieg adresowa nie większa niż 1M bajt.

Rozmiar segmentu stały, równy 64k bajtów.



Rysunek 1: Selektor segmentu



Rysunek 2: Deskryptor segmentu

## 1.4 Segmentacja

**Segmenty** nie muszą mieć stałej długości, a maksymalny rozmiar wynosi 4G bajty.

W architekturze i386 mamy 6 rejestrów segmentowych: CS, DS, ES, FS, GS, SS.

CS wskazuje na segment z kodem, SS na segment ze stosem, pozostałe na segmenty z danymi.

**Tablice deskryptorów segmentów** (globalna i lokalna)

GDT (Global Descriptor Table) jest strukturą umieszczoną w pamięci, wskazywaną przez GDTR (GDTRRegister). Może być tylko jedna w systemie. Zawiera wpisy dla segmentów zawierających LDT.

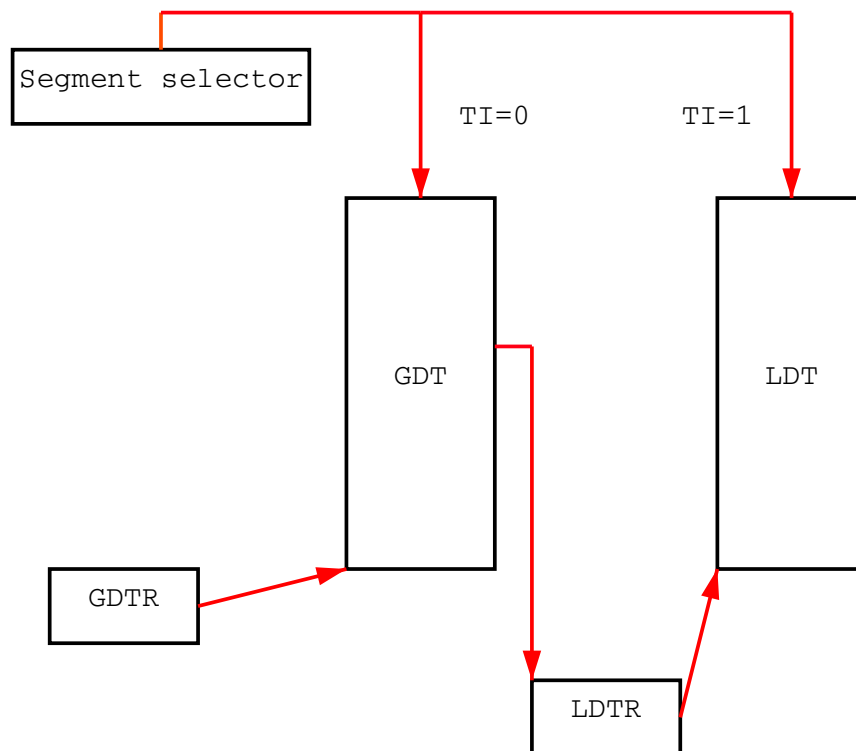
LDT (Local Descriptor Table) jest lokalną tablicą przechowującą deskryptory segmentów. Może być ich wiele, jednak zawsze wyróżnione jest, z której korzystamy, przez zawartość rejestru LDTR. Do rejestru lokalnej tablicy deskryptorów wczytujemy dane instrukcją LLDT z globalnej tablicy deskryptorów.

**Odwołanie do pamięci** składa się z dwóch rejestrów.

Do pierwszego, rejestru segmentu ładujemy selektor segmentu (16 bitów) jedn z operacji: MOV, POP, LDS, LES, LFS, LGS czy LSS. Selektor sementu określa, czy odwołujemy się do GDT, czy LDT Wtedy też do ukrytej części segmentu procesor wczytuje informacje z deskryptora segmentu, aby nie ładować w czasie odwołania do pamięci. Drugi rejestr jest offsetem wewnątrz segmentu (32 bity).

**Deskryptor segmentu** jest 64 bitowy. Zawiera takie informacje, jak:

- Base adres - początek segmentu w pamięci fizycznej (32 bity)
- Segment limit - jego rozmiar (20 bitów)
- DPL (descriptor privilege level) - poziom uprzywilejowania



Rysunek 3: Relacja tablic segmentów

- Flaga S - rodzaj. Może być: 0=system, 1=kod lub dane
- Typ - między innymi np. kierunek rozrostu, prawa odczytu, zapisu, czy był używany, czy jest wykonywalny
- Flaga P - obecność w pamięci, czyli czy generować wyjątek segment-not-present przy dostępie
- Flaga D/B - dla kodu specyfikuje długość adresu i operandów, dla danych górną granicę segmentu: 4GB/64kB.
- Flaga G (granulacja) - determinuje skalowanie limitu segmentu: 0-limit w bajtach, 1-limit w 4kB

## 1.5 Stronicowanie

Ze stronicowanie można korzystać w trybie adresowania liniowego, lub segmentacji.

Jeśli jest włączone, to adres fizyczny, jaki dadzą powyższe techniki zostanie zamieniony na adres logiczny dla stronicowania.

Mamy dostępne trzy techniki stronicowania:

- **PG** - paging (od 386)
- **PSE** - page size extension (Pentium, Pentium Pro)
- **PAE** - physical address extension (Pentium, Pentium Pro)

Technika PG wspomaga działanie wielu procesów w trybie virtual 8086. Jest to stronicowanie na żądanie. Tryb virtual 8086 jest determinowany flagą procesora w trybie chronionym. Poszczególne procesy widzą wtedy swoją przestrzeń adresową jak w trybie rzeczywistym. PSE umożliwia korzystanie z dużych stron (2MB, 4MB). PAE daje 36-bitową przestrzeń adresową.

**Tłumaczenie adresu dla stron 4-KB** Adres 32 bitowy dzielony jest na 3 części. Najstarsze 10 bitów jest indeksem w katalogu stron, określającym tablicę stron. Następne 10 bitów to numer indeksu w tablicy stron. Przesunięcie to ostatnie 12 bitów.

**Tłumaczenie adresu dla stron 4-MB** W tym wypadku mamy 10 starszych bitów jako numer indeksu w katalogu stron i 22 bitowy offset.

## 2 PowerPC

Architektura PowerPC pozwala mapować pamięć wirtualną na pamięć fizyczną. Segmenty są 256M bajtowymi ciągłymi obszarami pamięci. 16 segmentów pozwala adresować do 4G bajtów pamięci.

Przy wyliczaniu adresu fizycznego górne 4 bity 32 bitowego adresu oznaczają segment. Pozostałe 28 bitów jest offsetem w obrębie segmentu. Rejestry segmentów są 24 bitowe, więc adres pośredni jest 52 bitowy. Teraz starsze 40 bitów jest wirtualnym numerem strony, a 12 młodszych przesunięciem wewnątrz strony. Wirtualny adres strony jest zamieniany na 20 bitowy adres strony fizycznej. Adres wyjściowy jest więc ma tak jak na początku - 32 bity.

Tablice stron w architekturze PowerPC są haszujące. Oznacza to, że aby znaleźć stronę fizyczną dla danej strony wirtualnej, musimy przejrzeć listę wpisów.

Ochronę zasobów zapewniają rejestry systemowe, które modyfikować może tylko system. Posiadają one 24 bitowe identyfikatory dostępu, oznaczające kto ma dostęp do danego segmentu.

## 3 MIPS32

Pamięć wirtualna w architekturze MIPS wspiera stronicowanie, poprzez obecność rejestrów TLB. Umożliwia także ochronę obszarów pamięci dzięki identyfikatorom obszarów pamięci: identyfikator procesu musi się zgadzać z identyfikatorem zapisanym w tablicy TLB. Architektury R2000 oraz R3000 zapewniają 64 takie identyfikatory, architektura R10000 zapewnia 256. Zmusza to, by w przypadku, gdy jest więcej uruchomionych procesów, niektóre z nich dzieliły przestrzeń adresową.

### 3.1 Segmenty

W architekturze MIPS32 logiczna przestrzeń adresowa jest 32-bitowa, podzielona na 5 segmentów, zorganizowanych w następujący sposób:

Segment	Opis	Początek	Koniec
kseg3	Kernel Mapped	E000 0000	FFFF FFFF
kseg	Supervisor Mapped	C000 0000	DFFF FFFF
kseg1	Kernel Unmapped Uncached	A000 0000	BFFF FFFF
kseg0	Kernel Unmapped	8000 0000	9FFF FFFF
useg	User Mapped	0000 0000	7FFF FFFF

Segmenty opisane jako „Mapped” są stronicowane. Przy dostępie do nich adres logiczny tłumaczony jest przy pomocy tablicy TLB (lub przez inną jednostkę tłumaczącą) na adres fizyczny.

Segmenty „kseg0”, oraz „kseg1” nie są stronicowane. Umożliwiają one wgląd w najniższą część adresowej przestrzeni fizycznej, zaczynającą się od adresu 0. Segment „kseg1” dodatkowo umożliwia ominięcie wszystkich poziomów pamięci podręcznej.

Procesor w architekturze MIPS może działać w trzech trybach: jądra, superużytkownika oraz użytkownika. Każdy segment ma przypisane tryby, w których można z niego korzystać, tak jak w tabelce. Dodatkowo tryby te mają różne poziomy uprzywilejowania: najwyższy jest tryb jądra, najniższy użytkownika. Wyższy poziom uprzywilejowania oznacza zezwolenie na dostęp do wszystkich segmentów do których jest dostęp na niższym poziomie. Przykład: z segmentu „kseg” można korzystać tylko w trybach jądra i superużytkownika. W trybie użytkownika próba odwołania się do segmentu spowoduje Address Error Exception.

### 3.2 TBL

Pojedynczy wpis w rejestrze TLB zawiera pola:

- Mask
- VPN2 - Numer wirtualnej strony (podzielony przez dwa)
- ASID, G - Identyfikator przestrzeni adresowej i bit ogólnego dostępu
- PFN0, C0, D0, V0 - Numer strony fizycznej, oraz bity valid, dirty, cache coherency
- PFN1, C1, D1, V1 - Jak wyżej

Warto zauważyć, że jedna strona wirtualna skojarzona jest z dwoma stronami fizycznymi. To, która strona fizyczna zostanie wybrana zależy od adresu, oraz pola Mask. Dla każdej strony wirtualnej mamy zatem dwa wpisy w tablicy TLB.

Dodatkowo mamy możliwość zezwolenia na dostęp wszystkim procesom (bit G przy polu ASID).

## 4 Alpha

Alpha jest architekturą RISC przeznaczoną dla wielu procesorów. Zarówno instrukcje, jak i 32 całkowitoliczbowe oraz 32 zmiennoprzecinkowe rejestry są 64 bitowe. Instrukcje wczytania i zapisu przenoszą 8, 16, 32 lub 64 bitowe jednostki pomiędzy pamięcią a rejestrami. Adresy są 64 bitowe, liniowe, bez segmentacji.

Adresy fizyczne podzielone są na cztery obszary, zależnie od dwóch najstarszych zaimplementowanych bitów. Każdy z obszarów różni się pod względem:

- granulacji - 8, 16, 32, czy 64 bitowa.
- spójności - czy zapis jednego procesora wpływa na zmianę danych drugiego. Można ją ustawiać dla każdego rejonu.
- szerokości dostępu - jak wielki obszar pamięci jest czytany przy dostępie
- sposobu zachowania

Część pamięci może być dzielona między procesorami, dlatego Alpha daje mechanizmy takie jak: zakładanie blokad i zapisanie warunkowe (jeśli mamy blokadę, to zapisujemy, a potem zwalniamy blokadę). Pozwala to atomowo zapisywać i odczytywać dane. Każdy procesor jednak może założyć maksymalnie jedną blokadę.