**Ligjerata 10**

*Assembler* është paketë softuerike e cila konverton instruksionet në gjuhë të makinës. Gjuha e makinës është një listë e numrave 8 bitësh të cilët CPU mund ti dekodoj.

Instruksionet në 8051 janë 8-bitëshe, që do të thotë kemi 28 => 256 instruksione në bashkësinë e isntruksioneve të 8051. 255 prej tyre janë të implementuara. Dhe janë të ndara si në vijim: *139* – Instruksione 1 bajtëshe. *92* – Instruksione 2 bajtëshe. *24* – Instruksione 3 bajtëshe.

Instruksionet i tregojnë procesorit se cilat operacione ti ekzekutoj.

Siç e dimë instruksionet përbëhen nga opcode që është pjesa e cila i tregon CPU qfarë operacioni të kryes dhe pjesa e dytë janë operatorët, operandi në 8051 është gjithmonë 1 bajtësh, ndërsa operanti mund të mos jet fare prezent me ç’rast themi se kemi të bëjmë me instruksion 1 bajtësh, mund të kemi një operand (adresë ose e dhënë) me ç’rast themi se kemi të bëjmë me isntruksion 2 bajtësh, dhe mund të kemi raste kur 2 operand janë prezent ose një operand por ka gjatësi 16 bajtëshe (adres) dhe themi se instruksioni është 3 bajtësh.

Siç e dimë instruksionet i ndajmë në instruksione:

* Për lëvizjen e të dhënave.
* Aritmetike.
* Logjike.
* Boolean.
* Për kontrollrimin e rrjedhies së programit.

**Load** – Ky instruksion lexon përmbajtjen e lokacionit të specifikuar të memories dhe e kopjon atë në lokacionin e specifikuar të regjistrit në CPU.

**Store**- Ky instruksion kopjon përmbajtjen aktuale të regjistrit të specifikuar në lokacionin e specifikuar të memories.

**Add** - Ky instruksion shton përmbajtjen e lokacionit të specifikuar të memories në të dhënën e ndonjë regjistri.

**Compare** - Ky instruksion tregon nëse përmbajtja e regjistrit është më e madhe, më e vogël ose e barabartë me përmbajtjen e lokacionit të memories së specifikuar. Rezultati paraqitet në PSW si flag.

**Decrement** - Ky instruksion zbret 1 nga përmbajtja e lokacionit të specifikuar.

**AND** - Ky instruksion shërben operacionin logjik AND me përmbajtjen e lokacionit të memories dhe të dhënën në ndonjë regjistër.

**XOR** - Ky instruksion tregon operacionin logjik EXCLUSIVE-OR (Pa barazisë) me përmbajtje të lokacionit të memories së specifikuar dhe me të dhënën në ndonjë regjistër, bit për bit.

**Logical shift** (left-majtas apo right-djathtas) - Instruksioni logical shift mundëson zhvendosjen e sekuencës bitave në regjistër për një vend në të majtë apo në të djathtë duke vendosur 0 në fund ose fillim të numrit respektivisht. Për logical shift right, 0 është zhvendosur në MSB ashtu që biti i fundit është zhvendosur dhe ka vendosur carry flag (1) në PSW. 1011 -> 0101 (R)

**Arithmetic shift** (left ose right) – Ky instruksion mundëson zhvendosjen e bitave në një regjistër për një vend majtas ose djathtas, por kur kemi right shift biti i shenjës MSB duplikohet dhe vendoset në fillim kurse bitat tjerë shtyhen djathtas. 1011 -> 1101 (R).

**Rotate** (left ose right) – Ky instruksion mundëson lëvizjen e bitave në regjistër për një vend majtas ose djathtas duke vendosur bitat e përparëm në fund dhe të fundit në fillim. 1100 -> 0011.

**Jump**– Ky instruksion na mundëson ta ndrsyhojmë sekuencën e ekzekutimit të programit, ngase PC rritet sekuencialisht, ky instruksion nga mundëson që ta ndryshojmë vlerën e PC në mënyrë manuale duke kërcyer në lokacion të specifikuar të programit.

**Branch** – Ky instruksion na mundëson aplikimin e kushteve si prsh branch if zero, branch if plus. Prsh nëse A është zero shto 5 në të nëse jo JMP në një pjesë tjetër të programit.

**Halt** - Ky instruksion ndërpret të gjitha aktivitetet tjera të mikroprocesorit.

**Motorola** **68HC11 ka 6 mode adresimi**: immediate, direct, extended, indexed, inherent dhe relative.

**Intel 8051 ka 5 mode adresimi**: immediate, direct, register, indirect dhe indexed.

**PIC mikrokontrolleri** ka 3 mode adresimi: immediate, direct dhe indirect. Text

Description automatically generated with medium confidence

Programi i shkruajtur në asambler përbëhet nga një sekuencë e formularëve, një formular për rresht. Formulari përbëhet nga *label, opcode, operand* dhe *comment*.

*Label* – Është emri i të dhënës të cilës i referohet memoria.

*Op-code* – Tregon se si të dhënat duhet të manipulohen.

*Operand* - Jep adresën ose të dhënën me të cilën do të operohet.

*Comment* – Fushë opsionale në të cilën programuesi mund të jep komente.

Table

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated

*Loop* është përësritja e një operacioni të programit disa herë.

*Shembull i loop*: : Mbledhja e numrave të vendosur në 10 adresa të ndryshme (ky mund të jetë për shembull rezultati i hyrjeve prej 10 sensorëve të ndryshëm që duhet bërë semplimi).

Algoritmi:

1. Start.
2. Vendos 10 në counter.
3. Vendos në akumulator vlerën nga adresa e pointerit (Index registri).
4. Zvogëlo për një vlerën e counter.
5. Rrit për një vlerën e indexit.
6. Krahaso couter me zero, nëse nuk është zero shko në 3. Nëse po vazhdo.
7. Vendos shumën.
8. Stop.

Table

Description automatically generated