Adresovanie informácií v počítači

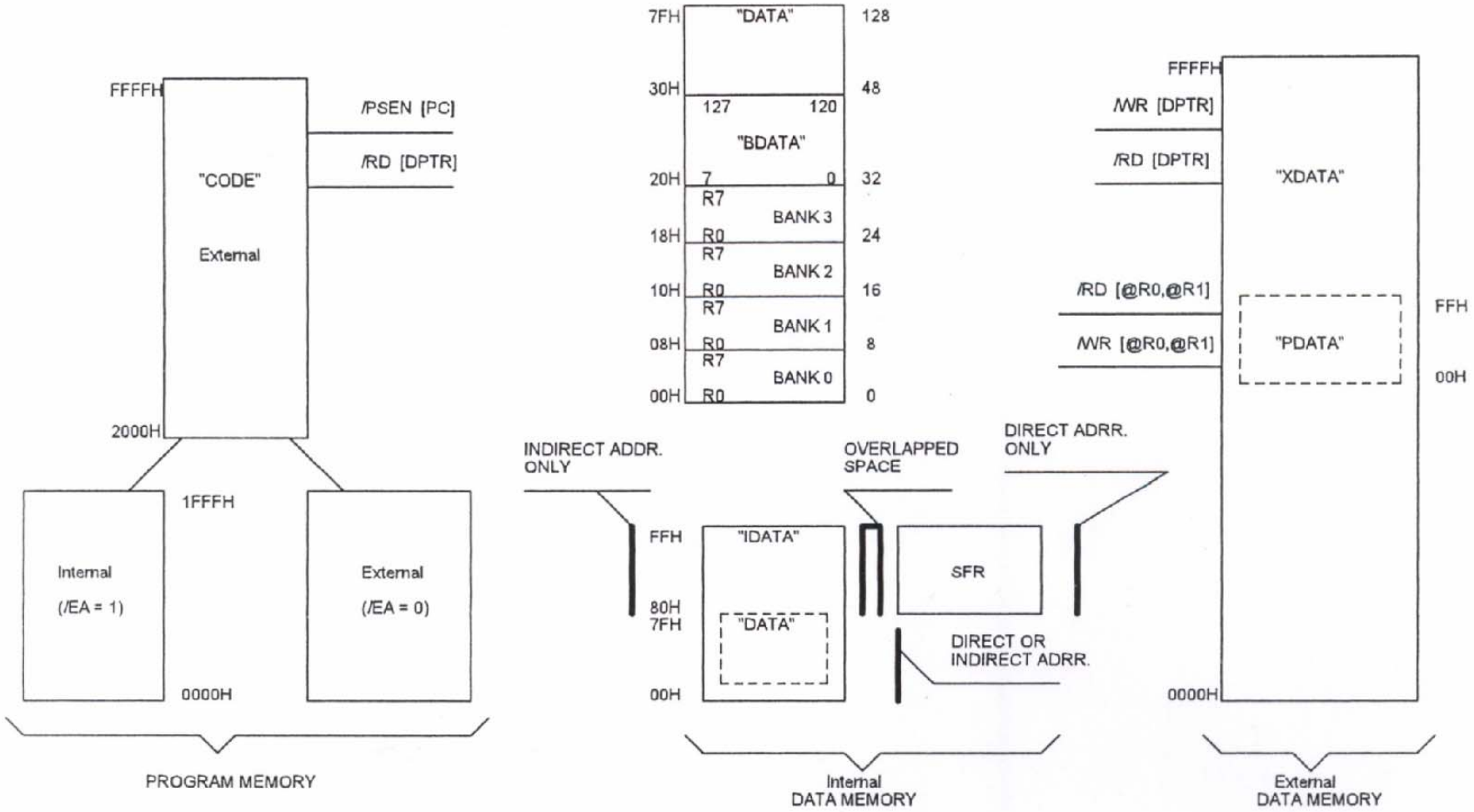
MMU – (Memory Management Unit)
- správa pamäťového systému počítača

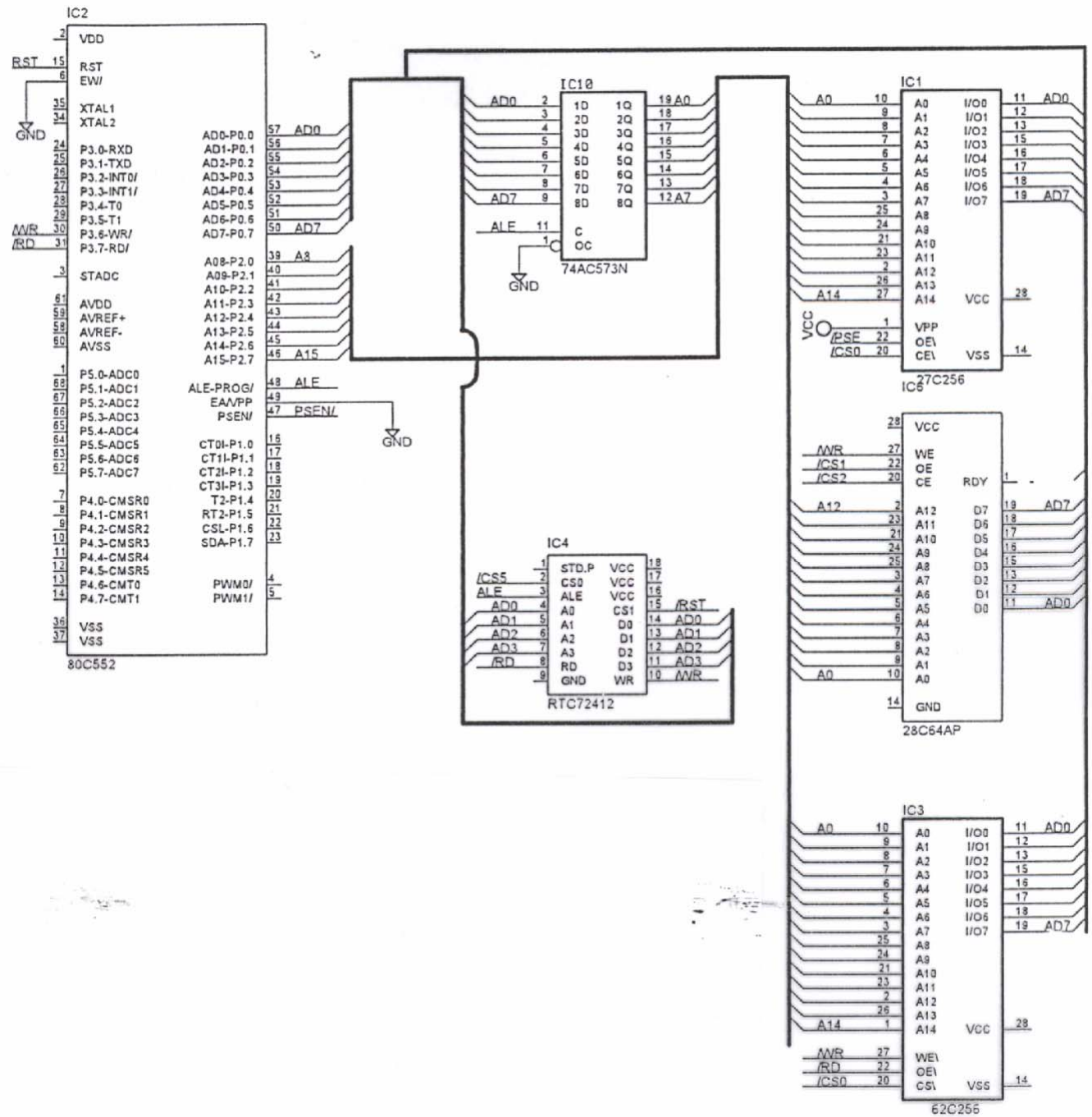
Jednoúrovňový pamäťový systém

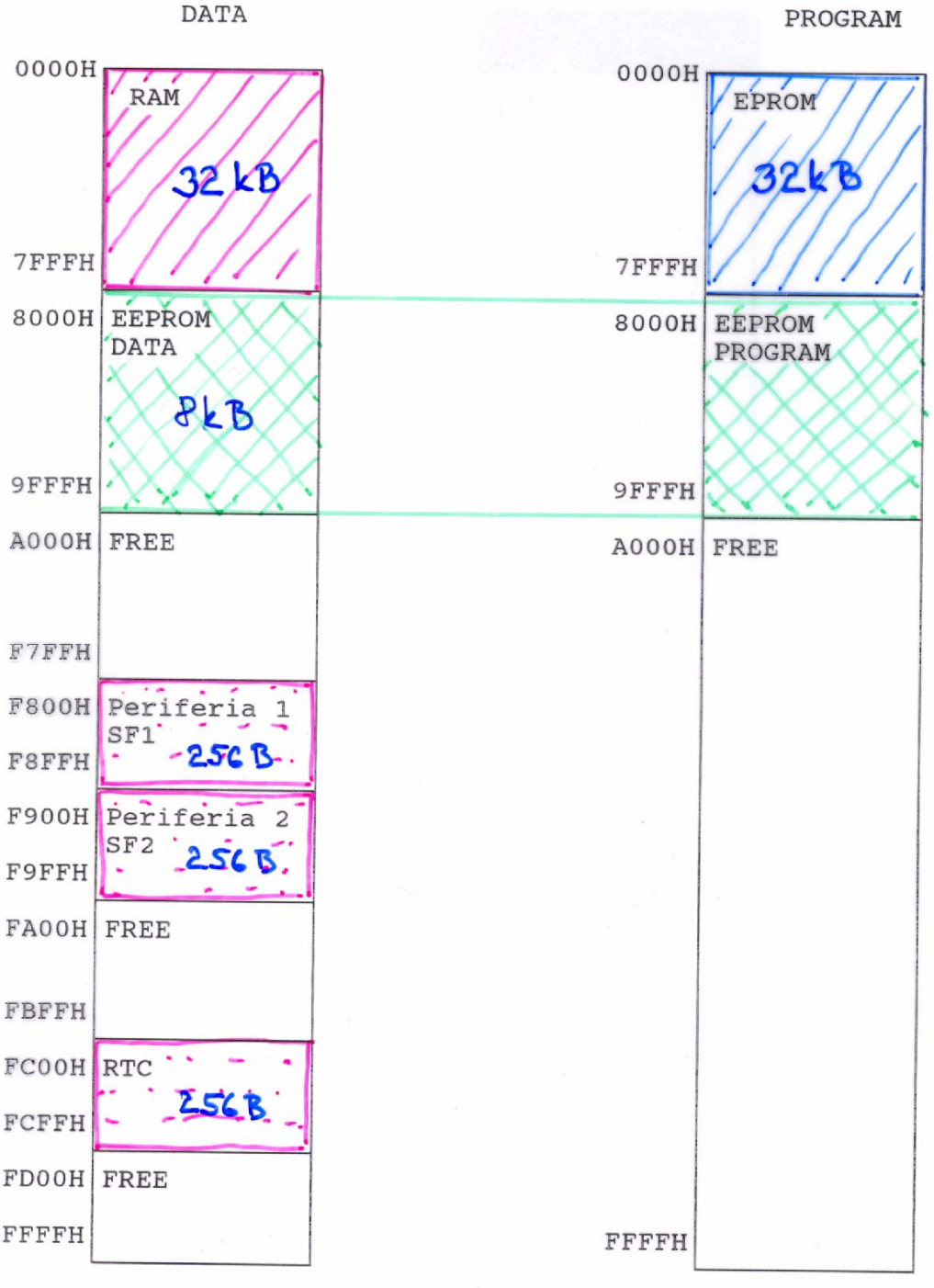
Adresovanie hlavnej pamäte
-fyzická adresa
-logická adresa

V malých systémoch sa používa výlučne **fyzická adresa** (binárne vyjadrenie adresy v inštrukciách je zhodné s binárnym vyjadrením signálov adresnej zbernice pripojenej k pamäťovému podsystému)

“51” “552” Organizácia pamäte, Pr.




IC2
VDD
RST
EW/
XTAL1
XTAL2
P3.0-RXD
P3.1-TXD
P3.2-INT0/
P3.3-INT1/
P3.4-T0
P3.5-T1
P3.6-WR/
P3.7-RD/
STADC
AVDD
AVREF+
AVREF-
AVSS
P5.0-ADC0
P5.1-ADC1
P5.2-ADC2
P5.3-ADC3
P5.4-ADC4
P5.5-ADC5
P5.6-ADC6
P5.7-ADC7
P4.0-CMSR0
P4.1-CMSR1
P4.2-CMSR2
P4.3-CMSR3
P4.4-CMSR4
P4.5-CMSR5
P4.6-CMT0
P4.7-CMT1
VSS
VSS
AD0-P0.0
AD1-P0.1
AD2-P0.2
AD3-P0.3
AD4-P0.4
AD5-P0.5
AD6-P0.6
AD7-P0.7
A08-P2.0
A09-P2.1
A10-P2.2
A11-P2.3
A12-P2.4
A13-P2.5
A14-P2.6
A15-P2.7
ALE-PROG/
EA/VPP
PSEN/
CT0I-P1.0
CT1I-P1.1
CT2I-P1.2
CT3I-P1.3
T2-P1.4
RT2-P1.5
CSL-P1.6
SDA-P1.7
PWM0/
PWM1/
80C552
RST
GND
AD0
AD7
A8
A15
ALE
PSEN/
IC10
74AC573N
IC1
27C256
IC6
28C64AP
IC4
RTC72412
IC3
62C256
/WR
/RD
/CS0
/CS1
/CS2
/CS5
/PSE
VCC



Signály adresného dekódera:

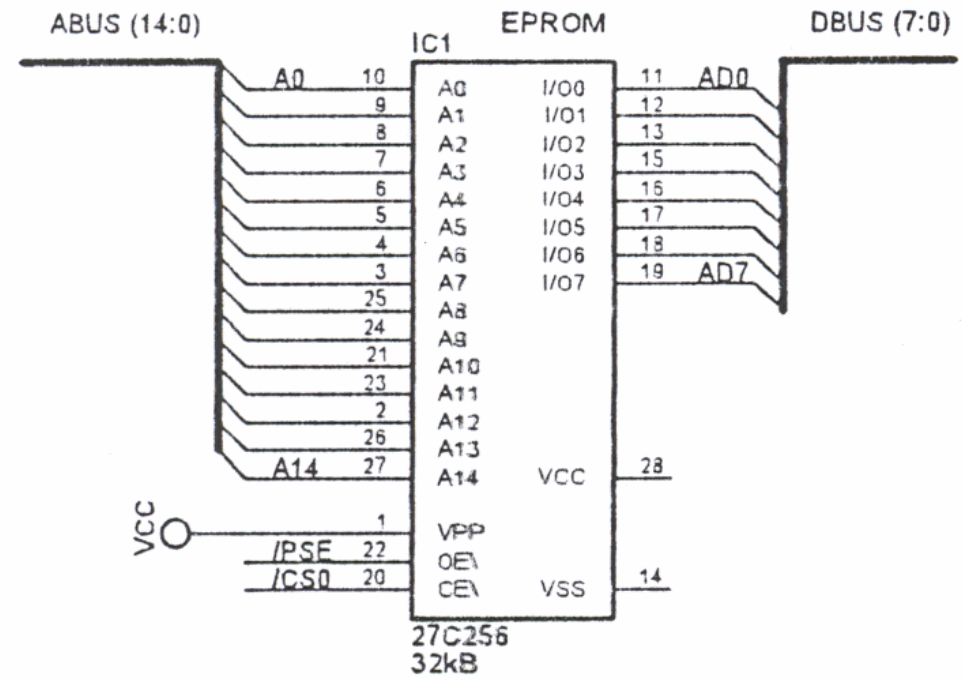
A0 až A15: adresné signály

/RD: signál na prenos dát medzi vonkajšou pamäťou a procesorom

/WR: signál na prenos dát medzi procesorom a vonkajšou pamäťou

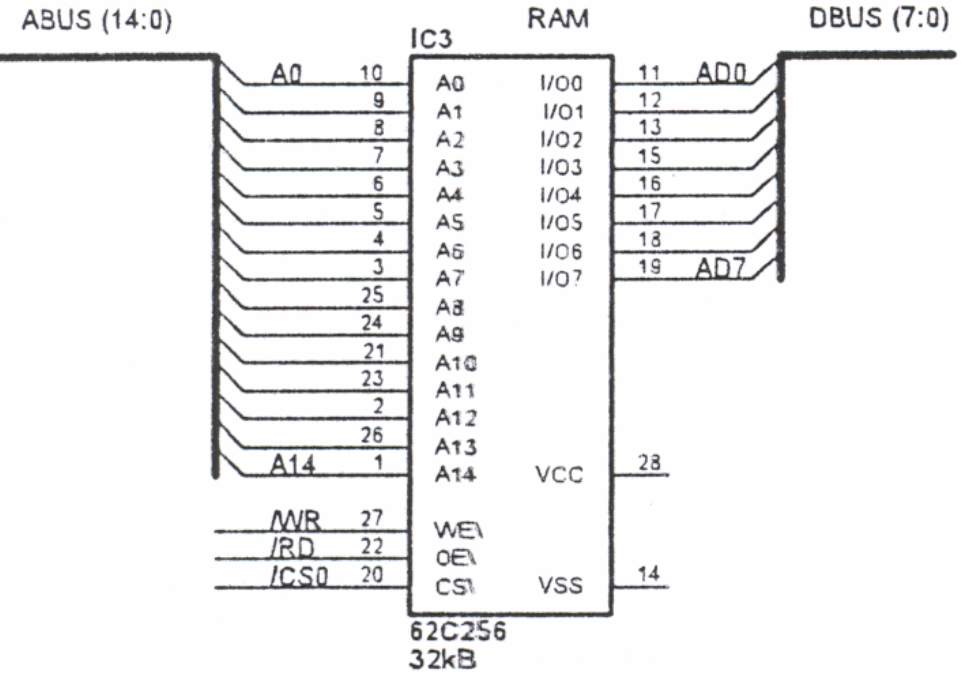
/PS: signál na prenos programu medzi vonkajšou pamäťou a procesorom

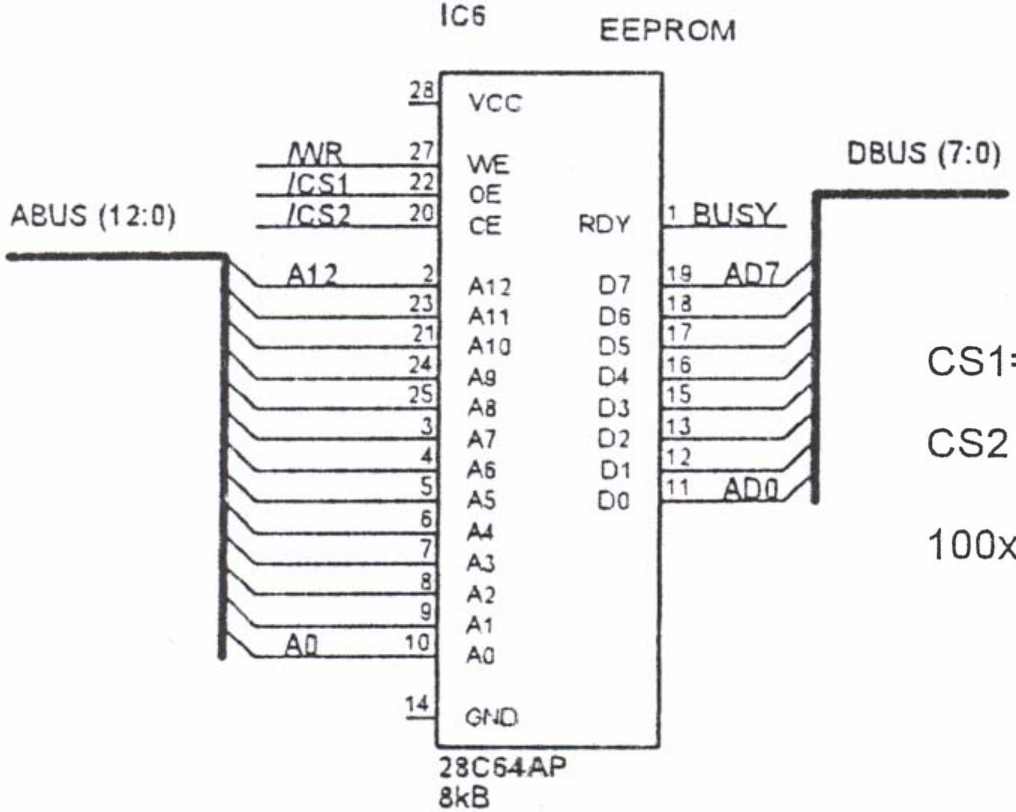
CS-ty obvodov aktívne do nuly



CS0 = /A15

0xxx xxxx xxxx xxxx = 0000H az 0FFFH

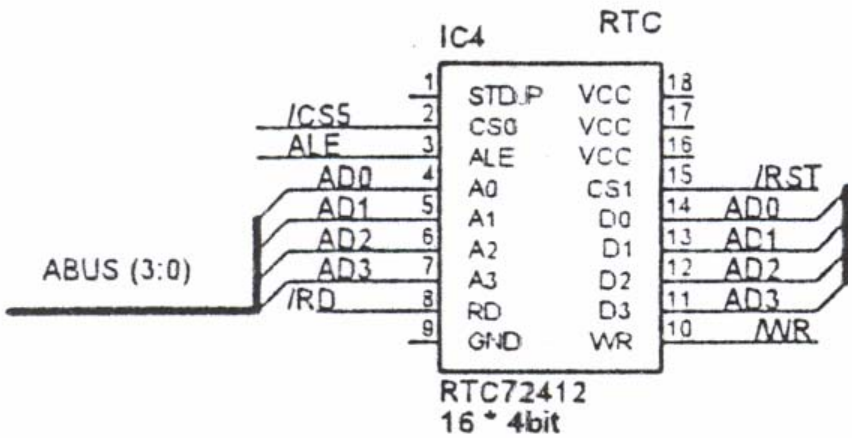




CS1= A15*/A14*/A13*PSE + A15*/A14*/A13*RD

CS2 = A15*/A14*/A13

100x xxxx xxxx xxxx = 8000H az 9FFFH



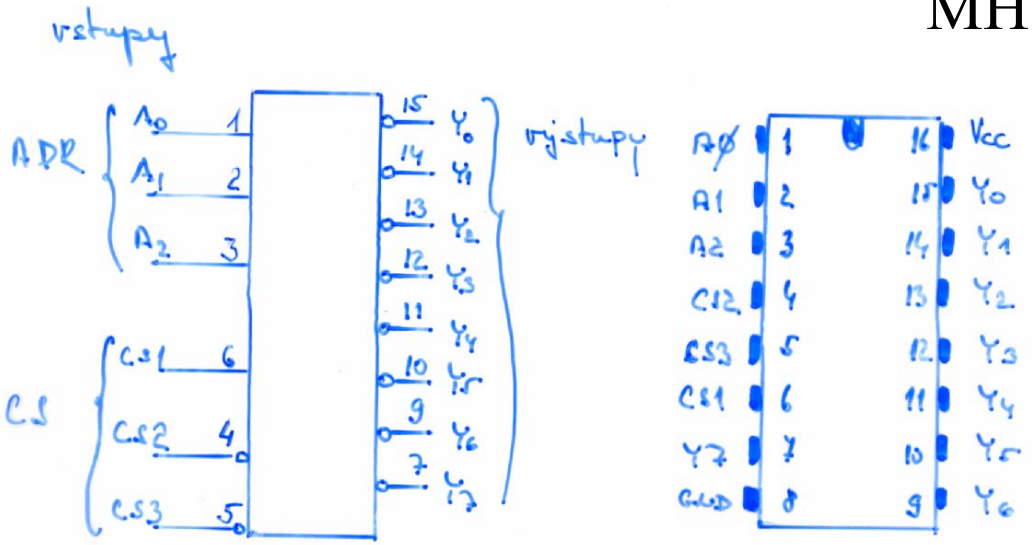
CS5= A15*A14*A13*A12*A11*A10*/A9*/A8

1111 1100 xxxx xxxx = FC00H az FCFFH

FC0XH = FC1XH = . . . = FCFXH

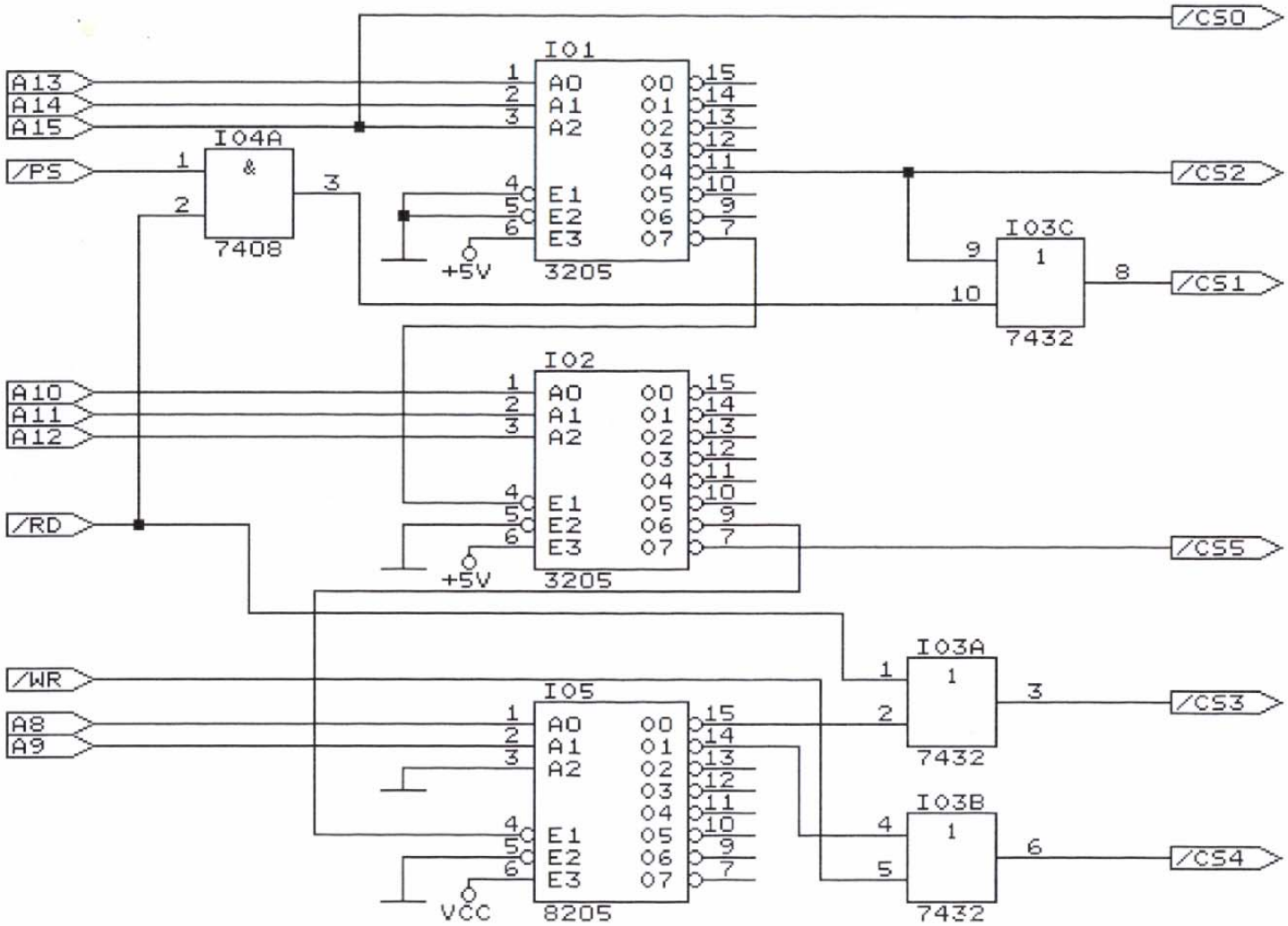
X = 0H az FH

MH 3205, 74HC138



A0	A1	A2	CS1	CS2	CS3	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
L	L	L	H	L	L	L							
H	L	L	H	L	L		L				H		
L	H	L	H	L	L			L					
H	H	L	H	L	L				L				
L	L	H	H	L	L					L			
H	L	H	H	L	L		H				L		
L	H	H	H	L	L							L	
H	H	H	H	L	L								L
X	X	X	L	H	H			H					

ADRESOVY DEKODER S VYUZITIM IO 3205



Popis signalov:
A10-A15 adresove signaly
/WR zapis do vonkajsej pamati dat
/RD citanie z vonkajsej pamati dat
/PS citanie z vonkajsej pamati programu

/CS0 CS pamati RAM a EPROM
/CS1 OE pamati EEPROM
/CS2 CS pamati EEPROM
/CS3 CSF1
/CS4 CSF2
/CS5 CS hodin RTC

Dekodovane adresne priestory:

RAM	0000H-7FFFH	DATA
EPROM	0000H-7FFFH	PROGRAM
EEPROM	8000H-9FFFH	DATA+PROGRAM
RTC	FC00H-FCFFH	DATA
CSF1	F800H-F8FFH	DATA
CSF2	F900H-F9FFH	DATA

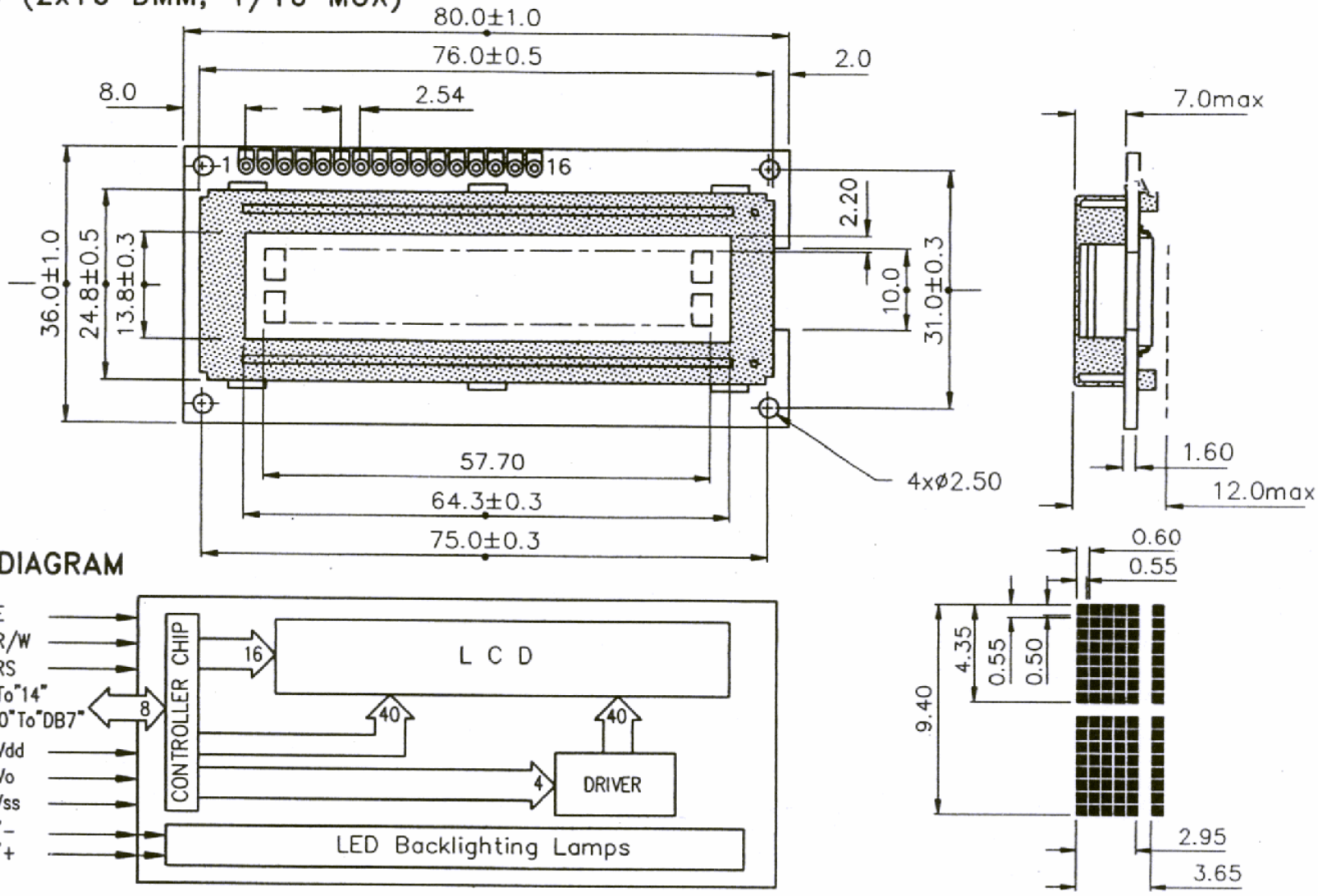
ALPHANUMERIC DOT MATRIX MODULES WITH BUILT-IN LED BACK LIGHTS

MODULE DIMENSIONS

VK2123 (2x16 DMM, 1/16 MUX)

DIMENSION IN MM

"L"



BLOCK DIAGRAM

ALPHANUMERIC DOT MATRIX MODULES

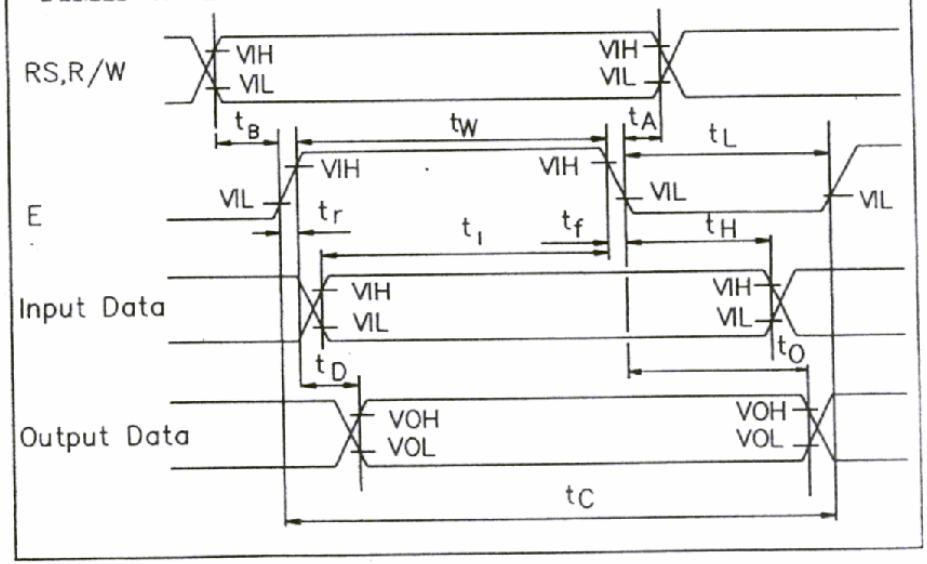
DEFINITION OF TERMINALS

PIN NO.	SYMBOL	FUNCTION
1.	Vss	Ground terminal of module.
2.	Vdd	Supply terminal of module,+5V.
3.	Vo	Power supply for Liquid Crystal Drive
4.	RS	Register Select RS = Ø... Instruction Register. RS = 1... Data Register.
5.	R/W	Read/Write R/W = 1...Read R/W = Ø...Write
6.	E	Enable
7~14.	DBØ~DB7	Bi-directional Data Bus. Data Transfer is performed once, thru DBØ-DB7, in the case of interface data length is 8-bits; and twice, thru DB4-DB7, in the case of interface data length is 4-bits. Upper four bits first then lower four bits.
15.	LAMP- (L-)	LED or EL lamp power Supply terminals
16.	LAMP+ (L+)	

OPERATING SPECIFICATIONS

	STANDARD TEMP	WIDE TEMP
Operating temperature range	0°C to +50°C	-20°C to +70°C
Storage temperature range	-40°C to +70°C	-40°C to +85°C
Operating relative humidity	90% MAX	90% MAX

TIMING DIAGRAM



4.3 Timing characteristics

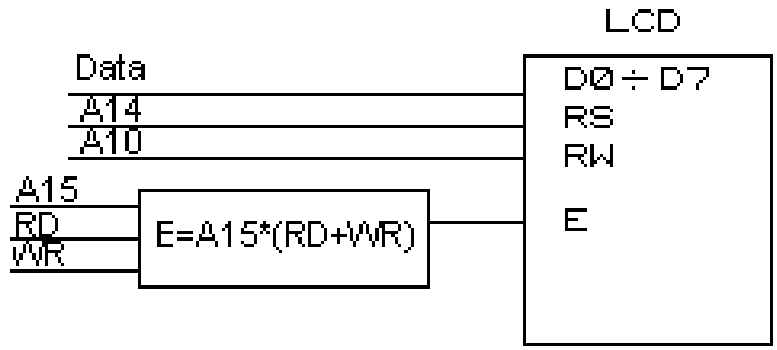
Table 4

VDD=5.0V±5%
Ta=0～50℃

Parameter	Symbol	Min.	Typ.	Max.	Unit
Enable cycle time	tcycE	1000	-	-	ns
Enable pulse width	PWEH	450	-	-	ns
Enable rise/fall time	tEr, tEf	-	-	25	ns
RS, R/W setup time	tAS	140	-	-	ns
Address hold time	tAH	10	-	-	ns
Data setup time	tDSW	195	-	-	ns
Data delay time	tDDR	-	-	320	ns
Data hold time(write)	tH	10	-	-	ns
Data hold time(read)	tDHR	20	-	-	ns

Timing chart: See Fig.1.

Číslo pinu	Symbol	I/O	Funkcia
1	V_{SS}	-	0V napájanie
2	V_{DD}	-	+5V napájanie
3	V_{EE}	-	Kontrast
4	RS	I	0 = vstup je inštrukcia 1 = vstup sú dáta
5	R/W	I	0 = zápis dát do LCD 1 = čítanie dát z LCD
6	E	I	Aktivácia displeja
7	DB0	I/O	Dáta, bit 0
8	DB1	I/O	Dáta, bit 1
9	DB2	I/O	Dáta, bit 2
10	DB3	I/O	Dáta, bit 3
11	DB4	I/O	Dáta, bit 4
12	DB5	I/O	Dáta, bit 5
13	DB6	I/O	Dáta, bit 6
14	DB7	I/O	Dáta, bit 7



čas trvania pulzu Enable:
min.450ns

čas trvania RD/WR je $6t_{ck}$

t.j. pre f_{osc}= 12MHz =>

t_{ck} = 1/12MHz = 0.08333us

$6t_{ck}$ = 500ns

a15	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
1	**RS**	**x**	**x**	**x**	**R/W**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

a10 = 1/0 : R/W , a14 = 1/0 : data/inštrukcia , a15 = 1 (“displej”)

Logická adresa sa používala napr. pri procesoroch I8086 (Ixx86 – reálny mód)
16- bitové registre sa používali na adresovanie cez 20- bitovú adresnú zbernicu (adresný priestor 1MB)

Bázové adresovanie (segmentovaná pamäť)
- **segment** - časti pamäte premenlivej dĺžky
- **adresa v segmente** je relatívne vztiahnutá k začiatku (báze) segmentu
- **logická adresa** sa skladá z **báze segmentu** a **offsetu** (posunutia v segmente)

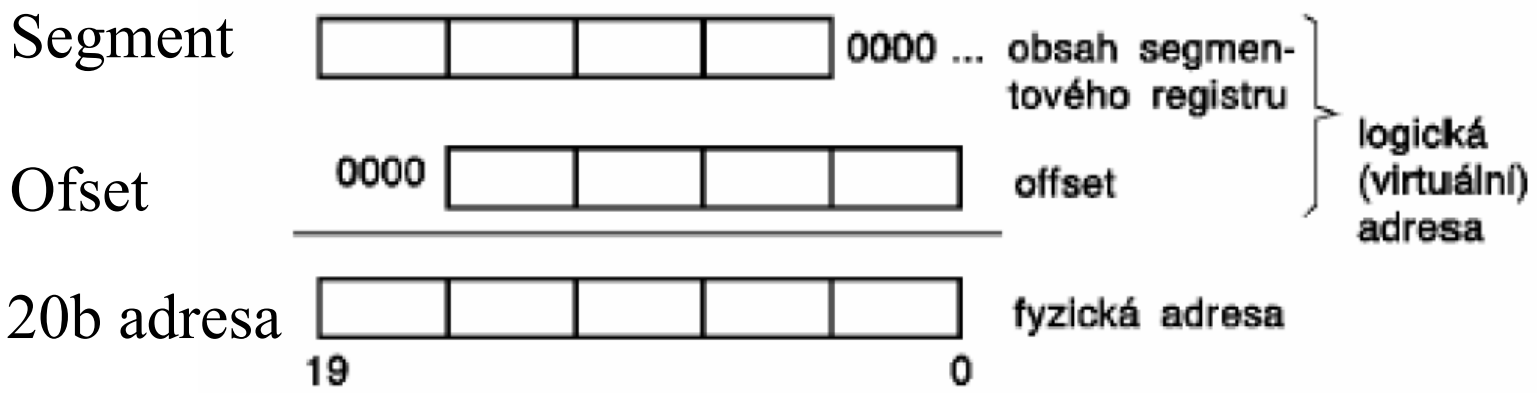
Báza segmentu je v segmentových registroch CS , SS,
DS - Data Segment a ES- Extra Segment(16-bitové registre)

- **logickú adresu** tvoria dvojice registrov napr.

CS:IP (Code Segment : Instruction Pointer)
SS:SP (Stack Segment: Stack Pointer)

- **fyzická adresa** sa vypočítavala v časti procesora (BIU)

(binárny obsah smerníka bol zhodný s logickou adresou)



Adresu zapisujeme: Segment:Ofset Pr.: 01A5:0012= 01A62
Začiatok segmentu na adresách 16*xS (xS – obsah segmentového registra), maximálna dĺžka segmentu 64kB)

Adresovanie pomocou tabuliek I80286 (16MB /1GB=2^{30})

Adresovanie:
- Reálny režim - rovnako ako 8086
- Chránený režim (protected mode) iné ako 8086
  Tabuľka deskriptorov (popisovačov) segmentu
  Logická adresa (pointer) **selektor:offset**

Segment: súvislý priestor pamäti veľkosti až 64kB, môže začínať na ľubovolnej adrese.
Modulárna štruktúra programu:
- Segment kódu
- Segment Dát
- Segment zásobníka

Segment (popisovač segmentu) je definovaný:
- báza segmentu (adresa začiatku segmentu)
-limit segmentu (dĺžka segmentu v slabikách – 1)
- prístupové práva a typ segmentu

Adresovanie pomocou tabuliek I80286

Pohyb v rámci segmentu: fyzická adresa = báza +16b offset

Ak hovoríme o organizácii pamäte, treba spomenúť pojem PROCES (Task).

Adresný priestor procesu možno rozdeliť na :

- Lokálny adresný priestor (použiteľné len procesom)
- Globálny adresný priestor (spoločné – napr. prekladač)

Virtuálna adresa

Virtuálna adresa je tvorená pomocou dvoch 16bitových zložiek.
Offset – chránený režim a reálny režim to isté.

Druhá zložka, chránený režim:“SELEKTOR SEGMENTU“

Adresovanie pomocou tabuliek I80286

Selektor. Segmentu:

-13b (8192kombinácií) index do tab. popisovačov segmentov lokálneho alebo globálneho adres. priestoru

-1b (indikátor tabulky - TI)

- TI = 1 – lokálny adr. priestor
- TI = 0 – globálny adr. priestor

-2b (úroveň oprávnenia - RPL)

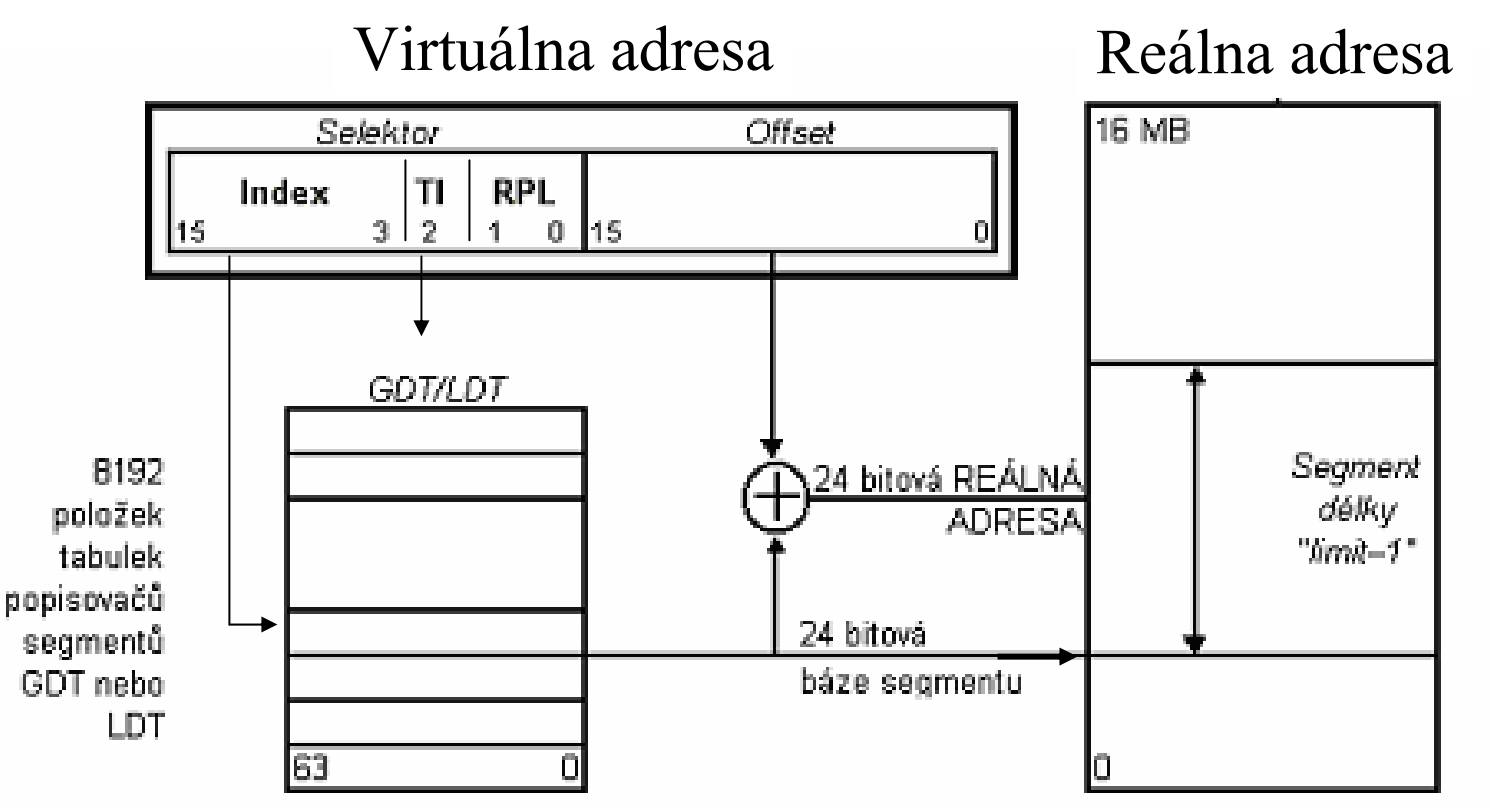
- ináč povedané, čo nám proces pri prístupe k SEGMENTU dovolí

Adresovanie pomocou tabuliek I80286

-deskriptory (8bytov)
- GDT – globálne pre systém,
- LDT lokálne pre aplikáciu

Tabuľky sú rezidentne uložené v pamäti. 32b adresa sa redukuje na 24b.
Štruktúra
2B – prázdne =0 , kvôli 386
1B - prístupové práva (R,W,X) (Read,Write,eXecute)
3B – báza segmentu (24-bitová adresa segmentu)⇒
16M adresný priestor
2B – limit segmentu (max. veľkosť 64kB)

Podľa obsahu bytu „prístupové práva“ delíme popisované segmenty na:
- Dátový segment (dáta aplikácie alebo zásobník)
- Inštrukčný segment (inštrukcie určené na spustenie)
- Systémový segment (je určený pre procesor a OS)
- Popisovač brány (riadenie prevedené na vyššiu úroveň, „prerušenia“, ..)



Pre I80386(4GB = 2^{32} /64TB)

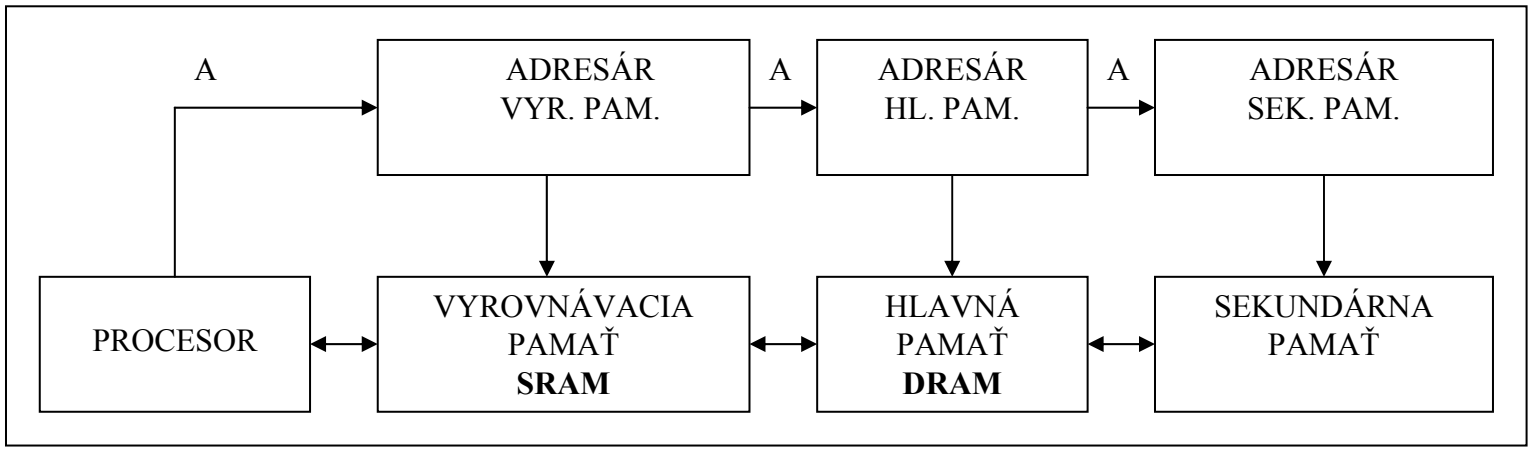
-Fyzická adresa: **4GB = 2^{32}**

-Virtuálna adresa: **64TB**

-deskriptor obsahuje okrem iného 32-bitovú bázu a 20- bitový limit

Vyrovnávacia pamäť (Cache memory)

Hlavná pamäť veľmi pomalá (relatívne k rýchlosti procesorov), preto sa vkladá ďalšia úroveň pamäte:
-vyrovnávacia pamäť (VP) (musí byť podstatne rýchlejšia ako HP cca 10 krát)
- slúži na urýchlenie najmä čítanie (aj zápis) do HP
- preto trojúrovňový pamäťový systém

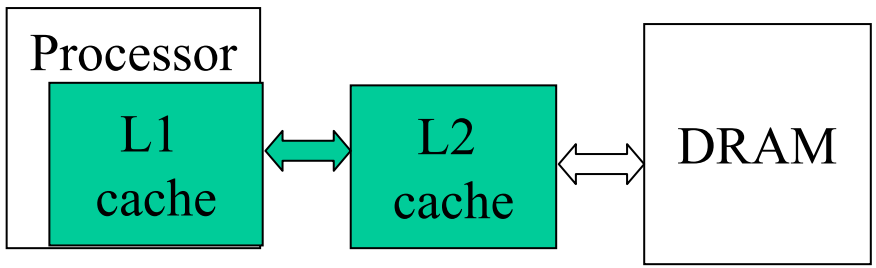


Word transfer

Block transfer

Prvýkrát (IBM S/360 model 85 r.v. 1968) – označenie „cache“ (skrýša), iný výrobcovia označujú ako „buffer memory“

- pre programátora je „ **úplne priehľadná**“ (nedá sa adresovať ani na úrovni strojového kódu) – činnosť VP riadi riadiaca jednotka (RJ)
- kapacita VP je **zlomkom kapacity** HP
- sú v nej uložené **kópie častí HP** (v trojúrovňovom systéme teoreticky môže byť zapísaná tá istá informácia 3-krát
- dnešné počítače majú dve cache pamäte. L1 cache je menšia a je súčasťou porcesora a L2 cache je väčšia a je zapojená medzi procesor a HP.



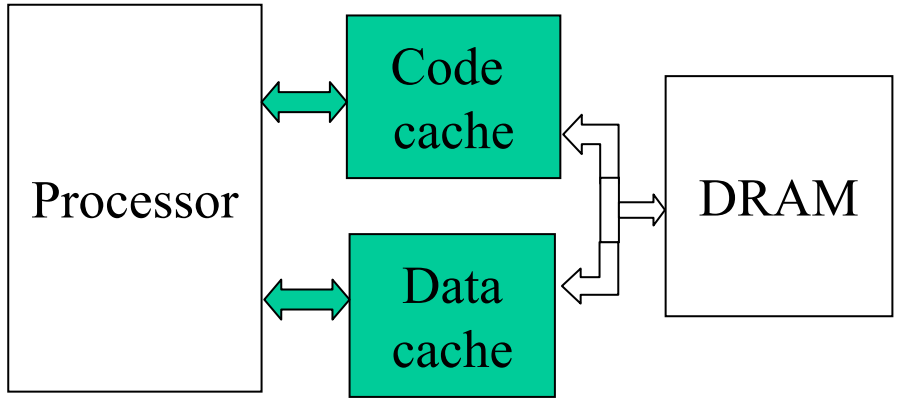
Princíp VP :
- obsahuje **adresár miest**, ktoré sa v danom čase nachádzajú vo VP
- podľa **obsahu adresára** sa dá rýchlo zistiť, či je daná informácia vo VP
- ak je tam, potom sa informácia prenáša do procesora z VP namiesto HP
- ak RJ zistí, že vo VP nie je požadovaná informácia, prečíta ju do CPU a VP
- očakáva sa vysoká pravdepodobnosť opakovaného čítania požadovanej informácie v krátkom časovom intervale
- okrem požadovanej informácie sa do VP presúva niekoľko susedných adries (**blok**) (dĺžka presúvaných blokov **8** až **32** bytov, extrémna veľkosť **128** bytov)

Definovanie bloku: ak je adresa HP 16-bitová a posledné 4-bity „nahradíme“ x-kami, získame blok veľkosti 16 bytov, slov.
Počet blokov pre náš pr.: $2^{(16-4)}$

Stratégia VP vychádza zo štatisticky overených vlastností programov pri práci s pamäťou. Sú to princípy:
-časovej lokality (vysoká pravdepodobnosť potreby informácie z tej istej adresy v krátkej budúcnosti
- a **miestnej lokality** (vysoká pravdepodobnosť potreby informácie z pamäťového miesta s adresou blízkou danej adrese), preto sa presúva blok so susednými bajtami

Zásadný problém – ako realizovať predvýber blokov do VP, aby **CPU čítal len z VP** – neexistuje uspokojivá stratégia:
- prvý presun až pri **požiadavke čítania** (on demand)
- ak je vyžiadané slovo, ktoré nie je vo VP, najskôr sa číta žiadané slovo do procesora, až potom sa číta zbytok bloku do VP. Ešte raz – celý blok.

Ďalšia otázka, čo presúvať do VP (pôvodne sa presúvalo všetko, čo bolo požadované (údaje aj inštrukcie)), v súčasnosti sa presadzuje koncepcia oddeleného čítania údajov a inštrukcií, čo umožňuje súčasné čítanie údajov a inštrukcií.



Adresovanie vyrovnávacej pamäte

- Informácie uložené vo VP sa volajú **reálnou adresou** t.j. ich adresou v HP
- VP je podstatne menšia, adresa nemôže priamo adresovať miesto vo VP, ale musí byť **prekladací (konverzný algoritmus)** na hľadanie informácie vo VP
-Stačí vymyslieť **algoritmus hľadania bloku** vo VP (hľadaná informácia je súčasťou bloku vo VP)
- Na vyhľadanie v bloku stačia najnižšie bity adresy (označujú sa ako „**číslo slova v bloku**“)

Ideálna cache obsahuje toľko položiek, že sa do nej zmestí celá HP, adresovanie položky cache a HP by bolo zhodné, len by bola VP obrovská a skoro prázdna

Pri redukcii počtu položiek VP - opäť problém prehľadávania obsahu VP, sekvenčné prehľadávanie je pomalé.

VP - Cache pamäte bývajú organizované ako tzv. **asociatívne pamäte**.
Asociatívne pamäte sú tvorené tabuľkou (-ami), ktorá obsahuje vždy:
- stĺpec s názvom: **tagy** (kľúče), slúžia na vyhľadávanie v asociatívnej pamäti. Tag je časťou pôvodnej adresy.
- stĺpec **data** – uchovávané údaje
- stĺpec s názvom **info**rmácie o stave správnej funkcie pamäte. (platnosť/neplatnosť uložených údajov, „LRU“)

Riadok tabuľky = „trieda“ = „rám“ =slot (tag, data, info)

Pr.:

Hlavná pamäť:

- 64kB (16 bitov adresy)
- 8kB blokov, každý po 8 B

Cache:

- 8B blok (3 bity adresy)
-1kB (1024B) blok = 8B => 128 slot-ov (rámov)

Priamo mapovaná cache pamäť (Direct mapping)

C = 128 (počet slotov v cache)

M = 8K (počet blokov v HP)

C << M

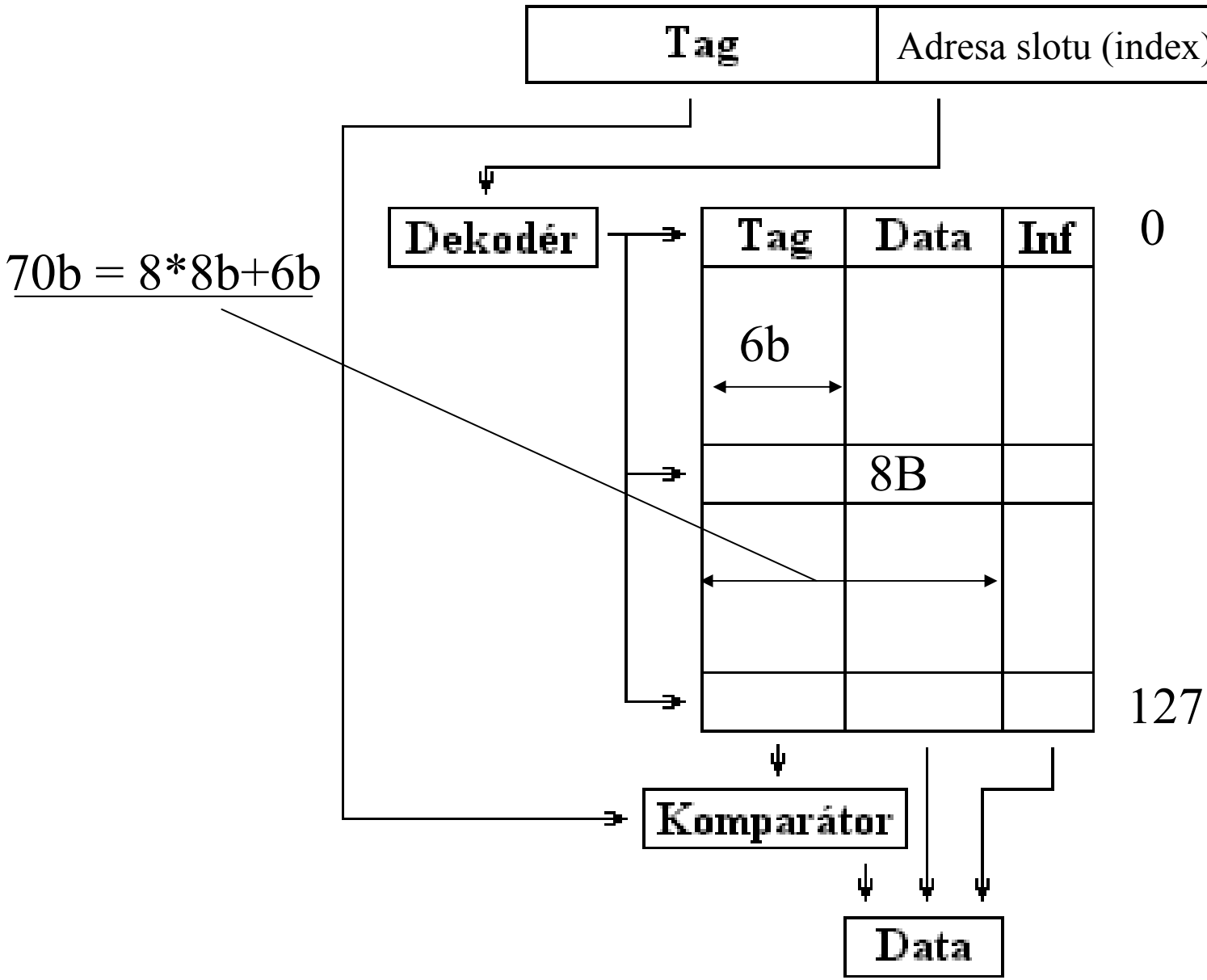
S – poradové číslo slotu

Priradenie blok HP do slotu Cache.

S = (adresa bloku HP) modulo (C)

T.j. Slotu 0 patria bloky: 0, 128,256, … 8 064

A Slotu 1 patria bloky: 1, 129,257, … 8 065, atd.


Tag
Adresa slotu (index)
Dekodér
Tag
Data
Inf
0
70b = 8*8b+6b
6b
8B
127
Komparátor
Data
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

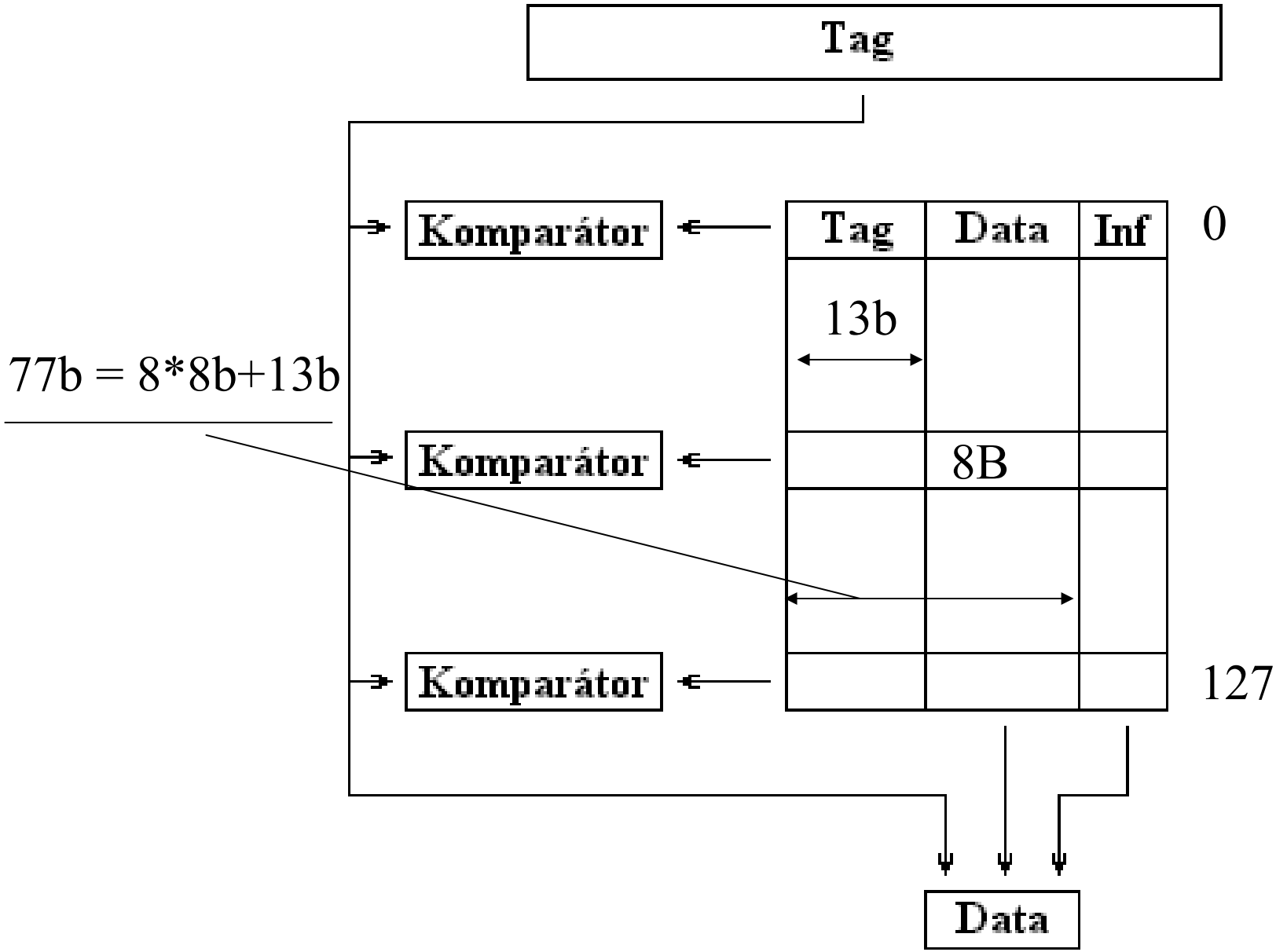
Priamo mapovaná cache pamäť

Nevýhody: blok HP má pevnú polohu v cache
T.j. ak program pendluje medzi dvoma blokmi v tom istom slote, tak je nám cache „nepotrebná“.
Výhody: jednoduché a lacné

Plne asociatívna cache pamäť

Tag – 13b
Blok – 3b
Blok môže byť presunutý do ľubovolného slotu.

Plne asociatívna cache pamäť



Plne asociatívna cache pamät

Nevýhody:
– zložitý blok riadenia cache
– požadujem veľa komparátorov
– veľká pamäť pre pamätanie si tagov
– nejasný mechanizmus výmeny blokov
Výhody:
– odstraňuje nevýhody priamo mapovanej cache

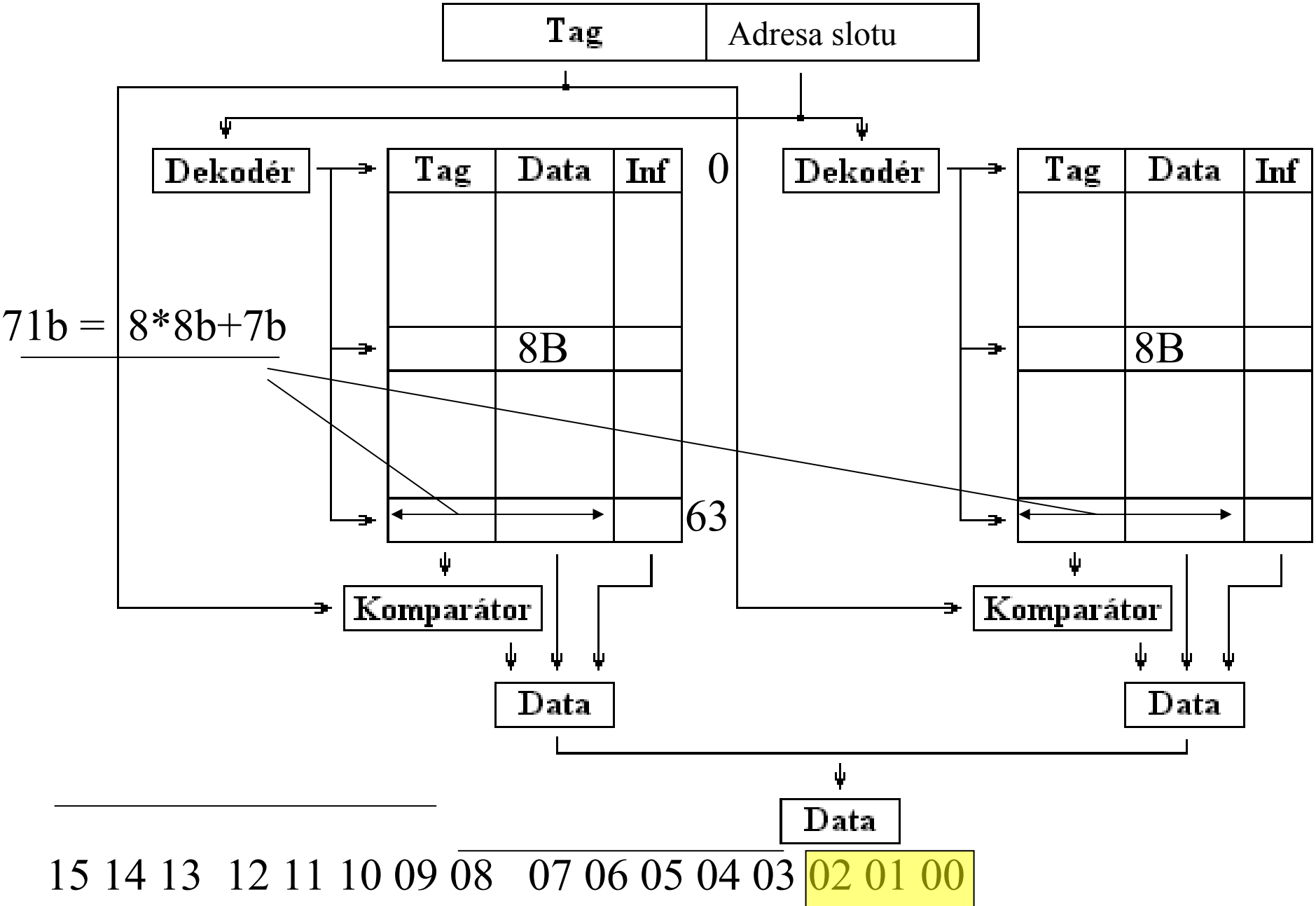
!!! Prakticky sa plne asociatívna pamäť nepoužíva. !!!

Asociatívna cache pamäť s obmedzením (stupňa n)

Odstraňuje nevýhody oboch predchádzajúcich.
Cache je rozdelená do **I** skupín a každá má **J** slotov.

C = I * J
Tento algoritmus umožňuje mapovať blok s adresou A do do Ľubovolného slotu v skupine.
Pre „krajné“ hodnoty:
I=1, J=C – plne asociatívna pamäť
I=C, J=1 – priamo mapovaná cache pamäť
Najčastejšie – najjednoduchšie sa realizuje takáto cache pre n=2

Asociatívna cache pamäť s obmedzením (stupňa n=2)



Stratégia výmeny údajov medzi VP a HP

Podobné problémy ako virtuálnom adresovaní

Efektívnosť VP sa vyjadruje ako pomer
(**počet výpadkov VP**)/(**počet prístupov k VP**)

Po spustení programu (úlohy) na začiatku je toto číslo veľké, po cca **100 000** prístupoch sa ustáli na hodnote, ktorá závisí od **stratégie uvoľňovania blokov** a **stupňa asociativity** (pomer býva blízky **0,05**)

Používajú sa stratégie uvoľňovania **LRU**, **FIFO** a **náhodná**, preferuje sa obvodová jednoduchosť a najmä rýchlosť

Stratégia výmeny údajov medzi VP a HP

LRU (Least Recently Used – **najdlhšie nepoužívaný**)

Vyraďuje najdlhšie nepoužívaný blok dát
– zložitá realizácia
– používajú sa zjednodušené varianty stratégie LRU

FIFO (First In First Out) – v podstate **fronta** – vyraďuje sa blok dát, ktorá je v cache **najdlhšie**,

Zabezpečenie zhody údajov vo VP a HP

- **write through** – zapisovanie z procesora do VP a súčasne aj do HP (pomalé)
- **write back** – zapisuje len pri vyraďovaní bloku (3 spôsoby)
  - **zapisuje vždy**
  - **zapisuje len pri zmene obsahu**
  - **zapisuje len pri zmene** cez **pomocnú pamäť bloku** (zápis do pamäte sa uskutoční z pomocnej pamäte po prečítaní nového bloku) (**najrýchlejšie**)

Efektívnosť VP – odhad, ak je rýchlosť **VP 10x** väčšia ako HP potom v strednej hodnote je rýchlosť (čítanie/zápis) **cca 6x** vyššia pri použití VP oproti systému bez VP.

Literatúra:

1. **Hlavička, J.:Architektura počítačú. ČVUT 1998**