

Číslicové počítače (Podporné učebné texty)

Úvod.

Texty sú venované predmetu “Číslicové počítače”. Obsahom i spôsobom spracovania sú určené skôr na podporu predmetu a nie ako náhrada prednášok. Ak napriek mojej snahe objavíte v nich chybné informácie, budem vďačný za kritiku na adrese pgubis@etc.sk.

1. Zásady dokumentácie elektronických systémov

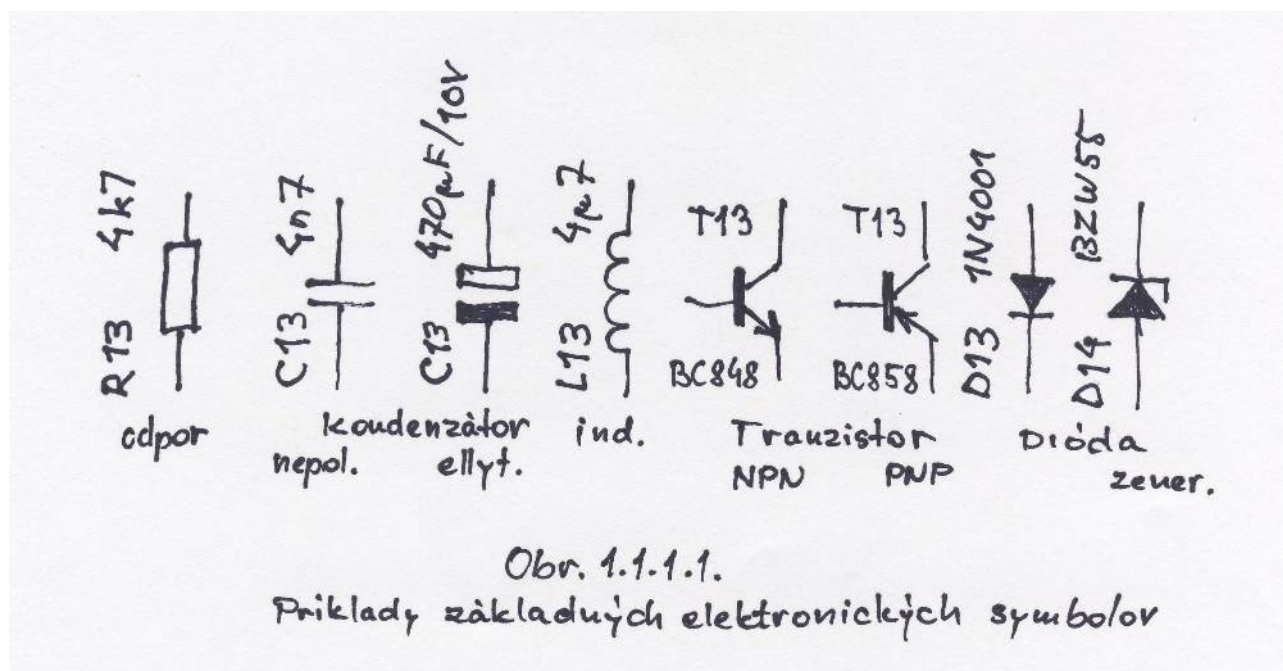
Podstatnými časťami dokumentácie elektronických systémov sú dva dokumenty. Schéma zapojenia a osadzovací výkres.

1.1. Schéma zapojenia

Schéma zapojenia má podať čo najprehľadnejšiu informáciu o spôsobe prepojenia elektronických súčiastok systému. Spôsob kreslenia a rozloženie súčiastok obvykle neodzrkadľuje tvar a spôsob rozloženia prvkov v reálnom systéme. Schéma však musí byť vybavená informáciami, podľa ktorých je jednoznačne určený vzťah schémy a osadzovacieho výkresu a tým aj fyzickej realizácie systému. Prvoradou požiadavkou na schému je jej dobrá čitateľnosť a prehľadnosť.

1.1.1. Symboly súčiastok

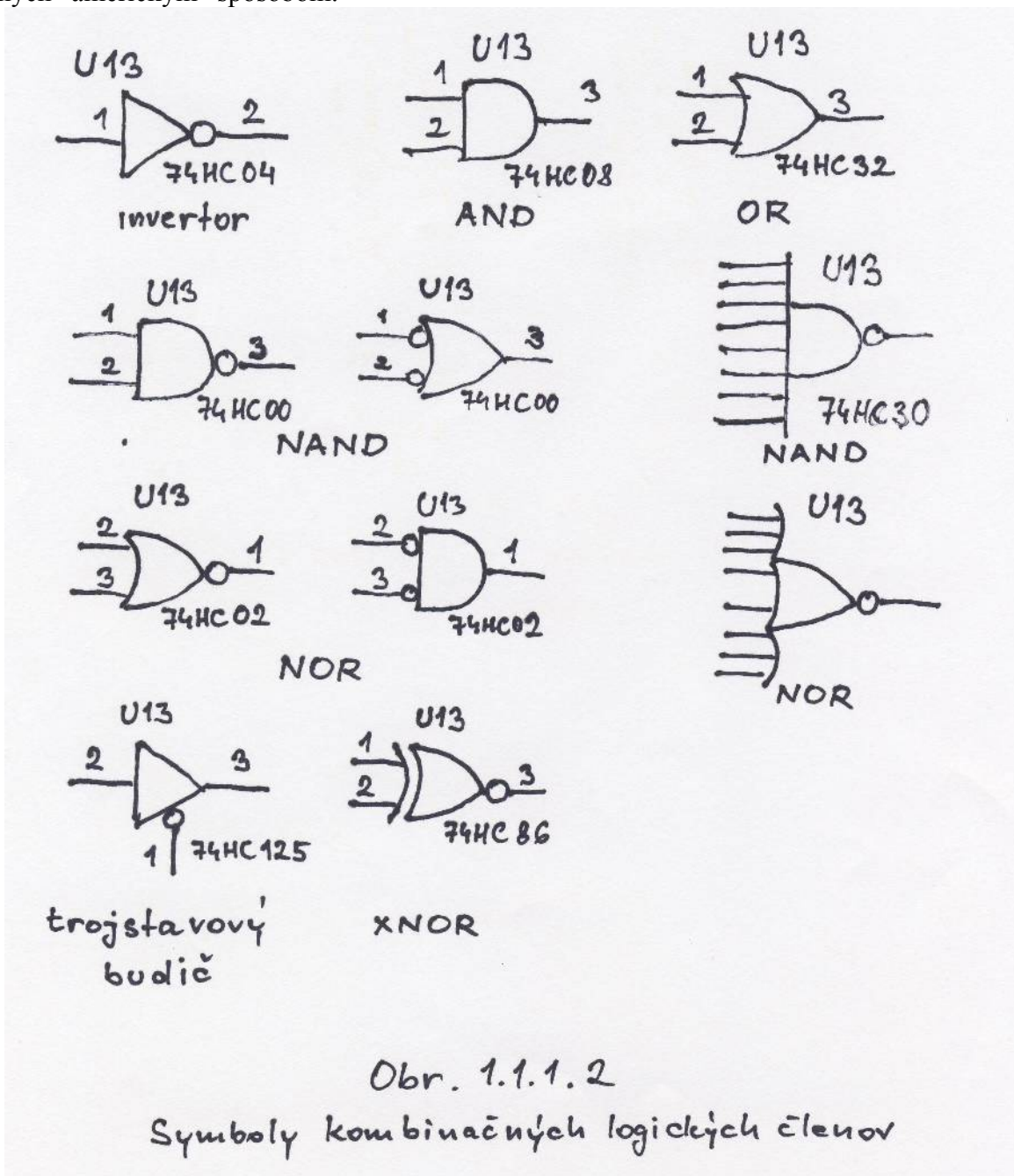
Spôsob kreslenia symbolov súčiastok závisí od ich zložitosti. Súčiastky, ktorých správanie je dobre definované a známe sa kreslia všeobecne používanými symbolmi. Typickými predstaviteľmi tejto skupiny sú základné elektronické súčiastky (odpory, kondenzátory, indukčnosti, tranzistory,viď obr. 1.1.1.1.) ale aj jednoduché kombinačné a sekvenčné prvky (logické členy typu AND, OR, NAND,Klopné obvody s individuálnym ovládaním D, JK,.....) ako i trojstavové hradlá, ktoré sa dajú ovládať individuálne (74125, 74126,.....). Pri kreslení kombinačných logických členov je možné použiť rôzne symboly vychádzajúce z De Morganových pravidiel.



Najpoužívanějšíe sú tzv. Americké symboly, kde platia nasledujúce zásady:

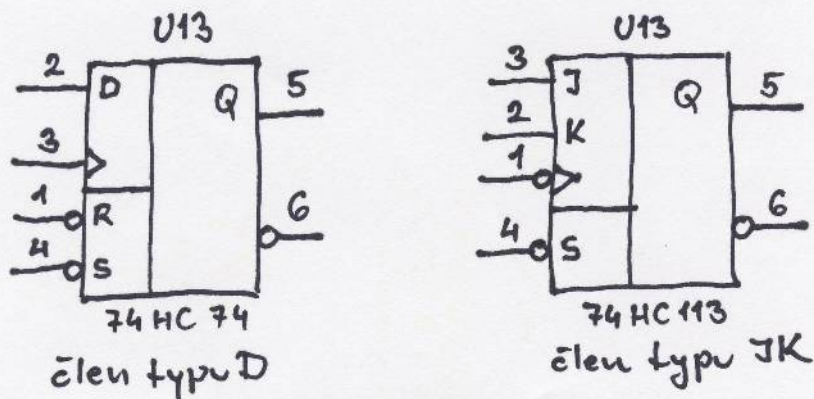
1. Používajú sa tri základné tvary súvisiace s booleovskými funkciami AND, OR a INV. Všetky symboly sú z nich odvodené.
2. Gulička na vstupe znamená, že príslušný vstupný signál je aktívny v nule (nahradzuje invertor zapojený v ceste ku vstupu).
3. Gulička na výstupe znamená, že príslušný výstup je invertovaný (ako by bol na výstupe v ceste signálu zaradený invertor).

Na obrázku 1.1.1.2. sú príklady kreslenia symbolov kombinačných logických členov kreslených "americkým" spôsobom.



Sekvenčné logické členy sa kreslia ako obdĺžniky s naznačenými vstupmi a výstupmi, pričom vstupy sú obvykle kreslené z ľavej a výstupy z pravej strany. Guličky na vstupoch a výstupoch majú

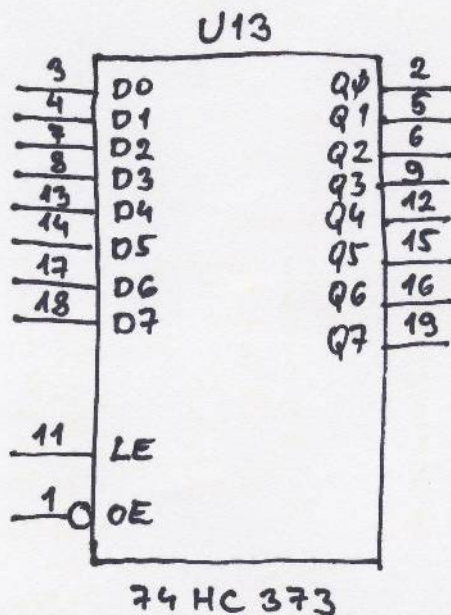
rovnaký význam ako tie v prípade symbolov kombinačných logických členov. Okrem týchto doplnkov sa môžu vyskytnúť na vstupoch aj trojuholníky, ktoré symbolizujú vstup s krátkym vzorkovaním (citlivý na zmenu (hranu) signálu z 0 na 1 (nábežná hrana)). Gulička má v tomto prípade funkciu zmeny smeru citlivosti). Na obrázku 1.1.1.3. sú príklady symbolov elementárnych sekvenčných členov.



Obr. 1.1.1.3.

Príklady základných sekvenčných log. členov.

Symbols logických obvodov väčšej zložitosti nie sú pevne definované. Návrhár ich môže vytvárať, alebo prevziať z iného zdroja. Zväčša je to obdĺžnik so vstupmi usporiadanými na ľavej a výstupmi na pravej strane. Všetky prvky popísané v predchádzajúcom môžu byť pri kreslení použité. Príklad symbolu zložitejšieho integrovaného obvodu je na obrázku 1.1.1.4.



74 HC 373

Príklad symbolu zložitého int. obvodu.

Obr. 1.1.1.4.

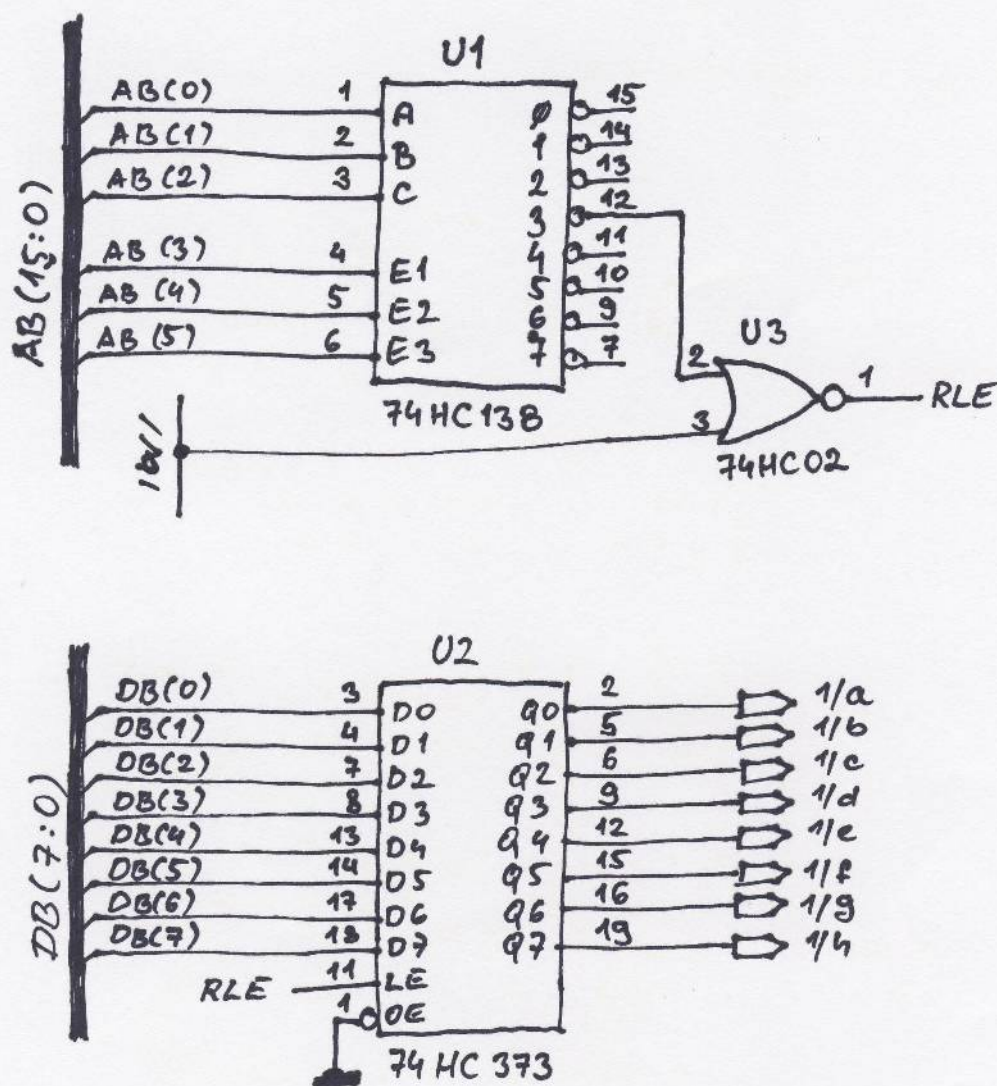
V predchádzajúcom popise sme predpokladali, že všetky vývody logických členov súvisia s ich funkciou (sú buď vstupy, alebo výstupy). Integrované obvody však musia mať aj napájacie vývody a na

púzdrach v ktorých sú zapúzdené môžu byť aj nevyužitý vývody, prípadne vývody, ktoré majú technologický, alebo diagnostický význam, ale pre bežného používateľa sú nepoužiteľné.

Väčšinou sa napájacie vývody logických členov v schémach nekreslia. Ak sa z nejakých dôvodov kreslia, potom sa odporúča označovať: spoločný vodič napájania ako GND a kladný pól napájania ako Vcc. Nepripojené vývody sa tiež obvykle nekreslia. Ak sú nakreslené označujú sa obvykle symbolom NC (Not Connected). Vývody s užívateľsky nepoužiteľnou funkciou sa označujú tiež symbolom NC, ale v katalogovom liste sú uvedené, ako "Do Not Connect".

1.1.2. Prepájanie symbolov

Prepájanie symbolov je možné pomocou čiar, pomocou zberníc, prípadne asociáciou mena prepojenia.



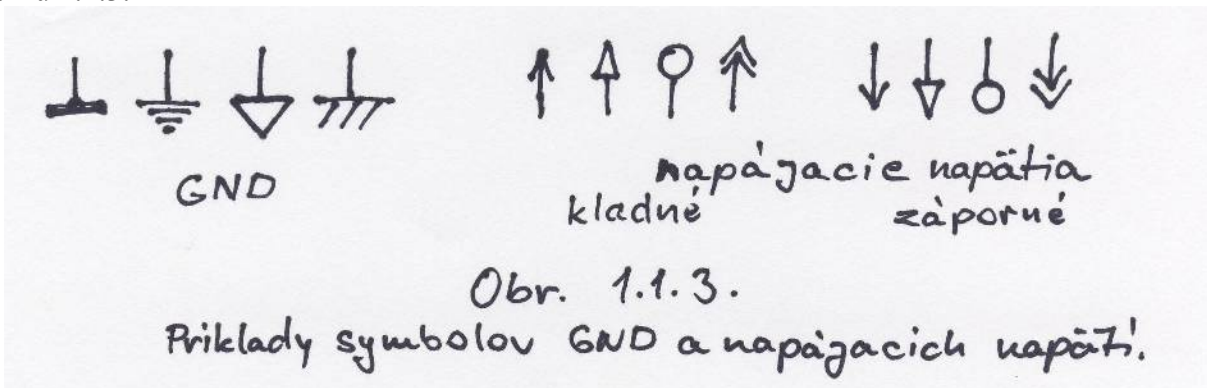
Obr. 1.1.2.1.
Príklad schémy zapojenia

Pod zbernicou rozumieme hrubú čiaru, ktorá symbolizuje väčší počet jednoduchých čiar, pričom zbernica je označená menom a hranicami indexov. Napríklad “ D(13 : 0)” označuje štrnásť signálov D(0), D(1),, D(13). Spojenie niektorého signálu, alebo skupiny signálov zbernice s iným prvkom schémy sa znázorňuje čiarou označenou príslušným symbolom, ktorá sa tesne pred spojením so zbernicou lomí v uhle 45 stupňov. Spojenie môže byť jednoduchý signál, alebo časť zbernice.

Ak k dvom neprepojeným čiaram, alebo zberniciam pripíšeme rovnaké meno, tieto prvky sa považujú za prepojené. Príklad jednoduchej schémy je na obrázku 1.1.2.1.

1.1.3. Symboly napájacích napätí

Spoločný potenciál napájacej sústavy je obvykle značený symbolmi ustálených tvarov podľa obrázku 1.1.3.



Podľa zvyklostí sa symbolom GND vpravo značí kostra mechaniky prístroja. Je veľmi neobvyklé kresliť napájacie symboly orientované inak, ako sú na obrázku. Teda, symboly GND majú byť orientované smerom dole, symboly kladných napájacích napätí smerom hore a symboly záporných napájacích napätí smerom dole, pričom rovnaké symboly označujú napätie rovnakej hodnoty. Orientáciu v schéme zlepši, ak sa k symbolu napájacieho napätia pripíše aj jeho hodnota. V každom prípade však treba v legende k schéme uviesť, ktorý symbol akej hodnote napájacieho napätia patrí.

1.1.4. Vzťah schémy a osadzovacieho výkresu

Symbole, ktoré súvisia so súčiastkou je treba označiť identifikátorom. Rovnaký identifikátor sa priradí aj súčiastke na osadzovacom výkrese. K bezproblémovej orientácii je nutné aj uvádzanie čísla vývodu púzdra integrovaného obvodu k vstupom a výstupom symbolu logického člena. Číslovanie vývodov je štandardné podľa použitého púzdra a je vždy uvedené v katalógovom liste súčiastky.

Historicky sa ustálilo použitie identifikátorov nasledovne:

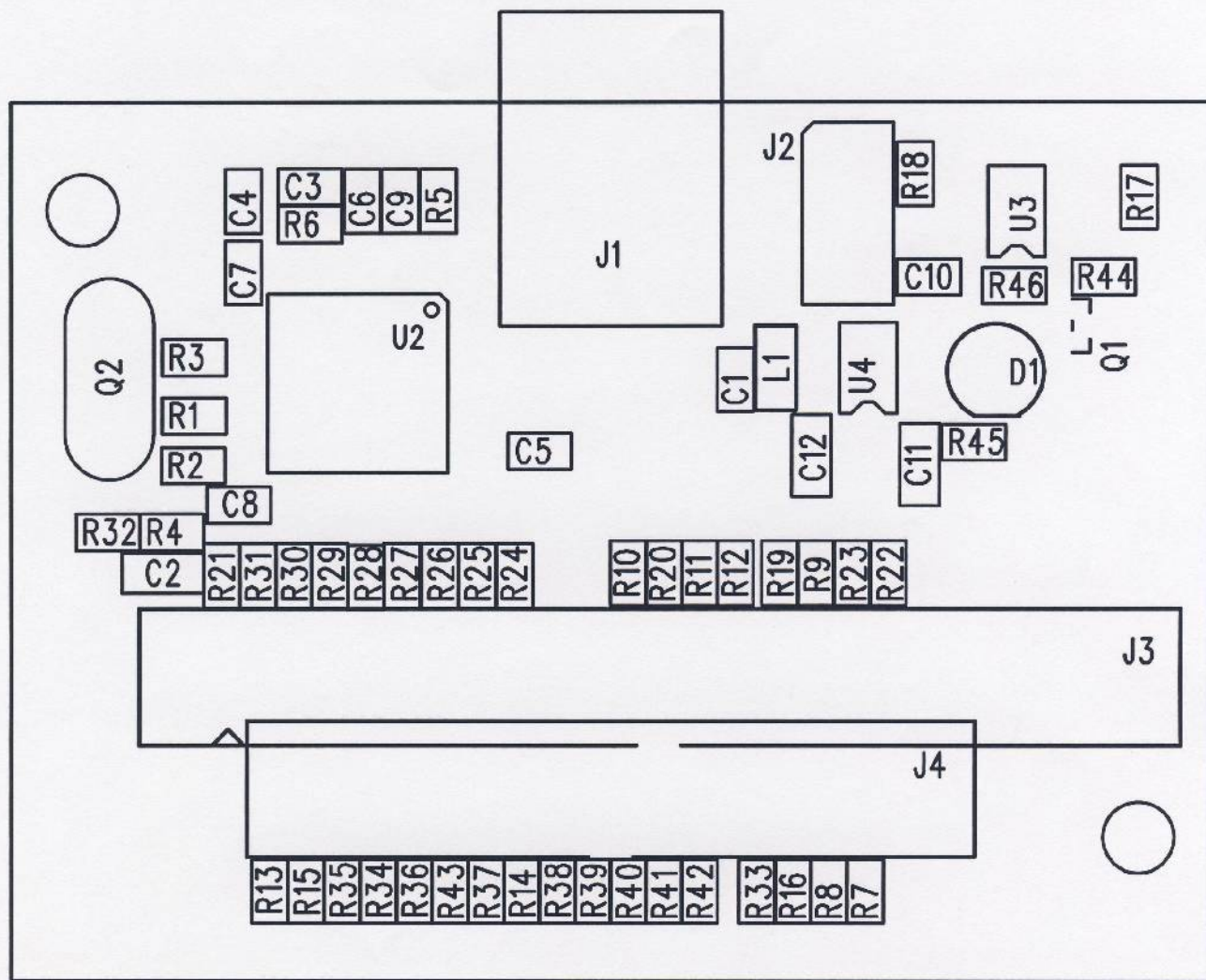
Integrované obvody	- U(číslo), napríklad U13
Aktívne polovodičové prvky (tranzistory, tyristory,...)	- Q(číslo), napríklad Q13
Diódy, zenerové diódy, varicapy,.....	- D(číslo), napríklad D13
Kondenzátory, kapacitné trimre, premenlivé kondenzátory	- C(číslo), napríklad C13
Odpory, reostaty, potenciometre, odporové trimre	- R(číslo), napríklad R13
Konektory	- J(číslo), napríklad J13

Uvedené identifikátory nie sú normované. Ich dodržanie uľahčuje orientáciu v dokumentácii, avšak použitie iných nenarúša na žiadne obmedzenie stanovené normou.

1.2. Osadzovací výkres

Osadzovací výkres je grafické zobrazenie rozloženia súčiastok na doske s plošnými spojmi, alebo na inom konštrukčnom prvku. Každá súčiastka je označená identifikátorom ktorý zodpovedá

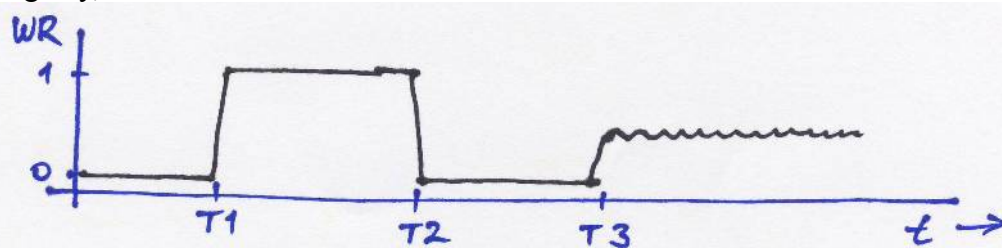
Obr. 1.2.1a



Obr. 1.2.1b

2. Popis udalostí v logických systémoch

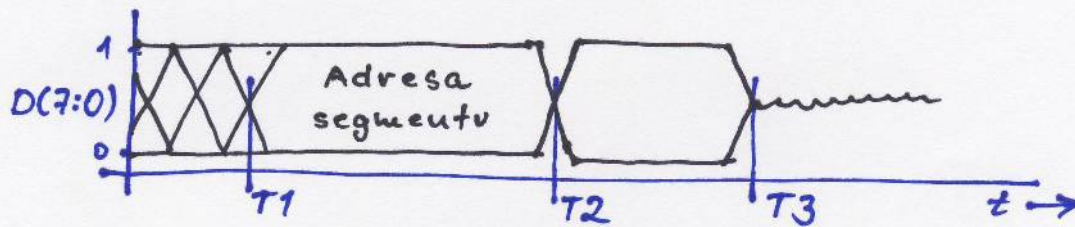
Udalosti sa v logických systémoch, kde čas je významným parametrom udalosti, popisujú tak, že sa popíše (najčastejšie zobrazí v grafickej forme) každý signál so zdôraznením časovej súradnice, prípadne jej kvantifikovaním kótami v časovej oblasti. Pri kreslení udalostí sa môžu zobrazovať jednoduché signály, alebo zbernice.



Obr. 2.1.

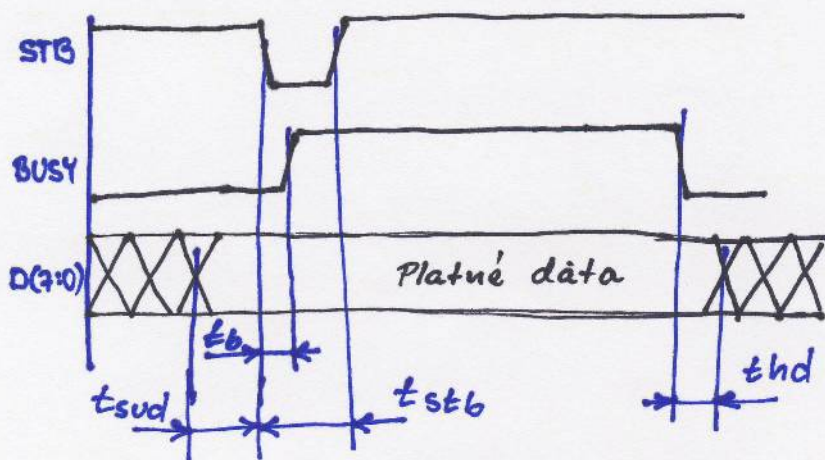
Časový priebeh jednoduchého
--jednoznačne definovaného signálu.

Na obrázku 2.1. je časový priebeh jednoduchého signálu s pevne definovaným správaním WR. Tento sa môže nachádzať v jednom z troch stavov. V log. 0, (pred T1 a medzi T2 a T3), log. 1 (medzi T1 a T2) a stave vysokej impedancie (po čase T3).



Obr. 2.2.
Časový priebeh stavov zbernice,
alebo jednoduchého nejednoznačne definovaného
signálu.

Na obrázku 2.2. je nakreslený časový priebeh stavov zbernice D(7:0). Zbernica sa môže nachádzať v jednom z troch stavov. V nedefinovanom stave (pred T1) sa nachádza vtedy, ak aspoň jeden zo signálov zbernice môže kedykoľvek zmeniť svoju hodnotu. V definovanom stave (medzi T1 a T2, resp. medzi T2 a T3) sa zbernica nachádza ak žiaden signál zbernice nesmie meniť svoju hodnotu. Ako atribút definovaného stavu môže byť aj špecifikovanie informácie, ktorú počas definovaného stavu zbernica reprezentuje (v našom prípade "Adresa segmentu"). Zbernica sa nachádza v stave vysokej impedancie (po T3) vtedy, ak sa všetky signály zbernice nachádzajú v stave vysokej impedancie. Rovnakým spôsobom zobrazujeme aj signál, ktorý je len jednobitový, ale jeho hodnoty nie sú presne definované (napríklad signál prenášajúci sériové dáta).



$$\begin{aligned} t_{sud} &\geq 50\text{ns} \\ t_b &\leq 30\text{ns} \\ t_{stab} &\geq 10\mu\text{s} \\ t_{hd} &\geq 0\text{ns} \end{aligned}$$

Obr. 2.3.
Časový priebeh udalosti

Na obrázku 2.3. je časový priebeh udalosti s kvantifikovaním časovej súslednosti kótami v časovej oblasti. Je ho možné interpretovať nasledovne:

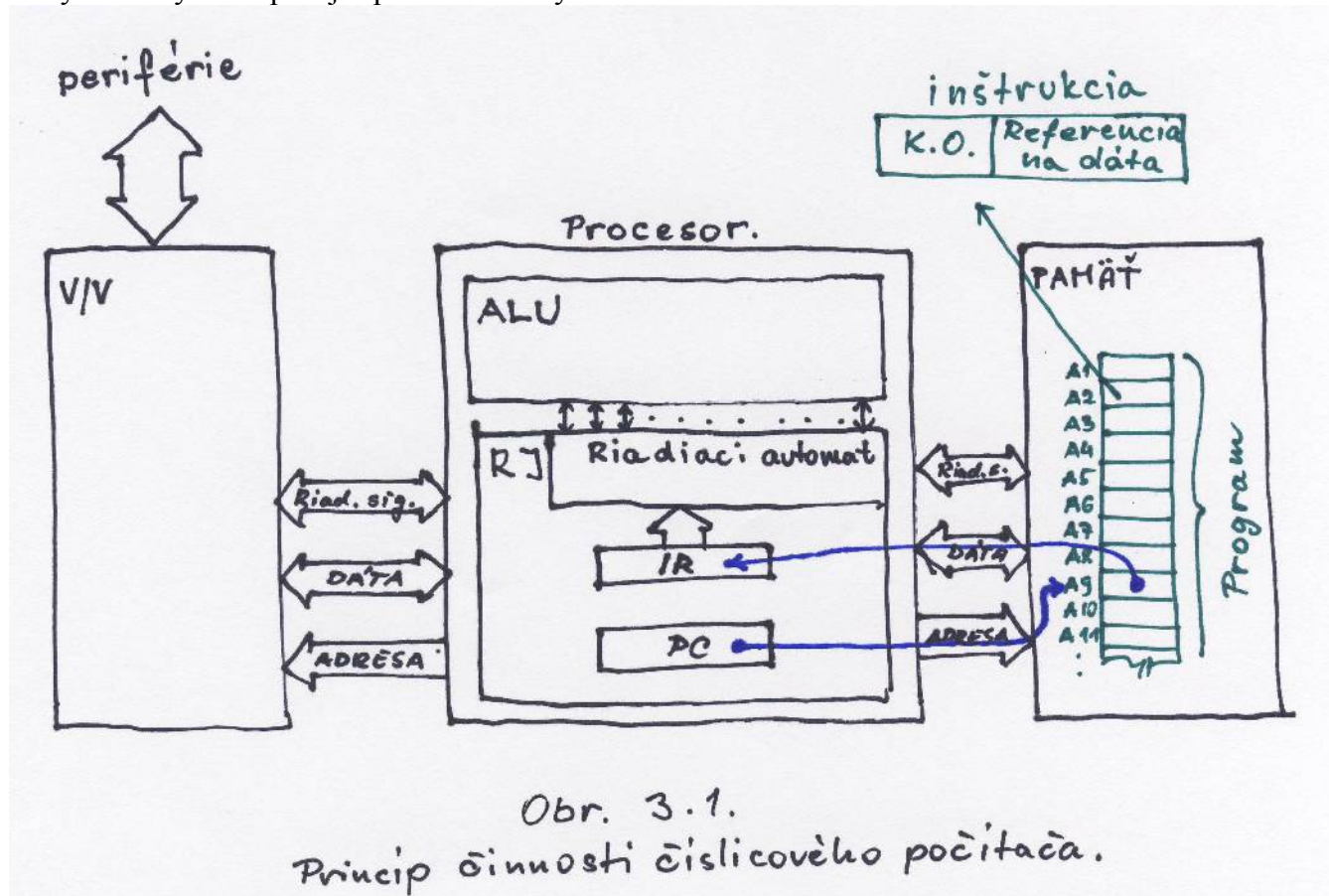
- Pred udalosťou je signál STB v stave log. 1, signál BUSY v stave log. 0 a zbernica D(7:0) v

nedefinovanom stave.

- Ako prvé sa nastaví stabilná hodnota zbernice D(7:0) tak, aby obsahovala platné dáta.
- Potom najskôr po uplynutí 50 ns sa nastaví signál STB na hodnotu log.0
- Potom sa signál BUSY musí nastaviť do stavu log.1 najneskôr do 30 ns.
- Signál STB musí trvať aspoň 10us.
- Zároveň so zmenou signálu BUSY na log.0 sa môže zmeniť aj stav zbernice.

3. Princíp činnosti číslicového počítača

Principiálne sporiadanie číslicového počítača je na obrázku 3.1. Skladá sa z troch blokov z ktorých každý zabezpečuje špecifické úlohy.



Obr. 3.1.
Princíp činnosti číslicového počítača.

Procesor riadi činnosť celého počítača prostredníctvom takzvanej riadiacej jednotky (RJ) a zabezpečuje transformáciu dát prostredníctvom takzvanej aritmetickej jednotky (ALU) ktorá obsahuje bloky určené na spracovanie dát, ako je sčítacia, násobička, logická jednotka a registre na krátkodobé uchovávanie dát.

Pamäť (hlavná pamäť, operačná pamäť) slúži na uloženie informácií, pričom pri manipulácii s informáciami v pamäti (čítanie, zápis) je potrebné pomocou adresy určiť miesto kde je informácia uložená (alebo kam ju chceme uložiť). Potom je možné dátovým kanálom prenášať príslušné dáta pod riadením procesora prostredníctvom riadiaceho kanálu.

V/V (vstup/výstup) slúži na komunikáciu počítača s perifériami, ktoré slúžia aj na komunikáciu počítača s obsluhou. Aktuálnu perifériu opäť určuje adresa. Cez prenosový kanál dát sa prenášajú informácie na a z periférie a riadiacim kanálom riadi procesor činnosť bloku.

V pamäti je uložený predpis činnosti celého počítača, ktorý nazývame "**program**". Program sa skladá z množiny inštrukcií, ktoré sú uložené na vzostupných adresách v tom poradí v akom sa majú

vykonávať. **Inštrukcia** obsahuje typicky dva druhy informácií. **Kód operácie** (K.O.), ktorý obsahuje každá inštrukcia, určuje procesoru, akú operáciu má vykonať. **Referencia na dáta** určuje, s akými dátami je treba príslušnú operáciu vykonať (datová časť). Pričom referenciou môže byť adresa (absolútna, alebo relatívna), prípadne priamo dáta. Pri niektorých inštrukciách táto informácia môže aj chýbať (napr. NOP – prázdna inštrukcia, DI – zakázanie prerušenia,).

Každá inštrukcia sa vykonáva v dvoch cykloch. Prvý cyklus je takzvaný výberový, kedy riadiaca jednotka vyberie z pamäte inštrukciu, ktorú je treba vykonať (aktuálna inštrukcia). V druhom cykle sa inštrukcia vykoná.

V riadiacej jednotke procesora sú dve dôležité registre. Jedným z nich je takzvaný **čítač inštrukcií** (PC – Programm Counter, IP – Instruction Pointer), ktorý v každom okamžiku obsahuje adresu aktuálnej inštrukcie. Druhým je **register inštrukcií** (IR), do ktorého sa počas výberového cyklu aktuálna inštrukcia uloží a zotrvá tam celý čas jej vykonávania.

Postupnosť vykonávania jednej inštrukcie nasledujúca:

Výberový cyklus – Obsah čítača inštrukcií vyšle riadiaci automat procesora na adresnú zbernicu pamäte. Vydá riadiaci signál čítania z pamäte a obsah dátovej zbernice uloží do registra inštrukcií. Tým výberový cyklus končí a v registre inštrukcií sa nachádza aktuálna inštrukcia, ktorú treba vykonať.

Výkonný cyklus – Riadiaci automat vyšle na základe obsahu registra inštrukcií postupnosť riadiacich signálov, čím zabezpečí vykonanie inštrukcie. Výkonný cyklus končí aktualizáciou obsahu čítača inštrukcií tak, aby v nasledujúcom cykle ukazoval opäť na aktuálnu inštrukciu. Vzhľadom na spôsob uloženia inštrukcií v pamäti je to najčastejšie pripočítanie čísla, ktoré zodpovedá dĺžke vykonanej inštrukcie (počet slov, ktoré daná inštrukcia v pamäti zaberá) k obsahu. Avšak v prípade, ak bola vykonaná inštrukcia vetvenia, upraví sa čítač inštrukcií už počas výkonného cyklu a po ňom sa už jeho aktualizácia nevykonáva. Ak je napríklad aktuálna inštrukcia inštrukciou absolútneho skoku, prepíše sa počas výkonného cyklu obsah čítača inštrukcií obsahom datovej časti inštrukcie, čím bude ukazovať na nasledujúcu aktuálnu inštrukciu.

Po pripojení napájacích napätí k číslicovému počítaču je potrebné zaistiť, aby sa program, ktorý je uložený v pamäti začal vykonávať od začiatku. Na tento účel slúži zvláštny signál (**RESET**), ktorý spôsobí, že čítač inštrukcií procesora sa nastaví na výrobcom stanovenú hodnotu (najčastejšie sa vynuluje). Prvá inštrukcia základného programového vybavenia počítača musí byť uložená práve na tejto adrese.

4. Hlavné vlastnosti pamäte

Podstatnými vlastnosťami pamäte číslicového počítača sú:

- **dĺžka slova** je udávaná v počte bitov a určuje paralelne prístupnú dĺžku dátovej informácie. V súčasných počítačoch je často možné voliť z niekoľkých alternatív, obvykle v násobkoch 8 bitov (bajtov).

- **kapacita** udáva počet binárnych slov, ktoré je možné do pamäte uložiť. Pričom za násobok “kilo” sa považuje $1024 (2^{10})$, čomu zodpovedá dĺžka adresnej informácie 10 bitov. Ak má pamäť premennú dĺžku slova, obvykle je kapacita udávaná v počte bajtov.

- **rýchlosť** pamäte je charakterizovaná hlavne dobou prístupu a dobou cyklu pamäte. Doba prístupu určuje, za aký čas je pamäť schopná vydať, resp. zapísať informáciu od nastavenia správnej adresy. Doba cyklu je najkratšia možná doba, ktorá uplynie medzi dvoma čítaniami, respektívne zápsismi.

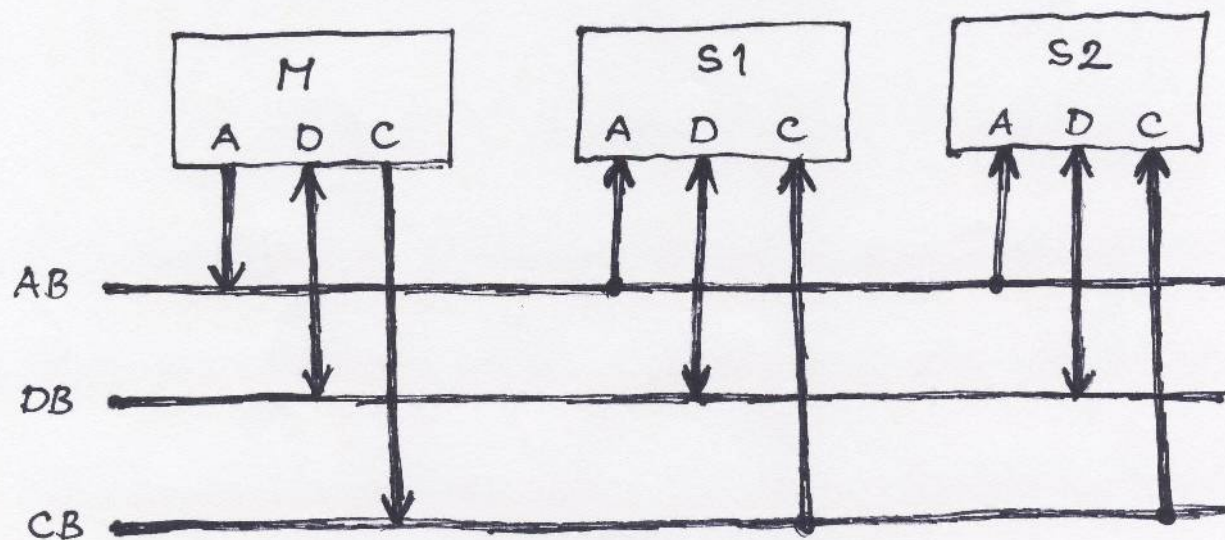
5. Zbernica, ako komunikačná cesta medzi stavebnými prvkami počítača

Na obrázku 5.1. je zobrazená zbernica (bus) číslicového počítača. Podľa druhu prenášaných informácií je ju možné rozdeliť na tri časti.

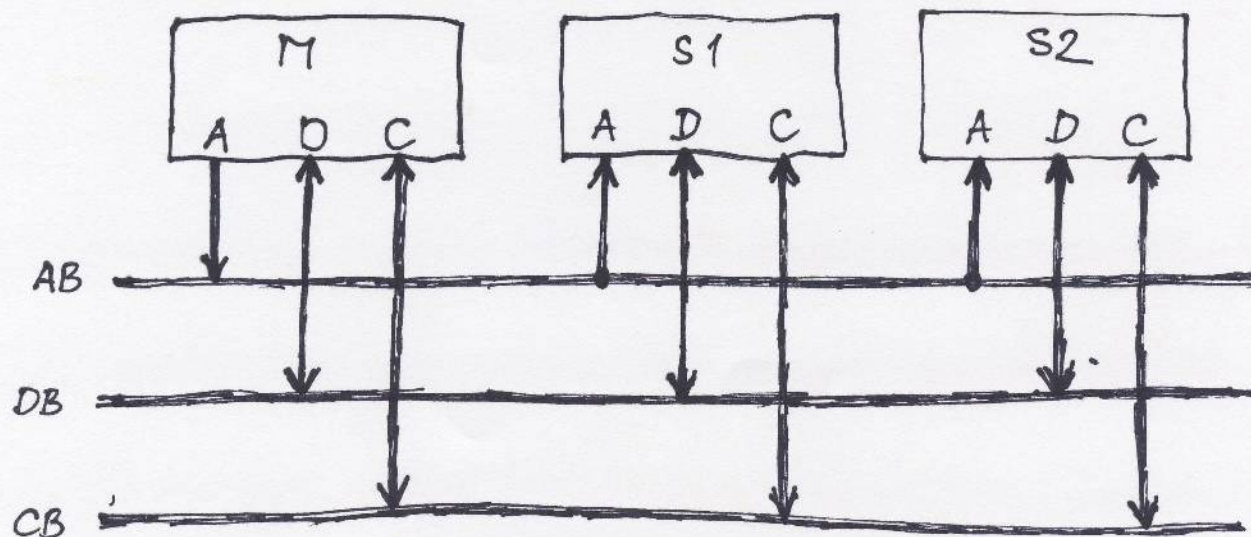
Adresná časť (AB) slúži na prenos adresy.

Dátová časť (DB) na prenos dát a inštrukcií.

Riadiaca časť (CB) na prenos riadiacich signálov.



Obr. 5.1.
Synchronná zbernica



Obr. 5.2.
Asynchronná zbernica.

Moduly číslicového počítača, ktoré sa na zbernicu pripájajú je možné rozdeliť na moduly **riadiace (Master)** a na moduly **riadené (Slave)**. V každom okamžiku môže byť k zbernici pripojený iba jediný modul typu Master a teoreticky ľubovoľný počet modulov typu Slave. Master je jediným modulom na zbernici, ktorý má právo nastaviť informáciu na adresnú zbernicu. Moduly Slave túto informáciu iba čítajú a na základe nej sa rozhodnú, či prenos, ktorého jedným účastníkom je vždy Master sa ich týka. Prenos informácií sa uskutočňuje cez dátovú časť zbernice. Pretože Master môže informácie do Slave-u zapisovať, alebo z neho čítať principiálne sa ako Master, tak i Slave viažu na

zbernicu obojsmerne. Master riadi prostredníctvom Riadiacej zbernice spolupracujúceho Slave-a a preto má právo vysielat' informáciu na riadiacu zbernicu. Na obrázku 5.1. je zobrazená takzvaná **synchronna** zbernica, kde Master predpokladá, že všetky Slave-i sú schopné reagovať na povel Master-a takou rýchlosťou, akou ich vysielajú.

V prípade ak je niektorý zo Slave-ov pomalší a nie je schopný reagovať tak rýchlo, ako to Master predpokladá, je treba vytvoriť signál v riadiacej časti zbernice, ktorým môže komunikujúci Slave požiadať Master-a o pozdržanie komunikácie na čas, ktorý na reakciu potrebuje. Takúto zbernicu nazývame **asynchronnou** (obr.5.2.).

5.1. Adresná časť zbernice

Adresná časť zbernice slúži na prenos adresnej informácie od Master-a k Slave-u. Pod pojmom **fyzická šírka zbernice** rozumieme počet vodičov, ktoré zbernicu tvoria. Niekedy sa cez zbernicu istej fyzickej šírky prenáša informácia inej šírky. Skutočný počet bitov, ktoré sa cez zbernicu prenášajú nazývame **logickou šírkou zbernice**. Logická šírka zbernice môže byť menšia (ak sa niektoré vodiče zbernice nevyužívajú), alebo väčšia (ak sa informácia o adrese prenáša v dvoch taktoch (tzv. Časový multiplex)) ako jej fyzická šírka. Pri tvorbe adresy na adresnej zbernici je možné využiť každú kombináciu hodnôt adresných bitov a preto je adresnou zbernicou logickej šírky a bitov možné preniesť 2^n rôznych adries.

5.2. Dátová časť zbernice

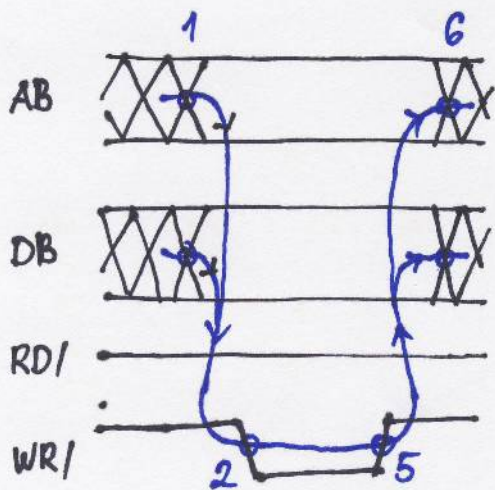
Slúži na prenos dát a inštrukcií medzi Master-om a Slave-om. Rovnako, ako v prípade adresnej časti zbernice je možné aj v prípade dátovej rozlišovať medzi fyzickou a logickou šírkou aj keď prípady rozdielnej fyzickej a logickej šírky bývajú menej časté, ako v prípade adresnej časti.

5.3. Riadiaca časť zbernice

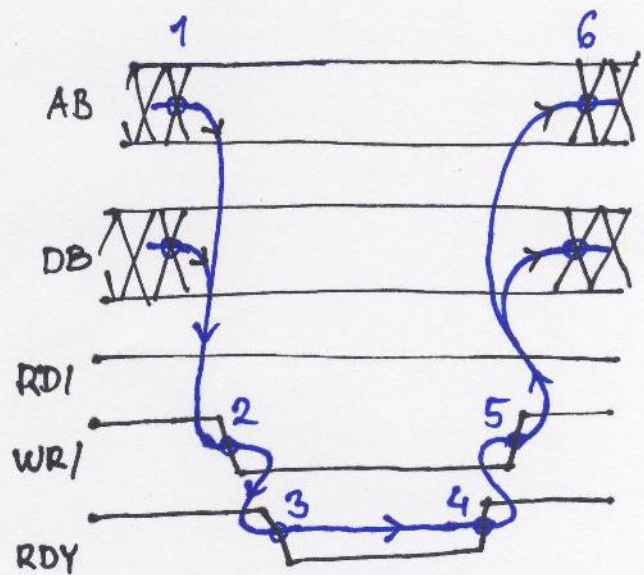
Cez riadiacu časť zbernice určuje Master druh komunikácie a stanovuje okamžik, kedy sa uskutočňuje. Kódovanie týchto informácií musí byť preto také, aby nebolo možné, aby vplyvom súbehových javov vznikali také kombinácie, ktoré znamenajú niektorý z povelov. Najjednoduchším riešením tohto problému je dodržanie zásady, aby na nastavenie aktívneho povelu na zbernici bolo potrebné zmeniť iba stav jediného vodiča (jediný riadiaci signál). Tejto zásade zodpovedá kódovanie riadiacich informácií systémom 1 z N, kedy je každému povelu pridelený jeden riadiaci signál. Pri tomto kódovaní pozostáva najjednoduchšia riadiaca časť synchronnej zbernice z dvoch signálov. Ich účel najlepšie odzrkadlí ak ich pomenujeme Read (RD) a Write (WR). Aktívnou úrovňou signálu RD dáva Master povel k čítaniu informácie (prenos Slave -> Master) a signálu WR k zápisu informácie (prenos Master -> Slave). Historicky sa vyvinulo, že aktívna hodnota povelov býva najčastejšie log. 0 a preto sú povelové signály označované často ako inverzné (RD/ a WR/). V prípade asynchronnej zbernice pribudne ešte signál, ktorým spolupracujúci Slave môže požiadať o pozdržanie komunikácie. Nazveme ho Ready (RDY). Nulová hodnota tohto signálu indikuje Master-u, že Slave ešte neukončil činnosť, ktorá sa od neho požaduje.

5.4. Prenos informácií v smere Master > Slave (Zápis)

Na obrázku 5.4.1. je zobrazený zápis dát. Postupnosť zápisu je nasledujúca. Začína tým, že Master vyšle adresu na AB a data na DB (1). Potom vydá povel WR/ = 0 (2), v prípade synchronnej zbernice po určitom čase opäť nastaví WR/ = 1 (5) a potom môže zmeniť informácie na AB a DB (6). V čase keď je WR/ = 0, musí Slave prevziať obsah dátovej časti zbernice. V prípade asynchronnej zbernice potom, čo Master vydal povel WR/ = 0 (2) môže požiadať spolupracujúci Slave o pozdržanie komunikácie nastavením RDY = 0 (3). Po tom, čo Slave prevzal obsah DB nastaví RDY = 1 (4) na čo Master odpovie zrušením povelu (WR/ = 1) (5) a potom môže zmeniť obsahy AB a DB (6).

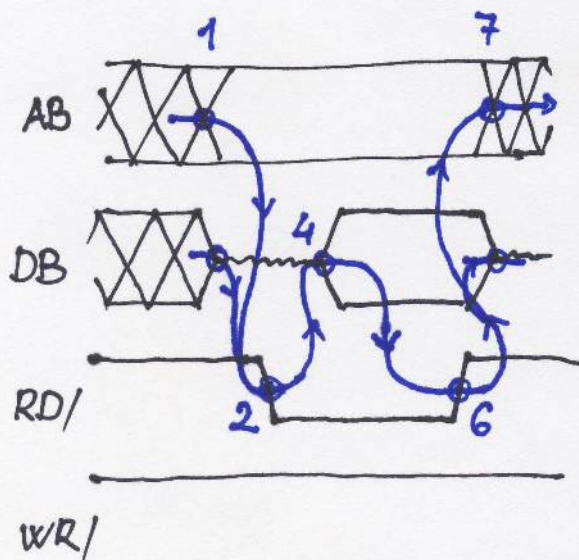


Synchronna
zbernica

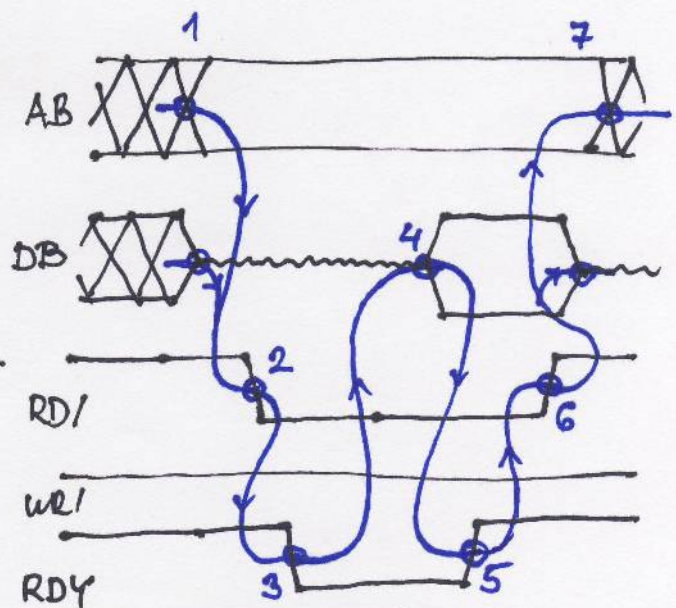


Asynchronna
zbernica

Obr. 5.4.1.
Prenos dat vsmeve $M \rightarrow S$ (zapis)



Synchronna
zbernica



Asynchronna
zbernica

Obr. 5.5.1.
Prenos dat v smere $S \rightarrow M$ (citatie)

5.5. Prenos informácií v smere Master < Slave (Čítanie)

Na obrázku 5.5.1. je zobrazené čítanie dát. Postupnosť čítania je nasledujúca. Začína tým, že Master vyšle adresu na AB a nastaví svoje výstupy na DB do stavu vysokej impedancie (1). Potom vydá povel k čítaniu ($RD/ = 0$) (2). V prípade synchronnej zbernice na to reaguje Slave tak, že vyšle na DB dáta. Master prečíta obsah DB a zruší povel na čítanie ($RD/ = 1$) (6) a potom môže zmeniť aj informáciu na AB. Slave musí po tom, čo detekoval $RD/ = 1$ nastaviť svoje výstupy na DB do stavu vysokej impedancie. V prípade asynchronnej zbernice potom, čo Master vydá povel k zápisu ($RD/ = 0$), môže Slave požiadať o pozdržanie komunikácie nastavením $RDY = 0$ (3). Po tom, čo nastaví dáta na DB nastaví $RDY = 1$ (5). Master potom prečíta obsah DB a zruší povel čítania ($RD/ = 1$) (6). Potom môže zmeniť stav AB (7) a Slave musí nastaviť svoje výstupy riadiace DB do stavu vysokej impedancie (7).

6. Adresné priestory

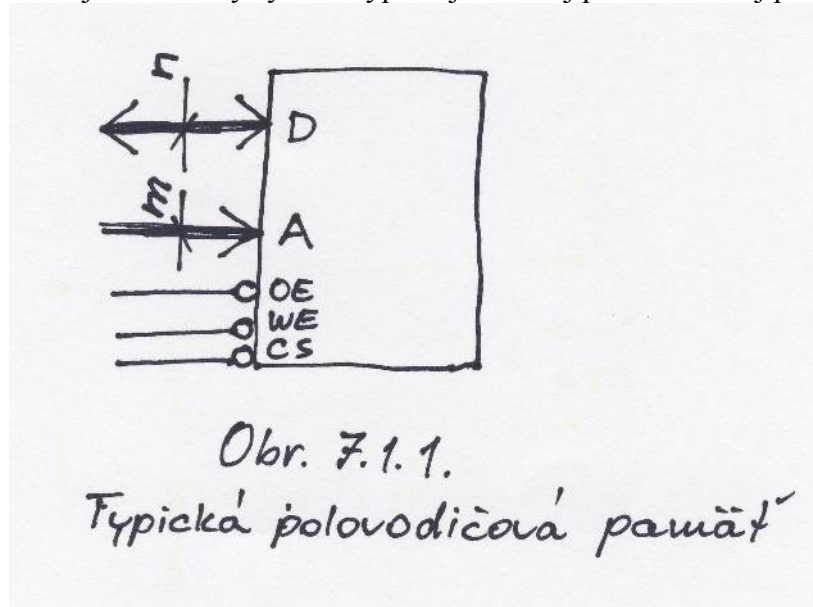
Pod pojmom "Adresný priestor" rozumieme **množinu adres a súbor riadiacich signálov** pomocou ktorých je možné so zariadeniami umiestnenými na daných adresách manipulovať. Aj keď adresný priestor môže byť atribútom zbernice, obvykle evokuje isté vlastnosti inštrukčného súboru procesora, ktorý je na zbernici umiestnený ako Master. Ak napríklad ponúka zbernica dvojicu riadiacich signálov MW/ a MR/ (Memory write a read) a tiež IOW/ a IOR/ (I/O write a read), predpokladá sa, že typický procesor, ktorý bude masťrom na zbernici bude mať inštrukcie manipulácie s pamäťou, ale aj inštrukcie manipulácie so vstupno/výstupnými zariadeniami. Taký adresný priestor, kde je možné zapisovať a aj čítať nazývame úplným adresným priestorom. Neúplný adresný priestor je taký, kde jedna z menovaných činností (najčastejšie zápis) nie je možná (napríklad adresný priestor pamäte programu v počítačoch s delenou pamäťou dát a programu).

Existencia adresného priestoru je teda definovaná jedným, alebo dvoma riadiacimi signálmi. Jeho veľkosť je určená logickou šírkou adresnej zbernice (a) pre tú skupinu inštrukcií, ktorá s danými riadiacimi signálmi pracuje a rovná sa 2^a .

7. Pripájanie polovodičových pamätí ku zbernici

7.1. Statický polovodičový pamäťový prvok

Na obrázku 7.1.1. je zobrazený symbol typickej statickej polovodičovej pamäte typu R/W.



K okoliu sa pripája n bitmi obojsmerných dátových signálov, m bitmi adresy a riadiacimi signálmi

OE/, WR/ a CS/. Počet dátových bitov (n) určuje dĺžku slova, ktoré je možné naraz (paralelne) do pamäte zapísať, respektíve z nej vyčítať. Počet adresných vstupov (m) určuje kapacitu pamäte, ktorá je rovná 2^m binárnych slov dĺžky n bitov. Riadenie pamäte typicky funguje podľa nasledujúcej tabuľky.

CS/	OE/	WE/	
1	x	x	D ≤ Z
0	1	1	D ≤ Z
0	0	1	D ≤ [A]
0	1	0	[A] ≤ D
0	0	0	[A] ≤ D

Signál CS/ (Chip Select non) je aktivačný signál celého prvku. Ak je jeho hodnota neaktívna (CS/ = 1), nereaguje pamäť na žiaden zo vstupov a svoje výstupy nastaví do vysokoimpedančného stavu. K tomu, aby sme pamäť donútili k normálnej činnosti, musí byť CS/ = 0. V takomto prípade aktívna úroveň riadiaceho signálu OE/ (Output Enable non) nastaví na dátové vývody pamäte obsah tej adresy, na ktorú ukazujú adresné bity. Aktívna úroveň signálu WE/ (Write Enable non) spôsobí, že sa informácia z dátových vývodov zapíše na adresu definovanú adresnými vstupmi. Treba poznamenať, že z hľadiska riadenia smeru toku dátovej informácie má signál WE/ pred signálom OE/ prednosť.

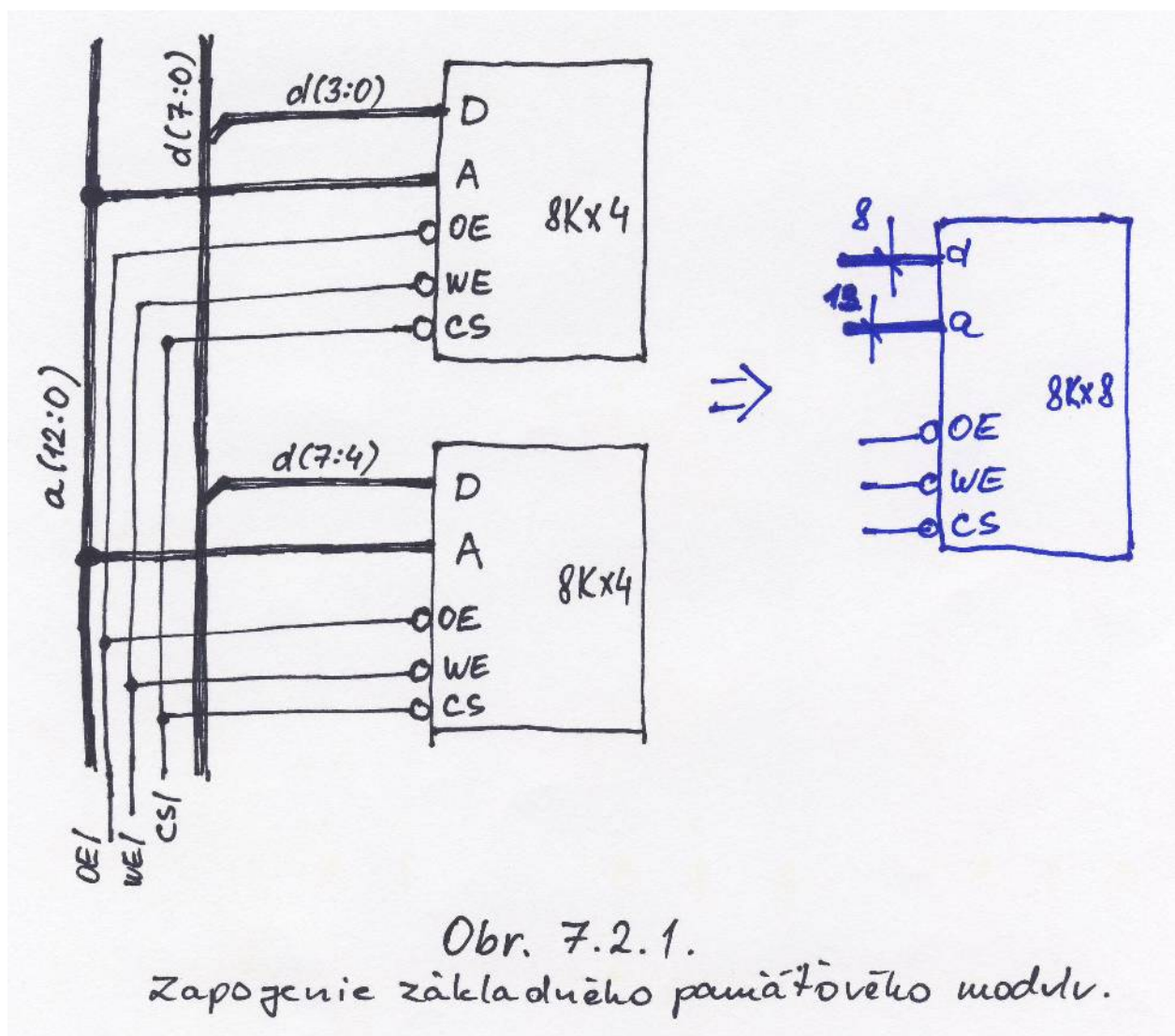
7.2. Tvorba pamäťového modulu s požadovanou šírkou slova (základný modul)

Ak vytvárame pamäť pripojiteľnú na zbernicu so šírkou dátovej časti d, musí byť aj šírka dátovej zbernice základného pamäťového modulu d. Musíme teda použiť taký počet pamäťových prvkov, aby súčet ich dátových signálov bol rovný (alebo väčší) šírke dátovej časti zbernice. Na obrázku 7.2.1. je zapojenie základného pamäťového modulu určeného na pripojenie k 8 bitovej dátovej časti zbernice zloženého zo štvorbitových pamäťových prvkov s kapacitou 8K x 4. Modul sa správa ako pamäťový prvok s kapacitou 8K x 8.

7.3. Umiestnenie základného pamäťového modulu do adresného priestoru.

Ak je kapacita základného pamäťového modulu menšia ako veľkosť adresného priestoru zbernice bude možné základný modul umiestniť na rôzne miesta adresného priestoru. Ak je kapacita základného modulu 2^m slov do adresného priestoru sa vždy vmestí práve $2^{(a-m)}$ základných modulov kde a je šírka adresnej časti zbernice a m šírka adresy základného modulu. Základné moduly sa najjednoduchšie umiestňujú do adresného priestoru tak, aby najnižšia adresa na ktorej sú prístupné mala m najnižších bitov nulových. Napríklad základný pamäťový modul z obrázku 7.2.1. je výhodné umiestniť do adresného priestoru 64 K počnúc adresami: 0x0000, 0x2000, 0x4000, 0x6000, 0x8000, 0xA000, 0xC000, alebo 0xE000. Predpokladajme, že chceme základný pamäťový modul umiestniť od adresy 0xA000 do pamäťového adresného priestoru zbernice, ktorému prislúchajú riadiace signály MW/ na zápis a MR/ na čítanie. Binárne tvary adres na ktorých má byť základný modul prístupný sú :

ab15	ab14	ab13	ab12	ab11	ab10	ab9	ab8	ab7	ab6	ab5	ab4	ab3	ab2	ab1	ab0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0



V prvom riadku je najnižšia, v druhom najvyššia adresa. Ako je z tabuľky zrejmé, 13 spodných adresných bitov adresuje vnútorný priestor základného modulu a tri najvyššie bity určujú jeho umiestnenie v adresnom priestore zbernice. Prostriedkom aktivácie, respektíve deaktivácie základného pamäťového modulu je signál CS/. Stačí zariadiť, aby v celej oblasti, kde má byť základný modul umiestnený bol CS/ = 0 a v ostatných miestach adresného priestoru CS/ = 1. Pretože z predchádzajúcej tabuľky vyplýva, že len v celej oblasti umiestnenia základného modulu sú ab15 = 1, ab14 = 0 a ab13 = 1, stačí vytvoriť kombinačný logický systém, ktorý za týchto podmienok zabezpečí správnu hodnotu signálu CS/. Môžeme ho popísať nasledujúcou mapou:

		<u>ab15</u>	
		<u>ab14</u>	
ab13	1	1	1
	1	1	0
		CS/	

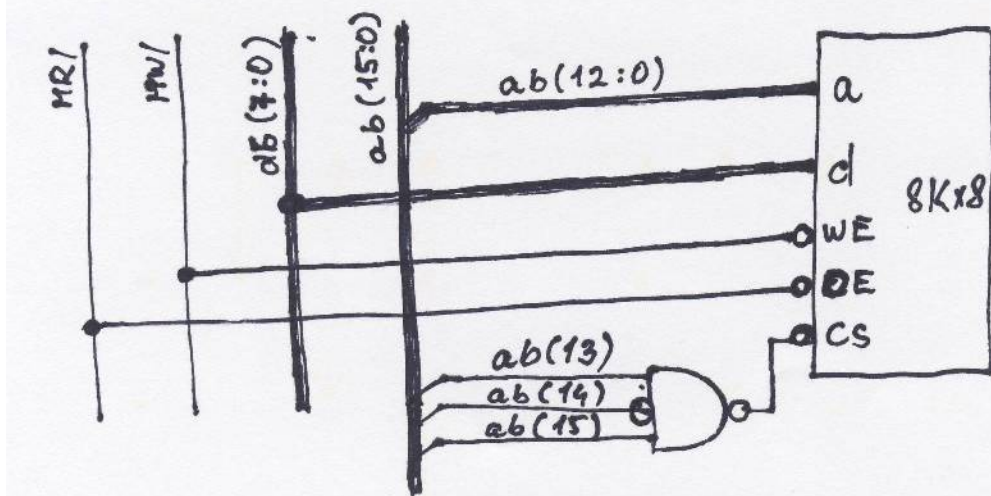
Kombinačnú logickú funkciu môžeme popísať nasledovne:

$$CS/ = ab13/ \vee ab14 \vee ab15/$$

Ak aplikujeme De Morganove pravidlo môžeme výraz upraviť na:

$$CS/ = (ab13 \cdot ab14/ \cdot ab15)/$$

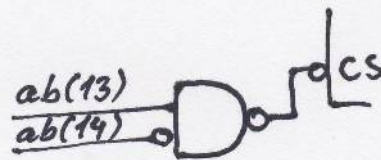
Na obrázku 7.3.1. je zapojenie základného pamäťového modulu umiestneného od adresy 0xA000.



Obr. 7.3.1.
Základný pamäťový modul 8Kx8
umiestnený od adresy 0xA000

7.3.1. Zrkadlenie v adresnom priestore

Základný pamäťový modul zapojený podľa obrázku 7.3.1. je umiestnený v adresnom priestore práve raz. Ak má kapacitu 8K, potom 56K adresného priestoru zostáva k ďalšiemu použitiu (je možné doňho umiestniť ďalšie zariadenia). Skúmajme, čo sa stane, ak adresný dekóder upravíme tak, že jeden z adresných signálov nepripojíme (obr. 7.3.1.1).



Obr. 7.3.1.1.
Upravený dekóder z obr. 7.3.1.

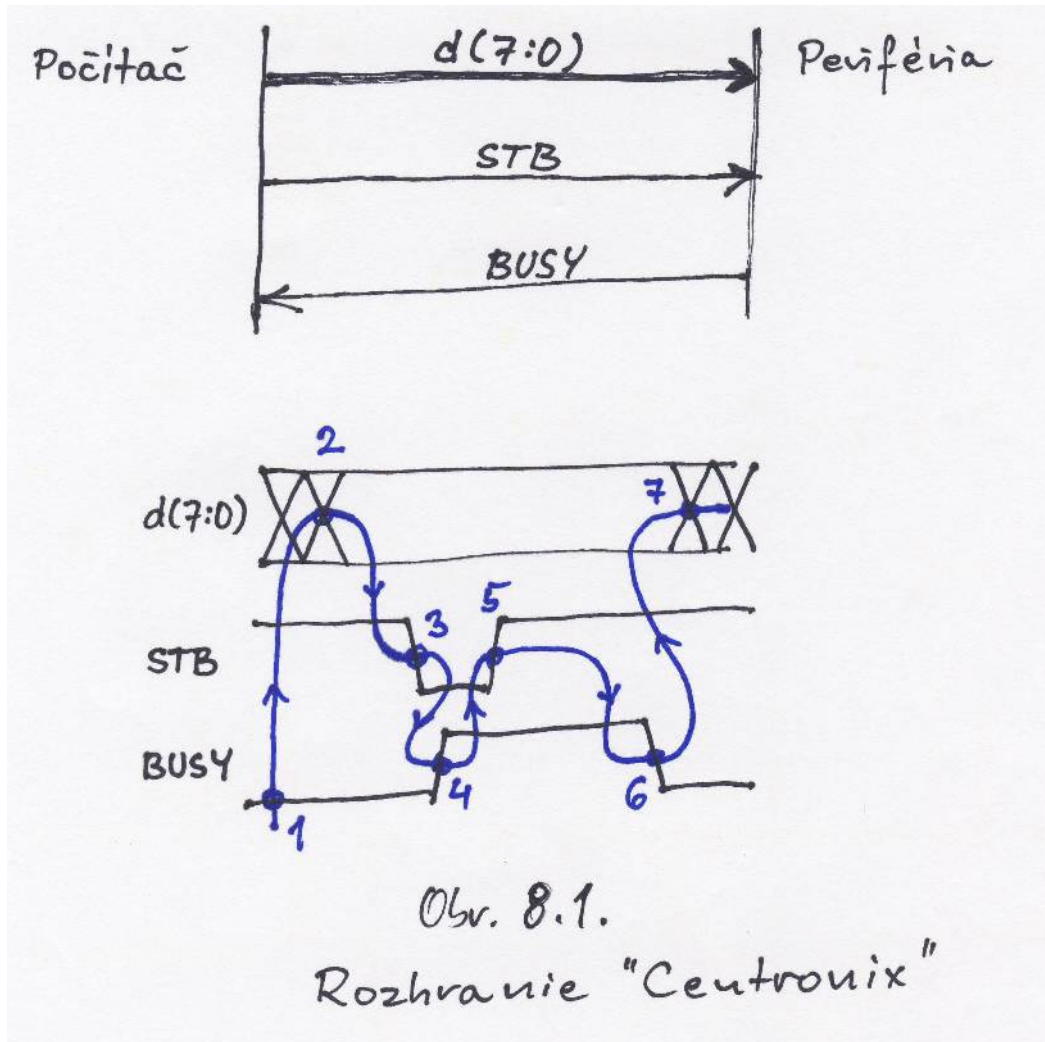
Signál CS/ teda nebude závisieť od signálu ab15. K tomu, aby CS/ bolo aktívne (= 0) postačí aby ab13 = 1 a ab14 = 0. Tomu však už vyhovuje nielen adresa 0xA000, ale aj adresa 0x2000, ako začiatky oblastí pamäťového priestoru, kde je základný pamäťový modul 8K x 8 umiestnený. Po tejto úprave bude úplne ľahostajné na ktorej z dvoch uvedených častí pamäťového priestoru bude programátor komunikovať s pamäťou, výsledok bude vždy rovnaký. Pamäť s takto upraveným dekóderom adresy bude zaberáť 16 K adresného priestoru. Ak by sme z adresného dekódera odpojili ďalší bit bude to už 32K a ak celý dekóder zrušíme a signál CS/ pripojíme na aktívnu úroveň (CS/ = 0),

bude celý adresný priestor zaplnený zrkadlovými obrazmi pamäte s kapacitou 8K x 8.

Všeobecne je možné konštatovať, že zariadenie sa v adresnom priestore objaví $2^{(a-p)}$ krát, kde a je šírka adresnej časti zbernice a p je počet bitov využitých pre adresovanie zariadenia.

8. Úloha procesora pri obsluhu V/V zariadení

Pri komunikácii so vstupno/výstupnými zariadeniami je potrebné okrem dát manipulovať aj s riadiacimi signálmi, ktoré príslušnú perifériu riadia. Ako príklad uvedieme paralelné rozhranie "Centronix", cez ktoré sú pripojené niektoré tlačiarne. Na obrázku 8.1. je nakreslené pripojenie periférie k rozhraniu "Centronix" a protokol jej ovládania.



Kreslený je len minimálny súbor riadiacich signálov, ktorý postačí k riadeniu periférie. Informácie sa z počítača do periférie prenášajú cez osem dátových signálov ($d(7:0)$). Povel k prevzatíu dát dáva počítač impulzom na výstupe STB . Periféria informuje o svojom stave signálom $BUSY$. Ak $BUSY = 0$, signalizuje periféria, že je pripravená prijať dáta. Obvyklý postup obsluhy je nasledovný: Počítač skontroluje signál $BUSY$. Ak je v stave 1, čaká, kým sa nenastaví do stavu 0 (1). Nastaví dáta na $d(7:0)$ (2), potom nastaví $STB = 0$ (3) a najskôr po 10 μ s $STB = 1$ (5). Ihneď po nastavení $STB = 0$ nastaví periféria signál $BUSY = 1$ (4). Tým sa starostlivosť počítača o perifériu končí aj keď celý cyklus končí až vtedy, keď periféria oznámi, že ukončila svoju činnosť nastavením $BUSY = 0$ (6). Dovtedy musí počítač držať informáciu na $d(7:0)$ nezmenenú (7).

V prípade, ak bude komunikáciu riadiť procesor počítača, môže to urobiť dvoma spôsobmi. Prvým spôsobom je **priame riadenie** komunikácie procesorom a druhý je **využitie prerušenia** procesora pri komunikácii. Tretím spôsobom komunikácie s perifériami je takzvaný **priamy prístup k pamäti (DMA)**, ktorý nevyužíva procesor k prenosu dát a používa sa hlavne na komunikáciu s rýchlymi perifériami.

8.1. Priame riadenie komunikácie procesorom

V prípade priameho riadenia komunikácie procesorom sa procesor venuje komunikácii nepretržite. V prvom rade je treba vytvoriť technické vybavenie, ktoré túto komunikáciu umožní. Vytvorme rozhranie “Centronix” podľa obrázku 8.1.

8.1.1. Technické vybavenie pre priame riadenie komunikácie s procesorom

Na obrázku 8.1.1.1. je zapojenie technického vybavenia rozhrania pre priame riadenie komunikácie procesorom. Je tvorené vyrovnávacou pamäťou dát (U1), vyrovnávacou pamäťou generujúcou riadiaci signál STB (U2) a trojstavovým oddelovačom signálu BUSY (U3). Vyrovnávacia pamäť dát je programátorovi prístupná prostredníctvom vstupno/výstupnej inštrukcie na adrese 0xFFFFE, vyrovnávacia pamäť signálu STB na adrese 0xFFFF a vstupný oddelovač signálu BUSY na adrese 0xFFFF. Uvedené informácie pre programátora je možné zhrnúť do nasledujúcej tabuľky:

Výstup

Adresa	7	6	5	4	3	2	1	0
0xFFFFE	d(7)	d(6)	d(5)	d(4)	d(3)	d(2)	d(1)	d(0)
0xFFFF	-	-	-	-	-	-	-	STB

Vstup

Adresa	7	6	5	4	3	2	1	0
0xFFFF	-	-	-	-	-	-	-	BUSY

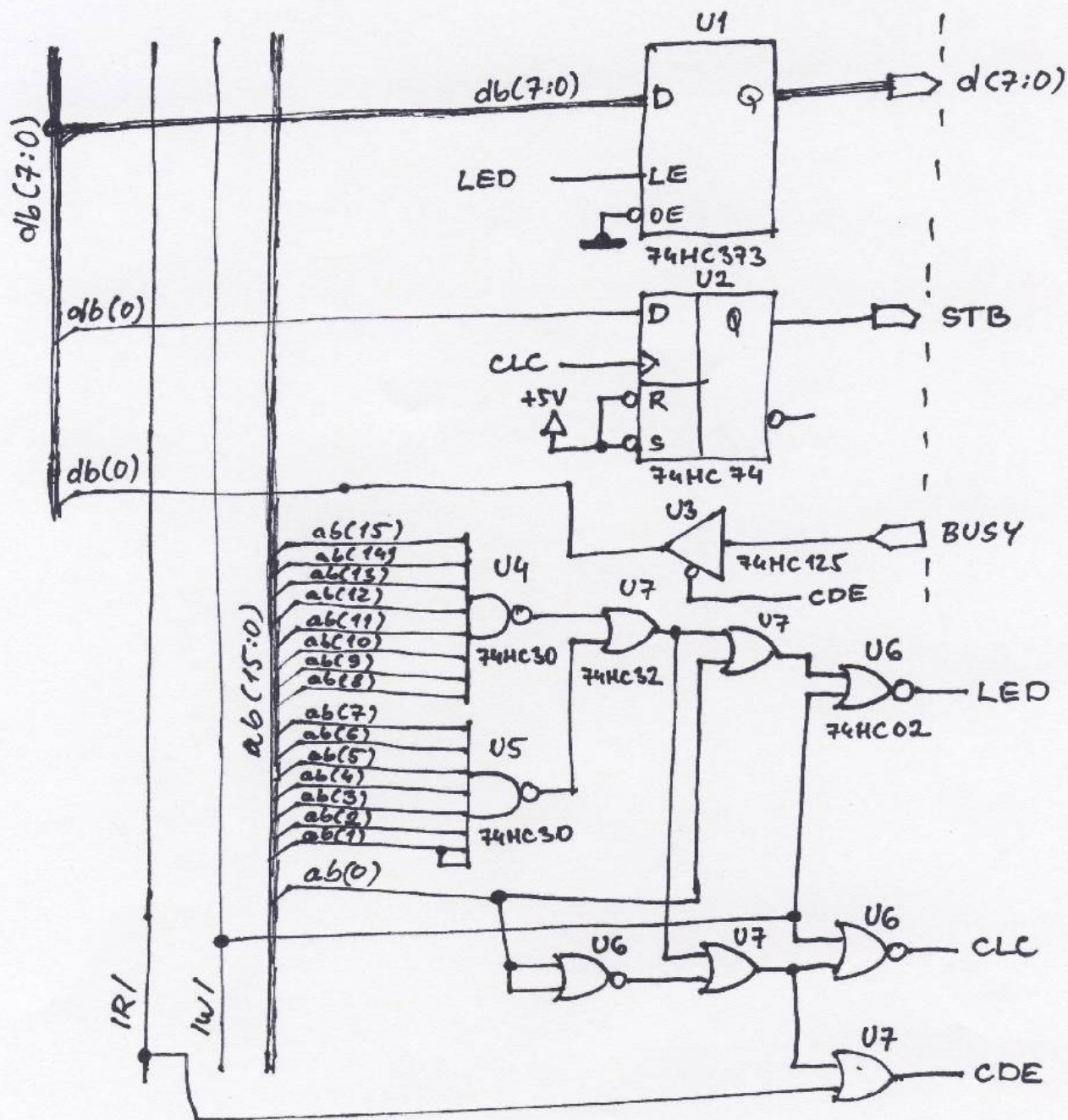
Je treba podotknúť, že pri čítaní signálu BUSY majú bity 1 až 7 náhodnú hodnotu, s čím je nutné počítať pri písaní obslužného programu.

8.1.2. Obslužný program

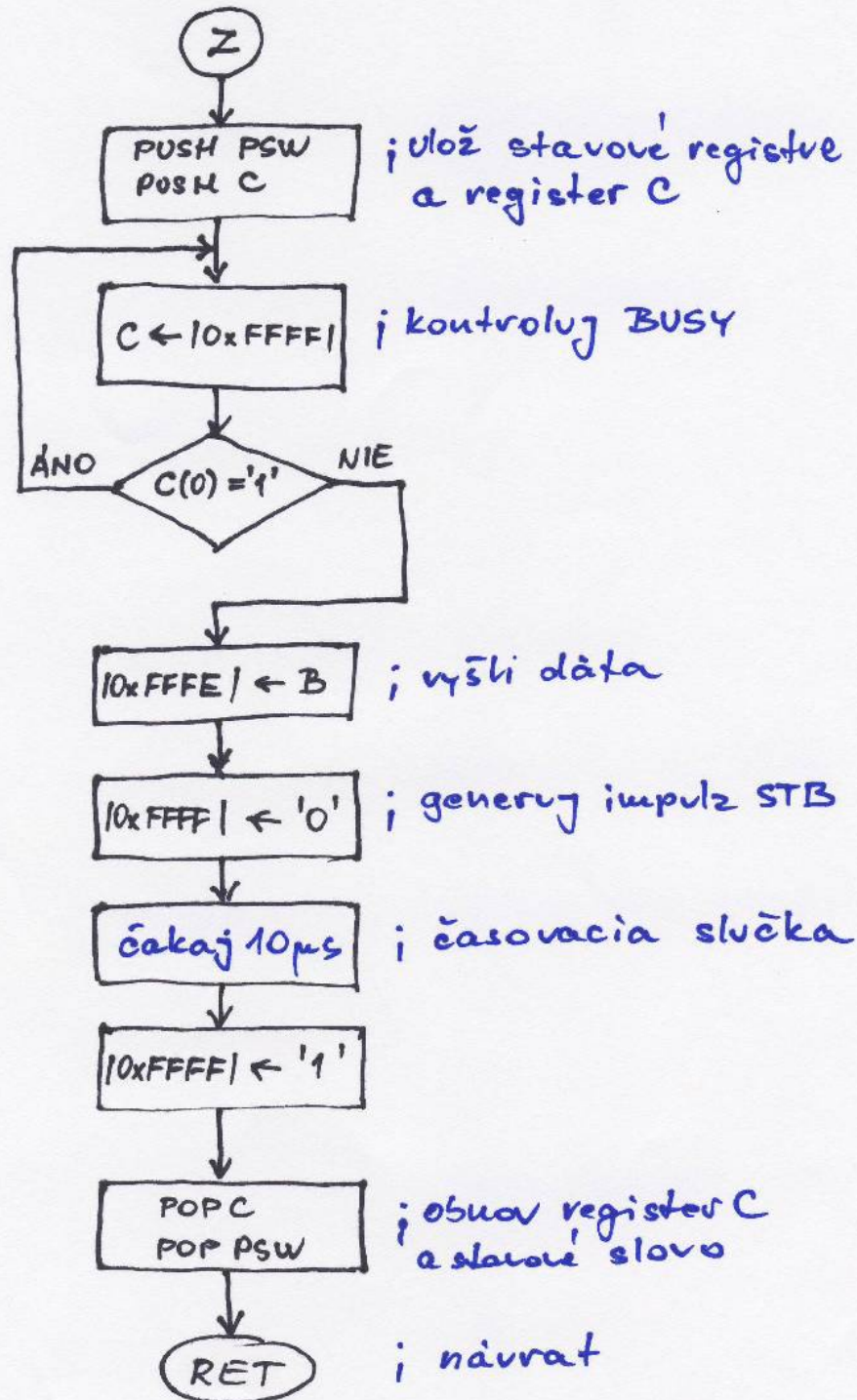
Podkladom k písaniu obslužného programu (obr. 8.1.2.1.) je časový diagram komunikácie z obr. 8.1. Predpokladajme, že pred zavolaním obslužného programu, uloží aplikácia (užívateľský program) dáta, ktoré chce vyslať do vnútorného registra B procesora. Ako prvá činnosť sa uložia obsahy tých častí procesora, ktoré sa v dôsledku činnosti obslužného programu môžu zmeniť. Je to stavový register a register C procesora. Potom je testovaný stav signálu BUSY, pokiaľ nenadobudne hodnotu 0. Ak je BUSY = 0, zapíše procesor data z registra B do vyrovnávacej pamäti rozhrania a potom vygeneruje impulz na výstupe STB, v minimálnej dĺžke 10 us. Blok, ktorý tento impulz generuje (čakaj 10 us), môže byť jednoduchá časovacia slučka, čo však obvykle môže spôsobiť problémy pri prenose programového vybavenia do iného technického prostredia, preto je výhodnejšie opierať časovanie o štandardné technické vybavenie (časovač). Po vygenerovaní impulzu na výstupe STB sa obnovia pôvodné stavy registra C a stavového registra a program inštrukciou návratu z podprogramu (RET) vráti procesor aplikácii, ktorá ho volala.

Ak predpokladáme, že spolupracujúcou perifériou je tlačiareň bez vyrovnávacej pamäte, alebo je už jej vyrovnávacia pamäť zaplnená uplynie medzi dvomi zápismi dát priemerne čas, ktorý je potrebný na vytlačenie jedného znaku, čo je asi 5 ms, by procesor s rýchlosťou 100 MIPS (čo nie je

žiadna závatne vysoká rýchlosť) urobil asi 500 000 inštrukcií použiteľných na riešenie nejakého problému. Táto skutočnosť je aj najväčšou nevýhodou tohto prístupu obsluhy periférie, ktorá je vzhľadom na rýchlosť procesora pomalá. Za výhody tohto prístupu je možné považovať jednoduchosť technického a dobrá laditeľnosť obslužného programového vybavenia, pretože pracuje synchronne s aplikáciou.



Obr. 8.1.1.1.
Rozbranie pre priame riadenie
procesorom



Obr. 8.1.2.1.

Obsluha rozhrania z obr. 8.1.1.1. pod
 priamym riadením procesora

8.2. Využitie prerušenia procesora pri komunikácii s perifériou

Iným prístupom riadenia komunikácie s perifériou je využitie možnosti prerušenia procesora tak, aby sa periférii venoval len v tých okamžikoch, keď to periféria potrebuje. Takto je možné výrazne šetriť čas procesora a navrhnuť počítač s vyšším výkonom. Výkon takéhoto počítača je však silne ovplyvnený vlastnosťami operačného systému a organizáciou vykonávania užívateľských programov.

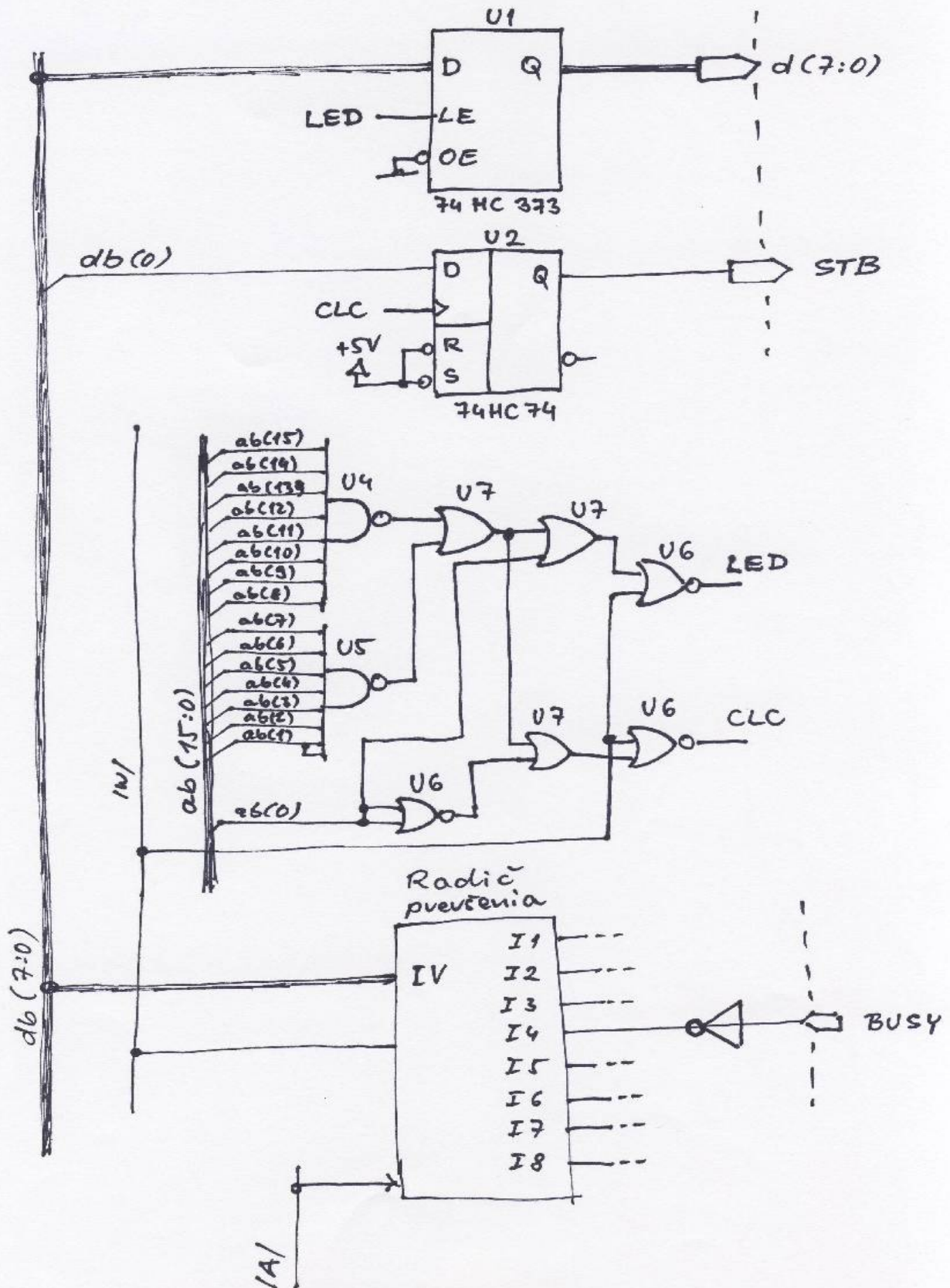
8.2.1. Prerušenie

Pod pojmom prerušenie rozumieme mechanizmus podľa ktorého preruší procesor vykonávanie jedného programu a začne vykonávanie iného programu, ktorý s daným prerušením súvisí. Tento program budeme nazývať “**Obslužný program prerušenia**”, alebo zjednodušene “**Obsluha prerušenia**”. Procesor je obvykle vybavený jedným vstupom často označeným “INT”, ktorým môže okolie požiadať o prerušenie. Ak sa signál INT nastaví na aktívnu úroveň, procesor reaguje na túto skutočnosť nasledovne: V prípade, ak má zakázané prerušenie (zákaz prerušenia je možné ovládať dvojicou inštrukcií Enable interrupt a Disable interrupt), žiadosť o prerušenie ignoruje. Ak je prerušenie povolené, dokončí procesor vykonávanie aktuálnej inštrukcie, uloží do zásobníka obsah registra PC procesora (adresu inštrukcie, ktorú by mal vykonať, keby nebol prerušený), nastaví zákaz prerušenia (aby nebol nekontrolovalelne prerušovaný) a prostredníctvom špeciálneho signálu (najčastejšie IA/ (Interrupt Acknowledge non)) požiada, aby mu zariadenie, riadiace prerušenia počítača poslalo informáciu (takzvaný **vektor prerušenia**) o adrese, kde sa nachádza obsluha prerušenia. Tento vektor obvykle nie je priamo adresa, ale procesor z neho adresu známym postupom vypočíta. Na túto adresu potom odskočí. Zariadenie, ktoré vektor prerušenia na dátovú zbernicu vysielá nazývame obvykle **radičom prerušenia**. Proces, ktorým sa procesoru oznamuje adresa obsluhy prerušenia sa nazýva **identifikáciou zdroja prerušenia**. Je treba si uvedomiť, že zdrojom prerušenia môže byť (a aj býva) viac ako jeden prerušovací signál. Radič prerušenia generuje žiadosť o prerušenie (INT) vždy, keď aspoň jeden z týchto prerušovacích signálov je aktívny. V prípade, ak je naraz viac ako jeden aktívny, je treba rozhodnúť o ich dôležitosti (**priorite**) a odskočiť na obsluhu toho najdôležitejšieho (toho s najvyššou prioritou). Je veľmi užitočné, ak radič prerušenia umožní selektívne blokovanie jednotlivých prerušovacích signálov. Preto býva vybavený takzvaným maskovacím registrom, ktorý dovoľuje zablokovat' účinok ktorejkoľvek žiadosti o prerušenie blokovacia informáciu budeme nazývať **maskou** prerušenia. V závislosti na správaní sa radiča prerušenia rozoznávame dve rôzne stratégie identifikácie zdroja prerušenia.

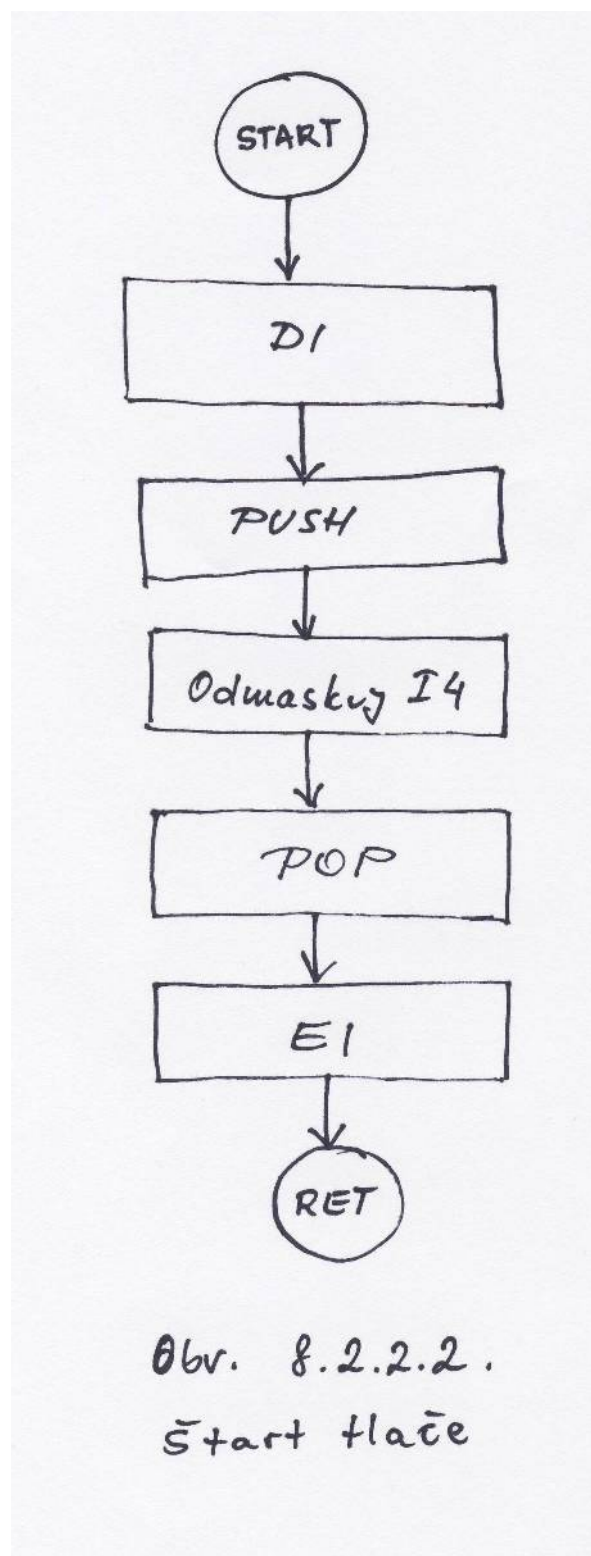
Prvá je identifikácia zdroja prerušenia **technickými prostriedkami** (takzvané vektorované prerušenie). V tomto prípade radič prerušenia vyšle ako odozvu na signál IA/ vektor prerušenia, ktorý ukazuje na adresu obsluhy aktívneho prerušenia s najvyššou prioritou. Z toho vyplýva, že o prioritě prerušovacích signálov musí rozhodnúť už radič prerušenia technickými prostriedkami.

Druhá stratégia je identifikácia zdroja programovými prostriedkami. V takomto prípade vsúva radič prerušenia vektor, ktorý nie je od aktívnych prerušovacích signálov a ich priority závislý. Vždy je to vektor ukazujúci na **identifikačný program zdroja prerušenia**. V rámci tohto programu potom procesor prečíta stav všetkých vstupných prerušovacích signálov a môže podľa okamžite platných pravidiel rozhodnúť o ich prioritě a odskočiť na obsluhu toho s najvyššou.

Ak máme zhodnotiť výhody a nevýhody týchto dvoch postupov, tak identifikácia zdroja prerušenia technickými prostriedkami vyžaduje zložitejšie technické vybavenie na strane radiča prerušenia, obvykle nedovoľuje ľubovoľné nastavenie priority prerušovacích signálov, avšak reakcia na žiadosť o prerušenie (tzv. **latencia prerušenia**) je kratšia ako v prípade identifikácie programovými prostriedkami. Dlhšia latencia prerušenia je jedinou nevýhodou identifikácie zdroja prerušenia programovými prostriedkami. Tento spôsob identifikácie vedie na jednoduchšie technické vybavenie radiča prerušenia a možnosť dynamickej zmeny priority prerušovacích signálov.

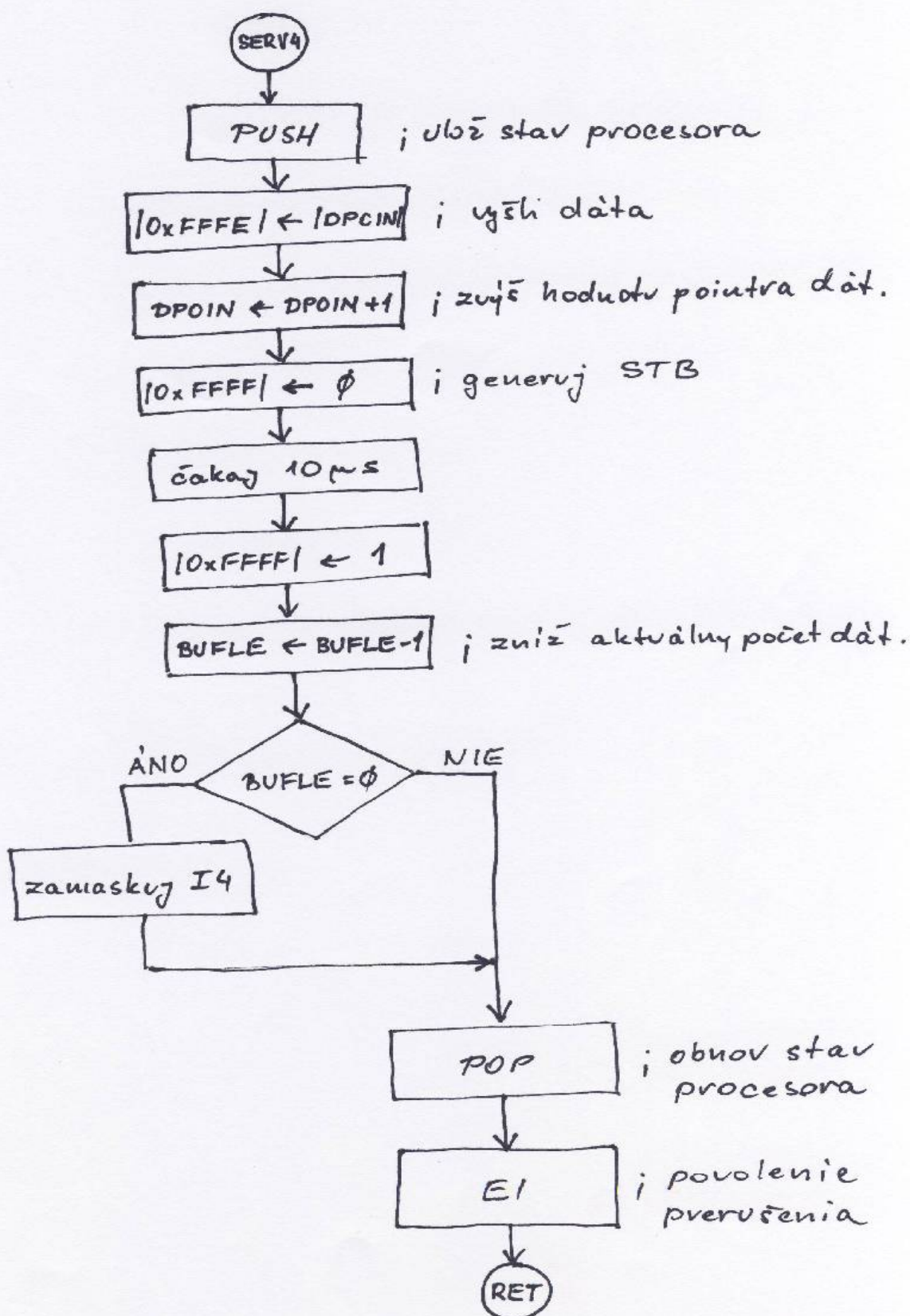


Obr. 8.2.2.1.
Rozhranie využívajúce prevetenie



8.2.2. Obsluha prerušenia

Na obrázku 8.2.2.1. je pripojenie tlačiarne s rozhraním "Centronix" k vstupu/výstupnému systému počítača tak, aby bola možná obsluha pomocou prerušenia. Technické vybavenie na vytváranie dát a signálu STB sa nelíši od toho z obrázku 8.1.1.1. Rozdiel je v pripojení signálu BUSY, ktorý je pripojený k vstupu I4 radiča prerušenia. Predpokladajme, že aktívna úroveň prerušovacích vstupov je log.1 a existuje možnosť maskou blokovat', alebo povoliť ktorýkoľvek prerušovací vstup.



Obr. 8.2.2.3
Obsluha tlačie.

Aby bol styk s perifériou pomocou prerušenia efektívny, treba zmeniť filozofiu odovzdávania dát obslužnému programu periférie. Predpokladajme, že aplikácia, ktorá chce s tlačiarňou spolupracovať zapíše na systémové adresy nasledujúce informácie:

- DPOIN - smerník na začiatok vyrovnávajúcej pamäti dát
- BUFLE - dĺžka vyrovnávajúcej pamäti dát.

Pre správnu činnosť prerušovacieho systému je treba vytvoriť dva obslužné programy. Prvý, slúži iba na štart tlačie. Nie je to obsluha prerušenia, ale podprogram, ktorý zavolá aplikácia, alebo operačný systém po tom, čo vytvoril vyrovnávaciu pamäť dát a zapísal príslušné systémové informácie DPOIN a BUFLE. Tento podprogram označme návestím START (obr. 8.2.2.2.).

Jeho činnosť spočíva hlavne v odmaskovaní úrovne I4 prerušenia, kam je pripojená inverzia signálu BUSY. Predtým však odloží obsah tých častí, ktoré môže program štart modifikovať do zásobníka (PUSH), zakáže prerušenie v procesore (DI), po odmaskovaní prerušovacieho vstupu I4 obnoví pôvodný stav procesora a pred návratom z podprogramu povolí prerušenie. Zakázanie a povolenie prerušenia plní účel zábrany pred prerušením vo vnútri štarovacieho podprogramu a teda aj účel udržania kontroly nad dĺžkou zásobníka.

Obslužný podprogram prerušovacieho signálu I4 (SERV4) zabezpečuje obsluhu prerušenia, ktoré vzniká pri činnosti periférie vždy vtedy keď $BUSY = 0$. Vývojový diagram tohto podprogramu je na obrázku 8.2.2.3. V prvom rade sa uložia obsahy tých častí procesora, ktoré sa v priebehu obsluhy môžu zmeniť. Potom sa z aktuálneho miesta vyrovnávacej pamäte zapíše bajt dát do vyrovnávacej pamäte rozhrania. Následne sa zvýši hodnota pointeru vyrovnávacej pamäte tak, aby ukazovala na nasledujúci aktuálny bajt. V ďalšom sa generuje impulz na výstupe STB. Dalším krokom je aktualizácia zostávajúcej dĺžky vyrovnávacej pamäte znížením BUFLE o 1. Ak bude následne $BUFLE = 0$, znamená to, že bol vyslaný posledný bajt dát a obsluha zamaskuje prerušovací vstup I4, čím zabráni ďalšiemu volaniu obsluhy. Obsluha končí obvyklou obnovou stavu procesora a povolením prerušenia pred návratom (RET).

8.2.3. Typická štruktúra programového vybavenia počítačov využívajúcich prerušenie

Technika obsluhy periférií prostredníctvom prerušenia procesora môže významne zlepšiť využitie času procesora. K tomu, aby sa želaný efekt dostavil, je však potrebné splniť aj iné podmienky. Predovšetkým si treba uvedomiť, že aplikácia (program), ktorá potrebuje vyslať, alebo prijať dáta z periférie nemôže ďalej pokračovať pokiaľ sa prenos neuskutoční. Ak by to bola jediná užitočná činnosť, ktorú počítač vykonáva, musel by procesor čakať na ukončenie komunikácie s perifériou. To by však vzhľadom na využitie času procesora malo zhruba rovnaký efekt, ako keby bola periféria obsluhovaná pod priamym riadením procesora. Z tohto dôvodu je dôležité, aby operačný systém počítača bol takzvaný **viacúlohový** (multitasking). V takomto systéme je súčasne niekoľko aplikácií. Každá má k dispozícii istý procesorový čas, ktorý jej je cyklicky pridelovaný. Zatiaľ, čo jedna z aplikácií využíva čas procesora (procesor ju vykonáva, **je aktívna**), ostatné sú buď čakajúce na pridelenie času procesora (**v poradi**), alebo sú dočasne **mimo poradia** z dôvodu čakania na ukončenie udalosti, ktorá znemožňuje ich spustenie. Takou udalosťou je typicky V/V komunikácia.

Operačný systém je ten prvok, ktorý riadi aktiváciu aplikácií a unifikuje prístup k perifériám z dvoch dôvodov. Prvým dôvodom je zjednodušenie písania aplikácií, pretože programátor aplikácie nemusí byť oboznámený so spôsobom komunikácie s perifériou, ale stačí, ak využije jednoduché funkcie, ktoré operačný systém ponúka. Druhý a zásadný dôvod je ten, že komunikácia aplikácie s perifériou zásadne ovplyvní jej "spustiteľnosť" a preto musí mať operačný systém informáciu o každej takejto činnosti.

Činnosť bežiackej aplikácie sa zastaví ak:

- uplynul jej čas
- obrátila sa na perifériu
- vykonala nedovolenú operáciu.

V prvom z týchto prípadov procesor nezmení stav aplikácie (zostáva v poradí) a pri ďalšej príležitosti (keď znova na ňu príde rad) ju opäť spustí.

V druhom prípade ju dočasne presunie mimo poradia a do poradia ju zaradí až po ukončení príslušnej V/V komunikácie.

V poslednom prípade ju vyradí zo zoznamu a už ju neaktivuje. O tejto skutočnosti oboznámi obsluhu obvykle chybovým hlásením.

8.3. Priamy prístup k pamäti (DMA (Direct Memmory Access))

V tých prípadoch komunikácie s perifériami s ktorými sme sa zaoberali každá slabika dát musela prejsť cez procesor. Bez ohľadu na to, či bola periféria obsluhovaná pod priamym riadením, alebo prostredníctvom prerušenia procesora musel procesor informáciu prečítať z pamäti a potom vyslať, respektívne prijať a potom uložiť do pamäte. Táto skutočnosť prenos dát spomaľuje a je pri prenose veľkého objemu dát obmedzujúca.

Pre prenos veľkého množstva dát je vhodný tretí spôsob V/V komunikácie, takzvaný priamy prístup k pamäti (**DMA**). Keď sa komunikácia neuskutočňuje, je modul DMA pripojený na zbernicu s právami riadeného modulu (Slave). Ak sa vyskytne potreba komunikácie so zariadením, ktoré je obsluhované modulom DMA, zapíše procesor do DMA modulu minimálne nasledujúce informácie:

- Informácie o periférii s ktorou treba komunikovať (napríklad číslo stopy a sektora disku, prípadne počet sektorov s ktorými treba manipulovať a spôsob manipulácie (čítanie, zápis,...))
- adresu začiatku dát v pamäti
- počet položiek.

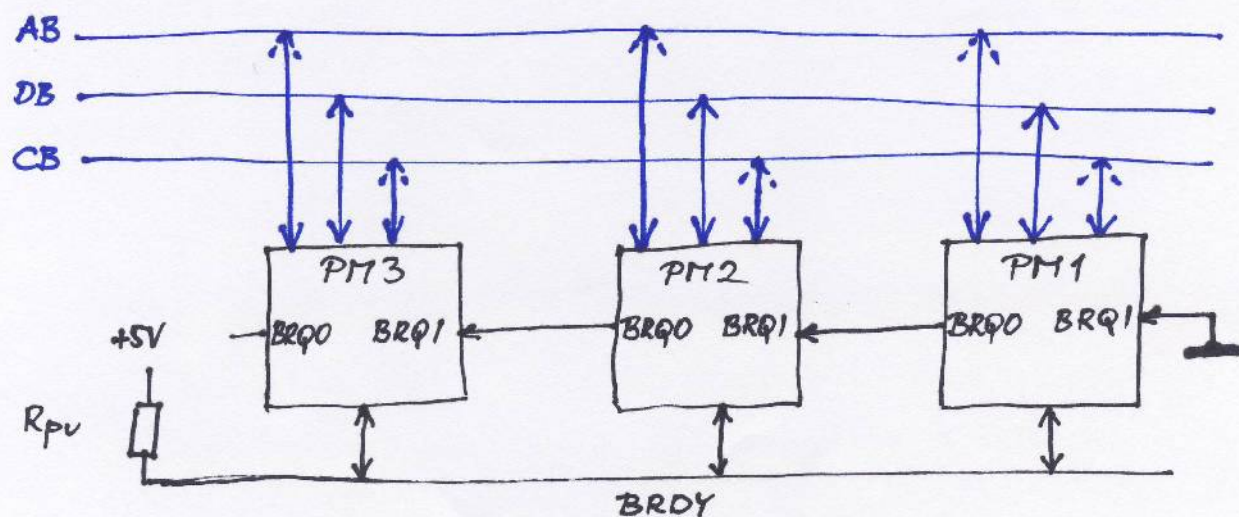
Po odovzdaní týchto informácií začne DMA modul autonómne pracovať s perifériou, pripraví manipuláciu s požadovanými dátami a požiada Mastra (procesor) o prístup k zbernici. Hneď, ako master zbernicu uvoľní, zmení DMA modul svoje práva prístupu na zbernicu na "Master", uskutoční požadovanú manipuláciu s dátami, opäť uvoľní zbernicu (zmení svoje práva na Slave), aby ako Master mohol pracovať procesor. To, že už požadovanú činnosť vykonal oznámi modul DMA procesoru dohodnutým spôsobom (zápisom na systémovú adresu, nastavením príznaku, prerušením,...).

8.3.1. Pridelovanie zbernice v systéme s viacerými potencionálnymi mastrami

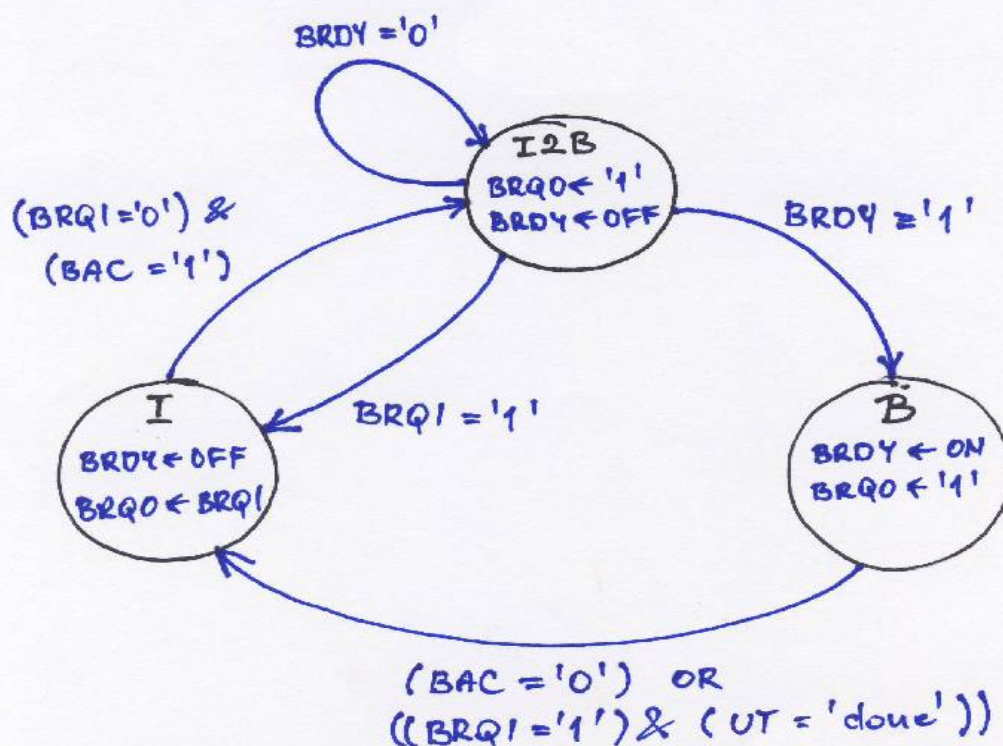
V prípade, ak je na zbernici viac potencionálnych mastrov, je treba definovať mechanizmus, akým spôsobom sa budú jednotlivé mastre v riadení zbernice striedať. Najjednoduchší, nie však jediný spôsob je systém "Daisy chaining", ktorý stanovuje priority pridelovania zbernice jednotlivým potencionálnym mastrom podľa ich umiestnenia v reťazci na zbernici. Na obrázku 8.3.1.1. sú zobrazené tri potencionálne mastre prepojené systémom "Daisy chain". Master PM1 má najvyššiu a PM3 najnižšiu prioritu pri súčasnej žiadosti o zbernicu. Každý z mastrov má vstup **BRQI** (Bus Request In) a **BRQO** (Bus Request Out), ktoré sú prepojené do reťazca. Každý z modulov je tiež pripojený na signál **BRDY** (Bus Ready), ktorý je odporom R_{pu} pripojený na napájacie napätie, čím je v čase, keď nie je riadený žiadnym výstupom zabezpečené aby bol v stave log.1. Každý z potencionálnych mastrov môže čítať stav tohto signálu. Môže ho tiež nastaviť na hodnotu 0 (pomocou otvoreného kolektorového výstupu).

Správanie sa každého potencionálneho mastra je zobrazené na obrázku 8.3.1.2. Stav označený symbolom **I** je kľudový stav, keď master nie je na zbernici aktívny. Naopak stav označený ako **B**, je stavom, keď master riadi zbernicu. Pomocný stav **I2B** je prechodový stav k získaniu riadenia zbernice.

Predpokladajme, že sledovaný master je v stave I (neriadi zbernicu). V takomto prípade signál BRQI prenesie na výstup BRQO, aby postúpil prípadnú žiadosť mastra s vyššou prioritou ďalšiemu mastrovi s nižšou prioritou v reťazci. Ak vznikne taká situácia, že by sledovaný master potreboval riadiť zbernicu a master s vyššou prioritou si ju nežiada ($BRQI = 0$), nastaví na výstup $BRQO = 1$, čím požiada mastre s nižšou prioritou o zbernicu. Nakoľko master riadiaci zbernicu nemôže žiadosti okamžite vyhovieť, pretože musí ukončiť aktuálnu neprerušiteľnú činnosť (napríklad započatý prenos



Obr. 8.3.1.1.
Mechanizmus podelovania zbernice.



BAC - snaha potencionalneho mastera o aktivitu kazbernici
 UT = 'done' - aktualna nepreusitelna cinnost ukoncena

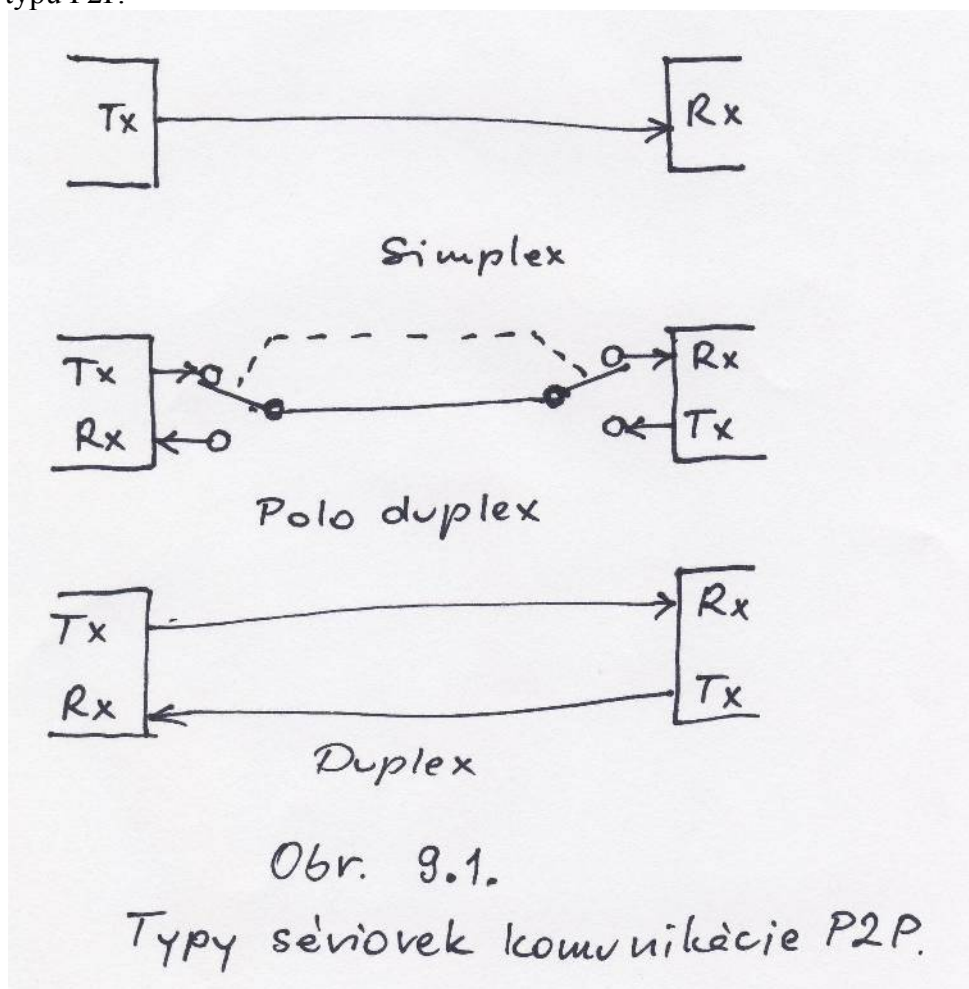
Obr 8.3.1.2.
Mechanizmus podelovania zbernice

dát na zbernici), sledovaný master sa neujme hneď riadenia zbernice, ale sleduje stav signálu BRDY, ktorý je držaný otvoreným kolektorovým výstupom mastra pracujúceho na zbernici v stave 0. Keď tento master uvoľní zbernicu, nastaví sa BRDY na 1, sledovaný master sa môže ujať riadenia zbernice, čo tiež vyznačí nastavením signálu BRDY = 0 (stav B).

Sledovaný master môže ukončiť riadenie zbernice z dvoch dôvodov. Jedným je ten, že už k svojej činnosti riadenie zbernice nepotrebuje ($BAC = 0$) a druhým je žiadosť mastra s vyššou prioritou o zbernicu ($BRQI = 1$). V druhom prípade ukončí aktuálnu neprerušiteľnú činnosť a rovnako ako v prvom prípade prejde do stavu "I" kde nastaví $BRQO = BRQI$ a odpodne svoj výstup, riadiaceho signálu BRDY.

9. Sériový prenos dát

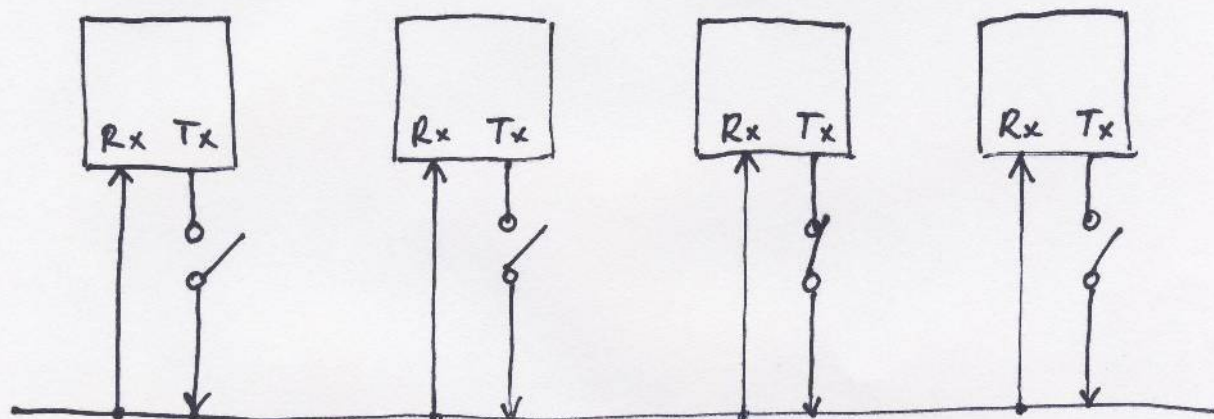
Pod pojmom "sériový prenos dát" označujeme taký prenos, kedy je šírka dátového toku 1 bit. Hlavnou výhodou takeéhoto prenosu je technická nenáročnosť prenosovej cesty. Sériový prenos dát môže byť použitý pre komunikáciu dvoch účastníkov (P2P Point to point) (napr. RS232), alebo pre komunikáciu väčšieho počtu účastníkov (RS485, CAN, ...). Na obrázku 9.1. sú zobrazené tri spôsoby komunikácie typu P2P.



Vysielač informácie je označený symbolom **Tx** a prijímač symbolom **Rx**. V prípade **simplexnej** komunikácie je prenos dát možný iba jedným smerom, **poloduplexná** komunikácia dovoľuje obojsmerný prenos, avšak v jednom okamžiku môže byť komunikácia iba jednosmerná, zatiaľ čo **duplexná** komunikácia je obojsmerná a dáta môžu byť prenášané súčasne obidvoma smermi.

Na obrázku 9.2. je naznačená situácia pri prenose dát po sériovej zbernici. Každý modul

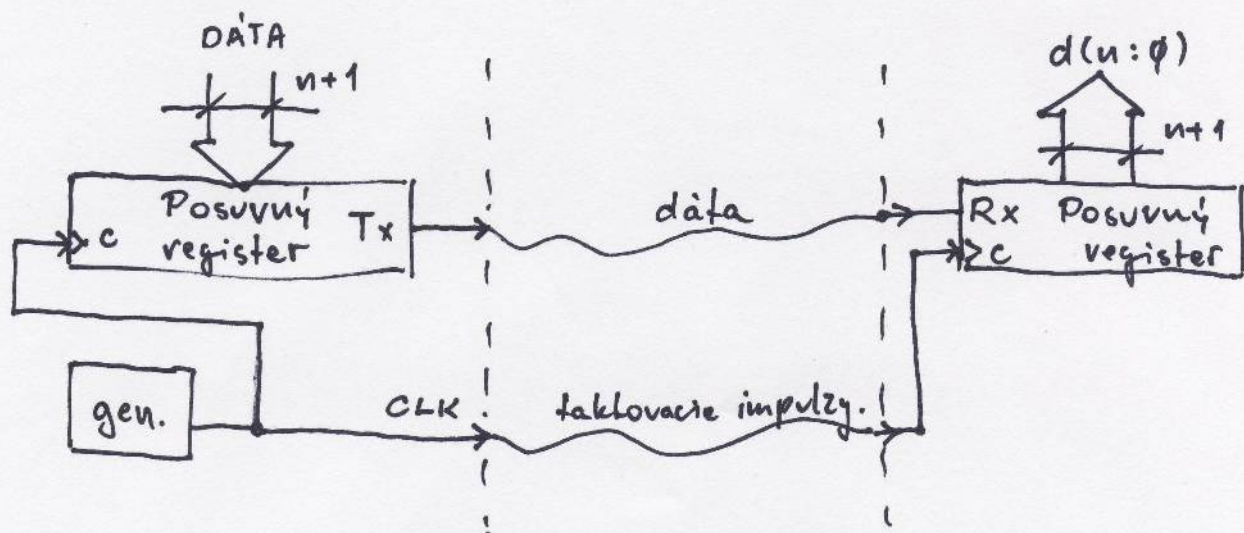
pripojený na zbernicu je prijímač a zároveň potenciálny vysielateľ. V čase, ak nevysiela, je výstup vysielateľa vo vysokoimpedančnom stave. Protokol musí byť navrhnutý tak, aby súčasne nevysielalo viac vysielateľov. Musí byť teda zabezpečená takzvaná **arbitráž** prístupu na zbernicu, kde v prípade snahy viacerých vysielateľov o vysielanie dát, zostane vysielateľ iba jeden. Princíp arbitráže môže byť rôzny, od náhodného výberu až po prioritný.



Obr. 9.2.
Situácia na sériovej zbernici.

9.1. Taktovanie sériového prenosu

Vysielateľ sériového prenosu vysielá dáta v pravidelných intervaloch. Základným problémom taktovania sériového prenosu je, aby prijímač čítal dátové bity v rovnakých okamžikoch, ako ich vysielateľ vysielá, prípadne iba s takou chybou v čase, ktorá nespôsobí ich chybnú interpretáciu.



Obr. 9.1.1.
Princíp synchronného prenosu dát.

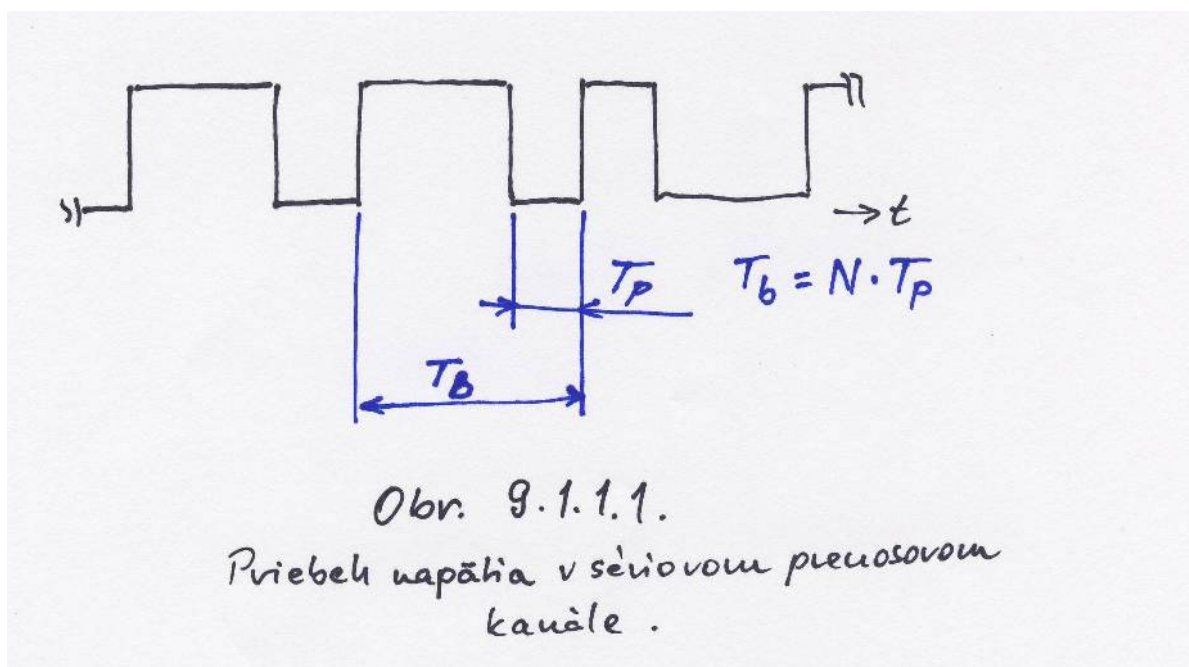
Na obrázku 9.1.1. je principiálne usporiadanie obvodov, ktoré sériovo prenášajú dátové slová šírky $n + 1$ bitov. Na strane vysielača sa dáta synchronne s taktovacím signálom CLK zapíšu do výstupného posuvného registra, z ktorého sú vysúvané na nábežnú hranu taktovacích impulzov. Na strane prijímača sa tento sériový signál vsúva na závernú hranu do vstupného posuvného registra s paralelným výstupom. Spolupracujúce obvody môžu potom v pravý čas (vždy po $n + 1$ impulzoch) prevziať dáta v paralelnej forme z výstupu $d(n:0)$. V uvedenom príklade sa medzi vysielačom a prijímačom okrem sériových dát prenášal aj taktovací signál. Takýto sériový prenos nazývame **synchronný**. Synchronný sériový prenos je logicky najjednoduchší a dovoľuje prenášať dáta s najmenšou "réžiou", pričom pod pojmom réžia rozumieme prenos bitov, ktoré nepredstavujú užitočnú informáciu (dáta), ale informáciu technologického charakteru, ktorá je potrebná pre správnu interpretáciu prenášaného signálu. Synchronný prenos, pri ktorom sa zvláštnym kanálom prenášajú aj taktovacie impulzy má tú nevýhodu, že okrem dátového kanálu je potrebné vybudovať aj kanál prenášajúci taktovaciu informáciu. Preto sa takýto typ prenosu používa iba na krátke vzdialenosti, veľmi často iba vo vnútorných štruktúrach zariadenia.

Na prenos krátkych správ sa častejšie používa takzvaný **asynchronný** sériový prenos dát, kde sa synchronizácia vysielača a prijímača deje vyslaním synchronizačnej udalosti na začiatku skupiny bitov.

V snahe odstrániť potrebu zvláštného prenosového kanálu pre taktovacie impulzy boli vytvorené niektoré spôsoby kódovania dát a taktovacieho signálu do jediného toku informácií. Pre porovnanie vlastností jednotlivých metód kódovania bude užitočné, keď sa v zjednodušenej forme budeme venovať otázke vzťahu rýchlosti prenosu dát a šírky pásma prenosového kanálu.

9.1.1. Vzťah šírky pásma prenosového kanálu a maximálnej rýchlosti prenosu

Principiálne platí, že rýchlosť prenosu je priamo úmerná šírke pásma prenosového kanálu, ktorá je ovplyvnená hlavne jeho hornou medznou frekvenciou. Čo pod pojmom "rýchlosť prenosu" rozumieme? Tok dát sériového prenosu sa prejavuje ako priebeh napätia v čase. Ak vykonáme frekvenčnú analýzu tohto signálu, vieme zistiť najvyššiu významnú frekvenčnú zložku, potlačením ktorej by došlo k významnému skresleniu prenášaného signálu. Zjednodušene je možné povedať, že horná medzná frekvencia prenosového kanálu, ktorá je potrebná na nezkreslený prenos dát závisí od najkratšieho času medzi dvomi zmenami, ktorý sa v toku dát vyskytuje.



Na obrázku 9.1.1.1. je naznačená perióda trvania jedného bitu a tiež naznačený najkratší impulz charakterizovaný časom medzi dvoma zmenami a ich vzťah.

Ak je šírka pásma prenosového kanálu taká, že prenesie impulz dĺžky T_p , potom podľa toho, aký je vzťah medzi dĺžkou impulzu a dĺžkou trvania prenosu bitu je možné vyjadriť využitie kapacity prenosového kanálu ako

$$K = f_b / f_{bM} \quad [9.1.1.1]$$

pričom

$$f_b = 1/T_b = 1/(N * T_p) \quad [9.1.1.2]$$

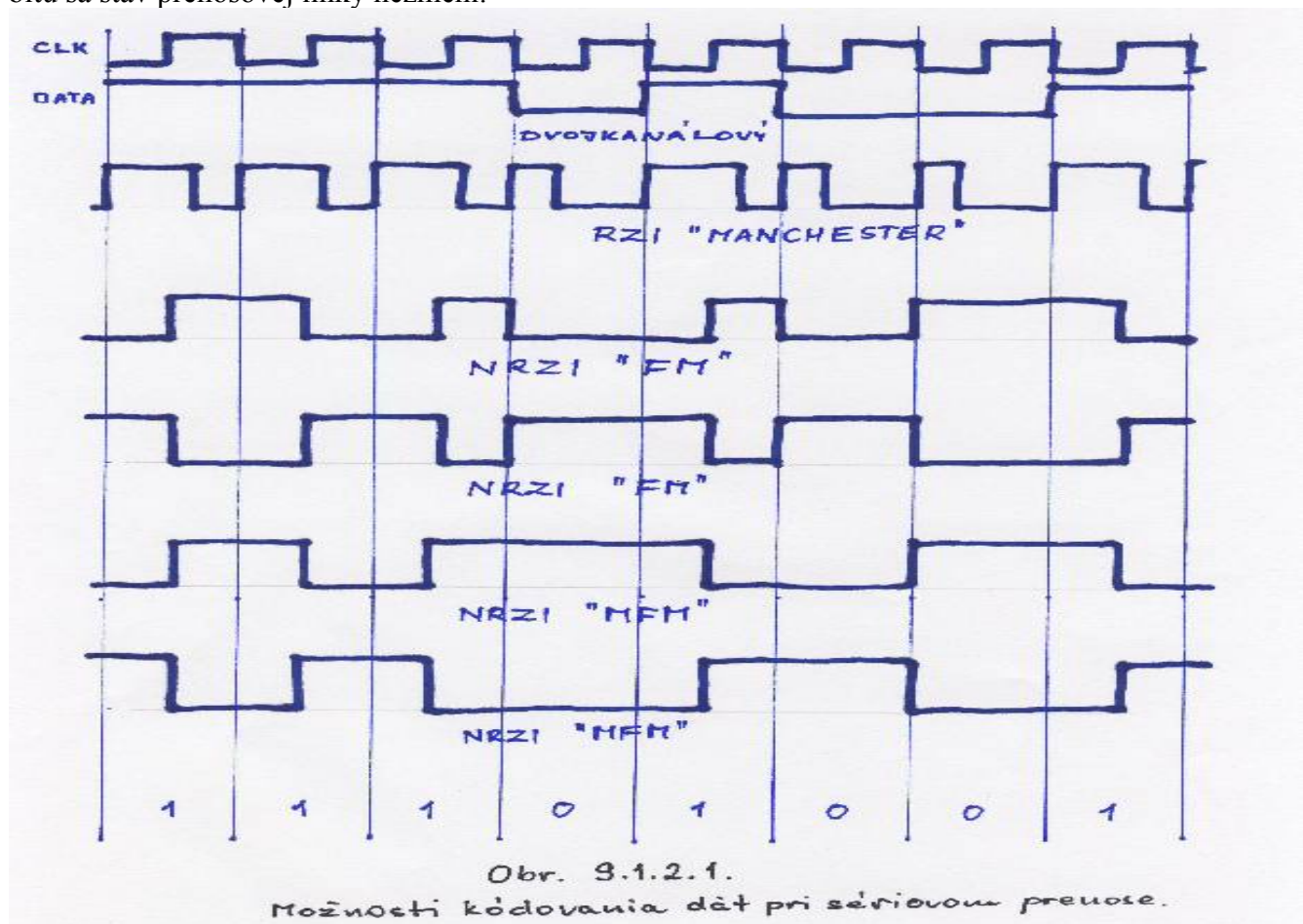
a f_{bM} rýchlosť prenosu dát pri optimálnom využití prenosového kanálu, kedy $N = 1$. Z čoho vyplýva

$$K = (1/(N * T_p)) / (1/T_p) = 1/N \quad [9.1.1.3]$$

kde n je vzťah medzi trvaním prenosu jedného bitu a trvaním najkratšieho impulzu toku dát sériového prenosu $N = T_b/T_p$.

9.1.2. Príklady kódovania dát pri synchronnom sériovom prenose

Kódy používané pri prenose sériových dát je možné rozdeliť na dve skupiny. Prvou z nich sú kódy typu **RZI**, označované aj prívlastkom “s návratom k nule (Return to Zero)”. V prípade týchto kódov stav prenosového kanálu počas trvania jedného bitu nadobudne hodnotu 1 aj hodnotu 0. Druhú skupinu tvoria kódy typu **NRZI**, označované aj ako kódy “bez návratu k nule (Not Return to Zero)”. Pri prenose dát niektorým z týchto kódov sa môže pri prenose vyskytnúť, že počas prenosu jedného bitu sa stav prenosovej linky nezmení.



Na obrázku 9.1.2.1. sú zobrazené štyri rôzne možnosti kódovania sériových dát pri synchronnom prenose slabiky (byte) 0x97 tak, že sa prenáša najmenej významný bit ako prvý. Prvý spôsob je štandardný dvojkanálový prenos, kde sa dáta a taktovací signál prenáša zvláštnymi kanálmi. Pri tomto prenose je dátový kanál prenášaný kódom typu NRZI a taktovací kódom RZI. Prenos taktovacích impulzov kladie na šírku pásma väčšie nároky, pretože $N = 2$, zatiaľ čo pri prenose dát je $N = 1$. Veľmi jednoduchá logika kódovania a dekódovania je v tomto prípade okrem potreby dvoch prenosových kanálov vykúpená aj zníženým využitím kapacity prenosového kanálu $K = 1/2$.

Druhý z kódov je kód typu RZI a je označený aj ako kód typu “**Manchester**”, alebo slangovo aj “bitovo asynchrónny kód”. V tomto prípade sa prenášajú dáta aj taktovacie informácie cez spoločný kanál. Taktovacie impulzy sú vyznačené závernými hranami signálu na začiatku prenosu každého bitu. Dáta sú kódované tak, že ak sa má preniesť 0, potom $2/3$ trvania bitu zostáva signál v nule a ak má byť kódovaná 1, potom zostáva signál v nule iba $1/3$ trvania bitu. K tomu, aby prijímač prečítal správnu hodnotu bitu, postačí, aby čítal v $1/2$ trvania prenosu bitu. Táto veľmi jednoduchá stratégia kódovania a teda i pomerne nenáročné technické vybavenie prenosu je vykúpené pomerne nízkym využitím prenosového kanálu $K = 1/3$.

Tretí z kódov je typu NRZI a označujeme ho ako Fázová Modulácia (**FM**). Používa jediný prenosový kanál pre dáta a taktovanie. Nula je kódovaná tak, že zmena stavu kanálu sa vyskytne na počiatku prenosu bitu a jednotka tak, že zmena sa uskutoční v $1/2$ času prenosu bitu. Z týchto informácií je možné rekonštruovať tak dáta, ako i taktovacie impulzy aj keď technické vybavenie kódera a dekódera je zložitejšie ako v prípade prenosu kódom typu “Manchester”. Využitie kapacity prenosového kanálu je však $K = 1/2$, čo je priaznivejšie.

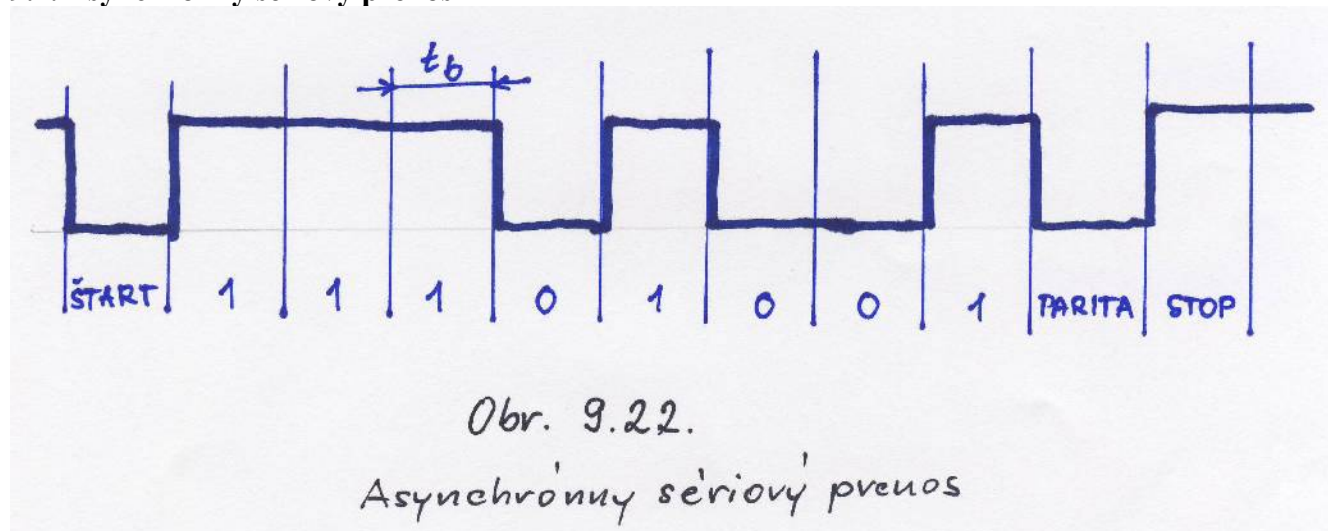
Štvrtý z kódov ponúka ďaleko najlepšie využitie prenosového kanálu ($K = 1$), avšak za cenu pomerne zložitého technického vybavenia kódera a dekódera. Je známy pod označením “Modifikovaná Fázová Modulácia” (**MF**FM). Od FM sa líši iba tým, že ak nasleduje po bite s hodnotou 1 bit s hodnotou 0, neuskutoční sa ani na začiatku ani v strede zmena v nulovom bite bezprostredne nasledujúcom po bite kódujúcom jednotku.

Treba ešte poznamenať že všetky prenosi, kde sú dáta kódované niektorým z kódov NRZI, potrebujú, aby bol prenos na začiatku synchronizovaný prenesením bloku dát so známym obsahom (napríklad blok nulových bitov známej dĺžky (takzvaný **synchronizačný blok**)).

MF

FM ponúka ideálne využitie prenosového kanálu. Ďalšie zvýšenie kapacity kanálu je možné už len pomocou kompresných metód, ktoré sú aplikované na oveľa väčší blok dát ako jeden bit.

9.2. Asynchrónny sériový prenos



Na obrázku 9.2.2. je zobrazený typický sled udalostí pri asynchrónnom sériovom prenose. Prenosová linka má definovaný kľudový stav, v ktorom sa nachádza v čase, keď žiaden prenos neprebíha. Treba povedať že vysielateľ a prijímač musia mať pred prenosom dohodnutú rýchlosť prenosu s presnosťou, ktorá závisí od množstva dát, ktoré po synchronizačnej udalosti prenáša. Prenos začína synchronizačnou udalosťou. Po synchronizačnej udalosti sa uskutočňuje prenos dát, potom nasleduje prípadná ochranná informácia dát a na konci úsek s vnúteným kľudovým stavom, ktorý prechádza po definovanom čase do prirodzeného kľudového stavu. Rozdiel medzi vnúteným a prirodzeným kľudovým stavom je ten, že na rozdiel od prirodzeného, počas vnúteného kľudového stavu nie je možné začať nový prenos.

Najčastejší typ asynchrónneho prenosu je zobrazený na obrázku 9.2.2. Často sa v literatúre označuje ako RS232, aj keď toto označenie nie je presné, lebo tento typ prenosu sa používa aj inde ako v súvislosti s rozhraním definovaným normou ako RS232. Kľudový stav prenosovej linky je log. 1. Na začiatku prenosu sa vysielateľ štarovací bit s hodnotou log. 0 v trvaní t_b , čo je dĺžkou trvania bitu. Potom sa vysielateľ podľa dohody 5 až 8 dátových bitov (na obr. 9.2.2. osem bitov (0x97)) počnúc najmenej významným bitom (LSB). Potom podľa dohody môže nasledovať paritný bit na párnou, alebo nepárnou paritu (nepárna), po čom nasleduje podľa dohody 1 až 2 stop bitov, ktoré majú hodnotu log. 1 a plnia úlohu vnúteného kľudového stavu. Každý bit trvá práve t_b . Prijímač považuje za začiatok prenosu prvú závernú hranu (začiatok štart bitu), od toho okamžiku počka $1.5 * t_b$, prečíta hodnotu LSB dát a potom pravidelne každých t_b číta ďalšie bity, paritný bit a overí správnosť stop bitu. V prípade, ak nie je hodnota paritného bitu správna, hlási prijímač spolupracujúcemu zariadeniu chybu parity (**parity error**) a v prípade, ak nemá niektorý zo stop bitov správnu hodnotu hlási tzv chybu ohraničenia (**framing error**).

Ak by sme skúmali využitie kapacity kanálu z hľadiska prenosu jedného bitu a použili by sme metodiku, ktorú sme používali pri skúmaní vlastností synchronného prenosu, dospeli by sme k záveru, že $K = 1$. Ak však uvažíme, že k prenosu užitočnej informácie potrebujeme bez ochrany a s jedným stop bitom dva bity technologickej informácie (štart bit a stop bit), vychádza využitie kapacity kanálu na $8/10$ tj $4/5$. Čo je pomer užitočných (dátových) bitov k celkovo preneseným bitom. Tento pomer je najpriaznivejší. Platí pri prenose 8 bitov bez ochrany s jedným stop bitom. Každá odchylka od týchto parametrov má za následok nižšie číslo.

Pri asynchrónnom prenose dát sa rýchlosť prenosu udáva v **Baudoch** (Bd). Vypočítame ho ako:

$$B = 1/t_b \quad [\text{Bd}]$$

kde t_b je definované na obrázku 9.2.2.

Nasledujúce rýchlosti prenosu sú obvyklé: 300 Bd, 600 Bd, 1200 Bd, 2400 Bd, 4800 Bd, 9600 Bd, 19200 Bd, 115200 Bd, 1 Mbd, 2 Mbd, 4 Mbd.

Vysielateľ i prijímač musí používať rovnakú rýchlosť prenosu. Je vhodné vedieť, že na presnosť taktovacích frekvencií sú v prípade asynchrónneho prenosu kladené zhruba o rád vyššie nároky ako pri synchronnom prenose.

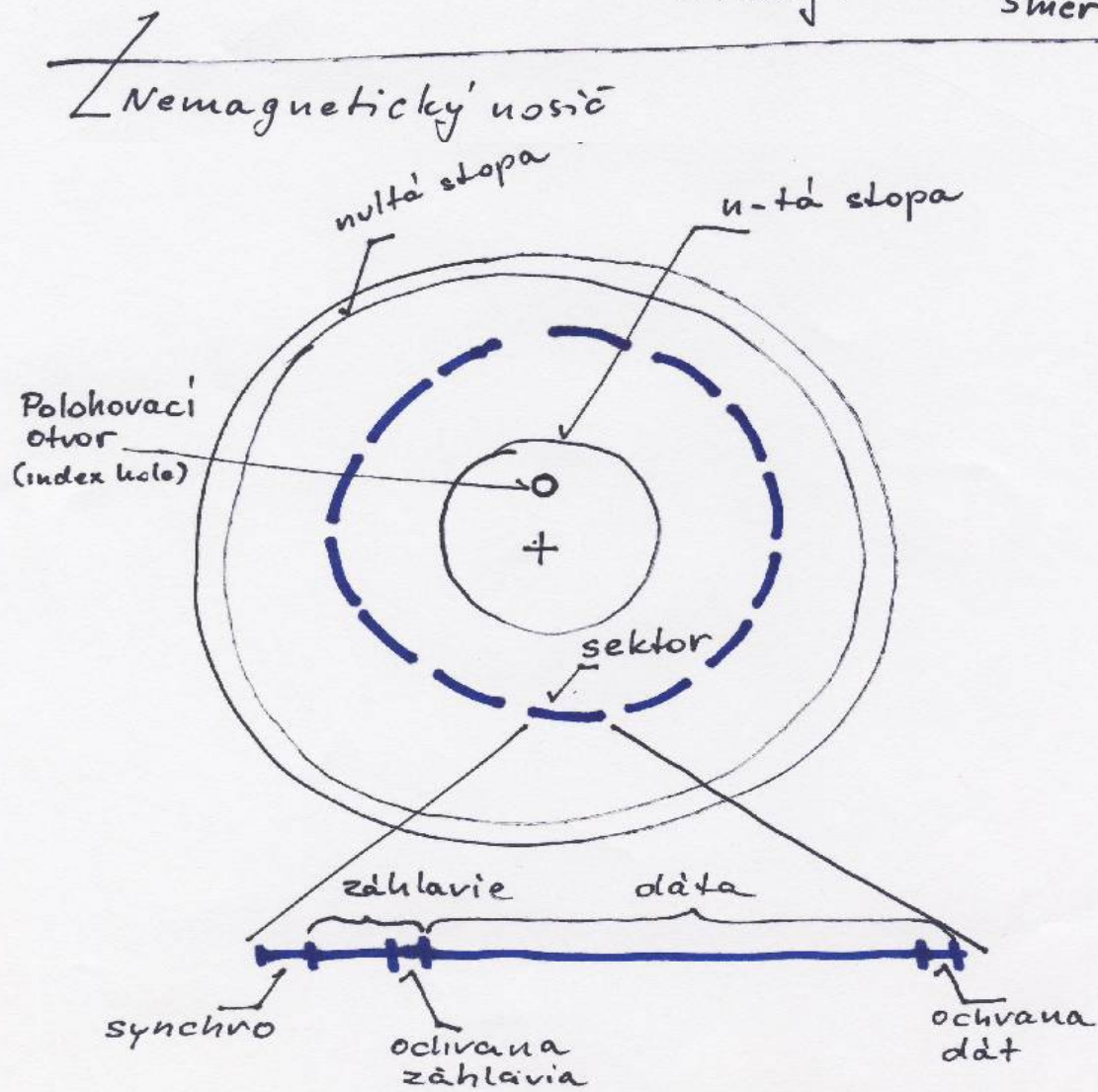
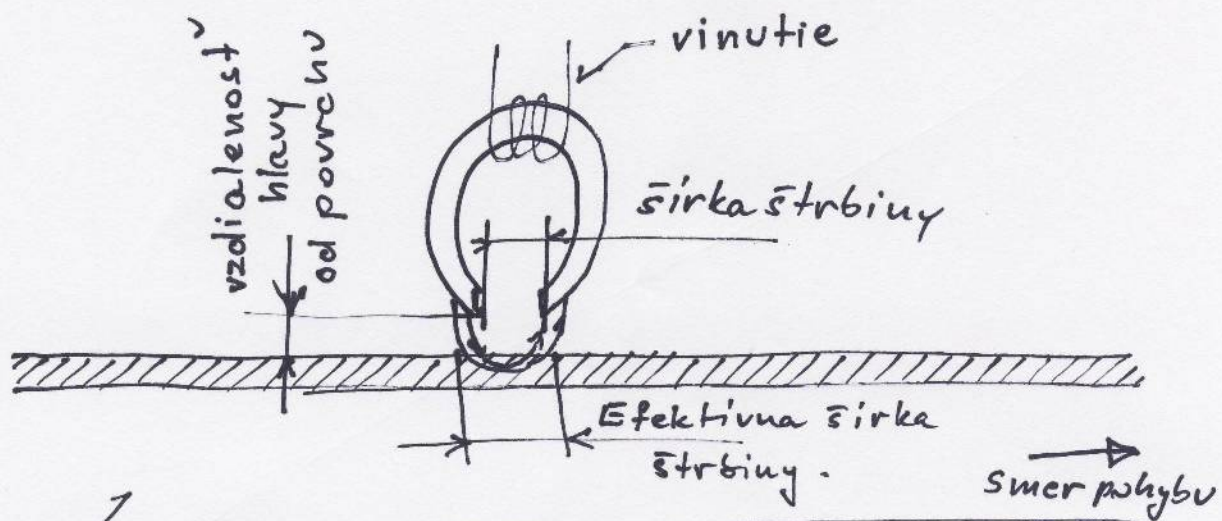
10. Magnetický záznam digitálnej informácie

10.1. Princíp záznamu

Princíp záznamu na pohybujúci sa magnetický povrch je naznačený na obrázku 10.1.1.

Nosič informácie (**médium**) sa skladá z nosiča vyrobeného z nemagnetického materiálu, ktorý je pokrytý tenkou vrstvou magnetického materiálu s veľkou hysterézou. Záznam a čítanie informácie sa vykonáva pomocou **hlavy**, ktorá je tvorená jadrom vyrobeným z magnetického materiálu s malou hysterézou. Na mieste, kde je hlava najbližšie k médiu je vytvorená štrbina z magneticky nevodivého materiálu, takže sa magnetické pole uzatvára cez magneticky vodivý materiál média a zanecháva na ňom magnetickú stopu, ktorej veľkosť označujeme aj ako efektívna šírka štrbiny. Veľkosť efektívnej

šírky štrbiny závisí (okrem iných vplyvov) hlavne od fyzickej šírky štrbiny a od vzdialenosti hlavy od média.



Obr. 10.2.1
Usporiadanie záznamu na disku.

Veľkosť efektívnej šírky štrbiny je hlavným faktorom, ktorý ovplyvňuje takzvanú hustotu záznamu (počet bitov na jednotku dĺžky záznamu), od ktorej závisí kapacita média. Čím je efektívna šírka štrbiny menšia, tým môže byť hustota záznamu väčšia. Inými slovami, čím je fyzická šírka štrbiny menšia a čím je hlava bližšie k povrchu, tým je možná väčšia hustota záznamu.

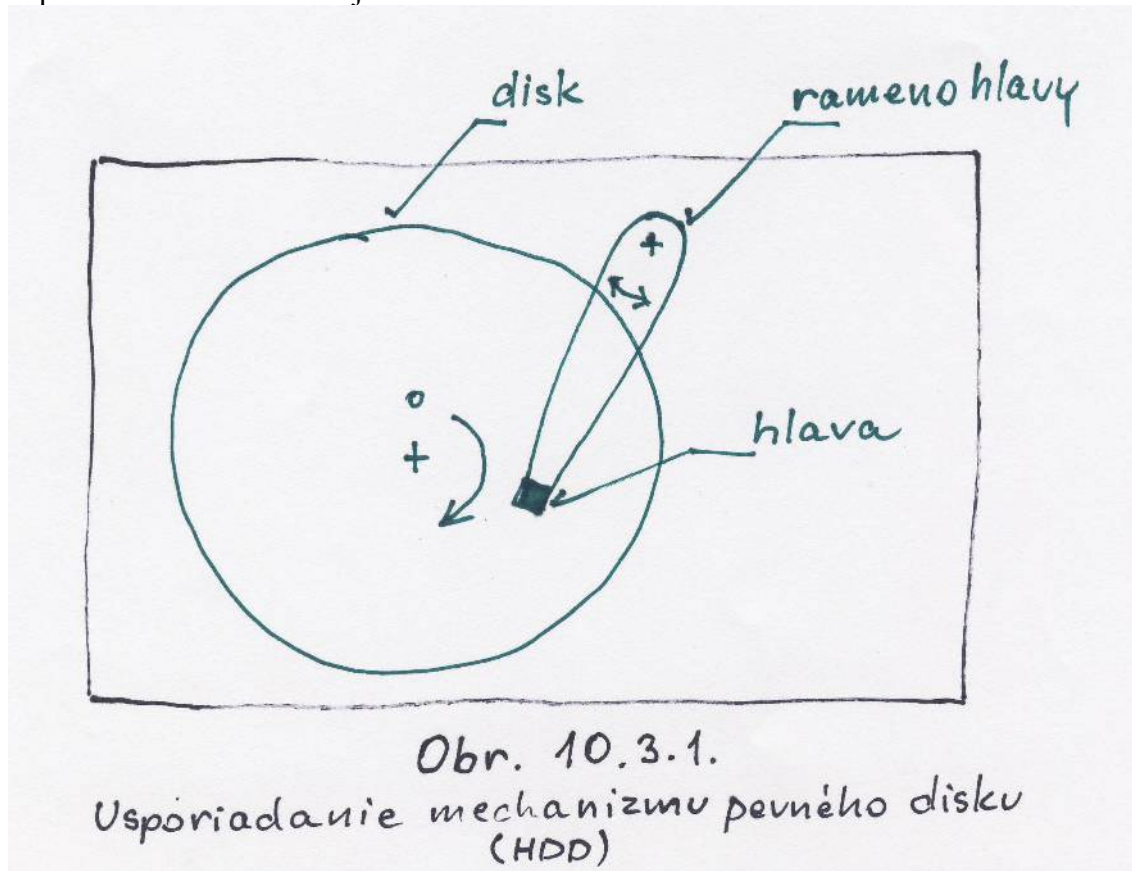
10.2. Usporiadanie záznamu na magnetickom disku

Na obrázku 10.2.1. je zobrazený spôsob usporiadania záznamu na disku. Záznam je usporiadaný do stôp. Stopa má tvar kružnice a skladá sa zo sektorov. Sektor je členený na záhlavie a dáta. Obidve tieto časti sú chránené kontrolnou sumou (CRC), ktorá je schopná s veľkou pravdepodobnosťou detekovať chybu. Záhlavia sa vytvárajú pri fyzickom formátovaní disku a pri zápise dát sa už neprepisujú. Záhlavie obsahuje identifikačné údaje sektora. Sektory sú oddelené medzerou bez záznamu a každý sektor začína synchronizačnou sekvenciou skúšiacou na synchronizáciu čítacích obvodov (viď. moduláciu FM a MFM).

Médium (disk) je opatrený polohovacím otvorom (index hole), pomocou ktorého sa určuje poloha nultého sektora v každej stope počas formátovania.

10.3. Mechanizmus pevného disku (HDD)

Usporiadanie mechanizmu je na obr. 10.3.1.



Skladá sa z disku (diskov) vyrobeného z nemagnetického materiálu (hliník, sklo,...) a je pokrytý tenkou vrstvou magnetického materiálu, ako je kysličník železa, alebo častejšie zliatina kobaltu.

Záznamová/čítacia hlava (hlavy) je upevnená na pohyblivom ramene, ktoré sa otáča okolo osi pomocou elektromagnetického mechanizmu. Nastavením polohy ramena sa volí aktuálna stopa záznamu.

Hlava nemá kontakt s povrchom disku, ale sa vznáša pôsobením aerodynamických síl vzduchu

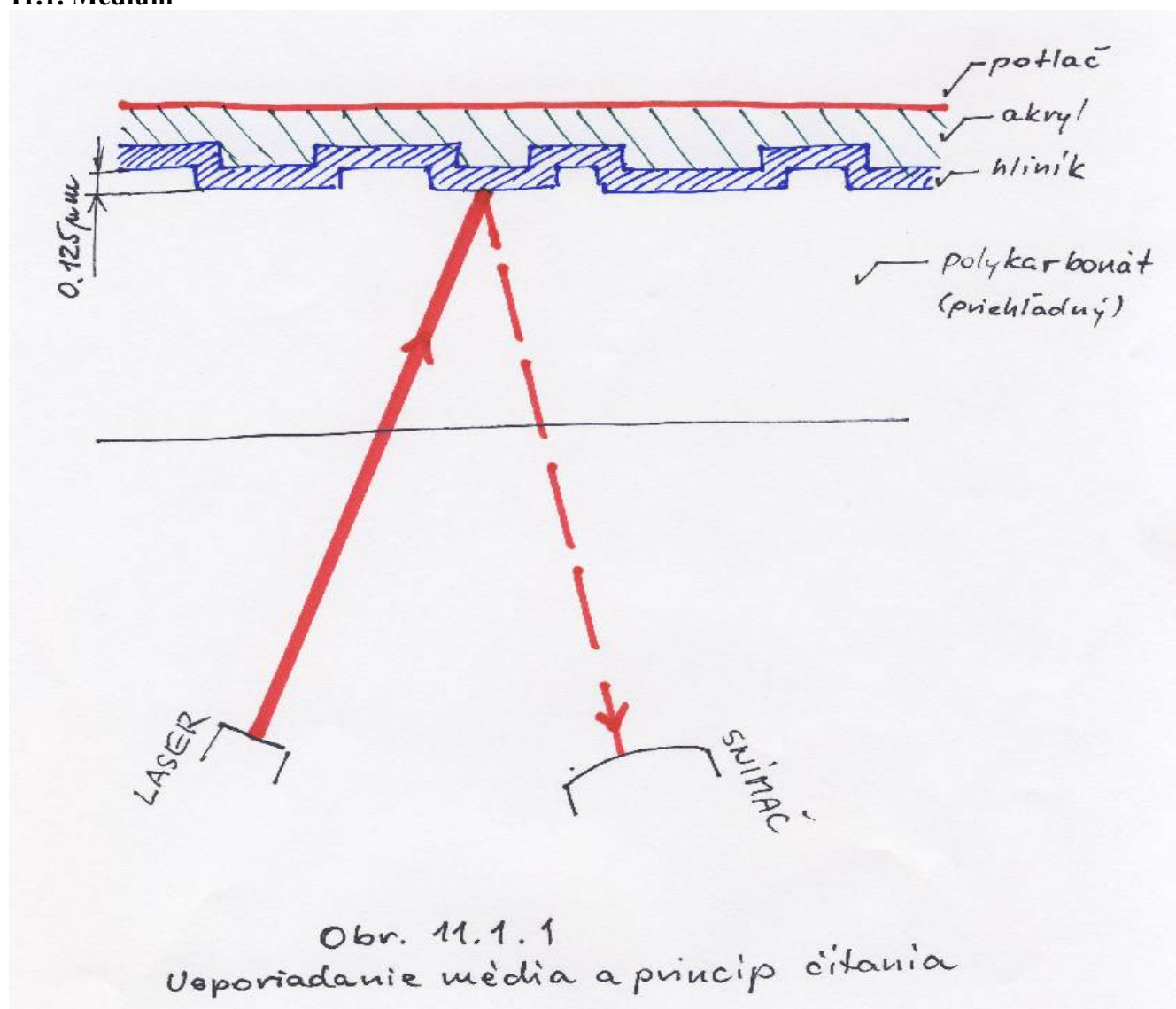
unášať pohybuje sa povrchom disku. Extrémne hladký povrch disku dovolí nastaviť malú vzdialenosť hlavy od povrchu disku, čo vedie spolu s úzkou štrbinou hlavy na veľmi vysokú hustotu záznamu.

K tomu, aby bolo možné nastaviť malú vzdialenosť hlavy od disku, musí obal disku a jeho ventilačný mechanizmus zabrániť vniknutiu prachových častíc, ktoré by mohli pri kolízii s hlavou spôsobiť poškodenie povrchu disku a teda aj zaznamenaných údajov.

11. Optický záznam údajov

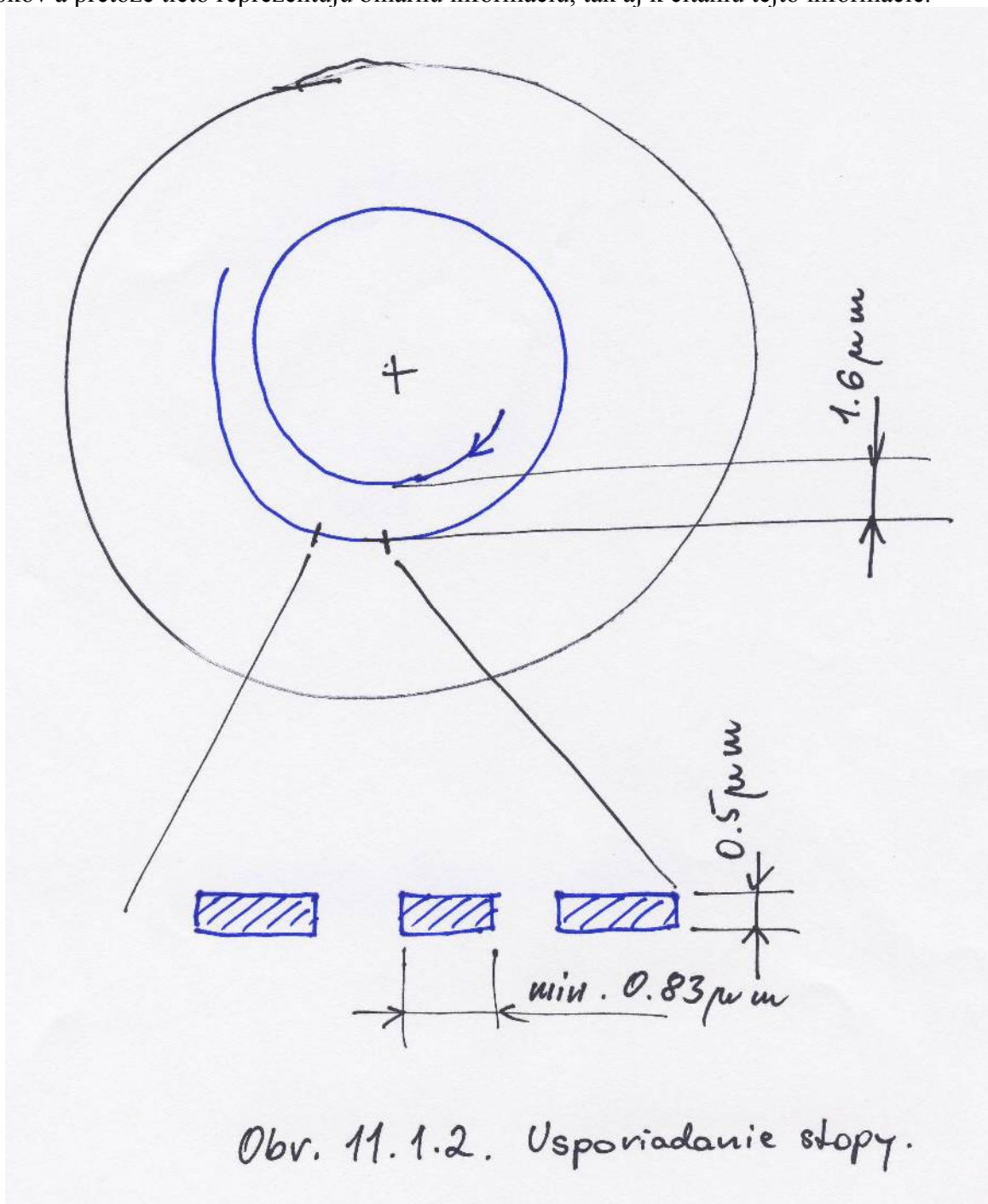
V tejto kapitole sa budem zaoberať princípom záznamu na médiu známeho ako CD-ROM. Všetky modifikácie tohto záznamu (CR-W, CD-RW, DVD,...) sú iba kvantitatívnou modifikáciou tohto princípu (DVD), alebo sa líšia možnosťou a spôsobom vytvárania záznamu.

11.1. Médium



Na obrázku 11.1.1. je naznačený rez médiom. skladá sa z hornej potlače, ktorá slúži na optickú identifikáciu obsahu, akrylovej vrstvy, ktorá pokrýva tenkú vrstvu hliníka nanesenú na priehľadný polykarbonátový nosič. Pred nanosením vrstvy hliníka je na polykarbonátovom nosiči vytvorený obsah disku výstupkami výšky 0.125μm. Čítanie informácie sa uskutočňuje tak, že zaostrený laserový lúč

dopadá na hliníkový povrch a odráža sa späť, pričom intenzita odrazeného svetla je detekovaná snímačom. Pretože sa intenzita svetla na hranách výstupkov mení, dochádza k detekovaniu výskytu výstupkov a pretože tieto reprezentujú binárnu informáciu, tak aj k čítaniu tejto informácie.

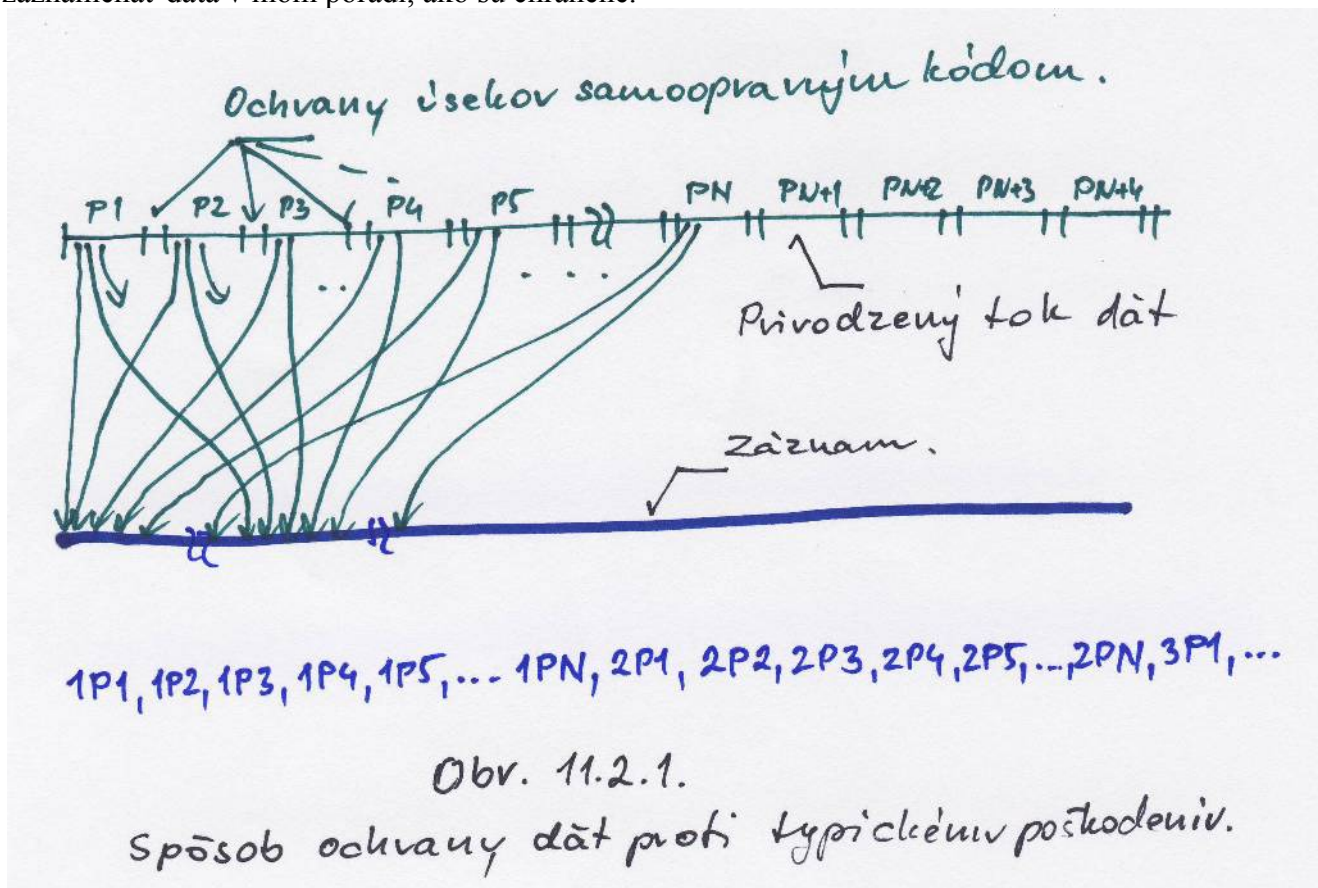


Obr. 11.1.2. Usporiadanie stopy.

Záznam je na povrchu média usporiadaný podľa obrázku 11.1.2.. Na rozdiel od magnetických diskov tvorí špirálu vytvárajúcu sa od stredu média. Tým je zabezpečené, že médium môže byť menšie než štandardných 120mm a bude ho možné bez problémov čítať. Šírka stopy je $0.5 \mu m$ a dĺžka najmenšieho výstupku $0.83 \mu m$. Stopy sú vzdialené $1.6 \mu m$.

11.2. Ochrana záznamu

Vzhľadom na malé rozmery častí záznamu a jeho optický charakter je pravdepodobnosť chybného čítania pôsobením nečistôt, alebo poškodenia povrchu nosiča vysoká. Z tohto dôvodu je prirodzený tok dát rozdelený na úseky, ktoré sú chránené samoopravným kódom schopným opraviť niekoľko chýb. Pretože nečistota na povrchu, alebo jeho poškodenie máva typicky také rozmery, ktoré by spôsobili chybné čítanie väčšieho počtu bitov ako je možné samoopravným kódom opraviť, je účelné zaznamenať dáta v inom poradí, ako sú chránené.



Obrázok 11.2.1. Zobrazuje spôsob reorganizácie dát pred záznamom. Podstatou tejto reorganizácie je to, že postupnosť zaznamenaných bitov je vytváraná z N úsekov chránených samoopravným kódom tak, že sa do postupnosti zaznamenaných bitov ukladajú skôr prvé bity z jednotlivých úsekov, potom druhé, tretie atď.. (včítane ochrannej informácie). Ak pri takto usporiadanej informácii vznikne nečitateľný úsek kratší ako N bitov, môže poškodiť iba jeden bit z každého chráneného úseku, čo je schopný samoopravný kód ochrany úseku opraviť.

Dôsledkom tohto spôsobu ochrany je potreba čítania dát do vyrovnávacej pamäte a pred ich využitím reorganizácia do prirodzenej postupnosti.