

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra Elektroniky

**Mikropočítačový řídicí systém se signálovým
procesorem TMS320F28335**
Microcomputer Control System with DSP
TMS320F28335

Zadání diplomové práce

Student: **Bc. Dominik Jurčo**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2612T015 Elektronika

Téma: **Mikropočítačový řídicí systém se signálovým procesorem
TMS320F28335
Microcomputer Control System with DSP TMS 320F28335**

Zásady pro vypracování:

1. Popište strukturu signálového procesoru TMS320F28335 s ohledem na využití ve výkonových polovodičových systémech.
2. Proved'te technický popis konkrétního mikropočítačového systému a jeho připojení k výkonové části polovodičového měniče.
3. Vytvořte softwarové vybavení pro možnost tvorby algoritmů pro výkonové polovodičové systémy a zapracujte jej do systémů knihoven.
4. Vypracujte podrobnou technickou dokumentaci pro využití tohoto systému ve výukové a výzkumné činnosti Katedry elektroniky.

Seznam doporučené odborné literatury:

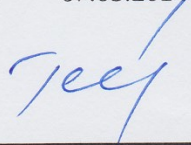
Dle pokynů vedoucího závěrečné práce

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

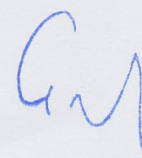
Vedoucí diplomové práce: **doc. Ing. Petr Palacký, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



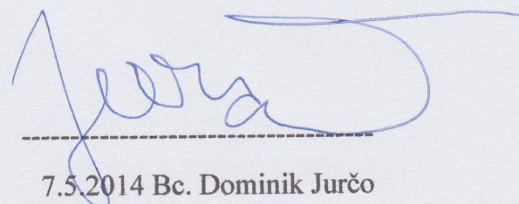
doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení:

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární parametry a publikace, ze kterých jsem čerpal.



7.5.2014 Bc. Dominik Jurčo

Abstrakt

Tato diplomová práce se zabývá popisem řídicího systému se signálovým procesorem TMS320F28335, pro využití ve výukové a výzkumné činnosti katedry Elektroniky. V hlavní části je popsán signálový procesor z hlediska jeho vnitřní struktury a dále jsou podrobněji popsány principy funkce nejvyužívanějších periférií. V přílohách je pak popsán reálný vývojový kit, který byl vyroben katedrou Elektroniky a osazený právě tímto procesorem. Jedná se o popis částí, konkrétních konektorů a pinů. Dále je k této práci přiloženo DVD s potřebným softwarovým vybavením pro obsluhu vývojového kitu. V ostatních přílohách jsou pak manuály pro práci s veškerým softwarovým vybavením.

Po nastudování této práce by měla být dotyčná osoba schopna samostatně pracovat a vyvíjet řídicí systém za pomoci vývojového kitu s tím, že podrobnější popis musí pak již samostatně studovat z dokumentace výrobce.

Klíčová slova

Signálový procesor, Texas Instruments, SPI, SCI, Řídicí systém, TMS320F28335.

Abstract

This thesis deals with the description of the control system with digital signal processor TMS320F28335, for use in teaching and research activities of the Department of Electronic. The main part of the thesis are described signal processor, in terms of its internal structure, and are further described in detail the principles functions of the most used peripherals. The appendix is then described real developmental kit that was made by the Department of Electronics and fitted just this processor. This is the description of parts specific connectors and pins. This work is accompanied by a DVD with the necessary software for operate with the developmental kit. In other annexes are then manuals to work with all software. After finishing of study this work should be the person able to work independently and develop control system with using the developmental kit, but the more detailed description must study in the manufacturer's documentation.

Key Words

Digital Signal Controller, Texas Instruments, SPI, SCI, Micro Computer control System, TMS320F28335.

Seznam použitých zkratek:

TMS – TMS320F2833

PLL – fázový závěs

CPU – procesor

DSC – Signálový procesor – Digital Signal Controller

FPU – Plovoucí řádová čárka – Float Point unit

ROM – paměť určená pouze ke čtení - read only memory

U – elektrické napětí [V]

OBSAH:

1. Mikropočítačový řídicí systém s TMS320F28335	3
1.1 Úvod	3
2. Procesor TMS320F28335	4
2.1 Úvod do procesoru	4
2.2 Popis jednotlivých bloků procesoru	5
2.2.1 C28x CPU	5
2.2.2 Vnitřní sběrniceový systém	5
2.2.3 Paměti SARAM	5
2.2.4 Paměti Flash	5
2.2.5 Blok zabezpečení – Code security module	6
2.2.6 Bootování – Boot ROM	6
2.2.7 GPIO	7
2.2.8 Řadič přímého přístupu do paměti DMA	7
2.2.9 Oscilátor, PPL a Watchdog	7
2.2.10 CPU časovače – CPU Timers	7
2.2.11 Blok PIE	7
2.2.12 ADC modul (A/D převodník)	7
2.2.13 SPI rozhraní	8
2.2.14 SCI rozhraní	8
2.2.15 McBSP	8
2.2.16 ePWM	8
2.2.17 eCAP	8
2.2.18 eQEP	8
2.2.19 eCAN	9
3. Popis principu funkce nejvyužívanějších periférií	10
3.1 Časování TMS	10
3.2 GPIO rozhraní	10
3.3 Systém přerušení	14
3.3.1 Nastavení PIE bloku	15
3.4 CPU časovače – CPU Timers	17
3.5 PWM moduly	18
3.5.1 Time Base (TB)	19

3.5.2 Synchronizace mezi moduly	21
3.5.3 Counter Compare (CC).....	22
3.5.4 Action Qualifier (AQ).....	24
3.5.5 Dead Band generator (DB)	24
3.5.6 Chopper (PC).....	25
3.5.7 Trip Zone (TZ).....	28
3.5.8 Event Trigger (ET).....	29
3.6 A/D převodník – ADC modul.....	30
3.7 SCI rozhraní	32
3.8 SPI rozhraní.....	35
4. Závěr.....	39
5. Seznam použité literatury.....	40

1. Mikropočítačový řídicí systém s TMS320F28335

1.1 Úvod

Tato práce popisuje způsob, jak pracovat s řídicím systémem se signálovým procesorem TMS320F28335. Katedrou elektroniky byl vyvinut vývojový kit osazený právě tímto procesorem. V hlavní části diplomové práce je vypracován první bod zadání, který je zaměřený na popis samotného signálového procesoru. Pro práci s vývojovým kitem je zapotřebí rozumět struktuře procesoru, aby uživatel věděl o jeho funkcích, způsobu jejich nastavení, a následně je pak mohl využít ve svých aplikacích. V samotném textu jsou pak uvedeny odkazy na přílohu č.2, kde jsou uvedena některá nastavení daných periférií v jazyku C.

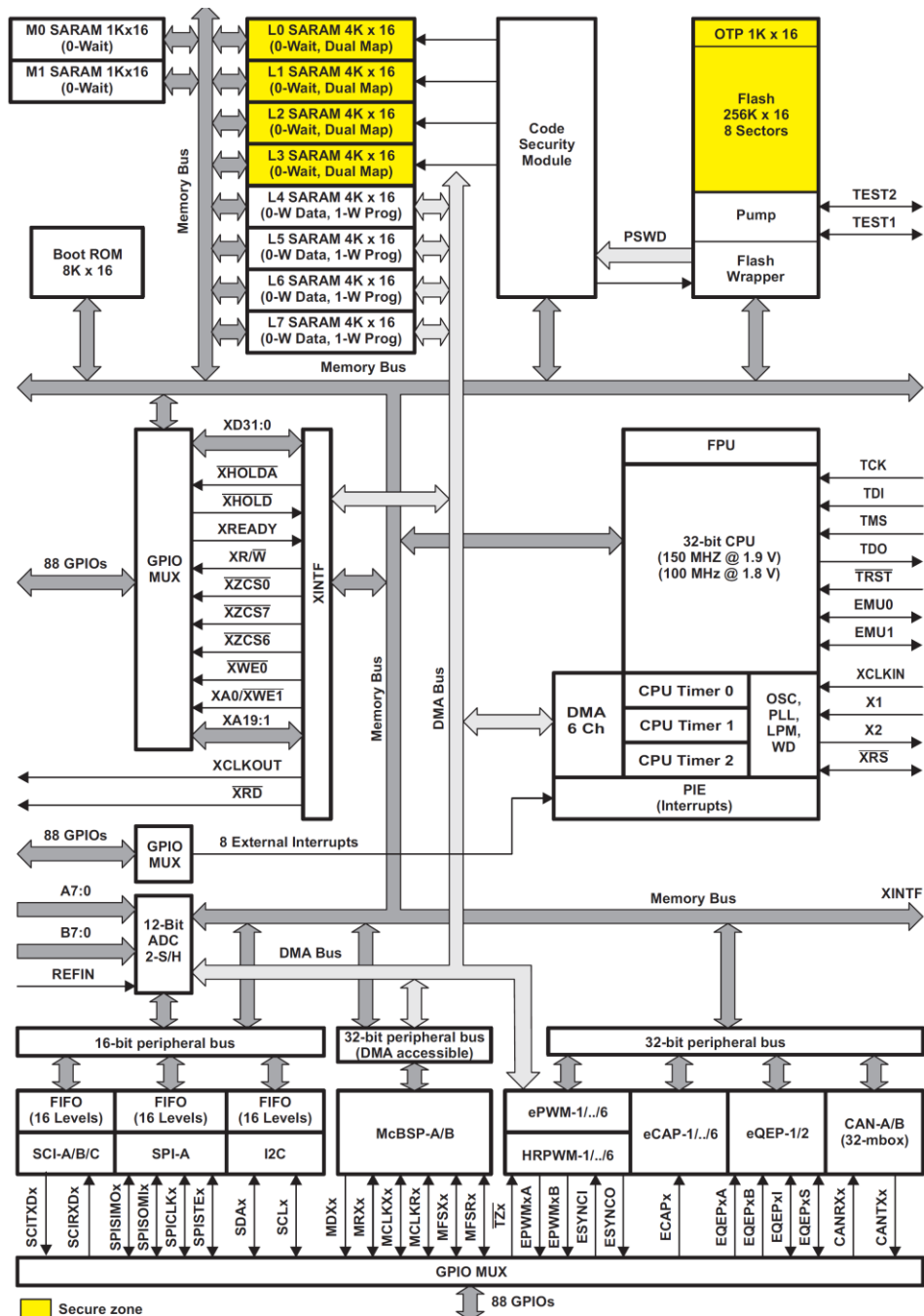
Druhý bod zadání je vypracován v příloze č.1, kde je uveden popis struktury vývojového kitu. Jsou zde popsány jednotlivé desky a konektory pro připojení externích zařízení.

V rámci třetího bodu zadání, je k diplomové práci přiloženo DVD, na kterém jsou uloženy všechny potřebné knihovny se základním programem pro obsluhu řídicího systému. Funkce jednotlivých příkazů v obsluhujícím programu jsou okomentovány. Popis zdrojových souborů, funkcí, knihoven a postup při práci s obslužným programem je uveden v příloze č.4. V páté příloze jsou uvedeny způsoby, jak přidat ovládací prvky v programu Labview, se kterým komunikuje obslužný program řídicího systému. Tato příloha také spadá do čtvrtého bodu zadání, v jehož rámci je vytvořen návod jak pracovat s programem Code composer studio 4 (příloha č.3). Tento program slouží jako programovací prostřední, které bylo vyvinuto přímo výrobcem signálového procesoru, firmou Texas Instruments.

2. Procesor TMS320F28335

2.1 Úvod do procesoru

TMS320F28335 je 32-ti bitový signálový procesor o frekvenci 150Mhz, schopný výpočtů v plovoucí řádové čarce s vícestupňovým systémem přerušení a širokou škálou periférií. Jeho vnitřní architektura je harvardského typu, což znamená, že používá více sběrnicevých systémů. Na obrázku č. 1 je zobrazena jeho vnitřní architektura, která je všeobecně popsána v kapitole 2.2. [1]



(Obrázek č. 2.1 – Vnitřní architektura TMS.)

2.2 Popis jednotlivých bloků procesoru

2.2.1 C28x CPU

Signálový procesor TMS320F28335 (dále pouze TMS) patří do rodiny s architekturou C28x. Je to rodina kontrolérů s DSC architekturou (DSC – digital signal controller), počítající v plovoucí řádové čárce (FPU) s 32- ti bitovou šířkou sběrnice.

TMS se nastavuje za pomoci programovacího jazyka vyšší úrovně C/C++, což umožňuje vytvářet softwarovou část řídicího systému na vysoké úrovni a vyvíjení složitých matematických algoritmů. Vysoký výkon odstraňuje nutnost využití dvou procesorů v mnoha systémech. [1]

2.2.2 Vnitřní sběrnice systém

Stejně jako u jiných DSC procesorů, je i zde využito k přesunu dat mezi pamětmi, perifériemi a procesorem vícenásobný systém sběrnic (multiple busses). Struktura paměťové sběrnice zahrnuje sběrnice pro: čtení programu (22 adresních linek a 32 datových), vyčítání dat a zápis dat (32 adresních a 32 datových). Více násobná sběrnice architektura běžně nazývaná Harvardská, umožňuje C28x načítat instrukce, číst a zapisovat hodnoty dat v jednom procesu.

Sběrnice systém periférií se skládá z 16-ti adresových a 16-ti nebo 32-ti datových linek a příslušných řídicích signálů. To vše je sjednoceno do jedné sběrnice. [1]

2.2.3 Paměti SARAM

Jsou paměti, u kterých je možné v jednom cyklu provést buď operaci čtení anebo zápisu s libovolným přístupem k paměťovým buňkám.

Bloky těchto pamětí M0 a M1 (viz obr.1.1) o velikosti 1Kx16 je možno využít dvěma způsoby. První způsob je klasické využití jako paměť pro data. Druhé využití je jako pro registr procesoru, kdy tyto paměti slouží pro program. Paměti L0 –L7 o velikosti 4Kx16 slouží pro ukládání dat, kdy L0 až L3 mohou být bezpečnostně chráněny. K druhé polovině tj. L4 až L7 je možno přistupovat za pomoci obvodu DMA. [1]

2.2.4 Paměti Flash

TMS obsahuje flash paměť o velikosti 256K x16 (viz obr.1), rozdělenou po 32K do 16-ti sektorů a jednoho samostatného sektoru OTP o velikosti 1x16K určeného pro

zabezpečení dat (uložení hesla). Uživatel může vymazat jeden sektor paměti, zatím co ostatní zůstanou beze změn. [1]

2.2.5 Blok zabezpečení – Code security module

Kontrolér je vybaven vysokou ochranou proti reversnímu inženýrství, což v tomto případě znamená zpětné získání řídicího algoritmu ze zakoupeného výrobku. TMS je chráněn 128-mi bitovým šifrováním. V praxi to znamená, že pro přístup do chráněných oblastí musí uživatel správně zapsat heslo, které obsahuje 16 znaků z ASCII tabulky. Chráněnými oblastmi TMS jsou paměti flash a čtyři bloky paměti SARAM (L0-L3), kdy je heslo rozděleno na dvě části, a to tak, že prvních 64 bitů jsou určeny pro paměti Flash a dalších 64 bitů pro paměti SARAM. [1]

2.2.6 Bootování – Boot ROM

Je blok, který udává s jakým nastavením softwaru má procesor nabootovat při obnovení napájení. Uživatel si zvolí, zdali se má procesor spustit obvyklým způsobem, či si má stáhnout nový software z externího zdroje anebo zdroje interního. V pamětech Boot ROM jsou také obsaženy standartní tabulky pro generování funkcí (např. sinus, cosinus). Zdroj bootování se nastavuje logickou kombinací připojenou na čtyři specifické GPIO porty a mohou být následující: [1]

Mód	GPIO				Typ bootování
	87	86	85	84	
F	1	1	1	1	Jump to Flash
E	1	1	1	0	SCI-A boot
D	1	1	0	1	SPI-A boot
C	1	1	0	0	I2C-A boot
B	1	0	1	1	eCAN-A boot
A	1	0	1	0	McBSP-A boot
9	1	0	0	1	Jump to XINTF x16
8	1	0	0	0	Jump to XINTF x32
7	0	1	1	1	Jump to OTP
6	0	1	1	0	Parallel GPIO I/O boot
5	0	1	0	1	Parallel XINTF boot
4	0	1	0	0	Jump to SARAM
3	0	0	1	1	Branch to check boot mode
2	0	0	1	0	Branch to Flash, skip ADC calibration
1	0	0	0	1	Branch to SARAM, skip ADC calibration
0	0	0	0	0	Branch to SCI, skip ADC calibration

(Tabulka č. 2.2.6 – Nastavení zdroje bootování.)

2.2.7 GPIO

General-purpose input/output. (Univerzálně využitelný vstup a výstup). Většina signálů od periférií (jejich výstupy) je připojena k procesoru pomocí univerzálních pinů GPIO. V případě, že se dané signály či periférie nevyužívají, lze jednotlivý GPIO pin nakonfigurovat pro vstup signálů z externích zdrojů. [1]

2.2.8 Řadič přímého přístupu do paměti DMA

Poskytuje metodu přenosu dat mezi periférií a paměťmi bez účasti procesoru, čímž je možno zvýšit výpočetní výkon systému. Navíc tento modul uspořádává přenášená data do struktury, optimalizované pro zpracování procesorem. [1]

2.2.9 Oscilátor, PLL a Watchdog

Kontrolér může být časován buďto externím oscilátorem anebo interním krystalem umístěným na čipu, kdy se frekvence externího oscilátoru upravuje v PLL bloku.

Hodiny typu watchdog jsou určeny ke zjišťování chybového stavu, přičemž jeho funkce je následující: Na watchdog se musí pravidelně přivádět nulovací signál. Pokud se tak nestane, tak tento blok generuje reset signál k procesoru. Tuto funkci procesoru je však také možno vypnout. [1]

2.2.10 CPU časovače – CPU Timers

Jsou 32-ti bitové časovače s 16-ti bitovou předděličkou. Čítače zde fungují v režimu dekrementace, kdy při dosažení nuly generuje časovač na svém výstupu signál pro přerušení TINT. Výstupy časovačů 1 a 2 jsou přivedeny přímo do procesoru (výhodné pro operace v reálném čase), a časovač 0 je přiveden do bloku PIE. [1]

2.2.11 Blok PIE

Tento blok slouží k výběru jednoho z 96-ti přerušení, kdy každý zdroj přerušení má vyhrazeno místo v paměti pro jeho vektor. Všechna přerušení jsou uskupena do dvanácti bloků po osmi. Celkem 58 zdrojů přerušení je zde prezentováno perifériemi. [1]

2.2.12 ADC modul (A/D převodník)

TMS má v sobě integrován 12-ti bitový A/D převodník. Tento převodník má dva kanály s osmi vstupy. Může však také fungovat v režimu jednobitového převodníku se 16-ti vstupy. Převodník je oproti klasickému vylepšen o různé funkce, jako jsou například simultánní převod (převod dvou kanálů v jednom čase), sekvenční převod či vícenásobný převod jednoho kanálu. [1]

2.2.13 SPI rozhraní

TMS obsahuje jeden modul pro komunikaci v režimu SPI, což je vysoko rychlostní synchronní sériové periferní rozhraní, které je možno využít pro komunikaci s ostatními řídicími prvky, či perifériemi. SPI rozhraní komunikuje v režimu master/slave, kdy zařízení master řídí komunikaci. SPI rozhraní je vybaveno FIFO pamětmi pro příjem i vysílání. [1]

2.2.14 SCI rozhraní

V TMS jsou integrovány tři moduly pro komunikaci ve standardu SCI, což je sériové asynchronní komunikační rozhraní, známé také jako UART. Vysílání i příjem jsou vybaveny FIFO pamětmi jako prevence proti přetížení. Toto rozhraní je možno nastavit pro standard RS232 a samozřejmě je také možné i nastavení s vlastní specifikací, vyhovující standardu SCI. [1]

2.2.15 McBSP

Je univerzální komunikační rozhraní, které lze nastavit pro více standardů komunikace. Těmito rozhraními jsou: T1/E1, IOM2, AC97, I2S, SPI. [1]

2.2.16 ePWM

TMS obsahuje šest modulů pro vytváření pulsně šířkové modulace, které jsou velice flexibilně nastavitelné a schopné pracovat téměř bez zásahu procesoru. Všechny moduly jsou vzájemně propojeny pro případnou synchronizaci. [1]

2.2.17 eCAP

Těchto modulů se využívá v systémech, kdy je důležité nastavení přesného načasování s vnějším prostředím. Což může být např. měření rychlosti rotačních strojů, či měření času mezi dvěma pulsy impulsního snímače. Při plné frekvenci procesoru, může tento modul dosahovat rozlišení 6,7ns. [1]

2.2.18 eQEP

Tento modul slouží k připojení inkrementálních čidel, kdy po připojení čidla se ze vstupních signálů vypočítává poloha a rychlost otáčení měřeného zařízení. Data vyhodnocená tímto modulem se pak dále využívají v řídicím algoritmu. [1]

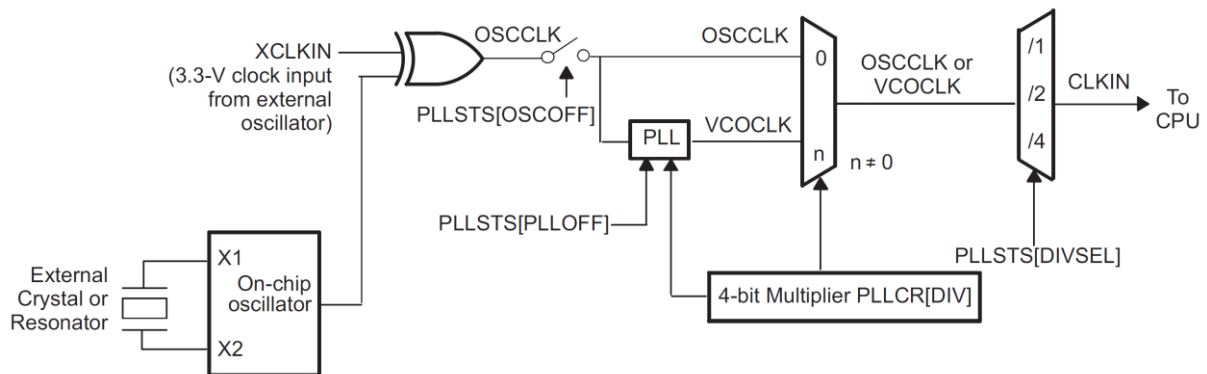
2.2.19 eCAN

TMS obsahuje modul pro realizaci komunikace na základě standardu CAN 2.0B. Využívá se převážně pro sběr dat v elektricky hlučných prostředích, jelikož má vysokou odolnost proti vnějšímu rušení. Nejtypičtější využití je v automobilovém průmyslu. [1]

3. Popis principu funkce nejvyužívanějších periferií

3.1 Časování TMS

Signál CLKIN vstupuje do procesoru a tím udává základní takt. Tento signál je pak z procesoru vyveden jako SYSCLKOUT, což je hlavní taktovací signál pro celé zařízení. Princip nastavení základního taktovacího signálu je vidět na obrázku č 3.1



(Obrázek č. 3.1.1 – Princip časování TMS)

Pro taktování kontroléru se mohou použít tři zdroje časovacích hodin a to:

- Oscilátor umístěný přímo na čipu.
- Externí oscilátor připojený k XCLKIN se vstupním napětím $U=3,3V$.
- Externí krystal připojený k pinům X1 a X2 se vstupním napětím $U=1,9V$.

Časovací signál z oscilátoru je zde nazván jako OSCCLK, který je možno vypnout pro celé zařízení v registru PLLSTS[OSCOFF]. Pokud není za potřebí tento signál dále upravovat, přivádí se rovnou na vstup děličky (/1; /2; /4;), ze které vystupuje jako CLKIN přímo do procesoru. Na druhou stranu, pokud je zapotřebí upravit frekvenci tohoto signálu, tak se přivádí na blok PLL, kde se jeho frekvence zvýší či sníží. Násobitel se pak nastavuje pomocí čtyřbitového registru PLLCR[DIV]. PLL může pracovat i v režimu bypass, což znamená, že propouští vstupní signál nepozměněn na výstup. [2]

3.2 GPIO rozhraní

TMS poskytuje celkem 88 GPIO pinů. Na každý GPIO pin lze připojit jeden ze dvou, tří nebo čtyř zdrojů signálu (funkcí). Jako příklad jsou uvedeny v tabulce č. 3.2.1 možnosti připojení zdrojů k pinu GPIO6:

Název	Popis	STAVY		
		I	O	Z
GPIO	GIPO6 Univerzální vstup/výstup	x	x	x
EPWM4A	Výstup kanálu A z EPWM4		x	
EPWMSYNCI	Vstup synchronizace ePWM	x		
EPWMSYNCO	Výstup synchronizace ePWM		x	

(**Tabulka č. 3.2.1** – Možnosti připojení zdrojů signálu k pinu GPIO6.)

Nastavení GPIO:

Nastavení v jazyku C, je uvedeno v příloze č.2 na straně č.2.

První krok: Tímto krokem je právě výběr funkce GPIO, kdy se jednotlivý port nastaví pro určitou možnou funkci. Toto nastavení se provádí přepínáním multiplexoru. Pro přepínání jsou určeny multiplexory s názvy GPxMUXy (univerzální název). Každý z těchto multiplexorů přepíná funkce určité skupině GPIO pinů, kdy lze každý pin nastavit v rámci jednoho multiplexoru jinak. Rozdělení multiplexorů podle přiřazení k číslu ovládaného GPIO pinu jsou uvedeny v tabulce č. 3.2.2, jejíž rozdělení je využito i nadále v textu. [2]

GPIO0-15	GPIO16-31	GPIO32-47	GPIO48-63	GPIO64-79	GPIO80-87	ovládané GPIO
GPAMUX1	GPAMUX2	GPBMUX1	GPBMUX2	GPCMUX1	GPCMUX2	výběr funkce
GPAQSEL1	GPAQSEL2	GPBQSEL1	GPBQSEL2	GPCQSEL1	GPCQSEL2	výběr synchronizace
GPADIR		GPBDIR		GPCDIR		směr toku dat
GPAPUD		GPBPUD		GPCPUD		pull-up odpor
GPASET		GPBSET		GPCSET		nastavení 1
GPACLEAR		GPBCLEAR		GPCCLEAR		nastavení 0
GPATOOGGLE		GPBTOOGGLE		GPCTOOGGLE		přepnutí stavu

(**Tabulka č. 3.2.2** – Rozdělení multiplexorů.)

Druhý krok: Ke každému pinu je připojen interní pullup rezistor, který je možno softwarově odpojit (nastavení GPxPUD). Ve výchozím stavu jsou tyto rezistory pro GPIO0 – 11 zakázány a pro zbytek jsou povoleny. [2]

Třetí krok: Je nastavení pinu pro probuzení procesoru z režimu spánku. Pin lze nastavit v bloku Low-Power pro probuzení procesoru. (pin jako vstup) [2]

Čtvrtý krok: Při nastavení pinu jako vstup se nastavuje způsob synchronizace. Ten se nastavuje v qualifier registrech GPxQUALy. Nastavit synchronizaci je možné s těmito variantami.

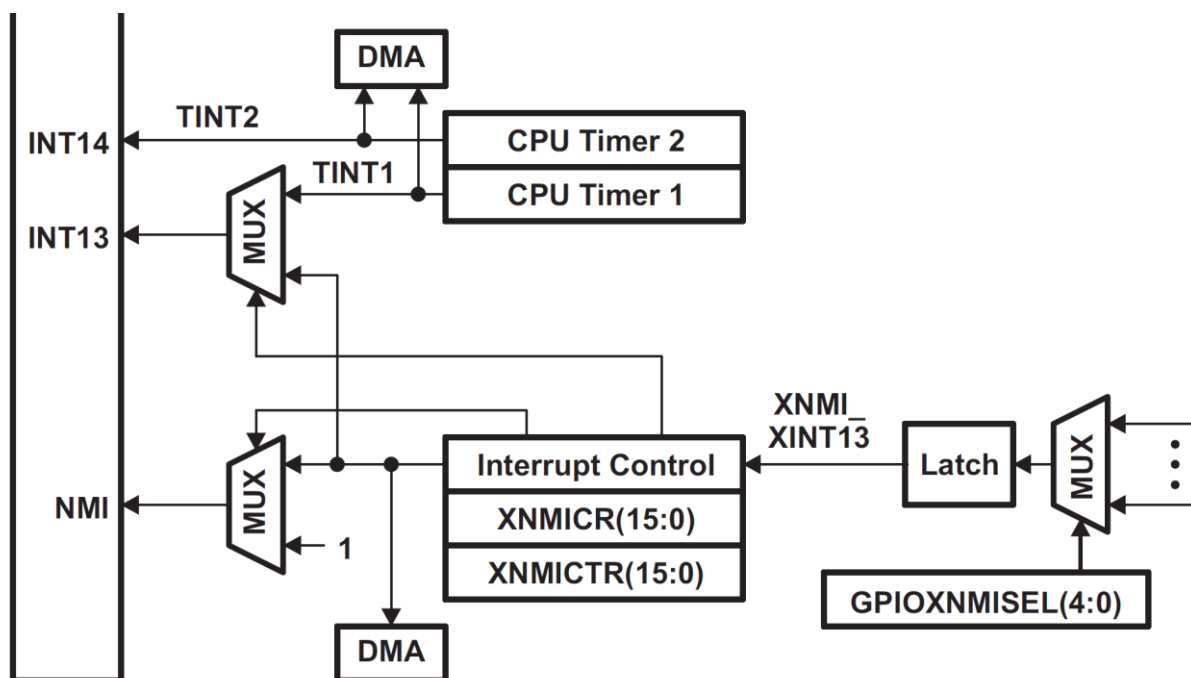
- 0 – synchronizace pinu se systémovými hodinami SYSCLK
- 1 – synchronizace podle prvních 3 vzorků. Platí pro piny konfigurovány jako GPIO nebo jako periferní funkce. Čas mezi vzorky je uveden v registrech GPxCTRL.
- 2 – synchronizace podle prvních 6-ti vzorků
- 3 – Asynchronní vstup. [2]

Pátý krok: Je - li pin nastaven na GPIO režim, je nutno nastavit směr toku dat (registry GPADIR, GPBDIR, GPCDIR vstup 0 /výstup 1). Základně jsou nastaveny jako vstup. Pokud je GPIO nastaven jako výstup, je možné provést celkem tři možná nastavení:

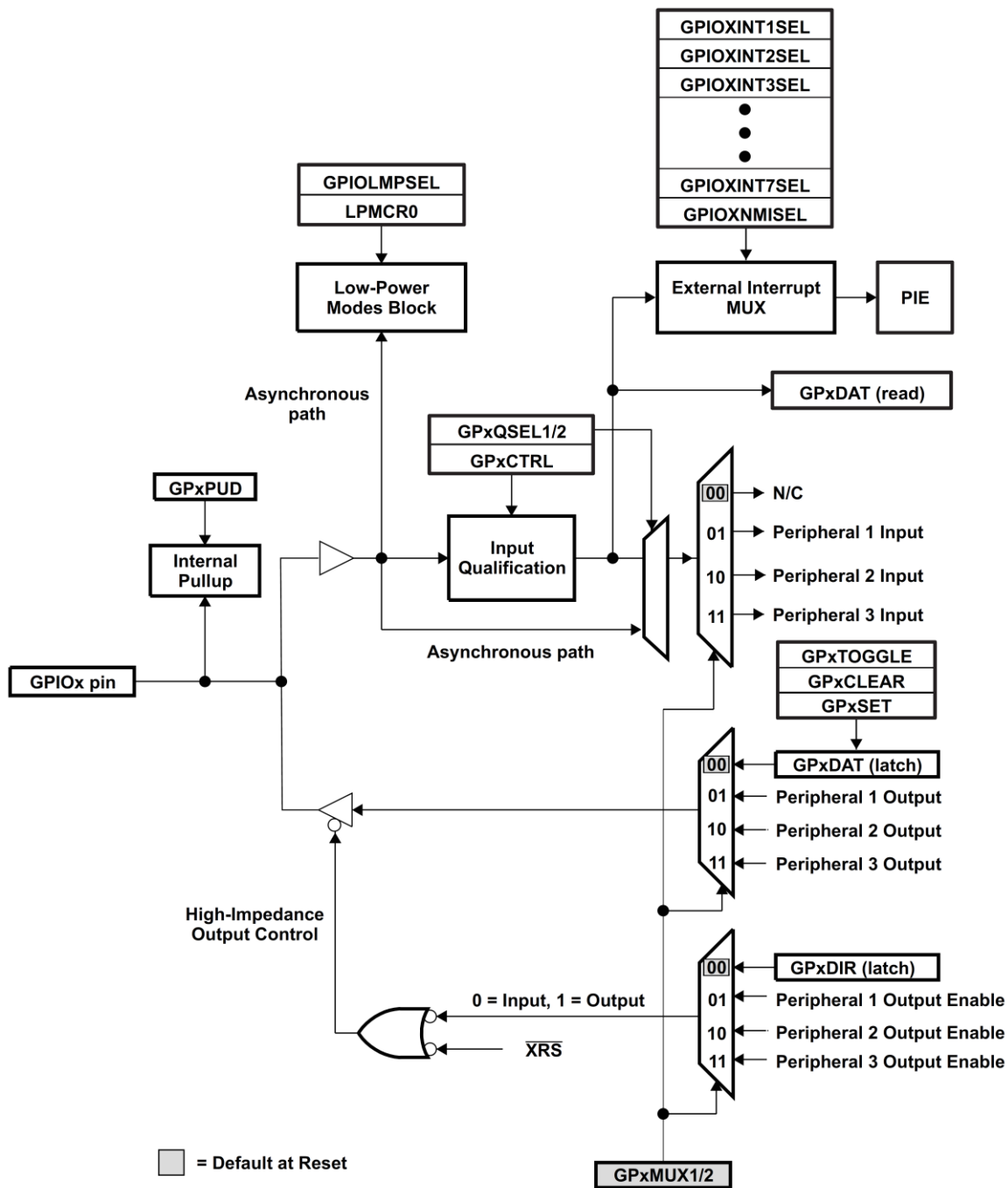
- GPxCLEAR – nastavení do nuly
- GPxSET – nastavení do jedničky
- GPxTOGGLE – přepnutí ze stavu na stav (z nuly do jedničky a naopak). [2]

Šestý krok: V případě, že je pro aplikaci potřeba nastavit vnější přerušení z univerzálního vstupu GPIO, je nutné přidělit tomuto pinu hodnotu XINT (XINT1 – XINT7, XNMI nebo XINT13). Hodnoty XINT jsou ve struktuře TMS ve své podstatě příznaky přerušení. To se provede zápisem daného GPIO pinu do registrů GPIOXINTxSEL (pro GPIO - A XINT1/2/3, pro GPIO-B-kanál XINT4/5/6/7). Tyto hodnoty jsou přivedeny do PIE přerušovacího bloku. Poslední variantou je přiřadit vstupním přerušením hodnoty XNMI nebo XINT13, což je nemaskovatelné (NMI) nebo maskovatelné (XINT13) přerušení, vedené do procesoru mimo blok PIE. Každému takovému přerušení se zadává polarita pro jeho vyvolání, která se nastavuje v registru XINTnCR. Kdy je přerušení generováno:

- 0. Sestupnou hranou signálu při přechodu z 1→0.
- 1. Náběžnou hranou signálu při přechodu 0→1.
- 2. Sestupnou hranou signálu při přechodu z 1→0.
- 3. Přerušení je generováno u náběžné i sestupné hrany v každém přechodu. [2]



(Obrázek č. 3.2.1 – Připojení přerušení XINT13 a NMI.)



(Obrázek č. 3.2.2 – Vnitřní schéma bloku ovládní funkce GPIO.)

3.3 Systém přerušení

Příznaky přerušení vstupující do procesoru se nazývají INTx (INT1 až INT14) a následně pak nemaskovatelný příznak NMI. Avšak v těchto příznacích je multiplexováno více podpříznaků. V dalším textu je uveden princip, jak je realizováno připojení jednotlivých příznaků k procesoru.

Nejprve je nutno rozdělit samotné vstupy INT:

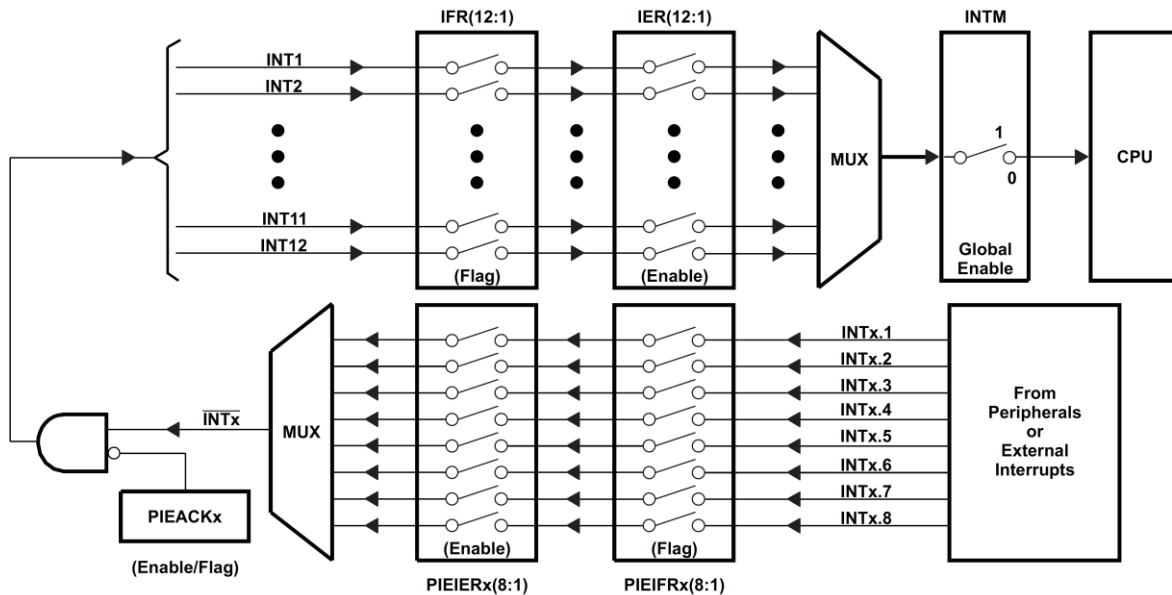
1. INT1 – INT12 zdroje přerušení jsou napojeny na PIE blok

2. INT 14 Tento vstup je vyhrazen čistě pro CPU timer 2
3. INT 13 vyhrazen pro vstup CPU timer 1 a vstupy GPIO0 – 31(viz kapitola 3.2)
4. Nemaskovatelné přerušení NMI (viz kapitola 3.2)

3.3.1 Nastavení PIE bloku

Účelem tohoto bloku je vybrat z 96-ti zdrojů přerušení právě jeden. Vstupují do něj jednak příznaky přerušení z vnitřních periférií, ale také přerušení vnějších zdrojů (za pomoci pinů GPIO). Všechny možné zdroje přerušení v PIE bloku jsou uvedeny v tabulce 3.3.1.1

Všech 96 přerušení se dělí do dvanácti skupin po osmi. (INT1 až INT12). Rozdělení do skupin je patrné z tabulky 3.3.1.1. Pro jednodušší pochopení principu lze proces výběru jednotlivého zdroje přerušení rozdělit do více fází, kdy jeho princip je zobrazen na obrázku č. 3.3.1.1



(Obrázek č. 3.3.1.1 – Princip funkce PIE bloku.)

Výběr INTx 1 z 8mi:

Pokud vyvolá nějaká periferie událost přerušení, nastaví se jednotlivý bit FLAG (příznak) v bloku PIEIFRx do jedničky (na obrázku sepnutý spínač). Tento příznak je vyhodnocován, zdali je povolen. Porovnání se provádí v bloku PIEIERx. Pokud je povolen, posílá se příznak dále do multiplexoru, jehož nastavením se vybírá jedno přerušení z 8mi periférií. Pokud je však daný příznak zakázán, zůstává i tak v hodnotě 1, dokud se nevymaže softwarově. Dále může nastat situace, kdy se při přetrvávajícím příznaku přerušení, může změnit povolení v bloku PIEIER. V tomto případě je daný příznak přiveden na vstup multiplexoru. Signálem PIEACKx se povoluje příznak od celé skupiny osmi periférií. [2]

Výběr INT 1 z 12ti:

V této fázi funguje výběr obdobně jako u výběru 1 z 8mi. Příznakem přerušení je zde však IFR a ověření povolení se provádí v bloku IER

INTM:

V tomto bloku se globálně povoluje přerušení od PIE jednotky.

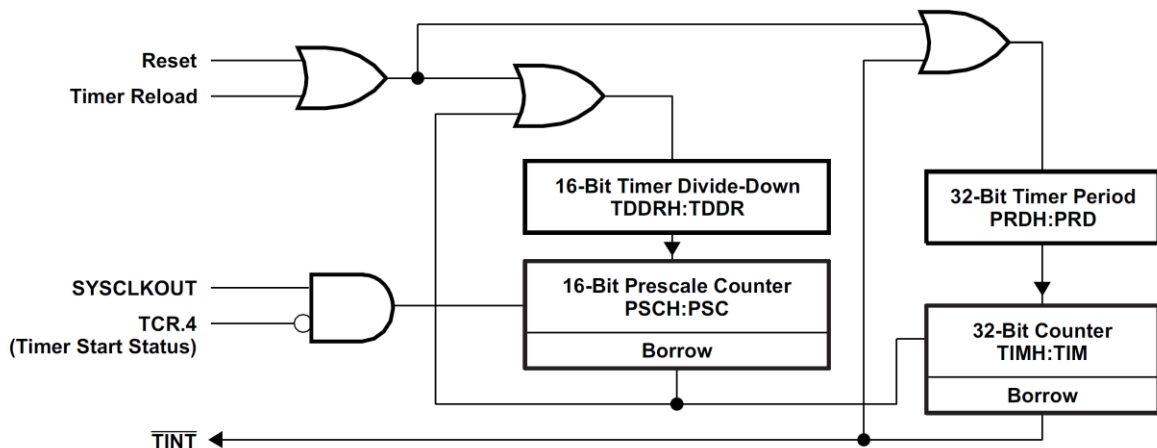
	INTx.8	INTx.7	INTx.6	INTx.5
INT1.y	WAKEINT	TINT0	ADCINT	XINT2
INT2.y	Rezervováno	Rezervováno	EPWM6_TZINT	EPWM5_TZINT
INT3.y	Rezervováno	Rezervováno	EPWM6_INT	EPWM5_INT
INT4.y	Rezervováno	Rezervováno	ECAP6_INT	ECAP5_INT
INT5.y	Rezervováno	Rezervováno	Rezervováno	Rezervováno
INT6.y	Rezervováno	Rezervováno	MXINTA	MRINTA
INT7.y	Rezervováno	Rezervováno	DINTCH6	DINTCH5
INT8.y	Rezervováno	Rezervováno	SCITXINTC	SCIRXINTC
INT9.y	ECAN1INTB	ECAN0INTB	ECAN1INTA	ECAN0INTA
INT10.y	Rezervováno	Rezervováno	Rezervováno	Rezervováno
INT11.y	Rezervováno	Rezervováno	Rezervováno	Rezervováno
INT12.y	LUF (FPU)	LVF (FPU)	Rezervováno	XINT7
	INTx.4	INTx.3	INTx.2	INTx.1
INT1.y	XINT1	Rezervováno	SEQ2INT	SEQ1INT
INT2.y	EPWM4_TZINT	EPWM3_TZINT	EPWM2_TZINT	EPWM1_TZINT
INT3.y	EPWM4_INT	EPWM3_INT	EPWM2_INT	EPWM1_INT
INT4.y	ECAP4_INT	ECAP3_INT	ECAP2_INT	ECAP1_INT
INT5.y	Rezervováno	Rezervováno	EQEP2_INT	EQEP1_INT
INT6.y	MXINTB	MRINTB	SPITXINTA	SPIRXINTA
INT7.y	DINTCH4	DINTCH3	DINTCH2	DINTCH1
INT8.y	Rezervováno	Rezervováno	I2CINT2A	I2CINT1A
INT9.y	SCITXINTB	SCITRINTB	SCITXINTA	SCIRXINTA
INT10.y	Rezervováno	Rezervováno	Rezervováno	Rezervováno
INT11.y	Rezervováno	Rezervováno	Rezervováno	Rezervováno
INT12.y	XINT6	XINT5	XINT4	XINT3

(Tabulka č. 3.3.1.1 – Rozdělení přerušení do jednotlivých skupin.)

Nastavení PIE jednotky v jazyce C, je uvedeno v příloze č.2 na 3.straně.

3.4 CPU časovače – CPU Timers

TMS320F28335, má k dispozici celkem 3 CPU časovače (CPU-Timer 0, CPU-Timer 1, CPU-Timer 2). Časovač 2 je vyhrazen pro DSP/BIOS, avšak zbylé dva časovače je možno využít v uživatelských aplikacích. Pokud se ovšem BIOS nevyužívá, je možno využít i CPU-Timer2 k uživatelským aplikacím. Tyto časovače jsou určeny k obecnému použití. Všechny periferie, které časovače využívají ke své činnosti mají svůj vlastní. [1]



(Obrázek č. 3.4.1– Princip funkce CPU časovače.)

Každý CPU - Timer se z hlediska funkce může rozdělit na dvě hlavní části: předdělicí část (prescale) a část generující přerušení.

Předdělicí část:

Na vstup SYSCLKOUT vstupuje hodinový signál, který uživatel může přerušit, pokud TCR.4 (Timer Start Status) nastaví do log.1. V případě, že tak neučiní, přivádí se hodinový signál do 16-ti bitového čítače Prescale Counter, který při každém taktu sníží jeho hodnotu o 1 („čítač dolů“). Jakmile dojde na hodnotu 0, vygeneruje se na jeho výstupu impuls.

V Registru Timer Divide Down je uložena návratová hodnota čítače Prescale Counter, která se do něj nahraje, pokud je právě vygenerován u Prescale counteru impuls na výstupu. Hodnoty registrů časovače jsou rozděleny na dvě části a to část spodních a horních bitů TDDR:H:TDDR, PSC:PSC (H- horní, high).

Část generující přerušení:

Tato část je na rozdíl od předdělicí 32-ti bitová ale funguje obdobně. Zde je hodinový signál prezentován jako impuls z výstupu čítače prescale Counter (výstup z předdělicí části), kterým se snižuje hodnota čítače TIMH:TIM o 1. Jakmile tento čítač dosáhne 0, generuje na výstupu signál pro přerušení TINT. Následně se načte návratová hodnota z registru PRDH:PRD.

Celý časovač se může vynulovat kladným impulsem na vstupy Reset a Timer Reload.

Nastavení CPU Časovačů v jazyce C, je uvedeno v příloze č.2 na 3. a 4. straně. [1]

3.5 PWM moduly

TMS320F28335 obsahuje celkem 6 rozšířených (enhanced) dvoukanálových PWM (A,B) modulů, tzv. „ePWM,“ s širokými možnostmi jejich nastavení. Jednotlivé bloky jednoho PWM modulu jsou:

Time Base (TB) - blok generující časování PWM.

Counter Compare (CC) – porovnává žádaný signál s výstupním signálem časovače.

Action Qualifier (AQ) – blok určující výstupní veličiny PWM.

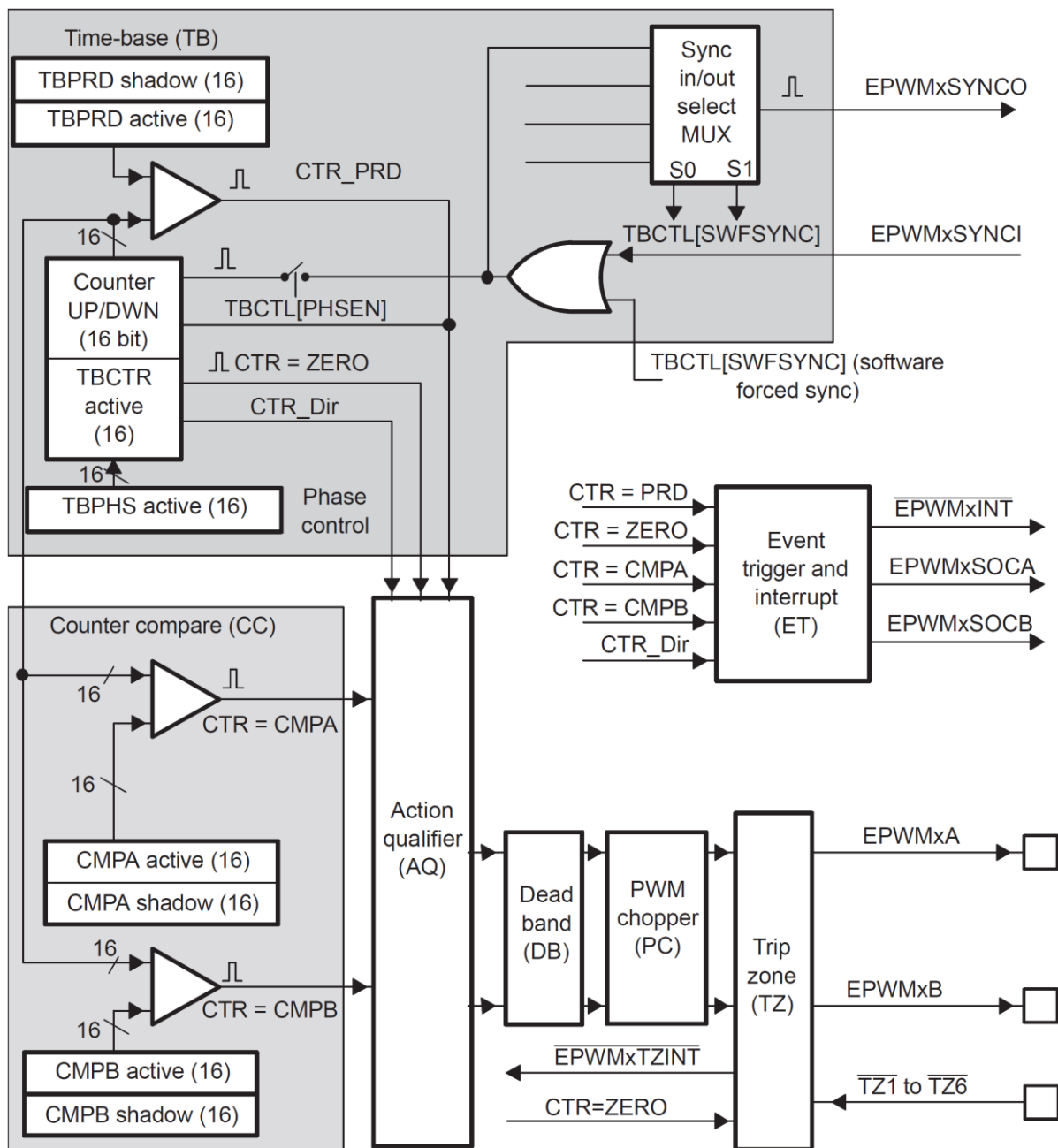
Dead Band generator (DB) – generátor mrtvých dob.

PWM Chopper (PC) - vytváří Vf nosný signál, pro přenos průběhů PWM přes galvanické oddělení

Trip Zone (TZ) – blok vyhodnocení chybových signálů.

Event Trigger (ET) – Modul komunikace s vnějším okolím.

Na obrázku č. 3.5.1 Je vyobrazeno vnitřní blokové uspořádání, které je popsáno v následujících kapitolách. Celkové Nastavení PWM jednotky v jazyce C, je uvedeno v příloze č.2 na 4. a 5.straně.



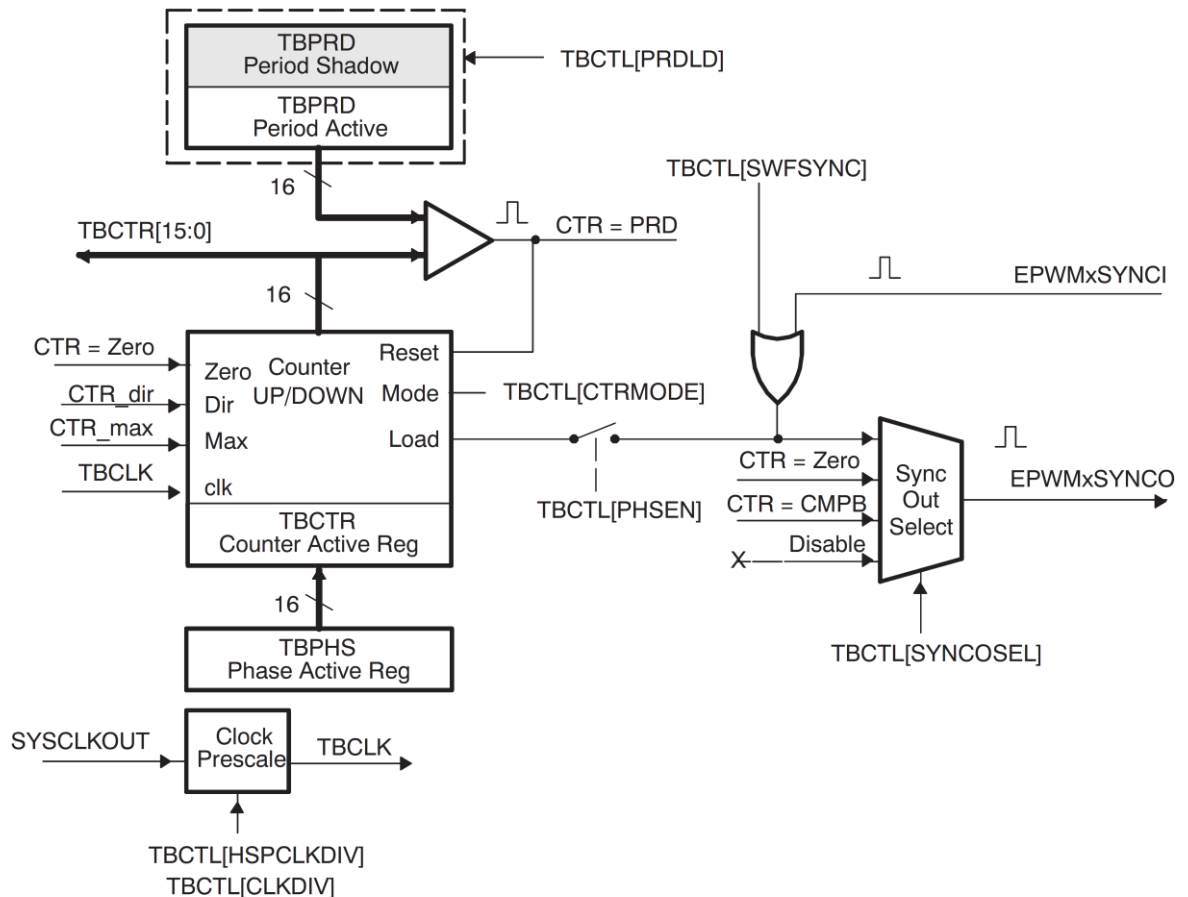
(Obrázek č. 3.5.1– Vnitřní blokové schéma jednoho ePWM modulu)

3.5.1 Time Base (TB)

V tomto bloku se generuje signál pilovitého tvaru, který je v dalších blocích využit pro komparaci s žádanými veličinami.

V registru TBPRD je uložena cílová hodnota čítání čítače CTR. Registr TBPRD má aktivní a stínovou část, kdy se nová hodnota načte nejprve do stínového registru a při vynulování čítače se přenesou do registru aktivního, čímž je zajištěno vyšší zabezpečení vůči kolizi. Povolení stínové části se provede nastavením $TBCTL[PRDL] = 0$. Registr je však možno nastavit i tak, aby stínovou část

nepoužíval. Rychlost čítání určuje frekvence signálu TBCLK, jehož frekvence je odvozena od systémových hodin (nastavení HSPCLKDIV a CLKDIV). Čítač CTR může být nastaven na tři možné varianty čítání. A to čítání dolů, nahoru anebo čítání v obou směrech, kdy se směr čítání určí nastavením TBCTL[CTRMODE]. [3]



(Obrázek č. 3.5.1.1– Funkční schéma bloku Time Base)

Čítání nahoru.

Výstupní hodnota čítače se porovnává s hodnotou uloženou v registru TBPRD. Ve chvíli, kdy se tyto hodnoty rovnají je vygenerován na výstupu komparátoru signál CTR=PRD, který je přiveden na reset čítače.

Čítání dolů

V registru TBPRD je uložena počáteční hodnota čítače, která je čítáním dekrementována k nule, kde dochází k resetu čítače a opětovné načtení hodnoty z registru TBPRD.

Čítání v obou směrech

Čítaná hodnota začíná na hodnotě 0 a inkrementuje se až do hodnoty TBPRD. Při dosažení této hodnoty začne čítač dekrementovat k nule.

Aktuální načtená hodnota čítače je uložena v registru TBCTR. Při načtení nuly čítač generuje signál CTR=Zero a pokud je načten v maximální hodnotě CTR_max. Znak směru čítání (inkrementace – dekrementace) je vyveden pomocí signálu CTR_dir. [3]

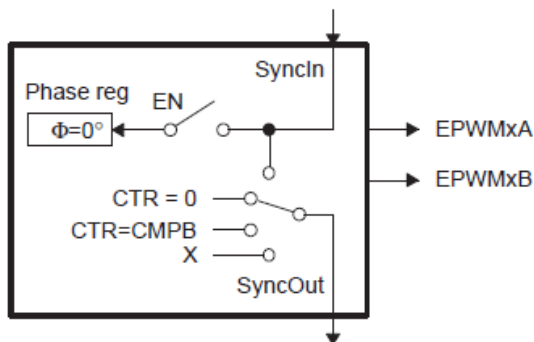
3.5.2 Synchronizace mezi moduly

Synchronizace modulů probíhá pomocí signálů SYNCI/SYNCO. V každém modulu může být synchronizace zakázána. Průběh synchronizace: pokud se sepne „spínač“ TBCTL[PHSEN] je povolena synchronizace. V momentě synchronizačního signálu se načte do registru čítače (TBCTL) hodnota z registru fáze (TBPHS), přičemž čítač pokračuje v čítání stejným směrem viz obrázek č.

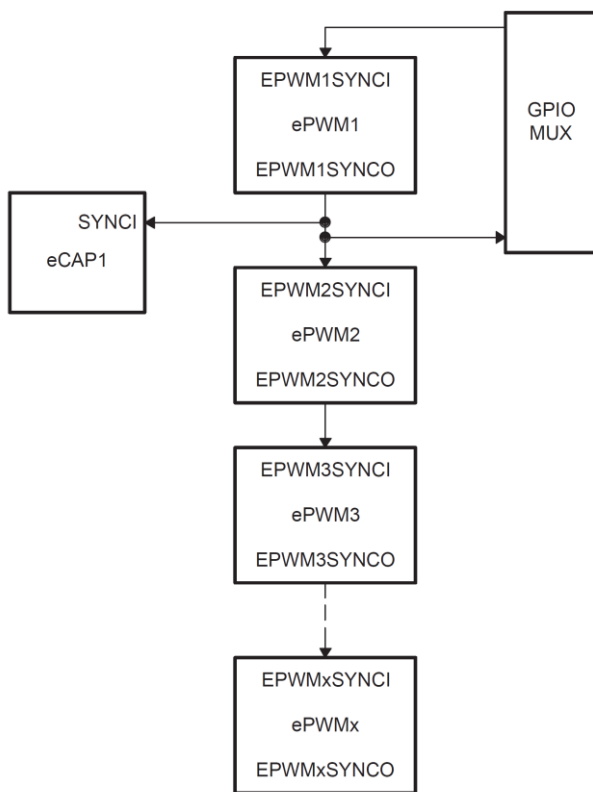
3.5.1.1. Impuls pro synchronizaci je přiváděn vždy z bloku Time-Base předchozího PWM modulu (První modul má vstup nastaven z vnějšího zdroje, viz obrázek č. 3.5.2.2). [3]

V každém modulu (kromě PWM1) je možno zvolit ze čtyř zdrojů (viz obrázek č. 3.5.2.1):

- Synchronizační signál připojen k předešlému modulu
- Vynulování čítače (CTR=Zero) předešlého modulu
- Z druhého kanálu modulu
- Zakázat – Disable



(Obrázek č. 3.5.2.1– Možnosti přepnutí zdroje synchronizace.)



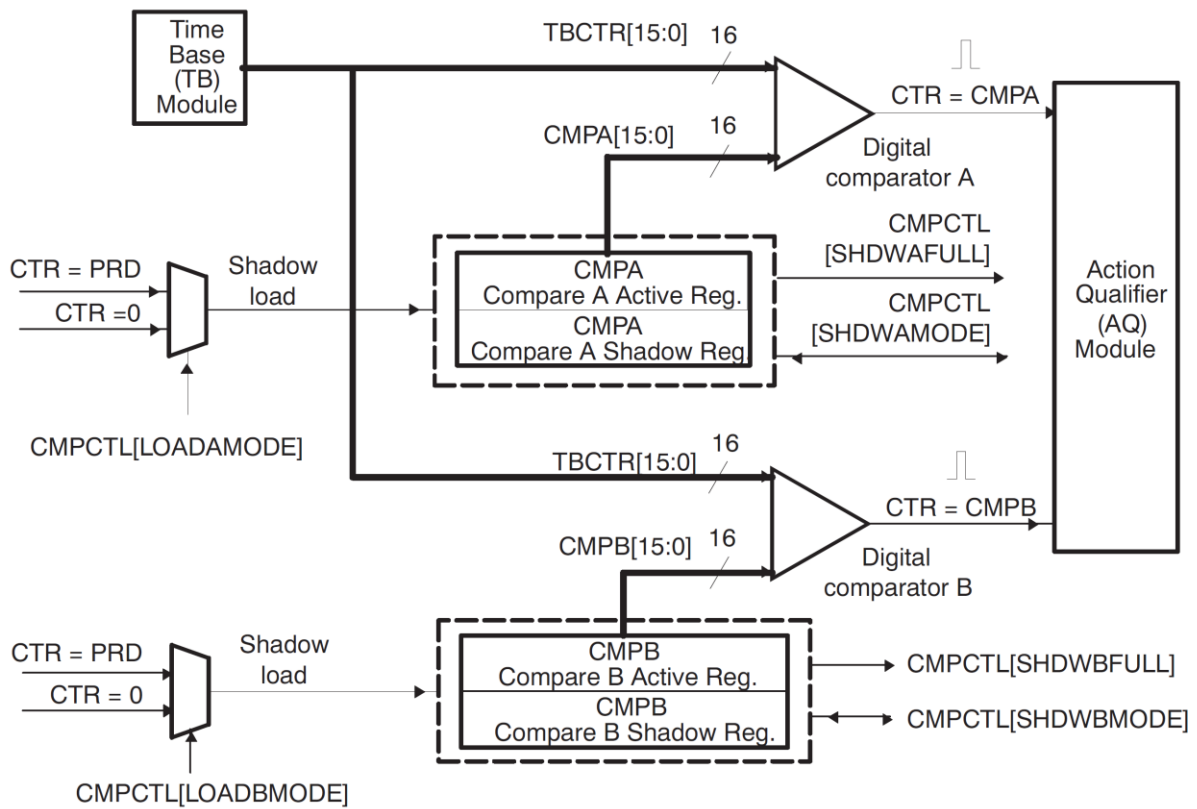
(Obrázek č. 3.5.2.2– Uspořádání návaznosti synchronizace.)

3.5.3 Counter Compare (CC)

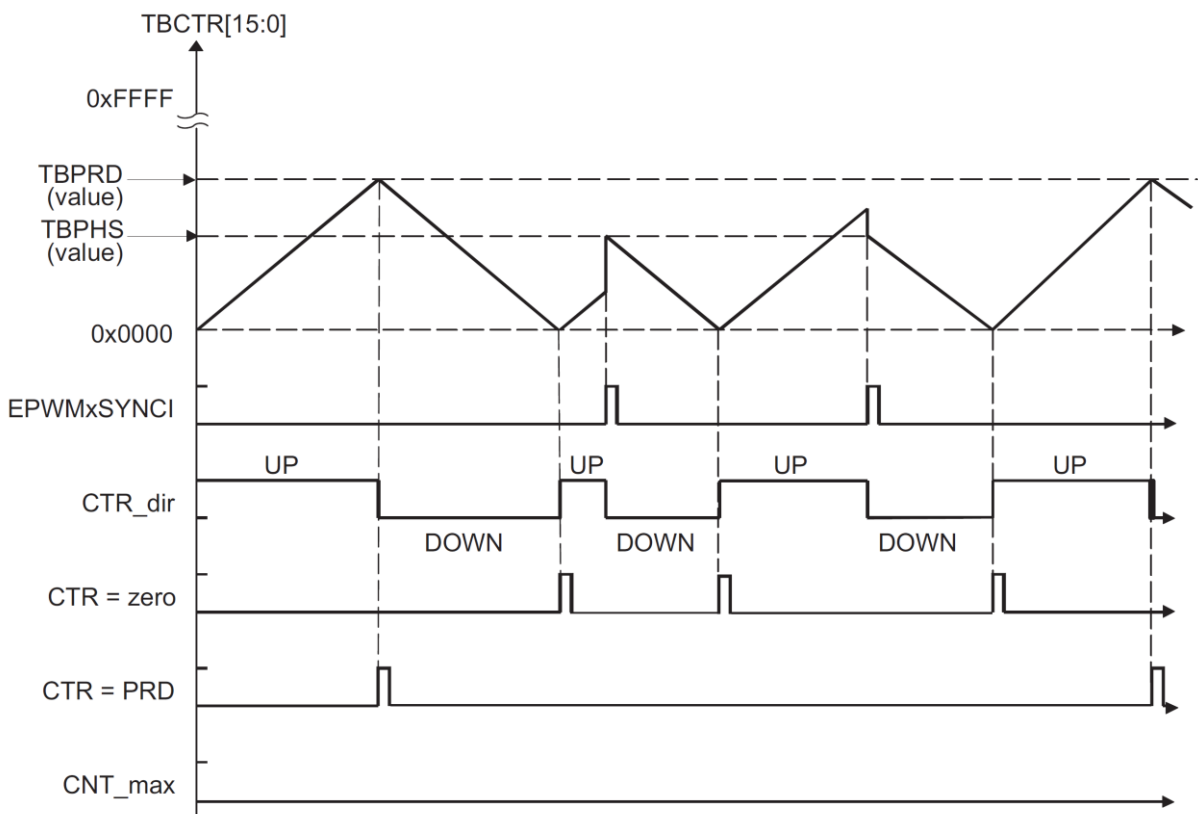
TBCTR je vstupním signálem z Time Base modulu. Tento signál se kontinuálně porovnává s hodnotou uloženou v registru CMPA a CMPB (obrázek č. 3.5.3.2). Dochází zde k rozvětvení modulu na dva kanály. Pokud je vstupní hodnota TBCTR rovna hodnotě uložené v registru CMPA (CMPB), generuje se na výstupu komparátoru signál CTR=CMPA (CTR=CMPB). Stínový registr zde funguje obdobně jako u předchozího případu, kdy se hodnota ze stínového registru zapíše do registru aktivního. Nastavení času zápisu hodnoty ze stínového do aktivního registru se může zvolit ze dvou variant:

- z bloku Time-Base: CTR=PRD
- z bloku Time-Base: CTR=0

Dobu přenosu hodnoty ze stínového do aktivního registru se nastavuje pomocí CMPCTL[LOADAMODE]. Povolení, či zakázání stínování registru se nastavuje v registru CMPCTL[SHDWAMODE] (0 enable, 1 disable). Signál z CMPCTL[SHDWAFULL] značí, zdali ve stínovém registru čeká nějaká hodnota na předání, či nikoli. [3]



(Obrázek č. 3.5.3.1– Funkční schéma bloku Counter Compare.)



(Obrázek č. 3.5.3.2 – Příklad porovnávání hodnot při obousměrném čítání TB.)

3.5.4 Action Qualifier (AQ)

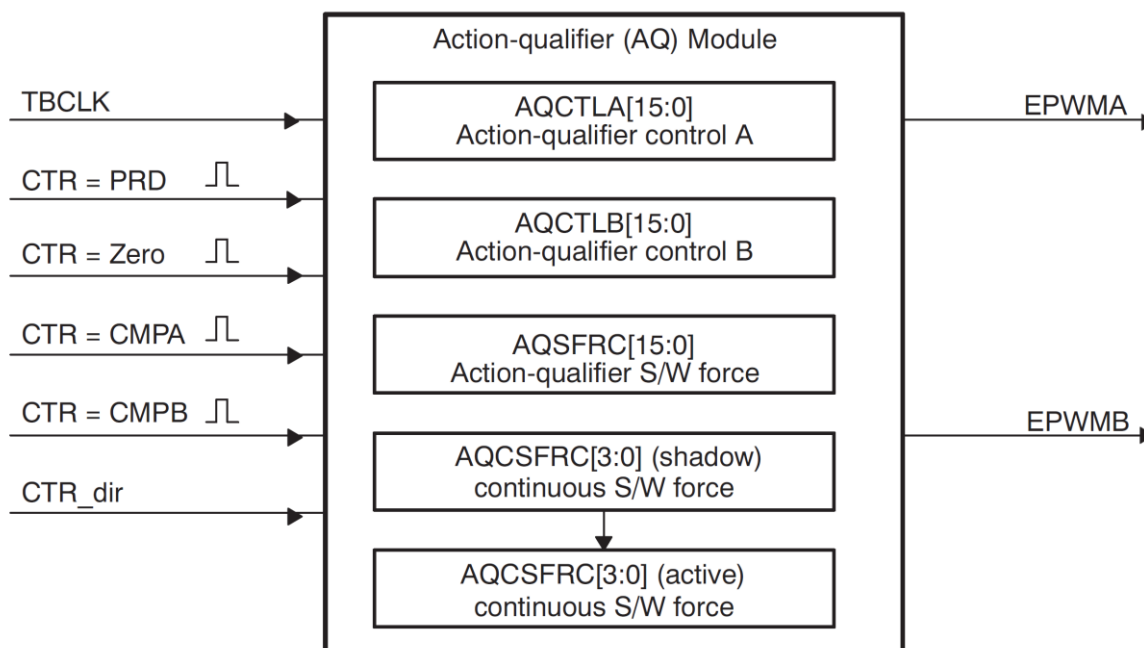
Tento blok má za úkol podle výstupních signálů z bloků TB a CC a směru hodin čítače CLK generovat na výstup signály EPWMA,EPWMB.

Dle uživatelského programu reaguje na vstupní signály a nastavuje výstupní signál čtyřmi způsoby:

- nastaví 1 (set)
- nastaví 0 (clear)
- přepne z jednoho do druhého stavu (toggle)
- výstup se nezmění

Při příchodu více signálů na vstup, vyhodnocuje Hardware, který signál přijme jako první. Toto vyhodnocení je rozdílné v závislosti na módu čítače CTR (nahoru, dolů, obousměrný režim). Způsoby reakce na vstupní signály jsou uloženy v registrech:

- AQCTLA – registr pro kanál A
- AQCTLB – registr pro kanál B
- AQSFRC – registr pro softwarovou akci[3]

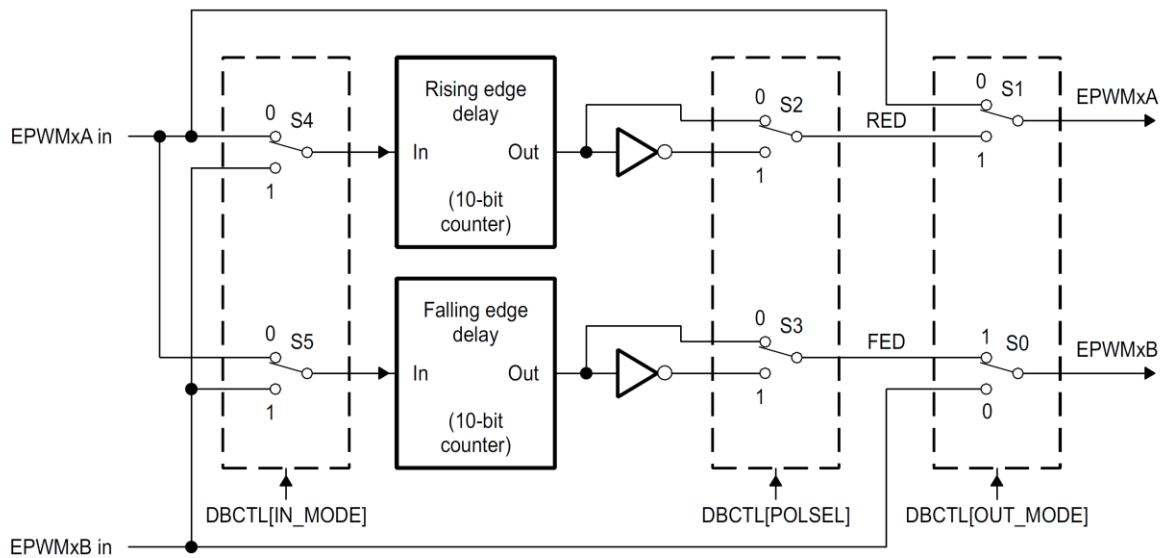


(Obrázek č. 3.5.4.1 – Blokové schéma bloku Action qualifier.)

3.5.5 Dead Band generator (DB)

V tomto bloku jsou realizovatelné všechny vnitřní přepínací kombinace, kdy za pomoci nich se generují ve vstupním signálu mrtvé doby. Některé mrtvé doby se dají realizovat nastavením v dřívějších funkčních blocích, avšak např: při rozdělení střídavy sepnutí tranzistorů, kdy kanály A a B

pracují v opačném režimu, je blok generující mrtvé doby jediným možným blokem pro realizaci mrtvých dob. [3]

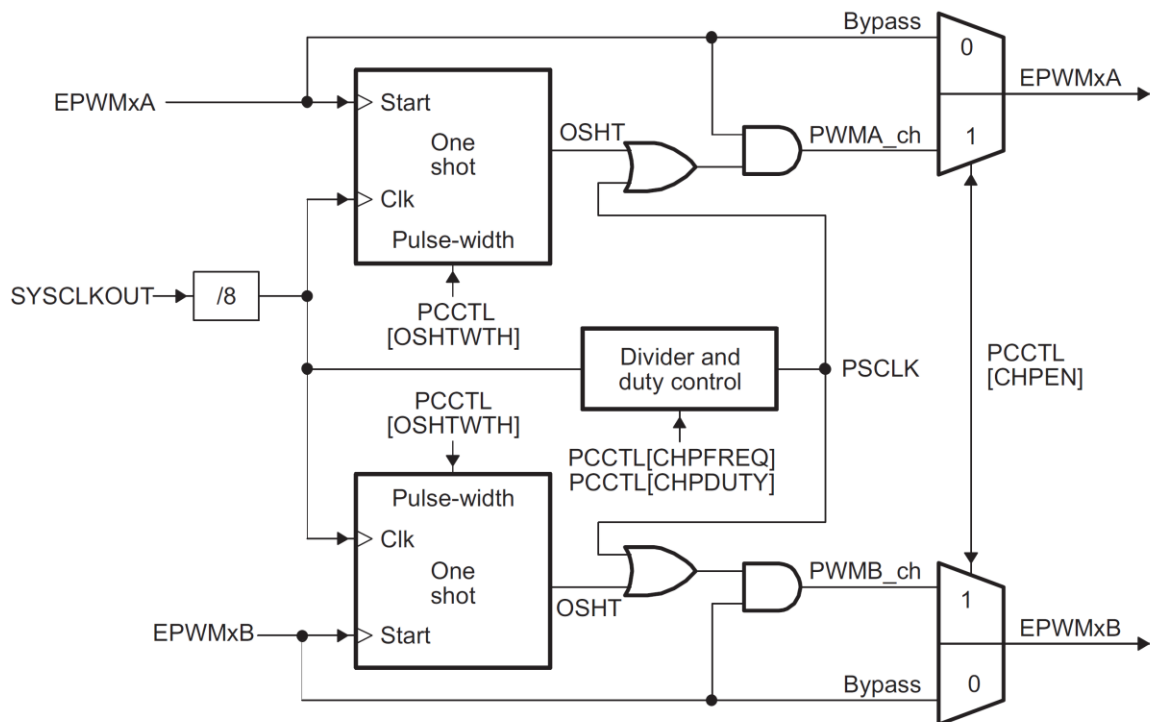


(Obrázek č. 3.5.5.1 – Funkční schéma bloku Dead Band generator.)

Tento modul podporuje nezávislé nastavení délky mrtvé doby pro náběžnou (RED) i sestupnou (FED) hranu signálu. Délka zpoždění se nastavuje do registrů DBRED a DBFED. Jedná se o 10-ti bitové registry, jejichž hodnota představuje násobek systémových hodin PWM modulu TBCLK. [3]

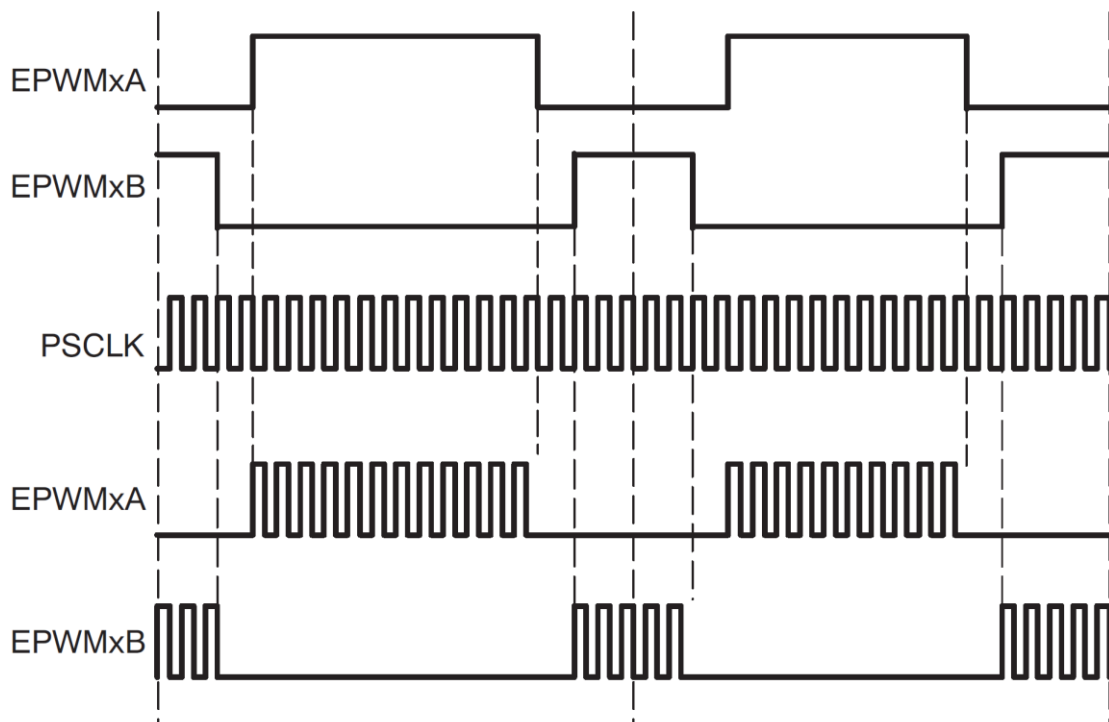
3.5.6 Chopper (PC)

V této části PWM modulu se výstupní pwm signál rozstřídá na vysokou frekvenci, čímž se prakticky vytvoří modulace výstupního PWM signálu. Tato funkce je důležitá při galvanickém oddělení, které je realizováno za pomoci pulsního transformátoru. [3]



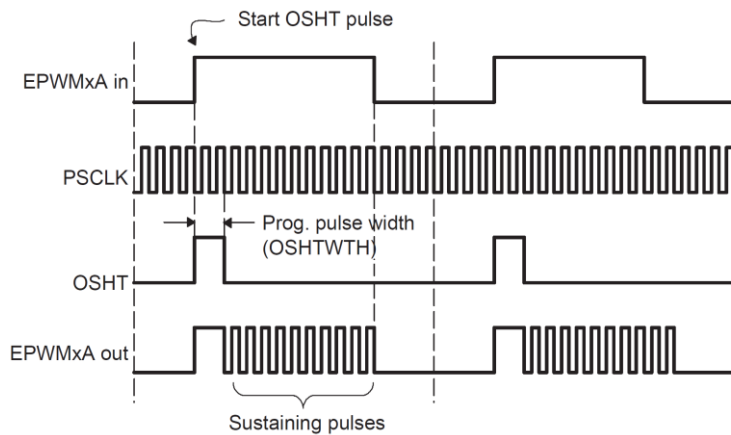
(Obrázek č. 3.5.6.1 – Funkční schéma bloku PWM střídače)

Časování obvodu je odvozeno od systémových hodin. Vstupní signál do tohoto bloku vystupuje z generátoru mrtvých dob. Vstupní signál je v obvodu rozstřídán, jak lze vidět na obrázku č. 3.5.6.2:



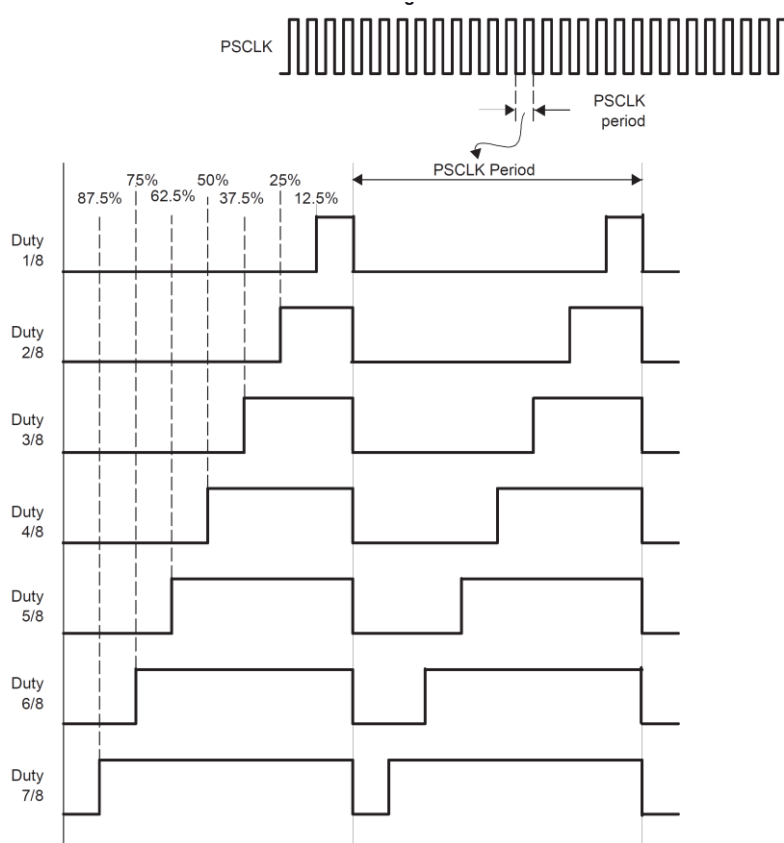
(Obrázek č. 3.5.6.2 – Rozstřídání PWM signálu)

Obvodem one shot se nastavuje šířka prvního rozstředaného signálu po změně hrany na vstupu. Toto nastavení se provádí nastavením PCCTL[OSHTWTH], kdy se do tohoto registru vkládá počet vzorkovacích hodin. PCCTL[OSHTWTH]=0x0 – 0xF. [3]



(Obrázek č. 3.5.6.3 – Nastavení šířky prvního pulsu)

Signály PCCTL[CHPFREQ] a PCCTL[DUTY] se nastavují střída a frekvence rozstředání. Střídu je možné volit vždy po násobku 12,5%. (12,5 – 87,5% dělitel 1 - 8).



(Obrázek č. 3.5.6.4 – Nastavení střída frekvence rozstředání.)

Na výstup tohoto bloku je možno připojit i vstupní signál nastavením PCCTL[CHPEN].

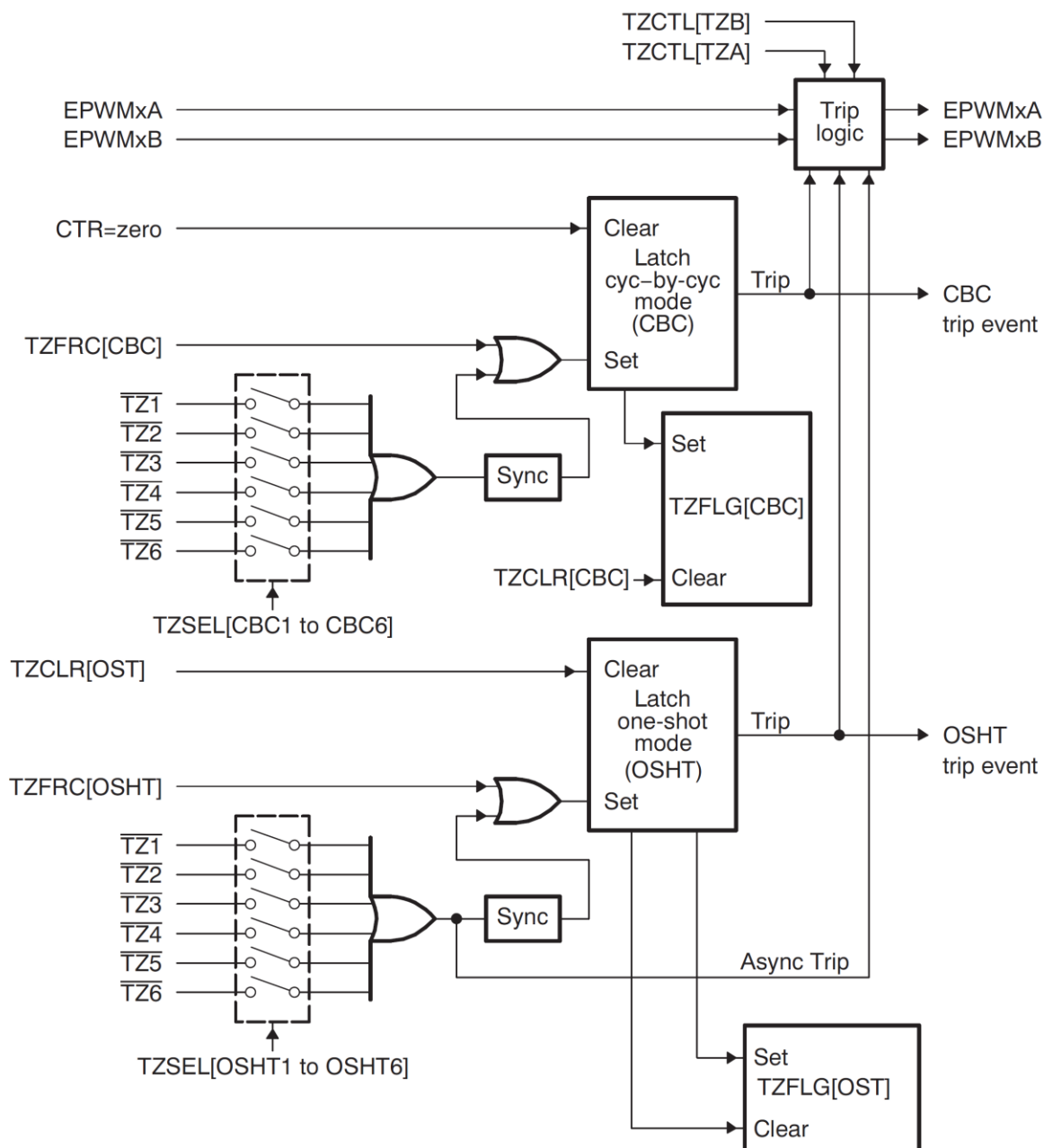
3.5.7 Trip Zone (TZ)

Signály TZ1 – TZ6 jsou generovány vnějším zdrojem (obrázek č. 3.5.7.1). Využívají se při vnějších poruchách, kdy je při chybě přivedena na vstup TZ log.0. Obvod TZ pak může být nastaven tak, aby na tyto signály nereagoval, či nastavil výstupní hodnoty EPWMA,B do stavu 0, 1 nebo stavu vysoké impedance, kdy reakce výstupu je definována v registru TZCTL (TZA, TZB). Povolení připojení signálu TZx se povoluje v registrech TZSEL. [3]

Nastavení kontroly přetrvávání chybového stavu a ovládní reakce na jeho odeznění je možno nastavit na dva módy a to Cycle-by-Cycle (CBC) nebo One-Shot (OSHT):

CBC – V případě signálu TZ se výstup EPWMX nastaví do požadovaného stavu a aktivuje se registr příznaku (TZFLG[CBC]), kterým se generuje přerušení TZEINT (pokud je povoleno v registru přerušení). Pokaždé, když je hodnota čítače TBCTR rovna nule se kontroluje, zda signál TZ odezněl. Pokud ano, výstupy se automaticky nastaví na provozní hodnotu. Příznak je však nutné vymazat ručně. [3]

OSHT - V případě signálu TZ se výstup EPWMX nastaví do požadovaného stavu a aktivuje se registr příznaku ((TZFLG[OST]), kterým se generuje přerušení TZEINT (pokud je povoleno v registru přerušení). Zpětné povolení výstupů se zde neprovádí automaticky jako u metody CBC ale je jej potřeba ručně povolit vymazáním bitem ZCLR[OST]. [3]



(Obrázek č. 3.5.7.1 – Funkční schéma submodulu Trip Zone.)

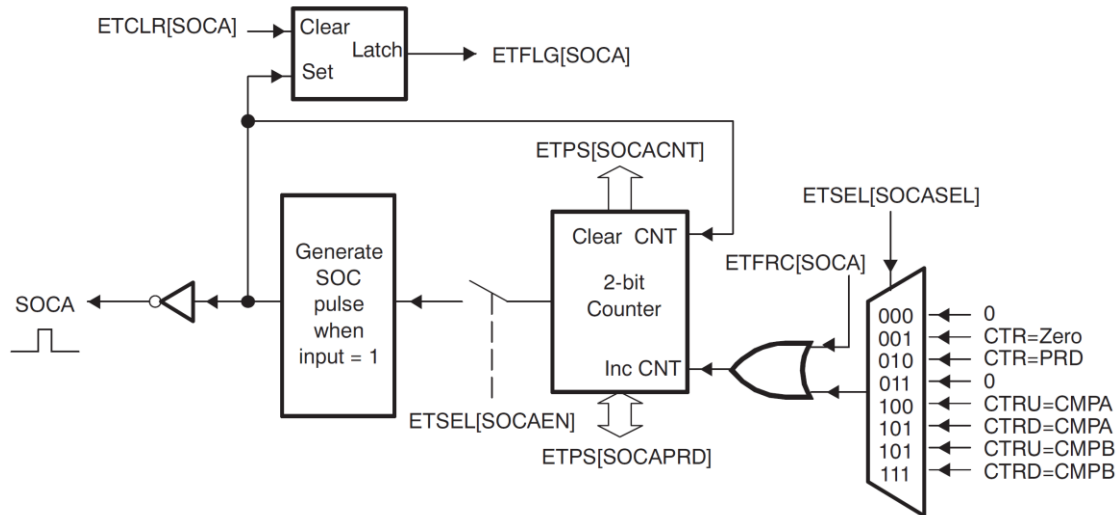
3.5.8 Event Trigger (ET)

Obvod ET generuje výstupní signál pro příznak přerušení v PIE bloku, či AD převodníku. Obvod čítá uživatelem zadaný vstupní signál, kdy při načtení požadovaného násobku vstupního signálu generuje signál výstupní. Požadovaný násobek je možno volit po celých číslech v rozmezí 1 až 3. Vstupní signál je možné vybrat z následujících signálů

- CTRU=CMPA - CTR=CMPA při čítání časovače nahoru
- CTRD=CMPA - CTR=CMPA při čítání časovače dolů

CTRU=CMPB - CTR=CMPB při čítání časovače nahoru
 CTRD=CMPB - CTR=CMPB při čítání časovače dolů
 CTR=ZERO - čítač dosáhl hodnoty nula (kapitola 3.5.1)
 CTR=PRD – reset čítače (kapitola 3.5.1)

Podrobnější popis funkce tohoto bloku lze získat z popisu obrázku č.3.5.8.1.

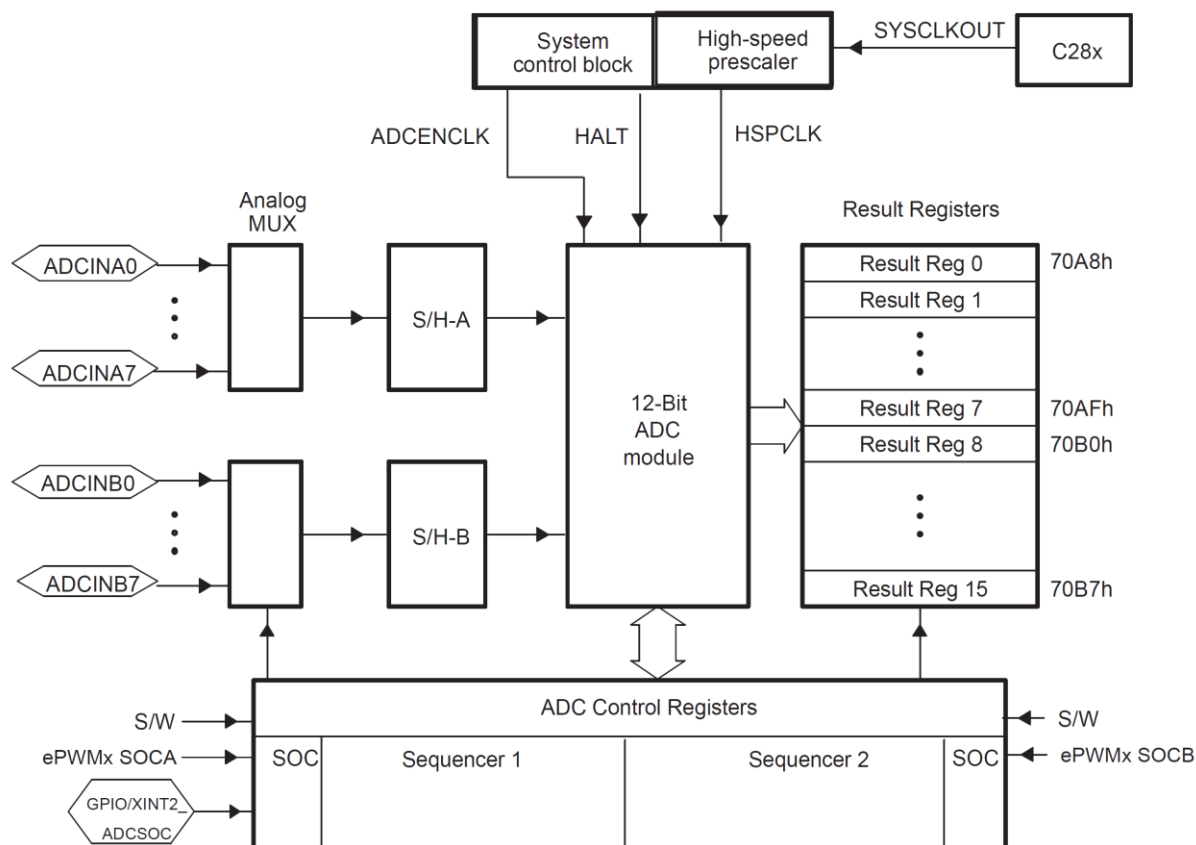


(Obrázek č. 3.5.8.1 – Funkční schéma submodule Event Trigger.)

V registru ETSEL[SOCASEL] se vybírá zdroj signálu. Pokud je zapotřebí využít softwarového zdroje signálu, provádí se tak signálem z ETFRM[SOCA]. Zdroj signálu je následně přiveden do dvou bitového čítače (2-bit Counter). V registru ETPS[SOCAAE] je uložena hodnota počtu příchozích pulsů, aby se na výstupu “sepnul spínač” ETSEL[SOCAEN], kterým se generuje signál SOCA, a zároveň je také generován příznak přerušení ETFLG[SOCA], který může být vymazán signálem z ETCLR[SOCA]. Díky zpětné vazbě se vymaže hodnota dvou bitového čítače a ten začíná čítat od nuly. [3]

3.6 A/D převodník – ADC modul

A/D převodník (ADC) má 16 kanálů, které jsou konfigurovány jako dva nezávislé osmi-kanálové bloky pro obsluhu ePWM modulů. Tyto dva bloky je možno seřadit kaskádově, čímž se vytvoří jeden šestnácti kanálový modul. Jak je vidět na obrázku č. 3.6.1, celý AD převodník má pouze jeden 12-ti bitový převodník.



(Obrázek č. 3.6.1 – Funkční schéma ADC modulu.)

Popis funkce tohoto modulu je následující: Taktování ADC modulu je odvozeno od systémových hodin, které se upravují v předdělicím bloku „Hi-speed prescaler“.

Start převodu je možno provést více signály a to jsou: S/W – softwarový start; ePWMxSOCA,B – výstupy z ePWM modulů; ADCSOC – externí signál připojený skrze GPIO porty.

Výběr vstupu pro převod se přepíná pomocí multiplexoru, kdy sekvence vyčítání jsou uloženy v blocích Sequencer1 a Sequencer2. Následně se pak převáděná hodnota zadrží v S/H obvodech (sample/hold – vzorek/držet - pracuje na principu analogové paměti). Převodní modul vyčítá z obvodů S/H hodnotu, kterou převede. Po dokončení převodu jsou získaná data uložena do registru pro výsledky. V kaskádovém režimu fungují oba sequencery jako jeden.

Převodník začíná převádět další úlohu, jakmile dokončí úlohu stávající. V nezávislém módu jsou primárně vyřizovány žádosti od Sequenceru1. V nekaskádovém zapojení je možno zvolit tzv. simultánní převod, což znamená, že se snímají hodnoty obou kanálů najednou. Toto umožňují obvody sample/hold (S/H), které fungují jako analogová paměť. Nejprve převodník vyčte převáděnou hodnotu z prvního S/H obvodu a následně zadrženou hodnotu druhého S/H obvodu. Rozsah vstupního napětí je 0 – 3V (12-bit; digitální hodnota 0 až 4095₍₁₀₎). [4]

Signály:

ADCENCLK – tímto signálem se povoluje časování A/D převodníku. (1-povoleno; 0 – zakázáno).

HALT – tímto signálem se zapíná/vypíná režim spánku, přičemž hodnoty v registrech zůstávají zachovány.

HSPCLK – je taktovací signál převodníku (je odvozen od systémových hodin).

Autosekvenční mód

Tento mód umožňuje při žádosti o start převodu, automaticky převádět více kanálů za sebou, či vícekrát převádět jeden kanál, a to se buď:

- Převádí stejný kanál vícekrát, čehož se dá využít v převzorkovacích algoritmech, kterými se dosahuje vyšších přesností, než u jednorázového převodu.
- Provádí se jednorázový převod více kanálů
- Kombinovaně, přičemž maximální počet převodů je 16 pro kaskádový režim a 8 pro normální režim.

Nastavení autosekvenčního módu jazyce C, je uvedeno v příloze č.2 na 5. a 6.straně. [4]

3.7 SCI rozhraní

SCI komunikace má více možností nastavení tvaru odesílaného packetu, proto se komunikující zařízení na jedné sběrnici musí nastavit na jeden přenosový formát. Všeobecný tvar packetu je následující:

- jeden startovací bit
- jeden až osm datových bitů
- paritní bit (možno i bez)
- adresní bit (módy s a bez tohoto bitu)
- jeden nebo dva stop bity

Na SCI sběrnici může být připojeno více než dvě zařízení např. multiprocesorová komunikace. Tento formát komunikace umožňuje přenos dat mezi více procesory po jedné sběrnici, kdy vysílat může v jeden čas pouze jedno zařízení, přičemž příjem přenosu může být prováděn libovolným počtem zařízení připojených ke sběrnici. [5]

Adresní bajt

Je první bajt vysílaný vysílačem, který je určen všem zařízením na sběrnici. Tímto bajtem se určují příjemci vysílaných dat. Tj. pouze zařízení nastavená na stejnou adresu přijímají vysílaná data, následující po daném adresním bajtu. Ostatní přijímače zůstávají v nečinnosti až do dalšího přijatého adresního bajtu. Adresní bajt je zapotřebí odlišit od ostatních, aby jej bylo v komunikaci jasně rozeznat. Toto se provádí dvěma způsoby:

- Režim adresního bitu (address-bit mode)

Zde se určení zdali se jedná o adresní bajt realizuje adresním bitem v každém přenášeném bajtu. Pokud je tento bit nastaven do 1, jedná se o adresní bajt. Pokud do 0, jedná se o datový bajt. Tento režim je vhodný pro komunikaci mezi více zařízeními.

- Režim nečinnosti (idle-line mode)

V tomto režimu se nevyužívá adresního bitu, jako v předešlém případě, který není vůbec obsažen v komunikaci. V tomto režimu se před adresním bajtem nějakou dobu nevysílají žádná data (tichý prostor). Tento režim je vhodný v případě, kdy se odesílají větší počty přenášených dat jednomu zařízení. [5]

Sleep bit

Nastavuje režim spánku SCI rozhraní. Jeho funkce je návazná na adresní bajt, kdy se při správném adresním bajtu nastaví sleep bit do nuly. Avšak nastavení sleep bitu musí být provedeno softwarově. Pak je teprve generováno přerušení RXDY,RXINT.

Popis bloku

Vysílání:

SCITXD je název pro vysílaný signál a výstupní pin. Data pro přenos jsou ukládána do vysílacího zásobníku „Transmitter –Data Buffer Registr“ a na výstup se propouštějí posuvným registrem TXSHF. Signál TXRDY značí, že je zásobník (buffer) připraven přijmout další data. TXEMPTY značí, že posuvný registr je prázdný. Rychlost propouštění posuvného registru určuje hodnota v Baud Rate registrech (Bd – BAUD jednotka modulační rychlosti). Blok TXWAKE slouží pro generování adresních bitů. Registry SCICCR se nastavuje formát přenášených dat, registry SCICTL1 a SCICTL2 se určuje styl komunikace. [5]

Příjem:

Je v podstatě obrácenou logikou vysílání, kdy se data přijímají přes posuvný registr do přijímacího zásobníku. Do registrů SCIRXST se ukládají typy chyb přijatého signálu. [5]

SCICCR7 - počet stop bitů (0 jeden, 1 dva)

SCICCR6 – typ parity (1 sudá, 0 lichá)

SCICCR5 – povolení paritního bitu (0 zakázáno, 1 povoleno)

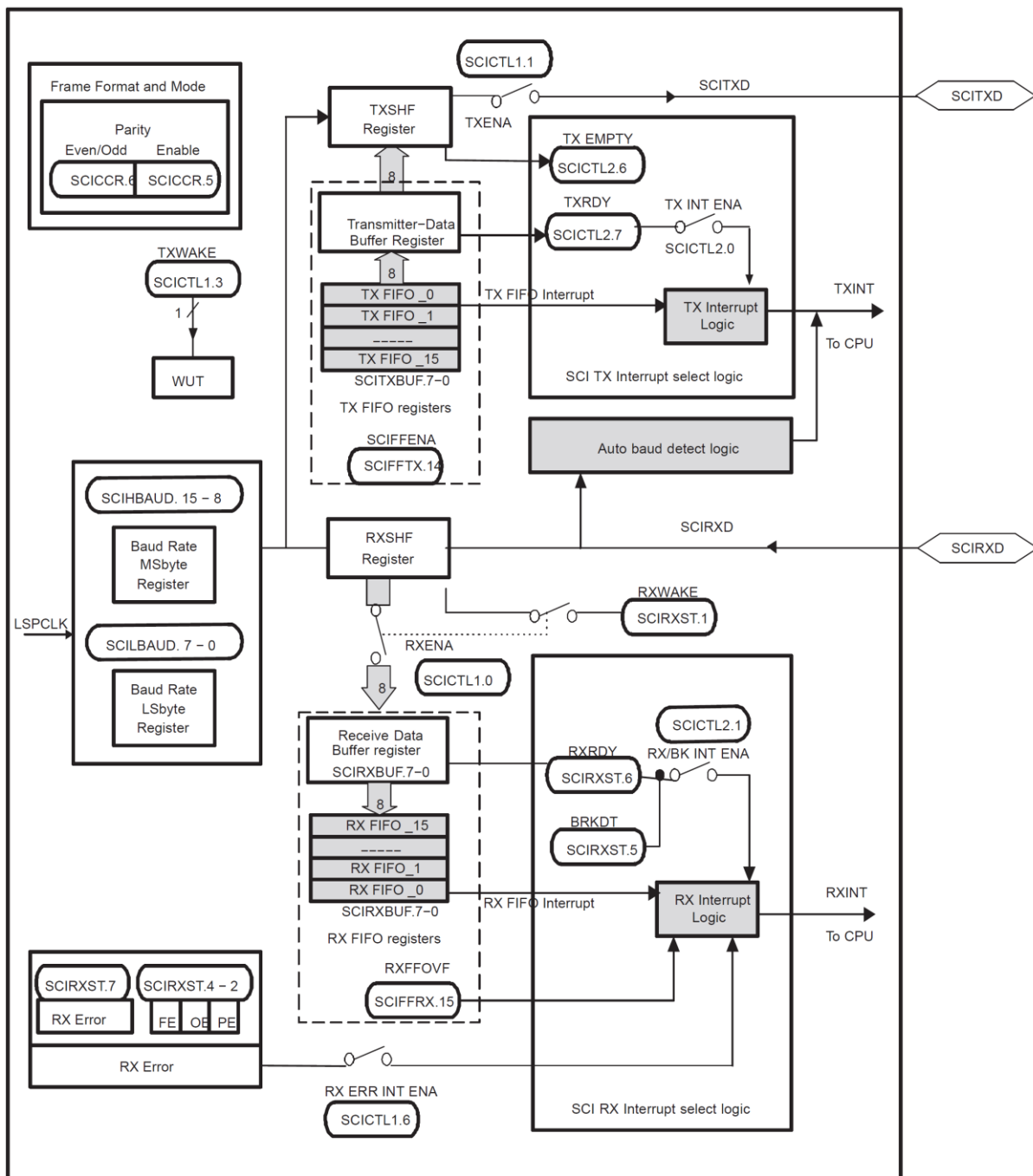
SCICCR4 – speciální bit pro testování, zdali se v obvodu nenachází smyčka SCICCR3 – způsob adresování (0 – režim nečinnosti, 1 – režim adresního bitu)

SCICCR2 – 0 – určuje počet datových bitů (000 – $111_{(2)}$ 1-8₍₁₀₎)

SCICTL1.6 – v případě výskytu chyb vyvolává přerušení (0 zakázáno, 1 povoleno)
SCICTL1.5 – softwarový reset nastaví sci do výchozího nastavení
SCICTL1.3 – TXWAKE způsob adresování (0 – režim nečinnosti, 1 – režim adresního bitu)
SCICTL1.2 – SLEEP bit (Režim spánku: 0 zakázán, 1 - povolen)
SCICTL1.1 – povolení vysílání (0 zakázáno, 1 povoleno)
SCICTL1.0 - povolení příjmu (0 zakázáno, 1 povoleno)

SCICTL 2.7 – vysílací zásobník je plný TXRDY=0 čeká na další znak TXRDY=1
SCICTL 2.6 – vysílací posuvný registr TX EMPTY=1 ke prázdný 0 posouvá data z bufferu
SCICTL 2.1 – přerušení přijímačem, 0 zakázáno RXRDY, 1 povoleno RXRDY
SCICTL2.0 – přerušení vysílacím zásobníkem, 0 zakázáno TXRDY, 1 povoleno TXRDY

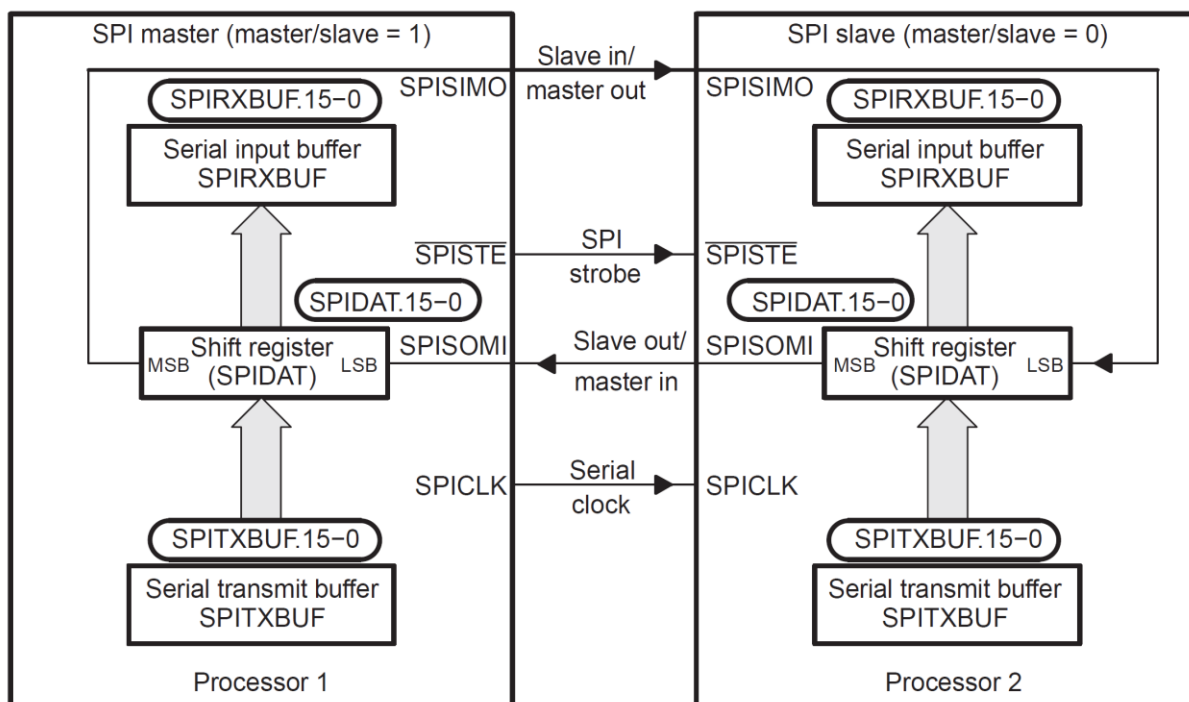
Nastavení registrů SCI jazyce C, je uvedeno v příloze č.2 na 6. a 7.straně.



(Obrázek č. 3.7.1 – Funkční schéma SCI modulu.)

3.8 SPI rozhraní

SPI komunikace probíhá v režimu master-slave, což znamená, že zařízení master je hlavním zařízením, které řídí komunikaci. Typické spojení dvou komunikačních zařízení po SPI rozhraní lze vidět na obrázku č. 3.7.1



(Obrázek č. 3.8.1 – Spojení komunikačních zařízení v SPI komunikaci.)

Zařízení master vyvolává komunikaci mezi dvěma zařízeními signálem SPISTE s tím, že také hodinový signál přenosu zajišťuje zařízení master. U vysílání se data ze zásobníku SPITXBUF (bufferu) posouvají na výstup posuvným registrem SPIDAT synchronně s hodinovým signálem SPICLK. Přijímač data naopak posouvá do vstupního zásobníku SPIRXBUF. Po zahájení komunikace obě zařízení vysílají data, i když nenesou žádnou informaci. Je na aplikačním softwaru aby rozeznal, zdali jsou předášená data s informací, či nikoli. V tomto případě nastávají tři možnosti datového přenosu. [6]

- Master posílá data, slave posílá prázdná data
- Master posílá prázdná data, slave posílá data
- Master posílá data, slave posílá data

Master obvod může iniciovat přenos dat kdykoli, avšak je na softwaru aby rozpoznal, kdy je slave zařízení připraveno vysílat.

SPI protokol má čtyři možné typy přenosů navazující na časovací signál:

- Sestupná hrana bez fázového zpoždění: SPICLK je v log.1: na sestupné hraně se signál vysílá a na náběžné přijímá.
- Sestupná hrana s fázovým zpožděním: SPICLK je v log.1: na první sestupné hraně se vysílá a na druhé přijímá.

-Náběžná hrana bez fázového zpoždění: SPICLK je v log.0: na náběžné hraně se signál vysílá a na sestupné přijímá.

- Sestupná hrana s fázovým zpožděním: SPICLK je v log.0: na první náběžné hraně se vysílá a na druhé přijímá. [6]

SPI rozhraní má celkově 4 piny pro připojení k externímu zařízení, kterými jsou: SPISIMO – vysílací pin, SPISOMI – přijímací pin, SPISTE – v master nastavení zahajuje přenos dat, SPICLK – v master režimu je to výstup časování pro slave zařízení. Přijímací i vysílací strana jsou vybaveny 16xFIFO paměťmi, kdy velikost jedné paměti je 16bitů. Přenosová rychlost je odvozena od výstupu časování z předděličky, kdy se jedná o signál LSPCLK a přenosová rychlost je LSPCLK/8. Hlavními registry řídicí operace modulu jsou:

SPICCR.7 – SW reset.

SPICCR.6 – Clock polarity, reakce vysílání a příjmu na hrany SPICLK.

SPICCR.4 – Loopback, kontrola smyčky ve vedení dat.

SPICCR.3-0 – SPI Char – počet datových bitů přenášených v jednom packetu.

SPICTL.4 – Overrun enable – povolení přerušení vysílání pro přetečení vysílacího zásobníku.

SPICTL.3 – Clock phase – určení fázového zpoždění vysílání a příjmu.

SPICTL.2 – Master/slave – určení role v zařízení v komunikaci.

SPICTL.1 - TALK – povolení vysílání.

SPICTL.0 – Povolení přerušení.

SPIST.7 – Receiver overrun flag – nastavení příznaku přerušení SPICCR.7.

SPIST.6 – SPI INT flag - nastavení příznaku přerušení SPICCR.0.

SPIST.5 – TX BUF FULL flag - nastavení příznaku přerušení při plném zásobníku pro vysílaná data.

SPIBRR.6 -0 SPI BIT RATE – Nastavení přenosové rychlosti.

SPIEMU – Posuvný registr přijímače.

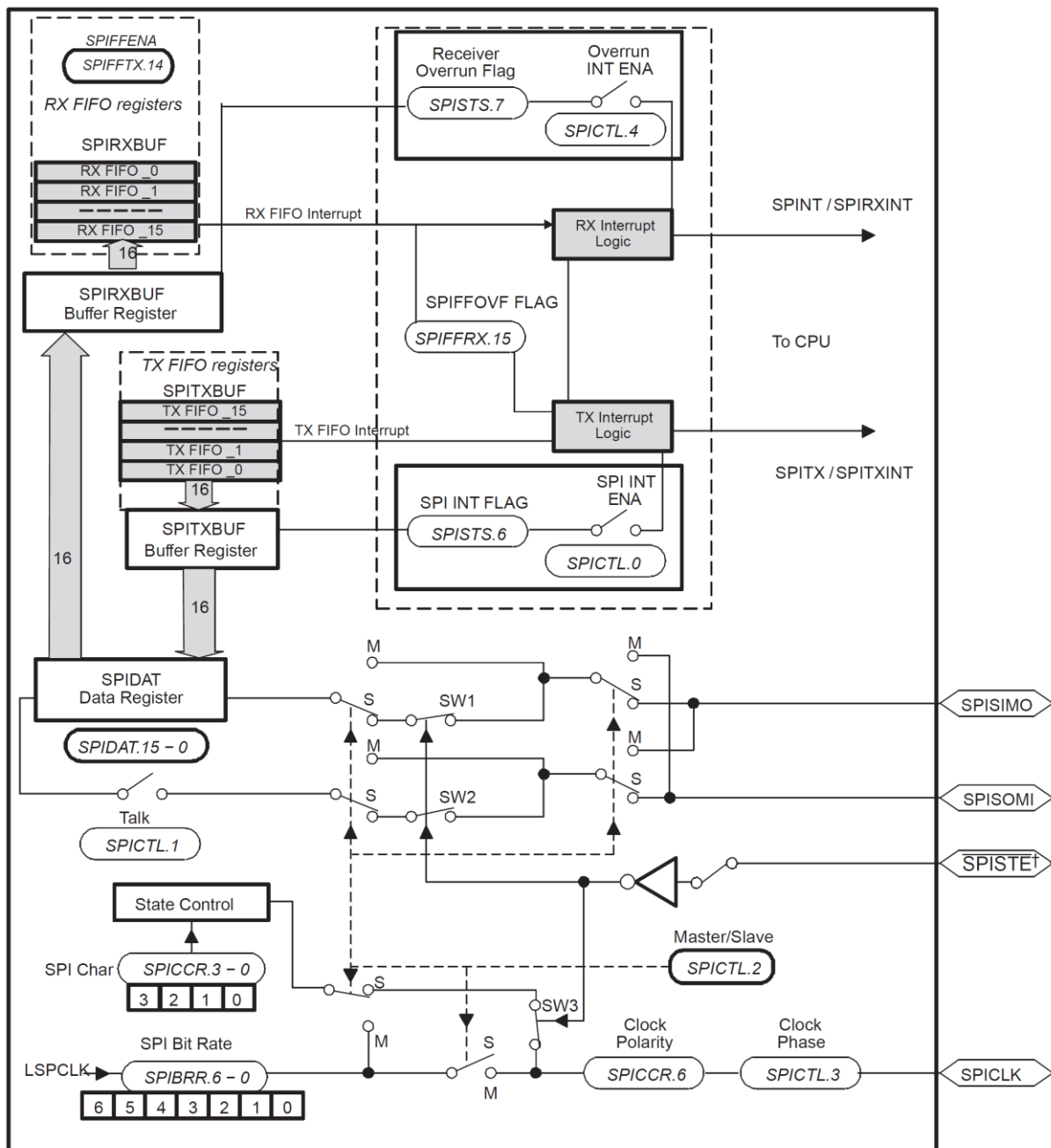
SPIRXBUF – Registr obsahuje přijatá data.

SPITXBUF – Registr obsahuje data k odeslání.

SPIDAT – Posuvný registr vysílače.

[6]

Nastavení autosekvenčního módu jazyce C, je uvedeno v příloze č.2 na 7.straně.



(Obrázek č. 3.8.2 – Funkční schéma SPI modulu.)

4. Závěr

V obsahu této práce jsem zdokumentoval zadaný řídicí systém. Jeho obsah je velice široký a rozsah diplomové práce nestačí na jeho úplný popis. Tato práce však vystihuje hlavní body pro práci s tímto systémem.

Diplomová práce má široké možnosti rozšíření, které lze ve své podstatě rozdělit na dva směry:

Vrátit se vývojově pozpátku na úplné začátky programování, kdy by se uživatel seznámil s jednotlivými příkazy, popisem struktur v jazyce c, tvorbou knihoven atd.

Druhou variantou je popsat podrobněji strukturu procesoru a popis aplikací, rozšířit obslužný program a přidat více zdrojových souborů s nastavením periférií na konkrétní funkce.

5. Seznam použité literatury

- [1] <http://www.ti.com/lit/ds/sprs439m/sprs439m.pdf>
- [2] <http://www.ti.com/lit/ug/sprufb0d/sprufb0d.pdf>
- [3] <http://www.ti.com/lit/ug/sprug04a/sprug04a.pdf>
- [4] <http://www.ti.com/lit/ug/spru812a/spru812a.pdf>
- [5] <http://www.ti.com/lit/ug/sprufz5a/sprufz5a.pdf>
- [6] <http://www.ti.com/lit/ug/sprueu3a/sprueu3a.pdf>