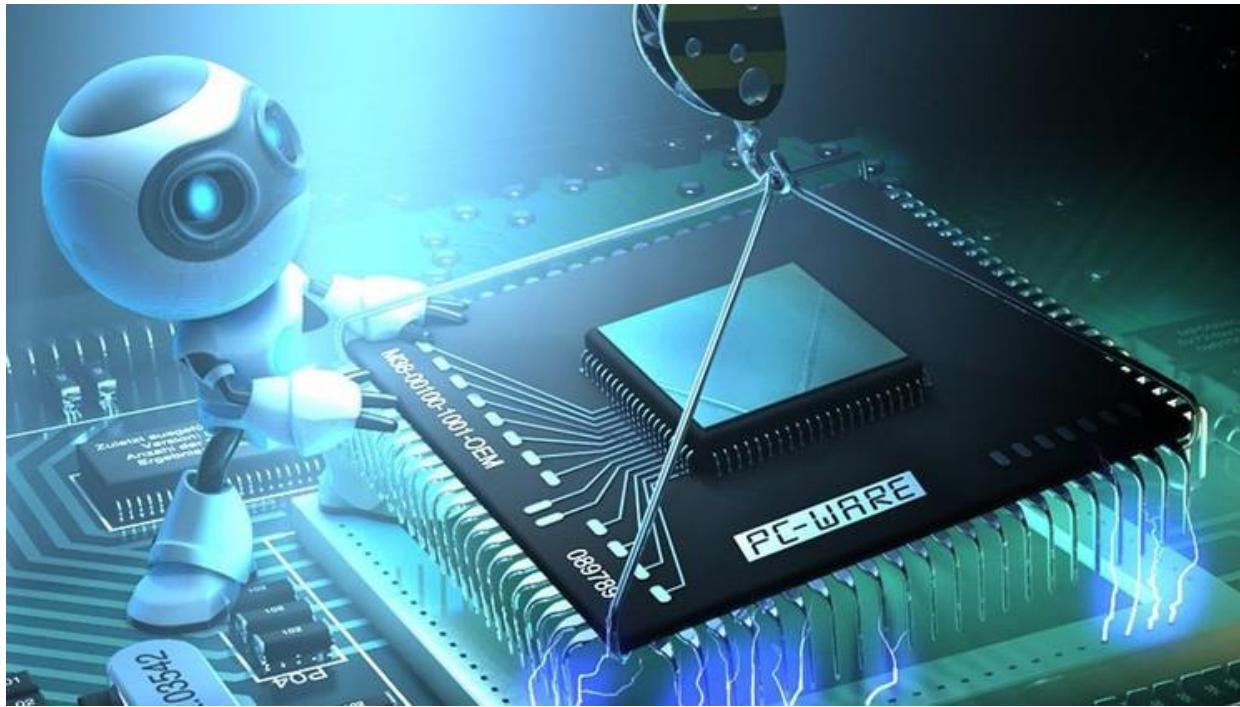


# Advanced Logical Instructions



# Advanced Logical Instructions



Instructions	Description	Notes	Example
<b>ANDx op1, op2</b>	$op2 \leftarrow op2 \& op1$	$x = \{L, W, B\}$	ANDL \$13,%EAX
<b>ORx op1, op2</b>	$op2 \leftarrow op2   op1$	$x = \{L, W, B\}$	ORW %CX,%AX
<b>XORx op1, op2</b>	$op2 \leftarrow op2 ^ op1$	$x = \{L, W, B\}$	XORL %EDX,%EAX
<b>NOTx op1</b>	$op1 \leftarrow \sim op1$	$x = \{L, W, B\}$	NOTB %AH
<b>SALx k,op1</b>	$op1 \leftarrow op1 \ll k$ (aritm.)	$x = \{L, W, B\}$ , k: imm. o %CL	SALL \$1,%EAX
<b>SHLx k,op1</b>	$op1 \leftarrow op1 \ll k$ (log.)	$x = \{L, W, B\}$ , k: imm. o %CL	SHLW %CL,%DX
<b>SARx k,op1</b>	$op1 \leftarrow op1 \gg k$ (aritm.)	$x = \{L, W, B\}$ , k: imm. o %CL	SARL \$1,%EAX
<b>SHRx k,op1</b>	$op1 \leftarrow op1 \gg k$ (log.)	$x = \{L, W, B\}$ , k: imm. o %CL	SHRW %CL,%DX
<b>CMPx op1, op2</b>	$op2 - op1$	$x = \{L, W, B\}$ , activa flags	CMPL \$13,%EAX
<b>TESTx op1, op2</b>	$op2 \& op1$	$x = \{L, W, B\}$ , activa flags	TESTW %CX,%AX



# SAL Example

SAL \$4,  $\times AH$

$AH \leftarrow AH \ll 4$

1100 0000

$\times AH$

~~1100~~ 1100 0000

# SAR Example



SAR \$3, ∵ CH

CH ← CH >> 3 (Arit)

11111110

↑ ∵ CH

11111110 000000

# SHR Example



SHR \$3, > CH

00011110
----------

  
> CH

CH ← CH >> 3 (Log)

Right Logical Shift

00011111 000000



# Summary

SHL }    / / / . . . 0 0 0  
SAL }

SHR } 0 0 0 . . . / / /  
      }

SAR } 1 1 1 . 1 . . / / /  
      } 0 0 0 0 . . / / /

# Programmer Vision



## ❑ Available registers

32 bits	16 bits	8 bits	
%eax	%ax	%ah , %al	
%ebx	%bx	%bh , %bl	
%ecx	%cx	%ch , %cl	
%edx	%dx	%dh , %dl	
%esi	%si		
%edi	%di		
%esp	%sp		Reserved for specific use of subroutines
%ebp	%bp		
%eip			Program counter
%eflags			Status word