Fast BLDC Motor Controller with 16 Bit CPU E523.06

PRODUCTION DATA – Apr 25, 2018

RoHS compliant

Features

- High integration device with Gate drive circuit for B6-NMOS bridge and 16 bit MCU
- IC supply voltage range 7 to 28V (extended 5V to 42V)
- CPU 16 bit, 4 - 48MHz for application tasks
- 32 kByte FLASH, ECC protected
- 24 kByte ROM
- 4.25 kByte SRAM, parity protected
- Typical deep-sleep mode current 20µA
- 2nd window watchdog and two independent clocks
- LIN2.x or bidirectional PWM Interface
- Flashable-via-LIN (high speed) for end-of-line programming
- High speed current amplifier for single shunt FOC
- Motor over-current protection with CPU interrupt
- 6 * FET short-circuit protection within gate driver
- Multiplication(16*16bit) & division(32/16bit) modules
- Coprocessor for PWM waveform generation
- Coprocessor for ADC tasking
- ROM library: LIN2.x stack routines
- ADC (SAR) 10bit 1Msample/s (12bit internal)
- S&H time programmable down to 167nsec
- 4 channel 16bit PWM generation + 8bit prescaler for left-, right-, center-aligned or free form toggle PWM
- 4 channel 16bit timer/capture/compare unit
- Clock fine tuning for EMC optimization
- Hardware support for spread spectrum
- Clock adjustment to LIN master possible
- AEC-Q-100 grade 0 qualified (Tamb=150°C)

Applications

- EC, BLDC, PMSM motors 50W to ~1500W
- Cooling fans, HVAC fans, positioning systems
- Fuel, hydraulic, oil and water pumps
- Universal LIN2.x slave with 6 push-pull outputs

General Description

E523.06 is a BLDC motor system-in-a-chip including a 16bit CPU core. It controls 3 NMOS half-bridges for driving BLDC motors, DC motors, or other loads. CPU architecture and motor driver peripherals are optimized for single shunt FOC (Field Oriented Control).

The IC includes a high-speed single shunt foot current measurement and protects against over-current (threshold continuously adjustable), over-temperature, over- and under-voltages and short-circuits (on the fly programmable thresholds for each FET). End-of-line programming is possible via JTAG or high-speed LIN. In-vehicle updates can be made using the bootloader located in ROM.

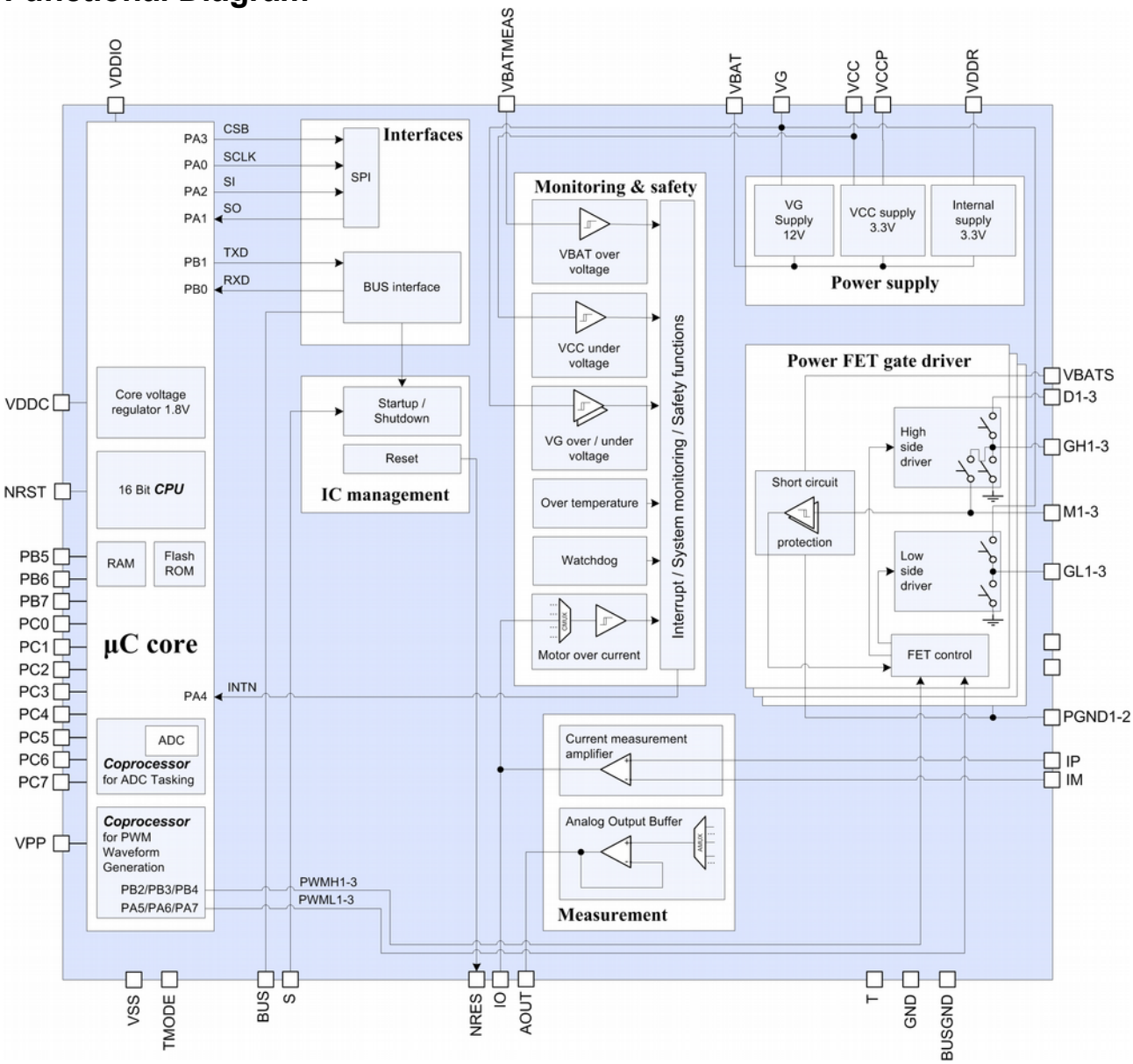
Highest performance is provided by a 16bit CPU. A co-processor for ADC tasking automatically collects all analog system information synchronously to the output PWM. These processing units optimize system performance, system reliability, EMC performance, current dissipation and development time.

The system clock is tunable in very fine steps to improve EMC behaviour and spread spectrum is supported by on-chip hardware. An adjustment of the system clock to a LIN-master is possible.
The E523.06 is suited for all commutation algorithms such as trapezoid, CZCD (Current Zero-Crossing Detection) and FOC (Field-Oriented Control). Single-shunt FOC is supported by the on-board high speed current amplifier and SARADC co-processor.

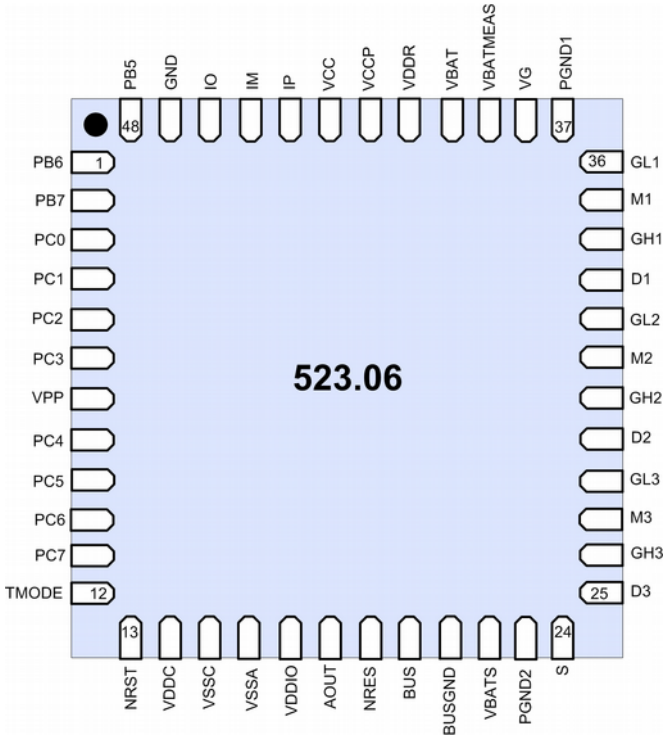
Ordering Information

Product ID	*Temp. Range*	*Package*
E52306A178B	-40°C to +150°C	QFN48L7

Functional Diagram



Pin Configuration



Pin Description

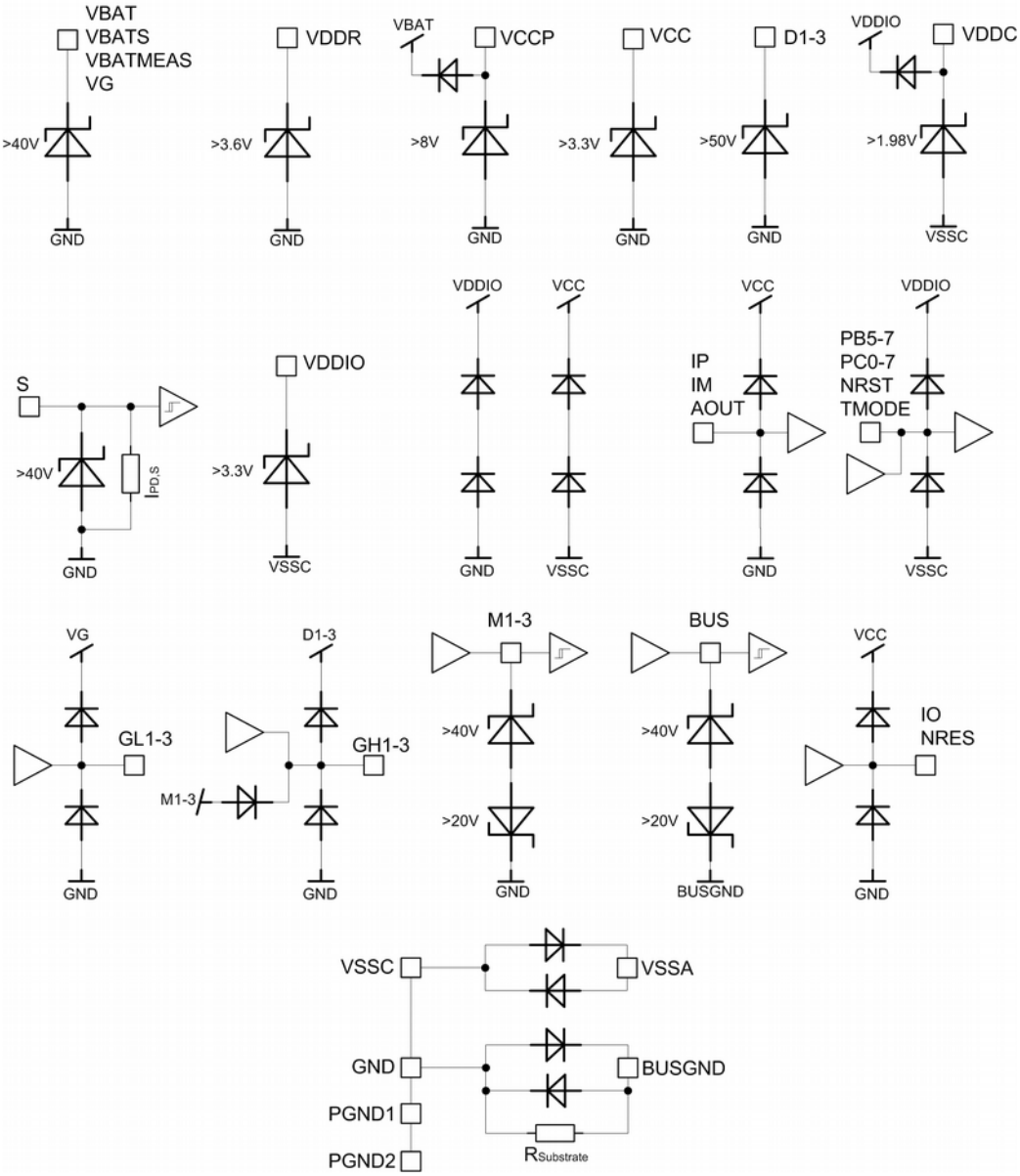
No	Name	Type	Description
1	PB6	AD_IO	DIO/AIN of µC core
2	PB7	AD_IO	DIO/AIN of µC core
3	PC0	AD_IO	DIO/AIN of µC core, JTAG TCK
4	PC1	AD_IO	DIO/AIN of µC core, JTAG TDA
5	PC2	AD_IO	DIO/AIN of µC core
6	PC3	AD_IO	DIO/AIN of µC core
7	VPP	HV_A_IO	FLASH test pad
8	PC4	AD_IO	DIO/AIN of µC core
9	PC5	AD_IO	DIO/AIN of µC core
10	PC6	AD_IO	DIO/AIN of µC core
11	PC7	AD_IO	DIO/AIN of µC core
12	TMODE	D_I	Test mode, to be connected to ground in application (pull-down)
13	NRST	D_IO	Reset for µC core (pull-up, bidirectional active low reset, open-drain driver)
14	VDDC	S	Core supply pad (internal regulator: 1.8V) of µC core for default buffer cap
15	VSSC	S	VSSC core ground pad of µC core for DC ground current
16	VSSA	S	VSSA analog ground pad of µC core

No	Name	Type	Description
17	VDDIO	S	IO supply pad (3.3V) of µC core
18	AOUT	A_O	Output of analog signal measurement
19	NRES	D_O	Microcontroller reset (active low, open-drain)
20	BUS	HV_A_IO	Bus interface (LIN or PWM)
21	BUSGND	S	Bus interface ground
22	VBATS	A_I	Motor bridge supply sense
23	PGND2	S	Power ground
24	S	A_I	KL15 wake-up
25	D3	S	High side supply, half bridge 3
26	GH3	A_O	High side gate drive output, half bridge 3
27	M3	A_I	Motor phase, half bridge 3
28	GL3	A_O	Low side gate drive output, half bridge 3
29	D2	S	High side supply, half bridge 2
30	GH2	A_O	High side gate drive output, half bridge 2
31	M2	A_I	Motor phase, half bridge 2
32	GL2	A_O	Low side gate drive output, half bridge 2
33	D1	S	High side supply, half bridge 1
34	GH1	A_O	High side gate drive output, half bridge 1
35	M1	A_I	Motor phase, half bridge 1
36	GL1	A_O	Low side gate drive output, half bridge 1
37	PGND1	S	Power ground
38	VG	S	Power FET gate driver voltage supply
39	VBATMEAS	HV_A_I	Battery sense voltage
40	VBAT	S	Main power supply (battery)
41	VDDR	S	Internal 3.3V voltage regulator
42	VCCP	A_O	VCC regulator driver output
43	VCC	A_I	Microcontroller - IC interface voltage and feedback for VCC regulator
44	IP	A_I	Motor current measurement amplifier (positive terminal)
45	IM	A_I	Motor current measurement amplifier (negative terminal)
46	IO	A_O	Motor current measurement amplifier (output)
47	GND		
48	PB5	AD_IO	DIO/AIN of µC core
EP	expose die		back plate; for thermal connection

Explanation of Types: A = Analog, D = Digital, S = Supply, I = Input, O = Output

Note: GND, BUSGND, PGND1, PGND2 have to be connected externally and with die paddle in shortest way.

Pin Clamping Illustration



Functional Safety

The development of this product is based on a process according to an ISO/TS 16949 certified quality management system. Functional safety requirements according to ISO 26262 have not been submitted to Elmos and therefore have not been considered for the development of this product.

Implementation of functional safety requirements according to ISO 26262 will have a significant impact on the development process and the technical concept. Therefore, in case functional safety requirements according to ISO 26262 are submitted, both parties have to agree on a DIA (Development Interface Agreement) before start of development.

1 Operating Conditions

1.1 Absolut Maximum Rating

- Operating the device at or beyond these limits may cause permanent damage.
- All voltages are referred to ground (0V).
- Currents flowing into the circuit have positive values.

Table 1.1-1: Absolute Maximum Ratings

No.	Description	Condition	Symbol	Min	Max	Unit
1	Storage Temperature		T_S			°C
2	Storage temperature mission profile	T_S < 55°C	$t_{T,S,55}$	15		years
3	Storage temperature mission profile	T_S < 125°C	$t_{T,S,125}$	2000		h
4	Storage temperature mission profile	T_S ≤ 150°C	$t_{T,S,150}$	300		h
5	Junction Temperature		$T_{junction}$	-40	125	°C
6	Junction Temperature	Excluding Flash programming	$T_{junction}$	-40	150	°C
7	Junction temperature mission profile[1)]	T_J < 23°C	$t_{T,J,23}$	3120		h
8	Junction temperature mission profile[1)]	T_J < 100°C	$t_{T,J,100}$	7800		h
9	Junction temperature mission profile[1)]	T_J ≤ 150°C	$t_{T,J,150}$	1080		h
10	Thermal resistance junction to case[2)]		$R_{th,J-C,QFN}$	-	5	K/W
11	VBAT, VBATS, VG, S voltage		V_{VBAT}	-0.3	40	V
12	VBAT, VBATS, VG, S voltage transient	t<500ms	V_{VBAT}	-	42	V
13	VCC voltage		V_{VCC}	-0.3	5.5	V
14	VCCP voltage		V_{VCCP}	-0.3	8	V
15	VDDR voltage		V_{VDDR}	-0.3	3.6	V
16	NRES voltage		V_{NRES}	-0.3V	V_{VCC}+0.3V	-
17	NRES current		I_{NRES}	-5	5	mA
18	D1-3 voltage		V_{D1-3}	-0.3	50	V
19	D1-3 voltage transient	t<500ms	V_{D1-3}	-	52	V
20	D1-3 to M1-3 voltage		V_{D1-3} - V_{M1-3}	-	40	V
21	D1-3 to M1-3 voltage transient	t<500ms	V_{D1-3} - V_{M1-3}	-	42	V
22	GH1-3 voltage		V_{GH1-3}	V_{M1-3}-0.3V	V_{D1-3}+0.3V	-
23	GH1-3 voltage		V_{GH1-3}	-0.3	50	V
24	GH1-3 voltage transient	t<500ms	V_{GH1-3}	-	52	V
25	GH1-3 static current, on-state[3)]		$I_{on,GH1-3}$	-75	75	mA
26	GH1-3 peak current, on-state[3)]	t < 10us	$I_{on,GH1-3}$	-400	400	mA
27	M1-3 phase voltage		V_{M1-3}	-10V	V_{BATS}+3V	-
28	M1-3 phase voltage		V_{M1-3}	-	40	V
29	M1-3 phase voltage transient	t<500ms	V_{M1-3}	-	42	V

No.	Description	Condition	Symbol	Min	Max	Unit
30	GL1-3 voltage		V_{GL1-3}	-0.3V	V_{VG}+0.3V	-
31	GL1-3 voltage		V_{GL1-3}	-0.3	40	V
32	GL1-3 voltage transient	t<500ms	V_{GL1-3}	-	42	V
33	GL1-3 static current, on-state[3)]		$I_{on,GL1-3}$	-75	75	mA
34	GL1-3 peak current, on-state[3)]	t < 10us	$I_{on,GL1-3}$	-400	400	mA
35	Current sum of all on-switched GL1-3		$I_{GL1-3,sum,on}$	-170	170	mA
36	Current sum of all off-switched GH1-3 and GL1-3		$I_{GH1-3,GL1-3,sum,off}$	-340	340	mA
37	IP voltage		V_{IP}	-0.3V	V_{VCC}+0.3V	-
38	IM voltage		V_{IM}	-0.3V	V_{VCC}+0.3V	-
39	IO, AOUT voltage		V_{IO}, V_{AOUT}	-0.3V	V_{VDDR}+0.3V	-
40	IO, AOUT current		I_{IO}, I_{AOUT}	-5	5	mA
41	VIN voltage		V_{VIN}	-0.3V	40	V
42	VIN voltage	$\|I_{VIN}\|$ < 3mA (series resistance >1kΩ required)	V_{VIN}	-3V	40	V
43	BUSGND shift to GND[4)]		V_{BUSGND}	-0.3	0.3	V
44	BUS voltage[5)]		V_{BUS}	-24	40	V
45	VDDIO IO supply voltage		V_{DDIO}	-0.3	3.6	V
46	VDDC digital core supply voltage		V_{DDC}	-0.3	1.98	V
47	IO pin voltage (PB5-7, PC0-7, NRST, TMODE)		V_{PORT}	-0.3	V_{DDIO}+0.3	V
48	VPP FLASH test pad voltage		V_{VPP}	0	14	V

[1)] According to various automotive conform medium- and high-temperature profiles, for other profiles contact Elmos
[2)] Based on simulation
[3)] Additional limitations due to IC total power dissipation have to be considered
[4)] To be connected on PCB
[5)] Circuity must comply with LIN standard

1.2 Recommended Operating Conditions

Table 1.2-1: Recommended Operating Conditions

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VBAT voltage		V_{VBAT}	7	13	28	V
2	VBAT voltage at cold crank (wake-up functionality limited)		V_{VBAT}	5	-	-	V
3	VCC load current		$-I_{VCCP}$	0	-	70	mA
4	NRES output current		I_{NRES}	-1	-	1	mA
5	VBATS voltage		V_{VBATS}	7	13	28	V

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
6	VBATS voltage at cold crank (short circuit protection limited)		V_{VBATS}	0	-	-	V
7	IP voltage		V_{IP}	0	-	1.5	V
8	IM voltage		V_{IM}	0	-	1.5	V
9	VDDIO IO supply voltage		V_{DDIO}	3	3.3	3.6	V
10	Motor current measurement amplification		A_{CM}	10	-	100	-

2 Electrical Characteristics

V_{VBAT} = 7V to 28V, for LIN parameters 7V to 18V, T_J = -40°C to +150°C, unless otherwise noted. Typical values are at V_{VBAT} = 12.0V and T_A = +25°C. Positive currents flow into the device pins.

2.1 Motor Driver

2.1.1 Design Description [E523.07A]

2.1.1.1 Thermal parameters

Table 2.1.1.1-1: Thermal parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Thermal resistance junction to case*) 1)		$R_{th,J-C}$	-	-	5	K/W

*) Not tested in production
1) Based on simulation

2.1.1.2 Power Supply and Management

Table 2.1.1.2-1: Current Consumption Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Current consumption in sleep mode	T_A < 50°C, V_{VBAT} = V_{VBATS} + $V_{VBAT-MEAS}$ = 12V	(I_{VBAT} + I_{VBATS} + $I_{VBATMEAS}$)$_{SLEEP}$	-	14	50	µA

2.1.1.2.1 VCC Microcontroller Supply

Table 2.1.1.2.1-1: VCC Supply Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VCC regulator output voltage1)	V_{VBAT} > 5V, -I_{VCC} < 55mA	V_{VCC}	3.1	3.3	3.5	V
2	VCC current limitation	V_{VCC} = 0V	$I_{VCC,lim}$	-120	-90	-60	mA
3	VCC reset threshold	Rising	$V_{VCC,UV,off}$	2.7	2.85	3.0	V
4	VCC reset threshold	Falling	$V_{VCC,UV,on}$	2.6	2.75	2.9	V
5	VCC reset threshold*)	Hysteresis	$V_{VCC,UV,hyst}$		0.1		V
6	VCC reset detection time		$t_{d,VCC,UV}$	2	6	10	µs

*) Not tested in production
1) I_{VCC} also limited due to maximum power dissipation

2.1.1.2.2 Power FET Gate Voltage Supply VG

Table 2.1.1.2.2-1: VG Supply Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VG regulator output voltage	V_{VBAT} >12V , -I_{VG} < 35mA	V_{VG}	11	12	13	V
2	VG regulator output voltage drop at low VBAT	5.5V < V_{VBAT}< 12V, -I_{VG} < 35mA	$V_{VG,drop,LV}$	-	-	0.6	V
3	VG current limitation		$I_{VG,lim}$	-110	-80	-35	mA

2.1.1.2.3 Internal Supply VDD

Table 2.1.1.2.3-1: Internal Supply Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Internal supply VDD voltage[1)]		V_{VDD}	3.0	3.3	3.6	V
2	Internal oscillator frequency		f_{OSC}	0.9	1	1.1	MHz

[1)] No external loads allowed

2.1.1.2.4 Chip Control

2.1.1.2.4.1 Wake Up

Table 2.1.1.2.4.1-1: Wake up electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	S (KL15) threshold voltage	V_{VBAT} > 7V	$V_{th,S}$	1.7	3.4	5	V
2	S (KL15) filter delay	V_{VBAT} > 7V	$t_{d,S}$	-	5	50	µs
3	Over all wake-up delay until controller is released (VG regulator is off)[*)]	Typical buffer capacities	$t_{WU,up}$	-	0.25	10	ms
4	S pull down current	V_S > 1.7V	$I_{pd,S}$	2	10	20	µA

[*)] Not tested in production

2.1.1.3 Power FET Gate Driver

Table 2.1.1.3-1: FET controller electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	On-resistance of high and low side drivers[1)]	Switched on	$r_{G1-3,on}$		8	20	Ω
2	On-resistance of high and low side drivers[2)]	Switched off	$r_{G1-3,off}$		4	10	Ω
3	Current consumption of high side drivers	Switched on	$I_{sup,D1-3,on}$	-	100	300	µA
4	Current consumption of high side drivers	Switched off	$I_{sup,D1-3,off}$	-	50	150	µA
5	Switch delay from input control signal to gate driver output		$t_{d,DRV}$	0	300	800	ns

[1)] $r_{GH1-3on}$, $r_{GL1-3on}$

[2)] $r_{GH1-3off,M}$, $r_{GH1-3off}$, $r_{GL1-3off}$

2.1.1.3.1 Short Circuit Protection

Table 2.1.1.3.1-1: Short circuit protection electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Minimum short circuit threshold	SCTH_HS/SCTH_LS = 3	$V_{th,SC,min}$	47.5	78	108.5	mV
2	Maximum short circuit threshold	V_{VBATS} > 5V, SCTH_HS/SCTH_LS = 80	$V_{th,SC,max}$	1.87	2.07	2.27	V

2.1.1.4 Measurement Functions

2.1.1.4.1 Fast Motor Current Measurement Amplifier

Table 2.1.1.4.1-1: Measurement amplifier electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Input offset after calibration	After calibration, 100mV < V_{IO} < V_{VDD} - 100mV, tested at V_{IP} = 300mV	$V_{offset,Amp}$	-1	0	1	mV
2	Input offset at typical calibration register value	CALIB_VAL = 0xA4	$V_{offset,typ}$	-20	0	20	mV
3	Input offset at minimum calibration register value	CALIB_VAL = 0x00	$-V_{offset,min}$	-	45	-	mV
4	Input offset at maximum calibration register value	CALIB_VAL = 0xFF	$-V_{offset,max}$	-	-23	-	mV
5	Input offset step size*)		$V_{offset,step}$		300		µV
6	Input leakage current	0V < V_{IP} < 0.3V, 0V < V_{IM} < 0.3V, $TEMP_j$ < 85°C	$I_{leak,in}$	-1	0	1	µA
7	Amplifier output resistance to rails (open loop)	$\|I_{IO}\|$ = 1mA	$r_{IO,Amp,ol}$	-	100	180	Ω
8	Amplifier settling time *)	A = 20, R_G= 18kΩ, V_{IO} = 300mV ... 2.7V	t_{AMP}	-	300	500	ns

*) Not tested in production

2.1.1.4.2 Analogue Signal Measurement

Table 2.1.1.4.2-1: Analog Signal Measurement Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VBAT measurement divisor		$RD_{VBATMEAS}$	11.5	12	12.5	-
2	M1-3 measurement divisor		RD_{M1-3}	11.5	12	12.5	-
3	VG measurement divisor		RD_{VG}	4.5	5	5.5	-
4	Temperature measurement tolerance*)	T_J > 125°C	$T_{err,HT}$	-11	0	11	K
5	AOUT buffer offset	100mV < V_{AOUT} < V_{VDD} - 100mV	$V_{offset,buf,AOUT}$	-10	0	10	mV

*) Not tested in production

2.1.1.5 Monitoring and Safety

2.1.1.5.1 VBAT Over Voltage

Table 2.1.1.5.1-1: VBAT over voltage parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VBAT over voltage threshold	Rising	$V_{VBAT,OV,off}$	29	30	31	V
2	VBAT over voltage threshold	Falling	$V_{VBAT,OV,on}$	28	29	30	V
3	VBAT over voltage threshold*)	Hysteresis	$V_{VBAT,OV,hyst}$		1		V
4	VBAT over voltage detection time		$t_{d,VBAT,OV}$	50	70	100	µs

*) Not tested in production

2.1.1.5.2 VG Over and Under Voltage

Table 2.1.1.5.2-1: VG over and under voltage parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VG over voltage threshold	Rising	$V_{VG,OV,off}$	13	14	15	V
2	VG over voltage threshold	Falling	$V_{VG,OV,on}$	12	13	14	V
3	VG over voltage threshold	Hysteresis	$V_{VG,OV,hyst}$		1		V
4	VG under voltage threshold for $V_{VBAT,min}$ > 7V (SAFECTRL[5] = 0)	Rising	$V_{VG,UV,off}$	6.0	6.4	6.8	V
5	VG under voltage threshold for $V_{VBAT,min}$ > 7V (SAFECTRL[5] = 0)	Falling	$V_{VG,UV,on}$	5.8	6.2	6.6	V
6	VG under voltage threshold for $V_{VBAT,min}$ > 7V (SAFECTRL[5] = 0)*)	Hysteresis	$V_{VG,UV,hyst}$		0.2		V
7	VG under voltage threshold for $V_{VBAT,min}$ > 6V (SAFECTRL[5] = 1)	Rising	$V_{VG,UV,6V,off}$	5.0	5.4	5.8	V
8	VG under voltage threshold for $V_{VBAT,min}$ > 6V (SAFECTRL[5] = 1)	Falling	$V_{VG,UV,6V,on}$	4.8	5.2	5.6	V
9	VG under voltage threshold for $V_{VBAT,min}$ > 6V (SAFECTRL[5] = 1)*)	Hysteresis	$V_{VG,UV,6V,hyst}$		0.2		V
10	VG over and under voltage detection time		$t_{d,VG}$	50	70	100	µs

*) Not tested in production

2.1.1.5.3 Motor Over Current

Table 2.1.1.5.3-1: Motor over current parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Motor over current comparator offset	1V < V_{IO} < 2.7V	$V_{offset,IO}$	-20	0	20	mV

2.1.1.5.4 Over temperature

Table 2.1.1.5.4-1: Over temperature parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Over temperature threshold1)	Rising	$T_{otemp,on}$	160	-	190	°C
2	Over temperature threshold1)	Falling	$T_{otemp,reset}$	140	-	-	°C
3	Over temperature threshold*) 1)	Hysteresis	$T_{otemp,hyst}$		15		°C
4	Over temperature detection time1)		$t_{d,otemp}$	50	70	100	µs

*) Not tested in production
1) Production test is based on calculation, not guaranteed by design

2.1.1.5.5 Watchdog

Table 2.1.1.5.5-1: Watchdog electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Watchdog first open window time*)		$WD_{T,FOW}$	230	256	282	ms
2	Watchdog closed window time*)	WDCTRL [7] = 1	$WD_{T,CW}$	5.4	8	8.8	ms
3	Watchdog open window time*)	WDCTRL [5:4] = 3	$WD_{T,OW3}$	115	128	143	ms

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
4	Watchdog open window time*)	WDCTRL [5:4] = 2	$WD_{T,OW2}$	57	64	73	ms
5	Watchdog open window time*)	WDCTRL [5:4] = 1	$WD_{T,OW1}$	28	32	38	ms
6	Watchdog open window time*)	WDCTRL [5:4] = 0	$WD_{T,OW0}$	14	16	20	ms
7	Watchdog reset activation duration*)		$WD_{T,RES}$	200	400	1000	µs

*) Not tested in production

2.1.1.6 Communication Interfaces
2.1.1.6.1 BUS interface (LIN or PWM)

Table 2.1.1.6.1-1: Bus interface parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	BUS dominant output voltage1)	TXD = 0, R_{PWM} = 1.3kΩ	$V_{BUS,DOM}$	-	0.7	2.0	V
2	BUS receiver dominant threshold		$V_{BUS,THDOM}$	-	-	0.4	V_{VBAT}
3	BUS receiver recessive threshold		$V_{BUS,THREC}$	0.6	-	-	V_{VBAT}
4	BUS receiver hysteresis*)		$V_{BUS,HYS}$	0.02	0.05	0.2	V_{VBAT}
5	BUS output current limitation		$I_{BUS,LIMIT}$	20	-	300	mA
6	BUS pull up resistance*)		$R_{BUS,SLAVE}$	10	40	100	kΩ
7	BUS wake-up debounce time		$t_{BUS,WU}$	70	-	150	µs
8	BUS dominant clamping time-out	IOCFG[7] = 1	$t_{BUS,LOW}$	3	9	12	ms

*) Not tested in production
1) External resistor to VBAT

Table 2.1.1.6.1-2: LIN electrical parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	LIN conformal functional range VBAT*)		$V_{LIN,VBAT}$	7	-	18	V
2	Recessive output voltage	TXD = 'H'	$V_{LIN,REC}$	V_{VBAT} - 1	-	V_{VBAT}	-
3	Dominant output voltage1)	TXD = 'L', V_{VBAT} = 7.0V, R_{LIN} = 500Ω	$V_{LIN,DOM}$	-	0.7	1.2	V
4	Dominant output voltage1)	TXD = 0, V_{VBAT} = 18V, R_{LIN} = 500Ω	$V_{LIN,DOM1}$	-	1.1	2.0	V
5	Receiver threshold	Falling edge	$V_{LIN,THDOM}$	0.4	-	0.6	V_{VBAT}
6	Receiver threshold	Rising edge	$V_{LIN,THREC}$	0.4	-	0.6	V_{VBAT}
7	LIN bus center voltage	$V_{LIN,BUSCNT}$= ($V_{LIN,THDOM}$+$V_{LIN,THREC}$)/2	$V_{LIN,BUSCNT}$	0.475	-	0.525	V_{VBAT}
8	Receiver hysteresis	$V_{LIN,THREC}$- $V_{LIN,THDOM}$	$V_{LIN,HYS}$	-	0.05	0.175	V_{VBAT}
9	Output current limitation	V_{LIN} > 2.5V	$I_{LIN,LIMIT}$	40	-	200	mA
10	Pull-up resistance		$R_{LIN,SLAVE}$	20	40	60	kΩ

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
11	Leakage current flowing into pin BUS	transmitter passive, $7V<V_{VBAT}<18V$, $7V<V_{LIN}<18V$, $V_{LIN}>V_{VBAT}$, T_J < 125°C	$I_{LIN,BUSREC}$	-	-	20	µA
12	Pull up current flowing out of pin BUS	transmitter passive, $7V<V_{VBAT}<18V$, $V_{LIN}=0V$	$I_{LIN,BUSDOM}$	-1	-	-	mA
13	Leakage current, ground disconnected (GND = VBAT)	$V_{VBAT}=13.5V$, $0V<V_{LIN}<18V$, T_J < 125°C	$I_{LIN,NOGND}$	-1	-	0.1	mA
14	Leakage current, supply disconnected	$V_{VBAT}=0V$, $0V<V_{LIN}<18V$, T_J < 125°C	$I_{LIN,NOBAT}$	-	-	20	µA
15	Input capacitance*)	$7V < V_{VBAT} < 18V$	$C_{LIN,PIN}$	-	-	30	pF
16	Receive propagation delay		$t_{RXD,PD}$	-	-	6	µs
17	Receive propagation delay symmetry		$t_{RXD,SYM}$	-2	-	2	µs
18	Duty cycle 1	$V_{LIN,THREC,max}=0.744*V_{VBAT}$, $V_{LIN,THDOM,max}=0.581*V_{VBAT}$, $V_{VBAT}=7\text{-}18V$, $t_{BIT}=50us$, $D_{LIN,1}=t_{BUSREC,min}/(2*t_{BIT})$	$D_{LIN,1}$	0.396	-	-	-
19	Duty cycle 2	$V_{LIN,THREC,min}=0.422*V_{VBAT}$, $V_{LIN,THDOM,min}=0.284*V_{VBAT}$, $V_{VBAT}=7.6\text{-}18V$, $t_{BIT}=50us$, $D_{LIN,2}=t_{BUSREC,max}/(2*t_{BIT})$	$D_{LIN,2}$	-	-	0.581	-
20	Duty cycle 3	$V_{LIN,THREC,max}=0.778*V_{VBAT}$, $V_{LIN,THDOM,max}=0.616*V_{VBAT}$, $V_{VBAT}=7\text{-}18V$, $t_{BIT}=96us$, $D_{LIN,3}=t_{BUSREC,min}/(2*t_{BIT})$	$D_{LIN,3}$	0.417	-	-	-

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
21	Duty cycle 4	$V_{LIN,THREC,min}$ =0.389*V_{VBAT}, $V_{LIN,THDOM,min}$ =0.251*V_{VBAT}, V_{VBAT}=7.6-18V, t_{BIT}=96us, $D_{LIN,4}$=$t_{BUSREC,max}$/ (2*t_{BIT})	$D_{LIN,4}$	-	-	0.590	-
22	LIN flash mode receive data baud rate*)	flash mode, V_{VBAT}=13V	$B_{LIN,RXD}$	-	115	-	kBd
23	LIN flash mode transmit data baud rate*)	flash mode, V_{VBAT}=13V	$B_{LIN,TXD}$	-	250	-	kBd

*) Not tested in production
1) External resistor to VBAT

2.2 Microcontroller

2.2.1 Design Description [E523.98A]

2.2.1.1 Analog Part

Table 2.2.1.1-1: Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	total supply current I_{DDIO} + I_{DDA} + I_{DDC} + I_{DDCA}	f_{core} = 48MHz IO current = 0 FLASH in active read ADC active	I_{DD_48MHz}	-	-	49.500	mA

2.2.1.1.1 Core Supply Regulator

Table 2.2.1.1.1-1: Voltage regulator: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	output voltage VDDC	ADC V_{REFH} trimmed	V_{OUT}	1.73	1.8	1.87	V

2.2.1.1.2 Oscillators and Reset

2.2.1.1.2.1 Power On Reset

Table 2.2.1.1.2.1-1: POR: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	power on threshold (monitors VDDA)		V_{POR}	1.75	2.28	2.68	V

2.2.1.1.2.2 Brownout Detection

Table 2.2.1.1.2.2-1: Brown Out: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VDDIO OK Threshold (rising edge)		$V_{DDIO\_OK\_RE}$	2.8	2.9	3.0	V

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
2	VDDIO Brown Out Threshold (falling edge)		$V_{DDIO\_OK\_FE}$	2.7	2.8	2.9	V
3	VDDIO_OK_RE - VDDIO_OK_FE (hysteresis)		$V_{DDIO\_OK\_HYST}$	50	100	200	mV

2.2.1.1.2.3 System Clock RC Oscillator

Table 2.2.1.1.2.3-1: Oscillators: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	48MHz output frequency	calibrated	$F_{OSC\_SYS\_48}$	46	48	49.5	MHz
2	frequency output range		F_{OSC_RANGE}	32		48	MHz

2.2.1.1.2.4 Watchdog Clock RC Oscillator

Table 2.2.1.1.2.4-1: Oscillators: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	output frequency		F_{OSC_WDOG}	0.6	0.8	1.0	MHz

2.2.1.1.2.5 NRST debouncer

Table 2.2.1.1.2.5-1: Debouncer: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	nReset signal debounce time pulses shorter than $T_{DEBOUNCE.min}$ will be suppressed pulses longer than $T_{DEBOUNCE.max}$ will pass the debouncer		$t_{DEBOUNCE}$	3.0	5.0	8.0	µs

2.2.1.1.3 SAR-ADC

Table 2.2.1.1.3-1: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	bandgap derived reference high voltage measured at ADC reference buffer output	ADC V_{REFH} trimmed	V_{REFH}	2.450	2.500	2.550	V
2	resolution (bit)*)		N	-	12	-	Bit
3	conversion rate*)	ADC clock = 12MHz	F_{CONV}	-	0.86	-	MSample/s
4	differential non-linearity*)	VREFL < VIN < VREFH	DNL	-1.0	-	3.0	LSB
5	integral non-linearity*)	VREFL < VIN < VREFH	INL	-4.0	-	4.0	LSB
6	Effective Number of Bits*)	VREFL < VIN < VREFH	ENOB	10	10.5	-	Bit
7	sampling time (number of ADC clock cycles) see ADC_CTRL.SAMPLE_EXT for additional information*)		$CYCLES_{SAMPLE}$	2	-	-	

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
8	conversion time (number of ADC clock cycles)*)		$CYCLES_{CONVERT}$	-	12	-	
9	ADC supply current when active	ADC ON	I_{ADC_SUPPLY}	-	3.0	3.5	mA
10	ADC warm-up time between ADC standby and run mode*)		$t_{WARM-UP}$	-	-	1	µs
11	ADC standby mode supply current	ADC in standby mode	$I_{ADC\_STANDBY}$	-	0.20	0.25	mA

*) Not tested in production

2.2.1.1.4 ADC Multiplexer

Table 2.2.1.1.4-1: ADC Multiplexer: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	on resistance (ESD protection + switch)		R_{ON}	-	400	800	Ω
2	temperature sensor voltage (ADC channel 24)*)	T = 25°C	VT		1.55		V
3	VT temperature coefficient*)		TC_{VT}		-3.1		mV/K

*) Not tested in production

Please contact Elmos for application note and further details.

2.2.1.1.5 IO Port Characteristics

Table 2.2.1.1.5-1: IO Pins: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	IO Supply Voltage (driving to external)*)		V_{DDIO}	3.0	3.3	3.6	V
2	Schmitt-Trigger Low to High Threshold Point		V_{TP}	-	-	2.0	V
3	Schmitt-Trigger High to Low Threshold Point		V_{TN}	0.8	-	-	V
4	Pull Up Resistor	at pin NRST V_{IN} = 0V	R_{PU}	30.00	48.00	74.00	kΩ
5	Pull Down Resistor	at pin TMODE V_{IN} = 3.3V	R_{PD}		60.00		kΩ
6	Low Level Output Voltage with smaller (drv_strength=0) driver strength.	I = 4.0mA	V_{OL_4}	-	-	0.4	V
7	Low Level Output Voltage with higher (drv_strength=1) driver strength.	I = 8.0mA	V_{OL_8}	-	-	0.6	V
8	High Level Output Voltage with smaller (drv_strength=0) driver strength.	I = -4.0mA	V_{OH_4}	2.4	-	-	V
9	High Level Output Voltage with higher (drv_strength=1) driver strength.	I = -8.0mA	V_{OH_8}	2.4	-	-	V
10	max current allowed to guaranty 80% output voltage*)	$V_{OH} > V_{DDIO} * 0.8$	I_{H80}	-	-	10	mA

*) Not tested in production

2.2.1.2 Digital Part
2.2.1.2.1 Sub Parts
2.2.1.2.1.1 LIN SCI Module (LINSCI)

Table 2.2.1.2.1.1-1: LIN Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Value of accuracy of the byte field detection*)		$t_{BFS,LINSCIModuleLINSCI}$		1/16	2/16	T_{BIT}
2	Earliest bit sample time t_{EBS} <= t_{LBS}*)		$t_{EBS,LINSCIModuleLINSCI}$	7/16			T_{BIT}
3	Latest bit sample time t_{EBS} <= t_{LBS}*)		$t_{LBS,LINSCIModuleLINSCI}$			10/16 - t_{BFS}	T_{BIT}

*) Not tested in production

3 Functional Description

3.1 Overview

The E523.06 is a programmable brush-less 3-Phase motor controller for supply voltages from 7V to 28V. The wide operating range makes it suitable for 12V automotive, industrial and consumer applications. During load dump situations the IC withstands peak voltages up to 42V.

It is realized as a dual die system consisting of a controller part and an driver part. In the following description the driver part is called "motor driver" and the microcontroller part is called "microcontroller".

The motor driver consists of an integrated linear regulator which provides efficient, reliable gate drive and can supply other loads (Hall Sensors) up to a total of 5mA current. For higher power requirements it is possible to attach an external booster transistor to increase maximum supply current.
The motor interface provides protection features such as under-voltage and short-circuit protection of the six Power FETs. An integrated current sense amp allows current control and over-current detection.

The microcontroller consists of an automotive rated 16bit RISC microcontroller with 32kByte of Flash Memory and 4.25kBytes of RAM. This allows the implementation of various motor control algorithms. The following digital modules take the computational load off the CPU:

- 16x16 bit multiply / divide and accumulate
- Concurrent hardware divider
- A table based Pre-PWM engine generates any modulation function such as SVM, flat bottom or sine
- Four 16 bit PWM generators with center and edge aligned modes and dead time insertion
- A sophisticated ADC controller with an autonomous sample sequencer synchronized to the PWM generators records sample sequences autonomously to memory

Eleven programmable IO pins provide various interface options with the motor and the outside circuitry. Two integrated SPI allow use of more complex communication interfaces.

The 12Bit 1MS/s successive approximation ADC can be used to sample all digital IO pins, the current sense amplifier and the back EMF interface consisting of the three motor connectors M[1-3] and the supply voltage VBAT.

Internal temperature monitoring and a thermally efficient QFN package enable the E523.06 to operate close to its maximum junction temperature.

3.1.1 Die to Die interface

The E523.06 is a dual die combination of the E523.07 motor driver and the Elmos motor controller E523.98. The following figure shows how the two ICs are interconnected.

Table 3.1.1-1: Die-to-die Interface

523.07 die pin	*523.98 die pin*
CSB	PA3
SCLK	PA0
SI	PA2
SO	PA1
TXD	PB1
RXD	PB0
INTN	PA4

523.07 die pin	*523.98 die pin*
PWMH1	PB2
PWMH2	PB3
PWMH3	PB4
PWML1	PA5
PWML2	PA6
PWML3	PA7

The most important interface data signals are the six PWM lines connected to port PB2/PB3/PB4 for the high-side gates and to port PA7/PA6/PA5 for the low-side gates.

Communication between the two chips is established via SPI lines CSB, SCLK, SI and SO for control and diagnosis. All driver settings like short-circuit detection thresholds, failure reactions, interrupt enables are programmed via this SPI interface. Firmware drivers are available in Elmos software development library.

Second serial interface of TX and RX pins is the LIN transceiver output to the microcontroller.

VDDIO needs to be connected to VCC on PCB, as well as NRST to NRES.

3.1.2 Software Development and Programming

A device specific IAR Programming environment including a programming/debug interface and sample application software is available through Elmos Sales and Support.

3.1.2.1 Programming Interface

External access to the microcontroller is provided through a two wire JTAG interface. The interface lines must be connected to the device as follows:

Table 3.1.2.1-1: JTAG interace connection

JTAG Line	*IC Pin Name*	*IC Pin Number*
VREF	VDDIO	16
TMODE	TMODE	12
TDA	PC1	4
TCK	PC0	3
NRSTI	NRST	13

3.2 Motor Driver

3.2.1 Design Description [E523.07A]

3.2.1.1 Power Supply and Management

The power supply contains:

- VCC linear voltage regulator for external microcontroller supply (3.3V, 50mA processor supply current, 5mA available for external use)
- VG linear voltage regulator for power FET switching (12V, 35mA)
- Power-on reset and microcontroller reset generation

- Power-up, sleep mode and wake-up sequencing

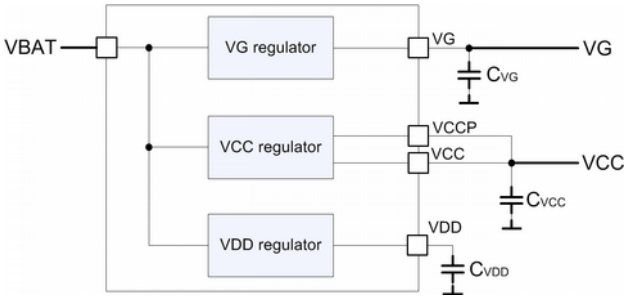


Figure 3.2.1.1-1: Power Supply Block Diagram

Table 3.2.1.1-1: Register **VREGCTRL** (0x07) VCC and VG supply control

	MSB							***LSB***
Content	-	-	-	-	-	-	VGON	VCCON
Reset value	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R/W	R/W
Bit Description	**VGON** : 1: Enable VG supply **VCCON** : 1: Enable VCC supply							

3.2.1.1.1 VCC Microcontroller Supply

The VCC regulator is a linear NMOS regulator with current limitation.

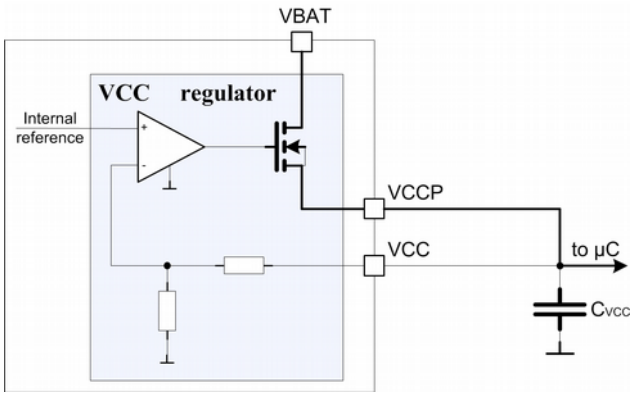


Figure 3.2.1.1.1-1: VCC regulator block diagram

To reduce power dissipation within the IC to allow higher ambient temperatures, an external NPN bipolar boost transistor may be inserted.

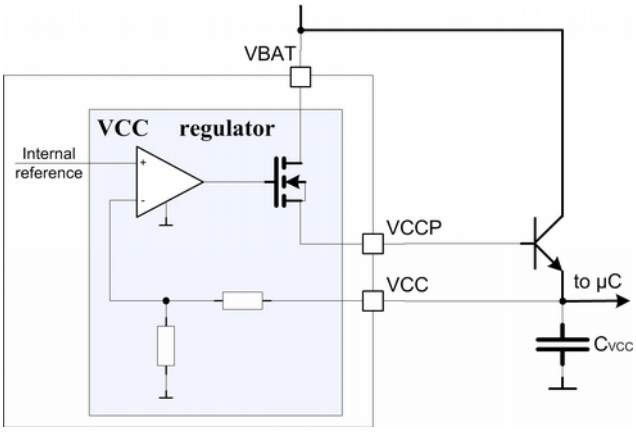


Figure 3.2.1.1.1-2: VCC block diagram, case external VCC boost transistor

Security warning: If using an external boost transistor an external current limitation must be inserted.

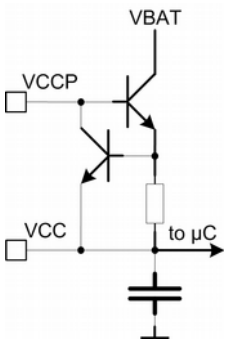


Figure 3.2.1.1.1-3: Proposal of short circuit protection at microcontroller supply, when using an external boost transistor

Note: The stability of the implemented circuitry has to be ensured by the customer.

3.2.1.1.2 Power FET Gate Voltage Supply VG

The VG regulator is used to:

- supply the low side gate drivers
- charge the bootstrap capacitors via external bootstrap diodes.

The VG regulator is a linear low drop PMOS regulator with current limitation. The output is safe against reverse currents.

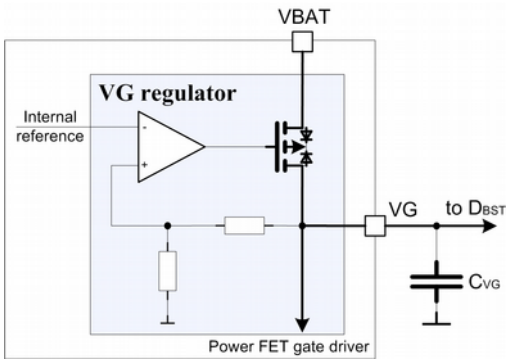


Figure 3.2.1.1.2-1: VG regulator block diagram

3.2.1.1.3 Internal Supply VDD

A linear 3.3V voltage regulator with current limitation supplies the internal digital and analogue components of the IC. External loads at VDD pin are not allowed.

3.2.1.1.4 Reset

The internal supply VDD and the microcontroller supply VCC are monitored by the IC. After power-up or in case of low voltage at VCC or VDD, the IC is reset and the signal at NRES pin for the microcontroller is set to L. The reset is set to H, if VDD and VCC is powered up successfully.

All reset sources are described in section Chip Control.

NRES pin is an open drain output.

3.2.1.1.5 Chip Control

The IC is clocked by an integrated oscillator with f_{OSC} frequency.

After power-up all device functions are disabled. The first successful watchdog access enables power fet driver and bus interface.

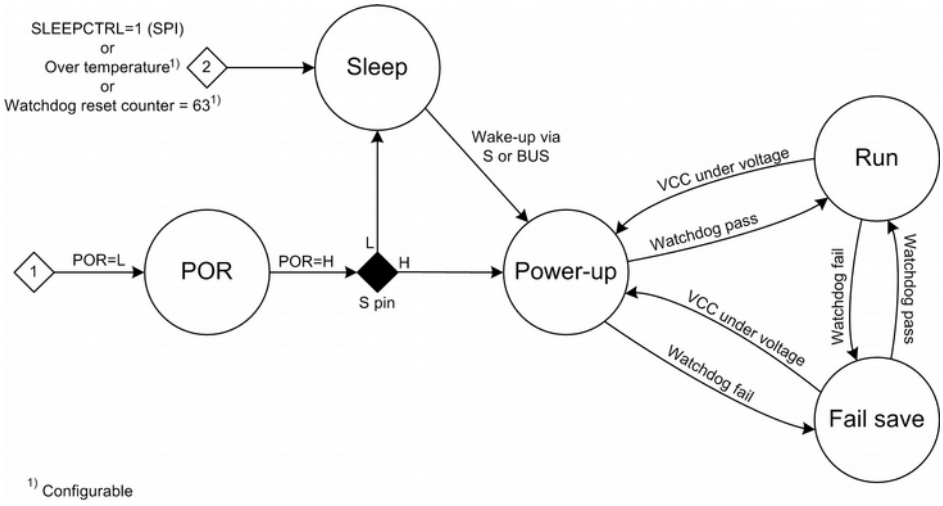


Figure 3.2.1.1.5-1: IC states diagram

Table 3.2.1.1.5-1: IC states description

	POR	*(Deep-) Sleep*	*Power-up*	*Run*	*Fail save*
Internal supply	on	on	on	on	on
Oscillator	off	off	on	on	on
VCC supply	off	off	on	on[2)]	on[2)]
VG supply	off	off	off	on[2)]	on[2)]
Power FET gate driver	disabled[1)]	disabled[1)]	disabled[1)]	enabled[2)]	disabled[1)]
Monitoring	off	off	on	on	on
Measurement	off	off	off	on	off
Bus (LIN / PWM)	off	off	off	on	on
Registers	reset	reset[3)]	writable	writable	reset[3)]
CHIPSTATE register content	-	0x00	0x11	0x13	0x12

1) High side and low side power FETs switched off
2) Depending on configuration
3) Except IRQSTAT1-2, WUSRC, RSTSRC, CHIPCTRL registers

Table 3.2.1.1.5-2: Chip Control Registers

Register Name	*Address*	*Description*
CHIPSTATE	0x01	Chip state
CHIPCTRL	0x02	Wake-up and sleep configuration
RSTSRC	0x0E	Reset sources
WUSRC	0x0F	Wake-up sources

Table 3.2.1.1.5-3: Register **CHIPCTRL** (0x02) Wake-up and sleep configuration

	MSB							*LSB*
Content	-	-	-	-	WDSLPEN	SSENS	BUSWA-KEEDGE	BUSWA-KEEN
Reset value	0	0	0	0	1	0	1	1
Access	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**WDSLPEN** : Enable watchdog sleep mode activation **SSENS** : 1: Wake up at S is level sensitive (high level leads to wake up), 0: Wake up at S is edge sensitive (rising edge) **BUSWAKEEDGE** : 1: BUS wake up at rising edge, 0: BUS wake up at falling edge **BUSWAKEEN** : BUS wake up enable							

Table 3.2.1.1.5-4: Register **RSTSRC** (0x0E) Reset sources

	MSB							***LSB***
Content	-	-	OT	SPI_SLP	WD_SLP	WD	VCC_UV	POR
Reset value	0	0						
Access	R	R	W1C	W1C	W1C	W1C	W1C	W1C
Bit Description	**OT** : Over temperature (cleared by writing 1) **SPI_SLP** : SPI sleep command sent (cleared by writing 1) **WD_SLP** : Watchdog sleep mode activation occured (cleared by writing 1) **WD** : Watchdog reset occured (cleared by writing 1) **VCC_UV** : VCC undervoltage occured (cleared by writing 1) **POR** : VDD undervoltage (power-on reset) occured (cleared by writing 1)							

Table 3.2.1.1.5-5: Register **WUSRC** (0x0F) Wake-up sources

	MSB							***LSB***
Content	-	-	-	-	-	-	WU_BUS	WU_S
Reset value	0	0	0	0	0	0		
Access	R	R	R	R	R	R	W1C	W1C
Bit Description	**WU_BUS** : BUS wake-up accured (cleared by writing 1) **WU_S** : S wake-up accured (cleared by writing 1)							

Table 3.2.1.1.5-6: Register **CHIPSTATE** (0x01) Chip state

	MSB							***LSB***
Content	STATE[7:0]							
Reset value	0							
Access	R							
Bit Description	**STATE[7:0]** : Current Chip State (see description in 3.2.1.1.5-1)							

3.2.1.1.5.1 Power Up

After applying battery voltage, the IC ensures a safe power-up of microcontroller and B6 bridge. All high side and low side driver outputs are off (clamped to ground or to motor phase) to prevent the external B6 bridge from cross current.

Sleep mode activation after power-up is **level sensitive** to S pin. If S pin is low, the IC enters sleep mode. If S pin is 'H', the IC powers up.

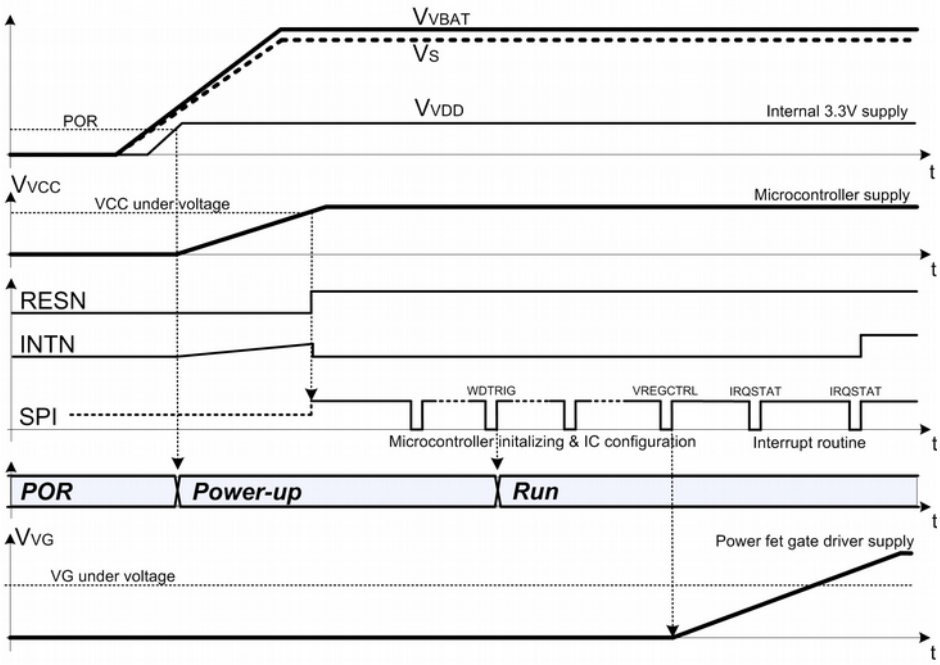


Figure 3.2.1.1.5.1-1: Power up timing diagram, typical scenario

3.2.1.1.5.2 Shut-down and Sleep Mode

Sleep mode (deep-sleep mode) is entered via SPI command written to SLEEPCTRL register. In sleep mode, all registers are cleared except the IRQSTAT1-2 (interrupt status), RESETSRC, WAKESRC, and CHIPCTRL registers. All regulators and monitoring are off except VDD and power-on reset (POR) generation.

Table 3.2.1.1.5.2-1: Register **SLEEPCTRL** (0x03) Sleep mode control[1)]

	MSB							***LSB***
Content	-	-	-	-	-	-	-	SLEEP
Reset value	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	W
Bit Description	**SLEEP** : Enter sleep mode							

[1)] Bits in register are self-clearing

3.2.1.1.5.3 Wake Up

Wake-up from sleep mode can be done by two independent wake-up sources:

- BUS pin: A falling edge at BUS pin followed by a dominant bus level maintained for a time period $t_{BUS,WU}$ results in a remote wake up request. If BUSWAKEEDGE bit in CHIPCTRL register is 0, the IC wakes up immediately. Else the IC wakes up at the next rising edge at BUS pin.
- S pin: A rising edge or high level at S pin wakes up the IC. The sensitivity can be configured by SSENS bit in CHIPCTRL register.

Warning: Wake-up from sleep mode via S pin with SSENS = 0 requires a rising edge at S pin after writing the SPI sleep command. In several applications it cannot be ensured under all circumstances. Then wake-up events may

not be recognized and the system remains in sleep mode (dead lock). To avoid this situation, SSENS bit in CHIPCTRL register has to be set to 1.

Table 3.2.1.1.5.3-1: BUS wake-up typical scenarios

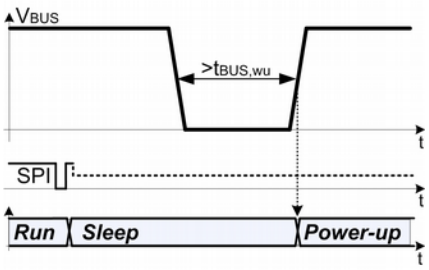
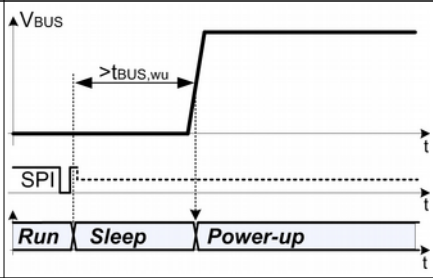
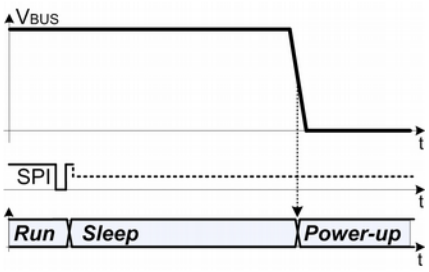
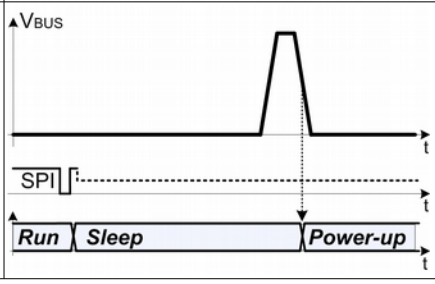
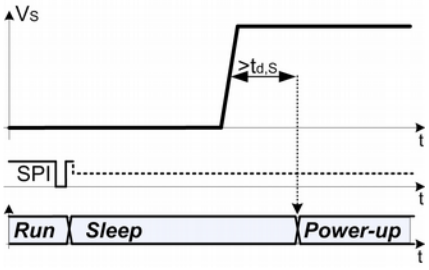
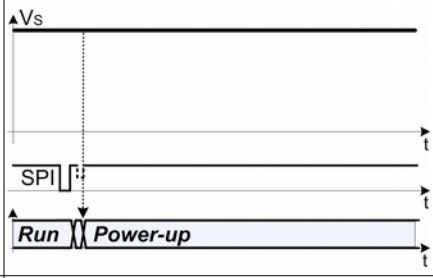
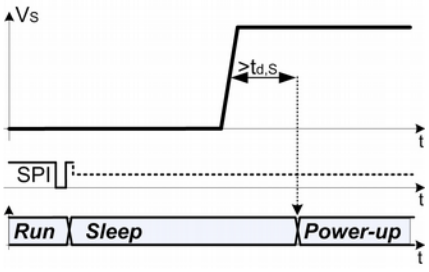
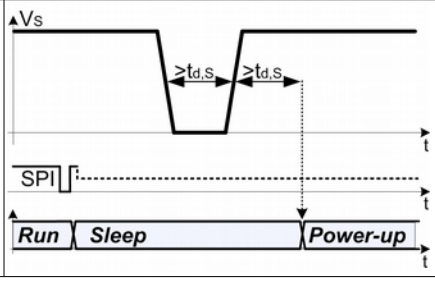
	Goto sleep with recessive BUS	*Goto sleep with dominant BUS*
CHIPCTRL[1] = 1 (wake-up at rising edge)		
CHIPCTRL[1] = 0 (wake-up at falling edge)		

Table 3.2.1.1.5.3-2: S wake-up typical scenarios

	Goto sleep with 'L' level S	*Goto sleep with 'H' level S*
CHIPCTRL[2] = 1 (level sensitive)		
CHIPCTRL[2] = 0 (edge sensitive)		

3.2.1.1.6 Board Level Protection

VBAT (VBATS), S and GND must be protected against reverse polarity and ISO pulses on PCB level. At VBAT (VBATS) filtering is necessary to prevent the IC from malfunction and destruction caused by EME and EMI.

3.2.1.2 Power FET Gate Driver

The power FET gate driver contains 3 independent half bridge gate drivers with short circuit protection. Each half bridge consists of a high side and a low side driver controlled by PWMH1-3 and PWML1-3 input pins.

The slew rate of the power FETs is adjusted by external gate resistors.

Note: The gate-drain capacitance of the power FET causes a positive gate-source voltage during motor phase transients. Additional PCB elements have to be inserted to prevent the half bridge from cross current:

- Greater off- resistance than on-resistance of RGATE (2 resistors, 1 diode, refer to typical application diagram, recommended)
- Additional gate-source capacitor, increases the VG current consumption and power losses, not possible for large power FETs

The high side supply voltage is done by bootstrap principle with external capacitors and diodes.

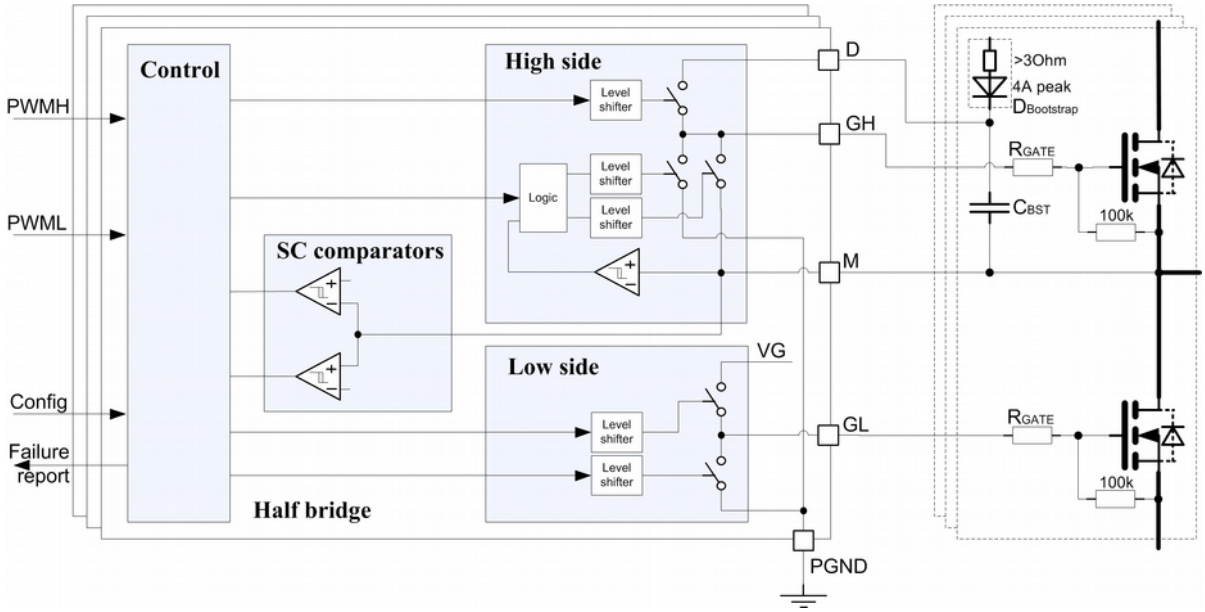


Figure 3.2.1.2-1: FET controller block diagram

Warning: The bootstrap principle and circuitry must be used even if a charge pump is inserted at PCB. It is not allowed to connect D1-3 pins to any charge pump terminal on PCB (neither directly nor via diodes).

Note: The bootstrap capacitors need to be recharged every PWM cycle. 100% duty cycle is not allowed. There is no bootstrap under voltage monitoring.

PWMH input pins control the high side drivers, PWML pins control the low side drivers independently. Dead time generation and cross current protection have to be implemented in the microcontroller.

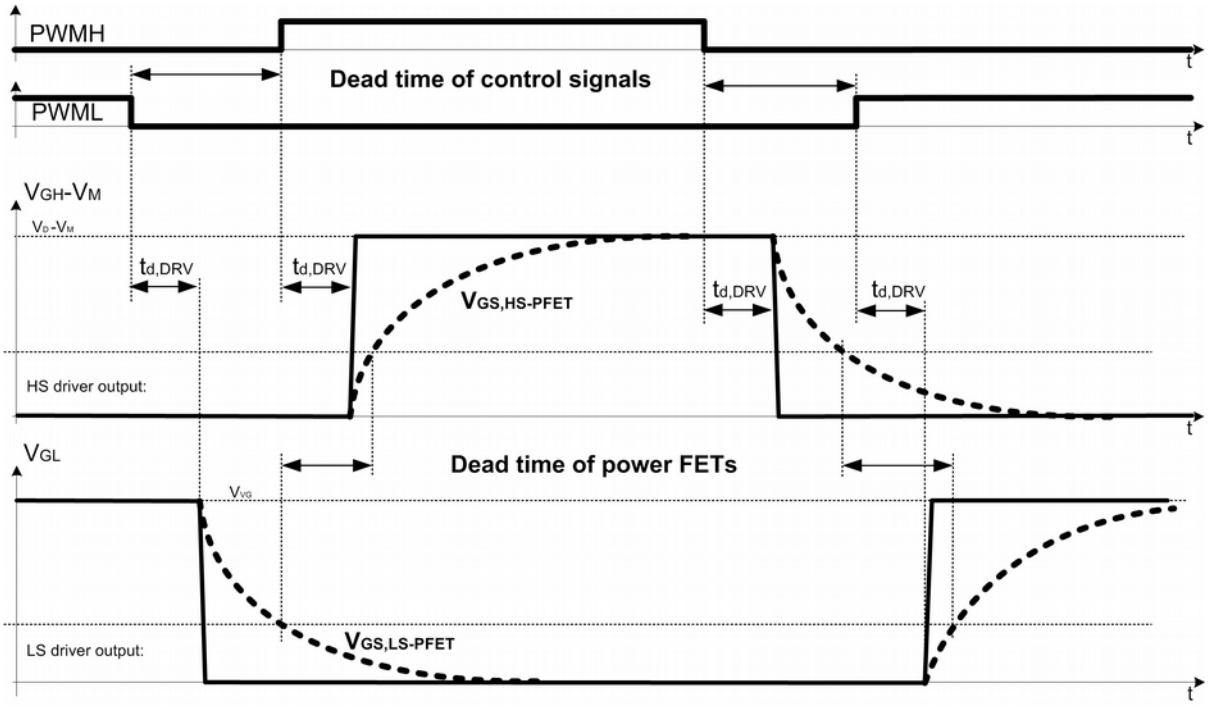


Figure 3.2.1.2-2: 6 PWM input mode transfer characteristic timing diagram

3.2.1.2.1 Short Circuit Protection

The drain source voltage of the external low side and high side power FETs in on-state are monitored by the IC. If the motor phase voltage M exceeds a programmable threshold, the gate driver switches the power FETs off immediately. Reference for the high side FETs is VBATS sense voltage, for the low side FETs ground.

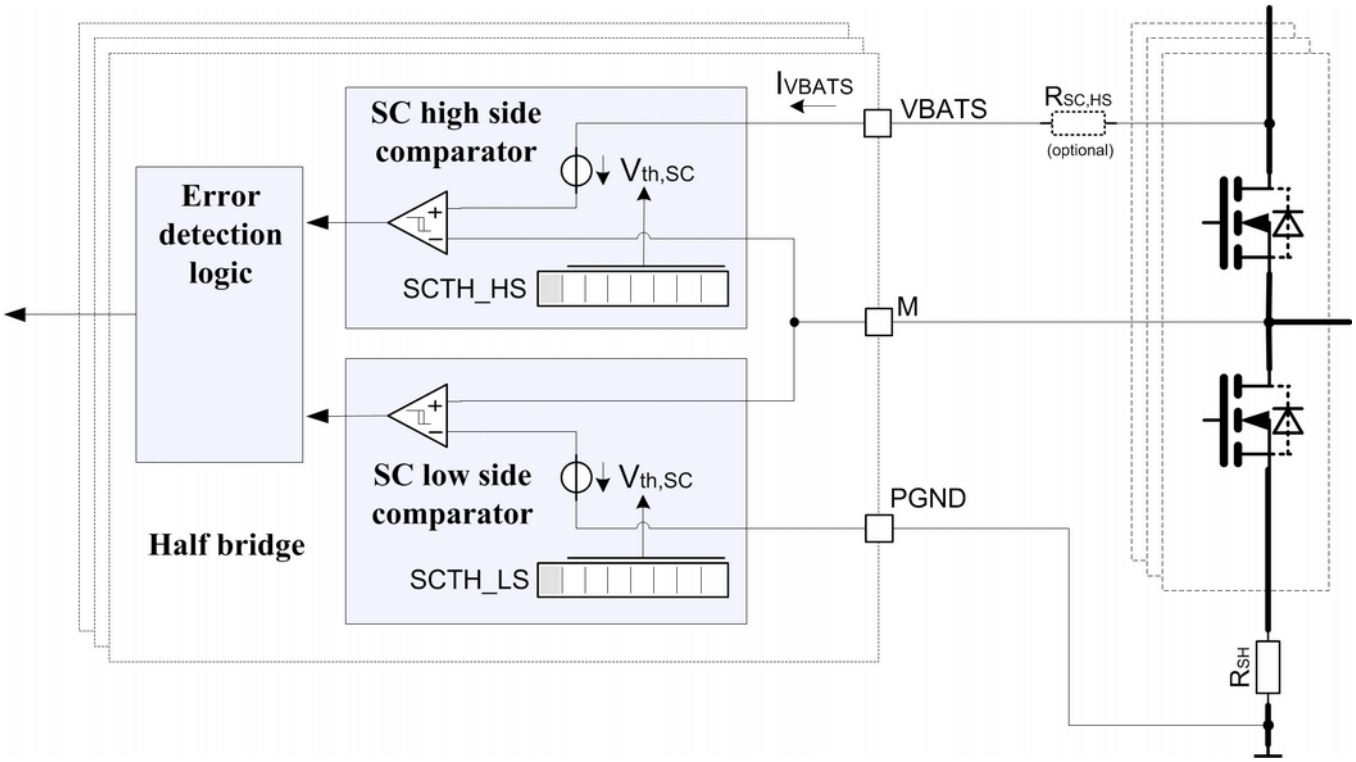
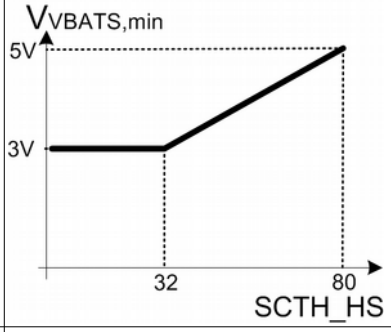


Figure 3.2.1.2.1-1: Block diagram

The thresholds of high side and low side power FETs are programmable in SCTH_HS and SCTH_LS registers. The calculation of the threshold values is given in the following table.

The short circuit protection can be disabled in SCEN register for each fet seperately.

Table 3.2.1.2.1-1: Short circuit threshold calculation

Side	*Mean value*	*Tolerance (+/-)*	*Minimum Battery Voltage*
High side	$\frac{SCTH\_HS}{128} \cdot V_{VDD}$	24mV	
Low side	$\frac{SCTH\_LS}{128} \cdot V_{VDD}$	24mV	-

After on-switching of the power FET the short circuit failure detection is masked for a programmable time:

$$t_{MT,SC} = SCMT \cdot \frac{1}{f_{CLK}}.$$

The short circuit failure debounce time is:

$$t_{DEB,SC} = (SCDEB + (0\_1)) \cdot \frac{1}{f_{CLK}}.$$

Table 3.2.1.2.1-2: Short circuit protection register table

Register Name	*Address*	*Description*
SCMT	0x26	Short circuit masking time selection
SCTH_HS	0x28	High side short circuit threshold selection
SCTH_LS	0x29	Low side short circuit threshold selection
SCEN	0x2C	Short circuit protection enable
SCDEB	0x2D	Short circuit debounce time

Table 3.2.1.2.1-3: Register **SCEN** (0x2C) Short circuit protection enable

	MSB							*LSB*
Content	-	-	SCEN_HS 3	SCEN_HS 2	SCEN_HS 1	SCEN_LS 3	SCEN_LS 2	SCEN_LS 1
Reset value	0	0	1	1	1	1	1	1
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**SCEN_HS3** : Highside 3 short circuit protection enable **SCEN_HS2** : Highside 2 short circuit protection enable **SCEN_HS1** : Highside 1 short circuit protection enable **SCEN_LS3** : Lowside 3 short circuit protection enable **SCEN_LS2** : Lowside 2 short circuit protection enable **SCEN_LS1** : Lowside 1 short circuit protection enable							

Table 3.2.1.2.1-4: Register **SCTH_HS** (0x28) High side short circuit threshold selection

	MSB							*LSB*
Content	-	D[6:0]						
Reset value	0	0b0001111						
Access	R	R/W						
Bit Description	**D[6:0]** : Short circuit threshold value (3...80).							

Table 3.2.1.2.1-5: Register **SCTH_LS** (0x29) Low side short circuit threshold selection

	MSB							*LSB*
Content	-	D[6:0]						
Reset value	0	0b0001111						
Access	R	R/W						
Bit Description	**D[6:0]** : Short circuit threshold value (3...80).							

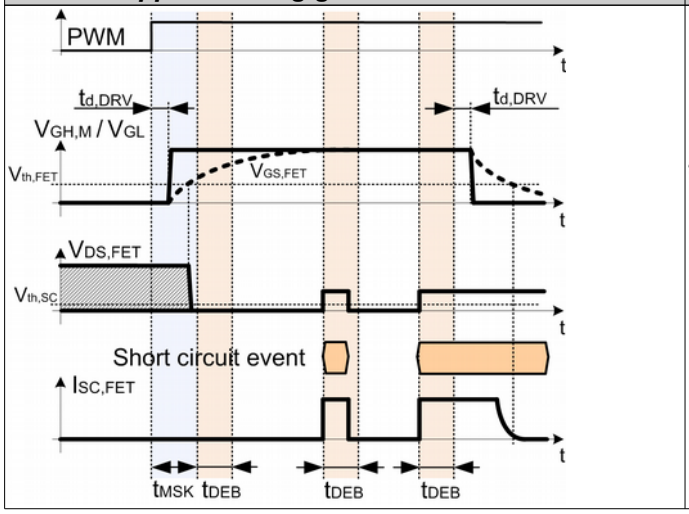
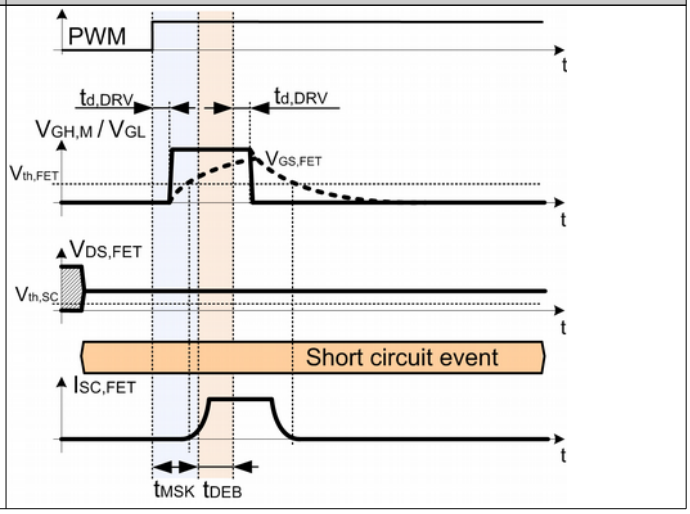
Table 3.2.1.2.1-6: Register **SCMT** (0x26) Short circuit masking time selection

	MSB							***LSB***
Content	-	-	-	-	MT[3:0]			
Reset value	0	0	0	0	0000			
Access	R	R	R	R	R/W			
Bit Description	**MT[3:0]** : Short circuit initial masking time value							

Table 3.2.1.2.1-7: Register **SCDEB** (0x2D) Short circuit debounce time

	MSB							***LSB***
Content	-	-	-	-	-	TDEB[2:0]		
Reset value	0	0	0	0	0	0		
Access	R	R	R	R	R	R/W		
Bit Description	**TDEB[2:0]** : Debounce time value							

Table 3.2.1.2.1-8: Short circuit detection examples

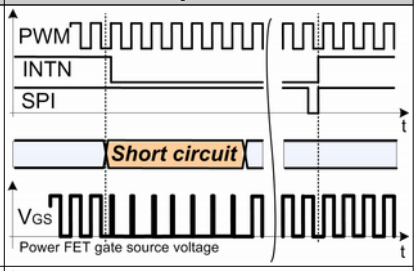
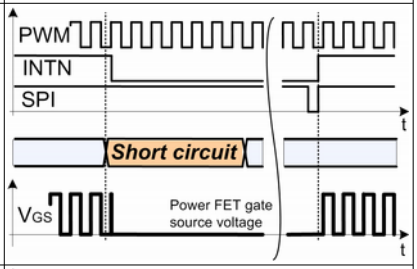
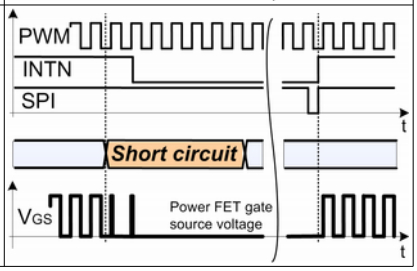
Regular gate driver on-switching, short circuit applied during gate driver on-state	***Gate driver on-switching during short circuit situation***
	

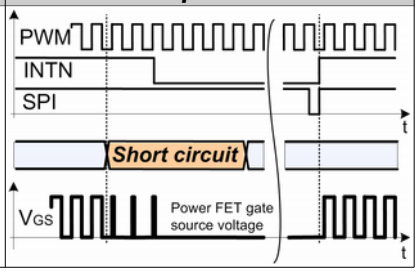
3.2.1.2.1.1 Superior Short Circuit Failure Reaction
A superior reaction to a short circuit failure can be configured in SCPCTRL register.

Table 3.2.1.2.1.1-1: Register **SCPCTRL** (0x0B) Short circuit protection control

	MSB							***LSB***
Content	MSK_OSIDE	MSK_SIDE	-	SCALL	-	-	SCFCT[1:0]	
Reset value	0	0	0	0	0	0	0	
Access	R/W	R/W	R	R/W	R	R	R/W	
Bit Description	**MSK_OSIDE** : 1: Apply masking of short circuit detection to all (on-switched) FETs **MSK_SIDE** : 1: Apply masking of short circuit detection to all (on-switched) FETs of the same side (on-switching of 1 half bridge -> all on-switched HS FETs are masked, off-switching -> all on-switched LS FETs are masked) **SCALL** : 1: Switch off all power FETs on superior short circuit failure. In case of SCFCT = 0, this bit has no affect. **SCFCT[1:0]** : refer to 3.2.1.2.1.1-2							

Table 3.2.1.2.1.1-2: Superior short circuit protection behaviour

SCFCT[1:0]	***Description***	***Depiction***
0	Re-try of power FET switching at every rising edge of the correspondent input control. Interrupt is thrown after first short circuit failure detection	
1	Drivers remain off until interrupt is cleared by the microcontroller	
2	One re-try of power FET switching allowed, after that all drivers remain off until interrupt is cleared by the microcontroller	

SCFCT[1:0]	Description	Depiction
3	Two re-tries of power FET switching allowed, after that all drivers remain off until interrupt is cleared by the microcontroller	

INTN: low-active interrupt

3.2.1.3 Measurement Functions

Measurement of motor current and internal analogue signals is available in the IC.

3.2.1.3.1 Fast Motor Current Measurement Amplifier

An integrated high speed operational amplifier with external elements measures the motor current via a low side shunt resistor. Offset and gain of the current measurement amplifier have to be adjusted by external resistors.

The amplifier output IO can be connected to an ADC input of the microcontroller. The maximum output voltage is limited by the internal 3.3V supply VDD.

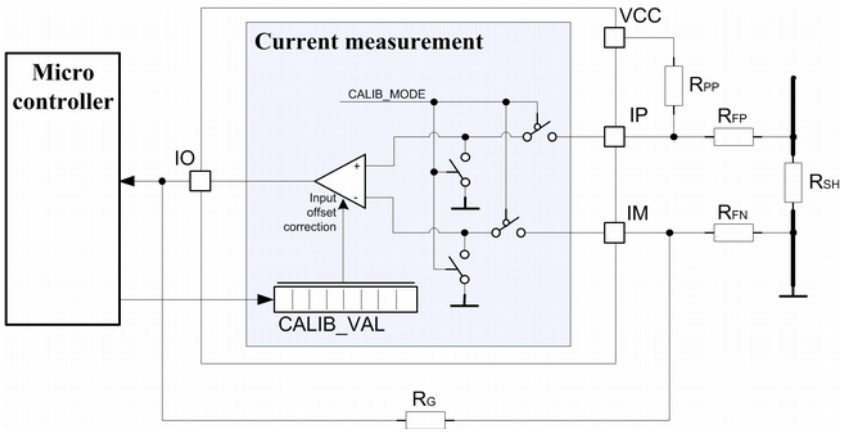


Figure 3.2.1.3.1-1: Motor current measurement amplifier circuitry

The offset of the operational amplifier has to be calibrated by the microcontroller.

Calibration routine example 1 (open loop):

- Set CALIB_MODE bit in OFFSET_CFG register to 1: The inputs IP and IM are disconnected and switched to ground internally (amplifier operates in open loop). Due to the high voltage amplification, the output IO can be used as digital signal (1: V_{VDD}, 0: V_{GND}) to detect a positive or negative offset of the amplifier.
- Perform offset calibration: Use software alghorithm to get register value for lowest offset, write it to OFFSET_VAL register.
- Set CALIB_MODE bit in OFFSET_CFG register to 0: Return to current measurement functionality.

Calibration routine example 2 (closed loop):

- Ensure, that there is no current flowing through the motor / shunt resistor
- Measure motor amplifier output at ADC of processor

- Perform offset calibration: Use software alghorithm to get register value for lowest offset, write it to OFFSET_VAL register.

Note: *The current measurement functionality is disabled during calibration.*

Note: *The over current interrupt flag is affected by calibration.*

Note: *Offset calibration needs to be reperformed at developing temperature, particularily when the offset target value is unequal 0mV.*

Table 3.2.1.3.1-1: Offset Calibration Registers

Register Name	*Address*	*Description*
OFFSET_CFG	0x30	Offset Configuration
OFFSET_VAL	0x31	Offset Value

Table 3.2.1.3.1-2: Register **OFFSET_CFG** (0x30) Offset Configuration

	MSB							*LSB*
Content	-	-	-	-	-	-	-	CALIB_MODE
Reset value	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W
Bit Description	**CALIB_MODE** : Operational amplifier mode selection: 1: calibration mode 0: current measurement mode							

Table 3.2.1.3.1-3: Register **OFFSET_VAL** (0x31) Offset Value

	MSB							*LSB*
Content	OFFS[7:0]							
Reset value	0b10100100							
Access	R/W							
Bit Description	**OFFS[7:0]** : Digital Offset Value							

3.2.1.3.2 Analogue Signal Measurement

Analog signals can be switched to AOUT pin via AMUX register. The observable voltage range at AOUT pin is limited from 0V to V_{VDD}.

Table 3.2.1.3.2-1: Register **AMUX** (0x17) Analog signal measurement selection

	MSB							*LSB*
Content	BUF_EN	-	-	S[4]	S[3:0]			
Reset value	1	0	0	0	0000			
Access	R/W	R	R	R/W	R/W			
Bit Description	**BUF_EN** : 1: Enable AOUT buffer 0: Bypass AOUT buffer **S[4]** : Reserved **S[3:0]** : Analogue signal selection. See 3.2.1.3.2-2 for coding.							

Table 3.2.1.3.2-2: Analog signals measurement code table

AMUX	*Analog signal to AOUT pin*
0, 1, 11-15	V_{GND}
2	V_{IP}
3	V_{TEMP}
4	$V_{VBATMEAS}$ / 12
5	V_{VG} / 5
6	Over current comparator threshold (IOCOMPTHR) [1)]
7	Low side power FET short circuit threshold (SCTH_LS) [1)]
8	V_{M1} / 12
9	V_{M2} / 12
10	V_{M3} / 12

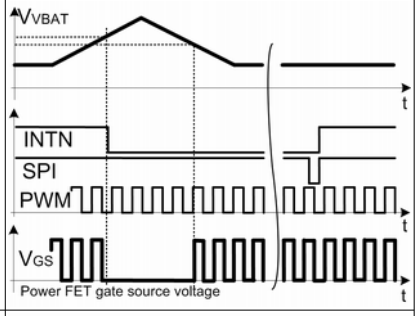
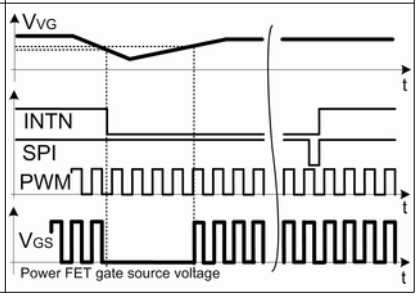
[1)] Do not switch AMUX position while motor is running. Monitoring functions can be triggered.

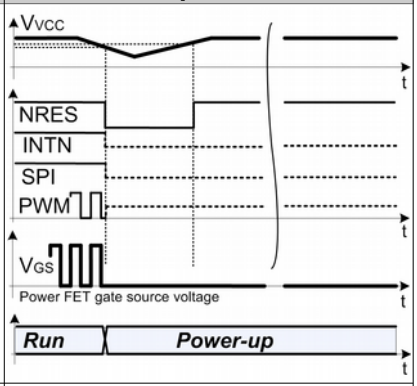
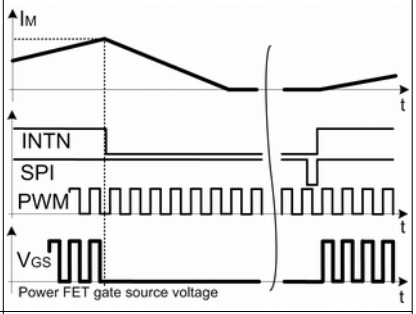
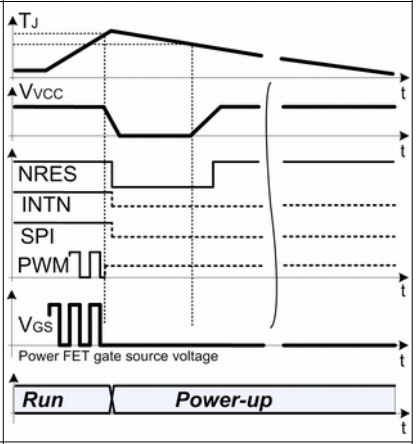
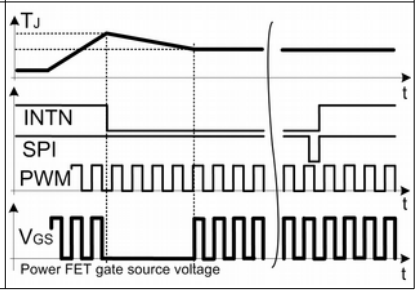
3.2.1.4 Monitoring and Safety

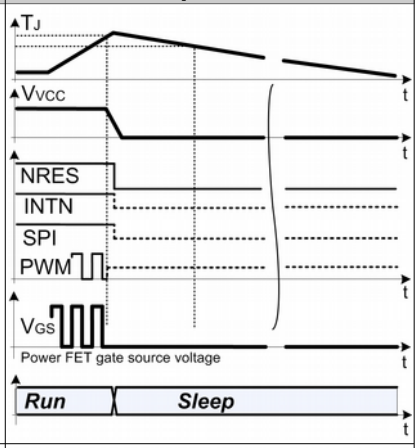
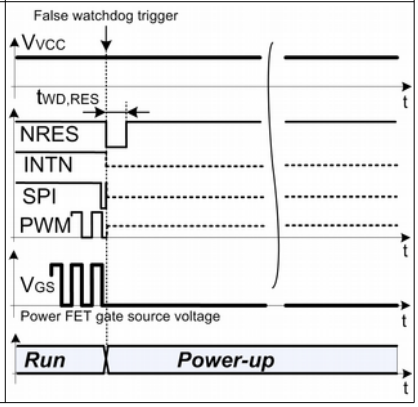
The IC provides several failure monitoring and protection functions:

- Battery over voltage at VBATMEAS pin
- VG power FET supply under voltage
- VCC microcontroller supply under voltage
- Motor over current
- IC over temperature
- Watchdog

Table 3.2.1.4-1: Monitoring functions

Function	*Failure condition*	*Failure actions*	*Reset condition*	*Depiction*
VBAT over voltage	$V_{VBATMEAS} > V_{VBAT,OV}$, SAFECTRL[3] = 1	Switch off all high side power FETs, allow switching of low side power FETs	$V_{VBATMEAS} < V_{VBAT,OV}$	
VG under voltage	$V_{VG} < V_{VG,UV}$, SAFECTRL[2] = 1	Switch off all power FETs	$V_{VG} > V_{VG,UV}$	

Function	Failure condition	Failure actions	Reset condition	Depiction
VCC under voltage	$V_{VCC} < V_{VCC,UV}$	Switch off all power FETs, reset of external microcontroller, reset of internal registers [1)]	$V_{VCC} > V_{VCC,UV}$, restart of the IC and external microcontroller	
Motor over current	$V_{IO} > V_{DAC,OC}$, SAFECTRL[4] = 1	Switch all power FETs off	$V_{IO} < V_{DAC,OC}$, cleared interrupt	
Over temperature, case VCC off	$T_J > T_{otemp}$, SAFECTRL[1:0] = 2	Switch off all power FETs, VCC and VG regulator, disable BUS, reset of external microcontroller, reset of internal registers [1)]	$T_J < T_{otemp}$, restart of the IC and external microcontroller	
Over temperature, case VCC on	$T_J > T_{otemp}$, SAFECTRL[1:0] = 0	Switch off all power FETs, disable BUS, **Safety warning:** The microcontroller has to switch into low power mode immediately, else the IC might be damaged.	$T_J < T_{otemp}$	

Function	Failure condition	Failure actions	Reset condition	Depiction
Over temperature, case sleep mode	$T_J > T_{otemp}$, SAFECTRL[1:0] = 1 or 3	Switch off all power FETs, VCC and VG regulator, disable BUS, reset of external microcontroller, reset of internal registers [1], go to sleep mode	$T_J < T_{otemp}$, wake-up	
Watchdog event	No watchdog trigger or false trigger, WDCTRL[1] = 1	Switch off all power FETs, reset of external microcontroller, reset of internal registers [1]	Restart of the IC and external microcontroller	

[1] except IRQSTATx, WU_SRC, RST_SRC, CHIPCTRL

The SAFECTRL safety functions register configures the failure reactions of the IC.

Table 3.2.1.4-2: Register **SAFECTRL** (0x05) Safety function configuration

	MSB							LSB
Content	-	-	VGUV6V	SOC	SVBATO	SVG	OTVC-COFF	OTSLEEP EN
Reset value	0	0	0	1	1	1	1	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**VGUV6V** : V_{VG} under voltage threshold selection: 1: $_{VBAT}$ > 6V, 0: V_{VBAT} > 7V **SOC** : Over current safety function enable **SVBATO** : VBAT over voltage safety function enable **SVG** : VG over and under voltage safety function enable **OTVCCOFF** : Disable VCC supply on over temperature. This bit has no effect when OTSLEEP is 1 (VCC is always off in sleep mode). **OTSLEEPEN** : IC shut down on over temperature enable							

3.2.1.4.1 Motor Over Current

The motor current measurement amplifier output at IO pin is monitored by the IC. If the voltage exceeds a programmable threshold, a motor over current is detected. The threshold value is calculated to:

$$V_{th,OC} = \frac{IOCOMPTHR}{64} \cdot V_{VDD}$$

The motor over current debounce time can be chosen in IOCOMPDEB register:

$$t_{deb,OC} = (4 * OCDEB + (0\_1)) \cdot \frac{1}{f_{CLK}}$$

An additional independent debouncing can be used for diagnosis purposes by the microcontroller:

$$t_{deb,OC,diag} = (OCDEB\_DIAG + (0\_1)) \cdot \frac{1}{f_{CLK}}$$

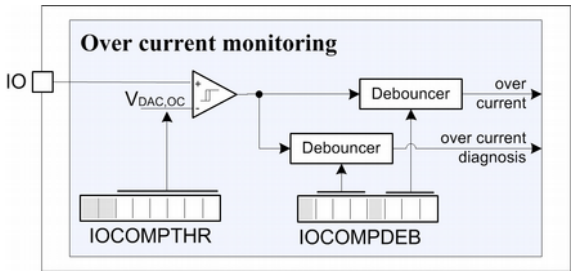


Figure 3.2.1.4.1-1: Motor over current circuitry

Table 3.2.1.4.1-1: IO comparator register

Register Name	Address	Description
IOCOMPTHR	0x16	Motor over current threshold
IOCOMPDEB	0x1D	Motor over current debounce time

Table 3.2.1.4.1-2: Register **IOCOMPTHR** (0x16) Motor over current threshold

	MSB							LSB
Content	-	-	D[5:0]					
Reset value	0	0	000000					
Access	R	R	R/W					
Bit Description	**D[5:0]** : Motor over current threshold value							

Table 3.2.1.4.1-3: Register **IOCOMPDEB** (0x1D) Motor over current debounce time

	MSB							LSB
Content	-	OCDEB_DIAG[2:0]			-	OCDEB[2:0]		
Reset value	0	0			0	0		
Access	R	R/W			R	R/W		
Bit Description	**OCDEB_DIAG[2:0]** : Motor over current diagnosis debounce time (failure reaction to be implemented in µC) **OCDEB[2:0]** : Motor over current debounce time							

3.2.1.4.2 Watchdog

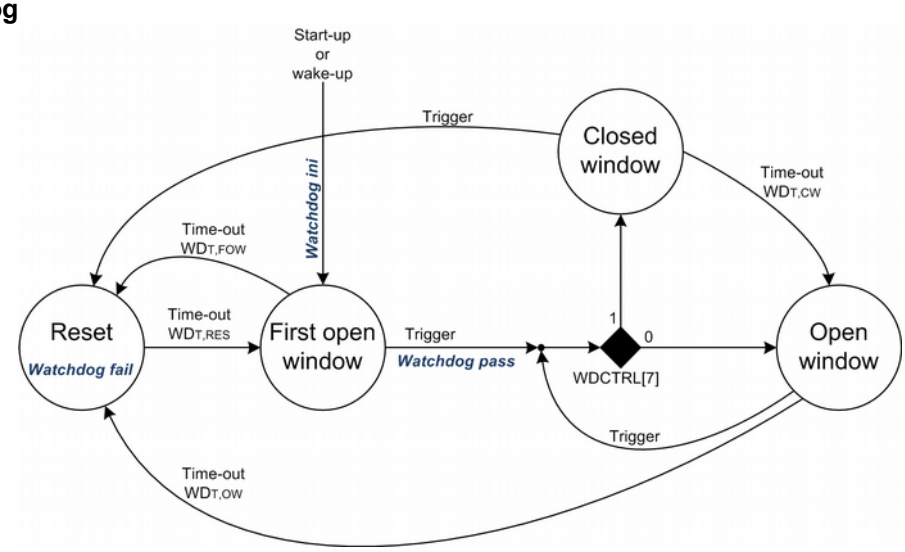


Figure 3.2.1.4.2-1: Watchdog state diagram

Table 3.2.1.4.2-1: Watchdog control registers

Register Name	*Address*	*Description*
WDCTRL	0x10	Watchdog control[1)]
WDTRIG	0x11	Watchdog trigger

[1)] For debugging purposes and software upload only, the watchdog can be stopped by setting T pin to VCC level.

Table 3.2.1.4.2-2: Register **WDCTRL** (0x10) Watchdog control[1)]

	MSB						*LSB*
Content	WDUSECW	-	WDTOW[1:0]	WDFOW	WDTEST	REG-WDEN	-
Reset value	1	0	11			1	0
Access	R/W	R	R/W	R	R	R/W	R
Bit Description	**WDUSECW** : Use closed window for register watchdog ($WD_{T,CW}$) **WDTOW[1:0]** : Register watchdog open window time ($W_{DT,OW}$) **WDFOW** : Register watchdog is in first open window **WDTEST** : 1: Register watchdog is stopped (T pin = 'H'), 0: Register watchdog is running **REGWDEN** : Enable register watchdog						

[1)] For debugging purposes and software upload only, the watchdog can be stopped by setting T pin to VCC level.

Table 3.2.1.4.2-3: Register **WDTRIG** (0x11) Watchdog trigger

	MSB							***LSB***
Content	-	-	-	-	-	-	-	TRIG
Reset value	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W
Bit Description	**TRIG** : Value has to be toggled on each write access in order to trigger the watchdog							

A watchdog can be used to monitor the external microcontroller.

The watchdog can be configured as standard or window watchdog in WDCTRL register within the first open window. It is triggered by SPI access to WDTRIG register.

Standard watchdog (WDUSECW = 0): The next trigger command has to be sent within $W_{DT,OW}$.

Window watchdog (WDUSECW = 1): The next trigger command has to be sent after $W_{DT,CW}$ and before $W_{DT,CW}$ + $WD_{T,OW}$.

If a watchdog reset occures 63 times after power-up successively, the device is sent to sleep mode autonomously.

3.2.1.4.3 Interrupt

The interrupt bits in IRQSTAT1 and IRQSTAT2 registers are set in case of the accordant failure event.

An interrupt can only be cleared, when the failure situation is over. The interrupt bit is cleared by writing 1 to the interrupt bit in IRQSTAT registers or after power-up (applying battery voltage). It is not cleared during sleep mode.

The interrupt mask bits in IRQMSK1 and IRQMSK2 registers mask the effect of the interrupt bits in IRQSTAT registers on the interrupt ins.

Interrupt output are INTN and SO pins (latter in case of CSB = H only).

Note: Even if the failure reaction is disabled in SAFECTRL register, the interrupt bits are set.

Note: After finishing initial configuration the VG under voltage interrupt is set (if C_{VG} is discharged), since the VG voltage regulator needs some time to power-up.

Table 3.2.1.4.3-1: Interrupt register

Register Name	***Address***	***Description***
IRQMSK1	0x12	Interrupt mask register 1
IRQSTAT1	0x13	Interrupt status register 1
IRQMSK2	0x14	Interrupt mask register 2
IRQSTAT2	0x15	Interrupt status register 2

Table 3.2.1.4.3-2: Register **IRQMSK1** (0x12) Interrupt mask register 1

	MSB							***LSB***
Content	EN_OT	EN_OC	EN_SC_HS3	EN_SC_HS2	EN_SC_HS1	EN_SC_LS3	EN_SC_LS2	EN_SC_LS1
Reset value	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**EN_OT** : Enable over temperatur interrupt **EN_OC** : Enable over current interrupt / IO comparator interrupt **EN_SC_HS3** : Enable HS 3 short circuit interrupt **EN_SC_HS2** : Enable HS 2 short circuit interrupt **EN_SC_HS1** : Enable HS 1 short circuit interrupt **EN_SC_LS3** : Enable LS 3 short circuit interrupt **EN_SC_LS2** : Enable LS 2 short circuit interrupt **EN_SC_LS1** : Enable LS 1 short circuit interrupt							

Table 3.2.1.4.3-3: Register **IRQSTAT1** (0x13) Interrupt status register 1

	MSB							***LSB***
Content	OT	OC	SC_HS3	SC_HS2	SC_HS1	SC_LS3	SC_LS2	SC_LS1
Reset value	0	0	0	0	0	0	0	0
Access	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Bit Description	**OT** : Over temperature interrupt flag (cleared by writing 1) **OC** : Over current interrupt flag / IO comparator interrupt flag (cleared by writing 1) **SC_HS3** : HS 3 short circuit interrupt flag (cleared by writing 1) **SC_HS2** : HS 2 short circuit interrupt flag (cleared by writing 1) **SC_HS1** : HS 1 short circuit interrupt flag (cleared by writing 1) **SC_LS3** : LS 3 short circuit interrupt flag (cleared by writing 1) **SC_LS2** : LS 2 short circuit interrupt flag (cleared by writing 1) **SC_LS1** : LS 1 short circuit interrupt flag (cleared by writing 1)							

Table 3.2.1.4.3-4: Register **IRQMSK2** (0x14) Interrupt mask register 2

	MSB							***LSB***
Content	EN_OC_DIAG	EN_TXDTO	EN_SPI_CS	-	EN_VBAT_OV	EN_VG_OV	EN_VG_UV	EN_VCC_UV
Reset value	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Bit Description	**EN_OC_DIAG** : Enable over current diagnosis interrupt **EN_TXDTO** : Enable TXD timeout failure interrupt **EN_SPI_CS** : Enable SPI check sum failure interrupt **EN_VBAT_OV** : Enable VBAT over voltage interrupt **EN_VG_OV** : Enable VG over voltage interrupt **EN_VG_UV** : Enable VG under voltage interrupt **EN_VCC_UV** : Enable VCC under voltage interrupt							

Table 3.2.1.4.3-5: Register **IRQSTAT2** (0x15) Interrupt status register 2

	MSB							***LSB***
Content	OC_DIAG	TXDTO	SPI_CS	-	VBAT_OV	VG_OV	VG_UV	VCC_UV
Reset value	0	0	0	0	0	0	0	0
Access	W1C	W1C	W1C	R	W1C	W1C	W1C	W1C
Bit Description	**OC_DIAG** : Over current diagnosis interrupt flag (cleared by writing 1) **TXDTO** : TXD timeout failure interrupt flag (cleared by writing 1) **SPI_CS** : SPI check sum failure interrupt flag (cleared by writing 1) **VBAT_OV** : VBAT over voltage interrupt flag (cleared by writing 1) **VG_OV** : VG over voltage interrupt flag (cleared by writing 1) **VG_UV** : VG under voltage interrupt flag (cleared by writing 1) **VCC_UV** : VCC under voltage interrupt flag (cleared by writing 1)							

3.2.1.4.4 Internal Digital Signal Monitoring

Internal digital signals, e. g. monitoring comparator outputs or digital input pin levels, can be monitored by the microcontroller. They can be switched to SO pin transparently via DMUX register or read via SPI in DMON1-2 register. DMUX register write access is locked, if EDMUX bit in SECURCTRL register is 0.

Table 3.2.1.4.4-1: Register for output of additional internal values

Register Name	***Address***	***Description***
DMUX	0x19	Digital value output SO selection
DMON1	0x1A	Digital value monitoring register 1
DMON2	0x1B	Digital value monitoring register 2

Table 3.2.1.4.4-2: Register **DMUX** (0x19) Digital value output SO selection

	MSB							***LSB***
Content	-	-	-	-	S[3:0]			
Reset value	0	0	0	0	0000			
Access	R	R	R	R	R/W			
Bit Description	**S[3:0]** : SO pin digital signal output selection. See 3.2.1.4.4-5 for coding.							

Table 3.2.1.4.4-3: Register **DMON1** (0x1A) Digital value monitoring register 1

	MSB							***LSB***
Content	OT	OC	SC_HS3	SC_HS2	SC_HS1	SC_LS3	SC_LS2	SC_LS1
Reset value								
Access	R	R	R	R	R	R	R	R
Bit Description	**OT** : Over temperature comparator (debounced) **OC** : Over current comparator (debounced) **SC_HS3** : High side 3 short circuit comparator (debounced) **SC_HS2** : High side 2 short circuit comparator (debounced) **SC_HS1** : High side 1 short circuit comparator (debounced) **SC_LS3** : Low side 3 short circuit comparator (debounced) **SC_LS2** : Low side 2 short circuit comparator (debounced) **SC_LS1** : Low side 1 short circuit comparator (debounced)							

Table 3.2.1.4.4-4: Register **DMON2** (0x1B) Digital value monitoring register 2

	MSB							***LSB***
Content	S	-	-	-	VBAT_OV	VG_OV	VG_UV	-
Reset value		0	0	0				0
Access	R	R	R	R	R	R	R	R
Bit Description	**S** : Pin S input comparator level (debounced) **VBAT_OV** : VBAT over voltage comparator level (debounced) **VG_OV** : VG over voltage comparator level (debounced) **VG_UV** : VG under voltage comparator level (debounced)							

Table 3.2.1.4.4-5: Digital signal output code table

DMUX	***Value at pin SO*** [1)]
0, 1, 6-8, 10-15	High impedance
2	S pin level
3	T pin level
4	Motor over current comparator level (debounced)
5	Over temperature comparator level (debounced)
9	VG under voltage comparator level (debounced)

[1)] Shared functionality with SO pin: If CSB = 'L', SO is used as SPI data output in any configuration.

3.2.1.5 Communication Interfaces

3.2.1.5.1 BUS interface (LIN or PWM)

The BUS interface is a bidirectional, single wire, high voltage, low active interface. It can be used as a transceiver for LIN standard or PWM control with error feedback.

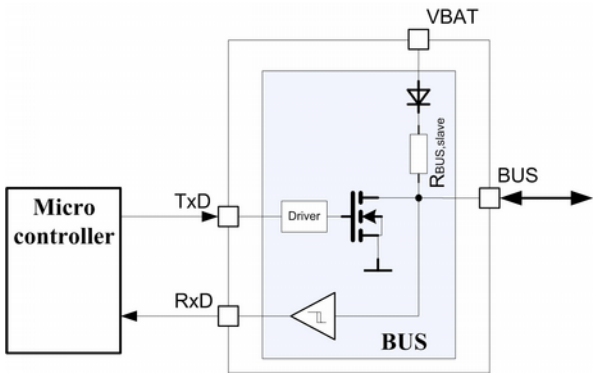


Figure 3.2.1.5.1-1: BUS block diagram

The BUS interface can be used as master or slave. The transmission data on TXD pin is converted into the BUS signal through a current limited, wave-shaping low side driver. The receiver converts the data stream from BUS to RXD pin.

In recessive state, BUS is pulled to VBAT by an internal pull-up resistor (30kΩ typically) and a diode in series, so no external pull-up components are required for slave applications. Master applications require an additional external pull-up resistor and a series diode.

The BUS transceiver can handle a voltage swing from 40V down to ground and survive -27V. The device also prevents from back current through the BUS pin to the supply pin in case of a ground shift or loss or supply voltage disconnection.

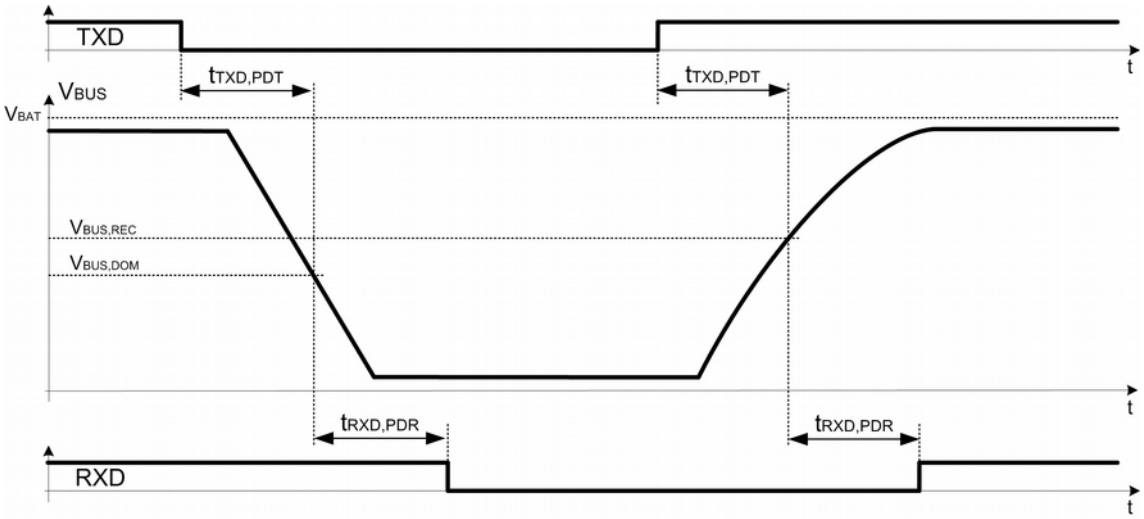


Figure 3.2.1.5.1-2: BUS transceiver transmit timing

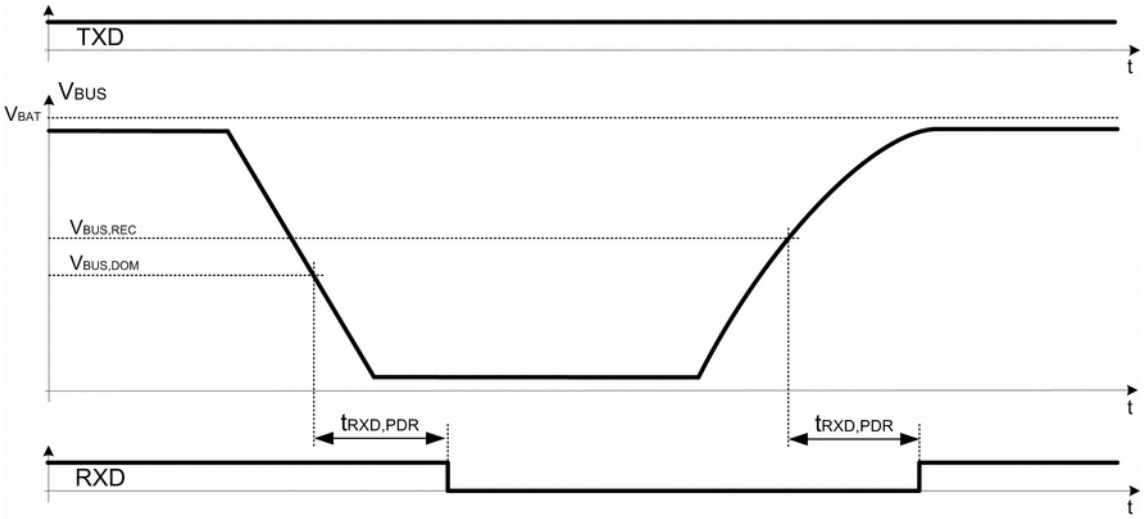


Figure 3.2.1.5.1-3: BUS transceiver receive timing

3.2.1.5.2 Flash Mode and TXD Time-Out

A flash mode allows an increasing of the transmit baud rate up to 115 kBd and the receive baud rate up to 250kBd. The flash mode can be activated by setting IOCFG[6] = 1.

In order to prevent the BUS from being permanent dominant in case of permanent low level at TXD pin, a TXD time-out timer switches BUS to recessive mode after approximately 8ms. The timer is triggered by a negative TXD edge and reset by a positive TXD edge. This function can be deactivated by writing IOCFG[7] = 0.

3.2.1.5.3 LIN Compatibility

The IC fulfils LIN standard 2.2A (ISO 9141). LIN conformity is given at battery voltage 7V to 18V.

LIN2.2 transceivers are backward compatible down to LIN1.3 at physical layer. But not the other way around. A node using LIN 2.2 physical layer can operate in a LIN 1.3 cluster. [1)]

1) Refer to LIN Specification Package Revision 2.2A December 31, 2010, Page 15

3.2.1.5.4 Communication Interfaces Configuration

BUS interface and interrupt output can be configured in IOCFG register.

Table 3.2.1.5.4-1: Register **IOCFG** (0x06) Interface configuration

	MSB							***LSB***
Content	TXDTO_EN	LIN_FLASH	-	-	-	-	-	-
Reset value	1	0	0	0	0	0	0	0
Access	R/W	R/W	R	R	R	R	R	R
Bit Description	**TXDTO_EN** : Enable TXD dominant clamping time-out **LIN_FLASH** : Enable LIN flash mode							

3.2.1.5.5 SPI

The SPI interface is used for data transfer between the microcontroller and the IC. The SPI clock is high active (CPOL = 0). The data is sampled with second clock edge (CPHA = 1).

The SPI interface supports two transfer modes:

- Two byte access (address + data) without checksum and
- Three byte access (address + data + checksum) with checksum.

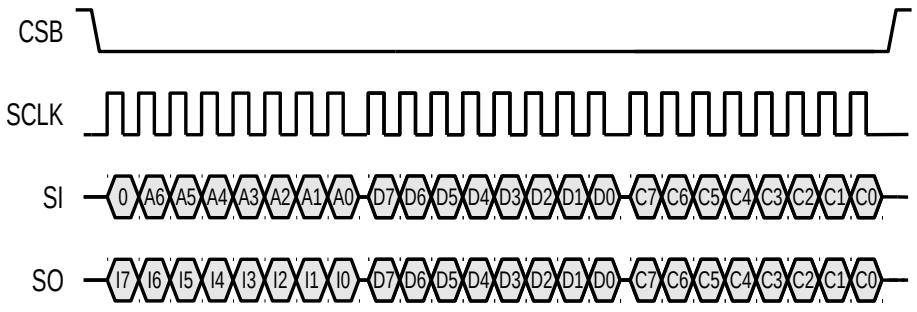


Figure 3.2.1.5.5-1: SPI write command with checksum

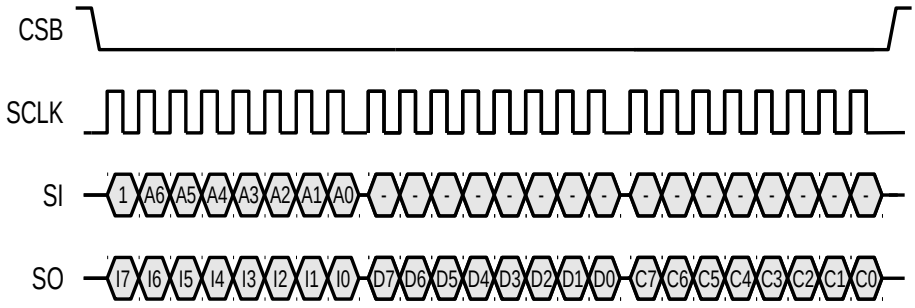


Figure 3.2.1.5.5-2: SPI read command with checksum

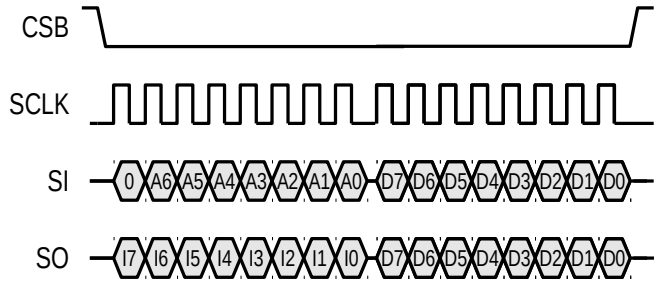


Figure 3.2.1.5.5-3: SPI write command without checksum

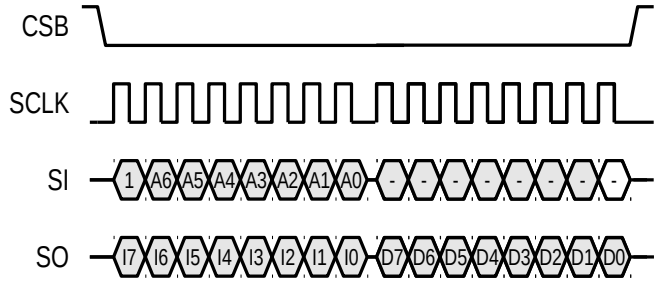


Figure 3.2.1.5.5-4: SPI read command without checksum

A[6:0]: register address
D[7:0]: register data
C[7:0]: checksum[7:0] = address[7:0] + data[7:0]
I[7:0]: interrupt registers (masked). (See table 3.2.1.5.5-1 for more information)

Checksum calculation:
In case of three byte mode the checksum is calculated as addition of first and second byte (including read/write byte).

Third byte of input (SI) checksum = first SI byte (address) + second SI byte (data).
Third byte of output (SO) checksum = first SI byte (address) + first SO byte (status) + second SO byte (data).

Table 3.2.1.5.5-1: SPI interrupt status byte

I7	*I6*	*I5*	*I4*	*I3*	*I2*	*I1*	*I0*
OC_DIAG	OT	SPI_CS	SC*	VBAT_OV	VG_OV	VG_UV	VCC_UV

* **SC** = IRQ of all short circuit interrupts.

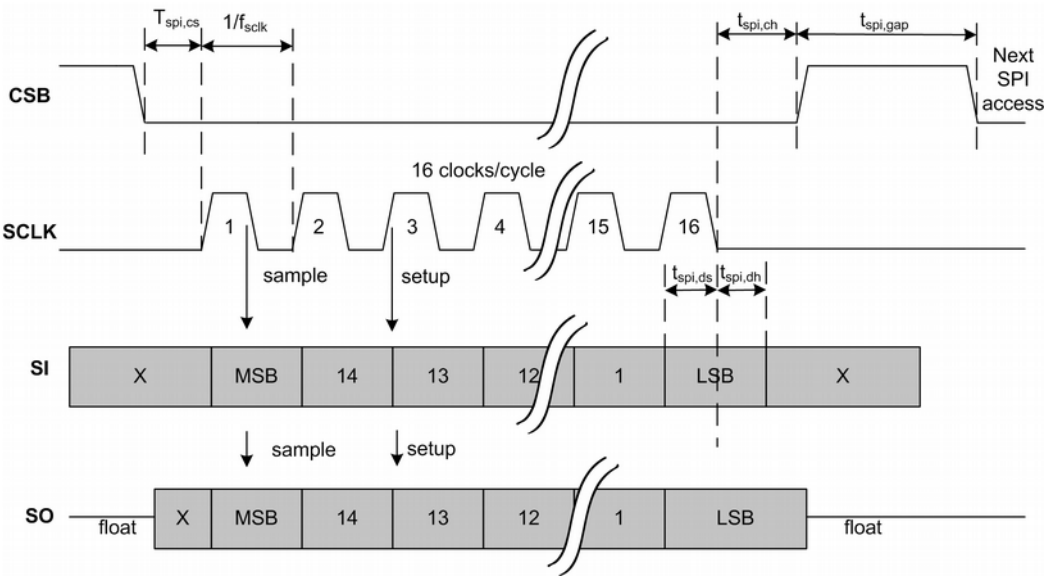


Figure 3.2.1.5.5-5: SPI timing diagram

3.3 Microcontroller

3.3.1 Design Description [E523.98A]

3.3.1.1 Analog Part

3.3.1.1.1 Core Supply Regulator

The regulator generates the digital core voltage VDDC from VDDIO. It receives an internal 1.8V reference voltage derived from the bandgap voltage and regulates with it the digital core voltage VDDC.
An external Buffer Capacitance CDDC can connected to the output.

3.3.1.1.2 Oscillators and Reset

Oscillators:
- System Clock RC Oscillator
- Watchdog Clock RC Oscillator

System Reset Sources:
- Power ON reset
- Power watches for brownout detection
- Reset inputs (device pin)
- Several digital core exceptions
- For details see SYS_STATE module RESET_STATUS and RESET_ENABLE registers.

3.3.1.1.2.1 Power On Reset

The Power-On-Reset is connected to the VDDA Power Supply.
Its main purpose is to reset the level shifter from the VDDC to VDDIO voltage domain.

The Power-On-Reset will assert a system reset on system start-up

3.3.1.1.2.2 Brownout Detection
There are two Brown-Out watches:
- one for the VDDIO voltage
- one for the VDDC voltage

Since VDDIO is bonded onto the same pin as VDDA in current targeted applications the brown-out effectively monitors VDDA and VDDIO voltage for outages.
In case the device will be used in an application where the two pins will be supplied by independent power supplies no reset may be generated if the VDDA supply fails!

- VDDC brownout will cause a System Reset when VDDC falls below brownout (VDDC_OK_FE) level
- VDDIO brownout will cause a System Reset when VDDIO falls below brownout (VDDIO_OK_FE) level

3.3.1.1.2.3 System Clock RC Oscillator
This oscillator clocks the digital system.

3.3.1.1.2.4 Watchdog Clock RC Oscillator
The oscillator is used to clock a part of the digital watchdog module.

3.3.1.1.2.5 NRST debouncer
NRST low active reset input signal debouncer - prevents system from reset by short spikes.

3.3.1.1.3 SAR-ADC
The pin AIN is the input to a single ended SAR ADC with a resolution of 12 Bits and at least an effective number of 10 Bits. The ADC has a single high reference voltages of 2.5V derived from the bandgap voltage reference. The low reference voltage is fixed to VSSA.

3.3.1.1.4 ADC Multiplexer
The ADC multiplexer is switching one of 25 input signals to the ADC amplifier. Channel 0-23 for IO port PA0-7, PB0-7, PC0-7 and channel 24 for temperature sensor voltage.

3.3.1.1.5 IO Port Characteristics
These are modified digital standard pads.
They have a configurable driver strength of 4 or 8mA (See SYS_STATE Module Registers).

Pad pull configuration:
- IO port pads do not have a pull resistance.
- TMODE have a pull-down resistance
- NRST and NRSTI will have a pull-up resistance

An analog signal by-pass path is added to the standard pads. So each digital pad can be used as analog input (ADC). A double-switch with in between a switch-to-ground-connection is added to the standard pad.

3.3.1.2 Digital Part
3.3.1.2.1 Base Addresses
Table 3.3.1.2.1-1: Address Table

base address	*size*	*module name*	*instance name*	*description*
0x8000	0x8000	FLASH	FLASH	FLASH Memory
0x2000	0x6000	SYS_ROM	SYS_ROM	System ROM Memory

base address	size	module name	instance name	description
0x0F00	0x1100	SRAM	SRAM	SRAM Memory
0x0600	0x80	PWMN	PWMN	Motor PWM Module
0x0580	0x80	PRE_PWM	PRE_PWM	Pre-PWM Module
0x0500	0x80	ADC_CTRL	ADC_CTRL	ADC Control Module
0x04C0	0x40	OSC_CTRL	OSC_CTRL	Oscillator Control Module
0x0480	0x40	CCTIMER	CCTIMER_3	Capture Compare Timer Module Instance 3
0x0440	0x40	CCTIMER	CCTIMER_2	Capture Compare Timer Module Instance 2
0x0400	0x40	CCTIMER	CCTIMER_1	Capture Compare Timer Module Instance 1
0x03C0	0x40	CCTIMER	CCTIMER_0	Capture Compare Timer Module Instance 0
0x0380	0x40	GPIO	GPIO_C	GPIO Module Port C Instance
0x0340	0x40	GPIO	GPIO_B	GPIO Module Port B Instance
0x0300	0x40	GPIO	GPIO_A	GPIO Module Port A Instance
0x02C0	0x40	LINSCI	LINSCI	LIN SCI Module
0x0280	0x40	SPI	SPI_1	SPI Module 1
0x0240	0x40	SPI	SPI_0	SPI Module 0
0x0200	0x40	IOMUX_CTRL	IOMUX_CTRL	IO Multiplexer Config Module
0x01C0	0x40	FLASH_CTRL	FLASH_CTRL	FLASH Control Module
0x0180	0x40	SYS_STATE	SYS_STATE	System State Module
0x0140	0x40	DIVIDER	DIVIDER	Divider Module
0x0100	0x40	H430_MUL	H430_MUL	H430 Multiplier Module
0x00C0	0x40	MEM_PROT	MEM_PROT	Memory Protection Module
0x0080	0x40	WDOG	WDOG	Watchdog Module
0x0040	0x40	VIC	VIC	Vector Interrupt Controller Module
0x0000	0x40	ROM	ROM	Startup ROM

3.3.1.2.2 Memory IP's

3.3.1.2.2.1 FLASH

The micro controller system includes one instance of a FLASH IP which are mapped into the address space as defined by the above Memory Map Table.
This FLASH IP block consists of two logical blocks: a large one which is called MAIN block and a small one which is called INFO block. MAIN and INFO block cannot be accessed at the same time. A FLASH mode change is required to do this. The FLASH instance is controlled by a FLASH_CTRL module.

- FLASH MAIN area size: 32K byte
  - the INFO area consists of 2 pages
  - the upper page contains the INFO boot code which is described in the System Start-up chapter
  - the lower page can be used for other purpose
- FLASH area size: 32Kbyte
- FLASH includes a 6 bit ECC per 16 bit data -> Hamming distance 4
- SEC-DED logic (single error correction - double error detection)
- FLASH IP geometry:
  - 32K byte = 8K x 32 (44) bit
  - MAIN block: 64 pages
  - INFO block: 2 pages

- 1 page = 128 x 32 (44) bit = 4 rows
- 1 row = 32 x 32 (44) bit

Info: Erase and write of FLASH is forbidden when temperature is over 125°C!

3.3.1.2.2.2 System ROM (SYS_ROM)
- size: 24Kbyte
- read only
- contains standard LIN routines which can be used by the executed user program and a boot loader program (see "boot code flow chart")

3.3.1.2.2.3 SRAM
- size: 4,25Kbyte
- byte write support
- per byte parity protection

3.3.1.2.2.4 Memory Access Protection
CPU and memory access is protected by a logic which has to be set up via JTAG with the correct 64 bit signature key to allow full system access.
The 64 bit signature key is stored by the costumer in the FLASH-MAIN-Area at addresses 0xFFE8 - 0xFFEF.
The default key is 0xFFFF_FFFF_FFFF_FFFF.

3.3.1.2.3 System Start-up
The digital system start up is done as follows:

- The CPU executes the Startup ROM code which checks the FLASH INFO memory for a valid boot vector (which points into FLASH INFO memory area).
  - If a valid FLASH INFO memory boot vector exists:
    - The CPU executes the FLASH INFO memory start up code which usually is used to initialize the micro controller analog part calibration registers as well as the analog IC calibration data. The calibration data may be included in the FLASH INFO memory code.
    - The FLASH INFO memory start up code may check the System ROM for a boot loader and execute it, depending on the system configuration.
      - Please see the system boot loader concept application note for details.
      - The following sequence also depends on the system boot loader concept.
      - One possible behavior may be:
      - The CPU returns to FLASH INFO memory boot code.
    - The CPU returns to ROM start up code.
  - The CPU switches to FLASH MAIN memory area access.
  - The CPU fetches the user program reset vector which is located at address 0xFFFE in the FLASH MAIN memory which also enables the JTAG interface for CPU debugging.
- The CPU starts executing the user program.

Note: the FLASH INFO memory start up code area is only visible during ROM start up code execution and will not be accessible during user program execution.

Boot code flow chart:

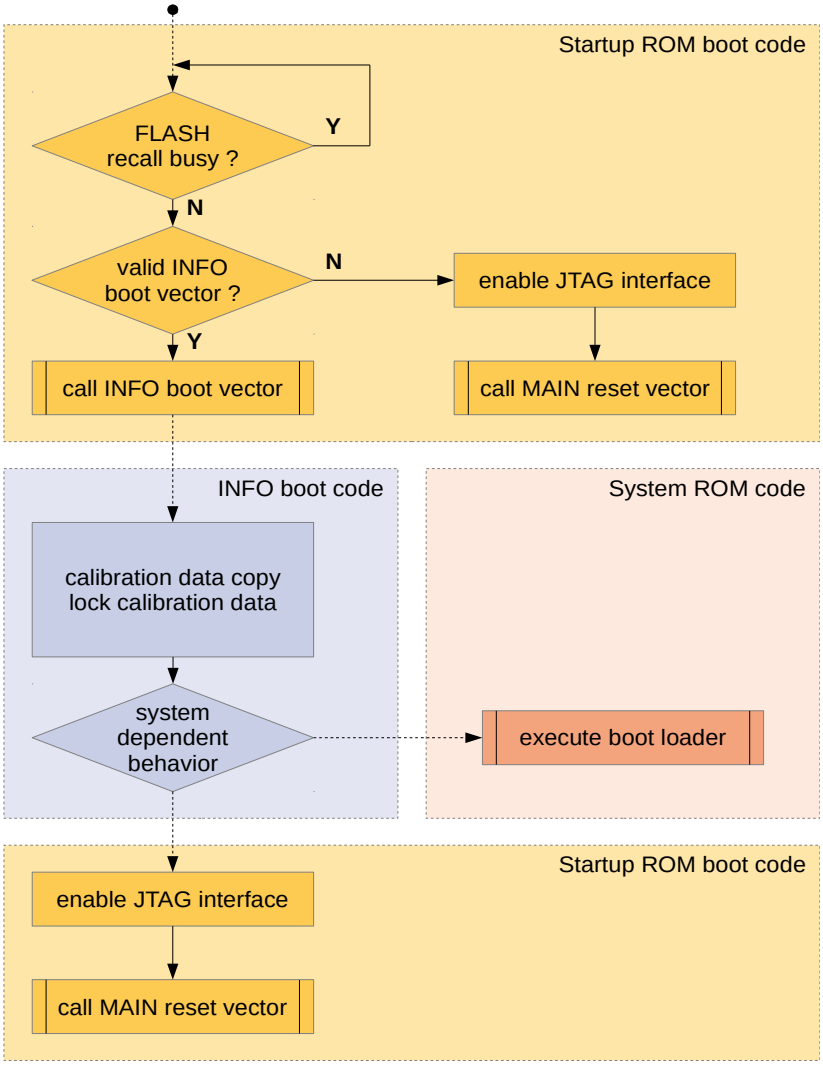


Figure 3.3.1.2.3-1: boot code flow chart

3.3.1.2.4 CPU - H430

Features

- 16 bit CPU
- MSP430 binary code compatible
- Harvard architecture with AHBL data and instruction bus interfaces
- RISC architecture with 27 instructions and 7 addressing modes
- Orthogonal architecture: every instruction usable with every addressing mode
- Full register access including program counter, status registers, and stack pointer
- 16 x 16-bit register
- 64 KByte linear address space
- 16-bit native data bus width
- Constant generator provides six most used immediate values and reduces code size
- Direct memory-to-memory transfers without intermediate register holding

- Word and byte addressing and instruction formats
- IAR development IDE compatible JTAG debug interface
- Several C compilers are available

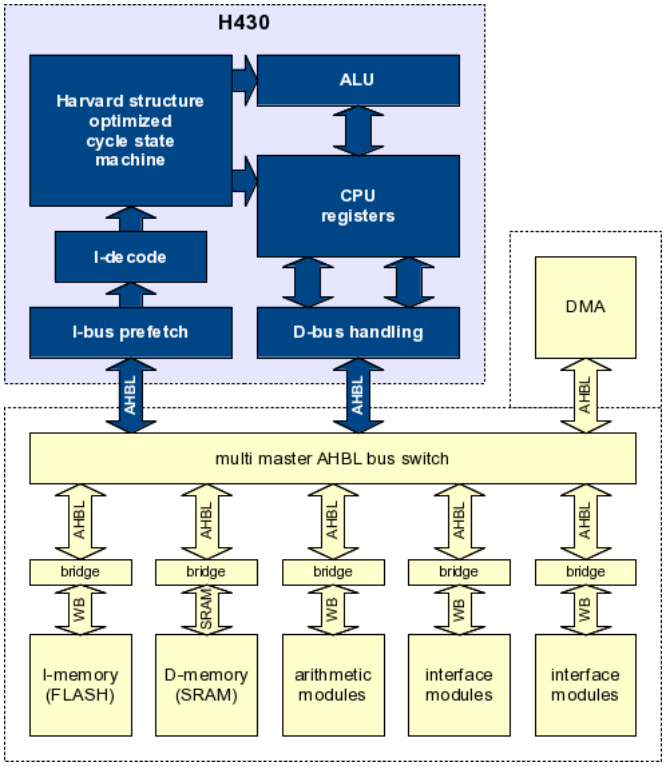


Figure 3.3.1.2.4-1: H430 Environment Example

Interrupts

The embedded H430 IP core does not contain a primary interrupt controller. It has only a IRQ request signal and an address, pointing to a vector table in memory, which contains addresses of the interrupt handlers.
Therefore the H430 IP does not support a fixed number of interrupts. Any number fitting reasonable in the 64k memory range is supported.
All interrupts can be enabled or disabled with the GIE bit in the status register.

Handling an interrupt (other than RESET) consists of:
- Push PC on stack.
- Push SR on stack.
- Choose the highest priority interrupt to service.
- If there are multiple possible sources, leave them for software to poll.
- Clear the SR, which disables interrupts and power-saving.
- Fetch the interrupt vector into the PC
- Start executing the interrupt handler

A reset is similar, but doesn't save any state.

You can nest interrupt handlers by disabling the current source and setting the GIE bit back to 1.

Byte and Word Issues

The H430 is byte-addressed, and Little-Endian. Word operands must be located at even addresses.
Most instructions have a byte/word bit, which selects the operand size. Appending ".B" to an instruction makes it a byte operation. Appending ".W" to an instruction, to make it a word operation, is also legal. However, since it is also the default behavior, if you add nothing, it is generally omitted.
A byte instruction with a register destination clears the high 8 bits of the register to 0. Thus, the following would clear the top byte of the register, leaving the lower byte unchanged:

MOV.B Rn,Rn

Mostly the on-chip peripherals supports only one bus size, e.g. the data width of the processor. These peripherals must be accesses only with the supported access mode and with correct alignment. Any other access may produce an undefined behavior.
When performing a word access, address bit 0 is undefined and has to be ignored.

CPU States

The CPU supports the following states:

Table 3.3.1.2.4-1: CPU States

state	*description*
RUN	• normal operation of the CPU • the CPU accesses program storage (e.g. Flash) and RAM • the CPU returns to RUN state on any interrupt
STANDBY	• the CPU is halted • the STANDBY state is entered when setting standby flag (CPUOFF) in status register • the CPU does not access program storage or RAM • the CPU returns to RUN state on any interrupt

CPU Standby Entry

After setting the standby bit in the CPU status register the following instruction will be executed, then standby mode will be entered. A good idea is to use the following sequence to ensure a later wake up.

BIS #0x18, SR ; sets standby flag and enables interrupts for wake up
NOP ; needed for correct standby entry behavior

CPU Standby Exit

- an interrupt will force the CPU to exit the standby mode. The CPU will enter the interrupt service routine directly.
- after the interrupt routine has been finished the CPU will NOT return to previous standby mode.
- a system reset (e.g. by the watchdog) will restart the device and therefore exit the standby mode.

3.3.1.2.4.1 CPU Registers
The processor has 16 16-bit registers, although only 12 of them are truly general purpose. The first four have dedicated uses:

3.3.1.2.4.1.1 Program Counter (PC)

The 16-bit Program Counter (PC/R0) points to the next instruction to be executed. Each instruction uses an even number of bytes (two, four, or six), and the PC is incremented accordingly. Instruction accesses in the 64-KB address space are performed on word boundaries, and the PC is aligned to even addresses. The PC can be addressed with all instructions and addressing modes.

3.3.1.2.4.1.2 Stack Pointer (SP)

The Stack Pointer (SP/R1) is used by the CPU to store the return addresses of subroutine calls and interrupts. It uses a pre-decrement, post-increment scheme. In addition, the SP can be used by software with all instructions and addressing modes. The SP is initialized into RAM by the user, and is aligned to even addresses.

3.3.1.2.4.1.3 Status Register (SR)

The Status Register (SR/R2), used as a source or destination register, can be used in the register mode only addressed with word instructions. The remaining combinations of addressing modes are used to support the constant generator.

Table 3.3.1.2.4.1.3-1: Register **Status Register** (SR/R2)

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8	-	-	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**8** : V **5** : OSC OFF **4** : CPU OFF **3** : GIE **2** : N **1** : Z **0** : C															

V: Overflow bit
This bit is set when the result of an arithmetic operation overflows the signed-variable range.

OSCOFF: Stop flag
OSCOFF (oscillator off), and CPUOFF are used to enter low-power states. OSCOFF may not be evaluated by the system if other clock controllers are implemented

CPUOFF: Standby flag
See "CPU States" for details

GIE: Global Interrupt Enable
GIE is the global interrupt enable. Turning off this bit masks interrupts. (NOTE: it may be delayed by 1 cycle, so an interrupt may be taken after the instruction after GIE is cleared. Add a NOP or clear GIE one instruction earlier than your "critical section".)

N: Negative bit
This bit is set when the result of a byte or word operation is negative and cleared when the result is not negative.
Word operation: N is set to the value of bit 15 of the result
Byte operation: N is set to the value of bit 7 of the result

Z: Zero bit
This bit is set when the result of a byte or word operation is 0 and cleared when the result is not 0.

C: Carry bit
This bit is set when the result of a byte or word operation produced a carry and cleared when no carry occurred.

3.3.1.2.4.1.4 Constant Generation Registers (CG1 / CG2)

Six commonly-used constants are generated with the constant generator registers R2 and R3, without requiring an additional 16-bit word of program code. This is one of the important features of the H430 instruction set, allowing it to achieve a high level of code density, and a flexible instruction set.
These constant registers can provide the numbers -1, 1, 2, 4 or 8. So, for example, the "CLR x" is actually emulated by the instruction "MOV #0,x". The constant "0" is taken from the constant register r3. The assembler understands both "CLR x" and "MOV #0,x", and produces the same code for either.
The constants are selected with the source-register addressing modes (As):

Table 3.3.1.2.4.1.4-1: Register Table

Register	*As*	*Value*	*Remarks*
R2	00	-	register mode (access R2)
R2	01	(0)	used for absolute address mode
R2	10	0x0004	constant +4
R2	11	0x0008	constant +8
R3	00	0x0000	constant 0
R3	01	0x0001	constant +1
R3	10	0x0002	constant +2
R3	11	0xFFFF	constant -1

The constant generator advantages are:

- No special instructions required
- No additional code word for the six constants
- No code memory access required to retrieve the constant

The assembler uses the constant generator automatically if one of the six constants is used as an immediate source operand. Registers R2 and R3, used in the constant mode, cannot be addressed explicitly; they act as source-only registers.

3.3.1.2.4.1.5 General Purpose Registers (R4 - R15)

The twelve registers, R4-R15, are general-purpose registers. All of these registers can be used as data registers or address pointers and can be used with byte or word instructions.

3.3.1.2.4.2 Addressing Modes

The available H430 instruction addressing modes have at most two operands, a source and a destination.
All instructions are 16 bits long, followed by at most two optional offsets words, one for each of the source and the destination.

As Modes

The source operand is specified with 2 addressing mode bits (As):

Table 3.3.1.2.4.2-1: As Modes

As	mnemonic	remarks
00	Rn	Register direct
01	X(Rn)	Register indexed
10	@Rn	Register indirect
11	@Rn+	Register indirect with post-increment

Ad Modes

The destination operand is specified with 1 addressing mode bit (Ad):

Table 3.3.1.2.4.2-2: Ad Modes

Ad	mnemonic	remarks
0	Rm	Register direct
1	Y(Rm)	Register indexed

The only addressing mode that uses an extension word is the indexed mode.
The destination operand in a two-operand instruction has only one addressing mode bit, which selects either register direct or indexed. Register indirect can obviously be faked up with a zero index.
When r0 (the program counter) is used as a base address, indexed mode provides PC-relative addressing. This is, in fact, the usual way that the H430 assembler accesses operands when a label is referred to.
@r0 just specifies the following instruction word, but @r0+ specifies that word and skips over it. In other word, an immediate constant! You can just write #1234 and the assembler will specify the addressing mode properly.
r1, the stack pointer, can be used with any addressing mode, but @r1+ always increments by 2 bytes, even on a byte access.

Table 3.3.1.2.4.2-3: Addressing Modes Table

As/Ad	Addressing Mode	Syntax	Description
00/0	Register mode	Rn	Register contents are operand
01/1	Indexed mode	X(Rn)	(Rn + X) point to the operand. X is stored in the next word.
01/1	Symbolic mode	ADDR	(Rn + X) point to the operand. X is stored in the next word. Indexed mode X(PC) is used.
01/1	Absolute mode	&ADDR	(Rn + X) point to the operand. X is stored in the next word. Indexed mode X(0) is used.
10/-	Indirect Register mode	@Rn	Rn is used as a pointer to the

As/Ad	Addressing Mode	Syntax	Description
11/-	Indirect auto increment	@Rn+	Rn is used as a pointer to the operand. Rn is incremented afterwards by 1 for .B instructions and by 2 for .W instructions
11/-	Immediate mode	#N	The word following the instruction contains the immediate constant N. Indirect auto-increment mode @PC+ is used.

Register Direct

Table 3.3.1.2.4.2-4: Register Direct

Assembler Code	MOV R10,R11
Length	One or two words
Operation	Move the content of R10 to R11. R10 is not affected.
Comment	Valid for source and destination
Note	The data in the register can be accessed using word or byte instructions. If byte instructions are used, the high byte is always 0 in the result. The status bits are handled according to the result of the byte instruction.

Register Indexed

Table 3.3.1.2.4.2-5: Register Indexed

Assembler Code	MOV 2(R5),6(R6)
Length	Two or three words
Operation	Move the contents of the source address (contents of R5 + 2) to the destination address (contents of R6 + 6). The source and destination registers (R5 and R6) are not affected. In indexed mode, the program counter is incremented automatically so that program execution continues with the next instruction.
Comment	Valid for source and destination

Register Indirect

Table 3.3.1.2.4.2-6: Register Indirect

Assembler Code	MOV @R10,0(R11)
Length	One or two words
Operation	Move the contents of the source address (contents of R10) to the destination address (contents of R11). The registers are not modified.
Comment	Valid only for source operand. The substitute for destination operand is 0(Rd).

Register Indirect with post increment

Table 3.3.1.2.4.2-7: Register Indirect with post-increment

Assembler Code	MOV @R10+,0(R11)
Length	One or two words
Operation	Move the contents of the source address (contents of R10) to the destination address (contents of R11). Register R10 is incremented by 1 for a byte operation, or 2 for a word operation after the fetch; it points to the next address without any overhead. This is useful for table processing.
Comment	Valid only for source operand. The substitute for destination operand is 0(Rd) plus second instruction INCD Rd.

3.3.1.2.4.3 Instruction Set

The complete H430 instruction set consists of 27 core instructions and 24 emulated instructions. The core instructions are instructions that have unique op-codes decoded by the CPU. The emulated instructions are instructions that make code easier to write and read, but do not have op-codes themselves, instead they are replaced automatically by the assembler with an equivalent core instruction. There is no code or performance penalty for using emulated instruction.
All instructions are 16 bits long, and there are only three instruction formats:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dual operand	operation				S-reg				Ad	B	As		D-reg			
single-operand	0	0	0	1	0	0	operation				As		D-reg			
jump	0	0	1	condition			PC offset (10 bit)									

Figure 3.3.1.2.4.3-1: Instruction Coding

All single-operand and dual-operand instructions can be byte or word instructions by using .B or .W extensions. Byte instructions are used to access byte data or byte peripherals. Word instructions are used to access word data or word peripherals. If no extension is used, the instruction is a word instruction.
The source and destination of an instruction are defined by the following fields:

Table 3.3.1.2.4.3-1: Source and destination of an instruction

Abbr.	*Description*
src	The source operand defined by As and S-reg
dst	The destination operand defined by Ad and D-reg
As	The addressing bits responsible for the addressing mode used for the source (src)
S-reg	The working register used for the source (src)
Ad	The addressing bits responsible for the addressing mode used for the destination (dst)
D-reg	The working register used for the destination (dst)
B/W	Byte or word operation: 0: word operation 1: byte operation

Dual Operand Instructions

These basically perform dst = src op dst operations. However, MOV doesn't fetch the destination, and CMP and BIT do not write to the destination. All are valid in their 8 and 16 bit forms.

+ The status bit is affected
- The status bit is not affected
0 The status bit is cleared
1 The status bit is set

Table 3.3.1.2.4.3-2: Dual Operand Instructions

Opcode	*Mnemonic*	*S-Reg, D-Reg*	*Operation*	*V*	*N*	*Z*	*C*	*Remark*
0100	MOV(.B)	src, dst	dst = src	-	-	-	-	The status flags are NOT set.
0101	ADD(.B)	src, dst	dst += src	+	+	+	+	
0110	ADDC(.B)	src, dst	dst += src + C	+	+	+	+	
1000	SUB(.B)	src, dst	dst += ~src + 1	+	+	+	+	
0111	SUBC(.B)	src, dst	dst += ~src + C	+	+	+	+	
1001	CMP(.B)	src, dst	dst - src	+	+	+	+	Sets status only; the destination is not written.
1010	DADD(.B)	src, dst	dst += src + C, BCD	0	+	+	+	
1011	BIT(.B)	src, dst	dst & src	0	+	+	+	Sets status only; the destination is not written.
1100	BIC(.B)	src, dst	dst &= ~src	-	-	-	-	The status flags are NOT set.
1101	BIS(.B)	src, dst	dst |= src	-	-	-	-	The status flags are NOT set.
1110	XOR(.B)	src, dst	dst ^= src	+	+	+	+	
1111	AND(.B)	src, dst	dst &= src	0	+	+	+	

Single Operand Instructions

The status flags are set by RRA, RRC, SXT, and RETI.
The status flags are NOT set by PUSH, SWPB, and CALL.

+ The status bit is affected
- The status bit is not affected
0 The status bit is cleared
1 The status bit is set

Table 3.3.1.2.4.3-3: Single Operand Instructions

Opcode	Mnemonic	S-Reg, D-Reg	Operation	V	N	Z	C	Remark
000	RRC(.B)	dst	C -> MSB -> ... -> LSB -> C	0	+	+	+	9-bit rotate right through carry. Clear the carry bit beforehand to do a logical right shift.
010	RRA(.B)	dst	MSB -> MSB ->... LSB -> C	0	+	+	+	Badly named, this is an 8-bit arithmetic right shift.
100	PUSH(.B)	src	SP-2 -> SP src -> @SP	-	-	-	-	Push operand on stack. Push byte decrements SP by 2.
001	SWPB	dst	swap bytes	-	-	-	-	The destination operand high and low bytes are exchanged. This has no byte form.
101	CALL	src	SP-2 -> SP PC+2 -> @SP src -> PC	-	-	-	-	Fetch operand, push PC, then assign operand value to PC. Note: the immediate form is the most commonly used. There is no easy way to perform a PC-relative call; the PC-relative addressing mode fetches a word and uses it as an absolute address. This has no byte form.

Opcode	Mnemonic	S-Reg, D-Reg	Operation	V	N	Z	C	Remark
110	RETI		TOS -> SR SP+2 -> SP TOS -> PC SP+2 -> SP	+	+	+	+	Pop SP, then pop PC. Note: The CPUOFF flag will not be stored to stack on interrupt entry, so the CPU will NOT return to low-power mode it was previously in.
011	SXT	dst	Bit 7 -> Bit 8.....Bit 15	0	+	+	+	Sign extend 8 bits to 16. No byte form.

Emulated Instructions

There are a number of zero- and one-operand pseudo-operations that can be built from these two-operand forms. These are usually referred to as "emulated" instructions:

Table 3.3.1.2.4.3-4: Emulated Instructions

Instruction	Emulation	Remark
NOP	MOV r3,r3	Any register from r3 to r15 would do the same thing. Note: that other forms of a NOP instruction can be constructed as emulated instructions, which take different numbers of cycles to execute. These can sometimes be useful in constructing accurate timing patterns in software.
POP dst	MOV @SP+,dst	
BR dst	MOV dst,PC	Branch and return can be done by moving to PC (r0)
RET	MOV @SP+,PC	Branch and return can be done by moving to PC (r0)
CLRC	BIC #1,SR	The constants were chosen to make status register (r2) twiddling efficient
SETC	BIS #1,SR	The constants were chosen to make status register (r2) twiddling efficient
CLRZ	BIC #2,SR	The constants were chosen to make status register (r2) twiddling efficient
SETZ	BIS #2,SR	The constants were chosen to make status register (r2) twiddling efficient
CLRN	BIC #4,SR	The constants were chosen to make status register (r2) twiddling efficient
SETN	BIS #4,SR	The constants were chosen to make status register (r2) twiddling efficient
DINT	BIC #8,SR	The constants were chosen to make status register (r2) twiddling efficient
EINT	BIC #8,SR	The constants were chosen to make status register (r2) twiddling efficient
RLA(.B) dst	ADD(.B) dst,dst	Shift and rotate left is done with add
RLC(.B) dst	ADDC(.B) dst,dst	Shift and rotate left is done with add
INV(.B) dst	XOR(.B) #-1,dst	Some common one-operand instructions
CLR(.B) dst	MOV(.B) #0,dst	Some common one-operand instructions
TST(.B) dst	CMP(.B) #0,dst	Some common one-operand instructions

Instruction	Emulation	Remark
DEC(.B) dst	SUB(.B) #1,dst	Increment and decrement (by one or two)
DECD(.B) dst	SUB(.B) #2,dst	Increment and decrement (by one or two)
INC(.B) dst	ADD(.B) #1,dst	Increment and decrement (by one or two)
INCD(.B) dst	ADD(.B) #2,dst	Increment and decrement (by one or two)
ADC(.B) dst	ADDC(.B) #0,dst	Increment and decrement carry.
DADC(.B) dst	DADD(.B) #0,dst	Increment and decrement carry.
SBC(.B) dst	SUBC(.B) #0,dst	Increment and decrement carry.

Relative Jumps

Conditional jumps support program branching relative to the PC and do not affect the status bits. The possible jump range is from -511 to +512 words relative to the PC value at the jump instruction. The 10-bit program-counter offset is treated as a signed 10-bit value that is doubled and added to the program counter:

PCnew = PCold + 2 + PCoffset × 2

Table 3.3.1.2.4.3-5: Relative Jumps

Opcode	Mnemonic	Jump Condition
000	JNE/JNZ	Z == 0
001	JEQ/JZ	Z == 1
010	JNC/JLO	C == 0
011	JC/JHS	C == 1
100	JN	N == 1
101	JGE	N == V
110	JL	N != V
111	JMP	unconditionally

3.3.1.2.4.4 Instruction Cycle Counts

command type	operation	cycles (dreg != PC)	cycles (dreg == PC)
MOV	sreg -> dreg	1	2
DOUBLE	sreg x dreg -> dreg	1	2
MOV	sreg -> Y(dreg)	2	
DOUBLE	sreg x Y(dreg) -> Y(dreg)	4	
MOV	@sreg -> dreg	3	4
DOUBLE	@sreg x dreg -> dreg	3	4
MOV	@sreg -> Y(dreg)	3	
DOUBLE	@sreg x Y(dreg) -> Y(dreg)	5	
MOV	#N -> dreg	2	3
DOUBLE	#N x dreg -> dreg	2	3
MOV	@sreg+ -> dreg	3	4
DOUBLE	@sreg+ x dreg -> dreg	3	4
MOV	#N -> Y(dreg)	3	
DOUBLE	#N x Y(dreg) -> Y(dreg)	5	
MOV	@sreg+ -> Y(dreg)	3	
DOUBLE	@sreg+ x Y(dreg) -> Y(dreg)	5	
MOV	X(sreg) -> dreg	4	5
DOUBLE	X(sreg) x dreg -> dreg	4	5
MOV	X(sreg) -> Y(dreg)	3	
DOUBLE	X(sreg) x Y(dreg) -> Y(dreg)	5	
SINGLE	dreg	1	2
SINGLE	@dreg	3	
SINGLE	#N	2	
SINGLE	@dreg+	3	
SINGLE	Y(dreg)	4	
JUMP		2	
RETI		4	
IRQE		3	
PUSH	reg	1	
PUSH	@reg	2	
PUSH	#N	2	
PUSH	@reg+	2	
PUSH	X(reg)	3	
CALL	reg	2	
CALL	@reg	3	
CALL	#N	3	
CALL	@reg+	3	
CALL	X(reg)	4	

notes:
SINGLE includes RRC, RRA, SWPB and SXT
DOUBLE includes all double operand instructions except MOV

Figure 3.3.1.2.4.4-1: Cycle Count Table

3.3.1.2.4.5 JTAG Debug Interface
To access the debug structures a standard JTAG interface is used.

The debugging logic provides the following features:

- CPU register read and write access
- Data bus (memory) read and write access
- Breakpoint logic
- IAR can be use as debug IDE

The H430 embedded breakpoint logic provides the following features:

- 3 breakpoint triggers

- each trigger can match a separate address or data bus value
- a trigger value compare mask can be defined
- trigger can match a greater, smaller, equal or non equal value
- trigger can be configured for read / write or instruction fetch / non instruction fetch bus cycles
- triggers can be combined (trigger dependency)
- all breakpoints can be used for stepping and run-stop a program

3.3.1.2.5 Sub Parts
3.3.1.2.5.1 Vector Interrupt Control Module (VIC)
Two Stage Vector Interrupt System

Description

The Vector Interrupt System is a two stage interrupt handling structure. The first stage is located inside the interrupt capable digital modules. The second stage collects all module interrupts and provides a single interrupt signal to the CPU. All module interrupts provided to the main interrupt controller are level interrupts.
The Vector Interrupt Control (VIC) logic - included in every module and the main interrupt controller - is build as follows :
The incoming interrupt sources are latched by hold elements if the interrupt source is classified to be an "event". "level" interrupt sources are not latched to hold elements. "event" interrupt sources are usually conditions which are active for a very short time and they need to be latched to be handled. Their latched status flag has to be cleared by the interrupt handling routine. "level" interrupt sources are usually slow signals and their status changes by the interrupt handling itself which removes the interrupt condition.
The unmasked interrupt status can be read via the IRQ_STATUS register. Writing to the IRQ_STATUS register clears all "event" status bits which are written as one. The value of IRQ_MASK bit wise makes the interrupt status. The IRQ_MASK register can be written directly or modified using the IRQ_VENABLE and IRQ_VDISABLE registers. These two registers implement a fast vector based mask modification possibility.
The masked interrupt status is converted to an integer value and compared with the value of the IRQ_VMAX register. It defines a maximum interrupt vector level for the outgoing interrupt.
The IRQ_VNO register implements the possibility to read the current interrupt vector of the highest priority. Low vector numbers have high priority. This value can be used for a fast table based interrupt routine entry. A write access to the IRQ_VNO register clears the interrupt status bit of the written vector.

VIC Logic Structure:

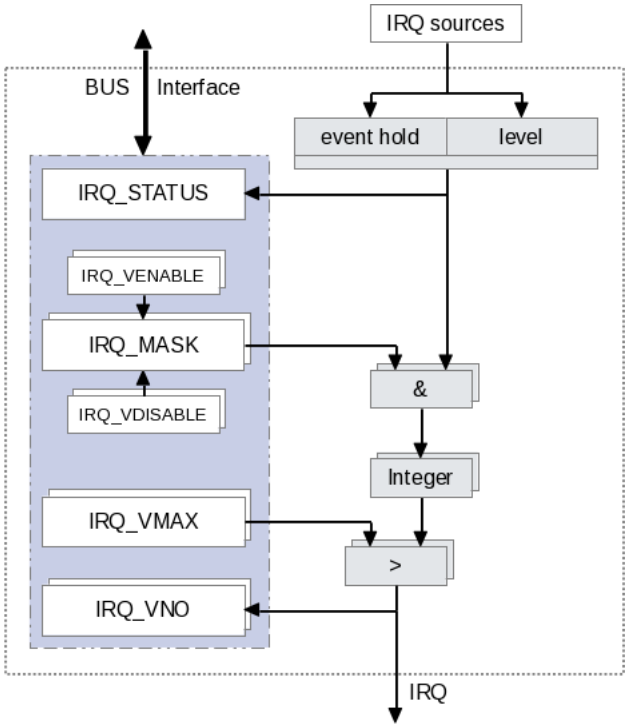


Figure 3.3.1.2.5.1-1: VIC logic structure

Two Stage Interrupt System Structure:

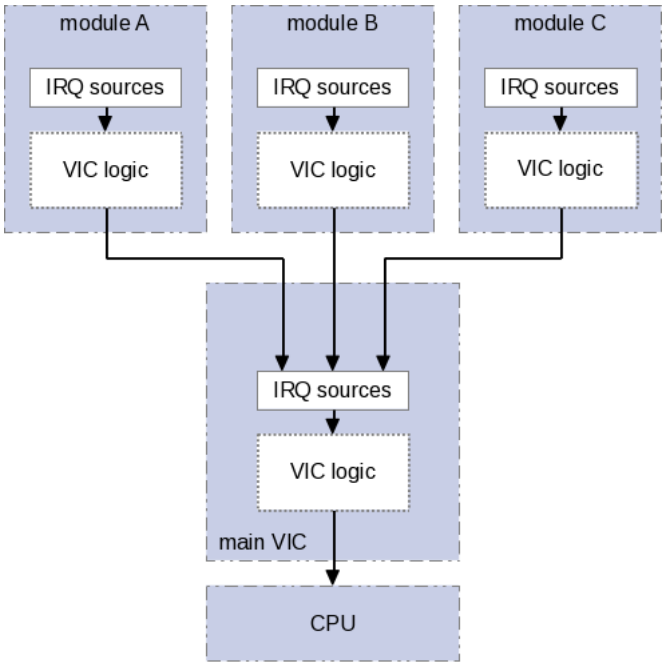


Figure 3.3.1.2.5.1-2: Two stage interrupt system structure

Features

- IRQ number for fast IRQ processing
- Main IRQ enable to enable or disable all IRQs
- Main IRQ enable MIE for easy cli() and sei() implementation
- IRQ base address for IRQ vector table in memory
- Prioritized IRQ sources where irq 0 has highest priority
- Fast vector based interrupt enable and disable
- Nested IRQ support

Table 3.3.1.2.5.1-1: Registers

Register Name	*Address*	*Description*
TABLE_BASE	0x00	table base register
TABLE_TYPE	0x02	table type register
MAIN_ENABLE	0x04	IRQ main enable register
IRQ_STATUS0	0x30	IRQ status register 0
IRQ_STATUS1	0x32	IRQ status register 1
IRQ_MASK0	0x34	IRQ mask register 0
IRQ_MASK1	0x36	IRQ mask register 1
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.1-2: Register **TABLE_BASE** (0x00) table base register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : base - base address of vector table in memory															

Table 3.3.1.2.5.1-3: Register **TABLE_TYPE** (0x02) table type register

	MSB															*LSB*
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : type - auto combine vector number and table base to create vector number related CPU interrupt pointer. 0: base value is combined with vector number to be used as CPU interrupt pointer (an interrupt service routine per module) 1: base value is directly used as CPU interrupt pointer (one common interrupt service routine)															

Table 3.3.1.2.5.1-4: Register **MAIN_ENABLE** (0x04) IRQ main enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : enable - main interrupt enable / disable 1: enabled 0: disabled															

Table 3.3.1.2.5.1-5: Register **IRQ_STATUS0** (0x30) IRQ status register 0

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : gpio_b **14** : gpio_a **13** : cctimer_3 **12** : cctimer_2 **11** : cctimer_1 **10** : cctimer_0 **9** : sci **8** : spi_1 **7** : spi_0 **6** : pre_pwm **5** : pwmn **4** : adc_ctrl **3** : divider **2** : wdog **1** : sys_state **0** : mem_prot															

Table 3.3.1.2.5.1-6: Register **IRQ_STATUS1** (0x32) IRQ status register 1

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : gpio_c															

Table 3.3.1.2.5.1-7: Register **IRQ_MASK0** (0x34) IRQ mask register 0

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.1-8: Register **IRQ_MASK1** (0x36) IRQ mask register 1

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.1-9: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W
Bit Description	**4:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.1-10: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W
Bit Description	**4:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.1-11: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.1-12: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.2 Watchdog Module (WDOG)

Features

- 8 bit pre-scaler
  - pre-scaler is driven by system clock
- 16 bit decrementing timer
  - this timer is driven by pre-scaled system clock
- the window-watchdog triggers a system reset when counter value = 0
- when system clock is not running or stops the watchdog will assert a system reset
  - the watchdog clock is used to implement this feature
- when watchdog clock oscillator is not running or stops a system reset is asserted
  - the system clock is used to implement this feature
- window-watchdog timer is disabled after reset and has to be armed by software
- window-watchdog generates an interrupt when watchdog is restarted outside specified window
- window-watchdog cannot be disabled or changed when armed
- NOTE: watchdog will be halted during FLASH erase / program
- NOTE: watchdog will be halted during CPU debug halt

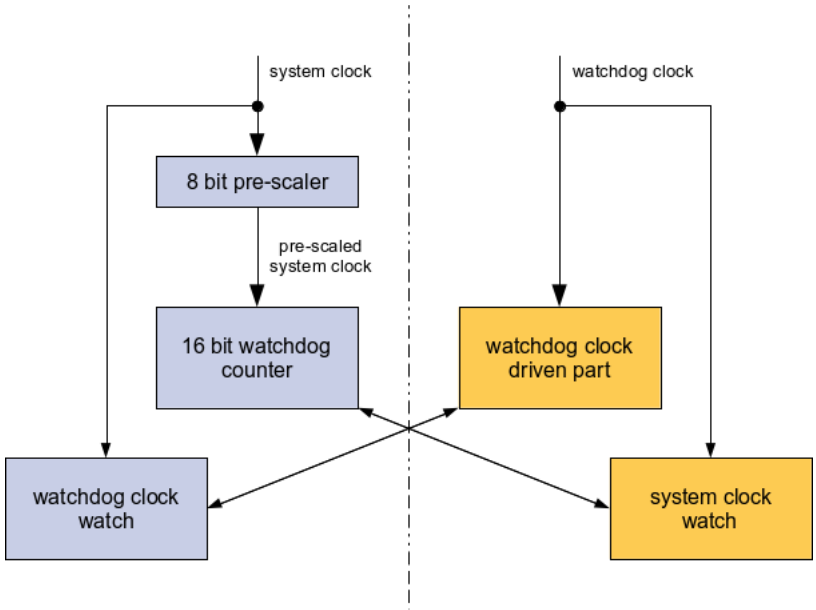


Figure 3.3.1.2.5.2-1: Structure

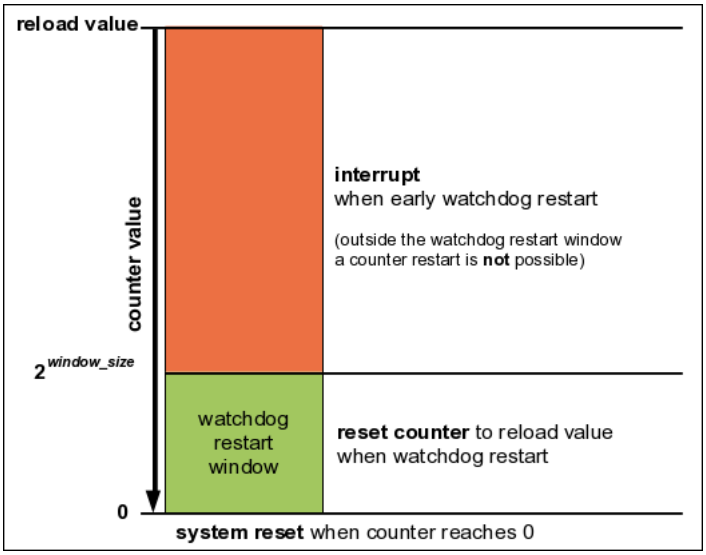


Figure 3.3.1.2.5.2-2: Timing

Table 3.3.1.2.5.2-1: Registers

Register Name	Address	Description
CONTROL	0x00	control register
WINDOW	0x02	window configuration register
PRESCALER	0x04	pre-scaler configuration register
RELOAD	0x06	counter reload value register
COUNTER	0x08	current counter value register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.2-2: Register **CONTROL** (0x00) control register

	MSB															***LSB***
Content	15:8								-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **1** : restart - 0 : no influence 1 : restart watchdog **0** : run_enable - 0 - watchdog stopped 1 - watchdog enabled															

Table 3.3.1.2.5.2-3: Register **WINDOW** (0x02) window configuration register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4** : enable - 0 - no window 1 - window active **3:0** : size - reset window is defined as: counter value < (2^window size)															

Table 3.3.1.2.5.2-4: Register **PRESCALER** (0x04) pre-scaler configuration register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : pre-scaler - watchdog counter pre-scaler (cycles = pre-scaler+1)															

Table 3.3.1.2.5.2-5: Register **RELOAD** (0x06) counter reload value register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : reload - counter restart value															

Table 3.3.1.2.5.2-6: Register **COUNTER** (0x08) current counter value register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : value - current counter value															

Table 3.3.1.2.5.2-7: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : evt_window (event) - watchdog restart before the "watchdog reset window"															

Table 3.3.1.2.5.2-8: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.2-9: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit Description	**0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.2-10: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit Description	**0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.2-11: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.2-12: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no IRQ is pending the first unused IRQ number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.3 Extended Multiplier Module (H430_MUL)

The hardware multiplier is a memory mapped peripheral (at a fixed address range from 0x130 to 0x13F). It can be accessed by CPU with full support of common compilers. Though the hardware architecture is different, the unit is fully compatible with the MPY16 multiplier unit in chips of the MSP430 family, providing the same interface, software support, and arithmetic results.

Features

- unsigned/signed multiplication (MPY / MPYS)
- unsigned/signed MAC (multiply and accumulate) operation (MAC / MACS)
- using the old result and adding the new product
- 16*16, 8*16, 16*8, and 8*8 bit input data width
- 32/33 bit output data width
- 1 system clock cycle calculation time
- no CPU wait states (no NOP required)
- extended functionality
  - signed x unsigned and unsigned x signed multiply and multiply-accumulate
  - 8 bit accumulator extension (total 40 bit accumulator)
  - setting of the 40 bit accumulator using a single 16bit word
    - the 16 bit word is either interpreted as signed (two's complement) or as Q1.15 (fractional) value
  - reading of bits 31:16 of the 40 bit accumulator with saturating and rounding
    - the 16 bit word returned is either a signed (two's complement) or a Q1.15 (fractional) value
  - 40 bit accumulator arithmetic left and right shift by up to 16 bits

The type of operation to be performed is selected by writing the first operand to one of the following four registers. Writing the first operand does not start the operation. The first operand (and thus the type of operation) may remain constant for more than one operation. Writing the second operand starts the operation.

RESLO stores the low word of the result, RESHI stores the high word of the result, and SUMEXT stores information about the result.
For signed operations, results are provided in two's complement format. The sum extension register SUMEXT allows calculations with results exceeding the 32-bit range. This read-only register holds the most significant part of the result (bits 32 and higher). The register simplifies multiple word operations, because straightforward additions can be performed without conditional jumps.

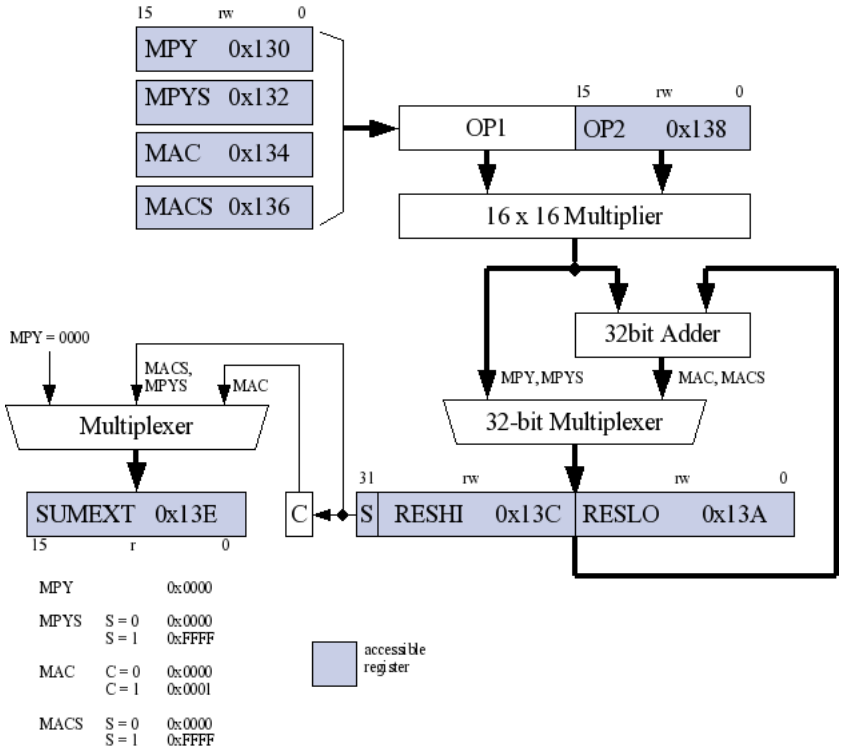


Figure 3.3.1.2.5.3-1: Multiplier Structure

Table 3.3.1.2.5.3-1: Registers

Register Name	Address	Description
LAST_MODE	0x00	last mode of multiply/MAC unit
OP2U	0x02	operand 2 unsigned register
OP2S	0x04	operand 2 signed register
OP2SN	0x06	operand 2 signed neg register
ACCU_EXT	0x08	accumulator extension register
RESHI_TC	0x0A	accu 2's complement access register
RESHI_Q1_15	0x0C	accu fractional access register
SHIFT_RIGHT	0x0E	accumulator shift right register
SHIFT_LEFT	0x10	accumulator shift left register
MPY	0x30	multiply unsigned register
MPYS	0x32	multiply signed register

Register Name	Address	Description
MAC	0x34	mac unsigned register
MACS	0x36	mac signed register
OP2	0x38	operand 2 register
RESLO	0x3A	sum register (low 16 bit)
RESHI	0x3C	sum register (high 16 bit)
SUMEXT	0x3E	sum extension register

Table 3.3.1.2.5.3-2: Register **LAST_MODE** (0x00) last mode of multiply/MAC unit

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**1:0** : last mode of multiply/MAC unit: 0x0 = MPY 0x1 = MPYS 0x2 = MAC 0x3 = MACS															

Table 3.3.1.2.5.3-3: Register **OP2U** (0x02) operand 2 unsigned register

	MSB															LSB
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:0** : op2u - unsigned second operand (writing starts the operation) The format (signed/unsigned) of the first operand is determined by its own register. The type of operation ("multiply" / "multiply accumulate") is determined by the first operand register.															

Table 3.3.1.2.5.3-4: Register **OP2S** (0x04) operand 2 signed register

	MSB															LSB
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:0** : op2s - signed second operand (writing starts the operation) The format (signed/unsigned) of the first operand is determined by its own register. The type of operation ("multiply" / "multiply accumulate") is determined by the first operand register.															

Table 3.3.1.2.5.3-5: Register **OP2SN** (0x06) operand 2 signed neg register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:0** : op2sn - signed second operand, used negated (- op2sn) (writing starts the operation) The format (signed/unsigned) of the first operand is determined by its own register. The type of operation ("multiply" / "multiply accumulate") is determined by the first operand register.															

Table 3.3.1.2.5.3-6: Register **ACCU_EXT** (0x08) accumulator extension register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : accu_ext - accumulator extension to 40 bits															

Table 3.3.1.2.5.3-7: Register **RESHI_TC** (0x0A) accu 2's complement access register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : reshi_tc - 40 bit accumulator read and write access RESHI interpreted as signed integer (two's complement) writing (with sign extension) is implemented as follows: ACCU_EXT = RESHI_TC[15] ? 0xFF : 0x00 RESHI = RESHI_TC RESLO = 0x0000 reading (rounded and then saturated) is implemented as follows: tmp[39:0] = (ACCU_EXT, RESHI, RESLO); // biased rounding if tmp[15] tmp[39:16] = tmp[39:16] + 1 // including signed saturation when overflow // saturation if all bits are the same in tmp[39:31] // no overflow RESHI_TC = tmp[31:16] else if tmp[39] = 0 // overflow RESHI_TC = 0x7FFF else // underflow RESHI_TC = 0x8000															

Table 3.3.1.2.5.3-8: Register **RESHI_Q1_15** (0x0C) accu fractional access register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : reshi_q1_15 - 40 bit accumulator read and write access RESHI interpreted as Q1.15 (fractional) writing (shift right by one and sign extension) is implemented as follows: ACCU_EXT = RESHI_Q1_15[15] ? 0xFF : 0x00 RESHI[15] = RESHI_Q1_15[15] RESHI[14:0] = RESHI_Q1_15[15:1] RESLO[15] = RESHI_Q1_15[0] RESLO[14:0] = 0 reading (shift left by one, rounding and saturation) is implemented as follows: tmp[39:0] = (ACCU_EXT, RESHI, RESLO) << 1 // including signed saturation when overflow // biased rounding and saturation same as above Note: A MPYS or MACS operation of two Q1.15 values will result in a Q2.30 value in the accumulator. Therefore upon writing a shift right by one and upon reading a shift left by one is required.															

Table 3.3.1.2.5.3-9: Register **SHIFT_RIGHT** (0x0E) accumulator shift right register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : shift - arithmetic shift right of 40 bit accumulator (ACCU_EXT, RESHI, RESLO) 0 : shift by 1 ... 15 : shift by 16 The shift starts immediately after writing this register.															

Table 3.3.1.2.5.3-10: Register **SHIFT_LEFT** (0x10) accumulator shift left register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : shift - arithmetic shift left of 40 bit accumulator (ACCU_EXT, RESHI, RESLO) 0 : shift by 1 ... 15 : shift by 16 The shift starts immediately after writing this register.															

Table 3.3.1.2.5.3-11: Register **MPY** (0x30) multiply unsigned register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - unsigned multiply															

Table 3.3.1.2.5.3-12: Register **MPYS** (0x32) multiply signed register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - signed multiply															

Table 3.3.1.2.5.3-13: Register **MAC** (0x34) mac unsigned register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - unsigned multiply accumulate															

Table 3.3.1.2.5.3-14: Register **MACS** (0x36) mac signed register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - signed multiply accumulate															

Table 3.3.1.2.5.3-15: Register **OP2** (0x38) operand 2 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op2 - write access starts multiplication Note: register value sign equals previously set op1 sign.															

Table 3.3.1.2.5.3-16: Register **RESLO** (0x3A) sum register (low 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : res_lo - bits 15..0 of the result / accumulator															

Table 3.3.1.2.5.3-17: Register **RESHI** (0x3C) sum register (high 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : res_hi - bits 31..16 of the result / accumulator In case of operation: MPY: upper 16 bit of result MPYS: The MSB is the sign of the result. The remaining bits are the upper 15-bits of the result. Two's complement notation is used for the result. MAC: upper 16 bit of result MACS: Upper 16-bits of the result. Two's complement notation is used for the result. Note: When writing, ACCU_EXT will be set to 0.															

Table 3.3.1.2.5.3-18: Register **SUMEXT** (0x3E) sum extension register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : sum_ext - In case of operation: MPY: always 0x0000 MPYS: contains the extended sign of the result 0x0000 if result was positive 0xFFFF if result was negative MAC: contains the carry of the result 0x0000 no carry result 0x0001 result with carry MACS: contains the extended sign of the result 0x0000 if result was positive 0xFFFF if result was negative															

3.3.1.2.5.4 Divider Module (DIVIDER)

Features

- unsigned and signed integer divide arithmetic
- 32bit / 16bit
  - 32 bit result
  - 16 bit remainder
- 16 system clock cycles calculation time
  - if a result or remainder register is accessed before calculation has finished the read access is halted until the calculation has finished and the value is valid

Table 3.3.1.2.5.4-1: Registers

Register Name	***Address***	***Description***
OP1LO	0x00	operand 1 (low 16 bit)
OP1HI	0x02	operand 1 (high 16 bit)
OP2	0x04	unsigned operand 2 register
OP2S	0x06	signed operand 2 register
RESULTLO	0x08	result register (low 16 bit)
RESULTHI	0x0A	result register (high 16 bit)
REMAINDER	0x0C	remainder register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.4-2: Register **OP1LO** (0x00) operand 1 (low 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - operand 1 (lower 16 bit)															

Table 3.3.1.2.5.4-3: Register **OP1HI** (0x02) operand 1 (high 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op1 - operand 1 (higher 16 bit)															

Table 3.3.1.2.5.4-4: Register **OP2** (0x04) unsigned operand 2 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op2 - write access starts unsigned operation															

Table 3.3.1.2.5.4-5: Register **OP2S** (0x06) signed operand 2 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : op2 - write access starts signed operation															

Table 3.3.1.2.5.4-6: Register **RESULTLO** (0x08) result register (low 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : result - equals "op1 div op2" (lower part)															

Table 3.3.1.2.5.4-7: Register **RESULTHI** (0x0A) result register (high 16 bit)

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : result - equals "op1 div op2" (higher part)															

Table 3.3.1.2.5.4-8: Register **REMAINDER** (0x0C) remainder register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : remainder - equals "op1 mod op2"															

Table 3.3.1.2.5.4-9: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : evt_div_by_zero (event) - divide by zero event															

Table 3.3.1.2.5.4-10: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.4-11: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.4-12: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.4-13: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.4-14: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no IRQ is pending the first unused IRQ number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.5 Memory Protection Module (MEM_PROT)
This module configures the accessability of the memory space.
It implements an opcode execution protection, a System ROM read protection and a stack area type configuration.

Features
- opcode execute area configuration
  - granularity: 1KByte
- System ROM read configuration
  - granularity: 1KByte
- stack area configuration
  - granularity: 256Byte

Table 3.3.1.2.5.5-1: Registers

Register Name	***Address***	***Description***
EXEC_ENABLE_0	0x00	execute enable register 0
EXEC_ENABLE_1	0x02	execute enable register 1
EXEC_ENABLE_2	0x04	execute enable register 2
EXEC_ENABLE_3	0x06	execute enable register 3
STACK_ENABLE	0x08	stack enable register
SRAM_WR_ENABLE	0x0A	sram write enable register
ACCESS_ADDR	0x10	access address register
ACCESS_PC	0x12	access PC register
ACCESS_TYPE	0x14	access type register
ACCES_CLEAR	0x16	access clear register
SYSROM_RD_ENABLE	0x20	System ROM read enable register

Register Name	Address	Description
SYSROM_RD_ENABLE_LOW	0x22	System ROM read enable low register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.5-2: Register **EXEC_ENABLE_0** (0x00) execute enable register 0

	MSB															LSB
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 1 KByte Note: this configuration applies to CPU instruction bus access 0 - execution of opcode denied 1 - execution of opcode allowed bit 15 : area 0x3000 to 0x3FFE ... bit 1 : area 0x0400 to 0x07FE bit 0 : area 0x0000 to 0x03FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset															

Table 3.3.1.2.5.5-3: Register **EXEC_ENABLE_1** (0x02) execute enable register 1

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 1 KByte Note: this configuration applies to CPU instruction bus access 0 - execution of opcode denied 1 - execution of opcode allowed bit 15 : area 0x7000 to 0x7FFE ... bit 1 : area 0x4400 to 0x47FE bit 0 : area 0x4000 to 0x43FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset															

Table 3.3.1.2.5.5-4: Register **EXEC_ENABLE_2** (0x04) execute enable register 2

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 1 KByte Note: this configuration applies to CPU instruction bus access 0 - execution of opcode denied 1 - execution of opcode allowed bit 15 : area 0xB000 to 0xBFFE ... bit 1 : area 0x8400 to 0x87FE bit 0 : area 0x8000 to 0x83FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset															

Table 3.3.1.2.5.5-5: Register **EXEC_ENABLE_3** (0x06) execute enable register 3

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 1 KByte Note: this configuration applies to CPU instruction bus access 0 - execution of opcode denied 1 - execution of opcode allowed bit 15 : area 0xF000 to 0xFFFE ... bit 1 : area 0xC400 to 0xC7FE bit 0 : area 0xC000 to 0xC3FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset															

Table 3.3.1.2.5.5-6: Register **STACK_ENABLE** (0x08) stack enable register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 256 Byte Note: this configuration applies to CPU data bus stack access 0 - stack not allowed 1 - stack allowed bit 15 : area 0x1F00 to 0x1FFE ... bit 1 : area 0x1100 to 0x11FE bit 0 : area 0x1000 to 0x10FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset															

Table 3.3.1.2.5.5-7: Register **SRAM_WR_ENABLE** (0x0A) sram write enable register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 256 Byte 0 - area write denied 1 - area write allowed bit 15 : area 0x1F00 to 0x1FFE ... bit 1 : area 0x1100 to 0x11FE bit 0 : area 0x1000 to 0x10FE Note: Access violation is handled by SYS_STATE module and can be configured to lead to an interrupt or reset Note: Write access to the area from 0x0F00 to 0x0FFF is always allowed															

Table 3.3.1.2.5.5-8: Register **ACCESS_ADDR** (0x10) access address register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : addr - address of the FIRST detected bad access event															

Table 3.3.1.2.5.5-9: Register **ACCESS_PC** (0x12) access PC register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : pc - CPU PC value of the FIRST detected bad access event															

Table 3.3.1.2.5.5-10: Register **ACCESS_TYPE** (0x14) access type register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	6:4			-	2:0		
Reset value	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**6:4** : bus - bus of the FIRST detected bad access event 0 : CPU instruction bus 1 : CPU data bus 2 : motor peripherals DMA bus 3 : SCI DMA bus Note: value 7 means no bad access event has occured. **2:0** : type - type of the FIRST detected bad access event 0-1 : equals interrupt vector number of related event See IRQ_STATUS flag bits for details. 2 : Execute Protection 3 : Stack Protection 4 : SRAM Write Protection 5 : System ROM Read Protection Note: value 7 means no bad access event has occurred.															

Table 3.3.1.2.5.5-11: Register **ACCES_CLEAR** (0x16) access clear register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W
Bit Description	**0** : clear - write to this register clears the ACCESS_* registers and reenables them to capture the next bad access event															

Table 3.3.1.2.5.5-12: Register **SYSROM_RD_ENABLE** (0x20) System ROM read enable register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : enable - area size: 1K Byte Note: this configuration applies to CPU data bus, motor control and SCI DMA bus access Note: bits can only be set to 0 ... once 0, a bit can not be changed to 1 again ! 0 - area read denied 1 - area read allowed bit 15 : area 0x7C00 to 0x7FFE ... bit 1 : area 0x4400 to 0x47FE bit 0 : area 0x4000 to 0x43FE															

Table 3.3.1.2.5.5-13: Register **SYSROM_RD_ENABLE_LOW** (0x22) System ROM read enable low register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : area size: 1K Byte Note: this configuration applies to CPU data bus, motor control and SCI DMA bus access Note: bits can only be set to 0 ... once 0, a bit can not be changed to 1 again ! 0 - area read denied 1 - area read allowed bit 7 : area 0x3C00 to 0x3FFE ... bit 1 : area 0x2400 to 0x27FE bit 0 : area 0x2000 to 0x23FE															

Table 3.3.1.2.5.5-14: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : evt_dmisaligned - misaligned 16 bit data access event **0** : evt_undefined - undefined opcode event															

Table 3.3.1.2.5.5-15: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.5-16: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.5-17: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.5-18: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.5-19: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.6 System State Module (SYS_STATE)

Features

- system clock frequency selection
- module clock enable (disabled modules save power)
  - Note: module has to be enabled before used by software
- reset source status
- bitwise reset status clear
- reset enable
- analog part calibration registers
- IC version
- Customer ID
- target die clock generator
  - f_target = f_sys_clk / (target_clk_h + target_clk_l + 2)

Note: Note: Once set, RESET_ENABLE and CALIBRATION_LOCK bits cannot be cleared again.
Note: CALIBRATION, IC_VERSION_X, CUSTOMER_ID and DEVICE_ID will be locked by CALIBRATION_LOCK. These registers will be initialized and locked by INFO BOOT program after power up and are therefore not changeable by customer software.

Table 3.3.1.2.5.6-1: Registers

Register Name	*Address*	*Description*
MODULE_ENABLE	0x00	module enable register
CONTROL	0x02	system control register
RESET_STATUS	0x04	reset status register
RESET_STATUS_CLEAR	0x06	reset status clear register
RESET_ENABLE	0x08	reset enable register Note: Once set, RESET_ENABLE bits cannot be cleared again.
SW_RESET	0x0A	sw reset register
ENABLE_JTAG	0x0C	enable JTAG register
DEVICE_ID	0x0E	Device ID register
SIGNATURE_0	0x10	Signature Register
SIGNATURE_1	0x12	Signature Register
SIGNATURE_2	0x14	Signature Register
SIGNATURE_3	0x16	Signature Register
CALIBRATION	0x18	calibration data register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.6-2: Register **MODULE_ENABLE** (0x00) module enable register

	MSB															***LSB***
Content	-	-	13	12	11	10	9	8	7	6	5	4	3	2	-	-
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Bit Description	**13** : pwmn - PWMN module enable **12** : pre_pwm - Pre PWM module enable **11** : saradc_ctrl - SAR ADC control module enable **10** : cctimer_3 - CCTIMER 3 module enable **9** : cctimer_2 - CCTIMER 2 module enable **8** : cctimer_1 - CCTIMER 1 module enable **7** : cctimer_0 - CCTIMER 0 module enable **6** : gpio_c - GPIO C module enable **5** : gpio_b - GPIO B module enable **4** : gpio_a - GPIO A module enable **3** : spi_1 - SPI 1 module enable **2** : spi_0 - SPI 0 module enable															

Table 3.3.1.2.5.6-3: Register **CONTROL** (0x02) system control register

	MSB															***LSB***
Content	-	-	-	-	-	-	9:7			6:4			3	2:0		
Reset value	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**9:7** : target_clk_h - target die clock high period cycles selection (base clock: f_sys_osc/2) 0: 1 cycle 1: 2 cycles 2: 3 cycles 3: 4 cycles ... 7: 8 cycles **6:4** : target_clk_l - target die clock low period cycles selection (base clock: f_sys_osc/2) 0: 1 cycle 1: 2 cycles 2: 3 cycles 3: 4 cycles ... 7: 8 cycles **3** : drv_strength - pad drive strength selection 0: 4mA 1: 8mA **2:0** : sys_clk_sel - system clock selection 0 : f_sys_clk=f_sys_osc/12 (4 MHz) 1 : f_sys_clk=f_sys_osc/6 (8 MHz) 2 : f_sys_clk=f_sys_osc/4 (12 MHz) 3 : f_sys_clk=f_sys_osc/2 (24 MHz) 4 : f_sys_clk=f_sys_osc (48 MHz) Example values in brackets refer to System Oscillator Frequency f_sys_osc = 48MHz.															

Table 3.3.1.2.5.6-4: Register **RESET_STATUS** (0x04) reset status register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15** : sw_bl_reset_1 - software reset flag (used for bootloader functionality) **14** : sw_bl_reset_0 - software reset flag (used for bootloader functionality) **13** : exec_prot - Execution protection reset flag **12** : stack_prot - Stack protection reset flag **11** : sram_wr_prot - Write access to protected sram reset flag **10** : flash_1bit_err - FLASH single bit error reset flag **9** : flash_2bit_err - FLASH double bit error reset flag **8** : sram_parity - SRAM parity reset flag **7** : cpu_parity - CPU register parity reset flag **6** : sw_reset - software reset flag **5** : watchdog - watchdog reset flag **4** : sys_clk_fail - watchdog system clock watch reset flag **3** : nrsti - NRSTI pad reset flag **2** : vddc_ok - vddc_ok reset flag **1** : vddio_ok - vddio_ok reset flag **0** : por - power on reset flag															

Table 3.3.1.2.5.6-5: Register **RESET_STATUS_CLEAR** (0x06) reset status clear register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:0** : clear - writing '1' clears related bit of RESET_STATUS register															

Table 3.3.1.2.5.6-6: Register **RESET_ENABLE** (0x08) reset enable registerNote: Once set, RESET_ENABLE bits cannot be cleared again.

	MSB															***LSB***
Content	-	-	-	-	-	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
Bit Description	**10** : sw_bl_reset_1 - software reset enable **9** : sw_bl_reset_0 - software reset enable **8** : exec_prot - Execution protection reset enable **7** : stack_prot - Stack protection reset enable **6** : sram_wr_prot - Write access to protected sram reset enable **5** : flash_1bit_err - FLASH single bit error reset enable **4** : flash_2bit_err - FLASH double bit error reset enable **3** : sram_parity - SRAM parity reset enable **2** : cpu_parity - CPU register parity reset enable **1** : software - software reset enable **0** : watchdog - watchdog reset enable															

Table 3.3.1.2.5.6-7: Register **SW_RESET** (0x0A) sw reset register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	R
Bit Description	**3** : sw_bl_reset_1 - assert a system reset Note: RESET_ENABLE.sw_bl_reset_1 has to be set to 1 before **2** : sw_bl_reset_0 - assert a system reset Note: RESET_ENABLE.sw_bl_reset_0 has to be set to 1 before **1** : sw_reset - assert a system reset which also clears SW_RESET.por_flag Note: RESET_ENABLE.software has to be set to 1 before **0** : por_flag - separate power-on-reset flag for bootloader usage															

Table 3.3.1.2.5.6-8: Register **ENABLE_JTAG** (0x0C) enable JTAG register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Bit Description	**1** : exit_boot_loader exit boot loader request (digital TMR) **0** : enable enable CPU and FLASH test interface JTAG access															

Table 3.3.1.2.5.6-9: Register **DEVICE_ID** (0x0E) Device ID register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : id - dual die device identification number which can be read via JTAG to identify the current combination of micro controller and analog die to allow individual device handling Note: Once written, this value cannot be changed again.															

Table 3.3.1.2.5.6-10: Register **SIGNATURE_0** (0x10) Signature Register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - customer specific signature value (bits 15:0) may be used by software as valid key value to allow FLASH read via LIN (eg. an access key is received via LIN and compared by software with this 64 bit signature value to allow FLASH content access)															

Table 3.3.1.2.5.6-11: Register **SIGNATURE_1** (0x12) Signature Register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - customer specific signature value (bits 31:16) may be used by software as valid key value to allow FLASH read via LIN (eg. an access key is received via LIN and compared by software with this 64 bit signature value to allow FLASH content access)															

Table 3.3.1.2.5.6-12: Register **SIGNATURE_2** (0x14) Signature Register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - customer specific signature value (bits 47:32) may be used by software as valid key value to allow FLASH read via LIN (eg. an access key is received via LIN and compared by software with this 64 bit signature value to allow FLASH content access)															

Table 3.3.1.2.5.6-13: Register **SIGNATURE_3** (0x16) Signature Register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - customer specific signature value (bits 63:48) may be used by software as valid key value to allow FLASH read via LIN (eg. an access key is received via LIN and compared by software with this 64 bit signature value to allow FLASH content access)															

Table 3.3.1.2.5.6-14: Register **CALIBRATION** (0x18) calibration data register

	MSB															***LSB***
Content	15	-	-	-	11:8				7:4				-	2:0		
Reset value	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
Access	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : lock write-locks this register Note: Once written as 1, this bit cannot be cleared again. **11:8** : msb_offset ADC MSB related offset correction value (signed) Note: this value will be added to ADC sample data if ADC sample data MSB=1 **7:4** : cal_bgap - bandgap selection (trim) value **2:0** : cal_vref - reference voltage trim value															

Table 3.3.1.2.5.6-15: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10** : sw_bl_reset_1 - software bootloader reset event 1 **9** : sw_bl_reset_0 - software bootloader reset event 0 **8** : exec_prot - Execution protection event **7** : stack_prot - Stack protection event **6** : sram_wr_prot - Write access to protected SRAM event **5** : flash_1bit_err - Flash one bit error corrected **4** : flash_2bit_err - Flash two bit error detected **3** : sram_parity - SRAM parity error **2** : cpu_parity - CPU parity error **1** : software - software reset event **0** : watchdog - Watchdog event															

Table 3.3.1.2.5.6-16: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	10:0										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.6-17: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.6-18: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.6-19: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.6-20: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.7 IO MUX Control Module (IOMUX_CTRL)

3.3.1.2.5.7.1 Description

This module can be used to configure the digital module IO signal to device pin assignment.

3.3.1.2.5.7.2 Operating environment

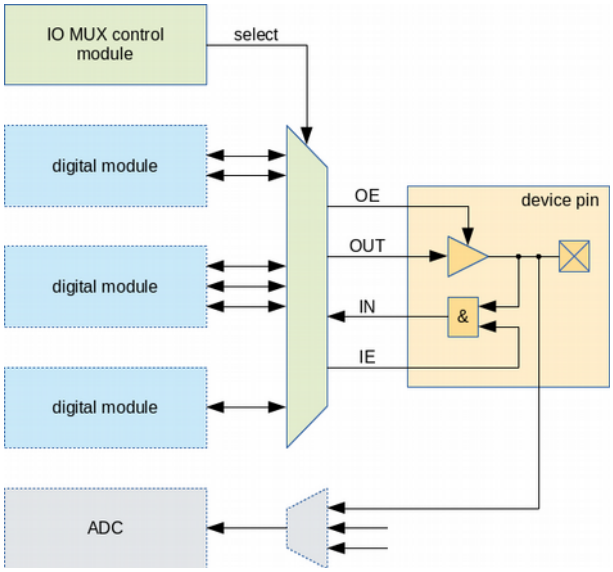


Figure 3.3.1.2.5.7.2-1: operating environment

3.3.1.2.5.7.3 Features

- the device implements 3 IO groups: PA, PB, PC
- each IO group includes 8 pins, e.g. PA group includes PA0 ... PA7
- a number of module IO signals can be connected to the pins
  - for each PC group pin, one of 16 possible module IO signals can be selected
  - for each PA or PB group pin, one of 4 possible module IO signals can be selected
  - the following tables define the possible module IO signal selections
    - please see the related module descriptions for module signal explanation
- the lock registers can be used to lock a previously set IO configuration to prevent it from changing by software write accesses
- to configure a pin to be used as analog input (ADC input signal), please select the GPIO signal for this pin and make sure, that the output enable bit of the GPIO signal is not set in the related GPIO module register

3.3.1.2.5.7.4 PA IO group pins signal selection table

- for each PA pin, one of 4 signals can be selected from the tables below
- each PA pin has it's own 4 to 1 IO signal multiplexer which can be configured using a 2 bit select value
- for every PA pin a different select value can be configured

Table 3.3.1.2.5.7.4-1: PA signal select table (select = 0)

pin name	*module name*	*signal name*	*signal direction*
PA 0	GPIO_A	IO0	in / out
PA 1	GPIO_A	IO1	in / out

pin name	module name	signal name	signal direction
PA 2	GPIO_A	IO2	in / out
PA 3	GPIO_A	IO3	in / out
PA 4	GPIO_A	IO4	in / out
PA 5	GPIO_A	IO5	in / out
PA 6	GPIO_A	IO6	in / out
PA 7	GPIO_A	IO7	in / out

Table 3.3.1.2.5.7.4-2: PA signal select table (select = 1)

pin name	module name	signal name	signal direction
PA 0	SPI_0	SCK	in / out
PA 1	SPI_0	SDI	in
PA 2	SPI_0	SDO	out
PA 3	SPI_0	NSS	in / out
PA 4	SPI_0	SCK	in / out
PA 5	PWMN	LS_U	in / out
PA 6	PWMN	LS_V	out
PA 7	PWMN	LS_W	out

Table 3.3.1.2.5.7.4-3: PA signal select table (select = 2)

pin name	module name	signal name	signal direction
PA 0	PWMN	NALLOFF	in
PA 1	CCTIMER_0	MEAS	in
PA 2	SARADC_CTRL	MUX_1	out
PA 3	SARADC_CTRL	MUX_2	out
PA 4	PWMN	LS_U	out
PA 5	PWMN	LS_V	out
PA 6	PWMN	LS_W	out
PA 7	CCTIMER_1	MEAS	in

Table 3.3.1.2.5.7.4-4: PA signal select table (select = 3)

pin name	module name	signal name	signal direction
PA 0	SCI	RXD	in
PA 1	SCI	TXD	out
PA 2	PWMN	LS_X	out
PA 3	PWMN	HS_X	out
PA 4	SYS_STATE	TARGET_CLK	out
PA 5	CCTIMER_0	PWM	out
PA 6	CCTIMER_1	PWM	out
PA 7	SARADC_CTRL	MUX_1	out

3.3.1.2.5.7.5 PB IO group pins signal selection table

- for each PB pin, one of 4 signals can be selected from the tables below
- each PB pin has it's own 4 to 1 IO signal multiplexer which can be configured using a 2 bit select value
- for every PB pin a different select value can be configured

Table 3.3.1.2.5.7.5-1: PB signal select table (select = 0)

pin name	*module name*	*signal name*	*signal direction*
PB 0	GPIO_B	IO0	in / out
PB 1	GPIO_B	IO1	in / out
PB 2	GPIO_B	IO2	in / out
PB 3	GPIO_B	IO3	in / out
PB 4	GPIO_B	IO4	in / out
PB 5	GPIO_B	IO5	in / out
PB 6	GPIO_B	IO6	in / out
PB 7	GPIO_B	IO7	in / out

Table 3.3.1.2.5.7.5-2: PB signal select table (select = 1)

pin name	*module name*	*signal name*	*signal direction*
PB 0	SCI	RXD	in
PB 1	SCI	TXD	out
PB 2	PWMN	HS_U	out
PB 3	PWMN	HS_V	out
PB 4	PWMN	HS_W	out
PB 5	PWMN	NALLOFF	in
PB 6	SCI	RXD	in
PB 7	SCI	TXD	out

Table 3.3.1.2.5.7.5-3: PB signal select table (select = 2)

pin name	*module name*	*signal name*	*signal direction*
PB 0	CCTIMER_2	MEAS	in
PB 1	PWMN	HS_U	out
PB 2	PWMN	HS_V	out
PB 3	PWMN	HS_W	out
PB 4	SYS_STATE	TARGET_CLK	out
PB 5	SARADC_CTRL	DBG_SAMPLING	out
PB 6	PWMN	LS_X	out
PB 7	PWMN	HS_X	out

Table 3.3.1.2.5.7.5-4: PB signal select table (select = 3)

pin name	module name	signal name	signal direction
PB 0	SARADC_CTRL	MUX_2	out
PB 1	SARADC_CTRL	MUX_1	out
PB 2	SCI	RXD	in
PB 3	CCTIMER_2	PWM	out
PB 4	SCI	TXD	out
PB 5	PRE_PWM / CCTIMER	SYNC_W	in
PB 6	PRE_PWM / CCTIMER	SYNC_W	in
PB 7	SARADC_CTRL	DBG_SAMPLING	out

3.3.1.2.5.7.6 PC IO group pins signal selection table

- for each PC pin, one of 16 signals can be selected from the table below
- each PC pin has it's own 16 to 1 IO signal multiplexer which can be configured using a 4 bit select value
- for every PC pin a different select value can be configured
- the following table is valid for all PC pins

Table 3.3.1.2.5.7.6-1: PC signal select table

select	module name	signal name	signal direction
0	GPIO_C	PC pin related IO	in / out
1	SPI_1	NSS	in / out
2	SPI_1	SCK	in / out
3	SPI_1	SDO	out
4	SPI_1	SDI	in
5	CCTIMER_0	MEAS	in
6	CCTIMER_0	PWM	out
7	CCTIMER_1	MEAS	in
8	CCTIMER_1	PWM	out
9	CCTIMER_2	MEAS	in
10	CCTIMER_2	PWM	out
11	CCTIMER_3	MEAS	in
12	CCTIMER_3	PWM	out
13	PRE_PWM	SYNC_U	in
14	PRE_PWM	SYNC_V	in
15	SARADC_CTRL	SYNC_OUT	out

3.3.1.2.5.7.7 Register Interface

Table 3.3.1.2.5.7.7-1: Registers

Register Name	Address	Description
PA_IO_SEL	0x00	IO signal select register

Register Name	Address	Description
PB_IO_SEL	0x02	IO signal select register
PC0123_IO_SEL	0x04	IO signal select register
PC4567_IO_SEL	0x06	IO signal select register
PA_LOCK	0x08	IO signal select lock register
PB_LOCK	0x0A	IO signal select lock register
PC_LOCK	0x0C	IO signal select lock register

Table 3.3.1.2.5.7.7-2: Register **PA_IO_SEL** (0x00) IO signal select register

	MSB															LSB
Content	15:14		13:12		11:10		9:8		7:6		5:4		3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:14** : sel_7 - PA7 module signal assignment selection **13:12** : sel_6 - PA6 module signal assignment selection **11:10** : sel_5 - PA5 module signal assignment selection **9:8** : sel_4 - PA4 module signal assignment selection **7:6** : sel_3 - PA3 module signal assignment selection **5:4** : sel_2 - PA2 module signal assignment selection **3:2** : sel_1 - PA1 module signal assignment selection **1:0** : sel_0 - PA0 module signal assignment selection															

Table 3.3.1.2.5.7.7-3: Register **PB_IO_SEL** (0x02) IO signal select register

	MSB															LSB
Content	15:14		13:12		11:10		9:8		7:6		5:4		3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:14** : sel_7 - PB7 module signal assignment selection **13:12** : sel_6 - PB6 module signal assignment selection **11:10** : sel_5 - PB5 module signal assignment selection **9:8** : sel_4 - PB4 module signal assignment selection **7:6** : sel_3 - PB3 module signal assignment selection **5:4** : sel_2 - PB2 module signal assignment selection **3:2** : sel_1 - PB1 module signal assignment selection **1:0** : sel_0 - PB0 module signal assignment selection															

Table 3.3.1.2.5.7.7-4: Register **PC0123_IO_SEL** (0x04) IO signal select register

	MSB															***LSB***
Content	15:12				11:8				7:4				3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	/WR	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:12** : sel3 - PC3 module signal assignment selection **11:8** : sel2 - PC2 module signal assignment selection **7:4** : sel1 - PC1 module signal assignment selection **3:0** : sel0 - PC0 module signal assignment selection															

Table 3.3.1.2.5.7.7-5: Register **PC4567_IO_SEL** (0x06) IO signal select register

	MSB															***LSB***
Content	15:12				11:8				7:4				3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	/WR	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:12** : sel7 - PC7 module signal assignment selection **11:8** : sel6 - PC6 module signal assignment selection **7:4** : sel5 - PC5 module signal assignment selection **3:0** : sel4 - PC4 module signal assignment selection															

Table 3.3.1.2.5.7.7-6: Register **PA_LOCK** (0x08) IO signal select lock register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : pass - password must be written as 0xA5 will always be read as 0x96 **7:0** : lock - 0: selection unlocked 1: selection locked Note: once locked, the selection cannot be unlocked again.															

Table 3.3.1.2.5.7.7-7: Register **PB_LOCK** (0x0A) IO signal select lock register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : pass - password must be written as 0xA5 will always be read as 0x96 **7:0** : lock - 0: selection unlocked 1: selection locked Note: once locked, the selection cannot be unlocked again.															

Table 3.3.1.2.5.7.7-8: Register **PC_LOCK** (0x0C) IO signal select lock register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : pass - password must be written as 0xA5 will always be read as 0x96 **7:0** : lock - 0: selection unlocked 1: selection locked Note: once locked, the selection cannot be unlocked again.															

3.3.1.2.5.8 FLASH Control Module (FLASH_CTRL)
This module implements FLASH protection, erase and program functionality.

Execution of program code located in FLASH memory strictly requires "read" mode.
When switching to other FLASH modes (program or erase) the user has to ensure
that during this time code is not executed from the same FLASH instance.
Before returning to code in FLASH, mode has to be switched back to "read".

Features

- MAIN area protection (16 x 2K byte areas)
  - protected bits prevent related FLASH MAIN areas parts from changes by erase or program
  - protect bits can be changed as long as the corresponding lock bits are not set
  - once lock bits are set they cannot be cleared again (protection lock can only be added)
- INFO area protection (2 x 512 byte areas)
  - this protection configuration can only be set once and not changed afterwards
  - this configuration will be set by INFO boot program after power up

- INFO read protection (2 x 512 byte areas)
  - this protection configuration can only be set once and not changed afterwards
  - this configuration will be set by INFO boot program after power up
- supported modes (MAIN and INFO when not protected):
  - mass erase
  - page erase (512 byte)
  - program
  - read
- down to double-word (32 bit) programming
- system frequency adaptive
- SECDED ECC protection
  - each 16 bit data word is extended by a 6 bit ECC
  - Hamming distance: 4 (1 bit error correctable, 2 bit errors detectable)

Erase and Program Sequences

The following sequence has to be done to do some FLASH erase:

1. select erase mode by MODE register
   - MAIN mass erase / MAIN page erase / INFO page erase
2. do a single (16 bit) write to the base address of the area which has to be erased to start erase process
3. poll STATUS.busy bit to determine when erase has been finished by the FLASH control module
4. switch back to previous FLASH mode (normally MAIN read)

The following sequence has to be done to do some FLASH program:

1. select program mode by MODE register
   - MAIN program / INFO program
2. split data into FLASH row (32 x 32 bit) aligned packets
3. for all these packets do:
   1. set WORD_CONFIG to number of 16 bit words which need to be programmed to FLASH row
      - only aligned 32 bit words are valid to be programmed !
      - for this only even number of 16 bit words are valid ! (2, 4, 6, ... up to 64)
      - to align a 16 bit words use the value 0xFFFF !
   2. for all 32 bit words of current packet do:
      1. write lower 16 bit word to desired FLASH address
      2. write upper 16 bit word to desired FLASH address to start 32 bit word program process
      3. poll STATUS.busy bit to determine when word program has been finished by FLASH control module
   3. at this point STATUS.incomplete has to be 0 !
4. switch back to previous FLASH mode (normally MAIN read)
5. verify written data by read and compare

Table 3.3.1.2.5.8-1: Registers

Register Name	*Address*	*Description*
AREA_MAIN_L	0x00	lower protection register
AREA_MAIN_H	0x02	upper protection register
MODE	0x04	mode register
STATUS	0x06	status register
AREA_INFO	0x08	info area write/erase protection

Register Name	Address	Description
READ_INFO	0x0A	info area read protection
BIT_ERROR_ADDR	0x0C	bit error address register
WORD_CONFIG	0x0E	word prog config register
AREA_MAIN_L_LOCK	0x20	lower protection lock register
AREA_MAIN_H_LOCK	0x22	upper protection lock register

Table 3.3.1.2.5.8-2: Register **AREA_MAIN_L** (0x00) lower protection register

	MSB															LSB
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **7:0** : area - lower 8 FLASH MAIN area write and erase protection 1 : area protected 0 : area writable / erasable MAIN protection areas offset address spaces: MAIN area 0 : 0x0000 - 0x07FF MAIN area 1 : 0x0800 - 0x0FFF ... MAIN area 6 : 0x3000 - 0x37FF MAIN area 7 : 0x3800 - 0x3FFF Note: Bits can only be changed if corresponding lock bits are 0.															

Table 3.3.1.2.5.8-3: Register **AREA_MAIN_H** (0x02) upper protection register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **7:0** : area - upper 8 FLASH MAIN area write and erase protection 1 : area protected 0 : area writable / erasable MAIN protection areas offset address spaces: MAIN area 8 : 0x4000 - 0x47FF MAIN area 9 : 0x4800 - 0x4FFF ... MAIN area 14 : 0x7000 - 0x77FF MAIN area 15 : 0x7800 - 0x7FFF Note: Bits can only be changed if corresponding lock bits are 0.															

Table 3.3.1.2.5.8-4: Register **MODE** (0x04) mode register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **7:0** : mode - 0x01 : MAIN read 0x02 : INFO read 0x04 : MAIN program 0x08 : INFO program 0x10 : MAIN page erase 0x20 : INFO page erase 0x40 : MAIN mass erase 0x80 : MAIN and INFO mass erase any other written mode value results in FLASH MAIN read mode program/erase modes: write access to appropriate flash address starts program/erase cycle (see busy flag of status register, consider word config and row programming incomplete flag in program mode)															

Table 3.3.1.2.5.8-5: Register **STATUS** (0x06) status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**3** : recall_busy - 1 : FLASH recall is active (only used at system startup and will be handled by Startup-ROM code) **2** : write_error - 1: unexpected write to area protected memory occurred, will be cleared when STATUS is read **1** : incomplete - 1: row programming incomplete current number of programmed row words != word_config (see below) **0** : busy - 0: ready 1: busy (program or erase is still in progress)															

Table 3.3.1.2.5.8-6: Register **AREA_INFO** (0x08) info area write/erase protection

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password must be written as 0xA5 will always be read as 0x96 **7:0** : area 2 FLASH INFO area write and erase protection 1 : area protected 0 : area writable / erasable INFO protection areas offset address spaces: INFO area 0 : 0xFC00 - 0xFDFF INFO area 1 : 0xFE00 - 0xFFFF Note: This register can only be written once !															

Table 3.3.1.2.5.8-7: Register **READ_INFO** (0x0A) info area read protection

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password must be written as 0xA5 will always be read as 0x96 **7:0** : area 2 FLASH INFO area read protection in user program 1 : area protected (read data will be 0x0000) 0 : area readable INFO protection areas offset address spaces: INFO area 0 : 0xFC00 - 0xFDFF INFO area 1 : 0xFE00 - 0xFFFF Note: This register can only be written once !															

Table 3.3.1.2.5.8-8: Register **BIT_ERROR_ADDR** (0x0C) bit error address register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : addr - address offset of last FLASH bit error															

Table 3.3.1.2.5.8-9: Register **WORD_CONFIG** (0x0E) word prog config register

	MSB															***LSB***
Content	15:8								-	-	5:0					
Reset value	1	0	0	1	0	1	1	0	0	0	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **5:0** : config - number of 16 bit words to program within a row 1 : 2 words ... 31 : 32 words ... 63 : 64 words (complete row) Note: number of words has to be even (double-word programming only, low address has to be written first)															

Table 3.3.1.2.5.8-10: Register **AREA_MAIN_L_LOCK** (0x20) lower protection lock register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **7:0** : area - lower 8 FLASH MAIN areas write and erase protection locking 0 : area protection configuration not locked 1 : area protection configuration locked Note: Bits set to '1' (locked) cannot be cleared again!															

Table 3.3.1.2.5.8-11: Register **AREA_MAIN_H_LOCK** (0x22) upper protection lock register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : password - must be written as 0xA5 will always be read as 0x96 **7:0** : area - upper 8 FLASH MAIN areas write and erase protection locking 0 : area protection configuration not locked 1 : area protection configuration locked Note: Bits set to '1' (locked) cannot be cleared again!															

3.3.1.2.5.9 SPI Module (SPI)

Features

- the SPI interface consists of the following 4 signals:
  - SCK: SPI clock (driven by master)
  - NSS: low active slave select (driven by master)
  - SDO: data out
  - SDI: data in
- can be used as master or slave
  - master speed up to f_sys / 4
  - slave speed upto f_sys / 8
- configurable phase, polarity and bit order
- configurable word bit length
- multiple data word transfer support
- slave mode SPI clock monitoring (timeout)
- 4 data word transmit and receive FIFOs

Table 3.3.1.2.5.9-1: Registers

Register Name	*Address*	*Description*
DATA	0x00	data register
DATA_KEEP_NSS	0x02	keep NSS data register
CONFIG	0x04	config register
BAUD_CONFIG	0x06	baud config register
TIMEOUT_CONFIG	0x08	timeout config register
RX_FIFO_TIMEOUT	0x0A	RX FIFO timeout config register
FIFO_CLEAR	0x0C	FIFO clear register
FIFO_LEVELS	0x0E	FIFO level config register
IRQ_STATUS	0x20	IRQ status register
IRQ_MASK	0x24	IRQ mask register
IRQ_VENABLE	0x28	IRQ vector enable register
IRQ_VDISABLE	0x2A	IRQ vector disable register
IRQ_VMAX	0x2C	IRQ max vector register
IRQ_VNO	0x2E	IRQ vector number register

Table 3.3.1.2.5.9-2: Register **DATA** (0x00) data register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : write: transmit data read: receiver data does not keep csb active after data transmit															

Table 3.3.1.2.5.9-3: Register **DATA_KEEP_NSS** (0x02) keep NSS data register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:0** : write: transmit data keep nss active after data transmit															

Table 3.3.1.2.5.9-4: Register **CONFIG** (0x04) config register

	MSB															***LSB***
Content	15	-	13	12	11	10	9	8	7:4				3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
Access	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : enable - 0: SPI interface is disabled 1: SPI interface is enabled Note: enable = 0 clears interface FIFO's (as long as the interface is disabled, the FIFO's are kept clean) **13** : swap_sdi_sdo - 0: SDI = data input, SDO = data output 1: SDI = data output, SDO = data input **12** : sdi_irq_po - ISDI interrupt active level, in case of inactive NSS **11** : pause_nss - minimum inter shift NSS inactive time 0: 1 bit length 1: 2 bit lengths when SPI interface is configured as slave: this config bit is ignored **10** : invert_data - invert SPI input and output data **9** : invert_nss - invert SPI select signal 0: active low 1: active high **8** : slave_high_z - when SPI interface is configured as slave: 0: drive miso / sdo 1: miso / sdo switched to high Z when SPI interface is configured as master: this config bit is ignored **7:4** : length - data bit count to transfer data word length = length + 1 example: length = 7 => 8 bit transfer example: length = 15 => 16 bit transfer **3** : slave - 0: master 1: slave **2** : polarity - 0: clock off level 0 1: clock off level 1 **1** : phase - 0: 1st edge shift, 2nd edge sample 1: 1st edge sample, 2nd edge shift **0** : order - 0: LSB first 1: MSB first															

Table 3.3.1.2.5.9-5: Register **BAUD_CONFIG** (0x06) baud config register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : divider - divider = (system clock frequency) / (2 * baudrate) min divider value = 2 -> max baudrate = Fsys / 4 max divider value = 65535 -> min baudrate = Fsys / 131070 example: Fsys = 32MHz max baudrate = 8 MBaud min baudrate = 244 Baud															

Table 3.3.1.2.5.9-6: Register **TIMEOUT_CONFIG** (0x08) timeout config register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : sclk_timeout - sclk timeout value maximum allowed count of system clock cycles between 2 SPI clock edges															

Table 3.3.1.2.5.9-7: Register **RX_FIFO_TIMEOUT** (0x0A) RX FIFO timeout config register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : rx_fifo_timeout - time in bits until receive timeout event timeout will be restarted when new incoming data byte timeout will be set when counted to zero and RX_FIFO.state = non empty															

Table 3.3.1.2.5.9-8: Register **FIFO_CLEAR** (0x0C) FIFO clear register

	MSB															***LSB***
Content														2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W
Bit Description	**2** : reset - 1: module reset **1** : tx_fifo_clear - 1: transmit FIFO clear **0** : rx_fifo_clear - 1: receiver FIFO clear															

Table 3.3.1.2.5.9-9: Register **FIFO_LEVELS** (0x0E) FIFO level config register

	MSB															***LSB***
Content	-	14:12			-	10:8			-	6:4			-	2:0		
Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit Description	**14:12** : tx_fifo_low_water - low water transmit FIFO level interrupt will be asserted when transmit FIFO fill level decreases to this value **10:8** : rx_fifo_high_water - high water receive FIFO level interrupt will be asserted when receive FIFO fill level increases to this value **6:4** : tx_fifo_level - transmit FIFO fill level (read only fifo status) **2:0** : rx_fifo_level - receive FIFO fill level (read only fifo status)															

Table 3.3.1.2.5.9-10: Register **IRQ_STATUS** (0x20) IRQ status register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15** : tx_fifo_nfull - transmit fifo is not full **14** : tx_fifo_low_water - transmit fifo elements <= low water level **13** : tx_fifo_empty - transmit fifo is empty **12** : rx_fifo_full - receive fifo is full **11** : rx_fifo_high_water - receive fifo elements > high water level **10** : rx_fifo_timeout - receive fifo timeout **9** : rx_fifo_nempty - receive fifo is not empty **8** : evt_shift_done - per word shift interrupt when a word shift completed **7** : evt_sclk_timeout - sclk timeout interrupt **6** : evt_sdi - SDI had config.sdi_irq_pol polarity while NSS was active **5** : evt_eot - end of transfer (nss has changed to inactive level) **4** : evt_sot - start of transfer (nss has changed to active level) **3** : evt_tx_fifo_ur_err - software has not written outgoing data fast enough **2** : evt_tx_fifo_ov_err - software wrote data to full transmit fifo **1** : evt_rx_fifo_ur_err - software has read from empty receive fifo **0** : evt_rx_fifo_ov_err - software did not read incoming data fast enough															

Table 3.3.1.2.5.9-11: Register **IRQ_MASK** (0x24) IRQ mask register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.9-12: Register **IRQ_VENABLE** (0x28) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.9-13: Register **IRQ_VDISABLE** (0x2A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.9-14: Register **IRQ_VMAX** (0x2C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.9-15: Register **IRQ_VNO** (0x2E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**4:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.10 LIN SCI Module (LINSCI)

Features

UART Function

- Full duplex operation
- 8N1 data format, standard mark/space NRZ format
- Parity bit (even, odd, zero, none)
- Programmable stop bit length (1,2)
- selectable module clock (system clock (f_sys_clk) or system oscillator clock (f_sys_osc) divided by 12 based)

LIN Support:

- LIN Master task support
  - LIN Header generation (software driven)
    - Break generation(programmable break length up to 255 x T_{Bit})
    - SYNC byte transmission
    - Break delimiter length: 2
    - PID byte transfer
- LIN Slave Task Support
  - LIN header processing
    - Programmable break detection threshold(9.5xT_{Bit}, 11xT_{Bit})
    - Break measurement counter (baud clock based) to detect concurrent break events
    - Measurement counter to measure bit times (used for baud rate recovery -> module clock base): Baud Measurement Results can directly be fed into the baud register to adjust the baud rate (Baud self-synchronization with SYNC byte)
    - SYNC Byte plausibility check: check T_{Bit} every RXD edge (1us clock based) -> supported baud rates for plausibility check: 2400 to 115200 BAUD
  - LIN response frame processing
    - Collision detection with auto transmit shut down
    - Auto checksum insertion (classic or enhanced) to transmit frame
    - Checksum calculation
    - DMA driven transmit/receive
  - Timer-Compare Module
    - used for
      - LIN Bus Idle measurement
      - LIN Break measurement (used for auto addressing)
      - LIN Header/Frame Length measurement
    - auto restart with falling RXD edge
    - break detection with auto compare register pre-loading
  - Timer-Capture Module
    - used to measure the break length

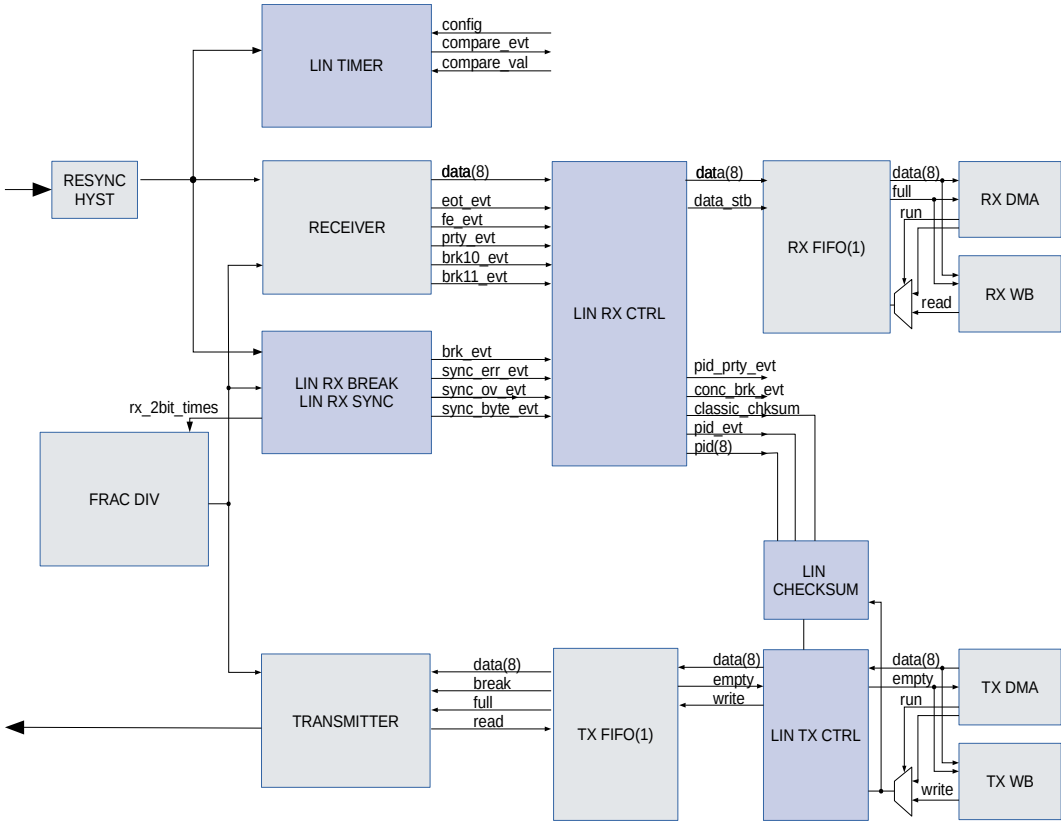


Figure 3.3.1.2.5.10-1: SCI block diagram

Functional Description
General function can be derived from Register description.

Concurrent Break Measurement
Concurrent break measurement works independent from the receiver status and detects breaks of length of 10 nominal bit length (respectively 11 nominal bit length when break threshold is set) A valid break signal starts measurement of a SYNC byte. After the SYNC byte measurement the sync_evt interrupt is raised and optionally re-triggers the fractional baud rate counter. Afterwards the PID will be received and the parity will be checked. In case of a valid parity the pid_evt interrupt will be raised and the PID will be copied to the LIN_PID register.

The concurrent break error flag will only be handled during header processing. The software has to handle occurring concurrent breaks during data transmission (unexpected break event during transfer).

Note: Since concurrent break measurement is based on the actual baud_rate and concurrent break measurement is also enabled during sync byte measurement the actual baud_rate must not exceed 10 times (respectively 11 times when LIN mode is set) the expected baud rate of the external sci. Otherwise low bits of the sync byte are detected as breaks and sync break measurement will be canceled:
Condition: Actual baud_rate < 11 times expected external baud_rate
Example: When setting internal baud_rate = 115200 concurrent baud measurement works with external baud_rates down to 10472 baud. The external baud_rate = 9600 baud cannot be synchronized.

Concurrent break measurement supports the LIN requirement of interrupting ongoing frames by a new break/sync header.

DMA
Start DMA transfer by writing a length to the LENGTH register. Set a valid base address to the DMA_ADDRESS register before. Write access to the Address register during DMA operation will be ignored.
The LENGTH register will be decremented, the ADDRESS register will be incremented with each transferred data. If an error occurs the DMA finish flag will be raised and the DMA controller will stop operation and has to be restarted by accessing the LENGTH register.
Check DMA_LENGTH register and SCI error flags when DMA finished flag is set to check if the DMA transfer aborted abnormally. Possible error cases are:
- For TXD: transmitter disabled, bus error
- For RXD: receiver disabled, bus error SCI flags are not suppressed during DMA operation. The FIFO_full flags will be handled by the DMA controller. Reading/writing of the DATA_IO register is prohibited during DMA operation.

Principle of Baud Rate Synchronization
In principal the baud rate synchronization is based on counting system clock (sys_clk) cycles, or oscillator clock (sys_osc) cycles divided by 12, for 8 x TBit. Refer to SYS_STATE.CONTROL.sys_clk_sel 3.3.1.2.5.6-3 for selecting the system clock frequency in dependency of the oscillator frequency (f_sys_osc).

The result of the bit time measurement can be fed into the fractional baud rate divider register. The error of the measurement is the ratio of the measurement clock to the Bit clock. See the following description for details.
- Sync Byte Measurement:
  - Measurement of 8 Bit Times (4 falling edges)
    - measurement counter (18bit) running at system clock frequency (f_sys_clk) or at oscillator clock frequency divided by 12 (f_sys_osc/12) (configurable)
    - measurement error$_{8Tbit}$[%] = T_{sys_clk} x 100 / (8 x T_{bit})
  - Measurement Averaging: divide counter value by 4 and rounding → 2 Bit Times in 16bit
    - measurement error after averaging and rounding error$_{2Tbit}$[%] ≈ (T_{sys_sclk}/2) x 100 / (2 x T_{bit}) (considering more accuracy because of rounding)
    - Example for 48MHz system clock and 20kbaud: (20,83ns/2) x 100 / (2 x 50µs) = 0,011% Error
    - Example for 43MHz system clock and 20kbaud: (23,26ns/2) x 100 / (2 x 50µs) = 0,012% Error
    - Example for 4MHz system clock and 20kbaud: (250ns/2) x 100 / (2 x 50µs) = 0,125% Error
- The 16 bit bit time measurement result can be fed to the fractional baud rate divider (see LIN_CONFIG.autobaud function)
  - the fractional divider interprets the value as a decimal number with 11 digits left of the decimal point and 5 fractional digits → This corresponds to a division by 32
  - the output of the fractional divider is a clock 16 times the baud rate (2 x Tbit/32 = Tbit/16). This clock is used for the LIN transmitter and receiver.
  - → no granularity error has to be added since the average output frequency corresponds to 16 times the measured bit rate: generated baud_clk = master_baud_clock + measurement_error

Table 3.3.1.2.5.10-1: Registers

Register Name	*Address*	*Description*
BAUD_RATE	0x00	Baud config register
UART_CONFIG	0x02	Control register
LIN_CONFIG	0x04	Configuration of LIN function
LIN_CONTROL	0x06	LIN control
STATUS	0x08	Status register

Register Name	Address	Description
ERROR	0x0A	Error flag register
LIN_CONFIGURATION	0x0C	LIN configuration
DATA	0x0E	LIN SCI data register
TBIT2_LENGTH	0x10	Length of $2T_{Bit}$
LIN_PID	0x12	LIN PID
LIN_CHECKSUM	0x14	LIN checksum logic
TIMER	0x16	Timer
TIMER_COUNTER	0x18	Timer Counter
TIMER_COMPARE	0x1A	Timer Compare
TIMER_CAPTURE	0x1C	Timer Capture
DMA_TX_ADDRESS	0x20	Transmit DMA Address
DMA_TX_ADDRESS_HIGH	0x22	Transmit DMA Address High Word[1)]
DMA_TX_LENGTH	0x24	Transmit DMA Length
DMA_RX_ADDRESS	0x28	Receive DMA Address
DMA_RX_ADDRESS_HIGH	0x2A	Receive DMA Address High Word[1)]
DMA_RX_LENGTH	0x2C	Receive DMA Length
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

[1)] not used (only 16 bit address widht)

Table 3.3.1.2.5.10-2: Register **BAUD_RATE** (0x00) Baud config register

	MSB															***LSB***
Content	15:5											4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:5** : DIV - SCI baud divisor select Divisor: 0x000 --> 1 (bypass divider) 0x001 --> 2 0x002 --> 3 ... 0x007 --> 8 ... **4:0** : FRAC - SCI baud divisor fractional part These bits select the number of clocks inserted in each 32 output cycle frame to achieve more timing resolution on the average baud frequency shown in the following table. FRAC[00000] = 0/32 = 0 FRAC[00001] = 1/32 = 0.03125 FRAC[00010] = 2/32 = 0.0625 ... FRAC[10000] = 16/32 = 0.5 ... FRAC[11111] = 31/32 = 0.96875 The divider can be used to achieve divisor values between 1 and 2047.96875. The baud divisor fractional part can be used to fine tune the baud rate in 1/32 steps of the divisor. Use the following formula to calculate the SCI baud rate: Baud rate = clk_{base}/(16*(DIV+FRAC)) **Note:** clk_{base} is clk_src dependent, see UART_CONFIG register for details. Note: The 16 bit baud divisor value represents the number of system clock cycles of two bit lengths. The result of a SYNC byte measurement(see below) can directly be written to the baud rate register.															

Table 3.3.1.2.5.10-3: Register **UART_CONFIG** (0x02) Control register

	MSB															***LSB***
Content	-	14:12			11	10	9	8	-	6	5	4	3:2		1	0
Reset value	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0
Access	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	MSB																LSB
Bit Description	**14:12** : debounce - Debounce filter for RXD signal, based on the oscillator clock frequency divided by 12 (f_{sys_osc}/12)) filter time: t_debounce = $2^{debounce}$ * 12/f_{sys_osc} µs 0: bypass 1: 500ns 2: 1µs 3: 2µs 4: 4µs Example values refer to System Oscillator Frequency f_{sys_osc} = 48MHz. Note: Also consider debounce filter in LIN PHY module **11** : rxd_val RXD Receiver signal (direct input) **10** : rxd_mask 0: RXD path unmasked, signal will be routed to RX logic 1: RXD path masked, RXD signal will be forced to '1' **9** : txd_val TXD value register, used when UART_CONFIG.txd_mask=1 **8** : txd_mask 0: TXD path unmasked, UART generated TXD will be used 1: TXD logic masked, UART_CONFIG.txd_val will be used as TXD signal (allows cpu generated txd) **6** : clk_src - Clock source 0: system clock used as input for fractional BAUD_RATE divider 1: oscillator clock divided by 12 used as input for fractional BAUD_RATE divider **5** : mask_brk_err - Mask break errors 0: frame error event and parity error event will be generated even when break is received 1: frame error event and parity error event will be suppressed when break is received **4** : STOP - number of stop bits 0: 1 stop bit 1: 2 stop bits **Note:** break delimiter has always 2 stop bits **3:2** : PARITY - parity selection 00: no parity bit 01: parity bit always zero 10: odd parity 11: even parity **1** : TE - transmitter enable If software clears TE while a transmission is in progress (tx_idle = 0) the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for tx_fifo_empty to go high after the last frame before clearing TE. Note: the transmitter immediately fills the transmit shift register with ones and clears the tx fifo when a collision event occurs and collision detection is enabled. **0** : RE - receiver enable RE set to '0' suppresses start bit recognition, setting RE to '1' during an ongoing transfer can cause erroneous data reception and interrupt generation setting RE to '0' during an ongoing transfer can cause erroneous data reception and interrupt generation, received data should be ignored																

Table 3.3.1.2.5.10-4: Register **LIN_CONFIG** (0x04) Configuration of LIN function

	MSB															***LSB***
Content	-	-	-	-	-	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	MSB																*LSB*
Bit Description	**10** : dma_rx_skip_last - Skip last RX byte during DMA operation Prevents last byte to be written to memory: Example: dma_rx_length = 9, DMA will handle 9 incoming data bytes but only the first 8 are written to memory. DMA finished irq will be generated after 9 data bytes. Note: Function can be used to prevent checksum to be written to memory **9** : sync_validation 0: sync byte validation logic disabled 1: sync validation enabled **Note:** Recommended for LIN communication Baud Rates between 2400 und 115200 Hz are supported by SYNC validation hardware. Too slow Baud Rates lead to sync validation hardware overflow, too fast Baud Rates lead to plausibility check errors (both resulting in a ERROR.sync_err) **8** : suppress_tx_fb 0: normal function 1: Suppress TXD -> RXD feedback loop when in transmit_mode Note: received data will be forwarded to checksum logic but not to fifo buffer **7** : chksum_insert - Auto checksum insertion 0: no automatic checksum insertion 1: automatically transmit checksum when DMA TX transfer is finished. **Note:** Auto checksum insertion only works in conjunction with DMA transfer and LIN_CONTROL.tx_chksum enabled **6** : chksum_type - checksum type selection 0: classic checksum 1: enhanced checksum (includes PID) **Note:** see chksum_enable for details **5** : chksum_enable - enable automatic checksum calculation 0: checksum auto feed hardware disabled 1: Add rx/tx data to checksum module (LIN_CONTROL.tx_chksum dependent) Initialize checksum with each pid_evt (init with zero for classic checksum, init with PID for enhanced checksum), used checksum type: for PID 0..59 lin_chksum_type will be used for PID 60..63 always classic checksum will be used **Note:** checksum hardware can also be accessed by software **4** : filter_pid - filter(mask) PID byte 0: pass through PID to receive FIFO 1: do not forward PID byte to receive FIFO **3** : header_processing - enable header processing 0: function disabled 1: enable receive header processing: BREAK/SYNC sequence will be detected, PID will be received and validated, SYNC byte will not be forwarded to receive FIFO, PID byte can be optionally be forwarded. PID reception initializes the checksum calculation and generates a header event. Note: Recommended for LIN communication Note: RX unit is disabled during sync byte reception to avoid data reception and framing errors. **2** : break_thd - break detection threshold length 0: break detection threshold 9.5 x T_{Bit} (UART mode) 1: break detection threshold 11 x T_{Bit} (LIN mode) **Note:** use msk_brk_err in UART_CONFIG to avoid frame/parity error generation during break recognition.															

Table 3.3.1.2.5.10-5: Register **LIN_CONTROL** (0x06) LIN control

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	W	W
Bit Description	**3** : rx_sleep - temporarily disable receiver logic (sleep) 0: receiver enabled (when UART_CONFIG.re=1) 1: receiver disabled(sleeping), transfers will be finished when rx_sleep is set during ongoing transfers, receiver will leave sleep mode when a valid sync event occurs (valid break/sync sequence received) Note: rx_sleep can be used to prevent data reception,e.g. when slave is not addressed by master **2** : tx_chksum 0: add rx data to checksum calculation 1: Slave Node transmit Mode: Add tx data to checksum calculation instead of rx data. Will be reset to 0 when valid LIN header is detected Note: only applicable when chksum_enable=1 **1** : abort_tx - transmitter abort stop transmitter, set txd=1, clear tx fifo, reset tx fsm to idle, stop tx DMA **0** : abort_rx - receiver abort stop receiver, clear rx fifo, reset rx fsm to idle, stop rx DMA, abort SYNC measurements															

Table 3.3.1.2.5.10-6: Register **STATUS** (0x08) Status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7	6	5:4	-	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**7** : rx_post_pid_edge Flag will be set if a falling edge is detected after a valid detected pid_evt. Flag will be reset with a LIN_CONTROL.abort_rx signal. Flag can be used to detect disturbances in the response space after a valid PID byte reception before a response frame will be sent. **Note:** only applicable when LIN_CONFIG.header_processing is enabled **6** : rx_chksum_valid 0: current calculated checksum invalid 1: current calculated checksum is valid (see LIN_CHECKSUM register for details) **Note:** bit can be evaluated after reception of complete frame **5:4** : rx_header_state - LIN RX Header state 00: idle/break receiving 01: sync byte receiving 10: PID receiving **3** : tx_fifo_full - Status of tx fifo (fifo depth=1) **2** : rx_fifo_full - Status of rx fifo (fifo depth=1) **1** : TX_IDLE 0: transmit in progress 1: transmitter in idle mode **0** : RX_IDLE 0: receive in progress 1: receiver in idle mode															

Table 3.3.1.2.5.10-7: Register **ERROR** (0x0A) Error flag register

	MSB															***LSB***
Content	-	-	-	-	-	-	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	MSB															LSB
Bit Description	**9** : rx_overflow - Receiver FIFO Overflow Write 1 to clear flag related to receiver_err interrupt flag will be raised when the receiver writes to a full rx FIFO **8** : txd_timeout - TXD dominant timeout Write 1 to clear flag related to bus_err interrupt flag will be raised when the hardware detects a permanent dominant txd line (see LIN transceiver chapter for details) **7** : concurrent_brk - Concurrent Break Write 1 to clear flag related to bus_err interrupt flag will be raised when a break occurs during an ongoing transfer **Note:** only concurrent breaks during header processing will set the concurrent_brk flag **6** : bus_collision - Bus Collision Write 1 to clear flag related to bus_err interrupt flag will be raised when bus collision is detected during tx **Note:** only applicable when collision is enabled in LIN_CONFIG register **5** : pid_parity_err - PID Parity Error Write 1 to clear flag related to header_err interrupt flag will be raised when received PID has wrong parity **Note:** only applicable when header_processing is enabled in LIN_CONFIG register **4** : sync_invalid Write 1 to clear flag related to header_err interrupt flag will be raised when receiver unit is in unexpected state after sync byte measurement **Note:** only applicable when header_processing is enabled in LIN_CONFIG register **3** : sync_ov - Sync byte overflow Write 1 to clear flag related to header_err interrupt flag will be raised when sync byte counter overflows, e.g. when time between two edges is too long **Note:** LIN header processing will be stopped when overflow error occurs **2** : sync_err - Sync byte error Write 1 to clear flag related to header_err interrupt flag will be raised when sync byte plausibility check failed **Note:** only applicable when sync_validation is enabled in LIN_CONFIG register **Note:** LIN header processing will be stopped when error occurs **Note:** In error case the RX unit is potentially out of sync. All incoming data should be refused until next break/sync sequence **1** : parity_err - UART parity error Write 1 to clear flag related to receiver_err interrupt flag will be raised when a parity error is detected Error flag can be suppressed during break with UART_CONFIG.mask_brk_err **0** : frame_err - UART frame error Write 1 to clear flag related to receiver_err interrupt															

Table 3.3.1.2.5.10-8: Register **LIN_CONFIGURATION** (0x0C) LIN configuration

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:4				3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**7:4** : version 0x0 and 0x8 reserved for former products **3** : dma - DMA module implemented **2** : timer - SCI internal timer module implemented **1** : txd_timeout_reg - TXD timeout enable register implemented **0** : cbm - Concurrent break measurement implemented															

Table 3.3.1.2.5.10-9: Register **DATA** (0x0E) LIN SCI data register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8** : send_break - send break write: send break of length ('data'+ 1) x T_{Bit} **7:0** : data - sci data register Write: fill transmit fifo, only applicable when tx_fifo is empty, written byte will be transmitted Read: read received byte from rx_fifo, only applicable when fifo is full **Note:** read access clears the fifo, data can only read once															

Table 3.3.1.2.5.10-10: Register **TBIT2_LENGTH** (0x10) Length of $2T_{Bit}$

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : data - result data Result of latest Sync Byte measurement															

Table 3.3.1.2.5.10-11: Register **LIN_PID** (0x12) LIN PID

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**7:0** : data - PID data byte Latest received PID															

Table 3.3.1.2.5.10-12: Register **LIN_CHECKSUM** (0x14) LIN checksum logic

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8** : initialize 0: perform checksum calculation with chksum_val(eight bit sum with carry) 1: initialize accumulator with chksum_val value **7:0** : chksum_val write access: add new value to checksum calculation, e.g. eight bit sum with carry (see initialize flag for details) read access: current checksum value **Note:** After a valid received LIN frame the chksum_val contains the checksum over all data bits with carry incl. the checksum byte and is expected to be 0xff. It is not sufficient only to evaluate the checksum register value. An erroneous CRC byte can cause an overflow in the checksum logic and can lead to the 0xff result. Therefore it is required to read STATUS.rx_cksum_valid to check validity of the checksum result. After a valid transmitted frame in DMA mode (with LIN_CONTROL.tx_chksum and auto insertion enabled) the chksum_val contains the checksum over over all data bits and is the non-inverted version of the transmitted checksum.															

Table 3.3.1.2.5.10-13: Register **TIMER** (0x16) Timer

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7	6	5	4	3:2		1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

	MSB															LSB
Bit Description	**7** : pid_cmp_val_e - Enable PID event preloads compare register 0: no function 1: set a compare value of 0xFFFF when a valid PID event occurs(only when TIMER.enable=1) Note: This function helps to prevent header timeout compare events after a valid LIN header was received. **6** : capture_e - Enable Timer Capture Mode 1: Timer value will be captured with the next rising edge of RXD either when: a) TIMER.prepare>0 while RXD falling edge or when b) TIMER.break_restart=1 and a valid break signal was detected (timer restart case) 0: Function disabled, TIMER_CAPTURE register cleared **5** : txd_timeout_e - Enable TXD timeout counter $t_{LIN,TXD,DOM}$ timeout counter: In case of a dominant TXD signal for a time greater than $t_{LIN,TXD,DOM}$ the TXD output will be forced to 1. See txd_timeout ERROR flag above. **4** : break_restart - Break Restart 0: no function 1: restart timer when LIN break signal is detected -> preload timer with 11 x T_{Bit} x 16 = 176 (break time when LIN_CONFIG.break_thd==1) -> preload timer with 9.5 x T_{Bit} x 16 = 152 (break time when LIN_CONFIG.break_thd==0) -> preload compare register with maximum header length => 48 x T_{Bit} x 16 = 768 **Note:** A disabled timer will be started when a LIN break signal is detected **Note:** Only applicable when clk_base=baud_rate and expected baud rate nearly actual baud rate (e.g. within LIN Specification Tolerance F_{TOL_UNSYNC}) **3:2** : prepare - Timer Prepare 0: normal operation 1: restart timer from 0 when a falling RXD edge is detected. timer_prepare bit will be reset to 0 immediately 2: restart timer from 0 when a falling RXD edge is detected timer_prepare bit will be reset reset to 0 when a valid break is detected 3: restart timer from 0 when a falling RXD edge is detected timer_prepare bit will not be reset automatically **Note:** As long as timer_prepare is >0 no compare events will be generated, this allows preloading of the timer compare register.timer_prepare works only with timer_enable=1 **1** : clk_src - timer_clk_base Timer counts with 0: 16 x baud rate 1: 1µs clock **Note:** The period of the 1µs clock depends on the correct setting of the OSC_CTRL.FREQ register (-). **0** : enable - Timer Enable 0: timer not running(reset counter to 0) 1: timer running timer counter is incremented by timer_clk_base **Note:** A disabled timer can be auto enabled when a valid break signal is detected and the 'break_restart' bit is set															

Table 3.3.1.2.5.10-14: Register **TIMER_COUNTER** (0x18) Timer Counter

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : timer_counter - Timer Counter Timer counter is running when timer is enabled with timer_enable. sci_timer_ov flag is set when timer overflows.															

Table 3.3.1.2.5.10-15: Register **TIMER_COMPARE** (0x1A) Timer Compare

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : timer_compare timer_cmp flag is set when timer reaches timer_compare value **Note:** Timer compare flag will not be set when timer_compare=0 **Note:** Timer compare flag will NOT be set as long as timer_prepare>1. Timer overflow events will be generated															

Table 3.3.1.2.5.10-16: Register **TIMER_CAPTURE** (0x1C) Timer Capture

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : timer_capture Can be used to measure the break length. See TIMER.capture_e for details.															

Table 3.3.1.2.5.10-17: Register **DMA_TX_ADDRESS** (0x20) Transmit DMA Address

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : dma_tx_addr Start address of source of transmit data. **Note:** Address is incremented with each DMA transfer executed.															

Table 3.3.1.2.5.10-18: Register **DMA_TX_ADDRESS_HIGH** (0x22) Transmit DMA Address High Word[1)]

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : dma_tx_addr High Word of 32 bit address. Start address of source of transmit data. **Note:** Address is incremented with each DMA transfer executed.															

[1)] not used (only 16 bit address widht)

Table 3.3.1.2.5.10-19: Register **DMA_TX_LENGTH** (0x24) Transmit DMA Length

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dma_tx_length Length of data to be transmitted via DMA in BYTE. Value will be decremented with each transfer. Write access to register will stop current DMA operation and will restart DMA controller.															

Table 3.3.1.2.5.10-20: Register **DMA_RX_ADDRESS** (0x28) Receive DMA Address

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : dma_rx_addr Start address of destination of receive data. Note: Address is incremented with each DMA transfer executed.															

Table 3.3.1.2.5.10-21: Register **DMA_RX_ADDRESS_HIGH** (0x2A) Receive DMA Address High Word[1)]

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : dma_rx_addr High Word of 32 bit address. Start address of destination of receive data. Note: Address is incremented with each DMA transfer executed.															

[1)] not used (only 16 bit address widht)

Table 3.3.1.2.5.10-22: Register **DMA_RX_LENGTH** (0x2C) Receive DMA Length

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dma_rx_length Length of data to be transmitted via DMA in BYTE. Value will be decremented with each transfer. Write access to register will stop current DMA operation and will restart DMA controller.															

Table 3.3.1.2.5.10-23: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : tick_1ms (event) - generates an interrupt each 1ms **14** : tx_finish_evt (event) - transmit transfer finished, transmitter returned to idle state **13** : tx_dma_finished (event) - transmit dma transfer finished **12** : tx_fifo_empty (level) - transmit fifo empty **11** : rx_dma_finished (event) - receive dma transfer finished **10** : rx_fifo_full (level) - receive fifo full **9** : pid_evt (event) - valid PID received **8** : sync_evt (event) - SYNC byte received **7** : break_evt (event) - valid BREAK detected **6** : header_err (level) - header error occurred, see ERROR register for details **5** : receiver_err (level) - receiver error occurred, see ERROR register for details **4** : bus_err (level) - bus error occurred, see ERROR register for details **3** : sci_timer_ov (event) - SCI timer overflow **2** : sci_timer_cmp (event) - SCI timer compare **1** : rxd_rising (event) - rising edge of RXD signal **0** : rxd_falling (event) - falling edge of RXD signal															

Table 3.3.1.2.5.10-24: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.10-25: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.10-26: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.10-27: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.10-28: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no IRQ is pending the first unused IEQ number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.11 GPIO Module (GPIO)
This module gives access to general purpose digital IOs.

Features
- 8 IOs (ports)
- Interrupt capable
  - Positive IO signal edge interrupt
  - Negative IO signal edge interrupt

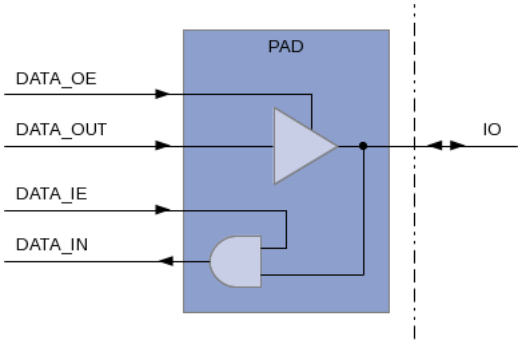


Figure 3.3.1.2.5.11-1: Structure

Table 3.3.1.2.5.11-1: Registers

Register Name	*Address*	*Description*
DATA_OUT	0x00	data out register
DATA_OE	0x02	output enable register
DATA_IN	0x04	data in register
DATA_IE	0x06	input enable register
DIRECTION_LOCK	0x08	direction lock register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.11-2: Register **DATA_OUT** (0x00) data out register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : data - output data															

Table 3.3.1.2.5.11-3: Register **DATA_OE** (0x02) output enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : enable - 0: input 1: output															

Table 3.3.1.2.5.11-4: Register **DATA_IN** (0x04) data in register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**7:0** : data - input data															

Table 3.3.1.2.5.11-5: Register **DATA_IE** (0x06) input enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : enable - 0: input path is disabled 1: input path is enabled															

Table 3.3.1.2.5.11-6: Register **DIRECTION_LOCK** (0x08) direction lock register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : lock - lock corresponding bits of OUTPUT_ENABLE and INPUT_ENABLE 0: unlocked 1: locked Note: once set, bits cannot be cleared again.															

Table 3.3.1.2.5.11-7: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : evt_neg_7 (event) - negative edge event at IO port bit 7 **14** : evt_pos_7 (event) - positive edge event at IO port bit 7 **13** : evt_neg_6 (event) - negative edge event at IO port bit 6 **12** : evt_pos_6 (event) - positive edge event at IO port bit 6 **11** : evt_neg_5 (event) - negative edge event at IO port bit 5 **10** : evt_pos_5 (event) - positive edge event at IO port bit 5 **9** : evt_neg_4 (event) - negative edge event at IO port bit 4 **8** : evt_pos_4 (event) - positive edge event at IO port bit 4 **7** : evt_neg_3 (event) - negative edge event at IO port bit 3 **6** : evt_pos_3 (event) - positive edge event at IO port bit 3 **5** : evt_neg_2 (event) - negative edge event at IO port bit 2 **4** : evt_pos_2 (event) - positive edge event at IO port bit 2 **3** : evt_neg_1 (event) - negative edge event at IO port bit 1 **2** : evt_pos_1 (event) - positive edge event at IO port bit 1 **1** : evt_neg_0 (event) - negative edge event at IO port bit 0 **0** : evt_pos_0 (event) - positive edge event at IO port bit 0															

Table 3.3.1.2.5.11-8: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : mask - enable IRQ source 1: enabled 0: disabled															

Table 3.3.1.2.5.11-9: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.11-10: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.11-11: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.11-12: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**4:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no IRQ is pending the first unused IRQ number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.12 Capture Compare Timer Module (CCTIMER)
This module includes two 16 bit timers which can also be used to measure incoming signal waveforms.

Features
- 8 bit counter clock pre-scaler
- 2 x 16 bit timer / measurement / event counter
  - Waveform measurement : counter A measures period, counter B measures signal high time
  - Waveform measurement : counter A measures signal low time, counter B measures signal high time
  - Period meas / timer : counter A measures period, counter B is used as timer
  - Timer : both counters are used as timers
  - PWM generate : counter A defines period, compare value B defines pulse width
  - Event counting : counter A defines period, counter B counts measurement signal events
- Measurement counter "start" and "stop" signal edges can be configured
- Timer "once" and "loop" modes are possible
- Counter clock source selection
- Up to 256*2^16 clock cycle measurement / timer duration
- Power saving pre-scaled architecture
- PWM generator
- Event counting + threshold IRQ
- Windowed event counting

Note: For the various measurements, please take note of the influence of the system oscillator settings (OSC_CTRL.OSC_CONFIG 3.3.1.2.5.13-4).

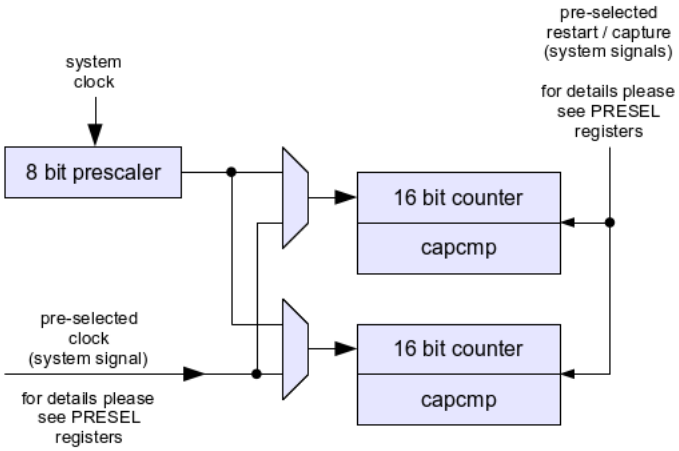


Figure 3.3.1.2.5.12-1: Structure

Table 3.3.1.2.5.12-1: Registers

Register Name	*Address*	*Description*
PRESCALER	0x00	pre-scaler register
CONTROL	0x02	control register
CONFIG_A	0x04	config A register
CONFIG_B	0x06	config B register
CAPCMP_A	0x08	capture compare A register
CAPCMP_B	0x0A	capture compare B register
COUNTER_A	0x0C	counter A register
COUNTER_B	0x0E	counter B register
PRESEL_A	0x10	pre_select A register
PRESEL_B	0x12	pre_select B register
IRQ_STATUS	0x30	IRQ status register
IRQ_MASK	0x34	IRQ mask register
IRQ_VENABLE	0x38	IRQ vector enable register
IRQ_VDISABLE	0x3A	IRQ vector disable register
IRQ_VMAX	0x3C	IRQ max vector register
IRQ_VNO	0x3E	IRQ vector number register

Table 3.3.1.2.5.12-2: Register **PRESCALER** (0x00) pre-scaler register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : val - counter clock pre-scale value (clock prediv by val+1)															

Table 3.3.1.2.5.12-3: Register **CONTROL** (0x02) control register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	W	W	W	R/W	R/W
Bit Description	**4** : restart_p - restart pre-scale counter **3** : restart_b - restart counter B **2** : restart_a - restart counter A **1** : enable_b - enable sub-timer B **0** : enable_a - enable sub-timer A															

Table 3.3.1.2.5.12-4: Register **CONFIG_A** (0x04) config A register

	MSB															***LSB***
Content	-	-	-	-	-	-	9:8		-	6:5		4:2			1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**9:8** : mode - mode selection 0: compare mode (CAPCMP is used as compare value) 1: like 0, but "enable" will be cleared at compare event (once) 2: like 0, but counter will be restarted at compare event (loop) 3: capture mode (CAPCMP is used to capture) **6:5** : capture_sel - capture event selection (capture mode only !) 0 : other counter compare event 1 : MEAS signal positive edge 2 : MEAS signal negative edge 3 : system signal (see pre-select PRESEL_*.capture_sel) **4:2** : restart_sel - restart event selection 0: counter enable gets active / own compare event (loop) 1: other counter compare event 2: MEAS signal positive edge 3: MEAS signal negative edge 4: system signal (see pre-select PRESEL_*.restart_sel) **1:0** : clk_sel - counter clock source selection 0: pre-scaled system clock 1: MEAS signal positive edge (up) 2: MEAS signal negative edge (up) 3: system signal (see pre-select PRESEL_*.clk_sel)															

Table 3.3.1.2.5.12-5: Register **CONFIG_B** (0x06) config B register

	MSB															***LSB***
Content	-	-	-	-	-	-	9:8		-	6:5		4:2			1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**9:8** : mode - mode selection 0: compare mode (CAPCMP is used as compare value) 1: like 0, but "enable" will be cleared at compare event (once) 2: like 0, but counter will be restarted at compare event (loop) 3: capture mode (CAPCMP is used to capture) **6:5** : capture_sel - capture event selection (capture mode only !) 0 : other counter compare event 1 : MEAS signal positive edge 2 : MEAS signal negative edge 3 : system signal (see pre-select PRESEL_*.capture_sel) **4:2** : restart_sel - restart event selection 0: counter enable gets active / own compare event (loop) 1: other counter compare event 2: MEAS signal positive edge 3: MEAS signal negative edge 4: system signal (see pre-select PRESEL_*.restart_sel) **1:0** : clk_sel - counter clock source selection 0: pre-scaled system clock 1: MEAS signal positive edge (up) 2: MEAS signal negative edge (up) 3: system signal (see pre-select PRESEL_*.clk_sel)															

Table 3.3.1.2.5.12-6: Register **CAPCMP_A** (0x08) capture compare A register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - compare value OR captured value															

Table 3.3.1.2.5.12-7: Register **CAPCMP_B** (0x0A) capture compare B register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : val - compare value OR captured value															

Table 3.3.1.2.5.12-8: Register **COUNTER_A** (0x0C) counter A register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : val - current counter value															

Table 3.3.1.2.5.12-9: Register **COUNTER_B** (0x0E) counter B register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : val - current counter value															

Table 3.3.1.2.5.12-10: Register **PRESEL_A** (0x10) pre_select A register

	MSB															***LSB***
Content	-	-	-	-	11:8				7:4				3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:8** : capture_sel - pre-select a system signal as capture trigger 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge **7:4** : restart_sel - pre-select a system signal as restart trigger 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge **3:0** : clk_sel - pre-select a system signal as timer clock 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge																

Table 3.3.1.2.5.12-11: Register **PRESEL_B** (0x12) pre_select B register

	MSB															***LSB***
Content	-	-	-	-	11:8				7:4				3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:8** : capture_sel - pre-select a system signal as capture trigger 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge **7:4** : restart_sel - pre-select a system signal as restart trigger 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge **3:0** : clk_sel - pre-select a system signal as timer clock 0: HALL decoder 60° trigger pulses 1: increment decoder step pulses 2: increment decoder zero marker 3: SCI baud clock (up) 8: SYNC U posedge 9: SYNC V posedge 10: SYNC W posedge 11: SYNC U negedge 12: SYNC V negedge 13: SYNC W negedge																

Table 3.3.1.2.5.12-12: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8** : compare_b (event) - counter 1 compare event **7** : compare_a (event) - counter 0 compare event **6** : capture_b (event) - counter 1 capture event **5** : capture_a (event) - counter 0 capture event **4** : restart_b (event) - counter 1 restart event **3** : restart_a (event) - counter 0 restart event **2** : overflow_b (event) - counter 1 overflow (counter type is signed int) **1** : overflow_a (event) - counter 0 overflow (counter type is signed int) **0** : restart_p (event) - pre-scaler restart event															

Table 3.3.1.2.5.12-13: Register **IRQ_MASK** (0x34) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.12-14: Register **IRQ_VENABLE** (0x38) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.12-15: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.12-16: Register **IRQ_VMAX** (0x3C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.12-17: Register **IRQ_VNO** (0x3E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no IRQ is pending the first unused IRQ number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.13 Oscillator Control Module (OSC_CTRL)

Features

- system oscillator frequency selection (48MHz or 43MHz)
- fine adjustment of system oscillator frequency
- activation of spread spectrum
- measurement of average system oscillator frequency

Set System Oscillator Frequency

To switch the system oscillator frequency from 48MHz to 43MHz OSC_CONFIG.osc_freq_sel must be set to '1'. Additional it is strictly recommended to store the new frequency value directly into the average system oscillator frequency register (FREQ), because this register value is used for internal timing. The setting of the register FREQ must be done by software.

Example sequence:
// switch system oscillator frequency to 43MHz
REG_OSC_CTRL_OSC_CONFIG.osc_freq_sel = 1;
// store 43MHz multiplied by 256 in FREQ register
REG_OSC_CTRL_FREQ = 0x2B00;

Note:: Nothing to do when using 48MHz by default.

Fine Adjustment of System Oscillator Frequency

See description in CONTROL register.

System Oscillator Frequency Measurement

The average system oscillator frequency can measure by counting the system oscillator cycles during 256 watchdog oscillator cycles. This needs typ. 320µs when F_{OSC_WDOG} = 0.8MHz. Then the system oscillator frequency is to calculate by using the internal DIVIDER module.

After this, the divider result must store into the average system oscillator frequency register (FREQ). This is important for the LIN SCI Timer, when 1µs clock base time is used.

The calculation formula is:

FREQ = 256*FREQ_COUNT/WD_TIMING

Following example sequence starts a system oscillator measurement and store the result into the register FREQ.

```
// start frequency measurement
REG_OSC_CTRL_FREQ_MEAS_CTRL = 1;
// poll until end of measurement
while (REG_OSC_CTRL_FREQ_MEAS_CTRL != 2) {}
// calculate frequency (256*FREQ_COUNT/WD_TIMING)
// by using internal DIVIDER module
REG_DIVIDER_OP1LO = REG_OSC_CTRL_FREQ_COUNT << 8;
REG_DIVIDER_OP1HI = REG_OSC_CTRL_FREQ_COUNT >> 8;
REG_DIVIDER_OP2 = REG_OSC_CTRL_WD_TIMING;
// store calculated frequency into FREQ register
REG_OSC_CTRL_FREQ = REG_DIVIDER_RESULTLO;
```

Hint: polling is also possible with:
`while (REG_OSC_CTRL_FREQ_COUNT == 0) {}`

Hint: The measurement time in µs can be see directly in register WD_TIMING.

Spread Spectrum

The Spread Spectrum feature can be use to reduce electromagnetic radiation.

It can be activated by setting OSC_CONFIG = 0x2. This enables spread spectrum with the maximal system oscillator frequency of 48MHz.
Activating of spread spectrum is only recommended when the maximal system oscillator frequency is set to 48MHz (OSC_TRIM1 selected).

The resulting average system oscillator frequency is lower then 48MHz when spread spectrum is active (~45.5 MHz).

A system oscillator frequency measurement is recommended when spread spectrum is activated.

Example sequence:

```
// enable Spread Spectrum @48MHz
REG_OSC_CTRL_OSC_CONFIG = 2;
// frequency measurement
System_Oscillator_Frequency_Measurement();
```

Figure below shows a spread spectrum frequency example with a triangle function.

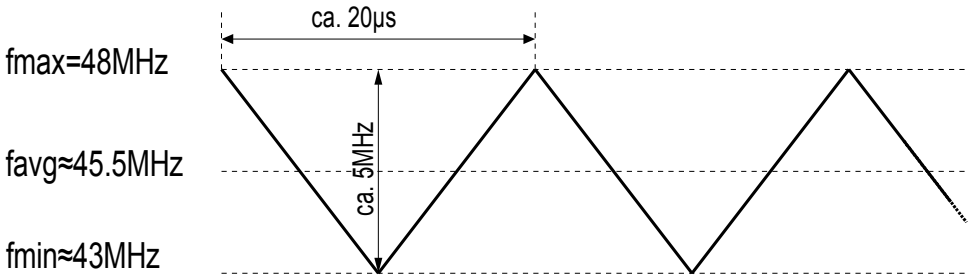


Figure 3.3.1.2.5.13-1: Spread Spectrum Example (Triangle)

Table 3.3.1.2.5.13-1: Registers

Register Name	***Address***	***Description***
CONTROL	0x00	control register for system oscillator adjustment
SYS_CLK_CONFIG	0x02	system clock configuration register
OSC_CONFIG	0x08	system oscillator configuration register
WD_TIMING	0x12	watchdog oscillator timing register
FREQ_COUNT	0x14	counter for system oscillator frequency calculation
FREQ	0x16	average system oscillator frequency register
FREQ_MEAS_CTRL	0x18	system oscillator frequency measurement control register

Table 3.3.1.2.5.13-2: Register **CONTROL** (0x00) control register for system oscillator adjustment

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5:0					
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5:0** : sys_osc_adjust system RC oscillator trim value offset This value is always subtracted from the internal oscillator trim value and only allows to lower the system oscillator frequency. One adjust LSB changes the system oscillator clock by approx. -0.225% of its non-adjusted value.															

Table 3.3.1.2.5.13-3: Register **SYS_CLK_CONFIG** (0x02) system clock configuration register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : adc_clk_phase selects an ADC clock phase signal to be sampled by non delayed sys_clk (sys_clk_osc) to generate ADC clock signal 0 : use FSM generated signal as ADC clock (non shift variant) 1 : use shifted FSM signal to be sampled **4:0** : sys_clk_delay ADC clock to system clock delay 0 .. 21 analog delays (see analog parameters for details)															

Table 3.3.1.2.5.13-4: Register **OSC_CONFIG** (0x08) system oscillator configuration register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : spread_enable spread spectrum enable 0 : spread spectrum disabled (default) 1 : spread spectrum enabled **Note:** When spread spectrum is activated the average oscillator frequency is lower then the selected oscillator frequency. The average oscillator frequency can be measured. **0** : osc_freq_sel select oscillator frequency 0 : f_sys_osc=48MHz (default) 1 : f_sys_osc=43MHz															

Table 3.3.1.2.5.13-5: Register **WD_TIMING** (0x12) watchdog oscillator timing register

	MSB															***LSB***
Content	-	-	-	-	-	-	9:0									
Reset value	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**9:0** : wdt_256 time in µs for 256 watchdog oscillator cycles (used for oscillator frequency calculation) **Note:** register is only one time writable during startup															

Table 3.3.1.2.5.13-6: Register **FREQ_COUNT** (0x14) counter for system oscillator frequency calculation

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : freq_count number of system oscillator cycles for 256 watchdog oscillator cycles During a running frequency measurement the value is masked to 0x0000. (used for oscillator frequency calculation)															

Table 3.3.1.2.5.13-7: Register **FREQ** (0x16) average system oscillator frequency register

	MSB															***LSB***
Content			13:0													
Reset value	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**13:0** : freq_256 average oscillator frequency in MHz multiplied by 256 The reset value is 256*48 and is trimmed for 48MHz.															

Table 3.3.1.2.5.13-8: Register **FREQ_MEAS_CTRL** (0x18) system oscillator frequency measurement control register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/S
Bit Description	**1** : freq_meas_status frequency measurement status 0 : no measurement result available in register FREQ_COUNT (no measurement done or measurement is still running) 1 : measurement finished; result is available in register FREQ_COUNT **Note:** Bit is read only. **0** : start_freq_meas start measurement of oscillator frequency Start measurement by setting this bit to '1'. Bit is reset automatically when frequency measurement cycle is finished. (Bit = '1' shows a running measurement.) **Note:** Bit can only set. Reset is automatically when measurement is finished.															

3.3.1.2.5.14 ADC Control Module (ADC_CTRL)

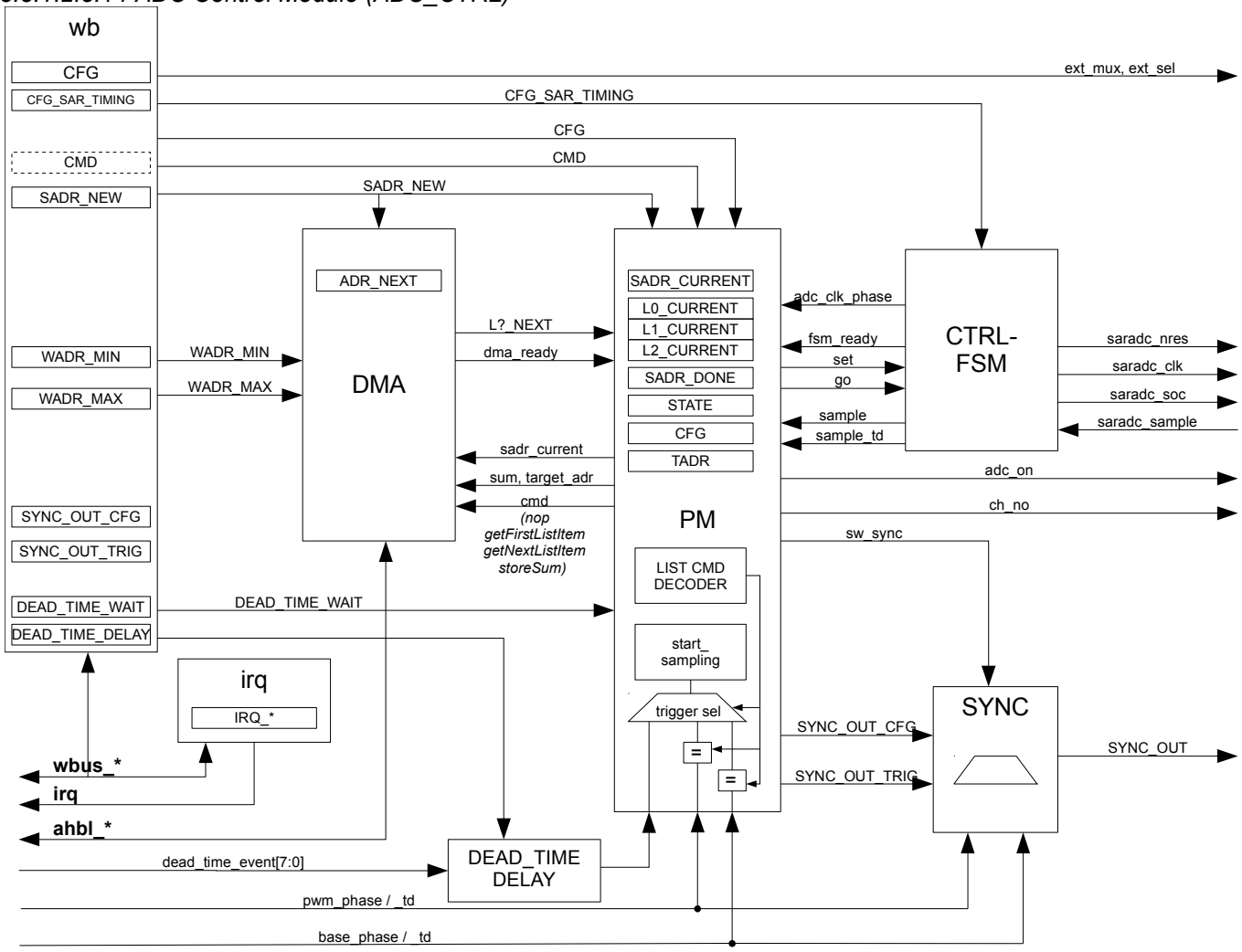


Figure 3.3.1.2.5.14-1: SAR ADC Control Operating Environment

The saradc_ctrl module controls the SAR ADC and its associated analog input multiplexer. The mapping of multiplexer channels to the channel number ch_no and the configuration used in the module is as follows:

Table 3.3.1.2.5.14-1: ADC Channels

ch_no	*ADC channel*	*comment*
3..0	• when CFG.mux_sel = 0 IO port A[3:0] • when CFG.mux_sel = 1 IO port A[3:0] selections will be translated to MUX control values. CFG.ext_sel selects the analog input IO pa[0] select -> MUX1 = 0, MUX0 = 0 pa[1] select -> MUX1 = 0, MUX0 = 1 pa[2] select -> MUX1 = 1, MUX0 = 0 pa[3] select -> MUX1 = 1, MUX0 = 1	gpio pins of die
7..4	IO port A[7:4]	gpio pins of die

ch_no	*ADC channel*	*comment*
15..8	IO port B[7:0]	gpio pins of die
23..16	IO port C[7:0]	gpio pins of die
24	temperature	system monitoring
25..27	ADC input undefined	no channel selected
28	select channel set in register AUTOCHANNEL.auto_ch_u	see register below
29	select channel set in register AUTOCHANNEL.auto_ch_v	see register below
30	select channel set in register AUTOCHANNEL.auto_ch_w	see register below
31	keep previous channel	

The measurements can be synchronized either to the commutation phase "base_phase" of the prePWM module or to the pwm signals of the PWMN module. For the pwm signal either the phase of the pwm period "pwm_phase" or the start of the dead times signalled by "dead_time_evt" is used for synchronization.

Functional Description
The ADC is normally used to measure at least three to four sources per PWM period:

- motor voltage for back emf measurements, synchronized to PWM period
- motor center point voltage This voltage must be measured just before and just after a switching event of a specific phase. Subsequent PWM periods use different switching patterns and thus measurement parameters. Up to 8 triggers per PWM period are required.
- leg sum current measurement, synchronized to PWM period This must be updated continuously for overload protection and torque control
- multiplexed general purpose measurements, not synchronized These values are to many to be converted within a single PWM period. Thus a scheduling is necessary.

Thus a flexible ADC scheduling scheme is required which allows to set the configuration for each ADC measurement independently. To keep the CPU load acceptable multiple measurements have to be configured in advance. To keep the gate count for the required number of configurations at an acceptable level the measurement configuration and the resulting samples reside in global memory.

The sequence and type of ADC measurements is controlled by a list of measurement configurations in memory. The start address "sadr" of this list is supplied by software through the register SADR_NEW. Once a SADR_NEW becomes available the processing of the list starts.

Within a list every list item (measurement configuration) specifies the trigger of the measurement, the ADC channel, the number of samples to sum and the number of sums to write to memory. Each sum is written atomic as a 16 bit word, so evaluation of partially completed lists is feasible. The next list item is already pre-fetched while the previous is executing, making jitter free reaction to triggers feasible.

Execution of a list stops when the END command has been reached or when a corresponding command is written to the command (CMD) register.

Format of the list for N+1 entries:

Table 3.3.1.2.5.14-2: Measurement Configuration

SADR+	bits	name	description
0x00 + N*0x6	15:0	target_adr	0x*0: Use target_adr. Memory[target_adr(15:0)]=Sum; 0x*01:Set and use TADR. (adr will only be set if no_sum > 0) TADR = target_adr(15:2) & 00 Memory[TADR]=Sum; TADR=TADR+2 0x*11:Use TADR. Memory[TADR]=Sum; TADR=TADR+2
0x02 + N*0x6	3:0	no_sample	0..15 number of samples sum up no_sample+1 samples, i.e. 1..16 samples can be summed
0x02 + N*0x6	8:4	no_sum	number of sums 0: do not perform any sampling, only set multiplexer when trigger occurs 1..31 number of sums to write to memory
0x02 + N*0x6	13:9	ch_no	0..31 channel number to set if (CFG_SAR_TIMING.mux_phase=0) before conversion of the this sample sequence f (CFG_SAR_TIMING.mux_phase>4) after conversion of the last sample of this sample sequence for the next sample sequence
0x02 + N*0x6	14	trigger_type_ext	Trigger type extension see trigger for details
0x02 + N*0x6	15	trigger_type	see trigger
0x04 + N*0x6	15:0	trigger	Sample Sequence Starts - for trigger type=0 and trigger_type_ext=0: when pwm_phase matches trigger
0x04 + N*0x6	15:0	trigger	- for trigger type=0 and trigger_type_ext=1: when 0x<ext>00 immediately -> The ADC starts sampling immediately. The min-imal sampling phase is extended by ext number of adc clock/2 cycles.

SADR+	bits	name	description
0x04 + N*0x6	15:0	trigger	- for trigger type=1 and trigger is • 0x0000..0x03FF: when base_phase matches trigger • 0x0400..04FF: when any of the dead_time_evt[7:0] occurs and the corresponding bit [7:0] is set and DEAD_TIME_WAIT0 clock cycles have passed after the dead time event • 0x1400..14FF: when any of the dead_time_evt[7:0] occurs and the corresponding bit [7:0] is set and DEAD_TIME_WAIT1 clock cycles have passed after the dead time event • 0x2400..24FF: when any of the dead_time_evt[7:0] occurs and the corresponding bit [7:0] is set and DEAD_TIME_WAIT2 clock cycles have passed after the dead time event • 0x0500..057F: when edge counter >= corresponding bits[2:0] and edge counter <= corresponding bits[6:4] and DEAD_TIME_WAIT0 clock cycles have passed after the dead time event, when bit[3]=1 set autochannel (see register AUTOCHANNEL) • 0x1500..157F: when edge counter >= corresponding bits[2:0] and edge counter <= corresponding bits[6:4] and DEAD_TIME_WAIT1 clock cycles have passed after the dead time event, when bit[3]=1 set autochannel (see register AUTOCHANNEL) • 0x2500..257F: when edge counter >= corresponding bits[2:0] and edge counter <= corresponding bits[6:4] and DEAD_TIME_WAIT2 clock cycles have passed after the dead time event, when bit[3]=1 set autochannel (see register AUTOCHANNEL) • 0xFFC0: immediate sampling The minimal sampling phase is extended by CFG_SAR_TIMING.mux_sampling_extension number of adc clock/2 cycles.

SADR+	bits	name	description
0x04 + N*0x6	15:0	trigger	- for trigger type=1 and trigger is Special functions (do not wait for a trigger, not use ADC nor modify mux): • 0xFFE0 "CMP" Assert "out of range" irq IRQ_STATUS.oor0 when the following condition is violated: 0x02[14:0] <= last_sum <= target_adr[15:0]. • 0xFFE1 " " IRQ_STATUS.oor1 " " • 0xFFE2 " " IRQ_STATUS.oor2 " " • 0xFFE3 " " IRQ_STATUS.oor3 " " • 0xFFE4 " " IRQ_STATUS.oor4 " " • 0xFFE5 " " IRQ_STATUS.oor5 " " • 0xFFE6 " " IRQ_STATUS.oor6 " " • 0xFFE7 " " IRQ_STATUS.oor7 " " • 0xFFE8 " " IRQ_STATUS.oor8 " " • 0xFFE9 " " IRQ_STATUS.oor9 " " • 0xFFF0 "WRITE_BUS" Store target_adr to 0x02[14:0] with AdrBit[15]=0 • 0xFFF1 "CFG_WRITE" Store target_adr to CFG register • 0xFFF2 "SYNC_OUT" trigger sync out event (see SYNC_OUT_CFG) • 0xFFF3 "WAIT" wait for target_adr+2 clk cycles • 0xFFF4 "ST_BP" store base_phase in place of sum • 0xFFF5 "ST_TADR" store TADR in place of sum • 0xFFF6 LOOP UNTIL (0x02[14:0] <=base_phase<= target_adr[15:0]) • 0xFFF7 LOOP UNTIL NOT(0x02[14:0] <=base_phase<= target_adr[15:0]) • 0xFFF8 LOOP UNTIL (0x02[14:0] <=TADR<= target_adr[15:0]) • 0xFFF9 LOOP UNTIL NOT(0x02[14:0] <=TADR<= target_adr[15:0]) • 0xFFFA "GOTO" target_adr when last_sum<=0x02[14:0] • 0xFFFB "GOTO" target_adr when not last_sum<=0x02[14:0] • 0xFFFC "GOTO" target_adr when base_phase<=0x02[14:0] • 0xFFFD "GOTO" target_adr when not base_phase<=0x02[14:0] • 0xFFFE "GOTO" target_adr when TADR[14:0] = 0x02[14:0] if 0x02[14:0] = 0 ¿ GOTO always • 0xFFFF "END" list completed (last list item)

Further functionality of saradc_ctrl

- Synchronisation output The module output sync_out can be used to synchronize external peripherals. Sync_out is asserted for SYNC_OUT_CFG.length clock cycles when the source specified in SYNC_OUT_CFG.src matches SYNC_OUT_TRIGGER.
- Debug outputs Mux_sel and a signal indicating the operation of the S/H stage can be multiplexed to digital outputs to facilitate software debugging.
- Edge Counter Within a PWM period the occurrence of the Dead Time Trigger signals are counted. Note: only the dead time trigger signals of UVW are considered (dead_time_trigger[5:0]). The edge counter counts from 0 to 6 each pwm period and will be reset to zero when PWM_PHASE == 0. ADC Sampling can be triggered after a specific amount of a dead time edges.

Table 3.3.1.2.5.14-3: Registers

Register Name	Address	Description
CFG	0x00	Configuration register. CFG can also be set/reset by special list command.
CFG_SAR_TIMING	0x02	Timing register
DEAD_TIME_DELAY	0x04	Common Dead Time Delay
DEAD_TIME_WAIT0	0x06	Dead time wait register
SYNC_OUT_CFG	0x08	sync out config register
SYNC_OUT_TRIGGER	0x0A	sync out trigger register
CMD	0x0C	command register
WADR_MIN	0x0E	Memory write protection register (min)
WADR_MAX	0x10	Memory write protection register (max)
SADR_NEW	0x12	New List start address register
SADR_CURRENT	0x14	Start address of current list
L0_CURRENT	0x16	L0 current register
L1_CURRENT	0x18	L1 current register
L2_CURRENT	0x1A	L2 current register
ADR_NEXT	0x1C	next address register
SADR_DONE	0x1E	start address complete
STATE	0x20	state register
DEAD_TIME_WAIT1	0x22	Dead time wait register
DEAD_TIME_WAIT2	0x24	Dead time wait register
TADR	0x26	TADR content register
AUTOCHANNEL	0x28	AUTO Channel register
IRQ_STATUS	0x70	IRQ status register
IRQ_MASK	0x74	IRQ mask register
IRQ_VENABLE	0x78	IRQ vector enable register
IRQ_VDISABLE	0x7A	IRQ vector disable register
IRQ_VMAX	0x7C	IRQ max vector register
IRQ_VNO	0x7E	IRQ vector number register

Table 3.3.1.2.5.14-4: Register **CFG** (0x00) Configuration register.CFG can also be set/reset by special list command.

	MSB															***LSB***
Content	-	-	-	-	-	10	9	8:4					3	2:1		0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10** : store_dt_evt 0: function disabled 1: when trigger_type=1 and trigger = 0x400 .. 0x250F store trigger causing dead_time_event to bits[15:12] of the ADC sampled value [12]-> U channel dead time event causes the trigger [13]-> V channel dead time event causes the trigger [14]-> W channel dead time event causes the trigger [15]-> X channel dead time event causes the trigger Note: only one of these 4 bits can be set at the same time, if two or more trigger events occur at the same time the highest bit will be set. Note: only applicable when 'no_sample' = 0 **9** : Enable missed dead time trigger logic 0: logic disabled 1: When during processing of dead time trigger a dead time trigger of the next command occurs the next command will be skipped and 0xffff will be written to the memory (only when no_sum > 0) Note: When no_sum > 1 0xffff will only be written once. A write error IRQ will be generated **8:4** : ext_sel selects the IO channel to use as ext muxed input used if mux_sel is 1 and ch_no is in the range 0..3. **3** : 1: external multiplexer is used. See multiplexer table. **2:1** : 0: adc_reset after every sample sequence 1: adc reset permanently not asserted (conversion possible)(recommended) 3: adc reset permanently asserted (conversion not possible) **0** : 0: adc is powered down 1: adc is powered up															

Table 3.3.1.2.5.14-5: Register **CFG_SAR_TIMING** (0x02) Timing register

	MSB															***LSB***
Content	15:8								7:3					2:0		
Reset value	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : sampling_extension 0..255 sampling period is extended by sampling_extension adc clk/2 cycles. **Note:** only used for immediate sampling **7:3** : mux_phase Clock edge number when the multiplexer is changed. 0 switches the muliplexer for current sampling command 5..28 switches the muliplexer for next sampling command **Note:** Values 1..4 must not be used **Note:** Clock edges counting from 1 to 28, sampling period ends with edge 5 (rising edge), Mux switching needs 3 sys_clk cycles including break before make **2:0** : adc_clk_div 1..MAX Period of adc clk is 2*adc_clk_div clk cycles. **Note:** Adc clk must not exceed 16MHz, e.g. with system clk = 48MHz set adc_clk_div=2 -> 12 MHz ADC clk															

Table 3.3.1.2.5.14-6: Register **DEAD_TIME_DELAY** (0x04) Common Dead Time Delay

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dead_time_delay Common delay of all dead time triggers in sys_clk cycles. Delay in sys_clk cycles = dead_time_delay + 1															

Table 3.3.1.2.5.14-7: Register **DEAD_TIME_WAIT0** (0x06) Dead time wait register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dead_time_wait (configuration 0) number of clock cycles between a dead time event and the sampling															

Table 3.3.1.2.5.14-8: Register **SYNC_OUT_CFG** (0x08) sync out config register

	MSB															***LSB***
Content						10:9		8	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:9** : src - 0: base_phase 1: pwm_phase 2: list command (see above) 3: disabled **8** : pol - polarity of sync_out when asserted **7:0** : length - length of the sync_out pulse in clock cycles															

Table 3.3.1.2.5.14-9: Register **SYNC_OUT_TRIGGER** (0x0A) sync out trigger register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : sync_out_trigger - sync_out is asserted for SYNC_OUT_CFG.length clock cycles when the source specified in SYNC_OUT_CFG.src matches SYNC_OUT_TRIGGER Note: SYNC_OUT signal has to be muxed to IO															

Table 3.3.1.2.5.14-10: Register **CMD** (0x0C) command register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W
Bit Description	**1** : seq_skip 1: terminate (skip) current list item (sample sequence or waiting for trigger) and continue with the next. **0** : list_skip 1: terminate (skip) current list of sample sequences and continue with the next Note: A skip list command will not feed sadr_done register and will not set IRQ_STATUS.sadr_done_nempty and list_done_evt															

Table 3.3.1.2.5.14-11: Register **WADR_MIN** (0x0E) Memory write protection register (min)

	MSB															***LSB***
Content	15:5											-	-	-	-	-
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R
Bit Description	**15:5** : wadr_min Memory write protection. The module only performs writes when WADR_MIN <= target_address < WADR_MAX															

Table 3.3.1.2.5.14-12: Register **WADR_MAX** (0x10) Memory write protection register (max)

	MSB															***LSB***
Content	15:5											-	-	-	-	-
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R
Bit Description	**15:5** : wadr_max Memory write protection. The module only performs writes when WADR_MIN <= target_address < WADR_MAX															

Table 3.3.1.2.5.14-13: Register **SADR_NEW** (0x12) New List start address register

	MSB															***LSB***
Content	15:1															0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Bit Description	**15:1** : start_adr new start address. Execution starts once the current list has completed or is skipped. **0** : list_skip same as CMD.list_skip															

Table 3.3.1.2.5.14-14: Register **SADR_CURRENT** (0x14) Start address of current list

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : sadr_current start address of the list currently executed															

Table 3.3.1.2.5.14-15: Register **L0_CURRENT** (0x16) L0 current register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : target_adr value of the currently active list item															

Table 3.3.1.2.5.14-16: Register **L1_CURRENT** (0x18) L1 current register

	MSB															***LSB***
Content	15	14	13:9					8:4					3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15** : trigger_type - value of the currently active list item **14** : trigger_type_ext - value of the currently active list item **13:9** : ch_no - value of the currently active list item **8:4** : no_sum - value of the currently active list item **3:0** : value of the currently active list item															

Table 3.3.1.2.5.14-17: Register **L2_CURRENT** (0x1A) L2 current register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : trigger - value of the currently active list item															

Table 3.3.1.2.5.14-18: Register **ADR_NEXT** (0x1C) next address register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : adr_next - last address read from SRAM. This is not the address of the currently active sample sequence but the last pre- fetched address															

Table 3.3.1.2.5.14-19: Register **SADR_DONE** (0x1E) start address complete

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : sadr_done start address of the list which last executed a "list completed" or "goto" list item. Reading this address will clear an asserted IRQ_STATUS.sadr_done_nempty. Note: A skip list command will not feed sadr_done register and will not set IRQ_STATUS.sadr_done_nempty															

Table 3.3.1.2.5.14-20: Register **STATE** (0x20) state register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	2:0		
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**2:0** : state 0x0: idle 0x1: waiting_for_trigger 0x2: sampling 0x3: executing_other_command 0x4: waiting_for_next_list_item 0x5: finish sampling															

Table 3.3.1.2.5.14-21: Register **DEAD_TIME_WAIT1** (0x22) Dead time wait register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0				-	-	-	-
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dead_time_wait (configuration 1) number of clock cycles between a dead time event and the sampling															

Table 3.3.1.2.5.14-22: Register **DEAD_TIME_WAIT2** (0x24) Dead time wait register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dead_time_wait (configuration 2) number of clock cycles between a dead time event and the sampling															

Table 3.3.1.2.5.14-23: Register **TADR** (0x26) TADR content register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : tadr Content of the TADR register, TADR can be set using special command table syntax described above															

Table 3.3.1.2.5.14-24: Register **AUTOCHANNEL** (0x28) AUTO Channel register

	MSB															***LSB***
Content	-	14:10					9:5					4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**14:10** : auto_ch_w When trigger source edge was W use auto_ch_w as next channel Note: only used for edge counter triggered events with autochannel bit set Note: also used when channel 30 is selected (see channel list above) **9:5** : auto_ch_v When trigger source edge was V use auto_ch_v as next channel Note: only used for edge counter triggered events with autochannel bit set Note: also used when channel 29 is selected (see channel list above) **4:0** : auto_ch_u When trigger source edge was U use auto_ch_u as next channel Note: only used for edge counter triggered events with autochannel bit set Note: also used when channel 28 is selected (see channel list above)															

Table 3.3.1.2.5.14-25: Register **IRQ_STATUS** (0x70) IRQ status register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Bit Description	**15** : oor9 - 1: sum out of range when compared with CMP command **14** : oor8 - 1: sum out of range when compared with CMP command **13** : oor7 - 1: sum out of range when compared with CMP command **12** : oor6 - 1: sum out of range when compared with CMP command **11** : oor5 - 1: sum out of range when compared with CMP command **10** : oor4 - 1: sum out of range when compared with CMP command **9** : oor3 - 1: sum out of range when compared with CMP command **8** : oor2 - 1: sum out of range when compared with CMP command **7** : oor1 - 1: sum out of range when compared with CMP command **6** : oor0 - 1: sum out of range when compared with CMP command **5** : sadr_new_nfull - 1: SADR_NEW can accept a new start address. **4** : sadr_done_nempty - 1: SADR_DONE contains a new completed start address. **3** : list_done_evt - 1: list completed **2** : sum_written_evt - 1: sum written to memory **1** : dma_write_err - 1: a sum was not written to memory. The associated address has been skipped. Cause: a) target address out of valid range, b) write fifo overflow c) missed dead time trigger no_sum>1 **0** : dma_read_err - 1: the next list item could not be read before the previous sample sequence completed. Only asserted when the previous list item was not a special function (i.e. had at least one adc conversion).															

Table 3.3.1.2.5.14-26: Register **IRQ_MASK** (0x74) IRQ mask register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.14-27: Register **IRQ_VENABLE** (0x78) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.14-28: Register **IRQ_VDISABLE** (0x7A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.14-29: Register **IRQ_VMAX** (0x7C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.14-30: Register **IRQ_VNO** (0x7E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	4:0				
Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**4:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.15 PRE PWM Module (PRE_PWM)

3.3.1.2.5.15.1 Description

The PRE_PWM module can be used to automatically generate PWM amplitude waveforms by linear interpolation of table values fetched by the module using a direct memory access (DMA).
The calculated waveform amplitude and "on" values are directly supplied to the PWM module which significantly reduces the CPU load.

3.3.1.2.5.15.2 Features

- base phase counter which represents current electrical rotation angle
  - angular motor speed set register (INC)
- 3 interpolation channels to calculate 3 PWM channel reload values
  - DMA based interpolation table value read
- PWM channel amplitude value offset, scale and limit logic
- external sync signals (e.g. HALL signals) evaluation logic
- vector based interrupt handling

3.3.1.2.5.15.3 Module block diagram

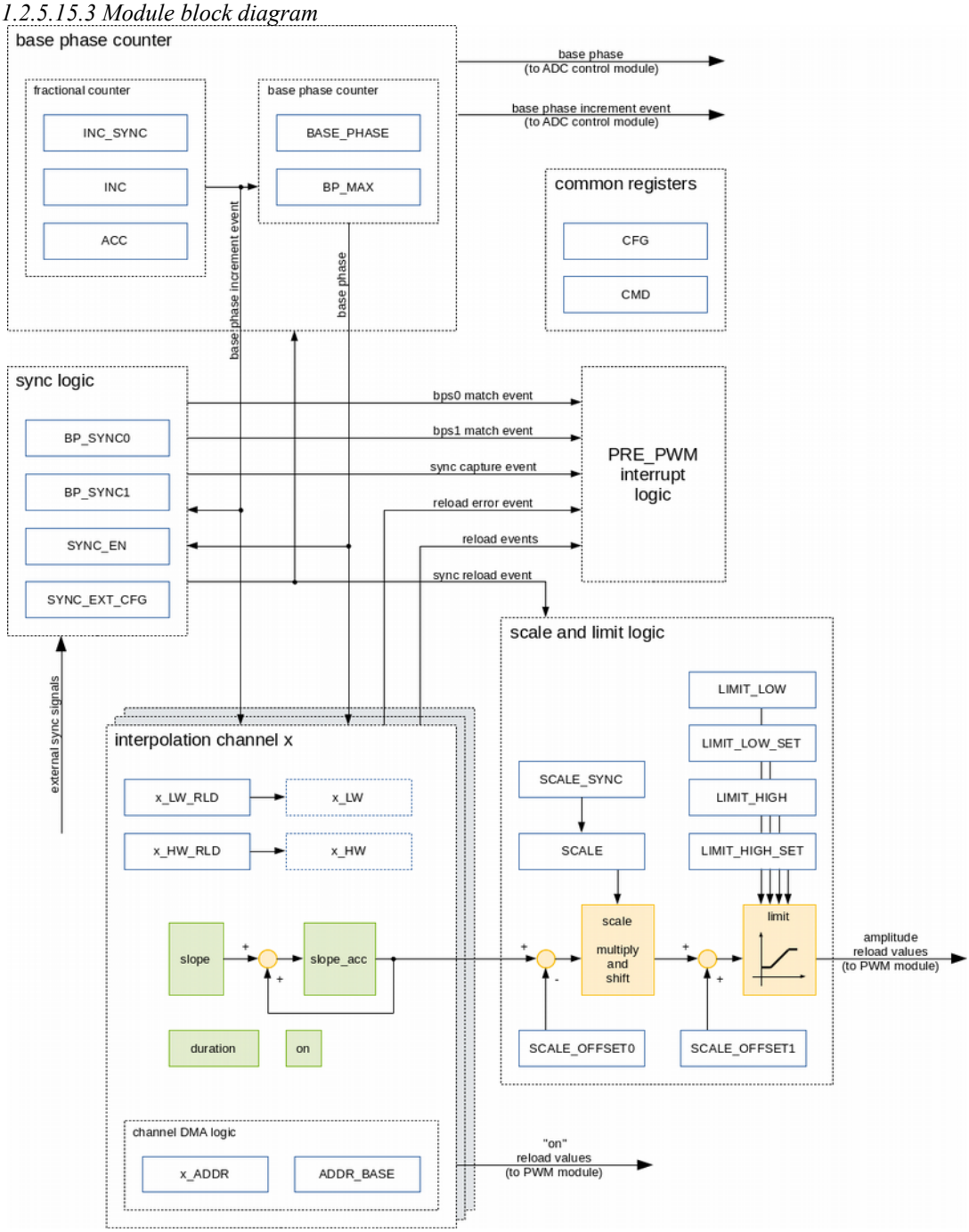


Figure 3.3.1.2.5.15.3-1: Module block diagram

3.3.1.2.5.15.4 Abbreviations

- HS : PWM channel high-side signal
- LS : PWM channel low-side signal
- U, V, W : PRE_PWM interpolation channels (correspond to PWM channels 0 .. 2)

- DMA : direct memory access

3.3.1.2.5.15.5 Module parts

The PRE_PWM module consists of five main parts:

- base phase counter
- 3 interpolation channels
- scale and limit logic
- sync logic
- module interrupt logic

The base phase counter generates the interpolation calculation time base using a modulo accumulator. It's increment value (INC register) can be used to set the calculation speed which is directly proportional to the electrical rotation speed. When the modulo accumulator overflows, a base phase increment event is generated which triggers the interpolation channels slope accumulators to be incremented by their signed slope values.

The interpolation channels generate a piecewise linear waveform based on a combination of start value (slope_acc), increment value (slope) and duration value normally fetched by DMA from a table to reload registers. Each time a base phase increment event occurs, the signed slope value is added to the slope accumulator and duration is decremented by 1. When the result of the duration decrement is zero, the reload register content is copied to the slope_acc, slope and duration working registers and the reload registers are refilled by DMA with the next table entry values. The DMA needs at least 44 clock cycles to fetch a complete table configuration from SRAM, which restricts the duration value: duration value >= 44 / (clock cycles between two successive base phase increment events). When the table is located in FLASH memory, 68 clock cycles are needed.

The scale and limit logic adapts the three interpolation channel slope accumulator values to valid PWM reload values by offsets, multiplication and shift right operations. This is done in a time multiplex manner and a handshake with the PWM module is implemented which restricts the minimum distance of two successive base phase increment events to 5 clock cycles. For this reason, the INC register has not to be set to values larger than 52428.

The sync logic can be used to evaluate external sync signals.

The module interrupt logic evaluates some sync logic events, the reload event and reload error event of all interpolation channels and supplies a module interrupt signal to the system main vector interrupt controller.

3.3.1.2.5.15.6 Common registers

- CFG register:
  - configuration to control PRE_PWM module behavior

- CMD register:
  - software commands register

Table 3.3.1.2.5.15.6-1: Common registers

Register Name	Address	Description
CFG	0x00	config register
CMD	0x06	command register

Table 3.3.1.2.5.15.6-2: Register **CFG** (0x00) config register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5	4	3:2		1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : switch_gd_td - interpolation channel trigger select 0 : interpolation channel calculation timing based on base phase increment events (typical value) In this case the PRE_PWM module generates more PWM reload values than the PWM module uses. It means that the PRE_PWM reload value calculation data rate is higher than the PWM module reload data usage rate. 1 : interpolation channel calculation timing based on PWM reload events In this case the PRE_PWM reload value calculation data rate is given by the PWM module reload data rate. **4** : swap_uv - swap U and V interpolation channel reload signals (amplitude and "on" signals) supplied to PWM module 0 : no signal swap 1 : swap signals (alternates rotary field direction) **3:2** : on_state - "on" signals (supplied to PWM module) state when interpolation channel x_HW.use_cfg_on is set 00 : both HS and LS off 01 : HS off, LS on 10 : HS on, LS off 11 : both HS and LS on (typical value) **Note:** If x_HW.use_cfg_on is set, changing on_state immediately changes "on" reload signals supplied to PWM module ! For clean operation, please change this value only when current used use_cfg_on is 0. **1** : dma_en - interpolation channels reload registers DMA enable 0 : reload register DMA disabled 1 : reload register DMA enabled **0** : inc_en - ACC register increment enable 0 : ACC remains unchanged 1 : INC added to ACC every clock cycle															

Table 3.3.1.2.5.15.6-3: Register **CMD** (0x06) command register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W
Bit Description	**7** : uvw_scale - 1 : trigger all interpolation channels slope accumulator scaling (e.g. in case software updates slope accumulator value) **6** : w_dma - 1 : trigger interpolation channel W DMA reload **5** : v_dma - 1 : trigger interpolation channel V DMA reload **4** : u_dma - 1 : trigger interpolation channel U DMA reload **3** : w_reload - 1 : generate interpolation channel W reload event **2** : v_reload - 1 : generate interpolation channel V reload event **1** : u_reload - 1 : generate interpolation channel U reload event **0** : bps0 - 1 : generate sync reload event															

3.3.1.2.5.15.7 Base phase counter

The base phase counter consists of a fractional counter part:

- ACC register:
  - represents upper 16 bits of the 18 bit fractional accumulator
  - incremented by INC register value every clock cycle

- INC register:
  - has to be set according to angular motor speed
  - has to be set to a value which guaranties a minimum number of clock cycles between two successive base phase increment events
    - if table is located in SRAM : minimum number of clock cycles = 44
    - if table is located in FLASH : minimum number of clock cycles = 68
  - loaded from INC_SYNC register at sync reload event when INC_SYNC has been written since last sync reload event

- INC_SYNC register:
  - INC register reload value

- base phase increment event:
  - generated at ACC overflow

and a base phase counter part:

- BASE_PHASE register:
  - represents current electrical angle
  - incremented at base phase increment event with modulo (BP_MAX+1)
  - supplied to ADC control module for electrical angle based ADC conversion

- BP_MAX register:
  - BASE_PHASE maximum value
  - BP_MAX+1 equals the number of base phase increment events per electrical rotation

Table 3.3.1.2.5.15.7-1: Base phase counter

Register Name	*Address*	*Description*
BP_MAX	0x02	base phase max register
INC	0x08	increment register
ACC	0x0A	fractional counter register
BASE_PHASE	0x0C	current base phase register
INC_SYNC	0x1C	increment reload register

Table 3.3.1.2.5.15.7-2: Register **ACC** (0x0A) fractional counter register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : acc - upper 16 bits of the 18 bit fractional accumulator **Note:** The fractional accumulator is 18 bit wide. Only the upper 16 bits are readable using this register. When writing, the bits 0 and 1 of the 18 bit accumulator are set to 0.															

Table 3.3.1.2.5.15.7-3: Register **INC** (0x08) increment register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : inc - fractional accumulator increment value **Note:** The maximum allowed inc value is 52428 to guaranty a minimum distance of at least 5 clock cycles between two successive base phase increment events. This is needed for correct scaling logic behavior.															

Table 3.3.1.2.5.15.7-4: Register **INC_SYNC** (0x1C) increment reload register

	MSB															*LSB*
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : inc - INC register reload value															

Table 3.3.1.2.5.15.7-5: Register **BASE_PHASE** (0x0C) current base phase register

	MSB															*LSB*
Content	-	-	-	-	11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:0** : base_phase - current base phase value															

Table 3.3.1.2.5.15.7-6: Register **BP_MAX** (0x02) base phase max register

	MSB															***LSB***
Content	-	-	-	-	11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:0** : bp_max - base phase maximum (modulo) value When base phase value equals bp_max, a base phase increment event causes base phase value to "overflow" to 0.															

3.3.1.2.5.15.8 Interpolation channels

- in the following description, x is used as place holder in channel U, V or W register names
  - interpolation channel U corresponds to PWM channel 0
  - interpolation channel V corresponds to PWM channel 1 (if not swapped using CFG register)
  - interpolation channel W corresponds to PWM channel 2 (if not swapped using CFG register)

- linear interpolation register bitfields:
  - duration, slope, slope accumulator start value

- x_LW register:
  - contains part of the interpolation channel configuration
  - loaded from x_LW_RLD register at reload event

- x_LW_RLD register:
  - x_LW register reload value
  - loaded from memory by DMA at reload event, when DMA is enabled

- x_HW register:
  - contains part of the interpolation channel configuration
  - loaded from x_HW_RLD register at reload event

- x_HW_RLD register:
  - x_HW register reload value
  - loaded from memory by DMA at reload event, when DMA is enabled

- duration counter
  - equals x_LW.duration
  - down counting at base phase increment event

- slope accumulator
  - equals x_HW.slope_acc
  - signed x_LW.slope value is added to unsigned slope accumulator
  - **Note:** Please take care to not underflow slope accumulator below 0, because there is no saturation of "negative" values to 0 !

- reload event:
  - generated when duration counter < 2 at base phase increment event
  - is interrupt source

- reload error event:
  - generated at reload event when not all reload registers (of all interpolation channels) have been updated by DMA or written by software since previous reload event
  - is interrupt source

- ADDR_BASE register:
  - common table base address (same for all channels)

- x_ADDR register:
  - interpolation channel DMA address offset

- DMA reload:
  - when CFG register DMA enable is set:
    - triggered by reload event
    - loads x_LW_RLD and x_HW_RLD registers from memory address ADDR_BASE + x_ADDR
    - post increments x_ADDR by 4
    - if loaded duration value is 0:
      - x_ADDR is set to 0 and DMA loading is repeated

- interpolation table:
  - consists of 32 bit structure entries which contain DMA load values for x_LW_RLD and x_HW_RLD registers
  - a table represents a complete electrical rotation
  - the same table is used for all three interpolation channels
  - a table entry defines an interpolation straight line with supporting point (starting point), slope and duration
  - the number of valid table entries should be a multiple of 6 to be able to start every 30 degrees with a table entry (e.g. number of entries = 6 * 10 = 60)
  - if the same duration value is used for all valid table entries, BP_MAX = duration value * number of table entries - 1 (e.g. BP_MAX = 66 * 60 - 1 = 3959).
    - BP_MAX = 3959 results in a base phase angle accuracy of 360 degrees / 3960 = 0.09 degrees per base phase step.

Table 3.3.1.2.5.15.8-1: Interpolation table entry

bit number	*field name*	*comment*
31	on	x_HW_RLD.on DMA load value
30:16	slope_acc	x_HW_RLD.slope_acc DMA load value the interpolation accumulator (slope accumulator) defines the straight line supporting point (starting point)
15:9	slope	x_LW_RLD.slope DMA load value defines the straight line increase (increment value) between two successive base phase increment events
8:0	duration	x_LW_RLD.duration DMA load value defines the number of interpolation straight line slope steps before next table entry has to be used **Note:** A value of 0 marks the end of the table.

- an interpolation table may be defined using the following steps:
  - for each table entry do the following:
    1. slope value [n] = round((ideal supporting point [n+1] - supporting point [n]) / duration [n])
    2. supporting point [n+1] = supporting point [n] + slope [n] * duration [n]
  - n : table entry index
  - ideal supporting point means the ideal function value (usually a float value)
  - supporting point means the table entry value (integer value)
  - the "end of table" entry may be a 32 bit zero value (at least duration has to be 0)

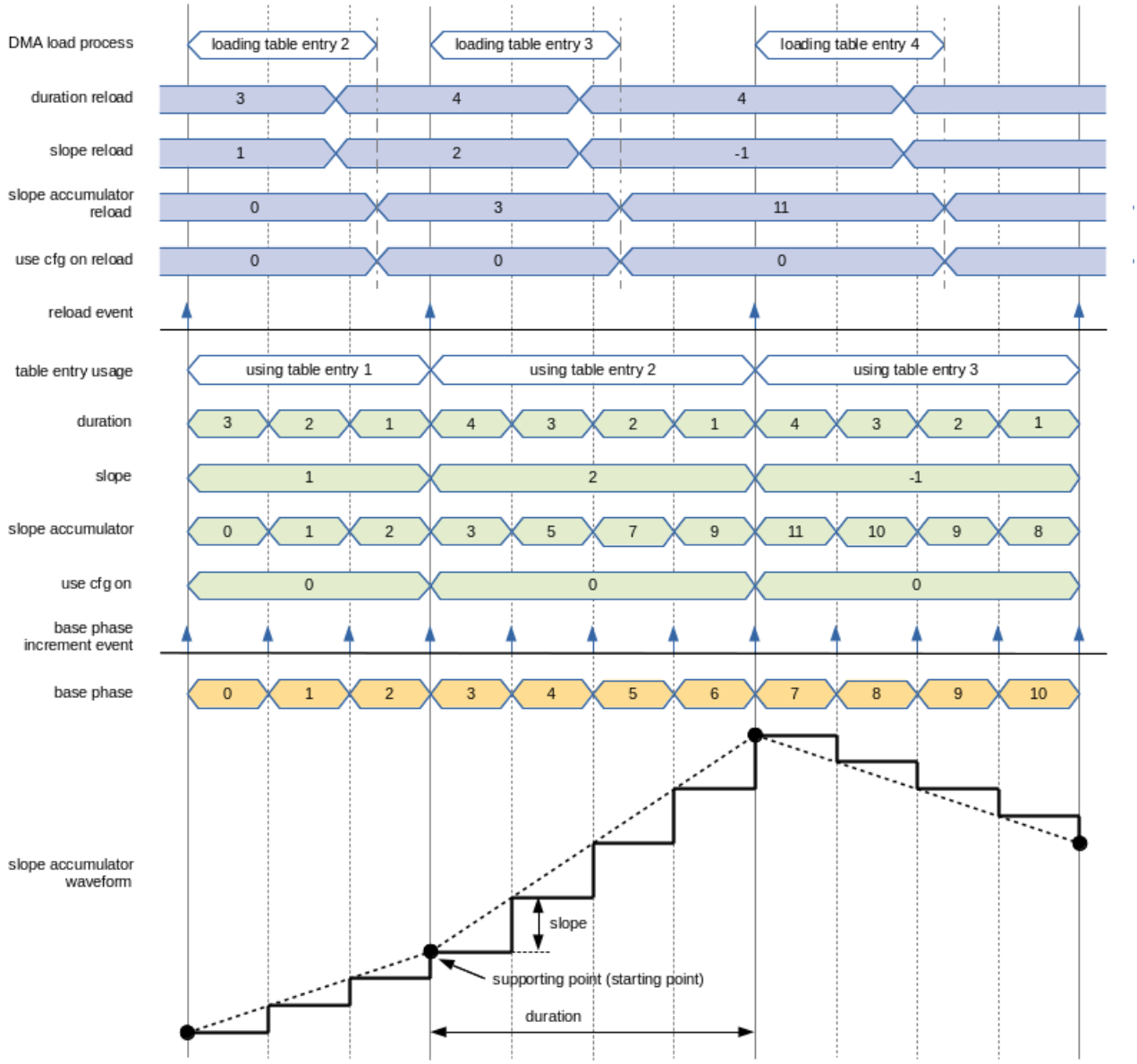


Figure 3.3.1.2.5.15.8-1: PRE_PWM timing example

Table 3.3.1.2.5.15.8-2: Interpolation channels

Register Name	*Address*	*Description*
ADDR_BASE	0x04	DMA base address register
U_ADDR	0x30	DMA address offset register
U_LW	0x32	table entry low word register
U_HW	0x34	table entry high word register
U_LW_RLD	0x36	table entry low word reload register
U_HW_RLD	0x38	table entry high word reload register
V_ADDR	0x40	DMA address offset register
V_LW	0x42	table entry low word register
V_HW	0x44	table entry high word register
V_LW_RLD	0x46	table entry low word reload register
V_HW_RLD	0x48	table entry high word reload register
W_ADDR	0x50	DMA address offset register
W_LW	0x52	table entry low word register
W_HW	0x54	table entry high word register
W_LW_RLD	0x56	table entry low word reload register
W_HW_RLD	0x58	table entry high word reload register

Table 3.3.1.2.5.15.8-3: Register **U_LW** (0x32) table entry low word register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - signed slope value, added to slope accumulator every base phase increment event **8:0** : duration - interpolation duration count (number of base phase increment events)															

Table 3.3.1.2.5.15.8-4: Register **V_LW** (0x42) table entry low word register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - signed slope value, added to slope accumulator every base phase increment event **8:0** : duration - interpolation duration count (number of base phase increment events)															

Table 3.3.1.2.5.15.8-5: Register **W_LW** (0x52) table entry low word register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - signed slope value, added to slope accumulator every base phase increment event **8:0** : duration - interpolation duration count (number of base phase increment events)															

Table 3.3.1.2.5.15.8-6: Register **U_HW** (0x34) table entry high word register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - CFG.on_state usage 0 : HS and LS are on 1 : HS and LS are handled like configured by CFG.on_state **14:0** : slope_acc - unsigned slope accumulator interpolation start value															

Table 3.3.1.2.5.15.8-7: Register **V_HW** (0x44) table entry high word register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - CFG.on_state usage 0 : HS and LS are on 1 : HS and LS are handled like configured by CFG.on_state **14:0** : slope_acc - unsigned slope accumulator interpolation start value															

Table 3.3.1.2.5.15.8-8: Register **W_HW** (0x54) table entry high word register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - CFG.on_state usage 0 : HS and LS are on 1 : HS and LS are handled like configured by CFG.on_state **14:0** : slope_acc - unsigned slope accumulator interpolation start value															

Table 3.3.1.2.5.15.8-9: Register **U_LW_RLD** (0x36) table entry low word reload register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - U_LW.slope reload value **8:0** : duration - U_LW.duration reload value															

Table 3.3.1.2.5.15.8-10: Register **V_LW_RLD** (0x46) table entry low word reload register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - V_LW.slope reload value **8:0** : duration - V_LW.duration reload value															

Table 3.3.1.2.5.15.8-11: Register **W_LW_RLD** (0x56) table entry low word reload register

	MSB															***LSB***
Content	15:9							8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:9** : slope - W_LW.slope reload value **8:0** : duration - W_LW.duration reload value															

Table 3.3.1.2.5.15.8-12: Register **U_HW_RLD** (0x38) table entry high word reload register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - U_HW.use_cfg_on reload value **14:0** : slope_acc - U_HW.slope_acc reload value															

Table 3.3.1.2.5.15.8-13: Register **V_HW_RLD** (0x48) table entry high word reload register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - V_HW.use_cfg_on reload value **14:0** : slope_acc - V_HW.slope_acc reload value															

Table 3.3.1.2.5.15.8-14: Register **W_HW_RLD** (0x58) table entry high word reload register

	MSB															***LSB***
Content	15	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : use_cfg_on - W_HW.use_cfg_on reload value **14:0** : slope_acc - W_HW.slope_acc reload value															

Table 3.3.1.2.5.15.8-15: Register **ADDR_BASE** (0x04) DMA base address register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : addr_base - interpolation channels DMA base address value **Note:** Base address value has to be even (bit 0 = 0) !															

Table 3.3.1.2.5.15.8-16: Register **U_ADDR** (0x30) DMA address offset register

	MSB															***LSB***
Content	-	-	-	-	-	10:0										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:0** : addr - interpolation channel DMA address offset effective DMA address = common DMA base address + channel DMA address offset **Note:** Address value has to be a multiple of 4 (bits 1 and 0 = 0) !															

Table 3.3.1.2.5.15.8-17: Register **V_ADDR** (0x40) DMA address offset register

	MSB															***LSB***
Content	-	-	-	-	-	10:0										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:0** : addr - interpolation channel DMA address offset effective DMA address = common DMA base address + channel DMA address offset **Note:** Address value has to be a multiple of 4 (bits 1 and 0 = 0) !															

Table 3.3.1.2.5.15.8-18: Register **W_ADDR** (0x50) DMA address offset register

	MSB															***LSB***
Content	-	-	-	-	-	10:0										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:0** : addr - interpolation channel DMA address offset effective DMA address = common DMA base address + channel DMA address offset **Note:** Address value has to be a multiple of 4 (bits 1 and 0 = 0) !															

3.3.1.2.5.15.9 Scale and limit logic

- scales and limits slope accumulator value to PWM amplitude reload signal values

- uses common scale and limit registers (same for all channels):

- SCALE register:
  - PWM amplitude reload value scaling (consists of a multiply and a shift right value)
  - loaded from SCALE_SYNC register at sync reload event when SCALE_SYNC register has been written by software since previous sync reload event

- SCALE_SYNC register:
  - SCALE register reload value

- SCALE_OFFSET0 registers:
  - pre-amplitude-scale offset value
  - shifts the unsigned slope accumulator value downward / centers the unsigned slope accumulator waveform at 0 to be able to linear scale the resulting signed value

- SCALE_OFFSET1 registers:
  - post-amplitude-scale offset value
  - shifts the signed scaled values (centered at 0) upward to a positive range to be used as PWM amplitude reload values.

- LIMIT_LOW register:
  - PWM amplitude low limit compare value

- LIMIT_LOW_SET register:
  - PWM amplitude minimum value to use when value < LIMIT_LOW compare value
  - can be used to setup a minimum PWM HS on time

- LIMIT_HIGH register:
  - PWM amplitude high limit compare value

- LIMIT_HIGH_SET register:
  - PWM amplitude maximum value to use when value > LIMIT_HIGH compare value
  - can be used to setup a minimum PWM LS on time

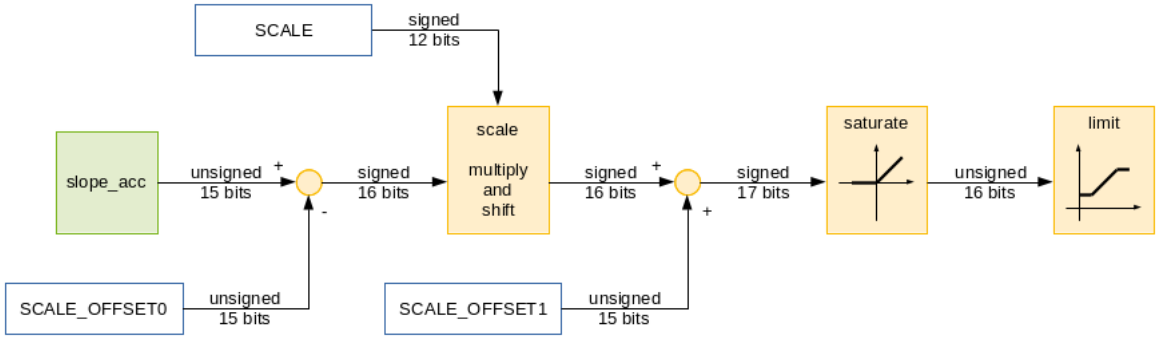


Figure 3.3.1.2.5.15.9-1: Scale and limit logic

Table 3.3.1.2.5.15.9-1: Scale and limit logic

Register Name	*Address*	*Description*
SCALE	0x0E	scale register
SCALE_OFFSET0	0x10	scale offset 0 register
SCALE_SYNC	0x1A	scale reload register
SCALE_OFFSET1	0x1E	scale offset 1 register
LIMIT_LOW	0x20	PWM low limit compare register
LIMIT_LOW_SET	0x22	PWM low limit set value register
LIMIT_HIGH	0x24	PWM high limit compare register
LIMIT_HIGH_SET	0x26	PWM high limit set value register

Table 3.3.1.2.5.15.9-2: Register **SCALE** (0x0E) scale register

	MSB															***LSB***
Content	-	14:12			11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**14:12** : scale_shift - scale sub block right shift value (please see SCALE.scale description) **11:0** : scale - signed amplitude multiplicand signed scale sub block input value = unsigned slope accumulator - unsigned SCALE_OFFSET0 signed scale sub block scaled value = signed scale sub block input value * signed SCALE.scale / (2 ^ (10 + SCALE.scale_shift)) signed scale sub block output value = signed scale sub block scaled value + unsigned SCALE_OFFSET1 unsigned limit sub block input value = saturated signed scale sub block output value (negative values forced to 0) **Note:** To guaranty a correct scale logic function, the allowed scale value ranges depending on scale_shift are: scale_shift = 0 : scale range is -512 .. +511, +512 scale_shift = 1 : scale range is -1024 .. +1023, +1024 scale_shift >= 2 : scale range is -2048 .. +2047 (full range) Scaling a 16 bit value by a 10 bit range multiplicant followed by a shift right of 10+0 bits, results in a 16 bit value: 16 + 10 - (10+0) = 16 Scaling a 16 bit value by a 11 bit range multiplicant followed by a shift right of 10+1 bits, results in a 16 bit value: 16 + 11 - (10+1) = 16 Scaling a 16 bit value by a 12 bit range multiplicant followed by a shift right of 10+2 bits, results in a 16 bit value: 16 + 12 - (10+2) = 16															

Table 3.3.1.2.5.15.9-3: Register **SCALE_SYNC** (0x1A) scale reload register

	MSB															***LSB***
Content	-	14:12			11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**14:12** : scale - SCALE.scale_shift reload value **11:0** : scale_shift - SCALE.scale reload value															

Table 3.3.1.2.5.15.9-4: Register **SCALE_OFFSET0** (0x10) scale offset 0 register

	MSB															***LSB***
Content	-	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**14:0** : scale_offset - pre scale sub block offset value (please see SCALE.scale description)															

Table 3.3.1.2.5.15.9-5: Register **SCALE_OFFSET1** (0x1E) scale offset 1 register

	MSB															***LSB***
Content	-	14:0														
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**14:0** : scale_offset - post scale sub block offset value (please see SCALE.scale description) **Note:** The value depends on the PWM period waveform used and the type of PWM modulation to be implemented (sinusoidal or space vector modulation).															

Table 3.3.1.2.5.15.9-6: Register **LIMIT_LOW** (0x20) PWM low limit compare register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : limit_low - PWM amplitude reload value: low limit compare value If limit sub block input value < LIMIT_LOW, LIMIT_LOW_SET value is used as PWM amplitude reload value															

Table 3.3.1.2.5.15.9-7: Register **LIMIT_LOW_SET** (0x22) PWM low limit set value register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : limit_low_set - minimum PWM amplitude reload value															

Table 3.3.1.2.5.15.9-8: Register **LIMIT_HIGH** (0x24) PWM high limit compare register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : limit_high - PWM amplitude reload value: high limit compare value If limit sub block input value > LIMIT_HIGH, LIMIT_HIGH_SET value is used as PWM amplitude reload value															

Table 3.3.1.2.5.15.9-9: Register **LIMIT_HIGH_SET** (0x26) PWM high limit set value register

	MSB															***LSB***
Content	15:0															
Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : limit_high_set - maximum PWM amplitude reload value															

3.3.1.2.5.15.10 Sync logic

- BP_SYNC0 register:
  - compare value for bps0 match event generation
- BP_SYNC1 register:
  - compare value for bps1 match event generation
  - captures BASE_PHASE value at sync capture event
- SYNC_EN register:
  - bps0 and pbs1 events behavior configuration
- SYNC_EXT_CFG register:
  - external sync signal handling behavior configuration
- bps0 match event:
  - generated if SYNC_EN.bps0 != 00 when BASE_PHASE == BP_SYNC0 at base phase increment event
  - is interrupt source
- bps1 match event:
  - generated if SYNC_EN.bps1 == 10 when BASE_PHASE == BP_SYNC1 at base phase increment event
  - is interrupt source
- sync reload event:
  - generated if SYNC_EN.bps0 == 10 at bps0 match event
- sync capture event:
  - generated if SYNC_EN.bps1 == 01 at selected external sync signal edge
  - is interrupt source
- external sync signals:
  - device input signals
  - filtered using an adjustable integral filter
  - generate sync capture events
  - can be evaluated as motor position reference

Table 3.3.1.2.5.15.10-1: Sync logic

Register Name	***Address***	***Description***
BP_SYNC0	0x12	base phase sync 0 register
BP_SYNC1	0x14	base phase sync 1 register

Register Name	Address	Description
SYNC_EN	0x16	sync enable register
SYNC_EXT_CFG	0x18	external sync config register

Table 3.3.1.2.5.15.10-2: Register **BP_SYNC0** (0x12) base phase sync 0 register

	MSB															LSB
Content	-	-	-	-	11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:0** : bp_sync - register usage depends on SYNC_EN register configuration															

Table 3.3.1.2.5.15.10-3: Register **BP_SYNC1** (0x14) base phase sync 1 register

	MSB															LSB
Content	-	-	-	-	11:0											
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**11:0** : bp_sync - register usage depends on SYNC_EN register configuration															

Table 3.3.1.2.5.15.10-4: Register **SYNC_EN** (0x16) sync enable register

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:2** : bps1 - BP_SYNC1 register usage configuration 00 : BP_SYNC1 unused 01 : sync capture event generation enabled 10 : bps1 match event generation enabled **1:0** : bps0 - BP_SYNC0 register usage configuration 00 : BP_SYNC0 unused 01 : bps0 match event and sync reload event generation enabled 10 : bps0 match event generation enabled															

Table 3.3.1.2.5.15.10-5: Register **SYNC_EXT_CFG** (0x18) external sync config register

	MSB															***LSB***
Content	-	-	-	12	11:10		9:2								1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**12** : edge - sync capture event source edge selection 0 : selected external sync signal rising edge 1 : selected external sync signal falling edge **11:10** : intf_set - external sync signal integration filter set command 10 : clear and set actual state to 0 11 : clear and set actual state to 1 **9:2** : intf - external sync signal integration filter threshold value **1:0** : src - sync capture event source selection 0 : external sync signal 0 selected 1 : external sync signal 1 selected 2 : external sync signal 2 selected 3 : external sync signal 3 selected															

3.3.1.2.5.15.11 Interrupt handling registers

Table 3.3.1.2.5.15.11-1: Interrupt handling registers

Register Name	***Address***	***Description***
IRQ_STATUS	0x70	IRQ status register
IRQ_MASK	0x74	IRQ mask register
IRQ_VENABLE	0x78	IRQ vector enable register
IRQ_VDISABLE	0x7A	IRQ vector disable register
IRQ_VMAX	0x7C	IRQ max vector register
IRQ_VNO	0x7E	IRQ vector number register

Table 3.3.1.2.5.15.11-2: Register **IRQ_STATUS** (0x70) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8** : w_dma_rdy - interpolation channel W DMA idle **7** : v_dma_rdy - interpolation channel V DMA idle **6** : u_dma_rdy - interpolation channel U DMA idle **5** : reload_error_evt - interpolation channels reload error event **4** : w_reload_evt - interpolation channel W reload event **3** : v_reload_evt - interpolation channel V reload event **2** : u_reload_evt - interpolation channel U reload event **1** : bps1_evt - bps1 match event or sync capture event **0** : bps0_evt - bps0 match event															

Table 3.3.1.2.5.15.11-3: Register **IRQ_MASK** (0x74) IRQ mask register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	8:0								
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**8:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.15.11-4: Register **IRQ_VENABLE** (0x78) IRQ vector enable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.15.11-5: Register **IRQ_VDISABLE** (0x7A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.15.11-6: Register **IRQ_VMAX** (0x7C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.15.11-7: Register **IRQ_VNO** (0x7E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

3.3.1.2.5.16 PWM Module (PWMN)

3.3.1.2.5.16.1 Description

The PWM module implements a 4 channel 16 bit PWM with dead time insertion and overcurrent handling.

3.3.1.2.5.16.2 Features

- common PWM period counter
  - sawtooth, inverted sawtooth, triangle and inverted triangle counting
  - 4 bit prescaler, 16 bit counter
  - generates reload event used to update PWM channel registers from their reload equivalents
  - 4 bit Nth PWM period start event counter to reduce PWM channel register reload rate

- 4 PWM channels
  - left and right edge aligned PWM waveform
  - center aligned PWM waveform
  - start and stop time stamp generated PWM waveform
  - dead time insertion
    - independent or common dead time configuration

- overcurrent handling
  - evaluation of NALLOFF device input signal
  - configurable asynchronous PWM signal masking
  - configurable filtered synchronous PWM signal masking and PWM state change

- generates ADC conversion trigger (dead time event) signals fed to the ADC control module

- PRE_PWM module interface to reduce CPU load
  - PRE_PWM module feeds PWM module registers with self-advancing channel reload values

- vector based interrupt handling
  - overcurrent interrupt
  - PWM Nth period start and middle event interrupts
  - dead time event interrupts

3.3.1.2.5.16.3 Module Block Diagram

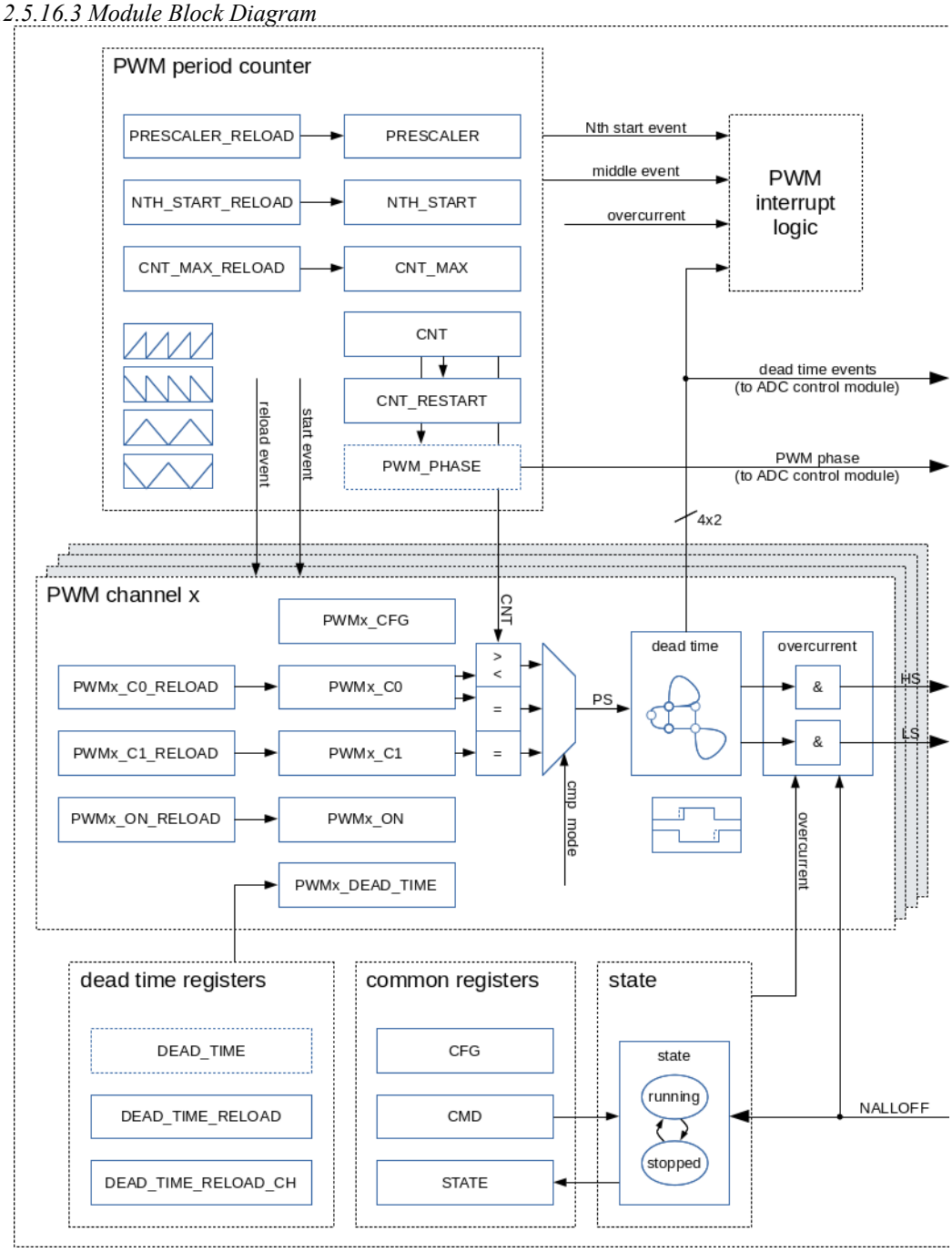


Figure 3.3.1.2.5.16.3-1: Module Block Diagram

3.3.1.2.5.16.4 Abbreviations

- HS : PWM channel high-side signal
- LS : PWM channel low-side signal
- PS : PWM signal before split to HS and LS and dead time insertion
- LH : PWM PS change from low to high
- HL : PWM PS change from high to low
- DT : dead time
- DT_LH : PS low to high dead time
- DT_HL : PS high to low dead time
- DT_CNT : dead time counter

3.3.1.2.5.16.5 Module parts

The PWM module consists of four main parts:

- PWM period counter
- 4 PWM channels
- PWM module state handling logic
- PWM module interrupt logic

The PWM period counter generates a mode dependent, prescaled CNT waveform, start and reload event used by each channel in parallel. In addition Nth start and middle events are supplied to the PWM module interrupt logic. The PWM phase signal is supplied to the ADC control module for time stamp triggered ADC conversion.

Each PWM channel implements a SP signal using the CNT waveform and a configurable compare logic. The internal PS signal is used to generate high and low side signals including dead time insertion. An overcurrent condition can be configured to mask the high and low side PWM signals. Each channel supplies two dead time events (LH and HL) to the PWM module interrupt logic. The dead time event signals are also supplied to the ADC control module for dead time triggered ADC conversion.

The PWM module state handling logic implements two states: running and stopped. The CMD register can be used to change the PWM module state. An overcurrent condition can be evaluated by the PWM state logic to switch off the PWM outputs and change PWM module state to stopped.

The PWM module interrupt logic evaluates the overcurrent flag, Nth start event, middle event and the dead time events of all PWM channels and supplies a module interrupt signal to the system main vector interrupt controller.

3.3.1.2.5.16.6 Common registers and PWM state

- CFG (configuration) register:
  - used to configure PWM module behavior
- CMD (command) register:
  - used to change the PWM module state
- STATE register:
  - allows to check the PWM module state
- overcurrent evaluation:
  - NALLOFF signal can be used to signal overcurrent
  - overcurrent can be configured to asynchronously mask PWM HS and LS signals
  - overcurrent can be configured to be filtered and evaluated by PWM state unit
    - integration filter threshold can be configured using CFG.oc_intf

- when a filtered overcurrent occurs:
  - STATE.oc flag, which is an interrupt source, will be set to 1
  - STATE.run changes from running to stopped
  - STATE.oc can be cleared by CMD.oc

Table 3.3.1.2.5.16.6-1: Common registers

Register Name	*Address*	*Description*
CFG	0x00	config register
CMD	0x02	command register
STATE	0x04	state register

Table 3.3.1.2.5.16.6-2: Register **CFG** (0x00) config register

	MSB															***LSB***
Content	15	14	13	12:6							5	4	3	2	1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : dt_evt_continue - dead time events behavior configuration 0 : generate dead time events only when PS does not have 0% or 100% duty cycle 1 : generate dead time events independent from PS duty cycle **14** : restart_cap_en - this configuration bit is not evaluated in hardware **13** : restart_en - this configuration bit is not evaluated in hardware **12:6** : oc_intf - synchronous overcurrent evaluation integral filter threshold value configuration **5** : oc_syn_en - synchronous overcurrent evaluation enable 0 : synchronous overcurrent evaluation disabled 1 : synchronous overcurrent evaluation enabled **4** : oc_asyn_en - asynchronous overcurrent PWM signal masking enable 0 : asynchronous PWM signal masking disabled 1 : asynchronous PWM signal masking enabled **3** : dead_time_mode - PWM dead time insertion mode 0 : only rising edge of PS is delayed 1 : rising and falling edge of PS are delayed **2** : middle_rl - PWM period middle channel register reload enable This bit is only evaluated for triangle modes. 0 : no reload event at middle of period 1 : reload event at middle of period **1:0** : cnt_mode - PWM period counter waveform mode 00 : sawtooth waveform 01 : inverted sawtooth waveform 10 : triangle waveform 11 : inverted triangle waveform															

Table 3.3.1.2.5.16.6-3: Register **CMD** (0x02) command register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W
Bit Description	**3** : intf_sync_clr - writing 1 : clears overcurrent integral filter **2** : restart - PWM period counter restart command writing 1 : restarts PWM period counter and captures CNT to CNT_RESTART register **1** : oc - writing 1 : clears PWM state overcurrent flag **0** : run - PWM start / stop command writing 0 : changes PWM module state to "stopped" writing 1 : changes PWM module state to "running"															

Table 3.3.1.2.5.16.6-4: Register **STATE** (0x04) state register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**1** : oc - 1 : overcurrent occurred **0** : run - PWM module state 0 : stopped 1 : running															

3.3.1.2.5.16.7 PWM period counter

- PRESCALER register:
  - down counting with system clock
  - loaded from PRESCALER_RELOAD when PRESCALER == 0

- CNT_MAX register:
  - loaded from CNT_MAX_RELOAD at reload event and
    - in case of up counting CNT when CNT == CNT_MAX
    - in case of down counting CNT when CNT == 0

- CNT (PWM period counter):
  - incremented or decremented when PRESCALER == 0
  - loaded with 0
    - in case of up counting CNT when CNT == CNT_MAX
  - loaded from CNT_MAX
    - in case of down counting CNT when CNT == 0
  - used by PWM channels

- CNT_RESTART register:
  - loaded with CNT value on CMD.restart

- PWM_PHASE register:
  - sawtooth waveform mode:
    - PWM phase = CNT
    - PWM period = one time CNT_MAX+1 cycles
  - triangle waveform mode:
    - PWM phase = 0x8000 + (CNT & 0x7FFF)
    - PWM period = two times CNT_MAX+1 cycles
  - PWM phase can be used by ADC control module for time stamp based ADC conversion

- NTH_START register:
  - down counting at start events
  - loaded from NTH_START_RELOAD at Nth start event

- start event:
  - generated at each PWM period start
  - generated in sawtooth and triangle modes
  - used by PWM channels

- Nth start event:
  - generated at start event when NTH_START == 0
  - is interrupt source

- middle event:
  - generated in the middle of the PWM period
  - generated in triangle modes only
  - no middle events will be generated when NTH_START_RELOAD is evaluated as non-zero value at Nth start event !
  - is interrupt source

- reload event:
  - combination of Nth start event and middle event
  - used by PWM channels

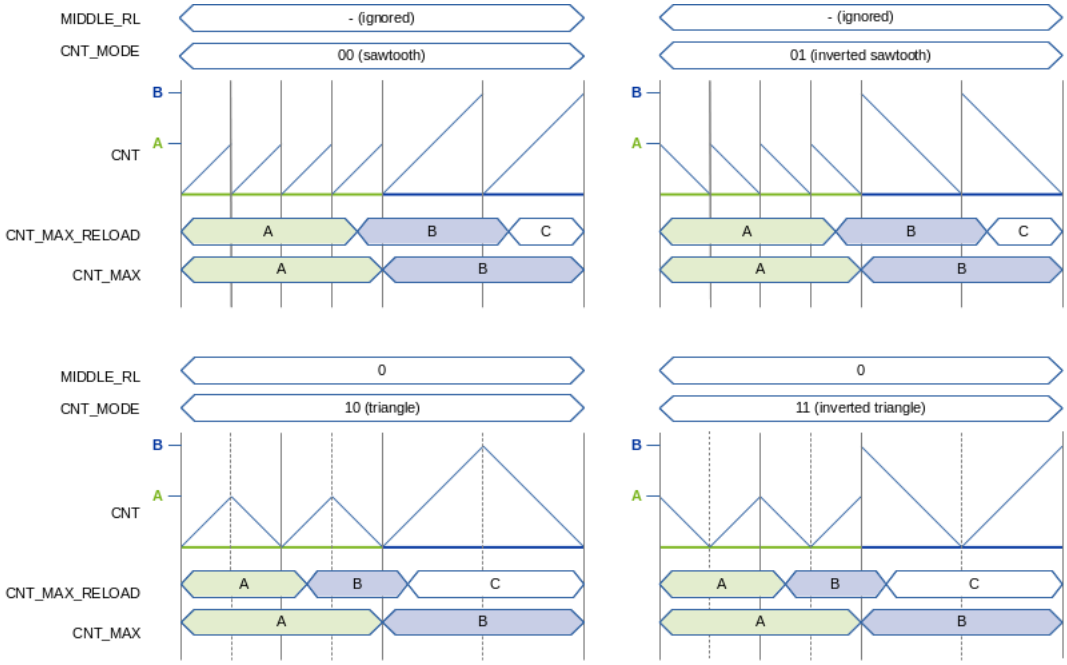


Figure 3.3.1.2.5.16.7-1: PWM period counter mode dependent CNT_MAX reload behavior (middle_rl = 0)

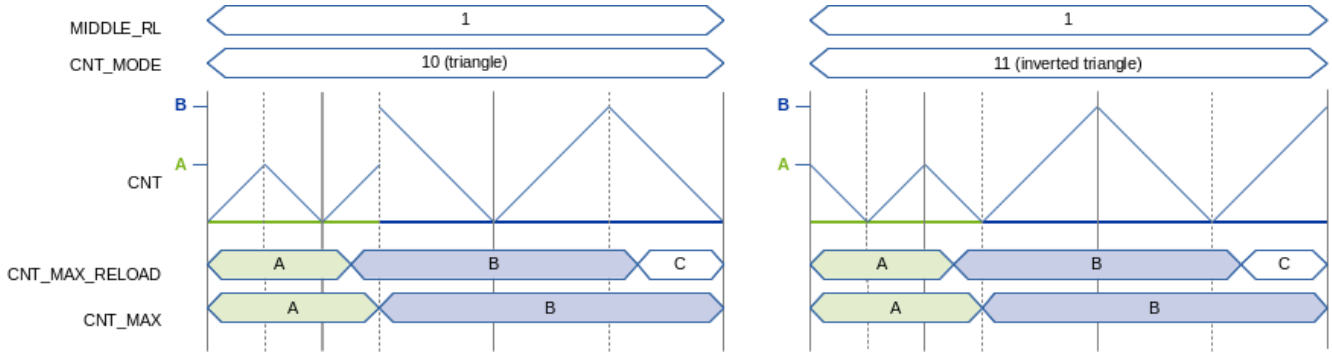


Figure 3.3.1.2.5.16.7-2: PWM period counter mode dependent CNT_MAX reload behavior (middle_rl = 1)

Table 3.3.1.2.5.16.7-1: PWM period counter

Register Name	Address	Description
CNT	0x06	counter register
PWM_PHASE	0x08	PWM phase register
PRESCALER	0x0A	current prescaler register
CNT_MAX	0x0C	current max counter register
PRESCALER_RELOAD	0x10	prescaler reload register
CNT_MAX_RELOAD	0x12	max counter reaload register
CNT_RESTART	0x16	value of CNT
NTH_START	0x18	Configure n-th start

Register Name	Address	Description
NTH_START_RELOAD	0x1A	n-th start reload value

Table 3.3.1.2.5.16.7-2: Register **PRESCALER** (0x0A) current prescaler register

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : prescaler - PWM period counter prescaler prescale period = PRESCALER.prescaler + 1 clock cycles															

Table 3.3.1.2.5.16.7-3: Register **PRESCALER_RELOAD** (0x10) prescaler reload register

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : reload - PRESCALER register reload value															

Table 3.3.1.2.5.16.7-4: Register **NTH_START** (0x18) Configure n-th start

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : nth_start - influences start event and middle event generation															

Table 3.3.1.2.5.16.7-5: Register **NTH_START_RELOAD** (0x1A) n-th start reload value

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : nth_start_reload - NTH_START register reload value															

Table 3.3.1.2.5.16.7-6: Register **CNT_MAX** (0x0C) current max counter register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : cnt_max - PWM period counting range configuration The allowed maximum value depends on the used counting mode: sawtooth waveform modes : (2^16)-1 triangle waveform modes : (2^15)-1															

Table 3.3.1.2.5.16.7-7: Register **CNT_MAX_RELOAD** (0x12) max counter reaload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : reload - CNT_MAX register reload value															

Table 3.3.1.2.5.16.7-8: Register **CNT** (0x06) counter register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : cnt - PWM period counter															

Table 3.3.1.2.5.16.7-9: Register **CNT_RESTART** (0x16) value of CNT

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : cnt_restart - value of CNT at the last time a restart command was issued															

Table 3.3.1.2.5.16.7-10: Register **PWM_PHASE** (0x08) PWM phase register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : pwm_phase - PWM phase signal value the coding of this value depends or the used counting mode: sawtooth waveform modes : equals CNT value triangle waveform modes : first half period : equals CNT value second half period : equals (CNT value & 0x7FFF) + 0x8000															

3.3.1.2.5.16.8 PWM channels

- PWMx_CFG (configuration) register:
  - used to configure the high-side (HS) and low-side (LS) waveform behavior
  - used to configure PRE_PWM module signals as register reload source

- PWMx_ON register:
  - used to set high-side (HS) and low-side (LS) PWM signal ON/OFF state
  - loaded from PWMx_ON_RELOAD at reload event

- PWMx_ON_RELOAD register:
  - PWMx_ON reload value

- PWMx_C0 and PWMx_C1 (compare value) registers:
  - used to configure PS signal LH and HL edge timestamps
  - loaded from PWMx_C0_RELOAD and PWMx_C1_RELOAD at reload event

- PWMx_C0_RELOAD and PWMx_C1_RELOAD registers:
  - PWMx_C0 and PWMx_C1 reload values

- PWMx_DEAD_TIME register:
  - used to configure PS signal LH and HL edge dead time values
  - loaded from DEAD_TIME_RELOAD_CH at reload event (depending on the DEAD_TIME_RELOAD_CH reload enable configuration)

- dead time counter (DT_CNT):
  - down counting with prescaled clock rate

- dead time events:
  - generated at PS signal LH and HL edge or after extension time
  - are interrupt sources
  - can be used by ADC control module as measurement trigger

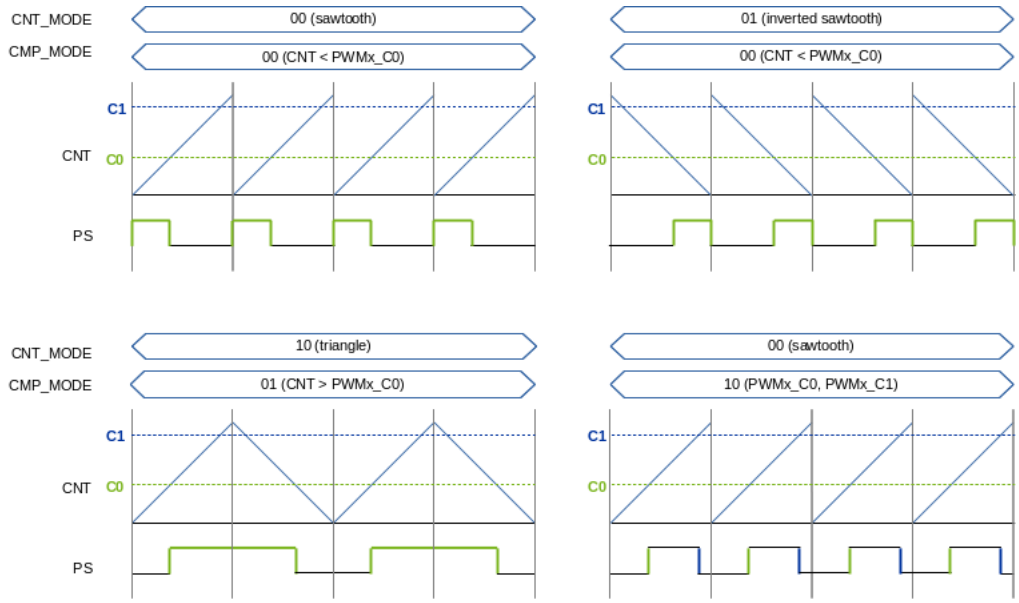


Figure 3.3.1.2.5.16.8-1: count mode and compare mode dependent PS waveform examples

Table 3.3.1.2.5.16.8-1: PWM channels

Register Name	Address	Description
PWM0_CFG	0x20	PWM0 config register
PWM0_C0	0x22	current PWM0 compare 0 register
PWM0_C1	0x24	current PWM0 compare 1 register
PWM0_ON	0x26	current PWM0 on register
PWM0_C0_RELOAD	0x28	PWM0 compare 0 reload register
PWM0_C1_RELOAD	0x2A	PWM0 compare 1 reload register
PWM0_ON_RELOAD	0x2C	PWM0 on reload register
PWM0_DEAD_TIME	0x2E	PWM0 dead time config register
PWM1_CFG	0x30	PWM1 config register
PWM1_C0	0x32	current PWM1 compare 0 register
PWM1_C1	0x34	current PWM1 compare 1 register
PWM1_ON	0x36	current PWM1 on register
PWM1_C0_RELOAD	0x38	PWM1 compare 0 reload register
PWM1_C1_RELOAD	0x3A	PWM1 compare 1 reload register
PWM1_ON_RELOAD	0x3C	PWM1 on reload register
PWM1_DEAD_TIME	0x3E	PWM1 dead time config register
PWM2_CFG	0x40	PWM2 config register
PWM2_C0	0x42	current PWM2 compare 0 register
PWM2_C1	0x44	current PWM2 compare 1 register
PWM2_ON	0x46	current PWM2 on register
PWM2_C0_RELOAD	0x48	PWM2 compare 0 reload register

Register Name	Address	Description
PWM2_C1_RELOAD	0x4A	PWM2 compare 1 reload register
PWM2_ON_RELOAD	0x4C	PWM2 on reload register
PWM2_DEAD_TIME	0x4E	PWM2 dead time config register
PWM3_CFG	0x50	PWM3 config register
PWM3_C0	0x52	current PWM3 compare 0 register
PWM3_C1	0x54	current PWM3 compare 1 register
PWM3_ON	0x56	current PWM3 on register
PWM3_C0_RELOAD	0x58	PWM3 compare 0 reload register
PWM3_C1_RELOAD	0x5A	PWM3 compare 1 reload register
PWM3_ON_RELOAD	0x5C	PWM3 on reload register
PWM3_DEAD_TIME	0x5E	PWM3 dead time config register

Table 3.3.1.2.5.16.8-2: Register **PWM0_CFG** (0x20) PWM0 config register

	MSB															*LSB*
Content	-	-	-	-	-	-	-	-	-	-	5	4	3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : ign_start_evt - Ignore start event if PWM0_CFG.cmp_mode == 10 0 : start event is used to set PS to 0 1 : start event does not influence PS signal state **4** : on_src - PWM0_ON_RELOAD register update source configuration The register can be configured by software in any case. 1 : PWM0_ON_RELOAD register can also be updated by PRE_PWM module **3:2** : c_src - PWM0_C0_RELOAD and PWM0_C1_RELOAD register update source configuration Both registers can be configured by software in any case. 00 : register update by software only 01 : PWM0_C0_RELOAD register can also be updated by PRE_PWM module 10 : PWM0_C1_RELOAD register can also be updated by PRE_PWM module **Note:** The combination 11 is invalid. **1:0** : cmp_mode - PS signal generation compare mode configuration 00 : if (CNT < PWM0_C0) PS = 1, else PS = 0 01 : if (CNT >= PWM0_C0) PS = 1, else PS = 0 10 : set-clear-mode when (CNT == PWM0_C0) PS is set to 1, else when (CNT == PWM0_C1) PS is set to 0, else when (start event) PS is set to 0															

Table 3.3.1.2.5.16.8-3: Register **PWM0_C0** (0x22) current PWM0 compare 0 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-4: Register **PWM0_C1** (0x24) current PWM0 compare 1 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-5: Register **PWM0_ON** (0x26) current PWM0 on register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - HS signal enable 0 : HS = 0 1 : HS PWM signal enabled **0** : ls - LS signal enable 0 : LS = 0 1 : LS PWM signal enabled															

Table 3.3.1.2.5.16.8-6: Register **PWM0_C0_RELOAD** (0x28) PWM0 compare 0 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM0_C0 register reload value															

Table 3.3.1.2.5.16.8-7: Register **PWM0_C1_RELOAD** (0x2A) PWM0 compare 1 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM0_C1 register reload value															

Table 3.3.1.2.5.16.8-8: Register **PWM0_ON_RELOAD** (0x2C) PWM0 on reload register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - PWM0_ON register hs reload value **0** : ls - PWM0_ON register ls reload value															

Table 3.3.1.2.5.16.8-9: Register **PWM0_DEAD_TIME** (0x2E) PWM0 dead time config register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : low_to_high - dead time inserted at PS (LH) 0 : no dead time inserted between LS and HS signal edges **7:0** : high_to_low - dead time inserted at PS (HL) 0 : no dead time inserted between LS and HS signal edges															

Table 3.3.1.2.5.16.8-10: Register **PWM1_CFG** (0x30) PWM1 config register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5	4	3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : ign_start_evt - Ignore start event if PWM1_CFG.cmp_mode == 10 0 : start event is used to set PS to 0 1 : start event does not influence PS signal state **4** : on_src - PWM1_ON_RELOAD register update source configuration The register can be configured by software in any case. 1 : PWM1_ON_RELOAD register can also be updated by PRE_PWM module **3:2** : c_src - PWM1_C0_RELOAD and PWM1_C1_RELOAD register update source configuration Both registers can be configured by software in any case. 00 : register update by software only 01 : PWM1_C0_RELOAD register can also be updated by PRE_PWM module 10 : PWM1_C1_RELOAD register can also be updated by PRE_PWM module **Note:** The combination 11 is invalid. **1:0** : cmp_mode - PS signal generation compare mode configuration 00 : if (CNT < PWM1_C0) PS = 1, else PS = 0 01 : if (CNT >= PWM1_C0) PS = 1, else PS = 0 10 : set-clear-mode when (CNT == PWM1_C0) PS is set to 1, else when (CNT == PWM1_C1) PS is set to 0, else when (start event) PS is set to 0															

Table 3.3.1.2.5.16.8-11: Register **PWM1_C0** (0x32) current PWM1 compare 0 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-12: Register **PWM1_C1** (0x34) current PWM1 compare 1 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-13: Register **PWM1_ON** (0x36) current PWM1 on register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - HS signal enable 0 : HS = 0 1 : HS PWM signal enabled **0** : ls - LS signal enable 0 : LS = 0 1 : LS PWM signal enabled															

Table 3.3.1.2.5.16.8-14: Register **PWM1_C0_RELOAD** (0x38) PWM1 compare 0 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM1_C0 register reload value															

Table 3.3.1.2.5.16.8-15: Register **PWM1_C1_RELOAD** (0x3A) PWM1 compare 1 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM1_C1 register reload value															

Table 3.3.1.2.5.16.8-16: Register **PWM1_ON_RELOAD** (0x3C) PWM1 on reload register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - PWM1_ON register hs reload value **0** : ls - PWM1_ON register ls reload value															

Table 3.3.1.2.5.16.8-17: Register **PWM1_DEAD_TIME** (0x3E) PWM1 dead time config register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : low_to_high - dead time inserted at PS (LH) 0 : no dead time inserted between LS and HS signal edges **7:0** : high_to_low - dead time inserted at PS (HL) 0 : no dead time inserted between LS and HS signal edges															

Table 3.3.1.2.5.16.8-18: Register **PWM2_CFG** (0x40) PWM2 config register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5	4	3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : ign_start_evt - Ignore start event if PWM2_CFG.cmp_mode == 10 0 : start event is used to set PS to 0 1 : start event does not influence PS signal state **4** : on_src - PWM2_ON_RELOAD register update source configuration The register can be configured by software in any case. 1 : PWM2_ON_RELOAD register can also be updated by PRE_PWM module **3:2** : c_src - PWM2_C0_RELOAD and PWM2_C1_RELOAD register update source configuration Both registers can be configured by software in any case. 00 : register update by software only 01 : PWM2_C0_RELOAD register can also be updated by PRE_PWM module 10 : PWM2_C1_RELOAD register can also be updated by PRE_PWM module **Note:** The combination 11 is invalid. **1:0** : cmp_mode - PS signal generation compare mode configuration 00 : if (CNT < PWM2_C0) PS = 1, else PS = 0 01 : if (CNT >= PWM2_C0) PS = 1, else PS = 0 10 : set-clear-mode when (CNT == PWM2_C0) PS is set to 1, else when (CNT == PWM2_C1) PS is set to 0, else when (start event) PS is set to 0															

Table 3.3.1.2.5.16.8-19: Register **PWM2_C0** (0x42) current PWM2 compare 0 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-20: Register **PWM2_C1** (0x44) current PWM2 compare 1 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-21: Register **PWM2_ON** (0x46) current PWM2 on register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - HS signal enable 0 : HS = 0 1 : HS PWM signal enabled **0** : ls - LS signal enable 0 : LS = 0 1 : LS PWM signal enabled															

Table 3.3.1.2.5.16.8-22: Register **PWM2_C0_RELOAD** (0x48) PWM2 compare 0 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM2_C0 register reload value															

Table 3.3.1.2.5.16.8-23: Register **PWM2_C1_RELOAD** (0x4A) PWM2 compare 1 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM2_C1 register reload value															

Table 3.3.1.2.5.16.8-24: Register **PWM2_ON_RELOAD** (0x4C) PWM2 on reload register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - PWM2_ON register hs reload value **0** : ls - PWM2_ON register ls reload value															

Table 3.3.1.2.5.16.8-25: Register **PWM2_DEAD_TIME** (0x4E) PWM2 dead time config register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : low_to_high - dead time inserted at PS (LH) 0 : no dead time inserted between LS and HS signal edges **7:0** : high_to_low - dead time inserted at PS (HL) 0 : no dead time inserted between LS and HS signal edges															

Table 3.3.1.2.5.16.8-26: Register **PWM3_CFG** (0x50) PWM3 config register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	5	4	3:2		1:0	
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**5** : ign_start_evt - Ignore start event if PWM3_CFG.cmp_mode == 10 0 : start event is used to set PS to 0 1 : start event does not influence PS signal state **4** : on_src - PWM3_ON_RELOAD register update source configuration The register can be configured by software in any case. 1 : PWM3_ON_RELOAD register can also be updated by PRE_PWM module **3:2** : c_src - PWM3_C0_RELOAD and PWM3_C1_RELOAD register update source configuration Both registers can be configured by software in any case. 00 : register update by software only 01 : PWM3_C0_RELOAD register can also be updated by PRE_PWM module 10 : PWM3_C1_RELOAD register can also be updated by PRE_PWM module **Note:** The combination 11 is invalid. **1:0** : cmp_mode - PS signal generation compare mode configuration 00 : if (CNT < PWM3_C0) PS = 1, else PS = 0 01 : if (CNT >= PWM3_C0) PS = 1, else PS = 0 10 : set-clear-mode when (CNT == PWM3_C0) PS is set to 1, else when (CNT == PWM3_C1) PS is set to 0, else when (start event) PS is set to 0															

Table 3.3.1.2.5.16.8-27: Register **PWM3_C0** (0x52) current PWM3 compare 0 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-28: Register **PWM3_C1** (0x54) current PWM3 compare 1 register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PS waveform generation compare values please see CFG.cmp_mode for details															

Table 3.3.1.2.5.16.8-29: Register **PWM3_ON** (0x56) current PWM3 on register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - HS signal enable 0 : HS = 0 1 : HS PWM signal enabled **0** : ls - LS signal enable 0 : LS = 0 1 : LS PWM signal enabled															

Table 3.3.1.2.5.16.8-30: Register **PWM3_C0_RELOAD** (0x58) PWM3 compare 0 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM3_C0 register reload value															

Table 3.3.1.2.5.16.8-31: Register **PWM3_C1_RELOAD** (0x5A) PWM3 compare 1 reload register

	MSB															***LSB***
Content	15:0															
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:0** : pwm_c - PWM3_C1 register reload value															

Table 3.3.1.2.5.16.8-32: Register **PWM3_ON_RELOAD** (0x5C) PWM3 on reload register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit Description	**1** : hs - PWM3_ON register hs reload value **0** : ls - PWM3_ON register ls reload value															

Table 3.3.1.2.5.16.8-33: Register **PWM3_DEAD_TIME** (0x5E) PWM3 dead time config register

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15:8** : low_to_high - dead time inserted at PS (LH) 0 : no dead time inserted between LS and HS signal edges **7:0** : high_to_low - dead time inserted at PS (HL) 0 : no dead time inserted between LS and HS signal edges															

3.3.1.2.5.16.9 PRE_PWM module connection

Depending on the PWM channel configuration (PWMx_CFG), the PWM channels 0, 1 and 2 can be supplied with reload values for PWMx_C0_RELOAD, PWMx_C1_RELOAD and PWMx_ON_RELOAD registers from PRE_PWM module.

Please see PRE_PWM module description for details on reload value generation.

3.3.1.2.5.16.10 Common dead time registers

- DEAD_TIME register:
  - is a write only register
  - a write access sets all channel PWMx_DEAD_TIME register LH and HL values to the same dead time value

- DEAD_TIME_RELOAD_CH register:
  - common dead time value
  - 4x2 reload enable bits to configure which channel dead times will be reloaded with the common dead time value

- DEAD_TIME_RELOAD register:
  - a write access behaves like a write access to the DEAD_TIME_RELOAD_CH register with all reload enable bits set to 1

- **Note:** There is a value restriction, configuring DT_LH and DT_HL dead times:
  - It's not allowed to set DT_LH to zero and DT_HL to a non-zero value or vice-versa.
  - For this reason all FSM conditions only refer to DT which means both DT_LH and DT_HL.

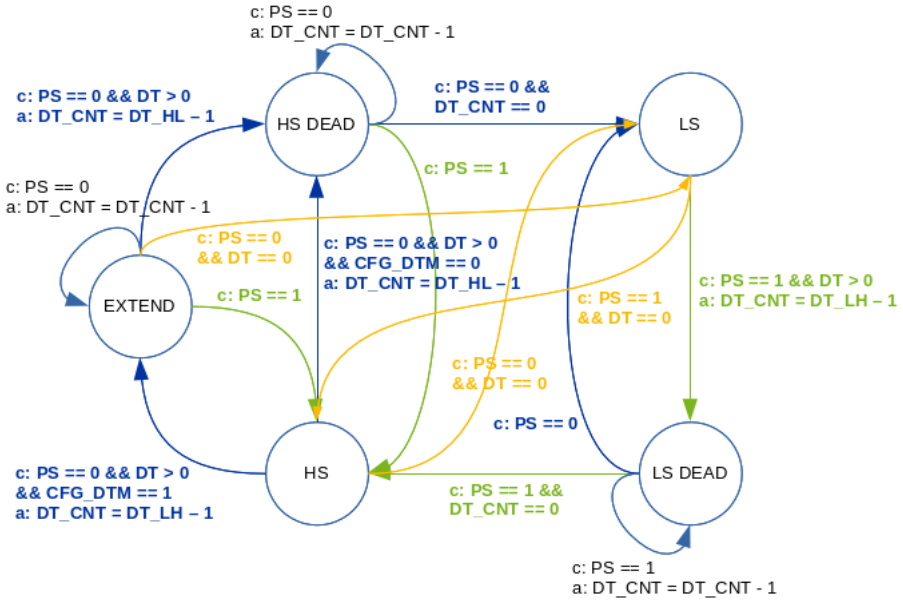


Figure 3.3.1.2.5.16.10-1: dead time insertion state machine

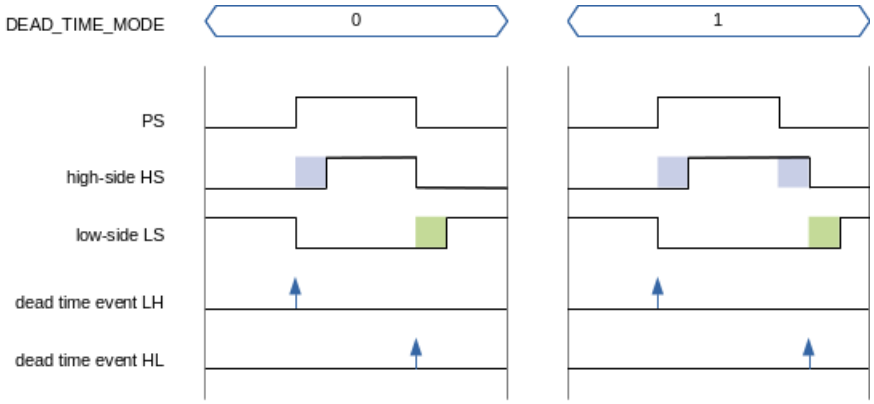


Figure 3.3.1.2.5.16.10-2: dead time modes

Table 3.3.1.2.5.16.10-1: Common dead time registers

Register Name	*Address*	*Description*
DEAD_TIME	0x0E	current dead time register
DEAD_TIME_RELOAD	0x14	dead time reload register
DEAD_TIME_RELOAD_CH	0x1E	dead time reload config register

Table 3.3.1.2.5.16.10-2: Register **DEAD_TIME** (0x0E) current dead time register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : dead_time - common dead time value a write access sets both LH and HL dead time values of all PWM channels (PWMx_DEAD_TIME) to a common value															

Table 3.3.1.2.5.16.10-3: Register **DEAD_TIME_RELOAD** (0x14) dead time reload register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7:0** : reload - common dead time value reload value a write access behaves like a write access to DEAD_TIME_RELOAD_CH with all reload enable bits set to 1															

Table 3.3.1.2.5.16.10-4: Register **DEAD_TIME_RELOAD_CH** (0x1E) dead time reload config register

	MSB															***LSB***
Content	15	14	13	12	11	10	9	8	7:0							
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**15** : up_ch3_lh - PWM channel 3 DT_LH dead time reload enable 0 : reload disabled, 1 : reload enabled **14** : up_ch3_hl - PWM channel 3 DT_HL dead time reload enable 0 : reload disabled, 1 : reload enabled **13** : up_ch2_lh - PWM channel 2 DT_LH dead time reload enable 0 : reload disabled, 1 : reload enabled **12** : up_ch2_hl - PWM channel 2 DT_HL dead time reload enable 0 : reload disabled, 1 : reload enabled **11** : up_ch1_lh - PWM channel 1 DT_LH dead time reload enable 0 : reload disabled, 1 : reload enabled **10** : up_ch1_hl - PWM channel 1 DT_HL dead time reload enable 0 : reload disabled, 1 : reload enabled **9** : up_ch0_lh - PWM channel 0 DT_LH dead time reload enable 0 : reload disabled, 1 : reload enabled **8** : up_ch0_hl - PWM channel 0 DT_HL dead time reload enable 0 : reload disabled, 1 : reload enabled **7:0** : reload - common dead time reload value 0 : no dead time inserted between LS and HS signal edges															

3.3.1.2.5.16.11 Interrupt handling registers

Table 3.3.1.2.5.16.11-1: Interrupt handling registers

Register Name	Address	Description
IRQ_STATUS	0x70	IRQ status register
IRQ_MASK	0x74	IRQ mask register
IRQ_VENABLE	0x78	IRQ vector enable register
IRQ_VDISABLE	0x7A	IRQ vector disable register
IRQ_VMAX	0x7C	IRQ max vector register
IRQ_VNO	0x7E	IRQ vector number register

Table 3.3.1.2.5.16.11-2: Register **IRQ_STATUS** (0x70) IRQ status register

	MSB															LSB
Content	-	-	-	-	-	10	9	8	7	6	5	4	3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**10** : dead_time_evt_7 - PWM channel 3 DT_LH dead time event **9** : dead_time_evt_6 - PWM channel 3 DT_HL dead time event **8** : dead_time_evt_5 - PWM channel 2 DT_LH dead time event **7** : dead_time_evt_4 - PWM channel 2 DT_HL dead time event **6** : dead_time_evt_3 - PWM channel 1 DT_LH dead time event **5** : dead_time_evt_2 - PWM channel 1 DT_HL dead time event **4** : dead_time_evt_1 - PWM channel 0 DT_LH dead time event **3** : dead_time_evt_0 - PWM channel 0 DT_HL dead time event **2** : middle_evt - middle event **1** : start_evt - Nth start event **0** : oc - overcurrent flag status															

Table 3.3.1.2.5.16.11-3: Register **IRQ_MASK** (0x74) IRQ mask register

	MSB															LSB
Content	-	-	-	-	-	10:0										
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**10:0** : mask - enable irq source 1: enabled 0: disabled															

Table 3.3.1.2.5.16.11-4: Register **IRQ_VENABLE** (0x78) IRQ vector enable register

	MSB															LSB
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to enable															

Table 3.3.1.2.5.16.11-5: Register **IRQ_VDISABLE** (0x7A) IRQ vector disable register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - vector number of interrupt to disable															

Table 3.3.1.2.5.16.11-6: Register **IRQ_VMAX** (0x7C) IRQ max vector register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vmax - needed for nested interrupt support software writes current vector number to this register, so only interrupts with higher priority (lower vector number) can nest															

Table 3.3.1.2.5.16.11-7: Register **IRQ_VNO** (0x7E) IRQ vector number register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	3:0			
Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
Access	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Bit Description	**3:0** : vno - read: vector number of enabled pending interrupt with highest priority (smallest vector number). when no irq is pending the first unused irq number is returned. write: vector number of interrupt event to clear															

4 Typical Applications

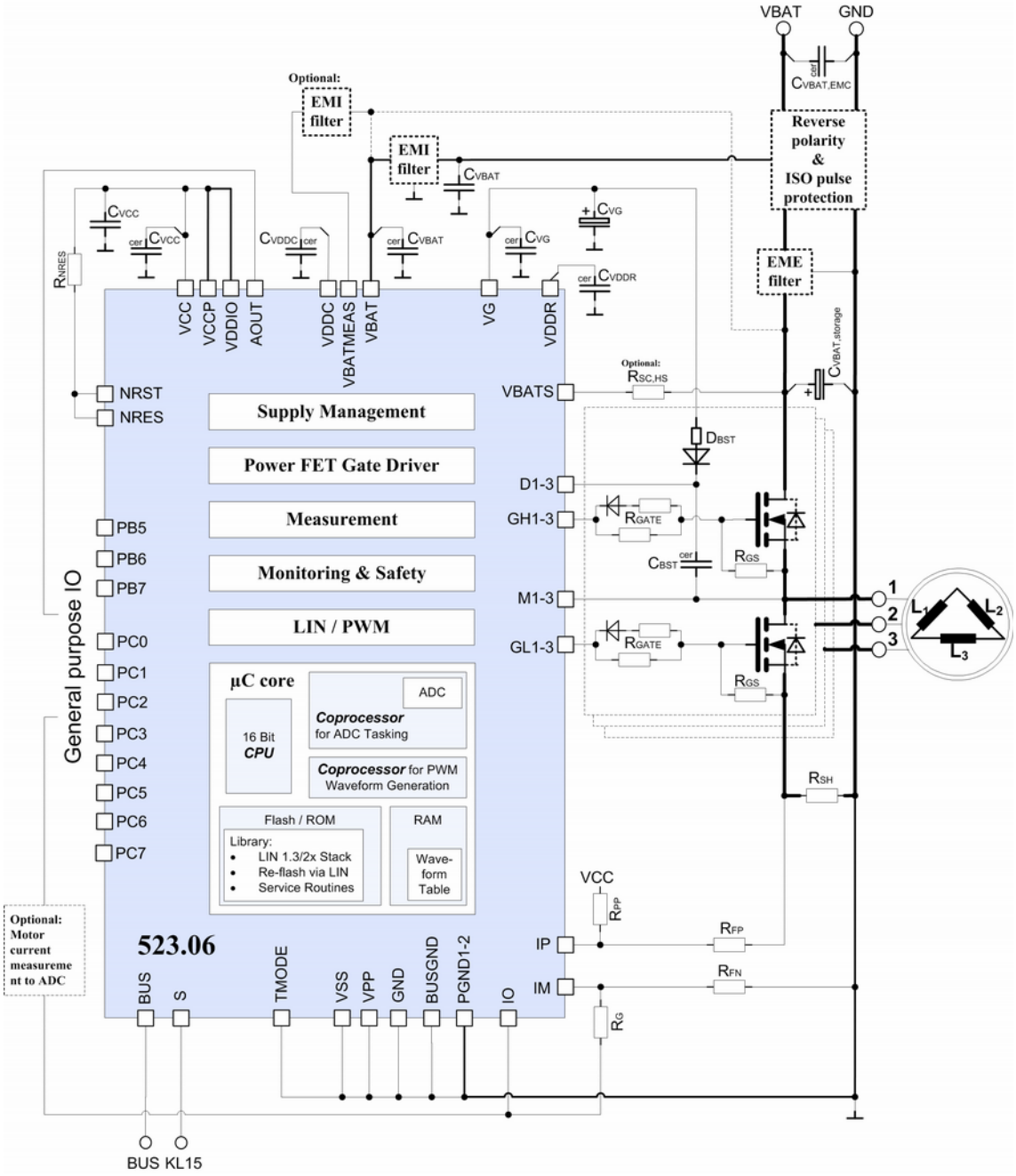


Figure 4-1: Typical Application Example

Table 4-1: External Components

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VBAT capacitance*) 1)		C_{VBAT}	33	47	-	µF
2	VBAT capacitance*) 1)	ceramic	C_{VBAT}	80	100	-	nF
3	Battery input capacitance, close to battery and ground PCB input*) 1)		$C_{VBAT,EMC}$	-	10	-	nF
4	Bridge supply storage capacitance*) 2)		$C_{VBAT,storage}$	-		-	mF
5	VG capacitance*) 1)		C_{VG}	8	10	50	µF
6	VG capacitance*) 1)	ceramic	C_{VG}	80	100	120	nF
7	VCC capacitance*) 3)		C_{VCC}	1		15	µF
8	VCC capacitance*)	ceramic	C_{VCC}	80	100	120	nF
9	VDDR capacitance*)	ceramic	C_{VDDR}	80	330	500	nF
10	NRES pull resistance*)		R_{NRES}	-	33	-	kΩ
11	Bootstrap capacitance*) 4)		C_{BST}	80	330		nF
12	Bootstrap diode*)	R_{ESR} > 3Ω (series resistance might be added alternatively), I_{peak} > 4A	D_{BST}	-	-	-	
13	Series gate resistances of power FETs*)		R_{Gate}	28.5	-	-	Ω
14	Power FET gate-source pull resistance*)		R_{GS}	-	100	-	kΩ
15	Capacitive gate charge of power FETs*)	f_{PWM} = 25kHz, 12V	Q_{FET}	-	-	250	nC
16	Capacitive gate charge of power FETs*)	f_{PWM} = 50kHz, 12V	Q_{FET}	-	-	100	nC
17	High side short circuit reference protection series resistance, comparator threshold is increased by $I_{SC,HS}$ * $R_{SC,HS}$*)	Case no reverse polarity protection at power FET stage	$R_{SC,HS}$	80	-	120	Ω
18	Current measurement amplifier resistances*) 5)		R_{SH}, R_{FP}, R_{FN}, R_{PP}, R_G	-		-	-
19	VDDC capacitance*)	ceramic	C_{DDC}	220-20%	270	4700+20%	nF

*) Not tested in production
1) For ISO pulse and load dump >50V required
2) Depending on application,
For ISO pulse and load dump >50V required
3) Depending on load transients
4) Depending on C_{VG} and Q_{FET} (C_{FET} << C_{BST} << C_{VG})
5) Amplifier to be designed by customer

5 ESD, Latchup and EMC

5.1 Electro Static Discharge (ESD)

Table 5.1-1: ESD on IC Level, Human Body Model (HBM)

Standard	AEC-Q100-002
Model	Human Body Model
Capacitance	100 pF
Resistance	1,5 kΩ
Minimum withstand Voltage	+/- 2 kV

Table 5.1-2: ESD Test on IC Level for Special Pins

Standard	AEC-Q100-002
Test point	Pin S, VIN, VBAT to system ground
Capacitance	100 pF
Resistance	1,5 kΩ
Minimum withstand Voltage	+/- 4 kV

Table 5.1-3: ESD Test on IC Level for Special Pins

Standard	AEC-Q100-002
Test point	Pin BUS to system ground
Capacitance	100 pF
Resistance	1,5 kΩ
Minimum withstand Voltage	+/- 8 kV

Table 5.1-4: ESD on IC Level, Charged Device Model (CDM)

Standard	AEC-Q100-011
Model	Charged Device Model
Resistance	1 Ω
Minimum withstand Voltage	+/- 750 V for edge pins
	+/- 500 V for all other pins
Pulse rise time (10%-90%)	<400 ps

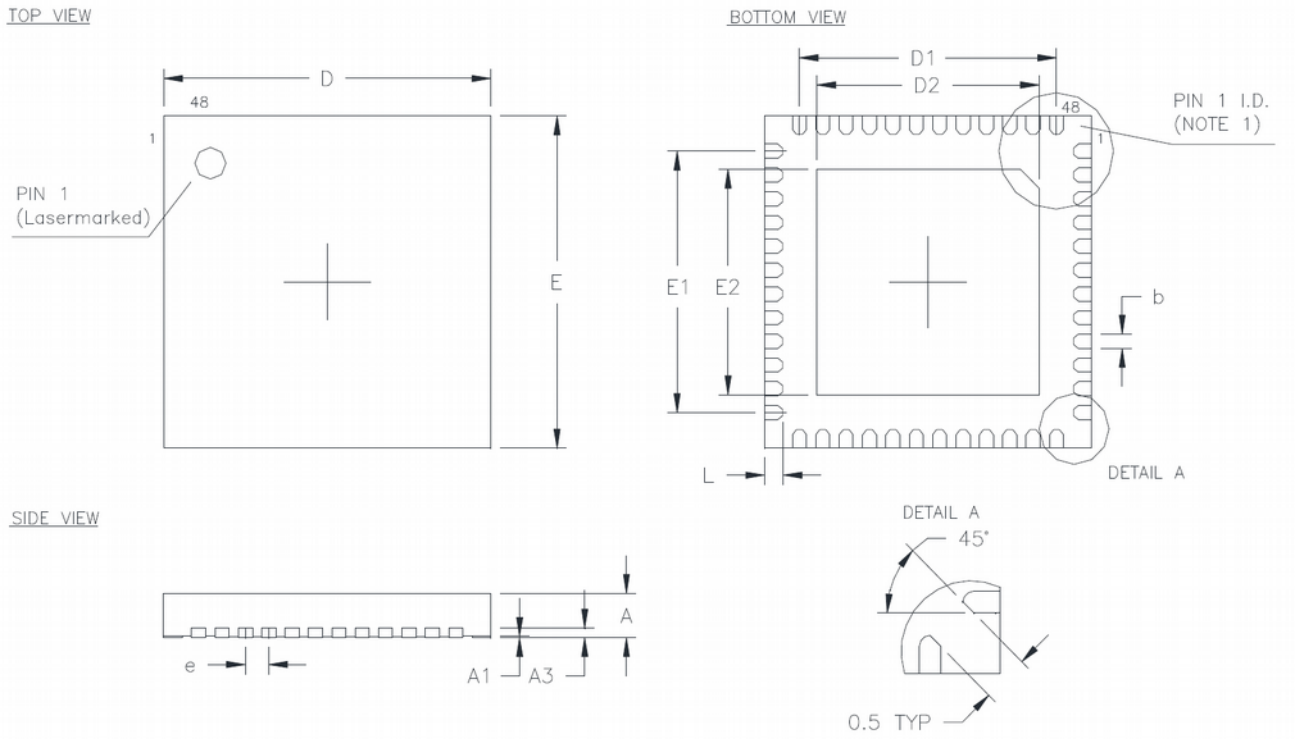
5.2 Latch-up

Latch-up performance is validated according JEDEC standard JESD 78 in its valid revision.

6 Package Reference

QFN48L7

Package outline and dimensions are according JEDEC MO-220 K, variant VKKD-6, except reduced terminal length to 0.4mm.



Description	Symbol	mm			inch		
		min	typ	max	min	typ	max
Package height	A	0.80	0.90	1.00	0.031	0.035	0.039
Stand off	A1	0.00	0.02	0.05	0.000	0.0008	0.002
Thickness of terminal leads, including lead finish	A3		0.20 REF			0.0079 REF	
Width of terminal leads	b	0.18	0.25	0.30	0.007	0.010	0.012
Package length / width	D / E	6.90	7.00	7.10	0.272	0.276	0.280
Distance between the two outer most terminals of one side (center to center)	D1 / E1		5.50 REF			0.217 REF	
Length / width of exposed pad	D2 / E2	4.60	4.75	4.90	0.181	0.187	0.193
Lead pitch	e		0.50 BSC			0.020 BSC	
Length of terminal for soldering to substrate	L	0.35	0.40	0.45	0.014	0.016	0.018
Number of terminal positions	N		48			48	

Figure 6-1: Package Outline

Note: The mm values are valid, the inch values contains rounding errors
Note: For assembler specific pin1 identification please see QM-document 08SP0363.xx (Pin 1 Specification)

7 Marking

7.1 Top Side

Table 7.1-1: Top Side

Elmos Logo
52306A
XXXXPZ
YWW * #

Table 7.1-2: Marking of the Devices

Signature	*Explanation*
E or " " / M / T	Volume production / prototype / test circuit
52306	ELMOS project number
A	ELMOS project revision code
XXXX	Production lot number
P	Assembler code
Z	Optional Elmos internal code
YWW	Year and week of assembly
*	Elmos internal code
#	Elmos internal code

7.2 Bottom Side

No marking.

8 Revision History

Table 8-1: Table of Revisions

Rev.	Chapter	Description of change	Changed by	Date
00	All	Initial version Describes sample version M523.06A with E523.99 controller and E523.07 driver	KFH	10.03.2017
01	All	Describes (pre)series samples E523.06A with E523.98 controller and E523.07 driver.	HHAU / KFH	20.09.2017
02	Measurement Functions / Fast Motor Current Measurement Amplifier Table 2.1.1.4.1-1	$V_{offset,Amp}$ conditions update	HHAU	14.3.2018
02	Communication Interfaces / BUS Interface Table 2.1.1.6.1-2	$V_{LIN,CLAMP}$ parameter removed	HHAU	14.3.2018
02	Pin Clamping Illustration Page 5	Figure update	HHAU	14.3.2018
02	Microcontroller / Analog Part / Electrical parameters Table 2.2.1.1-1	$I_{DD48MHz}$ removed	HHAU	16.3.2018
02	Electrical Characteristics Chapter 2	Initial remark update	HHAU	16.3.2018
02	Typical Applications / External Components Table 4-1	Name VDD changed to VDDR, R_{INTN} removed, C_{DDC} condition added	HHAU	26.3.2018
02	General Description Page 1	Product ID updated	ACO	29.03.2018
02	Fast Motor Current Measurement Amplifier Table 2.1.1.4.1-1	removed parameter toffset,settle removed tested at 25°C from Ileak,in	ACO	29.03.2018
02	Electrical Characteristics Chapter 2	updated remark for chapter	ACO	29.03.2018
02	Communication Interfaces Table 2.1.1.6-1	removed table: Digital Pins Electrical Parameters	ACO	29.03.2018

Rev.	Chapter	Description of change	Changed by	Date
02	Oscillators and Reset Table 2.2.1.1.2.3-1	Removed Parameter for 43MHz Removed Parameter Frequency temperature drift Condition Removed for Parameter F_{OSC_WDOG}	ACO	29.03.2018
02	SAR-ADC Table 2.2.1.1.3-1	Condition modified for Parameter V_{REFH} Removed Parameters C_{IN} and R_{IN}	ACO	29.03.2018
02	IO Port Characteristics Table 2.2.1.1.5-1	Change Limits for Parameter R_{PD}	ACO	29.03.2018
02	All Tables 2.1.1.3-1, 2.1.1.4.1-1	tbd's removed: minimal value for rG1-3,on and rG1-3,off; also removed customer flags measurement amplifier electrical parameters: Voffset,min, Voffset,max; inverted sign Voffset,buf,AOUT +/- 10mV	ACO	29.03.2018
02	Typical Applications / External Components Table 4-1	C_{BST} Footnote update	HHAU	03.04.2018
02	Fast Motor Current Measurement Amplifier Chapter 3.2.1.3.1	Note added	HHAU	03.04.2018
02	Analog Signal Measurement Table 3.2.1.3.2-2	Footnote added	HHAU	03.04.2018
02	Oscillators and Reset Table 2.2.1.1.2.3-1	change limits of $F_{OSC\_SYS\_48}$	ACO	03.04.2018
02	SAR-ADC Table 2.2.1.1.3-1	removed V_{REFH} automatic gain and offset compensation logic	ACO	03.04.2018
02	IO Port Characteristics Table 2.2.1.1.5-1	removed C_{IN}	ACO	06.04.2018
02	ADC Multiplexer Table 2.2.1.1.4-1	changed TC_{VT} value, not tested changed VT, not tested	ACO	06.04.2018
02	Absolute Maximum Ratings Table 1.1-1	removed V_{BUS} transients	ACO	06.04.2018
02	Recommended Operating Conditions Table 1.2-1	updated VBAT voltage	KJU	24.04.2018

9 General

9.1 WARNING - Life Support Applications Policy

ELMOS Semiconductor AG is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing ELMOS Semiconductor AG products, to observe standards of safety, and to avoid situations in which malfunction or failure of an ELMOS Semiconductor AG Product could cause loss of human life, body injury or damage to property. In development your designs, please ensure that ELMOS Semiconductor AG products are used within specified operating ranges as set forth in the most recent product specifications.

9.2 General Disclaimer

Information furnished by ELMOS Semiconductor AG is believed to be accurate and reliable. However, no responsibility is assumed by ELMOS Semiconductor AG for its use, nor for any infringements of patents or other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of ELMOS Semiconductor AG. ELMOS Semiconductor AG reserves the right to make changes to this document or the products contained therein without prior notice, to improve performance, reliability, or manufacturability .

9.3 Application Disclaimer

Circuit diagrams may contain components not manufactured by ELMOS Semiconductor AG, which are included as means of illustrating typical applications. Consequently, complete information sufficient for construction purposes is not necessarily given. The information in the application examples has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of ELMOS Semiconductor AG or others.

10 Storage, Handling, Packing and Shipping

10.1 Storage

Storage conditions should not exceed those given in Chapter 1.1 Absolute Maximum Ratings.
The Moisture Sensitivity Level is specified according to MSL3 (JEDEC J-STD-020 in its valid revision).

10.2 Handling

Devices are sensitive to damage by Electrostatic Discharge (ESD) and should only be handled at an ESD protected workstation.

Handling conditions should not exceed those given in Chapter 1.1 Absolute Maximum Ratings.

10.3 Packing

Material shall be packed for shipment as follows:

- Tape-on-Reel
- Drybag for MPC samples
- Every reel respectively drybag will be packed in a packing carton. Each packing carton will be marked and sealed by using the standard label for packings.

10.4 Shipping

Each delivery shall be accompanied by the following:

- Certificate of conformance to the specification
- Delivery note

11 Contact Info

Headquarters
Elmos Semiconductor AG
Heinrich-Hertz-Str. 1 • D-44227 Dortmund (Germany) ☏: +492317549100 ✉: sales-germany@elmos.com 🌍: www.elmos.com

Sales and Application Support Office North America
Elmos NA. Inc.
32255 Northwestern Highway • Suite 220 Farmington Hills
MI 48334 (USA) ☏: +12488653200 ✉: sales-usa@elmos.com

Sales and Application Support Office China
Elmos Semiconductor Technology (Shanghai) Co., Ltd.
Unit 16B, 16F Zhao Feng World Trade Building,
No. 369 Jiang Su Road, Chang Ning District,
Shanghai, PR China, 200050 ☏: +86216219 7502 ✉: sales-china@elmos.com

Sales and Application Support Office Korea
Elmos Korea
Office: C-301, Innovalley, 253, Pangyo-ro,
Bundang-gu, Sungnam-si, Gyeonggi-do,
13486 Korea ☏: +8231 714-1131 ✉: sales-korea@elmos.com

Sales and Application Support Office Japan
Elmos Japan K.K.
Tamachi 16th Fujishima Bldg. 6F
4-13-4 Shiba, Minato-ku,
Tokyo, 108-0014, Japan ☏: +81334517101 ✉: sales-japan@elmos.com

Sales and Application Support Office Singapore
Elmos Semiconductor Singapore Pte Ltd.
3A International Business Park
#09-13 ICON@IBP • 609935 Singapore ☏: +65 6908 1261 ✉: sales-singapore@elmos.com

12 Contents

Table of Content

Illustration Index