



CS120 - ORGANIZACIJA RAČUNARA

Savremene procesorske arhitekture

Lekcija 14

PRIRUČNIK ZA STUDENTE

CS120 - ORGANIZACIJA RAČUNARA

Lekcija 14

SAVREMENE PROCESORSKE ARHITEKTURE

- ✓ Savremene procesorske arhitekture
- ✓ Poglavlje 1: Intel x64 arhitektura
- ✓ Poglavlje 2: 64-bitna ARM arhitektura
- ✓ Poglavlje 3: RISC-V arhitektura
- ✓ Poglavlje 4: Qualcomm snapdragon procesori
- ✓ Poglavlje 5: Apple M procesori
- ✓ Poglavlje 6: Pokazne vežbe
- ✓ Poglavlje 7: Zadaci za samostalni rad
- ✓ Poglavlje 8: Domaći zadatak
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

✓ Uvod

UVOD

Većina modernih ličnih računara sadrži procesor koji podržava Intel ili AMD verziju x86 32-bitne i x64 64-bitne arhitekture.

Većina modernih ličnih računara sadrži procesor koji podržava Intel ili **AMD** (engl. **Advanced Micro Devices** = Napredni mikro uređaji) verziju **x86** 32-bitne **x64** 64bitne arhitekture. Nasuprot tome, skoro svi pametni telefoni, pametni satovi, tablet računari i mnogi ugrađeni sistemi sadrže **ARM** (engl. **Advanced RISC Machines**= Napredne RISC mašine) 32-bitne ili 64-bitne procesore.

U ovoj lekciji biće predstavljen detaljan pogled na klase i skupove instrukcija ovih brendova.

U lekciji ćemo objasniti arhitekture na visokom nivou i jedinstvene atribute

x86, **x64**, 32-bitni **ARM** i 64-bitni **ARM** registar, skupovi instrukcija, montažni jezici i ključne aspekte nasleđenih karakteristika podržanih u ovim arhitekturama.

U samoj lekciji obradićemo teme:

- Intel **x64** arhitektura
- 64-bitna **ARM** arhitektura
- **RISC-V** arhitektura
- Primer: Qualcomm snapdragon procesori
- Primer: **Apple** M procesori

▼ Poglavlje 1

Intel x64 arhitektura

INTEL X64 ARHITEKTURA – OSNOVE

Pojmovi "64-bitnost" odnosno "32-bitnost" odnose se na određene karakteristike glavnoga procesora (CPU - Central Processor Unit).

64-bitno računarstvo koristi se više od 10 godina u super računarstvu i upravljanju velikim bazama podataka (DEC, IBM, Motorola...).

Mnoge kompanije i organizacije koje rade s velikom količinom podataka već koriste 64-bitne servere radi efikasnijeg baratanja velikim brojem datoteka.

U tom smislu jedan 64-bitni server može zameniti više 32-bitnih servera na mreži.

Pojmovi "64-bitnost" odnosno "32-bitnost" odnose se na određene karakteristike **glavnoga procesora CPU** - (engl. **Central Processor Unit**).

64-bitni procesori imaju **registre opšte namene GPRs** - (engl. **General PurposeRegisters**) širine 64 bita, dok 32-bitni procesori imaju opšte namenske registre širine 32 bita.

U prošlosti su personalni računari imali registre širine 16 bita (XT 8086), zatim 32 bita (80386), a danas imamo uglavnom 64-bitne registre.

Osnovne prednosti 64-bitnih procesora u odnosu na 32-bitne su:

- veća količina radne memorije koju mogu koristiti i
- bolje performanse u nekim aplikacijama.

64-bitni računari imaju spomenute prednosti zbog veće širine registara opšte namene koji mogu baratati celim brojevima do 2^{64} , što je mnogo više od 2^{32} sa koliko barataju 32-bitni računari.

64-bitna arhitektura računara obuhvata skup funkcionalnih jedinica, kao što su aritmetičko logičke jedinice ili multiplikatori, celobrojne vrednosti i memorijske adrese širine 64 bita, tj osam bajta.

64-bitni **CPU** i **ALU** su zasnovani na registrima, adresnim magistralama, ili magistralama podataka te veličine. Iz perspektive softvera, 64-bitno računarstvo podrazumeva korišćenje koda sa 64-bitnim adresama virtuelne memorije.

INTEL X64 ARHITEKTURA - SET INSTRUKCIJA

Ime arhitekture koju dele AMD i Intel se zvalo x86-64, što odražava evoluciju x86 sa 32 na 64 bita, i, u popularnoj upotrebi, ovaj termin se koristi skraćenicom x64.

Originalnu specifikaciju za arhitekturu procesora koja proširuje x86 procesor i skup instrukcija na 64 bita, nazvanu **AMD64**, predstavila je kompanija AMD 2000. godine. Prvi **AMD64** procesor, Opteron, objavljen je 2003. godine.

Kompanija Intel je podržala i nastavila da sledi AMD-ov patent i razvila je arhitekturu koja je kompatibilna sa **AMD64**, koja je na kraju dobila ime **Intel 64**. Prvi Intelov procesor koji je implementirao 64-bitnu arhitekturu bio je **Xeon**, predstavljen 2004. godine.

Prva verzija **Linux** koja podržava x64 arhitekturu objavljena je 2001. godine, mnogo pre nego što su prvi x64 procesori uopšte bili dostupni. **Windows** je počeo da podržava x64 arhitekturu 2005. godine.

Procesori koji implementiraju **AMD64** i **Intel 64** arhitekture su uglavnom kompatibilni na nivou skupa instrukcija programa u korisničkom režimu. Postoji nekoliko bezznačajnih razlika između arhitektura.

Uopšteno govoreći, operativni sistemi i kompjajleri programskih jezika upravljaju ovim razlikama, što ih čini retko pitanjem od značaja za programere softvera i sistema.

Programeri softvera, drajvera i asemblerorskog koda moraju uzeti u obzir ove razlike od kojih je najznačajnija razlika u podršci za Intel instrukcije i AMD instrukcije.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

GLAVNE KARAKTERISTIKE X64 ARHITEKTURE

x64 je uglavnom kompatibilna 64-bitna ekstenzija 32-bitne x86 arhitekture.

Glavne karakteristike x64 arhitekture su:

- x64 je uglavnom kompatibilna 64bit ekstenzija 32bitne x86 arhitekture. Većina softvera, posebno aplikacija u korisničkom režimu, napisanih za 32bitno okruženje, trebalo bi da se izvršava bez modifikacija u procesoru koji radi u 64-bitnom režimu.
- Osam 32bitnih registara opšte namene x86 je prošireno na 64 bita u x64. Prefiks imena registra R označava 64bitne registre. Na pr., u x64, prošireni x86 **EAX** registar se zove **RAX**. Pod komponente registra x86 **EAX, AX, AH i AL** i dalje su dostupne u x64.
- Instrukcijski pokazivač, RIP, sada je 64 bitan. Registar flega, **RFLAGS**, takođe se proteže na 64 bita, iako su gornja 32 bita rezervisana. Donja 32 bita **RFLAGS** su ista kao **EFLAGS** u x86 arhitekturi.
- Dodato je osam 64-bitnih registara opšte namene, nazvanih R8 do R15.

- 64-bitni celi brojevi su podržani kao izvorni tip podataka.
- x64 procesori zadržavaju opciju rada u x86 kompatibilnom režimu. Ovaj režim omogućava korišćenje 32-bitnih operativnih sistema i omogućava da bilo koja aplikacija napravljena za x86 radi na x64 procesorima. U 32-bitnom režimu kompatibilnosti, 64-bitna proširenja nisu dostupna.

Virtuelne adrese u x64 arhitekturi su široke 64 bita, podržavaju adresni prostor od 16 eksabajta (**EB**), što je ekvivalentno 2^{64} bajta.

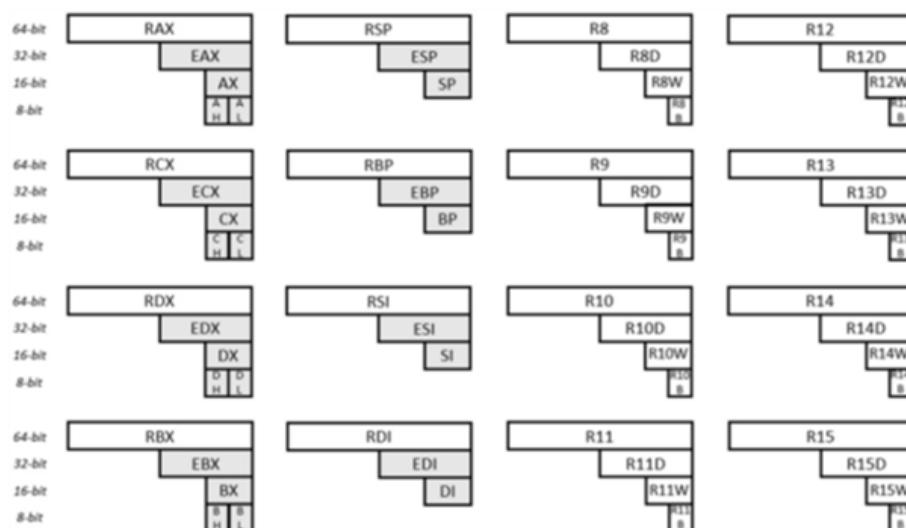
Trenutni procesori iz AMD-a i Intel-a, međutim, podržavaju samo 48 bita virtuelnog adresnog prostora. Ovo ograničenje smanjuje složenost procesorskog hardvera dok i dalje podržava do 256 terabajta (**TB**) virtuelnog adresnog prostora.

Procesori trenutne generacije takođe podržavaju maksimalno 48 bita fizičkog adresnog prostora. Ovo omogućava procesoru da adresira 256 TB fizičke **RAM** memorije sa slučajnim pristupom (engl. **Random-Access Memory**),

SKUP X64 REGISTARA

U arhitekturi x64, proširenje dužine x86 registra na 64 bita odgovara dodavanje registara

U arhitekturi x64, proširenje dužine x86 registra na 64 bita i dodavanje registara R8 do R15 rezultira mapom registara prikazanom na slici:



Slika 1.1 x64 registri [Izvor: Autor]

Na slici 1, x86 registri opisani u prethodnom odeljku (i prisutni u x64) su osenčeni x86 registri imaju ista imena i iste su veličine kada rade u 64-bitnom režimu.

64-bitne proširene verzije x86 registara imaju imena koja počinju slovom R. Novim 64-bitnim registrima (R8 do R15) može se pristupiti u manjim širinama koristeći odgovarajuće slovo sufiksa:

- Sufiks D pristupa nižim 32 bita registar: R11D
- Sufiks V pristupa nižim 16 bitova registra: R11V
- Sufiks B pristupa nižim 8 bitova registra: R11B

Za razliku od x86 registara, novi registri u x64 arhitekturi su zaista opšte namene i ne rade bilo koje posebne funkcije na nivou instrukcija procesora.

KATEGORIJE I FORMATI INSTRUKCIJA X64

Arhitektura x64 implementira u suštini isti skup instrukcija kao x86, sa 64-bitnim ekstenzijama.

Kada radi u 64-bitnom režimu, arhitektura x64 koristi podrazumevanu veličinu adrese od 64 bita i podrazumevanu veličinu operanda od 32 bita. Novi bajt prefiksa operacijskog koda, **REX**, specificira upotrebu 64-bitnih operanada.

Format x64 instrukcija u memoriji odgovara formatu x86 arhitekture, sa nekim izuzecima koji su, za naše potrebe, manji. Dodatak podrške za bajt prefiksa **REX** je najznačajnija varijacija u odnosu na format instrukcija x86.

Pomeranja adrese i neposredne vrednosti u okviru nekih instrukcija mogu biti široke 64 bita, pored svih širina bita koje podržava x86. Iako je moguće definisati instrukcije koje su duže od 15 bajtova, dekoder instrukcija procesora će pokrenuti opštu grešku zaštite ako se pokuša dekodirati instrukcija duža od 15 bajtova.

Izvorna datoteka x64 asemblerorskog jezika za „hello“ program je kao x86 verzija ovog koda, sa nekim značajnim razlikama:

- Ne postoji direktiva koja specificira memorijski model jer postoji samo jedan x64 memorijski model.
- Windows x64 aplikacijski programski interfejs (**API**) koristi konvenciju pozivanja koja čuva prva četiri argumenta pozvane funkcije u **RCX**, **RDX**, **R8** i **R9** registrima, tim redosledom.

```
.code
includelib libcmtd.lib
includelib legacy_stdio_definitions.lib

extern printf:near
extern exit:near

public main
main proc
    ; Reserve stack space
    sub rsp, 40

    ; Print the message
    lea rcx, message
    call printf

    ; Exit the program with status 0
    xor eax, eax
    mov rax, exit
    jmp rax

message db "Hello, World!", 0
```

```
xor rcx, rcx
call exit
main endp

.data
message db "Hello, Computer Architect!",0

end
```

Ovo se razlikuje od podrazumevane konvencije o x86 pozivanju, koja gura parametre u stek. Obe funkcije biblioteke koje poziva ovaj program (*printf* i *exit*) uzimaju jedan argument, prosleđen u *RCX*.

```
.code
includelib libcmt.lib
includelib legacy_stdio_definitions.lib
extern printf:near
extern exit:near
public main

main proc
    ; Reserve stack space
    sub rsp, 40
    ; Print the message
    lea rcx, message
    call printf
    ; Exit the program with status 0
    xor rcx, rcx
    call exit
main endp

.data
message db "Hello, Computer Architect!",0
end
```

Izvorna datoteka x64 asemblerorskog jezika

▼ Poglavlje 2

64-bitna ARM arhitektura

ARM ARHITEKTURA

ARM arhitekture definišu porodicu RISC procesora pogodnih za upotrebu u širokom spektru aplikacija.

Firma Arm je jedan od "preživjelih potomaka" firme Acorn.

Akronim **ARM** je prvo korišten 1983. i tada je značio "Acorn RISC Machine", a od 1990. god. znači Napredne RISC maštine (engl. **Advanced RISC Machines**) (kako se zvala i firma ARM od 1990. do 1998.).

ARM je arhitektura skupa instrukcija za procesore bazirane na **RISC** (engl. **Reduced Instruction Set Computer**) (procesor sa smanjenim skupom naredbi), arhitekturi razvijena od strane britanske kompanije **ARM Holdings**. **RISC**-bazirani pristup računarskog dizajna znači da **ARM** procesori zahtevaju znatno manje tranzistora od tipičnih procesora kod prosečnih računara.

Ovaj pristup smanjuje cenu, zagrevanje i potrošnju energije. Ovo su željene osobine kod lakih, prenosnih, uređaja koji rade na bateriju uključujući pametne telefone, laptopove, tablete, idruge ugrađene sisteme.

Jednostavniji dizajn omogućava efikasnije višejezgarne procesore i veći broj jezgara po manjoj ceni, što daje veću računarsku moć i višu energetsku efikasnost za servere i super računare.

Mnoge aplikacije **ARM** arhitektura su **sistem-na-čipu** (**SoC**) (engl. **System on chip**) dizajn koji kombinuje procesor sa specijalizovanim hardverom za podršku funkcijama kao što su ćelijske radio komunikacije u pametnim telefonima.

ARM procesori se koriste u širokom spektru aplikacija, od malih uređaja na baterije do super računara. **ARM** procesori služe kao ugrađeni procesori u bezbednosno kritičnim sistemima kao što su automobilske kočnice protiv blokiranja i kao procesori opšte namene u pametnim satovima, prenosivim telefonima, tabletima, laptop računarima, desktop računarima i serverima.

Od 2021. godine proizvedeno je preko 180 milijardi **ARM** procesora.

ARM procesori su pravi **RISC** sistemi sa velikim skupom registara opšte namene i izvršavanjem većine instrukcija u jednom ciklusu. Standardne **ARM** instrukcije imaju fiksnu širinu od 32 bita, iako je poseban skup instrukcija promenljive dužine pod nazivom T32 dostupan za aplikacije gde je memorija na prvom mestu. T32 skup instrukcija koristi mešavinu 16- i 32-bitnih instrukcija.

ARM procesori trenutne generacije podržavaju i ARM i T32 skupove instrukcija i mogu se prebacivati između dva seta u hodu. Većina operativnih sistema i aplikacija radije koristi T32 skup instrukcija u odnosu na ARM skup jer je gustina koda poboljšana.

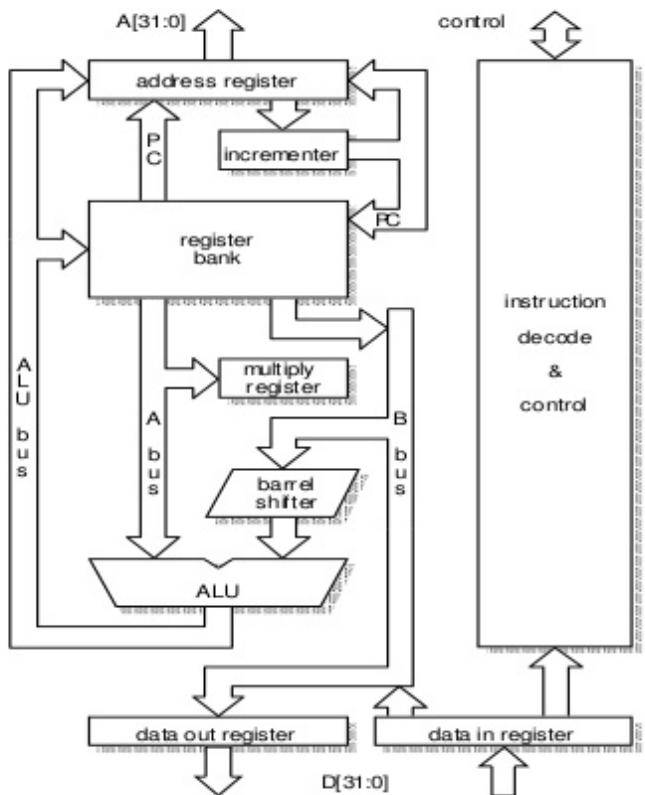
ARM ARHITEKTURA - STRUKTURA PROCESORA

ARM arhitektura definiše nekoliko režima procesora, u zavisnosti od implementiranih osobina arhitekture.

ARM arhitektura definiše nekoliko režima procesora, u zavisnosti od implementiranih osobina arhitekture.

U jednom trenutku, procesor se može nalaziti u samo jednom režimu, ali može menjati režime usled spoljašnjih događaja (prekida) ili programske.

- **Korisnički (USR) režim** - Korisnički režim (engl. User) jedini neprivilegovan režim,
- **Režim zahteva za prekid (FIQ)** - Privilegovan režim u koji se ulazi usled FIQ (engl. Fast Interrupt Request) prekida,
- **Režim zahteva za prekid (IRQ)** - Privilegovani režim u koji se ulazi usled IRQ (engl. Interrupt Request) prekida,
- **Supervizorski (SVC) režim** -Privilegovani režim u koji se ulazi usled resetovanja procesora ili izvršenja SVC (engl. Supervisor) instrukcije,
- **Nedefinisani režim (UND)** – Privilegovani režim u koji se ulazi usled nedefinisanog instrukcijskog izuzetka, (engl. Undefined)
- **Sistemski režim (SYS)** - Jedini privilegovani režim u koji se ne ulazi usled izuzetka. Može mu se pristupiti isključivo pomoću instrukcije koja direktno upisuje režimske bitove CPSR-a,
- **Zaustavni režim (ABT)** - Privilegovani režim u koji se ulazi usled prefetch abort ili data abort izuzetaka, ABT (engl. Abort).
- **Monitoring režim (MON)** - Priviligovani režim za monitoring



Slika 2.1 Struktura procesora sa ARM arhitekturom [Izvor: Autor]

ARM STRUKTURA PROCESORA

Za potrebe operativnih sistema i korisničkih aplikacija, najvažniji nivoi privilegija su USR i SVC.

ARM je arhitektura **LOAD/STORE**, koja zahteva da se podaci učitaju iz memorije u registar pre nego što se sa njim izvrši bilo kakva obrada kao što je **ALU** operacija.

Sledeća instrukcija pohranjuje rezultat nazad u memoriju. Iako ovo može izgledati kao korak unazad od x86 i x64 arhitektura, koje rade direktno na operandima u memoriji u jednoj instrukciji, u praksi, pristup **LOAD/STORE** dozvoljava da se nekoliko uzastopnih operacija izvrši velikom brzinom na operandu nakon što se je učitano u jedan od mnogih registara procesora.

Postoji potencijalno zbumujuća razlika između imena tipova podataka ARM arhitekture i onih u arhitekturi x86 i x64: u x86 i x64, reč ima 16 bita, a dvostruka reč je 32 bita. U ARM-u, reč je 32 bita, a dvostruka reč je 64 bita.

ARM arhitektura pruža sistemske mogućnosti izvan onih koje ima glavni procesor preko koncepta koprocesora. Svaki koprocesor implementira specijalizovanu kategoriju funkcionalnosti kao podršku glavnem procesoru.

U sistem se može implementirati do 16 koprocesora, sa unapred definisanim funkcijama dodeljenim četiri od njih.

Koprocesor 15 implementira [MMU](#) i druge sistemske funkcije. Ako postoji, koprocesor 15 mora podržavati instrukcijske opkodove, skup registara i ponašanja specificirana za [MMU](#). Koprocesori 10 i 11 se kombinuju da obezbede funkcionalnost sa pokretnim zarezom u procesorima opremljenim tom funkcijom. Koprocesor 14 obezbeđuje funkcije za otklanjanje grešaka.

[ARM](#) arhitekture su evoluirale kroz nekoliko verzija tokom godina.

Arhitektonska varijanta koja je trenutno u širokoj upotrebi je [ARMv8-A](#). [ARMv8-A](#) podržava 32-bitne i 64-bitne operativne sisteme i aplikacije.

32-bitne aplikacije mogu da rade pod 64-bitnim [ARMv8-A](#) operativnim sistemom. Praktično svi vrhunski pametni telefoni i prenosivi elektronski uređaji proizvedeni od 2016. dizajnirani su oko procesora ili [SoC](#)-a zasnovanih na [ARMv8-A](#) arhitekturi.

Opis koji sledi će se fokusirati na [ARMv8-A](#) 64-bitni režim.

64-BITNA ARM ARHITEKTURA I SKUP INSTRUKCIJA

64-bitna verzija ARM arhitekture, nazvana AArch64, objavljena je 2011.

64-bitna verzija [ARM](#) arhitekture, nazvana [AArch64](#), objavljena je 2011. Ova arhitektura ima 31 64-bitni registar opšte namene, 64-bitno adresiranje, 48-bitni virtualni adresni prostor, i novi skup instrukcija pod nazivom A64.

64-bitni skup instrukcija je nadskup 32-bitnog skupa instrukcija, dozvoljavajući postojećem 32-bitnom kodu da radi neizmenjeno na 64-bitnim procesorima.

Instrukcije su široke 32 bita, a većina operanada je 32 ili 64 bita. Funkcije registra A64 se u nekim aspektima razlikuju od 32-bitnog režima: programski brojač više nije direktno dostupan kao registar i obezbeđen je dodatni registar koji uvek vraća vrednost operanda nula.

Na nivou privilegija korisnika, većina A64 instrukcija ima istu mnemoniku kao i odgovarajuće 32-bitne instrukcije.

Asembler određuje da li instrukcija radi na 64-bitnim ili 32bitnim podacima na osnovu datih operanda.

Sledeća pravila određuju dužinu operanda i veličinu registra koje koristi instrukcija:

- 64-bitna imena registara počinju slovom X; na primer, X0
- 32-bitna imena registara počinju slovom W; na primer, W1
- 32-bitni registri zauzimaju niža 32 bita odgovarajućeg 64-bitnog broja registra.

Kada radite sa 32-bitnim registrima, primenjuju se sledeća pravila:

- Operacije registra kao što je pomeranje udesno ponašaju se isto kao u 32-bitnim registrima. bit arhitekture. 32-bitna aritmetička desna pomera koristi bit 31 kao predznak, a ne bit 63.

- Kodovi uslova za 32-bitne operacije se postavljaju na osnovu rezultata u niža 32 bita.
- Upisi u W registar postavljaju gornja 32 bita odgovarajućeg X registra na nulu.

64-BITNI ARM - POREĐENJE I RAZLIKE SA 32 BITNIM

Postoje neke razlike i ograničenja u poređenju sa 32-bitnim uputstvima za učitavanje i skladištenje

A64 je arhitektura **LOAD/STORE** sa istom mnemonikom instrukcija za memorijske operacije (**ldr** i **str**) kao 32-bitni režim. Postoje neke razlike i ograničenja u poređenju sa 32-bitnim uputstvima za učitavanje i skladištenje:

- Osnovni registar mora biti **X** (64-bitni) registar.
- Odstupanje adrese može biti bilo kojeg od istih tipova kao u 32-bitnom režimu, kao i **X** registar. 32-bitni offset može biti proširen na nulu ili predznak na 64 bita.
- Indeksirani načini adresiranja mogu koristiti samo trenutne vrednosti kao pomak.
- **A64** ne podržava **ldm** ili **stm** instrukcije za učitavanje ili skladištenje više registara u jednoj instrukciji. Umesto toga, **A64** dodaje **ldp** i **stp** instrukcije za učitavanje ili čuvanje para registara u jednoj instrukciji.
- **A64** podržava samo uslovno izvršenje za mali podskup instrukcija. Operacije steka se značajno razlikuju u **A64**. Možda je najveća razlika u ovoj oblasti u tome što pokazivač steka mora održavati 16-bajtno poravnanje kada pristupa podacima.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

▼ Poglavlje 3

RISC-V arhitektura

ARHITEKTURA SKUPA INSTRUKCIJA

Arhitektura skupa instrukcija (eng. ISA, Instruction Set Architecture) opisuje na koji način određeni procesor funkcioniše i koje su njegove mogućnosti.

Arhitektura skupa instrukcija ISA (engl. **Instruction Set Architecture**) opisujena koji način određeni procesor funkcioniše i koje su njegove mogućnosti. Ona opisuje registre koje će procesor imati kao i sve mašinske instrukcije koje će podržavati.

ISA specificira do detalja šta koja instrukcija radi i na koji način će ona biti kodovana u memoriji. **ISA** predstavlja spregu između hardvera i softvera. Hardverski inžinjeri dizajniraju i modeluju kola koja će izvršavati dati skup instrukcija.

Softverski inžinjeri pišu kod (operativne sisteme, kompjulere) bazirane na ovome skupu instrukcija.

Do danas je postojalo mnoštvo arhitektura u masovnoj primeni, pri čemu se dele u dve grupe:

- **RISC** (engl. **Reduced Instruction Set Computer**) Smanjena arhitektura skupa instrukcija,
- **CISC** (engl. **Complex Instruction Set Computer**) Arhitektura kompleksnog skupa instrukcija

RISC - koristi se veći broj jednostavnih instrukcije kako bi se izvršio program, pri čemu se instrukcije izvršavaju u jednom mašinskom ciklusu.

Dekodovanje instrukcija je brzo (fiksna širina kodovanja instrukcija) a implementacija protočne obrade laka. Implementacija procesora zahteva manji broj tranzistora. Veći akcenat se stavlja na softver (kompjuler).

Koristi registrsko - registrski model, pri čemu se aritmetičko-logičke operacije izvršavaju strogo nad registrima. Instrukcije LOAD i STORE se koriste za prebacivanje sadržaja iz memorije u registre i suprotno. Kodovi su često veliki i zazimaju mnogo prostora u RAM memoriji.

CISC - koristi se manji broj kompleksnih instrukcija kako bi se izvršio program, pri čemu se instrukcije izvršavaju više taktova.

Dekodovanje i implementacija instrukcija je komplikovana (promenljiva širina kodovanja instrukcija), te zahteva veći broj tranzistora.

Operacije nisu strogo registrarsko registrarske, te je moguće da su operandi u memoriji. Veličina koda je manja.

RISC-V

Jedna od osnovnih prednosti RISC-V je to da je ovaj set instrukcija otvorenog koda "open-source".

RISC-V arhitekturu, javno objavljenu 2014. god., razvili su na Univerzitetu Kalifornije, Berkli, Junsup Li, Krste Asanović, Dejvid A. Paterson i Endru Voterman. Ovaj napor je usledio nakon četiri prethodna velika RISC arhitektonska projekta na UC Berkeley, što je dovelo do imena RISC-V, gde V predstavlja rimski broj pet.

RISC-V je specifikacija potpuno otvorenog koda za procesor smanjenog skupa instrukcija. Kompletne specifikacije korisničkog režima (neprivilegovanog) i privilegovanog skupa instrukcija su objavljene i trenutno je dostupan širok izbor hardverskih implementacija ove arhitekture.

Postoje specifikacije za brojne ekstenzije skupa instrukcija koje podržavaju računarstvo opšte namene, računarstvo visokih performansi i ugrađene aplikacije. Komercijalno dostupni procesori implementiraju mnoga od ovih proširenja. Sledeće teme će biti obrađene u ovom poglavlju:

RISC-V projekat je počeo kao čist list sa nekoliko glavnih ciljeva:

- Dizajnirati **RISC** arhitekturu skupa instrukcija (**ISA**) pogodnu za upotrebu u širokom spektru aplikacija, od uređaja sa mikro napajanjem do multiprocesora u oblaku visokih performansi.
- Obezbedite **ISA** koji je sloboden da koristi bilo ko, za bilo koju aplikaciju. Ovo je u suprotnosti sa **ISA** ima skoro svih drugih komercijalno dostupnih procesora, koji su pažljivo čuvana intelektualna svojina kompanije koja ih poseduje.
- Uključiti lekcije naučene iz prethodnih decenija dizajna procesora, izbegavajući pogrešna skretanja i neoptimalne karakteristike koje druge arhitekture moraju da zadrže u novijim generacijama da bi održale kompatibilnost sa prethodnim, ponekad drevnim u tehnološkom smislu, generacijama.
- Obezbedite malu, ali kompletну osnovnu arhitekturu pogodnu za upotrebu u ugrađenim uređajima. Osnovni **RISC-V** je 32-bitni procesor sa 31 registrom opšte namene.
- Obezbedite opcione **ISA** ekstenzije za podršku matematici sa pokretnim zarezom, operacijama atomske memorije i množenju i deljenju.
- Obezbedite dodatna **ISA** proširenja za podršku privilegovanih režima izvršavanja, slično privilegovanim implementacijama **x86**, **x64** i **ARM**.
- Obezbedite opcione **ISA** ekstenzije za podršku 64-bitnim, pa čak i 128-bitnim veličinama reči procesora

RISC-V SET INSTRUKCIJA

Kao što je prethodno pomenuto, dizajner koji pravi procesor sa RISC-V arhitekturom bira koje će ekstenzije implementirati a koje ne.

Kako bi se lakše opisalo šta je prisutno u datoј hardverskoј implementaciji svaka od ovih ekstenzija ima sebi pridruženo veliko slovo abecede.

Svaki RISC-V skup instrukcija kreće slovima RV a zatim je praćen širinom registara u datoј implementaciji:

- **RV32 32-bitni procesor**
- **RV64 64-bitni procesor**
- **RV128 128-bitni procesor**

Nakon širine registara se određuju skupovi instrukcija koji će biti implementirani:

- **I** Osnovni set instrukcija za rad sa celim brojevima
- **M** Množenje i deljenje celobrojnih podataka
- **A** Atomičke operacije za interprocesnu sinhronizaciju
- **F** Operacije brojevima sa pokretnim zarezom, jednostruka preciznost
- **D** Operacije brojevima sa pokretnim zarezom, dvostruka preciznost

Sledeća proširenja procesora su takođe podržane:

- **S Nadzorni** (engl. *Supervisor*) mod (podrška za rad sa operativnim sistemima)
- **Q Četvorostruka** (engl. *Quadruple*) (128-bitna) preciznost za operacije sa pokretnim zarezom
- **C Kompresovane** (engl. *Compressed*) (16-bitne) instrukcije
- **E Ugrađeni** (engl. *Embedded*) mikroprocesor, broj registara smanjen na 16

U dokumentaciji za *RISC-V ISA* su takođe naglašene sledeće modifikacije koje sada nisu podržane ali će biti specificirane u budućnosti:

- **L** Decimalna aritmetika
- **V** Vektorske instrukcije
- **P** Pakovane instrukcije
- **B** Operacije za manipulisanje bitima
- **T** Podrška za transakcioni memorijski model
- **J** Podrška za dinamičko prevođenje instrukcija
- **N** Podrška za prekide na korisničkom nivou

RISC-V dokumentacija (<https://content.riscv.org>) pruža mnoge uvide u odluke prilikom specificiranja dizajna.

RISC-V - REGISTRI

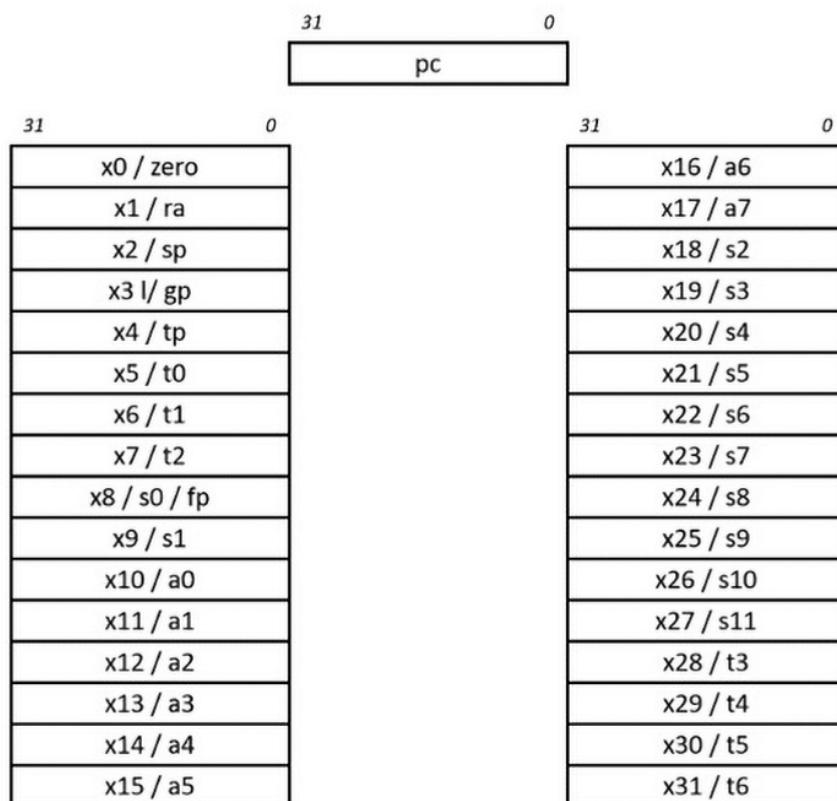
RISC-V procesori su danas dostupni na tržištu po konkurentnim cenama i možemo očekivati da će tržišni udio RISC-V procesora brzo rasti u narednim godinama.

Dostupne su **RISC-V** Linux distribucije koje uključuju sve alate za razvoj softvera potrebne za pravljenje i pokretanje aplikacija na računarima i pametnim uređajima zasnovanim na **RISC-V**.

Neke aplikacije u kojima su **RISC-V** procesori počeli da ostvaruju značajan prodor su:

- Veštačka inteligencija i mašinsko učenje
- Ugrađeni sistemi
- Računanje ekstremnih razmara
- Obrada sa ultra-niskom potrošnjom
- Obrada na bazi **IoT**-a (engl. **Internet of things**) Internet stvari

Registri su široki 32 bita. Registri opšte namene x1 do x31 mogu se koristiti bez ikakvih ograničenja i bez posebnih funkcija koje im dodeljuje hardver procesora. Registar k0 je ožičen da vrati nulu kada se pročita i odbaciće svaku vrednost koja je upisana u njega. Svaki registar ima jedno ili dva alternativna imena, prikazana na slici. Ova imena odgovaraju upotrebi registara u standardnom binarnom interfejsu aplikacije RISC-V koji je neophodan zato što su registri x1-x31 funkcionalno zamenljivi, a kompatibilnost softvera zahteva da navedemo koji registar treba da služi kao pokazivač steka, koji registri treba da sadrže argumente funkcijama, koji treba da sadrže povratne vrednosti,



Slika 3.1 RISC-V osnovni ISA registrarski set [Izvor: Autor]

RISC-V OSNOVNI SET INSTRUKCIJA

Osnovni set instrukcija RISC-V sastoji se od samo 47 instrukcija.

Osnovni set instrukcija RISC-V sastoji se od samo 47 instrukcija.

Osam su sistemske instrukcije koje obavljaju sistemske pozive i pristupaju brojačima performansi. Preostalih 39 instrukcija spadaju u kategorije računarskih instrukcija, instrukcija toka kontrole i instrukcija za pristup memoriji. Razmotrićemo svaku od ovih kategorija redom.

Računske instrukcije: Sve računske instrukcije osim *lui auipc* koriste formu sa tri operanda. Prvi operand je odredišni registar, drugi je izvorni registar, a treći je ili drugi izvorni registar ili neposredna vrednost. Mnemotehnika instrukcija koja koristi neposrednu vrednost (osim *auipc*) završava se slovom *i*.

Instrukcije toka kontrole: Instrukcije uslovnog grananja vrše poređenja između dva registra i, na osnovu rezultata, mogu preneti kontrolu unutar opsega potpisanih 12-bitnih pomaka adrese sa trenutnog računara. Dostupne su dve instrukcije bezuslovnog skoka, od kojih jedna (*jalr*) omogućava pristup celom 32-bitnom opsegu adresa.

Instrukcije za pristup memoriji: Instrukcije za pristup memoriji prenose podatke između registra i memorijske lokacije. Prvi operand je registar koji se učitava ili čuva. Drugi operand je registar koji sadrži memorijsku adresu. Potpisana 12-bitna neposredna vrednost se dodaje adresi u registru da bi se proizvela konačna adresa koja se koristi za učitavanje ili skladištenje.

Instrukcije za učitavanje vrše proširenje predznaka za vrednosti sa znakom ili proširenje nule za nepotpisane vrednosti. Operacija proširenja znaka ili nule popunjava sva 32 bita u odredišnom registru kada se učita manja vrednost podataka (bajt ili polureč).

Sistemske instrukcije: Od osam sistemskih instrukcija, jedna poziva sistemski poziv, jedna inicira tačku prekida debagera, a preostalih šest čita i upisuje sistemske kontrolne i statusne registre (CSR).

Instrukcije za CSR manipulaciju čitaju trenutnu vrednost izabranog CSR-a u registar, a zatim ažuriraju CSR pisanjem nove vrednosti, brisanjem izabranih bitova ili postavljanjem izabranih bitova.

Pseudo-instrukcije: Arhitektura *RISC-V* ima zaista smanjen skup instrukcija, nedostaje nekoliko tipova instrukcija prisutnih u skupovima instrukcija koje smo istraživali u ranijim lekcijama.

Funkcije mnogih od onih poznatijih instrukcija mogu se izvršiti pomoću *RISC-V* instrukcija, mada možda ne na odmah intuitivan način.

64-BITNI RISC-V

64-bitni skup instrukcija proširuje 32-bitni arhitekturu na 64-bitnu.

RV64I skup instrukcija proširuje *RV32I* na 64-bitnu arhitekturu. Kao i u *RV32I*, instrukcije su široke 32 bita. U stvari, *RV64I* skup instrukcija je skoro u potpunosti isti kao *RV32I*, osim ovih značajnih razlika:

- Celobrojni registri su prošireni na 64 bita.
- Adrese su proširene na 64 bita.
- Broj pomeranja bita u operacijskim kodovima instrukcija se povećava sa 5 na 6 bitova.

RISC-V arhitektura eksplisitno podržava varijacije koje definiše korisnik u obliku prilagođenih kodova operacija, koprocesora i drugih modifikacija, sve dok su kompatibilne sa pravilima prilagođavanja **RISC-V**.

Počevši od **RISC-V** dizajna otvorenog koda, možete implementirati prilagođene modifikacije koje će garantovano ostati kompatibilne sa budućim verzijama **RISC-V** standarda i proširenja.

31	27	26	25	24	20	19	15	14	12	11	7	6	0	
funct7		rs2		rs1		funct3		rd		opcode				R-type
imm[11:0]				rs1		funct3		rd		opcode				I-type
imm[11:5]		rs2		rs1		funct3		imm[4:0]		opcode				S-type
imm[12 10:5]		rs2		rs1		funct3		imm[4:1 11]		opcode				B-type
		imm[31:12]						rd		opcode				U-type
		imm[20 10:1 11 19:12]						rd		opcode				J-type

Slika 3.2 Formatni modeli instrukcija RV32I arhitekture [Izvor: autor]

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

▼ Poglavlje 4

Qualcomm snapdragon procesori

SNAPDRAGON PROCESORI

Snapdragon je najčešći SoC koji se koristi na tržištu Android pametnih telefona

Čipset je u stvari isto što i "sistem na čipu" (*SoC*). Verujemo da ste češće nailazili na termin *SoC* jer se on koristio kod računara takođe. Sve u svemu, čipset je skup integrisanih komponenti kao što su procesor, grafički čip, modem, kontroler memorije, kao i ostali kontrolери.

Znači, čipset je skup određnih komponenti koje su sinhronizovane tako da pruže maksimum od sebe. Jako je bitno da se zapamti da su to sve integrisane komponente. To govorimo jer čipset ne bi imao nikakvu važnost da nisu.

Kao što imamo *Intel* i *AMD* za desktop računare, tako imamo i **Snapdragon**, **Exynos**, **MediaTek** za pametne telefone.

Najpopularniji i najmoćniji čipseti kod pametnih telefona su **Qualcomm Snapdragon** čipseti. MediaTek je vlasnik povoljnog a veoma moćnih Helio čipseta.

Qualcomm, **MediaTek** i **Intel** su kompanije koje su prepoznatljive po svojim čipsetima, ali one nisu jedine koje prave tako nešto. Recimo Samsung pravi svoje čipsete, nazvao ih je **Exynos**. Huawei je takođe rešio da napravi svoje čipsete, a imenovao ih je **HiSilicon Kirin**.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

SNAPDRAGON PROCESORI - KRAIT

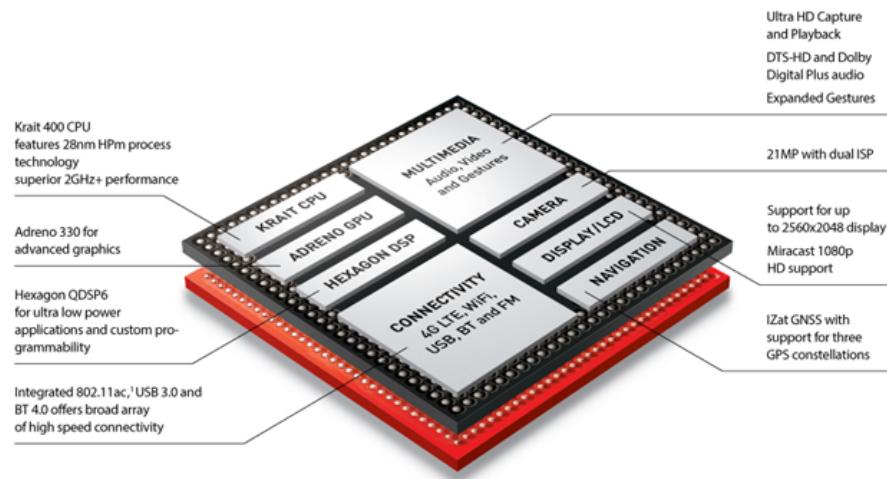
Krait je serija mirkoarhitekture procesora čiji je tvorac Qualcomm, a bazirani su na ARM arhitekturi.

Treba istaći prvo to da recimo **Snapdragon 820** nije procesor, već postavka čipova među kojima su uglavnom procesori kao što su **Krait** ili **Cortex**.

Krait je zapravo serija mirkoarhitekture procesora čiji je tvorac **Qualcomm**, a bazirani su na **ARM** arhitekturi. Šta to znači? To znači da svi procesori koriste **ARM v7** set instrukcija što podrazumeva 32-bitnu arhitekturu.

To je danas već zastarelo. Ovo se koristilo kod **Snapdragon 400/600/800**, sve do **Snapdragon 805** čipseta.

ARM v8, dakle serije 8, ima set instrukcija koje podrazumevaju 64-bitnu arhitekturu što je u poslednjih par godina korišćenije u odnosu na **ARM v7**. Prema tome, **Qualcomm** je izumeo novu generaciju microarhitekture za svoje procesore, a nazvao je Kyro. Ako pogledate detaljnije karakteristike **Snapdragon 820** čipseta, primetićete da je upravo **Kyro** tu zastupljen.

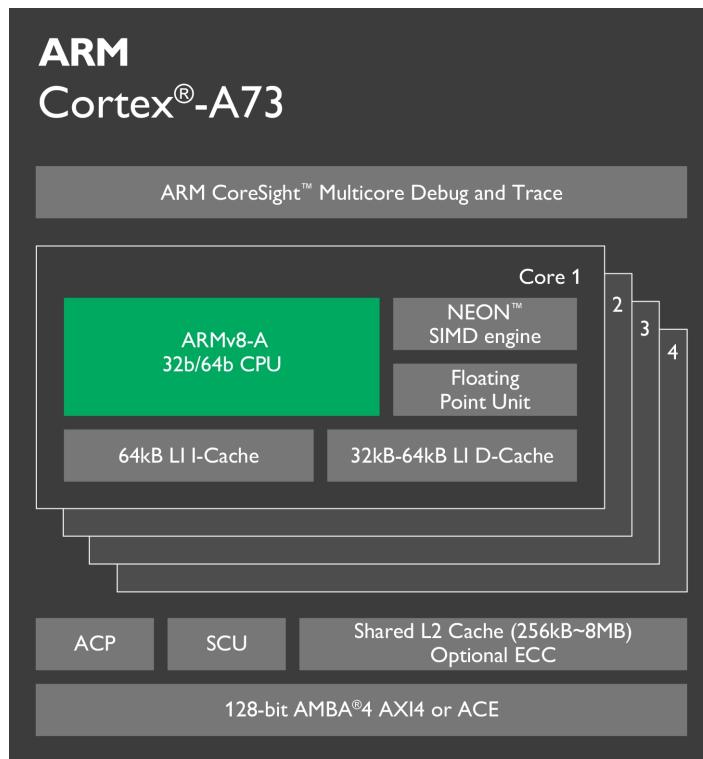


Slika 4.1 Arhitektura procesora Qualcomm Snapdragon [Izvor: Autor]

SNAPDRAGON PROCESORI - CORTEX

Coretx serija IP jezgara procesora takođe koristi ARM set instrukcija arhitekture.

Cortex je sa druge strane serija IP jezgara procesora koji su takođe mikroarhitektovani od strane **ARM** kompanije i licencirani kao IP kod drugih kompanija koji ne dizajniraju jezgra svojih procesora već samo sistem čipova oko IP jezgra samog procesora **SoC**.



Slika 4.2 Prikaz ARM Cortex A73 [Izvor: Autor]

Cortex serija IP jezgara procesora takođe koristi **ARM** set instrukcija arhitekture. **Cortex A15** je u suštini implementacija 32-bitnog **ARM v7** seta instrukcija arhitekture, a podržava 1 do 4 jezgra po klasteru tj. procesoru kavog mi znamo. Znači, ovo bi bilo nešto što bi moglo da se uporedi sa **Krait** serijom procesora.

Kasnije, kada je 64-bitni **ARM *v8** predstavljen, Cortex serija procesora su se takođe preselili na **Cortex A53** i **Cortex A73** što je nekako više ravno Krait generaciji procesora koji dolaze iz Qualcomm-a.

Kada se podvuče linija, Krait je u suštini i baziran na istom setu instrukcija arhitekture kao i Cortex, stim da je sve modifikovano i prilagođeno tako da pruži bolje i "preciznije" performanse uređaja.

Cortex jeste napredovao i omogućio 64-bitnu arhitekturu, ali u poslednje vreme sve se više koristi **Krait/Kyro**.

Cortex A72 i **A73** svakako pružaju vrhunske performanse, međutim praksa proizvođača najboljih telefona na svetu je takva da se oni radije odlučuju za **Krait** tj **Kyro**.

▼ Poglavlje 5

Apple M procesori

APPLE M1

Nova generacija Apple dizajniranih procesora nosi ime M1.

Nova generacija Apple dizajniranih procesora nosi ime **M1**. Ova nova generacija čipova koja se koristi u aktuelnim **iMac**, **MacMini** i **MacBook** računarima, te **iPad Pro** tabletu.

M1 procesor je, za razliku od dosadašnjih iz Intel-a, baziran na [ARM](#) arhitekturi, i predstavlja kompletno rešenje sistem na jednom čipu, ([SoC](#)). To znači da su sami glavni procesor, razni kontroleri kao i grafički procesor sadržani u jednom fizičkom čipu. M1 se proizvodi u pogonima najvećeg svetskog proizvođača čipova **TSMC (Taiwan Semiconductor)**.

M1 troši i daleko manje energije samo 39 vati pod punim opterećenjem, u poređenju sa 122 vati na prethodno korišćenoj generaciji Intel i7 procesora sa šest jezgara.

Performanse izmerene aplikacijama koje su standard u ovoj oblasti, pokazuju da M1 procesori imaju od 15-tak, pa sve do 40 odsto bolje rezultate u odnosu na konkurenciju.

Iako ovo pre svega zavisi od aplikacije i softvera koji se koristi, M1 se pokazao kao značajno brži kada je obrada videa visoke rezolucije u pitanju – naročito 4K i 8K fajlova.

Prelazak na novi procesor i [ARM](#) arhitekturu ima i svoje negativne strane. Popularne aplikacije trećih proizvođača rade kroz “softver za prevođenje” nazvan **Rosetta**.

On prevodi instrukcije namenjene Intel ([x86](#)) čipovima u instrukcije koje M1 može da “razume”. Sa druge strane, sve veći broj softverskih kompanija planira, ili su već predstavile, svoj softver koji radi direktno sa M1 čipovima.

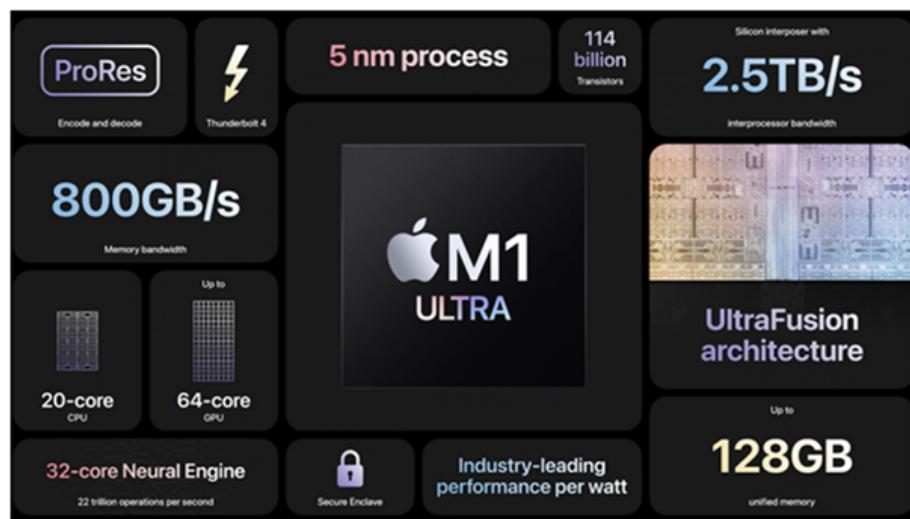
Apple je takođe upravo predstavio i unapređene verzije nazvane M1 Pro, koje donose još bolje performanse i veće brzine, čak i u odnosu na sam M1 procesor.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

APPLE M1 - STRUKTURA

Pojava M1 čipa je mnogostruko važna i za budućnost procesora.

Sam procesor se sastoji od 20 procesorskih jezgara, od kojih je 16 zaduženo za visoke performanse, a četiri za štednju energije, odnosno održavanje sistema u niskom intenzitetu rada. Za grafiku je zadužen 64-jezgreni grafički koprocesor.



Slika 5.1 Procesor Apple M1 [Izvor: Autor]

Pojava M1 čipa je mnogostruko važna i za budućnost PC računara, imajući u vidu da je reč o kulminaciji zbivanja koja se u industriji odvijaju već nekoliko godina.

Ključni podaci o M1 jesu da je to sistem na čipu (*SoC*) nastao prilagođavanjem postojećih A1x Bionic čipova koji pokreću najnovije modele iPhone a i iPad a, da je zasnovan na ARM arhitekturi i proizведен u 5 nm tehnologiji i sa 16 milijardi tranzistora.

Osnovna filozofija nove izrade čipova je tzv. Ultra Fusion, odnosno koncepcija spajanja više čipova u jednog.

U Apple-u navode da ovaj je ovaj procesor u stanju podneti bandwidth od 2,5 terabajta u sekundi. Sadrži 114 milijardi tranzistora te podržava do 128 GB RAM, odnosno objedinjene radne memorije, sa podrškom za brzinu od 800 GB/s.

SNAPDRAGON - APPLE M1 RAZLIKE

U ovom poređenju sagledavamo razlike i analiziramo dva procesora

Qualcomm Snapdragon 8 Gen 1 ili Apple M1

U ovom poređenju sagledavamo razlike i analiziramo ova dva CPU-a.

Qualcomm Snapdragon 8 Gen 1 ima 8 jezgara sa 8 niti i taktove sa maksimalnom frekvencijom od 3,00 GHz. Podržano je do 16 GB memorije u 4 memorijska kanala. Qualcomm Snapdragon 8 Gen 1 je objavljen u prvom tromesečju 2022. godine.

Apple M1 ima 8 jezgara sa 8 niti i taktove sa maksimalnom frekvencijom od 3,20 GHz. CPU podržava do 16 GB memorije u 2 memorijska kanala. Apple M1 je objavljen u pred kraj 2020. godine.

Processor	Apple M1	Qualcomm Snapdragon 8 Gen 1
ISA	ARMv8.5-A (64-bit)	ARMv9-A (64-bit)
Mikroarhitektura	Firestorm, Icestorm	Cortex-X2, Cortex-A710, Cortex-A510
Core naziv	Firestorm, Icestorm	Kryo
Familija procesora	M series	Snapdragon
Lithografija	5 nm FinFET N5	4 nm 4LPE
Broj jezgra	8	8
Broj niti	8	8
Osnovna frekfencija	2.1 GHz	1.8 GHz
Maksimalna frekfencija	3.2 GHz	3.0 GHz
Socket	SoC	SoC

Slika 5.2 Poređenje karakteristika Apple M1 i Qualcomm Snapdragon8 procesora [Izvor: Autor]

Brzina **ARM** procesora u mobilnim telefonima skoro da je pratila **Murov zakon** iz njegovog zlatnog doba.

Neki od standardizovanih testova za merenje performansi procesora, na primer Geekbench, konstantno su beležili objektivna ubrzanja kod Qualcomm i Apple procesora, bilo da je reč o performansama pojedinačnog jezgra ili ukupnoj „izlaznoj snazi“ kod paralelnog rada više jezgara.

	Redizajn ARM jezgre	Dizajn SoC čipa	Proizvodnja SoC čipa	Dizajn	Proizvodnja
Samsung	Da Exynos	Da Exynos	Da	Da	Da
Apple	Da Firestorm i Icestorm	Da M1 - za Macintosh	Ne	Da	Ne
Huawei	Ne	Da Kirin	Ne	Da	Da
Qualcomm	Da Kryo	Da Snapdragon	Ne	Ne	Ne
MediaTek	Ne	Da Dimensity	Ne	Ne	Ne

Slika 5.3 Podela posla u proizvodnji ARM čipova kod različitih proizvođača [Izvor: Autor]

KEŠ MEMORIJA QUALCOMM PROCESORA

Keš memorija Qualcomm Snapdragon procesora

Keš memorija Qualcomm Snapdragon procesora

Keš memorija, takođe nazvana **CPU** memorija, je memorija sa slučajnim pristupom male snage kojoj računarski mikroprocesor može pristupiti brže nego što može da pristupi običnoj **RAM** memoriji. Ova memorija je obično integrisana direktno sa **CPU**-om. Slika prikazuje hijerarhiju keša.



Slika 5.4 Snapdragon keš hijerarhija [Izvor: Autor]

Broj nivoa u hijerarhiji memorije i performanse na svakom nivou su se vremenom povećavali. Hijerarhija memorije Snapdragon procesora je:

Nivo 0 (L0): Keš keš nivoa 0 (**L0**) je keš operanda i veličine je od 5 KB do 50 KB. L0 je najbrži nivo keša među svim kešovima. L0 keš koji je korišćen za pristup steku L0 keš memorije je u prošlosti predložen kao jeftin način za poboljšanje performansi i smanjenje potrošnje energije. L0 keš hrani jezgra i registruje.

Nivo 1 (L1): Keš keš nivoa 1 (**L1**) je podeljen na dva dela, prvi je instrukcija, a drugi je keš podataka. Obično dolazi u samom čipu procesora. L1 keš se nalazi između L0 i L2. Keš instrukcija je veličine 128 KB, a keš podataka 128 KB. Najbolja brzina pristupa je oko 700 GB/sekundi. L1 keš hrani L0 keš, i sporiji od L0 memorije, ali brži od L2.

Nivo 2 (L2): Keš keš nivoa 2 (**L2**) je keš instrukcija i podataka (deljeni). L2 je veličine 128 KB do 1 MB. Najbolja brzina pristupa je oko 200 GB/sekundi. L2 je veći od primarne keš memorije. Takođe dolazi unutar samog procesorskog čipa. L2 keš se nalazi između L1 i LPRAMa. L2 keš hrani L1 keš memoriju i sporiji je od L1 memorije, ali brži od L3. Nivo 3 (L3):

Keš keš nivoa 3 (L3) je deljeni keš. L3 keš memorija je veličine 128 KB do 6 MB. Najbolja brzina pristupa je oko 100 GB/sekundi. L3 keš hrani L2 keš, a njegova memorija je obično sporija od L2 memorije, ali brža od glavne memorije.

KEŠ MEMORIJA APPLE PROCESORA

Keš memorija Apple M serija procesora

Keš memorija Apple M serija procesora

M1 čip kombinuje mnoge karakteristike koje su ranije bile implementirane u više čipova, od **CPU** i **GPU**, koji obično dolaze od različitih dizajnera, do svih specijalizovanih interfejsa za

periferne uređaje kao što su kontroleri za skladištenje podataka. Međutim, M1 ne integriše glavnu memoriju, koja se nalazi u paketu.

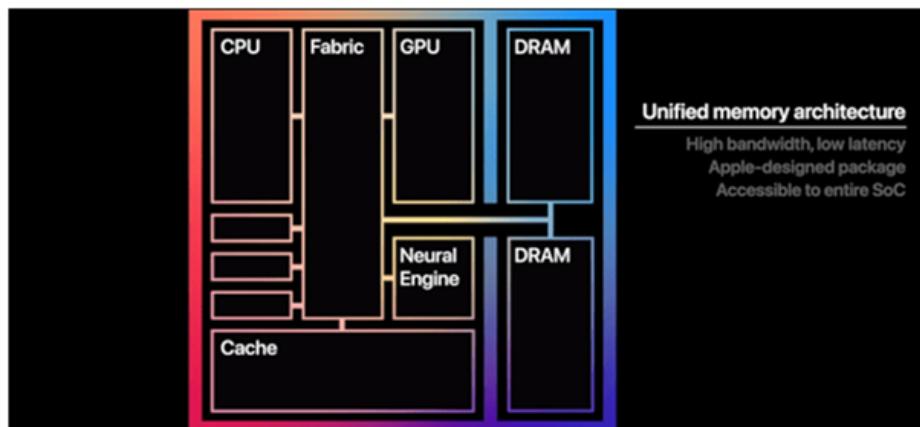
svaki M1 čip ima ukupno osam procesorskih jezgara, sve zasnovano na Apple-ovom razvoju tehnologije licencirane od strane [ARMA](#). Četiri su opisana kao jezgra performansi, nazvana Firestorm, a četiri su jezgra efikasnosti ili Icestorm. One se prvenstveno razlikuju po kompromisu između performansi i potrošnje energije, pri čemu Firestorm jezgra rade u istoj klasi kao bolje Intel jezgre, a Icestorm isporučuju niže performanse sa mnogo manjom potrošnjom energije i proizvodnjom toplote.

Međutim, postoje razlike u obezbeđivanju keš memorije:

Firestorm ima 192+128 KB L1 keš memorije po jezgru i deli ukupno 12 MB L2 keš memorije. Icestorm ima 128+64 KB L1 keš memorije po jezgru i deli ukupno 4 MB L2 keš memorije. Oba tipa jezgra podržavaju skup instrukcija [ARMv8.4-A](#), sa neonskim (128-bitni registar)

Svako jezgro ima 8široko dekodiranje na više celih izvršnih jedinica i četiri jedinice sa pomicnim zarezom [SIMD](#). Celobrojne operacije i operacije sa pokretnim zarezom, uključujući [SIMD](#), u potpunosti su podržane u kompajliranim jezicima visokog nivoa i asemblerским jezicima. Obično aplikacije ne mogu da biraju na kojoj vrsti jezgra će se pokrenuti njihovi procesi. Međutim, za zadatke u pozadini, kod može da izabere kvalitet usluge (engl. [Quality of service](#)-QoS) koji određuje kako ih macOS dodeljuje jezgrama.

Takvo asimetrično višeprocesiranje je karakteristično za M1 i retko se viđa na drugim mestima.



Slika 5.5 Arhitektura memorije kod Apple M serije procesora [Izvor: Autor]

✓ Poglavlje 6

Pokazne vežbe

POKAZNA VEŽBA #1

Pokazna vežba 1 (10 minuta)

Zadatak #1 (10 minuta)

Glavni program (GP, veličine 60KB) poziva u datom redosledu sledeće segmete programa:

A (20KB),

B (50K) i

C (40KB).

Segment B u jednom trenutku poziva segment D (40KB), koji opet poziva segment E (70KB). Segment C u jednom trenutku poziva segmet F (120KB).

- Skicirati kako se za dati primer može primeniti tehniku preklapanja (overlay)?
- Koliko je najmanje radne memorije potrebno za izvršavanje ovog programa?
- Koliko bi memorije bilo potrebno bez primene tehnike preklapanja?

Rešenje:

a) overlay 1: $GP+A = 60K+20K = 80K$ overlay 2: $GP+B+D+E = 60+50+40+70 = 220K$
overlay 3: $GP+C+F = 60+40+70 = 170K$

b) $\max(80, 220, 170) = 220K$

c) $GP+A+B+C+D+E+F = 60+20+50+40+70+40+70 = 350K$

POKAZNA VEŽBA #2

Pokazna vežba 2 (20 minuta)

Zadatak #2 (10 minuta)

Posmatrajte logički adresni prostor koji čini 8 stranica veličine 1024 memorijске reči, mapiran u 32 okvira fizičke memorije.

Koliko bita ima u logičkim, a koliko u fizičkoj adresama?

Rešenje:

Jedan okvir (odnosno jedna stranica) veličine 1024B može se alocirati sa 10 bitova.

Logički adresni prostor čini 16 stranica, koje se mogu alocirati sa 4 bita, a fizički 128 okvira, koji se mogu alocirati sa 7 bita.

Znači, u logičkim adresama ima 14 bitova, a u fizičkim 17

Zadatak #3 (10 minuta)

Zašto je veličina stranice uvek jednaka 2^n ?

Rešenje:

Straničenje se implementira razbijanjem adrese na X bitova za broj stranice i Y bitova za offset.

Kako je svaka pozicija u adresi stepen broja 2, razbijanje adrese na broj stranice i offset rezultuje veličinom stranice 2^n .

PROJEKTNI ZADATAK IZ PREDMETA CS120

Predmet CS120 od predispitnih obaveza sadrži i PZ koji se sastoji od tri mini-projekta.

Na predmetu CS120 - Organizacija računara svaki student samostalno radi projektni zadatak (PZ), koji se sastoji od tri mini-projekta.

Mini-projekti brane se u petoj, desetoj i petnaestoj nedelji semestra!

Svaki student dužan je da tok izrade projekta izveštava svake nedelje u obliku kratkih izveštaja (do jednog pasusa) preko LAMS lekcija u okviru dodatnih aktivnosti interaktivnih lekcija.

O formatu mini projekata svi studenti (tradicionalni i Internet) biće obavešteni mejlom od strane predmetnog profesora/asistenta.

✓ Poglavlje 7

Zadaci za samostalni rad

POKAZNA VEŽBA #1

Zadatak 1: (15 minuta)

Zadatak #1 (15 minuta)

Posmatrajte sistem na kome se tabela stranica čuva u memoriji.

- a.) Ako pristup memorijskoj lokaciji traje 0.2 mikro sec, koliko traje referenciranje adrese kroz tabelu stranica?
b.) Ako se koristi asocijativna memorija i ako se 25% memorijskih referenci kroz tabelu stranica mogu naći pomoću asocijativne memorije, koje je efektivno vreme pristupa nekoj memorijskoj lokaciji?

Pristup asocijativnoj memoriji traje 1 nano sec.

Rešenje:

$$a) 0,4\mu s = 0,2\mu s(\text{za pristup tabeli}) + 0,2\mu s(\text{za pristup memorijskoj lokaciji})$$

$$b) t_{eff} = 0,25 * (0,2\mu s + 1\eta s) + 0,75 * (0,4\mu s + 1\eta s) = 351\eta s$$

POKAZNA VEŽBA #2

Zadatak 2: (15 minuta)

Zadatak #2 (15 minuta)

Data je tabela stranica jednog procesa na sistemu sa veličinom stranice 2KB.

stranica	okvir
0	1
1	4
2	3
3	7
4	12

Slika 7.1 Tabela stranica jednog procesa[Izvor: Autor]

Koje fizičke adrese odgovaraju sledećim logičkim adresama:

- a) 251,
- b) 3129,
- c) 10000,
- d) 6066 ?

Rešenje:

broj stranice = ceo broj količnika logičke adrese i veličine stranice

offset = ostatak pri deljenju logičke adrese i veličine stranice

(a) Logička adresa 251

broj stranice=[251/2048]=0 okvir=1,

offset=251%2048=251, fizička adresa=1x2048+251=2299

(b) Logička adresa 3129

broj stranice=[3129/2048]=1 okvir=4,

offset=3129%2048=1081, fizička adresa=4x2048+1081=9273

(c) Logička adresa 10000

broj stranice=[10000/2048]=4 okvir=12,

offset=10000%2048=1808, fizička adresa=12x2048+1808=26384

(d) Logička adresa 6066

broj stranice=[6066/2048]=2 okvir=3,

offset=6066%2048=1970, fizička adresa=3x2048+1970=8114

✓ Poglavlje 8

Domaći zadatak

DOMAĆI ZADATAK #14

Domaći zadatak #14 okvirno se radi 45 minuta

Domaći zadatak #14/1 (15 minuta)

Data je tabela stranica jednog procesa na sistemu sa veličinom stranice 4KB.

stranica	okvir
0	3
1	5
2	8
3	12
4	15

Slika 8.1 Tabela stranica jednog procesa[Izvor: Autor]

Koja fizička adresa odgovara logičkoj adresi koja je jednaka broju vašeg indeksa?

Domaći zadatak #14/2 (30 minuta)

Tekst domaćeg zadatka:

Ukratko opisati (2 do 3 strane) arhitekturu procesora svog mobilnog telefona.

Koristiti uputstvo za izradu domaćeg zadatka.

Predaja domaćeg zadatka

Tradicionalni studenti:

Domaći zadatak treba dostaviti najkasnije 7 dana nakon vežbi, za 100% poena. Nakon toga poeni se umanjuju za 50%.

Domaći zadatak poslati predmetnim asistentima, sa predmetnim profesorima u CC.

Predati domaći zadatak koristeći .doc/.docx uputstvo dato u prvoj lekciji.

Internet studenti treba poslati domaće zadatke najkasnije do 10 dana pred izlazak na ispit predmetnom asistentu zaduženog za internet studente.

Napomena:

Svaki domaći zadatak treba da bude napisan prema dokumentu za predaju domaćih zadataka koji je dat na kraju interaktivne lekcije.

✓ Poglavlje 9

Zaključak

ZAKLJUČAK

Rezime lekcije #14

U ovoj lekciji smo obradili arhitekture *x86 ix64* koje predstavljaju uglavnom *CISC* pristup dizajnu procesora, koristeći instrukcije promenljive dužine za koje je potrebno mnogo ciklusa da se izvrše, dugačak cevovod i (u x86) ograničen broj procesorskih registara. *ARM* arhitekture, s druge strane, implementiraju *RISC* procesore sa uglavnom izvršavanjem instrukcija u jednom ciklusu, velikim skupom registara i (donekle) instrukcijama fiksne dužine.

Takođe predstavili smo arhitekturu *RISC-V* procesora i njegov skup instrukcija. *RISC-V* arhitektura definiše kompletну specifikaciju korisničkog režima i privilegovanog skupa instrukcija i nekoliko proširenja za podršku računarstvu opšte namene, računarstvu visokih performansi i ugrađenim aplikacijama koje zahtevaju minimalnu veličinu koda. *RISC-V* procesori se nude komercijalno, a dostupni su besplatni proizvodi otvorenog koda za implementaciju *RISC-V* u *FPGA* uređaje.

Na kraju dali smo primere arhitekture dva najzastupljenija procesora Qualcomm Snapdragon i Apple M serije.

Literatura:

J. Ledin, Modern Computer Architecture and Organization,

Chapter 10, pp. 249 – 288,

