-]

iRegisters – storage capacities – very small in terms of sizes (8,16, 32 or 64 bits) but very fast as access speed used for temporary store the operands (data, commands codes, ADDRESSES !!!!!!!!) with which a processor currently.

3654

76532

1234876453

7464646464098234098209842083490238409283048283428342438

Computer/processor “on N bits”:

a). software perspective – the size of the memory word = the size of the (majority of) the registers (in our case = 32 bits)

b).engineering perspective – the size of the communication buses (channels) - ABUS, CBUS, DBUS

A[i] ; I = INDEX

A[7] = *(A+7)   * = the DEREFERENCING operator in C

A – the name of an array in C is its starting address – it’s a pointer = THE BASE (the starting address)

Byte + byte (ADD) = byte ;

MULTIPLY op1(M positions) * op2 (N positions) → M+N positions

B*B → W(ord); W*W → DW ; DW*DW → QW (EDX:EAX)
                                        (DX:AX)

Data structures – array, list, queue (FIFO), stack (LIFO)

WHY is the stack SO important ?????

RUN-TIME Mechanism of ANY program in Computer Science FOLLOWS ALWAYS THE LIFO ORDER of activating and running the involved programming units (subroutines = functions + procedures).

A user defined type in C is defined by TYPEDEF (which is INCORRECT, because typedef is in fact defining only the structure)

C, Java, VB, Pascal, Fortran – were IMPERATIVE language , because they rely as a central element on the INSTRUCTIONS (commands).

DATA TYPE = structure + associated OPERATIONS !!!!

(essential in this definition is ASSOCIATED – we did not have until OOP AN ENCAPSULATION mechanism)

- You also have in OOP inheritance + polymorphism;

OOP = DATA ORIENTED PROGRAMMING (everything is built having as the central figure the notion of DATA).

From the point of view of the mP – which is its understanding of DATA TYPE notion ?

DATA TYPE for the mP (and in Assembly language) – The size of representation of that element;

On 32 bits these can be – byte, word, dword and qword (these are the assembly language DATA TYPES); You can define variables/operands in the RAM memory by using DATA DEFINITION DIRECTIVES – DB, DW, DD, DQ.

=============

RAM (Random Access Memory) – who is RANDOM ?

- The access time at any given location from the RAM is THE SAME independently of the position (randomely far from the beginning of the memory…)

- In contrast from ROM (read only memories) a RAM supports/allows any number of R/W and in any ORDER (Randomely… reads and writes in a randomely order… The order in which R/W appear is RANDOM…)

==============

101101100101 – representation (in base 2) of what do I need as a base 10 value as a user/human being

So, we need a TRANSFORMATION in base 10 !!!

REPRESENTATION (in base 2) vs. INTERPRETATION (in base 10) !!!!!!!!

UNSIGNED vs SIGNED – how can we REPRESENT them ?

The answer – develop 2's complement representation !!!!!

Contents of a computer memory ---→ correct and consistent INTERPRETATION in base 10 !!!!!!

Base 2 – 1…… → the 2 interpretations (SIGNED and UNSIGNED) will be ALWAYS DIFFERENT !!!!

Base 2 – 0…… → the 2 interpretations (SIGNED and UNSIGNED) will be IDENTICAL !!!!