

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Mikroprocesorová technika v embedded systémech

Učební texty k semináři

Autoři:

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

Datum:

27. 5. – 28. 5. 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. DSP56800E – Jádro Freescale DSC embedded mikropočítačů	5
1.1. Architektura jádra	5
1.2. Adresové sběrnice	7
1.3. Datové sběrnice	7
1.4. Dvojitá Harvardská struktura – paměť	8
1.5. Architektura mikropočítače a periferní rozhraní	9
1.6. Datová aritmeticko logická jednotka - ALU	11
1.6.1. Akumulátory	12
1.6.2. MAC jednotka a logická jednotka.....	13
1.7. Fractional a Integer aritmetika.....	14
1.7.1. DSP56800E – datové typy	14
1.7.2. Signed Integer.....	15
1.7.3. Unsigned Integer	16
1.7.4. Signed Fractional	16
1.7.5. Unsigned Fractional	16
1.7.6. Fractional a Integer data – porovnání	17
1.7.7. Násobení.....	18
1.8. Adresovací jednotka – AGU.....	18
1.8.1. Architektura adresovací jednotky	18
1.8.2. AGU – programovací model.....	19
1.9. Programový řadič – program controller	20
1.9.1. Architektura programového řadiče.....	20
1.9.2. Programový řadič – programovací model	21
1.10. Bit manipulation unit	22
1.10.1. Bit manipulation unit - architektura.....	23

1.11.	Instruction pipeline	24
1.12.	JTAG a Enhanced On-Chip Emulation (Enhanced OnCE).....	25
2.	Embedded mikropočítač Freescale dsc56F8025	26
2.1.	Rodina mikropočítačů Freescale DSC56F80xx	26
2.2.	DSC56F8025/56F8035 – základní parametry.....	27
2.3.	Organizace Freescale dokumentace.....	29
2.4.	ADC – Analog Digital Converter.....	30
2.4.1.	ADC - vlastnosti.....	30
2.4.2.	ADC – blokový diagram.....	31
2.4.3.	ADC – zpracování dat převodníkem	31
2.5.	COP – Computer Operating Properly	32
2.5.1.	COP – vlastnosti.....	33
2.5.2.	COP – blokový diagram	33
2.6.	PWM – Pulse Width Modulator	34
2.6.1.	PWM – vlastnosti.....	34
2.6.2.	PWM – blokový diagram.....	35
2.7.	OCCS – On-Chip Clock Synthesis.....	36
2.7.1.	OCCS – vlastnosti	36
2.7.2.	OCCS – blokový diagram	37
2.8.	FM – Flash Memory	37
2.8.1.	FM – vlastnosti	38
2.8.2.	FM – blokový diagram	39
2.9.	PIT – Programmable Interval Timer.....	40
2.9.1.	PIT – vlastnosti.....	40
2.9.2.	PIT – Blokový diagram.....	41
2.10.	TMR – Quad Timer	41
2.10.1.	TMR – vlastnosti	42
2.10.2.	TMR – blokový diagram	43

2.11.	QSCI – Qued Serial Communication Interface	43
2.11.1.	QSCI – vlastnosti	43
2.11.2.	QSCI – blokový diagram	45
2.11.3.	QSCI – single-wire komunikace	45
2.12.	QSPI – Qued Serial Peripheral Interface	46
2.12.1.	QSPI – vlastnosti	46
2.12.2.	QSPI – blokový diagram	47
2.12.3.	QSPI – master mód	47
2.12.4.	QSPI – wired-OR mód	48
2.13.	DAC – Digital-to-Analog Converter	48
2.13.1.	DAC – vlastnosti	48
2.13.2.	DAC – blokový diagram	49
2.14.	CMP – Comparator	49
2.14.1.	CMP – vlastnosti	49
2.14.2.	CMP – blokový diagram	50
2.15.	GPIO – General Purpose Input/Output	50
2.15.1.	GPIO – vlastnosti	51
2.15.2.	GPIO – blokový diagram	52
2.16.	I ² C – Inter-Integrated Circuit Interface	52
2.16.1.	I ² C – vlastnosti	53
2.16.2.	I ² C – blokový diagram	54
3.	<i>Quick Start Tool A Graphical Configuration Tool – vývojové prostředí</i>	55
3.1.	Quick Start Tool – vlastnosti	55
3.1.1.	Infrastruktura systému Quick_Start	55
3.1.2.	On-chip drivers	56
3.1.3.	Vzorové aplikace	56
3.1.4.	GCT – Graphical Configuration Tool	56
3.2.	Quick Start Tool – integrace do prostředí Metrowerks CodeWarrior	57

3.3.	Spolupráce mezi GCT a nástrojem Quick Start	58
3.4.	ArchIO struktura	59
3.5.	ioctl command – Input Output Control	60
4.	<i>FreeMASTER Tool – vývojové prostředí</i>	62
4.1.	FreeMASTER tool – vlastnosti	62
4.1.1.	FreeMASTER tool – monitor v reálném čase	62
4.1.2.	FreeMASTER tool – grafický řídicí panel	65
5.	<i>Freescale library</i>	67
5.1.	Freescale library – MCLIB	67
5.2.	Freescale library – GFLIB	68
5.3.	Freescale library – GDFLIB	68
5.4.	Freescale library – ACLIB	69
	Seznam použité literatury	70

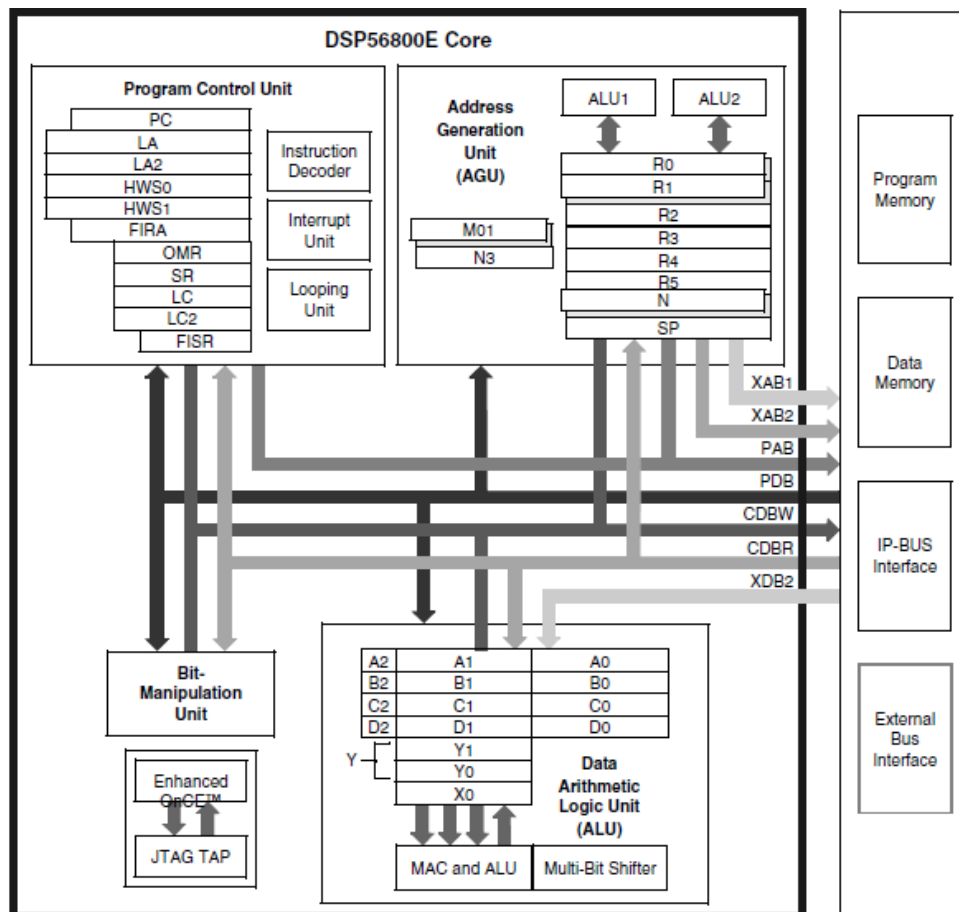
1. DSP56800E – JÁDRO FREESCALE DSC EMBEDDED MIKROPOČÍTAČŮ

Jádro DSP56800E kombinuje výpočetní výkon signálových procesorů DSP s výhodami klasického MCU. Jedná se o optimalizovanou variantu jádra z hlediska ceny, spotřeby a výpočetního výkonu. Jádro DSP56800E poskytuje poměrně komplexní instrukční sadu, která ve spojení s MAC (multiply-accumulace) jednotkou, rozsáhlou sadou registrů, výkonnou adresovací jednotkou, jednotkou umožňující manipulaci s jednotlivými bity a efektivním sběrnicovým systémem, je optimálním řešením pro rodinu číslicových signálových procesorů DSC (Digital Signal Controller) zaměřených na embedded aplikace v průmyslu, spotřební elektronice, bílé technice, atd.

1.1. Architektura jádra

Architektura jádra procesoru DSP56800E má dvojitou Harwardskou strukturu s oddělenými paměťovými prostory programu a dat. Tato architektura umožňuje současný přístup do programu dat a paměti. Rozhraní datové paměti rovněž umožňuje dva současné přístupy pro čtení do této paměti.

Jádro DSP56800E se skládá z aritmeticko logické jednotky (ALU), adresovací jednotky (AGU), řídicí jednotky, jednotky umožňující efektivní manipulaci s bity (bit-manipulation unit), ladicího modulu (Enhanced On-Chip Emulation module) a zpracovaného sběrnicového systému.



Obr. 1.1: Blokový diagram jádra DSP56800E

IP-BUS rozhraní podporuje efektivní propojení jádra s celou řadou různých periferních modulů. Příkladem periférií mohou být:

- COP (Computer Operating Properly)
- GPIO (General Purpose Input Output)
- ADC (Analog-to-Digital Converter)
- PWM (Pulse Width Modulator)
- DAC (Digital-to-Analog Converter)
- SCI (Serial Communication Interface)
- SPI (Serial Peripheral Interface)
- Modul časovačů a čítačů
- atd.

1.2. Adresové sběrnice

Jádro obsahuje tři adresové sběrnice:

- Programová adresová sběrnice – PAB (Program Memory Address Bus)
- Primární datová adresová sběrnice – XAB1 (Primary Data Address Bus)
- Sekundární datová adresová sběrnice – XAB2 (Secondary Data Address Bus)

Programová adresová sběrnice (PAB) má šířku 21-bitů a je určena k adresování programu v programové paměti. PAB dokáže adresovat 16-bitová slova.

Obě datové adresové sběrnice jsou 24-bitové a umožňují dva současné přístupy do datové (X) paměti.

Primární datová adresová sběrnice (XAB1) umožňuje adresovat byte (8-bit), word (16-bit) a long (32-bit) data.

Sekundární datová adresová sběrnice (XAB2) je limitována a umožňuje adresovat pouze word (16-bit) data.

Všechny tři sběrnice dovolují adresovat jak paměť na čipu, tak i externí paměť.

1.3. Datové sběrnice

Přenos dat uvnitř čipu je zajištěn prostřednictvím následujících sběrnic:

- Dvě jednosměrné 32-bitové sběrnice
 - Datová sběrnice jádra pro čtení – CDBR (Core Data Bus for Reads)
 - Datová sběrnice jádra pro zápis – CDBW (Core Data Bus for Writes)
- Dvě jednosměrné 16-bitové sběrnice
 - Sekundární datová sběrnice dat – XDB2 (Secondary X Data Bus)
 - Programová datová sběrnice – PDB (Program Data Bus)
- IP-BUS interface

Data jsou předávána mezi ALU a datovou pamětí prostřednictvím datových sběrnic CDBR a CDBW v případě vykonávání instrukce vyžadující pouze jeden přístup do datové paměti a to buď čtení, nebo zápis. V případě vykonávání instrukce vyžadující dvě současná čtení z datové paměti, je přenos dat vykonán prostřednictvím datových sběrnic CDBR a XDB2. Všechny ostatní přenosy dat do jednotlivých bloků jádra se provádí prostřednictvím sběrnic CDBR a CDBW.

Přenos dat mezi periferními moduly mikropočítače je zajištěn prostřednictvím IP-BUS rozhraní.

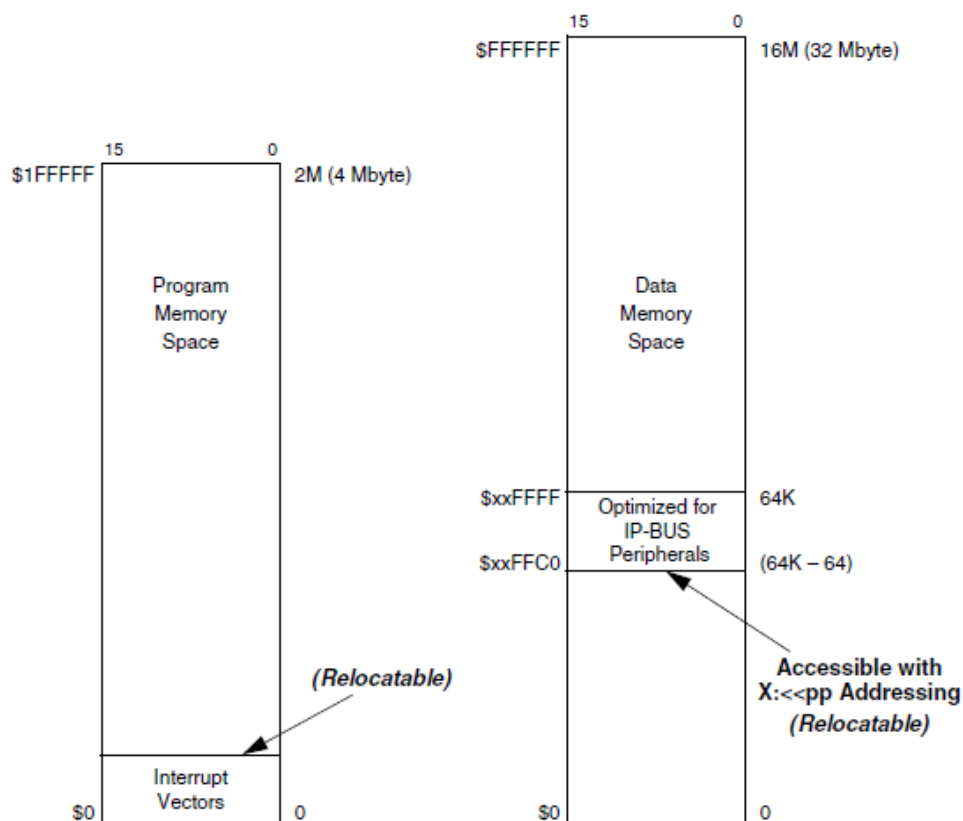
Instrukce programu je přenášena prostřednictvím sběrnice PDB

Tato sběrnice struktura podporuje až tři zároveň probíhající 16-bitové přenosy. Během jednoho hodinového cyklu může nastat libovolná z následujících kombinací:

- Jeden přenos instrukce
- Jedno čtení z datové paměti
- Jeden zápis do datové paměti
- Dvě čtení z datové paměti
- Jeden přenos instrukce a jedno čtení z datové paměti
- Jeden přenos instrukce a jeden zápis do datové paměti
- Jeden přenos instrukce a dvě čtení z datové paměti

1.4. Dvojitá Harvardská struktura – paměť

Jádro procesoru DSP56800E s dvojitou Harvardskou strukturou má oddělené paměťové prostory programu a dat. Tato architektura umožňuje současný přístup do programu dat a paměti. Rozhraní datové paměti rovněž umožňuje dva současné přístupy pro čtení z této paměti.



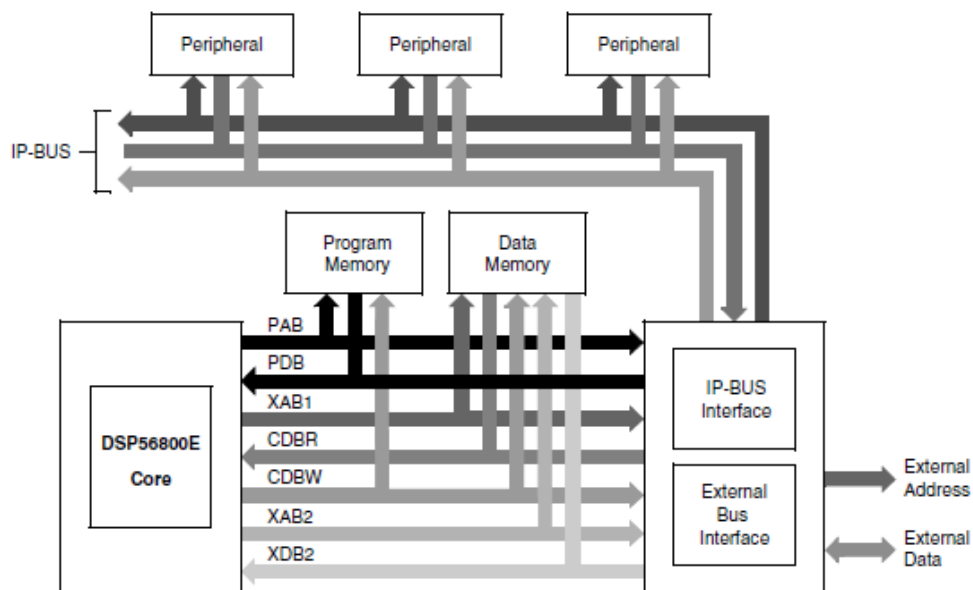
Obr. 1.2: Architektura paměti jádra DSP56800E

Jelikož adresová sběrnice PAB programové oblasti je 21-bitů, lze adresovat paměťový prostor 2 MWord (4 Mbyte). Blok paměti obsahující reset a vektory přerušení může mít libovolnou velikost a je relokovatelný kdekoli v rámci programové paměti.

Obě adresové sběrnice XAB1 a XAB2 datové oblasti mají šířku 24-bitů, což umožňuje adresovat 16 Mword (32 MByte) datového prostoru. Registry periférií jsou mapovány do datové oblasti a celý blok může být přesunutelný v rámci datového prostoru.

1.5. Architektura mikropočítače a periferní rozhraní

Mikropočítač na bázi jádra DSP56800E obsahuje komponenty zahrnující samotné jádro, interní paměť, periferní moduly a sběrnice, které jsou nezbytné k propojení jednotlivých modulů celého systému.



Obr. 1.3: Architektura DSC mikropočítačů na bázi jádra DSP56800E s externími sběrnici

Architektura DSC mikropočítačů na bázi jádra DSP56800E zahrnuje následující komponenty:

- Jádro DSP56800E
- Interní paměť programu
- Interní paměť dat
- Periferní moduly na čipu
- Freescale IP-BUS periferní rozhraní
- Externí sběrnice rozhraní

Některé součástky nemají implementované externí sběrnice rozhraní. Jedná se o případy, kdy jsou součástky distribuovány v malých pouzdech s malým počtem pinů.

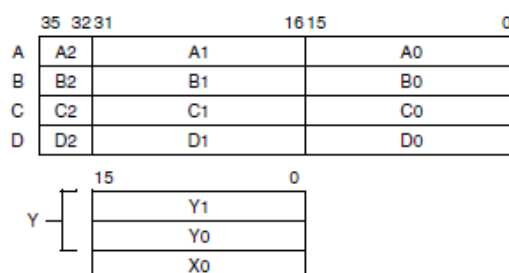
Všechny periférie vždy komunikují s jádrem mikropočítače prostřednictvím IP-BUS periferního rozhraní. IP-BUS periferní rozhraní je napojeno na dvě adresové sběrnice XAB1 a XAB2 umožňující adresovat datový prostor a tři jednosměrné datové sběrnice CDBR, CDBW a XDB2. Sběrnice připojující programovou paměť nejsou připojeny k perifériím.

1.6. Datová aritmeticko logická jednotka - ALU

Hlavní součástí ALU jsou:

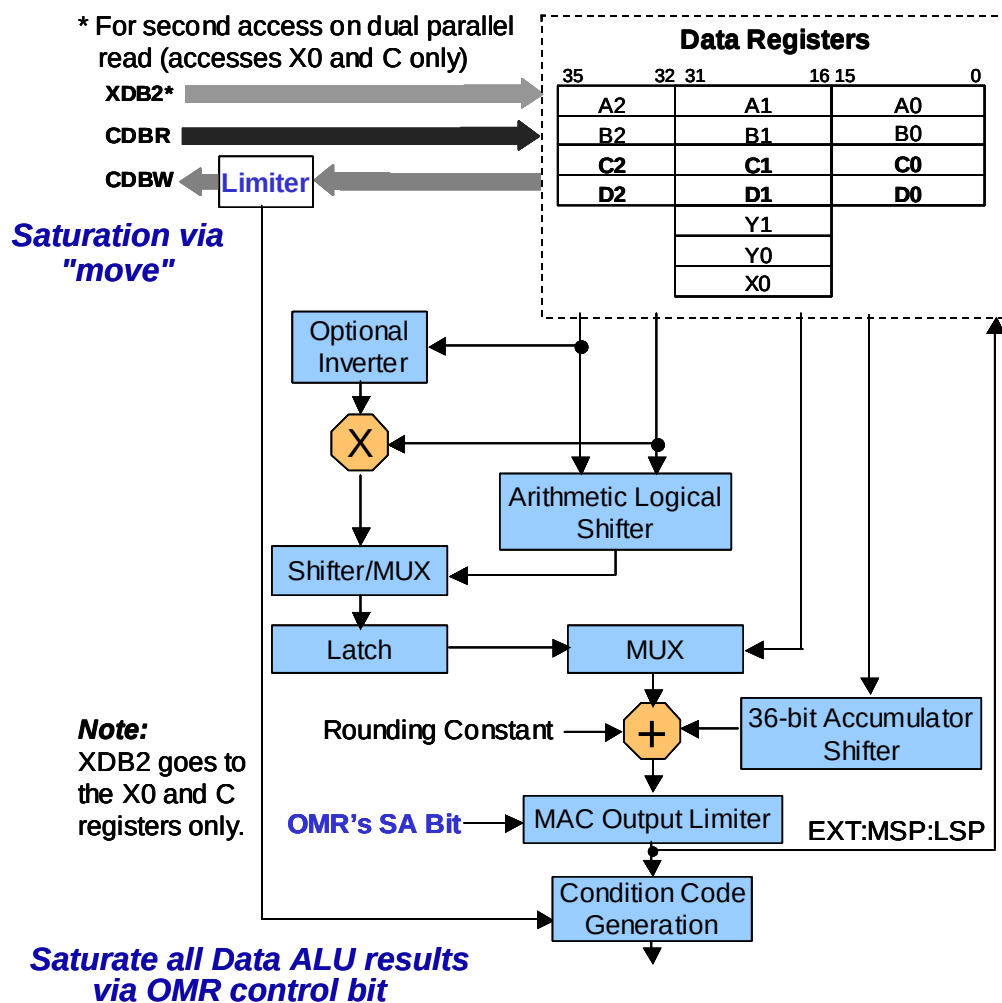
- Tři 16-bitové datové registry – X0, Y0 a Y1, přičemž registry Y0 a Y1 lze spojit do jednoho 32-bitového registru Y
- Čtyři 36-bitové akumulátory – A, B, C a D
- Jednocyklovou MAC jednotku
- Jedobitový akumulátor shifter
- Aritmetický a logický multi-bit shifter
- MAC output limiter
- Data limiter.

Programovací model datové ALU je na Obr. 1.4.



Obr. 1.4: Programovací model datové ALU

Blokový diagram data ALU je zobrazen na následujícím Obr. 1.5.

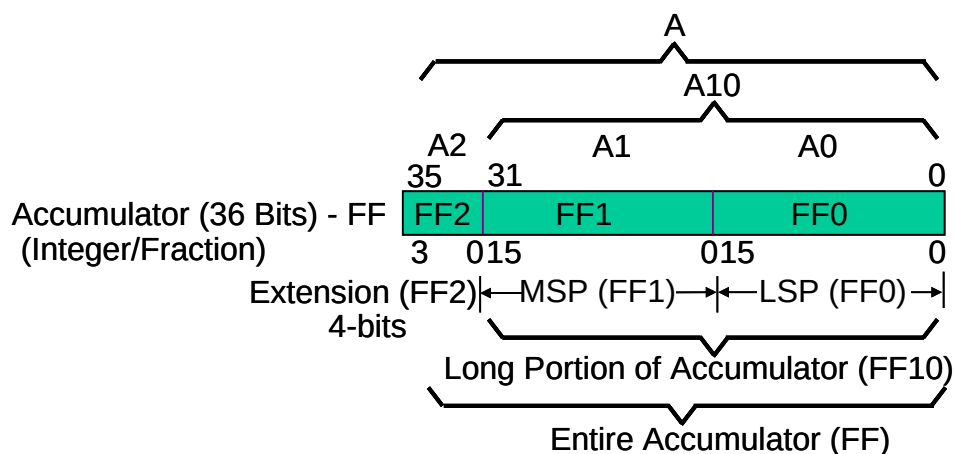


Obr. 1.5: Blokový diagram data ALU

1.6.1. Akumulátory

Data ALU obsahuje čtyři nezávislé 36-bitové registry zvané akumulátory, které slouží jako zdrojové a cílové registry pro operace prováděné na datové ALU.

Obr.1.6 demonstruje dělení akumulátoru na jednotlivé části



Obr.1.6: Architektura akumulátoru

Akumulátory datové ALU jsou složeny ze třech různých částí:

- Extension registr FF2 – 4-bity (FF2 reprezentuje A2, B2, C2 nebo D2)
- MSP část (Most Significant Product) FF1 – 16-bitů (FF1 reprezentuje A1, B1, C1 nebo D1)
- LSP část (Least Significant Product) FF0 – 16-bitů (A0, B0, C0 nebo D0)

K akumulátoru lze rovněž přistupovat jako k 32-bitovému slovu prostřednictvím FF10 notace (FF10 reprezentuje A10, B10, C10, D10).

1.6.2. *MAC jednotka a logická jednotka*

MAC (Multiply-Accumulator) jednotka a logická jednotka je základní jednotka ALU. Tento modul vykonává násobení, sčítání, odečítání, logické operace a ostatní aritmetické operace. Vstupem do MAC jednotky může být libovolný ze sedmi data ALU registrů (A1, B1, C1, D1, X0, Y0 a Y1), paměťové místo nebo přímá data. Volitelná saturace a zaokrouhlování zajišťuje správnou funkčnost implementovaného kódu. Aritmetické operace prováděné MAC jednotkou pracují nezávisle a paralelně s přístupem do paměti před core datové sběrnice. Tato schopnost umožňuje paralelní přenosy dat, zatímco akumulátory jsou využívány pro ALU operace

1.7. Fractional a Integer aritmetika

Jádro procesoru DSP56800E má optimalizovanou instrukční sadu, která podporuje provádění numerických výpočtů ve fractional aritmetice. Jádro rovněž poskytuje instrukce pro výpočty v integer aritmetice, ovšem podpora fractional aritmetiky je mnohem výraznější a její použití vede k efektivnějšímu a optimálnějšímu kódu v porovnání s integer aritmetikou.

Fractional aritmetiku lze s výhodou využít při matematických výpočtech jako jsou např. digitální filtry, maticové výpočty, digitální řídicí algoritmy, zpracování signálů a v podstatě pro všechny matematické výpočty v případě vhodné normalizace realizovaných rovnic.

1.7.1. *DSP56800E – datové typy*

Architektura jádra DSP56800E podporuje integer data velikosti byte (8-bit), word (16-bit) a long-word (32-bit) stejně jako fractional datové typy word, long-word a akumulátor (36-bit).

Jádro podporuje čtyři základní datové typy bez ohledu na velikost dat:

- Signed integer
- Unsigned integer
- Signed fractional
- Unsigned fractional

Seznam datových typů a jejich rozsah je uveden v Tab. 1.1.

Datové typy	Minimální hodnota	Maximální hodnota
Integer		
Unsigned byte	0	255
Signed byte	-128	127
Unsigned word	0	65535
Signed word	-32768	32767
Unsigned long	0	4 294 967 295
Signed long	-2 147 483 648	2 147 483 647
Fractional ¹		
Signed word	-1.0	0.999 969 482 4
Signed long word	-1.0	0.999 999 999 5
Signed 36-bit accumulator	-16.0	15.999 999 999 5

¹. Všechny fractional hodnoty jsou zaokrouhleny na 10 desetinných míst

Tab. 1.1: Datové typy a rozsah hodnot

1.7.2. *Signed Integer*

V tomto formátu je N-bitový operand reprezentovaný jako N.0 formát, kde N je počet bitů. Rozsah signed integer čísel je následující:

$$-2^{[N-1]} \leq SI \leq 2^{[N-1]} - 1$$

Tento datový formát je k dispozici pro velikosti dat typu byte, word a long word. V případě velikosti word je rozsah:

$$-32768 (0x8000) \leq SI \leq 32767 (0x7FFF)$$

V případě velikosti long word

$$-2\,147\,483\,648 (\$80000000) \leq SI \leq 2\,147\,483\,647 (0x7FFFFFFF)$$

1.7.3. *Unsigned Integer*

Čísla ve formátu unsigned integer jsou pouze kladná nebo rovna nule. V tomto formátu se N-bitový operand reprezentovaný jako N.0 formát, kde N je počet bitů. Rozsah signed integer čísel je následující:

$$0 \leq SI \leq 2^N - 1$$

Tento datový formát je k dispozici pro velikosti dat typu byte, word a long word. V případě velikosti word je rozsah:

$$0 \text{ (0x0000)} \leq SI \leq 65535 \text{ (0xFFFF)}$$

V případě velikosti long word

$$0 \text{ (0x00000000)} \leq SI \leq 4\,294\,967\,295 \text{ (0xFFFFFFFF)}$$

1.7.4. *Signed Fractional*

V tomto formátu se N-bitový operand reprezentovaný jako 1. [N-1] formát, kde před desetinnou tečkou je znaménkový bit (1 je znaménkový bit), za desetinnou tečkou je N-1 fractional bitů a N je počet bitů daného datového formátu. Rozsah signed fractional čísel je následující:

$$-1.0 \leq SF \leq +1.0 - 2^{-(N-1)}$$

Tento datový formát je k dispozici pro velikosti dat typu word a long word. V případě velikosti word je rozsah:

$$-1.0 \text{ (0x8000)} \leq SF \leq +1.0 - 2^{-15} \text{ (0x7FFF)}$$

V případě velikosti long word

$$-1.0 \text{ (0x80000000)} \leq SF \leq +1.0 - 2^{-31} \text{ (0x7FFFFFFF)}$$

1.7.5. *Unsigned Fractional*

Čísla ve formátu unsigned fractional jsou pouze kladná nebo rovna nule.

Rozsah signed fractional čísel je následující:

$$0.0 \leq UF \leq +2.0 - 2^{-(N-1)}$$

Tento datový formát je k dispozici pro velikosti dat typu word a long word.
V případě velikosti word je rozsah:

$$0.0 (0x0000) \leq UF \leq +2.0 - 2^{-15} (0xFFFF)$$

V případě velikosti long word

$$0.0 (\$80000000) \leq UF \leq +2.0 - 2^{-31} (0xFFFFFFFF)$$

1.7.6. *Fractional a Integer data – porovnání*

Hexadecimal Representation	Integer		Fraction	
	Binary	Decimal	Binary	Decimal
\$7FFF	0111 1111 1111 1111.	32767	0.111 1111 1111 1111	0.99997
\$7000	0111 0000 0000 0000.	28672	0.111 0000 0000 0000	0.875
\$4000	0100 0000 0000 0000.	16384	0.100 0000 0000 0000	0.5
\$2000	0010 0000 0000 0000.	8192	0.010 0000 0000 0000	0.25
\$1000	0001 0000 0000 0000.	4096	0.001 0000 0000 0000	0.125
\$0000	0000 0000 0000 0000.	0	0.000 0000 0000 0000	0.0
\$C000	1100 0000 0000 0000.	-16384	1.100 0000 0000 0000	-0.5
\$E000	1110 0000 0000 0000.	-8192	1.110 0000 0000 0000	-0.25
\$F000	1111 0000 0000 0000.	-4096	1.111 0000 0000 0000	-0.125
\$9000	1001 0000 0000 0000.	-28672	1.001 0000 0000 0000	-0.875
\$8000	1000 0000 0000 0000.	-32768	1.000 0000 0000 0000	-1.0

Tab. 1.2: Interpretace 16-bitových dat

Relace mezi 16-bitovou integer a fractional reprezentací je následující:

$$\text{Fractional value} = \text{Integer value} / 2^{15}$$

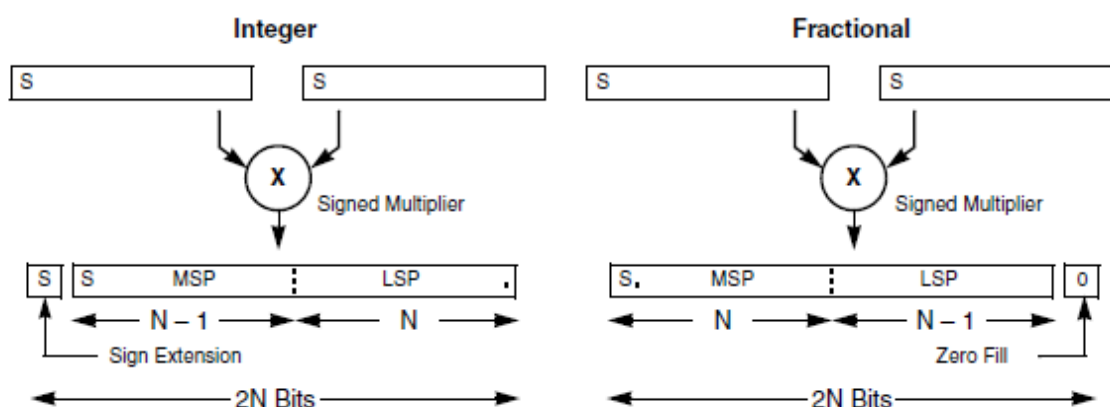
Relace mezi 32-bitovou integer a fractional reprezentací je následující:

$$\text{Fractional value} = \text{Integer value} / 2^{31}$$

1.7.7. *Násobení*

Výsledek integer a fractional násobení se liší. Výsledek fractional násobení je dvojnásobný oproti integer násobení. Jakékoliv binární násobení dvou signed N-bitových čísel dává výsledek, jehož velikost je $2N - 1$ bitů. V závislosti na tom, jak je výsledek umístěn do výsledného registru dostáváme buď integer nebo fractional výsledek. Integer a fractional násobení je ilustrováno na Obr. 1.7.

Signed násobení: $N \times N = 2N - 1$ bitů



Obr. 1.7: Porovnání integer a fractional násobení

Integer a fractional násobení je vykonáváno prostřednictvím příslušných instrukcí pro násobení. Instrukce MPY, MAC, MPYR a MACR vykonávají fractional násobení případně operaci multiple-accumulation. Instrukce pro integer (IMPY.W, IMPY.L, and IMAC.L) násobení neposkytují stejný výpočetní výkon a komfort jako instrukce pro násobení fractional.

1.8. Adresovací jednotka – AGU

Adresovací jednotka (AGU – Address Generation Unit) provádí výpočet a generování všech adres pro jádro DSP56800E. AGU poskytuje podporu pro aplikace, které vyžadují 24-bitové a 16-bitové ukazatele. V případě datové paměti podporuje generování adres pro přístup k datům typu byte, word a long-word.

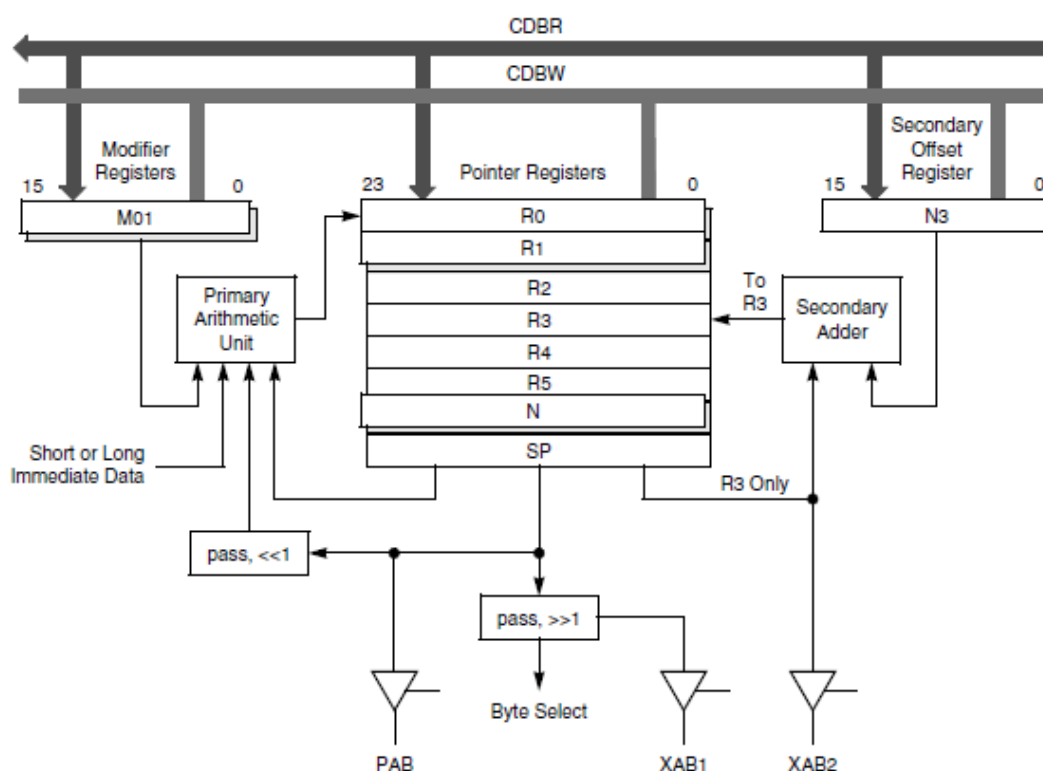
1.8.1. *Architektura adresovací jednotky*

AGU obsahuje registry a logiku nutnou k tomu, aby byla vygenerovaná efektivní adresa datových operandů. Podporuje dva druhy adresování: lineární a modulo.

Všechny AGU operace jsou vykonávány paralelně s ostatními funkcemi čipu, aby se minimalizovala režie na generování adres.

Hlavní bloky adresovací jednotky jsou:

- 24-bitová primární adresovací aritmetická jednotka
- 24-bitová sekundární adresovací aritmetická jednotka
- Dva single-bit shifters pro bytové adresování



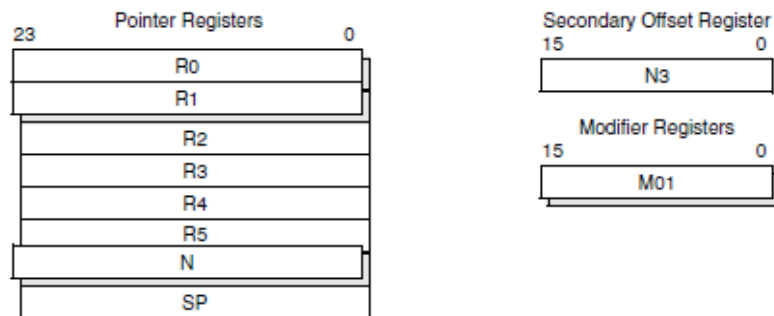
Obr.1.8: Blokový diagram adresovací jednotky

1.8.2. AGU – programovací model

Programovací model adresovací jednotky ilustrovaný na Obr.1.9. Obsahuje 14 programovatelných registrů

- Šest 24-bitových adresovacích registrů (R0 - R5)
- 24-bitový stack pointer register (SP)
- 24-bitový offset registr (N), který lze rovněž použít jako adresovací registr

- 16-bitový offset registr (N3)
- A 16-bitový modifier register (M01)
- Čtyři shadow registry (R0, R1, N, and M01)



Obr.1.9: Programovací model adresovací jednotky

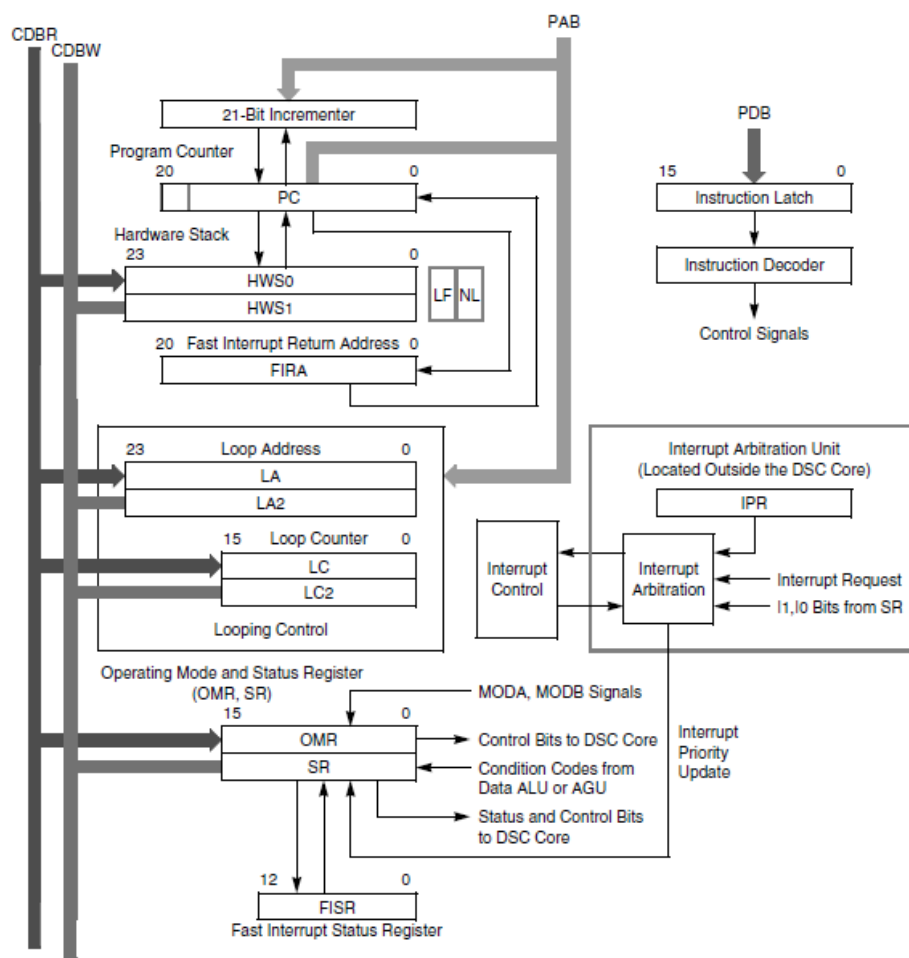
1.9. Programový řadič – program controller

Programový řadič je pravděpodobně nejdůležitější jednotkou jádra DSP56800E. Stará se o přenos a dekódování instrukcí, koordinuje ostatní jednotky jádra vykonávající jednotlivé instrukce, řídí chod programu a zpracovávání vyjímek. Obsahuje logiku pro podporu „looping“ operací.

1.9.1. *Architektura programového řadiče*

Programový řadič obsahuje následující bloky:

- Instruction latch and decoder
- Program counter (PC)
- Hardware stack
- Looping control unit
- Interrupt control unit

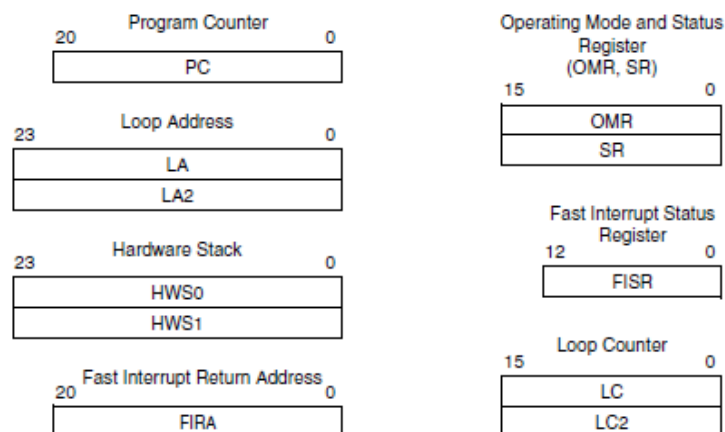


1.9.2. *Programový řadič – programovací model*

Programovací model programového řadiče se skládá ze sedmi uživatelsky přístupných registrů a dvou speciálních registrů pro zpracování rychlých přerušení

- Status register (SR)
- Operating mode register (OMR)
- Hardware stack register (HWS)
- Two loop address registers (LA and LA2)
- Two loop count registers (LC and LC2)
- Fast interrupt return address register (FIRA)

- Fast interrupt status register (FISR)



Obr.1.11: Programový řadič – programovací model

1.10. Bit manipulation unit

Bit manipulation unit vykonává operace na úrovni jednotlivých bitů přímo na registrech jádra DSP56800E nebo v datové paměti. Jednotka umožňuje testování, nastavování, mazání nebo invertování libovolných bitů, které jsou specifikovány 16-bitovou maskou. Instrukce vykonávané jednotkou jsou atomické, což znamená, že jsou nepřerušitelné. Principiálně se jedná o instrukce typu read-modify-write, tedy je nejprve nutné přečíst celý registr nebo datové místo, aplikovat logickou operaci s maskou a následně zapsat výsledek operace zpět do modifikovaného registru nebo paměťového místa.

Bit manipulation unit vykonává následující sadu instrukcí:

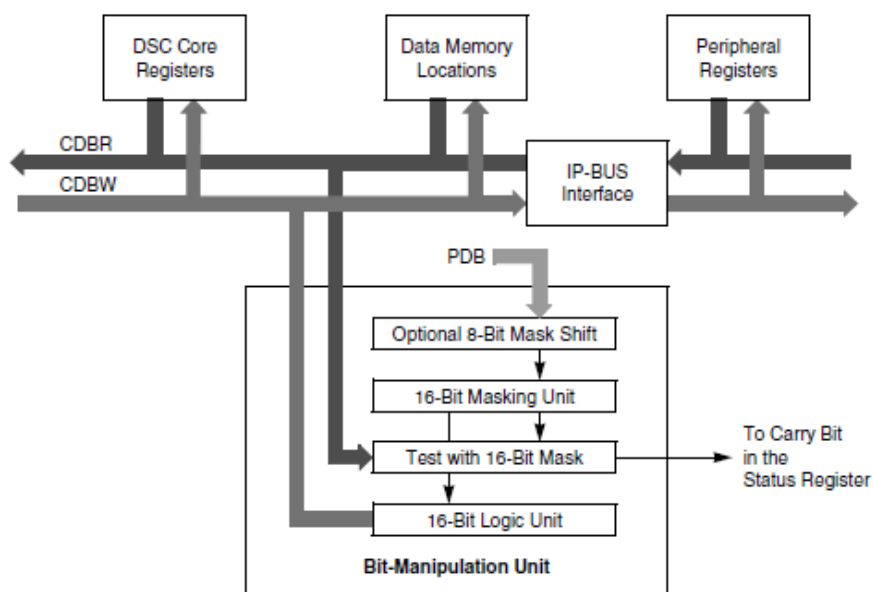
- Test and modifikace bitů v 16-bitovém slově:
 - BFSET: Test a následné nastavení zvolených bitů podle masky
 - BFCLR: Test a následné vymazání zvolených bitů podle masky
 - BFCHG: Test a následná inverze zvolených bitů podle masky
- Test zvolených bitů v 16-bitovém slově:

- BFTSTH: Test zvolených bitů 16-bitového slova podle zvolené masky na logickou jedničku
- BFTSTL: Test zvolených bitů 16-bitového slova podle zvolené masky na logickou nulu
- Test zvolených bitů horního a dolního bytu 16-bitového slova a větvení podle výsledku testu:
- BRSET: Větvení v případě že zvolené bity podle masky jsou všechny v logické jedničce
- BRCLR: Větvení v případě že zvolené bity podle masky jsou všechny v logické nule

1.10.1. *Bit manipulation unit - architektura*

Bit manipulation unit obsahuje následující moduly:

- 8-bit mask shifting unit
- 16-bit masking unit
- 16-bit testing unit
- 16-bit logic unit

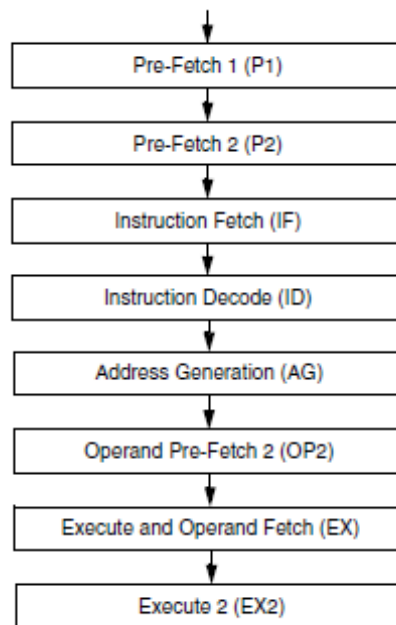


Obr.1.12: Bit manipulation unit – blokový diagram

1.11. Instruction pipeline

Jeden ze znaků DSP architektury je rovněž zpracování instrukcí v tzv. pipeline processing. DSP56800E architektura je založena na 8 stavovém pipeliningu. Instrukce typicky potřebují 7 nebo 8 hodinových cyklů na kompletní zpracování, což znamená že většina instrukcí je ukončena v 7 nebo 8 stavu pipeline. Aritmetické instrukce adresovací jednotky AGU jsou ukončeny ve stavu „Address Generation stage“, což je 5 stav pipeline.

Osmi stavový pipelining ukazuje Obr.1.13.



Obr.1.13: DSP56800E – osmi stavový pipelining

Příklad zpracování instrukcí jádrem DSP56800E:

MOVE.W	X:(R0),A	/* n1: 1-word, 1-cycle instruction */
ADD	A,B	/* n2: 1-word, 1-cycle instruction */
MOVE.W	B,C	/* n3: 1-word, 1-cycle instruction */
MOVE.W	C1,X:\$0C00	/* n4: 2-word, 2-cycle instruction */
INC.W	C	/* n5: 1-word, 1-cycle instruction */

Pipeline Stage	Instruction Cycle											
	1	2	3	4	5	6	7	8	9	10	11	•
P1 (Pre-Fetch 1)	n1	n2	n3	n4	n4e	n5	•	•	•	•	•	•
P2 (Pre-Fetch 2)		n1	n2	n3	n4	n4e	n5	•	•	•	•	•
IF (Instruction Fetch)			n1	n2	n3	n4	n4e	n5	•	•	•	•
ID (Instruction Decode)				mov ¹	add	mov	mov	mov	inc	•	•	•
AG (Address Generation)					mov	add	mov	mov	mov	inc	•	•
OP2 (Operand Pre-Fetch 2)						mov	add	mov	mov	mov	inc	•
EX (Execute and Operand Fetch)							mov	add	mov	mov	mov	•
EX2 (Execute 2)								—	—	—	—	—

Tab. 1.3: Zpracování instrukcí pipeline

1.12. JTAG a Enhanced On-Chip Emulation (Enhanced OnCE)

Rodina mikropočítačů s jádrem DSP56800E disponuje integrovanou podporou pro vývoj aplikačního software a ladění v reálném čase. Dva moduly „Enhanced On-Chip Emulation module“ (Enhanced OnCE) a Core Test Access Port (TAP), který se běžně nazývá JTAG port pracují společně. Oba moduly jsou přístupné přes JTAG/Enhanced OnCE rozhraní.

DSP56800E Enhanced OnCE module je specifický modul navržený firmou Freescale. Modul umožňuje neintruzivní interakci s jádrem. Je přístupný jak z vnějšku čipu přes piny JTAG rozhraní nebo softwarově z jádra DSP56800E.

Enhanced OnCE poskytuje podporu pro datovou komunikaci mezi DSC čipem a ladicím systémem v reálném čase. Mezi další vlastnosti mimo jiné patří podpora pro hardwarové breakpointy, monitorování a sledování chodu programu, sledování a modifikování obsahu registrů, paměti a periferních modulů na čipu

DSP56800E JTAG port slouží jako interface mezi Enhanced OnCE modulem a DSC JTAG piny.

2. EMBEDDED MIKROPOČÍTAČ FREESCALE DSC56F8025

2.1. Rodina mikropočítačů Freescale DSC56F80xx

Rodina mikropočítačů Freescale DSC56F80xx čítá devět zástupců. Jednotlivé mikropočítače se liší ve velikosti pouzdra, velikosti paměti RAM a FLASH, v počtu periférií implementovaných na čipu a typech periférií. Celá rodina je založena na jádře DSP56800E. Napájecí napětí je 3.3V a běží na frekvenci 32MHz. Standardní ladicí rozhraní je JTAG/Enhanced OnCE.

Do rodiny mikropočítačů Freescale DSC56F80xx patří následující zástupci:

- **DSC56F8011** – pouzdro je 32 pinové LQFP, programová/datová Flash – 12kB, programová/datová RAM – 2kB
- **DSC56F8013** – pouzdro je 32 pinové LQFP, programová/datová Flash – 16kB, programová/datová RAM – 4kB
- **DSC56F8014** – pouzdro je 32 pinové LQFP, programová/datová Flash – 16kB, programová/datová RAM – 4kB
- **DSC56F8023** – pouzdro je 32 pinové LQFP, programová/datová Flash – 32kB, programová/datová RAM – 4kB
- **DSC56F8025** – pouzdro je 44 pinové LQFP, programová/datová Flash – 32kB, programová/datová RAM – 4kB
- **DSC56F8036** – pouzdro je 48 pinové LQFP, programová/datová Flash – 64kB, programová/datová RAM – 8kB
- **DSC56F8037** – pouzdro je 64 pinové LQFP, programová/datová Flash – 64kB, programová/datová RAM – 8kB
- **DSC56F8002** – pouzdro je 32 pinové nebo 48 pinové LQFP nebo 28 pinové SOIC, programová/datová Flash – 12kB, programová/datová RAM – 2kB
- **DSC56F8006** – pouzdro je 32 pinové nebo 48 pinové LQFP nebo 28 pinové SOIC, programová/datová Flash – 16kB, programová/datová RAM – 2kB

Mikropočítače **DSC56F8011/13/14** interně nazývané jako **Anguilla** jsou kompatibilní z hlediska typu periférií.

Mikropočítače **DSC56F8023/25/36/37** interně nazývané jako **Anguilla Blue** jsou kompatibilní z hlediska typu periférií.

Mikropočítače **DSC56F8002/6** interně nazývané jako **Anguilla White** jsou kompatibilní z hlediska typu periférií.

	56F8011	56F8013	56F8014	56F8023	56F8025	56F8036	56F8037
Performance	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs	32MHz/MIPs
Temp Range (V)	-40C to 105C	-40C to 105C	-40C to 105C	-40C to 105C	-40C to 105C	-40C to 105C	-40C to 105C
Temp Range (M)		-40C to 125C					
Voltage Range	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
Program/Data Flash	12KB	16KB	16KB	32KB	32KB	64KB	64KB
Program/Data RAM	2KB	4KB	4KB	4KB	4KB	8KB	8KB
On Chip Relaxation Osc.	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PLL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
COP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PWM	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	3	4	4	4	4
12-bit ADCs	2 x 3ch	2 x 3ch	2 x 4ch	2 x 3ch	2 x 4ch	2 x 4ch	2 x 8ch
12-bit DACs				2 (Internal)	2 (Internal)	2 (Internal)	2 (External)
Analog Comparator				2	2	2	2
16-bit Timers	4	4	4	4	4	4	8
Prog. Interval Timers				1	3	3	3
GPIO (max)	26	26	26	26	35	39	53
IIC	1	1	1	1	1	1	1
SCI (UART) / LIN Slave	1 - SCI	1 - SCI	1 - SCI	1 - QSCI	1 - QSCI	1 - QSCI	2 - QSCI
SPI (Synchronous)	1 - SPI	1 - SPI	1 - SPI	1 - QSPI	1 - QSPI	1 - QSPI	2 - QSPI
CAN						MSCAN	MSCAN
JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE	JTAG/EOnCE
Package (V) - Industrial	32LQFP	32LQFP	32LQFP	32LQFP (.8p)	44QFP (.8p)	48LQFP (.5p)	64LQFP (.5p)
Package (M) - Automotive		32LQFP					

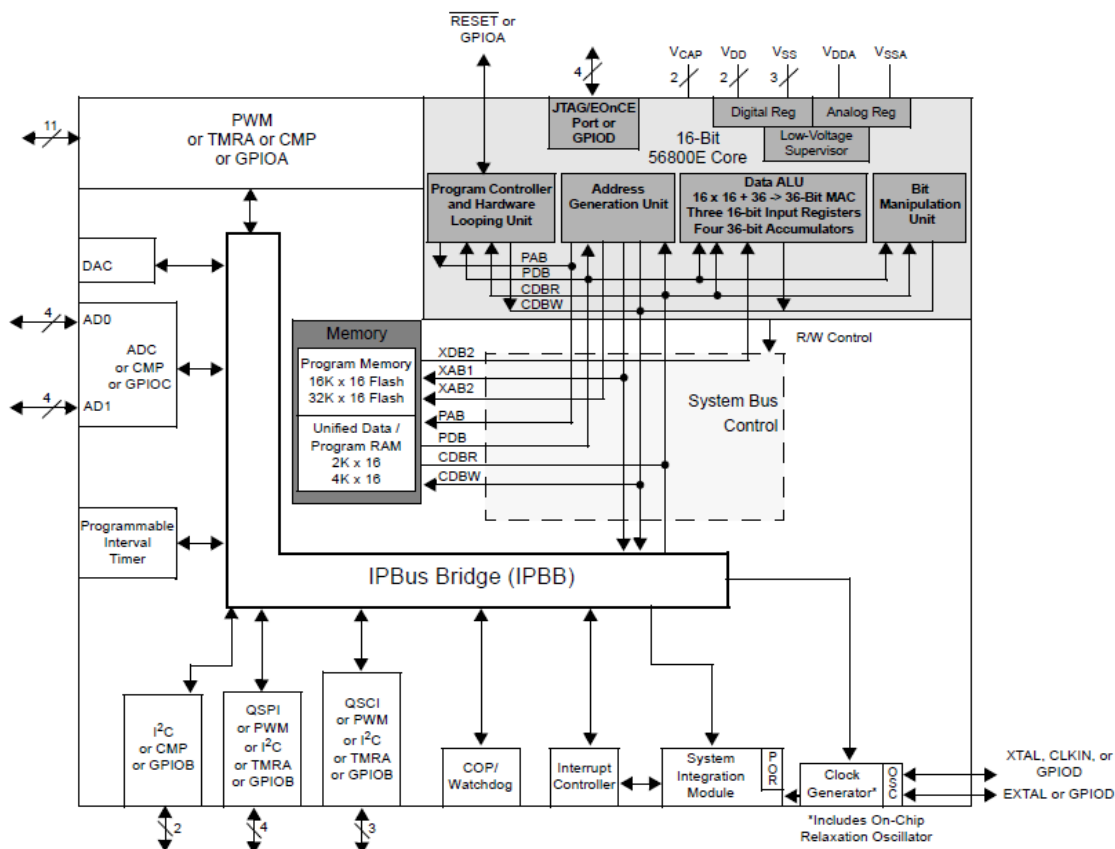
Tab. 2.1: Přehled čipů a jejich vlastnosti

2.2. DSC56F8025/56F8035 – základní parametry

Mikropočítače DSC56F8025/56F8035 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 DSC56F8025/35 cenově výhodné řešení pro širokou škálu aplikačního využití. Jsou vhodné 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
- Frekvence jádra – 32 MHz, což odpovídá výpočetnímu výkonu maximálně 32MIPS
- 56F8035 - 64KB (32K x 16) Program Flash
- 56F8025 - 32KB (16K x 16) Program Flash
- 56F8035 - 8KB (4K x 16) Unified Data/Program RAM
- 56F8025 - 4KB (2K x 16) Unified Data/Program RAM
- PWM modul – flexibilní a velmi komplexní peiferie s 6 PWM kanály
- Dva 4-kanálové 12-bitové Analog-to-Digital Converters (ADCs)
- 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 – Quad Timer (TMR)
- Jeden modul I2C (Inter-Integrated Circuit)
- Watchdog – COP (Computer Operating Properly)
- Relaxační oscillator integrovaný na čipu
- Integrovaný Power-On Reset (POR) a Low-Voltage Interrupt (LVI)
- JTAG/Enhanced On-Chip Emulation (OnCE™)
- Až 35 GPIO



Obr. 2.1: Blokové schéma mikropočítače 56F8025/56F8035

2.3. 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.
- Peripheral Reference Manual – dokument popisující periférie mikropočítače pro danou rodinu.
- Reference Manual – vyčerpávající popis jádra mikropočítače
- Chip Errata – seznam zjištěných chyb a jejich případná řešení

2.4. ADC – Analog Digital Converter

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

2.4.1. *ADC - vlastnosti*

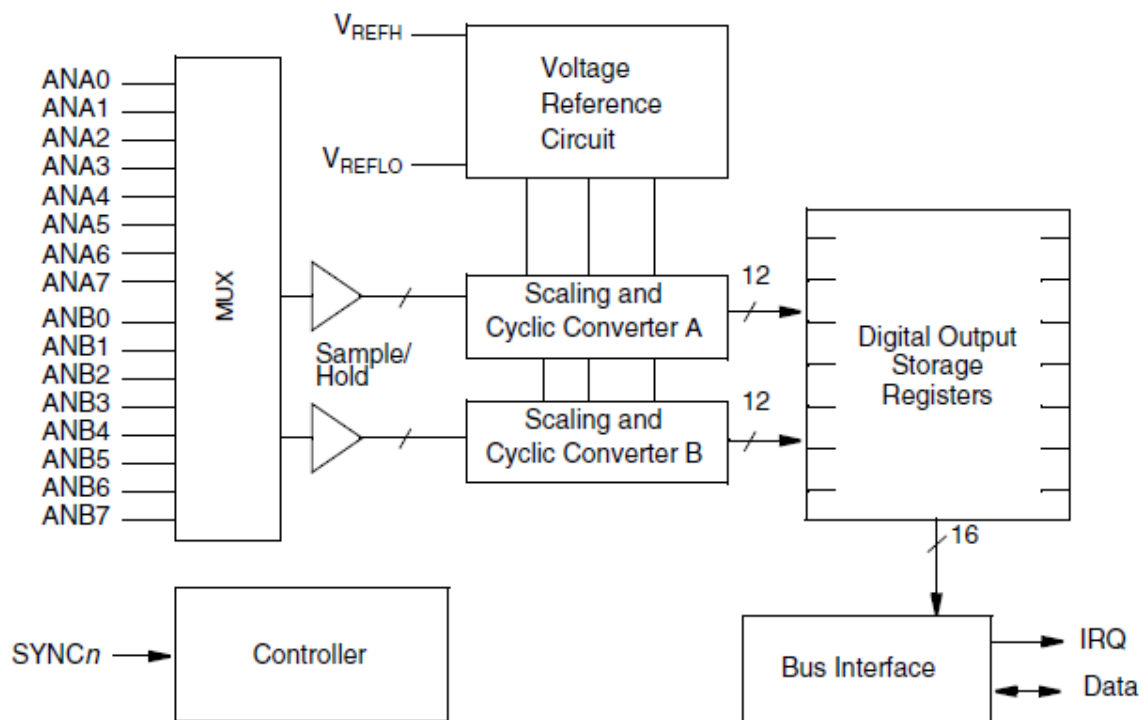
ADC modul charakterizují následující vlastnosti:

- 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 ($6 \times 187 \text{ ns} = 1.125 \mu\text{s}$)
- Doba odpovídající osmi převodům - 26.5 ADC hodinových cyklů ($26.5 \times 187 \text{ ns} = 4.97 \mu\text{s}$) 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 interrupty 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

2.4.2. *ADC – blokový diagram*

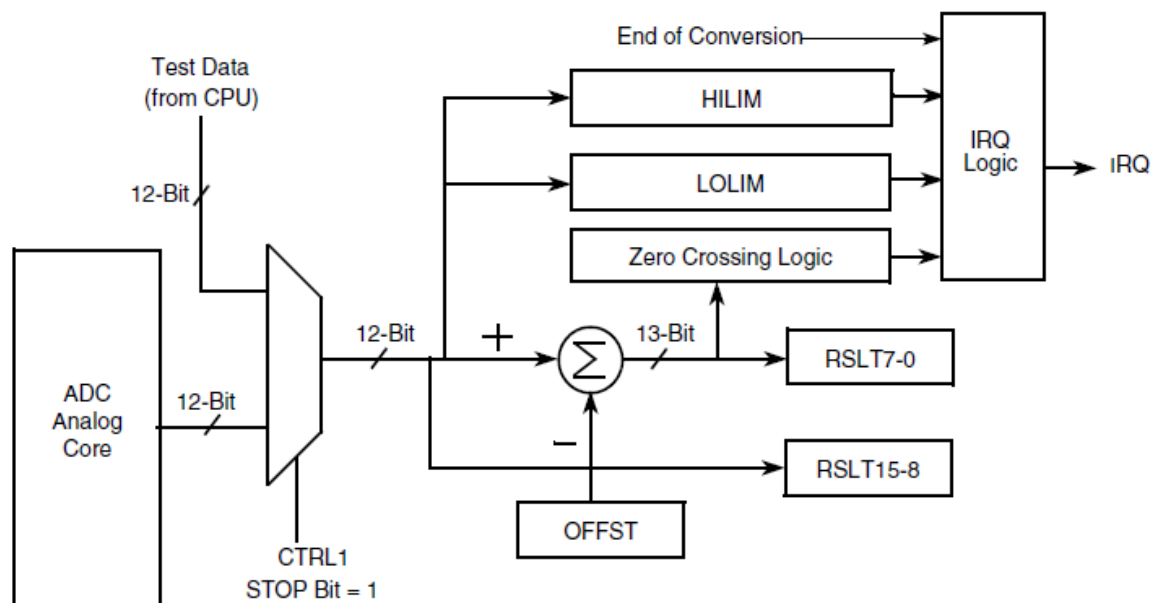
Následující obrázek Obr. 2.2 demonstruje konfiguraci ADC modulu



Obr. 2.2: Blokový diagram ADC modulu se dvěma nezávislými převodníky

2.4.3. *ADC – zpracování dat převodníkem*

Obr. 2.3 ukazuje zpracování výsledku převodu ADC převodníkem.



Obr. 2.3: 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 Zero Crossing logikou a v případě průchodu nulou může být opět generovaný interrupt.

2.5. 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 servisování COP logiky zápisem

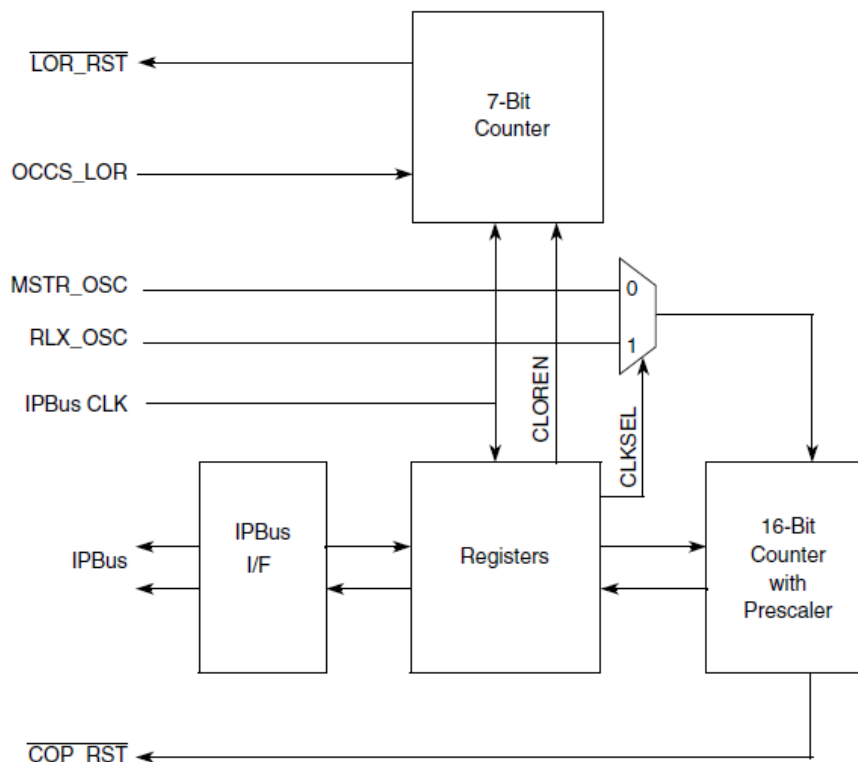
specifických čísel v do counter registru ve správném pořadí. V tomto případě je nutno periodicky updatovat CNTR (Counter Register) register sekvencí čísel 0x5555 a následně 0xAAAA.

2.5.1. *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 zakázán
- Volitelný zdroj hodin pro COP modul
- COP modul rovněž zpracovává signál z PLL modulu, při jehož ztrátě dochází k resetování DSC po 128 cyklech

2.5.2. *COP – blokový diagram*



Obr. 2.4: Blokový diagram COP modulu

2.6. PWM – Pulse Width Modulator

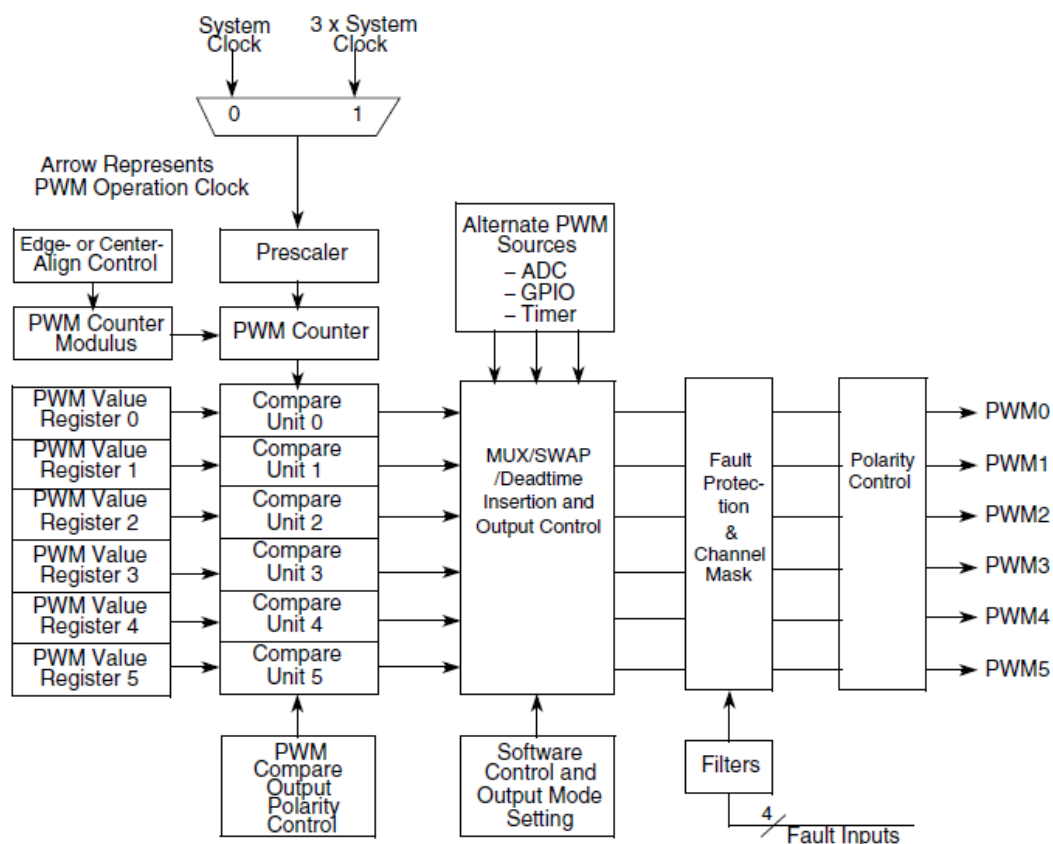
PWM modul je komplexní a poskytuje celou řadu komplikovaných nastavení v závislosti na připojeném externím zařízení. PWM modul umožňuje generovat tři komplementární výstupy, šest nezávislých (independent) výstupů nebo jejich kombinace jako např.: jeden komplementární a čtyři nezávislé výstupy. Oba módy edge- a center-aligned dovolují řídit šířku pulsu od 0% až do 100%. PWM modul pro generaci PWM signálu používá 15-bitový counter 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.

2.6.1. PWM – vlastnosti

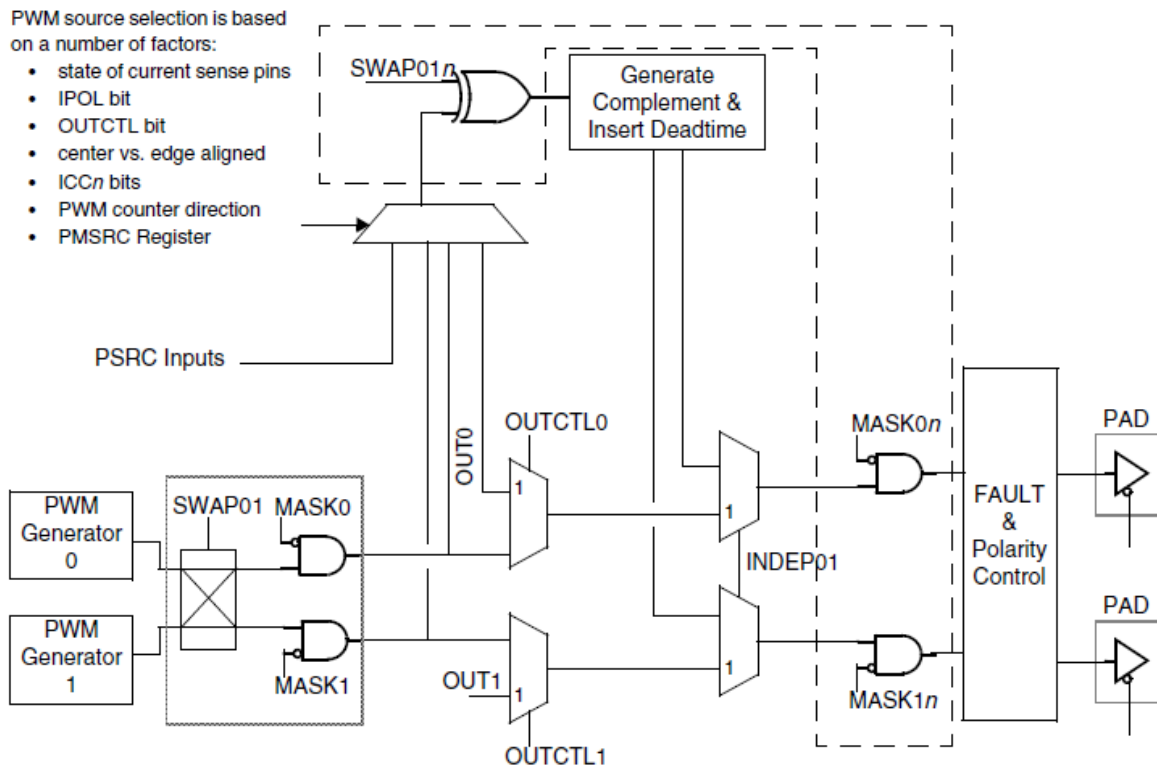
- PWM operační hodiny běží buď na stejné frekvenci jako systémové hodiny, nebo na trojnásobku systémových hodin.
- 6 PWM výstupních signálů
 - Všechny nezávislé (independent)
 - Tři komplementární páry (Complementary pairs)
 - Kombinace nezávislých a komplementárních párů
- Vlastnosti PWM výstupů v případě 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)
- Integrovaný 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 filter pro fault piny
- Ochrana PWM modulu ve formě write-protected registrů

2.6.2. PWM – blokový diagram



Obr. 2.5: Blokový diagram PWM modulu



Obr. 2.6: Detailní blokový diagram funkcí MUX, Swap and Deadtime

2.7. 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.

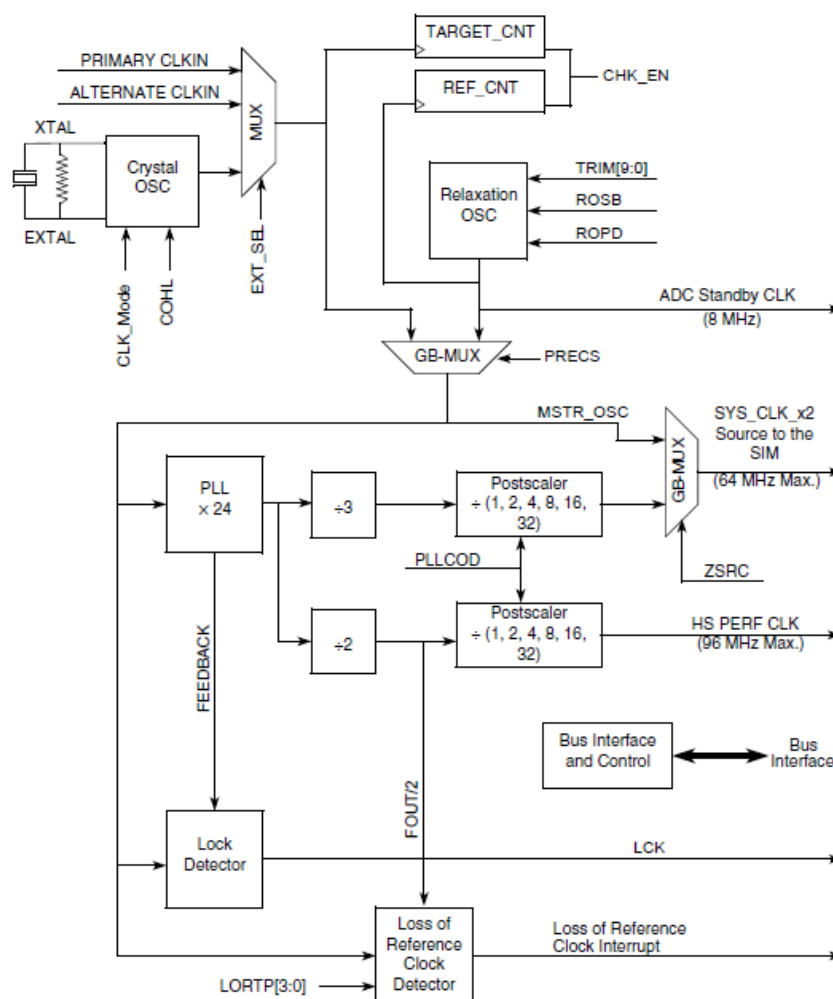
2.7.1. OCCS – vlastnosti

Vlastnosti OCCS modulu jsou následující:

- Interní relaxační oscilátor
- Zpracování signálu z crystalového 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 timer a PWM moduly
- Standby mód relaxačního oscilátoru

2.7.2. OCCS – blokový diagram



Obr. 2.7: Blokový diagram OCCS modulu s externím crystalovým oscilátorem

2.8. FM – Flash Memory

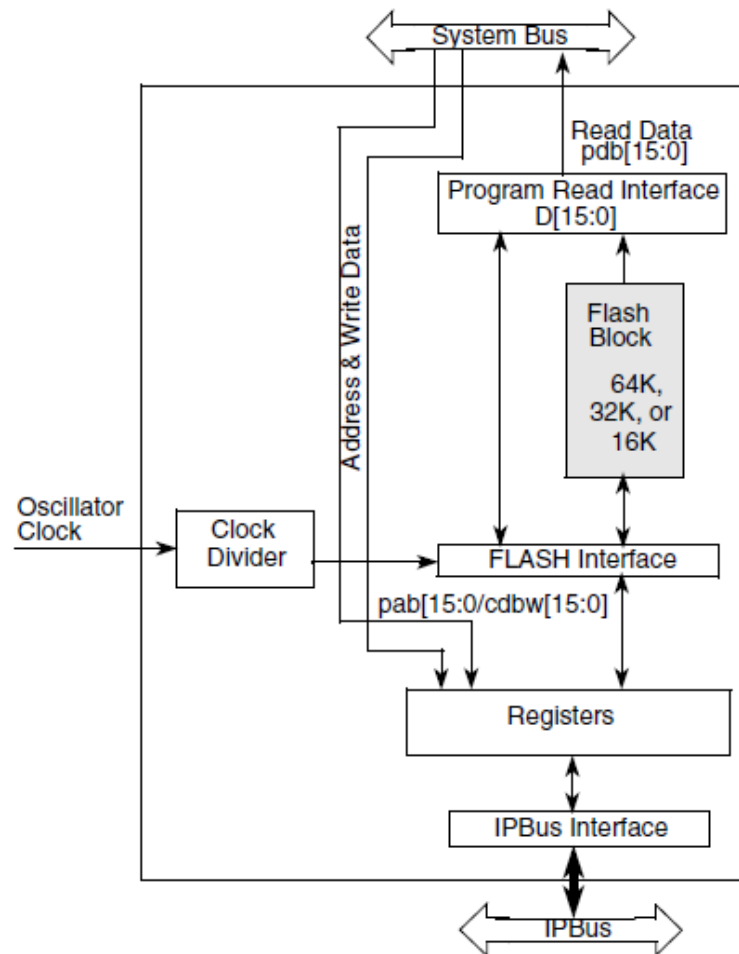
Paměť Flash (FM) je typu non-volatile, což znamená, že informace uložené v paměti zůstává zachovaná i po odpojení napájení. Používá se pro ukládání programu. FM implementovaná na mikropočítačích Freescale podporuje

přeprogramování bez použití externího napájení. Mazání a programování se provádí přes rozhraní jádra mikropočítače.

2.8.1. *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 ccess error
- Rychlé mazání stránky
- Jediné napájení pro programování a mazání
- Vlastnosti bezpečnosti
- System – sector protection

2.8.2. FM – blokový diagram



Obr. 2.8: 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 wait states)

2.9. PIT – Programmable Interval Timer

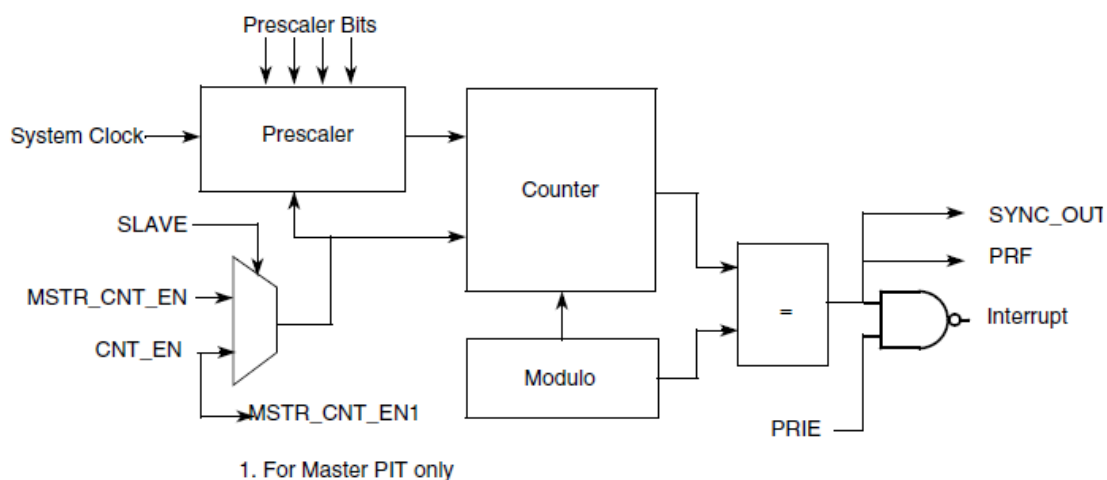
PIT timer modul je velmi jednoduchý časovací modul sloužící pro nenáročné časování aplikace. Obsahuje 16-bitový čítač vpřed, modulo register a řídicí register. Modulo a řídicí registry jsou read/write, zatímco counter register je pouze read only.

Modulo register 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.

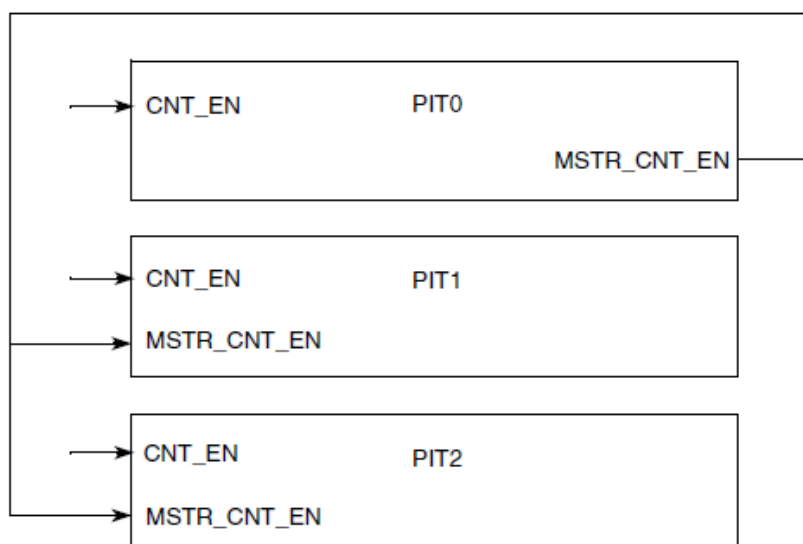
2.9.1. *PIT – vlastnosti*

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

2.9.2. PIT – Blokový diagram



Obr. 2.9: Blokový diagram PIT modulu



Obr. 2.10: Propojení PIT modulů ve slave módu

2.10. 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

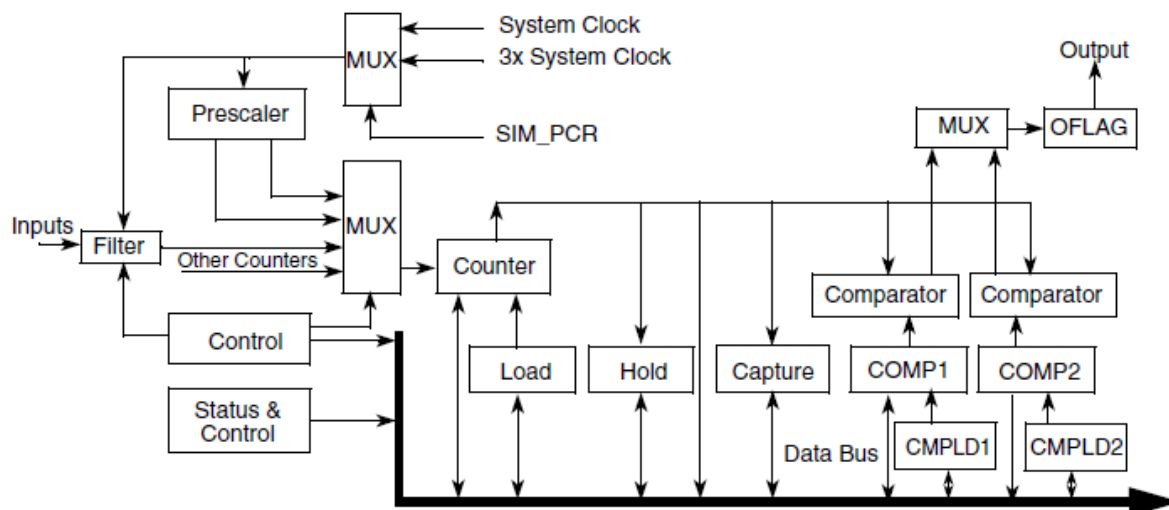
- Hold registr
- Capture registr
- Dva compare registry
- Jeden status a control registr
- Jeden control registr

2.10.1. *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 filter
- Start čítačů může být synchronizován

2.10.2. TMR – blokový diagram

Blokový diagram na Obr. 2.11 ukazuje základní zapojení jednoho submodulu časovacího modulu TMR. Stejná struktura se v modulu TMR vyskytuje 4x.



Obr. 2.11: TMR submodul – blokový diagram

2.11. QSCI – Qued Serial Communication Interface

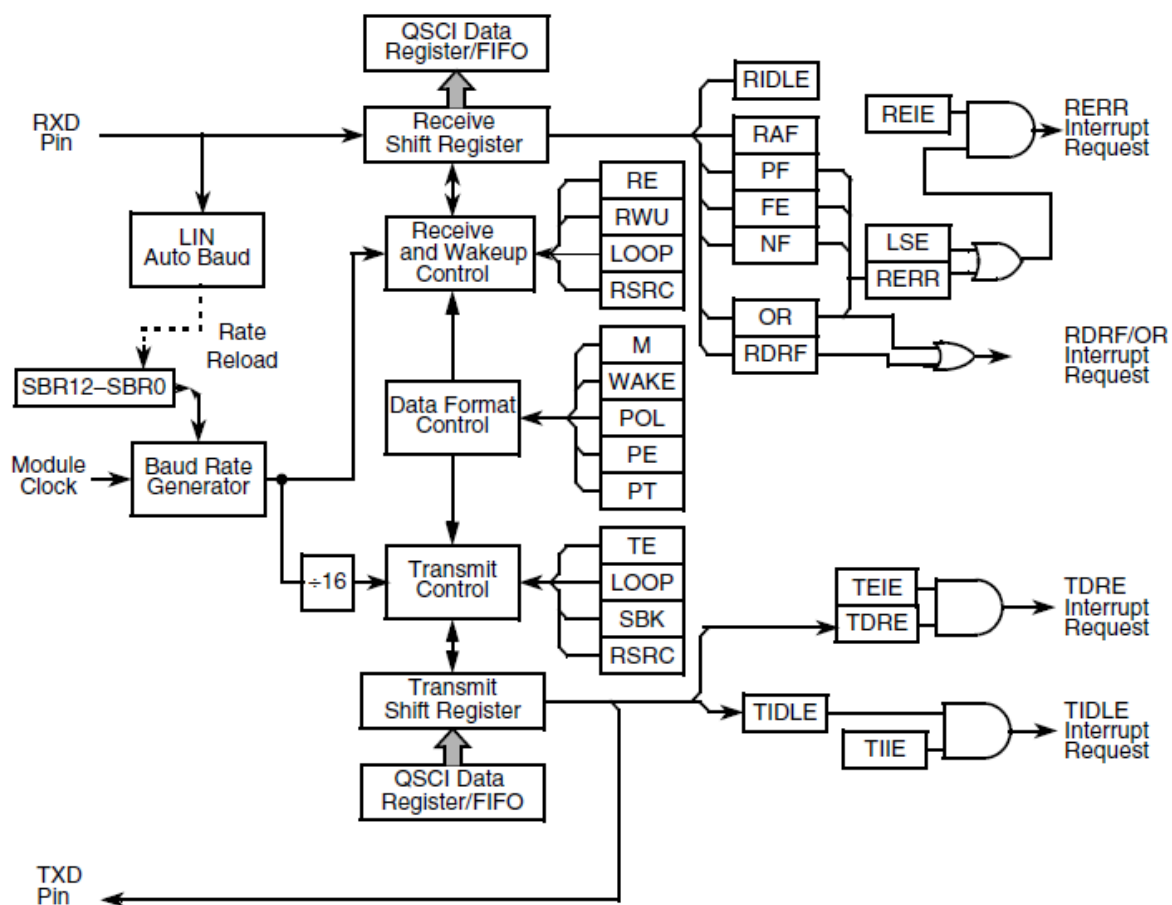
Tento modul umožňuje asynchronní sériovou komunikaci s periferními zařízeními a jinými mikropočítači.

2.11.1. QSCI – 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 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

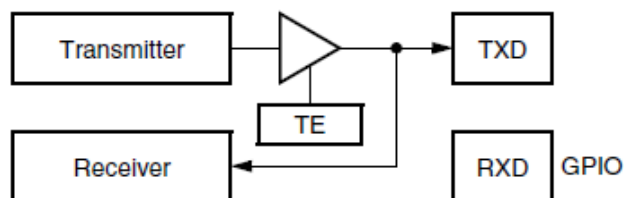
2.11.2. QSCI – blokový diagram



Obr. 2.12: QSCI modul – blokový diagram

2.11.3. 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. 2.13: QSCI modul – single-wire komunikace

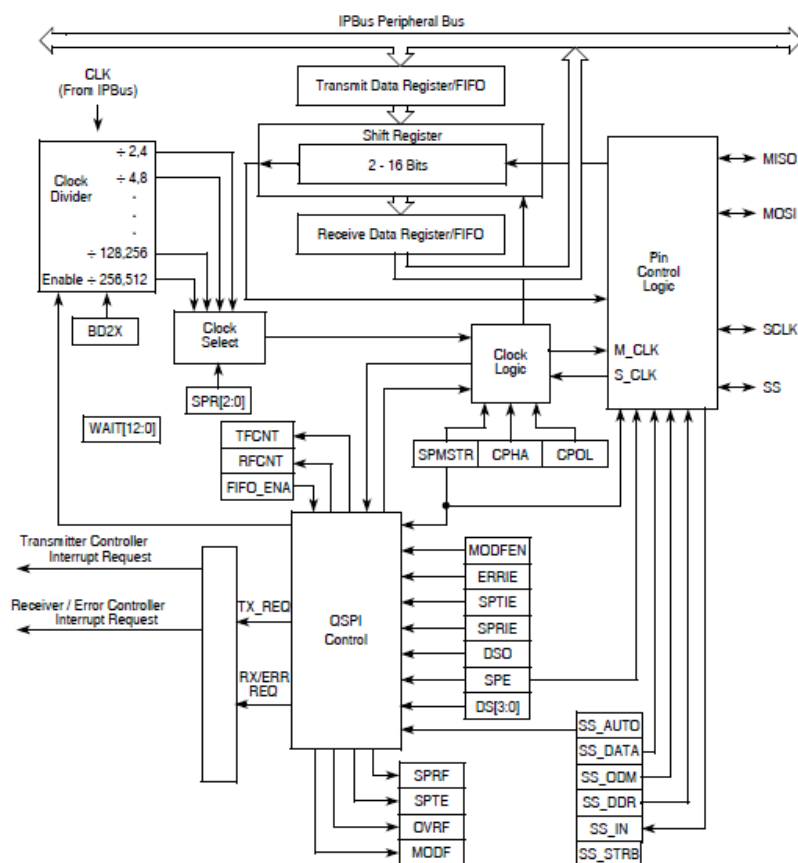
2.12. QSPI – Qued Serial Peripheral Interface

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

2.12.1. QSPI – vlastnosti

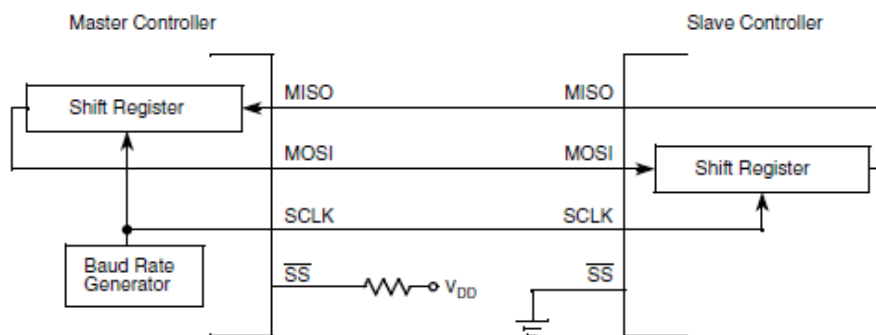
- Full-duplex
- Master a slave módy
- Double-buffered funkčnost separátně pro vysílací a přijímací registry
- FIFO se čtyřmi úrovněmi pro vysílací a přijímací mezipaměť
- Programovatelná délka přenosu (2 až 16-bitů)
- Maximální frekvence ve slave módu – $\text{frequency} < \text{bus frequency} / 2$
-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

2.12.2. QSPI – blokový diagram



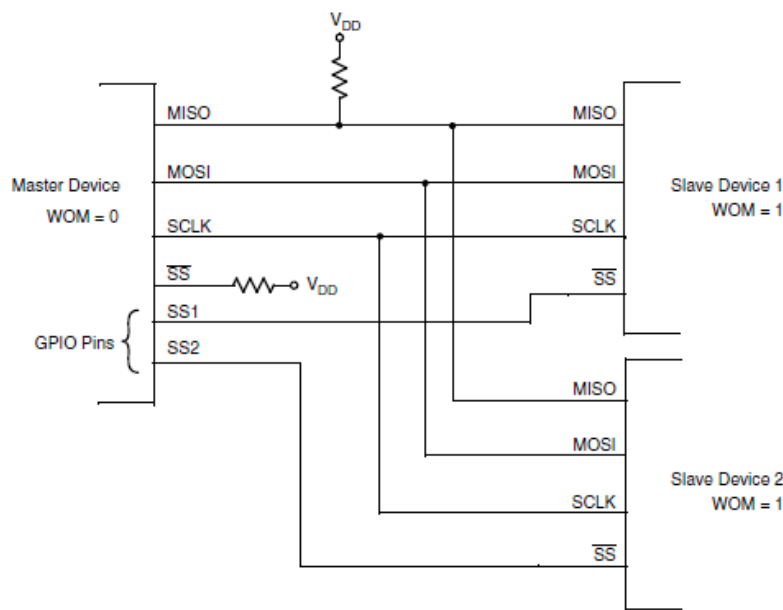
Obr. 2.14: QSPI modul – blokový diagram

2.12.3. QSPI – master mód



Obr. 2.15: QSPI modul – full-duplex master/slave zapojení

2.12.4. QSPI – *wired-OR* mód



Obr. 2.16: Propojení QSPI modulů - Master a dva slaves

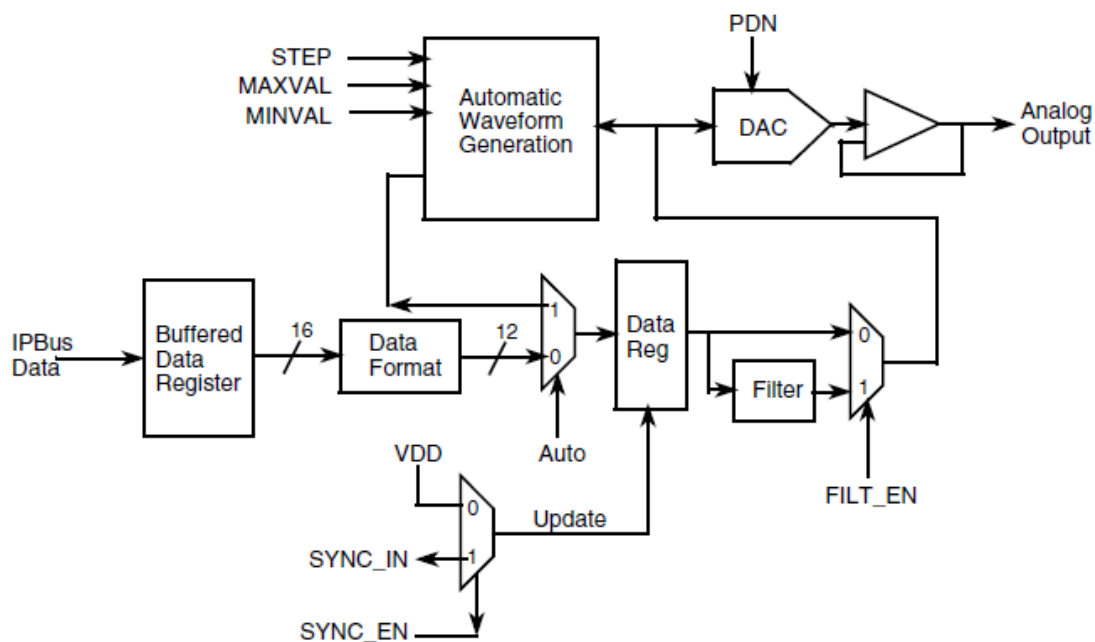
2.13. DAC – Digital-to-Analog Converter

DAC modul zpracovává 12-bitový signál a generuje analogové napětí v rozsahu VSSA aVDDA. 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.

2.13.1. DAC – *vlastnosti*

- 12-bitové rozlišení
- Conversion time - 2μs
- Power-down mód
- DAC výstup může být volitelně propojen s komparátorem, ADC modulem, případně na externí pin
- DAC může pracovat do definované impedance– 3 kΩ a 400 pF
- Automatic mód – DAC generuje vlastní časové průběhy - obdélník, trojúhelník a pilu

2.13.2. DAC – blokový diagram



Obr. 2.17: DAC modul – blokový diagram

2.14. CMP – Comparator

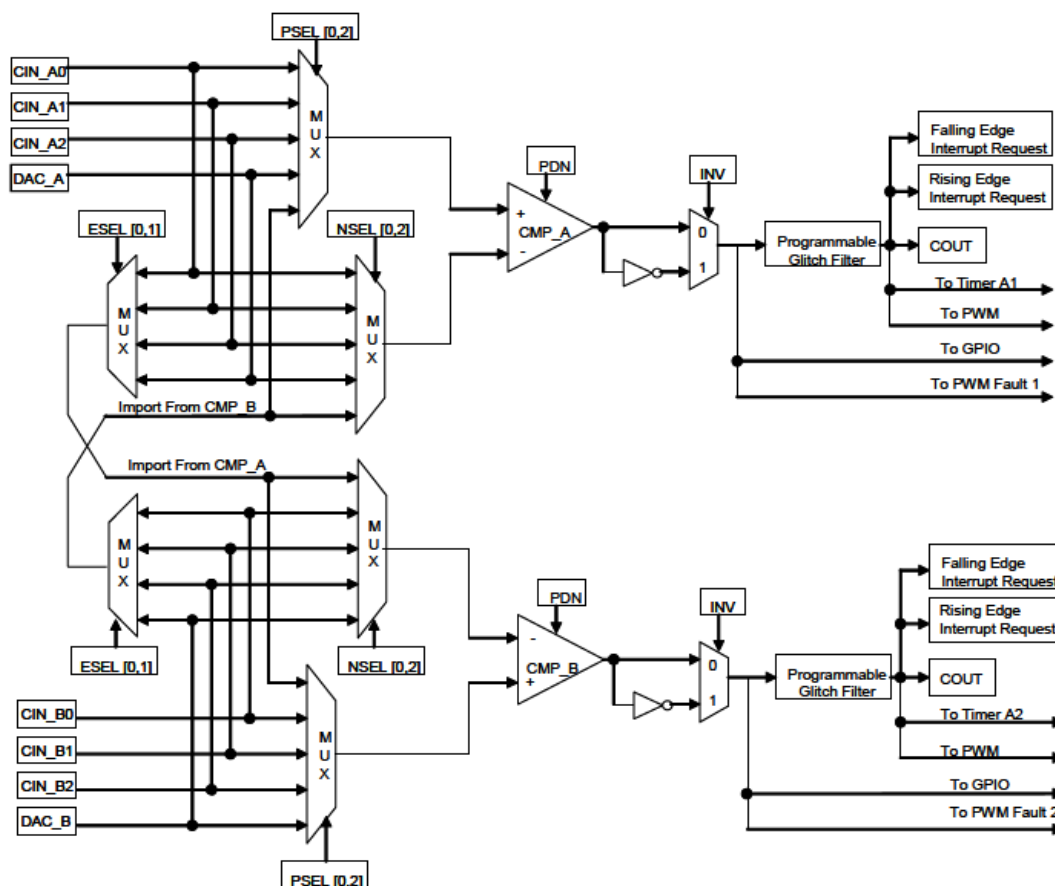
Komparátor se skládá analogového modulu komparátoru a řídicího digitálního rozhraní.

2.14.1. 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
- Power saving mód
- Detekce sestupné a nástupné hrany s volitelným přerušením na dané hrany

- Výstup může být použitý pro řízení vstupů timeru, PWM faults, PWM control. Dalé lze propojit výstup z komparátoru na pin nebo ho použít jako zdroj pro přerušení.

2.14.2. CMP – blokový diagram



Obr. 2.18: CMP modul – blokový diagram

2.15. 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 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.

2.15.1. *GPIO – vlastnosti*

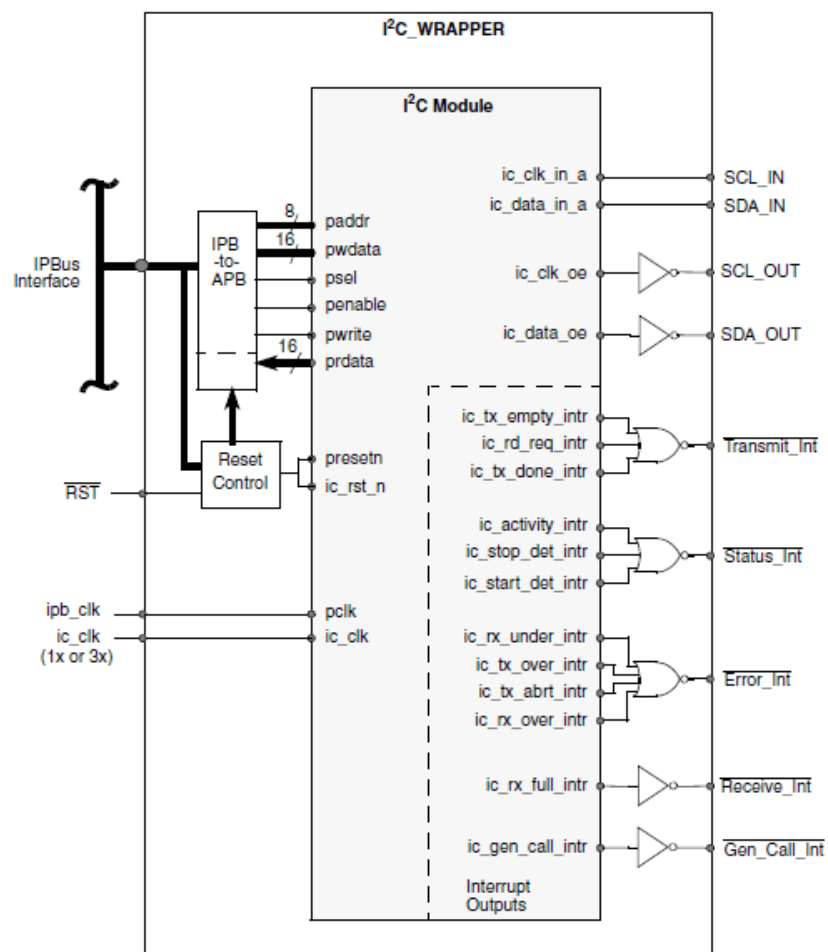
- Individuální konfigurace pinu – buď mód periferie nebo GPIO
- Konfigurace směru pinu v GPIO módu – input nebo output
- Volitelné připojení pullup rezistoru – konfigurace pinu jako input a to v módu periferie nebo GPIO
- Volba módu push-pull nebo open drain - konfigurace pinu jako output a to v módu 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.

2.16.1. I²C – vlastnosti

- Dvou vodičové seriové 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
- Digitální filter – SDA a SCL

2.16.2. I^2C – blokový diagram



Obr. 2.20: I^2C modul – blokový diagram

3. 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 portovatelný na jiné platformy. Portovatelnost 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.

3.1. Quick Start Tool – vlastnosti

Vývojové prostředí Quick_Start se skládá z několika součástí:

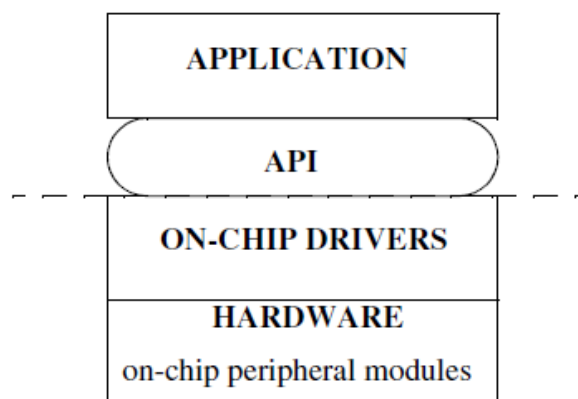
- Infrastruktura celého systému
- On-chip drivers s jednotným API
- Vzorové aplikace
- GCT (Graphical Configuration Tool)
- Softwarová podpora nástroje FreeMaster

3.1.1. *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, portovatelná deklarace registrů, mechanismus pro statickou konfiguraci čipu, tabulka přerušení a obsluha vektorů přerušení, start-up kód, soubory pro konfiguraci linkeru, project templates, soubory pro inicializaci debuggeru, atd.

3.1.2. *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 periférií (Obr. 3.1).



Obr. 3.1: Softwarová struktura aplikačního kódu

3.1.3. *Vzorové aplikace*

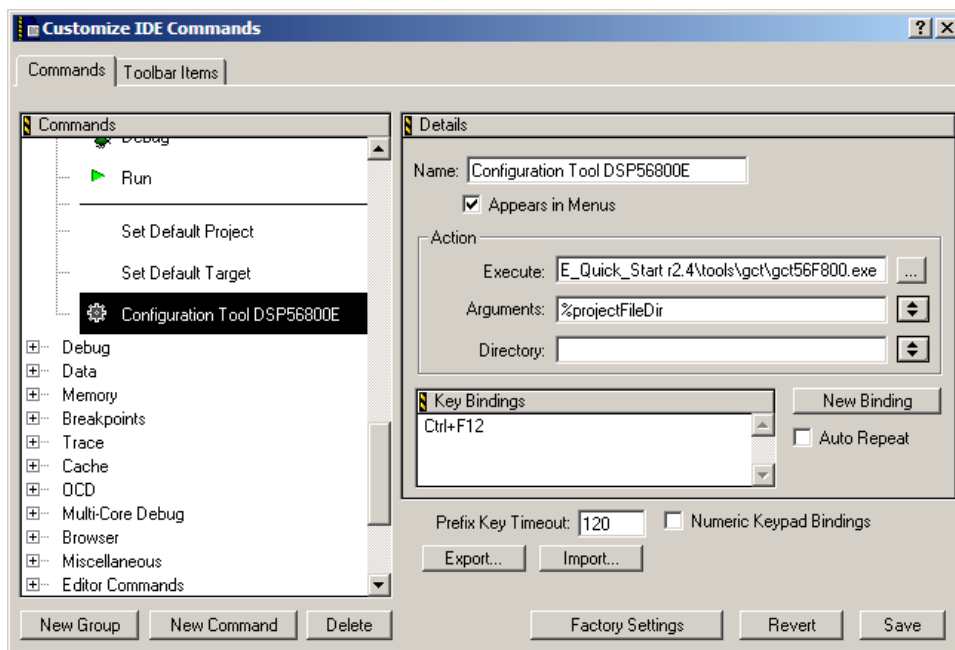
Prostředí Quick Start poskytuje uživateli celou řadu příkladů obsluhy typických periférií, na kterých se demonstruje použití on-chip drivers a zároveň se vysvětluje jak implementovat specifika nastavení mikropočítače. Příklady jsou jednoduché a ilustrativní. Jejich účelem je zkrátit dobu učení a tím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

3.1.4. *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 interruptů 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.

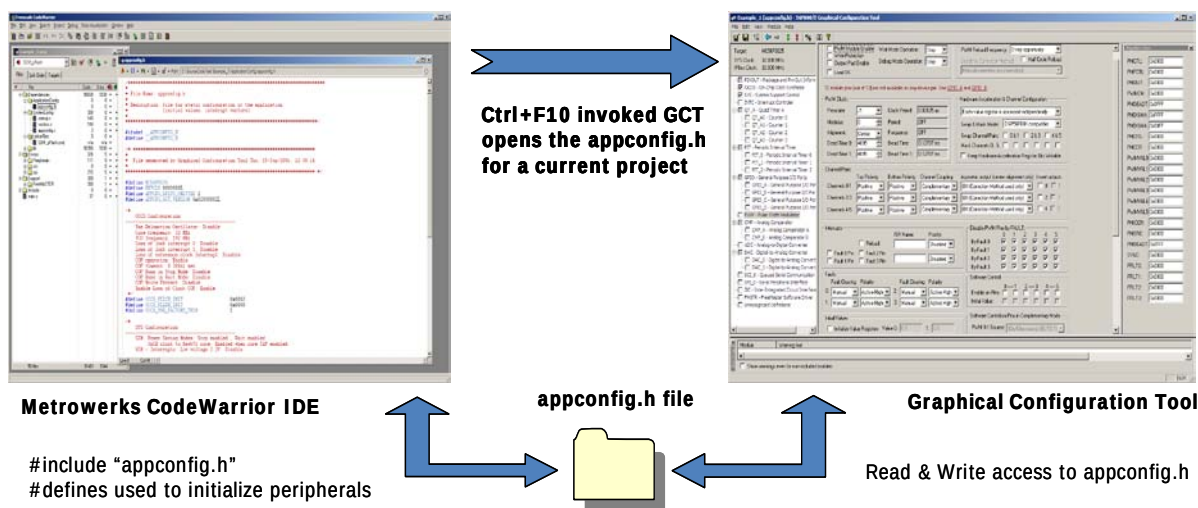
GCT je přímo volatelný z prostředí CodeWarrior, což v důsledku značně zjednodušuje práci s GCT nástrojem. Nastavení demonstruje Obr. 3.4.



Obr. 3.4: Intergace GCT do prostředí Metrowerks CodeWarrior

3.3. Spolupráce mezi GCT a nástrojem Quick Start

Spolupráci mezi nástrojem GCT a Quick Start demonstruje následující obrázek



Obr. 3.5: Spolupráce mezi nástroji GCT a Quick Start

Z obrázku je zřejmé, jak funguje nastavování vlastností čipu pomocí nástroje GCT. Prošředí 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.

3.4. 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     Pll;       /* 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

```

3.5. ioctl command – Input Output Control

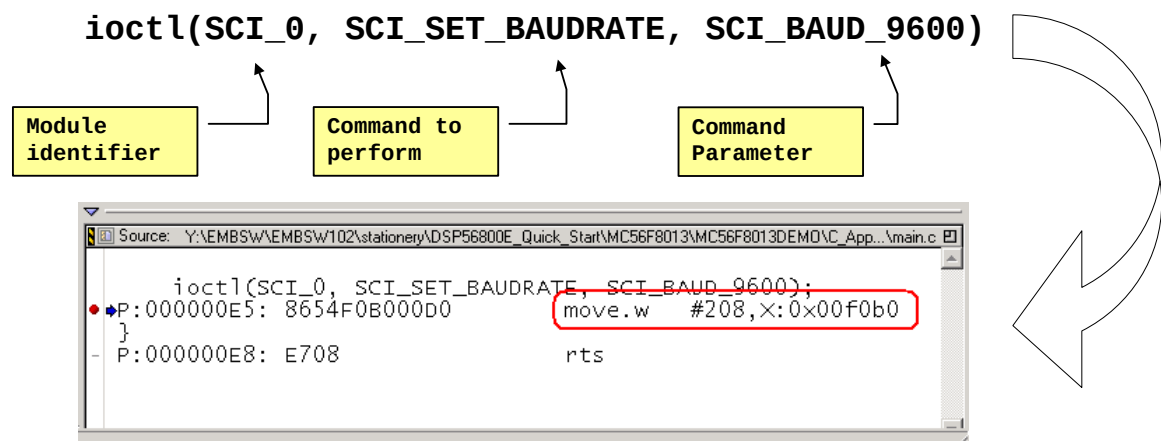
Jedním ze základních pilířů nástroje Quick Start jsou tzv. ioctl commands. Jedná se o low-level ovladače, pomocí nichž řídíme funkčnost mikropočítače za běhu programu.

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 „ioctl command“ je následující:



Obr. 3.6: ioctl command a jeho zpracování překladačem

4. 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.

4.1. 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 tool může sloužit jako:

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

4.1.1. 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 stacku. 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 byla umožněna 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

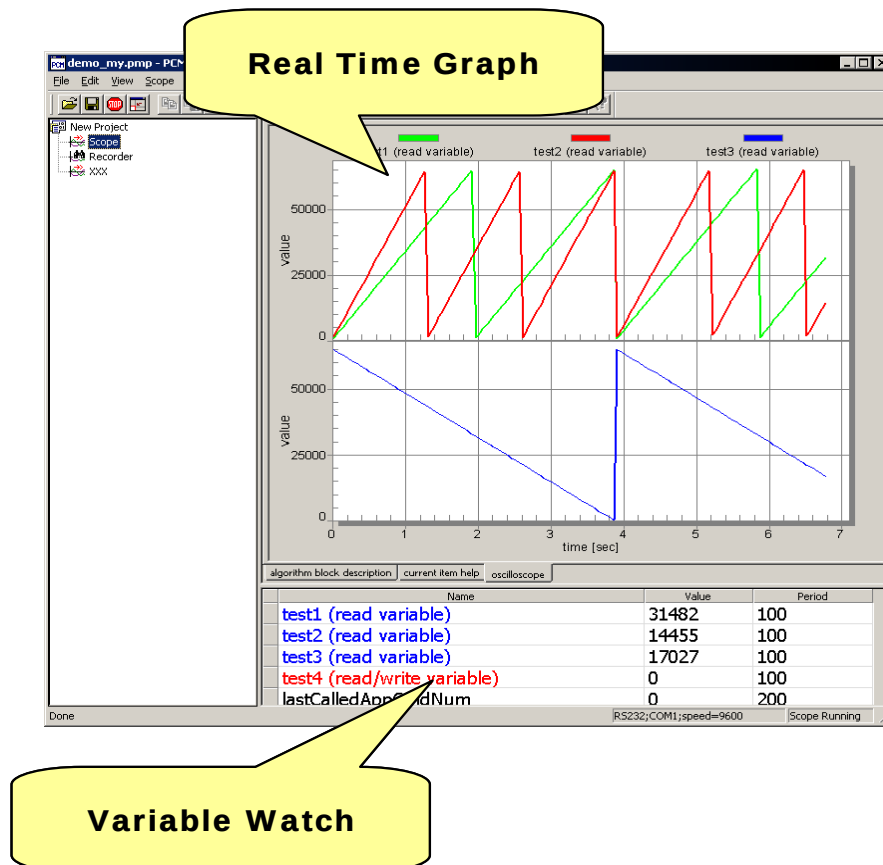
FreeMASTER tool dokáže zobrazovat proměnné flexibilním způsobem v závislosti na nastavení. Možnosti jsou následující:

- 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 ovladačem na embedded straně
- Aplikační příkazy

Ukázka typického zobrazení FreeMASTERu:



Obr. 4.1: FreeMASTER – typické zobrazení

4.1.2. *FreeMASTER tool – grafický řídicí panel*

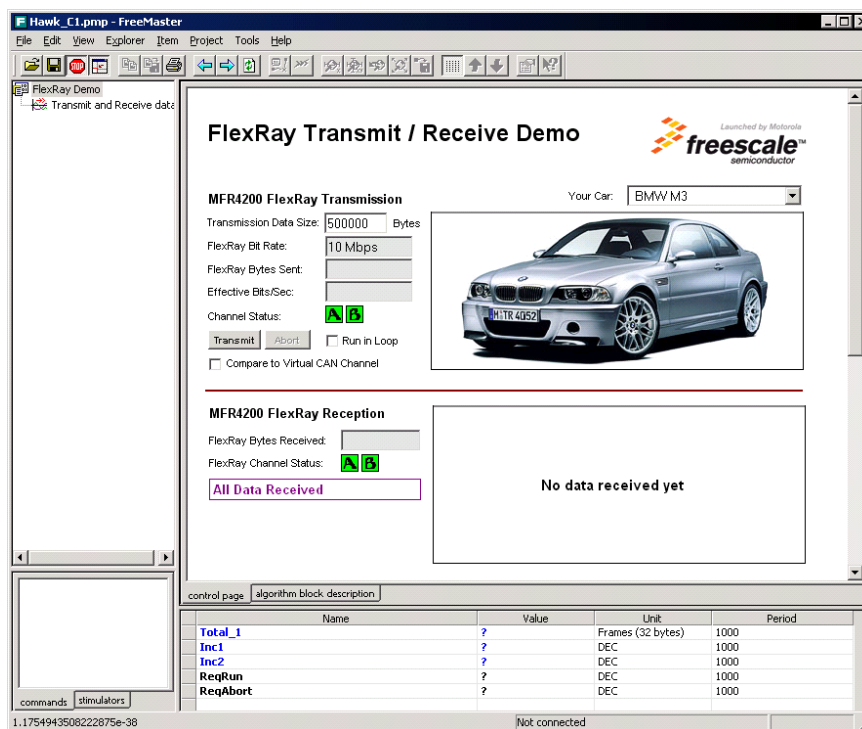
FreeMASTER PC strana umožňuje kromě standardní manipulace s proměnnými i poněkud komplikovanější zobrazení a modifikaci dat. FreeMASTER tool umožňuje vytvářet uživatelské grafické rozhraní vytvářením HTML stránek, spouštět scripty (JScript nebo VBScript), kreslit, animovat prvky jako push buttons, obrázky,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

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

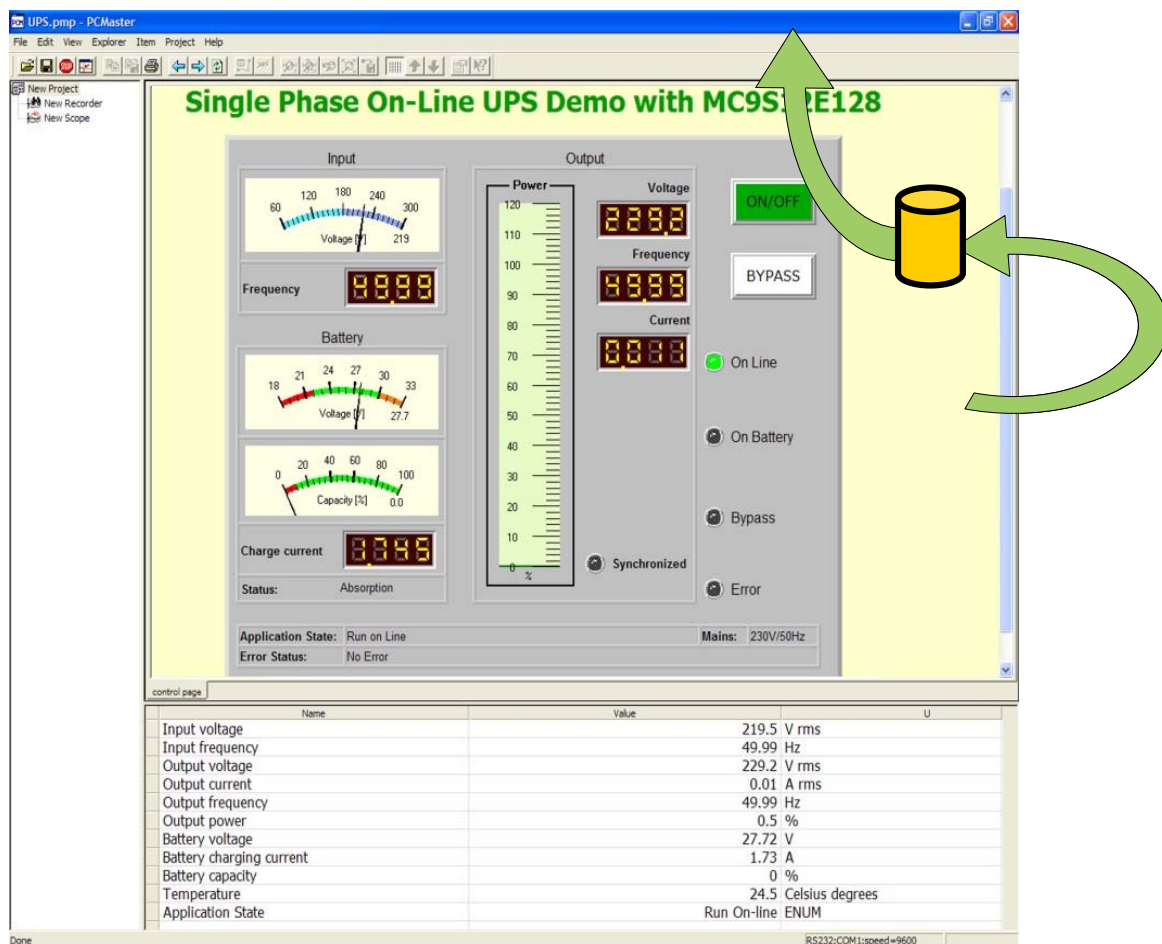
HTML může také spouštět celé aplikace, např.: Excel.

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

Ukázky možností použití FreeMASTER toolů jsou zobrazeny na následujících obrázcích.



Obr. 4.2: FreeMASTER – HTML



Obr. 4.3: 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.

5. FREESCALE LIBRARY

*Freescale poskytuje volně šiřitelné knihovny funkcí. Algoritmy jsou členěné do separátních knihoven podle typu implementovaných algoritmů. Algoritmy jsou napsány v assembleru a jejich rozhraní 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
-

5.1. 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_Pwmlct – 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

5.2. 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.

5.3. 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

5.4. 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 estimující indukované napětí ve stojících α , β souřadnicích
- ACLIB_PMSMBemfObsrvDQ – pozorovatel estimující 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_Itegrator - numerická integrace

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

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