

LM is used to copy a set of consecutive fullwords in main storage into a “consecutive” collection of registers. The collection of registers is specified by coding the beginning and ending registers as the first two operands. For instance,

```
LM R5, R8, XWORDS
```

would be used to load registers 5, 6, 7, and 8. If the second operand specifies a lower register than the first operand, then a “wrap-around” occurs. For instance,

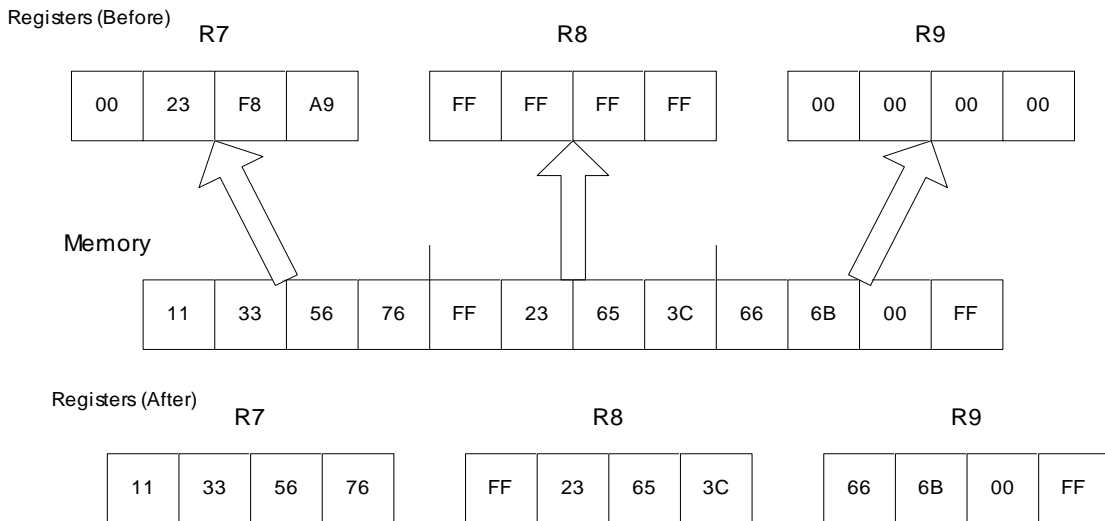
```
LM R14, R12, 12 (R13)
```

would be used to load registers 14, 15, 0, 1, ..., 12 - every register except 13.

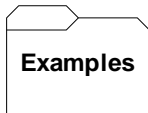
The registers are loaded starting with the fullword at the address specified in the third operand. In the first example above, register 5 would be loaded with the fullword at XWORDS, register 6 with the fullword at XWORDS+4, register 7 with the fullword at XWORDS+8, and register 8 with the fullword at XWORDS+12. In the second example, register 14 is loaded with the fullword at the explicit address indicated as a 12 byte displacement off register 13. Register 15 is loaded with the fullword at a 16 byte displacement off register 13. Register 0 is loaded with the fullword at a 20 byte displacement off register 13. The remaining registers are loaded in a similar manner.

Here is an illustrated example.

```
STM R7, R9, REGWORDS
```



In the example above, the consecutive range of registers (7 , 8 , and 9) are loaded with three consecutive fullwords in memory starting with the fullword specified as REGWORDS.



Some Unrelated LM's

```

XWORD  DC    F'200', F'50', F'30', F'90'
        ...
        LM    R5, R6, XWORD      R5 = F'200'
                                   R6 = F'50'
        LM    R15, R1, XWORD     R15 = F'200'
                                   R0 = F'50'
                                   R1 = F'30'
        LM    R4, R7, XWORD      R4 = F'200'
                                   R5 = F'50'
                                   R6 = F'30'
                                   R7 = F'90'
        LM    R14, R12, 12 (R13) THIS LOADS ALL REGISTERS EXCEPT 13
                                   STARTING WITH THE FULLWORD THAT
                                   IS 12 BYTES OFF REGISTER 13. (SEE
                                   PROGRAM LINKAGE.)

```

Tips

1) This instruction is helpful for creating subroutines. Branch to a subroutine and immediately save the registers with **STM**. This allows your subroutine to freely use the registers. Before exiting your subroutine, restore the registers with **LM**. Create a register "save area" for each subroutine.

```

        BAS    R6, SUBRTN      CALL THE SUBROUTINE
        ...
SUBRTN  EQU    *
        STM    R14, R12, SUBSAVE  SAVE THE REGISTERS
        ... (SUBROUTINE CODE)
        LM    R14, R12, SUBSAVE  RESTORE THE REGISTERS
        BR    R6                BRANCH BACK TO CALLER

```

SUBSAVE DS 15F
REGISTERS

FIFTEEN FULLWORDS FOR THE

- 2) Read about the role that **LM** plays in implementing the **Program Linkage** conventions.