Add two operands and store the result. SUB Subtract two operands and store the result. CMP Compare two operands and set the appropriate flags. JMP Unconditionally jump to a new location in the code. JZ/JNZ Jump if the zero flag is set or not set, respectively. JEJNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. AND Perform a bitwise AND operation on two operands. OR Perform a bitwise OR operation on two operands. Perform a bitwise OR operation on an operand. Shift the bits of an operand left by a Shift sthe contents of the EAX register. Sub eaX, ebX subtracts the contents of the EAX register. add eax, 10 adds the immediate value 10 to contents of the EAX register. Sub eaX, ebX subtracts the contents of the EAX register. Sub eaX, ebX subtracts the contents of the EAX register may be ax, ebx compares the contents of the EAX register on the code label. JZ/JNZ Jump if the zero flag is set or not set, jz label jumps to the code label if the zero flags. JE Jabel jumps to the code label if the operand are equal. Call my_function calls the function named my_function. Pet returns from the current function. Pos eax pushes the contents of the EAX register onto the EaX register. Xor eax, ebx performs a bitwise XOR operation on the contents of the EAX and EBX registers. NOT ereary ebx performs a bitwise OR operation on the contents of the EAX and EBX registers. NOT ereary ebx performs a bitwise OR operation on the contents of the EAX register. Shift the bits of an operand left by a Shift the purp of the EAX and the EAX register. Shift the bits of an operand left by a Shift the contents of the EAX register.	Instruction	Definition	Usage Example
SUB Subtract two operands and store the result. CMP Compare two operands and set the appropriate flags. JMP Unconditionally jump to a new location in the code. JZ/JNZ Jump if the zero flag is set or not set, respectively. JE/JNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. AND Perform a bitwise AND operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise NOT operation on an operand. Shift the bits of an operand left by a Shil eax. 1 shifts the contents of the EAX register. Shift the bits of an operand left by a Shil eax. 1 shifts the contents of the EAX sund EBX register.	MOV		mov eax, ebx moves the contents of the EBX register into the EAX register.
result. Compare two operands and set the appropriate flags. JMP Unconditionally jump to a new location in the code. JZ/JNZ Jump if the zero flag is set or not set, respectively. JE/JNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise NOT operation on two operand. Shift the bits of an operand left by a Shift eax. 1 shifts the contents of the EAX register. Cmp eax, ebx compares the contents of the EAX registers and sets the appropriate flags. Cmp eax, ebx compares the contents of the EAX and EBX registers. Cmp eax, ebx compares the contents of the EAX and EBX registers. Cmp eax, ebx compares the contents of the EAX and EBX registers. Cmp eax, ebx compares the contents of the EAX and EBAX registers. Cmp eax, ebx compares the contents of the EAX and EBAX register. Cmp eax, ebx compares the contents of the EAX and EBAX register. Cmp eax, ebx compares the contents of the EAX and EBAX register. Cmp eax, ebx performs a bitwise function flags. Imp label jumps to the code label if the zero flags. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero flags set. Je label jumps to the code label if the zero fl	ADD	±	add eax, 10 adds the immediate value 10 to the contents of the EAX register.
Compare two operands and set the appropriate flags. JMP Unconditionally jump to a new location in the code. JZ/JNZ Jump if the zero flag is set or not set, jz label jumps to the code label if the zero flag is set. JEJNE Jump if two operands are equal or not equal, respectively. JEJNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. Shift the bits of an operand left by a Shil eax. 1 shifts the contents of the EAX register. Shift the bits of an operand left by a Shil eax. 1 shifts the contents of the EAX register.	SUB	<u>=</u>	sub eax, ebx subtracts the contents of the EBX register from the EAX register.
JZ/JNZ Jump if the zero flag is set or not set, jz label jumps to the code label if the zero flag is set. JE/JNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. AND Perform a bitwise AND operation on two operands. OR Perform a bitwise OR operation on two operands. Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. NOT Perform a bitwise NOT operation on an operand. Shift the bits of an operand left by a Shill eax. 1 shifts the contents of the EAX of the EAX of the EAX into the contents of the EAX register. Shift the bits of an operand left by a Shill eax. 1 shifts the contents of the EAX of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX of the EAX is the contents of the EAX	СМР	± ±	EAX and EBX registers and sets the appropriate
respectively. JE/JNE Jump if two operands are equal or not equal, respectively. CALL Call a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. Perform a bitwise OR operation on the contents of the EAX and EBX registers. OR Perform a bitwise OR operation on the contents of the EAX and EBX registers. OR Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX registers. NOT Perform a bitwise NOT operation on the contents of the EAX registers. Shift the bits of an operand left by a shift the contents of the EAX register.	JMP		jmp label jumps to the code label.
CALL Call a subroutine or function. RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise NOT operation on an operand. Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. Shift the bits of an operand left by a Shill eax. 1 shifts the contents of the EAX Shill eax. 1 shifts the contents of the EAX Call my_function calls the function named my_function. Push eax pushes the contents of the EAX register onto the stack. Pop ebx pops the top value from the stack into the EBX register. Xor eax, ebx performs a bitwise XOR operation on the contents of the EAX and EBX registers. OR Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Shift the bits of an operand left by a Shill eax. 1 shifts the contents of the EAX	JZ/JNZ		
RET Return from a subroutine or function. PUSH Push a value onto the stack. POP Pop a value from the stack into a register. XOR Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. Perform a bitwise NOT operation on two operands. Perform a bitwise NOT operation on the contents of the EAX and EBX registers. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on the contents of the EAX and EBX registers. OR Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX registers. NOT Shift the bits of an operand left by a Shl eax. 1 shifts the contents of the EAX Shift the bits of an operand left by a Shl eax. 1 shifts the contents of the EAX OR NOT OR NOT OR Return from a subroutine or function. Push eax pushes the contents of the EAX register onto the stack. Pop ebx pops the top value from the stack into the EBX register. NOR AND operation on the contents of the EAX and EBX registers. OR OR Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX register.	JE/JNE	1 1	je label jumps to the code label if the operands are equal.
PUSH Push a value onto the stack. POP Pop a value from the stack into a register. Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. Shift the bits of an operand left by a Shill eax. I shifts the contents of the EAX Tegister. Pop ebx pops the top value from the stack into the EBX register. XOR Pop ebx pops the top value from the stack into the EBX register. XOR Operation on the contents of the EAX and EBX registers. OR Perform a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise OR operation on two operands. Shift the bits of an operand left by a Shill eax. I shifts the contents of the EAX	CALL	Call a subroutine or function.	call my_function calls the function named my_function.
POP Push a value onto the stack. Pop a value from the stack into a register. Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operands. Perform a bitwise NOT operation on two operands. Perform a bitwise NOT operation on two operands. Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers.	RET	Return from a subroutine or function.	ret returns from the current function.
register. XOR Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. OR Perform a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. NOT Shift the bits of an operand left by a Shill eax, ill shifts the contents of the EAX.	PUSH	Push a value onto the stack.	
AND Perform a bitwise exclusive OR operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise AND operation on two operands. Perform a bitwise OR operation on two operation on two operands. Perform a bitwise OR operation on two operation on two operands. Perform a bitwise OR operation on two operation on two operands. Perform a bitwise OR operation on two operation on two operands. Perform a bitwise OR operation on the contents of the EAX and EBX registers. Perform a bitwise NOT operation on two operands. Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise AND operation on the contents of the EAX and EBX registers. NOT Perform a bitwise AND operation on the contents of the EAX and EBX registers.	POP	-	pop ebx pops the top value from the stack into the EBX register.
AND two operation on two operands. OR Perform a bitwise OR operation on two operation on two operands. Or eax, ebx performs a bitwise OR operation on the contents of the EAX and EBX registers. Perform a bitwise OR operation on two operands. Or eax, ebx performs a bitwise OR operation on the contents of the EAX and EBX registers. NOT Perform a bitwise NOT operation on the contents of the EAX and EBX registers. Shift the bits of an operand left by a Shl eax. 1 shifts the contents of the EAX	XOR		operation on the contents of the EAX and EBX
two operands. On the contents of the EAX and EBX registers. Perform a bitwise NOT operation on not eax performs a bitwise NOT operation on an operand. Shift the bits of an operand left by a Shl eax. 1 shifts the contents of the EAX.	AND	-	operation on the contents of the EAX and EBX
an operand. the contents of the EAX register. Shift the bits of an operand left by a Shl eax. 1 shifts the contents of the EAX	OR	-	or eax, ebx performs a bitwise OR operation on the contents of the EAX and EBX registers.
Shift the bits of an operand left by a Shill eax. I shifts the contents of the EAX	NOT		
SHL specified number of bits. register left by one bit.	SHL	Shift the bits of an operand left by a specified number of bits.	shl eax, 1 shifts the contents of the EAX register left by one bit.
SHR Shift the bits of an operand right by a shr ebx, 3 shifts the contents of the EBX register right by three bits.	SHR		