

Little и Big endian (мала и велика крајност)

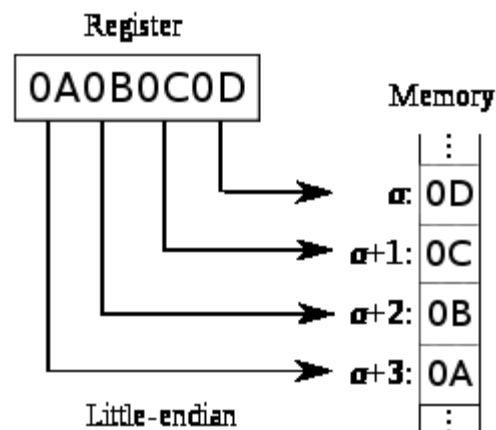


- Узмимо да је ширина меморије на неком процесору 8 бита, тј. да сваких 8 бита има своју адресу.
Питање: Како ћемо у меморију таквог процесора сместити 32-битну вредност?
Одговор: Поделићемо 32-битну вредност на 4 8-битне, нпр. вредност 0x0A0B0C0D се може поделити на следећа четири дела: 0x0A, 0x0B, 0x0C и 0x0D, поређаних по значајности, од највеће ка најмањој.
- Два су начина да се та четири дела сместе у меморију:
 - Делови мањег значаја иду на мању адресу - Литл ендијан
 - Делови већег значаја иду на мању адресу - Биг ендијан
- Са ког краја се вредности смештају у меморију највише зависи од самог процесора. Рецимо:
 - MIPS – може се изабрати LE и BE
 - Intel - LE
 - ARM - LE

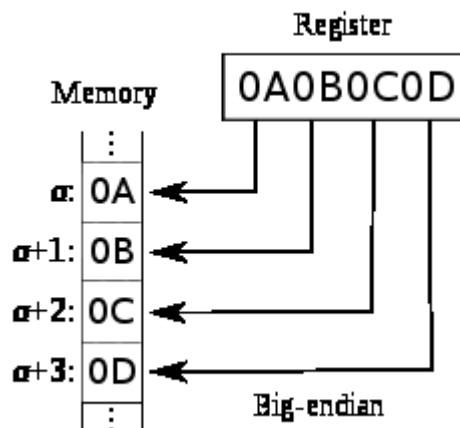


Little и Big endian

- Део најмањег значаја се уписује први = **little endian**



- Део највећег значаја се уписује први = **big endian**



Little и Big endian

Илустрација разлике



1. Смести 0x0A0B0C0D у меморију као 32-битну вредност (рецимо int типа)
2. Прочитај из тих меморијских локација као 2 узастопне 16-битне вредности
3. Прочитај из тих меморијских локација као 4 узастопне 8-битне вредности
4. Прочитане вредности:
 - 16bit access:**
 - **LE:** 0x0C0D и 0x0A0B
 - **BE:** 0x0A0B и 0x0C0D
 - 8bit access:**
 - **LE:** 0x0D, 0x0C, 0x0B и 0x0A
 - **BE:** 0x0A, 0x0B, 0x0C и 0x0D

Размена података између рачунара



- Постоји договор око формата којим се шаљу подаци преко мреже. **Мрежни стандард каже: big-endian.**
- За пребацивање у мрежни формат звати функцију `hton` (host-to-network). На big-endian рачунарима не ради ништа, али да би програм био портабилнији важно је користити `hton` увек пре слања података преко мреже.
- Аналогно постоји и функција `ntoh` (network-to-host) која служи за читање података са мреже.
- Функције за конверзију:
 - `htons()` – "Host to Network Short" 16бит
 - `htonl()` – "Host to Network Long" 32бит
 - `ntohs()` – "Network to Host Short" 16бит
 - `ntohl()` – "Network to Host Long" 32бит