



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

OVĚŘENÍ VYBRANÝCH KOMUNIKAČNÍCH ROZHRANÍ PROCESORU TC275

VERIFICATION OF SELECTED COMMUNICATION INTERFACES ON TRICORE TC275

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PATRIK ŠEBESTA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. PETR BLAHA, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Patrik Šebesta

ID: 125651

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Ověření vybraných komunikačních rozhraní procesoru TC275

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s procesory Infineon řady TC27x, s vývojovou deskou Aurix Application Kit TC275 a s vývojovým prostředím Free Tricore Entry Toolchain.
2. Realizujte ovladač a program pro SPI rozhraní, které umožní připojení rozšiřujícího modulu analogových vstupů CIC-751. Komunikace by měla být postavena na přerušení a na použití DMA kanálu.
3. Realizujte ovladač a program pro CAN rozhraní a otestujte propojení mezi Aurix Application Kit TC275 a počítačem PC s vloženou CAN kartou.

DOPORUČENÁ LITERATURA:

[1]Herout, P.: Učebnice jazyka C. Nakladatelství KOPP, České Budějovice, září 2004, IV. přepracované vydání, ISBN 80-7232-220-6.

Firemní dokumentace firmy Infineon a HighTec.

Termín zadání: 9.2.2015

Termín odevzdání: 18.5.2015

Vedoucí práce: doc. Ing. Petr Blaha, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá nastavením vybraných periférií procesoru TC275 rodiny AURIX vyvinuty firmou Infineon. Periferie procesoru QSPI realizuje komunikaci SPI nastavené jako master s využitím modulu DMA obsluhujícího přijímací a vysílací registre QSPI který v určitých časových intervalech čte hodnoty z analogově digitálního převodníku CIC751 pracujícího v režimu slave. Další nastavenou periferií procesoru je modul MultiCAN realizující CAN komunikaci. Vytvořené ovladače periférií využívají pouze základní hlavičkové soubory s definicí registrů procesoru, které jsou součástí vývojového prostředí TriCore Free Entry Tool Chain, ve kterém jsou vytvořené programy periférií.

Klíčová slova

Přímý přístup do paměti, sériová komunikační rozhraní SPI, sériová komunikace CAN, procesor TC27x, analogově digitální převodník CIC751

Abstract

Diploma thesis handles with set up of peripheral modules of the processor TC275 families' AURIX developed by Infineon. Processor's peripheral module QSPI implements communication SPI set up as master on a bus supported by another processor's module DMA. Module DMA periodically service transmit and receive shift buffers of QSPI which are connected with slave analog to digital converter IC CIC751. Another peripheral module is MultiCAN. Programmed drivers used only basic header files with register definition of processor TC275, which are part of IDE TriCore Free Entry Tool Chain used for created drivers.

Keywords

Direct memory access, serial peripheral interface SPI, serial communication CAN, processor TC27x, analog to digital converter CIC 751

Bibliografická citace:

ŠEBESTA, P. *Ověření vybraných komunikačních rozhraní procesoru TC275*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 52s. Vedoucí diplomové práce byl doc. Ing. Petr Blaha, Ph.D.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma *Ověření vybraných komunikačních rozhraní procesoru TC275* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **18. května 2015**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Petrovi Blahovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **18. května 2015**

.....
podpis autora

Obsah

| | | |
|-------|---|----|
| 1 | Úvod..... | 8 |
| 2 | Vývojová deska Aurix Application Kit TC275 a Free Tricore Entry Toolchain | 9 |
| 2.1 | Vývojová deska Aurix Application Kit TC275..... | 9 |
| 2.2 | Procesor TC275..... | 10 |
| 3 | Modulu CIC-751 připojený k Rozhraní SPI s použitím DMA..... | 12 |
| 3.1 | DMA modul procesoru TC275 | 12 |
| 3.1.1 | Blokové schéma DMA kontroléru | 12 |
| 3.1.2 | Řídící blok transakce (Transaction Control Engine)..... | 14 |
| 3.1.3 | Funkční popis DMA a názvosloví..... | 15 |
| 3.1.4 | Funkční princip | 15 |
| 3.1.5 | Shadow adresa zdroje nebo cíle | 15 |
| 3.1.6 | Řízení požadavku aktivace DMA kanálu..... | 16 |
| 3.2 | QSPI modul procesoru TC275 | 17 |
| 3.3 | Modul QSPI | 18 |
| 3.3.1 | Master Režim | 19 |
| 3.3.2 | Nastavení přenosové rychlosti | 20 |
| 3.3.3 | Rozhraní přenosu dat mezi modulem a registry..... | 21 |
| 3.3.4 | Nastavitelné módy přenosu dat QSPI..... | 23 |
| 3.3.5 | Loop back mode..... | 24 |
| 3.3.6 | Nastavitelná přerušení..... | 24 |
| 3.4 | Analogově digitální převodník CIC751 | 27 |
| 3.4.1 | Čtení hodnot z převodníku | 27 |
| 3.5 | Driver QSPI s obsluhou modulu DMA..... | 28 |
| 3.5.1 | Nastavení adres uložišť přenosu DMA | 28 |
| 3.5.2 | Implementované funkce driveru..... | 29 |
| 4 | Realizace Rozhraní can a otestování s pc kartou..... | 31 |
| 4.1 | Definice protokolu CAN podle ISO 11898-1 | 31 |
| 4.1.1 | Datová vrstva 2 | 32 |
| 4.1.2 | Fyzická vrstva 1 | 32 |
| 4.2 | Názvosloví používané v dokumentu ISO standardu protokolu..... | 33 |
| 4.3 | CAN bit..... | 36 |
| 4.4 | Struktura přenášeného rámce | 38 |
| 4.5 | Vlastnosti, detekce a zpracování poruch na sběrnici..... | 43 |
| 4.5.1 | Zpracování zachycených chyb | 43 |
| 4.6 | Periferie procesoru MultiCAN+..... | 44 |
| 4.6.1 | Struktura seznamu správ | 46 |
| 4.7 | Driver MultiCAN+..... | 47 |
| 5 | Závěr..... | 49 |

1 ÚVOD

Práce se zabývá nastavením periférií tří jádrového procesoru Infineon TC275 který je základem vývojové desky Application kit TC275 s využitím vývojového prostředí Free AURIX Entry Tool chain. Každá kapitola začíná obecnou teorií následující popisem periférie. Nastavované periférie procesoru jsou modul OSPI, DMA a CAN. Moduly QSPI a DMA plní funkci sériové komunikace sloužící k samostatné obsluze vyčítání hodnot z připojeného obvodu analogově digitálního převodníku CIC751. Další periférií je modul MultiCAN realizující komunikaci CAN. Práce obsahuje dva programy s nastavením uvedených periférií funkčních pouze v režimu testování, které umožňuje nastavení příslušného modulu.

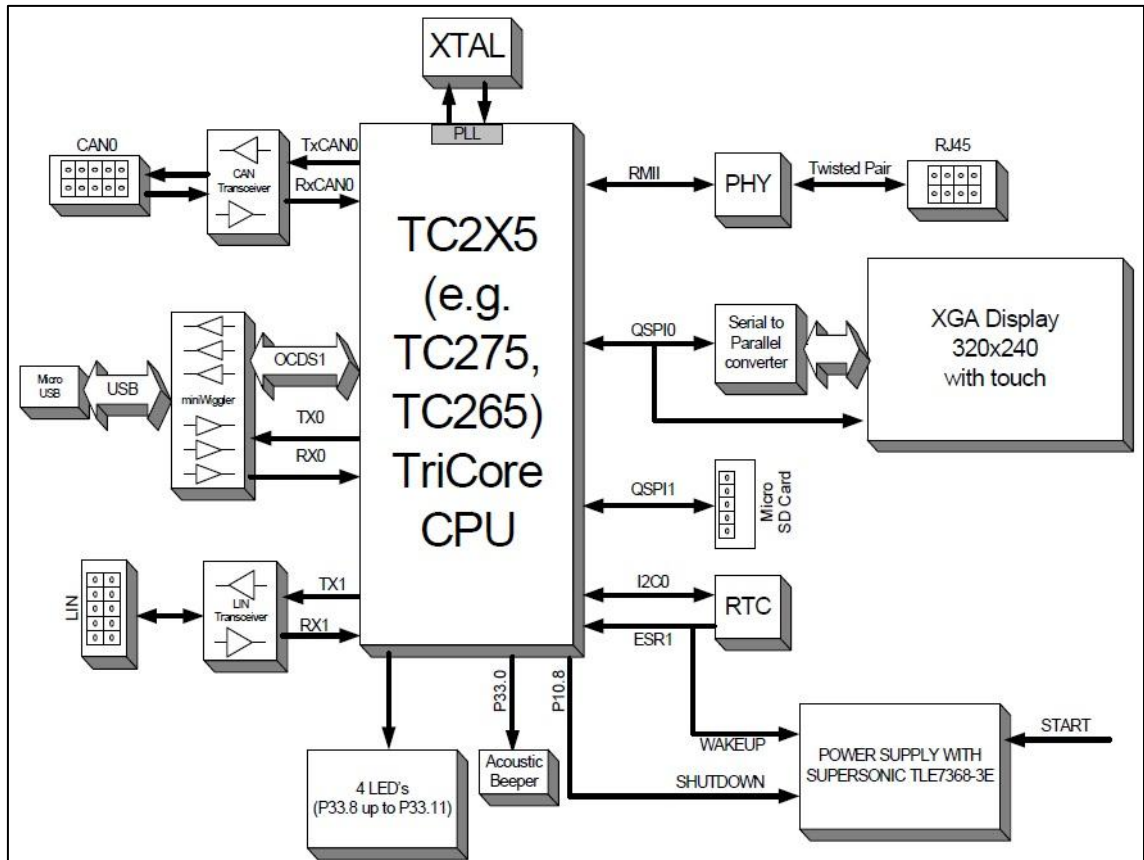
2 VÝVOJOVÁ DESKA AURIX APPLICATION KIT TC275 A FREE TRICORE ENTRY TOOLCHAIN

2.1 Vývojová deska Aurix Application Kit TC275

AURIX Application je vývojová platforma s procesorem TC275 od firmy Infineon, viz blokové schéma Obrázek 2.1.

Komponenty a konektory desky:

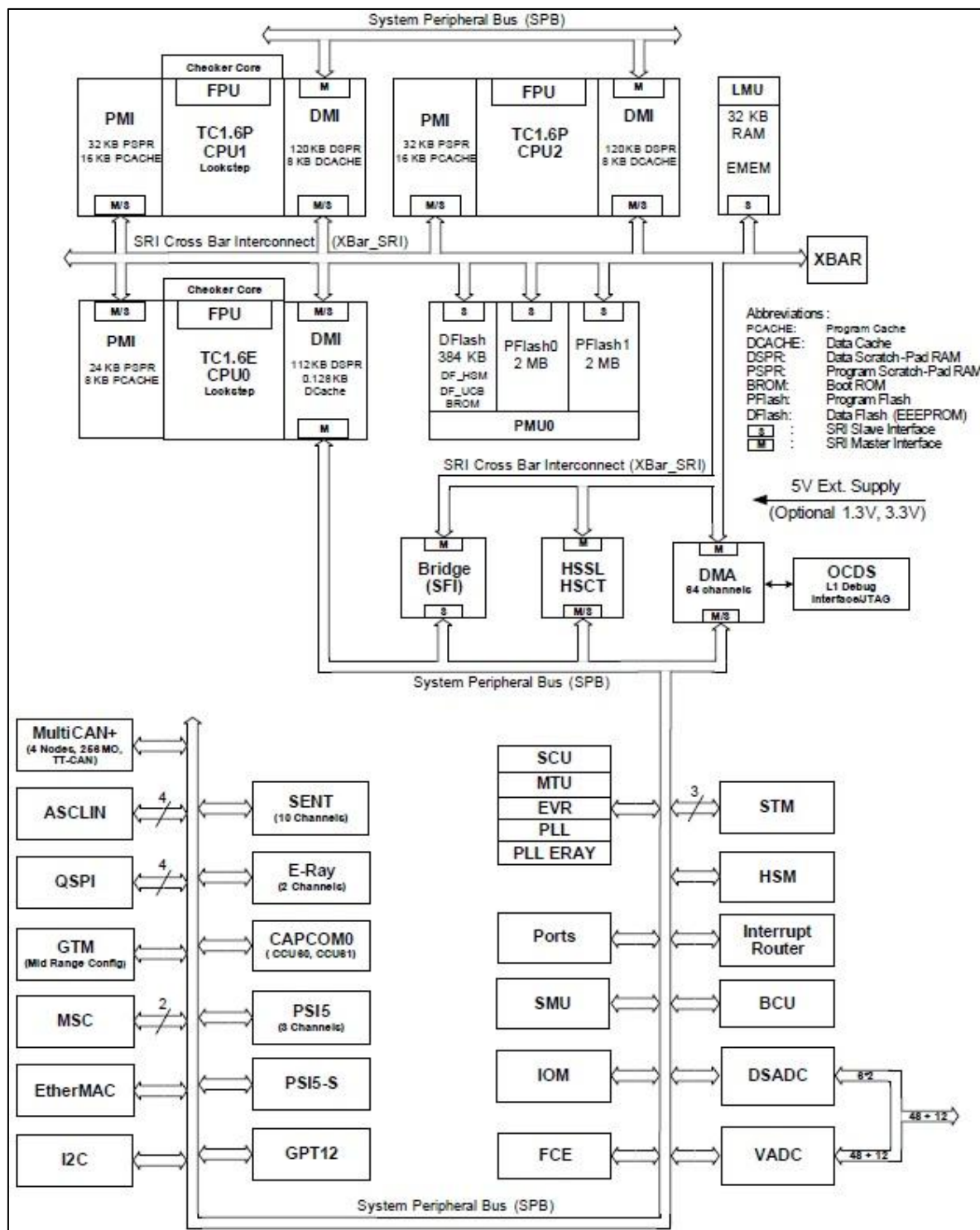
- Konektory
 - Micro USB pro ASC rozhraní ASCLIN a miniWiggler
 - RJ45 pro Ethernet
 - 16 pinové rozhraní pro JTAG
 - 10 pinové rozhraní pro DAP
 - 10 (2x5) pinový konektor pro LIN
 - 10 (2x5) pinový konektor pro CAN
 - dva 40 pinové konektory I/O portů procesoru
 - mini SD slot
- Komponenty
 - Infineon TLE 7368-3E kontrolér napětí
 - Indikace napájecího napětí LEDkou
 - LED indikace resetu
 - Infineon vysokorychlostní CAN driver TLE 6250G
 - Infineon LIN driver TLE 7259-2GE
 - I2C řadič, hodiny reálného času
 - Převodník USB na UART FT2232HL (FTDI)
 - Radič pro dotykový displej ADS7843
 - Tlačítko restartu
 - Tlačítko startu
 - Xilinx CPLD XC9572XL
 - Displej XGA 320x240



Obrázek 2.1 Blokové schéma vývojové desky AURIX Application Kit[2].

2.2 Procesor TC275

Procesor TC275 patří do produktové rodiny AURIX firmy Infineon. TC275 je 32 bitový mikroprocesor schopen pracovat v reálném čase s výkonem DSP splňující požadavky AUDIO. Obsahuje 3 na sobě nezávislé procesorové jednotky CPU. Řada AURIX může být aplikována k řízení spalovacích motorů, převodovek, posilovačů řízení atd [8]. Procesor TC275 je vysokovýkonný mikroprocesor obsahující tři TriCore procesorové jednotky CPU, programovou a datovou paměť, sběrnice, DMA kontrolér a nejpoužívanější periferie pro vestavné systémy.



Obrázek 2.2: Blokové schéma procesoru TC275 [3].

3 MODULU CIC-751 PŘIPOJENÝ K ROZHRAŇÍ SPI S POUŽITÍM DMA

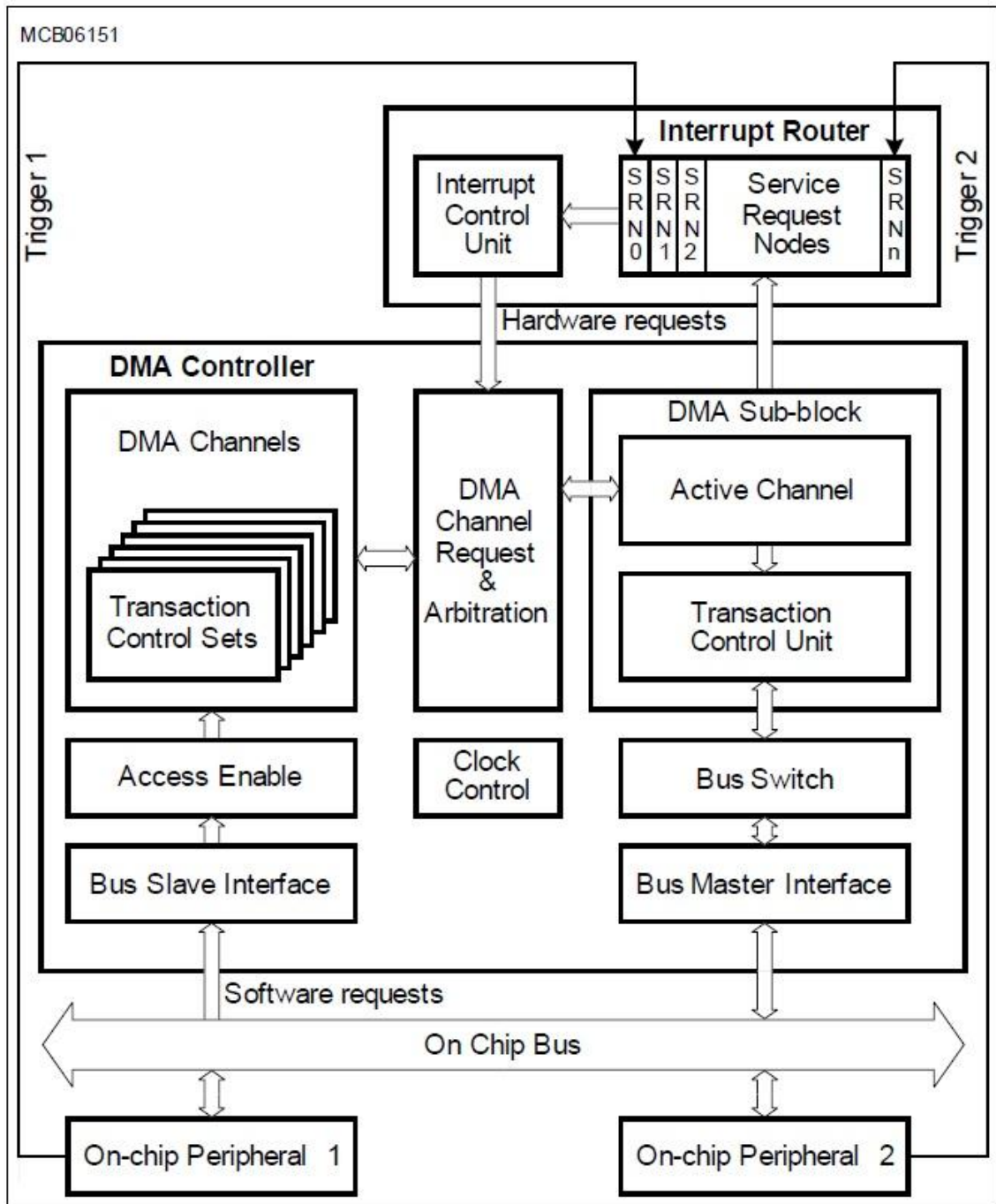
3.1 DMA modul procesoru TC275

Vlastnosti modulu DMA

- Podpora 64 kanálů DMA
- Hardwarová žádost kanálu spouští řadič přerušeni (IR) který je součástí jednoty přerušeni (ICU).
- Softwarová žádost spuštění DMA
- DMA kontrolér podporuje dvě řídicí jednoty (move engines) v případě paralelní obsluhy kanálů
- Individuální nastavení kanálu
- DMA nastavení uloženo v paměti DMARAM
- Přenos dat je směrován zdrojové do cílené místo v paměti na základě adresy zdroje a cílového úložiště.
- Programovatelná velikost rámce dat jednoho přesunu
 - SPB master : 8, 16, 32 bitů
 - SRI master: 8, 16, 32, 64, 128, 256 bitů
- Každý DMA kanál generuje jedno přerušeni s nastavitelnou prioritou
- Operační frekvence:
 - Jádro kontroléru DMA pracuje na frekvenci sběrnice SRI pro co nejrychlejší přesun dat z adresy zdroje do adresy cíle
 - Část nastavení DMA (DMARAM, slave rozhraní, přerušeni, SPB master rozhraní) pracuje na frekvenci SPB[3].

3.1.1 Blokové schéma DMA kontroléru

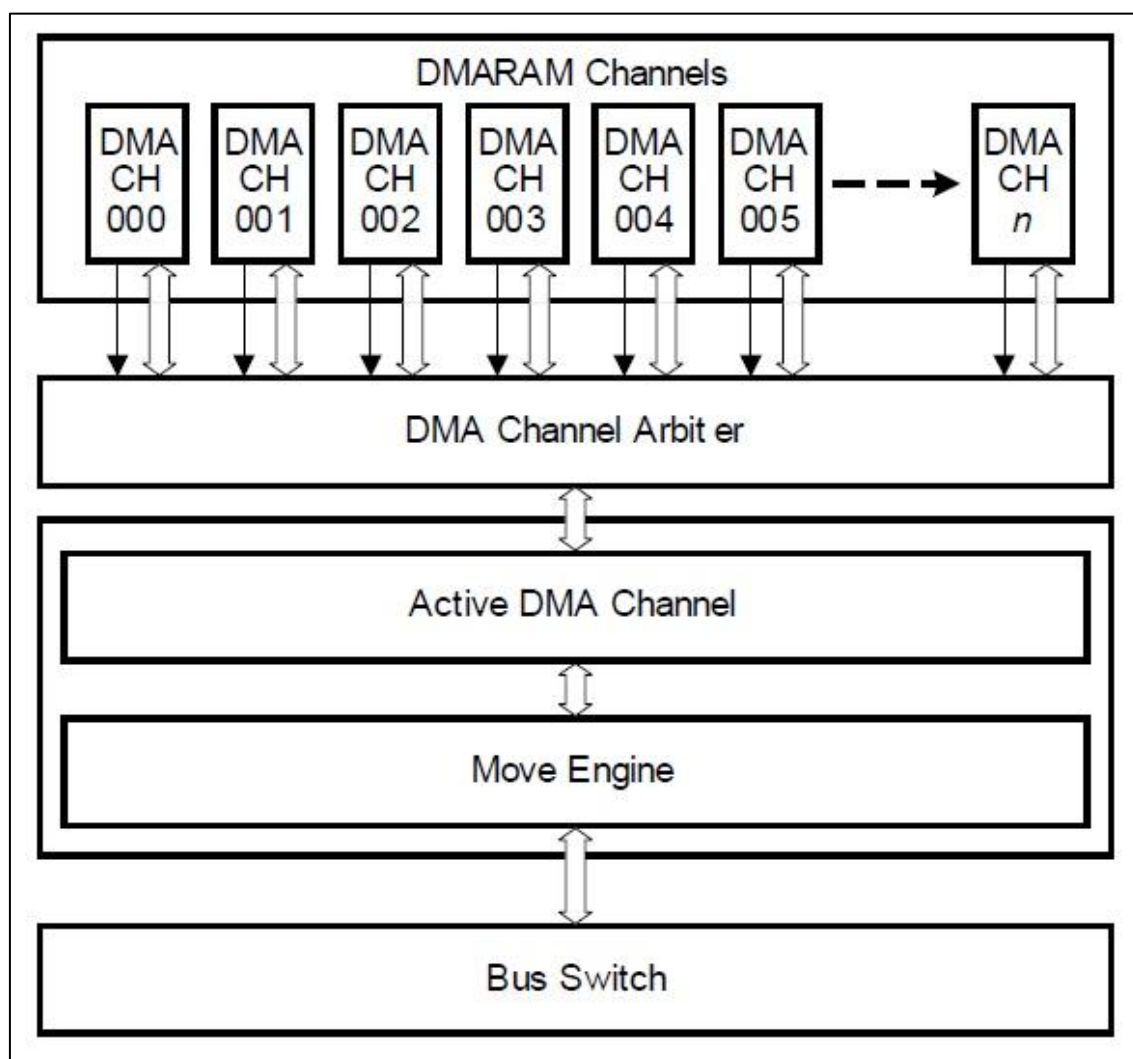
DMA kontrolér přesouvá data ze zdroje do cílového úložiště bez zásahu CPU. Přesun dat řídí jeden aktivní kanál. Blok označovaný DMA sub-block obsluhuje požadavek na přenos kteréhokoliv kanálu. Sběrníkový prepínač zajišťuje spojení mezi DMA sub blokem a vnitřní sběrnici. Programování nastavení DMA konfigurace probíhá přes rozhraní Bus slave interface[3].



Obrázek 3.1: Blokové schéma DMA periferie[3].

3.1.2 Řídicí blok transakce (Transaction Control Engine)

Každý DMA sub-blok obsahuje řídicí blok Transaction Control engine viz. Obrázek 3.2.



Obrázek 3.2:Řídicí blok transakce[3].

DMA Arbitr

Úlohou je monitorovat kanály žádající o přenos. Kanál s největším pořadovým číslem žádající o přenos získává arbitraci. Následně se přesune nastavení žádajícího kanálu z paměti DMARAM do části subbloku nazývaného registr aktivního kanálu (active channel register set) a přenos je aktivován. Ukončením transakce znovu probíhá proces arbitrace žádosti, přenos nastavení a zahájení komunikace [3].

DMA Move Engine

Žádá sběrnici a přemísťuje data podle nastavených parametrů aktivního kanálu. Probíhající transakce některého kanálu řízeného move engine nemůže být přerušena a je vždy dokončena. Přerušování, reset, pozastavení kanálu a pozastavení při ladění je aktivní až po ukončení transakce. Po ukončení přenosu, move engine odešle zpět aktualizovanou adresu do aktivního registru nastavení kanálu [3].

3.1.3 Funkční popis DMA a názvosloví

DMA přesun (DMA move)

Přesun je složen z dvou částí:

- Čtecí část (read move) kdy jsou data přesunuta ze zdrojové části do kontroléru DMA.
- Zápis dat (write move) přesune data z mezi úložiště kontroléru DMA na cílovou adresu [3].

Data jsou vždy přesouvána ze zdroje přes kontrolér, kde jsou dočasně uloženy, na cílovou adresu. Velikost jednoho přesunu čtení nebo zápisu je možno nastavit na hodnotu 8, 16, 32, 64, 128, 256 bitů [3].

DMA přenos (DMA transfer)

Každý přenos (DMA transfer) představuje určitý počet přesunů (DMA moves). Je možno nastavit 1, 2, 3, 4, 5, 8, 9, 16 přesunů [3].

DMA transakce (DMA transaction)

Transakce je složena z několika (minimálně jednoho) přenosů.

Např. 1024 slovo velikosti 32 bitů může být ze zdroje uložení na cílovou adresu přemístěno pomocí 256 transakcí, každá transakce obsahuje 4 přenosy s 1 přesunem o velikosti 32 bitů[3].

Propojený seznam (Linked List)

Je série DMA transakcí vykonávaných pomocí jednoho kanálu [3].

3.1.4 Funkční princip

Požadavek na přesun dat může být jak hardwarový, tak softwarový. Hardwarový požadavek se spouští pomocí jednotky přerušení (ICU) nebo pomocí jiného kanálu. Konfigurovatelné je i možnost paralelního požadavku na přesun dat a požadavek přerušení generovaný jedním zdrojem [3].

Primární částí kontroléru jsou kanály, sub-blok (move engines) a přepínač sběrnice. DMA sub-blok může na sběrnici SBP, SRI pracovat jako master nebo slave [3].

DMA kanál

Konfigurační nastavení každého kanálu je uloženo v DMARAM paměti velikosti 8x32bitů. V případě požadavku DMA na přesun dat se konfigurační nastavení kanálu kopíruje do DMA sub bloku [3].

3.1.5 Shadow adresa zdroje nebo cíle

Shadow adresování umožňuje automatickou změnu cílové adresy bez zásahu do registru cílové adresy. Na začátku nové transakce je adresa z shadow registru zapsaná do registru s cílovou adresou bez zásahu CPU. Shadow registr může být použit k uložení

nové zdrojové adresy. Do registru nemůže být uložena cílová i zdrojová adresa současně, pokud se požaduje změna adresy v nové transakci [3].

3.1.6 Řízení požadavku aktivace DMA kanálu

K aktivaci DMA kanálu máme dvě možnosti:

- Hardwarový požadavek
- Softwarový požadavek

Hardwarové požadavky jsou spouštěny řídicí jednotkou přerušení (ICU). Nastavením bitu CHCFGRz.PRSEL = 1, aktivní DMA kanál může spustit hardwarový požadavek dalšího nižšího kanálu (z-1), v tomto případě je přerušení blokováno. Hardwarové přerušení je povoleno bitem TSRz.HTRE.

Softwarové přerušení je spuštěno nastavením bitu CHCSRz.SCH. Příznak TSRz.CH indikuje, jestli je softwarový nebo hardwarový požadavek aktivní [3].

3.1.6.1 Operační módy DMA kanálu

Operační mód DMA kanálu je programovatelný individuálně pro každý kanál. DMA kanál může pracovat v jednom z následujících režimů:

- Softwarově řízený mód
- Hardwarově řízený mód: v tomto módu je možná další konfigurace:
 - Single mód
 - Continuous mód
 - Přepojený seznam (Linked mód) [3].

Softwarový mód řízení kanálu je spouštěný nastavením příslušného bitu. Hardwarový mód je řízen jednotkou přerušení (ICU) generující přerušení spouštěné vnější periferií.

Hardwarovým řízením nastaveným v režimu single mode se DMA kanál deaktivuje posledním přenosem DMA transakce, v continuous módu, kanál zůstává po celou dobu povolen [3].

V režimu propojeného seznamu (Linked list) je kanál přeprogramován novou konfigurací při ukončení právě probíhající transakce. Další transakce může být automaticky aktivována softwarovým požadavkem [3].

Softwarový mód

V tomto módu aktivací požadavku spustí jednu kompletní transakci nebo jeden přenos nastavením příslušného bitu[3].

Hardwarové módy

Nastavení umožňuje spuštění DMA transakce nebo jeden DMA přenos. Hardwarové řízení umožňuje nastavit dva módy:

- Single mód: Po přenosu hardwar zakáže další žádost o přenos a pro další použití musí být spuštěn znovu.
- Continuous mód: Skončením transakce zůstává žádost o přenos aktivní[3].

3.2 QSPI modul procesoru TC275

Sériové periferní rozhraní je komunikační obousměrný (full duplex)synchronní komunikační protokol s jedním hlavním a podřízenými zařízeními připojenými na sběrnici (master slave). Pouze hlavní (master) zařízení řídí přenos dat mezi ostatními zařízeními. Sběrnice je tvořena čtyřmi vodiči, dva řídicí (SCLK, SS/CS) a dva datové (MOSI, MISO). Každé zařízení je vybaveno dvěma posuvnými registry příjmu a odesílání dat[5],[6].

Zaužívaný způsob označení vodičů sběrnice:

- Časovací hodiny: SCLK, SCK, CLK
- Odesílací datový: MOSI, SIMO, SDO, SDI, DO, DOUT, SI, MTSR
- Přijímací datový: MISO, SOMI, SDO, SDI, DI, SO, MRST
- Výběrový: SS, CS, CSB CSN[5],[6].

Přenos se zahajuje inicializací podřízeného obvodu, hlavní uzel řídí časování(SCLK) a vybírá (SS/CS) jedno zařízení s kterým proběhne výměna dat. Na sběrnice může být připojen pouze jeden podřízený obvod a pokud to obvod podporuje, tvoří ji pouze tři vodič(SCLK, MISO, MOSI), výběrový vodič se připojí na konstantní úroveň. Komunikující uzly jsou navzájem synchronovány hodinovým impulzem (SCLK). Přenos dat na sběrnici probíhá synchronně tzn. každým hodinovým impulzem hlavního uzlu (SCLK) se současně přenesou jeden bit z hlavního do podřízeného (na datovém vodiči MOSI) a z podřízeného do hlavního posuvného registra (na datovém vodiči MISO) komunikujících obvodů. Každý obvod obsahuje samostatný přijímací a vysílací registr. Ukončením přenosu se odpojí hlavní uzel od podřízeného tím se ukončí komunikace a může se začít s jiným obvodem[5],[6].

Společnou konfiguraci jak hlavního obvodu tak podřízeného obvodu je nastavení citlivosti dat na jednu z hran hodinového impulzu (CPOL bit) a nastavení fáze určující kdy jsou data platná (CPH bit) na výstupu, můžou zde nastat čtyři kombinace.

SPI je podporován mnoha obvody, AD, DA převodníky, flash paměti, teplotní senzory atd[5],[6].

Výhodou je že na krátkou vzdálenost komunikovat vysokou rychlostí (řádově desítky MHz), jednoduché rozhraní a jednoduchost samotného obvodu podporující komunikaci[5],[6].

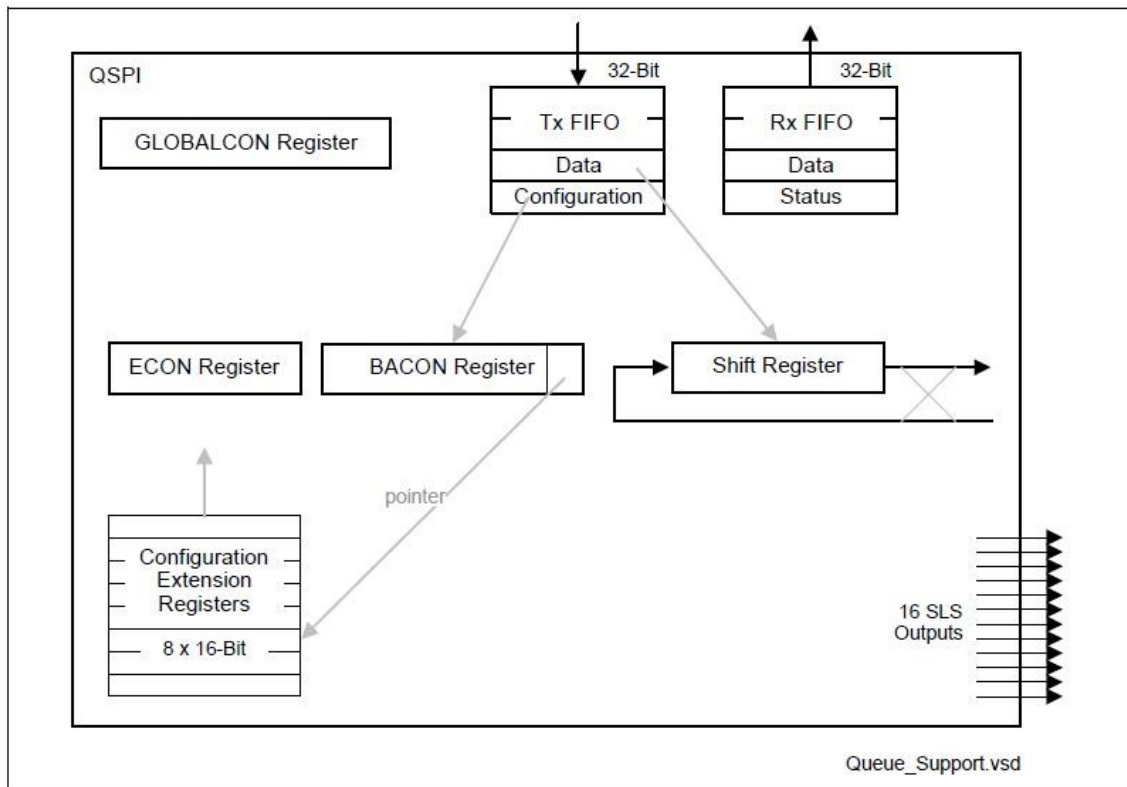
3.3 Modul QSPI

Modul může pracovat jak v režimu master tak v režimu slave s možností obousměrné (full duplex), jednosměrné (half duplex) komunikace, čtyřvodčivého nebo dvouvodčivého zapojení. Nastavitelná je rovněž délka přenášených dat (při délce od 4 do 32 bitů může být přenosová rychlost až 50 Mbit/s). QSPI podporuje obsluhu pomocí DMA modulu[2].

Vlastnosti modulu:

- Master nebo slave v režimu obousměrné nebo jednosměrné komunikace.
- Nastavitelná délka přenášených dat až do 256 bitů
- Maximální přenosová rychlost 50Mbit/s
- Přerušování generovaným odesláním nebo příjmem dat, určitý specifický časový komunikační fáze, errorová hlášení
- Nastavení testovacího režimu (loop back mode)
- Zastavení komunikace při detekci plné fronty RxFIFO[2]

Hlavní výhodou modulu řízení a zpracování dat pomocí fronty typu FIFO do které se vkládá jak konfigurace tak dat a modul už dále pracuje podle zvoleného nastavení. Výhodou je podpora modulem DMA která aktivací může sloužit k zápisu konfigurace a dat to registrů do fronty a spustit modul[2].



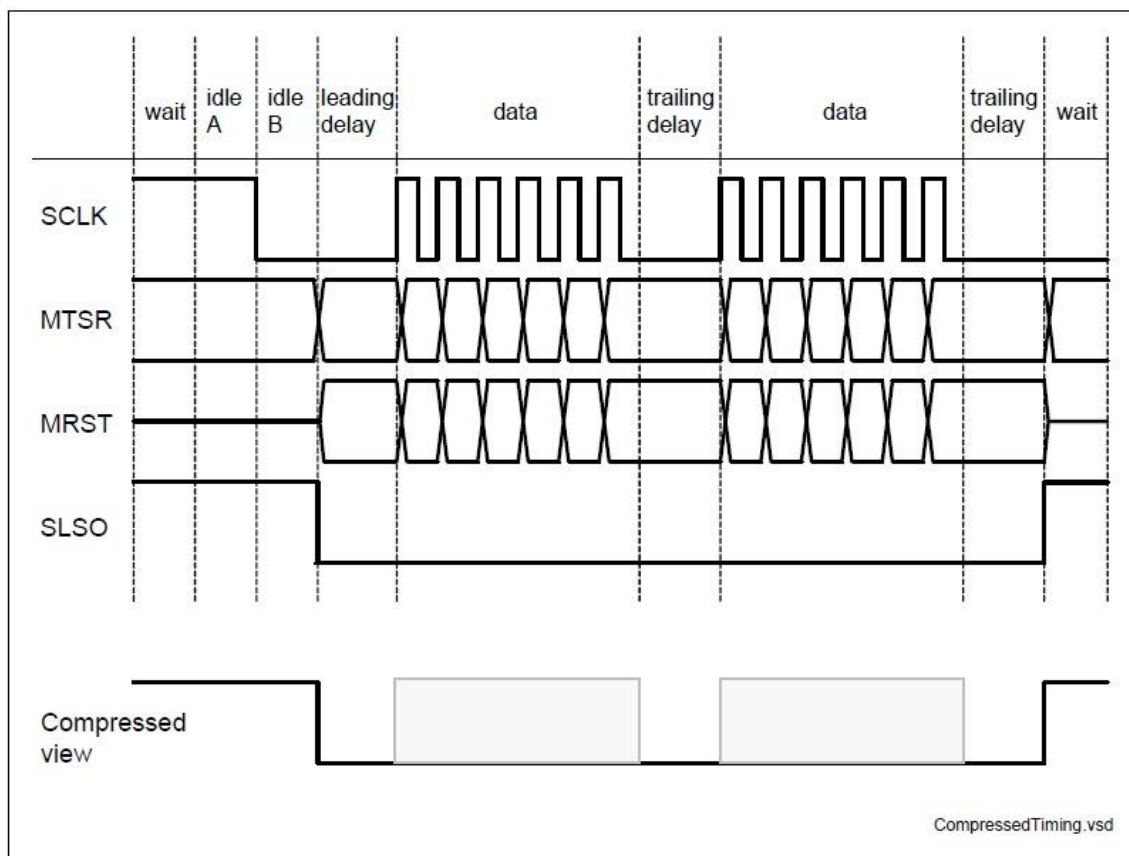
Obrázek 3.3: Blokové schéma principu činnosti QSPI[2].

3.3.1 Master Režim

3.3.1.1 Komunikační cyklus jedné fáze

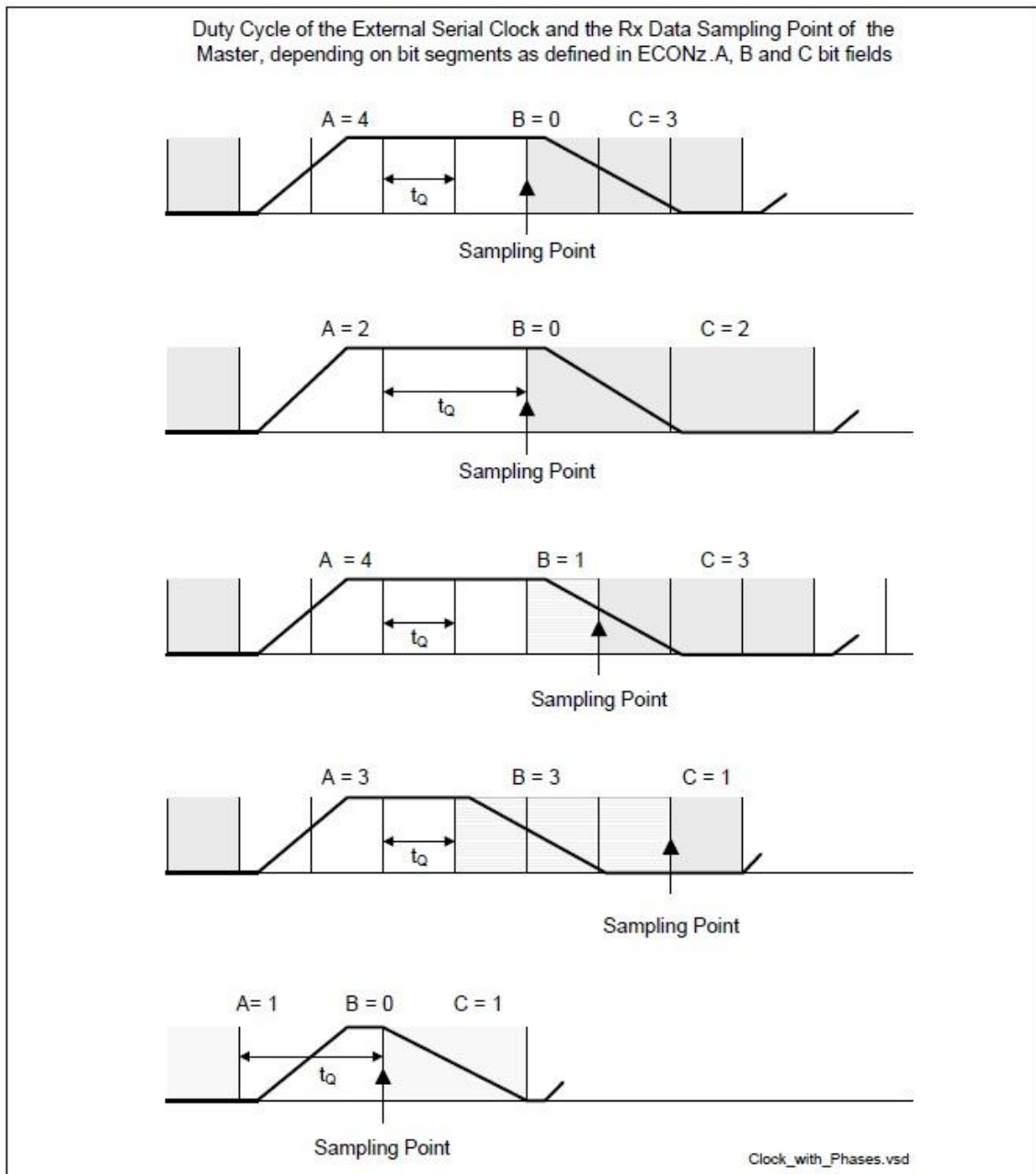
V master režimu je komunikační cyklus zahajován aktivací výběrového signálu (SLSO) podřízeného obvodu po kterém následují následující fáze komunikace:

- Wait (nelze nastavit)
- Idle A
- Idle B
- Leading delay
- Data
- Trailing delay[2].



Obrázek 3.4: Komunikační fáze cyklu[2].

Každou fází cyklu (viz. Obrázek 2) je možno nastavit příslušným registrem. Základní zdroj hodin pro blok obsluhující řídicí a datové signály je f_{BAUD2} . Velikost bitu v kontextu s modulem QSPI je označován jako časová jednotka t_Q která se skládá z nastavené hodnoty impulzů frekvence f_{BAUD2} a umožňuje tak flexibilitu přenosové rychlosti viz. Obrázek 3.4. Jednotlivé fáze jsou integrací nastavené hodnoty t_Q [2].



Obrázek 3.5: Časování a vzorkování bitu[2].

Dostupnost dat na výstupu po aktivaci výběrového signálu (SLSO) je ovlivněna nastavením fáze vzhledem k časovacím hodinám (nástupná nebo sestupná hrana hodinového signálu) na výstupu master obvodu, tato konfigurace se provádí pomocí bitu CPH[2].

3.3.2 Nastavení přenosové rychlosti

Přenosová rychlost pro slave obvody by se neměla výrazně měnit. Obvody slave mohou představovat kapacitní zátěž na výstupy master obvodu. Doporučuje se rozdíl taktovacích frekvencí pro slave obvody byl v pásmu 6:1 pro bitový segment[2].

Jednotlivé řídicí fáze je možno nastavit podle následujících rovnic[2]:

$$T_{SCLK} = T_{BAUD2} * GLOBALCON.TQ * ECON.Q * (A + B + C) \quad (1)$$

$$T_{LEAD} = T_{BAUD2} * BACON.LPRE * BACON.LEAD \quad (2)$$

$$T_{TRAIL} = T_{BAUD2} * BACON.TPRE * BACON.TRAIL \quad (3)$$

$$T_{IDLE\ A,B} = T_{BAUD2} * BACON.IPRE * BACON.IDLE \quad (4)$$

$$T_{STROBE} = T_{BAUD2} * GLOBALCON.TQ * ECON.Q * GLOBALCON.STROBE \quad (5)$$

Rovnice pro výpočet přenosové rychlosti[2]:

$$BR = \frac{f_{BAUD2}}{(TQ + 1) * \underbrace{(A + 1 + B + C)}_{\geq 4}} \quad (6)$$



Obrázek 3.6: Předdělička (nastavení přenosové rychlosti) [2].

3.3.3 Rozhraní přenosu dat mezi modulem a registry.

Konfigurace a data jsou v modulu uloženy do paměti typu FIFO. Konfigurace a data jsou uloženy na specifické adresy[2].

TxFIFO je rozdělen na :

- DATA_ENTRY: zápis do tohoto adresního prostoru je vždy považován za data
- BACON_ENTRY: Zápis do této lokace je interpretován jako konfigurace.
- MIX_ENTRY: Zápis do této lokace je podle určitých pravidel interpretován jako data nebo konfigurace[2].

RxFIFO :

- RX_EXIT: přijímací fronta modulu, podle pravidel se čtením se získávají data a příznaky[2].

TxFIFO v sobě zahrnuje jak data tak konfiguraci a automaticky rozlišuje o který druh dat se jedná.

Přenosem dat a následně data čekajících ve frontě:

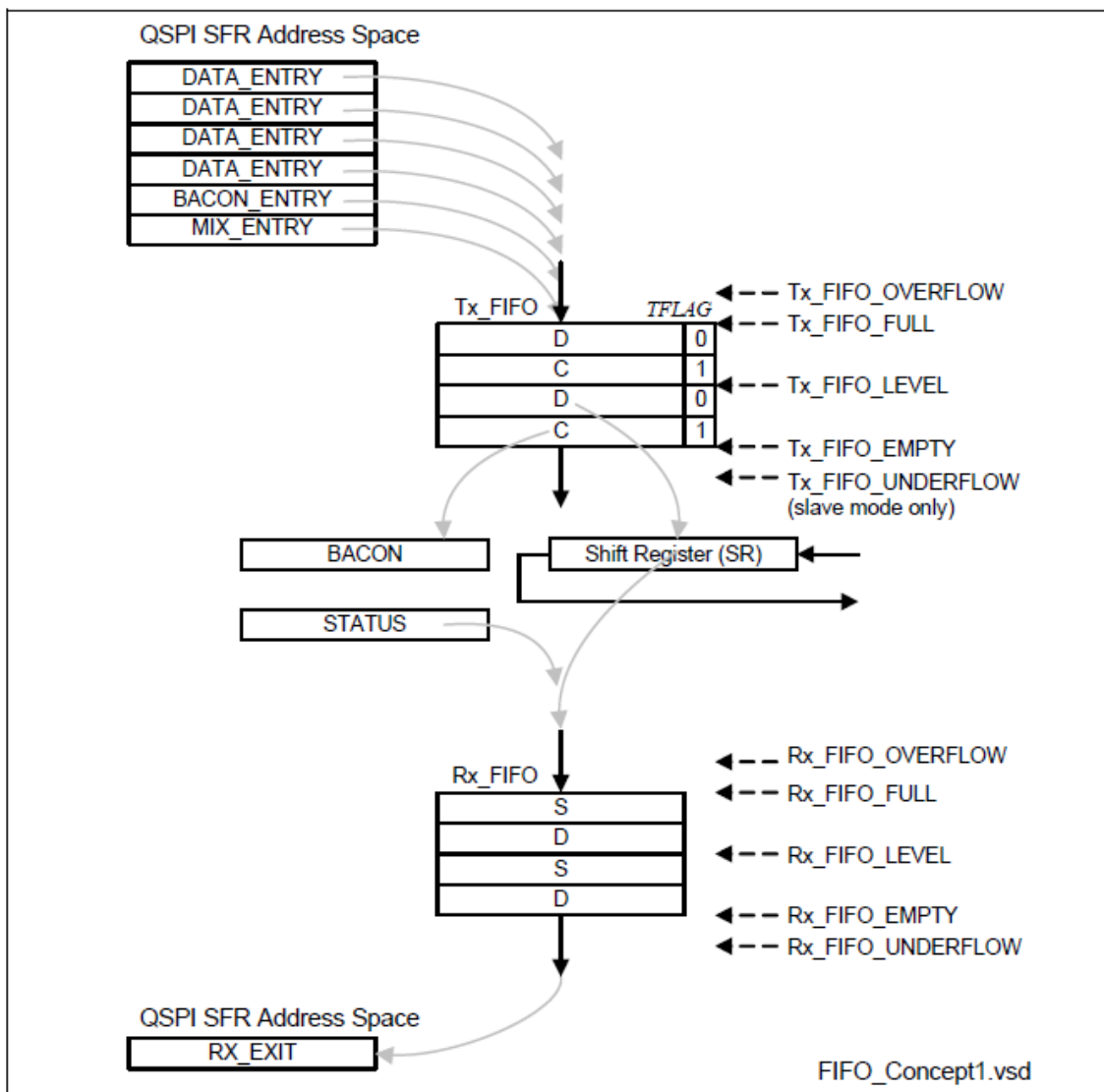
- V modu Short: nový přenos dat začne se stejným nastavením
- V módu long: další data jsou ignorovány[2].

Tx FIFO error podmínky

- V případě zápisu do plné fronty FIFO je generováno přerušení přetečení[2].

Rx FIFO error podmínky

- Čtením prázdného registru RXFIFO generuje přerušení podtečení.
- Hardwarový zápis do naplněného RxFIFO daty nebo příznaky generuje přerušení přetečení
- Nastavením bitu GLOBALCON.SFR, komunikace je zastavena v případě plné fronty[2]



Obrázek 3.7: Architektura Tx a RX FIFO[2].

3.3.4 Nastavitelné módy přenosu dat QSPI

Short data mode

Mód umožňuje přenos dat od 2 do 32 bitů v jednom rámcí. Zde je možno využít DMA modul spuštěný událostí stavu TxFIFO (např. požadavek k plnění registru dat) k zápisu konfigurace a dat na příslušné paměťové adresy z přednastavených míst[2].

Long data mode

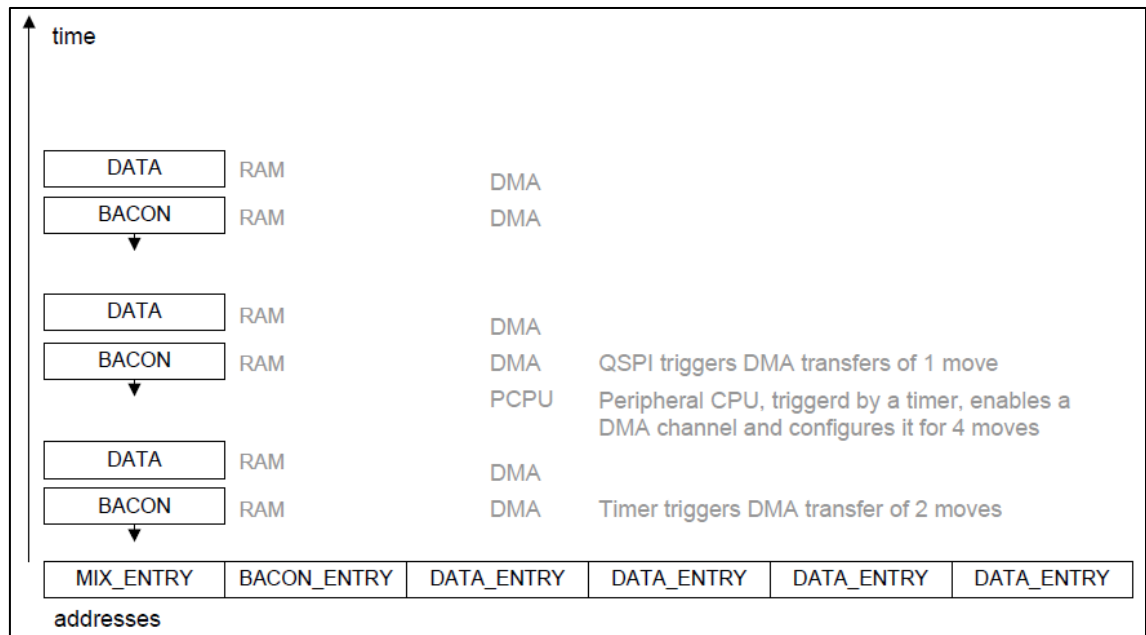
Umožňuje přenos dat až 256 bitů v jednom rámcí. Data jsou rozdělena na segmenty 32 bitů které představují délku pro uložení fronty[2].

Continuous mode

Nastavení umožňuje spojitý přenos datových rámců. Tento mód může být použit jak pro Short tak pro Long mód[2].

Single configuration (Maximálně 16 bitů přenosu)

Aplikacím pracujícím s délkou dat do 16 bitů poskytuje modul QSPI využití registru MIX_ENTRY do kterého se zapíše konfigurace a následně data s maximální uvedenou délkou. Modul automaticky rozpoznává data od konfigurace která je vždy 32 bitová a data obsahují maximálně 16 bitů[2].

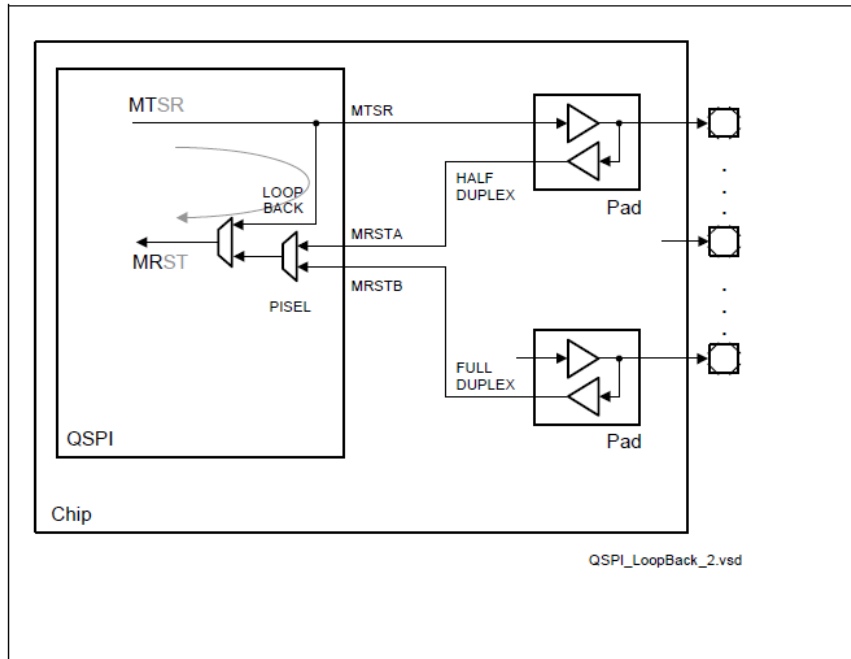


Obrázek 3.8: Single configuration[2].

Na Obrázku 3.8, je zobrazeny kroky při odesílání dat v režimu Single configuration. Jak konfigurační tak data se zapisují na jednu adresu MIX_ENTRY registru fronty a dále jsou modulem automaticky rozpoznány logikou modulu[2].

3.3.5 Loop back mode

Umožňuje odzkoušení nastaveného modulu bez připojeného vnějšího obvodu. Nastavením se vnitřně propojí vysílací a přijímací vedení. Mód je možno nastavit pouze v režimu master[2].

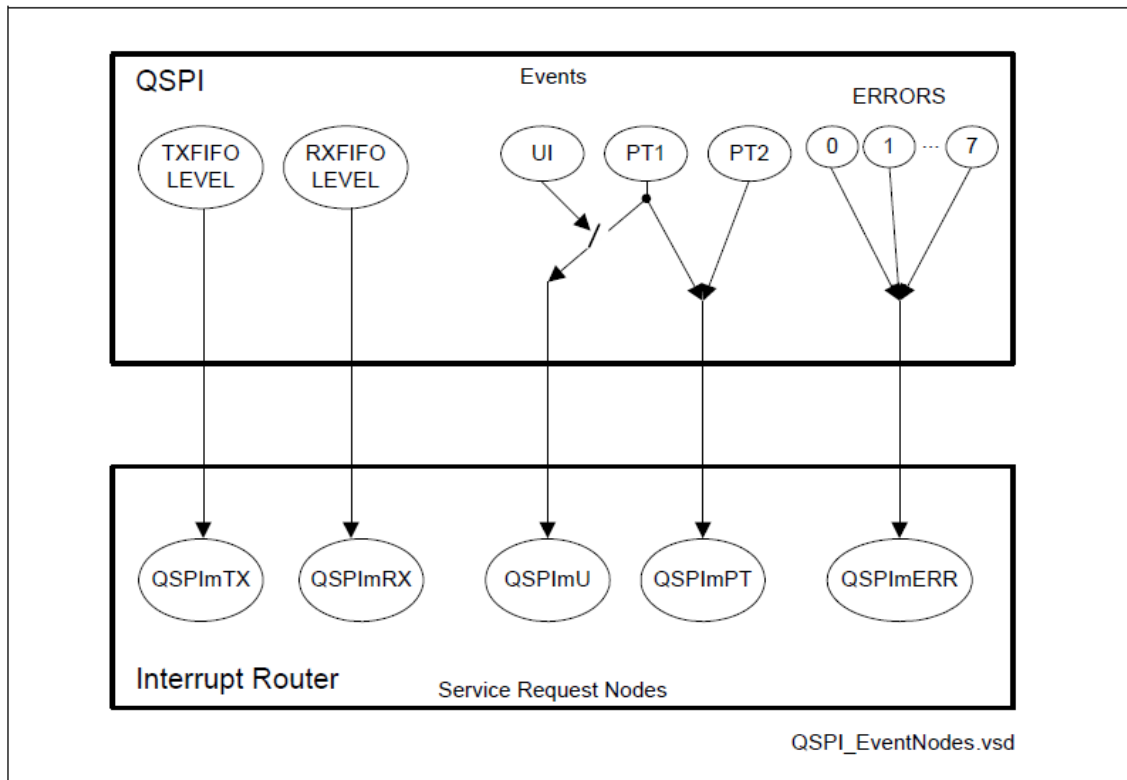


Obrázek 3.9: Loop Back mód[2].

3.3.6 Nastavitelná přerušení

Qspi modul umožňuje nastavit následující přerušení:

- Přerušení spojená s frontou FIFO
 - Odesílání dat (Tx FIFO fronta): požadavek na zápis do fronty
 - Příjem dat (Rx FIFO fronta): požadavek k čtení dat z fronty
- Přerušení spojená s poruchovými hlášenými:
 - Detekce přetečení, podtečení FIFO fronty sloužící k odesílání dat
 - Detekce přetečení, podtečení Rx FIFO fronty sloužící k čtení přijatých dat.
- Přerušení na některou fázi komunikačního cyklu (např. začátek rámce, prázdný registr odesílání/příjmu dat).
- U uživatelem definované přerušení[2].



Obrázek 3.10: Přerušeni generovaná module QSPI[2].

Přerušeni fronty TxFIFO

Nastavení poskytuje dva módy:

- Single Move Mode
- Batch Move Mode
- Combined move mode

Fronty Rx FIFO a Tx FIFO generují přerušeni nezávisle na sobě[2].

(Jednoduchý zápis) Single Move Mode

V tomto režimu se udržuje fronta TxFIFO naplněna, proto se ihned po uvolnění elementu okamžitě plní znovu. Mód podporuje zápis pomocí zápisu jednoho přenosu modulem DMA spouštěného generováním přerušeni TxFIFO. Požadavek na přerušeni se spouští vždy při volném datovém místě ve frontě. Je potřebné spustit inicializační přerušeni zápisem do TxFIFO sloužícího k nastavení příznaku, dále se obsluhuje fronta přerušeni s využitím DMA k zápisu do fronty[2].

(Dávkování) Batch Move Mode

Tento režim využívá obsluhu fronty TxFIFO pomocí CPU. Výhodou tohoto módu je omezení počtu přerušeni, které zde vzniká když z fronty odebráno více dat proto dochází k redukci. Obsluha je spouštěna když hodnota prvků ve frontě je menší jak nastavená hodnota[2].

Důležité je aby se při generování přerušení a následném plnění fronty překročila prahová elementů čekajících k odeslání ve frontě TxFIFO hodnota spouštění přerušení, co zajistí generování nového přerušení[2].

Kombinovaný mód

TxFIFO generuje přerušení když hodnota prvků ve frontě je menší jak nastavený práh. Přerušení se generují dvě, při zápisu konfigurace a při následném zápisu dat. V případě že přerušení vzniklé zápisem do fronty není ukončeno a vznikne následné přerušení zápisem dat první přerušení bude ukončeno[2].

Přerušení fronty RxFIFO

Podobně jako v předchozím případě jsou zde dvě možnosti nastavení přerušení geneovaná frontou RxFIFO:

- Single move mode
- Batch move mode
- Combined mode[2]

Single move mode

V tomto nastavení se udržuje fronta RXFIFO co nejprázdnější. Nastavení umožňuje použití DMA sloužícího k vyčítání hodnot. Přenos dat z fronty pomocí DMA je spouštěn přerušením RxFIFO. Inicializační přerušení je spouštěno zapsání první hodnoty. Dále je spouštění přenosu DMA obsluhováno generováním přerušení[2].

Batch move mode

Výhodou je omezení počtu přerušení které je generováno při větším počtu dat uložených ve frontě RxFIFO. Důležité je aby po vyčtení hodnot fronty byl počet zůstávajících elementů ve frontě nulový nebo pod nastavenou hodnotou kdy se spouští přerušení. Jinak nedojed k přerušení[2].

Combined mode

RxFIFO generuje přerušení vždy když počet dat ve frontě přesahuje přednastavenou hodnotu . V tomto módu se generují dvě přerušení, přerušení detekcí dat a přerušení odesílající příznaky. Pokud není obsluženo přerušení generované příjmem dat a následně se generuje přerušení příznaků, přerušení od příznaků bude ukončeno[2].

3.4 Analogově digitální převodník CIC751

3.4.1 Čtení hodnot z převodníku

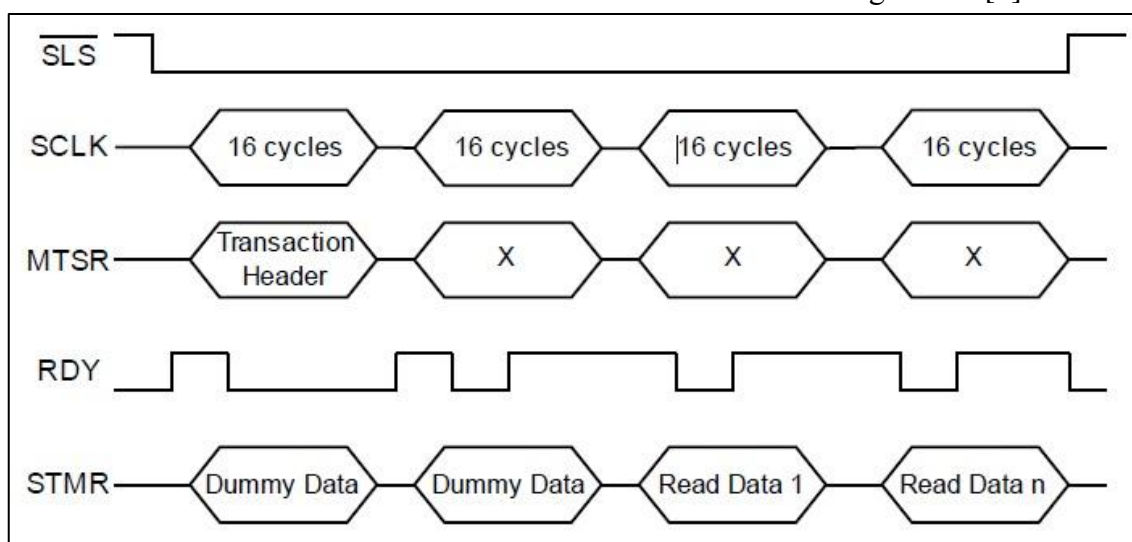
Čtení hodnot z převodníku začíná odesláním inicializačního slova složeného z bitů CMD, INCE a 14bitové adresy. Obvod master musí převodník inicializovat 16 bity než se začne komunikace s obvodem[4].

Operace čtení(CMD = 0):

Každá transakce vyčítání hodnot z převodníku začíná inicializací obvodu převodníku. Bit INCE nastavuje automatickou inkrementaci uvnitř obvodu adresy odeslané při inicializaci. Komunikace SPI by měla být doplněna synchronizačním bitem protože doba vybavení dat obvodu může být různá[4].

Řízení pinu RDY:

- Při změně úrovně z '0' na '1', obvod master začne generovat 16 hodinových impulzů.
- Při změně úrovně z '1' na '0' nedochází k žádnému řídicí signalizaci[4]



Obrázek 3.11: Čtecí sekvence obvodu CIC751[4].

Tabulka 3.1: Adresování obvodu CIC751[4].

| Název bitu | Funkce | Pozice bitu. |
|------------|--|--------------|
| CMD | 0 Sekvence čtení 1 Sekvence zápisu | 15 |
| INCE | Inkrementování adresy po každém přenosu: 0 Zakázáno 1 Povoleno | 14 |
| ADDR | Cílová adresa | 13..0 |

3.5 Driver QSPI s obsluhou modulu DMA

Periferie OSPI nastavená do režimu master využívá k obsluze registrů příjmu RxFIFO a odesílání TxFIFO dat modul DMA, zajišťující přenos přednastavených dat k přenosu a přijímaných dat z připojeného modulu slave. Aktivace DMA a QSPI probíhá pomocí časovače, u nastaveném časovém intervalu, který v přerušení spouští přenos a čtení front RxFIFO a TxFIFO modulu QSPI. Funkce driveru využívají hlavičkových souborů obsahujících deklaraci jednotlivých modulů registrů procesoru.

Integrované periferní moduly procesoru využívající ke své činnosti taktovací hodiny procesoru obsahují registr k povolení propustnosti taktovacích impulsů k vlastní činnosti ze zdroje hodin například komunikační moduly jako SPI, CAN. Moduly obsahující tento registr jsou strukturou přibližně identické a v definici registrů příslušného modulu nese jméno CLC (Clock Control Register).

Hlavní řídicí funkce pro modul jsou:

- Povolení periferních hodin s konstantní frekvencí k vlastní činnosti modulu
- Povolení režimu spánku/nečinnosti v čase kdy není modul využíván

V případě že periferní modul využívající hodin není potřebný v dané konkrétní aplikaci může být zakázán nastavením bitu DISR (Module Disable Request) v registru CLC. Moduly obsahující navíc bity předdělicky RMC musejí být nastaveny jinak je modul v nečinném režimu.

Přímý zápis do registru CLC všech modulů není povolený, zápisem do takého registru vzniká error na sběrnici. Registre CLC jsou chráněny proti zápisu a zápis je možný pouze odblokováním funkcí extern void unlock_wdtcon (void) a nazpět uzamčeny extern void lock_wdtcon (void) definovaných v hlavičkovém souboru #include <machine/wdtcon.h>. Při zápisu do registru se doporučuje číst obsah po zápisu a ujistit se nastavním příznakem který indikuje povolení nebo zakázání.

3.5.1 Nastavení adres uložení přenosu DMA

Při nastavování adres cílového a zdrojového uložení dochází k tomu že adresy báze adres bude vždy začínat buď 0xC0000000 nebo 0xD0000000. Dochází k tomu proto, že multiprocesorová platforma využívá lokální adresování při ukládání proměnných. Proto je nutné přepočítat adresu uloženou do registrů DMA.

Tabulka 3.2:Globální adresy

| Číslo CPU | Program Scratch Pad RAM | Data Scratch Pad RAM |
|-----------|-------------------------|----------------------|
| 0 | 0x7010 0000 | 0x7000 0000 |
| 1 | 0x6010 0000 | 0x6000 0000 |
| 2 | 0x5010 0000 | 0x5000 0000 |

Architektura procesorů TriCore využívá SRAM paměť označenou jako Program Scratch Pad RAM (PSPR) a Data Scratch Pad RAM (DSPR). Lokální paměť PSPR je začíná vždy bázovou adresou 0xC000 0000 a lokální DSPR 0xD000 0000 která se přepočítáním konvertuje na globální adresu podle Tabulky 3.2. Každé procesorové jednotce odpovídá jiná globální bázová adresa v paměti. Bázové adresy globálního adresování jsou vždy využívány při sběrníkových transakcích. Data s lokální bázovou adresou 0xC000 0000 budou vždy uloženy v paměti na adresách 0x5010 0000 – 0x701F FFFF a rovněž data na adresách 0xD000 0000 budou uloženy v paměťovém prostoru 0x5000 0000 až 0x700F FFFF. Příslušná adresa záleží na jednotce CPU viz. Tabulka 3.2.

Aby došlo k správnému přenosu DMA musí se přednastavená adresa konvertovat podle následujícího postupu:

- 1) Proměnnou **adresa** bitově násobíme hodnotou 0xF000 0000, aby se zjistilo do které části lokální adresy proměnná patří lokálního prostoru 0xDxxx xxxx.
- 2) Pokud patří do prostoru 0xDxxx xxxx, potom se adresa pro DMA nastaví podle následujících kroků:
 - Jednotlivé segmenty DSPR příslušného CPU jsou od posunuty o hodnotu 0x1000 0000.
 - **Lokální adresa** se vynásobí hodnotou 0x000F FFF aby se zjistilo paměťové místo v rámci lokálního bloku adres 0xD000 0000
 - Výsledek předchozího kroku se odečítá od hodnoty 0x7000 0000 - (číslo cpu). (konstanta posunu segmentu= 0x1000 0000)
= 0x7000 0000 – cpu(0, 1,2)*0x1000 0000
 - Výsledek druhého bodu se přičte k výsledku třetího bodu

Globální adresa =

(Lokální adresa * 0x000F FFFF) + (0x7000 0000 – cpu(0, 1,2)*0x1000 0000)

- 3) Pokud adresa nepatří do bloku začínajícího se 0xDxxx xxxx, adresa se nemodifikuje.

3.5.2 Implementované funkce driveru

Inicializace všech periférií potřebných k nastavení požadované činnosti komunikace QSPI s využitím DMA se provádí voláním funkce *extern void Init(void)* obsahující inicializační funkce ke konfiguraci jednotlivých modulů.

```
extern void Init(void)
{
    InitOSPI();//inicializace modulu OSPI0
    InitDma_0();//inicializace kanálů 1,2 modulu DMA
    InitDma_1();
    intit_GPTR1_Tim3();//inicializace auto reload timeru
}
```

Popis jednotlivých funkcí:

- **static void InitOSPI(void)** je hlavní inicializační funkce modulu QSPI. Modul ke své činnosti využívá taktovací frekvenci, proto je důležité v registru CLC(Clock Control Unit) nastavit povolení propustnosti hodin modulu bitem Module Disable Request Bit. Pro zpřístupnění zápisu do registru chráněného ENDINIT režimem je nutno použít funkci **unlock_wdtcon ()** která umožní modifikaci bitů. Vykonáním zápisu se registr zabezpečí opětovným uzamčením funkcí **lock_wdtcon ()**. Obě funkce jsou deklarovány v hlavičkovém souboru **wdtcon.h** která je součástí vývojového prostředí. Po zápisu se doporučuje číst obsah zpět k ujištění zapsané hodnoty.

Funkce nastavuje:

- registre předděličky základní frekvence
 - definování časové jednotky bitu t_Q (nastavené množství impulzů z předděličky)
 - definování velikosti bitu složeného
 - porty komunikace
- Funkce **void InitDma_0(void)**, **void InitDma_1(void)** nastavují kanály modulu DMA souřící k obsluze front čtení a zápisu (RxFIFO, TxFIFO) modulu QSPI. Každý kanál je nastaven tak aby se aktivací a následném přenosu dat ze zdrojové do cílové adresy deaktivoval a jeho opětovné spuštění je možné opakovanou aktivací (nastavením bitu).

Funkce nastavuje:

- Adresy zdrojového a cílového uložště
 - Velikost přenášeného rámce dat kanálem
 - Počet transakcí
 - Počet přenosů na jednu transakci
 - Nastavení modifikátoru adres (zapisování do bloku paměti s automatickou modifikací adres)
- Funkce **static void intit_GPTR1_Tim3(void)** modulu General Purpose Timer slouží k aktivaci transakce DMA v nastaveném časovém intervalu. Časovač pracuje v režimu autoreload s rutinou přerušení ve které se aktivuje přenos DMA obsluhující QSPI.

4 REALIZACE ROZHRANÍ CAN A OTESTOVÁNÍ S PC KARTOU

4.1 Definice protokolu CAN podle ISO 11898-1

CAN protokol podle standardního popisu vrstev modelu ISO/OSI. Specifikace 2.0 obsahuje dvě části označené A a B[1].

- Část A popisuje základní standard rámce CAN. Rámec obsahuje 11 bitů sloužících k identifikaci na sběrnici.
- Část B popisuje rozšíření standardního rámce CAN který se hlavně liší délkou identifikačních bitů kterých je v tomto případě 29[1].

Protokol je navrhnut se zpětnou kompatibilitou přenášených rámců. Hlavní vlastnosti struktury CAN protokolu:

- Hierarchické uspořádání správ
- Garantovaná latentní doba
- Flexibilní konfigurace
- Příjem dat z různých zdrojů s časovou synchronizací
- Multimaster řízení
- Detekce chyb
- Automatické opakované vysílání přerušenoho vysílání když je sběrnice vrácena do stavu nečinnosti.
- Automatické odpojení vadného uzlu (CAN node) [1]

| No. of layer | ISO/OSI model | CAN protocol |
|--------------|---------------|--|
| 7 | Application | User specified |
| 6 | Presentation | Blank |
| 5 | Session | Blank |
| 4 | Transport | Blank |
| 3 | Network | Blank |
| 2 | Data link | } CAN protocol (with free choice of medium) |
| 1 | Physical | |

Obrázek 4.1: OSI/ISO standard rámce[1].

Na obrázku 11 je ISO/OSI standard zobrazující základní architekturu komunikačního modelu pro návrh protokolu sítě. Protokol CAN popisuje pouze první dvě vrstvy modelu tj. fyzickou vrstvu 1 a datovou vrstvu 2[1].

4.1.1 Datová vrstva 2

Vrstva je dělena na dvě části:

MAC – identifikátor síťového zařízení, přijímá správy k prezentaci z podvrstvy LLC a odesílá správy k přenosu do podvrstvi LLC[1].

MAC vrstva plní funkci:

- Tvoří rámce odevzdávané do nižší podvrstvy
- Arbitraci
- Potvrzování
- Detekce chyb
- Signalizace chyb[1].

LLC je horní podvrstva linkové vrstvy která je odpovědná za:

- Filtraci správ
- Přetížení na sběrnici
- Procedura obnovy při detekci chyb[1]

4.1.2 Fyzická vrstva 1

Skládá se ze tří podvrstev:

- PLS (Physical signalling)
- PMA (physical medium attachment)
- MDI (medium dependent interface) [1].

Fyzická vrstva specifikuje přenos mezi stanicemi, po které je signál přenášen a elektrické, elektronické, optické vlastnosti systému:

- Reprezentace bitů na cestě přenosu (kódování, časování)
- Bitovou synchronizaci
- Definici elektrických a optických úrovní signálu
- Definici přenášeného média[1].

Referenční dokument standardu vydán firmou Bosch obsahuje detailní popis reprezentace bitu na sběrnici (PLS část), ale neobsahuje, jaké přenosové médium fyzické vrstvy má být použito, proto se transportní vrstva navrhuje s ohledem na danou aplikaci[1].

Operace systému se pro jednoduchost popisuje názvoslovím“ objektová vrstva“ a „přenosová vrstva“ [1].

Objektová vrstva

Úlohou objektové vrstvy je filtrace a zpracování správ a statusů.

Její cílem je:

- Nalézt zprávu která má být odeslána
- Vybrat z přijatých správ správu k zpracování
- Vytvořit rozhraní s aplikační vrstvou s ohledem na hardwar systému [1].

Transportní vrstva

Hlavní úlohou je zpracovávat přenášené rámce:

- Řídit formát přenášeného rámce dat, rychlost přenosu, a arbitraci konfliktů na sběrnici
- Verifikovat přítomnost nebo nepřítomnost chyb
- Signalizovat různé druhy chyb a zachycení chyb
- Schvalovat zprávu
- Potvrzovat zprávy
- Rozhodnout, jestli je sběrnice volná a nový přenos může začít [1].

4.2 Názvosloví používané v dokumentu ISO standardu protokolu

Standard CAN v dokumentu ISO 11898- x(kapitoly 3 a 4 standardu) obsahuje zaužívané názvosloví.

Vysílací uzel stanice (Sending node- station) [1]:

ISO: jednotka připojená do sítě komunikující podle komunikačního protokolu CAN.

- Vysílací uzel (station) generující správy.
- Vysílací stanice se nazývá vysílač dokud je sběrnice v nečinném stavu, nebo stanice ztratila arbitraci na sběrnici[1].

Přijímací uzel (Receiving node station) :

Přijímací uzel je ve stavu kdy na sběrnici probíhá komunikace, nebo když není vysílacím uzlem [1].

Úrovně na sběrnici:

Sběrnice má jednu ze dvou komplementárních úrovní nazývaných dominantní (dominant) a recesivní (recessive) hodnota [1].

Přenosová rychlost sběrnice:

Nastavená přenosová rychlost musí být konstantní a jednotná pro všechny stanice připojené na sběrnici[1].

Správa/rámce/formát:

Přenášení dat po sběrnici probíhá v určitém rámci s omezenou délkou bitů. Když je sběrnice v nečinném stavu, kterákoliv stanice (nodes) může začít nový přenos po sběrnici podle určitých pravidel [1].

Směrování informací:

Na sběrnici CAN stanice nemusí uvažovat přenášené informace, vzhledem ke stanici pro kterou jsou určeny. Zde jsou uvedeny výhody:

- **Flexibilita systému:** stanice (nodes) mohou být připojeny na sběrnici bez modifikace hardwaru nebo softwaru ostatních připojených stanic v síti.
- **Směrování správ:** Přenášené správy obsahují identifikační bity, čímž je každá správa přenášena sběrnici jedinečně.
- **Identifikátor (ID):** Identifikátor správy určuje jednoznačnost správy přenášené mezi připojenými stanicemi v síti. Každý uzel podle ID určuje jestli přenášenou správu přijat nebo ne. Výběr přenášených správ zajišťuje elektronický filtrační obvod[1].

Dokumenty standardu ISO/OSI neobsahují doporučení obsah a architekturu přenášeného rámce[1].

- **Multicast**
V ISO standardu je multicast adresovací mód při kterém šířený rámec adresuje skupinu uzlů sběrnice současně. Po přijetí správy a její filtraci a zpracování mohou přijímané uzly podle potřeby odpovídat na přijatou správu[1].
- **Konzistence dat**
Správa vysílaná po sběrnici může být přijata do všech uzlu, do jednoho uzlu nebo žádným uzlem. Konzistence dat CAN sběrnice se řídí principem multicastu a zpracováním chyb[1].
- **Priorita a koncept přístupu na sběrnici**
Bitově orientovaná arbitrace sběrnice a identifikátor definují statickou prioritu správy před jejím přístupem na sběrnici[1].
- **Datový rámec, rámec žádosti o data, přijímaná data.**
 - Datový rámec přenáší data.
 - Rámec žádosti o data a datový rámec obsahují stejný identifikátor. Rámec žádosti o data je odpovědí stanici/uzlu která žádala o získání dat z jiného uzlu/stanice[1].
- **Multimaster operation-** provoz s více hlavními stanicemi
Každý uzel na sběrnici může zahájit vysílání pokud je sběrnice v nečinném stavu. Při zahájení vysílání současně více stanicemi získá sběrnici jednotka s nejvyšší prioritou na sběrnici může vysílat data[1].
- **Arbitrace**
Při nečinném stavu na sběrnici a následném vysílání všech stanic současně vznikne konflikt řešený pomocí bitové (nedestruktivní) arbitrace počas rozhodování. Tato metoda arbitrace zajišťuje bezztrátovost času a informace.

Kolize může vzniknout při současném vysílání správy se stejným identifikátorem, v tomto případě má vyšší prioritu datový rámeček před rámečkem žádost o data. V průběhu žádosti o sběrnici každý vysílač porovnává hodnotu odeslaného bitu s předpokládaným bitem k odeslání. V případě shodných úrovní, uzly pokračují ve vysílání. V případě odeslání recesivního bitu a detekci dominantní úrovně, uzel ztrácí arbitraci a dále nevysílá[1].

- **Bezpečnost přenosu**

Zajištění bezpečnosti přenosu dat je implementováno detekcí chyb, pomocí signálů a vlastního testu zařízení[1].

- Detekce chyb – error detection:
Monitorování sběrnice-vysílač kontroluje zda napěťový úroveň mají požadovanou hodnotu
- Přítomnost CRC – kontrolování rámečků
- Bit Stuffing metoda[1].

- **Signalizace chyb a zotavovací čas:**

Všechny správy afektované chybou jsou signalizovány v každém uzlu příznakem (flag), proto jsou zamítnuty a automaticky vysílány opakovaně. Zotavovací čas, časový usek od začátku detekce chyby do momentu kdy je odeslána nová správa, není delší jak 29 bitů pro CAN 2.0A(31 bitů pro CAN 2.0B), když není žádná chyba detekována[1].

- **Zachycené poruchy**

Tento typ chyb vzniká působením vnějších vlivů. Stanice na sběrnici musí být schopna rozpoznat okamžité náhodné poruch od poruch běžných. Zjištěním této poruch se stanice vypne a odpojí od sběrnice[1].

- **Přístupový bod**

Topologie sériové komunikace CAN se skládá z jednotek – uzlů připojených na sběrnici. Počet připojených jednotek je teoreticky neomezený, prakticky je maximální počet jednotek určen dobou zpoždění při přenosu a zátěží při které je sběrnice schopna pracovat[1].

- **Potvrzování**

Všechny stanice po přijetí správy kontrolují konzistenci přijaté správy kterou potvrdí nebo odešlou chybný příznak[1].

Přenos správ

Komunikace CAN rozlišuje čtyři rozdílné druhy přenášených rámečků s různými časovými intervaly mezi rámečkami:

- Datový rámeček, přenáší data z vysílací do příjímající stanice

- Rámec žádost o data, stanice vyšle požadavek o získání dat z jiné stanice, identifikátory správ jsou stejné.
- Rámec poruch, správa vysílaná na sběrnici v okamžiku detekování poruchy
- Rámec přetížení, požadovaný při prodloužení časového intervalu mezi předcházejícími daty a následujícími daty.
- Mezirámeček, odděluje datové rámec a rámec žádosti o data od předcházejících rámců[1].

4.3 CAN bit

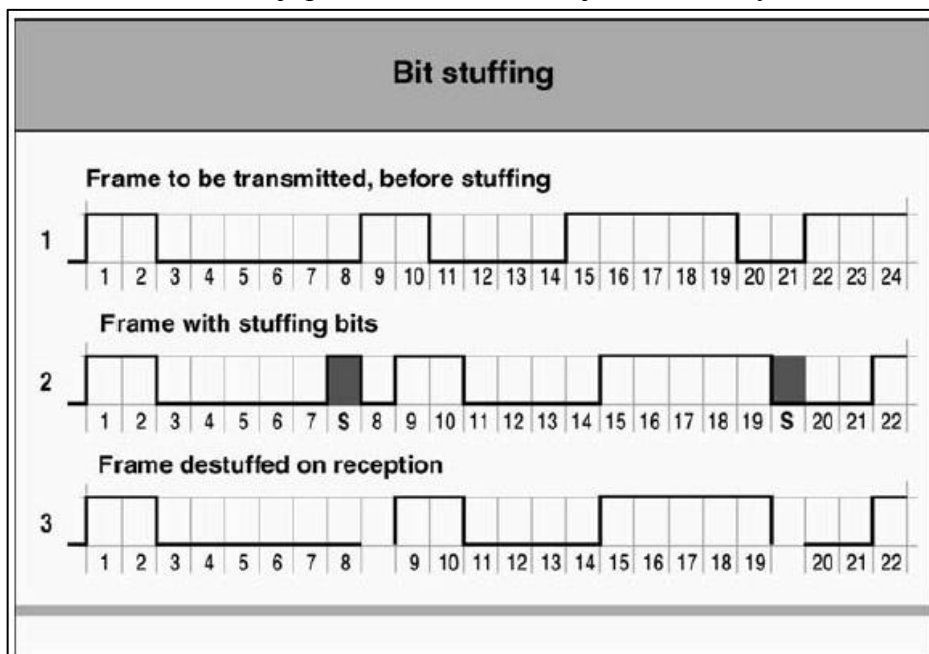
Kódování bitů přenášejících data na sběrnici:

NZR (non return to zero) je jednou z nejčastěji používanou metodou při kódování.

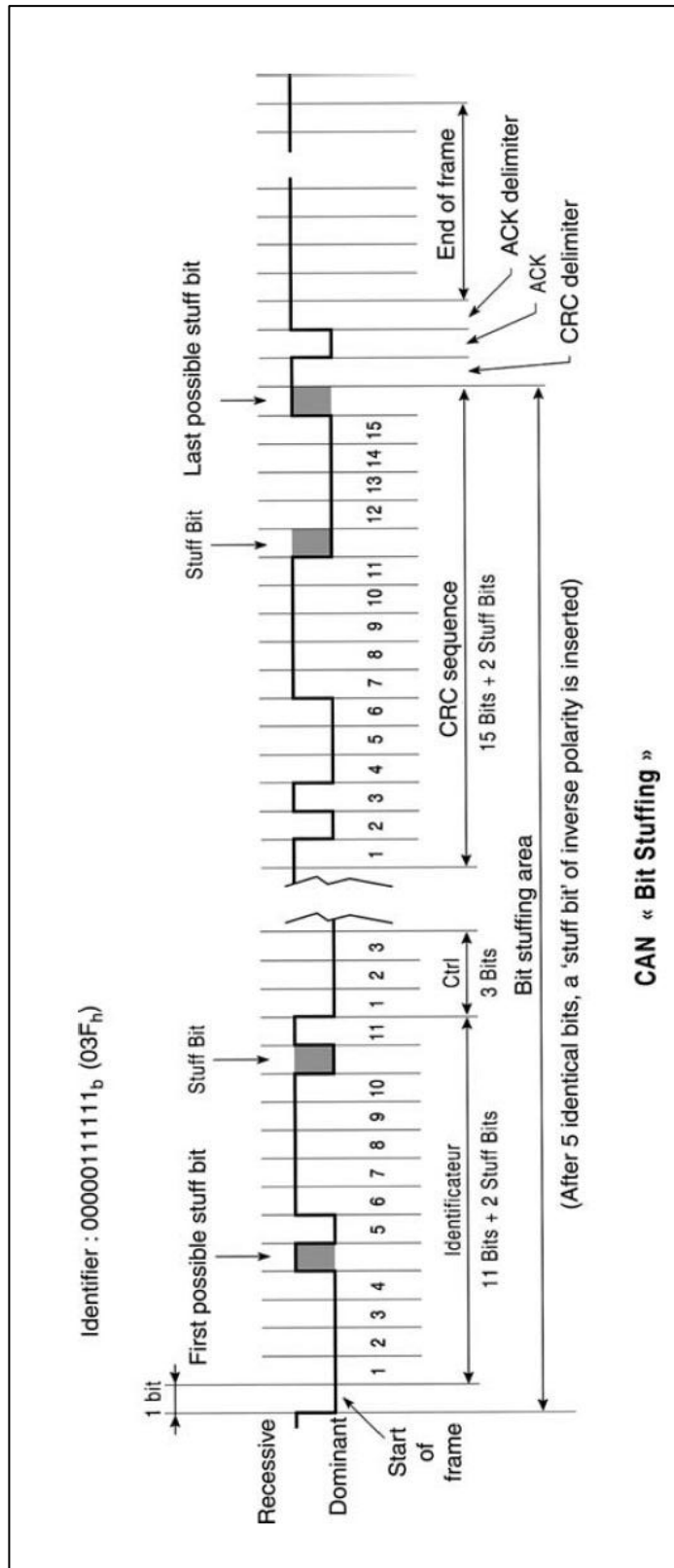
Princip: úroveň generovaného (dominantní nebo recesivní) bitu zůstává po definovanou časovou jednotku bitu konstantní[1].

Vkládání bitů (bit stuffing)

Přenášením bitů stejné hodnoty sběrnici může vzniknout pro přijímací stanice neobvyklý jev. Z důvodu bezpečnosti jsou přenášené bity jsou doplňovány automatickým vkládáním bitů při detekci určitého počtu bitů stejné úrovně. Detekcí 5 bitů stejné úrovně se vkládá navíc jeden další bit s opačnou polaritou(recesivní nebo dominantní), který indikuje správnost přenosu. Po přijetí přijímací stanicí je vložení bit odstraněn. Vkládání bitů je realizováno pouze pro pole arbitrace, kontrolní, datové a CRC. Ostatní části rámce mají pevnou strukturu a nejsou kódovány vkládáním bitů[1].



Obrázek 4.2: Vkládání bitů (Bit stuffing) [1].



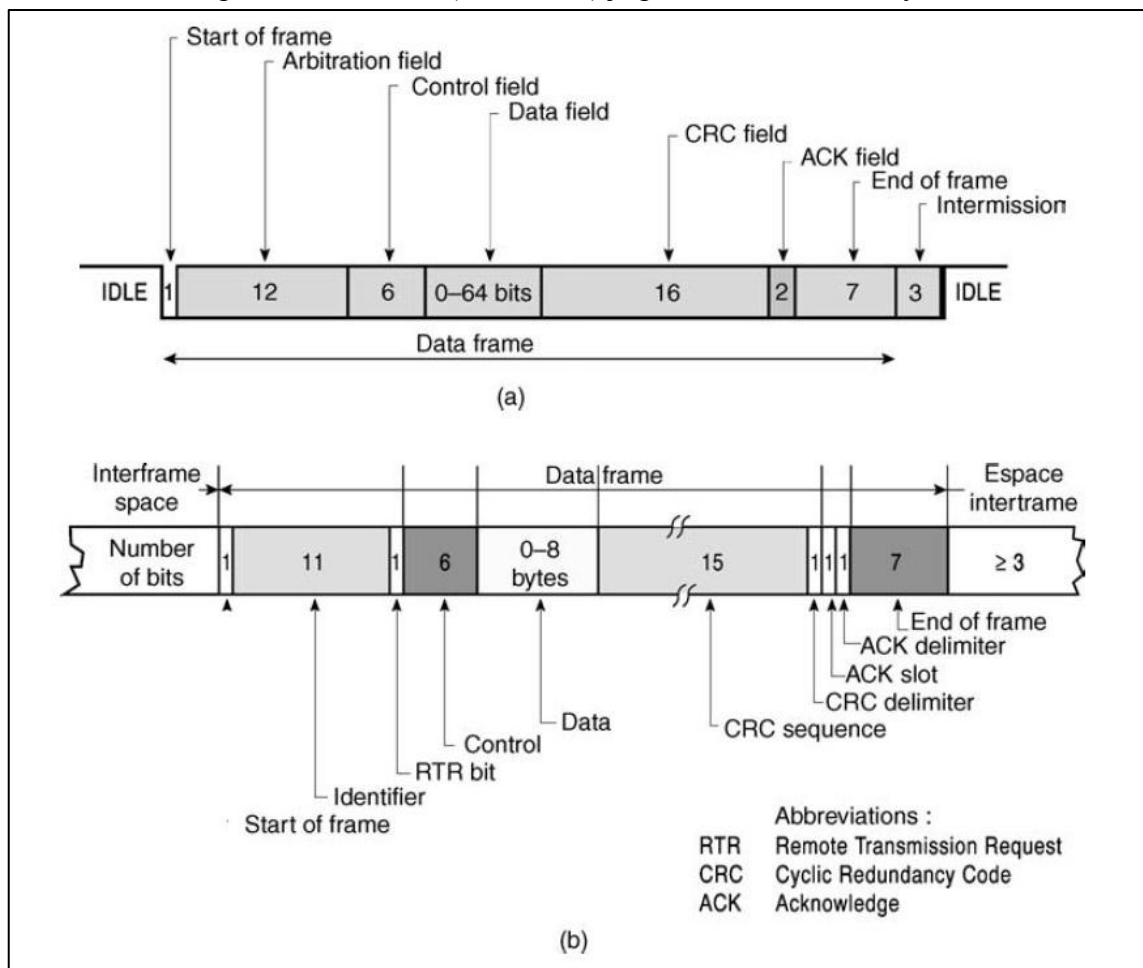
Obrázek 4.3:Vkládání bitů(Bit stuffing)[1].

4.4 Struktura přenášeného rámce

Struktura nejpoužívanějšího přenášeného rámce označovaného podle standardu 2.0A se dělí na bloky bitů rozdílné nebo nastavitelné délky a rozdílného řídicího významu[1].

Rámec se dělí na bloky bitů určitého významu:

- Začátek rámce (start of frame)
- Pole bitů arbitrace (arbitration field)
- Kontrolní pole (control field)
- Pole bitů dat (data field)
- CRC pole bitů (CRC sequence)
- ACK pole (ACKnowledgement)
- Konec rámce (end of frame)
- Poslední pole, mezi rámec (interframe) je přidáván automaticky[1].



Obrázek 4.4: Datový rámec - CAN 2.0A[1].

Začátek rámce (Start of frame):

Je tvořen jedním dominantním bitem který oznamuje všem přijímacím stanicím (uzlům) zahájení vysílání na sběrnici. Zahájení vysílání může nastat pouze v případě kdy je sběrnice v nečinném stavu. Všechny stanice musí být před zahajovacím bitem navzájem synchronizováni[1].

Pole bitů arbitrace (Arbitration field)

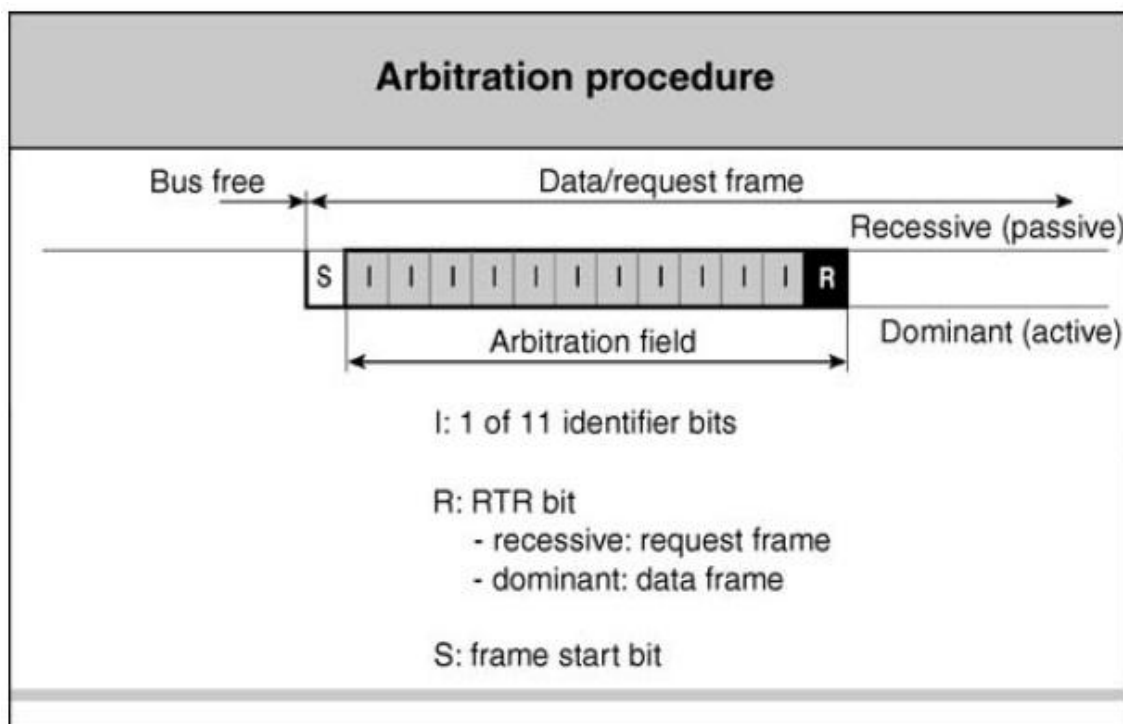
Pole bitů představující identifikátor kterým se řídí prioritá správ následované bitem RTR(remote transfer request) bit indikující žádost o data[1].

- Identifikátor, je složen z 11 bitů. Bity jsou odesílání v pořadí od nejvyššího platného bitu po nejmenší a 7 bitů (10 - 4) nesmějí svou úrovní všechny recesivní

Maximální možný počet identifikátorů může být[1]:

$$2^{11}-2^4=2048-16=2032$$

- RTR bit, bit musí být dominantní při žádosti o data[1].



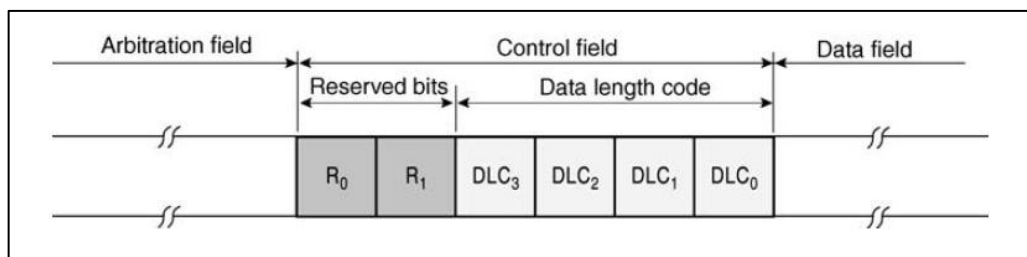
Obrázek 4.5: Pole bitů identifikátoru(Arbitration field)[1].

Kontrolní pole bitů (Control field)

Složeno z 6 bitů:

Rezervované bity

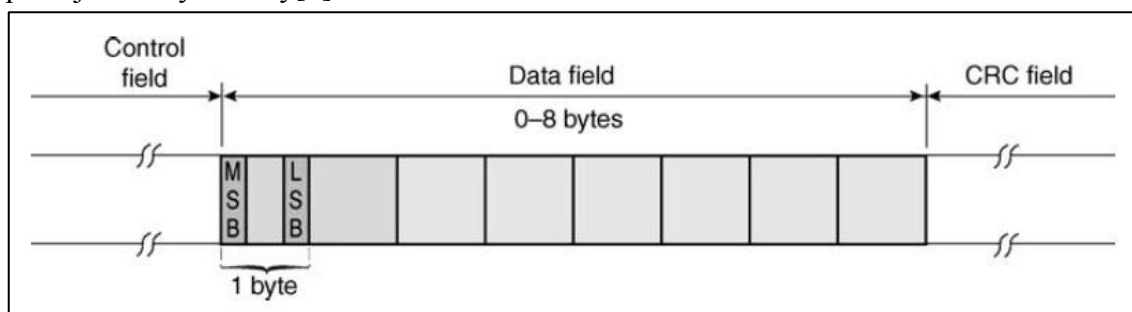
- První dva bity (r0, r1) jsou rezervovány pro kompatibilitu s prodlouženým rámcem podle standardu CAN 2.0B který může přijímat i standardní rámeček (CAN 2.0A).
- Počet bajtů přenášených rámcem je uloženo ve zbylých čtyřech bitech DLC3-DLC0[1].



Obrázek 4.6:Kontrolní pole(Control field[1].

Datové pole:

Obsahuje data sloužící k zpracování dalšími stanicemi/uzly připojenými na sběrnici. Délka tohoto pole může být 0 až 8 bajtů, odesílaných v pořadí od nevýznamnějšího bitu po nejméně významný[1].



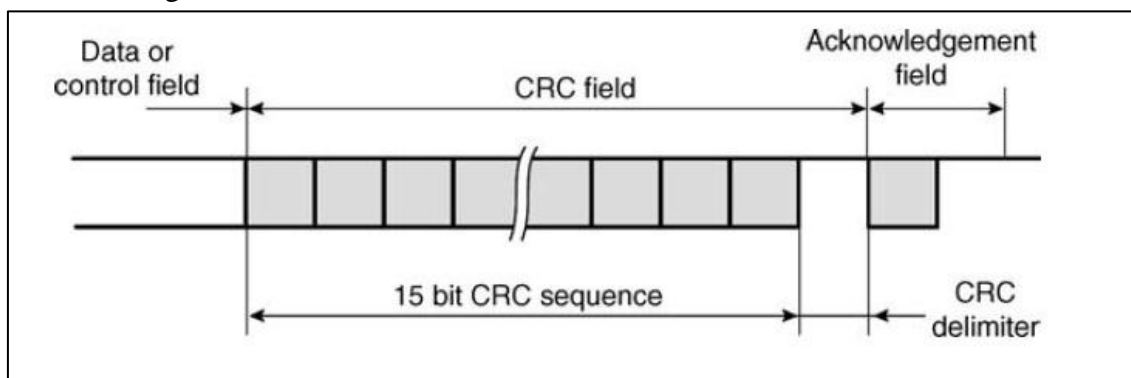
Obrázek 4.7:Datové pole bitů[1].

CRC(cyclic redundancy code) pole:

Pro zajištění bezpečnosti odesílaných zpráv, všechny zprávy musejí obsahovat CRC sekvenci generovanou odesílanou stanicí uloženou do rámce.

Kódy sběrnice CAN jsou podporované kontrolou prity s následujícími atributy:

- Maximální délka kódu je 127 bitů
- Maximální počet informačních číslic: 112 bitů
- Délka CRC sekvence: do 15 bitů
- Hammingova vzdálenost: $d = 6$ [1].



Obrázek 4.8:CRC pole (Cyclic Redundancy Code) [1].

Hammingova vzdálenost, mezi dvěma binárními slovy stejné délky, je počet binárních prvků na stejných pozicích s opačnou hodnotou[1].

CRC sekvence je počítána vysílací stanicí podle následující procedury:

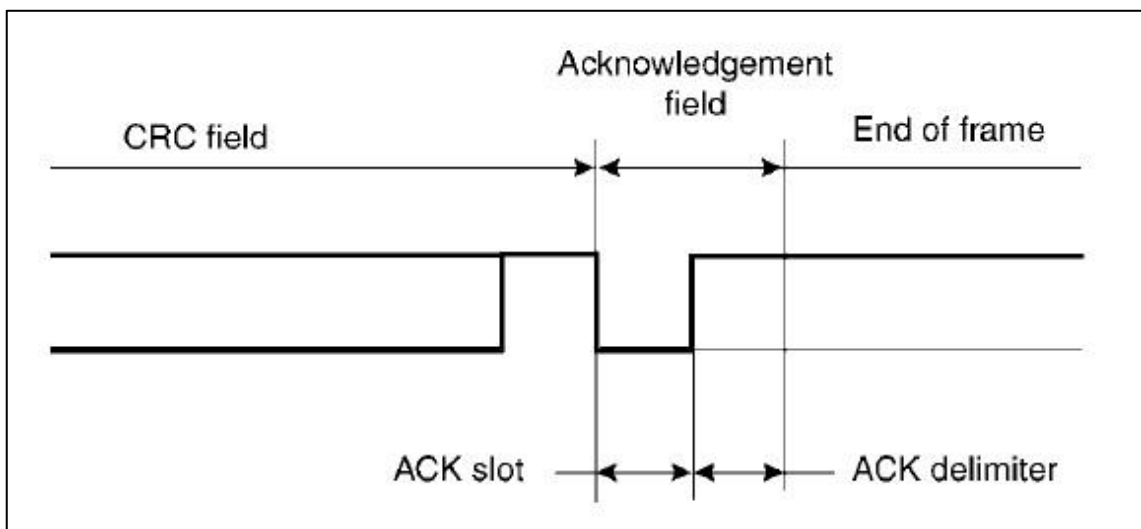
1. Bity rámce od počátku tj. začátek rámce(start of frame) až do konce přenášeného bloku dat (end of frame) jsou interpretovány jako polynomiální funkce $f(x)$ s koeficienty 0 a 1 při jednotlivých mocninách polynomu určujících jejich přítomnost nebo nepřítomnost, 15 posledních bitů je doplněno nulami.
2. Polynom formulován podle výše uvedené struktury je dělen celočíselným dělením (modulo 2) polynomem $g(x)$.

$$g(X) = X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1 \quad (7)$$

3. Dělením polynomu původního rámce $f(x)$ polynomem $g(X)$ vzniká CRC sekvence[1].

Potvrzovací pole (Acknowledgement field)

Blok obsahuje dva bity ACK slot a ACK delimiter. Při přenosu dat, vysílací stanice odešle dva recesivní bity.



Obrázek 4.9:Potvrzovací pole(Acknowledgement) [1].

- ACK slot
Kdykoli přijímací stanice správu při které nevznikne chybné hlášení a správa je přijata jako platná, je potvrzena nazpět vysílací stanici dominantním bitem kterému předchází recesivní bit[1].
Příjem potvrzovacího bitu vysílací stanici znamená, že alespoň jedna stanice na sběrnici přijala správu bez chyby. Všechny přijímací stanice musí potvrdit příjem správy na sběrnici pokud nedetekovaly chybu, jinak bude odeslaná správa

považován za neplatnou a přijímací stanice neodešlou potvrzení správného příjmu[1].

Pokud je správa neplatná musí obsahovat chybu a podle CAN protokolu, přijímací stanice chybné správy musí signalizovat chybu odesláním rámcem poruchy[1].

- ACK delimiter
Tento bit musí být vždy recesivní. Když je přijatá správa platná (bez detekce poruchy), ACK slot bit je obklopen dvěma recesivními bity CRC delimiter a ACK delimiter[1].

Blok bitů konec rámce (End of frame)

Datový blok je zakončen příznakem složeným ze sekvence 7 recesivních bitů[1].

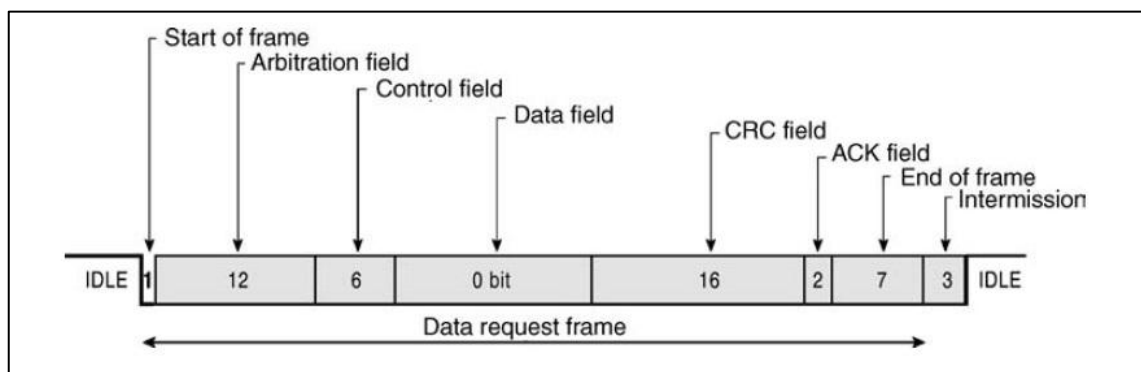
Mezi rámeček (Interframe)

Rámeček žádosti o data

Vysílací stanice může žádat o data, rámeček v tomto případě obsahuje následující části k inicializaci požadavku[1]:

- Začátek rámce (start of frame)
- Identifikátor (Arbitration field)
- Kontrolní pole (control field)
- CRC pole
- Potvrzení (Acknowledgement)
- Konec rámce (end of frame)

Konec rámce je následovaný posledním polem, mezi rámečkem (Interframe). Ostatní části rámce jsou shodné s rámečkem dat[1].



Obrázek 4.10: Rámeček žádosti o data[1].

4.5 Vlastnosti, detekce a zpracování poruch na sběrnici

Typy poruch:

- 1) Chyby fyzické vrstvy
 - Přenášený bit může být postihnut např. vnějším rušením
 - Chyba při vkládání bitů

- 2) Chyby přenášeného rámce
 - CRC omezovač (delimiter)
 - ACK delimiter error
 - Konec rámce
 - Error delimiter
 - Přetížení

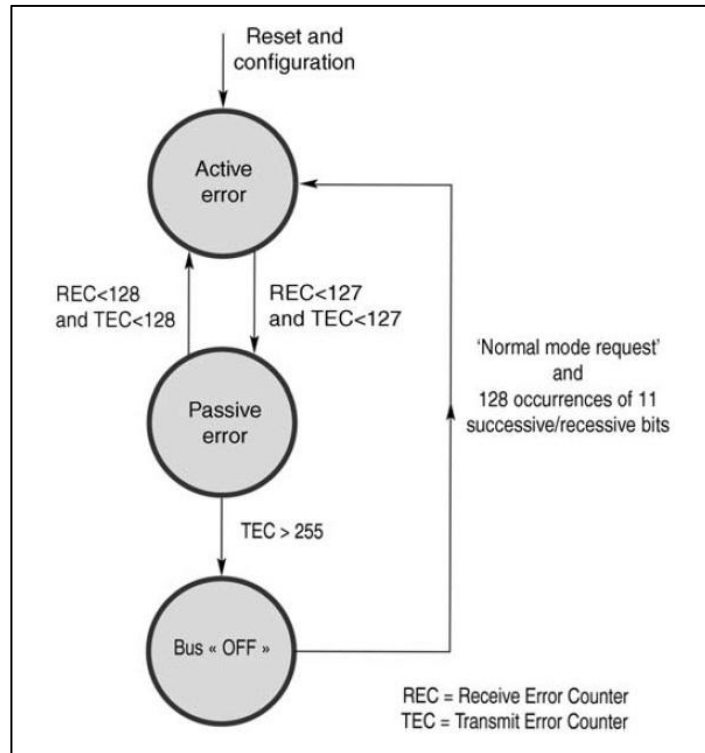
Uvedené poruchy budou signalizovány rámcem poruch (error frame) [1].

4.5.1 Zpracování zachycených chyb

Ohlášení zachycené chyby se signalizuje nejvyšší prioritou na sběrnici stanicí nejbližší od místa vzniku chyby. Všechny stanice připojené musejí být vybaveny dvěma oddělenými čítači, čítač poruch přenosu a čítač poruch příjmu, s úlohou zaznamenávat chybu přenosu nebo příjmu inkrementováním příslušného čítače[1].

- V případě přenosu nebo příjmu správy bez poruchy je příslušný čítač dekrementován
- V případě detekce poruchy je příslušný čítač inkrementován

Poměr inkrementování nebo dekrementování čítačů závisí na poměru správných a chybných správ. Inkrementační nebo dekrementační hodnota je nastavena na hodnotu 8[1].



Obrázek 4.11: Čítače REC a TEC[1].

Hodnota čítačů $<0;127>$, error active state:

Při hodnotě čítačů v mezích $<0;127>$ se stanice nachází v módu aktivní poruchy (error active). V tomto případě stanice nepřijímá ani neodesílá správy. Odesílá pouze příznak chyby (active error flag) pomocí rámce chyb[1].

Hodnota čítačů $<128;255>$, error passive state:

Při hodnotě kteréhokoli z čítačů v rozmezí $<128;255>$, stanice přejde do stavu error passive state. V tomto stavu stanice nepřijímá ani neodesílá správy pouze vysílá na sběrnici příznak poruchy (passive error flags) pomocí rámce poruch (error frame) [1].

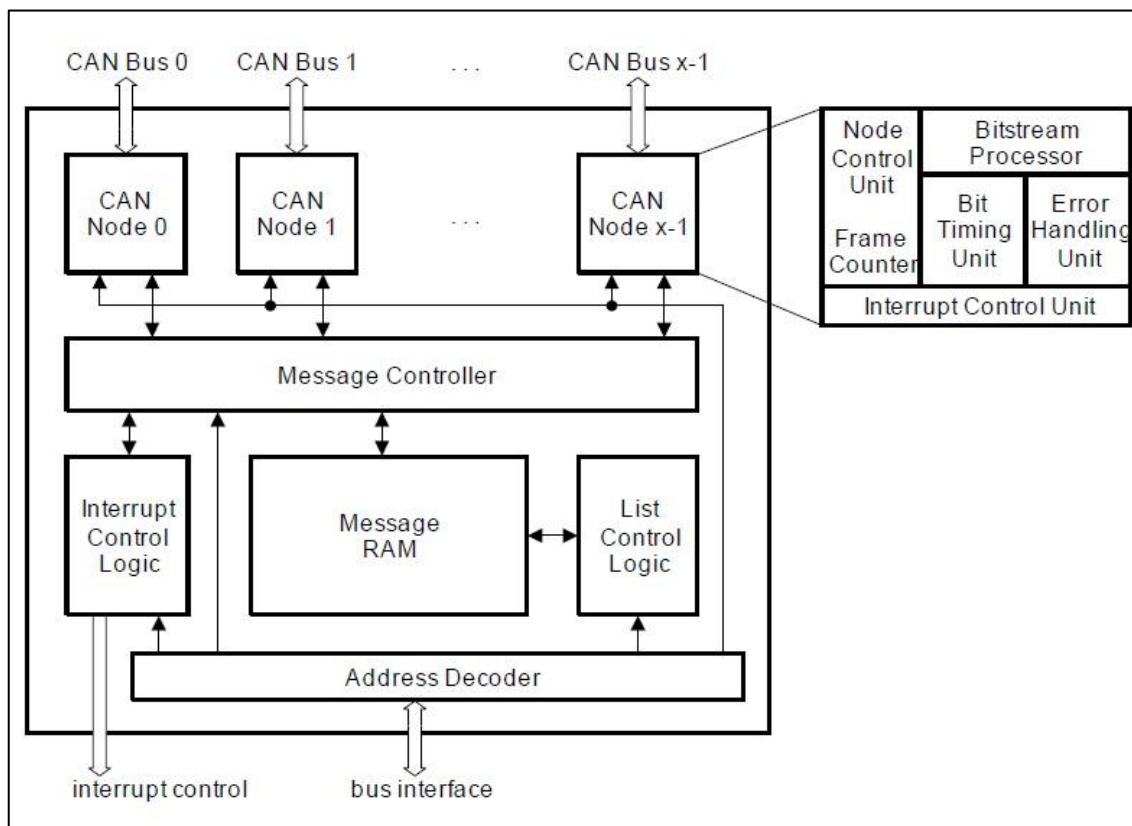
Hodnota čítačů ≥ 256 :

Stanice s hodnotou čítačů nad hodnotou 255 přejde do stavu bus off, odpojí se od sběrnice[1].

4.6 Periferie procesoru MultiCAN+

Procesor obsahuje jeden periferní modul komunikačního protokolu CAN. Modul obsahuje čtyři komunikační uzly. Každý uzel je připojen na sběrnici pomocí dvou vodičů TxD a RxD. Podporuje oba standardy přenášených rámců 2.0A, 2.0B. Všechny komunikační uzly sdílejí 256 správ(objektů). Každý objekt může být přiřazen do

jednoho komunikačního uzlu. Správy jsou organizovány v obousměrném přepojeném seznamu. Blokové schéma viz. Obrázek 22[3].



Obrázek 4.12: Blokové schéma modulu MultiCAN[3].

Popis jednotlivých bloků:

CAN uzel (CAN Node) tvoří následující části:

- **Vysílací procesor (Bitstream processor):** zprostředkovává vysílání dat připravených v registrech k odeslání nebo příjmu,
- **Jednotka pro nastavení bitu (Bit timing unit):** Určuje délku bitu na sběrnici, bod vzorkování.
- **Jednotka obsluhy poruch (Error Handling Unit):** obsluhuje registry čítačů poruch.
- **Řídící jednotka uzlu (Node Control Unit):** řídí povolení vysílání uzlu, nastavuje požadavku přerušování detekcí např. úspěšný přenos rámce
- **Jednotka přerušování (Interrupt control unit):** řídí přerušování generované uzlem[3].

Řadič správ zajišťuje výměnu rámců mezi uzlem a správou uloženou v paměti[3].

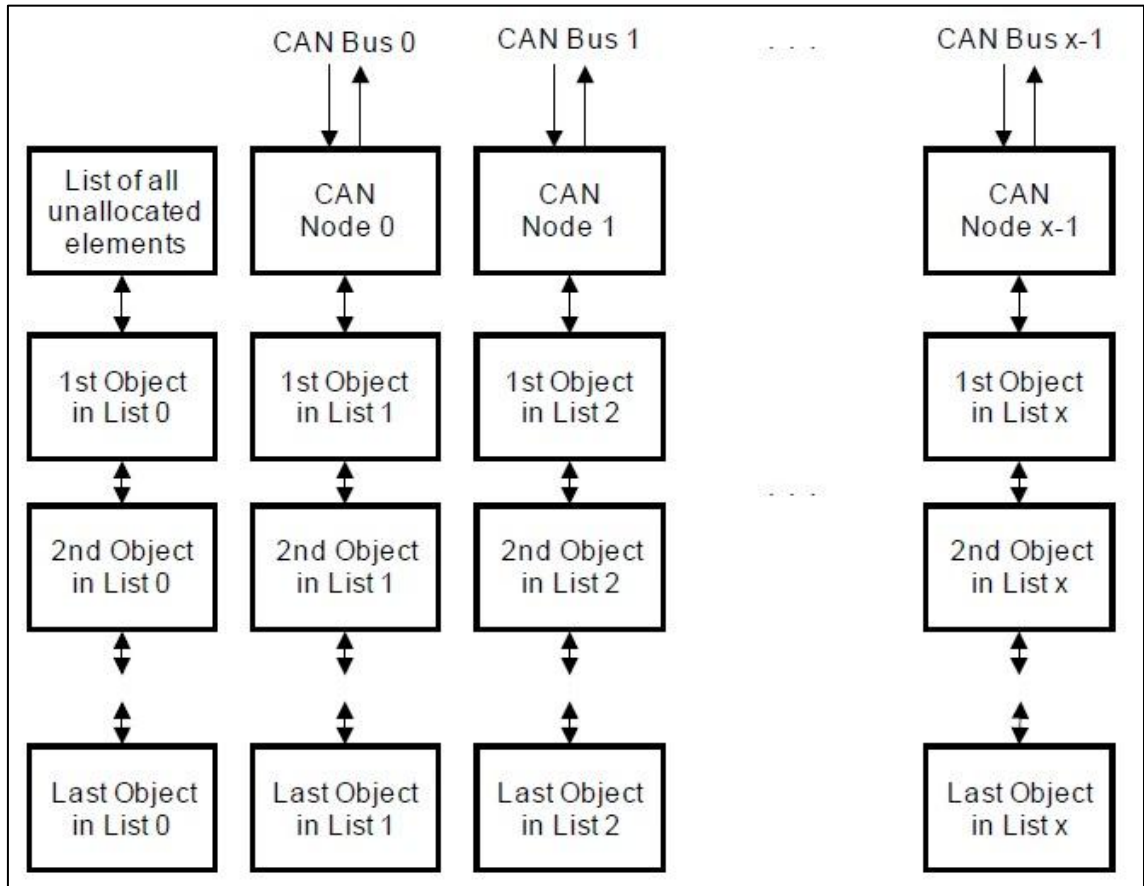
Řadič seznamu správ: pouze tento řadič může modifikovat do struktury seznamu. Zajišťuje alokaci, dealokaci správ do seznamu pomocí příkazů[3].

Řízení přerušování: každé přerušování je svázáno s výstupem generujícím přerušování.

Připojená přerušování může být 16 a více zdrojů přerušování může aktivovat stejné přerušování[3].

4.6.1 Struktura seznamu správ

Správy jsou uloženy v obousměrném seznamu kde každá správa má ukazatel na předchozí a následující správu. Modul MultiCAN poskytuje 16 seznamů. Každý uzel je spojen s jedním seznamem a přijímané a odesílané správy musejí být alokovány. Modul poskytuje více seznamů jak uzlů proto správy alokovány do seznamů které nepatří uzlu nemůžou být přímo odesílány ani přijímány[3].



Obrázek 4.13: Organizace správ v seznamu[3].

Modifikace správ se provádí pomocí řadiče uvnitř modulu. Řadič je řízen pomocí příkazů a umožňuje uživateli pracovat se správami. Před zápisem je nutné kontrolovat příznak jestli řadič není v činnosti[3].

4.7 Driver MultiCAN+

Implementované funkce k nastavení periferie.

- ***void ClcCanEnable(void)*** funkce nastavující povolení propustnosti časovacích hodin pro modul zápisem do registru CLC. Registr je chráněn endinit přístupem, proto je důležité použít k zápisu funkce ***void unlock_wdtcon (void)/void lock_wdtcon (void)***. Dalším nastavením je předdělička modulu.
- ***void NewMessageAllocation(unsigned char MessageNumber, unsigned char ListIndex, unsigned char Command)*** Každý CAN uzel pracuje s vlastním seznamem správ sloužících k odesílání nebo přijímání. Každá správa k odeslání nebo přijímání daným uzlem musí být alokována do seznamu příslušného CAN uzlu. Zápis do seznamu uzlu není možný přímo ale prostřednictvím příkazů řídicího řadiče který vykonává alokaci správy do listu. Řadič správ určující k alokaci do seznamu CAN uzlu potřebuje dostat následující informace:
 - Číslo seznamu CAN uzlu do kterého má být správa alokována
 - Číslo správy k alokaci do seznamu
 - Instrukce pro řadičVykonáním předchozích kroků se správa (objekt) alokuje do seznamu příslušného CAN uzlu a může být dále nastavena podle požadovaného použití.
- ***void SetUpMessage(unsigned int MessageNumber, unsigned int TransmitReception)*** funkce nastavuje parametry alokované správy přidělené do určitého seznamu uzlu. Parametry určují jestli je správa určena k odeslání nebo příjmu, délku přenášených dat, identifikátor správy určující jeho prioritu a formát rámce podle standardu CAN 2.0A nebo CAN2.0B
Nastavují se zde parametry:
 - Číslo správy
 - Správa určena k vysílání nebo příjmu
- ***void InitIrqCanN(void)*** nastavuje přerušování příjmu nebo odeslání správy generované příslušnou správou povolovacími bity. Jedno přerušování může být spouštěno více správami.
- ***void LoadDataToMessage(void)*** obsahuje podmínku jestli je správa určena k vysílání pomocí nastavených příznaků. V případě splnění podmínky se inicializují data určena k přenosu.
- ***void UnlockWrite2CanNode(void)*** funkce obsahuje nastavení uzlu do stavu kdy se odpojí od sběrnice a umožnění zápisu do registrů

- ***void LockkWrite2CanNode(void)*** funkce je komplementární funkcí k předchozí, zde se komunikační uzel připojí na sběrnici a další zápis do specifických registrů není umožněno
- ***void SendMessage(void)*** voláním funkce se odešle datový rámeček
- ***void RxCAN1(int arg)*** funkce přerušení která se aktivuje přijatým datovým rámečkem příslušné správy.

5 ZÁVĚR

Moje závěrečná práce nebyla tématem semestrálního projektu na který navazuje diplomová práce. Zvolení tématu je z důvodu získání nových znalostí a zájmu o mikroprocesorovou techniku s využitím aktuálně moderní technologie AURIX Application kit s tří jádrovým procesorem TC275 v dané oblasti.

K nastavení periférií jsou použity pouze hlavičkové soubory obsahující definici registrů procesoru které jsou součástí vývojového prostředí Free AURIX Entry Tool chain. Ovladače sériové komunikace QSPI s obsluhou DMA a ovladač modulu CAN řeší pouze základní nastavení periférií. Navržené programy jsou funkčně odzkoušeny pouze v testovacím režimu který umožňuje modul dané periferie a můžou sloužit jako základní struktura nastavení jiných modulů procesoru. Z důvodů problémů vzniklých při oživování modulů se nepodařilo reálné ověření připojení s pc kartou CAN a analogově digitálním převodníkem CIC751. Řešení problémů jako mapování paměti spojeno s přesunem dat pomocí DMA transakcí využíváno modulem QSPI, způsob odesílání a nastavení správ k odesílání nebo přijímání a alokace konkrétnímu komunikačnímu uzlu CAN za přínosné.

Literatura

- [1] PARET, Dominique. Multiplexed networks for embedded systems: CAN, LIN, flexray, safe-by-wire.. Hoboken: John Wiley, 2007, xiii, 418 s. ISBN 978-0-470-03416-3.
- [2] INFINEON. Application Kit TC2X5: Hardware: APPLICATION KIT TC2X5 V2.0. Munich, 2013.
- [3] INFINEON. Aurix TC27x C-STEP: 32-Bit Single-Chip Microcontroller. Munich, 2014.
- [4] INFINEON. CIC751 Companion IC [online]. 2005 [cit. 2015-05-16]. Dostupné z: http://www.infineon.com/dgdl/CIC751_UM_v10_print.pdf?fileId=db3a304412b407950112b40aab6c0642
- [5] Freescale Semiconductor, Inc. [online]. [20 januar 2000] [cit. 2015-05-16]. Dostupné z: <http://www.engr.colostate.edu/ECE251/S12SPIV4.pdf>
- [6] Wikipedia. Serial Peripheral Interface Bus. [5 May 2015]. Dostupné také z: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
- [7] Infineon. Highly Integrated and Performance Optimized: 32-bit Microcontrollers for Automotive and Industrial Applications [online]. 2015 [cit. 2015-05-16]. Dostupné z: http://www.infineon.com/dgdl/Infineon-Tricore+Family+BR_2015-BC-v01_00-EN.pdf?fileId=db3a30431f848401011fc664882a7648
- [8] HIGHTEC EDV-SYSTEME GMBH. *Toolchain User Guide v4.6.5.0*. Saarbrücken, 2014.

Seznam skratek

| | |
|-------|--|
| CAN | controlled areaNetwork |
| PMA | physical medium |
| MAC | medium acces |
| LLC | logic link control |
| PLS | physical signalling |
| PMA | physical medium attachment |
| MDI | meduim dependent interface |
| OSI | open system interconect |
| ISO | international standardization organization |
| CRC | cyclic redundancy code |
| NZR | non return to zero |
| SPI | serial peripheral interface |
| SCLK | serial clock |
| MTRS | master transmit |
| MRST | master receive |
| SS/CS | slave select/chip select |
| QSPI | Queued Synchronnous Peripheral interface |
| IR | interrupt router |
| ICU | interrupt control unit |
| DMA | direct memora access |
| SRI | ahared resource interconnect |
| SPB | system peripheral bus |
| CRC | cyclic redundancy code |
| DSP | digital signal processor |

Seznam obrázků

| | |
|--|----|
| Obrázek 2.1: Blokové schéma vývojové desky AURIX Application kit[2]. | 10 |
| Obrázek 2.2: Blokové schéma procesoru TC275[3]. | 11 |
| Obrázek 3.1: Blokové schéma DMA periferie[3]. | 13 |
| Obrázek 3.2: Řídící blok transakce[3]. | 14 |
| Obrázek 3.3: Blokové schéma principu činnosti QSPI[2]. | 18 |
| Obrázek 3.4: Komunikační fáze cyklu[2]. | 19 |
| Obrázek 3.5: Časování a vzorkování bitu[2]. | 20 |
| Obrázek 3.6: Předělička (nastavení přenosové rychlosti) [2]. | 21 |
| Obrázek 3.7: Architektura Tx a RX FIFO[2]. | 22 |
| Obrázek 3.8: Single configuration[2]. | 23 |
| Obrázek 3.9: Loop Back mód[2]. | 24 |
| Obrázek 3.10: Přerušení generovaná module QSPI[2]. | 25 |
| Obrázek 3.11: Čtecí sekvence obvodu CIC751[4]. | 27 |
| Obrázek 4.1: OSI/ISO standard rámce[1]. | 31 |
| Obrázek 4.2: Vkládání bitů (Bit stuffing) [1]. | 36 |
| Obrázek 4.3: Vkládání bitů (Bit stuffing) [1]. | 37 |
| Obrázek 4.4: Datový rámec - CAN 2.0A[1]. | 38 |
| Obrázek 4.5: Pole bitů identifikátoru (Arbitration field)[1]. | 39 |
| Obrázek 4.6: Kontrolní pole (Control field)[1]. | 40 |
| Obrázek 4.7: Datové pole bitů[1]. | 40 |
| Obrázek 4.8: CRC pole (Cyclic Redundancy Code) [1]. | 40 |
| Obrázek 4.9: Potvrzovací pole (Acknowledgement) [1]. | 41 |
| Obrázek 4.10: Rámec žádosti o data[1]. | 42 |
| Obrázek 4.11: Čítače REC a TEC[1]. | 44 |
| Obrázek 4.12: Blokové schéma modulu MultiCAN[3]. | 45 |
| Obrázek 4.13: Organizace správ v seznamu[3]. | 46 |

Seznam tabulek

| | |
|---|----|
| Tabulka 3.1: Adresování obvodu CIC751[4]. | 27 |
| Tabulka 3.2: Globální adresy | 29 |