

Manipulimi me të dhëna

Qëllimet dhe objektivat

- Si një kompjuter manipulon të dhëna dhe si komunikon me paisje periferike, si printerë dhe tastiera
- Eksplorimi i bazave të arkitekturës kompjuterike
- Si kompjuterët programohen me anë të instrukcioneve të koduara, të quajtura instrukcione në gjuhë të makinës

Përmbajtja

- 1 Arkitektura kompjuterike
 - Baza të CPU
 - Nocioni i programit të ruajtur
- 2 Gjuha e makinës
 - Repertoari i instrukcioneve
 - Një gjuhë ilustruese e makinës
- 3 Ekzekutimi i programit
 - Shembull ekzekutimi i një programi
- 4 Instrukcionet aritmetike/logjike
 - Operacionet logjike
 - Operacionet e rotacionit ose zhvendosjes
 - Operacionet aritmetike
- 5 Komunikimi me paisje tjera
 - Roli i kontrollerëve
 - Direct memory access
 - Shpejtësitë e komunikimit

Arkitektura kompjuterike

- *Njësi qendrore përpunuese (procesor) (central processing unit, CPU)*: Qarku i integruar në një kompjuter që kontrollon manipulimin me të dhëna
- CPU të PC-ve të sotëm janë pllaka me sipërfaqe disa cm², pinat e të cilave futen në një prizë të montuar në pllakën kryesore elektronike të makinës (të quajtur *pllaka amë, motherboard*.
 - Për shkak të madhësisë, këta procesorë shpesh quhen *mikroprocesorë*.
 - Poashtu, edhe qarqet e memories qendrore montohen më pllakën amë ngjashëm sikur procesori.

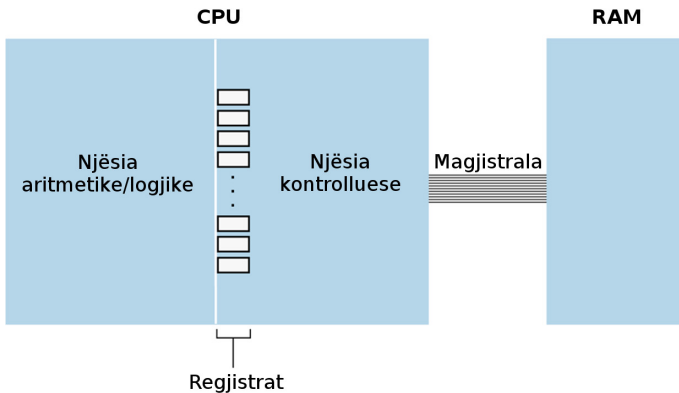
Baza të CPU

- Një CPU përbëhet nga tri pjesë:
 - *njësia aritmetike/logjike*
 - përmban qarqe që performojnë operacione mbi të dhëna
 - *njësia kontrolluese*
 - përmban qarqe për koordinimin e aktiviteteve të makinës
 - *njësia e regjistrave*
 - përmban qeliza për ruajtje të dhënash (të ngjashme me qelizat e memories qendrore), të quajtura *regjistra*, që përdoren për ruajtje të përkohshme të dhënash përbrenda CPU.
- Disa nga regjistrat brenda njësisë së regjistrave konsiderohen *regjistra me destinim të përgjithshëm*, ndërsa të tjerë janë *regjistra me destinim special*.
 - Në këtë pikë do të shqyrtojmë vetëm regjistrat me destinim të përgjithshëm

Baza të CPU (Vazhdim)

- Regjistrat me destinim të përgjithshëm mbajnë inputin e njësisë aritmetike/logjike dhe ofrojnë hapësirë ruajtjeje për rezultatet e prodhuara nga ajo njësi.
- Me qëllim transferimi të stringjeve të bitëve, CPU dhe memoria qendrore e një makine janë të lidhura me një koleksion telash të quajtur *magjistralë të dhënash* (*data bus*).

CPU dhe RAM



Figurë: CPU dhe RAM të lidhura me magjistralë

Shembull ekzekutimi i një instrukcioni

Algoritëm 1 Mbledhja e vlerave të ruajtura në memorie

- 1: Merr njërën nga vlerat për t'u mbledhur nga memoria dhe vendose në një regjistër.
 - 2: Merr vlerën tjetër për t'u mbledhur nga memoria dhe vendose në një regjistër tjetër.
 - 3: Aktivizo qarkun për mbledhje me regjistrat e përdorur në Hapat 1 dhe 2 si hyrje dhe një regjistër tjetër të disenjuar për të mbajtur rezultatin.
 - 4: Ruaj rezultatin në memorie.
 - 5: Ndalo.
-

Nocioni i programit të ruajtur

- Një zbulim i madh në shkencën kompjuterike ishte ai i John von Neumann mbi *nocionin e programit të ruajtur në memorie*.
 - Të kuptuarit se një program, posikurse të dhënat, mund të kodohet dhe të ruhet në memorie qendrore
- Në qoftë se njësia kontrolluese disenjohet ashtu që të ekstraktojë programin nga memoria, të dekodojë instrukcionet dhe t'i ekzekutojë ato, atëherë programi që ekzekuton makina mund të ndryshohet thjesht duke ndryshuar përmbajtjen e memories së kompjuterit në vend se të rilidhen telat e CPU.

Nocioni i programit të ruajtur (Vazhdim)

- Nocioni i programit të ruajtur në memorie është standard i kompjuterëve të sotëm.
 - Vështirësitë ishin rrjedhojë e perceptimit të të dhënave dhe programit si entitete të ndryshme: Të dhënat ruheshin në memorie, programet ishin pjesë e CPU.
 - Shembull i „mosparjes së pyllit prapa pemëve“
 - Mbase shkenca kompjuterike sot, shkenca përgjithësisht dhe praktika janë përplot me pengesa të tilla.

Gjuha e makinës

- Për të zbatuar nocionin e programit të ruajtur në memorie CPU-të janë të disanjuara të interpretojnë instrukcione të koduara si stringje bitësh.
- *Gjuha e makinës*: Koleksioni i instrukcioneve së bashku me sistemin e kodimit
- *Instrukcion i makinës*: Një instrukcion i shprehur në këtë gjuhë

Repertoari i instrukcioneve

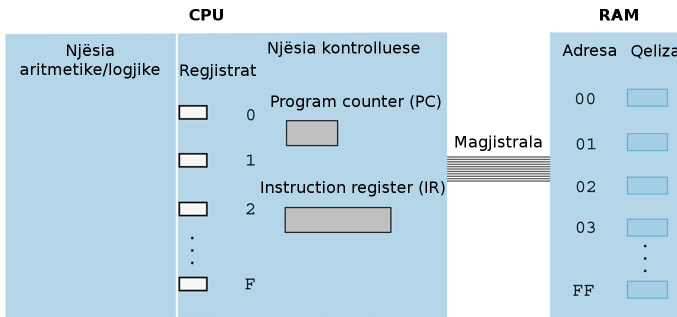
- Lista e instrukcioneve të makinës që mund të ekzekutojë një CPU është mjaft e shkurtër.
 - *RISC (reduced instruction set computer)* procesorët: dikur PowerPC, sot paisjet mobile
 - *CISC (complex instruction set computer)* procesorët: dikur PC, sot të gjithë dekstop kompjuterët
- Instrukcionet e makinës kategorizohen në tri grupe:
 - 1 grupi i transferit të të dhënave: si LOAD, STORE, *I/O instrukcionet*
 - 2 grupi aritmetik/logjik: si AND, OR, XOR, SHIFT, ROTATE, ADD, SUB, MUL, DIV
 - 3 grupi kontrollues: si JUMP
 - *kërcime të pakushtëzuara*
 - *kërcime të kushtëzuara*

Shembull ekzekutimi i një instrukcioni

Algoritëm 2 Pjesëtimi i vlerave të ruajtura në memorie

- 1: Ngarko (LOAD) një regjistrë me një vlerë nga memoria.
 - 2: Ngarko (LOAD) një regjistrë tjetër me një vlerë tjetër nga memoria.
 - 3: Në qoftë se vlera e dytë është 0, kalo (JUMP) në hapin 6.
 - 4: Pjesëto (DIV) përmbajtjet e regjistrit të parë me të regjistrit të dytë dhe vëj rezultatin në një regjistrë të tretë.
 - 5: Ruaj (STORE) vlerën e regjistrit të tretë në memorie.
 - 6: Ndalo (STOP).
-

Një gjuhë ilustruese e makinës

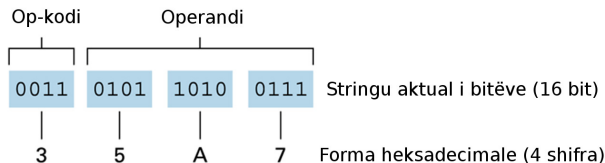


Figurë: Arkitektura e makinës së ilustruar në shtojcën

Kompozimi i një instrukcioni të makinës

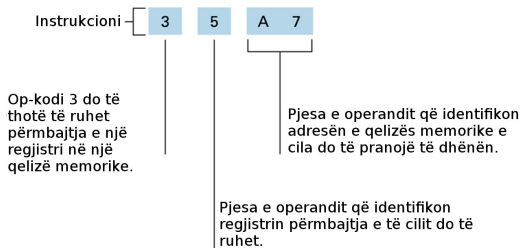
- Versioni i koduar i një instrukcioni të makinës përbëhet nga dy pjesë:
 - fusha e *op-kodit* (*op-code*): Specifikon cili operacion elementar të ekzekutohet
 - fusha e *operandit* (*operand*): Jep informacion më të detajizuar mbi operacionin e specifikuar me op-kodin
 - Interpretimi i operandit ndryshon varësisht nga op-kodi.

Kompozimi i një instruksioni të makinës (Vazhdim)



Figurë: Kompozimi i një instruksioni të makinës në shtojcën

Dekodimi i një instrukcioni të makinës



Figurë: Dekodimi i instrukcionit 35A7

- STORE stringun e bitëve nga regjistri 5 në qelizën memorike me adresë A7
- Vëreni se si shfrytëzimi i notacionit heksadecimal 35A7 është më intuitiv për programuesin: STORE 5 A7.
- Vëreni se instrukcioni në gjuhën e makinës është 0011010110100111.

Versioni i koduar i algoritmit të mbledhjes

- Supozojmë se vlerat për t'u mbledhur janë ruajtur në notacionin e komplementit të dyshes në adresat memorike 6C dhe 6D dhe shuma duhet të vëhet në qelizën memorike me adresë 6E.

Algoritëm 3 Versioni mnemonik i mbledhjes së vlerave të ruajtura në memorie

- 1: LOAD 5 6C
 - 2: LOAD 6 6D
 - 3: ADD 0 5 6
 - 4: STORE 0 6E
 - 5: STOP
-

Versioni i koduar i algoritmit të mbledhjes (Vazhdim)

Algoritëm 4 Versioni i koduar në gjuhën e makinës nga shtojca i mbledhjes së vlerave të ruajtura në memorie

156C

166D

5056

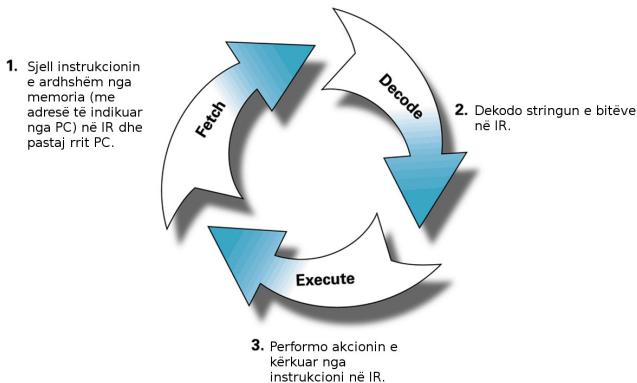
306E

C000

Ekzekutimi i programit

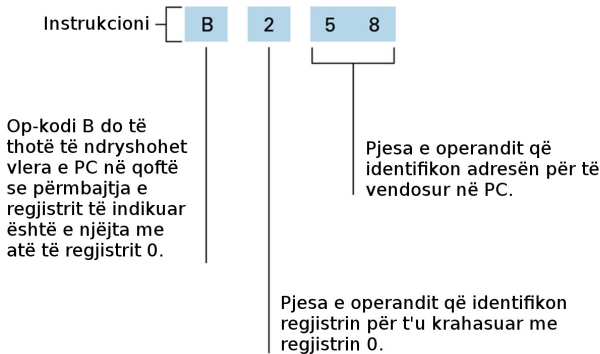
- Ekzekutimi i programit kontrollohet nga dy regjistra me destinim special në CPU:
 - *Numëruesi i programit (program counter, PC):*
 - Adresa e instrukcionit të ardhshëm
 - *Regjistri i instrukcioneve (instruction register, IR):*.
 - Instrukcioni aktual
- *Cikli i makinës:* Algoritmi trehapësh që përsëritet në mënyrë të përsëritur nga CPU:
 - Sjell (fetch)
 - Vëj në IR instrukcionin e ruajtur në qelizën memorike të indikuar nga PC dhe pastaj rrit PC (në shembullin tonë për 2).
 - Dekodo (decode)
 - Zbërthe fushën e operandit në komponente përkatëse mbështetur në op-kodin e instrukcionit në IR.
 - Ekzekuto (execute)
 - Aktivizo qarkun përkatës për të performuar detyrën e kërkuar.

Cikli i makinës



Figurë: Cikli i makinës si algoritëm trehapësh: sjell, dekodo, ekzekuto

Dekodimi i një JUMP instrukcioni



Figurë: Dekodimi i instrukcionit B258

Dekodimi i një JUMP instrukcioni (Vazhdim)

- Instrukcioni i makinës: B258
- Instrukcioni në notacionin mnemonik:

JUMP 2 58

- *Kërcim i kushtëzuar*
- Shqyrtoni instrukcionin e makinës: B058
- Në notacionin mnemonik:

JUMP 0 58

- *Kërcim i pakushtëzuar*

Shembull ekzekutimi i një programi

- Ndjekim ciklin e makinës të zbatuar mbi programin 4.
- Supozojmë se programi është ruajtur në RAM filluar nga adresa A0.
- Vëjmë vlerën A0 në regjistrin PC.

Shembull ekzekutimi i një programi (Vazhdim)

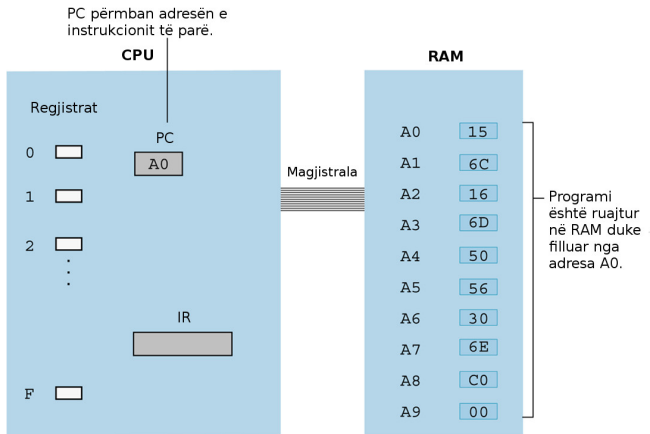
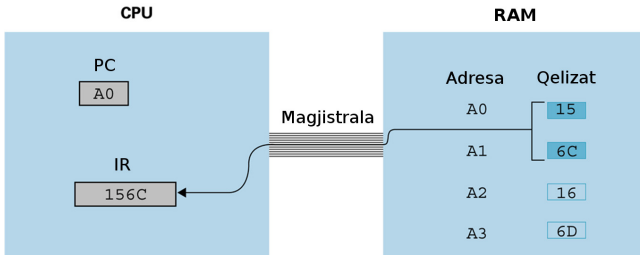


Figure: Programi i ruajtur në memorie i gatshëm për ekzekutim

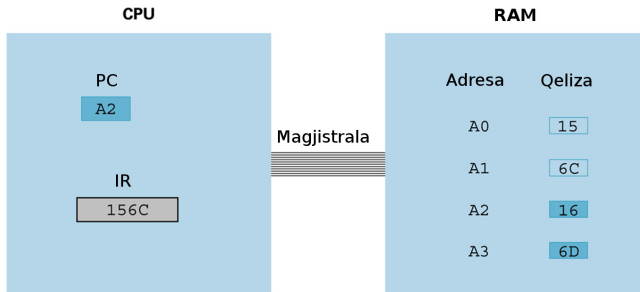
Performimi i hapit fetch të ciklit të makinës



- a. Në fillim të hapit fetch instrukcioni me fillim në adresën A0 nxjerret nga memoria dhe vëhet në IR.

Figurë: Performimi i fetch: Vendosja e instrukcionit në IR

Performimi i hapit fetch të ciklit të makinës (Vazhdim)



b. Pastaj PC rritet ashtu që të pointojë instrukcionin e ardhshëm.

Figurë: Performimi i fetch: Rritja e PC

Instrukcionet aritmetike/logjike

- Grupi i instrukcioneve aritmetike/logjike implementon:
 - operacione logjike
 - operacione rotacioni ose zhvendosjeje
 - operacione aritmetike
- Operacionet logjike:
 - AND, OR, XOR
 - Zgjerohen në operacione të cilat veprojnë mbi dy stringje bitësh dhe prodhojnë një string dalës duke zbatuar operacionet bazike përkatëse mbi bitët individualë.

- P.sh.,

	10011010
AND	11001001
<hr/>	
	10001000

Operacionet logjike: Maskimi

- **Maskimi:** Operacion ku njëri operand (*maska*) përcakton se cila pjesë e operandit tjetër do të ndikojë rezultatin.
- **Detyrë:** Vëj bitin e tretë nga skaji i rendit të lartë të një stringu 8-bitësh në 0 pa ndryshuar bitët e tjerë.
- **Zgjidhje:** Vepro në stringun e dhënë të bitëve AND me stringun 11011111
- P.sh.,

$$\begin{array}{r} 11110001 \\ \text{AND } 11011111 \\ \hline 11010001 \end{array}$$

Operacionet logjike: Maskimi (Vazhdim)

- **Detyrë:** Vëj bitin e tretë nga skaji i rendit të lartë të një stringu 8-bitësh në 1 pa ndryshuar bitët e tjerë.
- **Zgjidhje:** Vepro në stringun e dhënë të bitëve OR me stringun 00100000
- P.sh.,

$$\begin{array}{r} 11110001 \\ \text{OR } 00100000 \\ \hline 11110001 \end{array}$$

Operacionet logjike: Maskimi (Vazhdim)

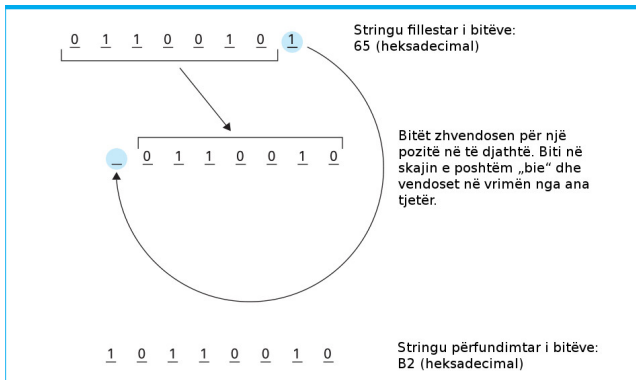
- **Detyrë:** Ndrysho bitin e tretë nga skaji i rendit të lartë të një stringu 8-bitësh pa ndryshuar bitët e tjerë.
- **Zgjidhje:** Vepro në stringun e dhënë të bitëve XOR me stringun 00100000
- P.sh.,

$$\begin{array}{r} \phantom{\text{XOR}} 11110001 \\ \text{XOR } 00100000 \\ \hline 11010001 \end{array}$$

Operacionet e rotacionit ose zhvendosjes

- Operacionet e rotacionit ose zhvendosjes:
 - *Zhvendosje cirkulare (rotacion):* ROTATE
 - *Zhvendosje logjike:* SHIFT
 - Zhvendosja për 1 bit në të majtë mund të përdoret për të shumëzuar me 2 paraqitjen në notacion binar ose në komplement të dyshes.
 - Zhvendosja për 1 bit në të djathtë mund të përdoret për të pjesëtuar me 2 paraqitjen në notacion binar ose në komplement të dyshes.
 - *Zhvendosje aritmetike:* Zhvendosje duke pasur parasysh bitin e parashenjës

Rotacioni i një stringu bitësh



Figurë: Rotacioni i stringut të bitëve 65 (heksadecimal) për 1 bit në të djathtë

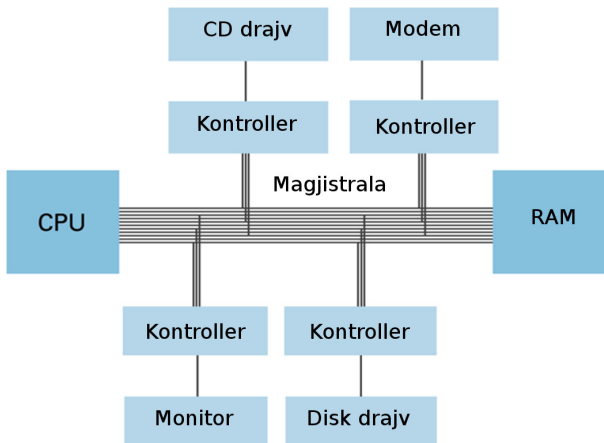
Operacionet aritmetike

- Mbledhja: ADD, zbritja: SUB, shumëzimi: MUL, pjesëtimi: DIV
 - Akcioni preciz varet nga mënyra e kodimit të vlerave (p.sh., si komplement i dyshes ose me pikë lëvizëse).
- Disa CPU disenjohen vetëm me instrukcione për mbledhje dhe zbritje.
- Operacionet aritmetike me vlera të paraqitura në aritmetikë me pikë lëvizëse, edhe në qoftë se janë të implementuara si instrukcione të CPU, kërkojnë aktivitet më madh të makinës.

Komunikimi me paisje tjera

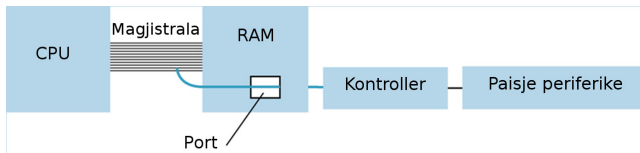
- *Kontroller*: Aparat ndërmjetës i cili kontrollon komunikimin ndërmjet kompjuterit (CPU dhe RAM) dhe një pasjeje periferike
 - Tipet e veçanta të paisjeve kanë kontrollerë të specializuar.
 - Ka edhe kontrollerë me destinim të përgjithshëm, si USB (universal serial bus), FireWire
- *Port*: Pika (konektor) në të cilën një paisje lidhet me kompjuterin
- *Hyrje/dalje e pasqyruar në memorie (Memory mapped I/O)*: CPU komunikon me paisjet periferike sikur të ishin qeliza memorike (d.m.th., sikurse që komunikon me RAM)

Kontrollerët



Figurë: Kontrollerët të lidhur me magjistralën e makinës

Memory mapped I/O



Figurë: Paraqitja konceptuale e Memory mapped I/O

Direct memory access (DMA)

- *Qasja e drejtpërdrejt memories (Direct memory access, DMA)*: Qasja në RAM nga kontrolleri përmes magjistralës.
- *Fyti i ngushtë i von Neumann*: Shpejtësia e pamjaftueshme e magjistralës pengon performancën e sistemit kompjuterik
- *Handshaking*: Procesi i koordinimit të transferit të të dhënave ndërmjet komponenteve
- *Status word*: String bitësh i cili gjenerohet nga njësia periferike dhe i dërgohet kontrollorit për të indikuar gjendjen e paisjes.

Komunikimi ndërmjet paisjeve

- *Komunikim paralel:* Disa rrugë („linja“) komunikuese transferojnë bit në mënyrë simultane (p.sh., magjistrala interne e kompjuterit, Ethernet, WiFi)
- *Komunikim serial:* Bitët transferohen njëri pas tjetrit përmes një rruge të vetme komunikimi (p.sh., USB)

Shpejtësitë e komunikimit

- Njësitë matëse:
 - Bps: bit për sekond
 - Kbps: kilo-bps (1,000 bps)
 - Mbps: mega-bps (1,000,000 bps)
 - Gbps: giga-bps (1,000,000,000 bps)
- *Gjerësi brezi (bandwidth)*: Shpejtësia maksimale në dispozicion
 - e USB: disa Mbps
 - e DSL (digital subscriber line): disa Mbps
 - e nevojshme për MP3 muzikë: 65 Kbps
 - e nevojshme për video: disa Mbps

Përfundim

- Arkitektura kompjuterike: CPU, memoria qendrore, magjistrala
- Instrukcionet e makinës
- Cikli i makinës: fetch, decode, execute
- Ekzekutimi i një programi të ruajtur në memorie në gjuhë të makinës
- Operacionet aritmetike/logjike
- Komunikimi me paisje periferike