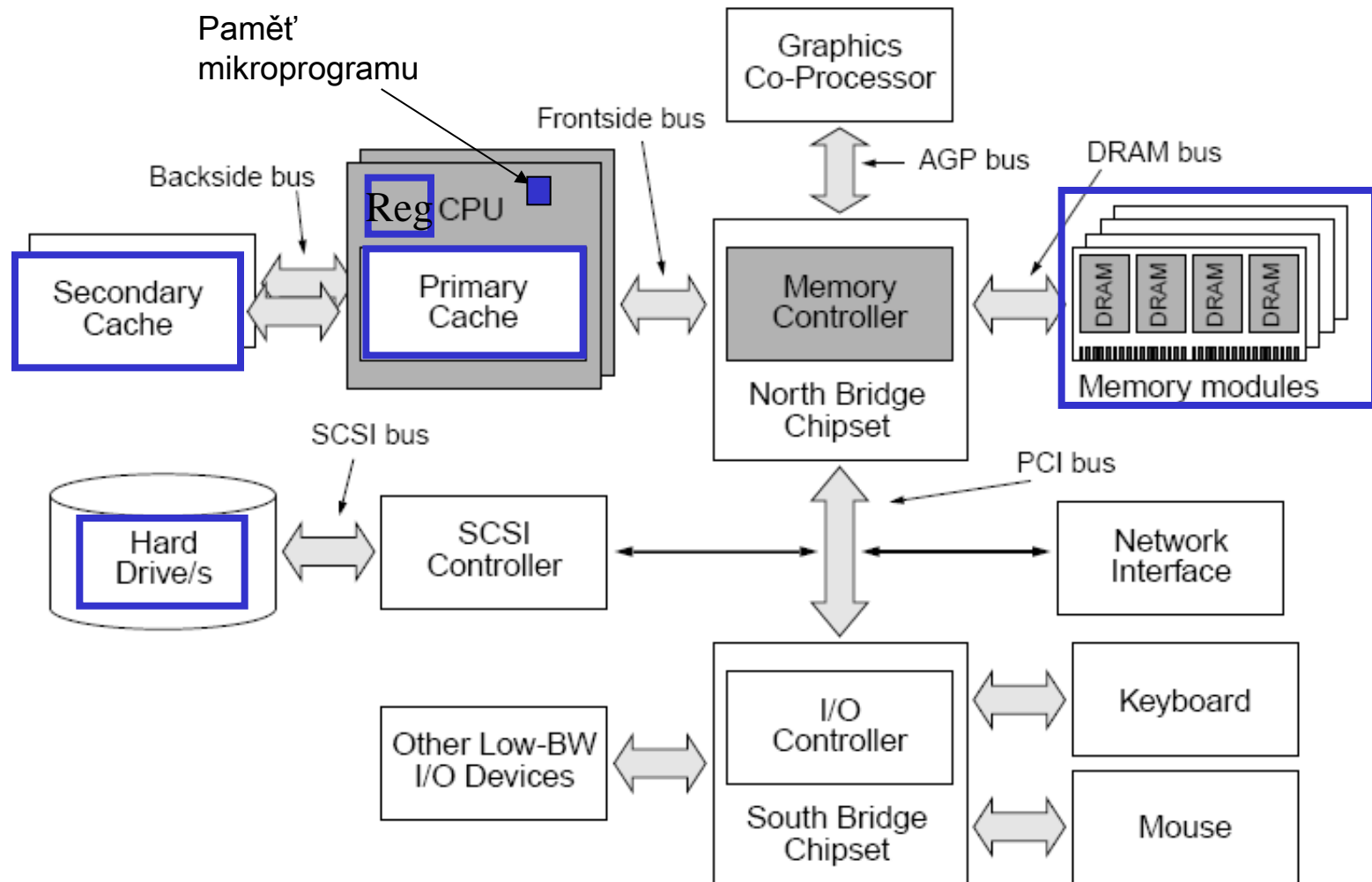


# Paměti

Návrh počítačových systémů  
INP 2019

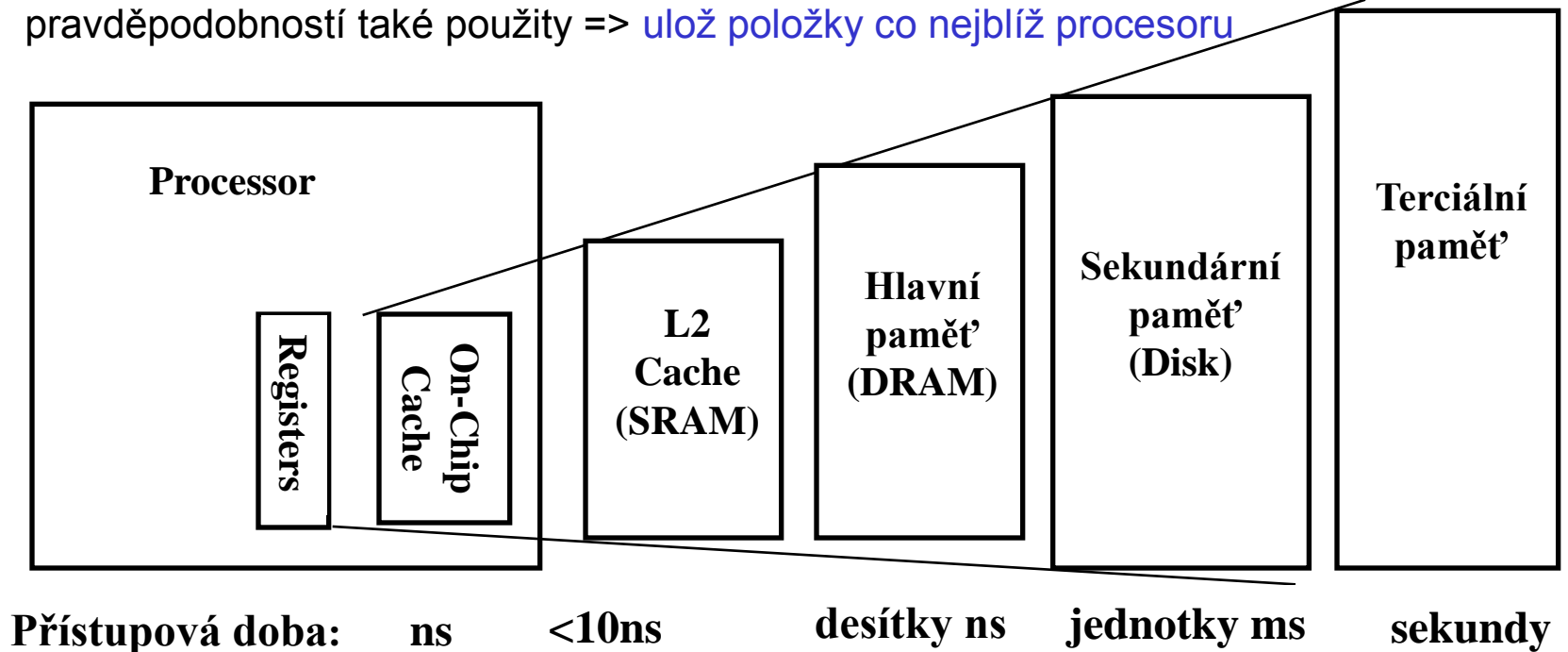


# Paměťová hierarchie v běžném počítači



# Proč paměťová hierarchie?

- chceme maximalizovat výkonnost počítače (tj. poměr výkon/cena)
- potřeba **nevolatilní** paměti, která nepotřebuje napájecí napětí
- **časová lokalita** - pokud procesor používá nějakou položku v paměti, je vysoká pravděpodobnost, že ji bude používat znovu => **ulož položku co nejbliž procesoru**
- **prostorová lokalita** – pokud procesor pracuje s nějakou položkou v paměti, potom položky, které jsou umístěny v paměti v blízkosti této položky, budou s vysokou pravděpodobností také použity => **ulož položky co nejbliž procesoru**



**Kapacita:** stovky B   kB   jednotky MB   jednotky GB   jednotky TB   desítky TB

**Cena/MB:**            nejvyšší                            střední                            nejnižší

# Paměti - typy a parametry

Paměťové prvky se používají v počítači na všech hierarchických úrovních v těchto funkcích:

- **vnitřní paměť procesoru** - registry, registrové sady, zásobníky, fronty, tabulky pro různé účely a paměť mikroprogramů v řadiči procesoru,
- **hlavní paměť** včetně rychlých vyrovnávacích pamětí,
- **vnější paměti**.

Ve všech těchto hlavních skupinách se používá mnoho různých typů pamětí, které se zásadně liší svými **parametry**, **fyzikálními principy**, **způsobem výběru dat** a dalšími vlastnostmi.

Parametry pamětí:

- **kapacita** se udává ve tvaru respektujícím organizaci paměti, tedy jako součin *počtu paměťových míst* a *délky paměťového místa*, tj.  $N \times n$  bitů, jako např. 16K x 1 bit, 64K x 1Byte, 4M x 4Byty atd.
- **přístupová doba**  $t_a$  je doba od zahájení čtení (tj. udáním adresy paměťového místa a povelu R) po získání obsahu paměťového místa.
- **doba cyklu** je doba od zahájení čtení nebo zápisu (Č/Z) až po skončení této operace, kdy je možno spustit další operaci Č/Z.

## Další parametry pamětí

*Přenosová rychlost* je parametr udávající počet datových jednotek (bitů, bytů atd.) přenášených do nebo z paměti za sekundu, např. 1Gb/s u disku.

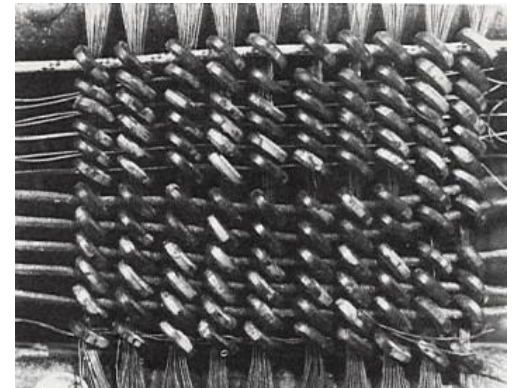
Pro cenové rozvahy je důležitým parametrem *cena/bit*.

K dalším významným parametrům patří *chybovost* paměti udávaná např. v počtu chyb na 1000 hodin a *poruchovost*, nejčastěji popisovaná parametrem *střední doba mezi poruchami*.

*Výkonnost* je u pamětí udána parametry: kapacita, přístupová doba a přenosová rychlost.

# Klasifikace pamětí podle fyzikálního principu

- paměti polovodičové - bipolární a
  - unipolární
- magnetické
  - disketové,
  - diskové,
  - páskové
  - atd.
- optické
  - CD
  - DVD
- magnetooptické
- nekonvenční, např. molekulární



feritová paměť z IBM 405

Mnoho typů pamětí zmizelo a další se objevují.

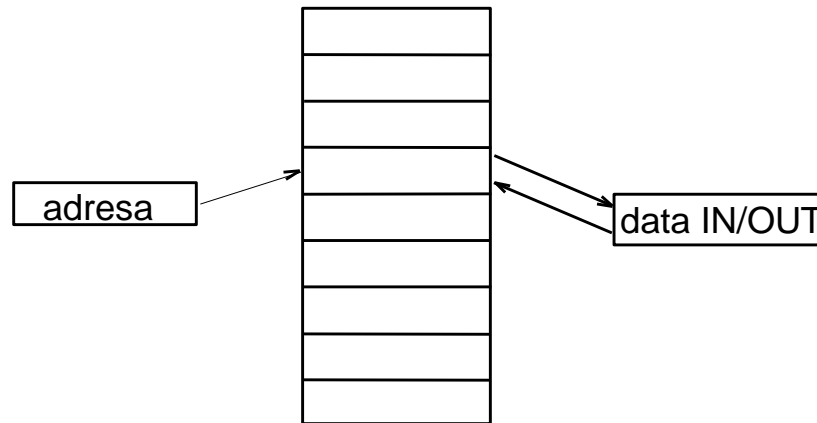
# Přístup k datům – libovolný a sériový

Pokud přístupová doba nezávisí na umístění požadované položky, jde o paměť *s libovolným přístupem* RAM (Random Access Memory).

Paměť *se sériovým přístupem* SAM (Serial Access Memory) vybavuje položky s různou dobou přístupu podle toho, jak dlouho to trvá, než se paměťové médium přisune k čtecí hlavě.

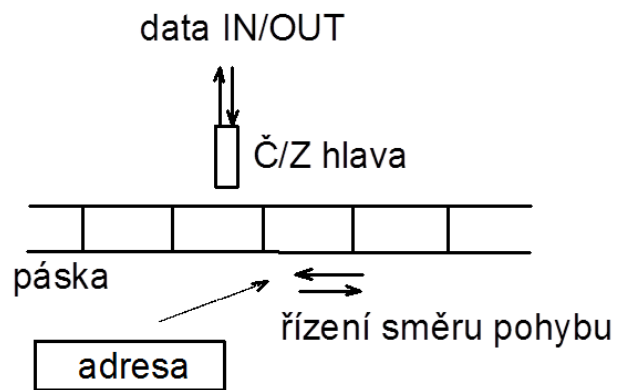
Disk s několika záznamovými povrchy je reprezentantem *smíšeného přístupu*; výběr záznamového povrchu je libovolný přístup, vystavování hlavy na požadovanou stopu a otáčení disku při čekání na požadovaný záznamový sektor je sériový přístup. Čistě sériový přístup má magnetická páska.

## Libovolný přístup RAM

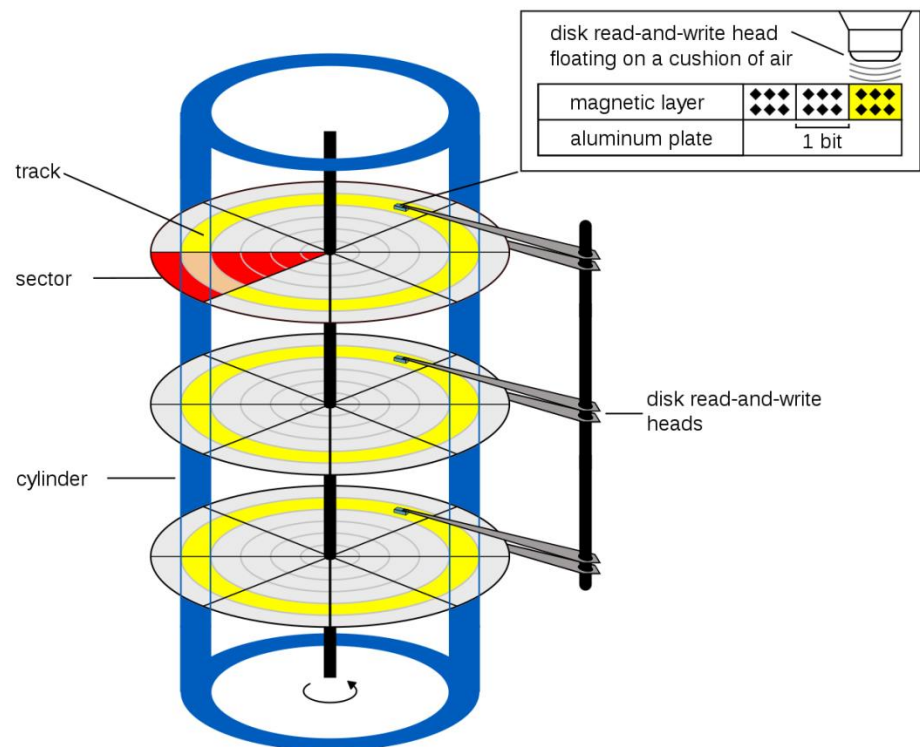


a) RAM

## Sériový přístup SAM

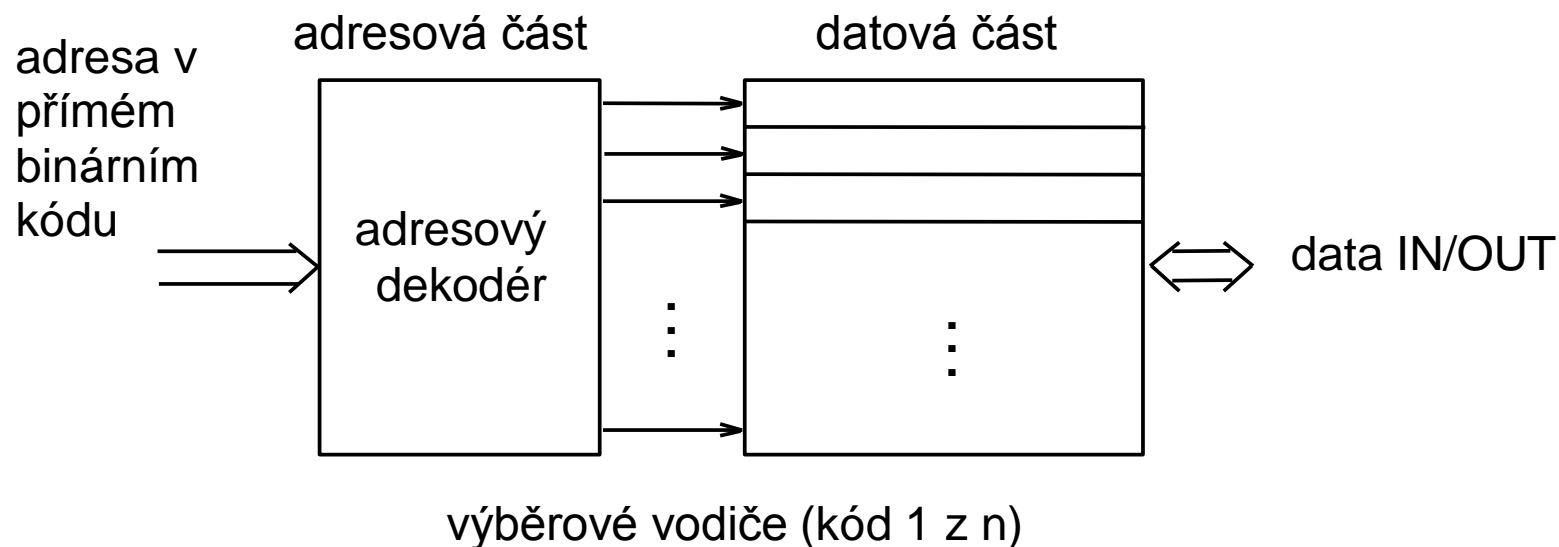


## Smíšený přístup





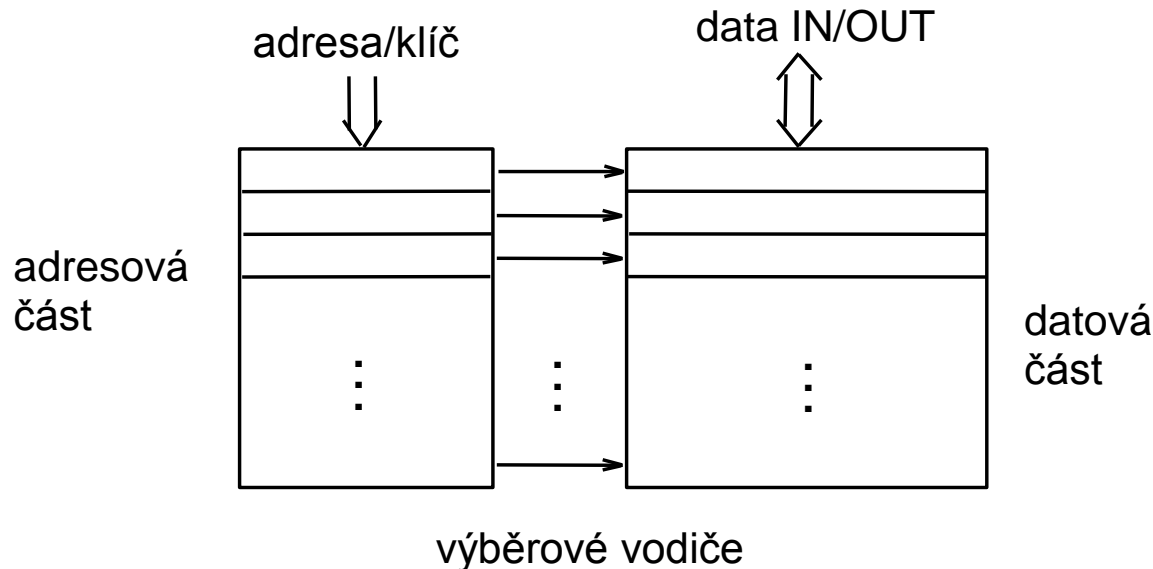
# Výběr z paměti – adresový



a) adresový výběr

Paměťová místa jsou uspořádána podle adres vzestupně, *adresový prostor je uspořádaný a souvislý*.

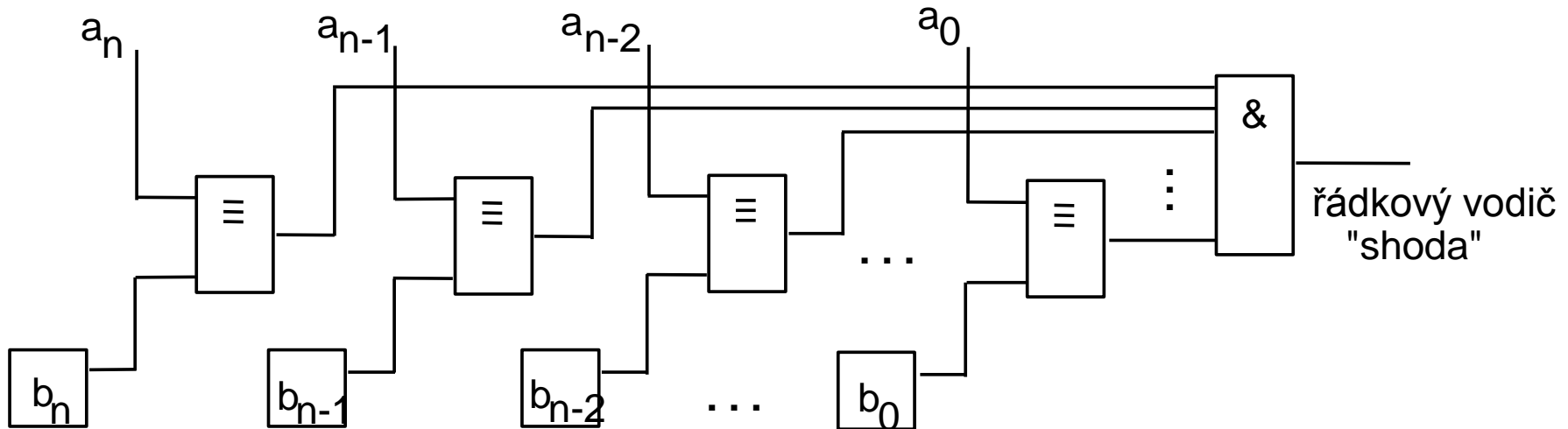
# Výběr z paměti - asociativní



b) asociativní výběr

- U **asociativního výběru** jsou v adresové části paměti poznamenány adresy paměťových míst datové části.
- Paměťová místa mohou být vzhledem k adresám uspořádána libovolným způsobem, některým adresám nemusí odpovídat žádné paměťové místo.
- Adresový prostor je **neuspořádaný a nesouvislý**. Výběrový vodič se aktivizuje na základě shody hledané adresy (klíče) s adresou daného řádku. Navíc se může pomocí *masky adresy* určit, které bity adresy se mají při porovnávání použít, a které ne. Proto je vhodné říkat adrese **klíč**.
- Princip asociativního výběru vyžaduje, aby ve všech řádcích adresové části paměti byl **komparátor adres - klíčů**.

# Komparátor adres - klíčů asociativní paměti



$a$  – klíč na vstupu paměti dodaný uživatelem

$b$  – jedna konkrétní hodnota klíče v paměti

Uvedená obvodová struktura musí být zopakována  $k$ -krát,  
kde  $k$  je počet položek uložitelných v paměti.

# Měnitelnost obsahu paměti – R/W

- **RWM** - Read/Write Memory - paměť umožňující čtení i zápis
- **ROM** - Read Only Memory - paměť umožňující pouze čtení, zapisovat nelze.

Varianty:

- **PROM** - programovatelná ROM; „čistá“ nenaprogramovaná paměť umožňuje jedno naprogramování, další změna již není možná.
- **EPROM** - vymazatelná PROM; naprogramovanou paměť je možné vymazat a znovu naprogramovat. Paměti s tímto označením se mažou ultrafialovým zářením.
- **EEPROM** - elektricky vymazatelná PROM. Zde je řada modifikací podle toho, zda je možno mazat vybraný řádek, nebo pouze celou paměť, jak rychle proces mazání probíhá atd. (např. Flash EEPROM).

# Stálost obsahu paměti

Podle energetické závislosti dělíme paměti na *nevolatilní* a *volatilní*. Je-li záznam stálý i po vypnutí napájecího napětí, jde o nevolatilní paměť.

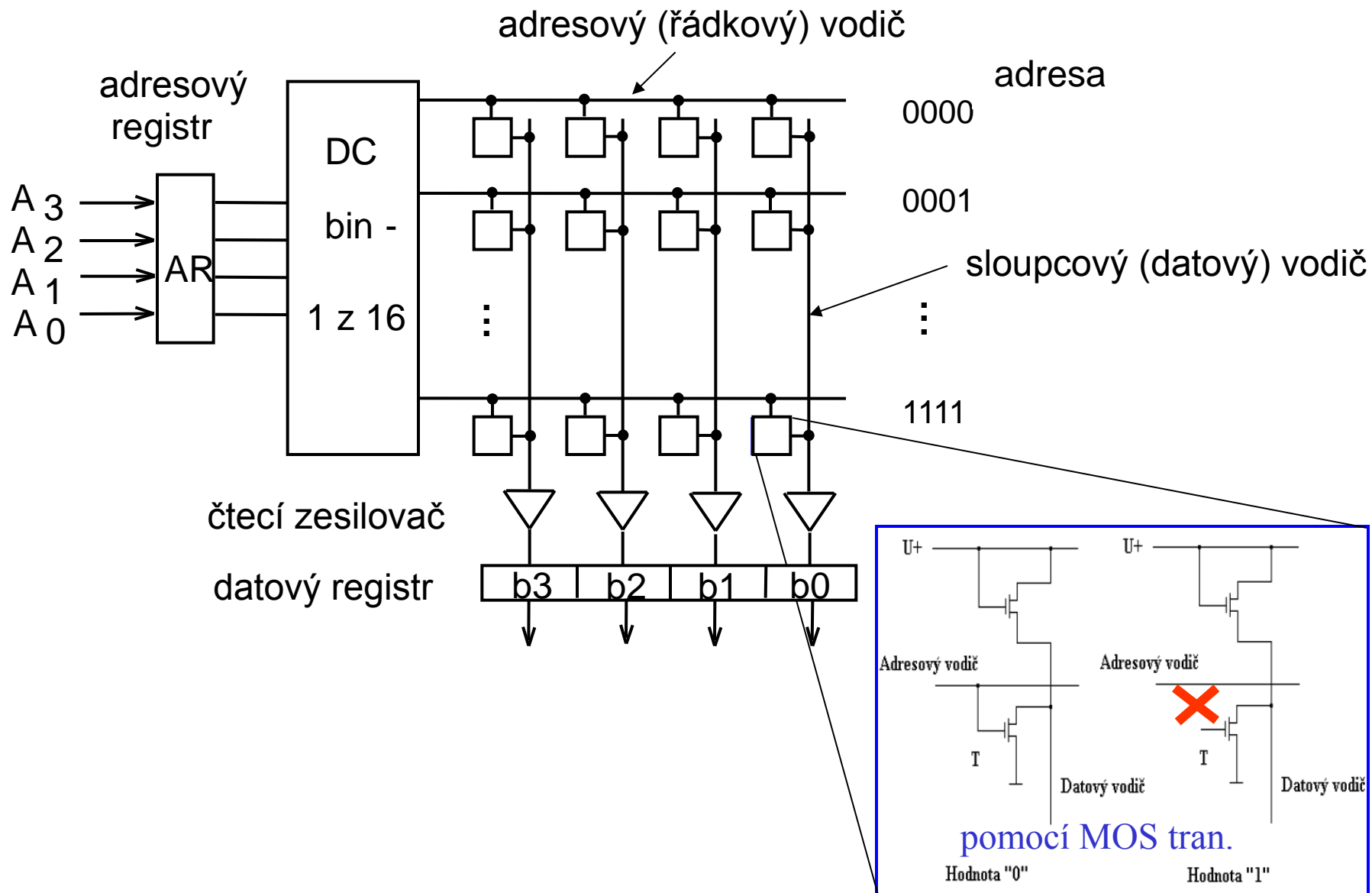
Některé fyzikální principy vedou na paměť, u které se čtením záznam vymaže. Znamená to, že se po cyklu čtení musí zařadit vždy cyklus zpětného zápisu. Takové paměti se označují jako *destruktivní*.

Podle *doby uchování informace* dělíme paměti na statické (*SRAM*), které při dodržení jistých provozních parametrů drží informaci libovolně dlouho, a dynamické (*DRAM*), které "zapomínají", a to docela rychle. U tohoto typu pamětí se proto musí závčas informace *obnovit* (*refresh*).

# Polovodičové paměti

- ROM
  - PROM, EPROM
- RWM
  - RAM
    - SSRAM, SRAM
  - DRAM
    - FPM DRAM, EDO DRAM, BEDO DRAM
    - SDRAM
    - DDR SDRAM, DDR2 SDRAM, DDR3 SDRAM
- EEPROM a FLASH

# Struktura paměti ROM 16 x 4b



# Komentář k příkladu paměti ROM

Čtyřbitová binární adresa se zachycuje do **adresového registru** AR a dále se dekóduje do kódu 1 z 16, což znamená, že pro jistou adresu se aktivizuje příslušný adresový vodič. Obsah vybraného čtyřbitového adresového místa se objeví na **sloupcových datových vodičích** a po zesílení pomocí **čtecích zesilovačů** ČZ se zapíše do výstupního **datového registru** DR.

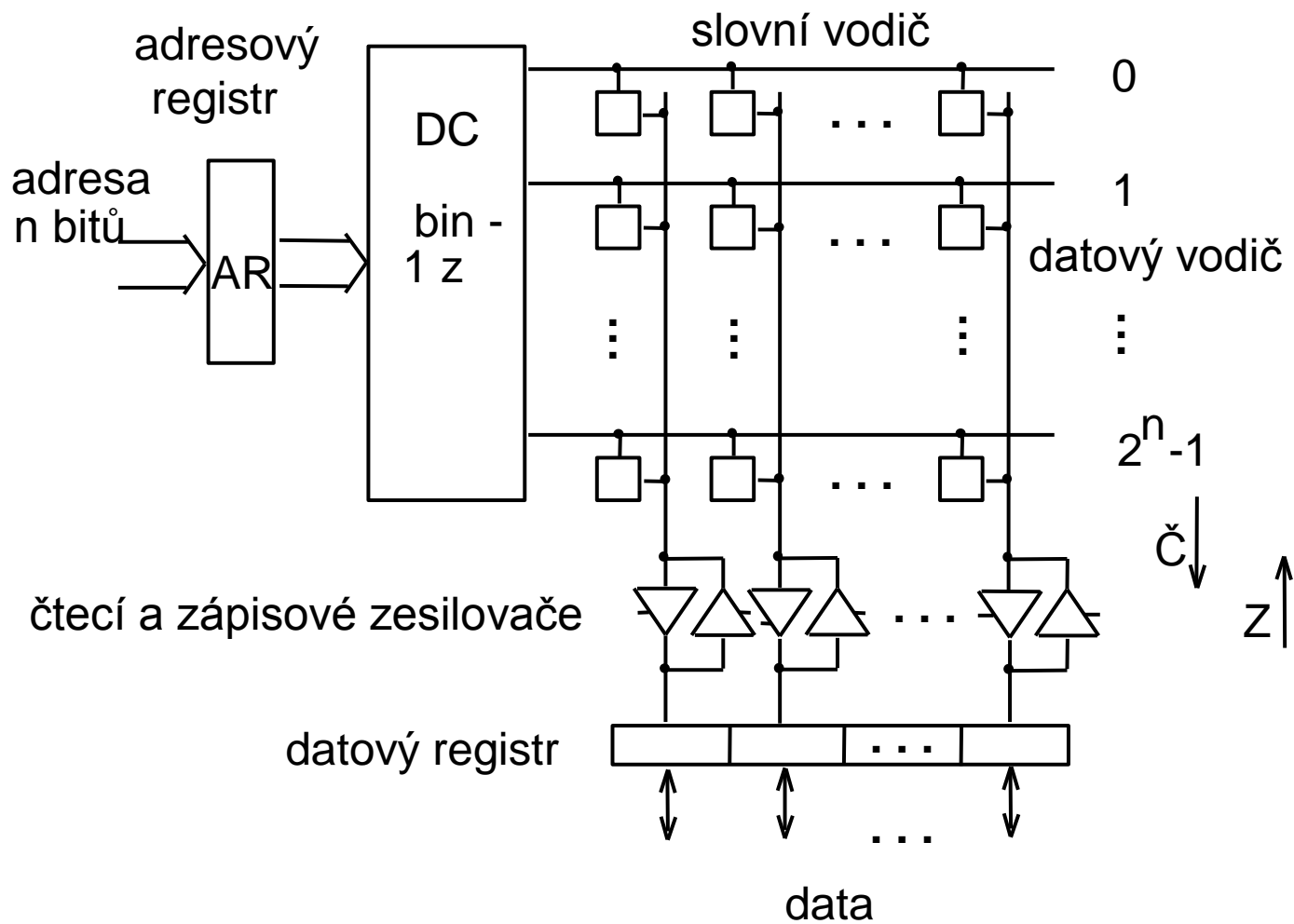
Ve funkci paměťových prvků pamětí ROM se v historii počítačů vystřídaly všechny základní pasivní i aktivní elektrické prvky. Byly tak použity rezistory, indukčnosti, transformátory, feritová jádra, kapacitory, diody, tranzistory bipolární i unipolární.

Nejrozšířenější typ **Flash** využívá kapacity izolovaného hradla tranzistoru MOS.

Historie: U pamětí PROM se programovalo přepalováním chromniklových nebo křemíkových propojek.

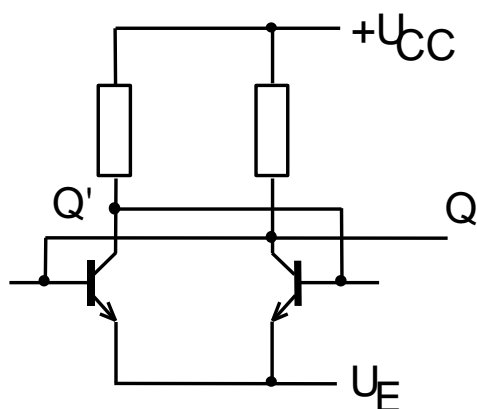


# Struktura paměti RWM

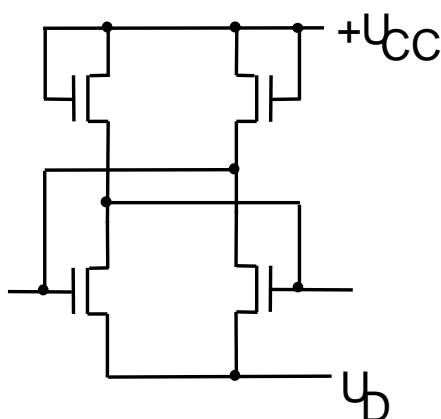


# Komentář k RWM a typy paměťových členů

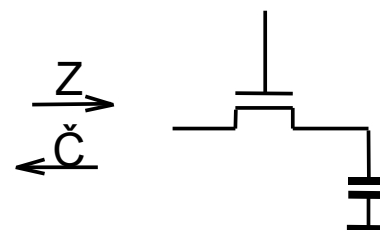
Struktura paměti RWM (nepřesně označovaná jako RAM) je principiálně velmi podobná struktuře paměti ROM. Rozdíl je v tom, že sloupcové datové vodiče nyní přenášejí data **obousměrně**, a jsou doplněny zápisové zesilovače ZZ. Rovněž vnější datová sběrnice je obousměrná.



a) Bipolární statický člen



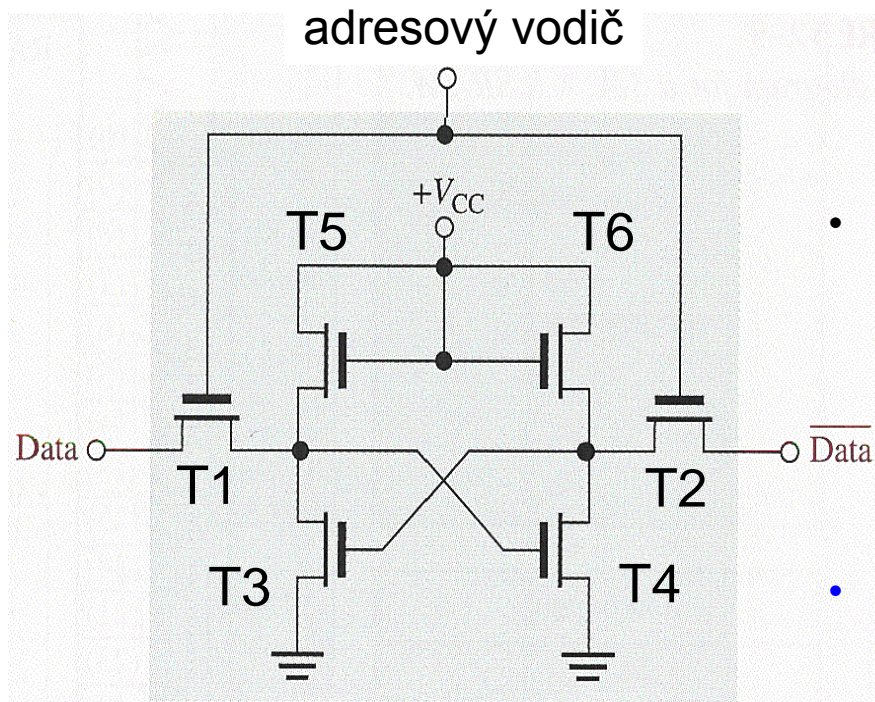
b) Unipolární statický člen



c) Unipolární dynamický paměťový člen DRAM

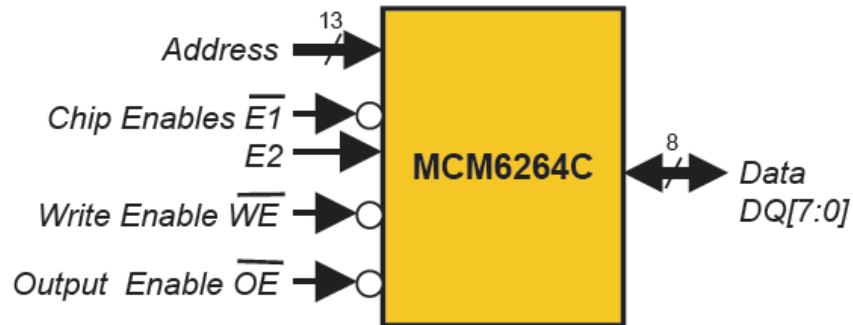
Každý paměťový člen má jiný způsob čtení a zápisu informace. Podle typu použitého paměťového členu se proto modifikuje vnitřní zapojení paměťové matice, čtecích a zápisových zesilovačů, případně se modifikuje celková struktura paměti. U statických paměťových členů se zápis a čtení provádí pomocí **dvojic** sloupcových datových vodičů, které nesou komplementární proudové nebo napěťové impulsy.

# Paměťová buňka SRAM (Static RAM)



- Vodič **Data** je určený k zápisu dat do paměti. Vodič označený jako **-Data** se používá ke čtení. Hodnota na tomto vodiči je vždy opačná než hodnota uložená v paměti => na konci procesu čtení je nutno ji ještě negovat.
- Při **zápisu** se na adresový vodič umístí hodnota logická 1, na vodič Data se přivede zapisovaná hodnota (např. 1). Tranzistor T1 se otevře => jednička na vodiči Data otevře tranzistor T4 => uzavře se tranzistor T3. Tento stav obvodu představuje uložení hodnoty 0 do paměti.
- **Čtení**: na adresový vodič je přivedena hodnota logická 1 => otevřou se tranzistory T1 a T2. Jestliže byla v paměti zapsána hodnota 1, je tranzistor T4 otevřen (tj. na jeho výstupu je hodnota 0), čtenou hodnotu obdržíme na vodiči -DATA. V případě uložené hodnoty 0 - tranzistor T4 je uzavřen (tj. na jeho výstupu je hodnota 1).

# Př. SRAM (8k x 8b)



**Same (bidirectional) data bus used for reading and writing**

**Chip Enables ( $\overline{E1}$  and  $E2$ )**

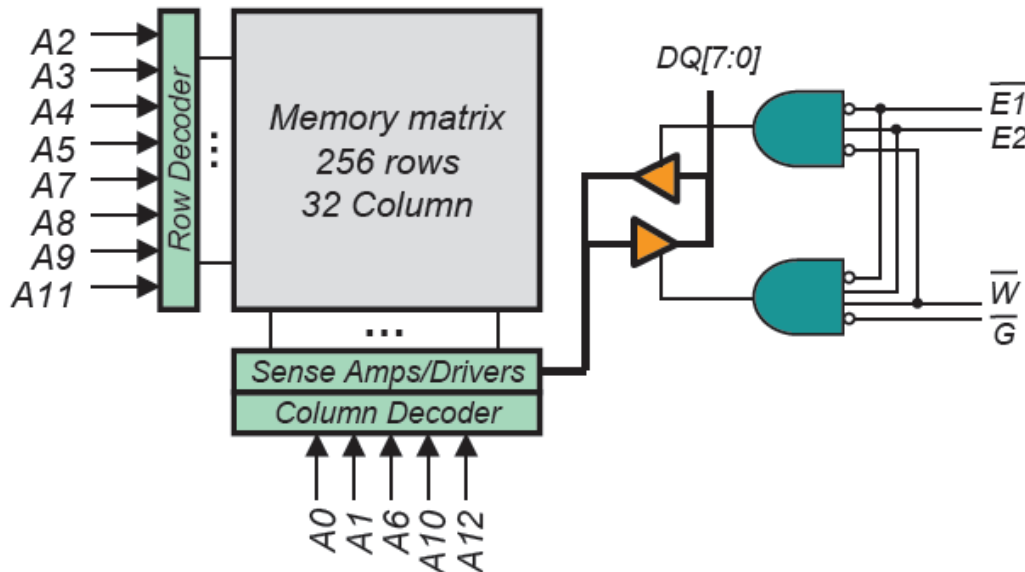
$E1$  must be low and  $E2$  must be high to enable the chip

**Write Enable ( $\overline{WE}$ )**

When low (and chip enabled), values on data bus are written to location selected by address bus

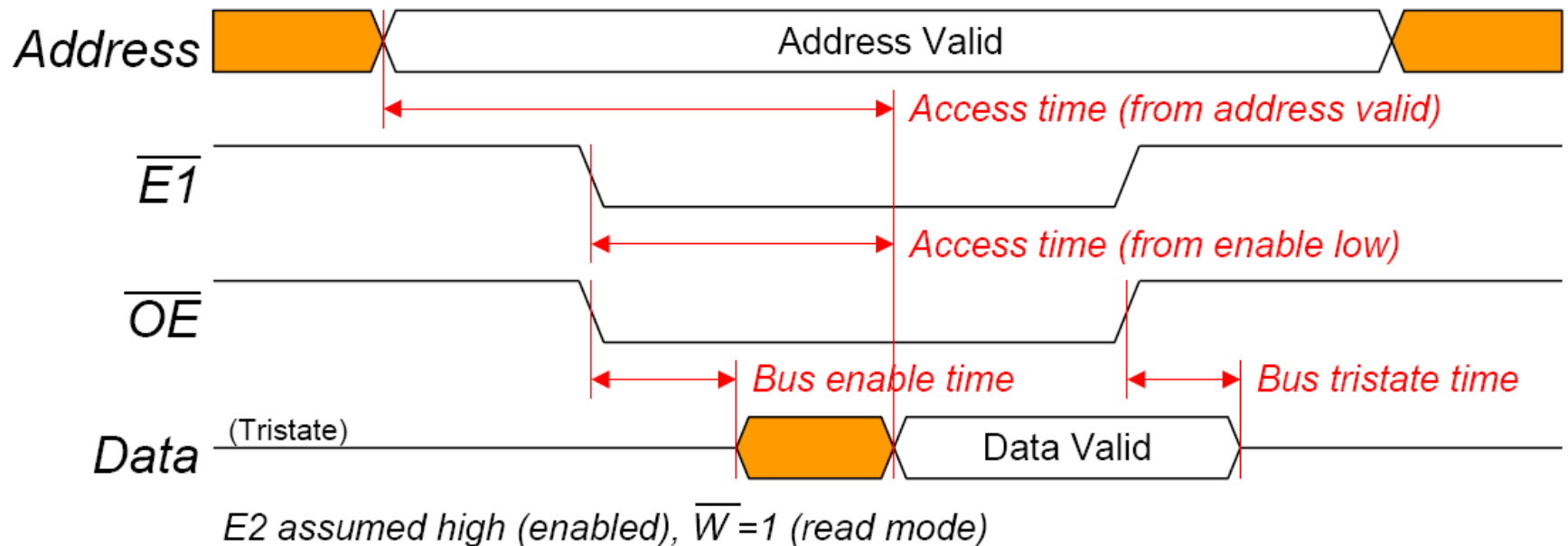
**Output Enable ( $\overline{OE}$  or  $\overline{G}$ )**

When low (and chip is enabled), data bus is driven with value of selected memory location

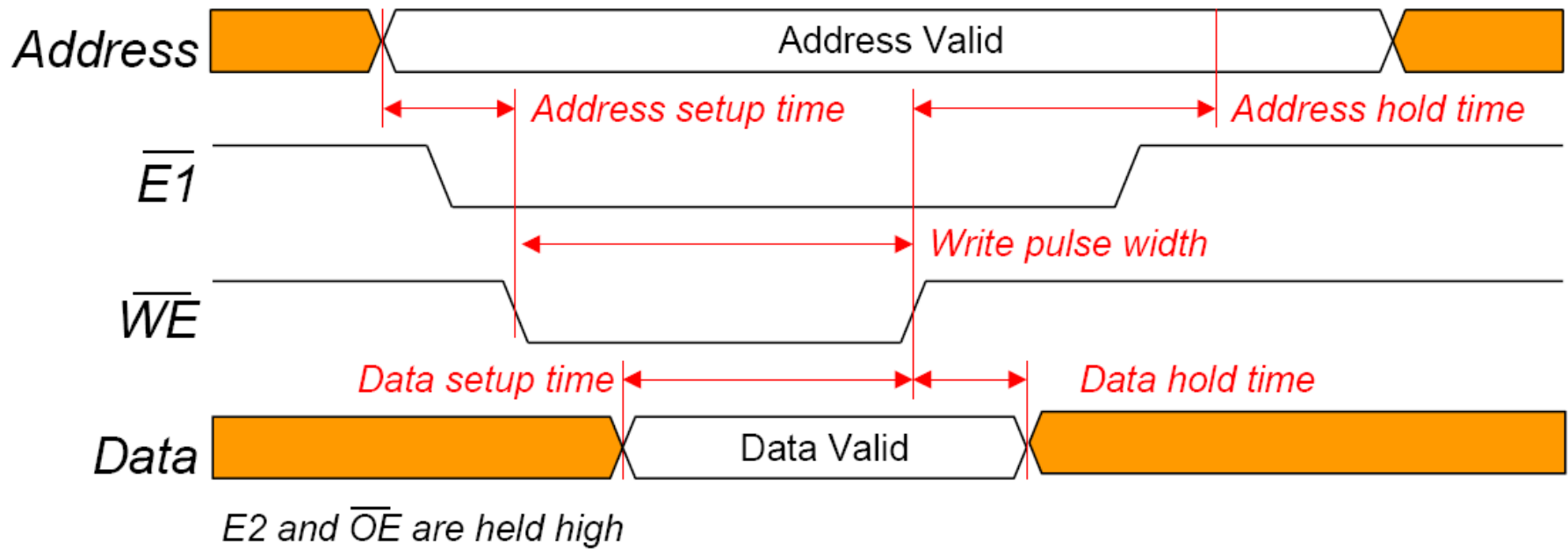


NC	1	28	V <sub>CC</sub>
A12	2	27	$\overline{W}$
A7	3	26	E2
A6	4	25	A8
A5	5	24	A9
A4	6	23	A11
A3	7	22	$\overline{G}$
A2	8	21	A10
A1	9	20	$\overline{E1}$
A0	10	19	DQ7
DQ0	11	18	DQ6
DQ1	12	17	DQ5
DQ2	13	16	DQ4
V <sub>SS</sub>	14	15	DQ3

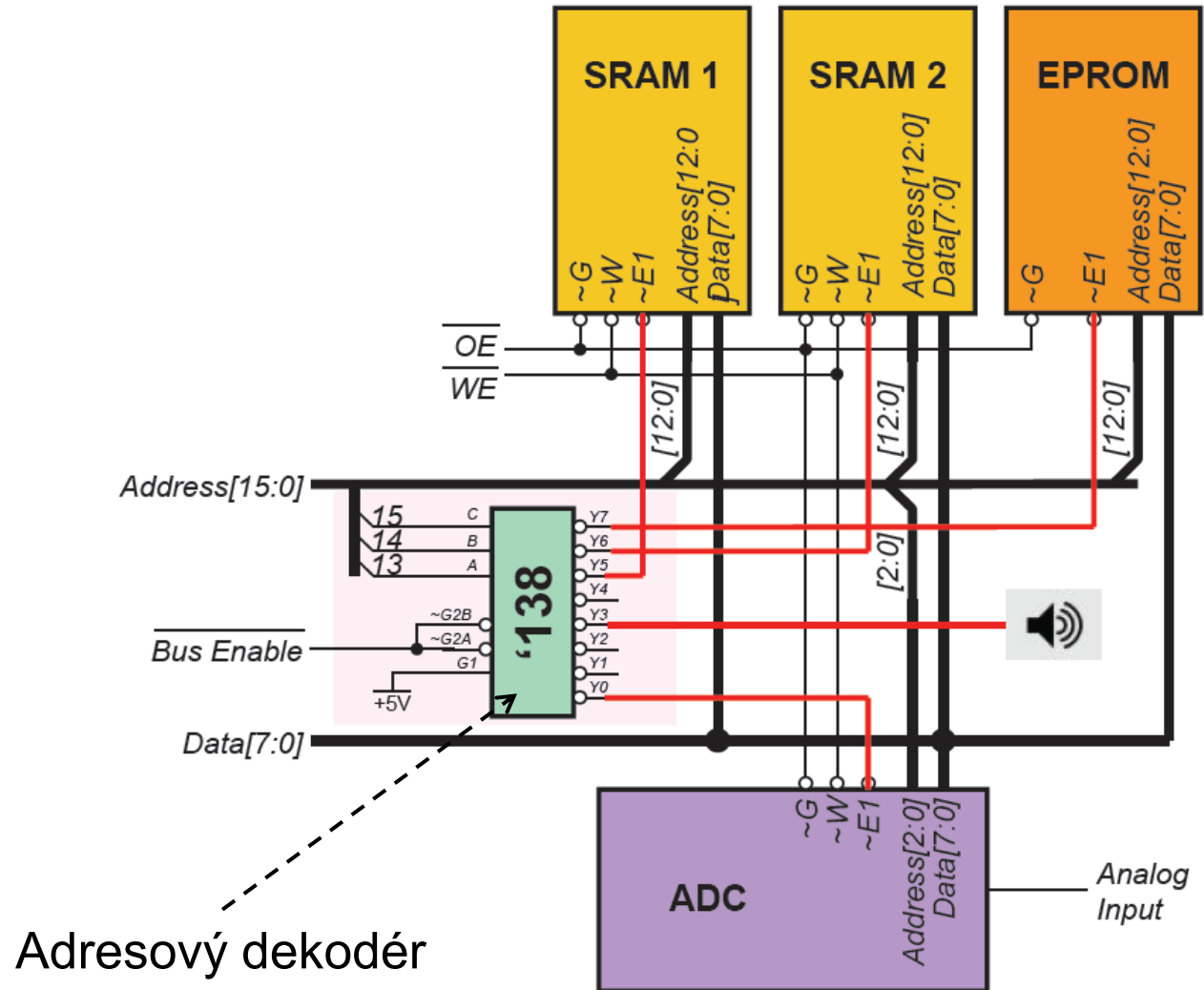
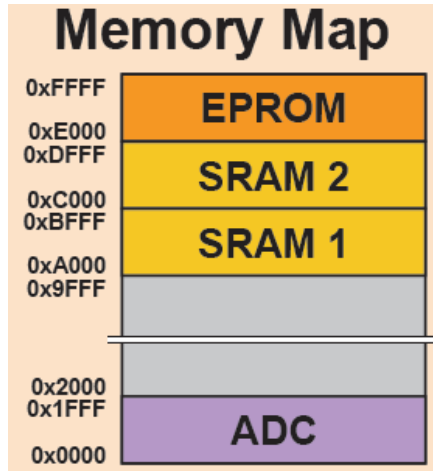
# Čtení asynchronní SRAM



# Zápis do SRAM

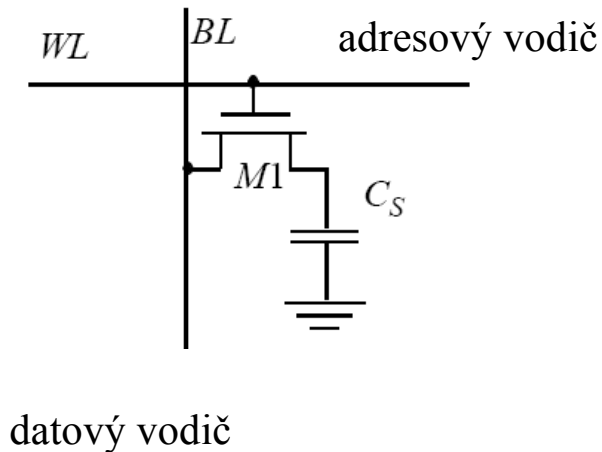


# Adresování dvou a více pamětí



# Dynamická paměť (DRAM) – od 70. let

Informace je uložena **pomocí elektrického náboje na kondenzátoru**.

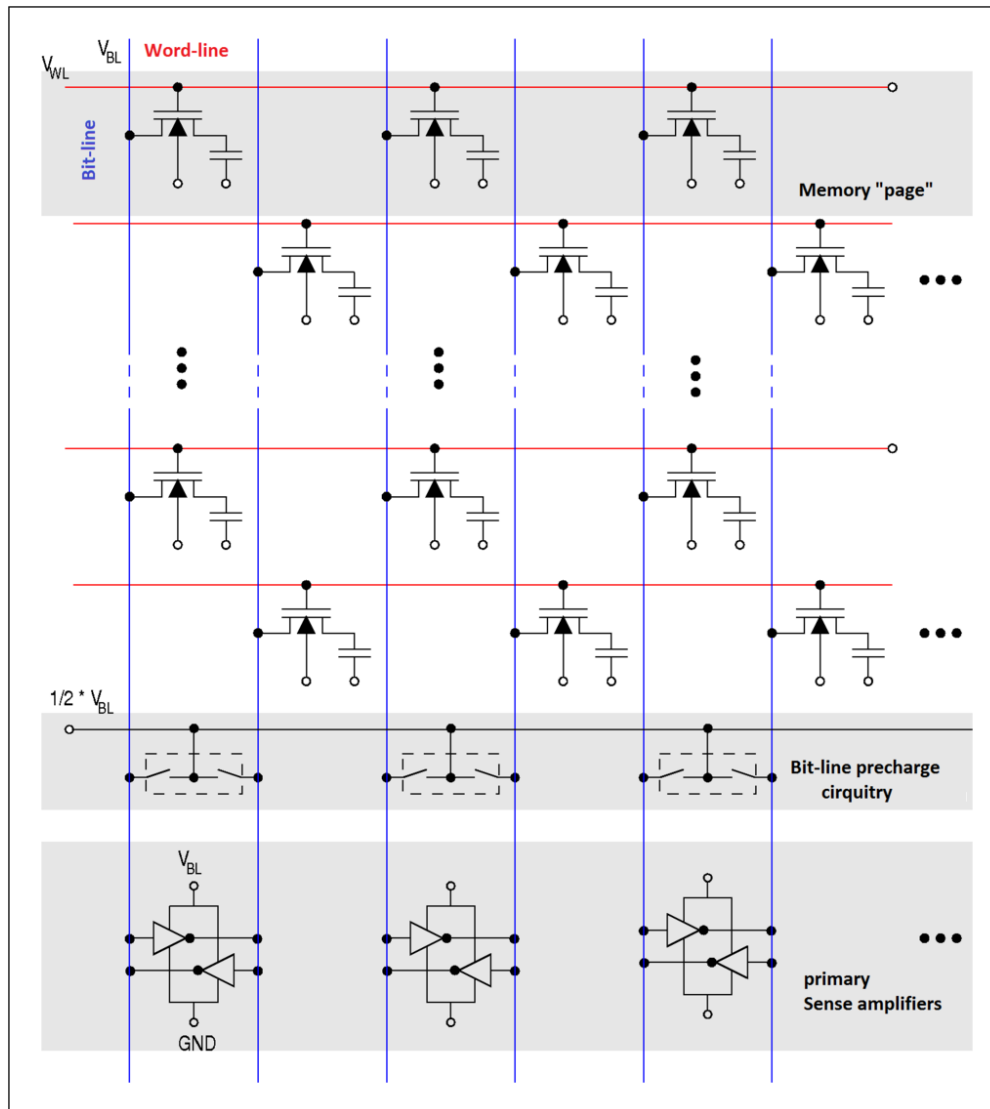


Při **zápisu** se na adresový vodič (WL – word line) přivede hodnota logická 1 => tranzistor se otevře. Na datovém vodiči (BL – bit line) je umístěna zapisovaná hodnota (např. 1), tato hodnota projde přes otevřený tranzistor a nabije kondenzátor. V případě zápisu nuly dojde pouze k případnému vybití kondenzátoru (pokud byla dříve v paměti uložena hodnota 1).

Při **čtení** se využívá obvodu nazývaného **diferenciální zesilovač**, který umí porovnat napětí „přečtené“ z kondenzátoru s referenčním napětím (typicky  $U_{cc}/2$ ), výsledek interpretovat jako log. 0 či log. 1 a dále tuto hodnotu uložit zpět (tzv. refresh). Podrobněji – viz další slide.



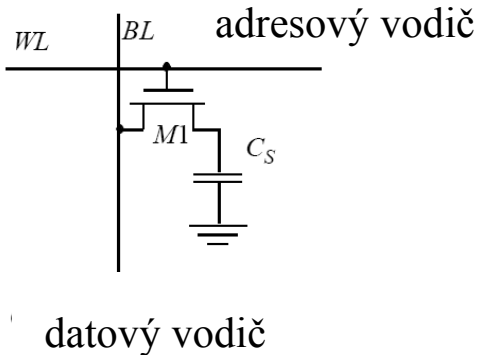
# Dynamická paměť – čtení



## Operace čtení:

1. Na vodiče BL je na okamžik přivedeno  $U_{cc}/2$  (tzv. precharging). Vodiče BL mají dostatečnou kapacitu na to, aby toto napětí na potřebnou dobu udržely.
2. Vybraný řádek ( $WL = 1$ ) je aktivován, tranzistory na vybraném řádku jsou nyní otevřeny.
3. Kondenzátory jsou tak připojeny na BL, což způsobí malý nárůst (pokud je kondenzátor nabitý) nebo malý pokles (pokud je na kondenzátoru 0V) napětí na příslušných BL.
4. Čtecí zesilovač detekuje tuto změnu na příslušném BL, porovná ji s  $U_{cc}/2$  a vyhodnotí ji jako 0 nebo 1 uloženou v buňce. Současně dojde k obnovení (refresh) napětí na kondenzátorech vybraného řádku.
5. WL je deaktivován, čtecí zesilovače jsou odpojeny.

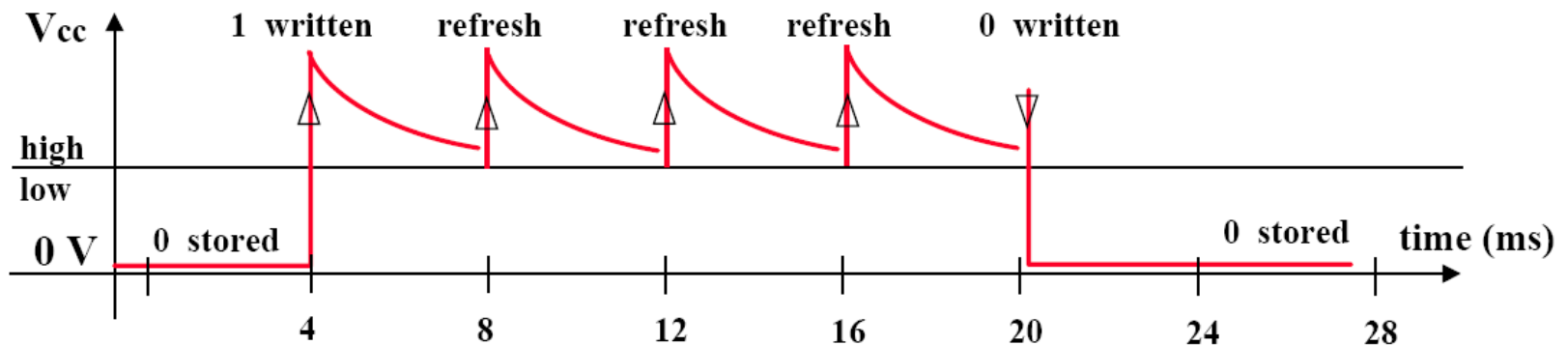
# DRAM – obnova informace



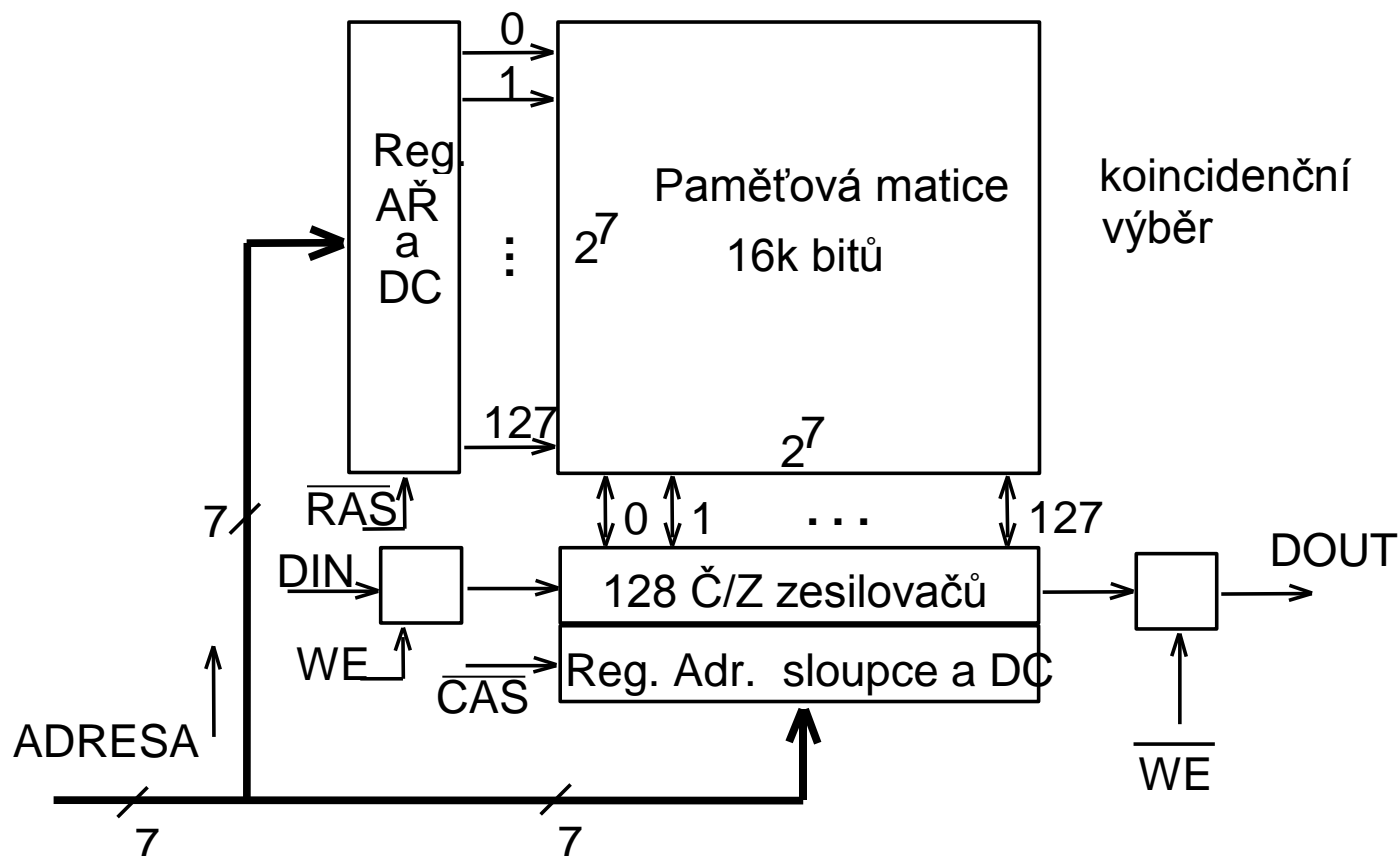
Náboj na kondenzátoru se **vybíjí** i v době, kdy je paměť připojena ke zdroji elektrického napájení => je nutné periodicky provádět tzv. **refresh**, tj. obnovení paměťové buňky. Tuto funkci plní některý z obvodů čipové sady.

Jednou za dobu  $t_r$  ( $t_r$  je např. 4 ms) dojde k obnově informace v paměti. Zdegenerovaný obsah řádku je načten do registru a vzápětí je obsah registru (již s upravenými log. úrovněmi) zapsán zpět do buněk paměti.

Obrázek ukazuje průběh napětí na kondenzátoru po zápisu log.1.

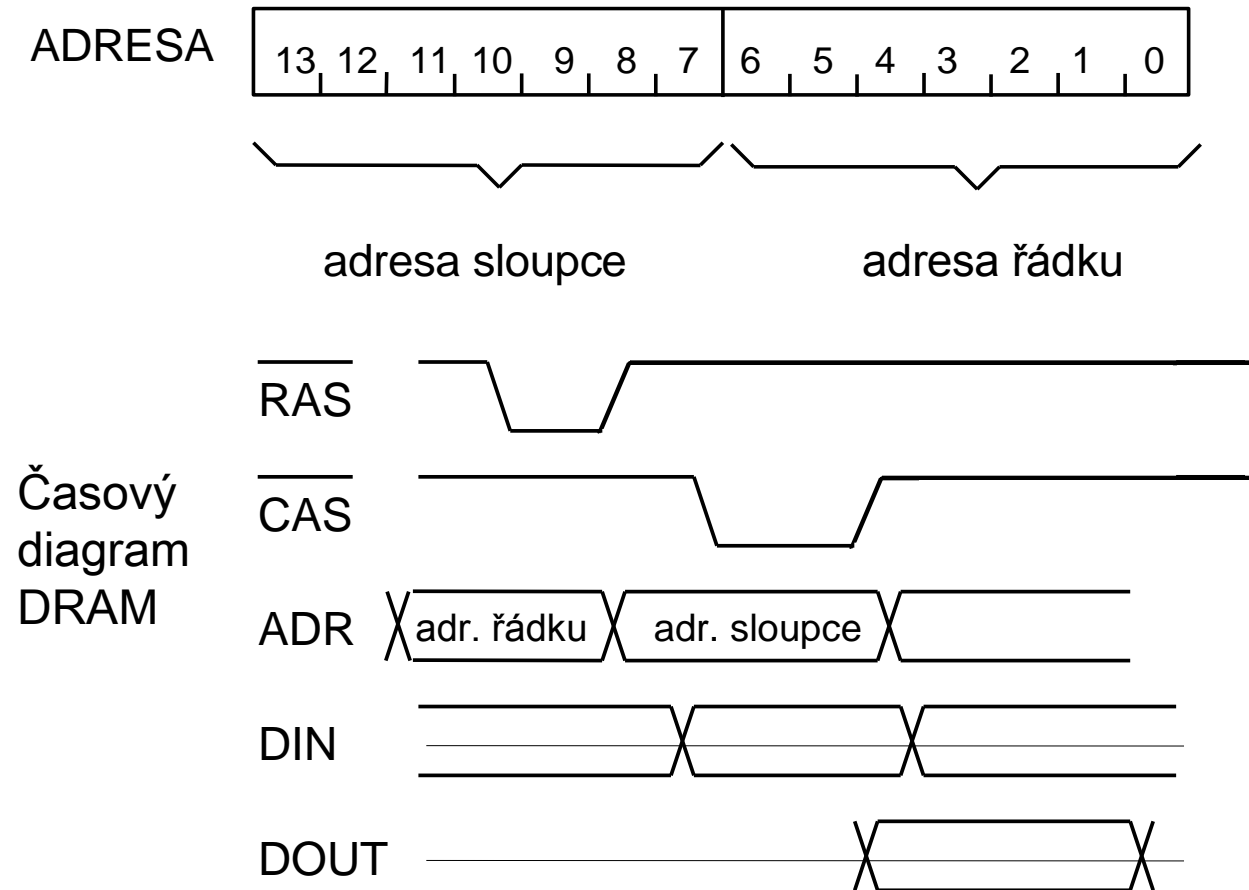


# Struktura DRAM 16K x 1bit



# Předávání adresy DRAM

Pro minimalizaci počtu vývodů pouzder DRAM se ustálilo předávání adresy *multiplexním způsobem* nadvakrát:



# Komentář k adresování DRAM

Nejdříve se předává **adresa řádku** AŘ; zápis této části adresy do registru AŘ z adresové sběrnice ADR je řízen signálem **RAS** - *Row Address Select*.

Poté se na adresové sběrnici ADR objeví **adresa sloupce** AS a ta se zapíše do registru AS signálem **CAS** - *Column Address Select*. Jde-li o zápis, musí být na datovém vstupu DIN ve stanoveném intervalu platný datový bit, který se zapíše do adresovaného místa. Jde-li o čtení, pak za dobu danou katalogovými hodnotami se na datovém výstupu DOUT objeví přečtený bit. Funkce čtení/zápis se řídí signálem WE (často se používá označení R/-W).

**Adresový výběr** provádějí dva adresové dekodéry s výstupem v kódu 1 z 128, a *koincidence* (tedy logický součin) *aktivovaného řádkového a sloupcového vodiče* určí adresované paměťové místo (1 z 16 384). *Koincidenčním výběrem* se výběr paměťového místa z paměťové matice výrazně zjednoduší.

**Čtecí/zapisovací** (Č/Z) zesilovače slouží jak pro vyčtení bitu z paměťové buňky, tak i pro obnovu hodnoty uložené v paměťové buňce.

# Řízení obnovy dat

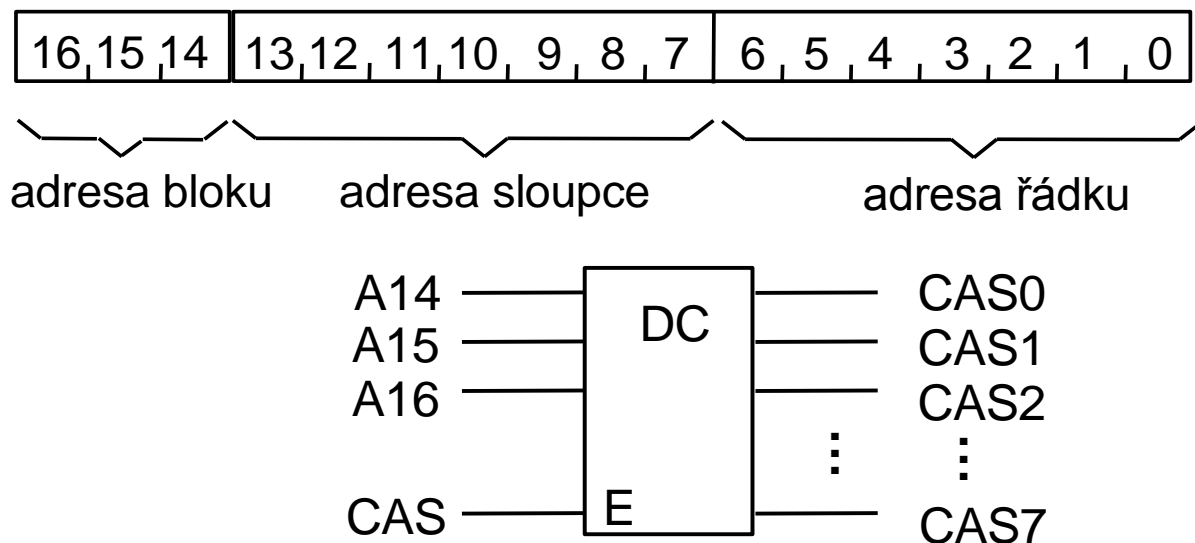
Mechanismus obnovy musí zajistit, aby se před uplynutím zaručené *doby uchování informace* adresovala všechna paměťová místa. Základem *řadiče obnovy* je čítač adresy řádku, který inkrementuje po 1 a po naplnění čítá znovu od nuly. Pro pouzdro DRAM 16K x 1 je čítač obnovy sedmibitový.

Obnova dat se zajišťuje většinou jako vnější obnova, speciální paměti mají vnitřní mechanismus obnovy, tedy obvody pro řízení obnovy mají přímo na čipu. Vnější obnova je uspořádána jako

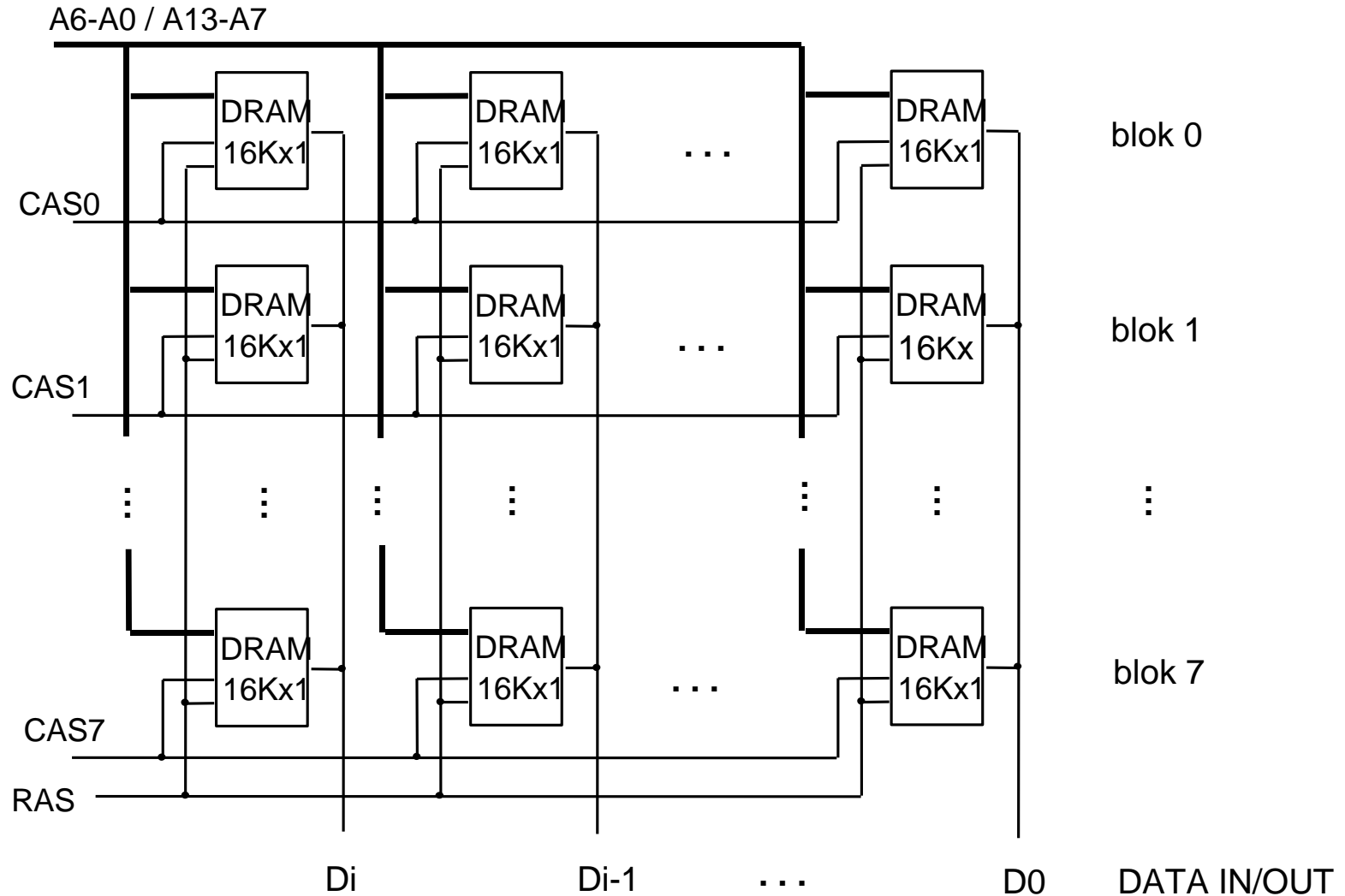
- rozložená
- dávková
- transparentní
- během normální činnosti

# Bloková struktura DRAM (1)

Zvětšení kapacity paměti (adresového prostoru) se provádí uspořádáním paměti do **bloků**. Na obrázku je naznačena *rozšířená adresa* a *dekodér adresy bloku*. Zde je použito jedno z více možných uspořádání. Protože u DRAM nejsou k dispozici výběrové signály *Chip Select*, jsou pomocí dekodéru vytvářeny v kódu 1 z n *selektivní signály*  $CAS_i$ . Signál RAS je rozveden do všech bloků. Výsledná bloková struktura paměti je na dalším obrázku. Zvětšování *šířky datového slova* je omezeno pouze výkonem výstupů zdrojů signálů  $CAS_i$  a RAS.

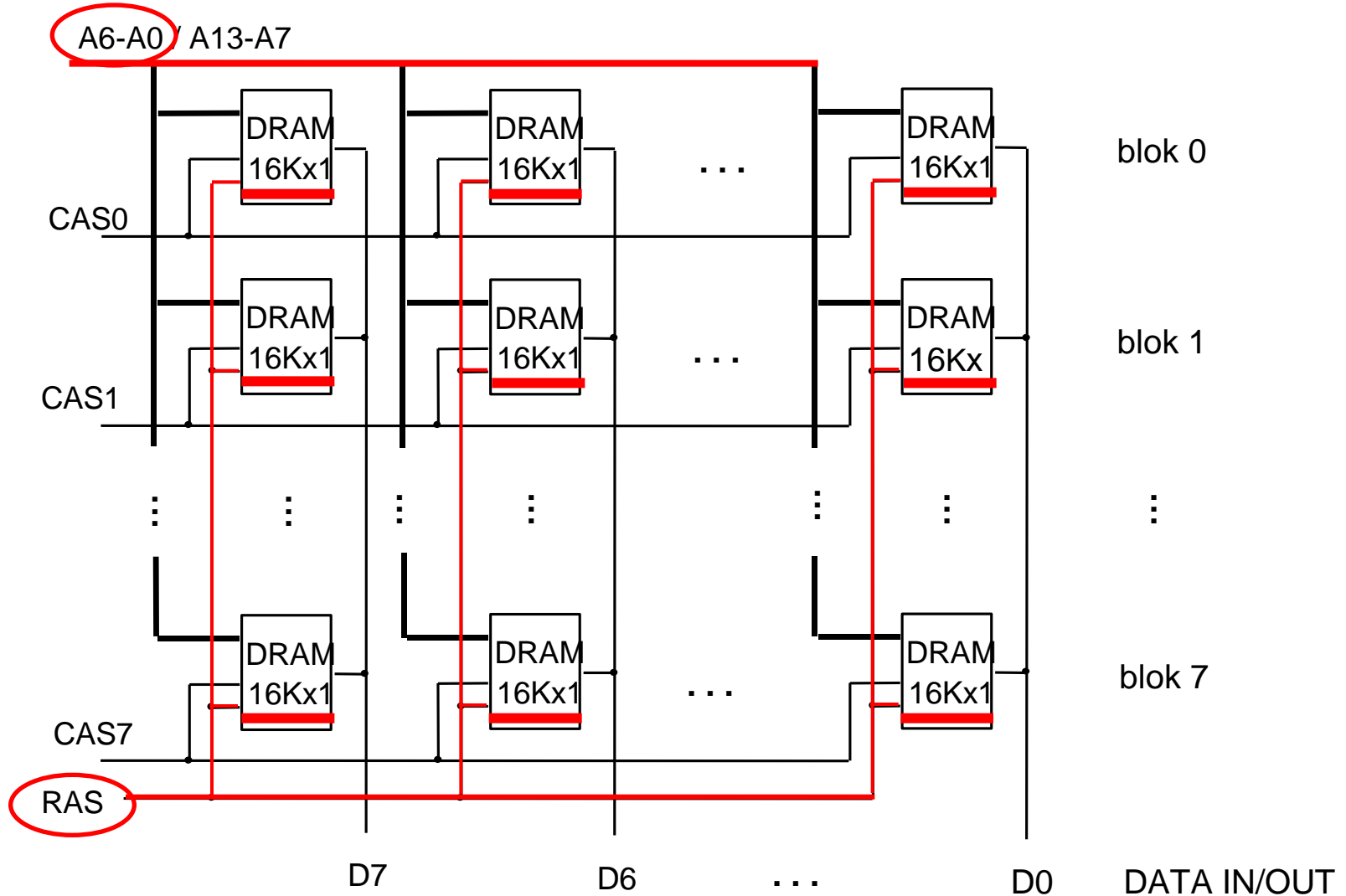


## Bloková struktura DRAM (2)



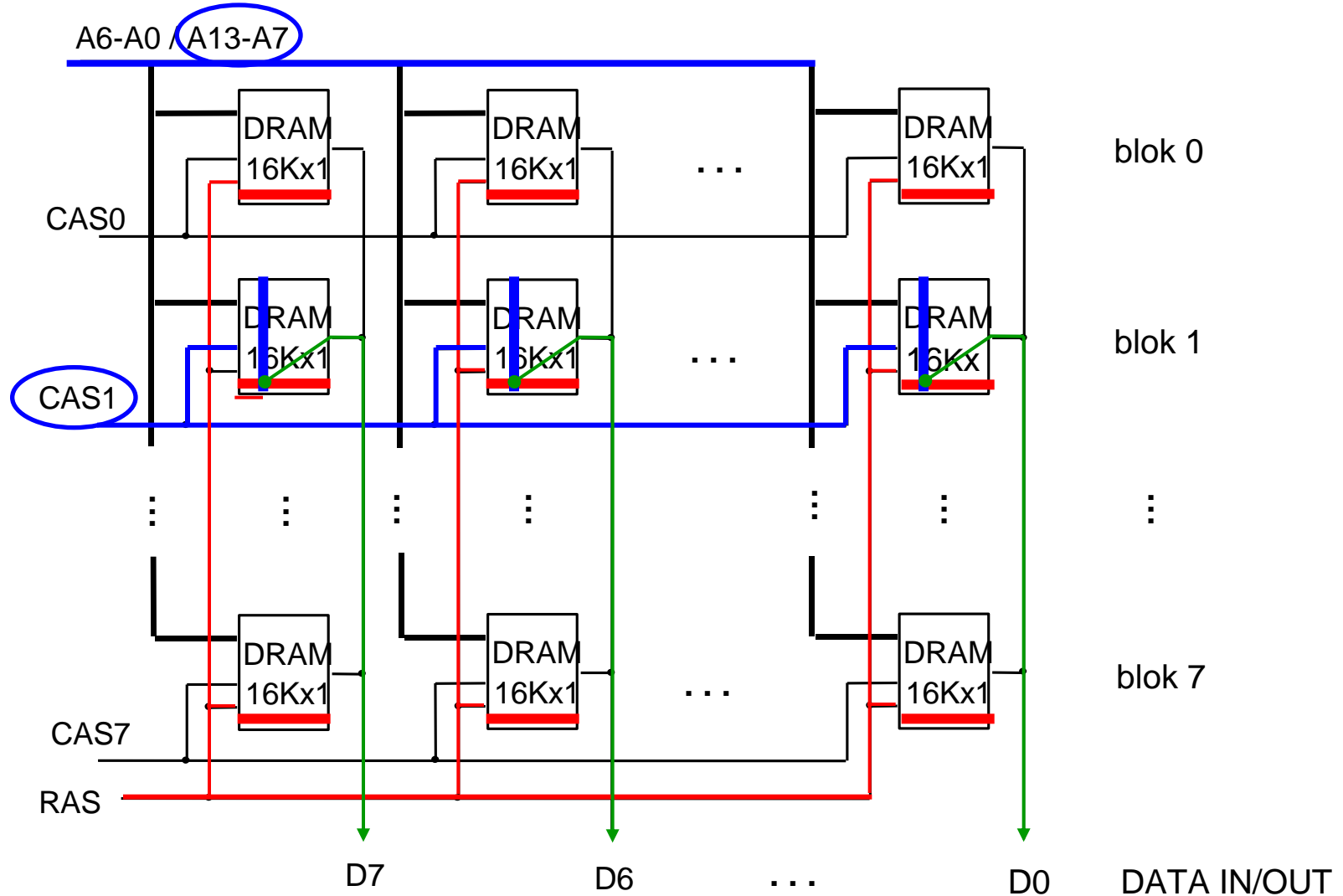


# Bloková struktura DRAM 16k x 8 bitů: Čtení - RAS



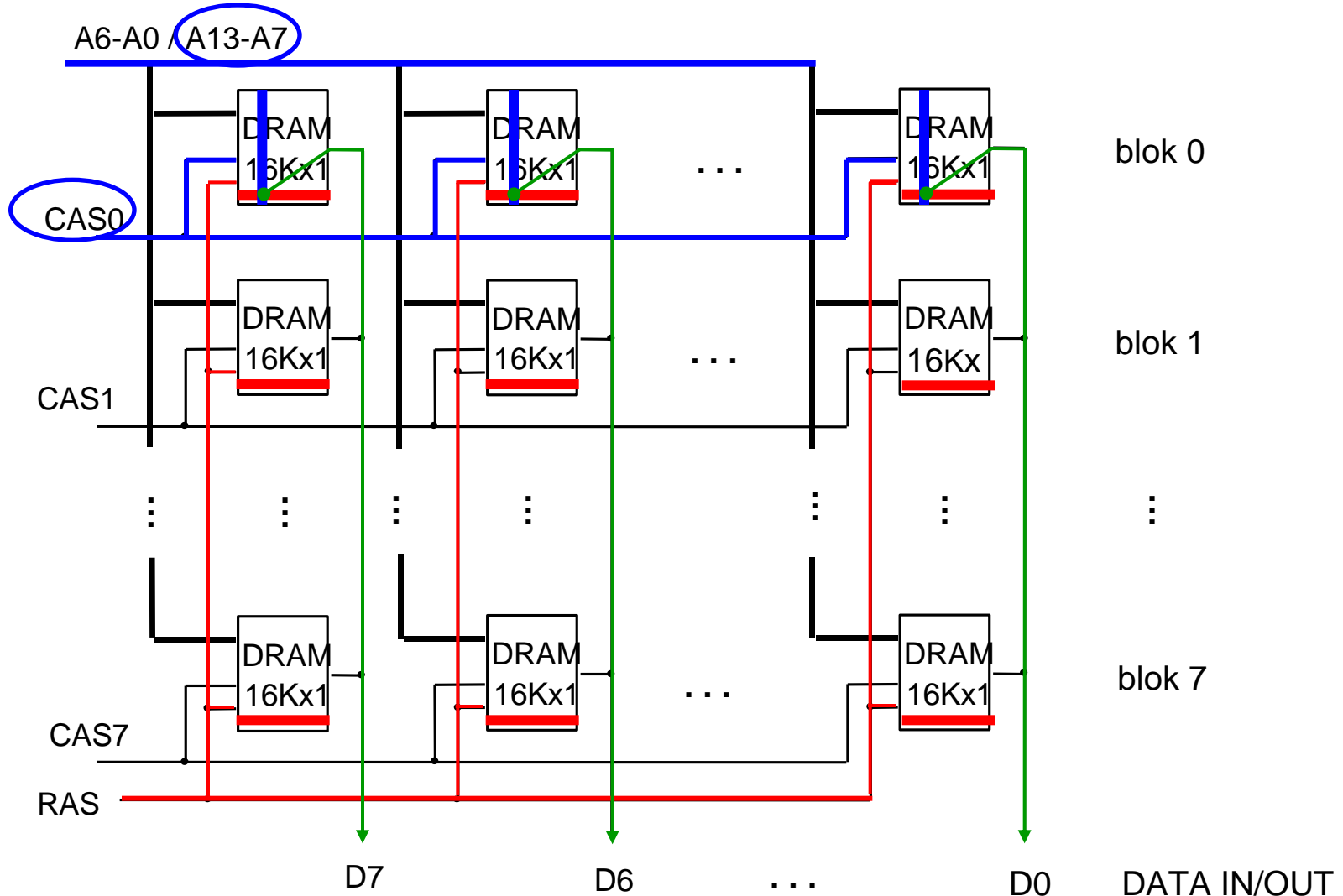
1) Adresování řádku.

# Bloková struktura DRAM 16k x 8 bitů: Čtení - CAS



2) Adresování sloupců v bloku 1. Vyčtení dat (jednoho bajtu) z bloku 1.

# Bloková struktura DRAM 16k x 8 bitů: Další čtení



Další čtení lze provést pouhou změnou adresy bloku, což je mnohem rychlejší než adresovat jiný řádek a sloupec. **Nejrychlejší možné vyčtení 8 bajtů z paměti nastane, pokud tyto bajty budou uloženy v blocích 0, 1, ..., 7 a všechny bity budou mít stejnou adresu řádku a sloupce.**

# Př. Topologie čipu DRAM 16 M x 1 bit


23	20	19	10	9	0
4 bity		10 bitů			10 bitů

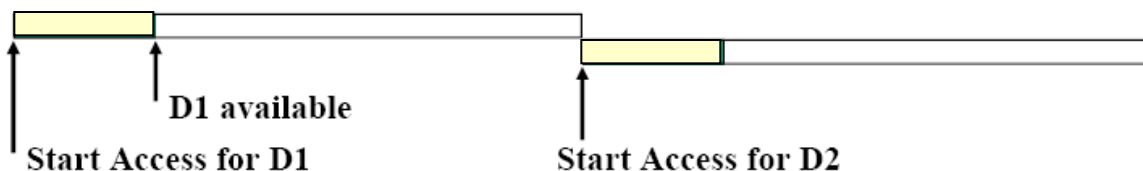
adresa bloku    adresa sloupce    adresa řádku

## Blokový režim paměti:

Jeden blok má kapacitu 1 Mbit, třetí rozměr adresy má 4 bity. **Přepnutí bloku** změnou nejvyšších čtyř bitů (bity 20 až 23) beze změny adresy řádku a sloupce **proběhne rychleji, než výběr podle nové adresy**. Toho se využívá u většiny pamětí zavedením tzv. **blokového** nebo též **stránkového režimu**. Po běžné sekvenci nastavení adresy pomocí signálů RAS a CAS se impulsním průběhem na CAS adresují postupně další bloky (hodnota RAS zůstává nezměněna).

# Princip prokládání paměťových operací

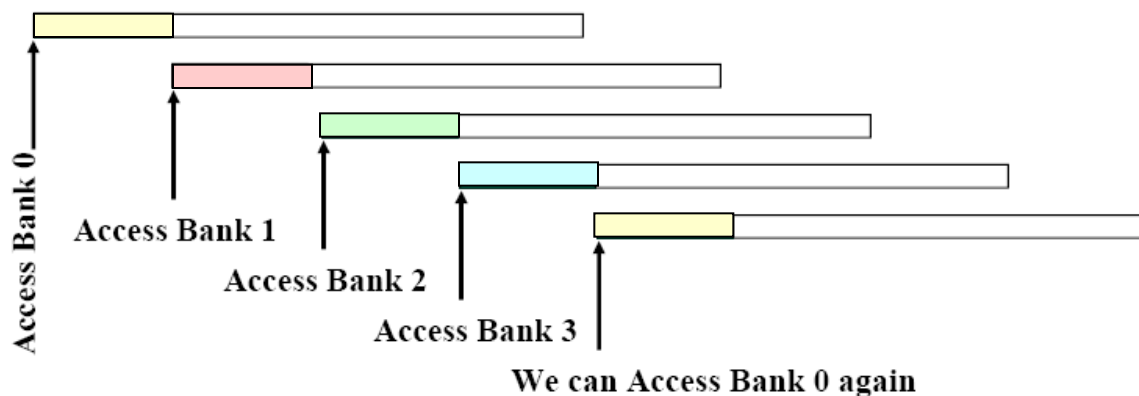
Bez prokládání



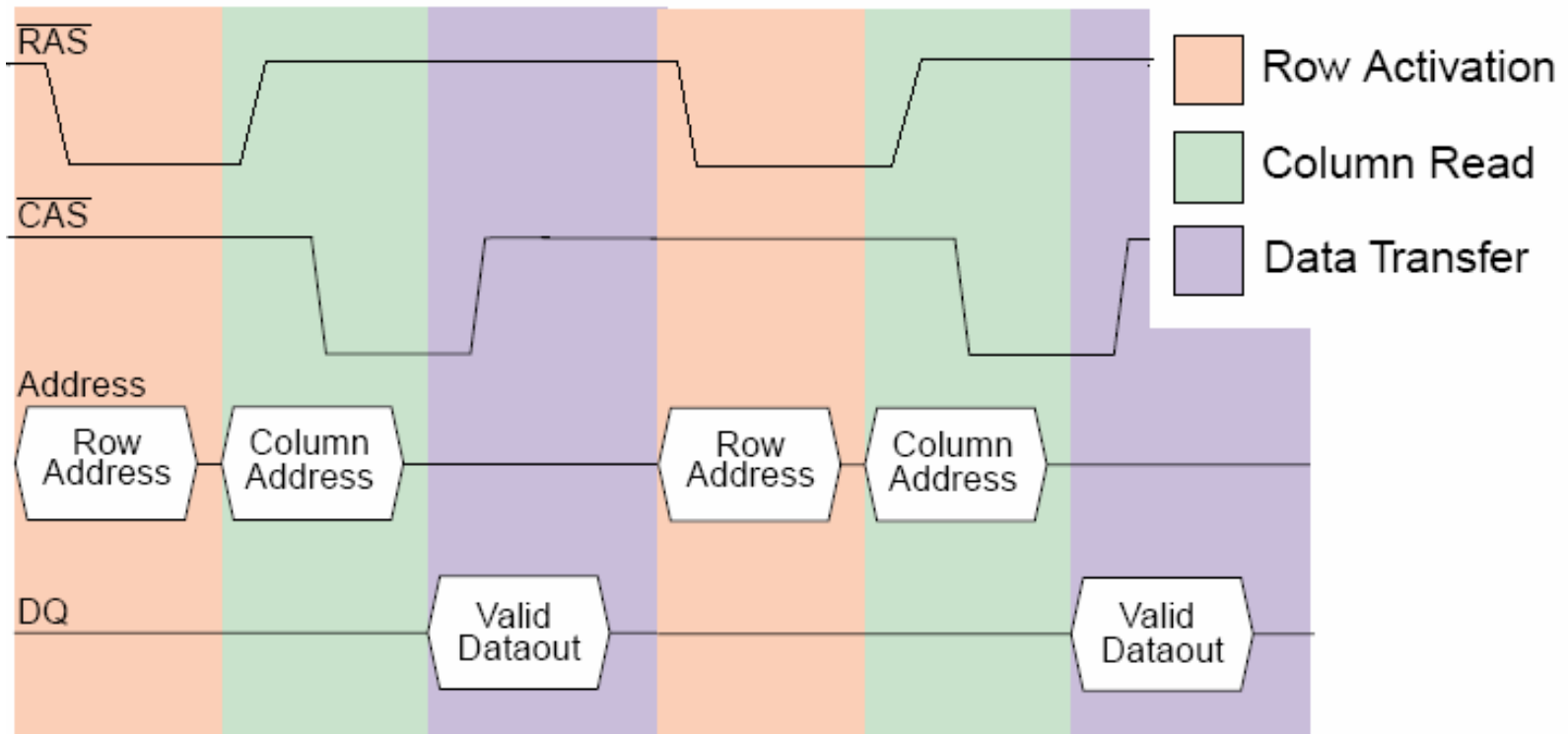
Bank Bez prokládání S prokládáním

0	1	1
	2	5
	3	9
	4	13
1	5	2
	6	6
	7	10
	8	14
2	9	3
	10	7
	11	11
	12	15
3	13	4
	14	8
	15	12
	16	16

Čtyřcestné prokládání – urychlení 4x



# Časování základní varianty DRAM



Neustálé střídání  $\overline{RAS}$  a  $\overline{CAS}$  vede k tomu, že jen v určitých okamžicích (fialová barva na obrázku) dochází k přenosu dat do procesoru. **Jak zvýšit propustnost? Změnit časování paměti!**

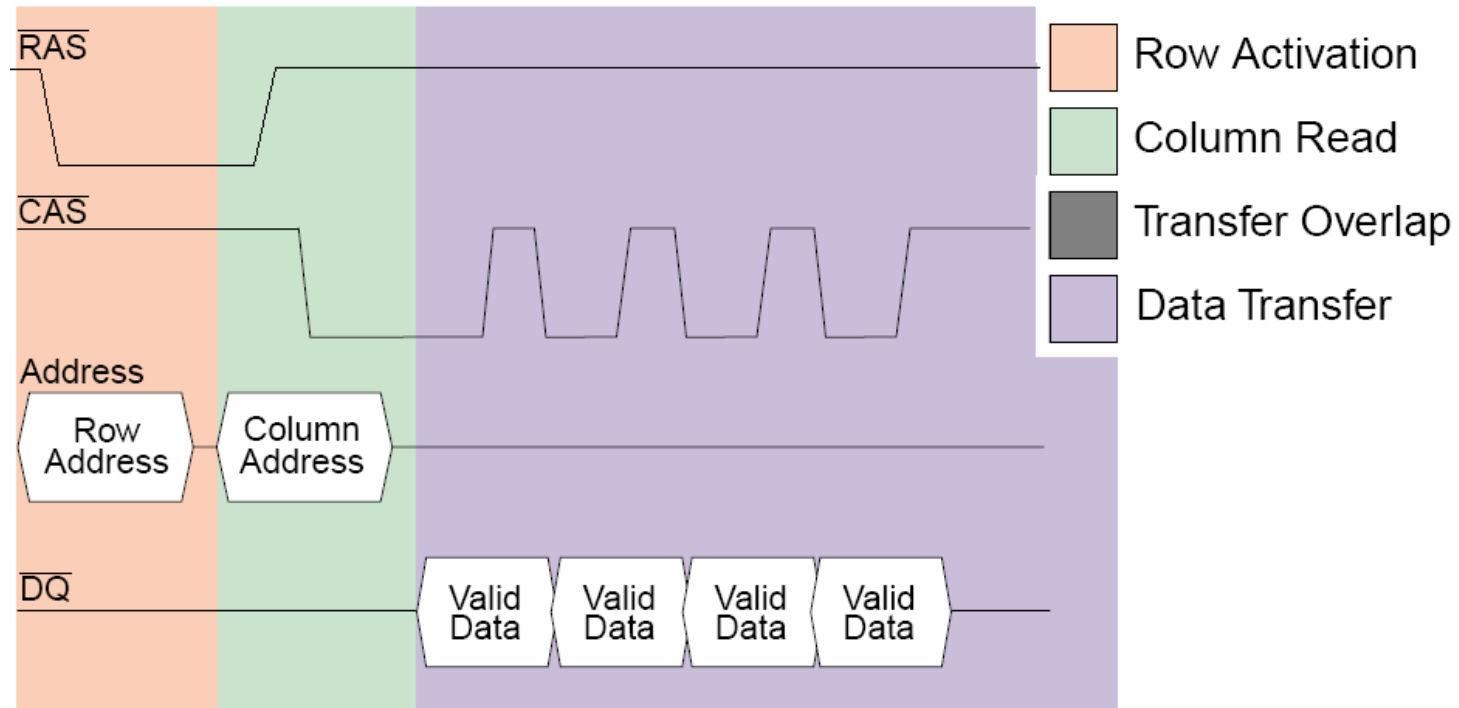
# Vývoj pamětí DRAM

(princip uložení dat zůstává, paměti se liší v zásadě pouze frekvencí a časováním)

- Fast page mode DRAM (**FPM DRAM**)
  - umožňuje realizovat čtení tak, že je nastaven ROW na určitou hodnotu a přičítá se hodnota COL
- Extended data out DRAM (**EDO DRAM**)
  - přidán latch - prodlužuje se doba, po níž jsou přečtená data k dispozici na datové sběrnici => řadič paměti má více času na to, aby předal data přes sběrnici do procesoru.
- Burst Enhanced Data-Out DRAM (**BEDO DRAM**)
  - interní čítač adres urychlí operace s pamětí (4 adresy v dávce)
- Synchronous DRAM (**SDRAM**)
  - jiný princip oproti předchozím, synchronizace paměti, ovládání příkazy
- Rambus™ DRAM (**RDRAM**)
  - speciální typ paměti – seriové propojení, sběrnice pouze 16 bitů
- Double data-rate synchronous DRAM (**DDR SDRAM**)
  - přenosové děje se ovíjejí od nástupné i sestupné hrany synchronizačních pulsů – je tak možné dvakrát zrychlit přenos
- DDR2 SDRAM, DDR3, DDR4

# BEDO – Burst Enhanced Data-Out DRAM

Paměti typu BEDO obsahují **interní čítač** adres, takže se do adresového registru zavádí pouze adresa první, zbývající čtyři se odvodí v čítači postupnou inkrementací => redukce objemu komunikace mezi řadičem paměti a pamětí (**burst = dávkové zpracování**). BEDO urychluje o 30% oproti EDO.

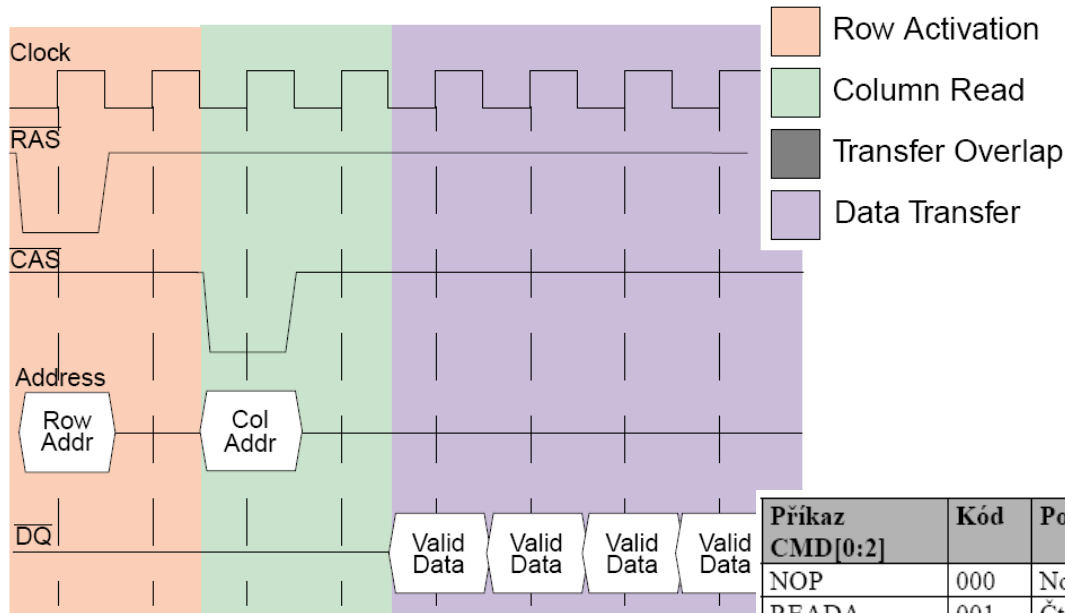


Zrychlení získáno za cenu, že data nejsou čtena z libovolných adres.



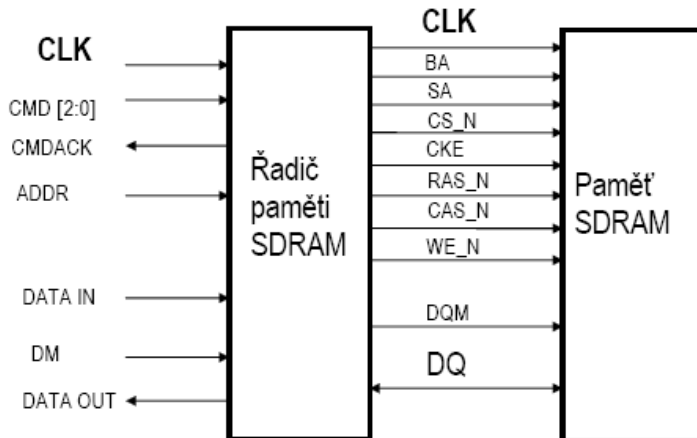
# Synchronní DRAM - SDRAM

(1993, 66 – 133 MHz, až 512MB, 3.3V )



Synchronní DRAM – operace jsou synchronizovány s ext. sběrnici  
Dávkový (burst) režim, podobně jako BEDO.

Paměť SDRAM je řízena kombinací signálů RAS, CAS a WE, které představují kód příkazu. Každou nástupnou hranou signálu CLK jsou tyto signály dekodovány a SDRAM pak provede požadovanou funkci.



Příkaz CMD[0:2]	Kód	Popis činnosti
NOP	000	No operation.
READA	001	Čti SDRAM + auto precharge
WRITEA	010	Piš do SDRAM + auto precharge
REFRESH	011	refreš
PRECHARGE	100	Uzavři všechny banky
LOAD MODE	101	Zapiš do registru mode
LOAD_REG1	110	Zapiš do konfiguračního registru R1
LOAD_REG2	111	Zapiš do konfiguračního registru R2

Příkazy přijímané řadičem paměti a jejich kódy

Příkaz	Zkratka	RASN	CASN	WEN
No operation	NOP	H	H	H
Activate	ACT	L	H	H
Read	RD	H	L	H
Write	WR	H	L	L
Burst terminate	BT	H	H	L
Precharge	PCH	L	H	L
Autorefresh	ARF	L	L	H
Load mode register	LMR	L	L	L

Příkazy generované řadičem pro SDRAM

## Burst režim (dávkový režim)

- Umožňuje čtení nebo zápis **bloku dat** o předem známé délce.
- V bloku jsou obsažena data z „nějak sousedních“ adres, ne z libovolných adres.
- Díky synchronnímu provozu stačí pouze inicializovat přenos (nastavit adresu řádku i adresu sloupce) a poté je v každém taktu provedeno čtení nebo zápis jedné hodnoty.
- Prvotní zpoždění způsobené výběrem řádku představuje tzv. **latenci paměti**.
- Dávkový přenos představuje hlavní přínos synchronních DRAM oproti asynchronním DRAM.
- V dávkovém režimu paměť pracuje na své maximální možné rychlosti. Je žádoucí používat přenosy bloků dat a nikoli čtení či zápis na náhodné adresy.

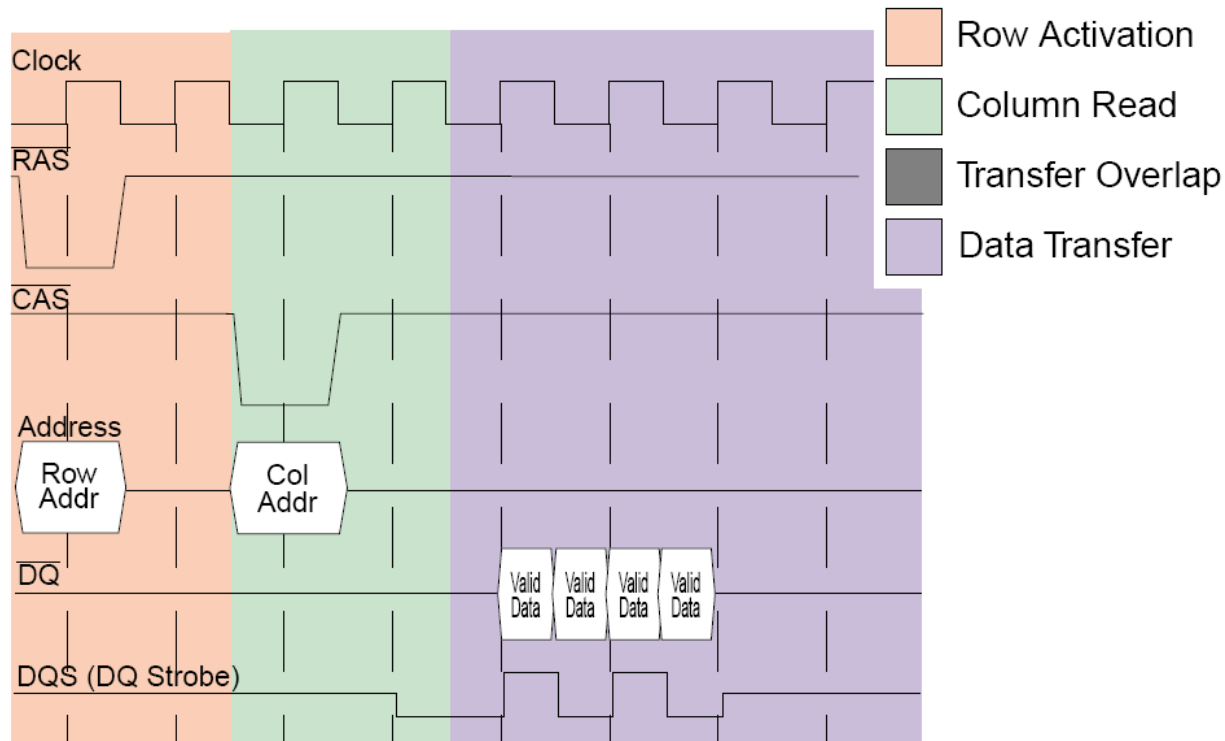
# DDR SDRAM – Double Data Rate

(r. 2000, 200 – 400 MHz, 2,5V, 64MB - 2GB )

DDR - data jsou přenášena na **náběžné i sestupné hraně** hodinového pulsu (propustnost: 1,6 - 3,2 GB/s).

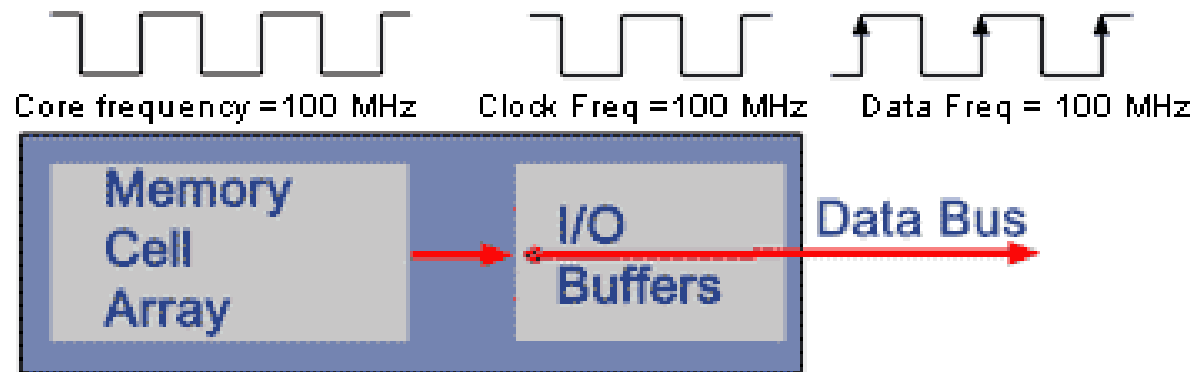
Startovací frekvence jsou od 266 MHz, které se označují standardem DDR266.

Maximální rychlosti technologie DDR se pohybují okolo 600 MHz efektivně, ale obvykle je pro jejich dosažení nutné značně vysoké napájecí napětí (i přes 3 V).

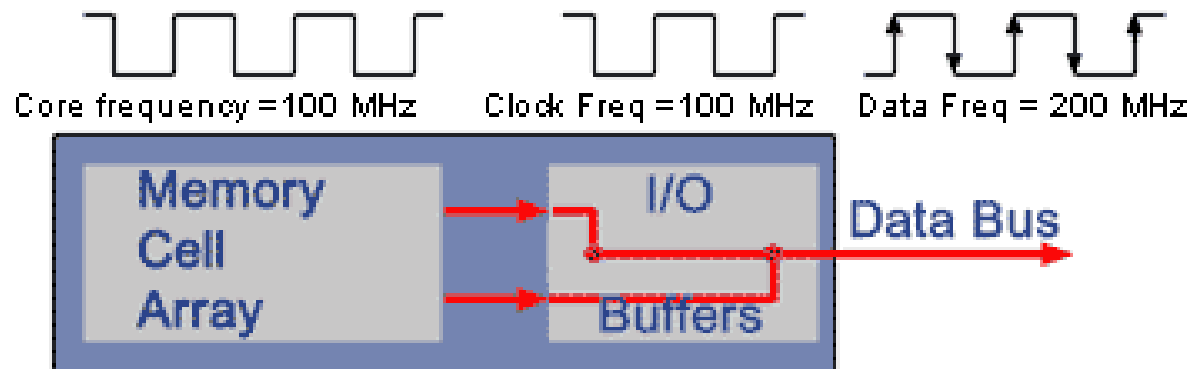


# SDRAM vs DDR

## SDRAM



## DDR I



# DDR2, DDR3, DDR4

## DDR2-400 - DDR2-1066

pracovní frekvence I/O: 200 – 533 MHz, propustnost: 3,2 – 8,5 GB/s, 1,8 V,  
128 MB až 8 GB

## DDR3-800 - DDR3-2133

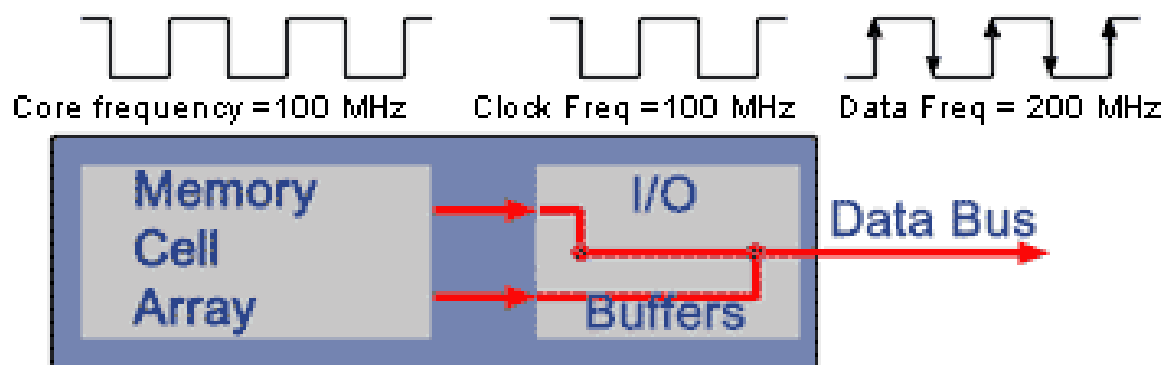
pracovní frekvence I/O: 400 – 1066 MHz, propustnost 6,4 GB/s – 17,066  
GB/s, napájecí napětí 1,35 - 1,5 V  
kapacita: 8 - 16 GB (teoreticky až 128 GB v DIMM)

## DDR4-1600 - DDR4-2400 (2014)

pracovní frekvence I/O: 800 – 1200 MHz, propustnost 12,8 GB/s – 19,2  
GB/s, napájecí napětí 1,05 V  
kapacita: teoreticky až 512 GB v DIMM

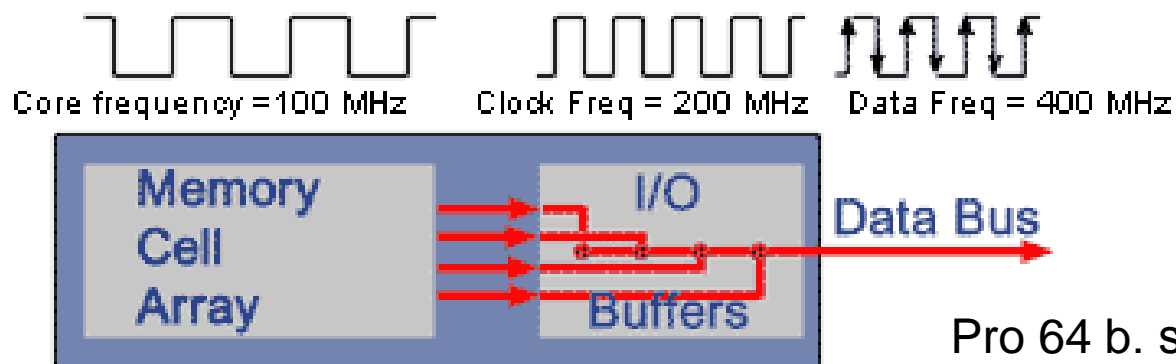
# DDR vs DDR2

## DDR I



prefetch 2

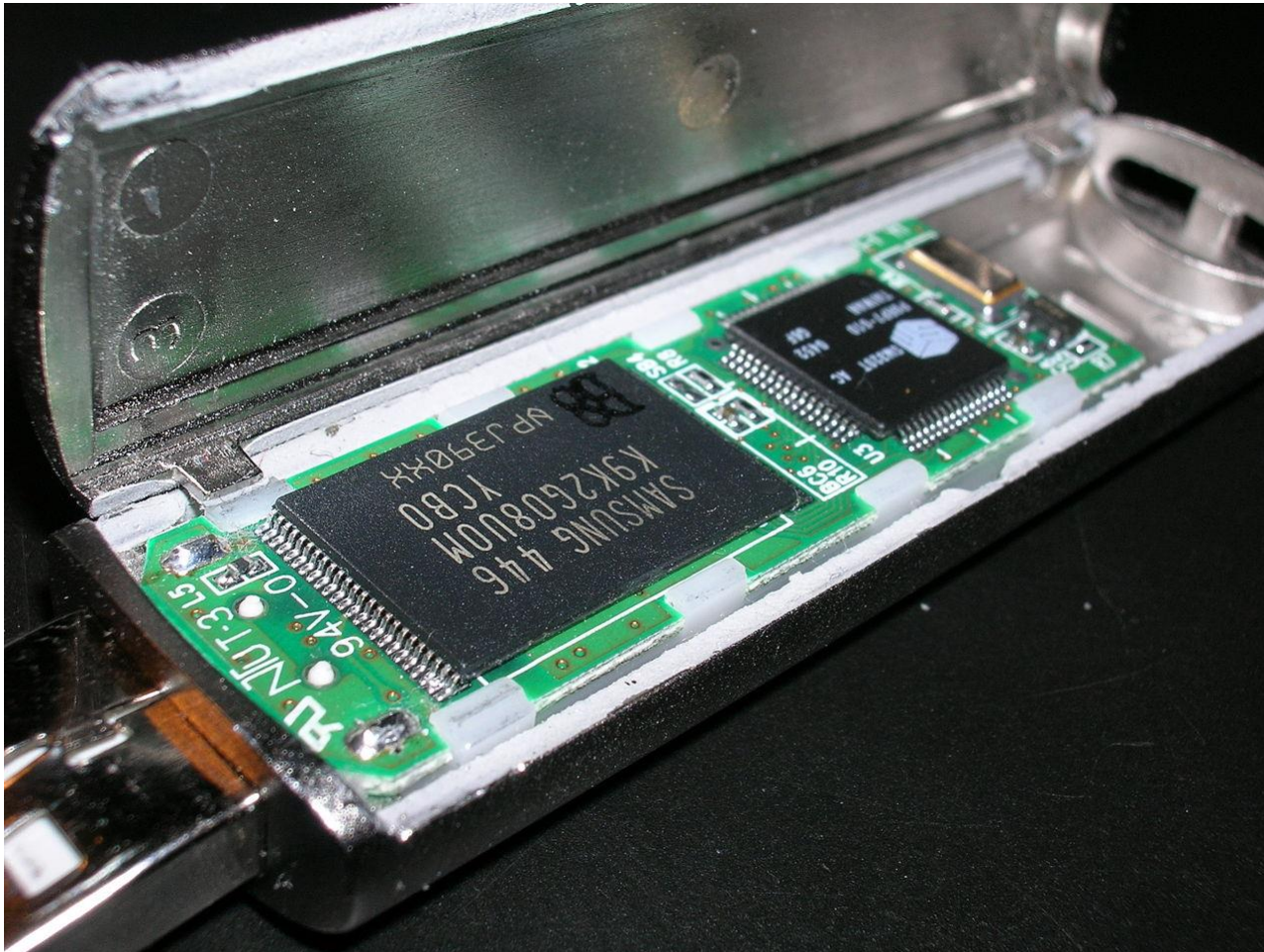
## DDR II



prefetch 4

Pro 64 b. sběrnici je max. propustnost:  
 $100 \text{ MHz} \times 2 \times 2 \times 64/8 = 3,2 \text{ GB/s}$

## Paměti Flash (princip objeven 1980)



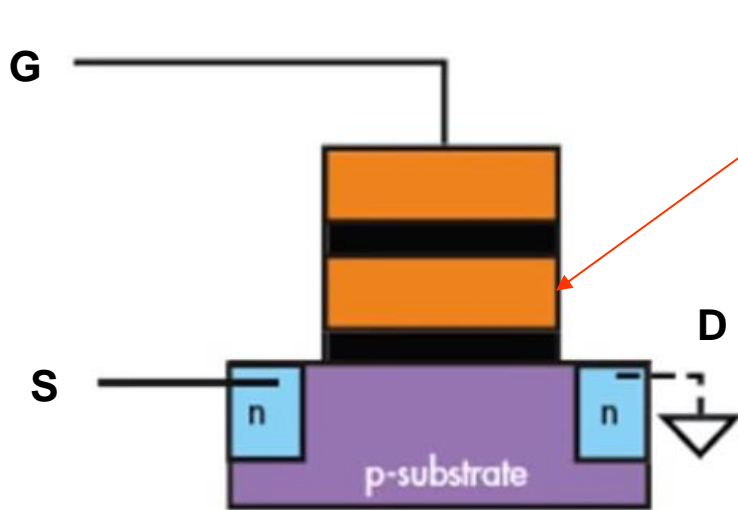
USB klíčenka: Paměťový čip + řídicí čip  
[https://en.wikipedia.org/wiki/Flash\\_memory](https://en.wikipedia.org/wiki/Flash_memory)

# Paměti Flash

- Nevolatilní **semipermanentní** paměť (bez napájení uchová data řádově roky)
- Paměťové buňky jsou tvořeny **tranzistory s plovoucím hradlem** (1 tranzistor na buňku) – levné, vysoká kapacita ~ stovky GB
  - **Plovoucí hradlo uchovává informaci v podobě elektrického náboje**
- Jedna buňka může uchovávat informaci jednoho bitu (Single Level Cell – SLC), případně i více bitů (Multi-Level Cell – MLC)
- Dva základní typy: **NOR Flash** (Intel, 1988), **NAND Flash** (Toshiba, 1989)
  - Liší se hlavně organizací paměťových buněk, způsobem adresování, řízení a rozhraním
- Před zápisem (programováním) je nutno provést mazání (paměti či bloku)



# Tranzistor s plovoucím hradlem - princip



Informace je uchována v podobě přítomnosti či nepřítomnosti záporného náboje na plovoucím hradle.

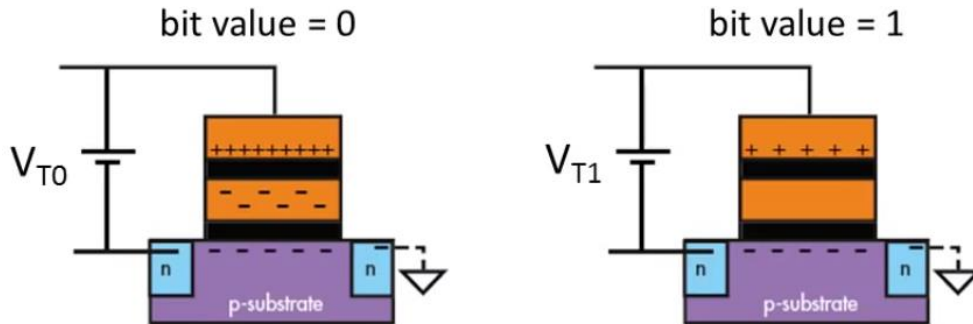
FGFET = Floating Gate FET

Předpokládejme, že FG bez náboje reprezentuje log. 1 („prázdná“ buňka – viz další slide).

Sepnutím tranzistoru ( **$+V_{GS}$  dostatečné velikosti**) jím začne procházet proud a je-li  $V_{SD}$  dostatečně vysoké, některé elektrony začnou prostupovat izolační vrstvou do FG (hot electron inject.). Tranzistor je tak „naprogramován“ - do buňky je „zapsána“ log. 0.

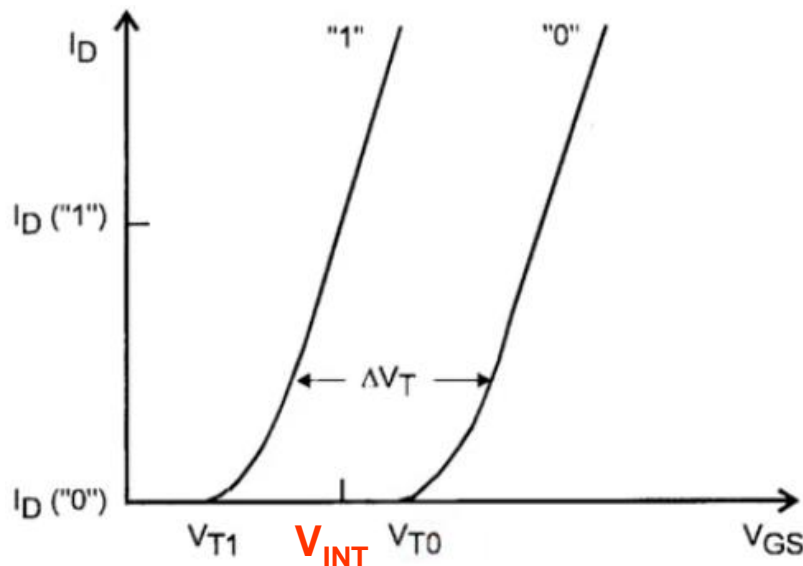
„Vymazání“ buňky (nastavení do log. 1) se napětím opačné polarity  **$-V_{GS}$**  (quantum tunneling).

# Interpretace informace na FGFET



**Přítomnost či nepřítomnost náboje na plovoucím hradle ovlivňuje prahové napětí  $V_T$  potřebné k otevření tranzistoru.**

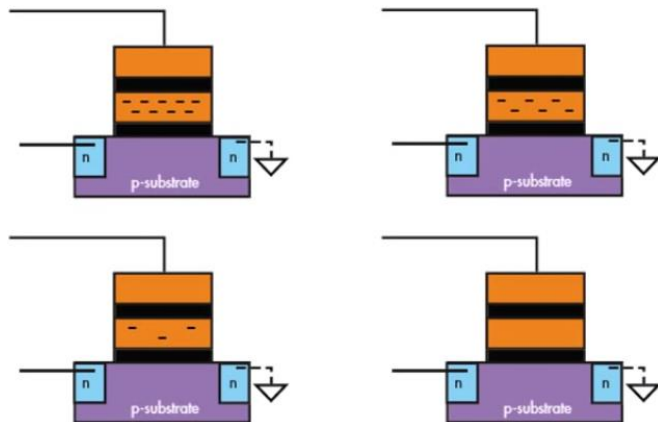
Položíme-li  $V_{GS} = V_{INT}$ , můžeme na základě proudu protékajícího tranzistorem (otevřený/uzavřený tranzistor) rozhodnout o log. hodnotě uloženého bitu (1 nebo 0) – operace **čtení**.



## Single Level Cell (SLC):

Tranzistor bez náboje na plovoucím hradle si „pamatuje“ hodnotu log. 1 (ekvivalent „vymazané“ paměti), záporný náboj je interpretován jako log. 0.

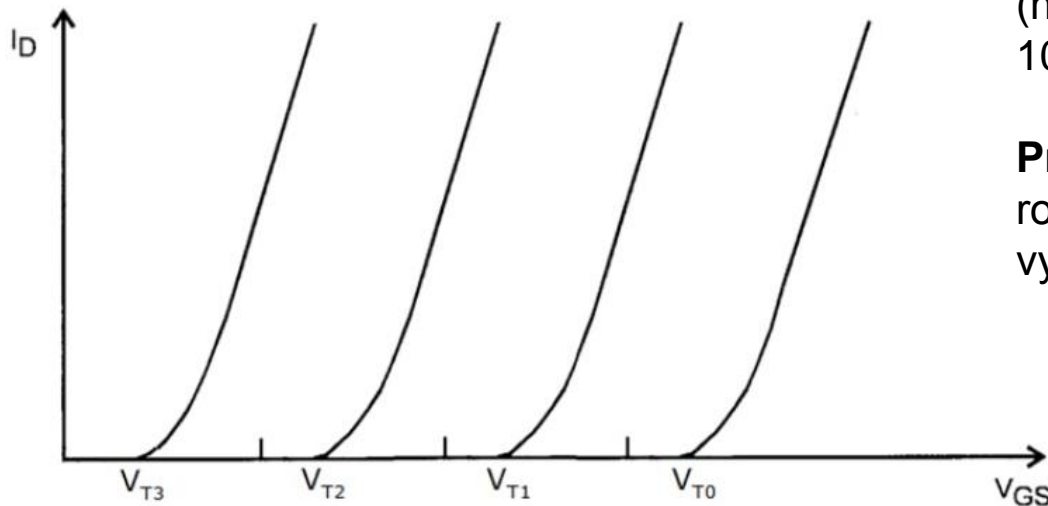
# Koncept Multi-Level Cell



Dříve uvedený princip SLC lze zobecnit, když budeme rozlišovat více hodnot náboje na plovoucím hradle a podle toho i více hodnot proudu protékajícího tranzistorem – koncept **Multi-Level Cell (MLC)**.

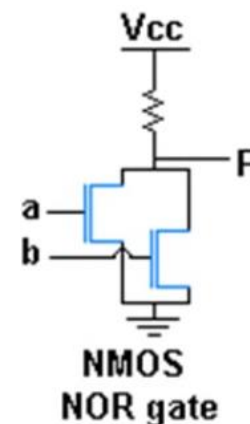
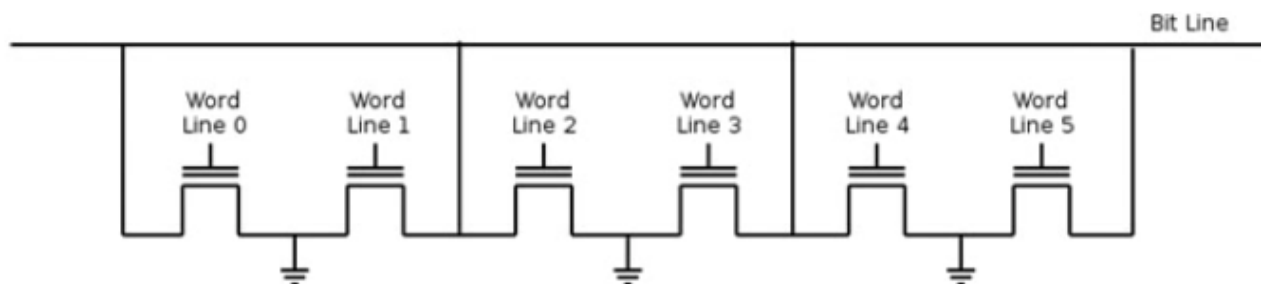
Tím je pak umožněno uložit více bitů pomocí jednoho FGFET (např. 2 bity – kombinace 00, 01, 10, 11).

**Problém:** je třeba precizně rozlišovat více hodnot proudu, vyšší náchylnost k chybám.

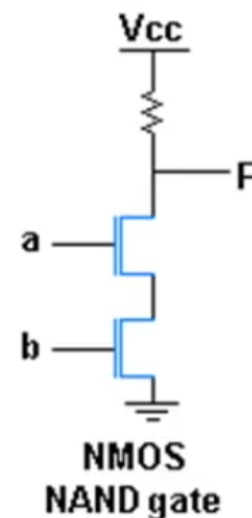
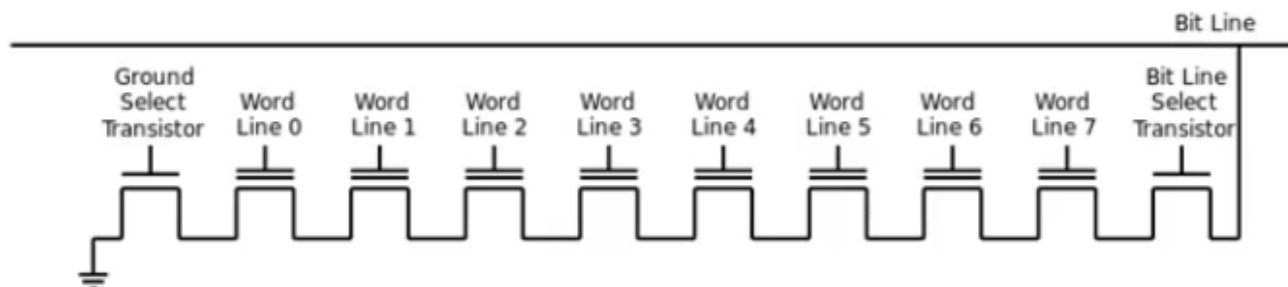


# Typy Flash (podobnost s hradly NOR, NAND)

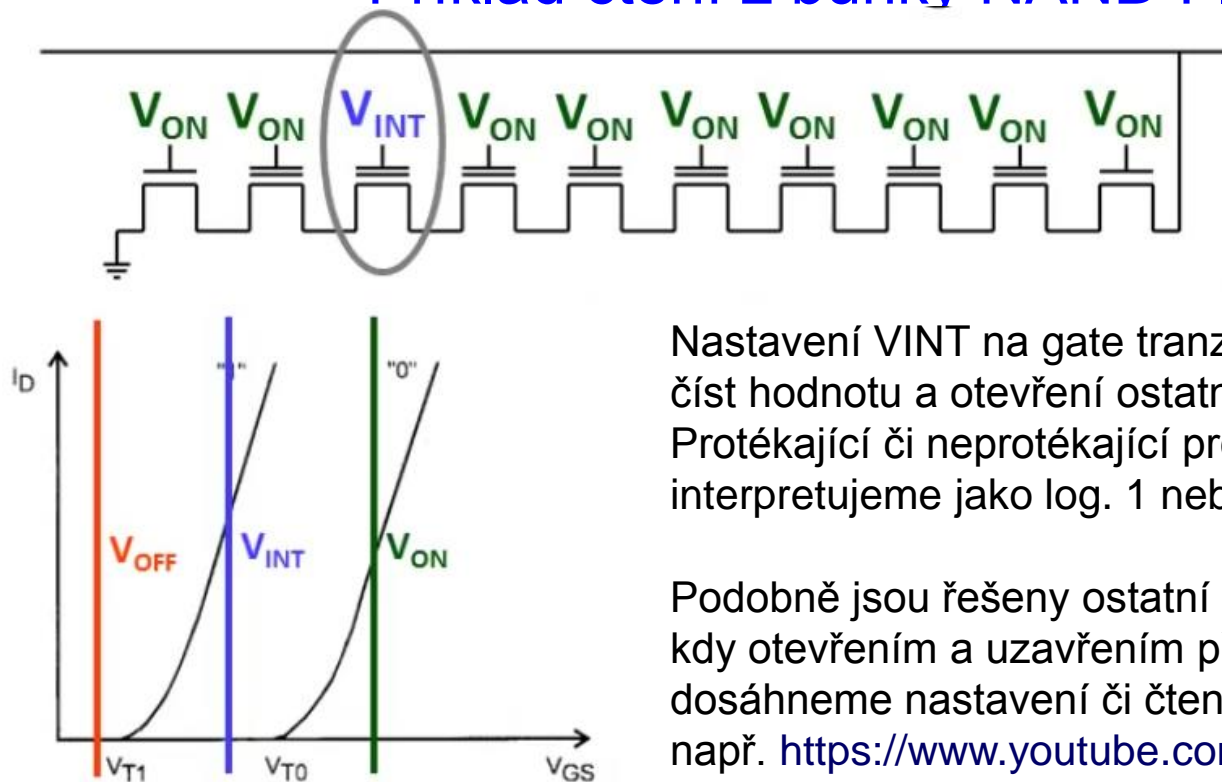
- NOR Flash** – čtení/zápis po jednotlivých buňkách (k dispozici kompletní adresové a datové rozhraní), mazání jako celek či po blocích, pomalé. Náročné na plochu => nízká kapacita (řádově MB). Použití: paměti pro firmware, uložení programu v mikrokontrolérech, OS v mobilech apod.



- NAND Flash** – manipulace po stránkách (~ 1000 bitů), komplexní řadič. Vysoká hustota a kapacita (řádově GB), rychlé operace čtení/zápis. Nižší spolehlivost, nutnost zavedení redundantních buněk a opravných kódů. Použití jako datová média – USB flash, SSD disky pro NB apod.



## Příklad čtení z buňky NAND Flash



Nastavení  $V_{INT}$  na gate tranzistoru, z něhož chceme číst hodnotu a otevření ostatních tranzistorů v buňce. Protékající či neprotékající proud tranzistorů interpretujeme jako log. 1 nebo 0.

Podobně jsou řešeny ostatní operace (též v NOR flash), kdy otevřením a uzavřením příslušných tranzistorů dosáhneme nastavení či čtení určité hodnoty (více viz např. <https://www.youtube.com/watch?v=s7JLXs5es7I>)

## Srovnání NOR a NAND flash z pohledu základních parametrů

	NOR Flash (typ.)	NAND Flash (typ.)
Random Read Access Time	80 ns/ 16 bit word	15 us /528 byte page
Sector Read Speed	13.2 MB/s	12.7 MB/s
Program Speed	1 word/ 10 us = 0.2 MB/s	528 bytes/250 us = 2.1 MB/s
Erase Speed	64kB/0.8 sec = 0.08 MB/s	16 kB/3 ms = 5.3 MB/s

# Porovnání (CIS 501, Dr. Joe Devietti, 2019)

- **Cost** - what can \$200 buy (2009)?
  - SRAM: 16MB
  - DRAM: 4,000MB (4GB) – 250x cheaper than SRAM
  - Flash: 64,000MB (64GB) – 16x cheaper than DRAM
  - Disk: 2,000,000MB (2TB) – 32x vs. Flash (512x vs. DRAM)
- **Latency**
  - SRAM: <1 to 2ns (on chip)
  - DRAM: ~50ns – 100x or more slower than SRAM
  - Flash: 75,000ns (75 microseconds) – 1500x vs. DRAM
  - Disk: 10,000,000ns (10ms) – 133x vs Flash (200,000x vs DRAM)
- **Bandwidth**
  - SRAM: 300GB/sec (e.g., 12-port 8-byte register file @ 3GHz)
  - DRAM: ~25GB/s
  - Flash: 0.25GB/s (250MB/s), 100x less than DRAM
  - Disk: 0.1 GB/s (100MB/s), 250x vs DRAM, **sequential** access only

Příště: Rychlá vyrovnávací paměť cache,  
paměťová hierarchie