

pomnilnik, naslovni signali: (n=st. bitov)

$1K=2^{10}, 1M=2^{20}, 1G=2^{30}$;
nasl.prostor: 2^n ;
max pomnilnik v $B=2^n$ B;
 $PC=n$ bitov;
povečava nasl. prostora za $X=2^n$ neki,
neki=st.bitov oz. nasl. signalov;

IEEE 754 standard:

stevilo je sestavljeno iz predznaka(s),
eksponenta(exp) in mantise(m), dolgo je 32b;

s=1	exp=8	m=23
-----	-------	------

s=0, ce je st. poz.;

exp=e+127(kolikokrat premaknemo vejico);

m=je oblike 1,???, prvega bita(1) ni treba zapisati;

pretvorba IEEE754 32bit:

$-1^s \cdot m \cdot 2^{\text{exp}}$;

IEEE754 64bit:

s=1	exp=11	m=52
-----	--------	------

exp=e+1023;

Pretvarjanje števil(2),(10),(16):

$0110(2)=0^8+1^4+1^2+0^1=6(10)$;

$100(10)=X(2) \rightarrow$ deljenje z 2, ostanki so rezultat, najnižja cifra je prva pri rezultatu;

(2) v (16)= razdelimo na skupine po 4bite, sestevamo. (16) A=10, B=11, C=12, D=13, E=14, F=15,
st. grejo od 0..F

predstavitev števil(8bit):

predznak in velikost: (-127..127), ce
je st. neg. potem je prvi bit=1;
z odmikom: (-128..127 ali -127..128),
neg. številom pristajemo +128 ali
+127 nato izračunamo(2);
eniski komplement: (-127..127),
vzamemo poz. ekv. neg. st. in
invertiramo bite;
dvojski komplement: (-128..127), isto
kot eniski komplement, le da na
koncu pristajemo +1;

CPI, MIPS:

$CPI = \sum CPI_i \cdot P_i$

$MIPS = (CPE_{cas}) / (CPI \cdot 10^6)$

$CPE_{cas} = (st. ukazov) / (MIPS \cdot 10^6)$;

$CPE_{cas} = st. ukazov \cdot CPI \cdot t_{cpe}$;

$t_{cpe} = 1 / f_{cpe}$;

$MIPS = st. ukazov \cdot 10^6 / s$;

$CPI = st. period na$

ukaz;

Prekinitve, čas med prekinitvami:

zmanjšanje zmogljivosti zaradi prekinitve:

$X\%MIPS = X\% \cdot MIPS$ (ce vemo, za koliko % se zmanjša)

st. izgubljenih period(a):

$(X\% \cdot MIPS) \cdot 10^6 \cdot CPI$;

$N = \text{stevilo iskanih prekinitve}$:

$N = (a) / (st. urinih period, potrebnih za 1 prekinitve)$;

povprečni čas med prekinitvami:

$t_p = 1/N$;

Pohitritev racunalnika(Amdahlov zakon):

$S(N) = N / (1 + (N-1) \cdot f)$

f=delež operacij, ki se ne pohitrijo;

N=faktor pohitritve za (1-f) operacij;

$S(N)$ =povečanje hitrosti celotnega racunalnika;

$1-f = 1 - ((N-S(N)) / (S(N) \cdot (N-1)))$;

rezultat so procenti, za koliko se mora uporabljati dol.
komponenta racunalnika;

Sestevanje dvojskih števil:

predznak in velikost:

pri sestevanju je potrebno posebej upoštevati predznak =>
komplikacija;

z odmikom:

pri rezultatu je odmik upoštevan 2x, en odmik je treba odšteti;

eniski:

pri prenosu iz mesta n-1 je treba k rezultatu pristeti 1, sicer

sestevamo predznak enako kot ostale bite;

dvojski:

uporaben, ker se n-1 bit ze pristee pri konverziji;

Debeli&tanki konec: gre za nacin zapisovanja(vrstni red) števil (hex) v pomnilnik, pri debelem so v pravem zaporedju: 1F01A05C(16), pri tankem, pa zacemo na koncu:
5CA0011F(16); **Poravnanoost:** kadar velja A mod s = 0; A(10)=prvi naslov operanda, s=stevilo besed operanda(v nasem primeru recimo so 4)

Naloga-pohitritev racunalnika:

Delovanje racunalnika zelimo pohitrili tako,
da 1.8-krat pohitrimo delovanje
predpomnilnika. Kolikokrat hitreje bo
delovanje racunalnika, ce je predpomnilnik
uporabljen pri 70% vseh operacij?

Rešitev: $N = 1,8$;

$f = 1 - 0,7 = 0,3$

$S(N) = ?$

$S(N) = N / (1 + (N-1) \cdot f)$

$= 1,8 / (1 + 0,3 \cdot 0,3)$

$= 1,45$

Naslovni prostor, pomnilnik

Miniračunalniki v osemdesetih letih so
imeli 18 naslovnih signalov in seveda 18-
bitno naslovno vodilo.

a) Kolikšen je bil naslovni prostor teh
računalnikov?

b) Kolikšen je bil lahko največji možni
pomnilnik(beseda dolga 1 Bajt)?

c) Kako dolg je moral biti programski
števec (PC)?

d) d) Kaj vse bi bilo potrebno v
računalniku spremeniti, če bi želeli
naslovni prostor 8-krat povečati?

$2^{18} = 2^8 \times 2^{10} = 256K$

256 KB

18 (za vsak naslovni signal en bit)

Vodilo povečamo za 3 bite (2^3),

programski števec za 3 bite (2^3),

naslovna polja spremeniti tako, da bi lahko
naslavljal cel naslovni prostor

Naloga-prenos podatkov:

Procesor in pomnilnik sta povezana s 64bitnim
podatkovnim vodilom. Izračunajte kapaciteto
te povezave, ce je frekvenca urinega signala
na vodilu 133Mhz, prenos pa se izvrsi ob
vsaki pozitivni in negativni fronti urinega
signala.

Rešitev:

$f_{cpe} = 133 \text{ MHz}$

Podatkovno vodilo = 64 bitov : 8 = 8 Bajtov

prenos = $\frac{1}{2}$

B = ?

$B = (f_{cpe} / \text{prenos}) \cdot \text{širina vodila}$

$= (133 \cdot 10^6 / \frac{1}{2}) \cdot 64 \text{ bitov}$

$= 266 \cdot 10^6 \cdot 8$

$= 2128 \cdot 10^6$

$= 2128 \text{ MB/s}$

Naloga-prekinitve:

Na računalniku s frekvenco urinega signala 250 MHz je v povprečju
potrebnih 5 urinih period za en ukaz. Zaradi prekinitve se zmogljivost CPE
merjena v MIPS zmanjša za 0,12%. Ugotovite povprečni čas med dvema
prekinitvama, če se ob vsaki prekinitvi porabi 48 urinih period za klic
prekinitvenega servisnega programa in 27 urinih period za vračanje iz
njega.

Frekvenca = 250 MHz

$CPI = 5$

$MIPS = f_{CPE} / (CPI \times 10^6) = 50$

MIPS se zmanjša za 0,12% to je \boxed{x} $50 \times 10^6 \times 0,0012 = 0,06$ je manjši
MIPS

$0,06 \times 10^6 \times CPI = 0,06 \times 10^6 \times 5 = 0,3 \times 10^6 = 3 \times 10^5$ izgubljenih
urinih period na sekundo (skupno)

Stevilo prekinitvev = zgubljen čas / čas na prekinitvev = $3 \times 10^5 / (48+27) =$
4000 v sek

Cas prekinitvev = $1/\text{stevilo prekinitvev} = 1/4000 = 0,25 \text{ ms}$.

Primerjava 2 racunalnikov, MIPS,CPI, CPE_{cas} ,za 12000ukazov, kdaj sta enako hitra?
MIPS

$CPI_{(R1)} = \sum CPI_{i(R1)} \times P_{i(R1)} = 6 \times 0,16 + 8 \times 0,1 + 3 \times 0,66 = 4,54$

$CPI_{(R2)} = -- \text{ II } == 20 \times 0,16 + 32 \times 0,1 + 66 \times 0,8 + 3 \times 0,66 = 13,66$

$MIPS_{(R1)} = (400 \times 10^6) / (4,54 \times 10^6) = 88,1$

$MIPS_{(R2)} = (400 \times 10^6) / (13,66 \times 10^6) = 29,28$

CPE cas izvajanja za program z 12.000 ukazov

$CPE_{(R1)} = CPI_{(R1)} \times 12.000 \times t_{0(CPE)} =$

$= (CPI_{(R1)} \times 12.000) / f_{(CPE)} = 12.000 / (MIPS_{(R1)} \times 10^6) = 136,2 \times 10^{-6} \text{ s} = 0,136 \text{ ms}$

$CPE_{(R2)} = CPI_{(R2)} \times 12.000 \times t_{0(CPE)} =$

$= (CPI_{(R2)} \times 12.000) / f_{(CPE)} =$
 $= 12.000 / (MIPS_{(R2)} \times 10^6) = 410,0 \times 10^{-6} \text{ s} = 0,410 \text{ ms}$

Pri kaksni mesanici ukazov pri programu sta oba racunalnika enaka?

Ce bi uporabljali samo ukaze, ki ne uporabljajo plavajoče vejice.

Števila s plavajoco vejico (IEEE 754):	0,4*2=0,8+0	zanima zgolj eksponent, ki je v tem primeru enak 1 (pride od 10 ¹).
Predznak: 1 za - in 0 za +	0,8*2=0,6+1	
Eksponent: zapisan v predstavitvi z odmikom 127	0,6*2=0,2+1	
Mantisa: vedno oblike 1,..., zato prvi bit izpustimo	0,2*2=0,4+0	
	0,4*2=0,8+0	
Poglejmo pretvorbo po korakih:	...(gres do nicle oz vseh 32 bitov mantise)	
1. Ugotovimo predznak:		
3,35 je več kot 0 => predznak = 0	Iz zgornjega torej sledi:	
	0,35 (10) = 0,0101100... (2)	
2. V dvojiški sistem pretvorimo število pred decimalno vejico:		
3 (10) = 11 (2)	4. Napišemo dvojiško število z decimalkami:	
	3,35 (10) = 11,0101100... (2)	
3. V dvojiški sistem pretvorimo število za decimalno vejico (glej tudi priponko):	5. Prestavimo eksponent:	
0,35 (10) = ? (2)	11,0101100... (2) = 1,10101100... * 10 ¹ (2)	
	V zgornjem primeru je vse napisano v dvojiškem. 10 ¹ v dvojiškem pomeni 2 ¹ v desetiškem! Iz tega zapisa nas	
0,35*2=0,7 +0		
0,7*2=0,4+1		
		6. Eksponent zapišemo v predstavitvi z odmikom 127:
		6.1. Najprej prištejemo odmik: 1+127 = 128
		6.2. To pretvorimo v dvojiško obliko: 128(10) = 10000000(2)
		7. Zapišemo število v obliki IEEE 754:
		1 10000000 10101100...
		Predznak Eksponent Prvih nekaj bitov mantise (skupaj 23 bitov)
		Dvojiško st. z decimalkami pretvorimo v (10) takole:
		101,1101=
		1*(2 ²)+0*(2 ¹)+1*(2 ⁰)+1*(2 ⁻¹)+1*(2 ⁻²)+0*(2 ⁻³)+1*(2 ⁻⁴);

V zbirnem jeziku za 68HC11 napisite program, ki izracuna povprečno vrednost osmih nepredznacenih števil. Števila so podana v tabeli:

TABELA FCB 15,20,10,5,25,40,50,35

Rezultat zapisite na prvo mesto v tabeli (prepisite ga cez stevilo 15). V programu obvezno uporabite zanko, v kateri sestejete števila v tabeli. Po koncu zanke rezultat delite z 8. Namig: z 8 delimo tako, da število pomaknemo za 3 bite v desno (ukaz LSR pomakne operand za 1 bit v desno)

	ORG	\$FFFE
	FDB	START
; -----		
TABELA	FCB	15,20,10,5,25,40,50,35
; -----		
	ORG	\$E000
START		
; -----		
	LDX	#TABELA
ZANKA	LDY	#TABELA
	LDA	0,X
	LDAB	1,Y
	ABA	
	STAA	0,X
	INY	
	CMPB	#35
	BNE	ZANKA
	LSR	0,X
	LSR	0,X
	LSR	0,X
; -----		
KONEC	BRA	KONEC
	END	
	ORG	\$FFFE
	FDB	START

Nacini naslavljanja:

IMM = takojšnje naslavljanje (torej takrat ko imate pri operandu #, npr. LDAA #10)

DIR = neposredno naslavljanje (kadar je operand **8 bitni** pomnilniški naslov, npr. LDAA \$A0)

EXT = razširjeno neposredno naslavljanje (kadar je operand **16 bitni pomnilniški naslov**, npr. LDAA \$2000 - ta način smo največkrat uporabljali na vajah)

IND,* = indeksno naslavljanje (operand je pomnilniški naslov podan z indeksnim registrom, npr. LDAA 0,X - bodite pozorni, da je število ciklov včasih odvisno od uporabljenega registra npr. 5 ciklov za 0,Y in 4 cikle za 0,X)

IEEE 754 Pretvarjanje nazaj se lahko naredi malce lažje (število je samo primer!):

Pretvarjamo torej:

1 1000 0011 0100 0000 0000 0000 0000

1. predznak

Predznak = 1 => število je negativno (-)

2. eksponent

Najprej pretvorimo dvojiško vrednost v desetiško:

1000 0011 (2) = 128 + 2 + 1 (10) = 128 + 3 (10)

Ker gre za predstavitev z odmikom moramo odšteti odmik (127):

128 + 3 - 127 = 4

Eksponent je torej enak 4.

3. mantisa

0100 0000 0000 0000 0000 000

Spomnimo se, da je prva enica za katero sledi decimalna vejica pri mantisi implicitna. Zgornje lahko torej zapišemo takole (preostanek ničel ignoriramo):

1,01

4. mantiso pomnižimo z eksponentom

1,01 (2) * 2⁴ (10) = 10100 (2)

Ker je eksponent 4 moramo mantiso pomnožiti z 2⁴. Pri množenju dvojiškega števila s potencami števila 2, gre za premikanje vejice.

5. Rezultat pretvorimo v desetiško število

10100 (2) = 4 + 16 = 20

6. Upoštevamo še predznak

In rezultat je -20.