BNN

< Previous instruction: BNC | Instruction index | Next instruction: BNOV >

BNN Branch if Not Negative

Syntax:	[*label*] BNN n
Operands:	$-128 \leq n \leq 127$
Operation:	if negative bit is '0' (PC) + 2 + 2n → PC
Status Affected:	None
Encoding:	`1110` `0111` `nnnn` `nnnn`
Description:	If the Negative 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 BNN Jump`

Before Instruction

PC	=	address (`HERE`)

After Instruction

If Negative	=	0;
PC	=	address (`Jump`)
If Negative	=	1;
PC	=	address (`HERE+2`)

< Previous instruction: BNC | Instruction index | Next instruction: BNOV >