

5. MEMORIA

[5.1. Notiuni generale](#)

[5.2. Schema functionala](#)

[5.3. Tipuri de memorii](#)

[5.4. Modul de stocare al datelor in memorie](#)

[5.5. Exploatarea memoriei](#)

5.1. Notiuni generale

Caracteristici:

- Ansamblul de registre
- Locatie de memorie
- Adresa de memorie
- Capacitate de memorie
- Timp de acces
- Viteza de transfer

- Memorie cache

- Memorie virtuala

5.4. Modul de stocare al datelor in memorie

- „*little endian*”
- „*big endian*”

Exemple:

- numărul zecimal 1776 se reprezintă în interiorul calculatorului în hexa și va fi 06F0, iar în memorie F006.
- Numărul zecimal 7, definit la nivel de W (word) va fi reprezentat în registrii microprocesorului sub forma 0007, iar în memorie, va fi reprezentat invers: 0700.
- Numărul hexa 1ABC (pe 2B=1W), în memorie va arăta BC1A, iar în registrii 1ABC.
- Șirul de caractere „ABC”, definit la nivel de B (byte), va arăta în memorie în ordinea normală, adică „31 32 33” (caracterul ASCII A, caracterul ASCII B, caracterul ASCII C).

```
.....B8 3412 MOV AX,1234H
```

```
C:>debug
```

```
- d ds:0000
```

```
care va afișa:
```

```
..... 34 12 .....
```

TEMĂ: identificați modurile de reprezentare ale datelor în memorie pentru programul în Assembler realizat de voi.

5.5. Explorarea memoriei

- *memoria expandată* – EMS (Expanded Memory Specification) - EMM386.EXE
- *memoria extinsă* – XMS (Extended Memory Specification) -HIMEM.SYS

mem /? - este un help scurt privind opțiunile comenzii)

mem – oferă o privire de ansamblu rapidă asupra stării sistemului de memorie

mem /c – afișează programele, în ordinea folosirii cantității de memorie, într-un tabel.

mem /p - arată starea programelor încărcate în memorie

mem /d – oferă mai multe amănunte, inclusiv dimensiunea și localizarea în memorie a fiecărui program încărcat, tipul fiecărui element (program, device driver etc.).