

CALL

< Previous instruction: [BZ](#) | Instruction [index](#) | Next instruction: [CLRF](#) >

CALL	Subroutine Call
Syntax:	[<i>label</i>] CALL k [,s]
Operands:	$0 \leq k \leq 1048575$ $s \in [0,1]$
Operation:	$(PC) + 4 \rightarrow TOS$, $k \rightarrow PC\langle 20:1 \rangle$, if $s = 1$ $(W) \rightarrow WS$, $(STATUS) \rightarrow STATUSS$, $(BSR) \rightarrow BSRS$
Status Affected:	None
Encoding:	
1st word ($k\langle 7:0 \rangle$)	1110 110s k_7kkk $kkkk_0$
2nd word($k\langle 19:8 \rangle$)	1111 $k_{19}kkk$ $kkkk$ $kkkk_8$
Description:	Subroutine call of entire 2 Mbyte memory range. First, return address ($PC + 4$) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into $PC\langle 20:1 \rangle$. CALL is a two-cycle instruction.
Words:	2
Cycles:	2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: HERE CALL THERE, 1

Before Instruction

PC = address (HERE)

After Instruction

PC = address (THERE)
TOS = address (HERE + 4)
WS = W
BSRS = BSR
STATUSS= STATUS

< Previous instruction: [BZ](#) | Instruction [index](#) | Next instruction: [CLRF](#) >