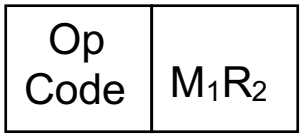
BCR M1,R2

Branch On Condition Register

RR



The Branch on Condition Register instruction examines the 2-bit condition code in the PSW and branches (or not) based on the value it finds. Operand 1 is a self-defining term which represents a 4-bit mask (binary pattern) indicating the conditions under which the branch should occur. Operand 2 is a register containing the target address to which the branch will be made if the condition indicated in Operand 1 occurs. If the condition code is one of the values specified in the mask, the instruction address in the PSW is replaced with the target address in R_2. This causes the processor to fetch the instruction located at the target address as the next instruction in the fetch/execute cycle. See **System/z Architecture** for more details.

There are four possible values for the condition code:

Condition Code	Meaning
00	Zero or Equal
01	Low or Minus
10	High or Plus
11	Overflow

When constructing a mask for Operand 1, each bit (moving from the high-order bit to the low-order bit) represents one of the four conditions in the following order: Zero/Equal, Low/Minus, High/Plus, Overflow. Consider the following instructions,

```
LA   3,THERE
BCR  8,3
```

In the BCR, the first operand,"8", is a decimal self-defining term and represents the binary mask B'1000'. Since the first bit is a 1, the mask indicates that a branch should occur on a zero or equal condition. Since the other bits are all 0, no branch will be taken on the other conditions. The first operand could be designated as any equivalent self-defining term. For example, the following instruction is equivalent to the BCR above.

```
BCR   B'1000',3
```

Extended mnemonics were developed to replace the awkward construction of having to code a mask. The extended mnemonics are easier to code and read. A listing of the extended mnemonics follows below.

```
BER    Branch Equal Register      BNER    Branch Not Equal Register
BZR    Branch Zero Register       BNZR    Branch Not Zero Register
BLR    Branch Low Register        BNLR    Branch Not Low Register
```

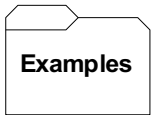
```
BMR    Branch Minus Register     BNMR    Branch Not Minus Register
BHR    Branch High Register      BNHR    Branch Not High Register
BPR    Branch Positive Register  BNPR    Branch Not Positive Register
NOPR   No Operation Register     BR      Unconditional Branch Register
```

Using extended mnemonics we could replace the previous Branch On Condition Register instruction with the following,

```
                          BZR   3
```

The “BZR” means “Branch on Condition Zero Register”. When the assembler processes this **BZR**, it generates the mask as B’1000’. The table below indicates the possible mask values and the equivalent extended mnemonics. Notice that not all masks have an equivalent extended mnemonic.

Eq/Low	Low/Min	High/Plus	Overflow	Decimal Condition	Extended Mnemonic
0	0	0	0	0	NOPR
0	0	0	1	1	BOR
0	0	1	0	2	BHR, BPR
0	0	1	1	3	No mnemonic
0	1	0	0	4	BLR, BMR
0	1	0	1	5	No mnemonic
0	1	1	0	6	No mnemonic
0	1	1	1	7	BNER, BNZR
1	0	0	0	8	BER, BZR
1	0	0	1	9	No mnemonic
1	0	1	0	10	No mnemonic
1	0	1	1	11	BNLR, BNMR
1	1	0	0	12	No mnemonic
1	1	0	1	13	BNHR, BNPR
1	1	1	0	14	No mnemonic
1	1	1	1	15	BR



Some Unrelated Branch on Condition Register Examples

```
            LA   7,THERE   POINT REGISTER AT TARGET ADDRESS
            LTR  R8,R8     SET THE CONDITION CODE
            BPR  7         BRANCH IF CONDITION CODE IS POSITIVE
            ...            OTHERWISE FALL THROUGH TO NEXT INSTRUCTION
THERE       EQU  *

            LA   5,THERE   POINT REGISTER AT TARGET ADDRESS
            CLC  X,Y       SET THE CONDITION CODE
            BER  5         BRANCH IF X = Y
            ...            OTHERWISE FALL THROUGH TO NEXT INSTRUCTION
THERE       EQU  *

            LA   9,THERE   POINT REGISTER AT TARGET ADDRESS
            CLC  X,Y       SET THE CONDITION CODE
            BHR  9         BRANCH IF X > Y
            ...            OTHERWISE FALL THROUGH TO NEXT INSTRUCTION
THERE       EQU  *
```