Arhitektura i organizacija računara (2+2+1)

nastavnici
Emina Milovanović, Ivan Milentijević
Fond sati: 2+2+1
Šifra za pristup kursu: AOR2019

Literatura

- Sopstvena sveska sa predavanja
- Sopstvena sveska sa vežbi
- Prezentacije sa predavanja
- D. Patterson and J. Hennessy, *Computer organization and design: The Hardware/Software Interface*, Fourth edition, Morgan Kaufmann Publishers, 2009, ISBN: 978-0-12-374493-7.
- W. Stallings, *Arhitektura i organizacija računara*, *Projekat u funkciji performansi*, Računarski fakultet i CET, Beograd, 2006.
- Nebojša Milenković, *Arhitektura i organizacija računara*, Elektronski fakultet, Niš, 2004, ISBN: 86-80135-85-2.

Način ocenjivanja

- Lab. Vežbe 20
- I kolokvijum 40 (20+20)
- II kolokvijum 40 (20+20)

- Ukupno 100

- Lab. Vežbe 20
- Pisani deo ispita 40
- Usmeni deo 40

- Ukupno 100

Arhitektura i organizacija 1

- Arhitektura računara (ili arhitektura skupa instrukcija, engl. Instruction Set Architecture – ISA) predstavlja atribute računara vidljive programeru, tj. atribute računara koji imaju direktan uticaj na logičko izvršenje programa
  - Skup instrukcija, formati instrukcija, načini predstavljanja podataka (broj bitova), tehnike adresiranja,
    - npr., postoje li instrukcije za množenje, djeljenje,...?
- Arhitektura je logički ili apstraktni opis računara
- Organizacija predstavlja implementaciju arhitekture računarskog sistema, tj. atribute koji nisu vidljivi programeru
  - obuhvata projektovanje funkcionalnih jedinica računara, kao što su centralni procesor, memorijski sistem, magistrale, ulazno-izlazni sistem, upravljačke signale
    - npr., postoji li HW jedinica za množenje ili je ona implementirana tehnikom ponavljanja operacije sabiranja?

Arhitektura i organizacija 2

- Mnogi proizvođači računara nude čitave familije modela računara sa istom arhitekturom, ali sa različitom organizacijom
- Ista arhitektura skupa instrukcija može imati mnogo različitih implemenatcija koje se razlikuju po ceni i performansama
- Ista arhitektura se može koristiti mnogo godina, sa različitim tehnologijama
  - Celokupn Intel x86 familija ima istu osnovnu arhitekturu skupa instrukcija
  - IBM System 360/370 familiju takođe karakteriše ista arhitektura skupa instrukcija
    - Ovo daje kompatibilnost kôda, u najmanju ruku prema generacijama unazad

Kratak sadržaj predmeta

- Uvod, performanse sistema
- Karakteristike CISC i RISC prilaza u projektovanju processora
- Upravljačka jedinica
  - Funkcija; hardverska UJ; mikroprogramska UJ
- Protočna organizacija procesora
  - Šta je protočnost, razlozi uvođenja, kako se protočnost koristi u projektovanju procesora, koje su posledice uvođenja protočnosti, hazardi i tehnike za prevazilaženje hazarda
- Računarska aritmetika
  - Sabiranje, množenje, deljenje (integer i FP)
  - Algoritmi i hw implementacija
- Memorijska hijerarhija
  - Organizacija memorije, keš memorija, virtuelna memorija

AOR

Kratak istorijat

Performanse računarskih sistema

Uvod

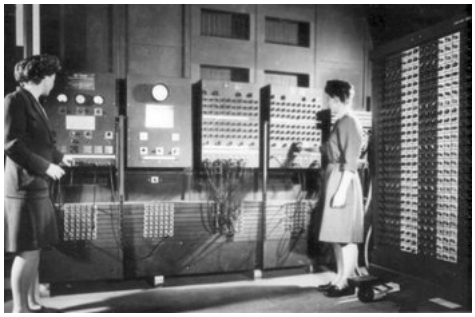
- Mada je istorija elektronskih računara kratka, ideja o konstrisanju uredjaja koji će automatizovati izračunavanja je stara nekoliko hiljada godina
- Brz i dinamičan razvoj **računara** počinje 40-ih godina XX veka
  - Preteča prvih cifarskih računara bio je MARK I konstruisan 1940 sa ciljem da reši izvesne nelinearne diferncijalne jednačine
    - Osnovne komponente računara su mehanički koturi i elektromehanički relei
    - Bio je nezgrapan, glomazan i spor
  - 1943. na Pensilvanijskom Univerzitetu započet je rad na prvom elektronskom računaru koji se može smatrati pretečom današnjih savremenih računara
    - 1946. ENIAC (Electronic Numeric Integrator And Computer) je završen
      - Dugačak 30m, težak 27t, zauzimao površinu od 167m^2
    - elektromehanički relei su zamenjeni vakuumskim cevima
    - i pored velikih dimenzija, velikog zagrevanja, kratkog veka, vakuumske cevi su omogućile drastično povećanje brzine izračunavanja
    - MARK I je operaciju sabiranja obavljao za 300.000 μs a ENIAC za 400 μs.
- U odnosu na tehnologiju koja se koristi u proizvodnji procesora, memorija i U/I urđaja, može se razlikovati 5 generacija računara.

Prva generacija računara (1946-1959)

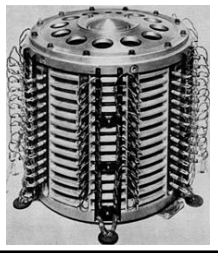
- Računari su bili izgrađeni od hiljada vakuumskih ELEKTRONSKH CEVI (to su elementi slični sijalici)
  - Velikih dimenzija, nepouzdane, kratak vek, velika potrošnja energije i veliko zagrevanje, a kao posledica velikog zagrevanja često je dolazilo do otkaza komponenti (kvarova)
- Primarna memorija od magnetnih doboša, veoma malog kapaciteta
- Za ulaz se koriste bušene kartice, a za izlaz štampač (nema tastatura i monitora)
- Programi napisani na mašinskom jeziku
- Brzina izvršenja instrukcija reda nekoliko (stotina) ms
- Veoma skupi, ograničena primena

Vakuumska elektronska cev - lampa

ENIAC

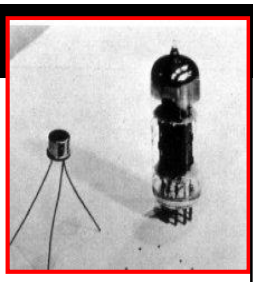


Reprogramiranje se obavljalo ručno, prevezivanjem žica.

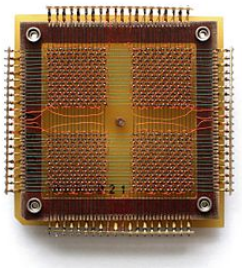


Magnetni doboš – preteča hard diska

Druga generacija 1959-1965

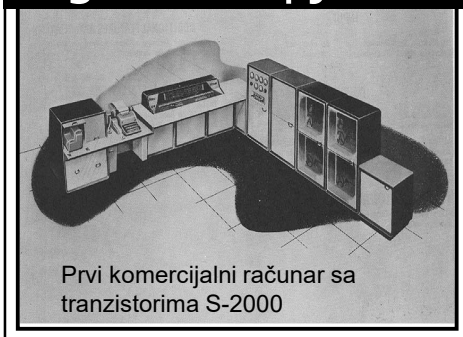


- Druga generacija računara počinje sa pronalaskom TRANZISTORA.
  - TRANZISTORI su mnogo manji, brži i jeftiniji od elektronskih cevi.
    - Operacije se izvode brzinom od nekoliko μs
  - Tranzistori su se pravili od silicijuma, elementa koji je pronađen u morskom pesku, a koji je rasprostranjen i jeftin za proizvodnju.
  - ulaz i dalje sa bušene kartice, izlaz na štampač
  - Primarna memorija od magnetnih jezgara
  - Sekundarna memorija – magnetne trake
  - Programiranje na simboličkom mašinskom jeziku (asembler)
  - Uprkos tome, i ovi računari su bili velikih dimenzija i striktno vezani za univerzitete i vlade.
  - Najpopularniji računar u to vreme bio je proizvod američke kompanije IBM, nosio je oznaku 1401.
    - Njegovom odličnom prodajom broj računara svetu se udvostručio.



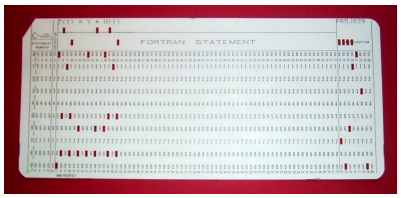
Magnetna jezgra-preteča RAM memorija

Izgled kompjutera 2. generacije



Prvi komercijalni računar sa tranzistorima S-2000

Najveći uspeh postigao je IBM-ov računar 1401



Treća generacija (1965-1971)

- Individualni tranzistori zamenjeni integrisanim kolima – sveti gral u računarskoj industriji
  - više (stotina, hiljada) minijaturnih tranzistora smeštenih u silikonskom čipu
- Pronalazak ČIPA izazvao je revoluciju u računarstvu.
  - Čipovi se odlikuju malim dimenzijama, niskom cenom, većom pouzdanošću, malom potrošnjom struje.
- Magnetne trake i diskovi su potpuno zamenili bušene kartice
  - Memorijski čipovi zamenjuju magnetna jezgra (poluprovodničke memorije)
- Za ulaz se koristi tastatura, izlaz- monitor
- Brzina izvršenja operacija meri se u ns
  - Od otkrića čipa, broj tranzistora koji mogu da stanu na jedan čip udvostručuje se svake dve godine. Time se povećava njihova snaga, a cena se smanjuje.
- Javljaju se viši programski jezici (FORTRAN, COBOL)
  - olakšano programiranje
- Sa trećom generacijom počinje ekspanzija – masovna primena računara.
  - Tipičan predstavnik – IBM System/360

Izgled računara 3. generacije

IBM 360

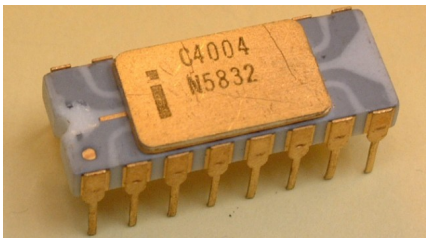
PDP-11 firme

Digital Equipment Corporation

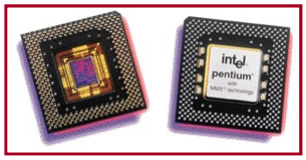
Četvrta generacija (1971-1980)

- Četvrta generacija počinje pojavom mikroprocesora
  - 1971: Intelovi inženjeri su konstruisali prvi mikroprocesor.
  - Bio je veličine 1cm^2 a sadržao je celokupnu logiku računara-CPU.
- Mikroprocesor objedinjuje dva dostignuća:
  - 1) zamenjuje hiljade integrisanih kola jednim, još manjim čipom i
  - 2) objedinjuje sve funkcije jednog računara
- Dakle, jedan mikročip izvršava sve radnje kao jedan kompletan računar
- Rezultat ovog otkrića je da ono što je nekada zauzimalo prostor čitave sobe danas može stati na dlan
- Godine 1981. kompanija IBM predstavlja prvi kućni personalni računar PC XT, a 1984. pojavljuje se računar Mekintoš kompanije Apple
- Razvijaju se računarske mreže
- Uveden pojam Interneta

- Prvi mikroprocesor:
  - Intel 4004



Mikroprocesori



Prvi personalni računari

PC XT

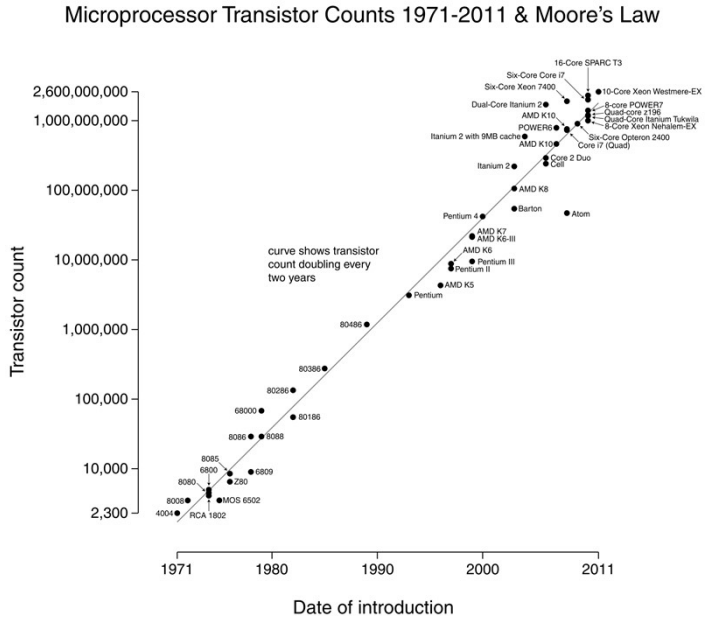
Apple 2

Peta generacija – sadašnjost i budućnost: Veštačka inteligencija

- Ova generacija računara je još u razvoju, bazira se na veštačkoj inteligenciji.
- Masovno korišćenje paralelizma u obradi
  - Više procesora na jednom čipu
- **Cilj** pete generacije računara je razviti uređaje koji "govore" prirodnim (ljudskim) jezikom i sposobni su za učenje i samoorganizovanje
- Korišćenje paralelnih procesora i super-provodnika učiniće da veštačka inteligencija postane stvarnost
- Kvantno izračunavanje, zatim molekularna i nano-tehnologija, radikalno će promeniti izgled kompjutera u vremenu koje dolazi.

Moore-ov zakon

- Kompjuterska tehnologija je učinila neverovatan napredak u poslednjih 60 godina od kada se pojavio prvi elektronski računar opšte namene.
  - Danas se za manje od 1000 Eur može kupiti PC koji ima bolje performanse, više memorije i više prostora na disku od računara proizvedenog 80-ih godina 20. v. Koji je koštao milione dolara.
- **Moore-ov zakon** kaže da se tokom istorije računarstva broj tranzistora na integrisanom kolu udvostručava približno svake 2 godine.
  - Zakon je dobio ime po suosnivaču Intela, Gordon E. Moore, koji je ovoaj trend opisao u svom radu koji je publikovan 1965. god.
  - Pokazalo se da je njegovo predviđanje prilično pouzdano.
- Moore-ov zakon se danas koristi da opiše trend performansi mnogih elektronskih komponente
  - Brzina procesiranja, kapacitet memorije, pa čak i broj piksela kod digitalne kamere.
  - Sve promene su eksponencijalne.
  - Smatralo se da će ovakav trend da se nastavi sve do 2015-2020. Međutim, od 2002 taj rast je sporiji i performanse se dupliraju na svake 3 godine.



Merenje performansi

- Kada kažemo da je jedan računar, recimo X, brži od drugog računara, recimo Y, imamo u vidu da računar X izvršava programe za kraće vreme od računara Y.
- Postizanje idealnih (vršnih) perfomansi računarskog sistema zahteva savršeno slaganje izmenju mogućnosti sistema i ponašanja programa
  - Mogućnosti mašine se mogu poboljšati
    - boljom tehnologijom
    - inovacijom arhitekture
    - efiksnijim upravljanjem resursa
  - Ponašanje programa je teško predvideti zbog visoke zavisnosti od
    - aplikacije
    - odabranog algoritma
    - struktura podataka
    - efikasnosti programskog jezika
    - tehnologije kompilatora
    - veštine programera

Merenje performansi

- Definicija vremena:
  - vreme odziva (engl. response time), ili vreme izvršenja (engl. execution time),
    - Vreme potrebno za izvršenje kompletnog zadatka
  - Ako imamo grupu servera koji izvršavaju programe mnogo korisnika, rećićemo da je brži onaj koji je obavio veći iznos posla u toku dana (ima veću propusnost)
    - propusnost (engl. throghput), određenu kao obim posla obavljen u jedinici vremena

Primer

- Da li sledeće izmene u računarskom sistemu povećavaju propusnost, smanjuju vreme izvršenja ili obe stvari?
  1. Zamena procesora bržim procesorom
  2. Dodavanje novih procesora postojećem sistemu koji ih koristi za izvršenje različitih zadataka – npr pretraživanje WWW
- Odgovor
  - smanjenje vremena izvršenje uvek dovodi i do povećenja prpusnosti.
    - U slučaju pod 1, imamo i smanjenje vremena izvršenja i povećanje propusnosti
  - U slučaju pod 2 se očigledno povećava propusnost, ali se može smanjiti i vreme izvršenja ako se uzme u obzir vreme čekanja na raspoloživost procesora.
- Kada govorimo o performansama mi ćemo se uglavnom baviti vremenom izvršenja

Mere za ocenu performansi – vreme odziva

- Vreme odziva (proteklo vreme)
  - vreme koje protekne od trenutka izdavanja zahteva do trenutka kada je zadatak izvršen (meri se u sec).
  - smatra se da je bolji onaj računar koji isti iznos posla obavi za kraće vreme
  - Vreme doziva obuhvata
    - korisničko procesorsko vreme
    - vreme pristupa memoriji
    - vreme pristupa diskovima
    - vreme potrebno za obavljanje U/I aktivnosti
    - dodatno vreme zbog poziva OS
- Performanse sistema koji izvršava odredjeni program su obrnuto proporcionalne vremenu izvršenja
  - Performanse=1/Vreme_izvršenja

Mere performsnsi – vreme odziva

- Poredjenje mašina X i Y koje izvršavaju isti program

$$n = \frac{\text{Performanse\_X}}{\text{Performanse\_Y}} = \frac{\text{Vreme\_izvršenja\_Y}}{\text{Vreme\_izvršenja\_X}}$$

Primer:
- Program na mašini A se izvršava za T_A =1 sec
- Program na mašini B se izvršava za T_B = 10 sec

- Performanse_A/Performanse_B= T_B / T_A=10
- ⇒ Mašina A ima 10 puta bolje performanse od mašine B

Merenje vremena u višekorisničkom režimu

- Računar često deli više korisnika, tj. izvršava više programa (recimo serveri)
  - U takvim slučajevima je mnogo bitnije da se poveća propusnost sistema (obavljeni obim posla u jedinici vremena), nego da se minimizira izvršenje jednog programa
    - za procenu ovih performansi koristi se aritmetička i težinska aritmetička sredina vremena izvršenja

$$T_{srednje} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

gde je Ti vreme izvršenja i-tog programa, a *n* broj programa koji se izvršava

$$T_{tezinsko} = \frac{\sum_{i=1}^{m} w_i T_i}{\sum_{i=1}^{m} w_i}$$

gde je w_i težina, tj. broj izvršavanja *i*-tog programa , a *m* broj različitih programa

Primer

Programi P1 i P2 izvršavaju se na tri računara A,B i C. Vremena izvršenja programa prikazana su u tabeli.
Odrediti
–ukupno vreme izvršenja ovih programa na računarima A, B i C
–aritmetičku sredinu vremena izvršenja
–težinsku sredinu vremena izvršenja ako se program P1 izvršava 20 puta, P2 jednom
–težinsku sredinu vremena izvršenja ako se program P1 izvrašva 20 puta a P2 10 puta

		Računar A	Računar B	Računar C
1	Program P1 (sec)	5	10	20
2	Program P2 (sec)	200	100	20
3	Ukupno vreme	205	110	40
4	Aritmetička sredina (Tsrednje)	102.5	55	20
5	Težinska aritm. sredina (20,1)	14.29	14.29	20
6	Težinska aritm. sredina (20,10)	70	40	20

Procesorske performanse

- Procesori današnjih računara se pobudjuju frekvencijom fiksne učestalosti
  - $f_{clk}=1/T_{clk}$, T_{clk} dužina trajanja taktnog intervala
- Procesorsko vreme
  - $T_{CPU} = N_{CPU} * T_{clk} = N_{CPU} / f_{clk}$

broj CPU-ovih taktnih impulsa

- očigledno je da se performanse mogu poboljšati
  - smanjenjem broja klok ciklusa
  - povećanjem fekvencije
  - Projektanti hardvera često moraju da prave kompromis između ova dva

Primer

- Program se na računaru A, koji radi na 4GHz, izvršava za 10sec.
  - Projektant pokušava da projektuje računar B koji će isti program izvršavati za 6 sec.
  - Da bi se to postiglo potrebno je povećati frekvenciju, ali povećanje frekvencije utiče na ostale delove CPU, uzrokujući da računar B zahteva 1.2 puta klok ciklusa u odnosu na računar A.
  - Kolika treba da bude frekvencija računara B?

Odgovor

$$T_{CPU_A} = \frac{N_{CPU_A}}{f_{clk_A}} \quad 10\,\text{sec} = \frac{N_{CPU_A}}{4\times10^9} \Rightarrow N_{CPU_A} = 10\times4\times10^9 = 40\times10^9 \ \textit{ciklusa}$$

$$T_{CPU_B} = \frac{1.2\times N_{CPU_A}}{f_{clk_B}} \quad 6\,\text{sec} = \frac{1.2\times40\times10^9}{f_{clk_B}} \Rightarrow f_{clk_B} = \frac{1.2\times40\times10^9}{6} = 8\times10^9\,\frac{1}{\text{sec}} = 8GHz$$

Performanse CPU

- Od čega zavisi N_{CPU} ?
  - od dužine programa, tj. broja mašinskih instrukcija koje treba izvršiti
    - Jedna mašinska instrukcija se može izvršavati za 1 ili više clk ciklusa
  - Od skupa instrukcija date arhitekture (ISA)

- Poželjno je poznavati srednji broj taktova po instrukciji – CPI

$$T_{cpu} = N_{\Sigma I} \times CPI \times T_{clk} \quad \text{[sec/program]}$$

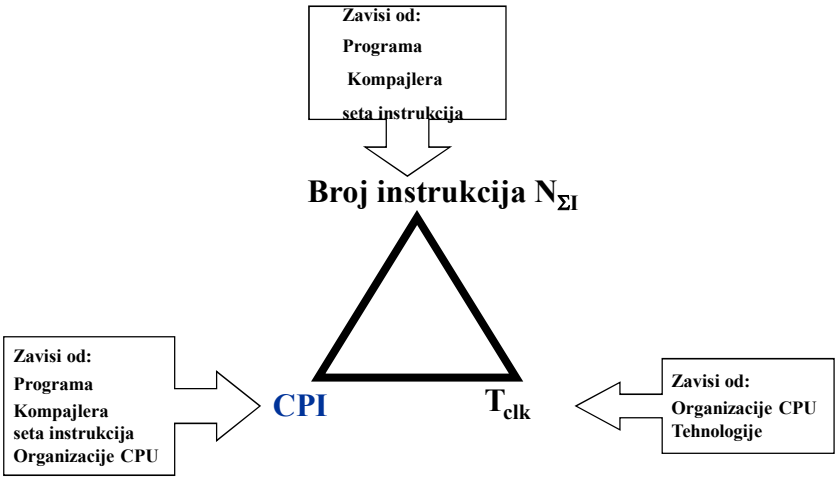
Procesorske performanse

- Kako odrediti srednji broj taktova po instrukciji za dati set instrukcija?
  - Praćenjem velikih programa u toku dužeg vremena
    - Ako učestalost (verovatnoća) pojavljivanja instrukcije tipa I_i iznosi p_i, i ako njeno izvršenje zahteva t_i procesorskih taktnih impulsa

$$CPI = \frac{1}{N}\sum_{i=1}^{N} p_i t_i, \quad N \text{ je broj instrukcija u setu instrukcija}$$

$$CPI = \frac{N_{cpu}}{N_{\Sigma I}}, \quad N_{\Sigma I} \text{ broj instrukcija u programu}$$

$$T_{cpu} = N_{\Sigma I} \times CPI \times T_{clk}$$



Primer

- Program se izvršava na odredjenoj mašini sa sledećim karakteristikama:
  - Ukupan broj instrukcija u programu: 10,000,000
  - Srednji broj taktova po instrukciji, CPI : 2.5 clk/instrukciji.
  - Radna frekvencija CPU : 200 MHz.
- Koliko je vreme izvršenje ovog programa?

$$T_{cpu} = N_{\Sigma I} \times CPI \times T_{clk} \quad \text{[sec/program]}$$

Tcpu = Broj instrukcija x CPI x Tclk
= 10,000,000 x 2.5 x $1 / f_{clk}$
= 10,000,000 x 2.5 x 5×10^{-9}
= 0.125 sec

Faktori koji utiču na performanse CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Broj instrukcija	CPI	Clock Cycle C
Program	X	X	
Compiler	X	X	
Instruction Set Architecture (ISA)	X	X	
Organization		X	X
Technology			X

Alternativne mere CPU performansi

- MIPS – Milion instrukcija po sekundi

$$MIPS = \frac{ukupan\_broj\_instrukcija}{T_{cpu} \times 10^6} = \frac{N_{\Sigma I}}{T_{cpu} \times 10^6} = \frac{N_{\Sigma I}}{N_{\Sigma I} \times CPI \times T_{clk} \times 10^6} =$$

$$= \frac{1}{CPI \times T_{clk} \times 10^6} = \frac{f_{clk}}{CPI \times 10^6}$$

- Veći broj MIPS-ova znači brža mašina (uglavnom)

MIPS - problemi

- MIPS zavisi od skupa instrukcija date arhitekture
  - teško je porediti arhitekture sa različitim setom instrukcija
- MIPS se menja u zavisnosti od programa koji se izvršava, čak i na istoj mašini
- Veći broj MIPS-ova u nekim slučajevima ne mora značiti bolje performanse
- MIPS može dati lošije rezultate za mašinu koja brže radi

Primer:

- Mašina poseduje sledeće klase instrukcija:

Klase instrukcija	CPI
A	1
B	2
C	3

- Za dati program dava komajlera generišu sledeći broj instrukcija:

	Broj instrukcija (u milionima) za svaku klasu instrukcija		
Kod iz:	A	B	C
Comp 1	5	1	1
Comp 2	10	1	1

- Mašina radi na učestalosti od 100 MHz

Primer (nastavak)

MIPS = f_{clk} / (CPI x 10^6) = 100 MHz / (CPI x 10^6)

CPI = N_{CPU} / Broj_instrukcjija

T_{cpu} = Broj_instrukcija x CPI / f_{clk}

- Za kompajler 1:
  - CPI_1 = (5 x 1 + 1 x 2 + 1 x 3) / (5 + 1 + 1) = 10 / 7 = 1.43
  - $MIPS_1$ = (100 *10^6)/ (1.428 x 10^6) = **70.0**
  - T_{cpu1} = ((5 + 1 + 1) x 1.43) / (100 x 10^6) = **0.10** msec
- Za kompajler 2:
  - CPI_2 = (10 x 1 + 1 x 2 + 1 x 3) / (10 + 1 + 1) = 15 / 12 = 1.25
  - $MIPS_2$ = (100 *10^6)/ (1.25 x 10^6) = **80.0**
  - T_{cpu2} = ((10 + 1 + 1) x 1.25) / (100 x 10^6) = **0.15** msec

Rešenje

- Da bi se otklonile anomalije koristi se relativni MIPS
  - Relativni MIPS =$T_{ref}/T_{oč}$ *$MIPS_{ref}$
    - T_{ref} - vreme izvršenja programa na referentnoj mašini
    - $T_{oč}$ - vreme izvršenja programa na mašini čije se performanse procenjuju
    - $MIPS_{ref}$ - broj MIPSova referentne mašine (VAX 11/780 1 MIPS mašina)
  - koristiti iste kompajlere

MFLOPS - Million FLOating-Point Operations Per Second

- $MFLOPS = \frac{\text{broj FP operacija u programu}}{T_{cpu} \times 10^6}$

- MFLOPS je mera zavisna i od mašine i od programa
  - Kao mera namenjen je samo za procenu performasi kod izvršenja operacija u pokretnom zarezu i ne sme se primenjivati van tog konteksta
    - Npr. kod TEX programa broj MFLOPS teži 0, bez obzira koliko je brza mašina (TEX programi ne koriste operacije u pokretnom zarezu)

Ubrzanje sistema

- Kod arhitektura kod kojih je uveden bilo koji vid poboljšanja, može se dati ocena o dobijenom poboljšanju sa stanovišta performansi korišćenjem mere UBRZANJE
  - $S = \frac{\text{Vreme izvršenja programa na arhitetkuri bez poboljšanja}}{\text{Vreme izvršenja programa na arh. sa izvedenim poboljšanjem}}$

Ako se poboljšanje može izvršiti samo na delu sistema, procena ukupnog poboljšana sa stanovišta ubrzanja se može dobiti kao

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

$Speedup_{overall}$ oznacava ukupno ubrzanje
$Fraction_{enhanced}$ oznacava deo nad kojim je izvrseno poboljsanje
$Speedup_{enhanced}$ oznacava koliko je ubrzanje poboljsanog dela

Primer 2

- Novi web-server ima CPU koji je 10 puta brzi od CPUa prethodnog web-servera. Preformanse U/I podsistema nisu poboljšane. Web-server provodi 40% vremena u izračunavanju a 60% u U/I aktivnostima. Koliko je brži novi web-server od prethodnog?

$$Fraction_{enh} = 0.4$$

$$Speedup_{enh} = 10$$

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enh}) + \frac{Fraction_{enh}}{Speedup_{enh}}} = \frac{1}{(1-0.4) + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

Primer

- Dve grupe inženjera dobile su zadatak da poboljšaju performanse nekog sistema iz proizvodnog programa preduzeća. Po isteku dobijenog vremena, prva grupa je najavila poboljšanje, koje aktivnosti sistema sa učešćem od 5% u ukupnim aktivnostima sistema ubrzava 20 puta. Druga grupa je najavila svoje rešenje, koje aktivnosti sistema sa učešćem od 50% u ukupnim aktivnostima sistema ubrzava 2 puta. Čije rešenje daje veće poboljšanje performansi sistema?

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

$$I \ \ grupa \qquad \frac{1}{1 - 0.05 + \frac{0.05}{20}} = \frac{1}{0.95 + 0.0025} = 1{,}05$$

$$II \ \ grupa \qquad \frac{1}{1 - 0.5 + \frac{0.5}{2}} = \frac{1}{0.5 + 0.25} = 1{,}33$$

Rešenje II grupe obezbeđuje mnogo veće poboljšanje performansi .

Kako meriti performanse celog sistema?

- Mnogi performanse računarskog sistema (pogrešno) vezuju za brzinu CPU
  - Mere za ocenu performansi CPU uključuju taknu frekvenciju i MIPS
    - Reći da je sistem A brži od sistema B zato što sistem A radi frekvenciji 1.4 GHz a sistem B na 900 MHz, može biti korektno samo ako su skupovi instrukcija oba procesora identični.
  - Sa različitim skupovima instrukcija, moguće je da oba sistema daju iste rezultate, tj. da izvrše neki program za isto vreme.
- Da bi se performanse sistema merile nezavisno od taktne frekvencije i skupa instrukcija koriste se tvz. benchmark programi.

Šta je Benchmark?

- **Benchmark** je standardna mera ili ocena nečega (Webster's II Dictionary).
  - U računrskoj tehnici se pod benchmark-om obično podrazumeva skup reprezentativnih programa za ocenu performansi računara koji generišu relativnu sliku o performansama sistema
    - Računarski benchmark obično meri brzinu – koliko brzo se program izvršava, ili propusnost – koliko posla u jedinici vremena je izvršeno.
- Izvršenje istog benchmark programa na više računara nam omogućava da napravimo komparaciju.

Benchmark

- Koji se programi mogu iskoristiti kao benchmark?
  - Realni programi koji se izvršavaju na ciljnoj arhitekturi
    - veoma specifični, nisu portabilni, kompleksni: teško je porediti performanse različitih mašina
  - predložen je veći broj tvz. Sintetičkih benchmark programa.
    - Ovi programi ne obavljaju nikakav realan posao; njihova jedina svrha je da ocene performanse (generišu neki broj)
      - Prvi sintetički benchmark programi, npr. Whetstone, Dhrystone, Linpak, su bili relativno kratki programi koji su se lako mogli optimizovati, što je pružalo prostor za zloupotrebu
      - Ovi programi su i suviše mali da bi mogli da procene performanse današnjih sistema
  - Jezgra (kernels)
    - ključni delovi realnih programa sa intenzivnim izračunavanjima
      - » Primeri: Matrix factorization, FFT, tree search, etc.
    - koriste se za testiranje specifičnih aspekata mašine.

Benchmark

- Skup (miks) realnih programa koji su tipični za ciljanu aplikaciju ili opterećenje
  - 1988 formirana je neprofitabilna korporacija pod nazivom Standard Performance Evaluation Corporation (SPEC) sa ciljem da obezbedi objektivnu procenu performansi
    - SPEC obezbeđuje različite skupove benhmark programa za različite klase računara i aplikacija
    - Njihov najpoznatiji proizvod je SPEC CPU
    - SPEC CPU2006 benchmark se sastoji od dva dela
      - CINT2006 koji meri performanse CPU kod izvršavanja integer operacija (12 aplikacija pisanih na C i C++)
      - CFP2006 koji meri performanse CPU kod izvršavanja FP operacija (17 aplikacija pisanih na Fortran-u, C i C++)
  - Ovi programi predstavljaju ključne delove realnih programa pri rešavanju određenih problema, pri čemu su uklonjeni delovi programa kao što su U/I aktivnosti
  - Na većini sistema potrebno je više od 24h da bi se izvršio SPEC CPU2006.

- Po okončanju programa, vreme izvršenja svakog kernela se deli sa vremenom izvršenja istog kernela na Sun Ultra Enterprise 2 workstation.
- Konačni rezultat predstavlja geometrijsku sredinu svih vremena izvršenja.

* Proizvođači mogu da navedu dve vrste rezulata:
  - Vršne performanse dobijene kompajlerskim optimizacijama
  - Osnovne performanse, dobijene bez kompajlerskih optimizacija.

Tipovi benchmarka

Za		Protiv
• Reprezentativi	Realni programi	• veoma specifični. • nisu portabilni. • Kompleksni
• Portabilni. • široka primena. • realne ocene	Miks realnih programa	• manje reprezentativi.
• Lako se izvršavaju, u ranoj fazi projektovanja sistema	"Kernel" Benchmarks	• Lako je "prevariti" tj. projektovati hw tako da se dobiju dobre ocene
• Identifikuju vršne performanse i potencijalna uska grla	Sintetički	• Dobijene vršne performanse mogu biti mnogo bolje od onih dobijenih realnim aplikacijama