

4. Přenos dat, + jejich organizace rozdělení paměti

01 Sériový vs. Paralelní přenos

02 Princip připojení s otevřeným kolektorem

a) Včetně vytvoření sběrnice

03 Princip připojení s třístavovým zesilovačem

a) Včetně vytvoření sběrnice

04 Popis vybraných sběrnic

a) I²C, SPI, RS-232, IEEE 1284, USB

05 Rozdělení paměti v PC včetně jejich popisu

a) RWM vs. ROM paměti

06 Organizace paměti

a) Kapacita, hloubka paměti, délka datového slova

b) Paměťová mapa

4. Přenos dat, rozdělení paměti a organizace

Přenos dat – sériový a paralelní přenos, srovnání

Sériový přenos dat

Sériový přenos dat znamená, že **jednotlivé bity dat jsou přenášeny postupně (sekvenčně)** po **jediné datové lince**. Tím se výrazně **snižuje počet potřebných vodičů**.

Typické příklady sériových rozhraní:

- USB (*Universal Serial Bus*)
- FireWire (*IEEE 1394*)
- SPI (*Serial Peripheral Interface*)
- Ethernet
- SATA (*Serial ATA*)

Paralelní přenos dat

Při paralelním přenosu je **více bitů přenášeno současně**, každý po **vlastním vodiči**, v **jednom hodinovém taktu**.

Vlastnosti a nevýhody:

- Zpočátku **vyšší rychlost** (např. 8bitový přenos přenese 8 bitů najednou).
- **Rušení při vyšších frekvencích:** s rostoucí frekvencí a délkou vodičů **vznikají interference, odrazy a rozdílné časování (skew)** mezi vodiči.
- **Nevhodný pro delší vzdálenosti:** rozdílné zpoždění mezi vodiči způsobuje chyby.
- **Používán hlavně v minulosti** pro tiskárny a vnitřní propojení komponent v počítači.

Typické příklady paralelních sběrnic:

- ISA, ATA (*Parallel ATA*), SCSI, PCI, FSB (*Front Side Bus*)

Srovnání sériového a paralelního přenosu

Kritérium	Paralelní přenos	Sériový přenos
Rychlost	Potenciálně vyšší (více bitů najednou), ale závislá na synchronizaci všech linek	Moderní technologie umožňují velmi vysoké rychlosti na jedné lince
Délka kabelu	Omezená – vyšší zpoždění mezi vodiči vede k chybám	Vhodný i pro větší vzdálenosti
Složitost	Jednodušší na implementaci u krátkých vzdáleností, ale více vodičů	Vyžaduje převod do paralelní podoby přes USART apod.
Rušení	Vyšší kvůli více vodičům blízko u sebe	Nižší rušení a přeslechy
Cena	Více vodičů = vyšší náklady	Úspornější na kabeláž i konektory

Trendy a vývoj

- Díky poklesu ceny integrovaných obvodů a rostoucím nárokům na rychlost a spolehlivost se trh přesouvá k sériovým přenosům.
- **Nahrazení starších paralelních rozhraní:**
 - IEEE 1284 (*paralelní port*) → nahrazen **USB**
 - Parallel ATA (*PATA*) → nahrazen **Serial ATA (SATA)**
 - SCSI (*paralelní*) → nahrazen **FireWire**, později **USB 3.0 a vyšší**

4. Přenos dat, rozdělení paměti a organizace

Princip připojení s otevřeným kolektorem a třístavovým výstupem

Otevřený kolektor (Open Collector – OC)

Jedná se o **typ výstupního zapojení běžně** používaný u integrovaných obvodů (*např. logické členy, čítače, sběrnice obvodu*), kde je **výstup tvořen pouze tranzistorem typu NPN**.

Princip činnosti:

- Tranzistor má **kolektor vyveden na výstup**, **emitor** je spojen **se zemí (GND)**.
- **Báze je řízena logickým členem** uvnitř obvodu (*např. hradlem*).
- Stav výstupu závisí na sepnutí tranzistoru:
 - Tranzistor **sepnutý (ON)** → výstup je připojen k zemi → logická 0
 - Tranzistor **rozepnutý (OFF)** → výstup "visí ve vzduchu" → bez pull-up odporu je stav nedefinovaný

Pull-up rezistor:

- Na **výstup** je připojen **pull-up rezistor** (*typicky 4,7 kΩ až 10 kΩ*), který táhne napětí směrem k logické **1**, pokud tranzistor není sepnut.
- Zajišťuje, že když **žádný obvod neaktivuje výstup**, na sběrnici je logická **1**.
- Pokud kterýkoli výstup aktivuje logickou **0**, celý spoj má úroveň **0** → vzniká tzv. **drátová logika (wired AND)**.

Výhody a využití:

- Možnost připojit **zátěž s vyšším proudovým odběrem** (*tranzistor snese vyšší proud než běžný TTL výstup*).
- **Nezatěžuje** sběrnici, pokud je v **log. 1** (*tranzistor rozepnut*).
- **Vhodné pro více zařízení** na jedné sběrnici – žádný obvod nemůže "**vnutit**" **log. 1** (*např. I²C sběrnice*).
- **Chrání před kolizemi**, pokud více zařízení komunikuje současně.
- **Typické použití:** I²C sběrnice, porovnávací obvody, logika s prioritou.

Třístavový výstup (Three-state logic)

Třístavový výstup (tzv. *tri-state*) umožňuje výstupu kromě logické 0 a 1 i třetí stav: vysokou impedanci (Hi-Z).

Princip činnosti:

- Kromě datového řízení je výstup řízen **signálem "Enable"** (*např. OE – Output Enable*).
- Pokud je **OE aktivní** → výstup se chová **běžně** (*log. 0 nebo 1*).
- Pokud je **OE neaktivní** → výstup přechází do stavu **vysoké impedance (Hi-Z)** = **elektricky odpojený od sběrnice**.

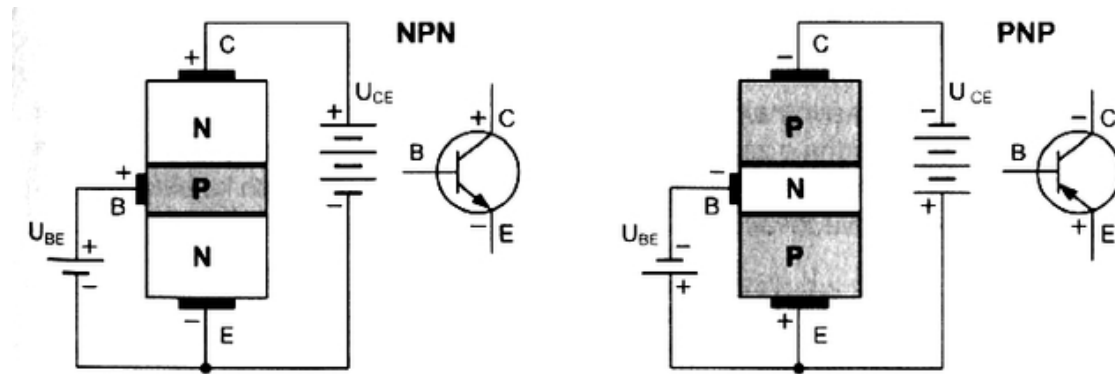
Vlastnosti a výhody:

- Umožňuje **sdílení jedné sběrnice více zařízeními**.
- **Zabrání konfliktům**, pokud pouze jeden výstup smí být aktivní.
- Když **jedno zařízení vysílá**, ostatní jsou ve stavu **Hi-Z** a **sběrnici nezatěžují**.
- **Efektivní využití vodičů** – šetří prostor na deskách plošných spojů.

Použití:

- Systémové sběrnice v PC, např. **adresové a datové sběrnice**.
- Řízení pamětí, vstupně-výstupních obvodů (*např. přístup k RAM*).

4. Přenos dat, rozdělení paměti a organizace



Shrnutí: Otevřený kolektor vs. Třístavový výstup

Vlastnost	Otevřený kolektor	Třístavový výstup
Stav výstupu	0 nebo „nepřipojeno“ (s pull-upem log. 1)	0, 1, nebo vysoká impedance (Hi-Z)
Řízení sběrnice	Pasivně – každý může stáhnout sběrnici do 0	Aktivně – pouze jeden výstup smí být aktivní
Potřeba pull-up rezistoru	Ano	Ne
Využití	I ² C, prioritní logika	Systémové sběrnice, řízení RAM
Odolnost proti kolizím	Vysoká (drátová logika)	Vyžaduje řízení pomocí OE signálu

Sběrnice – princip, dělení, parametry

Sběrnice je **soubor (svazek) vodičů**, který **propojuje jednotlivé součásti počítače** (např. CPU, RAM, GPU, periferní zařízení) a **zajišťuje přenos dat, adres a řídicích signálů** mezi nimi.

- Po sběrnici se mohou přenášet:
 - data (informace mezi komponentami),
 - adresy (určení cílového zařízení),
 - řídicí signály (např. zápis/čtení),
 - případně chybová hlášení.

Architektura připojení

- Každá periferní jednotka v PC (např. HDD, zvuková karta) je **připojena na některou sběrnici**.
- Tyto sběrnice jsou **propojeny s čipovou sadou (chipsetem)**, která se tradičně dělí na:
 - Severní můstek (Northbridge)** – propojuje **CPU, GPU, operační paměť (RAM)**.
 - Jižní můstek (Southbridge)** – připojuje **ostatní periférie**, jako jsou:
 - pevné disky (HDD, SSD)**, optické mechaniky, zvukové karty, USB zařízení, apod.

4. Přenos dat, rozdělení paměti a organizace

Dělení sběrnic

1. Lokální sběrnice

- Přímo **připojena k mikroprocesoru** nebo v rámci čipsetu.
- Slouží k propojení CPU s rychlými komponentami (*např. RAM*).
- **Nejrychlejší typ** sběrnice.
- **Příklady:**
 - **FSB** (*Front Side Bus*) – starší spojení **CPU ↔ RAM**, čipset.
 - **QPI** (*QuickPath Interconnect*) – **Intel** (*Core i7 a výš*).
 - **DMI** (*Direct Media Interface*) – spojení mezi CPU a čipsetem.

2. Rozšiřující sběrnice (*systémové*)

- Slouží k **připojení dalších komponent** – zvukové, grafické, síťové karty atd.
- Oproti lokálním **pomalejší**.
- **Zakončeny slotem nebo portem** na základní desce.
- **Příklady:**
 - **PCI, PCIe** (*PCI Express*) – moderní rozhraní pro grafické a rozšiřující karty.
 - **USB, SATA, M.2** – externí a úložné zařízení.

Základní parametry sběrnice

Parametr	Význam	Jednotka
Šířka sběrnice	Počet bitů, které může přenést současně	bit
Frekvence sběrnice	Počet operací za sekundu (rychlost taktu)	Hz
Přenosová rychlost (propustnost)	Kolik dat se přenese za jednotku času	bit/s (bps)
Přístupová doba	Doba čekání zařízení na přístup ke sběrnici	sekundy (s)

Rozdělení sběrnic

Sběrnici lze rozdělit podle více hledisek: **funkce vodičů, typ přenosu dat, synchronizace a směr přenosu.**

1) Podle funkce vodičů

Řídicí sběrnice (*Control Bus*)

- Slouží k přenosu **řídících signálů mezi zařízeními**.
- Přenáší např.:
 - **CLK** (*Clock*) – hodinový signál pro synchronizaci.
 - **Chip Select** (*CS*) – výběr konkrétního zařízení.
 - **Read/Write** – určuje směr operace (čtení/zápis).

4. Přenos dat, rozdělení paměti a organizace

Datová sběrnice (Data Bus)

- Slouží k **přenosu vlastních dat** mezi procesorem, pamětí a periferiemi.
- Označuje se např. **D0–D7** (pro 8bitovou sběrnici).
- **Obousměrná** – slouží pro čtení i zápis dat.

Adresní sběrnice (Address Bus)

- Přenáší **adresu zařízení nebo paměťové buňky**, se kterou má CPU pracovat.
- Označuje se např. **A0–A7** (8bitová adresace).
- **Jednosměrná** – pouze z CPU do periférií.

2) Podle typu přenosu

Sériový přenos

- Bity jsou **posílány postupně** (sekvenčně) po jednom vodiči v závislosti na CLK.
- **Nížší počet vodičů, menší rušení**, vhodné pro **delší vzdálenosti**.
- **Příklady sběrnic:**
 - RS-232
 - I²C, SPI
 - USB, FireWire
 - PCI Express
 - SATA, eSATA

Paralelní přenos

- Všechny bity jsou přenášeny **najednou po více vodičích** zároveň (např. 8, 16 nebo 32 bitů).
- **Vyšší rychlost na krátkou vzdálenost**, ale problém se synchronizací.
- **Příklady sběrnic:**
 - IEEE 1284 (LPT – tiskárna)
 - ISA, SCSI
 - PCI
 - PATA (Parallel ATA)

3) Podle synchronizace (časování)

Synchronní sběrnice

- Všechna zařízení jsou **řízena společným hodinovým signálem (CLK)** generovaným masterem.
- **Vysoká přesnost a jednoduché řízení**.
- **Příklady sběrnic:**
 - I²C
 - SPI

Asynchronní sběrnice

- Každé zařízení má **vlastní hodinový signál**, ke kterému se **musí v době přenosu synchronizovat**.
- **Vyšší flexibilita, ale složitější řízení**.
- **Příklady sběrnic:**
 - RS-232 (např. sériový port)

4. Přenos dat, rozdělení paměti a organizace

Typ	Popis	Příklad
Simplex	Přenos pouze jedním směrem	LPT (paralelní port pro tiskárny)
Half duplex	Přenos oběma směry, ale ne současně	I ² C
Full duplex	Přenos oběma směry současně	USB 3.0, Ethernet (moderní)

I²C sběrnice (Inter-Integrated Circuit)

- Vyvinuta firmou **Philips** (*dnes NXP*).
- Název: I²C = **Inter-Integrated Circuit**.
- Sběrnice určená pro **komunikaci mezi nízkorychlostními zařízeními** na krátké vzdálenosti (*např. uvnitř jednoho zařízení, mezi IC obvody na desce*).

Vlastnost	Popis
Počet vodičů	2 vodiče: SDA (data) a SCL (hodiny)
Přenos	Sériový, obousměrný, half-duplexní
Synchronní	Hodinový signál je generován masterem
Topologie	Multi-master / multi-slave – na sběrnici může být více zařízení, které se střídají v řízení komunikace
Typ výstupu	Otevřený kolektor – nutné použití pull-up rezistorů, aby byla v klidovém stavu úroveň log 1
Rychlost přenosu	- Standard mode: 100 kb/s

Použití sběrnice I²C

- **Komunikace s nízkorychlostními periferiemi, jako jsou:**
 - LCD displeje
 - AD/DA převodníky
 - RTC (*hodiny reálného času*)
 - Externí paměti EEPROM
 - Senzory (*např. teplota, tlak*)

4. Přenos dat, rozdělení paměti a organizace

Master–Slave princip

- Komunikaci řídí Master:
 - **Generuje CLK (SCL) signál.**
 - **Zahajuje a ukončuje přenos.**
- Slave zařízení **odpovídá na základě adresace.**
- Sběrnice umožňuje **více masterů (multi-master)**, přičemž se **řeší kolize řízením priority a detekcí kolize.**

Princip přenosu dat

1. Klidový stav:

- Oba vodiče SDA i SCL jsou v **log. 1** (díky pull-up rezistorům).

2. Zahájení přenosu – START bit:

- SDA přejde z **1 do 0**, zatímco SCL zůstává v 1 → **začátek komunikace.**

3. Adresa + R/W bit:

- Master vyšle **7bitovou (nebo 10bitovou) adresu zařízení + 1 bit (R/W)** určující směr:
 - **0 = zápis, 1 = čtení.**

4. Potvrzení – ACK bit:

- Po každém 1 bytu (8 bitů) následuje **ACKNOWLEDGE BIT**:
 - **ACK = 0** → potvrzení příjmu.
 - **ACK = 1** → chyba, nutné opakovat přenos.

5. Přenos dat:

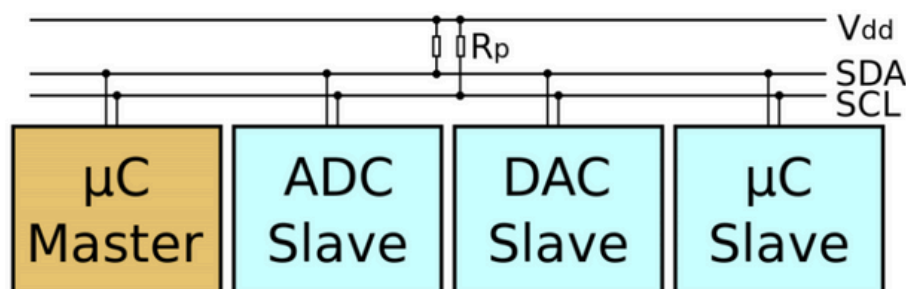
- Přenos probíhá **po 8 bitech (1 byte)**:
 - Od **MSB (nejvýznamnější bit)** po **LSB**.
 - **SDA** mění hodnotu, když je **SCL** v **log. 0**.
 - 1 bit je přenesen při každém pulzu CLK (SCL).

6. Ukončení přenosu – STOP bit:

- **SDA** přejde z **0 do 1**, zatímco **SCL** je v **1** → **značí konec přenosu.**

Bezpečnost přenosu

- Pokud více zařízení začne vysílat současně, sběrnice umožňuje detekci kolize:
 - Vysílající zařízení porovnává, co posílá, s tím, co "vidí" na sběrnici.
 - Pokud dojde k rozdílu, vysílání přeruší – žádné poškození přenosu.



4. Přenos dat, rozdělení paměti a organizace

SPI sběrnice (Serial Peripheral Interface)

- Vyvinuta firmou **Motorola**.
- Určena pro **rychlou a jednoduchou komunikaci mezi mikrokontrolérem a periferiemi**.
- Využívá se v **embedded systémech**, kde je důležitá **rychlost a jednoduchost**.
- **Typ přenosu:** Sériový, synchronní, obousměrný, full duplex
- **Topologie:** 1 master, více slave zařízení (multi-slave)
- → **SCLK** (*Serial Clock*) – hodiny generované masterem
- → **MOSI** (*Master Out Slave In*) – data z mastera do slave
- → **MISO** (*Master In Slave Out*) – data ze slave do mastera
- → **SS / CS** (*Slave Select / Chip Select*) – výběr konkrétního slave zařízení
- Master generuje **hodinový signál (SCLK)** a pomocí **CS (chip select)** **vybere**, se kterým slave zařízením bude komunikovat.
- **Komunikace je full duplexní** – obě zařízení posílají data současně.
- Každý **slave má vlastní CS vodič** – nutné při připojení více zařízení.

Použití SPI sběrnice

- Stejně jako **I²C** se používá pro **nízkorychlostní periferie**, ale s vyšší **přenosovou rychlostí**.
- **Typické použití:**
 - LCD displeje
 - A/D a D/A převodníky
 - EEPROM, Flash paměti
 - Senzory, hodiny reálného času (RTC)

Přenosová rychlost

- **Typické rozsahy:** 1 až 66,7 Mb/s (*v praxi až 100 Mb/s u moderních zařízení*).
- Záleží na **maximální frekvenci**, kterou zvládnou **master i slave zařízení**.
- **Vysoká rychlost** = výhoda oproti I²C.

Detail připojení a provozu

1 master, více slave zařízení (multi-slave)

- Každé **slave** zařízení má vlastní **CS (chip select) pin**.
- Master vybírá slave zařízení tím, že **aktivuje jeho CS pin** (*obvykle log. 0*).
- Při **neaktivním CS** je **slave nečinný** (*nevysílá ani neposlouchá*).

Vodič	Zkratka	Směr přenosu
SCLK	Serial Clock	Master → Slave
MOSI	Master Out Slave In	Master → Slave
MISO	Master In Slave Out	Slave → Master
SS / CS	Slave Select	Master → Slave (výběr zařízení)

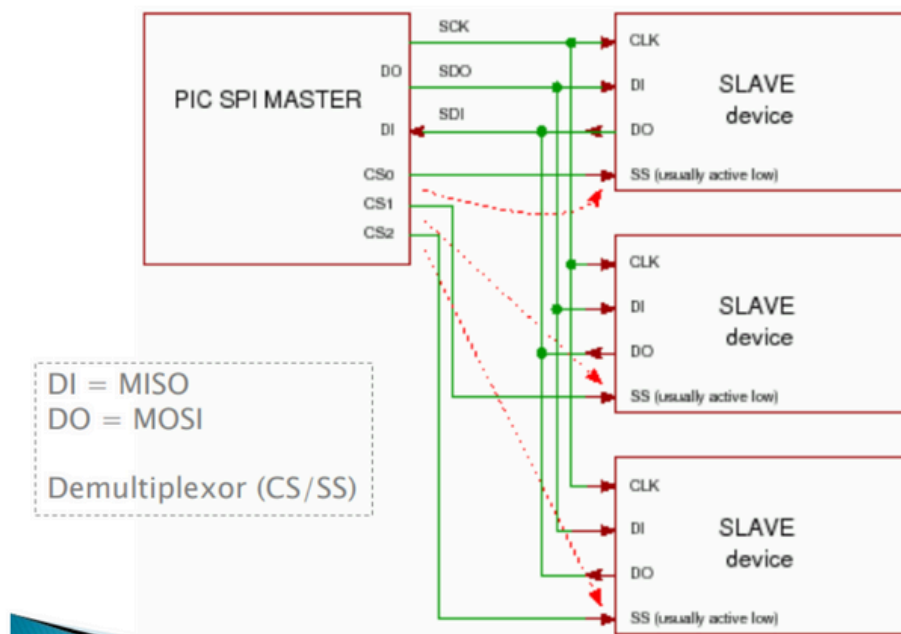
4. Přenos dat, rozdělení paměti a organizace

Výhody SPI

- **Vysoká rychlost** přenosu
- **Jednoduchý hardware** a řízení
- **Full duplexní** komunikace
- Každý slave je adresován samostatným CS pinem

Nevýhody SPI

- Nutnost **1 vodiče CS** na **každé slave** zařízení
- **Žádná standardizovaná detekce** kolizí jako u I²C
- Kratší dosah než I²C (*není navržena pro dlouhé vzdálenosti*)



Sběrnice RS-232 – Sériový port (COM)

- Jedna z **nejstarších a nejpoužívanějších sériových sběrnic**.
- Využívá se k propojení dvou zařízení: **vysílač ↔ přijímač**.
- Založena na standardu **EIA RS-232**.

Parametr	Popis
Typ přenosu	Sériový, asynchronní, full duplex
Počet zařízení	2 (bod–bod komunikace)
Hodinový signál (CLK)	Každé zařízení má vlastní CLK → při přenosu se synchronizují na základě časování bitů
Maximální rychlost	115 200 Bd (baudů) – počet změn na vedení za sekundu
Maximální vzdálenost	15 metrů (čím delší kabel, tím nižší rychlost a vyšší šum)
Použité kódování	Např. ASCII
Typ konektoru	D-Sub (např. DE-9, DB-25)

4. Přenos dat, rozdělení paměti a organizace

Řízení přenosu

- **Hardwarové řízení (HW):**
 - Použití řídicích vodičů (např. *RTS/CTS* – *Ready to Send / Clear to Send*).
 - Přenos je povolen až po **potvrzení připravenosti přijímače**.
- **Softwarové řízení (SW):**
 - Použití **speciálních řídicích znaků** (např. *XON/XOFF*) v datovém toku.

Datový rámec (formát přenosu)

- **Start bit** – 1 bit (*log. 0*) označující začátek přenosu.
- **Datové bity** – typicky 7 nebo 8 bitů (*od LSB → MSB*).
- **Paritní bit** – volitelný (*pro sudou nebo lichou paritu*).
- **Stop bity** – 1 nebo 2 bity (*log. 1*) – označují konec rámce.

Sběrnice IEEE 1284 – Paralelní port (LPT)

- Známá jako **LPT** (*Line Print Terminal*) – původně určená pro **připojení tiskáren**.
- **Paralelní přenos dat** – více bitů současně.
- **Typ přenosu:** Paralelní, původně simplexní, později half duplexní
- **Vodiče:** 8 datových, 9 řídicích, 8 zemnicích (*GND*)
- **Maximální vzdálenost:** 5 metrů, ideálně do 2 m
- **Použití:** Tiskárny, starší skenery, některé měřicí přístroje
- **Rychlost:** Rychlejší než RS-232 při krátkých vzdálenostech

Příklady dalších sběrnic

Sériové sběrnice

Sběrnice	Vlastnosti
RS-422	Vylepšený RS-232, diferenciální signál, větší odolnost vůči rušení, větší vzdálenosti
RS-485	Podobné jako RS-422, ale umožňuje více zařízení na sběrnici (multi-drop) – až 32
FireWire (IEEE 1394)	Vysoce výkonná sériová sběrnice, dříve konkurent USB, vhodná pro video a audio přenosy, podpora plug-and-play

Paralelní sběrnice

GPB (IEEE 488): Paralelní sběrnice používaná ve **vědě a průmyslu**, určená k propojení měřicích přístrojů, **až 15 zařízení**

4. Přenos dat, rozdělení paměti a organizace

USB – Universal Serial Bus

- USB je standard určený k **připojování různých periférií k počítači nebo jinému zařízení**.
- Vyvinuto jako **náhrada sériového portu RS-232 a paralelního portu (LPT)**.
- **Umožňuje:**
 - **Přenos dat mezi dvěma zařízeními** (např. PC ↔ mobil, PC ↔ externí disk).
 - **Napájení** připojených zařízení (např. lampičky, větráčky, telefony, powerbanky).

Parametr	Popis
Topologie	Hvězdicová síť – v centru je HUB
Počet zařízení	Až 127 zařízení na jedné sběrnici
Adresace	Každé zařízení má jedinečnou adresu
Plug & Play	Připojení za chodu bez nutnosti restartu systému

Rozdělení USB podle verzí

USB 1.0

- **Konektor:** 4 piny
 - **2 datové** vodiče (D^+ , D^-)
 - **1 napájecí** vodič (V_{cc})
 - **1 zem** (GND)
- **Half duplexní přenos** (data tečou jedním směrem v jednom okamžiku).
- Maximální rychlost: **12 Mb/s** (Full Speed).
- **Typy konektorů:**
 - **USB A** (standardní)
 - **USB B** (pro zařízení, např. tiskárny)

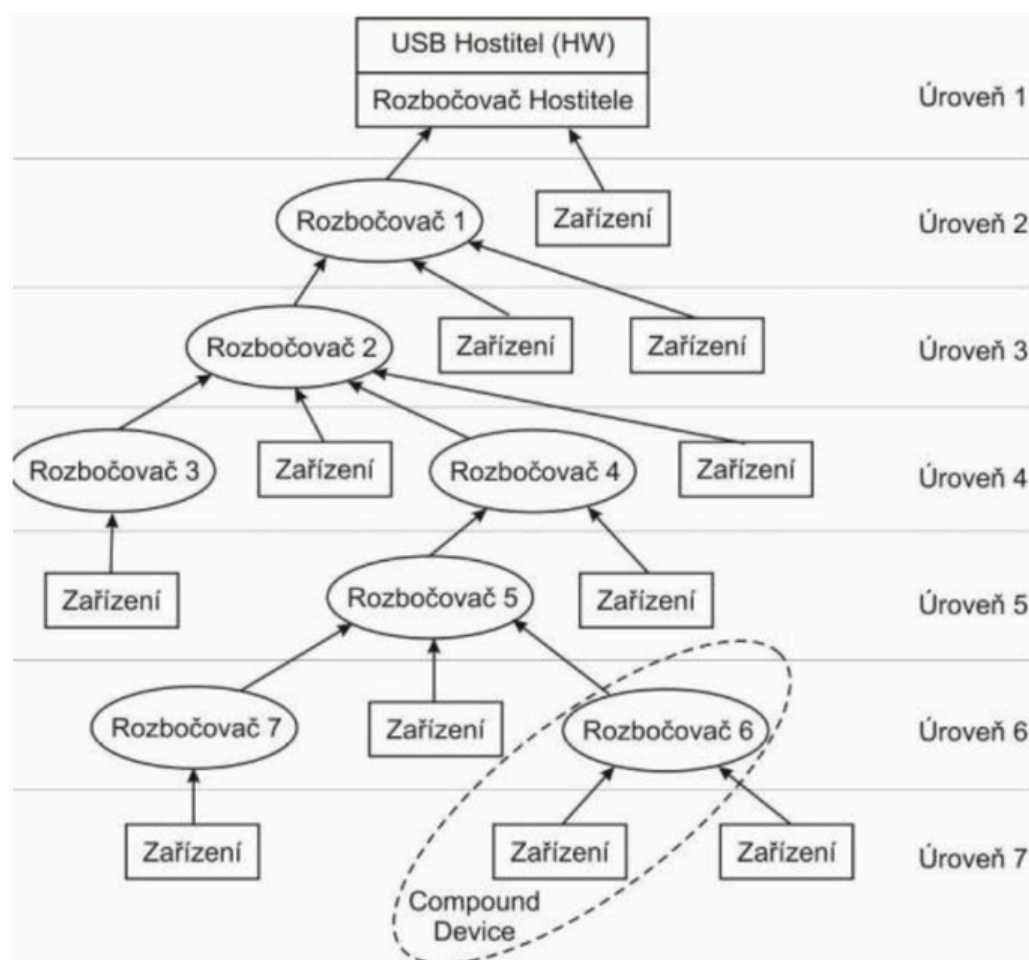
USB 2.0

- **Zpětná kompatibilita** s USB 1.0.
- **Half duplexní přenos**.
- Maximální rychlost: **480 Mb/s** (High Speed).
- **Nové typy konektorů:**
 - **Mini USB A, Mini USB B**
 - **Micro USB** (zejména pro mobily a menší zařízení)

USB 3.0

- **Zpětná kompatibilita** s **USB 2.0 a USB 1.0**.
- **Full duplexní přenos** (možnost současného vysílání i příjmu dat).
- Maximální rychlost: **5 Gb/s** (SuperSpeed).
- **Konstrukce konektoru:**
 - **8 vodičů:**
 - **6 datových** (dvě datové páry pro full duplex)
 - **2 napájecí** (V_{cc} a GND)

4. Přenos dat, rozdělení paměti a organizace



Princip činnosti USB (jednodušeji)

- Po sběrnici neustále **kolují datové a servisní rámce**.
- Hostitelský řadič (např. počítač) **vysílá USB paket**:
 - **Paket obsahuje: typ a směr přenosu, adresu cílového zařízení.**
- Zařízení si **rozpozná svou adresu** a připraví se na komunikaci.
- Zařízení **odpoví datovým paketem**, který obsahuje i **kontrolní bit** pro ověření správnosti.
- Po úspěšném přenosu se odešle handshake paket, který potvrzuje úspěšné přijetí dat.

Verze	Maximální rychlost	Typ přenosu	Počet vodičů	Nové konektory
USB 1.0	12 Mb/s	Half duplex	4	USB A, B
USB 2.0	480 Mb/s	Half duplex	4	Mini A/B, Micro USB
USB 3.0	5 Gb/s	Full duplex	8	SuperSpeed konektory

4. Přenos dat, rozdělení paměti a organizace

Rozdělení pamětí a jejich popis

1. Paměti závislé na napájení – RWM (*Read Write Memory*)

- Data se po odpojení napájení **ztratí**.
- Typicky jde o paměti s **možností zápisu i čtení**.
- **RAM** (*Random Access Memory*)
- **Paměť s nahodilým přístupem** – umožňuje **přímý přístup ke kterémukoliv místu paměti**.
 - **SRAM** (*statická RAM*)
 - **Statická paměť** – data jsou udržována bez potřeby obnovy
 - **Použití:** cache paměti (*L1, L2, L3*).
 - **Rychlejší, dražší, nižší kapacita** než DRAM.
 - **DRAM** (*Dynamic RAM*)
 - **Dynamická paměť** – data uchovává pomocí **kondenzátorů**, které je nutné **pravidelně obnovovat** (*refresh*).
 - **Použití:** operační paměť (*RAM*) v počítačích.
 - **Pomalejší, levnější, vyšší kapacita**.
 - Adresa je vystavována **na dvakrát** (*řádková a sloupcová adresa*) kvůli **snížení počtu pinů**.
 - **NoRAM** (*Non-random access Memory*)
 - Paměť s **přímým řízením přístupu** (*není náhodný jako u RAM*).
 - Používá **FIFO** (*First In First Out*) – **fronta** (*např. buffery*).
 - Používá **LIFO** (*Last In First Out*) – **zásobník** (*stack memory*).

2) Paměti nezávislé na napájení – ROM (*Read Only Memory*)

- Data zůstanou **zachována i po odpojení napájení**.
- Určené **převážně pro čtení**, některé verze umožňují **mazání a opětovné zápisy**.
 - **ROM** (*Mask ROM*)
 - **Pevně naprogramovaná** při výrobě pomocí masky.
 - Velká životnost.
 - **Použití:** **firmware** zařízení (*např. ROM BIOS*).
 - **Nelze přepisovat běžným způsobem** – vyžaduje vyšší napětí.
 - **PROM** (*Programmable ROM*)
 - **Programování po výrobě** podle potřeby zákazníka.
 - **Použití:** firmware zařízení.
 - **EPROM** (*Erasable Programmable ROM*)
 - Smazání vyžaduje **vystavení čipu ultrafialovému světlu**.
 - **Nutnost kompletního smazání** před dalším naprogramováním.
 - Vyžaduje **speciální programátor**.
 - **EEPROM** (*Electrically Erasable Programmable ROM*)
 - Mazání po **jednotlivých bytech**.
 - **Použití:** firmware, PLD obvody pro konfiguraci.
 - **FLASH paměť** - Energeticky mazatelná a přepisovatelná (*podobná EEPROM, ale rychlejší*).
 - **Nepotřebuje speciální programátor** – mazání a zápis přímo **elektrickým impulsem**.
 - **Použití:** USB flash disky, SSD, paměťové karty.

4. Přenos dat, rozdělení paměti a organizace

Paměť	Závislost na napájení	Možnost zápisu	Použití
SRAM	Ano	Ano	Cache paměti
DRAM	Ano	Ano	Operační paměť
NoRAM (FIFO, LIFO)	Ano	Ano	Buffery, zásobníky
ROM	Ne	Ne	Firmware, BIOS
PROM	Ne	Jednorázové	Firmware
EPROM	Ne	Opakovatelné (po UV mazání)	Specializovaná zařízení
EEPROM	Ne	Opakovatelné (elektricky)	Firmware, konfigurace
FLASH	Ne	Opakovatelné (rychlé elektrické mazání)	Úložiště dat

Statická paměťová buňka – SRAM (Static Random Access Memory)

- SRAM **uchovává informace po celou dobu**, kdy je připojena ke zdroji napájení.
- Po odpojení napájení se **obsah ztratí**.
- Na rozdíl od DRAM **nepotřebuje periodickou obnovu** (*refresh*).

Princip realizace

- Realizována jako **bistabilní klopný obvod tvořený tranzistory**:
 - **T3 a T4**: hlavní **NPN** tranzistory **uchovávající informaci**.
 - **T1 a T2**: tranzistory pro **zaadresování buňky** (*spojení s datovými vodiči*).
 - **T5 a T6**: tranzistory **představující zátěž klopného obvodu**.
- Používá **dva datové vodiče**:
 - **DATA** – **přímá hodnota** uložená v buňce.
 - **DATA neg.** – **inverzní hodnota** (*opačná hodnota vůči DATA*).
- **Adresa každé buňky je určena**:
 - **ROW select** (*výběr řádku*),
 - **COLUMN select** (*výběr sloupce*).
- **Výhody a použití SRAM**:
 - **Přístupová doba**: Velmi nízká (*1–20 ns*)
 - **Cena**: Vysoká (*drahá výroba*)
 - **Kapacita**: Menší než DRAM
 - **Rychlost**: **Vysoká** – vhodná pro cache paměti (*L1, L2, L3*) u procesorů

4. Přenos dat, rozdělení paměti a organizace

- **Stavy SRAM buňky**
 - **Klidový režim** - Buňka není zaadresována.

Popis obrázku:

- Tranzistory **T1 a T2** jsou **rozepnuté**.
- Klopný obvod stabilně **drží hodnotu** (bez zásahu).

Zápis

- **Aktivace adresy** → sepnutí tranzistorů **T1 a T2**.
- Na vodiče **DATA** a **DATA neg.** přivedena **příslušná logická hodnota (1/0)**.
- **T4 a T3** nastaví stabilní stav podle hodnoty:
 - **Zápis '1'**: DATA = 1, DATA neg. = 0 → **T4 sepnutý, T3 rozepnutý**.
 - **Zápis '0'**: DATA = 0, DATA neg. = 1 → **T3 sepnutý, T4 rozepnutý**.
- Po zápisu se adresa **deaktivuje** → buňka odpojena od datové sběrnice.

Čtení

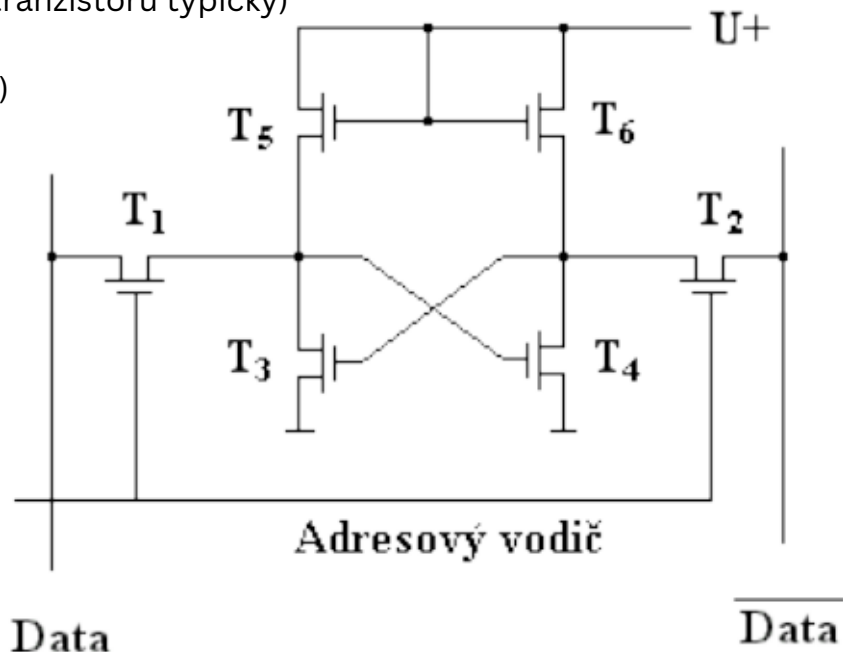
- **Aktivace adresy** → sepnutí **T1 a T2**.
- Hodnota buňky se přenesse na vodiče:
- Pokud je **T4 sepnutý** → DATA = 1, DATA neg. = 0.
- Pokud je **T3 sepnutý** → DATA = 0, DATA neg. = 1.
- Čtení probíhá přes **třístavový zesilovač** – ten zajišťuje, že při **neaktivní adrese** není sběrnice zatěžována.

Poznámky ke konstrukci SRAM buňky

- **T5 a T6**:
 - Tranzistory zajišťující zátěž bistabilního obvodu.
 - Vliv odporu těchto zátěží ovlivňuje stabilitu stavu:
 - Pokud $T5 < T6$, tak se **stabilizuje stav T4** (a opačně).
- **CHIP select (CS)**:
 - Pro zápis do **SRAM** je nutné, aby byl aktivní **signál CHIP SELECT**.
 - Jinak **není buňka dostupná** pro operaci.

Shrnutí vlastností SRAM buňky

- **Typ paměti**: Volatilní (závislá na napájení)
- **Realizace**: Bistabilní klopný obvod (6 tranzistorů typicky)
- **Obnovování**: Není potřeba
- **Přístupová doba**: Velmi nízká (1–20 ns)
- **Využití**: Cache paměti CPU (L1, L2, L3)
- **Výhody**: Rychlost, stabilita
- **Nvýhody**: Cena, větší velikost buňky, nižší kapacita



4. Přenos dat, rozdělení paměti a organizace

Dynamická paměťová buňka – DRAM (Dynamic Random Access Memory)

- **Typ paměti:** Energeticky závislá – po odpojení napájení **se data ztratí**
- **Uložení informace:** Pomocí **elektrického náboje v kondenzátoru (C1)**
- **Problém:** Náboj se **samovolně vybíjí**, i když je napájení aktivní
- **Řešení:** Nutnost **periodického obnovování náboje (refresh)**
- **Přístupová doba:** **Vyšší (cca 10–70 ns) oproti SRAM**
- **Výroba:** **Jednodušší, levnější, umožňuje větší kapacitu**
- **Použití:** Operační paměť počítačů (*RAM moduly*)

Struktura a adresování DRAM

- Paměťové buňky jsou **organizovány do matic**.
- Např.: **1024 × 1024 buněk = 1 Mbit** paměti.
- Adresa řádku a sloupce se **přenáší po stejné sběrnici** → **šetří počet vodičů**.

Signály pro adresování:

- **AS (Row Address Strobe): Výběr řádku** (aktivace řádkové adresy)
- **CAS (Column Address Strobe): Výběr sloupce** (aktivace sloupcové adresy)
 - Pro správné určení buňky musí být **aktivovány oba signály (RAS + CAS)**.

Činnost jednotlivých tranzistorů v DRAM buňce

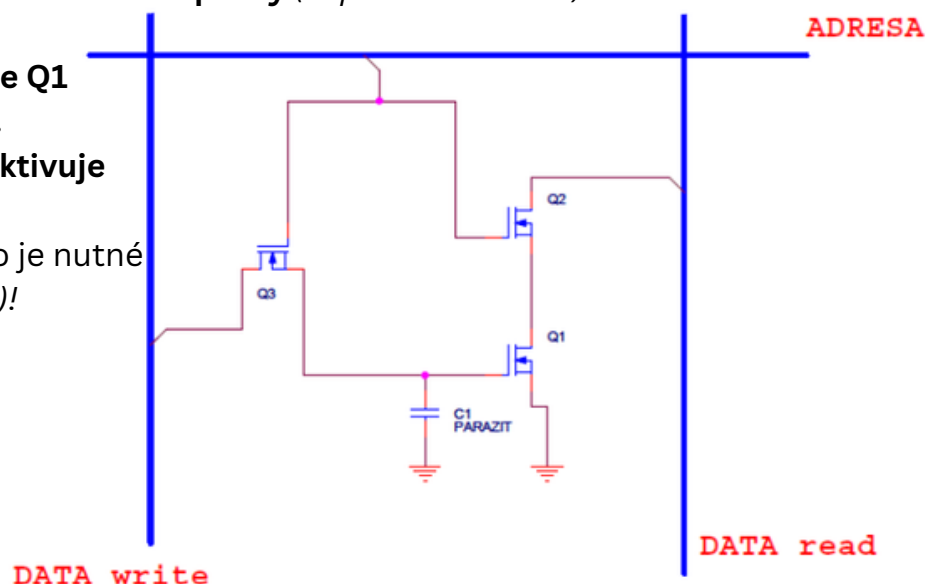
- **Q1:** Bistabilní tranzistor pro **zapamatování hodnoty**
- **Q2:** Tranzistor pro **čtení**
- **Q3:** Tranzistor pro **zápis**
- **C1:** Kondenzátor, který **uchovává náboj** (uloženou informaci)

Princip zápisu do DRAM

- **Aktivace adresy** (adresa → logická 1).
- **Sepnutí tranzistorů Q3 a Q2** (otevřou se pomocí vyššího napětí).
- Na datový vodič je přivedena hodnota:
 - Pokud **DATA write = 1**, kondenzátor C1 se nabije.
 - Pokud **DATA write = 0**, **C1** se vybije (pokud předtím nabítl).
- Po zápisu se **adresa deaktivuje**.

Princip čtení z DRAM

- Aktivace adresy (logická 1).
- **Sepnutí tranzistoru Q2 (čtení)**, **Q3 zůstává rozepnutý** (zápis není aktivní).
- Stav **C1**:
 - Pokud je **C1 nabitý** → **aktivuje se Q1**
→ na DATA read vodiči je **log. 1**.
 - Pokud je **C1 vybitý** → **Q1 se neaktivuje**
→ DATA read = 0.
- **Pozor: Čtením se C1 vybije** → proto je nutné okamžité obnovení (*refresh*)!



4. Přenos dat, rozdělení paměti a organizace

Parametry paměti

Kapacita paměti

- **Celkový objem dat**, které lze uložit do paměti.
- Počet paměťových buněk v paměti.
- **Výpočet:**
 - **Kapacita** = hloubka paměti × délka datového slova.
 - **Příklad:** 256 adres × 4 bity = **1024 bitů**.
- Udává se **v bitech** nebo **častěji v bajtech** (1 bajt = 8 bitů).

Hloubka paměti

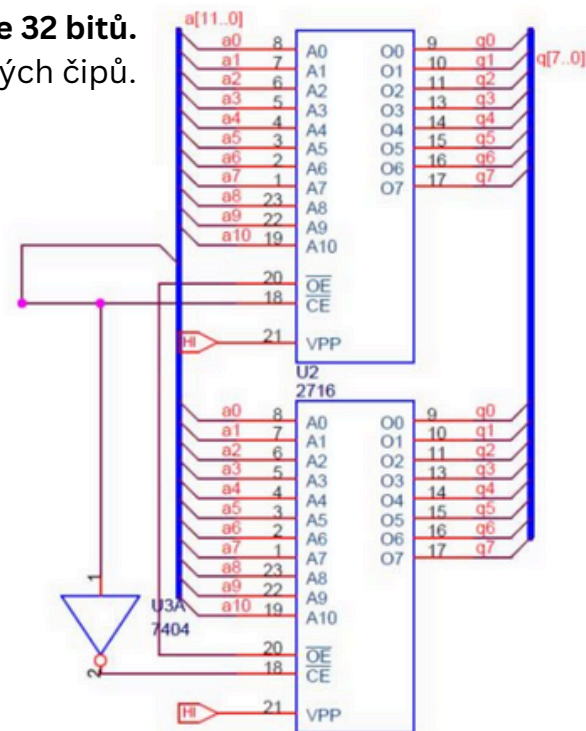
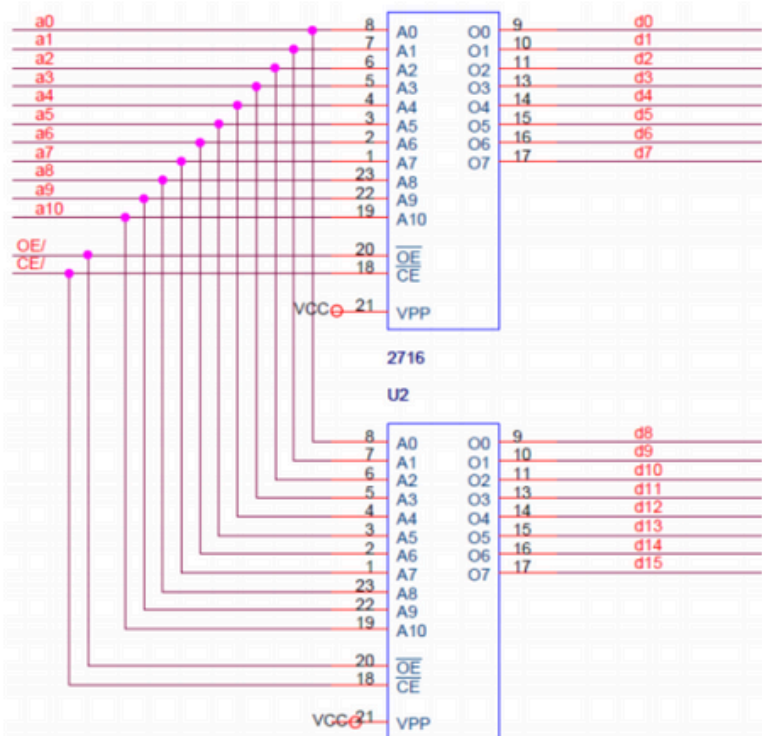
- **Počet adresovatelných paměťových míst.**
- Udává, **kolik slov paměť obsahuje.**
- Jednotka: **slovo**.
- **Možnost rozšíření** pomocí paralelního spojení více paměťových čipů.

Délka datového slova

- Počet bitů, které jsou **přeneseny nebo uloženy na jedné adrese najednou.**
- Odpovídá **šířce datové sběrnice CPU.**
- **Příklad:**
 - CPU má 32bitovou sběrnici → délka **datového slova je 32 bitů**.
- **Možnost rozšíření** pomocí sériového propojení paměťových čipů.

Shrnutí vlastností DRAM:

- **Typ paměti:** Volatilní (závislá na napájení)
- **Realizace:** Kondenzátor + tranzistor
- **Obnovování:** Nutné (periodický refresh)
- **Přístupová doba:** 10–70 ns
- **Využití:** Operační paměť (RAM)
- **Výhody:** Nízká cena, vysoká kapacita
- **Nevýhody:** Vyšší latence než SRAM, nutnost refresh



4. Přenos dat, rozdělení paměti a organizace

Organizace paměti – paměťová mapa

Paměť je organizována jako **soubor N paměťových míst** (*adresovatelných jednotek*) o šířce **n bitů** (*délka datového slova*).

- **Paměťové místo:** Nejmenší **adresovatelná jednotka v paměti**.
- **Adresa:** Binární hodnota zadaná na adresní sběrnici (*např. A0–AN*).
- Lze zapisovat i v **hexadecimálním tvaru** (*např. 0xFF*).
- **Data:** Přenášena pomocí **datové sběrnice** (*např. D0–DN*).

Zápis kapacity paměti

- **Kapacita paměti** (v bitech) se počítá jako:
 - $Kapacita = N * n$
 - **N** = počet **paměťových míst** (*slov*).
 - **n** = **délka datového slova** (*počet bitů ve slovu*).

Příklad zápisu:

- 16k(2k × 8) znamená:
 - **2k = 2048 paměťových míst** ($2^{11} = 2048$).
 - Každé místo má **8 bitů** (*1 bajt*).

Adresování paměti

- Každé paměťové místo má **svou unikátní adresu**.
- Adresa je **p-bitová**:
 - Počet potřebných adresních vodičů **p** určíme podle vztahu:
 - $p = \log N / \log 2$
 - tedy $p = \log_2(N)$.
 - **Příklad:** Pokud má paměť 2048 míst (2k), potřebujeme:
 - $p = \log_2(2048)$
 - tedy vodiče A0 až A10.

Struktura v paměti

- **Adresní sběrnice:** Zadává adresu (*výběr konkrétního paměťového místa*).
- **Datová sběrnice:** Přenáší data (*čtení/zápis dat mezi procesorem a pamětí*).
- **Paměťová buňka:** Uchovává **1 bit informace**.
- **Paměťové místo** (*slovo*): Uchovává **n bitů dat** pod jednou adresou.

Shrnutí – klíčové body

- **Hloubka paměti (N):** Počet paměťových míst (*slov*)
- **Délka datového slova (n):** Počet bitů uložených v jednom místě
- **Kapacita:** $N * n$ (*bity*)
- **Adresování:** pomocí p vodičů ($p = \log_2(N)$)
- **Maticové uspořádání:** Organizace řádků (ROW) a sloupců (COLUMN)

Paměťová mapa

