

BOV**Branch if Overflow**Syntax: $[label] \text{ BOV } n$ Operands: $-128 \leq n \leq 127$ Operation: if overflow bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0100	nnnn	nnnn
------	------	------	------

Description: If the Overflow bit is '1', 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 BOV Jump

Before Instruction

PC = address (HERE)

After Instruction

If Overflow = 1;
PC = address (Jump)

If Overflow = 0;
PC = address (HERE+2)

< Previous instruction: [BNZ](#) | Instruction [index](#) | Next instruction: [BRA](#) >