

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

---

# Mikroprocesorová technika v aplikacích řízení elektrických pohonů

Učební texty ke kurzu

---

Autoři:

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

Ing. Pavel Grasblum, PhD (Freescale Semiconductor, Rožnov p. R.)

Datum:

2. 6. – 3. 6. 2011

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 do teorie elektrických pohonů .....	5
1.1. Základní charakteristiky elektrických pohonů - třídění .....	5
1.2. Mechanická charakteristika pracovních mechanismů .....	7
1.3. Pracovní režimy pohonů .....	8
1.4. Pohon pracující ve čtyřech kvadrantech .....	9
1.5. Unipolární vs. bipolární spínání .....	10
1.5.1. Princip unipolárního spínání .....	11
1.5.2. Princip bipolárního spínání .....	12
1.6. Nezávislé vs. komplementární spínání .....	13
1.6.1. Princip nezávislého spínání .....	13
1.6.2. Princip komplementárního spínání .....	14
2. BLDC motor .....	15
2.1. BLDC motor - princip .....	15
2.1. Invertor pro řízení BLDC motoru .....	16
2.1. Six step komutace .....	16
2.1. BLDC komutace na základě Hallových sensorů .....	17
3. Embedded mikropočítač Freescale MC56F8006 .....	20
3.1. Rodina mikropočítačů Freescale MC56F800x .....	20
3.2. DSC56F8006/56F8002 – základní parametry .....	20
3.3. Organizace Freescale dokumentace .....	22
3.4. ADC – Analog Digital Converter .....	23
3.4.1. ADC - vlastnosti .....	23
3.4.2. ADC – blokový diagram .....	24
3.5. PDB – Programmable Delay Block .....	25

3.5.1.	PDB - vlastnosti .....	25
3.5.2.	PDB – blokový diagram .....	25
3.1.	PGA – Programmable Gain Amplifier .....	25
3.1.1.	PGA - vlastnosti.....	26
3.1.2.	PGA – blokový diagram .....	27
3.2.	COP – Computer Operating Properly .....	27
3.2.1.	COP – vlastnosti .....	27
3.2.2.	COP – blokový diagram .....	28
3.3.	PWM – Pulse Width Modulator .....	28
3.3.1.	PWM – vlastnosti .....	29
3.3.2.	PWM – blokový diagram .....	30
3.1.	OCCS – On-Chip Clock Synthesis.....	30
3.1.1.	OCCS – vlastnosti .....	30
3.1.2.	OCCS – blokový diagram .....	32
3.2.	FM – Flash Memory .....	32
3.2.1.	FM – vlastnosti .....	32
3.2.2.	FM – blokový diagram .....	33
3.3.	PIT – Programmable Interval Timer.....	34
3.3.1.	PIT – vlastnosti.....	34
3.3.2.	PIT – Blokový diagram.....	34
3.1.	DTMR – Dual Timer .....	34
3.1.1.	DTMR – vlastnosti.....	35
3.1.2.	DTMR – blokový diagram .....	36
3.2.	SCI – Serial Communication Interface .....	36
3.2.1.	SCI – vlastnosti.....	37
3.2.2.	SCI – blokový diagram.....	38
3.3.	SPI – Serial Peripheral Interface .....	38
3.3.1.	SPI – vlastnosti.....	38

3.3.2.	SPI – blokový diagram .....	39
3.3.3.	SPI – master mód .....	40
3.3.4.	SPI – wired-OR mód .....	40
3.4.	HSCMP – High Speed Comparator .....	40
3.4.1.	HSCMP – vlastnosti .....	40
3.4.2.	HSCMP – blokový diagram .....	41
3.5.	GPIO – General Purpose Input/Output .....	41
3.5.1.	GPIO – vlastnosti .....	42
3.5.2.	GPIO – blokový diagram .....	43
3.6.	I <sup>2</sup> C – Inter-Integrated Circuit Interface .....	43
3.6.1.	I <sup>2</sup> C – vlastnosti .....	44
3.6.2.	I <sup>2</sup> C – blokový diagram .....	45
4.	<i>Quick Start Tool A Graphical Configuration Tool – vývojové prostředí .....</i>	<i>46</i>
4.1.	Quick Start Tool – vlastnosti .....	46
4.1.1.	Infrastruktura systému Quick_Start .....	46
4.1.2.	On-chip drivers .....	47
4.1.3.	Vzorové aplikace .....	47
4.1.4.	GCT – Graphical Configuration Tool .....	47
4.2.	Quick Start Tool – integrace do prostředí Metrowerks CodeWarrior ..	48
4.3.	Spolupráce mezi GCT a nástrojem Quick Start .....	49
4.4.	ArchIO struktura .....	50
4.5.	ioctl command – Input Output Control .....	51
5.	FreeMASTER Tool – vývojové prostředí .....	53
5.1.	FreeMASTER tool – vlastnosti .....	53
5.1.1.	FreeMASTER tool – monitor v reálném čase .....	53
5.1.2.	FreeMASTER tool – grafický řídicí panel .....	56
6.	Freescale library .....	58
6.1.	Freescale library – MCLIB .....	58

6.2.	Freescale library – GFLIB .....	59
6.3.	Freescale library – GDFLIB.....	59
6.4.	Freescale library – ACLIB .....	59
Seznam použité literatury .....		61

## 1. ÚVOD DO TEORIE ELEKTRICKÝCH POHONŮ

*S rozvojem výkonových polovodičových prvků a mikropočítačů dochází k velkému rozmachu elektrických pohonů jako elektromechanických akčních členů. V minulosti používané stejnosměrné pohony jsou masivně nahrazovány pohony střídavými. Různé druhy elektrických pohonů můžeme v současnosti najít prakticky ve všech oblastech lidské činnosti jako např.: v automobilový průmysl (hybridní pohony, brzdový systém, EPS, starter-generátor, stahování oken, ventilátory, nastavování sedadel, atd.), průmyslové servopohony, bílá technika (pračky, myčky, sušičky, ledničky, kompresory, atd.), spotřební elektronika, modelářství, lékařská technika a další.*

---

### 1.1. Základní charakteristiky elektrických pohonů - třídění

Úkolem elektrického pohonu je uvést poháněný pracovní mechanismus do určitého stavu způsobem, který je definován pracovním mechanismem za účelem splnění požadované technologické operace. Elektrické pohony lze třídit podle celé řady různých hledisek. Některá základní hlediska jsou uvedena níže:

- Podle typu motoru z hlediska napájecího napětí
  - Stejnosměrný
  - Střídavý
- Podle typu motoru z hlediska konstrukce
  - Stejnosměrný s cizím buzením, sériovým buzením, kompaundní
  - ACIM
  - BLDC
  - PMSM
  - Krokový
  - SR

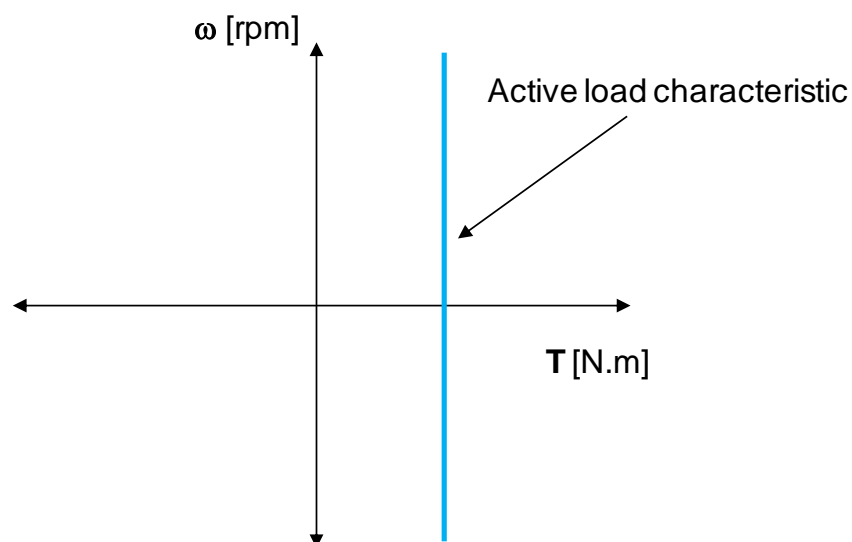
- Podle druhu pohybu, kterým se přenáší mechanická energie elektrického pohonu na pracovní mechanismus
  - Točivý nebo přímočarý (lineární) pohon
  - Spojitý nebo nespojitý pohon (nespojitý pohon např.: krokový, kmitavý, atd.)
- Podle stupně řiditelnosti pohonu
  - Jednorychlostní, vícerychlostní, s plynulým řízením rychlosti
  - Jednokvadrantový pohon - pracuje v jednom kvadrantu roviny  $\omega$ ,  $M$ , což znamená jeden smysl rychlosti a momentu, není určen pro elektrické brzdění
  - Dvoukvadrantový pohon pracující v 1. a 3. kvadrantu - pracuje ve dvou specifických kvadrantech  $\omega$ ,  $M$ , a to v 1. a 3., což znamená oba smysly rychlosti a momentu, rovněž není určen pro elektrické brzdění
  - Dvoukvadrantový pohon pracující v 1. a 2., respektive 3. a 4. kvadrantu - pohon pro oba smysly momentu či síly při stejném smyslu rychlosti, umožňuje elektrické brzdění
  - Čtyřkvadrantový pohon pracující ve všech 4 kvadrantech někdy nazývaný jako reverzační pohon pro oba smysly rychlosti, momentu či síly umožňující reverzaci pohonu za chodu, umožňuje elektrické brzdění
- Podle způsobu přenosu mechanické energie na pracovní mechanismus
  - Pohon s převodem nebo bez převodu
  - Pohon se spojkou (spojka tuhá, pružná, výsuvná, rozběhová, řiditelná, volnoběžná, atd.) nebo bez spojky
- Podle druhu poháněného pracovního mechanismu
  - Jeřáby, výtahy, dopravníky, lanovky, těžní stroje, atd.
  - Lokomotivy, trolejbusy, tramvaje, elektromobily, atd.



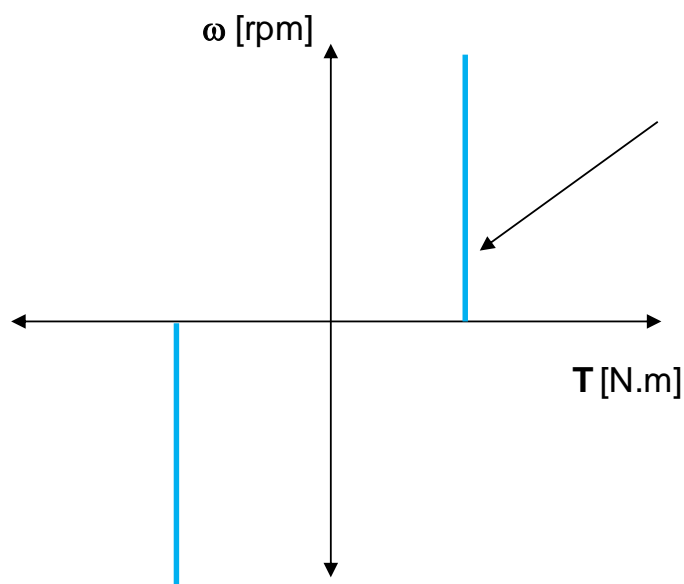
- Obráběcí stroje
- Hutní a válcovací stroje
- Bílá technika a spotřební elektronika - pračky, myčky, sušičky, kompresory, ventilátory, ledničky, atd.
- A spousta jiných
- Podle počtu motorů
  - Jednomotorové
  - Vícemotorové
- Podle funkce pohonu
  - Hlavní pohon
  - Pomocný pohon pohánějící pomocná zařízení hlavního pohonu
  - Vedlejší pohon zajišťující správnou funkci hlavního pohonu
  - Nouzový pohon
- Atd.

## 1.2. Mechanická charakteristika pracovních mechanismů

Mechanická charakteristika vyjadřuje závislost zatěžovacího momentu, kterým mechanismus zatěžuje motor. Udává se buď analyticky, nebo graficky. Zatěžovací momenty rozdělujeme na pasivní a aktivní. Aktivní zatěžovací moment mají mechanismy, u nichž dochází ke změnám potenciální energie. Pro aktivní zatěžovací moment je charakteristické, že při změně smyslu rychlosti nemění zatěžovací moment smysl. Pasivní zatěžovací moment generují mechanismy, v nichž dochází ke tření, obrábění, překonávání odporu vzduchu a jiného prostředí. Pasivní zatěžovací moment mění smysl momentu při změně smyslu otáčení.



Obrázek 1.1 Mechanická charakteristika mechanismu s aktivním zatěžovacím momentem



Obrázek 1.2 Mechanická charakteristika mechanismu s pasivním zatěžovacím momentem

### 1.3. Pracovní režimy pohonů

Dynamiku chování mechanické části pohonu definuje rovnice (1)

$$\frac{d\omega}{dt} = \frac{1}{J} [T_m(\omega) - T_L(\omega)] \quad (1)$$

Kde:

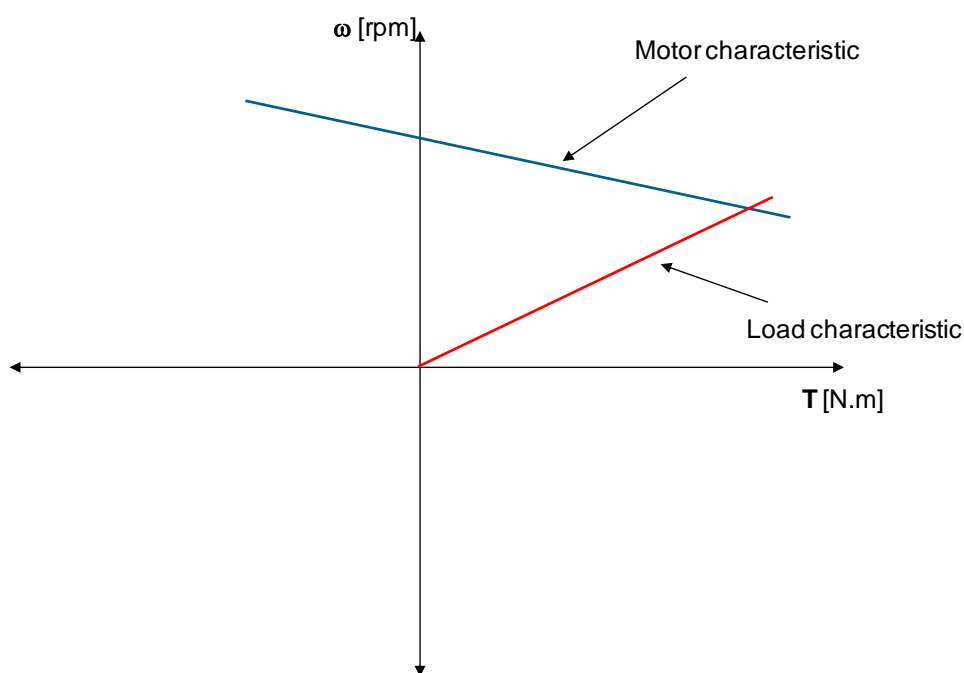
$J$  - moment setrvačnosti [kg.m<sup>2</sup>]

$\omega$  - úhlová rychlost [rad/s]

$T_m(\omega)$  - moment motoru [N.m]

$T_L(\omega)$  - zatěžovací moment [N.m]

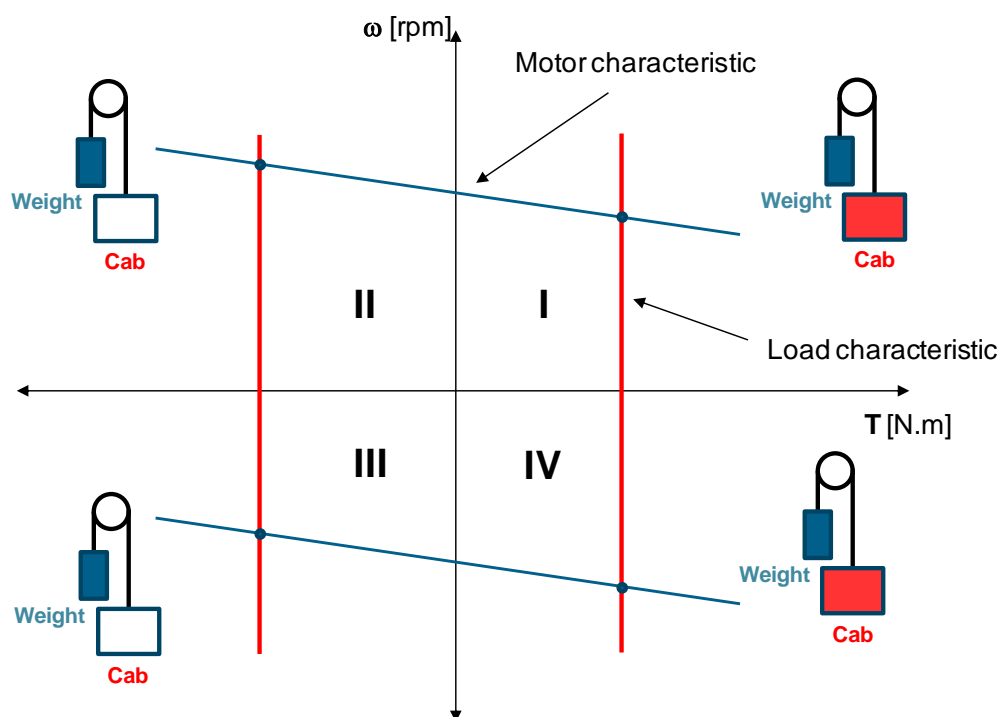
Elektrický pohon je v ustáleném stavu, pokud je moment motoru roven zatěžovacímu momentu. Pokud znázorňujeme graficky moment motoru a zatěžovací moment, pak je ustálený stav charakterizován průsečíkem obou charakteristik, jak ukazuje Obrázek 1.3.



**Obrázek 1.3** Mechanická charakteristika pohonu

## 1.4. Pohon pracující ve čtyřech kvadrantech

Typickým případem takového pohonu je zdvihací mechanismus s vyváženou hmotností klece a poloviny zátěže protizávažím (výtah). Grafické znázornění funkce pohonu, charakteristiky motoru a zatěžovacího mechanismu ukazuje Obrázek 1.4.

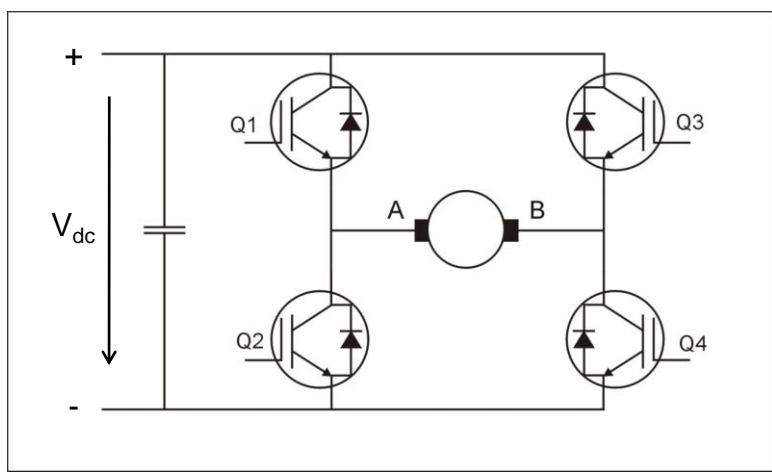


Obrázek 1.4 Pracovní režimy zdvihacího mechanismu (výtah)

Pracovní režimy v I a III kvadrantu odpovídají motorickému chodu pro zdvih s plnou zátěží a spouštění prázdné klece. Pracovní režimy ve II a IV kvadrantu odpovídají generátorickému chodu při zdvihu prázdné klece a spouštění plné klece.

## 1.5. Unipolární vs. bipolární spínání

Termíny "unipolární" a "bipolární" spínání souvisí s tím, jaká napětí vidí motor na svých svorkách. Pro jednoduchost předpokládejme, že budeme řídit stejnosměrný motor s cizím buzením (permanentní magnet) a tudíž máme k dispozici jedno vinutí se dvěma svorkami. K tomu, abychom byli schopni plně řídit tento motor, potřebujeme jednoduchou topologii střídače se čtyřmi tranzistory (dvě větve), kde máme možnost zcela nezávisle spínat kterýkoliv ze čtyř tranzistorů. Topologii znázorňuje Obrázek 1.5.

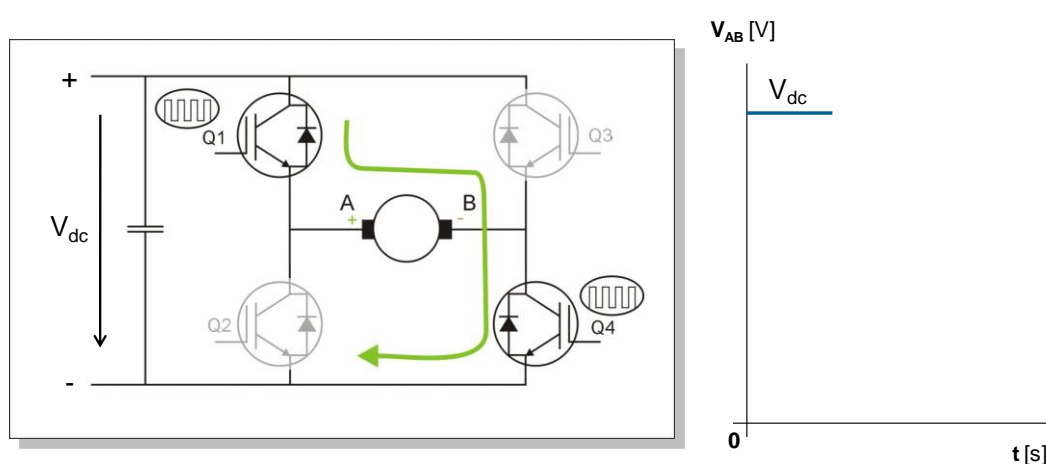


Obrázek 1.5 Topologie střídače ve tvaru H-můstku se 4 tranzistory

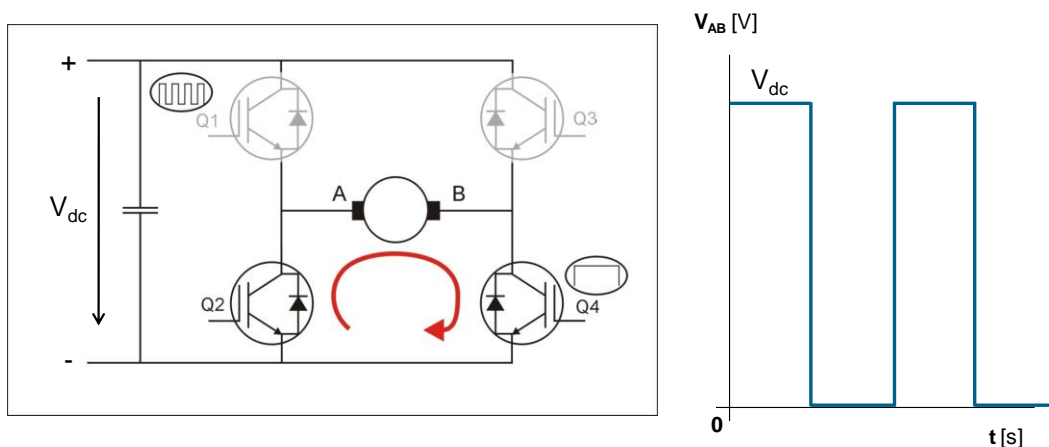
### 1.5.1. *Princip unipolárního spínání*

V případě unipolárního spínání pro jeden směr otáčení dochází k následnému postupnému spínání tranzistorů Q1-on a Q4-on (Q2-off, Q3-off) a poté Q2-on a Q4-on (Q1-off, Q3-off).

1. Q1-on a Q4-on (Q2-off, Q3-off) - v tomto případě na svorkách motoru A, B je přímo připojeno napětí  $V_{dc}$ . Tento stav ukazuje Obrázek 1.6.
2. Q2-on a Q4-on (Q1-off, Q3-off) - zde dochází ke zkratování svorek motoru A,B a de facto k připojení nulového napětí. Tento stav ukazuje Obrázek 1.7.



Obrázek 1.6 Unipolární spínání - případ 1.

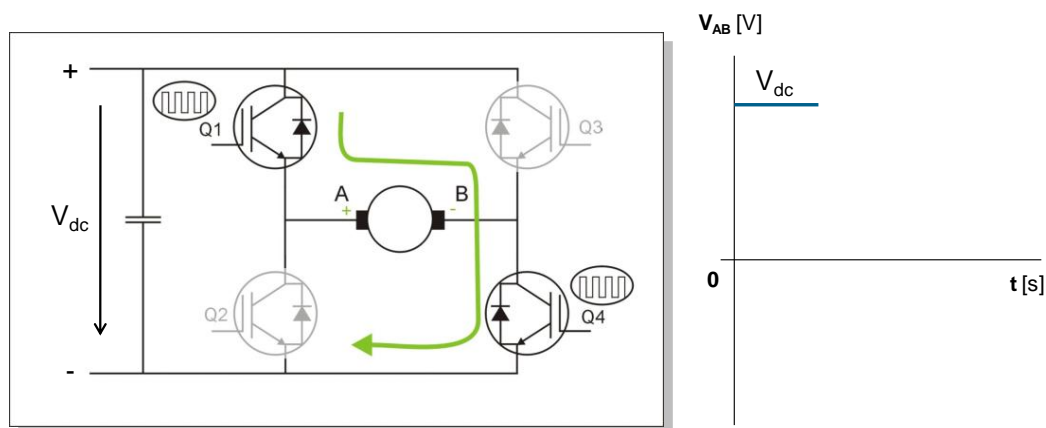


Obrázek 1.7 Unipolární spínání - případ 2.

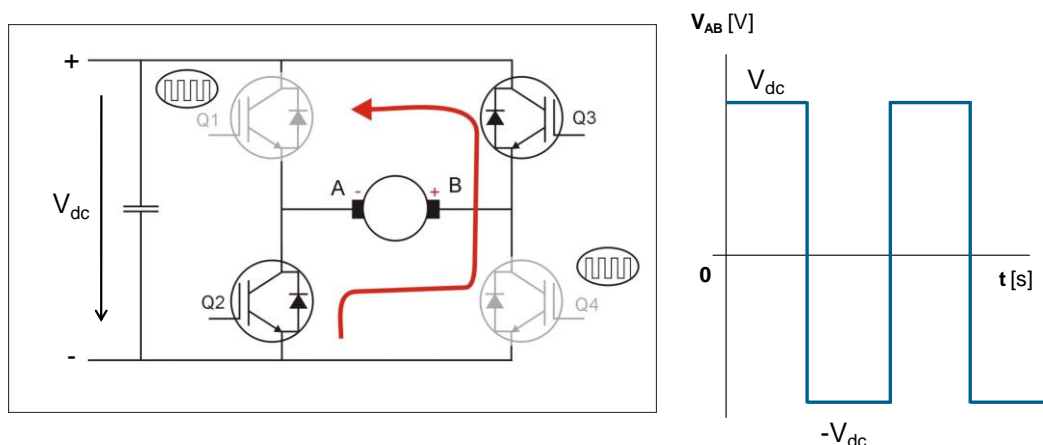
### 1.5.2. Princip bipolárního spínání

V případě bipolárního spínání dochází k následnému postupnému spínání tranzistorů Q1-on a Q4-on (Q2-off, Q3-off) a poté Q2-on a Q3-on (Q1-off, Q4-off).

1. Q1-on a Q4-on (Q2-off, Q3-off) - v tomto případě na svorkách motoru A, B je přímo připojeno napětí  $V_{dc}$ . Tento stav ukazuje Obrázek 1.8.
2. Q2-on a Q3-on (Q1-off, Q4-off) - v tomto případě na svorkách motoru A, B je připojeno napětí  $V_{dc}$  s opačnou polaritou, tedy  $-V_{dc}$ . Tento stav ukazuje Obrázek 1.9.



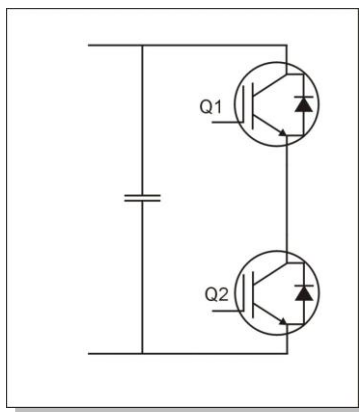
Obrázek 1.8 Bipolární spínání - případ 1.



Obrázek 1.9 Bipolární spínání - případ 2.

## 1.6. Nezávislé vs. komplementární spínání

Termíny "nezávislé" a "komplementární" spínání souvisí s tím, jaký je stav jednotlivých tranzistorů v jedné větvi střídače během jedné periody sepnutí. Pro jednoduchost předpokládejme, že budeme hovořit o jedné větvi střídače, tedy o tranzistorech  $Q1$  a  $Q2$  viz Obrázek 1.10.

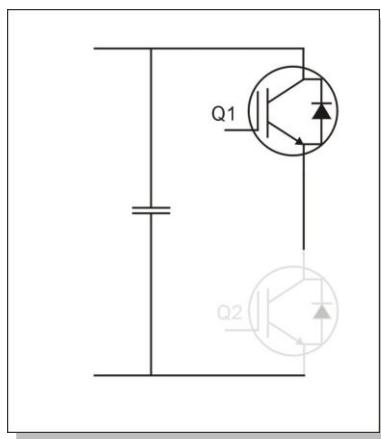


Obrázek 1.10 Topologie jedné větve střídače.

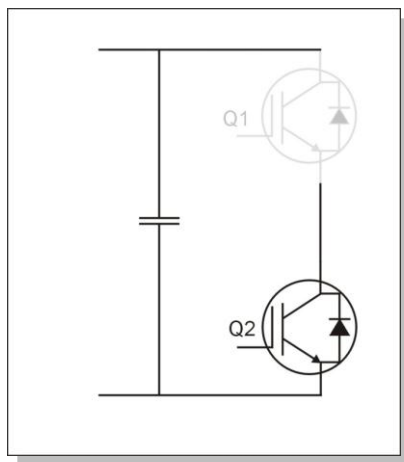
### 1.6.1. Princip nezávislého spínání

V případě nezávislého spínání je vždy jeden z tranzistorů sepnut po celou periodu spínací periodu a druhý je řízen PWM signálem tak, aby bylo dosaženo potřebné střední hodnoty napětí na svorkách motoru.

1. Q2-on po celou dobu periody, Q1-on/off v závislosti na střední hodnotě napětí na motoru, viz. Obrázek 1.11.
2. Q1-on po celou dobu periody, Q2-on/off v závislosti na střední hodnotě napětí na motoru, viz. Obrázek 1.12.



**Obrázek 1.11** Nezávislé spínání - případ 1.



**Obrázek 1.12** Nezávislé spínání - případ 2.

### **1.6.2.      *Princip komplementárního spínání***

V případě komplementárního režimu spínání jsou vždy ovládány oba tranzistory v jedné větvi invertoru tzv. komplementárním způsobem, což znamená, že pokud je jeden tranzistor sepnutý (např. Q1), druhý (Q2) je vypnutý a naopak. Specifický případ nastává v okamžiku přepnutí, kdy dochází k vypínání tranzistoru Q1 a spínání tranzistoru Q2. V tomto případě, abychom zabránili zkratu střídače, jsou po určitou dobu vypnuty oba tranzistory. Době, po kterou jsou oba tranzistory vypnuty, se říká "deadtime".



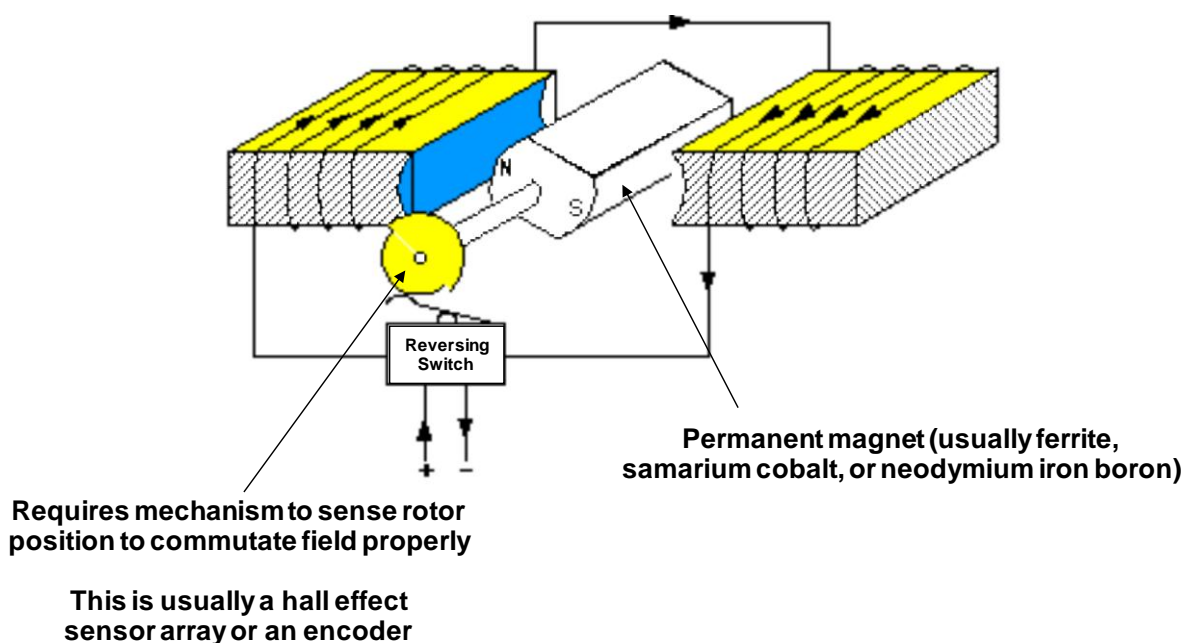
## 2. BLDC MOTOR

*BLDC motor (Brushless DC Motor) je bezkartáčový elektronicky spínaný DC motor, který je hojně používán v jednoduchých aplikacích, kde nezáleží na rovnoměrnosti momentu. Pro svoji konstrukční jednoduchost, relativně snadné řízení, cenovou dostupnost a spolehlivost téměř zcela vytlačil klasické stejnosměrné motory.*

---

### 2.1. BLDC motor - princip

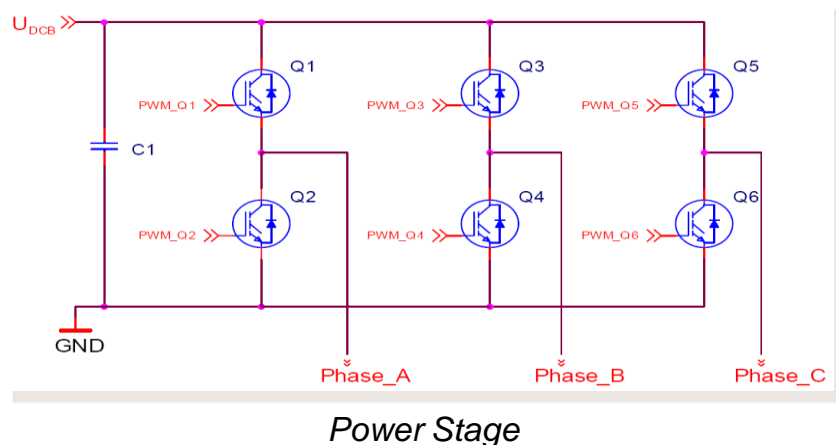
Konstrukčně je BLDC motor tvořen statorem, na kterém je navinuto statorové vinutí, obvykle 3-fázové. Rotor je tvořen permanentním magnetem, který vytváří konstantní magnetické pole ve vzduchové mezeře. BLDC motory se vyrábějí jako 2-pólové nebo n-pólové. BLDC motor na rozdíl od klasického DC motoru nemá mechanický komutátor. Komutaci je tedy zapotřebí provádět elektronicky. Abychom byli schopni provádět komutaci ve správném okamžiku, je zapotřebí znát informaci o poloze rotu. Toto se obvykle řeší pomocí Hallových sensorů v případě aplikací, kde záleží na ceně nebo pomocí obvykle inkrementálního snímače v případě vyšších nároků na pohon. Existují i řešení bez použití snímače a takováto řešení se pak nazývají obecně "sensorless řízení". Základní princip BLDC motoru ukazuje Obrázek 2.1.



Obrázek 2.1 Princip BLDC motoru.

## 2.1. Invertor pro řízení BLDC motoru

Základní řídicí algoritmus je tzv. "six step komutace". K tomu, abychom byli schopni elektronicky komutovat jednotlivé fáze statorového vinutí a tím docílit rotačního magnetického pole ve vzduchové mezeře a následně elektromagnetického momentu, který způsobí otáčení rotoru BLDC motoru, je zapotřebí mít možnost ovládat všechny tři fáze motoru (nejběžnější topologie motoru). K tomuto účelu se používá 3-fázový střídač (Obrázek 2.2).

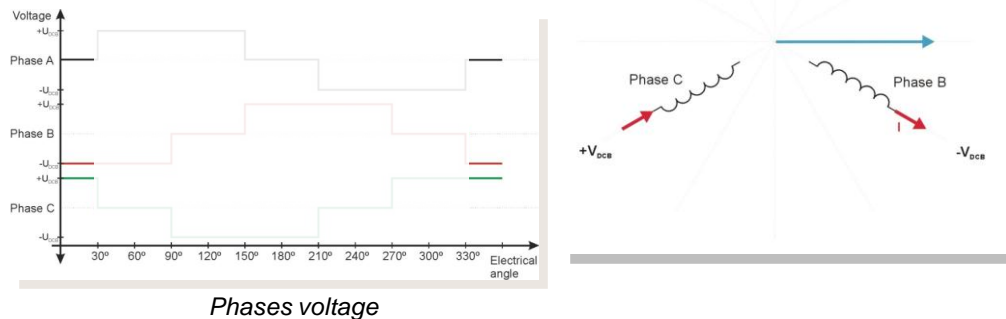


Obrázek 2.2 3-fázový střídač.

Z obrázku je patrné, že se střídač skládá ze tří větví tranzistorů, po dvou tranzistorech v každé větvi, což dohromady činí 6 tranzistorů. Středů jednotlivých větví jsou pak připojeny na jednotlivé fáze statorového vinutí BLDC motoru. Jednotlivé tranzistory jsou řízeny mikropočítačem, kde je implementovaný řídicí algoritmus.

## 2.1. Six step komutace

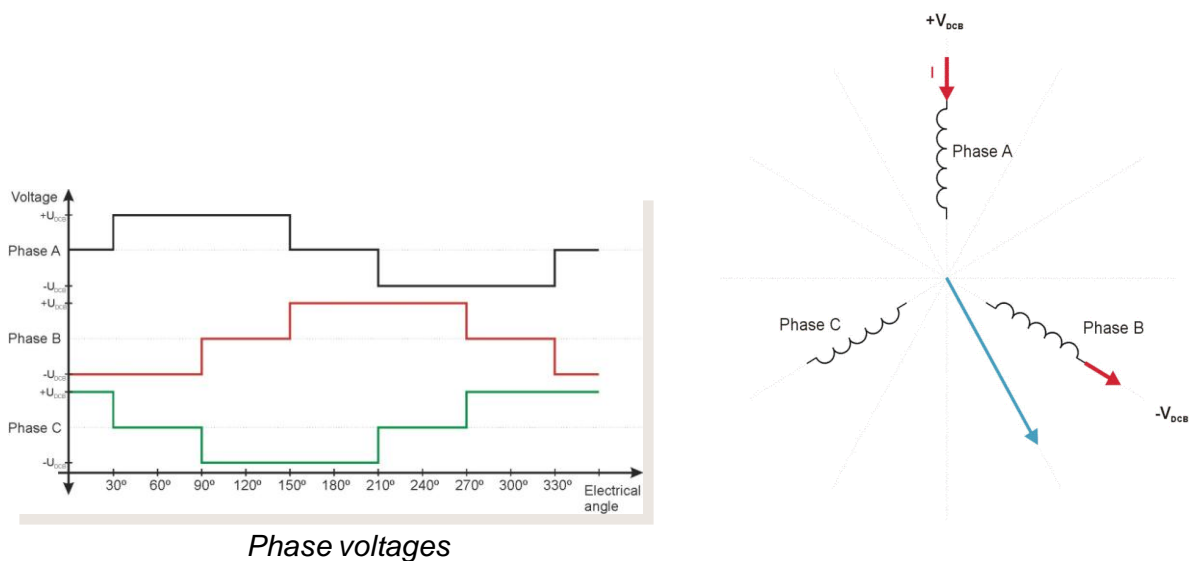
Tento základní řídicí algoritmus vyžaduje znalost o poloze rotoru. V případě "six step" komutace jsou vždy napájeny pouze dvě fáze motoru. Odpovídající fáze jsou napájeny na základě znalosti aktuální pozice rotoru a požadovaného směru otáčení. Jelikož na jednu elektrickou otáčku generujeme pouze 6 spínacích vzorů, mění se vektor magnetického pole po skocích  $60^\circ$  elektrických, což způsobuje zvlnění momentu, které je pro některé aplikace nepřijatelné. Proto je zapotřebí zvážit, jestli je algoritmus "six step" komutace vhodný pro daný pohon. Příklad jednoho spínacího vzoru ukazuje Obrázek 2.3.



**Obrázek 2.3 Příklad jednoho komutačního vzoru.**

Z obrázku je zřejmé, že se napájí pouze fáze B a C a fáze A je nezapojena. Vektor magnetického pole odpovídající dané kombinaci statorových fází je zobrazen modře a je kolmý na fázi A.

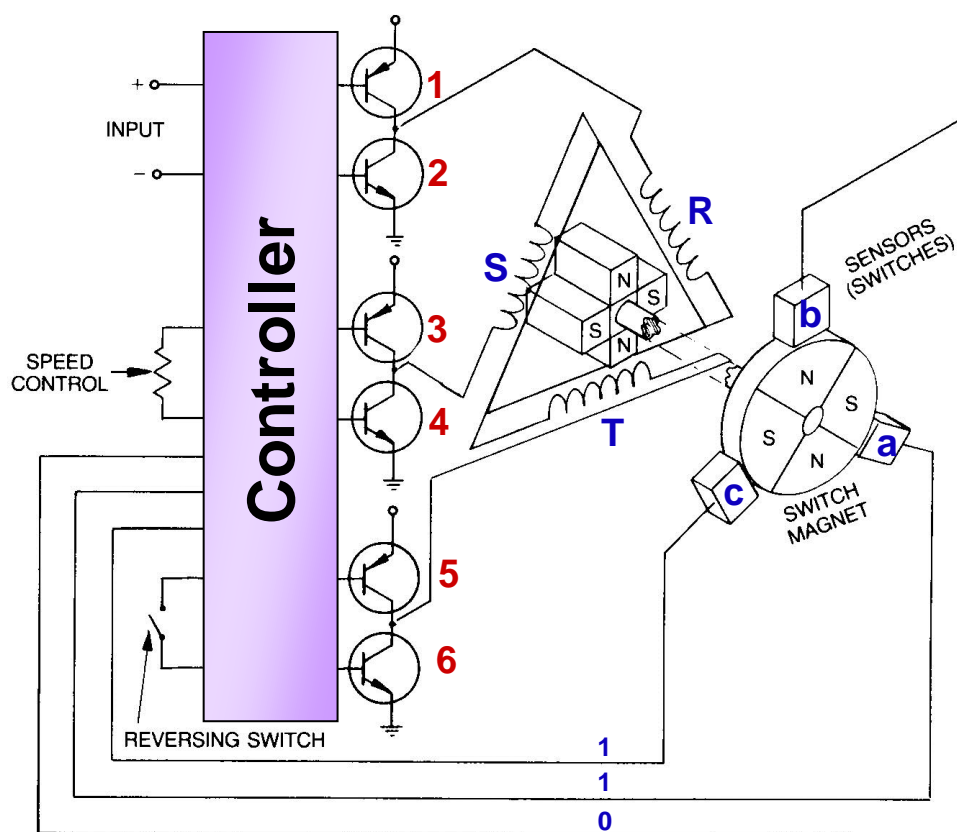
Kompletní spínací vzory pro jednu elektrickou otáčku zobrazuje Obrázek 2.4.



**Obrázek 2.4 Spínací vzory pro jednu elektrickou otáčku**

## 2.1. BLDC komutace na základě Hallových sensorů

Principiální schéma řízení BLDC motoru na základě informace z Hallových sensorů ukazuje Obrázek 2.5.

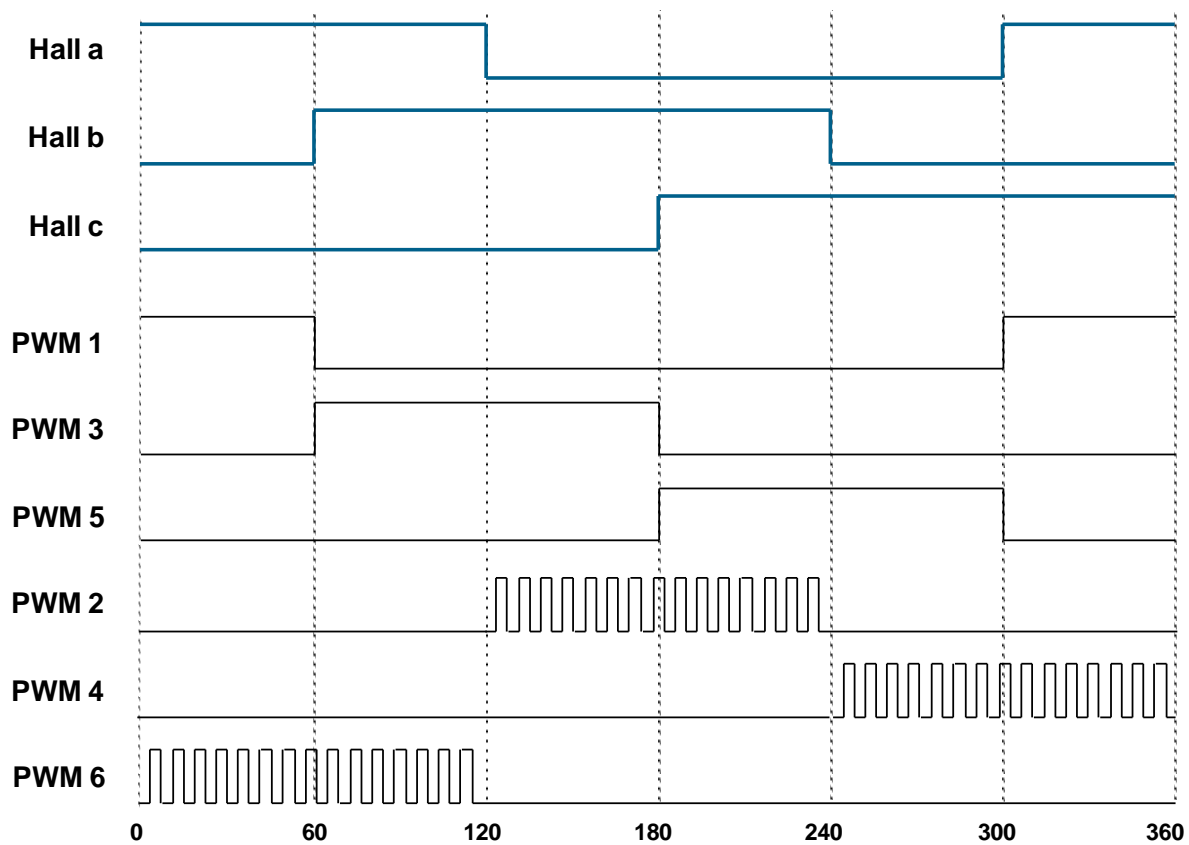


PRINCIPLES OF OPERATION

Source: Eastern Air Devices, Inc. Brushless DC Motor Brochure

#### Obrázek 2.5 Princip řízení BLDC motoru na základě Halloových sensorů

Tři Hallovy snímače jsou spojeny s rotorem tvořeným permanentními magnety a zarovnány takovým způsobem, aby odpovídaly sdruženému indukovanému napětí. Pak lze použít informace z Halloových snímačů pro rozpoznání aktuální polohy rotoru s přesností  $\pm 30^\circ$  elektrických. Řídicí algoritmus zpracovává informace z Halloových sensorů a na základě komutační tabulky naprogramované v mikropočítači pak následně volí odpovídající kombinace fází pro sepnutí. Stav jednotlivých Halloových sensorů a jim odpovídající PWM signály generované pro jednotlivé tranzistory ukazuje Obrázek 2.6.



Obrázek 2.6 Stav Hallových sensorů a odpovídající generace PWM signálu

### 3. EMBEDDED MIKROPOČÍTAČ FREESCALE MC56F8006

---

#### 3.1. Rodina mikropočítačů Freescale MC56F800x

Rodina mikropočítačů Freescale DSC56F800x čítá dva zástupce. Liší se ve velikosti a typu pouzdra, velikosti paměti FLASH, v počtu GPIO pinů a analogových vstupů ADC převodníku. Oba mikropočítače jsou založeny na jádře DSP56800E. Napájecí napětí je 3.3V. Hodiny mikropočítače běží na frekvenci 32MHz. Standardní ladící rozhraní je JTAG/Enhanced OnCE.

Do rodiny mikropočítačů Freescale DSC56F800x patří následující zástupci:

- **MC56F8002** – pouzdro je 32 pinové nebo 48 pinové LQFP nebo 28 pinové SOIC, programová/datová Flash – 12kB, programová/datová RAM – 2kB
- **MC56F8006** – pouzdro je 32 pinové nebo 48 pinové LQFP nebo 28 pinové SOIC, programová/datová Flash – 16kB, programová/datová RAM – 2kB

#### 3.2. DSC56F8006/56F8002 – základní parametry

Mikropočítače DSC56F8006/56F8002 jsou založeny 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 mikropočítače DSC56F8006/2 cenově výhodné řešení pro širokou škálu aplikačního využití. Jsou vhodné 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

	56F8002	56F8006
Performance	32MHz/MIPs	32MHz/MIPs
Temperature Range (V)	-40C~105C	-40C~105C
Voltage Range	1.8V - 3.6V	1.8V - 3.6V
Voltage Regulator	On-Chip	On-Chip
Program/Data Flash	12KB	16KB
Program/Data RAM	2KB	2KB
Program Security	Yes	Yes
On Chip Relaxation Osc.	Yes	Yes
PLL	Yes	Yes
COP (Watchdog)	Yes	Yes
PWM (96 Mhz Clock)	1 x 6ch	1 x 6ch
PWM Fault Inputs	4	4
12-bitADCs	2 x 8ch	2 x 12ch
12-bitDACs	0	0
Analog Comparator	3	3
Prog Gain Amp	2	2
16-bitTimers	3	3
Prog. Interval Timers	1 (RTC)	1 (RTC)
GPIO (max) (+/-8mA)	23	40
IIC	1	1
SCI (UART) / LIN Slave	1 - SCI	1 - SCI
SPI (Synchronous)	1 - SPI	1 - SPI
CAN	No	No
JTAG/EOnCE	Yes	Yes
Power Consumption	IDD = 45.6mA; IDDA = 4.5mA	
Package	32LQFP (.8p)	28SOIC 32LQFP 32SDIP 48LQFP

**Obrázek 3.1 Přehled mikropočítačů rodiny MC56F800x a jejich vlastnosti**

- Frekvence jádra – 32 MHz, což odpovídá výpočetnímu výkonu maximálně 32MIPS
- 56F8006 - 16KB (8K x 16) Program Flash
- 56F8002 - 12KB (6K x 16) Program Flash
- 56F8006 - 2KB (1K x 16) Unified Data/Program RAM
- 56F8002 - 2KB (1K x 16) Unified Data/Program RAM
- PWM modul – flexibilní a velmi komplexní periferie s 6 PWM kanály

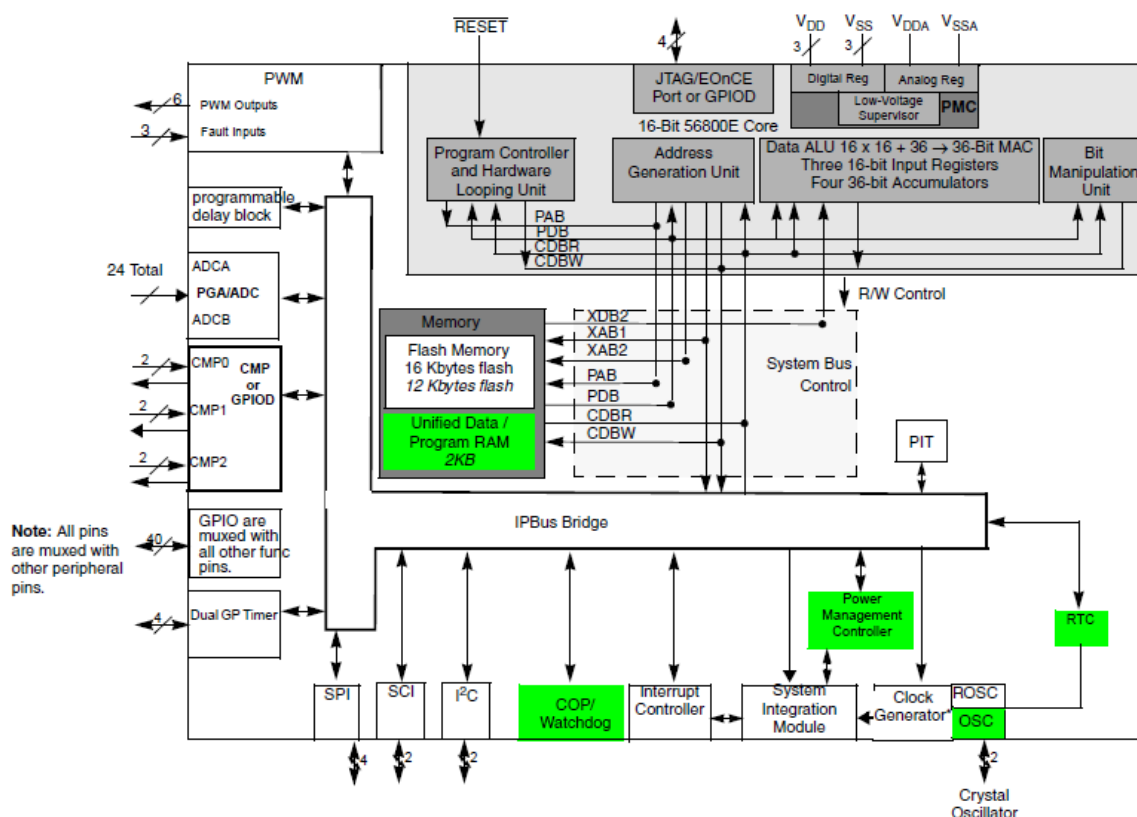
- Dva 28-kanálové 12-bitové A/D převodníky (ADCs)
- Dva programovatelné zesilovače (PGA - Programmable Gain Amplifier) se zesílením až 32x
- Tři analogové komparátory
- Jeden programovatelný časovač – PIT (Programmable Interval Timer)
- Jeden modul SCI (Serial Communication Interface) s LIN slave funkcí
- Jeden modul SPI (Serial Peripheral Interfaces)
- Jeden modul 16-bitových časovačů – Quad Timer (TMR)
- Jeden modul I2C (Inter-Integrated Circuit)
- Jeden modul programovatelného zpoždění - PDB (Programmable Delay Block)
- Watchdog – COP (Computer Operating Properly)
- Relaxační oscilátor integrovaný na čipu
- Integrovaný Power-On Reset (POR) a Low-Voltage Interrupt (LVI)
- JTAG/Enhanced On-Chip Emulation (OnCE™)
- Až 40 GPIO

### 3.3. 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 bázové adresy, rozsahy vstupních a výstupních napětí, různé důležité údaje specifické pro daný čip, atd.





**Obrázek 3.2** Blokové schema mikropočítače MC56F8006/MC56F8002

- 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

### 3.4. ADC – Analog Digital Converter

ADC modul se skládá ze dvou nezávislých ADC převodníků, z nichž každý obsahuje samostatný sample and hold (S/H) obvod. Převodníky dokáží zpracovat až 28 (2 x 14) externích analogových vstupů plus 7 interních vstupů. Analogové vstupy jsou jednoduše konfigurovatelné díky vstupnímu multiplexeru. ADC kromě analogové části obsahuje i výkonnou řídicí logiku umožňující efektivní a snadné nastavování a přeprogramování v reálném čase

#### 3.4.1. ADC - vlastnosti

ADC modul charakterizují následující vlastnosti:

- Rozlišení – 12-bitů
- ADC synchronizace s PWM modulem – propojeno vnitřně na čipu
- Schopnost sekvenčně převést a uložit až 4 měření
- Podpora pro převádění v simultánním režimu - oba převodníky se spouštějí ve stejném okamžiku - sample & hold + převod
- Teplotní sensor - připojený na kanály ANA26 a ANB26
- Ping-pong mode - hardwarová synchronizace umožňující převod 4 analogových signálů

Následující Obrázek 3.3 demonstruje konfiguraci ADC modulu

### Obrázek 3.3 Blový diagram ADC modulu

### 3.5. PDB – Programmable Delay Block

Aplikace vyžadující synchronizaci PWM signálu s okamžikem snímání analogových veličin využívají pro přesné nastavení okamžiku spouštění ADC modulu právě PDB modul. PWM modul generuje synchronizační signál SYNC. Primární funkce PDB modulu je vytvoření programovatelného zpoždění relativně k signálu SYNC a generování signálů pro spouštění ADC modulu a modulu PGA.

#### 3.5.1. *PDB - vlastnosti*

PDB modul charakterizují následující vlastnosti:

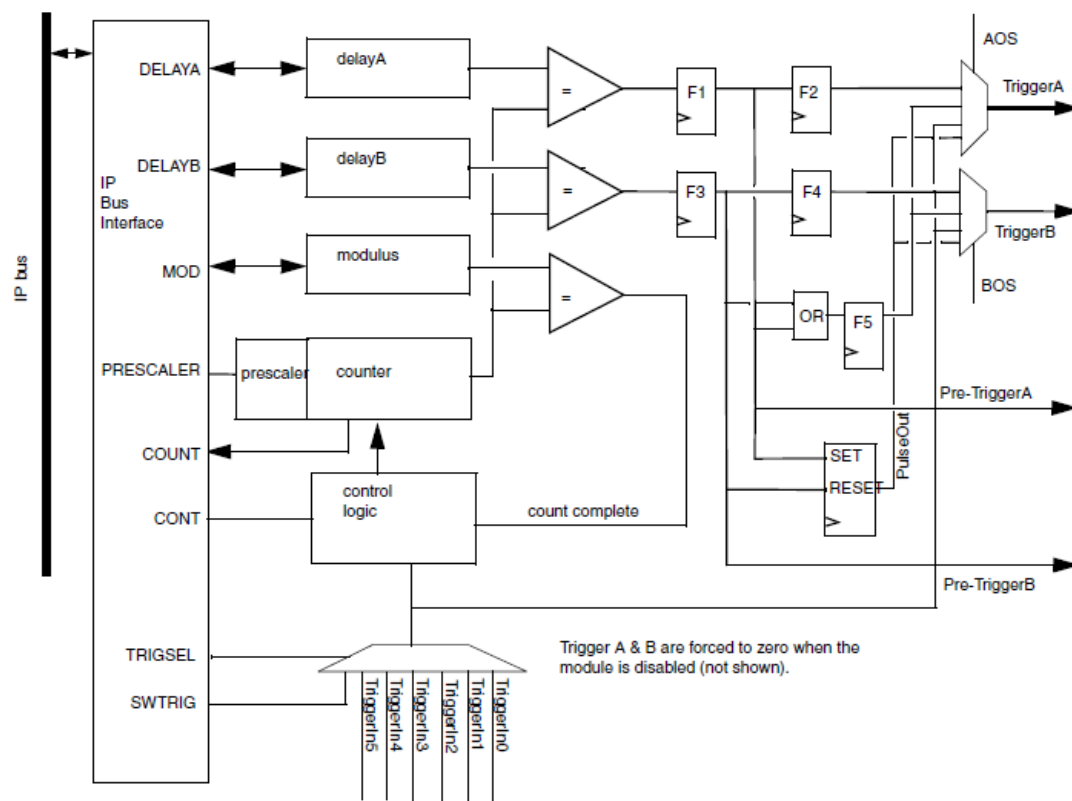
- Nástupná hrana signálu trigger\_in spouští čítač
- Podpora dvou trigger\_out signálů. Každý z nich umožňuje nezávisle řídit zpoždění oproti sync\_in signálu
- Výstupní spouštěcí signály mohou být zpracovány logickou funkcí OR, což umožňuje plánovat 2 převody iniciované jednou spouštěcí událostí
- Podporuje módy "continuous trigger" a "single-shot"
- Každý výstupní spouštěcí signál může být povolen/zakázán nezávisle

#### 3.5.2. *PDB – blokový diagram*

Obrázek 3.4 demonstruje konfiguraci PDB modulu

### 3.1. PGA – Programmable Gain Amplifier

PGA modul předpokládá spolupráci s ADC modulem. Samostatné využití PGA modulu nemá smysl. Slouží k zesilování analogových signálů následně pak převáděných ADC modulem



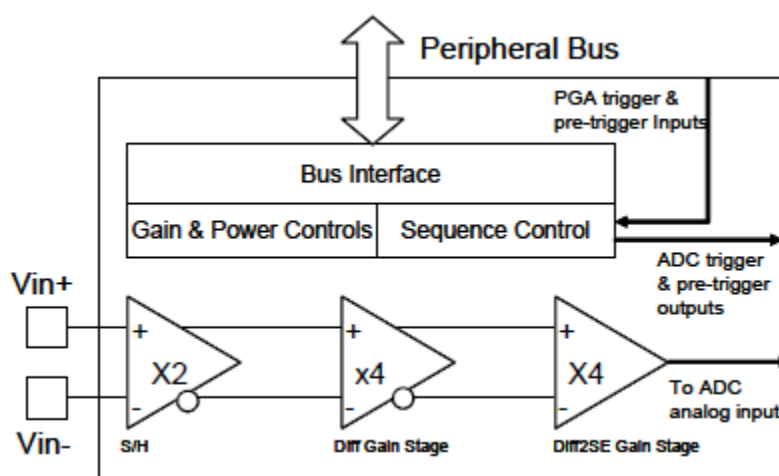
### 3.1.1. PGA - vlastnosti

PGA modul charakterizují následující vlastnosti:

- Několik programovatelných zesílení - 1x, 2x, 4x, 8x, 16x a 32x
- Samplovací čas je možno přesně řídit - méně než 0,1 us
- Samplovací okamžik je možno synchronizovat s PWM modulem a PDB modulem
- PGA není analogový zesilovač
- Rozsah vstupního napětí - rail-to-rail
- "Single-ended" výstup je přímo vedený na ADC kanály ANA15 a ANB15
- Trigrování - software nebo hardware
- Zahrnuje kalibrační vlastnosti umožňující kalibraci PGA modulu

### 3.1.2. PGA – blokový diagram

Obrázek 3.5 demonstruje konfiguraci PGA modulu



Obrázek 3.5 Blokový diagram PGA modulu

## 3.2. COP – Computer Operating Properly

COP (Computer Operating Properly) modul slouží k tomu, aby 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 countru, 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 obsluhu COP logiky zápisem specifických čísel v do counter registru ve správném pořadí. V tomto případě je nutno periodicky updatovat CNTR (Counter Register) registr sekvencí čísel 0x5555 a následně 0xAAAA.

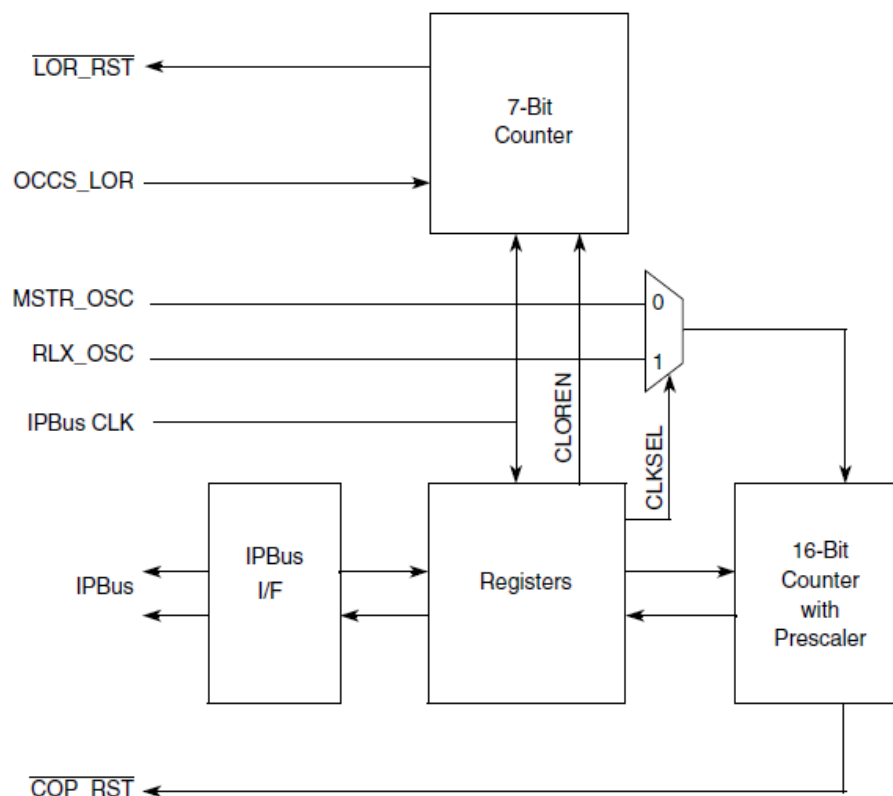
### 3.2.1. COP – vlastnosti

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

### 3.2.2. COP – blokový diagram



Obrázek 3.6 Blokový diagram COP modulu

## 3.3. 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 je to pak dva hodinové signály. Hodinový signál PWM modulu je volitelný a může být roven systémovým hodinám nebo trojnásobku systémových hodin. V případě

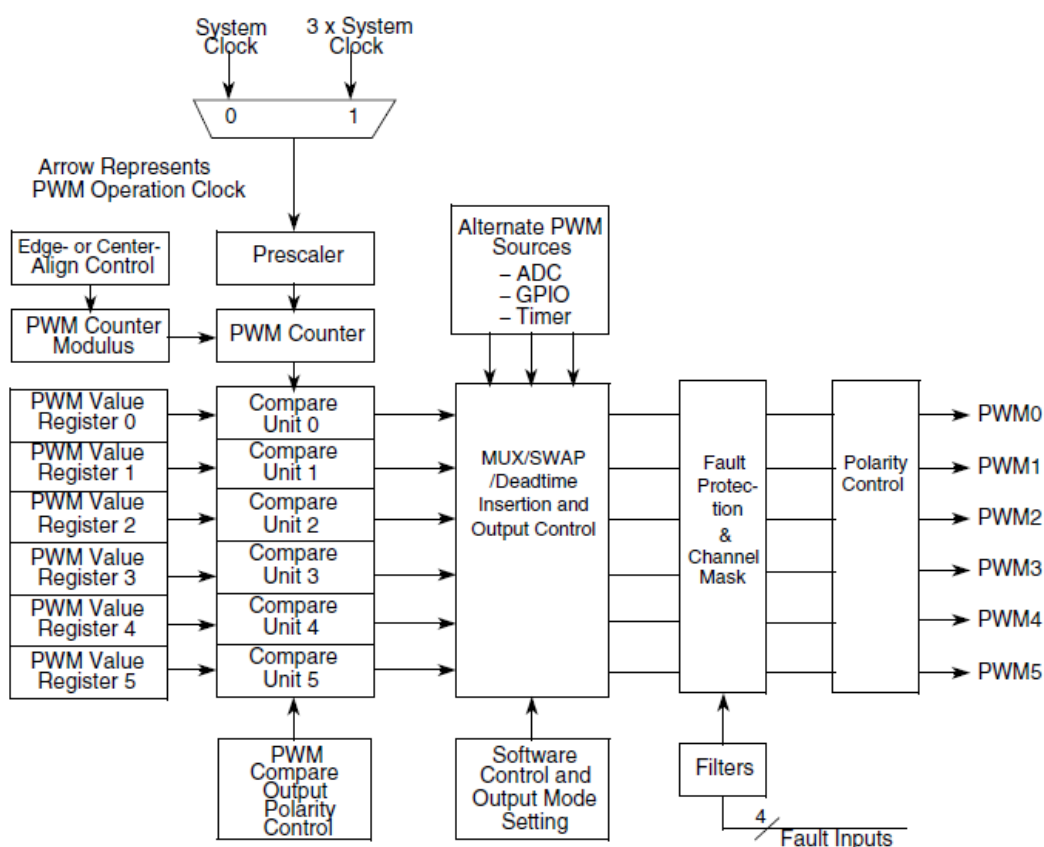
generování signálu v komplementárním módu je automaticky vkládaný deadtime (řízeno hardwarově). Každý PWM výstup může být volitelně řízen PWM generátorem, časovačem, výsledkem ADC převodu, GPIO pinem nebo manuálně softwarem. Nezávisle lze řídit výstupní polaritu PWM signálu pro horní a dolní výstupy. Rovněž lze generovat nesymetrický PWM výstup.

### **3.3.1. *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 fault ochrana
- 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 ochranou proti zápisu

### 3.3.2. PWM – blokový diagram



Obrázek 3.7 Blokový diagram PWM modulu

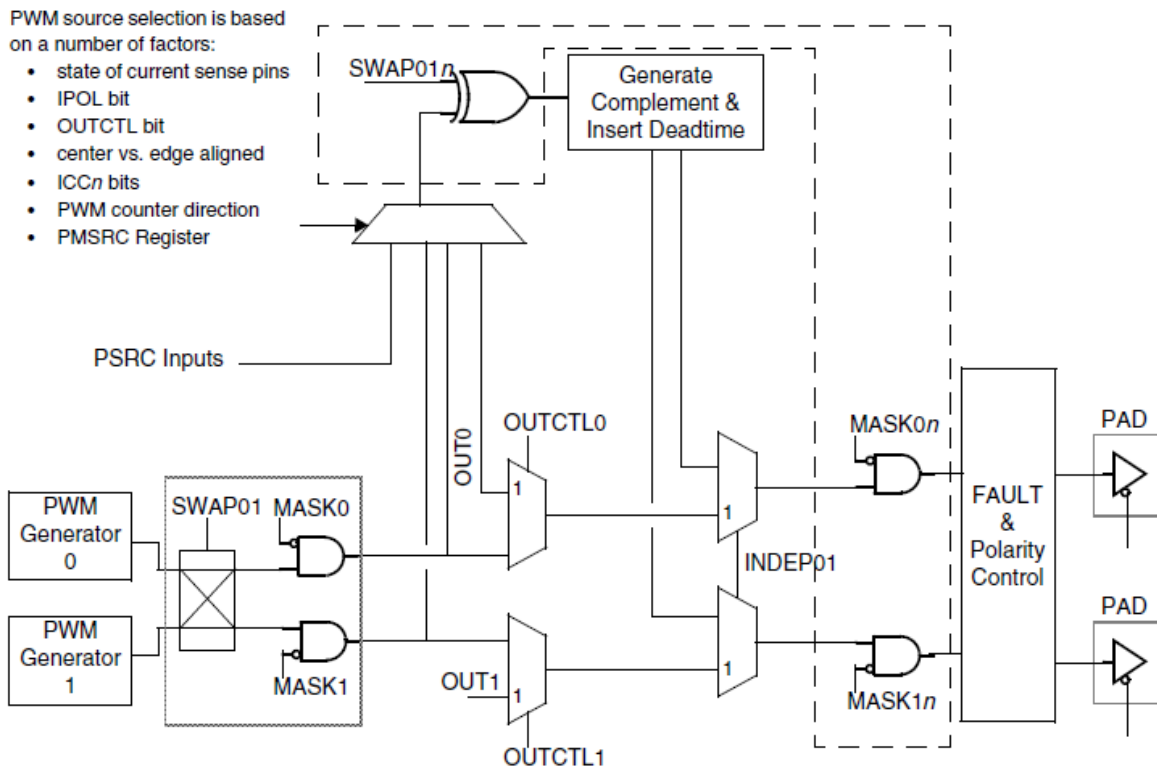
## 3.1. OCCS – On-Chip Clock Synthesis

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.

### 3.1.1. OCCS – vlastnosti

Vlastnosti OCCS modulu jsou následující:

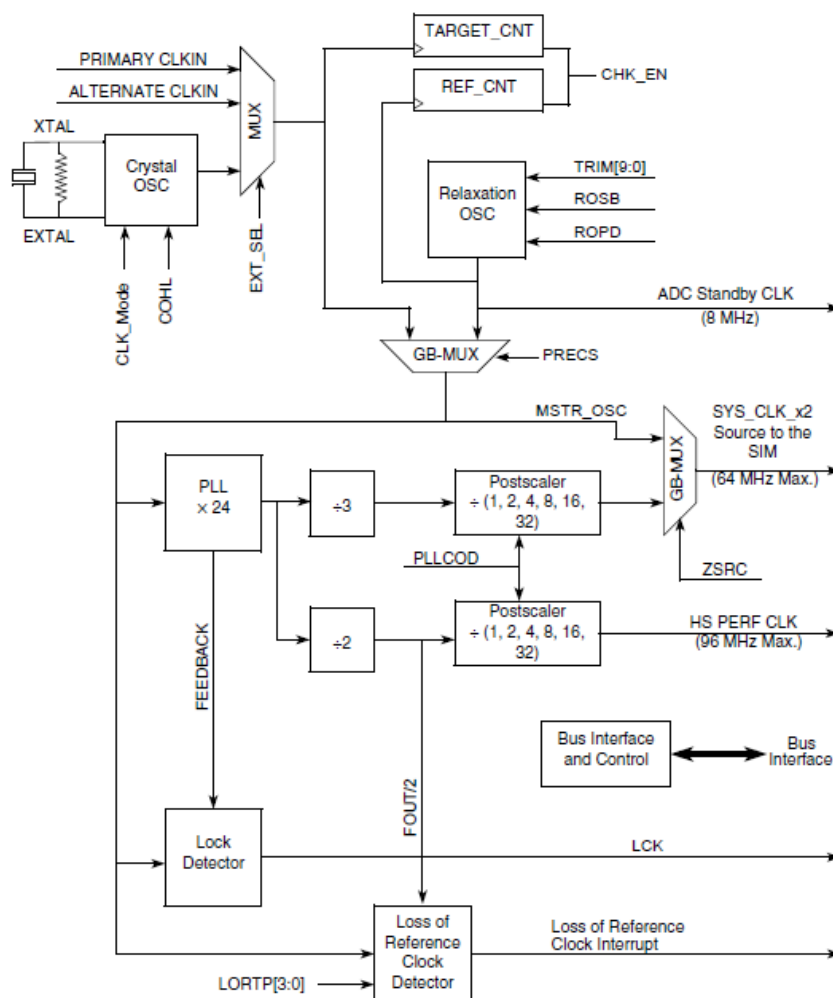




**Obrázek 3.8** Detailní blokový diagram funkcí MUX, Swap and Deadtime

- 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 čítač a PWM moduly
- Standby mód relaxačního oscilátoru

### 3.1.2. OCCS – blokový diagram



Obrázek 3.9 Blokový diagram OCCS modulu s externím krystalovým oscilátorem

## 3.2. FM – Flash Memory

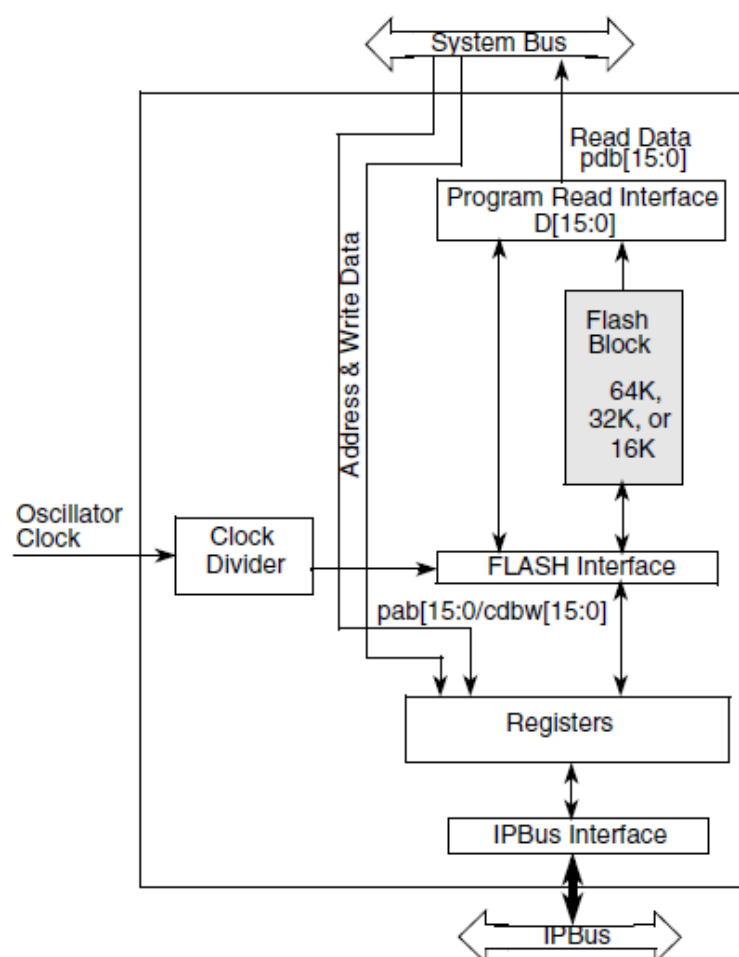
Paměť Flash (FM) je typu non-volatile, což znamená, že informace uložená v paměti zůstává zachovaná i po odpojení napájení. Používá se pro ukládání programu. FM implementovaná na mikropočítačích Freescale podporuje přeprogramování bez použití externího napájení. Mazání a programování se provádí přes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

### 3.2.1. FM – vlastnosti

- 32 MHz – pro všechny přístupy do programové Flash paměti
- 12 kB nebo 16 kB programové FLASH paměti

- Programování a mazání – automatizovaný proces
- Přerušení na události – command completion, command buffer empty a ccess error
- Rychlé mazání stránky
- Jediné napájení pro programování a mazání
- Bezpečnostní vlastnosti
- Systém – sector protection

### 3.2.2. *FM – blokový diagram*



Obrázek 3.10 Blokový diagram Flash paměti

### 3.3. PIT – Programmable Interval Timer

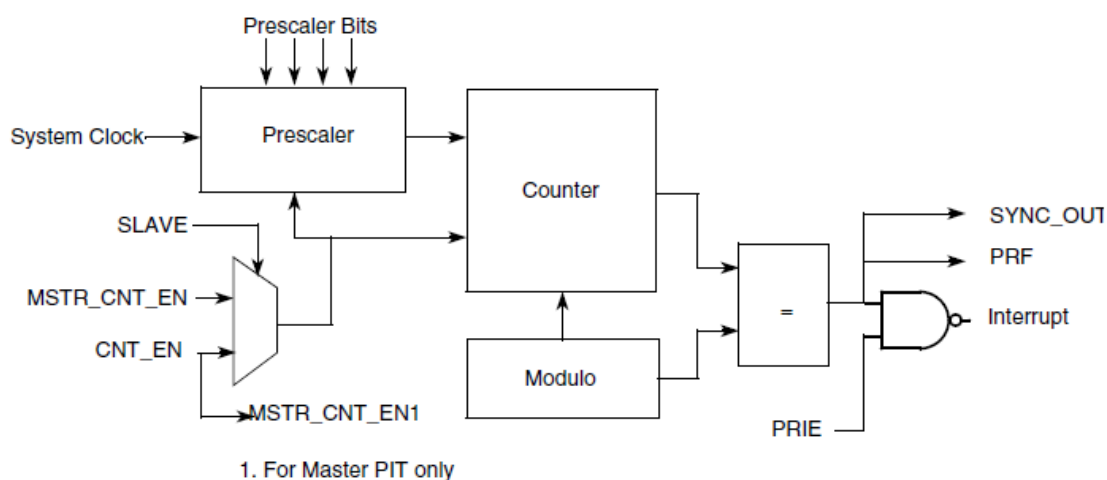
PIT timer modul je velmi jednoduchý časovací modul sloužící pro nenáročné časování aplikace. Obsahuje 16-bitový up counter, modulo registr a řídicí registr. Modulo a řídicí registry jsou pro zápis i pro čtení, zatímco counter registr je pouze pro čtení.

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

#### 3.3.1. PIT – vlastnosti

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

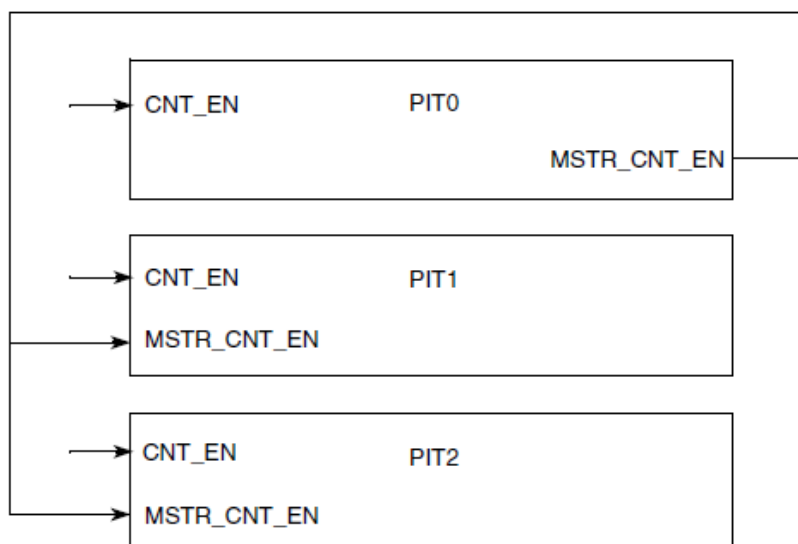
#### 3.3.2. PIT – Blokový diagram



Obrázek 3.11 Blokový diagram PIT modulu

### 3.1. DTMR – Dual Timer

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



**Obrázek 3.12** Propojení PIT modulů ve slave módu

- Prescaler
- Counter
- Load registr
- Hold registr
- Capture registr
- Dva compare registry
- Jeden status a control registr
- Jeden control registr

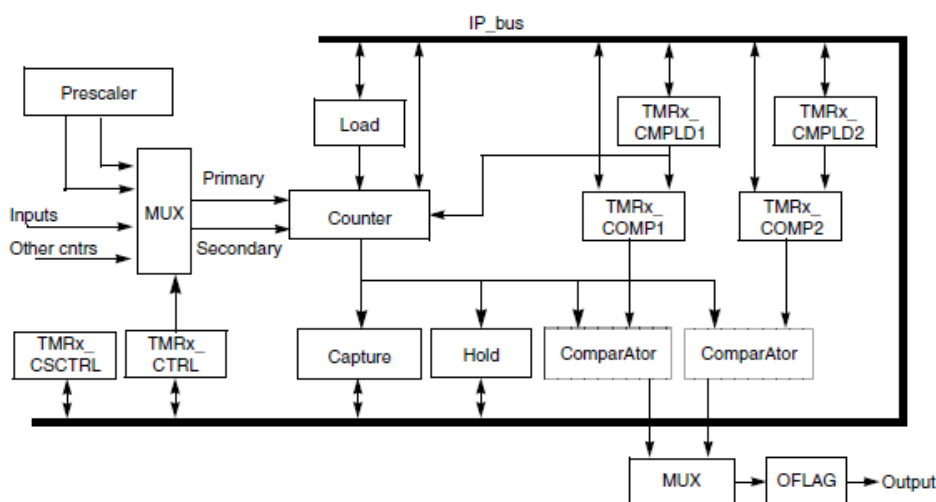
### **3.1.1. DTMR – vlastnosti**

- 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

### 3.1.2. DTMR – blokový diagram

Obrázek 3.13 ukazuje základní zapojení jednoho podmodulu časovacího modulu TMR. Stejná struktura se v modulu DTMR vyskytuje 2x.



Obrázek 3.13 Quad Timer – blokový diagram

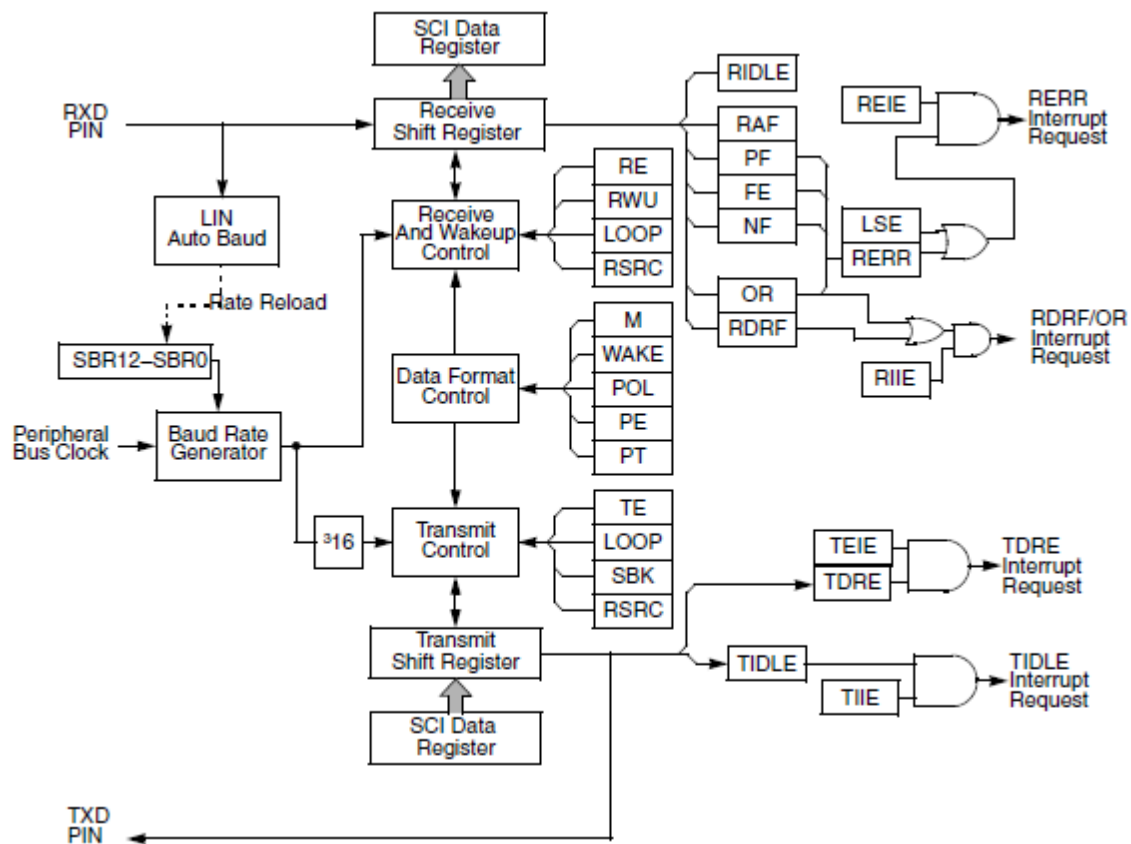
## 3.2. SCI – Serial Communication Interface

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

### 3.2.1. *SCI – vlastnosti*

- Full-duplex nebo single-wire komunikace
- Standardní formát – NRZ (Non-Return-to-Zero)
- Volba baud rate - 13-bitový integer a 3-bitový fractional
- Programovatelný formát dat – 8- nebo 9-bitový formát
- Separátně povolitelný vysílač a přijímač
- Separátně generovaný požadavek na přerušení od přijímače a vysílače
- 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

### 3.2.2. SCI – blokový diagram



Obrázek 3.14 SCI modul – blokový diagram

## 3.3. SPI – Serial Peripheral Interface

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

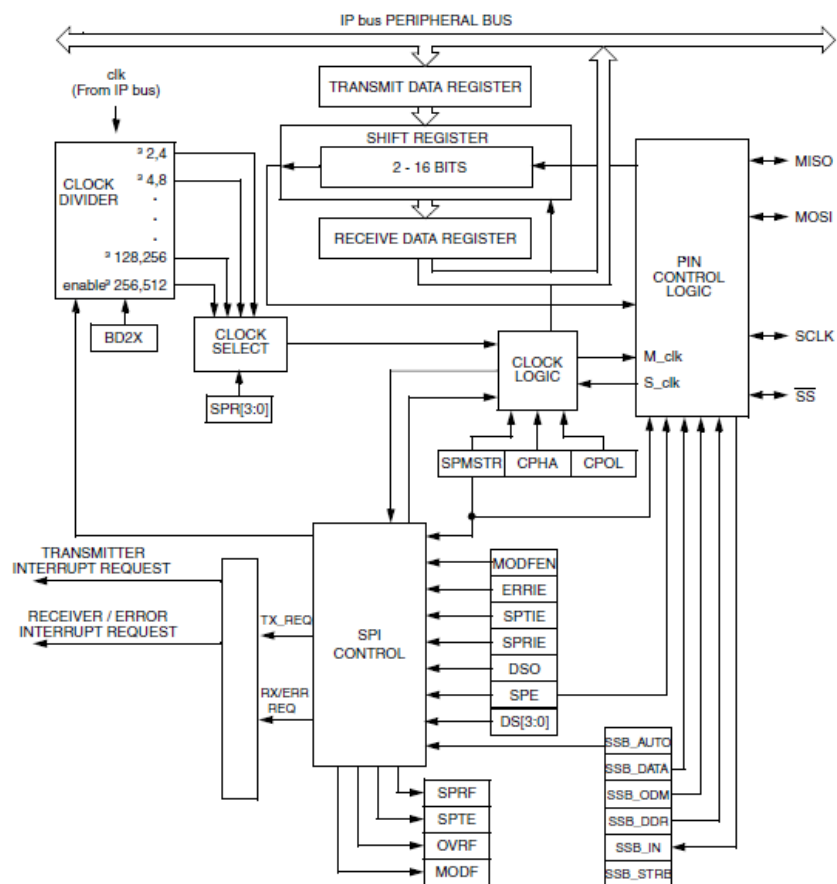
### 3.3.1. SPI – vlastnosti

- Full-duplex
- Master a slave módy
- Double-buffered funkčnost separátně pro vysílací a přijímací registry
- FIFO se čtyřmi úrovněmi pro vysílací a přijímací buffer
- Programovatelná délka přenosu (2 až 16-bitů)
- Maximální frekvence ve slave módu –  $\text{frequency} < \text{bus frequency} / 2$



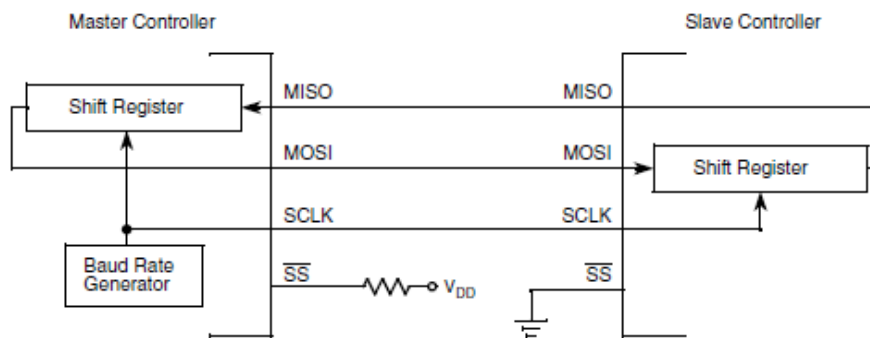
- Programovatelná polarita a fáze sériových hodin
- Dvě separátní přerušení
  - SPI receiver full/error
  - SPTE (SPI transmitter empty)
- Wired OR mód umožňuje propojení několika SPI modulů

### 3.3.2. SPI – blokový diagram



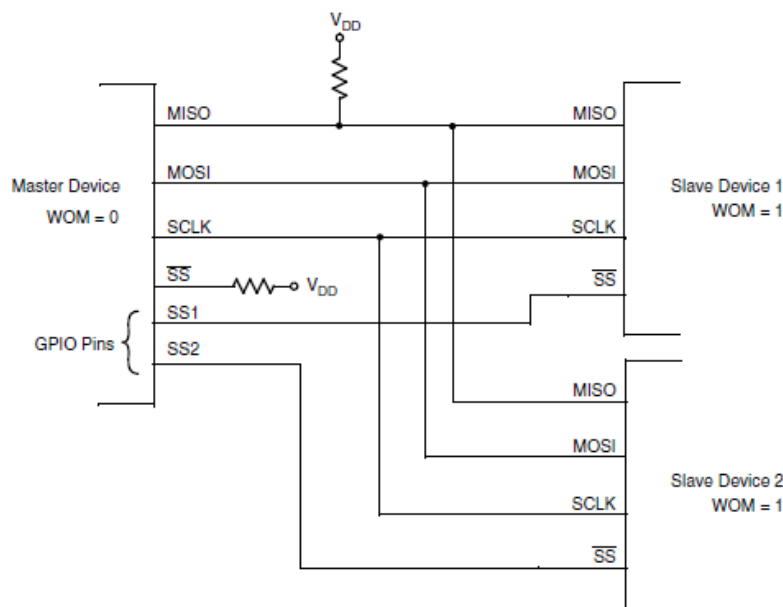
Obrázek 3.15 SPI modul – blokový diagram

### 3.3.3. SPI – master mód



Obrázek 3.16 SPI modul – full-duplex master/slave zapojení

### 3.3.4. SPI – wired-OR mód



Obrázek 3.17 Propojení SPI modulů - Master a dva slaves

## 3.4. HSCMP – High Speed Comparator

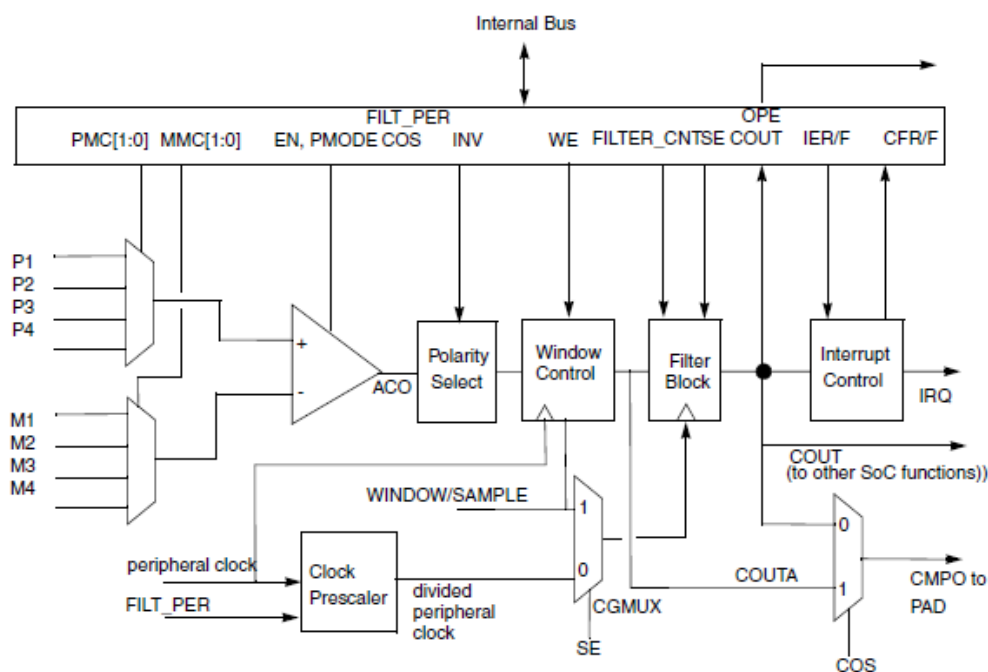
Komparátor se skládá analogového modulu komparátoru a řídicího digitálního rozhraní. Umožňuje komparovat dvě analogová vstupní napětí. Komparátor je navržený takovým způsobem, aby umožňoval pracovat v celém rozsahu napájecího napětí (rail-to-rail)

### 3.4.1. HSCMP – vlastnosti

- Pracuje v celém rozsahu napájecího napětí - rail-to-rail

- Vstupní offset - méně než 40 mV
- Hystereze - méně než 15 mV
- Volitelné přerušení na nástupnou hranu, sestupnou hranu nebo na obě hrany výstupu komparátoru
- Volitelné invertování výstupu komparátoru
- Možnost aplikace externí hystereze

### 3.4.2. HSCMP – blokový diagram



Obrázek 3.18 HSCMP modul – blokový diagram

## 3.5. GPIO – General Purpose Input/Output

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

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

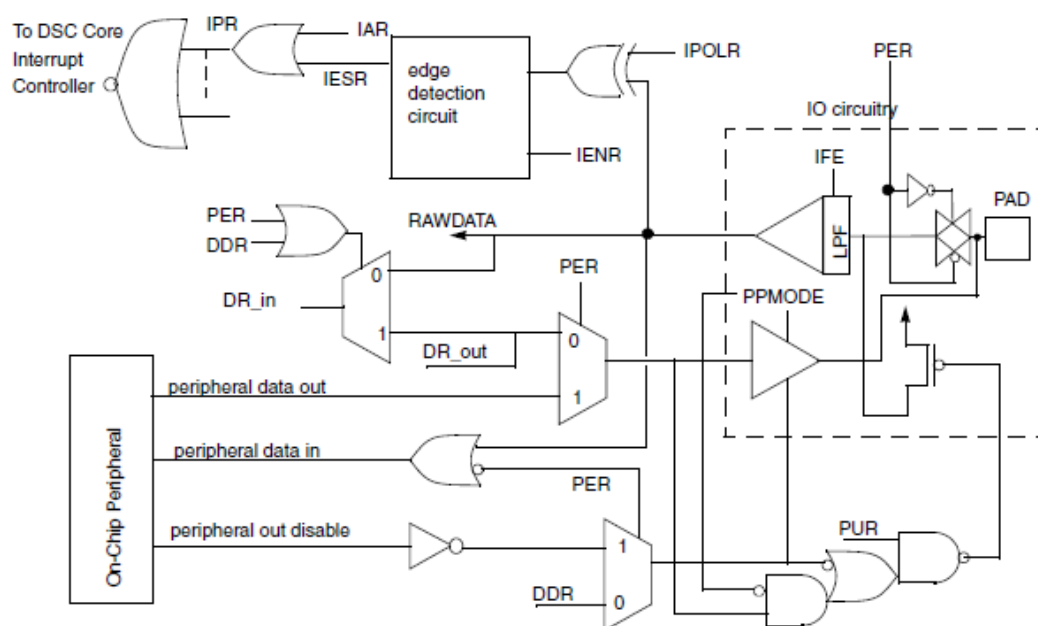
- Jako periferní pin v případě, že je pin multiplexovaný s jiným interním modulem

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

### **3.5.1. *GPIO – vlastnosti***

- Individuální konfigurace pinu – buď mód periferie nebo GPIO
- Konfigurace směru pinu v GPIO módu – input nebo output
- Volitelné připojení pullup rezistoru – konfigurace pinu jako input a to v módu periferie nebo GPIO
- Volba módu push-pull nebo otevřený kolektor - 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
- Generace přerušení
- GPIO input – 5 V tolerant

### 3.5.2. GPIO – blokový diagram



Obrázek 3.19 Logika GPIO modulu – blokový diagram

## 3.6. I<sup>2</sup>C – Inter-Integrated Circuit Interface

I<sup>2</sup>C modul je programovatelná řídicí sběrnice, která vytváří komunikační kanál mezi integrovanými obvody v systému. Jedná se o jednoduchou dvou vodičovou sběrnici se softwarově definovaným protokolem. Používá se např. pro komunikaci se sensory, A/D, D/A převodníky, kodeky a jinými mikropočítači.

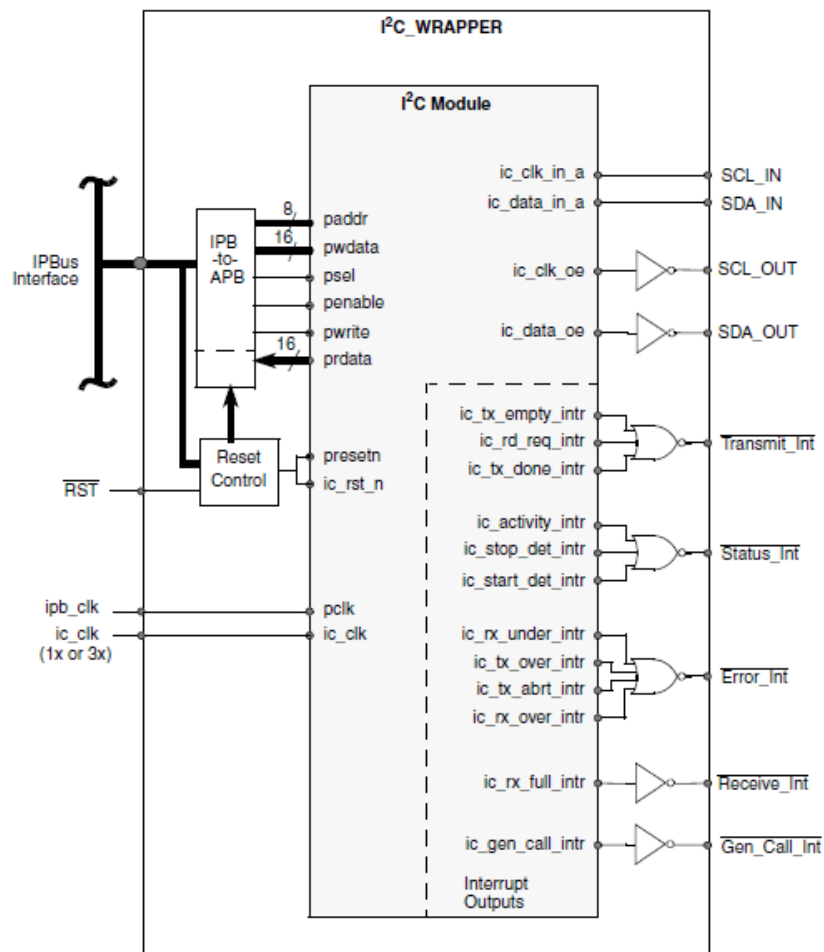
I<sup>2</sup>C je dvou vodičové sériové rozhraní skládající se z SDA (Serial Data Line) a SCL (Serial Clock Line). Tyto dva vodiče přenášejí informaci mezi zařízeními připojenými na sběrnici. Každé zařízení připojené na sběrnici má unikátní adresu a může pracovat buď jako přijímač nebo vysílač v závislosti na funkci daného zařízení. Zařízení může být buď master nebo slave. Master je zařízení, které iniciuje přenos dat po sběrnici a generuje hodinový signál. V tomto okamžiku jakékoliv adresované zařízení je považováno za slave. Aby sběrnice fungovala, je nutno mít alespoň jeden master na sběrnici.

I<sup>2</sup>C 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.

### 3.6.1. $I^2C$ – vlastnosti

- Dvou vodičové 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  $I^2C$ )
- Čtyři vysílací a čtyři přijímací buffery
- Interrupt nebo Polled mód
- Digitální filtr – SDA a SCL

### 3.6.2. I<sup>2</sup>C – blokový diagram



Obrázek 3.20 I<sup>2</sup>C modul – blokový diagram

## ***4. QUICK START TOOL A GRAPHICAL CONFIGURATION TOOL – VÝVOJOVÉ PROSTŘEDÍ***

DSP56800E\_Quick\_Start je vývojové prostředí, které umožňuje programátorovi rychle a efektivně vytvořit aplikační kód více méně nezávislý na architektuře mikropočítače. Poskytuje softwarovou infrastrukturu pro vytváření kódu, který je snadno přenositelný na jiné platformy. Přenositelnost je samozřejmě omezena funkcí cílového procesoru. Součástí vývojového prostředí Quick\_Start je i tzv. GCT (Graphical Configuration Tool), který podporuje efektivní nastavování chování mikropočítače pomocí propracovaného grafického prostředí, které spolupracuje s prostředím Quick\_Start.

### ***4.1. Quick Start Tool – vlastnosti***

Vývojové prostředí Quick\_Start se skládá z několika součástí:

- Infrastruktura celého systému
- On-chip drivers s jednotným API
- Vzorové aplikace
- GCT (Graphical Configuration Tool)
- Softwarová podpora nástroje FreeMaster

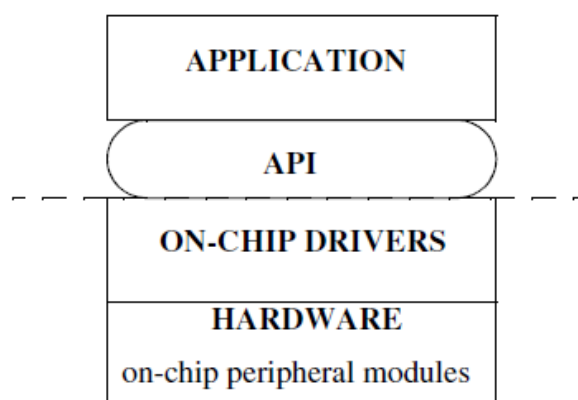
#### ***4.1.1. Infrastruktura systému Quick\_Start***

Infrastruktura systému vytváří propracovanou podporu pro práci s mikropočítači řady 56F8xxx a umožňuje integraci ostatních součástí prostředí Quick\_Start. Součástí infrastruktury je celá řada souborů usnadňujících práci při programování mikropočítače. Mezi infrastrukturu systému Quick\_Start patří: definice maker, přenositelná deklarace registrů, mechanismus pro statickou konfiguraci čipu, tabulka přerušení a obsluha přerušovacích vektorů, start-up kód, soubory pro konfiguraci linkeru, project templates, soubory pro inicializaci debuggeru, atd.



### 4.1.2. *On-chip drivers*

On-chip drivers izolují hardware od softwarové obsluhy. Jasně definované API usnadňuje programování čipu, zvyšuje čitelnost kódu a umožňuje poměrně snadnou přenositelnost výsledného kódu závislého na typu mikropočítače a druhu periferií.



Obrázek 4.1 Softwarová struktura aplikačního kódu

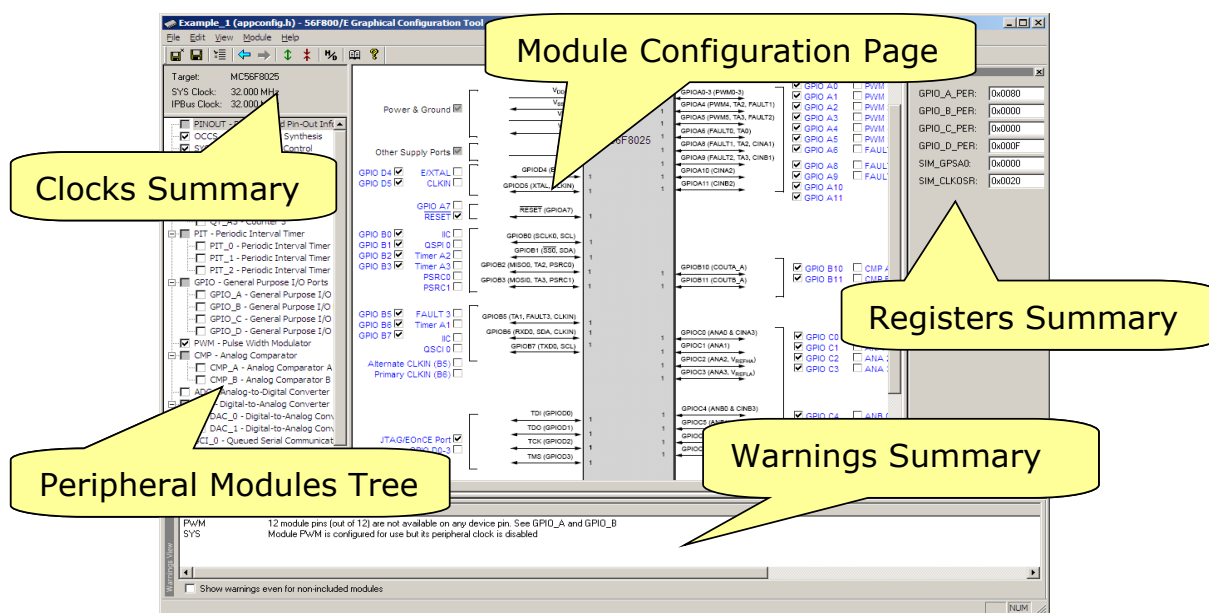
### 4.1.3. *Vzorové aplikace*

Prostředí Quick Start poskytuje uživateli celou řadu příkladů obsluhy typických periferií, na kterých se demonstruje použití on-chip drivers a zároveň se vysvětluje jak implementovat specifická nastavení mikropočítače. Příklady jsou jednoduché a ilustrativní. Jejich účelem je zkrátit dobu učení a tím zefektivnit vývoj uživatelského software.

### 4.1.4. *GCT – Graphical Configuration Tool*

GCT je grafické rozhraní, které uživateli umožňuje pohodlně a srozumitelně provést statické nastavení čipu. Statické nastavení zahrnuje i nastavení interrupt vektorů pro daný zdroj přerušení a jejich obsluhu (uživatelská service routine).

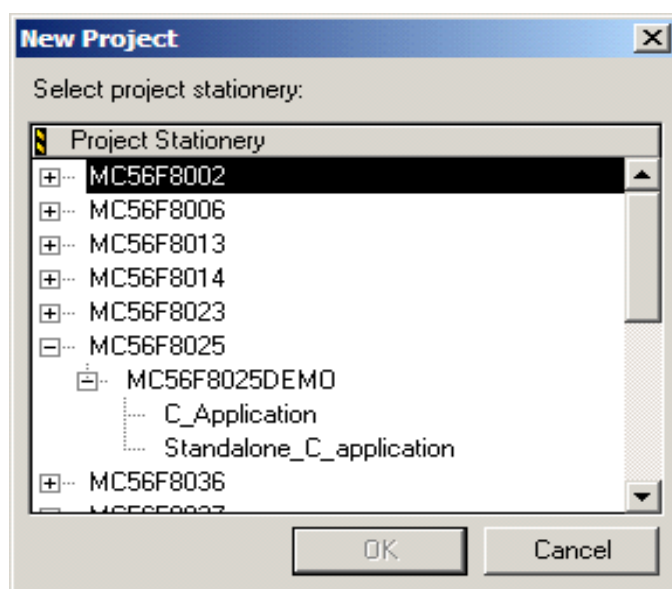
GCT je volitelný nástroj, není bezpodmínečně nutný pro funkci prostředí DSP56800E\_Quick\_Start. Nicméně tento nástroj výrazně zjednodušuje práci s mikropočítačem a zkracuje dobu nutnou na nastavení čipu na minimum. Proto je velmi doporučován.



Obrázek 4.2 GCT – Graphical Configuration Tool

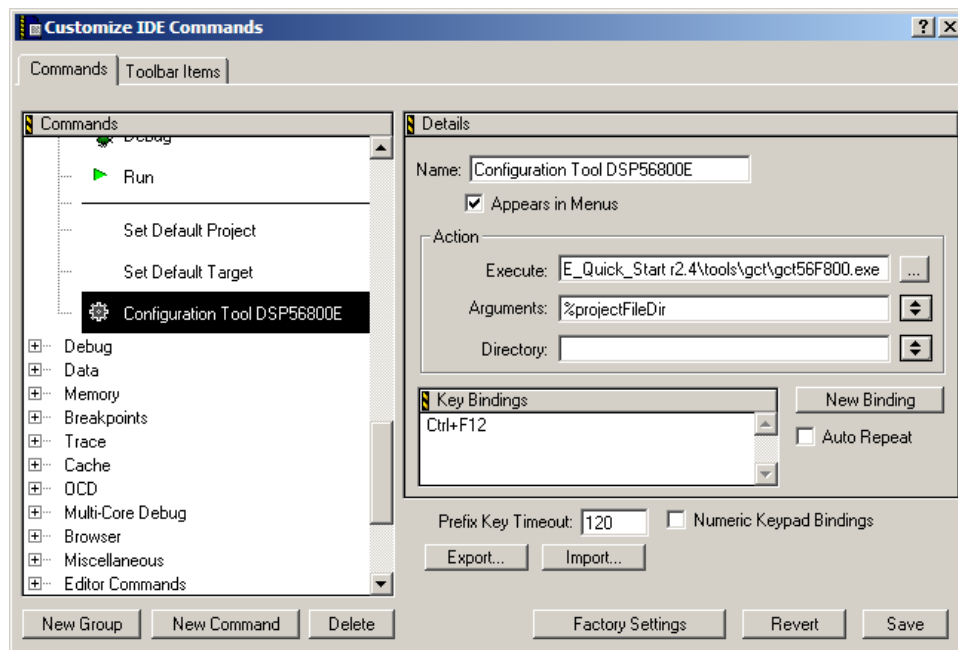
## 4.2. Quick Start Tool – integrace do prostředí Metrowerks CodeWarrior

Jednotlivé komponenty prostředí DSP56800E\_Quick\_Start jsou zakomponovány přímo do programovacího prostředí Metrowerks CodeWarrior. Quick Start poskytuje projekt template (project stationery), který po instalaci nástroje nainstalován do adresářové struktury CodeWarrioru.



Obrázek 4.3 Vytváření nového projektu založeného na Quick Start template

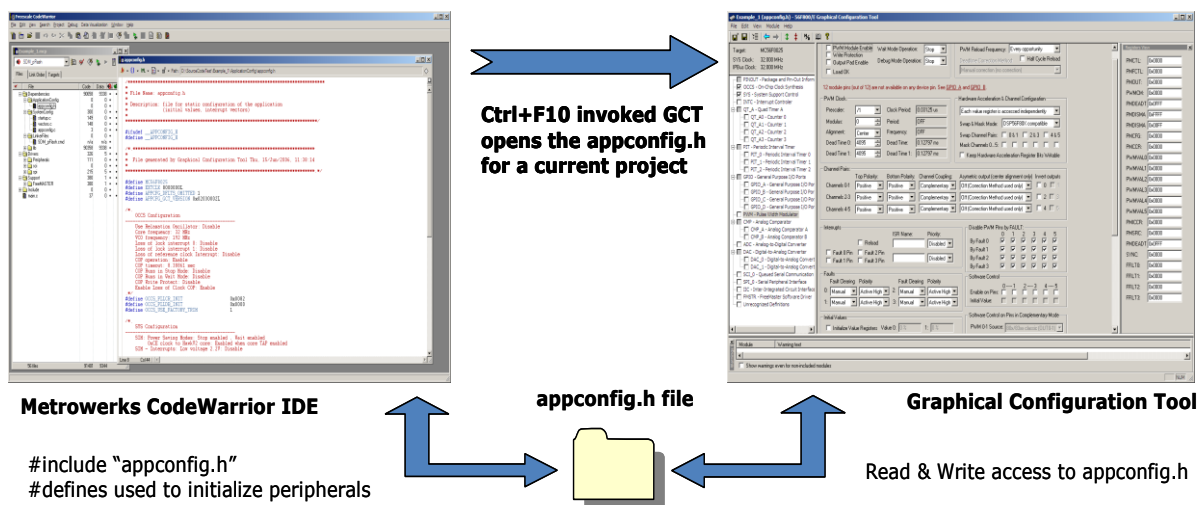
Při vytváření nového projektu je pak automaticky template projektu založený na prostředí Quick Start k dispozici. GCT je přímo volatelný z prostředí CodeWarrior, což v důsledku značně zjednodušuje práci s GCT nástrojem. Nastavení demonstruje Obrázek 4.4.



Obrázek 4.4 Intergace GCT do prostředí Metrowerks CodeWarrior

### 4.3. Spolupráce mezi GCT a nástrojem Quick Start

Spolupráci mezi nástrojem GCT a Quick Start demonstruje následující Obrázek 4.5.



Obrázek 4.5 Spolupráce mezi nástroji GCT a Quick Start

Z obrázku je zřejmé, jak funguje nastavování vlastností čipu pomocí nástroje GCT. Prostředí Metrowerks CodeWarrior z běžícího projektu otevře nástroj GCT. Ten vezme aktuální nastavení uložené v souboru „appconfig.h“ a zobrazí nastavení v grafické podobě. V případě, že „appconfig.h“ neobsahuje nastavení dané periferie, GCT zobrazí defaultní nastavení. V tomto okamžiku lze nastavit jakoukoliv funkčnost dostupnou na čipu prostřednictvím nástroje GCT. Uložení aktuálního nastavení dojde k přepsání souboru „appconfig.h“, který pak uchovává poslední nastavení. Prostředí Quick Start pak pracuje s tímto nastavením v závislosti na aplikaci. Nastavení uložená v souboru „appconfig.h“ je pak možno použít pro nastavení čipu pomocí „ioctl()“ příkazů.

#### 4.4. ArchIO struktura

Jedná se o strukturu definovanou v ANSI C, která obsahuje všechny registry jádra a periférií. Počáteční adresa struktury je pak přiřazena takovým způsobem, aby struktura byla namapovaná přes všechny registry

ArchIO struktura je deklarovaná v souboru arch.h a je zde deklarovaná jako extern proměnná. Adresa proměnné typu struktura je pak přiřazena přímo v souboru „linker command file“.

Prostředí Quick Start využívá této struktury pro přístup ke všem registrům mikropočítače.

Ukázka definice části ArchIO struktury – arch.h file:

```
typedef volatile struct
{
    arch_sTimer    TimerA;    /* TMRA_BASE      0xF000 */
    arch_sTimer    TimerB_unused;
    arch_sADC       Adc;       /* ADC_BASE      0xF080 */
    arch_sPWM       Pwm;       /* PWM_BASE      0xF0C0 */
    arch_sIntc      Intc;      /* INTC_BASE     0xF0E0 */
    arch_sSIM       Sim;       /* SIM_BASE     0xF100 */
```

```

arch_sCOP      Cop;      /* COP_BASE      0xF120 */
arch_sPLL      Pl1;      /* PLL_BASE      0xF130 */
arch_sLVI      Lvi;      /* LVI_BASE      0xF140 */
UWord16        reserved4[0xFF0600];
arch_sEOnCE    EOnCE; /* EOnCE_BASE    0xFFFF00 */
} arch_sIO;

```

arch.h file rovněž obsahuje deklaraci

```
Extern arch_sIO ArchIO
```

Linker command file pak obsahuje následující definici:

```
FArchIO = ADDR(.x_onchip_peripherals);
```

Tímto příkazem je pak struktura přiřazena adresa a můžeme s ní pracovat v rámci prostředí Quick Start a uživatelského kódu.

Příklad použití ArchIO struktury – read/write operace:

```

UWord16 RegValue;      // variable definition

RegValue = ArchIO.TimerA.Channel0.HoldReg;    // read
register

ArchIO.TimerA.Channel0.CompareReg1 = 0x8000;    //
write number to reg

```

## 4.5. ioctl command – Input Output Control

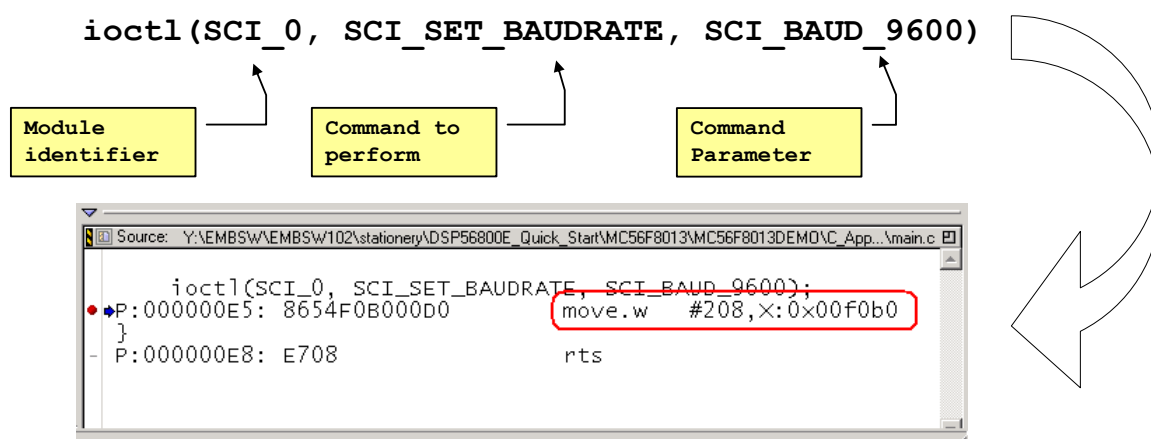
Jedním ze základních pilířů nástroje Quick Start jsou tzv. ioctl commands. Jedná se o low-level drivers, pomocí nichž řídíme funkčnost mikropočítače za běhu programu.

Vlastnosti „ioctl commands“:

- Unifikuje přístup do registrů mikropočítače
- Umožňuje plně řídit všechny zdroje mikropočítače

- Aplikační kód nesáhá na registry přímo, ale pomocí „ioctl commands“
- Volání „ioctl commands“ je kompilátorem překládáno optimálně
- Zvyšuje čitelnost kódu
- Zjednodušuje přenositelnost aplikace

Formát „icotl command“ je následující:



**Obrázek 4.6** ioctl command a jeho zpracování překladačem

## 5. FREEMASTER TOOL – VÝVOJOVÉ PROSTŘEDÍ

FreeMASTER tool je neocenitelný nástroj při vývoji aplikací běžících v reálném čase. FreeMASTER tool poskytuje uživateli celou řadu vlastností, které velmi výrazně urychlí čas potřebný pro vývoj komplexní aplikace. Je hodnocen zákazníky jako vynikající nástroj, který již po zběžném osahání nechtějí dát z ruky.

### 5.1. FreeMASTER tool – vlastnosti

FreeMASTER tool slouží především jako výkonný ladící nástroj v reálném čase. Jeho možnosti se v průběhu času neustále zlepšovaly a stále nové a nové vlastnosti byly dodávány. V současné době nástroj může sloužit jako:

- Monitor v reálném čase
- Grafický řídicí panel
- Platforma vhodná k demonstračním účelům
- Nástroj podporující prodej mikropočítačů

#### 5.1.1. *FreeMASTER tool – monitor v reálném čase*

FreeMASTER tool umožňuje přístup do libovolného paměťového místa s pevnou adresou známou při překladu a linkování. To znamená, že nepodporuje monitorování proměnných na zásobníku. FreeMASTER tool po linkování dokáže analyzovat ELF soubor (Executable File), analyzuje DWARF ladící informace obsažené v ELF souboru. Na základě této analýzy pak zná adresy globálních a statických proměnných. Zná rovněž jejich velikost, typ, velikost pole a jména, takže umožňuje uživateli velice jednoduše přistupovat k proměnným definovaným v aplikačním kódu.

FreeMASTER tool má dvě strany

- Embedded strana – běží na procesoru
- PC strana – běží na PC v prostředí Windows

Aby bylo možné spolupráce obou stran FreeMASTER toolu je zapotřebí mít komunikační kanál, kterým si obě strany budou předávat data. V tomto ohledu FreeMASTER tool podporuje všechny možné komunikační kanály dostupné na daném čipu.

- SCI, UART
- JTAG/EonCE (56F8xxx)
- BDM (HCS08, HCS12)
- CAN Calibration Protocol

FreeMASTER tool dokáže zobrazovat proměnné flexibilním způsobem v závislosti na nastavení. Možnosti jsou následující:

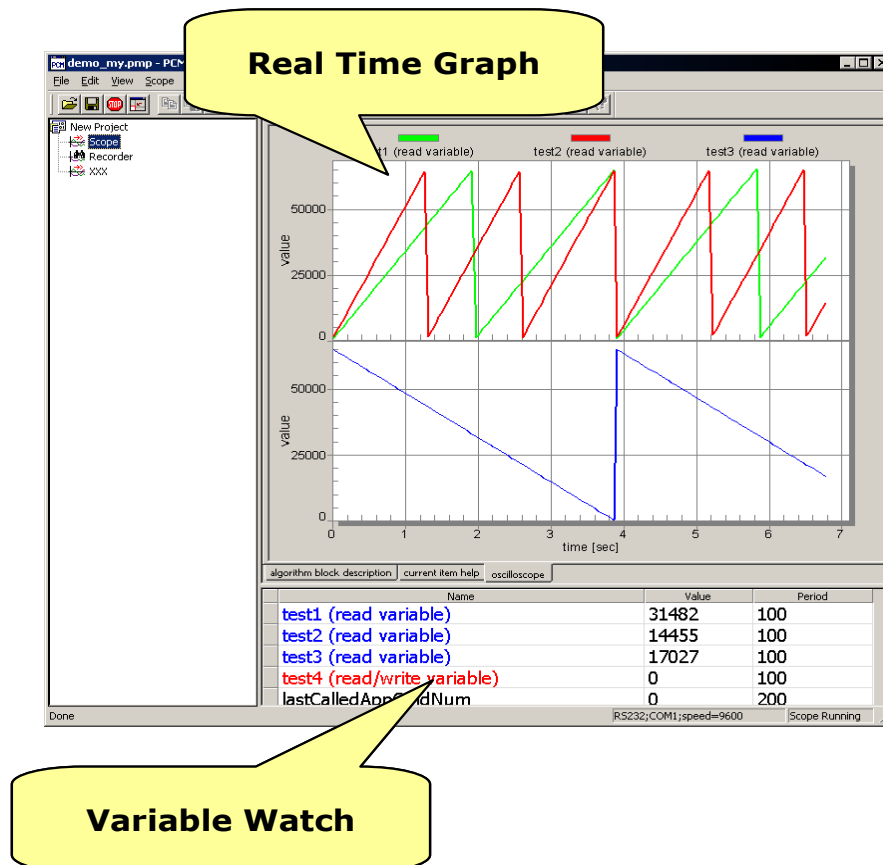
- Text
  - Zobrazí jméno proměnné
  - Hodnotu v různých formátech (hex, dec nebo bin)
  - Minimální a maximální hodnotu během monitorování
- Průběhy v reálném čase – režim „oscilloscope“
  - Zobrazení až 8 proměnných zároveň v režimu „oscilloscope“
  - Grafická podoba odpovídá grafu s časovou osou
- Vysokorychlostní záznam dat – režim „recorder“
  - Zobrazení až 8 proměnných zároveň v režimu „recorder“
  - Poskytuje přesné vzorkování
  - Data dočasně uložena v paměti procesoru – data buffer
- Transformace proměnných
  - Možnost transformace dat z formátu procesoru do uživatelsky přijatelného zobrazení – např.: místo fractional hodnot můžeme pracovat přímo s napětím ve [V], proudem v [A], otáčkami v [rad/s],



s teplotou ve [°C], atd. Transformace funguje obousměrně a to jak z formátu procesoru do formátu skutečné veličiny, tak i naopak, což nám umožňuje manipulovat s veličinami programu v jejich přirozených fyzikálních jednotkách

- Možnost ochrany části paměti proti nechtěnému přepsání
- Definice proměnných, které jsou pro FreeMASTER viditelné
- Deklarace proměnných jako read-write nebo jen read-only. Přístup k těmto proměnným je garantovaný driverem na embedded straně
- Aplikační komandy

Ukázka typického zobrazení FreeMASTERu:



Obrázek 5.1 FreeMASTER – typické zobrazení

### 5.1.2. FreeMASTER tool – grafický řídící panel

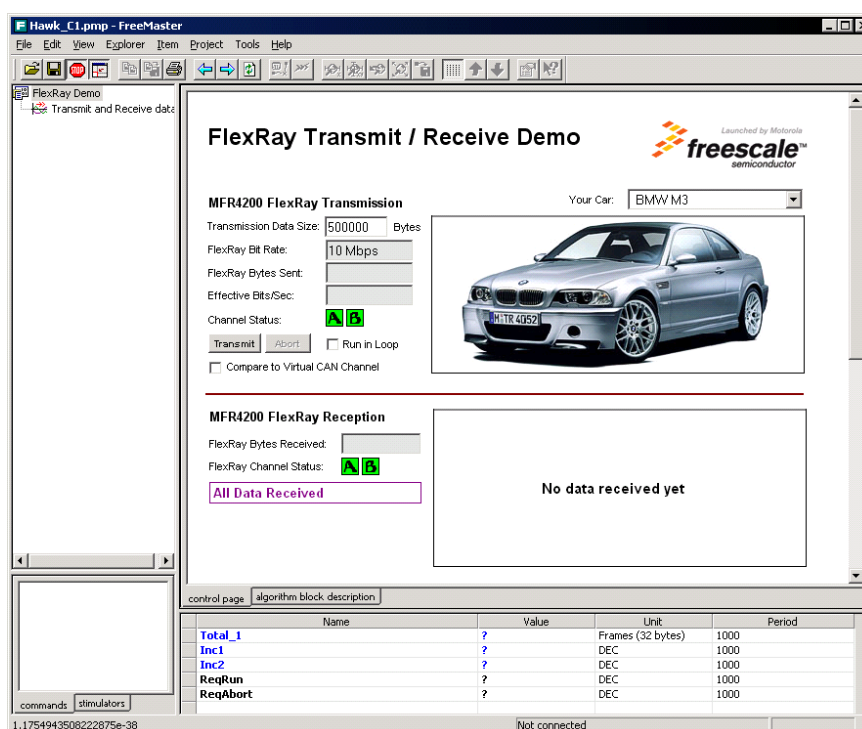
FreeMASTER PC strana umožňuje kromě standardní manipulace s proměnnými i poněkud komplikovanější a zobrazení a modifikaci dat. FreeMASTER tool umožňuje vytvářet uživatelské grafické rozhraní vytvářením HTML stránek, spouštět skripty (JScript nebo VBScript), kreslit, animovat prvky jako tlačítka, obrázky, různé indikátory, spouštět zvuk, přehrávat video, atd.

HTML stránky jsou zobrazeny přímo v prostředí FreeMASTER. HTML stránky mohou obsahovat skripty a ActiveX komponenty.

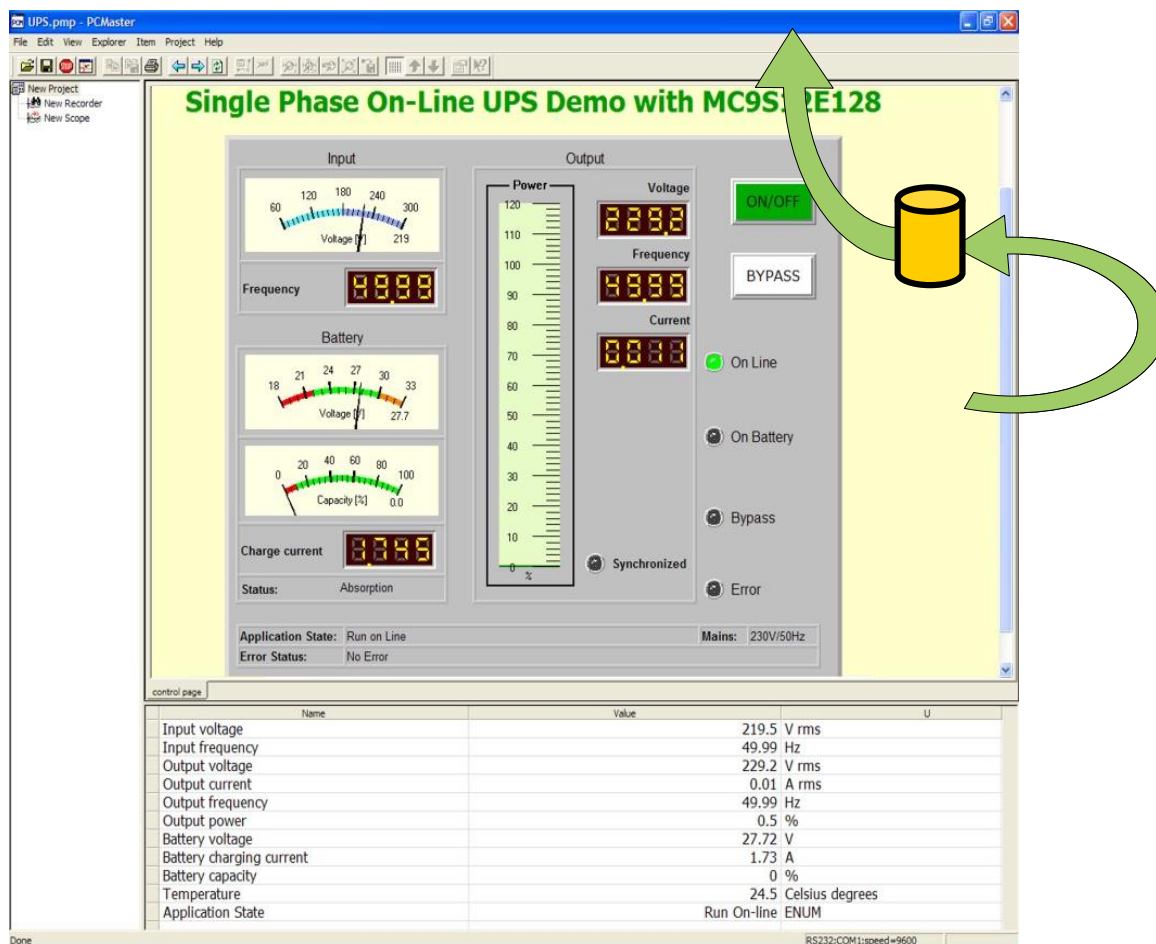
HTML může také spouštět celé aplikace, např.: Excel.

FreeMASTER samotný je implementován jako neviditelný ActiveX objekt, což umožňuje volat FreeMASTER z externích programů jako např.: C++ aplikace nebo VB aplikace, Excel & Visual Basic nebo Matlab/Simulink.

Ukázky možností použití FreeMASTER toolů jsou zobrazeny na následujících obrázcích.



Obrázek 5.2 FreeMASTER – HTML



**Obrázek 5.3 FreeMASTER – HTML**

FreeMASTER rovněž funguje jako web-browser, což umožňuje navigovat na webovské stránky bez opuštění FreeMASTER prostředí.

Zajímavá vlastnost FreeMASTER toolu je možnost komunikovat po ethernetu. Tato vlastnost je velmi užitečná v případě, že ladíme aplikaci v hlučném nebo jinak nebezpečném prostředí. Pak lze procesorovou desku propojit s FreeMASTER toolem přes libovolný komunikační kanál přímo v laboratoři a spustit další instanci FreeMASTER toolu někde úplně jinde a komunikovat mezi nimi prostřednictvím ethernetu.

## 6. FREESCALE LIBRARY

Freescale poskytuje volně šiřitelné knihovny funkcí. Algoritmy jsou členěné do oddělených knihoven podle typu implementovaných algoritmů. Algoritmy jsou napsány v assembleru a jejich interface vypadá jako standardní C funkce. Dodaná knihovna obsahuje \*.lib file, \*.h file a dokumentaci v pdf formátu. Algoritmy jsou rozděleny do čtyř knihoven:

- MCLIB – 56800E\_MCLIB.lib
- GFLIB - 56800E\_GFLIB.lib
- GDFLIB - 56800E\_GDFLIB.lib
- ACLIB - 56800E\_ACLIB.lib

### 6.1. Freescale library – MCLIB

Knihovna MCLIB obsahuje algoritmy optimalizované pro řízení elektrických pohonů různých typů od stejnosměrného motoru, přes asynchronní motor, synchronní motor s permanentními magnety až po spínaný reluktanční motor.

Použití funkcí je snadné. Dokumentace obsahuje popis API, popis algoritmu, popis vstupů a výstupů včetně rozsahů a ke každé funkci je uveden příklad

Seznam funkcí v MCLIB knihovně:

- MCLIB\_ClarkTfr – Clarkova transformace
- MCLIB\_ClarkTfrInv – inverzní Clarkova transformace
- MCLIB\_ParkTfr – Parkova transformace
- MCLIB\_ParkTfrInv – inverzní Parkova transformace
- MCLIB\_SvmStd – SVM (Space Vector Modulation)
- MCLIB\_SvmU0n - SVM (Space Vector Modulation)
- MCLIB\_SvmU7n - SVM (Space Vector Modulation)
- MCLIB\_SvmAlt - SVM (Space Vector Modulation)

- `MCLIB_SvmSci` - SVM (Space Vector Modulation)
- `MCLIB_PwmIct` – obecná sinusová modulace
- `MCLIB_DecouplingPMSM` – eliminace cross-coupling napětí v rotačním souřadném systému, odstranění nelinearity řízení
- `MCLIB_ElimDcBusRip` – transformace vypočtených napětí v závislosti na aktuální velikosti napětí na mezilehlém obvodu a eliminace zvlnění
- `MCLIB_VectorLimit` – limitace napěťového vektoru

## 6.2. Freescale library – GFLIB

Tato knihovna je nejrozsáhlejší a obsahuje celou řadu užitečných matematických funkcí optimalizovaných v assembleru. Výčet funkcí je rozsáhlý. Seznam a detailní popis algoritmů můžete najít v dokumentaci ke knihovně.

## 6.3. Freescale library – GDFLIB

GDFLIB obsahuje několik algoritmů realizovaných digitálních filtrů. Kód je napsaný v assembleru. Knihovna obsahuje následující algoritmy:

- `GDFLIB_FilterIIR1` – filter IIR prvního řádu
- `GDFLIB_FilterIIR1Init` – inicializační funkce filtru IIR prvního řádu
- `GDFLIB_FilterIIR2` - filter IIR druhého řádu
- `GDFLIB_FilterIIR2Init` - inicializační funkce filtru IIR druhého řádu
- `GDFLIB_FilterMA32` – rekurzivní forma moving-average filtru
- `GDFLIB_FilterMA32Init` - inicializační funkce rekurzivní formy moving-average filtru

## 6.4. Freescale library – ACLIB

Tato knihovna je zaměřená na pokročilé techniky řízení synchronních motorů s permanentními magnety bez použití sensoru polohy tzv. „sensorless control“.

Použití těchto funkcí vyžaduje poměrně rozsáhlé znalosti v oblasti bezsnímačového řízení pohonů. Knihovna obsahuje:

- `ACLIB_PMSMBemfObsrvAB` – pozorovatel odhadující indukované napětí ve stojících  $\alpha$ ,  $\beta$  souřadnicích
- `ACLIB_PMSMBemfObsrvDQ` – pozorovatel odhadující indukované napětí v rotujících souřadnicích d,q
- `ACLIB_AngleTrackingObsrv` – estimátor uhlové rychlosti a polohy vstupního sinusového signálu
- `ACLIB_TrackingObsrv` - estimátor uhlové rychlosti a polohy vstupního signálu
- `ACLIB_Integrator` - numerická integrace

## SEZNAM POUŽITÉ LITERATURY

- [1] MC56F8006RM.pdf - 56F8006 Peripheral Reference Manual
- [2] MC56F8006.pdf - 56F8006/56F8002 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] Zdeněk Caha, Miroslav Černý: Elektrické pohony. Praha, SNTL 1990

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](mailto:info@crr.vutbr.cz)