KATEDRA ZA ELEKTRONIKU I DIGITALNE SISTEME

ARHITEKTURA I ORGANIZACIJA NAMENSKIH RAČUNARA

HARIS TURKMANOVIĆ - VEŽBE 2026/27 - ČAS 11

PARAMETRI PERFORMANSI MEMORIJSKOG DELA SISTEMA

- HIJERARHIJSKA ORGANIZACIJA MEMORIJE
- PARAMETRI PERFORMANSI HIJERARHIJSKI ORGANIZOVANE MEMORIJE

Parametri performansi memorijskog dela sistema

Gde je problem?

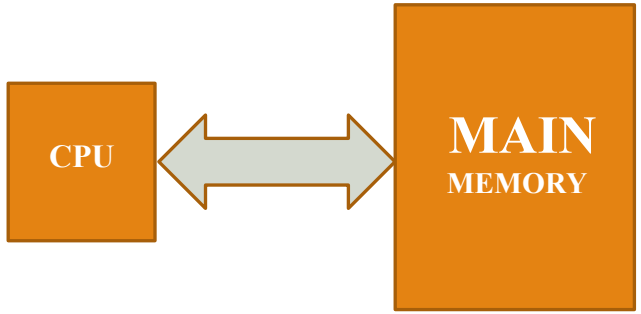
I	Operacija na magistrali	Podatak	Adresa
1	Read	*WaitForData*	0x00000000
2	Read	0x02000093	0x00000000
7	Read	*WaitForData*	0x00000004
8	Read	0xC0000137	0x00000004
12	Read	*WaitForData*	0x00000008

Parametri performansi memorijskog dela sistema

Kako poboljšati performanse?

$$CPU_{Time} = \downarrow \text{Clock Cycles For Program Execution} \times \downarrow T_{\text{Clock Cycle}}$$

- Prvi činilac dominantan
  - Trend rasta brzine CPUa nije propraćen proporcionalnim rastom performansi pristupa memorijama koje se koriste za skladištenje operanada i instrukcija.
    - Instrukcije se skladište u neki tip *non-volatile* memorije
      - *Flash, EEPROM*
      - *Spore, veliki kapacitet, mala gustina pakovanja, ...*
    - Operandi se skladište u neki tip *RAM* memorije
      - *SRAM*
        Brz, malog kapaciteta, mala gustina pakovanja, ...
      - *DRAM*
        Spor, velikog kapaciteta, velika gustina pakovanja, ...



- Da bi poboljšali ukupne performanse izvršavanja softvera neophodno je obezbediti da vreme pristupa podatku (instrukciji ili operandu) bude što je moguće manji.

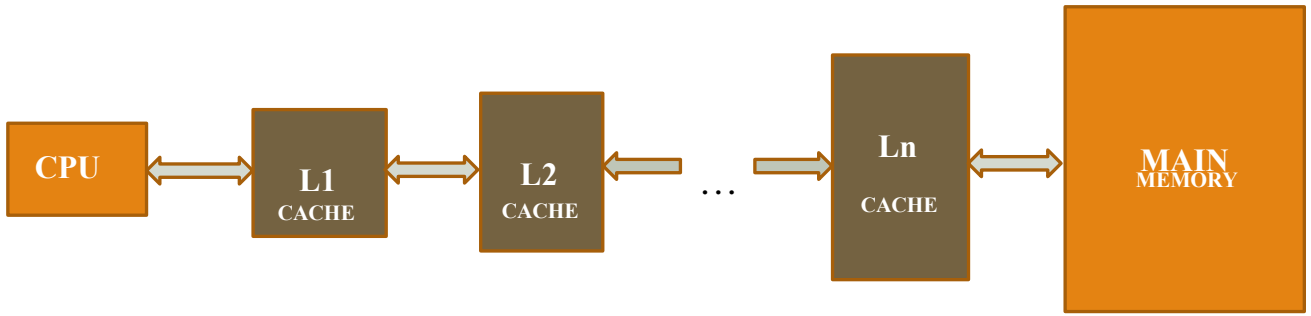
  ... *smanjenjem vremena pristupa ne smemo degradirati ukupan kapacitet memorije u sistemu*

Parametri performansi memorijskog dela sistema

VREME PRISTUPA JEDNAKO NAJBRŽOJ MEMORIJI U SISTEMU

KAPACITET MEMORIJE NEPROMENJEN

HIJERARHIJSKA ORGANIZACIJA MEMORIJE



Unapređenje performansi CPU-a

Hijerarhijski organizovana memorija

- Koncept hijerarhijske organizacije memorije baziran je na dve pretpostavke

1. Kada se neki podatak koristi verovatno je da će se koristiti i podatak koji mu je susedan

→ ***Prostorni lokalitet***

2. Kada se neki podatak koristi sa velikom verovatnoćom se predpostavlja da će se taj podatak ponovo koristiti u skorijoj budućnosti

→ ***Vremenski lokalitet***

Unapređenje performansi CPU-a

Hijerarhijski organizovana memorija – ciklusi na magistrali

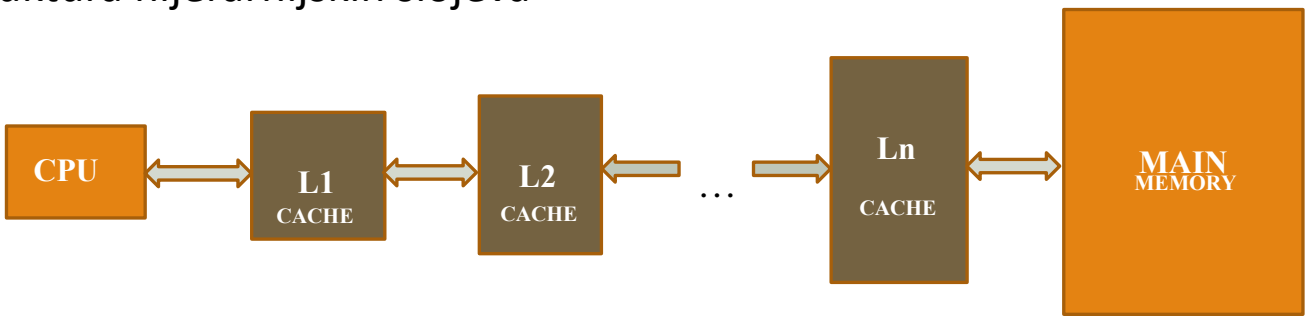
- U slučaju hijerarhijski organizovane memorije, u opštem slučaju, CPU realizuje sledeći set koraka kako bi pročitao neki podatak (instrukcija/operand)

1. CPU realizuje proces čitanja na magistrali sa ciljem dohvatanja podatka
2. Najviši hijerarhijski sloj memorije proverava da li on sadrži traženi podatak

   2.1. Ukoliko se podatak nalazi u najvišem hijerarhijskom sloju podatak se dostavlja procesoru ← ***HIT Scenario***

   2.2. Ukoliko se podatak ne nalazi u najvišem hijerarhijskom sloju, blok sa podatkom se iz nižeg hijerarhijskog sloja prebacuje u viši hijerarhijski sloj pa se konačni podatak dostavlja procesoru ← ***MISS Scenario***

3. Ažurira se struktura hijerarhijskih slojeva



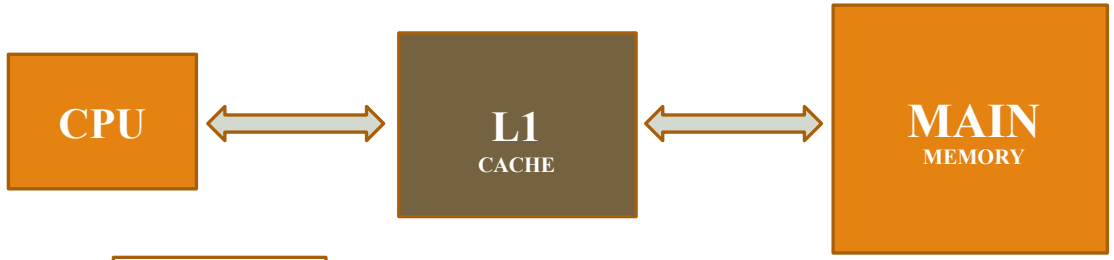
PARAMETRI PERFORMANSI MEMORIJSKOG DELA SISTEMA

- HIJERARHIJSKA ORGANIZACIJA MEMORIJE
- PARAMETRI PERFORMANSI HIJERARHIJSKI ORGANIZOVANE MEMORIJE

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju jednog hijerarhijskog nivoa

- Prilikom dizajniranja hijerarhijski organizovane memorije potrebno je na neki način izvršiti poređenje određenih realizacija memorijskog sistema.
- Osnovni parametak koji dizajneri koriste kako bi kvantifikovali performanse hijerarhijski organizovane memorije je srednje vreme pristupa memoriji – **AMAT** (*eng. **A**verage **M**emory **A**ccess **T**ime*)



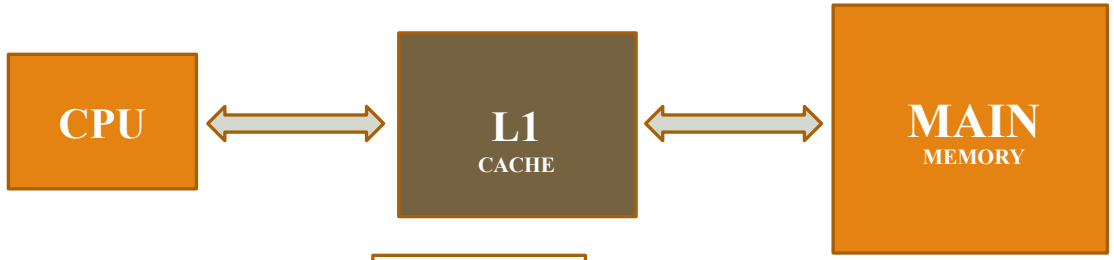
$$AMAT = HitRate \times HitTime + MissRate \times MissTime$$

[%] – procenat memorijskih pristupa podacima dostupnim i višem sloju hijerarhijske organizacije memorije (u nastavku ćemo ovaj parametar označavati sa r_{Hit}). Ovaj parametar se definiše kao *Broj uspešnih pristupa višem hijerarhijskom novu/Broj svih memorijskih pristupa*

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju jednog hijerarhijskog nivoa

- Prilikom dizajniranja hijerarhijski organizovane memorije potrebno je na neki način izvršiti poređenje određenih realizacija memorijskog sistema.
- Osnovni parametak koji dizajneri koriste kako bi kvantifikovali performanse hijerarhijski organizovane memorije je srednje vreme pristupa memoriji – **AMAT** (*eng. **A**verage **M**emory **A**ccess **T**ime*)



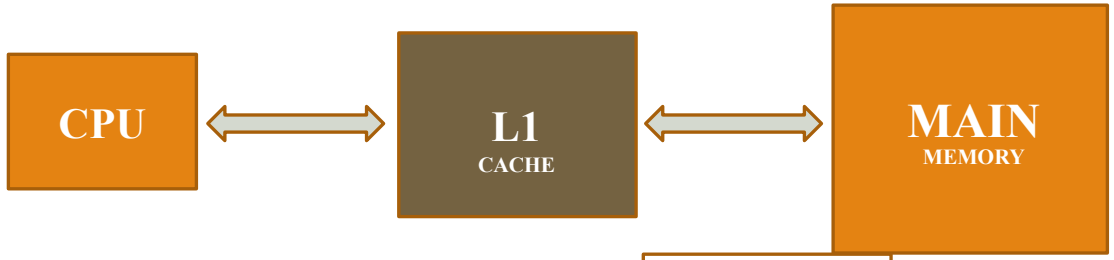
$$AMAT = HitRate \times HitTime + MissRate \times MissTime$$

[taktni ciklus] – ukupno vreme pristupa višem sloju hijerarhije. Potrebno je napomenuti da ovo vreme obuhvata vreme potrebno da se dekoduje da li trenutni hijerarhijski sloj sadrži podatak i vreme potrebno da se pristupi podatku ukoliko sloj hijerarhije sadrži podataka (u nastavku ćemo ovaj parametar označavati sa T_{Hit})

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju jednog hijerarhijskog nivoa

- Prilikom dizajniranja hijerarhijski organizovane memorije potrebno je na neki način izvršiti poređenje određenih realizacija memorijskog sistema.
- Osnovni parametak koji dizajneri koriste kako bi kvantifikovali performanse hijerarhijski organizovane memorije je srednje vreme pristupa memoriji – **AMAT** (*eng. **A**verage **M**emory **A**ccess **T**ime*)



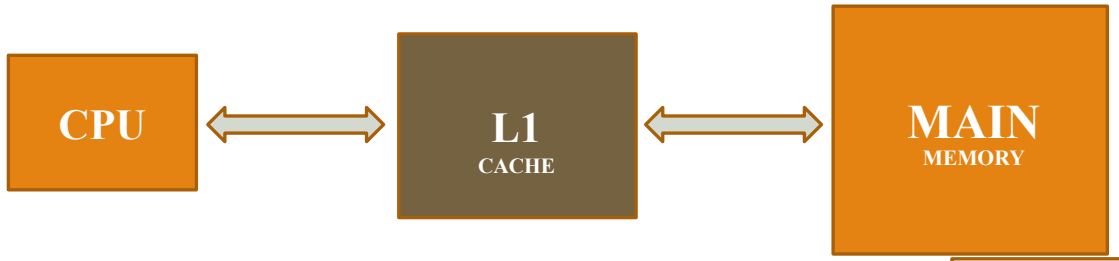
$$AMAT = HitRate \times HitTime + MissRate \times MissTime$$

[%] – Definišemo kao 1- r_{Hit} (u nastavku ćemo ovaj parametar označavati sa r_{Miss}) ili broj neuspešnih pristupa višem hijerarhijskom nivou/Broj svih memorijskih pristupa

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju jednog hijerarhijskog nivoa

- Prilikom dizajniranja hijerarhijski organizovane memorije potrebno je na neki način izvršiti poređenje određenih realizacija memorijskog sistema.
- Osnovni parametak koji dizajneri koriste kako bi kvantifikovali performanse hijerarhijski organizovane memorije je srednje vreme pristupa memoriji – **AMAT** (*eng. **A**verage **M**emory **A**ccess **T**ime*)



$$AMAT = HitRate \times HitTime + MissRate \times MissTime$$

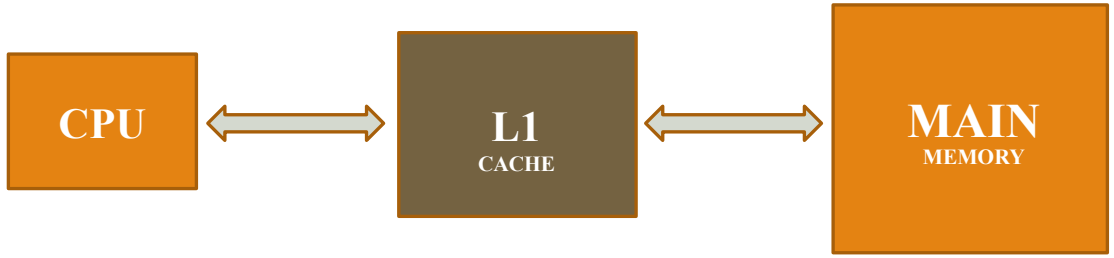
[taktni ciklus] – u slučaju da podatak ne postoji trenutnom sloju hijerarhije predstavlja vreme potrebno da se detektuje da li podatak postoji ili ne i vreme potrebno da se blok sa podatkom premesti iz nišeg sloja u trenutni sloj. Vrednost ovog parametra određena je izrazom:

$$T_{Miss} = T_{Hit} + T_{Penalty}$$

vreme potrebno da se podatak iz nižeg sloja premesti u viši hijerarhijski sloj (u nastavku ćemo ovaj parametar označavati sa $T_{Penalty}$)

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju jednog hijerarhijskog nivoa



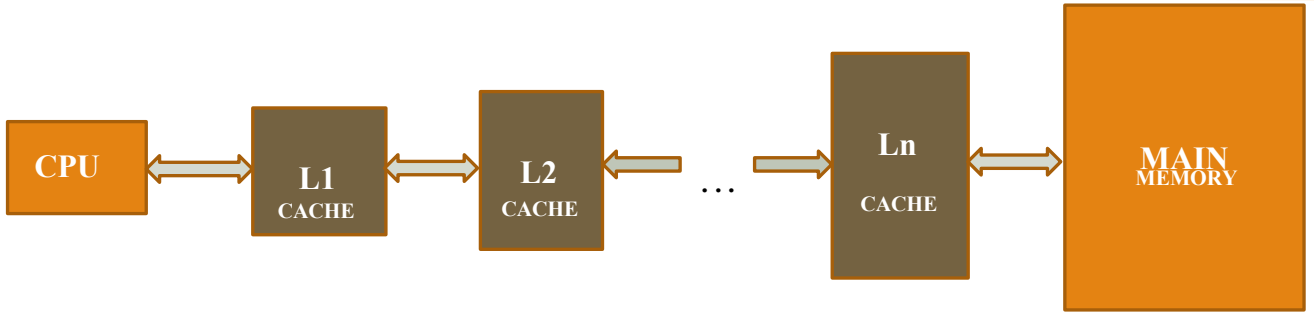
- Od ukupnog broja memorijskih pristupa ***M*** određeni deo pristupa memoriji uspešno realizuje na najvišem hijerarhijskom sloju (r_{Hit} deo pristupa) a jedan deo se realizuje na glavnoj programskoj memoriji (r_{Miss} deo)

$$AMAT = HitRate \times HitTime + MissRate \times MissTime$$

$$\mathbf{AMAT = T_{Hit} + r_{Miss}\ T_{Penalty}}$$

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju više hijerarhijskih nivoa

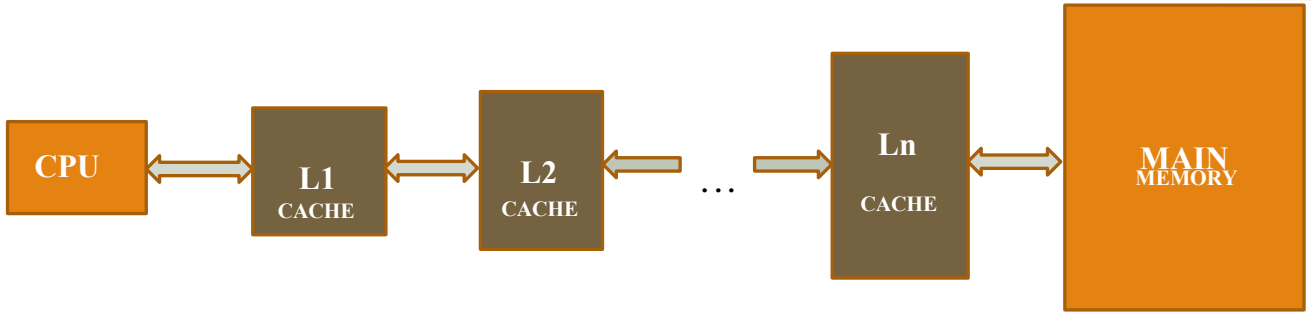


Local Miss Rate → Ukupan broj *miss*-eva podeljen sa ukupnim brojem pristupa kešu na istom hijerarhijskom nivou (u nastavku ćemo ovaj parametar označavati sa r_{Miss})

Global Miss Rate → Ukupan broj *miss*-eva podeljen sa ukupnim brojem memorijskih pristupa (u nastavku ćemo ovaj parametar označavati sa $r_{MissGlobal}$)

Hijerarhijski organizovana memorija

Parametri performansi – **AMAT** u slučaju više hijerarhijskih nivoa



$$AMAT = T_{Hit\text{-}i} + r_{Miss\text{-}i}\, T_{Penalty\text{-}i}$$

Hijerarhijski organizovana memorija

PRIMERI – Zadatak 3

3. Organizacija memorijskog dela jednog namenskog računarskog sistema prikazana je na slici 2.3. Parametri memorijskih komponenti dati su u tabeli 2.1. Odrediti srednje vreme pristupa memoriji.

Tabela 2.1 - Tabela uz zadatak 3

Naziv Memorije	Parametri		
	T_H[cycles-c]	r_H[%]	r_M[%]
L1 Cache	1	95	5
SDRAM	100	100	0

Slika 2.3 - Slika uz zadatak 3

Rešenje

Hijerarhijski organizovana memorija

PRIMERI – Zadatak 4

4. Memorija embedded platforme je hijerarhijski organizovana sa dva nivoa keširanja podataka L1 i L2. Ako je poznato da se od 100 zahteva CPUa za pristup podatku 80 zahteva uspešno realizuje na nivou L1 i da je rHit-2 = 75% odrediti parametre rHit-1, rMiss-1, rMissGlobal-1, rMiss-2, rMissGlobal-2.

Rešenje

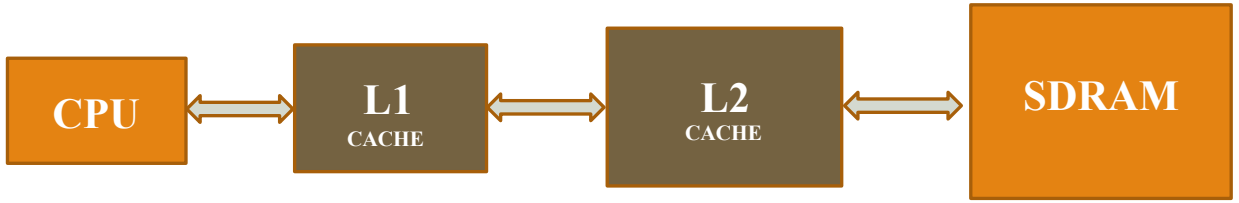
Hijerarhijski organizovana memorija

PRIMERI – Zadatak 5

5. Organizacija memorijskog dela jednog namenskog računarskog sistema prikazana je na slici 2.4. Izračunati srednje vreme pristupa memoriji ako su parametri korišćenih memorija dati u tabeli 2.2.

Tabela 2.2 - Tabela uz zadatak 5

Naziv Memorije	Parametri			
	T_H[cycles-c]	r_H[%]	r_{ML}[%]	r_{MG}[%]
L1 Cache	1	95		5
L2 Cache	3	75	25	
SDRAM	100	100	0	0



Slika 2.4 – Slika uz zadatak 5.

Rešenje

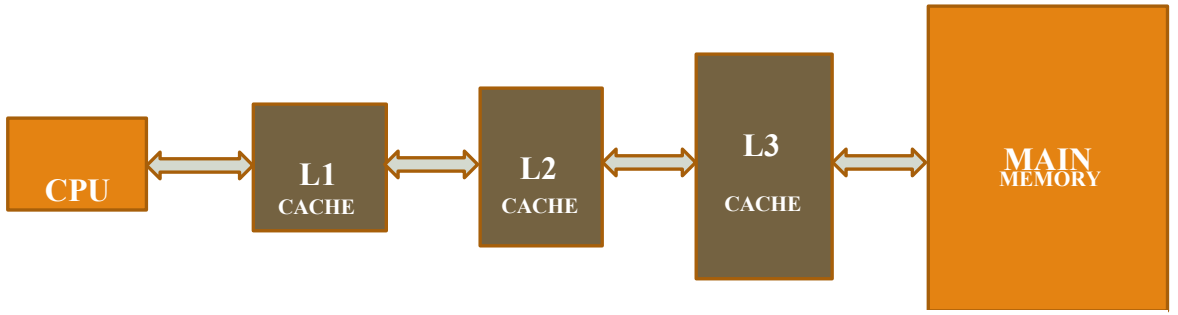
Hijerarhijski organizovana memorija

PRIMERI – Zadatak 6

6. Organizacija memorijskog dela jednog namenskog računarskog sistema prikazana je na slici 2.5. Izračunati srednje vreme pristupa memoriji i dopuniti tabelu 2.3.

Tabela 2.3 - Tabela uz zadatak 6

Naziv Memorije	Parametri			
	T_H[cycles-c]	r_H[%]	r_{ML}[%]	r_{MG}[%]
L1 Cache	1	70		
L2 Cache	3			6
L3 Cache	10	95		
SDRAM	100	100	0	0



Slika 2.5 Slika uz zadatak 6

Rešenje

PARAMETRI PERFORMANSI CPUA

- OSNOVNI PARAMETRI
- PARAMETRI PERFORMANSI U SLUČAJU HIJERARHIJSKI ORGANIZOVANE MEMORIJE

Parametri performansi CPU-a

OSNOVNI PARAMETRI - *CPU Time*

- Kada govorimo o performansama CPUa razmatraćemo samo one parametre koji se odnose na **performanse izvršavanja programskog koda.**
  - „Brzina“ izvršavanja programskog koda
- Osnovni parametar performansi CPUa označava se kao ***CPU executiom time***
  - Vreme koje CPU potroši izvršavajući određeni **program** **P**

$$CPU_{Time} = \text{Clock Cycles For Program Execution} \times T_{\text{Clock Cycle}}$$

Broj taktnih ciklusa *procesora potrebnih da bi se program* ***P*** *izvršio*

Trajanje taktnog ciklusa *procesora*

  - Program zapravo predstavlja određeni ***broj instrukcija*** **I**

$$CPU_{Time} = I \times CPI \times T_{\text{Clock Cycle}}$$

Ukupan broj instrukcija u okviru programa P

Prosečan broj *taktnih ciklusa po instrukciji*

Parametri performansi CPU-a

OSNOVNI PARAMETRI - *CPI*

- CPI parametar definiše prosečan broj taktnih ciklusa po instrukciji u okviru programa P

$$CPI = \frac{\sum I_i \; CPI_i}{I}$$

Broj instrukcija određenog tipa

Grupa instrukcija koju karakterišu neke zajedničke osobine

- *Load/Store*
- *Instrukcije skoka*
- *Aritmetičke instrukcije*
- *...*

Broj taktnih ciklusa potreban za izvršavanje određenog tipa instrukcija

Ukupan broj instrukcija u okviru programa **P**

- Učestanost pojavljivanja određenog tipa instrukcije

$$F_i = \frac{I_i}{I}$$

ne mešati ovaj parametar sa CPI jer se u literaturi značenje parametra povezuje sa kontekstom u kom se parametar navodi

Parametri performansi CPU-a

OSNOVNI PARAMETRI - Kako se praktično određuju vrednosti?

CPU_{TIME} → Merimo vreme potrebno za izvršavanje nekog programa P

$T_{Clock\ Cycles}$ → Dato u okviru *datasheet* dokumenta platforme

I → Možemo odrediti koristeći različite alate

Dissasembly, Map file, ...?

CPI → Ako znamo kako da odredimo prethodna tri, lako možemo izračunati i ovaj parametar primenom prethodno navedenih formula

Parametri performansi CPU-a

PRIMERI – Zadatak 1

Pretpostavimo da imaju dve implementacije A i B iste arhitekture instrukcijskog seta. Za CPU A poznato je da vrednosti CPI parametra iznosi 2 dok je radni takt CPUa 200Mhz. Za CPU B poznato je da vrednosti CPI parametra iznosi 1.5 dok je radni takt CPUa 100Mhz. Na kojoj od ove dve realizacije će biti potrebno manje vremena da se izvrši program koji ima *I* instrukcija?

Rešenje

Parametri performansi CPU-a

PRIMERI – Zadatak 2

Za određenu hardversku implementaciju CPUa u tabeli je data je vrednost CPI parametra za određenu grupu (***tip***) instrukcija:

	A	*B*	*C*
CPI	1	2	3

Za prevođenja programa P iz višeg programskog jezika korišćena su dva kompajlera C1 i C2. Rezultati prevođenja programa za svaki od kompajlera sumirani su u tabeli

	Broj instrukcija po klasama instrukcija		
	A	*B*	*C*
C1	2	1	2
C2	4	1	1

Koji od kompajlera kao rezultat kompajliranja daje manje instrukcija? Koji kompajler generiše „brži kod“? Koliko iznosi vrednost CPI parametra u oba slučaja?

Napomena: Radni takt CPUa iznosi 50MHz.

Parametri performansi CPU-a

PRIMERI – Zadatak 2

	A	*B*	*C*
CPI	1	2	3

	Broj instrukcija po klasama instrukcija		
	A	*B*	*C*
C1	2	1	2
C2	4	1	1

Rešenje

PARAMETRI PERFORMANSI CPUA

- OSNOVNI PARAMETRI
- PARAMETRI PERFORMANSI U SLUČAJU HIJERARHIJSKI ORGANIZOVANE MEMORIJE

Hijerarhijski organizovana memorija

Parametri performansi CPUa – Kako kvantifikovati performanse?

$$CPU_{Time} = \boxed{Clock\ Cycles_{\ For\ Program\ Execution}} \times T_{Clock\ Cycle}$$

$$Clock\ Cycles_{\ For\ Program} = Clock\ Cycles_{\ For\ Program\ Execution} + \boxed{Clock\ Cycles_{\ Waiting}}$$

$$Clock\ Cycles_{\ Waiting} = Clock\ Cycles_{\ Waiting\ Instruction} + Clock\ Cycles_{\ Waiting\ Operands}$$

- Ako je poznato da program ima **I** instrukcija, od kojih $\mathbf{F_i}$ zahteva pristup memoriji za podatke i da je učestanost neuspelih pristupa memoriji za instrukcije $\mathbf{r_{Miss\text{-}instruction}}$ i podatke $\mathbf{r_{Miss\text{-}data}}$, tada možemo napisati:

$$Clock\ Cycles_{\ Waiting} = I \times r_{Miss\text{-}instruction} \times \boxed{Clock\ Cycles_{To\ Acces\ Instruction\ Memory}} + I \times F_i \times r_{Miss\text{-}data} \times \boxed{Clock\ Cycles_{To\ Access\ Data\ Memory}}$$

$$T_{penalty\text{-}instruction} \qquad T_{penalty\text{-}data}$$

Hijerarhijski organizovana memorija

Parametri performansi CPUa – Kako kvantifikovati performanse?

$$CPU_{Time} = Clock\ Cycles_{For\ Program\ Execution} \times T_{Clock\ Cycle}$$

$$Clock\ Cycles_{For\ Program} = Clock\ Cycles_{For\ Program\ Execution} + Clock\ Cycles_{Waiting}$$

$$Clock\ Cycles_{Waiting} = Clock\ Cycles_{Waiting\ Instruction} + Clock\ Cycles_{Waiting\ Operands}$$

- Ako je poznato da program ima **I** instrukcija, od kojih **F_i** zahteva pristup memoriji za podatke i da je učestanost neuspelih pristupa memoriji za instrukcije **$r_{Miss\text{-}instruction}$** i podatke **$r_{Miss\text{-}data}$**, tada možemo napisati:

$$Clock\ Cycles_{Waiting} = I \times r_{Miss\text{-}instruction} \times T_{Penalty\text{-}instruction} + I \times F_i \times r_{Miss\text{-}data} \times T_{Penalty\text{-}data}$$

Hijerarhijski organizovana memorija

Parametri performansi CPUa – Kako kvantifikovati performanse?

$$CPU_{Time} = (Clock\ Cycles_{Executing} + Clock\ Cycles_{Waiting}) \times T_{CLK}$$

$$CPU_{Time} = (Clock\ Cycles_{Executing} + I \times r_{Miss\text{-}instruction} \times T_{Penalty\text{-}instruction} + I \times F_i \times r_{Miss\text{-}data} \times T_{Penalty\text{-}data}) \times T_{CLK}$$

$$I \times CPI_{IDEAL}$$ →

CPI_{IDEAL} definiše ***prosečan*** broj taktnih ciklusa potreban za izvršavanje instrukcije ukoliko su i instrukcija i podaci dostupni na najvišem hijerarhijskom sloju memorijskog dela sistema.

$$CPU_{Time} = (I \times CPI_{IDEAL} + I \times r_{Miss\text{-}instruction} \times T_{Penalty\text{-}instruction} + I \times F_i \times r_{Miss\text{-}data} \times T_{Penalty\text{-}data}) \times T_{CLK}$$

Hijerarhijski organizovana memorija

PRIMERI – Zadatak 7

3. Organizacija memorije jedne embedded platforme prikazana je na slici 2.2.1. Poznati su sledeći parametri sistema:
   - Radni takt od 200MHz,
   - r_{Miss} iznosi 20% za oba tipa memorija
   - Vreme pristupa memoriji za podatke je 100 taktnih ciklusa dok je vreme pristupa memoriji za instrukcije 200 taktnih ciklusa

Odrediti $\boldsymbol{CPU_{TIME}}$ za programski kod koji ima $\boldsymbol{I}$ instrukcija ako $\boldsymbol{CPI_{IDEAL}}$ iznosi 2.0 a učestanost instrukcija koje pristupaju memoriji za podatke je 40%.

Slika 3.1 - Slika uz zadatak 3

Rešenje

Hijerarhijski organizovana memorija

PRIMERI – Zadatak 8

4. Rešiti problem definisan zadatkom 7. ako se između L1 i SDRAM memorije ubaci još jedan nivo keša L2 i za instrukcije i za podatke ako su poznati parametri L2 keša T_{Hit-2}=2c i r_{Miss-2}=30%.

Rešenje

Hijerarhijski organizovana memorija

PRIMERI – Zadatak 9

5. Rešiti problem definisan zadatkom 8. ako se između L2 i SDRAM memorije ubaci još jedan nivo keša L3 i za instrukcije i za podattke. Parametri L3 keša su: T_{Hit-3}=3c i r_{Miss-3}=35%.

Rešenje