

## Hardware – vyřešené otázky k maturitě

- **Autor:** Karel Čermák, [info@k-cermak.com](mailto:info@k-cermak.com).
- **Ročník:** 2023.
- **Repozitář:** <https://github.com/K-cermak/SPSE-Maturita>.
- **Práva:** Materiály autor zveřejňuje bez záruk a pouze pro osobní nekomerční použití. Šíření těchto materiálů je povoleno pouze s původním (nezměněným) ponecháním této stránky či sdílením na oficiální repozitář uvedený výše.
- **Tip:** Pro rychlou orientaci v kapitolách lze použít klávesovou zkratku **CTRL + F** a přejít do funkce **Nadpis**.
- **Donate:** Pokud ti mé materiály pomůžou a jsi ochoten ocenit moji snahu nějakou kačkou, můžeš tak učinit přes QR kódy níže: ❤️ ❤️ ❤️
  - **Účet:** 2262692018/3030



- **Crypto:**

- **BTC:** bc1qasgxc552wjqlpcm9vt7ucmw6p4zuz007dxh8n4
- **ETH:** 0x29Ca9054B2241aB39010a1434fb50e504EE10871
- **LTC:** ltclqxpgp3jc5jyem6096n48w48qqqrwsrnj5eq9j890
- **ADA:** addr1q8c89cet02nyql4ygj96s0cz5ntusgzxfzuykfngmaf0zt2ftj7wrayqm7dx52et7k7tkjjl2edan0wykww6q4twn79shzx8vn
- **DOGE:** DCYFq9hPcVJkYKAgttkRXNSAkfbjEmLGdo



- A pokud jsi chudý student a všechny prachy prochlastáš, můžeš mi alespoň dát hvězdičku na GitHub repozitář...

# 1. Paměť a její adresace

- Reálný režim, segmenty, offset
- Chráněný režim, deskriptory, segmenty
- Globální a lokální tabulky deskriptorů
- Stránkování, TLB
- Paměťové modely nepoužívající adresaci (fronta, zásobník)

1.1. Vysvětlete, co je to paměť, data, adresa, adresovatelná paměť, paměť bez adresace.

- **Paměť** - zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat). Může se jednat o RAM či třeba Cache paměť.
- **Data** – ukládané informace (čísla, text, zvuk, obraz).
- **Adresa** – číslo (index), které označuje konkrétní paměťovou buňku.
- **Adresovatelná paměť** – data se vyhledávají podle adresy (indexu).
- **Paměť bez adresace** – data se vyhledávají například podle obsahu (klíče) – například takto funguje [HASH MAPA](#).

1.2. Vysvětlete význam kilabajtu, megabajtu, gigabajtu, terabajtu a také kibi, mebi, gibi, tebi...

- Aktuálně se používají dvě pojmenování:
  - $2^n$  pro RAM a cache označeného jako **kibi, mebi, gibi, tebi...**
  - $10^n$  pro pevné disky označeného jako **kilo, mega, giga, tera...**

<b>Ki</b>	Kibi	<b><math>2^{10}</math></b>	1024	
<b>Mi</b>	Mebi	<b><math>2^{20}</math></b>	$1024 * 1024$	= 1048576
<b>Gi</b>	Gibi	<b><math>2^{30}</math></b>	$1024 * 1024 * 1024$	= 1073741824
<b>Ti</b>	Tebi	<b><math>2^{40}</math></b>	$1024 * 1024 * 1024 * 1024$	= 1099511627776

<b>k</b>	Kilo	<b><math>10^3</math></b>	1000	
<b>M</b>	Mega	<b><math>10^6</math></b>	$1000 * 1000$	= milion
<b>G</b>	Giga	<b><math>10^9</math></b>	$1000 * 1000 * 1000$	= miliarda
<b>T</b>	Tera	<b><math>10^{12}</math></b>	$1000 * 1000 * 1000 * 1000$	= bilion

1.3. Vysvětlete, jakou adresou paměť začíná a jakou končí.

- Adresa vždycky začíná adresou **0** a končí adresou **n-1**, kdy **n** je celková velikost paměti.
- Pokud je paměť veliká **2 KB** s buňkou **1 B**, pak adresy budou v rozsahu **0 – 2047**.
- Je to stejné jako číslování v poli (**array**) v programovacích jazycích (mimo Matlab apod.).

1.4. Vysvětlete, kolik bitů je třeba k adresaci paměti o určité velikosti.

- Velikost paměti si převedu do mocniny 2 a podle odmocniny budu potřebovat tolik bitů k adresaci.
- $2^n$ , kdy **n** je počet bitů pro adresaci.
- Příklad: 2 KB paměť s buňkou 1 B =  $2^{11}$  B, k adresaci je potřeba **11 bitů**.

1.5. Vysvětlete, proč pro kapacitu pamětí je výhodné používat kilo o velikosti 1024 a nikoliv 1000.

- Pokud adresuju ve dvojkové soustavě, tak nedává smysl, abych měl méně buněk, než mohu adresovat.
- Pokud mám  $2^{11}$  B paměť (2 KB po buňce 1 B), tak adresuji 11 bitů, tedy adresy **0 - 2047**. Proč bych z paměti vzal 48 adres jen aby byla kulatá, když délka adresy se mi nezmenší a stále budu nucen použít 11 bitů?

1.6. Vysvětlete, co je to segment a jak je velký v reálném a chráněném režimu (IA-32).

- Segment – místo, kam se ukládají data (často zmiňovaný u procesorů Intel x86.).
- **Příklady:** datový segment, kódový segment, zásobníkový segment, pomocný datový segment
- **Reálný režim** (všechny Intel procesory x86)
  - Segment je velký **64 KB** (poslední adresa **FFFF**).
  - Maximální velikost paměti má **1 MB** (poslední adresa **FFFFF**).
  - Procesor je ale **16 bitový**, takže nemůžeme uložit **20 bitovou** adresu do registru.

- Proto poslední (první zprava) 4 bity segmentu jsou nastaveny na **0000**, tedy adresa segmentu je vždy xxxx**0**.
- Pozici uvnitř segmentu určuje **16 bitový offset**. Ten se zkombinuje s **počáteční adresou segmentu** a vytvoří **fyzickou adresu**, která má **20 bitů**.
- Pokud například segment začíná na adrese **1234(0)** a chceme zapsat na adresu **12** (hex) v daném segmentu, zapíšeme pak na adresu **12352** (hex).
  
- **Rozdíl mezi fyzickou a logickou adresou:**
  - **Fyzická** = adresa v paměti (20 bitů).
  - **Logická** = počáteční adresa segmentu (16 bitů) + 16 bitů offset (jsou to dvě informace, spojením těchto informací vzniká fyzická adresa, fyzickou adresu spočítá BIU).
  
- **Chráněný režim** (od Intel 80286)
  - **Intel 80286** – až 64 KB.
  - **Od dalších** – až 4 GB.

1.7. Vysvětlete, na jakých adresách smí začínat segment v reálném režimu.

- Viz předchozí otázka - na adresách xxxx**0**.

1.8. Vysvětlete, jak lze 20 bitovou počáteční adresu segmentu uložit do 16 bitového segmentového registru.

- Viz 2 otázky zpět – poslední (první zprava) **0** se vlastně nemusí ukládat.

1.9. Popište adresaci v reálném režimu metodou segment – offset.

- Viz 3 otázky zpět – zkombinuje se **počáteční adresa segmentu** s **offsetem** a vytvoří se **20 bitová adresa**.

1.10. Uvedte, jaké segmenty se používají a jaké k nim patří segmentové registry.

- **Registr CS** (Code Segment) - Kódová segment.
- **Registr SS** (Stack Segment) - Zásobníkový segment.
- **Registr DS** (Data Segment) - Datový segment.
- **Registr ES** (Extra Segment) – Pomocný datový segment.

1.11. Vysvětlete, že program a jeho data mohou být při každém novém spuštění umístěna do paměti někam jinam, a proto se používá relativní adresace – logická adresa.

- Mikroprocesor náhodně určuje to, kde vytvoří segmenty, proto programátor nemůže nastavit pevnou adresu.

1.12. Uvedte nějaké příklady – např. co udělá MOV [10],AL (pokud neuvedete něco sami, nějaký příklad vám vymyslím!).

- **MOV {kam}, {odkud}**
- V hranatých závorkách uvádíme offset datového segmentu.
- **MOV [2], AH**
  - Přesune z **registru AH** data do **datového segmentu + 2**.
- **MOV AH, [9]**
  - Přesune z **datového segmentu s offsetem 9** do **registru AH**.

1.13. Little-endian a Big-Endian - Popište, jakým způsobem se do paměti ukládají 16 bitová a 32 bitová data (jaké bajty leží na vyšší a jaké na nižší adrese).

- Ukládáme číslo 1234 5678 (hexa) na adresu 100.
- Entianita = Byte Order.

Adresa	100	101	102	103
Bajt ( <b>Big-Endian</b> )	12h	34h	56h	78h
Bajt ( <b>Little-Endian</b> )	78h	56h	34h	12h

- **Big-Endian**
  - **Nejnižší adresa - nejvíce významný bajt.**
  - **Procesory Motorola.**
  - Působí přirozeněji pokud bajty čteme jako normálně čteme čísla.
- **Little-Endian**
  - **Nejnižší adresa - nejméně významný bajt.**
  - **Intel x86.**
  - Působí přirozeněji, pokud s číslem budeme pracovat odprava (jako obvykle s binárními a hexadecimálními čísly pracujeme).

#### 1.14. Vysvětlete, k čemu je IP a SP.

- Jsou to tzv. **ukazatele**.
- **IP** – Instruction Pointer
  - Ukazuje na místo v paměti, ze kterého se právě čte strojový kód instrukce ke zpracování.
  - Obsah IP je offsetem ukazujícím do kódového segmentu.
- **SP** – Stack Pointer
  - Ukazatel vrcholu zásobníku (ukazuje adresu poslední přidané položky).
  - Pokud je prázdný, ukazuje SP na konec zásobníku (**SP = FFFF**). Tedy při přidání bajtu do zásobníku se (nepochopitelně) zásobník sníží o hodnotu 1 (**FFFA -> FFF9**), při odebrání se hodnota zvýší (**FFF9 -> FFFA**).

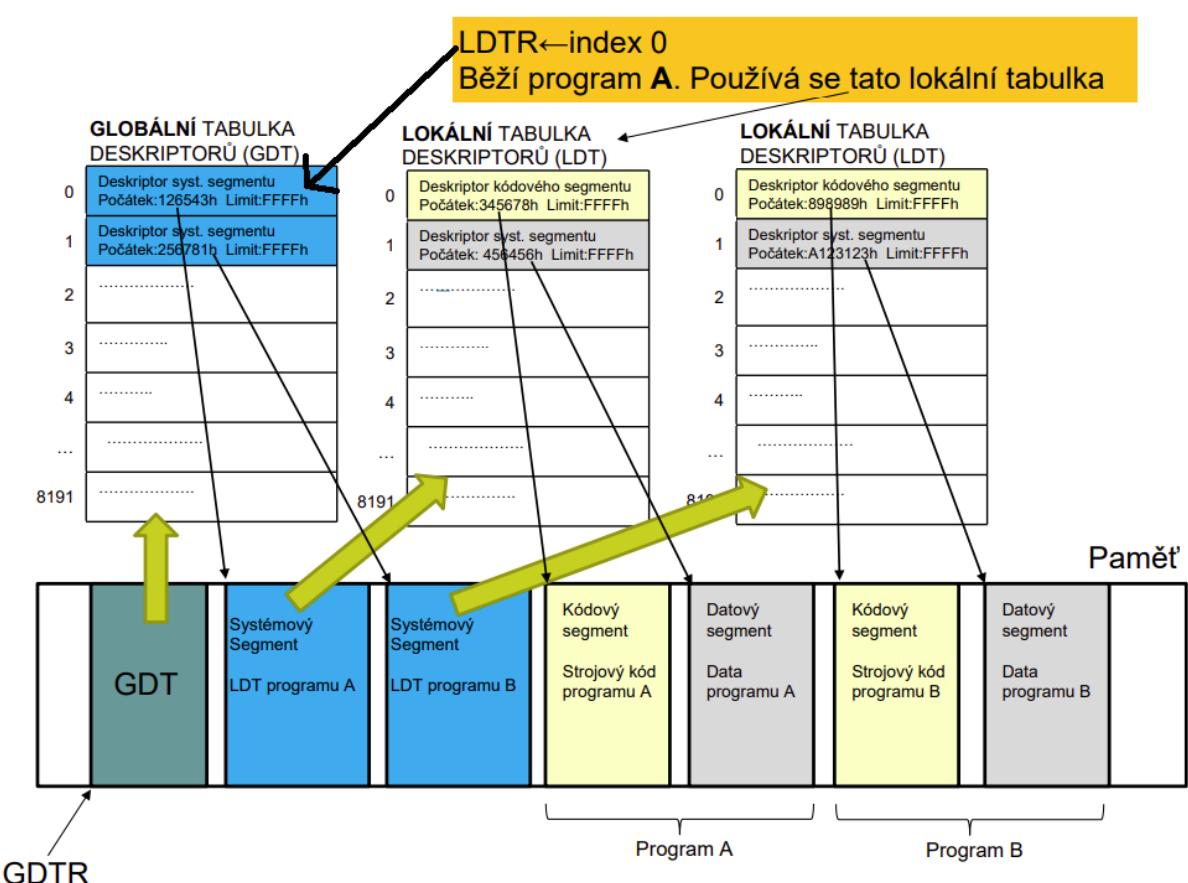
#### 1.15. Vysvětlete, proč vlastně vznikl chráněný režim.

- **Hlavní důvod – Multitasking** – současně provádění více programů.
- **Další důvody:**
  - **Vzájemná ochrana úloh** – ochrana před nežádoucím přepsáním paměťových oblastí přiřazených jednotlivým úlohám.
  - **Podpora přepínání úloh** – uložení a obnovení stavu úloh (vždy běží jedna úloha, ostatní jsou přerušené, úlohy se velmi rychle střídají).
  - **Privilegování operačního systému při provádění určitých instrukcí** – OS může vše, může přistupovat kamkoliv do paměti a ke vstupu a výstupu, běžné úlohy nikoliv.

- **Podpora pro práci s “virtuální pamětí”** – lze vytvořit a programům přidělit více segmentů, než kolik se skutečně může vejít do operační paměti.

1.16. Ukažte, jak selektor vybírá deskriptor v tabulce deskriptorů. Uvedte, co vše je uloženo v deskriptoru. Vysvětlete, co je to lokální tabulka deskriptorů, proč jich může být více, že jsou to vlastně systémové segmenty a mají svůj deskriptor v tabulce globální.

- Tabulka deskriptorů uchovává informace o tom, kde je uložen jaký kódový či datový segment.
- Začneme tedy registrem **GDTR** (ten je na 80286 24 bitový, později 32). Ten uchovává celou fyzickou adresu globální tabulky deskriptorů (**GDT**).
- V **GDT** jsou následně uloženy jednotlivé informace o jednotlivých lokálních tabulek deskriptorů (**LDT**).
- Pomocí registru **LDTR** vybereme index **LDT** (z **GDT**) se kterou chceme pracovat.
- Nyní můžeme standardně používat registry **DS**, **CS**, **ES** či **SS** a vybrat příslušný deskriptor z vybrané lokální tabulky (kterou vybral registr **LDTR** z **GPT**).



- Více lokálních tabulek deskriptorů používáme proto, že typicky každý proces využívá svoji.
- **Každá tabulka deskriptorů je systémovým segmentem!**
- **Deskriptor – záznam o segmentu!**
- Tabulka může obsahovat až **8192** položek o šířce 64 bitů (8 B), takže její velikost je až 64 kB a leží vlastně ve speciálním systémovém segmentu.
- Pokud se budeme snažit například přenášet data při vybraném kódovém segmentu, procesor to aplikaci zamítne.
- Offset má šířku na 80386 **32 bitů** (proto je maximální velikost segmentu **4 GB**).
- **V tabulce deskriptoru se nachází** (bity platné pro 80286 / 80386 s vypnutým stránkováním):
  - **Počátek segmentu** (celá adresa, nemusíme dávat žádnou **0** na konec (vpravo) jako u 8086) – 24 / 32 bitů.
  - **Limit** (maximální velikost offsetu – délka = limit + 1) – 16 / 32 bitů.
  - **Přístupová práva** – 8 bitů.

### 1.17. Vysvětlete pojmy selektor, deskriptor, segment a vzájemné vztahy.

- **Deskriptor** – popisuje, kde se nachází jaký segment, jeho limit (délku) a práva.
- **Selektor** – slouží k výběru požadovaného deskriptoru (či tabulky lokálních deskriptorů v případě registru **LDTR**). Selektor má velikost 16 b, pro index slouží 13 b ( $2^{13} = 8192$  – proto maximálně 8192 deskriptorů v tabulce). Jedná se o **DS, CS** apod.
- **Segment** – souvislý blok paměti.

### 1.18. Vysvětlete význam skryté části selektoru.

- Selektor má ještě u svých 16 b skrytých dalších 64 bitů (nejsou pro programátora dostupné).
- Při **výběru deskriptoru se zkopiuje do** těchto **64 bitů** právě **informace** z příslušného **deskriptoru**.
- Pokud se tedy selektor nemění, není nutné stále přistupovat do paměti.

1.19. Vysvětlete, proč vzniklo stránkování (hlavně netvrďte, že zrychluje přístup do paměti).

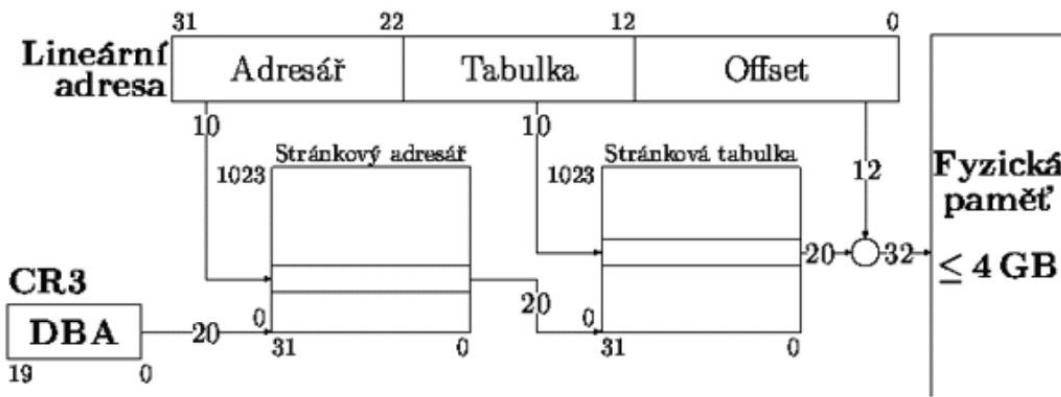
- Segmenty mohou mít velikost až 4 GB (což je opravdu hodně).
- Paměť ale můžeme být zabírána různě, **nemusí** tedy být na **vytvoření takto velkého segmentu** dostupné **ucelené místo**.
- Proto se segment rozdělí na mnoho menších tzv. stránek, která má každá velikost přesně 4 kB.
- Stránka vždy zabírá přesně 4 kB. Může jich být více a vytvoří velikost 8 kB, 12 kB, 16 kB, nikdy ne však například 10 kB!
- **STRÁNKOVÁNÍ NEZRYCHLUJE PŘÍSTUP DO PAMĚTI, NAOPAK HO CELÉ DĚLÁ DALEKO KOMPLIKOVANĚJŠÍ!**

1.20. Popište problém fragmentace paměti a obtížnost manipulace s velkými segmenty (odkládání na disk, hledání volného místa).

- Během mazání paměti se na disku vytváří mnoho malých volných míst, aby se tam dal umístit celý segment.
- Proto by nemuselo být možné, aby se nově vytvořený segment vůbec umístil do paměti.
- Je to výhodné i kvůli přenášení dat z RAM na disk – přenáší se pouze ty stránky, které nejsou používány, ne celý jeden velký segment. (Představte si, že by se uložil 2 GB segment na disk a jak dlouho by trvalo, než by byl znova celý přenesen na RAM.)

1.21. Uveďte, jak velká je stránka a jak funguje virtuální paměť při zapnutém stránkování.

- Stránka je velká **4 kB**.
- **Virtuální paměť** – programům je předstíráno, že je k dispozici větší paměť než je ve skutečnosti k dispozici.
- Nevyužívané stránky se pak odkládají na disk a pokud aplikace se stránkou chce znova začít pracovat, jednoduše je z disku zpět přenesena do RAM.
- První v hierarchii tabulek je **adresář stránek** – obsahuje informace o **tabulkách stránek**, kde už je **32 bitová informace o stránce**.
- **Tabulka stránek** je velká 4 kB a obsahuje informace o 1024 stránkách (32 bitů adresa \* 1024 = 4 kB).
- Každý proces má vlastní **adresář stránek**.



Adresář stránek obsahuje 20-bitový odkaz na tabulkou stránek

Tabulkou stránek obsahuje horních dvacet bitů z 32 počáteční adresy stránky. (spodních dvanáct bitů je vlastně offsetem do stránky)

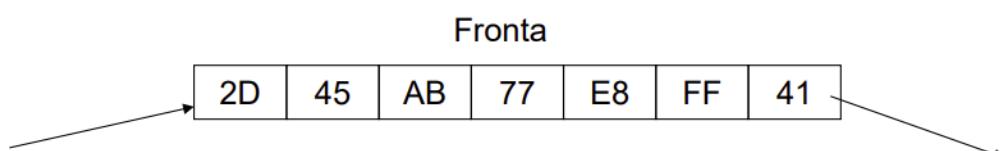
### 1.22. Vysvětlete, k čemu je TLB.

- **Translation Lookaside Buffer**
- Slouží k urychlení opakovaného přístupu k nedávno použitým stránkám (jsou zde uloženy adresy stránek, takže se znova nemusejí počítat).
- Malá paměť, funguje jako asociativní paměť ([HASH MAP](#)).

### 1.23. Ukažte, jak funguje zásobník, k čemu se dá použít, proč se mu říká LIFO a proč je to paměť bez adresace. Ukažte pak také FIFO frontu.

- **FIFO**
  - First in First out.
  - **První** uložený bajt bude přečten **první**.
  - **Fronta.**
- **LIFO**
  - Last in First out.
  - **Poslední** uložený bajt bude přečten **první**.
  - **Zásobník.**

Paměť bez adresace to je protože **nemá žádnou adresu** – prostě bud' jen data přidáme a nebo odebereme.



## 2. Polovodičové paměti

- Typické vývody, zápis a čtení, *chipselect*
- Spojení více pamětí za účelem zvýšení celkové kapacity nebo šířky slova
- SRAM, DRAM
- Paměti nezávislé na napájení - ROM, PROM, EPROM, EEPROM, Sériová EEPROM
- FLASH (SLC, MLC)
- Speciální paměti (vícebránová, NVRAM, Zálohovaná SRAM)

### 2.1. Popište tři způsoby, jak se paměti dělí.

- **Závislost na napájení**
  - **Energeticky nezávislé (neztrácí data při odpojení napájení)**
    - ROM.
    - PROM, OTP.
    - EPROM.
    - EEPROM.
    - FLASH.
    - NVRAM.
    - Zálohovaná RAM.
  - **Energetické závislé (ztrátí data při odpojení napájení)**
    - Dynamická (DRAM).
    - Statická (SRAM).
- **Možnost zápisu**
  - **ROM (Read-Only Memory)**
    - Do některých lze zapisovat, ale je to složitější než čtení.
    - Tyto paměti jsou vždy energeticky nezávislé.
  - **RWM (Read-Write Memory)**
    - Lze zapisovat stejně snadno jako číst.
    - Energeticky závislé.
    - Operační paměť je vlastně RWM, ale prakticky se označuje jako RAM.

- **Typ adresace**
  - **Adresovatelná paměť**
    - Data se vyhledávají pomocí adresy (indexu) či bloku dat.
  - **Paměť bez adresace**
    - Závisí na pořadí (fronta, zásobník).
  - **Asociativní paměť**
    - Nevyhledává se podle adresy, ale podle obsahu (klíče).

2.2. Vysvětlete, co to vlastně znamená, když je paměť závislá na napájení.

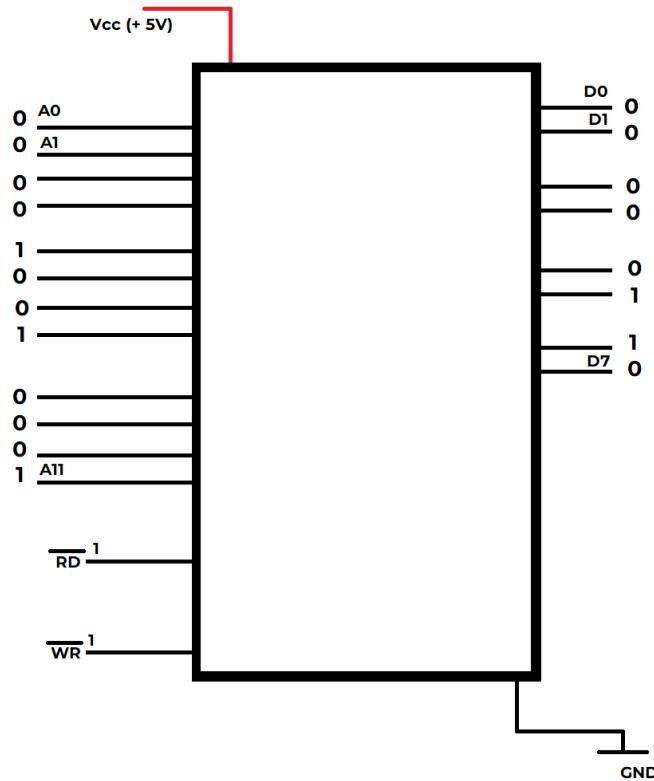
- Ztratí data při odpojení napájení.

2.3. Vysvětlete zkratky ROM, RAM, RWM.

- **ROM** – Read Only Memory
- **RAM** – Random Access Memory
- **RWM** – Read Write Memory

2.4. Nakreslete paměťový obvod a jeho typické vývody – kolik bude mít adresačních a datových vývodů.

- Navrhne si **4 kB RWM** paměť, zápis po **1 B** a provedeme zápis bajtu **60 h** na adresu **890 h**.
- $890\text{ h} = 1000\ 1001\ 0000$ .
- $60\text{ h} = 0110\ 0000$ .
- Paměť má 12 adresačních vývodů a 8 datových.
- Pozor, **RD** a **WR** je negované, takže je nastaveno na **1**. Když se signál změní na **0**, pak teprve dojde ke čtení či zápisu bajtu. **V tomto případě by tedy došlo k zápisu až v okamžiku, kdy bychom na WR nastavili hodnotu 0.**
- Bity na adresační a datové vstupy se určují tak, že **menší bit, menší index** vstupu (**0. bit na D0, 1. bit na D1**).



## 2.5. ChipSelect (vysvětlete, k čemu je dobrý).

- Značka **CS**, někdy **CE** (Chip Enabled).
- Tímto vývodem lze aktivovat / deaktivovat paměťový obvod.
- Pokud je deaktivovaný, jeho signály na vstupech a výstupech jsou ve stavu vysoké impedance (značka **Z**).
- Na tomto pinu bychom tak naměřili velmi vysoký odpor.
- Můžeme se s tím setkat například u Arduina. Ke sběrnici **SPI** může být připojeno více zařízení (například čtečka SD karty a RFID čtečka) a pomocí **CS** pinu vybereme, s jakým zařízením chceme pracovat.

## 2.6. Význam vývodů RD, WR.

- **RD** při nastavení na hodnotu 0 slouží k **přečtení** bajtu z vybrané adresy.
- **WR** při nastavení na hodnotu 0 slouží k **zápisu** nastaveného bajtu na vybranou adresu.

## 2.7. Jak se označují a jak bývají umístěny napájecí vývody.

- **Vcc** (Ucc) – 5 V
- **GND** (Ground)
- Měly by být v protějších rozích (co nejdál od sebe).

2.8. Vysvětlete, jak funguje adresový dekodér.

- **Převede vybranou adresu** (binárně z adresačních portů) **na jeden konkrétní výstup**, kterým bude **aktivována správná buňka**.
- Pokud má paměť 256 bajtů, pak má 8 vstupů a 256 výstupů.
- Pokud má paměť 1 GB, pak má 30 vstupů a milion výstupů – proto se pak používá více adresových dekodérů, kdy jeden slouží k výběru řádku a druhý k výběru sloupce.

2.9. Popište paměť DRAM a jak se tam využívá parazitní kapacita tranzistoru a proč se musí refreshovat.

- **Dynamická RAM.**
- Levná, jednoduchá struktura, používána jako RAM v PC.
- Závislá na napájení – jinak informace ztratí (tedy neuchovává data trvale).
- **Zapsaná hodnota** uchována jako náboj na **parazitní kapacitě tranzistoru MOSFET**.
- Kvůli struktuře na čipu (chová se jako kondenzátor) **dokáže udržet informaci** – tato vlastnost se nazývá **parazitní kapacita** (tuto kapacitu má každá součástka – tedy jde trochu nabít, ale to není úplně žádoucí).
- Nabitá parazitní kapacita = **1**, vybitá = **0**.
- Dochází ovšem k **samovolnému vybíjení** – tedy zapomínání.
- Proto je nutné **pravidelně** zapsané **informace obnovovat**.
- Z tohoto důvodu DRAM jsou velmi **energeticky náročné** a mohou se i zahřívat.

2.10. Popište, co se zhruba nachází v SRAM buňce – tedy bistabilní klopný obvod, složitá buňka.

- **Statická RAM.**
- Závislá na napájení – jinak informace ztratí (tedy neuchovává data trvale).
- Není třeba provádět refresh.
- Minimálně 6 tranzistorů pro jednu paměťovou buňku.
- Jiný typ tranzistorů než na DRAM.
- Základní buňku tvoří **bistabilní klopný obvod (složitá buňka)**, ale rychlejší a méně energeticky náročná.
- Má téměř nulovou parazitní kapacitu.
- Používá ji Cache.

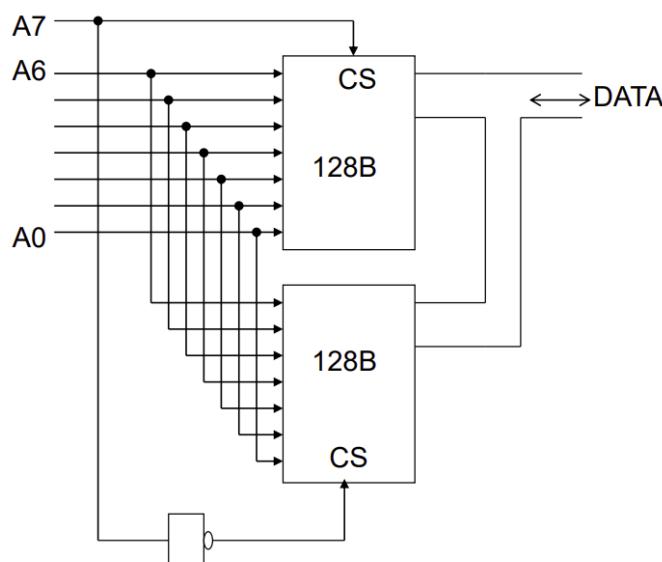
2.11. Porovnejte SRAM a DRAM paměti z hlediska složitosti paměťové buňky, ceny, nutnost refreshu, rychlosti, spotřeby elektrické energie, použití (cache, operační paměť), dosahovaná kapacita – HLAVNĚ TO NEŘEKNĚTE VŠE PŘESNĚ NAOPAK.

	<b>SRAM</b>	<b>DRAM</b>
<b>Paměťová buňka</b>	Složitá	Jednoduchá
<b>Kapacita</b>	Nízká (kB)	Vysoká (GB)
<b>Rychlosť</b>	Vysoká	Nízká (stále oproti HDD či SSD vysoká)
<b>Cena</b>	Až několik milionů Kč za GB	50 Kč / GB – DDR4 90 Kč / GB – DDR5
<b>Použití</b>	Cache	RAM
<b>Spotřeba</b>	Velmi nízká	Vysoká
<b>Refresh</b>	Není potřeba	Je nutný

2.12. Bude-li čas, mohl bych po vás chtít nakreslit spojení dvou paměťových chipů – pokud to umíte, nakreslete sami, bude vám to přičteno k dobru.

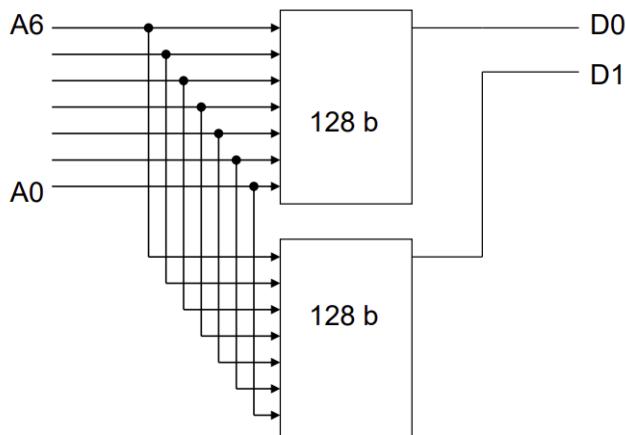
#### Dva režimy:

- **První typ (2x 128 B)**
  - V provozu je zároveň pouze jedna paměť
  - Datové cesty jsou sdílené.
  - Jeden adresační vstup (většinou poslední) se používá pro přepnutí vybrané paměti (na druhou paměť je signál znegovaný).



- **Druhý typ (2x 128 b)**

- V provozu obě paměti zároveň.
- Datové cesty nejsou sdílené.
- Jeden bit se ukládá do jedné paměti, druhý bit do druhé paměti.



2.13. Paměti nezávislé na napájení - Hlavně neříkejte, že funguje i bez napájení (bez napájení zůstane zachován uložený obsah, ale pracovat s ní nelze).

**Obsah je trvale uložen, u některých je přepisovatelný.**

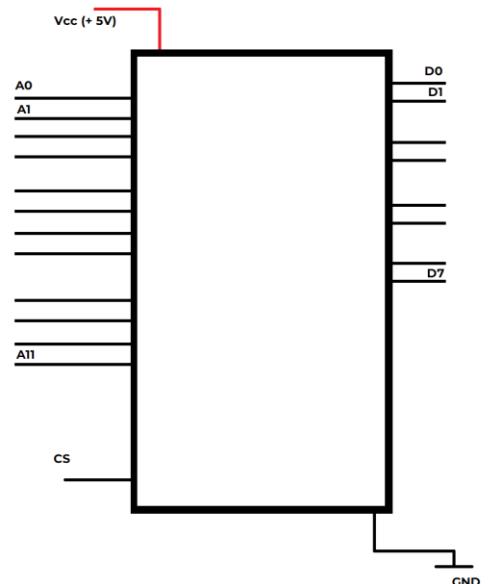
- ROM
- PROM
- EPROM
- EEPROM
- Sériová EEPROM
- FLASH

2.14. ROM – Co znamená tato zkratka, kdy se finančně vyplatí ji vyrábět, kde ji lze například dnes nalézt.

- **ROM** = Read Only Memory.
- Je to vlastně **logický kombinační obvod**, data nelze nijak už měnit, jak se uloží z výroby, tak už navždy zůstanou.
- Odolné vůči přírodním vlivům (radiace, teplota, ponoření do kapaliny..).
- Cena se odvíjí dle počtu vyrobených kusů – tedy vyplatí se například pouze pro pračky, myčky, fotoaparát (mají stejný firmware, bez možnosti aktualizace a vyrábí se jich velmi vysoké množství). Dnes ale každý chytrý spotřebič nabízí nějakou formu aktualizací, proto se tato paměť používá spíše u jednodušších zařízení.

2.15. Jaké vývody má ROM paměť s určitou kapacitou – nezapomeňte kromě datových a adresačních na CS, Ucc, GND a pozor není tady RD a WR.

- Příklad pro 4 kB ROM paměť:



2.16. PROM – Co znamená tato zkratka, proč už se nevyrábí, vysvětlit destruktivní zápis.

- **Programmable Read-Only Memory**
- Zápis je možné provést pouze jednou a poté již paměť slouží stejně jako paměť ROM.
- Programování se provádí ve speciálním přípravku (programátor PROM pamětí), je založeno na destruktivním přerušení spojů v těch místech, kde má být zapsána logická **0**, jinak v základu je všude **1**.
- **Zápis se provádí vyšším proudem, který způsobí přepálení tavné pojistky.**

2.17. EPROM – Co znamená tato zkratka. Vysvětlit, jak se maže. Jaký obsah má po smazání? Jakou má životnost?

- **Erasable Programmable Read Only Memory**
- Obsah paměti lze několikrát zapsat a smazat
- Mazání se neprovádí elektrickým signálem, ale ultrafialovým zářením v programátoru pamětí EPROM.
- Výchozí stav paměti: **Tranzistory ve všech paměťových buňkách jsou zavřené** (hodnota **0**).

- Zápis se provádí vyšším elektrickým napětím přivedeným na tranzistor, kde elektron překoná nevodivou vrstvičku, která elektrodu izoluje od substrátu a nemá již dostatečnou energii k tomu, aby přeskočil zpět.
- Tranzistor se otevře a zůstává otevřený - elektrický náboj v izolované elektrodě otevírá kanál tranzistoru, který se stává vodivým.
- Otevřený tranzistor má pak hodnotu 1.
- Pro **změnu nastavených hodnot** v paměti je nutné paměť **znova umístit do programátoru**, který paměť nastaví na nové požadované hodnoty.

## 2.18. EEPROM – Co znamená tato zkratka.

- **Electric Erasable Programmable Read Only Memory**

## 2.19. Rozdíl oproti FLASH – lze mazat a zapisovat jednotlivé bajty na náhodných adresách.

- Lze smazat jednotlivé buňky (EPROM a FLASH se maže celá).
- V Arduinu se třeba v základu ovládá přes `EEPROM.write(index, hodnota);`

## 2.20. Jaké má EEPROM vývody?

- Zapisuje se do ní úplně stejně jednoduše jako do paměti RAM – tedy není potřeba žádný **speciální programátor** či **zvýšené programovací napětí**.
- Lze přistupovat ke konkrétní adrese.
- Má ovšem omezenou životnost (1000 – 100 000 přepisů).

## 2.21. Rozdíl oproti SRAM – lze do ní úplně stejně zapisovat (má v podstatě stejné vývody), ale zápis je velmi pomalý (milisekundy) a lze provést pouze omezený počet zápisů do každé paměťové buňky. Adresa, na kterou byste neustále zapisovali “umře”. Do SRAM lze zapisovat nekonečněkrát na stejně místo, ale po odpojení napájení se obsah ztratí.

- Stejný způsob zápisu.
- **SRAM** rychlejší, **EEPROM** pomalejší.
- **SRAM** – nekonečná životnost, ale ztrácí obsah po odpojení.
- **EEPROM** – omezená životnost, ale neztratí obsah po odpojení.

2.22. Sériová EEPROM – k čemu to je dobré? Jak komunikuje s okolím? V čem se liší od všech ostatních pamětí?

- Komunikace probíhá sériově, tedy **bit po bitu**.
- Pouze 2 nebo 3 signálové vodiče.
- Počet vývodů nezávisí na kapacitě paměti.
- Nižší kapacita, malá datová propustnost.

2.23. FLASH – V jakých zařízeních jí najdeme (paměťové karty, SSD, mobilní telefony, tablety...)? Jak je organizovaná Flash s velkou kapacitou (sektory, bloky)? Jak se maže a co se stane při smazání? Běžně se používá Flash NOR nebo Flash NAND? Vysvětlete, co to znamená, že ji nelze adresovat po bajtech.

- **Použití** – USB disky, SSD, mobilní telefony, SD karty.
- **Organizace** – velké kapacity je dosaženo spojením několika tisíc malých Flash pamětí. Tyto malé paměti pak tvoří bloky (sektory). Není možné ovšem adresovat jednotlivé bajty, je potřeba zapsat / přečíst celou stránku (část bloku).
- **Mazání** – smazaná paměť ve stavu 1. Zápisem se pak nuluje bity. Zapsat 0 lze vždy, zapsat 1 nelze (je nutné paměť smazat). Reset je prováděn vyšším napětím. Čtení a zápis dat je prováděn po stránkách (nejmenší adresovatelná jednotka), mazání po blocích.
- **Životnost** – 100 000 až 1 000 000 přepisů, moderní kolem 1000 přepisů.
- **Cena** – dneska běžně kolem 1-2 Kč za GB (menší velikosti jsou dokonce levnější než HDD).
- **FLASH NOR**
  - Starší typ, moc se nepoužívá.
  - Umožňuje náhodný přístup – čtení i zápis je možné po bajtech.
- **FLASH NAND**
  - Rychlejší, vyšší hustota dat, nižší cena.
  - Neumožňuje náhodný přístup.

2.24. Jak funguje MLC a SLC buňka? (Toto je velmi důležité vysvětlit, na to se určitě budu ptát).

- **SLC** (Single Level Cell)
  - Jeden bit uložen pomocí jednoho tranzistoru – ten je buď otevřený nebo zavřený.
- **MLC** (Multi Level Cell)
  - V jedné buňce informace o dvou, třech bitech (**TLC**) anebo čtyřech (**QLC**).
  - Čtecí zesilovač dokáže rozlišit, více stavů než 2 (tranzistor zavřený, mírně otevřený, napůl otevřený, úplně otevřený) – tedy dokáže rozlišit konkrétní velikost proudu.

2.25. Jaká je cena FLASH za 1 GB ve srovnání s ostatními typy pamětí?

- **SRAM**
  - Jednotky milionů / GB
- **DRAM**
  - 50 Kč / GB – DDR4
  - 90 Kč / GB – DDR5
- **FLASH MLC**
  - 1-2 Kč / GB

2.26. Co je to vícebránová paměť a jaké má výhody?

- Jeden společný paměťový prostor, ke kterému existuje více "bran" – tedy je možné **současně adresovat** více paměťových míst, a tak **naráz číst** nebo **zapisovat na více místech v paměti**
- Nejčastěji dvoubránová paměť - do paměti může být současně zapisováno a nezávisle na tom odjinud čteno.
- Využití hlavně u grafických karet – **VRAM**.

2.27. Zálohovaná SRAM – kde ji najdeme v počítači a jak to funguje?

- K SRAM lze **připojit akumulátor**, který její obsah **udržuje i po odpojení napájení**.
- Obvykle je napětí nižší, než je napětí pracovní.
- V PC se používá pro zápis parametrů BIOSu.

### 3. RISC a CISC, pipelining

- Charakteristické rysy koncepce RISC a CISC
- Výhody RISC návrhu
- architektura moderních mikroprocesorů
- typické fáze pipeliningu
- hazardy, forwarding, load-use-delay, problém podmíněného skoku
- Výpočetní výkon počítače, oscilátor, hodinový signál, důvody nutnosti synchronizace
- Růst taktovací frekvence, příklady mikroprocesorů, zastavení růstu frekvence

#### 3.1. Začněte historií vývoje procesorů a tím, že rostl počet instrukcí v jejich instrukční sadě.

- Vývoj mikroprocesorů po roce 1980 postupně nezadržitelně vedl k jejich nadměrné složitosti.
- Okolo roku 1985 let vznikl nový směr vývoje mikroprocesorů, jehož **snahou bylo zjednodušit strukturu procesorů** a přitom zvýšit jejich výkon.
- Vzniká nová kategorie **RISC** (Reduced Instruction Set Computer) - tj. počítače (nebo spíš procesory) s redukovaným souborem instrukcí.
- Tento směr se stal populárním, ale měl i odpůrce, kteří dál prosazovali tradiční architekturu, kterou obhajovali pod označením **CISC** (Complex Instruction Set Computer) - tj. počítač se složitým souborem instrukcí.
- Podobnost můžeme vidět s programovacími jazyky:
  - **CISC** – mnoho instrukcí ([PHP](#), [Javascript](#)).
  - **RISC** – méně instrukcí, proto je jednodušší ([Arduino Wiring](#), [C](#)).

#### 3.2. Vysvětlete pojmem instrukční sada.

- Instrukční sada = **repertoár instrukcí, které dokáže daný mikroprocesor vykonávat**.
- Každý typ mikroprocesoru má jinou instrukční sadu.
- Mikroprocesory mohou být zpětně kompatibilní, pokud instrukční sada novějšího mikroprocesoru obsahuje všechny instrukce z instrukční sady nějakého staršího typu.
- Program pro daný mikroprocesor může obsahovat pouze instrukce z jeho instrukční sady.

### 3.3. Kdy se zrodila filozofie RISC a proč?

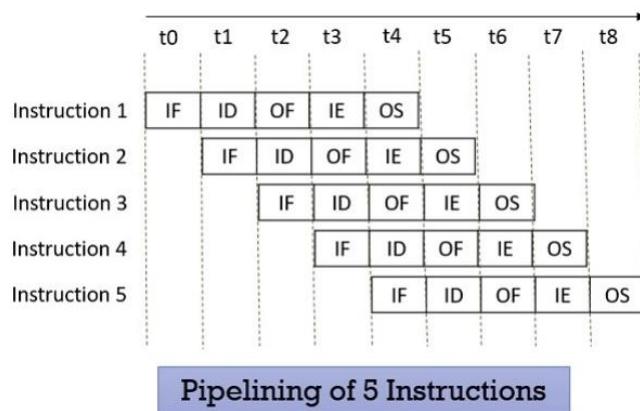
- Vývoj mikroprocesorů po roce 1980 postupně nezadržitelně vedl k jejich nadměrné složitosti, proto se v roce 1985 zrodila **filozofie RISC** a **zjednodušit počet instrukcí**.

### 3.4. Jaké jsou hlavní rysy procesorů typu RISC? (rozhodně nestačí říct, že mají redukovanou instrukční sadu – těch odlišností oproti CISC je tam celá řada (počet tranzistorů, cena, spotřeba, registry...) a výčet výhod by měl vyvrcholit pipeliningem.

	RICS	CISC
<b>Instrukční sada</b>	Malá	Velká
<b>Registry</b>	Univerzální	Mají svoje určení
<b>Počet registrů</b>	Vyšší	Nižší
<b>Cena</b>	Levnější	Dražší
<b>Doba návrhu</b>	Kratší	Delší
<b>Počet tranzistorů</b>	Malý	Velký
<b>Spotřeba</b>	Menší	Vyšší
<b>Doba trvání instrukcí</b>	Stejná	Různá
<b>Pipelining</b>	Ano	Velmi složitý, prakticky tedy ne
<b>Příklad</b>	ARM	Motorola 68000, 80386, 80486..

3.5. Vysvětlete, co je to pipeling – tady to bude chtít obrázek s překrývajícími se fázemi provádění jednotlivých instrukcí.

- Provedení každé instrukce je mikroprocesorem realizováno v několika krocích a lze ho tedy rozložit na několik standardních úkonů (fází).
- **Typické fáze provádění jedné instrukce jsou například tyto:**
  - Výběr strojového kódu instrukce z paměti (**IF** – Instruction Feed).
  - Dekódování strojového kódu (**ID** – Instruction Decode).
  - Výběr operandů (**OF** – Operands Feed).
  - provedení operace (**IE** – Instruction Execution).
  - Zápis výsledku (**OS** – Operands Store).



3.6. Ukažte typické fáze, na které se dělí provedení.

- Viz předchozí otázka.

3.7. Pozor, neříkejte, že provedení instrukce trvá jeden takt – to není pravda – provedení instrukce trvá několik taktů, ale během těch několika taktů, se začne rozpracovávat několik dalších instrukcí a v každém taktu pak může být hotova jedna instrukce.

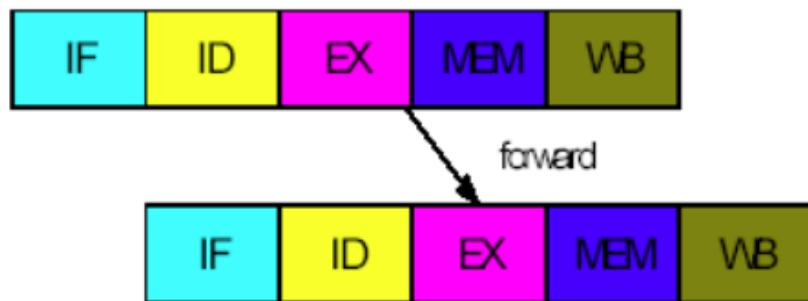
- **Provedení jedné instrukce trvá tolik taktů, kolika stupňový je pipeling.**
- V každém taktu je pak hotova jedna instrukce.

3.8. Měli byste být schopni reagovat na otázky typu: V kolikátém taktu bude dokončena 15. instrukce na mikroprocesoru s 5 stupňovým pipeliningem?

- **Pořadí instrukce + stupně pipeliningu -1** ( $15 + 5 - 1 = 19$ . takt)

3.9. Vysvětlete forwarding – připravte si nějaký příklad, kdy dvě instrukce po sobě jsou datově závislé a ukažte, jak výsledek první se neuloží do cílového registru, ze kterého by ho následující instrukce zase hned četla, ale zůstane v ALU pro provedení další operace.

- ADD R1,R5,R6      ( $R1 \leftarrow R5+R6$ )
- INC R1                ( $R1 \leftarrow R1+1$ )
- Inkrementace registru R1 by bez forwardingu mohla proběhnout až po zápisu výsledku operace ADD do registru R1.
- Díky forwardingu probíhá inkrementace ihned v návaznosti na ADD.
- Výsledek operace ADD zůstává v ALU a je ihned inkrementován.
- Výsledek operace ADD se ani nezapíše do registru R1.
- Do registru R1 bude zapsán až výsledek inkrementace, která na ADD navazuje.



3.10. Vysvětlete load-use-delay. Pozor – častá je chyba, že tvrdíte, že to je doba, během které se čeká na přečtení dat z paměti (to není úplná pravda). Je to narušení plynulého pipeliningu, když se instrukce snaží pracovat s daty ihned po instrukci LOAD, která je má přečíst. Připravte si příklad, na kterém to ukážete a můžete i hned ukázat, jak se to řeší.

- Problém, kdy se snažím pracovat s daty ihned po instrukci LOAD.
- Dá se vyřešit tak, že optimalizuju kód – tedy nebudu ihned po instrukci LOAD s registrem pracovat, ale vložím nezávislý krok mezi.
- Program:
  - $a = b + c$
  - $d = e - f$

Pomalý program:		Rychlý program:	
LW	Rb,b	LW	Rb,b
LW	Rc,c	LW	Rc,c
ADD	Ra,Rb,Rc	ADD	Ra,Rb,Rc
SW	a,Ra	SW	a,Ra
LW	Re,e	LW	Re,e
LW	Rf,f	LW	Rf,f
SUB	Rd,Re,Rf	SUB	Rd,Re,Rf
SW	d,Rd	SW	d,Rd

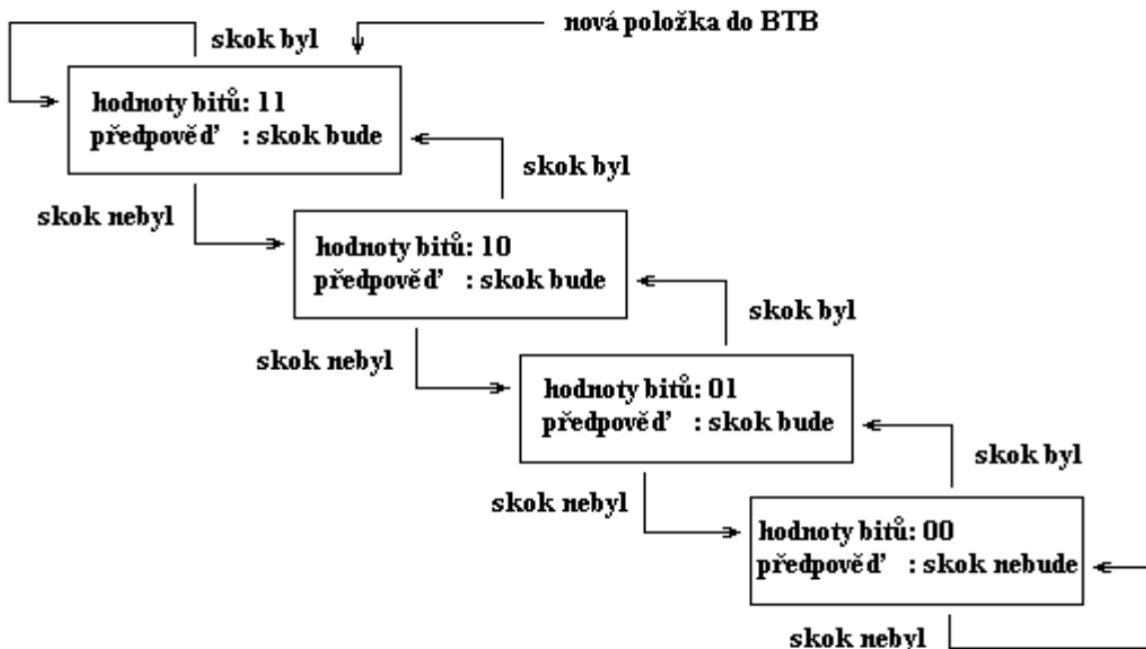
3.11. Vysvětlete vliv podmíněného skoku na pipeling – to je velmi důležité.

Jedna z nejdůležitějších částí této maturitní otázky a budu trvat na důkladném vysvětlení této problematiky.

- Podmíněný skok ve strojovém kódu vzniká například komplikací instrukcí **if**, **while**, **switch-case**....
- **Výsledek podmínky není dopředu znám**, nelze dopředu odhadnout, kterým směrem se bude program po podmíněném skoku ubírat - **pipeling se tedy zadrhne**.
- Mikroprocesor se pokusí odhadnout, kterým směrem bude pokračovat program, který se dle výsledku podmínky větví a rozpracovává instrukce, kterými bude program pravděpodobně pokračovat, ale může se zmýlit – pak se stav všech rozpracovaných instrukcí musí zahodit.
- Podmínky bychom měli psát **spíše** tak, aby se **splněna**, než nesplněna.

3.12. Vysvětlete, k čemu je dobré BTB na Pentiu. Nemusíte přesně vědět, k čemu tam jsou ty dva bity a jak přesně to funguje – hlavní je sdělit, k čemu je to dobré.

- **BTB** = Branch Target Buffer.
- Je zde uložena instrukce, která způsobila podmíněný skok a 2 bity, které odhadují další chování.



3.13. Proč je RISC vhodný k pipeliningu a CISC naprosto nevhodný – to je zásadní otázka, určitě se na ní zeptám.

- **CISC** instrukce trvají různě dlouho – pipeling by byl tedy velmi složitý.
- **RISC** instrukce trvají stejně dlouho – snadno tak lze udělat zřetězené zpracování instrukcí - pipeling.

3.14. Jsou moderní mikroprocesory pro PC RISC nebo CISC? – je třeba vědět, že navenek se předstírá velký počet instrukcí a malý počet registrů. Uvnitř se vše překládá na mikrokód – RISCVé instrukce pro RISCVé dělníky (ALU a FPU).

- Moderní CPU (Intel, ne ARM!) jsou **navenek CISC a uvnitř RISC**.
- Navenek tak mohou předstírat bohatou sadu instrukcí (**CISC**), vevnitř zase využívat efektivní **RISC pipeling**.
- Jednotlivé jednotky pracují s vlastním souborem jednoduchých instrukcí – **micro-ops**.

3.15. Proč moderní procesory předstírají navenek CISC?

- Kompatibilita s předchozími procesory.
- Velké množství instrukcí (snadnější assembler kódy).

3.16. Proč moderní procesory jsou uvnitř RISC?

- Aby šel provozovat lehce pipelining.

3.17. Proč má každý mikroprocesor nějakou taktovací frekvenci? Proč byla taktovací frekvence starých procesorů nízká a u moderních vyšší?

- Frekvence určuje, jak je dlouhý 1 takt mikroprocesoru.
- Protože logické obvody procesoru mají zpoždění, není instrukce vypracovaná okamžité. Je nutné instrukce nějak „sesynchronizovat“.
- Kdyby mikroprocesor začal provádět další instrukci dříve, než dokončí výpočet instrukci předchozí, **docházelo by k chybám** – následující instrukce by například mohla chtít pracovat s výsledkem předchozí, který se ovšem nestihl vypočítat.
- Frekvenci generuje **krystalový oscilátor**.

- Všeobecně by se dalo říct, že frekvence se **zvyšovala postupným vývojem** (auta v minulém století taktéž nedokázala dosáhnout takových rychlostí jako dnes).

3.18. Proč nemůže být taktovací frekvence procesoru libovolně vysoká? Podle čeho se nastaví základní taktovací frekvence procesoru?

- Základní taktovací frekvence procesoru je **nastavena výrobcem** tak, aby **zaručovala bezchybný chod**.
- **Celkově se problém odvíjí ode dvou aspektů – velikost (plocha), teplota a přenosové cesty:**
  - Zvýšíme-li frekvenci – procesor se bude více přehřívat.
  - Zvětšíme-li plochu – procesor se bude přehřívat méně, ale přenosové cesty budou příliš dlouhé.
  - I kdybychom brali, že signály se šíří rychlostí světla, tak v přetaktovaném procesoru na **6 GHz** urazí signál maximálně **5 cm!** ( $30\ 000\ 000\ 000 / 6\ 000\ 000\ 000$ ).
  - Celkově je to tedy takový problém, který se vlastně nedá moc vyřešit.

3.19. Jak je možné, že jdou procesory přetaktovat? Proč nejdě procesor přetaktovat na libovolnou frekvenci?

- Frekvenci můžeme zvyšovat, jak chceme, ale bude to mít za důsledek snížení, zvýšení výkonu nebo selhání procesoru.
- **Mezní frekvence** je taková frekvence, kdy perioda hodinového signálu je kratší, než zpoždění vznikající uvnitř mikroprocesoru – řadič generuje řídící signály rychleji, než jsou na ně jednotlivé funkční bloky schopny reagovat.

3.20. Jakou taktovací frekvenci měli první mikroprocesory?

- **Okolo 500 kHz.**

3.21. Jakou taktovací frekvenci mají moderní mikroprocesory?

- Dnešní okolo **4-5 GHz** (desktop), notebooky zpravidla méně (**2-3 GHz**).
- Dnešní procesory ale mívají většinou některá jádra **úsporná** (s menší frekvencí) a některé **výkonnější** (s větší frekvencí), takže nelze říct, že celý procesor má tuto frekvenci.

3.22. Jaký vliv má mnohastupňový pipelining na možnosti taktovací frekvence procesoru?

- **Čím více stupňů** má pipelining, tím vyšší může být taktovací frekvence procesoru, protože fáze, na které se rozdělilo provádění instrukce jsou **extrémně krátké**.

3.23. Proč taktovací frekvence mikroprocesorů již neroste?

- Viz 3.18.

3.24. Jakým způsobem se zvyšuje výpočetní výkon mikroprocesorů v posledních 15 letech?

- Více vláken, jader, lepší paralelizace.
- **Vyhrazené jednotky pro specifické úkony** (Neural Processing Unit v Apple Silicon ([M](#)) procesorech – procesor se nemusí na tyto úlohy ([AI](#)) vůbec zaměřovat a může využívat přímo tyto vyhrazené jednotky, které jsou navrhnuté pouze k této činnosti).
- Zvyšování **IPC** (počet instrukcí za takt).
- **DŮLEŽITÉ: VÝKON POSLEDNÍCH 15 LET NESTOUPÁ ZVYŠOVÁNÍM FREKVENCE!**

3.25. Jaký procesor překonal taktovací frekvenci 1 GHz, kdy se to povedlo.

- **AMD Athlon K75.**
- Později Intel Pentium III.
- V roce **2000**.

3.26. Mikroprocesor s vyšší taktovací frekvencí může mít mnohem nižší výpočetní výkon než procesor s nízkou taktovací frekvencí – vyjmenovat všechny důvody proč je to možné.

- Může mít více jader nebo vláken, která běží na menší frekvenci.
- Ten s menší frekvencí může mít efektivnější pipeling.
- Výkon zařízení záleží samozřejmě na daleko více aspektech – rychlosť RAM, disku, kvalitě zpracování základní desky apod.

## 4. Paralelizace

- Statická a dynamická paralelizace
- VLIW
- Superskalární procesor, IPC
- Hyperthreading, vlákna
- SIMD, MMX, SSE, pakované datové typy, výpočty se saturací
- Amdahlův zákon
- Problematika paralelního programování (vlákna, deadlock)
- Možnosti zvyšování výkonu mikroprocesorů, multicore

4.1. Uveďte příklad dvou instrukcí, které lze provést a příklad instrukcí, které nelze provést paralelně (datové závislosti).

- **Lze provést paralelně:**
  - ADD R1,R2,R3      ( $R1 = R2 + R3$ )
  - ADD R4,R5,R6      ( $R4 = R5 + R6$ )
- **Nelze provést paralelně:**
  - ADD R1,R2,R3      ( $R1 = R2 + R3$ )
  - ADD R5,R1,R4      ( $R5 = R1 + R4$ )

4.2. Vysvětlete, co je to statická a co je dynamická paralelizace.

- **Statická**
  - Pořadí a seskupení pro paralelní vykonání je **nemenné** a je **přímo zapsané** v programu.
  - Strojový kód umožňuje zakódování instrukcí, které mají být provedeny paralelně.
  - **Programátor píše paralelní program.**
  - **VLIW, SIMD.**
- **Dynamická**
  - Seskupování instrukcí probíhá za běhu programu.
  - **Programátor píše sekvenční program** a není schopen ovlivnit, které instrukce se provedou paralelně.
  - **Mikroprocesor si sám hledá** dvojice, trojice a čtverice instrukcí, které může paralelně vykonat.
  - **Dynamická paralelizace je to, co provádí superskalární procesor.**

4.3. Uveďte VLIW jako příklad statické paralelizace a popište, jak to funguje.

- **Very Long Instruction Word.**
- Jedno instrukční slovo sdružuje několik instrukcí, které mají být provedeny naráz paralelně – proto je velmi dlouhé.
- **Instrukční slovo** = slepenec instrukcí, které se provedou naráz (8 operací v každé instrukci u Texas Instruments VLIW). Někdy také slovo instrukční paket.
- Složitý strojový kód, velmi výkonný kompilátor, složité programování (**tedy je statické, protože mikroprocesor nic nehledá**).

4.4. Jako další příklad statické paralelizace uveďte SIMD – vysvětlete význam zkratky.

- **Single Instruction Multiple Data.**
- Jedna stejná instrukce provedena paralelně s více různými daty.
- **Příklad** – vynásob dvěma čísla **6, 8, 7, 20, 0, 17, 5, 3**.
- **Jedná se o statickou paralelizaci** - data s kterými má být paralelně proveden výpočet si procesor nenašel sám za běhu programu, ale připravil je v programu předem programátor.

4.5. MMX – musíte vědět, na jakém mikroprocesoru se poprvé objevuje, jaké používá registry a jak jsou velké, jaké datové typy lze zpracovávat.

- **Pentium MMX** (mezi Pentium Pro a Pentium II).
- **MultiMedia eXtension.**
- Využívá **64 bitové MMX registry** (**MM0, MM1, ... MM7**).
- Je to vlastně **SIMD** (**tedy statická paralelizace**).
- **Datové typy:**
  - **Packed byte** – 8 bajtů uvnitř 64 bitového MMX registru (**8 x 8 b**).
  - **Packed word** – 4 wordy uvnitř 64 bitového MMX registru (**4 x 16 b**).
  - **Packed dword** – 2 dwordy uvnitř 64 bitové MMX registru (**2 x 32 b**).
  - **Packed qword** – jedno velké 64 bitové číslo (**1 x 64 b**).

4.6. Ukažte nějaký příklad s pakovaným bajtem (např. instrukci PADD\$B se dvěma vymyšlenými obsahy MMX registrů) – když nepředvedete sami, zeptám se vás na to a něco vám zadám vypočítat.

<b>MM1</b>	<b>12</b>	<b>FF</b>	<b>7E</b>	<b>34</b>	<b>2D</b>	<b>BC</b>	<b>E6</b>	<b>89</b>
<b>MM2</b>	<b>9C</b>	<b>7A</b>	<b>D4</b>	<b>B9</b>	<b>FC</b>	<b>27</b>	<b>1D</b>	<b>11</b>
<b>Součet (B)</b>	AE	179	152	ED	129	E3	103	9A
<b>PADDB</b>	AE	79	52	ED	29	E3	03	9A
<b>PADD\$B</b>	AE	FF	FF	ED	FF	E3	FF	9A
<b>Součet (W)</b>	AF79		152ED		129E3		1039A	
<b>PADDW</b>	AF79		52ED		29E3		039A	
<b>PADDSW</b>	AF79		FFFF		FFFF		FFFF	

- PADD – pakované sčítání.
- S – saturace.
- B – bajt.
- W – word.

4.7. Vysvětlete, co je to saturace – opět by bylo dobré, mít na tabuli připravený nějaký vlastní příklad. Vysvětlete, proč jsou výpočty se saturací vhodné při zpracování obrazu a zvuku.

- Při **výpočtu se saturací (S) nemůže dojít k přetečení** – hodnota se zastaví na maximální nebo minimální možné hodnotě (např.: FF / 00 v případě bajtové výpočtu).
- Vhodné při sčítání barev, práci se zvukem – dioda displeje se nerozsvítí na větší hodnotu jak **255** (pro **true color** displeje, existují i 30 bitové barvy na displeji).
- Bez saturace nelze zjistit že k přetečení bajtu došlo – **CF** (Carry Flag) se sice nastaví, ale nevíme, která vypočtená část přetekla.

#### 4.8. 3DNow! – Jaké datové typy se zde používají.

- Reakce AMD na Intel MMX.
- Všechny původní instrukce MMX + instrukce na výpočet s pakovanými reálnými čísly.
- Maximálně 2 reálná čísla v pakovaném registru MMX.

1 bit	8 bitů	23 bitů
+ / -	<b>Exponent</b>	<b>Mantisa</b>

**Exponent** = celé číslo.

**Mantisa** = desetinné číslo.

#### 4.9. SSE – Kdy se poprvé objevuje, jaké registry se používají, co to umí navíc na rozdíl od MMX?

- **Streaming SIMD Extensions.**
- Poprvé na Pentium III.
- Nově až 128 bitové registry [XMM0](#) až [XMM7](#).
- Umožňuje už také výpočet s pakovanými reálnými čísly (4x 32 bitů).

#### 4.10. Jak se SIMD dál vyvíjelo SSE – SSE2, SSE3, AVX....

- **SSE2**
  - Pentium IV, 2001.
  - 16 registrů, **dvojnásobná přesnost u reálných čísel** (lze uložit jako 2 čísla o 64 bitech).
- **SSE3**
  - Pentium 4 a Pentium D, 2004.
  - Podporuje **komplexní čísla** (pro kompresi a dekompresi videa).
- **SSE4**
  - Intel Core, 2008.
  - Podporuje hashovacích funkcí, skalárních součinů (vektorů).

- **AVX**
  - Advanced Vector Extension.
  - Nástupce SSE.
  - Intel Core – Sandy Bridge.
  - 256 bitové registry [YMM0 – YMM15](#).
  - Zaměření na vektory a matice.
- **AVX-512**
  - 2013.
  - 512 bitové registry (64 bajtů).
  - Problém na Alder Lake CPU – úsporná E-cores nemají AVX-512, zatímco P-cores AVX-512 mají. Proto Intel záměrně AVX-512 vypíná na celém procesoru a necházá je spíše pro servery, kde mají podstatně větší využití.

4.11. Jako příklad dynamické paralelizace uveděte superskalární procesor – popište, jak vlastně funguje. Vysvětlete pojem IPC.

- **Superskalární procesor** - procesor, který má více než jednu frontu pro zřetězené zpracování instrukcí a více ALU, ve kterých se jednotlivé stejné fáze různých instrukcí současně provádějí.
- První superskalární procesor – **Pentium** – dvě fronty a 6 stupňový pipeling.
- Pokud procesor má taktovací frekvenci 60 MHz, díky pipelingu a tomu, že je superskalární dokáže vykonat mezi **60 až 120 miliony** instrukcí za sekundu.
- **IPC = Instructions per Clock / Cycle.**
  - Počet **vykonalých instrukcí** během **jednoho hodinového cyklu (taktu)**.
  - Pokud je **superskalární**, pak je **IPC > 1**.

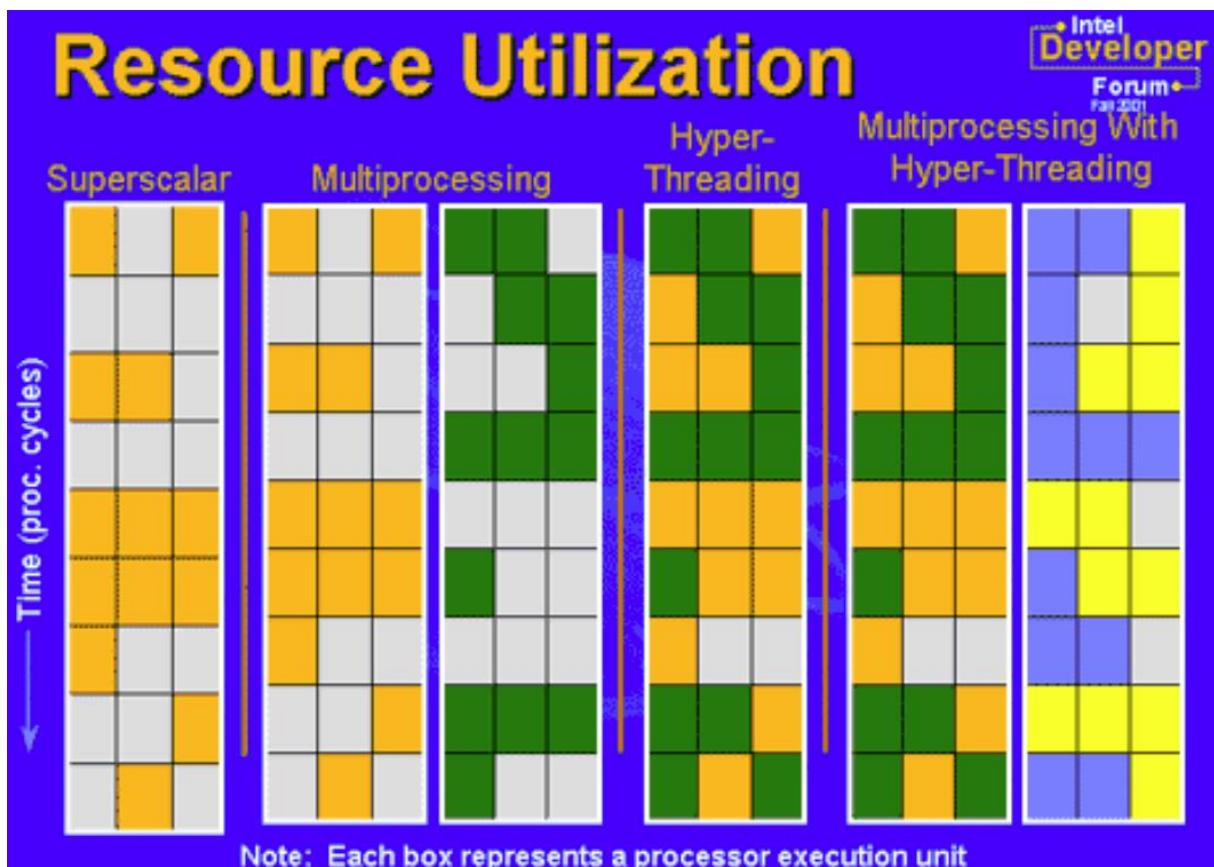
4.12. Který mikroprocesor byl první superskalární? Proč nelze v některých taktech dokončit několik instrukcí naráz?

- Intel Pentium.
- Někdy nelze dokončit více instrukcí najednou, protože mezi nimi jsou **datové závislosti** – musí se **počkat na provedení první**, až potom lze provádět delší.

4.13. Co je to hyperthreading? U jakého mikroprocesoru se poprvé objevuje?

Jak se zlepší efektivita dynamické paralelizace na mikroprocesoru s hyperthreadingem – na tuto otázku se určitě zeptám, pokud sami dopředu nevysvětlíte.

- Objevuje se na Pentium IV, 2001.
- **Hyperthreading** = Jeden procesor zpracovává dvě vlákna současně - chová se skoro jako dva procesory.
- **Thread = vláknko**; nit (posloupnost) instrukcí.
- Vlákna musí být vytvořena programátorem.
- Dalo by se to říct takto – mám 3 + 3 pipelingy – je ovšem složité najít 3 + 3 na sobě nezávislých instrukcí, které mohu vykonat současně (**viz 2. a 3. sloupeček obrázku níže**).
- Mohu ovšem nechat vykonávat ve volných místech (to kde nejsou obsazeny celé 3 pipelingy) **zcela nezávislé instrukce** (mohou třeba být definované programátorem či se jedná o zcela jiný program) (viz **4. sloupeček**).
- Tedy, místo toho, abych **potřeboval 2 jádra** (či 2 mikroprocesory), **postačí mi jeden**, který kombinuje instrukce z těchto 2. Je sice možné, že výkon nebude stejný jako u 2 oddělených procesorů (či jader), ale cenově to bude daleko levnější.



Hyperthreading je příkladem **dynamické paralelizace** – je ovšem nutné, aby byl **program napsán tak, aby mohl používat více jader**. Hyperthreading ale taktéž lze využít pro **dva zcela nezávislé procesy** (například jeden proces bude komprimovat video a druhý bude počítat složitý matematický výpočet).

Hyperthreading ovšem NELZE využít v případě, kdy je mezi procesy závislost.

4.14. Co udává Amdahlův zákon – pozor na častou chybu, kdy tvrdíte, že udává výkon mikroprocesoru – je to závislost zrychlení provádění programu na počtu paralelně pracujících jader (nebo mikroprocesorů) a parallel-portion tohoto programu.

- Závislost zrychlení výpočtu na počtu jader a Parallel portion vystihuje tzv. Amdahlův zákon.
- Je to matematický zákon, bude tedy platit vždy.
- Amdahlův zákon říká, kolikrát se teoreticky může maximálně zrychlit provedení programu – ve skutečnosti může být zrychlení nižší (nikdy ne vyšší).
- **Příklad: Úloha má trvat 200 jednotek času. Lze paralelizovat 80 % jednotek času a k paralelizaci můžeme využít 8 jader.**
  - Víme, že paralelizovat půjde 80 % z 200 – tedy **160 jednotek času**.
  - **40 jednotek času** tedy **nepůjde paralelizovat**.
  - 160 jednotek času si rozdělí mezi sebou 8 jader – každé svojí práci dokáže udělat za **20 jednotek času** ( $160 / 8 = 20$ ).
  - Úloha bude tedy trvat celkem **40 + 20 = 60 jednotek času**. (40 nešlo paralelizovat a 20 bylo paralelizováno 8 jednotkami).

4.15. Vysvětlete, jak lze program rozdělit na vlákna – uveděte nějaký příklad (příklad na řazení prvků pole, příklad s kódováním videa, příklad s hledáním klíče hrubou silou....).

- Jsou to části programu, které jsou na sobě **nezávislé**.

4.16. Uveděte příklady programů, které lze snadno dělit na vlákna a programů, které nelze dělit na vlákna.

- **Lze:**

- **Kódování videa** – [vlákno 1](#) zpracuje prvních 5 minut, [vlákno 2](#) posledních 5 minut.
- **Hledání klíče hrubou silou** – [vlákno 1](#) zkouší všechny hesla začínající na A, [vlákno 2](#) všechny hesla začínající na B.
- **Mining kryptoměn** (tedy jestli se ještě něco těží na CPU) – obě vlákna generují zcela nezávisle na sobě náhodné [Nonce](#), dokud nenarazí na [Golden Nonce](#).

- **Nelze:**

- **Rendering webové stránky v PHP** (pokud neprobíhá rendering pro více uživatelů).
- **Matematické výpočty** – třeba výpočet Pythagorova čísla na X desetinných míst.
- **Různé složité výpočty** – simulace různých fyzikálních jevů apod.

4.17. Vysvětlete, proč je složité paralelní programování a jako příklad uveděte deadlock – pokud budete uvádět příklad s večeřícími filozofy, tak ho přirovnejte ke počítačovým pojmem. Lepší je možná uvést příklad s dvěma procesy, které se blokují při přístupu ke stejnemu souboru a tiskárně.

- **Složité z mnoha důvodů:**

- Vymyslet zcela nový paralelní algoritmus - to je to úplně nejtěžší a ty nejdokonalejší paralelní algoritmy chápe jen pár lidí na celém světě (např. paralelní násobení matic nebo řazení polí s logaritmickou složitostí).
- Jak rozdělit program na nezávislá vlákna a jaký bude optimální počet vláken.
- Synchronizace vláken (některé akce prováděné v jednom vlákně lze provádět až po dokončení akcí v jiném vlákně).
- Sdílení společných prostředků (soubory, paměť, vstupy a výstupy).
- Slučování mezivýsledků jednotlivých vláken do finálního výsledku.
- Zamezení deadlocků.

- **Deadlock**

- **Vlákno A** má otevřený soubor a chce tisknout.
- **Vlákno B** je připojené k tiskárně a chce tisknout ze souboru, který je ale otevřený **vláknem A**.
- **Vlákno A** tedy nezavře soubor, dokud ho nevytiskne.
- **Vlákno B** neuvolní tiskárnu, dokud nezíská přístup k souboru.

4.18. Jakým způsobem roste výkon procesorů za posledních deset let?

- Více vláken, jader, lepší paralelizace.
- **Vyhrazené jednotky pro specifické úkony** (Neural Processing Unit v Apple Silicon ([M](#)) procesorech – procesor se nemusí na tyto úlohy ([AI](#)) vůbec zaměřovat a může využívat přímo tyto vyhrazené jednotky, které jsou navrhnuté pouze k této činnosti).
- Zvyšování **IPC** (počet instrukcí za takt).
- **DŮLEŽITÉ: VÝKON POSLEDNÍCH 10 LET NESTOUPÁ ZVYŠOVÁNÍM FREKVENCE!**

4.19. Vysvětlete zákon marginálních zisků.

- Celkový výkon počítače není dán jen výkonem mikroprocesoru, ale závisí také na kapacitě paměti, rychlosti paměti, rychlosti sběrnice, rychlosti pevného disku atd.
- Pokud budeme zrychlovat pouze jeden z parametrů, např. rychlosť paměti, poroste výkon celého počítače zpočátku rychle, ale od určité hranice další zvyšování rychlosť paměti nepřinese žádný účinek.

4.20. Jaký má každý další přidaný tranzistor vliv na příkon procesoru a jaký na výkon?

- Například [Apple M1 Ultra](#) má 114 miliard tranzistorů. Z tohoto lze usoudit, že každý přidaný tranzistor má menší vliv na celkový výkon.
- Například aby byl výpočetní výkon procesoru dvojnásobný, potřebujeme osmkrát složitější mikroprocesor.

4.21. Jakým způsobem lze nejsnadněji zdvojnásobit počet tranzistorů v mikroprocesoru? Zdvojnásobí se tím i výkon a proč?

- Zdvojnásobíme počet jader. Spíše ne, záleží na paralell portion.

## 5. První mikroprocesory řady x86

- Základní pojmy mikroprocesorové techniky (*registrový kód, bit, bajt, data, šířka dat, architektura počítače*)
- Základní vlastnosti prvních mikroprocesorů řady x86, *registry, režimy, přerušení a jeho obsluha, porty, DMA*

5.1. Vysvětlete, co je to počítač a co všechno lze považovat za počítač.

- Počítač je zařízení, které podle předem připraveného programu zpracovává data.
- Počítač je ale obecně jakékoliv zařízení, které pracuje s daty podle zadaného programu (tedy i mobilní telefon, televize, chytrý kávovar či kalkulačka).
- Základní dva typy: **Analogový počítač** (velmi úzce zaměřený) a **Číslicový počítač**.

5.2. Řekněte něco stručně k pojmul instrukční sada a strojový kód a vysvětlete, co dělá komplilátor a co je to vyšší programovací jazyk a v čem se liší od assembleru – pokud o tom nic neřeknete sami, tak se vás na strojový kód určitě zeptám. Můžete zmínit také jak se liší instrukční sada RISC a CISC procesorů.

- **Instrukční sada** = povely / instrukce, které umí procesor vykonávat.
  - **RISC** – menší množství jednodušších instrukcí.
  - **CISC** – rozmanitější sada složitějších instrukcí.
- **Strojový kód** = binárně zapsané instrukce, které procesor přímo vykoná.
- **Programovací jazyk** = prostředek mezi programátorem a počítačem.
  - Vyšší programovací jazyk – vyšší míra abstrakce ([Python](#), [Java](#), [C++](#), [C](#), [Rust](#)).
  - Nižší programovací jazyk – malá nebo žádná abstrakce od toho, jak funguje procesor ([Assembly](#)) – závislé na platformě.
- **Kompilátor**
  - Překládá program zapsaný v programovacím jazyce do strojového kódu, který je pak v procesoru spuštěn (i když toto je dost zjednodušené vysvětlení, ono to je o dost složitější).

5.3. Vysvětlete pojmem bit a bajt, popište číselné označení bitů v bajtu a pojmy MSB, LSB.

- **Bit** – hodnota **0 / 1**, je to nejnižší datová jednotka.
- **Bajt** – 8 bitů (hexadecimálně hodnota až **FF**).
- **MSB** – Most Significant Bit – Nejvyšší bit – vlevo.
- **LSB** – Least Significant Bit – Nejnižší bit – vpravo.

**Bit se značí odprava nultým.**

5.4. Vysvětlete, co je to registr, co je to střadač a co jsou to univerzální registry. Kdy se data ukládají do paměti a kdy do registrů.

- Velmi malá, ale velmi rychlá paměť nacházející se v přímo v mikroprocesoru.
- **Střadačová architektura**
  - Instrukce lze provádět pouze s některými **konkrétními registry**.
  - **Střadač** – lze s ním provádět **všechny operace** (většinou označení jako A, ACC, AX apod.).
  - **Výsledky** se většinou **ukládají** přímo **do střadače**.
  - Intel 8086.
- **Univerzální registry**
  - Procesor obsahuje několik registrů, které mají naprostotu **rovnoprávné postavení**.
  - **Neobsahuje střadač**, výsledky lze ukládat kamkoliv.
  - Typické pro **RISC**.

5.5. Co to znamená, když se řekne 8 bitový, 16 bitový, 32 bitový mikroprocesor?

- Šířka registru je dána počtem bitů procesoru (16 bitový procesor = 16 bitové registry).
- Taktéž 16 bitový procesor bude mít 16 bitovou ALU.
- 16 bitový procesor pracuje s 16 bitovými daty, ale adresaci může mít větší.

5.6. Můžete ukázat, že 8 bitový procesor umí sčítat 16 bitová čísla, ale trvá mu déle, protože to dělá po částech – tím se dá zabít docela dost času, když předvedete, jak takové výpočty probíhají.

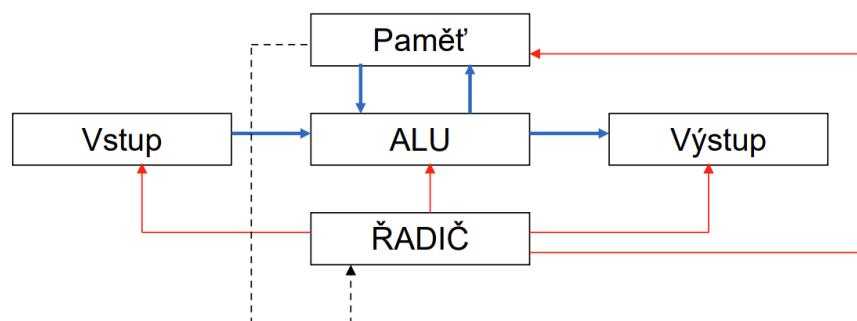
- Bude nutné rozdělit 16 bitové čísla do čtyř 8 bitových.
- Bude to trvat ale o dost déle, navíc je nutné brát v potaz možnost přetečení dat.
  - **MOV R0,1Ah** // do jednoho registru umístíme horních osm bitů prvního sčítance
  - **MOV R1,57h** // do tohoto registru umístíme spodních osm bitů prvního sčítance
  - **MOV R2,23h** // do tohoto registru umístíme horních osm bitů druhého sčítance
  - **MOV R3,62h** // do tohoto registru umístíme spodních osm bitů druhého sčítance
  - **ADD R1,R3** // sčítáme spodních osm bitů
  - **ADD R0,R2** // a nakonec horních osm bitů obou sčítanců

5.7. Nakreslete Von Neumannovu nebo harvardskou architekturu – stačí, když si vyberete jednu, druhou po vás nebudu chtít kreslit, ale zeptám se vás, v čem se liší. Pozor, obrázek je k ničemu, když v něm nebudou tři různé druhy šípek znázorňující tok dat, řídící signály a tok povelů ve strojovém kódu.

- **Von Neumannova**

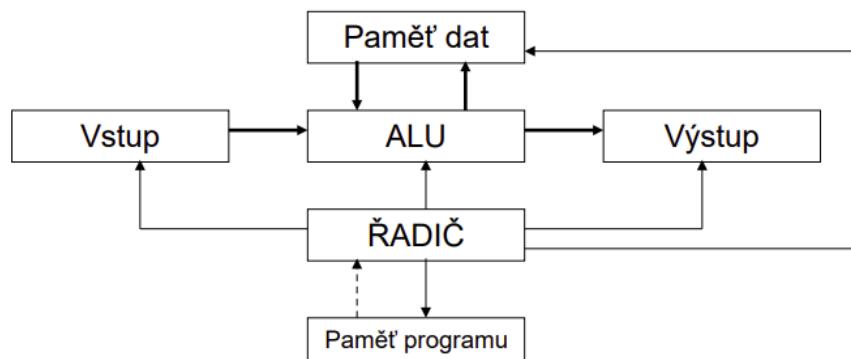
- Jedná se o myšlenkovou mapu, jak by počítač měl být rozdělen.
- **Řadič** – ovládá, co mají všechny součásti dělat, získává strojový kód ze vstupu.
- **ALU** – všechno počítá, získává data ze vstupu, komunikuje s pamětí a posílá na výstup.

—→ Řídící signál  
 —→ Tok dat  
 - - - - → Čtení strojového kódu



- **Harvardská**

- Hlavní rozdíl oproti Von Neumannově architektuře spočívá v **oddelené paměti programu a paměti dat**.
- Z paměti programu lze pouze číst strojový kód. Počítač sám neumí do této paměti zapisovat. Program tak nemůže přepsat sám sebe. Počítač nelze "zavirovat". Naprogramovat počítač lze pouze zvenku.



5.8. Bylo dobré vědět, co je ALU a řadič a jak z toho později vznikl mikroprocesor.

- **ALU**

- Aritmeticko-Logická Jednotka.
- "Dělník", který **vykonává operace s daty**.
- Dokáže provádět veškeré operace s bajty (sčítání, násobení, porovnání..).

- **Řadič**

- Řídí veškerý chod celého počítače.
- Generuje řídící signály pro ostatní bloky počítače.
- Rozumí strojovému kódu, kterým jsou zakódované povely tvořící program.

**Řadič a ALU později splynuly v jedno – vznikl mikroprocesor.**

5.9. Určitě se vás zeptám na výhody a nevýhody obou architektur.

- **Von Neumannova architektura**

- **Výhody**

- Struktura je nezávislá od zpracovávaných problémů, na řešení problému se musí zvenčí zavést návod na zpracování, program a musí se uložit do paměti, bez tohoto programu není stroj schopen práce.
    - Lze přepsat program (aktualizace apod.).
    - Lze zavádět programy z disku – můžete si spustit co chcete.

- **Nevýhody**

- Žádný paralelismus.
    - Snaha o maximální univerzálnost a neměnnou vnitřní strukturu - počítač, který je univerzální, dokáže sice (téměř) všechno, ale občas neefektivně.

- **Harvardská**

- **Výhody**

- Nemůže přepsat sám sebe.
    - Ideální pro řídící jednotky, automatizaci apod.

- **Nevýhody**

- Neexistuje možnost aktualizací či spuštění čehokoliv, co chci.

5.10. Pamatujte, že PC má von Neumannovu architekturu a musíte vědět, proč je tato architektura pro PC výhodnější než harvardská!

- Data počítače může sami přepsat (aktualizace apod.).
- Můžeme také spustit jakýkoliv program, programovat na něm apod.

5.11. Stručně popište nultou až čtvrtou generaci počítačů – žádné dlouhé povídání, pokud o tom budete mluvit moc podrobně a ztrácet čas, tak vás přeruším. Stačí uvést fakta – jaké součástky se používaly, v čem byla výhoda oproti předchozí generaci. Není potřeba uvádět názvy a vlastnosti pravěkých počítačů.

Nejprve byly k dispozici různé mechanické krabičky, hudební hrací skřínky apod.

- **0. generace** – 1936 - 1945
  - Elektromechanické prvky – **relé**.
  - Je to strašně pomalé (jako fakt strašně) – hlavně kvůli relé, ale hustě to cvaká ([https://youtu.be/yid\\_RYGp4x0](https://youtu.be/yid_RYGp4x0)).
  - Obrovské velikosti – několik místnosti.
- **1. generace** – 1945 - 1956
  - Používání **Elektronek** místo relé.
  - **Tištěné spoje** místo drátových spojů.
  - **Příkon desítky kW** (to je taky fakt hodně).
  - Elektronka se zahřívá a je energeticky náročná, navíc často poruchová (poruchy i několikrát za hodinu).
- **2. generace** – 1956 - 1964
  - Používají se **tranzistory**.
  - Programování ve Fortranu.
  - První OS, vznik přerušení.
- **3. generace**
  - **Integrované obvody**.
  - Zmenšování, nízká spotřeba, vysoká spolehlivost.
- **3½. generace**
  - **Magnetické disky, OS**.
  - Tisícové série, přenositelné.
- **4. generace**
  - **IBM PC/XT – Intel 8086**.
  - MS-DOS.
  - Běžné domácnosti, cena nízká, výkon roste.
  - Zvyšování kompatibility různých programů.

5.12. Uveďte, kdy vznikly první vyšší programovací jazyky a kdy se začalo používat přerušení.

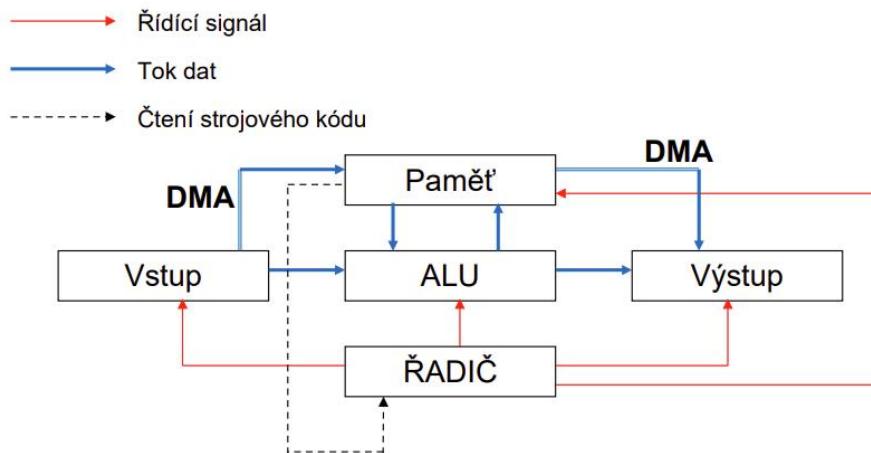
- **Přerušení**
  - **1956.**
    - I/O zařízení potřebuje mít k dispozici vhodný mechanismus, který umožní přihlásit se k procesoru a vynutit si jeho pozornost.
    - Procesor na přerušení zareaguje a začne obsluhovat dané zařízení. Nemusí tak neustále kontrolovat, zda je zařízení například připraveno k odesílání dat.
- **Vyšší programovací jazyky**
  - **1957.**
    - **Vyšší programovací jazyk** – programovací jazyk, který je nezávislý na konkrétním instrukčním souboru procesoru.
    - Od programátora neočekává znalost konkrétního počítače, pro který bude program psát.
    - První [FORTRAN](#), následně [COBOL](#) a [ALGOL](#).

5.13. Vysvětlete, k čemu je dobré přerušení – uveděte příklad s mikroprocesorem a připojenými I/O periferiemi. Co jsou to vektory přerušení a jak se provádí obsluha přerušení v chráněném režimu?

- Viz předchozí otázka.
- Pro každé přerušení je v paměti uložen **přerušovací vektor** (tj. **odkaz na adresu**, na které se nalézá jeho **obsluha přerušení**).
- Prvních 1024 bajtů paměti na Intelu 8086 je vyhrazeno pro 256 vektorů přerušení.
- V chráněném režimu se použije číslo přerušení jako index do **tabulky IDT**.
- **Položkami tabulky IDT** jsou **deskriptory** s informací o tom, kde leží v paměti pro dané **přerušení jeho obsluha**.

## 5.14. Vysvětlete k čemu je dobré DMA.

- **Direct Memory Access**
- Možnost přenosu dat mezi I/O zařízením a pamětí **bez účasti procesoru**.
- Procesor jen naváže přenos a dále se o přenos stará už jen **DMA řadič**.
- Nelze využít mezi dvěma I/O zařízeními – vždy jen přístup zařízení do paměti.



## 5.15. Uveďte základní parametry mikroprocesoru 8086 (kdy vznikl, kolikabitová data zpracovává, kolik paměti lze adresovat, kolik obsahuje tranzistorů, jak fungují jednotky EU a BIU, v jakém je pouzdře). Musíte vědět a vysvětlit, co to znamená „šestnáctibitový procesor s dvacetibitovou adresací“.

- **16 bitový mikroprocesor.**
- 1978.
- Maximálně 1 MB paměti (**20 bitová adresace**).
- První z rodiny x86.
- 5-10 MHz, střadačová architektura, 14 registrů.
- Pouzdro **DIL40**.
- **29 000 tranzistorů.**
- **EU (Execution Unit)**
  - **Obsahuje ALU.**
  - Zpracovává **operandy, ovládá registry, příznakové bity**.
  - **Nemá přístup k okolnímu světu.**

- **BIU (Bus Interface Unit)**
  - Obstarává **komunikaci mikroprocesoru s okolím**.
  - Čtení **strojového kódu z paměti**, čtení dat a zápis výsledků.
  - 6 B **fronta FIFO** pro bajty strojového kódu.
- **8086 pracuje s 16 bit daty, ale používá 20 bit adresaci.**
- **Adresy v paměti jsou o velikosti 20 bitů** (jsou složené z 16 bitového offsetu a z 16 bitové adresy počátku segmentu).
- **EU (ALU)** pracuje ovšem **16 bitově**, 20 bitové adresy pak generuje **BIU**.

5.16. Uveďte také, že tento prastarý mikroprocesor je praotcem rodiny mikroprocesorů rodiny x86 a že i dnešní moderní mikroprocesory umí zpracovávat programy psané pro tento procesor a mají z něj odvozený kompatibilní strojový kód a registry.

- Všechny mikroprocesory řady x86 se po zapnutí nebo po resetu dostanou do reálného režimu.
- Tedy i dnešní nejmodernější mikroprocesory se po svém zapnutí chovají jako prastarý i8086, dokud není zaveden operační systém, který je přepne do jiného složitějšího a výkonnějšího režimu.
- Starý strojový kód a operace, které používají staré registry, budou stále fungovat bez problému na nových procesorech x86, protože nový procesor stále obsahuje instrukce 8086 (myslím, že plná podpora je ale až od programů psaných pro 80386).
- Zamyšlení: Nemůže toto být důvod, proč Intel poslední dobou zaostává před Apple Silicon čipy či ARM čipy?

5.17. Vysvětlete, jak funguje AX ve vztahu k AH, AL a jaké speciální registry jsou k dispozici (IP, SP, FLAGS...).

- **AX je střadač.**

AX (16 bitů)															
AH (horních 8 bitů)								AL (dolních 8 bitů)							
0	1	0	1	1	0	1	0	0	1	1	1	1	0	0	0

$$\mathbf{AX} = 0101\ 1010\ 0111\ 1000 \text{ b}$$

$$\mathbf{AH} = 0101\ 1010 \text{ b}$$

$$\mathbf{AL} = 0111\ 1000 \text{ b}$$

- **Segmentové registry** (obsahuje adresu začátku segmentu bez poslední hexadecimální číslice):
  - CS, SS, DS, ES
- **Datové registry** – 16 bitové (případně 8 b (vyšší H) + 8 b (nižší L)) – slouží pro aritmetické a logické operace:
  - **AX** (AH + AL)
  - **BX** (BH + BL)
  - **CX** (CH + CL)
  - **DX** (DH + DL)
- **Ukazatelé a indexové registry:**
  - **IP** – Instruction Pointer – ukazuje na místo v paměti, kde se právě čte strojový kód.
  - **SP** – Stack Pointer - Ukazatel na vrchol zásobníku.
  - **BP, DI, SI**
- **Příznakové registry:**
  - **CF** – Carry Flag – Příznak přetečení.
  - **PF** – Parity Flag – nastaví se na 1, pokud dolní osmice bitů výsledku obsahuje sudý počet 1.
  - **ZF** – Zero Flag – nastaven při nulovém výsledku.
  - **SF** – Sign Flag – záporný výsledek.

5.18. Uveďte základní parametry procesoru 80286 – kdy vznikl, kolik bitů, jak velká paměť se dá obsluhovat, má stejné registry jako předchůdce, jaké taktovací frekvence...

- **1982.**
- 130 000 tranzistorů.
- Frekvence **6-12 MHz**, 2.5x výkonnější než 8086.
- Podporuje multitasking.
- 24 bitové adresy (stále ale 16 bitový – registry má taktéž stejné!) – umí tedy **adresovat 16 MB**, ale pouze v novém **chráněném režimu**, jinak 1 MB.
- **4 jednotky** – **BU** (Bus Unit), **UI** (Instruction Unit), **EU** (Execution Unit), **AU** (Address Unit).

5.19. Zmiňte, že hlavní novinkou je chráněný režim, ale nebudeme se zabývat tím, jak funguje. Řekněte pouze, k čemu je dobrý.

- **Hlavní důvod – Multitasking** – současné provádění více programů.
- **Další důvody:**
  - **Vzájemná ochrana úloh** – ochrana před nežádoucím přepsáním paměťových oblastí přiřazených jednotlivým úlohám.
  - **Podpora přepínání úloh** – uložení a obnovení stavu úloh (vždy běží jedna úloha, ostatní jsou přerušené, úlohy se velmi rychle střídají).
  - **Privilegování operačního systému při provádění určitých instrukcí** – OS může vše, může přistupovat kamkoliv do paměti a ke vstupu a výstupu, běžné úlohy nikoliv.

5.20. Mikroprocesor 80386 – Základ architektury IA32 – Kdy vznikl, kolik bitů, jak velká může být paměť v chráněném a reálném režimu.

- **1985.**
- 275 000 tranzistorů.
- 32 bitový (adresy i data).
- Maximálně 4 GB paměti (což je na svoji dobu fakt hodně), v reálném režimu ale jen **1 MB**.
- Novinka – **stránkování**.

5.21. Nové 32 bitové registry EAX, EBX... Vztah EAX k AX a AH a AL.

- **EAX = 1234 5678 h**
- **AX = 0101 0110 0111 1000 = 5678 h**
- **AH = 0101 0110 = 56 h**
- **AL = 0111 000 = 78 h**

EAX (32 b)																															
																AX (16 b)															
								AH (8 b)								AL (8 b)															
0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0

5.22. K čemu je V86 režim.

- Režim **virtuální 8086**.
- V operační paměti je vytvořen prostor 1 MB, ve kterém je adresováno pomocí segmentu a ofsetu stejně, jako je tomu v reálném režimu.
- V této oblasti paměti pak mohou být spuštěny programy, které byly napsány pro **reálný režim** (tedy bez selektorů, deskriptorů atd.).
- Vývoj SW nešel tak rychle, a proto migrace na chráněný režim zaostávala.

5.23. Zmiňte zavedení stránkování a k čemu je dobré a že je tu Paging unit, ale o detailech stránkování se bavit nebudeme (to patří do jiné otázky).

- **Paging Unit** - stránkovací jednotka, pracuje pouze pokud je aktivní stránkování a slouží k převodu lineární adresy na fyzickou.
- Segmenty mohou mít velikost až 4 GB (což je opravdu hodně).
- Paměť ale může být zabírána různě, **nemusí** tedy být na **vytvoření takto velkého segmentu** dostupné **ucelené místo**.
- Proto se segment rozdělí na mnoho menších tzv. stránek, která má každá velikost přesně 4 kB.
- Stránka vždy zabírá přesně 4 kB. Může jich být více a vytvoří velikost 8 kB, 12 kB, 16 kB, nikdy ne však například 10 kB!
- **STRÁNKOVÁNÍ NEZRYCHLUJE PŘÍSTUP DO PAMĚTI, NAOPAK HO CELÉ DĚLÁ DALEKO KOMPLIKOVANĚJŠÍ!**

5.24. Mikroprocesor 80486 – poslední nesuperskalární procesor, objevuje se FPU a Cache.

- **1989**
- Poslední **nesuperskalární procesor** – neumí více instrukcí zároveň.
- 32 bitové adresy i datová sběrnice (je to významně vylepšená 80386).
- Obsahuje nově integrovaný **matematický koprocesor (FPU)** a **cache**.

5.25. Vysvětlete pojem FPU, koprocesor.

- **Floating point unit.**
- Jednotka pro **výpočty s reálnými čísly** (čísla se znaménkem, desetinnou čárkou a exponentem).
- Označován také jako matematický koprocesor.

5.26. K čemu byly chipy 8087, 80287, 80387.

- Jednalo se o **přídavné matematické koprocesory** ([FPU](#)) pro 8086, 80286 a 80386.
- Standardní 80486 už má [FPU](#) integrovaný přímo v sobě.

5.27. Popište, jak je organizovaná první cache použitá u tohoto procesoru. Proč cache neměla smysl u předchozích procesorů? Popište varianty 80486SX, 80486DX, 80487, 80486DX2, 80486DX4.

- **Rychlosti procesoru** se počínaje typem 80386 dále **zvyšují mnohem rychleji než rychlosť paměti** (RAM). Integrovanou cache má až 80486.
- Procesor by **uměl vykonat mnoho instrukcí** pro čtení nebo zápis do paměti, ale paměť nestihá a trvá jí příliš dlouho než vybaví nebo zapíše data.
- **Cache na 80486:**
  - Čtyřcestná, 128 tříd.
  - Jeden záznam blok má šířku 16 bajtů.
  - Celková kapacita  $128 \times 4 \times 16 \text{ B} = \mathbf{8 kB}$ .
  - Je to vlastně asociativní paměť ([HASH MAP](#)).
- **80486 DX**
  - Plnohodnotná verze s integrovanou [FPU](#).
- **80486 SX**
  - Čip bez funkční [FPU](#) (je deaktivováno).
- **80487 SX**
  - Pro retardy, co si koupí [80486 SX](#), začnou brečet, že nemají [FPU](#), tak si dokoupí tento rozšiřující modul, který je prakticky stejný jako [80486DX](#) (má ale jiné vývody).
  - Ten převezme činnost [80486 SX](#) a prakticky [80486 SX](#) umrtví.
- **80486 DX2**
  - Procesor se zdvojenou vnitřní hodinové frekvenci – výkon stoupne asi o 50 %.
- **80486 DX4**
  - Trojnásobná vnitřní hodinová frekvence.
  - Větší cache.
  - Vychází v době Pentia, ale není superskalární.

## 6. Superskalární a vícejádrové mikroprocesory (IA-32, x86-64)

- Základní vlastnosti (frekvence, výkon, možnosti, vnitřní struktura, cache) a historické souvislosti
- Multicore procesory
- Fáze Tick-Tock ve vývoji procesorů
- AMD Ryzen
- Vysvětlení souvisejících pojmů (BTB, Inkluzivní a exkluzivní cache, PCU, spekulativní provádění, NetBurst, Intel Core, FPU, TurboBoost, Systém Agent, Media Engine)
- IA-32, x86-64
- Dennardovo škálování, Koomeyho zákon, dark-silicone a leakage current problém

6.1. Pentium – Nic nekreslit, k vnitřní struktuře stačí uvést frontu U, V – dvě ALU pracující paralelně a vedle nich ještě FPU.

- **1992.**
- Stále je vše 32 bitové, jen šířka datové sběrnice 64 bitů.
- Dvě fronty označované jako U, V – každá 6 stupňový pipelining.
- Daleko vyšší výkon než 80486 při stejné frekvenci.
- Dvě ALU (32 bitové) a FPU (64 b / 80 b (reálné čísla)).

6.2. Uvést, že složité instrukce se překládají na jednodušší RISCové.

- Pentium kombinuje filosofii RISC i CISC.
- Složité CISC instrukce jsou rozloženy na mikroprogram popsaný mikrokódem.
- Každý mikropovel tohoto mikroprogramu pak lze vykonat v 6 krocích.
- Jednoduché instrukce se vykonávají bez překladu na mikroprogram – přímo hardwarovými prostředky.

6.3. První superskalární procesor – co to znamená.

- Během jednoho taktu může dokončit až dvě instrukce zároveň (tedy více než 1).
- Má více ALU, hledá vhodné operace, které může provést zároveň.

6.4. Jaké byly taktovací frekvence? Jak se měl původně jmenovat a proč se tak nejmenoval. Jak se zde snížilo napájecí napětí a proč?

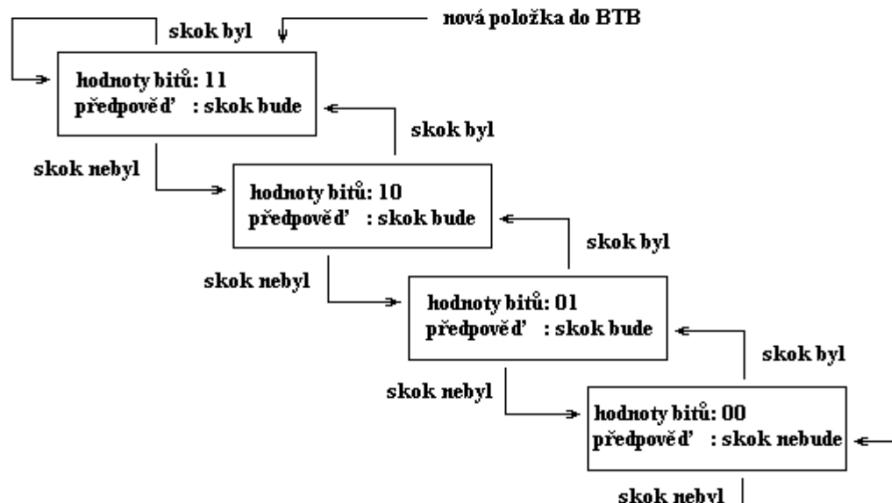
- **První generace – 60 – 66 MHz**, napětí 5 V, výkon 13 W.
- **Druhá generace – 75 – 200 MHz**, napětí 3.3 V, výkon 5 W.
- Původně se mělo **jmenovat 80586**, ale nakonec se ujalo označení Pentium (označení 80586 mělo problémy s licencemi).

6.5. Vysvětlete závislost příkonu a zahřívání procesoru na napájecím napětí.

- **Závislost výkonu na napětí je kvadratická!**
- Jestliže se **napětí zvýší na dvojnásobek, výkon se zvýší 4x**.
- Jestliže napětí klesne na polovinu, sníží se výkon 4x a tím pádem i 4x klesne zahřívání procesoru.
  - $P = \text{příkon} [W]$ .
  - $U = \text{napětí} [V]$ .
  - $I = \text{el. proud} [A]$ .
  - $R = \text{odpor} [\Omega]$ .
- $P = U \cdot I$
- $P = \frac{U}{R}$
- $P = U \cdot \frac{U}{R} = \frac{U^2}{R}$

6.6. K čemu slouží BTB?

- **BTB** = Branch Target Buffer.
- Je zde uložena instrukce, která způsobila podmíněný skok a 2 bity, které odhadují další chování.



6.7. Pentium PRO – Objevuje se L2 cache (vysvětlit rozdíl mezi L1 a L2 – cena, rychlosť, velikosť).

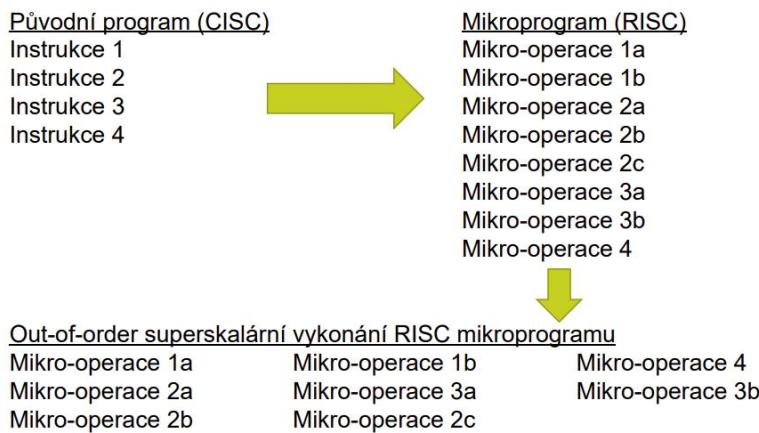
	L1 Cache	L2 Cache
Rychlosť	Větší	Menší
Velikosť	Menší	Větší
Cena	Větší	Menší
Velikosť (Pentium)	8 kB	-
Velikosť (Pentium PRO)	16 kB	256 kB (512 kB, 1 MB)

6.8. Popište CISC-RISC architekturu. Vysvětlete, proč je uvnitř RISC a navenek CISC.

- **RISC** - Reduced Instruction Set Computer – dostupné menší množství jednoduchých instrukcí.
- **CISC** - Complex Instruction Set Computer – dostupné větší množství složitějších instrukcí.
- Dnes se používají procesory, které se navenek tváří jako CISC (poskytují velkou řadu složitých instrukcí, které jsou pak uvnitř dekódovány na jednoduché RISC - microops), ale uvnitř jsou RISC - aby bylo možné provozovat pipelining, protože RISC instrukce lze všechny rozdělit na stejný počet podoperací o stejně době trvání.

6.9. Vysvětlit spekulativní provádění instrukcí a také pojem out-of-order.

- **Spekulativní provádění instrukcí**
  - Procesor si spekulativně provádí některé instrukce předem (a zpětně může některé výsledky odvolut)
  - V případě větvení lze spekulativně provádět nezávislé instrukce obou větví (jako by podmínka, dle které se program rozvětví byla splněna i nesplněna) a po zjištění výsledku podmínky skoku zrušit výpočty neplatné větve.
  - Tato vlastnost se Intelu v roce 2018 šerdeně vymstila – chyby [Meltdown](#) a [Spectre](#) umožňovaly využívat tuto vlastnost, a tak uživatel mohl získat data, která pro něj standardně nebyla přístupná. Nakonec bylo nutné CPU speciálně záplatovat, kdy se znatelně snížil jejich výkon.
- **Out-of-order**
  - Instrukce mohou být prováděny i v jiném pořadí, než ve kterém byly umístěny do instrukční fronty a v jakém je zapsal programátor.



6.10. Pentium MMX – stručně vysvětlit, co je to MMX, SIMD, pakované datové typy, saturace (detailně patří do jiné maturitní otázky).

- **Pentium MMX** (mezi Pentium Pro a Pentium II).
- **MultiMedia eXtension**.
- Využívá **64 bitové MMX registry** (MM0, MM1, ... MM7).
- Je to vlastně **SIMD (tedy statická paralelizace)**.
- **Datové typy**:
  - **Packed byte** – 8 bajtů uvnitř 64 bitového MMX registru (**8 x 8 b**).
  - **Packed word** – 4 wordy uvnitř 64 bitového MMX registru (**4 x 16 b**).
  - **Packed dword** – 2 dwordy uvnitř 64 bitové MMX registru (**2 x 32 b**).
  - **Packed qword** – jedno velké 64 bitové číslo (**1 x 64 b**).
- Při **výpočtu se saturací (S) nemůže dojít k přetečení** – hodnota se zastaví na maximální nebo minimální možné hodnotě (např.: FF / 00 v případě bajtové výpočtu).
- Vhodné při sčítání barev, práci se zvukem – dioda displeje se nerozsvítí na větší hodnotu jak **255** (pro **true color** displeje, existují i 30 bitové barvy na displeji).

6.11. Pentium 2 – nemusíte znát všechny jednotlivé varianty. Stačí přibližné časové zařazení, rozsah taktovacích frekvencí a hlavně, že je to v podstatě pořad Pentium PRO.

- **1997.**
- Vychází z Pentia PRO.
- Zvětšená cache, jiná patice.
- **Frekvence od 233 do 450 MHz.**
- **12 stupňový pipeling**.

6.12. Pentium 3 – není považováno za další samostatnou generaci procesorů Intel. Pořád vychází z Pentia PRO. Zde je zajímavá taktovací frekvence, která překročila 1 GHz a SSE.

- **1999.**
- Není samostatná generace, stále je to 6. generace x86 (pod Pentium PRO).
- **10stupňový pipeling.**
- Pentium III Coppermine – v roce 2000 překonává hranici 1 GHz.
- Přidáno SSE (Streaming SIMD Extensions) – nástupce MMX.

6.13. Nezapomeňte zmínit, že u těchto procesorů se objevují varianty Celeron. První Celerony se lišily tím, že nemají L2 cache. Čím se pak liší další generace Celeronů?

- Celerony mají sníženou rychlosť, menší velikost cache ([L2](#) nemá vůbec).
- Je to skvělý poměr pro ty, co nechtějí drahý procesor a nepotřebují mnoho výkonu a řeší poměr výkon / cena.
- Celerony D (na Pentium IV) mají až 40 stupňový pipeling, ale vlastně je to celé omyl (nesmyslné navyšování pipelingu) a nemá to moc smysl, proto je někdy Celeron D zařazován mezi nejhorší procesory všech dob.

6.14. Pentium 4 – v jakých letech se vyrábí, jaké taktovací frekvence dosahuje?

- **7. generace**, uveden **2000**, poslední verze **2007**.
- Architektura Netburst, SSE2.
- Frekvence od 1.4 GHz do 4 GHz (u novějších generacích) – ale už je to velmi problematické.

6.15. Naprosto nezbytné je tady vědět: kolikastupňový pipelining se tu používá a zda to je výhodné?

- NetBurst architektura používá 20 stupňový pipeling, což je extrém a slepá cesta.
- Rozdělit jednoduché instrukce na 20 stupňů je skoro nemožné a můžou nastávat různé problémy.
- Dalo by se říct, že to je pokus Intelu zvyšovat nesmyslně frekvenci procesorů na úkor použitelnosti.

6.16. V čem je výhodný mnohastupňový pipelining? V čem je naopak nevýhodný? – to musíte vědět, je to mnohem důležitější než parametry všech procesorů patřící do této otázky.

- **Výhoda** – frekvence procesoru může být vysoká.
- **Nevýhoda** – viz předchozí bod.

6.17. Co je to performance rating, jak AMD označuje své tehdejší procesory a proč?

- Je to marketingový tah AMD, které má výkonnější procesory jako Intel, ale ne s tak vysokou frekvencí.
- Nově tedy nikde neudávají frekvenci, ale tzv. Performance Rating – **kolik by mělo Pentium IV se stejným výkonem frekvenci**.
- Například, AMD Athlon XP 3000+ má frekvenci jen 2000 MHz, pro stejný výkon by ale Pentium IV potřebovalo frekvenci 3000 MHz.

6.18. Pentium D – stačí vědět, že existovalo a že je to “další” Pentium IV.

- Nástupce Pentia IV, stále architektura NetBurst.
- Dvě jádra, základní frekvence 3.5 GHz.
- Vydáno v roce 2007.

6.19. Napište si během přípravy na tabuli, který procesor je pátá, šestá, sedmá a osmá generace procesorů Intel.

- **1. generace** – 8086
- **2. generace** – 80286
- **3. generace** – 80386
- **4. generace** – 80486
- **5. generace** – Pentium
- **6. generace** – Pentium PRO
  - Pentium MMX, Pentium II, Pentium III, Pentium M
- **7. generace** – Pentium IV
- **8. generace** – Intel Core

## 6.20. Proč nelze dále zvyšovat taktovací frekvenci procesorů.

- **Celkově se problém odvíjí ode dvou aspektů – velikost (plocha), teplota a přenosové cesty:**
  - Zvýšíme-li frekvenci – procesor se bude více přehřívat.
  - Zvětšíme-li plochu – procesor se bude přehřívat méně, ale přenosové cesty budou příliš dlouhé.
  - I kdybychom brali, že signály se šíří rychlostí světla, tak v přetaktovaném procesoru na **6 GHz** urazí signál maximálně **5 cm!** ( $30\,000\,000\,000 / 6\,000\,000\,000$ ).
  - Celkově je to tedy takový problém, který se vlastně nedá moc vyřešit.
- Taktéž protože logické obvody procesoru mají zpoždění, není instrukce vypracovaná okamžité. Je nutné instrukce nějak „sesynchronizovat“.
- Kdyby mikroprocesor začal provádět další instrukci dříve, než dokončí výpočet instrukci předchozí, **docházelo by k chybám** – následující instrukce by například mohla chtít pracovat s výsledkem předchozí, který se ovšem nestihl vypočítat.

## 6.21. Jaký příkon zhruba má Pentium 4 a Pentium D?

- **Pentium 4** – většinou kolem těch 60 – 90 W (záleží na verzi).
- **Pentium D** – 140 W.

## 6.22. Intel Core – proč tato architektura nenavázala na Netburst?

- Myšlenka NetBurst je mít procesor s co nejvyšší frekvencí (plán byl až 10 GHz, porazit konkurenci, šlo o marketingový trik)
- To se ale ukázalo jako neštastné (problém energetické náročnosti, architektura se ukázala jako dlouhodobě neudržitelná).

## 6.23. Co je to Multicore? Jakým způsobem lze dále zvyšovat výkon procesorů?

- **Multicore = procesor s více jádry.**
- Jedná se vlastně o paralelní zapojení několika mikroprocesorů, přičemž jsou umístěny na jednom jediném čipu a tváří se jako jeden celistvý procesor.
- Výkon lze dále zvyšovat přidáváním dalších jader.

6.24. Fáze TICK – TOCK – Musíte vědět, co se ve které fázi mění a co zůstává – hlavně to neprohodit. Ke každé fázi musíte znát i nějaký konkrétní procesor.

- Vývojový model, který Intel uplatňuje (uplatňoval) pro své procesory.
- Střídání miniaturizace výrobního procesu a uvádění nových architektur po jednom roce.
- **TICK - zmenšení tranzistorů** (Westmere).
- **TOCK - uvedení nové architektury**, které využívají zmenšených tranzistorů. (Nehalem).

6.25. Jaký smysl mají označení i3, i5, i7?

- **i9**
  - Nejvyšší výkon, obrovská spotřeba, nejvyšší cena.
  - Výpočetní stanice.
- **i7**
  - Vysoký výkon, větší spotřeba, vysoká cena.
  - Herní a pracovní stanice.
- **i5**
  - Kompromis mezi **i7** a **i3**.
  - Méně složité úkony, základní hry.
- **i3**
  - Nízká cena, nízká spotřeba, nejnižší výkon.
  - Kancelářské počítače, notebooky.

Důvod je především v **rychlém určení využití daného procesoru** (víme, že pokud chceme výkonnou herní stanicí, asi nebude koukat po **i3**).

6.26. Nehalem – stačí vědět, co nového přináší – hlavně L3 cache, PCU a turbo-boost.

- **L3 cache**
  - 8 MB, sdílená mezi všechna jádra.
- **PCU**
  - **Power Control Unit.**
  - Je schopen velice sofistikovaně řídit dle potřeby napájecí napětí a frekvenci jednotlivých jader.
- **Turbo-boost**
  - **Turbo mode nebo Turbo boost.**
  - Pokud není zpracovávána vícevláknová aplikace, některá jádra vícejádrového procesoru jsou nezatížená.
  - Nezatížená jádra může **PCU** zcela vypnout.
  - Elektrický příkon procesoru s vypnutými jádry zdaleka nedosahuje hodnoty **TDP**.
  - Zapnutá jádra je tak možné přepnout do “Turbo režimu” – zvýšit frekvenci jejich běhu nad nominální frekvencí procesoru, a přesto se procesor nebude přehřívat.

6.27. Vysvětlete Inkluzivní L3-cache.

- **Inkluzivní**
  - Intel.
  - Obsah L1 je obsažen v L2 cache a L2 je obsažen v L3.
- **Exkluzivní**
  - AMD.
  - Data nejsou v cache uložena duplicitně a celkem tak lze cachovat více dat.
  - Pokud nejsou data nalezena ve velké sdílené cache, pak to ještě neznamená, že v některé z lokálních cache jiných sousedních jader se hledaná data nenachází. Takže je třeba všechny je projít, což stojí čas navíc.

## 6.28. Vysvětlete TurboBoost a TurboBoost 2.0.

- **TurboBoost** - Viz dva body zpět.
- **TurboBoost 2.0** - umožní jednomu "studenému" jádru zvýšit na několik sekund svůj výkon až nad úroveň maximální povolené TDP - teplo se "rozloží v čase".

## 6.29. U dalších procesorů je důležité zmínit integraci GPU, MediaEngine, severního můstku a řadiče PCI-E do mikroprocesoru a přibližně jejich hlavní rysy a pamatovat názvy architektur, jak šly po sobě až po současnost.

- **Intel Core** - 2006.
- **Penryn** – 2007 - **TICK**.
- **Nehalem** – 2007 - L3 cache, PCB, turbo-boost - **TOCK**.
- **Westmere** – 2008 – nově i3 - **TICK**.
- **SandyBridge** – 2010 – integrované GPU, Media Engine, AVX, System Agent (Severní můstek přímo v CPU – PCU, komunikace s PCI Express, RAM) - **TOCK**.
- **IvyBridge** – 2012 - **TICK**.
- **Haswell** – 2013 – podpora **SoC** pro mobilní HW, úsporné varianty **S, T** a výkonné **K**, celkově vylepšení vlastností - **TOCK**.
- **Broadwell** – 2014 – pseudocache L4 (je to eDRAM – DRAM v CPU) - **TICK**.
- **Skylake** – 2015 – i9 - **TOCK**.
- **Kaby-Lake** – 2017 – 4k video - **OPTIMALIZACE**.
- **Coffee-Lake** – 2018 – některé šesti jádrové - **DALŠÍ OPTIMALIZACE**.
- **Cannon-Lake** – 2019 – AVX512 - **TICK**.
- **Ice-Lake** – 2020.
- **Rocket-Lake** – 2020.
- **Tiger-Lake** – 2020 – Deep Learning Boost, GNA jádra pro strojové učení.
- **Alder-Lake** – 2021 – dvě jádra – výkonná (**P-cores**) a efektivní jádra (**E-cores**), DDR5 - **TOCK**.
- **Raptor-Lake** – 2022 – menší optimalizace.

6.30. Popište stručně moderní generace Alder-Lake, Raptor-Lake a uveděte pojmy DL-Boost, GNA, Intel Thread Director.

- **Deep Learning Boost** – pro AI.
- **GNA jádra** - pro strojové učení a neuronové sítě.
- **Intel Thread Director** - monitoruje vše, co se v procesoru děje. Má přehled o právě prováděných instrukcích, budoucích načtených instrukcích, taktech, teplotách, spotřebě a dalších stavech procesoru, které v reálném čase každou nanosekundu. Efektivní pouze pro Windows 11.

6.31. Popište, jak se zdokonalovala výrobní technologie a klesal rozměr tranzistoru. Jaká je současná situace? Na co vše má vliv zmenšení tranzistoru?

- **Čím menší tranzistor** – tím **větší leakage current** – neovlivnitelná spotřeba – nezávislá na frekvenci a překlápění.
- **Intel v současnosti používá výrobní proces 10 nm** (což je oproti konkurenci dost). Apple, ARM (Samsung / TSMC) i AMD mají daleko menší výrobní procesy (až 3 nm).

6.32. Vysvětlete pojmy MCM a CMT používané u AMD procesorů.

- **MCM**
  - **Multi-Chip Module.**
  - Mikroprocesor je rozdělen na “chiplety” a není již vyroben jako jeden monolitický chip.
  - Pokud se jednotlivá jádra nebo dvojice jader vyrábí jako samostatné “chiplety”, zůstane nám mnohem **více použitelných chipů**, ze kterých se “slepí” **vícejádrový mikroprocesor**.
- **CMT**
  - **Clustered Multithreading.**
  - Některé části procesoru jsou sdíleny dvěma vlákny, zatímco jiné části jsou přidělené jedinému vláknu.
  - **Například** – v jednom **CMT** modulu – 2 jádra pro celočíselné výpočty, ale jen jedno jádro pro **FP** výpočty. Jedna **L2 cache**, 2 **L1 cache**.

6.33. Uveďte základní rysy mikroprocesorů AMD Ryzen.

- Architektura Zen, AMD dohání po několika letech Intel.
- Zaměření na low-budget sestavy i naopak velmi výkonné stanice (AMD Threadripper).

6.34. Vysvětlete, co je to IA-32, x86-64, 32-bitová aplikace, 64-bitová aplikace, long-mode.

- **IA-32** – chráněný 32 bitový režim.
- **x86-64** – 64 bitová verze původní instrukční sady IA-32 (až  $2^{64}$  paměti = 16 Exabajtů).
- **32-bitová aplikace** - 32 bitové aplikace jsou spustitelné na libovolném z IA-32 vzájemně kompatibilních procesorů.
- **64-bitová aplikace** - aplikace napsaná pro long mode.
- **Long-mode** – název režimu x86-64.

6.35. Popište Moorův zákon.

- **Počet tranzistorů, které mohou být umístěny na integrovaný obvod se při zachování stejné ceny zhruba každých 18 měsíců zdvojnásobí.**
- Předpovězeno 1965.
- Přestává platit, zpožděuje se, výkon totiž roste málo.

6.36. Vysvětlete Dennardovo škálování, s tím souvisí pojem leakage-current.

- **Zmenšováním tranzistorů zůstává jejich hustota příkonu konstantní, takže spotřeba energie zůstává úměrná ploše.**
- Dennardovo škálování již několik let neplatí!
- Zmenšující se tranzistory mají naopak větší spotřebu kvůli prosakujícímu proudu.

6.37. Popište Koomeyho zákon.

- **Výpočetní výkon dosažený za 1 kWh spotřebované energie se zdvojnásobuje každých 1,57 roku.**
- Týká se hardwaru všeobecně (nemluví se zde pouze o procesorech).
- Kommeyho zákon tedy dále platí, ale kvůli fyzikálním zákonům nemůže platit navždy.

6.38. Vysvětlete, co je to dark-silicone (pozor, většina z těchto pojmu je uvedena v prezentaci o AMD procesorech, kterou při přípravě k maturitě nesmíte vynechat, ačkoliv pak procesory AMD detailně znát nemusíte).

- Aby se zabránilo příliš vysokému ztrátovému výkonu, musí určitá část chipu zůstat neaktivní - neaktivní část se nazývá dark silicone.
- Práce mikroprocesorového jádra musí být rozdělena v čase a prostoru tak, aby nikdy "netopili" všechny tranzistory naráz.
- Dark silicone = vypnuté části. Pouze některé části křemíkového chipletu jsou aktivní, většina je vypnutý. Zapnuté části nemohou svítit trvale, jinak by vznikly hot-spotty – příliš žhavá lokální místa. Aktivita se tedy musí v procesoru neustále "stěhovat".

## 7. Cache paměť

- Důvod zavedení vyrovnávací paměti
- Plně asociativní paměť, přímo mapovaná cache, vícecestná cache paměť
- L1, L2 a L3 cache exkluzivní a inkluzivní

7.1. Začněte tím, že RWM paměti se dělí na statické a dynamické a že z ekonomických důvodů nemůže být operační paměť počítače PC statická – bylo by dobré i nějaké konkrétní číslo, takže uveděte běžnou kapacitu a cenu paměti a kapacitu cache v moderních procesorech.

	<b>SRAM</b>	<b>DRAM</b>
<b>Paměťová buňka</b>	Složitá	Jednoduchá
<b>Kapacita</b>	Nízká (kB)	Vysoká (GB)
<b>Rychlosť</b>	Vysoká	Nízká (stále oproti HDD či SSD vysoká)
<b>Cena</b>	Až několik milionů Kč za GB	50 Kč / GB – DDR4 90 Kč / GB – DDR5
<b>Použití</b>	Cache	RAM
<b>Spotřeba</b>	Velmi nízká	Vysoká
<b>Refresh</b>	Není potřeba	Je nutný

Běžný domácí počítač potřebuje několik (někdy i desítek) GB RWM paměti. Pokud by měl PC používat čistě **SRAM**, byla by cena zařízení extrémně vysoká (několik desítek milionů). Proto **SRAM** je čistě využívána jenom v cache (v rádu kB či MB) a DRAM jako hlavní RAM paměť v rádu GB.

7.2. U jakého procesoru se poprvé cache objevuje (jako externí a jako interní) a proč by neměla smysl u předchozích procesorů?

- **První externí cache** – 80386.
- **První interní cache** – 80486.
- Procesor by uměl vykonat mnoho instrukcí pro čtení nebo zápis do paměti, ale paměť nestihá a trvá jí příliš dlouho než vybaví nebo zapíše data.

### 7.3. Vysvětlete pojem asociativní paměť a klíč – velmi důležité a někdy je problém to vlastními slovy správně formulovat, takže si to připravte.

- K datům v paměti cache se NEpřistupuje přes adresu, jak jsme zvyklý u běžné paměti.
- Asociativní paměti jsou tvořeny tabulkou, která obsahuje vždy sloupec, v němž jsou umístěny tzv. **klíče, podle kterých se v asociativní paměti vyhledává**.
- Dále jsou v tabulce umístěna **data**, která paměť uchovává, a také další informace nutné k zajištění správné funkce paměti (informaci o platnosti, posledního využití dat atd.).
- Je to stejné, jako kdybychom ve vyhledávání v seznamu zaměstnanců nepoužívali indexy, ale hledali podle zvoleného klíče – například rodného čísla. Prakticky je to podobné, jako funguje hledání v SQL databází.

### 7.4. Plně asociativní paměť – ukažte na vlastním příkladu, jak by mohla vypadat primitivní cache, kde by každý bajt měl vlastní klíč a vysvětlete, proč je neefektivní (Dlouhý klíč, krátká data, mnoho komparátorů...).

Klíč	Data	platnost
12345h	A7h	1
2A4D1h	FFh	1
00000h	00h	0
FF2C5h	14h	1
145ADh	BCh	1
75683h	11h	1
A1122h	22h	1
71243h	5Ch	1

Tento záznam v paměti cache znamená, že je v ní z hlavní operační paměti zkopiovaný bajt z adresy 145ADh jehož hodnota je BCh

- Na každém řádku je **komparátor** – ten porovná hledanou hodnotu se všemi řádky najednou (neprochází se tedy řádek po řádku).
- Tato cache ale má poměrně velký problém – je každému uloženému datu (8 bitů) potřebujeme 20 bitový klíč! Mnohem více místa v cache zabírají klíče než užitečná data. Navíc obsahuje mnoho komparátorů (pro každý bajt jeden).
- **Plně asociativní cache – celá adresa dat je uložena jako klíč.**

7.5. Vysvětlete, proč se data cachují po blocích a jak se tím zkracuje délka klíče.  
Jakou adresou musí začínat 2B, 4B, 8B, 16B, 32B blok?

- Místo ukládání každého bajtu jednotlivě můžeme ukládat celý blok v paměti – to může být například dva, čtyři, osm či až šestnáct bajtů naráz.
- Vzhledem k tomu, že bajty tohoto bloku leží v operační paměti za sebou, stačí jako klíč uložit adresu prvního bajtu celého bloku.
- **Například:**
  - Bajty **4A h, FF h, 7E h, 9D h** leží v operační paměti za sebou na adresách **1234 5678 h, 1234 5679 h, 1234 567A h, 1234 567B h**.
  - Do cache můžeme pak uložit 4 bajtový blok:
  - **Klíč: 1234 5678 h; Data: 4A FF 7E 9D**
  - Uložení dat je nyní mnohem efektivnější.
  - Navíc můžeme nyní chytře podle délky zkracovat klíč – nemusíme uvádět poslední dva bity adresy (adresa dalších bajtů v bloku se dá dopočítat, bajty ležely v paměti za sebou) – tedy místo 32 bitového klíče můžeme použít 30 bitový.
- **Blok velký 2 B:**
  - První bajt bloku musí ležet na **sudé adrese** (**0, 2, 4, 6, 8, A, C, E**).
  - Klíč **nebude obsahovat poslední bit adresy**.
- **Blok velký 4 B:**
  - První bajt bloku musí ležet na adrese **dělitelné 4** (**0, 4, 8, C**).
  - Klíč **nebude obsahovat poslední 2 bity adresy**.
- **Blok velký 8 B:**
  - První bajt bloku musí ležet na adrese **dělitelné 8** (**0, 8**).
  - Klíč **nebude obsahovat poslední 3 bity adresy**.
- **Blok velký 16 B:**
  - První bajt bloku musí ležet na adrese **dělitelné 16** (**0**).
  - Klíč **nebude obsahovat poslední 4 bity adresy**.
- **Blok velký 32 B:**
  - První bajt bloku musí ležet na adrese **dělitelné 32** (**00, 20, 40, 60, 80, A0, C0, E0**).
  - Klíč **nebude obsahovat posledních 5 bitů adresy**.

7.6. Přímo mapovaná cache – vysvětlit v čem je výhodnější oproti plně asociativní (stačí jediný komparátor) a zmínit pojem třída. Jak se přímým mapováním zkrátí klíč?

- V plně asociativní cache mohl být datový blok uložený na jakémkoliv řádku.
- Z toho ale plyne zásadní nevýhoda – data je třeba hledat podle klíče na všech řádcích současně a k tomu je třeba **velké množství komparátorů**.
- **Adresy z operační paměti jsou do cache mapovány na určené řádky.**
- **Jednotlivé řádky cache paměti se očíslují a adresy v operační paměti se rozdělí do tříd.**
- **Příklad:** mám cache o kapacitě **1 kB** a ukládám data po 1 bajtu.
  - **Rozdělíme cache na 1024 řádků (tříd).**
  - Chceme uložit bajt z adresy **1234 5678 h = 0001 0010 0011 0100 0101 0110 0111 1000 b**.
  - Třídu určíme z posledních 10 bitů ( $2^{10} = 1024$ ) adresy **10 0111 1000 b = 632** (dekadicky).
  - Na 632. řádek nyní uložíme tuto hodnotu a jako klíč uvedeme zbylých 22 bitů – **0100 1000 1101 0001 0101 b**.
  - Pokud mikroprocesor nyní bude chtít hledat v cache, bude stačit **jeden jediný komparátor**, který porovná horních **22 bitů s klíčem**.
- **Příklad:** mám cache o kapacitě **16 kB** (16 384 B) a ukládám data po 8 B.
  - **Rozdělíme cache na 2048 řádků (tříd)** (16 384 / 8).
  - Chceme uložit bajt z adresy **1234 5678 h = 0001 0010 0011 0100 0101 0110 0111 1000 b**.
  - Jelikož ukládáme 8 bajtů, poslední 3 byty vůbec nevyužijeme.
  - Třídu určíme z posledních 11 bitů ( $2^{11} = 2048$ ) adresy **010 1100 1111 b = 719** (dekadicky).
  - Na 719. řádek nyní uložíme těchto 8 B a jako klíč uvedeme zbylých 18 bitů – **00 0100 1000 1101 0001 b**.
  - Pokud mikroprocesor nyní bude chtít hledat v cache, bude stačit **jeden jediný komparátor**, který porovná horních **18 bitů s klíčem**.

**Výhoda tohoto řešení – stačí jeden jediný komparátor. Klíč se zkracuje** (v těchto případech o 10 / 11 bitů).

## 7.7. Vícecestná cache – v čem je lepší než jednocestná?

- Předchozí příklad měl ale menší problém – **rádek, kam se data ukládají, může být obsazený úplně jinými daty úplně jiné adresy, které však patří do stejné třídy.**
- **Řešení je to, že budeme mít více tabulek – pokud je třída v jedné tabulce obsazena, využijeme jinou.**
- Počet komparátorů = kolika cestná je cache.

## 7.8. Fungování vícecestné cache je třeba demonstrovat na nějakém příkladu.

Při vylosování této otázky mi bud' naznačíte, jaký vlastní příklad máte připravený nebo dostanete zadaný nějaký příklad na vícecestnou cache – dle zadání nakreslíte několik tabulek (podle toho kolikacestná bude) a vypočítáte počet řádků, délku bloku, šířku klíče atd. – vše nakreslíte během přípravy, při vlastní zkoušce už není moc čas něco řešit.

- Čtyřcestná cache s kapacitou **8 kB**, bloky po **8 B**. Chceme uložit data z adresy 1234 5789 h.
- Jedna tabulka – **2 kB**.
- Data ukládáme po 8 B – **256 tříd**.
- 1234 5678 h = **0001 0010 0011 0100 0101 0110 0111 1000**.
- Ukládáme po 8 B, takže poslední 3 byty neřešíme.
- Rozděleno do 256 tříd – použijeme 8 bitů – **1100 1111** = 207 (dekadicky).
- Jako klíč použijeme zbylých 21 bitů.

## 7.9. Dále uvést, že moderní mikroprocesory mají oddělenou cache pro instrukce a pro data.

- Například Pentium má cache pro program 8 kB, tak i pro data dalších 8 kB.

## 7.10. Vysvětlit L1, L2 a L3 cache (kapacita, rychlos, cena). Kdy se poprvé objevuje L2 a kdy L3 cache.

- **L1** – nejrychlejší, nejdražší, v řádu kB.
- **L2** – většinou do jednoho MB, **poprvé na Pentium PRO** (256 kB)
- **L3** – většinou sdílená mezi jádry, v řádu MB, **poprvé na Intel Core Nehalem**

7.11. Vysvětlit rozdíl mezi inkluzivní/exkluzivní L3 cache a související výhody a nevýhody.

- **Inkluzivní**
  - Intel.
  - Obsah L1 je obsažen v L2 cache a L2 je obsažen v L3.
- **Exkluzivní**
  - AMD.
  - Data nejsou v cache uložena duplicitně a celkem tak lze cachovat více dat.
  - Pokud nejsou data nalezena ve velké sdílené cache, pak to ještě neznamená, že v některé z lokálních cache jiných sousedních jader se hledaná data nenachází. Takže je třeba všechny projít, což stojí čas navíc.

7.12. Jak funguje uvolňování místa v zaplněné cache – stačí znát a popsat LRU.

- **LRU (Least Recently Used)**
  - Vyřadí nejdéle nepoužívaný blok.
  - Je nutné mít uložené časové razítko, používá většina cache pamětí.
- **MFU (Most Frequently Used)**
  - Nejméně používaný se smaže.
  - Nutné brát v úvahu, aby nebyl odstraněn záznam, který byl nedávno přidán.
- **RANDOM**
  - Náhodný výběr, funguje poměrně dobře.
- **FIFO**
  - Odebrána položka, která je ve FIFO nejdéle (cache funguje jako průtoková fronta).

7.13. Co je to write-back?

- **Write-Through**
  - Starší způsob.
  - Data ukládána do cache se zapisujou současně i do pomalé RAM.
- **Write-Back**
  - Data jsou zapisována do RAM (tam jsou neaktuální) až v okamžiku odstranění z cache, poskytuje vyšší výkon.

## 8. Sběrnice a základní deska

- Základní vlastnosti a parametry sběrnic
- Problematika sériové a paralelní komunikace, synchronizace, kódování, rušení
- Vývoj sběrnic
- Sběrnice PCI-E
- Základní deska, severní můstek, jižní můstek, chipset
- FSB, SDRAM, RAS, CAS, DDR paměti

8.1. Vysvětlit pojem sběrnice – co to je obecně a jaké má tato topologie výhody a nevýhody?

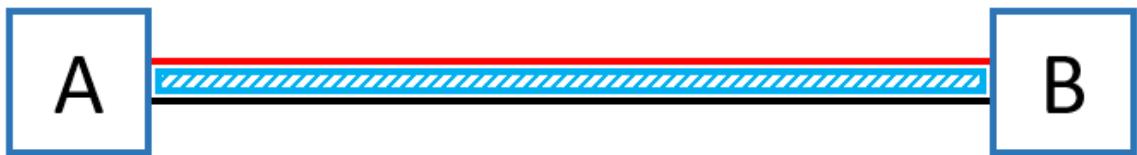
- Soustava vodičů, které umožňují přenos signálů mezi jednotlivými částmi počítače.
- Nejjednodušší topologie, snadno rozšiřitelná, ale může vysílat jen jedna stanice a je jich omezený počet.

8.2. Vysvětlit pojmy sériová / paralelní, jednosměrná / obousměrná, poloduplexní / plně duplexní, synchronní / asynchronní komunikace.

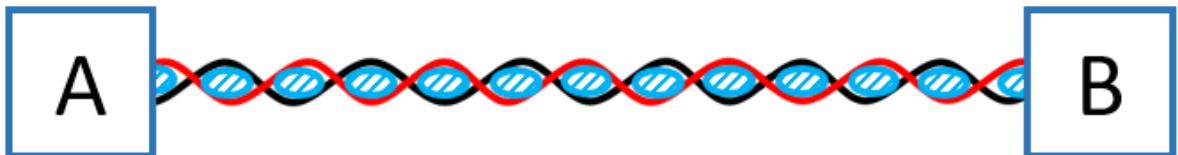
- **Sériová / paralelní**
  - **Sériová** – data se posílají bit za bitem.
  - **Paralelní** – posílá se více bitů najednou.
- **Jednosměrná / Polo-Duplexní / Plně-Duplexní**
  - **Jednosměrná** – vysílá se pouze jedním směrem.
  - **Polo-duplexní (Half Duplex)** – v jednu chvíli probíhá komunikace pouze jedním směrem.
  - **Plně-duplexní (Full Duplex)** – zařízení může současně přijímat i vysílat.
- **Synchronní / Asynchronní**
  - **Synchronní** – na lince je zastoupen hodinový signál (**CLK**), ten určuje platnost nastavení dat. Většinou u paralelních sběrnic.
  - **Asynchronní** – není zde žádný hodinový signál, většinou se používá kódování. Využívá se u sériových sběrnic.

8.3. Ukázat problém rušení – proč se používá vodič GND, proč má být GND veden těsně vedle datového vodiče ve dvoulince, proč vznikla kroucená dvoulinka, jak vypadá symetrická linka s diferenciálním signálem (data+, data-) – POZOR tohle není v prezentacích a probírali jsme to detailně a dlouho, musíte to znát, budu se na to ptát.

- Signál na datových linkách je definován napěťovou úrovní.
- **Napětí je rozdílem dvou potenciálů.**
- Napětí na datových vodičích proti nějakému potenciálu (proti nějaké "nule"). **GND** vodič zajistí stejný základní potenciál na obou koncích linky – napětí na datové lince se měří vůči "nule" na lince **GND**.
- Kdybychom neměli vodič **GND**, je dost možné, že bychom naměřili na jednom konci vodiče rozdílnou hodnotu než na druhém.
- Rušení na vodič přichází z různých směrů – čím menší bude rozestup mezi datovým a **GND** vodičem, tím vodiče zachytí menší rušení.



- Tím, že oba vodiče budou do sebe zakroucené, se významně zmenší plocha obvodu (modře vyšrafováné) – vzniká tak kroucená dvoulinka.



- Symetrická linka je velmi odolná proti rušení. Signál povedeme celkem dvakrát – na jedné lince je ovšem signál invertovaný (v opačném napětí). Následně signály můžeme od sebe odečíst a vydělit dvěma:
  - Původní signál **5 V**, invertovaný signál **-5 V**.
  - $5 - (-5) = 10 \Rightarrow 10 / 2 = \underline{\underline{5 V}}$ .
  - Signál bude rušen a "sníží" se o **1 V**:  $4 - (-6) = 10 \Rightarrow 10 / 2 = \underline{\underline{5 V}}$ .
- **Tx** = Transmit – vysílání
- **Rx** = Receive - příjem



8.4. Zejména důležité je chápat “odrušení” metodou (data+/data-) a ukázat jako příklad PCI-Express linku x1 (Tx+, Tx-, Rx+, Rx-).

- Viz předchozí otázka + obrázek.

8.5. Musíte vědět, k čemu je hodinový signál a synchronní komunikace a když se bude zdát, že tomu nerozumíte, budu chtít nakreslit průběh nějakého datového a hodinové signálu a vysvětlit, jak to funguje.

- **Hodinový signál** – u paralelní komunikace je problém, aby se poznalo, kdy jsou všechny vodiče nastaveny – proto vodič **CLK** udává tu dobu, kdy má příjímačem dojít k přečtení těchto hodnot na vodičích.
- **Asynchronní komunikace** – **CLK** se nepoužívá, používá se kódování.

8.6. Vysvětlete, proč paralelní komunikace musí být synchronní.

- Viz předchozí bod.

8.7. Uveďte všechny sběrnice v počítačích PC, jak šly historicky po sobě. Ke každé byste měli vědět, jestli je sériová nebo paralelní. Pokud je paralelní (a to jsou všechny kromě PCI-E), tak jakou má šířku. Dále přibližné období (jaký mikroprocesor se používal v počítači s danou sběrnicí).

- Všechny až na PCI-E jsou paralelní, Half Duplex a synchronní.

Označení	Bitová šířka [b]	Rychlosť	Procesory
<b>PC Bus</b>	8	4.77 MHz	1983 – IBM PC/XT – 8086
<b>ISA</b>	16	8 MHz	80286, 80386, 80486
<b>MCA</b>	32	8 MHz	IBM PS/2 – 80286, 80386
<b>EISA</b>	32	8 MHz	80386, 80486
<b>VL Bus</b>	32	33-50 MHz	80486, Pentium
<b>PCI</b>	32 / 64	33-66 MHz	Pentium
<b>PCI-X</b>	64	až 533 MHz	Pentium
<b>AGP</b>	32	66 MHz	Pentium
<b>PCI-E</b>	1+ (více sériových)	Asynchronní (2.5 GHz)	Pentium 4, AMD Athlon a vyšší

#### Nějaké základní vlastnosti:

- MCA se moc nerozšířila kvůli nekompatibilitě.
- ISA je rozšíření PC Bus, EISA rozšíření ISA.

8.8. U PCI sběrnice se zastavte déle – je třeba zmínit Plug-and-Play (častá chyba je, že tvrdíte, že umožňuje připojování a odpojování za běhu počítače), dobré si zapamatujte, co vše se dříve muselo nastavovat ručně. Dobré by bylo také znát různé varianty frekvence, šířky a napětí.

- **Na PCI přidáno PnP - Plug and Play:**
  - Dříve bylo po vsunutí nové karty nutné kartu ručně nastavit a následně i její ovladač (nastavení **IRQ** – přerušovací signály, DMA řadič, číslo brány apod.).
  - Kartu s **PnP** je možné vložit do slotu vypnutého počítače a počítač automaticky zasunutou kartu rozpozná a nakonfiguruje.
  - Automaticky se nastaví všechny informace tak, aby nedošlo k žádnému konfliktu mezi existujícími kartami. OS vyhledá potřebné ovladače, které s novou kartou spolupracují.
- První verze karty používají **5 V**, frekvenci **33 MHz** a šířku dat **32 / 64 bit**.
- U dalších verzích se snižuje napětí na **3.3 V** a frekvence roste na **66 MHz**.
- Propustnost můžeme vypočítat jako **(šířka přenosu \* frekvence) / 8**.
- První verze PCI (32 bit) má tedy propustnost **(32 \* 33) / 8 = 132 MB/s**.

8.9. U sběrnice PCI-X je potřeba vědět, že dále už to paralelně nešlo – nejde ani zvýšit frekvenci ani počet bitů (ledaže by se snížila frekvence). Co je to ECC a proč tam je?

- Paralelní přenos způsobuje tolik rušení, problému při synchronizaci a přeslechů, že je nutné používat **ECC – Error Correction Code**, který automaticky opravuje chyby vzniklé při přenosu dat.
- Datový tok až **(533 \* 64) / 8 = 4.3 GB/s**.

8.10. PCI-E musíte vědět, že je sériová a plně duplexní jako jediná. Vysvětlit, co je to PCIe-x16 a že to není paralelní komunikace (proč?).

- PCI-E se stává v roce 2004 novým standardem.
- Přechází na sériovou komunikaci a umožňuje dosažení velmi vysokých frekvencí (odpadá problém se synchronizací, protože komunikace je sériová).
- Základní “modul” x1 obsahuje dva symetrické páry vodičů (jeden pro **Tx** a jeden pro **Rx**).

- Vyšší násobky vznikají paralelním vedením několika těchto základních jednobitových kanálů – každý ale komunikuje v jiných časech, není to proto paralelní komunikace!

8.11. Přenosová rychlosť na PCI-E, PCI-E 2.0, PCI-E 3.0, PCI-E 4.0 a 5.0 - propustnosť jednotlivých “násobků“, použité kódování.

- **U 8b10b - 1 B = 10 b.**
- **Scrambling** = 128b130b – je to ale matematický algoritmus než přímý převod.

	<b>Frekvence</b>	<b>x1</b>	<b>x16</b>	<b>Kódování</b>
<b>PCI-E 1.0</b>	2.5 GHz	250 MB/s	4 GB/s	8b10b
<b>PCI-E 2.0</b>	5 GHz	500 MB/s	8 GB/s	8b10b
<b>PCI-E 3.0</b>	8 GHz	1 GB/s	16 GB/s	Scrambling
<b>PCI-E 4.0</b>	16 GHz	2 GB/s	32 GB/s	Scrambling
<b>PCI-E 5.0</b>	32 GHz	4 GB/s	64 GB/s	Scrambling

8.12. Vysvětlit kódování 8b10b – povinně musí znát každý. Také musíte vědět, že se používá u sériové asynchronní komunikace a jinde nemá smysl. Uvedte příklady, kde všude se používá.

- Jeden **užitečný bajt** je **transformován na desetibitovou posloupnost** – zaručen dostatečný počet změn signálu a vyvážený poměr nul a jedniček.
- Není moc dobré přenášet po lince příliš nul či jedniček za sebou – **je vhodné data střídat**.
- Použití pouze u **sériové asynchronní komunikace** (tam kde není [CLK](#)).
- **Použití** – USB 3.0, PCI-E 1.x a 2.x, Gigabit Ethernet, SATA, Thunderbold 1.x a 2.x.

8.13. Jaké kódování je na PCI-E 3.0 – 5.0?

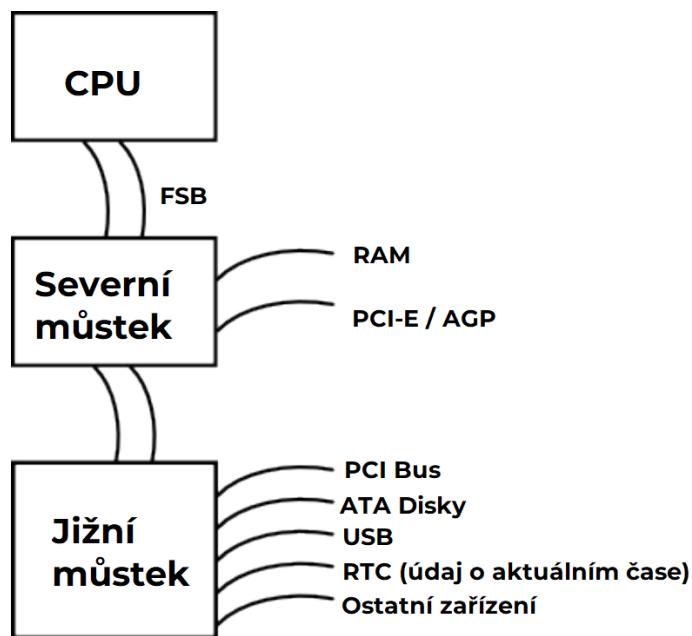
- **Scrambling** = 128b130b – je to ale matematický algoritmus než přímý převod.

#### 8.14. Severní můstek a jeho role.

- Zajišťoval komunikaci mezi CPU, pamětí RAM (řadič paměti), AGP portem nebo PCI Express sběrnicí a také zajišťuje spojení s jižním můstkem.
- Severní můstek tedy byl na základních deskách základním prvkem, který určuje rychlosť, typ procesorů a typ paměti RAM, který lze na dané základní desce použít.
- Později integrován do procesoru.

#### 8.15. Jižní můstek a jeho role – není potřeba umět kreslit všechny ty obrázky, které jsme si promítali, ale byl by dobrý jeden jednoduchý s procesorem nahoře, pod ním severní můstek, dole jižní můstek a kolem toho paměť, grafika, USB atd....

- **Realizuje pomalejší I/O funkce základní desky** – většinou obsluha pomalejších zařízení.



#### 8.16. Do moderních mikroprocesorů se nám severní můstek integroval – uvést detaily, výhody.

- **Mikroprocesor komunikuje s paměťovými moduly přímo, a ne prostřednictvím dalších chipů na základní desce.**
- Integrace řadiče paměti do procesoru má za následek značné zrychlení přístupu k RAM paměti.

8.17. Vysvětlit pojmy FSB a násobič. Proč uvnitř procesoru může být vysoká frekvence, ale s vnější okolím se na této frekvenci komunikovat nedá?

- **FSB = Front Side Bus.**
- **Soustava vodičů vedoucích přímo z vývodů mikroprocesoru**, kterými komunikuje s nejbližším okolím (severním můstkom).
- **Synchronní paralelní polo-duplexní** lokální systémová **sběrnice**.
- Je běžné, že frekvence, na které zde probíhá komunikace, je nižší než vnitřní taktovací frekvence mikroprocesor (**přenos mnoha bitů vysokou rychlostí je možný pouze v mikroprocesoru**).
- Moderní BIOS umožňuje nastavit frekvenci FSB a hodnotu násobiče taktovací frekvence pro mikroprocesor.
- FSB tvoří komunikační cestu mezi mikroprocesorem a chipsetem (chipset a jeho severní a jižní můstek je spojovacím bodem pro všechny sběrnice).

8.18. Popsat roli severního můstku při styku s pamětí SDRAM.

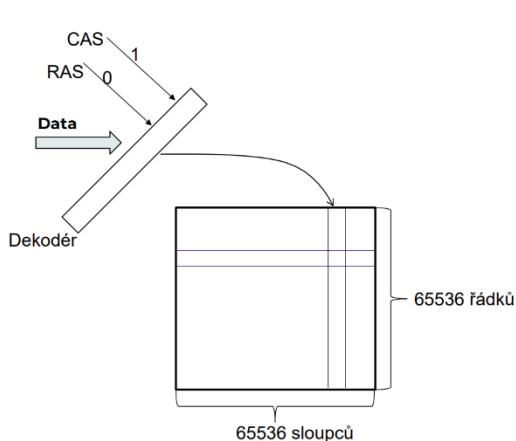
- **Synchronní DRAM = SDRAM.**
- SDRAM má vstup pro hodinový signál a veškeré události jsou synchronizovány a odvozeny od tohoto hodinového signálu
- Mikroprocesor dříve nekomunikoval s pamětí přímo – paměť a FSB mikroprocesoru mohou běžet na zcela jiných frekvencích, a navíc mikroprocesor nemá čas starat se o správné načasování výběru sloupce a řádku
- O adresaci a časování paměti se tedy staral severní můstek.

8.19. Vysvětlete, co znamená zkratka DDR a SDRAM.

- **SDRAM = Synchronní DRAM** – veškeré události odvozeny od signálu **CLK**.
- **SDR = Single Data Rate DRAM** – typ SDRAM, data se přenáší při vzestupné hraně **CLK**.
- **DDR = Double Data Rate DRAM** – typ SDRAM, data se přenáší při vzestupné i sestupné hraně **CLK**, používá se v současných PC.

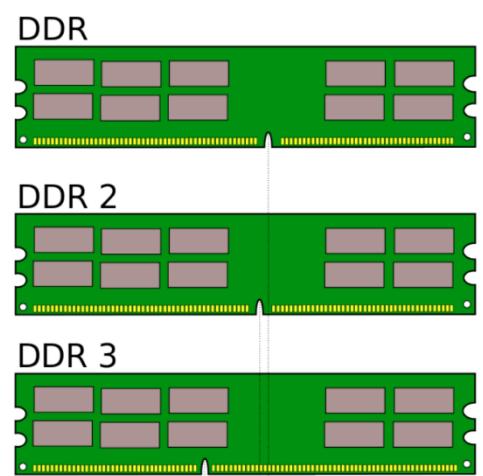
## 8.20. Co je to CAS, RAS a jak vlastně vypadá dekodér (řádky, sloupce).

- Výběr adresy paměti (vypadá jak tabulka) probíhá pomocí dekodéru:
  - **Horní bity adresy** – vybírájí řádek paměti.
  - **Spodní bity adresy** – vybírají sloupec.
- Nejprve vybíráme řádek a pak sloupec.
- Doba výběru řádku – **Row Access Strobe (RAS)**.
- Doba zpoždění zápisu do sloupce - **Column Access Strobe (CAS)**.
- **CAS Latency** – Označováno také jako **CL** – Kolik taktů je třeba počkat na data, po tom, co byl vybrán sloupec.
- Můžeme teoreticky použít ale jenom jeden adresový dekodér, protože řádek a sloupec se nevybírá ve stejnou chvíli. Pomocí **nastavení bitů RAS / CAS** nastavíme, zda se **vybírá řádek či sloupec**.



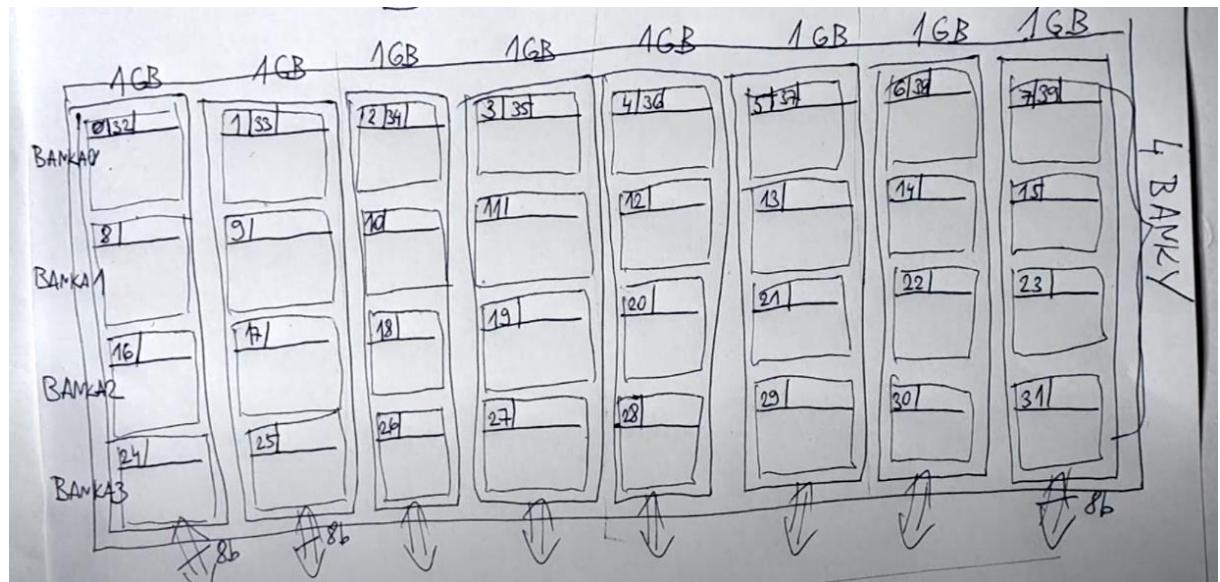
## 8.21. Jak vypadají paměťové moduly? V čem se liší DDR2 od DDR?

- **DDR** – 184 pinů, pracuje se 64 b.
- **DDR2** – 240 pinů, při stejné frekvenci nabízí dvakrát vyšší výkon než DDR (pracuje se 128 b).
- Velmi nízké napájení (do 3 V).



## 8.22. Co jsou to banky a jak zrychluje prokládané časování přístupu do paměti?

- Jeden paměťový modul (8 GB) je složen například z osmi chipů (po 1 GB).
- Tyto 1 GB moduly budou rozděleny do 4 bank (256 MB každý).
- V okamžiku, kdy požádám o data z adres 0-7, budu moci hned v dalším půl taktu žádat o data z adres 8-15. Data sice získám až po 2 taktech, ale žádat o další data, i když ještě jsem předchozí neobdržel.



## 8.23. Jakým způsobem se označují paměťové moduly – co znamená DDR-400, DDR2-400, PC3200 atd. – bylo pěkné, když k tomu budete mít připravený nějaký vlastní příklad, kde předvedete význam těchto čísel, jinak se vás něco zeptám já a budete to muset na místě pohotově vymyslet.

### • DDR-200

- Typ DDR (1) - (frekvence jako polovina z čísla v typu).
- Rychlosť **100 MHz**, vykoná **200 milionů cyklů** za sekundu.
- Šířka toku dat je 64 bitů –  $200 * 8 \text{ B} = 1600 \text{ MB/s} = \text{PC1600}$ .

### • DDR-400

- Typ DDR (1) - (frekvence jako polovina v typu).
- Rychlosť **200 MHz**, vykoná **400 milionů cyklů** za sekundu.
- Šířka toku dat je 64 bitů –  $400 * 8 \text{ B} = 3200 \text{ MB/s} = \text{PC3200}$ .

### • DDR2-400

- **Typ DDR2** - (frekvence jako čtvrtina v typu).
- Rychlosť **100 MHz**, vykoná **200 milionů cyklů** za sekundu.
- Šířka toku je 128 bitů –  $200 * 16 \text{ B} = 3200 \text{ MB/s} = \text{PC2-3200}$ .

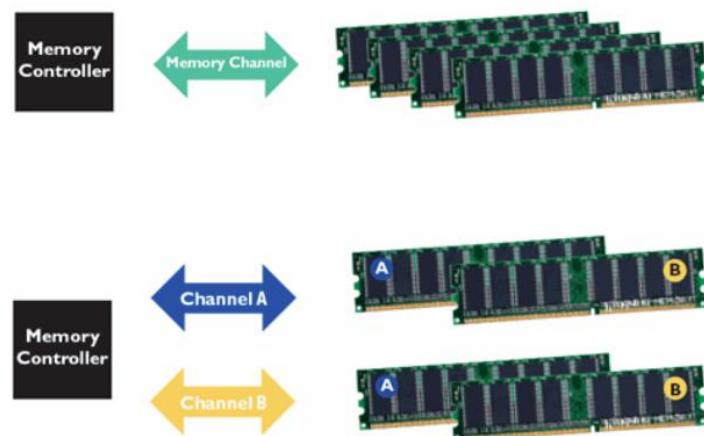
8.24. U moderních modulů vysvětlete, co znamená například: DDR4 2400 MHz

– PC4-19200.

- Nelze již standardně počítat jako DDR / DDR2.
- **DDR4 2400 MHz** – Paměť má **frekvenci 2400 MHz**.
- **PC4-19200** – **Propustnost je 19200 MB/s.**

8.25. Vysvětlete pojmy single-channel a dual channel.

- **Single-channel** – řadič paměti obsluhuje paměťové moduly přes jednu sdílenou komunikační sběrnici.
- **Dual-channel** – řadič paměti dokáže komunikovat dvěma oddělenými paměťovými sběrnicemi najednou. To pak zvyšuje propustnost v případě, že se čtou naráz data ležící ve dvou různých paměťových modulech a každý z nich je připojen k jiné paměťové sběrnici.



## 9. Připojování periferních zařízení

- Vývoj rozhraní pro připojování periferních zařízení (Sériová linka, Paralelní port, PS/2)
- USB
- Další rozhraní (Firewire, eSATA)
- Přenos obrazového signálu (VGA port, DVI, HDMI, DisplayPort)

9.1. Bylo by dobré začít nějak obecně a ne se rovnou vrhnout na popis konkrétní linky. Povídejte třeba o tom, že dnes vše dospělo k univerzální USB a veškerá moderní komunikace probíhá sériově. Dříve tomu tak nebylo a každé zařízení mělo svůj vlastní typ konektoru a linky.

- Dříve především použití sériového a paralelního portu.
- Většina zařízení svůj vlastní nekompatibilní konektor.
- Postupné směrování k sériové komunikaci a USB. Pro obraz z VGA na HDMI.

9.2. Jakým způsobem komunikuje mikroprocesor s okolím – porty, instrukce IN a OUT.

- Velmi podobné jako instrukce MOV.
- **Čtení dat ze vstupu** – IN.
- **Poslání dat na výstup** – OUT.
- **Vstupní a výstupní zařízení** mají svoji adresu – číslo **portu** (nebo brána).
- Signál M/IO (Memory / InputOutput) určuje, zda signál na adresové sběrnici má význam **adresy paměti** nebo identifikuje **I/O zařízení**.
- **Ovládáním I/O zařízení** probíhá **zápisem nebo čtením datové sběrnice**.
- Každé I/O zařízení je **identifikováno jiným číslem portu**.
- Na 8086 je dostupných 65535 bran.

9.3. DMA – k čemu to je dobré, jak by to třeba mohla využít zvuková karta (pozor, zařízení nepřistupuje samo do paměti, ale přenos dat kontroluje DMA řadič, který je naprogramovaný, aby z vybrané oblasti přenesl data do vstupně / výstupního zařízení bez účasti procesoru).

- Možnost přenosu dat mezi I/O zařízením a pamětí **bez účasti procesoru**.
- Procesor jen naváže přenos a dále se o přenos stará už jen **DMA řadič**.
- Nelze využít mezi dvěma I/O zařízeními – vždy jen přístup zařízení do paměti.

9.4. Stručně popsat, jakou roli hraje přerušení při komunikaci s periferními obvody.

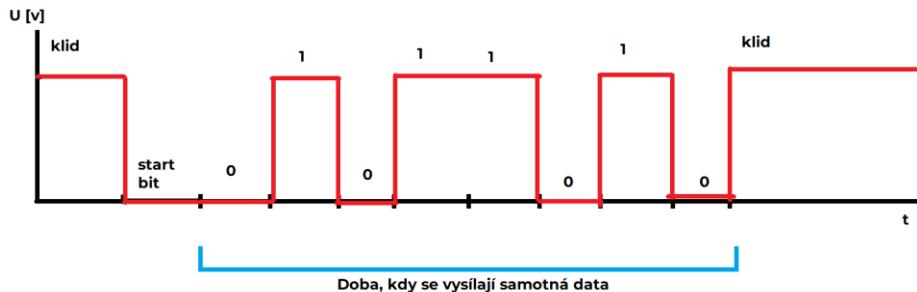
- I/O zařízení potřebuje mít k dispozici vhodný mechanismus, který umožní **přihlásit se k procesoru a vynutit si jeho pozornost**.
- Procesor na přerušení zareaguje a začne obsluhovat dané zařízení. Nemusí tak neustále kontrolovat, zda je zařízení například připraveno k odesílání dat.
- Pro každé přerušení je v paměti uložen **přerušovací vektor** (tj. **odkaz na adresu**, na které se nalézá jeho **obsluha přerušení**).
- Prvních 1024 bajtů paměti na Intelu 8086 je vyhrazeno pro 256 vektorů přerušení.
- V chráněném režimu se použije číslo přerušení jako index do **tabulky IDT**.

9.5. Vyjmenovat základní vstupy a výstupy moderního PC.

- Různé generace USB (včetně USB-C).
- HDMI / Display Port / VGA (spíše starší desky).
- Jack 3.5 mm (vstupní i výstupní).
- RJ45 Ethernet port.
- Optický audio konektor.
- Wi-Fi anténa (přijímač).

9.6. Sériová linka – nakreslete přenos jednoho bajtu (klid, start bit, osm bitů, stop bit). Ukažte nějakou typickou přenosovou rychlosť (např. 9600 b/s) a spočítejte kolik se dá přenést bajtů za sekundu (dělíme deseti).

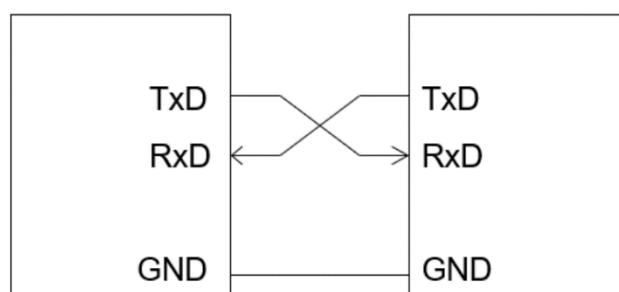
- Chceme odvysílat bajt 5A = **0101 1010**.
- První vysíláme poslední (nultý) bit – **používá se LSB!**



- **Typické rychlosť vysílania** – 4800 b/s, 9600 b/s, 19200 b/s (je to stejné jako u Arduina).
- **Počet prenesených bajtov bude ale menší** (musíme započítat start a stop bit) – takže skutečne prenesených bajtov bude 10x méné (napríklad 9600 / 10 = 960 B/s).
- Jeden bit (jedno okénko) trvá preniesť  $1 / 9600$  s.
- **Sériový port** – používal sa pro pripojenie myši, modemu, tiskárny, nahrazeno USB. Väčšinou modrý (vypadá ako female menšieho VGA). Označovaný ako **COM**.

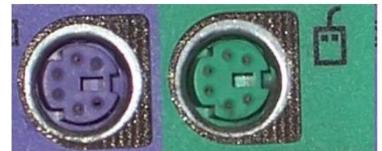
9.7. Nakreslete dvě full-duplexně propojené stanice přes sériovou linku (signály Tx, Rx a GND). Vysvětlete, že komunikace je asynchronní a přijímač rozpozná začátek vysílání díky startbitu, čas příjmu jednotlivých bitů si pak musí odměřit sám. Nezapomeňte, že klidová úroveň je jednička a vysvětlete, proč startbit musí byt nulový.

- **Startbit** = začátek vysílání – vždy z jedničky přejde na **0** a pak teprve se začnou vysílat samotná data.
- **Stopbit** = na konci vysílání musí byt alespoň **1** reset do klidu (na **1**).



9.8. PS2 – u kterých zařízení se používalo, přibližná rychlosť komunikace, nemusíte kreslit žádný průběh signálu. Musíte znát, jak vypadá konektor, jaké má barvy, jaké tam jsou signály. Komunikaci klasifikujte jako sériovou, synchronní a poloduplexní. Proč je tam hodinový signál?

- **6 pinový konektor** (využity pouze 4 ([+5V](#), [GND](#), [DATA](#) – sériový přenos a [CLK](#)).
- Slouží pro připojení klávesnice (fialový konektor) a myši (zelený konektor).
- Sériová, synchronní, polo-duplexní komunikace.
- 10 kHz (cca **900 B/s**).



- Má [CLK](#) signál, přesto je komunikace sériová (má i start a stopbity).
- Každá **klávesnice má totiž vlastní rychlosť vysílání** (není standardizované), navíc klávesnice většinou mlčí, takže je **nutné zahájit vysílání**.

9.9. Jaké existují možnosti připojení myši (PS2, USB), v jakém režimu komunikuje myš přes USB (interrupt) a jaké informace myš do počítače odesílá?

- **Drátová** (tedy ne bezdrátová) myš lze připojit dvěma způsoby: **PS2 a USB**.
- Myš zasílá informace o stisku / uvolnění tlačítek.
- Myš využívá **interrupt režim** (vysvětleno dále).

9.10. Paralelní port – nemusíte znát všechny signály. Musíte dálé vědět, že se posílá osm bitů naráz a že je to poloduplexní. Detaily o ECP a EPP není třeba znát. Bylo by pěkné popsat strobování a zabránění zahlcení – ale není nutné.

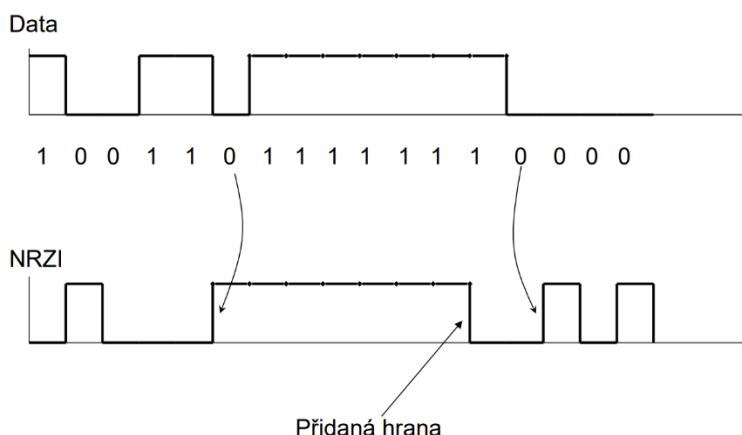
- **LPT – Line Printer Terminal.**
- Navržen pro komunikaci s tiskárnou, použit také pro připojení scanneru, externí zvukové karty, hardwarových klíčů.
- 25 vodičů (8 datových, 4 řídící, 5 stavových, 8 GND).
- Paralelní (posílá osm bitů), polo-duplexní, synchronní.
- [ECP](#) oproti [EPP](#) povoluje přístup do paměti přes [DMA](#), tedy rychlosť se zvyšuje z 2 MB/s na 2.5 MB/s. Navíc [ECP](#) i [EPP](#) umí komunikovat obousměrně a střídat se podle toho, kdo chce vysílat (nemusí využívat jen stavové vodiče).
- Signál na vodiči [Strobe](#) signalizuje, že data na datových vodičích jsou platná a tiskárna může zahájit jejich čtení.

9.11. USB – nejdůležitější část této otázky a měli bychom tomuto tématu věnovat většinu času - musíte znát linky, konektor a správně jí klasifikovat jako sériovou, asynchronní, poloduplexní.

- **Universal Serial Bus**, 1995.
- **Sériové, asynchronní, polo-duplexní.**
- Dva konektory **Typ A** a **Typ B** (většinou u tiskáren či Arduino Uno).
- Čtyři vodiče - **+5 V (red)**, **Data- (white)**, **Data+ (green)**, **GND (black)**.
- Lze připojit až 127 zařízení <1; 127>.
- Zařízení dvojího typu – **huby** (rozbočovače) a **koncová zařízení**.

9.12. Popsat NRZI kódování – obrázek není nutný (u kterých verzí se používalo?).

- **NRZI - nuly** v datech **vedou ke změně úrovně, jedničky** nechávají úroveň signálu **bez změny (pozor, v PS to je přesně obráceně!)**.
- Pokud je šest jedniček po sobě, přidá vysílač automaticky jednu nulu.
- Speciální zaváděcí bajt – **0000 0001 b**.
- Používá pouze **USB 1.1** a **USB 2.0** (USB 3.0 používá **8b10b**)



9.13. Povinně vysvětlit interrupt, isochronní a bulk přenos (budu se ptát na ping, pokud to nevysvětlíte sami) .

- **Interrupt přenos**
  - Zařízení, které vyžaduje řízení pomocí přerušení.
  - Například pro klávesnici či myš – generuje přerušení při každém stisku klávesy, které je následně zpracováno.
  - USB hostitel se pak každých 10 ms ptá, jestli nemá zařízení nové události či data.

- **Bulk přenos**
  - Přenos velkých bloků dat – tiskárna, externí disk.
- **Isochronní přenos**
  - Stálý příspun dat (**real time**), ale nepožaduje se zde doručení – mikrofon, reproduktory, web kamera.
  - Přenosová rychlosť nesmí kolísat a záleží na pravidelnosti.
- **Řídící přenos**
  - Používán pro konfiguraci zařízení.

**Ping** – slouží ke zjištění, jestli je připojené USB zařízení schopno zpracovat data, aby nebylo zahlceno. Používá se u **BULK** přenosu.

## 9.14. USB 3.0 – popsat nové signály a kompatibilitu, jaké je tu kódování.

- **USB 1.1 mělo dvě rychlosti:**
  - Low Speed – 1.5 Mb/s (klávesnice / myši).
  - Full Speed – 12 Mb/s (senzory, zvuková zařízení).
- **USB 2.0 přidává:**
  - High Speed – 480 Mb/s (disky, paměti, Wi-Fi modul).
- **USB 3.0 (Usb 3.1 Gen1) se podstatně liší:**
  - Používá **8b10b** místo NRZI.
  - **Nově 8 vodičů:**
    - 2 napájecí.
    - Data +, Data – pro starou kompatibilitu.
    - **SSTX (+/-)** – kroucený pár pro vysílání.
    - **SSRX (+/-)** – kroucený pár pro příjem.
  - **Nově je tedy Full Duplex.**
  - Zvýšení maximálního proudu ze 150 mA na **900 mA**.
  - Nový režim **SuperSpeed** – teoretická rychlosť **5 Gb/s**, reálná 3.2 Gb/s.
- **USB 3.1 (Gen2):**
  - **SuperSpeed+ – 10 Gb/s.**
  - Nový konektor **USB-C**, až 5A a 20 V – tedy **100 W**.
    - **Obousměrný** – 2 x 12 konektorů.
    - Mnoho využití – přenos obrazu (Display Port), napájení, připojení k počítačové síti.

9.15. USB 3.1 gen1 a gen2 – režim SuperSpeed a SuperSpeed+.

- Viz předchozí bod.

9.16. Popište přibližně konektor USB-C a jeho možnosti.

- Viz dva body zpět.

9.17. Rychlosť přenosu na USB, USB2.0, USB3.0, USB3.1 gen1, gen2.

- Viz tři body zpět.

9.18. Firewire – stačí vědět k čemu se to typicky používalo (dig. videokamery) a proč (USB 1.0 nestíhalo datový tok dig. videa).

- **Vyvinul Apple** v roce 1987 (Sony používá pod názvem i.Link).
- **Podporuje hotplug** (počítač není nutné mít vypnutý a zařízení se samo načte) a plug-and-play.
- Efektivnější protokol než USB (USB 1.1 mělo propustnost jen 12 Mbit/s), Firewire až 3200 Mb/s a 100 metrů, navíc Full Duplex.
- Umožňovalo jako jediný přenos videa z videokamery v reálném čase.
- Dražší, proto se tak nerozšířilo, navíc s příchodem USB 3.0 začalo upadat.

9.19. e-SATA – k čemu to je dobré, v čem je to lepší a horší než USB3.0.

- Externí SATA, zavedeno 2008.
- Rychlosť 3 Gb/s.
- V éře USB 2.0 vyšší propustnost, ale v éře USB 3.0 nemá smysl, navíc standardní konektor nemá integrované napájení.

9.20. VGA port – vysvětlit, co je to analogový signál a jak je přenášena informace o jasu. Klasifikovat jako analogový, paralelní, synchronní, poloduplexní. Popsat synchronizaci (pixely, řádky, snímky). Proč je do budoucna nevhodný? Jakým způsobem se připojovaly monitory před VGA (Hercules, CGA)?

- **Analogový** = nejsou to 1 a 0, ale je to hodnota úměrná velikosti napětí na lince.
- **Paralelní** (více datových vodičů), **synchronní** (dokonce i více CLK signálů), **jednosměrná** (data jdou pouze jedním směrem).
- Nelze přenášet zvuk, pouze obraz (maximálně 2048 x 1536 – cca Full HD).

- **Přenáší po třech linkách (paralelních) informace o jasu červené, modré a zelené** - tedy snadná možnost rušení a zkreslení obrazu.
- **Jeden hodinový** signál tiká při přenosu každého **pixelu** (signál [CLK](#)).
- **Druhý hodinový** signál tiká při přenosu každého **řádku** (signál [HSYNC](#)).
- **Třetí hodinový** signál tiká při přenosu každého **snímku** (signál [VSYNC](#)).

**Dříve proprietární konektory** – grafické karty **MDA** (Monochrome Display Adapter), **CGA** (Color Graphics Adapter) a **HCG** (Hercules).

9.21. DVI, HDMI – proč je lepší než VGA? Jaké se tu používá kódování? Jak je to se synchronizací a hodinovým signálem (pozor, dobře si to přečtěte a promyslete). Nemusíte znát všechny signály, ale přibližně způsob, jak se informace o barvě pixelu přenáší. Jak je to s rychlosí přenosu, je pevná? K čemu slouží HDMI-CEC? Co je to dual-link DVI?

- **DVI**

- Digitální obrazový signál (tedy ne analog) bez zvuku.
- **Sériová, polo-duplexní.**
- Rozlišení po jedné lince maximálně 2553 x 1436 – tedy skoro QHD (2K).
- Má i druhou nevyužitou linku ([DVI-DL](#) = DVI Dual Link), která se aktivuje při vyšším rozlišení.
- **6 vodičů pro přenos barev** (diferenciálně 3 + 3 – [data+](#) a [data-](#)) – RGB.
- Paralelně se odesírají tři čísla po třech různých linkách – přičemž hodnota jasu je pak na každé lince sériová.
- **Kódování TMDS** – pokročilé kódování 8b10b.
- **Synchronizace na lince je, ale jsou přenášeny jako desetibitové speciální kombinace.** Má ale i speciální synchronizační signály.
- Asynchronní typ přenosu s určitou doplňkovou synchronizací.
- Rychlosí přenosu se ovlivňuje dle rozlišení a snímkovací frekvence.

- **HDMI**

- **Vychází z DVI.**
- Podporuje i zvuk.
- Přes HDMI lze přenášet i povely zařízením ([HDMI-CEC](#)), je to stále ale polo-duplexní.
- Rozlišení až 10K.

9.22. DisplayPort – jak to vypadá a co to umí? Jsou zde linky pro R, G a B data? Je kompatibilní s HDMI? Přesná čísla s propustností není potřeba znát.

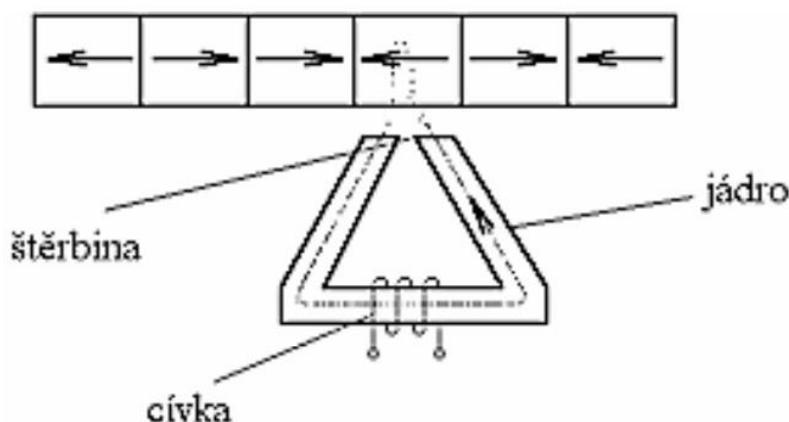
- Podporuje barevní hloubky až 16 bitů pro kanál, zvuk i možnost USB připojení. Rychlosť až 540 MB/s.
- Data jsou přenášena v paketech jako v počítačových sítích – mikro-paketový přenos.
- Dokáže vysílat i data jako HDMI, ale není to standardní.
- Kódování 8b10b, 1-4 linky, diferenciálně signál - [Data+](#) a [Data-](#).
- Vytvořila VESA, výhodné pro speciální požadavky (větší barevná hloubka), využití pro Thunderbolt, ale jinak není příliš důvod používat místo HDMI.
- Taktéž Half Duplex.

## 10. Pevný disk a jeho fyzická a logická struktura

- Princip magnetického záznamu dat, RLL kód
- Základní parametry pevných disků
- Fyzická struktura disku
- CHS, LBA, ZBR, LMR, PMR, SMR
- Souborové systémy FAT, NTFS, clustery a sektry
- Defragmentace
- MBR, oddíly, GPT, UEFI

10.1. Jak funguje magnetická záznamová hlava – chtělo by to obrázek, cívku, magnetický tok, zmagnetované domény...

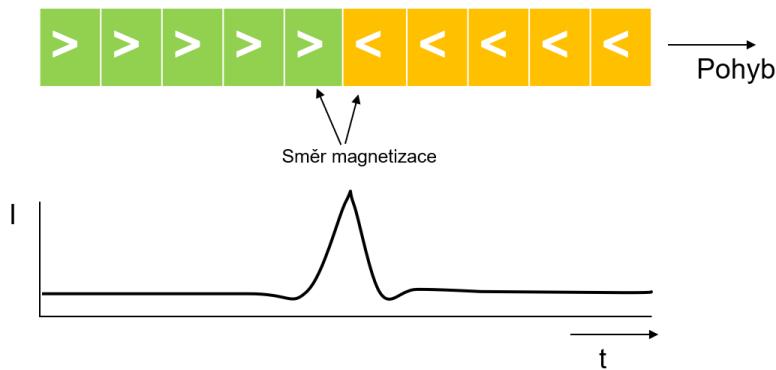
- Jednotlivé bajty se převádí na **sériový tok jednotlivých bitů**.
- Proud jedniček a nul je převáděn na **změny proudu v záznamové hlavě**.
- Změna proudu způsobí v záznamové hlavě způsobí **změnu magnetického pole v jejím okolí**.
- Hlava obsahuje **štěrbinu**, která umožňuje **uzavírání indukčních čar** přes magnetické médium v těsné blízkosti hlavy.



10.2. Jak funguje čtecí magnetorezistentní hlava? Proč používáme kódování?  
Jak vlastně funguje RLL kód?

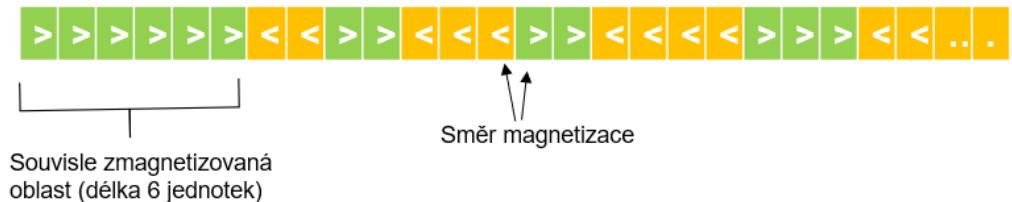
- Změna odporu vlivem změny magnetického pole se převede na proudovou špičku.
- **Detekovat lze jen změny magnetického pole!**
- **Informace musí být vhodným způsobem překódována, aby obsahovala dostatečný počet střídajících se jedniček a nul.**

## Záznamová vrstva



- Na magnetické médium se však vejde více dat, když počet změn (jejich hustota) bude co nejnižší. Je tedy nutné zvolit vhodný kompromis.
- **RLL kódování (Běžný Kód RLL 2-7):**
  - Maximálně 7 a minimálně 2 mezery.
  - Například, máme bity C2 h, 2A h.
  - Po RLL zakódování 1000.000100.100100.0100.1000.0100.0100.
  - Mezi jedničkami je 6, 2, 2, 3, 2, 4, 3 a 2 nuly.
  - **Řadič disku měří čas mezi impulzy, které přicházejí ze čtecí hlavy, a tím nepřímo délku oblasti.**

## Záznamová vrstva



## 10.3. Co je to LMR, PMR, SMR?

- **LMR** - Longitudinal Magnetic Recording – podélný směr magnetizace.
- **PMR** - Perpendicular Magnetic Recording – kolmý směr magnetizace (vyšší hustota oproti **LMR**).
- **SMR** – Shingle Magnetic Recording – vrstvený zápis – jednotlivé vrstvy se vzájemně překrývají – něco jako tašky na střeše.

10.4. Fyzická struktura disku = plotny, vychylovací cívky (lineární motor), raménka s hlavičkami – popsat, jak to funguje.

- Nad **plotnou** (kovový kotouč s tenkou magnetickou záznamovou vrstvou na obou) **se vznáší hlavy ve výšce** několika mikrometrů / nanometrů (čím menší, tím větší hustota zápisu).
- **Hardisk má několik ploten** nad sebou (běžné pro 3.5 disky jsou tak 4), všechny jsou na stejně ose.
- **Na jednu plotnu jsou dvě hlavy** (jedna ze zdola a jedna z vrchu – protože tam jsou dvě magnetické vrstvy).
- **Hlavičky** (plovoucí hlavy) jsou umístěny na **společném rameňi. Vystavovací mechanismus** je zařízení, které tyto hlavičky vystavuje na správné místo nad povrchem disku. Dříve se používal krokový motor, dnes **lineární motor**.
- Hlavičku ovšem **nazdvihuje proud vzduchu** (tedy plave na vzduchovém polštáři). Kdyby se disk zastavil, hlava přestane být odtlačována od povrchu a dotkne se ho – zničí ho.

10.5. Proč nesmí dovnitř proniknout prach? Proč se tomu říká "pevný" disk? Je uvnitř vakuum? (není a proč...) Co je to "parkování"?

- **Prach** usazený na plotnách **by mohl poškodit hlavičku a povrch**, protože jeho **zrnka jsou mnohem větší než mezera mezi hlavičkou a povrchem**, proto je disk uzavřen proti prachu.
- Kdyby vevnitř bylo vakuum, nefungoval by vzduchový polštář.
- Pokud je disk v klidu, tak jsou hlavičky umístěny mimo plotny = hlavičky jsou zaparkovány.
- **Pevný disk, protože je vyroben z pevného neohebného materiálu.**

10.6. Typické otáčky pevných disků (RPM).

- **RPM = otáčky za minutu.**
- **Typické hodnoty:** 3600 RPM, 4500 RPM, 5400 RPM (pro NAS), 7200 RPM (běžné domácí), 9600 RPM, 10000 RPM, 15000 RPM (serverové SAS).

10.7. Vysvětlit, co znamená CAV.

- **CAV (Constant angle velocity) = Konstantní úhlová rychlosť.**
- Otáčky jsou konstantní, z toho plyne, že rychlosť čtení je proměnlivá (nejvyšší u okraje, nejnižší u středu).

10.8. Co je to přístupová doba a jak se udává? Jakou přístupovou dobu mají současné disky? Jak závisí vybavovací doba na otáčkách?

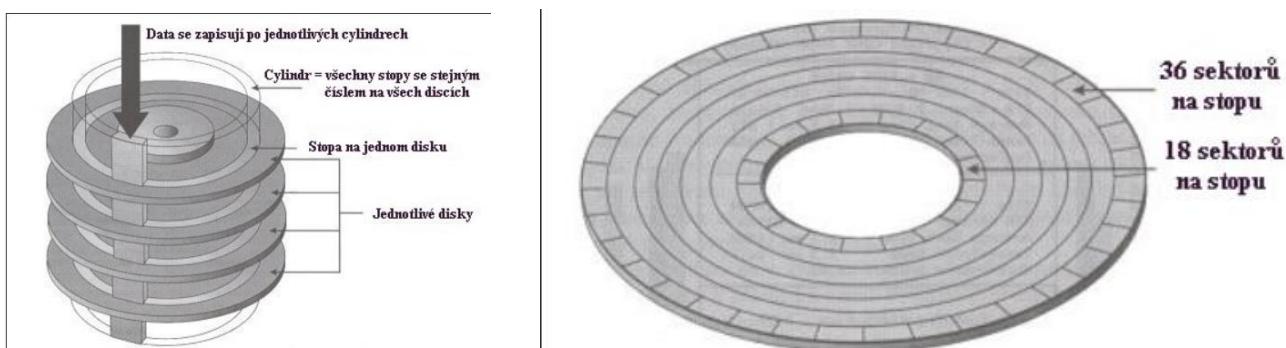
- **Doba mezi požadavkem na čtení dat a započetím jejich čtení.**
- Není konstantní (hlava se musí dostat na požadované místo, tam se utlumit rozkmit a plotna se pootočit na správné místo).
- Udává se v **ms. Dnešní běžné hodnoty 5 – 10 ms.**
- Čím vyšší otáčky, tím nižší přístupová doba.

10.9. Typické rozměry (průměr). Odolnost pevného disku vůči otřesům (závisí na průměru a otáčkách?).

- **Rozměr: 3.5 a 2.5 palce** (průměr plotny).
- **Vyšší otáčky = nižší odolnost.**
- 2.5 palcové disky jsou spíše odolnější než 3.5 palcové (mají menší otáčky a menší velikost plotny).

10.10. Vysvětlit pojmy cylindr, plotna, povrch, stopa, sektor – hodil by se obrázek.

- **Plotna** – jedna „deska“, na kterou lze ze zdola i z vrchu zapisovat.
- **Cylindr** – všechny stopy se stejným číslem na všech plotnách (takový válec).
- **Povrch** – plocha, na kterou lze zapisovat (tedy jedna strana plotny).
- **Stopa** – „kruh“ na jedné plotně.
- **Sektor** – jednotlivé místo pro zápis (rozdělená stopa) – zde probíhá zápis.
- Zápis probíhá skrz informaci: **číslo Cylindru – číslo Povrchu a číslo sektoru.**



10.11. Vysvětlit ZBR – nejlépe nakreslit dlouhou a krátkou stopu a na ní sektory.

- **Zone Bit Recording.**
- Nemá konstantní počet sektorů na stopu, a tak je povrch lépe využit (**u středu je sektorů méně**).

10.12. Proč u disku se ZBR není konstantní rychlosť čtení dat?

- **Na okraji je více sektorů**, takže data na kraji jsou **čtena rychleji než u středu** (otáčky jsou stejné, ale za dobu jedné otáčky tam pod hlavou „projede“ více sektorů).

10.13. Základní parametry moderních pevných disků – jakou kapacitu, vybavovací dobu, otáčky, rozměry a cenu mají současné disky?

- **Kapacita** – dneska běžné 1 – 20 TB
- **Vybavovací doba** - běžné hodnoty 5 – 10 ms.
- **Otáčky** - 5400 RPM (pro NAS), 7200 RPM (běžné domácí), 15000 RPM (serverové SAS).
- **Rozměr** – 2.5 palce (pro notebooky a servery) a 3.5 palce (desktop).
- **Cena** – cca 500 - 1000 Kč za TB (záleží na kapacitě).

10.14. Jak je to s GB, TB u udávané kapacity pevných disků? (Giga x Gibi).

- Používá se předpona **Giga / Tera** (tedy  $10^9 / 10^{12}$ ).  $1\text{ GB} = 1\,000\text{ MB} = 1\,000\,000\text{ KB} = 1\,000\,000\,000\text{ B}$ .

10.15. Jaké nevýhody má klasický pevný disk oproti SSD?

- Daleko menší rychlosť, delší vybavovací doba.
- Větší spotřeba (musí se stále točit).
- Náchylný na poškození.
- Hlučný, vydává více tepla.

10.16. Rozdíl mezi CHS a LBA adresací. Jaká omezení max. kapacity existovala/existují a z čeho vyplývají?

- **CHS (Cylinder / Head / Sector):**
  - Má omezení 1024 cylindrů, 16 hlav a 63 sektorů na stopu = cca 500 GB.
  - Příliš komplikované – muselo se různě přepočítávat pro novější disky.
  - Navíc nebylo příliš vhodné pro ZBR.
- **LBA (Logical Block Addressing):**
  - Lineární adresace – číslují se od nultého do posledního.
  - Standardní EIDE řadiče využívají **28 bitů**, což je také problematické = **maximálně 128 GB**.
  - Novější specifikace (ATA/ATAPI-6) má už **48 bitové adresování = maximálně 131072 TB**.

10.17. Co je to FAT tabulka?

- Starý souborový systém.
- Za sektorem boot následuje oblast **FAT** (File Allocation Table). Existuje i kopie hlavní **FAT**.
- V tabulce **FAT** jsou **informace o obsazených a neobsazených clusterech** (cluster odpovídá jednomu místu na disku – je to základní alokační jednotka souborového systému). Každý cluster má své číslo.
- Clustery jsou očíslovány pomocí 12, 16 nebo 32 bitů (podle toho **FAT12**, **FAT16**, **FAT32**).
- **Za FAT tabulkou je umístěn seznam souborů** (u každé položky je uveden název, přípona, atribut, čas, datum a poloha na disku (počáteční cluster)).
- **Samotná FAT neobsahuje informace o souborech**, pouze informace o tom, zda je volný, špatný, informace o následujícím clusteru souboru či zda je posledním clusterem souboru.
- Jednodušší příklad na rozdělení disku při FAT12 a disku 60 MB:
  - Na disku maximálně  $2^{12}$  clusterů = 4096 bloků.
  - 60 MB disk obsahuje 120 000 sektorů velkých **512 B (standardní velikost sektoru)**.
  - $120\ 000 \text{ sektorů} / 4096 \text{ clusterů} = 29.29 \text{ sektorů pro jeden cluster}$ .
  - Použijeme mocninu dvojky, takže **32 sektorů bude v jednom clusteru** = 16 KB / cluster.
  - Pokud budeme chtít uložit soubor o velikosti **40 KB**, využijeme **3 clustery = 48 KB**.

- Složitější příklad na rozdelení disku při **FAT16** na disku 1 GB:
  - Na disku maximálně  $2^{16}$  clusterů.
  - 1 GB disk obsahuje 2 000 000 sektorů velkých **512 B (standardní velikost sektoru)**.
  - $2\ 000\ 000$  sektorů / 65536 clusterů = 30.51 sektorů pro jeden cluster.
  - Použijeme mocninu dvojky, takže **32 sektorů bude v jednom clusteru** = 16 KB / cluster.
  - Pokud budeme chtít uložit soubor o velikosti **40 KB**, využijeme **3 clustery = 48 KB**.

10.18. Co je to cluster? Jaká je typická velikost sektoru. Předvedte nějaký vlastní ukázkový výpočet velikosti clusteru při určité kapacitě disku a max. počtu použitelných clusterů.

- Viz předchozí úloha.

10.19. Popište, jak dochází k fragmentaci a k čemu je defragmentace.

- Během používání disku dochází k fragmentaci disků – mazáním souboru vznikají volné clustery.
- **Volné clustery se ale vyskytují nerovnoměrně na disku** – je ovšem vhodné, aby velký soubor ležel na disku za sebou.
- Operační systémy nabízejí možnost defragmentace disku, kdy se přeskupí jednotlivé fragmenty souboru tak, aby soubory ležely po sobě jdoucích clusterech.

10.20. Co je to vlastně “souborový systém”? Jak se liší FAT12, FAT16 a FAT32?

- Kdyby se velikost clusteru = velikost sektoru, pak by byla tabulka s informacemi o clusterech a souborech velmi velká. Pracovat s malými clustery na velkém disku tedy není moc výhodné. Proto se zavedl souborový systém, který počet sektorů “omezí” – většina souborů je stejně větší jak 512 B. Navíc je nutné soubory nějak třídit a organizovat.
- **FAT12** – až  $2^{12}$  = **4096 clusterů**.
- **FAT16** – až  $2^{16}$  = **65536 clusterů**.
- **FAT32** – až  $2^{32}$  = **4294967296 clusterů**.

## 10.21. K čemu je CHKDSK?

- **Nástroj sloužící k opravě tabulky FAT:**
  - Ztracené clustery – ve **FAT** jsou označené jako používané ale k žádnému souboru nejsou přiřazeny.
  - Pro 2 a více souborů vyhrazen cluster se stejným číslem.

## 10.22. Jak byly dříve omezeny názvy složek a souborů?

- 8 B název a 3 B přípona – proto někdy místo přípony **.html** (4 B) je používána ještě přípona **.htm** (3 B).

## 10.23. Co se děje při smazání souboru? Jde obnovit?

- V tabulce **FAT** je označen příslušný cluster (nebo clustery) jako volný a položka v seznamu se označí jako neplatná.
- **Dokud se smazaný soubor nepřepíše, stále existuje.**

## 10.24. Popište NTFS – vysvětlit k čemu je MFT, BITMAP, LOGFILE, atributy.

- **NTFS** je navržen jako nativní souborový systém pro Windows NT a přináší spoustu novinek jako je třeba komprese, šifrování, kvóty a dlouhá jména souborů včetně diakritiky a různých znakových sad.
- **Je navržen jako obří databáze.** Může mít až  $2^{64}$  clusterů (ale většinou se používá 32 bitů).
- **MFT**
  - Master File Table.
  - Obsahuje záznamy o všech souborech a boot sektoru.
  - Na začátku disku.
- **BITMAP**
  - Jednorozměrné pole bitů, které slouží ke sledování volného místa.
- **LOGFILE**
  - **Žurnál.**
  - Všechny zápisy na disk se zaznamenávají do žurnálu. Pokud systém během zápisu havaruje, je možné podle záznamu všechny operace dokončit nebo anulovat.

- **Atributy (klasického souboru)**

- Název souboru, velikost, adresář apod.
- Unicode kódování.
- Přístupová práva, data souboru.

10.25. Jakou velikost clusteru používá NTFS?

- Standardně 4 kB.

10.26. Co je to MBR? Co vše je zde uloženo? Co se děje po startu počítače?

- Master Boot Record
- Prvních 512 Bajtů na disku.
- Informace o oddílech disku (v MBR až 4, jde ale rozdělit i navíc).
- Obsahuje zavaděč OS – krátký kód, který je při startu načtem BIOSem.
- **BIOS tedy načte zavaděč OS z aktivní partition (může být jen jedna) a spustí její spustitelný kód.**

10.27. K čemu může být dobré rozdělit disk na oddíly?

- Izolace souborů, více OS, více souborových systémů.

10.28. Co je to UEFI? SecureBoot? GPT?

- **GPT**

- **GUID Partition Table.**
- Nahrazuje starší tabulku MBR, která neumožňuje použít disk větší než 2 TB.
- Neomezené množství oblastí (informace o oddíle 128 b).
- Pouze pro novější OS, nekompatibilní s BIOSem (pouze s UEFI).

- **UEFI**

- **Nástupce BIOSu.**
- Většinou podporuje nové možnosti, jako je ovládání rozhraní UEFI myší (BIOS umí jen klávesnici).
- Má vlastní **zavaděč OS**, kde jsou uloženy boot managery operačního systému.

- **SecureBoot**

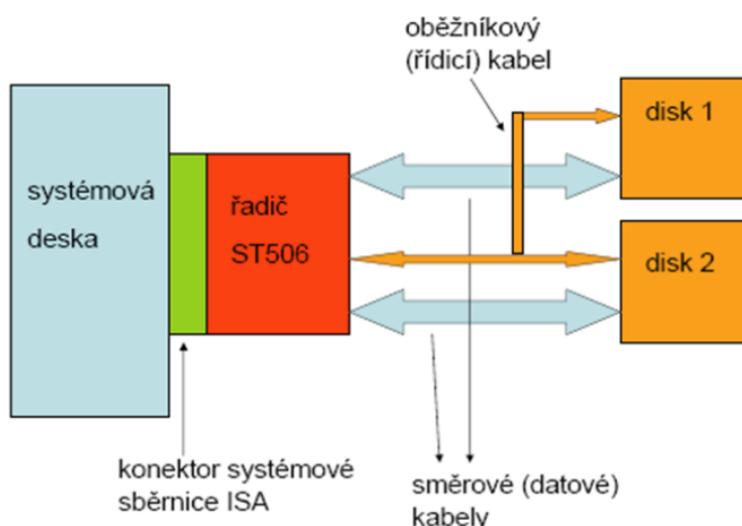
- Funkce UEFI, která umožňuje spouštět jen “certifikované” ovladače (oficiálně podepsané).
- Pro Windows 8 (64 bit) a novější.

## 11. Rozhraní pro připojování pevných disků, RAID, SSD

- Vývoj rozhraní pro připojování disků (ST-506, IDE, PATA, SATA, S.M.A.R.T.)
- Disková pole RAID
- SSD (srovnání s pevným diskem, SLC/MLC/TLC, TBW, WAF, Overprovisioning, TRIM, IOPS)
- Rozhraní pro připojování SSD

11.1. ST506 – nakreslit, jak to vypadalo, jaké tam byly linky (jde o to, že v disku není integrovaná řídící elektronika, disk je řízen externím řadičem).

- Používalo se u nejstarších pevných discích.
- Disk nemá žádnou vnitřní elektroniku, pouze mechanické části. Jeho řízení provádí externí řadič, který je zasunutý v konektoru ISA / EISA.
- Řadič má jeden řídící kabel (pro oba disky) a dva datové pro každý disk.



11.2. IDE – jaké se používaly kabely (40 žil, 80 žil a proč) a kolik k němu lze připojit zařízení. Co to je master / slave / single / cable select a jak se to konfigurovalo.

- V disku je již elektronika, nepotřebuje vnější řadič.
- Dnes spíše označení ATA (komunikační protokol) místo IDE (fyzického rozhraní disku).
- Kabel (někdy přezdíváný jako kšanda) má 40 žil, novější verze 80 žil (40 vodičů navíc vede GND), může se na něj připojit až 2 disky.

- IDE se ovládá z jižního můstku.
- Nepodporuje připojení CD-ROM, umí automaticky detektovat [C/H/S](#).
- Napájení pomocí Molexu (4 konektory).
  
- **Nastavitelné módy pro** disky (nastavuje se **jumpery** umístěnými na disku):
  - **Single** – nastavení disku, který je v počítači jediný.
  - **Master** – nastavení hlavního disku, který je v počítači zapojen spolu s druhým, který je podřízený.
  - **Slave** – nastavení druhého disku, který je v počítači spolu s hlavním masterem.
  - **Cable Select** – hlavní a podřízený disk jsou určeny tím, jak jsou připojeny ke kabelu (první je master, druhý slave).

11.3. Vývoj ATA1 až ATA6 - nejprve pouze jeden IDE kanál, potom dva. Od kdy lze připojit čtyři zařízení na dva kanály. Od kdy lze připojit CD-ROM. Jaká byla omezení max. kapacity disku. Jak postupně rostly přenosové rychlosti.

- **ATA-1** – 1986
  - Rychlosť 8.3 MB/s.
- **ATA-2** – 1996
  - EIDE.
  - Podporuje i CD-ROM.
  - Přenosová rychlosť až 16.7 MB/s, umožňuje [LBA](#) adresaci.
  - 2 IDE kanály, tedy až 4 zařízení.
- **ATA-3** – 1997
  - Podporuje S.M.A.R.T.
- **ATA-4** – 1998
  - Propojovací kabel s 80 vodiči.
  - CRC kontrolní součet přenosu.
  - Podpora řazení příkazů do front.
- **ATA-5** – 1999
  - Přenosová rychlosť až 66 MB/s.
- **ATA-6** (PATA) – 2000
  - Místo 28 bitové adresy (max 128 GB) podporuje 48 bitů (max 144 PB).

11.4. Co je to vlastně ATA? (sada příkazů pro univerzální ovládání všech disků).

- Standardizovaná sada příkazů.
- Příkazy **IN** a **OUT**.
- Všechny disky se ovládají stejně (bez ohledu na výrobce).

11.5. Co znamená zkratka IDE?

- **Integrated Device Electronic** – disk, který má integrovanou řídící elektroniku.

11.6. S.M.A.R.T – k čemu to je a co vše se typicky monitoruje.

- **Self Monitoring Analysys and Reporting Technology.**
- **Účel** – předcházet selhání disku a včas předejít havárii disky a ztrátě dat.
- **Typicky lze monitorovat:**
  - Počet zapnutí / vypnutí disku.
  - Počet hodin v provozu.
  - Počet vadných sektorů.
  - Teplota.

11.7. S-ATA – proč se přešlo na sériovou komunikaci. Jaké se používá kódování?

Jak je to rychlé? Jaké tam jsou signály a typ přenosu (je to sériové, asynchronní, linka je plně-duplexní ale komunikace je jen polo-duplexní, protože na disk nelze současně zapisovat i číst). Nezapomeňte zmínit, že už neexistuje master / slave / single, ale každé zařízení má vlastní kabel. Pro napájení zařízení už není Molex, ale SATA-power.

- Sériová ATA – data se posílají bit po bitu – vyšší rychlosť než PATA.
- 7 vodičů, 4 datové (**diferenciálně 2 páry**).
- **Kódování 8b10b.**
- Data jdou po jednom kabelu (neexistuje nic jako **master – slave**).
- **Rychlosť přenosu:**
  - **SATA1** – 150 MB/s.
  - **SATA2** – 300 MB/s.
  - **SATA3** – 600 MB/s.
- Pro napájení se používá SATA power.
- Konektor tvarovaný do písmena **L**, aby nešel umístit obráceně.

- Linka **Full Duplex**, ale používá se **Half Duplex** (na disk nejde zapisovat a zároveň z něho číst).

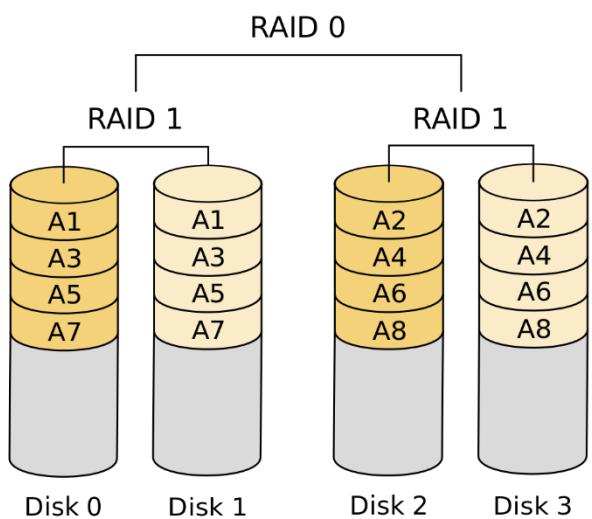
## 11.8. RAID – popsat všechny varianty (uvedené v prezentaci).

- **RAID = Redundant Array of Independent Disks.**
- Zapojení více disků může přinést zajímavé výhody – odolnost vůči chybám, ochrana dat proti jejich ztrátě (selhání disku) případně zvýšení výkonu.
- **RAID 0 – JBOD:**
  - Data se postupně ukládají na první disk, pak na druhý.
  - Vytvoří jeden velký logický disk.
  - Disky nejsou rovnoměrně opotřebovány, neposkytuje vyšší výkon.
- **RAID 0 – Stripping:**
  - Data jsou na disky ukládána prokládaně.
  - Zrychluje čtení i zápis, není ale odolný vůči chybám.
  - Vypadá jako jeden velký logický disk.
- **RAID 1:**
  - Data se zrcadlí mezi dvěma diskami.
  - **Redundance – počet nadbytečných dat – zde je vysoká.**
  - Zvyšujeme rychlosť čtení, ale ne zápisu.
- **RAID 1+0:**
  - Dvojice disků (RAID 0) vytvoří dvě podpole (RAID 1).
  - **Soubor rozdelen do podpolí, kde je zároveň zrcadlen.**
  - **Odolné proti výpadku jednoho disku z každého podpole.**
  - Čtyřnásobná rychlosť čtení, dvojnásobný zápis.
- **RAID 0+1:**
  - Dvě zrcadlená podpole RAID 0.
  - **Odolné proti výpadku různých disků z podpolí.**
  - Čtyřnásobná rychlosť čtení, dvojnásobný zápis.
- **RAID 3:**
  - Na jednom disku je uložena parita všech dat (paritní bit doplňuje počet jedniček na liché) – jeden disk není ale tak využíván – normálně se z něho nečte (paritu běžně nepotřebuju), jen se na něj ukládá.
  - Odolné vůči výpadku jednoho disku.
  - Soubor rozložen na jednotlivé disky.

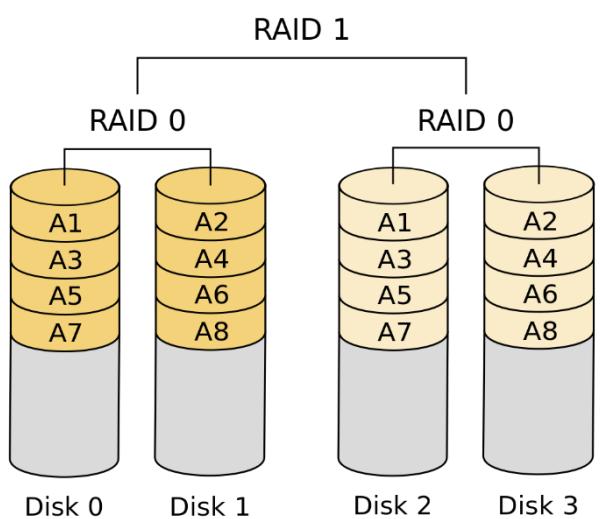
- **RAID 4:**
  - Stejné jako RAID 3, ale parita se počítá po blocích, ne po bitech.
- **RAID 5:**
  - Paritní data střídavě uložena na všech discích, ne jenom na jednom.
- **RAID 6:**
  - Dva paritní disky (různé typy), ukládání střídavě.

	<b>Snížení velikosti oproti celkové kapacitě</b>	<b>Zvýšení rychlosti - zápis</b>	<b>Zvýšení rychlosti - čtení</b>	<b>Odolnost vůči selhání (v závorce počet špatných disků)</b>	<b>Minimálně disků</b>
<b>RAID 0 - JBOD</b>	Ne	Ne	Ne	Ne	2
<b>RAID 0 – Stripping</b>	Ne	2x	2x	Ne	2
<b>RAID 1</b>	X / 2	Ne	2x	Ano (1)	2
<b>RAID 1+0</b>	X / 2	2x	4x	Ano (1+1)	4
<b>RAID 0+1</b>	X / 2	2x	4x	Ano (X+Y)	4
<b>RAID 3</b>	X - 1	Ano (ale menší)	Ano (X-1 krát)	Ano (1)	3
<b>RAID 4</b>	X - 1	Ano (ale menší)	Ano (X-1 krát)	Ano (1)	3
<b>RAID 5</b>	X - 1	Ne	Ano (X-1 krát)	Ano (1)	3
<b>RAID 6</b>	X - 2	Ne	Ano (X-2 krát)	Ano (2)	4

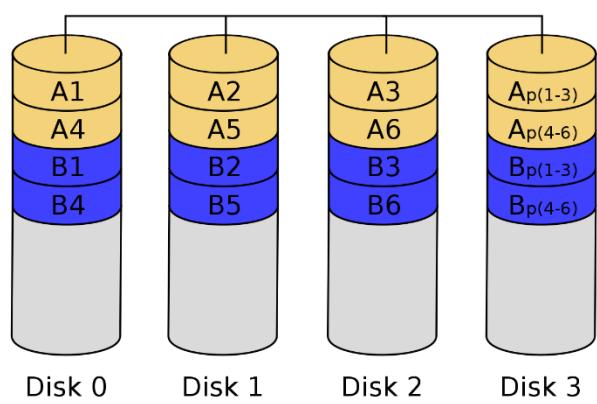
## RAID 1+0



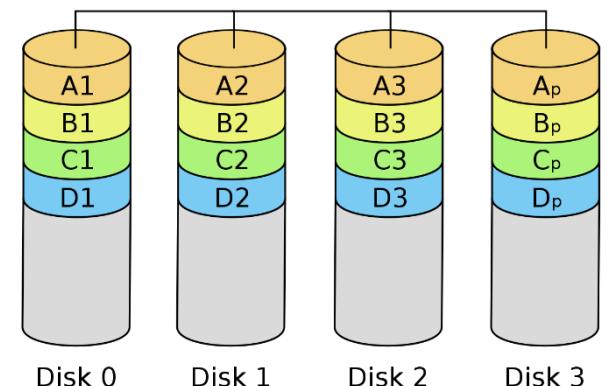
## RAID 0+1



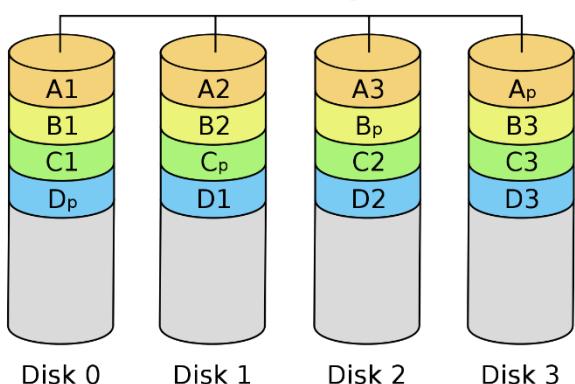
## RAID 3



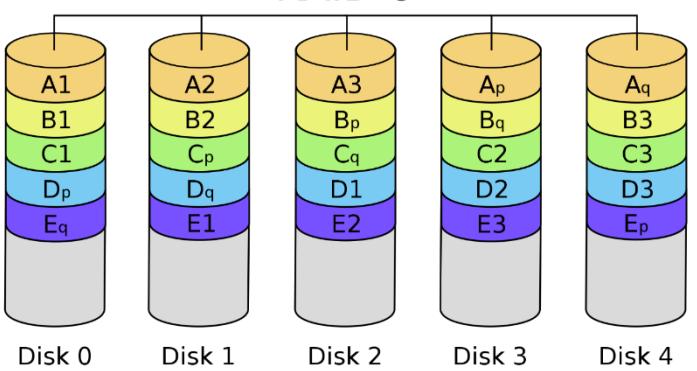
## RAID 4



## RAID 5



## RAID 6



11.9. Která varianta spolehlivost naopak snižuje?

- Raid 0 – Stripping.

11.10. Která varianta nezvýší rychlosť ani spolehlivost, pouze kapacitu?

- Raid 0 – JBOD.

11.11. Ukažte příklad s paritou na RAID3 - jak lze dopočítat ztracená data.

- Musím dopočítat jedničky tak, aby byl jejich počet lichý.

Disk 0	Disk 1	Disk 2	Paritní disk
0	1	?	1
0	0	?	1
1	0	?	0
0	1	?	0
1	1	?	1
1	1	?	0

Disk 0	Disk 1	Disk 2	Paritní disk
0	1	1	1
0	0	0	1
1	0	0	0
0	1	0	0
1	1	0	1
1	1	1	0

11.12. SSD – Zaměřte se na výhody a nevýhody oproti klasickému pevnému disku.

- **Výhody:**

- Neobsahuje pohyblivé části.
- Menší spotřeba (není potřeba v nečinnosti s ničím otáčet).
- Odolný vůči nárazům, tichý.
- Nižší hmotnost, menší rozměr.
- Nízká vybavovací doba, konstantní rychlosť čtení.
- Vysoká rychlosť čtení a zápisu.
- Není nutné defragmentovat.

- **Nevýhody:**

- Omezená životnost.
- Dražší než HDD.
- Nedostupné ve velkých velikostech (> 8 TB).
- Nutné použití funkce **TRIM** pro mazání (a pak nemožné obnovení).

### 11.13. Jakých kapacit, rychlostí a ceny dosahují moderní SSD?

- **Cena** – cca 1 - 2 Kč / GB.
- **Rychlosť** – PCI-E 3.0 – **3500 / 3000 MB/s**, PCI-E 4.0 – **7000 / 5000 MB/s**.
- **Kapacita** – běžně 512 GB – 4 TB.

### 11.14. Má defragmentace smysl na SSD?

- Ne, je to naopak nežádoucí kvůli zbytečnému opotřebování paměti a snižování životnosti.
- Navíc SSD má konstantní vybavovací dobu, tedy je to zbytečné.

### 11.15. Co je to TRIM? Jak je to s mazáním souborů na SSD?

- Pomocí funkce **TRIM** OS oznámí disku, jaké bloky paměti obsahují smazané soubory a SSD je připraví k odstranění (to proběhne, když SSD nic nedělá).
- Oproti HDD není možné blok dat jen tak přepsat – je nutné nejprve celý blok smazat (“vyresetovat”) a až pak je možné do něj zapisovat.

### 11.16. Co je to IOPS?

- Počet náhodného zápisu / čtení 4 kB souborů za sekundu.
- **Flash Disky** – 100 IOPS.
- **HDD** – 150 IOPS.
- **SSD** – 100 000 IOPS.

### 11.17. Vysvětlete proč by mohl být u úložiště na bázi NAND Flash problém se zápisem malých souborů – kde tento problém přetrívává a kde je vyřešen?

- Problém je, pokud máme příliš velké clustery (například 512 kB) a chceme změnit jen třeba 1 kB dat.
- Je nutné přečíst celých 512 kB dat (kam chceme obsah uložit), změnit 1 kB, blok smazat a znova zapsat celých 512 kB do tohoto bloku – **čeká se tedy na smazání** – protože nemá **TRIM**.
- Zápis mnoha malých souborů tak bude trvat na SD kartách či Flash Discích velice dlouho.
- Na moderních SSD je toto vyřešeno – stará data se změněnými se zapíší někam jinam a starý cluster se označí jako neplatný a později se smaže.

## 11.18. Jak je to s životností SSD? Co je to TBW? Co znamená Total NAND Writes a WAF?

- Životnost SSD se udává v **TBW** (Total Bytes Written) – znamená to, kolikrát jde obsah SSD přepsat. Pokud má disk 500 GB a životnost 200 TBW, obsah celého disku půjde přepsat 400x.
- **Total NAND Writes** – SSD přesouvá soubory, tak aby byly buňky rovnomořně opotřebovány. **Total NAND Writes** signalizuje, kolik dat už bylo celkem přesunuto po disku (tedy zapsáno - je to vždy větší než kolik bylo zapsáno uživatelem).
- **Total Host Writes** – kolik dat už zapsal uživatel.
- **WAF** je poměr mezi **Total NAND Writes** a **Total Host Writes**. U nového SSD bude 1.

## 11.19. Jaký význam má overprovisioning?

- U kvalitních SSD bývá velikost o něco vyšší díky kapacitě navíc.
- Tato kapacita navíc se nazývá **overprovisioning** a není pro uživatele viditelná.
- SSD tuto oblast využije během přesunů dat a pro efektivní práci s místem, i když je SSD plné – paměťové buňky můžou být tak stále rovnomořně opotřebovány.
- Zvýší se i výkon při zápisu, protože řadič má k dispozici více volných bloků.

## 11.20. Jaké jsou možnosti připojení SSD? (vysvětlit pojmy mSATA, NVMe, M.2, PCI-E...).

- **mSATA** – stejný výkon jak klasická SATA, jen menší konektor.
- **M.2 SATA** – moderní formát SSD s nižšími rozměry, ale komunikace stále probíhá přes SATA.
- **M.2 NVMe** – komunikace probíhá skrze sběrnici PCI-Express. Velmi rychlé.
- **PCI-Express** – stejně jako M.2 NVMe, ale využívají klasický PCI-E konektor.

## 12. Tiskárna

- Základní parametry tiskáren
- Používané barevné modely, dithering
- Technologie tisku, princip fungování tiskáren
- 3D tiskárny

12.1. Vysvětlete základní parametry tiskáren – DPI, rychlosť, zatížitelnosť, formát papíru, možnosť barevného tisku, životnosť tisku, náklady na tisk, možnosť pripojení.....

- **DPI – Dots Per Inch** – rozlišenie tisku v počte rozlišiteľných bodov na jeden palec (25.4 mm) – inkoustové – 4800 DPI, laserové 1200 DPI.
- **PPI – Pixel Per Inch** – počet vytvorených pixelov na jeden palec – pixel je na papieri tvoren z niekoľkých bodov (dots), ktoré dajú jednu farbu.
- **Rychlosť tisku** – PPM (Pages Per Minute) – počet vytlačených stránok za minuťu.
- **Formát papíru** – väčšinou A4, ale i menší či väčší (A3, A5, účtenky).
- **Možnosť barevného tisku** – barevné / černobílé.
- **Životnosť tiskárny** – rôzne pre jednotlivé komponenty – cca 30 000 stránok.
- **Životnosť výtisku** – termosublimačná má omezenú životnosť.
- **Náklady** – pri bežnom pokrytí stránky 5 % cca 0,15 Kč, kvalitné fotografie až 10 Kč.
- **Zatížitelnosť** – maximálny počet stránok za mesiac – 1500 stran / mesiac.
- **Možnosti pripojení** – LPT (paralelný port), USB, LAN, Bluetooth, Wi-Fi.

12.2. Jehličková tiskárna – nutné viedieť, že je hlučná (impaktná), tiskne pres levnou barvici pásku (nízké náklady na tisk), pomalá, nízké rozlišenie.

- Úderový tisk pomocí jehličiek – prenesenie farby z barvici pásky na papier.
- Výhody – levná barvici páška s dlouhou výdrží, nezáleží na kvalite papiera.
- Hlučná, pomalá, nízké rozlišenie (60 – 100 DPI).

12.3. Termální – jako příklad uvést účtenky, nízkou trvanlivost tisku, žádný toner ani pánska ani inkoust.

- **Termotiskárny = Tepelné tiskárny.**
- Používá se speciální papír, který po zahřátí zčerná (používá zahřáté jehličky které se přiblíží k papíru).
- Nehlučné, poměrně rychlé, doplňuje se jen papír, poměrně levné.
- Omezená životnost papíru – vybledne.

12.4. Termosublimační tisk – Co se tady přibližně děje? K čemu je vhodná?

- Pro tisk barevných fotografií.
- Tiskne se na speciální papír s barevnou fólií (Ribbon).
- Z barvové fólie, na které je nanесено barvivo v pevném skupenství, se po ohřevu nepohyblivou tepelnou hlavou uvolní barvivo a v plynném skupenství přechází do povrchové vrstvy papíru, kde opět změní skupenství na pevné.
- Ribbon obsahuje kromě standardních barev i čirý oddíl, který slouží jako zdroj ochranné vrstvy.

12.5. Inkoustová – popsat dvě základní technologie (se zahříváním inkoustu a piezo krystalem). Jaká rozlišení jsou dnes dosažitelná? Je vhodná pro barevný tisk?

- **Inkoustový tisk je vhodný pro barevný tisk.**
- Není příliš vhodné pro nárazový a nepravidelný tisk.
- Rozlišení až **4800 DPI** (nejvyšší ze všech tiskáren).
- Nehlučný tisk, nízká pořizovací cena.
- Kapičky inkoustu vystřelovány z mnoha trysek na papír – **1 kapka = 1 bod.**
- **Bubble-jet (termální)**
  - Kapičky inkoustu se rychle v komoře začnou vařit ( $200^{\circ} C$ ), čímž se zvýší jejich tlak a jsou vystřeleny na papír.
  - Inkoust má speciální chemické vlastnosti.

- **Ink-jet (piezoelektrické)**

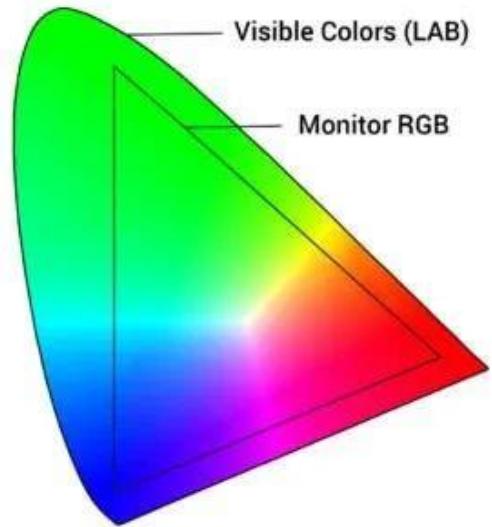
- Krystal, který funguje jako miniaturní pumpička, se při převedení napětí deformuje a zmenší prostor komůrky, ze které je vystřelen inkoust.
- Lze přesně řídit velikost kapičky (stupněm deformace).
- Není nutné příliš řešit, jaký inkoust je použit (nemusí mít přesné chemické složení).
- Dražší než Bubble-jet.

## 12.6. Barevný tisk – jaké barevné modely se používají (hlavně neříkejte RGB!).

Proč se používají doplňkové barvy? Proč se používá samostatná černá? Jak funguje dithering? Proč se jeden pixel musí tisknout pomocí mnoha bodů? Porovnejte PPI a DPI.

- **CMYK:**

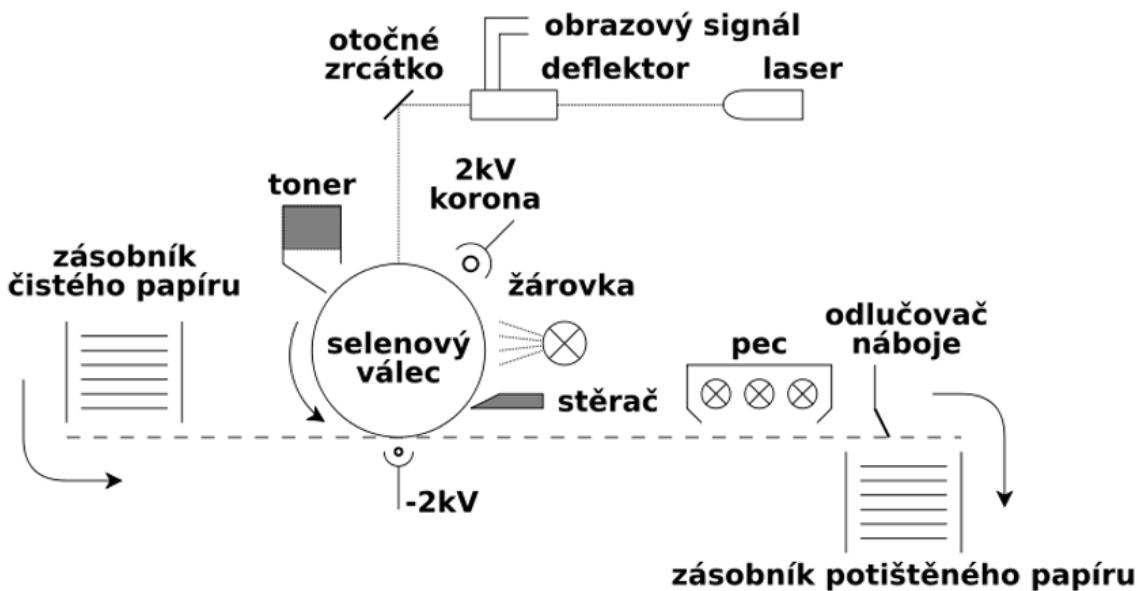
- **C = Cyan** (Azurová) – pohlcuje červenou.
- **M = Magenta** (Purpurová) – pohlcuje zelenou.
- **Y = Yellow** – pohlcuje modrou.
- **K = BlacK / Key** – pohlcuje celé spektrum.



- **C+M+Y** – netvoří dokonale černou, proto se používá extra černá.
- Některé lepší tiskárny (profesionálnější) mají ještě různé doplňkové barvy – například **CMYKLmLc** (Light Magenta, Light Cyan) – umožňují větší barevné spektrum.
- Klasický RGB pixel dokáže zobrazit 255 odstínů barvy, což tiskárna nijak nedokáže – je jen jedna Azurová / Purpurová / Žlutá / Černá.
- **Dithering** – střídavé nanášení různých barevných bodů je z dálky vnímané jako jedna barva.
- **Tisk probíhá na bílý papír a nanášením kapek barvy jas (odrazivost papíru) klesá** (subtraktivní míchání barev).
- **DPI – Dots Per Inch** – rozlišení tisku v počtu rozlišitelných bodů na jeden palec (25.4 mm).
- **PPI – Pixel Per Inch** – počet vytvořených pixelů na jeden palec – pixel je na papíře tvořen z několika bodů (dots), které dávají jednu barvu. Při barevném tisku menší než PDI (barevný bod je tvořen z několika bodů).

12.7. Laserová – popište na co svítí laser (hlavně neříkejte, že svítí na papír), co se děje na válci, co je to toner, jak se přenesе на papír, proč se zapéká. Jak funguje barevný laserový tisk a v čem je jeho problém? Jaké mají rozlišení (pozor, je nižší než u inkoustové tiskárny, přesto tisk vypadá lépe). Jaké jsou přibližné náklady na jednu stranu? Co je to LED tiskárna?

- **Toner = sypký barevný prášek.**
- Nabity **selenový válec** je na některých místech **osvícen laserovým paprskem**.
  - Deflektor podle dat tisku bud' propouští anebo nepropouští laser.
  - Otočné zrcátko rozprostírá paprsek po celé šířce válce.
- Na takto **osvícených místech z válce zmizí elektrický náboj**. Práškový toner se na tyto místa **uchytí** (na ostatních je odpuzován).
- Tonerový **prášek** je následně **přenesen** (obtisknut) **na papír** (ten má opačný náboj).
- Stěrač šetře zbytky toneru a žárovka odstraní náboj z předchozí fáze tisku.
- Nakonec je papír s tonerem **tlakově zafixován** a zapečen.



- U barevného tisku je problém, že **musí kolem válce projít 4x**. Musí být taktéž přesně zkalibrováno, aby se přesně trefil.
- U kvalitnějších tiskáren jsou 4 samostatné válce, takže je tisk rychlejší.
- **Rozlišení kolem 1200 DPI**, přesto je černobílý tisk kvalitnější než na inkoustové tiskárně, protože se nerozplýjí.

- **Dithering** je zde poměrně komplikovaný, protože se barvy nerozpijí, ale vzájemně překryjí.
  - **Náklady** na jednu stránku kolem 0.2 – 0.7 Kč za strana.
- 
- **LED tiskárna – místo laseru řada LED.**
  - Válec je fotocitlivý.
  - Levnější, rychlejší, ale pokud přestane fungovat jedna LED, tak v jejím řádku nepůjde tisknout. Je ale spolehlivější, protože obsahují méně pohyblivých prvků.

## 12.8. 3D tiskárna – Vyjmenujte několik základních variant a principů 3D tisku.

- **Stereolitografie (SLA)**
  - Nejstarší.
  - Tekutý fotopolymer je nality v nádrži, ve které je ponořená pohyblivá podložka. Místa na hladě kapaliny, která osvítí UV laser. Postupným zvedáním podložky „nahoru“ vznikají další vrstvy.
  - Velmi přesné a detailní.
  - Cena za litr fotopolymeru cca 800 - 1500 Kč.
- **DLP**
  - Podobné SLA, ale vytvrzuje celou vrstvu najednou pomocí LCD / OLED.
- **SLS**
  - Sintrování.
  - Prášek taven laserem – není třeba podpěr.
  - Tisk se postupně posouvá dolu.
- **3DCP**
  - Umí tisknout z cementu, betonu, sádry, roztaveného skla apod.
  - Některé „tiskárny“ dokážou ovládat i náklon či jsou na pohyblivé robotické konstrukci – tím pádem pak mohou vytisknout celé jednoduché domy.

12.9. FDM/FFF - popsat, jak to funguje. Jaké má základní parametry? Jaké materiály se používají? Co je to RepRap?

- **FDM / FFF je prakticky to samé.** FDM je ale licencované, a proto komunita RepRap (velký přispěvovatel Prusa Research) používá zkratku **FFF**.
- RepRap tvoří open hardware – tiskárny, které jsou schopné vytisknout část sami sebe. Klasický příklad – Průša MK3.
- Tisková hlava (hotend) taví plast ve formě pevné plastové struny (1.75 nebo 2.75 mm). Plast je **do hotendu tlačen pomocí extruderu**. Vytlačený materiál velmi rychle tuhne.
- Tisková hlava se pohybuje v ose **X**, podložka v ose **Y** a výška v ose **Z** pomocí krokového motoru. Jsou tedy nutné podpěry tisku.
- **Parametry – rozměr tiskové plochy (X, Y, Z), rychlosť tisku** (to je dost proměnlivé, záleží hodně na nastavení), **velikosť trysky** (dá se měnit, běžně 0.1 – 1 mm, nejpoužívanější 0.4 mm), **průměr filamentu** (plastové struny).
- **Speciální software** - tzv. Slicer (např.: Cura, PrusaSlicer) převede prostorový objekt (**.stl** / **.obj**) do tiskových instrukcí, podle kterých tiskárna tiskne daný model (tzv. **.gcode**).
- **Běžné Materiály:**
  - **PLA**
    - Nejrozšířenější mezi začátečníky, velmi jednoduchý na tisk.
    - Teplota tavení okolo 200 stupňů, nevyžaduje ani nahřívanou podložku.
    - Přírodně odbouratelný, ale časem degraduje.
  - **PETG**
    - Složitější mezi začátečníky, mírně složitější na tisk.
    - Teplota tavení okolo 230 stupňů, nutné nahřát podložku alespoň na 70 stupňů.
  - **ABS**
    - Vyrábí se z něj Lego.
    - Silný, odolný, ale při tavení se uvolňuje škodlivý zápach.
    - Teplota tavení okolo 210 stupňů, vhodné ale mít celou tiskárnu v uzavřené krabici kvůli teplotě.
  - **Nylon** – flexibilní materiál, tání kolem 260 stupňů.
  - **PVA** – rozpustné ve vodě.
- Cena za 1 kg **PLA / PETG / ABS** cca 300 – 700 Kč (záleží na výrobci a kvalitě).

## 13. Displeje

- Barevný model *RGB* a barevná hloubka, Gamma křivka
- Parametry LCD displejů (rozlišení, jas, kontrast, dynamický kontrast, pozorovací úhly, barevná hloubka, FRC dithering, odezva, input lag)
- Typy podsvícení
- Technologie TN, PVA/MVA, IPS, Quantum dot (srovnání vlastností)
- OLED
- e-ink
- dotykové displeje

### 13.1. Co je to rastrový displej? Co je to pixel? Co je to subpixel?

- **Rastrový displej** - tvoří obraz pomocí matice bodů, většina displejů. Vektorový displej by mohl být nějaký měřící přístroj, co kreslí linky.
- **Pixel** – 1 obrazový bod na displeji složený ze 3 subpixelů.
- **Subpixel** – 1 bod ze 3 v displeji – červený, zelený nebo modrý.

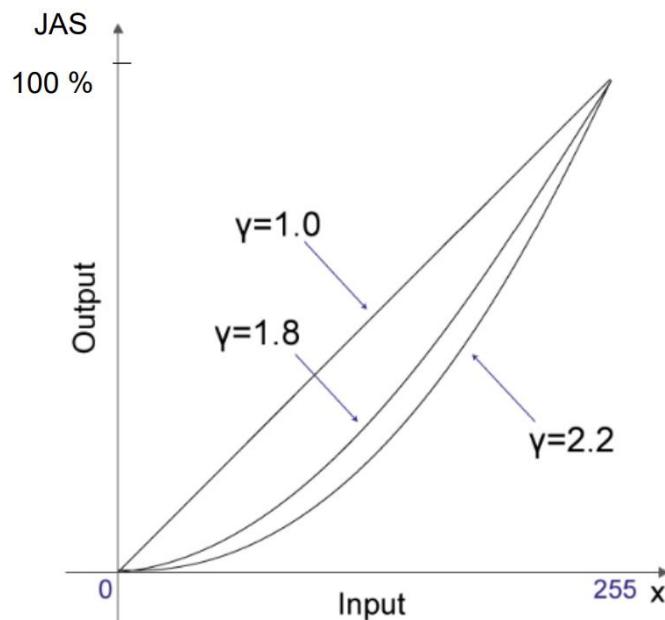
### 13.2. Jak funguje RGB model? (nemusíte znát vlnové délky barev a rozdíly citlivost oka) Kolik různých barev lze namíchat v truecoloru? Kolik intenzit každé základní barvy lze nastavit?

- Pomocí **3 základních barev** (červená, zelená a modrá) dokáže namíchat výslednou barvu.
- **Aditivní model** – barva vzniká součtem základních barev.
- V **True Color** lze namíchat  $256 * 256 * 256 = \textbf{16 777 216 barev}$ .
- V každé barvě lze namíchat **256 barev** (pokud počítáme i zhaslý subpixel).

### 13.3. Co je to Gamma křivka? Závislost čeho na čem vyjadřuje gamma křivka? Musíte jí umět nakreslit a vysvětlit o co jde.

- Pokud by se datové hodnotě = hodnota svítivosti pixelu, byl by rozdíl jedné hodnoty v tmavých odstínech velmi výrazný (4 vs 5 => 20 %) a naopak ve světlých mísťech velmi jemný (199 vs 200 => 0.5 %).
- Kvůli logaritmickému vnímání intenzity světla by nebylo vhodné, aby hodnota byla lineární. Byla proto zavedena tzv. **Gamma vyřazovací křivka**.
- Na **gamma křivce intenzita 50 % odpovídá hodnotě 186**.
- Nižší hodnoty jsou vyjádřeny s jemným krokem, vyšší hodnoty naopak více hrubě.

- Gamma funkce popisuje souvislost mezi datovou úrovní bodu a jeho vyzařovaným jasem.
- Optimální hodnota – **Gamma 2.2**.



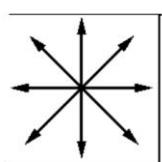
13.4. Jak vnímá oko jas? Jaká hodnota odpovídá číselně polovičnímu jasu?

Proč je popis tmavších odstínů odstupňován v jemnějších krocích a přesněji?

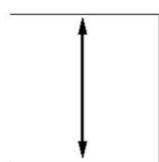
- **Logaritmicky**, ne lineárně. **Nevnímáme absolutní rozdíl** jasů, **ale jejich poměr**.
- Polovičnímu jasu odpovídá hodnota **186**.
- Nízké intenzity jsme schopni **rozeznávat v jemných krocích** (poznáme když se jas baterky zvýší z 5 W na 6 W), naopak **vysoké nerozeznáme** (reflektor z 1000 W na 1001 W).

13.5. Co znamená zkratka LCD. Co jsou to vlastně tekuté krystaly a co se s nimi dá dělat?

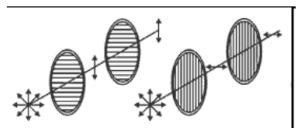
- **Světlo je elektromagnetické záření**, které **kmitá ve všech směrech**. Polarizační filtr propouští pouze světlo, jehož vlna kmitá v jednom směru.



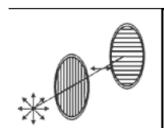
Nepolarizované světlo  
Vlny kmitají všemi směry



Polarizované světlo  
Vlny kmitají jediným směrem



Filtры ориентированы одинаково, свет проходит



Filtры ориентированы перпендикулярно друг другу, свет не проходит

- **Kapalné krystaly** = anizotropní kapalina = látka, která má různé vlastnosti v různých směrech.
- Molekuly kapaliny mají velmi protáhlý tvar a **polarizují světlo**.
- **Působením** vnějšího **elektrického pole** lze **měnit směr jejich natočení** (všechny molekuly **mají stejnou orientaci natočení**).
- Pomocí **ovlivnění stočení krystalů** v pixelu lze kontrolovat **množství procházejícího polarizovaného světla**.

#### LCD (Liquid Crystal Display) funguje takto:

- Za obrazovkou je umístěné podsvícení (světelné trubice nebo LED). Pomocí různých optických vláken a difuzerů se světlo rozptyluje a rozvádí po celé ploše obrazovky.
- Toto světlo prochází polarizačním filtrem – tak, aby mělo jeden směr.
- Následně toto polarizované světlo prochází **kapalnými krystaly**. Podle toho, **jak jsou natočené** (záleží na elektrickém poli) **je světlo dále pootočeno**.
- Následně je zde umístěn **další polarizační filtr**. Standardně, pokud by krystaly nic nedělaly (nebylo by přivedeno žádné napětí), tak žádné světlo nepropustí (je otočený vůči prvnímu o 90 stupňů – neplatí u TN). Pokud ale kapalné krystaly světlo pootočily, dokážou skrz třetí vrstvu projít.
- Barvu subpixelu zajišťuje pak průchod přes **barevný filtr** (celou dobu je světlo bílé).

13.6. Vysvětlete polarizaci světla a jak se dá regulovat množství polarizovaného světla dvěma polarizačními filtry.

- Viz předchozí bod.

13.7. Vysvětlete obecný princip všech LCD – podsvícení, polarizace, tekuté krystaly, barevné filtry na subpixelech...

- Viz předchozí bod.

13.8. Kontrast – jak se měří, co udává, v jakých jednotkách se udává (v žádných), v jakých hodnotách se obvykle pohybuje, proč je k ničemu dynamický kontrast.

- Vypočítána z **poměru svítivosti bílé a černé**.
  - Čím je **kontrast vyšší**, tím je **černá černější a bílá bělejší**.
  - Dnešní běžné hodnoty u průměrných LCD jsou 250:1, kvalitní LCD 1000:1.
  - Monitor, který by dokázal zobrazit **dokonale černou** (to LCD neumí) by měl **nekonečný kontrast**.
- 
- Monitor s **dynamickým kontrastem** může měnit i intenzitu bílého podsvícení.
  - Hodnota dynamického kontrastu je vypočítána jako **poměr nejsvětlejšího bodu s podsvícením na maximum a černého bodu s podsvícením na minimu**. Typické hodnoty jsou až 50000:1.
  - Tato hodnota je ovšem celkem zbytečná, protože u statických snímků (tedy až na video) vůbec tuto funkci nevyužijeme.

13.9. Doba odezvy – co udává, co znamená rise, fall, gray to gray, v jakých hodnotách se pohybuje, co to znamená vyrovnaná a nevyrovnaná doba odezvy, jak by se ideální měla udávat (3D graf nebo tabulka).

- Udává se v ms, dneska běžně udávaná výrobci do 5 ms.
- Původně chápáno jako čas, kdy se změní pixel z černé na bílou a zpět na černou.
- **Rise time** – přechod z 0 na 255 (rozsvícení).
- **Fall time** – přechod z 255 na 0 (zhasnutí).
- **Grey-to-grey** – například změna z tmavě šedé (32) do světlé šedé (128).
- **Vyšší hodnota = rychle měnící se obraz bude zobrazen rozmazaně**.
- **Odezvu displeje ale nemůžeme vyjádřit jedním číslem** – správně bychom měli mít k dispozici celou tabulku, kde budou zaznamenané různé změny (například všechny kombinace přechodů s hodnotami od 0 do 255 s posloupností 25).
- Výrobce přesto vezme nějakou hodnotu, která mu přijde optimální používat, i když ve skutečnosti u **většiny přechodů jsou výsledky daleko horší**. Většinou výrobce ani **neuvádí průměr všech hodnot**.

13.10. Pozorovací úhly – co se děje při pohledu shora, zdola, z boku, jaké hodnoty výrobci udávají, v čem je problém s udávanými hodnotami.

- Při pohledu **shora / zdola / z boku** je obraz na LCD obvykle **výrazně horší** – klesá kontrast a dochází viditelné barevné odchylce.
- Měl by se udávat úhel, do kterého je ještě **displej pozorovatelný bez větších odlišností**.
- Většina výrobců udává úhel spíše nepravdivé informace, tedy když například začne obraz přecházet do inverze (obraz tmavne a pak znova zesvětlá).

13.11. Jas – v jakých jednotkách se udává, v jakých hodnotách se pohybuje, jaký bude u TV a jaký u notebooku?

- Udává se v **cd/m<sup>2</sup>** nebo **nits** ( $1 \text{ cd/m}^2 = 1 \text{ nit}$ ).
- Měří se tak, že **všechny pixely zobrazí bílou barvu** a změří se **svítivost** monitoru. Změřená **svítivost** se pak **vydělí plochou displeje**.
- **Čím vyšší je jas**, tím **lepší** bude **obraz** na displeji například za použití venku **za slunečního světla**.
- Televize alespoň **500 cd/m<sup>2</sup>**.
- Počítačové monitory okolo **150-300 cd/m<sup>2</sup>**.

13.12. Input lag – co to je, kdy nám vadí vysoká hodnota, proč vzniká?

- Zpoždění na vstupu.
- Doba, jak dlouho zpracovává elektronika monitoru vstupní signál než ho převede na výstup.
- Důležité při hraní PC her.

13.13. Barevná hloubka – co je to Truecolor, které LCD displeje jí mají nejlepší a které nejhorší.

- V **True Color** lze namíchat  $256 * 256 * 256 = 16\,777\,216$  barev.
- Barvy se míchají pomocí kombinace tří barev – červené, zelené a modré.
- **IPS má velmi dobré barvy**, PVA a MVA také dobré.
- **TN má nejhorší barvy** – musí být zpracovány různě pomocí FRC ditheringu, jsou nevěrné, špatné.

13.14. Úhlopříčka a poměr stran – s jakými hodnotami se setkáme.

- **Úhlopříčka:**

- Udává se v palcích.
- **Počítačové monitory** – 19, 21, 24, 27, 32 palců.
- **Notebookové displeje** – 13, 15.6, 17 palce.
- **Televize** – nad 100 cm (40 palců).

- **Poměr stran:**

- Udává se jako velikost X:Y.
- **4:3, 5:4** – starší monitory.
- **16:9** – naprostá většina displejů.
- **21:9** – Ultra Wide.

13.15. Rozlišení – co je to nativní rozlišení, s jakými hodnotami se setkáme.

- **Nativní rozlišení** – počet pixelů v horizontálním x vertikálním směru.
- Obraz v jiném než nativním rozlišení musí být uvnitř LCD **přepočítán** (tím dochází k jeho **zhoršení**).
- Grafická karta by měla umět vygenerovat nativní rozlišení pro připojený monitor.
- **Běžné rozlišení:**
  - **1024 x 600** – mininotebooky, displeje k RPI apod.
  - **1024 x 768, 1280 x 800** – starší notebooky.
  - **1920 x 1080** – FullHD, běžné monitory
  - **2560 x 1440** – QHD (2K), lepší monitory.
  - **3840 x 2160** – 4K, kvalitní monitory.

13.16. LCD TN – jak dochází k regulaci jasu (co přesně se děje s polarizací světla při průchodu tekutými krystaly), proč mrtvý pixel svítí, jakou má barevnou hloubku. Popište, jak funguje FRC dithering. Jakou mají dobu odezvy.

- Pokud **není** přivedené **napětí**, pak tekuté krystaly umístěné nad sebou jsou **postupně otáčené (spirála)**, takže polarizované světlo **přivedou otočené** ve "správném" směru na druhý polarizační filtr - **subpixel svítí**.
- Pokud **je** přivedeno **napětí**, pak tekuté krystaly umístěné nad sebou jsou ve **stejné pozici**, polarizované **světlo vlastně nijak nemění** a druhý polarizační filtr **světlo zablokuje – subpixel nesvítí**.

- Postupným zvyšováním napětí můžeme postupně subpixel zhasínat.
- Pokud je na krystal přivedeno **nulové napětí (vada tranzistoru)**, světlo je krystaly propuštěno, a tak **pixel svítí**.
- **FRC Dithering**
  - **Spirála krystalů** TN displeje lze **pootočit pouze do 64** ( $2^6$ ) barev. To je problém, pokud bychom chtěli vytvořit True Color barvy.
  - **FRC Dithering** dokáže rychle blikat různými barvami, a tak vytvořit dojem True Color barev (například pro šedivou bude blikat mezi bílou a černou).
  - Ve skutečnosti ale displej umí zobrazit pouze jasy [0, 4, 8, 12, 16... 252](#).
  - Pokud displej používá FRC, je obraz neklidný.
- **Displej má velmi nízkou dobu odezvy**, což je společně s nízkou cenou jeho jediná výhoda.

[13.17. LCD xVA](#) – stačí vědět, že elektroda je jehlanová, dobrá barevná hloubka, nejlepší kontrast, ale nevyrovnaná doba odezvy.

- **MVA / PVA** – Vertical Alignent.
- Myšlenka orientovat molekuly tekutých krystalů vertikálně, v klidovém stavu připomíná jejich uspořádání stromeček.
- Elektrody mají specifický **jehlanový tvar**.
- V klidovém stavu **nepropouští světlo**.
- **Výhody** – zhasnuté vadné pixely (nesvítí), lepší pozorovací úhly, velmi dobrá černá (tedy i dobrý kontrast), 24 bitové barvy.
- **Nevýhody** – horší doba odezvy (navíc nevyrovnaná).

[13.18. LCD IPS](#) – stačí vědět, že je nejdražší, elektrody jsou umístěny z boku a nedíváme se skrz ně, nejlepší barevné podání (jaká je barevná hloubka?), vyrovnaná doba odezvy, nejlepší pozorovací úhly.

- Nejvěrnější podání barev (True Color).
- Elektrody se nacházejí po stranách pixelu.
- Tekuté krystaly v klidovém stavu souběžné se základní rovinou.
- **Výhody** – Skvělé barvy, skvělé pozorovací úhly, zhasnuté vadné pixely (nesvítí).
- **Nevýhody** – Nejdražší, horší černá (spíše šedá).

### 13.19. Podsvícení – CCFL x LED – výhody a nevýhody.

- Transmisivní podsvícení je CCFL nebo LED.
  - **CCFL** – v podstatě zářivka. Používá se stále méně, velká spotřeba, nižší životnost a postupem času klesá jas.
  - **LED** – energeticky efektivnější, má delší životnost a může produkovat lepší bílou.
- 
- Jak CCFL, tak LED mají horší čitelnost na přímém slunci a vyšší spotřebu energie.

### 13.20. Edge LED, Direct LED, RGB LED – které podsvícení umožňuje nastavení teploty bílé barvy? Které je nejčastější? Které umožňuje local dimming? Které je nejhorší pro kvalitu obrazu?

- **Edge LED**
  - Bílé LED na okraji panelu a světlo je rovnoměrně rozptýleno.
  - Malý počet LED, snížený výrobní ceny, panel je tenký, dnes nejpoužívanější, ale oproti ostatním nejhorší.
- **Direct LED**
  - Maticové rozložení bílých LED za displejem.
  - U jednotlivých LED lze snížit či zvýšit jejich jas, a tak uměle zvýšit kontrast (lokální stmívání = **local dimming**).
- **RGB LED**
  - 4 skupiny LED (červená, modrá a dvě zelené) za displejem v matici (stejně jako **Direct LED**).
  - LED svítí přesně z těch vlnových délek, které potřebujeme.
  - Poměrem jasu modrých a červených LED lze nastavit „teplotu“ bílé barvy.
  - Nejkvalitnější, nejdražší, nejméně obvyklý.

### 13.21. Co je to clouding? Proč vzniká? Trpí tím i jiné displeje kromě LCD?

- **Clouding** – při **tma**vých scénách můžeme vidět **světlejší místa** a ve **světlé scéně tmavší skvrny**.
- Způsobeno **nerovnoměrným podsvícením LCD** – různá svítivost LED či CCFL trubic, působením tepla, vlastní váhy nebo vůlí v uložení.
- Objevuje se jen u LED.
- Nejlépe viditelné při pohybu velké jednobarevné plochy přes obrazovku.

13.22. Vysvětlit, že není rozdíl mezi LCD TV a LED TV.

- LED displej – používá pro podsvícení **LED místo CCFL trubice**.
- Výhoda jen v **nižším příkonu** a **delší životnosti** (kvůli použití LED).
- Jinak ale platí, že **LCD TV = LED TV**.

13.23. QLED, Quantum dot – v čem je rozdíl oproti klasickým LCD displejům?

Které z parametrů jsou u nich výrazně lepší? Porovnejte "obarvení" subpixelu načerveno, modro a zeleno u klasického LCD a u QLED. Kolik světla projde klasickým barevným filtrem v subpixelu a o kolik je to lepší u QLED? Jak vypadá podsvícení QLED displejů? Jaká je přesná role nanočastic?

- **Podsvícení displeje je ultrafialové.**
  - UV záření má krátkou vlnovou délku, fotony mají vysokou energii.
  - V subpixelech jsou **nanočástice**, které **pohlcují UV záření**. Následně tuto získanou energii nanočástice **vyzáří jako světlo s jinou vlnovou délkou**, která **závisí na velikosti nanočastic** (dokonale červenou / zelenou / modrou).
  - **Výhody – dokonalé barvy** (přesná vlnová délka), **vynikající jas** (světlo není zachycováno barevnými filtry), **nižší spotřeba** (dvojnásobný jas při stejné spotřebě), **vysoký kontrast** (unikající světlo není bílé, ale ultrafialové, takže není vidět).
  - **Nevýhody** – dražší než LCD.
- 
- U klasického LCD se minimálně 2/3 podsvícení ztrácejí v barevných filtroch, v QLED prakticky žádné – veškeré světlo je "obarveno".

13.24. Jak funguje OLED a SOLED. Jak se u OLED reguluje jas pixelu? Výhody OLED oproti ostatním displejům (svítivost, tenký, ohebný, barvy, kontrast, odezva....).

- **OLED = Organic Light Emitting Diode.**
- **Organický materiál** při přívodu napětí (2 – 10 V) emituje světlo o určité barvě.
- Daleko jednodušší než LCD.
- Každý obrazový bod samostatným zdrojem světla červené / zelené / modré.
- Černá dokonale černá – bod prostě nesvítí.
- Doba odezvy extrémně nízká (ani se neudává).

- Vadné body nesvítí.
- Dokonalé pozorovací úhly, velmi vysoká svítivost, jakýkoliv rozměr (pro mobilní telefony i reklamní panely).
- **Teoreticky možné zobrazit nekonečný počet barev (záleží na hodnotě napětí).**
- Velmi tenké, ohebné, odolné, nízká spotřeba, clouding neexistuje.
- **SOLED** – jednotlivé tři subpixely jsou umístěny nad sebou, a tedy i při pohledu z blízka bude pixel vypadat jako čtvereček s libovolnou varvou.

### 13.25. Nevýhody OLED displejů?

- **Omezená životnost** - hlavně kvůli modré barvě.
- **Retence** – pamatuje si jas dlouhodobě zobrazených snímků. Možnost i trvalé retence.
- **Nerovnoměrné zahřívání** – zhasnutý pixel nevydává žádné teplo.

### 13.26. Které typy displejů mohou mít mrtvé pixely? Které mohou mít svítící mrtvé pixely?

- **Mají všechny typy** - LCD, QLED i OLED.
- Mrtvé pixely **LCD TN svítí**.

### 13.27. E-ink – co to je, kde se to používá, v čem je to lepší než LCD? K čemu se to naopak nehodí. Nebudeme se bavit o čtečkách knih, ale půjde hlavně o princip fungování toho displeje.

- **E-ink – elektronický inkoust.**
- **Výhoda** – potřebuje energii jen pro **překreslení obrazu** – tedy energeticky velmi málo náročné.
- **Výhody – displej nesvítí, funguje jako papír** (je reflexivní), **výborně čitelný ze všech stran i na slunci, málo zatěžuje očí**.
- **Nevýhody – pomalá změna obrazu** (maximálně 10 snímků za sekundu), **ve tmě nesvítí, většinou monochromatický** (nebo slabá barevná hloubka), **ghosting** (při změně stránky je nutné pixely nastavit na černou a pak na bílou).
- **Využití** – elektronické knihy, stavové displeje (například pro teplotu).

- **Fungování:**

- Základní části **mikrokapsle**.
- Ta obsahuje tekutinu, kde plavou **černé** (záporně nabité) a **bílé** (kladně nabité) **částice**.
- Pokud je **mikrokapsle vystavena** působení **záporného elektrického pole**, černé částice se přesunou do horní části mikrokapsle a stanou se viditelnými na displeji.
- Bílé části naopak putují dolu.

13.28. Co je to ghosting, retence, vypalování obrazu a kde se s tím setkáme?

- **Ghosting** – “duch” po minulých stránkách (zůstává zobrazený předešlý snímek) – **E-ink**.
- **Retence** – pamatují si jas dlouhodobě zobrazených snímků – **OLED**.
- **Vypalování obrazu** – snímek je trvale vypálen do obrazovky – **OLED**, (**Plasma**).

13.29. Jak fungují dotykové displeje? Jaké principy znáte a jaké mají výhody a nevýhody? Je třeba znát všechny čtyři!

- **Rezistivní displej:**

- Povrh displeje je **pokryt pružnou membránou** s průhlednou kovovou vrstvou, pod níž je ještě další vodivá vrstva obsahující miniaturní mřížku.
- Při dotyku stylusem nebo prstem se mírně prohne a spojí se tak dvě pod sebou.
- **Výhody** – klepání na displej, lze ovládat v rukavicích.
- **Nevýhody** – špatné určení místa doteku, nevhodné pro posouvání, špatné podání barev.

- **Kapacitní displej:**

- **Dvě průhledné vodivé vrstvy** nanesené na skle – **tvoří kondenzátor**.
- Na jedné vrstvě jsou v **rozích čtyři elektrody**, na které je přivedeno malé napětí.
- **Při dotyku displeje dojde ke vzniku parazitní kapacity** mezi vrchní průhlednou elektrodou a prstem. Tím se **zvětší odběr proudu z elektrod**. Podle odběru jednotlivých elektrod je pak vypočítáno, kde došlo k dotyku.
- **Výhody** – při dotyku se neprohýbá.
- **Nevýhody** – nefunguje v rukavicích.

- **Infračervený panel:**
  - Opatřen rámečkem s IR diodami a fotorezistory.
  - Vytvořena **hustá síť infračervených paprsků**.
  - **Výhody** – displej není ničím překryt.
  - **Nevýhody** – nedetektuje menší předměty, menší přesnost.
- **Displej s povrchovou akustickou vlnou (SAW):**
  - V rozích umístěny **vysílače a přijímače ultrazvukového signálu** (cca 5 MHz).
  - Ultrazvukové vlny se šíří napříč plochou displeje.
  - **Po vložení předmětu** do vlnového pole se **šíření vln změní** a řídící jednotka tak podle vyslaných a poslaných signálů **vyhodnotí polohu vložené překážky**.
  - **Výhody** – velmi přesné, displej není ničím překryt.
  - **Nevýhody** – velmi drahé (využitelné pouze do průmyslové a profesionální elektroniky).

## 14. Snímače obrazu

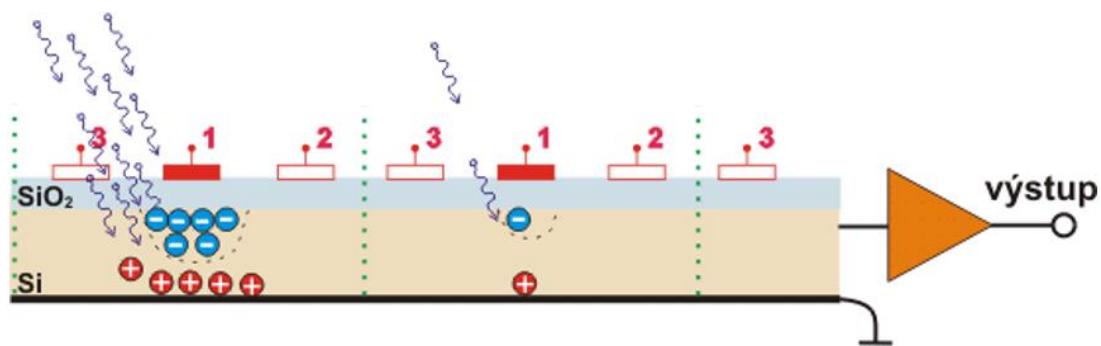
- *CCD, CMOS snímač (princip, vlastnosti, parametry, použití)*
- *Obrazové vady*
- *Digitální fotoaparát, objektiv, clona expozice*
- *Scanner*
- *Čárový a QR kód*
- *3D scan*

14.1. Popište, jak funguje CCD snímač. Bylo dobré vědět, kdy byl vynalezen a kde všude se používal.

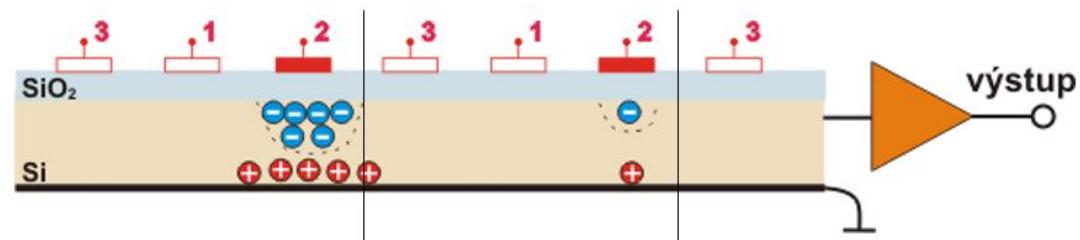
- **CCD = Charge-Coupled Device (zařízení s vázanými náboji).**
- **Cíl** – převést obraz na obrazový signál.
- Vynalezen 1969 v Bellových laboratořích, v televizní technice od roku 1975, ve fotoaparátech se používá od roku 1990.

### Využívá fotoelektrický jev:

- **Foton** (je to vlastně světlo) **dopadajícího světla** při nárazu **do atomu senzoru** dokáže převést některý z jeho elektronů ze základního do excitovaného stavu.
- **Tam, kde dopadne více fotonů, bude více excitovaných elektronů.**



- Postupně budeme zvyšovat napětí na elektrodách na hranici pixelů a **náboj převedeme k okraji snímače – výstupu**.



14.2. V čem spočívá rozdíl mezi lineárním a plošným snímačem?

- **Lineární snímač:**
  - Snímá jen jednu řadu pixelů.
  - Pro fax, scanner, čtečky čárových kódů.
- **Plošné CCD:**
  - Spojení několika lineárních CCD na jediném čipu.
  - Obraz na CCD je promítán **objektivem**.
  - **Na konci** všech lineárních CCD snímačů **vstupuje do dalšího lineárního snímače**, který slouží jako **posunovač do výstupu**.
  - Existují i snímač, který má **rychlý plošný snímač na konci všech řádků** a dokáže měřit hodnoty současně – je ovšem **dražší** a musí být **přesně zkalibrován**.
  - Pro fotoaparáty, kamery.

14.3. Základní parametry snímače – rozměr (jaké jsou běžné), poměr stran, počet pixelů (jaké jsou dnes běžné), přesnost AD převodu, dynamický rozsah, šum (proč vzniká).

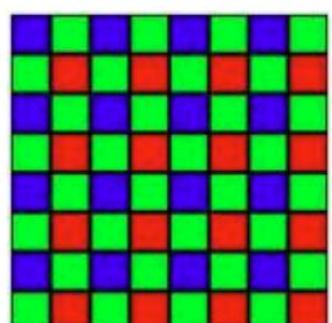
- **Rozměr**
  - Uzávěrka je udávána jako úhlopříčka v palcích nebo zlomcích palce ([1/3](#), [1/2](#) apod.)
- **Poměr stran**
  - Typicky [4:3](#), [3:2](#) (papír), [16:9](#).
- **Počet pixelů**
  - Uzávěrka je udávána v **Megapixezech** ([1 000 000 px = 1 Mpx](#)).
  - Detailní rozlišení je udáváno jako počet pixelů **X : Y**.
  - Ze snímače s vysokým rozlišením získáme “hodně čísel“, záleží také ale na jejich “kvalitě“.
- **Přesnost AD převodu**
  - Výstup CCD je analogový (elektrický náboj).
  - Změřená hodnota náboje je následovně převedena na digitální informaci.
  - 8 bitová informace je ale příliš malá (je nutné pak provést transformaci do RGB).
  - Digitální zrcadlovky proto používají [16](#) bitovou informaci, kvalitní fotoaparáty [14](#) bitů a levné snímače [8 / 10](#) bitové.

- **Dynamický rozsah**
  - **Interval mezi minimálním a maximálním množstvím dopadajícího světla**, který dokáže **snímač převádět na efektivní datové hodnoty**.
  - Snímač s malým dynamickým rozsahem nevidí kresbu v **tmavých plochách** – vše pod určitý jas je pro něj **černá**. Taktéž ale nevidí kresbu v **jasných plochách**, vše nad určitý jas je pro něj **bílá**.
- **Šum**
  - Vzniká především kvůli **tepelné energii**, která je schopna **excitovat elektrony bez jakéhokoliv působení dopadajících fotonů**.
  - Taktéž může být způsobem **vlivem elektromagnetického pole** nebo **ionizujícího záření**.
  - V každé buňce vzniká jiný počet, proto nejde systematicky odstranit.
  - Nejvíce je šum patrný na tmavých plochách.
- **Citlivost**
  - Na jaké **nej slabší množství světla** je **snímač schopen reagovat**.
  - Existuje i maximum, pokud bude překročeno, zůstane hodnota pixelu na své maximální úrovni.

14.4. Bayerova maska – jak vypadá (obrázek !), jak se barva každého pixelu ve vygenerovaném snímku získá z několika sousedních.

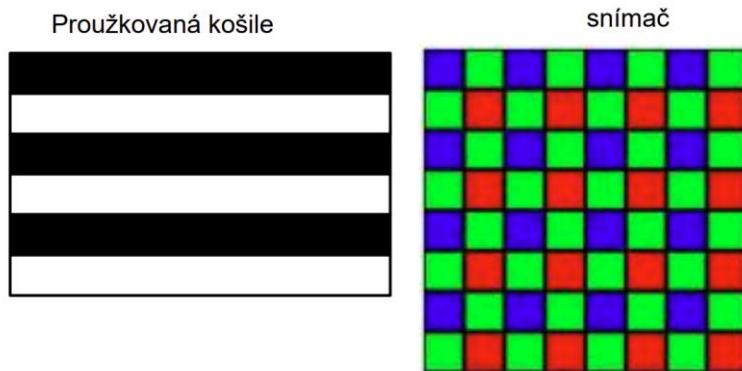
- CCD snímač obsahuje **pixely**, ale ty vidí **pouze intenzitu světla** – tedy ne barvu.
- Před CCD snímač dámě **barevný filtr** (Bayerovu masku). Každý pixel je pak citlivý na jinou základní barvu, ostatní nejsou do něj propuštěny.
- Lidské **oko** je **více citlivé na zelenou barvu**, proto je zelená propuštěna více oproti modré a červené.

- Pro vytvoření **jednoho pixelu výsledného obrazu** je nutné zkombinovat hodnoty **získané sousedními pixely** - dvěma zelenými, jedním červeným a jedním modrým.
- 12 Mpx snímač bude složen z 3 milionů pixelů citlivých na červené světlo, 3 na modré světlo a 6 na zelené. Z těchto informací se složí pak výsledný 12 Mpx snímek.
- **1 pixel snímače s Bajerovou maskou není co jeden pixel na displeji.**



14.5. Jak vzniká moiré? Jak funguje 3CCD snímač a proč zde nevzniká moiré?  
Proč má moiré červenou a modrou barvu?

- **Světlo dopadá pouze z bílých proužků.** Černé proužky světlo neodrážejí.
- Proužky mají takovou rozteč, že **dopadají přesně jen na červeně a zeleně citlivé pixely**. Košile tedy bude působit **načervenale** (případně namodrale).
- Protože **pixely citlivé na zelenou barvu** jsou v každém sloupci / řádku, nemůže objekt nikdy působit nazelenale.



- **3 CCD**
  - Přicházející světlo z objektivu je rozděleno **do 3 základních barev pomocí optických hranolů**.
  - Pro každou barvu je udělaný **vlastní snímač**, který následně **pracuje jen s jednou barvou**. Výsledný **snímek** je pak **zkombinován**.
  - Použití pouze v kamerách (je velké).

14.6. Vysvětlit, proč větší počet megapixelů nemusí znamenat více detailů.

- Aby se více pixelů vešlo na malý snímač, musejí být **velmi malé**.
- Tím pádem mají **špatnou citlivost**. Obraz je **degradován šumem**.
- Vyniknou při tom i **vady objektivu**.

14.7. Vysvětlit pojem hustota pixelů.

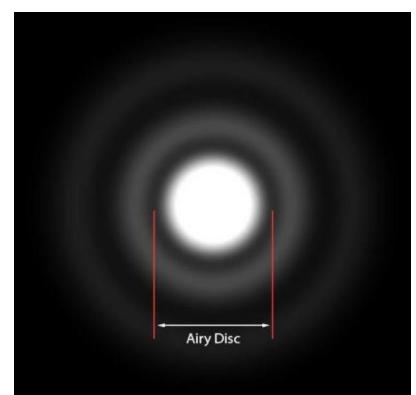
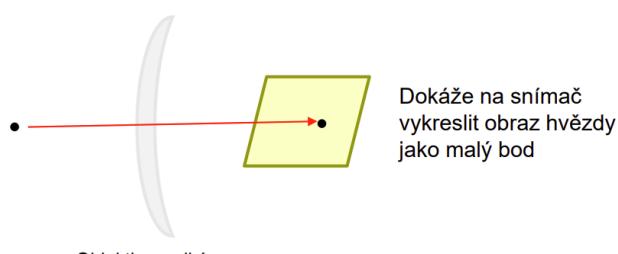
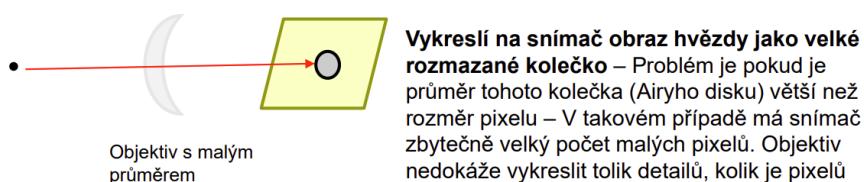
- **Pixel Density.**
- **Počet MPx/cm<sup>2</sup>.**
- Důležité hledět na **poměr rozlišení a velikost čipu**.
- Pokud je **hustota pixelů vysoká**, optika nezvládá tak **jemnou kresbu – sousední pixely vidí v podstatě totéž**.

- **Hvězdářské dalekohledy** – 0.5 až 1 MPx cm<sup>2</sup>.
- **Zrcadlovky** – 1 až 3 MPx cm<sup>2</sup>.
- **Nejkvalitnější kompakty** – 15 až 25 MPx cm<sup>2</sup>.
- **Průměrné kompakty** – 25 až 45 MPx cm<sup>2</sup>.
- **Mobilní telefony** – přes 50 MPx cm<sup>2</sup>.

14.8. Co je to hloubka ostrosti? Jak závisí rozlišovací schopnost na průměru objektivu? Jak funguje clona? Kdy dochází k difrakci, co je to Airyho disk a kdy lze pomocí clony pracovat s hloubkou ostrosti?

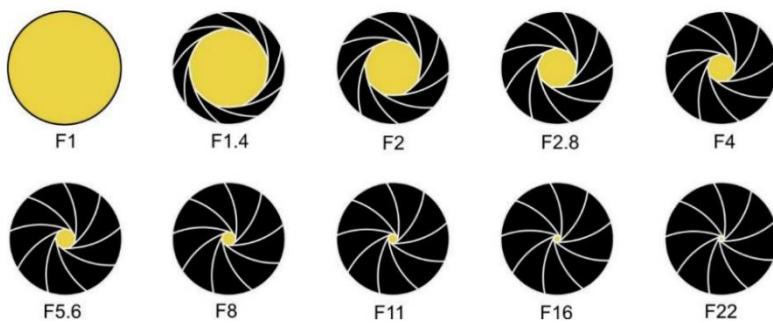
- **Difrakce:**

- **Obraz na snímač je promítnut objektivem.**
- Objektiv neumí vykreslovat nekonečně malé body (kvůli tomu vzniká difrakce a tvoří se airyho disky).
- **Světelný bod** bude i tím nejkvalitnějším objektivem **vykreslen** jen jako **rozmazané kolečko s určitým průměrem**.
- **Rozmazané kolečko**, které objektiv vykreslí (**místo dokonalého malého bodu**) se nazývá **Airyho disk** – je to **vlastně nejmenší bod, který dokáže objektiv vykreslit**.
- **Vykreslit detailní obraz na miniaturní pixely snímače umí lépe objektiv s velkým průměrem než objektiv s malým průměrem.**



- **Clona:**

- Objektiv jde zaclonit a tím **snížit množství zachyceného světla**.
- Tím se ovšem zhorší jeho **rozlišovací schopnosti – vykresluje větší Airyho disky**, kresba tedy nebude tak detailní.
- **Příliš zacloněný objektiv** nedokáže na snímač vykreslovat body menší, než jaká je velikost pixelů.



- **Hloubka ostrosti:**

- Lze pozorovat pouze u fotografií, kde je **pozadí nebo popředí**.
- **Čím vyšší clona, tím větší hloubka ostrosti** (bude dobře vykresleno pozadí a nebude rozmazané).
- Čím větší hloubka ostrosti, tím **hlouběji před a za ostřeným objektem bude obraz ostrý**.
- Pokud má fotoaparát **vysokou hustotu pixelů** (mobilní telefony), vyšší clona nezpůsobí **zvýšení hloubky ostrosti**, ale celkovou ztrátu ostrosti (Airyho disky budou větší než pixely). **Tedy místo toho, abychom dostali fotku, kde jsou například vidět v pozadí hory, bude celkově fotka působit neostře.**

14.9. Pozor, neplést si hloubku ostrosti a rozlišovací schopnost. Když zvýšíte clonu, hloubka ostrosti se zlepší, ale rozlišovací schopnost se zhorší a bylo by dobré vědět, v čem je tedy vlastně ten rozdíl.

- Zvýším clonu – objektiv je více zacloněný.
- **Hloubka ostrosti se zvětší** – budou **pozorovatelné detaily** v pozadí či popředí.
- **Airyho disky budou větší**, protože **rozlišovací schopnost** bude horší (body budou větší než pixel).

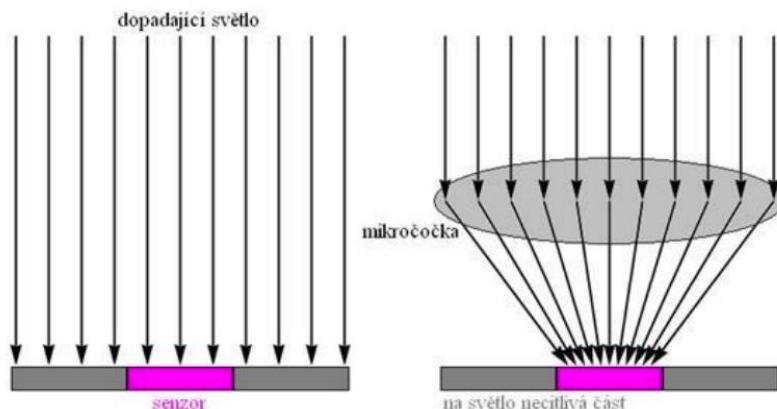
#### 14.10. Proč jsou lepší větší snímače?

- Na velký snímač lze snáze vykreslit objektivem kvalitní kresbu.
- Čím větší je každý jednotlivý pixel, tím víc na něj dopadne světla a tím kvalitnější informaci poskytne.
- Fotoaparát s polovičním rozměrem snímače také pořídí fotografii s dvojnásobným zvětšením než fotoaparát se stejným objektivem ale větším snímačem.

#### 14.11. Co je to vinětace a kdy a proč k ní dochází? Jak fungují mikročočky?

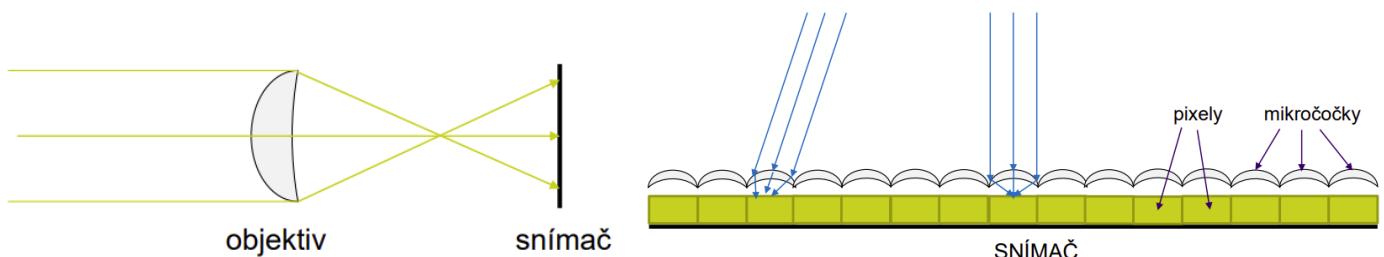
- **Mikročočky:**

- Jednotlivé senzory na snímače spolu nesousedí zcela těsně a jsou mezi nimi mezery. Tyto mezery pohlcují nebo odráží světlo.
- **Světlo, které by jinak dopadlo mimo pixely, je směrováno do středu buňky.**
- **Každý pixel má svou vlastní mikročočku.**



- **Vinětace:**

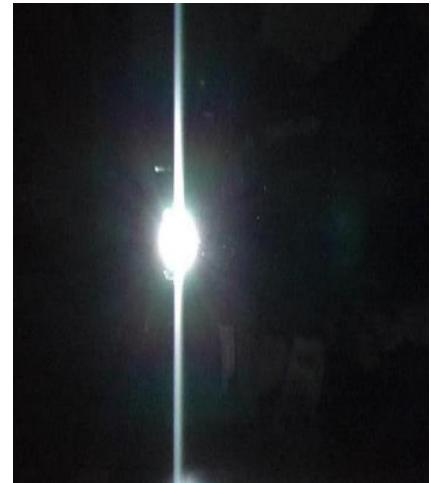
- Problém – z objektivu vychází paprsek na krajích šikmo, tedy mikročočky nedokážou paprsek správně nasměrovat do středu pixelu.



#### 14.12. Co je to smearing a blooming?

- **Smearing:**

- Během **čtení náboje** stále na snímač **dopadají nové fotony** a pixely pokračují ve sbírání světla.
- K nábojům z fáze expozice se **přičítají nové excitované elektrony**.
- **Způsobují velmi intenzivní objekty** (Slunce) a pouze u fotoaparát **bez mechanické uzávěrky**.



- **Blooming:**

- Na některé pixely dopadne tolik fotonů, že se překročí jejich kapacita udržet excitované elektrony a přeteče do sousedních pixelů.



#### 14.13. Co se děje při zvyšování citlivosti (ISO)? Proč se zvýší šum?

- **ISO 100** – základní nastavení.
- **ISO 200** – půjde snížit čas expozice na polovinu a **náboj na výstupu se znásobí**. Znásobí se ovšem taktéž šum.

#### 14.14. Jak lze realizovat noční vidění? Na jaké neviditelné vlnové délky jsou snímače běžně dobře citlivé?

- CCD snímače jsou citlivé i na fotony **UV** a **IR** záření.
- Pomocí IR LED je možné osvítit plochu či místnost, kterou lidské oko uvidí jako beze světla.
- **UV** – 100 - 400 nm.
- **IR** – 800 - 1 000 000 nm.

14.15. Co je to ohnisková vzdálenost a ekvivalentní ohnisková vzdálenost?

- **Ohnisková vzdálenost:**

- **Čím kratší:**

- Objektiv je širokoúhlejší.
    - Objektiv je kratší.
    - Má větší zorné pole.
    - **Menší přiblížení.**
    - Na snímač vykreslí menší obraz předmětu.

- **Čím větší:**

- Objektiv je užší.
    - Objektiv je delší.
    - **Větší přiblížení.**
    - Na snímač vykreslí větší obraz předmětu.

- **$F = f / d$**

- **F** = clonové číslo.
  - **f** = ohnisková vzdálenost objektivu.
  - **D** = průměr objektivu.



- **Ekvivalentní ohnisková vzdálenost:**

- Přepočítaná ohnisková vzdálenost na full-frame snímač (36 x 24 mm).
  - Pokud bude mít snímač fotoaparátu 18 x 12 mm a ohniskovou vzdálenost 25 mm, pak ekvivalentní ohnisková vzdálenost bude 50 mm.

#### 14.16. Co je to crop factor?

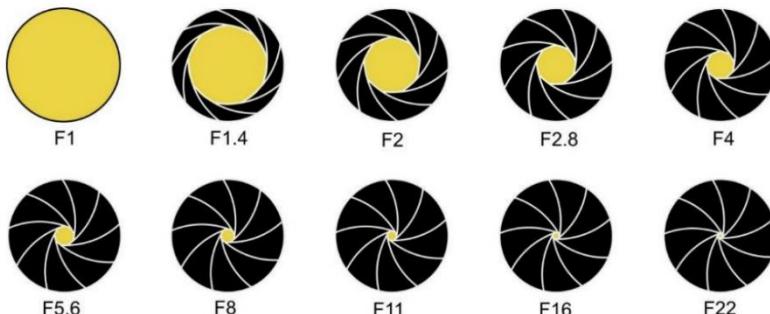
- Crop factor udává, **kolikrát je strana snímače kratší než šířka Full Frame** snímače (36 x 24 mm).

#### 14.17. Jaké běžné ohniskové vzdálenosti mají používané objektivy? Jakou ohniskovou vzdálenost má teleobjektiv a jakou širokoúhlý objektiv?

- **Teleobjektiv** – 100 až 500 mm.
- **Širokoúhlý** – do 25 mm.
- **Běžné objektivy** – 50 – 150 mm.

#### 14.18. Co je to clona? K čemu je dobrá clona? Je lepší menší nebo vyšší clonové číslo? Jak závisí na cloně množství zachyceného světla?

- **Clona:**
  - Objektiv jde zaclonit a tím **snížit množství zachyceného** světla.
  - Tím se ovšem zhorší jeho **rozlišovací schopnosti – vykresluje větší Airyho disky**, kresba tedy nebude tak detailní.
  - **Naopak zase čím vyšší clona, tím větší hloubka ostrosti** (bude dobře vykresleno pozadí a nebude rozmazané).



#### 14.19. Co je to expozice? K čemu je lepší delší a k čemu kratší čas expozice?

- Udává se ve zlomcích sekundy (například.: 1/250 s).
- Je to vlastně to, jak dlouho sbírá sběr fotonů.

- **Delší čas expozice:**
  - Snímač zachytí více fotonů a proto lze pořídit fotografii i při slabém osvětlení.
  - Více se projeví šum.
  - Fotoaparát i snímané předměty musí zůstat v klidu.
- **Kratší čas expozice:**
  - Méně šumu.
  - Ostřejší výsledný snímek (je zachyceno méně pohybu).
  - Méně zachycených fotonů, takže musí být dostatek světla.
  - Nepůjde použít velké clonové číslo.

14.20. Jak se nastavuje expozice? Jaký vliv má clona a délka expozice na snímek?

- Viz předchozí body.
- **Větší délka expozice** – zvyšuje počet fotonů, **Větší clona** – snižuje počet fotonů.

14.21. Co je přeexponovaný a co podexponovaný snímek? Co je horší a proč?

- **Přeexponovaný:**
  - Obsahuje místa, kde jas dosáhl maximální možné hodnoty.
  - Nepůjde opravit, protože chybí informace.
- **Podexponovaný:**
  - Nevyužívá celý dynamický rozsah.
  - Pixely v rozsahu 0-130.
  - Lze upravit a dorovnat na hodnoty 0-255.

14.22. CMOS snímač – v čem je výhodnější?

- Integrovaný obvod s tranzistory citlivými na světlo (je to zároveň i mikroprocesor).
- Používanější než CCD.
- **Menší spotřeba a levnější a jednodušší na výrobu.**
- **Umí přenášet data z každého bodu samostatně** (každou jednu buňku zvlášť).

#### 14.23. Základní rozdíly mezi CCD a CMOS snímačem.

- CCD funguje na principu fotoelektrického jevu a následného posunu excitovaných elektronů.
- CMOS je jednoduchý integrovaný obvod s obvody citlivými na světlo.
- CMOS je levnější na výrobu.

#### 14.24. Co je back-illuminated CMOS? Co je to fill factor?

- **Back-illuminated**
  - Elektronika je umístěna za světlocitlivou částí – činná plocha je téměř 100 %.
  - Výroba je dražší (je to vícevrstvý integrovaný obvod), dražší než CCD.
- **Fill Factor**
  - Udává, kolik plochy snímače je skutečně použito k snímání světla.

#### 14.25. Jak vypadají rolling-shutter artefakty a proč vznikají?

- **CMOS je mikroprocesor**, tedy má nějakou taktovací frekvenci.
- **Nedokáže tak zpracovat data ze všech pixelů naráz**.
- Nejprve dojde k **čtení horních řádků**, a **nakonec spodních řad**.
- Předměty jsou tak různě nakloněné, protažené anebo vznikají různé artefakty.



14.26. Jak funguje scanner? Jaké má základní parametry? Proč zde stačí použít lineární snímač? Jaké typy scannerů znáte? Co je to OCR, ICE?

- Snímač scanneru **se posouvá** nad nebo pod nehybným předmětem.
- Lze použít **lineární CCD** – předmět je nehybný, nepoužívá se optika.
- Není použita Bayerova maska, ale **barevné filtry** a různé algoritmy.
- **Základní parametry:**
  - **Rozměr snímané předlohy** – většinou A4.
  - **Rozlišení** – běžně 4800 DPI.
  - **Barevná hloubka** – běžně 16 bitů na kanál.
  - **Denzita** – prakticky dynamický rozsah.
- **Typy:**
  - **Ruční scanner**
    - Musí se plynule ručně posouvat přes předlohu – nekvalitní kvůli drobným posunům.
  - **Stolní scanner**
    - Běžný domácí, umisťuje se pod víko.
  - **Průchodové**
    - Má podavač papíru, takže je velmi rychlý.
  - **Bubnové**
    - Objekt umístěn na buben, obraz sejme jedna fotodioda.
    - Velmi vysoká cena, vysoké rozlišení, ale pomalé.
- **OCR** – optické rozpoznávání znaků – SW technologie.
- **ICE** – použit je ještě kanál citlivý na IR světlo, které dokáže zaznamenat i prach a povrchové vady.

14.27. K čemu slouží čárový kód, jak se jmenuje ten nejrozšířenější. Jaké typy čteček čárového kódu znáte? Co je to QR?

- **Čárový kód:**
  - K umístění na produkty v obchodě.
  - Nejrozšířenější **EAN-13**.
  - Mnoho standardů, ke skenování stačí lineární skener.

- **Typy čteček:**
  - **Laserové snímače** – jeden nebo více paprsků emitovaných laserovými diodami, běžné v supermarketech.
  - **Digitální snímač** – pracuje podobně jako fotoaparát, SW následně zpracuje.
  - **Přikládací CCD snímač** – nutné přímo přiložit, nízká cena.
- **QR kód:**
  - 40 velikostí (17 + 4v).
  - Nejmenší 21 x 21 px, největší 177 x 177 px.
  - Různé kódovací algoritmy, dokonce může být i část kódu poškozena.
  - Neexistuje žádné čtečka QR kódů, vše je řešeno SW.

14.28. Jak se dají skenovat 3D objekty? – uveděte aspoň 3 principy dle vlastního výběru.

- **Kontaktní 3D scanner**
  - Nutné objekt naskenovat dotykovou sondou.
- **Bezkontaktní laserový 3D scanner**
  - Krátké impulsy laseru. Podle času návratu paprsku je určena vzdálenost. Velmi kvalitní, ale spíše pro průmysl.
- **Triangulační laserový scanner**
  - Jezdí na objektu červeným paprskem a podle rychlosti a směru pohybu vypočítá tvar a vzdálenost.
- **Scanner s projekcí vzoru**
  - Na objekt promítnut vzor a podle deformace úhlu lze odvodit tvar.
- **Pasivní stereoskopický scanner**
  - Dva fotoaparáty, které snímají objekt ze dvou různých stran.
- **Fotogrammetrie**
  - Mnoho fotek z více úhlů, SW se snaží odhadnout.
- **Fotometrický**
  - Pomocí stínů ze světel se snaží vypočítat tvar objektu.
- **Siluetový**
  - Objekt je rotován kolem své osy.

## 15. Mobilní hardware

- Historie a vývoj mobilního hardwaru
- Notebooky a netbooky
- Mikroprocesory Atom
- Mikroprocesory ARM
- System on chip, Embedded system
- Smartphone, tablet
- Polohovací zařízení, akcelerometr
- GPS
- Paměťové karty
- NFC

### 15.1. Co považujeme za první mobilní hardware?

- Přenosné počítače, které **mohou být napájeny z baterie**. Lze s nimi pracovat v pohybu.
- Říká se jim přenosný počítač, ne notebook. Okolo roku 1980.

### 15.2. Vysvětlete, co to znamená notebook a laptop a kdy se poprvé tyto pojmy objevily.

- **Notebook = laptop = počítač na klín.**
- Notebook je přenosný osobní počítač, který má integrovaný displej, akumulátor a klávesnice.
- Termín notebook se začíná používat **okolo roku 1990**.

### 15.3. Zhruba popište, jak první notebooky vypadaly (displeje, touchpad, hmotnost, disketové mechaniky....), u prvních notebooků nebyl pevný disk, objevuje se až na přelomu osmdesátých a devadesátých let. Co je to 3-spindle? Srovnejte výkon notebooků a desktopů dříve dnes. Srovnejte cenu notebooků a desktopů dříve a dnes.

- První notebooky – velmi **těžké a drahé**.
- Neměly pevný disk – **používaly disketové mechaniky**.
- Rozšíření a v devadesátých letech – v té době cena klesá a stávají se běžnějšími.
- **Three spindle** – obsahuje tři integrované točivé mechaniky – **pevný disk, CD-ROM a disketovou mechaniku**.

15.4. Vyjmenujte specifické parametry notebooků (tzn. ty které nás zajímají typicky u notebooků a ne u desktopů – např. úhlopříčka, kapacita a výdrž akumulátoru...) a uveděte jejich typické hodnoty.

- **Rozměr** – obvykle dle úhlopříčky displeje – od 9 palců do 17 palců (a dle poměru stran, dnes 16:9 / 16:10, dříve 4:3).
- **Hmotnost** – udává se bez akumulátoru – dnes od 1 do 4 kg.
- **Typ použitého akumulátoru** – kapacita (např.: 42 Wh) a výdrž (např.: 10 hodin).
- **Klávesnice** – záleží na rozložení (typicky jestli má numerické klávesy).
- **Konektivita** – počet USB konektorů, Wi-Fi, Bluetooth, HDMI, VGA.
- **Možnost výměny pevného disku / RAM.**
- **Displej** – IPS / PVA / TN / OLED.

15.5. Jak vypadá napájení notebooku?

- Napájení probíhá z vnitřního akumulátoru (jinak by to nebyl mobilní HW).
- Nabíjení akumulátoru probíhá uvnitř zařízení a samo si ho řídí.
- Ze sítě nelze napájet přímo, notebooky mají dnes obvykle napájecí napětí mezi **12 – 24 V** a potřebují vhodný adaptér.
- **Napájecí zdroj není integrován**, je dodáván s notebookem a připojuje se zvěnčí.

15.6. Touchpad – vysvětlit, co to je, kdy se to poprvé objevilo a jak to zhruba funguje. Vysvětlit pojmem multitouch.

- Plocha citlivá na dotek prstu, nahrazuje myš.
- Pracuje na podobném principu jako **kapacitní displej**.
- Většinou **nízké rozlišení**.
- Moderní touchpad umí **Multi-touch** (dotek na více místech) – vyžaduje **více elektrod** rozložených po celé ploše, které detekují výměnu náboje.

15.7. Popište další používaná polohovací zařízení. Trackball, Trackpoint – kde se typicky objevuje, k čemu se dá použít, používá se ještě?

- **Trackball** – “kuličková myš naruby”. Kulička je umístěna pod klávesnicí a podle pohybu se otáčí ve zvoleném směru. Kdysi používal Apple.

- **Trackpoint** - Miniaturní joystick, nakloněním se bude kurzor pohybovat v příslušném směru. Používalo IBM, stále z nepochopitelného důvodu mají některé notebooky Lenovo.

## 15.8. Akcelerometr – co to je, jak to funguje, v jakých zařízeních ho najdeme a jak se dá využít k ovládání zařízení.

- Slouží k určení měření velikosti a směru zrychlení.
- Označován jako **G-Sensor**.
- Původně na foťácích pro správné natočení fotografie. Poté PDA a pro mobilní telefony.
- **Vhodné** – překlopení obsahu, AR aplikace, VR cardboard a různé jiné aplikace (hvězdná mapa, vypnutí budíku zatřesením, krokoměr).
- **Dva typy:**
  - **Kapacitní** – dvě oddělené pružným dielektrikem těžké elektrody kondenzátoru jsou od sebe gravitací odtahovány. Čím více, tím jsou více kolmo k zemi.
  - **Piezoelektrický** – piezokrystal **deformován ve směru působení gravitace**. Tři piezokrystaly jsou orientovány **kolmo na sebe**, nejvíce protažený bude ten, který je ve **směru rovnoběžném k zemi**.

## 15.9. Jaké uzpůsobené komponenty najdeme v notebooku – pevný disk, paměťové moduly, mikroprocesory, klávesnice... (jak se liší od desktop variant).

- **Mikroprocesor**
  - Úsporné varianty s jednoduchým slotem (v dnešní době většinou připájen).
  - Výkon standardních mikroprocesorů srovnatelný s desktopy, rozdíl v náročných aplikacích zejména kvůli dostatečnému chlazení.
- **Pevný disk**
  - Používají se disky o velikosti 2.5 palce s menšími otáčky, které jsou odolnější vůči otřesům.
  - Časté použití SSD, nižší kapacita než desktop.
- **Paměťové moduly**
  - Menší tvar (**SO-DIMM**), o trochu dražší, ale jsou menší a vyšší.
- **Klávesnice**
  - Neobsahuje například numerické klávesnice, či nějaké funkční klávesy (pravý ALT apod.).

## 15.10. Čtečky otisků prstů – jak fungují (vysvětlit 3 principy).

- **Optická čtečka**

- **Pořídí 2D snímek prstu** a podle algoritmu zjišťuje papilární linie (analyzují se světlá a tmavá místa).
- Snímání je rychlé, levné, musí být zezdola osvíceno, lze ji snadno podvést obrázkem.

- **Kapacitní čtečka**

- Malá destička pokrytá hustou sítí **miniaturních kapacitních senzorů** – při doteku s prstem dochází k výměně náboje.
- Díky výměně náboje zjistí papilární linie – tam dojde ke změně náboje, jinde ne.
- Snímá 3D, tedy nelze oklamat fotkou. Přikládaný materiál by musel měnit velikost náboje v kondenzátorech stejně jako lidský prst.

- **Ultrazvukové čtečky**

- Ultrazvukový signál je vyslán směrem k prstu.
- Přijímač vyhodnotí, za **jak dlouhou dobu se odražený signál vrátí nazpět** - ty, co se vrátí dřív, jsou papilární linie.
- Obvykle pro čtečku zabudovanou v displeji, ovšem poměrně drahé.

## 15.11. Konektivita notebooku (dnes USB, VGA, DVI, HDMI, Bluetooth, Ethernet, WiFi...).

- USB 2.0, USB 3.0.
- USB-C, Thunderbolt 3, Thunderbolt 4.
- VGA (staré), DVI, HDMI.
- Wi-Fi, Bluetooth, RJ45 port (na ústupu).

## 15.12. PCMCIA – jak to vypadalo, jaký typ komunikace se používá, jaké zařízení se nejčastěji připojuje.

- Rozšiřující slot pro periferní zařízení a komponenty (nahrazuje tak tehdejší sériový či paralelní port).
- Později označení PC Card.
- **Například:** ethernetová síťová karta, modem, zvuková karta, disketová mechanika, televizní karta, Wi-Fi, Bluetooth adaptér.
- Rozhraní se později změnilo na **ExpressCard** (přechod na sériovou komunikaci).



15.13. ExpressCard – jak to vypadalo, jaký typ komunikace se používá, jaké zařízení se nejčastěji připojuje.

- Vzniká v roce 2003, pro připojení čteček paměťových karet, TV tunerů, Wi-Fi karet, SSD, grafických adaptérů, šifrovacích karet.
- Sloučení signálů PCI-Express, USB 2.0 a USB 3.0.
- Rychlosť až 2.5 Gb/s, později 5 Gb/s.
- Sériová komunikace.



15.14. Intel Centrino – co to vůbec je. Není potřeba znát přesně jednotlivé generace, ale musíte vědět, jak postupně probíhal vývoj a co se postupně objevuje, integruje a umožňuje (bez přesného určení času a generace).

- Jedná se o čipovou sadu Intelu pro **mobilní zařízení a notebooky**.
- Optimalizované komponenty.
- **Carmel (1st gen, 2003):**
  - Vychází z Pentium M, slabý grafický procesor.
  - Sběrnice PCI, PATA
  - Spotřeba cca 11 W.
- **Sonoma (2nd gen, 2005):**
  - PCI-E, SATA.
  - Spotřeba 15 W:
- **Napa (3rd gen, 2006):**
  - Architektura Intel Core.
  - Nálepka Intel Centrino.
- **Santa Rosa (4th gen, 2007):**
  - TurboBoost, různé výkonnostní stupně (frekvence) celého chipsetu.
- **Centrino 2 (5th gen, 2008):**
  - Gigabitový ethernet, volitelné GPU, HDMI výstup.
- **Calpella (6th gen, 2009):**
  - Grafický procesor s podporou HD a H.264 a MPEG2.
- **Huron (7th gen, 2011):**
  - Rodina SandyBridge.

- **Chief River (8th gen, 2012):**
  - Rodina IvyBridge.
- **Shark Bay (9th gen, 2013):**
  - Rodina Haswell, podpora DDR4.

Následují generace série 100 a 200.

Poté se pojmen Intel Centrino používá pro bezdrátové síťové komponenty Intelu – **Intel Centrino Wireless**.

15.15. Netbook – v čem se liší od notebooku, jaké parametry jsou pro něj typické, typické využití a omezení (např. nemá optickou mechaniku, nemá numerickou klávesnici...), jakou má spotřebu energie.

- **Netbook = miniaturní notebook.** Od roku 2008.
- Minimální hmotnost, maximální mobilita a nízká spotřeba.
- Určen pro jednoduchou kancelářskou práci, připojení k internetu a použití síťových aplikací (**NETbook**).
- Po roce 2010 produkce klesá kvůli rostoucí popularitě tabletů.
- Používá Intel Atom (tedy je kompatibilní s [x86](#)) - nízká spotřeba procesoru – **jednotky W** (celkový **příkon zařízení 5-15 W**).
- Nemá integrovanou CD/DVD mechaniku.
- Klávesnice neobsahuje numerickou část a tlačítka mají menší rozlohu.
- Hmotnost – 800 až 1500 gramů.

15.16. Úsporné mikroprocesory Atom – stručný přehled generací (není nutné mít encyklopedické znalosti, ale musíte vědět, od kdy se přibližně vyráběly, jaké mají rozměry, příkon, možnosti...).

- **Intel Atom – procesor.**
- **Intel Centrino – chipset** (to znamená, že tam je severní a jižní můstek, Wi-Fi rozhraní a GPU čip).
- **Silverthorne (1st gen, 2008):**
  - Jednojádrové, většina tranzistorů cache.

- **Diamondvill (2nd gen, 2008):**
  - Jednojádrové – spotřeba 4W, cena 25 \$.
  - Dvoujádrové – spotřeba 8W, cena 40 \$.
  - Taktovací frekvence 1600 MHz.
- **Pineview (3rd gen, 2010):**
  - Jednojádrové – spotřeba 5.5W.
  - Dvoujádrové – spotřeba 13W.
  - Taktovací frekvence 1.67 / 1.83 GHz.
  - Cena 60 – 80 \$ (dle konfigurace).
- **Cadarview (4rd gen, 2011):**
  - Dvoujádrové, spotřeba 3.5 – 6.5 W.
  - Frekvence 1.6 – 1.9 GHz.
- **Penwell (5th gen, 2012):**
  - Motorola telefony, SoC.
  - Rozměry 12 x 12 x 1 mm.
  - Frekvence 1.3 GHz – 2 GHz.
- **Cloverview (6th gen, 2013):**
  - Dvě jádra, 1.2 – 2 GHz.
  - Velká L2 cache.
- **Bay-Trail-T (7th gen, 2013):**
  - GPU čip jako v IvyBridge, frekvence 1.33 – 2.39 GHz.
  - Rozměr 17 x 17 x 1 mm.

## 15.17. Co je to MID procesor?

- **Extra úsporné procesory pro tablety a smartphony.**
- **MID processor – Mobile Internet Device Processor.**
- Nejmenší spotřeba 0.5 W A (Atom Z500).
- Lze na nich spustit x86 aplikace.
  
- **Lincroft (2nd gen, 2010):**
  - 900-1600 MHz.
  - Spotřeba 1.3 W – 2.2 W.
  - Rozměr 13.8 x 13.8 x 1 mm.

15.18. Mikroprocesory ARM – co znamená zkratka ARM, v čem jsou odlišné oproti mikroprocesorům v PC (RISC, malý počet tranzistorů, díky tomu nízká spotřeba, velký počet registrů, nekompatibilita s PC....) – toto je vše velmi důležitá část této maturitní otázky. Dopadne to hodně špatně, pokud nebudeš vědět, že ARM jsou v současnosti nejpoužívanější procesorová jádra na světě, že jsou RISC, nekompatibilní s procesory v PC, mají malou spotřebu, nízkou cenu a firma ARM je nevyrábí.

- ARM jsou **RISC** vyvinuté firmou **Acorn** již v 80. letech.
- **ARM** = Acorn RISC Machine, dnes **Advanced RISC Machine**.
- **Firma ARM není sama výrobcem**, ale jen je navrhuje a prodává licenci dalším firmám – Apple, Nvidia, Qualcomm, Samsung, Texas Instruments. Tyto výrobci je integrují do svých SoC čipů.
- **Nejsou kompatibilní s procesory řady x86**.
- V současnosti **nejpoužívanější** (až 95 %) **na světě** – vyrobeno přes 6 miliard každý rok.
- **RISC architektura** – malý počet tranzistorů, nízká spotřeba, univerzální velký počet registrů, jednodušší na návrh.
- Vhodné do embedded systémů, mají malý příkon, nízkou cenu.

15.19. Kdo vyrábí procesory ARM? Co vlastně dělá firma ARM?

- **ARM poskytuje licenci k výrobě, kompilátory a softwarová vývojová prostředí**.
- Lze koupit například licence pro celé **detailní schéma** s jednotlivými tranzistory nebo naopak pouze **popis jádra v HDL** (Hardware Definition Language) jazyce – výrobce si sám návrh zkompiluje a podle toho se čip vyrobí.

15.20. Typické vlastnosti posledních řad procesorů ARM (počet registrů, počet bitů, použití...).

- **32 bitové procesory (Cortex** - ARMv8-R, ARMv7-A, ARMv7-R, ARMv7E-M, ARMv7-M, ARMv6-M):
  - **16 x 32 bitový** registr pro uložení celých čísel.
  - **32 x 64 bitový** registr pro **FP** (s desetinnou čárkou) či **SIMD** výpočty.
- **64 bitové procesory (Cortex A-5X** – ARMv8-A):
  - Od roku 2011.
  - **31 x 64 bitový** registr pro uložení celých čísel.
  - **32 x 128 bitový** registr pro **FP** (s desetinnou čárkou) či **SIMD** výpočty.

15.21. Co je to SoC? Co vše lze integrovat (je toho hodně a budu to chtít vyjmenovat).

- **SoC = System on Chip.**
- Slučuje všechny elektronické obvody a komponenty do jediného integrovaného obvodu.
- Levnější, rychlejší, spolehlivější, ale nemožnost rozšíření.
- **Lze integrovat:**
  - CPU, GPU.
  - Flash paměť, DRAM.
  - HDMI, zvukový modul.
  - Wi-Fi, Bluetooth, GPS, ethernet rozhraní.
  - Řadič LCD displeje, řadič obvodu dotykové obrazovky.
  - SATA, USB controller, SD card controller.
  - GSM, LTE, 5G – komunikační rozhraní.
- **Nelze integrovat:**
  - Akcelerometr

15.22. Co je to embedded-systém?

- Počítačový systém s **jednoúčelovou funkcí**, který je součástí většího elektrického nebo mechanického zařízení.
- Například laserová tiskárna, automatická pračka, počítačová myš, digitální fotoaparát.
- Optimalizací embedded systému pro konkrétní aplikaci lze dosáhnout **vysokého výkonu, spolehlivosti a nízké ceny** v porovnání s použitím univerzálního počítače.
- Dnes většina založena na **ARM** – tedy jedná se o **SoC**.

15.23. Co je to real-time systém?

- Nepřetržitě **interaguje se svým okolím přesně definovaným způsobem**.
- Nezávisí jenom na **logicky správném výsledku**, ale i **na čase**, do kterého musí být operace provedena.
- U každé operace lze stanovit **deadline**.
- **Použití** – řídící jednotka motoru (důležité řídit včasné vstřikování paliva), autopilot, ETCS zabezpečovací železniční systém.

- Používá se **SRP** (Synchronous Reactive Programming) – programovací jazyk s možností přesně definovat časovou soslednost (přesně definovat, co se má kdy provést).

15.24. Co je UMPC a PDA? K čemu se používaly? Jak se liší PDA od tabletu? Jak liší UMPC od notebooku a jak od PDA? Jaké OS se používaly na PDA? K čemu se typicky používalo PDA?

- **UMPC – Ultra Mobile Personal Computer**
  - Předchůdce dnešních tabletů a smartphonů.
  - Miniaturizovaný notebook s mikroprocesor [x86](#).
  - Rozvoj 2004 – 2007.
- **PDA – Personal Digital Assistant**
  - Vlastně předchůdce smartphonu, ale nemá funkce volání.
  - Funkce kalendáře, navigace, poslechu hudby, poznámky, čtení knih apod.
  - Menší než tablet (podobné jako mobil, ale spíše širší a nižší).
  - Například PDA Newton od Apple.
  - Palm OS, Windows Mobile, iOS, Android, Symbian, Linux.

15.25. Jak funguje GPS? Co vysílá GPS družice na zem? Vysílá něco navigace (hlavně neříkejte, že ano – GPS je PASIVNÍ a pouze přijímá časové údaje z družic). Jak daleko jsou družice? Jak přesně měří čas družice? Jaké fyzikální je nutné brát v úvahu při přesném měření času? Jak přesně měří zpoždění signálu GPS přijímač? Kdy je zpoždění nejvyšší a nejnižší? Kolik GPS satelitů je potřeba přijímat k přesnému určení polohy? Kdo je provozovatelem GPS? Kdo je dominantním poskytovatelem čipů pro dekódování signálu? Projde signál zdí? Lze signál rušit? Jaký důsledek mají odrazy signálu od budov?

- **GPS = Global Positioning System.**
- Provozuje **ministerstvo obrany USA**.
- Firma **SiRF** vyrábí řadu čipů pro příjem GPS signálu a GPS modulů, má patentovanou řadu chipsetů a zařízení.
- **Princip:**
  - Okolo Země **obíhá 24 družic** (ve výšce **20350 km**) po 6 různých dráhách.
  - Družice má v sobě **atomové hodiny** (velmi přesné, přesnost 0.1 pikosekundy) a všechny družice mají **přesně synchronizovaný čas**.

- Družice tento přesný čas vysílají na Zem. Signál k nám ale kvůli vzdálenosti družic **nedorazí ve stejnou chvíli** (každá družice je jinak daleko), ale se **zpožděním**.
  - Podle zpoždění signálu tak dokážeme **přesně vypočítat**, kde se **nacházíme** – podle tzv. **trilaterace**.
  - Pro správné určení polohy je nutné dostat signál **alespoň od 4 družic**, při 3 bychom neměli nadmořskou výšku (viditelných je vždycky 5-10).
  - Zpoždění signálu se pohybuje v rozsahu **67** až **86 ms**.
- **Samotný přijímač nic nevysílá, je to pasivní prvek a pouze obdrží signál. GPS družice nemá ponětí, kolik GPS přijímačů odebírá od ní signál.**
  - Při výpočtu je nutné vzít v potaz i **teorii relativity** (pohybující se hodiny běží pomaleji než hodiny v klidu, ale ve slabším gravitačním poli běží rychleji – je to cca **450 pikosekund za sekundu rozdíl**).
  - Vysokofrekvenční signál (1575.42 MHz) lze využít pouze na otevřeném prostranství, většinou neprojde zdí či jen velmi omezeně – má tendenci se odrážet od **železobetonových konstrukcí**.
  - Signál lze **rušit i lokálně** na Zemi – většinou u vojenských objektů apod.
  - GPS má **menší zpoždění přímo nade mnou**, nad **horizontem je zpoždění vyšší** (letadlo co letí nade mnou je taky blíž než co vidím někde daleko nad horizontem – pokud letí ve stejné výšce).

15.26. Smartphone – používané OS, výhody a nevýhody oproti klasickému telefonu, paměťové prostory (budu trvat na vyjmenování všech prostorů, typu paměti, kterou jsou realizované a jejich významu).

- **Výhody** – univerzální, alternativy k hlasovým hovorům (kolik korun měsíčně člověk provolá dnes a kolik před 10 lety?).
- **Nevýhody** – menší výdrž baterie, dražší, více náchylné na poškození, větší hmotnost, lze stáhnout malware.
- **OS:**
  - **iOs – Apple** – dražší, více spolehlivé, větší požadavky na zabezpečení.
  - **Android – Google** – levnější, méně spolehlivé, menší požadavky na zabezpečení.
  - **Windows Phone** – nevyrábí se, možná ještě někdo používá.
  - **Symbian** – nepoužívá se.
  - **Blackberry** – nepoužívá se.

- **Paměti:**

- **Hlavní operační paměť** – DDR SDRAM.
- **Interní úložiště** – FLASH NAND MLC.

15.27. Tablety – uvést základní parametry, používané OS.

- **Velikost úhlopříčky** – 7 až 13 palců.
- **Konstrukce** – plastová, kovová, voděodolné.
- **Další parametry** - kapacita vnitřního úložiště, porty, velikost RAM.
- **OS** – stejné jak u mobilů, navíc ještě Windows, pokud je **x86**.

15.28. NFC – vysvětlit co to je a k čemu se to dá použít.

- Bezdrátová technologie umožňující rychlou a bezpečnou výměnu dat na krátkou vzdálenost (cca do 5 cm).
- **Využití** – bezkontaktní platby, odemykání dveří, automatizační procesy (v mobilu je aplikace, která například nastaví budík při přiblížení ke štítku).

15.29. Paměťové karty – jaké typy dnes existují, popsat rozdíl oproti SSD. Jaká označení na nich najdeme? Podle čeho posuzujeme jejich kvalitu? Jaké jsou běžné rychlosti čtení a zápisu?

- **SD = Secure Digital.**
- Oproti SSD nemá **TRIM**, zápis bývá pomalejší, nižší **IOPS** (neumí pracovat s velkým množstvím malých souborů), chybí overprovisioning, nestěhuje dlouho ležící soubory.
- **Velikosti** – SD, micro SD.
- **Typy** – SDHC (4 – 32 GB), SDXC (64 GB až 2 TB).
- **Rychlostní třídy** (běžně se používá **Class 10** – zápis kolem **20 – 100 MB/s**, čtení kolem **100 – 200 MB/s**):

Minimální rychlosť zápisu	Třída rychlosťi	Třída UHS	Video třída	Vhodné pro
6 MB/s	 Class 6 (C6)	–	 Class 6 (V6)	HD videa
10 MB/s	 Class 10 (C10)	 UHS Class 1 (U1)	 Class 10 (V10)	Full HD videa
30 MB/s	–	 UHS Class 3 (U3)	 Class 30 (V30)	4K videa při 60/120 fps

Třída	Logo	Min. rychlosť zápisu
Třída 2		2 MB/s
Třída 4		4 MB/s
Třída 6		6 MB/s
Třída 10		10 MB/s
UHS-1		10 MB/s
V10 (video třída)		10 MB/s
UHS-3		30 MB/s
V30 (video třída)		30 MB/s

## 16. Grafické adaptéry

- Vývoj grafických adaptérů
- Uložení dat ve videopaměti
- 2D akcelerace, 3D akcelerace, základní pojmy

### 16.1. Jak vypadaly první grafické adaptery? Co je to textový režim? Jaký měly výstup?

- **První adaptéry** – pouze textový režim, monochromatický (zelený text na černém pozadí), GPU i monitor měl speciální konektor.
- **Textový režim:**
  - Pouze text.
  - Obrazovka rozdělena na rastr (mřížku) pro znaky textu. Určitý počet řádků a sloupců.
- **Všechny tyto karty vyšly okolo roku 1981-1984, analogové výstupy.**
- **MDA:**
  - Monochrome display adapter.
  - Textový režim – 25 řádků, 80 znaků na řádek, bez diakritiky.
  - 720 x 350 px, 50 Hz.
  - Možný i grafický režim 320 x 200 px.
- **CGA:**
  - Color graphics adapter.
  - 4 barvy při rozlišení 640 x 200 px.
  - Dalo se přeprogramovat a zobrazit i diakritika.
  - Možný i textový režim 320 x 200 px.
  - Umí pomocí bajtu určit barvu textu (4 b), pozadí (3 b) a blikání (1 b).
- **HGC:**
  - Hercules graphics card.
  - Textový režim 720 x 350 px.
  - Grafické dvobarevné zobrazení 720 x 348 px.
- **EGA:**
  - Enhanced graphic adapter.
  - 16 barev 340 x 350 px.
  - ISA / EISA.

## 16.2. Jak se postupně zlepšovaly možnosti grafických karet?

- **PGC**
  - **První grafický akcelerátor.**
  - Zabírá 3 (!!) ISA sloty.
  - **Vykonavatel grafických příkazů (cca 80):**
    - Vykreslení kružnice, vyplnění plochy barvou, vykreslení úsečky, rotace objektu, smazání celé obrazovky (jinak by bylo nutné zapsat tisíce nulových bajtů do obrazové paměti).
  - Ušetří tedy práci mikroprocesoru, který nemusí počítat souřadnice všech bodů - například při kreslení kružnice  $x, y$ .
- Nejstarší karty fungují jako **ROP** (Render Output Unit). Jednotka je **odpovědná za generování signálu pro monitor**. Neprobíhají zde žádné výpočty, pouze se **data převádí na výstupní signál**.
- Moderní grafický adaptér je vlastně samostatnou výpočetní jednotkou, která zpracovává obrazová data k tomuto účelu má speciální architekturu.

## 16.3. Jaká byla rozlišení používaná dříve, jaká dnes?

- **MDA** - 720 x 350 px, monochromatický.
- **CGA** - 640 x 200 px, 4 barvy.
- **HGC** - 720 x 348 px, 2 bary.
- Dnes nejpoužívanější rozlišení **1920x1080** a postupný přechod k **3840x2160, 24 bitová barevná hloubka** (běžně od 90. let).

## 16.4. Jak rostl počet barev?

- Viz předchozí bod.

## 16.5. Popište způsob uložení grafických dat v paměti (text, barvy, pixely...).

- **Textový režim:**
  - **ASCII** kódy znaků na jednotlivých řádcích.
  - **Tak jak jdou za sebou na displeji, tak jsou uloženy v paměti.**

- Úkolem grafické karty bylo pouze tato **data několikrát za sekundu přečíst** (například 50x za sekundu) a **převést data na analogový signál pro monitor.**
- **Barvy:**
  - Ke každému znaku je potřeba přidat také informaci o jeho barvě.
  - Většinou 2 B (1 B znak a informace o barvě).
    - 256 barev textu.
    - 16 barev textu a 16 barev pozadí.
    - 16 barev textu, 8 barev pozadí a bit označující aktivní blikání.
- **Grafický režim:**
  - Základní obrazový bod – **pixel**.
  - Každý obrazový bod má určitou barvu.
  - V monochromatickém režimu je uložena informace **jedním bitem – svítí / nesvítí**.
  - Podle počtu bitů barevná hloubka – True Color – 24 bitů.

16.6. Kolik paměti je třeba k uložení snímku s určitým rozlišením a barevnou hloubkou?

- **True Color pixel 1920x1080:**
- $1920 * 1080 * 24 \text{ bitů} = 49\ 766\ 400 \text{ b} = \mathbf{6\ 220\ 800\ B = cca\ 6\ MB}$ .

16.7. Co je to VESA? Co vše je výsledkem práce VESA?

- **Video Electronics Standard Association.**
- **Asociace výrobců grafických karet, monitorů a dalších komponent.**
- Snaží se sjednocovat grafické formáty, výstupy grafických karet a textové a grafické režimy.
- Není potřeba pro každou grafickou kartu programovat aplikaci jinak nebo kupovat jiný monitor.
- **Výsledky: VL-BUS, SVGA (TrueColor), Display Port, VESA BIOS** (jednotné nastavování, zjišťování parametrů grafických karet, struktura dat v obrazové paměti a komunikace grafického adaptéru přes sběrnice).
- **Členové:** NVIDIA, AMD, Samsung, Apple, Microsoft, ASUS, Qualcomm, LG...

16.8. Která karta jako první umí téměř megapixelové rozlišení a barevnou hloubku true-color?

- **SVGA** – rozlišení až 1024 x 768 px.
- **True Color** – 24 bitů (16.7 milionu barev).
- Pro uložení v maximálním rozlišení v True Color **2.25 MB**.

16.9. V čem spočívala funkce grafických karet bez akcelerace?

- Nazývají se jako **ROP** (Render Output Unit). Jednotka je odpovědná za **generování signálu pro monitor**. **Neprobíhají zde žádné výpočty**, pouze se grafická data převádí na výstupní signál pro monitor.

16.10. Co je to grafický akcelerátor? Jak vypadal první?

- Umí takéž **provádět i výpočty** (například polohy zadaných těles).
- Obsahuje **samostatný mikroprocesor**, který dokáže provádět různé specializované výpočty.
- **PGC**
  - **První grafický akcelerátor**.
  - Zabírá 3 (!! ) ISA sloty.
  - **Vykonavatel grafických příkazů (cca 80):**
    - Vykreslení kružnice, vyplnění plochy barvou, vykreslení úsečky, rotace objektu, smazání celé obrazovky (jinak by bylo nutné zapsat tisíce nulových bajtů do obrazové paměti).
    - Ušetří tedy práci mikroprocesoru, který nemusí počítat souřadnice všech bodů - například při kreslení kružnice **x, y**.

16.11. Co je interaktivní grafika? Co je to FPS?

- **Interaktivní grafika** – hra, animace, video nebo program musí okamžitě reagovat na ovládání uživatelem a působit plynulým dojmem.
- **FPS** – Frames Per Second – počet snímků za sekundu (15 FPS minimálně pro to, aby to bylo přijatelné, spíše ale 25 FPS).

16.12. 2D akcelerace – kdy přibližně se začíná používat? Jaké metody se používají – color blending, double-buffering, blitting, spriting, scrolling, mirroring, scaling, průhlednost, komprese a dekomprese, transformace mezi barevnými systémy – vysvětlit, jak to vše funguje a jak konkrétně to ulehčí zatížení mikroprocesoru a sníží tok dat na sběrnici.

- V první polovině 90. let.
  - **2D akcelerace** – výpočty s úsečkami, křivkami, textem, obdélníky, kružnice apod.
  - Využívá hojně Windows (či jiný OS) – překreslování oken, scrollování apod., jinak by bylo velmi pomalé.
- 
- **Metody:**
    - **Color blending** – plynulý barevný přechod od jedné barvě k druhé.
    - **Double buffering** – dva snímková buffery, jeden je viditelný, můžeme vykreslovat do toho skrytého. Až bude vykresleno, jednoduše prohodíme, takže uživatel nevidí žádné vykreslování.
    - **Blitting** – v paměti nejčastěji používané rastrové obrazce (ikony, tlačítka apod.). Pak lze snadno vykreslit.
    - **Spriting** – stejně jako blitting, ale jedná se o objekty s nepravidelným okrajem. Používá masku (funguje jako šablona) a pomocí ní se určuje, jaké pixely se posunou.
    - **Scrolling** – vertikální, horizontální či všesměrový posun celého obrazu.
    - **Mirroring** – zrcadlení.
    - **Scaling** – změna měřítka obrazu ([4:3 -> 16:9](#)).
    - **Průhlednost** – informace o tom, zda se má pixel vykreslit a nebo nechá pod sebou stávající pozadí.
    - **Komprese / Dekomprese** – Kosinova Transformace ([DCT](#)) – snaží se popsat obraz matematickou funkcí.
    - **Transformace mezi barevnými prostory** – barva pixelu může být reprezentována jako CMYK nebo YUV (fotografie).
    - **Cutting** – ořezávání obrazu.

16.13. 3D akcelerace – co vlastně umožňuje?

- Umožňuje **rychlé vykreslování prostorových těles**.
- Nekreslí to, co je vevnitř, ale pouze to, co je viditelné – **plochy a okraje (hraniční reprezentace)**.

- Základní operací, kterou musí tyto grafické akcelerátory umět, je **zobrazení trojúhelníku vyplněným zadaným povrchem**.
- Každý bod v prostoru – **vertex**.
- **Polygon** – nejjednodušší prostorové těleso (3 vertexy a 3 úsečky).
- **Mesh** – síť 3D povrchu – tvořena vertexy a plochami polygonů.
  
- **Bump mapping** – iluze nerovnosti povrchu.
- **Alpha mapping** – K RGB přidána ještě Alfa kanál (A), který určuje průhlednost.
- **Fog** – skrývání objektů v mlze.
- **AntiAliasing** – odstranění artefaktů textury při zmenšování a naklánění.

#### 16.14. Co je DirectX, OpenGL, Vulkan?

- Jedná se o **funkce**, které jsou implementovány **v grafických akcelerátorech** – není teda nutné pro každou kartu nutné popisovat všechny funkce, ale použije „univerzální“ pro všechny karty.
- **OpenGL** – Spravuje ARB – NVIDIA, SGI, Microsoft, AMD.
- **DirectX** – součást knihoven sloužící k ovládání zvuku, Microsoft
- **Vulkan** – nástupce OpenGL, vyvíjeno Khronos Group, není zpětně kompatibilní, lépe využívá vícejádrové grafické čipy.

#### 16.15. Co je to GPU?

- **Graphics processing unit - grafický procesor**.
- Samostatná výpočetní jednotka, která zpracovává obrazová data a má přizpůsobenou architekturu.

#### 16.16. Co je to Vertex, pixel, voxel, polygon, mesh, textura?

- **Vertex** – bod v prostoru – souřadnice X, Y, Z.
- **Pixel** – bod na displeji – souřadnice X, Y.
- **Voxel** – objemový element, využíváno ve zdravotnictví a v průmyslu.
- **Polygon** – nejjednodušší prostorové těleso – 3 vrcholy a 3 úsečky.
- **Mesh** – síť 3D modelu, tvořena vertexy a plochami polygonů.
- **Textura** – povrch 3D modelu, rastrový 2D obrázek.

## 16.17. Jak fungují bloky transform a lighting?

- **Transform:**
  - Slouží k transformací vertexů a hran a odstranění neviditelných nebo odvrácených polygonů před vykreslením.
  - Poměrně jednoduché, ale je nutné zpracovat mnoho dat.
  - Neobsahuje žádné větvení a složité rozhodování, proto se realizuje přímo na GPU.
- **Lighting:**
  - Vypočítává osvětlení jednotlivých polygonů.
  - **Vlastnosti světelných zdrojů:** pozice v prostoru, orientace světleného kuželu, koeficienty ambientní, difúzní a odrazové složky.

## 16.18. Co umožňují programovatelné akcelerátory? Co je to CUDA, GPGPU, OpenCL....

- Dříve sloužila ke komunikaci s grafickým akcelerátorem pouze API.
- Nebylo však možné funkce GPU nijak přeprogramovat.
- Moderní grafické karty již obsahují **programovatelné jednotky**, do nichž lze zadat **vlastní program**.
- **Podle typu dat je možné rozlišit na dvě jednotky:**
  - **Vertex shader** – transformace vektorových souřadnic.
  - **Pixel shader** – resterizační prostor, provádí vlastní vykreslování fragmentů – nanášení textur.
- **Jednotka může pracovat ve dvou režimech:**
  - **Programovatelný režim** – ovládání pomocí externě nahraného programu.
  - **Pevný režim** – předem nastavený výrobcem grafického adaptéru.
- **GPGPU**
  - General-purpose computing on graphics processing units.
  - Způsob, jak využít vysoké možnosti paralelizace na grafické kartě k výpočtu libovolných algoritmů.
  - Data neobsahují mezi sebou vazby.
  - **Vhodné pro algoritmy** – počítačového vidění, neuronových sítí, fyzikální simulace, kryptografické výpočty.

- **OpenCL**
  - Open Computing Language.
  - **Standard pro paralelní programování heterogenních počítačových systémů** (systému s různými paralelně pracujícími výpočetními jednotkami – např CPU + GPU).
- **CUDA**
  - Compute Unified Device Architecture.
  - Umožňuje na GPU od firmy NVIDIA spouštět programy napsané v jazycích C/C++, Fortran nebo programy postavené na technologiích OpenCL.
  - **SIMT – Single Instruction Multiple Threads.**
  - GPU obsahuje tisíce bloků – jednoduchých skalárních procesorů.

16.19. Oblasti paměti moderního grafického akcelerátoru – bezpodmínečně nutně musíte znát, jak byla dříve uložena stránka/snímek v paměti a co vše je dnes uloženo v několika různých oblastech moderní graf. karty!

- **System memory** – systémová paměť počítače PC, přístup přes DMA.
- **Video memory** – paměť přímo na grafickém akcelerátoru.
  - **Geometry** – geometrická data těles.
  - **Textures** – rastrové textury použité ve scéně.
  - **Framebuffer** – jednotlivé vykreslované fragmenty.
    - **Color Buffer** – část paměti, která může být vykreslena na obrazovku.
    - **Depth Buffer, Stencil Buffer** – nejsou pro uživatele viditelné, ale slouží k realizaci různých grafických algoritmů.
  - **Commands** – zde jsou uložené programy pro **vertex shader** a **pixel shader**.

16.20. Jakým způsobem je grafická karta připojena v počítači, jak je napájena.

- **PCI-E x16**, většina dnes 3. generace ([PCI-E 3.0](#)).
- Méně výkonné karty se napájí **přímo přes sběrnici**, výkonnější vyžadují **přídavné napájení (12 V)** přímo ze zdroje.

### 16.21. Jak měříme výpočetní výkon grafiky?

- **Použití Benchmarku** – speciální zátěžový test, který změří **výkon karty**.
- **Není možné porovnávat** karty **dle parametrů** – výkon se odvíjí podle mnoha parametrů.
- Populární 3DMark, PCmark, Cinebench.

### 16.22. Jaké paměti se používají na grafických kartách?

- Používá se **GDDR = Graphics DDR**.
- V řádu jednotek (4 - 24 GB).
- Čím více paměti, tím komplexnější grafická data lze uložit (více textur a snímků).

### 16.23. Integrace GPU nejprve do severního můstku a potom do procesorů od řady SandyBridge – výhody nevýhody tohoto trendu.

- Smysl má tam, kde **nejde o grafický výkon**, ale o **efektivitu a nízkou spotřebu** a cenu.
- Vhodné pro **kancelářské počítače**, které by jinak neměly žádnou grafickou kartu.
- Nevýhody může být to, že kvůli integrované GPU (kterou někdo nepotřebuje), je zbytečně **procesor o něco dražší**.
- Pokud je detekována externí karta, je integrovaná GPU většinou vypnuta a nečinná.

## 17. Optický záznam dat

- *Princip optického čtení dat, kódování*
- *CD, CD-R, CD-RW*
- *Otáčky a rychlosť čtení (CLV, CAV, P-CAV, základní rychlosti a násobky)*
- *DVD, DVD+R, DVD-R, DVD+RW*
- *BluRay*

### 17.1. Princip optického čtení dat – laser, pity, odraz, sledování stopy...

- Na CD jsou vytvořeny **miniaturní prohlubně** konstantní šířky - **pity**.
- Hloubka pitů cca 100 nm. Délka pitu je 1 – 3 mikrometry.
- Stopu sleduje **laserový paprsek**, který je zaostřen na místo dopadu (**odrazu**) na průměr 780 nm (není vidět), **výkon 0.5 mW** (0.0005 W).
- V místě **pitu** se paprsek odrazí velmi slabě a naopak tam, kde **pit není (land)** téměř dokonale.
- **Jednička** – změna (začátek nebo konec pitu).
- **Nula** – žádná změna (pokračování pitu nebo landu).
- Maximálně 1 až 3 kroky mezi dvěma změnami (1).
- **Tři lasery** – 2 krajní, které svítí mimo stopu a třetí přenášející datovou informaci.

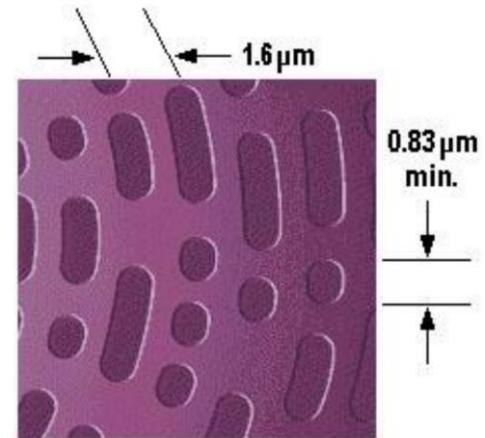
### 17.2. Zabezpečení a kódy – EFM, CIRC, zabezpečení datových sektorů další redundancí na datovém CD.

- **CIRC:**
  - Cross Interleaved Reed Solomon Code.
  - **Přidání řady kontrolních součtů.**
  - Za sebou ležící bajty jsou dále přeházeny – **odolnost vůči shluku chyb**.
  - U Audio CD při větší chybě dokáže chybějící signál odvodit matematickým odhadem.
  - Velikost bloku se několikrát zvětší.
- **EFM:**
  - Eight to fourteen modulation (**8b14b**).
  - **8 bitů je pak překódováno na 14 bitů.**
  - Mezi dvěma jedničkami jsou vždy 1 až 3 nuly.

- **Zabezpečení datového CD:**
  - V každém sektoru je 2352 B, z čehož 304 B je zabezpečovacích informací.
  - Zabezpečeno více než Audio CD – **chyba v jediném bitu učiní bezcenný celý soubor.**

17.3. Audio CD – průměr, tloušťka, počet stop (jedna jediná spirála od středu...), množství dat v jednom sektoru, kódování zvuku, kde je umístěna vrstva s pity, jakou barvu má laser, jak velké jsou pity, jakou šířku má stopa?

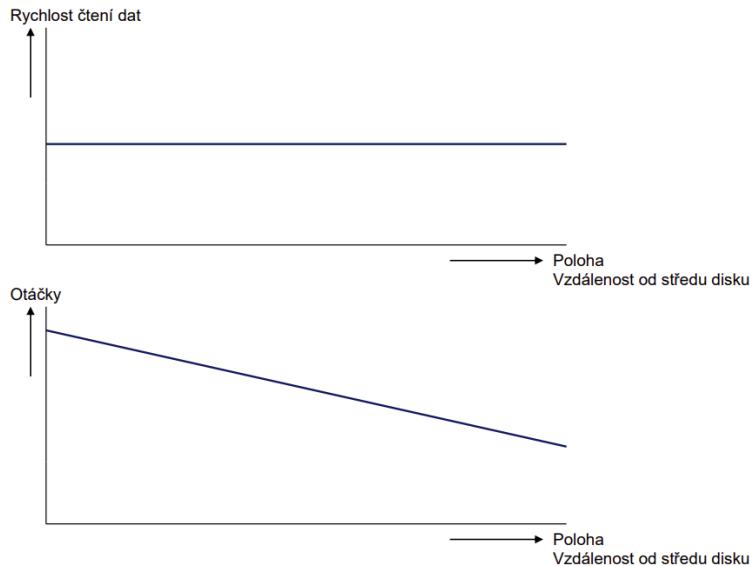
- **Průměr:** 12 cm.
- **Tloušťka:** 1.2 mm.
- **Počet sektorů:**
  - Standardní **333000 sektorů** = 74 minut hudby / 650 MB dat (mocnina dvojky u CD!).
  - V každém sektoru **2048 B informací** (2352 B celkem dat).
  - 1 sekunda na Audio CD zabírá **75 sektorů** ( $75 * 2352 = 176\,400$  B).
- **Počet stop:** 1 spirála od středu ke kraji.
- **Kódování zvuku:** CIRC a EFM.
- **Kde je umístěna vrstva s pity:** pod etiketou (tedy 1.2 mm hluboko).
- **Barva laseru:** infračervený (780 nm).
- **Délka pitu:** 1 – 3 mikrometry.
- **Šířka pitu:** 1.6 mikrometru.



17.4. Rychlosť otáčení pri čtení audio CD – vysvetliť proč jsou proměnné otáčky, vyšší při čtení u středu. Co je to CLV?

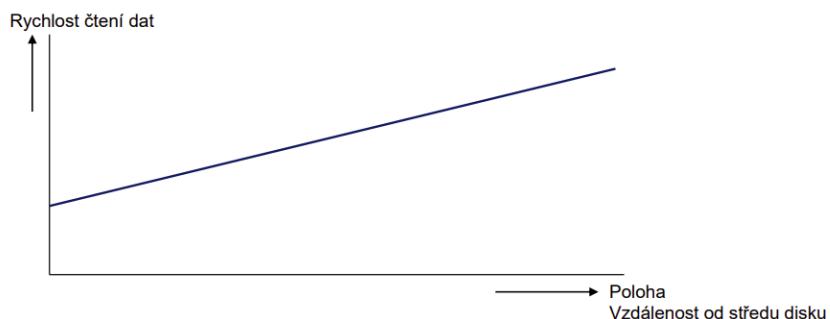
- **CLV = Constant Length Velocity.**
- **Rychlosť čtení dat je konstantná, dĺžka prečtené stopy za 1 sekundu je stejná** (každou sekundu je nutné prečítať 75 sektorov pro Audio CD).
- **Otáčky se mohou postupně snižovať** (od 550 do 200 RPM), protože u středu je méně dat (stejné jako ZBR) a proto u okraje nemusí být tak rychlá.

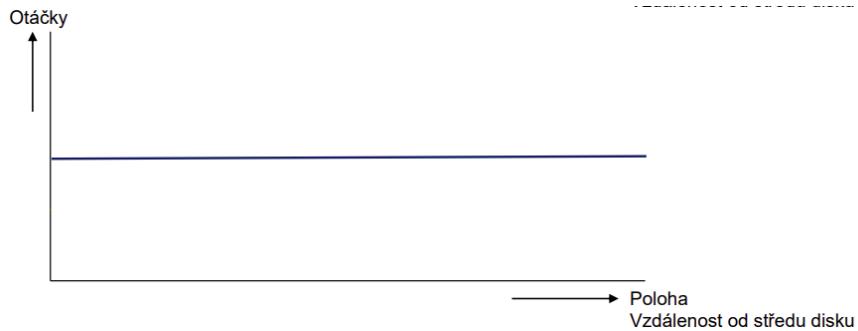
- **Problém s náhodným přístupem** – trvá velmi dlouho, nutné přesně zaměřit.



17.5. Jak je to otáčkami při čtení datového CD? Kde je rychlosť čtení nejvyšší a kde nejnižší při konstantních otáčkách?

- Datové CD může číst rychleji než Audio CD – nebude nám vadit, když každou sekundu přečteme jiné množství dat.
- **CLV** je ovšem nadměrně opotřebovávána střídáním rychlostí otáček, proto není výhodné je používat pro datové CD.
- Proto bylo vyvinuto **CAV = Constant Angle Velocity**, které má **konstantní rychlosť otáček**.
- Tím pádem se ale rychlosť dat odvíjí **dle vzdálenosti od středu (u okraje rychlejší)**.
- Jednorychlostní mechanika = **stejná rychlosť čtení, která se používá pro klasické Audio CD**.
- CD-ROM 48x tedy umí například přečíst  $48 * 2048 * 75 = 7200$  kB/s – této rychlosť ale dosahuje jen **na krajích disku** (tam **je tedy nejrychlejší**).
- To, že se ale čte různou rychlosťí nám nevadí – důležité je, aby u středu **dokázala stíhat přehrát Audio CD**.



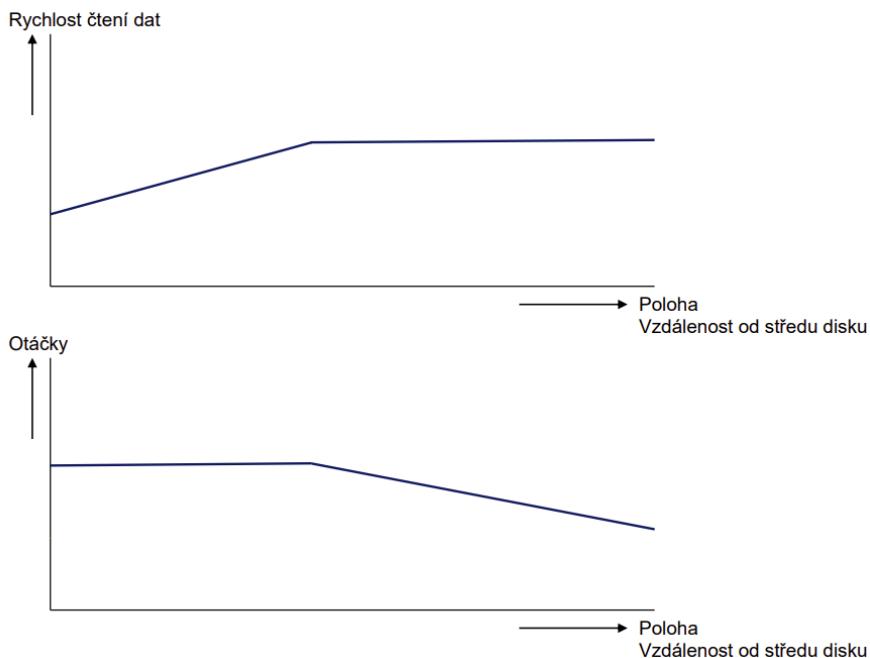


17.6. Co je to CAV? Jaká rychlosť je považovaná za základní? Jakou rychlosťí čte data 40-rychlostní mechanika? Lze data touto rychlosťí čítať z libovolného místa disku? – tyto otázky o CLV a CAV jsou velmi důležité a musíte to umět vysvětlit.

- Viz předchozí bod.

17.7. Jak vypadá graf závislosti rychlosťi čtenia na poloze a závislosti otáček na poloze pro CLV, CAV, P-CAV – Nakreslit aspoň tři nějaké grafy během přípravy na tabuli!

- **P-CAV** (Partial Constant Angular Velocity) - blíže ke středu se používá **CAV**, následně **CLV**. Používá se kvůli tomu, že u středu byly otáčky příliš vysoké a nebylo možné zabezpečit dostatečnou kvalitu nahrávání.



17.8. CD-R - Jaký výkon má vypalovací laser? Používá se vždy stejný výkon? Je lepší vypalovat rychle nebo pomalu? Jak volí mechanika optimální výkon laser pro danou rychlosť vypalování? Nezapomeňte, že spirálová stopa je již předlisovaná. Co se děje při vypalování v místě dopadu laseru?

- **CD-R – Recordable – zapisovatelné CD** (jedná rázové).
- Výkon laseru během vypalování **0.5 W** (tedy 1000x více než 0.5 mW), je ale nutné sladit **výkon laseru** při dané **rychlosti záznamu** pro **konkrétní médium**.
- Ve spodní vrstvě CD je již z výroby vytvořena **spirálová drážka** sloužící jako **vodítka pro laserový paprsek**, na disku je nanesena světloodrazivá vrstva ze zlata nebo stříbra.
- Při zápisu se pomocí laseru zahřeje místo, kde má **vzniknout pit** během pár nanosekund na vysokou teplotu ( $300^{\circ}\text{C}$ ).
- Dojde k nenávratné **změně barviva** a toto místo **odráží hůře laserový paprsek**.
- Lepší je **vypalovat pomalu**, aby **vypalovaná data byla včas doručována**.
- Pokud bude rychlosť nižší, výkon laseru bude také nižší.

17.9. CD-RW – jak je dosaženo změn reflexivity? Proč nelze provést neomezený počet přepisů? Jak se maže a navrací do krystalické fáze?

- Obsahuje slitinu, která může nabývat dvou stavů:
  - **Krystalické** – paprsek se odrazí zpět (tedy jakoby **land**).
  - **Amorfní** – paprsek se rozptýlí a neodrazí se (tedy jakoby **pit**).
- Na **prázdném disku** se nachází všude slitina v krystalické fázi.
- V místě, kde má **vzniknout pit**, ohřeje **laser slitinu na teplotu tání** ( $500$  –  $700^{\circ}\text{C}$ ) a ta následně přejde **během ochlazení do amorfní fáze**.
- **Mazání média** se provádí tak, že se slitina zahřeje **pod teplotu tání**, ale nad **krystalizační teplotu** ( $200^{\circ}\text{C}$ ). Po ochlazení je **celé médium smazáno** a lze znova zapsat.
- **Životnost je dána počtem přepisů** – většinou cca **100x**, maximálně 500x.

17.10. Jakou mají CD disky kapacitu a co zde znamená jeden MB?

- Maximálně **700 MB**, většinou **650 MB**.
- Používá se mocnina dvojký, takže ve smyslu  **$2^n$** , tedy **ne  $10^n$** .

17.11. DVD – velikost disku, sektoru, kapacita jednovrstvého a dvouvrstvého lisovaného disku (co zde znamená jeden GB?) Jakou barvu má laser?

- **Velikost disku:** 12 cm, 1.2 mm výška – stejné jako CD.
- **Velikost sektoru:** 2 kB dat (2048 B) - stejné jako CD.
- **Kapacita:** 4.7 GB (jednovrstvý, [DVD5](#)), 8.5 GB (dvouvrstvý, [DVD9](#)), ale už **10<sup>n</sup>**.
- **Laser:** červená barva (635 – 650 nm).

17.12. Jak funguje dvouvrstvé DVD a co to je OTP a PTP? Kde leží vrstvy s pity?

- **Dvouvrstvé DVD** – má dvě vrstvy těsně nad sebou – **poloreflexivní** a **reflexivní**. Na druhé vrstvě ovšem není tolik dat.
- **Vrstva s pity leží uprostřed** (0.6 mm pod povrchem).
- **PTP – Parallel Track Path** – Na obou vrstvách začíná záznam uprostřed a pokračuje k okraji.
- **OTP – Opposite Track Path** – Stopa na druhém povrchu začíná na vnějším okraji a čte se směrem ke středu. Běžné pro video.

17.13. Jak to bylo historicky s DVD-R a DVD+R?

- Protože firma Pioneer si **nechala patentovat** [DVD-R](#) a [DVD-RW](#), musely firmy jako Sony, Philips, HP a další platit svému přímému konkurentovi.
- Proto vznikla [DVD+](#) aliance, které vytvořily konkurenční technologii [DVD+R](#) a [DVD+RW](#).
- Používané jsou oba typy (+ i -) stejně, některé přehrávače ale dříve podporovaly jen jeden typ.

17.14. Jaká rychlosť (datový tok) je zde považována za základní?

- **Rychlosť** (pro násobky) **1350 kB/s.**
- Rychlosť není ale odvozeno od žádné základní rychlosti – každé video má jiný datový tok.

17.15. Vyplatí se vypalovat dvouvrstvá DVD?

- Ne, protože 8.5 GB dvouvrstvý disk je **2.5x dražší** než 4.7 GB jednovrstvé disky, přitom kapacita ani není dvojnásobná. Pouze vhodné, pokud chceme data mít vcelku (film).

17.16. Jaký kodek a souborový formát se používá pro uložení videa na DVD?

- Používá se kodek **MPEG2**.
- Pro všechny data se používá soubor formátu **VOB** – má v sobě informace o **menu rozhraní, možnostech titulků, bonusových záběrech, více zvukových stopách** apod.

17.17. DVD RAM – v čem se zásadně liší? Co umožňuje na rozdíl od DVD+RW?

- **Přepisovatelný disk.**
- Chová se stejně **jako pevný disk, data jsou ukládána do sektorů umístěných ve stopách**.
- Mnohem lépe vyřešená korekce chyb, volné místo je okamžitě dostupné, až sto tisíc přepisů.

17.18. BluRay – rozměry, kde leží vrstva s pity, kapacita, rychlosť čtení, vlnová délka laseru (na co má vliv, proč zde musí být modrý).

- **Velikost disku:** 12 cm, 1.2 mm výška – stejné jako CD / DVD.
- **Pity:** uloženy 0.1 mm pod povrchem (1.1 mm od etikety).
- **Kapacita:** 25 GB (jednovrstvý), 50 GB (dvouvrstvý), takéž **10<sup>n</sup>**.
- **Laser:** fialová barva (405 nm) – **čím má paprsek kratší vlnovou délku, tím lépe lze zaostřit** – dovoluje tedy použití menších pitů.
- **Dosažitelný datový tok až 54 MB** (jednorychlostní 4.5 MB/s).

17.19. Použití, historie.

- Původně BluRay určen pro záznam FullHD videa, používal se novější kodek H.264.
- Byl obvyklý pro přenos velkých filmů natočených ve FullHD.
- Stále se používá pro herní konzole (PlayStation 4 a PlayStation 5).
- **BD-ROM** – disk pouze pro čtení.
- **BD-R** – disk pro jednorázový zápis.
- **BD-RE** – přepisovatelný disk.

17.20. Aktuální ceny.

- **BD-R** – 1 – 2 Kč / GB, **BD-RE** – 1.5 – 3 Kč / GB.

17.21. Pravděpodobná budoucnost optických mechanik a médií...

- Pro počítače se prakticky CD, DVD ani BluRay dnes již nepoužívají.
- Flash média jsou příliš levná a výhody Cloudových a streamovacích služeb vytlačily tyto technologie.
- Firmy příliš nemají důvod do těchto technologií investovat a vylepšovat je.

## 18. Napájení počítače

- Napájecí zdroje (funkce, parametry, linky, konektory, spínaný napájecí zdroj)
- Akumulátory
- Záložní zdroje energie (UPS)

18.1. K čemu slouží napájecí zdroj, co je na vstupu a co na výstupu (stejnosměrné / střídavé napětí, frekvence, velikosti napětí).

- V elektrické sítí v Evropě je **střídavé napětí** 230 V o 50 Hz (v USA 110 V o 60 Hz). Počítačové komponenty ovšem potřebují **nízké stejnosměrné** napětí.
- **Na výstupu zdroje najdeme typicky tyto hodnoty:**
  - **+3.3 V** – proud až desítky ampér, využívá většina moderních obvodů (**oranžová**).
  - **+5 V** – proud až desítky ampér, pro některé pevné desky, přídavné karty, USB (**červená**).
  - **+12 V** – proud jednotky ampér, pro pevné disky, optické mechaniky, ventilátory (**žlutá**).
  - **-12 V** – proud menší než jeden ampér, pro speciální komponenty (stará sériová linka) (**hnědá**).
  - **5 V StandBy** – napětí +5 V i při vypnutém počítači (**fialová**).
  - **PS\_ON** – SW vypnutí / zapnutí počítače (**zelená**).
  - **Power Good** – slouží k informování základní desky o zahájení činnosti (**bílá**).

18.2. Jaké napětí potřebují komponenty uvnitř počítače (stejnosměrné, nízké!).

- Stejnosměrné nízké, viz předchozí otázka.

18.3. Používané napěťové úrovně - Zapamatuje si barvy vodičů.

- Viz dva body zpět.

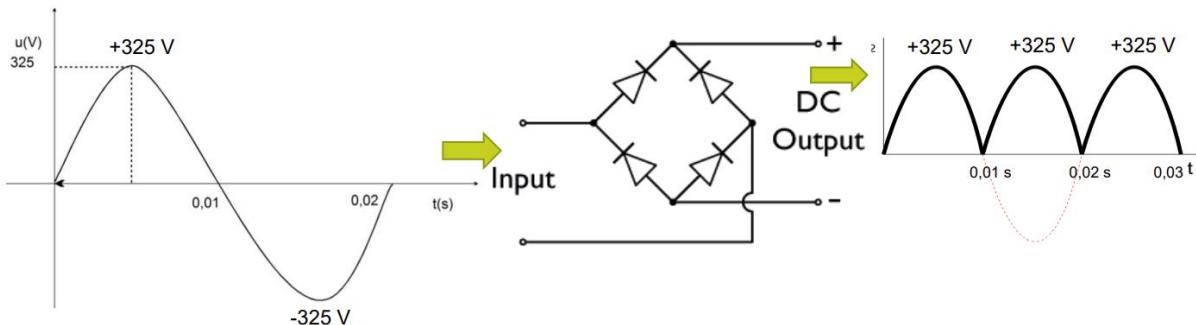
18.4. Proč se nepoužívají jednoduché transformátory.

- **Transformovat se dá pouze střídavé napětí!**
- Transformátor se skládá ze dvou cívek umístěných na společném feromagnetickém jádru.

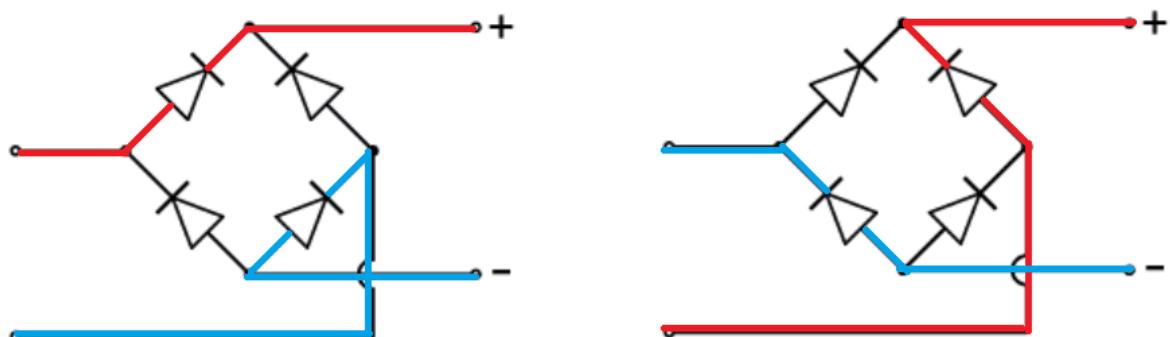
- Transformátory jsou ovšem velké, těžké a neefektivní (až 70 % ztrát), proto se pro moderní hardware nepoužívají.

18.5. Jak funguje spínaný zdroj? Je třeba znát základní princip a postup, co se tam při transformaci napětí děje a vědět, proč se mu říká "spínaný". Není nutné umět podrobně nakreslit všechny detaily.

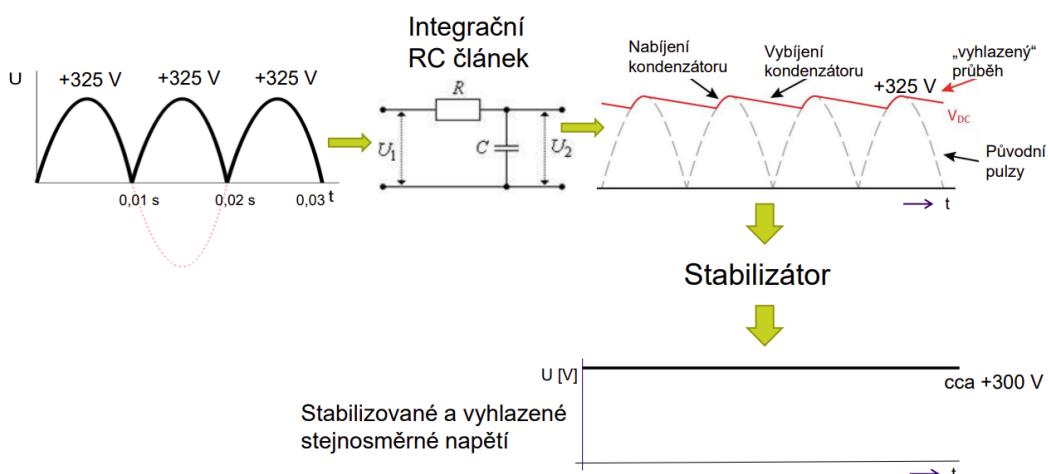
- Napětí ze sítě je ve spínaném zdroji rychlým **spínáním** / vypínáním převedeno na impulsy s vysokou frekvencí.
- Napětí v elektrické síti má amplitudu **325 V** (efektivní hodnota  $230 \text{ V} * \sqrt{2}$ ).
- Použijeme diodový můstek, který přemění napětí stejnosměrné.



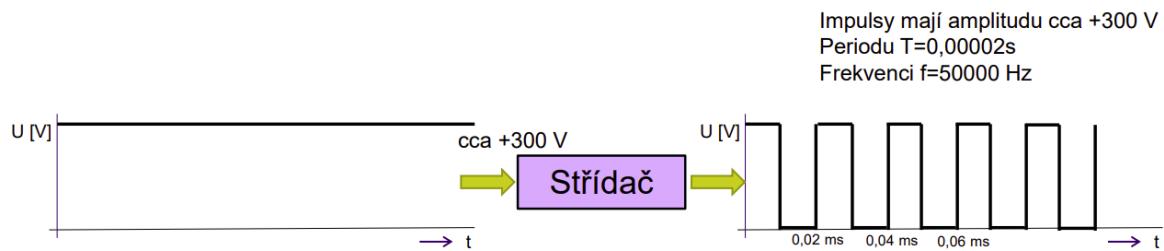
**Diodový můstek může být ve dvou stavech:**



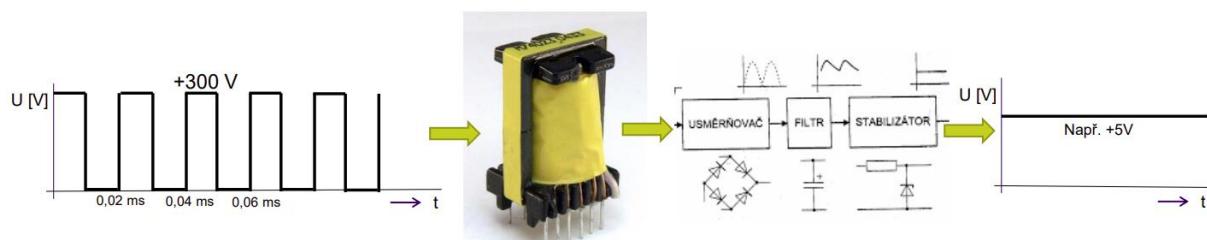
- Následně integrační článek a stabilizátor napětí vyhladí:**



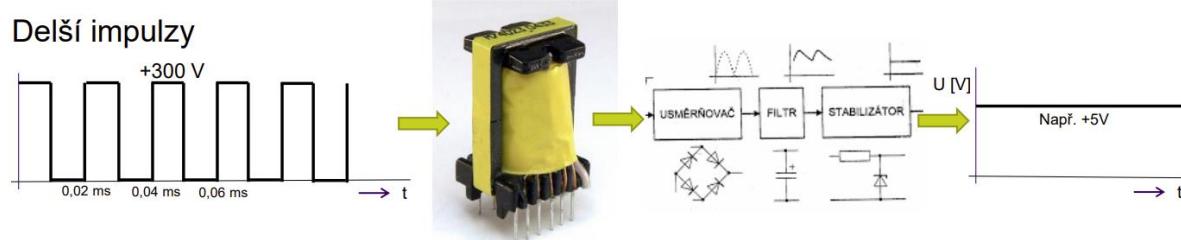
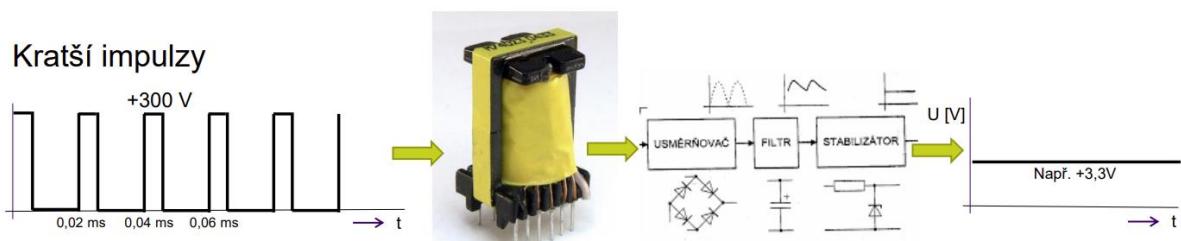
- Střídač následně toto vyhlazené napětí převede **opět na střídavý signál**, nyní ovšem s velmi vysokou frekvencí (50 kHz).



- Tento vysokofrekvenční signál je převeden na **impulsní transformátor**.
- Ten z tohoto napětí udělá **nízké střídavé vysokofrekvenční výstupní napětí**.
- Tato nízké vysokofrekvenční výstupní napětí se opět **usměrní, vyhladí a stabilizuje**, čímž vznikne nízké stejnosměrné napětí.



- Výstupní napětí lze ovlivňovat podle toho, jak dlouhé jsou impulzy před **impulsním transformátorem**.



## 18.6. Základní parametry napájecího zdroje?

- **Dodávaný výkon** – maximální odebíraný výkon zařízením, ve Wattech [[W](#)], typicky 350 – 800 W.
- **Sítové napětí a frekvence** – pro EU / USA.
- **Rozsah provozních teplot.**
- **Počet výstupních linek a kabelů.**
- **Rozměry a hmotnost** – standardní [ATX](#), dále [SFX](#) a [TFX](#).
- **Účinnost** – nejlevnější 80 %, kvalitní 95 %, certifikace 80 PLUS (Bronze – Titanium, měří se při 50 % zařízení).
- **Chlazení** – hlučnost, velikost ventilátoru.
- **Odpojitelné kabely** – zdroj je pak modulární.

## 18.7. Kvalita napájecího zdroje – ukázat, co je zatěžovací charakteristika, zvlnění napětí, odolnost vůči krátkému výpadku.

- **Zatěžovací charakteristika:**
  - **Udává závislost výstupního napětí na zatížení** (odebíraném proudu).
  - **Například:** U Tvrdého zdroje při odběru [2 A](#) klesne výstupní napětí z [12 V](#) na [11 V](#), za to u měkkého zdroje klesne na [9 V](#).
  - U počítačových linek by mělo platit, že **tolerovaná mez** bez ohledu na zátěž je maximálně **± 5 %** (u linky  $\pm 12 \text{ V} \pm 10 \%$ ).
- **Zvlnění napětí:**
  - Kvůli tomu, že, že je zdroj spínaný, **není hodnota výstupního napětí v čase stálá**, ale obsahuje zvlnění.
  - **Zvlnění větší než norma** může mít vliv na **stabilitu PC**.
  - Zvlnění lze změřit **osiloskopem**, ne ale voltmetrem.
  - **Norma:**
    - **120 mV** (pro [+12 V](#) a [-12 V](#)).
    - **50 mV** (pro [+5 V](#), [+ 5 V StandBy](#) a [+3.3 V](#)).
- **Odolnost vůči krátkému výpadku:**
  - Podle normy alespoň **17 ms** při úplném výpadku.
  - Dáno především **kapacitou vstupního kondenzátoru** – čím větší, tím lepší.

- Je taky vhodné posuzovat, jak se zdroj zachová při okamžitém zvýšení odběru – například při zatížení GPU by mělo napětí zůstat v tolerancích.

#### 18.8. Proč si často musíme i při nízkém příkonu pořídit silnější zdroj – rozdelení výkonu do jednotlivých linek.

- Dodávaný výkon **není rozdelen rovnoměrně do všech linek**.
- **Například na 200 W zdroji se často setkáme s tímto rozdelením:**
  - **5 V proud 12 A** - na této lince lze odebírat výkon **60 W**.
  - **3.3 V proud 12 A** - na této lince lze odebírat výkon **50 W**.
  - **12 V proud 8 A** - na této lince lze odebírat výkon **96 W**.
- Proto pokud budeme mít třeba **mnoho USB zařízení**, které odebírají větší proud než **60 W**, budeme si muset pořídit silnější zdroj, i když celkový odběr není větší jak **200 W**.

#### 18.9. Zapínání a vypínání počítače, linka 5V Standby a PowerGood.

- **Dříve:** zapínací tlačítko přímo připojeno na 230 V vstup zdroje.
- **Dnes:** zdroje neustále pod napětím, i když je počítač vypnutý. Spojením zeleného vodiče se zemí dojde k jeho zapnutí. Lze ale zapnout i například přes LAN.
- **StandBy**
  - Neustále poskytuje napájení 5V i při vypnutém počítači, nutné pro Wake On Ring apod.
- **Power Good (PWR OK)**
  - Po zapnutí zdroje je nutné cca 0.5 sekundy počkat na **ustálení úrovně všech napájecích linek**.
  - Dokud není signál **Power Good** aktivní, je znemožněn start počítače.
  - Pokud není **Power Good** signál aktivní kdykoliv v průběhu práce, čip časovače začne **automaticky resetovat procesor** a tím mu zabrání fungovat za špatných nebo nestabilních podmínek.

18.10. Napájecí konektory – P1, Molex, SATApower, přídavné napájení procesoru a grafiky. Není potřeba znát přesný význam jednotlivých pinů na konektoru, ale konektor musíte umět popsat.

- **P1:**
  - **20 / 24 pinů.**
  - Slouží k **napájení základní desky**.
  - 7x GND, 4x 3.3 V, 5x +5 V, 2x +12 V.
- **Molex:**
  - Dříve pro napájení pevných disků, mechanik či řadičů ventilátorů apod.
  - Konektor má 4 piny: 2 x GND, +5 V, +12 V.
- **SATApower:**
  - Nástupce Molexu, napájení pevných disků.
  - Menší rozměr.
  - **15 pinů** – 3x 12 V, 3x 5 V, 3x 3.3 V, 5x GND.
- **Přídavné napájení procesoru:**
  - Jde přes základní desku, samotný procesor není přímo napájen.
  - 4 / 8 pinů.
  - Vždy napětí 12 V.
- **Přídavné napájení GPU:**
  - **6 pinový** konektor.
  - **2x6 pinový** konektor.
  - **8 pinový** konektor.
  - Vždy napětí 12 V.
  - Některé grafické karty dokáží využít až 4x 8 pinový konektor.

18.11. Co je to napájecí kaskáda? Napájecí napětí moderních mikroprocesorů...

- Snižuje napětí které přichází z napájecího zdroje na základní desku na úroveň potřebnou pro procesor.
- **Procesor** – velmi nízké napájení (např.: 1.15 V), vysoký proud (např.: 50 A).
- Napětí a proud **transformuje napájecí kaskáda**.
- **Na vstupu** – 12 V / 5 A.
- **Na výstupu** – 1.15 V / 52 A.

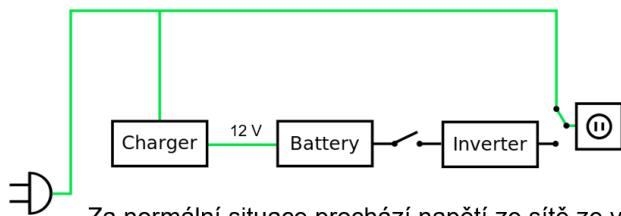
- **Napájecí kaskáda je tvořena několika MOSFET tranzistory, cívkami a kondenzátory.** Čím více cívek, tím vyšší proudy bude zvládat dodávat a transformace budou účinnější.
- **Nabíjením a vybíjením** kondenzátorů **velmi rychle spínaným napětím** a důmyslným zapojením cívek lze získat velmi vysoký proud, který procesor potřebuje.
- Napájecí kaskáda je často potřeba chladit.

18.12. UPS – vysvětlit, k čemu to je dobré, jaký typ akumulátorů se tu používá, jak může být propojený s počítačem, aby ho informoval o výpadku nebo jak informovat správce počítače.

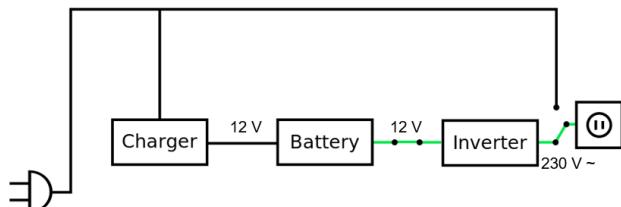
- **Uninterruptible Power Supply.**
- Záložní zdroj napájení – **ochrana proti výpadku napájení.**
- V případě výpadku energie se používá 12 V stejnosměrné napětí z **оловěného akumulátoru**, které je spínaným zdrojem (**měničem**) převedeno do **vysoké střídavé** – 230 V.
- Kvalitní obsahují komunikační rozhraní – například USB a informují zařízení při výpadku.
- Zařízení na něm může zareaguje různě – uloží důležitá data, ukončí aplikace a případně se i **připraví k vypnutí.**
- Může i informovat pomocí SMS zprávy či výstražného signálu.

18.13. Offline UPS – nakreslit obrázek, vysvětlit nevýhody.

- **Nejjednodušší princip.**
- Napětí z rozvodné sítě **přímo** prochází **na výstup UPS.**
- V případě přerušení napájení se **přepne na výstup** z měniče a **akumulátorů.**
- **Prodleva** při přepnutí **okolo 20 ms** (tentot čas většinou poskytuje napájení kondenzátory zdroje).
- **Není schopný úpravy podpětí nebo přepětí v rozvodné síti.**



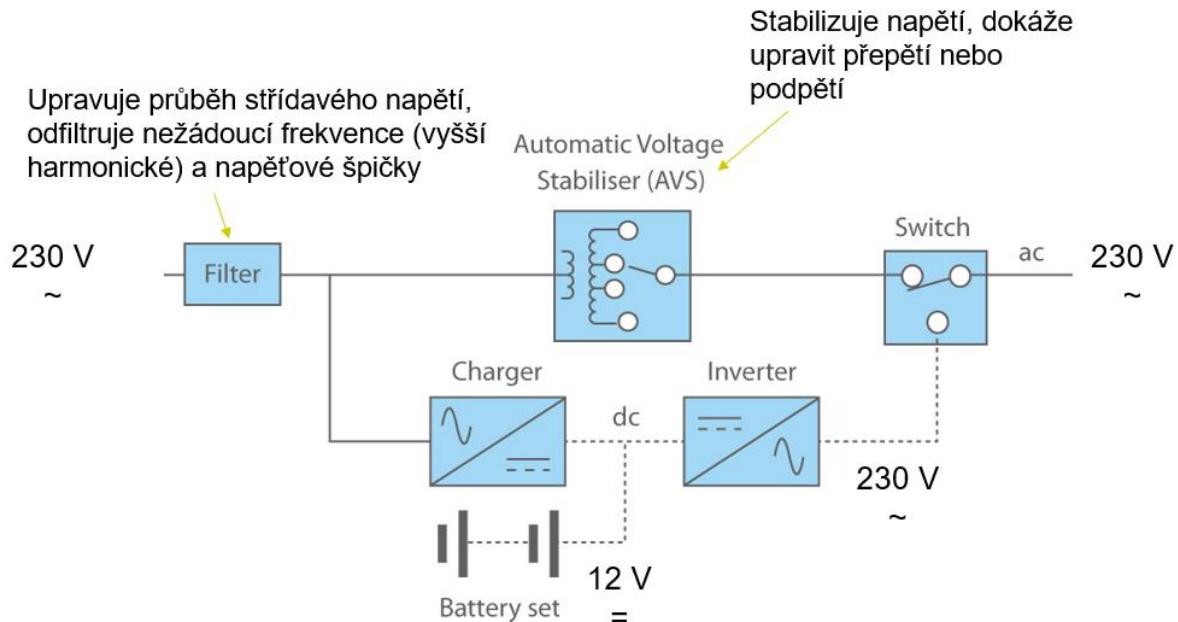
Za normální situace prochází napětí ze sítě ze vstupu rovnou na výstup a vedle toho nabíječka udržuje nabité akumulátor



Při výpadku dodávky energie se automaticky přejde na napájení z akumulátoru

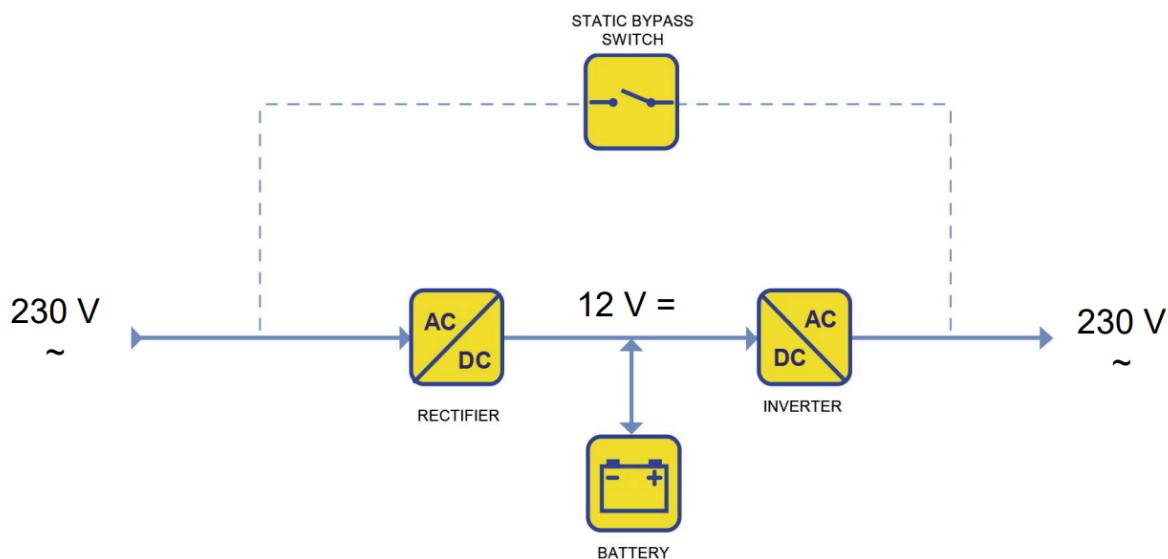
#### 18.14. Line-interactive UPS – nakreslit obrázek, vysvětlit výhody.

- Dokáže i stabilizovat **výstupní napětí**, aby se co nejvíce blížilo předepsanému napětí – obsahuje **stabilizační obvody**.
- Posílení se jmenuje **boost**, potlačení vyššího napětí **buck** nebo **trim**.
- Prodleva při přepnutí je 4 – 10 ms.



### 18.15. Dvojitá konverze – nakreslit obrázek, vysvětlit výhody.

- **Online UPS = s dvojitou konverzí.**
- Vstupní napětí se nejprve **sníží na 12 V stejnosměrných** a následně se střídačem **opět zvýší na střídavé výstupní napětí 230 V.**
- Při jakémkoliv zkreslení či výpadku vstupního napětí **nevzniká žádná prodleva.**
- Vhodné pro **nestabilní prostředí** a tam, kde by přepnutí mělo fatální následky.
- Taktéž obsahuje tzv. **bypass**, který v případě selhání elektroniky UPS dokáže **přímo propojit vstup a výstup.**



### 18.16. Vysvětlit pojem primární článek, sekundární článek, baterie, kapacita, vnitřní odpor.

- **Primární článek** – napětí článku se **nedá po vybití obnovit**, chemický proces je nevratný. Ve výpočetní technice se moc nepoužívá (mimo **CMOS** baterií).
- **Sekundární článek** – článek se dá po vybití **znova nabít**, chemický proces je tedy **obousměrný**. Nelze to ovšem dělat donekonečna. Obvykle nazýván jako **akumulátor**.
- **Baterie** – zdroj elektrické energie realizovaný jako sada sériově spojených elektrických článků. Dnes se ale tímto pojmem označuje i jeden samostatný článek (například tužková baterie). Zapojení jak do série, tak i paralelně snižuje jejich životnost a kvalitu.
- **Kapacita** – viz dále.

- **Vnitřní odpor** – čím nižší, tím lepší (tvrdší) zdroj bude – bude schopen dodávat vyšší proud při zatížení a nebude klesat jeho napětí. Označuje se jako  $R_i$ . Vzorec  $I_{max} = U / R_i$ . Nemůže být samozřejmě nulové.

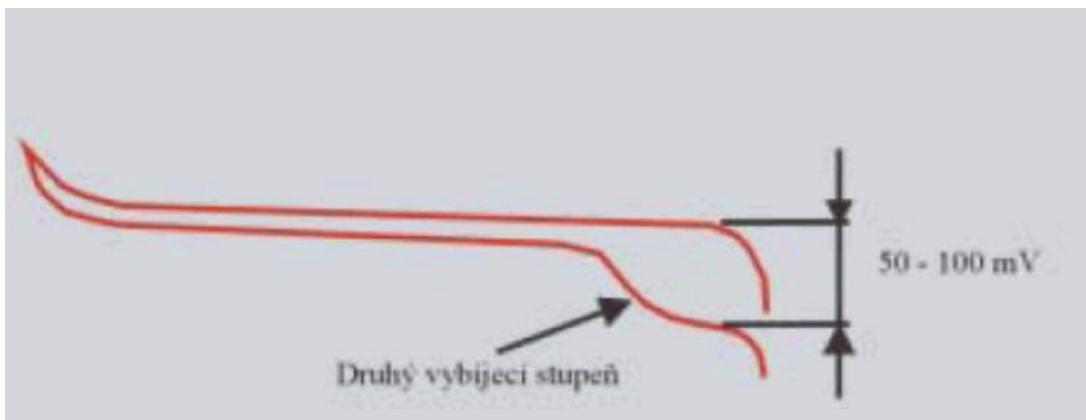
18.17. Jakými způsoby se udává kapacita akumulátorů? Jak se spočítá, za jak dlouho se vybije? Jak se převádí mezi Wh a Ah?

- Kapacita akumulátoru se udává v **mAh** nebo **Ah** (**1 Ah = 1000 mAh**).
- Kapacita vyjadřuje **součin proudu a času** během kterého se akumulátor vybije.
- Akumulátor **1 Ah** může dodávat **1 A** po dobu jedné hodiny.
- Pokud se akumulátor vybije rychleji než za hodinu, používáme značku **C**.
- **6C** znamená, že akumulátor se vybije za **1/6 hodinu** (10 minut).
- Pokud má akumulátor kapacitu **1000 mAh** a maximální rychlosť vybíjení je **10C**, pak znamená, že maximálně dokáže poskytnout  $1000 * 10C = 10000$  mA = **10 A**.
- **Wh** (Watthodiny) lépe vyjadřují skutečné množství energie uložené v akumulátoru – bere do úvahy i napětí.
- Akumulátor **1 Wh** může dodávat **1 W** po dobu jedné hodiny.
- Pokud adaptér má kapacitu **2400 mAh** a napětí **1.2 V**, tak jeho kapacita je  $2400 * 1.2 = \mathbf{2.88 Wh}$ .
- Pokud máme k dispozici pouze proud a napětí, musíme mít k dispozici ještě informaci **o času!**
- Ohmův zákon:  $U = R * I$ .

18.18. NiCd – Proč už se nepoužívají, co to je paměťový efekt, jaké měly výhody a nevýhody.

- **Niklo-kadmiový akumulátor.**
- Obsahuje jedovaté **Kadmium**.
- Plně nabity akumulátor má **1.35 V**, poté běžně **1.22 – 1.25 V**.
- Obvykle se nabijí proudem **0.1 – 0.3 C**.
- Do článku je při nabijí nutné dodat **1.4 násobek ukládané energie**.
- Rychlonabíječky dokáží nabít až proudem **1 – 2C**, ty ale musí zároveň sledovat nabítí článku. To většina levnějších neumí.

- **Výhody** - nevadí mu úplné vybití. Možnost nabíjet vysokým proudem, dobře pracuje v mrazu. Životnost 1000 nabijecích cyklů.
- **Nevýhody** – samovybíjení. Paměťový efekt (musí se plně nabíjet a plně vyvíjet). Obsahuje jedovaté kadmium, nelze zjistit stupeň nabití (většinou je okolo 1.2 V).
- **Paměťový efekt** – druhý vybíjecí stupeň. Pokud se akumulátor vyvíjí na středí úroveň vybití, vznikne **druhý vyvíjecí stupeň**, který značí **náhlý pokles napětí** o **50 až 100 mV**. Odstranitelné plným vybitím akumulátoru.



18.19. NiMH – Kde se dnes hlavně používají a jak vypadají. Jmenovité napětí, běžná kapacita AA článku, obtížné určení úrovně vybití. Životnost a hypercykly.

- **Niklo-metal-hydridový akumulátor.**
- **Nástupce NiCd** – neobsahuje nebezpečné kadmium.
- Použití – tužkové baterie.
- Napětí **1.2 V**, nabijí se na **1.4 V**.
- **Výhody** – vyšší kapacita než NiCd (kapacita běžně 2700 mAh), umožňuje dodávat vysoký proud, možnost rychlonabíjení (obvykle 2.5 hodiny), stále napětí **1.2 V**.
- **Nevýhody** – samovybíjení, nefunguje při nízkých teplotách, nižší životnost než NiCd (500 cyklů), pokud necháme NiMH déle bez nabití, dochází k nenávratným reakcím a ztrátě kapacity, nelze určit, jak je vybitá.
- **Hypercykly** – pokud bychom NiMH vybíjeli z 55 % na 45 % a pohybovali se v tomto rozsahu, vydrží akumulátor 300 000 částečných cyklů (to odpovídá 30 000 plných cyklů).
- Je ovšem složité určit, jak je NiMH přesně **nabitý / vybitý**. Družice, které této vlastnosti ovšem využívají, dokáží **přesně změřit svůj odběr**, a tak článek v nich vydrží velmi dlouho.

18.20. Li-Ion – tvar a podoba, jmenovité napětí, nabíjecí napětí. Možné nebezpečí, zabudovaná ochrana. Kde se používá. Životnost.

- **Lithium iontový akumulátor.**
  - Nejpoužívanější v mobilním HW.
  - **Vysoké jmenovité napětí 3.7 V.**
  - Životnost asi 1000 cyklů, při rychlém nabíjení ale i 150.
  - Vysoká energetická hustota, téměř **zádné samovybíjení**.
  - **Žádný paměťový efekt.**
  - **Nabijí** se napětím **4.2 V**, při **vybíjení zvolna klesá** z **4.2 V** na **2.8 V**.
  - Elektronika se většinou při poklesu pod **3.5 V** **sama vypne** (nebylo by už dost stabilní).
- 
- Novější mají pozměněné chemické složení a nabíjejí se až na **4.35 V**.
  - **Nabíjení se ukončí samo** – prostě do něj nepoteče žádný další proud.
  - Typická doba nabíjení 1.5 – 3 hodiny, lepší pomalé nabíjení – bude mít delší životnost.
  - **Nevýhody – nesmí se úplně vybit – pod 2.8 V dojde ke zničení** (má ale zabudovanou elektroniku, ale samovolně při nepoužívání se může i sám vybit), **sám stárne** (i když se nepoužívá), **není příliš vhodné tam, kde se rychle vybijí** (RC modely), **nebezpečí výbuchu při zkratu nebo mechanickém porušení**.

18.21. Li-Pol – v čem jsou nejlepší? V čem jsou lepší než Li-Ion? Proč se používají méně? Jakou mají životnost?

- **Lithium-polymerový akumulátor.**
- Inovace Li-Ion technologie.
- **Jmenovité napětí 3.7 V.**
- Životnost až **2000 cyklů**, při pomalém nabíjení jen cca 150.
- **Méně se zahřívá**, dokáže dodat **extrémně vysoký proud** (50 C).
- Menší než Li-Ion.
- Menší nebezpečí než Li-Ion.
- **Nevýhody** – po několika měsících nečinnosti přestane fungovat, vysoká cena, jinak podobné nevýhody jak Li-Ion.

18.22. Dejte si pozor na to, že nabíjecí napětí a jmenovité napětí není totéž.

- **Jmenovité napětí** – napětí ke kterému se vztahují provozní charakteristiky akumulátoru, slouží při výpočtech, udáváno výrobcem.
- **Nabíjecí napětí** – napětí plně nabitého / nabíjejícího akumulátoru.

18.23. Jak rychle se dá akumulátor vybít? Co to znamená např. 30C?

- Pokud se akumulátor vybije rychleji než za hodinu, používáme značku **C**.
- **30C** znamená, že akumulátor se vybije za **1/30 hodinu** (2 minuty).
- Pokud má akumulátor kapacitu **1000 mAh** a maximální rychlosť vybíjení je **30C**, pak znamená, že maximálně dokáže poskytnout  $1000 * 30C = 30\ 000$  mA = **30 A**.

18.24. LiFePO – Kde se s ním setkáme?

- **Lithium-železo-fosfátový (LiFePO<sub>4</sub>) akumulátor.**
- Netoxický, nehoří, neexploduje, nepřehřívá se.
- Má horší energetickou hustotu, proto se nehodí do mobilního HW.
- Jmenovité **napětí 3.3 V**, nabíjecí **3.6 V**.
- **Životnost** - 2000 – 7000 cyklů do 80 % původní udávané kapacity.
- **Použití** - elektromobily, domácnosti se solárními panely.

18.25. Jak lze prodloužit životnost Li-ion a Li-Pol? Co jim velmi vadí? Co může způsobit vadná nabíječka?

- **Vadí** – vysoká teplota, vysoká rychlosť nabíjení, nabíjení z 0 % na 100 %.
- **Vadná nabíječka** – může vést k přebíjení – hrozba exploze akumulátoru.

18.26. Porovnejte parametr kapacita/hmotnost pro různé typy akumulátorů.

Typ akumulátoru	Chemická značka / typ	Jmenovité napětí	Nabíjecí napětí	Životnost	Kapacita na hmotnost
<b>Niklo kadmiový</b>	NiCd	1.2 V	Až 2C	1000 cyklů	50 Wh / kg
<b>Niklo metal hydridový</b>	NiMH	1.2 V	1.4 V	500 cyklů	120 Wh / kg
<b>Lithium iontový</b>	Li-Ion	3.7 V	4.2 / 4.35 V	1000 cyklů	150 Wh / kg
<b>Lithium polymerový</b>	Li-Pol	3.7 V	4.2 V	2000 cyklů	200 Wh / kg
<b>Lithium železo fosfátový</b>	LiFePO <sub>4</sub>	3.3 V	3.6 V	až 7000 cyklů	90-110 Wh / kg

Kapacita na hmotnost – **VYŠŠÍ HODNOTA JE LEPŠÍ.**