72V Programmable Brushless Motor Controller with 16bit CPU E523.52

PRELIMINARY INFORMATION – Jan 18, 2017

RoHS compliant

Features

- Supply voltage range from 12V to 72V, operational down to 7V
- 11V, 100mA high efficiency DC-DC converter for gate supply and other loads
- Three complementary 200mA gate drivers with programmable dead time and protection features
- Integrated automotive 16Bit RISC processor with and hardware MAC and divider units
  - 32kByte Flash with ECC error protection
  - 16kByte ROM
  - 4 kByte SRAM, parity protected
- Self advancing half bridge PWM generators autonomously generate various commutation waveforms
- 1MS/s 12bit ADC autonomous sample sequence engine synchronized to PWM with direct memory access
- One SPI module
- 4 timer capture and compare units with QEI modes
- One high voltage enable, 9 general purpose IOs
- Thermally efficient 7x7mm 36pin QFN package
- AEC-Q-100 grade 0 qualified (T=150°C)

Brief functional description

The E523.52 is a programmable, high-voltage brushless motor controller for 24V..48V automotive applications, industrial applications and commercial vehicles. It integrates 3 half-bridge drivers, a 11V step down converter, two linear regulators, and a 16bit RISC microcontroller with 32kB Flash.
The DC-DC converter efficiently provides 11V for the six gate drivers, the internal linear regulators, and other loads such as external Hall sensors. Up to 9 configurable IO pins facilitate interfacing with the application and the outside world. All IOs can be sampled by a 12bit, 1MS/s ADC with direct memory access.
Three self advancing PWM generators with integrated dead time can implement various wave-forms. Fully integrated back-EMF channels allow sensor-less commutation. Automatic ADC triggering allows programming of complex commutation algorithms. The memory can be divided into protected and field programmable areas.
Internal temperature monitoring and a thermally efficient 36 Pin QFN package enable the driver section to operate close to its maximum junction temperature of 150°C.

Applications

- 48V automotive and commercial vehicle applications
- BLDC-Motors in industrial 24V to 72V applications

Ordering information

Product ID	Package	Ordering-No.
E523.52A	QFN36L7	E52352A279B

Typical Application

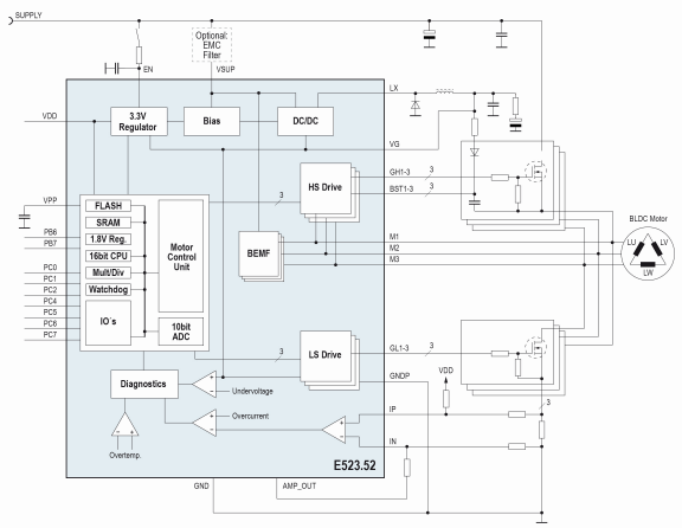

Figure 1: Typical application

1 Package and Pinout

1.1 Package Pinout

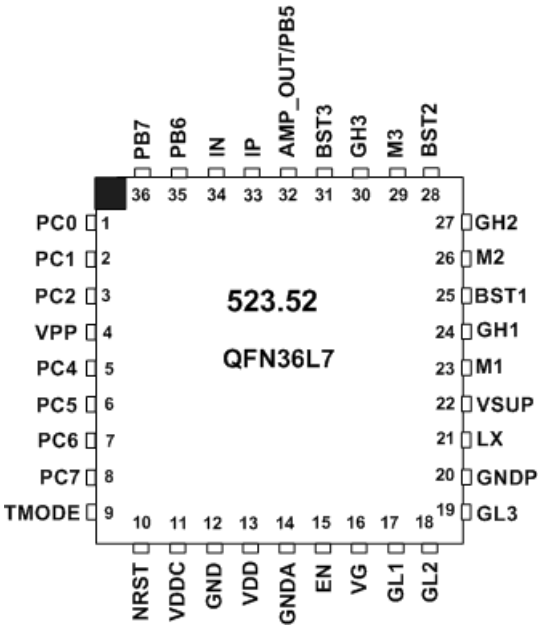

Figure 1.1-1: Package Pinout

1.2 Pin Description

Table 1.2-1: Pin Description

No	Name	Type	Description
1	PC0	AD_IO	DIO/AIN of µC core, JTAG TCK
2	PC1	AD_IO	DIO/AIN of µC core, JTAG TDA
3	PC2	AD_IO	DIO/AIN of µC core
4	VPP	HV_A_IO	FLASH test pad
5	PC4	AD_IO	DIO/AIN of µC core
6	PC5	AD_IO	DIO/AIN of µC core
7	PC6	AD_IO	DIO/AIN of µC core
8	PC7	AD_IO	DIO/AIN of µC core
9	TMODE	D_I	Test mode, to be connected to ground in application (pull-down)
10	NRST	D_IO	Reset for µC core (pull-up, bidirectional active low reset, open-drain driver)
11	VDDC	S	Core supply pad (internal regulator: 1.8V) of µC core
12	GND		Digital GND connection for the µC core
13	VDD	S	3.3V supply; connection of external capacitor
14	GNDA	S	Analog ground
15	EN	HV_D_I	Enable (high-voltage level)
16	VG	HV_A_IO	Feedback for DC-DC. DC-DC Converter output voltage, supply for gate drivers.
17	GL1	HV_D_O	Half bridge 1: gate driver output for lowside FET
18	GL2	HV_D_O	Half bridge 2: gate driver output for lowside FET

No	Name	Type	Description
19	GL3	HV_D_O	Half bridge 3: gate driver output for lowside FET
20	GNDP	S	Power ground
21	LX	HV_D_O	DC-DC switching output
22	VSUP	HV_S	Power supply voltage
23	M1	HV_A_IO	Motor phase 1
24	GH1	HV_D_O	Half bridge 1: gate driver output for highside FET
25	BST1	HV_S	Half bridge 1: supply pin for highside driver
26	M2	HV_A_IO	Motor phase 2
27	GH2	HV_D_O	Half bridge 2: gate driver output for highside FET
28	BST2	HV_S	Half bridge 2: supply pin for highside driver
29	M3	HV_A_IO	Motor phase 3
30	GH3	HV_D_O	Half bridge 3: gate driver output for highside FET
31	BST3	HV_S	Half bridge 3: supply pin for highside driver
32	AMP_OUT	A_O	Output of current sense amplifier
33	IP	A_I	Current amplifier positive input
34	IN	A_I	Current amplifier negative input
35	PB6	AD_IO	DIO/AIN of µC core
36	PB7	AD_IO	DIO/AIN of µC core
EP	EP		exposed die paddle, connect to GND

Explanation of Types:
A = Analog, D = Digital, S = Supply, I = Input, O = Output, IO = Bidirectional, HV = High Voltage

2 Block Diagram

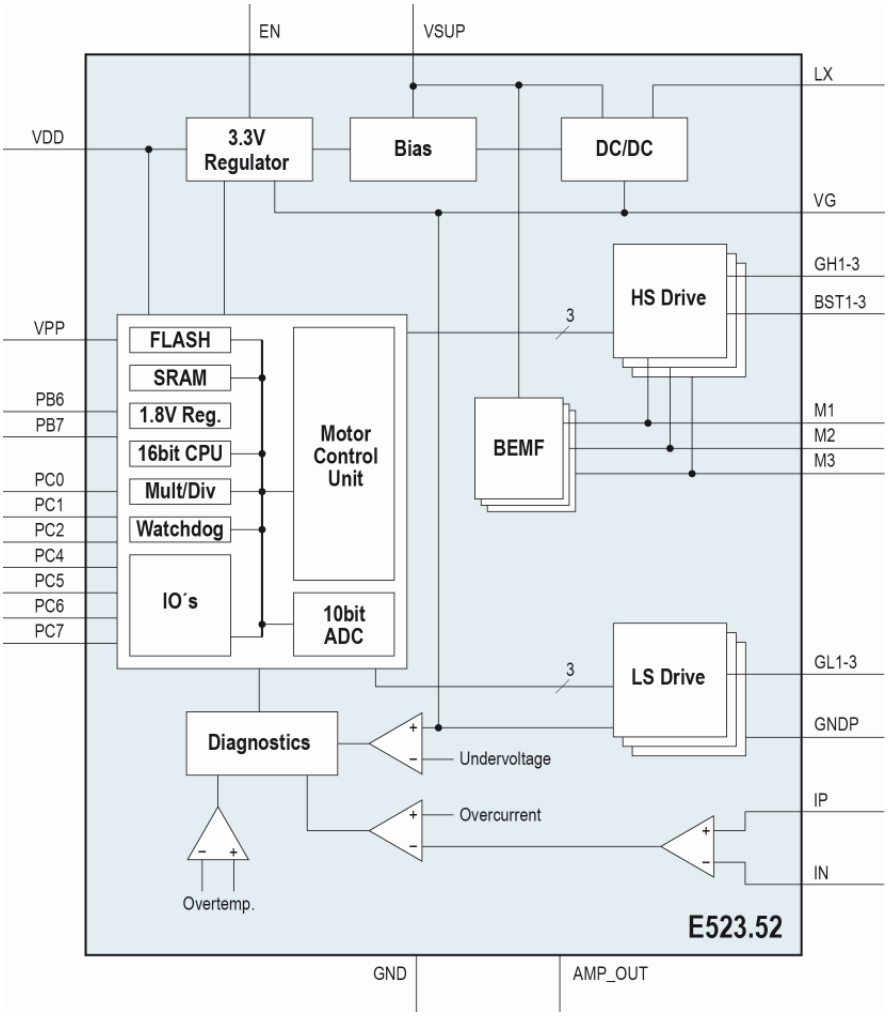

Figure 2-1: Block Diagram

3 Operating Conditions

3.1 Absolute Maximum Ratings

- 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 3.1-1: Absolute Maximum Ratings

No.	Description	Condition	Symbol	Min	Max	Unit
1	Storage Temperature		T_S	-40	150	℃
2	Storage temperature mission profile	T_s < 55℃	$t_{T,S,55}$	15		years
3	Storage temperature mission profile	T_s < 125℃	$t_{T,S,125}$	2000		h
4	Storage temperature mission profile	T_s ≤ 150℃	$t_{T,S,150}$	300		h
5	Junction Temperature		$T_{junction}$	-40	125	℃
6	Junction Temperature	Excluding Flash programming	$T_{junction}$	-40	150	℃
7	Junction temperature mission profile[1)]	T_J < 23℃	$t_{T,J,23}$	3120		h
8	Junction temperature mission profile[1)]	T_J < 100℃	$t_{T,J,100}$	7800		h
9	Junction temperature mission profile[1)]	T_J ≤ 150℃	$t_{T,J,150}$	1080		h
10	Thermal resistance junction to case		$R_{th,J-C,QFN}$	-	5	K/W
11	VSUP, EN voltage		V_{VSUP}	-0.3	72	V
12	VSUP, EN voltage	t<500ms	V_{VSUP}	-0.3	76	V
13	Ground shift between GNDP and GNDA		V_{GNDP}	-0.3	0.3	V
14	VG voltage		V_{VG}	-0.3	15	V
15	LX voltage		V_{LX}	-2.4	V_{VSUP}+0.3	V
16	VDD voltage		V_{VDD}	-0.3	3.6	V
17	BST[1-3] voltage		$V_{BST1-3,abs}$		87	V
18	BST[1-3] to M[1-3] voltage		$V_{BST1-3,rel}$	-0.3	15	V
19	GH1-3 voltage		V_{GH1-3}	V_{M1-3}-0.3V	V_{BST1-3}+0.3V	-
20	M[1-3] phase voltage		V_{M1-3}	-10V	V_{BATS}+2V	-
21	GL[1-3] voltage		V_{GL1-3}	-0.3V	V_{VG}+0.3V	-
22	IN, IP voltage		$V_{IN,IP}$	-0.3V	V_{VDD}+0.3V	-
23	AMP_OUT voltage		V_{AMP_OUT}	-0.3V	V_{VDD}+0.3V	-
24	AMP_OUT voltage		V_{AMP_OUT}	-0.3	3.6	V
25	AMP_OUT forced input current		I_{AMP_OUT}	-5	5	mA
26	IO pin voltage (PB6-7, PC0-2,PC4-7, NRST, TMODE)		V_{PORT}	-0.3	V_{VDD}+0.3	V
27	Total IO pin current (PB6-7, PC0-2,PC4-7)		$I_{PORTtotal}$		5	mA
28	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

3.2 Recommended Operating Conditions

- Parameters are guaranteed within the range of recommended operating conditions unless otherwise specified.
- All voltages are referred to ground (0V).
- Currents flowing into the circuit have positive values.
- The first electrical potential connected to the IC must be GND. (If not specified specify timing sequence of electrical contacts.)

Table 3.2-1: Recommended Operating Conditions

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VSUP supply voltage	RUN mode	V_{VSUP_run}	12	24	72	V
2	VSUP supply voltage	STANDBY & SLEEP mode	$V_{VSUP\_standby}$	7	24	72	V
3	DC-DC Buck Converter: LX inductance	V_{VSUP} = 24V	L_{LX}		150		µH
4	DC-DC Buck Converter: Series resistance of LX inductance		R_{LX}	1	5	10	Ω
5	DC-DC Buck Converter: Saturation current of LX inductance		$I_{SAT,LX}$	240			mA
6	DC-DC Buck Converter: VG capacitance		C_{VG}	26	33	100	µF
7	DC-DC Buck Converter: Ceramic capacitance at VG		C_{VG_cer}	10	22	33	nF
8	DC-DC Buck Converter: C_{VG} equivalent series resistance		$R_{ESR,CVG}$		1.8		Ω
9	DC-DC Buck Converter: Peak-to-Peak Ripple Voltage at VG for regulation		$V_{RIPPLE,VG}$	20		100	mV
10	DC-DC Buck Converter: Forward voltage of external diode at LX	I=100mA, T=25°C	$V_{F,DLX}$		0.6	0.8	V
11	DC-DC Buck Converter: Reverse recovery time of external diode at LX		$T_{REV,REC,DLX}$			25	ns
12	DC-DC Buck Converter: Reverse voltage of external diode at LX		$V_{REV,DLX}$	V_{VSUP}			
13	GNDP shift to GNDA		V_{GNDP}	-100		100	mV
14	Motor phase voltage		V_{M1-3}	-2V		VSUP +2V	
15	EN pin: LOW input voltage		V_{EN_LOW}	0		1.3	V
16	EN pin: HIGH input voltage		V_{EN_HIGH}	2.7		V_{VSUP}	V
17	Current Sense Amplifier: Input voltage at IN, IP		$V_{IN,IP}$	0		2.5	V
18	Current Sense Amplifier: AMP_OUT DC load current		I_{AMP_OUT}	-200		200	uA
19	Current Sense Amplifier: Load capacitance at AMP_OUT		C_{AMP_OUT}	0		40	pF
20	Current Sense Amplifier: Adjusted gain		gain	5		100	

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
21	Current Sense Amplifier: Feedback resistance from AMP_OUT to IN		R_{FB}	17,25			kOhm
22	Current Sense Amplifier: Common mode input range		CMRI	0		2.5	V

4 Detailed Electrical Specification

4.1 System Overview

4.1.1 System Current Consumption

Table 4.1.1-1: System Current Consumption Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Gate driver: RUN mode IC current consumption*)	T_{amb}= 25°C, No switching of gates	I_{VSUP_RUN}		2		mA
2	Gate driver: STANDBY mode IC current consumption*)	T_{amb}= 25°C	$I_{VSUP\_STANDBY}$		1.5		mA
3	Gate driver: SLEEP mode IC current consumption	T_{amb}= 25°C, V_{VSUP}=24V	$I_{VSUP\_SLEEP\_24}$			20	uA
4	Gate driver: SLEEP mode IC current consumption	T_{amb}= 25°C, V_{VSUP}=48V	I_{VSUP_SLEEP}			25	uA
5	Motor Control Unit: 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}			49.5	mA
6	Motor Control Unit: total supply current example $I_{DDIO} + I_{DDA} + I_{DDC} + I_{DDCA}$*)	f_{core} = 48MHz IO current = 0 CPU in standby (halt) state ADC active	$I_{DD\_STBY\_48MHz}$			24.7	mA

*) Not tested in production

4.2 Gate Driver

4.2.1 Power Supply

4.2.1.1 DC-DC Buck Converter

Table 4.2.1.1-1: DC-DC Converter: Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Buck converter output voltage	V_{VSUP} > 13 V; I_{VG} > -100 mA	V_{VG}	10	11	12	V
2	Buck converter output voltage standby mode*)	V_{VSUP} > 13 V; I_{VG} > -100 mA	$V_{VGstandby}$	8	11	12	V
3	Average available static load current at VG.*) 1)	V_{VG} > 90% of nominal value	I_{VG}			100	mA
4	Maximum avalable load current at VG during Start-Up scenario.*)		$I_{VG,START}$			50	mA
5	Over-current limitation		I_{LX}	160	200	240	mA
6	ON resistance of switch between VSUP and LX		R_{ON}		5	10	Ω

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
7	Maximum Switching frequency.*) 2)	15V<V_{VSUP}<72V; typical value for V_{VSUP}=24V; V_{VSUP} dependency is shown in 5.2.1.2-1	f_{SWITCH}	0.27	1.00	1.15	MHz
8	V_{VSUP} -V_{VLX} in case of 100% duty cycle	7V < V_{VSUP} < 11V I_{VG} = 75 mA	V_{DROP}			1	V
9	V_{VG_ext}. Over voltage protection for external components at open VG pin.	V_{VG} = open	$V_{LX,OV}$	15.8	17.4	19.0	V
10	Minimum off time in nominal mode.	Test Mode TM_DCDC_PFM	$t_{OFF,MIN,NOM}$		0.5		µs
11	Minimum on time in nominal mode.	Test Mode TM_DCDC_PFM	$t_{ON,MIN,NOM}$		0.5		µs

*) Not tested in production
1) Load current includes all loads on VG. (Including internal loads.)
2) Frequency if the inductor is in Continuos Current Mode, the V_{VG} is in specified range and no Over-current events are detected.

4.2.1.2 VDD Supply

Table 4.2.1.2-1: VDD Supply Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	VDD output voltage	I_{VDD} = -25 mA	V_{VDD}	3.2	3.3	3.4	V
2	VDD reset deactivation	VDD rising	V_{VDD_RSTD}			3.1	V

4.2.2 Power FET Gate Drivers

Table 4.2.2-1: Gate Drivers: Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Resistance of GH(1-3) to BST(1-3) in ON state of HS driver	V_{M1-3}= VSUP V_{BST}-V_M > 9V I_{GH1-3} = -200mA	R_{GH1-3_on}		10	20	Ohm
2	Resistance of GH(1-3) to M(1-3) in OFF state of HS driver	V_{M1-3}= GNDP V_{BST}-V_M > 9V I_{GH1-3} = 200mA	R_{GH1-3_off}		5	10	Ohm
3	Resistance of GL(1-3) to VG in ON-state of LS driver	I_{GL1-3} = -200mA	R_{GL1-3_on}		10	20	Ohm
4	Resistance of GL(1-3) to GNDP in OFF-state of LS driver	I_{GL1-3} = 200mA	R_{GL1-3_off}		5	10	Ohm
5	VDS cut-off threshold of high-side transistor (V_{VSUP}-V_{M1-3})	Default programming (SEL_VTH_SC=0)	$V_{DS\_off\_HS\_0}$		2.5		V
6	VDS cut-off threshold of low-side transistor (V_{M1-3}-V_{PGND})	Default programming (SEL_VTH_SC=0)	$V_{DS\_off\_LS\_0}$		2.5		V
7	Supply current of HS driver in ON state	GH(1-3) open HS driver in ON state	I_{BST1-3_on}		250	400	uA

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
8	Mask time of VDS cut-off signal after switch-on gate driver		$T_{mask\_VDS\_off}$		5		us
9	VDS desaturation detection debouncing time		$T_{deb\_VDS\_off}$	1.5	2.5	3.5	us

4.2.3 The Control Inputs

4.2.3.1 Enable Pin

Table 4.2.3.1-1: EN Pin Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	EN pin L/H threshold		V_{EN_LH}			2.6	V
2	EN pin H/L threshold		V_{EN_HL}	1.4			V
3	EN pin pull-down current		I_{EN_IN}		10		uA
4	EN pin debounce time		T_{DEB_EN}		100		us

4.2.3.2 Internal Control Ports

Table 4.2.3.2-1: Digital Inputs: Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	RUN debounce time before entering STANDBY or SLEEP mode		T_{RUN_DEB}		500		us

4.2.4 BEMF Detection

Table 4.2.4-1: Electrical Parameters of the BEMF Detection

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Voltage divider ratio for VS and high voltage divider ratio for M[1..3]		$V_{VS,M[1..3]}/V_{BEMF\_O}$	27.36	28.8	30.24	
2	Mismatch between the voltage divider ratios of M[1..3]		$\Delta V_{BEMF,Rel,Div}$	-2		2	%
3	Settling time of the BEMF detection*)		t_{settle}			1.5	µs
4	BEMF Amp output resistance		R_{BEMF_OUT}			200	Ω

*) Not tested in production

4.2.5 Current Sense Amplifier / Overcurrent Detection

Table 4.2.5-1: Current Sense Amplifier: Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	IP, IN input leakage current		I_{LEAK}	-1		1	µA
2	Current amplifier input offset voltage		V_{OFFSET}	-10		10	mV
3	Amplifier settling time *)	Gain = 20, R_{FB}= 18kΩ, V_{AMP_OUT} = 0.25...3.05 V	t_{AMP}		300	500	ns
4	AMP_OUT output resistance		R_{AMP_OUT}			200	Ω
5	AMP_OUT overcurrent threshold		$V_{AMP\_OUT,OC}$	0.85	0.9	0.95	VDD
6	Overcurrent debounce time		t_{OC}		10		µs

*) Not tested in production

4.2.6 Monitoring and Safety Functions

4.2.6.1 Temperature Monitoring and Over-Temperature Detection

Table 4.2.6.1-1: Temperature Monitoring Electrical Parameters

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	Overtemperature warning activation threshold*)		OT_{th_warn}	150		175	°C
2	Overtemperature shut-off threshold*)		$OT_{th\_switch\_off}$		OT_{th_warn} + 10K		

*) Not tested in production

4.3 Motor Control Unit

4.3.1 Analog Part

4.3.1.1 Core Supply Regulator

Table 4.3.1.1-1: 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

4.3.1.2 Oscillators and Reset

4.3.1.2.1 Power On Reset

Table 4.3.1.2.1-1: 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

4.3.1.2.2 Brownout Detection

Table 4.3.1.2.2-1: 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
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

4.3.1.2.3 System Clock RC Oscillator

Table 4.3.1.2.3-1: Oscillators: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	output frequency	calibrated, T = 25°C	F_{OSC_SYS}	47	48	49	MHz
2	frequency temperature drift*)		$TC_{F\_OSC\_SYS}$	-6.4	-	6.4	kHz/K

*) Not tested in production

4.3.1.2.4 Watchdog Clock RC Oscillator

Table 4.3.1.2.4-1: Oscillators: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	output frequency	T = 25°C	F_{OSC_WDOG}	0.6	0.8	1.0	MHz

4.3.1.2.5 NRST debouncer

Table 4.3.1.2.5-1: 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

4.3.1.3 SAR-ADC

Table 4.3.1.3-1: Electrical parameter table

No.	Description	Condition	Symbol	Min	Typ	Max	Unit
1	bandgap derived reference high voltage measured at ADC reference buffer output The gain error will be digitally eliminated by an automatic gain and offset compensation logic.	ADC V_{REFH} trimmed T = 25°C	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	input capacitance*)		C_{IN}	3.8	4.8	5.8	pF
7	ON resistance of sample switch*)		R_{IN}	-	-	300	Ω
8	Effective Number of Bits*)	VREFL < VIN < VREFH	ENOB	10	10.5	-	Bit
9	sampling time (number of ADC clock cycles) see ADC_CTRL.SAMPLE_EXT for additional information*)		$CYCLES_{SAMPLE}$	2	-	-	
10	conversion time (number of ADC clock cycles)*)		$CYCLES_{CONVERT}$	-	12	-	
11	ADC supply current when active	ADC ON	I_{ADC_SUPPLY}	-	3.0	3.5	mA
12	ADC warm-up time between ADC standby and run mode*)		$t_{WARM-UP}$	-	-	1	µs
13	ADC standby mode supply current	ADC in standby mode	$I_{ADC\_STANDBY}$	-	0.20	0.25	mA

*) Not tested in production

4.3.1.4 ADC Multiplexer

Table 4.3.1.4-1: 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	1.50	1.55	1.60	V
3	VT temperature coefficient		TC_{VT}	-	-3.3	-	mV/K

4.3.1.5 IO Port Characteristics

Table 4.3.1.5-1: 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	no pull down or pull up enabled	V_{TP}	-	-	2.0	V
3	Schmitt-Trigger High to Low Threshold Point	no pull down or pull up enabled	V_{TN}	0.8	-	-	V
4	Pull Up Resistor	pull up enabled V_{IN} = 0V	R_{PU}	30.00	48.00	74.00	kΩ
5	Pull Down Resistor	pull down enabled V_{IN} = 3.3V	R_{PD}	29.00	47.00	86.00	kΩ
6	pad input capacitance*)		C_{IN}	-	2.5	-	pF
7	Low Level Output Voltage with smaller (drv_strength=0) driver strength.	I = 4.0mA	V_{OL_4}	-	-	0.4	V
8	Low Level Output Voltage with higher (drv_strength=1) driver strength.	I = 8.0mA	V_{OL_8}	-	-	0.6	V
9	High Level Output Voltage with smaller (drv_strength=0) driver strength.	I = -4.0mA	V_{OH_4}	2.4	-	-	V
10	High Level Output Voltage with higher (drv_strength=1) driver strength.	I = -8.0mA	V_{OH_8}	2.4	-	-	V

*) Not tested in production

5 Functional Description

5.1 System Overview

The E523.52 is a programmable brush-less 3-Phase motor controller for supply voltages from 12V to 72V. The wide operating range makes it suitable for 48V automotive, industrial and consumer applications. Operation down to 7V in Standby Mode (drivers off) allows the E523.52 to be used in 24V commercial vehicles as well.

An integrated 11V PFM (pulse frequency mode) DC-DC buck converter provides efficient, reliable gate drive and can supply other loads (Hall Sensors) up to a total of 100mA current. It is possible to bypass the buck converter to reduce component cost.

The motor interface provides protection features such as under-voltage and short-circuit protection of the 6 Power FETs. An integrated current sense amp allows current control and over-current detection.

The IC incorporates an automotive rated 16bit RISC microcontroller with 32kByte of Flash Memory and 4kBytes of RAM allowing the implementation of various motor control algorithms. The following digital modules take the computational load off the CPU:

- 16x16 bit multiply and accumulate / divide
- 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

EN pin activates the power supplies and the microcontroller. Nine programmable IO pins provide various interface options with the motor and the outside world. 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 SUP.

Internal temperature monitoring and a thermally efficient QFN package enable the E523.52 to operate close to its maximum junction temperature.

5.1.1 Die to Die Interface

The E523.52 is a dual die combination of the E523.50 driver IC and the Elmos motor controller E523.99. The following figure shows how the two IC's are interconnected.

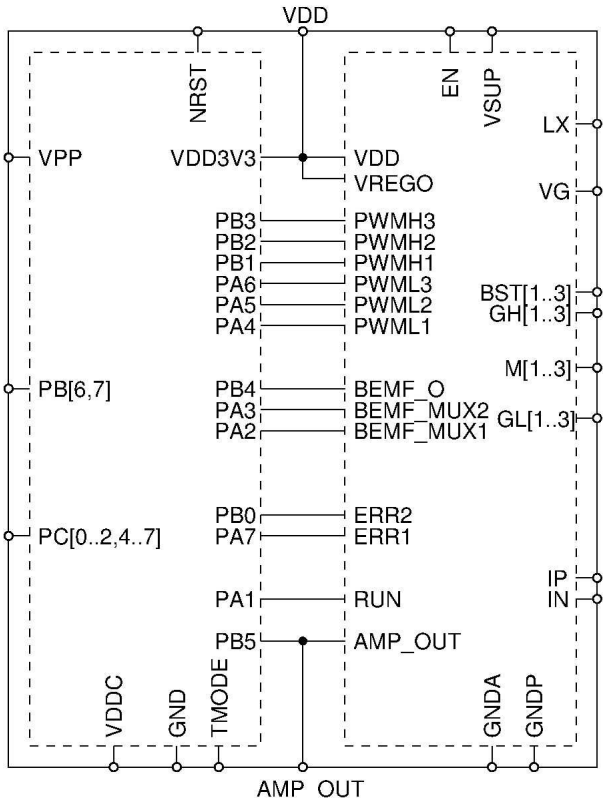

Figure 5.1.1-1: Internal Interconnects

The most important data signals are the six PWM lines connected to port B1/B2/B3 for the high-side gates and to port A4/A5/A6 for the low-side gates.

The CPU can activate the high-voltage gate driver by setting the "RUN" signal connected to Port A1 to "high". As long as this signal is at GND, the E523.50 will not switch the gates.

Since there is no SPI interface to read out the current status of the driver, there are two error signals "ERR1/ERR2" connected to "Port A7/B0", that notify of any abnormal operation.

The output of the integrated operational amplifier "AMP_OUT" is hard-wired to "Port B5" and can be measured by the SARADC-module.

The E523.50 has an internal multiplexer and an internal voltage divider, making it possible to measure the three phase voltages and the supply voltage directly as well as sample the status of then enable pin. With the help of the two control signals "BMUX1/2", connected to "Port A2/A3", the multiplexer of the E523.50 switches one of the motor-phases U,V or W to the output "BEMF_O". This output is connected to "Port B4" and can be measured by the SARADC of the CPU.

5.1.2 System Power Consumption

For the power loss estimation of the IC, several blocks have to be considered. The table gives an overview about the different blocks. P_{tot} is total power loss of the system.

Table 5.1.2-1: Overview of system power consumption

Block	Symbol	Calculation
Internal	P_{int}	$V_{VBAT} \cdot 200\mu A$
DC/DC converter	$P_{DC/DC}$	$V_{VBAT} \cdot 300\mu A + P_{LOSS,DCDC}$
VDD regulator	P_{VDD}	$(V_G - V_{DD}) * I_{SUP,VDD}$
MCU	P_{MCU}	$V_{VDD} \cdot I_{DD,48Mhz}$
Power Gate Driver	P_{PFET}	$V_G \cdot 1mA + 6 \cdot V_G \cdot Q_{PFET} \cdot f_{PWM} \cdot \frac{R_{G,on}}{R_{charge,ext} + R_{G,on}} + \frac{R_{G,off}}{R_{discharge,ext} + R_{G,off}}$
IC	P_{tot}	$P_{int} + P_{DC/DC} + P_{VDD} + P_{MCU} + P_{PFET}$
	$I_{SUP,VDD}$	$I_{DD,48Mhz} + 5mA$

The switching loss ($P_{LOSS,DCDC}$)of the DC/DC is given in the following figure. The power loss depends on the V_{bat} and is calculated with the worst case assumption of 100mA as DC current at VG.

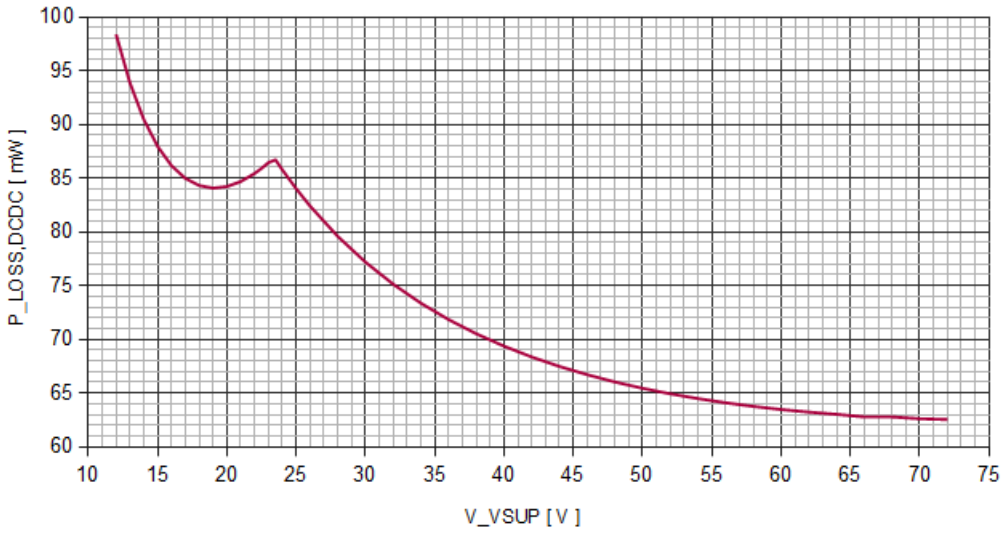

Figure 5.1.2-1: Power dissipation of DC/DC converter versus Vbat

5.1.3 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.
The remainder of this datasheet will explain the general capabilities of the IC to be used as a foundation upon which software algorithms can be based.

5.1.3.1 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 5.1.3.1-1: JTAG interace connection

JTAG Line	*IC Pin Name*	*IC Pin Number*
VREF	VDD	13
TMODE	TMODE	9
TDA	PC1	2
TCK	PC0	1
NRSTI	NRST	10

5.2 Gate Driver

The gate driver is a brushless DC motor gate driver for supply voltages V_{VSUP} from 12V to 72V. The wide operating range makes it suitable for many types of industrial applications. Functionality down to $V_{VSUP,min}$ = 7V(min)in Standby mode (drivers off) allows the gate driver to be used in 24V commercial vehicles as well. An integrated, internally compensated PFM DC-DC buck converter provides V_{VG} = 11V(typ) for efficient, reliable gate drive even at high supply voltages and can supply other loads (Hall Sensors) up to around 100mA. The buck converter is capable of 100% duty cycle operation which allows the gate driver to be used in 12V industrial applications as well. It is possible to bypass the buck converter by applying a voltage of 9.5V to 14V directly to its output VG and leave out the inductor to reduce component cost. In this case LX should be connected to VSUP.
VG serves as gate voltage for the 3 half-bridges and as the input of an internal V_{VDD} = 3.3V(typ) linear regulator. EN activates the power supplies and RUN activates the six pin motor interface and can keep the IC active after EN has gone low to allow a microcontroller to shut down the device in a controlled manner.
The IC incorporates protection features such as: temperature warning, overcurrent, and low enhancement protection of the 6 power FETs. An integrated current sense amp allows for current control and overcurrent detection. The gain of this current sense amplifier can be programmed with external components. Motor phase resistor dividers are also integrated to ease BEMF based commutation algorithms. A two input multiplexer BEMF_MUX[1,2] selects access to the divided phase voltages and scaled down supply voltage.

5.2.1 Power Supply

The gate driver has two integrated power supplies which are activated by EN.

1. An integrated step-down DC-DC converter provides VVG = 11V (typ) up to 100mA for the external gates, the internal linear regulator, and optional external loads such as Hall sensors, a microcontroller through an external regulator, etc.
2. A linear regulator to provide VVDD = 3.3V(typ) power to the internal circuitry and the integrated motor control unit.

Both regulators will be shut down if Sleep Mode is activated by taking RUN and EN low. During startup ERR[1,2] are both at VDD level until VDD is ready and for at least t_{RUN_DEB} = 500µ(typ).

5.2.1.1 Standby, Wake-up and Shutdown

When V_{VSUP} powers up, the gate driver comes up in Sleep mode. During Sleep mode, the IC uses lowest quiescent current, LX is high impedance and the gate drivers GL[1-3] are driving the external gates low. A high level at EN for at least $T_{DEB\_EN,typ}$ = 100µs (typ)powers up the internal DC-DC converter, which supplies the internal 3.3V regulator as well as the gate drivers. ERR[1,2] will be high until the VDD power good level has been exceeded and the IC is ready to operate.

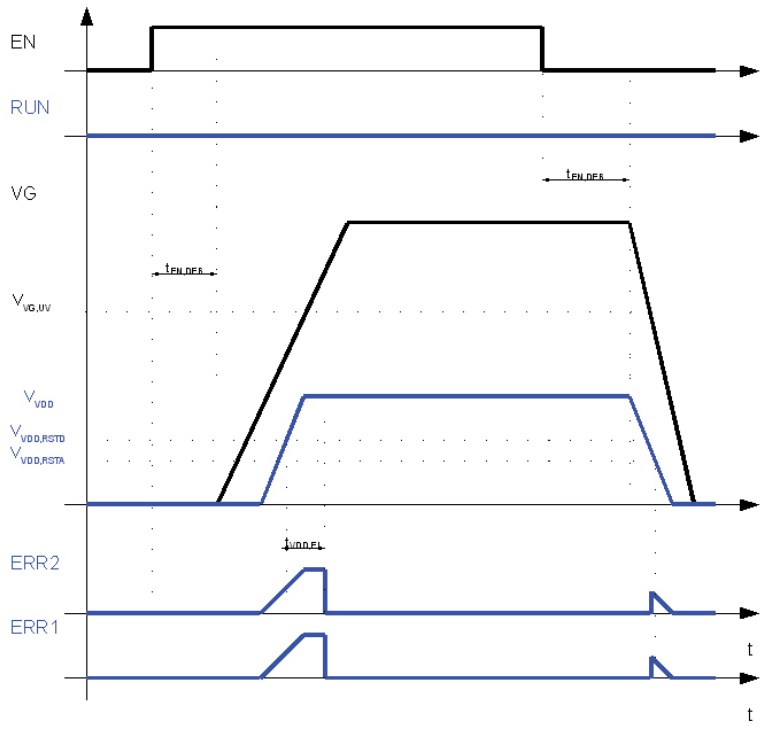

Figure 5.2.1.1-1: Wake-up and Shutdown Behaviour

Sleep mode can be entered by holding RUN and EN low for at least $t_{RUN\_DEB,typ}$ = 500µs.

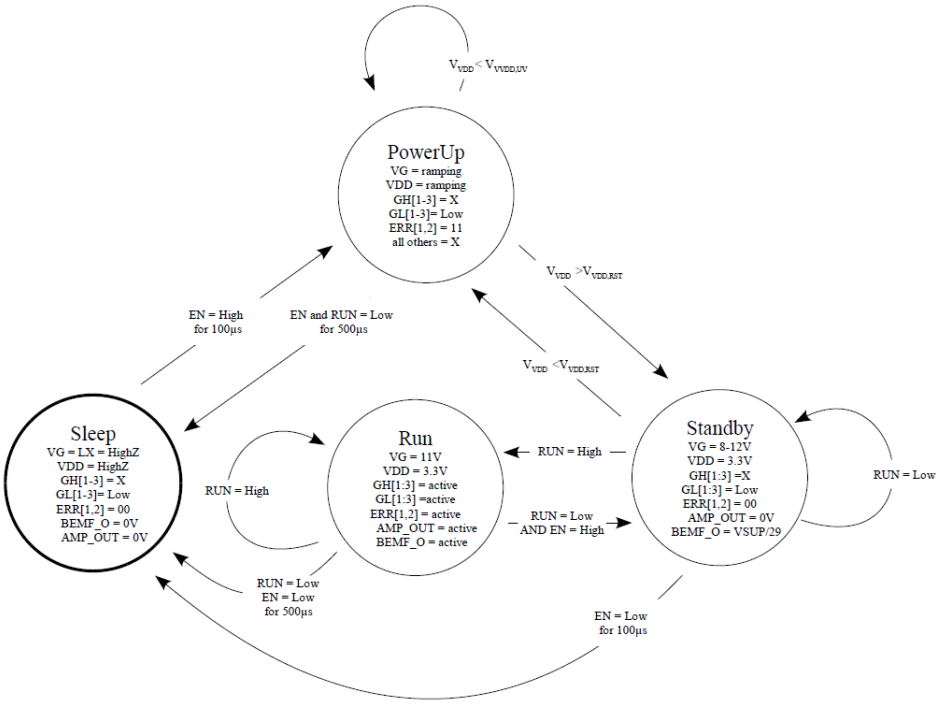

Figure 5.2.1.1-2: System State Diagram

5.2.1.2 DC-DC Buck Converter

The E523.52 includes a DC-DC step down converter with internal compensation to provide V_{VG} = 11V(typ) up to 100mA. The peak current is limited to I_{LX} = 160mA(min). An internal switch connects VSUP to LX during the on time of the converter. An external freewheeling diode,an inductor and filter capacitor complete the circuit. The feedback of the DC-DC converter VG is monitored for undervoltage, the driver section gets disabled if VG drops below 8V(min) and ERR1 goes high. Power to VG and the internal VDD supply is maintained to allow for proper shut-down. The converter uses a PFM (pulse frequency mode) algorithm which allows for low standby currents and does not require external compensation.
The switching frequency f_{SWITCH} varies input voltage and load dependent but stays below 1.15MHz(max).

DC-DC Converter Bypass

For component cost reasons it may be desirable to bypass the DC-DC converter by applying an external voltage between 9.5 and 14V directly to VG and shorting LX to VSUP. The inductor can then be eliminated. The total load current at VG will be the sum of the gate current, the external load at VDD and the IC supply current.

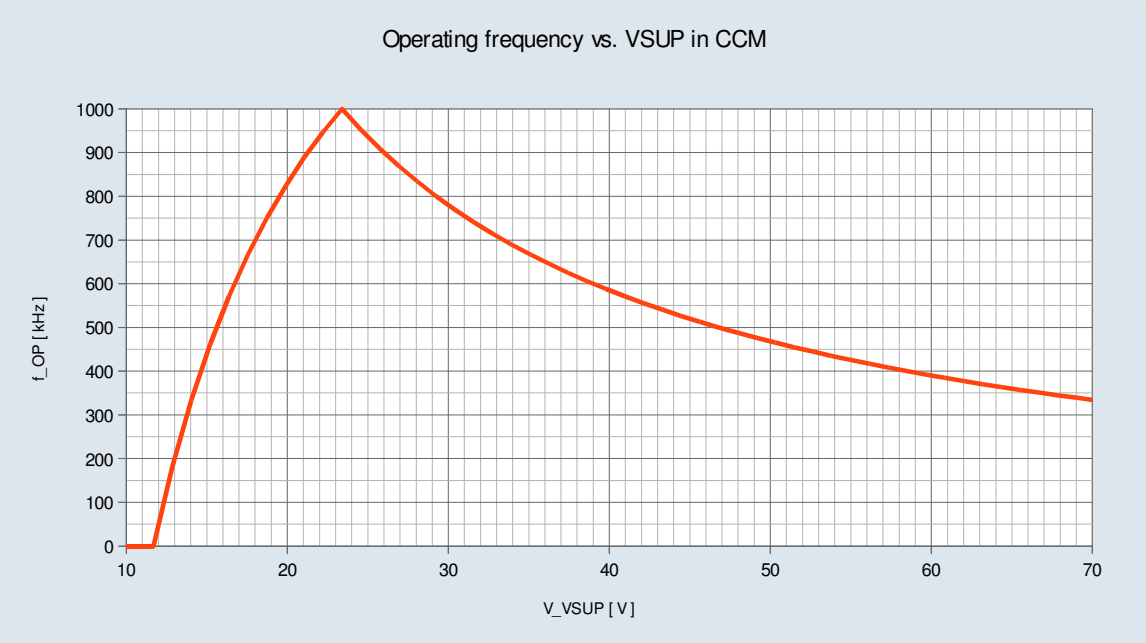

Figure 5.2.1.2-1: DCDC switching frequency vs V_{VSUP}

5.2.1.3 VDD Supply

Power to the internal analog and digital circuitry is provided by a linear regulator supplied from VG. VDD must be buffered with an external ceramic capacitor 1µF. VDD serves as the feedback for the regulator and as the input to the IC. If the voltage on VDD falls below the power-on reset level V_{VDD_RSTD} = 3.1V(max) a high appears on ERR[1,2] to indicate VDD undervoltage.

The VDD supply is a 3.3V linear regulator which supplies the internal low-voltage circuitry and external loads up to 30mA. The regulator input voltage is VG.

5.2.2 Power FET Gate Drivers

The low-side drivers GL[1-3] are powered from VG and return the gate current to GND, the high-side drivers GH[1-3] are powered from BST[1-3] and return the gate current to the motor connection M[1-3]. A bootstrap circuit consisting of a diode and a capacitor must be connected to each high-side driver to level shift its gate voltage. 100% duty cycle is possible within limits.

The drivers have a simple protection against shortcircuit of individual driver FETs and shoot through current through one B6 branch: A simple digital lock prevents both high-side and low-side from being activated simultaneously, and if the drain to source voltage across an active FET exceeds V_{DS_OFF} = 2.5V (typ) the branch is turned off and overcurrent error is asserted via a high on both error flags ERR[1,2].

5.2.2.1 Block Diagram

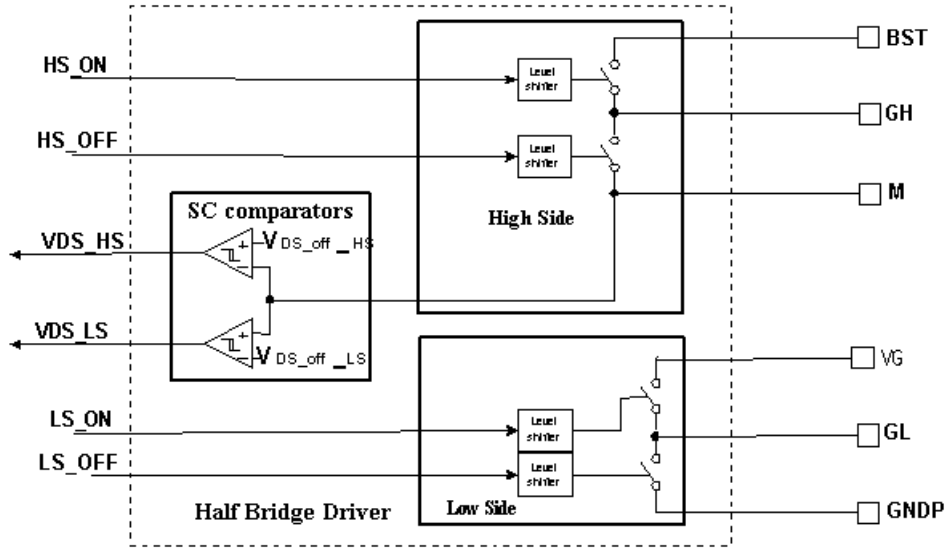

Figure 5.2.2.1-1: Half-Bridge Driver: Block diagram

5.2.2.2 Short-Circuit Protection

The drain-source voltage of all 6 Power-FETs in ON state (both highside drivers and lowside drivers) is monitored and compared with a reference voltage by short-circuit comparators for overcurrent and short-circuit detection. The typical comparator threshold is about 2.5V.

The short circuit detection is enabled about 5us after the L/H transition of the corresponding PWM input signal which switches on the Power-FET. Then the detection of a short-circuit condition switches off the corresponding FET driver after T_{deb,VDS_off} and reports an error via the ERR pins. The error diagnosis remains active until pin RUN is switched to Low. The FET driver will be disabled until the error condition is cleared.

5.2.2.3 Behaviour in STANDBY and SLEEP mode

In STANDBY mode and in SLEEP mode the gates of all high side drivers and low-side drivers are switched to Low, so all Power-FETs are switched-off. The short-circuit comparators are not active.

5.2.3 The Control Inputs

5.2.3.1 Enable Pin

The enable pin EN is used to wake up the IC from SLEEP mode. EN is a high-voltage pin and can be directly connected to VSUP, but the switching threshold is in the low-voltage range.

Note: The wake up sequence can only be finished when V_{VG} exceeds V_{VG_UV}. This means correct BEMF measurement scaling will be available only after V_{VG} exceeded V_{VG_UV} after wake up.

5.2.3.2 Internal Control Ports

The gate driver has a RUN input to activate RUN mode and 6 digital PWM control ports. The ports are internal controlled by the motor control unit. While RUN is low after wake-up, the IC is in STANDBY mode and the gate driver

outputs GH(1-3) and GL(1-3) are driven off. Once RUN has gone to high to activate the RUN mode, RUN must remain low longer than T_{RUN_DEB}= typ. 500us to enter STANDBY mode again. If EN is low in STANDBY mode, the IC enters SLEEP mode.

5.2.4 BEMF Detection

Back-EMF transition detection is done by sampling the un-driven motor phase during a PWM period until the value passes $V_{VSUP/2}$. This transition is used to advance the motor commutation to the next phase for example in block commutation. The motor driver has an integrated BEMF divider circuit to be used as input for an ADC in the micro-controller. The circuit consists of four voltage dividers, an analogue multiplexer and a buffer. The inputs are taken from the motor phases M[1-3] and the supply voltage. The buffered output appears at BEMF_O. To provide low voltage EN pin monitoring, the supply voltage measurement channel (BEMF_MUX[1,2]=00)will read GND if the EN pin has gone low.

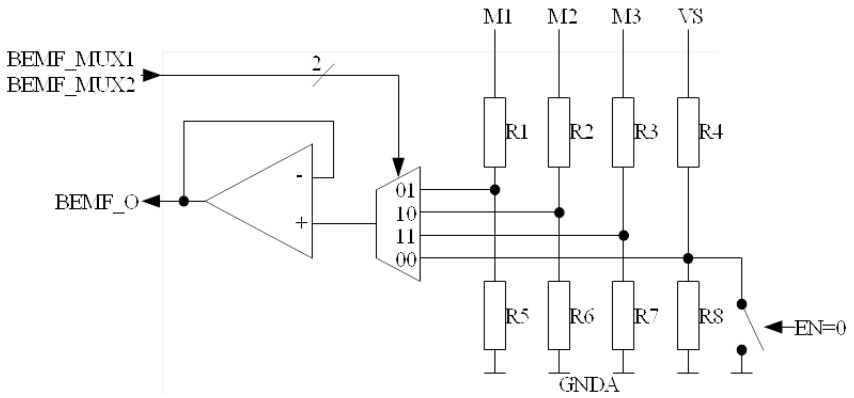

Figure 5.2.4-1: Block Diagram of the BEMF Detection

The output at BEMF_O can be selected via the pins BEMF_MUX1 and BEMF_MUX2. The supply and the motor phase voltage are always scaled by a factor of 1/28.8(typ). The gain settings are defined when the IC enters Run mode.

Table 5.2.4-1: BEMF_MUX1 and BEMF_MUX2 function

BEMF_MUX2	***BEMF_MUX1***	***Function***
0	0	Supply voltage selected,GND if EN is low.
0	1	Motor phase M1 selected
1	0	Motor phase M2 selected
1	1	Motor phase M3 selected

5.2.5 Current Sense Amplifier / Overcurrent Detection

An integrated high speed OPAMP helps measure the system current across a low side shunt resistor and triggers the overcurrent detection. The amplifier is only active during RUN mode (RUN = High). Amplifier gain and offset are dimensioned by connecting an external resistor network between IN, IP and AMP_OUT. The shunt resistor and gain have to be adjusted to the full scale input range of the ADC in the microprocessor.
If the output voltage at AMP_OUT exceeds 90% of the VDD supply: $V_{AMP\_OUT,OC}$ = 3.0V(typ) an overcurrent event is reported. For dimensioning information of the amplifier circuit see section.

5.2.6 Monitoring and Safety Functions

The motor driver can detect and report undervoltage, overtemperature, and overcurrent. These errors are reported as active high on the pins ERR1 and ERR2 in the following way.

Table 5.2.6-1: Error Diagnosis

Failure	*RUN*	*ERR2*	*ERR1*	*Comment*
No error	0/1	0	0	
VG Undervoltage	1	0	1	not latched
Overtemperature	1	1	0	not latched
Overcurrent/SC	1	1	1	latched
VDD Undervoltage	0/1	1	1	not latched

5.2.6.1 VG Monitoring / Under-Voltage Detection

To protect the external FETs from operating in their linear region and experiencing thermal stress, VG is observed. If VG falls below V_{VG_UV}=9.5V(max) undervoltage condition is reported on the ERR1 output. Undervoltage will shut down the driver stage. If VG recovers the ERR1 will go high and the IC will release the output stage.

5.2.6.2 Temperature Monitoring and Over-Temperature Detection

If the IC temperature exceeds OT_{th_warn}=+150°C(min), an overtemperature warning is reported, the system will keep running, but should be shut down as fast as possible. The overtemperature condition is not latched and clears on its own once the temperature returns below the threshold. If the temperature exceeds $OT_{th\_switch\_off}$=+160°C (min) the IC will shut down its power supplies and enter Sleep mode.

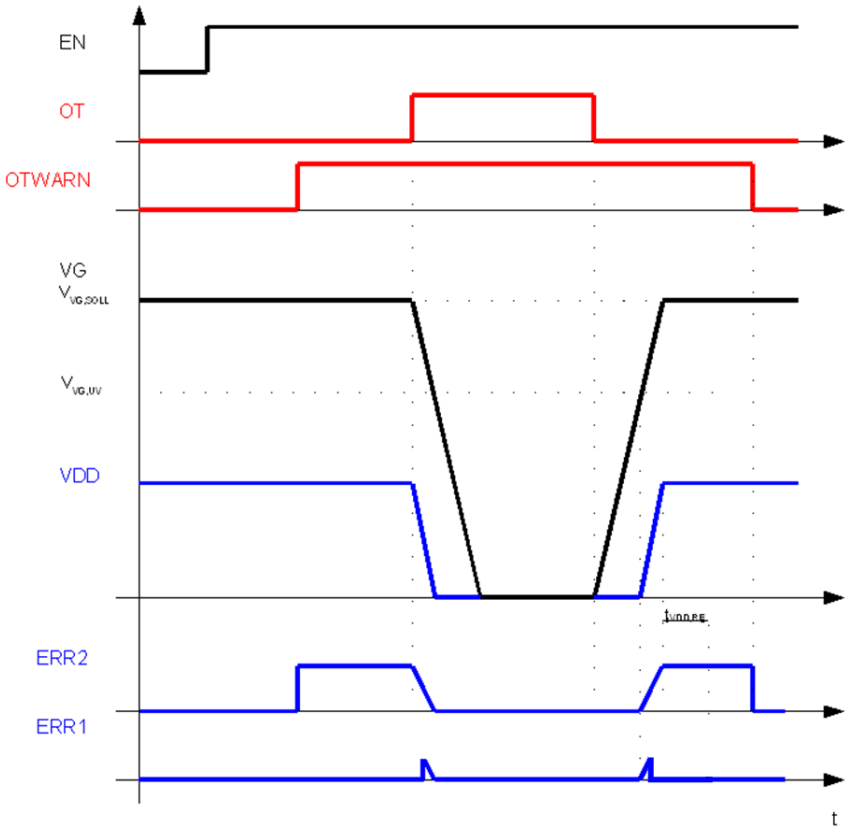

Figure 5.2.6.2-1: Over-Temperature Indication and Shut Down

5.2.6.3 Overcurrent/Fet short-circuit

An overcurrent comparator observes the output of the current sense amp. If the voltage on VAMP_OUT exceeds $V_{AMP\_OUT,OC}$ = 3V(typ) for longer then t_{OC}= 10µs(typ) the drivers are turned off and an overcurrent event is indicated. The drivers will stay off until the error is cleared. To clear this error, cycle the RUN input between low and high. Driver shortcircuit is detected by monitoring the voltage drop across each individual FET. If, during operation, the voltage across a FET exceeds V_{DS_OFF} =2.5V(typ), the corresponding half-bridge is disabled, until RUN is cycled to clear the error. The drain-source voltage of all 6 Power-FETs in ON state (both highside drivers and lowside drivers) is monitored and compared with a reference voltage by short-circuit comparators for overcurrent and short-circuit detection.
The typical comparator threshold is about 2.5V. The short circuit detection is enabled about 5µs after the L/H transition of the corresponding PWM input signal which switches on the Power-FET. Then the detection of a short-circuit condition switches off the corresponding FET driver after T_{deb,VDS_off} and reports an error via the ERR pins. The error diagnosis remains active until pin RUN is switched to Low. The FET driver will be disabled until the error condition is cleared.

5.2.6.4 VDD Undervoltage

The VDD undervoltage signal can be used to check for VDD integrity before activating the system. As long as RUN=Low a high on both ERR bits indicates VDD undervoltage. During RUN mode, ERR[1,2] = High could indicate an undervoltage event or an overcurrent event. Immediate shutdown (RUN=Low) should be the response in either

case, if the ERR signals remain high a VDD undervoltage was the cause, if the ERR signals go low, it was an over-current event.

5.3 Motor Control Unit

5.3.1 Analog Part

5.3.1.1 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.
The external Buffer Capacitance CDDC is connected to the output.

5.3.1.2 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.

5.3.1.2.1 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

5.3.1.2.2 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

5.3.1.2.3 System Clock RC Oscillator

This oscillator clocks the digital system.

5.3.1.2.4 Watchdog Clock RC Oscillator

The oscillator is used to clock a part of the digital watchdog module.

5.3.1.2.5 NRST debouncer

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

5.3.1.3 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.

5.3.1.4 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.

5.3.1.5 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 will 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.

5.3.2 Digital Part

5.3.2.1 Base Addresses

Table 5.3.2.1-1: Address Table

base address	*size*	*module name*	*instance name*	*description*
0x8000	0x8000	FLASH	FLASH	FLASH Memory
0x4000	0x4000	SYS_ROM	SYS_ROM	System ROM Memory
0x1000	0x1000	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
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	SCI	SCI	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

5.3.2.2 Memory IP's

5.3.2.2.1 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

5.3.2.2.2 System ROM (SYS_ROM)

- size: 16Kbyte
- read only
- contains standard LIN routines which can be used by the the executed user program and a boot loader program (see "boot code flow chart") or alternative "re-flash via LIN" support routines, which can also be started from the executed user program.

The System ROM content may vary depending on customer requirements and may contain for example:

- a standard ELMOS boot loader
- a customer specific boot loader delivered by ELMOS
- a customer boot loader

5.3.2.2.3 SRAM

- size: 4Kbyte
- byte write support
- per byte parity protection

5.3.2.2.4 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 value to allow full system access.
  - This signature value depends on the 8 bit customer ID register value of SYS_STATE module which is set and locked during device INFO BOOT before JTAG can halt device for debug purpose.
  - a customer ID and signature database is maintained by ELMOS
  - 253 signatures (customers) are possible

5.3.2.3 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 behaviour 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:

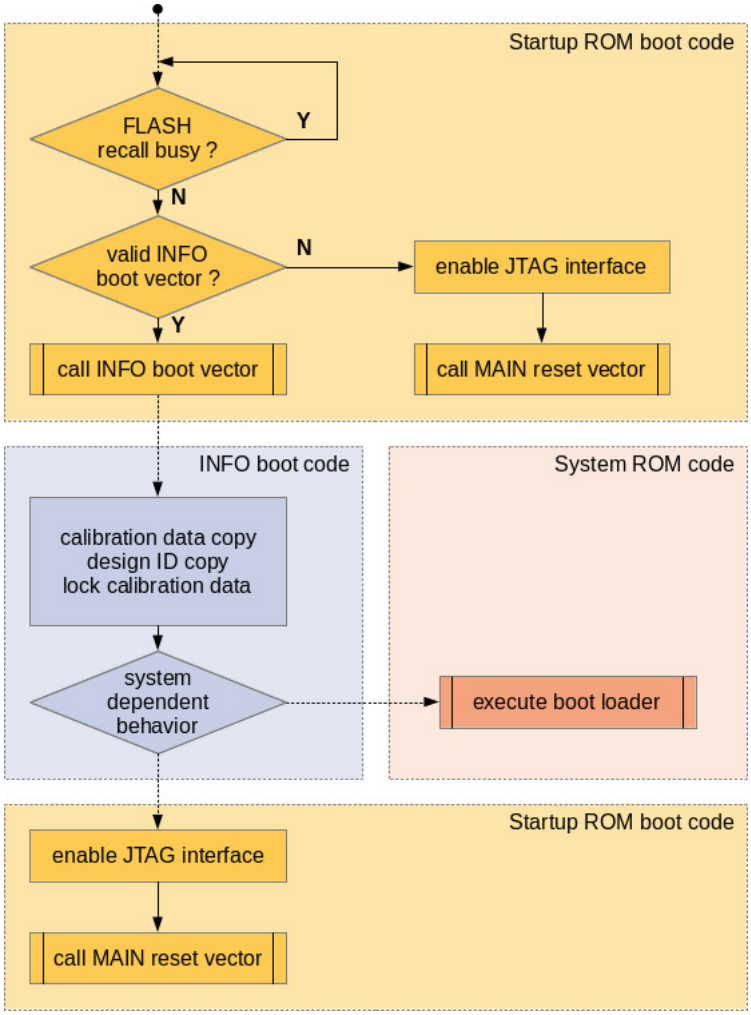

Figure 5.3.2.3-1: boot code flow chart

5.3.2.4 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

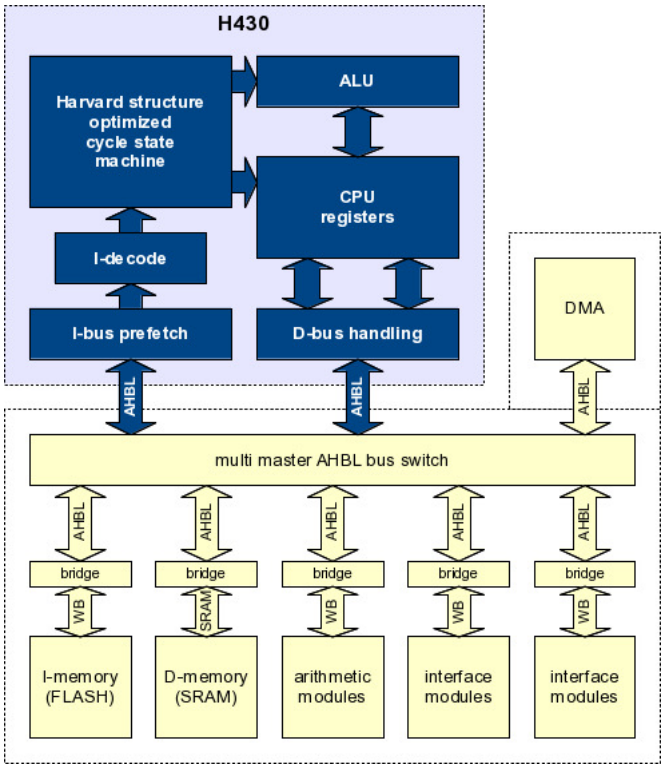

Figure 5.3.2.4-1: H430 Environment Example

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 5.3.2.4-1: 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.

5.3.2.4.1 CPU Registers

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

5.3.2.4.1.1 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.

5.3.2.4.1.2 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.

5.3.2.4.1.3 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 5.3.2.4.1.3-1: Register Table

Register Name	***Address***	***Description***
Status Register	SR/R2	

Table 5.3.2.4.1.3-2: 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.

5.3.2.4.1.4 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 5.3.2.4.1.4-1: 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.

5.3.2.4.1.5 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.

5.3.2.4.2 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 5.3.2.4.2-1: 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 5.3.2.4.2-2: 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 5.3.2.4.2-3: 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 5.3.2.4.2-4: 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 5.3.2.4.2-5: 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 5.3.2.4.2-6: 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 5.3.2.4.2-7: 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.

5.3.2.4.3 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 5.3.2.4.3-1: 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 5.3.2.4.3-1: 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 5.3.2.4.3-2: 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 5.3.2.4.3-3: 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.

Opcode	Mnemonic	S-Reg, D-Reg	Operation	V	N	Z	C	Remark
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.
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.

Opcode	Mnemonic	S-Reg, D-Reg	Operation	V	N	Z	C	Remark
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 5.3.2.4.3-4: 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
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 5.3.2.4.3-5: 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

5.3.2.4.4 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 5.3.2.4.4-1: Cycle Count Table

5.3.2.4.5 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

5.3.2.5 Sub Parts

5.3.2.5.1 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:

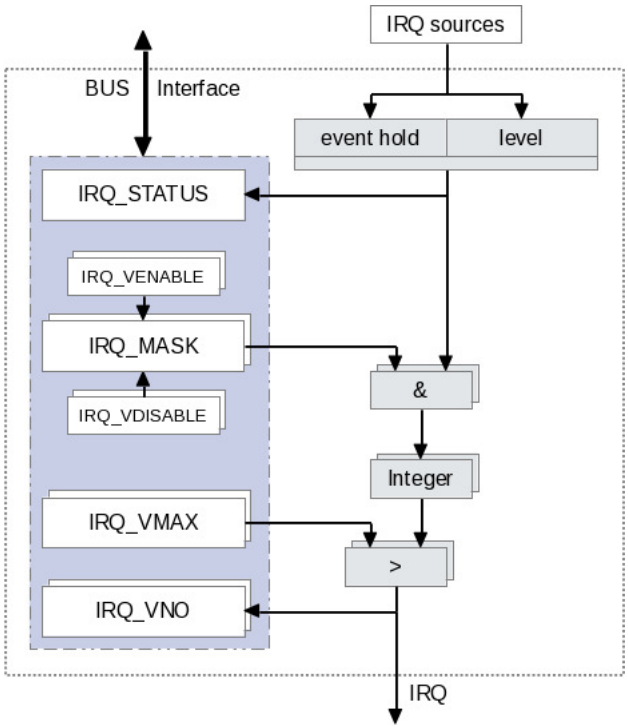

Figure 5.3.2.5.1-1: VIC logic structure

Two Stage Interrupt System Structure:

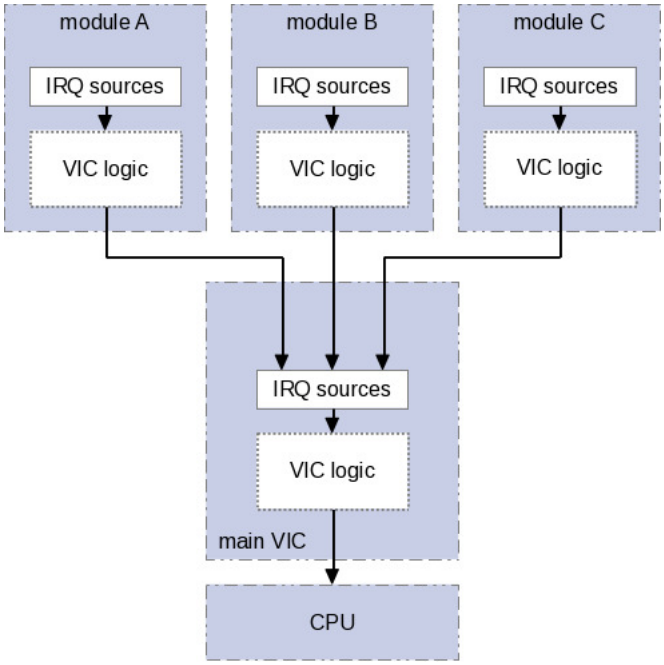

Figure 5.3.2.5.1-2: Two stage interrupt system structure

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 5.3.2.5.1-1: 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 5.3.2.5.1-2: 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 5.3.2.5.1-3: 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 5.3.2.5.1-4: 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 5.3.2.5.1-5: 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 5.3.2.5.1-6: 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 5.3.2.5.1-7: 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 5.3.2.5.1-8: 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 5.3.2.5.1-9: 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 5.3.2.5.1-10: 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 5.3.2.5.1-11: 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 5.3.2.5.1-12: 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															

5.3.2.5.2 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

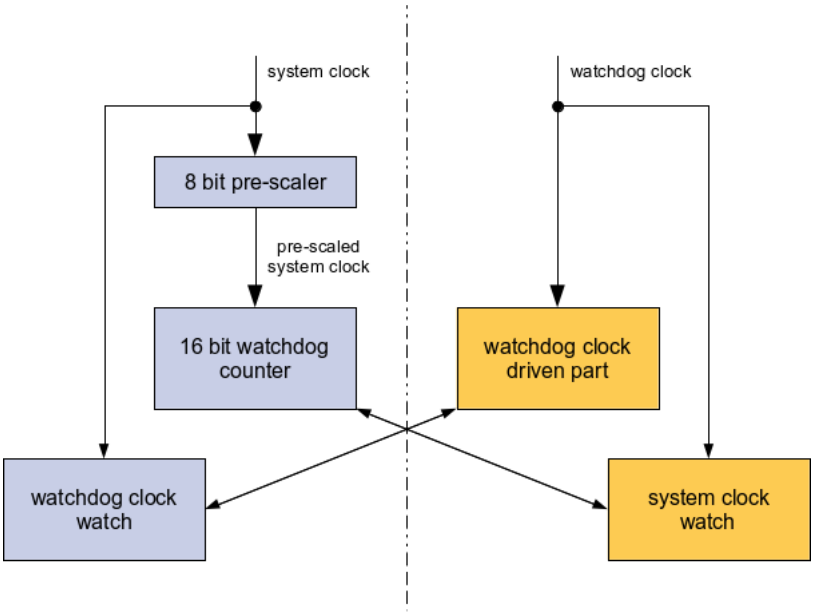

Figure 5.3.2.5.2-1: Structure

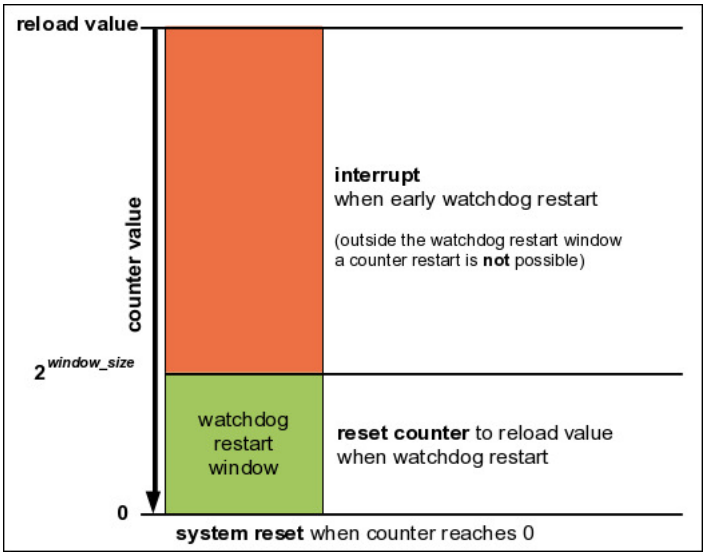

Figure 5.3.2.5.2-2: Timing

Table 5.3.2.5.2-1: 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 5.3.2.5.2-2: 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 5.3.2.5.2-3: 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 5.3.2.5.2-4: 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 5.3.2.5.2-5: 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 5.3.2.5.2-6: 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 5.3.2.5.2-7: 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 5.3.2.5.2-8: 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 5.3.2.5.2-9: 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 5.3.2.5.2-10: 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 5.3.2.5.2-11: 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 5.3.2.5.2-12: 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															

5.3.2.5.3 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.

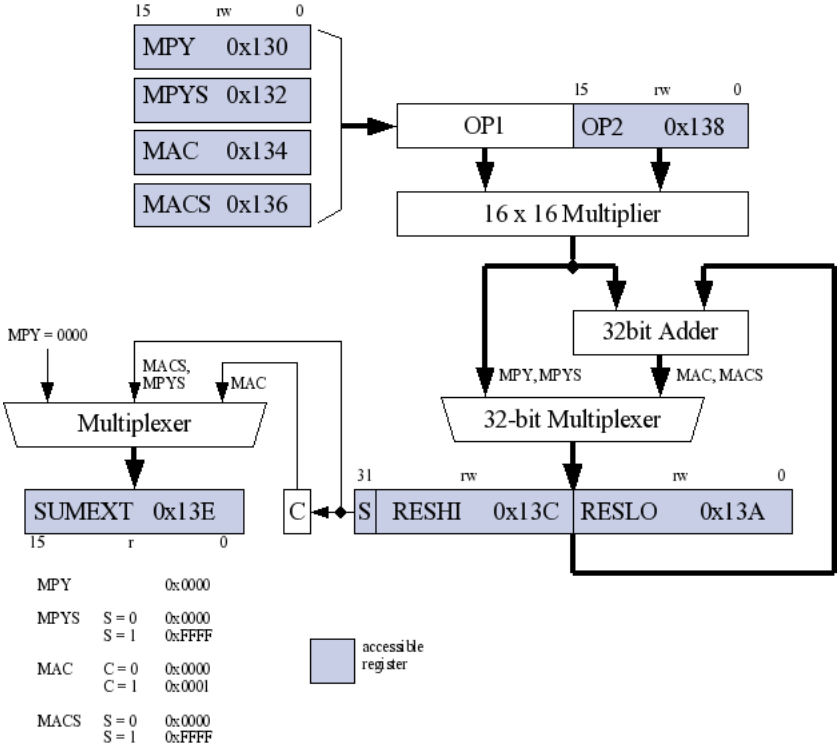

Figure 5.3.2.5.3-1: Multiplier Structure

Table 5.3.2.5.3-1: 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

Register Name	Address	Description
MPYS	0x32	multiply signed register
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 5.3.2.5.3-2: 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 5.3.2.5.3-3: 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 5.3.2.5.3-4: 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 5.3.2.5.3-5: 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 5.3.2.5.3-6: 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 5.3.2.5.3-7: 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 5.3.2.5.3-8: 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 5.3.2.5.3-9: 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 5.3.2.5.3-10: 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 5.3.2.5.3-11: 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 5.3.2.5.3-12: 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 5.3.2.5.3-13: 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 5.3.2.5.3-14: 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 5.3.2.5.3-15: 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 5.3.2.5.3-16: 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 5.3.2.5.3-17: 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 5.3.2.5.3-18: 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															

5.3.2.5.4 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 5.3.2.5.4-1: 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

Register Name	*Address*	*Description*
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 5.3.2.5.4-2: 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 5.3.2.5.4-3: 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 5.3.2.5.4-4: 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 5.3.2.5.4-5: 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 5.3.2.5.4-6: 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 5.3.2.5.4-7: 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 5.3.2.5.4-8: 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 5.3.2.5.4-9: 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 5.3.2.5.4-10: 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 5.3.2.5.4-11: 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 5.3.2.5.4-12: 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 5.3.2.5.4-13: 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 5.3.2.5.4-14: 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															

5.3.2.5.5 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 5.3.2.5.5-1: 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

Register Name	*Address*	*Description*
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
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 5.3.2.5.5-2: 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	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** : 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 5.3.2.5.5-3: 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 5.3.2.5.5-4: 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 5.3.2.5.5-5: 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 5.3.2.5.5-6: 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 5.3.2.5.5-7: 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															

Table 5.3.2.5.5-8: 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 5.3.2.5.5-9: 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 5.3.2.5.5-10: 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 5.3.2.5.5-11: 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 5.3.2.5.5-12: 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 5.3.2.5.5-13: 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 5.3.2.5.5-14: 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 5.3.2.5.5-15: 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 5.3.2.5.5-16: 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 5.3.2.5.5-17: 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 5.3.2.5.5-18: 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															

5.3.2.5.6 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
- reset enable
- analog part calibration registers
- IC version
- Customer ID
- target die clock generator
  - f_target = f_system_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 5.3.2.5.6-1: 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
SW_RESET	0x0A	sw reset register
CALIBRATION_LOCK	0x0C	calibration lock register
CALIBRATION	0x0E	calibration data register
IC_VERSION_0	0x10	IC version register 0
IC_VERSION_1	0x12	IC version register 1
IC_VERSION_2	0x14	IC version register 2
IC_VERSION_3	0x16	IC version register 3
CUSTOMER_ID	0x18	CUSTUMER ID register
DEVICE_ID	0x1A	Device ID register
SIGNATURE_0	0x20	Signature Register
SIGNATURE_1	0x22	Signature Register

Register Name	*Address*	*Description*
SIGNATURE_2	0x24	Signature Register
SIGNATURE_3	0x26	Signature Register
SYS_CLK_CON-FIG	0x28	System Clock Config 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 5.3.2.5.6-2: Register **MODULE_ENABLE** (0x00) module enable register

	MSB															***LSB***
Content	-	-	13	12	11	10	9	8	7	6	5	4	3	2	1	-
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
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 **1** : sci - SCI module enable															

Table 5.3.2.5.6-3: Register **CONTROL** (0x02) system control register

	MSB															***LSB***
Content	15:10						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	**15:10** : sys_clk_adjust - system clock RC oscillator trim value offset This value is always subtracted from the internal trim value and only allows to lower the system clock frequency. one adjust LSB changes the system clock by approx. -0.25% of its non-adjusted value **9:7** : target_clk_h - target die clock high period cycles selection (base clock: 24MHz) 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: 24MHz) 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 : 4 MHz 1 : 8 MHz 2 : 12 MHz 3 : 24 MHz 4 : 48 MHz															

Table 5.3.2.5.6-4: Register **RESET_STATUS** (0x04) reset status register

	MSB															***LSB***
Content	-	-	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	**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 5.3.2.5.6-5: Register **RESET_STATUS_CLEAR** (0x06) reset status 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 - writing clears RESET_STATUS register															

Table 5.3.2.5.6-6: Register **RESET_ENABLE** (0x08) reset enable 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	R	R
Bit Description	**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 5.3.2.5.6-7: Register **SW_RESET** (0x0A) sw reset register

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	-	-	-	-	-	-	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	W	R
Bit Description	**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 5.3.2.5.6-8: Register **CALIBRATION_LOCK** (0x0C) calibration lock 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** : lock - lock calibration values (cal values are changed to read only when lock_cal was set) Note: once set, calibration data cannot be unlocked again															

Table 5.3.2.5.6-9: Register **CALIBRATION** (0x0E) calibration data register

	MSB															***LSB***
Content	15:8								7:4				-	2:0		
Reset value	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
Access	R	R/W	R/W	R/W	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** : cal_sys_osc - system oscillator frequency trim value **7:4** : cal_bgap- bandgap selection (trim) value **2:0** : cal_vref - reference voltage trim value															

Table 5.3.2.5.6-10: Register **IC_VERSION_0** (0x10) IC version register 0

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	1	1	0	1	0	1	0	1	0	0	1	1	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** : high_byte - high_byte IC design version 0 **7:0** : low_byte - manufacturing type (M, P, E, ...)															

Table 5.3.2.5.6-11: Register **IC_VERSION_1** (0x12) IC version register 1

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	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** : high_byte - high_byte IC design version 1 **7:0** : low_byte - low_byte IC design version 1															

Table 5.3.2.5.6-12: Register **IC_VERSION_2** (0x14) IC version register 2

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	1	1	1	0	0	1	0	0	1	1	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** : high_byte - high_byte IC design version 2 **7:0** : low_byte - low_byte IC design version 2															

Table 5.3.2.5.6-13: Register **IC_VERSION_3** (0x16) IC version register 3

	MSB															***LSB***
Content	15:8								7:0							
Reset value	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	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** : high_byte - should be written 0 **7:0** : low_byte - IC design version (A, B, ...)															

Table 5.3.2.5.6-14: Register **CUSTOMER_ID** (0x18) CUSTUMER ID 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** : id - customer identification number used by hardware to allow JTAG access to CPU and FLASH usable values: 2 .. 255 Note: values 0 und 1 must not be used ! Note: value 255 (default) is the ELMOS ID value and has not to be used for other customer !															

Table 5.3.2.5.6-15: Register **DEVICE_ID** (0x1A) Device ID 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** : 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: will be administrated by ELMOS device database															

Table 5.3.2.5.6-16: Register **SIGNATURE_0** (0x20) Signature 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 - 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 5.3.2.5.6-17: Register **SIGNATURE_1** (0x22) Signature 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 - 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 5.3.2.5.6-18: Register **SIGNATURE_2** (0x24) Signature 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 - 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 5.3.2.5.6-19: Register **SIGNATURE_3** (0x26) Signature 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 - 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 5.3.2.5.6-20: Register **SYS_CLK_CONFIG** (0x28) System Clock Config 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 5.3.2.5.6-21: 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** : 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 5.3.2.5.6-22: 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 5.3.2.5.6-23: Register **IRQ_VENABLE** (0x38) IRQ vector enable 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	W	W	W
Bit Description	**2:0** : vno - vector number of interrupt to enable															

Table 5.3.2.5.6-24: Register **IRQ_VDISABLE** (0x3A) IRQ vector disable 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	W	W	W
Bit Description	**2:0** : vno - vector number of interrupt to disable															

Table 5.3.2.5.6-25: 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 5.3.2.5.6-26: 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															

5.3.2.5.7 IO MUX Control Module (IOMUX_CTRL)

5.3.2.5.7.1 Description

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

5.3.2.5.7.2 Operating environment

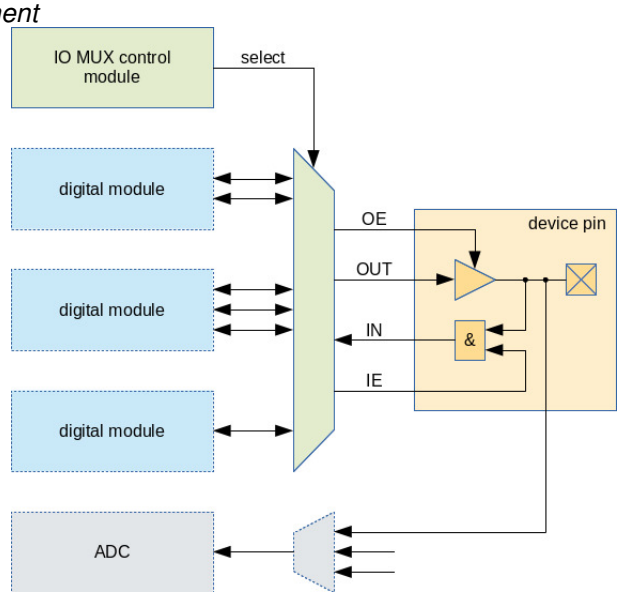

Figure 5.3.2.5.7.2-1: operating environment

5.3.2.5.7.3 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

5.3.2.5.7.4 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 5.3.2.5.7.4-1: 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
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 5.3.2.5.7.4-2: 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	SYS_STATE	TARGET_CLK	out
PA 5	PWMN	LS_U	out
PA 6	PWMN	LS_V	out
PA 7	PWMN	LS_W	out

Table 5.3.2.5.7.4-3: 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 5.3.2.5.7.4-4: 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	SYS_STATE	TARGET_CLK	out
PA 4	PWMN	HS_X	out
PA 5	CCTIMER_0	PWM	out
PA 6	CCTIMER_1	PWM	out
PA 7	SARADC_CTRL	MUX_1	out

5.3.2.5.7.5 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 5.3.2.5.7.5-1: 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 5.3.2.5.7.5-2: 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 5.3.2.5.7.5-3: 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

pin name	module name	signal name	signal direction
PB 3	PWMN	HS_W	out
PB 4	GPIO_B	IO4	in / out
PB 5	SARADC_CTRL	DBG_SAMPLING	out
PB 6	PWMN	LS_X	out
PB 7	PWMN	HS_X	out

Table 5.3.2.5.7.5-4: 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

5.3.2.5.7.6 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 5.3.2.5.7.6-1: 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

5.3.2.5.7.7 Register Interface

Table 5.3.2.5.7.7-1: Registers

Register Name	***Address***	***Description***
PA_IO_SEL	0x00	IO signal select register
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 5.3.2.5.7.7-2: 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 5.3.2.5.7.7-3: 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 5.3.2.5.7.7-4: 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 5.3.2.5.7.7-5: 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 5.3.2.5.7.7-6: 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 5.3.2.5.7.7-7: 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 5.3.2.5.7.7-8: 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.															

5.3.2.5.7.8 IO MUX setup example

```
/* 523.99 - 523.01 die-to-die configuration */

IOMUX_CTRL_PA_IO_SEL_bit.sel_0 = 1;  /* SPI_0 SCK */
IOMUX_CTRL_PA_IO_SEL_bit.sel_1 = 1;  /* SPI_0 SDI */
IOMUX_CTRL_PA_IO_SEL_bit.sel_2 = 1;  /* SPI_0 SDO */
IOMUX_CTRL_PA_IO_SEL_bit.sel_3 = 1;  /* SPI_0 NSS */
IOMUX_CTRL_PA_IO_SEL_bit.sel_4 = 1;  /* SYS_STATE TARGET_CLK */
IOMUX_CTRL_PA_IO_SEL_bit.sel_5 = 1;  /* PWMN LS_U */
IOMUX_CTRL_PA_IO_SEL_bit.sel_6 = 1;  /* PWMN LS_V */
IOMUX_CTRL_PA_IO_SEL_bit.sel_7 = 1;  /* PWMN LS_W */

IOMUX_CTRL_PB_IO_SEL_bit.sel_0 = 1;  /* SCI RXD */
IOMUX_CTRL_PB_IO_SEL_bit.sel_1 = 1;  /* SCI TXD */
IOMUX_CTRL_PB_IO_SEL_bit.sel_2 = 1;  /* PWMN HS_U */
IOMUX_CTRL_PB_IO_SEL_bit.sel_3 = 1;  /* PWMN HS_V */
IOMUX_CTRL_PB_IO_SEL_bit.sel_4 = 1;  /* PWMN HS_W */

/* 523.99 external pins configuration example */

IOMUX_CTRL_PB_IO_SEL_bit.sel_5 = 0;        /* GPIO_B IO5 */
IOMUX_CTRL_PB_IO_SEL_bit.sel_6 = 0;        /* GPIO_B IO6 */
IOMUX_CTRL_PB_IO_SEL_bit.sel_7 = 0;        /* GPIO_B IO7 */

IOMUX_CTRL_PC0123_IO_SEL_bit.sel_0 = 13;  /* PRE_PWM SYNC_U */
IOMUX_CTRL_PC0123_IO_SEL_bit.sel_1 = 0;   /* GPIO_C IO1 */
IOMUX_CTRL_PC0123_IO_SEL_bit.sel_2 = 0;   /* GPIO_C IO2 */
IOMUX_CTRL_PC0123_IO_SEL_bit.sel_3 = 0;   /* GPIO_C IO3 */

IOMUX_CTRL_PC4567_IO_SEL_bit.sel_4 = 0;   /* GPIO_C IO4 */
IOMUX_CTRL_PC4567_IO_SEL_bit.sel_5 = 0;   /* GPIO_C IO5 */
IOMUX_CTRL_PC4567_IO_SEL_bit.sel_6 = 0;   /* GPIO_C IO6 */
IOMUX_CTRL_PC4567_IO_SEL_bit.sel_7 = 0;   /* GPIO_C IO7 */
```

Figure 5.3.2.5.7.8-1: 523.05 IO MUX setup example

5.3.2.5.8 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 5.3.2.5.8-1: 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
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

Register Name	*Address*	*Description*
AREA_MAIN_H_LOCK	0x22	upper protection lock register

Table 5.3.2.5.8-2: 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 5.3.2.5.8-3: 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 5.3.2.5.8-4: 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 5.3.2.5.8-5: 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 5.3.2.5.8-6: 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 5.3.2.5.8-7: 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 5.3.2.5.8-8: 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 5.3.2.5.8-9: 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 5.3.2.5.8-10: 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 5.3.2.5.8-11: 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!															

5.3.2.5.9 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 5.3.2.5.9-1: 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_CON-FIG	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 5.3.2.5.9-2: 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 5.3.2.5.9-3: 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 5.3.2.5.9-4: 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 - lSDI 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 5.3.2.5.9-5: 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 5.3.2.5.9-6: 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 5.3.2.5.9-7: 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 5.3.2.5.9-8: 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 5.3.2.5.9-9: 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 5.3.2.5.9-10: 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 5.3.2.5.9-11: 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 5.3.2.5.9-12: 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 5.3.2.5.9-13: 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 5.3.2.5.9-14: 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 5.3.2.5.9-15: 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															

5.3.2.5.10 LIN-SCI Module (SCI)

Features

- Full duplex operation
- 8N1 data format, standard mark/space NRZ format
- Extended baud rate selection options
- Interrupt-driven operation with four flags: receiver full, transmitter empty, measurement finished, break character received

Special LIN Support:

- 13 Bit break generation
- 11 Bit break detection threshold
- A fractional-divide baud rate pre-scaler that allows fine adjustment of the baud rate
- Measurement counter which has 16 bits and can be used as a mini-timer to measure break and bit times (baud rate recovery).

- 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
- DMA support
- Timer-Compare Module for
  - LIN Bus Idle measurement
  - LIN Break measurement (used for auto addressing)
  - LIN Frame Length measurement

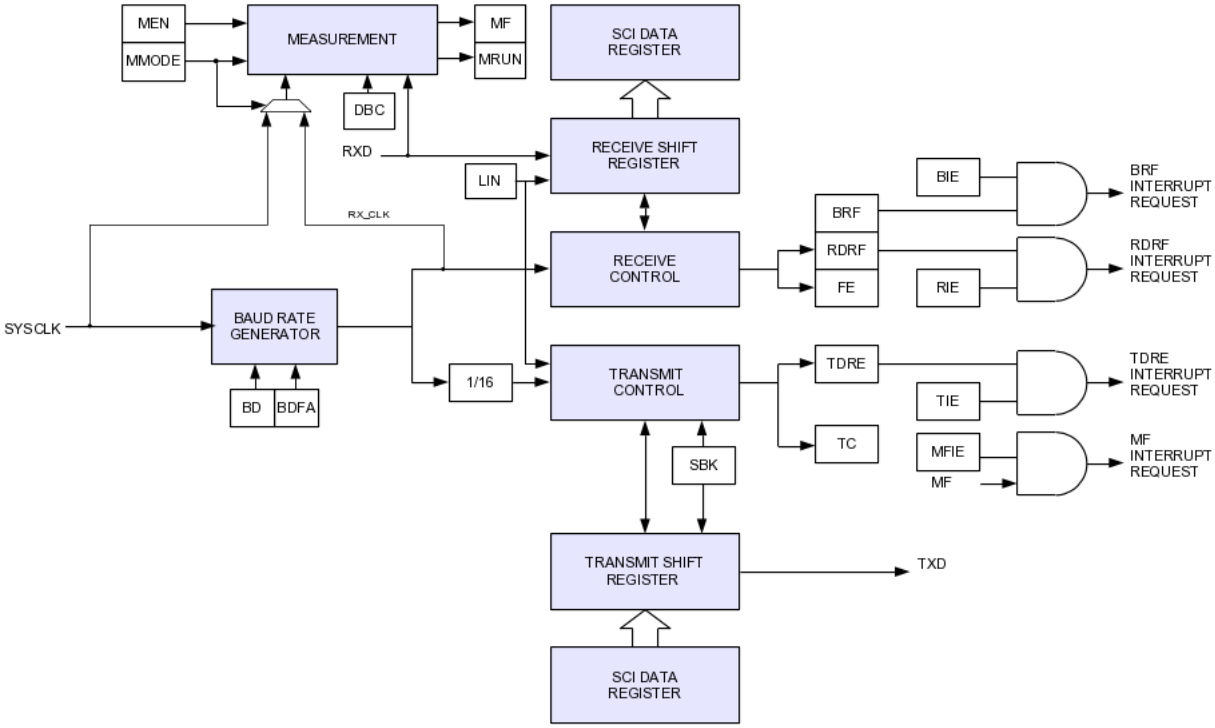

Figure 5.3.2.5.10-1: SCI block diagram

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 LIN mode is set) in combination with AUTO_MEAS a valid break signal starts measurement of a SYNC byte. After the SYNC byte measurement the MF (measurement finish) flag is set and must be processed by the software. The concurrent break measurement will only work when the MF bit is cleared.

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 x expected external baud_rate

Example: When setting internal baud_rate to 115200 baud concurrent baud measurement works with external baud_rates down to 10472 baud. The external baud_rate = 9600 baud can not be synchronized.

Concurrent break measurement can be used to support LIN requirement of interrupting ongoing frames by a new break/sync header.

Baud rate
The divider can be used to achieve divisor values between 1 and 2047,96875. The baud divisor fine adjust 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 = sys clock frequency / (16 * (BD + BDFA))

Note: The 16 bit baud divisor value represents the number of system clock cycles of two bit lengths. The result of a baud measurement(see measurement counter below) can directly be written to the baud rate register.

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 TX: transmitter disabled, bus_collision
for RX: receiver disabled, frame error, overflow error (Note: receiver will be disabled during sync byte measurement, this also leads to a RX DMA abort)
SCI flags are not suppressed during DMA operation. The RDRF flag and the TDRE flag will be handled and reset by the DMA controller. Reading/writing of the DATA_IO register is prohibited during DMA operation.

Table 5.3.2.5.10-1: LIN Parameters

Symbol	*Min*	*Typ*	*Max*	*Unit*	*Description*
t_{BFS}		1/16	2/16	T_{BIT}	value of accuracy of the byte field detection
t_{EBS}	7/16			T_{BIT}	earliest bit sample time $t_{EBS} <= t_{LBS}$
t_{LBS}			10/16 - t_{BFS}	T_{BIT}	latest bit sample time $t_{EBS} <= t_{LBS}$

Table 5.3.2.5.10-2: Registers

Register Name	*Address*	*Description*
BAUD_RATE	0x00	baud config register
CONTROL	0x02	control register
STATUS	0x04	status register
DATA_IO	0x06	data register
MEAS_CONTROL	0x08	measurement control register
MEAS_COUNTER	0x0A	measurement counter register
LIN_CONFIG	0x0C	LIN SCI Configuration
LIN_MODE	0x0E	LIN Mode Register
ADDON_IRQ_EN	0x10	Add-on Interrupt Enable
ADDON_IRQ_STAT	0x12	Add-on Interrupt Status
TIMER_COUNTER	0x14	Timer Counter

Register Name	*Address*	*Description*
TIMER_COMPARE	0x16	Timer Compare
DMA_TX_ADDRESS	0x18	Transmit DMA Address
DMA_TX_LENGTH	0x1A	Transmit DMA Length
DMA_RX_ADDRESS	0x1C	Receive DMA Address
DMA_RX_LENGTH	0x1E	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

Table 5.3.2.5.10-3: 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** : BD - SCI baud divisor select Divisor: 0x000 --> 1 (bypass divider) 0x001 --> 2 0x002 --> 3 ... 0x007 --> 8 ... **4:0** : BDFA - SCI baud divisor fine adjust 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. BDFA[00000] = 0/32 = 0 BDFA[00001] = 1/32 = 0.03125 BDFA[00010] = 2/32 = 0.0625 ... BDFA[10000] = 16/32 = 0.5 ... BDFA[11111] = 31/32 = 0.96875 The divider can be used to achieve divisor values between 1 and 2047.96875. The baud divisor fine adjust 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_{sys}/(16*(BD+BDFA)) Note: The 16 bit baud divisor value represents the number of system clock cycles of two bit lengths. The result of a baud measurement(see measurement counter below) can directly be written to the baud rate register.															

Table 5.3.2.5.10-4: Register **CONTROL** (0x02) control 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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7** : TIE - TXD interrupt enable (generates interrupt when TDRE is set) **6** : LIN - LIN-Mode: LIN break transmit enable (13 bit break symbol instead of 10 bit), LIN break receive detection enable (detects a 11 bit break symbol instead of 10 bit) **5** : RIE - RXD interrupt enable (generates interrupt when RDRF is set) **4** : BIE - break detection interrupt enable (generates interrupt when BRF is set) **3** : TE - transmitter enable If software clears TE while a transmission is in progress (TC = 0) a) lin=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 TDRE to go high after the last frame before clearing TE. b) lin=1: the transmitter immediately fills the transmit shift register with ones and sets the TDRE flag. Wait until TC = 1 before re-enable the transmitter. **2** : 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 (RDRF) setting RE to '0' during an ongoing transfer can cause erroneous data reception and interrupt generation (RDRF), received data should be ignored **1** : MFIE - measurement finish interrupt enable (generates interrupt when MF is set) **0** : SBK - send break bit Toggling SBK sends one break character (10 logic 0s, respectively 13 logic 0s if LINT is set). Toggling implies clearing the SBK bit before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10 bits respectively 13 bit). reset value: 0x0000															

Table 5.3.2.5.10-5: Register **STATUS** (0x04) status 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	R	R	R	R	R	R	R	R	R
Bit Description	**9** : AB_TRIG - AUTO_BAUD_TRIGGERED set when new Baud value was copied automatically to baud config register after a valid SNYC byte (0x55) measurement (see also measurement control register -> AUTO_BAUD) cleared when reading the MSB of the status word Note: If a SYNC byte error is detected during auto baud measurement the measurement finish flag MF will be set but the ABT flag will not be set. **8** : AM_TRIG - AUTO_MEAS_TRIGGERED set when measurement was started automatically after reception of a valid break (see also measurement control register -> AUTO_MEAS) cleared when reading the MSB of the status word **7** : TDRE - transmit data register empty Clear TDRE by writing to SCI data reg. Write will be ignored when transmit register is not empty -> check if TDRE = 1 before writing to transmit register **6** : TC - transmit complete flag TC is reset to '0' when a transmission is in progress **5** : RDRF - receive data register full flag Clear RDRF by reading SCI status with RDRF set and then reading SCI data reg NOTE: RDRF will be set: a) in case of data reception: 1/8 nominal bit length after the recognized stop bit, e.g. since the bits are sampled in the middle of a nominal bit length the flags and with it the irq will be set after the estimated end of the active stop bit. b) in case of break reception: see BRF description below **4** : BRF - break received flag (LIN-Mode dependent) Clear BRF by reading SCI status with BRF set and then reading SCI data reg. The BR flag will be set when the start bit is followed by 8 (respectively 9 when LIN-Mode is set) logic 0 data bits and a logic 0 where the stop bit should be. When BRF is set also FE and RDRF will be set, the SCI data register will be cleared Note: flag generation (incl. BRF) will be suppressed when AUTO_MEAS is set **3** : OV - receiver overrun detected Clear OV by reading SCI status with OV set and then reading SCI data reg. OV will be set when a received data byte is not read before the data byte of the next frame or a break character arrives. The second data byte will be disallowed **2** : MRUN - measurement running **1** : MF - measurement finish flag Clear MF by read accessing the measurement counter Note: Measurement logic is halted as long as MF flag is set **0** : FE - framing error flag FE is set when the logic does not detect a logic 1 where the stop bit should be. FE will be set and reset together with RDRF															

Table 5.3.2.5.10-6: Register **DATA_IO** (0x06) data 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 - SCI data register, write for transmitting byte, read received byte															

Table 5.3.2.5.10-7: Register **MEAS_CONTROL** (0x08) measurement control register

	MSB															***LSB***
Content	-	-	13:8						-	-	-	4	3	2	1	0
Reset value	0	0	1	0	1	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	R	R	R/W	R/W	R/W	R/W	R/W
Bit Description	**13:8** : DBC - debounce filter threshold for baud rate measurement (MMODE=0), filter based on system clock period(Tsys_clk) Filter Threshold [6:0] is mapped to register bits: Threshold[6:1] -> MEAS_CONTROL[13:8] Threshold[0] -> forced to logic 1 Example: Debounce filter threshold set to decimal 81 results in a minimum filter delay of 81*Tsys_clk (-> 10.125us@8MHz) **4** : BUS_COLLISION_E bus collision check enable 0: disable function 1: check for LIN SCI bus collision during transmit, the actual transmitted bit will be checked against the actual received bit. In case of a detected collision the transmitter will be halted immediately (TE reset to 0), a running DMA transfer will be stopped, lin_bus_collision irq will be raised **3** : AUTO_BAUD - automatically copy baud measurement result to baud config register after a valid baud measurement (expecting SYNC Byte) --> AUTO_BAUD_TRIGGERED will be set NOTE: During baud measurement the receiver is disabled an therefore no data will be received, only the measurement logic is active which will generate a measurement finish flag (configurable as interrupt) **2** : AUTO_MEAS - automatically start a baud rate measurement after reception of a valid break --> AUTO_MEAS_TRIGGERED will be set NOTE: AUTO_MEAS mode suppresses the flag specific flag generation (see sci_status -> BRF) **1** : MMODE - measurement mode select 0: baud rate measurement, counter runs with system clock and measures time between 4 falling edges (8 bit length are measured), debouncer is enabled NOTE: baud measurement expects a 0x55 data byte to measure, this is the SYNC byte in the LIN protocol 1: break time measurement, counter runs with 16 x baud rate, measures time when RXD line is zero NOTE: only applicable together with MEN control bit **0** : MEN - measurement enable Set to '1' to start a measurement When measurement is finished, MEN bit will be cleared automatically NOTE: When AUTO_MEAS bit is set MEN must not be used NOTE: Writing a '0' to MEN resets the measurement logic and allows a clean restart															

Table 5.3.2.5.10-8: Register **MEAS_COUNTER** (0x0A) measurement 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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit Description	**15:0** : count - MEASUREMENT COUNTER Counter is cleared by every start of a new measurement When the measurement counter overflows the counter value is saturated to 0xFFFF and the measurement will be stopped (MF flag set). The measurement should be repeated with an adapted baud rate setting. Note: In Baud measurement mode the result of the baud measurement (8 bit length) is divided by 4 and rounded (resulting 2 bit length value) the resulting 16 bit can be fed into the baud divider register to adjust the baud rate.															

Table 5.3.2.5.10-9: Register **LIN_CONFIG** (0x0C) LIN SCI Configuration

	MSB															***LSB***
Content	-	-	-	-	-	-	-	-	7:4				3	2	1	0
Reset value	0	0	0	0	0	0	0	0	0	0	1	0	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** : 0x0 and 0x8 reserved for former products **3** : DMA module implemented **2** : SCI internal timer module implemented **1** : TXD time-out enable register implemented **0** : Concurrent break measurement implemented															

Table 5.3.2.5.10-10: Register **LIN_MODE** (0x0E) LIN Mode 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	1	1
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
Bit Description	**8** : TXD time-out enable Enable TXD time-out counter 12ms timeout counter: In case of an active txd signal for more than 12ms the txd output will be forced to 1. See txd_timeout_irq flag below. **7** : timer_clk_base Timer counts with 0: 1us clock 1: 16 x baud rate **6:5** : Timer Prepare 0: normal operation 1: restart timer from 0 when a falling RXD edge is detected. timer_prepare bit will be reset immediately 2: restart timer from 0 when a falling RXD edge is detected. timer_prepare bit will be reset 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 **4** : Timer Enable 0: timer not running(reset counter to 0) 1: timer running timer counter is incremented by timer_clk_base **3** : High speed - unused, use signal from HV_Control Block **2** : SCI disable 0: SCI enabled, use SCI generated TXD 1: SCI transmission disabled, incoming RXD signal forced to '1', use txd_val as TXD signal (allows CPU generated TXD) **1** : TXD Value Txd value taken from register **0** : RXD Value Receiver signal (direct input)															

Table 5.3.2.5.10-11: Register **ADDON_IRQ_EN** (0x10) Add-on Interrupt Enable

	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/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7** : lin_bus_collision IRQ enable **6** : rx_dma_finished IRQ enable **5** : tx_dma_finished IRQ enable **4** : sci_timer_ov IRQ enable **3** : sci_timer_cmp IRQ enable **2** : txd_timeout IRQ enable **1** : rxd_rising IRQ enable **0** : rxd_falling IRQ enable															

Table 5.3.2.5.10-12: Register **ADDON_IRQ_STAT** (0x12) Add-on Interrupt Status

	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/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit Description	**7** : lin_bus_collision LIN bus collision detected, write '1' to clear interrupt **6** : rx_dma_finished Receive DMA transfers finished, e.g. dma_rx_length = 0 error condition occurred write '1' to clear interrupt **5** : tx_dma_finished Transmit DMA transfers finished, e.g. dma_tx_length = 0 or error condition occurred write '1' to clear interrupt **4** : sci_timer_ov SCI Timer Overflow event write '1' to clear interrupt **3** : sci_timer_cmp SCI Timer Compare event write '1' to clear interrupt **2** : txd_timeout 12ms LIN timeout exceeded write '1' to clear interrupt **1** : rxd_rising RXD line rising edge detected write '1' to clear interrupt **0** : rxd_falling RXD line falling edge detected write '1' to clear interrupt															

Table 5.3.2.5.10-13: Register **TIMER_COUNTER** (0x14) 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 is running when timer is enabled with timer_enable. sci_timer_ov flag is set when timer overflows. 8bit access: read LSB first, MSB data will be stored during LSB read (atomic read)															

Table 5.3.2.5.10-14: Register **TIMER_COMPARE** (0x16) 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 8bit access: write LSB first, value will be written to register with MSB access; read LSB first, MSB data will be stored during LSB read (atomic read/write) Note: Timer compare flag will NOT be set as long as timer_prepare > 0. Timer overflow events will be generated															

Table 5.3.2.5.10-15: Register **DMA_TX_ADDRESS** (0x18) 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. For 8 bit write access write LSB first then MSB. Note: Address is incremented with each DMA transfer executed.															

Table 5.3.2.5.10-16: Register **DMA_TX_LENGTH** (0x1A) 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 5.3.2.5.10-17: Register **DMA_RX_ADDRESS** (0x1C) 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. For 8 bit write access write LSB first then MSB. Note: Address is incremented with each DMA transfer executed.															

Table 5.3.2.5.10-18: Register **DMA_RX_LENGTH** (0x1E) 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 5.3.2.5.10-19: Register **IRQ_STATUS** (0x30) IRQ status register

	MSB															***LSB***
Content	-	-	-	-	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/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** : bus_collision (level) - (event is held by SCI internal logic and has to be cleared otherwise) **10** : rx_dma_finished (level) - (event is held by SCI internal logic and has to be cleared otherwise) **9** : tx_dma_finished (level) - (event is held by SCI internal logic and has to be cleared otherwise) **8** : sci_timer_ov (level) - (event is held by SCI internal logic and has to be cleared otherwise) **7** : sci_timer_cmp (level) - (event is held by SCI internal logic and has to be cleared otherwise) **6** : txd_timeout (level) - (event is held by SCI internal logic and has to be cleared otherwise) **5** : rxd_rising (level) - (event is held by SCI internal logic and has to be cleared otherwise) **4** : rxd_falling (level) - (event is held by SCI internal logic and has to be cleared otherwise) **3** : transmit (level) - transmit data register empty **2** : receive (level) - receive data register full **1** : evt_meas (level) - measurement finish (event is held by SCI internal logic and has to be cleared otherwise) **0** : evt_break (level) - break received (event is held by SCI internal logic and has to be cleared otherwise)															

Table 5.3.2.5.10-20: Register **IRQ_MASK** (0x34) IRQ mask 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** : mask - enable IRQ source 1: enabled 0: disabled NOTE: IRQ_STATUS read: unmasked status of all pending IRQs 1: pending 0: no request NOTE: IRQ STATUS write: clear event flags 1: clear related flag 0: do not change flag															

Table 5.3.2.5.10-21: 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 5.3.2.5.10-22: 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 5.3.2.5.10-23: 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	1	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** : 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 5.3.2.5.10-24: 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	1	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 - 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															

5.3.2.5.11 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

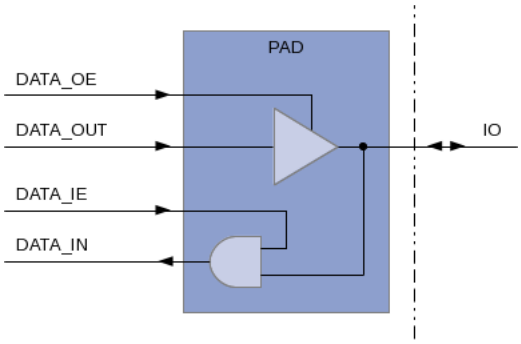

Figure 5.3.2.5.11-1: Structure

Table 5.3.2.5.11-1: 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 5.3.2.5.11-2: 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 5.3.2.5.11-3: 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 5.3.2.5.11-4: 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 5.3.2.5.11-5: 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 5.3.2.5.11-6: 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 5.3.2.5.11-7: 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 5.3.2.5.11-8: 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 5.3.2.5.11-9: 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 5.3.2.5.11-10: 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 5.3.2.5.11-11: 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 5.3.2.5.11-12: 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															

5.3.2.5.12 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

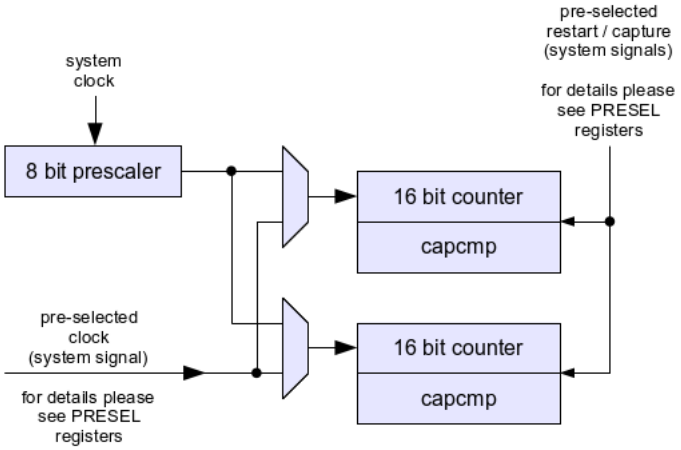

Figure 5.3.2.5.12-1: Structure

Table 5.3.2.5.12-1: 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 5.3.2.5.12-2: 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 5.3.2.5.12-3: 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 5.3.2.5.12-4: 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 5.3.2.5.12-5: 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 5.3.2.5.12-6: 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 5.3.2.5.12-7: 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 5.3.2.5.12-8: 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 5.3.2.5.12-9: 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 5.3.2.5.12-10: 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 5.3.2.5.12-11: 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 5.3.2.5.12-12: 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 5.3.2.5.12-13: 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 5.3.2.5.12-14: 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 5.3.2.5.12-15: 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 5.3.2.5.12-16: 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 5.3.2.5.12-17: 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															

5.3.2.5.13 ADC Control Module (ADC_CTRL)

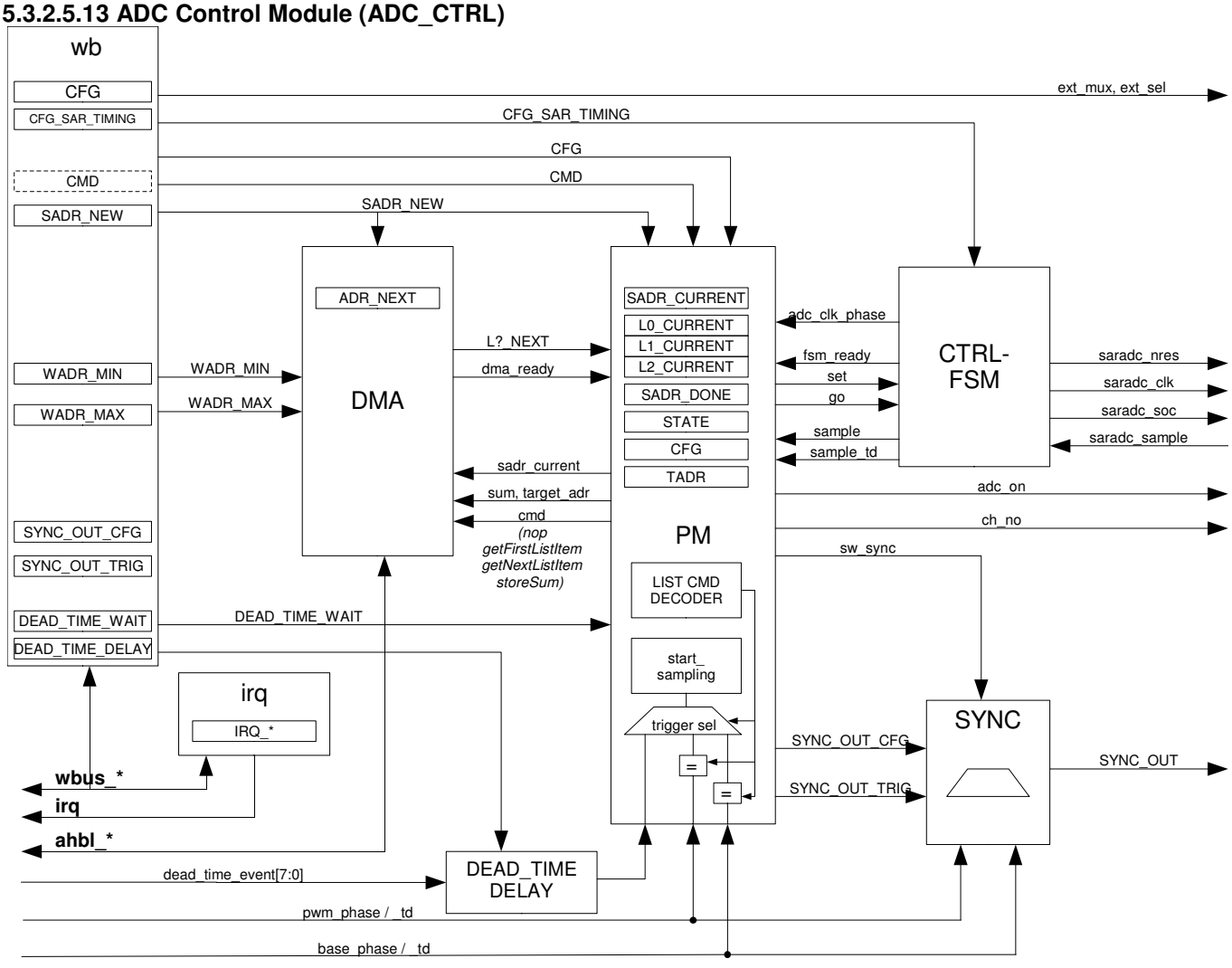

Figure 5.3.2.5.13-1: SAR ADC Control Operating Environment

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 5.3.2.5.13-1: 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
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 5.3.2.5.13-2: 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 5.3.2.5.13-3: 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_TRIG-GER	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 5.3.2.5.13-4: 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 5.3.2.5.13-5: 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 5.3.2.5.13-6: 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 5.3.2.5.13-7: 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 5.3.2.5.13-8: 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 5.3.2.5.13-9: 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 5.3.2.5.13-10: 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 5.3.2.5.13-11: 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 5.3.2.5.13-12: 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 5.3.2.5.13-13: 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 5.3.2.5.13-14: 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 5.3.2.5.13-15: 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 5.3.2.5.13-16: 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 5.3.2.5.13-17: 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 5.3.2.5.13-18: 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 5.3.2.5.13-19: 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 5.3.2.5.13-20: 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 5.3.2.5.13-21: 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 5.3.2.5.13-22: 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 5.3.2.5.13-23: 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 5.3.2.5.13-24: 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 5.3.2.5.13-25: 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 5.3.2.5.13-26: 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 5.3.2.5.13-27: 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 5.3.2.5.13-28: 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 5.3.2.5.13-29: 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 5.3.2.5.13-30: 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															

5.3.2.5.14 PRE PWM Module (PRE_PWM)

5.3.2.5.14.1 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.

5.3.2.5.14.2 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

5.3.2.5.14.3 Module block diagram

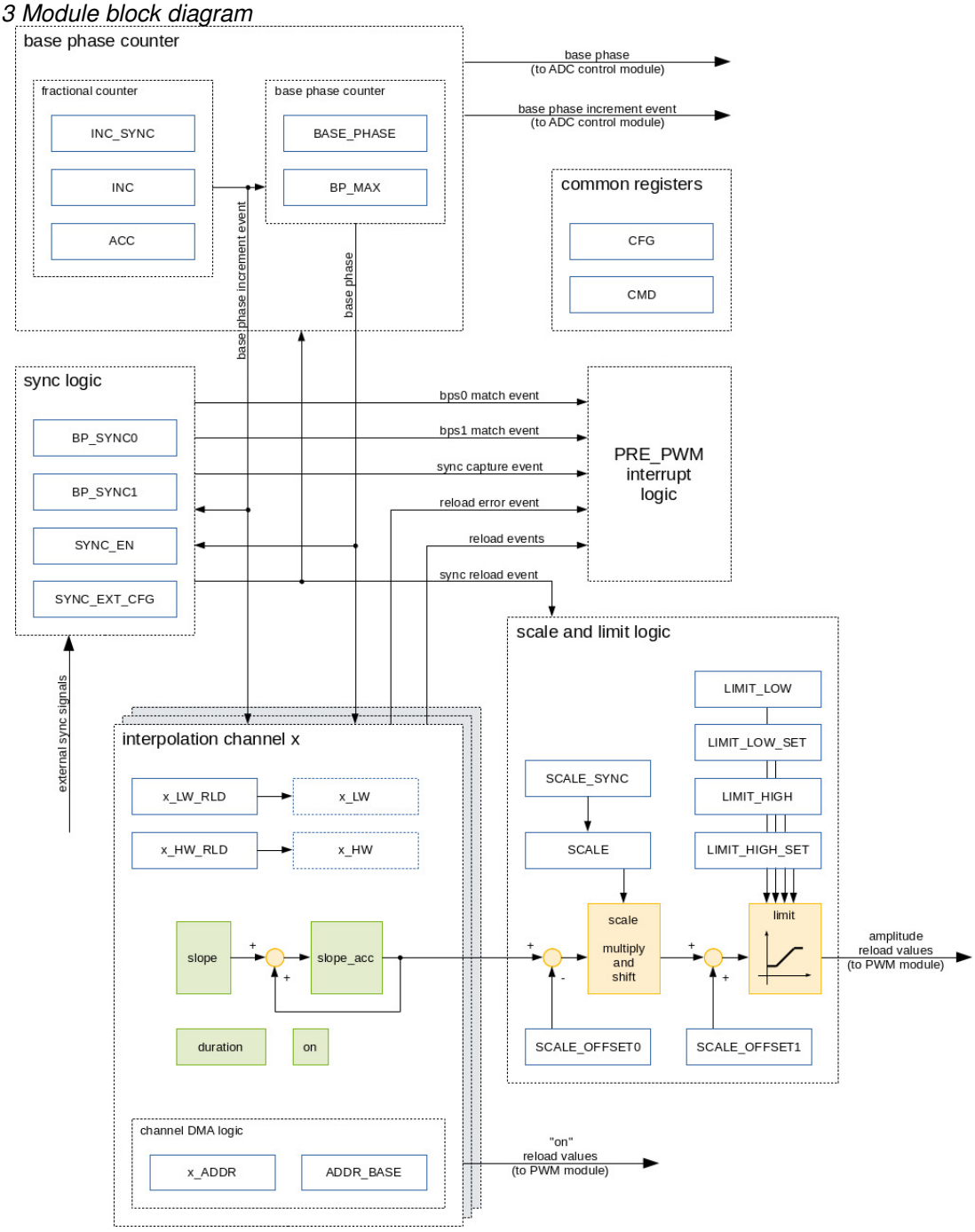

Figure 5.3.2.5.14.3-1: Module block diagram

5.3.2.5.14.4 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

5.3.2.5.14.5 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.

5.3.2.5.14.6 Common registers
- CFG register:
  - configuration to control PRE_PWM module behavior

- CMD register:
  - software commands register

Table 5.3.2.5.14.6-1: Common registers

Register Name	***Address***	***Description***
CFG	0x00	config register
CMD	0x06	command register

Table 5.3.2.5.14.6-2: 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 5.3.2.5.14.6-3: 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															

5.3.2.5.14.7 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 5.3.2.5.14.7-1: Base phase counter

Register Name	***Address***	***Description***
BP_MAX	0x02	base phase max register

Register Name	Address	Description
INC	0x08	increment register
ACC	0x0A	fractional counter register
BASE_PHASE	0x0C	current base phase register
INC_SYNC	0x1C	increment reload register

Table 5.3.2.5.14.7-2: 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 5.3.2.5.14.7-3: 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 5.3.2.5.14.7-4: 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 5.3.2.5.14.7-5: 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 5.3.2.5.14.7-6: 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.															

5.3.2.5.14.8 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 5.3.2.5.14.8-1: 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)

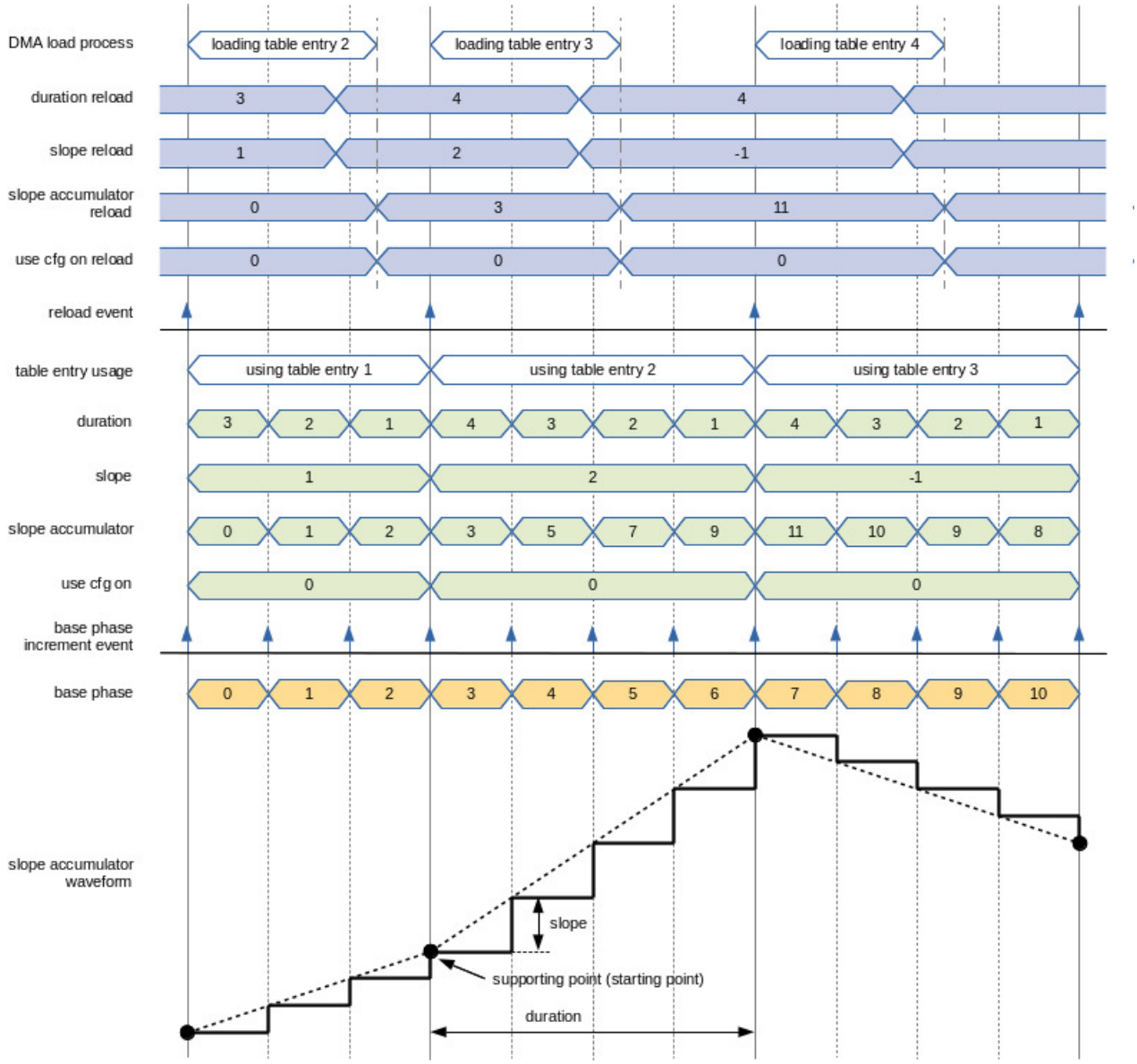

Figure 5.3.2.5.14.8-1: PRE_PWM timing example

Table 5.3.2.5.14.8-2: 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

Register Name	*Address*	*Description*
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 5.3.2.5.14.8-3: 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 5.3.2.5.14.8-4: 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 5.3.2.5.14.8-5: 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 5.3.2.5.14.8-6: 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 5.3.2.5.14.8-7: 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 5.3.2.5.14.8-8: 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 5.3.2.5.14.8-9: 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 5.3.2.5.14.8-10: 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 5.3.2.5.14.8-11: 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 5.3.2.5.14.8-12: 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 5.3.2.5.14.8-13: 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 5.3.2.5.14.8-14: 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 5.3.2.5.14.8-15: 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 5.3.2.5.14.8-16: 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 5.3.2.5.14.8-17: 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 5.3.2.5.14.8-18: 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) !															

5.3.2.5.14.9 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

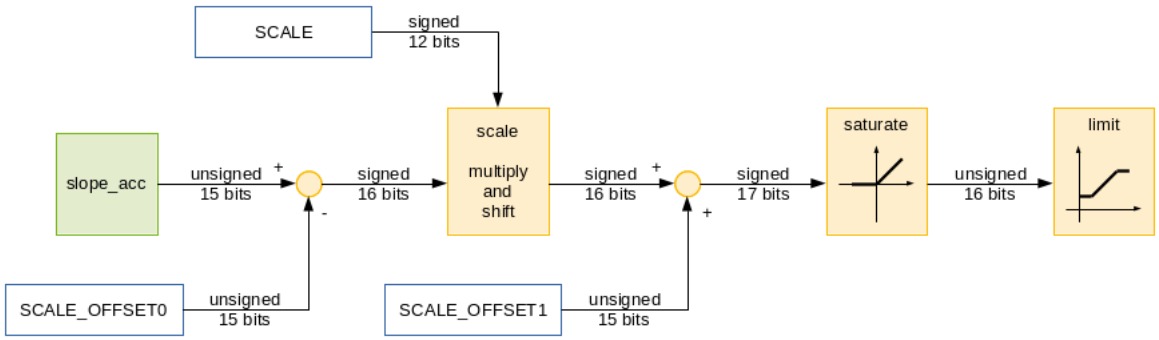

Figure 5.3.2.5.14.9-1: Scale and limit logic

Table 5.3.2.5.14.9-1: 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 5.3.2.5.14.9-2: 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 5.3.2.5.14.9-3: 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 5.3.2.5.14.9-4: 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 5.3.2.5.14.9-5: 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 5.3.2.5.14.9-6: 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 5.3.2.5.14.9-7: 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 5.3.2.5.14.9-8: 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 5.3.2.5.14.9-9: 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															

5.3.2.5.14.10 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 5.3.2.5.14.10-1: Sync logic

Register Name	***Address***	***Description***
BP_SYNC0	0x12	base phase sync 0 register
BP_SYNC1	0x14	base phase sync 1 register
SYNC_EN	0x16	sync enable register
SYNC_EXT_CFG	0x18	external sync config register

Table 5.3.2.5.14.10-2: 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 5.3.2.5.14.10-3: 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 5.3.2.5.14.10-4: 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 5.3.2.5.14.10-5: 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															

5.3.2.5.14.11 Interrupt handling registers

Table 5.3.2.5.14.11-1: 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 5.3.2.5.14.11-2: 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 5.3.2.5.14.11-3: 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 5.3.2.5.14.11-4: 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 5.3.2.5.14.11-5: 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 5.3.2.5.14.11-6: 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 5.3.2.5.14.11-7: 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															

5.3.2.5.15 PWM Module (PWMN)

5.3.2.5.15.1 Description

The PWM module implements a 4 channel 16 bit PWM with dead time insertion and overcurrent handling.

5.3.2.5.15.2 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

5.3.2.5.15.3 Module Block Diagram

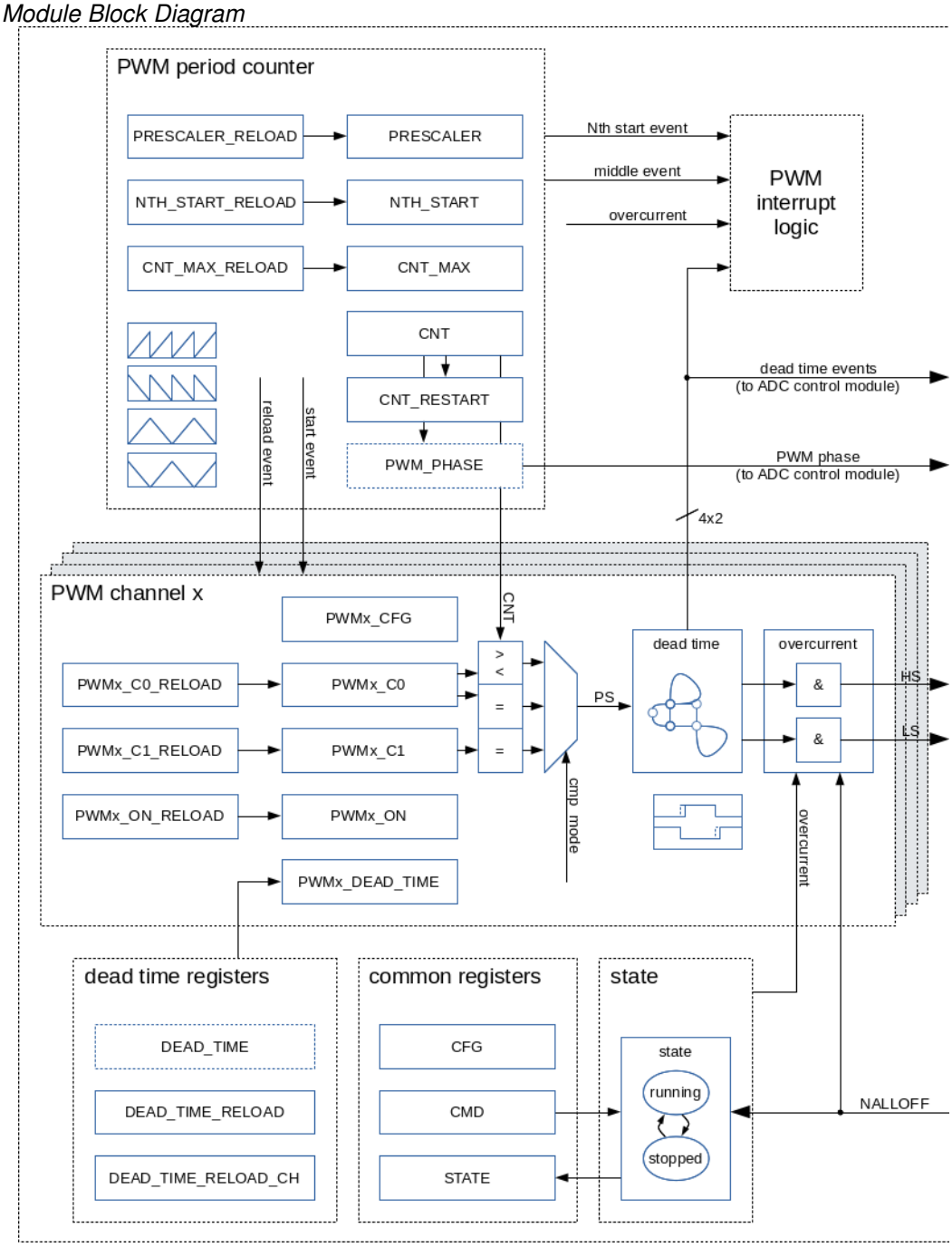

Figure 5.3.2.5.15.3-1: Module Block Diagram

5.3.2.5.15.4 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

5.3.2.5.15.5 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.

5.3.2.5.15.6 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 5.3.2.5.15.6-1: Common registers

Register Name	*Address*	*Description*
CFG	0x00	config register
CMD	0x02	command register

Register Name	*Address*	*Description*
STATE	0x04	state register

Table 5.3.2.5.15.6-2: 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 5.3.2.5.15.6-3: 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 5.3.2.5.15.6-4: 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															

5.3.2.5.15.7 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

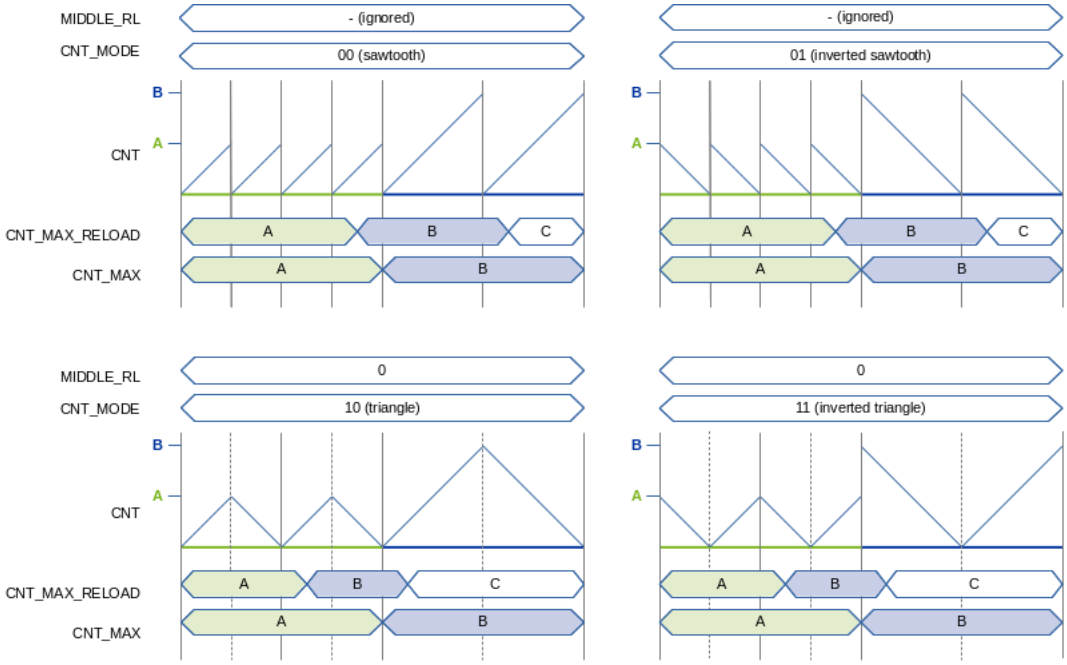

Figure 5.3.2.5.15.7-1: PWM period counter mode dependent CNT_MAX reload behavior (middle_rl = 0)

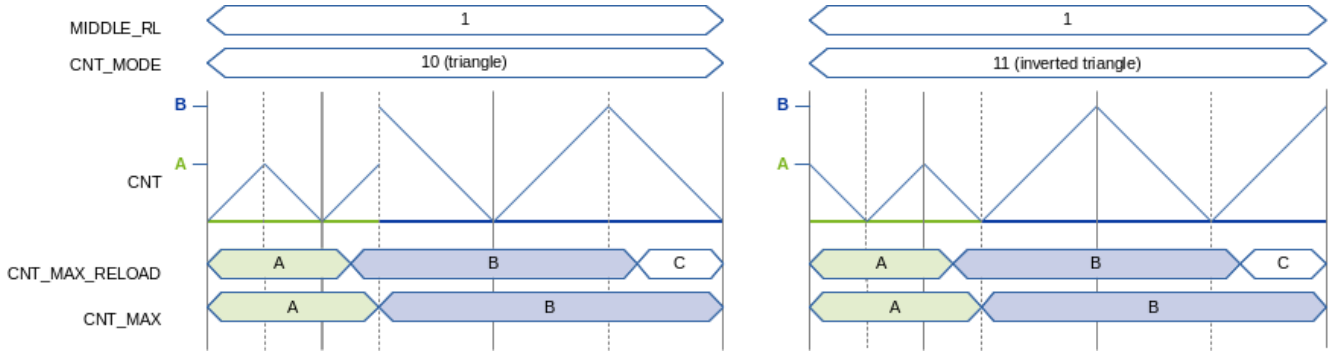

Figure 5.3.2.5.15.7-2: PWM period counter mode dependent CNT_MAX reload behavior (middle_rl = 1)

Table 5.3.2.5.15.7-1: 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
NTH_START_RELOAD	0x1A	n-th start reload value

Table 5.3.2.5.15.7-2: 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 5.3.2.5.15.7-3: 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 5.3.2.5.15.7-4: 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 5.3.2.5.15.7-5: 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 5.3.2.5.15.7-6: 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 5.3.2.5.15.7-7: 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 5.3.2.5.15.7-8: 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 5.3.2.5.15.7-9: 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 5.3.2.5.15.7-10: 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															

5.3.2.5.15.8 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

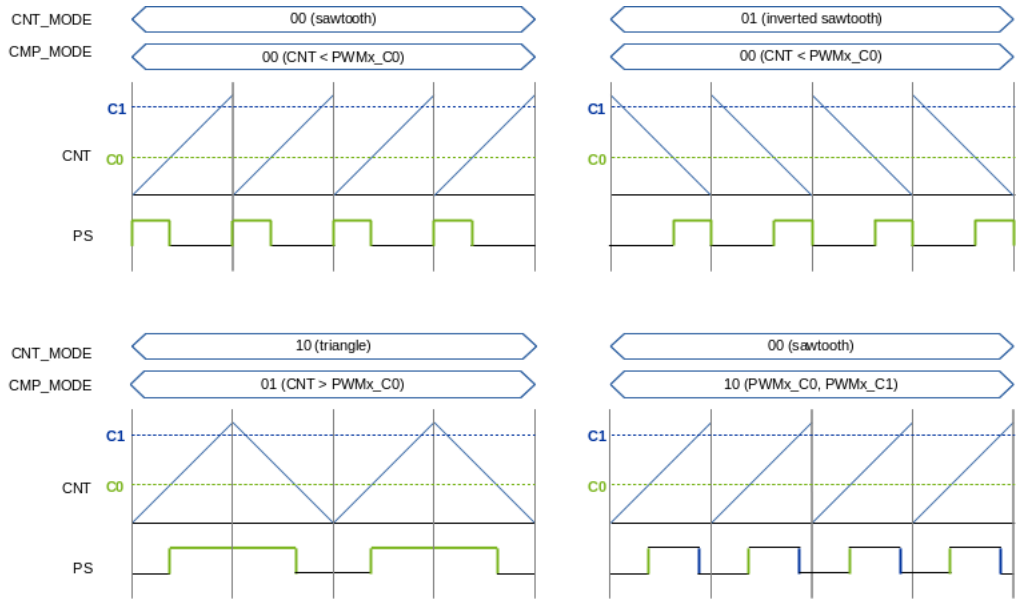

Figure 5.3.2.5.15.8-1: count mode and compare mode dependent PS waveform examples

Table 5.3.2.5.15.8-1: 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

Register Name	*Address*	*Description*
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
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 5.3.2.5.15.8-2: 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 5.3.2.5.15.8-3: 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 5.3.2.5.15.8-4: 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 5.3.2.5.15.8-5: 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 5.3.2.5.15.8-6: 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 5.3.2.5.15.8-7: 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 5.3.2.5.15.8-8: 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 5.3.2.5.15.8-9: 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 5.3.2.5.15.8-10: 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 5.3.2.5.15.8-11: 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 5.3.2.5.15.8-12: 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 5.3.2.5.15.8-13: 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 5.3.2.5.15.8-14: 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 5.3.2.5.15.8-15: 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 5.3.2.5.15.8-16: 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 5.3.2.5.15.8-17: 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 5.3.2.5.15.8-18: 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 5.3.2.5.15.8-19: 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 5.3.2.5.15.8-20: 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 5.3.2.5.15.8-21: 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 5.3.2.5.15.8-22: 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 5.3.2.5.15.8-23: 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 5.3.2.5.15.8-24: 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 5.3.2.5.15.8-25: 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 5.3.2.5.15.8-26: 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 5.3.2.5.15.8-27: 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 5.3.2.5.15.8-28: 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 5.3.2.5.15.8-29: 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 5.3.2.5.15.8-30: 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 5.3.2.5.15.8-31: 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 5.3.2.5.15.8-32: 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 5.3.2.5.15.8-33: 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															

5.3.2.5.15.9 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.

5.3.2.5.15.10 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.

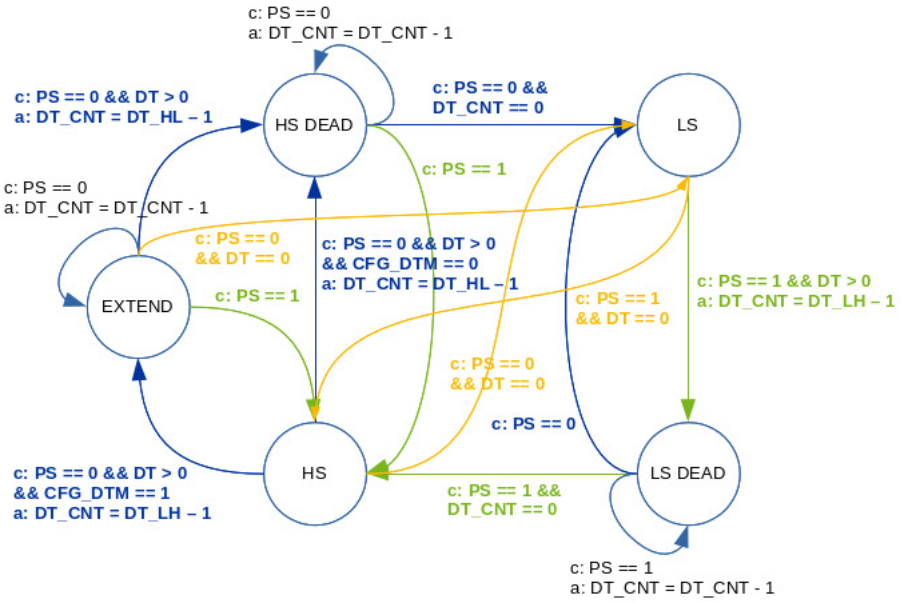

Figure 5.3.2.5.15.10-1: dead time insertion state machine

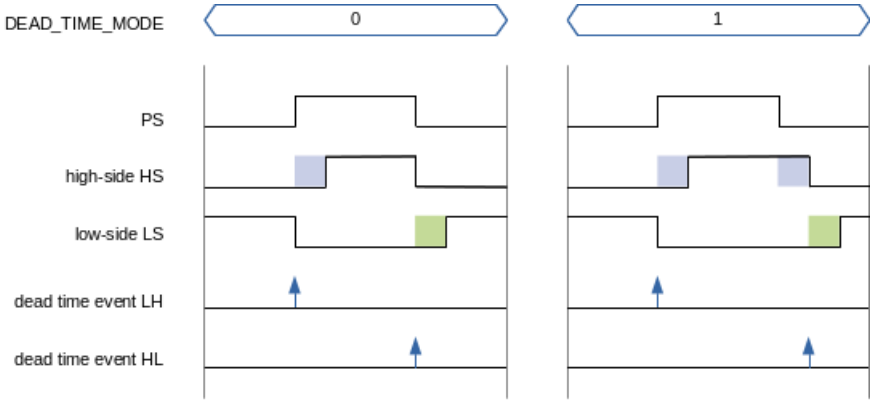

Figure 5.3.2.5.15.10-2: dead time modes

Table 5.3.2.5.15.10-1: 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 5.3.2.5.15.10-2: 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 5.3.2.5.15.10-3: 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 5.3.2.5.15.10-4: 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															

5.3.2.5.15.11 Interrupt handling registers

Table 5.3.2.5.15.11-1: 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 5.3.2.5.15.11-2: 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 5.3.2.5.15.11-3: 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 5.3.2.5.15.11-4: 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 5.3.2.5.15.11-5: 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 5.3.2.5.15.11-6: 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	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 5.3.2.5.15.11-7: 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	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															

5.4 System Errors

Two types of errors can be detected by the E523.52: errors concerning the motor driver section of the IC and errors concerning the microcontroller.

5.4.1 Motor driver errors

The integrated motor driver contains a number of supervisory circuits to protect the B6 bridge and to report error conditions. In section "Monitoring and Safety Functions" a detailed description of potential failures and their rectification is available. VG under-voltage, VDD under-voltage, motor over-current and over-temperatureare checked. Reaction to these conditions can be configured based on the severity of each failure occurrence.

5.4.2 Microcontroller errors

To allow for fail-safe operation of the E523.52, a number of microprocessor internal fault conditions cause system errors which may either force an interrupt or a system reset:

Table 5.4.2-1: Microcontroller errors

Error Name	***Effect***	***Explanation***
Power On Reset	Reset	VDD supply rises above 2.2V(typ)
Brownout	Reset	VDD below 2.9V(max) , or VDDC below 1.5V(max)

Error Name	*Effect*	*Explanation*
CPU Parity	Programmable	Parity error in CPU register
SRAM Parity	Programmable	SRAM access failed
Flash Bit corrected	Interrupt	Flash access has required a bit correction
Flash Bit failure	Programmable	Flash access has resulted in a bit error
Protected Write	Interrupt	Unexpected write access to a protected Flash area
Watchdog	Programmable	An error is asserted if the Software fails to trigger the watchdog timer correctly
Software	Programmable	Software has caused a reset event

6 Typical Application Circuit

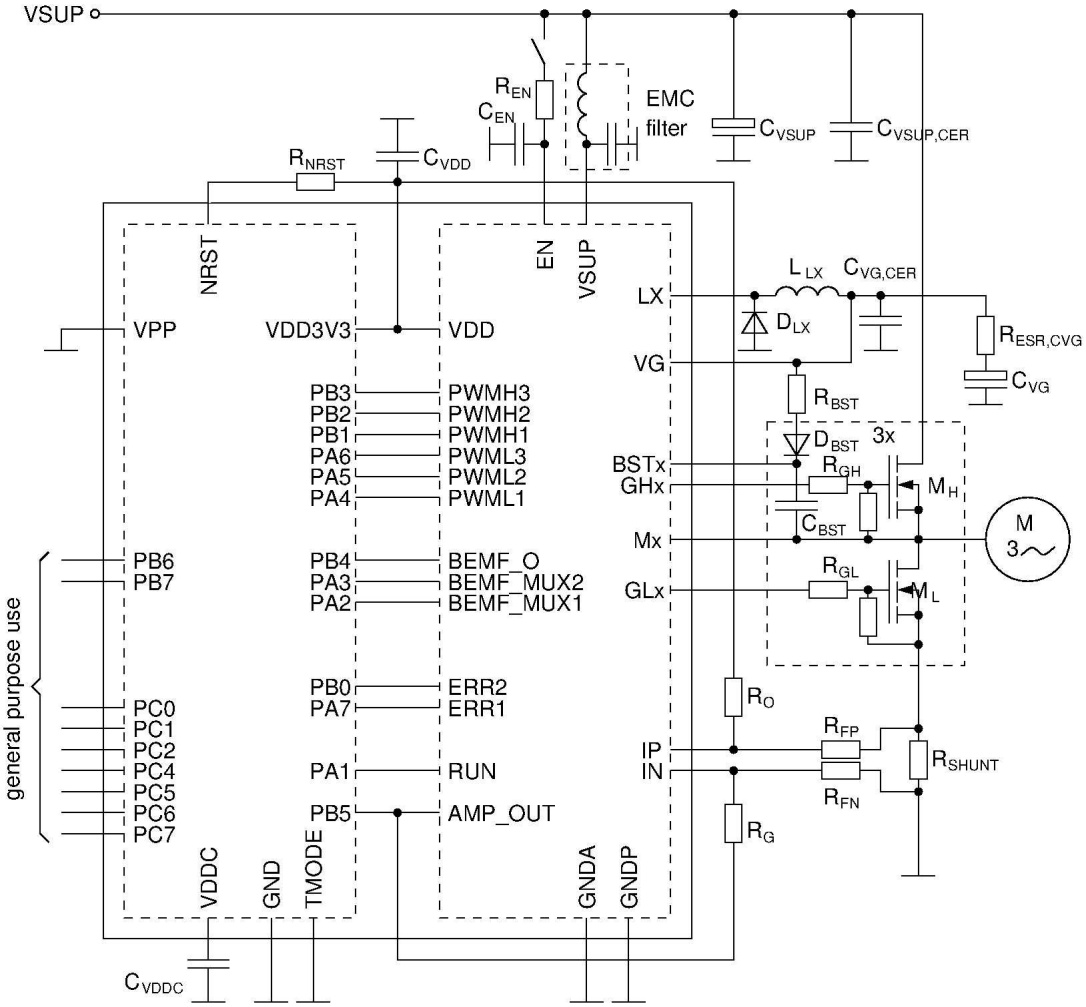

Figure 6-1: Typical Operating Circuit

Thin film Resistors and ceramic capacitors without rating can be 0604(metric) or larger. For more information see "Component Selection" below.

Table 6-1: Component Table

Symbol	Description	Type	Min	Typ	Max	Unit	Rating	Example
$C_{VSUP,CER}$	Supply capacitance HF	ceramic	80	1)		nF	100V	
C_{VSUP}	Supply capacitance LF	electrolytic	10	1)		µF	0.05Ω, 100V	
C_{VDD}	VDD output capacitance	ceramic	0.8	1	1.2	µF		
C_{VDDC}	VDDC output capacitance		220	270	4700	nF		
R_{EN}	Optional enable timing resistor	thinfilm	1	1)	100	kΩ		

Symbol	Description	Type	Min	Typ	Max	Unit	Rating	Example
C_{EN}	Optional enable timing capacitor	ceramic		100		nF	100V	
L_{LX}	Switching inductor for V_{VSUP} = 24V			150µH [1)]		µH	240mA saturation current	WE744777 24-26, EPC B82462A
R_{LX}	Series resistance of LX inductance		1	5	10	Ω		
D_{LX}	Switching diode	Schottky					500mA, 100V	SS1H9/MU H1PB
	Forward voltage of switching diode at LX			0.6	0.8	V		
	Reverse voltage via switching diode at LX		V_{VSUP}					
	Reverse Recovery Time				25	ns		
C_{VG}	Buck output capacitor		26	33	100	µF	25V	
R_{ESR_CVG}	ESR of C_{VG}			1.8		Ω		
$C_{VG,CER}$	VG bypass capacitor	ceramic	10	22	33	nF	25V	
R_{BST}	Inrush current limiter	thinfilm		10		Ω		
C_{BST}	Bootstrap capacitor	ceramic		100		nF		
D_{BST}	Bootstrap diode	bipolar		0.6	0.6	V	500mA, 100V	SS1H9/MU H1PB
R_{GH1}	Gate resistor high-side	thinfilm	30	100		Ω		
R_{GH2}	Gate resistor high-side pulldown	thinfilm			2.2	MΩ		
R_{GL}	Gate resistor low-side	thinfilm	30	100		Ω		
M_L,M_H	Power-MosFET			[1)]			100V	IRFS4610
R_{SN}	Optional snubber resistor	thinfilm		4.7		Ω	0.5W	
C_{SN}	Optional snubber capacitor	ceramic		2.2		nF		
R_{SH}	Current sense shunt resistor	metal	1	[1)]	10	mΩ	≤1%,3W	WSL3637
R_O	Amplifier input offset resistor	thinfilm					≤1%	
R_{FP}	Amplifier input feedback resistor	thinfilm		[1)]		kΩ	≤1%	
R_{FN}	Amplifier input feedback resistor	thinfilm		[1)]		kΩ	≤1%	
R_G	Amplifier gain resistor	thinfilm	17.25	18		kΩ	≤1%	
R_{NRST}	External NRST pullup	thinfilm	4.7		100	kΩ		

[1)] See Component Selection

6.1 Component Selection

The calculations suggested in the following section do not necessarily account for worst case parameters of the components suggested. The user is responsible for worst case calculations by considering min. and max. values in the corresponding datasheets.
For operation purpose, it is strongly forbidden to disconnect the supply voltage while the half-bridges are still connected to the supply voltage. The VSUP pin must always be connected to the drains of the high side power FETs. Any voltage drop >2.5V between a high side power FET drain and the VSUP pin will trigger the VDS desaturation detection and results in switching off the FETs and reporting an error.

6.1.1 Enable Threshold and Timing

It is possible to tune the activation threshold of the IC by attaching a resistor divider to EN. The E523.52 will activate once the voltage V_{EN} is greater than V_{EN_LH} = 2.7V(min). EN has an integrated pulldown current of I_{EN_IN} = 10µA(typ) and a debounce time of T_{DEB_EN} = 100µs(typ). For additional glitch filtering add a filter capacitor C_{EN} to EN. C_{EN} is also recommended to improve the immunity of the application against EMI, ESD and short supply voltage transients.

6.1.2 Supply Filtering

Reverse Polarity

The E523.52 does not have internal reverse polarity protection and has to be protected along with the rest of the application if so desired. Several options are possible:

- **No Protection** This is a useful solution for energy recuperation where the power supply must sink and source energy. But a reverse connection of the supply pins will most likely destroy the application and may result in fire.
- **Diode in the supply path** This simplest method is not very well suited for motor applications as the power dissipation of a diode is significant at higher load currents. Energy returned from the motor cannot fly back into the supply and has to be stored locally in the bypass capacitors. Significant power surges can result in overvoltage and may have to be protected against.
- **FET in the GND path** An N-Channel FET in the GND path is fairly simple to implement with a small pullup resistor from its gate to supply. Care has to be taken in case of voltage surges and significant ground offsets can result relative to the outside world.
- **FET in the supply path** P and N-channel FETs in the supply path are the best solution. P-Channel devices are comparatively more expensive and N-channel FETs are more complex to control to achieve fast turn-on but also turn-off.

C_{VSUP} Dimensioning

The dimensioning of the electrolytic bypass capacitor is more complex. It must absorb supply surges in case of input transients as well as energy fed back by the motor. For ETSI EN 300 132-2 surge protection at least 470µF with less than 50mOhm ESR are required to protect the E523.52 against overvoltage. This represents the minimum size! Its rating is defined by the ripple current of the motor. The peak to peak ripple value equals the peak motor current. It may be necessary to place several capacitors in parallel to increase the overall ripple current performance.
A small HF $C_{VSUP,CER}$ 100nF(typ) ceramic capacitor should be placed across the leads of CSUP to suppress fast transients.

Energy Recuperation

In addition the bypass capacitor may have to absorb the energy returned by the motor to prevent the local voltage from exceeding the maximum possible ratings of 72V. Worst case scenario is a motor that is actively braking and no energy can flow back into the supply because of a reverse polarity device or a line disruption.
In that case the capacitor will have to absorb all the energy present in the motor at the beginning of breaking:

$$C_{SUP} \geq \frac{2 \cdot E_{Mot}}{\left(72V^2 - V_{BATmax}^2\right)}$$

With E_{Mot} being the mechanical energy present in the motor, and V_{BATmax} the maximum possible supply voltage. If the capacitance becomes too large, resistive braking with a brake chopper may be necessary.

Inrush Current Limit Surges

Hot-plugging the application into a low-ohmic high-voltage supply can lead to significant inrush current when the on-board capacitors are charged. This will lead to arcing at the connector and may cause significant input voltage transients at the IC beyond the absolute maximum ratings. It is recommended to limit the inrush current into the application. See the evaluation kit schematic for a possible circuit.

6.1.3 The Step-Down Supply

6.1.3.1 L_{LX} Inductor selection

For the given voltage regulation application (in continuous conducting mode), it is either recommended to choose an inductance value that is suitable for a current ripple of ±30% of the maximum application current (at typical operating frequency) or to use the following equations:

Depending on V_{VSUP} to V_{VG} ratio, usually two equations describe the peak-to-peak current ripple in the inductor L_{LX}. In continuous current mode (CCM), it can either be calculated from the maximum input voltage at VSUP ($V_{VSUP,MAX,APP}$) by

$$I_{RIPPLE,LX,PP1} = \frac{T_{ON,MIN,NOM} \cdot \left(V_{VSUP,MAX,APP} - V_{VG,NOM}\right)}{L_{VG}} \quad (1)$$

or by the following equation (with $V_{F,DLX}$ being the forward voltage drop of the free-wheeling diode)

$$I_{RIPPLE,LX,PP2} = \frac{T_{OFF,MIN,NOM} \cdot \left(V_{VG,NOM} + V_{F,DLX}\right)}{L_{VG}} \quad (2)$$

Formula (1) describes the current ripple in on-time regulation (typ. at high V_{VSUP}), equation (2) is for off-time regulation (typ. at low V_{VSUP}). The maximum current ripple is the maximum from (1) and (2). For the minimum current ripple b) applies. Depending on VSUP to VG voltage ratio the ripple may be symmetric (with minimum and maximum being the same result). Consider tolerances for external components like L_{LX} during calculation.

For maximum ±30mA ripple current and inductor tolerances the calculation results in 6.1.3.1-1 below.

Table 6.1.3.1-1: Minimum recommended Inductances

V_{VSUP}	Min L_{LX}	tolerance
12 V	≥ 120 µH	±20%
24V	≥ 150 µH	±20%
36 V	≥ 270 µH	±20%
48 V	≥ 390 µH	±20%
60 V	≥ 560 µH	±20%
74 V	≥ 680 µH	±20%

The peak-to-peak ripple must be taken into account during choice of the external resistor $R_{ESR,CVG}$ to provide an adequate ripple voltage to the regulation at pin VG (see Recommended Operation Conditions).

To choose a sufficiently high saturation current for the inductor L_{LX}, consider
- the maximum application load current plus half of the maximum ripple current calculated above and
- the current limitation including tolerance (to avoid excessive current peaks in L_{LX} during start up).

In general an additional saturation margin of >10% for the inductor current rating is recommended for transient effects, especially at extreme V_{VSUP} to V_{VG} voltage ratios.

DC resistance of L_{LX} (referred here as R_{LX}) reduces efficiency and contributes to the losses in the inductor (combined with AC losses which may arise due the use of high frequency operation). For the complex impedance of the inductor refer to the vendors information.

At a given load current I_{LOAD}, resistance affects minimum input voltage required to regulate V_{VG} in the following way

$$V_{VSUP,MIN} = V_{VG,NOM} + I_{LOAD} \cdot (R_{ON} + R_{LX})$$

For Voltages below $V_{VSUP,MIN}$ the converter enters the 100% duty-cycle mode.

6.1.3.2 C_{VG} capacitor selection

During nominal operation the serial resistance $R_{ESR,CVG}$ is important to generate a minimum output voltage ripple at V_{VG}. This ripple is required to provide high-frequency switching. Too low ripple voltages may derate the operating frequency stability.
It depends on peripheral elements chosen (L_{LX} and $R_{ESR,VG}$). Basically, it can be calculated using the inductor current ripple and $R_{ESR,CVG}$ (see L_{LX} Inductor Selection) by:

$$V_{RIPPLE,VG} = R_{ESR,VG} \cdot I_{RIPPLE,LX,PP}$$

(Using an electrolytic capacitor type the parameter $R_{ESR,VG}$ represents the sum of the ESR of the capacitor and the external resistor value.)

To reduce output voltage noise, the electrical serial inductance should be kept low. This can be achieved by placing a parallel ceramic type capacitor C_{VG_cer} of typ. 22nF (≤33nF). The main capacitance should consider the following equation.

The intrinsic output capacitors voltage variation during one cycle must be neglectable compared to the AC voltage ripple caused by the $R_{ESR,CVG}$. This dimensioning avoids an out-of-phase ripple voltage signal for the regulation and ensures controlled operating frequency. With the assumption of $T_{PERIOD,MAX} = 1/f_{OP,MIN}$, the resulting recommended minimum value for the capacitor is:

$$C_{VG} \geq \frac{4 \cdot T_{PERIOD,MAX}}{R_{ESR,VG}}$$

Both the result of the equation above and the minimum recommended capacitance for the B6 boot strapping of 26µF have to be taken into account.

Remark:
In General, take into account, that many capacitor types show a strong temperature and voltage dependency, or may be sensible to high peak currents. Make sure, that at extreme temperatures and voltages the capacitance value is reached. For automotive environments capacitors of X7R material (or better) may be necessary.

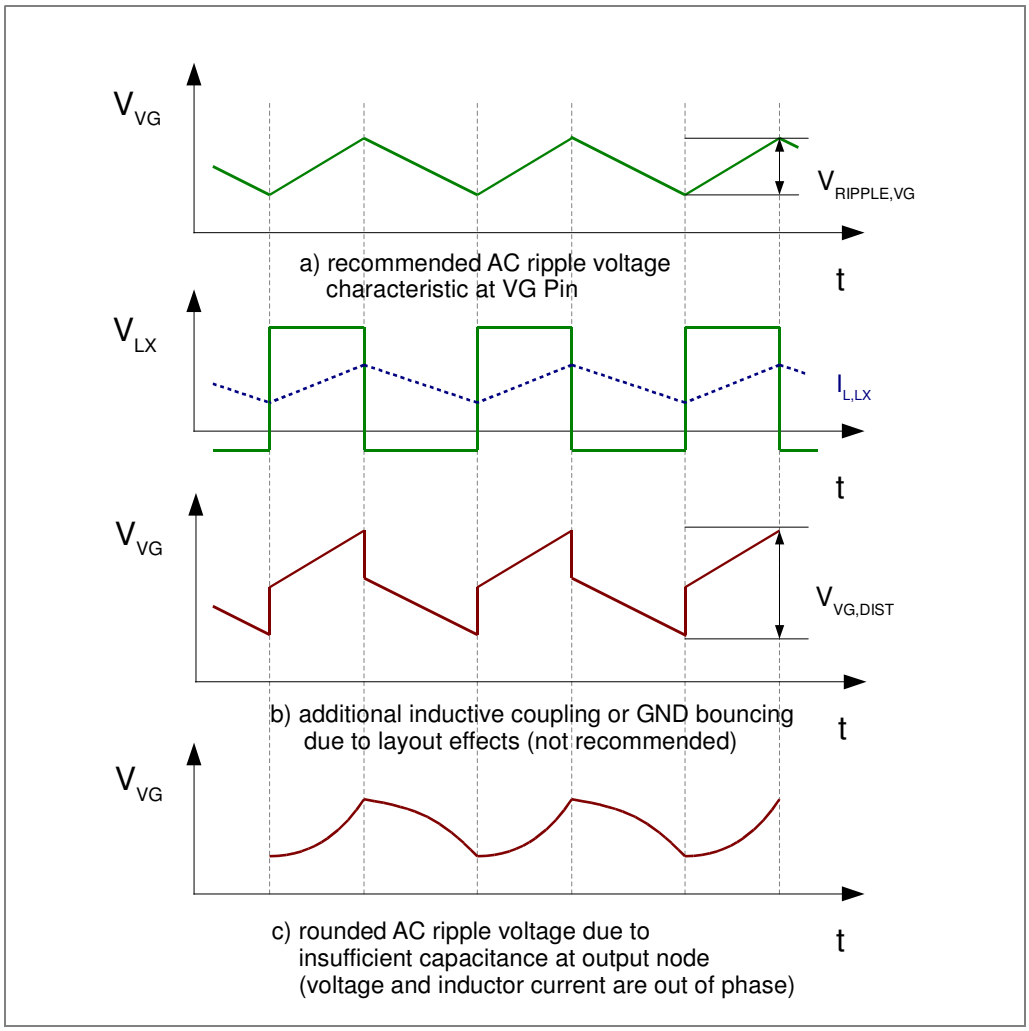

Figure 6.1.3.2-1: AC Output Voltage Ripple Characteristics

The operating frequency characteristic as shown in 5.2.1.2-1 requires a correct AC ripple as shown in the Figure 6.1.3.2-1 above (case a) at the pin VG). This characteristic is the recommended AC voltage which is a valid signal form to fulfil $V_{RIPPLE,VG}$ in the recommended operating conditions chapter.

Characteristics as given in b) are most probable a result of direct inductive coupling between the feedback loop and the changing magnetic field in the inductor L_{LX}. Other potential sources of this type of distortion may be GND distortions caused by changing current flows in the input circuitry or GND loop formed by PGND and the freewheeling diode at LX. Note, that the ripple voltage $V_{VG,DIST}$ is not suited to fulfil the recommended operating conditions $V_{RIPPLE,VG}$. (no mono frequent operation)
This behaviour can potentially be avoided by routing of the feedback signal.

Third case c) is caused by low output capacitance C_{VG}. It does not meet the requirement to have an AC ripple voltage at the output which is in phase to the inductor current. To avoid this it is possible to either increase $R_{ESR,CVG}$ (thus increasing the overall ripple) or to increase C_{VG}. If the voltage change at C_{VG} becomes dominant compared to the signal generated at $R_{ESR,CVG}$ the operating frequency may be a sinusoidal function of L_{LX} and C_{VG} instead of being controlled by E523.52.

6.1.3.3 Rectification-/Freewheeling Diode Selection

The free-wheeling diode should have a low forward voltage (to increase efficiency) as well as a very low reverse recovery time of typically 10ns and a low junction capacitance.

Parameters like parasitic capacitance as well as longer reverse recovery time may cause additional radiated emission at LX. Choose these parameters as low as possible. Parasitic capacitance of this diode decreases the overall efficiency and causes current spikes when the LX driver turns on, while long reverse recovery delays of the freewheeling diode may unintentionally conduct the LX ESD protection circuit.

For high ambient temperatures and/or high supply voltages at VSUP, ultra-fast bipolar rectifiers may perform better than schottky diodes due to the lower reverse leakage currents and lower junction capacitance. Reverse and forward recovery time of the diode must be taken into account.

6.1.3.4 Initial dimensioning calculation example

$V_{VSUP,APP,MAX}$ = 48V. → L_{LX} = 390µH → $T_{PERIOD,MAX}$ = 2.083µs
$V_{VSUP,APP,MIN}$ = 24V.
$I_{LOAD,DC}$ = 100mA

$$I_{PP1,48} = \frac{T_{ON,MIN,NOM} \cdot (V_{VSUP,APP,MAX} - V_{VG,NOM})}{L_{LX}} = \frac{500ns \cdot (48V - 11V)}{390\mu H} = 47.4359mA$$
$$I_{PP1,24} = \frac{T_{ON,MIN,NOM} \cdot (V_{VSUP,APP,MAX} - V_{VG,NOM})}{L_{LX}} = \frac{500ns \cdot (24V - 11V)}{390\mu H} = 16.6667mA$$
$$I_{PP2} = \frac{T_{OFF,MIN,NOM} \cdot (V_{VG,NOM} + V_{F,DLX})}{L_{VG}} = \frac{500ns \cdot (11V + 0.8V)}{390\mu H} = 15.1282mA$$

$$I_{L,PP,MAX} = max[I_{PP1,48}, I_{PP1,24}, I_{PP2}] = 47.4359mA$$
$$I_{L,PP,MIN} = min[I_{PP1,48}, I_{PP1,24}, I_{PP2}] = 15.1282mA$$

$$R_{ESR,CVG,MAX} = \frac{V_{RIPPLE,VG,MAX}}{I_{L,PP,MAX}} = \frac{100mV}{47.4359mA} = 2.108\Omega$$
$$R_{ESR,CVG,MIN} = \frac{V_{RIPPLE,VG,MIN}}{I_{L,PP,MIN}} = \frac{20mV}{15.1282mA} = 1.322\Omega$$

→ $R_{ESR,CVG}$ = 1.8Ω (using E12 series)

$$C_{VG,MIN} \geq \frac{4 \cdot 2.083\mu s}{1.8\Omega} \geq 4.63\mu F$$

→ C_{VG} = 26µF (minimum value for bootstrapping.)

6.1.4 Boot-strap Circuit

The bootstrap capacitors C_{BST} transfer their charge to the gates of the external FETs through the high-side drivers. To be on the safe side, their capacitance C_{BST} should be at least an order of magnitude higher than the total gate capacitance of the external high-side FET. For smaller C_{BST} consider the voltage drop during charge distribution. The resulting gate voltage at the external FET V_{Gate} should stay sufficiently high and is governed by the following equation:

$$V_{Gate} = (V_{VG} - V_{Diode}) \cdot \frac{C_{BST}}{C_{BST} + C_{Gate}}$$

with V_{Diode} being the forward voltage of the boostrap diodes D_{BST} and C_{Gate} the gate capacitance of the external FETs.
To reduce inrush currents upon activation of the IC, the boot-strap capacitors are charged through resistor R_{BST} and diode D_{BST} while the motor phase M[n] is at GND potential. R_{BST} should be at least 10 times the series resistance of C_{VG}.

For operation at or near 100% duty cycle a number of additional factors have to be considered. The bootstrap capacitors will loose their charge due to the current I_{BSleak} which is the sum of the input current on BST $I_{BST1\text{-}3\_on}$= 250µA(typ), gate leakage, and diode leakage. Maintaining 100% duty cycle longer than

$$t_{100percent} = \frac{C_{BST} \cdot 2V}{I_{BSleak}}$$

may lead to low enhancement at the corresponding high-side switch.

6.1.5 Gate Resistors

Gate resistors act together with the intrinsic gate capacitance of the external power FETs as an RC circuit which slows the rise times of the gate voltage. Slower turn on and turn off of the power FETs reduces the overshoot and ringing of the phase voltage, and also reduces the energy in the higher harmonics of the switching frequency. The disadvantage of slower rise and fall times are an increase of power dissipation during switching and a reduction of efficiency.
It is also possible to modify turn on and turn off slopes independently by placing a reverse conducting diode with a series resistor across the gate resistors.

6.1.6 Amplifier Dimensioning

The calculations for the current measurement path assume the motor current is to be sampled by the analog to digital converter (ADC) input of an external microcontroller. If an external ADC reference input is available, it is recommended to use the VDD output of the E523.52 as ADC reference. If only internal references are available 2.5V is also a good reference level, but it will reduce the measuring range.

Note: *Always use a zero current ADC reference measurement with RUN = High before turning the motor on to eliminate offsets introduced by component parameter variations!*

The first component to be chosen is the shunt resistor R_{SH} which provides the input voltage for the current measurement path. Its limiting factors are its power dissipation P_{SHmax} and the value of the maximum operational motor current I_{MOTmax} which - in block commutation - defines R_{SH} as:

$$R_{SH} \leq \frac{P_{SHmax}}{I_{MOTmax}^2}$$

For sine commutation use the RMS value of I_{MOTmax} for the power calculation!
Power dissipation should for practical reasons not exceed 3W at 85°C. The chosen shunt resistor must also support the maximum motor current I_{MOTmax}.
The current path gain ASH dimensioning depends on a number of factors:
1. The overcurrent trip threshold $V_{AMP\_OUT,OC}$ of the E523.52.
2. The maximum peak current $I_{MOTpeak}$ of the motor which should trigger overcurrent.
3. The input range V_{ADCin} of the sampling ADC
4. If motor current must be measured uni- or bi-directional.

If the ADC sampling range is from 0V to VDD = 3.3V(typ)motor current can be sampled up to the maximum short-circuit current $I_{MOTpeak}$ of the motor.

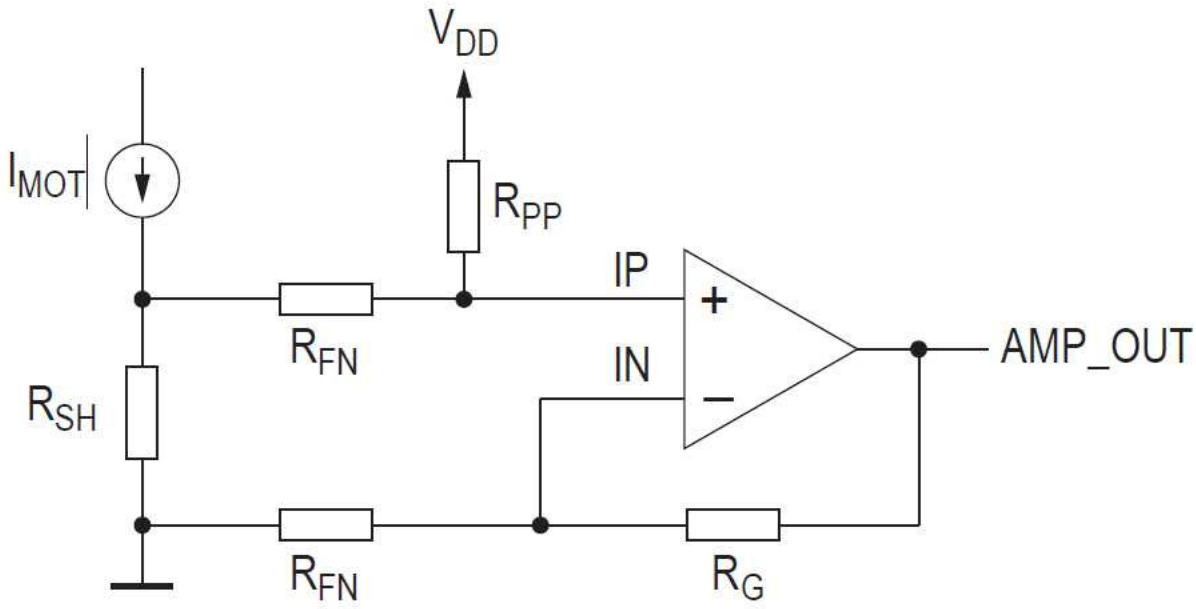

Figure 6.1.6-1: Bi-directional Current Measurement

To make optimum use of the input range of the ADC it is recommended to calculate the network such, that the maximum motor current can still be measured. Calculate the gain as follows:

$$A_{SH} = \frac{V_{REFH}}{2 \cdot I_{MOTmax}} \cdot R_{shunt}$$

and the offset

$$V_{SHoffs} = I_{MOTmax} \cdot R_{shunt} \cdot A_{SH}$$

The resistor gain network with RG = 18kOhm can be calculated as such:

$$R_{FP} = R_{FN} = \frac{R_G}{(A_{SH} - 1)} = \frac{18k\Omega}{(A_{SH} - 1)}$$

In order to place zero motor current in the middle of the full scale range V_{ADC} of the analog digital converter, the voltage offset V_{SHoffs} is applied by populating the offset pullup resistor R_{PP} as:

$$R_{PP} = \frac{V_{DD}}{V_{SHoffs}} \cdot (R_{FN} + R_G) - R_{FN}$$

To calculate the motor current I_{mot} from the voltage V_{ADC} sampled by the microcontroller from AMP_OUT use the following formula:

$$I_{mot} = \frac{\frac{V_{ADC}}{A_{SH}} - V_{DD} \cdot G_P}{R_{SH} \cdot (1 - G_P)}$$

with

$$G_P = \frac{R_{FN}}{R_{FN} + R_{PP}}$$

the gain of the offset resistor network.

6.1.7 Layout Considerations

The E523.52 comes in a thermally effective QFN package. It is important to connect the exposed pad of the package to circuit ground and using a ring of vias to have a good thermal connection into the PCB. To prevent the IC package from being lifted by the surface tension of a solder ball forming under the exposed pad, it might be advisable to use a checker board landing area. The metallized area should be between 50% and 75% of the exposed pad area.
The IC GND pin should be connected to the ground plane directly and not through current carrying traces. A power ground island should connect the ground terminal of the motor bypass capacitances C_{VSUP} and $C_{VSUP,CER}$ and the ground terminal of the shunt resistor. GNDP should be connected to the power ground terminal as well.
Keep switching traces away from sensitive circuits and current carrying traces as wide as possible.

6.1.8 Switching Power-supply

The switching power supply consists of $C_{VG,CER}$, D_{LX}, L_{LX}, and C_{VG}. The ground connections of these components should be as close together as possible to keep the switching current loop small. Use a small via grid to tie the switching ground to the PCB ground plane. The wire loop from LX through L_{LX} to VG should be short. The connection from C_{VG} to VG is less critical.

6.1.9 Square Law Circuit

The traces connected to the pins BST[1-3], GH[1-3], M[1-3], GL[1-3] and GNDP are considered the gate control circuit. Keep the gate control circuitry traces as short and wide as possible to carry peak currents.
Keep the bootstrap components C_{BST} close to the IC.

6.1.10 Current Sensing

The motor current is measured across a shunt resistor in the ground path of the motor. The current sense lines on the shunt side of R_{FN}, R_{FP} MUST be kelvin connected to each end of the shunt resistor and should be routed in parallel to the IC. Do NOT connect R_{FN} to a local ground terminal by the IC. The internal OPAMP is very fast. Place the gain network as close to the IC as possible, to ensure minimum parasitics at the OPAMP pins.

7 Package Reference

The device is assembled in a 7x7mm QFN package with 36 pins. The QFN36L7 package outline and dimensions are according JEDEC MO-220 K.

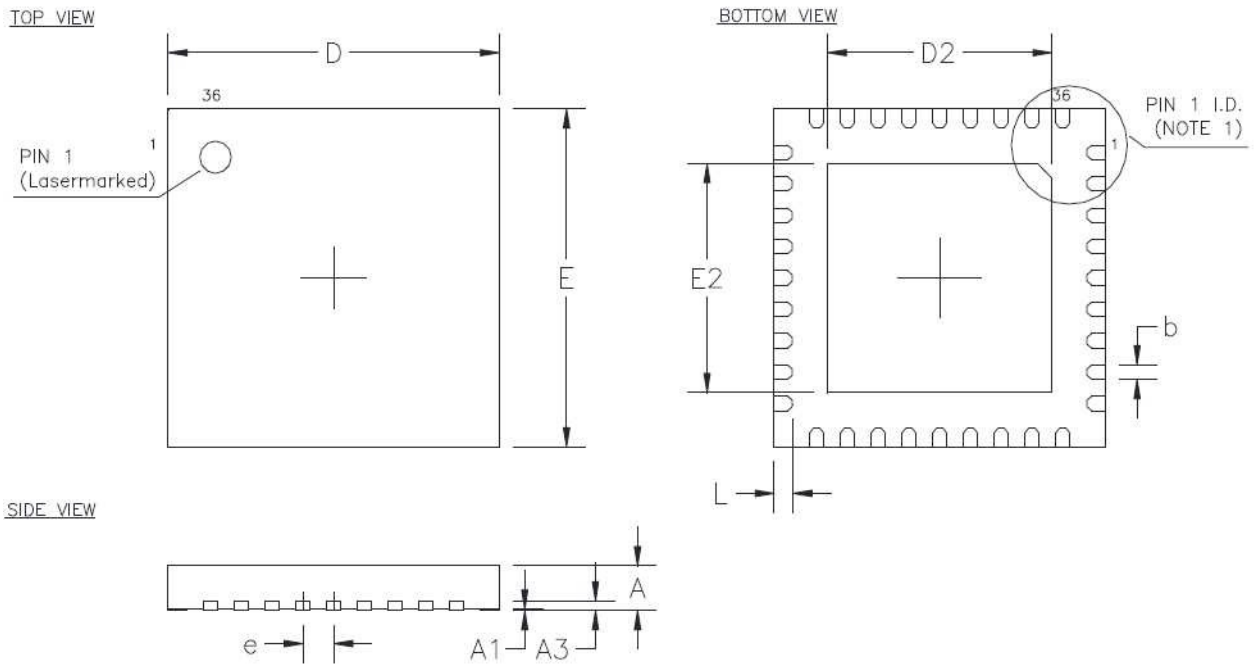

Figure 7-1: Package Outline

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	0.25	0.30	0.35	0.010	0.012	0.014
Package length / width	D / E	--	7.00 BSC	--	--	0.276 BSC	--
Length / width of exposed pad	D2 / E2	4.60	4.75	4.90	0.181	0.187	0.193
Lead pitch	e		0.65 BSC			0.026 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		36			36	

Figure 7-2: Package Dimension Table

Note: The mm values are valid, the inch values contains rounding errors
Note 1: For assembler specific pin1 identification please see QM-document 08SP0363.xx (Pin 1 Specification)

8 ESD, Latchup and EMC

8.1 Electro Static Discharge (ESD)

Table 8.1-1: 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 8.1-2: Optional ESD Test on IC Level for Special Pins

Optional ESD Test	Test equipment similar to AEC-Q100-002
Test point	Pins VSUP, EN to system ground
Capacitance	100 pF
Resistance	1,5 kΩ
Minimum withstand Voltage	+/- 4 kV

Table 8.1-3: 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

8.2 Latch-up

Latch-up performance is validated according JEDEC standard JESD 78 in its valid revision.

9 Marking

9.1 Top Side

Table 9.1-1: Top Side

Elmos Logo
E52352A
XXXXU
YWW**

where

Table 9.1-2: Marking of the Devices

Signature	***Explanation***
E / M / T	Volume production / prototype / test circuit
52352	ELMOS project number
A	ELMOS project revision code
XXXX	Production lot number
U	Assembler code
YWW	Year and week of assembly
* *	ELMOS internal code

9.2 Bottom Side

No marking.

10 General

10.1 Disclaimer

10.1.1 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 utilising 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. ELMOS will not assume neither warranty nor liability for risks resulting from unauthorized use of the product in applications the product is not designated and intended for.

10.1.2 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.

10.1.3 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.

10.1.4 Copyright

11 Storage, Handling, Packing and Shipping

11.1 Storage

Storage conditions should not exceed those given in Chapter 3.1 Absolute Maximum Ratings.
The Moisture Sensitivity Level is specified according to MSL3 (JEDEC J-STD-020 in its valid revision).

11.2 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 3.1 Absolute Maximum Ratings.

11.3 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.

11.4 Shipping

Each delivery shall be accompanied by the following:
- Certificate of conformance to the specification
- Delivery note

12 Record of Revisions

Table 12-1: Record of Revisions

Chapter	*Rev.*	*Description of change*	*Date*	*Released*
all	00	Initial revision	10.11.2015	JOKR/ZOE
all	01	General revision including the microcontroller and motor driver datasheet	16.01.2017	TKL

13 Contact Information

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 ☏: +86216210 0908 ✉: sales-china@elmos.com

Sales and Application Support Office Korea
Elmos Korea
B-1006, U-Space 2, 670 Daewangpangyo-ro,
Sampyoung-dong, Bundang-gu, Seongnam-si,
Gyeonggi-do, 13494 Korea ☏: +82317141131 ✉: sales-korea@elmos.com

Sales and Application Support Office Japan
Elmos Japan K.K.
BR Shibaura N Bldg. 7F
3-20-9 Shibaura, Minato-ku,
Tokyo 108-0023 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

14 Index

Table of Content

Illustration Index