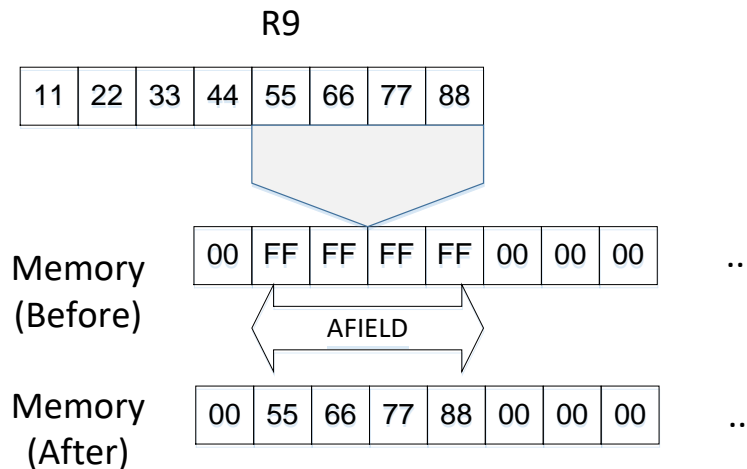


ST is used to copy the fullword stored in the register specified by operand 1 into the fullword memory location specified by operand 2. Only the rightmost 32 bits (bits 32-63) of the register are copied. The leftmost 32 bits (bits 0-31) do not participate.

Consider the following example,

ST R9, AFIELD



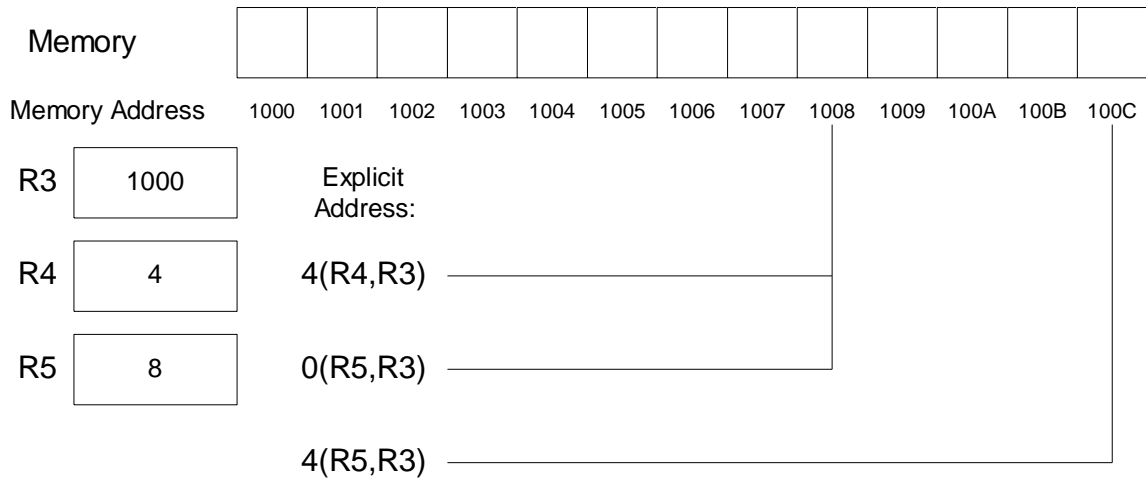
In this case, the rightmost 32 bits of register 9 are copied to the fullword in memory denoted by AFIELD. This operation destroys the previous contents of AFIELD but leaves R9 unchanged.

Since **ST** is an RX instruction, an index register may be coded as part of operand 2. In the previous example, no index register was specified. When the index register is omitted, the assembler chooses register 0, which does not contribute to the address. The following example illustrates this idea.

ST R9, AFIELD(R5)

The assembler converts the symbol AFIELD to a base register and displacement, while R5 is the index register. The expression AFIELD(R5) might (we cannot determine the base register without more information) be equivalent to the explicit address $0(R5, R3)$ - displacement = 0, index register = R5, base register = R3. The effective address is computed by adding the base register contents to the index register contents plus the displacement. If the index register

contains an “8”, then AFIELD(R5) refers to the fullword that begins at an 8 byte displacement from the beginning byte of AFIELD. The following examples illustrate several explicit addresses that include an index register.



In the first explicit address, 4(R4, R3), the effective address is computed by adding the contents of base register 5, the contents of index register 3, and the displacement (1000 + 4 + 4 = 1008). The second address 0(R5, R3) is computed as 1000 + 8 + 0 = 1008, and the third address, 4(R5, R3) is computed to be 1000 + 8 + 4 = 100C (hexadecimal).

If an index register is not explicitly coded, as in the instruction “ST R9,AFIELD”, the assembler chooses R0 as the index register, which does not contribute to the effective address.

Examples

Some Unrelated ST's

```
R7 = X'00000000000001000'
R8 = X'00000000000000004'
R9 = X'00000000000000008'
```

```
AFIELD DC F'20'      AFIELD = X'00000016'
BFIELD DC F'-1'     BFIELD = X'FFFFFFFF'
CFIELD DC F'0'      CFIELD = X'00000000'
```

```
ST R7,AFIELD      AFIELD = X'00001000'
ST R8,AFIELD      AFIELD = X'00000004'
ST R8,BFIELD      BFIELD = X'00000004'
ST R7,AFIELD(R8)  CHANGES BFIELD TO X'00001000'
ST R7,AFIELD(R9)  CHANGES CFIELD TO X'00001000'
```

👉 Tips

1) Operand 2 should denote a fullword in memory. It is possible to store the contents of a register into 4 bytes of memory that are not aligned on a fullword, but the assembler will warn you that operand 2 is not properly aligned. If the field involved cannot be aligned conveniently, consider using **STCM** to copy the contents of a register into memory.