

BNC

< Previous instruction: [BN](#) | Instruction [index](#) | Next instruction: [BNN](#) >

BNC	Branch if Not Carry			
Syntax:	[<i>label</i>] BNC n			
Operands:	$-128 \leq n \leq 127$			
Operation:	if carry bit is '0' $(PC) + 2 + 2n \rightarrow PC$			
Status Affected:	None			
Encoding:	1110	0011	nnnn	nnnn
Description:	If the Carry bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC+2+2n$. This instruction is then a two-cycle instruction.			
Words:	1			
Cycles:	1(2)			
Q Cycle Activity:				
If Jump:				

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BNC Jump

Before Instruction

PC = address (HERE)

After Instruction

 If Carry = 0;

 PC = address (Jump)

 If Carry = 1;

 PC = address (HERE+2)

< Previous instruction: [BN](#) | Instruction [index](#) | Next instruction: [BNN](#) >