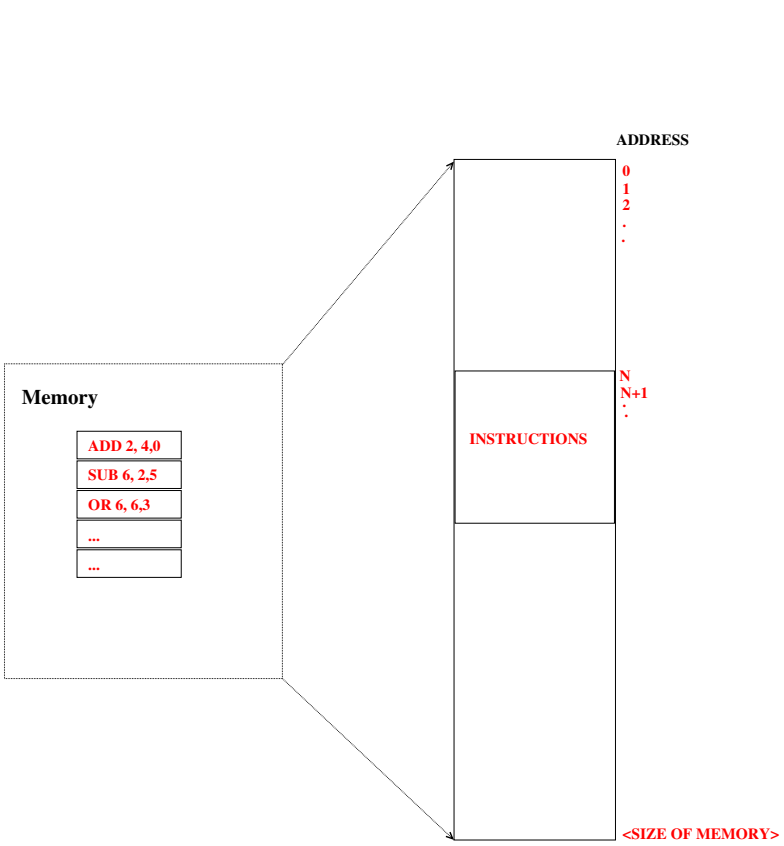
INF-1100
Arrays and composite data structures

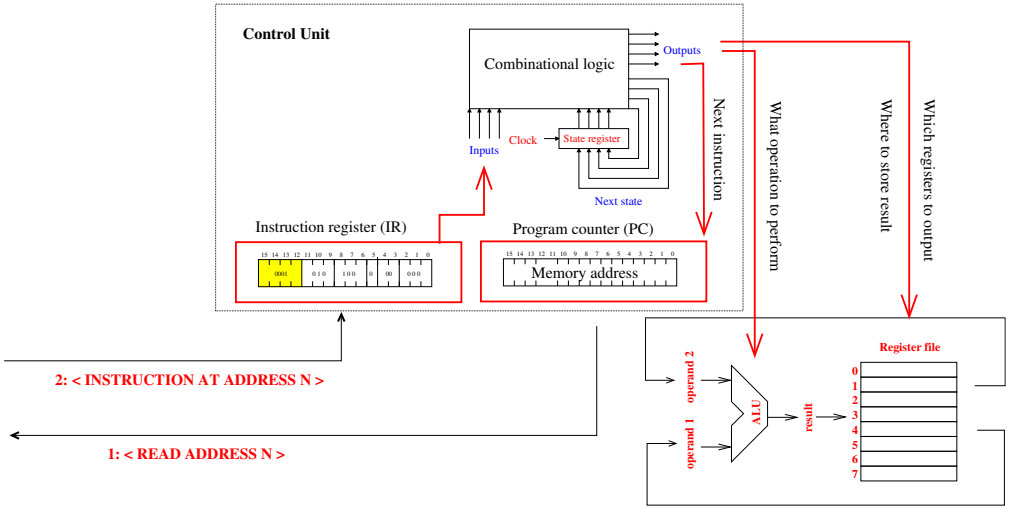
Åge Kvalnes

University of Tromsø, Norway

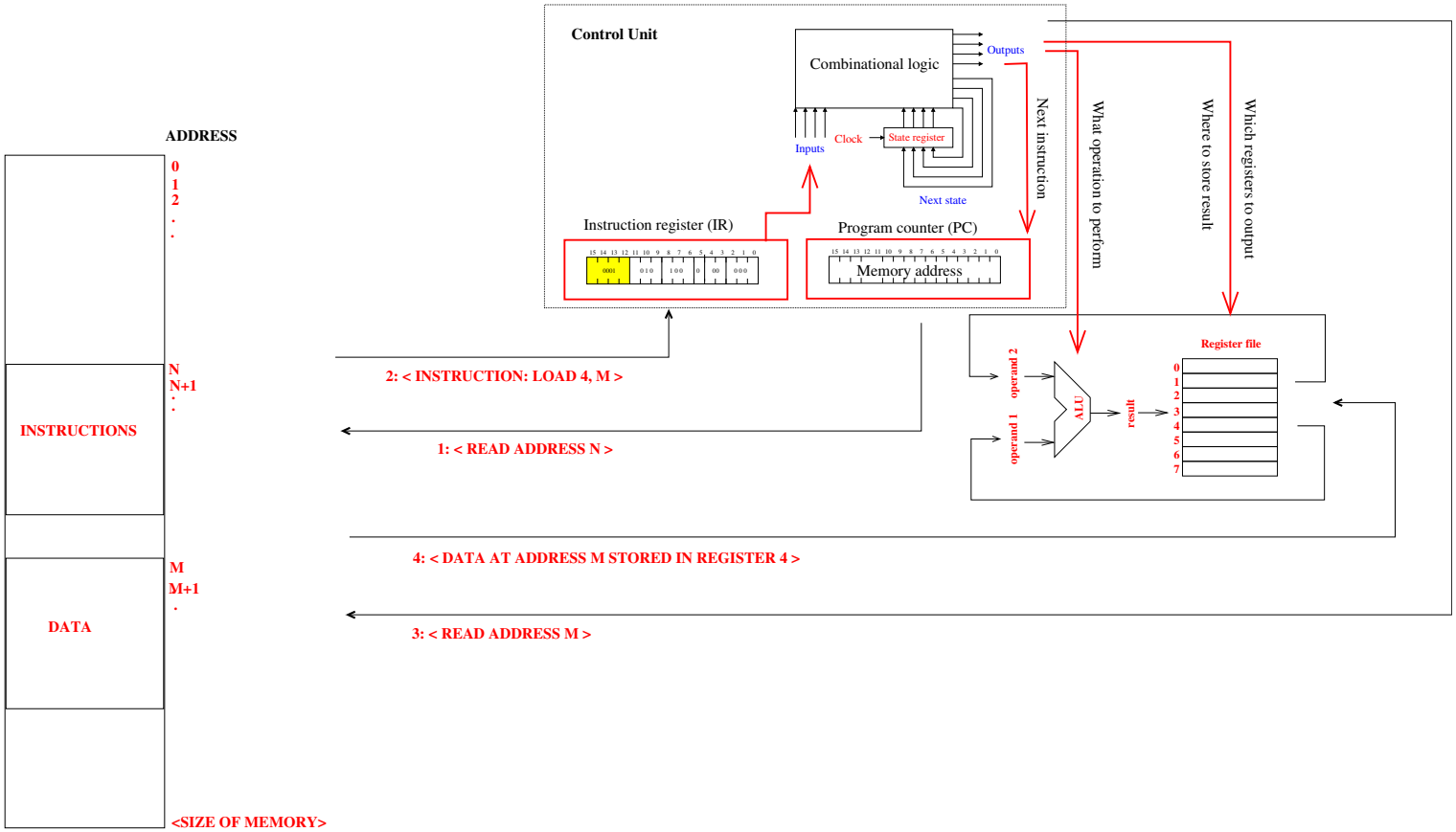
September 24, 2014

Instructions are located at a specific memory address





Data is located at a specific memory address

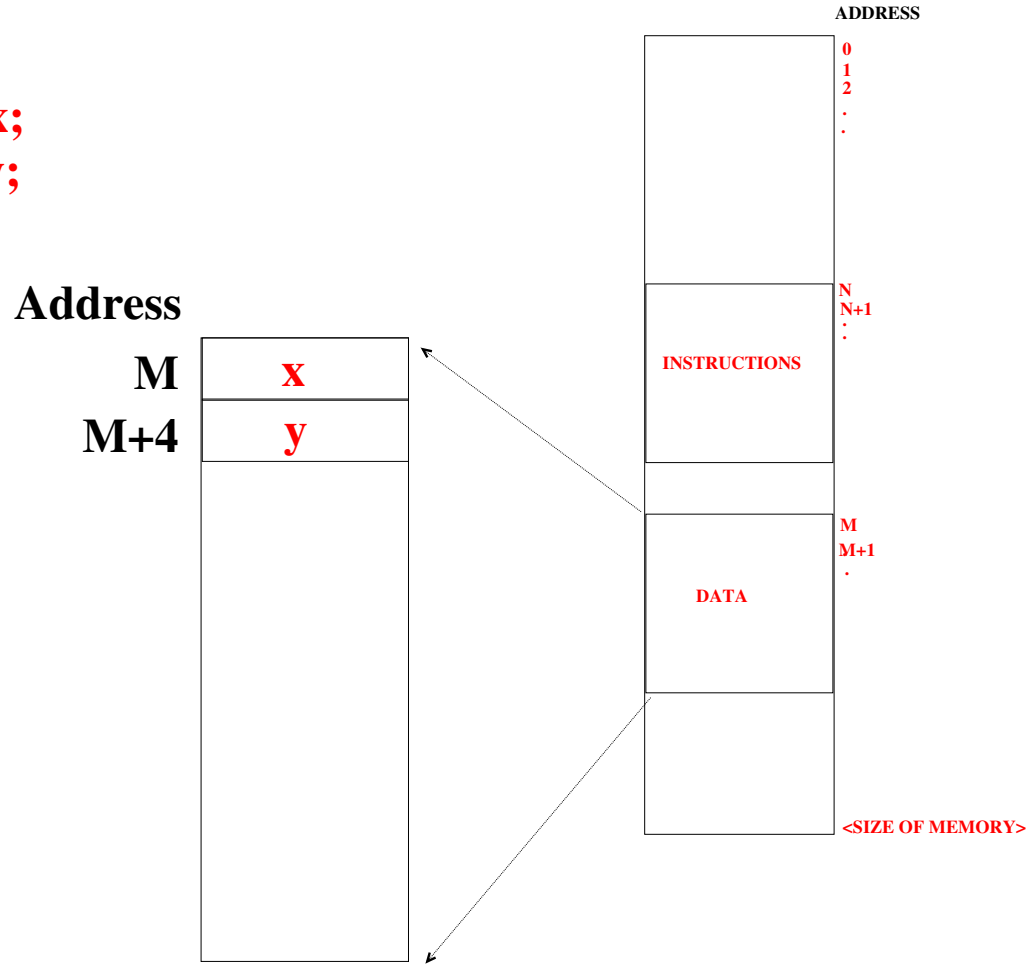


Data manipulated by instructions is also stored in memory

Declared variables are stored in memory

C code:

```
int x;
int y;
```



Primitive data types in C

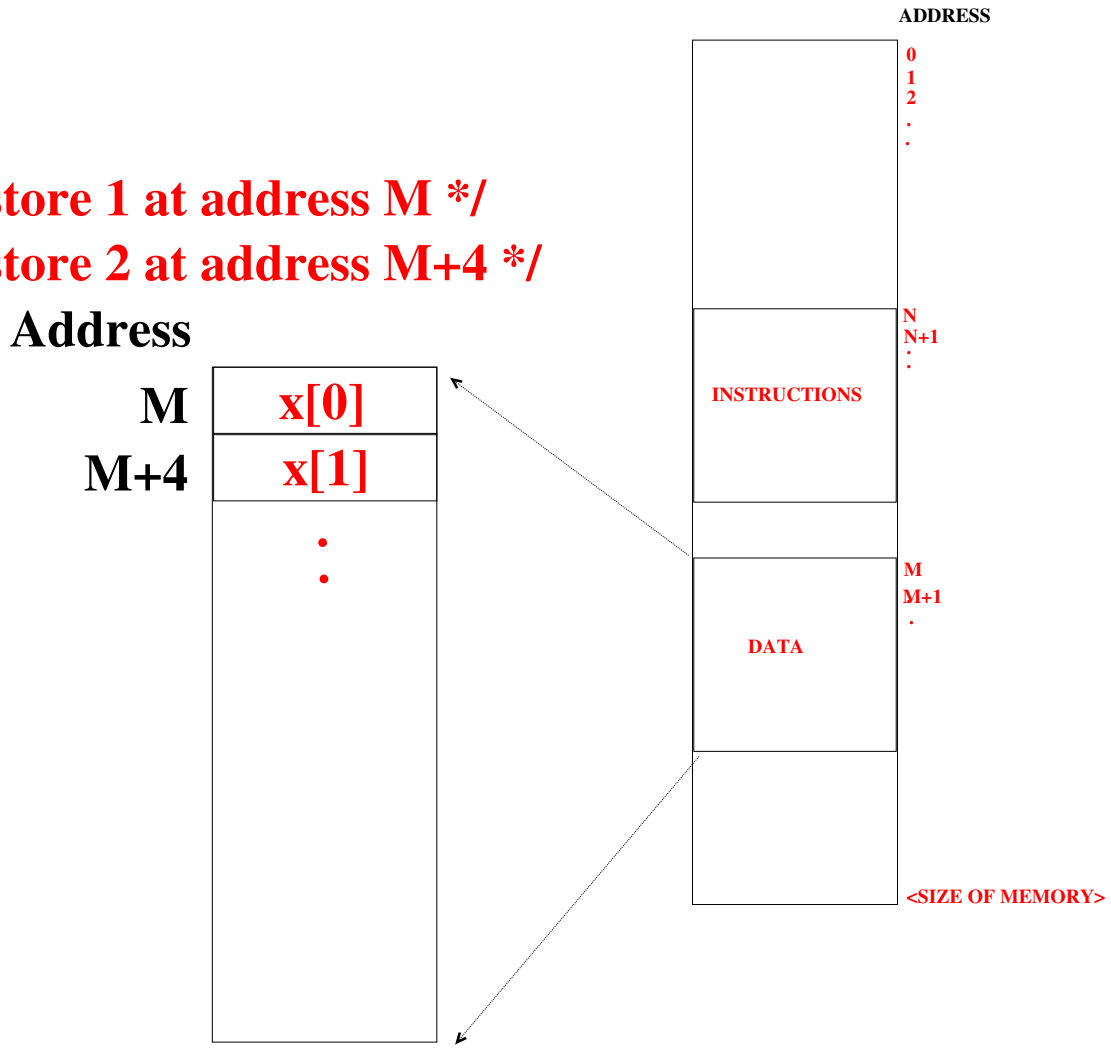
- char signed integer. 8 bits minimum, usually 8.
- short signed integer. 16 bits minimum, usually 16.
- int signed integer. 16 bits minimum, usually 32.
- long signed integer. 32 bits minimum, 64 if ISA is 64-bit.
- float fractional number. 6 decimal digits and +-37 exponent range minimum.
- double fractional number. 10 decimal digits and +-37 exponent range minimum, usually much higher.

Adding unsigned in front of a signed integer turns it into an unsigned integer (values from 0 and up).

Arrays of primitive data types

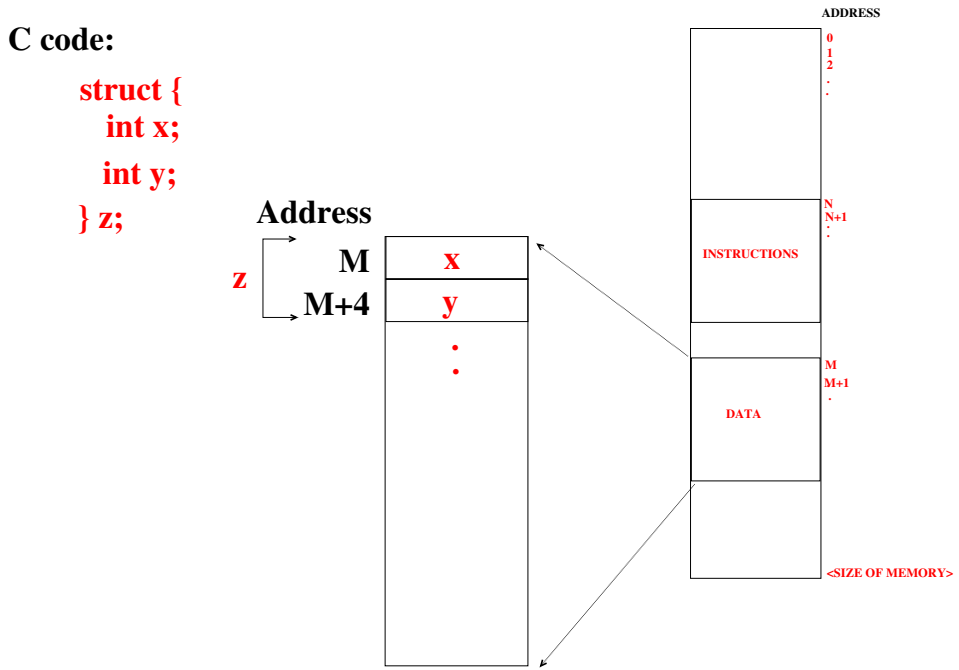
C code:

```
int x[10];
x[0] = 1; /* store 1 at address M */
x[1] = 2; /* store 2 at address M+4 */
```



C allows you to create *arrays* of primitive data types

Composite data structures



The *variable* z is a composite data structure.

- z.x refers to x
- z.y refers to y

Palindrome

Problem: Determine whether a word is a palindrome.

Algorithm:

1. Scan word from left and right and compare characters. If scan crosses the middle of the word and all left and right characters are the same, the word is a palindrome.

Caesar cipher

Problem: Transform plain text into encrypted text or encrypted text into plain text by use of the Caesar cipher.

Algorithm:

1. Encrypt: $E_n(x) = (x + n)\ mod\ 26$
2. Decrypt: $D_n(x) = (x - n)\ mod\ 26$