PUSH—Push Word, Doubleword or Quadword Onto the Stack

Opcode*	Instruction	64-Bit Mode	Compat/ Leg Mode	Description
FF /6	PUSH *r/m16*	Valid	Valid	Push *r/m16.*
FF /6	PUSH *r/m32*	N.E.	Valid	Push *r/m32.*
FF /6	PUSH *r/m64*	Valid	N.E.	Push *r/m64.* Default operand size 64-bits.
50+*rw*	PUSH *r16*	Valid	Valid	Push *r16.*
50+*rd*	PUSH *r32*	N.E.	Valid	Push *r32.*
50+*rd*	PUSH *r64*	Valid	N.E.	Push *r64.* Default operand size 64-bits.
6A	PUSH *imm8*	Valid	Valid	Push sign-extended *imm8. Stack pointer is incremented by the size of stack pointer.*
68	PUSH *imm16*	Valid	Valid	Push sign-extended *imm16. Stack pointer is incremented by the size of stack pointer.*
68	PUSH *imm32*	Valid	Valid	Push sign-extended *imm32. Stack pointer is incremented by the size of stack pointer.*
0E	PUSH CS	Invalid	Valid	Push CS.
16	PUSH SS	Invalid	Valid	Push SS.
1E	PUSH DS	Invalid	Valid	Push DS.
06	PUSH ES	Invalid	Valid	Push ES.
0F A0	PUSH FS	Valid	Valid	Push FS and decrement stack pointer by 16 bits.
0F A0	PUSH FS	N.E.	Valid	Push FS and decrement stack pointer by 32 bits.
0F A0	PUSH FS	Valid	N.E.	Push FS. Default operand size 64-bits. (66H override causes 16-bit operation).
0F A8	PUSH GS	Valid	Valid	Push GS and decrement stack pointer by 16 bits.
0F A8	PUSH GS	N.E.	Valid	Push GS and decrement stack pointer by 32 bits.

Opcode*	Instruction	64-Bit Mode	Compat/ Leg Mode	Description
0F A8	PUSH GS	Valid	N.E.	Push GS, default operand size 64-bits. (66H override causes 16-bit operation).

NOTES:

* See IA-32 Architecture Compatibility section below.

Description

Decrements the stack pointer and then stores the source operand on the top of the stack. The address-size attribute of the stack segment determines the stack pointer size (16, 32 or 64 bits). The operand-size attribute of the current code segment determines the amount the stack pointer is decremented (2, 4 or 8 bytes).

In non-64-bit modes: if the address-size and operand-size attributes are 32, the 32-bit ESP register (stack pointer) is decremented by 4. If both attributes are 16, the 16-bit SP register (stack pointer) is decremented by 2.

If the source operand is an immediate and its size is less than the address size of the stack, a sign-extended value is pushed on the stack. If the source operand is the FS or GS and its size is less than the address size of the stack, the zero-extended value is pushed on the stack.

The B flag in the stack segment's segment descriptor determines the stack's address-size attribute. The D flag in the current code segment's segment descriptor (with prefixes), determines the operand-size attribute and the address-size attribute of the source operand. Pushing a 16-bit operand when the stack address-size attribute is 32 can result in a misaligned stack pointer (a stack pointer that is not be aligned on a doubleword boundary).

The PUSH ESP instruction pushes the value of the ESP register as it existed before the instruction was executed. Thus if a PUSH instruction uses a memory operand in which the ESP register is used for computing the operand address, the address of the operand is computed before the ESP register is decremented.

In the real-address mode, if the ESP or SP register is 1 when the PUSH instruction is executed, an #SS exception is generated but not delivered (the stack error reported prevents #SS delivery). Next, the processor generates a #DF exception and enters a shutdown state as described in the #DF discussion in Chapter 5 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A*.

In 64-bit mode, the instruction's default operation size is 64 bits. In a push, the 64-bit RSP register (stack pointer) is decremented by 8. A 66H override causes 16-bit operation. Note that pushing a 16-bit operand can result in the stack pointer misaligned to 8-byte boundary.