Memory Operands

- **Memory operands**
    - Memory operand is specified by a **starting address**
    - Starting from that address, 1, 2 or 4 bytes are used for byte, word and long word operands as follows:

```
                 Memory:

               +-----------+
               | ........  |
    Starting   +-----------+
    address:   | aaaaaaaa  |
               +-----------+
               | bbbbbbbb  |
               +-----------+
               | cccccccc  |
               +-----------+
               | dddddddd  |
               +-----------+
               | ........  |
               +-----------+

     Byte Operand at Starting Address:
               +-----------+
               | aaaaaaaa  |
               +-----------+

     Word Operand at Starting Address:
               +-----------+----------+
               | aaaaaaaa  | bbbbbbbb |
               +-----------+----------+

     Long Word Operand at Starting Address:
               +-----------+----------+----------+----------+
               | aaaaaaaa  | bbbbbbbb | cccccccc | dddddddd |
               +-----------+----------+----------+----------+
```

- **DEMO:** click here

- **Warning:** (second part of demo program mem-operands.s)
    - Make **dead sure** that you use the ***correct* operand size** in assembler programs !!!

```
Example:

     move.l #-8, d0          Result: d0 = 11111111 11111111 11111111 11111000

     move.l d0, 5672         Result in memory:
                                      5672:  11111111
                                      5673:  11111111
                                      5674:  11111111
                                      5675:  11111000

     move.b 5672, d1         Will move BYTE at address 5672 (= 11111111)
                             into (lower 8 bits) of d1

                             d1 will NOT be equal to -8 !

     move.w 5672, d2         Will move WORD at address 5672 (= 11111111 11111111)
```

```
                         into (lower 16 bits) of d1

                         d2 will NOT be equal to -8 !

   move.l 5672, d3       d3 = -8 --- only this instruction will move -8 in a register
```

- DEMO: click here