

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra měření



Bakalářská práce

## **Využití sítí Ethernet v osobních automobilech**

*Dmitrij Bučkovský*

Vedoucí práce: doc. Ing. Jiří Novák, Ph.D.

Studijní program: Otevřená informatika, Bakalářský

Obor: Počítačové systémy

17. 5. 2016



**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

Student: **Dmitrij Bučkovský**

Studijní program: **Otevřená informatika**  
Obor: **Počítačové systémy**

Název tématu česky: **Využití sítí Ethernet v osobních automobilech**

Název tématu anglicky: **Ethernet Application in Passenger Cars**

**Pokyny pro vypracování:**

Seznamte se s technologií sítí Ethernet a se specifiky jejího využití v automobilovém průmyslu. Zpracujte rešerši zahrnující problematiku nasazení Ethernetu pro aplikace reálného času ve vozidlech. Zvýšenou pozornost věnujte algoritmům časové synchronizace, prioritizace rámců, funkcím síťové infrastruktury a funkčním principům implementace specifické varianty fyzické vrstvy BroadR Reach. Vytipujte vhodné komponenty pro sestavu demonstrátoru technologie a navrhnete jeho strukturu.

**Seznam odborné literatury:**

- [1] Kocourek, P., Novák, J.: Přenos informace. Skripta ČVUT FEL, Praha 2003
- [2] BroadR-Reach® Physical Layer Transceiver Specification For Automotive Applications, Broadcom Corporation 2014
- [3] Standardy IEEE802.3 a IEEE802.1

Vedoucí bakalářské práce: doc. Ing. Jiří Novák, Ph.D.

Datum zadání bakalářské práce: 15. prosince 2015

Platnost zadání do<sup>1</sup>: 30. září 2017

L. S.

Doc. Ing. Jan Holub, Ph.D.  
vedoucí katedryProf. Ing. Pavel Řípka, CSc.  
děkan

V Praze dne 15. 12. 2015

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.



## Poděkování

Chtěl bych poděkovat svému vedoucímu doc. Ing. Jiřímu Novákovi, Ph.D. za cenné rady, pomoc a čas, který mi věnoval při psaní této bakalářské práce.



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 17. 5. 2016

.....





# Abstract

There are many electronic devices in the cars as the car technology evolves. Because of that, the requirements for bandwidth and transmission rate grow up. The solution, how satisfy superior demands, could be usage of the Ethernet. This thesis is occupied with the BroadR-Reach standard, which should be used as physical layer, and with the other standards and protocols which could be usefull in automotive industry, e.g. VLAN, Time synchronization or Audio Video Bridging.

## Keywords

BroadR-Reach, automotive ethernet, VLAN, PTP, audio video bridging

# Abstrakt

S vývojem technologií přibývá ve vozidlech čím dál více elektronických zařízení. S tím roste i požadavek na větší šířku pásma a vyšší přenosovou rychlost. Řešením, jak uspokojit vyšší nároky, může být využití Ethernetu. Tato práce se zabývá standardem BroadR-Reach, který by měl tvořit fyzickou vrstvu sítě a dalšími standardy a protokoly, které by se mohly využít v automobilovém průmyslu, jako např. VLAN, časová synchronizace nebo Audio Video Bridging.

## Klíčová slova

BroadR-Reach, ethernet v automobilech, VLAN, PTP, audio video bridging



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Obsah práce . . . . .	1
1.2	Terminologie . . . . .	2
<b>2</b>	<b>Úvod do problematiky</b>	<b>3</b>
2.1	Přenosové médium . . . . .	3
2.2	Časová synchronizace . . . . .	3
2.3	Nízké zpoždění . . . . .	3
2.4	Alokace přenosového pásma sítě . . . . .	3
2.5	Spolehlivost . . . . .	4
<b>3</b>	<b>Fyzická vrstva BroadR-Reach</b>	<b>5</b>
3.1	Terminologie specifická pro fyzickou vrstvu BroadR-Reach . . . . .	5
3.2	BR-PHY Physical Coding Sublayer (BR-PCS) . . . . .	6
3.2.1	Funkce PCS Reset . . . . .	8
3.2.2	Povolení PCS Transmit . . . . .	8
3.2.2.1	Proměnné . . . . .	8
3.2.3	4B3B konverze . . . . .	8
3.2.3.1	4B3B konverze pro kontrolní signály . . . . .	8
3.2.3.2	4B3B konverze MII dat . . . . .	9
3.2.4	Funkce PCS Transmit . . . . .	11
3.2.4.1	Proměnné . . . . .	12
3.2.4.2	Funkce . . . . .	13
3.2.4.3	Časovač . . . . .	13
3.2.4.4	Zprávy . . . . .	13
3.2.5	PCS Transmit mapování symbolů . . . . .	13
3.2.5.1	Scrambler . . . . .	13
3.2.5.2	Generování $Sy_n[2:0]$ . . . . .	15
3.2.5.3	Generování $Sc_n[2:0]$ . . . . .	15
3.2.5.4	Generování scamblovacích bitů $Sd_n[2:0]$ . . . . .	15
3.2.5.5	Generování ternárního páru $(TA_n, TB_n)$ . . . . .	15
3.2.5.6	Generování $(TA_n, TB_n)$ když $TXMODE = SEND\_I$ . . . . .	16
3.2.5.7	Generování $(TA_n, TB_n)$ , když $TXMODE = SEND\_N$ a $tx\_enable = 1$ . . . . .	16

3.2.5.8	Generování ( $TA_n$ , $TB_n$ ) pro idle sekvenci když TXMODE = SEND_N . . . . .	16
3.2.5.9	Generování ( $TA_n$ , $TB_n$ ), když TXMODE = SEND_Z . . . . .	17
3.2.5.10	Generování sekvence symbolů . . . . .	17
3.2.6	Funkce PCS Receive . . . . .	18
3.2.6.1	Proměnné . . . . .	19
3.2.6.2	Funkce . . . . .	20
3.2.6.3	Časovač . . . . .	20
3.2.6.4	Zprávy . . . . .	20
3.2.6.5	Dekódování symbolů v PCS Receive . . . . .	20
3.2.6.6	PCS Receive descrambler . . . . .	20
3.2.6.7	PCS Receive automatická detekce polarity (volitelné) . . . . .	21
3.2.6.8	3B4B konverze MII signálů . . . . .	21
3.3	Podvrstva Physical Media Attachment (BR-PMA) . . . . .	21
3.3.1	Funkce PMA Reset . . . . .	21
3.3.2	Funkce PMA Transmit . . . . .	22
3.3.3	Funkce PMA Receive . . . . .	22
3.3.4	Funkce PHY Control . . . . .	24
3.3.5	Funkce Link Monitor . . . . .	25
3.3.6	Funkce PMA Clock Recovery . . . . .	25
3.3.7	Proměnné . . . . .	26
3.3.8	Časovače . . . . .	26
3.4	Elektrická specifikace podvrstvy PMA . . . . .	26
3.4.1	Imunita – DPI test . . . . .	26
3.4.2	Emise – 150Ohm řízený test emisí . . . . .	27
3.4.3	Frekvence hodin vysílače . . . . .	27
3.4.4	Testovací módy . . . . .	27
3.4.5	Testování . . . . .	28
3.4.6	Elektrické specifikace vysílače . . . . .	29
3.4.6.1	Pokles napětí (droop) výstupu vysílače . . . . .	29
3.4.6.2	Zkreslení vysílače . . . . .	30
3.4.6.3	Jitter . . . . .	30
3.4.6.4	Power Spectral Density vysílače . . . . .	30
3.4.6.5	Frekvence hodin vysílače . . . . .	31
3.4.7	Elektrické specifikace přijímače . . . . .	31
3.4.7.1	Diferenciální vstupní signál přijímače . . . . .	31
3.4.7.2	Frekvence hodin přijímače . . . . .	31
3.4.7.3	Omezení přeslechů . . . . .	31
3.5	Rozhraní pro správu . . . . .	32
3.5.1	MASTER-SLAVE konfigurace . . . . .	33
3.5.2	PHY – Inicializace . . . . .	33
3.5.3	Hodiny pro řídicí data - Management Data Clock (MDC) . . . . .	33
3.5.4	Vstup/výstup řídicích dat . . . . .	33
3.6	Charakteristiky fyzického segmentu . . . . .	33
3.6.1	Charakteristika kabeláže . . . . .	33
3.6.1.1	Charakteristická impedance . . . . .	34

3.6.1.2	Útlum na vedení . . . . .	34
3.6.1.3	Útlum odrazem . . . . .	34
3.6.1.4	Šum z okolního prostředí . . . . .	34
3.7	MDI specifikace . . . . .	35
3.7.1	MDI konektory . . . . .	35
3.7.2	Elektrické specifikace MDI . . . . .	35
3.7.2.1	MDI charakteristická impedance . . . . .	36
3.7.2.2	MDI Return loss . . . . .	36
3.8	Omezení zpoždění . . . . .	36
3.9	Testování na úrovni systému . . . . .	36
3.9.1	Interní loopback funkce . . . . .	36
3.9.2	Externí loopback funkce . . . . .	36
<b>4</b>	<b>Časová synchronizace a Precision Time Protocol (PTP)</b>	<b>39</b>
4.1	Precision Time Protocol . . . . .	39
4.2	PTP zprávy . . . . .	39
4.3	Typy PTP zařízení . . . . .	40
4.3.1	Ordinary clock (OC) . . . . .	40
4.3.2	Boundary clock (BC) . . . . .	41
4.3.3	Transparent clock (TC) . . . . .	41
4.3.4	End-to-end transparent clock . . . . .	41
4.3.5	Peer-to-peer transparent clock . . . . .	41
4.3.6	Management node . . . . .	42
4.4	Master-slave hierarchie . . . . .	42
4.4.1	Best Master Clock Algoritmus (BMCA) . . . . .	42
4.5	Synchronizace hodin . . . . .	43
4.6	Generování časových značek . . . . .	44
<b>5</b>	<b>Virtual Local Area Network (VLAN)</b>	<b>47</b>
5.1	Zařazení portu do VLAN . . . . .	47
5.1.1	Členství podle portů . . . . .	47
5.1.2	Členství podle MAC adresy . . . . .	48
5.1.3	Členství podle protokolu . . . . .	49
5.2	Komunikace v rámci VLAN . . . . .	49
5.2.1	IEEE 802.1Q tagging . . . . .	49
5.3	Směrování mezi VLAN . . . . .	50
5.4	Využití VLAN v automobilovém průmyslu . . . . .	50
<b>6</b>	<b>Quality of Service (QoS)</b>	<b>51</b>
6.1	Class of Service (CoS) . . . . .	51
6.2	Třídění rámců do front . . . . .	52
<b>7</b>	<b>Audio Video Bridging</b>	<b>53</b>
7.1	IEEE 802.1Qat: Stream Reservation Protocol (SRP) . . . . .	53
7.2	IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)	55
7.2.1	Detekce SRP domény . . . . .	56

7.2.2	Parametry popisující dostupnost pásma . . . . .	56
7.2.3	Vztahy mezi parametry . . . . .	56
7.2.4	Správa parametrů . . . . .	57
7.2.5	Závislost aktuální šířky pásma na velikosti MAC service data unit (MSDU) . . . . .	57
7.2.6	Mapování priorit k třídám provozů . . . . .	58
7.2.7	Talker . . . . .	59
7.2.8	Listener . . . . .	60
7.3	Credit-based shaper algoritmus . . . . .	60
<b>8</b>	<b>Návrh struktury sítě</b>	<b>63</b>
8.1	Nasazení Ethernetu . . . . .	63
8.2	Vysílače/přijímače a přepínače . . . . .	65
<b>9</b>	<b>Závěr</b>	<b>67</b>
<b>A</b>	<b>Obsah příloženého CD</b>	<b>71</b>

# Seznam obrázků

3.1	Funkční blokový diagram BR-PHY (převzato z [1]) . . . . .	7
3.2	Povolení přenosu dat (převzato z [1]) . . . . .	9
3.3	4B3B konverze MII signálu (3n bitů dat, žádný stuff bit) (převzato z [1]) . . .	10
3.4	4B3B konverze MII signálu ((3n+1) bitů dat, 2 stuff bity) (převzato z [1]) . .	10
3.5	4B3B konverze MII signálu ((3n+2) bitů dat, 1 stuff bit) (převzato z [1]) . . .	11
3.6	Stavový diagram funkce PCS Transmit (STD = symb_timer_done, viz sekce 3.2.4.3) (převzato z [1]) . . . . .	12
3.7	Blokový diagram mapování symbolů (převzato z [1]) . . . . .	14
3.8	Scrambler realizovaný posuvným registrem (převzato z [7]) . . . . .	14
3.9	Konverze symbolů z 2-D do 1-D (převzato z [1]) . . . . .	17
3.10	Stavový diagram funkce PCS Receive (převzato z [1]) . . . . .	18
3.11	Stavový diagram funkce JAB (převzato z [1]) . . . . .	19
3.12	3B4B konverze MII signálů (převzato z [1]) . . . . .	22
3.13	BR-PMA diagram (převzato z [1]) . . . . .	23
3.14	Stavový diagram PHY Control (převzato z [1]) . . . . .	24
3.15	Stavový diagram Link Monitoru (převzato z [1]) . . . . .	25
3.16	Testovací obvod 1: Droop, Jitter (převzato z [1]) . . . . .	28
3.17	Testovací obvod 2: Zkreslení (převzato z [1]) . . . . .	29
3.18	Testovací obvod 2: Testování PSD (převzato z [1]) . . . . .	29
3.19	Výstup testovacího módu 1, měřítko není ve správném poměru (převzato z [1])	30
3.20	Měření jitteru v normálním módu (převzato z [1]) . . . . .	31
3.21	PSD horní a spodní limity (převzato z [1]) . . . . .	32
3.22	Obvod pro testování přeslechů (převzato z [1]) . . . . .	32
3.23	BR fyzický segment (převzato z [1]) . . . . .	34
3.24	Interní loopback (převzato z [1]) . . . . .	37
3.25	Externí loopback (převzato z [1]) . . . . .	37
4.1	Přičtení času pobytu v hodinách (převzato z [3]) . . . . .	41
4.2	Přičtení času pobytu v hodinách a zpoždění na lince (převzato z [3]) . . . . .	42
4.3	Přenos zpráv pro synchronizaci master a slave hodin (převzato z [3]) . . . . .	44
4.4	Generování časových značek . . . . .	45
5.1	Rozdělení sítě do dvou VLAN . . . . .	48
5.2	Originální Ethernet rámeček a označený rámeček hlavičkou 802.1Q . . . . .	49
7.1	Pole obsažena ve zprávách (převzato z [5]) . . . . .	55

7.2	Model front pro Talker stanici (převzato z [6]) . . . . .	59
7.3	Credit-based shaper algoritmus (převzato z [6]) . . . . .	61
8.1	Struktura sítě v dnešních vozidlech . . . . .	64
8.2	Struktura sítě s použitím Ethernetu a přepínačů . . . . .	64
8.3	Návrh struktury sítě s několika koncovými zařízeními s integrovanými čipy podporující technologii BroadR-Reach . . . . .	66



# Seznam tabulek

3.1	Mapování idle symbolů v trénovacím módu (převzato z [1]) . . . . .	16
3.2	Mapování symbolů v módu SEND_N (převzato z [1]) . . . . .	16
3.3	Idle symboly v módu SEND_N (převzato z [1]) . . . . .	17
3.4	Registr pro testovací módy (převzato z [1]) . . . . .	27
3.5	Mapování symbolu v testovacím módu 4 (převzato z [1]) . . . . .	28
3.6	PSD maska (převzato z [1]) . . . . .	31
3.7	Maximální útlum pro dané frekvence (převzato z [1]) . . . . .	34
3.8	Útlum odrazem (převzato z [1]) . . . . .	35
3.9	MDI return loss (převzato z [1]) . . . . .	36
6.1	Úrovně priority podle 802.1Q-2005 (převzato z [4]) . . . . .	51
6.2	Doporučené mapování priorit a tříd provozů (převzato z [4]) . . . . .	52
7.1	Doporučené mapování pro SR třídy A (priorita 3) a B (priorita 2) (převzato z [6]) . . . . .	58
7.2	Doporučené mapování jen pro SR třídu B (priorita 2) (převzato z [6]) . . . . .	59



# Kapitola 1

## Úvod

V dnešní době se automobily stávají čím dál komplikovanějšími. Kladou se větší nároky na bezpečnost posádky i okolí, na pohodlí řidiče i spolucestujících, ale i na výkon automobilu. Do automobilu se přidávají kamery, aby řidič mohl kontrolovat všechny čtyři strany, senzory pro detekování překážky, aby auto mohlo řidiče včas varovat popř. samo se překážce vyhnout, na zadních sedačkách jsou multimediální přehrávače videa a audia, některé automobily mají přístup k LTE síti a přístupu na Internet. To vše vede k přidávání další elektroniky, roste hardware i software, komplikuje se propojení mezi zařízeními a roste kabeláž, zvyšuje se velikost dat, které se přenášejí. Současné standardy, které se běžně v automobilech používají, jako CAN, LIN, FlexRay, MOST a LVDS jsou pomalé nebo drahé. Proto se výrobci automobilů rozhodli pro použití Ethernetu v automobilech.

Běžná fyzická vrstva Ethernetu se ale nemohla v automobilech používat, protože nesplňovala elektromagnetickou kompatibilitu (EMC), a proto vznikla nezisková organizace OPEN Alliance, kterou založily firmy NXP, Broadcom a BMW v listopadu 2011 (nyní má, podle oficiálních stránek<sup>1</sup>, přes 300 členů). S podporou aliance vyvinula firma Broadcom technologii BroadR-Reach (dále jen BR) speciálně pro automobilový průmysl. BR je standard Ethernetové fyzické vrstvy, jedná se o jeden pár kroucené dvojlinky s přenosovou rychlostí 100 Mb/s v obou směrech (plný duplex).

### 1.1 Obsah práce

Tato práce se zabývá první a druhou vrstvou OSI modelu, protože převážně tam se liší implementace Ethernetu v automobilech než v jiných prostředích. Hlavní částí je popis standardu BroadR-Reach, který používá některé technologie 100BASE-TX a hlavně 1000BASE-T, ale liší se rozhraní mezi MII a MDI, včetně fyzického media. Dále jsou popsány standardy a protokoly důležité pro aplikace reálného času nebo bezpečnosti, jako např. časová synchronizace, nebo funkce VLAN, pro oddělení automobilových domén, prioritizaci rámců a spolu s Audio Video Bridging k rezervování kanálů pro spolehlivý přenos videa, audia a kritických dat. Na závěr jsem se podíval na produkty, které firmy jako Broadcom, NXP apod. vyrábějí nebo vyvíjejí.

---

<sup>1</sup><http://www.opensig.org/>

## 1.2 Terminologie

I přesto, že je tato práce psaná v českém jazyce, tak jsou zde použita i anglická slova. Jedná se o termíny, které se běžně v praxi používají a buď v češtině nemají ekvivalent, nebo se nepoužívají. Přesto jsou tyto termíny vysvětleny.

## Kapitola 2

# Úvod do problematiky

V dnešní době jsou některé modely automobilů, kde se používá Ethernet, není ale nasazen v kritických oblastech, používá se např. k přenosu medií v zábavní doméně automobilu. K rozšíření Ethernetu do dalších domén a nahrazení některých standardů je potřeba zajistit základní požadavky pro nasazení Ethernetu v časově závislých aplikacích a v prostředí s důrazem na bezpečnost.

### 2.1 Přenosové médium

Klasického Ethernetu v automobilu není možné použít. Fyzická vrstva nesplňuje EMC požadavky a kromě toho klasický kabel se čtyřmi páry kroucené dvojlinky zabírá hodně místa, je těžký a drahý. Standard BroadR-Reach používá jen jeden pár kroucené dvojlinky a díky snížení šířky pásma splňuje EMC požadavky.

### 2.2 Časová synchronizace

V automobilu máme spoustu senzorů a ke správnému vyhodnocení situace potřebujeme čas pořízení dat. Také můžeme potřebovat souběžné odebírání nebo dodání vzorků ze senzorů, např. kamery by měly být synchronizovány a zobrazovat obrazy ze stejných časů.

### 2.3 Nízké zpoždění

V automobilu je důležitá především bezpečnost, proto řídicí počítač musí obdržet data ze senzorů co nejrychleji, v řádech stovek mikrosekund až jednotek milisekund. Je třeba také zajistit aby kritické pakety měly na přepínačích přednost před obyčejnými pakety a nemusely zbytečně čekat, než se odešlou méně důležitá data.

### 2.4 Alokace přenosového pásma sítě

Pro kvalitní přenos audia nebo videa si potřebujeme alokovat část přenosového pásma a vytvořit kanál, přes který nebudou procházet jiná data a tím pádem přenos nebude ničím

brzděn. Tyto kanály by mohly být využity i pro přenos kritických dat, která by ale měla mít přednost před audio/video pakety.

## 2.5 Spolehlivost

Jelikož spousta systémů v automobilu jsou časově závislá, je důležité, aby komunikační linka byla spolehlivá, dokázala kontrolovat a zaručit maximální zpoždění a aby nedocházelo k hromadění paketů na přepínačích.

## Kapitola 3

# Fyzická vrstva BroadR-Reach

BroadR-Reach je ethernetová fyzická vrstva vyvinutá společností Broadcom s podporou OPEN Alliance (One Pair Ethernet), umožňující plně duplexní (full-duplex) komunikaci přes jeden pár kroucené dvojlinky s přenosovou rychlostí 100Mb/s. BroadR-Reach byl vyvinut pro použití v automobilovém průmyslu, kde by měl nahradit (aspoň z části) stávající standardy a topologii a měl by snížit celkovou váhu a cenu kabeláže. BroadR-Reach využívá některé výhody Fast Ethernetu (100 Mb/s) a hlavně 1000Base-T Ethernetu, např:

- Full-duplexní přenos, přes jeden pár nestíněné kroucené dvojlinky.
- Echo cancellation – BroadR-Reach umí rozpoznat a filtrovat odražený signál (jelikož medium, které vysílá zároveň i přijímá).
- Použití PAM3 modulace a snížení šířky pásma na 33,3 MHz (Fast Ethernet nebo Gigabit Ethernet má 65-80 MHz), díky čemuž BroadR-Reach splňuje přísné EMC požadavky.

### 3.1 Terminologie specifická pro fyzickou vrstvu BroadR-Reach

V této kapitole jsem čerpal z [1].

- PHY – Fyzické vrstva zařízení.
- BR-PHY – Fyzická vrstva BroadR-Reach.
- 4B3B – 4B3B je v BR-PHY technika kódování, kdy se data (4 bity) z Media-independent interface (MII) s frekvencí 25 MHz konvertují na 3 bity, které se vysílají s frekvencí 33,3 MHz (viz 3.2.3).
- SSD, ESD – Start-of-Stream delimiter (SSD), End-of-Stream delimiter (ESD), sekvence symbolů označující začátek/konec přenosu sekvenci dat.
- PAM3 – Pulzně amplitudová modulace 3. úrovně (tj. 3 diskrétní hodnoty [-1, 0, 1]). Díky PAM3 je snížena šířka pásma na 33,3 MHz, díky čemuž BR splňuje elektromagnetickou kompatibilitu pro automobilový průmysl.

- 1D-PAM3 – Je metoda kódování symbolu, kdy se 1D ternární symbol<sup>1</sup> z BR-PCS vysílače vysílá pomocí tří napěťových úrovní (PAM3). Během každé periody se vysílá jeden symbol.
- Inicializace PHY – V automobilovém průmyslu potřebujeme rychlé nastavení spojení mezi zřízenými, proto se zde nepoužívá funkce auto-negotiation, jako v jiných standardech. V technologii BR jsou implementovány primitivní funkce pro určení kdo je MASTER a kdo SLAVE. Start-up proces, od link\_control=ENABLE po validní data, musí být dokončen do 200 ms pro BR 15 m segment.
- Scrambling – Scrambling je metoda používaná BR-PCS k promíchání, náhodně rozdělení sekvencí vysílaných symbolů, aby se omezil výskyt spektrálních čar, vznikajících řetězením stejných znaků. Je důležitá synchronizace mezi scramblerem a descramblerem zapojených PHY. Viz 3.2.5.1.
- Idle – Idle je sekvence znaků, která je vysílána např. mezi datovými rámci, nebo pokud je to vyžadováno a slouží k synchronizaci hodin mezi vysílačem a příjemcem.
- Physical Coding Sublayer (PCS) – Physical Coding Sublayer je podvrstva fyzické vrstvy, která vysílá/přijímá signály do/z MII a signály do/z BR-PMA (viz 3.2).
- Podvrstva Physical Medium Attachment (PMA) – Physical Medium Attachment je podvrstva fyzické vrstvy, která vysílá/přijímá signály do/z PCS do jednoho páru kroucené dvojlinky a podporuje správu linky a funkce po řízení BR-PHY. PMA zprostředkovává full-duplexní komunikaci s rychlostí přenosu 100Mb/s (viz 3.3)
- Signalizace – Signalizaci vykonává PCS, která generuje sekvenci symbolů, a PMA je následně vysílá přes kroucenou dvojlinku. Cíle signalizace:
  - Mapování nibble (4 bity) dat na ternární symboly.
  - Nevytváření vzájemných vztahů mezi symboly putující v obou směrech.
  - Schopnost rozpoznat jestli jsou sekvence symbolů data nebo idle.
  - Oddělování dat pomocí SSD, ESD aj.
  - Signalizování statusu lokálního přijímače k vzdálenému PHY, aby se dalo najevo, že přijímač nemůže spolehlivě operovat a potřebuje přenastavit.

## 3.2 BR-PHY Physical Coding Sublayer (BR-PCS)

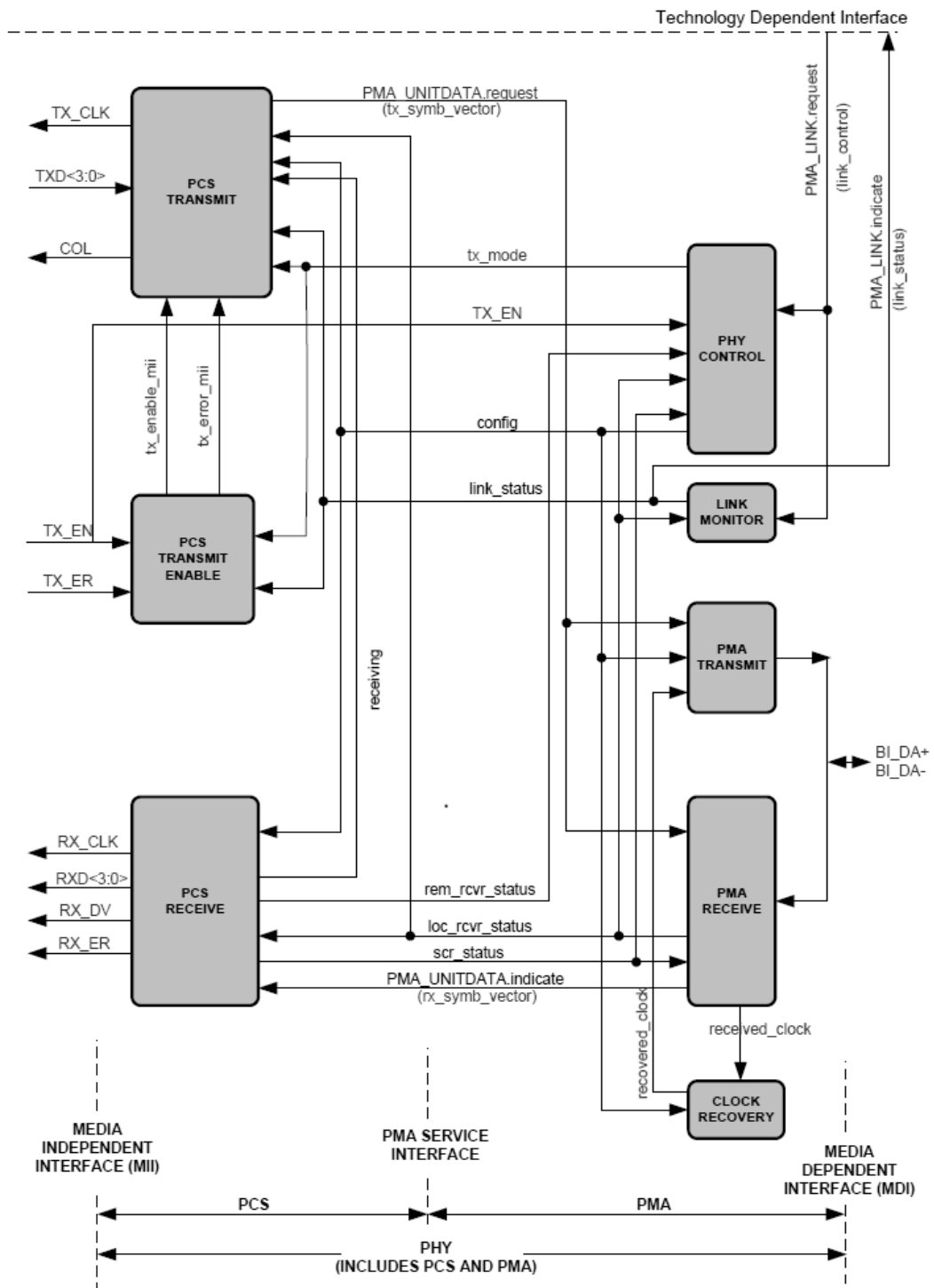
V této kapitole a podkapitolách jsem čerpal z [1].

BR-PCS přijímá 4 bity z rozhraní s MII s frekvencí 25 MHz, a převádí je pomocí 4B3B konverze na 3 bity s frekvencí hodin 33,3 MHz. Bity jsou scramblerem promíchány a poté zakódovány do ternárních symbolů. Ternární symboly se seskupí do párů, tyto páry se serializují a přenášejí s frekvencí hodin 66,6 MHz. Na obrázku 3.1 (levá polovina) jsou znázorněny funkce PCS Transmit a PCS Receive.

---

<sup>1</sup>Ternární symbol nabývá tří hodnot, v tomto případě -1, 0 a 1.





Obrázek 3.1: Funkční blokový diagram BR-PHY (převzato z [1])

### 3.2.1 Funkce PCS Reset

PCS Reset inicializuje všechny PCS funkce. Funkce PCS Reset by se měla vykonat po každé, když nastane jedna z následujících událostí:

- zapnutí
- požadavek na resetování

Pokud nastane jedna z těchto událostí, PCS Reset nastaví `pcs_reset=ON`. Obrázek 3.1 nezobrazuje PCS Reset funkci.

### 3.2.2 Povolení PCS Transmit

Na obrázku 3.2, PCS Data Transmission Enabling proces generuje signály `tx_enable_mii` a `tx_error_mii`, které přeposílají hodnotu signálů `TX_EN` a `TX_ER` z rozhraní MII pokud `tx_mode=SEND_N`. Pokud `tx_mode!=SEND_N` jsou signály nastaveny na `FALSE`.

#### 3.2.2.1 Proměnné

- `tx_enable_mii` – Je generováno pomocí stavového diagramu z obrázku 3.2.
- `tx_error_mii` – Je generováno pomocí stavového diagramu z obrázku 3.2.
- `link_status` – PMA Link Monitor nastavuje hodnotu `OK` nebo `FAIL` a posílá do PCS přes `PMA_LINK.indication`.
- `pcs_reset` – Funkce PCS Reset nastavuje hodnotu `ON` nebo `OFF`.
- `TX_EN` – Signál z rozhraní MII.
- `TX_ER` – Signál z rozhraní MII.
- `tx_mode` – Je parametr, který nastavuje funkce PMA PHY Control a posílá do PCS přes `PMA_TXMODE.indication`. Hodnoty jsou `SEND_Z`, `SEND_N` nebo `SEND_I`.

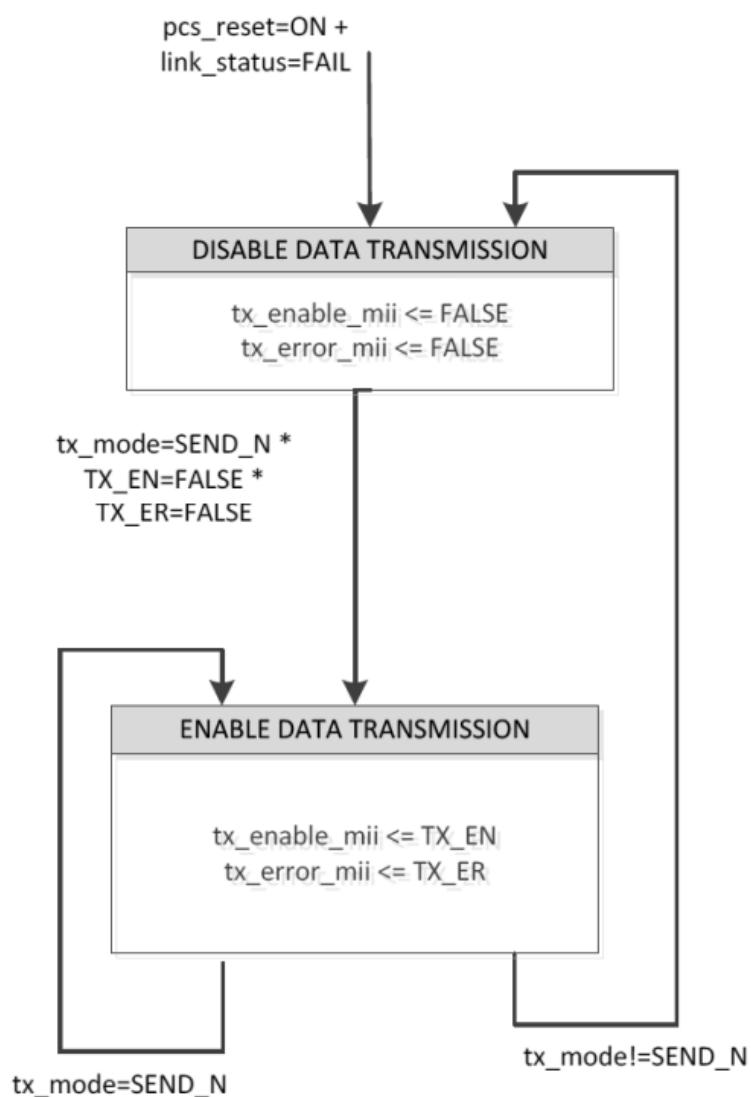
### 3.2.3 4B3B konverze

V této kapitole a podkapitolách jsem čerpal z [1].

#### 3.2.3.1 4B3B konverze pro kontrolní signály

Vstupem 4B3B funkce jsou signály `tx_enable_mii`, `tx_error_mii` a `TXD[3:0]`, synchronizované s MII hodinami `TX_CLK`. Po 4B3B konverzi by měli být vysílající signály `tx_data[2:0]`, `tx_enable` a `tx_error` synchronizované s hodinami PCS vysílače `pcs_txclk`.

Pro rovnoměrnou bitovou propustnost musí být frekvence hodin `TX_CLK` 25 MHz a frekvence hodin `pcs_txclk` 33,3 MHz. Zdrojem `TX_CLK` může být lokální krystal nebo oscilátor,



Obrázek 3.2: Povolení přenosu dat (převzato z [1])

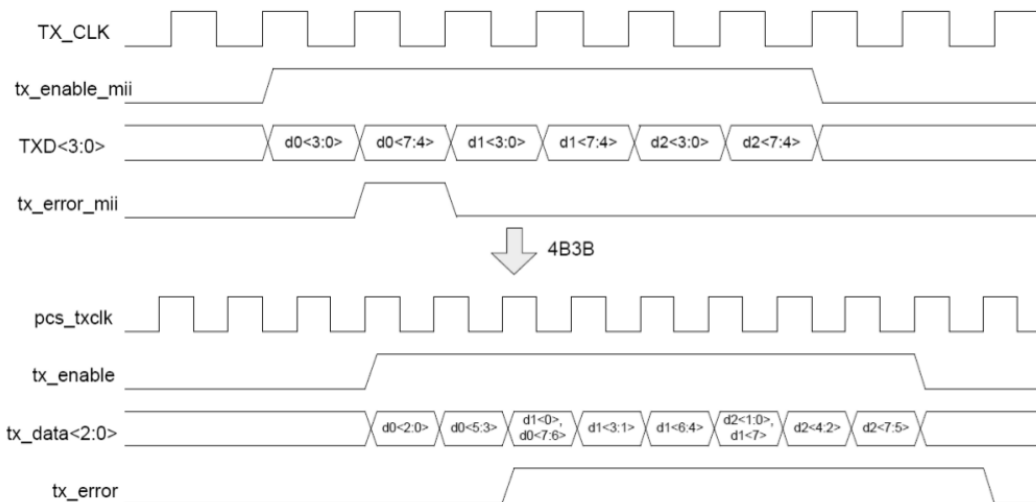
pokud je v MASTER módu nebo z tzv. recovered clock<sup>2</sup>, pokud je ve SLAVE módu. Hodiny pcs\_txclk mohou být přivedeny ze stejného zdroje jako TX\_CLK, avšak se správně upravenou požadovanou frekvencí.

### 3.2.3.2 4B3B konverze MII dat

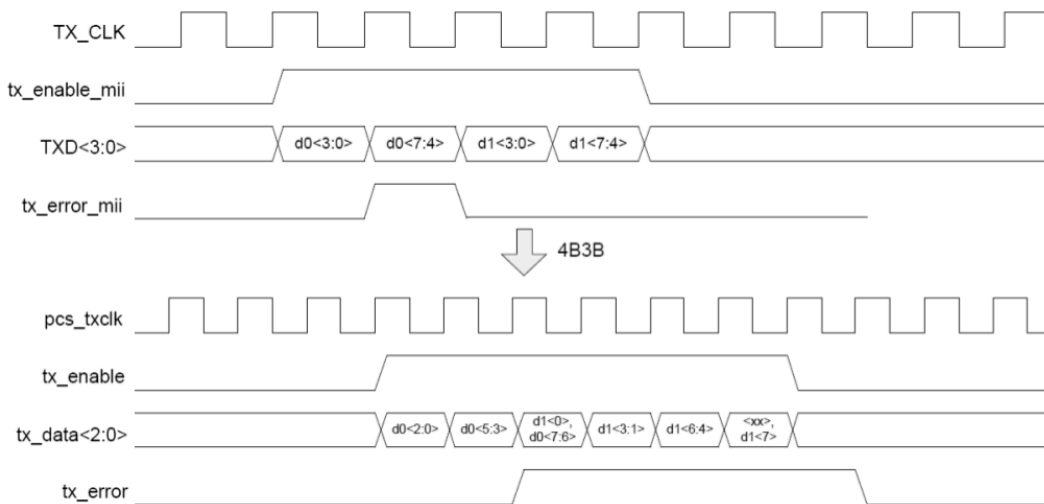
Data TXD[3:0] z rozhraní MII jsou nejdříve převedeny na skupiny po 3 bitech tx\_data[2:0]. Jak je vidět na obrázku 3.3, 3.4 a 3.5, pokud počet bitů není násobek tří, pak funkce 4B3B doplní do poslední skupiny podle potřeby 1 nebo 2 bity (tzv. stuff bity). Signál tx\_enable

<sup>2</sup>Clock recovery je metoda jak ze sekvence příchozích znaků získat hodiny, na kterých běží vysílací zařízení.

zůstává TRUE po celou dobu převodů do skupin po 3 bitech. Stuff bity jsou zvoleny náhodně, závisí na výrobci. Na přijímající straně jsou tyto bity zahozeny. Aby přijímající strana stíhala zahodit stuff bity a nedocházelo k přetečení FIFO zásobníku, je mezi vysílanými daty nastavena pauza 96 bit time IPG (interpacket gap<sup>3</sup>).

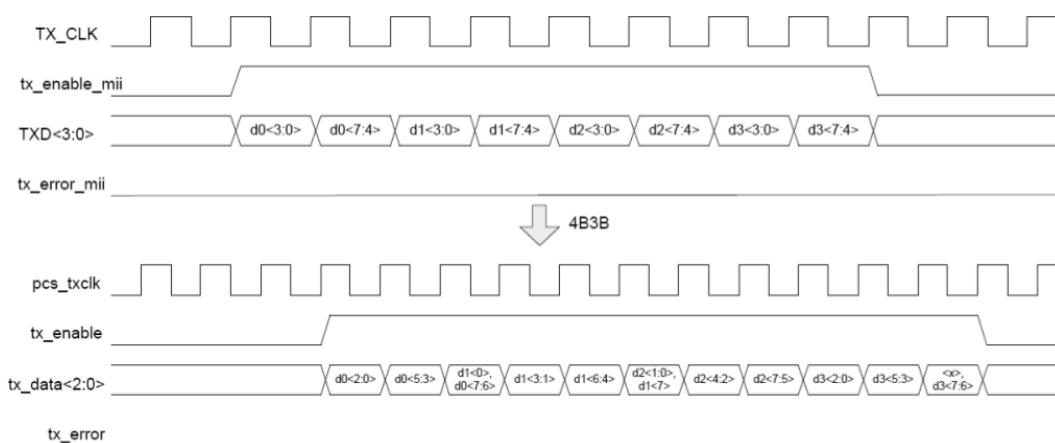


Obrázek 3.3: 4B3B konverze MII signálu (3n bitů dat, žádný stuff bit) (převzato z [1])



Obrázek 3.4: 4B3B konverze MII signálu ((3n+1) bitů dat, 2 stuff bity) (převzato z [1])

<sup>3</sup>IPG – výplň, pauza mezi rámci



Obrázek 3.5: 4B3B konverze MII signálu ((3n+2) bitů dat, 1 stuff bit) (převzato z [1])

### 3.2.4 Funkce PCS Transmit

V této kapitole a podkapitolách jsem čerpal z [1].

Funkce PCS Transmit by měla splňovat stavový diagram na obrázku 3.6, spolu se stavovými proměnnými, funkcemi, časovači a zprávami.

Každou periodu PCS Transmit generuje sekvenci symbolů  $\mathbf{A}_n$  a posílá jí do PMA v podobě jednoho ze tří módů (TXMODE). Symbol  $\mathbf{A}_n$  je ternární kód, který nabývá hodnot (-1, 0, 1). PMA vysílá symbol  $\mathbf{A}_n$  přes kroucenou dvojlinku (BI\_DA). Číslo  $n$  je časový index rozlišující jednotlivé periody vysílání symbolů. Jmenovitá hodnota této periody je 15ns.

Po přivedení signálu na tx\_enable PCS Transmit vyšle 6 po sobě jdoucích symbolů (SSD) do PMA, které nahrazují prvních 9 bitů preamble. Po SSD následují data tx\_data[2:0], která jsou kódovaná pomocí PAM3 techniky do vektoru ternárních symbolů, dokud nemá tx\_enable hodnotu false. Nakonec se pošle speciální kód ESD (nebo ERR\_ESD, pokud se vyskytla chyba při přenosu) šesti po sobě jdoucích symbolů, které naznačují konec přenosu a návrat do idle módu.

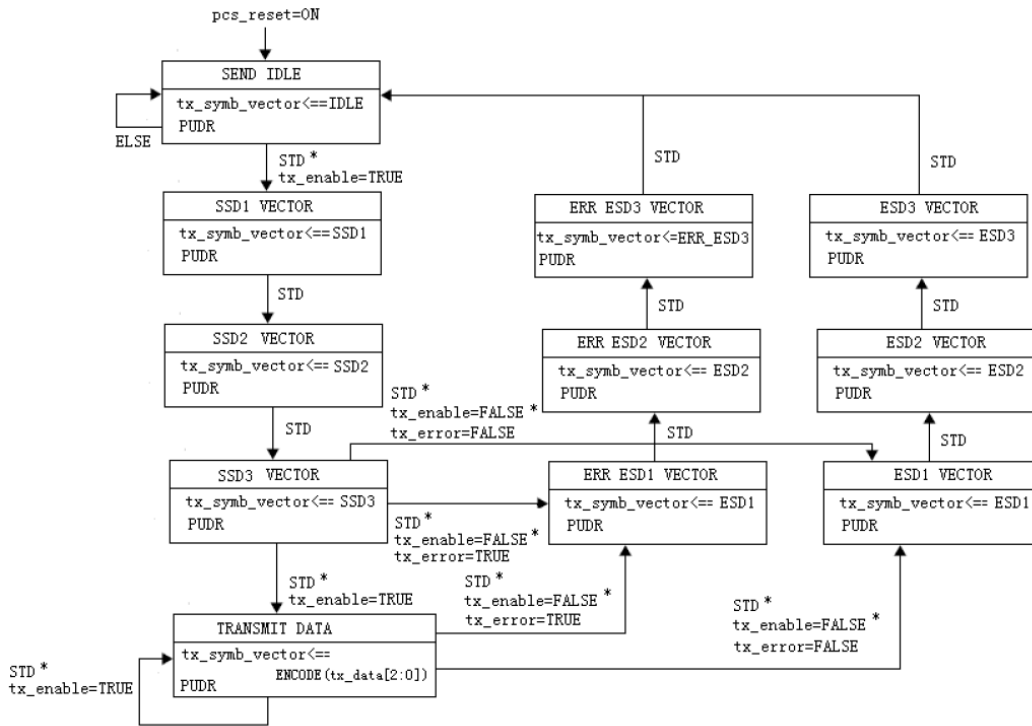
Pokud je přiveden signál do tx\_error\_mii (přes MII TX\_ER) během přenosu dat, tak je tx\_error nastaven na hodnotu TRUE a zůstává v TRUE do konce přenosu, aby zaznamenal událost, viz obrázek 3.3 a 3.4.

BroadR-Reach má jenom jeden speciální pár symbolů (0, 0), které nejsou používány jako idle nebo data symboly. Proto místo toho aby TX\_ER posílal ERROR symboly (jako v 100BASE-TX nebo 1000BASE-T), BroadR-Reach, na konci datového paketu posílá symboly ESD3 (1, 1) nebo ERR\_ESD3 (-1, -1), podle kterých se určí jestli nastala chyba (tx\_error). Symboly ESD3 nebo ERR\_ESD3 následují bezprostředně po dvou speciálních párech ESD1 (0, 0) a ESD2 (0, 0), viz obrázek 3.6.

Pokud má TXMODE hodnotu SEND\_Z, pak PCS Transmit posílá do PMA vektor nul. Pokud má TXMODE hodnotu SEND\_I, pak PCS Transmit generuje sekvenci symbolů podle kódovacích pravidel v trénovacím módu (training mode). Pokud má TXMODE hodnotu

SEND\_N, pak PCS Transmit generuje každou periodu symbol  $\mathbf{A}_n$ , který reprezentuje data, speciální řídicí symboly jako SSD/ESD nebo IDLE, které jsou definovány níže. Přechod z idle módu do data módu (tj. začátek vysílání) je signalizováno vložení SSD a konec vysílání je signalizováno pomocí ESD.

V training módu, TXMODE=SEND\_I, vysílané symboly jsou v přijímači použity k nastavení časové synchronizace a spojení. Když je spojení nastaveno, PHY se přepne do SEND\_N módu a vysílané PAM3 symboly jsou na přijímací straně (PHY) použity k dalšímu sledování frekvence/fáze hodin.



Obrázek 3.6: Stavový diagram funkce PCS Transmit (STD = symb\_timer\_done, viz sekce 3.2.4.3) (převzato z [1])

### 3.2.4.1 Proměnné

- ESD1 – První vektor ze sekvence ESD, skládá se ze dvou ternárních symbolů (0,0).
- ESD2 – Druhý vektor ze sekvence ESD, skládá se ze dvou ternárních symbolů (0,0).
- ESD3 – Třetí vektor ze sekvence ESD, skládá se ze dvou ternárních symbolů (0,0).
- ERR\_ESD3 – Třetí vektor ze sekvence ESD v případě, indikuje chybu, skládá se ze dvou ternárních symbolů (-1,-1).
- SSD1 – První vektor ze sekvence SSD, skládá se ze dvou ternárních symbolů (0,0).

- SSD2 – Druhý vektor ze sekvence SSD, skládá se ze dvou ternárních symbolů (0,0).
- SSD3 – Třetí vektor ze sekvence SSD, skládá se ze dvou ternárních symbolů (0,0).
- tx\_data[2:0] – Data generována funkcí PCS Transmit, synchronizována s hodinami pcs\_txclk (33,3 MHz).
- tx\_enable – Generován funkcí PCS Transmit, viz obrázek 3.2.
- tx\_error – Generován funkcí PCS Transmit, viz obrázek 3.2.
- tx\_symb\_vector – Vektor ternárních symbolů, který generuje funkce PCS Transmit a posílá do PMA přes PMA\_UNITDATA.request, hodnota je SYMB\_1D.

#### 3.2.4.2 Funkce

- ENCODE – Během PCS Transmit procesu tato funkce kóduje přijatá data z MII tx\_data[2:0] z 3-bitů do 2-D ternárních symbolů a poté překládá 2-D symboly na 1-D a vrací tx\_symb\_vector.

#### 3.2.4.3 Časovač

- symb\_timer – Po vypršení časovače, podmínka symb\_timer\_done (STD) se nastaví na TRUE. Časovač je nastaven na 30ns a je synchronizovaný s hodinami pcs\_txclk.

#### 3.2.4.4 Zprávy

- PUDR – Zkratka pro PMA\_UNITDATA.request(tx\_symb\_vector).
- STD – Zkratka pro symb\_timer\_done.

### 3.2.5 PCS Transmit mapování symbolů

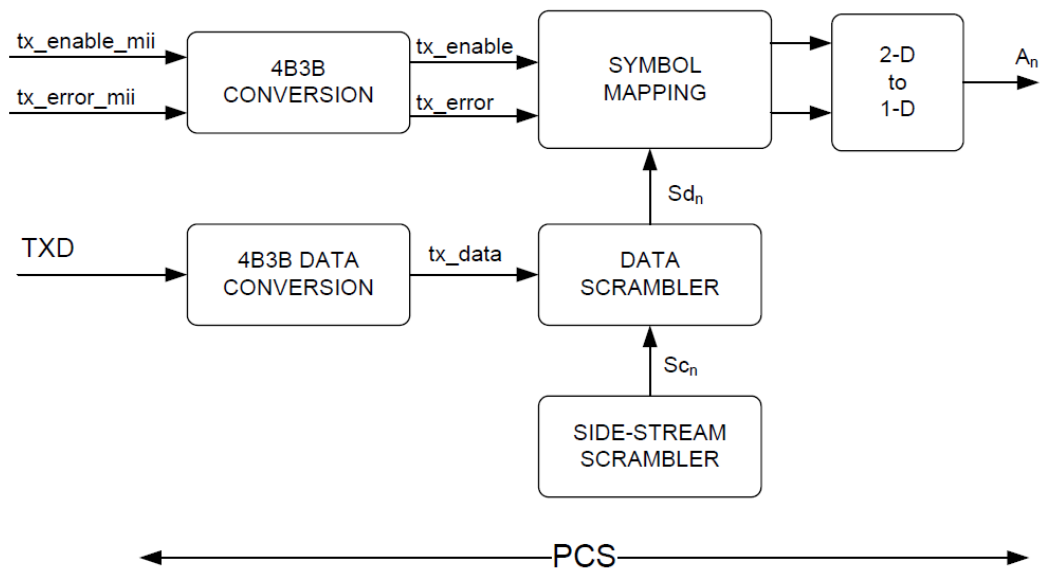
V této kapitole a podkapitolách jsem čerpal z [1] a [7].

#### 3.2.5.1 Scrambler

Scrambling je metoda používaná BR-PCS k promíchání bitů. Tím se omezí výskyt spektrálních čar, vznikající řetězením stejných znaků. Pokud je BR-PHY v MASTER módu, pak scrambler používá k promíchání dat polynom:

$$g_M(x) = 1 + x^{13} + x^{33}$$

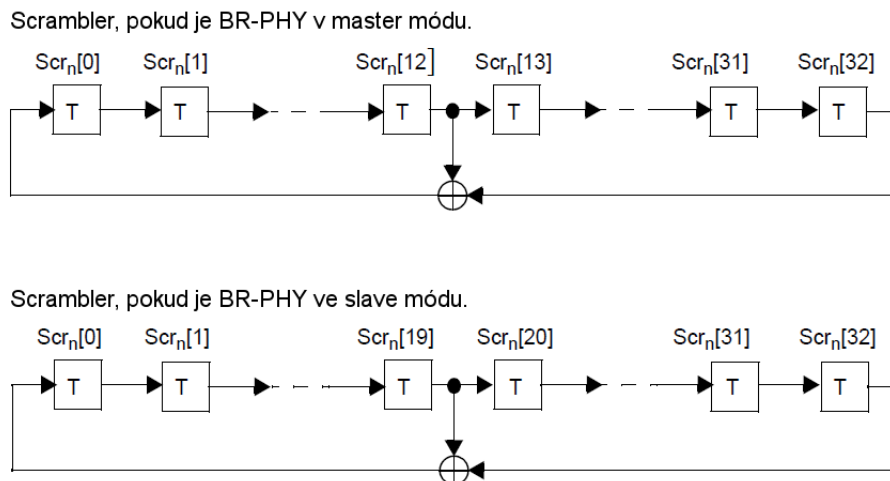
Pokud se jedná o SLAVE mód, použije se polynom:



Obrázek 3.7: Blokový diagram mapování symbolů (převzato z [1])

$$g_M(x) = 1 + x^{20} + x^{33}$$

Implementace scrambleru pomocí lineárního zpětnovazebního posuvného registru vidíme na obrázku 3.8. Bity uloženy v posuvném registru v čase  $n$  jsou označeny jako  $Scr_n[32:0]$ . Každou periodou je posuvný registr rozšířen o jeden bit a jeden nový bit je generovaný –  $Scr_n[0]$ . Scrambler je resetovaný, když se vykoná funkce PCS Reset, poté jsou všechny bity 33-bitového vektoru volitelně nastaveny (záleží na výrobci, nebo se nastaví na nuly).



Obrázek 3.8: Scrambler realizovaný posuvným registrem (převzato z [7])



### 3.2.5.2 Generování $Sy_n[2:0]$

$Sy_n[2:0]$  a  $Sc_n[2:0]$  jsou tři bitové vektory. PCS Transmit vytvoří vektor  $Sy_n[2:0]$  a z něj vektor  $Sc_n[2:0]$  a dále jsou tyto vektory použity k eliminaci korelací mezi vysílanými daty  $tx\_data[2:0]$  a ke generování idle a training symbolů. Z hodnoty scrambleru  $Scr_n[0]$  a polynomu  $g(x) = x^3 \oplus x^8$ , jsou generovány 3 bity  $Sy_n[2:0]$ :

$$Sy_n[0] = Scr_n[0]$$

$$Sy_n[1] = g(Scr_n[0]) = Scr_n[3] \oplus Scr_n[8]$$

$$Sy_n[2] = g^2(Scr_n[0]) = Scr_n[6] \oplus Scr_n[16]$$

### 3.2.5.3 Generování $Sc_n[2:0]$

Bity  $Sc_n[2:1]$  jsou generovány následovně:

$$Sc_n[2:1] = \begin{cases} [0\ 0], & \text{pokud } tx\_mode = SEND_z \\ Sy[2:1], & \text{jinak} \end{cases}$$

$$Sc_n[0] = \begin{cases} 0, & \text{pokud } tx\_mode = SEND_z \\ Sy[0], & \text{jinak} \end{cases}$$

### 3.2.5.4 Generování scamblovacích bitů $Sd_n[2:0]$

Bity  $Sd_n[2:0]$  jsou generovány z bitů  $Sc_n[2:0]$  a  $tx\_data[2:0]$ :

$$Sd_n[2] = \begin{cases} Sc_n[2] \oplus tx\_data\_n[2], & \text{pokud } tx\_enable_{n-3} = 1 \\ Sc_n[2] \oplus 1, & \text{pokud } loc\_rcvr\_status = OK \\ Sc_n[2] & \text{jinak} \end{cases}$$

$$Sd_n[1:0] = \begin{cases} Sc_n[1:0] \oplus tx\_data\_n[:0], & \text{pokud } tx\_enable_{n-3} = 1 \\ Sc_n[1:0] & \text{jinak} \end{cases}$$

### 3.2.5.5 Generování ternárního páru ( $TA_n$ , $TB_n$ )

Bity  $Sd_n[2:0]$  generují ternární páry ( $TA_n$ ,  $TB_n$ ). Ternární symbol (0,0) je speciálně použit pro SSD (sekvence (0,0), (0,0), (0,0)), ESD (sekvence (0,0), (0,0), (1,1)) a ESD spolu s  $tx\_error$  (sekvence (0,0), (0,0), (-1,-1)).

### 3.2.5.6 Generování ( $TA_n$ , $TB_n$ ) když $TXMODE = SEND\_I$

Ačkoliv je možných 9 kombinací ternárního páru ( $TA_n$ ,  $TB_n$ ), v training sekvenci je jich použito jenom 6, viz Tabulka 3.1. Ternární pár SSD/ESD není v tomto módu používán.

$Sd_n[2:0]$	Ternární sym. A	Ternární sym. B
000	-1	0
001	0	1
010	-1	1
011	0	1
SSD/ESD	0	0
100	1	0
101	0	-1
110	1	-1
111	0	-1

Tabulka 3.1: Mapování idle symbolů v trénovacím módu (převzato z [1])

### 3.2.5.7 Generování ( $TA_n$ , $TB_n$ ), když $TXMODE = SEND\_N$ a $tx\_enable = 1$

Mapování z  $Sd_n[2:0]$  na ternární páry v data módu je v Tabulce 3.2.

$Sd_n[2:0]$	Ternární sym. A	Ternární sym. B
000	-1	-1
001	-1	0
010	-1	1
011	0	-1
SSD/ESD	0	0
100	0	1
101	1	-1
110	1	0
111	1	1

Tabulka 3.2: Mapování symbolů v módu  $SEND\_N$  (převzato z [1])

### 3.2.5.8 Generování ( $TA_n$ , $TB_n$ ) pro idle sekvenci když $TXMODE = SEND\_N$

Kvůli rovnoměrnému rozložení hustoty výkonu na ternárních párech ( $TA_n$ ,  $TB_n$ ) je použit extra bit  $Sx_n$ , který je generován následovně:

$$Sx_n = Scr_n[7] \oplus Scr_n[9] \oplus Scr_n[12] \oplus Scr_n[14]$$

tx_mode = SEND_N					
		S <sub>x<sub>n</sub></sub> = 0		S <sub>x<sub>n</sub></sub> = 1	
Sdn <sub>n</sub> [2:0]	Ternární s. A	Ternární s. B	Ternární s. A	Ternární s. B	
000	-1	0	-1	0	
001	0	1	1	1	
010	-1	1	-1	1	
011	0	1	1	1	
100	1	0	1	0	
101	0	-1	-1	-1	
110	1	-1	1	-1	
111	0	-1	-1	-1	

Tabulka 3.3: Idle symboly v módu SEND\_N (převzato z [1])

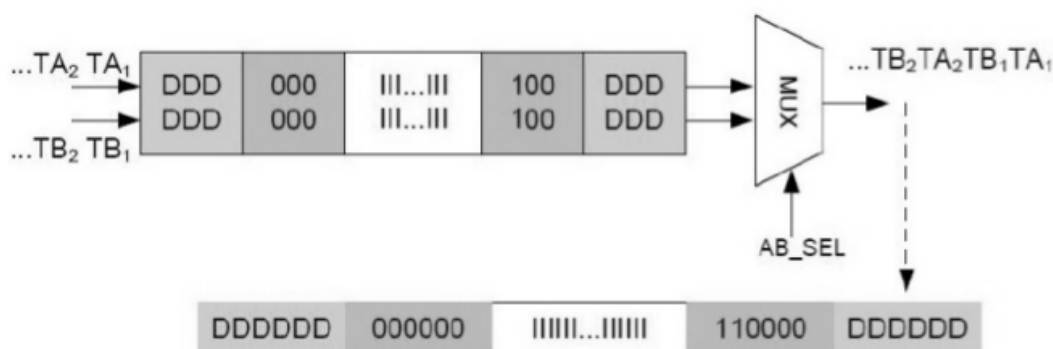
### 3.2.5.9 Generování (TA<sub>n</sub>, TB<sub>n</sub>), když TXMODE = SEND\_Z

Ternární symboly mají hodnotu (0, 0).

### 3.2.5.10 Generování sekvence symbolů

Sériový výstup je vytvořen skládáním ternárních symbolů za sebou, buď pokud máme pár (TA<sub>n</sub>, TB<sub>n</sub>), tak vložíme symbol TA<sub>n</sub> a za něj TB<sub>n</sub>, nebo pokud máme pár (TB<sub>n</sub>, TA<sub>n</sub>), tak vložíme symbol TB<sub>n</sub> a za něj symbol TA<sub>n</sub>. Přijímající strana skládá symboly zpět do párů stejně, aby nedocházelo ke záměně symbolů.

Na obrázku 3.9 je vidět generování 1-D sekvenci symbolů z ternárního páru (TA<sub>n</sub>, TB<sub>n</sub>), výstup čteme zprava doleva. Nejdříve je poslán DATA paket, za ním ESD. Prodlevu mezi daty zaplní IDLE symboly. Poté následuje SSD spolu s dalšími daty.



Obrázek 3.9: Konverze symbolů z 2-D do 1-D (převzato z [1])

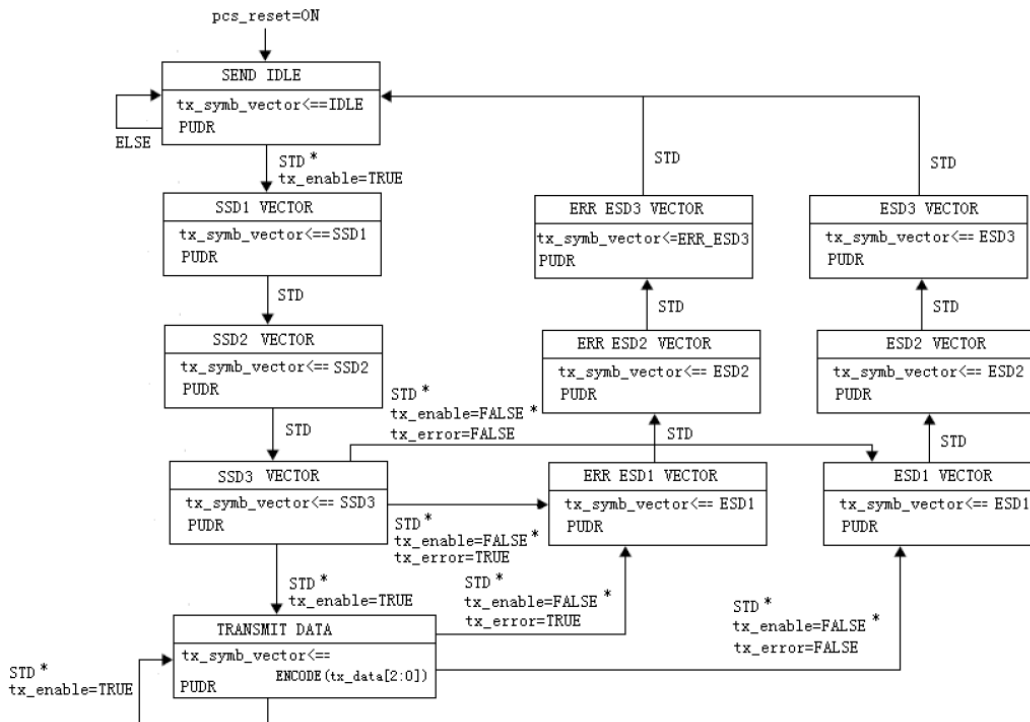
### 3.2.6 Funkce PCS Receive

V této kapitole a podkapitolách jsem čerpal z [1].

Funkce PCS Receive odpovídá stavovému diagramu na obrázku 3.10. Pokud se PCS Receive stavový automat zasekne ve stavu DATA protože nedetekuje ESD1 a ESD2, tak je implementován tzv. JAB stavový automat (viz obrázek 3.11), který zaručuje, že doba ve kterém může automat zůstat ve stavu DATA bude menší než určitá specifická hodnota `rcv_max_timer`. Pokud je dosaženo doby `rcv_max_timer`, automat se resetuje a přechází do stavu IDLE.

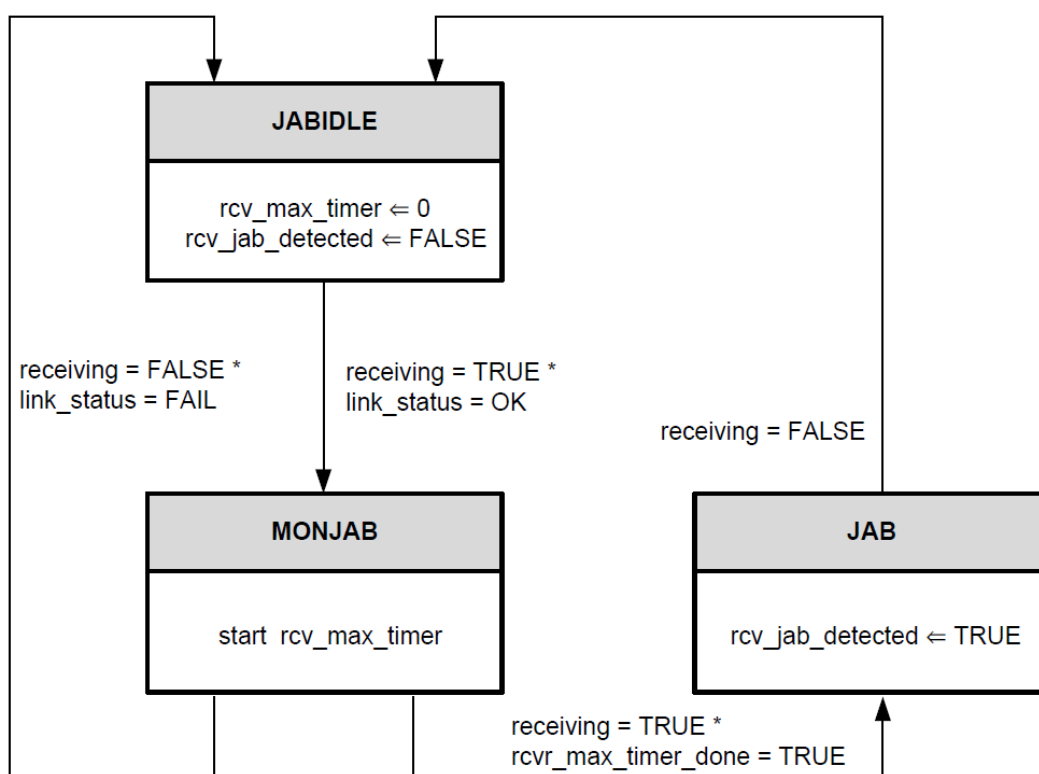
Na PCS Receive stavovém diagramu můžeme vidět, že mezi detekcí SSD3 a stavem DATA jsou další 4 stavy a také tu jsou 4 stavy mezi stavem DATA (včetně) a stavem IDLE, kdy dochází k dekódování dat. Proto zpoždění mezi příchozími daty a odchozími z PCS Receive je stejné a slouží k řádnému přijímání paketů na rozhraní MII.

Funkce PCS receive přijímá symboly poskytované funkcí PMA receive. Přijaté symboly jsou převedené do 2-D ternárních párů ( $RA_n, RB_n$ ). K dosažení korektní operace, PCS receive používá znalosti kódovacích pravidel, které jsou použity v idle módu. Po průchodu dat descramblerem<sup>4</sup> se nastaví parametr `scr_status = OK` a z ternárních párů ( $RA_n, RB_n$ ) se dekóduje a generují signály `rx_data[2:0]`, `rx_dv` a `rc_error`, které se přes konverzi 3B4B převedou na signály `RXD[3:0]`, `RX_DV` a `RX_ER` pro rozhraní MII.



Obrázek 3.10: Stavový diagram funkce PCS Receive (převzato z [1])

<sup>4</sup>Descrambler ze zamíchaných dat vytvoří opět platná data, je to opačná funkce než scrambler.



Obrázek 3.11: Stavový diagram funkce JAB (převzato z [1])

### 3.2.6.1 Proměnné

- `mii_fc_err` – Chyba na lince (false carrier error).
- `Receiving` – Generováno funkcí PCS Receive, pokud je hodnota TRUE, znamená to, že se přenáší data.
- $Rx_n$  – Poslední přijatý symbol v čase  $n$ .
- `pcs_rx_er` – Signál indikující chybu ve funkci PCS Receive, synchronizovaný s `pcs_rxclk`.
- `pcs_rx_dv` – Signál indikující validní data ve funkci PCS Receive, synchronizovaný s `pcs_rxclk`.
- `rx_data[2:0]` – Dekódovaná data, synchronizovaná s `pcs_rxclk`.
- `JBstate` – Proměnná indikující v jakém stavu se nachází funkce JAB.
- `rcv_jab_detected` – Parametr je nastaven na TRUE, pokud se nacházíme ve stavu JAB, viz obrázek 3.11.
- `INVALID` – Libovolný tří bitový neplatný a ignorovaný výstup.

- `loc_rcvr_status` – Stav lokálního přijímače. Parametr je nastaven funkcí PMA Receive a poslán do PCS přes `PMA_RXSTATUS.indication`. Hodnota OK nebo NOT\_OK.

### 3.2.6.2 Funkce

- `check_idle` – Je funkce použita procesem PCS Receive k detekci, jestli byla přijatá validní idle sekvence po tom, co nastala chyba během procesu. Funkce `check_idle` operuje nad aktuálními 2-D ternárními symboly, které se dekodovali z `tx_symb_vector` a nad dalšími pěti 2-D ternárními symboly.
- `DECODE` – Během PCS Receive procesu, tato funkce dekoduje data přijatá z `tx_symb_vector`, z 1-D ternárních symbolů vytvoří 2-D a z nich pak `rx_data[2:0]`.

### 3.2.6.3 Časovač

- `rcv_max_timer` – Časovač vyprší po 36000 (+/- 1800) cyklech hodin `pcs_rxclk`.

### 3.2.6.4 Zprávy

- `PUDI` – Zkratka pro `PMA_UNITDATA.indication` (obsahuje hodnotu vektoru `rx_symb_vector`).

### 3.2.6.5 Dekódování symbolů v PCS Receive

Když PMA Receive nastaví `loc_rcvr_status = OK`, funkce PCS Receive zkontroluje sekvenci symbolů a hledá buď SSD nebo error.

Když PCS Receive přijme SSD, nastaví `pcs_rx_dv = TRUE`. Pokud přijme ESD nebo ESD s chybou nastaví `pcs_rx_dv = FALSE`. Počet bitů přijatých v rámci je vždy násobek tří, proto po 3B4B konverzi se zahodí zbytek, 1 nebo 2 bity z dat.

PCS Receive nastaví `pcs_rx_er = TRUE` pokud se přijmou špatné ESD, `ERR_ESD` nebo špatné SSD. Když se stavový automat dostane do stavu IDLE, `pcs_rx_er` se resetuje na FALSE.

### 3.2.6.6 PCS Receive descrambler

Fyzická vrstva musí přijatá data descramblerem zpět dekodovat, aby vytvořila data `rx_data[2:0]`, která pošle do MII. Pokud je PHY v MASTER módu, použije descrambler pro dekodování polynom  $g'_M(x) = 1 + x^{20} + x^{33}$  a pokud je ve SLAVE módu, použije polynom  $g'_M(x) = 1 + x^{13} + x^{33}$ .

### 3.2.6.7 PCS Receive automatická detekce polarity (volitelné)

Během trénovacího módu PCS Receive může, volitelně, detekovat správnost polarity. V IDLE módu, scrambler generuje vektor  $Sd_n[2:0]$ , kde  $Sd_n[0] = Scr_n[0]$ . Podle Tabulky 1, když  $Sd_n[0] = 0$ , pak  $\mathbf{TA}_n$  je buď +1 nebo -1, jinak se  $\mathbf{TA}_n$  rovná 0. Podle tohoto pravidla bit  $Scr_n[0]$  závisí pouze na hodnotě  $\mathbf{RA}_n$ . V každém cyklu, se porovná bit  $Scr_n[0]$  se zpracovávanou hodnotou  $\mathbf{RA}_n$ . Pokud nastane souvislá shoda, polarita je správná.

Další způsob detekce polarity je předpokládat určitou hodnotu polarity, poté se monitoruje každou periodu synchronizace descrambleru a pokud je předpoklad špatný, opakujeme proceduru s opačnou polaritou.

Další způsob detekce je na začátku během konfigurace linky, když MASTER PHY začne vysílat symboly k SLAVE PHY. V této době jsou všechny statusy handshaking signálů, např. `rem_rcvr_status`, nastaveny na FALSE. Z této znalosti, by SLAVE měl určit polaritu, pokud má SLAVE polaritu obráceně, tak si otočí polaritu u přijatých signálů i vysílaných. Jelikož si polaritu upravit SLAVE, tak MASTER má polaritu nastavenou vždy správně.

### 3.2.6.8 3B4B konverze MII signálů

Do MII vedou signály `RXC[3:0]`, `RX_DV` a `RX_ER` a jsou synchronizovány s hodinami `RX_CLK` na frekvenci 25 MHz. PCS Receive pracuje se signály `rx_data[2:0]`, `rx_dv`, a `rx_error`. Aby docházelo k synchronizovanému průchodu dat, jsou nastaveny hodiny `pcs_rxclk` na frekvenci 33,3 MHz. Pokud je PHY v MASTER módu, může mít hodinový signál `RX_CLK` stejný zdroj jako `TX_CLK`. Pokud je PHY ve SLAVE módu, frekvence se odvodí pomocí `recovered clock`. Hodiny `pcs_rxclk` mohou mít stejný zdroj jako `RX_CLK`, ale převedena na správnou frekvenci.

Pokud počet bitů v datech `rx_data` není dělitelný čtyřmi, tak zbývající bity jsou stuff bity, které byly přidány během 4B3B konverze na vysílající straně a měly by být zahozeny. `RX_DV` se nastaví na false po posledním konvertovaném nibblu.

## 3.3 Podvrstva Physical Media Attachment (BR-PMA)

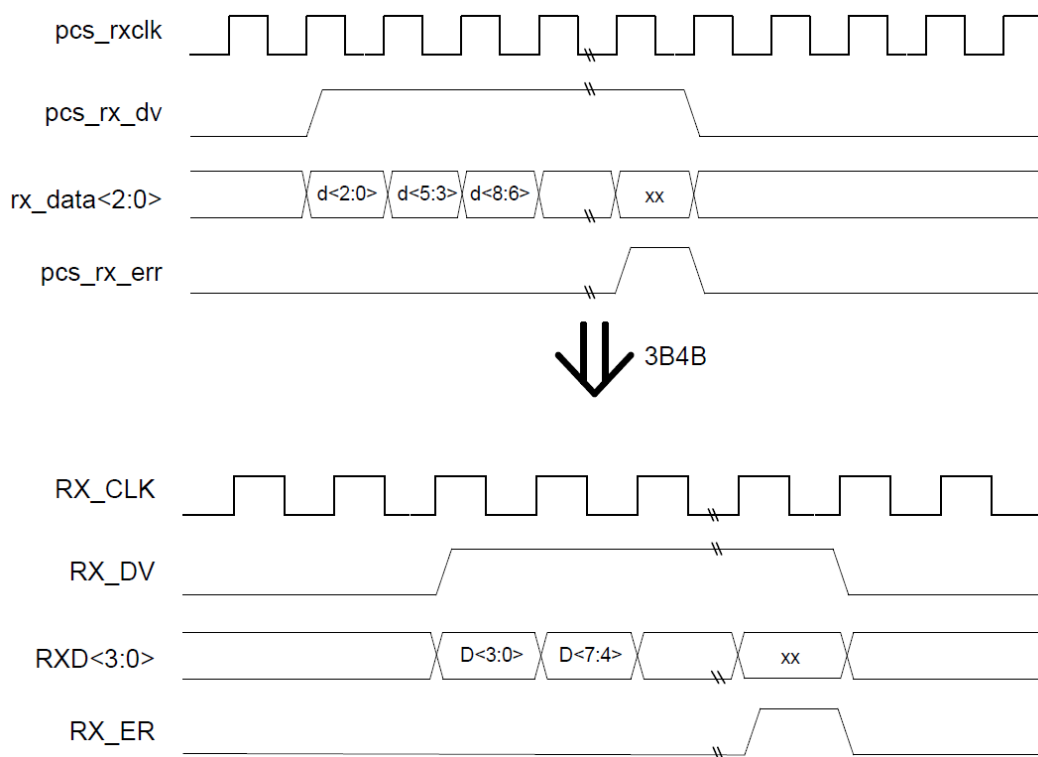
V této kapitole a podkapitolách jsem čerpal z [1].

BR-PMA zprostředkovává rozhraní mezi PCS a Medium Dependent Interface (MDI). Hlavní funkcí BR-PMA je vysílat a přijímat data přicházející do a z MDI přes PAM3. BR-PMA používá pulzně amplitudová modulace, která má diferencovaný výstup s třemi napěťovými úrovněmi (-1, 0, +1).

### 3.3.1 Funkce PMA Reset

PCS Reset inicializuje všechny PCS funkce. Funkce PCS Reset by se měla vykonat po každé když nastane jedna z následujících událostí:

- zapnutí



Obrázek 3.12: 3B4B konverze MII signálů (převzato z [1])

- požadavek na resetování

Pokud nastane jedna z těchto událostí PCS Reset nastaví pcs\_reset=ON. Obrázek 3.13 nezobrazuje PCS Reset funkci.

### 3.3.2 Funkce PMA Transmit

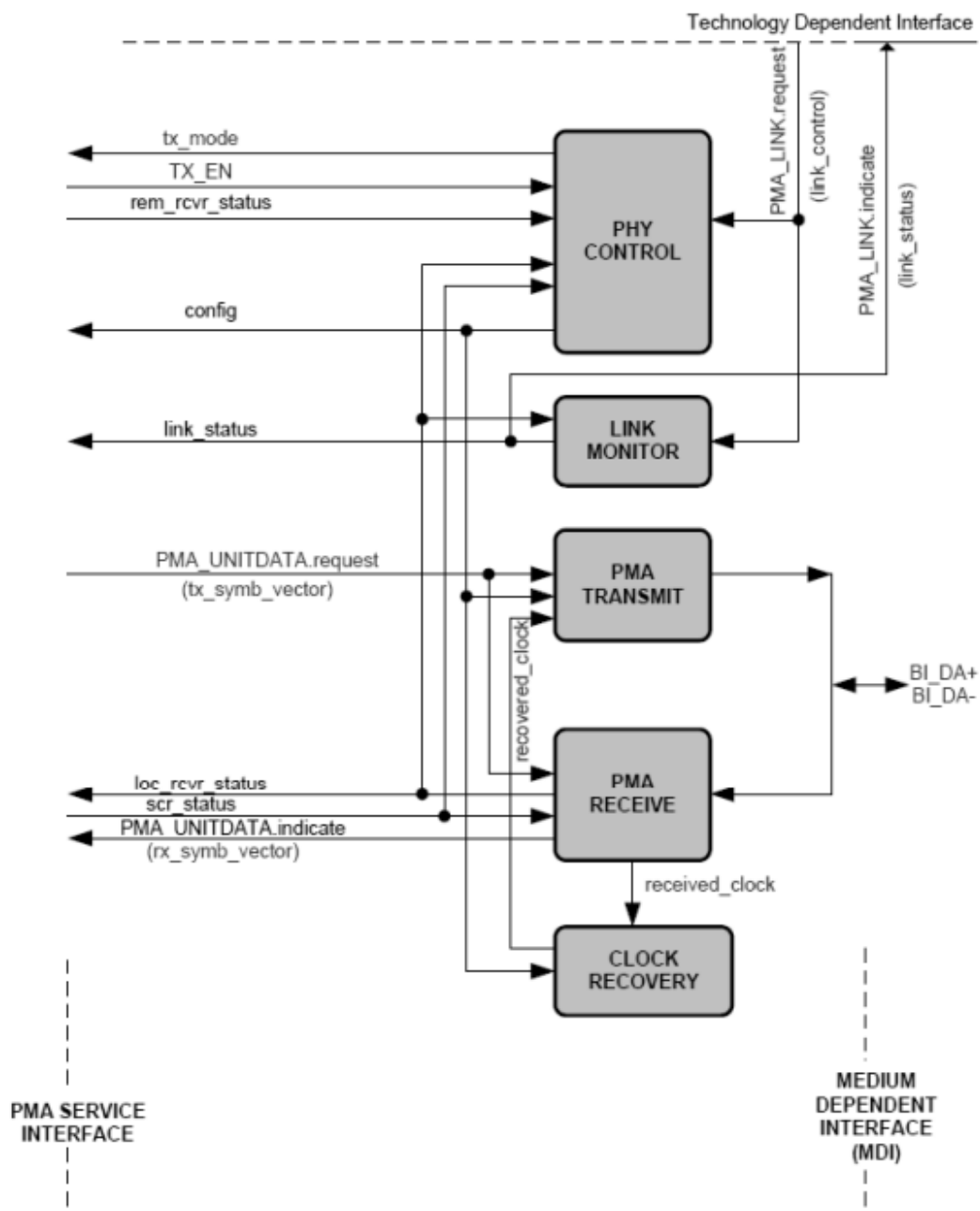
Během přenosu PMA\_UNITDATA.request posílá do PMA data pomocí vektoru tx\_symb\_vector, které jsou dále vysílány přes kroucenou dvojlinku.

PHY Control Config nastavuje tx\_mode na SEND\_N (přenos normálních dat, informace o konfiguraci nebo idle), SEND\_I (přenos skupiny znaků pro IDLE mód) nebo SEND\_Z (přenos nul). Vysílač generuje PAM3 signály BI\_DA s použitím hodin TX\_TCLK s frekvencí 66,6 MHz. Pokud je v PMA\_CONFIG nastaven MASTER mód, pak PMA Transmit použije jako zdroj hodin lokální hodiny, pokud se jedná o SLAVE mód, hodiny se získají z recovered clock.

### 3.3.3 Funkce PMA Receive

BR-PMA Receive se skládá se samotného přijímače pro ternární PAM signály, BI\_DA. PMA Receive překládá přijaté signály z kroucené dvojlinky do vektoru rx\_symb\_vector, který je parametrem funkce PMA\_UNITDATA.indicate.



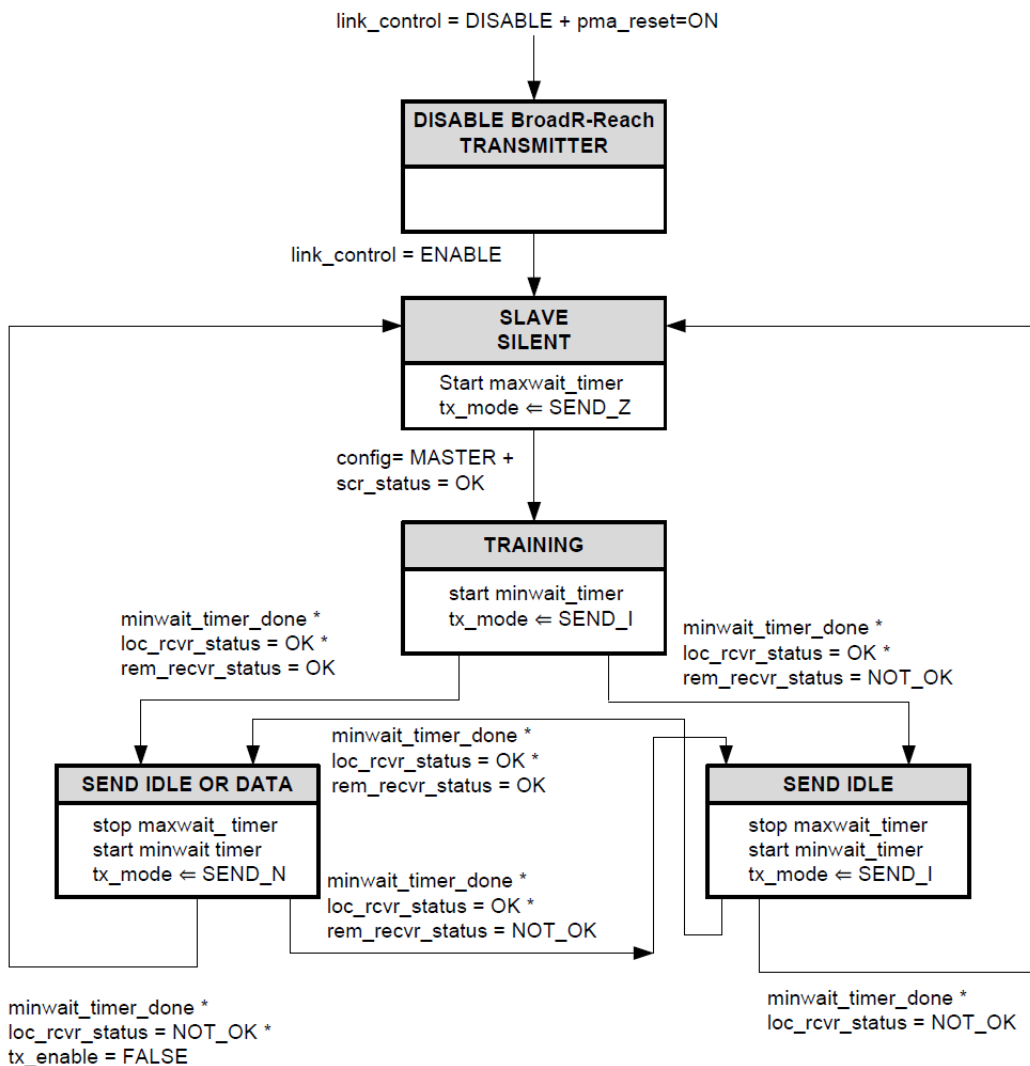


Obrázek 3.13: BR-PMA diagram (převzato z [1])

PMA Receive má dílčí funkce Signal Equalization a Echo Cancellation. Tyto pod-funkce generují `loc_rcvr_status` (status lokálního přijímače). Tento příznak se posílá do PCS Transmit, PCS Receive, kontrolní funkce PMA PHY a Link Monitoru a říká, jestli linka pro příjem dat je OK. `SCR_STATUS` je generován v PCS Receive pro komunikaci s descramblerem pro lokální PHY. Signál posílá zprávu do PMA Receive, jestli je scrambler synchronizovaný.

### 3.3.4 Funkce PHY Control

Na rozdíl od 1000Base-T BR nepoužívá auto-negotiation<sup>5</sup>. Ke spojení dvou BroadR-Reach uzlů a vytvoření linky se používá FORCE mód. Během FORCE módu se v pma\_config nastaví, kdo je Master a kdo Slave, přes tzv. Management Control, během inicializace, nebo přes výchozí hardwarové nastavení. PMA PHY Control také generuje signály, které řídí PCS a PMA operace, určují zda PHY pracuje v normálním módu (je umožněn přenos dat přes segment linky) nebo jestli PHY posílá speciální skupinu znaků reprezentující idle mód.



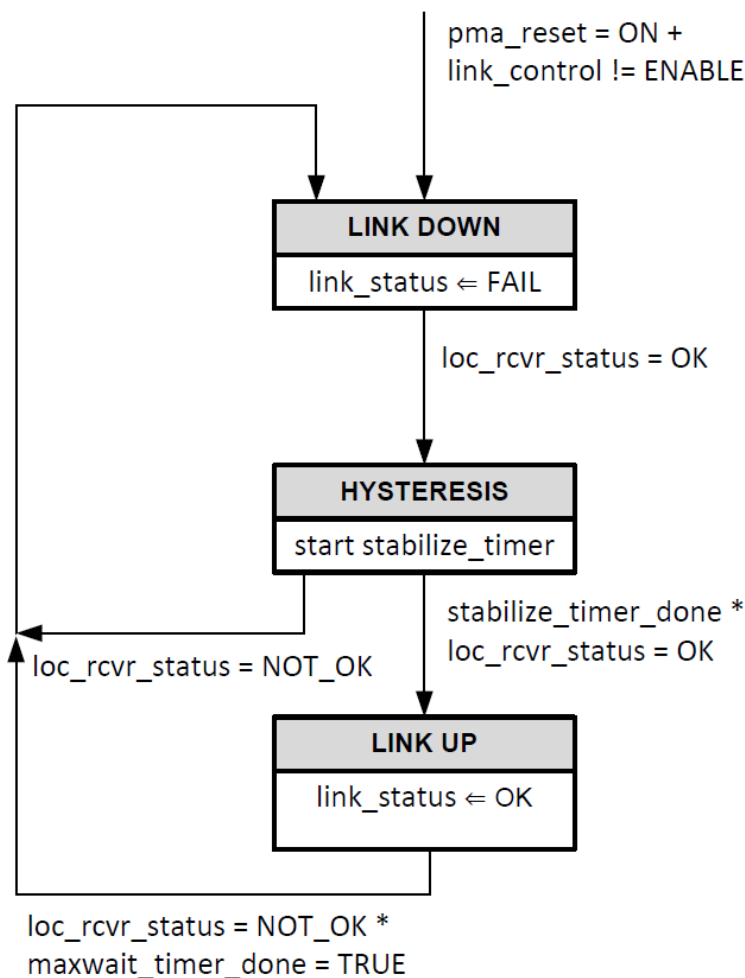
Obrázek 3.14: Stavový diagram PHY Control (převzato z [1])

<sup>5</sup>Technologie pro automatické nastavení a vytvoření ethernetového kanálu mezi dvěma zařízeními

### 3.3.5 Funkce Link Monitor

Link Monitor určuje status přijímajícího kanálu (proměnná `link_status`). Pokud dojde k selhání, tak se činnosti PMA funkcí pozastaví. Funkce mohou pokračovat v činnosti hned, jakmile je dosaženo spolehlivého přenosu a proměnná `link_status` se nastaví na hodnotu OK.

`link_control` se nastavuje do stavu ENABLE během PHY inicializace, během FORCE módu. Během FORCE módu Link Monitor podporuje BR-PHY Control operace.



Obrázek 3.15: Stavový diagram Link Monitoru (převzato z [1])

### 3.3.6 Funkce PMA Clock Recovery

Tato PMA funkce dokáže z přijaté sekvence znaku zpětně získat hodinový signál. Výstup z PMA Clock Recovery je použit jako hodiny pro další PMA funkce.

### 3.3.7 Proměnné

- `pma_reset` – Pomocí proměnné `pma_reset` PMA Reset resetuje všechny PMA funkce.
- `config` – Proměnnou `config`, posílá PMA do PCS informaci jestli je PHY ve stavu MASTER nebo SLAVE, pomocí `PMA_CONFIG.indication`.
- `link_control` – Proměnná je nastavena na ENABLE nebo DISABLE pomocí managementu nebo implicitně.
- `rem_rcvr_status` – Tuto proměnnou nastavuje PCS Receive a indikuje přijímající linka po vzdálené PHY pracuje spolehlivě. Hodnoty jsou OK nebo NOT\_OK.
- `scr_status` – Proměnná indikující jestli byl descrambler synchronizován. Hodnota je OK, nebo NOT\_OK

### 3.3.8 Časovače

- `maxwait_time` – Časovač je použit ve stavový diagramech PHY Control a Link Monitor, aby omezil dobu, během které přijímač setrvává ve stavu SLAVE SILENT a TRAINING. Časovač vyprší po  $1406 \text{ ms} \pm 18 \text{ ms}$ , pokud `config = MASTER`, nebo  $656 \text{ ms} \pm 9 \text{ ms}$ , pokud `config = SLAVE`. Časovač je testován Link Monitorem, aby nastavil `link_status` na FAIL, pokud časovač vyprší a `loc_rcvr_status` je stále NOT\_OK.
- `minwait_timer` – Časovač určuje minimální dobu, po kterou PHY Control zůstane ve stavu TRAINING, SEND, IDLE, nebo DATA. Časovač vyprší po  $1,8 \mu\text{s} \pm 0,18 \mu\text{s}$ .
- `stabilize_timer` – Časovač je použit pro řízení minimální doby, po kterou musí být `loc_rcvr_status = OK`, než se přejde k Link Up. Časovač je nastaven na  $1,8 \mu\text{s} \pm 0,18 \mu\text{s}$ .

## 3.4 Elektrická specifikace podvrstvy PMA

V této kapitole a podkapitolách jsem čerpal z [1].

Systémy obsahující BroadR-Reach PHY by měly splňovat požadavky elektromagnetické kompatibility (EMC) pro automobilové aplikace. V CISPR 25 jsou definovány metody pro testování a měření EMC, hlavně RF imunitu a vyzařování.

### 3.4.1 Imunita – DPI test

V reálných aplikacích, kvůli elektromagnetické interferenci, dochází k radiofrekvenčnímu rušení mezi fyzickými medii, které vedeme vedle sebe. K změření citlivosti PMA přijímače vzhledem k radiofrekvenčnímu rušení se používá testovací metoda Direct Power Injection (DPI) – popsáno v IEC62132-4<sup>6</sup>.

---

<sup>6</sup>International Electrotechnical Commission je organizace zabývající standardizací elektronických technologií, viz <http://www.iec.ch/>

### 3.4.2 Emise – 150Ohm řízený test emisí

Pro otestování vysílání emisí v elektrickém prostředí PMA vysílačem, může být použita 150Ohm testovací metoda – popsáno v IEC61967-4.

### 3.4.3 Frekvence hodin vysílače

Přenos ternární symbolů na rozhraní MDI by měl probíhat na frekvenci  $66,6 \text{ MHz} \pm 0,01 \%$ .

### 3.4.4 Testovací módy

V Tabulce 4 jsou popsány testovací módy. Testuje se časový průběh vysílajícího signálu, zkreslení vysílaného signálu, jitter<sup>7</sup> a tzv. droop<sup>8</sup>.

Pokud je PHY v testovacím módu, smí měnit jenom datové symboly určené pro vysílající obvod a neměl by měnit elektrické charakteristiky a jitter vysílače a přijímač z normálního operačního módu. Tyto módy jsou zpřístupněny nastavením 3-bitovým ovládacím registrem.

Registr			Mód
0	0	0	Normální operace
0	0	1	Test mód 1 – Droop test
0	1	0	Test mód 2 – Jitter test v MASTER módu
0	1	1	Test mód 3 – Jitter test ve SLAVE módu (rezervováno)
1	0	0	Test mód 4 – Test zkreslení signálu
1	0	1	Test mód 5 – Normální operace při plném výkonu – PSD testování
1	1	0	Rezervováno
1	1	1	Rezervováno

Tabulka 3.4: Registr pro testovací módy (převzato z [1])

- Test mód 1 – PHY vysílá N symbolů „+1“ a následně N symbolů „-1“. Hodnota N by měla být taková, aby doba periody, za kterou se vyšle N symbolů, byla větší než 500 ns. Tato sekvence se vysílá nepřetržitě. Např. by mělo být dostatečné vysílat 40 symbolů (600 ns) pro testování. Na obrázku 3.19 je znázorněn příklad výstupu vysílače.
- Test mód 2 – PHY vysílá sekvence  $\{+1, -1\}$  za sebou s frekvencí  $66,6 \text{ MHz} \pm 0,01 \%$  v MASTER módu.
- Test mód 3 – Tento mód je volitelný a nejsou tu žádné časové požadavky na jitter ve SLAVE módu. Nicméně v sekci 3.4.6.3 jsou požadavky pro SLAVE zařízení v normálním módu, pokud je linka nastavena pomocí MASTER zařízení. Pokud je test mód 3 povolen, PHY opakovaně posílá sekvenci znaků  $\{+1, -1\}$  za sebou s frekvencí  $66,6 \text{ MHz} \pm 0,01 \%$  ve SLAVE módu.

<sup>7</sup>Jitter je nežádoucí odchylka periodického signálu.

<sup>8</sup>Droop je pokles napětí během stavu v log.1.

- Test mód 4 – PHY vysílá sekvenci symbolů generovanou polynomem  $g_{s1} = 1 + x^9 + x^{11}$ . Maximální délka posuvného registru, který generuje sekvenci symbolů pomocí polynomu  $g_{s1}$  musí být aktualizovaný každou periodou (15 ns). Bity v posuvném registru jsou v čase  $n$  vyjádřeny pomocí  $Scr_n[10:0]$ . Každou periodou je registr posunut o jeden bit a jeden nový bit se generuje v  $Scr_n[0]$ . Bity  $Scr_n[8]$  a  $Scr_n[10]$  se spolu XORují a vytvoří nový bit  $Scr_n[0]$ . Pomocí scrambler bitů se spočítají bity  $x0_n$  a  $x1_n$  a z nich se vytvoří ternární symbol, viz Tabulka 3.5. Ternární symboly jsou vysílány s frekvencí  $66,6 \text{ MHz} \pm 0,01 \%$  v MASTER módu.

$$x0_n = Scr_n[0]$$

$$x1_n = Scr_n[1] \oplus Scr_n[4]$$

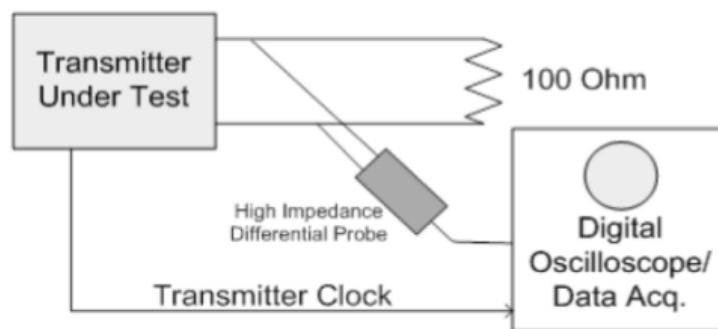
$x1_n$	$x0_n$	Vysílaný PA M3 symbol
0	0	0
0	1	1
1	0	0
1	1	-1

Tabulka 3.5: Mapování symbolu v testovacím módu 4 (převzato z [1])

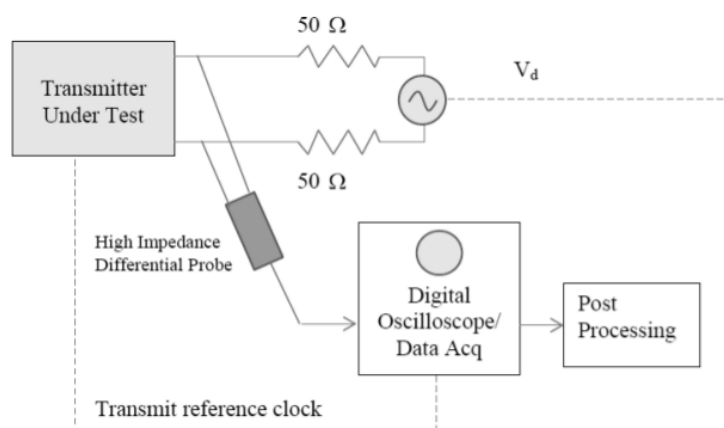
- Test mód 5 – PHY vysílá pseudonáhodné sekvence PAM3 symbolů generované scramble funkcí (ze sekce 3.2.5.1) a ověřuje se, jestli vysílač splňuje PSD požadavky (viz sekce 3.4.6.4).

### 3.4.5 Testování

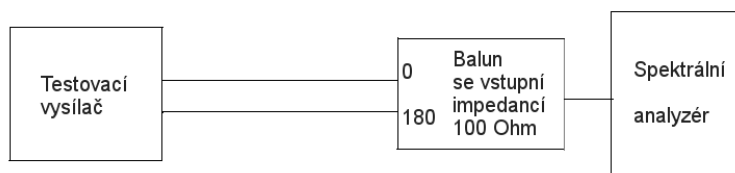
Na obrázku 3.16, 3.17 a 3.18 jsou znázorněné způsoby měření specifikací vysílače.



Obrázek 3.16: Testovací obvod 1: Droop, Jitter (převzato z [1])



Obrázek 3.17: Testovací obvod 2: Zkreslení (převzato z [1])



Obrázek 3.18: Testovací obvod 2: Testování PSD (převzato z [1])

Abychom mohli měřit jitter ve SLAVE módu, PHY by mělo umožnit přístup k hodinám TX\_TCLK s frekvencí 66,6 MHz, které jsou použity k vysílání symbolů. Pokud výstup hodin není povolen, PHY by mělo poskytnout k tomu prostředky.

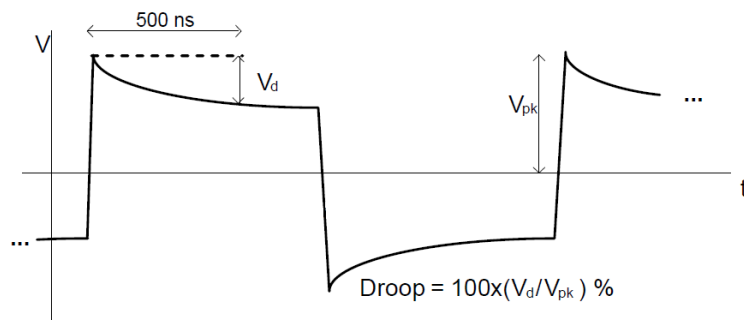
Rušící signál  $V_d$  má amplitudu 5,4 V peak-to-peak a frekvenci  $\frac{66,6}{6}$  MHz. Generátor rušivého signálu musí mít dostatečnou linearitu a rozsah, aby nezaváděl značné zkreslení, když je zapojen k výstupu vysílače, jinak můžeme např. generátor vyměnit za jiný s jinou výstupní frekvencí a ověřit jestli výsledné spektrum nevykazuje významné zkreslení.

### 3.4.6 Elektrické specifikace vysílače

PMA musí pracovat s AC s vazbou na MDI. Pokud není zátěž specifikována, použije se  $100\ \Omega \pm 1\%$  diferenční odporovou zátěž ke každému výstupu.

#### 3.4.6.1 Pokles napětí (droop) výstupu vysílače

Na obrázku 3.19 je vidět výstupní pokles napětí testovacího módu 1 s použitím testovacího obvodu 1 (viz obrázek 3.16). Měříme po prvním dosažení maximální hodnoty (peak) hodnoty napětí a po dobu 500 ns, pokles napětí (v kladném nebo záporném směru) musí být menší než 26,9 %.



Obrázek 3.19: Výstup testovacího módu 1, měřítko není ve správném poměru (převzato z [1])

### 3.4.6.2 Zkreslení vysílače

Zkreslení vysílače je měřeno pomocí časového průběhu testovacího módu 4 a použitím testovacího obvodu 2. K určení zkreslení vrcholové hodnoty napětí je potřeba nejdříve navzorovat výstupní diferenční signál s frekvencí hodin TX\_TCLK s libovolnou fází, a pak blok po sobě jdoucích vzorků zpracujeme pomocí MATLAB kódu (viz [1, 62/63]) nebo jemu podobnému. Vzorky měříme s minimálně 10 odlišnými rovnoměrně rozloženými fázemi periody. Zkreslení vrcholové hodnoty napětí (peak) by mělo být menší než 15 mV.

MATLAB kód odstraňuje rušivý signál z naměřených hodnot a vypočítá zkreslení. Kód předpokládá, že rušivý signál a frekvence sběru dat jsou stejné jako hodiny vysílače, který testujeme.

### 3.4.6.3 Jitter

Při testování jitteru v testovacím módu 2, musí být efektivní hodnota výstupního jitteru z MDI,  $J_{TXOUT}$ , vzhledem k referenční hodnotě (jitter = 0) menší než 50 ps. Pro měření jitteru není definována horní propust, takže komponenty s nízkou frekvencí by měly být filtrovány během nastavení pomocí omezení paměti osciloskopu. Např. osciloskop s 20 GSsample/s a 2 MB paměti, můžeme použít pro měření frekvence 1 kHz a více.

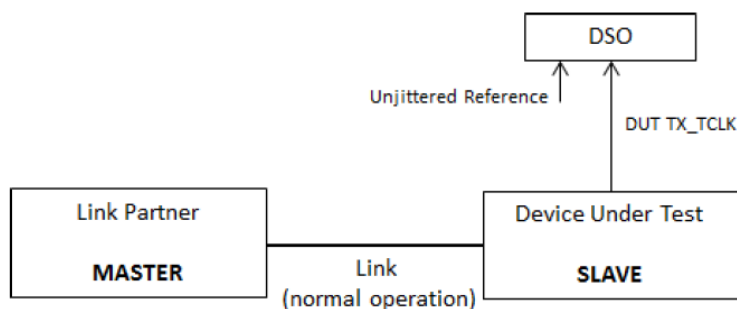
Během normálního módu jitter přijímače, operující ve SLAVE módu, odráží jitter hodin TX\_TCLK. Poté co přijímač začne správně přijímat data od vysílače v MASTER módu, by kvadratický průměr jitteru SLAVE hodin TX\_TCLK vůči referenční hodnotě měl být menší než 0,01 doby periody.

Měření jitteru by mělo probíhat na intervalu větším než 1 ms.

### 3.4.6.4 Power Spectral Density vysílače

Během testovacího módu 5, PHY vysílá náhodné sekvence ternárních symbolů, které jsou reprezentovány jako diferenciální napěťové úrovně [-1, 0, +1]. K testování se použije obvod na obrázku 3.18. PSD vysílače by měl být mezi horní a spodní hranicí, viz Tabulka 3.6 a obrázek 3.21.





Obrázek 3.20: Měření jitteru v normálním módu (převzato z [1])

Frekvence [MHz]	Horní hranice [dBm]	Spodní hranice [dBm]
1	-23,3	-30,9
20	-24,8	-35,8
40	-28,5	-49,2
57-200	-36,5	–

Tabulka 3.6: PSD maska (převzato z [1])

#### 3.4.6.5 Frekvence hodin vysílače

Frekvence vysílání symbolů v MASTER módu je  $66,6 \text{ MHz} \pm 0,01 \%$ .

#### 3.4.7 Elektrické specifikace přijímače

BR-PMA musí splňovat Funkční specifikace vysílače ze sekce 3.3.3 a elektrické specifikace z této sekce. Kroucená dvojlanka v testovací konfiguraci musí splňovat limity ze sekce 3.6.

##### 3.4.7.1 Diferenciální vstupní signál přijímače

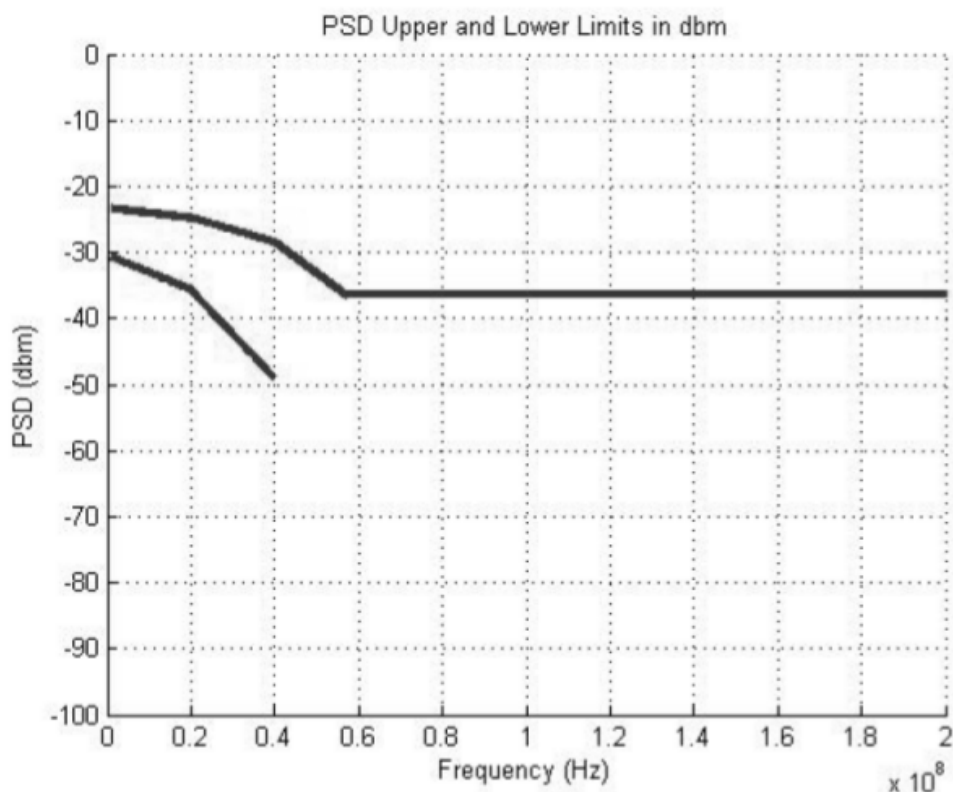
Diferenciální signály, které byly vyslány z vysílačů, splňující 3.4.6, a prošly fyzickým spojem specifikovaným v 3.6, jsou přijaty s chybovostí (bit error rate – BER) menší než  $10^{-10}$ , a pak jsou poslány do PCS po dokončení resetování linky.

##### 3.4.7.2 Frekvence hodin přijímače

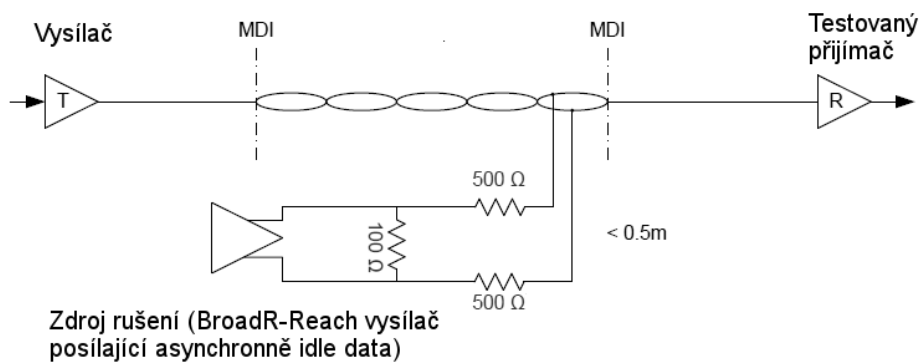
Přijímač musí přijímat příchozí data frekvencí hodin  $66,6 \text{ MHz} \pm 0,01 \%$ .

##### 3.4.7.3 Omezení přeslechů

Tato sekce ověřuje toleranci testovaného zařízení vůči cizímu rušení. Jako zdroj rušení je zvoleno 100 Mb/s BroadR-Reach vysílač připojený k testovanému segmentu, který posílá idle symboly. Úroveň šumu je 100 mV peak-to-peak, tj. asi o 26 dB méně než vysílaný signál. Na obrázku 3.22 vidíme zapojení. Chybovost (BER) by měla být menší než  $10^{-10}$ .



Obrázek 3.21: PSD horní a spodní limity (převzato z [1])



Obrázek 3.22: Obvod pro testování přeslechů (převzato z [1])

### 3.5 Rozhraní pro správu

V této kapitole a podkapitolách jsem čerpal z [1].

BR využívá funkce pro správu, které poskytuje MII Management Interface a automobilová specifická konfigurace <PHY – Inicializace>, viz sekce 3.5.2.

### 3.5.1 MASTER-SLAVE konfigurace

Všechna BroadR-Reach zařízení, po zapnutí nebo resetování, jsou implicitně nastavena jako SLAVE zařízení, dokud je řídicí systém (např. procesor nebo mikrokontrolér) nepřepne do MASTER módu.

Rozdělení zařízení na MASTER-SLAVE je důležité pro nastavení časování každé PHY.

### 3.5.2 PHY – Inicializace

Obě zařízení na každé straně segmentu mohou být MASTER nebo SLAVE. Na základě fyzického rozložení PHY v automobilu se během zapnutí zařízení nebo resetování, pomocí přiřazovacího schématu, rychle rozhodne, kdo bude MASTER. Řídicí systém nakonfiguruje BroadR-Reach PHY jako MASTER, zatímco partnerské PHY zůstane implicitně jako SLAVE.

### 3.5.3 Hodiny pro řídicí data - Management Data Clock (MDC)

Station Management Entity<sup>9</sup> používá MDC jako referenční hodiny pro přenos informací přes MDIO (viz sekce 3.5.4 signály). Minimální délka logické úrovně je 160 ns a minimální doba periody je 400 ns bez ohledu na nominální periodu hodin TX\_CLK a RX\_CLK.

### 3.5.4 Vstup/výstup řídicích dat

Management data input/output (MDIO) je obousměrná sériová sběrnice mezi PHY a STA. MDIO přenáší řídicí informace z STA do PHY a informační statusy z PHY do STA, s ohledem na hodiny MDC. MDIO je řízený třístavovým automatem, který umožňuje, aby signál řídil jak STA tak i PHY.

## 3.6 Charakteristiky fyzického segmentu

V této kapitole a podkapitolách jsem čerpal z [1].

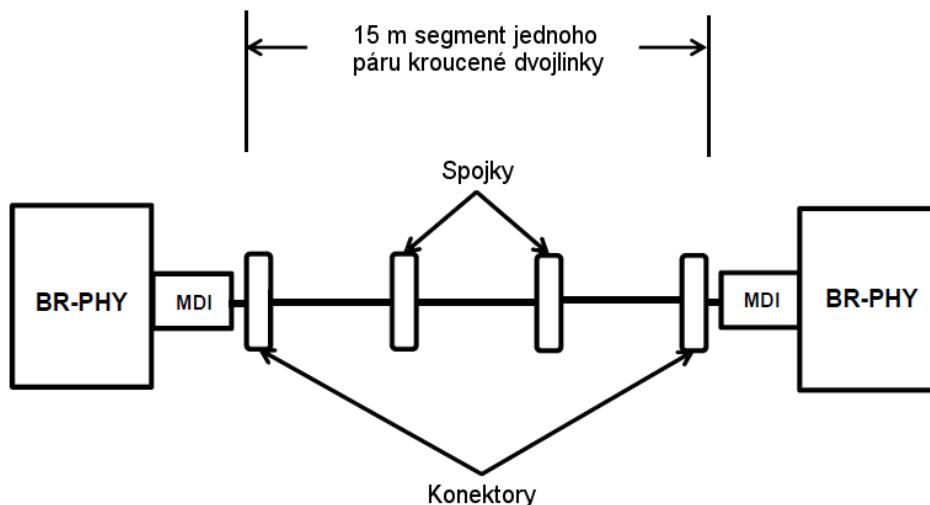
BR-PHY pracuje s jedním párem kroucené dvojlinky podporující rychlost přenosu 100 Mb/s v obou směrech zároveň. Segment se skládá z 15 m UTP měděného kabelu s impedancí 90 - 110 Ohm a dvou koncových konektorů.

### 3.6.1 Charakteristika kabeláže

Přenosové parametry (insertion loss, return loss a charakteristická impedance) zaručují, že segment kroucené dvojlinky bude spolehlivé přenosové medium.

---

<sup>9</sup>STA – řídí MII rozhraní.



Obrázek 3.23: BR fyzický segment (převzato z [1])

### 3.6.1.1 Charakteristická impedance

Charakteristická impedance kabelu by mělo být  $100 \text{ Ohm} \pm 10 \%$  a doba náběhu hrany menší než 700 ps.

### 3.6.1.2 Útlum na vedení

Útlum signálu, při průchodu vedením a konektory, by měl odpovídat hodnotám z tabulky 3.7.

dB	f [MHz]
< 1,0	1
< 2,6	10
< 4,9	33
< 7,2	66

Tabulka 3.7: Maximální útlum pro dané frekvence (převzato z [1])

### 3.6.1.3 Útlum odrazem

Útlum odrazem je výkon signálu, který se odráží proti zdroji signálu. Útlum by měl být odpovídat hodnotám z tabulky 3.8.

### 3.6.1.4 Šum z okolního prostředí

V prostředí automobilu máme různé typy rušení, např.:

dB	f [MHz]
18	1 – 20
$18 - 10\log_{10}(f/20)$	20 – 66

Tabulka 3.8: Útlum odrazem (převzato z [1])

Echo z vysílače na stejné kroucené dvojlince. To je způsobené tím, že komunikace po kroucené dvojlince probíhá v obou směrech zároveň nebo impedanční nespojitostí fyzického segmentu. Echo by se mělo ošetřit, aby se zmenšila chybovost přijatého signálu. Techniky ošetření jsou ponechány na výrobci PHY zařízení.

Tepelný šum je typický pro prostředí automobilu. Tepelný šum do -140 dBm/Hz není kritický, aby ovlivnil výkon.

Pokud je k sobě připevněno více párů, může mezi nimi docházet k přeslechům (NEXT<sup>10</sup>, FEXT<sup>11</sup>) a snižování kvality signálu. Svazek by měl obsahovat max. 6 páru kroucené dvojlinky.

Podmínky pro Power Sum NEXT<sup>12</sup> pro svazek 6-ti párů:

$$\text{Power Sum NEXT LOSS [dB]} > 31,5 - 10 * \log_{10}\left(\frac{f}{100}\right)$$

Podmínky pro Power Sum ELFEXT<sup>13</sup> pro svazek 6-ti párů:

$$\text{Power Sum ELFEXT [dB]} > 16,5 - 20 * \log_{10}\left(\frac{f}{100}\right)$$

Frekvence  $f$  je 1 – 100 MHz.

## 3.7 MDI specifikace

V této kapitole a podkapitolách jsem čerpal z [1].

Tato sekce definuje MDI pro BR automobilové aplikace.

### 3.7.1 MDI konektory

Mechanické rozhraní může být 2-pinové nebo multi-pinové. Konektory by neměly zhoršovat signál v rámci insertion loss nebo return loss.

### 3.7.2 Elektrické specifikace MDI

MDI konektory by měli také splňovat elektrické specifikace ze sekce 3.6.1.

<sup>10</sup>Near-end Crosstalk je přeslech mezi vodiči na blízkém konci.

<sup>11</sup>Far-end Crosstalk je přeslech mezi vodiči na vzdáleném konci.

<sup>12</sup>Power Sum NEXT je součet přeslechů NEXT.

<sup>13</sup>Power Sum Equal Level FEXT je součet vyrovnaných přeslechů na vzdálených koncích. V úvahu se bere i útlum na vodiči.

### 3.7.2.1 MDI charakteristická impedance

Charakteristická impedance kabelu by mělo být  $100 \text{ Ohm} \pm 10 \%$  a doba náběhu hrany menší než 700 ps.

### 3.7.2.2 MDI Return loss

Return loss konektorů by měl odpovídat následujícím hodnotám:

dB	f [MHz]
20	1 – 40
$26 - 0.15 * f$	40 – 66

Tabulka 3.9: MDI return loss (převzato z [1])

## 3.8 Omezení zpoždění

Čerpáno z [1].

Každé BR-PHY by mělo splňovat maximální zpoždění pro přenosovou cestu mezi MII a kroucenou dvojlinkou. Zpoždění přenosu z MII do kroucené dvojlinky musí být menší než 240 ns, zpoždění přenosu mezi kroucenou dvojlinkou a MII musí být menší než 780 ns.

## 3.9 Testování na úrovni systému

V této kapitole a podkapitolách jsem čerpal z [1].

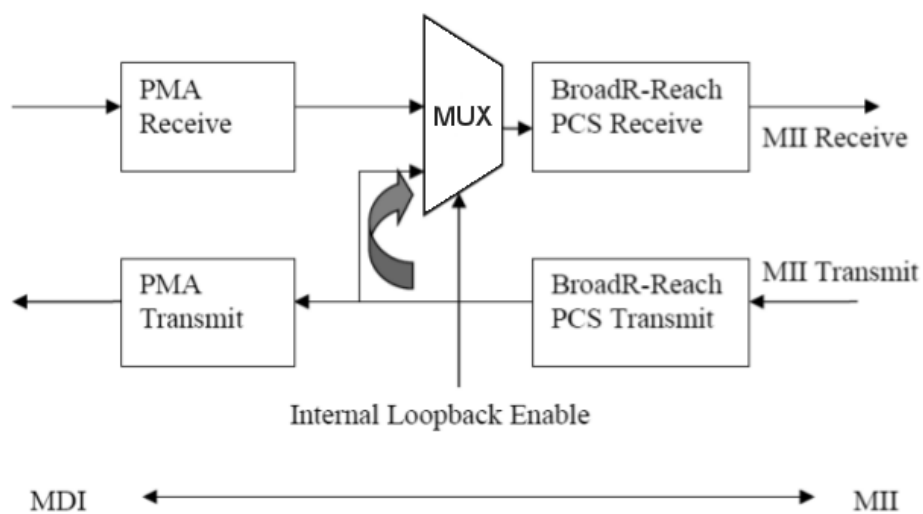
BR-PHY podporuje 2 loopback testovací módy pro podporu MAC. Můžeme tak testovat funkcionalitu PHY bez pomoci dalšího zařízení.

### 3.9.1 Interní loopback funkce

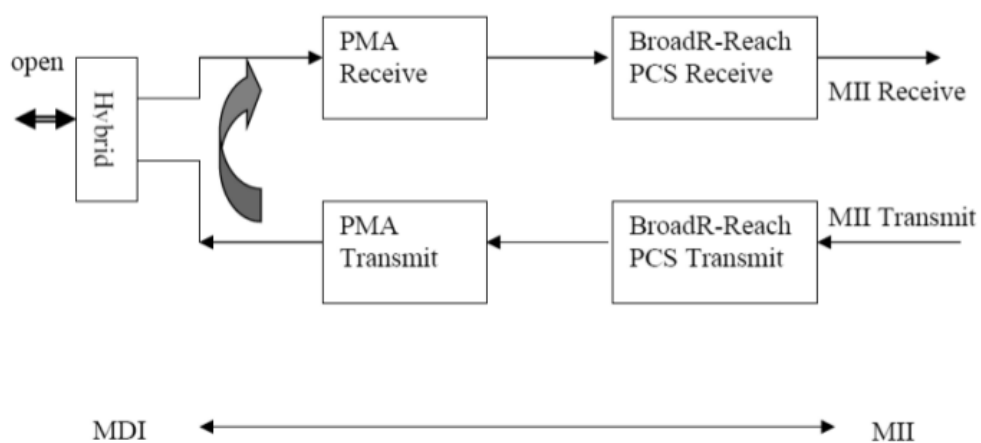
Když je PHY v interním loopback testovacím módu, namísto toho, aby PCS Receive přijímalo symboly z PMA Receive, tak přijímá PAM3 symboly přímo z PCS Transmit. Tudiž data neprocházejí PMA funkcemi. MAC porovnává rámce, které poslala přes MII Transmit funkci s rámci přijatými z MII Receive funkce a tím kontroluje funkčnost PCS funkcí.

### 3.9.2 Externí loopback funkce

Když je PHY v externím loopback testovacím módu, funkce PMA Receive využívá echo signály z nezakončeného MDI rozhraní, zpracovává tyto signály a posílá zpět na rozhraní MII Receive. MAC porovnává vyslaná a přijatá data a vyhodnocuje funkčnost PCS a PMA funkcí.



Obrázek 3.24: Interní loopback (převzato z [1])



Obrázek 3.25: Externí loopback (převzato z [1])





## Kapitola 4

# Časová synchronizace a Precision Time Protocol (PTP)

V automobilovém průmyslu je důležitá především bezpečnost, proto automobil obsahuje hodně senzorů a jiných prvků, které nám bezpečnost zajišťují. Pro vyhodnocení situace, potřebujeme mít nejaktuálnější data, většinou z více senzorů zároveň, nebo musíme znát čas jejich pořízení. Potřebujeme, aby zařízení, které spolu pracují, byla časově synchronizovaná a věděla přesně s jakými daty v jaký okamžik pracují. Proto pro nasazení Ethernetu v automobilu je taky nezbytná přesná časová synchronizace. Jelikož je v automobilu na prvním místě bezpečnost, musíme mít synchronizaci co nejpřesnější. K tomu nám poslouží protokol Precision Time Protocol (PTP) verze 2 ze standardu IEEE 1588-2008.

### 4.1 Precision Time Protocol

Protokol PTP popisuje synchronizaci času v distribuovaných systémech s jedním a více uzlů se síťovou komunikací. Hodiny jsou aplikacemi použity např. ke generování časových značek (timestamps), synchronnímu sběru dat, časovému řazení událostí aj.

Máme několik druhů PTP hodin: ordinary clock, boundary clock, end-to-end transparent clock, peer-to-peer transparent clock.

Protokol popisuje, jak se hodiny navzájem synchronizují. Hodiny jsou organizovány do hierarchie master-slave. Hodiny na vrcholu hierarchie se nazývají „grandmaster“ clock. Tyto hodiny poskytují referenční čas pro celý systém. Synchronizace je dosažena pomocí výměny časových zpráv, které slave hodiny používají ke srovnání svého času s časem master hodin.

### 4.2 PTP zprávy

Čerpáno z [3].

PTP definuje 2 druhy zpráv: *zprávy událostí* a *obecné zprávy*. Zprávy událostí jsou časové zprávy, ve kterých jsou generovány časové značky. Obecné zprávy neobsahují časové značky.

1. Zprávy událostí

- Sync
- Delay\_Req
- Pdelay\_Req
- Pdelay\_Resp

2. Obecné zprávy

- Announce
- Follow\_Up
- Delay\_Resp
- Pdelay\_Resp\_Follow\_Up
- Management
- Signaling

Zprávy Sync, Follow\_Up, Delay\_Req a Delay\_Resp jsou použity ke generování a přenos časových informací, potřebných k synchronizaci ordinary a boundary hodin. Zprávy Pdelay\_Req, Pdelay\_Resp a Pdelay\_Resp\_Follow\_Up jsou použity k měření zpoždění přenosu na síti mezi dvěma časově synchronizovanými uzly. Čas zpoždění přenosu paketu na síti je pak použit k synchronizaci hodin. Zpráva Announce je použita k nastavení hierarchie. Management zprávy jsou použity ke správě PTP dat obsažených ve zdrojích času a k nastavení managementu. Signaling zprávy jsou použity pro všechny ostatní účely.

### 4.3 Typy PTP zařízení

V této kapitole a podkapitolách jsem čerpal z [3].

Máme pět typů zařízení:

- Ordinary clock
- Boundary clock
- End-to-end transparent clock
- Peer-to-peer transparent clock
- Management node

#### 4.3.1 Ordinary clock (OC)

OC komunikuje se sítí přes jeden port. OC mohou být jenom v jednom režimu, buď jako grandmaster clock nebo jako slave clock. Pokud jsou hodiny v režimu slave, potom si lokální hodiny synchronizuje podle master hodin. Pokud jsou hodiny v režimu master, pak lokální hodiny volně běží, nebo jsou synchronizovány s externím zdrojem času, např. Global Positioning System (GPS). V (automobilovém) průmyslu mohou být OC také sdruženy se senzory.

### 4.3.2 Boundary clock (BC)

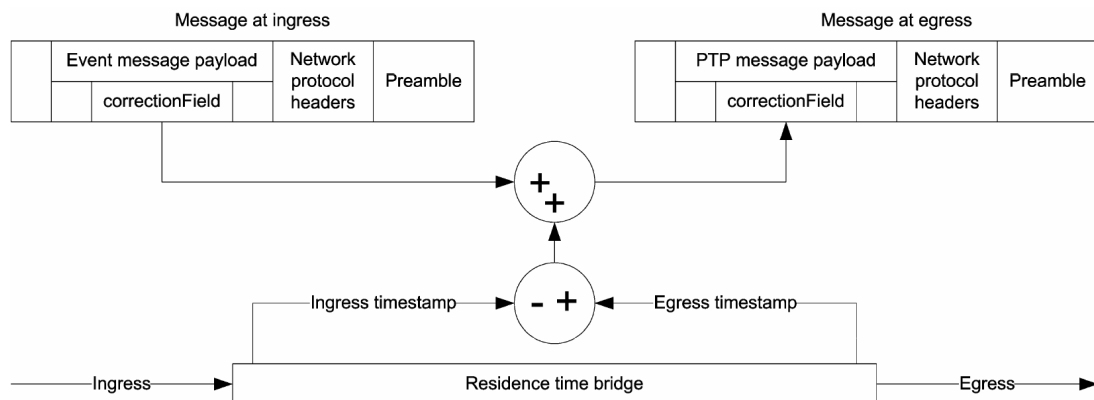
BC má více portů. Maximálně jeden je využíván jako slave a synchronizuje lokální hodiny podle nadřazených hodin, ostatní porty pracují jako master pro podřazené hodiny.

### 4.3.3 Transparent clock (TC)

Pokud máme v síti moc zřetězených prvků za sebou, vnikají při přenosu časových zpráv zpoždění a nepřesnosti v synchronizaci, proto se mezi master hodinami a koncovými hodinami vkládají tzv. transparent clock. Ty sami o sobě nejsou synchronizovány, ale měří dobu, po kterou zpráva putuje přes tyto hodiny a součet je uchováván v tzv. „correctionField“, tento čas je poté použit koncovými hodinami k přesnější synchronizaci nebo po překonfigurování sítě k rychlejší synchronizaci.

### 4.3.4 End-to-end transparent clock

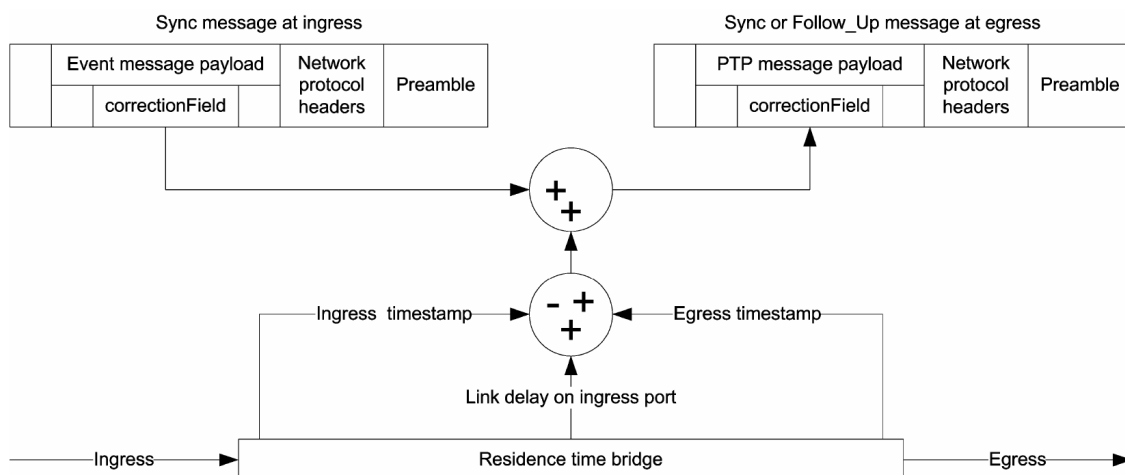
Všechny příchozí zprávy jsou dále přeposílány stejně jako u mostu, směrovače nebo opakováče. U PTP zpráv se počítá čas strávený v tomto zařízení (uchováván v correctionField), který slouží k přesnější synchronizaci (viz obrázek 4.1).



Obrázek 4.1: Přičtení času pobytu v hodinách (převzato z [3])

### 4.3.5 Peer-to-peer transparent clock

Peer-to-peer TC se chovají jako End-to-end TC, ale navíc si předem spočítají zpoždění přenosu na lince každého portu se sousedem (pomocí zpráv Pdelay\_Req, Pdelay\_Resp a Pdelay\_Resp\_Follow\_Up). Toto zpoždění se také přičítá do pole correctionField (viz obrázek 4.2). Na rozdíl od end-to-end TC, který opravuje časy všech PTP časových zpráv, peer-to-peer TC upravuje časy jenom zprávám Sync a Follow\_Up. Tyto časy zpoždění linky slouží také k rychlejší resynchronizaci při překonfigurování sítě, nebo změně trasy časových paketů.



Obrázek 4.2: Přičtení času pobytu v hodinách a zpoždění na lince (převzato z [3])

#### 4.3.6 Management node

Zařízení, které nastavuje a monitoruje hodiny.

### 4.4 Master-slave hierarchie

V master-slave hierarchii je jedno zařízení vybráno jako master, které řídí jedno nebo více zařízení, kterým se říká slave. Master zařízení se vybere buď pomocí algoritmu nebo se role všem zařízením nastaví ručně.

#### 4.4.1 Best Master Clock Algorithmus (BMCA)

Čerpáno z [3].

BMCA slouží k postavení master-slave hierarchie, na každém portu se rozhoduje, které hodiny jsou lepší a podle toho se určí jestli je port slave nebo master. Potenciální grandmaster hodiny vysílají Announce zprávy, pro rozhodnutí, které hodiny jsou nejlepší. Algoritmus se taky používá, když se připojí cizí hodiny (implicitně nastaveny na master), ty začnou vysílat Announce zprávu se svými daty a porovnává se jestli jsou lepší než lokální hodiny a podle toho se zařadí do hierarchie.

BMCA porovnává data podle následujících kritérií:

- Priorita – Nastavitelná hodnota.
- Zdroj času
- Přesnost hodin
- Stabilita hodin

- Identifikační číslo – Pokud se předchozí kriteria schodují.

Podle rozhodnutí se určí v jakém stavu je daný port:

- Master – Port je zdrojem hodin.
- Slave – Hodiny se synchronizují podle hodin připojenými k tomuto portu.
- Passive – Port není master, ani se nesynchronizuje podle připojeného zařízení.

## 4.5 Synchronizace hodin

Čerpáno z [3].

OC a BC se synchronizují pomocí výměnou PTP časových zpráv na komunikační cestě mezi dvěma hodinami viz obrázek 4.3. Synchronizace se dělí na jedno-krokovou a dvou-krokovou. Při jedno-krokové synchronizaci se posílá jen zpráva Sync, která obsahuje čas  $t_1$ , ve který byla zpráva vyslána. Při tom ale je potřebný hardware, který dokáže, aby zpráva Sync obsahovala přesný čas odeslání sama sebe. Jinak se po odeslání zprávy Sync pošle zpráva Follow\_Up, která obsahuje čas odeslání zprávy Sync (dvou-kroková synchronizace).

Odesílání zpráv Sync (a Follow\_Up) probíhá implicitně každé 2s, interval odesílání zpráv Delay\_Req a Delay\_Resp je obvykle mnohem delší, aby nezahlcovaly síť.

Proces výměny zpráv:

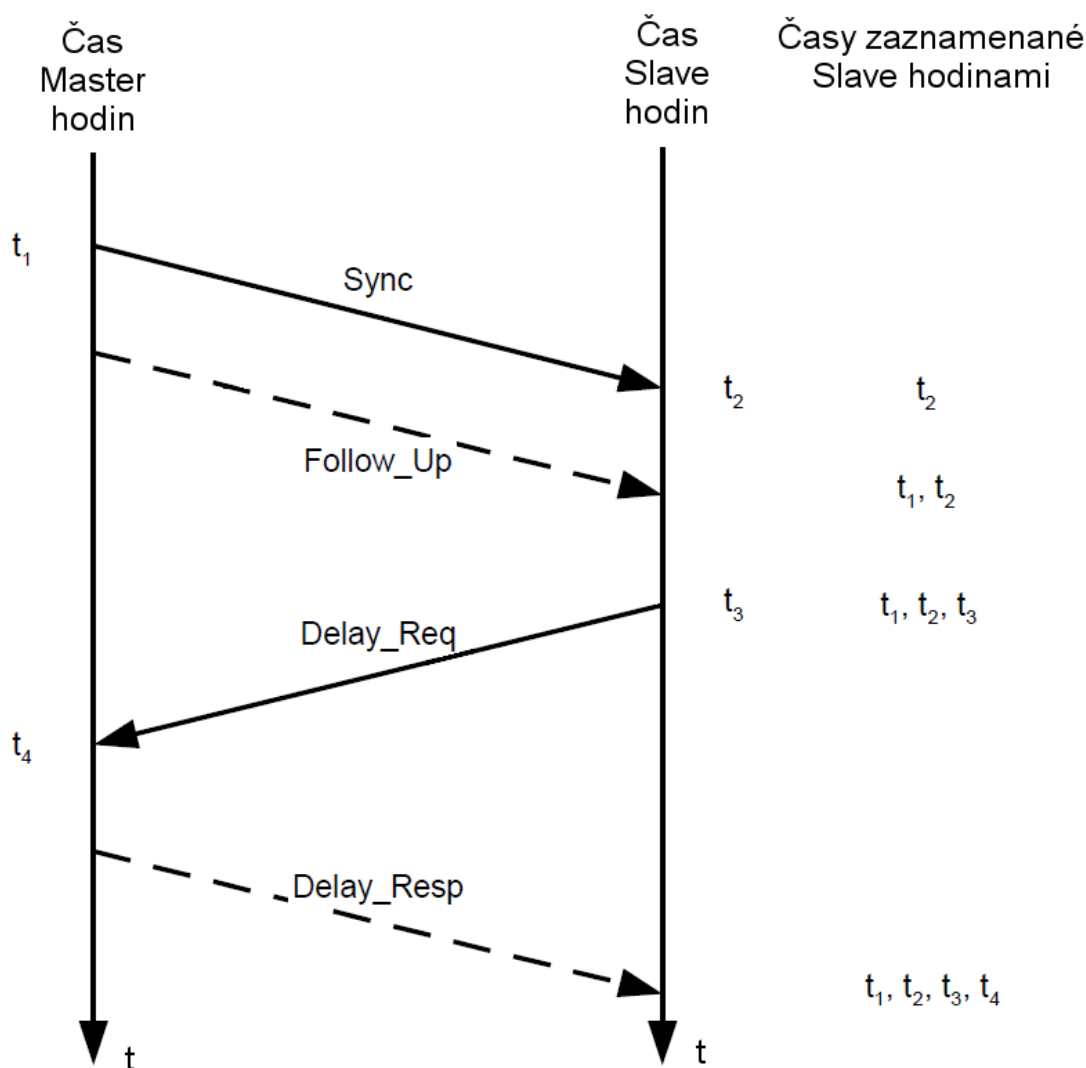
1. Master pošle zprávu Sync a zaznamená čas odeslání  $t_1$ .
2. Slave přijme zprávu Sync a zaznamená čas přijetí  $t_2$ .
3. Master pošle pomocí zprávy Follow\_Up čas  $t_1$  (pokud nebyla obsažena ve zprávě Sync).
4. Slave pošle zprávu Delay\_Req a zaznamená si čas její odeslání  $t_3$ .
5. Master přijme zprávu Delay\_Req a zaznamená si čas přijetí  $t_4$ .
6. Master odešle čas  $t_4$  pomocí zprávy Delay\_Resp.

Na konci, slave zná všechny čtyři časové značky a díky ním dokáže spočítat čas zpoždění přenosu na síti (delay viz 4.1) a hlavně, posun hodin mezi slave a master (offset viz 4.2). Pokud jsou použity TC, musíme do výpočtů zahrnout i časy z pole correctionFiled (viz 4.3.3).

Vzorce pro výpočet delay a offset:

$$delay = \frac{(t_4 - t_1) - (t_3 - t_2)}{2} \quad (4.1)$$

$$offset = \frac{(t_2 - t_1) + (t_3 - t_4)}{2} \quad (4.2)$$



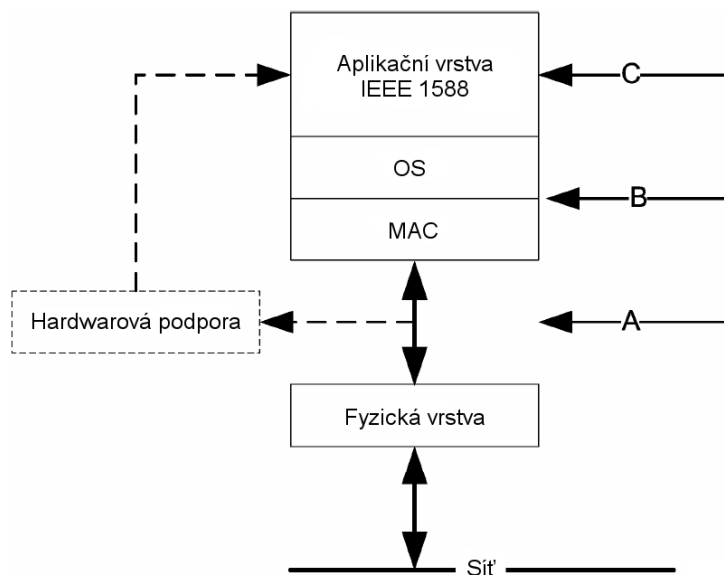
Obrázek 4.3: Přenos zpráv pro synchronizaci master a slave hodin (převzato z [3])

## 4.6 Generování časových značek

Čerpáno z [3].

Časová značka je generovaná při každém odeslání nebo přijetí PTP zprávy události (viz 4.2). Časová značka může být vytvořena na několika místech, během přenosu zprávy z fyzické vrstvy k aplikační vrstvě (viz obrázek 4.4). Značka může být pořízena v aplikační vrstvě (bod C), čas je nejméně přesný, protože zpráva musí projít všemi nižšími vrstvami a tím dochází ke zpoždění. Dalším místem pořízení značky je jádro OS nebo při obsluze přerušení (ISR), bod B. Posledním místem je ve fyzické vrstvě, na rozhraní MII (Media Independent Interface), bod C. Obecně platí, že čím blíže k síti se časová značka vygeneruje, tím je čas přesnější. Pokud se značka generuje ve fyzické vrstvě, je zapotřebí implementovat

hardwarovou podporu (na obrázku 4.4 zobrazeno tečkovanými čarami). Časová značka je poté dopravena do aplikační vrstvy přes vedlejší cestu, mimo vrstvy. Aby se rozpoznalo jaké značky patří ke kterým PTP zprávám, tak hardwarová podpora posílá spolu se značkami taky doplňující informace z PTP časových zpráv.



Obrázek 4.4: Generování časových značek





## Kapitola 5

# Virtual Local Area Network (VLAN)

VLAN, neboli virtuální lokální síť, je technologie, která rozděluje přepínanou síť na menší podsítě, nezávisle na jejím fyzickém rozložení. Můžeme vzít uzly, které jsou připojené k různým fyzickým segmentům a vytvořit z nich menší podsít LAN, oddělenou od ostatních uzlů.

VLAN můžeme vytvořit buď na jednom přepínači (switch), nebo na více. Pokud použijeme více přepínačů, tak pro komunikaci mezi nimi použijeme tzv. „trunk port“ (viz 5.2). VLAN můžou mezi sebou komunikovat stejně jako jiné sítě a to buď pomocí směrovače (routeru), nebo pomocí L3 přepínače<sup>1</sup>.

Na obrázku 5.1 můžeme vidět jednoduchou síť rozdělenou do dvou VLAN.

### 5.1 Zařazení portu do VLAN

V této kapitole a podkapitolách jsem čerpal z [10].

Jakým způsobem a do jaké VLAN se bude port nebo uzel přiřazovat, se nastavuje na přepínači. To je také výhoda VLAN, že při změně konfigurace, nebo topologie sítě nemusíme měnit fyzické části sítě, ale stačí překonfigurovat přepínače. Ve výchozím nastavení existuje VLAN 1, která nejde zrušit nebo smazat a jsou v ní zařazeny všechny porty. Členství ve VLAN se určí podle:

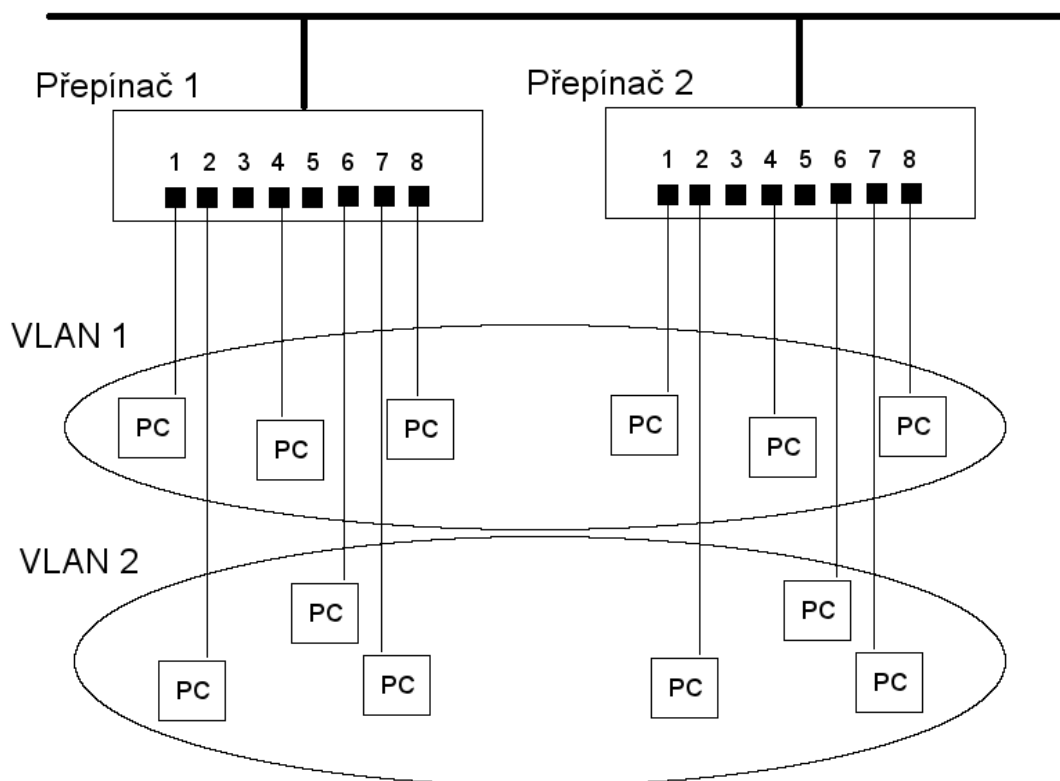
- Portů
- MAC adres
- Protokolu

#### 5.1.1 Členství podle portů

Jedná se o nejjednodušší a nejvíce používanou metodu. Každý port přepínače se ručně nakonfiguruje, do jaké VLAN bude patřit. Portem poté bude moct procházet jenom komunikace, která patří do stejné VLAN. V automobilu, např. při opravě, nebo výměně poškozeného

---

<sup>1</sup>Přepínač, který pracuje na třetí vrstvě OSI modelu.



Obrázek 5.1: Rozdělení sítě do dvou VLAN

přepínače, by se muselo dát pozor, abychom koncové uzly zapojily zpět do správných portů, jinak by se např. senzor vyskytl ve špatné VLAN a nemohli bychom zpracovávat jeho data.

### 5.1.2 Členství podle MAC adresy

Uživatel se přiřadí do určité VLAN podle své MAC adresy, čili administrátor musí na přepínači nastavit pro každý koncový uzel (MAC adresu) do jaké VLAN patří. Přepínač pak uchovává tabulku MAC adres a do jakých VLAN patří. Výhodou je, že když koncový uživatel změní port, tak pořád bude patřit do své VLAN. Nevýhodou je, že uživatel si může svojí MAC adresu změnit softwarově v operačním systému. V automobilovém průmyslu, by mohl nastat problém, při výměně např. poškozeného senzoru za nový, čili s novou MAC adresou, pak by se muselo na přepínačích přepsat tabulka. Tam by ale z bezpečnostních důvodů neměl být přístup.

### 5.1.3 Členství podle protokolu

Uživatel se zařadí do VLAN podle protokolu ze třetí vrstvy OSI modelu, nebo podle IP adresy. Jelikož přepínač musí zkoumat paket na úrovni třetí vrstvy, tak se komunikace zpomaluje a proto tahle metoda není příliš rozšířena.

## 5.2 Komunikace v rámci VLAN

Přepínač si pamatuje které porty patří do které VLAN a povoluje komunikaci jen mezi porty ve stejné VLAN. V reálné situaci ale většinou máme více přepínačů a potřebujeme aby věděli o všech VLAN v síti a dokázali správně přeposílat přijaté rámce. K přenesení informace, v jaké VLAN se nachází daný rámec slouží standard IEEE 802.1Q. Ten dokáže odchozí rámec označkovat (tagging) na tzv. trunk portu<sup>2</sup> a posílá rámec do druhého přepínače přes trunk link<sup>3</sup>.

### 5.2.1 IEEE 802.1Q tagging

Čerpáno z [4].

Při odeslání rámce z jednoho přepínače na druhý dojde na trunk portu k rozbalení rámce, přidání informace o VLAN (tagging) a protože došlo k modifikaci rámce, tak se musí znova přepočítat kontrolní součet. Tag má velikost 4B a vkládá se za zdrojovou MAC adresu. Na obrázku 5.2 můžeme vidět originální a otagovaný Ethernet rámec.

Originální L2 Ethernetový rámec:

6B	6B	2B	64B – 1500B	4B
DA – cílová adresa	SA – zdrojová adresa	Typ/délka	Data	FCS (kontrolní součet)

Otagovaný rámec:

6B	6B	4B	2B	64B – 1500B	4B
DA – cílová adresa	SA – zdrojová adresa	802.1Q tag	Typ/délka	Data	FCS (kontrolní součet)

802.1Q tag:

2B	3b	1b	12b
TPID - 0x8100	Priorita (802.1p)	CFI	VLAN ID

Obrázek 5.2: Originální Ethernet rámec a označený rámec hlavičkou 802.1Q

Hlavička 802.1Q tagu obsahuje:

- Tag Protocol ID (TPID) – TPID má hodnotu 0x8100 a značí, že se jedná o tagovaný rámec a že následující dva bajty nesou informace o VLAN.

<sup>2</sup>Port, na kterém dochází k označkování rámce.

<sup>3</sup>Linka mezi dvěma trunk porty.

- Priorita (Priority Code Point – PCP) – Říká jakou má rámec prioritu. Ta může nabývat osmi hodnot 0 – 7 (7 – nejvyšší priorita).
- Canonical Format Indicator (CFI) – CFI indikuje jestli je MAC adresa v kanonickém formátu nebo indikuje přítomnost RIF (routing information field) pole u Token Ring rámců.
- VLAN ID (VID) – indikuje, do jaké VLAN rámec patří. Můžeme mít celkem 4096 ( $2^{12}$ ) VLAN. VID 0x000 značí, že rámec nepatří do žádné VLAN a nese jenom informaci o prioritě. Hodnota 0xFF F je rezervovaná. Takže můžeme použít jen 4094 různých VLAN.

### 5.3 Směrování mezi VLAN

Čerpáno z [10].

K VLAN se chováme stejně jako k fyzicky odděleným podsítím. Pokud chceme směrovat mezi VLAN musíme z přepínače vyvést každou VLAN a připojit ke směrovači. Tím ale zbytečně plýtváme porty, proto můžeme použít směrovač který má také trunk port a dokáže číst 802.1Q tag, nebo můžeme použít L3-switch. To je přepínač, který dokáže pracovat na třetí vrstvě OSI modelu a je rychlejší než klasický směrovač.

### 5.4 Využití VLAN v automobilovém průmyslu

Jedno z použití VLAN v automobilu je například oddělení jednotlivých domén (informační, bezpečnostní, diagnostika, pohon atd.). Je to bezpečnostní opatření, aby se potenciální útočník nemohl přes diagnostiku dostat do jiných domén.

Jiná výhoda VLAN je, že může zjednodušit fyzické rozložení sítě. Pokud bychom měli například přepínač, který by spojoval kamery, ze všech čtyř stran automobilu a nacházel by se v přední části, tak je zbytečné vést kabel ze zadní kamery přes celý automobil. Místo toho můžeme zadní kameru připojit na bližší přepínač ale do správné VLAN, ve které leží i ostatní kamery.

Další využití je priorita rámce. Jelikož automobil a jízda v něm musí být převážně bezpečná, tak si nemůžeme dovolit, aby docházelo ke zpoždění doručení rámců nesoucí kritická data a informace například z kamer monitorující okolí automobilu. Rámce s vyšší prioritou by pak měly na přepínači přednost.

## Kapitola 6

# Quality of Service (QoS)

Quality of Service je termín, který zahrnuje různé metody pro zajištění parametrů provozu na síti. Při větším zatížení sítě může nastat situace, že přepínač nestíhá odesílat rámce. V kritických situacích nebo aplikacích si ale nemůžeme dovolit velké zpoždění paketů. Typické je přenos videa a audia, v automobilu ale máme taky spoustu senzorů, kamer a radarů monitorující okolí automobilu a pokud nastane kritická situace, informace musí do řídicí jednotky přijít co nejrychleji.

### 6.1 Class of Service (CoS)

Čerpáno z [4].

Jedna z metod, pro zajištění kvality provozu, používá Class of Service (CoS). Jsou to tři bity v hlavičce 802.1Q tagu, určující prioritu rámce. CoS má 8 hodnot priority 0-7 (7 je nejvyšší priorita). V Tabulce 1 můžeme vidět hodnotu PCP (viz sekce 5.2.1), jeho prioritu a typ provozu. PCP s hodnotou 1 má nižší prioritu než PCP s hodnotou 0. PCP s hodnotou 0 je výchozí hodnota (např. pro neotagované rámce). Také to umožňuje nastavit prioritu nižší než je výchozí hodnota.

PCP	Priorita	Akronym	Typ provozu
1	0 (nejnižší)	BK	Background
0 (výchozí)	1	BE	Best Effort
2	2	EE	Excellent Effort
3	3	CA	Critical Applications
4	4	VI	Video, < 100ms zpoždění a jitter
5	5	VO	Voice, < 10ms zpoždění a jitter
6	6	IC	Internetwork Control
7	7 (nejvyšší)	NC	Network Control

Tabulka 6.1: Úrovně priority podle 802.1Q-2005 (převzato z [4])

## 6.2 Třídění rámců do front

Čerpáno z [4].

Každý port přepínače obsahuje několik front (Queue), do kterých se řadí příchozí rámce podle priority. Fronty s vyšší prioritou mají při vysílání rámců přednost před frontami s nižší prioritou. Když přijde otagovaný rámec, přepínač podle jeho priority a podle mapovací tabulky rozhodne do jaké fronty se zařadí. Pokud jsou fronty přeplněné, mohou být rámce zahozeny. Přeplnění nižších front by nemělo mít vliv na fronty s vyšší prioritou a ty by měly pracovat stále správně.

Můžeme mít různý počet front. V Tabulce 6.2 je uvedeno 8 doporučených způsobů rozdělení front podle 802.1Q-2005.

PCP	Priorita	Třídy provozů							
		1	2	3	4	5	6	7	8
1	0 (nejnižší)	0	0	0	0	0	0	0	0
0 (výchozí)	1	0	0	0	0	0	1	1	1
2	2	0	0	0	1	1	2	2	2
3	3	0	0	0	1	1	2	3	3
4	4	0	1	1	2	2	3	4	4
5	5	0	1	1	2	2	3	4	5
6	6	0	1	2	3	3	4	5	6
7	7	0	1	2	3	4	5	6	7

Tabulka 6.2: Doporučené mapování priorit a tříd provozů (převzato z [4])

V automobilu nebudeme potřebovat velké množství front, protože dokážeme předvídat chování různých domén a známe časy posílání informací senzorů apod., proto může výrobce přizpůsobit komunikaci tak, aby nedocházelo k hromadění rámců, nebo zahlcení šířky pásma. Jediné, co nedokážeme předvídat, jsou různé podněty a situace z okolí automobilu. Kvůli zajištění bezpečnosti bychom tato data bychom měli posílat k vyhodnocení do centrální jednotky přes fronty s vyšší prioritou.

## Kapitola 7

# Audio Video Bridging

Audio Video Bridging (AVB) je společný název pro skupinu IEEE standardů zabývajících přenosem audia, videa a ostatních časově závislých informací přes LAN síť. Ve vozidlech se AVB může využít pro přenos audia a videa pro zábavu, dat z radarů nebo z kamer monitorujících okolí automobilu nebo k spolehlivému a rychlému přenosu kritických dat ze senzorů.

AVB pracuje na principu zarezervování určité části pásma mezi dvěma uzly. Pakety, posílané touto linkou, mají přednost před ostatním přenosem, takže nedochází ke kolizím. AVB pro svojí funkci využívá následujících standardů:

- IEEE 802.1BA: Audio Video Bridging Systems
- IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications
- IEEE 802.1Qat: Stream Reservation Protocol
- IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams

Tyto standardy pro AVB zajišťují např. časovou synchronizaci, pro přesnější časy doručení paketů a zmenšení zpoždění (po zapnutí hudby tak může pravý, levý a zadní reproduktory hrát zároveň), kontrolu dostatečné kapacity pásma a její rezervaci, identifikaci zařízení, které nepodporují AVB technologii, nebo zajišťují řízení provozu (traffic shaping).

### 7.1 IEEE 802.1Qat: Stream Reservation Protocol (SRP)

Čerpáno z [5].

Než dojde k přenosu paketů, musí se na zúčastněných zařízeních a koncových uzlech zaregistrovat a zarezervovat pásmo, pokud je volné. Pokud se povede zarezervovat pásmo na všech zařízeních mezi zdrojem a cílem, vytvoří se mezi nimi kanál, přes který se budou posílat rámce. Tomuto rezervovanému kanálu se říká „stream“. K registraci a rezervaci slouží SRP, který k tomu používá tři protokoly, MMRP, MVRP a MSRP.

Multiple MAC Registration Protocol (MMRP) slouží k řízení propagace registračních zpráv z vysílače přes přepínanou síť. Zdroj a cíl je určen MAC adresou.

Multiple VLAN Registration Protocol (MVRP) je použit koncovými uzly a mosty k deklarování členství ve stejné VLAN jako je zdroj streamu. To umožňuje propagování a používání Data Frame Priority (k vygenerování Priority Code Point, viz 5.2.1) od Talker (zdroj dat) k Listener(s) (příjemce dat) v otagovaných rámcích.

Multiple Stream Registration Protocol (MSRP) je protokol, který poskytuje koncové uzly se schopností rezervace síťových prostředků a garantování odeslání a přijetí dat přes síť s požadovaným QoS (Quality of Service). Tyto koncové uzly dělíme na:

- Talker – Zařízení, které zprostředkovává a posílá data.
- Listener(s) – Jedno či více zařízení, která přijímají a zpracovávají data.

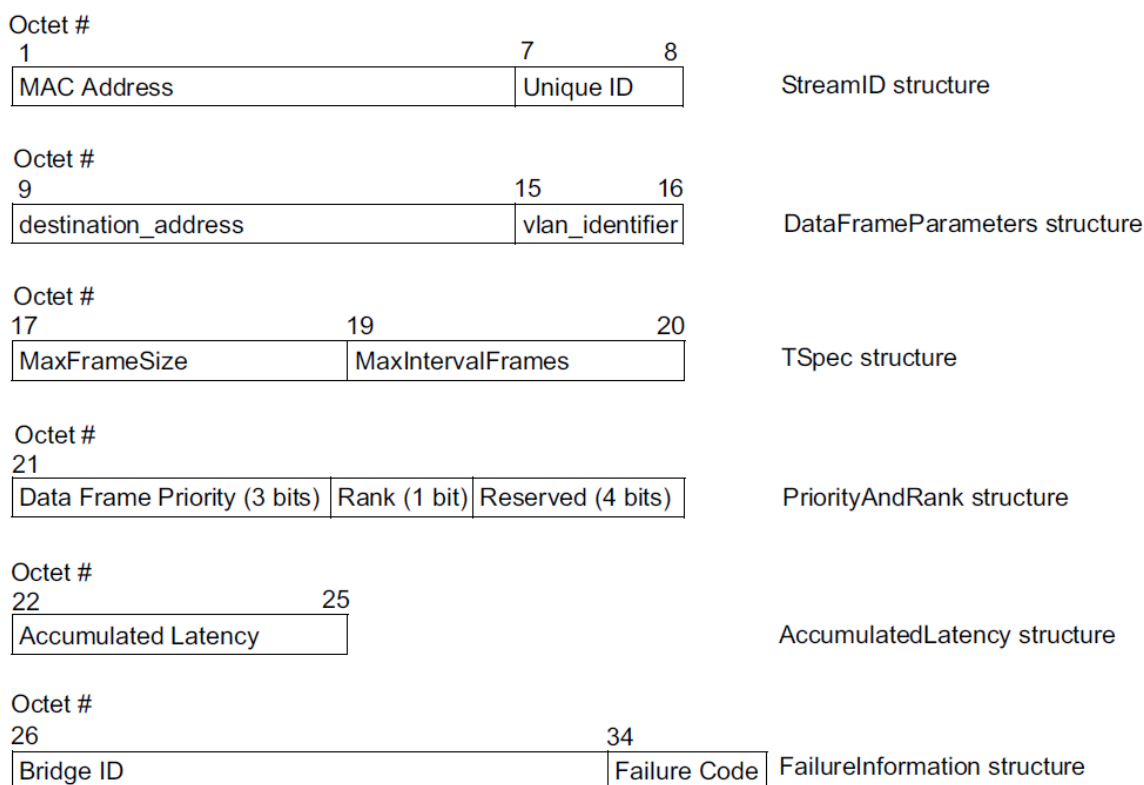
Talker posílá tzv. „talker advertise“ zprávu, která obsahuje atributy, které charakterizují přenosový kanál. Zpráva obsahuje:

- StreamID – MAC adresa zdroje + UniqueID.
- Adresa příjemce – Multicast nebo unicast.
- VLAN – Používá MVRP k nastavení VLAN mezi Talker a Listener(s).
- TSpec (Traffic Specification) – Maximální velikost rámce a maximální počet vyslaných rámců během jednoho intervalu.
- Priorita (PCP)
- Rank – 0 (naléhavá data), 1 (obyčejná data)
- Accumulated Latency – Během přechodu mezi mosty si zpráva předběžně počítá zpoždění.
- FailureInformation – Pokud dojde k nějaké chybě, zpráva je změněna z „talker advertise“ na „talker failed“, v poli je zaznamenáno ID mostu, kde došlo k chybě a důvod.

Každý most nebo přepínač, který obdrží „talker advertise“ zprávu si na výstupních portech zkontroluje, jestli má dostatek zdrojů na rezervaci kanálu. Pokud most má dostatek zdrojů, tak přeposílá zprávu dál a připočítává zpoždění, AVB povoluje zpoždění 2 ms a 7 mostů/přepínačů (hopů) mezi Talker a Listener. Když Listener obdrží „talker advertise“ zprávu, tak ví, že je možné rezervovat stream, zná zpoždění na lince a posílá zpět, stejnou cestou, „listener ready“ zprávu. Když most tuto zprávu obdrží, tak si zarezuje požadovanou velikost pásma a zapamatuje si na jaký port bude potom posílat AVB pakety a pošle „listener ready“ zprávu dál směrem k Talker zařízení.

Pokud most nemá dostatek zdrojů nebo pásma, tak MSRP změní „talker advertise“ zprávu na „talker failed“ zprávu, která obsahuje ID mostu, kde došlo k chybě a důvod chyby a pošle tuto zprávu dál, aby Listener věděl, že nedojde k vytvoření kanálu. Stejně tak, se „listener ready“ zpráva změní na „listener asking failed“ zprávu pokud je nedostatek zdrojů nebo pásma.





Obrázek 7.1: Pole obsažena ve zprávách (převzato z [5])

Může nastát případ, kdy máme dvě (nebo více) Listener zařízení, která mají zájem o rezervovaný kanál s Talker. Jeden Listener má dostatek zdrojů a pošle zpět k Talker „listener ready“ zprávu a druhý Listener nemá dostatek zdrojů a posílá „listener asking failed“ zprávu. Na síti existuje přepínač, kde na jednom portu obdrží první zprávu a na jiném portu obdrží druhou zprávu. MSRP spojí tyto zprávy do „listener ready failed“ zprávy a přepoše ji k Talker zařízení. Když Talker obdrží tuto zprávu, tak bude vědět, že jsou zařízení, která mají zájem o stream a mají k tomu dostatek zdrojů a taky, že jsou zařízení, která sice mají zájem, ale nemají dostatek zdrojů nebo pásma. Talker pořád může vytvořit stream a posílat data, ale musí si uvědomit, že ne všechny koncová zařízení (Listeners) dokážou data přijímat.

Talker může také odregistrovat stream a tím zpět uvolnit pásmo.

## 7.2 IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)

V této kapitole a podkapitolách jsem čerpal z [6].

Standard 802.1Qav se zabývá přeposíláním a řazením paketů do front na mostu nebo přepínači. Pokud budou koncová zařízení vysílat pakety rovnoměrně a bude použit standard

802.1Qav, pak nebude docházet k hromadění paketů na zařízeních, celý přenos bude rovnoměrný a deterministický.

FQTSS se skládá ze šesti částí, které jsou potřeba pro AVB provoz.

### 7.2.1 Detekce SRP domény

SRP doména je oblast, která zahrnuje mosty a přepínače, které podporují Stream Reservation Protokol. Pokud zřízení přijímá SRP registrační zprávu obsahující prioritu, pak je pravděpodobné, že vysílací zařízení podporuje SRP a port, který přijal zprávu patří do SRP domény. Postupně se určí, které porty patří do SRP domény, nebo už jsou na hranici.

### 7.2.2 Parametry popisující dostupnost pásma

Pro každý port a pro každou třídu provozu máme 4 parametry:

- `portTransmitRate` – Přenosová rychlost (bit/s), kterou port poskytuje.
- `deltaBandwidth` – Procento přenosové rychlosti (`portTransmitRate`), které může být rezervované frontou, přiřazené k určité třídě provozu.
- `adminIdleSlope` – Šířka pásma (bit/s), kterou požaduje management, aby byla rezervovaná pro frontu, určité třídy provozu. Pokud je použit SRP, tento parametr nemá žádný efekt, pokud SRP není použit, pak se `operIdleSlope` rovná `adminIdleSlope`.
- `operIdleSlope` – Šířka pásma (bit/s), která je právě rezervovaná frontou. Tato hodnota je použita „credit-based shaper“ algoritmem<sup>1</sup> jako `idleSlope` pro danou frontu.

### 7.2.3 Vztahy mezi parametry

Doporučená výchozí hodnota `deltaBandwidth` pro nejvyšší třídu provozu je 75% (to znamená, že 25% pásma zbývá pro nerezervovaný provoz) a pro ostatní, nižší třídy provozu je doporučená výchozí hodnota 0%. Celkové procento pásma, které je možné rezervovat na portu pro danou frontu a fronty s vyšší prioritou, je součet `deltaBandwidth` dané fronty a `deltaBandwidth` front s vyšší prioritou. Pro frontu s nejvyšší prioritou, maximální hodnota `operIdleSlope` je `deltaBandwidth % z portTransmitRate`. Pokud je ale aktuálně používané pásmo `operIdleSlope` menší než maximální hodnota, tak fronty s nižší prioritou mohou zbývající nepoužívanou šířku pásma používat dokud ji fronta s vyšší prioritou nebude zase potřebovat. Např. máme dvě fronty s prioritou 3 (SR třída<sup>2</sup> A) a 2 (SR třída B). Fronta s vyšší prioritou má `deltaBandwidth(3) = 20%` a fronta s nižší prioritou má `deltaBandwidth(2) = 30%`. Pokud je `operIdleSlope(3) = 10%`, tak zbývajících 10% může použít fronta 2, takže maximální hodnota `operIdleSlope(2) = 40%` (do té doby dokud fronta 3 nebude potřebovat všech 20%).

---

<sup>1</sup> Algoritmus pro řízení provozu na síti, založený na principu kreditu, viz sekce 7.3.

<sup>2</sup> Stream Reservation (SR) třída : Třída provozu, jejichž pásmo může být rezervováno pro AV přenos. Každé třídě se přiřadí priorita. Třídy se označují písmeny, ve výchozím nastavení máme 2 třídy, A a B.

#### 7.2.4 Správa parametrů

Pokud je podporován SRP a jeho funkce, pak hodnota `operIdleSlope` závisí na SRP. Pokud SRP není podporováno, pak hodnota `operIdleSlope` se rovná hodnotě požadované managementem (`adminIdleSlope`).

Hodnotu `portTransmitRate` a `deltaBandwidth` můžeme pomocí managementu změnit, ale v každém případě může dojít k následujícím důsledkům:

- Součet hodnot `operIdleSlope` pro port může překročit celkové rezervované pásmo pro daný port, nebo hodnota `operIdleSlope` pro danou frontu může překročit rezervované pásmo pro danou frontu. Následkem toho, se mohou vyskytovat aktivní streamy na portu, kterou už nemohou být podporované.
- Na portu jsou neaktivní streamy, ale v důsledku zvýšení šířky pásma a její dostupnosti, mohou teď být streamy podporované.
- Aktivní streamy, které jsou stále podporované můžou po změně také změnit garantované zpoždění.

Pokud nastane jakýkoliv z těchto případů, management nebo SRP by měl zajistit opravu parametrů a překontrolování podpory streamů na portu. K naznačení, že došlo ke změně týkající se šířky pásma, je generován signál „`bandwidthAvailabilityChanged`“ pro každý port vždy, když dojde ke změně parametrů `portTransmitRate` nebo `deltaBandwidth`. Tento signál spouští překontrolování streamů, které můžou být rezervované na portu.

#### 7.2.5 Závislost aktuální šířky pásma na velikosti MAC service data unit (MSDU)

Pro výpočet aktuálně potřebné šířky pásma pro stream se předpokládá konstantní velikost dat a rychlost přenosu. Během průchodu dat vrstvami a podvrstvami se ale data obalují různými metadaty, jako hlavička protokolů apod.

SRP používá pole `TSpec` (Traffic Specification, viz 7.1), které definuje maximální velikost rámce (`maxFrameSize`) a maximální počet vyslaných rámců během jednoho intervalu (`maxIntervalFrames`) pro stream. `Tspec` nepočítá s `overhead`<sup>3</sup> rámce spojený s přenosem `MSDU`<sup>4</sup>, který zvětšuje velikost `MSDU` a tedy celého rámce. Když ale SRP vyhodnocuje velikost hodnoty `operIdleSlope` pro danou frontu, musí také počítat s `overhead` rámce, který je přidán k `MSDU` na portu. `Overhead` můžeme předvídat, máme minimálně tyto zdroje:

- VLAN a bezpečnostní tagy
- `Overhead` při vytváření rámce MAC službou
- `Overhead` fyzické vrstvy (hlavička preamble apod.)

---

<sup>3</sup>Overhead – metadata, která obalují data z vyšších vrstev (payload). Např. hlavička, preamble, SoD apod.

<sup>4</sup>MSDU – data z podvrstvy LLC (část 2. vrstvy OSI modelu) a IP paketu (data z vyšších vrstev) předtím než se zabalí a pošlou do fyzické vrstvy.

Pásmo, která je potřebná pro daný stream se spočítá jako:

$$\mathit{maxFrameRate} = \mathit{maxIntervalFrames} * \left( \frac{1}{\mathit{classMeasurementInterval}} \right)$$

$$\mathit{assumedPayloadSize} = \mathit{maxFrameSize}$$

$$\mathit{actualBandwidth} = (\mathit{perFrameOverhead} + \mathit{assumedPayloadSize}) * \mathit{maxFrameRate}$$

- $\mathit{maxFrameRate}$  – Určuje frekvenci vysílání rámců.
- $\mathit{classMeasurementInterval}$  – Interval pro SR třídu A je 125us a pro třídu B 250us.
- $\mathit{assumedPayloadSize}$  – Velikost payload<sup>5</sup> se zvolí jako maximální.
- $\mathit{perFrameOverhead}$  – Overhead rámce.

### 7.2.6 Mapování priorit k třídám provozů

Mapování priorit má některé omezení:

- Priority, které se shodují se SR třídami jsou mapovány na třídy provozů, které podporují credit-based shaper algoritmus (viz tabulka 1,2).
- Třídy provozů, které podporují credit-based shaper algoritmus mají vyšší prioritu než třídy provozů podporující jiné algoritmy pro řízení provozu.
- Existuje aspoň jedna třída provozů, která podporuje credit-based shaper algoritmus a aspoň jedna třída podporující algoritmy pro řízení provozu jen podle priorit.

		Dostupné třídy provozů							
PCP	Priorita	2	3	4	5	6	7	8	
1	0 (nejnižší)	0	0	0	0	0	0	0	
0 (výchozí)	1	0	0	0	0	0	0	1	
2	2	1	1	2	3	4	5	6	
3	3	1	2	3	4	5	6	7	
4	4	0	0	1	1	1	1	2	
5	5	0	0	1	1	1	2	3	
6	6	0	0	1	2	2	3	4	
7	7	0	0	1	2	3	4	5	

Tabulka 7.1: Doporučené mapování pro SR třídy A (priorita 3) a B (priorita 2) (převzato z [6])

<sup>5</sup>Payload – data obdržaná z vyšších vrstev, která se pak zabalí a pošlou dále.

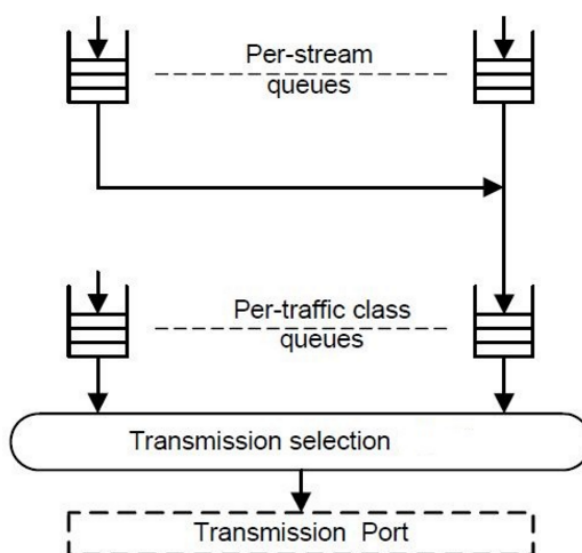
PCP	Priorita	Dostupné třídy provozů							
		2	3	4	5	6	7	8	
1	0 (nejnižší)	0	0	0	0	0	0	0	0
0 (výchozí)	1	0	0	0	0	0	1	1	
2	2	1	2	3	4	5	6	7	
3	3	0	0	0	1	1	2	2	
4	4	0	1	1	2	2	3	3	
5	5	0	1	1	2	2	3	4	
6	6	0	1	2	3	3	4	5	
7	7	0	1	2	3	4	5	6	

Tabulka 7.2: Doporučené mapování jen pro SR třídu B (priorita 2) (převzato z [6])

### 7.2.7 Talker

Talker používá priority, aby přepínače mohly rozpoznat, do které SR třídy rámec patří. Talker spolu s přepínači, které vedou k Listener, musí mít přehled o požadované šířce pásma a společná pravidla pro její rezervaci. O tyto pravidla a požadavky se stará SRP.

Talker posílá rámce do front rozdělených podle streamů, do kterých rámce patří. Rámce se do front posílají podle Tspec parametru streamu. Během classMeasurementInterval (viz 7.2.5) se může poslat maximálně maxIntervalFrames rámců, menších než maxFrameSize, do fronty svého streamu. Pomocí credit-based shaper algoritmu se rámce posílají do dalších front podle priority, kterou stream používá a z nich se už rámce posílají do dalšího přepínače.



Obrázek 7.2: Model front pro Talker stanici (převzato z [6])

### 7.2.8 Listener

Listener má z úkol příjem rámců. Musí stíhat je přijímat, ukládat a zpracovávat.

## 7.3 Credit-based shaper algoritmus

Čerpáno z [6].

Credit-based shaper algoritmus zajišťuje rovnoměrné vysílání rámců na portu. Rámce vysílá podle dostupného kreditu (pásma), pokud má k dispozici kredit, rámce se vysílají, pokud ne, rámce musí počkat dokud se kredit nedobije.

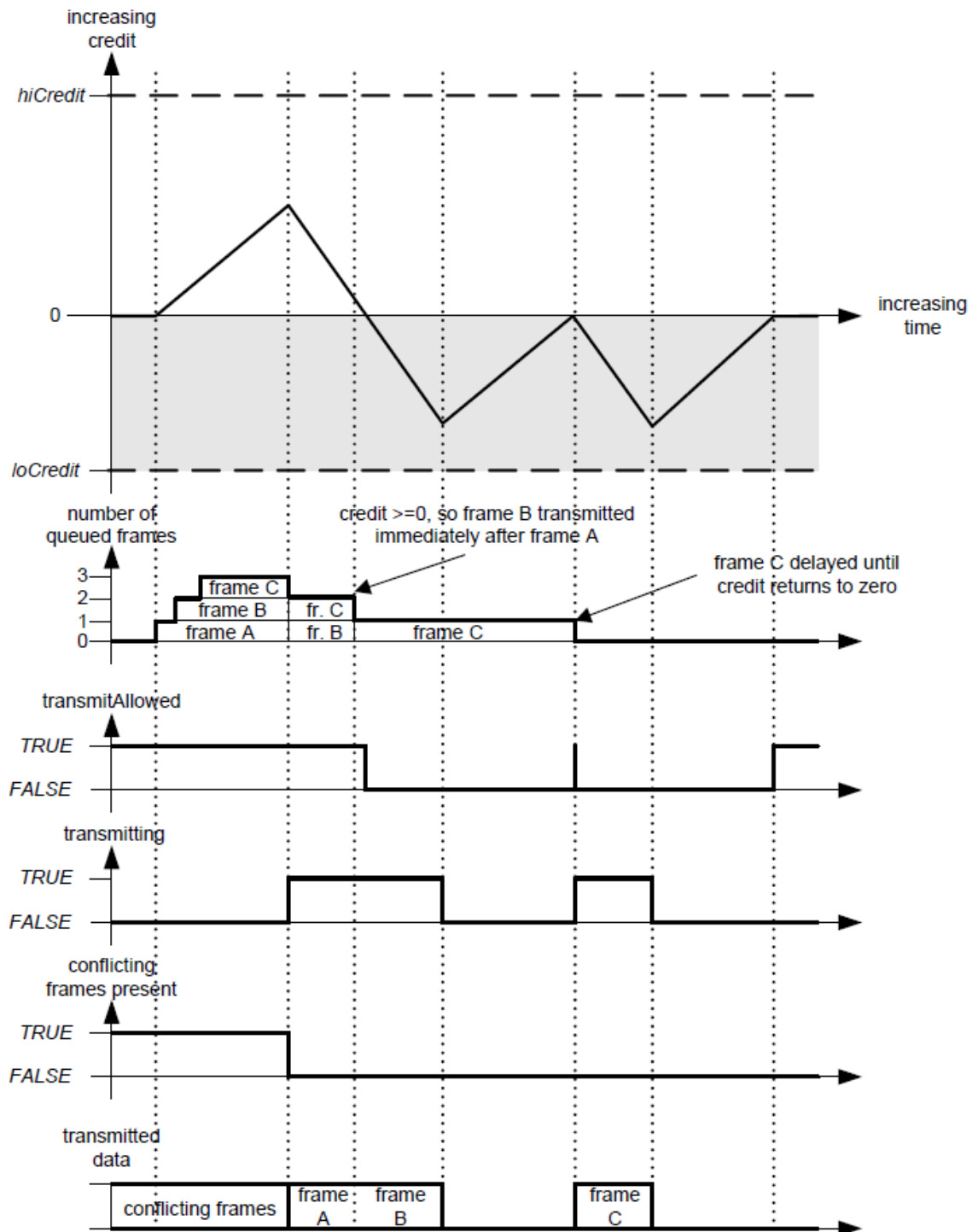
Algoritmus používá následující parametry:

- `portTransmitRate` – Přenosová rychlost v bit/s.
- `idleSlope` – Rychlost nabývání kreditu (bit/s) (např. když v době kdy se nevysílají rámce). Hodnota `idleSlope` nemůže být větší než hodnota `portTransmitRate`. Hodnota `idleSlope` se nastaví podle hodnoty parametru fronty `operIdleSlope`.
- `transmit` – Hodnota je nastaven na `TRUE`, když se vysílá rámeček, `FALSE`, když je rámeček odeslán.
- `kredit` – Kredit (v bitech), který je dostupný pro frontu. Pokud ve frontě nejsou žádné rámce, `transmit = false` a aktuální kredit je kladný, pak je kredit nastaven na nulu.
- `sendSlope` – Rychlost spotřeby kreditu (bit/s) (např. když v době kdy se vysílají rámce). `SendSlope` je definován jako:

$$sendSlope = idleSlope - portTransmitRate$$

- `transmitAllowed` – Nabývá hodnoty `TRUE`, když je kredit nezáporný a `FALSE`, když je kredit záporný.

Na obrázku 7.3 můžeme vidět jak credit-based shaper algoritmus pracuje. Máme tři rámce (A, B, C) ve frontě, které čekají až se odešlou konfliktní rámce (např. rámce s vyšší prioritou) a kredit se zvyšuje rychlostí `idleSlope`. Když se konfliktní rámce odešlou, může se poslat rámeček A, protože po jeho odeslání je kredit kladný, může se odeslat rámeček B. Nyní je kredit záporný, proto rámeček C musí počkat dokud se kredit nebude rovnat nule. Pak je odeslán i rámeček C a kredit se dobije. Jelikož nemáme ve frontě žádný další rámeček, hodnota kreditu zůstane nula.



Obrázek 7.3: Credit-based shaper algoritmus (převzato z [6])





## Kapitola 8

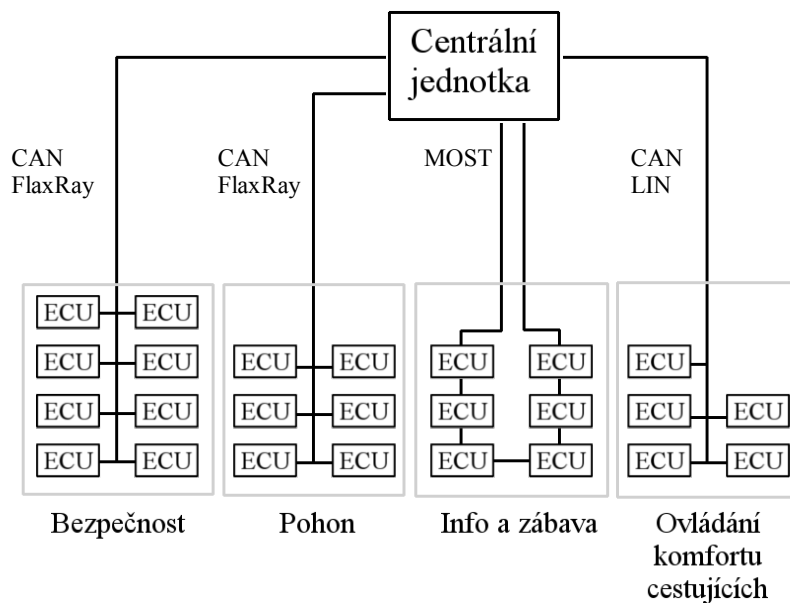
# Návrh struktury sítě

V dnešních automobilech se setkáme s následujícími standardy (podle [8])

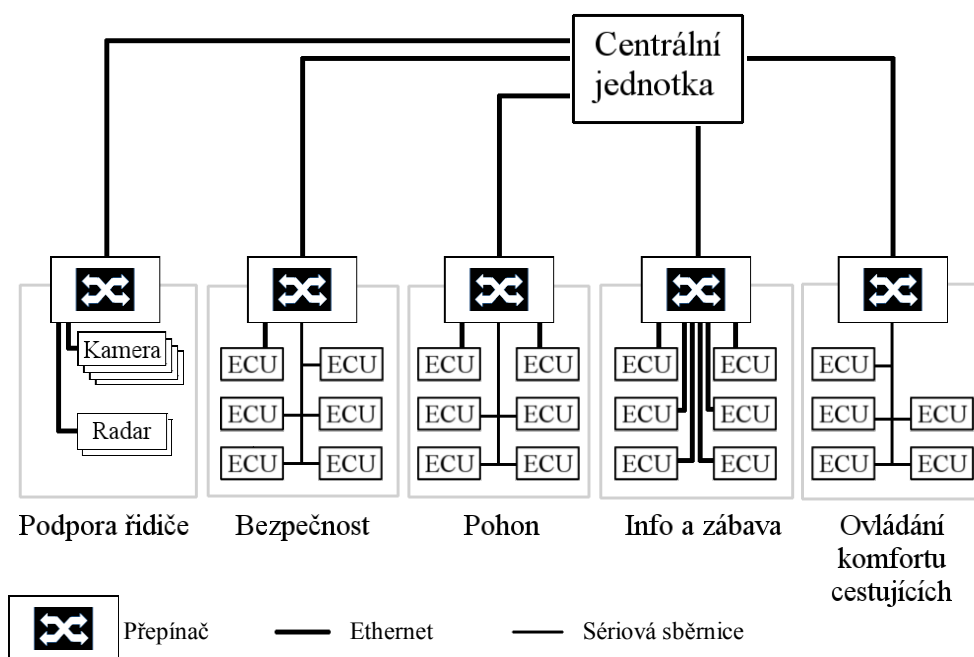
- CAN (Controller Area Network) – CAN je sdílená sériová sběrnice s přenosovou rychlostí až 1Mb/s. Výhodou je její nízká cena a spolehlivost. Nevýhodou je její poměrně malá přenosová rychlost. CAN se používá v oblastech pohonu, podvozku a k ovládání elektroniky zajišťující komfort.
- LIN (Local Interconnect Network) – LIN je sériová sběrnice s master-slave architekturou. Přenosová rychlost je pouze 19,2kb/s. Výhodou je nižší cena než u CAN. LIN se používá hlavně pro ovládání komfortu (ovládání zrcadel, sedadel, oken a dalšího elektrického příslušenství), které nepotřebuje velkou přenosovou rychlost.
- FlexRay – FlexRay je sdílená sériová sběrnice s přenosovou rychlostí 10Mb/s. Nevýhodou je větší cena než u CAN. Výhodou je větší přenosová rychlost, díky tomu je použit v časově kritičtějším oblastech jako pohon a bezpečnost.
- MOST (Media Oriented Systems Transport) – MOST je standard s kruhovou topologií a přenosovou rychlostí až 150Mb/s. Přenosové médium může být optické vlákno, UTP nebo koaxiální kabel. S porovnáním s ostatními standardy, má MOST velkou přenosovou rychlost, ale nevýhodou je vysoká cena, proto se používá hlavně k přenosu audia a videa.
- LVDS (Low Voltage Differential Signaling) – LVDS má za úkol nahradit MOST. Výhodou je vyšší přenosová rychlost a nižší cena. Nevýhodou je, že se jedná o sběrnici point-to-point, takže ke každé kameře nebo video zařízení vede zvlášť sběrnice.

### 8.1 Nasazení Ethernetu

Nasazením Ethernetu v automobilu se přejde z heterogenní sítě s proprietárními protokoly na homogenní Ethernetovou síť s hierarchickou strukturou. Na následujících obrázcích je zobrazena struktura sítě v dnešních automobilech a návrh struktury s použitím Ethernetu v budoucnu.



Obrázek 8.1: Struktura sítě v dnešních vozidlech



Obrázek 8.2: Struktura sítě s použitím Ethernetu a přepínačů

Obrázek 8.1 ukazuje strukturu sítě v dnešních automobilech. Na Obrázku 8.2 je použit Ethernet s prepínači. Ethernet je použit jako páteřní síť a propojuje jednotlivé domény mezi sebou a s centrální jednotkou. Také je použit pro přenos videa a audia nebo v časově kritických oblastech. V rámci domén, jednotlivé ECU<sup>1</sup>, které nepotřebují rychlý přenos dat nebo zaručené zpoždění, jsou propojené sériovou sběrnici, např. CAN (Ethernet je point-to-point linka, takže ke každému ECU bychom potřebovali zvlášť fyzické médium a port na prepínači např. místo toho aby ke každému oknu vedlo zvlášť ethernetový kabel, stačí použít jednu sběrnici).

## 8.2 Vysílače/přijímače a prepínače

V době psaní této práce jsem našel jen dvě společnosti, které nabízeli vysílače/přijímače (tzv. transceiver) a prepínače – Broadcom a NXP. Některé čipy jsou hotové a některé jsou teprve v přípravě. Čipy nejsou ani v prodeji pro veřejnost ani nejsou zveřejněné katalogové listy (datasheet), kromě čipu TJA1100 od NXP. Níže je seznam (podle [2] a [9]) vysílačů a přijímačů a prepínačů podporující technologii BroadR-Reach.

Broadcom:

- BCM89810 – Vysílač/přijímač firmy Broadcom integrovaný v CMOS čipu. Podporuje všechny funkce fyzické vrstvy standardu BroadR-Reach a přenos dat přes jeden pár kroucené dvojlinky rychlostí 100Mb/s. Je určený pro koncová zařízení jako kamery, radar apod.
- BCM89610 – Vysílač/přijímač podporující funkce fyzické vrstvy 10BASE-T, 100BASE-TX a 1000BASE-T Ethernetu. Podpora kabelu UTP Cat5 a konektoru RJ45. Vhodné hlavně k připojení externích zařízení, např diagnostiky.
- BCM89200 – Čtyřportový prepínač firmy Broadcom pro automobilový průmysl se dvěma integrovanými BR porty. Stejně jako ostatní prepínače od Broadcom podporuje funkce časových protokolů IEEE 1588 a 802.1AS, peer-to-peer a end-to-end transparentní hodiny a AVB.
- BCM89500 – Sedmiportový prepínač se čtyřmi integrovanými BR porty.
- BCM89501 – Sedmiportový prepínač s pěti integrovanými BR porty.
- BCM89811 – Vysílač/přijímač. V přípravě.
- BCM8923x, BCM8953x – Prepínače, v přípravě.

NXP:

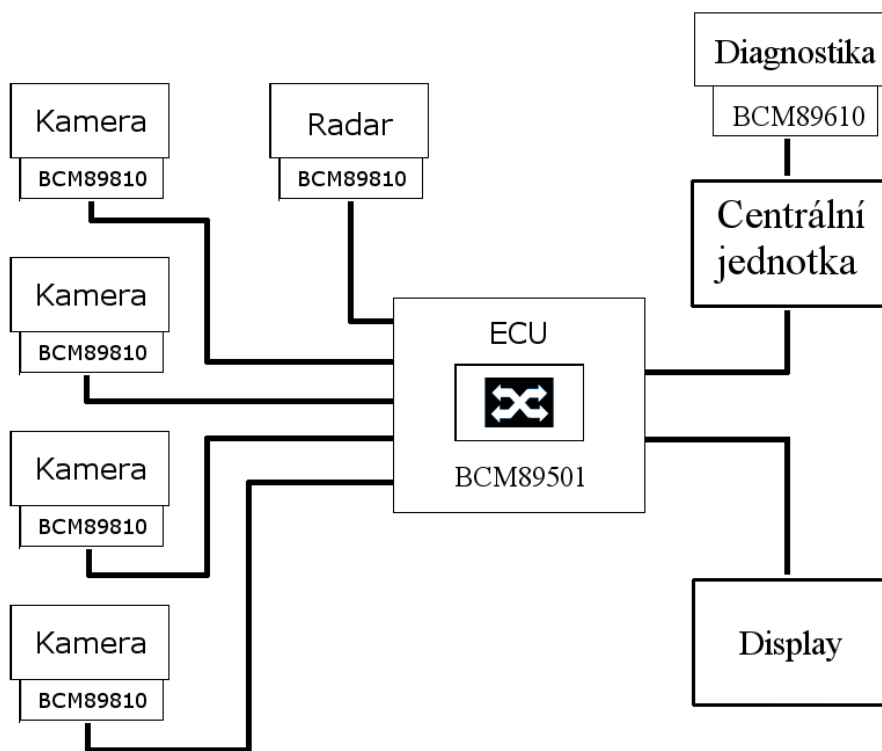
- TJA1100 – Vysílač/přijímač od firmy NXP, jedna se o alternativu k čipu BCM89810 od Broadcom. Podporuje fyzickou vrstvu BroadR-Reach a přenos 100Mb/s přes jeden pár kroucené dvojlinky. Na rozdíl od BCM89810 je k dispozici katalogový list a začíná se uvádět do prodeje<sup>2</sup>.

<sup>1</sup>Electronic Control Unit – řídicí jednotka pro řízení automobilových systémů.

<sup>2</sup>Na stránkách <http://cz.mouser.com>, s dodací lhůtou 11-16 týdnů.

- SJA1105 – Pětiportový přepínač firmy NXP, kde každý port může být zvlášť nakonfigurován pro podporu MII, RMI a RGMII, díky tomu jsou podporované fyzické vrstvy BroadR-Reach, Fast Ethernet i Gigabit Ethernet. Dále je podporováno VLAN, AVB a časově závislé aplikace. Začíná se uvádět do prodeje<sup>3</sup>.

Na obrázku 8.3 jsem sestavil blokové schéma z několika komponent. Jedná se o 4 kamery a jeden radar s fyzickou vrstvou BroadR-Reach, kterou zajišťuje čip BCM89810. Ty jsou připojené pomocí jednoho páru kroucené dvojlinky k přepínači (BCM89501) s řídicí jednotkou. Odtud je video přeposláno na obrazovku řidiči, popř. pokud by kamery uměly rozpoznávat nebezpečí, nebo značky, tak tyto informace by mohly být taky poslány do centrální jednotky pro další zpracování. Pomocí čipu BCM89610 je možnost k centrální jednotce připojit diagnostiku.



Obrázek 8.3: Návrh struktury sítě s několika koncovými zařízeními s integrovanými čipy podporující technologii BroadR-Reach

<sup>3</sup>Na stránkách <http://cz.mouser.com> a v minimálním množství 1000 kusů.

## Kapitola 9

### Závěr

Vypracoval jsem rešerši na problematiku nasazení technologie Ethernet v automobilovém průmyslu. Nejvíce odlišností od Ethernetu v běžné počítačové síti se nachází na první a druhé vrstvě OSI modelu, proto jsem zde popsal technologie a protokoly pracující na těchto vrstvách. Nejvíce času jsem věnoval technologii BroadR-Reach, která řeší funkce fyzické vrstvy a díky tomu, že používá jen jeden pár kroucené dvojlinky, tak se podstatně liší od běžného Ethernetu. Dále jsem popsal a vysvětlil protokoly zajišťující časovou synchronizaci, rozdělení sítě do VLAN (a použití priorit), kvalitu přenosu dat na síti a přenos audia a videa. Myslím, že tyto protokoly a technologie jsou důležité v časově závislých aplikacích a v automobilovém průmyslu. V době vypracování práce jsem nenalezl volně k dispozici čipy podporující technologii BroadR-Reach, proto jsem na závěr uvedl produkty, které výrobci zveřejnili, ale ještě nebyly volně k dostání. Z tohoto důvodu je práce spíše teoretického charakteru a může sloužit jako podklad pro další práci.

V budoucnu, v rámci diplomové práce, bych se rád zaměřil na praktické implementování fyzické vrstvy BroadR-Reach. V tomto směru by se mohl vytvořit některý z testovacích obvodů ze sekce 3.4.5, nebo by se mohla propojit dvě zařízení a posílat mezi nimi data. V úvahu taky připadá fyzické zařízení na monitorování a analýzu komunikace.



# Literatura

- [1] Broadcom Corporation. *BroadR-Reach Physical Layer Transceiver Specification For Automotive Applications* [online]. 2014. [cit. 17. 5. 2016]. Dostupné z: <[http://www.ieee802.org/3/1TPCESG/public/BroadR\\_Reach\\_Automotive\\_Spec\\_V3.0.pdf](http://www.ieee802.org/3/1TPCESG/public/BroadR_Reach_Automotive_Spec_V3.0.pdf)>.
- [2] Broadcome. *Broadcome Products* [online]. 2016. [cit. 17. 5. 2016]. Dostupné z: <<https://www.broadcom.com/products/ethernet-communication-and-switching>>.
- [3] IEEE Std 1588-2008. *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems* [online]. New York, 2008. [cit. 17. 5. 2016, str. 16–40]. ISBN 978-0-7381-5400-8. Dostupné z: <<https://standards.ieee.org/getieee802/download/802.3-2012.zip>>.
- [4] IEEE Std 802.1Q-2005. *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks* [online]. New York, 2006. [cit. 17. 5. 2016, str. 33–37, 279–283]. ISBN 0-7381-4877-6. Dostupné z: <<http://standards.ieee.org/findstds/standard/802.1Q-2005.html>>.
- [5] IEEE Std 802.1Qat-2010. *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP)* [online]. New York, 2010. [cit. 17. 5. 2016, str. 49–75]. ISBN 978-0-7381-6501-1. Dostupné z: <<http://standards.ieee.org/findstds/standard/802.1Qat-2010.html>>.
- [6] IEEE Std 802.1Qav-2009. *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams* [online]. New York, 2010. [cit. 17. 5. 2016, str. 41–47]. ISBN 978-0-7381-6143-3. Dostupné z: <<http://standards.ieee.org/findstds/standard/802.1Qav-2009.html>>.
- [7] IEEE Std 802.3-2012. *IEEE Standard for Ethernet* [online]. New York, 2012. [cit. 17. 5. 2016, str. 179–295]. ISBN 973-07381-7312-2. Dostupné z: <<https://standards.ieee.org/getieee802/download/802.3-2012.zip>>.
- [8] ixia. *Automotive Ethernet: An Overview* [online]. 2014. [cit. 17. 5. 2016]. Dostupné z: <[https://www.ixiacom.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper\\_1.pdf](https://www.ixiacom.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper_1.pdf)>.

- [9] NXP. *NXP Products* [online]. 2014. [cit. 17.5.2016]. Dostupné z: <[http://www.nxp.com/products/interface-and-connectivity/wired-connectivity/ethernet:MC\\_1436432488692?tab=Products](http://www.nxp.com/products/interface-and-connectivity/wired-connectivity/ethernet:MC_1436432488692?tab=Products)>.
- [10] Petr Bouška. *VLAN — Virtual Local Area Network* [online]. 2007. [cit. 17.5.2016]. Dostupné z: <<http://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network/>>.



## Příloha A

# Obsah přiloženého CD

Přiložené CD obsahuje adresář **text** s elektronickou verzí této práce.