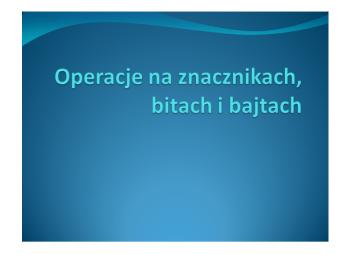
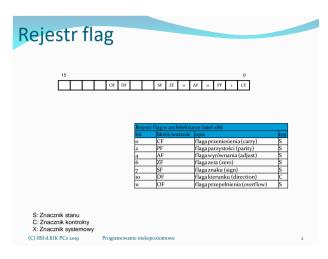
2019



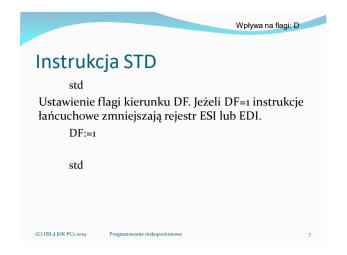


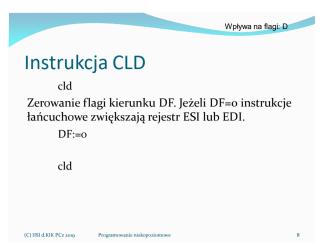






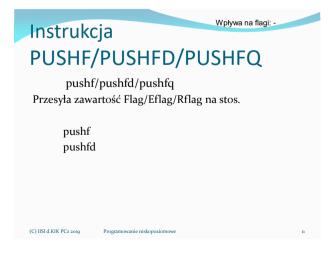




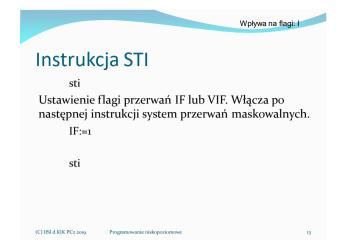


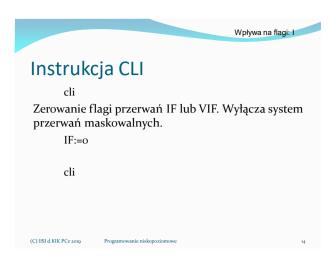




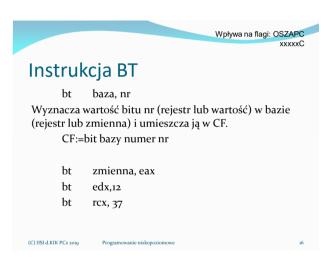


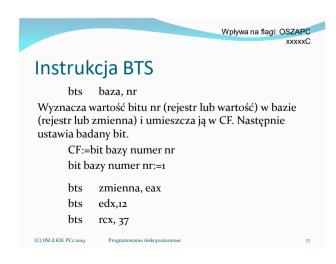


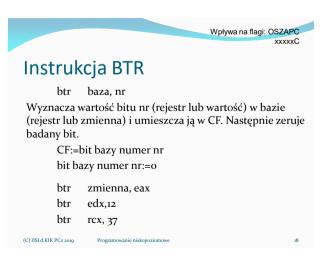


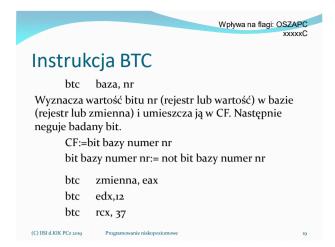


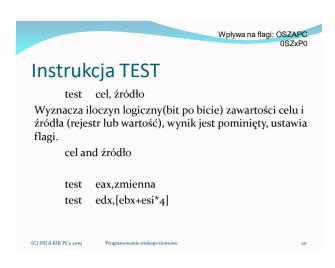
## Operacje na bitach Testowanie bitu BTS Testowanie bitu z ustawianiem BTR Testowanie bitu z zerowaniem BTC Testowanie bitu z negacją TEST Porównanie logiczne BSF Przeszukiwanie bitów w przód BSR Przeszukiwanie bitów wstecz LZCNT Zlicza zerowe bity od najstarszego TZCNT Zlicza zerowe bity od najmłodszego BEXTR Wycina ciąg bitów BLSI Kopiuje najmłodszy ustawiony bit BLSR Zeruje najmłodszy ustawiony bit BLSMSK Tworzy maskę do bitu=o BZHI Zeruje starsze bity





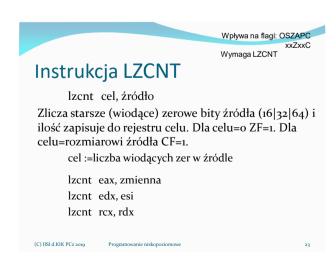






Wpływa na flagi: OSZAPC Instrukcja BSF cel, źródło bsf Przeszukiwanie bitów w przód. Szuka w rejestrze lub zmiennej źródła najmłodszego bitu=1, jego indeks umieszcza w rejestrze celu (ZF=o). Jeśli źródło=o, wówczas ZF=1, a cel jest niezdefiniowany cel :=indeks najmłodszego bitu=1 źródła bsf eax,zmienna bsf edx,esi bsf rcx, rdx (C) IISI d.KIK PCz 2019

Wpływa na flagi: OSZAPC Instrukcja BSR cel, źródło bsr Przeszukiwanie bitów wstecz. Szuka w rejestrze lub zmiennej źródła najstarszego bitu=1, jego indeks umieszcza w rejestrze celu (ZF=o). Jeśli źródło=o, wówczas ZF=1, a cel jest niezdefiniowany cel :=indeks najstarszego bitu=1 źródła eax, zmienna bsr edx, esi bsr bsr rcx, rdx (C) IISI d.KIK PCz 2016





Wpływa na flagi: OSZAPC
OxZxxO
Wymaga BM1

Instrukcja BEXTR

bextr cel, źródło, st\_ile

Wycina z rejestru|zmiennej źródła (32|64) ciąg bitów i umieszcza w rejestrze celu. Początkowy bit określa rejestr st\_ile[7:0], a ilość bitów st\_ile[15:8]. Jeśli cel=o, wówczas ZF=1.

cel :=źródło[start+ile-1:start]

bextr eax, zmienna, edx
bextr edx, esi, eax
bextr rcx, rdx, rax

Mpływa na flagi: OSZAPC OSZXXC

Mymaga BM11

Instrukcja BLSI

blsi cel, źródło

Izoluje z rejestru lub zmiennej źródła najmłodszy bit=1 i umieszcza w rejestrze celu (CF=1). Zeruje pozostałe bity. Jeśli źródło=0, wówczas CF=0, a cel=0. cel :=(-źródło) and źródło

blsi eax, zmienna blsi edx, esi blsi rcx, rdx

Wymaga BM11

Instrukcja BLSR

blsr cel, źródło

Kopiuje bity z rejestru lub zmiennej źródła (32 | 64) i umieszcza w rejestrze celu, zeruje najmłodszy bit=1 (CF=0). Jeśli źródło=0, wówczas CF=1, a cel=0. cel:=(źródło-1) and źródło

blsr eax, zmienna blsr edx, esi blsr rcx, rdx

Wpływa na flagi: OSZAPC 0S0xxC Wymaga BMI1 Instrukcja BLSMSK blsmsk cel, źródło Ustawia młodsze bity rejestru celu (32 | 64) na 1 aż do numeru najmłodszego bitu=1 z rejestru lub zmiennej źródła włącznie (CF=o). Zeruje pozostałe bity. Jeśli źródło=o, wówczas CF=1, a cel=not o. cel :=(źródło-1) xor źródło blsmsk eax, zmienna hlsmsk edx, esi blsmsk rcx, rdx (C) IISI d.KIK PCz 2019

Wpływa na flagi: OSZAPC OSZXXC

Instrukcja BZHI

bzhi cel, źródło, idx

Kopiuje bity z rejestru lub zmiennej źródła (32|64) do rejestru celu (32|64) i kasuje starsze bity od numeru z rejestru idx (CF=0). Jeśli idx>31|63, wówczas CF=1. cel:=źródło; cel[rozmiar-1:idx]=0

bzhi eax, zmienna, edx
bzhi edx, esi, eax
bzhi rcx, rdx, rax

Instrukcja SETcc

SETcc cel

Jeśli jest spełniony warunek cc, ustawia bajt na 1, w przeciwnym wypadku na o.

if cc then cel:=1
 else cel:=0

sets al setge [esi+8]

## Instrukcje SETcc Ustaw bajt jeśli equal/ zero SETNE/SETNZ Ustaw bajt jeśli not equal/ not zero SETS Ustaw bajt jeśli sign (negative) SETNS Ustaw bajt jeśli not sign (non-negative) Ustaw bajt jeśli overflow SETO SETNO Ustaw bait jeśli not overflow SETPE/SETP Ustaw bajt jeśli parity even/ parity SETPO/SETNP Ustaw bajt jeśli parity odd/ not parity SETA/SETNBE Ustaw bajt jeśli above/ not below or equal SETAE/SETNB/SETNC Ustaw bajt jeśli above or equal/ not below/ not carry SETB/SETNAE/SETC Ustaw bajt jeśli below/ not above or equal/ carry SETBE/SETNA Ustaw bajt jeśli below or equal/ not above SETG/SETNLE Ustaw bajt jeśli greater/ not less or equal SETGE/SETNL Ustaw bajt jeśli greater or equal/ not less SETL/SETNGE Ustaw bajt jeśli less/ not greater or equal SETLE/SETNG Ustaw bajt jeśli less or equal/ not greater (C) IISI d.KIK PCz 2019

```
Przykład — int na bin(string)

procedure szb(var s-stringi-integer);
asm

push ebx
br eckcex: //w eck nr najstarszej i
jinz @i
mow word ptr [eax],$3001
jmp @e
@i: inc edx
mow [eax],dl //długość
dec edx
inc eax
@p. bt ecx.edx: //testuj
set bl
add bl.$30
mow [eax],bl //tzapisz znak
inc eax
dec edx
jins @p
@e: po pe bx
end;

(C) IISI d.KIK PCz 2019

Programowanie niskopoziomowe

Programowanie niskopoziomowe

procedure szbz(var sztringi-integer);
asm
br eck.ecx: //w eck nr najstarszej i
jins @p
@e: mow word ptr [eax],$300
jins @p
@e: end;
```

```
Instrukcja RDPID

rdpid cel
Czyta 32-bitowy identyfikator procesora do rejestru celu.
cel=PID

rdpid eax
rdpid rdx
```

```
Instrukcja RDTSC

rdtsc

Read Time Stamp Counter. Czyta 64-bitowy licznik do rejestrów EDX:EAX.
EDX:EAX:=licznik
rdtsc
```

```
Przykład – funkcja pomiaru czasu

function Pomiar(a:integer):integer;
var Cykle_H,Cykle_L:integer;
asm
rdtsc
mov Cykle_H,edx
mov Cykle_L,eax
...
...
...
rdtsc
sub eax,Cykle_L
sbb edx,Cykle_L
sbb edx,CykleH
sub EAX,9 ; odliczenie 9?
end;
(C) IISI d.KIK PCz 2010
```

