

# **Principy uplatňované v technice vysokorychlostních sériových přenosů**

## **Principy využití v SATA**

Cíl přednášky:

Vysvětlení základních principů rozhraní SATA.

Vysvětlení principů kódování 8b/10b.

Zabezpečení SATA.

## **Osnova přednášky:**

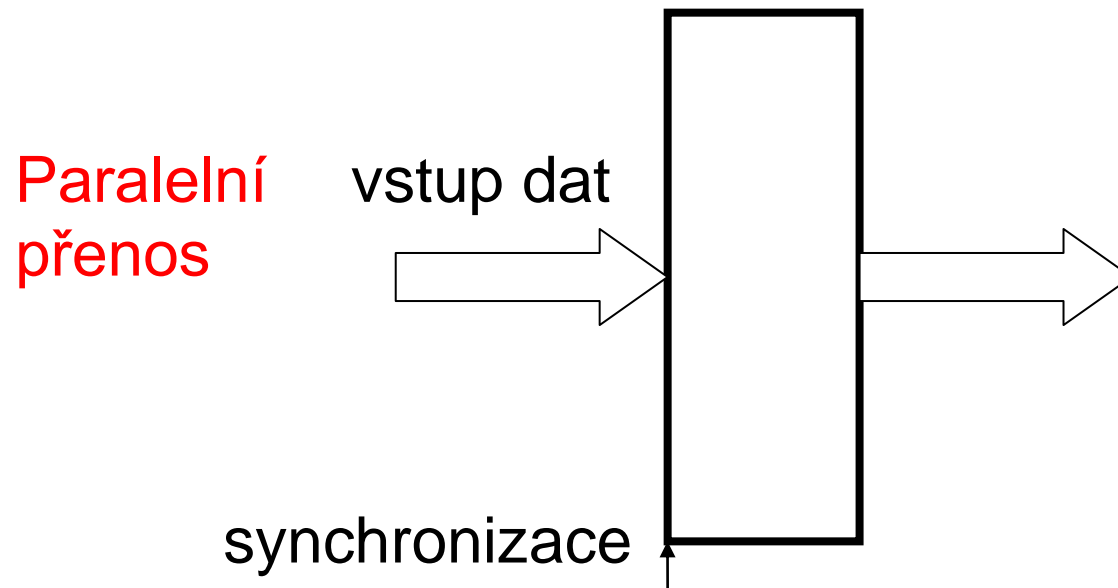
- Paralelní v. sériové přenosy, stav dříve (před 25 lety) v. stav dnes.
- Možnosti synchronizace sériově přenášených dat.
- Zajištění sériových přenosů proti možným chybám v přenosu.
- Kódování přenášených znaků.
- Význam nulové klidové úrovně přenášeného signálu.

- **Paralelní v. sériové přenosy**

- **Stav před 25 lety:** paralelní přenosy byly rychlejší z hlediska rychlosti přenosu (objem dat/s) – právě proto, že byly realizovány paralelně (více bitů přenášených paralelně).
- **Stav dnes:** sériové přenosy jsou viděny jako perspektivní – je jisté, že parametry těchto technik se budou dále zdokonalovat (rychlosti, objemy přenášených dat). To se nedá konstatovat o přenosech paralelních.
- Paralelní techniky jsou využívány, dosáhly však svého vrcholu.

## Možnosti synchronizace paralelně přenášených dat

Data je nutno na přijímací straně synchronizovat.



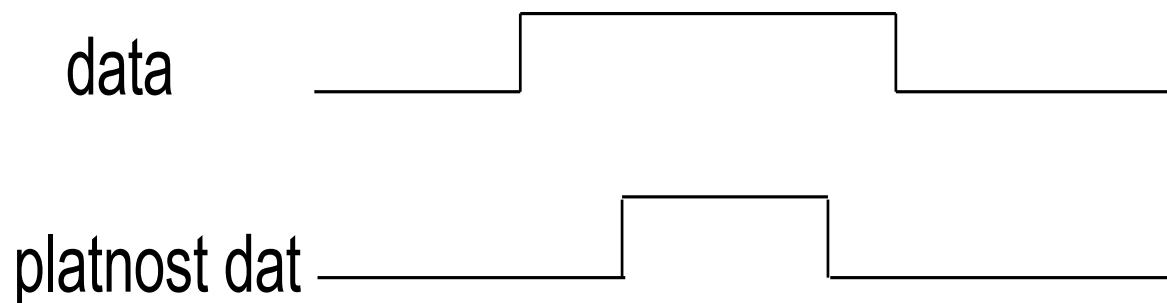
Obr. 1

Data je nutno na přijímací straně synchronizovat, při zvyšování kmitočtu může vzniknout zpoždění na datových vodičích proti synchronizaci (clock skew) => možná chybovost při příjmu dat.

## Vysvětlení „clock skew“

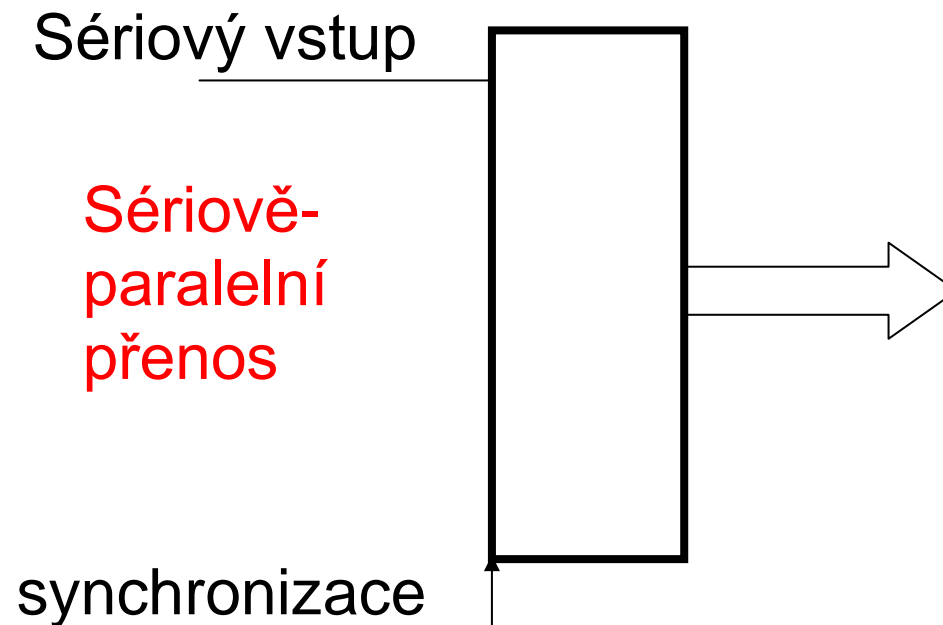
Při paralelně realizovaném přenosu (např. 32 bitů) se při navyšování kmitočtu přenosu zvyšuje pravděpodobnost, že jeden či více bitů nebudou synchronizačním signálem správně vzorkovány.

Nutnost dodržet dynamické parametry typu „předstih / přesah“, aby byla data správně zapsána do registru. Navyšující se kmitočet / navyšování šířky toku dat – problém (vliv např. parazitních kapacit mezi datovými bity).



## Možnosti synchronizace sériově přenášených dat

Data je nutno na přijímací straně synchronizovat a převést na paralelní formu. Při zvyšování kmitočtu opět problém „clock skew“. Označuje se pojmem „deserializér“.



Obr. 2 Deserializér

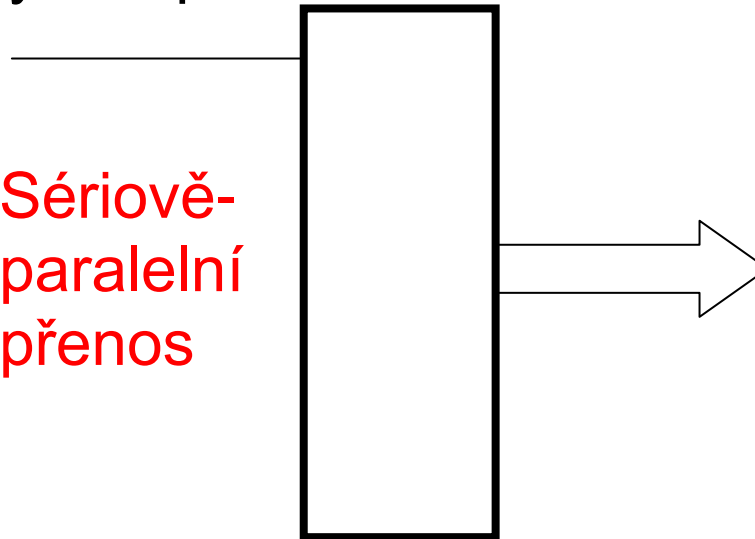
Clock skew – posuv datových bitů vůči synchronizaci.

### Možnosti řešení „clock skew“

Zrušíme přenos synchronizace jako samostatného signálu – budeme ji buď odvozovat z dat nebo bude synchronizace vnitřní (bude ale nějakým způsobem odvozena od kmitočtu snímaných dat – s využitím např. fázového závěsu).

Sériový vstup

Sériově-  
paralelní  
přenos



Obr. 3

- Synchronizace přijímaných dat – možnosti řešení:
  - **Součástí rozhraní jsou pouze data, nikoliv synchronizace** (synchronizace není ani odvozována ze snímaného průběhu) – na přijímací straně jsou generovány synchronizační signály dohodnuté frekvence (dá se realizovat na nízkých kmitočtech).
  - **Synchronizační signály jsou součástí spoje**, pak jsou tyto možnosti:
    - alternativa 1**: synchronizace se odvodí z dat,
    - alternativa 2**: pro synchronizaci je vyčleněn samostatný spoj.



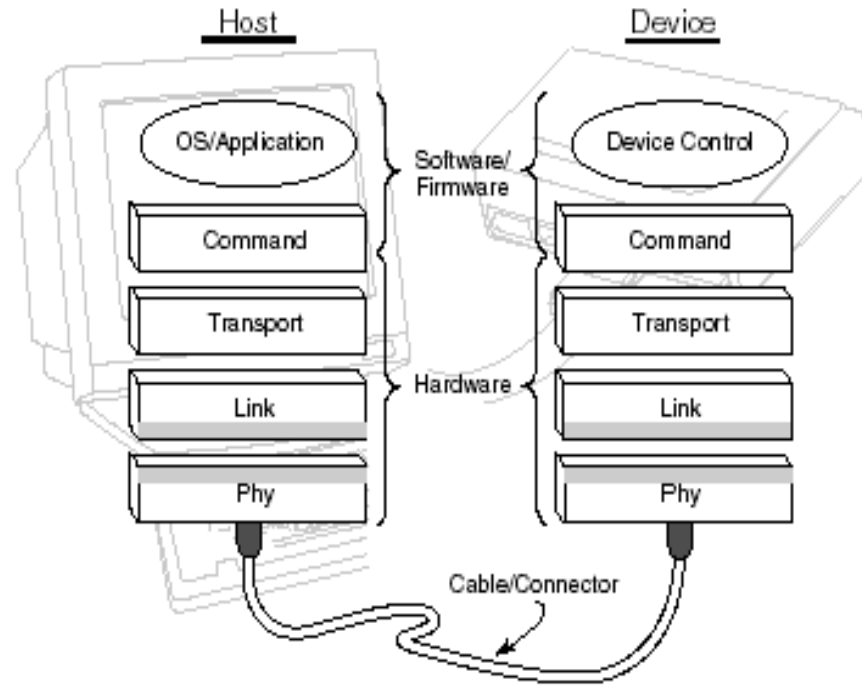
- **Alternativa 1** – synchronizace se odvodí z dat:
  - Pokud jsou zapisovaná data beze změn přenášeného signálu, tak proces získávání synchronizace je obtížný.
  - **Vlastnost, kdy je možno synchronizaci odvodit na přijímací straně z dat s využitím fázového závěsu – takto je to v rozhraní SATA a v PCI Express.**
  - Situace u metod záznamu do diskových pamětí: pokud má signál snímaný z disku podobu takovou, že z něj lze získat synchronizaci (přenášený signál musí mít jisté vlastnosti), pak mluvíme o vestavěné synchronizaci (embedded clock) nebo o záznamu s vlastní synchronizací – dnes se v diskových pamětech už nepoužívá.

- **Alternativa 2** - pro synchronizaci je vyčleněn samostatný spoj:
  - další spoj v kabelu a další pozice v konektoru,
  - **synchronizační signál by měl vyšší kmitočet než přenášená data (změna synchronizačního signálu v každém bitovém intervalu),**
  - z hlediska rušení a přeslechů nevýhodné (tento problém bychom museli řešit kvůli přenosu synchronizace a nikoliv kvůli přenášeným datům - přidáváme si další problém),
  - **pokud by data měla posloupnost 010101..., tak to bude ještě pořád dvakrát nižší kmitočet než kmitočet synchronizace, protože ten se bude měnit uvnitř bitového intervalu,**
  - distribuce synchronizace jinak než s daty činí problémy v situaci, kdy zvyšujeme kmitočet,

- problém: realizovat při vysokém kmitočtu správně vzorkování – zpoždění na datech a synchronizaci mohou být různá a nemusí se nám dařit se správně „trefovat“ vzorkovacími hranami (nebo úrovněmi) synchronizačního signálu do přenášených dat a správně je vzorkovat (clock skew) – **takto je to v rozhraní DVI** (sériový spoj, v němž jsou synchronizace a data vedeny odděleně).

### Závěr:

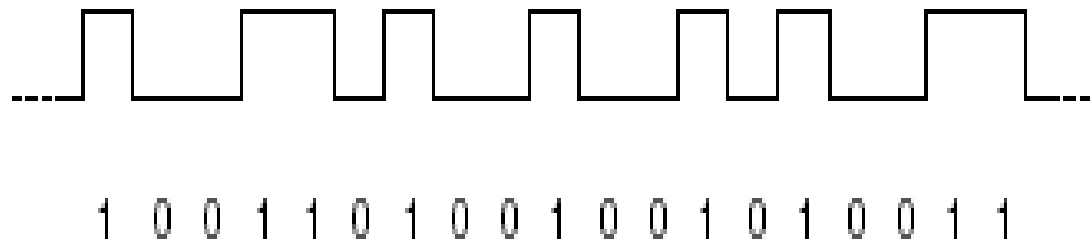
- vysoké kmitočty přenosu – zásadně se nedoporučuje přenášet data a synchronizaci odděleně na samostatných vodičích,
- reflexe v SATA: rozhraní SATA není synchronizováno pulsy přenášenými samostatným vodičem.



Obr. 4

- Vrstva Link je zodpovědná za zakódování přenášených dat, dekodování dat přijatých a za dodržování protokolu. Jednotlivé vrstvy jsou zobrazeny na obr. 4.
- Vrstva Transport – vytváření formátu přenášených dat.

- Obecný poznatek z obr. 4 – pokud se na vysílací straně v patřičné vrstvě něco do přenášených dat vkládá, pak na přijímací straně se tato informace ve stejné vrstvě rozpoznává (odděluje).

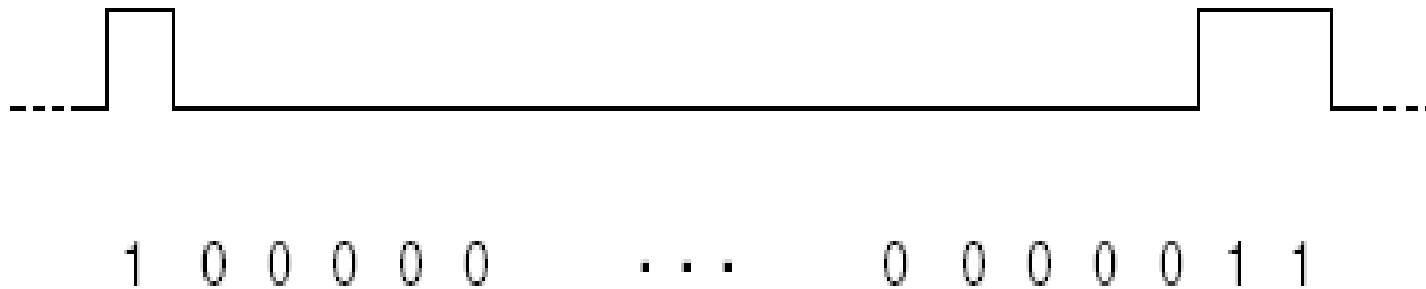


Obr. 5 Ilustrační příklad: tok dat s velkým počtem změn signálu

- Komentář k obr. 5:

Bitový interval je představován šířkou nejkratšího intervalu mezi změnami, pak lze rozpoznat hodnoty v dalších bitech.

Signál reprezentující data má pak takový počet změn (předepsaný), že lze na přijímací straně odvodit synchronizaci – využití techniky fázového závěsu (PLL - Phase Locked Loop). Parametr RL je splněn.



Obr. 6 Ilustrační příklad – data s nízkým počtem změn signálu

Pozn.: tato posloupnost bitů obsahuje málo změn, což činí problémy při rozpoznávání dat – dlouhý interval samých „0“

- Komentář k obr. 6:

Vzhledem k tomu, že přijatá data mají malý počet změn, budou se data zjišťovat obtížněji než v posloupnostech s větším počtem změn. Pokud se průběh nemění, pak je proces zjištění, kolik „neměnicí se signál“ představuje bitů, výrazně obtížnější.

- Závěr:
  - Pokud nejsou v datech zakódovány synchronizační pulsy, pak v přijímaných datech musí existovat změny signálu.
  - **Parametr, který reprezentuje počet bitů, po něž se signál nemění, se nazývá Run Length (délka běhu). Ten může nabývat pouze povolenou hodnotu.**
  - Pokud má tento parametr požadovanou hodnotu, pak je možné data z přijaté posloupnosti změn získat.
- Odraz těchto principů v technikách sériových vysokorychlostních přenosů:
  - Co musíme dosáhnout: požadovanou hodnotu RL a nulovou klidovou úroveň přenášeného signálu (přenášené hodnoty napětí: +- 250 mV),
  - Oba požadavky jsou řešeny způsobem kódování 8b/10b.
  - **Cíl kódování: dosažení parametru RL a stejný počet „0“ a „1“ v přenášeném signálu.**



- Základní principy kódování 8b/10b v rozhraní SATA:  
256 původně 8 bitových znaků je zakódováno do 10 bitů.  
8 bitů nabízí vytvořit celkem 256 různých kombinací, 10 bitů celkem 1024 kombinací. Z této množiny je možné vybrat potřebný počet kombinací tak, aby splňoval konkrétní požadavky.
- Jedním z cílů kódování 8b/10b je právě dodržení parametru RL na přijatelné hodnotě.
- Dosáhne se toho tak, že se z 10 bitových symbolů vybírají pouze takové kombinace, které tuto vlastnost mají.
- Nejsou např. využity kombinace se samými „1“ a se samými „0“. Z 1024 kombinací je vybráno pouze 256 takových hodnot, které tyto vlastnosti mají a vyloučí se takové hodnoty, které tyto požadavky nesplňují.

- Základní vlastnosti SATA:
  - I když rychlost přenosu je např. 1,5 Gb/s (dnes už 3 Gb/s a 6 Gb/s, je rychlost přenosu přepočtená na data 150 MB/s (1500/10) a nikoliv 187,5 MB/s (1500/8).
  - Tato „nesrovnalost“ souvisí se způsobem kódování, který je v sériovém ATA využíván.
  - Ten pro zakódování dat využívá 10 bitů, nikoli 8.
  - Je to princip, který je využíván u všech vysokorychlostních sériových přenosů (i když třeba v jiné podobě).
  - To má sice nevýhody v tom, že se objem přenesených dat redukuje o 20 %, tento princip má však řadu jiných výhod, které zde budou diskutovány (IBM patent 4 486 739 – [www.uspto.gov](http://www.uspto.gov)).

- Další požadavek na kódování 8b/10b:
  - **Nulová klidová úroveň přenášeného signálu.**
  - Ten se splní tak, že zakódovaná data obsahují stejný počet „0“ a „1“.
  - **Pozn.:** jiná možnost, jak to splnit, je kapacitní vstupní člen, ten zablokuje ss napětí.
  - Vhodnější je řešení – stejný počet „0“ a „1“ v přenášeném signálu (takto je koncipováno SATA).
- Další úvahy:
  - **10 bitů – máme k dispozici 1024 kombinací.**
  - Pokud z 1024 kombinací vybereme takové, které obsahují stejný počet „0“ a „1“ (tzn. pět „0“ a pět „1“), pak získáme pouze 252 kombinací (tzn. mají nulovou klidovou úroveň napětí).
  - **Z těchto kombinací jsou pak vybrány takové, které mají požadovaný parametr RL – další redukce počtu kombinací.**

- Důsledek: musí se použít i takové kombinace, které obsahují čtyři „0“ a šest „1“ a naopak (nikoliv pouze pět „0“ a pět „1“).
- **S těmito kombinacemi však není dodržen požadavek nulové klidové úrovně přenášeného signálu – musí se řešit.**

- Řešení:

- V množině 10 bitových znaků je celkem 210 znaků, které obsahují šest „0“ a čtyři „1“.
- Tyto znaky mají i inverzní podobu, tzn. šest „1“ a čtyři „0“ (inverze).
- O těchto znacích pochopitelně platí, že klidová úroveň napětí není nulová (protože neobsahují stejný počet „0“ a „1“).
- Z toho důvodu je nutné počet vyslaných „0“ a „1“ průběžně sledovat a pokud není shodný, tak je nutno tento stav v průběhu komunikace kompenzovat.
- **Pojem disparity** - rozdíl mezi počtem vyslaných „0“ a „1“.

- Záporná disparita – počet vyslaných „0“ je menší než počet vyslaných „1“.
- **Průběžná disparita – aktuální disparita.**
- Pozn.: znak s pěti „0“ a pěti „1“ nemění průběžnou disparitu.
- Znak se šesti „1“ a čtyřmi „0“ (a naopak) průběžnou disparitu mění.
- Princip:
  - Každá slabika má přiděleny dvě hodnoty, pokud je průběžná disparita negativní, pak se použije jedna z 10 bitových hodnot, pokud je pozitivní, pak hodnota druhá.
- Příklad:
  - Průběžná disparita není vyvážená, pak kód pro zápornou průběžnou disparitu má šest „1“ a čtyři „0“, pro kladnou průběžnou disparitu hodnotu opačnou. Použije se jedna z těchto dvou hodnot podle hodnoty průběžné disparity.

- Příklad z tabulky:
  - Slabika 00h má přiděleny dvě 10 bitové hodnoty: 1001110100 a 0110001011, obě hodnoty mají stejný počet „0“ a „1“, takže nemění průběžnou disparitu.
  - Slabika 06h má rovněž přiděleny dvě 10 bitové hodnoty: 0110011011 nebo 0110010100, obě hodnoty mají různý počet „0“ a „1“, takže mění průběžnou disparitu.
- Výběrem hodnot z tabulky se zajistí, že zakódovaná posloupnost bitů je vyvážená a má zároveň požadovanou hodnotu RL.
- Zpětný proces (příjem dat na přijímací straně): napřed je nutno rozdělit přijatou informaci do 10 bitových segmentů, pak provést zpětné dekódování.

# Kódovací tabulka

Name	Byte	Output	
		Current rd-	Current rd+
D0.0	00h	100111 0100	011000 1011
D1.0	01h	011101 0100	100010 1011
D2.0	02h	101101 0100	010010 1011
D3.0	03h	110001 1011	110001 0100
D4.0	04h	110101 0100	001010 1011
D5.0	05h	101001 1011	101001 0100
D6.0	06h	011001 1011	011001 0100
D7.0	07h	111000 1011	000111 0100
D8.0	08h	111001 0100	000110 1011
D9.0	09h	100101 1011	100101 0100
D10.0	0Ah	010101 1011	010101 0100
D11.0	0Bh	110100 1011	110100 0100
D12.0	0Ch	001101 1011	001101 0100
D13.0	0Dh	101100 1011	101100 0100
D14.0	0Eh	011100 1011	011100 0100
D15.0	0Fh	010111 0100	101000 1011

- Pojmy: rd – running disparity – průběžná disparita, output – výstup

- Příklad: odeslání zprávy HELLO

ASCII znak	Kód	Počáteční disparita	10 bitová hodnota	Konečná disparita
H	48h	-	1110010101	+
E	45h	+	1010010101	+
L	4Ch	+	0011010101	+
L	4Ch	+	0011010101	+
O	4Fh	+	1010000101	-

Termíny: starting disparity – počáteční disparita, ending disparity – konečná disparita

- Zásady:
  - Slabika se šesti „1“ a čtyřmi „0“ nastaví zápornou disparitu.
  - Slabika se šesti „0“ a čtyřmi „1“ nastaví kladnou disparitu.
  - Slabika s pěti „1“ a pěti „0“ nemění disparitu.



- Zakódovaná zpráva má pak tuto podobu:
- 1110010101      1010010101      0011010101      0011010101  
1010000101
- **Výsledek: dvacet pět „0“, dvacet pět „1“.**
- Důsledek pro kódování 8b/10b:
  - Z 1024 možných kombinací jsou vybrány takové kombinace, které splňují oba diskutované požadavky tzn. požadované RL a nulovou klidovou úroveň napětí.
  - Takových kombinací je 252 (tzn. takové, které obsahují pět „0“ a pět „1“).

- Kontrola dat na přijímací straně:
  - Spočívá v kontrole průběžné disparity přijímaných dat. Podívejme se na příklad níže, kdy předpokládáme, že v přijímaných datech vznikla chyba (překlopením jednoho bitu do opačné hodnoty) – důsledek: místo „H“ bylo rozpoznáno „E“.

ASCII znak	8 bitová reprezentace	Průběžná disparita	Vyslaná 10 bitová hodnota	Přijatá 10 bitová hodnota	Průběžná disparita	Dekódovaná 8 bitová hodnota	ASCII znak
		-			-		
H	48h		1110010101	1010010101			E
		+			-		
E	45h		1010010101	1010010101			E
		+			-		
L	4Ch		0011010101	0011010101			L
		+			-		
L	4Ch		0011010101	0011010101			L
		+			-		
O	4Fh		1010000101	1010000101		CHYBA	
		-			-		

- Výsledek:
  - Chyba v jednom bitě způsobí vznik další chyby (nedodržená průběžná disparita), poslední znak „0“ by měl podle hodnoty průběžné disparity mít opačnou verzi své 10 bitové hodnoty (tzn. šest „1“ a čtyři „0“).
  - Je však přijat znak, který podpoří zápornou disparitu a to je vyhodnoceno jako chyba.
  - Celkem bylo přijato dvacet čtyři „1“ a dvacet šest „0“ (poslední přijatý znak podpořil zápornou disparitu místo, aby ji změnil na kladnou disparitu).

# Registry disku SATA

- Sériové ATA se z hlediska obsluhy diskové operace s využitím registrů jeví obdobně jako paralelní ATA (tzn. IDE disk se svým řadičem, viz registrový model disku). Sériové ATA však obsahuje další registry (reflektující specifičnost sériového rozhraní ATA).
- Pozn.: komunikace přes paralelní ATA byla stejná jako s jakýmkoliv jiným zařízením (přes registry), v případě sériového ATA je to zcela jiné a odlišné.
- Sériová komunikace – protokoly jsou složitější a musí proto být k dispozici pomůcky na „hlídání“ komunikace.
- Registry, které jsou v paralelním ATA (IDE), jsou označovány jako „taskfile registers“, registry, které jsou navíc v SATA jsou označovány jako „superset registers“ – tyto registry jsou definovány v „Serial ATA II Specification“ – zajišťují funkce, které v paralelním ATA nejsou k dispozici.
- Jejich funkcí je **sledování a řízení stavu rozhraní.**

- **SStatus Register** (Superset Status Register)

- Jeho stav reflektuje aktuální stav zařízení na rozhraní, takže při každém čtení jeho obsahu získáme odlišný obsah. Registr má tři pole:

- **Pole DET (4 bity)**

Registr, v němž je detekována přítomnost zařízení (různé úrovně přítomnosti). Cílem detekce je rozpoznat, zda na rozhraní je připojeno nějaké zařízení a zda s ním bylo již komunikováno. Rozpoznání patřičných stavů má za úkol fyzická vrstva.

Obsah = „0“ (0000):

Přítomnost zařízení nebyla doposud detekována .

Obsah = „1“ (0001):

Zařízení bylo detekováno, nebyla však zatím realizována žádná komunikace.

Obsah = „3“ (0011):

Zařízení bylo detekováno, byla realizována komunikace.

- **Pole SPD (4 bity)**

Rychlost, na níž se komunikuje.

- **Pole IPM (4 bity)**

Aktuální stav úsporného režimu (Power Management).

- **SError Register** (Superset Error Register)

Reflektuje informaci o chybách, které jsou specifické pro rozhraní SATA a sériovou komunikaci.

Jakmile chyba vznikne a nastaví se patřičný bit v registru, vynuluje se až zásahem z procesoru (na rozdíl od SStatus Register, kde se stav aktuálně mění). Díky tomu „neunikne“ pozornosti procesoru žádná vzniklá chyba.

**Mechanismus: rozhraní SATA nastavuje bity, procesor je testuje a softwarově nuluje.**

Registr sestává ze dvou polí: ERR a DIAG.

ERR – jednoznačně specifikované chyby, aby na ně systémový obslužný software mohl reagovat.

DIAG – registr využitelný pro diagnostické účely.

#### - **Pole ERR**

Dvě sady bitů

[Problémy, po nichž se rozhraní zotavilo:](#)

Jedna sada obsahuje informaci o chybách, z nichž se rozhraní SATA zotavilo bez vnějšího zásahu – slouží jako varování.

Jsou to poruchy komunikace a poruchy dat potenciálně vzniklé v důsledku vysoké rychlosti komunikace. Při následných komunikacích pak počítač může rychlost komunikace na základě této informace snížit.

Poruchy dat – nesouhlasilo CRC, na základě opakovaného pokusu se přenos dat podařil.

Porucha komunikace – rozpadlo se spojení.

### Problémy, po nichž se rozhraní nezotavilo:

To jsou problémy, které musí být odstraněny patřičnými obslužnými prostředky, protože se to nepodařilo jinými prostředky, tzn. prostředky, které má k dispozici.

Jsou to opět chyba v datech - bit T a chyby komunikace – bit C (např. nedodržení časových relací).

Jiný typ chyby: chyba protokolu – bit P.

Bit E – jiný typ poruchy zařízení, např. nějaký problém s hardware řadiče.



- **Pole DIAG**

Informace, kterou lze použít spolu s informací v poli ERR.

Důležité: obsah tohoto pole není definován specifikací (normou), může být specifický pro každou implementaci řadiče SATA.

Konkrétní implementace nemusí definovat celou množinu bitů.

- **SControl Register (Superset Control Register)**

Poskytuje počítači možnost řídit způsob provádění řídicích funkcí rozhraní SATA (nastavovat patřičné bity v řadiči SATA).

Způsob dojednávání rychlosti komunikace.

Způsob řízení úsporného režimu.

- **Shrnutí:**

- Disk SATA obsahuje stejné registry jako disk IDE (PATA) pro řízení diskové operace, kromě toho však obsahuje registry další, typické pro SATA.

- **Pojem eSATA**

- Je určený pro připojení externích disků. Vychází z interní varianty SATA (serial ATA), pouze vylepšuje konektor kvůli vyššímu namáhání. Bohužel nepřidává napájecí vodič, a tak připojené disky musí mít vlastní síťový adaptér.
- Rozhraní nabízí teoretickou přenosovou rychlost 3 Gbit/s. Reálná rychlost však záleží na pevném disku, neboť ten není takové rychlosti schopen dosáhnout. eSATA připojení se proto hodí pro připojení externích diskových polí, kde by USB či FireWire brzdily výkon.
- Omezení - délka kabelu. Zařízení může být od konektoru na základní desce vzdáleno maximálně 2 m, kabel mezi počítačem a diskem by tak neměl přesáhnout přibližně metr a půl.
- Další nevýhoda - malé rozšíření. Základních desek, potažmo počítačových sestav, s konektory eSATA je málo => malá přenositelnost k jiným počítačům.

- **Shrnutí:**

- Cíle SATA: dosažení požadovaného parametru RL a nulové klidové úrovně přenášeného signálu.
- Zdůvodnění parametru RL: na přijímací straně musíme být schopni generovat synchronizační signál s využitím fázového závěsu.
- Zdůvodnění nulové klidové úrovně přenášeného signálu: přenášíme nízká napětí. Jakýkoliv posun klidové úrovně by mohl způsobit chybné rozpoznání přenášeného signálu.
- Přenášené signály mají nízkou úroveň – příspěvek k rychlosti přenosu, budící elektronické prvky stíhají rychleji přepínat.
- Signály jsou přenášeny jako diferenciální signály.
- Realizace: sériový přenos, synchronizace generována na přijímací straně.
- Na tyto principy a požadavky se dívejme jako na obecně platné v moderních principech přenosu dat.

- **Závěr:**

- Byl vysvětlen princip kódování 8b/10b využívaný ve vysokorychlostních sériových přenosech.
- Byly vysvětleny techniky, které jsou u těchto principů využívány.
- Byly vysvětleny funkce registrů, které jsou využívány pro účely těchto přenosů.
- Stejná technika je využita v PCI Express.
- Obdobné principy existují v rozhraní DVI, kódování 8b/10b je však realizováno jiným způsobem. Součástí rozhraní je synchronizační signál.