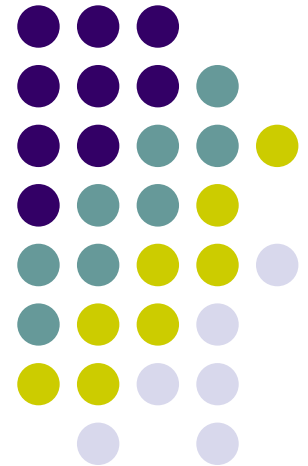


# Paměti

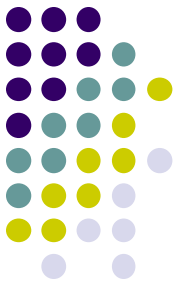
## 2. část

(Adresový dekodér, spojování chipů, SRAM, DRAM)

Hardware

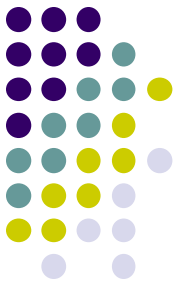


# Adresový dekodér



- Signály z **adresačních vstupů** jsou přivedeny do **dekodéru**
- Jeho úkolem je **binárně zakódovanou adresu** převést na signál, kterým bude **aktivována** jedna správná paměťová buňka
- Například dekodér v paměti s kapacitou 256 Bajtů bude mít *8 vstupů (A0..A7)* a *256 výstupních signálů*
- Bude-li na vstupu takového dekodéru kombinace 00010111, objeví se na výstupu dekodéru jednička na signálu, který vede do paměťové buňky s adresou 23 a na všech ostatních výstupech bude 0
- Bude-li na vstupu takového dekodéru kombinace 11111111, objeví se na výstupu dekodéru jednička na signálu, který vede do paměťové buňky s adresou 255 a na všech ostatních výstupech bude 0
- Bude-li na vstupu takového dekodéru kombinace 01010101, objeví se na výstupu dekodéru jednička na signálu, který vede do paměťové buňky s adresou 85 a na všech ostatních výstupech bude 0

# Dekodér



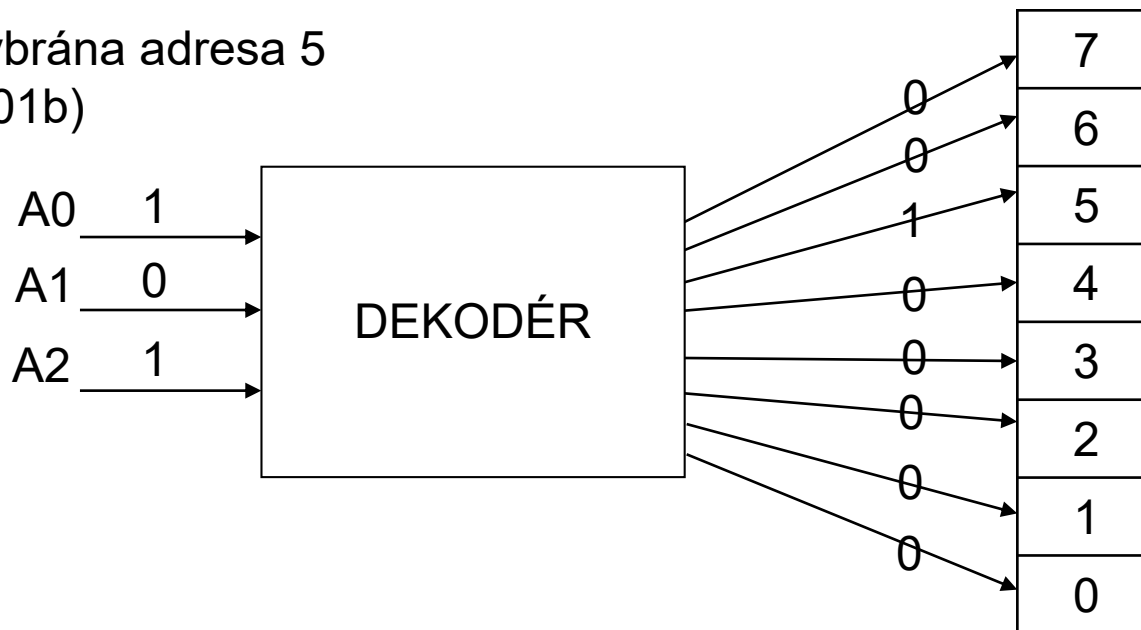
- Čím má paměť **větší kapacitu**, tím je její adresový dekodér **složitější**
- Například v paměti s kapacitou **1 MB** musí mít dekodér **20 vstupů** (A0..A19) a **1048576 výstupů**
- Pokud by na vstupu takového dekodéru byla kombinace 10110111110000010000, objeví na výstupu jednička na vývodu číslo 752656 a na všech ostatních výstupech bude nula
- Paměťový chip s kapacitou **1 GB** by již musel mít extrémně složitý dekodér s **30 adresačními vstupy** a **1073741824 výstupy** – to by bylo nerealizovatelné a proto u paměti s větší kapacitou dochází k rozložení dat do „řádků a sloupců“ a použití dvou jednodušších dekodérů

# Adresový dekodér



Paměť s kapacitou 8B

Je vybrána adresa 5  
(5=101b)



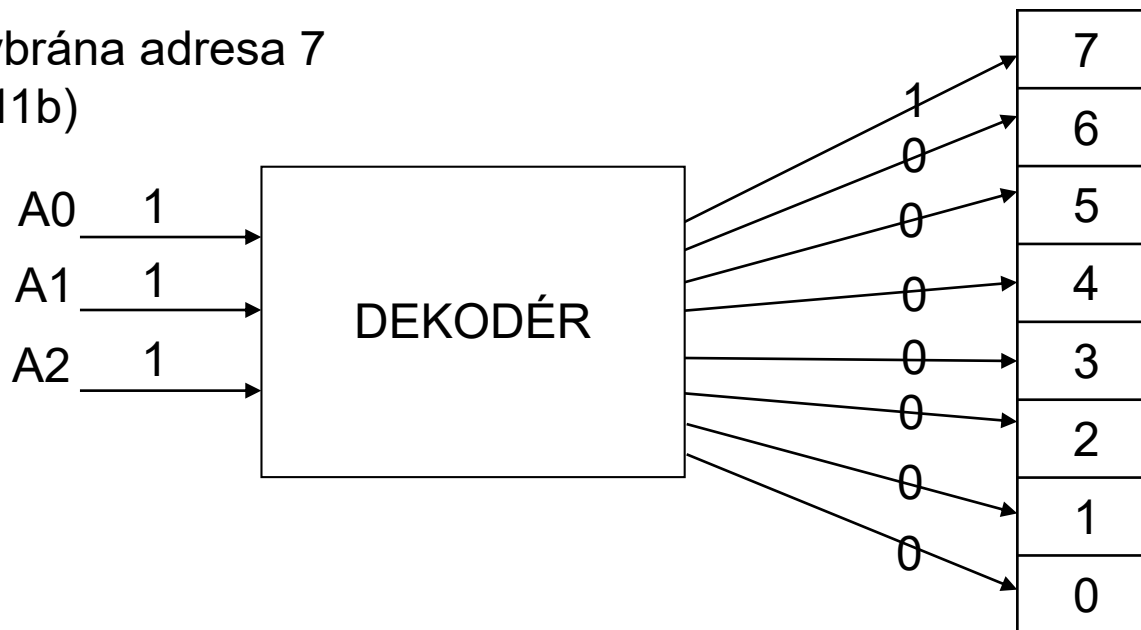
Na všech výstupech dekodéru jsou nuly, kromě signálu, který vede do paměťové buňky s adresou 5

# Adresový dekodér



Paměť s kapacitou 8B

Je vybrána adresa 7  
(7=111b)



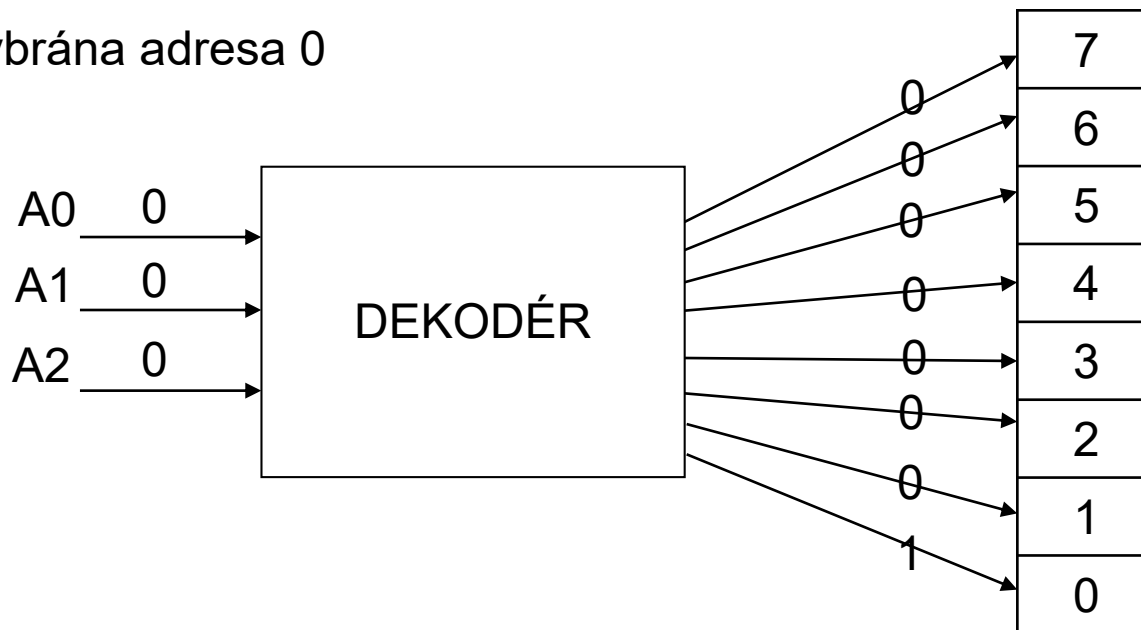
Na všech výstupech dekodéru jsou nuly, kromě signálu, který vede do paměťové buňky s adresou 7

# Adresový dekodér



Paměť s kapacitou 8B

Je vybrána adresa 0



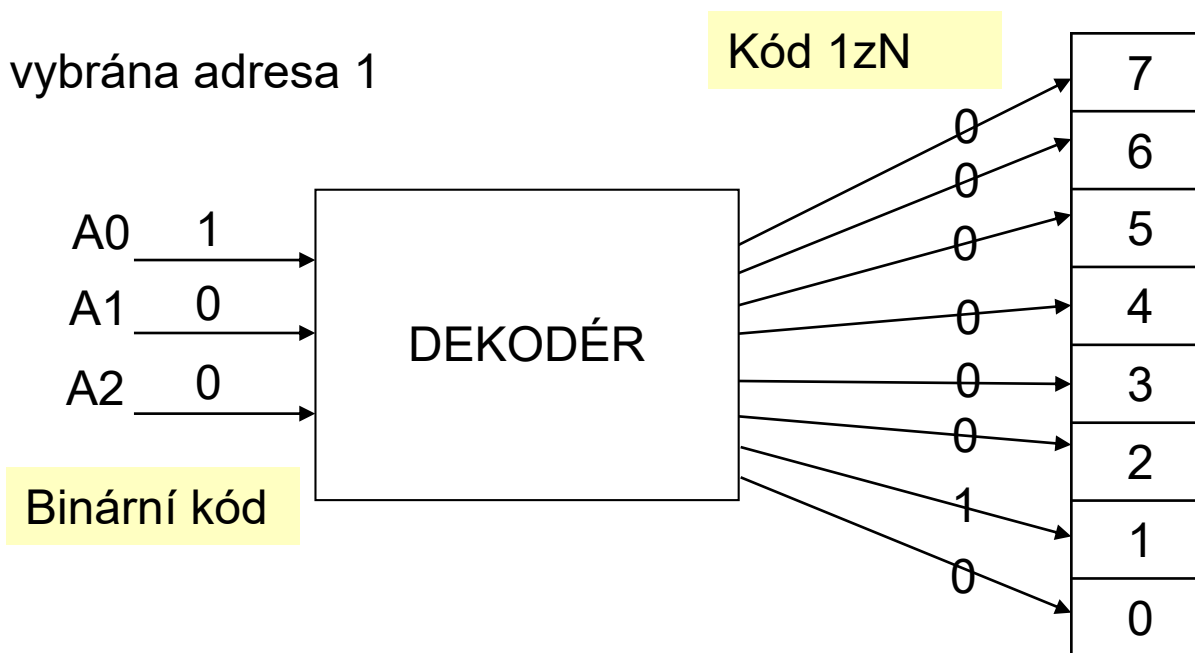
Na všech výstupech dekodéru jsou nuly, kromě signálu, který vede do paměťové buňky s adresou 0

# Adresový dekodér



Paměť s kapacitou 8B

Je vybrána adresa 1

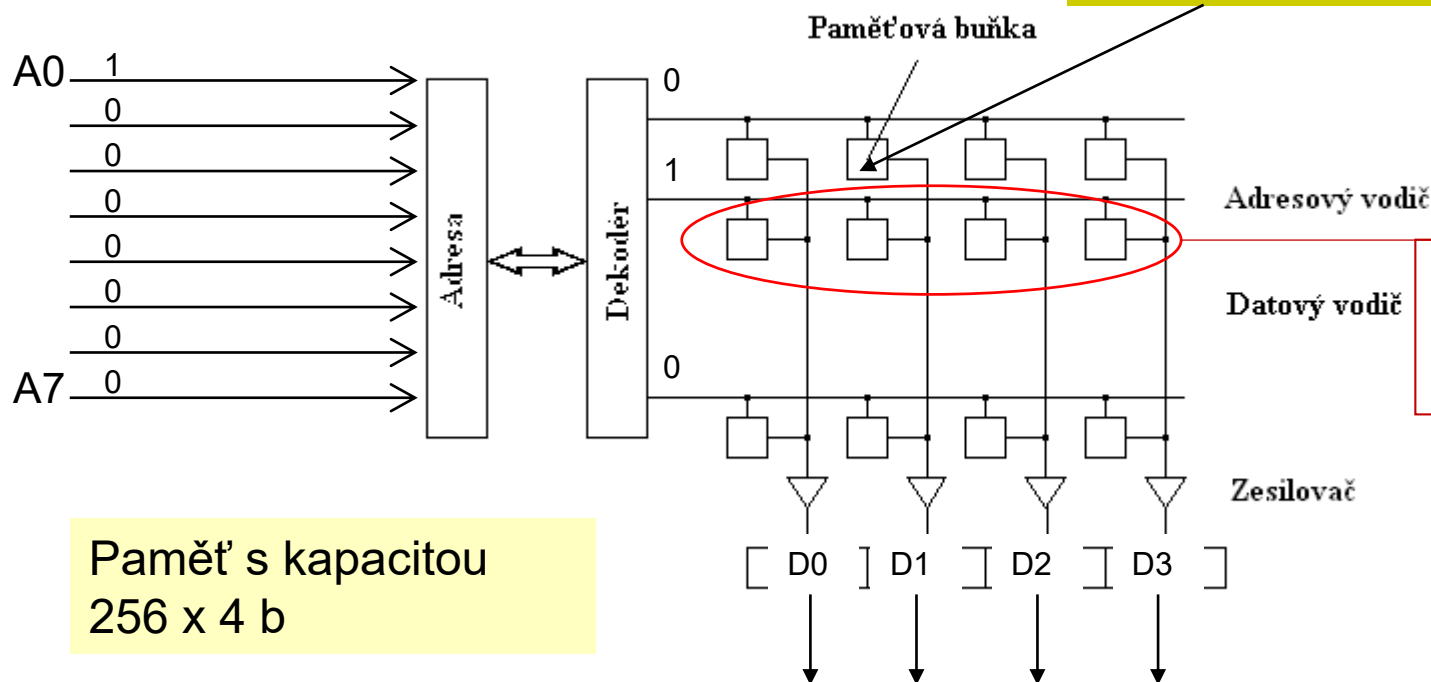


Na všech výstupech dekodéru jsou nuly, kromě signálu, který vede do paměťové buňky s adresou 1

# Obecná struktura paměti

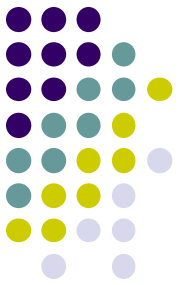


- Při přístupu do paměti (čtení nebo zápis) je vždy udána adresa paměťového místa, se kterým se bude pracovat
- Tato adresa je přivedena na vstup **dekodéru**
- Dekodér pak podle zadané adresy vybere **jeden z adresových vodičů** a nastaví na něm hodnotu logická 1 – tím se aktivují paměťové buňky, do který tento adresový vodič vede



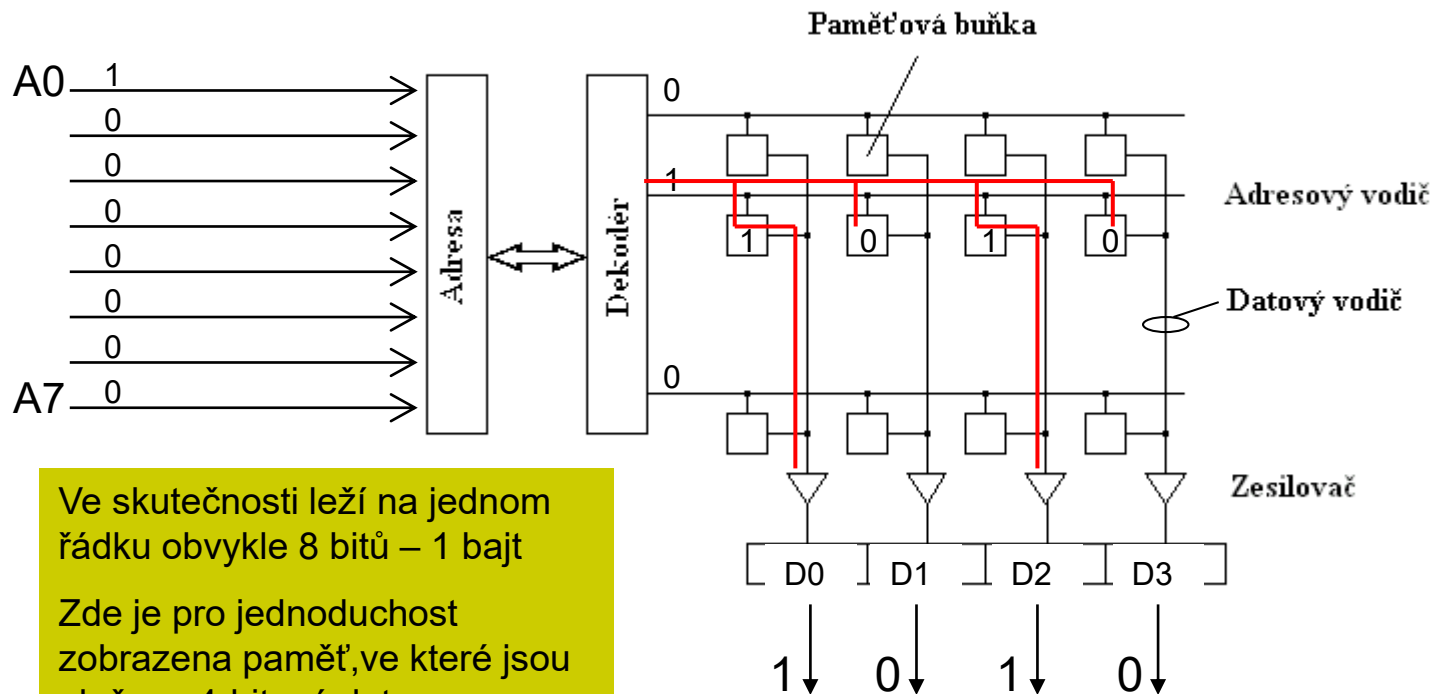
Tyto paměťové buňky jsou právě aktivované (zde jsou uložena 4-bitová data ležící na vybrané adrese 1)

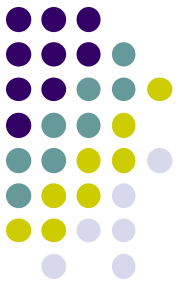




# Obecná struktura paměti

- Podle toho, jak jsou zapojeny jednotlivé paměťové buňky na příslušném řádku, který byl vybrán dekodérem, projde resp. neprojde hodnota logické jedničky na datové vodiče
- Informace je dále na koncích datových vodičů zesílена zesilovačem
- V případě, že hodnota logická jedna projde přes paměťovou buňku, obdržíme na výstupu hodnotu bitu 1. V opačném případě je na výstupu hodnota bitu 0.

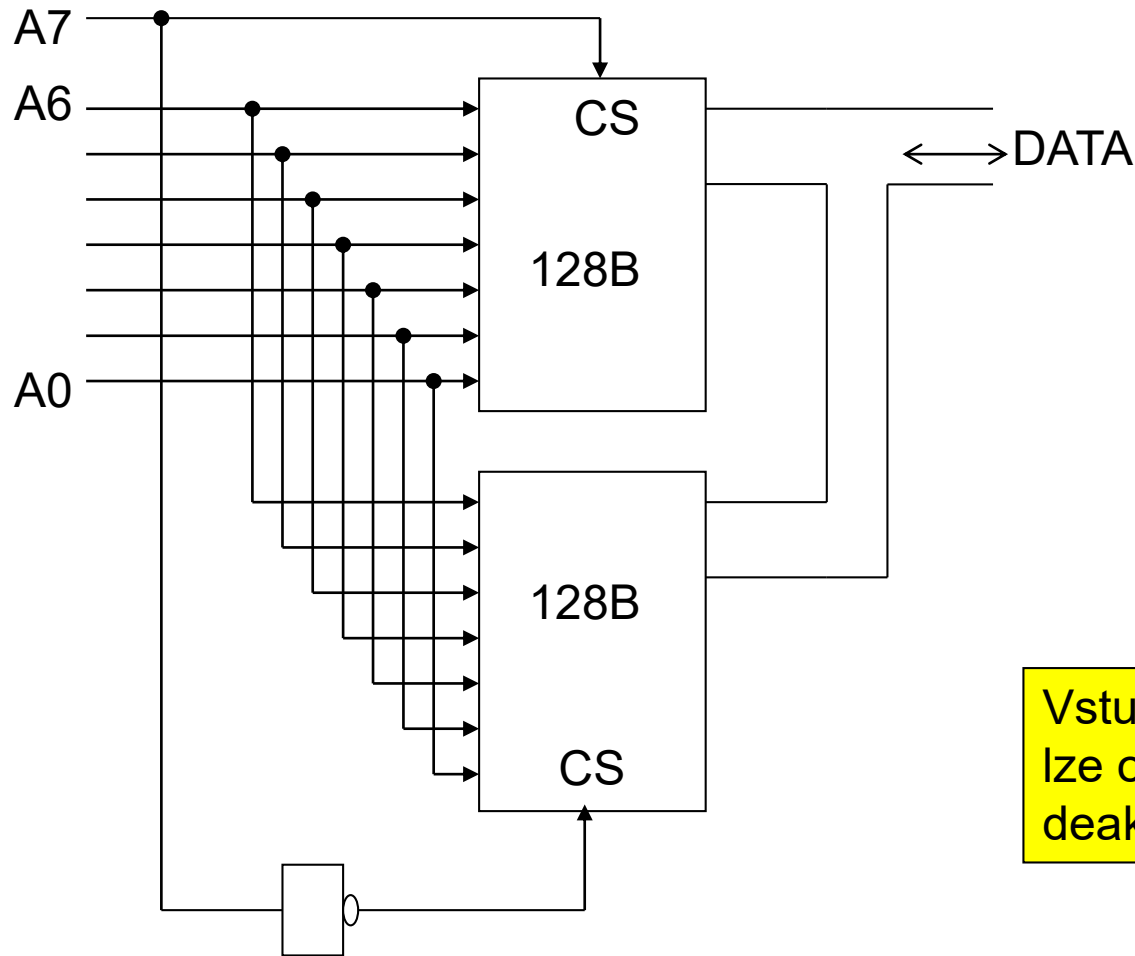
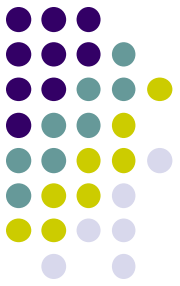




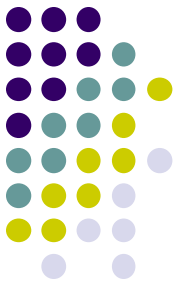
# Paměťový prostor a obvody

- Jak ze dvou paměťových obvodů vytvořit dvojnásobný paměťový prostor ?
- Paměťový obvod má kapacitu  $K$  bajtů a je tedy adresován  $n$ -bitovou adresou, přičemž  $K=2^n$
- Dvojnásobný paměťový prostor bude mít dvojnásobný rozsah adresovatelných paměťových míst a tedy adresa bude muset být o jeden bit širší. Původní  $n$ -bitová adresa bude mít nyní  **$n+1$  bitů**.
- Paměťové obvody budou nadále adresovány  $n$  bity.
- Ten bit který byl přidán navíc se použije k výběru jednoho ze dvou obvodů.
- Nejvyšší bit adresy tedy určuje, ve kterém ze dvou obvodů se nachází požadované paměťové místo. Zbylé bity adresují paměťové místo v rámci paměťového obvodu

# 256B paměť vzniklá spojením dvou 128B pamětí



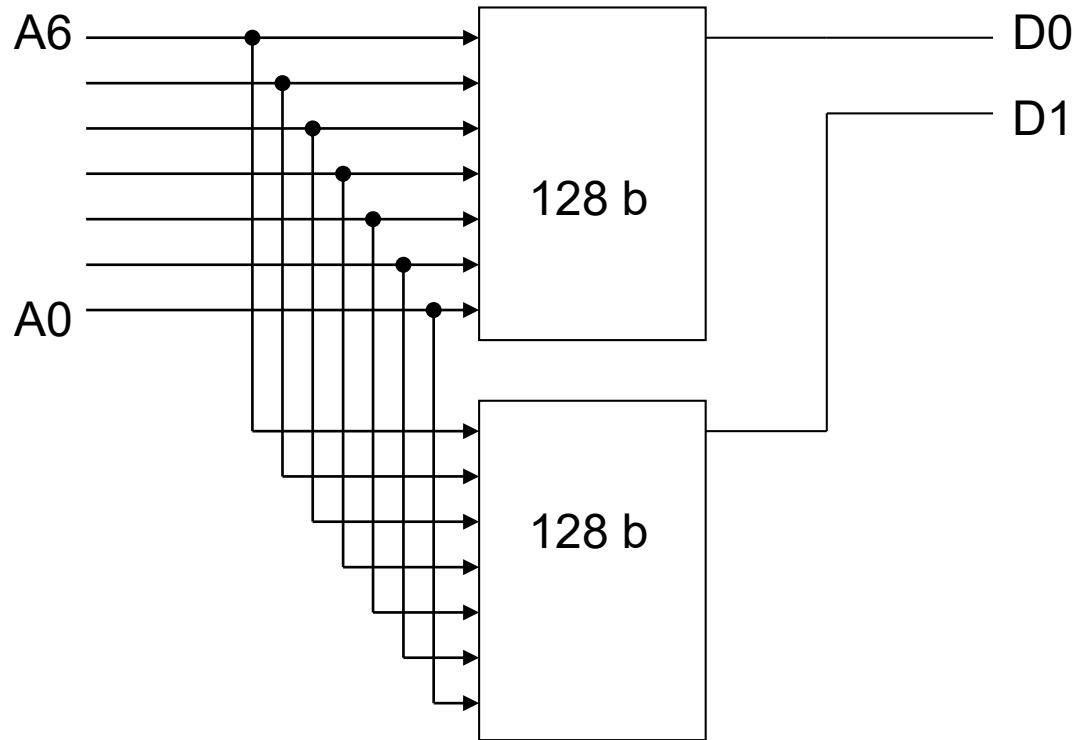
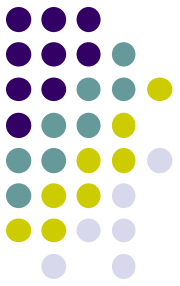
Vstupem CS (ChipSelect) lze obvod aktivovat nebo deaktivovat



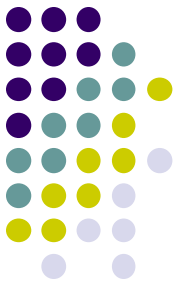
# Paměťový prostor a obvody

- Mnoho paměťových obvodů pracuje se slovem různé šířky.
- Běžné jsou například jednobitové paměti. Adresuje se v nich jednobitová paměťová buňka, do které se zapisuje nebo z ní čte jeden bit.
- Jak z takové paměti sestavit paměť se slovem zvolené šířky ?
- **Počet adres se nezmění**, ale rozšíří se množství informace uložené na dané adrese (bude se číst/zapisovat více bitů naráz)
- Více paměťových obvodů bude paralelně adresováno stejnou adresou
- Informace (data – jednotlivé bity) se rozloží na více částí podle šířky slova použitých paměťových obvodů

# Paměť 128x2b složená z obvodů 128x1b

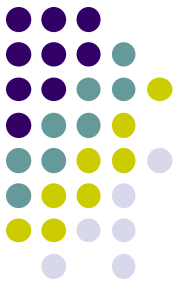


# Paměti DRAM



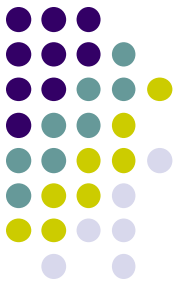
## Dynamická paměť RAM

- DRAM – Dynamic Random Acces Memory
- Nejčastěji používaný typ paměti ve výpočetní technice (operační paměť počítače PC)
- Informace uchována jen při zajištěném napájení – závislá na napájení
- Zapsaná hodnota uchována jako **náboj na parazitní kapacitě** tranzistoru MOSFET
- Informace o jednom bitu je uchována pomocí malého náboje uloženého v “kondenzátoru“, který nechtěně vznikl výrobou tranzistoru - Ve skutečnosti nejde o kondenzátor, ale strukturu na čipu, která se chová jako kondenzátor s malou kapacitou (cca 0,1 pF)
- Normálně se snažíme, aby kapacita mezi jednotlivými vývody tranzistoru byla co nejmenší (proto jí říkáme **parazitní**)
- Parazitní kapacitu mají i jiné součástky – vlastně i každý rezistor nebo vodič má nějakou kapacitu (lze ho trochu nabít)



# Paměti DRAM

- Čím nižší parazitní kapacitu tranzistor má, tím je rychlejší (přepínání není zpomaleno postupným nabíjením a vybíjením parazitních kapacit)
- Tranzistory v mikroprocesorech jsou tedy vyrobeny tak, aby jejich parazitní kapacita byla pokud možno nulová
- Výroba rychlých tranzistorů s téměř nulovou parazitní kapacitou je velmi složitá a drahá
- V paměti DRAM naopak můžeme použít snadno výrobitelné tranzistory s velkou parazitní kapacitou (proto bude levná)
- Tady totiž umíme tu parazitní kapacitu prakticky využít
- Nabitá parazitní kapacita = bit 1, vybitá kapacita = bit 0
- Paměťová buňka je velmi jednoduchá, což umožňuje dosažení vyšší kapacity na menší ploše
- Jednoduchá struktura = nízká cena paměti a možnost vyrobit chip s vysokou kapacitou

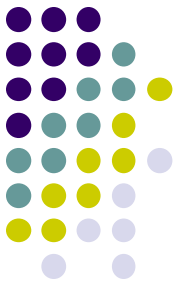


# Refresh DRAM

- Vzhledem k tomu, že parazitní kapacita uchovávající informaci je velmi malá a okolní prostředí není zcela nevodivé, dochází k **samovolnému vybíjení náboje** a tím i k postupnému přechodu z úrovně logické jedničky do úrovně nuly
- Dochází tedy k „zapomínání“ zapsané informace
- Z tohoto důvodu je nutné, aby byly údaje zapsané v dynamické paměti **periodicky obnovovány**
- Tento refresh provádí k tomu určené vlastní vnitřní obvody DRAM paměti a nemusí ho provádět mikroprocesor, který paměť využívá k ukládání dat
- Také při čtení se parazitní kapacita vybije tak, že se náboj spotřebuje na otevření tranzistorů – čtení je tedy destruktivní operací a přečtená (zničená) informace se musí obnovit
- Kvůli nutnosti provádět Refresh mají dynamické paměti velmi vysokou spotřebu el. energie – neustále je třeba přivádět nový náboj, který se ztrácí
- Kvůli vysoké spotřebě energie se mohou paměti DRAM i značně zahřívat a v některých zařízeních je dokonce nalezneme s chladičem
- Doba životnosti uložené informace by byla bez refreshe pouze pár milisekund



# Paměti SRAM



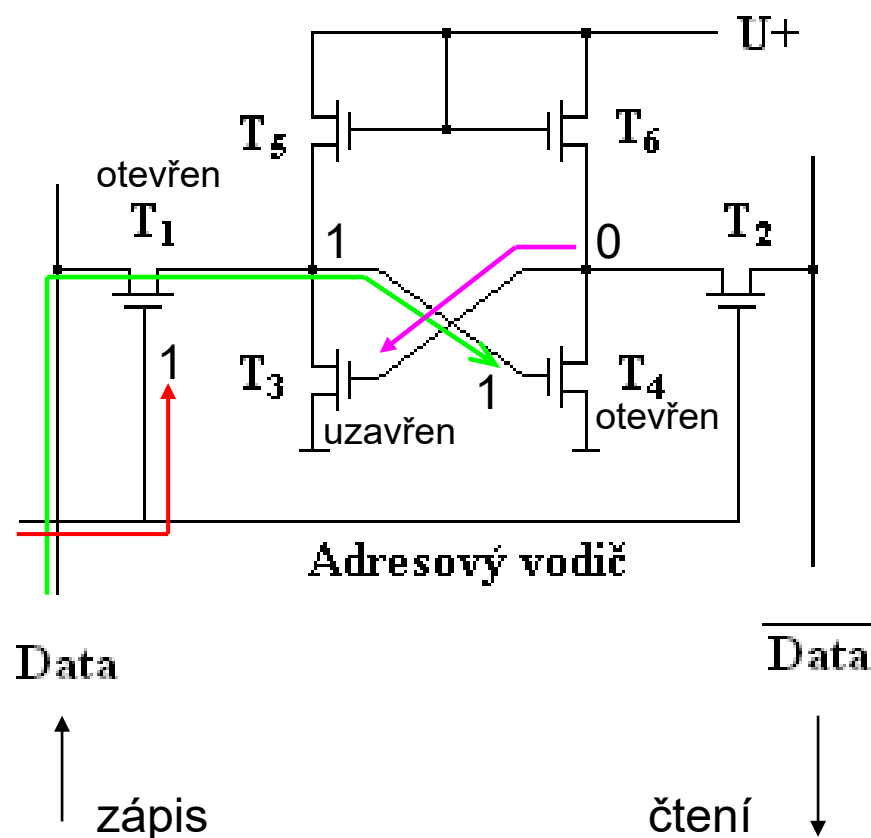
## Statická RAM

- Informace uchována jen při zajištěném napájení – závislá na napájení
- Slovem **statická** je myšleno to, že jednou zapsaný bit je v paměťové buňce držen po libovolně dlouhou dobu (pokud se nepřeruší napájení) a **není třeba provádět refresh**
- čtení není destruktivní, tzn. přečtením hodnoty bitu se obsah paměťové buňky neztrácí
- Použito je minimálně 6 tranzistorů pro jednu paměťovou buňku a z toho plyne vyšší cena za jeden bit a větší plocha, kterou na čipu jedna paměťová buňka zabírá
- Používá se úplně jiný typ tranzistorů než u pamětí DRAM (chip je vyroben jinou technologií).
- Tranzistory v paměti SRAM musí být téměř nulovou parazitní kapacitu (což je mnohem dražší)
- Základní buňku tvoří **bistabilní klopný obvod** (mnohem složitější buňka než u paměti dynamické, ale je rychlejší a není nutné ji občerstvovat)

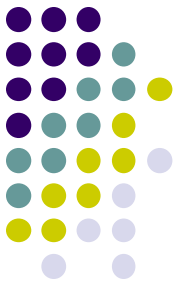
# SRAM – Statická paměť



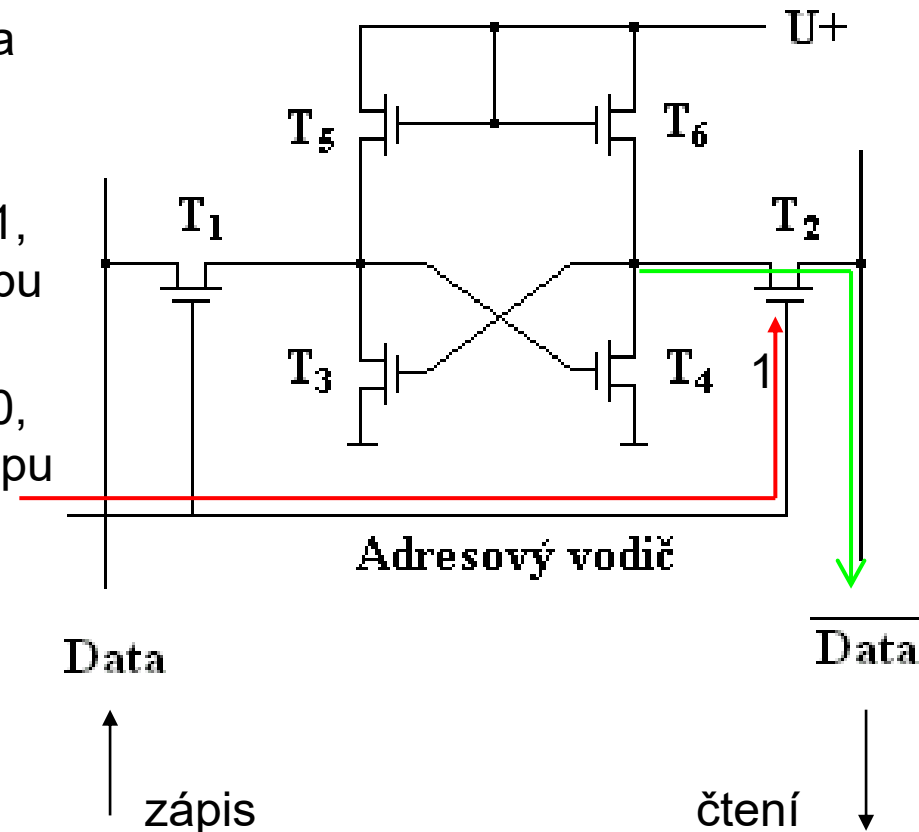
- 2 datové vodiče
- **Data** je určený k zápisu do paměti. Vodič označený jako **Data** se používá ke čtení. Hodnota na tomto vodiči je vždy opačná než hodnota uložená v paměti.
- **Při zápisu** se na adresový vodič umístí hodnota logická 1. Tranzistory T1 a T2 se otevřou. Na vodič Data se přivede zapisovaná hodnota (např 1). Tranzistor T1 je otevřen, takže jednička na vodiči Data otevře tranzistor T4 a tímto dojde k uzavření tranzistoru T3. Tento stav obvodu představuje uložení hodnoty 0 do paměti.

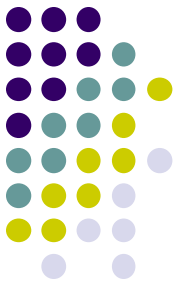


# SRAM – Statická paměť



- **Při čtení** je na adresový vodič přivedena hodnota logická 1, což způsobí otevření tranzistorů T1 a T2.
- Jestliže byla v paměti zapsána hodnota 1, je tranzistor T4 otevřen (tj. na jeho výstupu je hodnota 0)
- Jestliže byla v paměti zapsána hodnota 0, je tranzistor T4 uzavřen (tj. na jeho výstupu je hodnota 1)
- Výstupní log. úroveň je opačná oproti úrovni zapsané
- Výstupní log. Úroveň je čtena přes vodič /DATA





# Paměti SRAM

- Typická je **velmi nízká spotřeba energie** (nenabíjí se kondenzátory, překlápí se tranzistory a neprovádí se refresh)
- Pokud zrovna neprobíhá zápis ani čtení dat, je spotřeba elektrické energie téměř nulová
- K uchování dat v paměti (pokud se s ní nebude jinak pracovat) stačí i napájení z kondenzátoru, který ji udrží několik hodin
- Malá knoflíková baterie vydrží napájet paměť pro uchování dat několik let. (Paměť DRAM by jí vybila za pár hodin)
- Kratší přístupová doba než u DRAM – Paměti SRAM jsou velmi rychlé – obvykle jsou **nejrychlejší ze všech typů pamětí**
- Jde o nejčastěji používaný typ paměti v malé mikroprocesorové technice (**jednoduchá zařízení** jako cyklocomputer, řízení semaforu, blinkačka, kalkulačka, alarm apod..)
- **Například Arduino** - Paměť dat je SRAM
- Zde se používají právě proto, že mají nízkou spotřebu energie, netřeba provádět Refresh a paměti zde běžně používané nízké kapacity (1KB – 64 kB) se prakticky ani nevyrábějí jako dynamické
- Problémem je **velmi vysoká cena** rychlých SRAM pamětí.
- Nejlevnější (a docela pomalé) paměti SRAM mají cenu okolo 100000 Kč/GB
- Rychlé špičkové SRAM paměti mají cenu i přes 10 000 000 Kč/GB
  - (vypadá to hrozně, ale je to vlastně jen 10 Kč/KB)
- Z tohoto důvodu nelze SRAM použít jako hlavní operační paměť počítače
- V počítačích PC ji používáme jako **Cache paměť** – rychlá malá paměť k uložení nejčastěji používaných dat

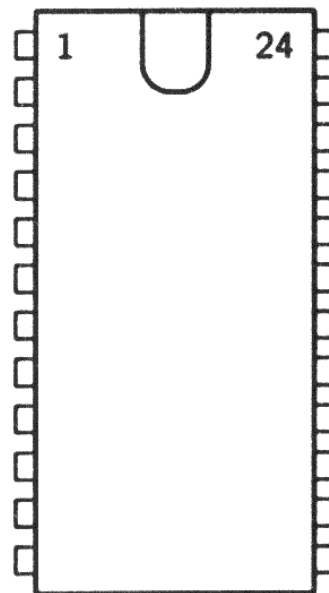
# Statické paměti SRAM



- Běžně lze zakoupit v prodejnách s elektrosoučástkami
- Typické obvody
  - 6116 – kapacita 2 kB
  - 6264 – kapacita 8 kB
  - 62256 – kapacita 32 kB

6116

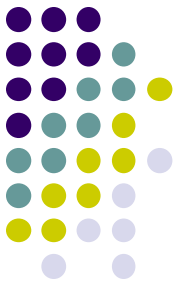
A7  
A6  
A5  
A4  
A3  
A2  
A1  
A0  
D0  
D1  
D2  
GND



6116

V<sub>CC</sub>  
A8  
A9  
 $\overline{\text{WE}}$   
 $\overline{\text{OE}}$   
A10  
 $\overline{\text{CE}}$   
D7  
D6  
D5  
D4  
D3

# SRAM x DRAM



	Statická	Dynamická
Paměťová buňka	složitá	jednoduchá
Kapacita	nízká	vysoká
Rychlost	vysoká	nízká
Cena	velmi vysoká (100000 až 10000000 Kč/GB)	relativně nízká (250 Kč/GB)
použití	cache	operační paměť
Spotřeba energie	Velmi nízká	vysoká
Refresh	Není potřeba	Je nutný