

x86 Instruction Set Reference

BT

Bit Test

Opcode	Mnemonic	Description
0F A3	BT r/m16, r16	Store selected bit in CF flag
0F A3	BT r/m32, r32	Store selected bit in CF flag
0F BA /4 ib	BT r/m16, imm8	Store selected bit in CF flag
0F BA /4 ib	BT r/m32, imm8	Store selected bit in CF flag

Description
<p>Selects the bit in a bit string (specified with the first operand, called the bit base) at the bitposition designated by the bit offset operand (second operand) and stores the value of the bit in the CF flag. The bit base operand can be a register or a memory location; the bit offset operand can be a register or an immediate value. If the bit base operand specifies a register, the instruction takes the modulo 16 or 32 (depending on the register size) of the bit offset operand, allowing any bit position to be selected in a 16- or 32-bit register, respectively. If the bit base operand specifies a memory location, it represents the address of the byte in memory that contains the bit base (bit 0 of the specified byte) of the bit string. The offset operand then selects a bit position within the range -2^{31} to $2^{31} - 1$ for a register offset and 0 to 31 for an immediate offset.</p> <p>Some assemblers support immediate bit offsets larger than 31 by using the immediate bit offset field in combination with the displacement field of the memory operand. In this case, the loworder 3 or 5 bits (3 for 16-bit operands, 5 for 32-bit operands) of the immediate bit offset are stored in the immediate bit offset field, and the high-order bits are shifted and combined with the byte displacement in the addressing mode by the assembler. The processor will ignore the high order bits if they are not zero.</p> <p>When accessing a bit in memory, the processor may access 4 bytes starting from the memory address for a 32-bit operand size, using by the following relationship:</p> <p>Effective Address + (4 * (BitOffset / 32))</p> <p>Or, it may access 2 bytes starting from the memory address for a 16-bit operand, using this relationship:</p> <p>Effective Address + (2 * (BitOffset / 16))</p> <p>It may do so even when only a single byte needs to be accessed to reach the given bit. When using this bit addressing mechanism, software should avoid referencing areas of memory close to address space holes. In particular, it should avoid references to memory-mapped I/O registers.</p> <p>Instead, software should use the MOV instructions to load from or store to these addresses, and use the register form of these instructions to manipulate the data.</p>

Operation
CF = Bit(BitBase, BitOffset);

Flags affected
The CF flag contains the value of the selected bit. The OF, SF, ZF, AF, and PF flags are undefined.

Protected Mode Exceptions	
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit. If the DS, ES, FS, or GS register contains a null segment selector.
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit. If the DS, ES, FS, or GS register contains a null segment selector.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.

Real-Address Mode Exceptions
#GP If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#GP If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

Virtual-8086 Mode Exceptions	
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#GP(0)	If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.
#SS(0)	If a memory operand effective address is outside the SS segment limit.
#PF(fault-code)	If a page fault occurs.