

Instrukcje typu SIMD

Instrukcje typu SIMD

Single Instruction Multiple Data - przetwarzanych jest wiele strumieni danych przez jeden wykonywany program – cecha tzw. komputerów wektorowych.

Instrukcje SIMD dzieli się na:

- MMX (MultiMedia eXtensions lub Matrix Math eXtensions) - liczby całkowite,
- SSE (Streaming SIMD Extensions) - liczby zmiennoprzecinkowe.

Instrukcje typu MMX

MMX

- wprowadzone w 1997 przez Intela dla procesorów Pentium MMX.
- Przykłady zastosowań:
 - wyświetlanie grafiki trójwymiarowej: przekształcenia geometryczne, cieniowanie, teksturowanie;
 - dekodowanie obrazów JPEG i PNG;
 - dekodowanie i kodowanie filmów MPEG (m.in. wyznaczanie transformat DCT i IDCT);
 - filtrowanie sygnałów: obrazów statycznych, filmów, dźwięku;
 - wyświetlanie grafiki dwuwymiarowej (blue box, maskowanie, przezroczystość);
 - wyznaczanie transformat: Haara, FFT.

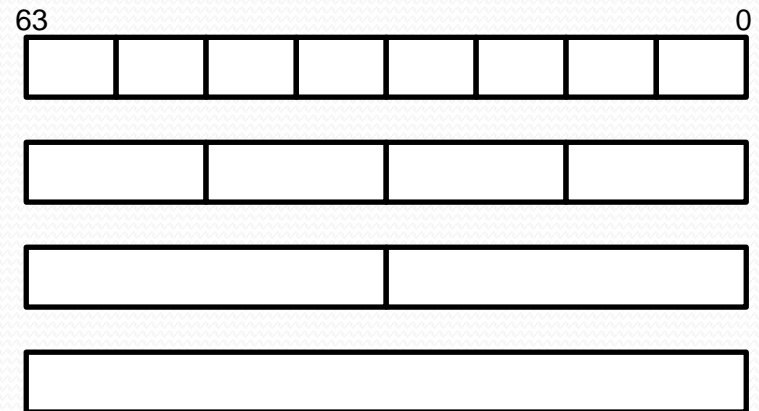
Rejestry MMX

- 8 rejestrów 64 bitowych oznaczanych jako MM0, ..., MM7,
- wykorzystują rejestry koprocesora – młodsze 64 bity (mantysa),
- odczyt i zapis wartości, powoduje **wzięcie w użycie zawartości wszystkich rejestrów koprocesora**, nie można mieszać obliczeń MMX z obliczeniami w koprocesorze, po zakończeniu obliczeń MMX należy zwolnić rejestry.

Typy danych

MMX wprowadził nowe wektorowe (macierzowe lub tablicowe) typy danych (ang. packed, czyli dosłownie spakowane, upakowane). „Spakowanie” polega traktowaniu danych 64-bitowych jako składających z odrębnych elementów o tej samej wielkości:

- 8×8 bitów (packed byte),
- 4×16 bitów (packed word),
- 2×32 bity (packed dword),
- 1×64 bity (quad word).



Budowa rozkazów

- Mnemoniki prawie wszystkich rozkazów MMX rozpoczynają się od litery **p** (od słowa packed);
- 3-4 literowy skrót wykonywanego działania (np. add, sub, mul);
- litera **s** lub **u** określa, że działanie jest wykonywane na liczbach ze znakiem (signed) lub bez znaku (unsigned);
- litera **s** oznacza operację wykonywaną z nasyceniem;
- rozmiar elementu wektora: **b** - bajt (8 bitów), **w** - słowo (16 bitów), **d** - podwójne słowo (32 bity).



paddusb

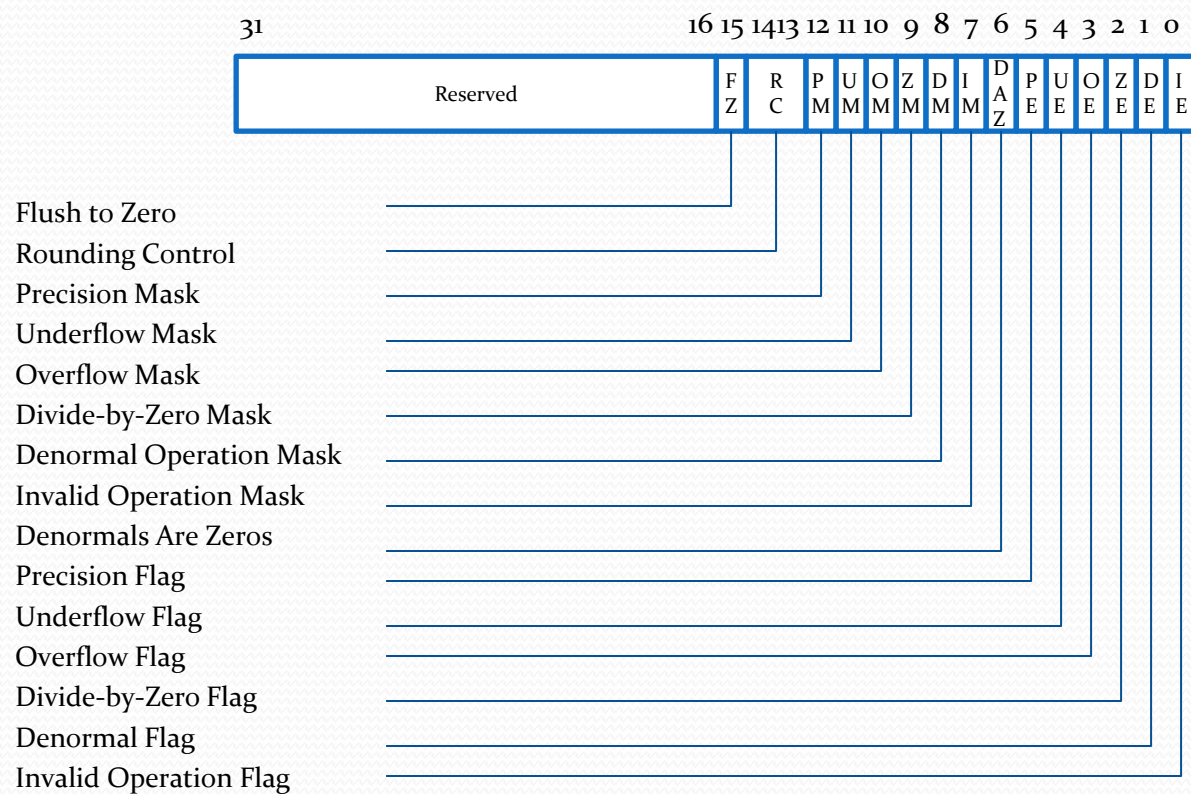
- Rozkaz **paddusb** wykonuje równoległe (**p**) dodawanie (add) bez znaku (**u**) z nasyceniem (**s**) liczb o rozmiarze bajtu (**b**)

Nasycenie

Zakresy liczb (dla 8 bitów):

- bez znaku $0 \dots 2^n - 1$ (0 ... 255)
- ze znakiem $-2^{n-1} \dots 2^{n-1} - 1$ (-128 ... 127)
- Jeśli wynik jest mniejszy od najmniejszej liczby z zakresu jest ustawiany na tę liczbę.
- Jeśli wynik jest większy od największej liczby z zakresu jest ustawiany na tę liczbę.
- Dla operacji bez nasycenia wynik jest obcinany do odpowiedniej ilości bitów.

Rejestr MXCSR



Operacje przesłania

- MOVD przesłanie podwójnego słowa
- MOVQ przesłanie poczwórnego słowa

`movd/movq cel, źródło`

Przesłanie podwójnego/poczwórnego słowa do/z rejestru mmx. Przy zapisie podwójnego słowa bity 32-63 rejestru mmx wypełniane są zerami.

Operacje konwersji

- PACKSSWB pakowanie z nasyceniem słów ze znakiem do bajtów
- PACKSSDW pakowanie z nasyceniem podwójnych słów ze znakiem do słów
- PACKUSWB pakowanie z nasyceniem słów bez znaku do bajtów
- PUNPCKHBW rozpakowanie z przeplotem starszych bajtów
- PUNPCKHWD rozpakowanie z przeplotem starszych słów
- PUNPCKHDQ rozpakowanie z przeplotem starszych podwójnych słów
- PUNPCKLBW rozpakowanie z przeplotem młodszych bajtów
- PUNPCKLWD rozpakowanie z przeplotem młodszych słów
- PUNPCKLDQ rozpakowanie z przeplotem młodszych podwójnych słów

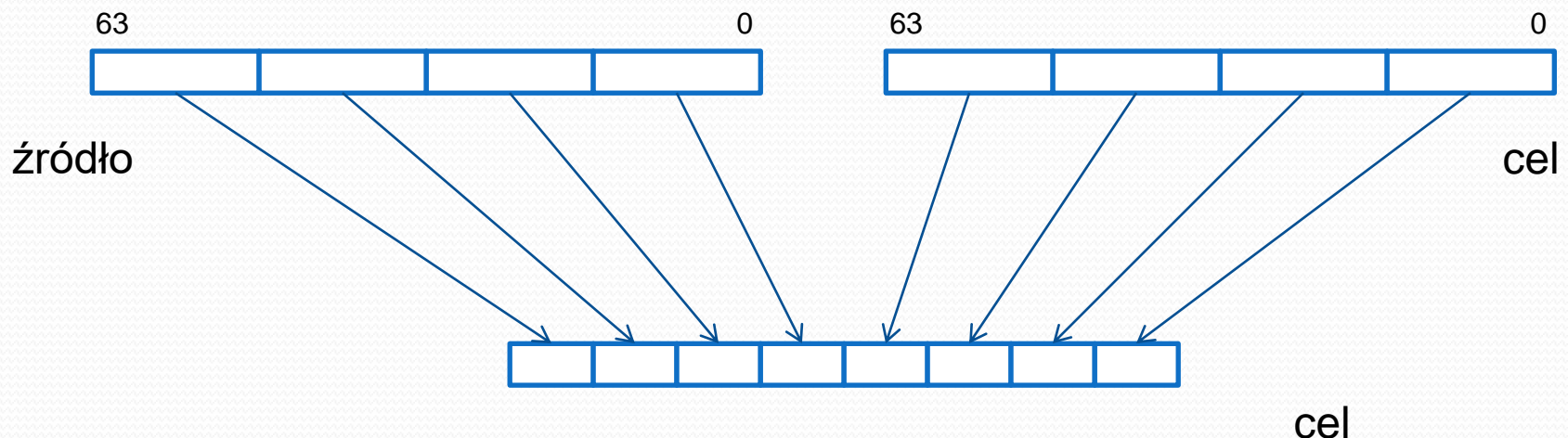
Instrukcja

PACKSSWB/PACKUSWB

PACKSSWB cel, źródło

PACKUSWB cel, źródło

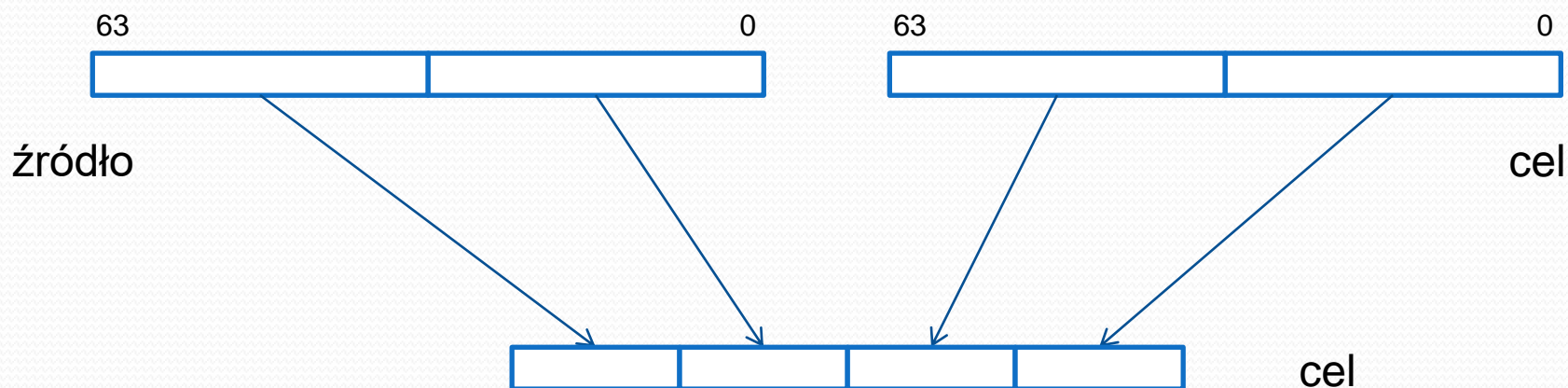
Pakowanie z nasyceniem słów ze znakiem/bez znaku do bajtów. Cel musi być rejestrem mmx.



Instrukcja PACKSSDW

PACKSSDW cel, źródło

Pakowanie z nasyceniem podwójnych słów ze znakiem do słów. Cel musi być rejestrem mmx.

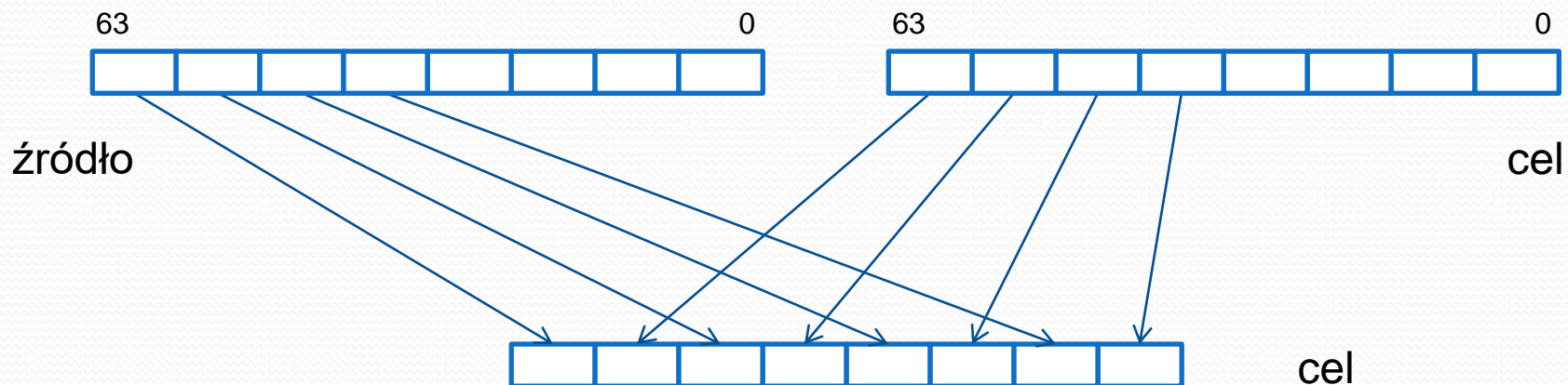


Instrukcje PUNPCKHBW

PUNPCKHWD PUNPCKHDQ

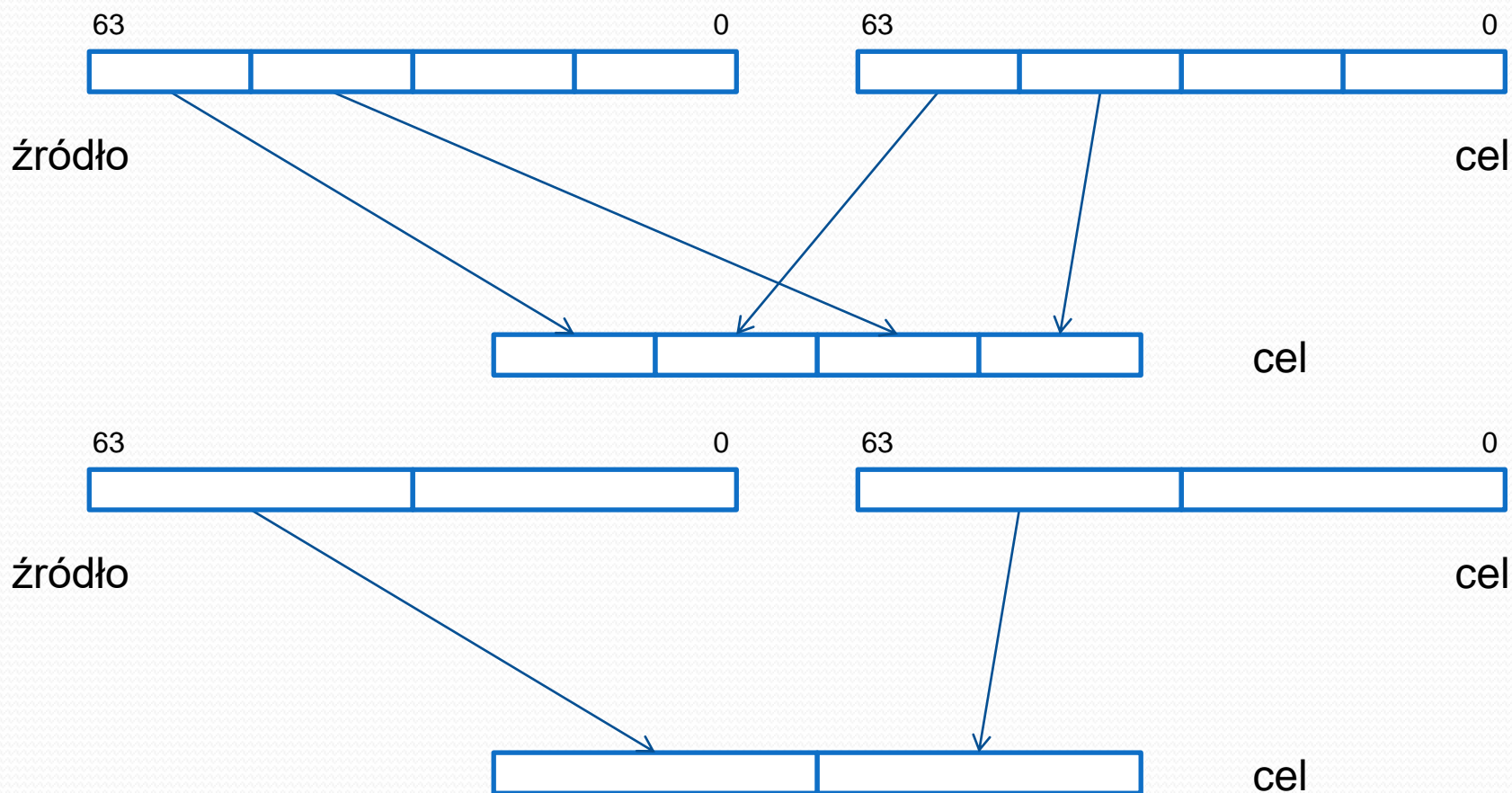
PUNPCKHBW/PUNPCKHWD/PUNPCKHDQ cel, źródło

Rozpakowanie z przeplotem starszych bajtów/słów/
podwójnych słów bez znaku. Cel musi być rejestrem mmx.
Jeśli źródło = 0 to następuje konwersja do słów, podwójnych i
poczwórnego słowa.



Instrukcje

PUNPCKHWD PUNPCKHDQ

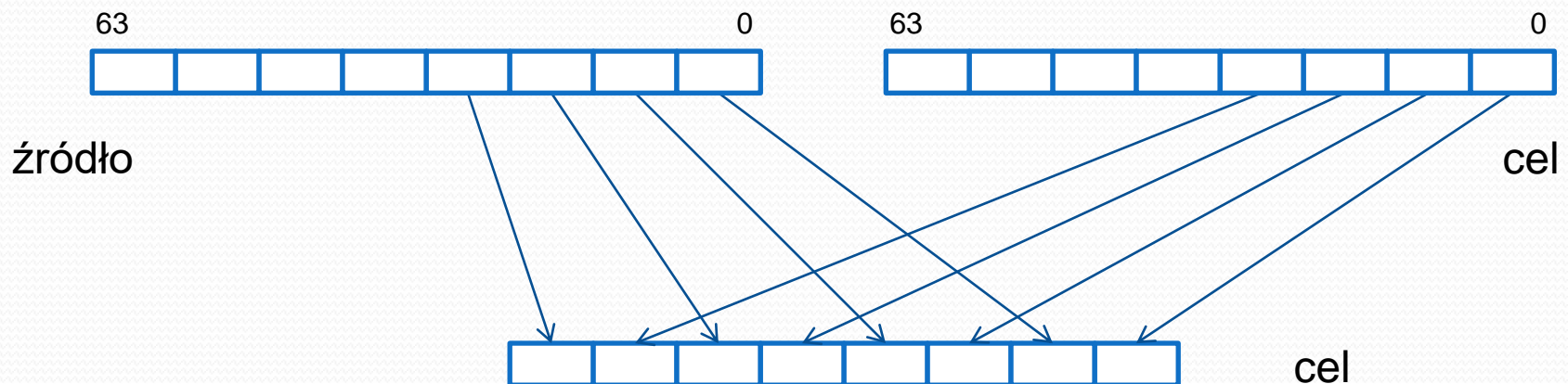


Instrukcje PUNPCKLBW

PUNPCKLWD PUNPCKLDQ

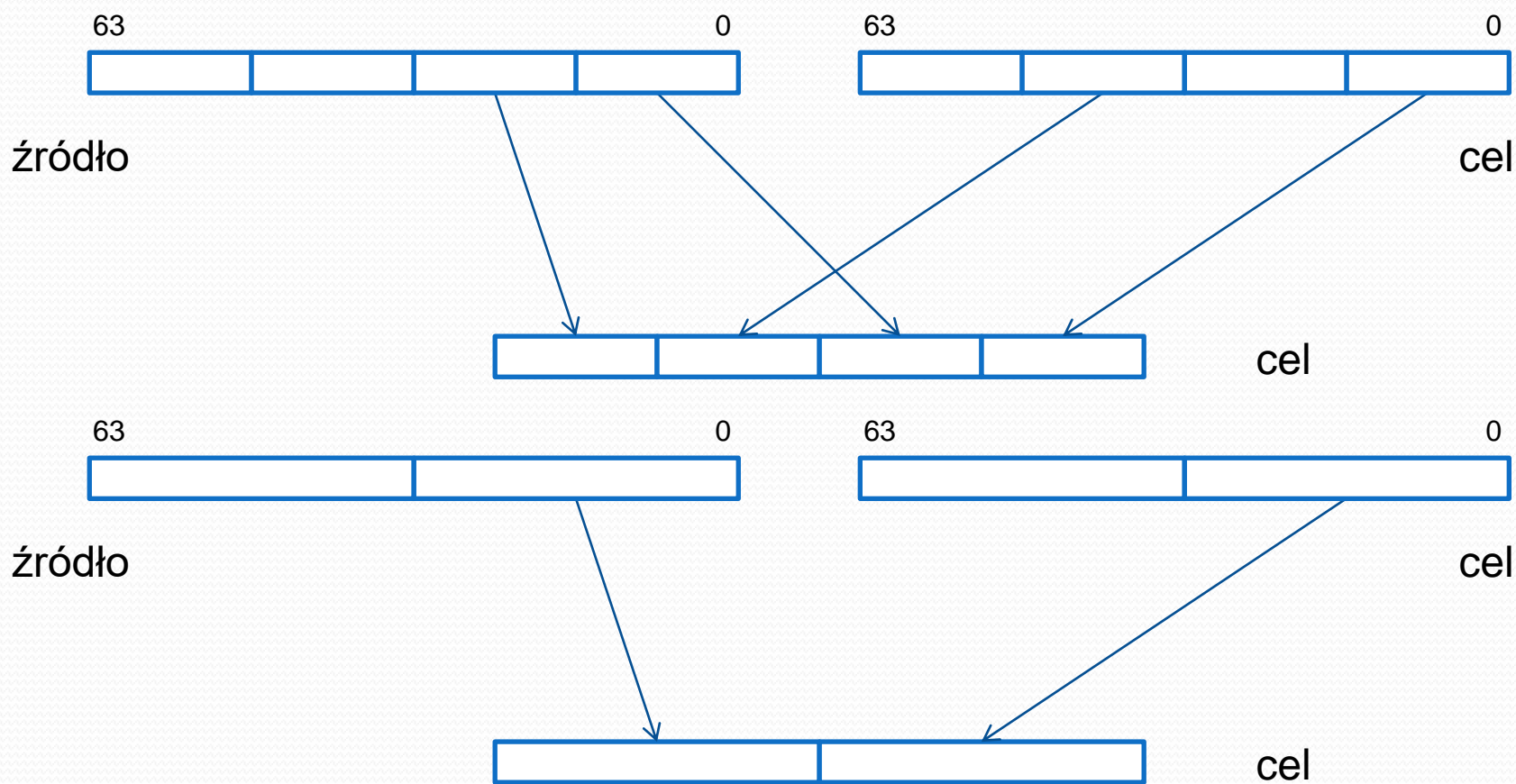
PUNPCKLBW/PUNPCKLWD/PUNPCKLDQ cel, źródło

Rozpakowanie z przeplotem młodszych bajtów/słów/
podwójnych słów bez znaku. Cel musi być rejestrem mmx.
Jeśli źródło = 0 to następuje konwersja do słów, podwójnych i
poczwórnego słowa.



Instrukcje

PUNPCKLWD PUNPCKLDQ



Operacje arytmetyczne

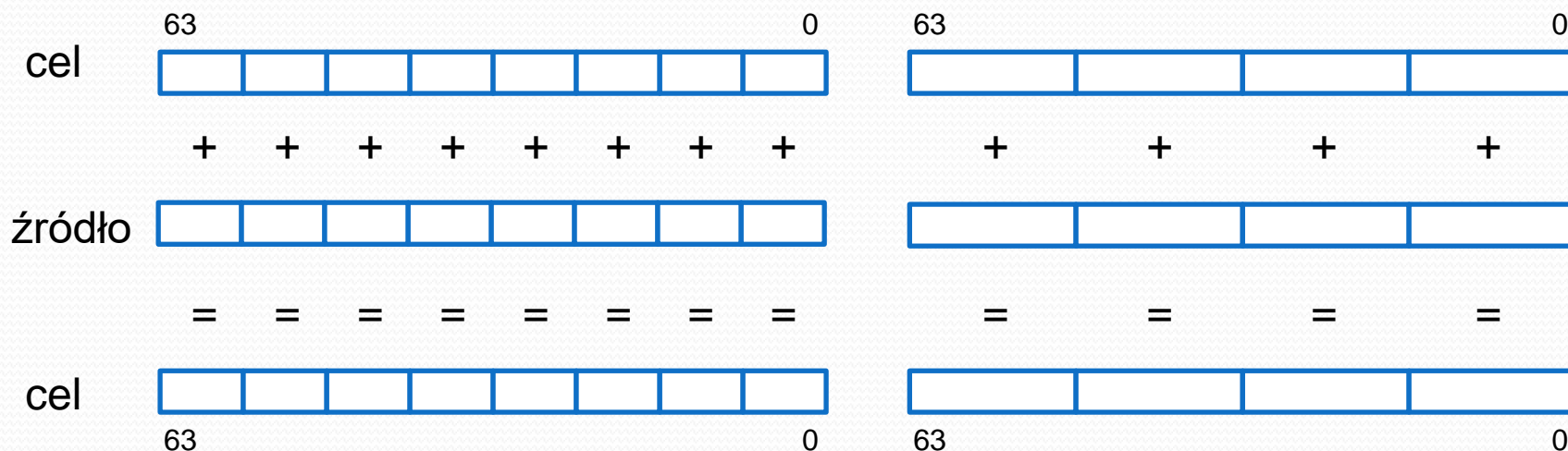
- PADDB dodawanie wektorów bajtów
- PADDW dodawanie wektorów słów
- PADDD dodawanie wektorów podwójnych słów
- PADDSB dodawanie z nasyceniem wektorów bajtów ze znakiem
- PADDSW dodawanie z nasyceniem wektorów słów ze znakiem
- PADDUSB dodawanie z nasyceniem wektorów bajtów bez znaku
- PADDUSW dodawanie z nasyceniem wektorów słów bez znaku

Instrukcje

PADDB PADDW PADDD

PADDB/PADDW/PADDD cel, źródło

Dodawanie wektorów bajtów/słów/podwójnych słów. Cel musi być rejestrem mmx. Wynik nie uwzględnia przeniesienia.

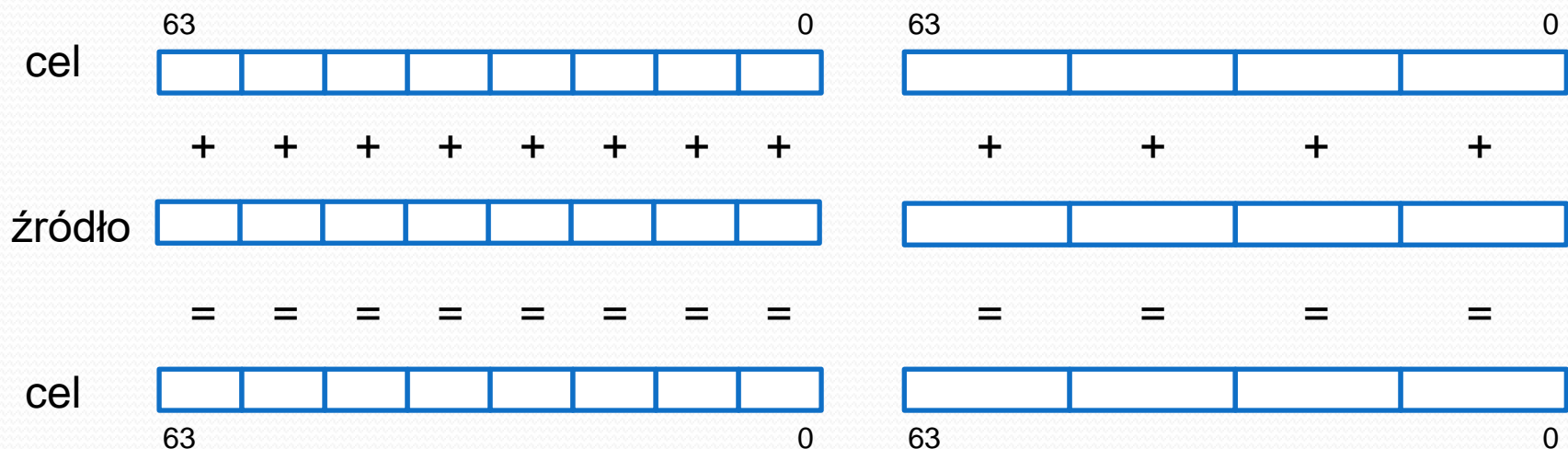


Instrukcje PADDUSB PADDUSW

PADDUSB PADDUSW

PADDUSB/PADDUSW/PADDUSB/PADDUSW cel, źródło

Dodawanie z nasyceniem wektorów bajtów/słów ze znakiem/bez znaku. Cel musi być rejestrem mmx.



Operacje arytmetyczne

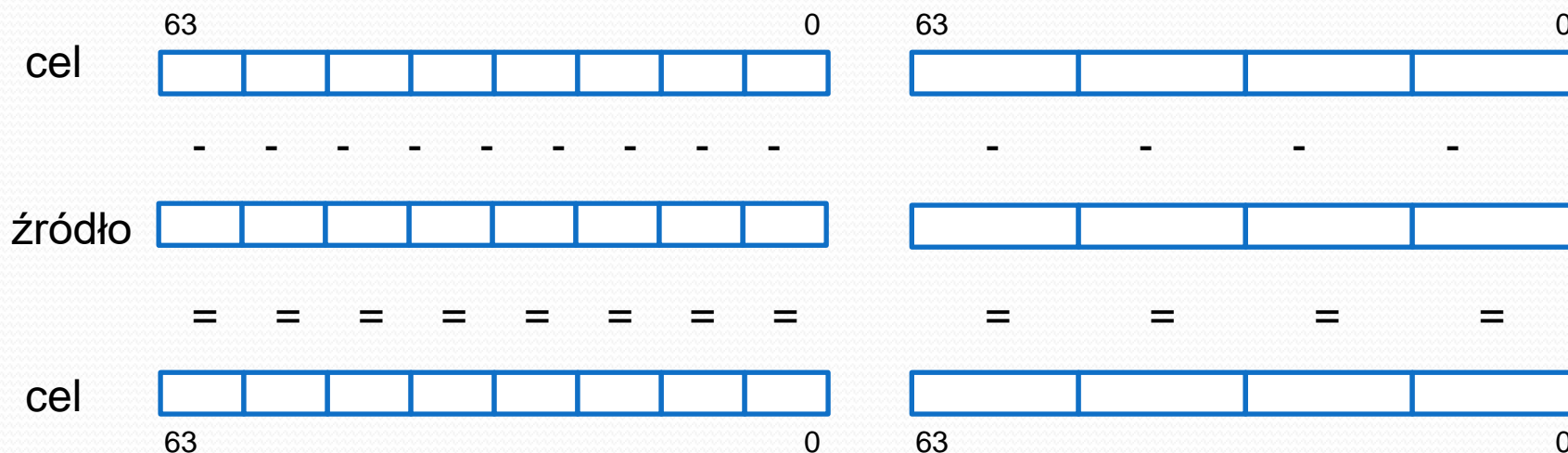
- PSUBB odejmowanie wektorów bajtów
- PSUBW odejmowanie wektorów słów
- PSUBD odejmowanie wektorów podwójnych słów
- PSUBSB odejmowanie z nasyceniem wektorów bajtów ze znakiem
- PSUBSW odejmowanie z nasyceniem wektorów słów ze znakiem
- PSUBUSB odejmowanie z nasyceniem wektorów bajtów bez znaku
- PSUBUSW odejmowanie z nasyceniem wektorów słów bez znaku

Instrukcje

PSUBB PSUBW PSUBD

PSUBB/PSUBW/PSUBD cel, źródło

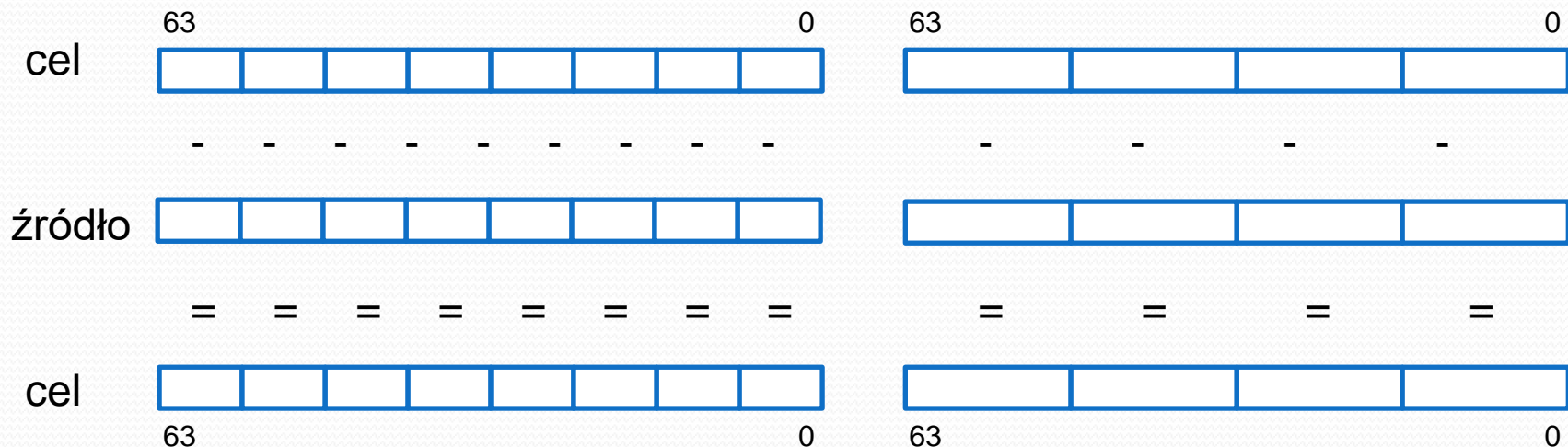
Odejmowanie wektorów bajtów/słów/podwójnych słów. Cel musi być rejestrem mmx. Wynik nie uwzględnia przeniesienia.



Instrukcje PSUBSB PSUBSW PSUBUSB PSUBUSW

PSUBSB/PSUBSW/PSUBUSB/PSUBUSW cel, źródło

Odejmovanie z nasyceniem wektorów bajtów/słów ze znakiem/bez znaku. Cel musi być rejestrem mmx.



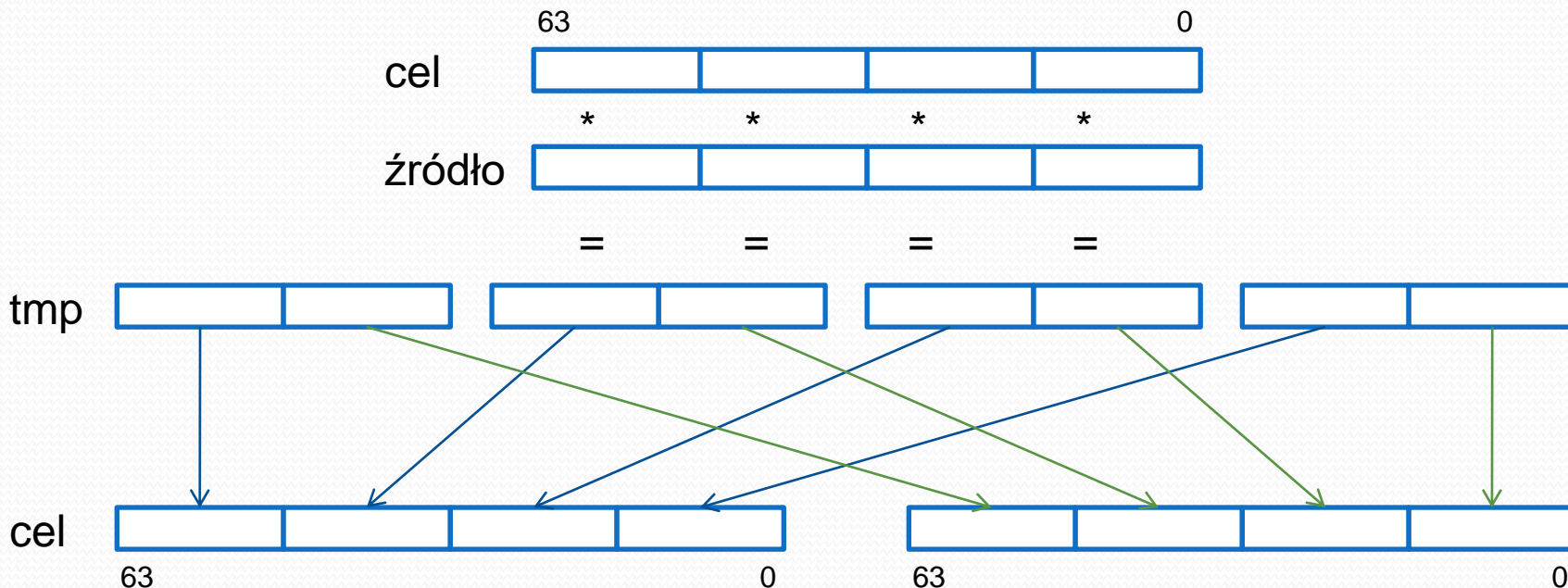
Operacje arytmetyczne

- **PMULHW** mnożenie wektorów słów i zapamiętanie starszych słów wyniku
- **PMULLW** mnożenie wektorów słów i zapamiętanie młodszych słów wyniku
- **PMADDWD** mnożenie i dodawanie wektorów słów

Instrukcje PMULHW PMULLW

PMULHW/PMULLW cel, źródło

Mnożenie wektorów słów i zapamiętanie starszych/młodszych słów wyniku. Cel musi być rejestrem mmx.



Operacje porównania

- PCMPEQB sprawdzenie równości wektorów bajtów
- PCMPEQW sprawdzenie równości wektorów słów
- PCMPEQD sprawdzenie równości wektorów podwójnych słów
- PCMPGTB sprawdzenie większości wektorów bajtów ze znakiem
- PCMPGTW sprawdzenie większości wektorów słów ze znakiem
- PCMPGTD sprawdzenie większości wektorów podwójnych słów ze znakiem

Instrukcje PCMPEQB PCMPEQW PCMPEQD

PCMPEQB/PCMPEQW/PCMPEQD cel, źródło

Sprawdzenie równości składowych wektorów bajtów/słów/podwójnych słów. Cel musi być rejestrem mmx.

```
if cel[i] = źródło[i] then cel[i] := offh/offffh/offffffffh  
else cel[i] := 0
```

Instrukcje PCMPGTB

PCMPGTW PCMPGTD

PCMPGTB/PCMPGTW/PCMPGTD cel, źródło

Sprawdzenie relacji większości składowych wektorów bajtów/słów/podwójnych słów ze znakiem. Cel musi być rejestrem mmx.

```
if cel[i] > źródło[i] then cel[i] := offh/offffh/offffffffh
else cel[i] := 0
```

Operacje logiczne

- PAND bitowy iloczyn logiczny
- PANDN bitowy iloczyn logiczny z negacją
- POR bitowa suma logiczna
- PXOR bitowa suma modulo 2

Instrukcje

PAND PANDN POR PXOR

PAND/PANDN/POR/PXOR cel, źródło

Obliczenie bitowego iloczynu logicznego/bitowego iloczynu logicznego z negacją/bitowej sumy logicznej/bitowej sumy modulo 2. Cel musi być rejestrem mmx.

cel := cel and źródło

cel := (not cel) and źródło

cel := cel or źródło

cel := cel xor źródło

Operacje przesunięć

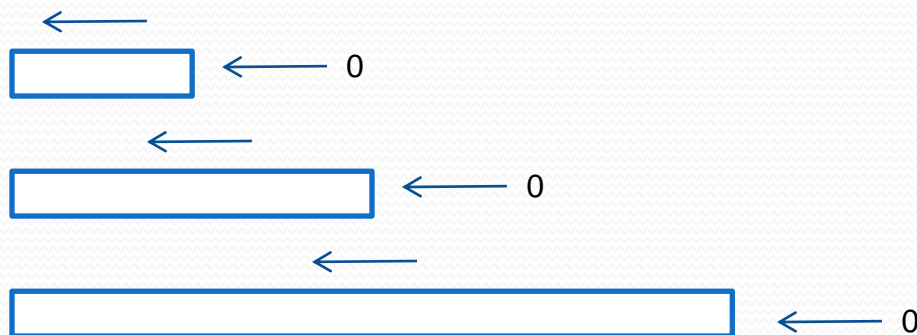
- PSLLW logiczne przesunięcie w lewo wektora słów
- PSLLD logiczne przesunięcie w lewo wektora podwójnych słów
- PSLLQ logiczne przesunięcie w lewo wektora poczwórnych słów
- PSRLW logiczne przesunięcie w prawo wektora słów
- PSRLD logiczne przesunięcie w prawo wektora podwójnych słów
- PSRLQ logiczne przesunięcie w prawo wektora poczwórnych słów
- PSRAW arytmetyczne przesunięcie w prawo wektora słów
- PSRAD arytmetyczne przesunięcie w prawo wektora podwójnych słów

Instrukcje

PSLLW PSLLD PSLLQ

PSLLW/PSLLD/PSLLQ cel, ile

Logiczne przesunięcie w lewo elementów wektora słów/podwójnych słów/poczwórnych słów. Cel musi być rejestrem mmx, ile jest rejestrem mmx, zmienną lub stałą. Młodsze bity wypełniane są zerami.

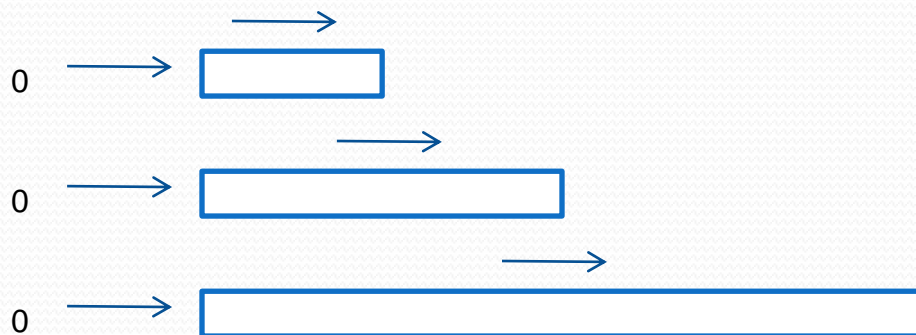


Instrukcje

PSRLW PSRLD PSRLQ

PSRLW/PSRLD/PSRLQ cel, ile

Logiczne przesunięcie w prawo elementów wektora słów/podwójnych słów/poczwórnych słów. Cel musi być rejestrem mmx, ile jest rejestrem mmx, zmienną lub stałą. Starsze bity wypełniane są zerami.

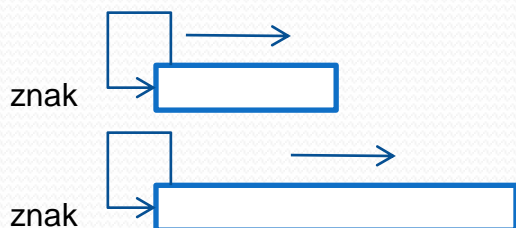


Instrukcje

PSRAW PSRAD

PSRAW/PSRAD cel, ile

Arytmetyczne przesunięcie w prawo elementów wektora słów/podwójnych słów. Cel musi być rejestrem mmx, ile jest rejestrem mmx, zmienną lub stałą. Starsze bity wypełniane są znakiem.



Operacje sterujące

- FXSAVE zapisanie stanu x87 FPU i rejestrów SIMD
- FXRSTOR wczytanie stanu x87 FPU i rejestrów SIMD
- EMMS zwalnia wszystkie rejestry koprocesora
- LDMXCSR wczytanie rejestru MXCSR
- STMXCSR zapisanie rejestru MXCSR

Instrukcje

FXSAVE

FXRSTOR

FXSAVE/FXRSTOR cel512

zapisanie/wczytanie stanu

x87 FPU i rejestrów SIMD

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|------------|----|----|----|---------|----|---|---|----------|---|-----|---|-----|---|-----|---|-----|
| FPU IP | | | | | | | | FOP | | FTW | | FSW | | FCW | | 0 |
| MXCSR_MASK | | | | MXCSR | | | | FPU DP | | | | | | | | 16 |
| Reserved | | | | ST0/MM0 | | | | | | | | 32 | | | | |
| Reserved | | | | ST1/MM1 | | | | | | | | 48 | | | | |
| Reserved | | | | ST2/MM2 | | | | | | | | 64 | | | | |
| Reserved | | | | ST3/MM3 | | | | | | | | 80 | | | | |
| Reserved | | | | ST4/MM4 | | | | | | | | 96 | | | | |
| Reserved | | | | ST5/MM5 | | | | | | | | 112 | | | | |
| Reserved | | | | ST6/MM6 | | | | | | | | 128 | | | | |
| Reserved | | | | ST7/MM7 | | | | | | | | 144 | | | | |
| | | | | | | | | XMM0 | | | | | | | | 160 |
| | | | | | | | | XMM1 | | | | | | | | 176 |
| | | | | | | | | XMM2 | | | | | | | | 192 |
| | | | | | | | | XMM3 | | | | | | | | 208 |
| | | | | | | | | XMM4 | | | | | | | | 224 |
| | | | | | | | | XMM5 | | | | | | | | 240 |
| | | | | | | | | XMM6 | | | | | | | | 256 |
| | | | | | | | | XMM7 | | | | | | | | 272 |
| | | | | | | | | XMM8 | | | | | | | | 288 |
| | | | | | | | | XMM9 | | | | | | | | 304 |
| | | | | | | | | XMM10 | | | | | | | | 320 |
| | | | | | | | | XMM11 | | | | | | | | 336 |
| | | | | | | | | XMM12 | | | | | | | | 352 |
| | | | | | | | | XMM13 | | | | | | | | 368 |
| | | | | | | | | XMM14 | | | | | | | | 384 |
| | | | | | | | | XMM15 | | | | | | | | 400 |
| | | | | | | | | Reserved | | | | | | | | 416 |
| | | | | | | | | Reserved | | | | | | | | 432 |
| | | | | | | | | Reserved | | | | | | | | 448 |
| | | | | | | | | Reserved | | | | | | | | 464 |
| | | | | | | | | Reserved | | | | | | | | 480 |
| | | | | | | | | Reserved | | | | | | | | 496 |

Instrukcja EMMS

EMMS

Zwalnia wszystkie rejestry koprocesora wpisując do pól TAG[i] rejestru stanu zawartości rejestrów stosu wartość 11b (rejestr pusty). Wszystkie instrukcje MMX wpisują do pól TAG[i] 0, co oznacza liczbę prawidłową!

Instrukcje LDMXCSR STMXCSR

LDMXCSR/STMXCSR zmienna

Wczytanie/zapisanie zawartości rejestru MXCSR.

Operacje MMX wprowadzone z SSE

- PAVGB oblicza średnią z elementów wektorów bajtów bez znaku
- PAVGW oblicza średnią z elementów wektorów słów bez znaku
- PEXTRW wydobyć słowa
- PINSRW wstawienie słowa
- PMAXUB oblicza maksimum z elementów wektorów bajtów bez znaku
- PMAXSW oblicza maksimum z elementów wektorów słów ze znakiem
- PMINUB oblicza minimum z elementów wektorów bajtów bez znaku
- PMINSW oblicza minimum z elementów wektorów słów ze znakiem
- PMOVMSKB przesłanie maski bajtów
- PMULHUW mnożenie wektorów słów bez znaku i zapamiętanie starszych słów
wyniku
- PSADBW oblicza sumę wartości bezwzględnych różnic
- PSHUFW tasuje słowa w rejestrze MMX

Instrukcje PAVGB PAVGW

PAVGB/PAVGW cel, źródło

Oblicza średnią z elementów wektorów bajtów/słów bez znaku. Cel jest rejestrem MMX.

$$e_cel := (e_cel + e_źródło + 1) \gg 1$$

Instrukcje PEXTRW PINSRW

PEXTRW cel, źródło, numer

PINSRW cel, źródło, numer

Wydobycie/wstawienie słowa o podanym numerze z/do rejestru MMX do/z rejestru ogólnego przeznaczenia lub pamięci.

Instrukcje PMAXUB PMAXSW PMINUB PMINSW

PMAXUB/PMAXSW/PMINUB/PMINSW cel, źródło

Oblicza maksimum/minimum z elementów wektorów bajtów bez znaku/słów ze znakiem.

$e_cel := e_cel \text{ max/min } e_źródło$

Instrukcja PMOVMSKB

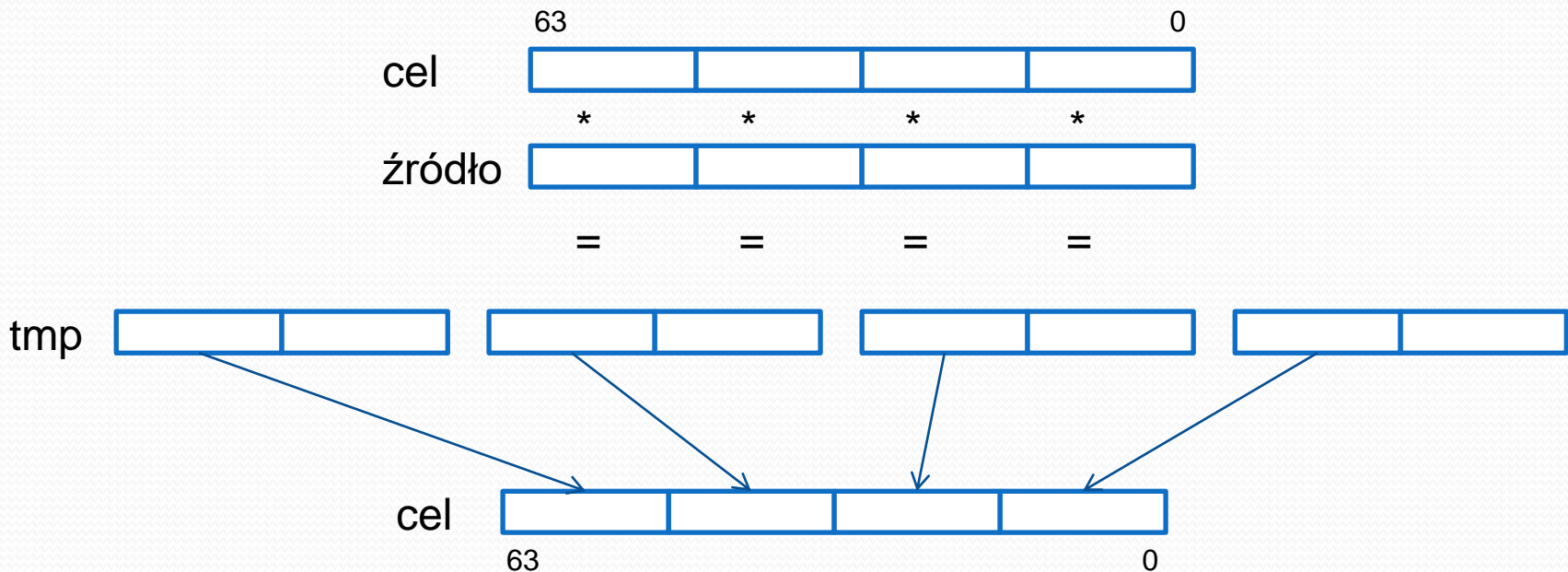
PMOVMSKB cel, źródło

Przesłanie maski bajtów. Celem jest rejestr ogólnego przeznaczenia. Najstarsze bity elementów wektora z rejestru MMX wpisywane są na bity o ..7 celu. Stosowane w celu określenia znaku lub sprawdzenia wyniku porównania.

Instrukcje PMULHUW

PMULHUW cel, źródło

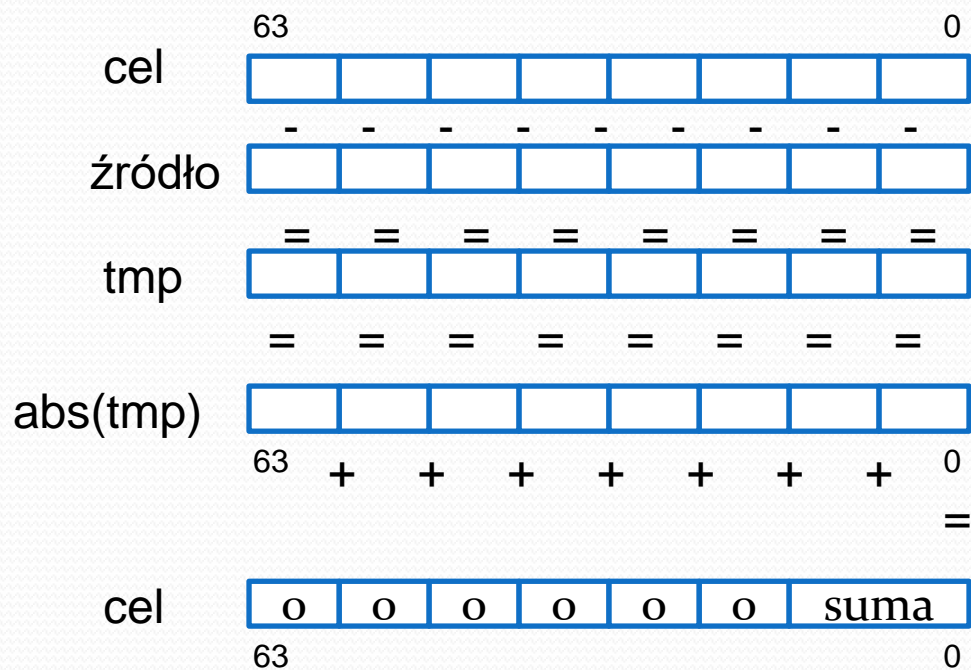
Mnożenie wektorów słów bez znaku i zapamiętanie starszych słów wyniku. Cel musi być rejestrem mmx.



Instrukcja PSADBW

PSADBW cel, źródło

Oblicza sumę wartości bezwzględnych różnic.



Instrukcja PSHUFW

PSHUFW cel, źródło, kolejność

Tasuje słowa w rejestrze celu MMX. Źródłem jest rejestr MMX lub pamięć. Bity 7,6; 5,4 ;3,2 i 1,0 stałej kolejność określają numer słowa w źródle, które zostanie umieszczone jako 3,2,1 i 0 w rejestrze celu np.

dla źródła= **0123 4567 89ab cdefh** i kolejności 00 01 10 11b
będzie: cel=**ocdef 89ab 4567 0123h**