Fully Integrated Single Motor Controller for Electrical Motors

1. Functional Description

The AMG-XB403 is a feature rich SoC, providing a single chip solution for motor control, including Digital Motor Control Engine (DMCE), Power Factor Correction (PFC), high level system control MCU, and Switch Mode Power Supply (SMPS). The IC's multi-core design enables parallel execution of all system functions while keeping the MCU load to a minimum. The DMCE includes a 32bit floating point processing unit, PWM generator, and power stage driver allowing for sensorless drive. The AMG-XB403's switching power supply circuit, which includes software adjustable switching frequency, requires the IC to only have a single, low-grade power supply.

2. Features

- ❑ Supply voltage: 4.5VDC...5.5VDC
- ❑ SoC for asynchronous & synchronous motors
- ❑ Independent DMCE (32bit floating point)
- ❑ Digital PFC (16bit RISC)
- ❑ SMPS with self start
- ❑ AVR® compatible 8 bit MCU with 32kB program memory
- ❑ Over-current protection with single shunt current control
- ❑ Fully digital, programmable PFC controller
- ❑ Internal ±2% RC-oscillator
- ❑ Software-adjustable clock frequency FLL
- ❑ 10-bit 2-MSPS multiplexed ADC
- ❑ Integrated switching power supply (SMPS)
- ❑ Integrated EEPROM memory
- ❑ 7 GPIO (1 thereof which can be analog)
- ❑ Ambient temperature range: -25°C...85°C
- ❑ Package: LQFP48 – Body size: 7mm x 7mm x 1.4mm
- ❑ RoHS compliant

® AVR® is a registered trademark of Atmel Corporation

3. Application

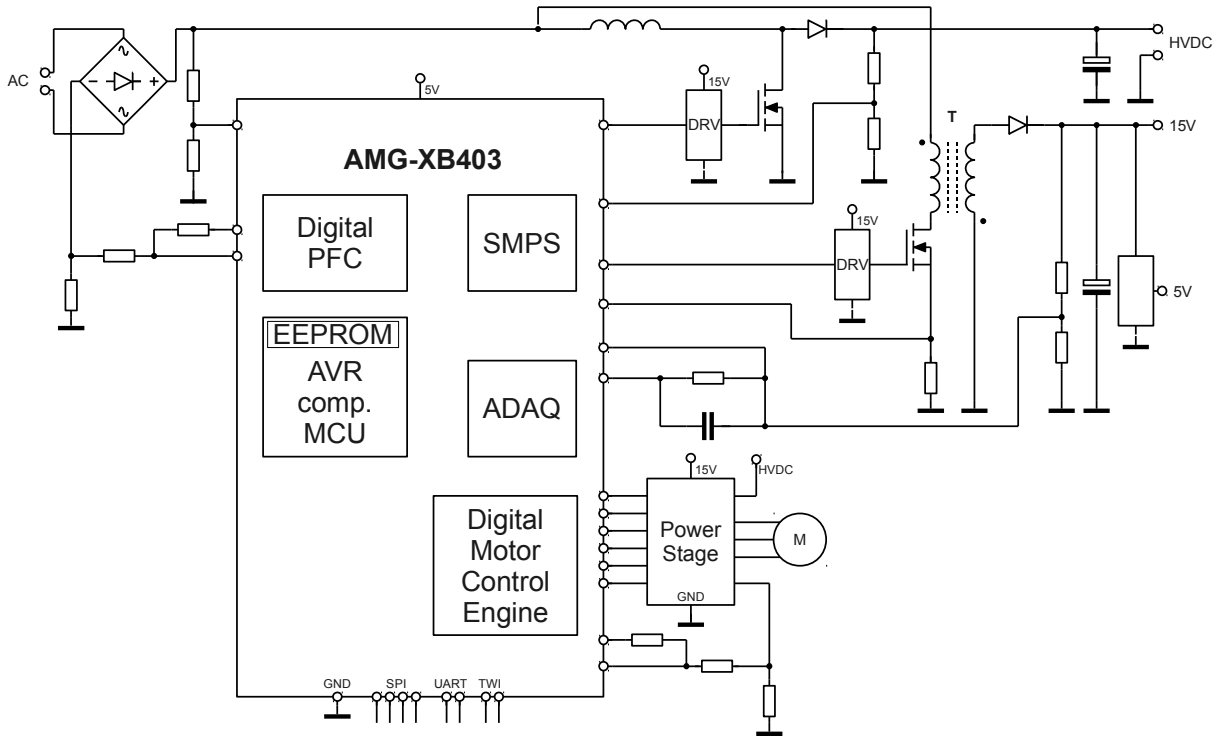


Figure 1: AMG-XB403 Simplified Application Circuit

Table of Contents

4. Block Diagram

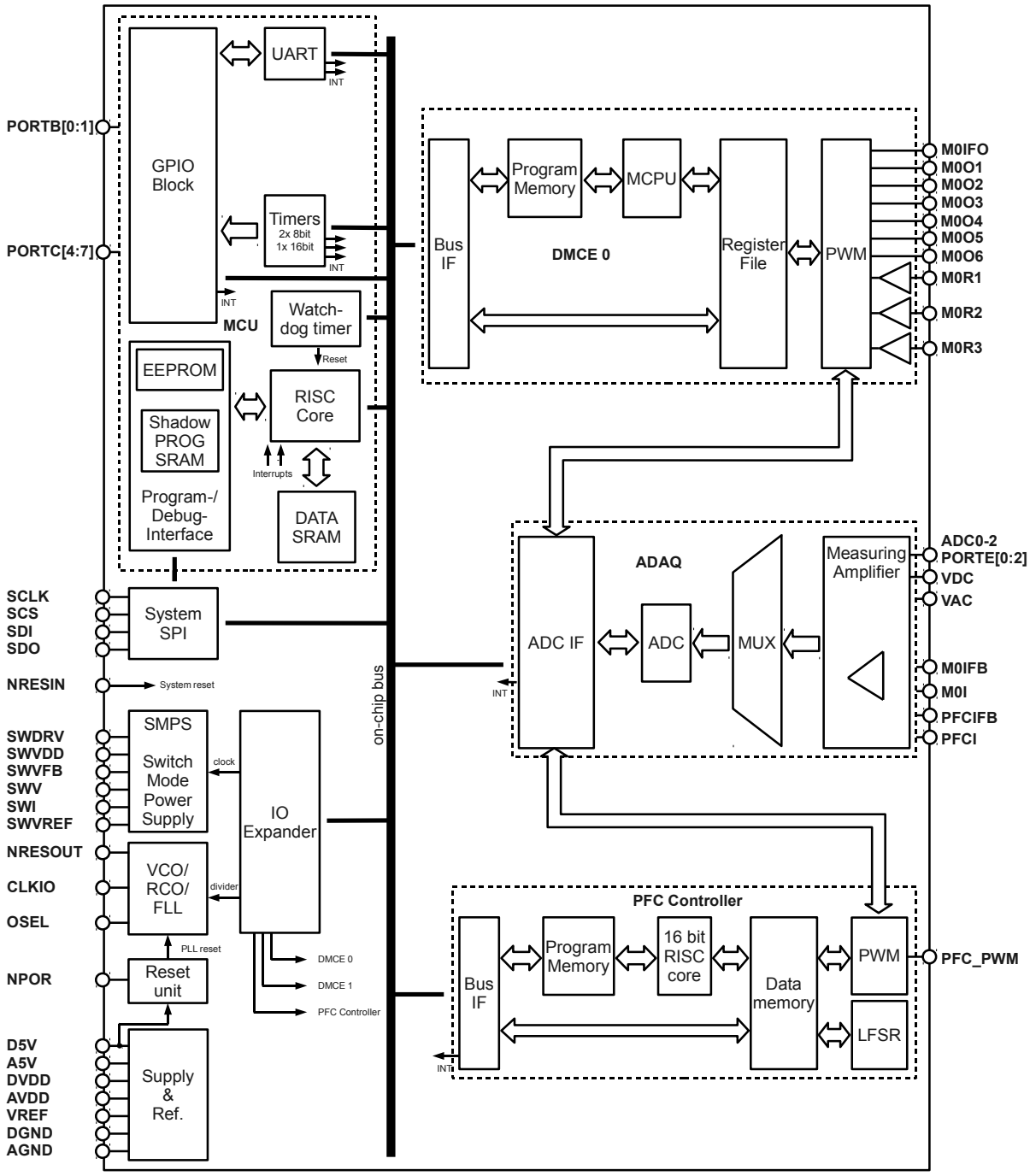


Figure 2: AMG-XB403 Block Diagram

5. Block Descriptions

5.1. System SPI Unit and On-Chip Bus

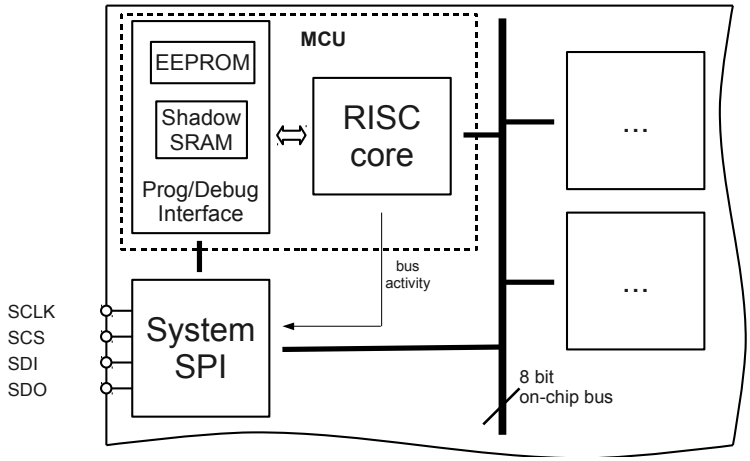


Figure 3: SPI bus masters and clients overview

On-chip communication is built around a 8 bit on-chip bus. There are two bus masters: the MCU core is the primary bus master and the System SPI unit is the secondary bus master. The System SPI unit is used to program, configure, and debug the AMG-XB403. An activity signal from the MCU to the System SPI unit indicates activity on the bus, and ensures that the System SPI can only access the bus if the MCU is not. The MCU bus reads and writes, can slow down but not inhibit the execution of System SPI commands. Please note that the EEPROM and debug interface can be accessed via the System SPI only.

There are four basic System SPI commands to write and read data, 'Table 1' shows an overview. See 'Table 30' on page 55 for an overview of all 8 bit buses addresses.

Command	Description
0	not used
1	write 8 bit data to 5 bit addressed register (see figure 4)
2	read 8 bit data from 5 bit addressed register (see figure 5)
3	write 8 bit data to 5 bit addressed register; wait for LSB=0 at the same address (see figure 6)
4	read 32 bit data from 5 bit addressed register (see figure 7)

Table 1: System SPI commands

The bus employs 6 bit wide addressing, while the MCU may access all 64 addresses, the System SPI can only access the 32 lower addresses.

System SPI transmissions are framed by the chip select signal SCS, the System SPI unit is active only when SCS is LOW. The transmission timing is determined by the serial clock signal SCLK. Data is received via the serial data input signal SDI and sent via the serial data output signal SDO. Data signals are valid at the rising edge of the serial clock signal. Every transmission starts with the MSB and ends with an inverted XOR parity bit generated over all previously sent bits. If the transmission is received correctly, i.e. ends with a valid parity bit, the command will be processed by the System SPI unit. Handshaking is accomplished by the master waiting for SDO to become HIGH after issuing a command. In the case of an unsuccessful parity check SDO will stay LOW permanently and a time-out mechanism must take effect on the master's side. The signal

waveforms for all commands are shown below.

A digital low-pass filter function can be activated for all inputs of the System SPI unit in order to improve noise rejection, refer to section 5.2, page 8. The maximum usable clock frequency of the System SPI unit will be reduced as the filter depth is increased.

The 8 bit write command contains 3 command bits followed by the 5 bit address of the register to be written to and 8 bits of data (see 'Figure 4').

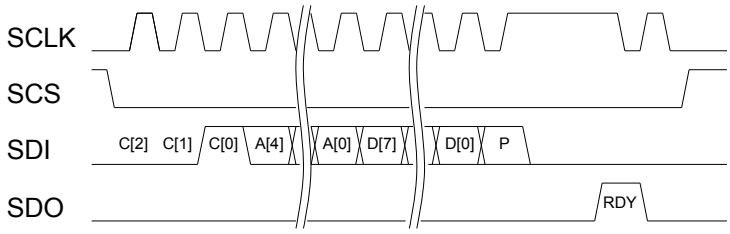


Figure 4: Command 1, write (8 bit)

The 8 bit read command requests the content of a register. The System SPI responds by setting SDO to HIGH when data is ready and returning 8 bits of data starting with the second negative edge of SCLK (see 'Figure 5').

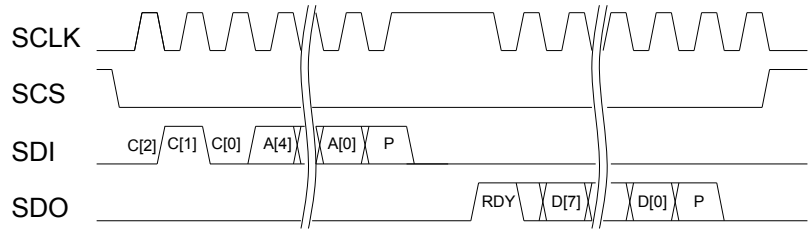


Figure 5: Command 2, read (8 bit)

Command 3 writes 8 bits of data into a register and will not return a ready status until the LSB of the previously written register becomes 0. This command is used when synchronization with units running at a different clock frequency, such as the DMCE core, is required (see 'Figure 6').

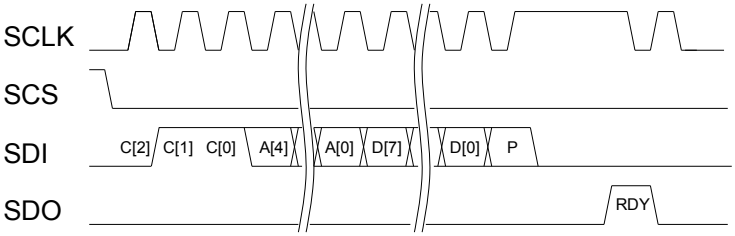


Figure 6: Command 3, write/read (8 bit)

Command 4 reads a 32 bit DMCE data register value in a single transmission cycle which allows for higher effective data rates than using a series of 8 bit reads and writes, see section 5.4.4, page 21 for the description of DMCE register reads. As shown in 'Figure 7', a 5 bit DMCE base address and an 8 bit register address R[7:0] are sent by the master and a 32 bit wide reply is returned by the System SPI unit. The base address always must be written as 1.

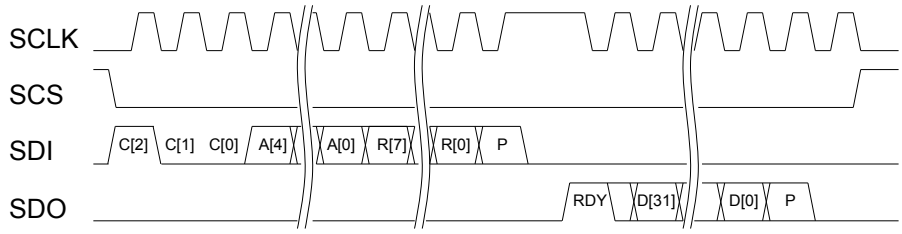


Figure 7: Command 4, register read (32 bit)

5.2. IO Expander

The IO expander module serves to provide additional MCU configuration registers (see 'Table 2'). The address register IOEXPA is used to set the index for the configuration register accessed through the data register IOEXPD. IOEXPD provides access to to following functions.

5.2.1. IC Revision Information

Address 0 contains the number of the current AMG-XB403 revision.

5.2.2. Test Modes, EEPROM Self-Programming Mode, Lock Indicator, Clock Control

Bit 0 of address 1 enables the digital test mode if set to 1 and cannot be read. The digital test mode can only be left by performing an IC reset. This function is used during IC test only.

Bit 1 of address 1 sets and indicates EEPROM self-programming mode which is used to write and read from the AMG-XB403's program EEPROM. EEPROM self-programming mode is enabled by setting EME to 1, and left by setting EME to 0. This mode is only available when the AMG-XB403 is not locked. For details on self-programming see section 5.7.1, page 34.

Bit 2 of address 1 indicates if the AMG-XB403 is locked, 1 indicates a locked IC, which is the initial state. See section 5.7.1, page 34 for the details of unlocking the IC.

Bit 3 of address 1 is reserved and must not be set to 1.

Bit 4 of address 1 is used to control the output of the system clock via the CLKIO pin. When this CLKOFF flag is set to 1 output via pin CLKIO will be disabled, e.g. when the AMG-XB403's system clock is not required to be provided to off-chip circuitry. Initially CLKOFF is set to 0.

Bit 5 of address 1 is used to fix the clock select signals in the present state. When this CLKLOCK flag is set to 1 the clock selection will be fixed according to the value present at pins CLKSEL and XO_OSEL, refer to section 5.8. page 51, i.e. noise present at these pins will not interfere with the clock source selection. Initially, CLKLOCK is set to 0.

5.2.3. Power Output Polarity and Over-Current Flag (OFI) Control for DMCE

The lower nibble of address 2 controls the logic level at Power output pins. All six signal pins will remain in high ohmic state unless DMCE0OE is set to 1. If DMCE0POL is set to 1 the connected power stage must have an inverting input to output characteristic, otherwise the power stage must be non-inverting.

The higher nibble of address 2 controls the power stage over-current protection interface, refer to 'Figure 14' on page 23. OFIPOL0 selects the polarity of the OFI signal, for an active-HIGH OFI signal input these flags must be set to zero. OFIERR0 is set to one if an over-current error of the associated power stage is encountered. If either flag is set the shared DMCE/PFC interrupt flag will be set. The error flag can be reset by writing a value of 1 to OFIRES0 . Resetting the error flag is mandatory when the shared DMCE/PFC interrupt is enabled, see also section 5.7.7, page 49.

5.2.4. SMPS PWM Frequency

The clock prescaler SWCPRE for the SMPS is set using addresses 3 and 4. The lower byte has to be written before the high byte. For the changed value to have an effect, both bytes have to be written. The SMPS clock frequency is calculated as follows:

$$f_{SWC} = \frac{f_{MCU}}{2 \cdot (SWCPRE + 1)} \quad , \quad 1 \leq SWCPRE \leq 32767 \quad .$$

Setting SWCPRE to 0 enables an internal RC oscillator to generate the SMPS's PWM clock. Its typical frequency is 22kHz. This is the default on startup.

5.2.5. Polarity of the PFC Switching Signal

The LSB of address 5 sets the polarity of the PFC controller's logic level PWM output. 0 means active HIGH, 1 means active LOW.

5.2.6. PWM Frequency Setting for DMCE

DMCE's PWM clock setting is controlled through a register pair C1, C2 (accessible via addresses 6 to 8). The range of C1 is 0...7 (3 bit) and the range of C2 is 0...255 (8 bit). Address 6 also holds the flag CNT0X2, to double the PWM base frequency of the DMCE. The PWM frequency can be calculated as follows:

$$f_{PWM} = \begin{cases} f_{FPU} \cdot \dfrac{2^8 - C2 + C1}{2^{13} \cdot (2^8 + C1\,(C2 - 1))} \cdot 2^{CNTX2} & ;\ \text{for}\ C2 > 1 \\ f_{FPU} \cdot \dfrac{2^8 - C2}{2^{21} - C2} \cdot 2^{CNTX2} & ;\ \text{for}\ C2 \leq 1 \end{cases}$$

5.2.7. FLL Configuration and Status

FLL related functions are controlled using addresses 9,10, and 12. The lower byte of the divider setting (address 9) has to be written before the high byte (address 10). The FLL divider FLLDIV is used to set the clock frequency of the PFC, ADC, and DMCE which can be calculated as:

$f_{VCO} = \frac{FLLDIV + 1}{16 \cdot T_{RC}}$, where T_{RC} is the RC-ramp's rise time.

The MCU's clock frequency is calculated as follows: $f_{MCU} = \frac{f_{VCO}}{8}$. Upon reading FLLDIV the most recently acquired VCO frequency counter value will be returned, thus allowing the monitoring of the VCO frequency directly, please note that the high byte and low byte of the counter value are not guaranteed to belong to the same VCO frequency counter value.

PREC determines the FLL's frequency step for regulating the system clock frequency. Vaild values are 4 to 11. Larger values correspond to smaller frequency steps. PREC is increased when the target frequency is crossed within 3 frequency adjustment steps otherwise PREC will be decreased. The maximum value of PREC is given by the value of MPREC. Both maximum PREC value and current PREC value can be read from address 12. MPREC may also be set, its initial value is 10.

5.2.8. U, V, W Phase Feedback Monitor

The comparator result of the signals present at M0R1, M0R2, and M0R3 can be read back from address 13. RU, RV, and RW are assigned according to the setting described in section 5.2.13.

Writing bit 3, DTC0OVR, disables hardware dead time compensation for DMCE0.

Writing DTC0RNG sets up the allowable error range for dead time compensation of DCME0. If the actual duty cycle, determined by the feed back pins, differs from the expected duty cycle by more than $\frac{2^{(DTC0RNG+CNT0X2)}}{8}$, dead time compensation will be skipped. Refer to section 5.2.6, page 9 for clarification regarding CNT0X2 .

5.2.9. ADC Offset Calibration Data

Addresses 14 to 25 hold the ADC offset compensation values. The order of the channels are described in 'Table 4' on page 17. The offset values are 8-bit signed integer numbers for positive and negative offset-correction.

5.2.10. PFC, DMCE and ADC Software Controlled Reset Signals

Address 26 holds the reset flags for the DMCE core, the PFC controller, and the ADC interface. The reset flags are active if set to one and initially reset to zero. Setting these flags can be used to disable inactive units.

5.2.11. Operational Amplifier Power Down Signals

Addresses 27 and 28 hold the power-down flags for all analog signal operational amplifiers. Bits 0 to 7 of address 27 control the voltage follower for ADC7 to ADC0. Bits 0 to 6 of address 28 control the enable state of the VDC voltage follower (bit 0), VAC voltage follower (bit 1), DMCE0 current amplifier (bit 2), PFC current amplifier (bit 4), ADC buffer amplifier (bit 5), and reference voltage buffer amplifier (bit 6). The MSB (bit 7) of address 28 can be used to power-down the bias supply for all amplifiers. Disabling of the operational amplifiers is performed by setting individual flags to one, initially all flags are set to zero.

Please note that disabling the current amplifier of DMCE0 will in turn disable the three associated phase voltage read-back comparators.

5.2.12. SPI Unit Low-Pass Filter Settings

Low-pass filtering is available for both the AMG-XB403's system SPI unit and the MCU-controlled SPI unit. Address 29 holds the time constants for each SPI unit's input signal low pass filter. The value stated for SPILP specifies the filter depth of the low pass filter for the system SPI as a multiple of system clock cylces. The value stated for SPILP2 specifies the filter depth of the low pass filter for the MCU's SPI unit. The time base of the filter is the system clock. A value of 0 for SPILP deactivated low-pass filtering, this is the default value.

5.2.13. Assignment of Power Outputs and High Voltage Read-Back Pins

In order to accommodate any phase order with a minimum of wiring effort, the AMG-XB403's gate driver outputs and high voltage feedback signals are connected to freely programmable physical IC pins. They are assigned using addresses 30 through 35.

Address 30 contains the configuration for the three pins associated with the U-phase of DMCE0. Bits 0 to 2 configure the high-side drive signal pin. When set to a value of 1 through 6 UH will be assigned to one of pin M0O1 through M0O6. When set to a value of 0 the pin M0O1 is driven to a static low state, when set to 7 the pin M0O1 will be driven to a static high state.
Bits 3 to 5 configure the low-side drive signal pin. When set to a value of 1 through 6 UL will be assigned to one of pin M0O1 through M0O6. When set to a value of 0 the pin M0O2 is driven to a static low state, when set to 7 the pin M0O2 will be driven to a static high state.
Bits 6 and 7 configure the output state read-back pin RU. When set to a value of 1 through 3 RU will be assigned to one of pin M0R1 through M0R3. When set to a value of 0 no read-back pin will be assigned.

Addresses 31 and 32 contain the configuration for the three pins associated with the V-phase, and W-phase of DMCE0 respectively, assigned in analogous fashion to address 30. They control the assignment of VH, VL, WH and WL to M0O1 through M0O6, the static logic state of pins M0O3, M0O4, M0O5, and M0O6, and the position of the output state read-back pins RV and RW to M0R1 through M0R3.

5.2.14. DMCE and PFC Driver Signal Read-Back

Addresses 36 and 37 are used to read back the logic level at the driver outputs of both DMCE and the PFC, this can be used for system health checking.

The MSB of both addresses switches between analog and digital read back inputs. If M0RMUX is set to 1, analog comparators are used for reading back DMCE0 driver outputs, otherwise digital inputs are used. The analog and digital inputs rely in the same I/O-Pads, so switching between those does not affect pin layout configuration.

5.2.15. Band Gap Reference Tuning Setting

Address 38 holds the four bits wide band gap reference tuning setting, VBGTUN, allowing for the on-chip reference voltage generator to be adjusted in steps of 0.4% per LSB. The initial setting of VBGTUN is eight, corresponding to a native reference voltage. Tuning of the reference will ensure the correct setting of all derived regulated voltages.

5.2.16. OFI Direct Read-Back

Address 39 is used for reading the state of the M0OFI input pin, bypassing the configured polarity setting and low-pass filtering.

5.2.17. EEPROM Direct Access

Address 40 gives access to the serial interface of the EEPROM. Bit 0 is used for serially writing data to and reading data from the EEPROM. The serial clock signal is transmitted via bit 1.

5.2.18. MCU-Accessible Engineering Mode Enable

Bit 0 of address 41 holds the MCU-accessible engineering mode enable flag EME2. By setting EME2 to 1 the MCU can access the EEPROM via address 40 of the IO expander. To leave the self-programming mode EME2 is set to 0, this is the default value. This bit cannot be set by the SPI bus master.

5.2.19. ADC TUN Mode, DMCE Value Offset Selection

Bit 0 of address 42 holds the ADC tuning mode flag, TUNMODE. When set the ADC will perform a tune cycle with each conversion. By default TUNMODE is not set.

Bit 1 and bit 2 of address 42 must remain 0, this is the default value.

Bit 3 of address 42 holds the offset select flag, OFFSSEL, used to select the offset value of DMCE0 when set to 1, this is the default. For details see section 5.3, page 14.

5.2.20. ADC Sample Triggers for Test, Supply Voltage, and Temperature

Address 43 gives access to additional ADC request flags.

Bit 0, IREQ, is used to trigger sampling of the DC bus voltage, rectified AC voltage, DMCE's current values and the PFC controller's current value. Readiness is indicated by reading this flag as 0.

Bit 1, VDDREQ, is used to trigger sampling of 5V supply voltage. Readiness is indicated by reading VDDREQ as 0.

Bit 2, TREQ, is used to trigger sampling of on-chip temperature values. Readiness is indicated by reading TREQ as 0.

5.2.21. IO Expander Overview

Bit	7	6	5	4	3	2	1	0	Register name	Index address	Comment
Write/Read	ADDR[7:0]								IOEXPA	-	Address register
Initial value	0	0	0	0	0	0	0	0			
Write	-	-	-	-	-	-	-	-	IOEXPD	0	IC revision
Read	0	0	0	0	0	0	1	1			
Write/Read	-	-	CLK-LOCK	CLKOFF	reserved	LOCKED (read)	EME	TME		1	Engineering and test mode enable Lock indicator
Initial value	-	-	0	0	0	1	0	0			
Write	reserved	reserved	OFIRES0	OFIPOL0	reserved	reserved	DMCE0POL	DMCE0OE		2	DMCE output configuration
Read	reserved	reserved	OFIERR0	OFIPOL0	reserved	reserved	DMCE0POL	DMCE0OE			
Initial value	0	0	0	0	0	0	0	0			
Write/Read	SWCPRE[7:0]									3	SMPS pre-scaler low/high
Initial value	0	0	0	0	0	0	0	0			
Write/Read	-	SWCPRE[14:8]								4	
Initial value	-	0	0	0	0	0	0	0			
Write/Read	-	-	-	-	-	-	-	PFCPOL		5	PFC PWM output polarity
Initial value	-	-	-	-	-	-	-	0			
Write/Read	reserved	reserved			CNT0X2	DMCE0C1				6	DMCE C1 values; PWM double frequency
Initial value	0	0	0	0	0	0	0	0			
Write/Read	DMCE0C2[7:0]									7	DMCE0 C2 value
Initial value	0	0	0	0	0	0	0	0			
Write/Read	reserved									8	reserved
Initial value	0	0	0	0	0	0	0	0			
Write/Read	FLLDIV[7:0]									9	FLL down-scaler; low byte
Initial value	0	0	0	0	0	0	0	0			
Write/Read	-	-	-	FLLDIV[12:8]						10	FLL down-scaler; high byte
Initial value	-	-	-	0	1	0	0	0			
Write	MPREC[3:0]				-	-	-	-		12	maximum precision; current FLL precision
Read	MPREC[3:0]				PREC[3:0]						
Initial value	1	0	1	0	-	-	-	-			
Write	reserved	-	reserved		DTC0OVR	-	DTC0RNG[1:0]			13	Output comparator read-back and DTC setup
Read	reserved	reserved	reserved	reserved	DTC0OVR	RW0	RV0	RU0			
Initial value	0	0	0	0	0	0	0	0			
Write/Read	ADCOFF[7:0]									14..25	ADC offset registers
Initial value	0	0	0	0	0	0	0	0			
Write/ Read	-	-	-	-	ADCRES	PFCRES	DMCE0RES reserved	reserved DMCE0RES		26	Unit reset flags
Initial value	-	-	-	-	0	0	0	0			

Table 2a: IO port expander registers

Bit	7	6	5	4	3	2	1	0	Register name	Index address	Comment
Write/Read	ADC[0:7] OPAMPEN								IOEXPD	27	Operational amplifier and bias enable flags
Initial value	0	0	0	0	0	0	0	0			
Write/Read	BIASEN	Vref buffer E.	ADC buffer E.	PFC current E.	reserved	DMCE0 E.	VAC E.	VDC E.		28	
Initial value	0	0	0	0	0	0	0	0			
Write/Read	SPILP2[3:0]				SPILP[3:0]					29	SPI low pass filter values
Initial value	0	0	0	0	0	0	0	0			
Write/Read	M0RUCONF[1:0]		M0ULCONF[2:0]			M0UHCONF[2:0]				30	DMCE0 U-phase configuration
Initial value	0	0	0	0	0	0	0	0			
Write/Read	M0RVCONF[1:0]		M0VLCONF[2:0]			M0VHCONF[2:0]				31	DMCE0 V-phase configuration
Initial value	0	0	0	0	0	0	0	0			
Write/Read	M0RWCONF[1:0]		M0WLCONF[2:0]			M0WHCONF[2:0]				32	DMCE0 W-phase configuration
Initial value	0	0	0	0	0	0	0	0			
Write/Read	reserved		reserved			reserved				33	reserved
Initial value	0	0	0	0	0	0	0	0			
Write/Read	reserved		reserved			reserved				34	reserved
Initial value	0	0	0	0	0	0	0	0			
Write/Read	reserved		reserved			reserved				35	reserved
Initial value	0	0	0	0	0	0	0	0			
Write	M0RMUX	-	-	-	-	-	-	-		36	DMCE0 and PFC driver output read-back
Read	M0RMUX	PI	M0I[5:0]								
Initial value	0	-	-	-	-	-	-	-			
Write	reserved	-	-	-	-	-	-	-		37	reserved
Read	reserved	-	reserved								
Initial value	0	-	-	-	-	-	-	-			
Write/Read	-	-	-	-	VBGTUN[3:0]					38	Band gap reference adjust
Initial value	-	-	-	-	1	0	0	0			
Write	-	-	-	-	-	-	-	-		39	MxOFI direct read-back
Read	-	-	-	-	-	-	reserved	M0OFI			
Initial value	-	-	-	-	-	-	-	-			
Write	-	-	-	-	-	-	EE_SCLO	EE_SDAO		40	EEPROM access port
Read	-	-	-	-	-	-	-	EE_SDAI			
Initial value	-	-	-	-	-	-	0	0			
Write/Read	-	-	-	-	-	-	-	EME2		41	engineering mode enable - MCU only
Initial value	-	-	-	-	-	-	-	0			
Write/Read	-	-	-	-	OFFSSEL	-	-	TUN-MODE		42	ADC tune mode selection; DMCE offset selection
Initial value	-	-	-	-	0	0	0	0			
Write/Read	-	-	-	-	-	TREQ	VDDREQ	IREQ		43	Additional ADC functions port
Initial value	-	-	-	-	-	0	0	0			

Table 2b: IO port expander registers

5.3. Analog Data Aquisition Unit (ADAQ)

The Analog Data Aquisition block features the measurement of external voltages and their use for system control purposes. Its heart piece is a single 10 bit 2M samples/s SAR[1] ADC. The ADC is shared by the AMG-XB403's DMCE, PFC controller, and MCU. In addition, it contains measuring amplifiers, a multiplexer and an ADC interface.

5.3.1. ADC Interface

The ADC interface serves to arbitrate between all units requiring sample values and the ADC.

There is one general purpose ADC channel, a voltage monitor channel and a temperature monitor channel dedicated to the MCU. The conversion status and the results of these channels can be acquired by reading the ADC's IO registers via the on-chip bus.

The general purpose ADC channel conversions are triggered by port writes, the unit can raise an interrupt flag to indicate a completed conversion. The conversion status and result can be acquired by reading the ADC interface's IO registers. An analog multiplexer is used to select the desired internal or external signal source as shown in 'Figure 8'.

Every ADC channel has a systematic offset. Offset correction values are stored in the ADCOFF registers in the IO port expander module (see section 5.2, page 8). The ADC channel, PFC VAC, PFC VDC, chip-temperature and VDD are automatically corrected by their offset values when sampled.

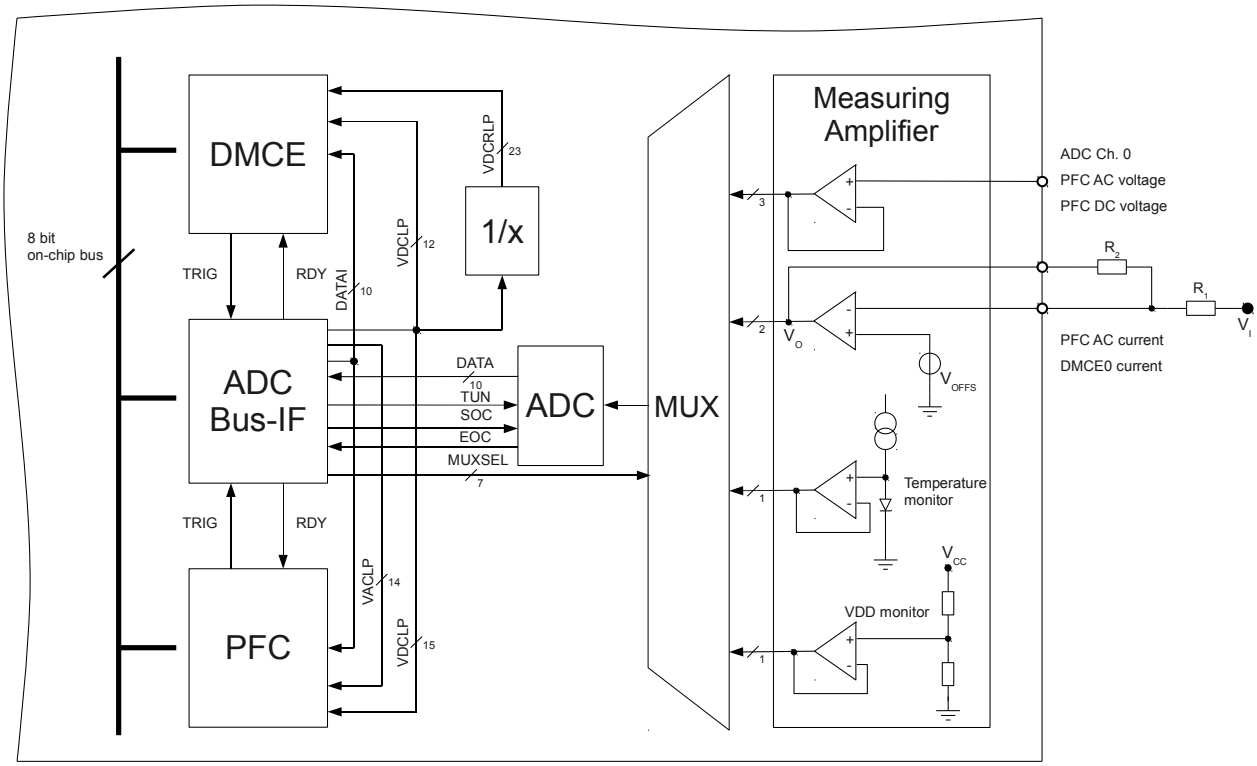


Figure 8: XB403 ADAQ block schematic

1 successive approximation register

5.3.2. Analog Input Signal Ranges

The input signal range at the analog input pins of the general purpose ADC channel, DC bus voltage, rectified AC voltage, DMCE's current signal and PFC AC current signal is a nominal input voltage range of 0...5V.

The ADC input voltage for the current signal inputs can be calculated as follows:

$V_O = -V_I \cdot \frac{R_2}{R_1} + V_{OFFS}$ and is shown in 'Figure 9' (refer to 'Figure 8', page 14). The V_{OFFS} of the DMCE's current channel and the PFC's current channel are different due to the DMCE's current may have either polarity whereas the PFC's current is always smaller than zero.

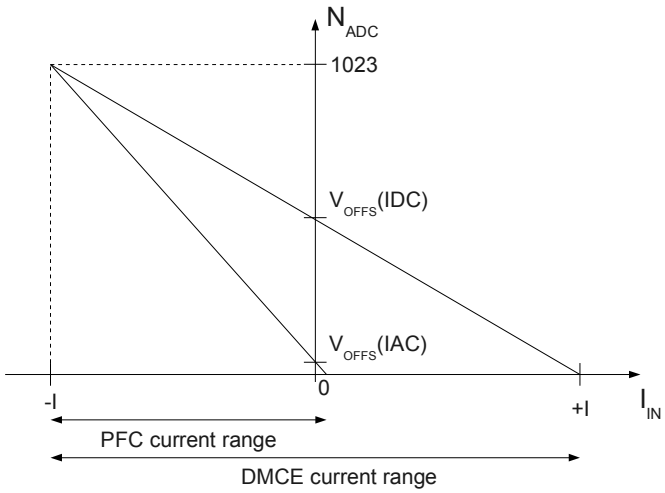


Figure 9: ADC current input characteristic

For the DC current channel VOFFS is set to nominally 445 mV and, for the AC current channel VOFFS is set to about 24 mV.

For improved noise rejection and accuracy the rectified AC voltage and the DC bus voltage values are digitally low-pass filtered using IIR filters, the low-pass filtered values VACLP and VDCLP are fed directly to the PFC controller. VDCLP is fed to the DMCE core and an inverter. The low-pass filtered signal value for VACLP and VDCLP can be calculated as follows:

$$VACLP_N = \frac{15}{16} \cdot VACLP_{N-1} + VAC_N \qquad VDCLP_N = \frac{31}{32} \cdot VDCLP_{N-1} + VDC_N$$

5.3.3. Triggering Analog Signal Acquisition

The sampling of the general purpose ADC channels 0 through 7 is triggered by writing the connected flag within the register ADCREQ as one. Completion is indicated by reading the connected flag in register ADCINT as one.

The rectified AC voltage and the DC bus voltage are sampled at the analog inputs VAC and VDC. Sampling of these signals is triggered automatically every 1024 cycles of the system clock if DMCE core is not reset. Sampling can also be triggered by setting the indirect request flag, IREQ, via the IO expander (see section 5.2.20., page 11). The raw sample value can be read via the ADC's registers (see 'Table 3', p. 17), the low-pass filtered value using the respective DMCE and PFC registers.

Sampling of DMCE current offset values is triggered when a zero-vector is set by the DMCE's PWM unit. It is performed autonomously at the beginning of each PWM cycle when all low-side switches of the PWM module are in ON-state, if the DMCE is not reset. Sampling can also be triggered by setting the indirect request flag, IREQ, via the IO expander (see section 5.2.20., page 11). The raw sample value can be read using the ADC unit's registers as shown in 'Table 4', page 17, the low-pass filtered value using the DMCE's register.

An on-chip temperature sensor provides a 0...1.8V analog signal that is sufficient for measurements with an accuracy of 1.25K/bit. The calibration setting of the temperature sensor is stored in the EEPROM. Temperature sampling is triggered by setting the TREQ flag within the IO expander, see section 5.2, page 8. The TREQ flag will be reset to 0 when a temperature sample has been acquired.

A voltage divider with an accuracy of 8mV/bit is used to monitor the voltage at the A5V pin. Voltage sampling is triggered by setting the VDDREQ flag within the IO expander, see section 5.2, page 8. The VDDREQ flag will be reset to 0 when a A5V sample has been acquired. The supply voltage can be calculated as follows:

$$A5V = \frac{ADC_{VMON} \cdot 5 \cdot 1.8\text{V}}{1023}$$

5.3.4. Accessing ADC Sample Data, ADC Interrupts

The ADC is configured through the ADCCONF register shown in 'Table 3, page 17. The ADC channel to be addressed is chosen by setting the four least significant bits of the ADCCONF register (as shown in detail in 'Table 4', page 17). By setting ADCIE to 1 ADC interrupt generation is enabled.

The interrupt flags for ADC channel 0 to 7 can be accessed through the interrupt flag register ADCINT. To reset a specific interrupt flag the corresponding bit in the interrupt reset register ADCIRES must be written as 1.

The two least significant bits of the conversion result are accessed through bits four and five of ADCCONF. The eight most significant bits are held by the data register ADCD. The most significant bit's value must always be retrieved first.

The ADC can be performed in either synchronous or asynchronous mode. By setting the MSB in ADCCONF to 0 synchronous mode is activated. In synchronous mode, ADC channel 0 through 7 will only be sampled after an acquisition of the DC bus voltage to reduce distortions caused by the power stage. In synchronous mode, the sampling rate is limited to the DMCE's PWM frequency. The sampling of ADC channel 0 through 7 will be performed at the next available time, i.e. when no ADC channel with a higher priority is pending.

Any ADC acquisition initiated by either DMCE, or the PFC controller has a higher priority than ADC channel 0 through 7. Only temperature measurements; D5V measurements and tuning cycles have a lower priority. The pending ADC channel with the lowest number will always be sampled first (refer to 'Table 4').

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	**ASYNC**	**ADCIE**	**SEL[1:0]**		**0**	**0**	**0**	**ADCCH[0]**	**ADCCONF**	ADC configuration register
Read	**ASYNC**	**ADCIE**	**ADCD[1:0]**		**0**	**0**	**0**	**ADCCH[0]**		
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**IRCH[0]**	**ADCIRES**	ADC interrupt reset register; Channel 0
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Read	**ADCD[9:2]**								**ADCD**	ADC data register
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**REQCH[0]**	**ADCREQ**	ADC request register; Channel 0
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Read	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**INTADCCH [0]**	**ADCINT**	ADC interrupt flag register; Channel 0
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 3: XB403 ADC registers

Code	Priority	ADC channel offset register address (IO expander module)	ADCCH[3]	ADCCH[2]	ADCCH[1]	ADCCH[0]	Input pin	Triggered by
0	6	14	0	0	0	0	ADC Channel 0	ADC request register
8	14	22	1	0	0	0	VDC	Hardware timer
9	15	23	1	0	0	1	VAC	
10	2	-	1	0	1	0	M0IDC	DMCE0
12	1	-	1	1	0	0	PFCI	PFC
13	16	24	1	1	1	0	on-chip temperature sensor	IO expander flags
14	17	25	1	1	0	1	D5V	
15	4	-	1	1	1	1	M0IDC	DMCE0

Table 4: XB403 ADC channel encoding in ADCCONF[3:0]

Delay tuning is the process of adjusting internal signals of the ADC to compensate for temperature and process-dependent delays. Tuning is performed once, after IC reset and must be repeated in the face of large temperature changes. Upon reset the tuning mode is off, this mode can be enabled by setting the TUNMODE flag within the IO expander, see section 5.2, page 8.

5.4. Digital Motor Control Engine (DMCE)

The AMG-XB403 DMCE's features are:

- ❑ 32 bit Motor Control Processing Unit (MCPU) with single precision floating point format running at 64 MHz nominally
- ❑ 384 words x 24 bits program memory
- ❑ 128 floating point (32 bit) registers
- ❑ Register-mapped hardware ports to control PWM unit from MCPU
- ❑ 2 lookup-tables (32 floating point entries each)
- ❑ PWM unit including phase current reconstruction and dead time compensation
- ❑ Safety shut-down on external error event (e.g. over-current)

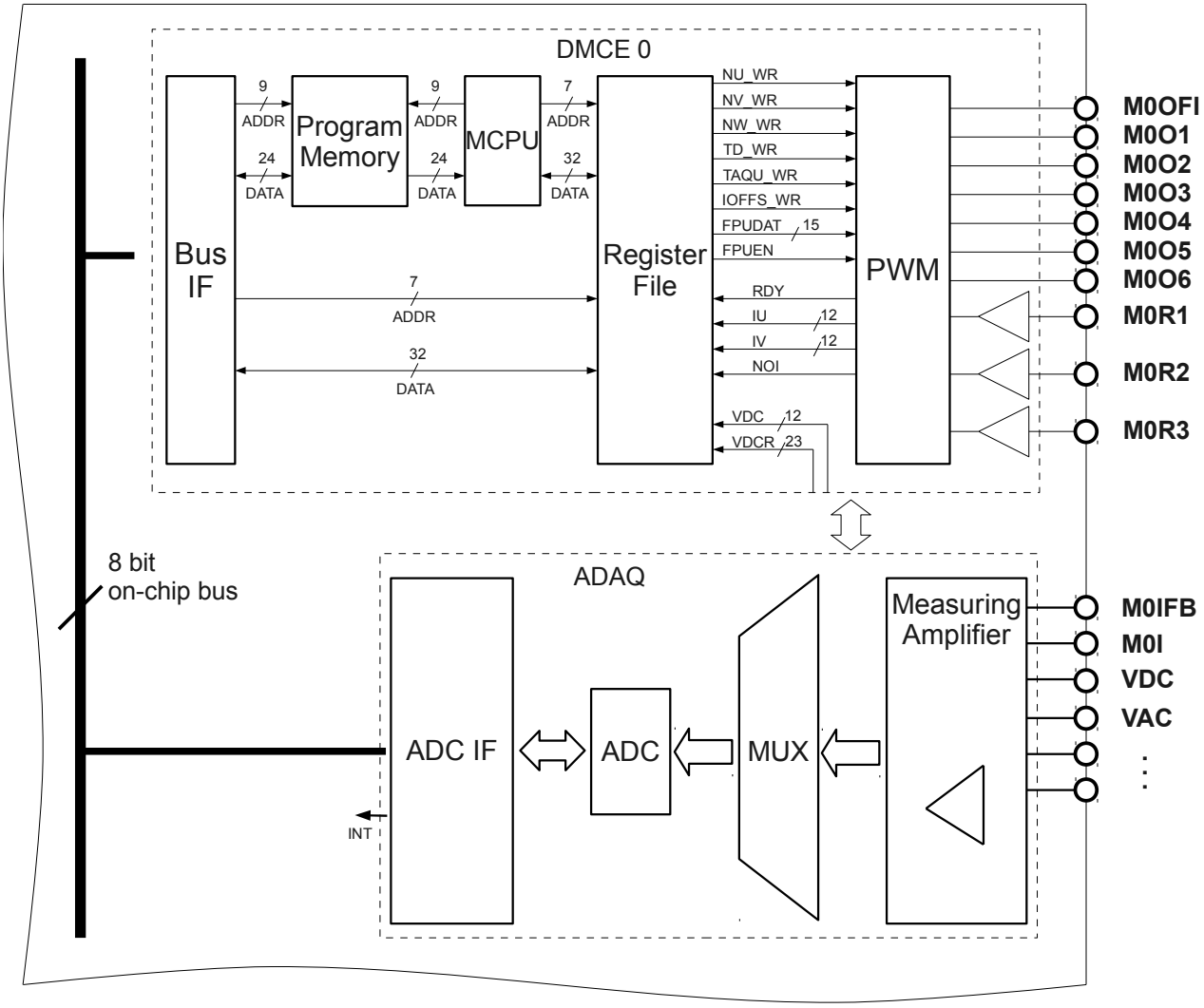


Figure 10: DMCE block schematic

5.4.1. Motor Control Processing Unit (MCPU)

The DMCE contains a single precision floating point Motor Control Processing Unit (MCPU) capable of executing computationally expensive motor control algorithms. The Harvard Architecture MCPU executes programs stored in a 384 words deep program memory SRAM.

The MCPU can execute eight different operations. They are performed on the MCPU's 128 directly addressed registers including access to 64 indirectly addressed registers, and register-mapped hardware ports. Each register represents a 32 bit single precision floating point number.

The eight supported commands have a uniform width of 24 bits and the format is shown in 'Figure 11'. The following table gives an overview of each command's function.

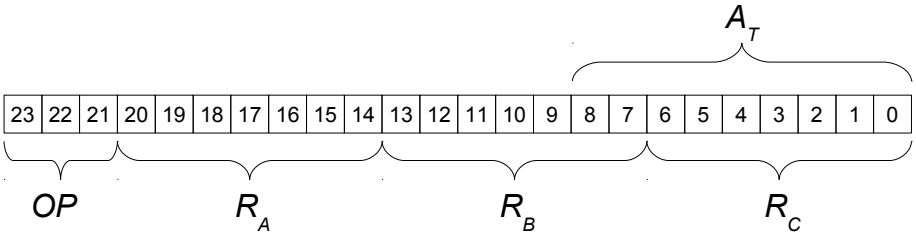


Figure 11: MCPU command format

OP	Operation	Description
0	$R_C = R_A + R_B$	Add R_A and R_B and store result in R_C.
1	$R_C = R_A - R_B$	Subtract R_B from R_A and store result in R_C.
2	$R_C = R_A \cdot R_B$	Multiply R_A with R_B and store result in R_C.
3	$R_C = \sin(R_A - 2)$	Calculate sine of (R_A-2) and store result in R_C. R_A must satisfy $0 \le R_A \le (\pi/2+0.1)$.
4	$R_C = \cos(R_A' - 2)$	Get the corresponding cosine value for the last sine calculation and store the result in R_C. For angles greater that (π/2) the command returns cos(R_A'-2)+4. Consequently, 4 needs to be subtracted from the result in these cases.
5	jump A_T	Perform an unconditional jump to the address given in A_T.
6	jump A_T if ($R_A < 0$)	Perform a jump to the address given in A_T if R_A is negative.
7	$R_C = \sqrt{(R_A)}$	Calculate square root of R_A and store result in R_C.

Table 5: MCPU commands

The floating point values used by the AMG-XB403 are similar to the IEEE754 standard. 'Figure 12' shows the bit assignments. "*S*" denotes the number's sign. 0 stands for a positive sign and 1 for a negative sign. The number's exponent "*E*" is offset by +127 i.e. exponents ranging from -127 to 128 can be processed. The mantissa "*M*" is normalized and always includes a hidden 1. The mantissa's value becomes 1.*M* in binary format.

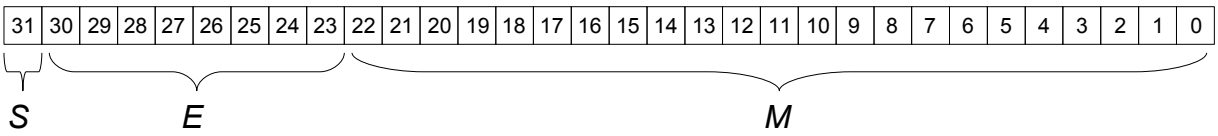


Figure 12: Floating point number format

A floating point number's value can thus be calculated as $((S)?-1:1)\cdot(1.M)\cdot 2^{E-127}$. For the sake of simplicity, the MCPU can represent neither ±0 nor infinity and has no means of reporting

exceptions. Consequently, any algorithm implemented to the AMG-XB403 must be avoided or handled within these limitations in order to prevent malfunction.

5.4.2. Register File

Out of the 128 directly addressed registers, registers 0 through 9 are special purpose registers reserved for the communication with the PWM unit (see 'Table 6', page 20).

Registers can be read from and written by the DMCE and via the 8 bit on-chip bus. Two times 32 indirectly addressed registers can be read from and written to via registers 8 and 9. The address index is set via register 6. For the details of indirect addressing refer to 'Table 6'.

Address	Register name	Read value	Register name	Write value
0	RDY	Ready flag, sign bit set at the start of each PWM cycle	NU	U-phase PWM value in bits [14:0] of mantissa
1	IU	U-phase current produced by virtual current sensor, value in bits [14:0] of the mantissa, offset by 8192	NV	V-phase PWM value in bits [14:0] of mantissa
2	IV	V-phase current produced by virtual current sensor, value in bits [14:0] of the mantissa, offset by 8192	NW	W-phase PWM value in bits [14:0] of mantissa
3	VDC	DC bus voltage, value in bits [11:0] of the mantissa	TD	PWM dead time TD measured in MCPU clock cycles, value in bits [8:0] of mantissa
4	VDCR	1 divided by DC bus voltage, value contained in bits [22:0] of the mantissa	TAQU	Minimum current sample acquisition time TAQU measured in MCPU clock cycles, value in bits [14:0] of mantissa
5	IOFFS	Zero current offset contained in bits [13:0] of mantissa	IOFFS	Zero current offset, value in bits [13:0] of mantissa
6	NOI	No current obtained if sign bit of this register is set	INDEX	Index for indirect addressing, value in bits [22:18] of mantissa
7	MCPUEN	MCPU enable flag, bit 0 of the mantissa	MCPUEN	MCPU enable flag, if bit 0 of mantissa is set to zero MCPU will be reset
8	IREGA	Read from indirect addressed registers A	IREGA	Write to indirect addressed registers A
9	IREGB	Read from indirect addressed registers B	IREGB	Write to indirect addressed registers B

Table 6: Special purpose registers

5.4.3. PWM Unit

The PWM unit is responsible for generating the PWM timings for each of the three phases according to the duty cycles requested by the MCPU. The voltage vectors are selected by the PWM unit to enable a safe reconstruction of the motor's phase currents. Duty cycles are specified by the values of NU, NV, and NW written to register addresses 0 to 2 (refer to 'Table 6'). For a duty cycle DC between 0 and 1 the floating point value for NU is calculated as follows: $NU = 2 + DC \cdot 2^{-22}$. The same equation applies to both NV and NW.

Moreover, the PWM unit performs autonomous dead-time compensation by means of the internally assigned feedback pins RU0, RV0, RW0 ensuring the equality of intended and factual duty cycle for each phase.

The PWM unit is also in charge of triggering voltage and current sampling at the appropriate time slots within each PWM cycle and reconstructing the U and V phases' currents (IU, IV). The 12 bit wide values of IU and IV are available to the MCPU core through the special purpose registers at addresses 1 and 2 (refer to 'Table 6').

The special purpose register at address 3 contains the DC bus voltage floating point value VDC, which is calculated as follows: $VDC = 2 + VDC_{ADC} \cdot 2^{-22}$. With VDC_{ADC} being the DC bus voltage integer value from the ADC. The inverted DC bus voltage's floating point value VDCR can be accessed at address 4 and is calculated as follows: $VDCR = 2 + \frac{1}{VDC_{ADC}} \cdot 2^{-22}$.

The PWM unit is capable of indicating a faulty current reconstruction by means of the NOI flag in register 6. The 18 least significant bits form the no-current error vector NOIV. The most significant seven bits of them indicate an error in calculation of the duty cycle. The three least significant bits indicate an error in dead-time compensation. The NOI flag is set if any bit of NOIV is set.

The PWM unit requires the offset value of the current channel operation amplifier for current reconstruction. This floating point value called IOFFS has to be written to address 5 of the register file and is calculated as follows: $IOFFS = 2 + O \cdot 2^{-22}$. With O being the ADC's integer value with zero current.

In order to prevent an inverter shoot-through the PWM unit inserts an adjustable delay between the high-side and low-side active states. The delay value is set via register 3 and is a 9 bit wide multiple of the MCPU's clock cycle. The floating point value of this dead-time register is calculated as follows: $TD = 2 + t_{delay} \cdot f_{FPU} \cdot 2^{-22}$.

A minimum sample acquisition time is guaranteed by setting the value of register 4, TAQU. The acquisition time is a 15 bit wide multiple of the MCPU's clock cycle. The floating point value of the acquisition time register TAQU is calculated as follows: $TAQU = 2 + t_{aqu} \cdot f_{FPU} \cdot 2^{-22}$.

5.4.4. Accessing the DMCE via the On-Chip Bus

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write/Read	D0[7:0]								M0DAT0	Data register D0 Bits [7..0]
Initial value	0	0	0	0	0	0	0	0		
Write/Read	D1[7:0]								M0DAT1	Data register D1 Bits [15..8]
Initial value	0	0	0	0	0	0	0	0		
Write/Read	D2[7:0]								M0DAT2	Data register D2 Bits [23..16]
Initial value	0	0	0	0	0	0	0	0		
Write/Read	D3[7:0]								M0DAT3	Data register D3 Bits [31..24]
Initial value	0	0	0	0	0	0	0	0		
Write	ADR[7:0]								M0ADR	Address register
Initial value	0	0	0	0	0	0	0	0		
Write	-	-	-	-	ADR[8]	OP[2:0]			M0SC	Status and command register
Read	-	-	-	-	-	-	-	START		
Initial value	-	-	-	-	-	-	-	0		
Write	-	-	-	-	-	-	-	0	reserved	reserved
Read	-	-	-	-	-	-	-	0		
Initial value	-	-	-	-	-	-	-	0		

Table 7: DMCE access registers

The DMCE has six dedicated 8 bit registers on the 8 bit on-chip bus, refer to 'Table 7'. The DMCE can only be accessed via the SPI interface if the IC is not locked, see section 5.7.1, page 34. Four data ports are used to write and read program or register data. The 7 bit wide register address is set using the address register. The 9 bit wide program memory address is set using the address register for the eight least significant bits and bit three of the command register for the most significant bit. Both the read and write operations are triggered by writing the status and command port's three least significant bits. All available commands are summarized in 'Table 8'.

When the DMCE is disabled the on-chip bus has read and write access to the program memory for debugging purposes.

Code	Description
0	No valid command, zero indicates finished command on register reads
1	Load command ROM value to data registers D0, D1, D2
2	Load data registers D0, D1, D2 to DMCE command ROM
3	Load data registers D0, D1, D2, D3 to DMCE data register
4	Load register value to data registers D0, D1, D2, D3
5	Load PWM cycle counter[2] to data registers D0, D1, D2, D3
6	Load program counter to data register D0, D1
7	Load MCPU idle indicator to LSB of data register D0

Table 8: DMCE command register codes

Normally, program memory and register file information are stored in the EEPROM and loaded into the respective memory by the MCU.

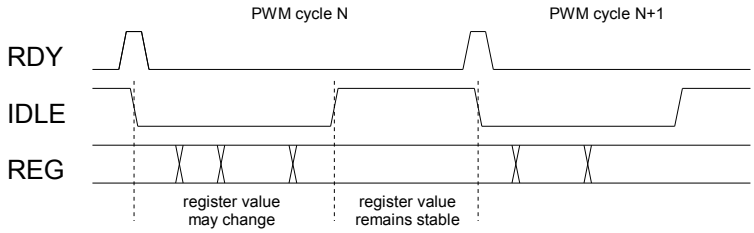


Figure 13: Idle indicator

The idle indicator is used when monitoring register and is reset by the ready flag. While the idle flag is 1, register values remain stable (see 'Figure 13'). Therefore, it is only guaranteed that register values of a single PWM cycle are read at a time.

2 Number of PWM cycles since reset

5.4.5. Power Stage Over-Current Protection

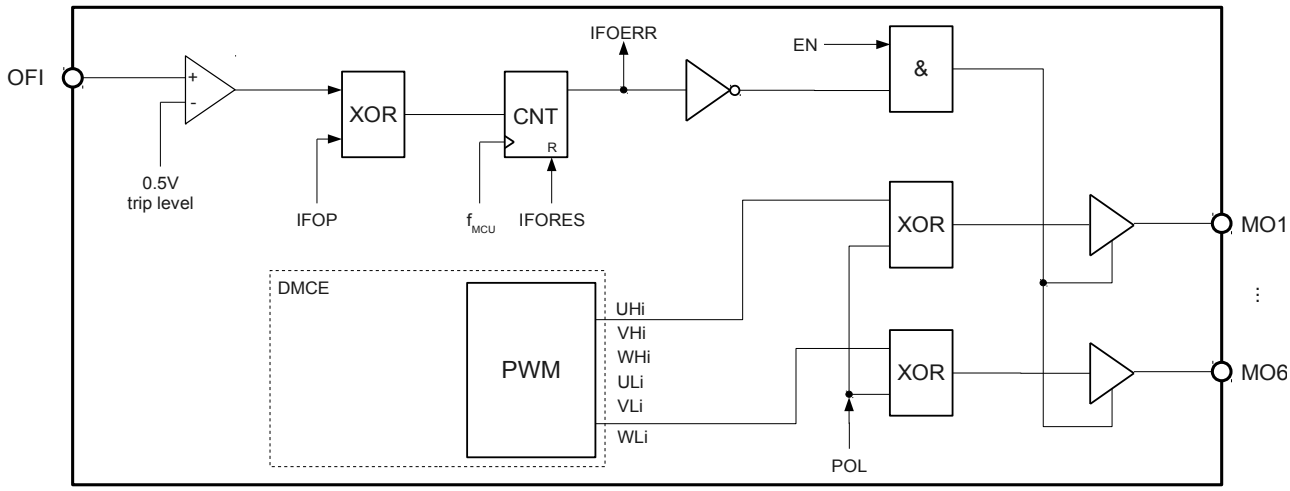


Figure 14: Over-current protection block schematic

In the event of over-current a dedicated hardware unit for the DMCE will turn off all logic outputs, indicate this error via a dedicated IO expander flag (see section5.2.16., page 11), and optionally trigger the shared DMCE/PFC interrupt of the MCU.

If the power stage connected to the AMG-XB403 provides a digital over-current fault indicator (OFI) output, the AMG-XB403's OFI input acts as a digital input with a logic threshold level of 0.5V nominally.
If no digital over-current fault indicator flag is available, the OFI input acts as an analog comparator with a trip level of 0.5V nominally, which is the standard power stage module trip level. In this case the shunt resistor connected to the power stage needs to be dimensioned accordingly.
The AMG-XB403 can handle power stage modules having any combination of driver and over-current feedback polarity, refer to 'Figure 14' (above). As described in section 5.2, page 8 five dedicated signals control the polarity of drivers and over-current fault input, error indication and reset, and enabling of the driver outputs.

The over-current fault input signal OFI is low-pass filtered digitally, it needs to continually indicate an error for at least 1µs nominally to detect an over-current condition.

The pin order of the driver and high voltage feed-back pins can be configured freely as described in section 5.2.13., page 10.

5.5. Power Factor Correction Controller (PFC)

The AMG-XB403 PFC's features are:

- ❑ Fully configurable digitally controlled PFC
- ❑ Constant frequency, continuous conduction mode (CCM)
- ❑ 16 bit RISC core
- ❑ 256 words x 22 bits program memory
- ❑ 64 words x 16 bits data memory
- ❑ Adjustable PWM-Frequency, Range: 5kHz ~ 150kHz, Spread spectrum option
- ❑ On-chip over-current and short-circuit protection, brown-out control
- ❑ On-chip AC-phase-locked sinusoidal oscillator (45Hz ~ 65Hz) for improved AC line noise reduction

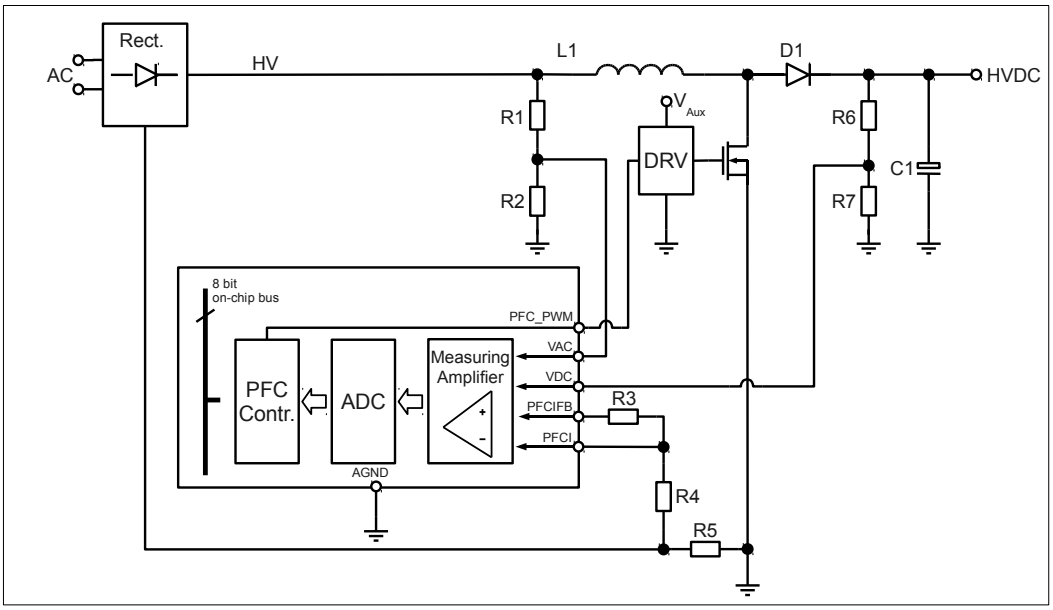


Figure 15: Simplified PFC controller application schematic

The AMG-XB403's PFC controller supports standard boost-topology power factor correction depicted in 'Figure 15'. It is designed to achieve IEC61000-3-2 level PFC performance when combined with appropriately dimensioned power electronics and line filtering. The PFC controller interacts directly with the ADC unit to acquire the rectified AC voltage, the DC bus voltage, and the rectified AC current value. It generates a pulse width modulated logic-level switching signal connected to an external gate driver circuit. A detailed block schematic of the PFC controller is shown in 'Figure 16', page 25.

The PFC controller is built around an autonomous 16 bit RISC processor core supporting 15 different instructions (see 'Table 9', page 26). It runs at the same clock frequency as the DMCE. A 22 bits wide, 256 words deep SRAM serves as the processor's program memory. All instructions operate on a 6 bit data address space which is divided into address ranges for general purpose registers, constants, hardware ports, and values stored in a dedicated SRAM (see 'Table 10', page 28).

The unit includes a PWM generator that can be used to generate frequencies in a range of 5~150 kHz. A hardware-based pseudo random number generator can be used to generate a spread-spectrum PWM signal. The PFC controller is able to set an interrupt line to indicate errors to the MCU. A hardware-based counter with an adjustable increment can be used to form a phase-locked signal that is synchronous to the mains frequency.

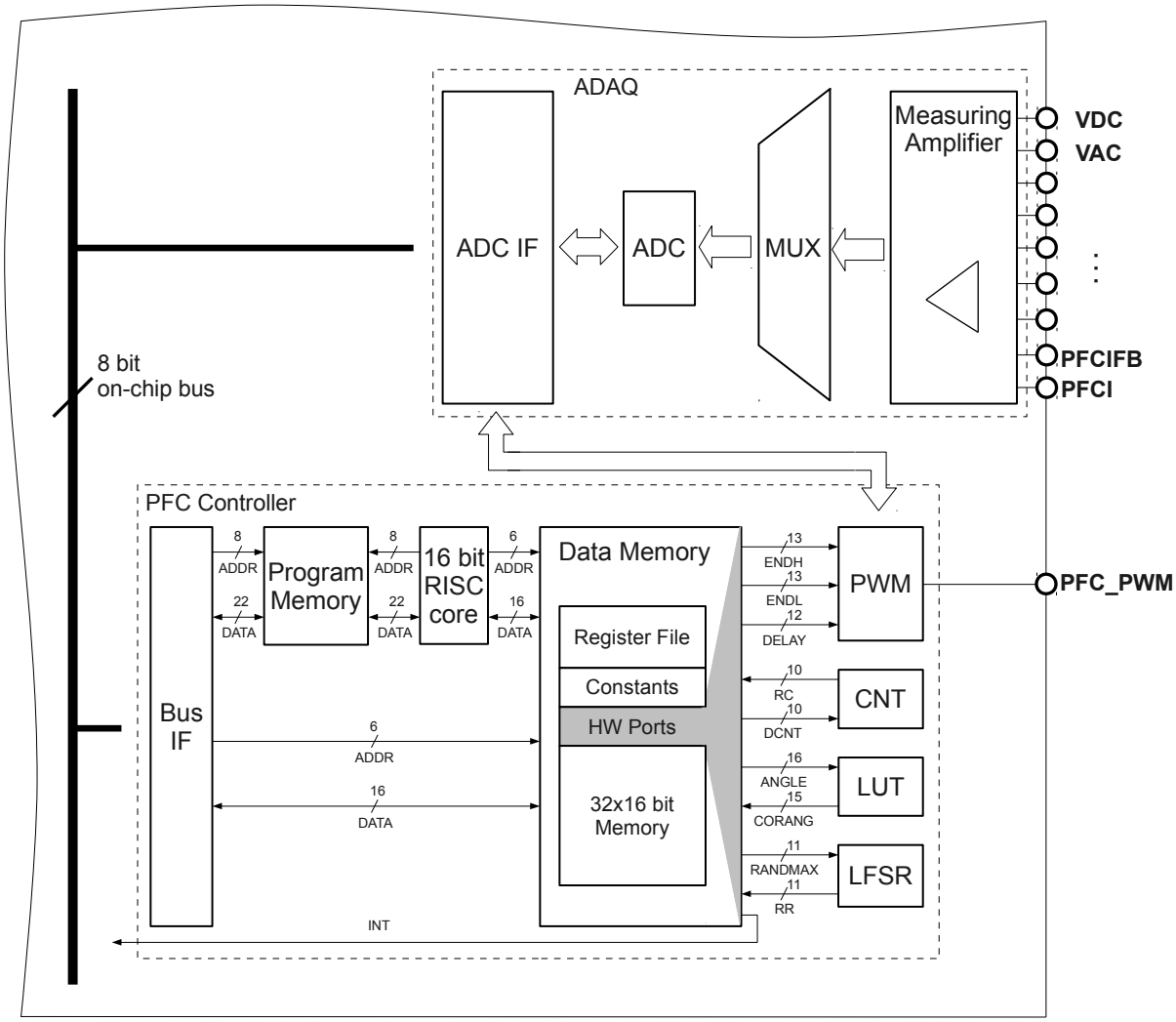


Figure 16: PFC controller block schematic

5.5.1. PFC Instruction Set Description

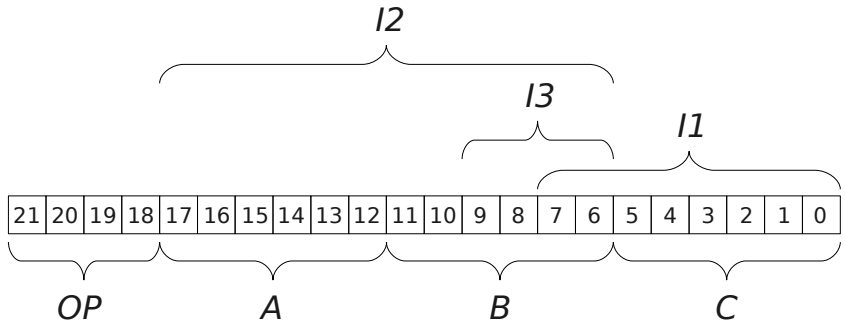


Figure 17: PFC command format

The 15 supported commands have a uniform width of 22 bits and the format shown in 'Figure 17'. A, B, C being register or memory locations upon which the operation OP is executed. Operations can also use the immediate values I1, I2, and I3 which are part of the command word. 'Table 9', page 26 provides an overview of each command's function.

OP	Mnemonic	Operation	Clock cycles	Description
0	JMP	jump *I1*	1	jump to 8 bit address immediate *I1*
1	BRN	jump *I1* if $(A<0)$	1 (2)	jump to 8 bit address immediate *I1* if register/memory *A* is negative
2	BRP	jump *I1* if $(A\geqslant 0)$	1 (2)	jump to 8 bit address immediate *I1* if register/memory *A* is positive
3	BRZ	jump *I1* if $(A=0)$	1 (2)	jump to 8 bit address immediate *I1* if register/memory *A* is zero
4	ST	$C=A$	1 (2)	store register/memory *A* in register/memory *C*
5	ADD	$C=A+B$	1 (2)	add register/memory *A* and register *B* and store result in register/memory *C*
6	SUB	$C=A-B$	1 (2)	subtract register *B* from register/memory *A* and store result in register/memory *C*
7	DIV	$ANGLE=CORD\_I:CORD\_Q$	50	divide *CORD_I* by *CORD_Q* in *CNT (1-15)* iterations, result stored in *ANGLE; for iterations register CNT is used, begin with setting ANGLE to Zero*
8	CORD	$CORD\_I=23167\cdot\sin(\frac{ANGLE}{2^{14}}\cdot\frac{\pi}{2})$ $CORD\_Q=23167\cdot\cos(\frac{ANGLE}{2^{14}}\cdot\frac{\pi}{2})$	80	execute cordic algorithm on *CORD_I*, *CORD_Q*, etc. in *CNT* 1-15 iterations, result stored in *CORD_I*, *CORD_Q; for iterations register CNT is used*
9	MUL	$M=A\cdot B$	1 (2)	multiply two registers *A*, *B* or a memory *A* and a register *B* and store result in internal register *M*
10	SHMR	$C=M\gg I3$	1	shift multiplier result in internal register *M* right by *I3* bits and store result in target memory/register *C*
11	SHL	$C=A\ll I3$	1 (2)	shift register/memory *A* left by *I3* bits and store result in register/memory *C*
12	SHR	$C=A\gg I3$	1 (2)	shift register/memory *A* right by *I3* bits and store result in register/memory *C*
13	STI	$C=I2$	1	store immediate *I2* (max. 12 bit) in register/memory *C*
14	SYNC	wait while $(PFC\_IAC==0)$	1	wait for *PFC_IAC* impulse

Table 9: PFC instruction set (The clock cycle values in braces denote the number of clock cycles if an SRAM value needs to be read during command execution.)

5.5.2. PFC Data Memory Description

The structure of the data memory is shown in 'Table 10'. The first seven addresses of the PFC controller's data memory hold general purpose registers which can be used in any combination to execute commands in a minimum number of clock cycles. Please note, that register values are modified by sine value computation and division. Consequently, their value must be rescued to data memory as required. Addresses 8 through 15 provide access to eight constant read-only values. Addresses 16 through 29 are used for hardware ports which are described hereafter.

ADDR	Type	Name or value	Number of valid bits
1	Registers	TMP	16
2		CORD_I	16
3		CORD_Q	16
4		ANGLE	16
5		ANGLESUM	16
6		CNT	16
7		TESTREG	16
8	Constants	0	1
9		1	1
10		512	10
11		8176	13
12		9948	14
13		700	10
14		430	9
15		4095	12
16	Hardware ports	VAC	14
17		VDC	15
18		IAC	10
19		CORANG	15
20		RR	11
21		RANDMAX	11
22		PFC_IAC	1
23		DELAY	12
24		PFC_EN	1
25		RC	10
26		INTERRUPT	1

ADDR	Type	Name or value	Number of valid bits
27		DCNT	10
28		ENDL	13
29		ENDH	13
32..63	SRAM	SRAM0 ... SRAM31	16

Table 10: Data memory overview (constants and hardware ports are used as registers)

The PFC controller uses center-aligned PWM signals as shown in 'Figure 18'. The start and end of a PWM cycle are determined by the PWM reference counter's maximum value which equals the value of the ENDH register divided by two.

The PWM period has thus a multiple of $ENDH / f_{FPU}$. ENDH is double buffered, i.e. its value can be set at any time by writing the value to the buffer ENDH, but will only be applied to the internally used register ENDHi in the next PWM cycle. Consequently, randomized duty cycles can be set asynchronously. The duty cycle of the PWM is determined by the ratio of ENDHi and ENDLi, i.e. $DC = ENDLi / ENDHi$. ENDL is double buffered as well and will be updated to its internally used register ENDLi at the same time as ENDH to ENDHi (see 'Figure 18').

Due to the low-pass filtered characteristic of the current signal its sampling must be delayed relative to the PWM signal. Setting the value of the DELAY register will postpone current sampling by an adjustable number of PFC clock cycles. Each sampling of the rectified AC current is followed by a sampling of the rectified AC voltage. The DC bus voltage value is updated whenever a new sample delivered to the DMCE core. The PFC has no means of triggering DC voltage value sampling.

Normally, the calculation and sample acquisition are synchronized by the means of the PFC controller's SYNC operation which halts execution until the LSB of the PFC_IAC register becomes 1, indicating the completion of rectified AC current sampling. The PFC_IAC register can also be read directly.

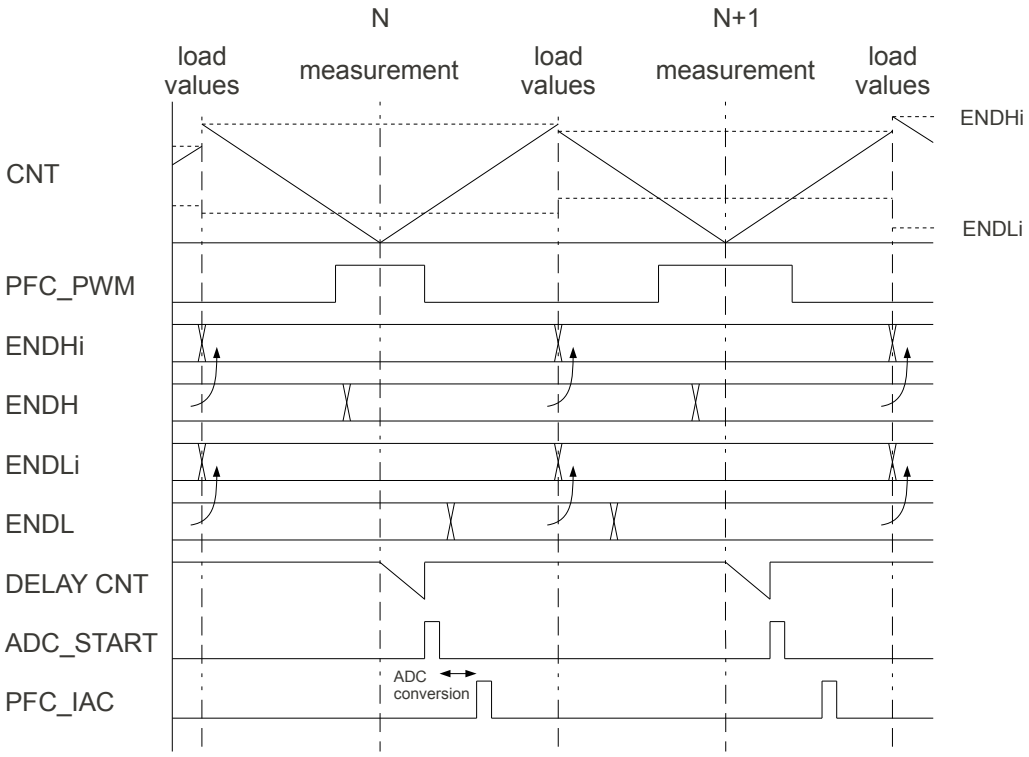


Figure 18: PFC's PWM cycle

The AD-converted, rectified AC voltage, DC voltage, and rectified AC current are buffered in and can be read from the registers VAC, VDC and IAC. As shown in 'Figure 18' the rectified AC current is sampled in the middle of the PFC transistor's on-period which coincides with the zero-crossing of the PFC's reference counter. The PFC's PWM output PFC_PWM can be inverted by setting the PFCPOL bit in the IO port expander (see'Table 2', page 13).

The PFC controller's pseudo random number generator makes use of an LFSR[3] capable of producing values with a length between four and eleven bits. A new random value will be generated each time the RR is read. The random values' length is set through the RANDMAX register, its most significant set bit determines the length of the generated value. With N denoting the position of the most significant set bit, a pseudo random value between one and $(2^{(N+1)}-1)$ is calculated. If RANDMAX is smaller than 16 a four bit value will be generated. An optimum feedback seed is chosen automatically for any bit width. The value returned by RR may be greater than RANDMAX only because the most significant set bit is taken into account by the PFC algorithm. The PFC algorithm may reread RR if a value greater than RANDMAX is encountered.

The read/write register RC returns a 10 bit wide reference counter value which is incremented once every DCNT clock cycles. DCNT has a width of ten bits and can be adjusted to obtain a variable frequency ramp signal on RC. The RC can usually be employed to generate a clean reference sine value for power factor correction.

The PFC controller can signal an interrupt to the MCU by setting the LSB of the INTERRUPT hardware port to 1. Upon interrupt execution, the interrupt flag must be cleared by either the PFC's algorithm or the MCU by setting the LSB of the INTERRUPT register to 0.

The PFC_EN hardware port is reserved for enabling and disabling power factor correction. The

3 Linear feedback shift register

PFC algorithm is in PLL mode when the LSB of PFC_EN is set to 0. In this state the reference counter is synchronized with the mains frequency, the PWM output is not modulated. Power factor correction is started by setting the LSB of PFC_EN to 1. The PWM output will be modulated accordingly.

The value of the CORANG register returns a lookup-value of an incremental angle used when performing a CORDIC algorithm for sine and cosine calculation. The value is addressed by the four least significant bits of the CNT register. This register is meant for the CORDIC hardware's internal use only.

Addresses 32 to 63 of the data memory are used to access a 16 bit wide 32 words deep data SRAM. In contrast to data registers, reading constants and ports from the SRAM requires two clock cycles. Moreover, a data SRAM address may not be used in arbitrary combination with other values. See 'Table 9', page 26 for a list of all valid commands.

5.5.3. Accessing the PFC Controller via the On-Chip Bus

The PFC controller can only be accessed via the SPI interface if the IC is not locked, refer to section 5.7.1, page 34. As shown in 'Table 11'; two 8 bit registers, one data register, and one command and status register can be used to communicate with the PWM. When writing data to the PFC controller it is written to the data register before executing the write by sending the appropriate command to the command and status register. The data is read from the PFC controller by sending the appropriate read command to the command and status register and reading back one byte of the data register. Values wider than eight bits are transferred using multiple 8 bit transfers.

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write/Read	**DAT[7:0]**								**PFCD**	Data register
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**CMD[7:0]**								**PFCSC**	Status and command register
Read	**ITCNT[6:0]**							**START**		
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 11: PFC controller registers

Command	Description
0x11	set address register
0x22	set buffer register [7:0] to value in data register
0x33	set buffer register [15:8] to value in data register
0x44	set buffer register [21:16] to value in data register
0x55	set data register to value in buffer register [7:0]
0x66	set data register to value in buffer register [15:8]
0x77	set data register to value in buffer register [21:16]
0x88	set buffer register [15:0] to PFC data RAM or register
0x99	set data from PFC data RAM or port to buffer register [15:0]
0xAA	set buffer register [21:0] to PFC program RAM
0xBB	set data from PFC program RAM to buffer register [21:0]
0xCC	enable PFC controller
0xDD	disable PFC controller
0xEE	set data register to value of the PFC program counter

Table 12: PFC bus interface command codes

5.6. Switch Mode Power Supply (SMPS)

The AMG-XB403 SMPS's features are:

- ❑ Constant frequency, discontinuous conduction mode (DCM)
- ❑ Primary coil current control input
- ❑ One directly regulated output voltage
- ❑ Other output voltages (depending on the transformer) indirectly regulated
- ❑ Independent low frequency start-up clock source
- ❑ Automatic clock selector to enable system clock after start-up
- ❑ Clock-scaling for higher PWM-Frequencies (10kHz ~ 150kHz)
- ❑ Energy saving cycle-skipping mode for low levels of output power
- ❑ On-chip over-current protection with disable function

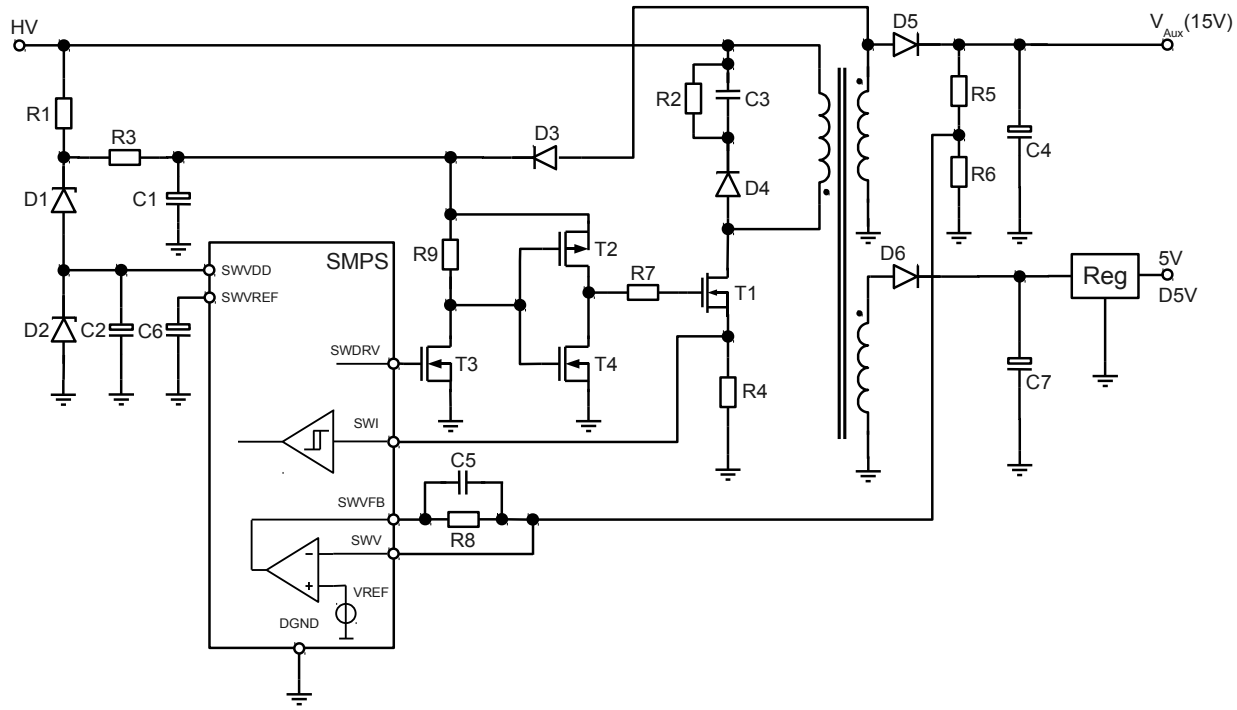


Figure 19: Example Switch Mode Power Supply application schematic (simplified)

The AMG-XB403's Switch Mode Power Supply operates in constant frequency, discontinuous conduction mode (DCM). Depending on the external circuitry it is capable of generating at least one variable output voltage. One of these voltages is directly controlled, the others indirectly. For example in 'Figure 19' an adjustable output voltage V_{AUX} of 15V is generated and controlled by using an external voltage divider (see R5 and R6 in 'Figure 19'). The output voltage V_{AUX} of the switching supply is regulated in such a way that the reference voltage level at the SWVI pin is equal to an internal reference. The SWVFB pin is used for the compensation of the sense amplifier. The primary coil's current control loop is closed by means of the SWI input pin.

The logic level signal SWDRV drives a simple external level shifting circuit consisting of T2 through T4, and R9 which in turn drives the power transistor T1 attached to the primary winding of the switching supply's transformer. The level shifter supply of nominally 15V is fed by the same transformer winding as the output voltage V_{AUX}.

The constant switching frequency of the SMPS can be determined by a low-precision RC oscillator which is part of the SMPS, or by an adjustable, digitally generated reference clock signal which is adjustable via the IO expander (see section 5.2, page 8). At start-up, when no digitally generated clock is available the RC oscillator is used. Typically, the RC oscillator's frequency is significantly lower than the optimum switching frequency to accommodate the oscillator's lack of precision. This means that less than the rated power can be drawn at start-up. To achieve full performance the SMPS must be operated using the well-defined, digitally generated switching frequency.

The SMPS is supplied through the SWVDD pin and must be biased for start-up. Zener diodes are used to provide the supply voltage for the level shifter and SMPS. The SMPS's internal reference voltage at the SWVREF pin needs to be stabilized using a capacitor connected to DGND. The SMPS shares the digital ground pin DGND with the rest of the IC.

As shown in 'Figure 19', a secondary winding of the transformer can be used to derive a secondary, indirectly regulated output voltage helping to improve over-all efficiency.

In the presence of low loads the SMPS enters a cycle skipping mode.

The SMPS is designed to limit the maximum power via the duty cycle of the switching signal. The shunt resistor R4 must be dimensioned in such a way that the transformer's specifications are not exceeded for the given switching frequency, the highest possible voltage across R4 (approximately 1.26V), and the given regulated output voltage. If this maximum load is exceeded for a duration of more than one second (typical) the SMPS's output will be disabled in order to protect the circuit. The secondary switching output is short-circuit protected by monitoring the power-on reset signal. If voltage is present at the digital 5V pin of the AMG-XB403 it will remain below the minimum operating voltage for longer than one second the SMPS's output will be disabled in order to protect the circuit.
The AMG-XB403 requires a power-on cycling to resume normal operation after the occurrence of one of the above error conditions described above.

5.7. Micro Controller Unit (MCU)

The AMG-XB403's MCU features are:

- ❑ 8 bit AVR® compatible RISC core
- ❑ 1kByte Data SRAM
- ❑ 16384 words x 16 bit program memory
- ❑ 8 MHz system clock nominally
- ❑ Watch dog timer
- ❑ Three general purpose interrupt-generating timers
- ❑ One channel ADC function
- ❑ 7 general purpose IOs

Refer to section 5.11, page 56 for a summary of the MCU's instruction set.

5.7.1. Programming and Debug Interface

The programming interface is used to access the MCU's 16 bits wide and 16384 words deep program memory. The programs are executed from Shadow SRAM whose contents are loaded from an EEPROM after power-on reset. For testing purposes the loading process can be stopped, and restarted, by using commands 19 and 20 (see 'Table 14', page 35). The MCU executes the program starting at address 0 of the SRAM.

The unit is also used to debug the MCU core. The programming and debug interface is accessible via the System SPI interface only, the associated bus addresses are undefined for the MCU core.

Before any functionality of the programming, and debug interface is available the IC must be unlocked. This is accomplished by setting the SRAM address to zero and sending the first 1024 words stored in program memory, which serve as a key, to the AMG-XB403.
Unlocking is achieved by setting low byte and high byte to the data registers and issuing the unlock command for each data word (see 'Table 14', page 35). The address will be incremented automatically. The lock/unlock status can be checked through the IO expander as described in section 5.2.2, page 8.

'Table 13' provides a summary of the registers associated with the programming and debug interface. An overview of the available commands is shown in 'Table 14'.

® AVR® is a registered trademark of Atmel Corporation

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	**D0[7:0]**								**SRD0**	data register, lower byte
Read	**D0[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**D1[7:0]**								**SRD1**	data register, higher byte
Read	**D1[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**A0[7:0]**								**SRA0**	address register, lower byte
Read	**PCR[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**A1[7:0]**								**SRA1**	address register, higher byte
Read	**MCU_CLK EN**	**EE_DONE**	**0**	**PCR[12:8]**						
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**0**	**0**	**0**	**CMD[4:0]**					**SRSC**	status and command register
Read	**0**	**0**	**0**	**0**	**0**	**EE_RDY**	**EE_FOUND**	**Status bit**		
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 13: Program and debug interface registers

Code	Description	Purpose
0	Write 16 bit data {D1,D0} to Shadow SRAM at address {A1,A0}	Shadow SRAM programming
1	Load Shadow SRAM data to {D1,D0}	
2	Enable MCU reset	MCU reset
3	Disable MCU reset	
4	Load low byte of MCU stack pointer to D0	MCU debugging
5	Load MCU status register to D0	
6	Load MCU general purpose register to D0	
7	Load MCU RAM to D0	
8	Set break point A[1:0] to {D1,D0}	
9	Erase break point A[1:0]	
10	Continue after break point and disable stepping	
11	Enable stepping	
12	Load high byte of MCU stack pointer to D1	
13	Load program counter value to {A1,A0}	
16	Check data in {D1, D0} against contents of EEPROM	IC unlock
17	start SRAM March C- test	SRAM test
19	start loading EEPROM to Shadow SRAM	EEPROM loading
20	stop loading EEPROM to Shadow SRAM	

Table 14: Command register codes for Program and Debug Interface

5.7.1.1. Program Memory Initialization

After power-on reset the IC probes for an EEPROM by sending a device address word and evaluating its acknowledge status.
If an EEPROM is detected, its contents will be copied to Shadow SRAM and the MCU's reset signal will be released.

If no EEPROM is found reset will not be released, and a “no detect” status will be indicated via the unit's status and command register.

5.7.1.2. Shadow SRAM Reads and Writes

For quick MCU program development the IC's Shadow SRAM can be read and written. Before reading and writing the MCU needs to be reset using command 2 described in 'Table 14'.
In order to execute writes to Shadow SRAM memory the 16 bits of data of the program word are written to the two 8 bit data registers, SRD0 and SRD1. Next, the 14 bit wide address is written to SRA0 and SRA1[5:0]. The actual write is initiated by writing command 0 to the status and command register SRSC.

In order to execute a read from Shadow SRAM memory, the 14 bit address is written to SRA0 and SRA1[5:0]. By writing command 1 to SRSC the SRAM read is executed and the 16 bits of data can be read from SRD0 and SRD1 via the 8 bit on-chip bus after SRSC[0] has been read as 0.

5.7.1.3. EEPROM Reads and Writes

To permanently the store program and configuration data the EEPROM is used. Each byte inside the EEPROM corresponds to one half of a Shadow SRAM program word. The EEPROM address 0 contains the low-byte of Shadow SRAM address 0, EEPROM address 1 contains the high-byte of Shadow SRAM address 0, EEPROM address 2 contains the low-byte of Shadow SRAM address 1, etc.

Serial two-wire communication is performed via port 40 of the IO expander (see 'Table 2b', page 13), and is activated by enabling engineering mode via port 1 of the IO expander. In 'Figure 20' the start and stop conditions are indicated.

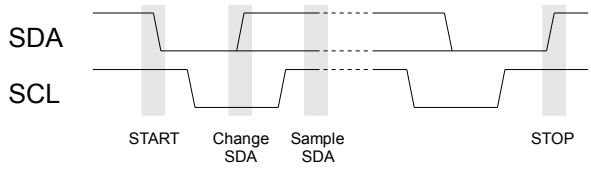


Figure 20: Two-wire communication start stop conditions

State changes of the serial data line SDA when the serial clock SCL is HIGH indicate either the start condition (falling edge of SDA) or the stop condition (rising edge of CLK). Consequently, SDA may only change its state when SCL is LOW during data transfers, master (IO expander) and slave (EEPROM) must sample SDA when SCL is HIGH (see 'Figure 20').

Every start condition is followed by an address cycle as shown in 'Figure 21'. The 7 bit wide address is transmitted with its MSB first and is followed by the read/write flag. Data is transmitted from the IO expander to the EEPROM, if this flag is set to 0 and from slave to master if set to 1. The EEPROM acknowledges an address cycle by pulling SDA to LOW during the clock cycle following the read/write flag. In slave mode the two wire communication interface will only acknowledge the address cycle if the sent address and the address mask are valid. An address cycle can be followed by a data cycle or a stop condition.

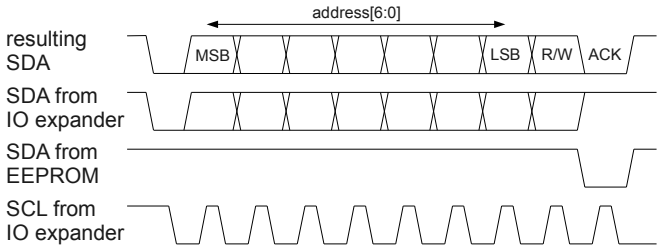


Figure 21: Serial two-wire communication address cycle

As shown in 'Figure 22' a data cycle contains eight data bits and one acknowledge bit. The data is sent to the MSB first. The receiving side acknowledges the received data by pulling SDA low.

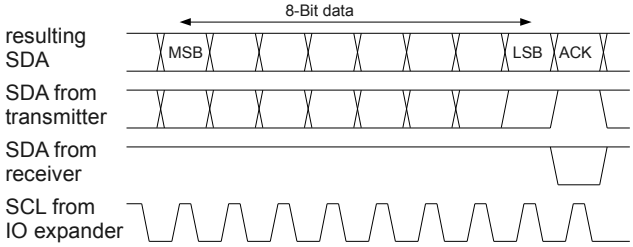


Figure 22: Serial two-wire communication data cycle

When data is sent from IO expander to EEPROM the EEPROM may set STOPREG to send a non-acknowledge after receiving the next byte from the IO expander, thus indicating that it will receive no more data. The EEPROM will enter a passive state after STOPREG has been set until a start condition is received. If data requested by the IO expander is not ready to be sent by the EEPROM the EEPROM will pull the serial clock to LOW until the data is ready. This is the only case in which the EEPROM influences the serial clock.

All serial communication is initiated by a device address word (see 'Figure 23') indicating the EEPROM address and the read/write access mode, writes have the LSB set to 0, reads have the LSB set to 1. For the AMG-XB403 the EEPROM address is always zero. The device address word will be acknowledged if the IC is ready.

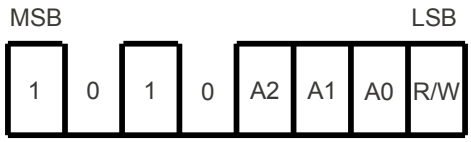


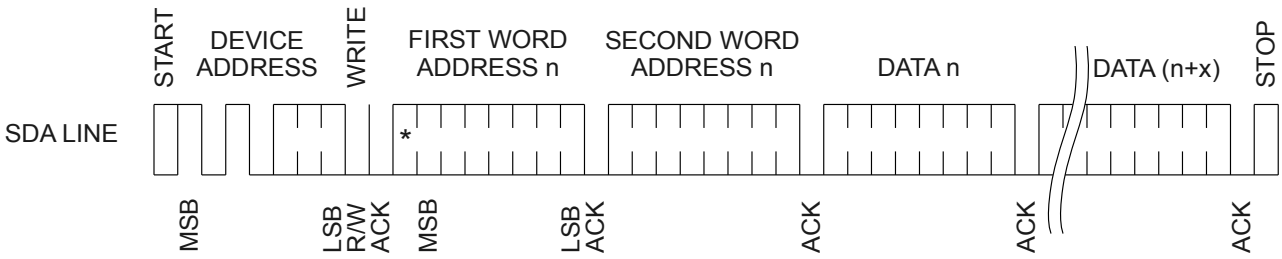
Figure 23: Device address word

5.7.1.4. EEPROM Writes

The 256 kBit EEPROM is capable of up to 64-byte page writes. A page write operation requires the device address word followed by a 16 bits wide start byte address, up to 64 bytes of data, and a stop condition to initiate a self-timed EEPROM write. This is demonstrated in 'Figure 24'. It can be seen that all data and address bytes need to be acknowledged by setting the data line low. Writes start at the given byte address, the address is incremented automatically.

After initiating a page write the EEPROM will not acknowledge requests for typically 5ms. Consequently, waiting for the end of the write operation consists of sending a start condition followed by a device address word, and evaluating the acknowledge non-acknowledge status. The

next write or read command may be sent after receiving an acknowledge signal from the EEPROM.



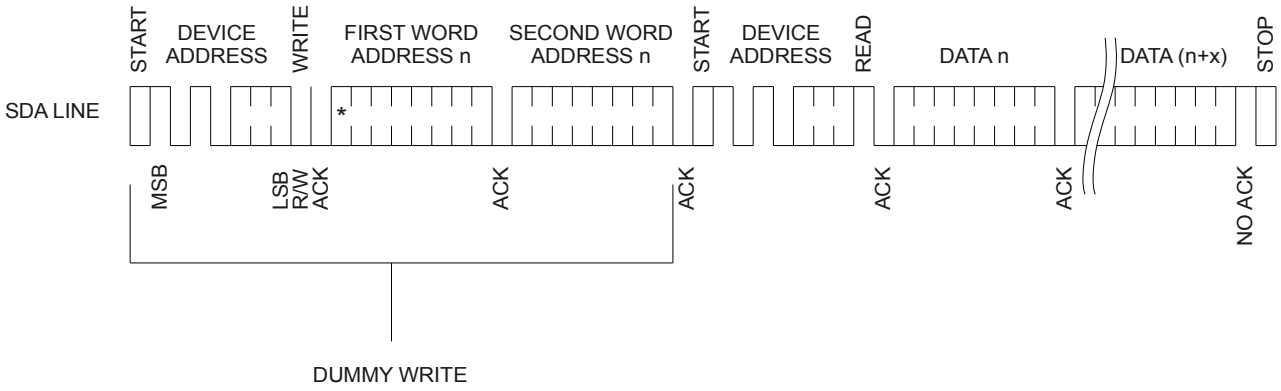
Note: **Don't CARE* bits.

Figure 24: Write procedure

5.7.1.5. EEPROM Reads

As shown in 'Figure 25', the EEPROM reads consist of a dummy write operation, consisting of a “write” device address word followed by a 16 bits wide start byte address, followed by a second start condition and a “read” device address word.

Following this command sequence the EEPROM will return bytes starting at the specified address, and increment the read address for every received acknowledge. The maximum byte address is 32767, after that the address will “roll over” to address 0. The sequential read operation is terminated by sending a stop condition.



Note: **Don't CARE* bits.

Figure 25: Read procedure

The serial bus' timing is shown in 'Figure 26' (below), while the timing values can be found in section 8.3., page 65.

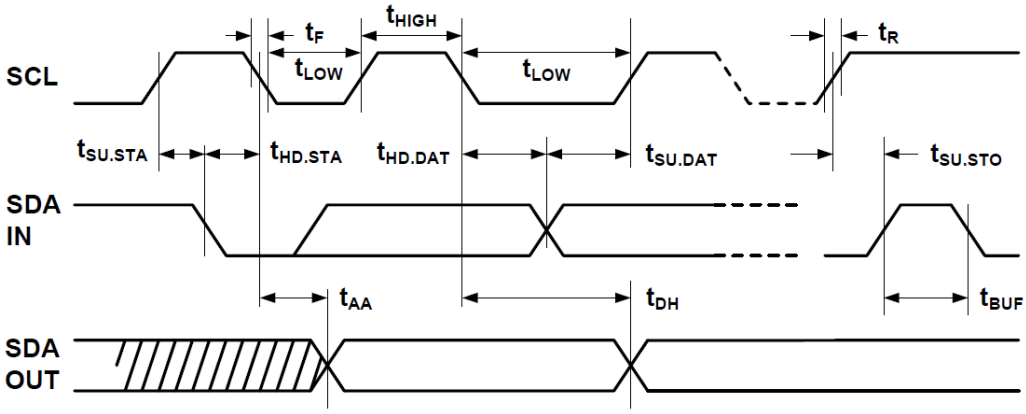


Figure 26: EEPROM serial bus timing

5.7.1.6. March C- SRAM Test

A structural SRAM test is started by using command 17. During memory test data register SRD0 is read back as 1. When the test is finished SRD0 indicates the test result, 0 for pass, and 2 for fail.

5.7.1.7. EEPROM Calibration and User Data Space

Located at the bottom of the EEPROM there is a file system storing calibration data regarding the ADC, RC oscillator and others, along with DMCE and PFC firmware. The calibration data includes the FLL divider, as well as the ADC offset and gain values. After power-on reset the EEPROM's content is copied to the Shadow SRAM, this includes the calibration and user data space.

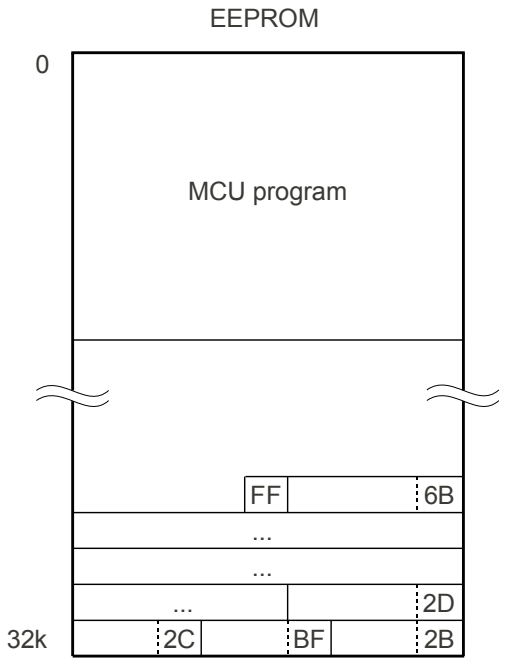


Figure 27: Calibration and user data space

The data space is organized in records (an example is illustrated in 'Figure 27'). Every record consists of a 1 Byte ID number and a data item, which depending on the ID number has a certain length. Records are read ID first and then data. In 'Figure 28' the record format is demonstrated.

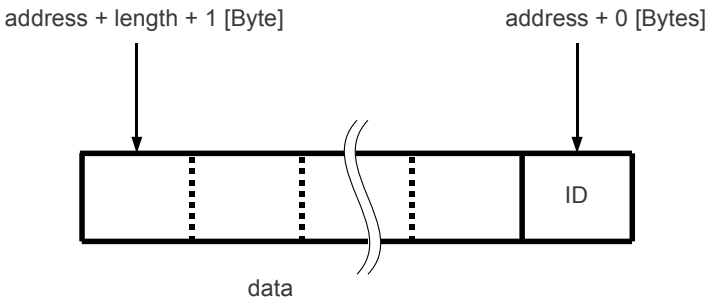


Figure 28: Record format

Once an ID is reserved it cannot be deleted. There are four categories of ID's to differentiate data field lengths, please refer to 'Table 15'.

ID number	Data length
0x01..0x3F	1 Byte
0x40..0x7F	2 Byte
0x80..0xBF	4 Byte
0xC0..0xFE	n Byte

Table 15: Types of ID's

ID numbers 0x01 to 0x1F and 0xC8 to 0xDF can be used for specific application related data, such as constants and parameters for example. ID numbers 0x00, 0xE0 to 0xFE are reserved and ID number 0xFF is used to indicate the end of the data space.

Reading the data space starts at the highest possible address of the Shadow SRAM and continues until ID number 0xFF is found. An ID overview is demonstrated in 'Table 16'.

ID number	Comment
0x00	reserved
0x01..0x1F	available for custom data
0x2B	Band gap reference adjustment value
0x2C..0x35	reserved
0x36..0x3F	ADC offset with buffering Channels: VDC, VAC, 7..0
0x40..0x5F	available for custom data
0x65	ADC VDD calibration value
0x66	ADC temperature calibration value
0x67	FLL calibration parameter 2
0x68	FLL calibration parameter 1
0x69	FLL calibration parameter 0
0x6A	FLL temperature calibration value
0x6B	FLL divider
0x6C..0x75	ADC gain without buffering Channels: VDC, VAC, 7..0
0x76..0x7F	ADC gain with buffering Channels: VDC, VAC, 7..0
0x80..0x9F	available for custom data
0xBF	time stamp (seconds since epoch)
0xC0	DMCE0 program
0xC1	DMCE0 registers
0xC2	reserved
0xC3	reserved
0xC4	PFC program
0xC5	PFC registers
0xC6	PowerStage0 pin order
0xC7	reserved
0xC8..0xDF	available for custom data
0xE0..0xFE	reserved
0xFF	end of data space

Table 16: ID overview

5.7.1.8. Debug Interface

The debug interface allows read access to the MCU's status register, stack pointer, program counter, general purpose registers, and RAM (see commands 4, 5, 6, 7, 12, 13 in 'Table 14'). For accessing general purpose registers a 5 bit address {SRA0[4:0]}, and for RAM data a 10 bit address {SRA1[1:0],SRA0} must be specified.

Four break points can be set and enabled using command 8. When the program counter reaches a break point the MCU’s clock is stopped. This is indicated by MCU_CLKEN going LOW (see 'Table 13'). The continue command 10 is used to re-enable the MCU’s clock.

A stepping mode is also available. In this mode the MCU clock will be stopped after each command. The stepping mode is enabled using command 11. The MCU clock is re-enabled by using the continue command.

5.7.2. General Purpose Input Output Interface (GPIO)

The GPIO block interfaces the IC’s general purpose IO pins, grouped in 8 bit wide IO ports. The AMG-XB403 features 7 general purpose IOs (see 'Figure 29'). The GPIO block relays IO pins to

UART and SPI as shown in 'Table 17'.

IO pin	Signal	Direction	Condition	Comment
PORTB[0]	UART TX	O	UARTEN	UART data output enabled if UARTEN=1, else UART disabled
PORTB[1]	UART RX	I		UART data input enabled if UARTEN=1, else UART disabled
PORTC[4]	SPI SCS	IO	SPIEN	SPI chip select pin enabled if SPIEN=1, else SPI disabled
PORTC[5]	SPI SDO	O		SPI data output pin enabled if SPIEN=1, else SPI disabled
PORTC[6]	SPI SDI	I		SPI data input pin enabled if SPIEN=1, else SPI disabled
PORTC[7]	SPI SCLK	IO		SPI clock pin enabled if SPIEN=1, else SPI disabled

Table 17: XB403 dedicated function GPIO ports

PORTB[0] and PORTB[1] become TX and RX line of the UART if UARTEN is set. The polarity of the TX line can be inverted by setting PORTBO[0] to 1 for non-inverted operation it needs to be set to 0. The polarity of the RX line is set via PORTBO[1] in the same way. Refer to section 5.7.3, page 44 for a detailed description of the UART.

PORTC[7:4] are used for the SPI controller's SCLK, SCS, SDI, and SDO pins. The polarity of the clock can be inverted by setting PORTCO[4] to 1, the standard non-inverting behaviour is achieved by setting PORTCO[4] to 0. Refer to section 5.7.4, page 45 for a detailed description of the SPI.

PORTE[0] is shared with the general purpose ADC channel see section 5.3.1, page 14 for details.

Please note that dedicated GPIO functions may override output pins so these pins are not controlled by the output buffer register. When reading a port, the actual input is read instead of the buffer content, even if the port is configured to be an output. This needs to be considered when conducting bit-manipulation on GPIO ports with special functions enabled. For example, PORTBO[0] determines TX polarity when UARTEN is set. When reading back the port, PORTB[0] keeps the actual TX line instead of the TX polarity setting. Therefore, these bits must always be written as intended, not as read.

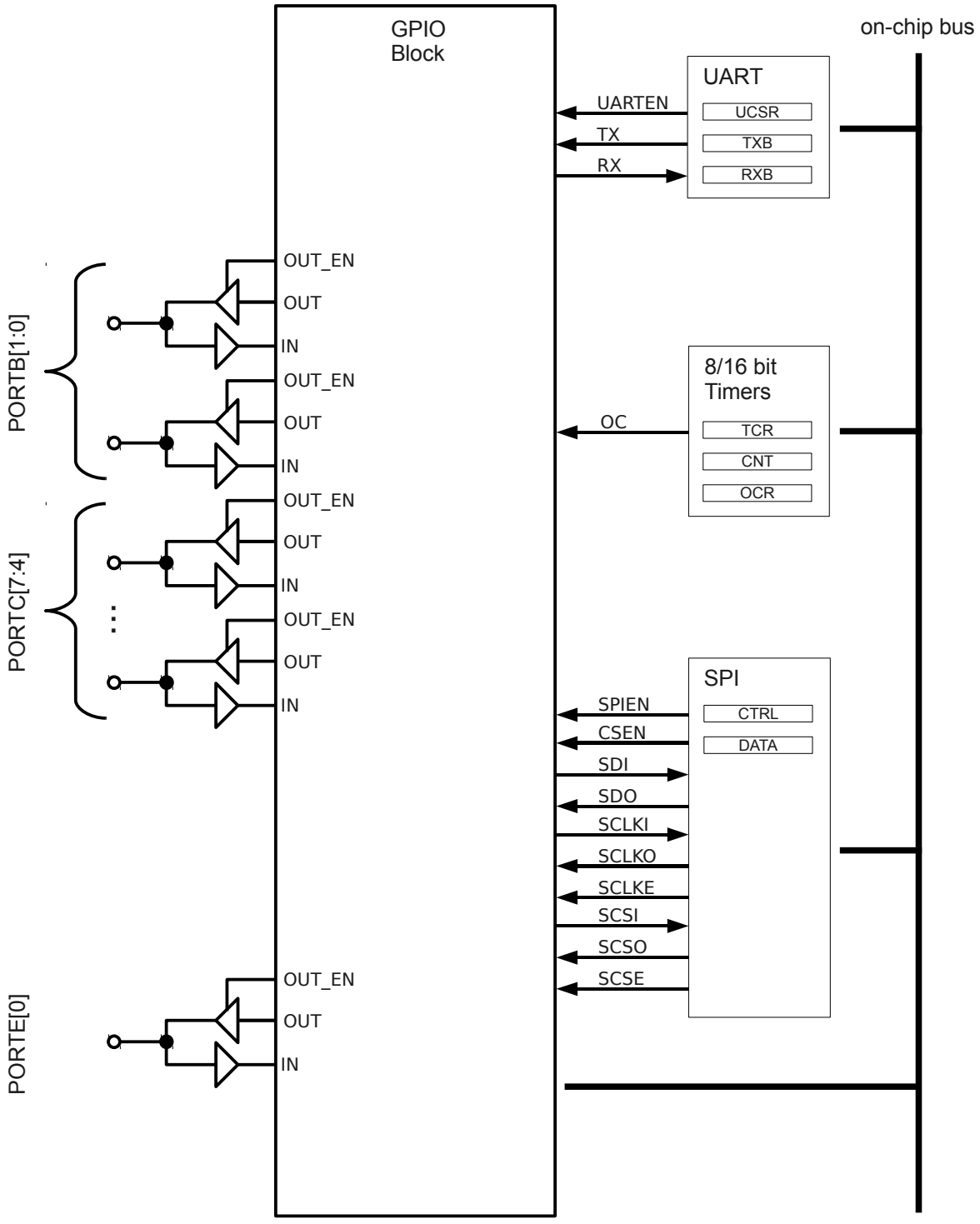


Figure 29: XB403 GPIO overview

Bit	7	6	5	4	3	2	1	0	Port	Comment
Write	PORTBO[7]	OC2	OC1	OC0	PORTBO[3:2]		UARTRXPOL	UARTTXPOL	B	Port B data register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTBI[1:0]									
Write	PORTCO[7:5]			SCKLIPOL	PORTCO[3]	IC2POL	IC1POL	IC0POL	C	Port C data register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTCI[7:4]									
Write	PORTBE[1:0]								B	Port B output enable register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTBE[1:0]									
Write	PORTCE[7:4]								C	Port C output enable register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTCE[7:4]									
Write	PORTIE[7:0]								-	Interrupt enable register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTIE[7:0]									
Write	PORTIFR[7:0]								-	Interrupt flag register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTIFR[7:0]									
Write	PORTEO[0:0]								E	Port E data register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTEI[0:0]									
Write	PORTEE[0:0]									Port E output enable register
Initial value	0	0	0	0	0	0	0	0		
Read	PORTEE[0:0]									

Table 18: XB403 GPIO registers

5.7.3. Universal Asynchronous Receiver Transmitter Interface (UART)

The UART's IO pins are mapped to GPIO pins if enabled (see 'Figure 29'). The baud rate is adjustable. All data is sent and received in blocks of a start bit, eight data bits, and a stop bit (see 'Figure 30').

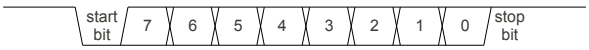


Figure 30: UART data transfer

Data is sent by writing a byte to the transmit buffer register TXB, the most recent received byte can be read from the receive buffer RXB, see 'Table 30', page 55.

The UART is configured by writing to the control and status register UCSR. As shown in 'Table 20' the interface is enabled by setting UART_EN to 1, the baud rate is adjusted by setting the lower three bits as indicated in 'Table 19'. If the ready flag RDY is set to 1 the interface has sent all pending data and is ready for another transmission. If the stop bit of an incoming transmission is not correct a framing error is indicated by the framing error flag FERR being set to 1. An overflow error will be indicated by the flag OERR. It is set to 1 if a byte is received before the previous byte has been read from RXB.

The RX and TX polarities can be set by writing UARTRXPOL respectively UARTTXPOL (refer to

'Table 18', page 44). If UARTRXPOL is set to 1, the RX line is interpreted active low, otherwise it is interpreted active high. If UARTTXPOL is set to 1, data is sent out active low, otherwise it is sent out active high. Active high is the default setting for both lines.

The UART has got two interrupt signals that can be enabled as shown in section 5.7.7, page 49. An RX interrupt is generated when a new byte is received and stored in RXB, the interrupt flag is reset by reading the RX buffer. A TX interrupt is generated when the byte in TXB has been sent, the interrupt flag is reset by writing to the TX buffer.

Code	BAUDRATE[2:0]	Baud rate
0	010	9600
1	011	19200
2	100	38400
3	101	57600
4	110	115200

Table 19: UART baud rates

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	**UART_EN**	**-**	**-**	**-**	**-**	**BAUDRATE[2:0]**			**UCSR**	Status and control register
Read	**UART_EN**	**RDY**	**FERR**	**OERR**	**0**	**BAUDRATE[2:0]**				
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**RXB[7:0]**								**RXB**	Receive buffer
Read	**RXB[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**TXB[7:0]**								**TXB**	Transmit buffer
Read	**TXB[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 20: UART registers

5.7.4. Serial Parallel Interface (SPI)

The SPI's IO pins are mapped to GPIO pins (see 'Figure 29', page 43). The SPI can be run as a master or a slave. In master mode the MCU can communicate with one or more SPI slaves. In slave mode a SPI master can control the SPI. There is a control and a data register which are shown in detail in 'Table 21'.

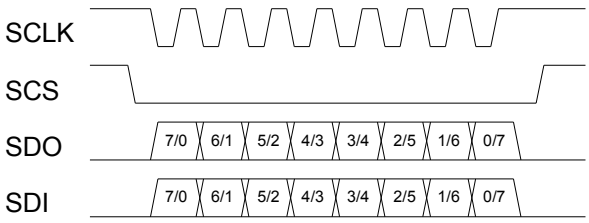


Figure 31: SPI timing

SPI chip select SCS and SPI clock SCLK are always driven by the master, they frame and time the transmission. The data is transferred in blocks of up to eight bits, data is always sent out synchronously via SDO and received via SDI using a single 8 bit shift register controlled by the serial clock. Both 'Figures 31' and ' 32' illustrate a 8 bit SPI transmission. By default SDO and SDI are sampled on the rising edge of SCLK, sampling will take place on the falling edge of SCLK if the clock's polarity is inverted in the GPIO register (see section 5.7.2, page 41).

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	DATA[7:0]								SPIDAT	Data register
Read	DATA[7:0]									
Initial value	0	0	0	0	0	0	0	0		
Write	-	MODE[1:0]		SPIEN	MSBLSB	CSEN	-	-	SPICTRL	Control register
Read	-	MODE[1:0]		SPIEN	MSBLSB	CSEN	WCOL	TRF		
Initial value	0	0	0	0	0	0	-	-		

Table 21: SPI registers

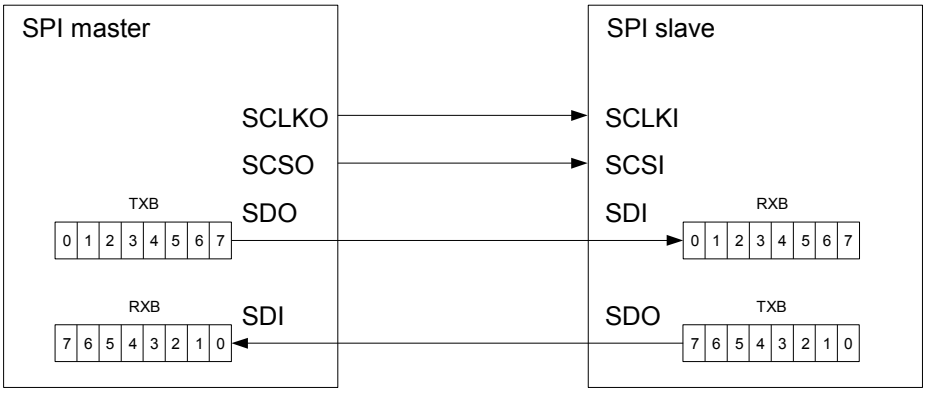


Figure 32: SPI communication principle

To enable the SPI the flag SPIEN must be set to 1. With the MSBLSB flag set to 1, the MSB will be transferred first otherwise the LSB is transferred first. The flag CSEN controls the usage of the chip select line, if set to 1 chip select is used, otherwise chip select will be ignored.

WCOL and TRF are read-only flags for write collision WCOL and readiness for next transmission TRF. In master mode a collision occurs if new data is written to SPIDAT while the previous transmission has not been finished. The SPI's mode controls master/slave operation and the serial clock speed in master mode. It is selected by setting MODE[1:0] as shown in 'Table 22'.

Mode	CTRL[6]	CTRL[5]	Clock speed
Master	0	0	2 MHz
Master	0	1	500 kHz
Master	1	0	125 kHz
Slave	1	1	-

Table 22: SPI mode selection

The SPI has got a single interrupt flag which can be enabled through the interrupt control register (see section 5.7.7, page 49). The interrupt is raised after each completed transmission. The interrupt flag is cleared by reading from or writing to SPIDAT.

5.7.5. General Purpose Timers

The AMG-XB403 has got two 8 bit timers and one 16 bit timer. All timers are controlled by the MCU via the on-chip bus. Each timer has one dedicated interrupt generated by match and input capture timer events. Timer interrupts can be enabled independently. Interrupt flags are cleared automatically upon timer interrupt execution or IC reset. 'Table 24' lists all timer-related registers, each timer can be configured to use an individual system clock pre-scaler ratio.

The timers have a 8 or 16 bit wide counter register CNT, which can be read from and written to, and an output compare register OCR, containing the compare value for the timer match function, which also can be read from and written to. Each timer's control register contains the individual timer's pre-scaler setting in its least significant 3 bits (see 'Table 23'). The timer will be disabled if the pre-scaler is set to 0.

PRESCALER[2:0]	Timer reference clock
0	timer off
1	MCU clock frequency
2	MCU clock frequency divided by 2
3	MCU clock frequency divided by 4
4	MCU clock frequency divided by 16
5	MCU clock frequency divided by 64
6	MCU clock frequency divided by 256
7	MCU clock frequency divided by 512

Table 23: Clock modes for 8 bit and 16 bit timers

If the timer is enabled the timer's counter register will be incremented according to the pre-scaler setting. If the counter overflows CNT changes from 65535 to 0 (16 bit timer) or 255 to 0 (8 bit timers).

Each timer has two principle modes, match mode and input capture mode. Match mode is always active. In addition, input capture mode can be enabled if input capture enable ICEN is set to 1.

In match mode, (see 'Figure 33') the interrupt flag is set if CNT equals OCR, and interrupt enables INTEN in TCR is set to 1. If reset on match ROM in TCR is set to 1 CNT will be reset to 0. An interrupt on match is only generated if input capture enable ICEN is set to 0.

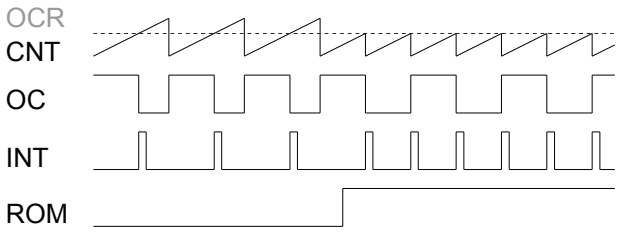


Figure 33: Timer match mode (assuming immediate interrupt handling)

By setting OCEN and the according GPIO PORTB output enable to 1 the output compare output OC is enabled (see 'Table 18, page 44). OC is set to 0 on reset and will be toggled on match if reset on match is enabled. By writing to TCR[4] OC can be pre-set. If reset on match is disabled OC is set to 0 on match and set to 1 on counter overflow.

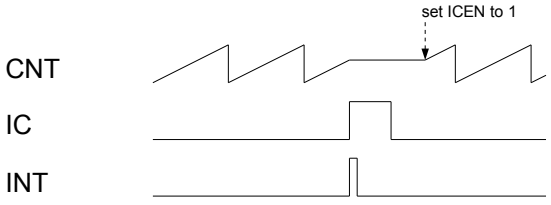


Figure 34: Timer input capture mode (assuming immediate interrupt handling)

In input capture mode (ICEN=1, see 'Figure 34') an interrupt is generated if a rising edge at the input capture input pin IC is detected and interrupt enable INTEN in TCR is set to 1. CNT will be stopped and can be read via the bus. By writing 1 to TCR[6] CNT will continue to be incremented.

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	**TMRIE**	**ICEN**	**OCEN**	**OCIR**	**RESOM**	**PRESCALER[2:0]**			**TCR8**	8 bit Timer control register
Read	**TMRIE**	**ICEN**	**OCEN**	**OCIR**	**RESOM**	**PRESCALER[2:0]**				
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**CNT[7:0]**								**TCNT8**	8 bit Timer counter register
Read	**CNT[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**OCR[7:0]**								**OCR8**	8 bit Timer output compare register
Read	**OCR[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**TMRIE**	**ICEN**	**OCEN**	**OCIR**	**RESOM**	**PRESCALER[2:0]**			**TCR16**	16 bit Timer control register
Read	**TMRIE**	**ICEN**	**OCEN**	**OCIR**	**RESOM**	**PRESCALER[2:0]**				
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**TCNTL[7:0]**								**TCNTL16**	16 bit Timer counter register; low byte
Read	**TCNTL[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**TCNTH[15:8]**								**TCNTH16**	16 bit Timer counter register; high byte
Read	**TCNTH[15:8]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**OCRL[7:0]**								**OCRL16**	16 bit Timer output compare register; lower byte
Read	**OCRL[7:0]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	**OCRH[15:8]**								**OCRH16**	16 bit Timer output compare register; higher byte
Read	**OCRH[15:8]**									
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 24: General purpose timer registers, registers for second 8bit timer are identical to first and not shown

5.7.6. Watchdog Timer

The dedicated watchdog timer can be used to restart program execution starting at the reset vector address (zero) if the MCU fails to update it periodically. The watchdog is controlled by the watchdog timer control register WDTCR, see 'Table 25'.

The watchdog is enabled by setting the watchdog enable flag WDE to 1. The time-out value is set using the watchdog pre-scaler WDP, the time-out value is calculated as follows: $t_{TO} = \frac{2^{(17+WDP)}}{f_{MCU}}$.

Following a regular power-on reset the watchdog time-out flag WDTO is set to 1. If the reset vector is executed following a watchdog time-out WDTO will be set to 0, thus indicating a watchdog time-out event.

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	-	-	-	-	**WDE**	**WDP[2:0]**			**WDTCR**	Watchdog timer control register
Read	-	-	-	**WDTO**	**WDE**	**WDP[2:0]**				
Initial value	-	-	-	**1**	**0**	**0**	**0**	**0**		

Table 25: Watchdog timer control register

5.7.7. Interrupt Controller

The AMG-XB403 supports ten interrupt sources. Interrupts are enabled by setting the global interrupt enable flag GIE and the desired per-interrupt interrupt enable flags to 1. If more than one interrupt signal is active the interrupt with the lowest interrupt vector will be executed first. The GIE flag is disabled during interrupt vector execution and re-enabled when returning from an interrupt. It is possible to manually re-enable the GIE flag during interrupt execution, and thus employ nested interrupts.

As shown in 'Table 26', all interrupts are indicated by interrupt flags in registers INTCONL and INTCONH. The interrupt enable flags TWIIE (two wire interface), TXCIE (UART transmission complete), RXCIE (UART reception complete), and SPIIE (MCU SPI ready) are contained in INTCONH. All other interrupt enable flags are contained in the control registers of the individual units as summarized in 'Table 27'.

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	-	-	-	-	-	-	-	-	**INTCONL**	Interrupt controller register; low byte
Read	**IF_ADC**	**IF_TXC**	**IF_RXC**	**IF_TWI**	**EXTINT**	**IF_TMR2**	**IF_TMR1**	**IF_TMR0**		
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
Write	-	-	-	**PFCIE**	**SPIIE**	**RXCIE**	**TXCIE**	**TWIIE**	**INTCONH**	Interrupt controller register; high byte
Read	**IF_PFC**	**IF_SPI**	**1**	**PFCIE**	**SPIIE**	**RXCIE**	**TXCIE**	**TWIIE**		
Initial value	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**		

Table 26: Interrupt controller registers

Interrupt priority	Source	Auto clear	Interrupt events	Interrupt enable flag	Interrupt flag
1	TMR0	yes	Timer match Input capture	TCR8[7]	INTCONL[0]
2	TMR1	yes	Timer match Input capture	TCR8[7]	INTCONL[1]
3	TMR2	yes	Timer match Input capture	TCR16[7]	INTCONL[2]
4	GPIO	no	External interrupt	PORTIE	INTCONL[3], PORTIFR
5	UART RXC	no	Data reception complete	INTCONH[2]	INTCONL[5]
6	UART TXC	no	Data transmission complete	INTCONH[1]	INTCONL[6]
7	TWI	no	Data reception/ transmission complete	INTCONH[0]	INTCONL[4]
8	ADC	no	Sampling finished	ADCCONF[6]	INTCONL[7]
9	SPI	no	Data transmission complete	INTCONH[3]	INTCONH[6]
10	PFC	no	DMCE/PFC exception	INTCONH[4]	INTCONH[7]

Table 27: Interrupt sources

5.7.8. Status Register

The status register SREG contains the global interrupt enable flag GIE and the MCU's status bits. These bits are the transfer flag (TO), the half carry flag (HO), the signed flag (SO), the twos complement overflow flag (VO), the negative flag (NO), the zero flag (ZO), and the carry flag (CO) as summarized in 'Table 28'.

Bit	7	6	5	4	3	2	1	0	Register name	Comment
Write	**GIE**	**TO**	**HO**	**SO**	**VO**	**NO**	**ZO**	**CO**	**SREG**	MCU status register
Read	**GIE**	**TO**	**HO**	**SO**	**VO**	**NO**	**ZO**	**CO**		
Initial value	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		

Table 28: Status register

5.8. System Clock

The AMG-XB403 system clock's features:

- ❑ supports two different system clock modes
- ❑ nominal 64MHz system clock
- ❑ internally generated clock can be put out to feed external circuitry

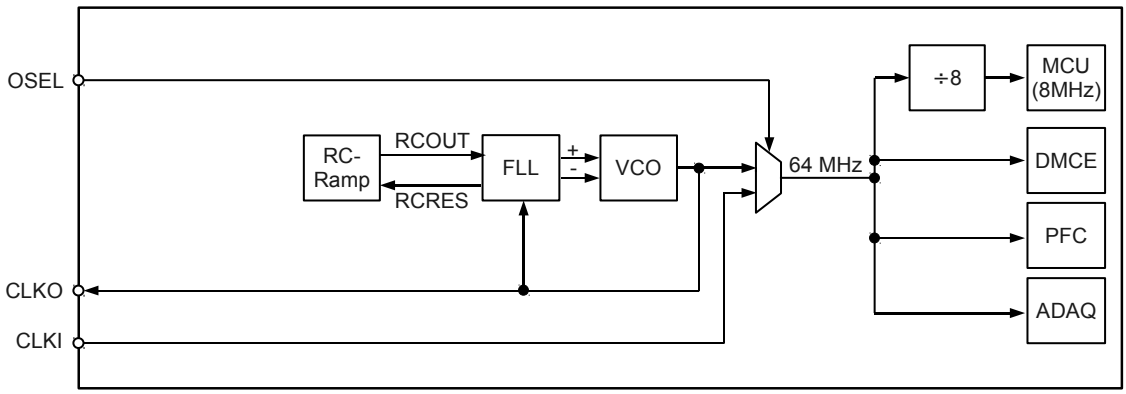


Figure 35: Oscillator and FLL block schematic

The clock mode is set by connecting the CLKSEL and OSEL pin to D5V or DGND as shown in 'Table 29':

- On-chip RC-ramp-based, Frequency Locked Loop (FLL) -controlled, VCO-generated on-chip clock
- Clock input from external clock input pin

OSEL	CLKIO	Description
0	Input	External clock input signal via CLKIO pin
1	Output/Off	Internal RC-ramp based clock generation

Table 29: Clock source configuration

The recommended clock source of the AMG-XB403 uses an integrated RC-ramp of nominally 60 µs for adjusting the on-chip VCO. As shown in 'Figure 35' this ramp signal serves as the time standard for the FLL block which generates the system clock. The FLL block uses a 13 bit wide divider ratio specifying the number of VCO clock cycles per RC-ramp. The MCU clock frequency is generated by dividing the VCO's clock frequency by eight, consequently all system clocks are in-phase signals.

Initially, the FLL divider ratio is set to 2048, resulting in an approximately 48 MHz clock output for a typical RC-ramp at room temperature. The FLL's divider ratio is trimmed via the IO port expander (see section 5.2, page 8) to achieve the target clock frequency of 64MHz.

A clock rate error of less than ±2% over the operating temperature range can be achieved by

providing a temperature-dependent FLL divider ratio based on a polynomial function of chip temperature. The initial divider value and the temperature coefficient are measured and stored in the EEPROM during final test. They can be used by the MCU to perform temperature compensation in combination with the ADC value of the temperature channel.

The AMG-XB403's system clock signal can also be input via the CLKIO pin when configured appropriately.

In order to prevent external noise from interfering with the clock selection setting, the CLKLOCK flag can be set via the IO expander. When set, the flag will preserve the selection preset at the time of setting CLKLOCK, see section 5.2., page 8 for details.

When an internal clock source is selected the CLKIO pin acts as a clock output, allowing for system clock monitoring. CLKIO output may be disabled by setting CLKOFF within the IO expander, see section 5.2., page 8 for details.

5.9. Power-On and System Reset

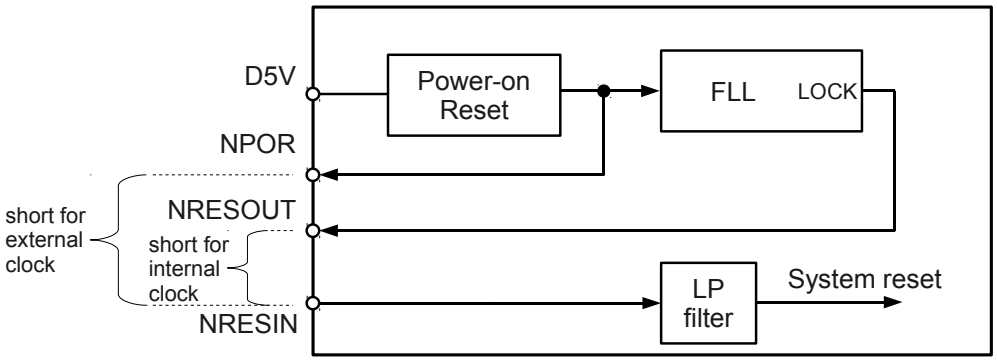


Figure 36: AMG-XB403 reset scheme block schematic

The AMG-XB403 has got a dedicated active-LOW reset signal input pin named NRESIN. The reset input is low-pass filtered for improved noise rejection. Any reset impulse shorter than 2µs (typical) will be discarded.

As shown in 'Figure 36', system reset must be handled differently for externally and internally generated clock signals. For internally generated clocks the active-LOW FLL lock signal at the output pin NRESOUT must be connected to NRESIN. For externally generated clocks the active-LOW output pin NPOR of the power-on reset unit must be connected to NRESIN. An external reset circuit may also feed NRESIN directly but it must ensure that the operating conditions are fulfilled as indicated below.

The power-on reset unit monitors the supply voltage D5V and keeps the FLL unit reset while the supply voltage is too low for supplying the AMG-XB403. When D5V rises above the safe threshold voltage V_{POR} the FLL reset signal POR will be released with power-on delay of t_{POR}. The FLL will then start to adjust the AMG-XB403's VCO and indicated a frequency lock by setting its lock signal to LOW, the lock signal can be observed at pin NRESOUT. The start-up to lock time of the FLL is defined as t_{LOCK}.

If D5V drops below the brown-out voltage V_{BO} the power-on reset unit will trigger a reset of the FLL unit which will in turn reset the digital core of the AMG-XB403. The AMG-XB403 will resume operation with the power-on reset cycle described above as soon as D5V rises above V_{POR}. Power-on reset and brown-out reset scenarios are detailed in 'Figure 37'.

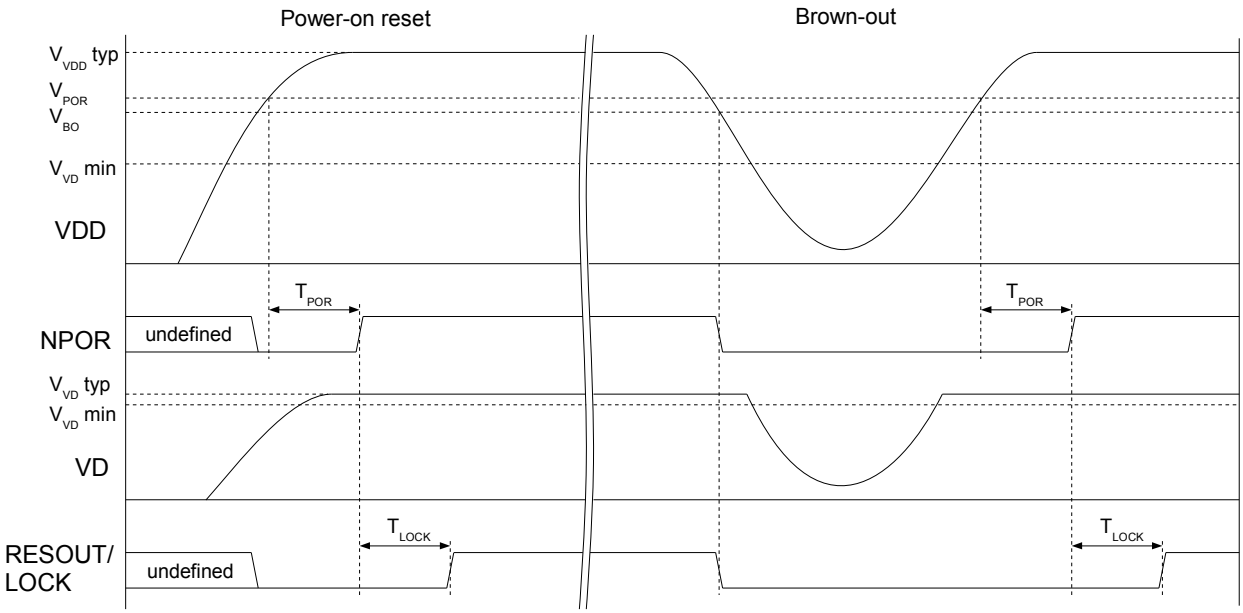


Figure 37: Power-on reset and brown-out reset timing

5.10. On-Chip Bus Register Summary

INDIRECT 10 Bit ADDRESS	DIRECT 6 Bit ADDRESS	REGISTER NAME	COMMENTS	page
0x0 ~ 0x1F		R0~R31	General purpose registers	
0x21	0x1	MDAT0	M0 data register 0	
0x22	0x2	MDAT1	M0 data register 1	
0x23	0x3	MDAT2	M0 data register 2	
0x24	0x4	MDAT3	M0 data register 3	
0x25	0x5	MADR	M0 address register	
0x26	0x6	M0SC	M0 status and configuration register	
0x27	0x7	PFCD	PFC data	
0x28	0x8	PFCSC	PFC status and command	
0x29	0x9	MCPUSEL	DMCE select register	
0x2A	0xA	ADCCONF	ADC configuration	
0x2B	0xB	ADCIRES	ADC interrupt reset (write only)	
		ADCD	ADC data (read only)	
0x2C	0xC	ADCREQ	ADC sample request (write only)	
		ADCINT	ADC interrupts (read only)	
0x2D	0xD	PORTEI PORTEO	GPIO port E data input/output	
0x2E	0xE	PORTEE	GPIO port E output enable	
0x2F	0xF	IOEXPA	IO expander address	
0x30	0x10	SRD0	Shadow RAM low byte of data	
0x31	0x11	SRD1	Shadow RAM high byte of data	
0x32	0x12	SRA0	Shadow RAM low byte of address	
0x33	0x13	SRA1	Shadow RAM high byte of address	
0x34	0x14	SRSC	Shadow RAM status and command	
0x35	0x15	reserved reserved	reserved	
0x36	0x16	PORTBI PORTBO	GPIO port B data input/output	
0x37	0x17	PORTCI PORTCO	GPIO port C data input/output	
0x38	0x18	reserved reserved	reserved	
0x39	0x19	reserved	reserved	
0x3A	0x1A	PORTBE	GPIO port B output enable	
0x3B	0x1B	PORTCE	GPIO port C output enable	
0x3C	0x1C	reserved	reserved	
0x3D	0x1D	PORTIE	GPIO interrupt enable register	
0x3E	0x1E	PORTIFR	GPIO interrupt flag indicator and reset register	
0x3F	0x1F	IOEXPD	IO expander data	
0x30	0x20			
0x41	0x21	WDTCR	watchdog timer control register	
0x42	0x22	TCR0	timer 0 control register	
0x43	0x23	TCNT0	timer 0 counter value register	
0x44	0x24	OCR0	timer 0 output compare value register	
0x45	0x25	TCR1	timer 1 control register	
0x46	0x26	TCNT1	timer 1 counter value register	
0x47	0x27	OCR1	timer 1 output compare value register	
0x48	0x28	TCR2	timer 2 control register	
0x49	0x29	TCNTL2	timer 2 counter value register, low byte	
0x4A	0x2A	TCNTH2	timer 2 counter value register, high byte	
0x4B	0x2B	OCRL2	timer 2 output compare value register, low byte	
0x4C	0x2C	OCRH2	timer 2 output compare value register, high byte	
0x50	0x30	UCSR	UART status and configuration register	
0x51	0x31	RXB	UART receive register	
0x52	0x32	TXB	UART transmit register	
0x53	0x33	INTCONL	interrupt controller low configuration register	
0x54	0x34	INTCONH	interrupt controller high configuration register	
0x55	0x35	TWISC	two wire interface status and configuration	
0x56	0x36	TWIADR	two wire interface address	
0x57	0x37	TWIDAT	two wire interface data	
0x58	0x38	SPICTRL	SPI control register	
0x59	0x39	SPIDAT	SPI data register	
0x5D	0x3D	SPL	stack pointer low byte	
0x5E	0x3E	SPH	stack pointer high byte	
0x5F	0x3F	SREG	MCU status register	

Table 30: On-Chip bus Register summary

5.11. MCU Instruction Set

Flag	Description
C	Carry flag
Z	Result is zero
N	Result is negative
V	Two's complement overflow indicator
S	N+V, used for signed tests
H	Half carry flag
T	Transfer bit used (BLD and BST instructions)
I	Global interrupt Enable flag

Table 31: MCU flags

Operand	Description
Rd	Destination register in register file
Rr	Source register in register file
R	Result after instruction executed
K	Constant data
k	Constant address
b	Bit in register file or IO register
s	bit in status register
X,Y,Z	Indirect address register (X={R27:R26}, Y={R29:R28}, Z={R31:R30})
A	I/O location address
q	displacement for indirect addressing

Table 32: MCU operands

Mnemonic	Operands	Description	Operation		Flags	W	N	Binary Opcode
ARITHMETIC AND LOGIC INSTRUCTIONS								
ADD	Rd, Rr	Add two registers	Rd ← Rd + Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0000 11rd dddd rrrr
ADC	Rd, Rr	Add with carry two registers	Rd ← Rd + Rr + C	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0001 11rd dddd rrrr
ADIW	Rdl, K	Add immediate to word	Rdh:Rdl ← Rdh:Rdl + K	d = 24, 26, 28, 30; 0 ≤ K ≤ 63	Z,C,N,V,S	1	1	1001 0110 KKdd KKKK
SUB	Rd, Rr	Subtract two registers	Rd ← Rd - Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0001 10rd dddd rrrr
SUBI	Rd, K	Subtract constant from register	Rd ← Rd – K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,C,N,V,H,S	1	1	0101 KKKK dddd KKKK
SBC	Rd, Rr	Subtract with carry two registers	Rd ← Rd – Rr - C	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0000 10rd dddd rrrr
SBCI	Rd, K	Subtract constant from register with carry	Rd ← Rd – K - C	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,C,N,V,H,S	1	1	0100 KKKK dddd KKKK
SBIW	Rdl, K	Subtract immediate from word	Rdh:Rdl ← Rdh:Rdl + K	d = 24, 26, 28, 30; 0 ≤ K ≤ 63	Z,C,N,V,S	1	1	1001 0111 KKdd KKKK
AND	Rd, Rr	Logical AND registers	Rd ← Rd & Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,N,V,S	1	1	0010 00rd dddd rrrr
ANDI	Rd, K	Logical AND register and constant	Rd ← Rd & K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,N,V,S	1	1	0111 KKKK dddd KKKK
OR	Rd, Rr	Logical OR registers	Rd ← Rd | Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,N,V,S	1	1	0010 10rd dddd rrrr
ORI	Rd, K	Logical OR register and constant	Rd ← Rd | K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,N,V,S	1	1	0110 KKKK dddd KKKK
EOR	Rd, Rr	Exclusive OR registers	Rd ← Rd ^ Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,N,V,S	1	1	0010 01rd dddd rrrr
COM	Rd	One's complement	Rd ← 8'hFF - Rd	0 ≤ d ≤ 31	Z,C,N,V,S	1	1	1001 010d dddd 0000
NEG	Rd	Two's complement	Rd ← 8'h00 - Rd	0 ≤ d ≤ 31	Z,C,N,V,H,S	1	1	1001 010d dddd 0001
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd I K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,N,V,S	1	1	0110 KKKK dddd KKKK
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd & (8'hFF - K)	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,N,V,S	1	1	0110 !K!K!K!K dddd !K!K!K!K
INC	Rd	Increment	Rd ← Rd + 1	0 ≤ d ≤ 31	Z,N,V,S	1	1	1001 010d dddd 0011
DEC	Rd	Decrement	Rd ← Rd - 1	0 ≤ d ≤ 31	Z,N,V,S	1	1	1001 010d dddd 1010
TST	Rd	Test for zero or minus	Rd ← Rd & Rd	0 ≤ d ≤ 31	Z,N,V,S	1	1	0010 00dd dddd dddd
CLR	Rd	Clear register	Rd ← Rd ^ Rd	0 ≤ d ≤ 31	Z,N,V,S	1	1	0010 01dd dddd dddd
SER	Rd	Set register	Rd ← 8'hFF	16 ≤ d ≤ 31	None	1	1	1110 1111 dddd 1111
MUL	Rd, Rr	Multiply unsigned	R1:R0 ← Rd * Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C	1	1	1001 11rd dddd rrrr
MULS	Rd, Rr	Multiply signed	R1:R0 ← Rd * Rr	16 ≤ d ≤ 31; 16 ≤ r ≤ 31	Z,C	1	1	0000 0010 dddd rrrr
MULSU	Rd, Rr	Multiply signed with unsigned	R1:R0 ← Rd * Rr	16 ≤ d ≤ 23; 16 ≤ r ≤ 23	Z,C	1	1	0000 0011 0ddd 0rrr
FMUL	Rd, Rr	Fractional multiply unsigned	R1:R0 ← (Rd * Rr) << 1	16 ≤ d ≤ 23; 16 ≤ r ≤ 23	Z,C	1	1	0000 0011 0ddd 1rrr
FMULS	Rd, Rr	Fractional multiply signed	R1:R0 ← (Rd * Rr) << 1	16 ≤ d ≤ 23; 16 ≤ r ≤ 23	Z,C	1	1	0000 0011 1ddd 0rrr

Mnemonic	Operands	Description	Operation		Flags	W	N	Binary Opcode
FMULSU	Rd, Rr	Fractional signed by unsigned multiply	R1:R0 ← (Rd * Rr) << 1	16 ≤ d ≤ 23; 16 ≤ r ≤ 23	Z,C	1	1	0000 0011 1ddd 1rrr
BRANCH INSTRUCTIONS								
RJMP	k	Relative jump	PC ← PC + k + 1	-2K ≤ k < 2K	None	1	1	1100 kkkk kkkk kkkk
IJMP	k	Indirect jump to (Z)	PC ← Z[12:0]		None	1	1	1001 0100 0000 1001
JMP	k	Direct jump	PC ← k	0 ≤ k < 8K	None	2	2	1001 010k kkkk 110k kkkk kkkk kkkk kkkk
RCALL	k	Relative subroutine call	PC ← PC + k + 1	-2K ≤ k < 2K	None	1	2	1101 kkkk kkkk kkkk
ICALL	k	Indirect call to (Z)	PC ← Z[12:0]		None	1	2	1001 0101 0000 1001
CALL	k	Direct subroutine call	PC ← k	0 ≤ k < 2K	None	2	2	1001 010k kkkk 111k kkkk kkkk kkkk kkkk
RET		Subroutine return	PC ← STACK		None	1	2	1001 0101 0000 1000
RETI		Interrupt return	PC ← STACK		None	1	2	1001 0101 0001 1000
CPSE	Rd, Rr	Compare, skip if equal	If (Rd = Rr) PC ← PC + (2 or 3)	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	None	1	1/2	0001 00rd dddd rrrr
CP	Rd, Rr	Compare	Rd - Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0001 01rd dddd rrrr
CPC	Rd, Rr	Compare with carry	Rd - Rr - C	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	Z,C,N,V,H,S	1	1	0000 01rd dddd rrrr
CPI	Rd, K	Compare register with immediate	Rd - K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	Z,N,V,C,H,S	1	1	0011 KKKK dddd KKKK
SBRC	Rr, b	Skip if bit in register cleared	If (Rr(b) = 0) PC ← PC + (2 or 3)	0 ≤ r ≤ 31; 0 ≤ b ≤ 7	None	1	1/2	1111 110r rrrr 0bbb
SBRS	Rr, b	Skip if bit in register is set	If (Rr[b] = 1) PC ← PC + (2 or 3)	0 ≤ r ≤ 31; 0 ≤ b ≤ 7	None	1	1/2	1111 111r rrrr 0bbb
SBIC	A, b	Skip if bit in I/O register cleared	If (I/O(A,b) = 0) PC ← PC + (2 or 3)	0 ≤ A ≤ 31; 0 ≤ b ≤ 7	None	1	1/2	1001 1001 AAAA Abbb
SBIS	A, b	Skip if bit in I/O register is set	If (I/O(A,b) = 1) PC ← PC + (2 or 3)	0 ≤ A ≤ 31; 0 ≤ b ≤ 7	None	1	1/2	1001 1011 AAAA Abbb
BRBS	s, k	Branch if status flag set	If (SREG[s] = 1) PC ← PC + k + 1	0 ≤ s ≤ 7; -64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk ksss
BRBC	s, k	Branch if status flag cleared	If (SREG[s] = 0) PC ← PC + k + 1	0 ≤ s ≤ 7; -64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk ksss
BREQ	k	Branch if equal	If (Z = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k001
BRNE	k	Branch if not equal	If (Z = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k001
BRCS	k	Branch if carry set	If (C = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k000
BRCC	k	Branch if carry cleared	If (C = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k000
BRSH	k	Branch if same or higher	If (C = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k000
BRLO	k	Branch if lower	If (C = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k000
BRMI	k	Branch if minus	If (N = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k010
BRPL	k	Branch if plus	If (N = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k010

Mnemonic	Operands	Description	Operation		Flags	W	N	Binary Opcode
BRGE	k	Branch if greater or equal, signed	If (S = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k100
BRLT	k	Branch if less than zero, signed	If (S = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k100
BRHS	k	Branch if half carry flag set	If (H = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k101
BRHC	k	Branch if half carry flag cleared	If (H = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k101
BRTS	k	Branch if T flag set	If (T = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k110
BRTC	k	Branch if T flag cleared	If (T = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k110
BRVS	k	Branch if overflow flag set	If (V = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k011
BRVC	k	Branch if overflow flag cleared	If (V = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k011
BRIE	k	Branch if interrupt enabled	If (I = 1) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 00kk kkkk k111
BRID	k	Branch if interrupt disabled	If (I = 0) PC ← PC + k + 1	-64 ≤ k ≤ 63	None	1	1	1111 01kk kkkk k111
DATA TRANSFER INSTRUCTIONS								
MOV	Rd, Rr	Move between registers	Rd ← Rr	0 ≤ d ≤ 31; 0 ≤ r ≤ 31	None	1	1	0010 11rd dddd rrrr
MOVW	Rd, Rr	Copy register word	Rd+1:Rd ← Rr+1:Rr	d = 0, 2...30; r = 0, 2...30	None	1	1	0000 0001 dddd rrrr
LDI	Rd, K	Load immediate	Rd ← K	16 ≤ d ≤ 31; 0 ≤ K ≤ 255	None	1	1	1110 KKKK dddd KKKK
LD	Rd, X	Load indirect	Rd ← (X)	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1100
LD	Rd, X+	Load indirect and post-increment	Rd ← (X), X ← X + 1	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1101
LD	Rd, -X	Load indirect and pre-decrement	X ← X - 1, Rd ← (X)	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1110
LD	Rd, Y	Load indirect	Rd ← (Y)	0 ≤ d ≤ 31	None	1	1	1000 000d dddd 1000
LD	Rd, Y+	Load indirect and post-increment	Rd ← (Y), Y ← Y + 1	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1001
LD	Rd, -Y	Load indirect and pre-decrement	Y ← Y - 1, Rd ← (Y)	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1010
LDD	Rd, Y+q	Load indirect with displacement	Rd ← (Y + q)	0 ≤ d ≤ 31 0 ≤ q ≤ 63	None	1	1	10q0 qq0d dddd 1qqq
LD	Rd, Z	Load indirect	Rd ← (Z)	0 ≤ d ≤ 31	None	1	1	1000 000d dddd 0000
LD	Rd, Z+	Load indirect and post-increment	Rd ← (Z), Z ← Z + 1	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 0001
LD	Rd, -Z	Load indirect and pre-decrement	Z ← Z - 1, Rd ← (Z)	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 0010
LDD	Rd, Z+q	Load indirect with displacement	Rd ← (Z + q)	0 ≤ d ≤ 31 0 ≤ q ≤ 63	None	1	1	10q0 qq0d dddd 0qqq
LDS	Rd, k	Load direct from SRAM	Rd ← (k)	0 ≤ d ≤ 31 0 ≤ k ≤ 607	None	2	2	1001 000d dddd 0000 kkkk kkkk kkkk kkkk
ST	X, Rr	Store indirect	(X) ← Rr	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 1100

Mnemonic	Operands	Description	Operation		Flags	W	N	Binary Opcode
ST	X+, Rr	Store indirect and post-increment	(X) ← Rr, X ← X + 1	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 1101
ST	-X, Rr	Store indirect and pre-decrement	X ← X – 1, (X) ← Rr	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 1110
ST	Y, Rr	Store indirect	(Y) ← Rr	0 ≤ r ≤ 31	None	1	1	1000 001r rrrr 1000
ST	Y+, Rr	Store indirect and post-increment	(Y) ← Rr, Y ← Y + 1	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 1001
ST	-Y, Rr	Store indirect and pre-decrement	Y ← Y – 1, (Y) ← Rr	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 1010
STD	Y+q, Rr	Store Indirect with displacement	(Y + d) ← Rr	0 ≤ r ≤ 31 0 ≤ q ≤ 63	None	1	1	10q0 qq1r rrrr 1qqq
ST	Z, Rr	Store Indirect	(Z) ← Rr	0 ≤ r ≤ 31	None	1	1	1000 001r rrrr 0000
ST	Z+, Rr	Store indirect and post-increment	(Z) ← Rr, Z ← Z + 1	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 0001
ST	-Z, Rr	Store indirect and pre-decrement	Z ← Z – 1, (Z) ← Rr	0 ≤ r ≤ 31	None	1	1	1001 001r rrrr 0010
STD	Z+q, Rr	Store indirect with displacement	(Z + d) ← Rr	0 ≤ r ≤ 31 0 ≤ q ≤ 63	None	1	1	10q0 qq1r rrrr 0qqq
STS	K,Rr	Store direct to data space	(k) ← Rr	0 ≤ d ≤ 31 0 ≤ k ≤ 607	None	2	2	1001 001d dddd 0000 kkkk kkkk kkkk kkkk
LPM		Load program memory	R0 ← (Z)		None	1	2	1001 0101 1100 1000
LPM	Rd, Z	Load program memory	Rd ← (Z)	0 ≤ d ≤ 31	None	1	2	1001 000d dddd 0100
LPM	Rd, Z+	Load program memory and post-increment	Rd ← (Z)	0 ≤ d ≤ 31	None	1	2	1001 000d dddd 0101
IN	Rd, A	IO port in	Rd ← I/O(A)	0 ≤ d ≤ 31 0 ≤ A ≤ 63	None	1	1	1011 0AAd dddd AAAA
OUT	A, Rr	IO port out	I/O(A) ← Rr	0 ≤ d ≤ 31 0 ≤ A ≤ 63	None	1	1	1011 1AAr rrrr AAAA
PUSH	Rr	Push register on stack	STACK ← Rr	0 ≤ r ≤ 31	None	1	1	1001 001d dddd 1111
POP	Rd	Pop register from stack	Rd ← STACK	0 ≤ d ≤ 31	None	1	1	1001 000d dddd 1111
BIT AND BIT-TEST INSTRUCTIONS								
SBI	A, b	Set bit in I/O register	I/O(A,b) ← 1	0 ≤ A ≤ 31 0 ≤ b ≤ 7	None	1	1	1001 1010 AAAA Abbb
CBI	A, b	Clear bit in I/O register	I/O(A,b) ← 0	0 ≤ A ≤ 31 0 ≤ b ≤ 7	None	1	1	1001 1000 AAAA Abbb
LSL	Rd	Logical shift left	Rd[n+1] ← Rd[n], C ← Rd[7], Rd[0] ← 0	0 ≤ d ≤ 31	Z,C,N,V,H,S	1	1	0000 11dd dddd dddd
LSR	Rd	Logical shift right	Rd[n] ← Rd[n+1], C ← Rd[0], Rd[7] ← 0	0 ≤ d ≤ 31	Z,C,N,V,S	1	1	1001 010d dddd 0110
ROL	Rd	Rotate left through carry	Rd[0] ← C, Rd(n+1) ← Rd[n], C ← Rd[7]	0 ≤ d ≤ 31	Z,C,N,V,H,S	1	1	0001 11dd dddd dddd
ROR	Rd	Rotate right through carry	Rd[7] ← C, Rd[n] ← Rd[n+1], C ← Rd[0]	0 ≤ d ≤ 31	Z,C,N,V,S	1	1	1001 010d dddd 0111

Mnemonic	Operands	Description	Operation		Flags	W	N	Binary Opcode
ASR	Rd	Arithmetic shift right	C ← Rd[0], Rd[n] ← Rd[n+1], Rd[7] ← Rd[7]	0 ≤ d ≤ 31	Z,C,N,V,S	1	1	1001 010d dddd 0101
SWAP	Rd	Swap nibbles	Rd[3:0] ← Rd[7:4], Rd[7:4] ← Rd[3:0]	0 ≤ d ≤ 31	None	1	1	1001 010d dddd 0010
BSET	s	Flag set	SREG[s] ← 1	0 ≤ s ≤ 7	SREG[s]	1	1	1001 0100 0sss 1000
BCLR	s	Flag clear	SREG[s] ← 0	0 ≤ s ≤ 7	SREG[s]	1	1	1001 0100 1sss 1000
BST	Rr, b	Store bit from register to T	T ← Rr[b]	0 ≤ r ≤ 31 0 ≤ b ≤ 7	T	1	1	1111 101d dddd 0bbb
BLD	Rd, b	Load bit from T to register	Rd[b] ← T	0 ≤ r ≤ 31 0 ≤ b ≤ 7	None	1	1	1111 100d dddd 0bbb
SEC		Set carry	C ← 1		C	1	1	1001 0100 0000 1000
CLC		Clear carry	C ← 0		C	1	1	1001 0100 1000 1000
SEN		Set negative flag	N ← 1		N	1	1	1001 0100 0010 1000
CLN		Clear negative flag	N ← 0		N	1	1	1001 0100 1010 1000
SEZ		Set zero flag	Z ← 1		Z	1	1	1001 0100 0001 1000
CLZ		Clear zero flag	Z ← 0		Z	1	1	1001 0100 1001 1000
SEI		Global interrupt enable	N ← 1		I	1	1	1001 0100 0111 1000
CLI		Global interrupt disable	N ← 0		I	1	1	1001 0100 1111 1000
SES		Set signed test flag	S ← 1		S	1	1	1001 0100 0100 1000
CLS		Clear signed test flag	S ← 0		S	1	1	1001 0100 1100 1000
SEV		Set twos complement overflow	V ← 1		V	1	1	1001 0100 0011 1000
CLV		Clear twos complement overflow	V ← 0		V	1	1	1001 0100 1011 1000
SET		Set T in SREG	T ← 1		T	1	1	1001 0100 0110 1000
CLT		Clear T in SREG	T ← 0		T	1	1	1001 0100 1110 1000
SEH		Set half carry flag in SREG	H ← 1		H	1	1	1001 0100 0101 1000
CLH		Clear half carry flag in SREG	H ← 0		H	1	1	1001 0100 1101 1000
MCU CONTROL INSTRUCTIONS								
NOP		No operation			None	1	1	0000 0000 0000 0000
SLEEP		Sleep			None	1	1	1001 0101 1000 1000
WDR		Watchdog reset			None	1	1	1001 0101 1010 1000

Table 33: MCU instruction set, W denotes number of program words, N denotes number of clock cycles

6. Pinning

Pin #	Symbol	Description	Pin #	Symbol	Description
1	PC7/ SPI_CLK	MCU PORTC pin 7 GPIO	25	M0R2	M0 phase return input 2
2	CLKIO	Clock input/output	26	M0R3	M0 phase return input 3
3	DGND	Digital GND	27	OSEL	Osc select
4	NRESIN	Reset input	28	NPOR	Power On Reset Monitor (output)
5	NRESOUT	Reset output	29	PFC_PWM	PFC PWM output
6	SWV	SMPS voltage sense input	30	M0OFI	M0 over current input pin
7	SWVFB	SMPS voltage sense feedback	31	M0O1	M0 gate driver output 1
8	SWDRV	SMPS gate driver output	32	M0O2	M0 gate driver output 2
9	SWI	SMPS current sense input	33	M0O3	M0 gate driver output 3
10	SWVDD	SMPS supply voltage input	34	M0O4	M0 gate driver output 4
11	SWVREF	SMPS reference voltage	35	M0O5	M0 gate driver output 5
12	ADC0	ADC input channel MCU PORTE pin 0 GPIO	36	M0O6	M0 gate driver output 6
13	D5V2	Digital 5V input	37	DVDD1	2.5V digital supply voltage
14	VDC	PFC DC bus voltage input	38	PB0/ UART_TX	MCU PORTB pin 0 GPIO
15	VAC	PFC rectified AC voltage input	39	PB1/ UART_RX	MCU PORTB pin 1 GPIO
16	VREF	ADC reference voltage	40	SDI	SPI unit serial data input
17	AGND	Analog GND	41	SDO	SPI unit serial data output
18	AVDD	2.5V analog voltage output	42	SCS	SPI unit chip select input
19	A5V	Analog 5V input	43	SCLK	SPI unit clock input
20	PFCI	PFC AC current sense input	44	DVDD3	2.5V digital supply voltage
21	PFCIFB	PFC AC current sense feedback	45	D5V3	Digital 5V input
22	M0I	M0 current sense input	46	PC4/ SPI_CS	MCU PORTC pin 4 GPIO
23	M0IFB	M0 current sense feedback	47	PC5/ SPI_DO	MCU PORTC pin 5 GPIO
24	M0R1	M0 phase return input 1	48	PC6/ SPI_DI	MCU PORTC pin 6 GPIO

Table 34: AMG-XB403 Pin List

7. Absolute Maximum Ratings

The Absolute Maximum Ratings may not be exceeded under any circumstances.

#	Symbol	Parameter	Min	Max	Unit
1	V_{IN}	Voltage on any analog or digital signal pin	-0.3	V_{VDD} +0.3V	V
2	V_{SWDRV}	Voltage on switching supply driver output SWDRV	-0.3	V_{SWVDD} +0.3	V
3	V_{SWVDD}	Voltage on switching supply pin SWVDD	-0.3	6	V
4	V_{D5V}	Voltage on digital supply pin D5Vx	V_{SWVDD} - 0.3	V_{SWVDD} + 0.3	V
5	V_{A5V}	Voltage on analog supply pin A5V	-0.3	6	
6	V_{AGND}	Voltage on analog ground pin AGND	-0.3	0.5	V
7	V_{VD}	Voltage on digital supply pin DVDDx	-0.3	2.75	V
8	V_{VREF}	Voltage on analog reference voltage pin VREF	-0.3	2.75	V
9	I_{OUT}	Logic output current	-20	20	mA
10	V_{ESD}	ESD protection voltage (all pins)	-1	1	kV
11	T_{STB}	Storage temperature before programming	-40	125	°C
12	T_{STA}	Storage temperature after programming	-40	85	°C
13	T_J	Junction temperature		125[4]	°C

Table 35: AMG-XB403 Absolute Maximum Ratings

4 The junction temperature is given as a chip temperature which depends on the IC's total power dissipation. The junction temperature can be calculated as $T_J = T_A + R_{TH} \cdot P_{tot}$, where P_{tot} denotes the total power dissipation.

8. Electrical Characteristics

Unless otherwise specified, the minimum and maximum characteristics contain the spread of values guaranteed within the specified operating conditions.

Unless otherwise specified, typical values are given with D5V = A5A = 5V, T_A = 25°C

8.1. Operational Range

#	Symbol	Parameter	Min	Typ	Max	Unit
1	V_{SWVDD}	Voltage on switching supply pin SWVDD	4.3		5.5	V
2	V_{D5V}	Voltage on general supply pin D5V	4.5	5.0	5.5	V
3	V_{AGND}	Voltage on analog ground pin AGND	-0.1	0	0.1	V
4	V_{INH}	Logic HIGH signal input voltage [V_{D5V}=5V]	1.7		5.3	V
5	V_{INL}	Logic LOW signal input voltage [V_{D5V}=5V]	0		0.8	V
6	T_A	Ambient operating temperature	-25	25	85	°C
7	R_{TH}	Package thermal resistance			40	K/W

Table 36: AMG-XB403 Operational Range

8.2. DC Characteristics

#	Symbol	Parameter	Min	Typ	Max	Unit
1	V_{VREF}	Voltage on analog reference pin VREF	1.62	1.8	1.98	V
2	V_{VCC}	Voltage on analog supply pin AVDD	2.25	2.5	2.75	V
3	V_{VD}	Voltage on logic core supply pin DVDDx	2.25	2.5	2.75	V
4	V_{SWVREF}	Voltage on switching supply reference pin SWVREF		1.29		V
5	V_{VSWDRV}	HIGH voltage on switching supply driver pin SWDRV [I_{VSWDRV}=20mA]	3.5		V_{SWVDD}	V
6	V_{POR}	Power-on reset voltage	3.8	4.1	4.4	V
7	V_{BO}	Brown-out voltage	3.3	3.6	3.9	V
8	N_{TEMP}	Temperature channel resolution		0.625		K/bit
9	N_{VDD}	Voltage channel resolution		8		mV/bit

Table 37: AMG-XB403 DC Characteristics

8.3. AC Characteristics

#	Symbol	Parameter	Min	Typ	Max	Unit
1	f_{CLKI}	Internal RC-oscillator frequency	175	350	500	kHz
2	d_f	Internal RC-oscillator frequency deviation	-2		+2	%
3	f_{IN}	Initial FLL frequency	22.5	45	64	MHz
4	T_{POR}	Power-on reset delay time [after D5V 0V→5V step]		5		ms
5	T_{LOCK}	FLL reset delay time		10		ms
6	f_{SW}	Switching supply frequency (RC oscillator) [Start up, before setting divider]	10	22	35	kHz
7	f_{SCL}	EEPROM Clock Frequency, SCL			1000	kHz
8	t_{LOW}	EEPROM Clock Pulse Width Low	0.4			µs
9	t_{HIGH}	EEPROM Clock Pulse Width High	0.4			µs
10	t_{AA}	EEPROM Clock Low to Data Out Valid	0.05		0.55	µs
11	$t_{HD.STA}$	EEPROM Start Hold Time	0.25			µs
12	$t_{SU.STA}$	EEPROM Start Setup Time	0.25			µs
13	$t_{HD.DAT}$	EEPROM Data In Hold Time	0			µs
14	$t_{SU.DAT}$	EEPROM Data In Setup Time	0.1			µs
15	t_R	EEPROM Inputs Rise Time			0.3	µs
16	t_F	EEPROM Inputs Fall Time			0.1	µs
17	$t_{SU.STO}$	EEPROM Stop Setup Time	0.25			µs
18	t_{DH}	EEPROM Data Out Hold Time	50			ns
19	t_{WR}	EEPROM Write Cycle Time			5	ms

Table 38: AMG-XB403 AC Characteristics

8.4. ADC Characteristics

#	Symbol	Parameter	Min	Typ	Max	Unit
1	N_A	Resolution	-	10	-	bit
2	f_{ADC}	Conversion rate	-	2	-	MSmp/s
3	INL	Integral non-linearity [0.1V ≤ V_{IN} ≤ 4.8V]	-5		5	LSB
4	DNL	Differential non-linearity [0.1V ≤ V_{IN} ≤ 4.8V]	-1		4	LSB
5	O_{offset}	Offset [0.1V ≤ V_{IN} ≤ 4.8V]	-1		1	LSB

Table 39: AMG-XB403 ADC Characteristics

8.5. Supply Current Characteristics

#	Symbol	Parameter	Min	Typ	Max	Unit
1	I_{D5V_SLEEP}	Idle mode digital supply current [f_{FLL}=64MHz, all units off]		40		mA
2	I_{D5V_OP}	Operating mode digital supply current [f_{FLL}=64MHz, all units on]		60	90	mA
3	I_{A5V_SLEEP}	Quiescent analog supply current [f_{FLL}=64MHz, bias off]		1.2		mA
4	I_{A5V_OP}	Operating analog supply current [f_{FLL}=64MHz, all amplifiers enabled]		7.7	12	mA
5	I_{SWVDDq}	SMPS quiescent supply current [V_{SWVDD}=3.5V]		130		µA
6	I_{SWVDDo}	SMPS operating supply current [V_{SWVDD}=5V]		0.7	1.2	mA

Table 40: AMG-XB403 Supply Current Characteristics

8.6. Logic-Level Characteristics

#	Symbol	Parameter	Min	Typ	Max	Unit
1	$I_{IN,H}$	Logic HIGH signal input current	-0.1		0.1	µA
2	$-I_{IN,L}$	Logic LOW signal input current	-0.1		0.1	µA
3	$V_{OUT,H}$	Logic HIGH signal output voltage [I_{OUT}=-15mA, V_{D5V}=5V]	4.0	4.7		V
4	$V_{OUT,L}$	Logic LOW signal output voltage [I_{OUT}=15 mA, V_{D5V}=5V]		0.2	0.8	V

Table 41: AMG-XB403 Logic-level Characteristics

9. IC-Package

9.1. LQFP48 – Low-profile Quad Flat Package, 48 leads, RoHS compliant

- Body size: 7mm x 7mm x 1.4mm, Thermal resistance: R_{TH} = 40 K/W

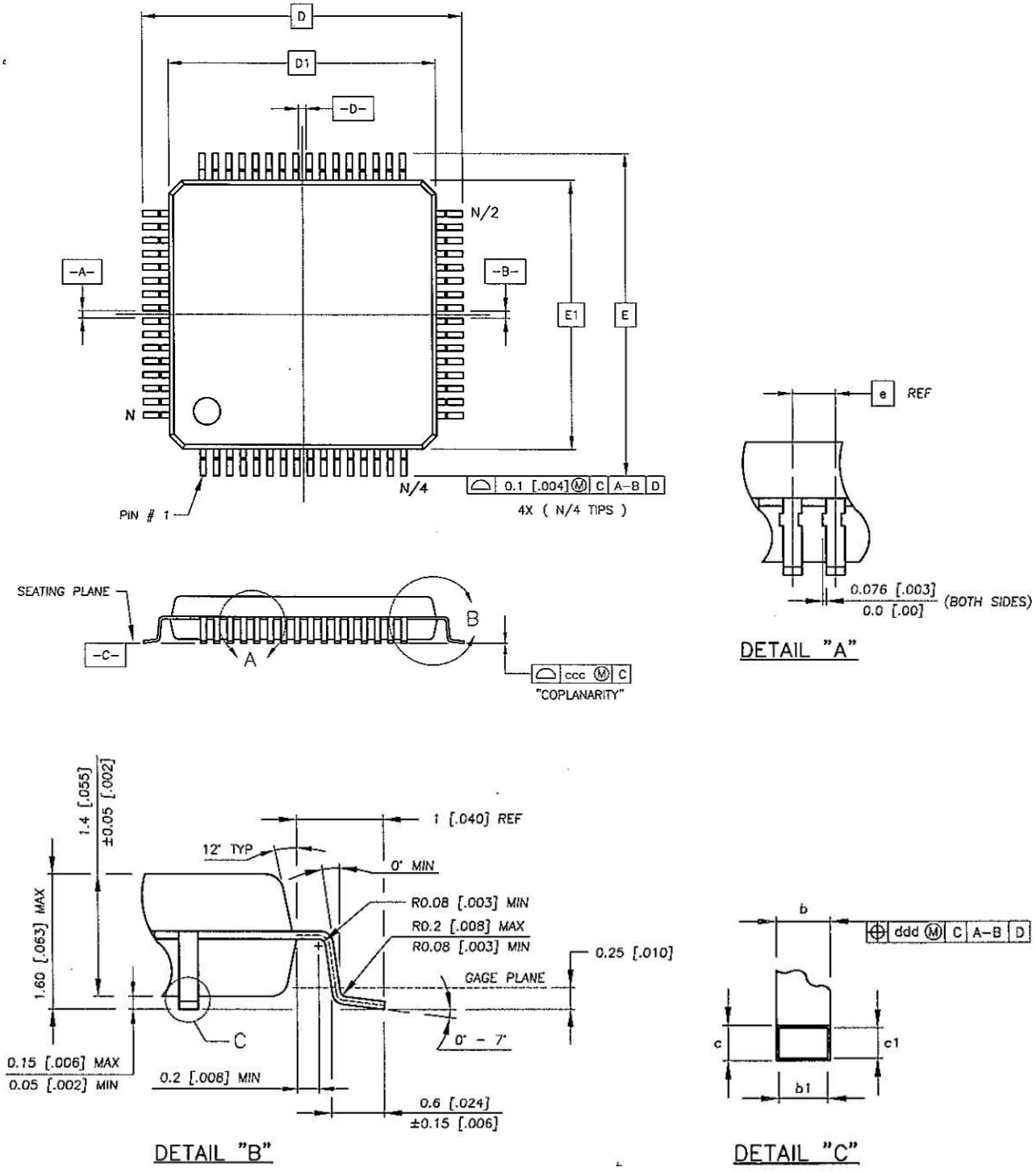


Figure 38: LQFP package drawing

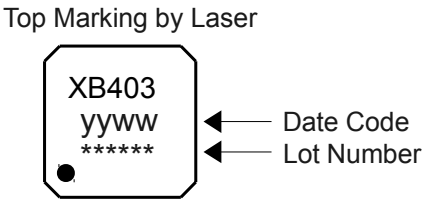
Notes:

1. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 [.010] per side. D1 and E1 are maximum plastic body size dimensions including mold mismatch
2. The top package body size may be smaller than the bottom body size by as much as 0.15 [.006]
3. Drawing conforms to JEDEC MS-026 Rev. D
4. Controlling dimensions in mm

Symbol	Min	Typ	Max
D	8.80	9.00	9.20
	.346	.354	.362
D1	6.90	7.00	7.10
	.272	.276	.280
E	8.80	9.00	9.20
	.346	.354	.362
E1	6.90	7.00	7.10
	.272	.276	.280
b	0.17	0.22	0.27
	.007	.009	.011
b1	0.17	0.20	0.23
	.007	.008	.009
c	0.09		0.20
	.004		.008
c1	0.09		0.16
	.004		.006
e		0.80	
		.031	
ccc		0.10	
		.004	
ddd		0.20	
		.008	
N		**48**	

Table 42: LQFP48 package dimensions

10. IC-Marking



11. Packing Specification

- ❑ IC in tray (ESD bag, vacuum sealed, maximum 10 trays per bag)
- ❑ 250 ICs per tray

12. Notes and Cautions

12.1. ESD Protection

The Requirements for Handling Electrostatic Discharge Sensitive Devices are described in the JEDEC standard JESD625-A. Please note the following recommendations:

- ❑ When handling the device, operators must be grounded by wearing a for the purpose designed grounded wrist strap with at least 1MΩ resistance and direct skin contact.
- ❑ Operators must wear at all times ESD protective shoes or the area should be surrounded by for ESD protection intended floor mats.
- ❑ Opening of the protective ESD package that the device is delivered in must only occur at a properly equipped ESD workbench. The tape with which the package is held together must be cut with a sharp cutting tool, never pulled or ripped off.
- ❑ Any unnecessary contact with the device or any unprotected conductive points should be avoided.
- ❑ Work only with qualified and grounded tools, measuring equipment, casing and workbenches.
- ❑ Outside properly protected ESD-areas the device or any electronic assembly that it may be part of should always be transported in EGB/ESD shielded packaging.

12.2. Storage Conditions

The AMG-XB403 corresponds to moisture sensitivity classification MSL2, according to JEDEC standard J-STD-020, and should be handled and stored according to J-STD-033.

13. Disclaimer

Information given in this data sheet is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties that may result from its use. alpha microelectronics gmbh does not authorize or warrant any of its products for use in life support system equipment.

The values stated in Absolute Maximum Ratings may under no circumstances be exceeded. No warranty is given for use in life support systems or medical equipment without the specific written consent of alpha microelectronics gmbh. For questions regarding the application please contact the publisher.

The declared data are only a description of the product. They are not guaranteed properties as defined by law. Examples are given without obligations and cannot give rise to any liability. Reprinting of this data sheet – or any part of it – is not allowed without the license of the publisher. Data sheets are subject to change without any notice.

14. Contact Information

This data sheet is published by alpha microelectronics gmbh. To order samples or inquire information please contact:

alpha microelectronics gmbh
Im Technologiepark 1
15236 Frankfurt (Oder)
Germany

am.info@alpha-microelectronics.de
www.alpha-microelectronics.de

+49-335-557-1750 (telephone)
+49-335-557-1759 (fax)