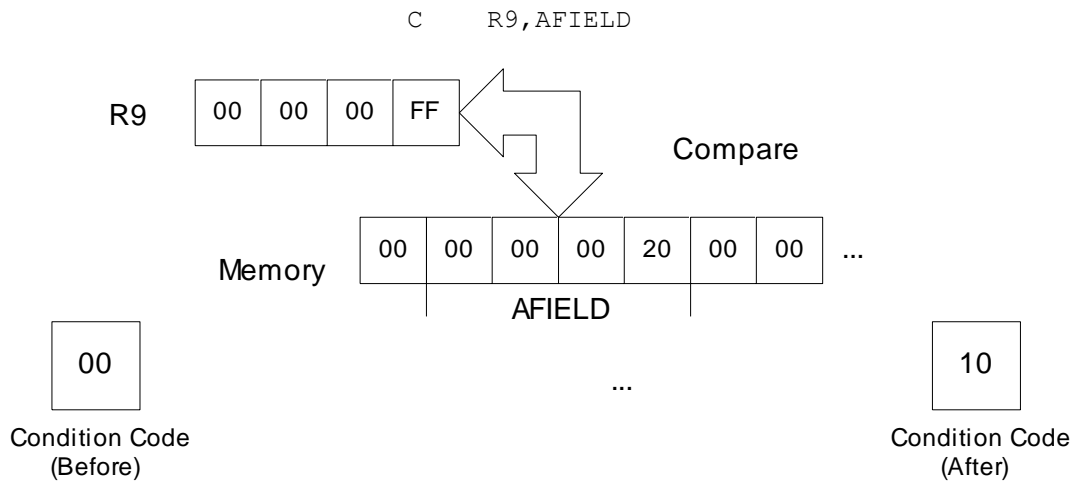


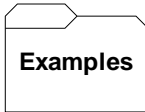
The Compare instruction is used to compare a binary fullword in a register, Operand 1, with a fullword in memory, Operand 2. The operands are compared as 32-bit signed binary integers. The instruction sets the condition code to indicate how Operand 1 compares to Operand 2:

Condition Code	Meaning	Test With
0	Operand 1 = Operand 2	BE, BZ
1	Operand 1 < Operand 2	BL, BM
2	Operand 1 > Operand 2	BH, BP



The contents of the fullword “AFIELD”, x’00000020’, is compared to the contents of register 9 which contains x’000000FF’. Since the contents of the register (Operand 1) is greater than the value than the fullword in memory (Operand 2), the condition code is set to “High”. The condition code in the diagram above is specified using 2 binary digits. After comparison, the condition code is set to a binary 10 which is 2 in decimal - a “High” condition.

Since **C** is an RX instruction, an index register may be coded as part of operand 2 (see **Explicit Addressing**).



Some Unrelated Compares

```
R4 = X'FFFFFFD5'  -43 IN 2'S COMPLEMENT
R5 = X'00000028'  +40 IN 2'S COMPLEMENT
R6 = X'00000004'  +4  IN 2'S COMPLEMENT
```

```
DOG  DC   F'35'
CAT  DC   F'4'
```

```
C   R4,=F'20'    CONDITION CODE = LOW
C   R5,=F'-20'   CONDITION CODE = HIGH
C   R6,=F'20'    CONDITION CODE = LOW
C   R6,=F'4'     CONDITION CODE = EQUAL
C   R5,=F'40'   CONDITION CODE = EQUAL
C   R5,DOG       CONDITION CODE = HIGH
C   R6,DOG(R6)  CONDITION CODE = EQUAL
```