**Ligjerata 8**

*Regjistri* është grup i flip-flopa-ve të vendosur në paralel që përdoren për ruajtie të përkohëshme të bit-ve. Flip-fllopat në regjistra janë të pa varur në mes vete dhe secili ruajnë nga një bitë me peshë të caktuar binare sipas pozitave të tyre. Nga kjo rrjedhë se regjistrat janë elemente përbërëse të memories dhe përdoren si lokacione te veqanta të memories nga CPU.

Regjistrat i ndajmë në *General Prurpose Registers* dhe *Special Function Registers*.

*Regjistri A (Accumulator)*:

Ky regjistër është i lidhur ngushtë me ALU dhe merr pjesë në gjdo veprim aritmetiko logjik që ALU e realizon, duke ruajtur rezultatin e operacionit, kur një operacion kryet njëri operant ndodhet në A ndërsa tjetri operant merret nga një lokacion në memorie. (Operant – është e dhëna me të cilën kryhet operacioni aritmetiko-logjik). Accumulator (A) lidhet me memorien e jashtme me anë të data bus.

Text

Description automatically generated

*Regjistri B*:

Ky është një regjistër 8-bit i cili përdoret për ruajtien e njërit nga operantët gjatë ekzekutimit të opercaioneve të shumëzimit ose pjesëtimit.

*Regjistri SP (Stack pointer)*:

CPU e përdorë një pjesë të memories si Stack, që i mundëson ruajtien e më shumë të dhënave dhe adressave ngase numri i regjistrave është i kufizuar, për qasjen në stack CPU përdorë një regjistër të quajtur SP (Stack pointer) i cili mbanë adresën e *TOS* (Top Of The Stack), ky regjistër është 8-bit pra mund të mbaj vlera nga 0x00 -> 0xFF, kur incializohet në SP vendoset adresa 0x07, që do të thotë se stack në memorie fillon në adresën 0x08. Përveq se na mundëson ruajtien e të dhënave dhe adresave, stacku na mundëson kthimin në adresën e saktë të programit kryesorë pas kërcimit në ekzekutimin e nënprogrameve tjera.

*Regjistri DPR (Data pointer register)*:

Ky regjistër mund të përdoret si regjistër 16-bit ose 8-bit. Kur përdoret si 16-bit për mbulimin e memories së jashtme përdoren instruksione speciale për ti arritur të tërë 16-bit. Kur përdoret si 8-bit atëher manipulimi me DPR bëhet si me dy regjistra të ndarë DPH dhe DPL.

*Regjistri PSW (Program status word)*:

Ky është regjistër 8-bitësh ndryshe njihet si *Status Register*, për dallim nga regjistrat e tjerë në këtë regjistër bëhet deklarimi i komplet bajtit.

*Regjistri IR (Instruction register)*:

Ky regjistër përdoret nga CPU për të vendosur instrstruksiont nga ROM, ashtu që ti dekodoj e pastaj të inciohen qarqet për ekzekutim.

*Dekoderi i instruksioneve*:

Ka për detyrë dekodimin e instruksioneve dhe përcaktimin e operacionit që CPU duhet të kryej.

*Regjistri PC (Program counter)*:

Ky regjistër mban adresën e instruksionit të radhës që CPU duhet ta lexoj nga memoria dhe ta ekzekutoj, duke mundësuar kështu ezkeutimin e instruksioneve hap pas hapi ashtu siç janë të shkruajtura në program.

*Clock generator*: Burimi i impulseve për tërë mikrokontrollerin.

*Control Unit*: Mundëson paraqitjen e dukurive hap pas hapi në CPU.

*Oscilatori* në mikrokontrollerin 8051 lidhet në pina 18 dhe 19, ekzistojnë dy mënyra të realizimit të oscilatorit në 8051: 1. Me anë të një kristalit të kuarcit dhe dy kondensatorëve. 2. Sjellia e takt sinjalit nga një burim i jashtëm (TTL).

Graphical user interface

Description automatically generated with medium confidence

Pas resetimit të 8051 këto do të jenë vlerat e regjistrave:

* PC: 0x00;
* A, B, PSW: 0x00;
* SP: 0x07;
* DPTR: 0x00;
* Portat: 0xFF;
* SCON, SBUF (Regjistra të timers): 0x00;

*Dy variante të çarkut për resetim*:

Diagram

Description automatically generated

*Interrupts*: Janë sinjale të cilat kërkojnë vëmendien e menjëherëshme të mikrokontrollerit, duke e bërë që të ndërpresë ekzekutimin e programit aktual për të filluar ekzekutimin e një programi tjetër, për tu rikthyer më vonë në ekzekutimin e programit paraprak aty ku kishte mbetur para se të vinte sinjali.

*Rrjedha e veprimeve kur ndodhë një interrupt*:

* Ndërpritet ekzekutimi i programit kryesorë dhe degëzohet në *ISR* (Interruption service routine) ose *Interrupt Handler*.
* ISR bën ekzekutimin e programit të thirrur nga Interrupt.
* Kryhet operacioni i dëshiruar.
* Ekzekutohet instruksioni *RETI* (Return from interrupt).
* Vazhdohet ezekutimi i programit paraprak aty ku kishte mbetur.

*Në 8051 ekzistojnë 5 lloje të interrupts*:

* TF0 (Timer 0 overflow interrupt).
* TF1 (Timer 1 overflow interrupt.
* INT0 (External 0 hardware interrupt).
* INT1 (External 1 hardware interrupt).
* RI/TI (Serial communication interrupt).

Përveq në rastin e RI/TI (serial interrupt) për të gjitha interrupts tjerë vendosen *flag* që tregojnë se është dërguar sinjal për interrupt, *flag* i cili pastrohet pasi që programi të degëzohet tek *ISR*.

*Adresim* nënkupton përcaktimin e lokacionit të të dhënave në memorie ose në rigjistra.

Në 8051 kemi 8 metoda të adresimit (address modes), mirpo 5 janë më praktike dhe më të përdorura:

* *Register addressing:* Siç e dimë 32 bit e pjesës së poshtme të RAM, janë të organizuara në 4 Banka me nga 8 regjistra. Secila bank mban regjistra me emra R0...R7, pra emrat janë të njëjtë dhe për tu adresuar në bazë të emrit duhet të përdorim edhe emrin e Bankës, siç e dimë emrat e bankave janë B0...B3, pra së pari e caktojmë bankën regjistrat e së cilës duam ti përdorim, caktimi i bankës bëhet me anë të regjistrit *PSW* (Program status word) ky regjistër ka 8 bit PSW.0 -> PSW.7. PSW.3 dhe PSW.4 përdoren për të mbajtur bankën aktuale të zgjedhur.
  + *Shembull i adresimit me regjistra:* MOV PSW, #0x00 // zgjedhia e bankes 0 MOV A, R4 MOV PSW, #0x01 // zgjedhia e bankes 1 ADD A, R4

Diagram, schematic

Description automatically generated

* *Direct addressing:* Në këtë formë të adresimit, adresa e të dhënës specifikohet si operand në instruksion. Me këtë llojë të adresimit ne kemi qasje në *RAM pjesën e poshtme, SFR, Portet I/O, regjistrat e kontrollit.* Biti më i rëndësishëm (MSB) i adresës vendoset në 0 nëse adresa i përket RAM dhe vendoset 1 nëse i përket SFR. *Shembull*: MOV A, 47H. Këtu mund të shohim se kemi të bëjmë me një instruksion 2 bajtësh dhe ka një cikël që do të thotë se vlera e PC do të rritet për dy (0204), MOV A(E5) ruhet në njërin bllok të memories (0202) dhe 47H ruhet në një bllok (0203) tjetër të memories. Me ekzekutimin e instruksionit nga 0202 akumuluesi bëhet aktiv dhe i gatshëm të lexoj të dhëna, pastaj kontrolluesi i programit shkon në adresën 0203 ku gjen adresën 47H që simbolizon një adresë në memorien e të dhënave nga ku do ta mar të dhënën dhe ta bart në A (E0H). Chart, box and whisker chart

  Description automatically generated
* *Indirect addressing:* Në këtë mënyrë të adresimit adresa e operandit specifikohet si përmbajtie e një regjistri. Shembull: MOV A, @R1. Këtu @ tregon se kemi të bëjmë me adresim indirekt, kështu që nëse përmbajtia e R1 është 56H atëherë operanti do të merret nga adresa 56H ku le të themi e kemi vlerën 46H vlerë kjo që do të vendoset në regjistrin A. Vetëm regjistri R0 dhe R1 mund të përdoren për adresim indirekt me qrastë quhet *pointer register*. Instruksioni në këtë rast do të jetë 1 bajt dhe 1 cikël.Diagram, schematic

  Description automatically generated
* *Imediate addressing:* Në këtë mënyrë të adresimit e dhëna definohet në instruksion, si një konstanë 8 ose 16 bit-she. Regjistri ku kopjohet e dhëna duhet të jetë e madhësisë së njëjtë me operantin e specifikuar. Këtu simboli # tregon se nuk kemi të bëjmë me adresë por me të dhënë. Ky lloj i adresimit është më i shpejti përshkak se nuk kemi nevoj të lexojmë dy herë nga memoria ngase të dhënën e kemi në instruksion.
* Shembull: MOV A, #030H. Këtu kemi të bëjmë me një instruksion 2 baj dhe 1 cikël që do të thotë se vlera e PC regjistrit do të rritet për 2.

Diagram

Description automatically generated

* *Indexed addressing:* Në këtë mënyrë të adresimit, adresa efektive e operantit merret duke bërë kalkulimin e përmbajties së një regjistri bazë dhe një regjistri të kompensimit, regjistër bazë mund të jetë *DPTR* ndërsa regjistër i kompensimit mund të jetë *A*. Në këtë lloj adresimi mund të përdoret vetëm instruksionet MOVC dhe JMP. Shembull: MOVC A, @A + DPTR. Ky instruksion është 1 bajtësh dhe kërkon 2 cikle. Le të themi se A e ka vlerën 02H ndërsa DPTR e ka vlerën 01FE (të ndarë në DPH dhe DPL), kur këto i mbledhim fitojmë 0200H, që është një adresë në memorien e programit nga e cila do të mirret e dhëna.

Graphical user interface

Description automatically generated

*\*Munen me ra edhe opcode në hex për secilën komand të adresimit\**