

Çfarë është një sistem?

Një sistem është një mënyrë për të punuar, organizuar ose kryer një ose më shumë detyra sipas një plani fiks, programi ose grup rregullash. Një sistem është gjithashtu një rregullim sipas të cilit mbledhen të gjitha njësitë e saj dhe punojnë së bashku sipas një plani ose programi.

Shembull: ORA

Ora është një system që shërben për matjen e kohës.

Pjesët: Hardueri, gjilpërat, musha për matje, rripat, etj.

Rregullat:

1. Të gjitha gjilpërat lëvizin vetëm në drejtim të akrepave të orës
2. Një gjilpërë e hollë rrotullohet çdo sekondë
3. Një gjilpërë e gjatë rrotullohet çdo minutë
4. Një gjilpërë e shkurtër rrotullohet çdo orë
5. Të gjitha gjilpërat kthehen në pozicionin origjinal pas 12 orësh

Çfarë është një embedded system?

An embedded system is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system. Termi sistem i ngulitur i referohet përdorimit të elektronikës dhe softuerit brenda një produkti, në krahasim me një kompjuter me qëllim të përgjithshëm, të tilla si një kompjuter Laptop ose Desktop.

- Pajisje të tilla përfshinë: telefona celulare, aparat fotografik dixhital, video kamera, kalkulatorë, furra me mikrovalë, sisteme të sigurisë në shtëpi, lavatriçe, sisteme ndriçimi, sisteme të ndryshme automobilistike, furça dhëmbësh etj.

Mund të ketë një **shumëllojshmëri të ndërfaqeve** të cilat i mundësojnë sistemit për të matur, manipuluar dhe të bashkëveproj me mjedisin e jashtëm. Sistemet e ngulitura shpesh ndër-veprojnë me botën e jashtme përmes **sensorëve dhe aktuatorëve**, dhe prandaj janë sisteme tipike reaktive. Ndërfaqja njerëzore mund të jetë aq e thjeshtë sa një dritë e ndezur ose aq e komplikuar sa vizioni robotik në kohë reale. **Porta diagnostike** mund të përdoret për diagnostikimin e sistemit që po kontrollohet.

Karakteristikat e embedded systems:

1. Kërkon performancë në kohë reale
2. Duhet të ketë disponueshmëri dhe besueshmëri të lartë.
3. Zhvilluar rreth një sistemi operativ në kohë reale
4. Zakonisht, keni një funksion të lehtë dhe pa disk, ROM boot
5. Projektuar për një detyrë specifike
6. Duhet të lidhet me pajisje periferike për të lidhur pajisjet hyrëse dhe dalëse.
7. Ofron besueshmëri dhe qëndrueshmëri të lartë
8. Nevojitet ndërfaqja minimale e përdoruesit
9. Kujtesë e kufizuar, kosto e ulët, më pak konsum të energjisë
10. Nuk ka nevojë për ndonjë memorie sekondare në kompjuter.

Internet of Things (IoT)

The Internet of things describes the network of physical objects—a.k.a. "things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.

Internet of Things (IoT) është një term që i referohet zgjerimit të ndërlidhjes së pajisjeve inteligjente, duke filluar nga pajisjet në sensorë të vegjël. Objektet japin informacione sensorit, sensorët veprojnë në mjedisin e tyre dhe, në disa raste, modifikojnë veten e tyre për të krijuar menaxhim të përgjithshëm të një sistemi më të madh, si një fabrikë ose qytet. IoT drejtohet kryesisht nga pajisje të ngulitura. Këto pajisje janë me gjerësi të ulët të brezit. Komunikojnë me njëra-tjetrën dhe sigurojnë të dhëna përmes ndërfaqeve të përdoruesit.

Duke iu referuar përkrahjes së sistemeve fundore, interneti ka shkuar drejt katër brezeve që të përkrah sistemet IoT:

1. **Teknologjia e informacionit (IT):** PC, servera, router, firewall, etj.,
2. **Teknologji operacionale (OT):** Makineri / pajisje me IT të ngulitur të ndërtuara nga kompani jo-IT, të tilla si makineri mjekësore, SCADA (kontroll mbikëqyrës dhe marrje e të dhënave), etj.
3. **Teknologji personale:** Telefonat inteligjentë, tabletët dhe lexuesit eBook të blera si pajisje IT nga konsumatorët.
4. **Teknologjia e sensorit / aktuatorit:** Pajisjet me një qëllim të caktuar të blera nga konsumatorët, IT dhe njerëzit të rendomt duke përdorur lidhje pa tel

Sistemet Operative te Mbyllura

Ekzistojnë dy qasje të përgjithshme për zhvillimin e një sistemi operativ të ngulitur (OS).

- Qasja e parë është marrja e një OS ekzistues dhe përshtatja e tij për aplikacionin e ngulitur. Për shembull, ekzistojnë versione të ngulitura të Linux, Windows dhe Mac, si dhe sisteme të tjerë operativë komerciale dhe pronësore të specializuar për sisteme të ngulitura.
- Qasja tjetër është të projektohet dhe zbatohet një OS i destinuar vetëm për përdorim të ngulitur. Një shembull i këtij të fundit është TinyOS, i përdorur gjerësisht në rrjetet e sensorit pa tel.

Struktura e një sistemi të ngulitur

Pavarësisht nga funksioni që kryen një sistem i ngulitur, struktura e tij mund të grupohet në dy grupe kryesore, me një bashkveprim të ngushtë mes përbërësve të tyre:

- Një grup i përbërësve të pajisjeve që përfshijnë një njësi qendrore të përpunimit, zakonisht në formën e një mikrokontrolluesi;
- Një seri programesh softuerësh që i japin funksionalitet harduerit.

Hyrjet tipike në një sistem të ngulitur janë variablat e procesit dhe parametrat që arrijnë përmes sensorëve dhe portave hyrëse/dalëse (I/O). Rezultatet janë në formën e veprimeve të kontrollit mbi aktuatorët e sistemit ose informacionin e përpunuar për përdoruesit ose nënsistemet e tjerë brenda aplikacionit. Softueri është pjesa më abstrakte e sistemit dhe po aq thelbësore sa vetë hardueri. Ai përfshin programet që diktojnë sekuencën sipas të cilës veprojnë përbërësit e harduerit.

Komponentët e harduerit:

Kur shikohen nga një këndvështrim i përgjithshëm, përbërësit e harduerit të një sistemi të ngulitur përfshijnë të gjithë elektronikën e nevojshme për sistemin për të kryer funksionin për të cilin është krijuar.

Pavarësisht nga këto ngjashmëri, tre përbërës thelbësorë të pajisjeve janë thelbësore në një sistem të ngulitur:

- Njësia Qendrore e Përpunimit (CPU)
- Memoria e sistemit
- Një sërë portesh hyrje-dalje

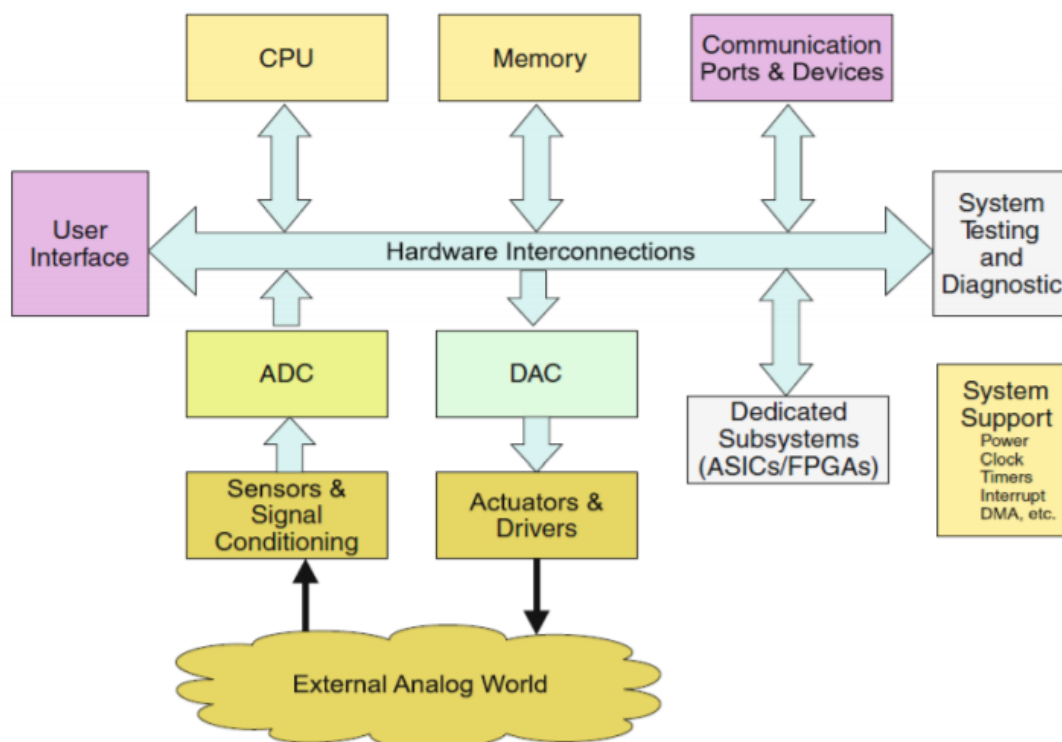
-**CPU** ekzekuton udhëzimet e softuerit për të përpunuar të dhënat hyrëse të sistemit dhe merr vendime që mundësojnë funksionimin e sistemit.

-**Memoria** ruan programet dhe të dhënat e nevojshme për funksionimin e sistemit.

Shumica e sistemeve bëjnë dallimin midis memories së programit dhe të dhënave. **Memoria e programit** ruan programet softuerike të ekzekutuara nga CPU. **Memoria e të dhënave** ruan të dhënat e përpunuara nga sistemi.

- **Portat I/O** lejojnë përcjelljen e sinjaleve ndërmjet CPU-së dhe botës së jashtme të saj.

Disa shembuj që përfaqësojnë **portat e komunikimit I / O** janë: Portat USB, portat e printerit, RF pa tel dhe portat infra të kuqe. Shembuj të **ndërfaqeve I/O të përdoruesit** janë: Tastierat, çelsat (switches), sinjalizuesit dhe audioja, dritat, ekranet numerike, alfanumerike dhe grafike.



-**Sensorët dhe Aktivizuesit** elektromekanikë shërbejnë për të bashkëvepruar me mjedisin e jashtëm të sistemit.

Sensorët sigurojnë hyrje në lidhje me parametrat fizikë siç janë: temperatura, presioni, zhvendosja, nxitimi, rrotullimi, etj. Aktivizuesit (actuators) marrin rezultatet nga portet I/O të sistemit, si për shembull: kontrolluesit e shpejtësisë së motorit, kontrolluesit e stepper të motorit, ndërruesit dhe drejtuesit e energjisë.

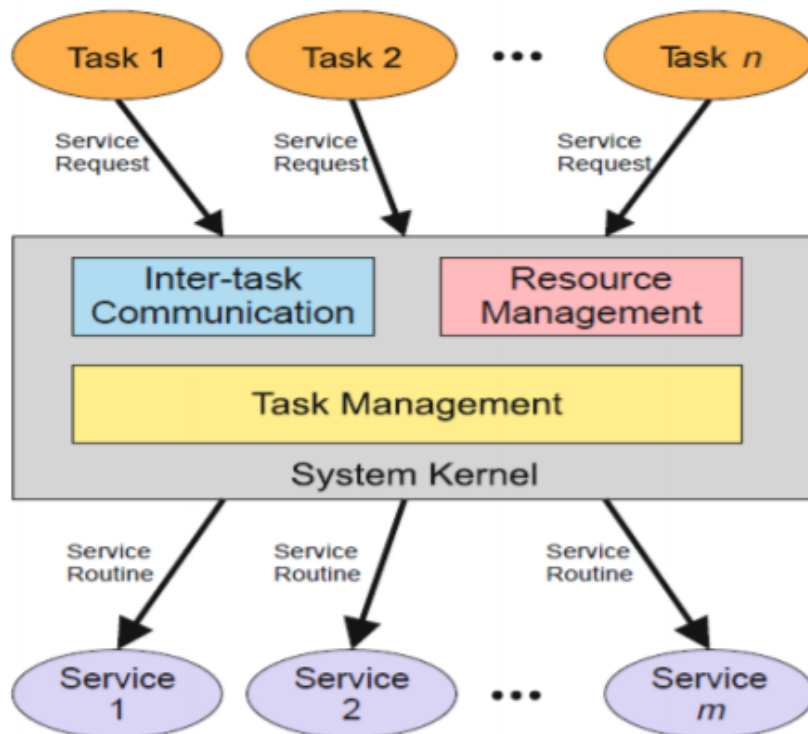
-**Konvertuesit e të dhënave** (Analog-në-dixhital (ADC) dhe / ose Digital-në-Analog (DAC)) shërbejnë për të lejuar bashkëveprimin me sensorë dhe aktuatorë analog. Kur sinjali që del nga një ndërfaqe e sensorit është analog, një ADC e kthen atë në formatin dixhital të kuptuar nga CPU. Në mënyrë të ngjashme, kur CPU duhet të komandojë një aktuator analog, kërkohet një DAC për të ndryshuar formatin e sinjalit.

-Komponentët e Diagnostikimit (System Testing and Diagnostic) shfrytëzohen për të verifikuar dhe siguruar funksionimin e fuqishëm dhe të besueshëm të sistemit.

-Komponentët e Mbështetjes (System Support) siguron shërbimet thelbësore që lejojnë funksionimin e sistemit. Pajisjet thelbësore të një System Support përfshijnë: komponentët e furnizimit me energji dhe menaxhim, dhe gjeneratorët e frekuencës së orës (clock). Komponentë të tjerë opsionalë të mbështetjes përfshijnë: kohëmatës, logjika e menagjimit të ndërprerjes (interrupts), kontrollorët DMA, etj.

Komponentët e softuerit:

Komponentët e softuerit të një sistemi të ngulitur përfshijnë të gjitha programet e nevojshme për t'i dhënë funksionalitet harduerit të sistemit. Këto programe, shpesh të referuara si **firmware** i sistemit, ruhen në një lloj memorie të qëndrueshme. Firmware nuk ka për qëllim të modifikohet nga përdoruesit, megjithëse disa sisteme mund të sigurojnë mjete për kryerjen e azhurnimeve. Sistemet operative mund të jenë të thjeshta dhe joformale në rastet e aplikacioneve të vogla, por ndërsa rritet kompleksiteti i aplikacioneve, sistemi operativ kërkon më shumë strukturë dhe formalitet. Në këto raste, dizajnet zhvillohen rreth Sistemeve Operative me Real-Time (RTOS).



Komponentët kryesorë të identifikuar në një softuer të sistemit përfshijnë:

- **Detyrat e sistemit (Tasks)**

Softueri në sistemet e ngulitura ndahet në një grup programesh më të vegjël të quajtur Detyra (Tasks). Çdo detyrë merret me një veprim të veçantë në sistem dhe kërkon përdorimin e burimeve specifike të sistemit. **Detyrat paraqesin kërkesa shërbimi në kernel** (service request) në mënyrë që të kryejnë veprimet e tyre të përcaktuara. Kërkesat e shërbimit mund të vendosen përmes regjistrave ose ndërprerjeve (interrupts).

- **Kernelin e sistemit (Kernel)**

Komponenti i softuerit që trajton burimet e sistemit në një aplikacion të ngulitur quhet Kernel. Burimet e sistemit janë të gjithë ata përbërës të nevojshëm për të shërbyer detyrat.

Këto përfshijnë:

- memorjen,
- pajisjet I/O,
- vetë CPU-në dhe përbërës të tjerë të pajisjeve.

Kerneli merr kërkesa për shërbime nga detyrat, dhe i planifikon ato sipas përparësive të diktuar nga menaxheri i detyrave (task management). Kur detyra të shumta pretendojnë për një burim të përbashkët, një pjesë e kernelit përcakton politikën e menaxhimit të burimeve të sistemit (resource management). Nuk është e pazakontë gjetja e detyrave që duhet të shkëmbejnë informacion midis tyre. Kerneli siguron një kornizë që mundëson një komunikim të besueshëm ndërmjet detyrave për të shkëmbyer informacion dhe për të koordinuar operacionin bashkëpunues.

- **Shërbimet (Services)**

Detyrat shërbehen përmes rutinave të shërbimeve (service routines). Një rutinë e shërbimit është një pjesë e kodit që i jep funksionalitet një burimi të sistemit. Në disa sisteme, ato referohen si drejtues të pajisjeve. Shërbimet mund të aktivizohen bazuar në votim (polling) ose si rutina të shërbimit të ndërprerjes (ISR), në varësi të arkitekturës së sistemit.

Klasifikimi i sistemeve të ngulitura

Ekzistojnë tri kategori përcaktuese që mund t'i përdorim për të klasifikuar sistemet e ngulitura:

- Sisteme të ngulitura të vogla
- Sisteme të ngulitura të shpërndara
- Sisteme të ngulitura me performancë të lartë

Sistemet e vogla të ngulitura:

Texas Instruments, me TMS1000 krijoi mikrokontrollerin, i cili është bërë përbërësi themelor i sistemeve të ngulitura.

Kjo klasë e sistemeve zakonisht përqendrohet rreth një çipi të vetëm të mikrokontrolluesit që komandon të gjithë aplikacionin. Këto sisteme funksionojnë me mirëmbajtje minimale ose pa

mirëmbajtje, janë me kosto shumë të ulët dhe prodhohen në sasi në masë. Softueri në këto sisteme është zakonisht me një detyrë të vetme, dhe rrallë kërkon një RTOS. Këto sisteme janë shumë të integruara, duke shtuar vetëm disa: përbërës analogë, sensorë, aktuatorë dhe ndërfaqen e përdoruesit, sipas nevojës. Shembuj të këtyre sistemeve përfshijnë sistemet e monitorimit të presionit të gomave, kontrollorët e furrave me mikrovalë, kontrollorët e furrave për përgaditjen e tosterëve, dhe kontrollorët elektronikë të lodrave, etj.

Sisteme e ngulitura të shpërndara:

Stili i krijuar nga Intel me 4004 është përfaqësues i këtij lloji të sistemeve të ngulitura.

Me sistemet e ngulitura të shpërndara i referohemi klasës së sistemeve të ngulitura ku, CPU qëndron në një çip të veçantë, ndërsa pjesa tjetër e komponentëve si memoriet, I/O, co-processors dhe funksionet e tjera të veçanta përhapen në një ose disa chip-a në atë që quhet zakonisht chipset i procesorit. Tek këto sisteme megjithëse qëndrueshmëria nuk është një çështje kritike, kërkojnë mirëmbajtje dhe azhurnim, dhe përfshijnë disa mjete të diagnozës së sistemeve. Këto sisteme zakonisht menaxhojnë detyra të shumta, kështu që përdorimi i RTOS për zhvillimin e sistemit nuk është i pazakontë. Vëllimi i prodhimit është relativisht i lartë dhe kostot drejtohen kryesisht nga niveli i pritur i performancës. Aplikimet mund të kërkojnë operacione me performancë të lartë. Shembuj të kësaj kategorie të sistemeve janë si: përpunuesit e videove, kontrolluesit e video lojërave, Logger-et e të dhënave dhe përpunuesit e rrjetit.

Sistemet e ngulitura me performancë të lartë:

Rasti i CADC (Central Air Data Computer) përfaqëson klasën e sistemeve të ngulitura shumë të specializuara që kërkojnë:

- llogaritje të shpejtë,
- tolerancë ndaj defekteve
- qëndrueshmëri të lartë.

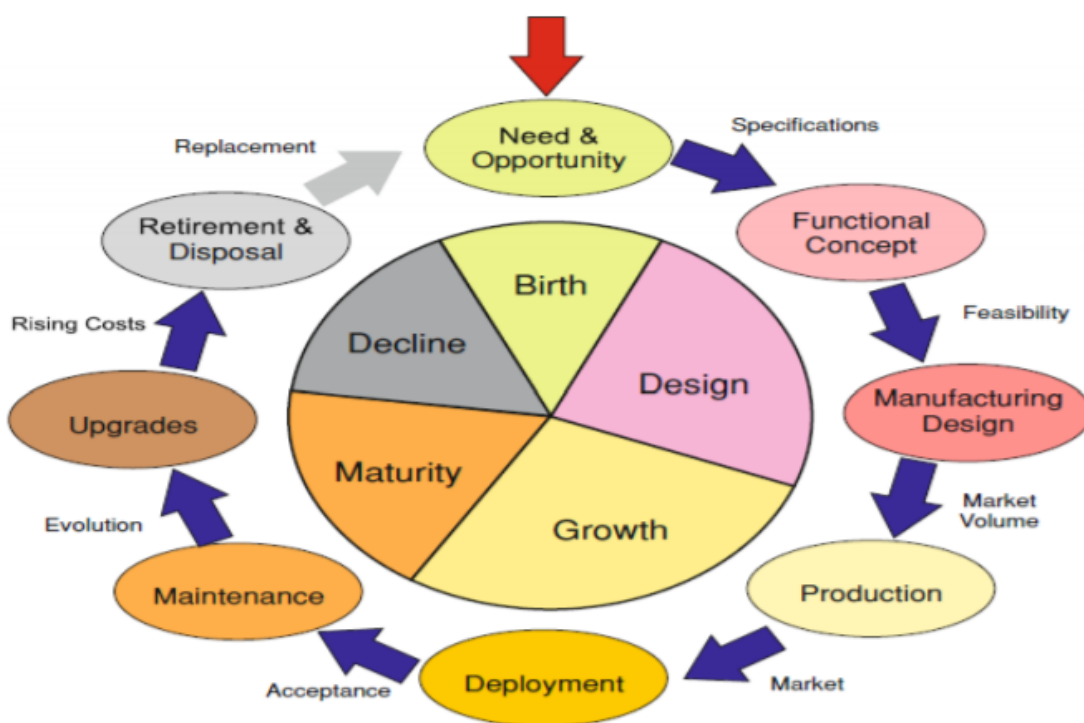
Këto sisteme zakonisht kërkojnë: ASICS të dedikuara, janë zakonisht të shpërndara, mund të përfshijnë DSP dhe FPGA si pjesë e pajisjes bazë. Në shumë raste, kompleksiteti i softuerit të tyre e bën të detyrueshëm përdorimin e RTOS-it për të menaxhuar shumëllojshmërinë e detyrave. Ato prodhohen në sasi të vogla dhe kostoja e tyre është shumë e lartë. Këto janë llojet e sistemeve të ngulitura të përdorura në aplikimet ushtarake dhe hapësinore, të tilla si: kontrollorët e fluturimit, sistemet e udhëzimit të raketave dhe sistemet e navigacionit për anijet hapësinore.

Midis këtyre llojeve të sistemeve të ngulitura mund të gjejmë "zona gri" ku karakteristikat e dy ose tre prej tyre mbivendosen dhe aplikimet mund të bëhen të vështira për t'u shoqëruar në një klasë të veçantë specifike. Sidoqoftë, nëse shohim gamën e gjerë të aplikacioneve të ngulitura, në shumicën e rasteve bëhet përgjithësisht e lehtë të identifikosh klasën, së cilës i përket një zbatim i veçantë.

Cikli jetësor i dizajnit të një sistemi të ngulitur

Sistemet e ngulitura kanë një jetëgjatësi të fundme gjatë të cilës ato i nënshtrohen fazave të ndryshme, duke filluar nga konceptimi ose lindja e sistemit tek asgjësimi dhe në shumë raste, rilindja. Ne e quajmë këtë cikël i jetës i një sistemi të ngulitur. Pesë faza mund të identifikohen në cikël:

1. Konceptimi ose Lindja,
2. Projektimi,
3. Rritja,
4. Pjekuria
5. Rënia.



-Lindja- Një sistem i ngulitur së pari konceptohet nga identifikimi i një nevojë për të zgjidhur një problem të veçantë, ose mundësia për të aplikuar teknologjinë e ngulitur në një problem të vjetër. Shumë sisteme të ngulitura janë konceptuar nga nevoja për të ndryshuar mënyrën se si zgjidhen problemet e vjetra ose nevoja e sigurimit të një zgjidhjeje me kosto më efektive.

-Dizajni- Në fazën e dizajnit, sistemi kalon së pari përmes një faze të dizajnit funksional, ku zhvillohen prototipa të provës së konceptit. Kjo fazë lejon të akordojmë funksionalitetin e sistemit që të përshtatet me aplikacionin për të cilin ishte krijuar. Analiza e cila merr në konsiderat: koston e produktit, tregun dhe vjetërsimin përcakton se një prodhim dhe dizajn i

produktit vazhdon apo jo. Dizajni është deri tani faza më e kushtueshme në ciklin jetësor të një sistemi të ngulitur, pasi që shumica e kostove jopërsëritëse të inxhinierisë lindin këtu.

-Faza e rritjes- fillon me prodhimin e sistemit për të furnizuar një kërkesë të vlerësuar të tregut. Kjo fazë përfshin: shpërndarjen, instalimin dhe konfigurimin e sistemit. Përshtatja e sistemit për nevojat e tij origjinale përcakton pranimin e tij, duke e çuar sistemin e ngulitur në fazën e tij të pjekurisë.

-Faza e pjekurisë- produkti mbahet funksional dhe i azhurnuar. Kjo përfshin sigurimin e mbështetjes teknike, mirëmbajtjen periodike dhe servisimin. Ndërsa aplikacioni evoluon, sistemi mund të kërkojë përshtatje për ndryshimin e nevojave të aplikacionit përmes programeve dhe/ose azhurnimeve të pajisjeve. Faza e pjekurisë pritet të lejojë drejtimin e produktit me kosto minimale. Ndërsa sistemi fillon të plakët, kostot e mirëmbajtjes fillojnë të rriten, duke sinjalizuar kalimin e tij në fazën e rënies.

-Faza e rënies- përbërësit e sistemit bëhen të vjetëruar dhe të vështirë për tu mbajtur, duke sjellë koston e mirëmbajtjes, shërbimit dhe përmirësimit në nivele që afrohet ose tejkalojnë ato të zëvendësimit të të gjithë sistemit. Kjo është pika ku sistemi duhet të tërhiqet nga shërbimi dhe të asgjësohet.

Kufizimet e Dizajnit

Funksionaliteti: Aftësia e sistemit për të kryer funksionin për të cilin është krijuar.

Kosto: Sasia e burimeve të nevojshme për të: konceptuar, dizajnuar, prodhuar, mirëmbajtur dhe hedhur poshtë një sistem të ngulitur.

Performanca: Aftësia e sistemit për të kryer funksionin e tij në kohë.

Madhësia: Hapësira fizike e marrë nga një zgjidhje e sistemit.

Fuqia dhe energjia: Energjia e kërkuar nga një sistem për të kryer funksionin e tij.

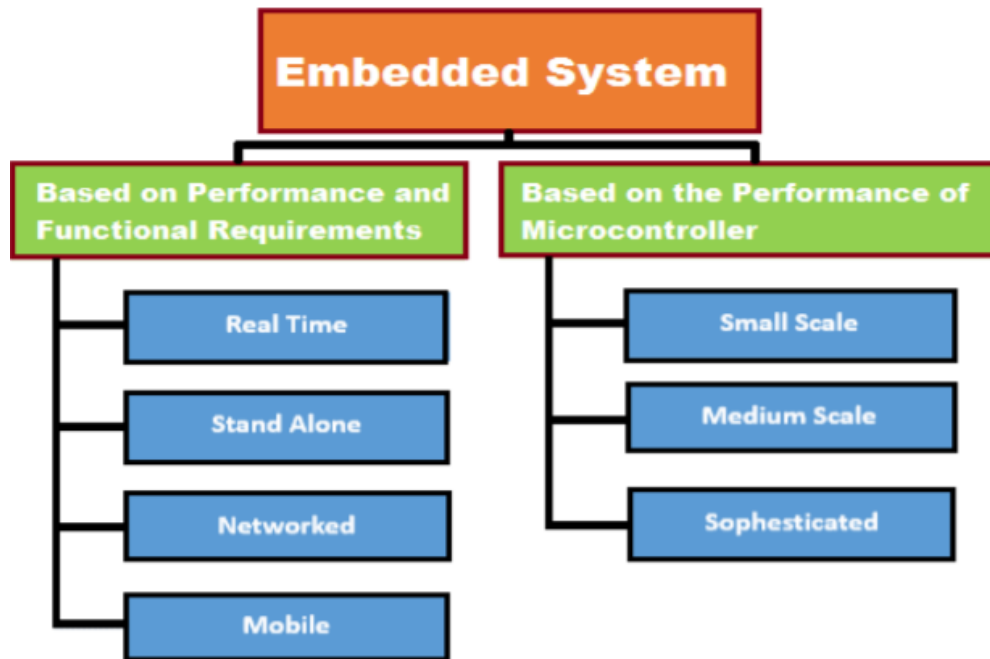
Koha për në treg: Koha që duhet nga konceptimi i sistemit deri tek vendosja.

Mirëmbajtja: Aftësia e sistemit që të mbahet funksional për sa më gjatë të jetë e mundur gjatë jetës së tij të pjekurisë.

Llojet/Klasifikimet e sistemit të ngulitur:

Ekzistojnë dy klasifikime të sistemit të ngulitur:

- Bazuar në performancën dhe kërkesat funksionale
- Bazuar në performancën e mikrokontrolluesit



-Bazuar në performancën dhe kërkesat funksionale-

- **Stand Alone Embedded Systems**

Në këtë lloj sistemi të ngulitur, nuk ka nevojë për ndonjë sistem pritës siç është kompjuteri, ndryshe procesori. Këto sisteme kryejnë vetë aktivitetet e tyre, dhe pastaj rezultatet e prodhuara merren përmes pajisjeve të lidhura dalëse, si monitorët ose printerët. Në këto lloje të sistemit të ngulitur, futja e të dhënave hyrëse nga portet në formën e sinjaleve dixhitale ose analoge, dhe ekzekutimi i tyre bëhet në vetë portin. Ky lloj sistemi ofron fleksibilitet dhe efikasitet sepse këto sisteme funksionojnë vetëm.

Shembuj:

- Lavatriçet dhe mikrovalët, sepse nuk kanë nevojë për ndonjë sistem kompjuterik për funksionimin e tyre.
- Kamera dixhitale
- Mp3 players
- Video game consoles
- Sistemet e matjes së temperaturës

- **Real-Time Embedded Systems**

Sistemet e ngulitura në kohë reale përdoren në zona të tilla ku ka nevojë për rezultat në një periudhë kohe specifike. Pra, ky sistem i ngulitur përdoret në sektorë: ushtarakë, mjekësorë dhe industrialë. Për të dizajnuar një sistem të ngulitur të tillë, duhet analizë e kohës së veprimit, dizajn me shumë detyra, korrigjim i gabimeve dhe testimi ndër-platformash dhe dizajni i

arkitekturës. Për të përmirësuar performancën dhe për të shmangur dështimet, në kuadër të harduerit duhet të vendoset sistemi i ngulitur.

Shembuj të aplikimit:

- Zbatimi për kontrollin e dritës në rrugë.
- Zbatimi për sistemin e kontrollit të semaforave.
- Sistemi i Kontrollit të Trafikut si:
 - autostradat, hapësira ajrore, linjat e transportit ose linja hekurudhore
- Sisteme me inteligjencë artificiale dhe robotikë.

Aplikimi në fushën e mjekësisë

Pajisjet mjekësore në ditët e sotme mbështeten shumë tek sensorët pasi ato ofrojnë ndihmë në: monitorimin, diagnostikimin dhe trajtimin e shëndetit mjekësor të pacientit. Disa nga fushat në mjekësi ku sistemet e ngulitura aplikohen janë:

- Skanuesit MRI dhe CT
- Sonografia
- Defibrillator
- Sensorë dixhitalë të rrjedhës (për të monitoruar sistemin e frymëmarrjes së pacientit)
- Pajisjet e presionit të gjakut
- Glucose Test Set
- Makina për monitorimin e zemrës fetale
- Pajisjet në gjendje që të monitorojnë shëndetin (Kjo i lejon përdoruesit të monitorojnë rrahjet e tyre të zemrës, presionin e gjakut, glukozën, peshën dhe shumë parametra të tjerë)

-MRI (Magnetic resonance imaging) dhe skanues CT-

Ky është një shembull numër një ku aplikohen sistemet e ngulitura në aplikacionet mjekësore. Sistemet e ngulitura përdoren në skanuesin MRI dhe CT i cili përdor impulse radio frekuence dhe rrezet x përkatësisht për të prodhuar fotografi të hollësishme të pjesëve të trupit dhe strukturave brenda trupit.

-Sonografia-

Sistemet e ngulitura përdoren gjithashtu në sonografi, të quajtura gjithashtu imazhe ultratinguj, të cilat përdor valë zanore me frekuencë të lartë për të marr imazhin e indeve të buta të tilla si muskujt dhe organeve të brendshme në kohë reale.

-Defibrillator-

Ky është një shembull i tretë i aplikacioneve mjekësore të cilat aplikojnë sisteme të ngulitura. Shumë pacientë me smundje të zemrës vdesin për shkak të arrestit kardiak i cili ndodh pa ndonjë paralajmërim. Pra, një sistem i ngulitur u fut në disa pajisje mjekësore të cilat përdoren për të zbuluar anomalitë kardiake. Pajisjet e tilla të ngulitura mjekësore përfshijnë defibrillator i cili është një makinë që përdoret për të monitoruar rrahjet e zemrës së një pacienti.

-Digital Flow Sensors-

Kjo është një pajisje e cila shërben për monitorimin e sistemit të frymëmarrjes së pacientit. Në rast të ndonjë frymëmarrje të parregullt, mjeku menjëherë informohet.

-Pajisja e presionit të gjakut dhe grupi i testit të glukozës-

Sistemet e ngulitura janë instaluar gjithashtu në aparaturat për matje të presionit të gjakut që janë efikase në zbulimin e presionit sistolik dhe diastolik të trupit të njeriut dhe për testimin e nivelit të glukozës në trupin e njeriut.

-Makinë për monitorimin e zemrës fetale-

Një pajisje tjetër e ngulitur mjekësore është makina për monitorimin e zemrës fetale e cila përdoret gjatë shtatzënisë dhe lindjes e cila aplikohet për të mbajtur nën kontroll rrahjet e zemrës së foshnjës, e cila tregon nëse foshnja është mirë apo jo.

Ndarja e sistemeve të ngulitura në kohë reale

Sistemet e ngulitura në kohë reale ndahet në dy kategori:

- Soft Embedded Systems
- Hard Embedded Systems

Soft Embedded Systems - Ky lloj sistemi, i konsideron proceset si detyrë kryesore, kontrollojnë plotësisht detyrën. Soft embedded system është një sistem funksionimi i cili priset nëse rezultatet nuk prodhohen sipas kërkesës së specifikuar të kohës.

Shembull:

- Multimedia Transmission and Reception
- Computer Games

Hard Embedded Systems - Ky lloj sistemi siguron që të gjitha proceset kritike të jenë përfunduar brenda kornizës kohore të dhënë. Kjo do të thotë që të gjitha vonesat në sistem janë të kufizuara në kohë. Gjithashtu, ka pak ose aspak memorie sekondare dhe të dhënat ruhen në kujtesën afatshkurtër.

Shembull:

- Sistemi i Kontrollit të Avionëve
- Sistemi i kontrollit të motorit automobilistik dhe frenimi kundër bllokimit

HARD REAL TIME SYSTEM	SOFT REAL TIME SYSTEM
In hard real time system, the size of data file is small or medium.	In soft real time system, the size of data file is large.
In this system response time is in millisecond.	In this system response time are higher.
Peak load performance should be predictable.	In soft real time system, peak load can be tolerated.
In this system safety is critical.	In this system safety is not critical.
A hard real time system is very restrictive.	A Soft real time system is less restrictive.
In case of an error in a hard real time system, the computation is rolled back.	In case of an soft real time system, computation is rolled back to previously established a checkpoint.
Satellite launch, Railway signaling system etc.	DVD player, telephone switches, electronic games etc.

Network Embedded Systems

Nëse, ndonjë program ekzekutohet në pajisjet e tjera, si një formë rrjeti, atëherë quhet "Sistemet e ngulitura në rrjet" por këtu, ka nevojë për mikroprocesorin ose kontrolluesin për të kontrolluar ato programe në distancë në rrjet. Ky lloj sistemi i ngulitur po rritet më shpejt, për shkak të fleksibilitetit të tyre më të mirë. Rrjeti i tërë menaxhohet dhe aksesohet nga shfletuesi i internetit.

Llojet më efikase të rrjetit të përdorur në sistemin e ngulitur janë rrjeti BUS dhe një rrjet Ethernet.

Shembull:

- Sistemet e sigurisë në çdo organizatë ose zyrë të vogël.

Mobile Embedded Systems

Të gjitha pajisjet të cilat janë të lëvizshme (pa ndonjë kufizim kohe dhe vendi) dhe që aplikojnë sistemin e ngulitur, njihen si "Sisteme të Ngulitura Mobile".

Sistemet e ngulitura mobile përdoren në pajisje të lëvizshme si:

- telefona celulare,
- kamerat dixhitale,
- mp3 player
- ndihmësit dixhitalë personalë, etj.

Kufizimi themelor i këtyre pajisjeve janë burimet e tjera dhe kufizimi i kujtesës.

Aplikimi në sektorin e automjeteve

Në sistemet e automobilave gjithnjë e më shumë pajisjet po ndryshojnë nga sistemet mekanike në sistemet elektronike. Sistemi i ngulitur është zemra e sistemit elektronik të një automjeti për shkak të shkathtësisë dhe fleksibilitetit të tij. Sot një automjet tipik përmban rreth 25 deri në 35 mikrokontrollues, dhe disa automjete luksoze përmbajnë afërsisht 60 deri në 70 mikrokontrollues për automjet. Sot, një automobil tipik në rrugë ka sisteme elektronike të kontrolluara nga kompjuteri, dhe sistemet e ngulitura.

Aplikimi i këtyre sistemeve mund të gjendet në:

- Sistemet e kontrollit të jastëkëve (Airbag)
- Sistemet e navigacionit
- Adaptive Cruise Control
- Sistemin e ngulitur të ndjeshmërisë ndaj shiut
- Sensorë rezervë (Backup)
- Kontrollin e pezullimit (Suspension)
- Sistemin e kontrollit motorik
- Sistemet për argëtim dhe multimedia
- Sistemin e kontrollimit të xhamave elektronike të automjetit
- Timonin elektrik të automjetit
- Sistemin e injektimit të karburantit të automjetit
- Sistemin e frenimit kundër bllokimit

- Kutinë e zezë
- Kontrollin e emitimit të gazrave
- Kontrollin e tërheqjes
- Parkimin automatik
- Monitorimin e presionit të gomave
- Kontrollin e klimës, etj.

Embedded Airbag System

Kjo është një pajisje e rëndësishme e sigurisë që siguron mbrojtje shtesë kundër përplasjes ballë për ballë për udhëtarët në ulset e para. Ky sistem punon bazuar në komandat nga mikrokontrolluesi. Kontrolluesi i këtij sistemi merr energji nga bateria. Nëse sensorët zbulojnë një aksident, ky mikrokontrollues aktivizon sistemin e airbags duke përdorur alternatorin.

Embedded Navigation System

Një përparim tjetër i sistemit të ngulitur në automobila është sistemi i navigacionit që funksionon duke përdorur sistemin GPS. Ky sistem navigimi përbëhet nga një: qark i ngulitur i ndërtuar me një marrës GPS, një xhiroskop (speed/direction sensor), një DVD-ROM (map and voice data), kontrollues kryesor dhe një sistem shfaqjeje. Marrësi GPS merr vlerat aktuale të gjatësisë dhe gjerësisë që krahasohen me hartën e ruajtur. Xhiroskopi dhe sensorët e tjerë sigurojnë drejtimin dhe shpejtësinë e rrugës. Nga të gjitha informacionet e mbledhura në kontrolluesin kryesor, sistemi i ekranit shfaq një ekran navigimi ose rrugë të destinacionit në ekranin e automjetit.

Adaptive Cruise Control

Risi e sistemit të ngulitur të përdorur në automjete është teknologjia Adaptive Cruise Control. Duke përdorur këtë teknologji, mund të bëjmë kontrollin e automjeteve pa veprime të shoferit dhe shumë prodhues të automjeteve janë gjithashtu të angazhuar tashmë në punën për këto koncepte. Ky kontroll i lëvizjes ju mundëson makinave të mbajnë distancën e sigurisë nga automjetet e tjera në rrugë si në p.sh. autostradë. Drejtuesi i makinës mund të vendosë shpejtësinë e automjetit të tij dhe distancën ndërmjet automjetit të tij dhe automjeteve të tjera. Kur trafiku ngadalësohet, ACC ndryshon shpejtësinë e automjetit duke përdorur frenim të moderuar. Çdo makinë ka një laser si marrës ose një radar mikrovalorë i cili është i fiksuar përpara makinës për të matur shpejtësinë dhe distancën e çdo automjeti tjetër në trafik. Ky proces punon sipas parimit të Efektit të Doppler-it; Ky proces nuk është gjë tjetër veçse ndryshim në frekuencën e valëve.

Embedded Rain-Sensing System

Sistemi i ngulitur i ndjeshmërisë ndaj shiut në automjet është një tjetër sistem automatizimi i implementuar me përdorimin e sistemit elektronik. Një sensor optik vendoset në një zonë të vogël të xhamit të përparmë. Ky sensor optik vendoset në një kënd ku realizon lëshim të rrezeve infra të kuqe si dhe lexon sasinë e dritës pas reflektimit prapa. Kjo dritë reflektohet në rastet kur

xhami i përparmë është i lagësht ose i ndotur. Kështu sensori optik varësisht nga drita e reflektuar në sensor përcakton shpejtësinë e nevojshme dhe frekuencën e fshirëseve të xhamit.

Sistemi Automatik i Parkimit

Ky sistem është një sistem automatik i pavarur i parkimit i cili mundëson manipulimin me automjetet duke mundësuar lëvizjen e automjetit nga korsia e trafikut në një vend parkimi. Ky sistem kryesisht përdor metoda të ndryshme për të zbuluar objekte përreth makinës si sensorët e instaluar në pjesën e përparme të automjetit dhe pjesën e pasme si transmetues dhe marrës. Këta sensorë dërgojnë një sinjal që do të replikohet përsëri kur të takojë një pengesë afër automjetit. Në këtë rast pas kalkulimeve do të caktohet saktë pozicioni i pengesës. Automjeti do të përcaktoj hapësirën e parkimit dhe distancën nga ana e rrugës, pastaj do ta ngas automjetin në vendin e parkimit. Kështu, sistemet e ngulitura në ditët moderne kanë shënuar një ndryshim revolucionar në çdo aspekt të proceseve të modelimit dhe prodhimit të automobilitave për shkak të adaptueshmërisë dhe fleksibilitetit të tyre.

Qëllimet e aplikimit të sistemeve të ngulitura

Secili sistem i integruar është dizajnuar për të realizuar një detyrë specifike si:

- Grumbullim/Ruajtjen/Përfaqësimin e të dhënave
- Komunikim të të dhënave
- Përpunim të të dhënave
- Monitorim
- Kontrollim
- Ndërfaqësim specific.

Grumbullimi i të dhënave zakonisht bëhet për: **ruajtje, analizë, manipulim dhe transmetim.**

Termi, ‘të dhëna’ i referohet gjërave të matshme si: teksti, zëri, imazhi, sinjalet elektrike, etj.

Të dhënat mund të jenë **analoge** (të vazhdueshme) ose **dixhitale** (diskrete).

Sistemi i ngulitur me teknikat e kapjes së të dhënave analoge mbledh të dhëna direkt në formë analog dhe shndërron sinjalin analog në dixhital duke përdorur konvertuesit A/D dhe më pas mbledh ekuivalentin binar të të dhënave analoge. Nëse sinjali është dixhital ai mund të kapet direkt pa ndonjë ndërfaqe shtesë nga sistemi dixhital i ngulitur. Të dhënat e mbledhura mund të ruhen drejtpërdrejt në sistem ose mund të transmetohen gjithashtu edhe në sistemet e tyre ose mund të përpunohen nga sistemi ose mund të fshihen menjëherë pasi të keni dhënë një paraqitje kuptimplote.

Një aparat fotografik dixhital është një shembull tipik i një sistemi të ngulitur me mbledhjen/ruajtjen/paraqitjen e të dhënave. Imazhet kapen dhe imazhi i kapur mund të ruhet në kujtesën e kamerës. Imazhi i kapur gjithashtu mund t'i paraqitet përdoruesit përmes njësisë së ekranit LCD.

Data Communication

Sistemet e integruara të komunikimit të të dhënave janë zhvilluar në aplikacione duke filluar nga sistemet komplekse të komunikimit satelitor në sistemet e thjeshta të rrjetave shtëpiake.

Data (Signal) Processing

Të dhënat e mbledhura nga sistemi i ngulitur mund të përdoren për lloje të ndryshme të përpunimit të sinjalit. Një aparat dëgjimi dixhital është një shembull tipik i një sistemi të ngulitur që përdor përpunimin e të dhënave.

Monitoring

Të gjitha produktet e ngulitura që vijnë nën domenin mjekësor janë me funksione monitorimi vetëm. Ato përdoren për përcaktimin e gjendjes së disa variablave duke përdorur sensorë të vendosur. Një shembull shumë i mirë është makina e elektrokardiogramës (EKG) për monitorimin e zemrës së pacientit.

Control

Sistemi i ngulitur me funksionalitete kontrolli imponon kontroll mbi disa variabla duke regjistruar ndryshoret hyrëse. Një sistem me funksionalitet kontrolli përmban edhe sensorë edhe aktuatorë. Sensorët janë porta hyrëse për kapjen e ndryshimeve në ndryshoret e mjedisit ose ndryshoren matëse. Aktuatorët janë portet dalëse që kontrollohen në përputhje me variablat hyrëse.

Application Specific User Interface

Këto janë sisteme të ngulitura me aplikacione, ndërfaqe specifike të përdoruesit, si: butona, çelësa, tastierë, drita, zile, njësi shfaqjeje, etj. Telefoni celular është një shembull për këtë, në telefonin celular ndërfaqja e përdoruesit sigurohet përmes: tastierës, modulit grafik LCD, altoparlantit të sistemit, alarmit të dridhjeve, etj.

Organizimi i Mikrokompjuterit

Bashkësia minimale e komponentëve të kërkuar për të krijuar një sistem informatik është emërtuar me termin Mikrokompjuter.

Struktura bazë e mikrokompjuterit

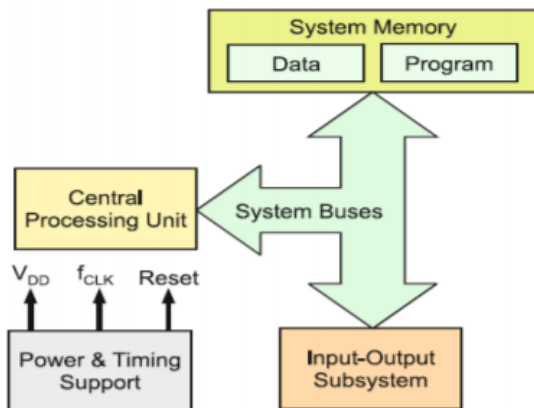
Konfigurimi minimal i harduerit për një sistem mikrokompjuterik është i përbërë nga tre komponentë themelorë:

- Njësia Qendrore e Përpunimit (CPU)

- Memoria e sistemit
- Ndërfaqet Input / Output (I / O).

Këto komponentë janë të ndërlidhur nga grupe të shumta të linjave të grupuara sipas funksioneve të tyre, të cilat njihen si **bus të sistemit**. Një grup shtesë i komponentëve siguron sinkronizimin e duhur të energjisë dhe kohën për funksionimin e sistemit.

Komponentët e një mikrokompjuteri mund të implementohen në mënyra të ndryshme. Ato mund të vendosen me chips të shumët në një bord të mikrokompjuterit ose të integrohen në një çip të vetëm, në një strukturë të quajtur mikrokompjuter në çip ose thjesht një mikrokontroller. Në ditët e sotme, shumica e sistemeve të ngulitura janë zhvilluar rreth mikrokontrollerëve.



Njësia qendrore e përpunimit (CPU)

CPU-ja formon zemrën e sistemit të mikrokontrollerit. Ajo tërheq udhëzimet nga memoria e programit, i deshifron ato dhe në përputhje me rrethanat, operon në të dhëna dhe/ose në pajisjet periferike në nënsistemin Input-Output për t'i dhënë sistemit funksionalitetin.

Memoria e Sistemit

Vendi ku ruhen programet dhe të dhënat për t'u qasur nga CPU-ja është memoria e sistemit. Brenda sistemit identifikohen dy lloje të elementeve të kujtesës: Program Memory dhe Data Memory.

Program Memory ruan programet në formën e një sekuence udhëzimesh. Programet diktojnë funksionimin e sistemit.

Data Memory ruan të dhënat që do të përdorën nga programet.

Nënsistemi Input/Output

Nënsistemi I/O, i quajtur ndryshe Nënsistem Periferik përfshin të gjithë përbërësit ose pajisjet periferike që lejojnë CPU-në të shkëmbejë informacion me pajisje të tjera, sisteme ose botën e jashtme.

Bus-et e Sistemit

Grupi i linjave që ndërlidhin CPU-në, Memorian dhe Nën sistemin I / O janë emërtuar bus-et e sistemit. Grupet e linjave në bus-et e sistemit kryejnë funksione të ndryshme.

Bazuar në funksionin e tyre, linjat e sistemit të bus-eve ndahen në:

- bus adresash
- bus të dhënash
- bus kontrolli.

Njësitë e Mikroprocesorit

Një njësi e Mikroprocesor-it (MPU), përmban një CPU me qëllim të përgjithshëm. Për të zhvilluar një sistem bazë duke përdorur një MPU, nevojiten të gjithë përbërësit e përshkruar më lartë, përveç CPU-së, dmth: Bus-et, Memoria dhe Ndërfaqet I/O. Karakteristika të tjera të MPU-ve përfshijnë një arkitekturë të optimizuar për të lëvizur kodin dhe të dhëna nga memoria e jashtme në chip siç janë memoriet cach.

Shembujt më të zakonshëm të sistemeve të dizajnuara rreth një MPU janë kompjuterët personalë dhe kornizat kryesore. Ka edhe shumë sisteme të tjera të zhvilluara rreth MPU-ve tradicionale.

Prodhuesit e MPU përfshijnë: INTEL, Freescale, Zilog, Fujitsu, Siemens dhe shumë të tjerë.

Sidoqoftë, shumica e sistemeve të vogla të ngulitura nuk kanë nevojë për fuqi të madhe

llogaritëse dhe përpunuese që karakterizojnë mikroprocesorët, dhe për këtë arsye orientimi është drejtuar drejt mikrokontrolluesve.

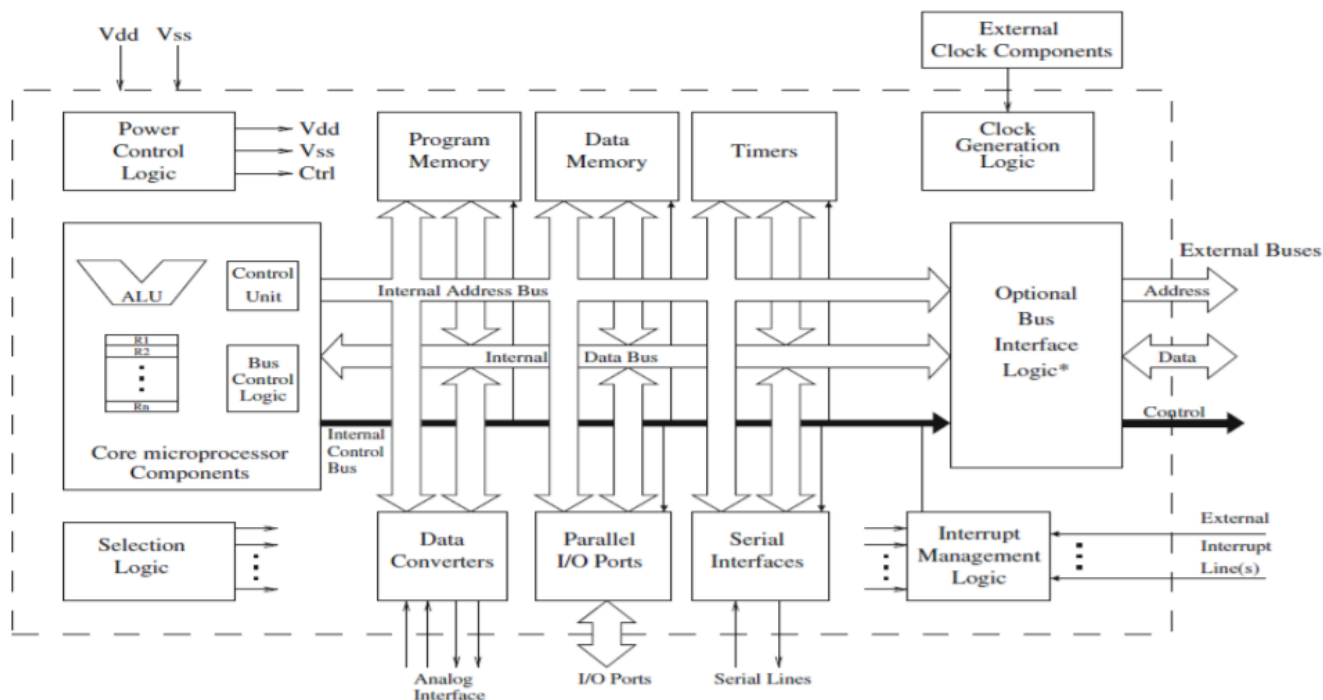
Njësitë e Mikrokontrollerit

Një njësi e mikrokontrollerit (MCU), është zhvilluar duke përdorur një bërthamë mikroprocesori ose njësi përpunuese qendrore (CPU), zakonisht më pak komplekse se ajo e një MPU. Kjo CPU bazë rrethohet me memorie të të dy llojeve (memoria për program dhe të dhëna) dhe disa lloje të pajisjeve periferike, të gjitha të ngulitura në një qark të vetëm të integruar, ose çip. *Kjo përzierje e CPU-së. Memories dhe I/O brenda një çipi të vetëm është ajo që e quajmë mikrokontroller.*

Ndër komponentë tjerë që gjenden zakonisht në shumicën e mikrokontrollerve janë: kohëmatësit periferikë, portat hyrëse/dalëse (I/O), mbajtësit e ndërprerjeve dhe konvertuesit e të dhënave.

Sigurimi i një shumëllojshmërie të tillë të burimeve brenda të njëjtit mikroprocesor është ajo që u ka fituar atyre emërtimin e kompjuterëve në një çip.

Struktura e një mikrokontrolleri tipik:



Mikrokontrolleret ndajnë një numër karakteristikash me mikroprocesorët me qëllim të përgjithshëm. Megjithatë, përbërësit kryesorë arkitektonikë në një MCU tipike janë më pak komplekse dhe më të orientuara drejt aplikimit sesa ato në një mikroprocesor me qëllim të përgjithshëm. Mikrokontrolluesit zakonisht tregtohen si anëtarë të familjes. Çdo familje është zhvilluar rreth një arkitekture themelore e cila përcakton karakteristikat e përbashkëta për të gjithë anëtarët.

Këto përfshijnë, ndër të tjera: gjerësinë e të dhënave dhe shtigjeve të programit, stilin arkitektonik, strukturën e regjistrit, grupin bazë të udhëzimeve dhe mënyrat e adresimit. Karakteristikat që dallojnë anëtarët e familjes përfshijnë: sasinë e të dhënave në chip dhe memorien e programit, dhe shumëllojshmërinë e pajisjeve shtesë periferike.

/*

One of the main differences between microcontrollers and microprocessors is that a microprocessor will typically run an operating system. An operating system allows multiple processes to run at the same time via multiple threads. Drivers are required to support peripherals.

A microcontroller will run a "bare metal interface," which means there is not an operating system. Without an operating system, a microcontroller can only run one

control loop at a time. From a software perspective, this means a single thread is running on the microcontroller's processor or Central Processing Unit (CPU).

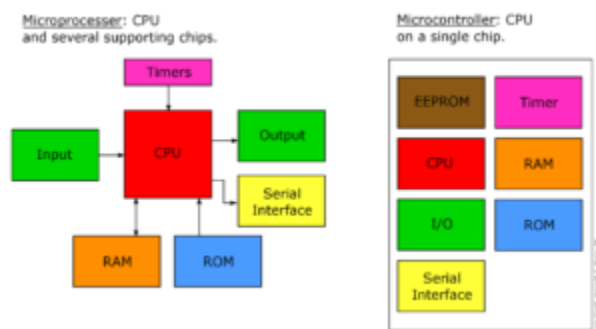


Figure 1: An MCU provides more on a single chip than an MPU. Here, "CPU" refers to a Central Processing Unit, or simply "processor." A CPU is the brains of any computing device.

*/

Arkitektura RISC vs CISC

Sistemet e mikrokompjuterëve funksionojnë me një softuer i cili mbështetet nga arkitektura e tij harduerike. Nën këtë këndvështrim, ne flasim për arkitekturat CISC dhe RISC.

Makineritë CISC (Complete Instruction Set Computing) karakterizohen nga fjalë udhëzuese me gjatësi të ndryshueshme, d.m.th. me numër të ndryshëm bitësh, madhësi të vogla kodi dhe udhëzime të shumëfishta komplekse në nivelin e makinës. Arkitektura CISC përqendrohet në arritjen e secilit udhëzim, në mënyrë që të gjenerojë programe të thjeshta. Ky fokus ndihmon detyrën e programuesit ndërsa rrit kompleksitetin e harduerit.

Nga ana tjetër, **makinat RISC** (Reduced Instruction Set Computing) janë krijuar me fokus në udhëzime të thjeshta, por kjo rezulton në programe më të gjata. Ky orientim thjeshton strukturën e harduerit. Dizajni pret që çdo ekzekutim i vetëm i udhëzimeve të zvogëlohet - më së shumti një cikël i vetëm i memories së të dhënave - kur krahasohet me "udhëzimet komplekse" të një sistemi CISC.

Programmer and Hardware Model

Programuesit e sistemeve të ngulitura duhet të bëjnë një hap përpara dhe të marrin në konsideratë çështjet e harduerit dhe softuerit. Prandaj, ata duhet të shikojnë sistemin si nga pikëpamja e harduerit, modeli i harduerit, ashtu edhe nga pikëpamja e softuerit, modeli i programuesit.

Në **modelin e harduerit**, përdoruesi përqendrohet në:

- karakteristikat e harduerit
- nënsistemet që mbështesin udhëzimet

- bashkëveprimet me botën e jashtme.

Në këtë model ne flasim për:

- nënsistemet e harduerit
- karakteristikat e pajisjeve periferike
- ndërfaqen me memorien
- pajisjet periferike dhe botën e jashtme
- kohën, etj

Hardueri mbështet modelin e programuesit.

Në modelin e programuesit, ne përqendrohemi në:

- grupin e udhëzimeve dhe sintaksat
- mënyrat e adresimit
- mapimin e memories
- kohën e transferimeve dhe ekzekutimit, etj.

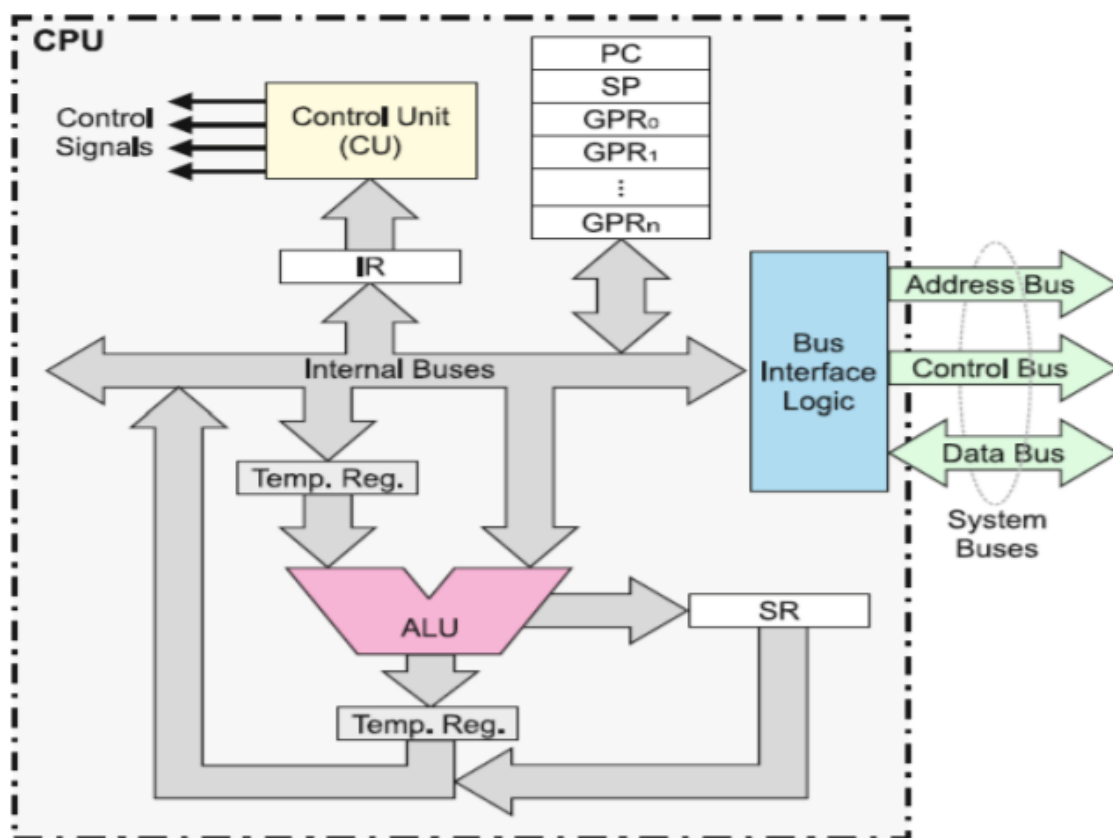
Shumë shpesh, kur një mikrokontrollues projektohet nga e para, procesi fillon me grupin e dëshiruar të udhëzimeve.

Njësia qendrore e përpunimit (CPU)

Njësia Qendrore e Përpunimit (CPU) në një sistem mikrokompjuterik është tipikisht një njësi mikroprocesor-i (MPU) ose bërthamë. CPU është njësia e cila realizon përpunimin e udhëzimeve të marra nga njësit tjera.

Lista minimale e përbërësve që përcaktojnë arkitekturën e një CPU përfshin:

- **Komponentët e harduerit**
 - *Një njësi logjike aritmetike (ALU)*
 - *Një njësi kontrolli (CU)*
 - *Një set regjistrash*
 - *Ndërfaqja logjike e busit (BIL)*
- **Komponentët e softuerit**
 - *Bashkësia e udhëzimeve*
 - *Mënyrat e Adresimit*



Këta përbërës lejojnë që CPU-ja të ketë qasje në programet dhe të dhënat e ruajtura diku në memorie ose nënsistemin e hyrjes / daljes, dhe të veprojë si një kompjuter i programit të ruajtur. Sekuenca e udhëzimeve që bëjnë një program zgjidhen nga grupi i udhëzimeve të procesorit.

Një program i ruajtur në memorie dikton sekuencën e operacioneve që do të kryhen nga sistemi. Në përpunimin e të dhënave, secili komponent i CPU-së luan një rol të domosdoshëm që plotëson ato të të tjerëve.

-Njësia e Kontrollit (CU)-

Njësia e kontrollit (CU) menaxhon funksionimin e CPU-së duke punuar si një makinë e kufizuar që ciklet përgjithmonë të kalojnë nëpër tri gjendje: marrja, dekodimi dhe ekzekutimi. Ky cikël i ekzekutimit, marrjes dhe dekodimit njihet gjithashtu si cikël udhëzimi ose cikël i CPU-së.

Cikli mund të përshkruhet si më poshtë:

1. **Gjendja e marrjes:** Gjatë gjendjes së marrjes një udhëzim i ri sillet nga memoria në CPU përmes ndërfaqës logjike të magjistralsë (BIL). Numëruesi i programit (PC) ofron adresën e udhëzimit që do të merret nga memoria. Udhëzimi i sapo marrë është lexuar përgjatë magjistralsë së të dhënave dhe më pas ruhet në regjistrin e udhëzimeve (IR).
2. **Gjendja e dekodimit:** Pas marrjes së udhëzimit, CU kalon në një gjendje dekodimi, ku deshifrohet kuptimi i udhëzimit. Informacioni i dekoduar përdoret për të dërguar sinjale te përbërësit e duhur të CPU-së për të ekzekutuar veprimet e specifikuar nga udhëzimi.
3. **Gjendja e ekzekutimit:** Në gjendjen e ekzekutimit, CU komandon njësitë përkatëse funksionale të CPU-së për të kryer veprimet e specifikuar nga udhëzimi. Në fund të fazës së ekzekutimit, PC është rritur për të treguar adresën e udhëzimit tjetër në memorie.

Pas fazës së ekzekutimit, CU urdhëron BIL të përdorë informacionin në numëruesin e programit për të marrë udhëzimin tjetër nga memoria, duke filluar përsëri ciklin. Duke qenë se CU është një makinë e fundme, i duhet një sinjal Reset për të filluar ciklin për herë të parë.

-Njësia Logjike Aritmetike (ALU)-

Njësia logjike aritmetike (ALU) është njësi përbërëse e CPU-së ku kryhen të gjitha veprimet logjike dhe aritmetike të mbështetura nga sistemi. Operacionet themelore aritmetike si mbledhja, zbritja dhe komplimenti, mbështeten nga shumica e ALU-ve. Disa mund të përfshijnë gjithashtu pajisje për operacione më komplekse siç janë shumëzimi dhe pjesëtimi. Operacionet logjike të kryera në ALU mund të përfshijnë veprime logjike AND, OR, NOT, dhe XOR, si dhe operacione të regjistrimit si SHIFT dhe ROTATE.

-Bus Interface Logic (BIL)-

Bus Interface Logic (BIL) i referohet strukturave të CPU-së që koordinojnë ndërveprimin midis magjistrave të brendshme dhe magjistrave të sistemit. BIL përcakton se si funksionojnë adresimet e jashtme, të dhënat dhe magjistralet e kontrollit. Në sistemet e vogla të ngulitura, BIL përmbahet plotësisht brenda CPU-së dhe transparente për projektuesin. Në sistemet e shpërndara

dhe me performancë të lartë, BIL mund të përfshijë pajisje shtesë periferike të dedikuara për të vendosur ndërfaqen e CPU-së në magjistralen e sistemit.

-Registers-

Regjistrat e CPU sigurojnë ruajtje të përkohshme për të dhënat, adresat e memories dhe informacionin e kontrollit në një mënyrë që mund të arrihet shpejt. Ato janë forma më e shpejtë e ruajtjes së informacionit në një sistem kompjuterik, ndërsa në të njëjtën kohë ato janë më të voglat në kapacitet. Përmbajtja e regjistratit është e paqëndrueshme, që do të thotë se ajo humbet kur CPU nuk furnizohet me energji. Regjistrat e CPU mund të klasifikohen si regjistra me qëllime të përgjithshme dhe qëllime të specializuara.

Regjistrat me qëllim të përgjithshëm (GPR) janë ata që nuk janë të lidhur me funksione specifike të procesorit dhe mund të përdoren për të ruajtur të dhëna, variabla ose tregues adresash sipas nevojës. Bazuar në këtë përdorim, disa autorë i klasifikojnë ata gjithashtu si regjistra të të dhënave ose adresave. Në varësi të arkitekturës së procesorit, një CPU mund të përmbajë nga dy deri në disa GPR.

Regjistrat me qëllim të veçantë kryejnë funksione specifike që i japin funksionalitet CPU-së. Struktura më themelore e CPU-së përfshin katër regjistrat e specializuar të mëposhtëm:

- Instruction Register (IR)
- Program Counter (PC), also called Instruction Pointer (IP)
- Stack Pointer (SP)
- Status Register (SR)

-Instruction Register (IR)-

Ky regjistër mban udhëzimin që aktualisht po deshifrohet dhe ekzekutohet në CPU. Veprimi i transferimit të një instruksioni nga memoria në IR quhet marrje udhëzimi. Në shumë sisteme të vogla të ngulitura, IR mban një udhëzim në të njëjtën kohë. CPU-të e përdorura në sistemet e shpërndara dhe me performancë të lartë zakonisht kanë regjistra të shumëfishtë udhëzimesh të rregulluar në një radhë, duke lejuar lëshimin e njëkohshëm të udhëzimeve për njësi të shumta funksionale. Në këto arkitektura IR zakonisht quhet radhë udhëzimi.

-Program Counter (PC)-

Ky regjistër mban adresën e udhëzimit që CPU do të marr nga memoria. Ndonjëherë quhet edhe treguesi i udhëzimeve (IP). Sa herë që një udhëzim merret dhe deshifrohet, njësia e kontrollit rrit vlerën e PC për të treguar udhëzimin tjetër në memorie. Duke qenë PC një regjistër adresash, gjerësia e tij gjithashtu mund të përcaktojë madhësinë e hapësirës më të madhe të memories së programit që drejtohet drejtpërdrejt nga CPU.

-Stack Pointer (SP)-

Stack është një segment i specializuar i memories që përdoret për ruajtjen e përkohshme të të dhënave në një sekuencë të veçantë. Operacionet e ruajtjes dhe rikuperimit të artikujve sipas kësaj sekuence menaxhohen nga CPU me regjistrin e treguesit të stack (SP). Sidoqoftë, shumica e modeleve, i lejojnë përdoruesit të përcaktojnë pirgin brenda seksionit RAM, ose përndryshe ajo përcaktohet automatikisht gjatë procesit të përpilimit. Përmbajtja e SP referohet si Top of Stack (TOS). Kjo i tregon CPU-së se ku është ruajtur një e dhënë e re ose lexuar.

-Status Register (SR)-

Regjistri i statusit, i quajtur ndryshe Fjalori i Statusit të Procesorit (PSW), ose Regjistri i Flamurit përmban një grup bitësh tregues të quajtur flamuj, si dhe bit të tjerë që i përkasin ose kontrollojnë statusin e CPU-së. Një flamur është një bit i vetëm që tregon ndodhjen e një gjendjeje të veçantë. Numri i flamujve dhe kushteve të treguara nga një regjistër i statusit varet nga modeli i MCU.

Statusi i flamujve varet edhe nga madhësia e operandëve ALU. Mirëpo, le të identifikojmë së pari flamujt më të zakonshëm që mund të gjenden në pothuajse të gjitha MCU-të.

Këta janë:

- **Zero Flag (ZF)**
 - Quhet edhe biti zero. Vendoset kur rezultati i një operacioni ALU është zero.
- **Flag Carry (FC)**
 - Ky flamur vendoset kur një operacion aritmetik ALU prodhon një bartje.
- **Negative or Sign Flag (NF)**
 - Ky flamur vendoset nëse rezultati i një operacioni ALU është negativ.
- **Overflow Flag (VF)**
 - Ky flamur sinjalizon tejmbushje përveç mbledhjes ose zbritjes me numra të nënshkruar.
- **Interrupt Flag (IF)**
 - Ky flamur, i quajtur gjithashtu Aktivizimi i Përgjithshëm i Ndërprerjes (GIE), nuk është i lidhur me ALU. Ai tregon nëse një program mund të ndërpritet nga një ngjarje e jashtme (ndërprerje) apo jo.

Detyrë

Operacionet e mëposhtme janë disa shembuj që lidhen me veprimet e kryera nga ALU duke përdorur të dhëna 8-bitëshe.

Për secilin, përcaktoni Carry, Zero, Negative, and Overflow flags.

$\begin{array}{r} 01001010 + \\ 01111001 = \\ \hline 0\ 11000011 \\ \uparrow\ \uparrow \\ C\ N \end{array}$	$\begin{array}{r} 10110100 + \\ 01001100 = \\ \hline 1\ 00000000 \\ \uparrow\ \uparrow \\ C\ N \end{array}$	$\begin{array}{r} 10011010 + \\ 10111001 = \\ \hline 1\ 01010011 \\ \uparrow\ \uparrow \\ C\ N \end{array}$	$\begin{array}{r} 11001010 + \\ 00011011 = \\ \hline 0\ 11100101 \\ \uparrow\ \uparrow \\ C\ N \end{array}$
---	---	---	---

Zgjidhje:

Most significant bit në këtë grup është flamuri N. Biti në të majtë është C.

Flamuri zero vendoset nëse rezultati është 0, duke hequr bartjen, dhe flamuri i tejmbushjes vendoset nëse shtimi i numrave të së njëjtës shenjë (domethënë, me bit më të rëndësishëm të barabartë) jep një rezultat të shenjës së ndryshme (sinjalizuar nga N).

Me këtë informacion kemi atëherë:

Operation 4Ah + 79h = C3h : C = 0, N = 1, Z = 0 and V = 1.

Operation B4h + 4Ch = 100h : C = 1, N = 0, Z = 1 and V = 0.

Operation 9Ah + B9h = 153h : C = 1, N = 0, Z = 0 and V = 1.

Operation CAh + 1Bh = E5h : C = 0, N = 1, Z = 0 and V = 0.

/*

Note:

Overflow occurs when the value affects the sign:

- *overflow when adding two positives yields a negative*
- *or, adding two negatives gives a positive*
- *or, subtract a negative from a positive and get a negative*
- *or, subtract a positive from a negative and get a positive*

*/

MSP430 CPU Basic Hardware Structure

Familja MSP430 bazohet në një CPU 16-bit i cili u prezantua në modelet e hershme të serisë 3xx. Më vonë, arkitektura u shtri në 20-bit, CPUX, duke mbajtur pajtueshmërinë e plotë me CPU-në origjinale 16-bit.

Regjistrat MSP430: Ekzistojnë gjashtëmbëdhjetë regjistra 16-bitësh në CPU-në MSP430 me emrin R0, R1..., R15.

Regjistrat R4 deri R15 janë regjistra me qëllim të përgjithshëm.

Regjistrat për qëllime të specializuara janë:

- Regjistri i Counter Programit, me emrin R0 ose PC.
- Regjistri i Stack Pointer, me emrin R1 ose SP.
- Regjistri i Statusit, me një funksion të dyfishtë edhe si Gjenerator i Vazhdueshëm. Emërtohet R2, SR ose CG1.
- Gjenerator i vazhdueshëm, me emrin CG2.

Regjistri R3 përdoret ekskluzivisht si një gjenerator konstant me udhëzime dhe nuk përdoret për ruajtjen e të dhënave. Regjistrat PC dhe SP tregojnë gjithmonë një adresë të barabartë dhe kanë pak më pak domethënëse të lidhur me 0. Një karakteristikë e veçantë në familjen MSP430 është se këto dy regjistra mund të përdoren si operandë në udhëzime, duke lejuar programuesit të zhvillojnë aplikacione me algoritme më të thjeshtë të softuerit .

Regjistri i Statusit - Regjistri SR ka flamujt e përbashkët si: Carry (C), Zero (Z), Sign or Negative (N), Overflow (V) dhe General Interrupt Enable (GIE).

Njësia aritmetike-logjike - MSP430 CPU ALU ka një kapacitet operandi 16-bit; CPUX ka kapacitet operandi 16 ose 20-bit. Ai merret me veprimet aritmetike të mbledhjes me dhe pa mbartje, mbledhjes dhjetore me bartje, zbritjes me dhe pa bartje. Këto operacione ndikojnë në mbingarkesë, zero flag, negative flag. Operacionet logjike AND dhe XOR ndikojnë në flamuj, por ka dhe të tilla që nuk ndikojnë, si Bit Set (OR) ose Bit Clear.

Magjistralet e Sistemit

Memoria dhe pajisjet I/O arrihen nga CPU përmes magjistrave të sistemit. Një magjistrale është thjesht një grup linjash që kryejnë një funksion të ngjashëm.

Magjistrat e sistemit janë grupuar në tri klasë:

- Magjistralla e të dhënave
- Magjistralla e adresave
- Magjistralla e kontrollit

Magjistranja e të dhënave

Grupi i linjave që bartin të dhëna dhe udhëzime në ose nga CPU quhet magjistrane e të dhënave. Një operacion i leximit ndodh kur informacioni transferohet në CPU. Transferimi i një magjistranje të të dhënash nga CPU-ja në memorie ose në një pajisje periferike, quhet një veprim i shkrimit. Vini re se përcaktimi i një transferimi në magjistranet e të dhënave gjatë leximit ose shkrimit bëhet gjithmonë në lidhje me CPU-në. Kjo konventë vlen për çdo përbërës të sistemit. Linjat e magjistranës të të dhënave janë përgjithësisht dykahëse sepse i njëjti grup linjash na lejon të bartim informacione nga ose nga CPU-ja. Një transferim i informacionit referohet si transaksion i bus-it të të 29 dhënave. Numri i linjave në magjistranën e të dhënave përcakton gjerësinë maksimale të të dhënave që CPU mund të trajtojë në një transaksion të vetëm; Transferime më të gjera të të dhënave janë të mundshme, por kërkojnë transaksione të shumëfishta të magjistranëve të të dhënave. Për shembull, një magjistranë e të dhënave 8-bit mund të transferojë të paktën një bajt në një transaksion të vetëm, dhe një transaksion 16-bit do të kërkonte dy transaksione të magjistranës së të dhënave. Në mënyrë të ngjashme, një magjistranë 16 bitësh i të dhënave do të ishte në gjendje të transferonte më së shumti, dy bajte për transaksion; transferimi i më shumë se 16 bit do të kërkojë transaksione të shumëfishta.

Magjistranja e adresave

CPU bashkëvepron me vetëm një regjistrë memorie ose pajisje periferike njëkohësisht. Secili regjistrë, qoftë në memorie ose një pajisje periferike, identifikohet në mënyrë unike me një identifikues të quajtur adresë. Grupi i linjave që transportojnë këtë informacion të adresës formojnë magjistranën e adresës. Këto linja janë zakonisht njëdrejtimore dhe dalin nga CPU-ja. Adresat zakonisht emërtohen në shënime heksadecimale. Gjerësia e shiritit të adresës përcakton madhësinë e hapësirës më të madhe të memories që mund të adresojë CPU. Një magjistranë adresash me m bit do të jetë në gjendje të adresojë më së shumti 2 milion vende të ndryshme të memories. Për shembull, me një magjistranë adresash 16-bitësh, CPU-ja mund të ketë deri në $= 64K$ vendndodhje të emërtuara $0x0000, 0x0001, \dots, 0xFFFF$. Vini re se pjesët e linjave të magjistranës të adresave punojnë si një grup, i quajtur fjalë adrese dhe nuk konsiderohen individualisht kuptimplotë.

Shembull. Përcaktoni se sa vendndodhje të ndryshme të kujtesës mund të aksesohen dhe diapazoni i adresave (d.m.th., adresat fillestare dhe përfundimtare) në shënimet magji me një magjistranë adresash prej (a) 12 bit, dhe (b) 22 bit. Arsyetoni përgjigjen tuaj.

Zgjidhja: (a) Për 12 bit, ka $2^{12} = 2^2 \times 2^{10} = 4 K$ vendndodhje të ndryshme që mund të adresohen. Në terma binarë, ne i mendojmë si paraqitje të plota të $0000\ 0000\ 0000B$ deri $1111\ 1111\ 1111B$, të cilat në shënimet magjike bëhen $0x0000$ deri $0xFFFF$.

(b) Duke punuar në mënyrë të ngjashme, për 22 bit ka $2^{22} = 2^2 \times 2^{20} = 4M$ vendndodhje. Megjithatë, meqenëse ka 22 bit dhe integruetit hex përfaqësojnë vetëm katër bit, numri më i madh me dy bitët më domethënës është 3, kështu që diapazoni i adresës është $0x000000$ deri $0x3FFFFFF$.

Shembull: Një sistem i caktuar ka një madhësi memorie prej 32 fjalësh memorie. Cili është numri minimal i linjave të kërkuara për magjistralen e adresës? Në fuqitë prej 2, $32 K = 25 \times 2^{10} = 215$. Prandaj, magjistralla e adresës duhet të përmbajë të paktën 15 rreshta, një për bit. (Përgjigja mund të ishte shprehur edhe si $n = \log_2 (32 * 1024) = 5 + 10 = 15$).

Magjistralla e kontrollit

Magjistralla e kontrollit grupon të gjitha linjat që mbajnë sinjalet që rregullojnë aktivitetin e sistemit. Ndryshe nga linjat e magjistraleve të adresave dhe të dhënave të cilat zakonisht interpretohen si një grup (adresë ose të dhëna), sinjalet e kontrollit zakonisht funksionojnë dhe interpretohen veçmas. Sinjalet e kontrollit përfshijnë ato që përdoren për të treguar nëse CPU po kryen një qasje leximi ose shkrimi, ato që sinkronizojnë transferimet duke thënë se kur fillon dhe mbaron transaksioni, ata që kërkojnë shërbime në CPU dhe detyra të tjera. Shumica e linjave të kontrollit janë njëdrejtimëshe dhe hyjnë ose largohen nga CPU, në varësi të funksionit të tyre. Numri dhe funksioni i linjave në një magjistrallë kontrolli do të ndryshojnë në varësi të arkitekturës dhe aftësive të CPU-së.

Krahasimi mes Mikrokontrollerëve dhe procesorëve

Përderisa, mikroprocesori apo CPU (Central Processing Unit) është një çip i vetëm, mikrokontrolleri përmban në një **qark të integruar**, mikroprocesorin si dhe qarqet tjera të nevojshme që përbëjnë një sistem komplet mikrokompjuterik. **Qarku i integruar** (integrated circuit) në të cilin është i vendosur mikrokontrolleri përmban:

- CPU
- memorie RAM
- memoria ROM
- interfejs serik (serial interface)
- interfejs paralel (parallel interface)
- timers
- qarku për kontroll të ndërprerjeve të ciklit të programit apo interapteve (interrupt control circuit).

Mikroprocesorët janë të dizajnuar që të bëjnë procesimin e të dhënave në sistemet kompjuterike, kurse mikrokontrollerët janë të dizajnuar për të bërë kontrollin e pajisjeve hyrësedalëse (input-output devices) që janë të kyçura në ato pajisje.

Mikroprocesorët me qëllim të përgjithshëm **nuk kanë RAM, nuk kanë ROM, nuk kanë porte H/D (I/O)**

Mikrokontrolleri është kompjuter në kuadër të një qarku të integruar, i përgaditur me teknologji të shkallës së lartë të integritimit (VLSI) dhe që shfrytëzohet për kontrollin e pajisjeve të ndryshme elektronike.

Mikrokontrollerët zakonisht nuk kanë nevojë për hardware shtesë (përkundër mikroprocesorëve) dhe shfrytëzohen për kontrollim të aplikacioneve.

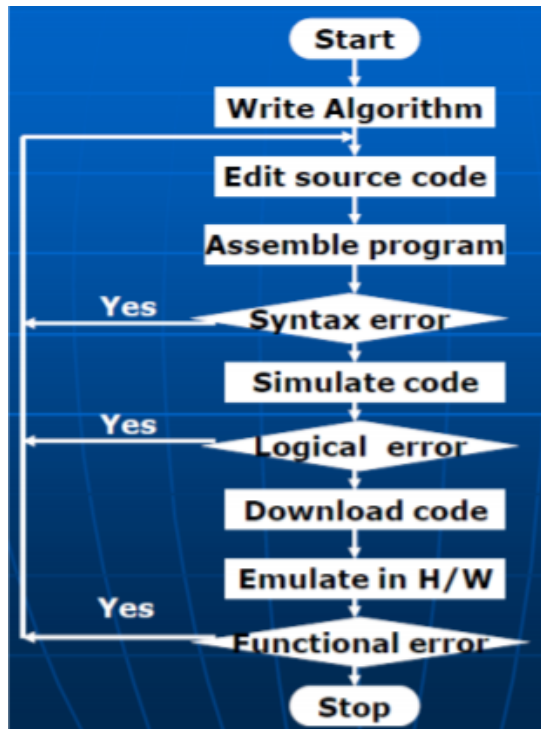
Mikrokontrolleri është në gjendje të kryejë funksionet themelore të një kompjuteri.

Shembull: Mund të shfrytëzohet indikator i 7-segmentësh (në vend të monitorit) dhe numërori (në vend të tastierës) me një mikrokontroller për procesim të zakonshëm me fjalë 8-bitëshe.

Mikrokontrolleri i zbrazët nuk vlen për asgjë, andaj duhet të:

- Shkruhet programi (kodi) në kompjuter
- Inçizohet në softuerin përkatës në kompjuter
- Testohet programi përmes simulatorit
- Programohet mikrokontrolleri real
- Testohet mikrokontrolleri në qark real.

Fazat e programimit:



Pllakat për programim të mikrokontrollerëve

- Pasi programi të jetë simuluar dhe konvertuar në assembler, kodi heksadecimal shkarkohet në qarkun e integruar (mikrokontroller). Kjo realizohet përmes portit serik (ose paralel) që ndërlidh kompjuterin me pllakën përkatëse ku ndodhet mikrokontrolleri.

Selektimi i mikrokontrollerëve në dispozicion:

- Atmel
- ARM
- Intel
 - 8051 (8-bit)
 - 8096 (16-bits)
 - 80960 (32-bits)
- Microchip
 - PIC 16Fxxx (8-bits)
- Texas Instruments
 - TMS370 (16/32 bit)
 - MSP430 (16 bit)
- Motorola
 - 68HC11 (8-bits)
 - 68HC16 (16-bits)

- MPC500 (32-bits)
- Zilog
 - Z8
 - Z86E02

VON-NEUMANN AND HARVARD ARCHITECTURES

Këto dy arkitektura të mikrokontrollerëve mund të klasifikohen nga mënyra se si ata përdorin memorien.

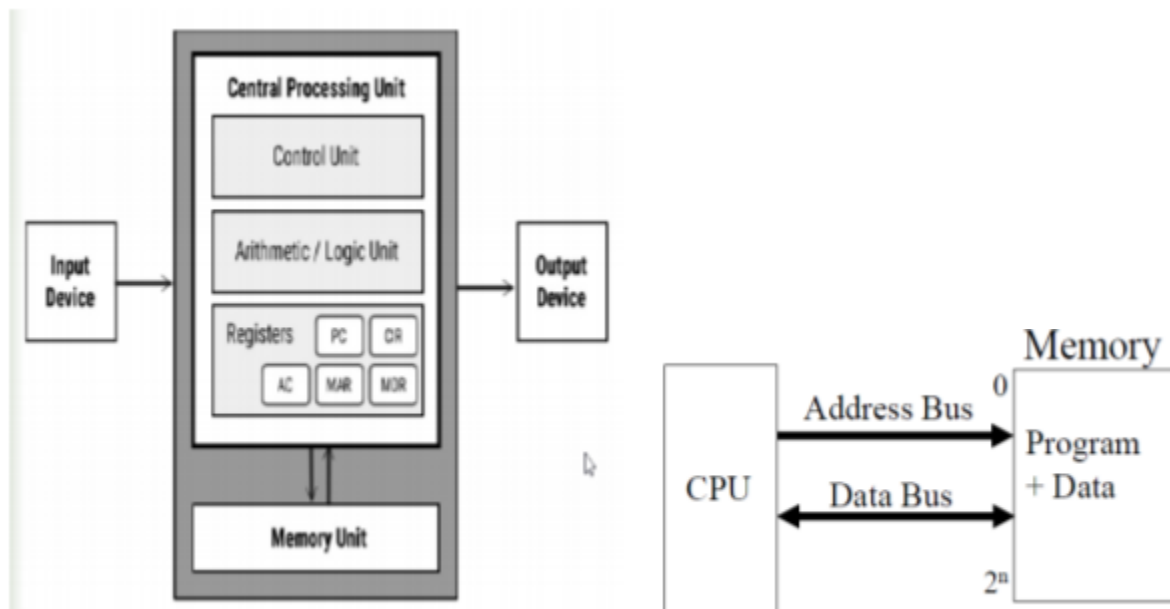
-Arkitektura Von-Neumann

Në një arkitekturë Von-Neumann, e njëjta memorie dhe i njëjti bus përdoren për të ruajtur të dhënat dhe udhëzimet që ekzekuton programi (data memory and program memory). Meqenëse nuk mund të përdorni njëkohësisht memorien e programit dhe memorien e të dhënave, arkitektura Von Neumann është e ndjeshme ndaj ngushticave dhe ndikon në performancën e sistemit.

Arkitektura Von-Neumann u publikua nga John Von Neumann në vitin 1945.

Arkitektura e kompjuterit të tij përbëhet nga:

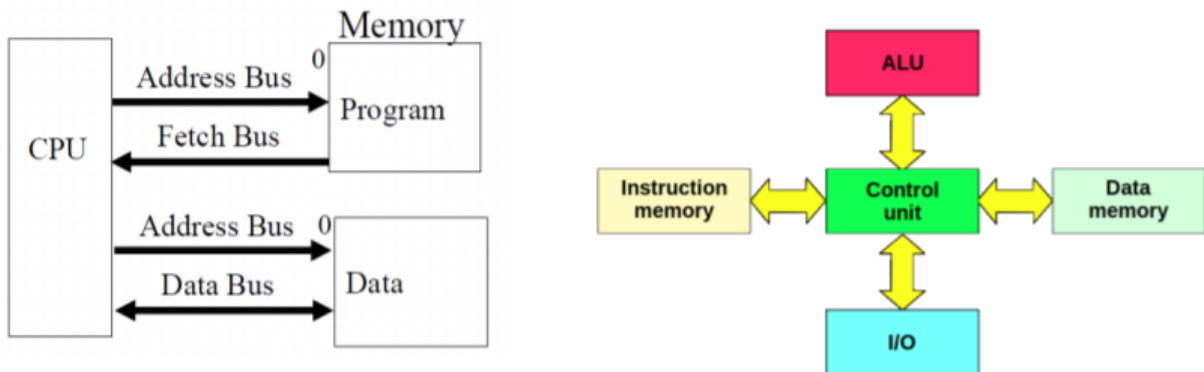
- Control Unit
- Arithmetic and Logic Unit (ALU)
- Memory
- Registers
- Input/Output



Arkitektura Harvard

Arkitektura e Harvard-it ruan udhëzimet dhe të dhënat e makinës në njësi të veçanta të memories që janë të lidhura nga busë të ndryshëm. Në këtë rast, ka të paktën dy hapësira të adresave të memories për të punuar, kështu që ekziston një regjistër memorie për udhëzimet e makinës dhe një tjetër regjistër memorie për të dhënat. Kompjuterët e dizajnuar me arkitekturën e Harvardit janë në gjendje që njëkohësisht të ekzekutojnë një program dhe të kenë qasje në të dhëna në mënyrë të pavarur.

Arkitektura e Harvardit ka një ndarje të rreptë midis të dhënave dhe kodit. Kështu, arkitektura e Harvardit është më e komplikuar, por tubacionet e ndara heqin ngushticën që krijon arkitektura Von Neumann.



Mikrokontrollerët e ndryshëm ofrojnë kombinime të performancave të ndryshme. Andaj, për zgjedhjen e mikrokontrollerit të përshtatshëm për aplikacione të caktuara duhet të kemi parasysh karakteristikat vijuese:

- Numri i pinave për njësitë H/D
- Tipi i timers
- Resurset për ndërprerje (interrupts)
- Hyrjet analoge
- Shpejtësia max. e clock-ut
- Kapaciteti i memories programuese
- Madhësia e magjistrale së të dhënave
- Komunikimi serik
- Simulimi
- Programimi
- Çmimi, etj.

Mikrokontrollerët 8051

Mikrokontrolleri 8051 është nënversion i 8052.

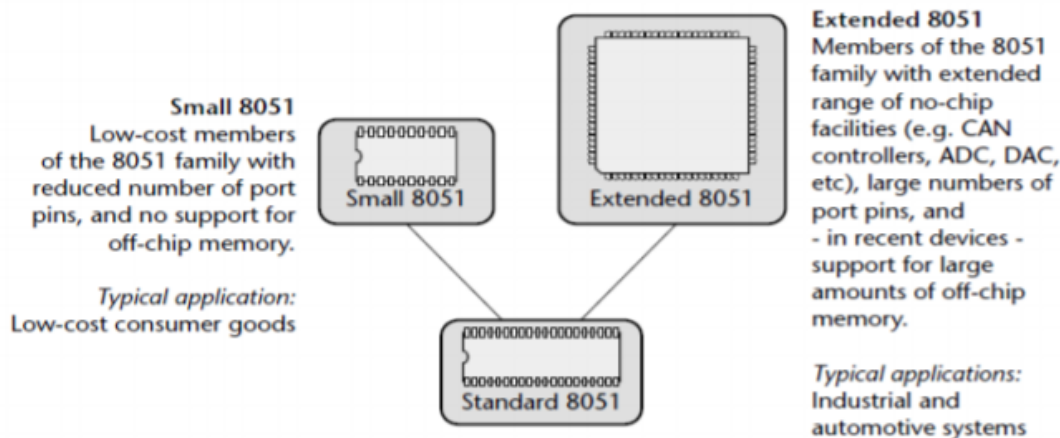
Mikrokontrolleri 8031 është pa ROM, mund t'i shtohet ROM external, por humben dy pina dhe ngelin vetëm 2 porte për operime H/D.

Krahasimi i vetive të tre mikrokontrollerëve të familjes 8051:

Feature	8051	8052	8031
ROM (on-chip program space in bytes)	4K	8K	0K
RAM (bytes)	128	256	128
Timers	2	3	2
I/O pins	32	32	32
Serial port	1	1	1
Interrupt sources	6	8	6

Në kuadër të familjes 8051, ekzistojnë tri versione të ndryshme të prodhimit:

- Standard 8051
- versioni i vogël (Small) 8031
- versioni i zgjeruar (Extended) 8051

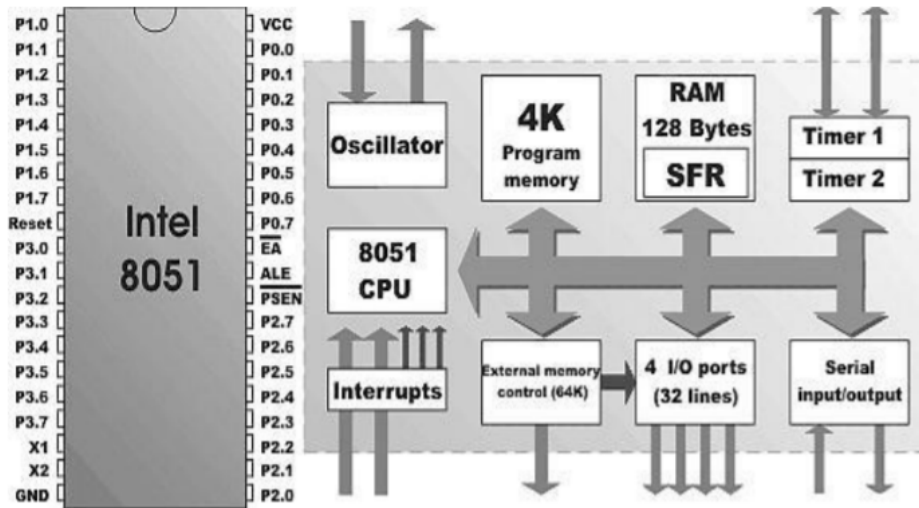


Mikrokontrolleri 8051 është dizajnuar nga Intel në 1981. Karakteristikat e mikrokontrollerit Intel 8051 janë:

- 8 bitësh
- 4kB ROM
- 128B RAM
- 2 timera 16 bitësh
- 4 porte paralele 8 bitëshe
- Interfejsë serik
- 64k hapësirë memorike për kod të jashtëm

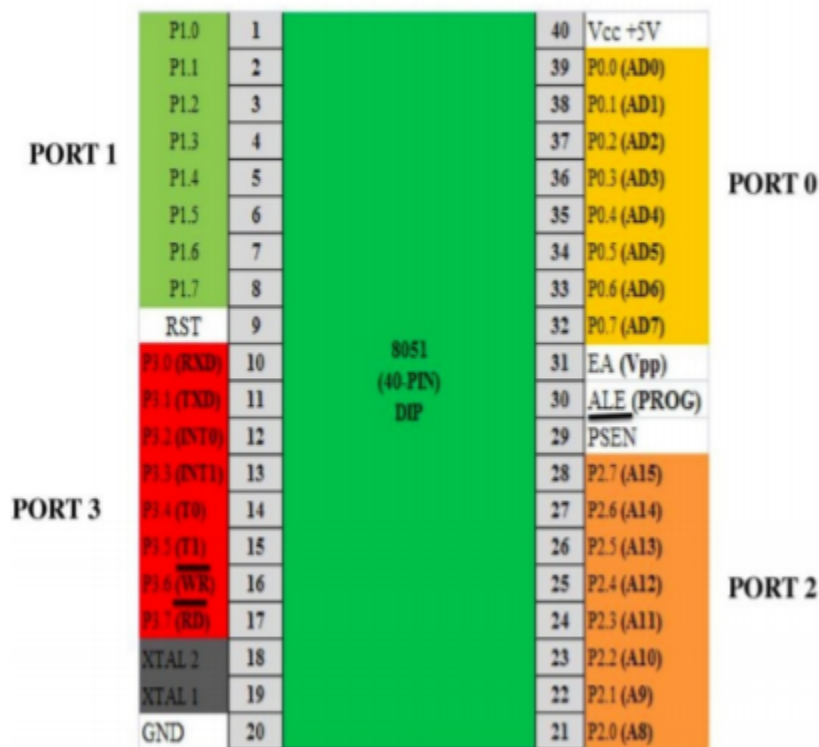
- 64k hapësirë memorike për të dhëna të jashtme
- procesor Boolean (Boolean processor), operon me bita individual
- 210-bit lokacione që mund të adresohen(addressable locations)

Arkitektura e mikrokontrollerit 8051.



Pinat e mikrokontrollerit 8051:

8051 Pin - out



- **Pinat 1 deri në 8** njihen si Port 1.
- **Pini 9** –Ky është një pin RESET, i cili përdoret për të rivendosur mikrokontrollerin në vlerat e tij fillestare.
- **Pinat 10 deri në 17** –njihen si Porti 3, dhe shërben për disa funksione si:
 - ndërprerjet,
 - hyrja me timer,
 - sinjalet e kontrollit,
 - sinjalet e komunikimit serial RxD dhe TxD, etj.
- **Pinat 18 & 19** –Këta pinë janë përdorur për të ndërfaqësuar një kristal të jashtëm për të marrë orën e sistemit (system clock).
- **Pini 20** –Ky pin siguron furnizimin me energji elektrike në qark.
- **Pinat 21 deri në 28** –Këta pinë janë të njohur si Port 2.
 - Ky port shërben si port I/O.
 - Sinjalet e bus-ve të rendit të lartë gjithashtu janë multipleksuar duke përdorur këtë port.
- **Pini 29** –Ky është pin PSEN i cili qëndron për Program Store Enable. Përdoret për të lexuar një sinjal nga memoria e programit të jashtëm.
- **Pini 30** –Ky është pin EA (Enable Access) të jashtme. Përdoret për të aktivizuar/çaktivizuar lidhjen e memories së jashtme.
- **Pini 31** –Ky është pin ALE (Address Latch Enable). Përdoret për të demultipleksuar sinjalin e adresës – së të dhënave të portit.
- **Pinat 32 deri në 39** –Këta pina njihen si Port 0. Shërben si port I/O. Adresa e renditjes së ulët dhe sinjalet e bus-ve të të dhënave multipleksohen duke përdorur këtë port.
- **Pini 40** –Ky pin përdoret për të siguruar furnizimin me energji elektrike në qark.

Konvertimi nga binary në hexadecimal:

Decimal	Binär	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Shembull:

$\underbrace{1000}_8 \underbrace{1110}_E$ (binar)
(hex)

Mënyra të shënimit: **0x8E** ose **8Eh** ose **8E16**

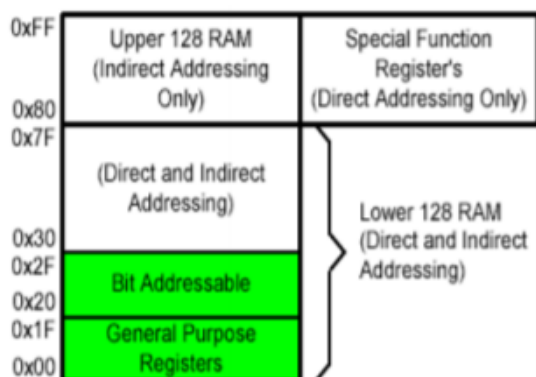
Memoria e miktrokontrollerit 8051 është e ndarë në:

- Memorie të programit (Program Memory)
- Memorie të të dhënave (DataMemory)
 - Memoria e programit (ROM) përdoret për ruajtje të përhershme të programit që ekzekutohet.
 - Memoria e të dhënave (RAM) përdoret për ruajtje të përkohshme të rezultateve dhe variablave.

Internal RAM

Në varësi të mikrokontrollerit 8051 janë në dispozicion deri në 256 bajt të memories për të dhëna të brendshme.

Lokacionet në dispozicion të përdoruesit janë nga 0 deri në 7F, p.sh 128 regjistrat e parë dhe kjo pjesë e RAM është e ndarë në disa blloqe. 128 bajtët e parë të memories për të dhëna të brendshme mund të adresohen direkt dhe indirekt. 128 bajtët e sipërm (0x80–0xFF) mund të adresohen vetëm indirekt.

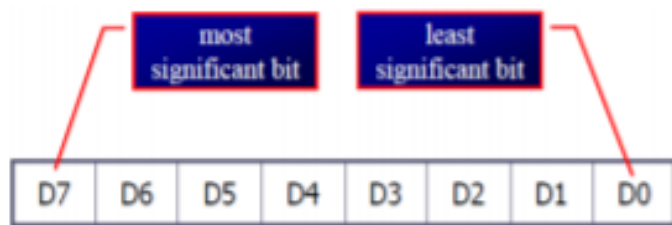


Regjistrat

CPU përdor regjistrat për të ruajtur informacionin përkohësisht si: *të dhënat që do të pëdoren dhe adresat e të dhënave që do të merren nga memoria.*

Regjistri është një vend me kapacitet të vogël që shërben për ruajtje të të dhënave që janë pjesë e procesorit kompjuterik. Regjistri mund të mbajë një instruksion, një adresë të dhënash ose çdo lloj të dhëne si: p.sh. një sekuençë të bitëve ose karakter individual. Shumica dërmuese e regjistratit të 8051 janë regjistrat 8-bit.

8 bitët e një regjistri tregohen nga MSB D7 tek LSB D0. Çdo e dhënë më e madhe se 8 bit duhet të ndahet në copa 8-bit për para se të përpunohet.



Regjistrat e funksioneve special janë:

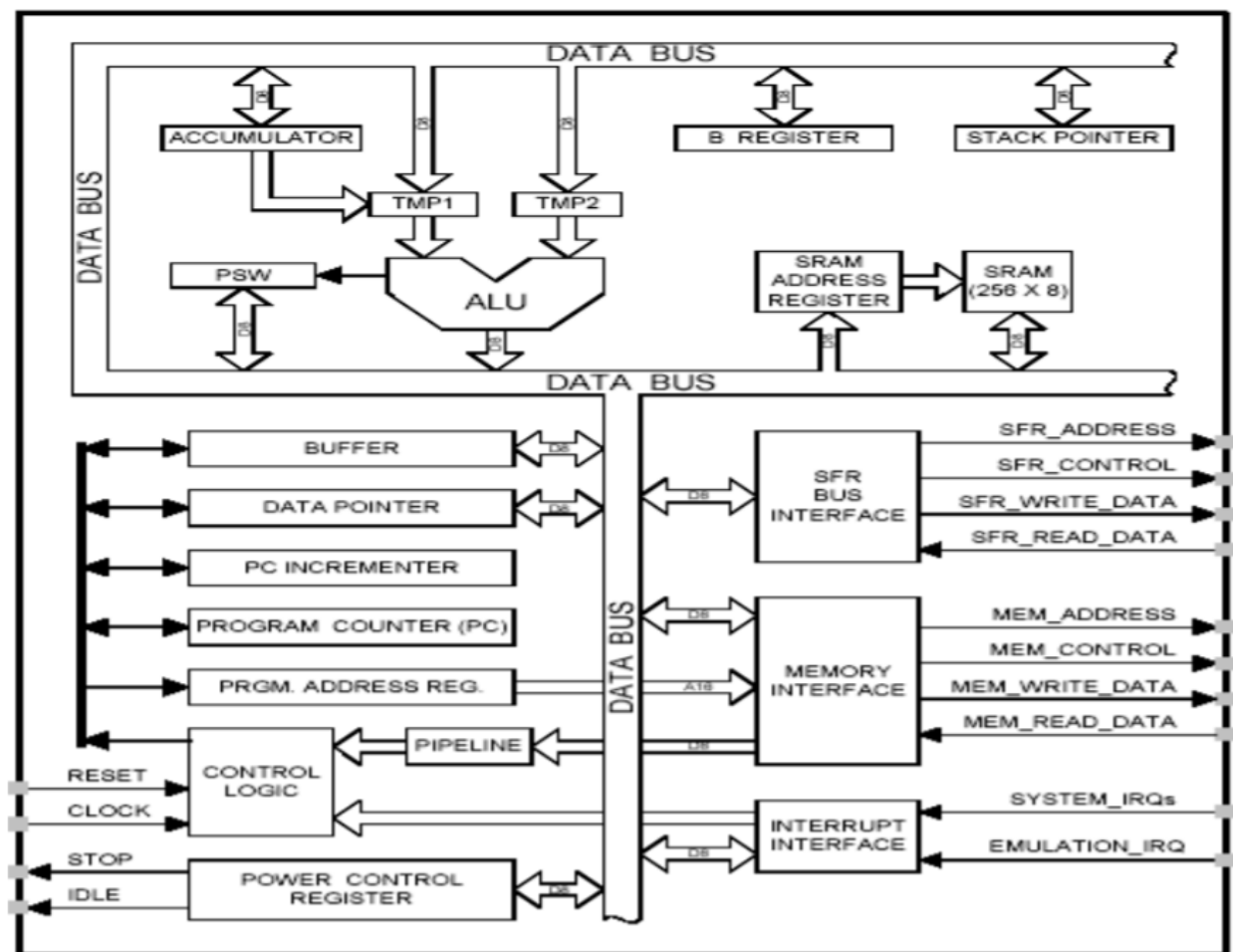
- DATA regjistrat
- CONTROL regjistrat

Këtu mund të hynë: Timerat, Portet serike, Sistemi Interrupt, Konvertori Analog në Digjital, Konvertori Digjital në Analog, etj.

Siç dihet, një element bistabil me dy gjendjet e veta 0 ose 1 mund të paraqesë një bit të shënimit binar. Mirëpo, shënimet binare, zakonisht përbëhen nga më shumë bita, të cilët së bashku e formojnë, një apo më shumë fjalë binare apo bajt (grupi prej 8 bitëve). Pasi që, në rastin tonë (pasi që CPU-ja është 8-bitësh), kemi të bëjmë, me shënime, tetë-bitëshe, rrjedh se, për t'i ruajtur këto shënime, na nevojiten më shumë flip-flop, të organizuar në atë mënyrë që t'i përgjigjen kësaj kërkesë. Si përgjigje për këtë sfidë është krijuar një qark logjik sekuencial me emërtimin regjistër.

Regjistri, është një grup i bistabilëve (flip-flop) të vendosur paralel, të cilët përdoren për ruajtje të përkohshme të shënimeve binare (bitëve). Flip-flop-ët në regjistra, janë të pavarur mes veti dhe janë të renditur në atë mënyrë që, secili flip-flop, ta ruaj një bit të caktuar me peshë të caktuar binare, sipas pozitës së flip-flop-ëve. Nga kjo rrjedh se, regjistrat janë elemente përbërëse të kujtesës (memory elements) dhe përdoren si lokacione të veçanta të kujtesës (memories) në vet CPU-në.

Bllok skema e regjistrave:



Dallojmë dy lloje të regjistrave:

- regjistrat punues për aplikim të gjerë
- regjistrat për dedikime të veçanta apo të caktuara më parë, SFR (Special Function Register).

Regjistri A apo Akumulatori. Akumulatori, është i lidhur me ALU-në, kështu që, praktikisht ai merr pjesë në të gjitha operacionet aritmetiko-logjike dhe pas ekzekutimit të operacionit të caktuar, e ruan rezultatin. Pra, gjatë operacioneve aritmetiko-logjike, një operand gjendet në një lokacion të memories kurse tjetri gjendet në A.

Operanti, paraqet një shënim me të cilin kryhet ndonjë veprim aritmetik apo logjik në mikroprocesorin e mikrokontrollerit 8051. Akumulatori (A), është i lidhur me memorien e jashtme përmes magjistrales së të dhënave.

Regjistri B është regjistër 8-bitësh, i cili përdoret gjatë ekzekutimit të operacioneve të shumëzimit dhe pjesëtimit, me ç'rast në këtë regjistër gjendet njëri nga operandët.

Regjistri SP apo Stack Pointer-i, e ka mundësinë e ruajtjes së më shumë adresave dhe lokacioneve të memories. Regjistri SP është një regjistër 8-bitësh dhe përmban adresën e shënimit apo përmbajtjes së një lokacioni të memories, që është në majë të stekut (stack). Pas resetimit, SP-ja inicializohet në adresën 07H dhe steku fillon në adresën 08H. Pas kryerjes së kërcimeve nga programi kryesor nëpër nënprograme duhet të kthehemi në adresën e saktë. Këtë detyrë e ka regjistri SP apo Stack Pointer-i. Kthimi në adresën e saktë të programit kryesor kryhet sipas principit LIFO (Last In First Out).

DPTR (Data Pointer Register) apo regjistri i treguesit të të dhënave, mund të përdoret si një regjistër 16-bitësh apo si dy regjistra 8-bitësh. Kur DPTR përdoret **si regjistër 16-bitësh**, për mbulimin e memories së jashtme, përdoren instruksione speciale për t'i mbushur apo inkrementuar të tërë 16-bitët. Kur përdoret **si regjistër 8-bitësh**, atëherë me DPTR manipulohet si me dy regjistra të ndarë, DPH (Data Pointer High) dhe DPL (Data Pointer Low).

Regjistri PSW (Program Status Word), në mikrokontrollerin Intel 8051 është si status regjistër. Ky regjistër është 8-bitësh dhe për dallim nga regjistrat e përmendur më parë, te këta regjistra, bëhet vendosja e komplet shënimit, d.m.th bëhet deklarimi i plotë i bajtit.

Regjistri i urdhrave (Instruction Register IR) – Pasi që, programi është i vendosur në memorien e jashtme ROM, është e nevojshme që, çdo urdhër (instruction), të sillet në CPU, para se të ekzekutohet programi.

Urdhrat në CPU, vendosen në regjistrin e urdhrave apo **IR (Instruction Register)**, në mënyrë që pastaj, të dekodohen dhe të iniciohen qarqet për ekzekutim.

Dekoderi i urdherave, ka për detyrë dekodimin e urdherave dhe përcaktimin e operacionit të cilin do ta kryej mikroprocesori i 8051-shit.

Numëruesi i programit apo PC (Program Counter), bën udhëheqjen e CPU-së nëpër program. PC-ja ka për detyrë, që të mundësojë nxjerrjen e urdherave nga memoria ashtu që mikroprocesori të mund t'i kryej ato urdhra hap pas hapi, sipas renditjes së tyre në program. Detyra e PC-së

është që të japë adresën e memories në të cilën ruhet apo gjindet urdhëri që është në radhë për t'u ekzekutuar.

Njësia drejtuese dhe kohore (clock generator) –kjo njësi së bashku me dekoderin e urdhërave formon, tërësinë e udhëheqjes dhe të drejtimit të mikroprocesorit të 8051-shit. Gjeneratori i impulseve kohore është burim i impulseve për tërë mikrokontrollerin. Njësia drejtuese mundëson paraqitjen e dukurive hap pas hapi në mikroprocesor, gjegjësisht në mikrokontroller.

Oscilatori

Oscilatori tek mikrokontrolleri Intel 8051 lidhet në pinat 18 dhe 19. Egzistojnë dy mënyra të realizimit të oscilatorit tek Intel 8051. Mënyra e parë është realizimi përmes një kristali të kuarcit dhe dy kondensatorë. Kurse mënyra e dytë është që të sjellim takt sinjalin nga një burim i jashtëm (p.sh nga një oscilator TTL 7404).

Pas resetimit të mikrokontrollerit Intel 8051 do të kemi këtë gjendje:

- PC do ta ketë vlerën 0000h
- A, B, PSW do ta kenë vlerën 0000h
- SP do ta ketë vlerën 00007h
- DPTR do të kalon në gjendje fillestare 0000h
- Portet do të jenë në FFh
- Regjistrat e tajmerëve, SCON dhe SBUF do ta kenë vlerën 0000h.
- Dy variantat të qarkut për resetim (RESET)

Interrupts

Interrupt është sinjali që i dërgohet mikrokontrolluesit për të shënuar ngjarjen që kërkon vëmendje të menjëhershme.

Ky sinjal kërkon që mikrokontrolluesi të ndalojë për të kryer programin aktual përkohësisht kohën për të ekzekutuar një kod të veçantë. Do të thotë kur pajisja e jashtme përfundon detyrën e vendosur mbi të, mikrokontrolluesi do të njoftohet se mund të ketë qasje dhe të marrë informacionin dhe ta përdorë atë. Ndërprerjet janë njësoj si të presësh që të bie telefoni. Me fjalë tjera interapti apo ndërprerja, është një ngjarje që paraqitet si shkaktar i një kushti i cili bën ndërprerjen e përkohshme të rrjedhës normale të programit kryesor, përderisa ky kusht shërbehet (serviced) nga një program tjetër. Me përfundimin e ngjarjes që ka shkatuar interaptin apo ndërprerjen e përkohshme të rrjedhës normale të programit kryesor, do të vazhdojë puna normale e programit kryesor në hapin aty ku kishte mbetur para ndërprerjes nga interapti.

Kur paraqitet një Interrupt, do të kemi këto veprime siq vijon:

1. Ndërpritet ekzekutimi i programit kryesor dhe degëzohet në ISR (Interrupt Service Routine) apo Interrupt Handler.
2. ISR-ja bën ekzekutimin e programit të thirrur nga Interapti
3. Kryhet operacioni i dëshiruar
4. Ndalet me instruksionin RETI apo “return from interrupt ” (kthehu/dalje nga interapti)

5. Vazhdon punën normale programi kryesorë (aty ku kishte mbetur para paraqitjes së Interaptit)

Mikrokontrolluesi 8051 mund të konfigurohet në mënyrë të tillë që të përfundojë përkohësisht programin bazë në shfaqjen e ndërprerjeve. Kur kompletohet një nën rutinë, atëherë fillon ekzekutimi i programit kryesor.

Në përgjithësi, janë pesë burime të ndërprerjeve në mikrokontrolluesin 8051.

Këto ndërprerje vektoriale janë të paraqitura më poshtë:

1. TF0 (Timer 0 Overflow Interrupt)
2. TF1 (Timer 1 Overflow Interrupt)
3. INT0 (External Hardware Interrupt)
4. INT1 (External Hardware Interrupt)
5. RI/TI (Serial Communication Interrupt)

Nga këto (INT0) dhe (INT1) ekziston një ndërprerje e jashtme, e cila mund të jetë një avantazh negativ i shkaktuar ose i nivelit të ulët të shkaktuar. Kur të aktivizohen të gjitha këto ndërprerje, vendosni flamujt përkatës, përveç ndërprerjes serike. Në fakt, flamujt e ndërprerjes pastrohen kur procesori degëzohet në rutinën e shërbimit të ndërprerjes. Nga ana tjetër, flamujt e jashtëm të ndërprerjes pastrohen kur procesori degëzohet në rutinën e shërbimit të ndërprerjes, me kusht që ndërprerja të shkaktohet një avantazh negativ. Në të kundërt, kohëmatësit dhe porta serike ndërpresin dy prej tyre janë ndërprerje të jashtme; dy prej tyre janë ndërprerje të kohëmatësit 19 dhe një terminal i ndërprerjes së portës serike.

Modet e adresimit (Addressing Modes)

Me fjalën adresim nënkuptojmë përcaktimin e lokacionit të shënimit në memorie (data location in memory) ose në regjistrat punues të mikroprocesorit gjegjësisht të mikrokontrollerit Intel 8051. Mënyra se si përcaktohet në mnemonikë burimi i shënimit dhe adresa e destinimit që e vë në lëvizje shënimin e definon mënyrën apo modin e adresimit. Mikrokontrolleri Intel 8051 përkrahë tetë mode të adresimit prej të cilave, katër të parat janë më të rëndësishme pasi që përdoren më shumë në zbatimet praktike.

Metodat e adresimit janë:

1. Adresimi përmes regjistrave apo Register addressing
2. Adresimi në mënyrë direkte apo Direct addressing
3. Adresimi në mënyrë indirekte apo Indirect addressing
4. Adresimi aty për aty apo Immediate addressing
5. Adresimi nëpërmes të indekseve/i indeksuar apo Indexed addressing

/* Sqarim shitesë

Like any other microprocessor, the 8051 has a flag register to indicate arithmetic conditions such as the carry bit. The flag register in the 8051 is called the program status word (PSW) register. In this section we discuss various bits of this register and provide some examples of how it is altered.

PSW (program status word) register The program status word (PSW) register is an 8-bit register. It is also referred to as the flag register. Although the PSW register is 8 bits wide, only 6 bits of it are used by the 8051. The two unused bits are user-definable flags. Four of the flags are called conditional flags, meaning that they indicate some conditions that result after an instruction is executed. These four are CY (carry), AC (auxiliary carry), P (parity), and OV (overflow).

The bits PSW.3 and PSW.4 are designated as RSO and RSI, respectively, and are used to change the bank registers. They are explained in the next section. The PSW.5 and PSW.1 bits are general-purpose status flag bits and can be used by the programmer for any purpose. In other words, they are user-definable.

CY	AC	F0	RS1	RS0	OV	—	P
----	----	----	-----	-----	----	---	---

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1.
RS0	PSW.3	Register Bank selector bit 0.
OV	PSW.2	Overflow flag.
—	PSW.1	User-definable bit.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of 1 bits in the accumulator.

RS1	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

The following is a brief explanation of four of the flag bits of the PSW register. The impact of instructions on these registers is then discussed.

- **CY, the carry flag**

This flag is set whenever there is a carry out from the D7 bit. This flag bit is affected after an 8-bit addition or subtraction. It can also be set to 1 or 0 directly by an instruction such as “SETB C” and “CLR C” where “SETB C” stands for “set bit carry” and “CLR C” stands for “clear carry”.

- **AC, the auxiliary carry flag**

If there is a carry from D3 to D4 during an ADD or SUB operation, this bit is set; otherwise, it is cleared. This flag is used by instructions that perform BCD (binary coded decimal) arithmetic. See Chapter 6 for more information.

- **P, the parity flag**

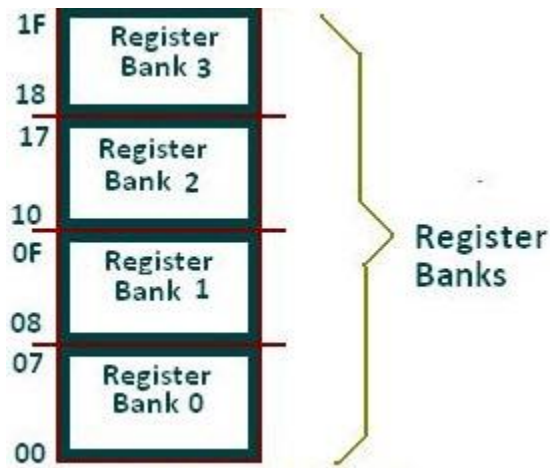
The parity flag reflects the number of 1 s in the A (accumulator) register only. If the A register contains an odd number of 1s, then $P = 1$. Therefore, $P = 0$ if A has an even number of 1s.

- **OV, the overflow flag**

This flag is set whenever the result of a signed number operation is too large, causing the high-order bit to overflow into the sign bit. In general, the carry flag is used to detect errors in unsigned arithmetic operations. The overflow flag is only used to detect errors in signed arithmetic operations

The collection of general purpose registers (R0-R7) is called register banks, which accept one byte of data. The bank register is a part of the **RAM memory in the embedded** microcontrollers, and it is used to store the program instructions. Each microcontroller consists of various memory banks, and each bank register consists of a unique address for recognizing the storage location.

Register Banks in 8051



The 8051 microcontroller consists of four register banks, such as Bank0, Bank1, Bank2, Bank3 which are selected by the PSW (Program Status Word) register. These register banks are present in the internal RAM memory of the 8051 microcontroller, and are used to process the data when the microcontroller is programmed.

Switching of Register Banks

By default, the 8051 microcontroller is powered up with register bank 0; and, by using the Program Status Word (PSW), we can switch to other banks. The two bits of PSW are used for switching between the register banks. These two bits are accessed by the bit-addressable instructions SETB and CLR.

Based on the possible combinations of RS1 and RS0 of PSW, the register bank is changed accordingly, i.e., if RS1 and RS0 are 0, then Bank 0 is

selected. Similarly, Bank1, 2&3 are selected as per the values of RS1 and RS0.

*/

Addressing Modes:

In this section, we will see different addressing modes of the 8051 microcontrollers. In 8051 there are 1-byte, 2-byte instructions and very few 3-byte instructions are present. The opcodes are 8-bit long. As the opcodes are 8-bit data, there are 256 possibilities. Among 256, 255 opcodes are implemented.

The clock frequency is 12MHz, so 64 instruction types are executed in just 1 μ s, and the rest are just 2 μ s. The Multiplication and Division operations take 4 μ s to execute.

In 8051 There are six types of addressing modes.

- Immediate AddressingMode
- Register AddressingMode
- Direct AddressingMode
- Register IndirectAddressing Mode
- Indexed AddressingMode
- Implied AddressingMode

Immediate addressing mode

In this Immediate Addressing Mode, the data is provided in the instruction itself. The data is provided immediately after the opcode. These are some examples of Immediate Addressing Mode.

```
MOVA, #0AFH;
```

```
MOVR3, #45H;
```

```
MOVDPTR, #FE00H;
```

In these instructions, the # symbol is used for immediate data. In the last instruction, there is DPTR. The DPTR stands for Data Pointer. Using this, it points to the external data memory location. In the first instruction, the immediate data is AFH, but one 0 is added at the beginning. So when the data is starting with A to F, the data should be preceded by 0.

Register addressing mode

In the register addressing mode the source or destination data should be present in a register (R0 to R7). These are some examples of RegisterAddressing Mode.

```
MOVA, R5;
```

```
MOVR2, #45H;
```

```
MOVR0, A;
```

In 8051, there is no instruction like **MOVR5, R7**. But we can get the same result by using this instruction **MOV R5, 07H**, or by using **MOV 05H, R7**. But these two instructions will work when the selected register bank is **RB0**. To use another register

bank and to get the same effect, we have to add the starting address of that register bank with the register number. For example, if the RB2 is selected, and we want to access R5, then the address will be (10H + 05H = 15H), so the instruction will look like this **MOV 15H, R7**. Here 10H is the starting address of Register Bank 2.

Direct Addressing Mode

In the Direct Addressing Mode, the source or destination address is specified by using 8-bit data in the instruction. Only the internal data memory can be used in this mode. Here are some of the examples of direct Addressing Mode.

```
MOV80H, R6;
```

```
MOVR2, 45H;
```

```
MOVR0, 05H;
```

The first instruction will send the content of register R6 to port P0 (Address of Port 0 is 80H). The second one is forgetting content from 45H to R2. The third one is used to get data from Register R5 (When register bank RB0 is selected) to register R5.

Register indirect addressing Mode

In this mode, the source or destination address is given in the register. By using register indirect addressing mode, the internal or external addresses can be accessed. The R0 and R1 are used for 8-bit addresses, and DPTR is used for 16-bit addresses, no other registers can be used for addressing purposes. Let us see some examples of this mode.

```
MOV0E5H, @R0;
```

```
MOV@R1, 80H
```

In the instructions, the @ symbol is used for register indirect addressing. In the first instruction, it is showing that the R0 register is used. If the content of R0 is 40H, then that instruction will take the data which is located at location 40H of the internal RAM. In the second one, if the content of R1 is 30H, then it indicates that the content of port P0 will be stored at location 30H in the internal RAM.

```
MOVXA, @R1;
```

```
MOV@DPTR, A;
```

In these two instructions, the X in MOVX indicates the external data memory. The external data memory can only be accessed in register indirect mode. In the first instruction if the R0 is holding 40H, then A will get the content of external RAM location 40H. And in the second one, the content of A is overwritten in the location pointed by DPTR.

Indexed addressing mode

In the indexed addressing mode, the source memory can only be accessed from program memory only. The destination operand is always the register A. These are some examples of Indexed addressing mode.

```
MOVCA, @A+PC;
```

```
MOVCA, @A+DPTR;
```

The C in MOVC instruction refers to code bytes. For the first instruction, let us consider A holds 30H. And the PC value is 1125H. The contents of program memory location 1155H (30H + 1125H) are moved to register A.

Implied Addressing Mode

In the implied addressing mode, there will be a single operand. These types of instruction can work on specific registers only. These types of instructions are also known as register specific instructions. Here are some examples of Implied Addressing Mode.

```
RLA;
```

```
SWAPA;
```

These are 1- byte instructions. The first one is used to rotate the A register content to the Left. The second one is used to swap the nibbles in A.

Llojet dhe struktura e udhëzimeve në grupin e udhëzimeve të mikrokontrollerit 8051

Një udhëzim 8051 përbëhet nga Opcode (Operation Code) i ndjekur nga Operandët me madhësi:

- Zero Byte
- Një Byte
- Dy Bytes

Pjesa e Opcode-it e udhëzimit përmban Mnemonic, e cila specifikon llojin e operacionit që do të kryhet. Të gjitha Mnemonics ose pjesët e Opcode-it të udhëzimeve janë me madhësi Një Bajt. Duke ardhur në pjesën Operand të udhëzimit, ajo përcakton të dhënat që përpunohen nga udhëzimet.

Operand mund të jetë ndonjë nga këto:

- Jo Operand
- Vlera e të dhënave
- Porti I/O
- Vendndodhja e memories
- Regjistri i CPU-së

Një udhëzim i thjeshtë përbëhet vetëm nga opcode-i. Udhëzime të tjera mund të përfshijnë një ose më shumë operandë.

Bazuar në operacionin që ata kryejnë, të gjitha udhëzimet në bashkësinë e udhëzimeve të mikrokontrolluesit 8051 ndahen në pesë grupe.

Ato janë:

- Udhëzimet për transferimin e të dhënave
- Udhëzimet aritmetike
- Udhëzimet logjike
- Udhëzimet boolean or bit manipulation
- Udhëzimet e degës së programit

Udhëzimet për transferimin e të dhënave shoqërohen me transferimin e të dhënave ndërmjet regjistrave ose memories së jashtme të programit ose memories së jashtme të të dhënave

Mnemonic-a e lidhur me transferimin e të dhënave:

Mnemonic	Description
MOV	Move Data
MOVC	Move Code
MOCX	Move External Data
PUSH	Move Data to Stack
POP	Copy Data from Stack
XCH	Exchange Data between two Registers
XCHD	Exchange Lower Order Data between two Registers

Udhëzimet aritmetike

Duke përdorur udhëzimet aritmetike, ju mund të kryeni: mbledhjen, zbritjen, shumëzimin dhe pjesëtimin. Udhëzimet aritmetike përfshijnë edhe: rritjen për nga një (increment by one), zvogëlimin për nga një (decrement by one) dhe një udhëzim të veçantë të quajtur Akumulator i Rregullimit Decimal (Decimal Adjust Accumulator).

Mnemonic-a e lidhur me udhëzimet aritmetike:

Mnemonic	Description
ADD	Addition without Carry
ADDC	Addition with Carry
SUBB	Subtract with Carry
INC	Increment by 1
DEC	Decrement by 1
MUL	Multiply
DIV	Divide
DA A	Decimal Adjust the Accumulator (A Register)

Udhëzimet aritmetike nuk kanë njohuri në lidhje me formatin e të dhënave, d.m.th.: të nënshkruar, të pa nënshkruar, ASCII, BCD, etj. Gjithashtu, operacionet e kryera nga udhëzimet aritmetike ndikojnë në flamuj si: carry, overflow, zero, në Regjistrin e PSW, etj.

Të gjitha Mnemonikat e mundshme të shoqëruara me udhëzimet aritmetike:

Mnemonic	Instruction	Description	Addressing Mode	# of Bytes	# of Cycles
ADD	A, #Data	$A \leftarrow A + \text{Data}$	Immediate	2	1
	A, Rn	$A \leftarrow A + Rn$	Register	1	1
	A, Direct	$A \leftarrow A + (\text{Direct})$	Direct	2	1
	A, @Ri	$A \leftarrow A + @Ri$	Indirect	1	1
ADDC	A, #Data	$A \leftarrow A + \text{Data} + C$	Immediate	2	1
	A, Rn	$A \leftarrow A + Rn + C$	Register	1	1
	A, Direct	$A \leftarrow A + (\text{Direct}) + C$	Direct	2	1
	A, @Ri	$A \leftarrow A + @Ri + C$	Indirect	1	1
SUBB	A, #Data	$A \leftarrow A - \text{Data} - C$	Immediate	2	1
	A, Rn	$A \leftarrow A - Rn - C$	Register	1	1
	A, Direct	$A \leftarrow A - (\text{Direct}) - C$	Direct	2	1
	A, @Ri	$A \leftarrow A - @Ri - C$	Indirect	1	1
MUL	AB	Multiply A with B ($A \leftarrow \text{Lower Byte of } A*B$ and $B \leftarrow \text{Higher Byte of } A*B$)	--	1	4
DIV	AB	Divide A by B ($A \leftarrow \text{Quotient}$ and $B \leftarrow \text{Remainder}$)	--	1	4
DEC	A	$A \leftarrow A - 1$	Register	1	1
	Rn	$Rn \leftarrow Rn - 1$	Register	1	1
	Direct	$(\text{Direct}) \leftarrow (\text{Direct}) - 1$	Direct	2	1
	@Ri	$@Ri \leftarrow @Ri - 1$	Indirect	1	1
INC	A	$A \leftarrow A + 1$	Register	1	1
	Rn	$Rn \leftarrow Rn + 1$	Register	1	1
	Direct	$(\text{Direct}) \leftarrow (\text{Direct}) + 1$	Direct	2	1
	@Ri	$@Ri \leftarrow @Ri + 1$	Indirect	1	1
	DPTR	$DPTR \leftarrow DPTR + 1$	Register	1	2
DA	A	Decimal Adjust Accumulator	--	1	1

Udhëzimet Logjike

Udhëzimet Logjike, kryejnë veprime logjike si **AND, OR, XOR, NOT, Rotate, Clear** dhe **Swap**. Udhëzimet Logjike kryhen në Bytes të të dhënave në bazë bit-by-bit.

Mnemonic	Description
ANL	Logical AND
ORL	Logical OR
XRL	Ex-OR
CLR	Clear Register
CPL	Complement the Register
RL	Rotate a Byte to Left
RLC	Rotate a Byte and Carry Bit to Left
RR	Rotate a Byte to Right
RRC	Rotate a Byte and Carry Bit to Right
SWAP	Exchange lower and higher nibbles in a Byte

Udhëzimet Boolean or Bit Manipulation

Udhëzimet e manipulimit boolean ose bit merren me variablat bit. Ekziston një zonë e veçantë e adresueshme për bit në RAM dhe disa nga Regjistrat e Funksionit Special (SFR). Këto udhëzime mund të kryejnë: set, clear, and, or, complement, etj. në nivelin bit.

Mnemonic	Description
CLR	Clear a Bit (Reset to 0)
SETB	Set a Bit (Set to 1)
MOV	Move a Bit
JC	Jump if Carry Flag is Set
JNC	Jump if Carry Flag is Not Set
JB	Jump if specified Bit is Set
JNB	Jump if specified Bit is Not Set
JBC	Jump if specified Bit is Set and also clear the Bit
ANL	Bitwise AND
ORL	Bitwise OR
CPL	Complement the Bit

Udhëzimet e Degës së Programit

Këto udhëzime kontrollojnë rrjedhën e logjikës së programit. Të gjitha këto udhëzime, përveç NOP (Pa Operacion) ndikojnë në Numëruesin e Programit (PC) në një mënyrë ose në një tjetër. Disa nga këto udhëzime kanë aftësinë e vendimmarrjes para se të transferojnë kontrollin në pjesën tjetër të programit.

Mnemonic	Description
LJMP	Long Jump (Unconditional)
AJMP	Absolute Jump (Unconditional)
SJMP	Short Jump (Unconditional)
JZ	Jump if A is equal to 0
JNZ	Jump if A is not equal to 0
CJNE	Compare and Jump if Not Equal
DJNZ	Decrement and Jump if Not Zero
NOP	No Operation
LCALL	Long Call to Subroutine
ACALL	Absolute Call to Subroutine (Unconditional)
RET	Return from Subroutine
RETI	Return from Interrupt
JMP	Jump to an Address (Unconditional)

Llojet e memorieve

Memoriet ndahen në:

- Memorie sekondare
- Memorie primare
 - RAM
 - SRAM
 - DRAM
 - ROM
 - PROM
 - EPROM (erasable programmable read-only memory)
 - Hybrid
 - EEPROM (electrically erasable programmable read-only memory)
 - NVRAM (Non-Volatile Random Access Memory)
 - Flash Memory
 - Cache Memory
 - Virtual Memory

Memoriet Sekondare

Kompjuteri zakonisht përdor kanalet e tij të hyrjes / daljes për të hyrë në memorien sekondare dhe transferon të dhënat e dëshiruara duke përdorur zonën e ndërmjetme në memorien kryesore. Memoria sekondare nuk i humb të dhënat kur pajisja ndalet - është memorie e qëndrueshme. Për njësi, zakonisht kjo memorie është më pak e kushtueshme sesa memoria kryesore. Memoriet sekondare shpesh formatohet në përputhje me një format të sistemit të skedarit, i cili siguron abstraksionin e nevojshëm për të organizuar të dhënat në skedarë dhe drejtori, duke siguruar hapësirë shtesë në formim (të quajtur meta të dhëna) që përshkruajnë pronarin e një skedari të caktuar, kohën e hyrjes, lejimin e hyrjes dhe të dhëna tjera të nevojshme. Hard disk zakonisht përdoret si hapësirë ruajtëse dytësore. /*Secondary memory is not directly accessible by the CPU, the secondary memory devices are magnetic and optical memories.*/

Memoriet Primare

Storage kryesore (ose memoria kryesore ose memoria e brendshme), shpesh e referuar thjesht si memorie, është e vetmja memore e cila arrihet drejtpërdrejt nga CPU. CPU lexon vazhdimisht udhëzimet e ruajtura atje dhe i ekzekuton ato siç kërkohet. Memoria kryesore është e lidhur direkt ose indirekt me CPU-në përmes një bus-i të memories. Në të vërtetë janë dy bus-ë (jo në diagram): një bus adresash dhe një bus i të dhënash. CPU së pari dërgon një numër përmes një busi adresash, një numër i quajtur adresë memorie, që tregon vendndodhjen e dëshiruar të të dhënave. Pastaj lexon ose shkruan të dhënat vetë duke përdorur busin e të dhënave. Kjo memorie ndahet në RAM dhe ROM.

RAM

Kjo memorie përfshin dy memorie të rëndësishme: *Static RAM (SRAM)* dhe *Dynamic RAM (DRAM)*.

Dallimi kryesor në mes të SRAM dhe DRAM është koha e ruajtjes së të dhënave.

SRAM ruan përmbajtjen e tij për sa kohë që fuqia elektrike zbatohet në çip. Nëse energjia fiket ose humbet përkohësisht, përmbajtja e tij do të humbasë përgjithmonë.

DRAM- - ka një jetë jashtëzakonisht të shkurtër të të dhënave - tipikisht rreth katër milisekonda. Kjo memorie ekziston kur furnizimi me energji aplikohet vazhdimisht. Kontrolluesi DRAM përdoret për të rifreskuar të dhënat para se të skadojë koha, përmbajtja e memories mund të mbahet gjallë për aq kohë sa nevojiten. Një DRAM është më i dendur dhe më pak i kushtueshëm se një SRAM. Nga ana tjetër, një DRAM kërkon qarqet mbështetëse të rifreskimit. DRAM-et priren të favorizohen për kërkesa të mëdha të memories. SRAM janë disi më të shpejta se DRAM. Për shkak të këtyre karakteristikave relative, SRAM përdoret për memorie cache (si chip on dhe off), dhe DRAM përdoret për memorien kryesore.

SRAM	DRAM
SRAM has lower access time, which is faster compared to DRAM.	DRAM has a higher access time. It is slower than SRAM.
SRAM is costlier than DRAM.	DRAM cost is lesser compared to SRAM.
SRAM needs a constant power supply, which means it consumes more power.	DRAM requires reduced power consumption as the information stored in the capacitor.
SRAM offers low packaging density.	DRAM offers a high packaging density.
Uses transistors and latches.	Uses capacitors and very few transistors.
L2 and L3 CPU cache units are some general application of an SRAM.	The DRAM is mostly found as the main memory in computers.
The storage capacity of SRAM is 1MB to 16MB.	The storage capacity of DRAM is 1 GB to 16GB.
SRAM is in the form of on-chip memory.	DRAM has the characteristics of off-chip memory.
The SRAM is widely used on the processor or lodged between the main memory and processor of your computer.	The DRAM is placed on the motherboard.
SRAM is of a smaller size.	DRAM is available in larger storage capacity.
This type of RAM works on the principle of changing the direction of current through switches.	This type of RAM works on holding the charges.

Double Data Rate synchronous dynamic random access memory ose e njohur si **DDR1**

SDRAM është një klasë e qarqeve të integruara të memories që përdoren në kompjuterë.

Ndërfaqja përdor pompimin e dyfishtë (transferimin e të dhënave në të dy skajet në rritje dhe në rënie të sinjalit të orës) për të ulur frekuencën e orës. Një avantazh i mbajtjes poshtë të frekuencës së clock-ut është se zvogëlon kërkesat e integritetit të sinjalit në bordin e qarkut që lidh memorien me kontrolluesin. Memoria DDR2 është krejtësisht e ngjashme me DDR SDRAM. Për më tepër, ndërsa DDR SDRAM mund të transferojë të dhëna nëpër bus dy herë për clock, DDR2 SDRAM mund të realizojë katër-transferime për clock. DDR2 përdor të njëjtat qeliza memorie, por dyfishon gjerësinë e bandwidth duke përdorur teknikën e multipleksimit. Qeliza e memories DDR2 është akoma clocked në të njëjtën frekuencë si qelizat DDR SDRAM dhe SDRAM, por frekuenca e bufferëve të hyrjes / daljes është më e lartë me DDR2 SDRAM. Busi që lidh qelizat e memories me tamponët është dy herë më i gjerë krahasuar me DDR. Kështu, buffer-et I / O kryejnë multipleksim: të dhënat vijnë nga qelizat e memories përgjatë një busi të gjerë dhe dalin nga buffer-et në një bus me të njëjtën gjerësi si në DDR SDRAM, por me një frekuencë dy herë më të madhe. Kjo lejon rritjen e bandwidth-it të memories pa rritur frekuencën operacionale.

Ndërfaqja përdor pompimin e dyfishtë (transferimin e të dhënave në të dy skajet në rritje dhe në rënie të sinjalit të clock-ut për të ulur frekuencën e clock-ut. Një avantazh i mbajtjes poshtë të frekuencës së clock-ut është se zvogëlon kërkesat e integritetit të sinjalit në bordin e qarkut që lidh memorien me kontrolluesin.

EDRAM

Një teknologji gjithnjë e më e përhapur e përdorur në hierarkinë e memories është DRAM i ngulitur (eDRAM). eDRAM është një DRAM i integruar në të njëjtin çip ose MCM (multi-chip module) të një qarku të integruar specifik ose mikroprocesor. eDRAM është diku midis SRAM on-chip dhe DRAM off-chip:

- Për të njëjtën sipërfaqe, eDRAM siguron një memorie me madhësi më të madhe se SRAM, por më e vogël se DRAM pa chip (off chip).
- Kostoja për bit eDRAM është më e lartë kur krahasohet me chipa DRAM ekuivalente të përdorura si memorie të jashtme, por ka një kosto më të ulët për bit sesa SRAM.
- Koha e qasjes në eDRAM është më e madhe se SRAM por, për shkak të aftësisë për të përdorur bus më të gjerë, eDRAM siguron qasje më të shpejtë se DRAM.

Një shumëllojshmëri teknologjishë përdoren në fabrikimin e eDRAM, por në thelb ato përdorin të njëjtat dizajne dhe arkitekturë si DRAM.

ROM

Një read-only memory (ROM) është memorie vetëm për lexim dhe ruan të dhënat në menyrë të përhershëm dhe që nuk mund të ndryshohen. Një ROM është një memorie e qëndrueshme; domethënë nuk kërkohet asnjë burim energjie për të ruajtur të dhënat në memorie. Ndërsa është e mundur të lexoni një ROM, nuk është e mundur të shkruani të dhëna të reja në të. Një aplikim i rëndësishëm i ROM-ve është mikroprogramimi (microprogramming).

Aplikime të tjera të ROM janë:

- Nën-rutinat e librarive për funksionet e kërkuara shpesh
- Programet e sistemit
- Tabelat e funksioneve

Përparësia e ROM është se të dhënat ose programi është përgjithmonë në memorien kryesore dhe nuk ka nevojë të ngarkohet kurrë nga një pajisje ruajtëse dytësore. Një ROM krijohet si çdo çip tjetër i qarkut të integruar, me të dhënat përkatëse në çip si pjesë e procesit të fabrikimit.

Kjo paraqet dy probleme: hapi i futjes së të dhënave përfshin një kosto fikse relativisht të madhe, pavarësisht nëse janë fabrikuar një ose mijëra kopje të një ROM-mi të veçantë, dhe nuk ka vend për gabime. Nëse një bit është i gabuar, e gjithë grumbulli i ROM-ve duhet të hidhet.

Kur nevojitet vetëm një numër i vogël i ROM-ve me një përmbajtje të veçantë të kujtesës, një

alternativë më pak e kushtueshme është ROM-i i programueshëm (PROM). Ashtu si ROM edhe PROM është i qëndrueshëm dhe mund të shkruhet vetëm një herë.

Për PROM, procesi i shkrimit kryhet në mënyrë elektrike dhe mund të kryhet nga një furnizues ose klient në një kohë më vonë se sa prodhimi origjinal i çipit. Kërkohej pajisje speciale për procesin e shkrimit ose "programimit". PROM-të ofrojnë fleksibilitet dhe komoditet.

Një variant tjetër i memories vetëm për lexim është read-mostly memory, e cila është e dobishme për aplikime (qelizat më të vogla = më shumë qeliza për njësi zone) kur operacionet e leximit janë shumë më të shpeshta sesa operacionet e shkrimit, por për të cilat kërkohet ruajtje e qëndrueshme. Ekzistojnë tre forma të zakonshme të read-mostly memory: EPROM, EEPROM dhe memoria flash. Memoria erasable programmable read-only memory (EPROM) fshihet dhe shkruhet në mënyrë elektrike, si PROM. Sidoqoftë, para një operacioni të shkrimit, të gjitha qelizat e ruajtjes duhet të fshihen duke realizuar ekspozimin e çipit të paketuar ndaj rrezatimit ultravjollcë. Fshirja kryhet duke ndriçuar atë me një dritë ultravjollcë intensive përmes një dritare që është projektuar në çipin e memories. Ky proces i fshirjes mund të kryhet në mënyrë të përsëritur; çdo fshirje mund të kërkojë deri në 20 minuta për t'u kryer. Kështu, EPROM mund të ndryshohet shumë herë dhe, si ROM dhe PROM, i mban të dhënat praktikisht për një kohë të pacaktuar. EPROM është më e shtrenjtë se PROM por, ka avantazhin e aftësisë së azhurnimit të shumfışhtë.

Hybrid Memory

Ndërsa teknologjia e memorieve është pjekur në vitet e fundit, kufiri midis RAM dhe ROM është mjegulluar. Tani, disa lloje të memories kombinojnë tiparet e të dyjave. Këto pajisje nuk i përkasin asnjërit grup dhe mund të quhen kolektivisht si pajisje memorie hibride. Memoriet hibride mund të lexohen dhe shkruhen sipas dëshirës, si RAM, por mbajnë përmbajtjen e tyre pa energji elektrike, ashtu si ROM. Dy nga pajisjet hibride, EEPROM dhe flash, janë pasardhës të pajisjeve ROM. Këto përdoren zakonisht për të ruajtur kodin. Hibridi i tretë, NVRAM, është një version i modifikuar i SRAM. NVRAM zakonisht mban të dhëna të qëndrueshme.

EEPROM -Electrically Erasable Programmable Read-Only Memory.

Kjo është një memorie që mund të shkruhet në çdo kohë pa fshirë përmbajtjen paraprake; vetëm bajti ose bajtet e adresuara azhurnohen. Operacioni i shkrimit zgjat dukshëm më shumë se operacioni i leximit, me rendin e disa qindra mikrosekondave për bajt. EEPROM kombinon avantazhin e paqëndrueshmërisë me fleksibilitetin e azhurnimit, duke përdorur kontrollin e zakonshëm të bus-it, adresën dhe linjat e të dhënave. EEPROM është më e shtrenjtë se EPROM, gjithashtu është më pak i dendur, duke mbështetur më pak bit për chip.

Memoria flash (e quajtur kështu për shkak të shpejtësisë me të cilën mund të ri-programohet). Memoria flash është e ndërmjetme midis EPROM dhe EEPROM si nga kostoja ashtu edhe nga funksionaliteti. Ashtu si EEPROM, flash memoria përdor një teknologji elektrike të fshirjes.

Një memorie e tërë flash mund të fshihet në një ose disa sekonda, e cila është shumë më e shpejtë se EPROM. Sidoqoftë, memoria flash nuk siguron fshirje në nivelin e bajtave. Ashtu si EPROM, memoria flash përdor vetëm një tranzitor për bit, dhe kështu arrin dendësi më të lartë (krahasuar me EEPROM) të EPROM.

NVRAM (RAM jo i paqëndrueshëm)- një NVRAM është zakonisht vetëm një SRAM me një kopje rezervë të baterisë. Kur energjia është on, NVRAM funksionon ashtu si çdo SRAM tjetër. Kur energjia është off, NVRAM tërheq mjaftueshëm energji nga bateria për të ruajtur të dhënat e saj. NVRAM është mjaft e zakonshme në sistemet e ngulitura. Sidoqoftë, është i shtrenjtë - madje edhe më i shtrenjtë se SRAM, për shkak të baterisë - kështu që aplikimet e tij zakonisht kufizohen në ruajtjen e disa qindra bajtëve të informacionit kritik të sistemit që nuk mund të ruhen në ndonjë mënyrë më të mirë.

Cache Memory

Një cache memory e CPU-së është një cache memorie e përdorur nga njësia qendrore e përpunimit për të zvogëluar kohën mesatare të çasjes në memorien kryesore. Memoria cache është një memorie më e vogël, më e shpejtë e cila ruan kopjet e të dhënave nga vendet e memories kryesore që përdoren më shpesh. Kur procesori ka nevojë të lexojë ose të shkruajë në alokimin në memorien kryesore, së pari kontrollon nëse një kopje e atyre të dhënave është në memorien cache. Nëse është në cache, procesori menjëherë lexon ose shkruan në memorien cache, e cila është shumë më e shpejtë sesa të lexosh nga ose të shkruash në memorien kryesore. Çdo vendndodhje në secilën memorie ka një të dhënë (një vijë memorie cache), e cila në dizajne të ndryshme varion në madhësi nga 8 në 512 bajt. Madhësia e linjës së cache-it është zakonisht më e madhe se madhësia e hyrjes së zakonshme të kërkuar nga një udhëzim CPU, i cili varion nga 1 në 16 bajt.

Çdo vendndodhje në secilën memorie gjithashtu ka një indeks, i cili është një numër unik që përdoret për t'iu referuar asaj vendndodhjeje. Indeksi për alokim në memorien kryesore quhet adresë. Çdo vendndodhje në memorie ka një etiketë që përmban indeksin e të dhënave në kujtesën kryesore që është memorizuar. Në memorien e të dhënave të një CPU këto shënime quhen linja të memorjes ose blloqe të memories.

Virtual Memory

Është një teknikë e sistemit kompjuterik që i jep një programi aplikacioni përshtypjen se ai ka memorie pune të afërt (një ritëm adresash), ndërsa në fakt mund të jetë i fragmentuar fizikisht dhe madje mund të tejkalojë në ruajtjen e diskut. Sistemet operative kompjuterike zakonisht përdorin teknika të memories virtuale për aplikacione të zakonshme, të tilla si: përpunuesit e fjalëve, fletëllogaritje, multimedia, etj.

Assembly Language

Assembleri është paketë softverike që konverton instrukcionet në gjuhën e makinës. Gjuha e makinës është thjeshtë listë e numrave 8-bitësh që mikrokontrolleri mund t'i dekodoj. Me fjalë tjera, këta numra, që e përbëjnë programin dhe që ruhen në ROM memorie, i tregojnë mikrokontrollerit çfarë veprimi të ndërmarrë.

Bashkësia e instruksioneve

Instrukcionet 8051 kodin operacional e kanë 8 bitësh, gjithsej $= 2^8 = 256$.

Prej tyre 255 janë implementuar:

- 139 - Instrukcione 1-Bajtëshe
- 92 - Instrukcione 2-Bajtëshe
- 24 - Instrukcione 3-Bajtëshe

Instrukcionet

Instrukcionet i tregojnë procesorit se cilat operacione duhet të ekzekutohen. Shembull: MOV A, #5EH.

Kodi operacional dhe Operandi

Bajti i parë i instrukcionit paraqet kodin operacional (operation code) ngase ky është bajti që dekodohet nga procesori. Nga ky kod procesori e kupton se çfarë veprimi duhet të ndërmarrë. Instrukcionet 1-bajtëshe përbëhen vetëm nga kodi op (opcode). Te instrukcionet 2-bajtëshe bajti i dytë është operandi. Operandi mund të jetë e dhënë ose adresë 8-bitëshe. Te instrukcionet 3-bajtëshe bajti i dytë dhe i tretë, që të dy janë operandë; operandet 2-bajtësh zakonisht janë adresa 16-bitëshe.

Mikroprocesorët punojnë në kodin binar. Instrukcionet e shkruara në kodin binar duhet të jenë kod i makinës që paraqet një proces të lodhshëm. Alternativë është që të përdoren kode të shkurta për mostrat 0 dhe 1. P.sh., operacioni për shtimin e të dhënës së një akumulatori mund të paraqitet me ADDA. Ky kod i shkurtuar i referohet kodit mnemonik. Termi assembly language përdoret për këtë lloj kodi.

Në përgjithësi, instrukcionet mund të ndahen në:

- Transferi i të dhënave (Data transfer)
- Aritmetike (Arithmetic)
- Logjike (Logical)
- Kontrolli i programit (Program control)

Këto instruksione (urdhura) ndryshojnë prej një mikroprocesori në tjetrin. Disa prej tyre, megjithatë, janë të përbashkëta në shumicën e mikroprocesorëve.

Data Transfer

- **Load**

Ky instruksion lexon përmbajtjen e lokacionit të specifikuar të memories dhe e kopjon atë në lokacionin e specifikuar të regjistrit në CPU, p.sh. *Para instruksionit*: E dhëna në lokacionin e memories 0010. *Pas instruksionit*: E dhëna në lokacionin 0010 => E dhëna prej 0010 në akumulator.

- **Store**

Ky instruksion kopjon përmbajtjen aktuale të regjistrit të specifikuar në lokacionin e specifikuar të memories, p.sh., *Para instruksionit*: E dhëna në akumulator. *Pas instruksionit*: E dhëna në akumulator => E dhëna e kopjuar në lokacionin e memories 0011.

Arithmetic

- **Add**

Ky instruksion shton përmbajtjen e lokacionit të specifikuar të memories në të dhënë të ndonjë regjistri, p.sh., *Para instruksionit*: Akumulator me të dhënë 0001. Lokacioni i memories me të dhënë 0010. *Pas instruksionit*: Akumulator me të dhënë 0011.

- **Decrement**

Ky instruksion zbret 1 nga përmbajtja e lokacionit të specifikuar. P.sh., ne mund të kemi akumulator si lokacion i specifikuar dhe kështu: *Para instruksionit*: Akumulator me të dhënë 0011. *Pas instruksionit*: Akumulator me të dhënë 0010.

- **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 statusin e regjistrit si flag (flamuri).

Logical

- **AND**

Ky instruksion shërben operacionin logjik AND me përmbajtjen e lokacionit të memories dhe të dhënë në ndonjë regjister. AND vepron në numra bit për bit p.sh., *Para instruksionit*: Akumulator me të dhënë 0011. Lokacioni i memories me të dhënë 1001. *Pas instruksionit*: Akumulator me të dhënë 0001. Vetëm në bitin e fundit në të dhënë e mësipërme kemi 1 në të dy vargjet e të dhënës dhe operacioni AND jepë vetëm 1 në bitin e fundit të rezultatit.

- **EXCLUSIVE-OR**

Ky instruksion tregon operacionin logjik EXCLUSIVE-OR me përmbajtje të lokacionit të memories së specifikuar dhe me të dhënë në ndonjë regjistrë, bit për bit.

- **Logical shift (left-majtas apo right-djathtas)**

Instruksioni logical shift mundëson zhvendosjen e sekuencës (mostrës) së bitave në regjistrë për një vend në të majtë apo në të djathtë duke vendosur 0 në fund të numrit. P.sh., për logical shift right, 0 është zhvendosur në bitin signifikant ashtu që biti i fundit është zhvendosur për të batur flag (flamurin) në regjistrin e statusit. *Para instruksionit:* Akumulator me të dhënë 0011. *Pas instruksionit:* Akumulator me të dhënë 0001. Regjistri i statusit tregon Carry 1.

- **Arithmetic shift (left ose right)**

Instruksioni arithmetic shift mundëson zhvendosjen e mostrës së bitave në një regjistrë për një vend majtas ose djathtas, por ruan bitin e shenjës në fund majtas numrit. P.sh., për arithmetic right: *Para instruksionit:* Akumulator me të dhënë 1011. *Pas instruksionit:* Akumulatori me të dhënë 1101. Regjistri i statusit tregon Carry 1.

- **Rotate (left ose right)**

Instruksioni rotate mundëson lëvizjen e mostrës së bitave në regjistrë për një vend majtas ose djathtas dhe të bitit i cili përhapet prapa në një fund tjetër /*A rotation (or circular shift) is an operation similar to shift except that the bits that fall off at one end are put back to the other end. In left rotation, the bits that fall off at the left end are put back at the right end. In right rotation, the bits that fall off at right end are put back at left end*/ , p.sh., për rotate right: *Para instruksionit:* Akumulator me të dhënë 0011. *Pas instruksionit:* Akumulatori me të dhënë 1001.

Program control

- **Jump**

Ky instruksion ndryshon sekuencën në të cilën realizohen hapat e programit. Normalisht numratori i programit ndikon që programi të realizohet në mënyrë sekuenciale me një sekuencë numerike strikte. Instruksioni Jump shkakton që numratori i programit të kërcejë në ndonjë lokacion të specifikuar në program. P.sh., programit i nevojiten këto instruksione: Zvogëlimi i akumulatorit, kërcimi nëse akumulatori nuk është zero në instruksionin...

- **Branch**

Ky instruksion është kushtëzues që mund të jetë *branch if zero* ose *branch if plus*. Ky instruksion i degëzimit përcjellat nëse paraqiten kushte. P.sh., programit i nevojiten sekuencat e instruksioneve në diagramin vijues:



- **Halt**

Ky instruksion ndërpret të gjitha aktivitetet tjera të mikroprocesorit. Llojet e kodeve mnemonike që përdoren si instruksione në gjuhën assembler varen nga mikroprocesori/ mikrokontrolleri që shfrytëzohet.

Adresimi

Nëse kodi mnemonik, p.sh., LDA nevojitet që të specifikojë një instruksion, ai do të shoqërohet me informacionin shtesë për të treguar burimin dhe destinimin e të dhënës që kërkohet nga instruksioni (urdhri). E dhëna që e shoqëron instruksionin quhet operand. Ekzistojnë metoda të ndryshme që nevojiten për të specifikuar lokacionet e të dhënave, d.m.th., adresimi, pra mënyra se si programi e shtyen mikroprocesorin që të zgjedhë instruksionet ose të dhënat.

Mikroprocesorët e ndryshëm kanë mode të ndryshme të adresimit.

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.

Shembull i adresimit:

<u>Modi i adresës</u>	<u>Instrukcioni</u>	<u>Përshkrimi</u>
Immediate	LDA A #\$F0	Ngarko akumulatorin A me të dhënë F0
Direct	LDA A \$50	Ngarko akumulatorin A me të dhënë në adresën 0050
Extended	LDA A \$0F01	Ngarko akumulatorin A me të dhënë në adresën 0F01
Indexed	LDA A \$CF,X	Ngarko akumulatorin me të dhënë në adresën e dhënë me indeks regjistër plus CF
Inherent	CLR A	Pastro akumulatorin A
Extended	CLR \$2020	Pastro adresën 2020, d.m.th., vendos 0 në adresën 2020
Indexed	CLR \$10,X	Pastro adresën e dhënë me indeks regjistër plus 10, d.m.th., vendos 0 në atë adresë.

Programi në gjuhën programuese assembler përbëhet prej një numri të madh të instruksioneve që pastaj do të prodhojnë programin në kodin e makinës. Programi i shkruar në gjuhën assembler përbëhet prej sekuencave të formulime (gjendjeve), një formulim për rresht. Formulimi përbëhet prej katër pjesëve ose fushave duke filluar:

Label Op-code **Operand** **Comment**

Label – është emri të dhënës që i referohet memoria (shkronja, numra ose karaktere tjerë)

Op-code - tregon se si e dhëna do të manipulohet, p.sh., LDA A

Operand – jepë adresën e të dhënës me të cilën do të operohet.

Comment – kjo fushë është opsionale (programeri mund të japë komente).

Problemi: Mbledhja e dy numrave 8 bitësh të vendosur në dy adresa të ndryshme të memories, ndërsa vendosja e rezultatit të kthehet në memorie.

Algoritmi është:

1. Start.
2. Ngarko akumulatorin me numrin e parë. Akumulatori është aty ku rezultatet e operacioneve aritmetikore janë akumuluar. Ky është një regjistër punues, d.m.th., sikurse notepadi në të cilin kalkulimet bëhen para se rezultati të transferohet dikund tjetër. Kështu duhet të kopjojmë të dhënë në akumulator para se të bëjmë arithmetic. Me PIC është përdorur termi regjistri punues (w).
3. Shtojë numrin tjetër.
4. Vendos shumën në lokacionin e dedikuar të memories.
5. Stop.

Shembull: Mikrokontrolleri M68HC11

Mbledhja e dy numrave:

Label	Op-code	Operand	Comment
NUM1	EQU	\$00	; lokacioni i numrit 1
NUM2	EQU	\$01	; lokacioni i numrit 2
SUM	EQU	\$02	; lokacioni për shumën
	ORG	\$C000	; adresa e fillimit të userit RAM
START	LDAA	\$NUM1	; ngakro numrin 1 në akumulatorin A
	ADDA	\$NUM2	; shto numrin 2 te A
	STAA	SUM	; ruaje shumën në \$02
	END		

Shembull: Mikrokontrolleri 8051

Mbledhja e dy numrave:

Label	Op-code	Operand	Comment
NUM1	EQU	20H	; lokacioni i numrit 1
NUM2	EQU	21H	; lokacioni i numrit 2
SUM	EQU	22H	; lokacioni për shumën
	ORG	8000H	; adresa e fillimit të userit RAM
START	MOV	A, NUM1	; ngarko numrin 1 në akumulatorin A
	ADD	A, NUM2	; shto numrin 2 te A
	MOV	SUM, A	; ruaje shumën në adresën 22H
	END		

Looping

Në shumë programe mund të ketë kërkesa që detyra të kryhet me numër të limituar të përsëritjeve. Në këto raste programin duhet bërë ashtu që operacioni të kalojë nëpër pjesën e njëjtë disa herë. Ky term quhet looping, një loop (unazë) që është pjesë e programit dhe përsëritet disa herë.

Problemi 1. 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 simplimi).

Algoritmi mund të jetë:

1. Start
2. Vendos 10 te numratori.
3. Shko (Point) në lokacionin e numrit me adresë të fundit.
4. Shto numrin me adresë të fundit.
5. Zvogëlo numratori për 1.
6. Shto 1 në pointerin me adresë të lokacionit.
7. A është numratori 0? Nëse jo shko në 4. Nëse po vazhdo.
8. Vendos shumën.
9. Stop.

Problemi 2: Përcaktimi i numrit më të madh në regjistrin e numrave (ai mund të jetë p.sh., përcaktimi i temperaturës më të lartë që është lexuar nga sensorët e temperaturës).

Algoritmi mund të jetë:

1. Pastro adresën përgjigje.
2. Listo adresën fillestare.
3. Ngarko numrin nga adresa fillestare.
4. Krahase numrin me numër në adresën përgjigje.
5. Vendos përgjigje nëse është më e madhe.
6. Përndryshe ruaje numrin.
7. Rrit adresën fillestare për 1.
8. Degezë (shko) në 3 nëse adresa nuk është adresa e fundit.
9. Stop.

Shembull: Mbledhja e dy numrave

Label	Op-code	Operand	Comment
FIRST	EQU	\$0030	
LAST	EQU	\$0040	
ANSW	EQU	\$0041	
	ORG	\$0000	
	CLR	ANSW	; Fshije pergjigje
	LDX	FIRST	; Ngarko adresen e pare
NUM	LDA A	\$30,X	; Ngarko numrin
	CMP A	ANSW	; Krahase me pergjigje
	BLS	NEXT	; Degezimi tek NEXT nese me e vogel ose e njejte
	STA A	ANSW	; Vendos pergjegje
NEXT	INX		; Rrite regjistrin me index
	CPX	LAST	; Krahase regjistrin me index me LAST
	BNE	NUM	; Degezo nese nuk eshte baras me zero
	WAI		; Stop programin

Procedura është e tillë që së pari adresa përgjegje duhet pastruar. Adresa e parë pastaj ngarkohet dhe numri në atë adresë vendoset në akumulatorin A. LDA A \$30,X nënkupton që duhet ngarkuar akumulatorin A me të dhënë tek adresa e dhënë me regjister indeksi plus 30. Krahase numrin me përgjegje, duke mbajtur numrin nëse është më i madh se numri që gjendet në akumulator, përndryshe degëzohet për të përsëritur unazën me numrin e radhës.

1. Shembull:

Cilat janë mnemonikët për Motorola 68000 për:

- 1) Pastro regjistrin A;
- 2) Vendos akumulatorin A;
- 3) Ngarko akumulatorin A;
- 4) Krahase akumulatorët;
- 5) Ngarko regjistrin indeksor.

Zgjidhja:

- 1) CLR A
- 2) STA A
- 3) LDA A
- 4) CB A
- 5) LD X

2. Shembull:

Shkruani një rresht të programit në Asembler për:

- 1) Load accumulator me 20 (hex);
- 2) Decrement accumulator A;
- 3) Clear address \$0020;
- 4) ADD accumulatorin e numrin A në adresën \$0020.

Zgjidhja:

- 1) LDA A \$20
- 2) DEC A
- 3) CLR \$0020
- 4) ADD A \$0020

3. Shembull

Sqaroni operacionet e specifikuara me instruksionet në vijim:

- 1) STA B \$35
- 2) LDA A #\$F2
- 3) CLC
- 4) INC A
- 5) CMP A #\$C5

- 6) CLR \$2000
- 7) JMP 05, X

Zgjidhja:

- 1) Vendos akumulatorin e vlerës B në adresën 0035
- 2) Ngarko akumulatorin A me të dhënë F2
- 3) Pasto flamurin bartës
- 4) Shto 1 në vlerën e akumulatorit A
- 5) Krahaso C5 me vlerën e akumulatorit A
- 6) Pastro adresën 2000
- 7) Kërce adresën e dhënë me index regjistër plus 05