



RAČUNALNIŠKA ARHITEKTURA

7 Merjenje zmogljivosti CPE



7 Merjenje zmogljivosti CPE

- Pogostost izvajanja posameznih vrst ukazov
- Čas izvajanja posameznih vrst ukazov
- CPI – povprečno število urinih period za izvedbo ukaza v določenem programu
- MIPS – povprečno število ukazov (v milijonih), ki se izvedejo v eni sekundi
- CPEčas – čas izvrševanja programa v sekundah
- Benchmark programi
- SPEC



- Zmogljivost računalnikov se tradicionalno meri predvsem z zmogljivostjo CPE.
- Pomnilnik in V/I naprave se pri tem običajno ignorira.
- Opravičilo za to je, da naj bi bila pomnilnik in V/I sistem dovolj zmogljiva, da ne povzročata čakanja CPE.
- V takem primeru lahko zmogljivost računalnika enačimo z zmogljivostjo CPE.



Pogostost izvajanja posameznih vrst ukazov

- Z merjenjem (štetjem) lahko ugotovimo, kako pogosto (odstotek celotnega časa izvajanja programa) se v nekem programu izvajajo posamezne skupine ukazov.
- Statična pogostost - število posameznih vrst ukazov v programu. (V programu je npr. 20 ukazov od tega je 6 ukazov v zanki, ki se izvede 10-krat \Rightarrow vseh ukazov je 20; število ukazov, ki se izvedejo v programu, pa je 74)
- **Dinamična pogostost** - število izvrševanj posameznih vrst ukazov pri izvajanju programa.
- Dinamično pogostost izvajanja ukazov vrste i označimo s p_i in izrazimo v odstotkih celotnega časa izvajanja programa.



- Pogostost izvajanja posameznih vrst ukazov se med programi običajno razlikuje.
- Primer dinamične pogostosti izvajanja posameznih vrst ukazov pri dveh različnih programih P1 in P2:

i	Vrsta ukazov	p_i % za program P1	p_i % za program P2
1	Load/store	40%	25%
2	ALE	45%	35%
3	Kontrolni ukazi	10%	10%
4	V/I ukazi	5%	8%
5	FP ukazi	0	22%
	Vsi ukazi skupaj	100%	100%



Čas izvajanja posameznih vrst ukazov

- Trajanje izvajanja ukaza se običajno podaja v urinih periodah. Tako podatek velja za vse (istovrstne) procesorje z različnimi frekvencami ure.
- Podatek o trajanju posamezne vrste ukaza ima oznako **CPI (Cycles per Instruction)** in je enak številu urinih period, ki so potrebne za izvedbo ukaza.
- Za posamezen ukaz je *CPI* vedno celo število (ukaz se v CPE izvaja vedno celo število urinih period)
- Če *CPI* pomnožimo s časom trajanja urine periode t_{CPE} , dobimo čas izvajanja ukaza v sekundah.



- Primer časov izvajanja ukaza v sekundah pri različnih frekvencah ure, če se ukaz izvaja npr. 5 urinih period $\Rightarrow CPI = 5$

CPI Trajanje ukaza [urine periode]	f_{CPE} [Hz]	t_{CPE} [s]	Trajanje ukaza [s]
5	100 MHz	10 ns	50 ns
5	800 MHz	1,25 ns	6,25 ns
5	2 GHz	0,5 ns	2,5 ns
5	3 GHz	0,33 ns	1,65 ns

$$1 \text{ MHz} = 10^6 \text{ Hz}$$

$$1 \text{ GHz} = 10^9 \text{ Hz}$$

$$1 \text{ ns} = 10^{-9} \text{ s}$$



- Za nek program lahko iz podatkov o:
 - dinamični pogostosti izvajanja posameznih vrst ukazov p_i
 - in številu urinih period CPI_i , potrebnih za izvedbo posamezne vrste ukazov,
- izračunamo povprečno število urinih period za izvedbo enega ukaza.
- **Povprečno število** urinih period CPI , ki so potrebne za izvedbo enega ukaza v določenem programu z n vrstami ukazov:

$$CPI = \sum_{i=1}^n CPI_i \cdot p_i$$



- Če poznamo frekvenco ure f_{CPE} , lahko za določen program izračunamo tudi, koliko ukazov povprečno izvede procesor v sekundi.
- Ker je rezultat velikostnega reda nekaj milijonov (ukazov/sek), ga običajno delimo z 10^6 in dobimo rezultat v milijonih ukazov, ki jih procesor izvede v sekundi.
- **MIPS (Million Instructions Per Second).**

$$MIPS = \frac{f_{CPE}}{CPI \cdot 10^6}$$

$$f_{CPE} = \frac{1}{t_{CPE}} \Rightarrow MIPS = \frac{1}{CPI \cdot t_{CPE} \cdot 10^6}$$



- Primer izračuna *CPI* (povprečno število urinih period za izvedbo ukaza) za pogostost izvajanja ukazov v programu P2:

i	Vrsta ukazov	p_i v % za program P2	CPI_i
1	Load/Store	25	6
2	ALE	35	4
3	Kontrolni ukazi	10	3
4	V/I ukazi	8	7
5	FP ukazi	22	8

$$CPI(P2) = \sum_{i=1}^5 p_i \cdot CPI_i = 0,25 \cdot 6 + 0,35 \cdot 4 + 0,10 \cdot 3 + 0,08 \cdot 7 + 0,22 \cdot 8 = 5,52$$

$CPI(P1) = 4,85$ Podatki o pogostosti posameznih vrst ukazov v programu P1 so iz tabele na [str. 5](#), podatki o trajanju posamezne vrste ukazov pa so v obeh primerih isti, ker se programa izvajata na istem računalniku

- V programu P2 traja izvajanje enega ukaza povprečno 5,52 urine periode (v programu P1 pa povprečno 4,85 periode)



- Vzemimo, da je frekvenca ure procesorja $f_{CPE} = 1,8 \text{ GHz}$, potem lahko izračunamo *MIPS*:

$$MIPS = \frac{1,8 \cdot 10^9}{5,52 \cdot 10^6} = \frac{1800}{5,52} = 326,08$$

- Procesor pri izvajanju tega programa izvede povprečno 326 milijonov ukazov na sekundo.
- Običajno rečemo poenostavljeno, da ima procesor zmogljivost 326 *MIPS*-ov.



- Če smatramo, da sta pomnilnik in V/I sistem dovolj zmogljiva, da ne povzročata čakanja CPE, lahko zmogljivost računalnika enačimo z zmogljivostjo CPE.
- Edino pravo merilo zmogljivosti CPE je **čas izvrševanja programa** (CPEčas) merjen v sekundah/program.
- Če zanemarimo čas za V/I prenose, lahko čas izvajanja programa izenačimo s časom, ki ga potrebuje CPE.



$$CPE_{čas} = \text{število_ukazov} \cdot CPI \cdot t_{CPE}$$

- Zmogljivost CPE (CPE čas) je odvisna od:
 - **števila strojnih ukazov** v katere se prevede določen program,
 - **CPI** povprečnega števila urinih period za izvedbo enega ukaza,
 - urine periode t_{CPE} oziroma frekvence ure ($f_{CPE} = 1 / t_{CPE}$)
- Te tri lastnosti so med seboj odvisne in vplivajo ena na drugo. Kaj vpliva nanje?



- Na urino periodo t_{CPE} ali frekvenco ure vplivajo:
 - Hitrost in število digitalnih vezij, s katerimi je zgrajena CPE
 - Zgradba kontrolne in podatkovne enote

- Na CPI - povprečno število urinih period za izvedbo ukaza - vplivajo:
 - Zgradba kontrolne in podatkovne enote
 - Vrsta in število ukazov procesorja
 - Program



- Na število ukazov, v katere se prevede program, vplivajo:
 - Število in vrsta ukazov procesorja
 - Lastnosti prevajalnika

- Pri primerjavi zmogljivosti računalnikov je treba primerjati vse tri lastnosti (oziroma njihov produkt - $CPE_{čas}$), ne pa samo eno od njih.



- Kljub tej ugotovitvi pa čas za praktično uporabo ni najbolj primeren.
 - Odvisen je od programa
 - Pri istem programu odvisen tudi od vhodnih podatkov

- Vendar so se vsi drugi načini, ki naj bi bili neodvisni od programov in bi služili za primerjavo različnih računalnikov, izkazali za slabše ali celo zavajajoče.



- Za določen program, ki se je prevedel in izvedel na dveh različnih računalnikih R1 in R2, so bili izmerjeni naslednji podatki:

Meritev	Računalnik R1	Računalnik R2
Število strojnih ukazov	10×10^9	8×10^9
Frekvenca ure	4 GHz	4 GHz
CPI	1,0	1,1

- Kateri računalnik ima večji MIPS?
- Kateri računalnik je hitrejši?



- **Frekvenca ure f_{CPE}** je najbolj neustrezna za mera primerjavo hitrosti (zmogljivosti) različnih računalnikov.

- Tudi **parameter MIPS** ni najboljši za primerjavo zmogljivosti različnih računalnikov:
 - Odvisen je od števila in vrste ukazov, zato ni primeren za primerjavo računalnikov z različnim naborom ukazov

 - Odvisen je od programa (od vrste ukazov, ki se izvajajo v programu)



- Večji MIPS ne pomeni vedno bolj zmogljiv računalnik (primerjava računalnika s programsko realizacijo FP ukazov in računalnika s FP enoto ali primerjava RISC in CISC računalnikov)

- MIPS – Meaningless Indication of Processor Speed

- Da bi odpravili te pomanjkljivosti, se je začel uporabljati **relativni MIPS**, kjer se čas izvajanja programa primerja s časom, izmerjenim na referenčnem računalniku.
 - Težave pri izbiri referenčnega računalnika

 - Potrebno meriti čas, kar pri MIPS ni potrebno



- **MFLOPS** (Million Floating Point Operations Per Second)
 - Tudi odvisen od programa
 - Smisel ima samo pri procesorjih, ki imajo operacije v plavajoči vejici
 - Resnična zmogljivost v MFLOPS pogosto samo 10 % teoretične, ki jo navajajo proizvajalci

- Uporaba posebej za merjenje zmogljivosti napisanih programov – **sintetični benchmarki.**



- Benchmark program je program, s katerim želimo primerjati zmogljivosti računalnikov.
 - Whetstone (Algol 60, FORTRAN, Pascal,...)
 - Linpack (FORTRAN)
 - Livermore loops (FORTRAN)
 - Dhrystone (ne vsebuje FP operacij, Ada, Pascal, C)
 - Quicksort (sortiranje)
 - Sieve (iskanje praštevil)
 - Puzzle



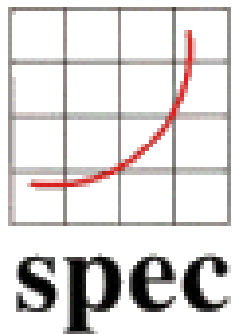
- Benchmark programi so bili znani, zato so proizvajalci računalnikov optimizirali delovanje (prevajalniki) samo za te programe!
- Do sedaj najboljša rešitev je uporaba večjega števila benchmark programov in izračun aritmetične ali geometrijske srednje vrednosti izmerjenih rezultatov.
- **SPEC** (Standard Performance Evaluation Corporation) Leta 1988 več firm ustanovi neprofitno organizacijo SPEC



- Izberejo standardno množico benchmark programov za merjenje zmogljivosti CPE.
 - Prvi nabor programov objavljen leta 1989: SPECmark 89 (aritmetična srednja vrednost razmerja časov 10 programov glede na čas izmerjen na VAX-11/780)
 - Zadnja peta generacija objavljena leta 2006: SPEC CPU2006
 - CINT2006 za celoštevilčne operacije v fiksni vejici – 12 programov,
 - CFP2006 za operacije v plavajoči vejici – 17 programov
 - Referenčni čas za posamezen program (določi ga SPEC) se deli z izmerjenim časom programa, rezultat je »SPEC ratio« posameznega programa ⇒ večji rezultat pomeni boljšo zmogljivost
 - Končni rezultat je geometrična srednja vrednost posameznih zmogljivosti



- To niso več sintetični programi (narejeni posebej za merjenje zmogljivosti), temveč realne aplikacije, ki jih obnavljajo in dopolnjujejo vsakih nekaj let.
- Ti programi omogočajo ovrednotenje zmogljivosti CPE, pomnilniške arhitekture in prevajalnika.



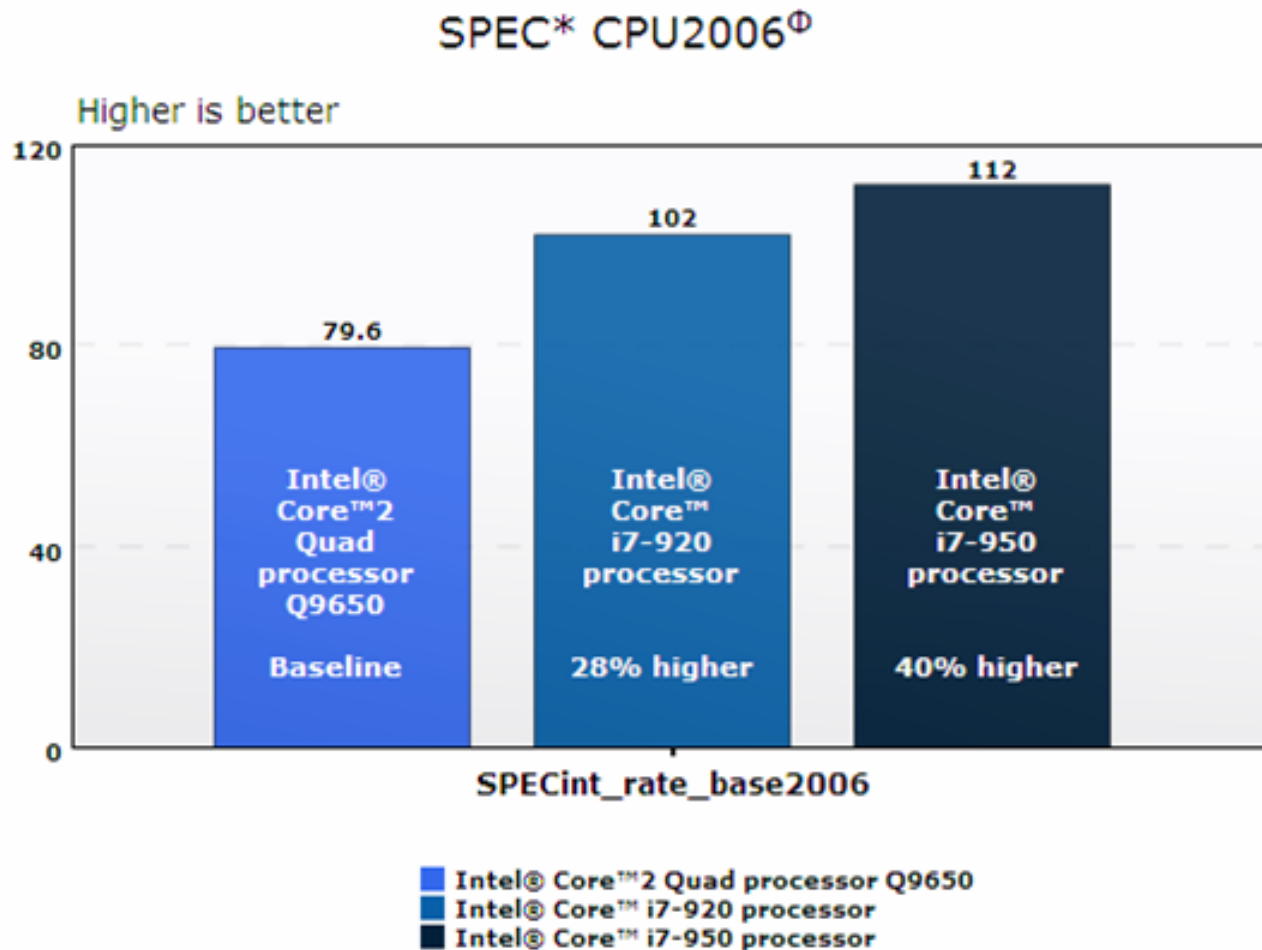
<http://www.spec.org>



- Organizacija SPEC objavlja benchmark programe in rezultate testiranja ne samo za CPE, temveč tudi za:
 - Grafične in delovne postaje
 - Javanske odjemalce in strežnike
 - Visoko zmogljivo računanje (OpenMP, MPI)
 - Poštne strežnike
 - Mrežne datotečne sisteme
 - Merjenje porabe (pri velikem številu strežnikov)
 - SOA (Service Oriented Architectures)
 - SIP strežnike (Session Initiation Protocol)
 - Virtualizacijo
 - Web strežnike



Primerjava zmogljivosti Intelovih procesorjev





<http://www.intel.com/performance>

Benchmark Description:

SPECint_rate_base2006 is a compute-intensive benchmark that measures the integer throughput performance of a computer system carrying out a number of parallel tasks.

The benchmark consists of a suite twelve integer workloads developed from actual end-user applications. System throughput is measured by running multiple copies of the benchmark simultaneously with the number of copies typically set to the number of logical hardware cores seen by the operating system. The performance score reported is a measure of the throughput of the system measured in jobs/hour.