CPE

$$CPE_{čas} = št.ukazov \cdot CPI \cdot t_{CPE}$$

$$CPE_{čas} = \frac{št.ukazov}{MIPS \cdot 10^6}$$

$$CPI = \frac{CPE_{čas}}{št.ukazov \cdot t_{CPE}}$$

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

$$CPI = \frac{t_{ap}}{t_{CPE}}$$

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

$$MIPS = \frac{CPE_{čas}}{št.ukazov \cdot t_{CPE}}$$

$$t_{CPE} = \frac{1}{f_{CPE}}$$

$$f_{CPE} = \frac{1}{t_{CPE}}$$

$$t_{ap} = 1 \cdot t_{CPE} = 1 \cdot \frac{1}{f_{CPE}}$$

Predpomnilnik, seti, bloki, asociativnost

$$M = S \cdot E \cdot B = 2^{s+e+b}$$

$$E = \frac{M}{S \cdot B}$$

$$S = \frac{M}{E \cdot B}$$

$$B = \frac{M}{S \cdot E}$$

$$št.blokov = \frac{M}{B}$$

$$st = \frac{M}{S}$$

$$S_i = A_i(b:n-1) \bmod 2^S$$

M - velikost predpomnilnika
S - število setov ($S = 2^s$)
E - stopnja asociativnosti (število blokov v setu, $E = 2^e$)
B - velikost bloka($B = 2^b$)
st - velikost seta
S_i - naslov seta, v katerega se lahko preslika blok iz gl. pomnilnika
$A_i(b:n-1)$ - zgornjih n-b bitov pom. naslova = številka bloka
2^S - število setov v predpomnilniku

zadetki pri predpomnilniku

$$H = \frac{N_p}{N} = \frac{N_p}{N_g + N_p}$$

H - verjetnost zadetka
N - število vseh dostopov do predpomnilnika
N_p - število zadetkov
N_g - število zgrešitev

CPI_R - realni CPI
CPI_I - idealni CPI brez zgrešitev v predpomnilniku
$CPE_{čas}$ - čas, ki ga CPE porabi za določen program
CPI - povprečno število urinih period na ukaz
$MIPS$ - milijon ukazov na sekundo
f_{CPE} - urin signal frekvence
M_I - povprečno število pomnilniških dostopov na ukaz
$(1-H)$ - verjetnost zgrešitve
H - verjetnost zadetka
I - število ukazov
t_a - povprečni dostopni čas
t_{ap} - čas dostopa do predpomnilnika
t_{ag} - čas dostopa do glavnega pomnilnika
t_B - čas prenosa enega bloka iz navideznega pomnilnika
b_a - hitrost dostopa
t_c - čas cikla med dvema zaporednima dostopoma
t_m - mrtvi čas (DRAM)
t_{CPE} - urina perioda
t_{ap} - čas dostopa do pomnilnika
P_W - delež pisalnih dostopov (npr. $20\% = 0,2$)
P_R - delež bralnih dostopov (npr. $80\% = 0,8$)
N_W - pisalna zgrešitvena kazen (urine periode)
N_R - bralna zgrešitvena kazen (urine periode)
H_{pp} - verjetnost zadetka pri predpomnilniku

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Pomnilniško prepletanje

- pri m modulih imamo m-kratno prepletanje
- pri m-kratnem spodnjem prepletanjem je možno $\sqrt{m}$ istočasnih dostopov

Predpomnilnik in pomnilnik

$$CPI_R = CPI_I + M_I \cdot (1-H) \cdot zg.kazen$$

$$CPE_{čas} = I(CPI_I + M_I \cdot (1-H) \cdot zg.kazen) \cdot t_{CPE}$$

$$CPI_{zgrešitvena} = \frac{servis\ napake}{t_{CPE}}$$

$$CPI_{raz} = CPI_V - CPI$$

$$t_{ap} = t_a - (1-H) \cdot t_{ag}$$

$$t_a = t_{ap} + (1-H) \cdot zg.napaka$$

$$t_{ag} = \frac{t_a - t_{ap}}{(1-H)}$$

$$t_b = \frac{t_a + vel.strani}{t - prenosa}$$

$$št.blokov\ v\ pp = \frac{vel.pp}{vel.bloka}$$

$$pov.zg.kazen = P_W \cdot N_W + P_R \cdot N_R$$

$$verjetnost = \frac{CPI_{raz}}{CPI_{zgrešitvena}}$$

$$b_a = \frac{1}{t_c} = \frac{1}{t_a + t_m}$$

$\frac{1}{s} = Hz$
$m = 10^{-3}$
$\eta = 10^{-6}$
$n = 10^{-9}$
$1Bajt = 8bitov$
$10^9 = 1GHz$
$10^6 = 1MHz$
$10^3 = 1KHz$
$2^{10} = 1K$
$2^{20} = 1M$
$2^{30} = 1G$

čisti asociativni pomnilnik
- lahko se preslika v katerikoli blok

set-asociativni pomnilnik
- se preslika v set po modulu asociativnosti

(npr. 13. naslov, št. setov je 4 → $13 \bmod 4 = 1$)

$(fizični\ naslov) \bmod (št.\ setov) = set,\ v\ katerega\ se\ preslika$

- lahko se preslika v katerikoli blok v danem setu

direktni predpomnilnik

$(fizični\ naslov) \bmod (št.\ blokov) = blok,\ v\ katerega\ se\ preslika$

-preselika se v točno določeni blok

Amdahlov zakon

$$S(N) = \frac{N}{1 + (N-1) \cdot f}$$

$$f = \frac{N - S(N)}{S(N) \cdot (N-1)}$$

$S(N)$ - povečanje hitrosti celotnega računalnika
f - delež operacij, ki niso pohitrene
$(1-f)$ - delež operacij, ki so N-krat pohitrene

Ne pozabiti pretvoriti par zadevic pred računanju!!!

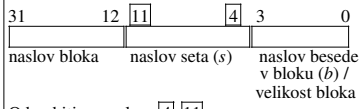
$60ns = 60 \cdot 10^{-9}$

$16KB = 2^4\,KB = 2^{14}\,B$

če imaš dano dolžino urine periode ali frekvenco ter vse podatke v urinih periodah, potem množi dane podatke s časom ene urine periode

$čas\ dostopa = št.\ urinih\ period \cdot dolžina\ urine\ periode$

Vprašanje: Kateri biti v pomnilniškem naslovu določajo naslov seta?
- 32 bitni pomnilniški naslov
- $B = 2^4 = 2^b; b = 4$
- $S = 2^8 = 2^s; s = 8$



Odg.: biti za naslov: 4-11.

Naslednje vprašanje: V kateri set se preslika vsebina iz pomnilniškega naslova $10FFCFF_{(hex)}$?

$^{31}0000\,|\,0001\,|\,0000\,|\,1111\,|\,1111\,|\,1100\,|\,1111\,|\,1111$
$\quad\quad 1 \quad\quad 0 \quad\quad F \quad\quad F \quad\quad C \quad\quad F \quad\quad F$

$^{7\,6\,5\,4\,3\,2\,1\,0}11001111 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 207$

Odg.: Naslov se preslika v set 207.

Vrste zgrešitev v predpomnilniku

- obvezne zgrešitve
  - ob prvem dostopu do besede, ki je ni v predpomnilniku (zgrešitev prvega dostopa)
  - velikost predpomnilnika, stopnja asoc. in zamenjalna strategija na te zgrešitve ne vpliva
  - pri večjih blokih je obveznih zgrešitev manj
- velikostne zgrešitve
  - nastopajo zaradi končne velikosti predpomnilnika, ko je ta poln
  - z večanjem predpomnilnika se št. teh zgrešitev manjša
- konfliktne zgrešitve
  - nastopajo samo pri set-asoc. in direktnih predpomnilnikih zaradi omejitev pri preslikavi
  - do konfliktne zgrešitve pride, ko se zamenja blok, ki se bo kmalu spet uporabil
  - do zamenjave pride, ker se morata oba bloka zaradi omejitev pri preslikavi preslikati v isti set ali blok

cevovodne nevarnosti
- kadar več stopenj cevovoda v isti urini periodi potrebuje isto enoto

podatkovne nevarnosti
- kadar nek ukaz potrebuje rezultat predhodnega ukaza, ki pa še ni končan

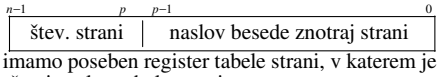
kontrolne nevarnosti
- pri ukazih, ki spreminjajo vrednost PC (skoki, klici,...)

Za računalnik s predpomnilnikom in glavnim pomnilnikom izračunajte povprečni dostopni čas do pomnilnika kot ga vidi CPE. CPE deluje z urino periodo 10ns. Verjetnost zadetka v predpomnilniku je 96%, čas dostopa do predpomnilnika je 1 urina perioda, zgrešitvena kazen pa je 8 urinih period pri branju in 16 urinih period pri pisanju. Pri vseh pomnilniških dostopih je 75% bralnih dostopov.

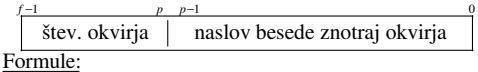
$t_{ap} = 1 = 10ns$
$t_{CPE} = 10ns = 10 \cdot 10^{-9}\,s$
$H = 0,96$
$(1-H) = 0,04$
$N_R = 8$
$N_W = 16$
$P_R = 0,75$
$P_W = 0,25$
$t_a = ?$
$t_a = t_{ap} + (1-H) \cdot zg.kazen$
$\boxed{t_a} = 10ns + 0,04 \cdot (0,75 \cdot 8 \cdot 10ns + 0,25 \cdot 16 \cdot 10ns) = \boxed{14ns}$

Dostop in navidezni pomnilnik

navidezni pomnilnik z odstranjevanjem:
- pomožni pomnilnik (disk) je razdeljen na strani (pages)
- glavni pomnilnik (RAM) je razdeljen na okvirje strani (page frames)
- vsako stran je možno prenesti v poljuben okvir
- velikost strani je potenca števila 2 $\rightarrow$ 2^{neki}
- preslikava navideznega v fizični naslov je definirana s pomočjo tabele strani (page table) - ponavadi v RAM
  - eno polje v tej tabeli je deskriptor strani
    - zgradba deskriptorja: | parametri | štev. okvirja/FN |
- zgradba navideznega pomnilniškega naslova:



- imamo poseben register tabele strani, v katerem je shranjen začetni naslov tabele strani
- CPE generira navidezni naslov $\rightarrow$ vzame se od p do n-1 bitov in prišteješ k registru tabele strani $\rightarrow$ tako dobimo naslov deskriptorja v tabeli strani $\rightarrow$ $\rightarrow$ v deskriptorju imamo štev. okvirja, ki nam poda št. bloka v gl. pomn., ki ga iščemo
- zgradba fizičnega naslova:



Formule:

$$št.strani\ v\ navideznem\ pom. = 2^{n-p}$$

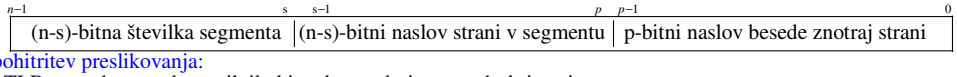
$$št.okvirjev\ v\ gl.\ pom. = 2^{f-p}$$

$$največja\ možna\ velikost\ tabele\ strani = št.strani \cdot dolžina\ deskriptorja$$

n - dolžina navideznega naslova v bitih
f - dolžina fizičnega naslova v bitih
p - velikost strani (okvirja) v bitih

navidezni pomnilnik s segmentacijo:
- namesto tabele strani imaš tabelo segmentov, ki vsebuje deskriptorje segmentov
- v deskriptorju je fizični naslov segmenta (namesto št. naslova/okvirja) in velikost segmenta
- število segmentov ni fiksno, velikost tabele ni vnaprej določena
- vsak program ima svojo tabelo segmentov

navidezni pomnilnik s segmentacijo z odstranjevanjem (linearna seg.):
- vsakemu programu pripada tabele segmentov in ena tabela strani za vsak segment tega programa



pohitritev preslikovanja:
- TLB - poseben predpomnilnik, ki vedno vsebuje samo deskriptorje
- dolžina bloka v TLB je enaka dolžini deskriptorja, v kontrolnem delu pa je številka strani, ki ji deskriptor pripada
- pri zadetku v TLB dostop do RAM-a ni potreben (v tem primeru ne pišemo t_{ap} dvakrat, ampak le enkrat)

$$t_a = t_{ap} + t_{ap} + (1-H) \cdot t_B$$

*če bi bila n-nivojska preslikava, moraš t_a množiti z n (!!)

Navidezni pomnilnik

$$N = N_1 + N_2 + \ldots + N_n$$

$$H_1 = \frac{N_1}{N} \qquad H_2 = \frac{N_1 + N_2}{N}$$

$$H_n = \frac{N_1 + N_2 + \ldots + N_n}{N} = 1$$

$$t_a = t_{a1} + (1-H_1)t_{a2} + \ldots + (1-H_{i-1})t_{ai} + \ldots + (1-H_{n-1})t_{an}$$

H_n - verjetnost zadetka pri dostopu do navideznega pomnilnika, vedno je enak 100% oz. 1 (notri je VEDNO informacija, ki jo iščemo)
N - skupno število dostopov do pomnilniške hierarhije (dostop do vseh pomnilnikov)
N_x - število dostopov do pomnilnika x v hierarhiji
$t_{a1}, t_{a2}, \ldots$ - dostopni časi do pomnilnika a v hierarhiji

*primer za štirinivojsko hierarhijo: $t_a = t_{a1} + (1-H_1)t_{a2} + (1-H_2)t_{a3} + (1-H_3)t_{a4}$