

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Použití mikropočítačů pro vývoj embedded aplikací

Učební texty k semináři

Autoři:

Ing. Jaroslav Lepka (Freescale Semiconductor, Rožnov p. R.)

Datum:

11. 11. – 12. 11. 2010

Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií CZ.1.07/2.3.00/09.0031

TENTO STUDIJNÍ MATERIÁL JE SPOLUFINANCOVÁN EVROPSKÝM SOCIÁLNÍM
FONDEM A STÁTNÍM ROZPOČTEM ČESKÉ REPUBLIKY

OBSAH

Obsah	1
1. Úvod – vývoj embedded aplikací	4
2. přehled embedded mikropočítačů freescale typu DSC (Digital Signal Controller).....	5
2.1. Rodina mikropočítačů Freescale DSC56F80xx	7
3. Freescale MC56F8025 - embedded mikropočítač.....	10
3.1. MC56F8025 – přehled parametrů	10
3.2. Organizace Freescale dokumentace.....	13
3.3. ADC – Analog Digital Converter.....	13
3.3.1. ADC - vlastnosti.....	13
3.3.2. ADC – blokový diagram	14
3.3.3. ADC – zpracování dat převodníkem	15
3.4. COP – Computer Operating Properly	16
3.4.1. COP – vlastnosti	16
3.4.2. COP – blokový diagram	17
3.5. PWM – Pulse Width Modulator	17
3.5.1. PWM – vlastnosti.....	18
3.5.2. PWM – blokový diagram.....	19
3.6. OCCS – On-Chip Clock Synthesis.....	20
3.6.1. OCCS – vlastnosti	20
3.6.2. OCCS – blokový diagram	21
3.7. FM – Flash Memory	21
3.7.1. FM – vlastnosti	21
3.7.2. FM – blokový diagram	22
3.8. PIT – Programmable Interval Timer.....	23
3.8.1. PIT – vlastnosti.....	23

3.8.2.	PIT – Blokový diagram.....	24
3.9.	TMR – Quad Timer	24
3.9.1.	TMR – vlastnosti	25
3.9.2.	TMR – blokový diagram	26
3.10.	QSCI – Qued Serial Communication Interface	26
3.10.1.	QSCI – vlastnosti	26
3.10.2.	QSCI – blokový diagram	28
3.10.3.	QSCI – single-wire komunikace	28
3.11.	QSPI – Qued Serial Peripheral Interface	29
3.11.1.	QSPI – vlastnosti	29
3.11.2.	QSPI – blokový diagram	30
3.11.3.	QSPI – master mód	30
3.11.4.	QSPI – wired-OR mód	31
3.12.	DAC – Digital-to-Analog Converter	31
3.12.1.	DAC – vlastnosti	31
3.12.2.	DAC – blokový diagram	32
3.13.	CMP – Comparator	32
3.13.1.	CMP – vlastnosti	32
3.13.2.	CMP – blokový diagram	33
3.14.	GPIO – General Purpose Input/Output	33
3.14.1.	GPIO – vlastnosti.....	34
3.14.2.	GPIO – blokový diagram.....	35
3.15.	I ² C – Inter-Integrated Circuit Interface	35
3.15.1.	I ² C – vlastnosti	36
3.15.2.	I ² C – blokový diagram	37
4.	Hardwarová Synchronizace PWM modulu s ADC modulem	38
4.1.	PWM a ADC synchronizace – motivace	38
4.2.	PWM a ADC synchronizace – princip.....	41

4.3.	PWM a ADC synchronizace – postup konfigurace	42
4.3.1.	PWM modul.....	42
4.3.2.	Quad Timer modul.....	43
4.3.3.	ADC modul.....	43
4.3.4.	Synchronizace - výsledek z osciloskopu.....	44
5.	Přerušení - jeden ze stavebních kamenů embedded aplikací.....	46
5.1.	Co je to přerušení	46
5.2.	Princip zpracování přerušení.....	47
5.3.	MC56F8025 - podpora přerušení z hlediska mikropočítače	49
5.4.	MC56F8025 - Fast Interrupt	50
5.5.	MC56F8025 - Fast Interrupt - zpracování	50
	Seznam použité literatury	52

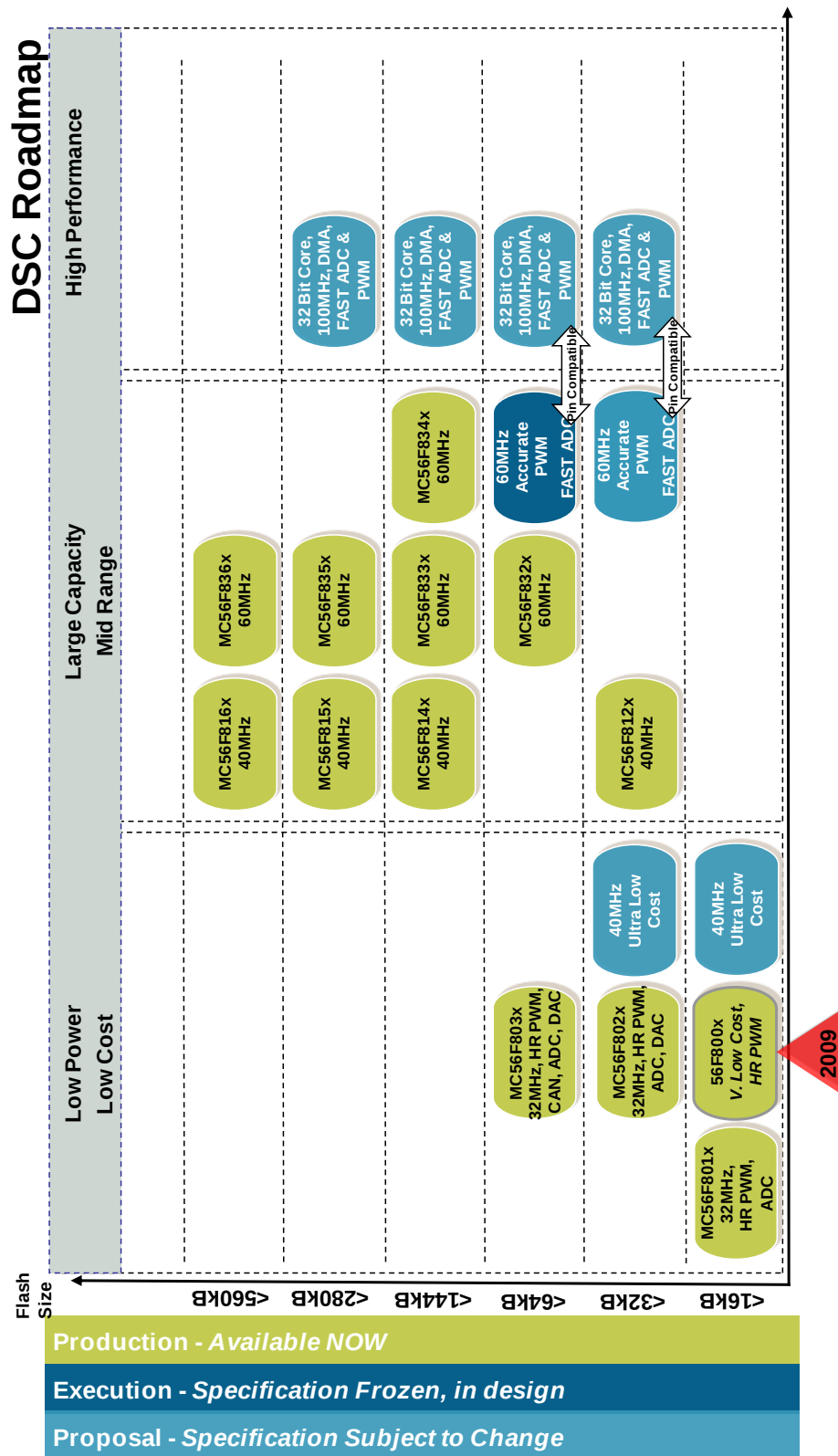
1. ÚVOD – VÝVOJ EMBEDDED APLIKACÍ

Vývoj embedded aplikace je komplexní úloha, která obvykle předpokládá poměrně rozsáhlé znalosti problematiky v dané oblasti. V mnoha případech je zapotřebí skloubit znalosti z různých oborů, což klade vysoké nároky na vývojového pracovníka. Pokud předpokládáme, že vyvíjené zařízení má být na vysoké technické úrovni a že hovoříme o mikroprocesorových systémech, je zřejmé, že řešený problém se rozpadá na podmnožinu mnohdy ne zcela souvisejících podúkolů. Příkladem může být polohový servomechanismus, od kterého očekáváme přesné řízení polohy elektrického servopohonu s definovanými přechodovými ději při změně žádané polohy a to pro oba směry otáčení. Taková úloha se na první pohled může zdát poměrně snadná do okamžiku, než si uvědomíme, co všechno je zapotřebí vyvinout a uvést do provozu, aby byly splněny požadavky na dané zařízení. Předpokládejme, že známe detailní specifikace výsledného produktu. Na řadu přichází analýza vyvíjeného zařízení a volba klíčových komponent. Základem bude zvolení vhodného typu servomotoru (PMSM, DC, krokový motor, atd.), momentu setrvačnosti rotoru s ohledem na zátěž, elektrických vlastností vinutí, které mohou významně ovlivnit požadavky na řídicí elektroniku a řídicí algoritmus. Další a neméně důležitý problém bude volba vhodného typu měření polohy a rychlosti z hlediska přesnosti, rozlišení, mechanických vlastností, možnosti montáže, zpracování signálů z čidla, atd. Následně pak přichází na řadu volba vhodného mikropočítače, který bude schopen dané signály zpracovat, spočítat řídicí algoritmus a následně generovat vhodné signály k řízení servopohonu. Signály zpracováváné a generované mikropočítačem jsou nízkovýkonové a tak bude zapotřebí vyvinout elektroniku, která bude schopna tyto nízkovýkonové signály převést na výkonové úrovně požadované servopohonem, čili bude zapotřebí vyvinout vhodný měnič. Již z výše uvedeného krátkého přehledu činností je vidět, co všechno by měl vývojový pracovník znát (programování v nějakém vyšším programovacím jazyce obvykle se jedná o jazyk C, assembler obecně a pro daný mikropočítač, detailní znalost programovacího prostředí, kompilátoru a linkeru, znalosti z návrhu hardware - analogový návrh a digitální návrh, specifické znalosti v oblasti návrhu měničů, kreslení elektrických schémat a následný návrh plošného spoje, znalosti osazovacích technologií pro hromadnou výrobu, teorii elektrických strojů, řídicích struktur a algoritmů pro řízení elektrických pohonů, teorii řízení obecně, měření elektrických a neelektrických veličin, mechaniku, simulace, mikropočítače, atd.).

2. PŘEHLED EMBEDDED MIKROPOČÍTAČŮ FREESCALE TYPU DSC (DIGITAL SIGNAL CONTROLLER)

Rodina mikropočítačů 56800/E dosahuje výpočetního výkonu až 120 MIPS. Jenda z podstatných výhod této rodiny spočívá v kombinaci výpočetního výkonu procesorů typu DSP a vlastností klasických mikrokontrolérů (MCU) a to na jednom čipu. Jednou z velmi důležitých podmínek využití mikropočítačů pro vývoj embedded aplikací je kromě vhodného jádra sada výkonných periférií, které vytvářejí rozhraní mezi mikropočítačem a řízeným systémem. Neméně důležitý požadavek kladený na mikropočítač je jeho cena. Rozsah použití těchto typů mikropočítačů je obrovský v různých oborech lidské činnosti jako např.: průmyslové řídicí systémy, řízení motorů všech typů, domácí spotřebiče, inteligentní řízení osvětlení, automobilový průmysl, záložní zdroje (UPS), spínané zdroje, měniče pro solární panely, aplikace se senzory, monitorovací systémy v medicíně, atd.

Přehled DSC mikropočítačů, které jsou aktuálně k dispozici pro vývoj, je na Obr. 2.1. Zároveň je z tohoto obrázku možno vyčíst směry, kterými se vývoj ubírá. Zeleně označené rámečky říkají, že daný produkt je aktuálně v produkci. Tmavě modré rámečky označují produkty, jejichž specifikace byla ukončena a jsou ve fázi návrhu, testování a verifikace. Poslední množina světle modrých rámečků ukazuje vizi do budoucna, což znamená, že dané produkty jsou ve stádiu specifikace.



Obr. 2.1: Freescale DCS - přehled

Poměrně velké množství odnoží rodiny mikropočítačů s jádrem 56800E poskytuje vývojáři velkou flexibilitu při volbě vhodného typu pro danou embedded aplikaci. Jednotlivé typy mikropočítačů se liší ve velikosti paměti RAM (paměť dat), paměti FLASH (paměť programu a konstantních dat), druhu periférií na čipu a jejich počtu. Základem pro všechny typy je jádro mikropočítače 56F800E, což vývojáři umožňuje jednoduše převést software na libovolný mikropočítač se stejným jádrem. Vývojář si nicméně musí dát pozor na to, jaký typ periférie potřebuje pro danou aplikaci a jak by periférie měly mezi sebou spolupracovat. Některé specifické požadavky na aplikaci pak vedou na daný typ mikropočítače. Ikdyž je většina zdrojového kódu bez problémů přenositelná na jiný mikropočítač z téže rodiny, aplikace jako celek nemusí být na jiném mikropočítači realizovatelná.

Pro naši práci si vybereme jeden konkrétní mikropočítač z řady cenově přijatelných typů - MC56F8025. Tento mikropočítač je členem rodiny DCS s označením MC56F80xx, kde poslední dvojčíslí reprezentované znaky xx specifikuje jeden konkrétní typ.

2.1. Rodina mikropočítačů Freescale DSC56F80xx

V současné době má rodina mikropočítačů Freescale DSC56F80xx celkem devět zástupců. Základem všech typů je jádro 56F800E. Společným faktorem je napájecí napětí 3.3 V a frekvence, na které mikropočítače běží, v tomto případě to je 32 MHz. Jedním z faktorů omezujících frekvenci je přístupová doba do paměti FLASH na dané technologii.

Faktory odlišující jednotlivé zástupce rodiny jsou: velikosti pamětí RAM, FLASH, velikost a typ pouzdra (většinou se jedná o pouzdro LQFP), počet periférií a typy periférií. Zde bych rád podotknul, že mezi jednotlivými zástupci mohou být rozdíly i co se týká periférií stejného názvu jako např. ADC (stejný název ještě neznamena stejné vlastnosti periférie)

Standardní ladicí rozhraní je JTAG/Enhanced OnCE.

	56F8002	56F8006	56F8011	56F8013	56F8014	56F8023	56F8025	56F8036	56F8027/37
Performance	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs
Temperature Range (V)	-40C~105C	-40C~105C	-40C~125C	-40C~125C	-40C~125C	-40C~105C	-40C~105C	-40C~105C	-40C~105C
Voltage Range	1.8V - 3.6V	1.8V - 3.6V	3.0V - 3.6V	3.0V - 3.6V	3.0V - 3.6V	3.0V - 3.6V	3.0V - 3.6V	3.0V - 3.6V	3.0V - 3.6V
Voltage Regulator	On-Chip	On-Chip	On-Chip	On-Chip	On-Chip	On-Chip	On-Chip	On-Chip	On-Chip
Program/Data Flash	12KB	16KB	12KB	16KB	16KB	32KB	32KB	64KB	32KB / 64KB
Program/Data RAM	2KB	2KB	2KB	4KB	4KB	4KB	4KB	8KB	4KB/8KB
Program Security	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
On Chip Relaxation Osc.	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PLL	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
COP (Watchdog)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PWM (96 Mhz Clock)	1 x 6ch	1 x 6ch	1 x 6ch	1 x 6ch	1 x 5ch	1 x 6ch	1 x 6ch	1 x 6ch	1 x 6ch
PWM Fault Inputs	4	4	4	4	3	4	4	4	4
12-bit ADCs	2 x 8ch	2 x 12ch	2 x 3ch	2 x 3ch	2 x 4ch	2 x 3ch	2 x 4ch	2 x 5ch	2 x 8ch
12-bit DACs	0	0	No	No	No	2	2	2	2 (Pinned out)
Analog Comparator	3	3	No	No	No	2	2	2	2
Prog Gain Amp	2	2	No	No	No	No	No	No	No
16-bit Timers	3	3	4	4	4	4	4	4	8
Prog. Interval Timers	1 (RTC)	1 (RTC)	No	No	No	1	3	3	3
GPIO (max) (+/-8mA)	23	40	26*	26*	26*	26*	35*	39*	53*
IIC	1	1	1	1	1	1 - QIIC	1 - QIIC	1 - QIIC	1 - QIIC
SCI(UART) / LIN Slave	1 - SCI	1 - SCI	1 - SCI	1 - SCI	1 - SCI	1 - QSCI	1 - QSCI	1 - QSCI	1 - QSCI
SPI (Synchronous)	1 - SPI	1 - SPI	1 - SPI	1 - SPI	1 - SPI	1 - QSPI	1 - QSPI	1 - QSPI	1 - QSPI
CAN	No	No	No	No	No	No	No	MSCAN	MSCAN
JTAG/EO nCE	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Power Consumption	IDD = 45.6mA; IDDA = 4.5mA	IDD = 45.6mA; IDDA = 4.5mA	IDD = 42mA; IDDA = 13.5mA	IDD = 42mA; IDDA = 13.5mA	IDD = 48mA; IDDA = 18.8mA	IDD = 48mA; IDDA = 18.8mA	IDD = 48mA; IDDA = 18.8mA	IDD = 48mA; IDDA = 18.8mA	IDD = 48mA; IDDA = 18.8mA
Package	32LQFP (8p)	28SOIC 32LQFP 32SDIP 48LQFP	32LQFP	32LQFP	32LQFP	32LQFP	44LQFP	48LQFP	64LQFP

Obr. 2.2: Rodina mikropočítačů Freescale MC56F80xx

Mikropočítače **DSC56F8011/13/14** jsou kompatibilní z hlediska typu periférií.

Mikropočítače **DSC56F8023/25/36/37** jsou kompatibilní z hlediska typu periférií.

Mikropočítače **DSC56F8002/6** jsou kompatibilní z hlediska typu periférií.

3. FREESCALE MC56F8025 - EMBEDDED MIKROPOČÍTAČ

3.1. MC56F8025 – přehled parametrů

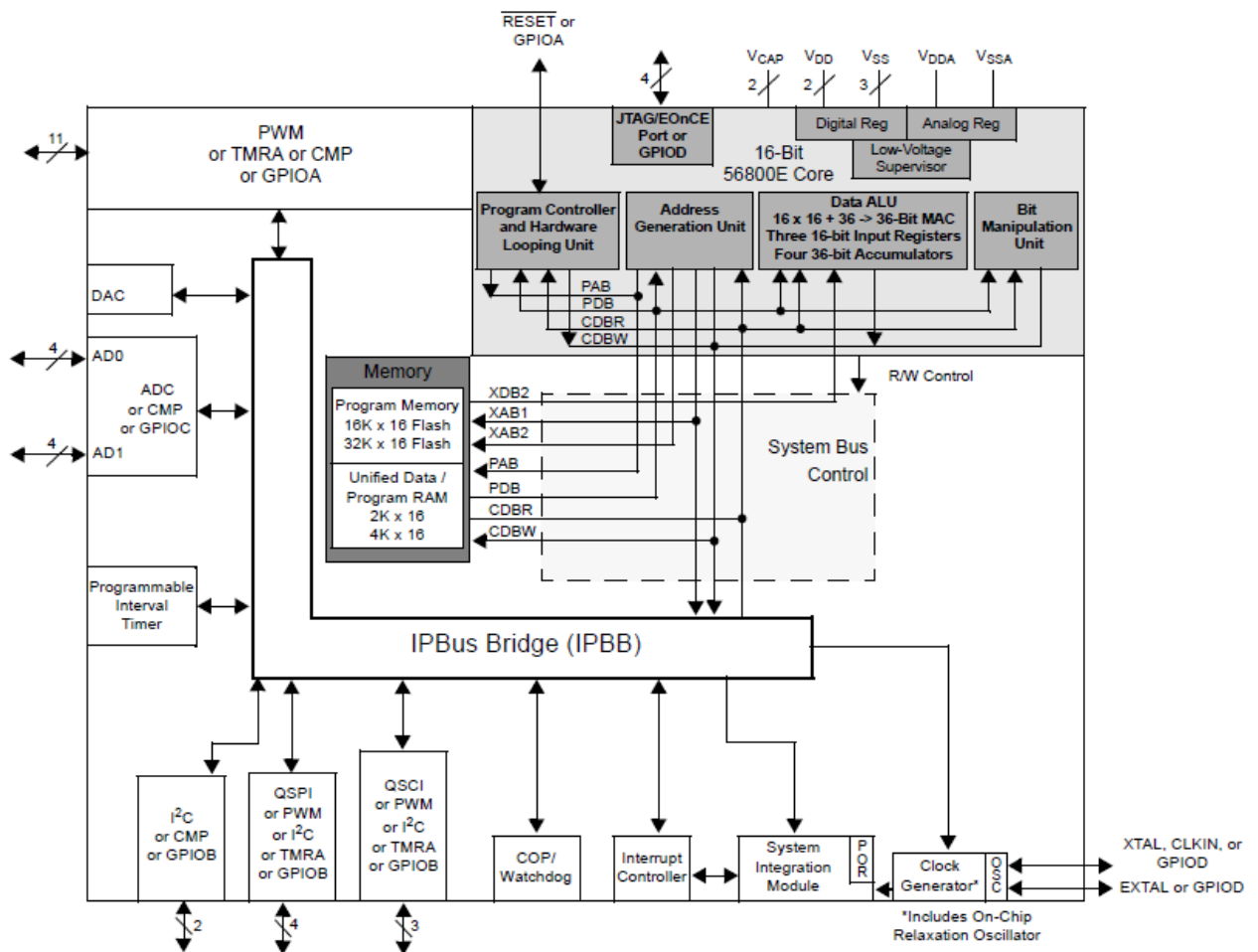
Mikropočítač DSC56F8025 je založený na jádře DSP56800E, které kombinuje výpočetní výkon DSP procesoru a výhodami klasického mikrokontroléru. Doplněním o flexibilní sadu výkonných periferních modulů integrovaných spolu s pamětí RAM a Flash na jednom čipu nabízí mikropočítač DSC56F8025 cenově výhodné řešení pro širokou škálu aplikačního využití. Je vhodný pro aplikace v průmyslu, řízení pohonů, domácích spotřebičů, zpracování inteligentních sensorů, bezpečnostní systémy, spínané zdroje, střídače pro obecné použití, UPS systémy, lékařské aplikace, atd.

Základní vlastnosti:

- Napájení digitální – 3.3 V
- Napájení analogové – 3.3 V
- Napěťový regulátor integrovaný na čipu
- Frekvence jádra – 32 MHz, což odpovídá výpočetnímu výkonu maximálně 32MIPS
- 32KB (16K x 16) Program Flash
- 4KB (2K x 16) Unified Data/Program RAM
- PWM modul – flexibilní a velmi komplexní peiferie s 6 PWM kanály
- Dva nezávislé 4-kanálové 12-bitové ADC převodníky (Analog-to-Digital Converter)
- Dva interní 12-bitové Digital-to-Analog Converters (DACs)
- Dva analogové komparátory
- Tři programovatelné časovače – PIT (Programmable Interval Timer)
- Jeden modul QSCI (Queued Serial Communication Interface) s LIN slave funkcí

- Jeden modul QSPI (Queued Serial Peripheral Interfaces)
- Jeden modul 16-bitových časovačů čítající čtyři časovací/čítací moduly – Quad Timer (TMR)
- Jeden modul I2C (Inter-Integrated Circuit)
- Watchdog – COP (Computer Operating Properly)
- Relaxační oscillator integrovaný na čipu
- Možno použít - interní oscilátor nebo externí krystal, rezonátor
- PLL na čipu
- Integrovaný Power-On Reset (POR) a Low-Voltage Interrupt (LVI)
- JTAG/Enhanced On-Chip Emulation (OnCE™)
- Až 35 GPIO
- Pouzdro - 44LQFP
- Teplotní rozsah - -40°C až 105°C

Blokové schéma mikropočítače je na Obr. 3.1.



Obr. 3.1: Blokové schéma mikropočítače 56F8025

3.2. Organizace Freescale dokumentace

Firma Freescale dělí dokumentaci pro jednotlivé mikropočítače do několika ucelených dokumentů:

- Data sheet – dokument k jednotlivým mikropočítačům. Obsahuje specifické údaje pro daný čip jako např.: napájení, frekvence jádra, mapa pamětí, velikosti Flash a RAM pamětí, typy periférií a jejich bazové adresy, rozsahy vstupních a výstupních napětí, různé důležité údaje specifické pro daný čip, atd.
- Peripheral Reference Manual – dokument popisující periferie mikropočítače pro danou rodinu.
- Reference Manual – vyčerpávající popis jádra mikropočítače
- Chip Errata – seznam zjištěných chyb a jejich případná řešení

3.3. ADC – Analog Digital Converter

ADC modul se skládá ze dvou nezávislých ADC převodníků, z nichž každý obsahuje samostatný sample and hold (S/H) obvod. Každý z převodníků má osm analogových vstupů jednoduše ovladatelných díky vstupnímu multiplexeru. ADC kromě analogové části obsahuje i výkonnou řídicí logiku umožňující efektivní a snadné nastavování a přeprogramování v reálném čase.

3.3.1. ADC - vlastnosti

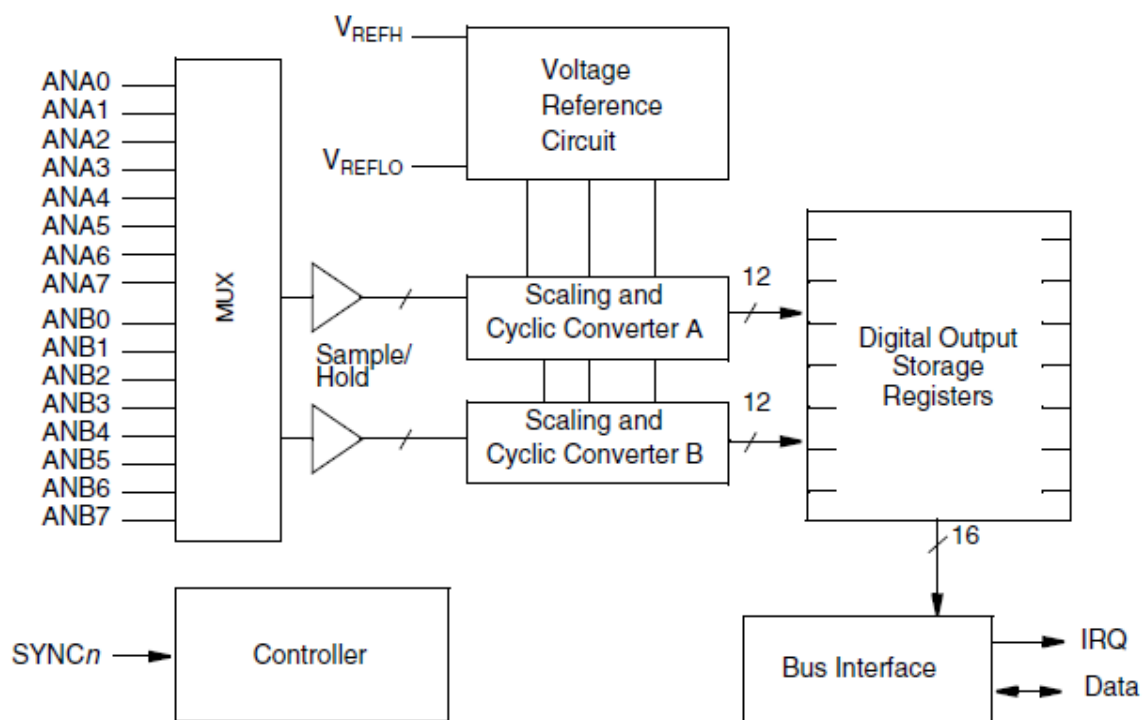
ADC modul charakterizují následující vlastnosti:

- Rozlišení – 12-bitů
- Maximální frekvence hodin převodníku – 5.33 MHz, což odpovídá periodě 187 ns
- Vzorkovací rychlost – až $1.78 \cdot 10^6$ vzorků za sekundu
- Doba převodu jednoho převodníku (první převod) – 8.5 ADC hodinových cyklů ($8.5 \times 187 \text{ ns} = 1.59 \mu\text{s}$)

- Doba převodu jednoho převodníku (následující převody) – 6 ADC hodinových cyklů ($6 \times 187 \text{ ns} = 1.125 \text{ } \mu\text{s}$)
- Doba odpovídající osmi převodům - 26.5 ADC hodinových cyklů ($26.5 \times 187 \text{ ns} = 4.97 \text{ } \mu\text{s}$) s využitím paralelního módu
- ADC synchronizace s PWM modulem přes SYNC0/1 vstupní signály – propojeno vnitřně na čipu
- Schopnost sekvenčně převést a uložit až 16 měření
- Schopnost převést a uložit až 8 měření pro každý z převodníků v případě použití simultánního a paralelního módu
- Schopnost převést a uložit až 8 měření pro každý z převodníků v případě, že převodníky běží asynchronně
- Volitelné přerušení v případě ukončení převodu (end of scan), překročení limitů (out-of-range limit exceeded) nebo v případě průchodu nulou (zero crossing)
- Volitelná korekce převedeného výsledku pomocí odečtení předprogramované offset hodnoty
- Signed nebo unsigned výsledky
- Diferenciální vstupy nebo signál vztažený k zemi ADC modulu

3.3.2. *ADC – blokový diagram*

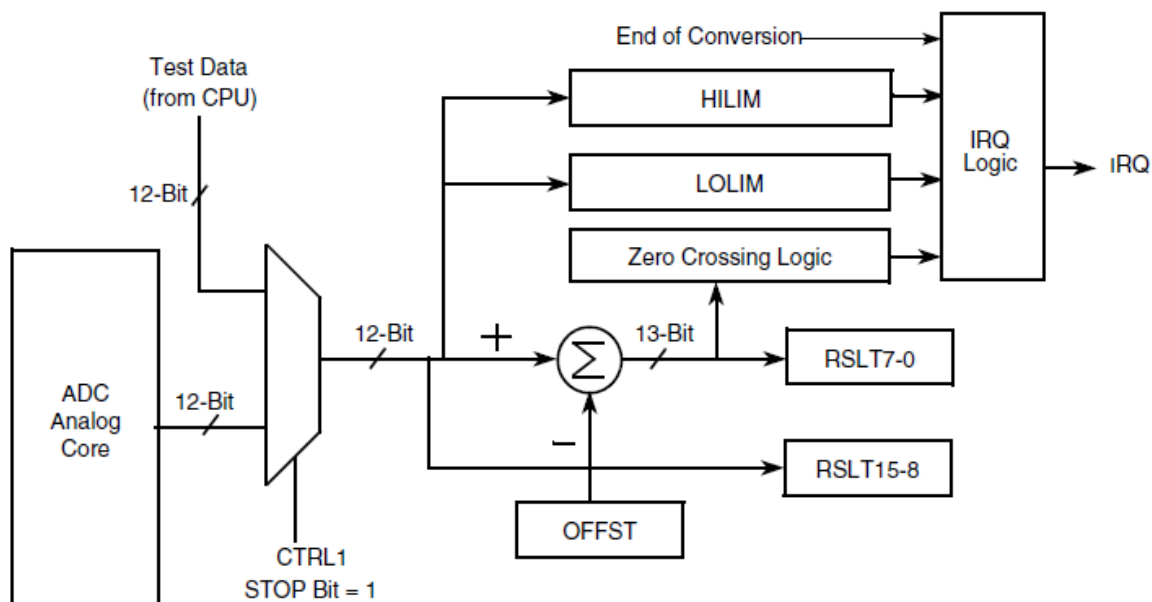
Následující obrázek Obr. 3.2 demonstruje konfiguraci ADC modulu



Obr. 3.2: Blokový diagram ADC modulu se dvěma nezávislými převodníky

3.3.3. *ADC – zpracování dat převodníkem*

Obr. 3.3 ukazuje zpracování výsledku převodu ADC převodníkem.



Obr. 3.3: Zpracování výsledku převodu ADC převodníkem

Výsledky převodu (hrubá data) se následně zpracovávají jednotkou, která je součástí ADC převodníku. Analogový signál po převodu jde do sčítačky, kde

dochází volitelně k odečtení hodnoty nastavené uživatelem v offset registru (OFFST). Následně je výsledek uložen do result registru (RSLT). Hodnota offsetu může být aplikována na result registry 0 až 7 (RSLT0-7) a to proto, že máme k dispozici osm offset registrů (OFFST0-7). Ve zbývajících result registrech 8-15 (RSLT8-15) jsou uložena přímo hrubá data převedená převodníkem bez jakéhokoliv předzpracování. Ve stejném okamžiku jsou data jdoucí z převodníku porovnávána s uživatelsky definovanými mezemi HILIM (horní limit) a LOLIM (dolní limit) a generují se přerušení v případě, že jsou povolena. Data z result registru jsou ještě porovnávána logikou testující průchod nulou a v případě průchodu nulou může být opět generováno přerušení.

3.4. COP – Computer Operating Properly

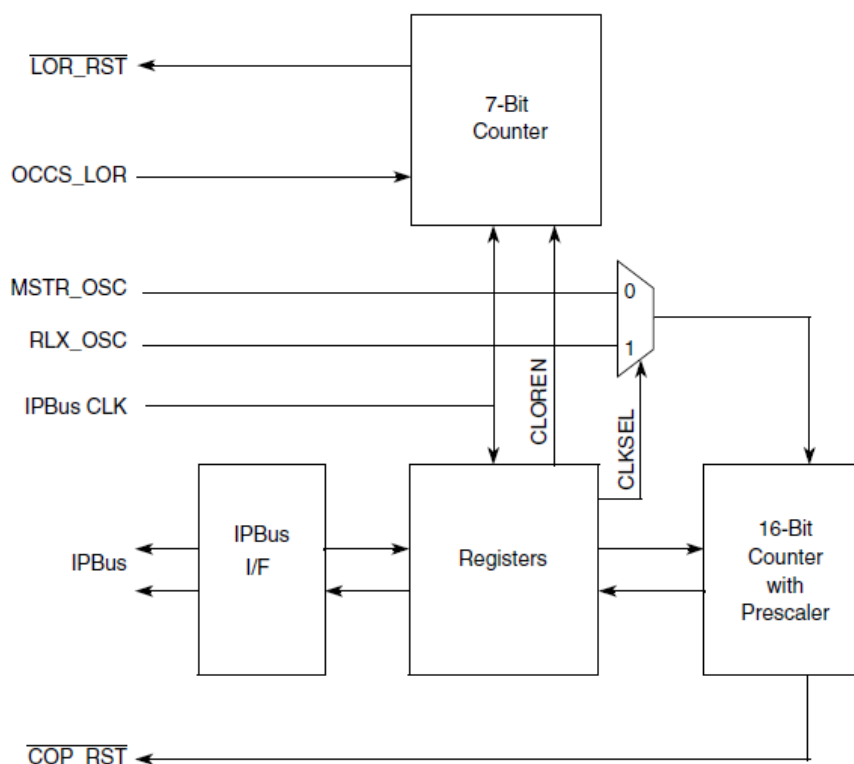
COP (Computer Operating Properly) modul slouží pro opětovné zotavení naprogramovaného kódu v případě, že program z nějakých důvodů zabloudí. Běžně se tento systém nazývá Watchdog. Princip Watchdogu spočívá ve volně běžícím down counteru, který generuje reset v případě, že counter dočítá do 0. COP je možné povolit nebo zakázat v závislosti na požadavcích aplikace. Uživatelský software musí pravidelně provádět obsluhy COP logiky zápisem specifických čísel v do counter registru ve správném pořadí. V tomto případě je nutno periodicky obnovovat CNTR (Counter Register) registr sekvencí čísel 0x5555 a následně 0xAAAA.

3.4.1. COP – vlastnosti

COP modul obsahuje následující charakteristické vlastnosti:

- Programovatelný timeout = $(\text{cop_prescaler} \times (\text{TIMEOUT} + 1)) \times (\text{Selected OSC clock cycles})$, kde TIMEOUT může být v rozsahu 0x0000 až 0xFFFF
- Programovatelné chování v případě Wait a Stop módů
- V debug módu je COP modul zakázaný
- Volitelný zdroj hodin pro COP modul
- COP modul rovněž zpracovává signál z PLL modulu, při jehož ztrátě dochází k resetování DSC po 128 cyklech

3.4.2. COP – blokový diagram



Obr. 3.4: Blokový diagram COP modulu

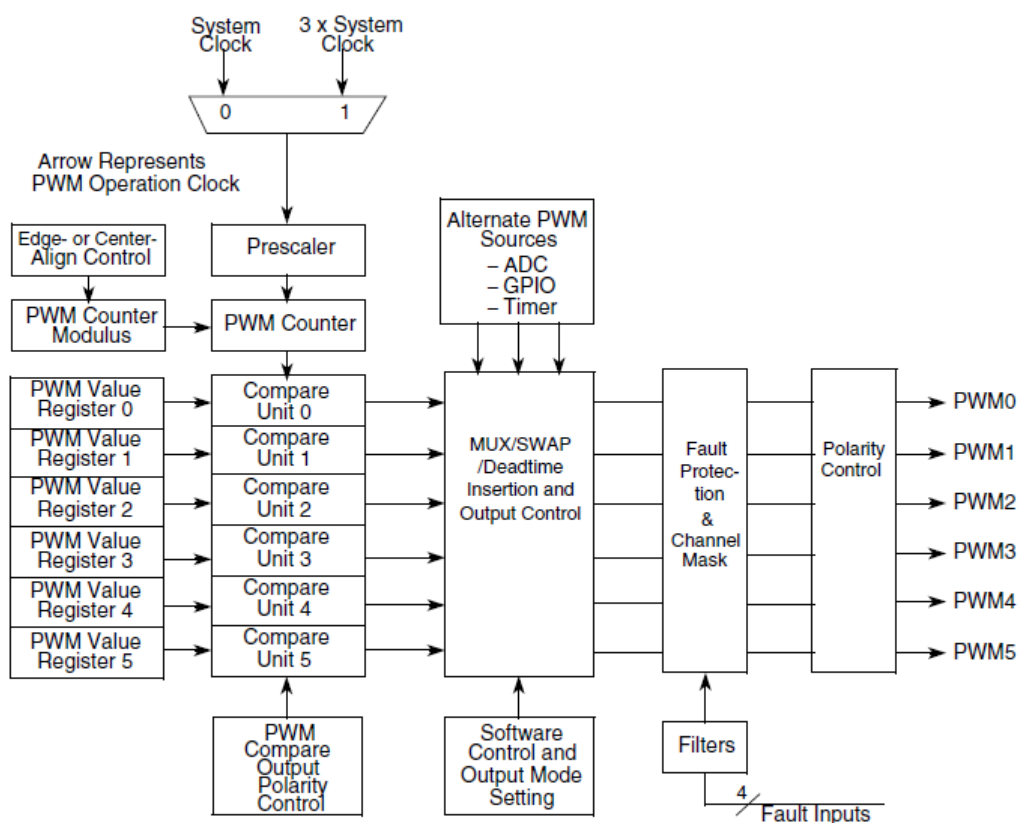
3.5. PWM – Pulse Width Modulator

PWM modul je komplexní a poskytuje celou řadu komplikovaných nastavení v závislosti na připojeném externím zařízení. PWM modul umožňuje generovat tři komplementární výstupy, šest nezávislých (independent) výstupů nebo jejich kombinace jako např.: jeden komplementární a čtyři nezávislé výstupy. Oba módy edge- a center-aligned dovolují řídit šířku pulsu od 0% až do 100%. PWM modul pro generaci PWM signálu používá 15-bitový čítač a to pro všech 6 kanálů. Rozlišení PWM signálu závisí na zvoleném módu a pro edge-aligned mód činí jeden hodinový signál, v případě center-aligned módu jsou to pak dva hodinové signály. Hodinový signál PWM modulu je volitelný a může být roven systémovým hodinám nebo trojnásobku systémových hodin. V případě generování signálu v komplementárním módu je automaticky vkládaný deadtime (řízeno hardwarově). Každý PWM výstup může být volitelně řízen PWM generátorem, časovačem, výsledkem ADC převodu, GPIO pinem nebo manuálně softwarem. Nezávisle lze řídit výstupní polaritu PWM signálu pro horní a dolní výstupy. Rovněž lze generovat nesymetrický PWM výstup.

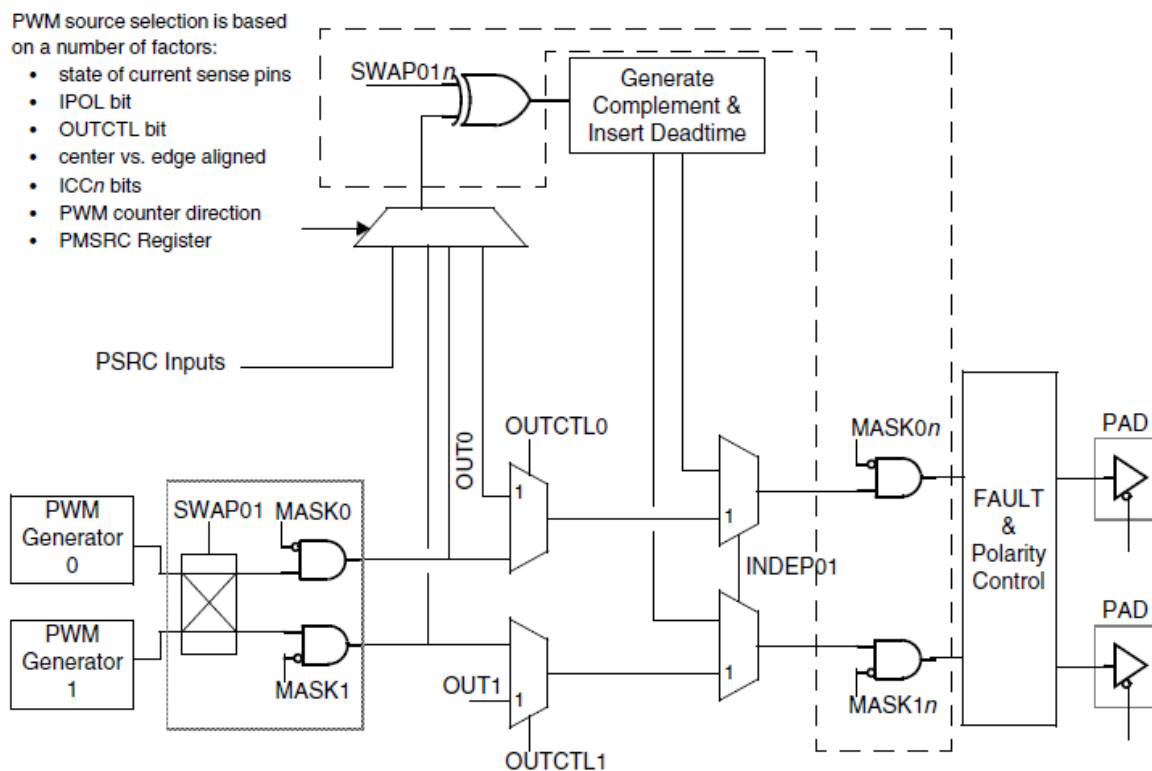
3.5.1. *PWM – vlastnosti*

- PWM operační hodiny běží buď na stejné frekvenci jako systémové hodiny, nebo na trojnásobku systémových hodin.
- 6 PWM výstupních signálů
 - Všechny nezávislé (independent)
 - Tři komplementární páry (Complementary pairs)
 - Kombinace nezávislých a komplementárních párů
- Vlastnosti PWM výstupů v případě generování komplementárních párů
 - Generování deadtime nezávisle pro nástupnou a sestupnou hranu
 - Oddělené generování korekce šířky pulsu pro horní a spodní výstupy prostřednictvím software
 - Asymetrický PWM výstup v případě center align módu
 - Oddělené řízení polarity výstupů pro horní a dolní PWM výstupy
 - Edge- nebo center-aligned PWM signály
 - 15-bitové rozlišení
 - Reload schopnost uprostřed PWM cyklu (half-cycle reload)
 - Integrovaná reload rate od 1 do 16
 - Možnost řízení PWM výstupů softwarem
- Programovatelná ochrana proti chybám
- Řízení polarity PWM výstupů
- Push-Pull nebo open drain módy pro PWM piny
- Programovatelný vstupní filtr pro chybové piny
- Ochrana PWM modulu ve formě registrů s ochrannou proti zápisu

3.5.2. *PWM – blokový diagram*



Obr. 3.5: Blokový diagram PWM modulu



Obr. 3.6: Detailní blokový diagram funkcí MUX, Swap and Deadtime

3.6. OCCS – On-Chip Clock Synthesis

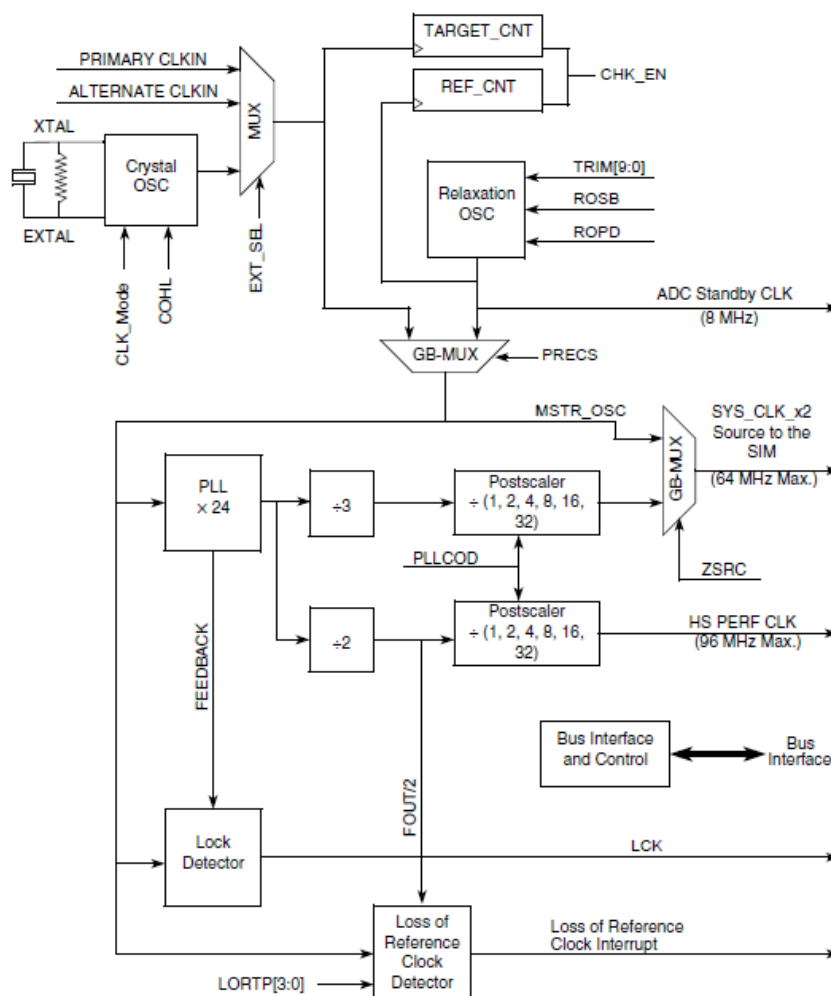
OCCS modul vyrábí systémové hodiny pro SIM (System Integration Module) modul, který pak generuje různé hodinové signály pro celý čip. Kromě tohoto signálu generuje rovněž signál odpovídající trojnásobku hodinového signálu, maximálně však 96 MHz. Jeho součástí je interní relaxační oscilátor, který umožňuje, aby čip běžel i bez externího hodinového signálu.

3.6.1. OCCS – vlastnosti

Vlastnosti OCCS modulu jsou následující:

- Interní relaxační oscilátor
- Zpracování signálu z krystalového oscilátoru, rezonátoru, externích hodin
- PLL (Phase Locked Loop) – generace hodinového signálu technikou fázového závěsu
- 3-bitový postscaler umožňuje řídit frekvenci signálu generovaného PLL modulem
- Generace trojnásobné frekvence hodin pro časovač a PWM moduly
- Standby mód relaxačního oscilátoru

3.6.2. OCCS – blokový diagram



Obr. 3.7: Blokový diagram OCCS modulu s externím krystalovým oscilátorem

3.7. FM – Flash Memory

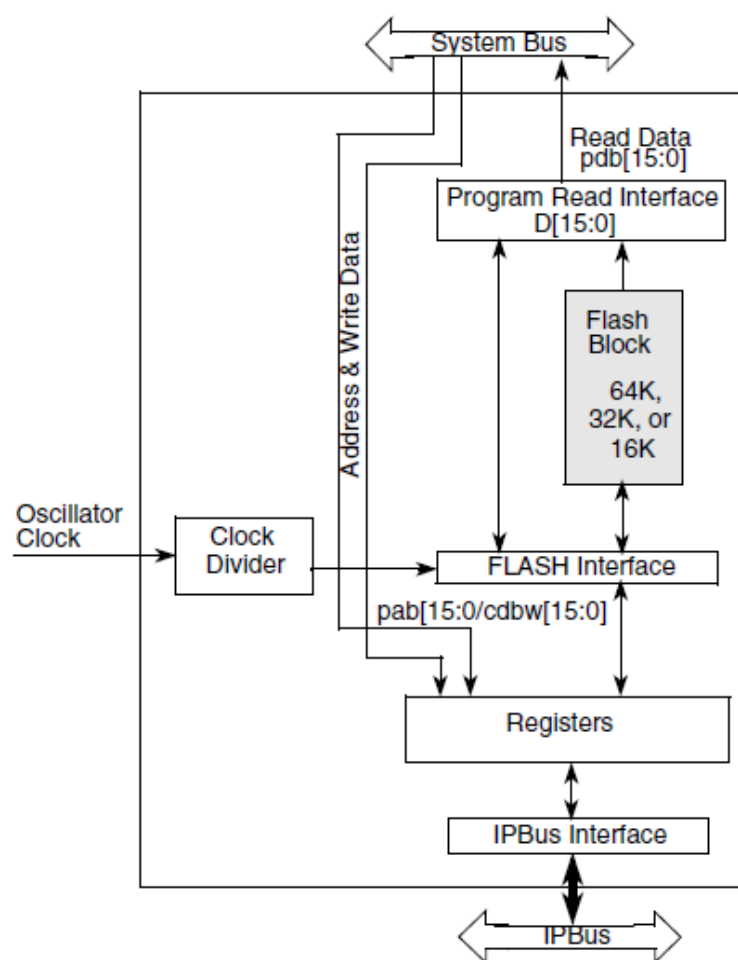
Paměť Flash (FM) je typu non-volatile, což znamená, že informace uložená v paměti zůstává zachovaná i po odpojení napájení. Používá se pro ukládání programu. FM implementovaná na mikropočítačích Freescale podporuje přeprogramování bez použití externího napájení. Mazání a programování se provádí přes rozhraní jádra mikropočítače.

3.7.1. FM – vlastnosti

- 32 MHz – pro všechny přístupy do programové Flash paměti
- Programování a mazání – automatizovaný proces

- přerušení na události – command completion, command buffer empty a access error
- Rychlé mazání stránky
- Jediné napájení pro programování a mazání
- Vlastnosti bezpečnosti
- System – sector protection

3.7.2. *FM – blokový diagram*



Obr. 3.8: Blokový diagram Flash paměti

Flash paměť je organizovaná následujícím způsobem:

- Flash paměť o velikosti od 32 kB do 64 kB je konstruovaná jako 64 kB blok

- Blok paměti o velikosti 64 kB je organizovaný do 1024 řádků po 64 bytech
- Blok paměti o velikosti 32 kB je organizovaný do 512 řádků po 64 bytech
- Je rozdělena do stránek, každá stránka obsahuje 8 řádků po 64 bytech, což je 512 bytů
- Nejmenší oblast, kterou lze smazat je jedna stránka
- Mazací proces podporuje i tzv. mass erase, který maže celý paměťový blok
- Vymazaný bit je čten jako logická jednička, zatímco naprogramovaný jako logická nula
- Na paměťovou oblast se odkazujeme prefixem P:
- Přístup do paměti Flash pro čtení probíhá každý hodinový cyklus bez penalizace (žádné čekací cykly - wait states)

3.8. PIT – Programmable Interval Timer

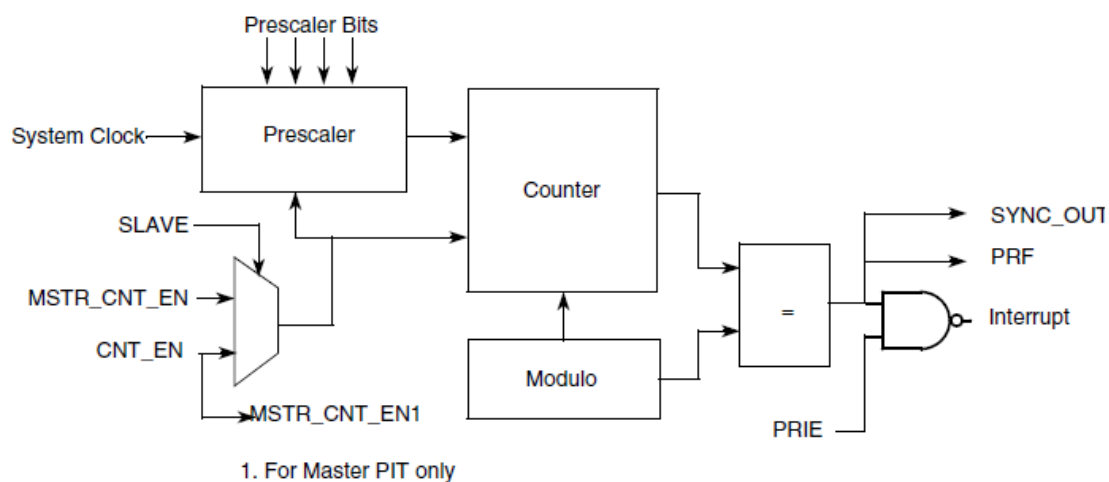
PIT modul je velmi jednoduchý časovací modul sloužící pro nenáročné časování aplikace. Obsahuje 16-bitový čítač vpřed, modulo registr a řídicí registr. Modulo a řídicí registry slouží jak pro čtení, tak pro zápis, zatímco registr čítače slouží pouze pro čtení.

Modulo registr je naplněn číslem, do kterého bude čítač čítat. Pokud čítač dočítá do hodnoty modulo, znovu se nastaví na číslo 0x0000 a začne nové čítání. V případě dosažení hodnoty modula se nastaví příznak, který může generovat přerušení, pokud je povoleno.

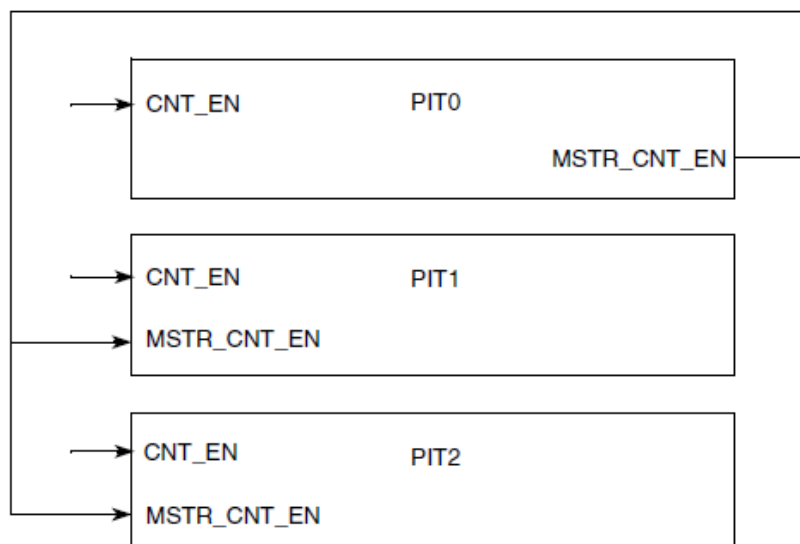
3.8.1. PIT – vlastnosti

- 16-bitový časovač/čítač
- Programovatelné čítání modulo
- Slave mode – umožňuje synchronizaci povolení několika PIT modulů

3.8.2. PIT – Blokový diagram



Obr. 3.9: Blokový diagram PIT modulu



Obr. 3.10: Propojení PIT modulů ve slave módu

3.9. TMR – Quad Timer

Quad Timer modul je velmi komplexní modul, který poskytuje obrovskou flexibilitu v nastavení funkčnosti podle požadavků aplikace. Modul obsahuje čtyři identické submoduly. Každý submodul obsahuje:

- Prescaler
- Counter
- Load registr

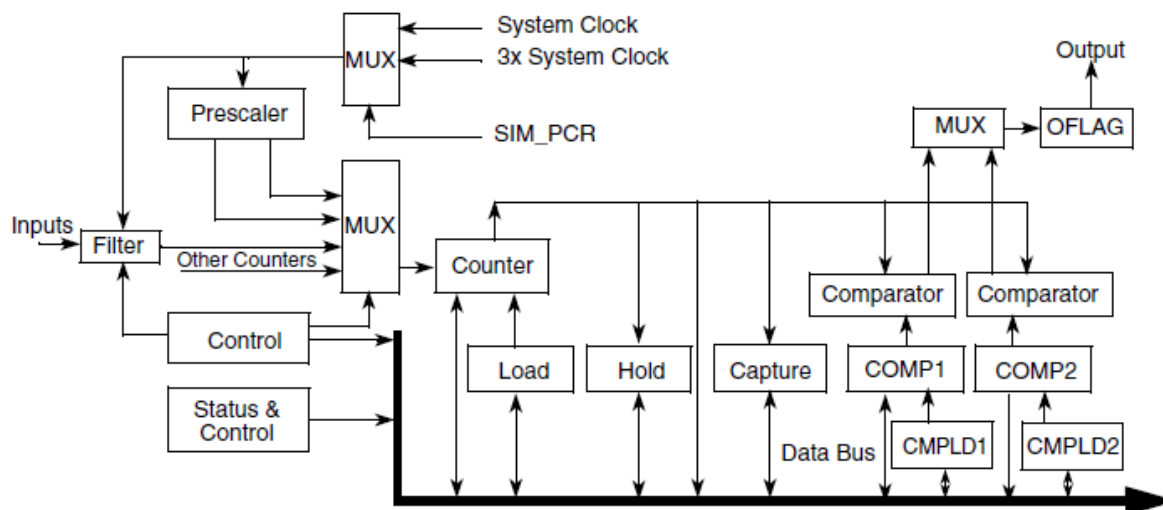
- Hold registr
- Capture registr
- Dva compare registry
- Jeden status a control registr
- Jeden control registr

3.9.1. *TMR – vlastnosti*

- Čtyři 16-bitové časovače/čítače
- Schopnost čítání nahoru a dolů (counting up and down)
- Kaskádování čítačů
- Čítání modulo
- Maximální rychlost čítání – sytem clock/2 v případě externího signálu
- Maximální rychlost čítání – sytem clock nebo 3x system clock v případě interních hodin
- Čítání – jednou nebo opakovaně
- Čítač může být preloaded
- Časovače mohou sdílet vstupní piny
- Separátní prescaler pro každý čítač
- Každý čítač umožňuje capture a compare
- Programovatelné chování během debug módu
- Programovatelný vstupní filtr
- Start čítačů může být synchronizován

3.9.2. TMR – blokový diagram

Blokový diagram na Obr. 3.11 ukazuje základní zapojení jednoho submodulu časovacího modulu TMR. Stejná struktura se v modulu TMR vyskytuje 4x.



Obr. 3.11: TMR submodul – blokový diagram

3.10. QSCI – Qued Serial Communication Interface

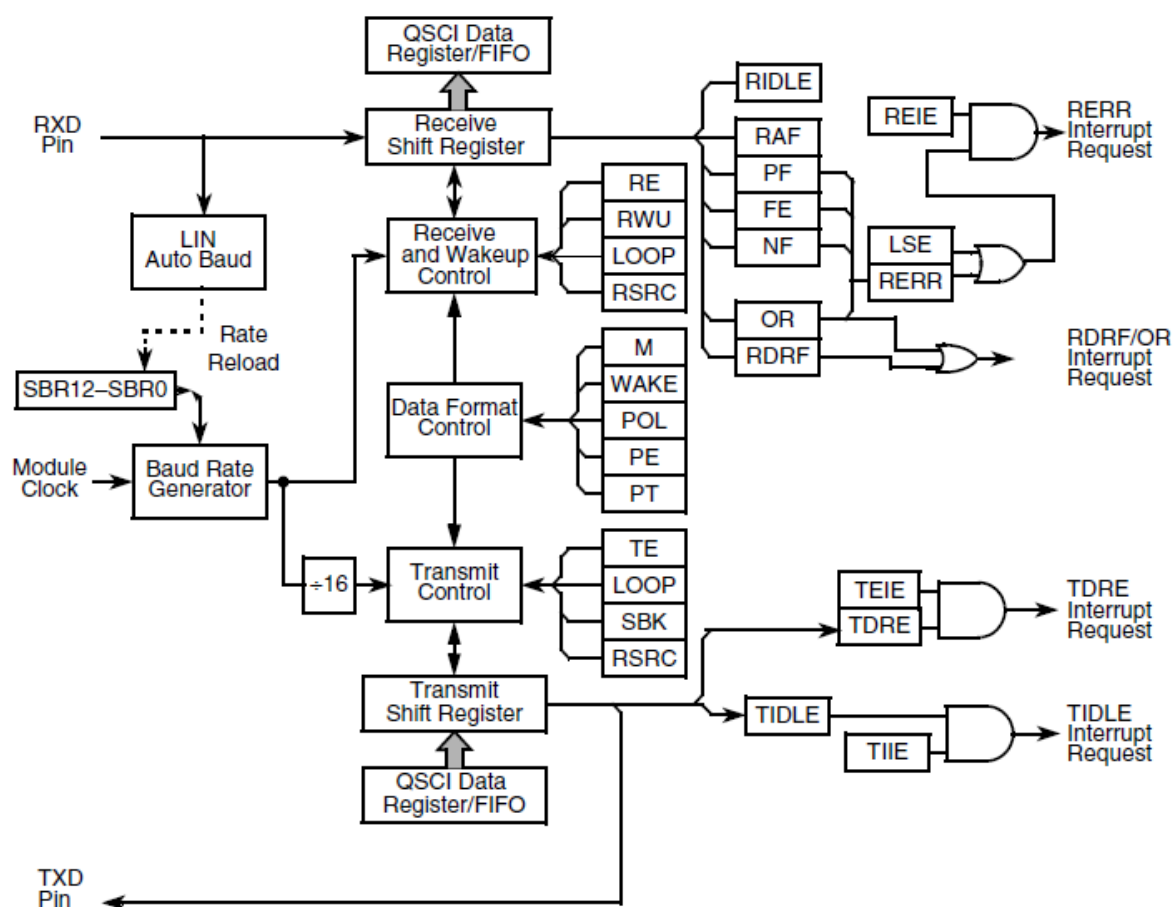
Tento modul umožňuje asynchronní sériovou komunikaci s periferními zařízeními a jinými mikropočítači.

3.10.1. QSCI – vlastnosti

- Full-duplex nebo single-wire komunikace
- Standardní formát – NRZ (Non-Return-to-Zero)
- Volba přenosové rychlosti - baud rate - 13-bitový integer a 3-bitový fractional
- Programovatelný formát dat – 8- nebo 9-bitový formát
- Separátně povolovaný vysílač a přijímač
- Separátně generovaný požadavek na přerušení od vysílání a přijímání
- Programovatelná polarita pro vysílač a přijímač
- Dvě metody pro probuzení přijímače

- Idle line
- Address mark
- Sedm příznaků generujících přerušení
 - Transmitter empty
 - Transmitter idle
 - Receiver full
 - Receiver overrun
 - Noise error
 - Framing error
 - Parity error
 - LIN sync error
- Detekce – receiver framing error
- Hardwarová kontrola parity

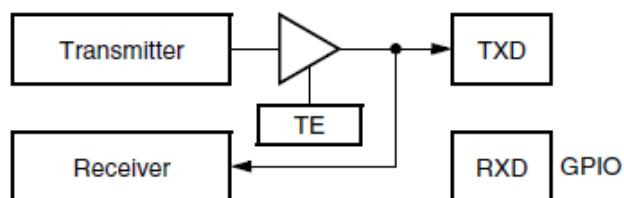
3.10.2. QSCI – blokový diagram



Obr. 3.12: QSCI modul – blokový diagram

3.10.3. QSCI – single-wire komunikace

Za normálních okolností QSCI využívá pro komunikaci dvou pinů, jeden pro příjem (receiving) a jeden pro vysílání (transmitting). V režimu single-wire, pin RxD je odpojen od QSCI modulu a může být použit jako běžný GPIO pin. QSCI modul pak využívá TxD pin jak pro příjem, tak pro vysílání.



Obr. 3.13: QSCI modul – single-wire komunikace

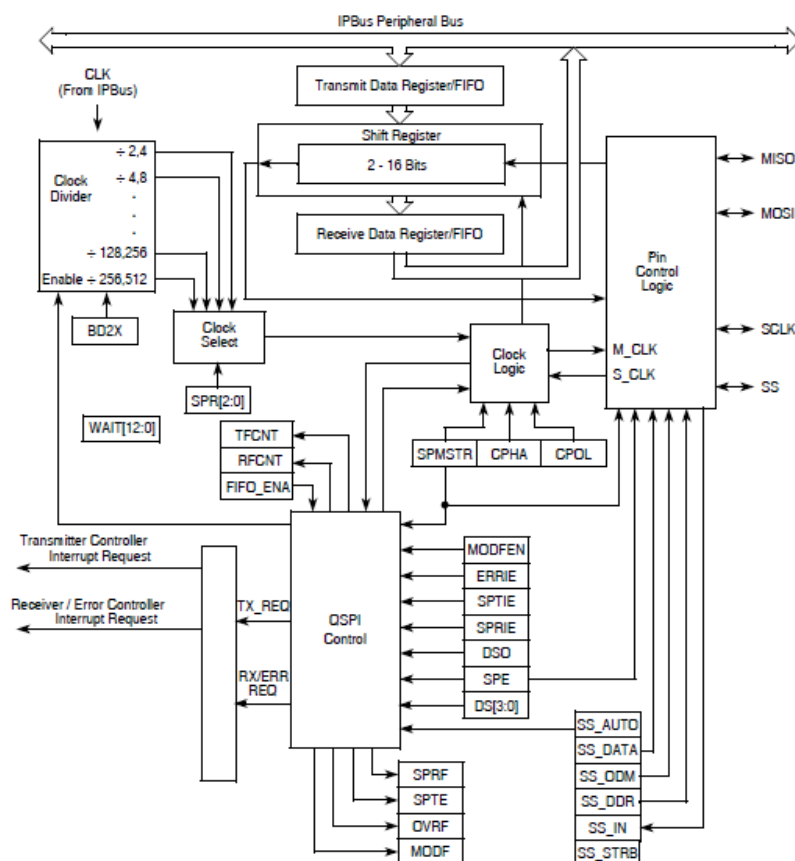
3.11. QSPI – Qued Serial Peripheral Interface

QSPI modul umožňuje full-duplex synchronní sériovou komunikaci mezi periferními zařízeními a jinými mikropočítači.

3.11.1. *QSPI – vlastnosti*

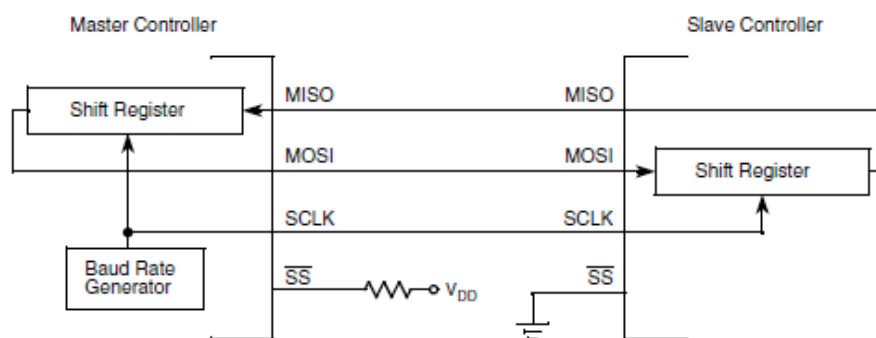
- Full-duplex
- Master a slave módy
- Double-buffered funkčnost separátně pro vysílací a přijímací registry
- FIFO se čtyřmi úrovněmi pro vysílací a přijímací mezipaměť
- Programovatelná délka přenosu (2 až 16-bitů)
- Maximální frekvence ve slave módu – $\text{frequency} < \text{bus frequency} / 2$
- Programovatelná polarita a fáze sériových hodin
- Dvě oddělená přerušení
 - QSPI receiver full/error
 - SPTE (QSPI transmitter empty)
- Wired OR mód umožňuje propojení několika QSPI modulů

3.11.2. QSPI – blokový diagram



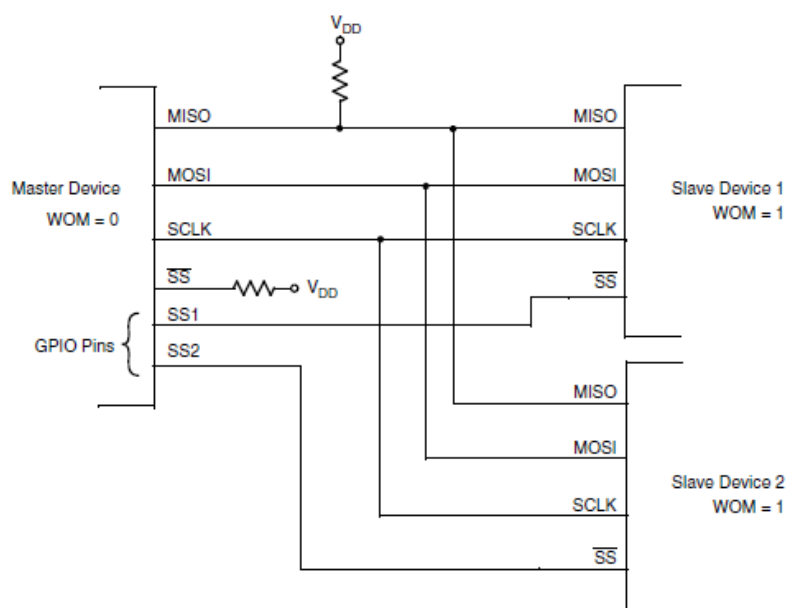
Obr. 3.14: QSPI modul – blokový diagram

3.11.3. QSPI – master mód



Obr. 3.15: QSPI modul – full-duplex master/slave zapojení

3.11.4. QSPI – wired-OR mód



Obr. 3.16: Propojení QSPI modulů - Master a dva slaves

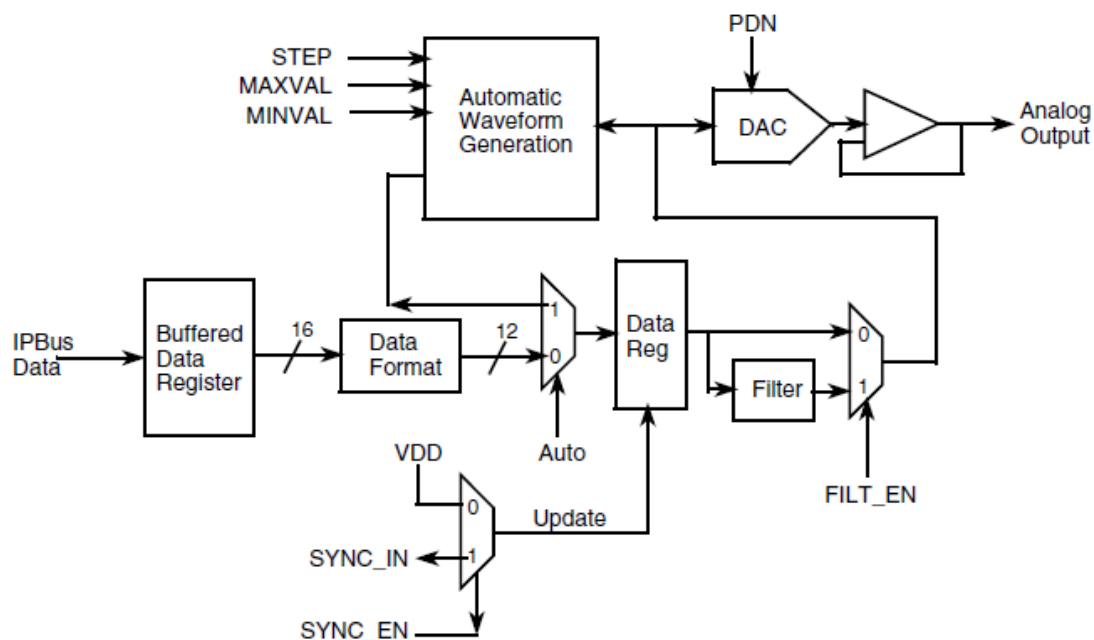
3.12. DAC – Digital-to-Analog Converter

DAC modul zpracovává 12-bitový signál a generuje analogové napětí v rozsahu VSSA a VDDA. DAC se skládá ze samotného převodníku, výstupního zesilovače a řídicích bloků. DAC výstup může být využit interně jako vstup do periférií mikropočítače jako jsou analogový komparátor a v závislosti na implementaci čipu může být rovněž přiveden na výstupní pin.

3.12.1. DAC – vlastnosti

- 12-bitové rozlišení
- Doba převodu - 2 μ s
- Mód výpadku napájení
- DAC výstup může být volitelně propojen s komparátorem, ADC modulem, případně na externí pin
- DAC může pracovat do definované impedance– 3 k Ω a 400 pF
- Automatický mód – DAC generuje vlastní časové průběhy - obdélník, trojúhelník a pilu

3.12.2. DAC – blokový diagram



Obr. 3.17: DAC modul – blokový diagram

3.13. CMP – Comparator

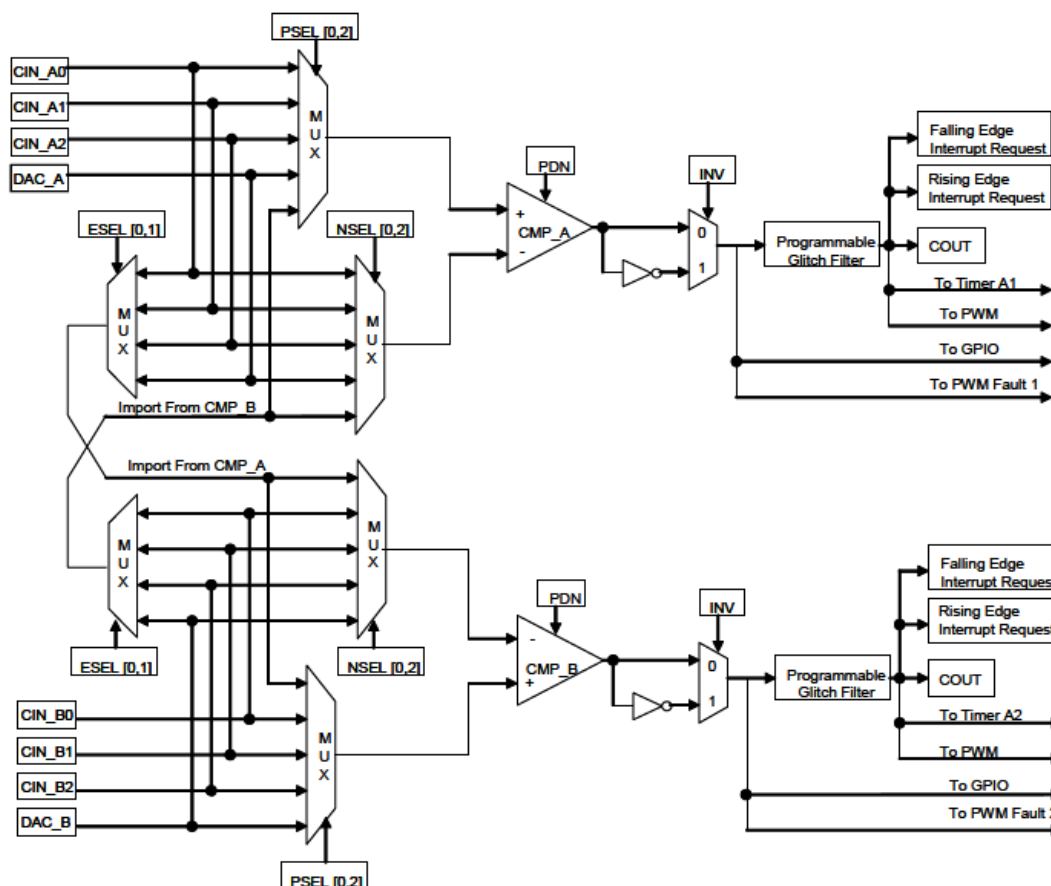
Komparátor se skládá analogového modulu komparátoru a řídicího digitálního rozhraní.

3.13.1. CMP – vlastnosti

- Jedná se o diferenciální analogový komparátor
- Vstupní přepínací matice umožňuje nezávislé propojení analogových vstupů na vstupy + a – komparátoru
- Vstupy jednotlivých komparátorů mohou být – tři GPIO vstupy (CIN1, CIN2 a CIN3), DAC výstup a vstup z jiného komparátoru
- Komparátor umožňuje řízení výstupní polarity (volitelná inverze)
- Programovatelný filtr typu dolní propusti
- Mód pro šetření energií
- Detekce sestupné a nástupné hrany s volitelným přerušením na dané hrany

- Výstup může být použitý pro řízení vstupů čítače, PWM chybové logiky, PWM řízení. Dále lze propojit výstup z komparátoru na pin nebo ho použít jako zdroj pro přerušení.

3.13.2. CMP – blokový diagram



Obr. 3.18: CMP modul – blokový diagram

3.14. GPIO – General Purpose Input/Output

GPIO modul umožňuje přímé čtení nebo zápis hodnoty/stavu pinu. GPIO piny mohou být multiplexované s ostatními perifériemi mikropočítače. GPIO můžeme konfigurovat třemi způsoby:

- GPIO input – s připojeným nebo odpojeným interním pull-up rezistorem
- GPIO output – push-pull mód nebo open drain mód

- Jako periferní pin v případě, že je pin multiplexovaný s jiným interním modulem

GPIO jsou na čipu organizovány do skupin, nazývají se porty a jsou označeny písmeny A, B, C atd.

3.14.1. *GPIO – vlastnosti*

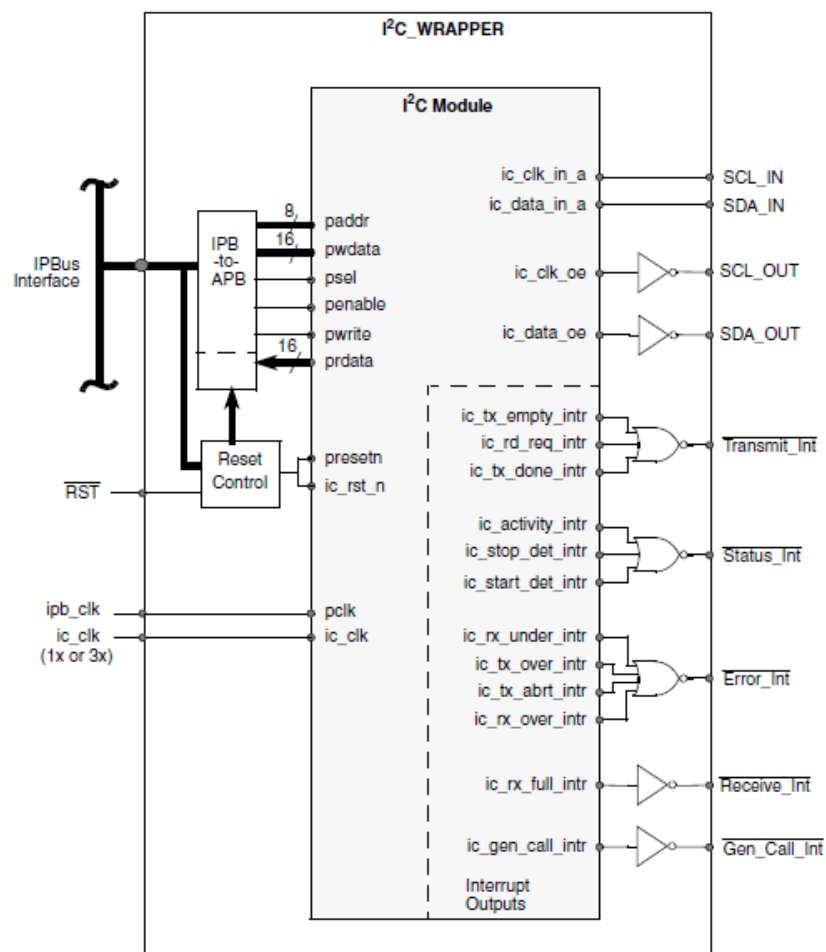
- Individuální konfigurace pinu – buď mód periferie nebo GPIO
- Konfigurace směru pinu v GPIO módu – input nebo output
- Volitelné připojení pull-up rezistoru – konfigurace pinu jako input a to v módu periferie nebo GPIO
- Volba módu push-pull nebo open drain - konfigurace pinu jako output a to v módu periferie nebo GPIO
- Volitelný proud výstupního pinu – 4 mA nebo 8 mA
- Možnost monitorovat logický stav pinu i v případě, že GPIO funkčnost není povolena
- Generování přerušení
- GPIO input – 5 V tolerant

I²C modul může pracovat ve standardním režimu až do přenosové rychlosti 100 kHz nebo v režimu fast, kde lze dosáhnout přenosové rychlosti až 400 kb/s.

3.15.1. I²C – vlastnosti

- Dvouvodičové sériové rozhraní
- Dva rychlostní módy
 - Standard
 - Fast
- Synchronizace hodinami
- Master nebo slave
- Podporuje multi-master (nutná arbitrace sběrnice)
- 7- nebo 10-bitové adresování
- Ignoruje CBUS adresy (předchůdce I²C)
- Čtyři vysílací a čtyři přijímací mezipaměti
- Interrupt nebo Polled mód
- Číslicový filtr – SDA a SCL

3.15.2. I^2C – blokový diagram



Obr. 3.20: I^2C modul – blokový diagram

4. HARDWAROVÁ SYNCHRONIZACE PWM MODULU S ADC MODULEM

V celé řadě praktických aplikací dochází k požadavku předávání dat a vzájemné komunikaci mezi periferiemi mikropočítače. Tento problém lze v zásadě řešit dvěma způsoby:

a) Periferie komunikují pouze s jádrem mikropočítače prostřednictvím sběrnic a jádro se pak stará o vzájemnou propojitelnost modulů. Toto řešení vyžaduje časté přerušování chodu programu a ztrátu výpočetního výkonu obzvláště pokud se jedná o děje periodické, asynchronní a relativně rychlé.

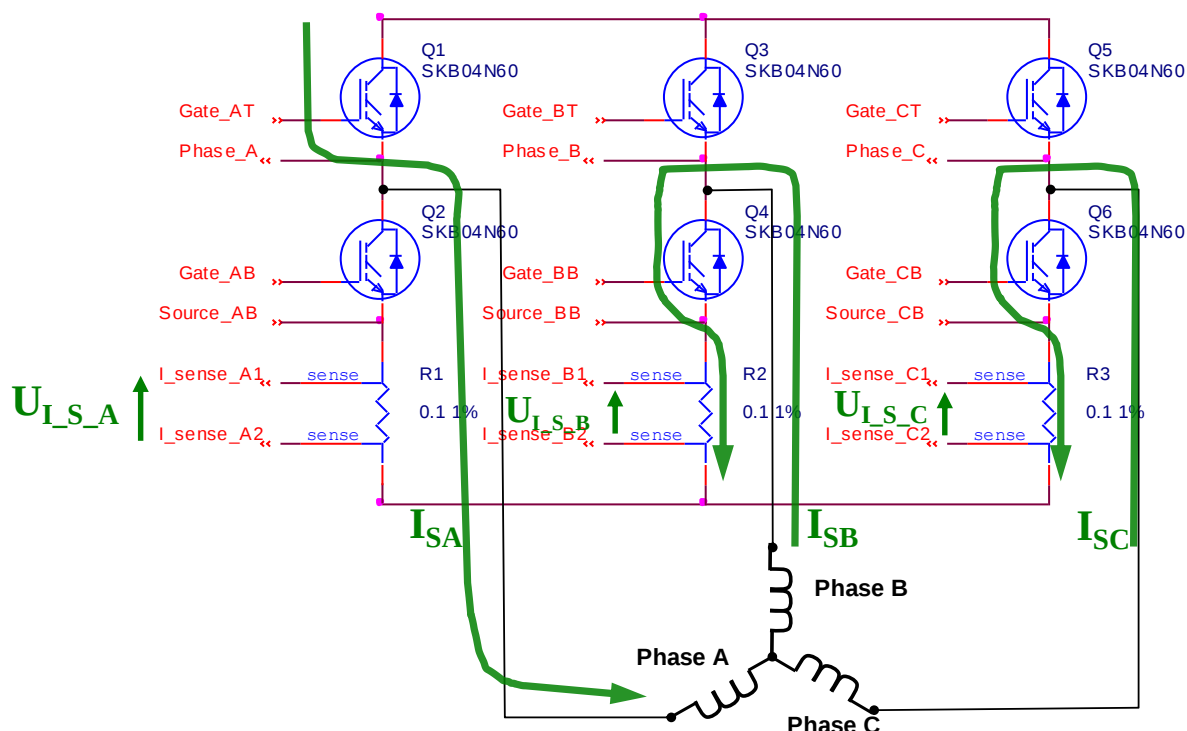
b) Periferie komunikují s jádrem prostřednictvím standardních sběrnic a zároveň moduly si mohou předávat informace formou signálů bez nutnosti zásahu jádra mikropočítače. V tomto případě, kdy moduly v nějaké míře spolupracují na úrovni signálů, dochází k uvolnění výpočetního výkonu jádra pro vykonávání "užitečného" kódu a tedy k větší propustnosti mikropočítačového systému.

V tomto příkladu si rozebereme možnosti synchronizace modulů PWM a ADC pomocí časovacího modulu, v našem případě Quad Timeru, bez intervence jádra.

4.1. PWM a ADC synchronizace – motivace

Typickým příkladem využití této vlastnosti je měření proudu tekoucího do motoru u měniče kmitočtu s měřením realizovaným pomocí snímacích odporů. Požadavek znalosti proudu je fundamentální pro většinu řídicích struktur motorů.

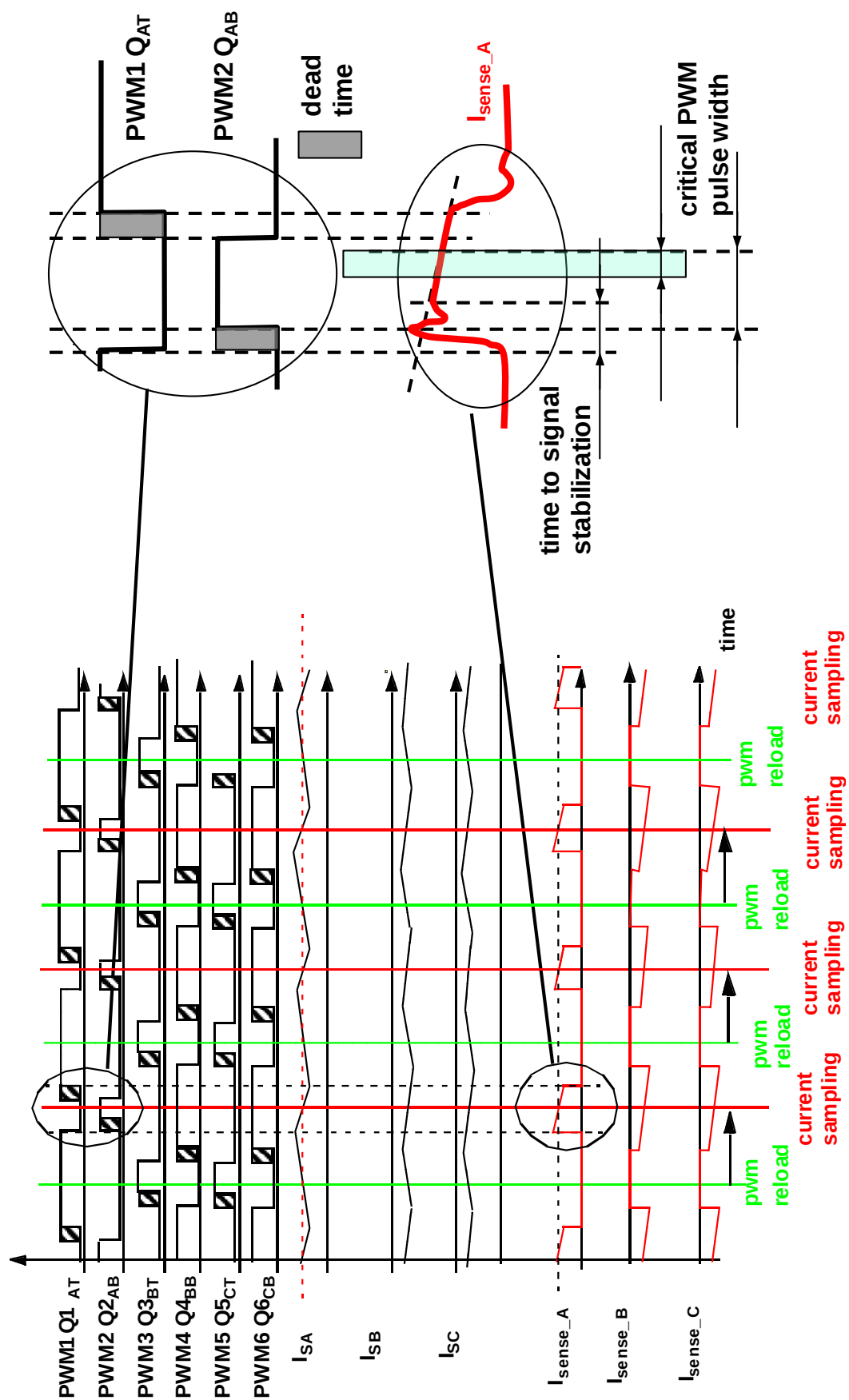
Na Obr. 4.1 je zobrazena topologie 3-fázového tranzistorového střídače, který se skládá ze tří větví. Každá větev obsahuje dva tranzistory: horní a dolní. V našem případě se jedná o tranzistory IGBT. V každé větvi se nachází jeden snímací odpor, který je umístěn pod dolním tranzistorem. Jednotlivé fáze vinutí motoru jsou připojeny do bodu mezi horním a dolním tranzistorem. Ze schématu je patrné, že pro řízení takového střídače je zapotřebí generovat šest signálů.



Obr. 4.1: Topologie střídače s měřením proudu pomocí snímacích odporů

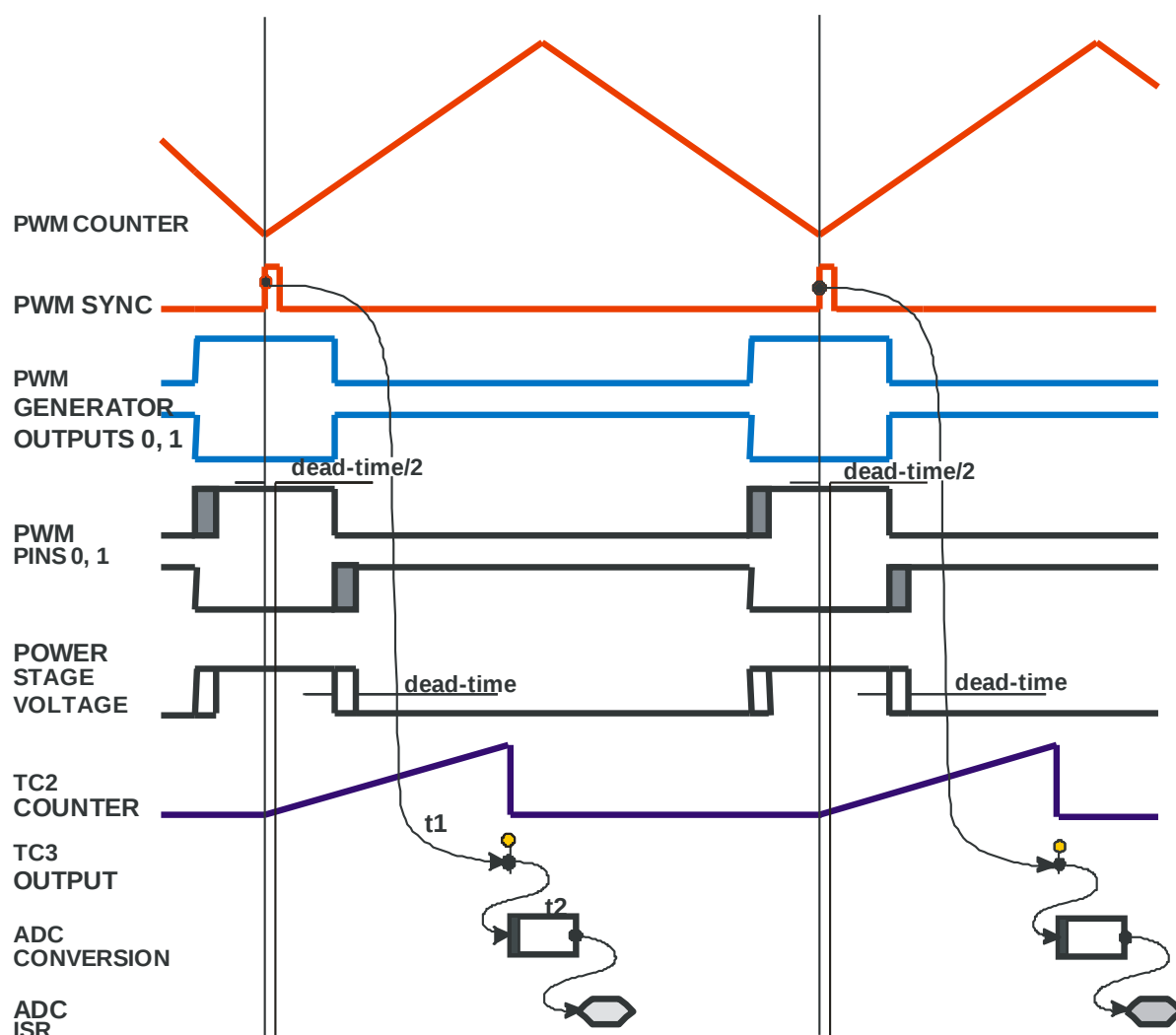
Zelené šipky s označením I_{SA} , I_{SB} a I_{SC} značí okamžité směry proudů, přičemž proudy tekoucí do motoru můžeme uvažovat za kladné a naopak proudy z motoru vytékající za záporné. Napětí U_{LS_A} , U_{LS_B} , U_{LS_C} s označením polarity úbytků definuje kladné napětí a tedy kladné směry proudů.

Z Obr. 4.1 je zřejmé, že proud je na snímacím odporu měřitelný pouze za předpokladu sepnutého dolního tranzistoru. Detailní průběhy signálů na střídači je možno vidět na Obr. 4.2. Detail obrázku zobrazuje tvar měřeného proudu včetně přechodového děje. Je zřejmé, že napětí na měřicím odporu není možno měřit v libovolném časovém okamžiku, ale jen přesně vymezených časových okamžicích, které souvisejí s generováním PWM signálu. Tato logická úvaha nás vede k tomu, že by bylo vhodné odvozovat okamžiky pro měření od PWM signálu.



Obr. 4.2: Průběhy signálů na střídači

4.2. PWM a ADC synchronizace – princip



Obr. 4.3: Průběhy signálů na střídači

Princip synchronizace PWM modulu a modulu ADC spočívá v generování vhodných signálů příslušnými periferiemi, které se na synchronizaci podílejí a v jejich vzájemném propojení. Na samotné synchronizaci se podílejí tři moduly: PWM, ADC a Quad Timer.

PWM modul, který generuje příslušné signály pro spínání tranzistorů střídače, generuje synchronizační signál (PWM_SYNC), který udává začátek PWM cyklu. Flexibilita PWM modulu umožňuje různé konfigurace generování synchronizačního signálu jako např.: na začátku PWM cyklu, na začátku násobku PWM cyklu, uprostřed PWM cyklu a různé kombinace zmíněných možností.

PWM synchronizační signál je interně v rámci čipu připojen do časovacího/čítacího modulu zvaného Quad Timer. Synchronizační signál lze časovacím/čítacím modulem libovolně zpracovávat. Při vhodné konfiguraci časovacího/čítacího modulu můžeme docílit generování signálu, který je hardwarově propojen s modulem ADC a dokáže automaticky spouštět konverzi převodníku. K tomuto účelu je vyčleněn jeden konkrétní časovač/čítač označený jako "TC3 OUTPUT".

ADC modul lze nakonfigurovat zcela libovolně podle požadavků aplikace a jen spouštění nakonfigurovat synchronně s PWM modulem. Časovací/čítací modul definuje přesný okamžik, kdy dojde ke startu ADC převodníku. Po dokončení převodu převodníků můžeme nakonfigurovat přerušení a vyčíst převedená data z ADC registrů a použít je v rámci vykonávaného algoritmu.

4.3. PWM a ADC synchronizace – postup konfigurace

4.3.1. *PWM modul*

GCT (Graphical Configuration Tool)

- PWM modul
 - Nastavení požadované funkčnosti PWM modulu (např.: PWM frekvence - 20kHz, komplementární spínání, center-aligned PWM generace, deadtime - 1us, pozitivní polarita signálu pro horní a dolní tranzistory, povolení vypnutí PWM signálu v případě detekce poruchy, atd.)
- SYS modul
 - Povolení hodinového signálu do PWM modulu
- GPIO_A
 - Nastavení pinů P0 až P5 z režimu GPIO do režimu PWM výstupů

Zdrojový kód

- Inicializace GPIO_A podle nastavení GCT - `ioctl(GPIO_A, GPIO_INIT, NULL);`
- Inicializace PWM modulu podle nastavení GCT - `ioctl(PWM, PWM_INIT, NULL);`

4.3.2. *Quad Timer modul*

GCT (Graphical Configuration Tool)

- QT_A3 - Quad Timer Channel3
 - Count mode -Triggered mode, edge of secondary source triggers primary count till compare
 - Primary source – Prescaler (IPB clock/1)
 - Secondary – Counter #3 input pin
 - Counter length – Count until compare and reinitialize
 - Disable tick box – Timer Channel Enable
 - Output mode – Set on compare, cleared secondary source input edge
 - Compare value 1 – definuje okamžik, kdy ADC vykoná sample & hold a zahájí převod
 - OFLAG / Timer Pin Control – zvolit "output enable (OFLAG to pin)" – toto nastavení umožňuje sledovat okamžik spouštění převodníku na pinu mikropočítače, vhodné pro ladění aplikace
- SYS modul
 - Peripheral Clock Enable (PCE) for TMR A3 - povolení hodinového signálu do časovacího/čítacího modulu
 - DAC and Timer Internal Peripheral Source Select (IPS) -> Timer A3 a v drop down list boxu zvolit - PWM reload sync. signal
- GPIO_B
 - Pin 3 – zvolit TA3

Zdrojový kód

- Inicializace GPIO_B podle nastavení GCT - `ioctl(GPIO_B, GPIO_INIT, NULL);` - umožní sledování okamžiku, kdy dochází ke startu ADC převodníku na pinu mikropočítače.
- Inicializace "Quad Timer channel3" podle nastavení GCT - `ioctl(QTIMER_A3, QT_INIT, NULL);` - nastavení funkčnosti časovače/čítače po resetu mikropočítače.

4.3.3. *ADC modul*

GCT (Graphical Configuration Tool)

- ADC - Analog-to-Digital Converter - ANA a ANB nastavení
 - Start-up mode – Normal

- Triggered source – SYNC0 input
- Power control – zakázat všechny volby
- Interrupts – zvolit "ANA Conversion complete", definovat "ISR name" a "priority"
- SYS modul
 - Peripheral Clock Enable (PCE) - zvolit ADC - povolení hodinového signálu do ADC modulu
- Zvolit INTC - Interrupt Controller

Zdrojový kód

- Inicializace ADC modulu podle nastavení GCT - `ioctl(ADC, ADC_INIT, NULL);`
- Definovat jméno obsluhy přerušení podle konvencí kompilátoru - `void ????(void)` - kde `????` značí jméno, které bylo nadefinováno v GCT pro modul ADC
 - Pozor!!!! Nezapomenout čistit EOSI flag uvnitř přerušovací rutiny - `ioctl(ADC, ADC_CLEAR_STATUS_EOSI, NULL);`

4.3.4. *Synchronizace - výsledek z osciloskopu*

Z Obr. 4.4 je zřejmé, jak vypadají skutečné signály generované mikropočítačem a jejich vzájemné souvislosti. Modrý a žlutý signál zobrazují generovaný PWM signály včetně nezbytného deadtimu, přičemž žlutý signál je spínací signál pro horní tranzistor jedné větve střídače a druhý signál modrý je jeho komplement a definuje spínání dolního tranzistoru téže větve.

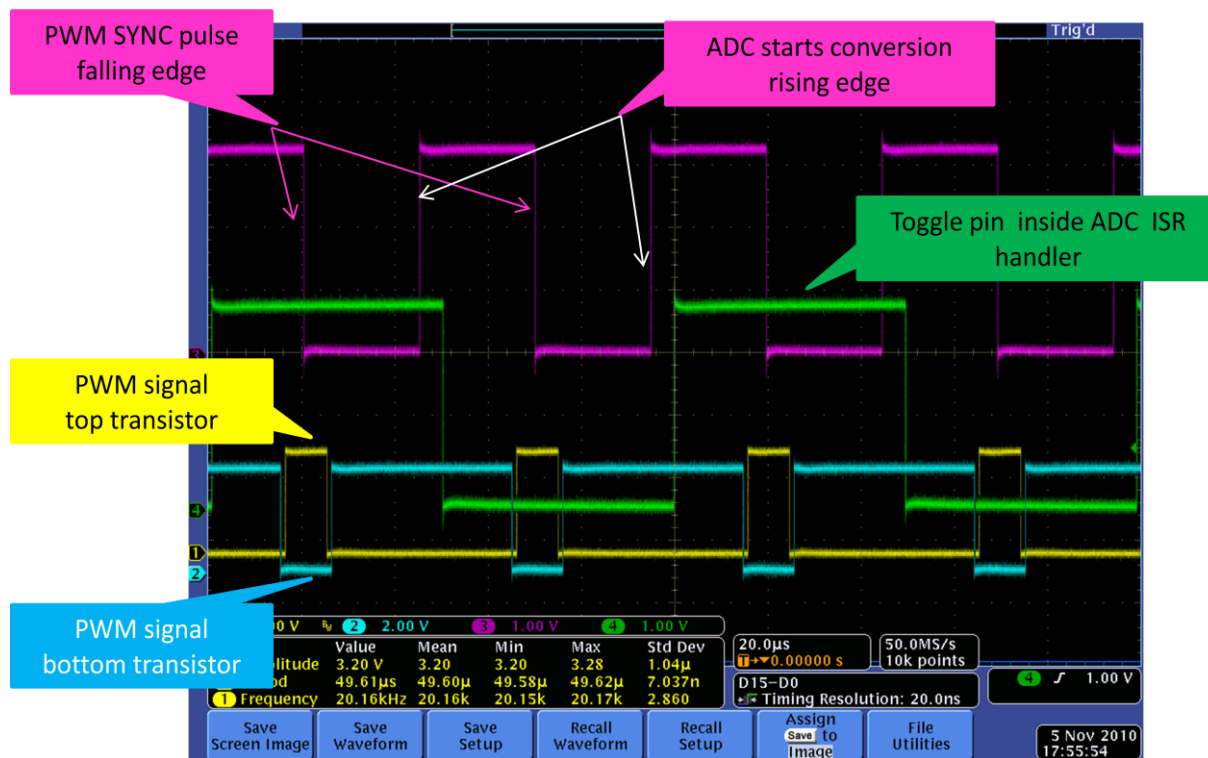
Fialový signál v sobě nese dvě informace:

- Sestupná hrana signálu je totožná se synchronizačním pulzem generovaným PWM modulem, v našem případě, vždy na začátku nového PWM cyklu
- Zatímco nástupná hrana signálu definuje okamžik, kdy dochází k iniciaci ADC převodníku - `sample & hold` a následný převod

Zelený průběh, respektive jeho hrany, definuje okamžik, kdy došlo k převedení všech signálů a přerušovací funkce začala zpracovávat převedené hodnoty z ADC převodníku. Časový rozdíl mezi nástupnou

hranou fialového signálu a zeleného signálu závisí na následujících parametrech, které můžeme ovlivnit nastavením:

- Počet převodů ADC převodníku od spouštěcího signálu - max. 16
- Typ obsluhy přerušení - standardní nebo rychlé.



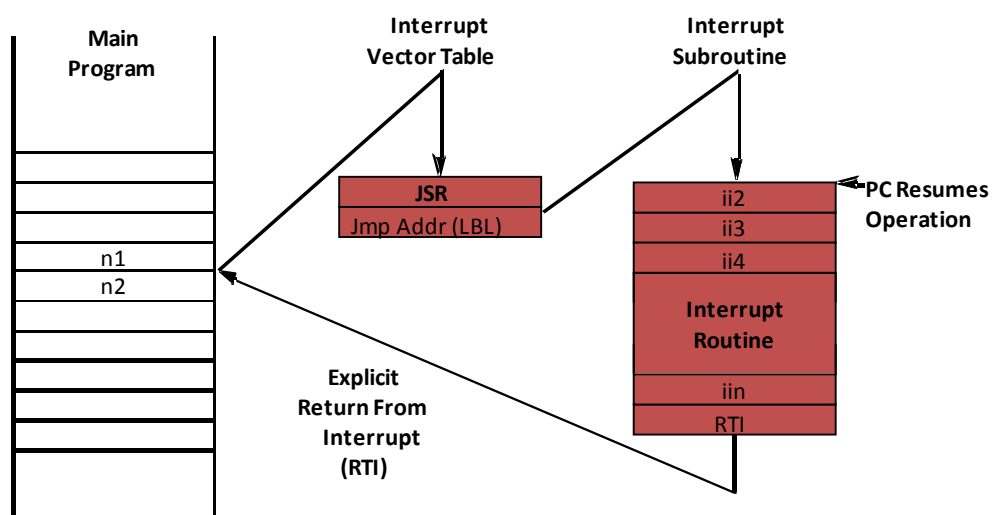
Obr. 4.4: Průběhy signálů na střídači

5. PŘERUŠENÍ - JEDEN ZE STAVEBNÍCH KAMENŮ EMBEDDED APLIKACÍ

Vhodný přerušovací systém mikropočítače výraznou měrou napomáhá vývojáři řešit problémy s efektivní obsluhou asynchronních a periodických událostí zpracovávaných aplikačním programem mikropočítače. Podpora zpracování asynchronních událostí ze strany mikropočítače ve spolupráci s podporou ze strany kompilátoru mohou hrát významnou roli při volbě vhodného mikropočítače pro danou embedded aplikaci.

5.1. Co je to přerušení

Přerušení je mechanismus mikropočítače umožňující přerušení běžícího programu nějakou událostí. Mikropočítač vykonává instrukce podle programu v paměti, když přijde nějaká asynchronní událost a to ať z vnějšího prostředí nebo z mikropočítače samotného. Daná událost, v případě vhodného nastavení, zastaví chod hlavního programu a začne zpracovávat obsluhu přerušení. Jakmile dojde k ukončení obsluhy přerušení, jádro předá řízení zpět hlavnímu programu a program pokračuje ve vykonávání jednotlivých instrukcí.



Obr. 5.1: Princip obsluhy přerušení

Zdrojů přerušení může být několik typů:

- Události iniciované externím hardware mimo samotný mikropočítač - např.: Interrupt request signal (vnější zdroj přerušení připojený na pin mikropočítače), měření pulsů inkrementálního snímače, timer input capture (zachycení události na připojené na pin mikropočítače)
- Události vyvolané periferními moduly mikropočítače - např.: ukončení převodu ADC převodníku, přetečení časovače, atd.
- Události zpracovávané v rámci jádra mikropočítače - např.: illegal instruction interrupt, misaligned data access interrupt, atd.
- Události generované software, přímo instrukcí mikropočítače - např.: SWI (Software Interrupt)

5.2. Princip zpracování přerušení

Mechanismus zpracování požadavku na přerušení se skládá z celé řady kroků. Některé záležitosti zpracovává mikropočítač bez "účasti" programátora. Podstatná část obsluhy přerušení však zůstává na bedrech programátora. Je tedy zapotřebí důkladně porozumět tomu, jak správně přerušení obsloužit, aby nedocházelo k náhodnému chování celého programu. Tato situace je obvykle velmi těžko odhalitelná.

Mechanismus obsluhy krok za krokem:

- Pokud se vyskytne událost generující požadavek na přerušení, hardware mikropočítače analyzuje daný požadavek (může se jednat o jádro nebo speciální modul zpracovávající přerušení).
- Pokud má mikropočítač více úrovní přerušení, přerušení s nevyšší prioritou se začne zpracovávat a další zpracování se předá obvykle do jádra mikropočítače.
- Pokud je mikropočítač nastaven tak, aby akceptoval přerušení, jádro akceptuje požadavek na přerušení.
- Jádro mikropočítače dokončí "atomickou instrukci", kterou právě vykonává.

- Jádru předá řízení do tabulky přerušovacích vektorů, kde se obvykle nachází skok do obslužné rutiny, tedy instrukce JSR + adresa obslužné rutiny.
- Nyní je nutné uchovat návratovou adresu programu, čili dojde k uložení PC (Program Counter) a důležitých nastavení mikropočítače, tedy SR (Status Register) na zásobník.
- Nyní jádro mikropočítače začne vykonávat obslužnou rutinu. Zde je zapotřebí dávat pozor, aby nedošlo k nenávratné ztrátě obsahu registrů, které přerušovací rutina používá. V závislosti na tom, zda obslužnou rutinu píšeme v jazyce C nebo přímo v assembleru, se musíme postarat o uchování registrů po dobu vykonávání přerušovací rutiny.
 - V jazyce C používáme #pragma interrupt, případně spoustu dalších nastavení závisících na implementaci kompilátoru.
 - V assembleru se musí o registry postarat programátor.
- Na konci přerušovací rutiny se musí nacházet instrukce RTI (Return From Interrupt) namísto standardní instrukce RTS (Return From Subroutine). **Nesprávně ukončená obslužná rutina je častá chyba vedoucí ke zhroucení programu.**
 - V případě implementace v jazyce C se o správné ukončení rutiny postará kompilátor.
 - V případě implementace v assembleru se stará programátor.
- Pak dojde k obnovení použitých registrů, PC a SR.
- Jádru pokračuje v místě programu, kde došlo k požadavku na přerušení.

5.3. MC56F8025 - podpora přerušení z hlediska mikropočítače

Jádro tohoto procesoru 56800E podporuje 5 úrovní přerušení. Status registr obsahuje dva bity IO, I1, které definují úrovně přerušení - CCPL (Current Core Interrupt Priority Level) podle tabulky Tab. 5.1

IPL	Description	Priority	Interrupt Sources
LP	Maskable	Lowest	SWILP Instruction
0	Maskable	*	On-chip peripherals, IRQA and IRQB, SWI #0 Instruction
1	Maskable	*	On-chip peripherals, IRQA and IRQB, SWI #1 Instruction
2	Maskable	*	On-chip peripherals, IRQA and IRQB, SWI #2 Instruction
3	Non-maskable	Highest	Illegal instruction, hardware stack overflow, SWI instruction, EOnCE Interrupts, misaligned data access

Tab. 5.1: Úrovně přerušení

Příchozí přerušení je akceptované, pokud je jeho priorita vyšší než priorita přerušení, které se právě vykonává. Situace je přehledně znázorněna v následující tabulce Tab. 5.2.

I1	IO	CCPL	Exceptions Accepted	Exceptions Masked	Comments
0	0	0	IPL 0,1,2,3, and SWILP	None	All interrupts are allowed including the SWILP
0	1	1	IPL 1,2,3	IPL 0 and SWILP	All non-maskable interrupts and maskable interrupts that are programmed at level 1 or 2 are allowed
1	0	2	IPL 2,3	IPL 0, 1 and SWILP	All non-maskable interrupts and maskable interrupts that are programmed at level 1 are allowed
1	1	3	IPL 3	IPL 0, 1, 2 and SWILP	Only non-maskable interrupts are allowed

Tab. 5.2: Logika zpracování přerušení podle úrovní

Modul "Interrupt Controller" může zpracovat až 64 zdrojů přerušení. Jednotlivým zdrojům přerušení lze přidělit jejich prioritu. Pokud by došlo k současnému požadavku na obsluhu dvou nezávislých zdrojů přerušení na stejné úrovni, pak se pořadí jejich vykonávání bude řídit číslem daného

zdroje přerušení, přičemž přerušovací vektor s číslem 0 má nejvyšší prioritu a přerušovací vektor s číslem 63 má prioritu nejnižší.

5.4. MC56F8025 - Fast Interrupt

Interrupt Controller podporuje tzv. Fast Interrupt. Fast Interrupt slouží k rychlé obsluze požadavku na přerušení, kde není zapotřebí vykonávat dlouhý kód. K tomuto účelu slouží sada speciálních registrů prodporující rychlou obsluhu přerušení.

Podmínky pro vykonání Fast Interruptu:

- Fast Interrupt musí mít prioritu level 2.
- Fast Interrupt nepoužívá tabulku vektorů přerušení.
- Adresa obslužné rutiny přerušení musí být zapsána do dvojice registrů FIVAL0 & FIVALH0 a FIVAL1 & FIVALH1.
- Registry FIM0 a FIM1 obsahují číslo přerušení a definují tím, které přerušení bude zpracováno jako Fast Interrupt

5.5. MC56F8025 - Fast Interrupt - zpracování

Mechanismus zpracování Fast Interruptu pracuje následovně:

- V okamžiku požadavku na přerušení dojde ke zmrazení programového čítače PC (Program Counter), který je zkopírován do FIRA (Fast Interrupt Return Address) registru.
- Status register s výjimkou bitů P4-P0 a NL bitu z operating mode registru se uloží do FISR (Fast Interrupt Status Registr) registru.
- Stack pointer (SP) je zarovnaný na 32-bitový přístup.
- Registr Y se uloží na Stack.
- V případě registrů R0, R1, N a M01 dojde k jejich svapnutí a během vykonávání Fast ISR se používají stínové registry.

- Po ukončení přerušovací rutiny dojde k rekonstrukci původního stavu a program pokračuje ve vykonávání naprogramovaných instrukcí z bodu, kde došlo k přerušení programu.

Pro ukončení přerušovací rutiny psané v assembleru se používá instrukce FRTID (Delayed Return from Fast Interrupt).

SEZNAM POUŽITÉ LITERATURY

- [1] MC56F80XXRM.pdf - 56F802X and 56F803X Peripheral Reference Manual
- [2] MC56F8025.pdf - 56F8035/56F8025 Data Sheet
- [3] DSP56800ERM.pdf - DSP56800E Reference Manual
- [4] DSP56800E_Quick_Start_Users_Manual.pdf – DSP56800E_Quick_Start User's Manual, Targeting Freescale 56F8xxx Platform
- [5] 56800E_aclib.pdf – Advanced Control Library for 56800E, User Reference Manual
- [6] 56800E_mclib.pdf – Motor Control Library, User Reference Manual
- [7] 56800E_gflib.pdf – General Functions Library, User Reference Manual
- [8] 56800E_gdflib.pdf – General Digital Filters Library for 56800E, User Reference Manual
- [9] Targeting_56800E_cd.pdf – CodeWarrior™ Development Studio for Freescale™ 56800/E Digital Signal Controllers: MC56F8xxx/DSP5685x Family Targeting Manual
- [10] 56800x_Build_Tools_Reference.pdf – CodeWarrior™ Build Tools Reference for Freescale™ 56800/E Digital Signal Controllers

Centrum pro rozvoj výzkumu pokročilých řídicích a senzorických technologií
CZ.1.07/2.3.00/09.0031

Ústav automatizace a měřicí techniky
VUT v Brně
Kolejní 2906/4
612 00 Brno
Česká Republika

<http://www.crr.vutbr.cz>

info@crr.vutbr.cz