1. Features and Benefits

- Motor driver
  - 3x PreDriver for DC or BLDC motor
  - Up to 200nC NFETs (12V-48V supply)
    - 0.7 A_{pk} typ. charge current
    - 1.5 A_{pk} typ. discharge current
  - charge-pump for top-NFETs
  - V_{DS} protection for all NFETs
- Microcontroller:
  - MLX16-FX, application CPU
  - MLX4, communication CPU
  - Programmable digital watch-dog
  - Interrupt controller
  - Common purpose timer
- Memories split per CPU
  - MLX16-FX memories:
    - MLX81346: 64kByte Flash with ECC
    - 20kByte ROM
    - 4kByte RAM
    - 576Byte EEPROM
  - MLX4 memories:
    - 6kByte ROM
    - 512Byte RAM
- Fast end-of-line programming via LIN pin (64kB Flash in < 4sec)
- Pin-compatible family in QFN32
    - 90KB Flash+ROM
- Periphery
  - Configurable RC-clock 12..32MHz
  - 12x general purpose IO's, digital, analog, 3x high-voltage IO's, 2x UART, SPI, I^2C-slave
  - 2x high-side supply <50mA
  - 5x 16-bit motor PWM timers
  - 2x 16-bit timers
  - 12-bit ADC with < 1µs conversion time with 64 channels
  - Differential current sense amplifier with 8-bit programmable overcurrent
  - Temperature sensor, over-temperature detection
  - Over-current detection, over-voltage and under-voltage protection
- Voltage regulators
  - operating motor voltage VSM= 5.5V-60V
  - operating supply voltage VS= 5.5V-32V* (*operating voltage up to 36V limited to 24h over lifetime)
  - Internal voltage regulators, directly powered from VS supply
  - Operation down to 3.5V with reduced analog characteristics, down to 3.0V without losing register content, down to 1.5V with intact RAM memory
  - Low standby current consumption of typ 25µA in sleep mode
  - Wake-up possible via LIN, external pins or internal wake-up timer
- Bus interface
  - LIN 2.x/SAE J2602 and ISO17987-4 compliant LIN slave
- **Automotive AEC-Q100 Qualified**

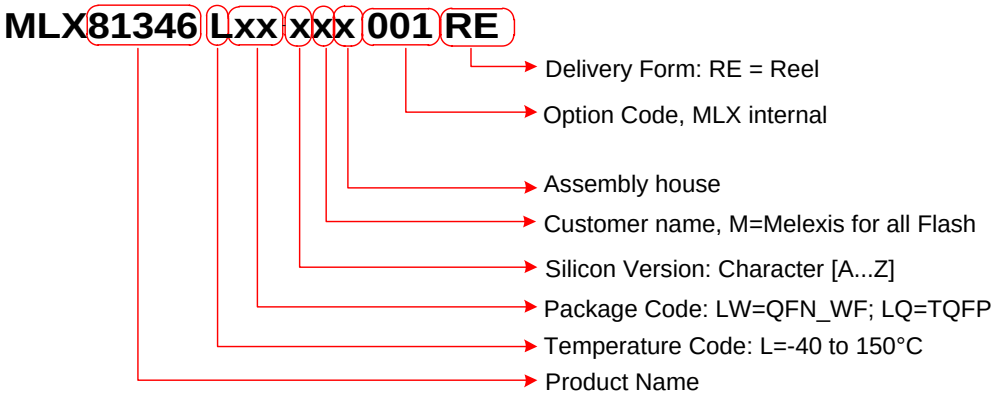
2. Application Examples

- 12V-48V DC/BLDC Engine Cooling Fans, Pumps, Compressors
- 12V-48V DC/BLDC Positioning applications

3. Ordering Information

Order Code	Temp. Range	Package	Delivery	Remark
MLX81346 LLW-BMT-003-RE	-40 - 150 °C	QFN32_WF 5x5	Reel	
MLX81346 LPF-BMA-003-RE	-40 - 150 °C	TQFP48 7x7 eP	Reel	

Table 1 – Ordering Information



4. Revision history

Version	Date	Description
0.5	07/06/2021	• Chapter 3 Ordering Information updated • Figure 19 Block diagram updated • Table 3, Table 4, Table 75 – names TDI/TDO added • Table 6 – Operating range updated • Table 9 – Electrical specifications: VDDD regulator updated • Table 14 – Electrical specifications: charge pump updated • Table 18 – Electrical specifications: OSD updated • Table 24 – Electrical specifications: VDS monitor IO updated • Figure 37, Figure 38 and Figure 39 updated • Table 65 – PORT_MISC2_OUT – added VDSMON_VTH_SEL bits • Chapter 14.2.2.2 updated • Chapter 13.3.4.3 updated: parameter V_CSH_IR for CSA_HIGHGAIN=1 added • Figure 89 — LIN transceiver block diagram updated

Table 2 – Revision History

5. Contents

6. Abbreviations

ADC	Analog Digital converter
API	Application Program Interface
ASSP	Application Specific Standard Product
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
DMA	Direct Memory Access
EEPROM	Electrically Erasable/Programmable Read-Only Memories
ECC	Error Correction Code
ECU	Electronic Control Unit (with µ-Controller/µ-Processor)
GTSM	Go To Sleep Mode – request to enter SLEEP mode
HV	High Voltage Pin
IC	Integrated Circuit
ID	Identifier
IO	Input Output
IP	Intellectual Property
I2C	I^2C interface
LIN	Local Interconnect Network
LSB	Least Significant Bit
LV	Low Voltage Pin
NB	Narrow Body
MCU	Microcontroller Unit
MCU_CLK	main clock of CPU
MSB	Most Significant Bit
OD	Open drain output
OSI	Open Systems Interconnection Model
PHY	Physical Layer
POR	Power on reset
PP	Push-pull stage
PPM	Pulse Position Modulation (physical layer for fast Flash programming)
PWM	Pulse Width Modulator
RAM	Random Access Memory
RCO	RC oscillator clock
ROM	Read Only Memory
SMD	Surface Mount Device
TBD	To Be Defined

7. References

Following documents are referred to in this document:

[1] Calibration Process for CAMCU Projects.

[2] Melexis LIN API documentation, Softdist download area.

[3] MLX16-FX Data book, Softdist download area.

[4] Melexis PPM bootloader specification.

[5] LIN specification package 1.3, LIN consortium, 2002-12-12.

[6] LIN specification package 2.0, LIN consortium, 2003-09-16.

[7] LIN specification package 2.1, LIN consortium, 2006-11-24.

[8] LIN specification package 2.2A, LIN consortium, 2010-12-31.

[9] MLX81346 Safety manual.

[10] Stress test qualification standard AEC-Q100, Automotive Electronics Council, 2003-07-18

This documentation as well as application notes, software tools, libraries and descriptions is not scope of this specification and can be found under http://softdist.melexis.com/. Please contact your Melexis Sales channel for getting access.

The descriptions in this document overrule the descriptions in the referred documents.

8. System block diagram and system functions

The system block diagram is shown in Figure 1, where the key system functions of MLX81346 are illustrated, i.e. the capability to drive the phases of a motor over N-FET halfbridges, to read data from sensors and to communicate with the engine control unit (ECU) over a LIN compliant interface.

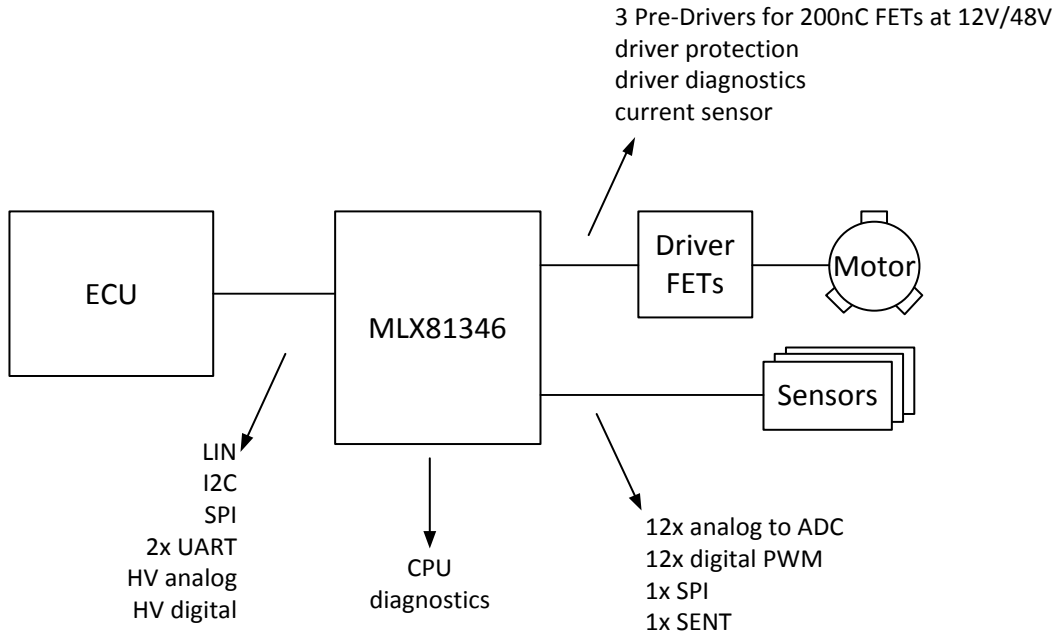


Figure 1 System block diagram

The system functions of MLX81346 are:

- Motor driving:
  - The IC can drive up to 3 half-bridges consisting of 6 N-FETs with max 200nC gate charge to support motors ranging from 10W - 1200W
    - 3-phase BLDC motor
  - The IC can process standard or complex motor drive algorithms
    - Sensor-less FOC (field-oriented control)
    - Sensored FOC (field-oriented control)
- Sensing:
  - The IC senses the motor current over an external shunt
  - The IC can read up to 12 analog outputs of external sensors

  - The IC can read up to 12 digital outputs of external sensors
  - The IC can receive as SPI master the output of an external sensor
  - The IC can receive the SENT output of an external sensor
  - The IC can supply external sensors, limited by maximum supply current <15mA
  - The IC can sense the motor supply and phase voltages
- Communication:
  - The IC supports LIN 2.x, SAE J2602 and ISO17987-4 standards as a slave node
  - The IC supports I^2C Standard-mode, Fast-mode and Fast-mode Plus as a slave node
  - The IC supports the SPI standard
  - The IC can transmit a digital SENT signal
  - The IC can read up to 3 high-voltage analog levels
  - The IC can read up to 3 high-voltage digital signals
  - The IC supports receiving and transmitting a PWM communication signal at the LIN pin
  - The IC supports receiving and transmitting up to 2 UART signals

The diagnostic functions of the IC are covered in the safety manual [9].

9. Functional safety

The MLX81346 is an ASSP and developed as SEooC [ISO 26262] with assumed technical safety requirements with ASIL-B capability targets. The technical safety concept is described in the MLX81346 Safety manual [9].

10. IC Block diagram

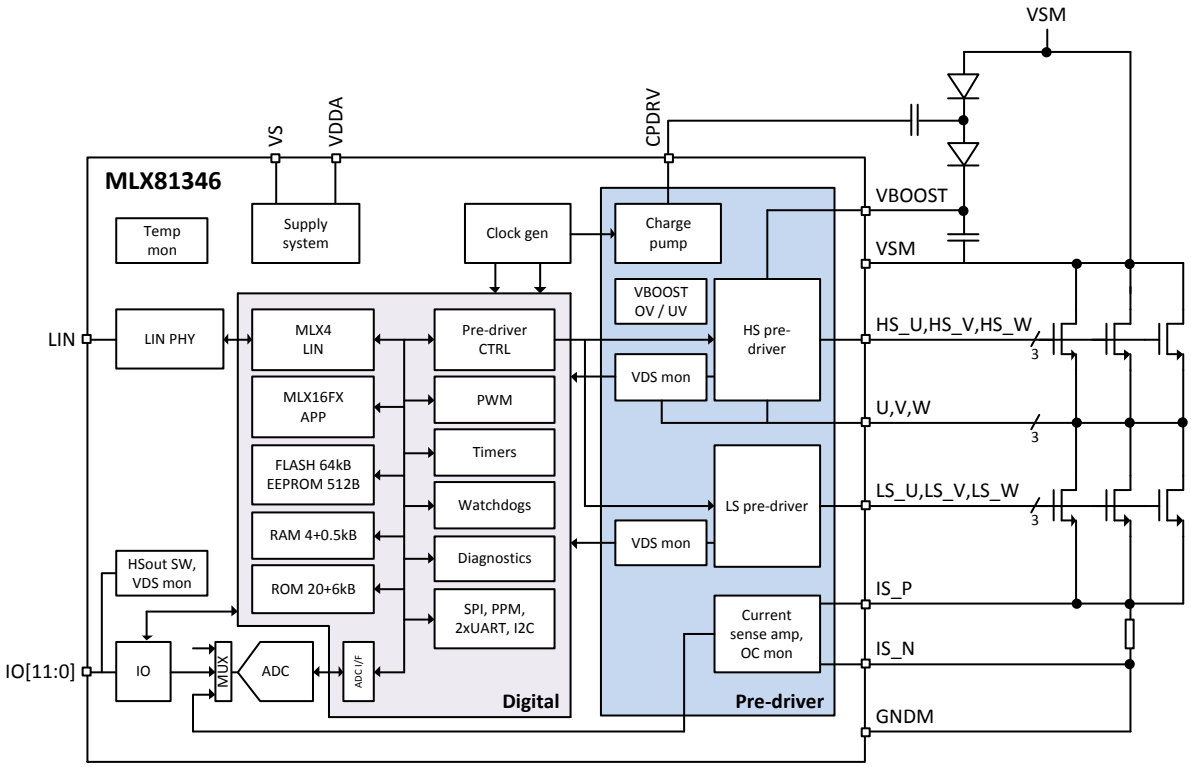
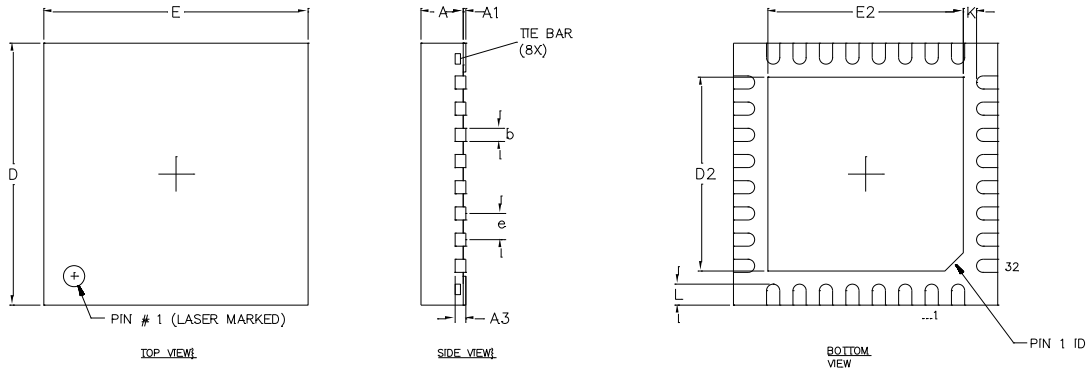


Figure 2 – 81346 IC Block diagram with external power bridge (BLDC)

11. Technical description

11.1. Package data QFN32



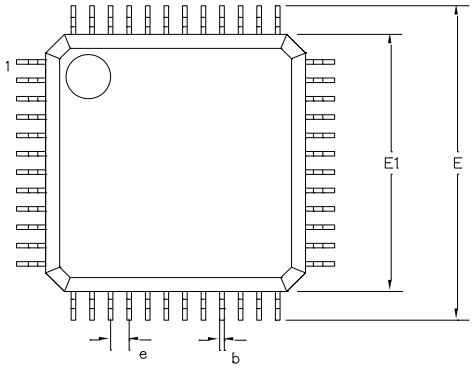
COMMON DIMENSIONS AND TOLERANCES

SYMBOL	ALL DIMENSION ARE IN MILLIMETERS		
	MINIMUM	NOMINAL	MAXIMUM
A	0.80	0.90	1.00
A1	0	0.02	0.05
A3		0.20 REF	
D	4.90	5.00	5.10
E	4.90	5.00	5.10
D2	3.60	3.70	3.80
E2	3.60	3.70	3.80
L	0.30	0.40	0.50
K	0.20	---	---
b	0.18	0.25	0.30
e		0.50 BSC	

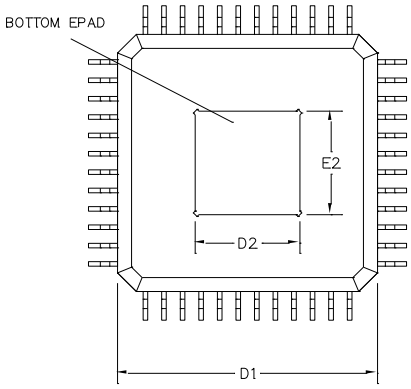
NOTE :
1. EXPOSED TIE BAR SHOULD BE KEPT FREE FROM SOLDER.
2. SIDE WALL IMMERSION TIN PLATING MIN 1um THICK.

Figure 3 – Package data QFN32

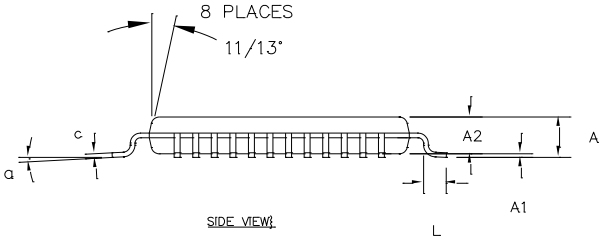
11.2. Package Data – TQFP48



TOP VIEW



BOTTOM VIEW



SIDE VIEW

COMMON DIMENSIONS AND TOLERANCES

SYMBOL	ALL DIMENSION ARE IN MILLIMETERS		
	MINIMUM	NOMINAL	MAXIMUM
A	---	---	1.20
A1	0.05	---	0.15
A2	0.95	1.00	1.05
D		9.00 REF	
E		9.00 REF	
D1	6.90	7.00	7.10
E1	6.90	7.00	7.10
D2		5.00 REF	
E2		5.00 REF	
L	0.45	0.60	0.75
b	0.17	0.22	0.27
c	0.09	---	0.20
e		0.50 BSC	
a	0°	---	7°

Note:
1. DIMENSION D1 AND E1 DO NOT INCLUDE MAX 0.254 mm MOLD PROTRUSION
2. DIMENSION b DOES NOT INCLUDE MAX 0.080 mm DAM BAR PROTRUSION.

Figure 4 – Package data TQFP48 7x7 eP

11.3. Package Pin-out

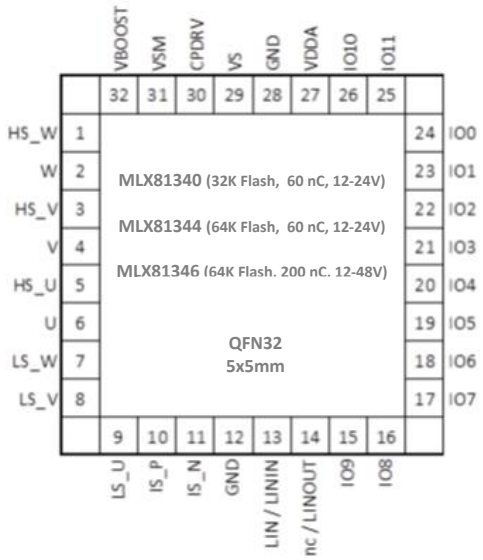


Figure 5 – Pin-out diagram QFN32 (-LLW variant)

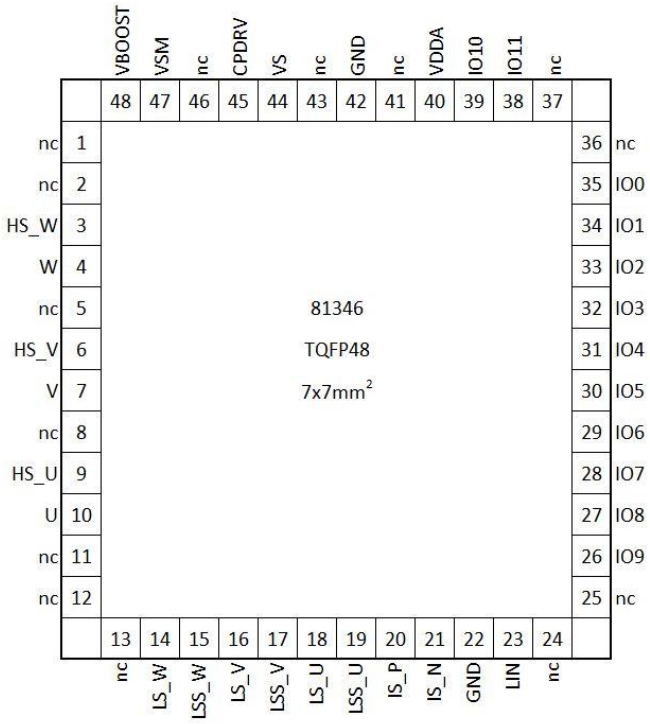


Figure 6 – Pin-out diagram TQFP48 (LPF variant)

11.4. Pin out description

Pin	Pin name	Description	Comment
QFN32, 5x5mm			
1	HS_W	High-Side FET Predriver for W	
2	W	W-phase	
3	HS_V	High-Side FET Predriver for V	
4	V	V-phase	
5	HS_U	High-Side FET Predriver for U	
6	U	U-phase	
7	LS_W	Low-Side FET Predriver for W	
8	LS_V	Low-Side FET Predriver for V	
9	LS_U	Low-Side FET Predriver for U	
10	IS_P	Current sense input (+)	
11	IS_N	Current sense input (-)	
12	GNDM	GND Motor	
13	LIN	LIN interface pin	
14	nc	not connected	
15	IO9	LVIO	
16	IO8	LVIO	
17	IO7	LVIO	1st test interface output TDO
18	IO6	LVIO	1st test interface input TDI
19	IO5	LVIO	
20	IO4	LVIO + HVS (high-voltage supply <50mA)	
21	IO3	LVIO + HVS (high-voltage supply <50mA)	
22	IO2	LVIO + HVIO (high-voltage in/out)	2nd Test interface output TDO
23	IO1	LVIO + HVIO (high-voltage in/out)	2nd Test interface input TDI
24	IO0	LVIO + HVIO (high-voltage in/out)	
25	IO11	LVIO	
26	IO10	LVIO	
27	VDDA	3.3V	3.3V IO supply, option for external sensors ≤ 25mA

Pin	Pin name	Description	Comment
QFN32, 5x5mm			
28	GND	GND IC	
29	VS	Supply IC	For 48V : connect 12V external supply to VS
30	CPDRV	Driver for charge pump	
31	VSM	Supply Motor	
32	VBOOST	Boost voltage to drive High-Side FETs	

Table 3 – Pin-out description for QFN32

#	Pin name	Description	Comment
TQFP48, 7x7mm			
1	NC	Not connected	
2	NC	Not connected	
3	HS_W	High-Side FET Predriver for W	
4	W	W phase	
5	NC	Not connected	
6	HS_V	High-Side FET Predriver for V	
7	V	V phase	
8	NC	Not connected	
9	HS_U	High-Side FET Predriver for U	
10	U	U phase	
11	NC	Not connected	
12	NC	Not connected	
13	NC	Not connected	
14	LS_W	Low-Side FET Predriver for W	
15	LSS_W	GND W phase	
16	LS_V	Low-Side FET Predriver for V	
17	LSS_V	GND V phase	
18	LS_U	Low-Side FET Predriver for U	
19	LSS_U	GND U phase	
20	IS_P	Current sense input (+)	
21	IS_N	Current sense input (-)	
22	GNDM	GND Motor	
23	LIN	LIN interface pin	
24	NC	Not connected	
25	NC	Not connected	
26	IO9	LVIO	
27	IO8	LVIO	
28	IO7	LVIO	1st test interface output TDO
29	IO6	LVIO	1st test interface input TDI
30	IO5	LVIO	

#	Pin name	Description	Comment
TQFP48, 7x7mm			
31	IO4	LVIO + HVS (high-voltage supply <50mA)	
32	IO3	LVIO + HVS (high-voltage supply <50mA)	
33	IO2	LVIO + HVIO (high-voltage in/out)	2nd Test interface output TDO
34	IO1	LVIO + HVIO (high-voltage in/out)	2nd Test interface input TDI
35	IO0	LVIO + HVIO (high-voltage in/out)	
36	NC	Not connected	
37	NC	Not connected	
38	IO11	LVIO	
39	IO10	LVIO	
40	VDDA	3.3V	3.3V IO supply, option for external sensors <25mA
41	NC	Not connected	
42	GND	GND IC	
43	NC	Not connected	
44	VS	Supply IC	For 48V : connect 12V external supply to VS
45	CPDRV	Driver for charge pump	
46	NC	Not connected	
47	VSM	Supply Motor	
48	VBOOST	Boost voltage to drive High-Side FETs	

Table 4 – Pin-out description for TQFP48

11.5. Marking Instruction

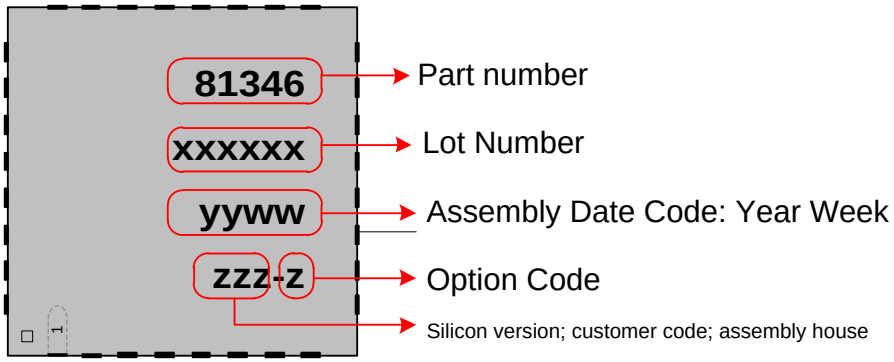


Figure 7 – Marking example on IC package QFN32 5x5

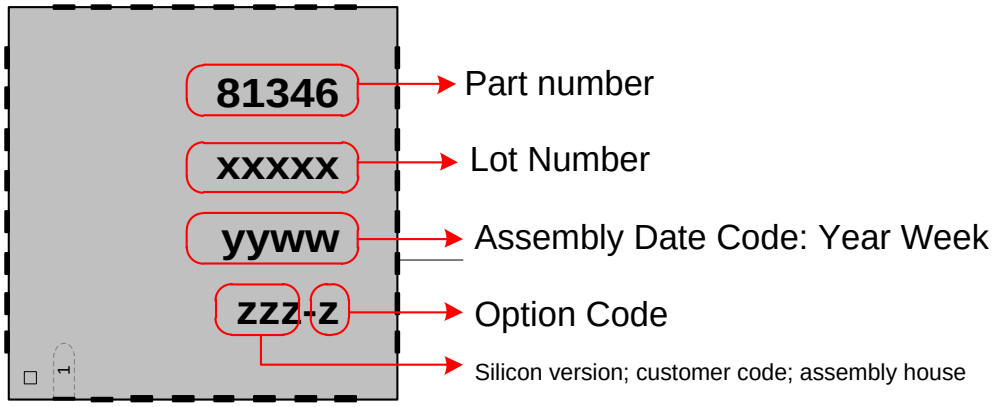


Figure 8 – Marking example on IC package TQFP48

12. Typical application schematic

In the following schematic examples, external components are indicated that may be needed to protect the IC against EMC/ESD pulses. Depending on ECU conditioned power, over-voltage and reverse polarity discretes may be needed. Capcacitor discretes or capacitor values will depend on specific OEM ESD/EMC requirements.

12.1. BLDC Application 12-24V

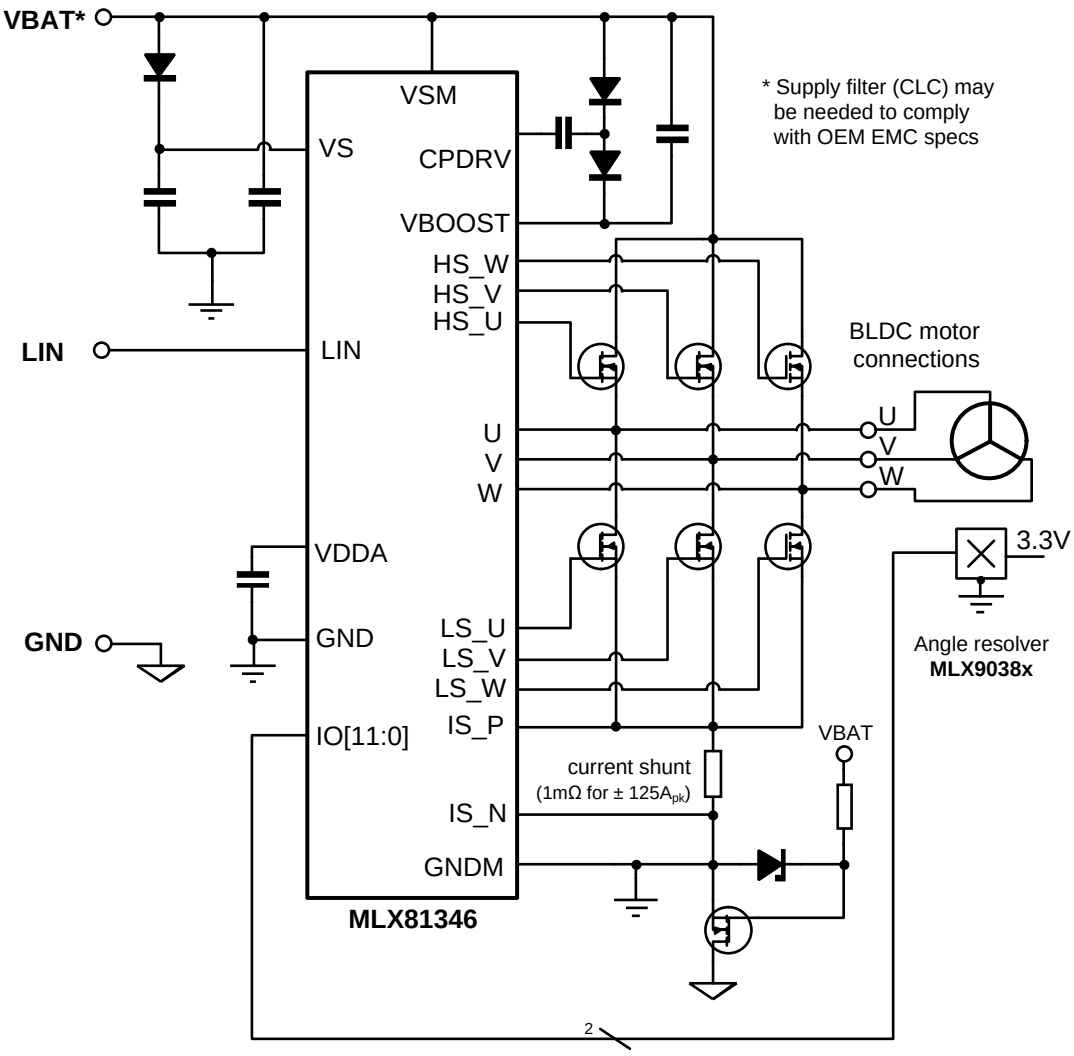


Figure 9 – Typical motor schematic with MLX81346

12.2. BLDC Applications 48V

For 48V automotive applications, 2 changes are needed versus the normal 12V application:

- VS pin is supplied by external 12V regulator to reduce IC heating (linear regulator or DC/DC)
- ECU – Peripheral communication is achieved via capacitive or optical coupling to enable isolation

The concept is shown in below figure:

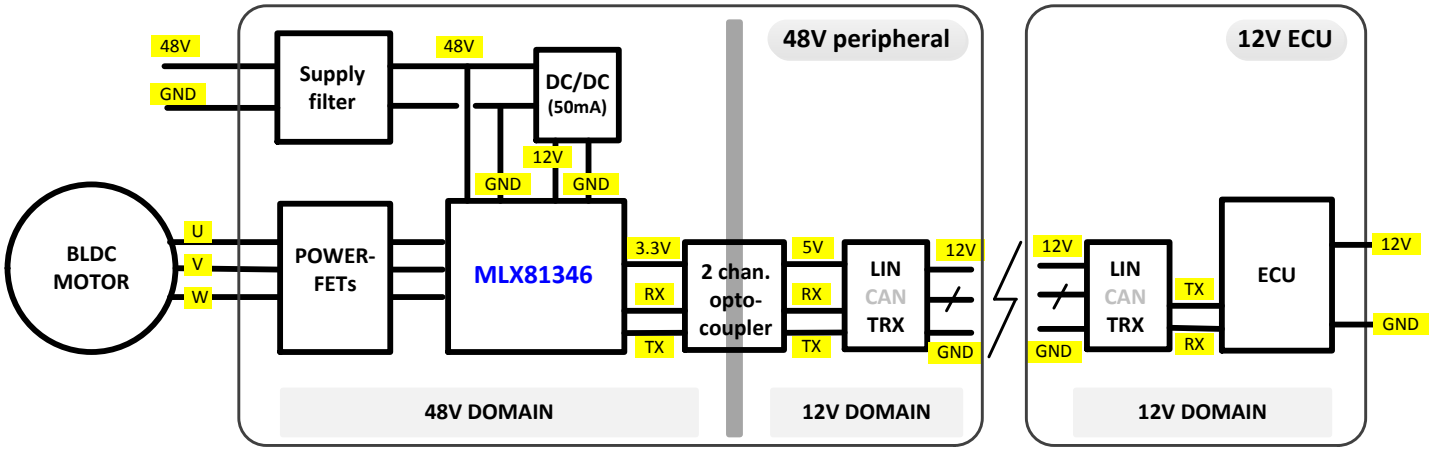


Figure 10 – Principle block diagram for MLX81346 in an isolated 48V application

The second application concept shows a version w/o isolation:

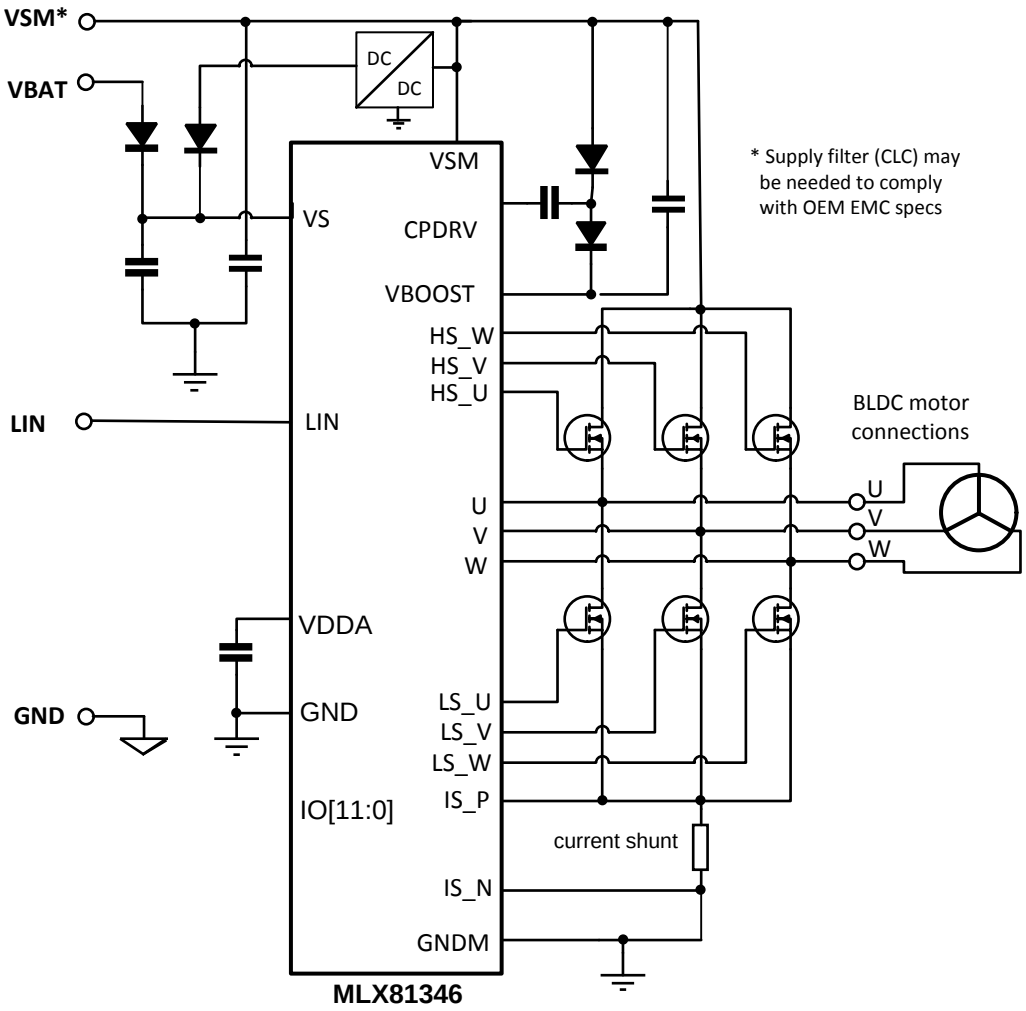


Figure 11 – Principle block diagram for MLX81346 in a 48V application

13. Electrical characteristics

13.1. Absolute maximum ratings

Parameter	Symbol	Min.	Max.	Unit	Condition
Supply voltage IC	VS	-0.5	32 36 [1]	V	
Supply voltage IC	VS	-0.5	45	V	t < 500ms
Supply voltage Motor	VSM	-0.5	60	V	
Supply voltage Motor	VSM	-0.5	70	V	t <200ms
Supply voltage transient	VS.tr1	-100		V	ISO 7637-2 pulse 1
Supply voltage transient	VS.tr2		75	V	ISO 7637-2 pulse 2
Supply voltage transient	VS.tr3	-150	100	V	ISO 7637-2 pulses 3a, 3b
Output voltage	VDDA	-0.3	3.6	V	
LIN bus voltage	VLIN	-40	40	V	
LIN bus voltage transient	VLIN.tr1	-30		V	ISO 7637-3 DCC slow -
LIN bus voltage transient	VLIN.tr2		30	V	ISO 7637-3 DCC slow +
LIN bus voltage transient	VLIN.tr3	-150	100	V	ISO 7637-2 pulses 3a, 3b
Analog HV voltage	VAN_HS_U(V,W)	-0.3 -3.0 [2]	VBOOST +0.3	V	HS_U, HS_V, HS_W
Analog HV voltage	VAN_U(V,W)	-0.3 -3.0 [2]	VSM+0.3 VSM+3.0 [2]	V	U, V, W
Analog HV voltage	VAN_LSx	-0.3	10	V	LS_U, LS_V, LS_W
VBOOST voltage	VAN_VBOOST		70 [4]	V	Switching transients at 60V motor drive
IS_P, IS_N voltage	VAN_ISx	-0.3	VDDA+0.3	V	
Analog HV voltage	VAN_HV	-0.3	VS+0.3	V	IO0, IO1, IO2 (HV input mode)
Analog LV voltage	VAN_LV	-0.3	VDDA+0.3	V	IO0..IO11
Digital input voltage	VIN_DIG	-0.3	VDDA+0.3	V	IO0..IO11
Digital output voltage	VOUT_DIG	-0.3	VDDA+0.3	V	IO0..IO11

[1] 36V operation is limited to maximum 24 hours over life; 28..36V motor driving may require a 100Ω resistor at VBOOST pin for protection reasons in case of PCB switching transients >45V.
[2] Design target for B version
[4] 80V for <200ms

ESD HBM capability	ESD_HBM	-2	2	kV	All pins
ESD HBM capability	ESD_HBM_LIN	-6	6	kV	Pin LIN. ESD applied on LIN pin versus shorted GND pins
ESD CDM capability	ESD_CDM	-500	500	V	All pins
Junction temperature	TJ	-55	175	ºC	

Table 5 – Absolute maximum ratings

13.2. Operating range

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Supply voltage	VS	4	12	32	V	Analog full performance
Supply voltage	VS	3.5		4	V	Analog reduced performance [5]
Supply voltage	VS	32		36	V	Analog reduced performance [6]
Supply voltage	VS	3.0		32 36 [6]	V	Digital functional
Supply voltage	VS	1.6 [2]		32 36 [6]	V	SRAM content valid
Supply voltage	VS	10		32 36 [6]	V	Pre-driver full performance
Supply voltage	VS	5.5		10	V	Pre-driver reduced performance [7]
Motor Supply voltage	VSM	10	12/24 /48	60	V	Operational motor driving
Motor Supply voltage	VSM	5.5		10	V	Operational motor driving reduced performance [7]
Junction temperature	T_J	-40		175	°C	Limited time at T_J=175 °C [8]

Table 6 – Operating range

[5] *IC is functional down to 3.5V with reduced analog performance. During IC start-up, VS needs to be >5.5V for a certain time to guarantee a correct reset.*
[6] *IC is functional up to 36V with reduced analog performance. 36V operation is limited to maximum 24 hours over life; 28..36V motor driving may require a 100Ω resistor at VBOOST pin to protect in case of PCB switching transients >45V.*
[7] *Pre-driver module is functional with reduced performance (lower gate drive voltage).*
[8] *Extended temperature range with Tj=175 ºC is only allowed for a limited time, customer's mission profile has to be agreed by Melexis as an obligatory part of the Part Submission Warrant (PSW).*

13.3. Electrical specifications

13.3.1. Current consumption

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Normal working current	INOM		10	15	mA	
Sleep mode current	ISLEEP		25 100 tbd tbd		µA	VS=13V; VSM=13V; $T_J \leq 125°C$ VS=13V; $T_J > 125°$ VS=18V VS>18V
Stop mode current	ISTOP		250	500	µA	
Holding mode current	IHOLD		7		mA	

Table 7 – Electrical specifications: current consumption

13.3.2. Supply system

13.3.2.1. VDDA 3.3V regulator (including 5V option)

Parameter	Symbol	Min	Typ	Max	Unit	Condition
3.3V analog supply voltage	VDDA	3.2	3.3	3.4	V	Bandgap and VDDA regulator trimmed
3.3V external current capability	IDDEXT_ VDDA	25			mA	VS ≥ 4V, external supply for sensors
3.3V under-voltage detection threshold	VTH_UV_ VDDA	2.75	2.85	2.95	V	VDDA ramping down
3.3V under-voltage detection hysteresis	VHYST_ UV_VDDA	0.1	0.175	0.25	V	Guaranteed by design
Under-voltage debouncing time	TUV_ VDDA	1.0	3.0	10	µs	
3.3V over-voltage detection threshold	VTH_OV_ VDDA	3.85	4.0	4.15	V	VDDA ramping up
3.3V over-voltage detection hysteresis	VHYST_ OV_VDDA	0.1	0.175	0.25	V	Guaranteed by design
5V analog supply voltage (option)	VDDA	4.85	5	5.15	V	Bandgap and VDDA regulator trimmed, SWITCH_VDDA_TO_5V=1
5V external current	IDDEXT_	25			mA	VS ≥ 5.5V, external supply for

Parameter	Symbol	Min	Typ	Max	Unit	Condition
capability	VDDA5V					sensors
5V under-voltage detection threshold	VTH_UV_ VDDA5V	4.05	4.2	4.35	V	VDDA ramping down, SWITCH_VDDA_TO_5V=1
5V under-voltage detection hysteresis	VHY_UV_ VDDA5V	0.1	0.175	0.25	V	
5V over-voltage detection threshold	VTH_OV_ VDDA5V	5.6	5.8	6.0	V	VDDA ramping up, SWITCH_VDDA_TO_5V=1
5V over-voltage detection hysteresis	VHY_UV_ VDDA5V	0.1	0.175	0.25	V	
Over-voltage debouncing time	TOV_ VDDA	1.0	3.0	10	µs	
Short detection threshold	ISH_LH_ VDDA	40.0	65.0	90.0	mA	
Short detection hysteresis	IHYST_SH _VDDA	1.0	1.5	2.0	mA	

Table 8 – Electrical specifications: VDDA regulator

13.3.2.2. VDDD 1.8V regulator

Parameter	Symbol	Min	Typ	Max	Unit	Condition
1.8V digital supply voltage	VDDD	1.80	1.875	1.95	V	Bandgap and VDDD regulator trimmed
1.8V current capability	IDDINT_ VDDD	15			mA	internal supply only, no external load

Table 9 – Electrical specifications: VDDD regulator

13.3.2.3. VSM under-voltage and VSM over-voltage detection

Parameter	Symbol	Min	Typ	Max	Unit	Condition
VSM under-voltage detection threshold	VUV_LH_ VS_0	3.5	4	4.5	V	Under-voltage detection on, PRUV_VS=0
VSM under-voltage detection threshold	VUV_LH_ VS_1	4.5	5	5.5	V	Under-voltage detection on, PRUV_VS=1
VSM under-voltage detection threshold	VUV_LH_ VS_2	5.5	6	6.5	V	Under-voltage detection on, PRUV_VS=2
VSM under-voltage detection threshold	VUV_LH_ VS_3	6.5	7	7.5	V	Under-voltage detection on, PRUV_VS=3

Parameter	Symbol	Min	Typ	Max	Unit	Condition
VSM under-voltage detection threshold	VUV_LH_VS_4	7.5	8	8.5	V	Under-voltage detection on, PRUV_VS=4
VSM under-voltage detection threshold	VUV_LH_VS_5	8.5	9	9.5	V	Under-voltage detection on, PRUV_VS=5
VSM under-voltage detection hysteresis	VHYST_UV_VS	0.1	0.5	1	V	
VSM under-voltage debouncing time	TUV_VS	1.0	3.0	10	µs	
VSM over-voltage detection threshold	VOV_LH_VS_0	20	22	24	V	Over-voltage detection on, PROV_VS=0
VSM over-voltage detection threshold	VOV_LH_VS_1	22	24	26	V	Over-voltage detection on, PROV_VS=1
VSM over-voltage detection threshold	VOV_LH_VS_2	38	40	42	V	Over-voltage detection on, PROV_VS=2
VSM over-voltage detection threshold	VOV_LH_VS_3	61	63	65	V	Over-voltage detection on, PROV_VS=3 [2]
VSM over-voltage detection hysteresis	VHY_OV_VS	1	2	3	V	
VSM over-voltage debouncing time	TOV_VS	1.0	3.0	10	µs	

Table 10 – Electrical specifications: VSM over- and under-voltage detection

13.3.2.4. Wake-up circuit

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Wake-up filter time IO pins	TWU_IO	15		80	µs	SLEEP mode , IO rising & falling edge
Wake-up filter time LIN pin	TWU_LIN	28		145	µs	Time for dominant level after SLEEP mode
Wake-up time internal timer	TWU_INT_0		0			WUI=00 (no wake-up)
Wake-up time internal timer	TWU_INT_1		$\frac{4096}{FRC\_10K}$			WUI=01 (~0.4s)
Wake-up time internal timer	TWU_INT_2		$\frac{8192}{FRC\_10K}$			WUI=10 (~0.8s)
Wake-up time internal timer	TWU_INT_3		$\frac{16384}{FRC\_10K}$			WUI=11 (~1.6s)

Table 11 – Electrical specifications: wake-up circuit

13.3.2.5. Bandgap

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Bandgap voltage	VBG	1.15	1.185	1.22	V	Trimmed
Bandgap voltage temperature coeff.	TC_VBG	0		180	ppm/K	

Table 12 – Electrical specifications: bandgap

13.3.3. Clock generation

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Frequency 1MHz oscillator	FRC_1M	-5%	1	+5%	MHz	
Frequency 32MHz oscillator	FRC_32M	-5%	32	+5%	MHz	MCU clock: MCU_CLK info : 32MHz results in ~25 MIPS
Frequency 10kHz oscillator	FRC_10K	5	10	20	kHz	
Timing accuracy	TIMING_ ACC	-1.5		1.5	%	Timing accuracy after sw correction using EEPROM calibration values

Table 13 – Electrical specifications: clock generation

13.3.4. PowerFET Pre-driver

13.3.4.1. Charge Pump and Boost voltage

The charge pump parameters are measured for Cfly = 680nF, Ctank = 3.3µF (see Figure 25).

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Charge pump clock	FRC_60K	50	60	70	KHz	
VBOOST Charge pump voltage	VBOOST_ VS_HI	7.5	9	10	V	VS ≥ 10V, ILOAD ≤ 15mA [9] MLX81346B
VBOOST Charge pump voltage	VBOOST_ VS_NOM	5.5	VS -1 [10]	9.5	V	8V < VS < 10V, ILOAD ≤ 15mA
VBOOST Charge pump voltage	VBOOST_ VS_LO	3	VS -1 [10]	8	V	5.5V < VS < 8V, ILOAD ≤ 5mA

[9] *In case external diodes are used, then BAV99 or BAT54S needs to be placed, both with Tj <= 150ºC*
[10] *with BAT54S and estimated 0.35V@ILOAD=3mA at 35°C*

Parameter	Symbol	Min	Typ	Max	Unit	Condition
VBOOST Output resistance	RBOOST_OUT		20	70	Ω	
VBOOST under-voltage level	VUV_HL_VBOOST	6.2	6.4	6.6	V	
VBOOST under-voltage hysteresis	VHYST_UV_VBOOST	0.2	0.25	0.35	V	
VBOOST Over-voltage level	VOV_LH_VBOOST	9.5	10.0	10.5	V	
VBOOST Over-voltage hysteresis	VHYST_OV_VBOOST	0.2	0.25	0.3	V	

Table 14 – Electrical specifications: charge pump

13.3.4.2. Predriver stage

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Predriver charge resistance	R_HIGH		8	16	Ω	VSM = 10V… 60V
Predriver discharge resistance	R_LOW		4	8	Ω	VSM = 10V… 60V
Predriver peak charge-current	I_ON		700		mA	Rgate=5Ω at FET gate
Predriver peak discharge current	I_OFF		1500		mA	Discharge diode at FET gate
FET turn-on time	T_ON		200		ns	1V⇨5.5V, 10Ω Rgate at FET (200nC)
FET turn-off time	T_OFF		150		ns	7V⇨1V, discharge diode at FET (200nC)
Interlock delay	T_EILD	0.03		60	µs	Dead-time programmable with 7-bit
FET gate drive voltage		7	9	10	V	VSM ≥ 10V [9]
	VPH	5 3	VSM-1.5 [10]	9.5 8		8V < VSM < 10V [9] 5.5V ≤ VSM ≤ 8V)[9]

Table 15 – Electrical specifications: predriver stage

13.3.4.3. Current sense amplifier

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Input range	V_CSL_IR	-125		125	mV	Current sense range for CSA_HIGHGAIN = 0
Input range	V_CSH_IR	-60		60	mV	Current sense range for CSA_HIGHGAIN = 1
Amplifier low gain	A_CSL	9.5	10	10.5		CSA_HIGHGAIN = 0
Amplifier high gain	A_CSH	19	20	21		CSA_HIGHGAIN = 1
Low-pass filter time		0.25	0.5	1.0	µs	guaranteed by design
Over-current detection level	VTH_OC	10		300	mV	programmable 8-bit DAC (1.56mV/LSB)
Over-current detection accuracy		-10		10	%	
Over-current debounce time	TDEB_OC	1		16	µs	programmable 7-bit timer

Table 16 – Electrical specifications: current sense amplifier

13.3.4.4. FET VDS monitor

Parameter	Symbol	Min	Typ	Max	Unit	Condition
VDS over-voltage level	VTH_OV_VDS_0	0.3	0.5	0.7		VDSMON_VTH_SEL = 00
	VTH_OV_VDS_1	0.8	1.0	1.2		VDSMON_VTH_SEL = 01
	VTH_OV_VDS_2	1.3	1.5	1.7		VDSMON_VTH_SEL = 10
	VTH_OV_VDS_3	1.8	2.0	2.2	V	VDSMON_VTH_SEL = 11
VDS over-voltage hysteresis		0.07		0.2	V	
VDS over-voltage debounce time	TDEB_OV_VDS	1		16	µs	programmable 7-bit timer

Table 17 – Electrical specifications: FET VDS monitor

13.3.4.5. OSD (off-state-diagnostics)

Parameter	Symbol	Min	Typ	Max	Unit	Condition
OSD Pull up current	I_OSD	150	200	250	µA	
OSD Pull down resistance	R_OSD	25	35	50	kΩ	

Table 18 – Electrical specifications: OSD

13.3.4.6. Active CDI **(current-direction-indicator)**

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Comparator threshold	VTH_CDI	-7.5	0	7.5	mV	[2]

Table 19 – Electrical specifications: Active CDI

13.3.4.7. VSM supply sensor

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Voltage range for ADC measurement				70	V	Measurement of VSM/52
VSM filter cut-off frequency				4	kHz	

Table 20 – Electrical specifications: VSM supply sensor

13.3.5. Over-temperature detection

Parameter	Symbol	Min	Typ	Max	Unit	Condition
OTD threshold	TTH_OT_LH	175	185	195	°C	Temperature ramping up
OTD threshold	TTH_OT_HL		160		°C	Temperature ramping down
OTD hysteresis	THY_OT		25		°C	

Table 21 – Electrical specifications: over-temperature detection

13.3.6. ADC

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Reference voltage	VREF_ADC		1.48		V	
Resolution			12		bit	ADC cyclic mode for differential input from -VREF_ADC to +VREF_ADC
Sample & Hold time				1	µs	
Conversion time	TCONV			1	µs	ADC_CLK= 16MHz
DNL			tbd		LSB	
INL			tbd		LSB	
ADC channel selection		0		29		see section 14.2.7.4

Table 22 – Electrical specifications: ADC

13.3.7. IOs

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Input level L⇨H	VTH_LH			2.4	V	IO[11:0]
Input level H⇨L	VTH_HL	1			V	IO[11:0]
Input hysteresis	VHY	0.1			V	IO[11:0]
LV output voltage L	VOL			0.4	V	IO[11:0] (LV-mode) ILOAD = 3mA
LV output voltage H	VOH	VDDA-0.4			V	IO[11:0] (LV-mode) ILOAD = 3mA
LV input range for ADC measurement		0		VDDA	V	IO[11:0] (LV-mode) Measurement of IO[11:0] / 2.4
HV output voltage L	VOL_HV			1.0	V	IO0, IO1, IO2 (HV-mode) ILOAD = 5mA
HV output voltage H	VOH_HV	VS-1.0			V	IO0, IO1, IO2 (HV-mode) ILOAD = 5mA
HV input range for ADC measurement		0		36	V	IO0, IO1, IO2 (HV-mode) Measurement of IOx/26
HS output current		50			mA	IO3, IO4 (HS-mode)
HS output resistance			10	20	Ω	IO3, IO4 (HS-mode)
I2C SCL/SDA debounce time	TDEB_I2C	50	70	100	ns	
I2C SDA hold time	TH_SDA	-50	0	50	ns	Referenced to SCL; SDAFILT_IO=00
I2C SDA hold time	TH_SDA	150	260	340	ns	Referenced to SCL; SDAFILT_IO=01
I2C SDA hold time	TH_SDA	220	320	420	ns	Referenced to SCL; SDAFILT_IO=10
I2C SDA hold time	TH_SDA	360	500	640	ns	Referenced to SCL; SDAFILT_IO=11

Table 23 – Electrical specifications: IO

13.3.7.1. **VDS monitor** IO3 and IO4

Parameter	Symbol	Min	Typ	Max	Unit	Condition
VDS monitor level	VTH _ VDS_IO	70	100	130	mA	
VDS monitor		5	10	20	mA	

Parameter	Symbol	Min	Typ	Max	Unit	Condition
hysteresis						
VDS debounce time	TDEB_OV_VDS_IO	0.032		16000	µs	programmable 4-bit timer with 4 bit prescaler

Table 24 – Electrical specifications: VDS monitor IO

13.3.8. LIN

13.3.8.1. LIN transceiver - static

The parameters are according to ISO17987-4, SAE J2602-1 and are valid for 7V ≤ VS ≤ 18V. [11]

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Transmitter internal capacitance[12]	CLIN		30	40	pF	Response on 14V pulse via 1K
Bus short circuit current	IBUS_LIM	40		200	mA	VLIN = VS = 18V, VTxD = 0V
Pull up resistance bus, untrimmed	RSLAVE	20	35	60	kΩ	VDISTERM = 0
Pull up current bus, sleep mode	ISLAVE_SLEEP	-50	-20	-5	µA	VLIN = 0V, VSBY = VAUX, VEN =0
Dominant input leakage current including pull up resistor	IBUS_PAS_dom	-1			mA	VLIN =0V, VS =12V, VTxD = VDDD, VDISTERM = 0 VEN = VDDD, VSBY = 0
Recessive input leakage current	IBUS_PAS_rec		0.25	20	µA	VEN = VDDD, VSBY = 0, VTxD = VDDD, VLIN > VS
Bus reverse current loss of battery [13]	IBUS_NO_BAT		0.25	23	µA	VS = 0V, 0V < VLIN ≤18V
Bus current during loss of ground [13]	IBUS_NO_GND	-100		1	µA	VS = VGND = 12V, 0 < VLIN ≤ 18V
Transmitter dominant output voltage [13]	VolBUS	0		0.2×VS	V	Rload = 500Ω
Transmitter recessive output voltage [13]	VohBUS	0.8×VS		1×VS	V	VEN = VDDD, VSBY = 0, VTxD = VDDD or sleep mode
Receiver dominant voltage	VBUSdom			0.4×VS	V	
Receiver recessive voltage	VBUSrec	0.6×VS			V	
Center point of	VBUS_CN	0.475×	0.5×	0.525×	V	VBUS_cnt = (Vth_dom+

[11] *For 5.5V < VS < 7V Reduced performance*

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
receiver threshold	T	VS	VS	VS		Vth_rec)/2
Receiver hysteresis	VHYS			0.175× VS	V	VHYS = (Vth_rec –Vth_dom)

The parameters are valid for 18V < VS ≤ 36V.

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Transmitter internal capacitance [12]	CLIN		30	40	pF	Response on 14V pulse via 1K
Bus short circuit current	IBUS_LIM	40		300	mA	VLIN = VS = 36V, VTxD = 0V
Pull up resistance bus, untrimmed	RSLAVE	20	35	60	kΩ	VDISTERM = 0
Pull up current bus, sleep mode	ISLAVE_SLEEP	-50	-20	-5	µA	VLIN = 0V, VSBY = VAUX, VEN =0
Dominant input leakage current including pull up resistor	IBUS_PAS_dom	-2			mA	VLIN =0V, VS =24V, VTxD = VDDD, VDISTERM = 0 VEN = VDDD, VSBY = 0
Recessive input leakage current	IBUS_PAS_rec		0.5	20	µA	VEN = VDDD, VSBY = 0, VTxD = VDDD, VLIN > VS
Bus reverse current loss of battery [13]	IBUS_NO_BAT		0.5	23	µA	VS = 0V, 0V < VLIN ≤ 36V
Bus current during loss of ground [13]	IBUS_NO_GND	-200		2	µA	VS = VGND = 12V, 0 < VLIN ≤ 36V
Transmitter dominant output voltage [13]	VolBUS	0		0.2×VS	V	Rload = 500Ω
Transmitter recessive output voltage [13]	VohBUS	0.8×VS		1×VS	V	VEN = VDDD, VSBY = 0, VTxD = VDDD or sleep mode
Receiver dominant voltage	VBUSdom			0.4×VS	V	
Receiver recessive voltage	VBUSrec	0.6×VS			V	
Center point of receiver threshold	VBUS_CNT	0.475×VS	0.5×VS	0.525×VS	V	VBUS_cnt = (Vth_dom+ Vth_rec)/2
Receiver hysteresis	VHYS			0.175×VS	V	VHYS = (Vth_rec –Vth_dom)

Table 25 – Electrical specifications: LIN transceiver – static

[12] *No production test, guaranteed by design and qualification*
[13] *In accordance to SAE J2602*

13.3.8.2. LIN transceiver – dynamic

The parameters are according to ISO17987-4, SAE J2602-1 and are valid for 7 ≤ VS ≤ 18V. [11]

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Propagation delay receiver [15] [16]	trx_pdf			6	µs	CRxD =25pF falling edge
Propagation delay receiver [15] [16]	trx_pdr			6	µs	CRxD =25pF rising edge
Propagation delay receiver symmetry	trx_sym	-2		2	µs	Calculate trx_pdf - trx_pdr
Receiver debounce time	trx_deb	0.5		4	µs	LIN rising & falling edge
LIN duty cycle 1 [16] [17]	D1	0.396				20kbps operation, normal mode
LIN duty cycle 2 [16] [17]	D2			0.581		20kbps operation, normal mode
LIN duty cycle 3 [16] [17]	D3	0.417				10.4kbs operation, low speed mode
LIN duty cycle 4 [16] [17]	D4			0.590		10.4kbs operation, low speed mode
tREC(MAX) – tDOM(MIN) [18]	Δt3			15.9	µs	10.4kbs operation, low speed mode
tDOM(MAX) – tREC(MIN) [18]	Δt4			17.28	µs	10.4kbs operation, low speed mode
Slew rate on pin LIN normal mode, trimmed			1.2		V/µs	dV/dt between duty cycle measurement points, VS=12V
Slew rate on pin LIN low speed mode, trimmed			0.6		V/µs	dV/dt between duty cycle measurement points, VS=12V
TxD dominant timeout [14]	ttxd_to		15		ms	Normal mode, vTxD=0V

[14] *Parameter in relation to internal signal TxD*

The following parameters are valid for 18V < VS ≤ 36V.

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Propagation delay receiver [15] [16]	trx_pdf			6	µs	CRxD =25pF falling edge
Propagation delay receiver [15] [16]	trx_pdr			6	µs	CRxD =25pF rising edge
Propagation delay receiver symmetry	trx_sym	-2		2	µs	Calculate trx_pdf - trx_pdr
Receiver debounce time	trx_deb	0.5		4	µs	LIN rising & falling edge
LIN duty cycle 1 [16] [17]	D1	0.330				20kbps operation, normal mode
LIN duty cycle 2 [16] [17]	D2			0.642		20kbps operation, normal mode
LIN duty cycle 3 [16] [17]	D3	0.386				10.4kbs operation, low speed mode
LIN duty cycle 4 [16] [17]	D4			0.591		10.4kbs operation, low speed mode
tREC(MAX) – tDOM(MIN) [18]	Δt3			21.89	µs	10.4kbs operation, low speed mode
tDOM(MAX) – tREC(MIN) [18]	Δt4			17.47	µs	10.4kbs operation, low speed mode
Slew rate on pin LIN normal mode, trimmed			2.4		V/µs	dV/dt between duty cycle measurement points, VS=24V
Slew rate on pin LIN low speed mode, trimmed			1.2		V/µs	dV/dt between duty cycle measurement points, VS=24V
TxD dominant timeout [19]	ttxd_to		15		ms	Normal mode, vTxD = 0V

Table 26 – Electrical specifications: LIN transceiver – dynamic

[15] *This parameter is tested by applying a square wave signal to the LIN. The minimum slew rate for the LIN rising and falling edges is 50V/µs*
[16] *See Figure 12*
[17] *Standard loads for duty cycle measurements are 1kΩ/1nF, 660Ω/6.8nF, 500Ω/10nF, internal master termination disabled*
[18] *In accordance to SAE J2602, see Figure 13*
[19] *Parameter in relation to internal signal TxD*

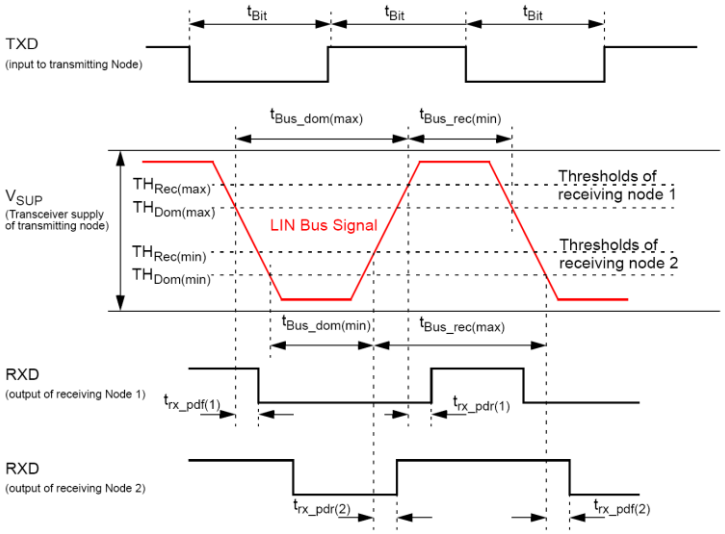


Figure 12 – LIN timing diagram (reference LIN2.1 specification)

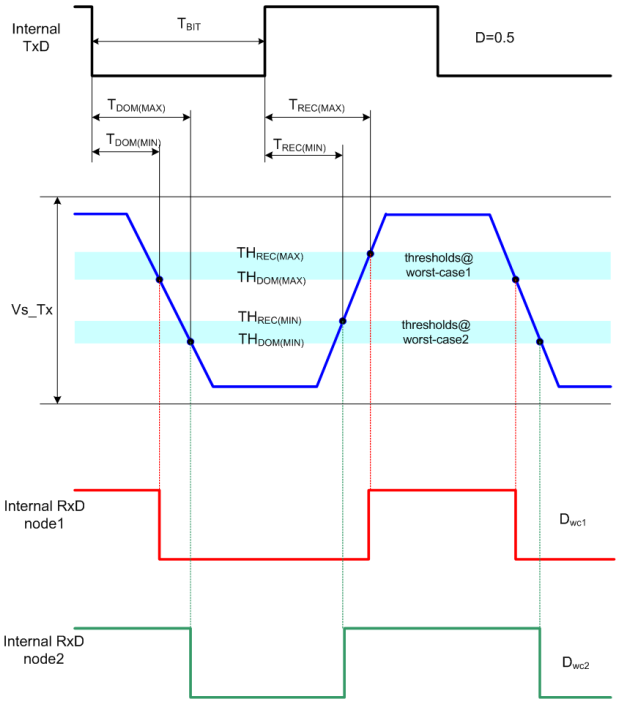


Figure 13 – LIN timing diagram, relation between propagation delay and duty cycle

(Reference SAE J2602 specification)

14. Functional description

14.1. System behaviour description

14.1.1. Supply system

Four different supply systems are internally used to supply different parts of the IC. The following section describes to usage of these supplies:

- Supply voltage VS (derived from battery voltage VBAT)
  - The car battery voltage VBAT is connected via the reverse polarity diode to the IC supply voltage VS. A decoupling capacitor of 100nF..220nF is required to ensure a stable operation of the IC
  - A larger capacitor must be used in parallel if user data have to be saved to the EEPROM memory in case of an under-voltage condition on VBAT. In this case, a proper voltage on VS has to be guaranteed for the EEPROM write procedure. Please refer to corresponding Application Notes for details about type and size of the capacitor

- Auxiliary supply VAUX
  - This internal voltage supplies those blocks of the MLX81346, which are used to control the main supply regulators VDDA and VDDD and all circuitry used for SLEEP mode and RESET feautures. VAUX is always active in case of sufficient voltage on VS is available. All blocks that are active during SLEEP mode are supplied by VAUX

- Analog Supply VDDA
  - This voltage supplies the analog part of the MLX81346 and is switched off during SLEEP mode.
  - Pin VDDA needs a decoupling capacitor in the range of 100nF..220nF to ensure a stable operation of the IC

- Digital Supply VDDD
  - This voltage supplies the digital part of the MLX81346. It is switched off during SLEEP mode. VDDD has no connection to an IC pin and is decoupled internally

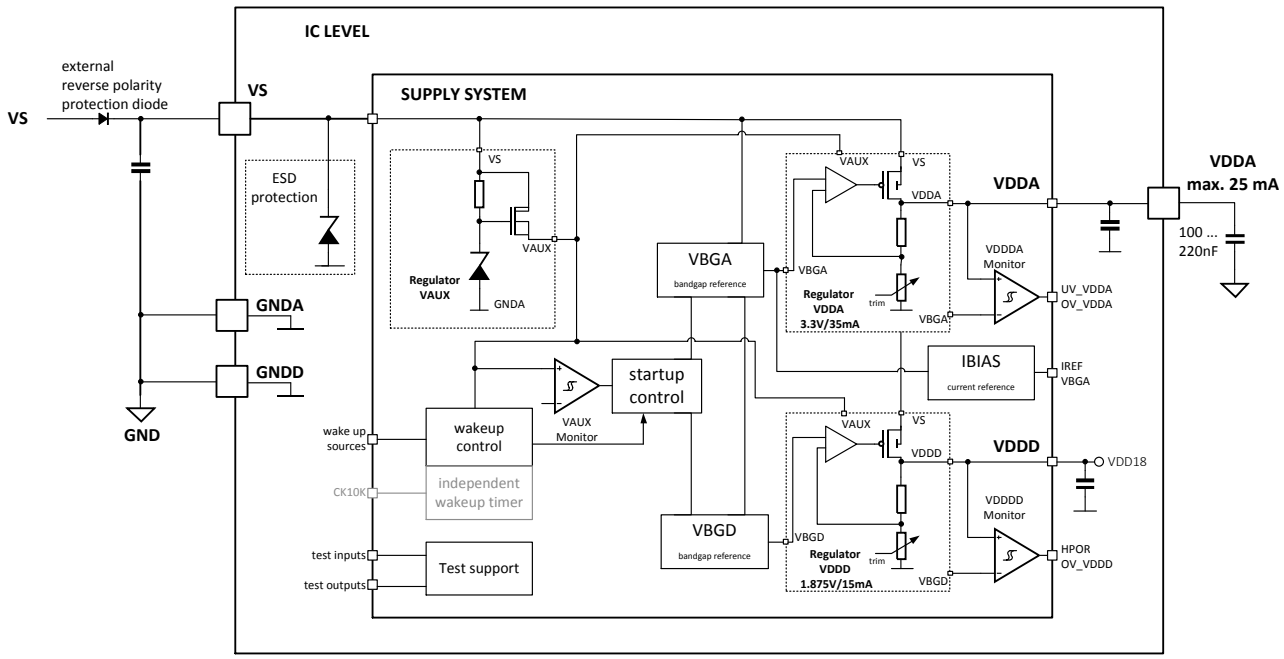


Figure 14 – Block diagram of supply system

14.1.2. Power On

The power up sequence is implemented in the following order (refer to figure below):

1. VAUX follows VS
2. Bandgaps are activated if VAUX reset threshold (VPOR_VAUX_LH) is reached
3. After bandgaps are stable, digital and analog regulators are activated at once (EN_REG)
4. VDDD > HPOR threshold (VHPOR_LH) ⇨ HPOR is released
5. VDDA > VDDA under voltage threshold (VTH_UV_VDDA) ⇨ UV VDDA is released

The whole sequence takes <200µs for a very fast slope at VS (1V/µs).

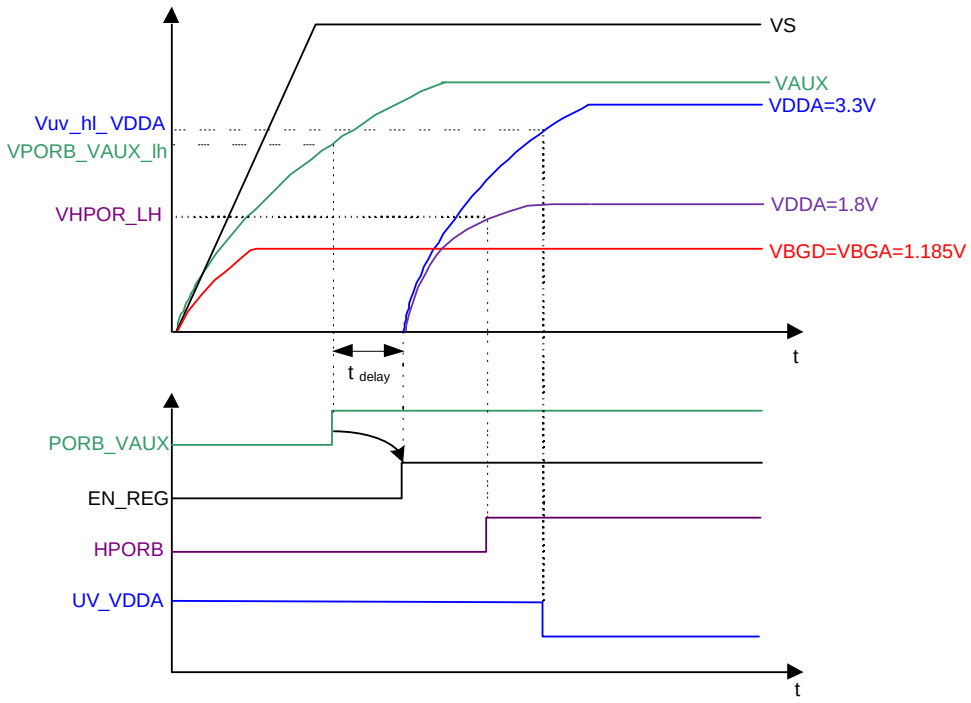


Figure 15 – Power on sequence

14.1.3. Power Down

When VS drops below its programmable under voltage detection level, the digital part gets noticed by an interrupt (UV_VS). In case the application needs to save data to the EEPROM, it is recommended to switch off power consuming parts immediately and start the EEPROM save procedure. The external buffer capacitance at VS shall be dimensioned in a way to provide enough energy for the save procedure.

When VDDA drops below the VDDA under-voltage threshold (VTH_UV_VDDA), the digital part will get the UV_VDDA flag which can trigger an interrupt. The analog part supplied by VDDA is disabled by an internal standby signal.

When VDDD drops below its reset level (HPORB activation threshold) the digital part will be reset.

When VS drops below VAUX reset activation threshold, all regulators and reference sources are switched off. The IC is powered down.

14.1.4. System Initialization and Trimming

For correct system start-up several registers in the IO part will be restored after every RESET or WAKE UP by writing corresponding trimming values taken from the non-volatile memory. This will be handled by the start-up code of the software platform provided by Melexis.

The values are stored during Melexis chip test. The defined addresses are write-protected and reserved for Melexis to avoid manipulation by the application software.

Start-up steps:

1. Trimming FLASH (VREF, VMG)
2. Trimming analog bandgap
3. Trimming digital bandgap
4. Trimming VDDA supply voltage
5. Trimming VDDD supply voltage
6. Trimming biasing current
7. Trimming 32MHz RC Oscillator
8. Trimming 1MHz RC Oscillator
9. Trimming EEPROM charge pump
10. Trimming charge pump oscillator frequency
11. Trimming charge pump voltage
12. Trimming LIN slew rate
13. Trimming LIN pull up resistor

14.1.5. Entering SLEEP mode (GTSM)

The SLEEP MODE of the controller is enabled by a write to a special port location. This mode is used to save power by switching off any not needed function, keeping only some minimum parts alive, which will watch events or conditions for possible WAKE UP requests.

Following sequence is recommended to enter sleep mode:

1. The application is stopped
2. All interrupts are disabled as an interrupt clears the HALT request of MLX16-FX
3. If used the ADC is switched off by setting the STOP bit of ADC_CTRL port. Otherwise the ADC can generate a DMA (direct memory access) that clears the HALT request similarly to an interrupt
4. All other DMA channels (e.g. I^2C-Slave, UART, PPM, SPI) are switched off by their respective stop bit
5. The digital and clock independent watchdog are acknowledged as a watchdog reset or interrupt is resetting the GTSM procedure
6. Waiting until Flash / EEPROM is idle by reading FL_BUSY / EE_BUSY bit
7. If COLIN is enabled it is set to HALT
8. Activatation of desired wake up sources (LIN wake up is always active and cannot be disabled)
9. Setting MLX16-FX to HALT (this activates the internal signal GTSM and put the IC into sleep mode)

Note: The system shutdown cannot be interrupted by any wake-up mechanism as long as the process has not been completed!

14.1.6. WAKE UP from SLEEP mode

Before sending a SLEEP mode request, the application software has to ensure that at least one WAKE UP source is enabled. Otherwise a re-start from SLEEP MODE is only possible by LIN activity or POR via VS under voltage.
After WAKE UP the software starts in the same way as after power on reset (POR).
The source of the WAKE UP can be read from port PORT_MISC_IN as shown below.

There are 3 different sources for a WAKE UP:

1. After a falling edge on the LIN bus followed by a dominant voltage level for longer than the specified value (TWU_LIN) and a rising edge on pin LIN, a WAKE UP signal is generated. If SLEEP mode was initiated by a short on the LIN bus (failure), and the short was removed afterwards, the IC wakes up immediately by the next rising edge on pin LIN. The integrated debounce filter suppresses an accidential WAKE UP caused e.g. by EMC interferences.

2. Falling or rising edge on digital input IO0 (with debounce filter, parameter TWU_IO, see Table 11

3. The configurable internal WAKE UP timer. It is possible to configure 3 wake up times ranging from 0.41s to 1.64s. The WAKE UP timer will be configured by port PORT_MISC_OUT as shown below. A WAKE UP timer event will be signalized by the INTERNAL_WU bit (port PORT_MISC_IN).

Port: PORT_MISC_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		
Field name	**Bit**	**R/W**	**Description**
XOR_WU_IO	15	R	XOR'ed IO wake up signals before wake up detection logic
LOCAL_WU	14	R	Flag indicating an IO wake up
LIN_RXD	13	R	Digital input from LIN shell
PHYSTAT1	12	R	From LIN Physical layer
LIN_WU	11	R	Flag indicating LIN as wake up source
INTERNAL_WU	10	R	Flag indicating wake up from internal timer
RSTAT	9	R	Status of RSTAT flip flop in analog
STOP_MODE	8	R	Indicates state after digital power up. If 1: return from STOP_MODE, else from POR or SLEEP
AIN_SUP	[7:0]	R	Spare inputs for supply system

Table 27 – PORT_MISC_IN

Port: PORT_MISC_OUT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
SEL_TEMP	[15:12]	RW	Select the temperature diode input for the temp. sensor: 0000: default temperature sensor other: MLX internal usage
PRUV_VDDA	11	RW	Under-voltage programming for VDDA

Address	Reset		Access
PROV_VS	[10:9]	RW	Control for VS over-voltage monitor 00: 22V 01: 24V 10: 40V 11: off
PRUV_VS	[8:6]	RW	Under-voltage programming for VS; detection level (V) = (PRUV_VS+4) -> from 4 to 9V
SWITCH_VDDA_TO_5V	5	RW	Switch VDDA supply to 5V
WUI	[4:3]	RW	Define internal wake up time delay in periodes of the 10kHz RC oscillator: 00: off 01: 4096 × FRC_10K (~0.41s) 10: 8192 × FRC_10K (~0.82s) 11: 16384 × FRC_10K (~1.64s)
CLEAR_RSTAT	2	RW	Setting this bit will force the RSTAT flip flop in analog asynchronously to low
SET_RSTAT	1	RW	Setting this bit will force the RSTAT flip flop in analog asynchronously to high
CLEAR_STOP	0	RW	Signal to clear STOP_MODE flip flop in analog part, must be cleared before entering STOP mode

Table 28 – PORT_MISC_OUT

14.1.7. System behaviour in case of under-voltage conditions

Table 29 describes the MLX81346 system behaviour for under-voltage conditions at the different supply pins.

VS	VDDA	VDDD	System behaviour
VS > 4V	> VTH_UV_VDDA	> VHPOR_LH	Normal working range
3.5V ≤ VS < 4V	> VTH_UV_VDDA	> VHPOR_LH	VS under-voltage; digital part fully functional; current consumption on VDDA should be limited by software; analog parameters cannot be guaranteed
3.0V ≤ VS < 3.5V	≤ VTH_UV_VDDA	> VHPOR_LH	VS under-voltage; digital part fully functional; diagnostic signals should be masked (because they are not reliable); the analog part should be shutdown by software
1.6V ≤ VS < 3.0V	≤ VTH_UV_VDDA	< VHPOR_LH	Digital reset active; VAUX is still on; analog and digital parts are in reset state; RAM content remains valid
VS < 1.6V	x	x	Chip function cannot be guaranteed

Table 29 – System behaviour at different under-voltage conditions

14.1.8. Supply voltage VSM over/under-voltage detection

The over-voltage and under-voltage detection circuits are based on the same principle as shown in Figure 16. The motor supply voltage VSM is divided internally and compared to the internal bandgap voltage (VBG). In SLEEP mode the input voltage dividers and the comparators are disabled.

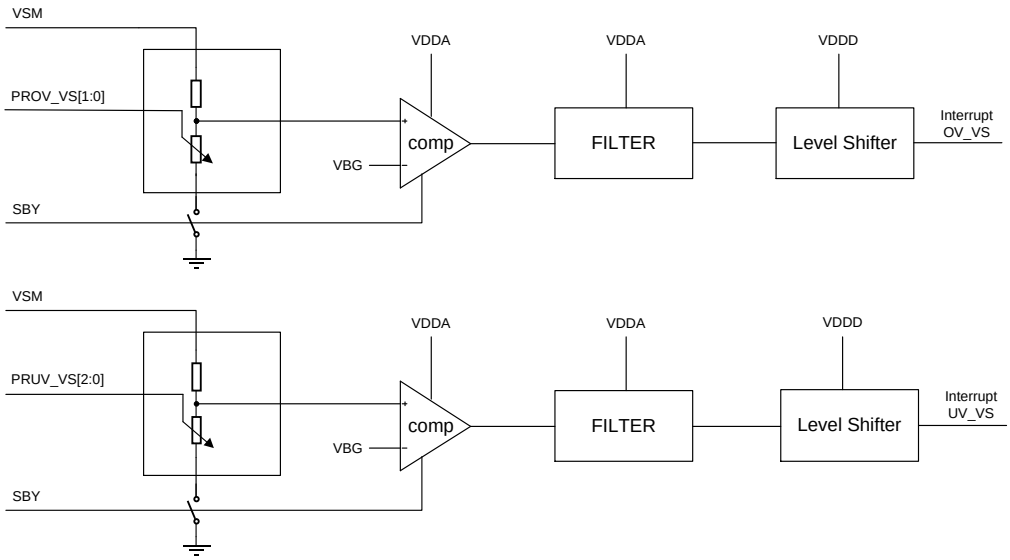


Figure 16 – Block diagram over- and under-voltage detection

The VSM over- and under-voltage detection thresholds can be programmed with PRUV_VS[2:0] from 4V to 9V and with PROV_VS[1:0] from 22V to 40V respectively (see Table 28).

14.1.9. Enter the STOP Mode (hybrid sleep mode)

The STOP mode targets to a very low current consumption while the RAM content remains valid. The microcontroller is halted, similar to the sleep mode. To keep the RAM content, the digital voltage supply is kept alive in a low current mode. All parts in the analog circuitry with high current consumption are switched off. STOP mode is achieved by setting bit SEL_STOP_MODE = 1 (see Table 30) following this procedure:

1. Disabling of all interrupts
2. Disabling of all periphery which can cause a DMA (e.g. ADC, UART, PPM, I^2C and SPI)
3. The digital and clock independent watchdog are acknowledged as a watchdog reset or interrupt is resetting the STOP procedure
4. Waiting until Flash / EEPROM is idle by reading FL_BUSY / EE_BUSY bit
5. Setting corresponding stop/disable bits to put analog functions in low power mode
6. Activatation of desired wake up sources (LIN wake up is always active and cannot be disabled)
7. Bit CLEAR_STOP must not be set as this prevents the STOP mode
8. Setting SEL_STOP_MODE = 1 which enables the STOP mode instead of the SLEEP mode (see Table 30)
9. Setting MLX16-FX to HALT (this activates the internal signal SET_STOP and put the IC into STOP mode)

The STOP mode can be left by any wake up source. The startup sequence is similar to the return from SLEEP mode. The STOP_MODE bit indicating the return from STOP mode will be set. This information can be used by the software.

Port: PORT_STOPMD_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		System, Word, Byte
Field name	**Bit**	**R/W**	**Description**
SEL_STOP_MODE	0	RW	CPU HALT will enter STOP mode, not SLEEP mode

Table 30 – STOP mode control port

It is possible to further reduce the current consumption of the IC by decreasing the system clock frequency. Please note that this is only possible if bit SEL_STOP_MODE is set to '1'.

Lowering the system clock frequency can be achived via the PRE_SBY_RCO1M and PRE_SBY_RCO32M bits in port STOPMD_CFG as shown below.

Following limitations have to be taken into account:

- LIN communication is not possible
- PWM frequency is reduced

- ADC sampling time is increased
- Simple and Complex Timer operate slower
- The number of operations between watchdog acknowledges must be reduced
- Bit SEL_STOP_MODE must be set to '1'

Port: STOPMD_CFG

Address	Reset		Access
0x0022C	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
PRE_SBY_RCO1M	4	RW	Set 1 MHz oscillator in standby mode
PRE_SBY_RCO32M	3	RW	Set 32 MHz oscillator in standby mode
PRE_SWITCHOFFOV_VS	2	RW	Disable over voltage detection for VS
PRE_SWITCHOFFUV_VS	1	RW	Disable under voltage detection for VS
PRE_SWITCHOFFUV_VDDA	0	RW	Disable under voltage detection of VDDA regulator to allow low voltage test

Table 31 – System behaviour at different under-voltage conditions

14.1.10. Enter the HALT Mode

The IC supports a third low power mode which does not have a direct impact on the supply system. For completeness an overview is given here.

HALT Mode is controlled by the CPU; no analogue supplies nor frequencies are disabled. Only the CPU is halted. It can be woken up by any interrupt and will continue program execution without delay. All periphery remains active (e.g. timers, PWM, ADC), a DMA is always possible. All register and RAM content is kept.

To enter the HALT mode following sequence is recommended:

1. Keeping only those interrupts active which re intended to finish the HALT mode, all others are disabled
2. SEL_STOP_MODE is set to '0' (otherwise the HALT bit triggers STOP mode)
3. DIS_GTSM bit is set to '1' - this prevents entering SLEEP mode after the CPU was halted
4. Setting MLX16-FX to HALT

14.2. Analog

14.2.1. Clock generation

The 81346 comprises four independed RC oscillators (see Figure 17).

1. The VDDD supplied trimmable 32MHz oscillator acts as the main clock source for the digital part (MCU_CLK). Calibration data is stored in the EEPROM to compensate temperature drifts. The oscillator can be switched to other frequencies (e.g. 28MHz, 24MHz) by means of the trim bits. The necessary trim values are stored in the EEPROM

2. The VDDD supplied, trimmable 1MHz oscillator is used to provide a (main) clock speed independent timing source

3. The VDDA supplied, trimmable 60kHz oscillator provides the charge pump clock

4. The VAUX (internal always-on analog supply) supplied, non-trimmable, 10kHz oscillator is used:

    - for wake up timing during sleep mode
    - as timer input
    - as time base for the (main) clock independent watch dog

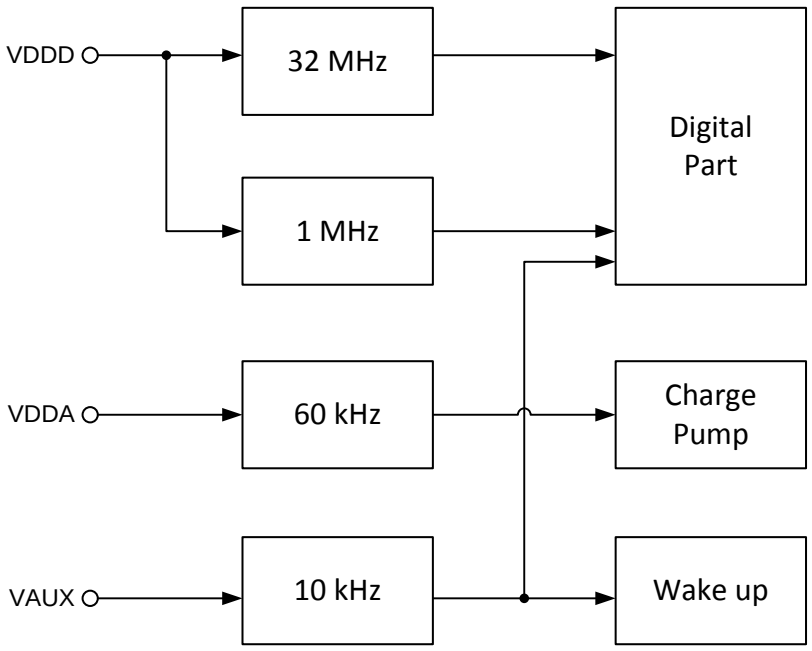


Figure 17 – Block diagram clock generation

Port: TRIM_RCO32M

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
LOCK	15	RW	Lock the port. Write is invalid when set
TR_UNUSED	[14:10]	RW	Free for spare use
TR_RCO32M_IN	[9:0]	RW	32 MHz RC oscillator trimming

Table 32 – TRIM_RCO32M port

Port: TRIM_RCO1M

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
LOCK	15	RW	Lock the port. Write is invalid when set
TR_UNUSED	14	RW	Free for spare use
PRE_TR_LIN_SLVTERM	[13:11]	RW	Slave termination trimming
PRE_TR_LIN_SLEWRATE	[10:8]	RW	Slew rate trimming
PRE_TR_RCO1M	[7:0]	RW	1 MHz RC oscillator trimming

Table 33 – TRIM_RCO1M port

14.2.1.1. Clock modulation

The purpose of the clock modulation block (PORT_SSCM_CONF & PORT_STEP_CONF) is the use of trim bits of the MCU oscillator to generate a modulation of the main system clock, which will distribute the spectral energy of the clock and its harmonics to a wider frequency range (see Figure 18). This helps to reduce single peaks in the EMC emission spectrum of the IC.

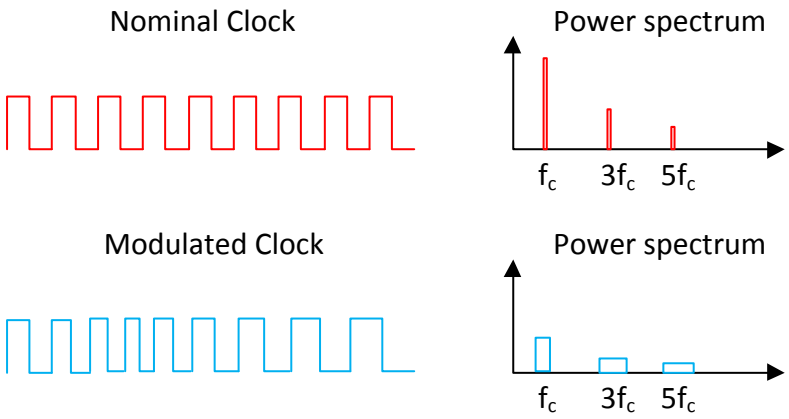


Figure 18: Impact of clock modulation in spectrum

The frequency is increased and decreased around a nominal value in linear sequence. All parameters of the modulation scheme are controllable; the step increment (STEP_INC), the duration per step (STEP_DUR) and the total step count (STEP_CNT).

STEP_DUR: the length of one step in main clock periods
STEP_INC: the increment/decrement to be added/subtracted from the recent trimming value TR_RCO32M_IN[9:0] per counting step
STEP_CNT: the number of steps needed for one period of an up/down modulation of the frequency

The modulation starts at the nominal trim value of TR_RCO32M_IN[9:0]. The block has an enable signal (SSCM_EN) that activates the modulated clock generator. If not enabled, the output trim value is provided without modification to the main clock MCU_CLK.

The signal SSCM_SINGLEBIT modifies the mode from an up/down modulation to a change of only 1 bit. The limits defined by the parameters STEP_DUR, STEP_INC and STEP_CNT shall be respected.

Port: PORT_SSCM_CONF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
SSCM_CENTERED	2	RW	0: Frequency of modulated clock shifted such that max frequency equals trimmed frequency 1: Frequency of modulated clock is centered around

			trimmed frequency
SSCM_SINGLEBIT	1	RW	Output shall provide a code with hamming distance of 1 instead of up/down modulated clock
SSCM_EN	0	RW	Enable the Spread-spectrum modulation

Table 34 – PORT_SSCM_CONF

Port: PORT_STEP_CONF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
STEP_CNT	[15:8]	RW	Step count per period of triangular modulation for Spread- spectrum: 00000000b: Spread-spectrum modulation off 00000001b: 1 step 11111111b: 255 steps
STEP_DUR	[7:4]	RW	Step duration in main clock pulses for Spread-spectrum: 0000b: Spread-spectrum modulation off 0001b: 1 step 1111b: 15 steps
STEP_INC	[3:0]	RW	Step increment for Spread-spectrum: 0000b: Spread-spectrum modulation off 0001b: 1 step 1111b: 15 steps

Table 35 – PORT_STEP_CONF

14.2.2. Motor driver module

14.2.2.1. Global description

The block diagram of the motor driver control block is shown in Figure 19.

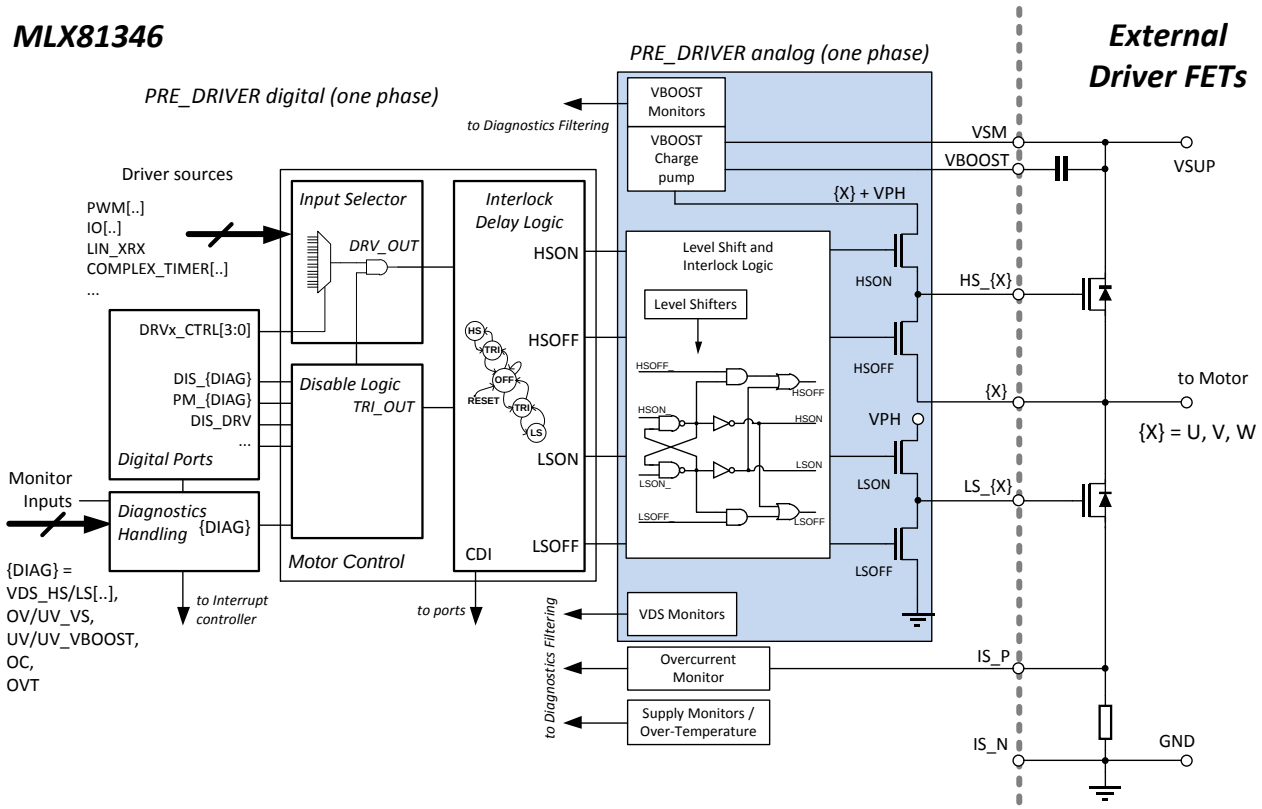


Figure 19 – Motor driver control block diagram

The three drivers U, V, W can be controlled independently by the digital. Three pairs of push-pull output drivers are implemented for the external high- and low-side driver NFETs. Each driver can be controlled by any of the PWM units, the IO[3:0] pins, the LIN pin or the complex timers. Furthermore they can be set individually to high, low or high-impedance state. The digital logic ensures a non-overlapping activation of the internal and external FETs with programmable internal (LSON vs. LSOFF and HSON vs. HSOFF) and external (HS_{X} vs. LS_{X}) interlock delays.

Several diagnostic functions are implemented which will disable the driver outputs:

- A GND based over-current comparator for the detection of over-current events of the motor (DRV_PROT_OC)
- Drain-source-voltage (VDS) monitors for the high- and low-side driver NFETs (DRV_PROT_OV_HS & DRV_PROT_OV_LS)

- Over- and under-voltage monitors of the supply voltages VDDA (incl. overcurrent), VS and VBOOST (DRV_PROT_UV_VDDA, DRV_PROT_UV_VS, DRV_PROT_OV_VS, DRV_PROT_OC_VDDA, DRV_PROT_OV_VDDA, DRV_PROT_UV_VBOOST, DRV_PROT_OV_VBOOST)
- Over-temperature protection (DRV_PROT_OVT)

14.2.2.2. Detailed description Motor driver control

The driver consists of three phases U, V, W with the corresponding indices 0, 1 and 2 respectivly. For generic descriptions below that are similar for all phases the {X} denotes any of the phases U, V, W, while the letter [x] denotes their indices [0, 1, 2]. VPH is the FET gate drive voltage (see Table 15).

The reset state of the driver is disabled, i.e. EN_DRV = 0 (see port PORT_DRV2_PROT, Table 46). In this state all HS_{X} are pulled to phase and all LS_{X} are pulled to GND.

To use the driver, the charge pump must be switched on (see section 14.2.3) and the driver block must be enabled by EN_DRV = 1. In this case each driver has one of three states:

1. State ‘0’: LS_{X} is switched to VPH and HS_{X} is switched to {X}
2. State ‘1’: LS_{X} is switched to GND and HS_{X} is switched to {X} + VPH
3. State ‘TRISTATE’: LS_{X} is switched to GND and HS_{X} is switched to {X}

In normal operation mode of the driver block, EN_DRV = 1 and DIS_DRV = 0. In this case the signal source for each driver is set by bits DRV[x]_CTRL[3:0] (see Table 36) in port PORT_DRV_CTRL (see Table 44). Each driver can be controlled by any of the PWM units, the IO[3:0] pins, the LIN pin or the complex timers. Furthermore they can be set individually to high, low or high-impedance state. The selected signal source can be overwritten by motor diagnostic signals (see section 14.2.2.3).

If the driver is enabled by EN_DRV = 1, setting bit DIS_DRV = 1 (port PORT_DRV2_PROT, Table 46) puts all drivers to state ‘TRISTATE’. This is the recommended setting in case the driver is switched off and the motor is still rotating. The driver itself shall remain enabled during that time (EN_DRV = 1)!

DRV[x]_CTRL[3:0]	DIS_DRV	DRV_OUT[x]	TRI_OUT[x]
0000	0	PWM_IN0 (PWM_MASTER1)	0
0001	0	PWM_IN1 (PWM_SLAVE1)	0
0010	0	PWM_IN2 (PWM_SLAVE2)	0
0011	0	PWM_IN3 (PWM_SLAVE3)	0
0100	0	PWM_IN4 (PWM_MASTER2)	0
0101	0	IO0	0
0110	0	IO1	0
0111	0	IO2	0
1000	0	IO3	0

1001	0	LIN_XRX	0
1010	0	COMPLEX_TIMER0_OUT	0
1011	0	COMPLEX_TIMER1_OUT	0
1100	0	(Reserved)	0
1101	0	0 (LOW)	0
1110	0	High-impedance (tristate)	1
1111	0	1 (HIGH)	0
XXXX	1	0	1

Table 36 – Driver control bits

14.2.2.3. Diagnostics handling

Diagnostics handling ensures that the software is informed about events (see Table 45 and Table 47) that can affect the safety of the driver FETs or the motor. Fast hardware switching can be enabled that switches the drivers into a predefined state.

The diagnostics handling operates similar for each of the following sources. It is described in generic terms where {DIAG} stands for one of the event sources {OV_LS_VDS, OV_HS_VDS, OC, UV_VDDA, UV_VS, OV_VS, OVT, OC_VDDA, OV_VDDA, UV_BOOST, OV_BOOST}.

Figure 20 shows the block diagram of the diagnostics handling. The diagnostic input signals are first synchronized and debounced before sent to the interrupt controller and captured in an IO port. The debounce filter time can be selected by {DIAG}_FILT_SEL (0: 1-2µs, 1: 100-110µs).

Diagnostic signals OV_HS_VDS[3:0] and OV_LS_VDS[3:0] are also masked as they are only valid when the corresponding high-side or low-side FET is turned on, section 14.2.2.3.1.

Several internal signals are shown in the block schematic:

- {DIAG}_IN: The diagnostic signal coming from analog
- {DIAG}_filtered: The diagnostic debounce filtered signal
- {DIAG}_masked: the diagnostic masked signals
- {DIAG}_IT: The diagnostic interrupt that captures {DIAG}_filtered or {DIAG}_masked signals (see Table 41 and Table 42)
- DIAG_IT: An extra interrupt that is set when one of the {DIAG}_OUT signals is ‘1’. This is when the motor controller takes over control of the outputs. i.e. puts the driver high/low or high-impedance
- {DIAG}_MEM: This is a read-only IO port that captures the {DIAG}_filtered or {DIAG}_masked signals. This port can only be set to ‘1’ by the {DIAG} signals (see Table 49 – PORT_DIAG_IN
- DIS_{DIAG}: is a read-write accessible IO port. When set to ‘1’ it resets the corresponding {DIAG}_MEM port to ‘0’ (see Table 45 and Table 47)

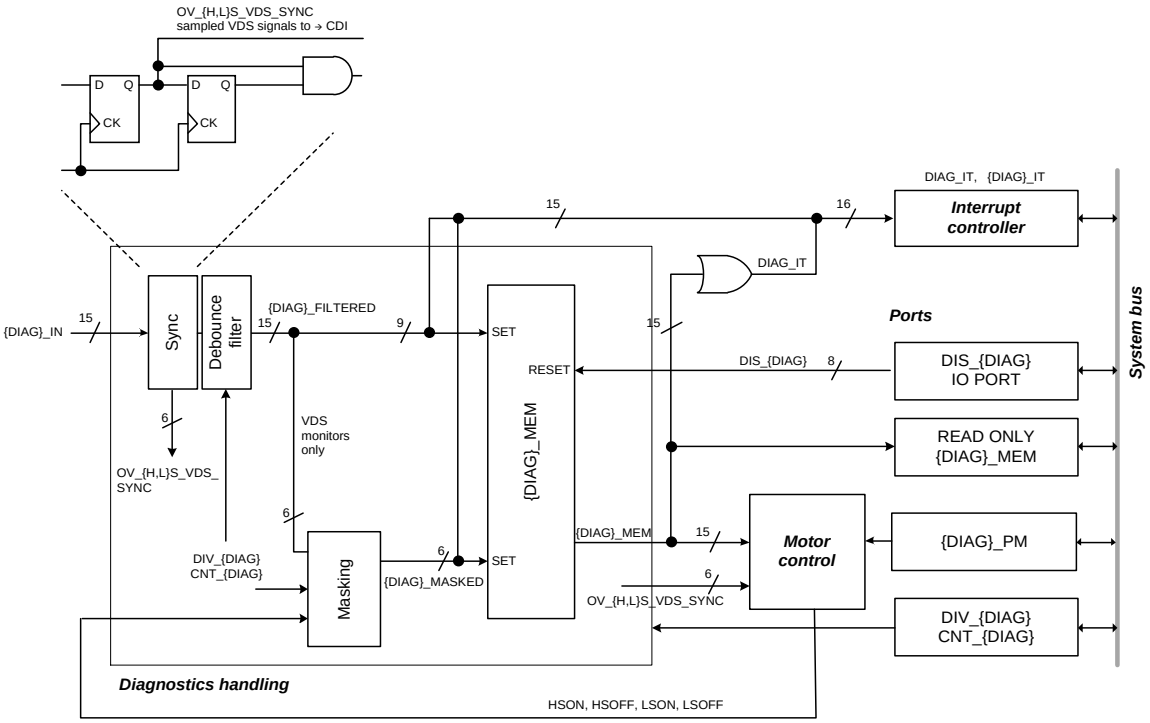


Figure 20 – Diagnostics handling block diagram

{DIAG}_FILTERED {DIAG}_MASKED	DIS_{DIAG}	{DIAG}_MEM	{DIAG}_IT	DIAG_IT
0	1	0	0	0
1	1	0	1	0
0	0	0	0	0
1	0	1	1	1

Table 37 – {DIAG}_MEM and IT logic

A hardware shutdown caused by {DIAG}_MEM should never be combined with a software {DIAG}_IT. These mechanisms must not be enabled in parallel. The drivers are protected against over-current, over-temperature, VS and VDS over-voltage conditions. Also in case of VS or VDDA under-voltage conditions, the drivers can be set in a safe state. For each {DIAG} signal, a software or hardware protection can be defined.

- Software control: uses the specific {DIAG}_IT signal and disables the hardware protection with signal DIS_{DIAG}= '1'.
- Hardware control: disables the specific {DIAG}_IT and enables the hardware protection with signal DIS_{DIAG}= '0'. In case of a hardware protection the DIAG_IT will be set.

The table below shows the use of the specific DIS_{DIAG} and {DIAG}_PM selection bit:

{DIAG}_MEM	DIS_{DIAG}	{DIAG}_PM	DIS_DRV	DRV_OUT[x]	TRI_OUT[x]	DIAG_IT
0	x	x	0	DRV[x]_CTRL[3:0] [20]	DRV[x]_CTRL[3:0] [20]	0
1	0	0	0	0/1	0	1
1	0	1	0	DRV[x]_CTRL[3:0] [20]	1	1
1	1	x	0	DRV[x]_CTRL[3:0] [20]	DRV[x]_CTRL[3:0] [20]	0
x	x	x	1	DRV[x]_CTRL[3:0] [20]	1	0

Table 38 – Driver control with diagnostics

In most cases the DRV_OUT signal is driven low in case of a diagnostic shut down. Only when an over voltage VDS appears on the low side driver, we force the High side on.

Source of diagnostic {DIAG}_MEM	Corresponding Enable bit DIS_{DIAG)	Corresponding Protection Mode bit {DIAG}_PM	DRV_OUT, in case of diagnostic shutdown
OVT_MEM	DIS_OVT	OVT_PM	0
OV_VS_MEM	DIS_OV_VS	OV_VS_PM	0
UV_VS_MEM	DIS_UV_VS	UV_VS_PM	0
UV_VDDA_MEM	DIS_UV_VDDA	UV_VDDA_PM	0
OC_MEM	DIS_OC	OC_ PM	0 Error! Bookmark not defined.
OV_HS_VDS_MEM[3:0]	DIS_OV_HS_VDS	OV_HS_VDS_PM	0
OV_LS_VDS_MEM[3:0]	DIS_OV_LS_VDS	OV_LS_VDS_PM	1

Table 39 – Driver control with diagnostics

Note that in case of a diagnostic shut down of the driver, ALL three FET's will go to the same state (low, high or high impedance).

[20] *no change means it is driven by DRV[x]_CTRL[3:0] signal*

14.2.2.3.1. VDS monitor masking principle

The masking is explained in following picture:

- The High side VDS detector mask is a delay time which starts after switching on the high side FET (DRV_OUT ⇨ '1' + T_{MASK}). After the delay time has elapsed, the mask becomes inactive. It becomes active again when the the high side FET is switched off (DRV_OUT ⇨ '0') even if the delay time has not elapsed yet
- The Low side VDS detector mask is a delay time which starts after switching on the low side FET (DRV_OUT ⇨ '0' + T_{MASK}). After the delay time has elapsed, the mask becomes inactive. It becomes active again when the the high side FET is switched off (DRV_OUT ⇨ '1') even if the delay time has not elapsed yet
- Both VDS detector signals are disregarded when the bridge is switched to tristate i.e. when TRI_OUT[x] = '1'
- When TRI_OUT[x] is switched back to '0' the delay time T_{MASK} must have elapsed before interpreting the VDS detectors

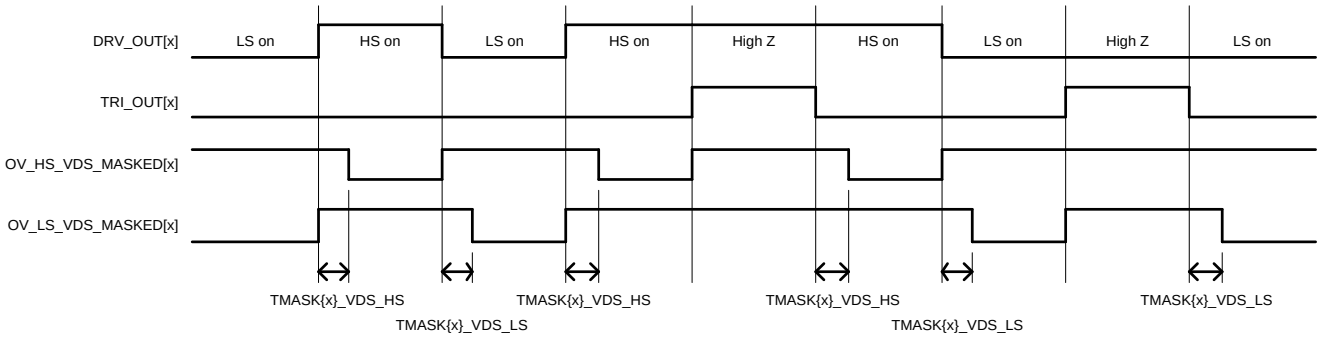


Figure 21 – Masking of HS and LS VDS over voltage diagnostic

14.2.2.4. Off state Diagnostics (OSD)

The operation of the off-state diagnostics is implemented in software.

When off-state diagnostics (OSD) is enabled, a current sources is connected to phase U, and two resistors are connected between phase V and GND and between phase W and GND respectively (see Figure 22)

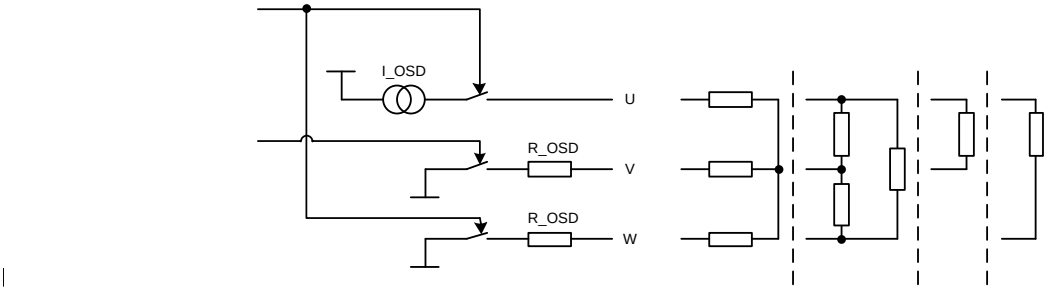


Figure 22 – Off state Diagnostics

In a typical software implementation, the phase voltages V and W are measured by the ADC and compared to a high and low threshold (VTHH and VTHL – optimal threshold value to be determined based on IC verification and qualification results). The comparison results can be used to generate three error flags:

- V_TOO_HIGH = 1 if V > VTHH = 2.5 x I_OSD x R_OSD
- V_TOO_LOW = 1 if V < VTHL = 0.25 x I_OSD x R_OSD
- W_TOO_LOW = 1 if W < VTHL = 0.25 x I_OSD x R_OSD

These error flags allow detecting failures for all phases: open-circuit (OC), short-circuit to ground (SCG), short-circuit to battery (SCB). If all error flags are 0, then no failure is detected. If an error flag is 1, then a failure is detected (see Table 40):

Failure	V_TOO_HIGH	V_TOO_LOW	W_TOO_LOW
No failure	0	0	0
OC phase U	0	1	1
OC phase V	0	1	0
OC phase W	0	0	1
SCG phase U	0	1	1
SCG phase V	0	1	1
SCG phase W	0	1	1
SCB phase U	1	0	0
SCB phase V	1	0	0
SCB phase W	1	0	0

Table 40 – OSD Error flags

14.2.2.5. Current Direction Indicator

Active CDI detection allows to detect the current direction in the low-side power FET's while they are switched on.
A clocked comparator and a phase multiplexer in the analog part detect the VDS of the selected low-side power FET to indicate a zero crossing of this voltage, which also indicates a change in current direction (from positive to negative or vice versa).

The Figure 23 gives an overview of the configuration of the ACTIVE CDI comparator logic.
ACTIVE CDI - current direction indicator when motor phase is actively driven helps to determine the exact commutation point when the current of the previous phase (coil) had been consumed completely.

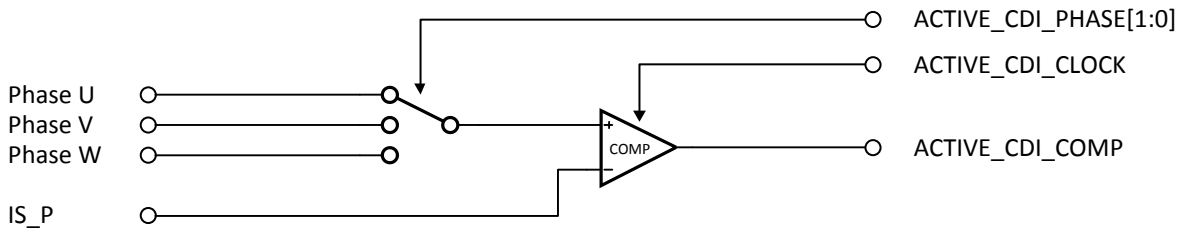


Figure 23 – ACTIVE CDI monitor - current direction indication when phase is driven

The Figure 24 explains an example for CDI usage, incl. settings in the – PORT_ACTIVE_CDI and Table 60 – PORT_ACTIVE_CDI_IN and the result in – PORT_ACTIVE_CDI_IN:

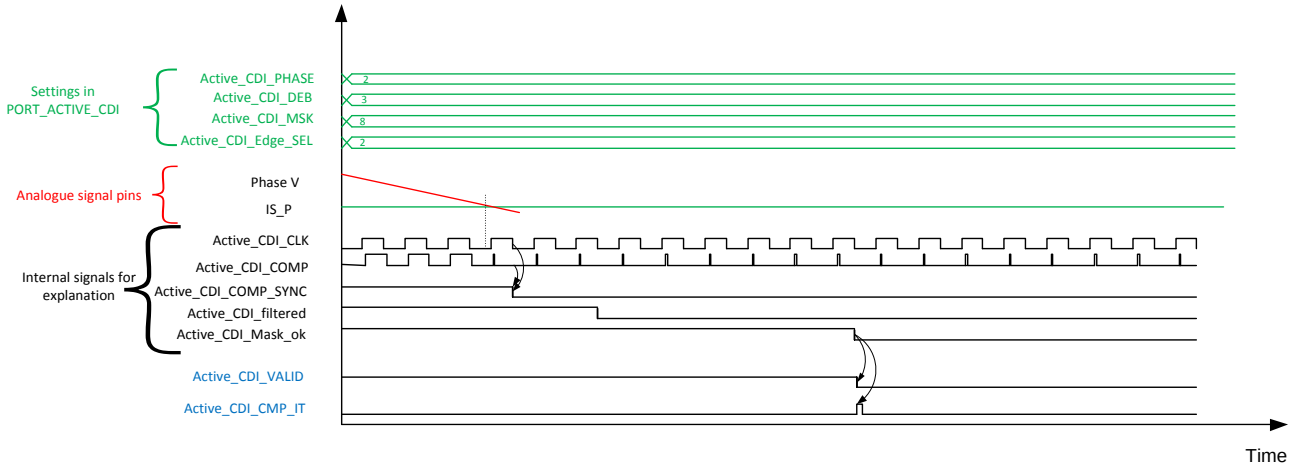


Figure 24 – ACTIVE CDI monitor – detailed signal generation

14.2.2.6. Ports map of motor driver module

Port: PORT_SUPP_IN

Address	Reset		Access
See Table 106 – MLX81346	0x00		Word, Read Only

ports overview			
Field name	**Bit**	**R/W**	**Description**
OV_BOOST_SYNC	15	R	Over-voltage VBOOST (unfiltered)
OV_BOOST_IT	14	R	Over-voltage VBOOST (filtered)
UV_BOOST_SYNC	13	R	Under-voltage VBOOST (unfiltered)
UV_BOOST_IT	12	R	Under-voltage VBOOST (filtered)
OVC_SYNC	11	R	Motor over-current (current sensor, unfiltered)
OVC_IT	10	R	Motor over-current (current sensor, filtered)
OVT_SYNC	9	R	Over-temperature (unfiltered)
OVT_IT	8	R	Over-temperature (filtered)
OV_VS_SYNC	7	R	Over-voltage VS (unfiltered)
OV_VS_IT	6	R	Over-voltage VS (filtered)
UV_VS_SYNC	5	R	Under-voltage VS (unfiltered)
UV_VS_IT	4	R	Under-voltage VS (filtered)
OV_VDDA_SYNC	3	R	Over-voltage VDDA (unfiltered)
OV_VDDA_IT	2	R	Over-voltage VDDA (filtered)
UV_VDDA_SYNC	1	R	Under-voltage VDDA (unfiltered)
UV_VDDA_IT	0	R	Under-voltage VDDA (filtered)

Table 41 – PORT_SUPP_IN

Port: PORT_SUPP2_IN

Address	**Reset**		**Access**
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
OV_HS_IO4_SYNC	5	R	VDS monitor IO4 (unfiltered)
OV_HS_IO4_IT	4	R	VDS monitor IO4 (filtered)
OV_HS_IO3_SYNC	3	R	VDS monitor IO3 (unfiltered)
OV_HS_IO3_IT	2	R	VDS monitor IO3 (filtered)
OC_VDDA_SYNC	1	R	Overcurrent VDDA (unfiltered)
OC_VDDA_IT	0	R	Overcurrent VDDA (filtered)

Table 42 – PORT_SUPP2_IN

Port: PORT_SUPP_CFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
OV_HS_IO4_FILT_SEL	9	RW	IO4 VDS monitor filter time — 0 : 1-2µs, 1 : 100-110us
OV_HS_IO3_FILT_SEL	8	RW	IO4 VDS monitor filter time — 0 : 1-2µs, 1 : 100-110us
OVT_FILT_SEL	7	RW	Over-temperature filter time — 0 : 1-2µs, 1 : 100-110us
OV_VS_FILT_SEL	6	RW	Over-voltage VS filter time — 0 : 1-2µs, 1 : 100-110us
UV_VS_FILT_SEL	5	RW	Under-voltage VS filter time — 0 : 1-2µs, 1 : 100-110us
OV_BOOST_FIL_SEL	4	RW	Over-voltage VBOOST filter time; 0 : 1-2µs, 1 : 100-110us
UV_BOOST_FILT_SEL	3	RW	Under-voltage VBOOST filter time — 0 : 1-2µs, 1 : 100-110us
OC_VDDA_FILT_SEL	2	RW	Over-current VDDA filter time — 0 : 1-2µs, 1 : 100-110us
OV_VDDA_FILT_SEL	1	RW	Over-voltage VDDA filter time — 0 : 1-2µs, 1 : 100-110us
UV_VDDA_FILT_SEL	0	RW	Under-voltage VDDA filter time — 0 : 1-2µs, 1 : 100-110us

Table 43 – PORT_SUPP_CFG

Port: PORT_DRV_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DRV2_CTRL	[11:8]	RW	Control of signal source for phase W (see Table 36 for a description of the bits)
DRV1_CTRL	[7:4]	RW	Control of signal source for phase V (see Table 36 for a description of the bits)
DRV0_CTRL	[3:0]	RW	Control of signal source for phase U (see Table 36 for a description of the bits)

Table 44 – PORT_DRV_CTRL

Port: PORT_DRV1_PROT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**

DIS_OV_LS_VDS	13	RW	Disable low-side FET VDS over-voltage HW protection
OV_LS_VDS_PM	12	RW	Low-side FET VDS over-voltage HW protection mode — 0: high side on, 1: tristate
DIS_OV_HS_VDS	11	RW	Disable high-side FET VDS over-voltage HW protection
OV_HS_VDS_PM	10	RW	High-side FET VDS over-voltage HW protection mode — 0: low side on, 1: tristate
DIS_OC	9	RW	Disable over-current HW protection
OC_PM	8	RW	Over-current HW protection mode — 0: low side on, 1: tristate
DIS_UV_VDDA	7	RW	Disable VDDA under-voltage HW protection
UV_VDDA_PM	6	RW	VDDA under-voltage HW protection mode — 0: low side on, 1: tristate
DIS_UV_VS	5	RW	Disable supply under-voltage HW protection
UV_VS_PM	4	RW	Supply under-voltage HW protection mode — 0: low side on, 1: tristate
DIS_OV_VS	3	RW	Disable supply over-voltage HW protection
OV_VS_PM	2	RW	Supply over-voltage HW protection mode — 0: low side on, 1: tristate
DIS_OVT	1	RW	Disable over-temperature HW protection
OVT_PM	0	RW	Over-temperature HW protection mode — 0: low side on, 1: tristate

Table 45 – PORT_DRV1_PROT

Port: PORT_DRV2_PROT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
EN_DRV	1	RW	Enable analog FET driver block
DIS_DRV	0	RW	Set all drivers to tristate

Table 46 – PORT_DRV2_PROT

Port: PORT_DRV3_PROT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DIS_OC_VDDA	7	RW	Disable VDDA over-current HW protection

OC_VDDA_PM	6	RW	VDDA over-current HW protection mode: 0: high side on 1: tristate
DIS_OV_VDDA	5	RW	Disable VDDA over-voltage HW protection
OV_VDDA_PM	4	RW	VDDA over-voltage HW protection mode 0: high side on 1: tristate
DIS_UV_BOOST	3	RW	Disable VBOOST under-voltage HW protection
UV_BOOST_PM	2	RW	VBOOST under-voltage HW protection mode 0: high side on 1: tristate
DIS_OV_BOOST	1	RW	Disable VBOOST over-voltage HW protection
OV_BOOST_PM	0	RW	VBOOST over-voltage HW protection mode 0: high side on 1: tristate

Table 47 – PORT_DRV3_PROT

Port: PORT_DRV_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
OV_LS_VDS2_IT	11	R	Low-side FET phase W VDS over-voltage detection interrupt, after debounce and masking
OV_LS_VDS1_IT	10	R	Low-side FET phase V VDS over-voltage detection interrupt, after debounce and masking
OV_LS_VDS0_IT	9	R	Low-side FET phase U VDS over-voltage detection interrupt, after debounce and masking
OV_LS_VDS_SYNC	[8:6]	R	Low-side FET VDS over-voltage detection, before digital debounce Bit 8: Low-side FET of phase W Bit 7: Low-side FET of phase V Bit 6: Low-side FET of phase U
OV_HS_VDS2_IT	5	R	High-side FET phase W VDS over-voltage detection interrupt, after debounce and masking
OV_HS_VDS1_IT	4	R	High-side FET phase V VDS over-voltage detection interrupt, after debounce and masking
OV_HS_VDS0_IT	3	R	High-side FET phase U VDS over-voltage detection interrupt, after debounce and masking

OV_HS_VDS_SYNC	[2:0]	R	High-side FET VDS over-voltage detection, before digital debounce Bit 2: High-side FET of phase W Bit 1: High-side FET of phase V Bit 0: High-side FET of phase U

Table 48 – PORT_DRV_IN

Port: PORT_DIAG_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
OV_LS_VDS_MEM	[14:12]	R	Low-side FET VDS over-voltage detection, cleared if DIS_OV_LS_VDS = 1
OV_HS_VDS_MEM	[11:9]	R	High-side FET VDS over-voltage detection, cleared if DIS_OV_HS_VDS = 1
OV_BOOST_MEM	8	R	VBOOST Over-voltage detection, cleared if DIS_OV_BOOST = 1
UV_BOOST_MEM	7	R	VBOOST Under-voltage detection, cleared if DIS_OV_BOOST = 1
OC_VDDA_MEM	6	R	VDDA over-current detection, cleared if DIS_OC_VDDA = 1
OVC_MEM	5	R	Over-current detection, cleared if DIS_OC = 1
OV_VDDA_MEM	4	R	VDDA over-voltage detection, cleared if DIS_OV_VDDA = 1
UV_VDDA_MEM	3	R	VDDA under-voltage detection, cleared if DIS_UV_VDDA = 1
UV_VS_MEM	2	R	Supply under-voltage detection, cleared if DIS_UV_VS = 1
OV_VS_MEM	1	R	Supply over-voltage detection, cleared if DIS_OV_VS = 1
OVT_MEM	0	R	Over-temperature detection, cleared if DIS_OVT = 1

Table 49 –PORT_DIAG_IN

Port: PORT_DIV_VDS_DEB

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DIV_VDS	[2:0]	RW	VDS debounce pre-divider, 0-7dec

Table 50 –PORT_DIV_VDS_DEB

Port: PORT_CNT_VDS_DEB

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CNT2_VDS	[11:8]	RW	VDS debounce for phase W, counter setting 0-15dec;

			VDS debounce time: TDEB_VDS = (CNT{x}_VDS + 1) * 2^{DIV_VDS} / MCU_CLK
CNT1_VDS	[7:4]	RW	VDS debounce for phase V
CNT0_VDS	[3:0]	RW	VDS debounce for phase U

Table 51 –PORT_CNT_VDS_DEB

Port: PORT_DIV_MASK_VDS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DIV_MASK_VDS	[3:0]	RW	VDS mask pre-driver, 0-7dec

Table 52 –PORT_DIV_MASK_VDS_LS

Port: PORT_CNT_MASK_VDS_LS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CNT_MASK2_VDS_LS	[11:8]	RW	VDS debounce for phase W; VDS mask time: TMASK{x}_VDS_LS = (CNT_MASK{x}_VDS_LS + 1) * $2^{DIV\_MASK\_VDS}$) / MCU_CLK
CNT_MASK1_VDS_LS	[7:4]	RW	VDS debounce for phase V
CNT_MASK0_VDS_LS	[3:0]	RW	VDS debounce for phase U

Table 53 –PORT_DIV_MASK_VDS_LS

Port: PORT_CNT_MASK_VDS_HS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CNT_MASK2_VDS_HS	[11:8]	RW	VDS debounce for phase W; VDS mask time: TMASK{x}_VDS_HS = (CNT_MASK{x}_VDS_HS + 1) * $2^{DIV\_MASK\_VDS}$) / MCU_CLK
CNT_MASK1_VDS_HS	[7:4]	RW	VDS debounce for phase V

CNT_MASK0_VDS_HS	[3:0]	RW	VDS debounce for phase U

Table 54 –PORT_DIV_MASK_VDS_LS

Port: PORT_DIV_EILD

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DIV_EILD	[2:0]	RW	External interlock delay pre-divider

Table 55 –PORT_DIV_EILD

Port: PORT_CNT_EILD_LSOFF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CNT_EILD2_LSOFF	[11:8]	RW	External LS interlock delay for phase W; EILD time: TEILD{x}_LSOFF = (CNT_EILD{x}_LSOFF + 1) * 2^{DIV_EILD}) / MCU_CLK
CNT_EILD1_LSOFF	[7:4]	RW	External LS interlock delay EILD for phase V
CNT_EILD0_LSOFF	[3:0]	RW	External LS interlock delay EILD for phase U

Table 56 –PORT_CNT_EILD_LSOFF

Port: PORT_CNT_EILD_HSOFF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CNT_EILD2_HSOFF	[11:8]	RW	External HS interlock delay for phase W; EILD time: TEILD{x}_HSOFF = (CNT_EILD{x}_HSOFF + 1) * 2^{DIV_EILD}) / MCU_CLK
CNT_EILD1_HSOFF	[7:4]	RW	External HS interlock delay EILD for phase V
CNT_EILD0_HSOFF	[3:0]	RW	External HS interlock delay EILD for phase U

Table 57 –PORT_CNT_EILD_HSOFF

Port: PORT_CDI_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read only
Field name	**Bit**	**R/W**	**Description**
CDI	[2:0]	RW	Current direction indicator bits: Bit 2: CDI of phase W Bit 1: CDI of phase V Bit 0: CDI of phase U

Table 58 –PORT_CDI_IN

Port: PORT_ACTIVE_CDI

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read only
Field name	**Bit**	**R/W**	**Description**
CDI_FET_SWITCH_ALL	15	RW	The phase selection can be overwritten by a bit CDI_FET_SWITCH_ALL. When this bit is set, every rising edge at any HSON[x]/LSON[x] control signal will restart the mask window
ACTIVE_CDI_CLK_PREDIV2	14	RW	clock predivider for Active CDI, Active CDI clock period=2*(2^ACTIVE_CDI_PREDIV2 * (ACTIVE_CDI_CLK_DIV+1) * MCU period
ACTIVE_CDI_EDGE_SEL	[13:12]	RW	edge selection for the ACTIVE CDI interrupt (00=rising, 01=falling,1x=both edges)
ACTIVE_CDI_MSK	[11:8]	RW	Active CDI masking time in periods of active CDI clock from 0 to 15
ACTIVE_CDI_DEB	[7:6]	RW	Active CDI debounce time in periods of active CDI clock (0x=0,01=1..2,10=2..3 periods)
ACTIVE_CDI_CLK_DIV	[5:2]	RW	clock divider for Active CDI, Active CDI clock period=2*(2^ACTIVE_CDI_PREDIV2 * (ACTIVE_CDI_CLK_DIV+1) * MCU period
ACTIVE_CDI_PHASE	[1:0]	RW	select the phase for the Active CDI measurement (00=disabled, 01 - U, 10 - V, 11 - W)

Table 59 – PORT_ACTIVE_CDI

Port: PORT_ACTIVE_CDI_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read only

Field name	Bit	R/W	Description
-	[15:2]	R W	Always read 0 Do nothing
ACTIVE_CDI_VALID	1	R	After filtering and masking
ACTIVE_CDI_COMP_SYNC	0	R	Before digital filtering

Table 60 – PORT_ACTIVE_CDI_IN

14.2.3. Charge Pump

A charge pump is required to drive the gates of the high side NFETs above VSM. The output CPDRV provides a clock for an external Dickson charge pump. The output of the charge pump should be connected to pin VBOOST for voltage regulation (the low level of the charge pump clock CPDRV is regulated) and for supplying the high side drivers.

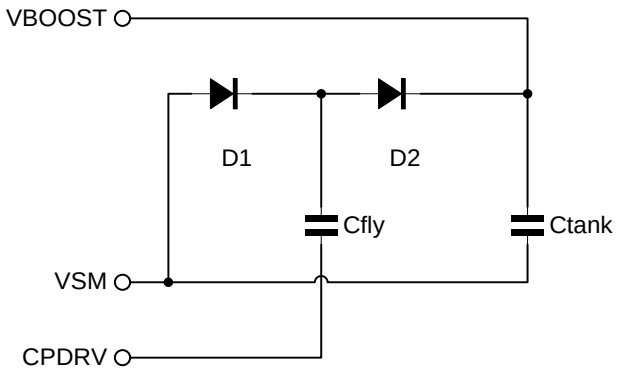


Figure 25 – Charge pump external circuits

Port: PORT_CP

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
CP_TRI	2	RW	Set output CPDRV to tristate
SET_CPDRV	1	RW	Sets the value of CPDRV if TR_RCO50K[3:0] = 0x7
EN_CP	0	RW	Enables the charge pump and the charge pump RCO

Table 61 – PORT_CP

Port: PORT_TRIM_CP

Address	Reset		Access
See	0x00		Word, Byte

Table 106 – MLX81346 ports overview			
Field name	**Bit**	**R/W**	**Description**
LOCK	15	RW	Lock the port. Write is invalid when set
TR_CP	[7:4]	RW	Charge pump voltage trimming
TR_RCO50K	[3:0]	RW	Charge pump RCO frequency trimming

Table 62 – PORT_TRIM_CP

14.2.4. Current sense amplifier (CSA) and Overcurrent detection

14.2.4.1. Description

An external current sense resistor can be connected between pins IS_P and IS_N. The sense resistor voltage is amplified with a selectable voltage gain of G=10 (Bit CSA_HIGHGAIN = 0) or G=20 (Bit CSA_HIGHGAIN = 1) and internally connected to the ADC (see Figure 26). The measured output voltage is used to assess the level of the current through the sense resistor (shunt):

$$I_{\_CS} = \frac{CS_{OUT\_P} - CS_{OUT\_N}}{G * R_{CS}}$$

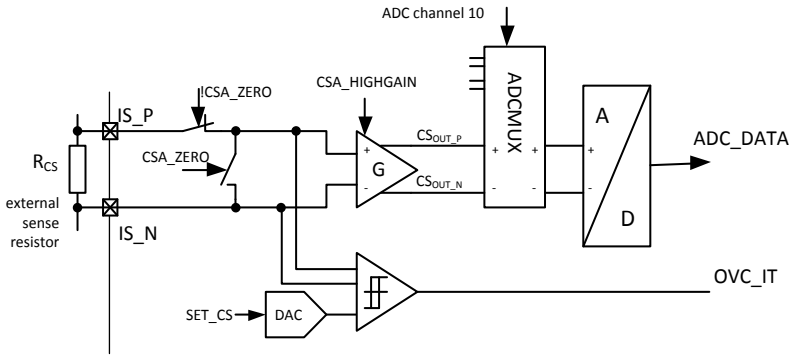


Figure 26 – CSA block diagram

The CSA parameters allow a shunt resistor measurement voltage range of ±125mV which matches the ADCs voltage range of VREF_ADC = ±1.48V (including margin for offsets and other imperfections).

The ADC channel connected to the CSA (ch 10) operates in differential mode.

The current sensor resolution for the ADC in 12bit cyclic mode is:

$$LSB_{ADC} = \frac{2 * VREF\_ADC}{G * 2^{12}} = \frac{2 * 1.48V}{10 * 2^{12}} = 72\mu V$$

The formula to calculate the digital ADC output for a given input voltage is:

$$ADC_{out_{DIG}} = \left(\left(CS_{OUT\_P} - CS_{OUT\_N} \right) * G - VREF\_ADC \right) * \frac{2^{12}}{2 * VREF\_ADC}$$

Examples:

An input voltage *IS_P - IS_N = +125mV* would result in an ideal ADC output value of:

$$\begin{aligned} ADC_{out_{DIG}} &= \left(\left(CS_{OUT\_P} - CS_{OUT\_N} \right) * G + VREF\_ADC \right) * \frac{2^{12}}{2 * VREF\_ADC} \\ &= (125mV * 10 + 1.48V) * \frac{4096}{2 * 1.48V} \\ &= 3778 \end{aligned}$$

The over-current detection is implemented with a comparator and an 8-bit DAC controlled by SET_CS (port PORT_CURR_SENS, see Table 63). The comparator inputs are directly connected to the current sense amplifier inputs IS_P and IS_N. Its output is connected to the diagnosis block of the motor driver module signal OVC_IT (port PORT_SUPP_IN, see Table 41), which is able to switch off the predrivers in case of an overcurrent event (see section 14.2.2.3 for details about the configuration). The diagnosis block is also connected to the interrupt controller and triggers the OVC interrupt, if enabled.

A digital debouncer helps to suppress high frequent distortions of the current sense signal. It is programmable and can be controlled by PORT_OC_DEB (see Table 64).

The voltage range of the DAC is adapted to suit all sense resistor values used for current sensing. See chapter 13.3.4.3 for details on parameter VTH_OC.

The OC DAC continues to work for settings above 300 mV, but is not guaranteed. The space above VTH_OC_max is reserved to ensure the specified range in the presence of offset and gain errors. The overcurrent DAC can be programmed only for positive thresholds.

Port: PORT_CURR_SENS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
EN_OC	11	RW	Enable Overcurrent comparator
CSA_HIGHGAIN	10	RW	Enable gain=20 of CSA (if Bit is 1), else gain =10
CSA_ZERO	9	RW	Offset calibration — 0: normal operation, 1: the inputs of the current sensor are internally shorted
EN_CSA	8	RW	Enable the current sensor
SET_CS	[7:0]	RW	Set the level of the current sensor

Table 63 – PORT_CURR_SENS

Port: PORT_OC_DEB

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DIV_OC	[6:4]	RW	OC debounce pre-divider, 0-7dec
CNT_OC	[3:0]	RW	OC debounce counter setting; OC debounce time: TDEB_OC = (CNT_OC * 2^{DIV_OC}) / MCU_CLK

Table 64 – PORT_OC_DEB

14.2.5. VSM supply sensor

The VSM supply sensor, shown in Figure 27, divides the VSM voltage by 26 or 52, followed by a filter and a buffer which drives ADC channel 8 (see Table 68). The filter can be enabled by setting VSM_FILT_ON = 1 (port PORT_MISC2_OUT, see Table 65). The filter and the buffer are enabled by VSM_FILT_ON = 1.

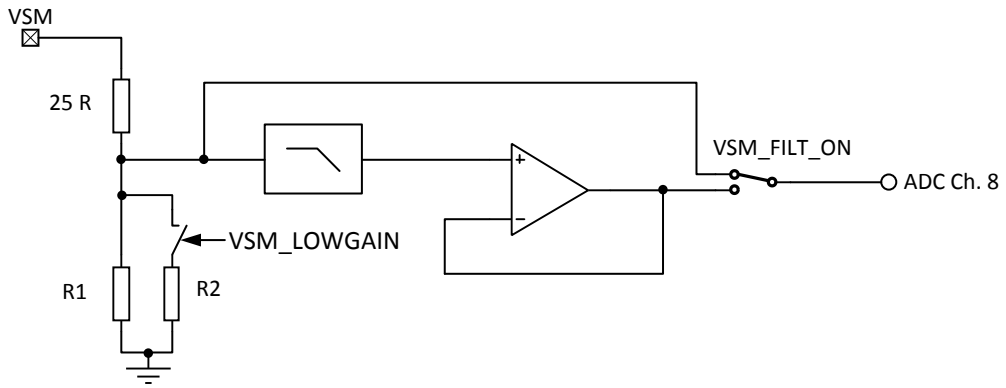


Figure 27 – VSM supply sensor block diagram

Port: PORT_MISC2_OUT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
VSM_LOWGAIN	12	RW	Select VSM sensor divider 0: 26 1: 52

DIS_TEMPSENSE	11	RW	Disable the ADC temperature sensor
VSM_FILT_ON	10	RW	Enable VSM supply filtering 0: no filtering 1: filter is on
ENABLE_OTD	9	RW	Enable over-temperature detection
WU_IO_EN	8	RW	Enable IO0 wake up circuit
VDSMON_VTH_SEL	[5:4]	RW	Select VDS monitor threshold 00: 0.5V 01: 1.0V 10: 1.5V 11: 2.0V
ENABLE_OSD_DRV	[1:0]	RW	Off state diagnostic mode 00: Off state diagnostic is off 01: Enable pull-up of U and pull-down of W 10: Enable pull-down of V 11: Enable pull-up of U and pull-down of V and W

Table 65 – PORT_MISC2_OUT

14.2.6. Over-temperature detection

The over-temperature detection block (see Figure 28) compares the voltage over a bipolar diode and a resistor with the bandgap voltage. The current through the resistor can be adjusted by TRIM_OTD[5:0] (port TRIM_MISC, see Table 66). The over-temperature block is enabled by setting ENABLE_OTD = 1 (port PORT_MISC2_OUT, see Table 65).

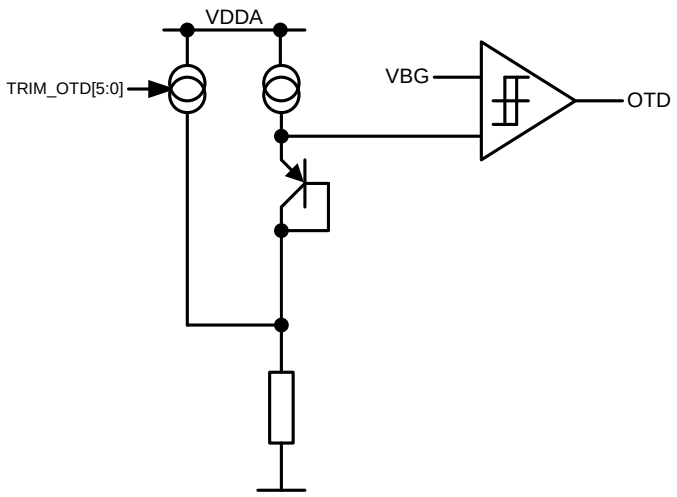


Figure 28 – Over-temperature detection block diagram

Port: TRIM_MISC

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
LOCK	15	RW	Lock the port, write is invalid when set
TRIM_SDA_FILT_IO	[7:6]	RW	Calibration of I^2C SDA filter, write is invalid when LOCK is set
TRIM_OTD	[5:0]	RW	Calibration of the over-temperature detection, write is invalid when LOCK is set

Table 66 – TRIM_MISC

14.2.7. Analog to Digital Converter (ADC)

14.2.7.1. Overview

The MLX81346 uses a 12bit ADC with a programmable DMA interface for supporting cyclic AD conversions at minimal CPU load (see Figure 29). The sequence of conversions is controlled by a RAM table that sets up the ADC channels, the trigger sources and the conversion modes. The result of the ADC conversions is also stored in a RAM table.

The ADC is controlled and monitored via ports shown in section 14.2.7.8. The ADC channel configuration will be defined in RAM fields, SDATA[x][15:0]. Each field contains the ADC channel to be sampled (Config 0, 1, ..., N) and the RAM address (DBASE[x][15:0]) where the result is saved (Sample 0, 1, ..., N).

To make the ADC usage more comfortable, the Melexis Software Platform offers a software library with macros and C-functions for configuring and controlling the ADC unit.

The ADC clock is the MCU clock (parameter MCU_CLK) divided by clock divider ADC_CLK_DIV in port PORT_ADC_CLK. The default divider value after reset is ADC_CLK_DIV[6:0] = 2 which sets the ADC clock to 1/3 of the MCU clock.

The formula for calculating the ADC clock is:

$$ADC\_CLK = \frac{MCU\_CLK}{ADC\_CLK\_DIV[6:0] + 1}$$

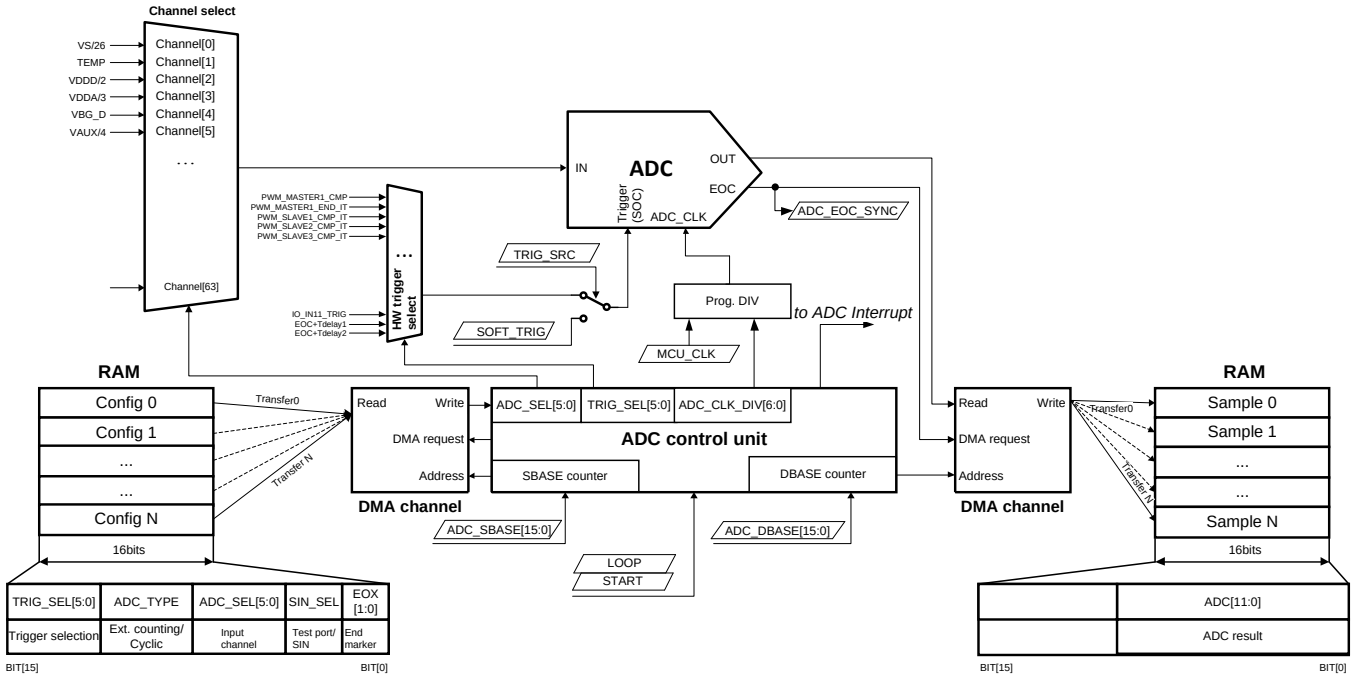


Figure 29 – Block Diagram of ADC Unit

14.2.7.2. DMA interface

The ADC is controlled and monitored via ports shown in section 14.2.7.8. The ADC channel configuration is defined by M frames located in the RAM. The number of frames M > 0 can be chosen by the customer. The starting RAM address of the first frame is set in the register SBASE_0 (port ADC_BLOCK, see Table 71). Each frame contains the RAM address (DBASE[i]) pointing to an ADATA table where the ADC results are stored and a list of ADC channels to be sampled (see Figure 29 and Figure 30).

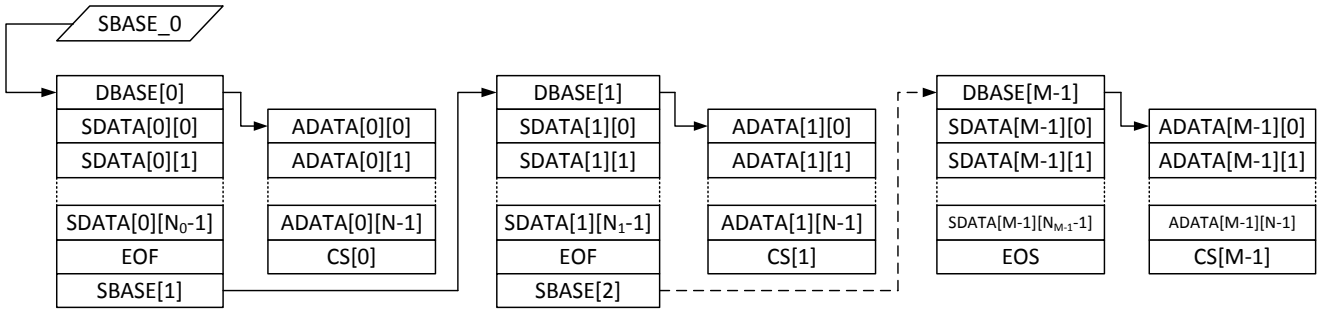


Figure 30 – Layout of ADC configuration fields.

All but the last frames are terminated by the EOF (End of Field) marker followed by the address SBASE[i+1] of the next frame. The last frame is terminated by the EOS (End of Sequence) marker. Note that it is an error to have empty frames without SDATA entries, i.e. there should be $N_i > 0$, i = 0, …, M-1.

Each entry SDATA[i][j] specifies a trigger source for the ADC conversion, an ADC channel, the conversion mode and the ADC channel (see Table 67). The two LSBs indicate a normal conversion, EOF (end-of-frame) or EOS (end-of-sequence). In case of EOF or EOS, the other bits, SDATA[i][15:3] are not evaluated.

RAM field: SDATA[i][j]

Field name	Bits	Description
TRIG_SEL[5:0]	[15:10]	Hardware trigger selection (see Table 69)
ADC_TYPE	[9]	Reserved, must be set to 1
ADC_SEL[5:0]	[8:3]	ADC channel (see Table 68)
SIN_SEL	[2]	Reserved for test purposes, should be 0
End marker EOx	[1:0]	0x: Normal conversion, use other fields 10: EOF (End Of Frame), other fields are ignored 11: EOS (End Of Sequence), other fields are ignored

Table 67 – SDATA RAM field definition for ADC DMA mode

The ADATA[i] table of each frame contains a 16bit checksum CS[i] calculated for every frame by the ADC interface:

$$CS[i] = \sum_{m=0}^{N_i-1} SDATA[i][m] + \sum_{n=0}^{N_i-1} ADATA[i][n]$$

It covers both, the SDATA table and the ADATA table. Firmware can compute its own checksum and verify if it matches the hardware computed one in case of high safety requirements.

14.2.7.3. Conversions timing

The ADC clock is the MCU clock (parameter MCU_CLK, see Table 13) divided by clock divider ADC_CLK_DIV (port PORT_ADC_CLK, see Table 73). The default divider value after reset is ADC_CLK_DIV[6:0] = 2 which sets the ADC clock 1/3 of the main clock MCU_CLK.

The formula for calculating the ADC clock is:

$$ADC\_CLK = \frac{MCU\_CLK}{ADC\_CLK\_DIV[6:0] + 1}$$

Conversions are launched by ADC interface upon trigger events. The first conversion of a sequence is triggered by a STRIG event and all next conversions are triggered by CTRIG events. A simplified hardware schematic of STRIG generation is shown on Figure 31 and the corresponding conversion behavior on Figure 32. The behavior of PAUSE and RESUME ports is shown on Figure 33.

The source of the trigger can be configured by SOC_SOURCE and SOS_SOURCE (port ADC_BLOCK, see Table 70) to be Hardware (HARD_xTRIG) or Software (SOFT_xTRIG) or Permanent.

The Software trigger is generated inside the ADC Interface by writing logic 1 to the SW_TRIG field of the STATUS port (see Table 72).

Note: The SW_TRIG is not memorized and should be engaged in case the ADC Interface waits for a trigger (STATE = 11), otherwise it is missed.

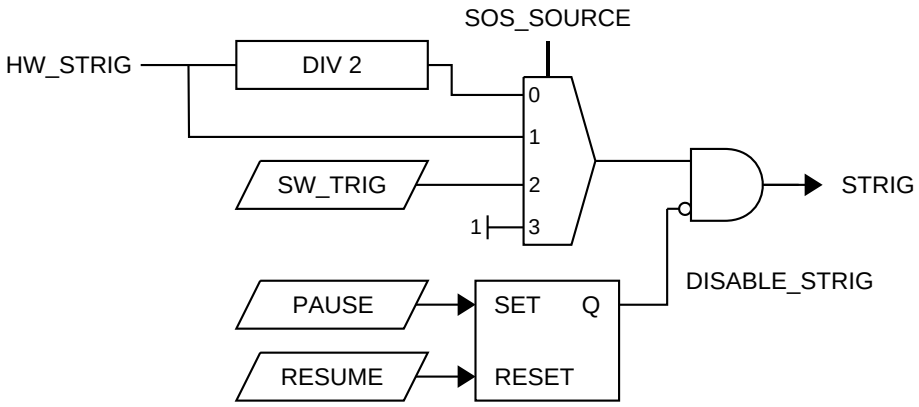


Figure 31 – STRIG generation

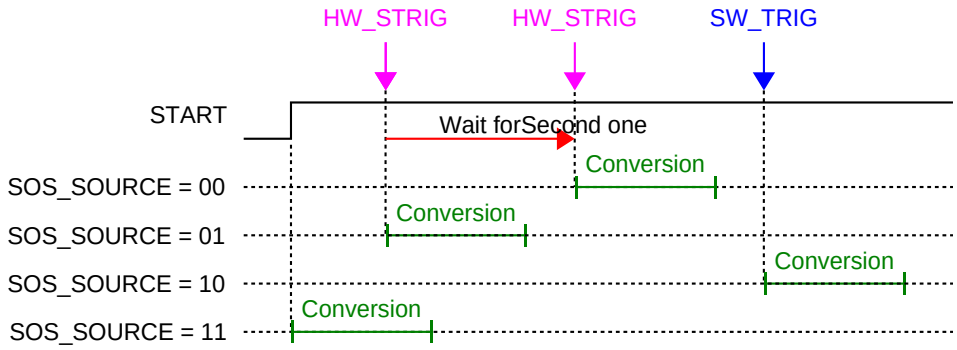


Figure 32 – First conversion triggering

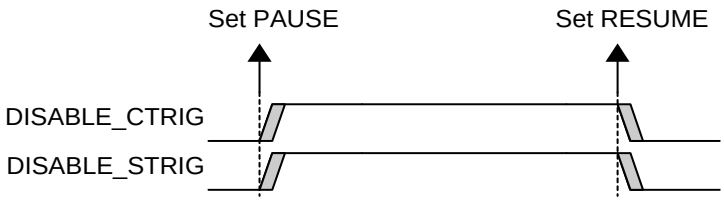


Figure 33 – PAUSE and RESUE effects.

A simplified hardware schematic of CTRIG generation is shown on Figure 34 and the corresponding conversion behavior on Figure 35. The behavior of PAUSE and RESUME ports is shown on Figure 36.

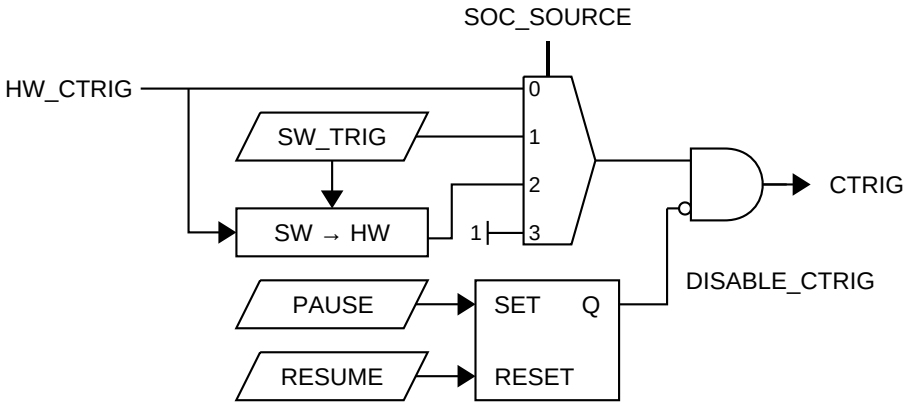


Figure 34 – CTRIG generation

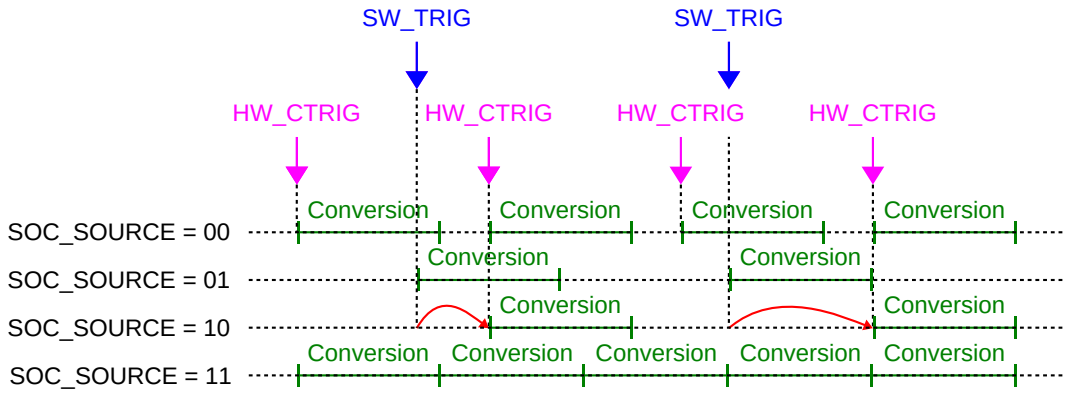


Figure 35 – Conversion triggering

Note: In the next chapters, the name xTRIG is used when it applies to both STRIG and CTRIG.

An example ADC conversion sequence is shown in Figure 36.

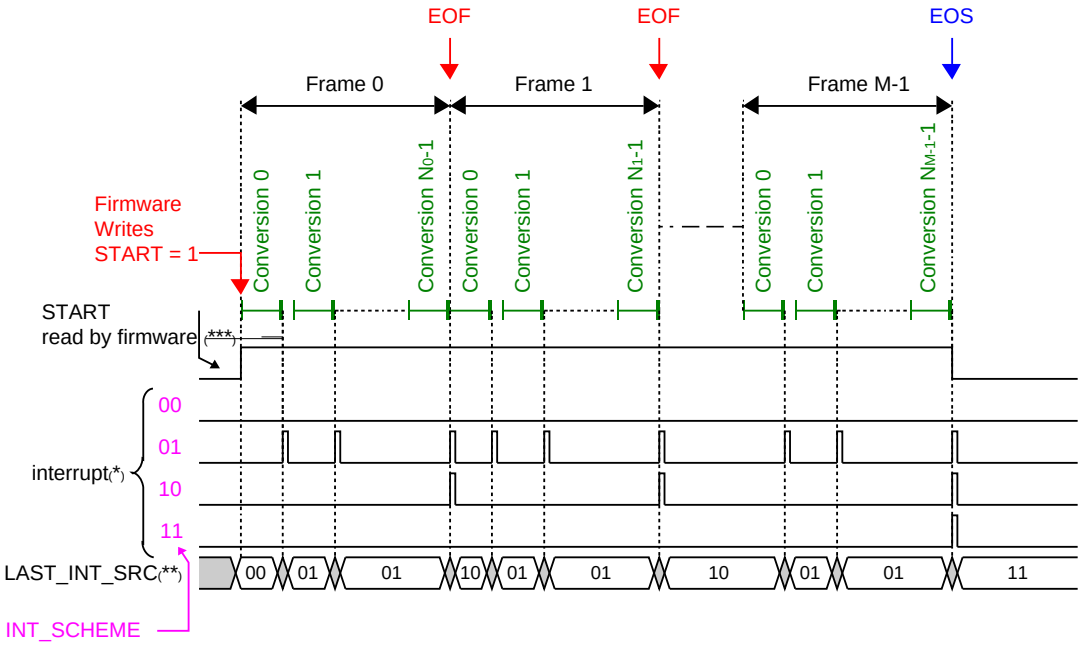


Figure 36 - ADC Interface timing example

14.2.7.4. Conversion algorithm

The ADC applies the cyclic conversion algorithm. The input voltage is compared to VREF_ADC. If VREF_ADC is bigger than the input voltage, VREF_ADC is subtracted and the corresponding output bit (from MSB to LSB) is set to '1', otherwise nothing is subtracted and the corresponding output bit is set to '0'. The resulting voltage after this operation is multiplied by 2 and the process is repated for the next output bit.

14.2.7.5. Channel selection

The ADC channel is selected by control bits ADC_SEL[5:0] (= SDATA[i][8:3]), see Table 68 – ADC channel selection) here below.

Dec	ADC_SEL[5:0]	ADC_TYPE	HV/LV	Source	Input Resistance [kΩ] [21]	Description
0	000000	1	HV	VS/26	263	Supply voltage divided by 26
1	000001	1	LV	TEMP	-	Temperature sensor
2	000010	1	LV	VDDD/2	100	Digital supply voltage (1.8V) divided by 2
3	000011	1	LV	VDDA/3	100	Analog supply voltage (3.3V) divided by 3
4	000100	1	LV	VBG_D	-	Digital bandgap voltage
5	000101	1	LV	VAUX/4	226	Auxiliary (always-on) supply voltage divided by 4
6	000110	1	LV	LIN AA	-	not used
7	000111	1	LV	LIN AA	-	not used
8	001000	1	HV	VSM/26_F	301 (Gain 26) 295 (Gain 52)	Filtered motor supply voltage divided by 26 (VSM_LOW_GAIN = 0) or 52 (VSM_LOW_GAIN = 1)
9	001001	1	HV	VBOOST/32	251	VBOOST voltage divided by 32
10	001010	0	HV	CSOUT	-	Current sense ampilfier output voltage
11	001011	1	LV	LIN/3	100	LIN pin voltage (after reverse polarity switch) divided by 3
12	001100	1	LV	IO0/2.5	100	IO0 voltage divided by 2.5
13	001101	1	LV	IO1/2.5	100	IO1 voltage divided by 2.5

[21] *typ. values; for information only; not tested in production*

Dec	ADC_SEL[5:0]	ADC_TYPE	HV/LV	Source	Input Resistance [kΩ] [21]	Description
14	001110	1	LV	IO2/2.5	100	IO2 voltage divided by 2.5
15	001111	1	LV	IO3/2.5	100	IO3 voltage divided by 2.5
16	010000	1	LV	IO4/2.5	100	IO4 voltage divided by 2.5
17	010001	1	LV	IO5/2.5	100	IO5 voltage divided by 2.5
18	010010	1	LV	IO6/2.5	100	IO6 voltage divided by 2.5
19	010011	1	LV	IO7/2.5	100	IO7 voltage divided by 2.5
20	010100	1	LV	IO8/2.5	100	IO8 voltage divided by 2.5
21	010101	1	LV	IO9/2.5	100	IO9 voltage divided by 2.5
22	010110	1	LV	IO10/2.5	100	IO10 voltage divided by 2.5
23	010111	1	LV	IO11/2.5	100	IO11 voltage divided by 2.5
24	011000	1	HV	IO0/26	263	IO0 voltage divided by 26
25	011001	1	HV	IO1/26	263	IO1 voltage divided by 26
26	011010	1	HV	IO2/26	263	IO2 voltage divided by 26
27	011011	1	HV	U/26	301	U phase output divided by 26
28	011100	1	HV	V/26	301	V phase output divided by 26
29	011101	1	HV	W/26	301	W phase output divided by 26
30	011110	1	LV	TA0	-	reserved channel
31	011111	1	LV	TA1	-	reserved channel
32					-	reserved channel
33	100001	1	LV	VSSA	-	Analog ground
34-40						reserved channels
41	101001	1	HV	VBOOST/64	247	VBOOST voltage divided by 64
42-58						reserved channels
59		1	U/52		295	U phase output divided by 52
60		1	V/52		295	V phase output divided by 52
61		1	W/52		295	W phase output divided by 52
62-63	1xxxxx	-	-	-		reserved channels

Table 68 – ADC channel selection

14.2.7.6. ADC Reference

The ADC uses a fixed reference voltage of VREF_ADC = 1.48V, derived from the analog bandgap. Calibration of the ADC reference is included in the gain calibration of the ADC. See also Table 22 for the electrical specifications of the ADC and [1] for the calibration of the ADC.

14.2.7.7. Hardware trigger selection

The ADC hardware trigger is selected by control bits TRIG_SEL[5:0] (= SDATA[i][15:10]), see Table 69 below.

TRIG_SEL[5:0]	ADC trigger (CTRIG)	Description
000000	PWM_MASTER1_CMP_IT	PWM compare interrupt
000001	PWM_MASTER1_END_IT	PWM counter interrupt
000010	PWM_SLAVE1_CMP_IT	PWM compare interrupt
000011	PWM_SLAVE2_CMP_IT	PWM compare interrupt
000100	PWM_SLAVE3_CMP_IT	PWM compare interrupt
000101	PWM_MASTER2_CMP_IT	PWM compare interrupt
000110	PWM_MASTER2_END_IT	PWM counter interrupt
000111	IO_IN0_TRIG	IO trigger, undebounced
001000	IO_IN1_TRIG	IO trigger, undebounced
001001	IO_IN2_TRIG	IO trigger, undebounced
001010	IO_IN3_TRIG	IO trigger, undebounced
001011	CTIMER0_COMPLEX_TIMER_1_IT	Complex timer interrupt 1
001100	CTIMER0_COMPLEX_TIMER_2_IT	Complex timer interrupt 2
001101	CTIMER0_COMPLEX_TIMER_3_IT	Complex timer interrupt 3
001110	CTIMER1_COMPLEX_TIMER_1_IT	Complex timer interrupt 1
001111	CTIMER1_COMPLEX_TIMER_2_IT	Complex timer interrupt 2
010000	CTIMER1_COMPLEX_TIMER_3_IT	Complex timer interrupt 3
010001	IO_IN4_TRIG	IO trigger, undebounced
010010	IO_IN5_TRIG	IO trigger, undebounced
010011	IO_IN6_TRIG	IO trigger, undebounced
010100	IO_IN7_TRIG	IO trigger, undebounced
010101	IO_IN8_TRIG	IO trigger, undebounced
010110	IO_IN9_TRIG	IO trigger, undebounced
010111	IO_IN10_TRIG	IO trigger, undebounced

TRIG_SEL[5:0]	ADC trigger (CTRIG)	Description
011000	IO_IN11_TRIG	IO trigger, undebounced
100xxx[22]	EOC + Tdelay1	Tdelay = (xxx + 1)×2/MCU_CLOCK (where MCU_CLK is the frequency of the main clock, see sections 13.3.3 and 14.2.1)
110xxx[22]	EOC + Tdelay2	Tdelay = (xxx + 1)×2/ADC_CLK (see section 14.2.7.3 for ADC_CLOCK)

Table 69 – ADC hardware trigger selection bits

The ADC sample timing is typically synchronized to the PWM cycle timing. A typical use-case for BLDC motor applications is to have 6 ADC sampling events per PWM cycle. To trigger the ADC sampling, following interrupt signals (generated by the PWM units) are used:

- PWM1_CMP_IT, programmed to $\frac{1}{6}$ of the PWM period
- PWM2_CMP_IT, programmed to $\frac{2}{6}$ of the PWM period
- PWM3_CMP_IT, programmed to $\frac{3}{6}$ of the PWM period
- PWM4_CMP_IT, programmed to $\frac{4}{6}$ of the PWM period
- PWM5_CMP_IT, programmed to $\frac{5}{6}$ of the PWM period
- PWM1_END_IT, which always occurs at the end of the PWM period ($\frac{6}{6}$)

For settling of slow analog sources an additional delay before the start of conversion can be set by CM_PRCHRG and CM_PRCHRG_TIME (port ADC_BLOCK, see Table 73).

14.2.7.8. ADC ports map

Port: ADC_BLOCK

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0002	Word, Byte, Invalid

Field name	Bit	R/W	Description
-	[15:13]		Not used
ADC_WIDTH	12	RW	ADC data width: 0: Up to 16 bits → Result is one word per ADC 1: Up to 32 bits → Result is two word per ADC

[22] *The EOC trigger allows to start the next AD conversion with a delay after the end of the current conversion. SDATA[15]=1 enab les the EOC trigger. SDATA[14] selects the delay clock. Either the ADC clock (ADC_CLK) or the MCU clock (MCU_CLK) can be used. SDATA[12:10] select the number of clocks for the delay time.*

Address	Reset		Access
ASB	[11:10]	RW	Auto standby: 00: ADC is in stand-by only when not used (START = 0) 01: ADC is also in standby while waiting for triggers 1x: ADC I never in stand-by
INT_SCHEME	[9:8]	RW	Message interrupt and: 00: No interrupt 01: Interrupt at each end of conversion 10: Interrupt at each end of frame 11: Interrupt at end of sequence
SATURATE	7	RW	0: ADC data is garbage in case of overflow or underflow 1: ADC result list is saturated to: 2^{N-1} in case of overflow (for a N bit sample data) 0 in case of underflow
NO_INTERLEAVE	6	RW	0: EOA triggers configuration update 1: EOC triggers configuration update
SOC_SOURCE	[5:4]	RW	Start Of Conversion (SOC) triggered by: 00: Hardware (HARD_CTRIG) 01: Firmware (SOFT_TRIG) 10: Hardware (HARD_CTRIG) validated by firmware (SOFT_TRIG) 11: Permanent
SOS_SOURCE	[3:2]	RW	Start Of Sequence (SOS) triggered by: 00: Second hardware trigger (HARD_STRIG) 01: First hardware trigger (HARD_STRIG) 10: Firmware (SOFT_TRIG) 11: Permanent
STOP	1	W	1: Stop the ADC, all other bits are discarded 0: No effect
		R	0: ADC is running 1: ADC is stopped
START	0	W	1: Start the ADC, all other bits are discarded 0: No effect
		R	0: ADC is stopped 1: ADC is running

Table 70 – ADC_BLOCK reg 0

Port: ADC_BLOCK

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
SBASE_0	[15:0]	W	Initial SBASE pointer
		R	Current pointer to SDATA

Table 71 – ADC_BLOCK reg 1

Port: ADC_BLOCK

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0002	Word, Byte, Invalid

Field name	Bit	R/W	Description
-	[15:13]		Not used
ABORTED	12	R	0: Sequence running normally 1: Sequence has been aborted
		W	0: No effect 1: Clear flag
FRAME_ERR	11	R	0: No frame error 1: A Frame error occurred
		W	0: No effect 1: Clear flag
MEM_ERR	10	R	0: No error occurred 1: A memory error occurred
		W	0: No effect 1: Clear flag
ADC_ERR	9	R	0: No ADC error occurred 1: An ADC error occurred
		W	0: No effect 1: Clear flag
ADC_OVF	8	R	0: No error occurred 1: An error occurred
		W	0: No effect 1: Clear flag
STATE	[7:6]	R	00: Idle 01: Memory transfer10: Conversion ongoing11: Waiting for trigger
		W	No effect
LAST_INT_SRC	[5:4]	R	00: No interrupt occurred since last start of ADC 01: Last interrupt is due to an end of conversion 10: Last interrupt is due to an end of frame 11: Last interrupt is due to an end of sequence
		W	No effect
READY	3	R	READY signal from ADC
		W	No effect
SW_TRIG	2	W	1: Triggers a Start of Sequence or a Start of Conversion 0: No effect
		R	Always 0
RESUME	1	W	1: Enable all triggers, all other bits are discarded 0: No effect
		R	0: All triggers are disabled 1: All triggers are enabled

Address	Reset		Access
PAUSE	0	W	1: Disable all triggers, all other bits are discarded 0: No effect
		R	0: All triggers are enabled 1: All triggers are disabled

Table 72 – ADC_BLOCK reg 2

Port: ADC_BLOCK

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0002		Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
-	15		Not used
CM_PRCHRG	[14:13]	RW	Should be set 00
CM_PRCHRG_TIME	[12:10]	RW	Pre-charge time: $T_{PRECHARGE}=2^{CM\_PRCHRG\_TIME[2:0]+1}/ADC\_CLK$
CLK_TO_INT	[9:8]	RW	Should be set 00
-	[7]		Not used
ADC_CLK_DIV	[6:0]	RW	ADC_CLK = MCU_CLK / (ADC_CLK_DIV + 1)

Table 73 – ADC_BLOCK reg 3

Port: ADC_BLOCK

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Test, Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
REF_ALWAYS_ON	15	RW	ADC analog test and internal reference buffers configuration 0 : driven by ADC standby signal 1 : leave on between conversions
SPEED	[14:13]	RW	Bias current selection (saves current, but needs slower ADC_CLK) 00 : full bias 01 : 2/3 bias 10 : 1/2 bias 11 : 1/3 bias
INTREF	12	RW	Should be set 1
NOCHOP	11	RW	Should be set 0
FORCE	[10:9]	RW	Should be set 00
OUTMODE	8	RW	Should be set 0
START_PHI	7	RW	Should be set 0

Address	Reset		Access
FR	[6:5]	RW	Change the ADC Frequency 00: ADC_CLK used 01: ADC_CLK / 2 used 10: ADC_CLK / 4 used 11: ADC_CLK / 8 used
MODE	4	RW	Defines the operation mode 00 : normal mode 01 : selftest mode
TYPE	3	R	Set by SDATA (see section 14.2.7.2) for the current conversion Type of conversion 0: Extended Counting 1: Cyclic-only
COUNT	[1:0]	RW	Define number of counting step during an extended conversion #Steps = 2∗(COUNT+3); Valid values are 0, 1 and 2

Table 74 – ADC_BLOCK reg 4

14.2.8. General-purpose IO

14.2.8.1. Features

The general-purpose IO block shown in Figure 40 contains 12 low voltage, general-purpose IOs. All IO pins provide bi-directional digital IO functionality and low-voltage ADC measurement (range: VDDA, input divider 2.5).

For MLX81346, IO0-IO4 pins additionally provide high-voltage digital input and open-drain/push-pull (OD/PP) output capability while IO0-IO2 also feature high-voltage ADC inputs (range: VS, input divider 26). The high-side OD outputs of IO3 and IO4 can drive higher currents up to 50mA. A VDS monitor checks the drain-source voltage of the HS driver transistor of IO3 and IO4 against a fixed threshold. A port bit is set (OV_HS_IO[1:0]) and an interrupt is generated if this threshold is exceeded to signal a short or over current condition at the respective IO.

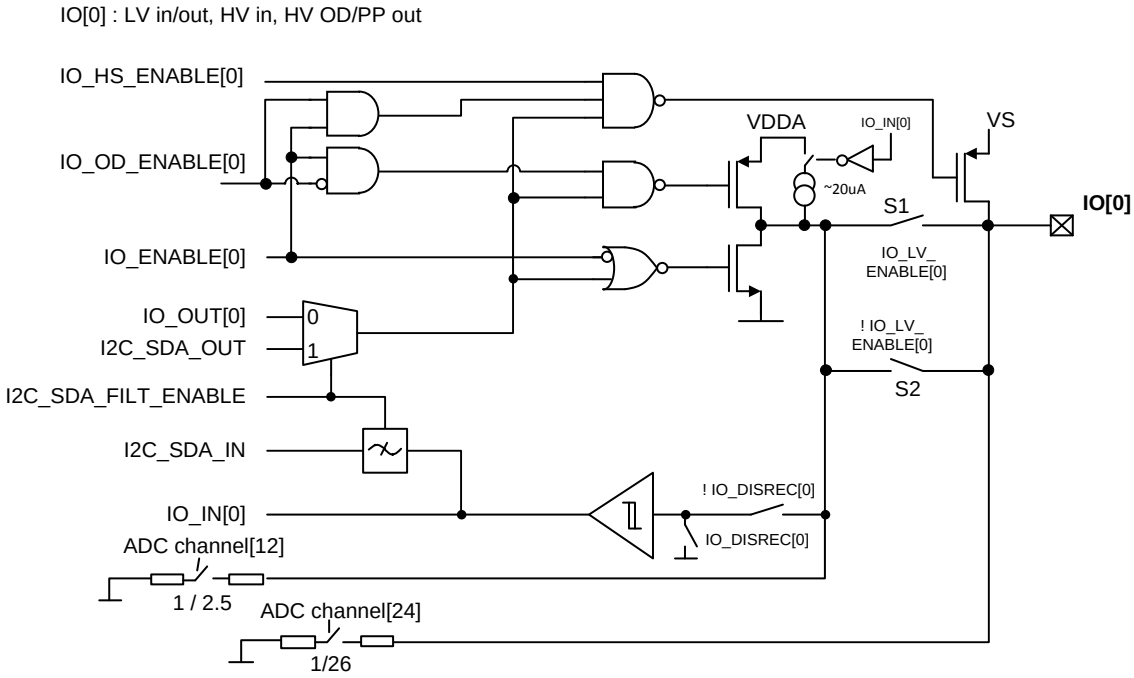


Figure 37 – IO0 block diagram

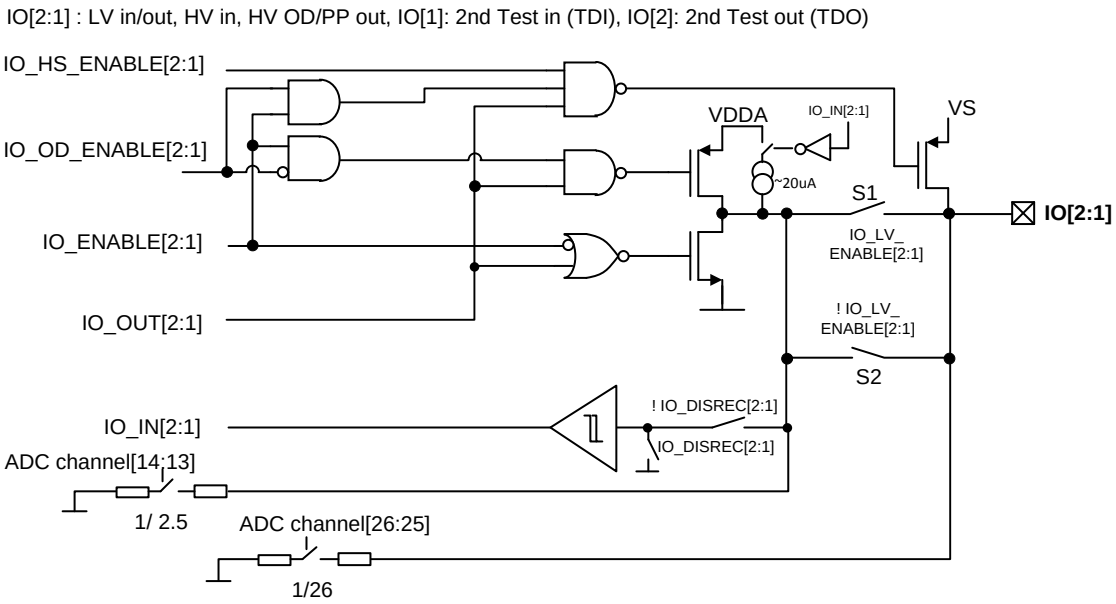


Figure 38 – IO[2:1] block diagram

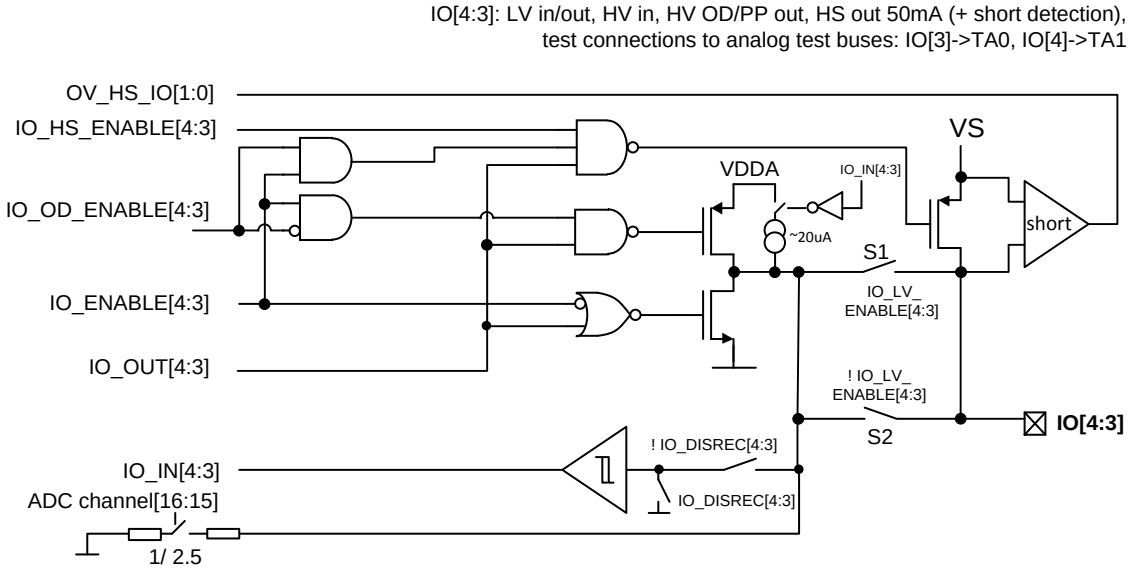


Figure 39 – IO[4:3] block diagram

Note: Please consider the internal switches S1 and S2 for the HV IOs IO[4:0]. S1 is a low-ohmic, low voltage switch to allow the use of the IO as low voltage output. The gate of the switching transistor of S1 is controlled by voltages >VDDA. S2 is a switch controlled by VDDA at its gate. The HV from outside will be limited to VDDA internally. This switch must be activated by the corresponding portsmap bit when the IO is used as HV input. Be aware that applying voltages >VDDA with S1 open may lead to big cross-current in the IO cell and has therefore the risk of IC damage.

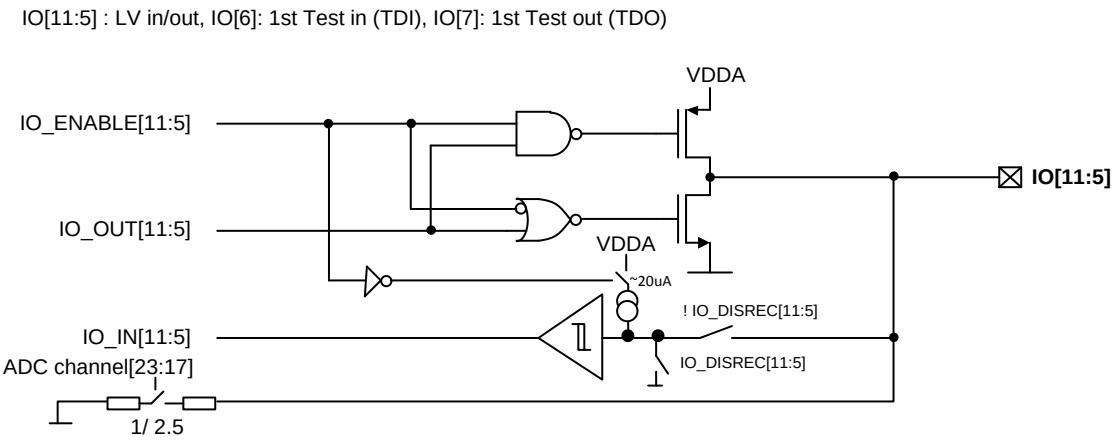


Figure 40 – IO[11:5] block diagram

Note on switches in the IO drawings:
The switches are displayed with their control signal as closing switches. Setting the control signal to high will close the switch. Inverted control signals are marked by exclamation marks,
i.e. the notation "! IO_DISREC" means the switch is closed when IO_DISREC is zero (what is the reset value of IO_DISREC)

Pin	Description
IO0	Dig: LV in/out, HV in, HV push pull / open-drain out; Analog (ADC ch.): LV in, HV in
IO1	Dig: LV in/out, 2nd Test in (TDI), HV in, HV push pull /open-drain out; Analog (ADC ch.): LV in, HV in
IO2	Dig: LV in/out, 2nd Test out (TDO), HV in, HV push pull /open-drain out; Analog (ADC ch.): LV in, HV in
IO3	Dig: LV in/out, HV in, HV PP/OD out, HS out (50mA + short detection); Analog (ADC ch.): LV in
IO4	Dig: LV in/out, HV in, HV PP/OD out, HS out (50mA + short detection); Analog (ADC ch.): LV in
IO5	Dig: LV in/out; Analog (ADC ch.): LV in
IO6	Dig: LV in/out, 1st Test in (TDI); Analog (ADC ch.): LV in
IO7	Dig: LV in/out, 1st Test out (TDO); Analog (ADC ch.): LV in
IO8	Dig: LV in/out; Analog (ADC ch.): LV in
IO9	Dig: LV in/out; Analog (ADC ch.): LV in
IO10	Dig: LV in/out; Analog (ADC ch.): LV in
IO11	Dig: LV in/out; Analog (ADC ch.): LV in

Table 75 – IO pin functionality

The general-purpose IO block is controlled by PORT_IO_OUT_EN and PORT_IO_ENABLE:

Port: PORT_IO_OUT_EN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO_HV_ENABLE IO_LV_ENABLE	[14:10]	RW	Activates the high voltage stages of IO[4:0] Activates the low voltage stages of IO[4:0]
IO_HS_ENABLE	[9:5]	RW	Activates the HS transistor for IO[4:0]
IO_OD_ENABLE	[4:0]	RW	Activates the open drain mode for IO[4:0]

Table 76 – PORT_IO_OUT_EN

Port: PORT_IO_ENABLE

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO_DISREC	[15:8]	RW	0: IO[7:0] connect input receiver of IO_IN[7:0], see Figure 38 - Figure 40 1: input receiver of IO_IN[7:0] is disconnected, IO pin is used as ADC input
IO_ENABLE	[7:0]	RW	Set the state of the IO interface, IO[7:0]

Table 77 – PORT_IO_ENABLE

Port: PORT_IO_ENABLE1

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO_DISREC	[7:4]	RW	0: IO[11:8] connect input receiver of IO_IN[11:8], see Figure 40 1: input receiver of IO_IN[11:8] is disconnected, IO pin is used as ADC input
IO_ENABLE	[3:0]	RW	Set the state of the IO interface [11:8]

Table 78 – PORT_IO_ENABLE1

14.2.8.2. Digital IO output configuration

Figure 41 shows the digital IO output configuration where several digital signals (PWM, timer, IO software port, LIN, SPI, PPM) can be multiplexed to any IO pin. The multiplexer is described in Table 79.

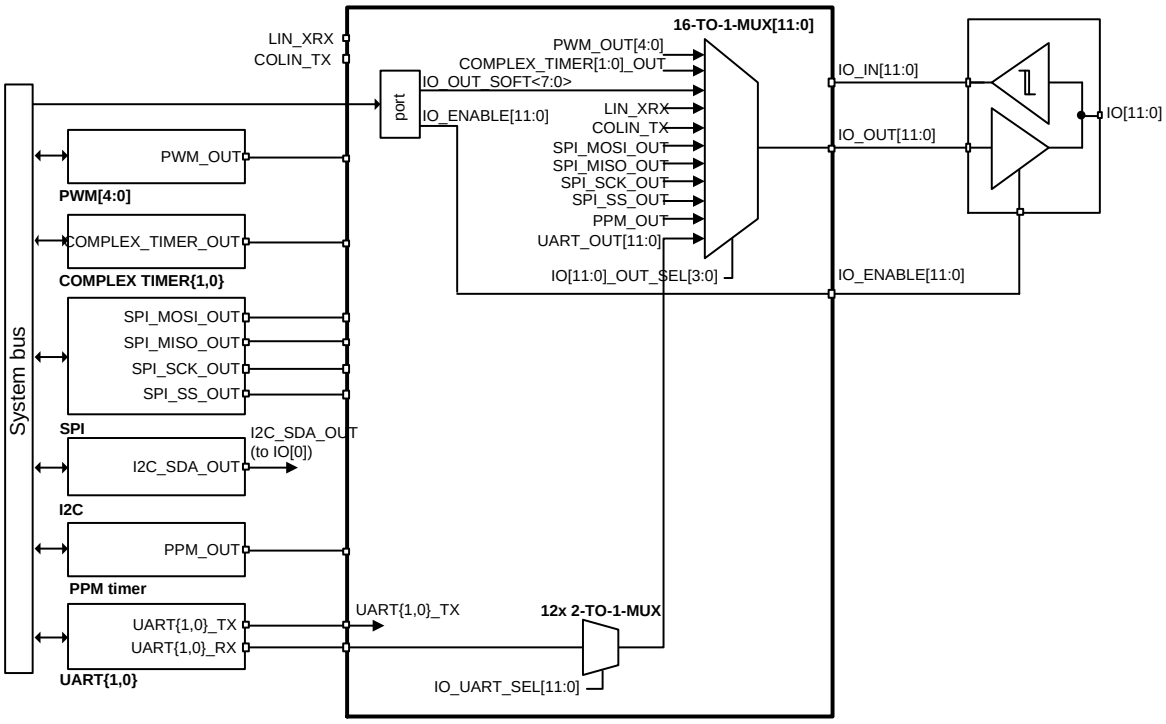


Figure 41 – Digital IO output configuration

IO[x]_OUT_SEL[3:0]	IO_OUT[x]
0000	PWM0_OUT (PWM_MASTER1)
0001	PWM1_OUT (PWM_SLAVE1)
0010	PWM2_OUT (PWM_SLAVE2)
0011	PWM3_OUT (PWM_SLAVE3)
0100	PWM4_OUT (PWM_MASTER2)
0101	COMPLEX_TIMER0_OUT
0110	COMPLEX_TIMER1_OUT
0111	IO_OUT_SOFT[x]
1000	LIN_XRX

1001	COLIN_TX
1010	SPI_MOSI_OUT
1011	SPI_MISO_OUT
1100	SPI_SCK_OUT
1101	SPI_SS_OUT
1110	PPM_OUT
1111	UART_OUT

Table 79 – Digital IO output selection

IO_UART_SEL[x]	UART selector port
0	IO[x] is connected to UART0
1	IO[x] is connected to UART1

Table 80 – UART selector (for IO[x]_OUT_SEL=15)

The digital IO output configuration is controlled by PORT_IO_CFG0, PORT_IO_CFG1 PORT_IO_CFG2 and PORT_IO_UART_SEL.

Port: PORT_IO_CFG0

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO3_OUT_SEL	[15:12]	RW	Output selection for IO3
IO2_OUT_SEL	[11:8]	RW	Output selection for IO2
IO1_OUT_SEL	[7:4]	RW	Output selection for IO1
IO0_OUT_SEL	[3:0]	RW	Output selection for IO0

Table 81 – PORT_IO_CFG0

Port: PORT_IO_CFG1

Address	Reset		Access

See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO7_OUT_SEL	[15:12]	RW	Output selection for IO7
IO6_OUT_SEL	[11:8]	RW	Output selection for IO6
IO5_OUT_SEL	[7:4]	RW	Output selection for IO5
IO4_OUT_SEL	[3:0]	RW	Output selection for IO4

Table 82 – PORT_IO_CFG1

Port: PORT_IO_CFG2

Address	**Reset**		**Access**
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO11_OUT_SEL	[15:12]	RW	Output selection for IO11
IO10_OUT_SEL	[11:8]	RW	Output selection for IO10
IO9_OUT_SEL	[7:4]	RW	Output selection for IO9
IO8_OUT_SEL	[3:0]	RW	Output selection for IO8

Table 83 – PORT_IO_CFG2

Port: PORT_IO_UART_SEL

Address	**Reset**		**Access**
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO_UART_SEL	[11:0]	RW	Select UART0 or UART1 for corresponding IO output (see Table 80)

Table 84 – PORT_IO_UART_SEL

14.2.8.3. Digital IO input configuration

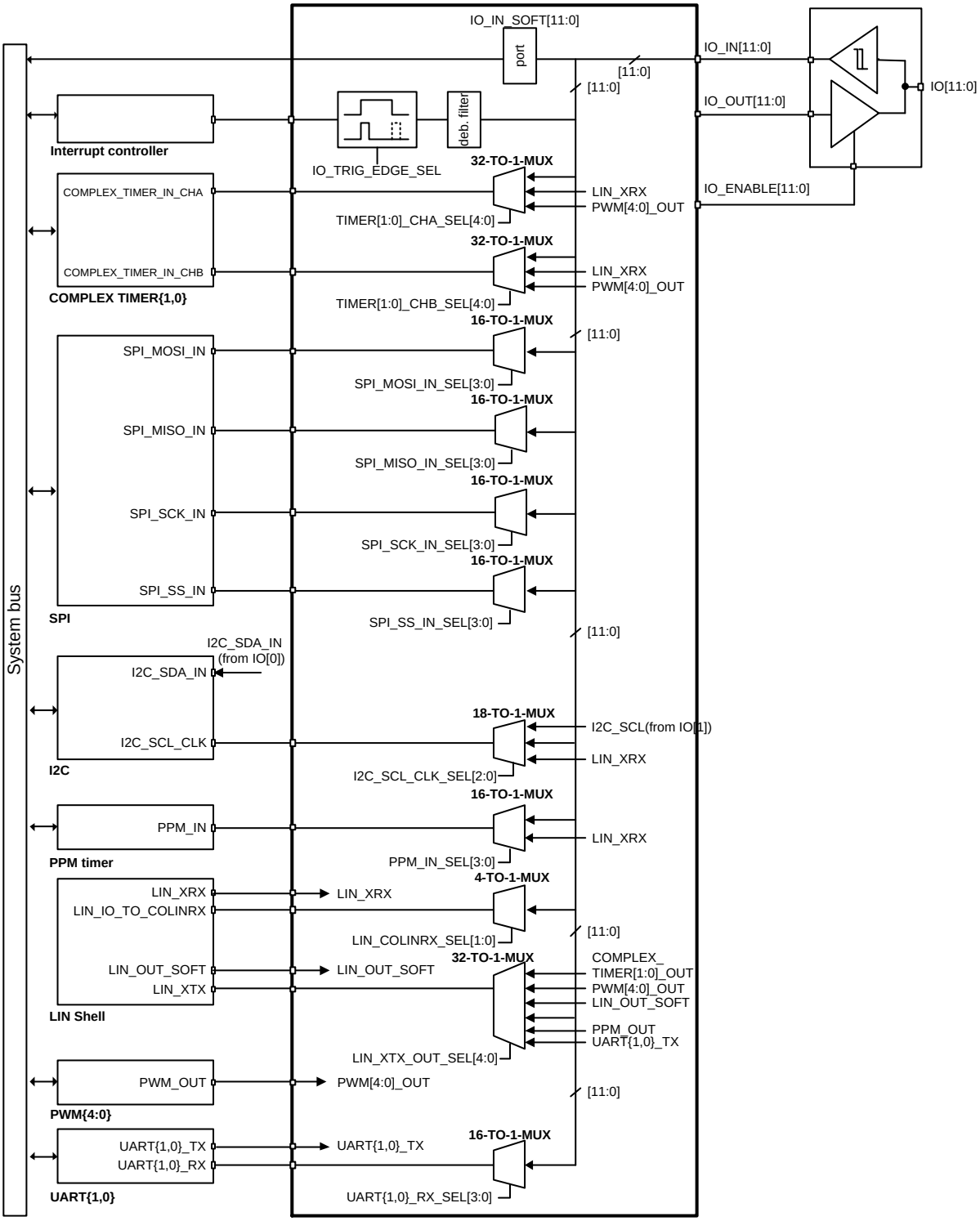


Figure 42 – Digital IO input configuration

Figure 42 shows the digital IO input configuration where several digital functions (timer, IO port, SPI, I^2C, PPM, LIN) can be accessed by all 12 IO pins.

The IO inputs are captured in port PORT_IO_IN (see Table 93) according to PORT_I2C_CFG (see Table 101) and PORT_PPM_CTRL (see Table 102).

Each input IO[11:0] can trigger an interrupt. The IO inputs are debounced by a filter before a trigger pulse is generated. The filter debounces both, rising and falling edges, where a pulse shorter than 1µs is not passed, while a pulse longer than 2µs is passed. The trigger pulse is then generated according to Table 85, controlled by ports PORT_IO_TRIG_EDGE_CFG:IO[7:0]_TRIG_EDGE_SEL and PORT_IO_TRIG_EDGE_CFG1:IO[11:8]_TRIG _EDGE_SEL.

IO[x]_TRIG_EDGE_SEL[1:0]	Select IO_IN[x] trigger pulse
00	On rising edge
01	On falling edge
10	On rising and falling edges
11	No pulse shaping → level-sensitive triggering

Table 85 – IO trigger edge selection for IO interrupt

14.2.8.3.1. Digital IO as capture input

The inputs IO[11:0] can also be connected to the capture inputs of both complex timers TIMER{0,1}. The multiplexers for channel A and B of these timers are described in Table 86. Next to a connection to the 12 IO pins, these multiplexers also enable access through LIN_XRX and PWM_OUT[4:0]. The timer input selection is controlled by ports PORT_TIMER_CFG:TIMER0_CH{A,B}_SEL and PORT_TIMER_CFG1:TIMER1_CH{A,B}_SEL.

TIMER{1,0}_CH{A,B}_SEL[4:0]	Select input for TIMER{1,0}, CH{A,B}
00000	IO_IN[0]
00001	IO_IN[1]
00010	IO_IN[2]
00011	IO_IN[3]
00100	LIN_XRX
00101	PWM0_OUT (PWM_MASTER1)
00110	PWM1 _OUT (PWM_SLAVE1)
00111	PWM2 _OUT (PWM_SLAVE2)
01000	PWM3 _OUT (PWM_SLAVE3)
01001	PWM4 _OUT (PWM_MASTER2)

01010	IO_IN[4]
01011	IO_IN[5]
01100	IO_IN[6]
01101	IO_IN[7]
01110	IO_IN[8]
01111	IO_IN[9]
10000	IO_IN[10]
10001	IO_IN[11]
10010-11111	'0'

Table 86 – TIMER{0,1} channel{A,B} input selection

14.2.8.3.2. Digital IO as SPI input

The input multiplexer for the SPI function is controlled by port PORT_COMM_CFG:SPI_{MOSI,MISO,SCK,SS}_IN_ SEL according to the description in Table 87.

SPI_{MOSI,MISO,SCK,SS}_IN_SEL[3:0]	Select input for SPI signals {MOSI,MISO,SCK,SS}
0000	IO_IN[0]
0001	IO_IN[1]
0010	IO_IN[2]
0011	IO_IN[3]
0100	IO_IN[4]
0101	IO_IN[5]
0110	IO_IN[6]
0111	IO_IN[7]
1000	IO_IN[8]
1001	IO_IN[9]
1010	IO_IN[10]
1011	IO_IN[11]
1100-1111	'0'

Table 87 – SPI input selection

14.2.8.3.3. Digital IO as I²C input

The SDA data input of the I²C function is connected to pin IO0, shown in Figure 42/Figure 37, after an analog filter that ensures a 300ns delay with regards to signal SCL. The SDA data output is similarly connected to pin IO0 through an analog delay filter, as shown in Figure 41/Figure 37. The input selection for the I²C SCL signal is controlled by port PORT_I2C_CFG:I2C_SCL_CLK_SEL[11:9] according to the description in Table 88.

I2C_SCL_CLK_SEL[2:0]	Select I²C clock signal SCL from
000	I2C_SCL (from IO1)
001	IO_IN[2]
010	IO_IN[3]
011	LIN_RXD
100	IO_IN[4]
101	IO_IN[5]
110	IO_IN[6]
111	IO_IN[7]

Table 88 – I²C SCL input selection

14.2.8.3.4. Digital IO as PPM input

The input selection for the PPM function is controlled by port PORT_PPM_CTRL:PPM_IN_SEL[14:11] according to the description in Table 89.

PPM_IN_SEL[3:0]	Select input for PPM interface
0000	LIN_XRX
0001	IO_IN[0]
0010	IO_IN[1]
0011	IO_IN[2]
0100	IO_IN[3]
0101	IO_IN[4]
0110	IO_IN[5]
0111	IO_IN[6]
1000	IO_IN[7]
1001	IO_IN[8]
1010	IO_IN[9]
1011	IO_IN[10]

1100	IO_IN[11]
1101-1111	'1'

Table 89 – PPM input selection

14.2.8.3.5. Digital IO as input for LIN conformance testing

The input selection multiplexers for the LIN function are described in Table 90 and Table 91. For LIN conformance testing, the LIN PHY's RX signal can be bypassed by the signal connected to LIN_IO_TO_COLINRX, while the TX signal can be bypassed by the signal connected to LIN_XTX. Thereby, an external device can test the conformance of the LIN PHY to the LIN standard by using IO pins. Additionally, a set of other digital signals can be multiplexed to LIN_XTX for debugging purposes. The input selection for these LIN functions are controlled by port PORT_LIN_XTX_CFG:LIN_COLIN_RX_SEL[6:5] and PORT_LIN_XTX_CFG:LIN_XTX_OUT_SEL[4:0] according to the descriptions in Table 90 and Table 91.

LIN_COLIN_RX_SEL[1:0]	Select input for LIN_IO_TO_COLINRX
00	IO_IN[0]
01	IO_IN[1]
10	IO_IN[2]
11	IO_IN[3]

Table 90 – LIN_IO_TO_COLINRX input selection

LIN_XTX_OUT_SEL[4:0]	LIN_XTX
00000	COMPLEX_TIMER0_OUT
00001	COMPLEX_TIMER1_OUT
00010	PWM0_OUT
00011	PWM1_OUT
00100	PWM2_OUT
00101	PWM3_OUT
00110	PWM4_OUT
00111	LIN_OUT_SOFT
01000	IO_IN[0]

01001	IO_IN[1]
01010	IO_IN[2]
01011	IO_IN[3]
01100	PPM_OUT
01101	IO_IN[4]
01110	IO_IN[5]
01111	IO_IN[6]
10000	IO_IN[7]
10001	IO_IN[8]
10010	IO_IN[9]
10011	IO_IN[10]
10100	IO_IN[11]
10101	UART0_TX
10110	UART1_TX
10111-11111	'1'

Table 91 – LIN_XTX input seletion

14.2.8.3.6. Digital IO as UART input

The input multiplexer for both UART0 and UART1, function is controlled by port PORT_COMM_CFG1:UART0_RX_SEL[3:0] and PORT_COMM_CFG1:UART1_RX_SEL[7:4] according to the description in Table 92.

UART{0,1}_RX_ SEL[3:0]	Select input for UART{0,1} RX
0000	IO_IN[0]
0001	IO_IN[1]
0010	IO_IN[2]
0011	IO_IN[3]
0100	IO_IN[4]
0101	IO_IN[5]
0110	IO_IN[6]
0111	IO_IN[7]
1000	IO_IN[8]

1001	IO_IN[9]
1010	IO_IN[10]
1011	IO_IN[11]
1100-1111	'0'

Table 92 – UART RX input selection

14.2.8.3.7. Digital IO input portsmap

The digital IO input configuration is controlled by PORT_IO_IN, PORT_IO_TRIG_EDGE_CFG, PORT_IO_TRIG_EDGE_CFG1, PORT_LIN_XTX_CFG, PORT_TIMER_CFG, PORT_TIMER_CFG1, PORT_COMM_CFG, PORT_COMM_CFG1, PORT_I2C_CFG and PORT_PPM_CTRL.

Port: PORT_IO_IN

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
-	[15:12]		Not used
IO_IN_SYNC	[11:0]	R	From Schmitt trigger I/O (after resynchronisation)

Table 93 – PORT_IO_IN

Port: PORT_IO_TRIG_EDGE_CFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
IO7_TRIG_EDGE_SEL	[15:14]	RW	Trigger edge selection for IO7
IO6_TRIG_EDGE_SEL	[13:12]	RW	Trigger edge selection for IO6
IO5_TRIG_EDGE_SEL	[11:10]	RW	Trigger edge selection for IO5
IO4_TRIG_EDGE_SEL	[9:8]	RW	Trigger edge selection for IO4
IO3_TRIG_EDGE_SEL	[7:6]	RW	Trigger edge selection for IO3
IO2_TRIG_EDGE_SEL	[5:4]	RW	Trigger edge selection for IO2
IO1_TRIG_EDGE_SEL	[3:2]	RW	Trigger edge selection for IO1
IO0_TRIG_EDGE_SEL	[1:0]	RW	Trigger edge selection for IO0

Table 94 – PORT_IO_TRIG_EDGE_CFG

Port: PORT_IO_TRIG_EDGE_CFG1

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:8]		Not used
IO11_TRIG_EDGE_SEL	[7:6]	RW	Trigger edge selection for IO11
IO10_TRIG_EDGE_SEL	[5:4]	RW	Trigger edge selection for IO10
IO9_TRIG_EDGE_SEL	[3:2]	RW	Trigger edge selection for IO9
IO8_TRIG_EDGE_SEL	[1:0]	RW	Trigger edge selection for IO8

Table 95 – PORT_IO_TRIG_EDGE_CFG1

Port: PORT_LIN_XTX_CFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:9]		Not used
LIN_IN_SOFT	8	RW	Software port for LIN reception (connected via mux to LIN_XTX)
LIN_OUT_SOFT	7	RW	Software port for LIN transmission (connected via mux to LIN_XTX)
LIN_COLIN_RX_SEL	[6:5]	RW	Input selection for Colin RX input
LIN_XTX_OUT_SEL	[4:0]	RW	Output selection for LIN_XTX

Table 96 – PORT_LIN_XTX_CFG

Port: PORT_TIMER_CFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:13]		Not used
TIMER0_CHB_SEL	[12:8]	RW	Input selection for TIMER0 channel B
	[7:5]		Not used, gap to align on byte
TIMER0_CHA_SEL	[4:0]	RW	Input selection for TIMER0 channel A

Table 97 – PORT_TIMER_CFG

Port: PORT_TIMER_CFG1

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:13]		Not used
TIMER1_CHB_SEL	[12:8]	RW	Input selection for TIMER1 channel B
-	[7:5]		Not used
TIMER1_CHA_SEL	[4:0]	RW	Input selection for TIMER1 channel A

Table 98 – PORT_TIMER_CFG1

Port: PORT_COMM_CFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
SPI_SS_IN_SEL	[15:12]	RW	IO selection for SPI_SS_IN
SPI_SCK_IN_SEL	[11:8]	RW	IO selection for SPI_SCK_IN
SPI_MISO_IN_SEL	[7:4]	RW	IO selection for SPI_MISO_IN
SPI_MOSI_IN_SEL	[3:0]	RW	IO selection for SPI_MOSI_IN

Table 99 – PORT_COMM_CFG

Port: PORT_COMM_CFG1

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:8]		Not used
UART1_RX_SEL	[7:4]	RW	IO selection for UART1 RX
UART0_RX_SEL	[3:0]	RW	IO selection for UART0 RX

Table 100 – PORT_COMM_CFG1

Port: PORT_I2C_CONF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:12]		Not used
I2C_SCL_CLK_SEL	[11:9]	RW	Select IO[x] or LIN_RXD input for SCL signal, see Table 88
	[8:0]		See paragraph 14.3.11 for full port description

Table 101 – PORT_I²C_CONF

Port: PORT_PPM_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	15		Not used
PPM_IN_SEL	[14:11]	RW	Select IO[x] or LIN_XRX input for PPM interface, see Table 89
	[10:0]		See paragraph 0 for full port description

Table 102 – PORT_PPM_CTRL

14.3. Digital

14.3.1. Common Melexis MCU with MLX16-FX CPU

For more information regarding the MLX16-FX (CPU architecture, instruction set, register set, addressing modes etc.) please refer to the documentation given in chapter 6.

The MLX16-FX core has a co-processor for fast multiplication and division of 32-bit and 16-bit values and uses a 5 stages pipeline (Fetch, Decode, Address, Memory and Execute). Further information can be found in [3], chapter 7.

14.3.1.1. CPU Architecture

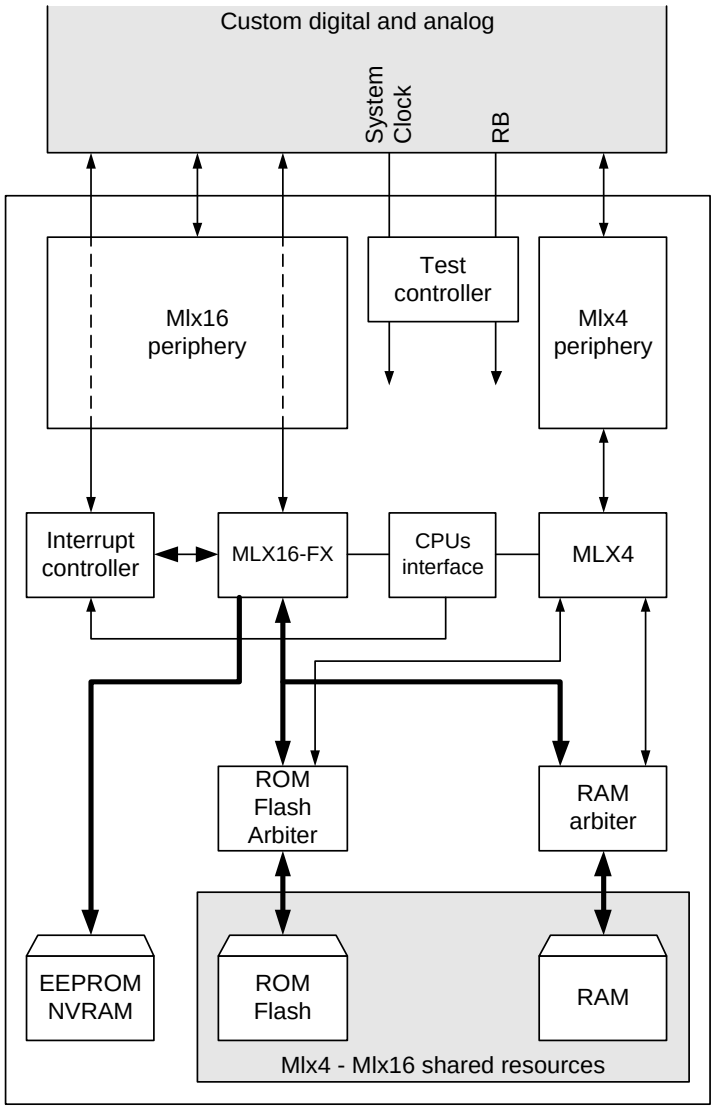


Figure 43 – CPU architecture

14.3.1.2. Memory mapping

MLX81346 contains a unified 20-bit bus for accessing devices as shown in Figure 44

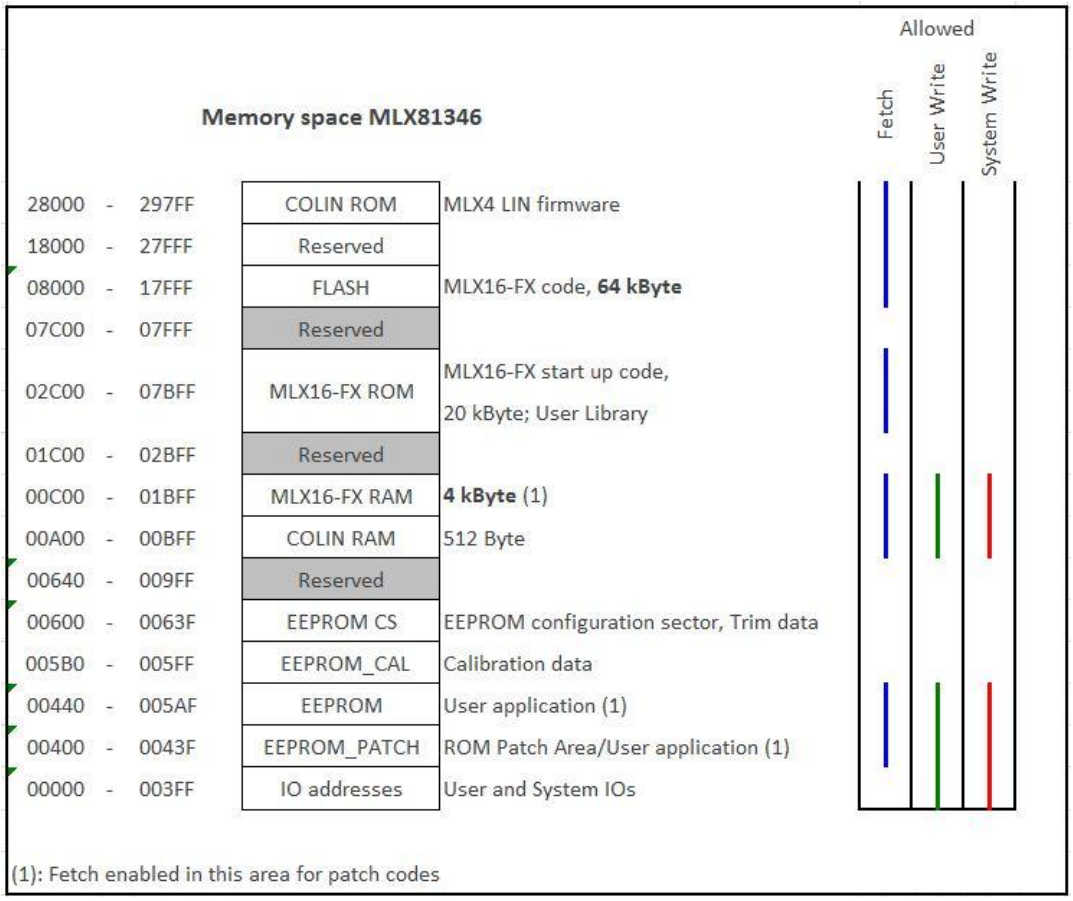


Figure 44 – Memory mapping MLX81346

The predefined pages of the MLX16-FX are encoded as given below.

Name	Address	Note
Fp0:	0x02C00	First ROM page (used by interrupt controller)
Fp1:	0x02D00	Can be used for C runtime
Fp2:	0x02E00	Can be used for C runtime
Fp3:	0x02F00	Can be used for C runtime
Fp4:	0x03000	Can be used for C runtime
Fp5:	0x03100	Can be used for C runtime
Fp6:	0x03200	Can be used for C runtime
Fp7:	0x00400	EEPROM, to allow single instruction patch start
Dp:	0x00C00	MLX16-FX RAM
Io:	0x00000	User Ports
	Legend:	**Fp: Far page; Dp: Data page; Io: IO page**

Table 103 – MLX16-FX predefined pages

14.3.2. Watchdog system

14.3.2.1. Absolute watchdog (AWD)

The absolute watchdog is automatically started at power on reset. It is a resettable, free-running counter with several compare units. It operates on clock AWD_CLK which is based on the 1MHz RC oscillator clock (FRC_1M). The timeout comparator value is fixed to 250000 which sets a nominal 'TIMEOUT' value of 250ms.

A block diagram is shown in Figure 45 and a timing diagram in Figure 46.

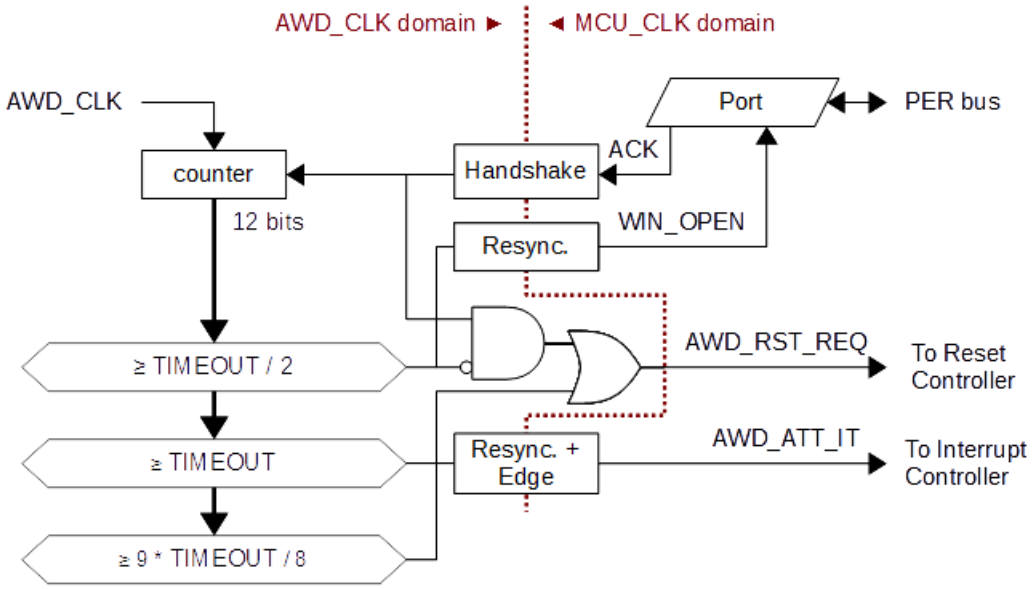


Figure 45 – Absolute watchdog block diagram

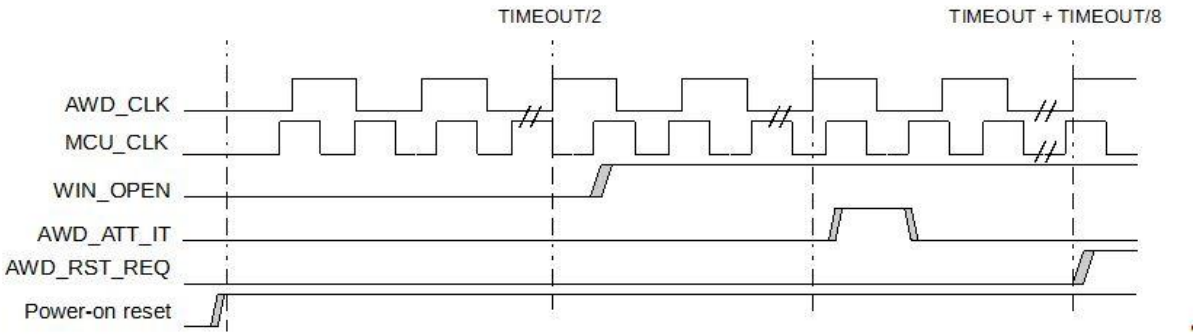


Figure 46 – Absolute watchdog timing diagram

14.3.2.1.1. Functional Description

The absolute watchdog is automatically started at power on reset. It is a resettable, free-running counter counting the incoming clock pulses of AWD_CLK (FRC_1M). The counter is reset by software acknowledge or by a counter timeout in case of no reset by acknowledge (AWD_RST_REQ) is performed within the specified time.

When the counter reaches TIMEOUT/2, then WIN_OPEN is set and synchronized to the MCU clock (MCU_CLK). The WIN_OPEN bit indicates if the window to acknowledge the watchdog is open. The field is not accessible by a word read.

When the counter reaches TIMEOUT, an attention interrupt (AWD_ATT_IT) is generated. An extra time of TIMEOUT/8 is added to allow application software to acknowledge the watchdog. After this delay, a reset is performed if the watchdog has not been acknowledged. The software must therefore acknowledge the watchdog within a time frame starting at TIMEOUT/2 and ending at 9xTIMEOUT/8. If the watchdog is acknowledged outside this time window (i.e. before TIMEOUT/2), a reset is generated.

Note: Acknowledge can only be performed by writing '1' to bit ACK. Any other write on the corresponding port by byte or word access generates a memory error interrupt (MLX16_MEMERR). Due to the resynchronization of ACK from AWD_CLK to the MCU_CLK clock domain, only the first acknowledge within the same AWD_CLK clock period is taken into account while following acknowledges are ignored.

Port: AWD

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Byte, Bit
Field name	**Bit**	**R/W**	**Description**
WIN_OPEN	15	R	1: Window is open, software can acknowledge the watchdog 0: Window is closed
		W	No effect
-	[14:7]		Not used
ACK	6	R	Always read 0
		W	1: Acknowledge the watchdog if window is open, else generate a CPU reset 0: No effect
-	[5:0]		Not used

Table 104 – AWD (Absolute Watchdog)

14.3.2.2. Intelligent watchdog (IWD)

The intelligent watchdog is started by application software. It contains a programmable and resettable free-running counter (WDC) which operates based on clock WDC_CLK. This clock is derived from the main MCU clock MCU_CLK and 3-bit predivider DIV. A block diagram is shown in Figure 47.

$$WDC\_CLK = \frac{MCU\_CLK}{2^{(5+2*DIV)}}$$

The application software should allocate a TIMEOUT delay and tag WTG for a certain task. WTG can be used by the software developer e.g. as a software marker or as a task ID. WTG can be of any value and is not used by the watchdog itself. See paragraph 14.3.2.2.1 for more information about the usage of WTG.

The watchdog TIMEOUT delay is defined by the compare value in WDT and predivider DIV. DIV is used to determine the timeout counter clock as shown in the above formula. It cannot be written as a byte, but only as a word, i.e. WDT and DIV at the same time. Writing DIV as a byte has no effect.

The combination of WDT and DIV generate a TIMEOUT delay equal to:

$$IWD_{Timeout} = WDT * \frac{1}{WDC\_CLK}$$

WDT can be written as byte alone or, together with DIV, as word.

It is possible to enable a windowing mechanism by setting port bit WIN_ENABLE. The mechanism can be disabled by setting port bit WIN_DISABLE. If the windowing mechanism is disabled, the software can acknowledge the watchdog at any time. If the windowing mechanism is enabled, the application software can only acknowledge the watchdog when the window is opened, i.e. from TIMEOUT/2 to 9xTIMEOUT/8. When windowing is enabled, any acknowledge outside the open window (i.e. before TIMEOUT/2) causes a reset interrupt (IWD_RST_REQ).

Once the TIMEOUT value has elapsed, the watchdog generates an attention interrupt (IWD_ATT_IT). It can be considered as a warning that a reset interrupt will occur if the acknowledge is not performed soon. A new waiting sequence of 1/8 of the TIMEOUT value is started after this interrupt. If the application software has not acknowledged the watchdog by the end of this sequence, a reset is finally generated.

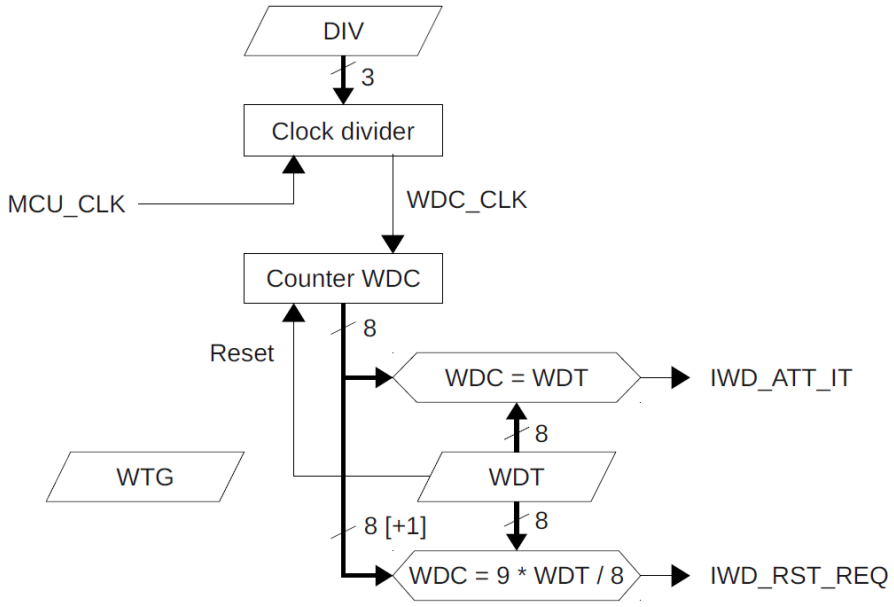


Figure 47 – Intelligent watchdog block diagram

14.3.2.2.1. Acknowledge

To start the watchdog, the application software must write a tag (WTG) and timeout value 'TIMEOUT' (WDT and optionally DIV). In case a 'TIMEOUT' value is written prior to the tag, the watchdog is not started.

To acknowledge the watchdog, the software must load a tag and a timeout value. The tag has to be written first directly followed by a write of the timeout value (WDT and optionally DIV). Acknowledge is effective when WDT is updated. It is also possible to update both WDT and DIV at the same time. On a valid acknowledge, the counter WDC is reset and restarts counting.

14.3.2.2.2. Reset

There are 3 reasons for getting a reset:

- No acknowledge at WDC = 9x WDT/8
- Acknowledge outside the window when window mechanism is enabled
- Update of WDT (or WDT and DIV) without update to WTG before

If a reset occurs, a flag located in PORT_RST_CTRL:IWD_WBOOT allows to detect this situation by the application software.

14.3.2.2.3. Timing without acknowledge

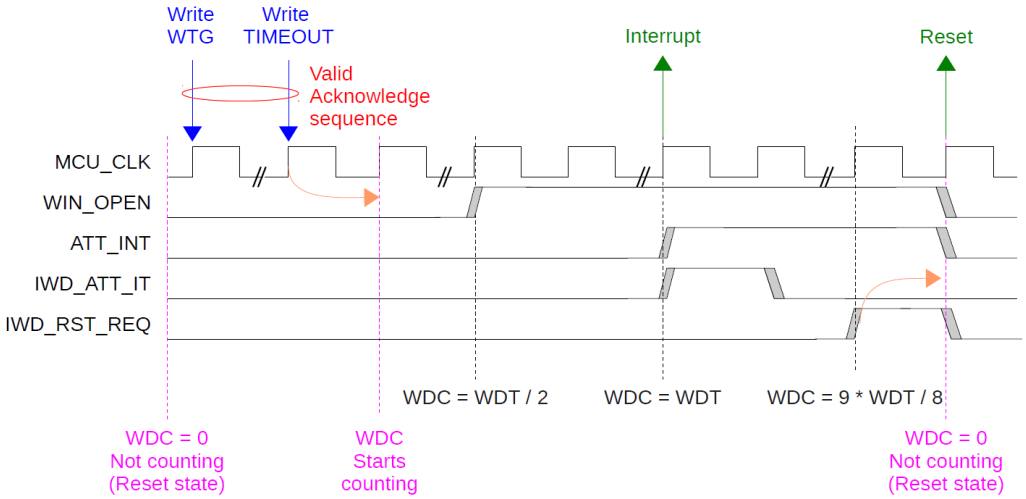


Figure 48 Intelligent watchdog timing diagram without acknowledge

At power-on reset the watchdog is inactive. It becomes active on a write to WTG followed by a write to 'TIMEOUT' (or only WDT). When WDC=WDT/2, signal WIN_OPEN is set to '1' to inform the application software that the window is opened. At WDC=WDT, the attention interrupt (IWD_ATT_IT) is generated, it lasts one MCU_CLK clock period. At the same time signal PORT_IWD1:ATT_INT is set to '1' to inform the application software that the interrupt signal is generated and the extra phase is active (WDT < WDC < 9*WDT/8). When WDC=9*WDT/8, the reset request signal (IWD_RST_REQ) is generated, it lasts one MCU_CLK clock period. The watchdog itself is reset and remains inactive until it is enabled again by the application software.

14.3.2.2.4. Timing with a correct acknowledge

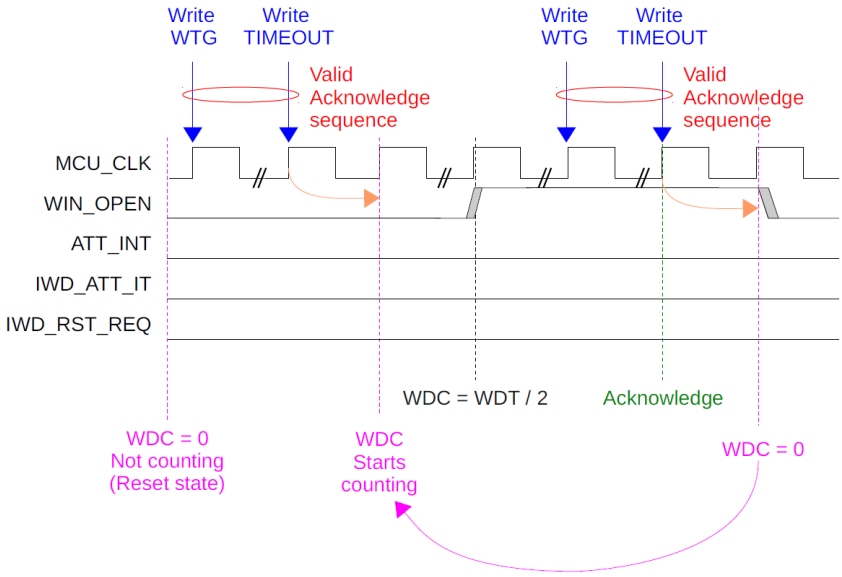


Figure 49 Intelligent watchdog timing diagram with a correct acknowledge

At power-on reset the watchdog is inactive. It becomes active on a write to WTG followed by a write to 'TIMEOUT' (WDT and/or DIV). WIN_ENABLE is written at the same time as WTG. When WDC = WDT/2, signal WIN_OPEN is set to '1' to inform the application software that the window is opened. The software first writes WTG and then writes 'TIMEOUT' (WDT and/or DIV). This acknowledges the watchdog, and thus its counter is reset (WDC = 0) and restarts counting.

If the windowing mechanism is not activated, the acknowledge sequence can be executed independent of the status of the window.

14.3.2.2.5. Timing with an incorrect acknowledge

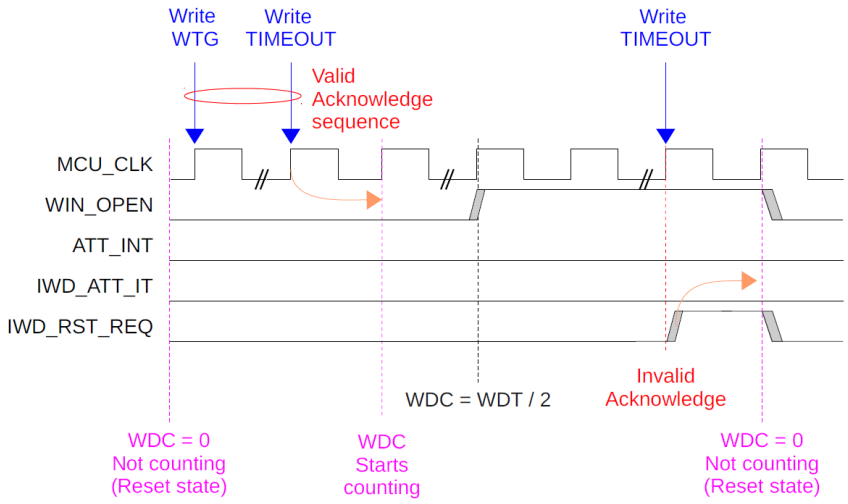


Figure 50 Intelligent watchdog timing diagram with an incorrect acknowledge

In the example of Figure 50, the application software writes 'TIMEOUT' (WDT and/or DIV), without writing into WTG previously. It is an incorrect acknowledge, so the watchdog generates a reset.

The configuration of the intelligent watchdog IWD is controlled by block port IWD.

Block: IWD

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:11]		Not used
DIV	[10:8]	RW	Counter clock: WDC_CLK = MCU_CLK / $2^{(5+2*DIV)}$
WDT	[7:0]	RW	Timeout

Address	Reset		Access

See Table 106 – MLX81346 ports overview	0x2000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
WIN_OPEN	15	R	Window state: 0: Window is closed 1: Window is opened
		W	no effect
ATT_INT	14	R	1 = Attention interrupt has been generated 0 = Attention interrupt has not been generated
		W	no effect
WIN_DISABLE	13	W	1 = Disable Window. Acknowledges can occur at any time 0 = No effect
		R	1: Windowing disabled 0: Windowing enabled
WIN_ENABLE	12	W	1 = Enable Window. Acknowledges have to occur when the window is open 0 = No effect
		R	1: Windowing enabled 0: Windowing disabled
-	[11:8]		Not used
WTG	[7:0]	RW	Watchdog tag register

Table 105 – IWD (Intelligent Watchdog)

14.3.3. IO Registers

14.3.3.1. General

For the MLX16-FX two different types of ports exist:

- System protected ports, MLX16-FX bit USER must be cleared
- User ports, not protected

Writing to non-existing custom ports will have no effect and reading from non-existent custom ports will always return 0.

Bits in the port description which are not mentioned or described in following chapters are reserved for Melexis.

14.3.3.2. System Protected and Standard ports

Table 106 below shows the available system and standard ports. The system ports are protected. Thus, accessing them requires a reset of bit USER to '0' (M register of MLX16-FX).

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x002AB		
PORT_ACTIVE_CDI_IN	User	0x002AA	Word/Read only	CDI status register
		0x002A9		
PORT_ACTIVE_CDI	User	0x002A8	Word/Byte	CDI control port
		0x002A7		
TRIM_CP	User	0x002A6	Word/Byte	Trim port for charge pump
		0x002A5		
PORT_CP	User	0x002A4	Word/Byte	Charge pump control port
		0x002A3		
PORT_CURR_SENS	User	0x002A2	Word/Byte	Current sensor control port
		0x002A1		
PORT_UDMA1_STATUS	User	0x002A0	Word/Read only	UART1 DMA status register
		0x0029F		
UART1	User	0x0029E	Word/Read only	UART1 status register
		0x0029D		
UART1	User	0x0029C	Word	UART1 control and status register
		0x0029B		
UART1	User	0x0029A	Word	UART1 frame duration/bit counter data
		0x00299		
UART1	User	0x00298	Word/Read only	UART1 receive data register
		0x00297		
UART1	User	0x00296	Word	UART1 transmit data register
		0x00295		
UART1	User	0x00294	Word	UART1 baud rate register
		0x00293		
PORT_UDMA1_CTRL	User	0x00292	Word/Byte	UART1 DMA control register
		0x00291		
PORT_UDMA1_SIZE	User	0x00290	Word/Byte	UART1 DMA buffer size
		0x0028F		
PORT_UDMA1_TX	User	0x0028E	Word/Byte	UART1 DMA transmit register
		0x0028D		
PORT_UDMA1_RDB	User	0x0028C	Word/Byte	UART1 DMA receive register
		0x0028B		
PORT_UDMA1_RDA	User	0x0028A	Word/Byte	UART1 DMA receive register
		0x00289		
PORT_UDMA0_STATUS	User	0x00288	Word/Read only	UART0 DMA status register
		0x00287		

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
UART0	User	0x00286	Word/Read only	UART0 status register
		0x00285		
UART0	User	0x00284	Word	UART0 control and status register
		0x00283		
UART0	User	0x00282	Word	UART0 frame duration/bit counter data
		0x00281		
UART0	User	0x00280	Word/Read only	UART0 receive data register
		0x0027F		
UART0	User	0x0027E	Word	UART0 transmit data register
		0x0027D		
UART0	User	0x0027C	Word	UART0 baud rate register
		0x0027B		
PORT_UDMA0_CTRL	User	0x0027A	Word/Byte	UART0 DMA control register
		0x00279		
PORT_UDMA0_SIZE	User	0x00278	Word/Byte	UART0 DMA buffer size
		0x00277		
PORT_UDMA0_TX	User	0x00276	Word/Byte	UART0 DMA transmit register
		0x00275		
PORT_UDMA0_RDB	User	0x00274	Word/Byte	UART0 DMA receive register
		0x00273		
PORT_UDMA0_RDA	User	0x00272	Word/Byte	UART0 DMA receive register
		0x00271		
PORT_TX_TIMEOUT	User	0x00270	Word/Read only	Timeout value in TX mode for reading
		0x0026F		
PORT_PPM_FLAGS	User	0x0026E	Word/Read only	PPM flag status for reading
		0x0026D		
PPM_TIMER	User	0x0026C	Word/Byte	PPM on time for TX mode
		0x0026B		
PORT_PPM_BUF_DATA	User	0x0026A	Word/Read only	PPM data register for reading
		0x00269		
PORT_PPM_CTRL	User	0x00268	Word/Byte	PPM control register
		0x00267		
PORT_PPM_TIMEOUT	User	0x00266	Word/Byte	PPM timeout register
		0x00265		
PORT_PPM_TBASE_ADD	User	0x00264	Word/Byte	PPM base address for DMA in TX mode
		0x00263		
PORT_PPM_RBASE_ADD	User	0x00262	Word/Byte	PPM base address for DMA in RX mode
		0x00261		
PORT_MISC2_TEST	User	0x00260	Word/Byte	Melexis Test

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x0025F		
PORT_MISC_ TEST	User	0x0025E	Word/Byte	Melexis Test
		0x0025D		
PORT_I2C_ READ_OFFSET	User	0x0025C	Word/Byte	I^2C interface
		0x0025B		
PORT_I2C_ DMA_OFFSET	User	0x0025A	Word/Byte	I^2C interface
		0x00259		
PORT_I2C_ CONF	User	0x00258	Word/Byte	I^2C interface
		0x00257		
PORT_CDI_IN	User	0x00256	Word/Read only	CDI bits of motor phases
		0x00255		
TRIM_IILD	User	0x00254	Word/Byte	Calibration value for internal predriver ILD
		0x00253		
PORT_CNT_ EILD_HSOFF	User	0x00252	Word/Byte	External ILD for HSx outputs
		0x00251		
PORT_CNT_ EILD_LSOFF	User	0x00250	Word/Byte	External ILD for LSx outputs
		0x0024F		
PORT_DIV_EILD	User	0x0024E	Word	Diagnostics handling
		0x0024D		
PORT_CNT_ MASK_VDS_HS	User	0x0024C	Word	Diagnostics handling
		0x0024B		
PORT_CNT_ MASK_VDS_LS	User	0x0024A	Word/Byte	Motor driver control
		0x00249		
PORT_DIV_ MASK_VDS	User	0x00248	Word/Byte	Motor driver control
		0x00247		
PORT_CNT_ VDS_DEB	User	0x00246	Word/Byte	Motor driver control
		0x00245		
PORT_DIV_ VDS_DEB	User	0x00244	Word/Byte	Melexis Test
		0x00243		
PORT_DIAG_IN	User	0x00242	Word/Read only	Melexis Test
		0x00241		
PORT_DRV_IN	User	0x00240	Word/Read only	Melexis Test
		0x0023F		
PORT_DRV3_ PROT	User	0x0023E	Word/Byte	Motor driver control
		0x0023D		
PORT_DRV2_ PROT	User	0x0023C	Word/Byte	Motor driver control
		0x0023B		
PORT_DRV1_ PROT	User	0x0023A	Word/Byte	Motor driver control

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x00239		
PORT_DRV_ CTRL	User	0x00238	Word/Byte	Motor driver control
		0x00237		
TRIM_MISC	User	0x00236	Word/Byte	Calibration register
		0x00235		
PORT_CLOCK_ CTRL	System	0x00234	Word/Byte	Enable/Disable 16MHz system clock
		0x00233		
PORT_LIN_ XCFG_VALID	User	0x00232	Word/Read only	Digital read of PORT_LIN_XCFG
		0x00231		
PORT_LIN_ XCFG	System	0x00230	Word/Byte	LIN external mode config register
		0x0022F		
PORT_DIS_ GTSM	User	0x0022E	Word/Byte	Enable/Disable Go to SLEEP mode (GTSM)
		0x0022D		
PORT_ STOPMD_CFG	User	0x0022C	Word/Byte	Configure CPU STOP mode
		0x0022B		
PORT_STOPMD_ CTRL	System	0x0022A	Word/Byte	Enable/Dis. CPU STOP mode (not SLEEP)
		0x00229		
PORT_MISC2_ OUT	User	0x00228	Word/Byte	Ctrl register for IO wake-up, OTD, VSM filt.
		0x00227		
PORT_MISC_ OUT	User	0x00226	Word/Byte	Ctrl register for under- / over-voltage det.
		0x00225		
PORT_COMM_ CFG1	User	0x00224	Word/Byte	IO input selection for SPI interface
		0x00223		
PORT_COMM_ CFG	User	0x00222	Word/Byte	IO input selection for SPI interface
		0x00221		
PORT_TIMER_ CFG1	User	0x00220	Word/Byte	IO input selection for Complex TIMER1
		0x0021F		
PORT_TIMER_ CFG	User	0x0021E	Word/Byte	IO input selection for Complex TIMER0
		0x0021D		
PORT_LIN_XTX_ CFG	User	0x0021C	Word/Byte	IO input/output selection for LIN RX/TX
		0x0021B		
PORT_IO_TRIG_ EDGE_CFG1	User	0x0021A	Word/Byte	IO trigger edge selection
		0x00219		
PORT_IO_TRIG_ EDGE_CFG	User	0x00218	Word/Byte	IO trigger edge selection
		0x00217		
PORT_IO_	User	0x00216	Word/Byte	IO output selection for UARTs

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
UART_SEL				
		0x00215		
PORT_IO_CFG2	User	0x00214	Word/Byte	IO output selection
		0x00213		
PORT_IO_CFG1	User	0x00212	Word/Byte	IO output selection
		0x00211		
PORT_IO_CFG0	User	0x00210	Word/Byte	IO output selection
		0x0020F		
PORT_IO_ ENABLE1	User	0x0020E	Word/Byte	IO output selection
		0x0020D		
PORT_IO_ ENABLE	User	0x0020C	Word/Byte	IO output selection
		0x0020B		
PORT_IO_OUT_ EN	User	0x0020A	Word/Byte	IO control register
		0x00209		
PORT_IO_OUT_ SOFT	User	0x00208	Word/Byte	IO output port
		0x00207		
PORT_MASK_ OV_HS_IO	User	0x00206	Word/Byte	High side over-voltage settings
		0x00205		
PORT_OC_DEB	User	0x00204	Word/Byte	Overcurrent debounce settings
		0x00203		
PORT_SUPP_ CFG	User	0x00202	Word/Byte	Filter for under-/over-voltage detect. flags
		0x00201		
PORT_MISC_IN	User	0x00200	Word	Different Status Flags
		0x001FF		
PORT_SUPP2_ IN	User	0x001FE	Word/Read only	Diagnostics handling
		0x001FD		
PORT_SUPP_IN	User	0x001FC	Word/Read only	Diagnostics handling
		0x001FB		
PORT_IO_IN	User	0x001FA	Word/Byte	IO input port
		0x001F9		
PORT_SPARE_ TEST	User	0x001F8	Word/Byte	Melexis Test
		0x001F7		
PORT_LIN_TEST	User	0x001F6	Word/Byte	Melexis Test
		0x001F5		
PORT_SUPP2_ TEST	User	0x001F4	Word/Byte	Melexis Test
		0x001F3		
PORT_SUPP_ TEST	User	0x001F2	Word/Byte	Melexis Test
		0x001F1		
PORT_STEP_ CONF	User	0x001F0	Word/Byte	SSCM – Ctrl Port Spread Spectrum Mod.
		0x001EF		
PORT_SSCM_	User	0x001EE	Word/Byte	SSCM – Crtl Port Spread Spectrum Mod.

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
CONF				
		0x001ED		
TRIM_RCO1M	User	0x001EC	Word/Byte	Trim port for 1MHz RCO
		0x001EB		
TRIM_RCO32M	User	0x001EA	Word/Byte	Trim port for 32MHz RCO
		0x001E9		
TRIM_VDD	User	0x001E8	Word/Byte	Trim port for VDDA and VDDD regulators
		0x001E7		
TRIM_BG_BIAS	User	0x001E6	Word/Byte	Trim port for bandgaps, biasing current
		0x001E5		
COLIN	User	0x001E4	Word/Byte	MLX4 RAM area protection
		0x001E3		
COLIN	User	0x001E2	Word/Byte	Melexis Test
		0x001E1		
COLIN	User	0x001E0	Word/Byte	MLX4 Memory Timings
		0x001DF		
COLIN	User	0x001DE	Byte	MLX4 Protocol Handler Port
		0x001DD		
COLIN	User	0x001DC	Byte	MLX4 Clock Settings
		0x001DB		
COLIN	User	0x001DA	Word	MLX4 Patch Data 3
		0x001D9		
COLIN	User	0x001D8	Word	MLX4 Patch Control 3
		0x001D7		
COLIN	User	0x001D6	Word	MLX4 Patch Data 2
		0x001D5		
COLIN	User	0x001D4	Word	MLX4 Patch Control 2
		0x001D3		
COLIN	User	0x001D2	Word	MLX4 Patch Data 1
		0x001D1		
COLIN	User	0x001D0	Word	MLX4 Patch Control 1
		0x001CF		
COLIN	User	0x001CE	Word	MLX4 Patch Data 0
		0x001CD		
COLIN	User	0x001CC	Word	MLX4 Patch Control 0
		0x001CB		
EEPROM_FLASH	Test/User	0x001CA	Word	EEPROM – Test
		0x001C9		
EEPROM_FLASH	Test/User	0x001C8	Word/Byte	EEPROM – Test
		0x001C7		
EEPROM_FLASH	Test/User	0x001C6	Word/Byte	EEPROM – Test
		0x001C5		
EEPROM_FLASH	Test/User	0x001C4	Word/Byte	EEPROM – Test
		0x001C3		
EEPROM_FLASH	Test/User	0x001C2	Word	FLASH – Test
		0x001C1		
EEPROM_FLASH	Test/Use	0x001C0	Word	FLASH – Test

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
	r			
		0x001BF		
EEPROM_FLASH	Test/User	0x001BE	Word/Byte	FLASH – Test
		0x001BD		
EEPROM_FLASH	Test/User	0x001BC	Word/Byte	FLASH – Test
		0x001B3		
EEPROM_FLASH	User	0x001B2	Word/Byte	EEPROM – Control
		0x001B1		
EEPROM_FLASH	User	0x001B0	Word/Byte	EEPROM – Control
		0x001AF		
EEPROM_FLASH	User	0x001AE	Word/Byte	EEPROM – Control
		0x001AD		
EEPROM_FLASH	User	0x001AC	Word/Byte	EEPROM – Control
		0x001AB		
EEPROM_FLASH	User	0x001AA	Word/Byte	EEPROM – Control
		0x001A9		
EEPROM_FLASH	User	0x001A8	Word/Byte	EEPROM – Control
		0x001A7		
EEPROM_FLASH	System	0x001A6	Word/Byte	EEPROM – Control
		0x001A5		
EEPROM_FLASH	User	0x001A4	Word	FLASH – Control
		0x001A3		
EEPROM_FLASH	User	0x001A2	Word/Byte	FLASH – Control
		0x001A1		
EEPROM_FLASH	User	0x001A0	Word/Byte	FLASH – Control
		0x0019F		
EEPROM_FLASH	User	0x0019E	Word/Byte	FLASH – Control
		0x0019D		
EEPROM_FLASH	System	0x0019C	Word/Byte	EEPROM – Status Port
		0x0019B		
EEPROM_FLASH	Test/User	0x0019A	Word/Byte	EEPROM – Test Port
		0x00197		
EEPROM_FLASH	User	0x00196	Word/Byte	FLASH – Control
		0x00177		
ADC_BLOCK	User	0x00176	Word/Byte	ADC – Correction vector
		0x00175		
ADC_BLOCK	User	0x00174	Word/Byte	ADC – Correction vector
		0x00173		
ADC_BLOCK	User	0x00172	Word/Byte	ADC – Correction vector
		0x00171		
ADC_BLOCK	User	0x00170	Word/Byte	Melexis Test
		0x0016F		
ADC_BLOCK	User	0x0016E	Word/Byte	Melexis Test
		0x0016D		
ADC_BLOCK	User	0x0016C	Word/Byte	ADC – Clock Divider
		0x0016B		
ADC_BLOCK	User	0x0016A	Word/Byte	ADC – Control and Status Port
		0x00169		
ADC_BLOCK	User	0x00168	Word/Byte	ADC – Configuration Start Address

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x00167		
ADC_BLOCK	User	0x00166	Word/Byte	ADC – Control and Status Port
		0x00165		
PWM_MASTER2	User	0x00164	Word/Byte	PWM Master2 – Control Port
		0x00163		
PWM_MASTER2	User	0x00162	Word/Byte	PWM Master2 – PWM Period
		0x00161		
PWM_MASTER2	User	0x00160	Word/Byte	PWM Master2 – Low Threshold Value
		0x0015F		
PWM_MASTER2	User	0x0015E	Word/Byte	PWM Master2 – High Threshold Value
		0x0015D		
PWM_MASTER2	User	0x0015C	Word/Byte	PWM Master2 – Interrupt Comp. register
		0x0015B		
PWM_SLAVE3	User	0x0015A	Word/Byte	PWM Slave3 – Control Port
		0x00159		
PWM_SLAVE3	User	0x00158	Word/Byte	PWM Slave3 – PWM Period
		0x00157		
PWM_SLAVE3	User	0x00156	Word/Byte	PWM Slave3 – Low Threshold Value
		0x00155		
PWM_SLAVE3	User	0x00154	Word/Byte	PWM Slave3 – High Threshold Value
		0x00153		
PWM_SLAVE3	User	0x00152	Word/Byte	PWM Slave3 – Interrupt Comp. register
		0x00151		
PWM_SLAVE2	User	0x00150	Word/Byte	PWM Slave2 – Control Port
		0x0014F		
PWM_SLAVE2	User	0x0014E	Word/Byte	PWM Slave2 – PWM Period
		0x0014D		
PWM_SLAVE2	User	0x0014C	Word/Byte	PWM Slave2 – Low Threshold Value
		0x0014B		
PWM_SLAVE2	User	0x0014A	Word/Byte	PWM Slave2 – High Threshold Value
		0x00149		
PWM_SLAVE2	User	0x00148	Word/Byte	PWM Slave2 – Interrupt Comp. register
		0x00147		
PWM_SLAVE1	User	0x00146	Word/Byte	PWM Slave1 – Control Port
		0x00145		
PWM_SLAVE1	User	0x00144	Word/Byte	PWM Slave1 – PWM Period
		0x00143		
PWM_SLAVE1	User	0x00142	Word/Byte	PWM Slave1 – Low Threshold Value
		0x00141		
PWM_SLAVE1	User	0x00140	Word/Byte	PWM Slave1 – High Threshold Value
		0x0013F		
PWM_SLAVE1	User	0x0013E	Word/Byte	PWM Slave1 – Interrupt Comp. register
		0x0013D		
PWM_MASTER1	User	0x0013C	Word/Byte	PWM Master1 – Control Port
		0x0013B		
PWM_MASTER1	User	0x0013A	Word/Byte	PWM Master1 – PWM Period
		0x00139		
PWM_MASTER1	User	0x00138	Word/Byte	PWM Master1 – Low Threshold Value
		0x00137		
PWM_MASTER1	User	0x00136	Word/Byte	PWM Master1 – High Threshold Value
		0x00135		
PWM_MASTER1	User	0x00134	Word/Byte	PWM Master1 – Interrupt Comp. register

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x00133		
SPI_CTRL	User	0x00132	Word/Byte	SPI interface- control signals
		0x00131		
SPI_MASTER_BAUD_RATE	User	0x00130	Word/Byte	SPI interface – SPI master baud rate config
		0x0012F		
SPI_DMA_CTRL	User	0x0012E	Word/Byte	SPI interface – DMA control
		0x0012D		
SPI_RBASE	User	0x0012C	Word/Byte	SPI interface - Base address of received data
		0x0012B		
SPI_SBASE	User	0x0012A	Word/Byte	SPI interface - Base address of data to send
		0x00129		
SPI_DATA	User	0x00128	Word/Byte	SPI interface - Data port
		0x00127		
CTIMER1	User	0x00126	Word/Byte	Complex TIMER1 – Control / Status
		0x00125		
CTIMER1	User	0x00124	Word/Read only	Complex TIMER1 – Current Counter Value
		0x00123		
CTIMER1	User	0x00122	Word	Complex TIMER1 – Time Counter ChannelA
		0x00121		
CTIMER1	User	0x00120	Word	Complex TIMER1 – Time Counter ChannelB
		0x0011F		
CTIMER0	User	0x0011E	Word/Byte	Complex TIMER0 – Control / Status
		0x0011D		
CTIMER0	User	0x0011C	Word/Read only	Complex TIMER0 – Current Counter Value
		0x0011B		
CTIMER0	User	0x0011A	Word	Complex TIMER0 – Time Counter ChannelA
		0x00119		
CTIMER0	User	0x00118	Word	Complex TIMER0 – Time Counter ChannelB
		0x00117		
RAM_BIST	User	0x00116	Word/Byte	Melexis Test
		0x00115		
RAM_BIST	User	0x00114	Word/Byte	Melexis Test
		0x00113		
RAM_BIST	User	0x00112	Word/Byte	Melexis Test
		0x00111		
RAM_BIST	User	0x00110	Word/Byte	Melexis Test
		0x0010F		
RAM_BIST	User	0x0010E	Word/Byte	Melexis Test
		0x0010D		
RAM_BIST	User	0x0010C	Word/Read only	Melexis Test
		0x0010B		
RAM_BIST	User	0x0010A	Word/Read only	Melexis Test
		0x00109		
RAM_BIST	User	0x00108	Word/Read only	Melexis Test
		0x00107		
RAM_BIST	User	0x00106	Word/Read only	Melexis Test
		0x00105		
RAM_BIST	User	0x00104	Word	Melexis Test
		0x00103		
VERSION	User	0x00102	W/B/Read only	Melexis Internal
		0x00101		

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
VERSION	User	0x00100	W/B/Read only	Melexis Internal
		0x000C1		
MLX16	System	0x000C0	Word/Byte	Melexis Internal
		0x000BF		
MLX16	System	0x000BE	Word/Read only	Melexis Internal
		0x00097		
MLX16	System	0x00096	Word/Byte	Interrupt Priority Port ITC_PRIO7[5:0]
		0x00095		
MLX16	System	0x00094	Word/Byte	Interrupt Priority Port ITC_PRIO6[15:0]
		0x00093		
MLX16	System	0x00092	Word/Byte	Interrupt Priority Port ITC_PRIO5[15:0]
		0x00091		
MLX16	System	0x00090	Word/Byte	Interrupt Priority Port ITC_PRIO4[15:0]
		0x0008F		
MLX16	System	0x0008E	Word/Byte	Interrupt Priority Port ITC_PRIO3[15:0]
		0x0008D		
MLX16	System	0x0008C	Word/Byte	Interrupt Priority Port ITC_PRIO2[15:0]
		0x0008B		
MLX16	System	0x0008A	Word/Byte	Interrupt Priority Port ITC_PRIO1[15:0]
		0x00089		
MLX16	System	0x00088	Word/Byte	Interrupt Priority Port ITC_PRIO0[15:0]
		0x00077		
MLX16	System	0x00076	Word/Byte	Interrupt Mask Port ITC_MASK5[0]
		0x00075		
MLX16	System	0x00074	Word/Byte	Interrupt Mask Port ITC_MASK4[15:0]
		0x00073		
MLX16	System	0x00072	Word/Byte	Interrupt Mask Port ITC_MASK3[15:0]
		0x00071		
MLX16	System	0x00070	Word/Byte	Interrupt Mask Port ITC_MASK2[15:0]
		0x0006F		
MLX16	System	0x0006E	Word/Byte	Interrupt Mask Port ITC_MASK1[15:0]
		0x0006D		
MLX16	System	0x0006C	Word/Byte	Interrupt Mask Port ITC_MASK0[15:0]
		0x0006B		
MLX16	System	0x0006A	Word/Byte	Trigger Software interrupt
		0x0005B		
MLX16	System	0x0005A	Word/Byte	Interrupt Pending Port ITC_PEND5[0]
		0x00059		
MLX16	System	0x00058	Word/Byte	Interrupt Pending Port ITC_PEND4[15:0]
		0x00057		
MLX16	System	0x00056	Word/Byte	Interrupt Pending Port ITC_PEND3[15:0]
		0x00055		
MLX16	System	0x00054	Word/Byte	Interrupt Pending Port ITC_PEND2[15:0]
		0x00053		
MLX16	System	0x00052	Word/Byte	Interrupt Pending Port ITC_PEND1[15:0]
		0x00051		
MLX16	System	0x00050	Word/Byte	Interrupt Pending Port ITC_PEND0[15:0]
		0x0004F		
MLX16	System	0x0004E	Word/Byte	Control port for debugger
		0x0004D		
MLX16	System	0x0004C	Word/Byte	Control port for debugger
		0x0004B		

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
MLX16	System	0x0004A	Word/Byte	Control port for debugger
		0x00049		
MLX16	System	0x00048	Word/Byte	Control port for debugger
		0x00047		
MLX16	System	0x00046	Word/Byte	Control port for debugger
		0x00045		
MLX16	System	0x00044	Word/Byte	Control port for debugger
		0x00043		
MLX16	System	0x00042	Word/Byte	Control port for debugger
		0x00041		
MLX16	System	0x00040	Word/Byte	Control port for debugger
		0x0003F		
MLX16	User	0x0003E	Word/Byte	Control port for debugger
		0x0003D		
MLX16	User	0x0003C	Word/Byte	Control port for debugger
		0x0003B		
MLX16	User	0x0003A	Word/Byte	Control port for debugger
		0x00039		
MLX16	User	0x00038	Word/Byte	Control port for debugger
		0x00037		
PORT_LIN_XKEY	System	0x00036	Word/Byte	LIN port KEY register
		0x00035		
PORT_ADC_CTRL	User	0x00034	Word/Byte	ADC – Control and Status Port
		0x00033		
STIMER	User	0x00032	Word	Simple Timer – Control port
		0x00031		
STIMER	User	0x00030	Word	Simple Timer – Current Counter Value
		0x0002F		
IWD	User	0x0002E	Word/Byte	Intelligent Watchdog control / status
		0x0002D		
IWD	User	0x0002C	Word/Byte	Intelligent Watchdog control / status
		0x0002B		
RAM_SHELL	User	0x0002A	Word/Byte	Melexis Test
		0x00027		
ROM_BIST	User	0x00026	Word/Byte	Melexis Test
		0x00025		
ROM_BIST	User	0x00024	Word/Byte	Melexis Test
		0x00023		
ROM_BIST	User	0x00022	Word/Byte	Melexis Test
		0x00021		
ROM_BIST	User	0x00020	Word/Byte	Melexis Test
		0x0001F		
ROM_BIST	User	0x0001E	Word/Byte	Melexis Test
		0x0001D		
ROM_BIST	User	0x0001C	Word/Byte	Melexis Test
		0x0001B		
ROM_BIST	User	0x0001A	Word/Byte	Melexis Test
		0x00019		
ROM_BIST	User	0x00018	Word/Byte	Melexis Test
		0x00017		
ROM_BIST	User	0x00016	Word/Byte	Melexis Test
		0x00015		
ROM_BIST	User	0x00014	Word/Byte	Melexis Test

Block Name	System / User prot.	Address MLX81346	Access Mode	Description
		0x00013		
ROM_SHELL	Test/User	0x00012	Word/Byte	Melexis Test
		0x00011		
ROM_SHELL	User	0x00010	Word/Byte	Melexis Test
		0x0000F		
AWD	User	0x0000E	Byte/Bit	Analog Watchdog control / status
		0x0000D		
MUPET	User	0x0000C	Word/Byte	Control port for debugger
		0x0000B		
MUPET	User	0x0000A	Word/Byte	Control port for debugger (send/receive)
		0x00009		
FUNC_TEST	System	0x00008	Word/Byte	Melexis Test
		0x00007		
FUNC_TEST	Test/Syst.	0x00006	Word/Byte	Melexis Test
		0x00003		
RST_CTRL	System	0x00002	Word/Byte	Shows Reset source

Table 106 – MLX81346 ports overview

14.3.4. Memories

14.3.4.1. Flash Memory

The MLX81346 contains a flash memory of 64kB and is organized in 512 pages of 16 x 72-bit (64 user bits per 72-bit). The 64 data bits are secured by a hardware ECC mechanism (ECC= Error Checking and Correcting) with additional 8 check bits. By this measure, the memory can correct a single bit fail and detect a double bit fail per quad word. The double bit error will be signalized via signal FL_DATA_CORRUPTED in port EEPROM_FLASH (see section 14.3.4.1.1). Interrupt FL_ECC can be enabled on request in order to bring the system in a safe mode and/or to execute a reset.

The write of a flash memory page takes typically 5ms while the erase procedure takes typically 30ms. The memory is read by the CPU at full system clock speed MCU_CLK.

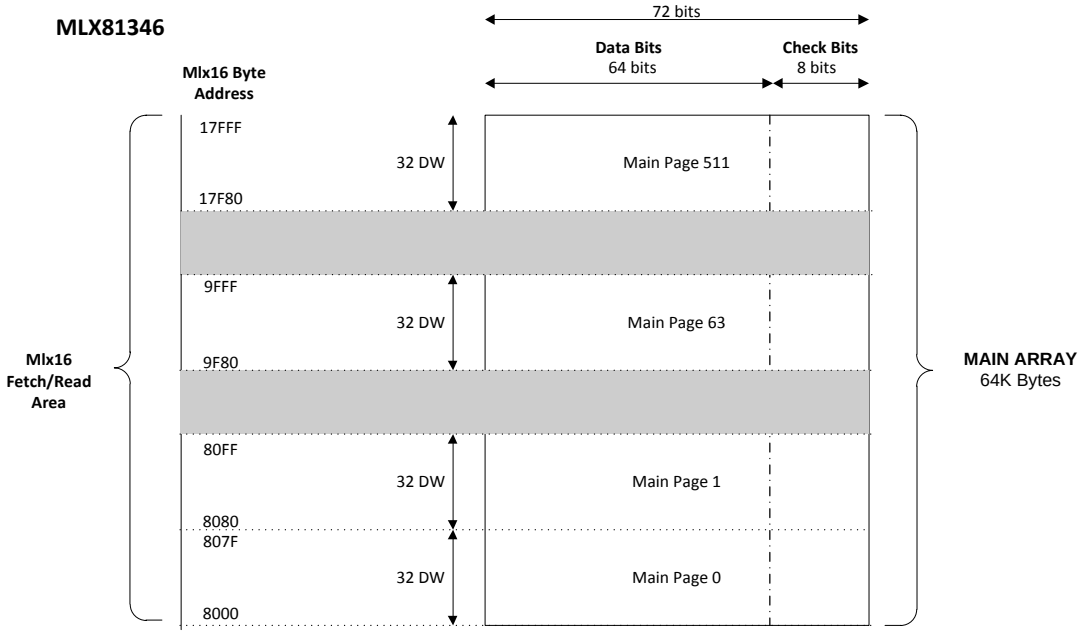


Figure 51 – Flash organization of MLX81346 (16Kx32)

The Flash memory can be programmed:

- during development: via test pin interface using the Melexis Mini E-Mlx emulator
- for mass production: via pin LIN on PCB or module level:
  - using standard LIN protocol and 3rd party LIN equipment (customer programming software in FLASH)
  - using fast PPM protocol [4] and 3rd party production equipment (MLX firmware support in ROM)

Data retention **after**	
up to 1,000 programming / erase cycles at $T_{j,av} \leq 85°C$	min. 2000h at $T_j \leq 175°C$

Table 107 – Flash Cycles and data retention

The maximum programming and erase temperature is 85°C. Data retention is still guaranteed after applying the maximum allowed number of cycles.

14.3.4.1.1. Flash status port

Port: EEPROM_FLASH

Address	Reset	Access

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:9]		Not used
FL_EXTENDED_DATA	8	RW	During a write, allows to extend the 16 bits of the write bus on all the bits of the memory (including ECC)
-	[7:2]	R	Always read 0
		W	Do nothing
FL_DATA_CORRUPTED	1	R	Corruption of data is detected (2 errors in 64 -bit data or 1 error in address)
		W	Writing 1 to clear
FL_SBE	0	R	Single bit error correction. The data was successfully corrected
		W	Writing 1 to clear

Table 108 — EEPROM_FLASH

14.3.4.2. EEPROM

The EEPROM allows storing non-volatile information of customer application data. MLX81346 has one EEPROM block of 576Byte. It is organized in 72 pages and contains an error detection and correction mechanism (ECC). The internal structure of the EEPROM is 64 data bits + 14 ECC bits. For each 32-bit double word in a 64-bit page, a single bit error correction and a double bit error detection is possible.

The EEPROM can be programmed:

- during development: via the test pin interface using the Melexis Mini E-Mlx emulator
- for mass production: via the pin LIN in the application on PCB or module level:
  - using standard LIN protocol and 3rd party LIN equipment (customer programming software in FLASH)
  - using fast PPM protocol [4] and 3rd party production equipment (MLX firmware support in ROM)

Data retention **after**	
up to 100,000 programming / erase cycles at $T_{j,av} \leq 25°C$ **or** up to 10,000 programming / erase cycles at $T_{j,av} \leq 150°C$	min. 2000h at $T_j \leq 175°C$

Table 109 – EEPROM Cycles and data retention

Note: Melexis supports read, write and store operations with library routines. The store operation is completely under software control. It can be performed at any time.

14.3.4.2.1. EEPROM organization

Device	Size	Read	Write	Address range	Usage
EEPROM_CS	64 Byte (32 Words)	User	Testmode	0x00600 – 0x0063F	Melexis trim data
EEPROM_CAL	80 Byte (40 Words)	User	Testmode	0x005B0 – 0x005FF	Calibration data
EEPROM	368 Byte (184 Words)	User	User	0x00440 – 0x005AF	User application
EEPROM_PATCH	64 Byte (32 Words)	User	User	0x00400 – 0x0043F	ROM code patch area (User application)

Table 110 – EEPROM organization

Syntax:
User: Store and recall only in System mode
Testmode: Access possible only in Test mode

Each page is separately selectable for non-volatile operation.

Note: EEPROM sections EEPROM_CS and EEPROM_CAL are reserved for Melexis and are used for storing of IC Specific trimming and calibration data. These sections are not accessible by software and cannot be erased or overwritten. EEPROM section EEPROM_PATCH is reserved for possible ROM patch code. This section is fully accessible by software and can be principally used for customer data in case it is not used for a ROM patch. The user should check with Melexis.

14.3.4.2.2. EEPROM status ports

Port: EEPROM_FLASH

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0xC000	Word, Byte, Invalid

Field name	Bit	R/W	Description
EE_BUSY	15	R	The EEPROM is busy, any access is invalid
EE_BUSY_STBY	14	R	The EEPROM is in stand-by mode
EE_BUSY_WR	13	R	A write is in progress
EE_BUSY_BUF_NOT_EMPTY	12	R	Buffers are not empty
-	11		Not used
EE_W_MODE	[10:8]	RW	Indicate the EEPROM operation mode: 000: performs the complete XFAB flow (pre-write, erase, write steps) 001: performs only write 010: performs pre-write and erase 011: performs only erase 100: performs only pre-write Others : The state machine performs XFAB flow (prewrite, erase, write)
EE_WE_KEY	[7:4]	RW	Protection bit: Key for write enable WE_KEY = 0x7 → write accesses are valid WE_KEY != 0x7 → write accesses are invalid This port is automatically reset at the end of a write access in normal mode. Read access only possible if WE_KEY != 0x7.
	[3:2]		Not used
EE_CONFIGURED	1	RW	When set, the EEPROM leaves configurable state
EE_ACTIVE	0	RW	When set, the EEPROM is not in stand-by mode

Port: EEPROM_FLASH

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Address	Reset		Access
Field name	**Bit**	**R/W**	**Description**
-	[15:4]		Not used
EE_DATA_CORRUPTED_2	3	R W	Corruption of the second 32-bit word is detected (2 errors in data or 1 error in address) Writing '1' clears the bit
EE_SBE_2	2	R W	Single bit error correction of the second 32-bit word. If '1', the data was corrected Writing '1' clears the bit
EE_DATA_CORRUPTED_1	1	R W	Corruption of the first 32-bit word is detected (2 errors in data or 1 error in address) Writing '1' clears the bit
EE_SBE_1	0	R W	Single bit error correction of the first 32-bit word. If '1', the data was corrected Writing '1' clears the bit

Table 111 — EEPROM_FLASH

14.3.5. Interrupts

This chapter explains how the interrupts are handled inside the chip. It clarifies and gives add-on information on the description of the interrupts in the MLX16-FX data book. It also gives an overview of the interrupt allocation table.

14.3.5.1. Interrupt sources

There are a number of different interrupt sources in the chip:

- system interrupts like system reset, stack overflow, address error …
- user block interrupts like PWM interrupts, timer interrupts …

14.3.5.1.1. High level system interrupts

The system interrupts are the interrupts at position 0 to 7. For those interrupt routines, Melexis provides an interrupt service routine to handle them correctly.
These routines handle:

- The correct start-up and power down of the chip
- EEPROM write routines,
- Invalid address, Program and Protection errors
- Stack overflows

14.3.5.1.1.1. Reset and Watchdog interrupt

This interrupt is generated at power on reset or if the intelligent watchdog requests a reset. In order to track back the reason for the reset, port RST_CTRL shows all possible reset sources.

Port: RST_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		System, Word, Byte
Field name	**Bit**	**R/W**	**Description**
HVDIG_OK	15	R	1: High Voltage Digital OK 0: High Voltage Digital not OK
		W	No effect
HVDIG_USED	14	RW	0: No high-voltage digital part used 1: High-voltage digital part used
SOFT_RESET	13	RW	0: No request to reset hardware 1: Request to reset hardware (automatically cleared)
-	[12:5]	R	Always read 0
		W	Do nothing
IWD_WBOOT	4	R	Warm boot reset: 1: Due to this source 0: Not due to this source
		W	1: Clear bit 0: No effect

Address	Reset		Access
DBG_WBOOT	3	R	Warm boot reset: 1: Due to debugger 0: Not due to debugger
		W	1: Clear bit 0: No effect
HVDIG_WBOOT	2	R	Warm boot reset: 1: Due to High Voltage Digital lost (HVDIG_OK = 0 and HVDIG_USED = 1) 0: Not due to High Voltage Digital
		W	1: Clear bit 0: No effect
SOFT_WBOOT	1	R	Warm boot reset: 1: Due to SOFT_RESET 0: Not due to SOFT_RESET
		W	1: Clear bit 0: No effect
AWD_WBOOT	0	R	Warm boot reset: 1: Due to Absolute Watchdog 0: Not due to Absolute Watchdog
		W	1: Clear bit 0: No effect

Table 112 — Port RST_CTRL

14.3.5.1.1.2. Stack error

A stack error occurs when the MLX16-FX uses the stack pointer to access an invalid or an unauthorized area. No protection error or invalid address interrupt is generated. Obviously this interrupt uses a jump and does not push the program counter into the stack.

14.3.5.1.1.3. Exception error

There are 3 interrupts due to software errors on the MLX16-FX, being a protection error, an invalid address or a program error. These interrupts are at level 0, therefore no return to the interrupted program is possible. The program counter value pushed on the stack must be used only for debugging purposes.

14.3.5.1.1.4. Protection error

A Protection Error is caused by the following actions:

- Write in user mode into the Flash or EEPROM
- Access to EEPROM while it is busy (EE_BUSY=1)
- Write in port while it is busy (EE_BUSY=1)
- ADC conversion request while ADC is busy (READY=0)
- Write in user mode into a system port
- Access error in the Intelligent Watchdog
- Clear the USER bit of MLX16-FX register M (try to enter system mode) not after a *jump* or *call* far page instruction

14.3.5.1.1.5. Memory error (Invalid address)

An Invalid Address Interrupt occurs when the Mlx16-FX executes an invalid memory access. The interrupt is caused by the following actions:

- Read, write or fetch a word at an odd address
- Read, write or fetch into an unused area
- Write a byte or a bit in EEPROM or Flash
- Write a bit in a digital port supporting only word or byte
- Fetch into port area
- Access to an undefined custom IO port ('User' or 'System')

14.3.5.1.1.6. Operation error (Program error)

A program error occurs when the MLX16-FX tries to execute an invalid instruction. Typically, it means that the Mlx16-FX has an invalid value in its program counter (PC).

14.3.5.1.2. User block interrupts

The user block interrupts contains interrupts coming from:

- Timer modules
- PWM modules
- UART modules
- Watchdog attention interrupt

The user is supposed to write the appropriate interrupt handlers for these interrupts.

Those interrupts typically share a single interrupt line of the MLX16-FX interrupt controller. They are organized in a second level interrupt controller as described below.

14.3.5.2. Interrupt management

14.3.5.2.1. Interrupt enabling and masking

Every non system interrupt can be enabled or masked. The mask bit does not disable the interrupt source.

All masking bits are grouped into dedicated ports; please refer to the ports map for details. When the ITC_MASKx bit is set, the corresponding interrupt is enabled.
After reset, the ITC_MASKx ports are reset so that all interrupts are disabled by default.

In case interrupts are enabled by their corresponding enable flags but disabled by priority, the interrupt sources are still active. An interrupt is memorized only once and will be handled as soon as the CPU operates on a priority which allows the execution of it.

14.3.5.2.2. Pending interrupts

When an interrupt request (IRQ) occurs and cannot be served immediately, the pending bit of this IRQ is set. All the pending bits are dedicated ports; please refer to the ports map for details.
Even if interrupts are masked, the state of the sources can be checked by reading the correponding pending bits.

The pending bit is set only once, even if more than one IRQ occurred before being serviced.

Software can clear a pending interrupt source by setting the corresponding bit in the IO port ITC_PENDx.

14.3.5.2.3. CALL and JUMP interrupts

When an interrupt occurs, the interrupt controller suppresses the next instruction fetched from memory and displays another instruction to the MLX16-FX CPU instead.
This instruction is typically a CALL to the specified interrupt vector address (for reset and stack error, a JUMP instead of CALL is issued).

At the interrupt vector the priority is set and a jump to the appropriate interrupt service routine (ISR) follows. At the end of the ISR a return can be placed, and the program counter returns to where it came from before the IRQ occurred.

As after a stack error or a reset a return point cannot be defined, those interrupts execute a JUMP to the vector address.

14.3.5.3. Interrupt priorities

Every interrupt source has its own priority. The priorities are grouped into eight classes, from priority zero to priority seven; zero is the highest priority, seven the lowest. An ISR can be interrupted by any other IRQ with the same priority or higher – i.e. the same class or lower.

There are some mechanisms that define the interrupt priority:

- The user priority defines the current working priority level of the running software. This is set by the PR bits in the MLX16-FX M register and is used to decide, if an interrupt request is allowed to interrupt the current code (ISR or normal routines).
- The software priority (or absolute priority) defines the priority of the interrupt request. It must be higher or equal than the current user priority (lower number) to allow interrupting the current code. The software priority of the non-system interrupts can be programmed through the corresponding port.
- The hardware priority (or priority position) is used as conflict resolver in case 2 interrupt sources of same software priority are pending.

14.3.5.3.1. User priority

During execution of every piece of code, there is a certain level of interrupt masking. The PR bits in the M register of the MLX16-FX define the interrupt priority of an IRQ that can interrupt the currently executing code.
Every IRQ with a software interrupt priority equal or higher (equal or lower number) than this value can interrupt this routine.
This is true for all pieces of code, ISR as well as the main loop. If during the main loop, these PR bits are set to 4, an IRQ of priority 5 will never be seen.

It is the responsibility of the customer to make sure that in the main loop these PR bits have the correct value so that all wanted interrupts can be acknowledged. In the Melexis firmware platform, this interrupt level is set to seven (lowest priority) when entering the main function. This means that every IRQ can interrupt the main function.

When entering an ISR, the first instruction has to be a PSUP, #constant instruction that pushes this M register on the stack and sets a new value in the PR bits (#constant parameter in this instruction, value from 0 to 7). By writing a new value in the PR bits in the ISR, another level of interrupts can be masked.

It is possible that an ISR of an IRQ with software priority 5 sets these bits to 2. By doing so, all IRQs with software priority less than 2 (higher number) are blocked. This means that an ISR routine of an IRQ with software priority 5 can block an IRQ with software priority 4 by changing the PR bits in its ISR. This is called “interrupt priority inversion”. The customer has the responsibility to check if this can cause problems in his application.

The M register has to be manually restored when exiting the ISR by popping it from the stack. This has to be the very last instruction of the ISR.

The PSUP and pop instructions to keep track of the M register and PR bits are handled automatically in the Melexis firmware platform.

14.3.5.4. Interrupt vectors

Name	Pos	Prio-rity	Type	Priority Port	Mask Port	Pending Port	Page	Address	Description
MLX16_RESET	0	0	Jump				FP0:00	0x02C00	Reset
MLX16_STACKERR	1	0	Jump				FP0:08	0x02C08	Stack error
MLX16_PROTERR	2	0	Jump				FP0:10	0x02C10	Protection error
MLX16_MEMERR	3	0	Jump				FP0:18	0x02C18	Memory error
MLX16_OPERR	4	0	Jump				FP0:20	0x02C20	Operation error
MLX16_EXCHG	6	1	Call		ITC_MASK0[0]	ITC_PEND0[0]	FP0:30	0x02C30	Exchange (PTC and Debugger)
MLX16_DMAERR	7	1	Call		ITC_MASK0[1]	ITC_PEND0[1]	FP0:38	0x02C38	DMA error
AWD_ATT	8	1	Call		ITC_MASK0[2]	ITC_PEND0[2]	FP0:40	0x02C40	Absolute watchdog attention
IWD_ATT	9	1	Call		ITC_MASK0[3]	ITC_PEND0[3]	FP0:48	0x02C48	Intelligent watchdog attention
FL_ECC	10	1	Call		ITC_MASK0[4]	ITC_PEND0[4]	FP0:50	0x02C50	Flash ECC double/multiple bit error interrupt
EE_ECC	11	1	Call		ITC_MASK0[5]	ITC_PEND0[5]	FP0:58	0x02C58	EEPROM ECC error interrupt
UV_VDDA	12	1	Call		ITC_MASK0[6]	ITC_PEND0[6]	FP0:60	0x02C60	VDDA under-voltage interrupt
UV_VS	13	1	Call		ITC_MASK0[7]	ITC_PEND0[7]	FP0:68	0x02C68	Supply under-voltage interrupt
UV_BOOST	14	1	Call		ITC_MASK0[8]	ITC_PEND0[8]	FP0:70	0x02C70	Under-voltage at predriver charge pump
ANA_PLL_ERR	15	2	Call		ITC_MASK0[9]	ITC_PEND0[9]	FP0:78	0x02C78	placeholder for PLL error interrupt
OVT	16	2	Call		ITC_MASK0[10]	ITC_PEND0[10]	FP0:80	0x02C80	Over-temperature interrupt
OVC	17	2	Call		ITC_MASK0[11]	ITC_PEND0[11]	FP0:88	0x02C88	Over-current interrupt
OC_VDDA	18	2	Call		ITC_MASK0[12]	ITC_PEND0[12]	FP0:90	0x02C90	Over-current at VDDA
OV_HS_IO3	19	2	Call		ITC_MASK0[13]	ITC_PEND0[13]	FP0:98	0x02C98	Over-voltage HS drain-source monitor at IO3
OV_HS_IO4	20	2	Call		ITC_MASK0[14]	ITC_PEND0[14]	FP0:A0	0x02CA0	Over-voltage HS drain-source monitor at IO4
OV_HS_VDS0	21	2	Call		ITC_MASK0[15]	ITC_PEND0[15]	FP0:A8	0x02CA8	VDS monitor interrupt at HS phase U
OV_HS_VDS1	22	2	Call		ITC_MASK1[0]	ITC_PEND1[0]	FP0:B0	0x02CB0	VDS monitor interrupt at HS phase V

OV_HS_VDS2	23	2	Call		ITC_MASK1[1]	ITC_PEND1[1]	FP0:B8	0x02CB8	VDS monitor interrupt at HS phase W
OV_LS_VDS0	24	2	Call		ITC_MASK1[2]	ITC_PEND1[2]	FP0:C0	0x02CC0	VDS monitor interrupt at LS phase U
OV_LS_VDS1	25	2	Call		ITC_MASK1[3]	ITC_PEND1[3]	FP0:C8	0x02CC8	VDS monitor interrupt at LS phase V
OV_LS_VDS2	26	2	Call		ITC_MASK1[4]	ITC_PEND1[4]	FP0:D0	0x02CD0	VDS monitor interrupt at LS phase W
STIMER	27	3-6	Call	ITC_PRIO0[1:0]	ITC_MASK1[5]	ITC_PEND1[5]	FP0:D8	0x02CD8	Simple Timer interrupt
CTIMER0_1	28	3-6	Call	ITC_PRIO0[3:2]	ITC_MASK1[6]	ITC_PEND1[6]	FP0:E0	0x02CE0	Complex Timer interrupt 1
CTIMER0_2	29	3-6	Call	ITC_PRIO0[5:4]	ITC_MASK1[7]	ITC_PEND1[7]	FP0:E8	0x02CE8	Complex Timer interrupt 2
CTIMER0_3	30	3-6	Call	ITC_PRIO0[7:6]	ITC_MASK1[8]	ITC_PEND1[8]	FP0:F0	0x02CF0	Complex Timer interrupt 3
CTIMER1_1	31	3-6	Call	ITC_PRIO0[9:8]	ITC_MASK1[9]	ITC_PEND1[9]	FP0:F8	0x02CF8	Complex Timer interrupt 1
CTIMER1_2	32	3-6	Call	ITC_PRIO0[11:10]	ITC_MASK1[10]	ITC_PEND1[10]	FP1:00	0x02D00	Complex Timer interrupt 2
CTIMER1_3	33	3-6	Call	ITC_PRIO0[13:12]	ITC_MASK1[11]	ITC_PEND1[11]	FP1:08	0x02D08	Complex Timer interrupt 3
SPI_TE	34	3-6	Call	ITC_PRIO0[15:14]	ITC_MASK1[12]	ITC_PEND1[12]	FP1:10	0x02D10	SPI transmit register is empty
SPI_RF	35	3-6	Call	ITC_PRIO1[1:0]	ITC_MASK1[13]	ITC_PEND1[13]	FP1:18	0x02D18	SPI receive register full
SPI_ER	36	3-6	Call	ITC_PRIO1[3:2]	ITC_MASK1[14]	ITC_PEND1[14]	FP1:20	0x02D20	SPI transmission error
PWM_MASTER1_CMP	37	3-6	Call	ITC_PRIO1[5:4]	ITC_MASK1[15]	ITC_PEND1[15]	FP1:28	0x02D28	Custom interrupt during the PWM period
PWM_MASTER1_END	38	3-6	Call	ITC_PRIO1[7:6]	ITC_MASK2[0]	ITC_PEND2[0]	FP1:30	0x02D30	Interrupt at the end of the PWM period
PWM_SLAVE1_CMP	39	3-6	Call	ITC_PRIO1[9:8]	ITC_MASK2[1]	ITC_PEND2[1]	FP1:38	0x02D38	Custom interrupt during the PWM period
PWM_SLAVE2_CMP	40	3-6	Call	ITC_PRIO1[11:10]	ITC_MASK2[2]	ITC_PEND2[2]	FP1:40	0x02D40	Custom interrupt during the PWM period
PWM_SLAVE3_CMP	41	3-6	Call	ITC_PRIO1[13:12]	ITC_MASK2[3]	ITC_PEND2[3]	FP1:48	0x02D48	Custom interrupt during the PWM period
PWM_MASTER2_CMP	42	3-6	Call	ITC_PRIO1[15:14]	ITC_MASK2[4]	ITC_PEND2[4]	FP1:50	0x02D50	Custom interrupt during the PWM period
PWM_MASTER2_END	43	3-6	Call	ITC_PRIO2[1:0]	ITC_MASK2[5]	ITC_PEND2[5]	FP1:58	0x02D58	Interrupt at the end of the PWM period
ADC_BLOCK	44	3-6	Call	ITC_PRIO2[3:2]	ITC_MASK2[6]	ITC_PEND2[6]	FP1:60	0x02D60	ADC interrupt
EE_COMPLETE	45	3-6	Call	ITC_PRIO2[5:4]	ITC_MASK2[7]	ITC_PEND2[7]	FP1:68	0x02D68	EEPROM write completed interrupt

FL_COMPLETE	46	3-6	Call	ITC_PRIO2[7:6]	ITC_MASK2[8]	ITC_PEND2[8]	FP1:70	0x02D70	Flash write completed interrupt
COLIN_OWNMTX	47	3-6	Call	ITC_PRIO2[9:8]	ITC_MASK2[9]	ITC_PEND2[9]	FP1:78	0x02D78	mutex interrupt
COLIN_LIN	48	3-6	Call	ITC_PRIO2[11:10]	ITC_MASK2[10]	ITC_PEND2[10]	FP1:80	0x02D80	LIN interrupt
DIAG	49	3-6	Call	ITC_PRIO2[13:12]	ITC_MASK2[11]	ITC_PEND2[11]	FP1:88	0x02D88	Diagnosis interrupt
I2C_GLOBAL_RESET	50	3-6	Call	ITC_PRIO2[15:14]	ITC_MASK2[12]	ITC_PEND2[12]	FP1:90	0x02D90	I^2C reset interrupt
PPM_RX	51	3-6	Call	ITC_PRIO3[1:0]	ITC_MASK2[13]	ITC_PEND2[13]	FP1:98	0x02D98	PPM receive interrupt
PPM_TX	52	3-6	Call	ITC_PRIO3[3:2]	ITC_MASK2[14]	ITC_PEND2[14]	FP1:A0	0x02DA0	PPM transmit interrupt
PPM_ERR	53	3-6	Call	ITC_PRIO3[5:4]	ITC_MASK2[15]	ITC_PEND2[15]	FP1:A8	0x02DA8	PPM error interrupt
OV_VS	54	3-6	Call	ITC_PRIO3[7:6]	ITC_MASK3[0]	ITC_PEND3[0]	FP1:B0	0x02DB0	Supply over-voltage interrupt
OV_VDDA	55	3-6	Call	ITC_PRIO3[9:8]	ITC_MASK3[1]	ITC_PEND3[1]	FP1:B8	0x02DB8	Over-voltage at VDDA
OV_BOOST	56	3-6	Call	ITC_PRIO3[11:10]	ITC_MASK3[2]	ITC_PEND3[2]	FP1:C0	0x02DC0	Over-voltage at predriver charge pump
IO_IN0	57	3-6	Call	ITC_PRIO3[13:12]	ITC_MASK3[3]	ITC_PEND3[3]	FP1:C8	0x02DC8	IO interrupt for IO0
IO_IN1	58	3-6	Call	ITC_PRIO3[15:14]	ITC_MASK3[4]	ITC_PEND3[4]	FP1:D0	0x02DD0	IO interrupt for IO1
IO_IN2	59	3-6	Call	ITC_PRIO4[1:0]	ITC_MASK3[5]	ITC_PEND3[5]	FP1:D8	0x02DD8	IO interrupt for IO2
IO_IN3	60	3-6	Call	ITC_PRIO4[3:2]	ITC_MASK3[6]	ITC_PEND3[6]	FP1:E0	0x02DE0	IO interrupt for IO3
IO_IN4	61	3-6	Call	ITC_PRIO4[5:4]	ITC_MASK3[7]	ITC_PEND3[7]	FP1:E8	0x02DE8	IO interrupt for IO4
IO_IN5	62	3-6	Call	ITC_PRIO4[7:6]	ITC_MASK3[8]	ITC_PEND3[8]	FP1:F0	0x02DF0	IO interrupt for IO5
IO_IN6	63	3-6	Call	ITC_PRIO4[9:8]	ITC_MASK3[9]	ITC_PEND3[9]	FP1:F8	0x02DF8	IO interrupt for IO6
IO_IN7	64	3-6	Call	ITC_PRIO4[11:10]	ITC_MASK3[10]	ITC_PEND3[10]	FP2:00	0x02E00	IO interrupt for IO7
IO_IN8	65	3-6	Call	ITC_PRIO4[13:12]	ITC_MASK3[11]	ITC_PEND3[11]	FP2:08	0x02E08	IO interrupt for IO8
IO_IN9	66	3-6	Call	ITC_PRIO4[15:14]	ITC_MASK3[12]	ITC_PEND3[12]	FP2:10	0x02E10	IO interrupt for IO9
IO_IN10	67	3-6	Call	ITC_PRIO5[1:0]	ITC_MASK3[13]	ITC_PEND3[13]	FP2:18	0x02E18	IO interrupt for IO10
IO_IN11	68	3-6	Call	ITC_PRIO5[3:2]	ITC_MASK3[14]	ITC_PEND3[14]	FP2:20	0x02E20	IO interrupt for IO11

TX_TIMEOUT	69	3-6	Call	ITC_PRIO 5[5:4]	ITC_MAS K3[15]	ITC_PEN D3[15]	FP2:28	0x02E28	LIN - Dominant TX Timeout
UART0_SB	70	3-6	Call	ITC_PRIO 5[7:6]	ITC_MAS K4[0]	ITC_PEN D4[0]	FP2:30	0x02E30	UART Stop bit error
UART0_RS	71	3-6	Call	ITC_PRIO 5[9:8]	ITC_MAS K4[1]	ITC_PEN D4[1]	FP2:38	0x02E38	UART Receive error
UART0_RR	72	3-6	Call	ITC_PRIO 5[11:10]	ITC_MAS K4[2]	ITC_PEN D4[2]	FP2:40	0x02E40	UART Receive
UART0_TS	73	3-6	Call	ITC_PRIO 5[13:12]	ITC_MAS K4[3]	ITC_PEN D4[3]	FP2:48	0x02E48	UART Transmit end
UART0_TR	74	3-6	Call	ITC_PRIO 5[15:14]	ITC_MAS K4[4]	ITC_PEN D4[4]	FP2:50	0x02E50	UART Transmit beginning
UART0_TE	75	3-6	Call	ITC_PRIO 6[1:0]	ITC_MAS K4[5]	ITC_PEN D4[5]	FP2:58	0x02E58	UART Transmit error
UDFR0	76	3-6	Call	ITC_PRIO 6[3:2]	ITC_MAS K4[6]	ITC_PEN D4[6]	FP2:60	0x02E60	UART0 - DMA frame/buffer received
UDTF0	77	3-6	Call	ITC_PRIO 6[5:4]	ITC_MAS K4[7]	ITC_PEN D4[7]	FP2:68	0x02E68	UART0 - DMA frame/buffer transmitted
UART1_SB	78	3-6	Call	ITC_PRIO 6[7:6]	ITC_MAS K4[8]	ITC_PEN D4[8]	FP2:70	0x02E70	UART Stop bit error
UART1_RS	79	3-6	Call	ITC_PRIO 6[9:8]	ITC_MAS K4[9]	ITC_PEN D4[9]	FP2:78	0x02E78	UART Receive error
UART1_RR	80	3-6	Call	ITC_PRIO 6[11:10]	ITC_MAS K4[10]	ITC_PEN D4[10]	FP2:80	0x02E80	UART Receive
UART1_TS	81	3-6	Call	ITC_PRIO 6[13:12]	ITC_MAS K4[11]	ITC_PEN D4[11]	FP2:88	0x02E88	UART Transmit end
UART1_TR	82	3-6	Call	ITC_PRIO 6[15:14]	ITC_MAS K4[12]	ITC_PEN D4[12]	FP2:90	0x02E90	UART Transmit beginning
UART1_TE	83	3-6	Call	ITC_PRIO 7[1:0]	ITC_MAS K4[13]	ITC_PEN D4[13]	FP2:98	0x02E98	UART Transmit error
UDFR1	84	3-6	Call	ITC_PRIO 7[3:2]	ITC_MAS K4[14]	ITC_PEN D4[14]	FP2:A0	0x02EA0	UART1 - DMA frame/buffer received
UDTF1	85	3-6	Call	ITC_PRIO 7[5:4]	ITC_MAS K4[15]	ITC_PEN D4[15]	FP2:A8	0x02EA8	UART1 - DMA frame/buffer transmitted
MLX16_SOFT	86	7	Call		ITC_MAS K5[0]	ITC_PEN D5[0]	FP2:B0	0x02EB0	Software Interrupt request

Table 113 — Interrupt vector table

14.3.6. PWM

The MLX81346 provides 2 PWM masters and 3 PWM slaves. A master can supply all slaves with the clock signal in a daisy-chain-like hierarchical scheme (see Figure 52).

The characteristics of each of the PWM units are:

- PWM resolution of up to 16-bit
- Max. PWM input frequency 32MHz
- Programmable prescaler
- Programmable duty cycle: 0 – 100%
- Programmable phase shift and/or output period with double buffers
- Mirror mode with double buffer for symmetrical output waveform creation
- Programmable interrupt output signal anywhere within the PWM period.
- Fix interrupt output signal when new programmable data becomes active (PWM-Counter == 0 and Master mode active)
- supports synchronized operation between multiple PWM modules (Master-Slave)
- double buffer mechanism for the threshold and period registers, their update is always connected to the low threshold ports write and the counter equal to 0

To make the use of the PWM module more comfortable a macro library and an application note are available at the Softdist server.

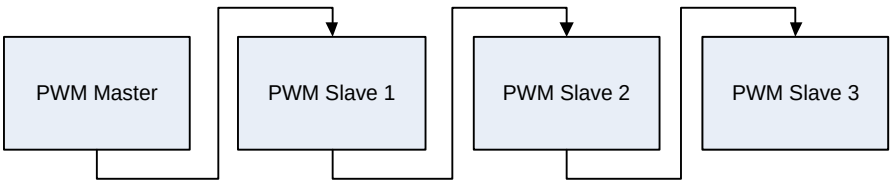


Figure 52 – Internal daisy chain connection of the PWM modules.

The block diagram of a master PWM unit is shown in Figure 53 while a slave PWM init is shown in Figure 54.

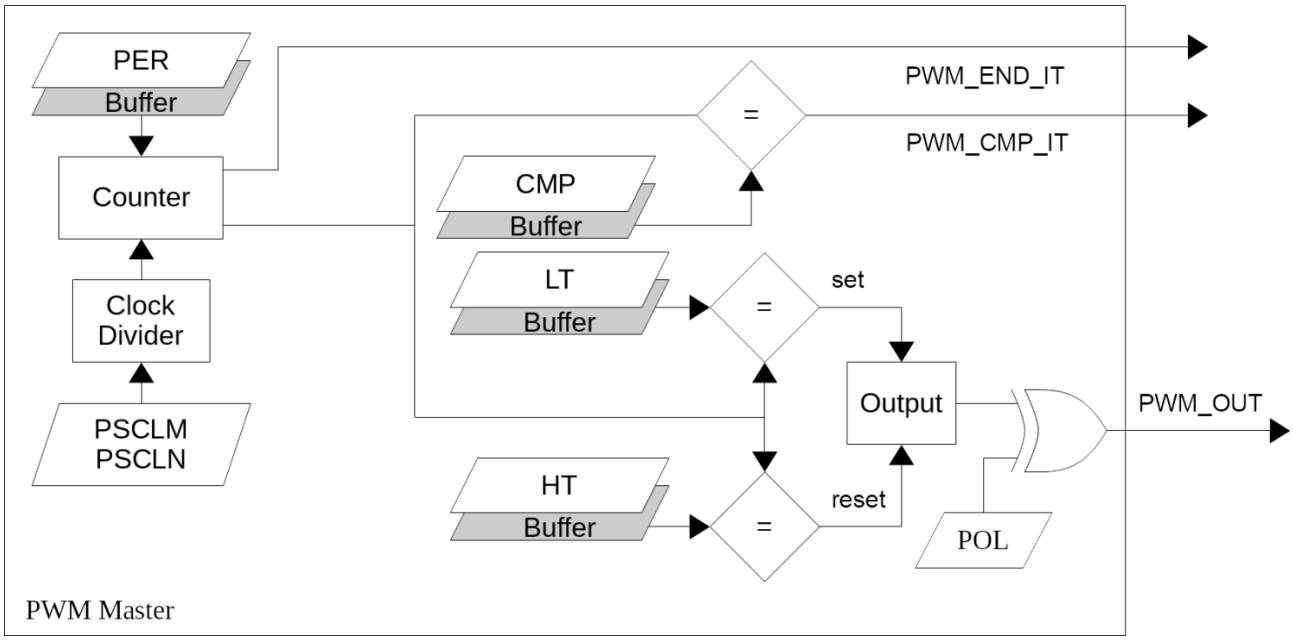


Figure 53 – Block Diagram of PWM Master unit

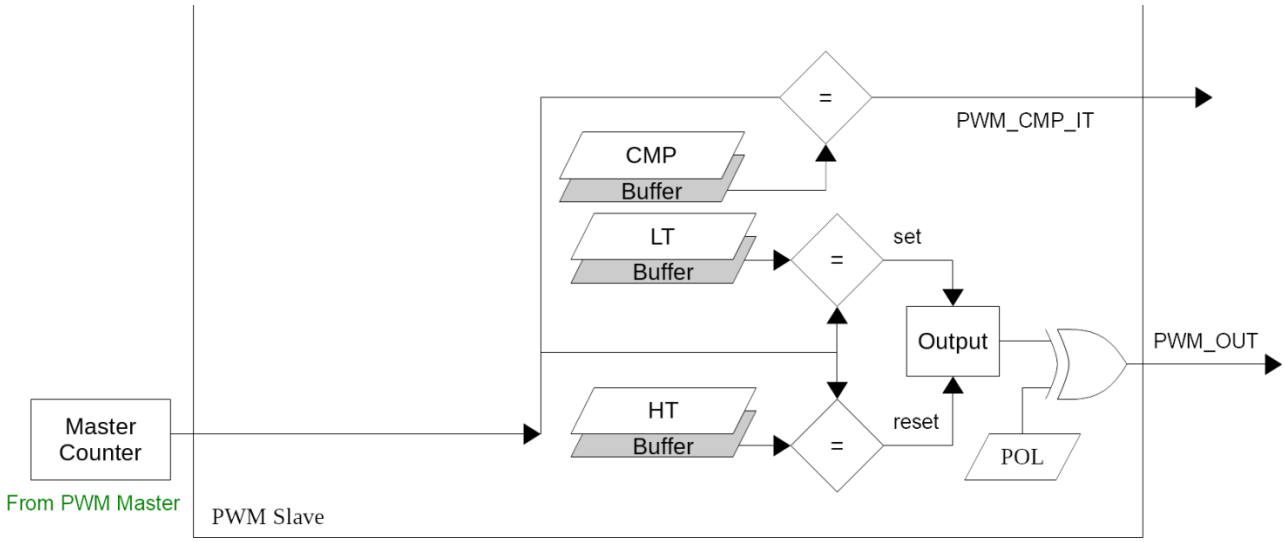


Figure 54 – Block Diagram of PWM Slave unit

14.3.6.1. Frequency control

The PWM frequency is controlled by the ports PSCLM, PSCLN and PER.
PSCLM and PSCLN control the clock of the PWM counter. This port is not buffered so any change will be effective immediately. The frequency of this clock is given by the following equation:

$$\frac{1}{Pcnt} = F_{cnt} = \frac{F_{ck}}{(PSCLM + 1)} \times \frac{1}{2^{PSCLN}}$$

with:

- P_{cnt} is the counter clock period
- F_{cnt} is the counter input frequency

F_{ck} is the frequency of the PWM internal clock (see 1.5.8 for more details).
PSCLM is the port value between 0 and 15.
PSCLN is the port value between 0 and 11.

The period of the PWM output is controlled by the PER value. This value is double buffered and is updated only when the PWM starts a new period and the LT port has been updated. The frequency of the PWM output PWMO is given by the equation:

$$F_{pwm} = \frac{F_{cnt}}{PER}$$

With
F_{cnt} is the counter input frequency.
F_{pwm} is the frequency of the PWM output.
PER is the port value.

In order to keep the maximum resolution, you need to keep PER as high as possible with PSCLM and PSCLN as low as possible, increasing PSCLM before PSCLN. By example to have a PWM running at 20 KHz with a PWM clock running at 32 MHz:

- $PER = \frac{F_{ck}}{F_{pwm}} = \frac{32MHz}{20kHz} = 1600$
- The maximum resolution is achieved with an 11-bit PWM with PSCLM = 0 and PSCLN = 0. So with an 11-bit counter the value 1600 can be reached, it is not necessary to divide $F_{ck.}$
- If you have only an 8-bit PWM:
  - the maximum period will be 255, so 1600 cannot be reached, a division of F_{ck} is necessary
  - you need to divide the clock by $\frac{1600}{255} = 6.27$
  - as we are limited to integer number, we need to divide the clock by 7
  - this is possible with PSCLM =6 keeping PSCLN = 0
  - in this configuration $PER = \frac{F_{ck}}{7} \times \frac{1}{F_{pwm}} = \frac{32MHz}{20kHz} \times \frac{1}{7} = 228.57 \approx 229$

- If you have only a 4-bit PWM:
  - the maximum period will be 15, so 1600 cannot be reached, a division of F_{ck} is necessary
  - you need to divide the clock by $\frac{1600}{15} = 106.66$
  - as we are limited to integer number, we need to divide the clock by 107
  - it has be divided first by 8 = 2^3 using PSCLN = 3 and PSCLM = 13, so F_{ck} will be divided by 8 x (13 +1) = 112, it is the closest value reachable
  - in this configuration $PER = \frac{F_{ck}}{F_{pwm}} \times \frac{1}{112} = \frac{32MHz}{20kHz} \times \frac{1}{112} = 14.28 \approx 14$

14.3.6.2. Interrupts

The PWM block generates two interrupts:

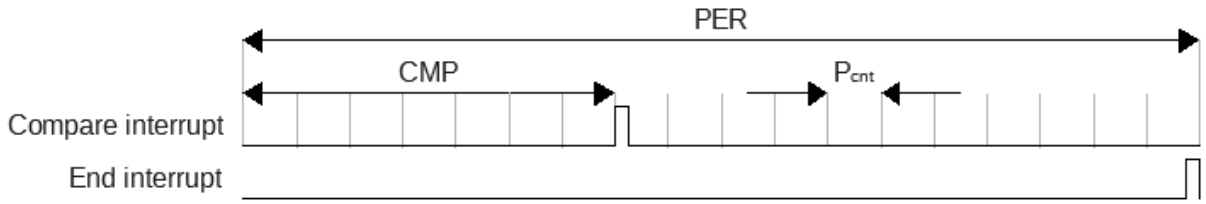


Figure 55 – PWM interrupt

The first interrupt is generated at the end of the PWM period. You can use it to change the configuration because the PWM settings have been updated just before if needed. This interrupt is only generated if the PWM is in master mode.

The second interrupt can be generated anywhere within the PWM period. Its position is configurable with the CMP port. It is generated in both slave and master mode.

Those interrupts have no enable bits but you can mask them in the interrupt controller. They are not generated if the PWM is stopped though.

14.3.6.3. Read back output

The PWM output can be read back. This bit is normally connected to the PWM analog output, so it is possible that there is a few clock periods delay between the set of the PWM in the digital and the reading of a 1 on this input. This input can
be used to check that the PWM is working.

14.3.6.4. Updating settings

When the PWM is stopped, all settings can be updated at any time.
When the PWM is running PER, CMP, HT, LT and POL registers are buffered. These values can be changed at any time but are taken into account only on the next period of the PWM after writing in the LT port. When changing several values at a time, LT must be the last register written and it must be updated anyway if another value is changed.

When one of this register is read, the buffer value is read which can be different from the value currently used by the PWM.

Note: In case LT port is changed on the last CPU cycle of a PWM period (i.e at the same time than the end interrupt), the new register values are not take into account on the next PWM period but on the next one.

The other configuration bits MODE, IDLE, SLAVE, PSCLM, PSCLN cannot be changed when the PWM is started. Nothing is changed and an invalid signal is generated if a write is done.

14.3.6.5. Starting PWM

When the PWM is stopped, its output is the value of the POL bit. A change of this bit has immediately effect to the output when the PWM is stopped.

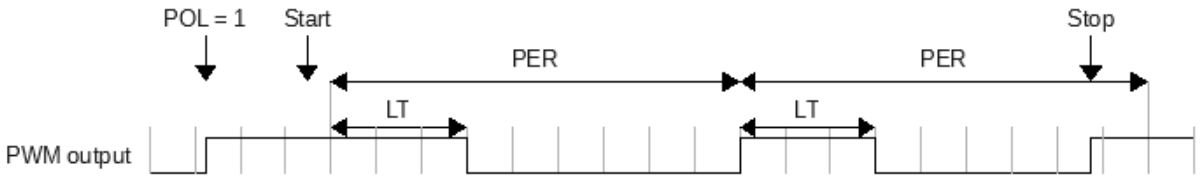


Figure 56 – Starting and Stopping PWM

Here is an example of changing the POL bit, starting the PWM and stopping it at the end of a period. The output of the PWM can be set to 0 or 1 using the LT port too.

Figure 57 – Starting PWM and changing LT with POL = 0

With LT equal 0, when the PWM is started, the output has the opposite value of POL whatever is the value of PER. Then, the PWM can be restarted while it is running by setting the start bit again.

14.3.6.6. Operation Modes

14.3.6.6.1. Simple Mode

This is the simplest mode.

Figure 58 – Simple PWM with POL = 0

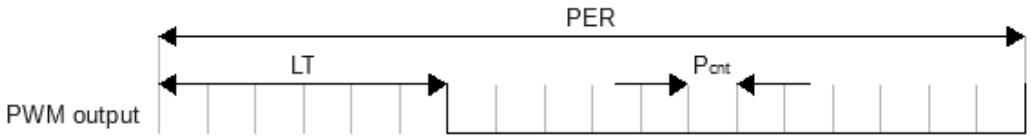


Figure 59 – Simple PWM with POL = 1

The PWM output starts at POL value, changes when the PWM counter reaches the LT value and changes back to the POL value at the end of the period. The HT register is not used.

If LT is 0, the PWM output is always at the opposite value of POL. If LT is higher or equal than $\frac{PER}{2}$, the PWM output is always at the value of POL.

14.3.6.6.2. Independent mode

In this mode, if POL is 0, the LT value defines when the PWM output is rising and the HT value defines when the PWM output is falling. If POL is 1, the output is inverted.

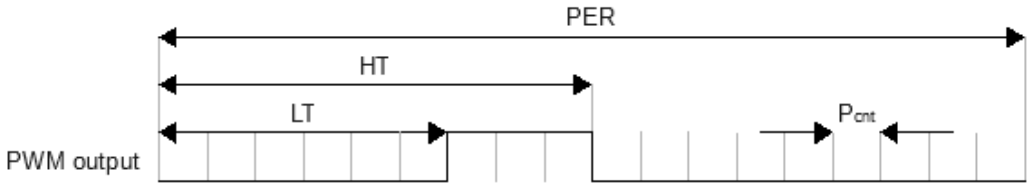


Figure 60 – Independent PWM with POL = 0

The PWM output is undefined if LT is higher than HT.

If LT is 0 and HT is greater or equal to PER, the PWM output is always at the opposite value of POL. If LT is higher or equal to PER, the output is always at the value of POL.

14.3.6.6.3. Mirror mode

This mode allows to output a centered positive or negative pulse.

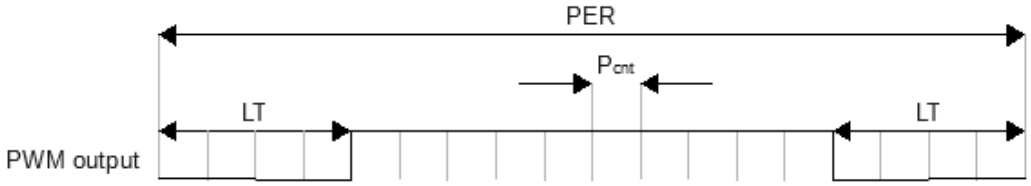


Figure 61 – Mirror PWM with POL = 0

The period of the PWM is still PER, but the pulse width is now $PER - 2 \times LT$ instead of $PER - LT$. The pulse is always centered. The HT register is not used.

If LT is 0, the PWM output is always at the opposite value of POL. If LT is equal or greater than $\frac{PER}{2}$, the PWM output is always at the value of POL.

14.3.6.7. PWM Master{1,2}

Port: PWM_MASTER{1,2}

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
CMP	[15:0]	RW	Interrupt comparator register

Address	Reset	Access

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
HT	[15:0]	RW	High threshold value

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
LT	[15:0]	RW	Low threshold value

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
PER	[15:0]	RW	PWM period

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
PSCLM	[15:12]	RW	Divide input clock by M+1
PSCLN	[11:8]	RW	Divide input clock by 2N with N between 0 and 11 included.
PWM_IN	7	R	Read back PWM output
		W	Do nothing
IDLE	6	RW	When PWM is stopped, 1: Output is invert of POL, 0: Output is POL
POL	5	RW	1: Output start at 1, 0: Output start at 0
MODE	[4:3]	RW	10: Independent mode, 01: Mirror mode, 00: Simple mode
SLAVE	2	RW	1: Slave mode, 0: Master mode
STOP	1	W	1: stop the PWM, all other bits are unused, 0 do nothing
		R	0: if the PWM is running else 1
START	0	W	1: to start the PWM, all other bits are unused, 0 do nothing
		R	1: if the PWM is running else 0

Table 114 — PWM_MASTER1 and PWM_MASTER2

14.3.6.8. PWM Slave{1,2,3}

Port: PWM_SLAVE{1,2,3}

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
CMP	[15:0]	RW	Interrupt comparator register

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
HT	[15:0]	RW	High threshold value

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
LT	[15:0]	RW	Low threshold value

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
PSCLM	[15:12]	RW	Not used
PSCLN	[11:8]	RW	Not used
PWM_IN	7	R	Read back PWM output
		W	Do nothing
IDLE	6	RW	When PWM is stopped, 1: Output is invert of POL, 0: Output is POL
POL	5	RW	1: Output start at 1, 0: Output start at 0
MODE	[4:3]	RW	10: Independent mode, 01: Mirror mode, 00: Simple mode
SLAVE	2	RW	1: Slave mode
STOP	1	W	1: stop the PWM, all other bits are unused, 0 do nothing
		R	0: if the PWM is running else 1
START	0	W	1: to start the PWM, all other bits are unused, 0 do nothing
		R	1: if the PWM is running else 0

Table 115 — PWM_SLAVE1, PWM_SLAVE2 and PWM_SLAVE3

14.3.7. Simple Timer

The Timer has following features:

- 14-bit incrementing counter
- Generates an interrupt when counter reaches the timer value
- Reset counter when it reaches the timer value
- Selectable clock enable (MCU_CLK, 1MHz or 10kHz)

The timer is disabled on a reset. It is controlled by 1 word port. The 2 MSB define the timer mode, the others 14 bits define the timer value. The counter is an incrementing counter, when it reaches VALUE, an interrupt is generating and the counter is reset.

The timer has 4 modes defined by the field MODE:

- 00: Timer disable (default value)
- 01: Counter is incremented on MCU_CLK
- 10: Counter is incremented on 1MHz clock FRC_1M (every 1µs)
- 11: Counter is incremented on 10kHz clock FRC_10K (every 100µs)

The reset value of the counter is 1, like this, the timer value really corresponds to the number of clock enable. For example, if the port value is equal to C005h (MODE = 11b and VALUE = 5h), an interrupt will occur every 500µs.

During a sequence, if the port value changes, the new value is immediately take into account. If the new timer value is smaller than the counter value, an interrupt is immediately generated. If the new timer value is bigger than the counter value, the counter will continue to be incremented until it reaches the new value.

Be careful, on a mode change it is possible that the first period is wrong (maximum 1 new clock period). It is because the mode change is immediate and it is difficult to know when the next clock enable of the new clock will come. This problem occurs when the new clock enable is slower than the previous one.

Whatever the mode chosen, the interrupt signal lasts always one MCU_CLK period.

Using value '0' for the timer value is highly inadvisable because it will generate an interrupt every clock enable. The timer has a read-only port containing the current value of the counter. The unused bits (2 MSB) are always equal to 0.

Port: STIMER

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0001	Word, Read Only

Field name	Bit	R/W	Description
-	[15:14]	R	Not used
CURRENT	[13:0]	R	Current counter value

Address	Reset	Access	Address
0x00032	0x0000		Word

Field name	Bit	R/W	Description
MODE	[15:14]	RW	00 = Disable 01 = on MCU clock

Address	Reset	Access
		10 = on 1MHz clock 11 = on 10kHz clock
VALUE	[13:0] RW	Period value

Table 116 — STIMER

14.3.7.1. Examples

Example 1: Timer on MCU_CLK with a new timer value bigger than the counter value.

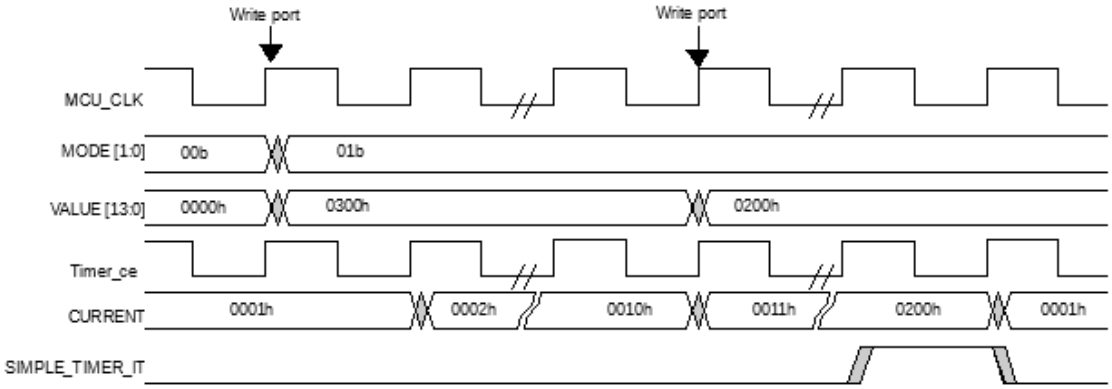


Figure 62 – Timer on MCU_CLK with a new timer value bigger than the counter value

The timer starts when MODE is different to 00. Counter (CURRENT) is incrementing on each MCU_CLK because MODE = 01. When counter is equal to 0010h, a new timer value is required by the user. Because this new value is bigger than the counter value, it changes nothing and the counter is still incremented. When it reaches VALUE, an interrupt is generated (SIMPLE_TIMER_IT) and the counter is reset.

Example 2: Timer on MCU_CLK clock with a new timer value smaller than the counter value.

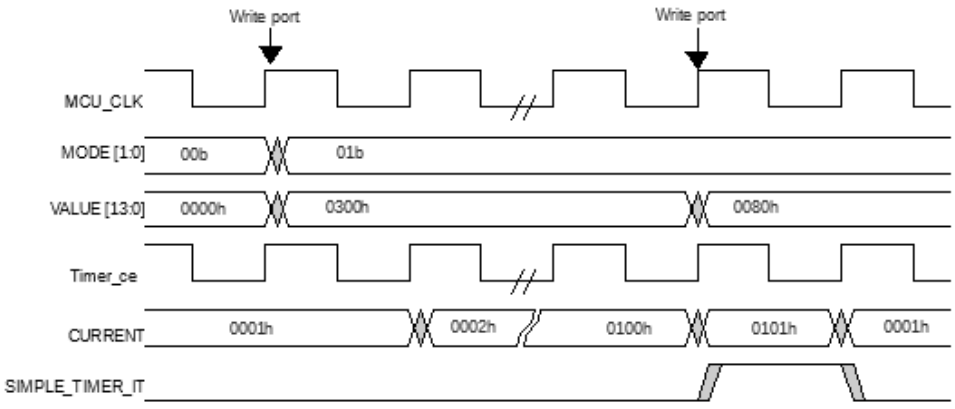


Figure 63 – Timer on MCU clock with a new timer value smaller than the counter value

The timer starts when MODE is different to 00. Counter (CURRENT) is incrementing on each MCU_CLK because MODE = 01. When counter is equal to 0100h, a new timer value is required by the user. Because this new value is smaller than the counter value, an interrupt is generated (SIMPLE_TIMER_IT) and the counter is reset.

Example 3: Timer on MCU_CLK clock first then on 1MHz clock (MCU_1US_CE)

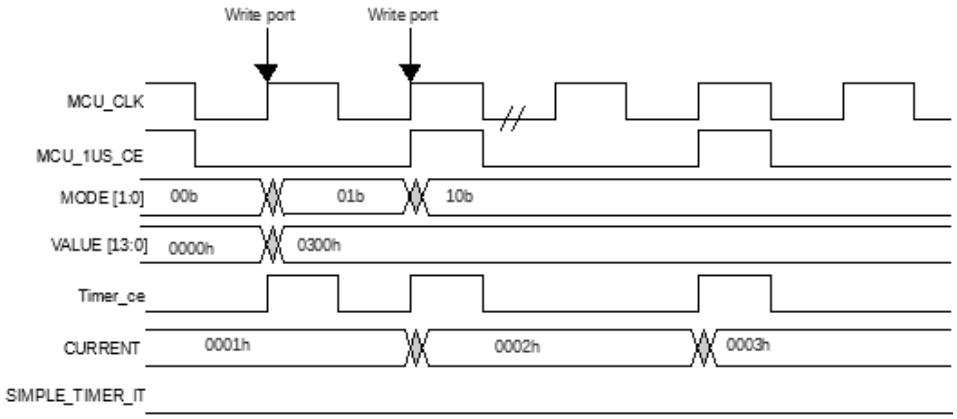


Figure 64 – Timer on MCU clock first then on 1MHz clock

The timer starts when MODE is different to 00. Because MODE = 01, the clock enable of the module is MCU_CLK. On the next access, MODE = 10 is selected, so the clock enable of the module will be 1MHz clock (MCU_1US_CE) and the counter will be incremented every 1µs.

14.3.8. Complex Timer

This timer can configured to work in 5 different modes.

- Single 16-bit auto-reload timer
- Dual 16-bit timer compare
- Dual 16-bit timer capture
- 16-bit timer compare and capture
- High resolution 16-bit PWM (no shadow register)

MODE[2:0]	Function
000	Timer
001	Dual timer compare
010	Dual timer capture
011	Timer capture/compare
110	PWM

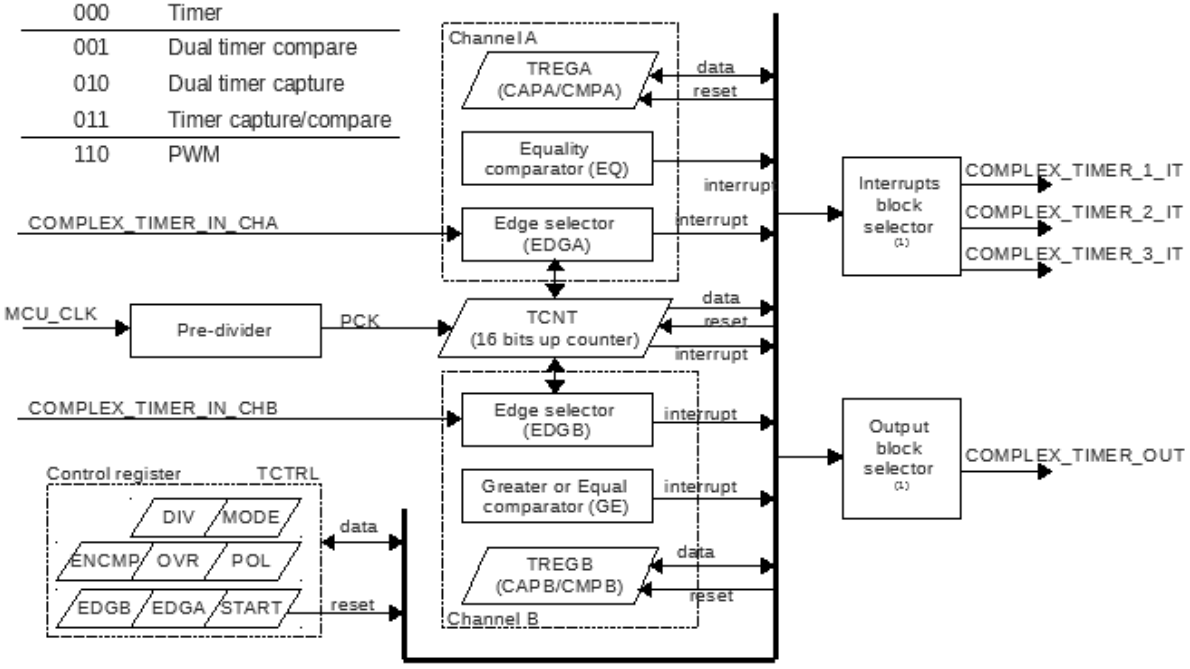


Figure 65 – Complex Timer block diagram

According to the selected mode the block generates one digital output signal and up to 3 interrupt signals.

A programmable and resettable 16-bit synchronous counter is the principal part of the Timer unit and a programmable pre-divider fixes the ratio between the clock of this counter and the input clock frequency.

For each exclusive mode, different parameters are programmable by software:

- In Single 16-bit auto-reload timer:
  - The pre-divider ratio between the MCU clock and the 16-bit timer clock
  - One value to define the period
- In Dual 16-bit timer compare:
  - The pre-divider ratio between the MCU clock and the 16-bit timer clock
  - The two values to be compared to the 16-bit timer value
- In Dual 16-bit timer capture:
  - The pre-divider ratio between the MCU clock and the 16-bit timer clock
  - The active edge of each channel:
    - Rising
    - Falling

        - Rising and falling
- In 16-bit timer compare and capture:
    - The pre-divider ratio between the MCU clock and the 16-bit timer clock
    - The value to be compared to the 16-bit timer value
    - The active edge of the capture channel
        - Rising
        - Falling
        - Rising and falling
- In High resolution 16-bit PWM (no shadow register):
    - The period of the PWM output signal
    - The duty cycle of the PWM output signal

The bits MODE in the Control register define which mode is selected by the Timer unit. All modes use a common set of hardware being mainly:

- A resettable 16-bit up counter, TCNT
- A clock pre-divider,
- Two 16-bit registers, TREGA and TREGB
- An equality comparator,
- A greater or equal comparator, and
- Two edge selectors

The two 16-bit registers, TREGA and TREGB, must be updated before any use.

Important:

- The update of the registers must be done when the control register bit START is low
- All modes are disabled if the bit START is low. The PWM output and all interrupts are frozen to 0
- The START and STOP bits are special
    - Writing any value with the START bit set will start the complex timer without changing the other bits
    - Writing any value with the STOP bit set will stop the complex timer without changing the other bits
    - These two bits have to be cleared to write the other bits. Writing both bits to one is not allowed

The following Table 117 shows the different modes of the complex timer:

Mode	Functiuon	Interrupt output signals			COMPLEX_TIMER_OUT
		COMPLEX_TIMER_1_IT	COMPLEX_TIMER_2_IT	COMPLEX_TIMER_3_IT	
000	Timer	-	-	INT_TIMER	-
001	Dual timer compare	INT_CMPA	-	INT_CMPB	-
010	Dual timer capture	INT_CAPA	INT_OVF or OVRA or OVRB	INT_CAPB	-
011	Timer capture/compare	INT_CMPA	INT_OVF or OVRB	INT_CMPB	-
110	PWM	INT_PWMA	-	INT_PWMB	OUT_PWM

other	Not used				

Table 117 — Complex Timer modes

COMPLEX_TIMER_IN_CHA and COMPLEX_TIMER_IN_CHB are respectivly the inputs of the channel A and B, they are used in modes Dual capture and Capture/Compare.
COMPLEX_TIMER_1_IT, COMPLEX_TIMER_2_IT and COMPLEX_TIMER_3_IT are the interrupt output signals. Assuming the control register bit START is high, an interrupt can be detected by an interrupt controller. Each interrupt last one MCU_CLK period. See Table 118 for details about interrupts signification.

COMPLEX_TIMER_2_IT is composed by 2 or 3 signals. Signal which generates the interrupt can be deduced from OVRA and OVRB value. If COMPLEX_TIMER_2_IT occurs and OVRA is set to 1 then the interrupt has been generated by an overrun on channel A. It is the same for OVRB. If COMPLEX_TIMER_2_IT occurs and OVRA and OVRB are set to 0 then the interrupt has been generated by a counter overflow. Be careful, this method works only if OVRA and OVRB are set to 0 before the interrupt occurs, they are cleared by reading TREGA and TREGB.

COMPLEX_TIMER_OUT is the digital output signal. His value is only defined in PWM mode.

Interrupt output	Interrupt description
INT_TIMER	Greater than or equality comparator (at the end of CNT > DATA_B)
INT_CMPA	Equality comparator (at the end of CNT = DATA_A)
INT_CMPB	Greater than or equal (at the end of CNT > DATA_B). Reset CNT if ENCMP is high
INT_CAPA	Capture signal for channel A. Active edge programmed with EDGA
INT_CAPB	Capture signal for channel B. Active edge programmed with EDGB
INT_OVRA	Overrun on Channel A. TREGA not read before the second capture
INT_OVRB	Overrun on Channel B. TREGB not read before the second capture
INT_OVF	Counter overflow. (CNT > FFFFh)
INT_PWMA	Indicates when OUT_PWM is set to the opposite of POL
INT_PWMB	Indicates when OUT_PWM is set to POL

Table 118 — Complex Timer Interrupt description

Port: CTIMER0…1

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word

Field name	Bit	R/W	Description

Address	Reset		Access
TREGB	[15:0]	RW	Timer Channel B

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word

Field name	Bit	R/W	Description
TREGA	[15:0]	RW	Timer Channel A

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0001	Word, Read Only

Field name	Bit	R/W	Description
TCNT	[15:0]	R	Counter value

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0002	Word, Byte

Field name	Bit	R/W	Description
DIV	[15:14]	RW	00: Timer Clock = MCU_CLK 01: Timer Clock = MCU_CLK/16 1X: Timer Clock = MCU_CLK/256
MODE	[13:11]	RW	Timer mode
ENCMP	10	RW	Enable reset control for the 16-bit up counter
OVRB	9	R	Overrun interrupt signal, clear after reading
OVRA	8	R	Overrun interrupt signal, clear after reading
POL	7	RW	Define the polarity of the PWM
PWMI	6	R	Read back PWM output
		W	Do nothing
EDGB	[5:4]	RW	Select edge sensitivity on input channel B
EDGA	[3:2]	RW	Select edge sensitivity on input channel A
STOP	1	W	1:Stop the complex timer, all other bits are discarded 0: No effect
		R	1: Complex timer is stopped 0: Complex timer is running
START	0	W	1:Start the complex timer, all other bits are discarded 0: No effect
		R	1: Complex timer is running 0: Complex timer is stopped

Table 119 – CTIMER0 and CTIMER1

The START and STOP bits are specials. Writing any value with the START bit set will start the complex timer without changing the other bits. Writing any value with the STOP bit set will stop the complex timer without

changing the other bits. These two bits have to be cleared to write the other bits. Writing both bits to one is not allowed.

14.3.8.1. Pre-divider block

When the pre-divider functionality is used, the control register DIV fix the pre-divider ratio between the MCU clock, MCU_CLK, and the 16-bit up counter clock, PCK.

DIV [1:0]	Fpck
0 0	Fck
0 1	Fck / 16
1 X	Fck / 256

Table 120 – Pre-divider ratio programming

Where: Fck is the frequency of the MCU_CLK and
Fpck is the frequency of the 16-bit up counter

14.3.8.2. 16-bit Auto-Reload Timer

This mode allows the block to generate interrupts at fixed intervals. It uses the following hardware:

- The resetable 16-bit up counter,
- The clock pre-divider,
- A 16-bit register, CMPB, and
- A greater or equal comparator

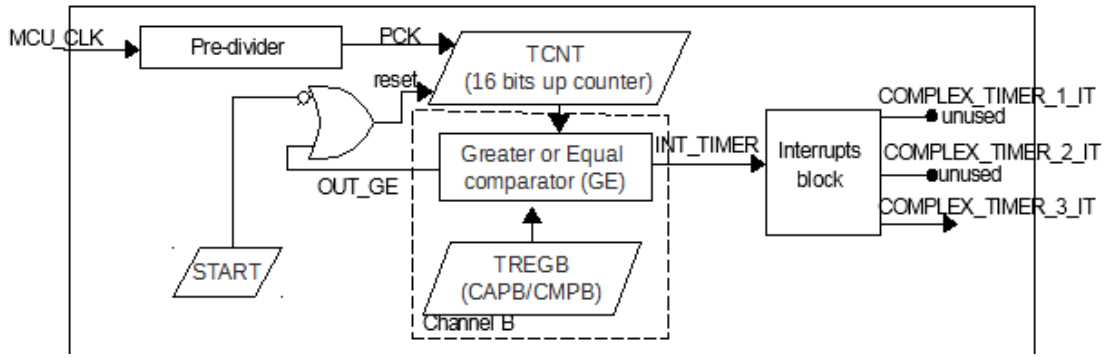


Figure 66 – Timer block diagram

The counter clock, PCK, is the MCU clock, MCU_CLK, divided by a predefined number, DIV, being 1, 16 or 256. The counter is incremented on PCK rising edge and a predefined comparator value, CMPB[15:0], is loaded in the 16-bit register TREGB.

The comparator output, OUT_GE, is set to 1 if the counter, TCNT, is greater than or equal to CMPB. And the counter is reset to '0001h' on the next MCU_CLK rising edge.

An interrupt, INT_TIMER, is generated:

- at the end of the period where the counter reaches the predefined comparator value, CMPB, or

- at the end of the period when a new comparator value, CMPB, greater than the current counter value, TCNT, is loaded in the 16-bit register.

Assuming **CMPB>0001h**, the period, Tint_timer, of the interrupt signal INT_TIMER is given by the following equations:

$$Fpck = \frac{\text{Fmcu\_clk}}{\text{DIV}}$$

$$Tint_{timer} = \frac{\text{DIV}}{\text{Fmcu}_{\text{clk}}} \times CMPB$$

$$Fint_{timer} = \frac{1}{\text{Tint\_timer}} = \frac{Fmcu\_ckl}{DIV} \times \frac{1}{CMPB}$$

Or, if the comparator value CMPB is extracted from the equation:

$$CMPB = \frac{\text{Fmcu\_clk}}{DIV} \times \frac{1}{Fint\_timer}$$

Where: F_{mcu_clk} is the frequency of the MCU clock,
Fpck is the frequency of the 16-bit up counter, and
DIV = 1, 16, or 256.

If **CMPB≤0001h** the interrupt signal, INT_TIMER is generated every PCK.

The different possibilities for the output signal are summarized in Table 121 – Timer output signal

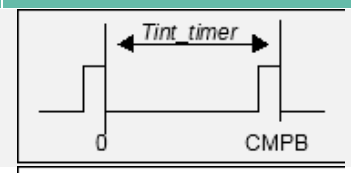
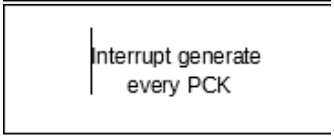
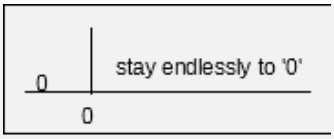
START	CMPB	INT_TIMER	NOTES
1	> 0001h		$Tint_{timer} = \frac{\text{DIV}}{\text{Fmcu\_clk}} \times CMPB$
1	≤ 0001h		- 16-bit up counter reset to 0001h
0	-		- Timer mode is disabled - 16-bit up counter reset to 0001h

Table 121 – Timer output signal

14.3.8.3. Dual 16-bit Timer Compare

This mode uses the following hardware:

- The resettable 16-bit up counter
- The clock pre-divider
- Two 16-bit registers, CMPA and CMPB
- The equality comparators
- The greater or equal comparator

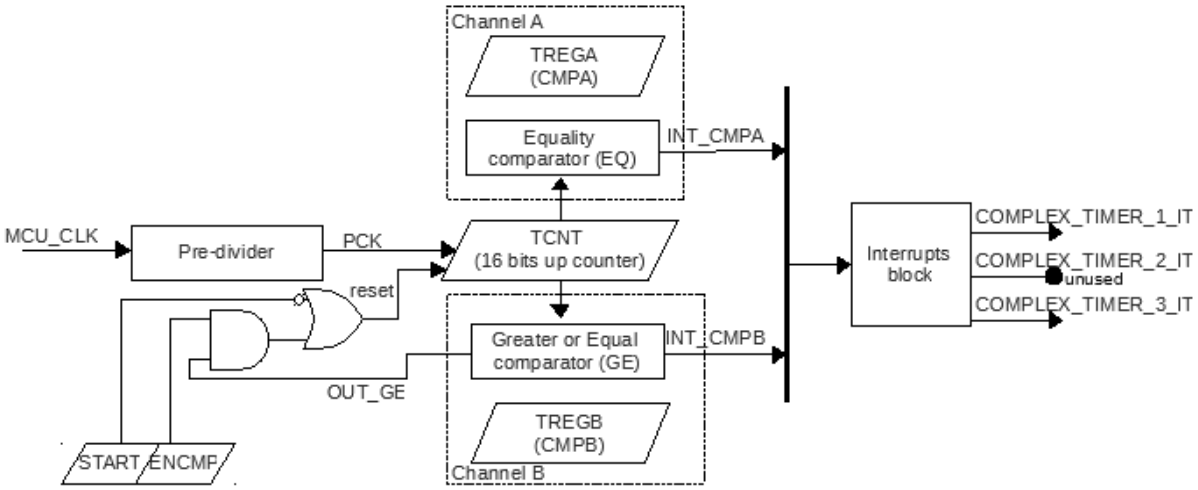


Figure 67 – Dual timer compare block diagram

This mode uses the 16-bit counter as a free running counter. The counter clock, PCK, is the MCU clock, MCU_CLK, divided by a predefined number, DIV, being 1, 16 or 256. The counter is incremented on the PCK rising edge.

Two predefined comparator values, CMPA[15:0] and CMPB[15:0], are loaded in the 16-bit registers.

The comparator output, OUT_GE, is set to 1 if the counter, TCNT, is greater than or equal to CMPB. And the counter is reset to '0001h' on the next PCK rising edge.

An interrupt, INT_CMPA, is generated at the end of the period when the counter value is equal to the predefined comparator value CMPA.

The programmed time for this interrupt signal, Tint1, is given by the following equation:

$$Tint1 = \frac{\text{DIV}}{\text{Fmcu\_clk}} \times CMPA$$

Where: F_{mcu_clk} is the frequency of the MCU clock and
DIV = 1, 16, or 256.

An interrupt, INT_CMPB, is generated:

- at the end of the period where the counter reaches the predefined comparator value, CMPB
- at the end of the period when a new comparator value, CMPB, greater than the current counter value, TCNT, is loaded in the 16-bit register

The programmed time for this interrupt signal, Tint2, is given by the following equation:

$$Tint2 = \frac{\text{DIV}}{\text{Fmcu\_clk}} \times CMPB$$

The reset of the 16-bit counter is controlled with the control register bits ENCMP and START.

Assuming bit START is high and CMPB>0001h

If ENCMP is set to 1, the 16-bit counter is reset when its value reaches the comparator value CMPB.

The programmed time for interrupts INT_CMPA and INT_CMPB is given by Equation 1 and Equation 2.

Then if the bit START is not set to 0 by MCU, the counting sequence is repeated indefinitely.

And the frequency at which the interrupt signals are generated is given by Equation 3.

$$Fint1 = Fint2 = Fout = \frac{\text{Fmcu\_clk}}{\text{DIV}} \times \frac{1}{CMPB}$$

Or, if the comparator value CMPB is extracted from the equation:

$$CMPB = \frac{\text{Fmcu\_clk}}{\text{DIV}} \times \frac{1}{Fout}$$

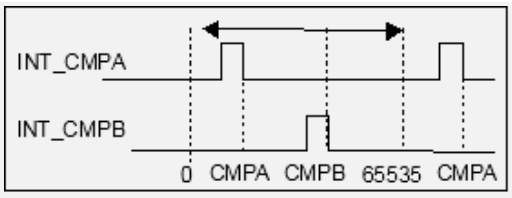
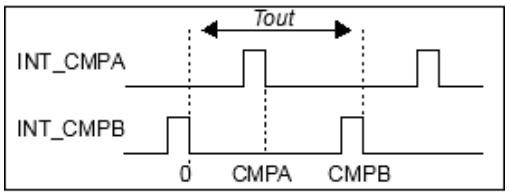
If ENCMP is set to 0, the 16-bit counter is reset when the maximum counting value 'FFFFh is reached.

The programmed time for interrupts INT_CMPA and INT_CMPB is given by Equation 1 and Equation 2. But if counter is running indefinitely, the frequency at which the interrupt signals are generated is given by:

$$Fout = \frac{1}{Tout} = \frac{\text{Fmcu\_clk}}{\text{DIV}} \times \frac{1}{65535}$$

Where: F_{mcu_clk} is the frequency of the MCU clock and DIV = 1, 16, or 256.

The different possibilities for the output signals are summarized in Table 122 – Timer compare output signals.

START	ENCMP	CMPB	CMPA	INT_CMPA / INT_CMPB	NOTES
1	0	-	-		
1	1	> 0001h	< CMPB		

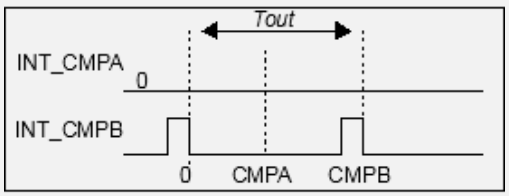
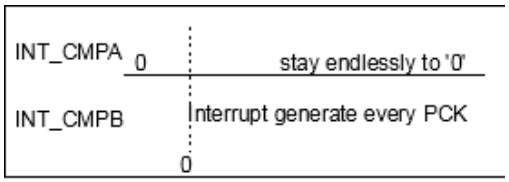
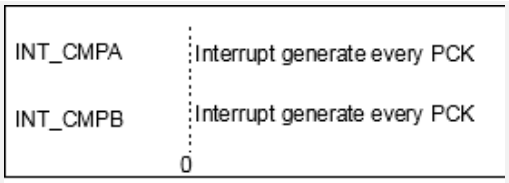
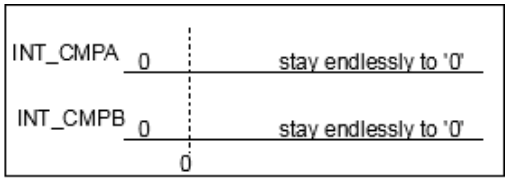
START	ENCMP	CMPB	CMPA	INT_CMPA / INT_CMPB	NOTES
			> CMPB		
1	1	≤ 0001h	<> 0001h	 INT_CMPB stuck at '1' if PCK = MCU_CLK	- 16-bit counter reset to 0001h
			= 0001h	 INT_CMPA and INT_CMPB stuck at '1' if PCK = MCU_CLK	
0	-	-	-		- Timer Compare mode disabled - 16-bit counter reset to 0001h

Table 122 – Timer compare output signals

14.3.8.4. Dual 16-bit Timer Capture

This mode uses the following hardware:

- The resettable 16-bit up counter
- The clock pre-divider
- Two 16-bit registers, CAPA and CAPB
- Two edge selectors

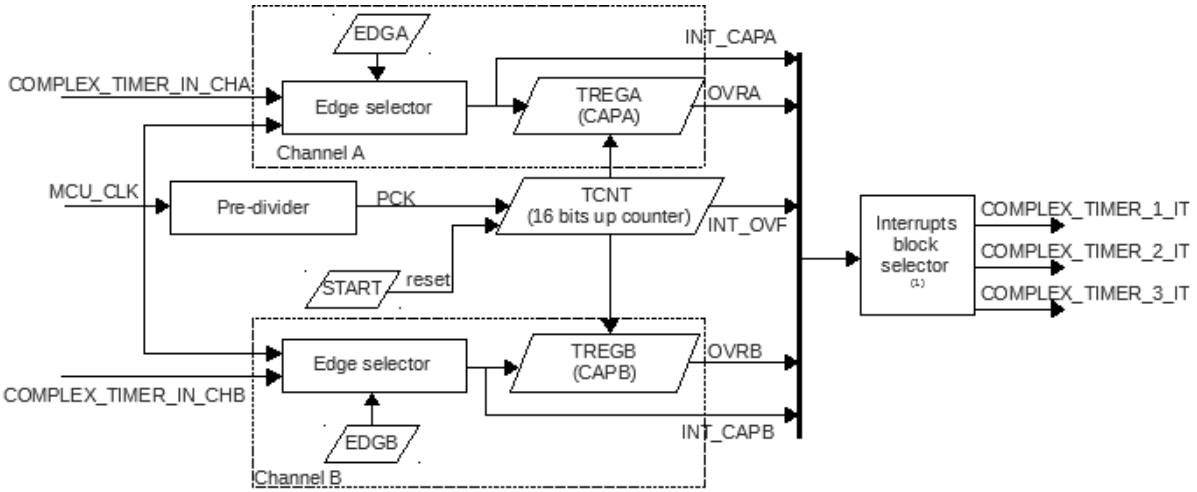


Figure 68 – Dual timer capture block diagram

This mode uses the 16-bit counter as a free running counter. The counter clock, PCK, is the MCU clock, MCU_CLK, divided by a predefined number, DIV, being 1, 16 or 256. The counter is incremented on the PCK rising edge.

The input signal, COMPLEX_TIMER_IN_CHA, is sampled by MCU clock, MCU_CLK, and when an event is detected on channel A:

- The content of the free-running counter is saved in the 16-bit register CAPA,
- An interrupt, INT_CAPA, is generated.

The edge selector can be programmed with the control register bit EDGA[1:0] to detect the following events: rising, falling, or rising and falling edges.

The input signal, COMPLEX_TIMER_IN_CHB, is also sampled by MCU clock MCU_CLK, and when an event is detected on channel B:

- The content of the free-running counter is saved in the 16-bit register CAPB,
- An interrupt, INT_CAPB, is generated.

The edge selector can be programmed with the control register bit EDGB[1:0] to detect the following events: rising, falling, or rising and falling edges.

COMPLEX_TIMER_2_IT is composed by 3 signals in this mode, INT_OVF, OVRA and OVRB as described in paragraph 14.3.8.

An interrupt INT_OVF is generated when the counter overflows, i.e. reaches the value 65535. Using this interrupt the counter lenght can be extended by software.
The interrupt signals, OVRA and OVRB, respectively controlled by channel A and B, are generated if two consecutive capture actions occur without reading the capture value in TREGA and TREGB. Reading these registers will reset OVRA and OVRB, respectively. If they remain on '1' all future capture events are masked until a read operation is performed.

The functionality of the Timer unit in Capture mode is summarized Table 123.

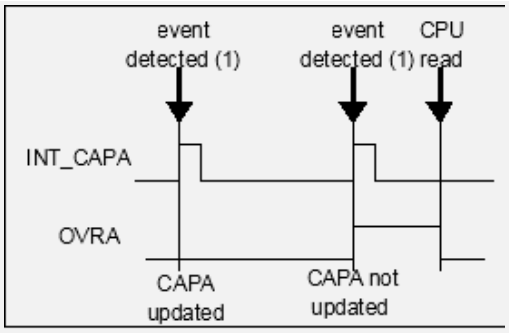
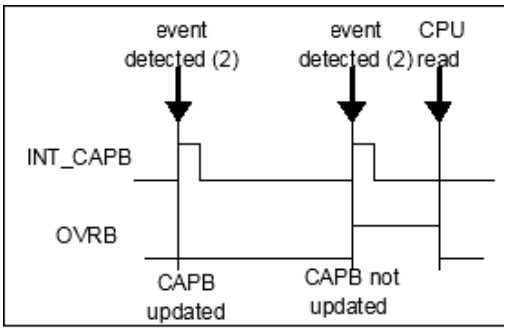
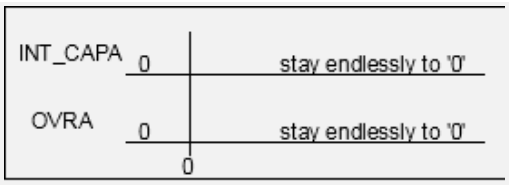
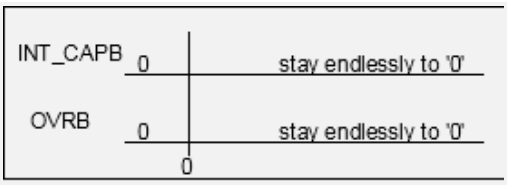
START	INT_CAPA / OVRA	INT_CAPB / OVRB	NOTES
1		-	(1) According to the programming of the edge selector EDGA
1	-		(2) According to the programming of the edge selector EDGB
0			- Timer Capture mode disabled - 16-bit counter reset to 0001h

Table 123 – Timer capture output signals

14.3.8.5. 16-bit Timer Compare and Capture

This mode uses the following hardware:

- The resettable 16-bit up counter
- The clock pre-divider
- Two 16-bit register, CMPA and CAPB
- The equality comparator
- An edge selector

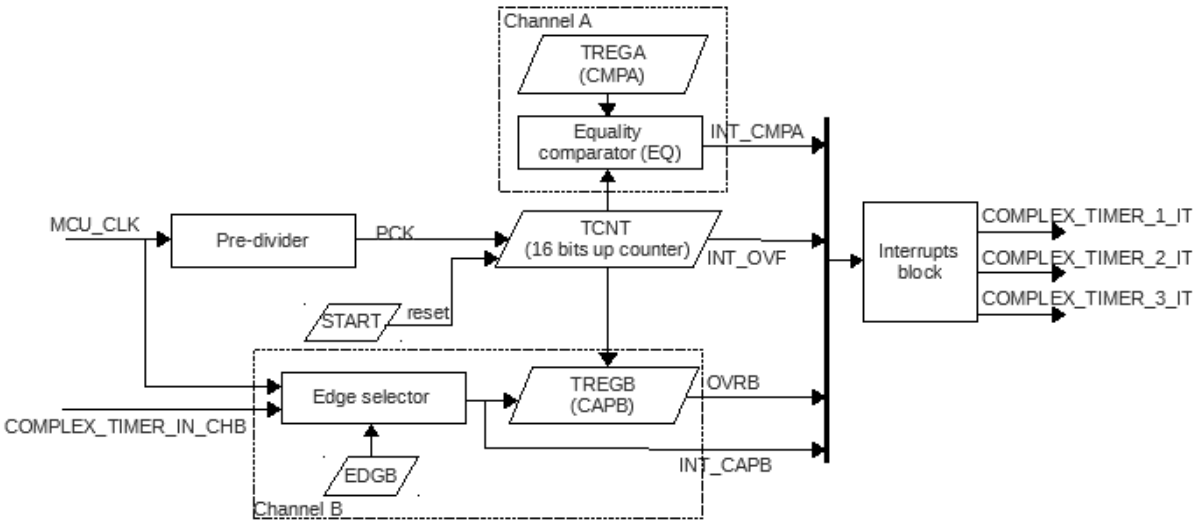


Figure 69 – Timer compare/capture block diagram

Compare Channel A

This mode uses the 16-bit counter as a free running counter. The counter clock, PCK, is the MCU clock, MCU_CLK, divided by a predefined number, DIV, being 1, 16 or 256. The counter is incremented on the PCK rising edge.

Assuming bit START is high,

the 16-bit counter is reset when its value reaches the maximum counting value 'FFFFh.

An interrupt, INT_CMPA, is generated at the end of the period when the counter value is equal to the predefined comparator value CMPA.

The programmed time for this interrupt signal, Tint1, is given by the following equation:

$$Tint1 = \frac{\text{DIV}}{\text{Fmcu\_clk}} \times CMPA$$

Where: Fmcu_clk is the frequency of the MCU clock, and

DIV = 1, 16, or 256.

Then if the bit START is not set to 0 by CPU, the counting sequence is repeated indefinitely. And the frequency at which the interrupt signals is generated is given by Equation 7 :

$$Fout = \frac{\text{Fmcu\_clk}}{\text{DIV}} \times \frac{1}{65535}$$

Capture Channel B

The input signal, COMPLEX_TIMER_IN_CHB, is sampled by MCU_CLK, and when an event is detected on channel input B:

- The content of the free-running counter is saved in the 16-bit register CAPB
- An interrupt, INT_CAPB, is generated

The edge selector can be programmed with the control register bit EDGB[1:0] to detect events caused by rising, falling, or rising and falling edges.

An interrupt, INT_OVF, is activated when the counter overflows. Using this interrupt the counter length can be extended by software.

The interrupt signal, OVRB, is generated if two consecutive capture actions occur without CPU reading operation in between. The first value memorized in CAPB register is not overwritten.

The functionality of the Timer unit in Compare/Capture mode is summarized in Table 124.

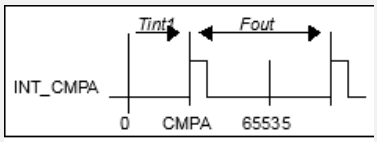
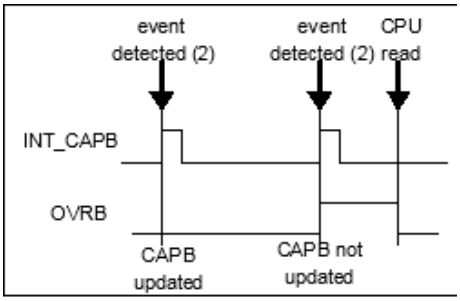
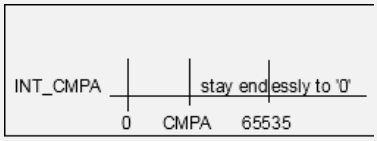
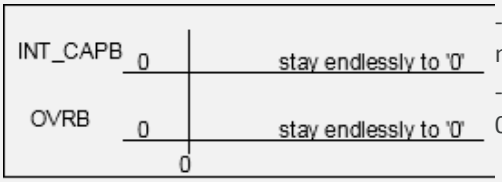
START	INT_CMPA	INT_CAPB / OVRB	NOTES
1		-	$Tint1 = \frac{\text{DIV}}{\text{Fmcu\_clk}} \times CMPA$ $Fout = \frac{\text{Fmcu\_clk}}{\text{DIV}} \times \frac{1}{65535}$
1	-		(2) According to the programming of the edge selector EDGB
0			- Timer Compare/Capture mode disabled - 16-bit counter reset to 0001h

Table 124 – Timer Compare/Capture output signals

14.3.8.6. High Resolution 16-bit PWM

This mode uses the following hardware:

- The resetable 16-bit up counter
- The clock pre-divider
- Two 16-bit registers, CMPA and CMPB
- The equality comparator
- The greater or equal comparator

It should be noted that the Timer in PWM mode differs from the PWM module itself. The Timer in PWM mode does not include all the functionalities of a PWM module (no double buffer register) and the register update must follow some rules. Please read all the sub-paragraphs before any use.

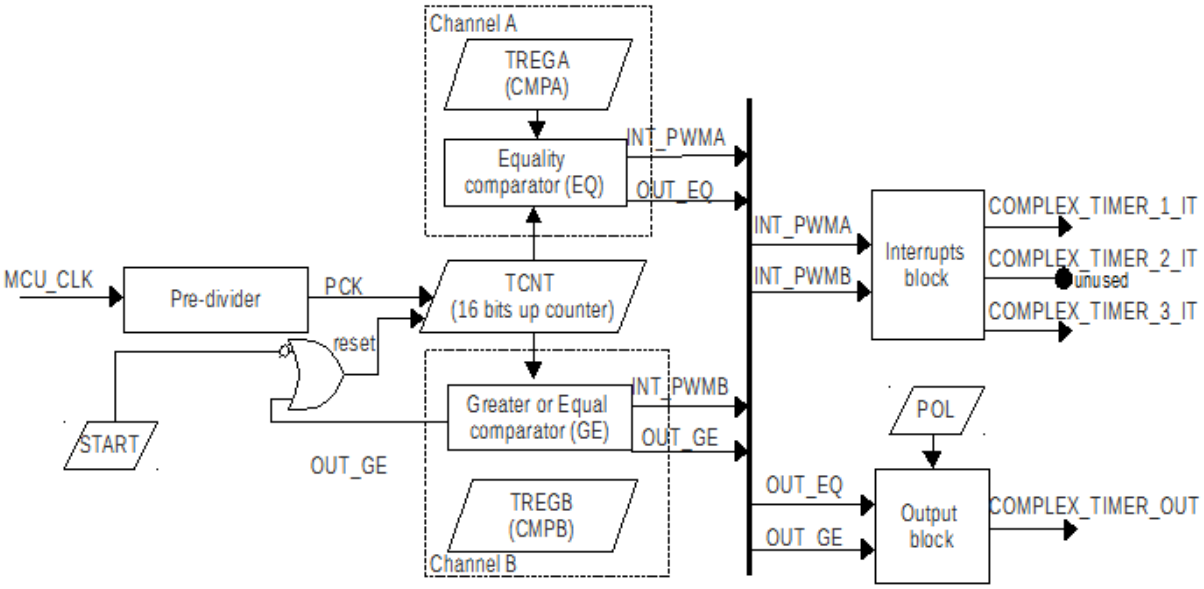


Figure 70 – 16-bit PWM block diagram

This mode allows creating a 16-bit high resolution PWM output.

The counter clock, PCK, is the MCU clock, MCU_CLK, divided by a predefined number, DIV, being 1, 16 or 256. The counter is incremented on the PCK rising edge.

Two predefined comparator values, CMPA[15:0] and CMPB[15:0] are loaded in the 16-bit registers. The comparator output signals, OUT_GE and OUT_EQ, associated with the control register bit POL, control the outputs waveform as described below:

- OUT_GE, is set to 1 if the counter, TCNT, is greater than or equal to CMPB
- OUT_EQ, is set to 1 when the counter, TCNT, is equal to CMPA

And:

- When OUT_EQ is high, the inverse of POL is put on COMPLEX_TIMER_OUT, and an interrupt is generated on the output INT_PWMA at the end of the period
- When OUT_GE is high, POL is put on COMPLEX_TIMER_OUT, and an interrupt is generated on the output INT_PWMB

Warning: The registers CMPA and CMPB are not buffered: when updated the new values take effect immediately on theoutput state.

Assuming **CMPB > 0001h** and **CMPA < CMPB** the PWM output period, Tpwm, and the duty cycle, DCpwm, are given by the equations:

$$Tpwm = \frac{DIV}{Fmcu\_clk} \times CMPB$$

$$Fpwm = \frac{1}{Tpwm} = \frac{Fmcu\_clk}{DIV} \times \frac{1}{CMPB}$$

$$CMPB = \left[\frac{Fmcu\_clk}{DIV} \times \frac{1}{Fpwm} \right]$$

$$DCpwm = \frac{(CMPB - CMPA)}{Tpwm}$$

$$DCpwm = \left[\frac{(CMPB - CMPA)}{CMPB} \right] \times \left[\frac{Fmcu\_clk}{DIV} \right]$$

Where: Fmcu_clk is the frequency of the MCU clock and
DIV = 1, 16, or 256.

The functionality of the Timer unit in PWM mode is summarized in Table 125.

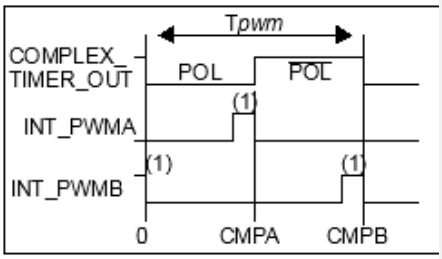
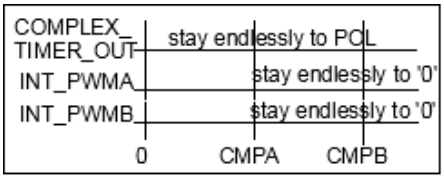
START	OUT_PWM	NOTES
1		$Tpwm = \frac{DIV}{Fmcu\_clk} \times CMPB$ - (1) Interrupt is generated on PCK - If CMPA > CMPB, COMPLEX_TIMER_OUT stays at the opposite of POL and INT_PMWA is never raised
0		- PWM mode disabled - 16-bit counter reset to 0001h

Table 125 – PWM output signal

COMPLEX_TIMER_OUT is equal to POL if:

- COMPB ≤ 0001h or
- COMPA > COMPB or COMPA = 0000h

14.3.9. PPM interface

14.3.9.1. Protocol description

PPM interface allows transmitting data regardless of a clock. The delay between the pulses isn't referenced from the edge of a clock but from the edge of the previous pulse.

Please see also [4] in chapter 7.

Here a symbol represents 2 bits of data. The 4 symbols used are the following:

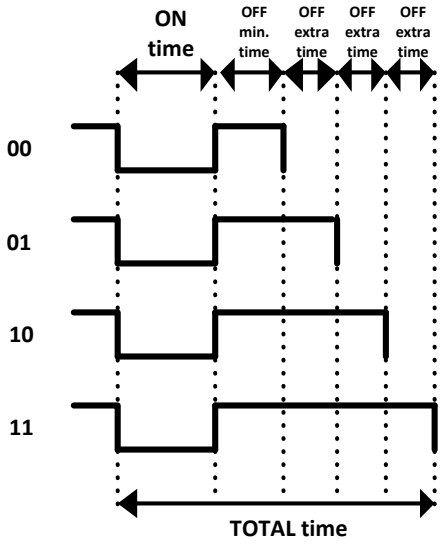


Figure 71 – Bit encoding

One characteristic of the DPPM is that the length of the encoded signal is not fixed. The shortest frame is the one containing only 0's and the longest the one containing only 1's.

Computations for FLASH page programming application

Assuming F_{mcu}=20MHz, ON time=3us, OFF min. time=2us and OFF extra time=1.5us

The bitrate varies between:

$$bitrate\,[\min] = \frac{2*1000}{3+2+(3*1.5)} = 210.5\ kbps \qquad bitrate\,[\max] = \frac{2*1000}{3+2} = 400\ kbps$$

The average bitrate is 275.9 kbps. It will take $16 * 1000/bitrate\ \max = 40us$ to transfer a 16 bits frame of 0's and $16 * 1000/bitrate\ min = 76us$ to transfer a 16 bits frame of 1's. With a 20MHz non-pipelined CPU this represents respectively 800 CPU cycles and 1520 CPU cycles.

The average time to program a FLASH page (1 page = 64 frames = 1024 bits) is then:

$1024 * 1000/bitrate\ avg = 3.71ms$ (lower than the 5ms maximum specification)

At the beginning of a communication the 4 symbols will be sent in a predefined order and the corresponding times (computed by the PPM timer) will be saved in a LUT by the SW. This will then allow the SW to decode the "useful" data (command + payload) by comparing the following PPM timer values with the LUT content.

14.3.9.2. Integration in MLX products

The PPM timer implementation follows strict rules in order to allow integration into the digital platform. The block has direct (read and write) access to the RAM thanks to a DMA master (no need for CPU intervention). It is very similar (except for the core functionality) to peripherals like the SPI.

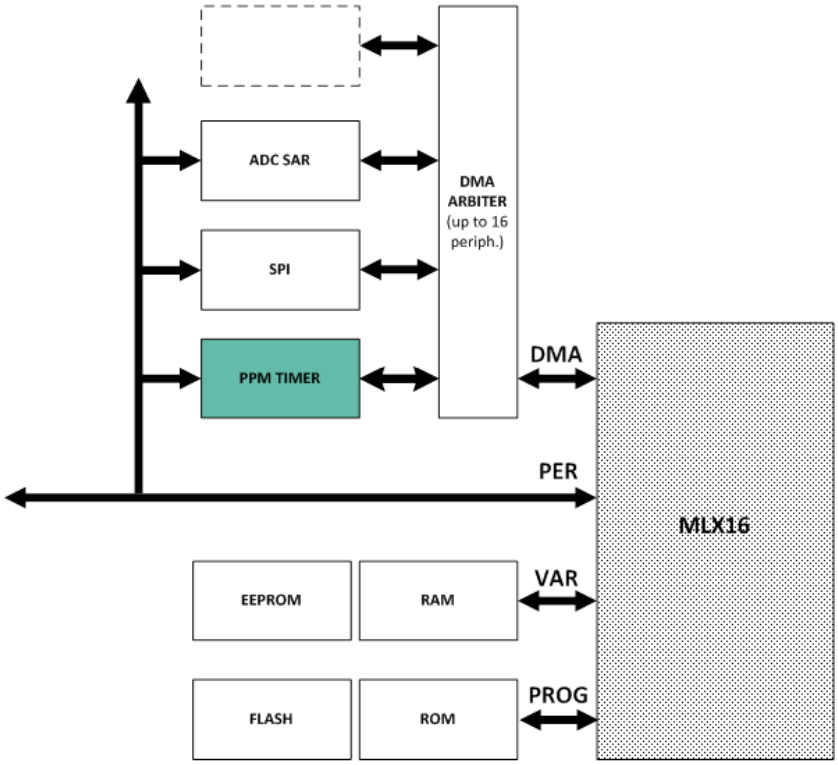


Figure 72 – PPM timer integration in MLX products

14.3.9.3. Block diagram

The figure below provides a high level view of the PPM timer implementation and the interactions between the blocks composing it.

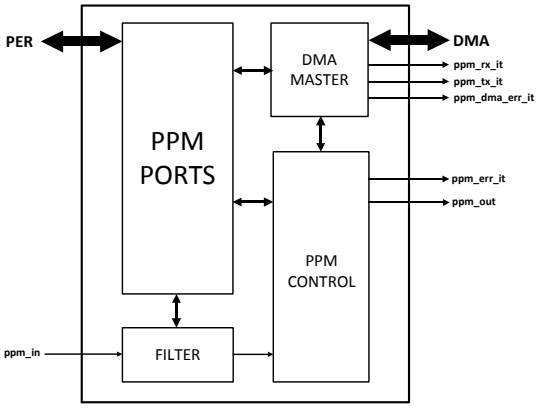


Figure 73 – PPM timer block diagram

14.3.9.4. PPM control

The PPM timer can be configured as a receiver (RX) or as a transmitter (TX). These 2 modes and their implementations are described in the following sections.

14.3.9.4.1. RX mode

In RX mode the PPM control block is responsible for computing input edge to edge times and making them available to the DMA master for storage in memory and for later decoding by the SW. The control bloc can be configured to compute and store TOTAL symbol times only (RX0 mode) or both ON and TOTAL symbol times (RX1 mode).

The RX part implementation is composed of an edge detector on the PPM (filtered) input, a 16 bits counter, a 16 bits buffer and some logic for interrupt generation and error conditions flagging.

The counter needs to be enabled by SW (PPM_EN=1) and is reset on either rising edge (RX0) or every edge (RX1) of the input signal. At reset the last counter value is saved in the PPM buffer. Every time the buffer is updated a write access in memory is automatically triggered (taken care of by the DMA master).

An error interrupt **ppm_err_it** is triggered in case one of the following conditions happens:

- the buffer receives a new data while a DMA access is still ongoing (RCVF error flag)
- the timeout is reached before the next edge can be detected (TOUT error flag)

After receiving an error interrupt the associated flags can be read back. The SW must then stop the PPM timer by setting PPM_EN=0 (this will automatically clear the flags) and restart it afterwards.

The error interrupt is level 3-6 programmable.

14.3.9.4.2. TX mode

In TX mode the PPM control block is responsible for reading edge to edge times stored in memory and for recreating the PPM signal. The control bloc can be configured either in TX0 mode where the ON time is fixed and read from a port (see **ON_TIME[15:0]** setting) and the TOTAL time is read from memory or in TX1 mode where both ON and TOTAL times are read from memory. In the latest mode the ON time can then vary but the data occupies twice more memory space.

The TX part implementation uses the same 16 bits counter and 16 bits buffer than the RX part. An additional 16b register is needed for TX1 mode. Some logic is responsible for interrupt generation and error conditions flagging.

In TX0 mode a read access in memory is triggered at address **TBASE_ADD[15:0]** as soon as the bloc is enabled by the SW (**PPM_EN=1**). If the access is successful the data is stored in the PPM buffer and the counter is started. To create the PPM signal (**PPM_OUT**) the counter value is first compared to the **ON_TIME[15:0]** stored in the port and then to the TOTAL time stored in the PPM buffer. When the TOTAL time is reached a new count is started and a new read access in memory at the next address is triggered.

In TX1 mode two consecutive read accesses are triggered at address **TBASE_ADD[15:0]** and at the next address as soon as the bloc is enabled by the SW (**PPM_EN=1**). The ON time (first read) is stored in the PPM ON buffer and the TOTAL time (second read) is stored in the PPM buffer. If the two accesses are successful then the counter is started. To create the PPM signal (**PPM_OUT**) the counter value is first compared to the ON time stored in the PPM ON buffer and then to the TOTAL time stored in the PPM buffer. When the TOTAL time is reached a new count is started and two new consecutive read accesses in memory at next addresses are triggered.

The total number of “Words” to be read is set by the **DMA_LEN[3:0]** setting. Once the last word has been sent no more memory access is done and an interrupt **ppm_tx_dat** is generated in order to notify the SW which in turn has to set PPM_EN=0. The line then goes back to ‘1’ in case RE_FE=0 or to ‘0’ in case RE_FE=1.

An error interrupt **ppm_err_it** is triggered in case the TOTAL time read from RAM is greater than the programmed **TIMEOUT[15:0]** (same counter compare than in RX mode).

14.3.9.5. DMA master

The PPM block includes a DMA master which allows writing and reading messages of up to 128 "Words" in memory without CPU intervention. The length of the message is configured thanks to **DMA_LEN[3:0]** setting.

The first “Word” of the message is written (or read) at address **RBASE_ADD** (or TBASE_ADD) in memory. The address is then incremented automatically for the next “Words”. The address is finally reinitialized to **RBASE_ADD** (or TBASE_ADD) once the full message has been written (or read).

In RX mode a DMA write access is triggered every time the PPM buffer is updated. Every time 1, 2, 4 or 8 PPM data have been saved in memory (see **DAT_IT_NB[1:0]** option) an interrupt **ppm_dat_it** is triggered in order to tell the SW that it can start decoding the PPM data.

In TX mode the PPM control block requests data to the DMA master which in turn triggers read accesses in RAM.

DMA of the PPM is user mode only. Therefore it cannot address any system protected area or port.

The PPM timer has the lowest priority among DMA peripherals. It is user’s responsibility to ensure that SPI, ADC,… are not running at the same time than the PPM timer.

If the PPM timer tries to access a non-existing area (for example port 0x0000 in case RBASE or TBASE is not set) or try to write any non-writable zone (like ROM) or protected zone (like Flash), then the DMA access will be inhibited and a DMA_ERR interrupt (level 1) will be triggered.

Please note that the occurrence of bit **PPM_DMA_ERR** (set by DMA_ERR_IT) will prevent any other DMA access by the PPM timer before the block has been disabled and then re-enabled by the SW.

14.3.9.6. PPM Ports

14.3.9.6.1. DMA RX base address port

Port: PORT_PPM_RBASE_ADD

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
PPM_RBASE_ADD	[15:0]	RW	Base address of the PPM DMA message in RX mode

Table 126 — PORT_PPM_RBASE_ADD

This register defines the address in memory of the first "Word" of a DMA message in RX mode. The following "Words" are written at the next addresses (address incremented by 2).

14.3.9.7. DMA TX base address port

Port: PORT_PPM_TBASE_ADD

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
PPM_TBASE_ADD	[15:0]	RW	Base address of the PPM DMA message in TX mode

Table 127 — PORT_PPM_TBASE_ADD

This register defines the address in memory of the first "Word" of the DMA data in TX mode. The following "Words" are read at the next addresses (address incremented by 2).

14.3.9.8. Timeout port

Port: PORT_PPM_TIMEOUT

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
PPM_TIMEOUT	[15:0]	RW	Timeout value

Table 128 — PORT_PPM_TIMEOUT

A **ppm_err_it** error interrupt will be generated if no edge is detected before the programmed timeout value (valid for both RX and TX modes).

14.3.9.9. PPM control port

Port: PORT_PPM_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:14]		Not used
PPM_IN_SEL	[13:11]	RW	RX mode input selection 000 : LIN_XRX 001 : IO_IN[0] 010 : IO_IN[1] 011 : IO_IN[2]

100 : IO_IN[3]
101-111: 1 (instead of 0, recessive level of LIN)

PPM_TX_RX	10	RW	0: RX mode 1: TX mode
PPM_DAT_IT_NB	[9:8]	RW	Program ppm_dat_it to trigger every $2^{DAT\_IT\_NB}$ PPM data saved in RAM
PPM_MODE	7	RW	0: TOTAL times only are stored in RAM 1: Both ON and TOTAL
PPM_FILT_BYP	6	RW	Input filter bypass
PPM_DMA_LEN	[5:2]	RW	DMA buffer length = (DMA_LEN+1) * PPM data 1 PPM data = 8 “Words” if PPM_MODE=0 1 PPM data = 16 “Words” if PPM_MODE=1
PPM_RE_FE	1	RW	1: time reference = PPM input rising edge 0: time reference = PPM input falling edge
PPM_EN	0	RW	1: PPM timer enabled 0: PPM timer disabled

Table 129 — PORT_PPM_CTRL

The PPM timer is disabled at reset. Setting PPM_**EN** enables it.

After reset the PPM timer is set to work on input falling edges. Setting PPM_**RE_FE** allows switching to rising edge.

Note: PPM_RE_FE setting doesn’t affect RX1 mode operation

PPM_TX_RX option allows setting the timer either for data decoding (RX, default) or data encoding (TX).

PPM_MODE defines the type of data saved (RX) or expected (TX) in memory. If set to 0 only TOTAL times (full symbol length = symbol ON time + symbol OFF time) are saved/expected in memory. If set to 1 both symbol ON times (start of symbol until first edge) and symbol TOTAL times are saved/expected in memory.

Combining PPM_**TX_RX** and **PPM_MODE** settings we get 4 functional modes for the PPM timer:

PPM modes	TX_RX=0	TX_RX=1	Description
PPM_MODE=0	RX0	TX0	TOTAL times only
PPM_MODE=1	RX1	TX1	ON and TOTAL times

Table 130 — PPM functional modes

The DMA buffer length can be set with PPM_**DMA_LEN[3:0]**. It defines the number of “Words” to be saved (RX) or read (TX) in memory. A maximum number of 256 “Words” can be stored in memory meaning 32 PPM data if PPM_MODE=0 and 16 PPM data if PPM_MODE=1.

The input filter can be disabled by setting **PPM_FILT_BYP** high.

PPM_DAT_IT_NB[1:0] setting allows to program how often the **ppm_rx_it** interrupt triggers. It can be every time 1, 2, 4, 8 PPM data have been saved in memory.

Note: setting 3 doesn't make sense when PPM_MODE is set to 1 (interrupt triggers when RAM buffer is full)

These settings should not be changed on the fly i.e. when **PPM_EN=1**. They have to be set upfront or latest when enabling the timer.

14.3.9.10. PPM buffer data port

Port: PORT_PPM_BUF_DATA

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
PPM_BUF_DATA	[15:0]	R	PPM buffer data

Table 131 — PORT_PPM_BUF_DATA

In RX mode reading this port allows to get the latest time value computed between 2 edges on the input signal (see RE_FE & PPM_MODE settings). In TX mode the port contains the latest value read from memory.

14.3.9.11. TX0 mode ON time

Port: PPM_TIMER

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
PPM_ON_TIME	[15:0]	RW	PPM On Time for TX mode

Table 132 — PPM_TIMER

In TX0 mode (see PPM control port) the ON time is fixed and is equal to the value set in this register. In TX1 mode this register is also (re)used to store data read from the RAM via DMA.

14.3.9.12. PPM flags port

Port: PORT_PPM_FLAGS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**

-	[15:4]		Not used
PPM_DMA_ERR	3	R	set if the DMA tries to access a non-existing or protected address
PPM_TOUT	2	R	counter timeout flag (both RX & TX modes)
PPM_RCVF	1	R	RX: receive buffer overflow TX: end of transmission
PPM_DMA_OP	0	R	1: DMA access (R/W) on going 0: no DMA access

Table 133 — PORT_PPM_FLAGS

PPM_DMA_OP flag is set high when a DMA access (read or write) is ongoing.

PPM_RCVF flag is set when:
- RX mode: the PPM buffer receives a new data while the previous one hasn't been saved in memory yet (DMA access on going)
- TX mode: at the end of the transmission

PPM_TOUT flag is set when the programmed timeout counter value is reached (both in RX and TX modes).

14.3.10. SPI interface

14.3.10.1. Features

The MLX81346 comprises one Serial Peripheral Interface block, SPI, with the following characteristics:

- supports word or byte transmission/reception
- full-duplex operation
- double-buffered architecture for queued transmission and continuous reception
- master and slave mode
- wide range of programmability
- SPI receiver and transmitter interrupt sources
- DMA access for multi words messages
- Communication Error detection

The SPI can work in one of the following main modes:

- Master mode, or
- Slave mode

14.3.10.2. Block diagram

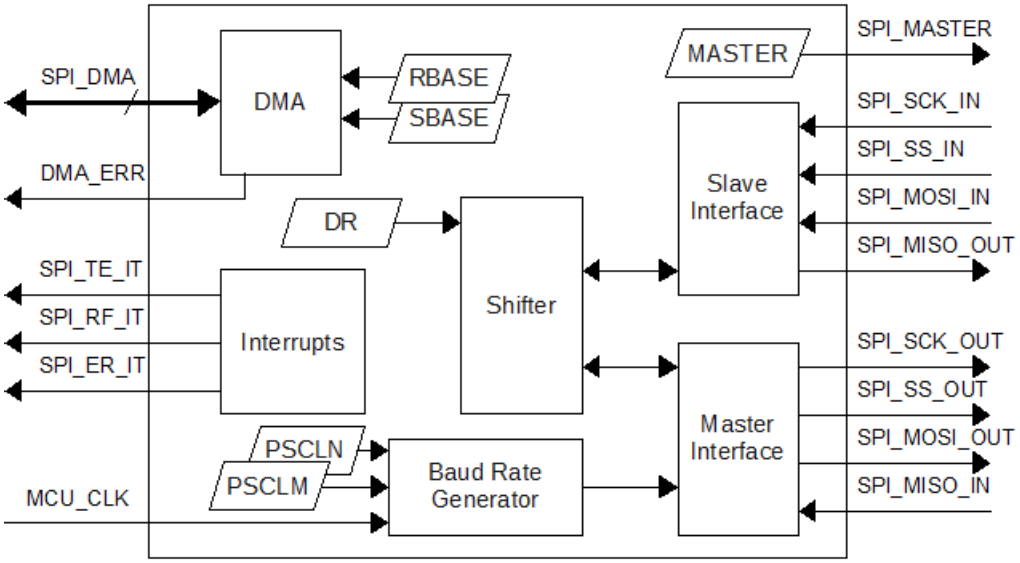


Figure 74 — SPI block diagram

The SPI bus is a synchronous serial protocol on 4 pins allowing to exchange "W*ords*" in full duplex mode. A "W*ord*" can be 8 bits or 16 bits.

This block can work as a master, initiating the communication by controlling the clock, SPI_SCK_OUT, the slave select SPI_SS_OUT and the master data signal SPI_MOSI_OUT or as a slave waiting for a communication request and driving only the slave data pin SPI_MISO_OUT. The SPI_MASTER output allows to know if the block is working as a slave or as a master allowing to multiplex the slave and the master signals on only 4 pins.

In master mode, the SPI clock can be configured using the baud rate generator. In slave mode, the clock is directly coming from the SPI interface and can be up to the frequency of the MCU clock (MCU_CLK).

The block generates three interrupts. Interrupt SPI_TE_IT is generated when the transmit register is empty, which happens at the beginning of the transmission as it starts by a transfer of the transmit register in the shifter. The CPU can write a new value during the transmission. When the SPI is stopped (by setting the STOP bit in the Control port), while transmit register is not empty, the SPI_TE_IT will be generated even if the SPI is stopped. Interrupt SPI_RF_IT is generated when the receive register is full. The CPU can read it. During this time, it is possible to receive a new "W*ord*" in the shifter. Last, interrupt SPI_ER_IT is generated when an error or an overflow is detected during the transmission.

The SPI block includes a DMA master, allowing receiving or sending a message of up to 64 "W*ords*" without needing the intervention of the CPU. When DMA mode is enabled, the transmit interrupt is generated when the last "W*ord*" is transmitted. The received interrupt is generated when the last "W*ord*" has been written in the memory.

14.3.10.2.1. Master mode

In master mode, pins SPI_SCK_IN, SPI_SS_OUT, SPI_MOSI_IN and SPI_MISO_OUT are not used. The connection to the outside world is shown in Figure 75. The loop-back from SPI_SS_OUT to SPI_SS_IN is mandatory. It is used to detect errors on the physical net SS.

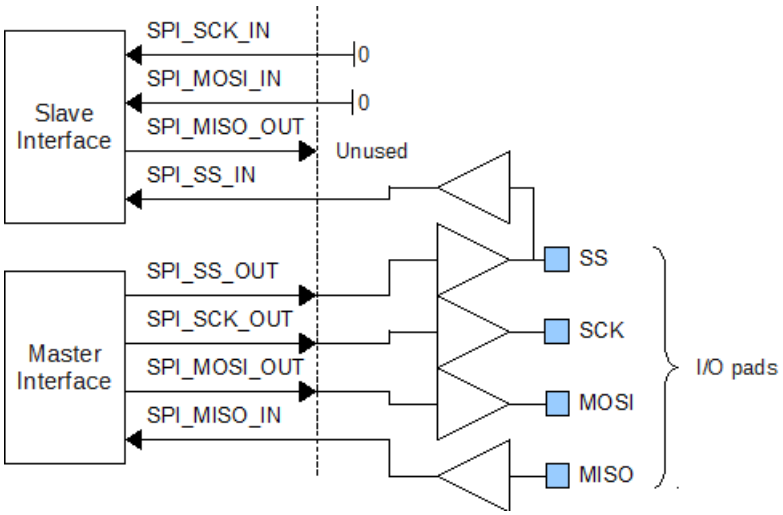


Figure 75 — SPI connections in Master mode

14.3.10.2.2. Slave mode

In slave mode, pins SPI_SCK_OUT, SPI_SS_OUT, SPI_MOSI_OUT and SPI_MISO_IN are not used. The connection to the outside world is shown in Figure 76.

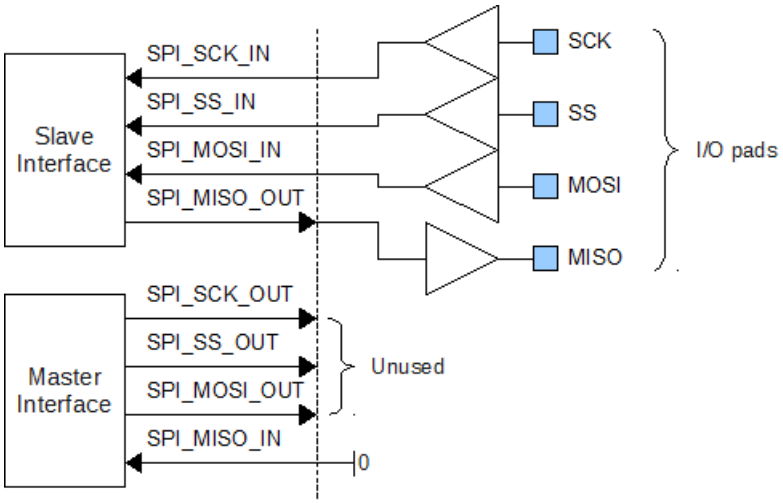


Figure 76 — SPI connections in Slave mode

14.3.10.3. SPI IO ports

14.3.10.3.1. Data port

Port: PORT_SPI_DATA

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
DR	[15:0]	W	Write to transmit register and Send data
		R	Read the receive register

Table 134 — SPI Data port

Writing in this port write the transmit register. If the block is configured as a master, the transmission starts. If the block is configured as a slave, the transmit register will be shifted out under control of the master. Reading this port reads the receive register.

14.3.10.3.2. DMA send base address port

Port: PORT_SPI_SBASE

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte
Field name	**Bit**	**R/W**	**Description**
SBASE	[15:0]	RW	Base address of data to send

Table 135 — SPI DMA send base address port

This register defines the address of the first "W*ord*" transmitted in a message. The following "W*ords*" are read at the following addresses. If the block is configured to work with 8 bits "W*ords*", the address is incremented by 1, else it is incremented by 2. SBASE must be word aligned in case block is configured to work with 16 bits "W*ords*", else an error (DMA_ERR) will be generated.

14.3.10.3.3. DMA receive base address port

Port: PORT_SPI_RBASE

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x0000	Word, Byte

Field name	Bit	R/W	Description
RBASE	[15:0]	RW	Base address of received data

Table 136 — SPI DMA receive base address port

This register defines the address of the first "W*ord*" received in a message. The following "W*ords*" are written at the following address. If the block is configured to work with 8 bits "W*ords*", the address is incremented by 1, else it is incremented by 2. RBASE must be word aligned in case block is configured to work with 16 bits "W*ords*", else an error (DMA_ERR) will be generated.

14.3.10.3.4. DMA control port

Port: PORT_SPI_DMA_CTRL

Address	Reset	Access
See Table 106 – MLX81346 ports overview	0x003F	Word, Byte

Field name	Bit	R/W	Description
-	[15:6]		Not used
MSGLEN	[5:0]	RW	Number of Words exchanged in DMA (0 means 64 Words)

Table 137 — SPI DMA control port

This port defines the length of the DMA message.
In this mode, the block can send and receive a message of up to 64 "*Words*". It sends data directly read from memory at the address defined by SBASE and writes received data directly in the memory at the address defined by RBASE.
The number of "*Words*" exchanged is (MSGLEN + 1)[1]. A "*Word*" can be 8-bits or 16-bits depending on the value of the bit BYTE_MODE (see SPI control port).
For a master, a transmission is still triggered by writing in the data port (DR) but the value is overwritten by the data from the memory at SBASE address.

14.3.10.3.5. Master baud rate port

Port: PORT_SPI_MASTER_BAUD_RATE

Address	Reset		Access
See	0x00FF		Word, Byte, Invalid

Address	Reset		Access
Table 106 – MLX81346 ports overview			
Field name	**Bit**	**R/W**	**Description**
-	[15:8]	R	Not used
PSCLM	[7:4]	RW	Divide MCU_CLK frequency by (PSCLM + 1)
PSCLN	[3:0]	RW	Divide MCU_CLK frequency by 2PSCLN, with 0 ≤ PSCLN ≤ 11

Table 138 — SPI Master baud rate port

This port defines the SPI master clock frequency (when SPI block is used as a master).

14.3.10.3.6. SPI Control port

Port: PORT_SPI_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x0000		Word, Byte, Invalid
Field name	**Bit**	**R/W**	**Description**
MODF	15	R	Mode fault error
		W	Write a 1 to clear bit
OVRF	14	R	Receive overflow interrupt
		W	Write a 1 to clear bit
RF	13	R	SPI Receive register full
TE	12	R	SPI Transmit register empty
-	[11:10]		Not used
RX_EN	9	RW	Enable errors due to reception
TX_EN	8	RW	Enable errors due to transmission
DMA	7	RW	Enable DMA mode
CPHA	6	RW	Clock phase: 0: Get data on the leading edge, set data on the trailing edge 1: Set data on the leading edge, get data on the trailing edge
CPOL	5	RW	Clock polarity: 0: Clock wait state at 0 1: Clock wait state at 1
BYTE_MODE	4	RW	Word size: 0: 16-bit 1: 8-bit
SS_FORCE	3	RW	SPI_SS behaviour (Master mode): 0: SPI_SS = 1 between Words, SPI_SS = 0 while transmitting 1: SPI_SS always at 0
MASTER	2	RW	Master/Slave: 0: Slave mode (reception)

Address	Reset		Access
			1: Master mode (emission)
STOP	1	W	0: No effect 1: Stop the SPI, all other bits are discarded
		R	0: SPI is running 1: SPI is stopped
START	0	W	0: No effect 1: Start the SPI, all other bits are discarded
		R	0: SPI is stopped 1: SPI is running

Table 139 — SPI Control port

The SPI is stopped at reset. Setting START starts the SPI. Setting STOP stops the SPI. When either START or STOP is set all other bits are **NOT** written. Setting both START and STOP (abnormal situation) starts the SPI. Setting START while already started has no effect, setting STOP while already stopped has no effect. All other bits can only be written if START = STOP = 0.
MASTER defines if the block is in master or slave mode. In master mode it initiates transmission; in slave mode it is waiting for the master to initiate the transmission.
SS_FORCE is used to define if the slave select signal should be kept active between transmission or not.
BYTE_MODE defines if the block is sending and receiving 8 bits or 16 bits "*Words*".
CPOL and CPHA define if the clock is active high or low and on which edge of the clock the data are sent and received.
DMA enables DMA mode.
TX_EN and RX_EN enable the generation of errors due to transmission or reception. They are used to disable errors when the SPI is used only in reception or only in emission.
TE is high if the transmit register is empty.
RF is high if the receive register is full.
OVRF is high if there is an overflow. It happens if the block receive a new data while the CPU has not read the receive register or if the block has to transmit a new data while the CPU has not written a new value. It is cleared by writing 1 in it.
MODF is high when there is an error during the transmission like additional or missing clock pulses. It is cleared by writing 1 in it.

14.3.10.4. Functional description

14.3.10.4.1. Initialization

The SPI block is started by setting START and stopped by setting STOP in the CONTROL port. As other bits of CONTROL port are not changed when START or STOP are set, they must be set before starting the block.

Trying to change the bits CPHA, CPOL, BYTE, SS_FORCE, MASTER and DMA when the SPI block is started generates an invalid address error.

Setting the STOP bit while a communication is running aborts the transmission. If the block is configured as a master, this will drive SPI_SS to 1 unless SS_FORCE is set and will stop the clock in its idle state.

14.3.10.4.2. Frequency control

In master mode, the SPI clock is driven by a programmable hardware divider generating the clock from the main clock.

The SPI clock frequency is

$$F_{sck} = \frac{F_{MCU\_CLK}}{(PSCLM+1)} \times \frac{1}{2^{PSCLN}}$$

with

- F_{sck} is the SPI master clock frequency.
- F_{MCU_CLK} is the frequency of the MCU clock.
- PSCLM is the port value between 0 and 15.
- PSCLN is the port value between 0 and 11

All changes in the Baud rate port must be done while the SPI bock is stopped, else an invalid address interrupt is generated and the write in not done.

There is no restriction on the value of PSCLM which can go from 0 to 15, whereas PSCLN has restriction and can go from 0 to 11. If PSCLM and PSCLN are equal to 0, the SPI will run at the frequency of the CPU clock.

The duty cycle of the SPI clock is always 50% if one of these conditions is true:

- MCU_CLK clock has a duty cycle of 50%
- PSCLN ≠ 0
- PSCLM is odd (so PSCLM+1 is even)

Else the duty cycle of the SPI clock with CPOL = 0 is given by the following formula

$$D_{sck} = \left| CPOL - \frac{\frac{PSCLM}{2} + D_{MCU\_CLK}}{(PSCLM+1)} \right|$$

with

- D_{sck} is the SPI master clock duty cycle.
- D_{MCU_CLK} is the duty cycle of the MCU clock.
- PSCLM is the port value between 0 and 15
- CPOL is the polarity of SCK

In slave mode, the SPI block is running with the clock of the master. The SPI can run at the same frequency than the MCU but the delay in the pads could lower this frequency.

14.3.10.4.3. Data format

The SPI can send and receive data in various formats.

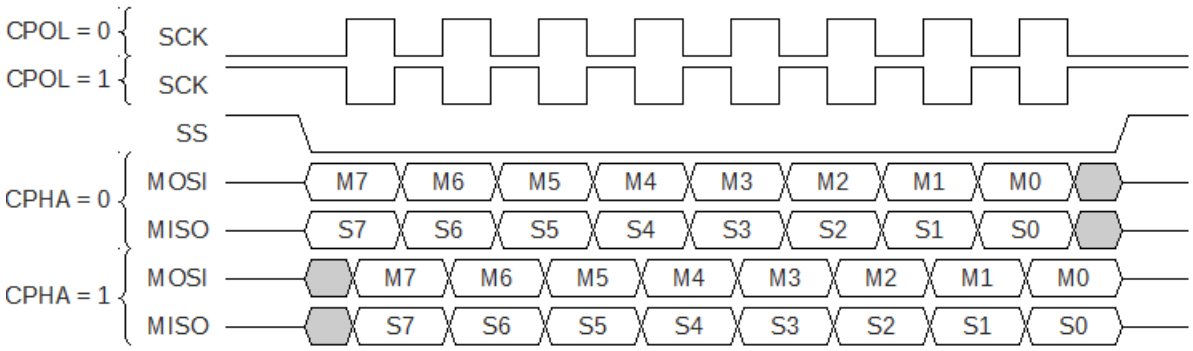


Figure 77 — SPI Data format

In addition to setting the clock frequency, the master can configure the clock polarity (CPOL) and the phase of the data (CPHA).

The clock polarity is configured using the CPOL bit. If CPOL is 0, the clock is active high, so its starts and stops at 0. If CPOL is 1, the clock is active low, it's idle state is 1.

Then CPHA defines when the data are captured and changed. With CPHA equal 0, the data are sampled on the first edge of the clock and are changed on the second edge. The first data must be stable half clock period before the first edge. With CPHA equal to 1, the data are changed on the first edge of the clock and are captured on the second edge.

The data are sent and received in serial starting with the highest significant bit. If BYTE_MODE bit is high, 8 bits of data are sent and received. The value write in DR[15:8] is not used and the value read in DR[15:8] is unknown else 16 bits of data are sent and received.

The SS_FORCE bit allows forcing the slave select signal, SPI_SS, to 0. It is used mainly in continuous mode to force SPI_SS active during the whole message.

14.3.10.4.4. Master mode

The SPI operates in master mode when the bit MASTER is set. Only a master SPI can initiate transmissions by writing in the SPI data register (DR). If the shifter is empty the transmission starts immediately else it waits until the last word has been transmitted.

A diagram representing the transmission of 2 bytes Ma and Mb by the master (BYTE_MODE=1), receiving Sa and Sb from the slave is shown on Figure 78

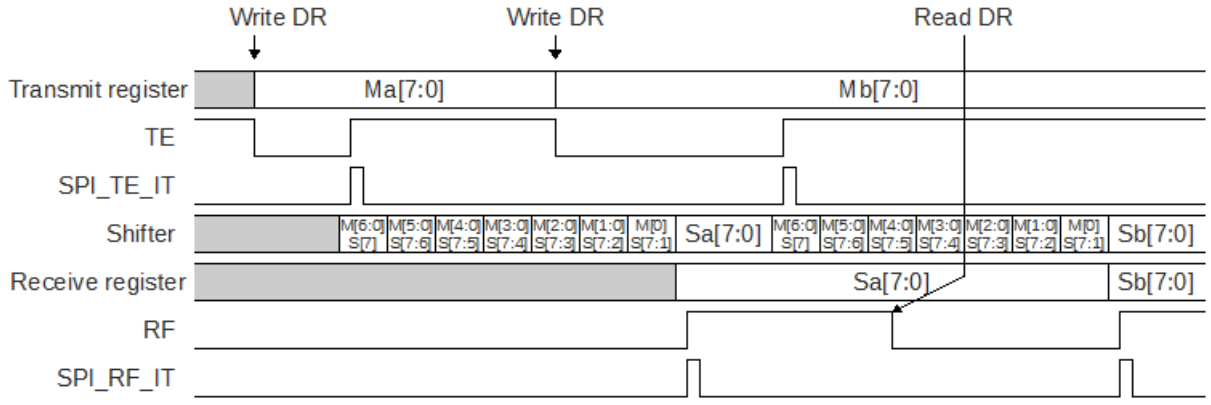


Figure 78 — SPI Send and receive 2 bytes

Software begins the transmission by writing DR. It clears the TE bit and triggers the beginning of the transmission. The SPI_SS pin is lowered and after half SPI clock period, the clock is toggled and the value in the transmit register is transferred in the shifter. At that time, TE goes high again and the interrupt SPI_TE_IT is generated to indicate that a new value can be written in the transmit register.

The shift register is used to shift out the data Ma from the master and shift in the data Sa from the slave. During the transmission the firmware writes a new byte to transmit Mb. This clears the TE bit but does not trigger a new transmission until the current one is completed.

When Sa is received, the RF bit goes high and the SPI_RF_IT interrupt is generated to indicate that the receive register is full and can be read. This is the end of the transmission of the first byte. As the transmit register is not empty, the second transmission is started to send Mb.

During the second transmission, the register DR is read by the software which clears the RF bit. The RF bit is set and a SPI_RF_IT interrupt is generated at the end of the second transmission.

The shifter uses the SPI clock generated by the master which is not synchronous with the CPU clock. The signal going to the CPU is resynchronized. This is why there is small delay between the write in the DR register and the start of the shift.

14.3.10.4.5. Slave mode

The SPI operates in slave mode when the bit MASTER is cleared. In slave mode, the block waits until the SPI_SS pin is low and it received clock pulses to sent data.

The timing diagram of the emission of the byte S and the reception of the byte M from the master is shown on Figure 79.

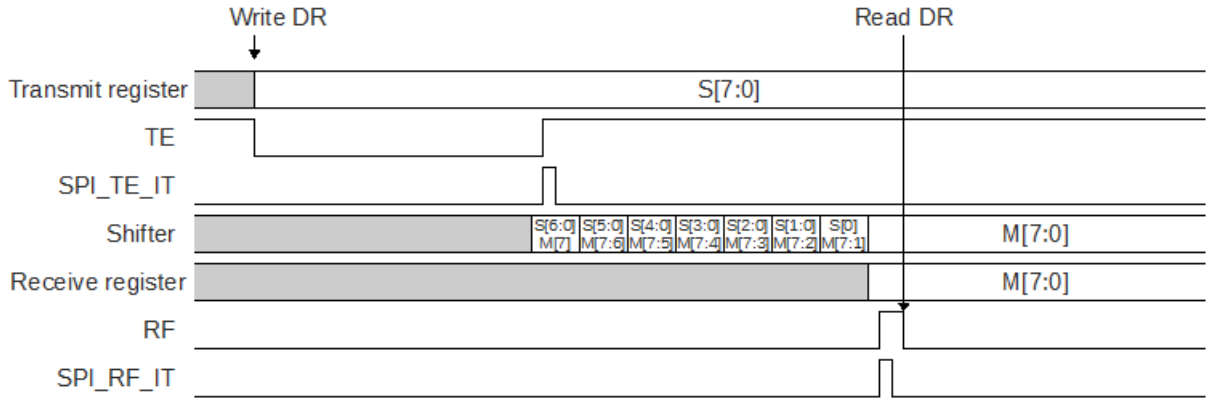


Figure 79 — SPI Receive and send 1 byte

The Software writes DR in the transmit register, lowering the TE bit. This value is sent when the master starts an exchange. At that time the bit TE goes high and the SPI_TE_IT interrupt is triggered.

At the end of the transmission, the bit RF is set and the interrupt SPI_RF_IT is generated. The firmware can read the DR register to get it.

The shifter uses directly the clock from the master and the signal going to the CPU is resynchronized. This is why there is small delay between the end of the shift and the setting of the receive register and the RF bit.

14.3.10.4.6. Error conditions

The SPI block checks for two kinds of errors:

- An overflow error that happens in three cases:
  1. The slave get a transmission but the transmit register is empty and bit TX_EN is set to 1
  2. The master or the slave write in the transmit register but it is not empty and bit TX_EN is set to 1
  3. The master or the slave complete a transmission but the receive register is full and bit RX_EN is set to 1
- A mode error that happens in one case:
  - The slave gets less than 8 or 16 clock pulses before having SPI_SS going high

When such errors happen, the corresponding bit MODF or OVRF is set. It can be read back by the CPU and cleared by writing a 1 in it, writing 0 does nothing. In addition each error triggers the interrupt SPI_ER_IT.

14.3.10.4.7. DMA transfer

The SPI block is able to send and receive a message of up to 64 words reading and writing data directly from the RAM using DMA. This mode is enabled when the bit DMA is set. Such transfer can be done when the SPI block is in master or slave mode. For a master, like transfer without DMA, it is started by a dummy write (the written value is not used) in DR. In the case of a slave, it is also important to perform a dummy write in DR because it allows preloading the first data into the shift register to send to the master, the beginning of the communication itself is triggered by the master.

The length of the message is fixed in both slave and master mode and configured with the MSGLEN field.

The data sent have to be written in memory at the address defined by SBASE. The data received will be written in memory starting from the address RBASE. After each read and each write, the memory address is incremented by 1 if the DMA is in byte mode and 2 if it is in word mode. The SBASE and RBASE register could use the same address, in this case the data sent will be overwritten by the received data but it is not possible to only send or receive data.

A timing diagram of transfer initiated by the SPI block or four words is shown in Figure 80. The master sends Ma, Mb, Mc, Md and receives Sa,Sb, Sc, Sd.

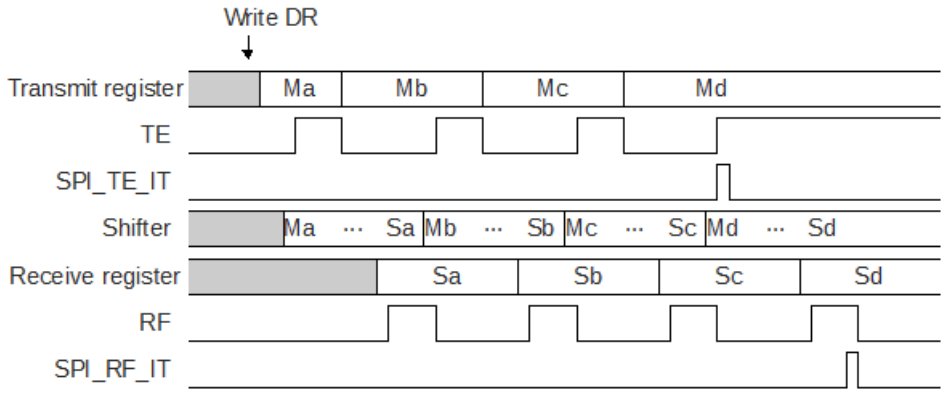


Figure 80 — SPI Send 4 bytes in DMA mode

The transmit register is automatically filled with data from the memory which clears the TE bit. The value written in DR to start the transfer is not used. The TE_IT interrupt is generated only at the end of the message when the DMA has read all the data from the memory.

The content of receive register is automatically saved in the memory, which clears the RF bit. The SPI_RF_IT interrupt is generated only at the end of the transfer when all data have been written in the memory.

14.3.11. I²C interface

The I²C can only address a pre-specified zone of the RAM:

- A 16-bit address port PORT_DMA_OFFSET:I2C_DMA_OFFSET[15:0] defines the start address for I²C
- An “addressed read” command has an 8-bit-only address, so we can read maximum 256 words
- An “addressed write” command has an 8-bit-only address. It is limited to I2C_READ_OFFSET[7:0], i.e. addressed write address < I2C_READ_OFFSET[7:0]
- A direct read (for fast reading) starts at I2C_DMA_OFFSET[15:0] + I2C_READ_OFFSET with a maximum of 256 words above the I2C_DMA_OFFSET

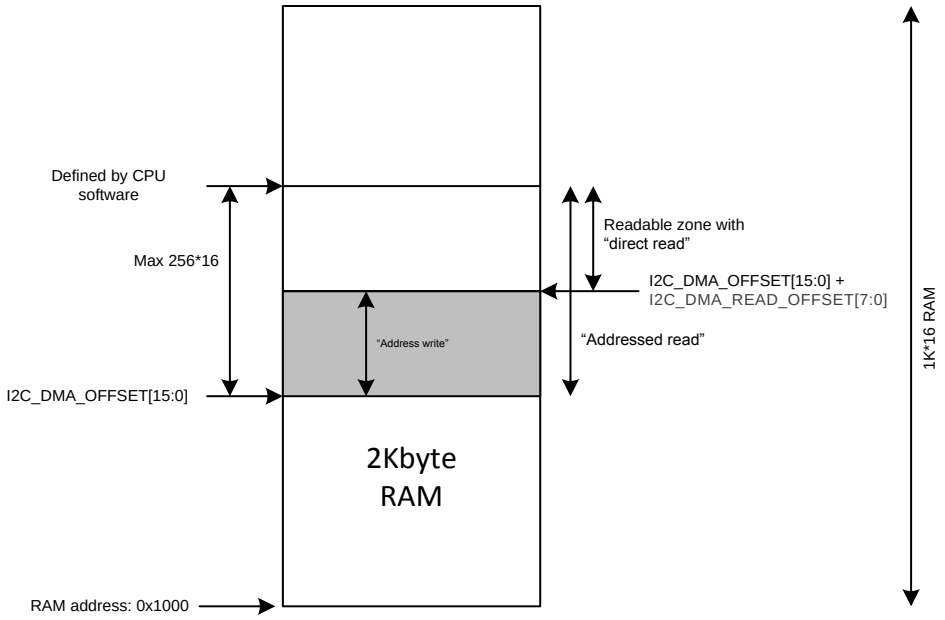


Figure 81 — I²C write/read RAM addressing

14.3.11.1. Different communication modes

14.3.11.1.1. I²C global reset

An I²C global reset command is supported and leads to an interrupt. The interrupt must check for EEEPROM/FLASH-busy before executing a global reset.

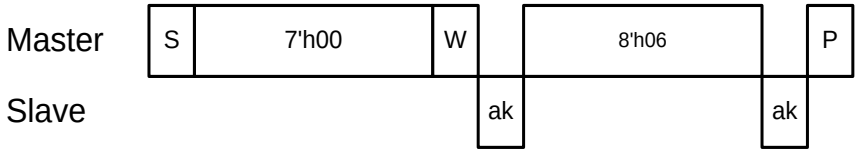


Figure 82 — I²C global reset

14.3.11.1.2. Addressed read mode

The addressed read mode can read any 16-bit word from address I2C_DMA_OFFSET to I2C_DMA _OFFSET + 255.

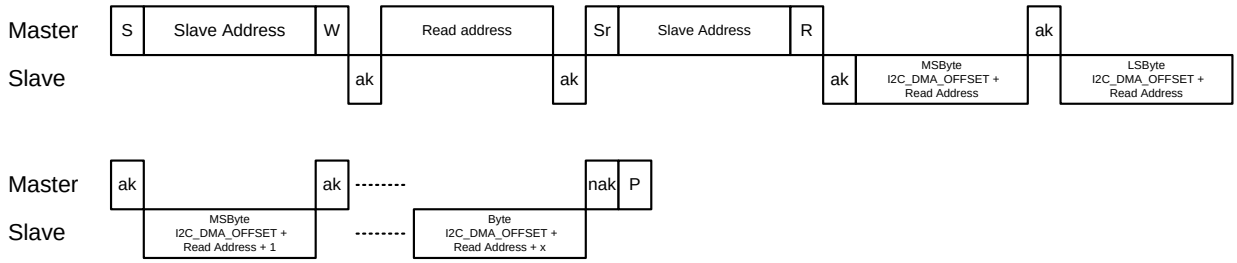


Figure 83 — I²C addressed read mode

14.3.11.1.3. Direct read mode

The direct read mode reads all words from address I2C_DMA_OFFSET + I2C_READ_OFFSET to maximum I2C_DMA_ OFFSET +255.

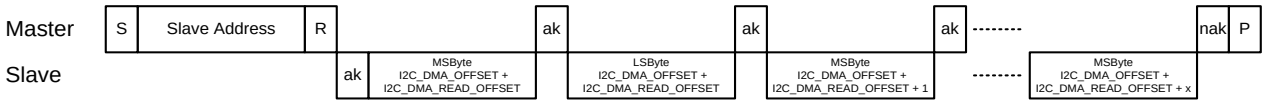


Figure 84 — I²C direct read mode

14.3.11.1.4. Addressed write mode

The addressed write mode can write any 16-bit word from address I2C_DMA_OFFSET to I2C_DMA_OFFSET + I2C_READ_OFFSET – 1.

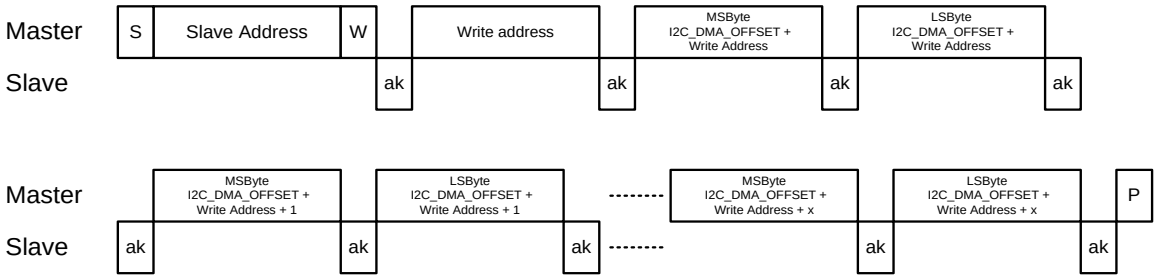


Figure 85 — I²C addressed write mode

14.3.11.2. Address zones

I²C communication	Start address	Highest address space

mode		
Addressed read	I2C_DMA_OFFSET	I2C_DMA_OFFSET + 255
Addressed write	I2C_DMA_OFFSET	I2C_DMA_OFFSET + I2C_ READ_OFFSET - 1
Direct read	I2C_DMA_OFFSET + I2C_READ_OFFSET	I2C_DMA_OFFSET + 255

Table 140 — I²C address zones

Note: I2C_DMA_OFFSET is a Byte address (the LSB is discarded). Example: to set the I²C data block to the beginning of the RAM, I2C_DMA_OFFSET[15:0] should be set to 0x1000 (0x1000 and 0x1001 are the same).

Note: I2C_ READ_OFFSET[7:0] is a WORD address.

When the DMA receives an 8-bit address that is bigger than I2C_READ_OFFSET[7:0] - 1, the write will be refused and a NACK will automatically be sent back to the master. No data is written to the memory and an I2C_IT request is sent to the interrupt controller.

14.3.11.3. Ports map

Port: PORT_I2C_CONF

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:12]		Not used
I2C_SCL_CLK_SEL	[11:9]	RW	Select IO[x] or LIN_RXD input for SCL signal, see Table 88
SDA_FILT_ENABLE	8	RW	Enable I²C data signal filter
I2C_ADDR_VALID	7	RW	enable I²C
I2C_ADDR	[6:0]	RW	7-bit address for I²C

Table 141 — PORT_I²C_CONF

Port: PORT_I2C_DMA_OFFSET

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
I2C_DMA_OFFSET	[15:0]	RW	Byte address defines the start of DMA accesses (so for beginning of

			RAM it is 0x1000). LSB is ignored to align on word for DMA

Table 142 — PORT_I2C_DMA_OFFSET

Port: PORT_I2C_READ_OFFSET

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
-	[15:8]		Not used
I2C_READ_OFFSET	[7:0]	RW	WORD address for “direct read” startup offset. Up to 255

Table 143 — PORT_I2C_READ_OFFSET

14.3.12. UART

14.3.12.1. Features

The MLX81346 contains two full-duplex UART's with a maximum baud rate of 2Mbaud. Both support the operation through RAM buffers using DMA.

DMA mode features:

- Full Duplex UART with NRZ format 8N1 (Startbit + 8 Data bit + Stop Bit)
- 2 receive RAM buffers (A and B) which are swapped by hardware automatically, one transmit RAM buffer
- Buffer start addresses and length defined by ports
- Interrupts for receive, transmit and status reporting
- 16-bits programmable baud rate generator up to 2Mbit/s at 32MHz system frequency
- Programmable data format supporting MSB first, LSB first, big endian, little endian

When using the UART without DMA additional features as variable message length up to 16 bit, other formats like RZ or Manchester encoding, LIN master supporting advanced LIN specifications as time slot error, free line and message collision detection. Please contact Melexis application support for further detail.

14.3.12.2. Block diagram and description

The functional description of the UART block is shown in Figure 86. The control bits are described in 14.3.12.4.

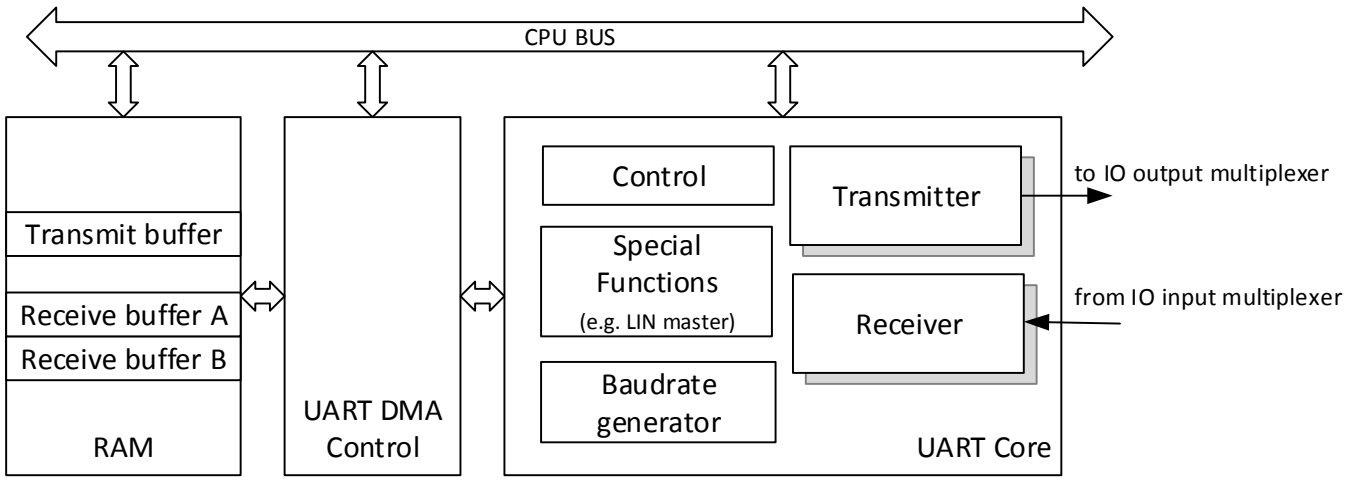


Figure 86 — UART block diagram

The UART DMA extension uses NRZ 8N1 data format:

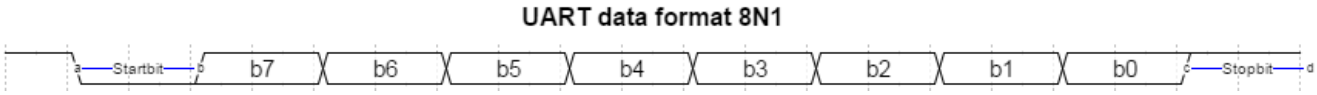


Figure 87 - UART Data format NRZ - 8N1

The UART core needs to be programmed to that data format by the port fields BSC=01 and MLS=001, see below.

The UART DMA bit and byte organisation is controllable by the port bits LSB_FIRST and LOWBYTE_FIRST as shown in the figure below.
Please note that the words in the RAM are always stored big endian, i.e. the high byte is at address n+1, the low byte is at address n.

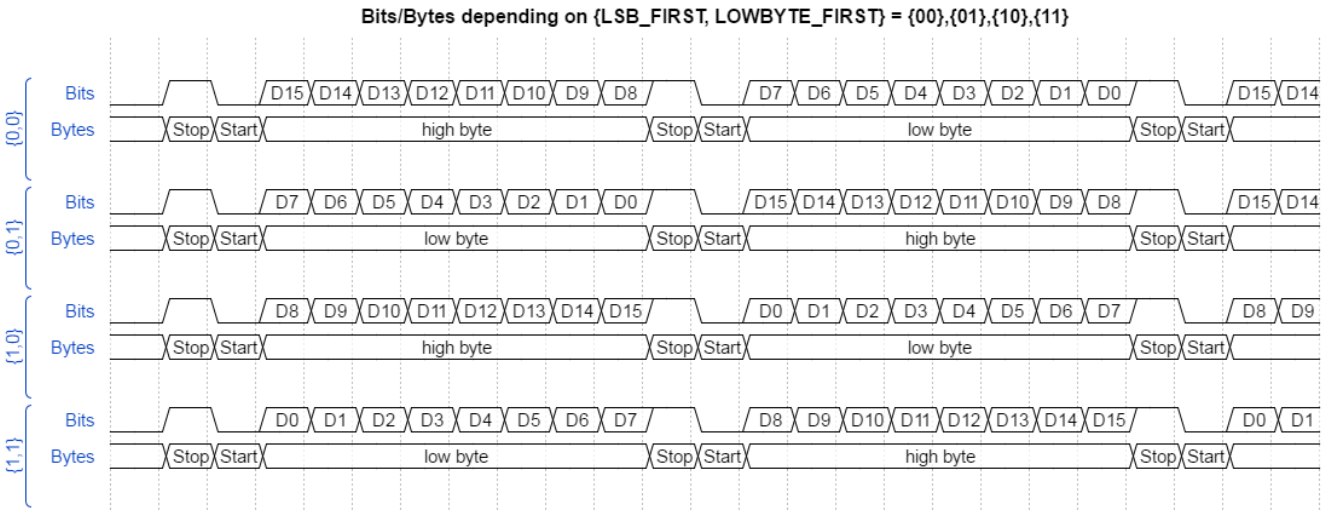


Figure 88: UART DMA — Bits and bytes organization controlled by LSB_FIRST and LOWBYTE_FIRST

14.3.12.2.1. Baudrate generation and data sampling

The common baudrate for receiver and transmitter is derived from the MCU clock through the BRRD register containing the (divider -1) followed by a fix divider by 16:

$$Baudrate = \frac{F_{MCU\_CLK}}{(BRRD + 1)} * \frac{1}{16}$$

The accuracy for a nominal clock is given in the table below:

Baudrate	Divider, error @MCU_CLK 32MHz	Divider, error @MCU_CLK 28MHz	Divider, error @ MCU_CLK 24MHz	Divider, error @ MCU_CLK 20MHz
9600	208, -0.16%	182, -0.16%	156, -0.16%	130, -0.16%
19200	104, -0.16%	91, -0.16%	78, -0.16%	65, -0.16%

38400	52, -0.16%	46, +0.94%	39, -0.16%	33, +1.38%
57600	34, -2.12% 35, +0.79%	30, -1.26% 31, +2.03%	26, -0.16% 27, +3.68%	22, +1.38%
76800	26, -0.16%	23, +0.94%	20, +2.40%	16, -1.70%
115.2k	17, -2.12% 18, +3.55%	15, -1.26%	13, -0.16%	11, +1.38%
230.4k	9, +3.55%	-	-	-
460.8k	-	-	-	-
1 MB	2, 0.00%	-	-	-
2 MB	1, 0.00%			

Table 144 — UART baud-rate divider and baud-rate error

The UART receiver uses the slices of 1/16 of the bit time as defined by the BRRD register for input sampling and resynchronization.
The incoming data is sampled at every slice 0-15. Majority voting of the samples 7, 8 and 9 (out of samples 0-15) is applied to determine the bit value received.
If the samples are not identical, the noisy bit receiver flag NBR is set.
The stop bit validity check requires the majority of the samples 7, 8 and 9 to be 1, otherwise the stop bit error flag SBE is set.
The control bits UART{1,0}_STOP_MODE can be used to relax the stop bit sampling. When this bit is set the stopbit may occur latest at samples 14 and 15 without causing a stop bit error.

The receiver re-synchronizes with every start bit detected at the accuracy of 1/16 of the bit time.

14.3.12.3. UART Interrupts

The UART blocks generate the following interrupts:

Label	Interrupt	Status Bits	Description
Interrupts provided by the DMA control			
UDFR{0,1}	UART0 - DMA frame/buffer received	UDMA_FRC	The receive RAM buffer is full. Hardware switched to other receive buffer.
UDTF{0,1}	UART0 - DMA frame/buffer transmitted	UDMA_FRT	UART DMA frame has been transmitted, transmit buffer has been fully read and can be overwritten by the software.
Interrupts provided by the UART core			
UART{0,1}_RR	UART receive begins	RSB, RSO	The UART_RR interrupt indicates the begin of a single UART message, the interrupt is triggered when the start bit was detected. The flag receiver shifter busy RSB is set. If the previous UART data was not read, the overrun bit RSO is set.
UART{0,1}_RS	UART receive ends	RFF NBR (RSB)	Indicates the end of the reception of the current single UART message together with the receiver full flag RFF. The RSB is cleared again. The noisy bit receiver flag NBR maybe set on noisy reception. The baudrate configuration should be checked.

UART{0,1}_SB	UART stop bit error	SBE	Indicates a framing error by the stop bit error flag SBE.
UART{0,1}_TR	UART transmit begins	TSB	Indicates the begin of a transmit, will be triggered at every UART message (i.e. every byte in DMA mode). The flag transmit shifter busy TSB is set.
UART{0,1}_TS	UART transmit ends	(TSB)	Indicates the end of a transmit, will be triggered at every UART message (i.e. every byte in DMA mode). The flag TSB is cleared again.
UART{0,1}_TE	UART transmit error	TRO	Indicates on overwrite of the transmit data before transmit is ready. The transmitter overrun flag TRO is set. If that interrupt occurs in DMA mode, the synchronization between UART core and UART DMA hardware maybe lost. The UART core and the UART DMA hardware shall be restarted.

Table 145 UART interrupts

Note: In DMA mode the application software shall monitor the UART core error interrupts (UART_SB and UART_TE) and reinitialize the UART core and DMA hardware on the transmit error interrupt.

14.3.12.4. UART IO ports

Both UART blocks have the identical set of ports differentiated only by the index in the name. This is noted by UDMA{0,1} and UART{0,1} in the ports table below.

Port: PORT_UDMA{0,1}_RDA

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
UDMA{0,1}_RDA	[15:0]	RW	Read buffer address for buffer A

Table 146 — UDMA{0,1}_RDA port

Port: PORT_UDMA{0,1}_RDB

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
UDMA{0,1}_RDB	[15:0]	RW	Read buffer address for buffer B

Table 147 — UDMA{0,1}_RDB port

Port: PORT_UDMA{0,1}_TX

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
UDMA{0,1}_TX	[15:0]	RW	Transmit buffer address

Table 148 — UDMA{0,1}_TX port

Port: PORT_UDMA{0,1}_SIZE

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**
UDMA{0,1}_SIZTX	[15:8]	RW	Length of transmit buffer in words
UDMA{0,1}_SIZRX	[7:0]	RW	Length of receive buffers A and B in bytes

Table 149 — UDMA{0,1}_SIZE port

Port: PORT_UDMA{0,1}_CTRL

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Byte
Field name	**Bit**	**R/W**	**Description**

Address	Reset		Access
UDMA{0,1}_TX	[15:5]	RW	R: always read 0; W: do nothing
UART{0,1}_STOP_MODE	4	RW	Relaxed UART stop bit sampling is active when =1 (rising edge is allowed anywhere in the bit time)
UDMA{0,1}_LOWBYTE_FIRST	3	RW	Data is sent/received low byte first if "lowbyte_first"=1, else high byte first
UDMA{0,1}_LSB_FIRST	2	RW	Data is sent/received least significant bit (LSB) first if "lsb_first"=1, else MSB first
UDMA{0,1}_TXSTART	1	RW	A rising edge will begin the transmission of current buffer. The bit needs to be cleared by software before the next frame can be transmitted. Clearing the bit in a running transmission will abort it after the current byte has been sent.
UDMA{0,1}_EN	0	RW	0: block is not active (CPU has direct UART access), 1: block is active

Table 150 — UDMA{0,1}_CTRL port

Port: PORT_ UDMA{0,1}_STATUS

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
-	[15:4]		Not used
CUSTOM_UART{0,1}_DMA_ERR	3	R	DMA error occurred during RAM buffer access
UDMA{0,1}_FTR	2	R	frame transmit ready (reset 0, cleared at begin of transmission, set after end of frame)
UDMA{0,1}_FRC	1	R	frame received
UDMA{0,1}_RD_BUFFER_VALID	0	R	0 means data in read buffer B is valid to be read by software, buffer A is currently filled by the DMA, 1 = data in read buffer A is valid, buffer B is in use by the DMA (reset : 0)

Table 151 — UDMA{0,1}_STATUS port

Port: UART{0,1}

Address	Reset		Access
See Table 106 – MLX81346 ports overview UART{0,1} baud rate register	0x00		Word
Field name	**Bit**	**R/W**	**Description**
BRRD	[15:0]	RW	Baud rate register data

Address	Reset	Access

Address	Reset	Access
See Table 106 – MLX81346 ports overview UART{0,1} transmit data register	0x00	Word

Field name	Bit	R/W	Description
TRD	[15:0]	RW	Transmit register data

Address	Reset	Access
See Table 106 – MLX81346 ports overview UART{0,1} receive data register	0x00	Word, Read Only

Field name	Bit	R/W	Description
RRD	[15:0]	R	Receive register data

Address	Reset	Access
See Table 106 – MLX81346 ports overview UART{0,1} frame duration/bit counter data register	0x00	Word

Field name	Bit	R/W	Description
LFDD	[15:8]	RW	Lin frame duration data, set 0 in DMA mode
LFCD	[7:0]	R	Lin frame bit counter data

Address	Reset	Access
See Table 106 – MLX81346 ports overview UART{0,1} control register	0x00	Word

Field name	Bit	R/W	Description
LDC	15	RW	LIN master control bit, set 0 in DMA mode
LSC	14	RW	LIN master control bit, set 0 in DMA mode
LBC	13	RW	LIN master control bit, set 0 in DMA mode
BSC	[12:11]	RW	Bit scrambler control, set to **01** in DMA mode
MLS	[10:8]	RW	Message length selector, set to **001** in DMA mode
LTE	7	R	LIN master status bit, set 0 in DMA mode
LSE	6	R	LIN master status bit, set 0 in DMA mode
LBE	5	R	LIN master status bit, set 0 in DMA mode
-	4		Not used
ISB	3	RW	Transmit IDLE state, set 0 in DMA mode
REE	2	RW	Receiver enable , set **1** in DMA mode
TRE	1	RW	Transmitter enable, set **1** in DMA mode
-	0		Not used

Table 152 - UART{0,1} core data and control ports

Address	Reset	Access
See Table 106 – MLX81346 ports overview		
UART{0,1} status register	0x00	Word, Read Only, cleared with receiver enable REE = 0

Field name	Bit	R/W	Description
SBE	15	R	Stop bit error
NBR	14	R	Noisy bit reception
RRF	13	R	Receive register full
RSB	12	R	Receive shifter busy
RSO	11	R	Receive shifter overrun
TSB	10	R	Transmit shifter busy
TRB	9	R	Transmit register busy
TRO	8	R	Transmit register overrun
-	[7:0]		Not used

Table 153 — UART{0,1} core status port

14.4. LIN interface

14.4.1. LIN transceiver

14.4.1.1. Description

Figure 89 shows the block diagram of the LIN transceiver.

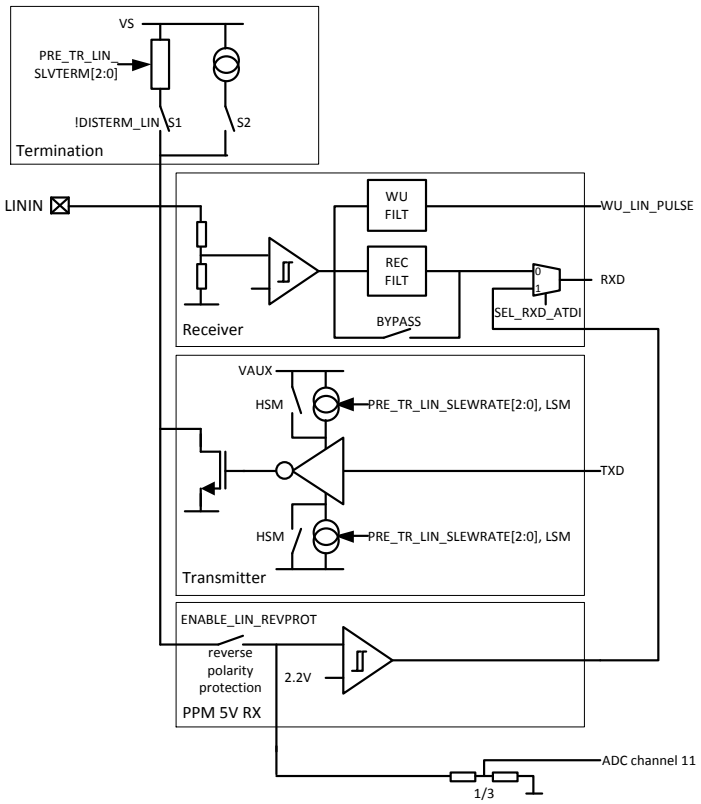


Figure 89 — LIN transceiver block diagram

The termination resistor can be disconnected by DISTERM_LIN (= PORT_LIN_XCFG:LIN_XCFG[9]). The termination resistor value is trimmed by TRIM_RCO1M: PRE_TR_LIN_SLVTERM[2:0], see Table 154.

The receiver consists of a comparator followed by a receive EMC filter which can be bypassed by BYPASS (= PORT_LIN_XCFG:LIN_XCFG[8]). A multiplexer controlled by SEL_RXD_ATDI (= PORT_LIN_XCFG:LIN_XCFG[11]) either selects the output of the EMC filter or the output of a second comparator with a fixed 2.2V threshold, which can be used for PPM communication with 5V signal levels.

The transmitter consists of an NMOST which is driven by current sources. The slew rate can be controlled by adjusting the current sources through TRIM_RCO1M:PRE_TR_LIN_SLEWRATE[2:0], see Table 155.

The transceiver can be operated in next modes:

- Normal speed mode: default operating mode with maximum transmission bit rate of 20kB/s (LSM=0, HSM=0, BYPASS=0).

- Low speed mode: recommended operating mode for J2602 applications with maximum baud rate of 10.4kBd, reduced slew rate for minimum radiated noise (LSM=1).
- High speed mode: mode for special point-to-point communication, slew rate control disabled (HSM=1), receive filter bypassed (BYPASS=1).

PRE_TR_LIN_SLVTERM	Resistor value [kΩ]
0	36
1	33
2	30
3	27
4	45
5	42
6	39
7	36

Table 154 — LIN termination resistor trimming

PRE_TR_LIN_SLEWRATE	Relative value LSM=0	Relative value LSM=1
0	100%	50%
1	87.5%	43.75%
2	125%	62.5%
3	112.5%	56.25%
4	50%	25%
5	37.5%	18.75%
6	75%	37.5%
7	62.5%	31.25%

Table 155 — LIN transmitter slew rate trimming

Port: PORT_LIN_XKEY

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		System, Word, Byte
Field name	**Bit**	**R/W**	**Description**
LIN_XKEY	[15:0]	RW	Store a valid key for LIN XCFG (0xB2A3)

Table 156 — PORT_LIN_XKEY

Port: PORT_LIN_XCFG

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		System, Word, Byte; result taken in account only if XKEY is valid
Field name	**Bit**	**R/W**	**Description**
CXPI_DIS_WU_DEB	13	RW	decrease the debounce time of the wake-up comparator from 70µs to 5.5µs to support CXPI protocol
EN_LIN_REVPROT	12	RW	disconnects the reverse polarity protection from internal LIN node, is needed to measure LIN level by ADC or to run fast protocol at 5V level (PPM, CXPI)
SEL_RXD_ATDI	11	RW	if 1, the fast comparator used in test mode (ATDI) will be switched to the RX input (this allows protocol with higher baudrate, e.g. PPM, FASTLIN or CXPI)
RX_INVERT	10	RW	invert the RX input before any multiplexing
DISTERM	9	RW	disable bus termination for auto-addressing
BYPASS	8	RW	bypass the receiver for high speed mode
HSM	7	RW	high speed mode (slew rate disabled)
LSM	6	RW	low speed slope control
SLEEPB	5	RW	disable sleep mode
SEL_COLIN_B	4	RW	SEL_COLIN_B
SEL_IO_TO_COLINRX	3	RW	select COLIN_RX driven from IO
SEL_RX_IO	2	RW	select Rx driven from IO
TX_INVERT	1	RW	invert Tx output
SEL_TX_EXT	0	RW	select Tx driven from IO

Table 157 — PORT_LIN_XCFG

Port: PORT_LIN_XCFG_VALID

Address	Reset		Access
See Table 106 – MLX81346 ports overview	0x00		Word, Read Only
Field name	**Bit**	**R/W**	**Description**
LIN_XCFG_VALID	[15:0]	R	LIN_XKEY valid: port mirrors LIN_XFCG LIN_XKEY not valid: port keeps previous value

Table 158 — PORT_LIN_XCFG_VALID

14.4.2. LIN digital shell

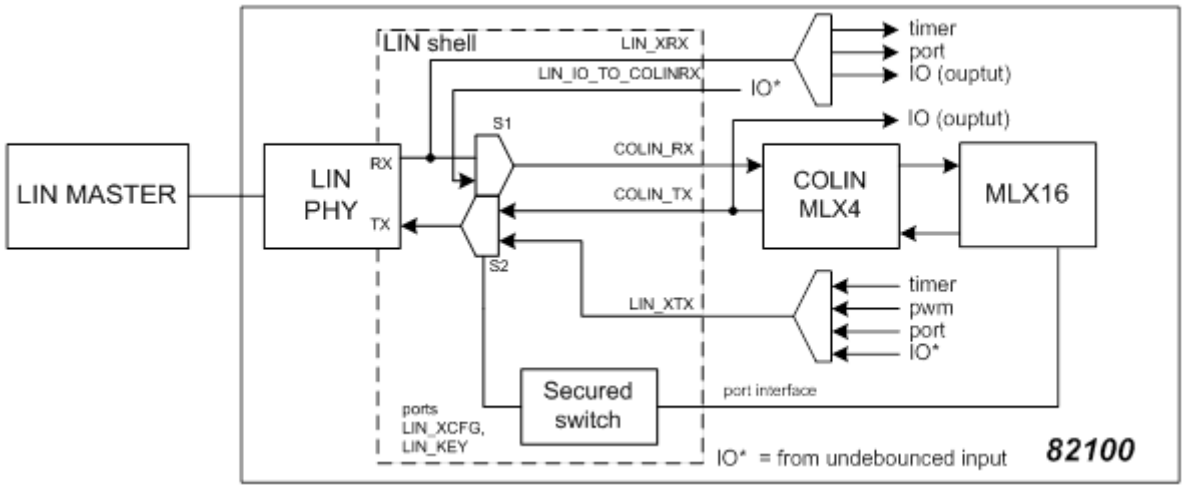


Figure 90 — LIN PHY RX/TX connections

15. Disclaimer

The content of this document is believed to be correct and accurate. However, the content of this document is furnished "as is" for informational use only and no representation, nor warranty is provided by Melexis about its accuracy, nor about the results of its implementation. Melexis assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. Customer will follow the practices contained in this document under its sole responsibility. This documentation is in fact provided without warranty, term, or condition of any kind, either implied or expressed, including but not limited to warranties of merchantability, satisfactory quality, non-infringement, and fitness for purpose. Melexis, its employees and agents and its affiliates' and their employees and agents will not be responsible for any loss, however arising, from the use of, or reliance on this document. Notwithstanding the foregoing, contractual obligations expressly undertaken in writing by Melexis prevail over this disclaimer.

This document is subject to change without notice, and should not be construed as a commitment by Melexis. Therefore, before placing orders or prior to designing the product into a system, users or any third party should obtain the latest version of the relevant information.
Users or any third party must determine the suitability of the product described in this document for its application, including the level of reliability required and determine whether it is fit for a particular purpose.

This document as well as the product here described may be subject to export control regulations. Be aware that export might require a prior authorization from competent authorities. The product is not designed, authorized or warranted to be suitable in applications requiring extended temperature range and/or unusual environmental requirements. High reliability applications, such as medical life-support or life-sustaining equipment or avionics application are specifically excluded by Melexis. The product may not be used for the following applications subject to export control regulations: the development, production, processing, operation, maintenance, storage, recognition or proliferation of:
1. chemical, biological or nuclear weapons, or for the development, production, maintenance or storage of missiles for such weapons;
2. civil firearms, including spare parts or ammunition for such arms;
3. defense related products, or other material for military use or for law enforcement;
4. any applications that, alone or in combination with other goods, substances or organisms could cause serious harm to persons or goods and that can be used as a means of violence in an armed conflict or any similar violent situation.

No license nor any other right or interest is granted to any of Melexis' or third party's intellectual property rights.

If this document is marked "restricted" or with similar words, or if in any case the content of this document is to be reasonably understood as being confidential, the recipient of this document shall not communicate, nor disclose to any third party, any part of the document without Melexis' express written consent. The recipient shall take all necessary measures to apply and preserve the confidential character of the document. In particular, the recipient shall (i) hold document in confidence with at least the same degree of care by which it maintains the confidentiality of its own proprietary and confidential information, but no less than reasonable care; (ii) restrict the disclosure of the document solely to its employees for the purpose for which this document was received, on a strictly need to know basis and providing that such persons to whom the document is disclosed are bound by confidentiality terms substantially similar to those in this disclaimer; (iii) use the document only in connection with the purpose for which this document was received, and reproduce document only to the extent necessary for such purposes; (iv) not use the document for commercial purposes or to the detriment of Melexis or its customers. The confidentiality obligations set forth in this disclaimer will have indefinite duration and in any case they will be effective for no less than 10 years from the receipt of this document.

This disclaimer will be governed by and construed in accordance with Belgian law and any disputes relating to this disclaimer will be subject to the exclusive jurisdiction of the courts of Brussels, Belgium.

The invalidity or ineffectiveness of any of the provisions of this disclaimer does not affect the validity or effectiveness of the other provisions. The previous versions of this document are repealed.

IATF 16949 and ISO 14001 Certified