

# BUDOWA PROCESORA RODZINY X86

---

## REJESTRY

# REJESTRY – CZYM SĄ?

- Za Wikipedią:

**Rejestry procesora** to komórki pamięci o niewielkich rozmiarach (najczęściej 4/8/16/32/64/128 bitów) umieszczone wewnątrz procesora i służące do przechowywania tymczasowych wyników obliczeń, adresów lokacji w pamięci operacyjnej itd. Większość procesorów przeprowadza działania wyłącznie korzystając z wewnętrznych rejestrów, kopiując do nich dane z pamięci i po zakończeniu obliczeń odsyłając wynik do pamięci.

# TYPY

- rejestry danych
  - rejestry adresowe
  - rejestry ogólnego zastosowania (ang. *general purpose*)
  - rejestry zmiennoprzecinkowe
  - rejestry stałych
  - rejestry wektorowe
  - rejestry specjalne
  - rejestry instrukcji
-

# TYPY X86

- 8 rejestrów ogólnego przeznaczenia (32 bity)
- 6 rejestrów segmentowych (16 bitów)
- rejestr flag (32 bity)
- wskaźnik instrukcji - EIP (32 bity)
- 8 rejestrów jednostki zmiennoprzecinkowej procesora (w procesorach nowszych od 80386), kiedyś koprocesora arytmetycznego (80 bitów)
- w nowszych procesorach 8 rejestrów rozszerzeń MMX/3DNow/3DNow! (64 bity)
- w nowszych procesorach 8 rejestrów XMM rozszerzeń SSE, SSE2, SSE3 (128 bitów)

# IP (EIP)

- (ang.) *extended insertion point*
- Rola: wskazuje na adres aktualnie wykonywanej instrukcji
- Uruchomienie programu => EIP wskazuje na początek segmentu kodu

# PODZIAŁ REJESTRÓW OGÓLNEGO PRZEZNACZENIA

8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b	8b
		AH	AL			BH	BL			CH	CL			DH	DL
		AX				BX				CX				DX	
EAX				EBX				ECX				EDX			

16b	16b	16b	16b	16b	16b	16b	16b
	SP		BP		SI		DI
ESP		EBP		ESI		EDI	

# AX (EAX)

- Akumulator (ang. *accumulator*)
- Jego pamięć wykorzystuje arytmometr
- Rejestr operacji arytmetycznych, logicznych, przechowywanie wyników
- Używany zawsze tam gdzie zachodzi mnożenie, dzielenie
- Najbardziej efektywny

## BX (EBX)

- Rejestr bazowy (ang. *base register*)
- Rola główna: adresowanie, lokalizacja w pamięci

## CX (ECX)

- Rejestr licznikowy (ang. *counter register*)
  - Rola główna: licznik (pętli)
-



# DX (EDX)

- Rejestr danych (ang. *data register*)
- Rola główna: umożliwia przekaz/odbiór danych z portów wejścia/wyjścia
- Używany w operacjach mnożenia i dzielenia

## SP (ESP)

- Wskaźnik stosu (ang. *stack pointer*)
- Przechowuje wskaźnik wierzchołka stosu i pozwala na pobieranie/edycję danych.

## BP (EBP)

- Wskaźnik bazowy (ang. *base pointer*)
  - Wskaźnik danych w segmencie stosu
-

## SI (ESI)

- Rejestr źródłowy (ang. *source index*)
- Wskaźnik źródła

## DI (EDI)

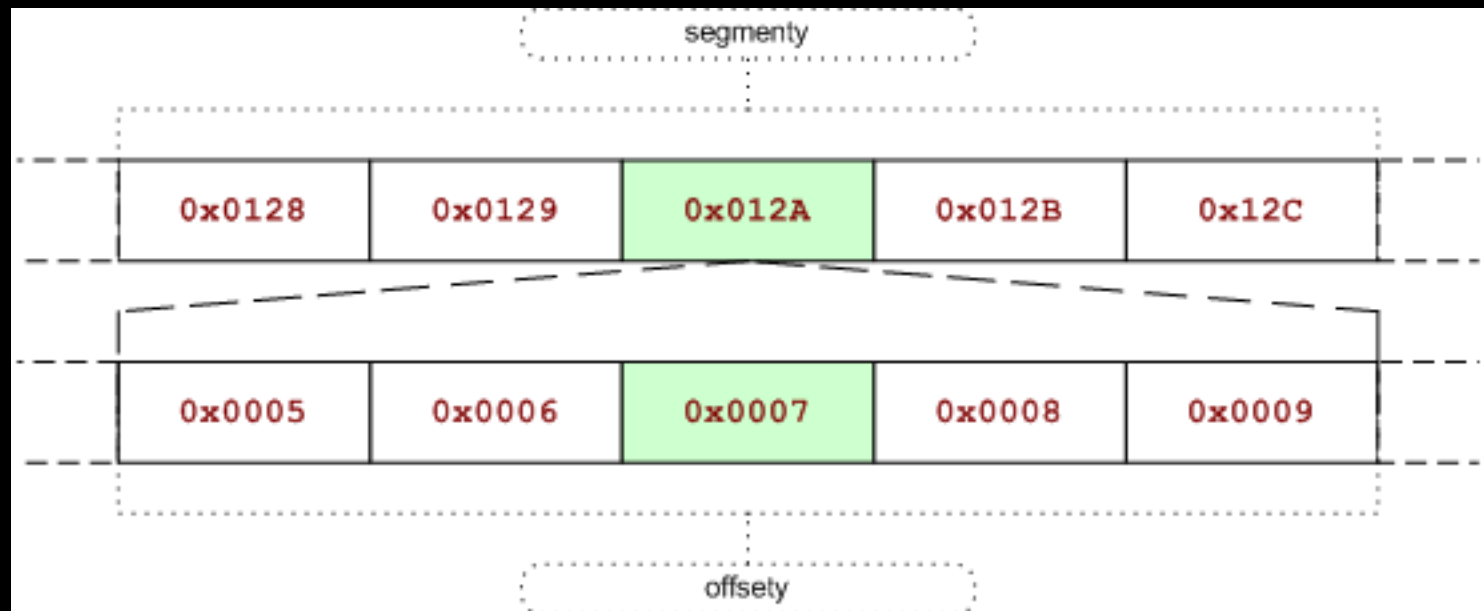
- Rejestr przeznaczenia (ang. *destination index*)
  - Wskaźnik przeznaczenia
-

# REJESTRY SEGMENTOWE

- CS - segment kodu
  - DS - segment danych
  - SS - segment stosu
  - ES - dodatkowy segment danych
  - FS - dodatkowy segment danych
  - GS - dodatkowy segment danych
-

# SEGMENTOWY MODEL PAMIĘCI

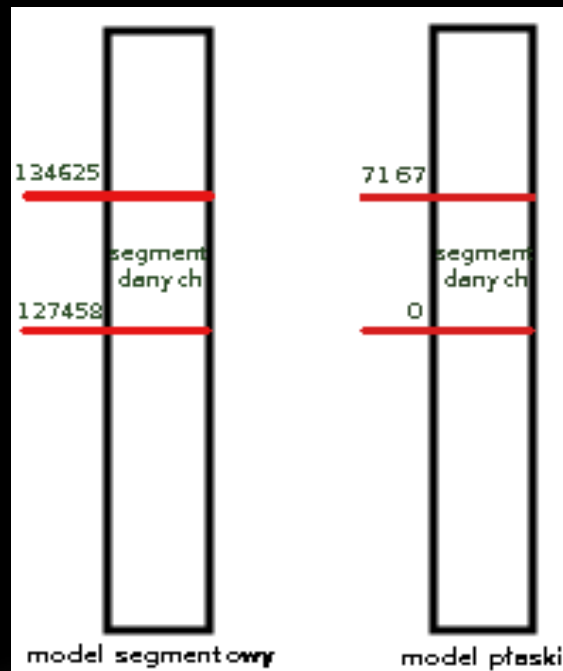
- Pamięć podzielona na bloki po 64 kilobajty



012A:0007

# PŁASKI MODEL PAMIĘCI

- Cała pamięć jest jednym segmentem.



# REJESTR FLAG - ARYTMETYCZNE

bit	id	nazwa	przyjmuje wartość 1, gdy
0	CF	carry flag	przekroczono zakres zmiennej (dodawanie/odejmowanie)
2	PF	parity flag	liczba jedynek w najmłodszym bajcie jest parzysta
4	AF	auxiliary flag	znacznik przeniesienia połówkowego
6	ZF	zero flag	wynik ostatniej operacji to 0
7	SF	sign flag	liczba ujemna
11	OF	overflow flag	dodanie dwóch liczb o tym samym znaku zmienia znak

# REJESTR FLAG - INFORMACJE

bit	id	nazwa	ustawiona, gdy
14	NT	nested task	obecny proces został wywołany instrukcją CALL
16	RF	resume flag	wystąpiło przerwanie zatrzymujące wykonanie procesu
20	VIP	virtual interrupt pending	trwa wirtualne przerwanie
21	ID	ID flag	jeśli można zmienić jej wartość oznacza to, że procesor obsługuje instrukcję CPUID



# REJESTR FLAG – DZIAŁANIE PROCESORA

bit	id	nazwa	gdy równy 1
8	TF	trap flag	procesor wchodzi w tryb pracy krokowe
9	IF	interrupt flag	przerwania są dopuszczalne
10	DF	direction flag	przetwarzanie łańcuchów odbywa się od lewej do prawej
12/13	IOPL	input-output privilage level	odczyt/zapis z/do wyjścia jest dozwolony; jest to 2-bitowa flaga
17	VM	virtual 8086 mode	przejsie w tryb emulacji procesora 8086
18	AC	alignment check	sprawdzanie wyrównania (
19	VIF	virtual interrupt flag	przerwania są dopuszczalne (kopia IF w trybie wirtualnym)

# REJESTR KONTROLNE – CR0

bit	id	nazwa	gdy równy 1
31	PG	paging flag	stronicowanie włączone
30	CD	cache disable	wyłączona pamięć cache
29	NW	not write-through	zapis do pamięci bez użycia cache
18	AM	alignment mask	steruje kiedy bit flagowy AC generuje niepowodzenie braku wyrównania argumentu
16	WP	write protection	procesor nie może zapisać danych do stron 'tylko do odczytu'
5	NE	numeric error	włącza raportowanie błędów przez j. zmiennoprzecinkową
4	ET	extension type	informuje, że mamy koprocessor 80387 (0 - 80287)
3	TS	task switched	informuje, że należy zapisać stan koprocessora
2	EM	emulate flag	nie ma koprocessora (0 - x87)
1	MP	monitor coprocessor	sprawdza flagę TS podczas wykonywania WAIT/FWAIT
0	PE	protection enabled	system pracuje w trybie chronionym (0 - w rzeczywistości)

# REJESTRY KONTROLNE

- CR2 - zawiera wartość będącą błędem w adresowaniu pamięci (ang. Page Fault Linear Address)
- CR3 - jeśli stronicowanie jest włączone, umożliwia procesorowi zlokalizowanie położenia tablicy katalogu stron dla obecnego zadania
- CR4 - służy do kontroli błędów sprzętowych, stronicowania, wspomaga debugowanie

# REJESTRY JEDNOSTKI ZMIENNOPRZECINKOWEJ

- Dane trzymane są w rejestrach R0 - R7
  - Podwójna rozszerzona precyzja
  - Odwołania przez aliasy ST0 - ST7
  - Struktura stosu
  - Flagi
-

# SUPERSKALARNOŚĆ

- "Cecha mikroprocesorów umożliwiająca równoległe wykonywanie różnych instrukcji w danej jednostce czasu"

$$\begin{aligned}a &= b + 5 \\c &= a + 10\end{aligned}$$

$$\begin{aligned}a &= b + 5 \\c &= b + 15\end{aligned}$$

# POTOKOWOŚĆ (PIPELINE)

- "Technika komputerowa polegająca na zwiększeniu przepustowości wykonywanych instrukcji"

Instr. No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

# POTOKOWOŚĆ – DATA HAZARD

- RAW
    - i.  $R2 \leftarrow R1 + R3$
    - ii.  $R4 \leftarrow R2 + R3$
  - WAW
    - i.  $R2 \leftarrow R4 + R7$
    - ii.  $R2 \leftarrow R1 + R3$
  - WAR
    - i.  $R4 \leftarrow R1 + R3$
    - ii.  $R3 \leftarrow R1 + R2$
-

# POTOKOWOŚĆ - ROZWIĄZANIA

- Zatrzymywanie
- Przekierowanie
  - Instruction 0: Register 1 = 6
  - Instruction 1: Register 1 = 3
  - Instruction 2: Register 2 = Register 1 + 7 = 10

"Forwarding involves feeding output data into a previous stage of the pipeline. Forwarding is implemented by feeding back the output of an instruction into the previous stage(s) of the pipeline as soon as the output of that instruction is available."

- Wykonanie poza kolejnością



# PRZEMIANOWANIE REJESTRU (REGISTER RENAMING)

- "Wykonywanie instrukcji na innym niż domyślnie użytym rejestrze"

1.R1 = MEM[1024]

2.R1 = R1 + 2

3.MEM[1032] = R1

4.R1 = MEM[2048]

5.R1 = R1 + 4

6.MEM[2056] = R1

1. R1 = MEM[1024] 4. R2 = MEM[2048]

2. R1 = R1 + 2 5. R2 = R2 + 4

3. MEM[1032] = R1 6. MEM[2056] = R2

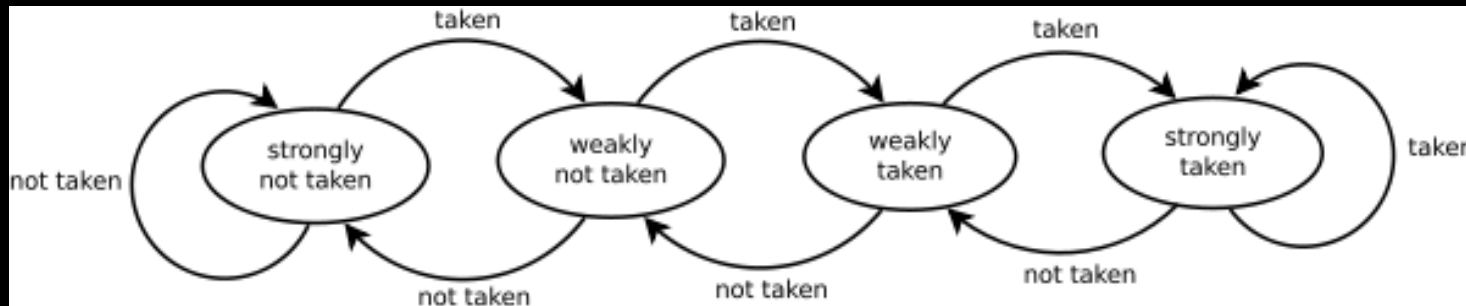
# WYKONANIE POZA KOLEJNOŚCIĄ (OUT-OF-ORDER EXECUTION)

- "Zdolność procesorów skalarnych do wykonywania instrukcji w kolejności innej niż ogólnie ustalona sekwencja"

1)  $a = b + 5$   
2)  $c = a + 10$   
3)  $d = e + 5$   
4)  $f = d + 10$

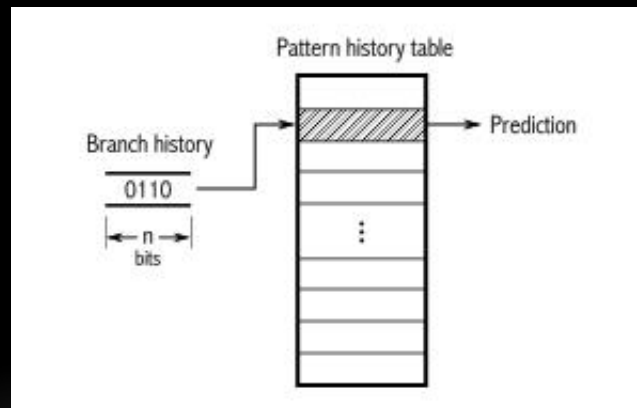
# SPEKULACYJNE WYKONANIE (SPECULATIVE EXECUTION)

- "Wykonywanie instrukcji zanim zaistnieje pewność co do potrzeby ich przetworzenia"
- Algorytmy
  - Przewidywanie statyczne
  - Wskaźnik nasycenia (ang. *saturating counter*)



# SPEKULACYJNE WYKONANIE (SPECULATIVE EXECUTION)

- "Wykonywanie instrukcji zanim zaistnieje pewność co do potrzeby ich przetworzenia"
- Algorytmy
  - Przewidywanie statyczne
  - Wskaźnik nasycenia (ang. *saturating counter*)
  - Two-level adaptive predictor



# AUTORZY

- Mateusz Krzyżanowski
  - Michał Markiewicz
  - Katarzyna Ostrowicz
- 
- Przedmiot: Programowanie mikroprocesorów x86
  - Semestr letni 2013
  - Bibliografia: wszystkie zdjęcia i cytaty pochodzą z Wikipedii.
-