



UNIVERSITETI I PRISHTINËS
FAKULTETI I SHKENCAVE MATEMATIKE–NATYRORE
DEPARTAMENTI I MATEMATIKËS

PUNIM SEMINARIK

Lënda: Arkitektura e kompjuterit dhe sistemet operative.
Tema: Dizajni i njësisë përpunuese qendrore (CPU).
Punuar nga: Fidan Abdiu, student.

Prishtinë, Janar 2007.

PËRMBAJTJA:

1. Arkitektura e njësisë përpunuese (CPU).
2. Bashkësia e regjistrave.
 - 2.1. Regjistrat “Memory Access Registers”.
 - 2.2. Regjistrat “Instruction Fetching Registers”.
 - 2.3. Regjistrat “Condition Registers”.
3. Shtegu i të dhënave (Datapath).
 - 3.1. Shtegu i të dhënave me një magjistrale.
 - 3.2. Shtegu i të dhënave me dy magjistrale.
 - 3.3. Shtegu i të dhënave me tri magjistrale.
4. Cikli i ekzekutimit të instruksionit.
 - 4.1. Leximi i instruksionit.
 - 4.2. Shembull: Ekzekutimi i një veprimi të thjeshtë aritmetik.
 - 4.3. Procedura “Interrupt Handling”.
5. Njësia kontrolluese (CU).
6. Përmbledhje.
7. Literatura.

1. Arkitektura e njësisë përpunuese (CPU).

Para se të flasim për arkitekturën e njësisë përpunuese, të theksojmë se çdo sistem kompjuterik përbëhet prej tri pjesëve kryesore:

1. Njësia përpunuese (CPU),
2. Memoria, dhe
3. Sistemi për hyrje/dalje (I/O System).

Tani mund të shtrohen pyetjet:

Çfarë funksioni ka memoria?

Memoria ka për funksion ruajtjen e të dhënave dhe programeve. Programet paraqesin bashkësi instruksionesh.

Çfarë funksioni ka sistemi për hyrje/dalje?

Sistemi për hyrje/dalje paraqet mekanizmin për bartjen e të dhënave brenda dhe jashtë rrjetës (Sistemi kompjuterik mund të paraqitet si rrjetë në të cilën lidhen pjesët përbërëse të tij).

Tani le të flasim për pjesën më kryesore të çdo sistemi kompjuterik e që është njësia përpunuese.

Çfarë funksioni ka njësia përpunuese?

Njësia përpunuese ka për funksion ekzekutimin e instruksioneve të cilët lexohen nga memoria. Më vonë do të shohim më gjerësisht se si bëhet leximi dhe ekzekutimi i instruksioneve.

Çdo njësi përpunuese përbëhet prej tri pjesëve kryesore:

1. Bashkësia e regjistrave.
2. Njësia aritmetike-logjike (ALU), dhe
3. Njësia kontrolluese (CU).

Çfarë janë regjistrat?

Regjistrat janë pjesë harduerike, të cilat për funksion kanë ruajtjen e disa të dhënave.

Regjistrat janë të vendosur brenda njësisë përpunuese, në mënyrë që kjo e fundit t'i shfrytëzojë më shpejtë të dhënat që ndodhen në regjistra. Të gjithë regjistrat e një njësie përpunuese përbëjnë bashkësinë e regjistrave të njësisë përpunuese.

Bashkësitë e regjistrave dallojnë nga një arkitekturë në tjetrën, por në përgjithësi, regjistrat klasifikohen në dy grupe:

1. Regjistrat “General-Purpose”, dhe
2. Regjistrat “Special-Purpose”.

Regjistrat “General-Purpose” kanë funksione të shumta, ndërsa, regjistrat “Special-Purpose” kanë funksione të veçanta. Si shembull mund ta marrim regjistrin PC (Program Counter) i cili është “Special-Purpose” regjistër, në të cilin ndodhet adresa e instruksionit që do të ekzekutohet në vijim. Do të shohim më vonë se ekzistojnë edhe klasifikime të tjera të regjistrave.

Çfarë është njësia aritmetike-logjike?

Njësia aritmetike-logjike paraqet bashkësinë e qarqeve përmes të cilëve kryhen veprimet aritmetike dhe logjike me të dhëna.

Çfarë është njësia kontrolluese?

Edhe njësia kontrolluese paraqet bashkësi të qarqeve e cila është përgjegjëse për leximin, dekodimin dhe ekzekutimin e instruksionit.

Tani le ta shohim arkitekturën e njësisë përpunuese, bashkëveprimin e saj me memorien dhe sistemin për hyrje/dalje.

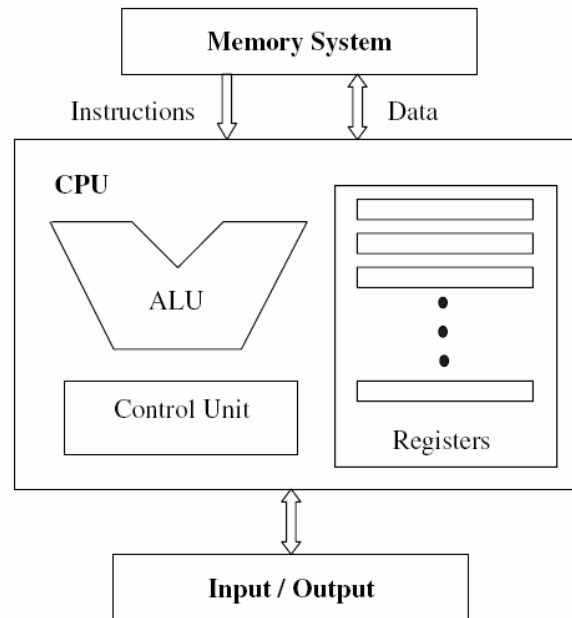


Figura 1: Arkitektura e njësisë përpunuese.

Siç theksuam më lartë, njësia përpunuese ka për detyrë ekzekutimin e instruksioneve. Ato instruksione së pari duhet të lexohen nga memoria ku edhe ruhen. E gjithë kjo procedurë mund të përmblihet në një cikël i cili njihet si cikli i ekzekutimit të instruksionit.

1. Instruksioni i cili do të ekzekutohet në vijim, adresa e të cilit ndodhet në regjistrin PC (Program Counter), vendoset në regjistrin IR (Instruction Register).
2. Instruksioni dekodohet.
3. Operandët (adresat) e instruksionit lexohen nga memoria dhe vendosen në regjistrat e njësisë përpunuese.

-
4. Instruksioni ekzekutohet, dhe në fund
 5. Rezultatet ruhen në memorie.

Të mos harrojmë se ky cikël përsëritet gjithnjë përdërisa ka ende instruksione që presin të ekzekutohen.

Veprimet e njësisë përpunuese gjatë ciklit të ekzekutimit të instruksionit përcaktohen nga mikro-procedurat (Micro-orders) të cilat kryhen nga ana e njësisë kontrolluese. Në fakt, mikro-procedurat paraqesin sinjale kontrolli të cilat u dërgohen pjesëve përkatëse për kryerjen e detyrave të caktuara.

Ta sqarojmë këtë me një shembull.

Të supozojmë se dëshirojmë ta bartim përmbajtjen e regjistrit X në regjistrin Y . Të supozojmë po ashtu se të dy regjistrat janë të lidhur përmes magjistrales D (Siç e dimë të gjitha pjesët e sistemit janë të lidhura në magjistrale e cila mundëson bartjen e të dhënave).

Në këtë rast, njësia kontrolluese ia dërgon një sinjal kontrolli regjistrit X në mënyrë që ky i fundit ta vendosë përmbajtjen e tij në magjistralen D . Pas një kohe, edhe regjistrit Y i dërgohet një sinjal kontrolli në mënyrë që ky i fundit tani ta lexojë atë të dhënë nga magjistrala D .

2. Bashkësia e regjistrave.

Tani do të flasim më gjerësisht për bashkësinë e regjistrave.

Siç thamë më parë, regjistrat janë pjesë harduerike në të cilët ruhen të dhënat. Regjistrat janë të vendosur brenda njësisë përpunuese, në mënyrë që kjo e fundit t'i shfrytëzojë më shpejtë ato të dhëna.

Bashkësitë e regjistrave dallojnë nga një arkitekturë në tjetrën. Ato dallojnë për nga sasia e regjistrave, lloji i tyre, për nga gjatësia e secilit regjistër dhe për nga funksioni i secilit regjistër.

Për shembull, regjistrat e adresës (regjistra që ruajnë adresën e ndonjë lokacioni të caktuar të memories) duhet të jenë aq të gjatë sa të mund ta mbajnë adresën me gjatësinë më të madhe.

Regjistrat “General-Purpose” mund të përdoren për qëllime të shumta, ndërsa, regjistrat “Special-Purpose” mund të përdoren vetëm për qëllime të veçanta.

2.1. Regjistrat “Memory Access Registers”.

Regjistrat “Memory Access Registers” për funksion kanë qasjen në memorie, përkatësisht leximin dhe shkrimin në memorie.

Në leximin dhe shkrimin e të dhënave në memorie marrin pjesë këta dy regjistra me rëndësi:

1. Regjistri MDR (Memory Data Register), dhe
2. Regjistri MAR (Memory Address Register).

Të shohim tani se si bëhet shkrimi në një lokacion të caktuar të memories:

-
1. Fjala që do të shkruhet në lokacionin e caktuar të memories kopjohet në MDR.
 2. Adresa e lokacionit të memories në të cilin do të vendoset fjala kopjohet në MAR.
 3. Dërgohet një sinjal kontrolli nga njësia kontrolluese në mënyrë që ajo fjalë të shkruhet në lokacionin e caktuar të memories.

Në mënyrë analoge lexohet fjala nga lokacioni i caktuar i memories. Procedura e tillë zhvillohet si në vijim:

1. Adresa e lokacionit të memories nga i cili do të lexohet fjala, vendoset në MAR.
2. Dërgohet një sinjal kontrolli nga njësia kontrolluese në mënyrë që fjala përkatëse të lexohet.
3. Fjala vendoset në MDR, tani e gatshme për përpunim nga njësia përpunuese.

2.2. Regjistrat “Instruction Fetching Registers”.

Regjistrat “Instruction Fetching Registers” për detyrë kanë leximin e instruksionit që do të ekzekutohet në vijim.

Në leximin e instruksioneve marrin pjesë këta dy regjistra me rëndësi:

1. Regjistri PC (Program Counter), dhe
2. Regjistri IR (Instruction Register).

Regjistri PC mbanë adresën e instruksionit i cili do të lexohet në vijim.

Pasi të lexohet instruksioni nga memoria, instruksioni vendoset në regjistrin IR, tani i gatshëm për ekzekutim.

Pas ekzekutimit të instruksionit në PC vendoset adresa e instruksionit tjetër që do të lexohet dhe më pas të ekzekutohet.

2.3. Regjistrat “Condition Registers”.

Regjistrat “Condition Registers” për funksion kanë ruajtjen e informatave në lidhje me gjendjen momentale në njësinë përpunuese. Disa arkitektura përmbajnë regjistrin e llojit “Special-Purpose” PSW (Program Status Word). Regjistri PSW i përmban bitët të cilët ia dërgon njësia përpunuese, për ta treguar gjendjen momentale në lidhje me ekzekutimin e programit.

3. Shtegu i të dhënave (Datapath).

Njësia përpunuese mund të ndahet në dy pjesë:

1. Pjesa e të dhënave, dhe
2. Pjesa e kontrollit.

Pjesa e të dhënave apo shtegu i të dhënave përbëhet prej bashkësisë së regjistrave dhe njësisë aritmetike-logjike.

Shtegu i të dhënave ka për detyrë kryerjen e veprimeve të caktuara mbi të dhënat.

Pjesa e kontrollit, në fakt, paraqet njësinë kontrolluese. Njësinë kontrolluese i dërgon sinjale kontrolli shtegut të dhënave për kryerjen e detyrave të caktuara.

Brenda njësisë përpunuese të dhënave barten në këto relacione:

1. Regjistër-regjistër, dhe
2. ALU – regjistër.

Këto bartje mundësohen përmes magjistraleve të caktuara.

Bartja e të dhënave nga ana e jashtme, d.m.th., nga regjistrat në memorie dhe në pajisjet e sistemit për hyrje/dalje, zakonisht mundësohet nga ana e magjistrales së sistemit e cila mundëson komunikim e këtyre pjesëve të sistemit.

Bartja e brendshme e të dhënave, d.m.th. bartja në relacionet regjistër-regjistër apo ALU-regjistër mund të kryhet duke përdorur:

1. Një magjistrale,
2. Dy magjistrale, ose
3. Tri magjistrale.

3.1. Shtegu i të dhënave me një magjistrale.

Në shtegun e të dhënave me një magjistrale, bartja e të dhënave në relacionet e lartpërmendura mundësohet nga një magjistrale e vetme.

Meqenëse me një magjistrale mund të kryhet vetëm një veprim përkatësisht një bartje e të dhënave brenda një cikli të orës, atëherë, kjo ndikon në ngadalësimin e procesit të bartjes së të dhënave.

Normalisht që kjo pastaj ndikon në efektivitetin e përgjithshëm të sistemit.

Tani le të shohim se si funksionon bartja e të dhënave në mes të pjesëve të shtegut të dhënave me një magjistrale.

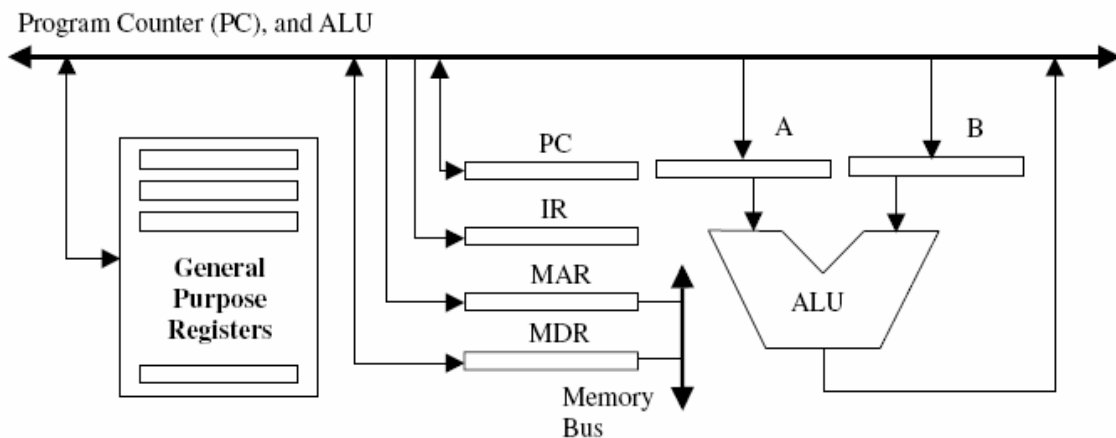


Figura 2: Shtegu i të dhënave me një magjistrale.

3.2. Shtegu i të dhënave me dy magjistrale.

Bartja e të dhënave brenda shtegut të dhënave me dy magjistrale është zgjidhje më e mirë se ajo paraprake. Meqenëse kemi dy magjistrale, atëherë, brenda një cikli të orës mund të kryhen dy veprime. Tani le ta shohim bartjen e të dhënave brenda shtegut të dhënave me dy magjistrale:

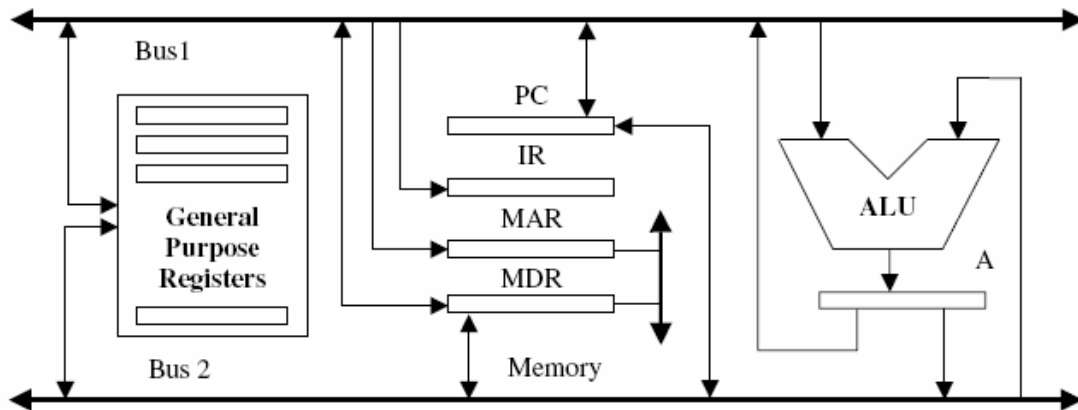


Figura 3: Shtegu i të dhënave me dy magjistrale.

3.3. Shtegu i të dhënave me tri magjistrale.

Në shtegun e të dhënave me tri magjistrale, dy magjistrale (**Out-bus 1, 2**) mund t'i përdorim për nxjerrjen e të dhënave nga regjistrat, ndërsa, magjistralen e tretë (**In-bus**) mund ta përdorim për të futur të dhëna në regjistra.

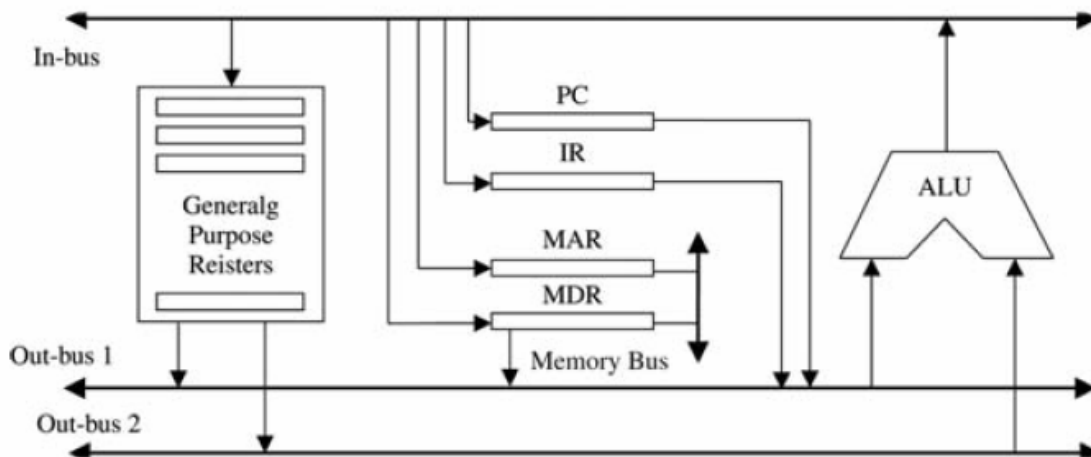


Figura 4: Shtegu i të dhënave me tri magjistrale.

Çfarë kuptuam nga shembujt e mësipërm?

Nga shembujt e mësipërm kuptuam se sa më shumë magjistrale të kemi në shtegun e të dhënave aq më e shpejtë do të jetë bartja e të dhënave brenda pjesëve përkatëse.

Të mos harrojmë se sa më shumë magjistrale të kemi, aq më kompleks do të jetë hardueri.

4. Cikli i ekzekutimit të instruksionit.

Siç kemi thënë më parë, cikli i ekzekutimit të instruksionit vazhdon të përsëritet përdherisa ka ende instruksione që presin të ekzekutohen.

Instruksioni ekzekutohet bazuar në fushën e Opcode-it (Varg i caktuar bitësh) ku përcaktohet veprimi apo lloji i instruksionit. Pas ekzekutimit të instruksionit, bëhet një test në rast të ndërprerjeve (Interrupts) të mundshme.

Në rast se ka ndonjë ndërprerje në punën e njësisë përpunuese, atëherë ekziston një procedurë përkatëse për rregullimin e saj “Interrupt Handling”. Do të shohim më vonë se si zhvillohet procedura “Interrupt Handling”.

Le ta shohim vargun e hapave për ekzekutimin e instruksionit:

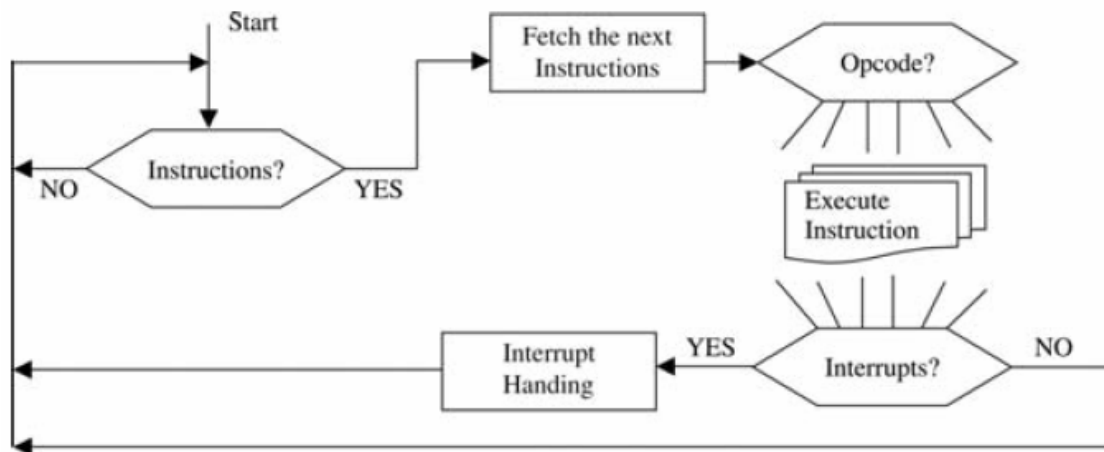


Figura 5: Cikli i ekzekutimit të instruksionit.

4.1. Leximi i instruksionit.

Të gjitha veprimet gjatë ciklit të ekzekutimit të instruksionit siç ishin: leximi i instruksionit, ekzekutimi i instruksionit apo “Interrupt Handling” mund të paraqiten si vargje të mikro-veprimeve (Micro-operations).

Të shohim se si leximi i instruksionit nga memoria mund të paraqitet si varg i mikro-veprimeve:

1. Përmbajtja në PC kopjohet në MAR,
2. Vlera në PC rritet,

3. Instruksioni vendoset në MDR, dhe në fund
4. Nga MDR, instruksioni vendoset në IR.

4.2. Shembull: Ekzekutimi i një veprimi të thjeshtë aritmetik.

Të supozojmë se kemi instruksionin **Add R₁, R₂, R₀**.

Çfarë bën instruksioni i mësipërm?

Ky instruksion ia shton regjistrin **R₂**, përmbajtjen e regjistrin **R₁**, dhe e ruan rezultatin në regjistrin **R₀**.

Vargu i hapave për ekzekutimin e këtij instruksioni paraqitet si në vijim:

1. Regjistrat **R₀**, **R₁**, dhe **R₂** ekstrahohen nga IR.
2. Përmbajtjet e regjistrave **R₁** dhe **R₂** transferohen në ALU për mbledhje, dhe në fund
3. Rezultati ruhet në regjistrin **R₀**.

Në qoftë se kemi shtegun e të dhënave me një magjistrale atëherë mbledhja e tillë mund të kryhet në tre hapa, siç tregon tabela:

Step	Micro-operation
t_0	$A \leftarrow (R_1)$
t_1	$B \leftarrow (R_2)$
t_2	$R_0 \leftarrow (A) + (B)$

Tabela 1: Ekzekutimi i instruksionit **Add R₁, R₂, R₀** në tre hapa.

Në qoftë se kemi shtegun e të dhënave me dy magjistrale, atëherë, instruksioni për mbledhjen e mësipërme do të kryhet në dy hapa:

Step	Micro-operation
t_0	$A \leftarrow (R_1) + (R_2)$
t_1	$R_0 \leftarrow (A)$

Tabela 2: Ekzekutimi i instruksionit **Add R₁, R₂, R₀** në dy hapa.

Në qoftë se kemi shtegun e të dhënave me tri magjistrale, atëherë, instruksioni për mbledhjen e mësipërme do të kryhet po ashtu në dy hapa:

Step	Micro-operation
t_0	$A \leftarrow (R_1)$
t_1	$R_0 \leftarrow (A) + (R_2)$

Tabela 3: Ekzekutimi i instruksionit **Add R₁, R₂, R₀** në dy hapa.

4.3. Procedura “Interrupt Handling”.

Siç kemi thënë më parë, pas çdo ekzekutimi të instruksioneve, bëhet një test nëse ka ndonjë ndërprerje (Interrupts).

Pse shkaktohen ndërprerjet?

Ndërprerjet mund të shkaktohen nga kërkesat e sistemit për hyrje/dalje, për shkak të gabimeve aritmetikore (për shembull pjesëtimi me zero) ose kur jepen instruksione të pavlefshme.

Procedura “Interrupt Handling” zhvillohet si varg i këtyre hapave:

1. Përmbajtja në PC kopjohet në MDR (në mënyrë që ajo të ruhet),
2. Në MAR kopjohet adresa ku do të ruhet përmbajtja që është në MDR,
3. Në PC kopjohet adresa e instruksionit të parë të procedurës “Interrupt Handling”, dhe
4. Përmbajtja që ndodhet në MDR (përmbajtja e vjetër që ishte në PC) ruhet në memorie.

Tabela e mëposhtme na tregon se si zhvillohet kjo procedurë:

Step	Micro-operation
t_1	MDR \leftarrow (PC)
t_2	MAR \leftarrow address1 (where to save old PC); PC \leftarrow address2 (interrupt handling routine)
t_3	Mem[MAR] \leftarrow (MDR)

Tabela 4: Zhvillimi i procedurës “Interrupt Handling”.

5. Njësia kontrolluese (CU).

Njësia kontrolluese paraqet pjesën kryesore të njësisë përpunuese. Ajo e udhëheqë punën brenda njësisë përpunuese duke i dërguar sinjale kontrolli shtegut të dhënave për kryerjen e detyrave të saj.

Njësia kontrolluese kontrollon rrjedhën e të dhënave brenda njësisë përpunuese dhe është përgjegjëse për leximin, dekodimin dhe ekzekutimin e instruksioneve.

Njësinë kontrolluese mund ta krahasojmë me “policin në trafik” i cili kontrollon trafikun. Ora sistimore prodhon në mënyrë të vazhdueshme vargje të pulseve në një frekuencë të caktuar. Instruksionet ekzekutohen në një varg të caktuar hapash.

Tani le të shohim figurën:

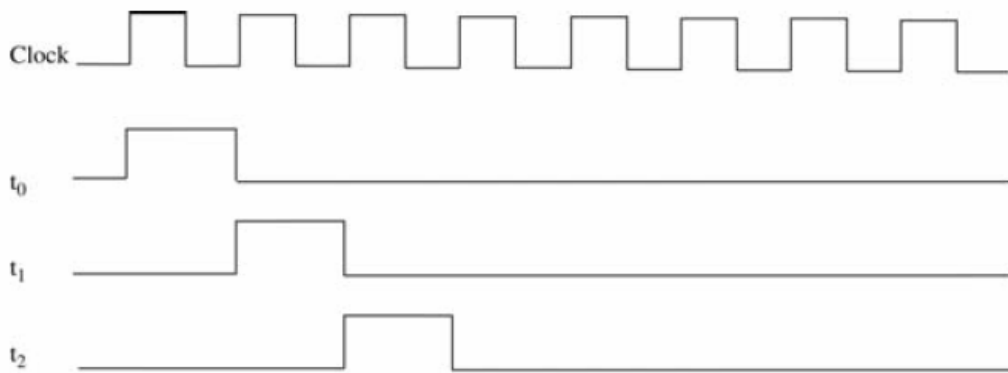


Figura 6: Prodhimi i pulseve nga ora sistemore.

Siç e theksuam më lartë, njësia kontrolluese i udhëheqë të gjitha punët brenda njësisë përpunuese përmes dërgimit të sinjaleve të kontrollit.

Tani mund të shtrohet pyetja:

Si gjenerohen sinjalet e tilla?

Së pari, fusha e Opcode-it e instruksionit të lexuar nga memoria dekodohet nga një dekodues i caktuar. Pas kësaj, gjeneruesi i sinjaleve të kontrollit pajiset me informacione mbi instruksionin i cili do të ekzekutohet (Informacione që kanë të bëjnë me llojin e instruksionit). Përveç dekoduesit dhe gjeneruesit të sinjaleve të kontrollit, në gjenerimin e sinjaleve, merr pjesë edhe një qark logjik i cili vendosë për hapin që do të ndërmerret për ekzekutimin e instruksionit.

Tani le të shohim përmes figurës së mëposhtme se si bëhet gjenerimi i sinjaleve të kontrollit:

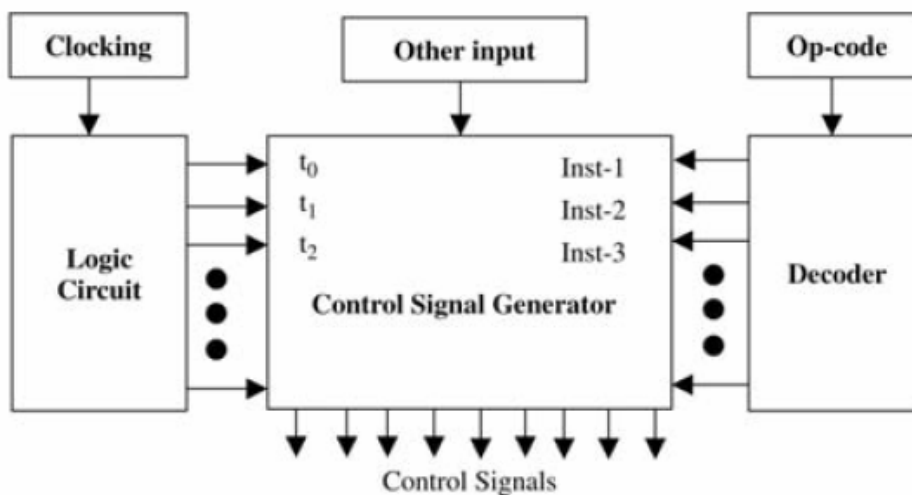


Figura 7: Gjenerimi i sinjaleve të kontrollit.

6. Përmbledhje.

Çfarë mësuam nga ky punim?

Nga ky punim mësuam se:

1. Njësia përpunuese i ekzekuton instruksionet të cilët janë pjesë përbërëse të programeve që ne i shkruajmë.
2. Njësia përpunuese përbëhet prej: bashkësisë së regjistrave, njësisë aritmetike-logjike (ALU) dhe njësisë kontrolluese (CU).
3. Regjistrat janë pjesë harduerike që ruajnë të dhëna.
4. Regjistrat janë të vendosur brenda njësisë përpunuese në mënyrë që kjo e fundit t'i shfrytëzojë të dhënat më shpejtë.
5. Njësia aritmetike-logjike i kryen veprimet aritmetike dhe logjike mbi të dhënat.
6. Njësia kontrolluese është pjesa kryesore e njësisë përpunuese.
7. Njësia kontrolluese i udhëheqë punët brenda njësisë përpunuese përmes dërgimit të sinjaleve të kontrollit shtegut të dhënave.

7. Literatura.

1. "Fundamentals of Computer Organization and Architecture", Mostafa Abd-El-Barr, Hesham El-Rewini, John Wiley & Sons, Inc. © 2005.
2. "The Essentials of Computer Organization and Architecture", Linda Null, Julia Lobur, Jones and Bartlett Publishers © 2003.