Universiteti i Prishtinës "Hasan Prishtina" Kosovë

Arkitektura e kompjuterit

Pjesa 5 – Arkitekturat e bashkësive të instruksioneve

Prof. Asoc. Dr. Ermir Rogova

Objektivat

- Të kuptojmë faktorët prej të cilëve varet dizanjimi i arkitekturës të bashkësisë së instruksioneve (ISA)
- T'i njohim më mirë regjimet e adresimit të memories
- Të njihemi me kuptimin e "gypimit" (pipeline) në nivel instruksioni dhe efektin e tij në procesin e ekzekutimit

Hyrje

- Do t'i bëjmë një vështrim të detajuar formateve të instruksioneve, tipeve të operandëve dhe mënyrave për qasje memories
- Do të shohim ndërlidhjen në mes të organizimit të makinës dhe formateve të instruksioneve
- Kjo do të na ndihmojë ta kuptojmë më mirë arkitekturën e kompjuterit

Formatet e instruksioneve

- Bashkësitë e instruksioneve dallohen sipas këtyre elementeve:
  - Numri i bitëve për instruksion (tek MARIE instruksionet janë 16-bitëshe)
  - Numri i operandëve eksplicitë për instruksion (0-3 është rasti më i shpeshtë)
  - Të bazuara në pirg (stack) ose në regjistër (MARIE bazohet në regjistër)
  - Lokacioni i operandëve (instruksionet mund të jenë të tipit regjistër-regjistër, regjistër-memorie, memorie-memorie që ka të bëjë me kombinimet e lejueshme të operandëve)
  - Operacionet (veprimet)
  - Tipet dhe madhësia e operandëve (operandët mund te jenë adresa, numra, ose edhe simbole)

Formatet e instruksioneve

- Gjatë disenjimit të arkitekturës së kompjuterit, së pari duhet të përcaktohet ISA
- Kjo është punë shumë komplekse, sepse ISA duhet të jetë konsistente me arkitekturën, ndërsa arkitektura e disenjuar mirë mund të zgjasë me dekada
- Kriteret që përdoren në këtë rast janë:
  - Hapësira e memories punuese që okupohet nga një program
  - Kompleksitetit të instruksioneve
  - Gjatësia e instruksioneve (në bit)
  - Numri i përgjithshëm i instruksioneve në bashkësinë e instruksioneve (IS)

Formatet e instruksioneve

- Në procesin e disenjimit të IS, duhet pasur parasysh:
  - Gjatësinë e instruksionit (i shkurtër, i gjatë ose variabil)
    - Instruksionet e shkurtëra janë zakonisht më të mira, sepse zënë më pak hapësirë dhe transmetohen më lehtë. Sidoqoftë, gjatësia e kufizuar kufizon numrin e përgjithshëm të instruksioneve, si dhe numrin e operandëve që mund të përdoren
    - Instruksionet me gjatësi fikse dekodohen më lehtë, por shfrytëzojnë më shumë hapësirë të memories

Formatet e instruksioneve

- Numrin e operandëve
- Numrin e regjistrave që mund të adresohen
- Organizimin e memories
  - E adresueshme në bajtë ose fjalë
- Regjimet e adresimit
  - Një ose më shumë: direkt, indirekt ose i indeksuar

Formatet e instruksioneve

- Rënditja e bajtëve ose "endianimi" është një element i rëndësishëm i arkitekturës
- Nëse e kemi një numër të plotë 2-bajtësh, ai mund të ruhet duke filluar nga bajti i djathtë (më pak i rëndësishëm) tek bajti i majtë (më i rëndësishëm) ose e kundërta
  - Tek makinat e tipit "little endian" shënimi fillon nga bajti më pak i rëndësishëm
  - Tek makinat e tipit "big endian" shënimi fillon nga bajti më i rëndësishëm

Formatet e instruksioneve

- Shohim numrin heksdecimal 0x12345678 dhe ruajtjen e tij në dy tipet e makinave

Address →	00	01	10	11
Big Endian	12	34	56	78
Little Endian	78	56	34	12

Formatet e instruksioneve

- Shembull tjetër:
  - Kompjuteri përdor integer 32-bitësh
  - Vlerat
    - 0xABCD1234
    - 0x00FE4321
    - 0x10

  duhet të ruhen në memorje njëra pas tjetrës duke filluar prej adresës 0x200

Address	Big Endian	Little Endian
0x200	AB	34
0x201	CD	12
0x202	12	CD
0x203	34	AB
0x204	00	21
0x205	FE	43
0x206	43	FE
0x207	21	00
0x208	00	10
0x209	00	00
0x20A	00	00
0x20B	10	00

Formatet e instruksioneve

- Arkitektura "Big endian":
  - Është më e natyrshme
  - Parashenja e numrit mund të përcaktohet nga bajti në adresën 0
  - Numrat e plotë dhe vargjet e simboleve (strings) ruhen me renditje të njëjtë
- Arkitektura "Little endian":
  - Lejojnë manipulimin me adresa çifte dhe teke, edhe në ato raste kur fjala është e gjatësisë 2, 4, etj…
  - Konvertimi nga adresa 16-bitëshë në adresë 32-bitëshe apo 64-bitëshe nuk kërkon ndonjë llogaritje

Formatet e instruksioneve

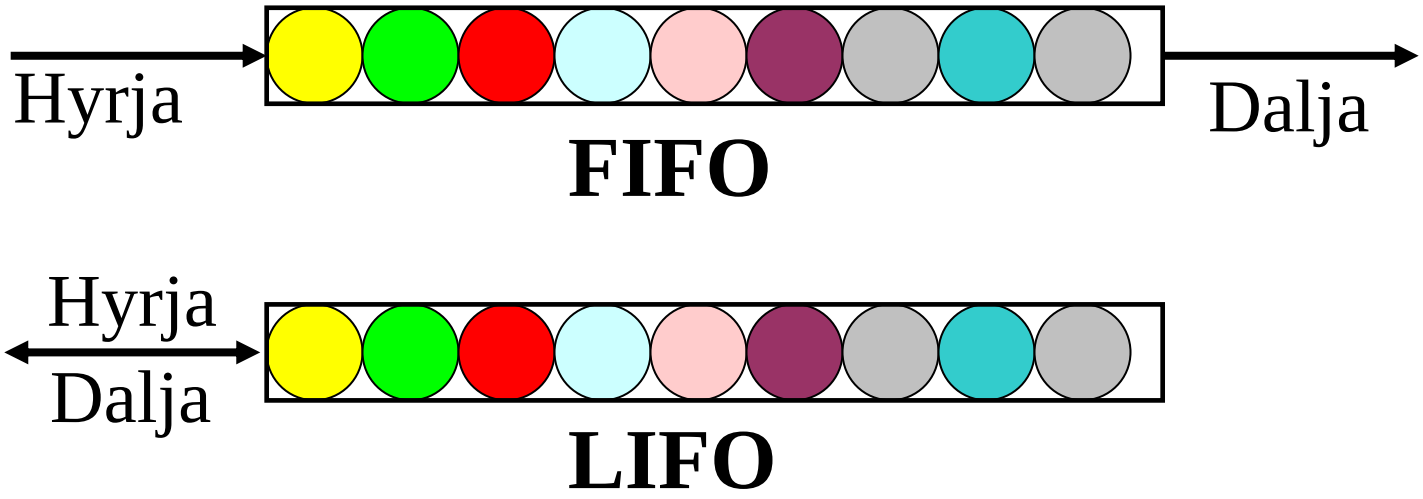
- Arkitektura "Big endian" janë:
  - Procesorët MOTOROLA
  - Rrjetat kompjuterike
  - Formatet BMP, Adobe Photoshop, JPEG
- Arkitektura "Little endian":
  - Procesorët INTEL
  - Formatet GIF, PC Paintbrush, RTF
- Arkitektura duale:
  - Procesorët ARM
  - AVI, TIFF, WAV

Formatet e instruksioneve

- Problemi tjetër që duhet të zgjidhet me rastin e dizajnimit të një arkitekture është se si i ruan të dhënat CPU
- Kemi tri zgjedhje:
  - 1. Arkitektura pirg (stack)
  - 2. Arkitektura e bazuar në akumulator
  - 3. Arkitektura e bazuar në regjistra me dedikim të përgjithshëm (GPR)
- Gjatë përcaktimit duhet t'i kushtohet rëndësi thjeshtësisë dhe çmimit që duhet paguar për dizajnimin e hardverit, me shpejtësinë e ekzekutimit dhe lehtësinë e përdorimit

Formatet e instruksioneve

- Në arkitekturat pirg instruksionet dhe operandët merren në mënyrë implicite nga maja e pirgut (stackut)
  - Pirgut nuk mund t'i qasemi në mënyrë të rastit.



Formatet e instruksioneve

- Në arkitekturat e bazuara në akumulator një operand i një veprimi binar ndodhet gjithnjë në akumulator
  - Operandi tjetër është në memorie dhe shkakton trafik të dendur në magjistrale
- Te arkitekturat GPR regjistrat mund të përdoren në vend të memories
  - Janë më të shpejta se arkitekturat e bazuara në akumulator
  - Kompilatorët punojnë lehtë me to
  - Instruksionet janë më të gjata, sepse të gjithë operandët duhet të emërtohen

Formatet e instruksioneve

- Shumica e sistemeve aktuale janë GPR
- Ekzistojnë tri tipe të GPR:
  - Memorie-memorie, ku dy ose tre operandë mund të jenë në memorie. (Shembull: DE VAX)
  - Regjistër memorie, ku të paktën një operand duhet të jetë në regjistër. (Shembuj: Intel, Motorola)
  - Load-store, ku asnjë operand nuk mund të jetë në memorie, por ata duhet të vendosen paraprakisht në regjistra. (Shembuj: ARM, MIPS, PowerPC)
- Numri i operandëve dhe numri i regjistrave në dispozicion kanë ndikim të drejtëpërdrejtë në gjatësinë e instruksionit

Formatet e instruksioneve

- Mënyra tradicionale e përshkrimit të një arkitekture kompjuterike është të specifikohet numri maksimal i operandëve ose adresave për çdo instruksion
- Kjo ndikon drejtpërdrejt në gjatësinë e vet instruksionit
- MARIE përdorë instruksione fikse 16-bitëshe me op-code 4-bitësh dhe operand 12-bitësh
- Instruksionet mund të formatohen në dy mënyra:
  - Me gjatësi fikse (zënë më shumë hapësirë, por dekodohen më lehtë)
  - Me gjatësi variabile (kursejnë hapësirën, por dekodohen më vështirë)

Formatet e instruksioneve

- Aktualisht përdoren 2-3 gjatësi të ndryshme të instruksioneve, të cilat mund të dallohen lehtë
- Nëse gjatësia e instruksionit përputhet me gjatësinë e fjalës në makinë, atëherë kjo është situata më e volitshme
- Ndërkaq, instruksionet me gjatësi sa dyfishi, një e katërta, gjysma ose trefishi i gjatësisë së fjalës rezultojnë me shfrytëzim joracional të hapësirës

Formatet e instruksioneve

- Formatet e zakonshme të instruksioneve:
  - OPCODE (pa adresë)
  - OPCODE +1 adresë (zakonisht adresa e memories)
  - OPCODE +2 adresa (zakonisht regjistra , ose një regjistër dhe nje qelizë e memories)
  - OPCODE +3 adresa (zakonisht regjistra , ose kombinime të regjistrave dhe memories)

Formatet e instruksioneve

- Makinat pirg përdorin instruksione me 0 ose 1 operandë
- Instruksionet LOAD dhe STORE kanë vetëm nga një operand që është një adresë e memories
- Instruksionet tjera përdorin operandë nga pirgu
- Për këto arkitektura tipike janë dy instruksione:
  - Push X që vendos përmbajtjen e adresës së memories X në maje të pirgut
  - Pop X që e heq elementin nga maja e pirgut dhe e vendos në adresën X
- Instruksionet binare si ADD dhe MULT përdorin si operandë dy elementet në maje të pirgut

Formatet e instruksioneve

- Arkitekturat "pirg" kërkojnë t'i vëzhgojmë shprehjet aritmetikore pak më ndryshe
- Ne jemi mësuar që veprimin e mbledhjes ta shprehim me "infix" Z = X + Y
- Tek aritmetika e pirgut duhet përvetësuar shënimin "postfix" Z = XY+ (RPN – Reverse Polish Notation)
- Disa arkitektura përdorin shënimin "prefix" Z=+XY

Formatet e instruksioneve

- Një përparësi e postfix-it është se nuk përdoren kllapat
- P.sh. në infix shënojmë:

$$Z = (X \times Y) + (W \times U),$$

ndërsa në postfix:

$$Z = X\ Y \times W\ U \times +$$

Formatet e instruksioneve

- Në ISA të tipit "stack" shprehja postfix:

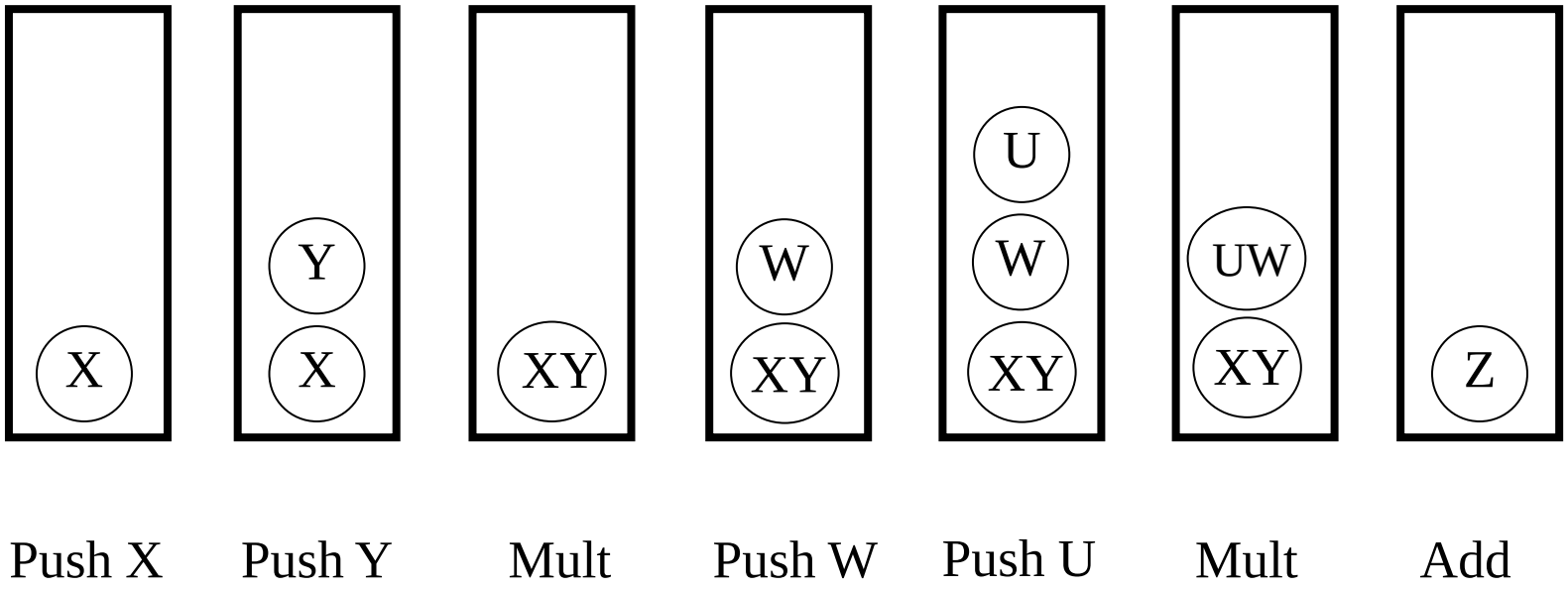
$$Z = X\ Y \times W\ U \times +$$

merr trajtën:

```
PUSH X
PUSH Y
MULT
PUSH W
PUSH U
MULT
ADD
STORE Z
```

Supozojmë se veprimet binare ekzekutojnë nga një POP në dy elementet në maje të pirgut, e kryejnë veprimin, ndërsa rezultatin e kthejnë sërish në pirg me një PUSH.

Formatet e instruksioneve



Kështu funksionon arkitektura "pirg" (LIFO).

Formatet e instruksioneve

- Në ISA me një adresë, si MARIE, instruksioni infix

$$Z = X \times Y + W \times U$$

duket kështu:

```
LOAD X
MULT Y
STORE TEMP
LOAD W
MULT U
ADD TEMP
STORE Z
```

Formatet e instruksioneve

- Në ISA me dy adresa (si Intel ose Motorola), shprehja infix

$$Z = X \times Y + W \times U$$

mund të duket kështu:

```
LOAD R1,X
MULT R1,Y
LOAD R2,W
MULT R2,U
ADD  R1,R2
STORE Z,R1
```

Formatet e instruksioneve

- Në ISA me tri adresa, shprehja infix:

$$Z = X \times Y + W \times U$$

duket kështu:

```
MULT R1,X,Y
MULT R2,W,U
ADD  Z,R1,R2
```

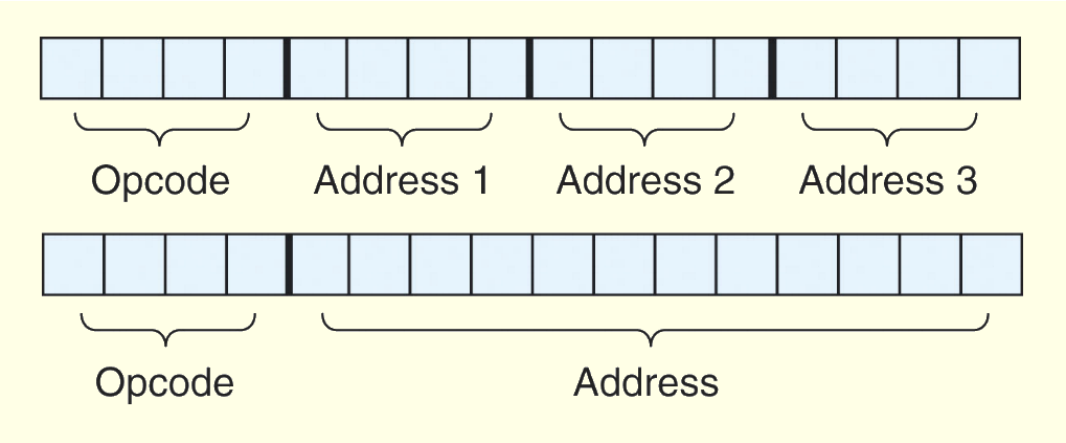
Pra, me shtimin e numrit të operandëve zvogëlohet numri i instruksioneve të nevojshmë për ekzekutimin e programeve!

Formatet e instruksioneve

- Kemi parë varshmërinë e numrit të operandëve dhe gjatësisë së instruksionit
- Jo të gjitha instruksionet në një ISA kanë numër të njëjtë operandësh
- Instruksionet pa operandë (si HALT) patjetër do të humbin pak hapësirë, nëse punohet me instruksione të gjatësisë fikse
- Një mënyrë për shfrytëzimin racional të kësaj hapësire është zgjerimi i op-codeve

Formatet e instruksioneve

- Sistemi ka 16 regjistra dhe 4K memorie
- Nevojiten 4 bit për t'iu qasur një regjistri dhe 12 bit për t'iu qasur memories.
- Nëse instruksionet do të jenë 16-bitëshe, i kemi dy zgjedhje të mundshme:

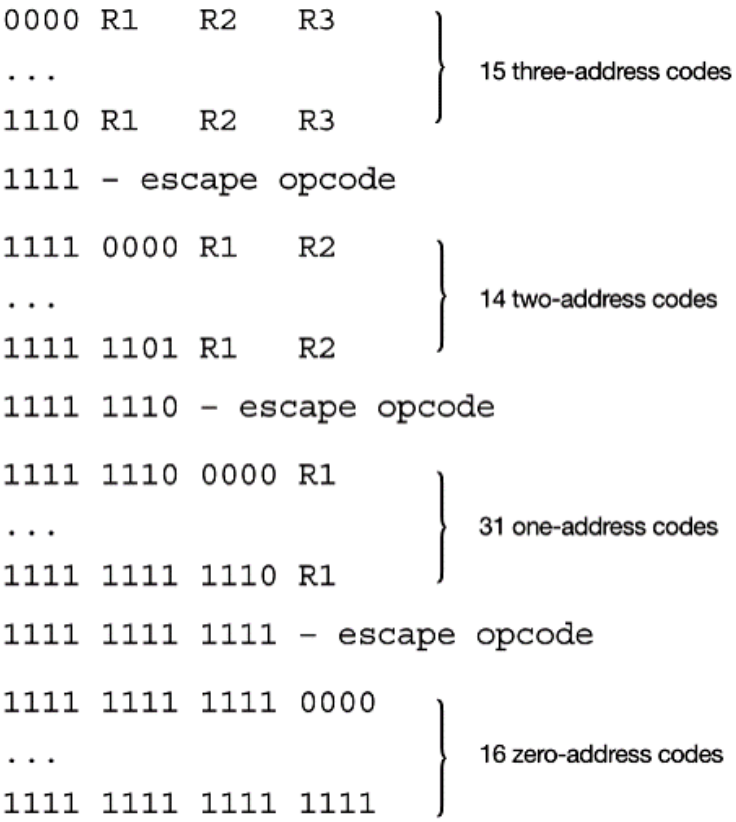


Formatet e instruksioneve

- Supozojmë se dëshirojmë të kemi:
  - 15 instruksione me 3 adresa
  - 14 instruksione me 2 adresa
  - 31 instruksione me 1 adresë
  - 16 instruksione me 0 adresa
- Për këtë qëllim përdorim një skemë për zgjërimin e op-codeve.

5.2 Formatet e instruksioneve

- Nëse lejojmë që gjatësia e op-codeve të jetë variabile do të fitojmë një ISA mjaft të pasur



Formatet e instruksioneve

- Dekodimi tani është relativisht kompleks:

if (leftmost four bits != 1111){

Execute appropriate 3-address instruction}

else if (leftmost seven bits != 1111 111){

Execute appropriate 2-address instruction}

else if (leftmost twelve bits != 1111 1111 1111){

Execute appropriate 1-address instruction}

else {

Execute appropriate 0-adress instruction}

Tipet e instruksioneve

- Instruksionet e makinës klasifikohen në disa kategori:
  - Transferi i të dhënave
  - Instruksionet aritmetikore
  - Instruksionet logjike
  - Manipulimi me bitë
  - I/O
  - Instruksionet kontrolluese
  - Instruksionet speciale (puna me stringje, cache, etj.)

Adresimi

- Teknikisht, adresimi është pjesë e formatit të instruksionit
- Regjimet (mode) e adresimit përcaktojnë se ku ndodhet operandi
- Ato mund të paraqesin konstanta, regjistra ose lokacione të memories
- Lokacioni aktual i një operandi quhet adresa efektive e operandit
- Disa regjime të adresimit na mundësojnë ta përcaktojmë adresën e një operandi në mënyrë dinamike

Adresimi

- Adresimi i menjëhershëm është atëherë kur të dhënat janë pjesë e instruksionit
  - P.sh. tek ky lloj adresimi në LOAD 008, pjesa 008 nuk është adresa e memories, por vetë vlera që do të vendoset në akumulator
  - Ky lloj adresimi është shumë i shpejtë, por jo edhe shumë fleksibil
- Adresimi direkt është kur instruksioni përmban në vete adresën e memories ku ndodhet e dhëna
  - Këtu 008 është adresa e memories, ndërsa çkado që përmban ajo vendoset në akumulator
- Adresimi i regjistrit është kur operandi ndodhet në regjistër në vend se në memorie

Adresimi

- Adresimi indirekt jep adresën e adresës ku ndodhet e dhëna e kërkuar
  - Nëse në adresën 008 ndodhet vlera 3F2, ndërsa në adresën 3F2 ndodhet vlera C31, atëherë LOAD 008 vendos në akumulator vlerën C31
- Ngjashëm intepretohet adresimi indirekt i regjistrit

Adresimi

- Tek adresimi indeksor përdoret një regjistër i veçantë (regjistri i indeksit), i cili ruan vlerën që i shtohet operandit, duke dhënë adresën efektive të të dhënës
  - Nëse operandi është X, ndërsa në regjistrin indeksor kemi vlerën 003F, adresa efektive është X+003F
- I ngjashëm është adresimi bazik, i cili në vend të regjistrit indeksor përdor të ashtuquajturin regjistër bazik

Adresimi

- Në rastin e adresimit "pirg" supozohet se operandi është në maje te pirgut
- Ekzistojnë shumë variacione te këtyre regjimeve të adresimit:
  - Indeksim indirekt
  - Bazik/offset
  - Auto rritës-zvogëlues, etj
- Këto nuk do të përpunohen në hollësi

"Gypimi" në nivel instruksioni

- Disa procesorë e zbërthejnë ciklin lexo-dekodo-ekzekuto në hapa më të vegjël
- Këto hapa shpesh mund të ekzekutohen paralelisht duke e shtuar efektivitetin e punës
- Ky lloj i ekzekutimit paralel quhet instruction-level pipelining (gypim në nivel instruksioni)
- Shkurtimisht: ILP

"Gypimi" në nivel instruksioni

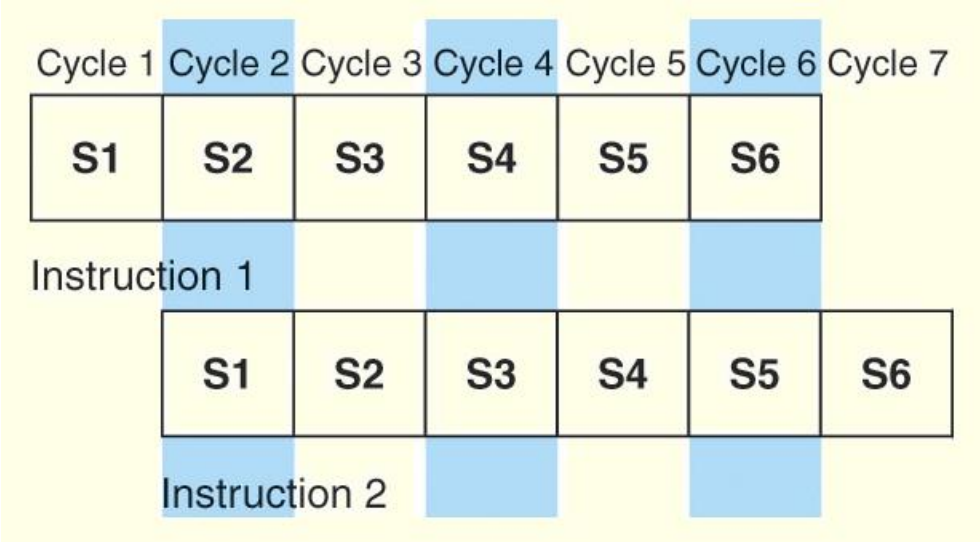
- Supozojmë se cikli lexo-dekodo-ekzekuto zbërthehet në këta hapa më të vegjël:

  1. *Lexo instruksionin*
  2. *Dekodo op-kodin*
  3. *Kalkulo adresën efektive të operandëve*
  4. *Lexo operandët*
  5. *Ekzekuto instruksionin*
  6. *Ruaj rezultatin*

- Supozojmë se e kemi gypin (pipeline) me gjashtë faza.
- S1 lexon instruksionin, S2 e dekodon, S3 përcakton adresën e operandëve, S4 lexon operandët, S5 i ekzekuton, ndërsa S6 e ruan rezultatin

"Gypimi" në nivel instruksioni

- Në çdo cikël të orës kryhet nga një hap i vogël dhe fazat përputhen..



S1. Lexo instruksionin
S2. Dekodo op-code
S3. Kalkulo adresën efektive të operandëve
S4. Lexo operandët
S5. Ekzekuto instruksionin
S6. Ruaj rezultatin

"Gypimi" në nivel instruksioni

- Shpejtimi teorik (speed up) mund të përcaktohet si vijon:
  - Le të jetë t_p koha e nevojshme për kryerjen e një faze. Çdo instruksion paraqet një detyrë, T, në gyp.
  - Për kryerjen e instruksionit të parë në gypin me k-faza nevojitet koha prej $k \times t_p$. Të (n - 1) detyrat e mbetura dalin nga gypi për çdo cikël. Pra, koha e nevojshme për kryerjen e detyrave të mbetura është $(n - 1)t_p$
  - D.m.th. kryerja e n detyrave në gyp me k-faza kërkon:

$$(k \times t_p) + (n - 1)t_p = (k + n - 1)tp$$

"Gypimi" në nivel instruksioni

- Tani marrim kohën e nevojshmë për t'i kryer të n detyrat pa gypim dhe e ndajmë me kohën e nevojshme kur përdoret gypimi:

$$\text{Speedup } S = \frac{n t_n}{(k + n - 1)\, t_p}$$

- Për $n \to \infty$ kemi:

$$\text{Speedup } S = \frac{k t_p}{t_p} = k$$

"Gypimi" në nivel instruksioni

- Këto ekuacione nuk reflektojnë plotësisht realitetin
- Së pari, duhet të supozojmë se arkitektura është në gjendje t'i lexojë paralelisht instruksionet dhe të dhënat
- E dyta, duhet supozuar se gypi do të jetë e mbushur tërë kohën, gjë që nuk ndodhë gjithmonë
- Në linja të tilla shpesh mund të shkaktohen konflikte rreth resurseve

"Gypimi" në nivel instruksioni

- Gypimi mund të dështojë ose të ketë vonesa për cilëndo nga këto arsye:
  - Konfliktet e resurseve
  - Varshmëria e të dhënave
  - Degëzimi me kusht
- Mund të merren masa si në nivel të softverit, ashtu edhe në nivel të hardverit që të reduktohen efektet e këtyre rreziqeve, por ato nuk mund të eliminohen plotësisht

Shembuj të ISA-ve

- Intel
  - Little endian
  - Me dy adresa
  - Instruksionet me gjatësi variabile
  - Arkitekturë e tipit GPR regjistër-memorie
  - Modelet 8086, 8088, 80286, 80386 dhe 80486 janë serike
  - Pentium I ka pasur dy linja (pipeline), U dhe V, me nga 5 faza:
    - Prefetch (paralexim)
    - Dekodimi i instruksionit
    - Gjenerimi i adresës
    - Ekzekutimi
    - Ruajtja (Write Back)

Shembuj të ISA-ve

- Intel
  - Pentium II: 12 faza
  - Pentium III: 14 faza
  - Pentium IV: 24 faza
  - Itanium (procesor i familjes RISC): 10 faza

Shembuj të ISA-ve

- Procesorët e Intelit përkrahin regjime të llojllojshme të adresimit
- Intel 8086 ofronte 17 mënyra për ta adresuar memorien, shumica prej tyre variante të regjimeve të prezantuara këtu
- Edhe procesorët Pentium i përkrahin këto 17 regjime, por aplikojnë edhe regjime të reja adresimi
- Itanium përkrah vetëm një formë te adresimit (një lloj të adresimit indirekt)

Shembuj të ISA-ve

- MIPS është shkurtesë për Microprocessor Without Interlocked Pipeline Stages
- Arkitektura është e tipit "little endian", e adresueshme sipas fjalëve, me tri adresa dhe instruksione të gjatësisë fikse
- Edhe këtu numri i fazave në gyp ka shkuar duke u rritur: në R2000 dhe R3000 i kemi nga 5 faza, ndërsa në R4000 dhe R4400 nga 8 faza

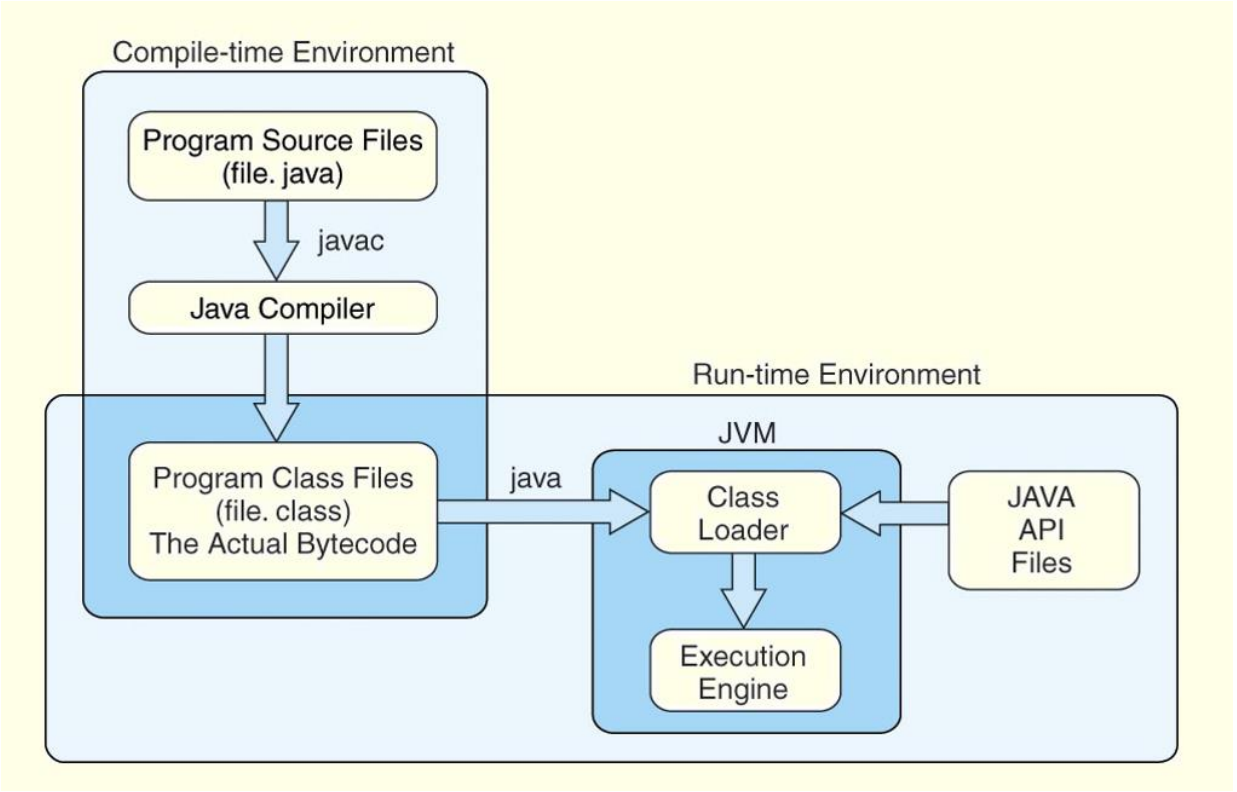
Shembuj të ISA-ve

- Procesori R10000 ka tri gypa: Një 5-fazëshe për veprime me numra të plotë, një 7-fazëshe për veprime me presje dhjetore, dhe një 6-fazëshe për instruksionet LOAD/STORE
- Tek MIPS vetëm instruksionet LOAD dhe STORE mund t'i qasen memories
- Përdoret vetëm regjimi bazik i adresimit
- Ndërkaq, asembleri i këtyre procesorëve lejon edhe disa regjime të tjera të adresimit

Shembuj të ISA-ve

- Gjuha programuese Java është gjuhë interpretuese e cila ekzekutohet në makinë softverike të quajtur Java Virtual Machine (JVM)
- JVM shkruhet në gjuhën amtare të procesorëve të ndryshëm përfshirë MIPS dhe Intel
- Si një makinë reale, JVM ka një ISA të saj që quhet bytecode
- Kjo ISA është dizajnuar të jetë kompatibile me cilëndo makinë ku ekzekutohet JVM

Shembuj të ISA-ve



Shembuj të ISA-ve

- Java bytecode është gjuhë e bazuar në pirg
- Shumica e udhëzimeve janë me 0 adresë
- JVM ka 4 regjistra të cilët sigurojnë qasje në 5 regjionet e memories kryesore
- Të gjitha referimet drejt memorjes janë offsetë të këtyre regjistrave
- Java nuk përdorë pointerë apo referenca absolute
- Java është dizajnuar për përshtatje ndërmjet platformave, jo për performancë!

Shembuj të ISA-ve

- Ndoshta nuk keni dëgjuar për procesorët ARM por gjasat janë që përdorni processor ARM çdo ditë
- Është arkitektura më e përdorur me instruksione 32-bitëshe:
  - Mbi 95% e telefonëve të mençur (smartphones)
  - Mbi 80% e kamerave digjitale
  - Mbi 40% e të gjithë televizorëve digjitalë
- E krijuar në 1990, nga Apple dhe të tjerë, ARM (Advanced RISC Machine) tash është firmë Britanike, ARM Holdings
- ARM Holdings nuk i prodhon këta procesorë; ajo i shet licencat për prodhim
- Të gjithë procesorët e iPhonëve janë të arkitekturës ARM
- Procesori M1 është ARM procesori më i ri (2020) për përdorim në Macbook

Shembuj të ISA-ve

- ARM është arkitekturë load/store: I gjithë procesimi i të dhënave duhet të kryhet në regjistra, jo në memorje.
- Përdor udhëzime me gjatësi fikse me tre operandë dhe regjim të thjeshtë adresimi
- Procesorët ARM kanë minimum gypime me tri nivele (fetch, decode, and execute)
  - Procesorët më të rinjë ARM kanë më shumë nivele

Shembuj të ISA-ve

- ARM ka 37 regjistra në total por dukshmëria e tyre mvaret nga regjimi i procesorit
- ARM lejon transferë të shumtë të regjistrave
  - Mundet njëkohësisht të lexojë apo ruajë (load apo store) cilindo subset të 16 regjistrave të përgjithshëm prej/në adresa memorike sekuenciale
- Udhëzimet e kontrollës së rrjedhjes (Control flow instructions) përfshijnë degëzim me dhe pa kusht si dhe thirrje procedurale
- Shumica e udhëzimeve të ARM ekzekutohen në një cikël të vetëm, me kusht që nuk ka pengesa në gyp apo qasje në memorje

Përmbledhje

- ISA-t dallohen përkah numri i biteve për instruksion, numri i operandëve për instruksion, lokacioni i operandëve, si dhe përkah tipi dhe madhësia e operandëve
- "Endianimi" është poashtu me rëndësi për shqyrtimin e arkitekturës
- CPU mund të ruajë të dhëna bazuar në:
  - Arkitekturën "pirg"
  - Arkitekturën në bazë akumulatori
  - Arkitekturën në bazë te regjistrave të përgjithshëm

Përmbledhje

- Instruksionet mund të jenë të gjatësisë fikse ose variabile
- Për ta pasuruar bashkësinë e instruksioneve me gjatësi fikse, op-codet mund të zgjerohen
- Regjimi i adresimit është poashtu me rëndësi për ISA-t. Llojet e adresimit:
  - I menjëhershëm
  - Direkt
  - Regjistër
  - Regjistër Indirekt
  - Indirekt
  - I indeksuar
  - Bazik
  - "Pirg"

Përmbledhje

- Pipeline k-fazësh teorikisht mund të prodhojë shpejtim në ekzekutim për k herë në krahasim me makinat serike
- Konfliktet e resurseve dhe degëzimi me kusht janë ato që pengojnë realizimin e shpejtimeve të tilla në praktikë
- Arkitekturat Intel, MIPS, JVM dhe ARM janë shembuj të mirë të koncepteve të prezantuara në këtë kapitull

Pyetje???