AS8267 / AS8268
Single-Phase 2-Current Energy Measurement Integrated Circuits with Microcontroller, RTC, Programmable Multi-Purpose I/Os, LCD Driver and On-Chip FLASH Memory

DATA SHEET

1. Key Features

- Precision single-phase, one or two current input energy measurement front-end including Sigma-Delta modulators for A/D-conversion and digital signal processor (DSP).
- Low current consumption of 5mA, depending on MCU activity.
- Digital phase correction and selectable gain on both current channels for use with two current transformers (CT) or one CT and one shunt.
- Power-supply monitor (PSM) for power-on reset and reset when the supply voltage falls below a defined threshold.
- Customer programmable 8-bit 8051 compatible microcontroller (MCU).
- Programmable MCU clock for optional low power operating conditions.
- Highly reliable 32kBytes of non-volatile Flash memory is provided on-chip for storage of both program and data.
- Program and data security is provided by optional password and attack counter protection.
- 2 x Universal Asynchronous Receiver / Transmitters (UART) for external communications such as programme download and debugging.
- Programmable watchdog timer (WDT) and external system reset pin.
- Real-time clock/calendar (RTC) with on-chip digital calibration and separate battery supply pin.
- On-chip temperature sensor for optional temperature compensation.
- On-chip voltage reference (VREF) with small temperature coefficient (15ppm/K typ.).
- Low power 3.0 – 4.0MHz crystal oscillator.
- SPI compatible interface for optional external non-volatile EEPROM memory selectable up to 32kBytes.
- Mains current lead/lag status indication for reactive energy measurement.
- Low power battery operating mode for meter reading when Mains voltage is not present.
- AS8267: 20 x 4 segment LCDD
  9 x multi-purpose I/O (MPIO)
- AS8268: 24 x 4 segment LCDD
  12 x multi-purpose I/O (MPIO)

2. General Description

The AS8267 / AS8268 are highly integrated CMOS single-phase energy metering devices for fully electronic LCD meter systems. The AS8267 / AS8268 have been designed to ensure a meters full compliance with the international Standards IEC62052 and ANSI.

The AS8267 / AS8268 ICs include all the functions required for conventional 1 current or 2-current anti-tamper meters. The functions include precision energy measurement, an 8-bit microcontroller unit (MCU) with 32kBytes of Flash memory, an on-chip Liquid Crystal Display driver (LCDD), programmable multi-purpose Inputs/Outputs (MPIO), a real time clock/calendar (RTC) for complex tariff functions such as time-of-use or maximum demand billing and a Serial Peripheral Interface (SPI) for reading data from and writing data to an optional external non-volatile memory (EEPROM).

The AS8267 / AS8268 ICs have a dedicated energy measurement front-end, which includes an analog front-end and programmable Digital Signal Processor (DSP) from which active energy, mains voltage and mains current are provided. Reactive and apparent energy can also be calculated.

The on-chip 8-bit 8051 compatible microcontroller is freely programmable and provides user access to the various functional blocks. The dedicated Universal Asynchronous Receiver / Transmitter (UART1) in the System Control block allows access to various system functions and blocks. A second UART (UART2) is also provided, which may for example be used for debugging. The on-chip memory includes 32kByte of highly reliable non-volatile Flash program (and data) memory and

1kByte volatile data memory. The meter system designer also has the option of an additional external EEPROM memory, which is selectable in size from 1kByte to 32kByte (in binary steps).

Program and data stored in the on-chip non-volatile Flash memory can be secured by password protection, in addition to an attack counter which 'locks' access after 5 unauthorised attacks.

An on-chip programmable watchdog timer (WDT) is available to automatically initiate a system reset if a regular 'hold-off' signal is not detected.

The system timing and real time clock (RTC) has a dedicated external battery supply pin (VDD_BAT), enabling the oscillator and RTC to continue operation during 'power-down'. The RTC may be digitally calibrated for oscillator frequency accuracy.

The on-chip temperature sensor provides the meter designer the option of temperature compensation for any of the measured parameters or functional blocks provided, over the full operating temperature range of the device.

The LCD Driver (LCDD) block enables the display of information provided by the microcontroller, directly to the LCD. Two dedicated data register banks are provided to simplify programming, particularly in the case where scrolled display data is required.

The programmable multi-purpose I/O pins (MPIO) may be independently configured as inputs or outputs. All the I/O pins are programmable for data direction, pull-up/pull-down resistors and drive strength (4mA/8mA). Typical functions may include LED energy consumption pulse output, energy direction and fault condition indication depending on current 1 or current 2 being active for the energy calculation, push button for display scrolling, mains isolation relay control for prepayment meters, optical interface etc.

An on-chip analog ground buffer (ABUF) and voltage reference (VREF) ensures that no external circuitry is required. A power-supply monitor (PSM) provides a reset, when VDD falls below a safe operating threshold.

A reset pin (RES_N) is available for external system reset.

The AS8267 / AS8268 ICs are available in LQFP64 plastic package.

3. Typical Application Circuit

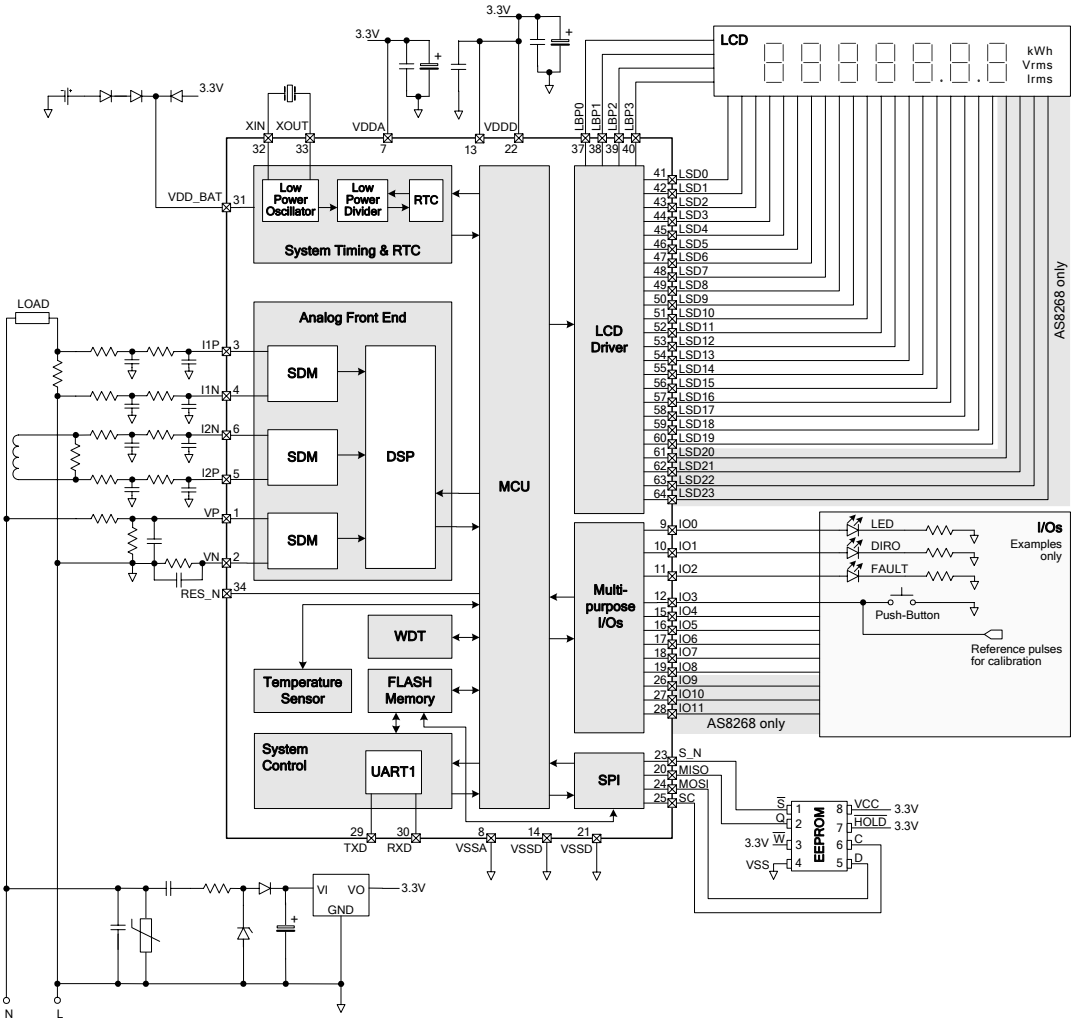
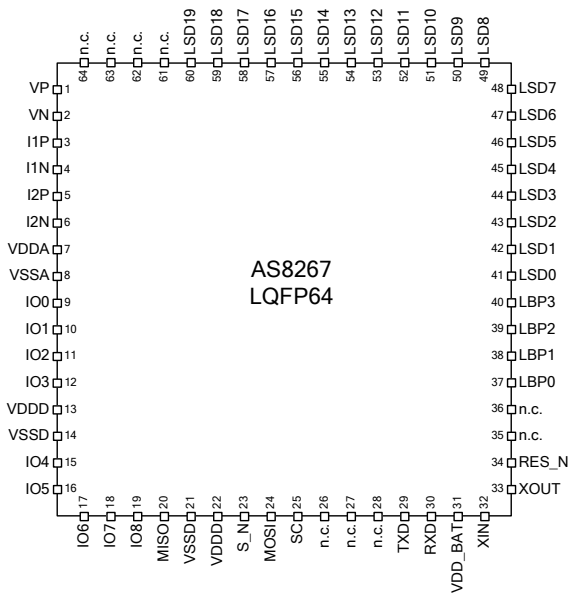
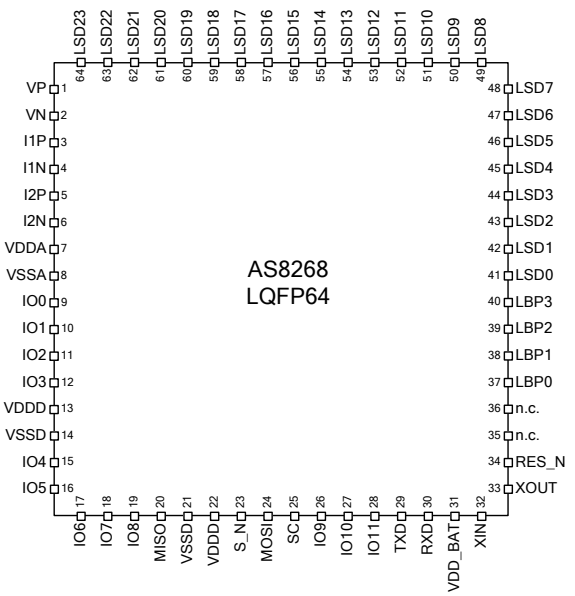


Figure 1: Typical application circuit of the AS8267 / AS8268

4. Pin Out





5. Pin Description

Pin No.	Pin Name AS8267	Pin Name AS8268	Type	Description
1	VP	VP	AI	Positive input for the voltage channel. VP is a differential input with VN. The typical differential voltage is ±100mV peak.
2	VN	VN	AI	Negative input for the voltage channel. VN is a differential input with VP.
3	I1P	I1P	AI	Positive input for the first current channel. I1P is a differential input with I1N. The input gain is programmable depending on the desired current sensor. The typical differential voltage is ±150mV peak (Gain = 4).
4	I1N	I1N	AI	Negative input for the first current channel. I1N is a differential input with I1P. The input gain is programmable depending on the desired current sensor. The typical differential voltage is ±150mV peak (Gain = 4).
5	I2P	I2P	AI	Positive input for the second current channel. I2P is a differential input with I2N. The input gain is programmable depending on the desired current sensor. The typical differential voltage is ±150mV peak (Gain = 4).
6	I2N	I2N	AI	Negative input for the second current channel. I2N is a differential input with I2P. The input gain is programmable depending on the desired current sensor. The typical differential voltage is ±150mV peak (Gain = 4).
7	VDDA	VDDA	S	Positive supply voltage for the analog circuitry. The required supply voltage is 3.3V ±10%.
8	VSSA	VSSA	S	Ground reference for the analog circuitry.
9	IO0	IO0	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
10	IO1	IO1	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.

Pin No.	Pin Name AS8267	Pin Name AS8268	Type	Description
11	IO2	IO2	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
12	IO3	IO3	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
13	VDDD	VDDD	S	Positive supply voltage to the digital circuitry and is internally connected to pin 22. The required supply voltage is 3.3V ±10%.
14	VSSD	VSSD	S	Ground reference for the digital circuitry.
15	IO4	IO4	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
16	IO5	IO5	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
17	IO6	IO6	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
18	IO7	IO7	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
19	IO8	IO8	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
20	MISO	MISO	DIOPD	Serial peripheral interface (SPI): Serial Data input in Master mode Serial Data output in Slave mode
21	VSSD	VSSD	S	Ground reference for the digital circuitry.
22	VDDD	VDDD	S	Positive digital supply. VDDD provides the positive supply voltage to the digital circuitry and is internally connected to pin 13. The required supply voltage is 3.3V ±10%.
23	S_N	S_N	DIOPU	Serial peripheral interface (SPI): Chip select
24	MOSI	MOSI	DIOPD	Serial peripheral interface (SPI): Serial Data output in Master mode Serial Data input in Slave mode
25	SC	SC	DIOPU	Serial peripheral interface (SPI): Serial clock
26	n.c.	IO9	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
27	n.c.	IO10	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
28	n.c.	IO11	DIO	Programmable multi-purpose input/output, with selectable pull-up or pull-down resistors and selectable drive strength.
29	TXD	TXD	DO	Universal Asynchronous Receiver/Transmitter (UART1) serial transmit data output.
30	RXD	RXD	DIPU	Universal Asynchronous Receiver/Transmitter (UART1) serial receive data input.
31	VDD_BAT	VDD_BAT	S	Battery backup supply voltage input for the real time clock (RTC).
32	XIN	XIN	AI	A 3.0 to 4.0MHz crystal may be connected across XIN and XOUT. Alternatively, an external clock signal may be applied to XIN.
33	XOUT	XOUT	AO	See XIN above, for the connection of a crystal. When an external clock is applied to XIN, XOUT is not connected.
34	RES_N	RES_N		System reset active low.
35	n.c.	n.c.		Not connected

Pin No.	Pin Name AS8267	Pin Name AS8268	Type	Description
36	n.c.	n.c.		Not connected
37	LBP0	LBP0	AO	LCD back-plane driver output signal.
38	LBP1	LBP1	AO	LCD back-plane driver output signal.
39	LBP2	LBP2	AO	LCD back-plane driver output signal.
40	LBP3	LBP3	AO	LCD back-plane driver output signal.
41	LSD0	LSD0	AO	LCD segment driver output signal.
42	LSD1	LSD1	AO	LCD segment driver output signal.
43	LSD2	LSD2	AO	LCD segment driver output signal.
44	LSD3	LSD3	AO	LCD segment driver output signal.
45	LSD4	LSD4	AO	LCD segment driver output signal.
46	LSD5	LSD5	AO	LCD segment driver output signal.
47	LSD6	LSD6	AO	LCD segment driver output signal.
48	LSD7	LSD7	AO	LCD segment driver output signal.
49	LSD8	LSD8	AO	LCD segment driver output signal.
50	LSD9	LSD9	AO	LCD segment driver output signal.
51	LSD10	LSD10	AO	LCD segment driver output signal.
52	LSD11	LSD11	AO	LCD segment driver output signal.
53	LSD12	LSD12	AO	LCD segment driver output signal.
54	LSD13	LSD13	AO	LCD segment driver output signal.
55	LSD14	LSD14	AO	LCD segment driver output signal.
56	LSD15	LSD15	AO	LCD segment driver output signal.
57	LSD16	LSD16	AO	LCD segment driver output signal.
58	LSD17	LSD17	AO	LCD segment driver output signal.
59	LSD18	LSD18	AO	LCD segment driver output signal.
60	LSD19	LSD19	AO	LCD segment driver output signal.
61	n.c.	LSD20	AO	LCD segment driver output signal.
62	n.c.	LSD21	AO	LCD segment driver output signal.
63	n.c.	LSD22	AO	LCD segment driver output signal.
64	n.c.	LSD23	AO	LCD segment driver output signal.

Note: Shaded pins above only available with AS8268 IC

PIN Types:

S	Supply pin
AI	Analog Input pin
AO	Analog Output pin
DIPU	Digital Input pin with pull-up resistor
DO	Digital Output pin
DIO	Programmable Digital Input or Output pin
DIOPD	Digital Input or Output pin with pull-down resistor
DIOPU	Digital Input or Output pin with pull-up resistor

Table of Contents

6. Electrical Characteristics

6.1 Absolute Maximum Ratings (Non-Operating)

Stresses beyond the 'Absolute Maximum Ratings' may cause permanent damage to the AS8267 / AS8268 ICs. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated under 'Operating Conditions' is not implied.

Caution: Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Parameter	Symbol	Min	Max	Unit	Notes
DC supply voltage	VDD	-0.3	+5.0	V	
Input pin voltage	V_{in}	-0.3	VDD+0.3	V	
Electrostatic discharge	ESD		1000	V	Norm: MIL 883 E method 3015
Storage temperature	T_{strg}	-55	125	°C	
Lead temperature profile	T_{lead}				Norm: IPC/JEDEC-020C
Humidity non-condensing		5	85	%	

6.2 Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Positive analog supply voltage	VDDA	3.0	3.3	3.6	V	
Negative analog supply voltage	VSSA	0		0	V	
Difference of supplies	A - D	-0.1		0.1	V	VDDA – VDDD VSSA – VSSD
Positive digital supply voltage	VDDD	3.0	3.3	3.6	V	
Negative digital supply voltage	VSSD	0		0	V	
Battery supply voltage	VDD_BAT	2.0	3.3	3.6	V	
Ambient temperature	T_{amb}	-40	25	85	°C	
Supply current	I_{supp}		5		mA	Depending on MCU activity
System clock frequency	f_{osc}	3.0	3.579545	4.0	MHz	

6.3 DC/AC Characteristics for Digital Inputs and Outputs

CMOS Input with Schmitt Trigger and Pull-up Resistor (RXD, RES_N)

Parameter	Symbol	Min	Typ	Max	Unit	Notes
High level input voltage	V_{IH}	0.7 x VDD			V	
Low level input voltage	V_{IL}			0.3 x VDD	V	
Low level input current	I_{IL}	-100		-15	µA	Tested at VDD=3.6V and Vin=0V

CMOS Output (TXD)

Parameter	Symbol	Min	Typ	Max	Unit	Notes
High level output voltage	V_{OH}	2.5			V	Tested at VDD=3.0V
Low level output voltage	V_{OL}			0.4	V	Tested at VDD=3.0V
High level output current	I_{OH}			4	mA	Tested at VDD=3.0V and Vout=VOH
Low level output current	I_{OL}	-4			mA	Tested at VDD=3.0V and Vout=VOL

MPIO Inputs with Pull-up or Pull-down Resistor (SC, S_N, MISO, MOSI)

Parameter	Symbol	Min	Max	Unit	Notes
High level input voltage	V_{IH}	0.7*VDD		V	
Low level input voltage	V_{IL}		0.3*VDD	V	
High level output voltage	V_{OH}	2.5		V	
Low level output voltage	V_{OL}		0.4	V	
High level output current	I_{OH}		4	mA	
Low level output current	I_{OL}	-4		mA	
Pull-up					
High level input leakage	I_{IH}	-1	1	µA	Tested at VDD=Vin=3.6V
Low level input current	I_{IL}	-100	-15	µA	Tested at VDD=3.6V and Vin=0V; ‘pull-up’
Pull-down					
High level input leakage	I_{IH}	100	15	µA	Tested at VDD=Vin=3.6V; ‘pull-down’
Low level input current	I_{IL}	-1	1	µA	Tested at VDD=3.6V and Vin=0V

Note:
V_{OH}, V_{OL}, I_{OH} and I_{OL} are tested at VDD=3.0V.
I_{OL} is tested at Vout=V_{OL}
I_{OH} is tested at Vout=V_{OH}

MPIO Inputs with Schmitt Trigger and Selectable Pull-up/Pull-down

Parameter	Symbol	Min	Typ	Max	Unit	Notes
High level input voltage	V_{IH}	0.7 x VDD			V	
Low level input voltage	V_{IL}			0.3 x VDD	V	
High level input current	I_{IH}	15		100	µA	Tested at VDD=3.6V and Vin=3.6V; 'pull-down'
Low level input current	I_{IL}	-100		-15	µA	Tested at VDD=3.6V and Vin=0V; 'pull-up'

MPIO Outputs with Programmable Drive Strength

Parameter	Symbol	Min	Typ	Max	Unit	Notes
High level output current	V_{OH}	2.5			V	Tested at VDD=3.0V
Low level output current	V_{OL}			0.4	V	Tested at VDD=3.0V
High level output current	I_{OH}			4	mA	If '4mA' is selected. Tested at VDD=3.0V and Vout=VOH
Low level output current	I_{OL}	-4			mA	If '4mA' is selected. Tested at VDD=3.0V and Vout=VOL
High level output current	I_{OH}			8	mA	If '8mA' is selected. Tested at VDD=3.0V and Vout=VOH
Low level output current	I_{OL}	-8			mA	If '8mA' is selected. Tested at VDD=3.0V and Vout=VOL

LCDD Outputs

The Liquid Crystal display driver (LCDD) outputs are specified in the LCD Driver section of this data sheet.

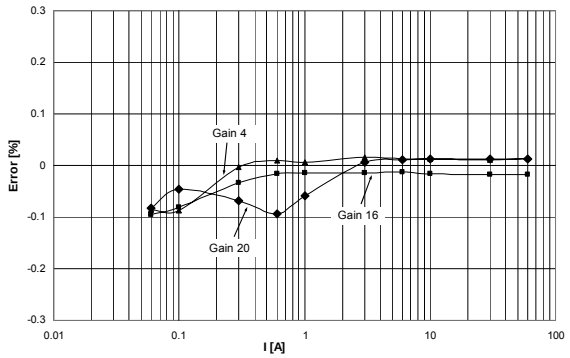
6.4 Electrical System Specification

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Input Signals						
Voltage channel input voltage	$\|V_{VP}\|$		100	212	mVp	Referenced to VSSA
Current channel input voltage (Gain=4)	$\|V_{I1P}\|$, $\|V_{I2P}\|$		150	212	mVp	Referenced to VSSA
Current channel input voltage (Gain=16)	$\|V_{I1P}\|$, $\|V_{I2P}\|$		38	54	mVp	Referenced to VSSA
Current channel input voltage (Gain=20)	$\|V_{I1P}\|$, $\|V_{I2P}\|$		30	42	mVp	Referenced to VSSA
Mains frequency	f_{mains}	45		65	Hz	
Dynamic range current	DR(I)	600:1				
Dynamic range power	DR(P)	2000:1				
Accuracy		0.1			%	Reading
Error variation over dyn. range	err(dr)			0.2	%	1)

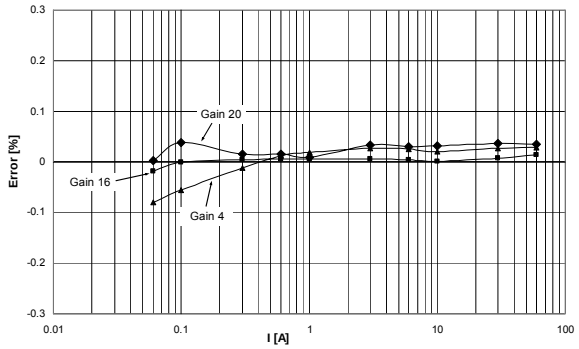
Parameter	Symbol	Min	Typ	Max	Unit	Notes
Error variation over temperature	err(temp)			0.5	%	Within operating temperature range, 1)
Error variation over cos(phi)	err(cosphi)			0.5	%	From 1 to 0.5, 1)
Error variation with VDD	err(VDD)			0.2	%	1)
Output pulse jitter	J			0.1	%	2)
Mains voltage	V_{mains}			264	V(rms)	240V + 10%, 3)
Measured current	I_{max}			120	A(rms)	3)
Measurement bandwidth	BW		1.75		kHz	

Notes:
1) Errors determined during energy measurement using a demo board and a reference meter with high accuracy (0.05%), which calculates the actual error.
2) Difference between largest and smallest error of 20 successive error samples; maximum meter constant: 1,600i/kWh; reference meter: 10,000 x DUT-meter-constant; measured at 5% Ib, Ib and I_{max}.
3) What is used for system considerations/calculations.

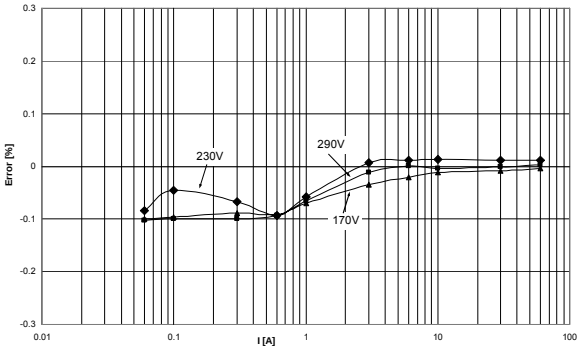
7. Performance Graphs



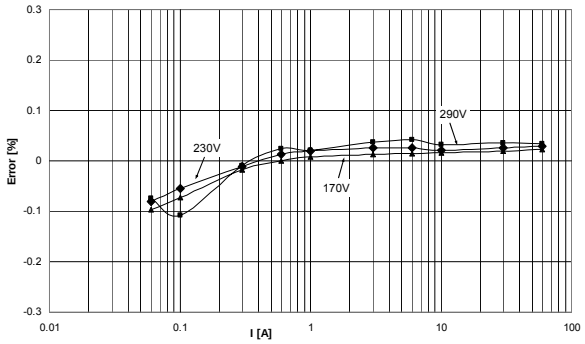
Graph 1: Error as a % of reading for gain setting 4, 16, 20 – Channel I1



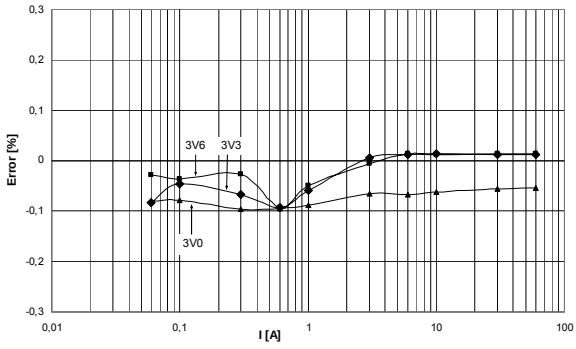
Graph 2: Error as a % of reading for gain setting 4, 16, 20 – Channel I2



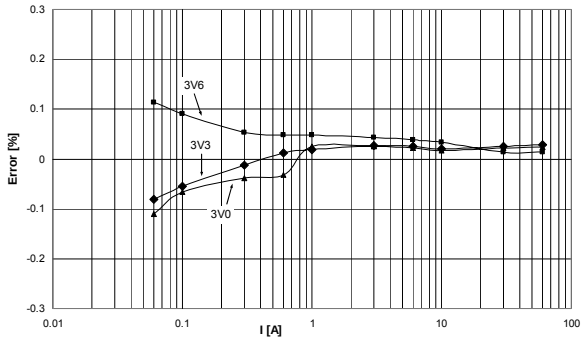
Graph 3: Error as a % of reading with mains voltage variation – Channel I1



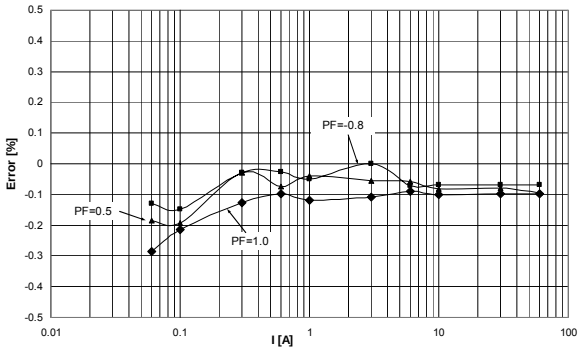
Graph 4: Error as a % of reading with mains voltage variation – Channel I2



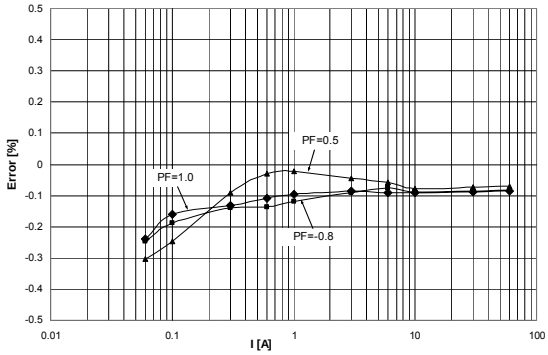
Graph 5: Error as a % of reading with variation in VDD – Channel I1



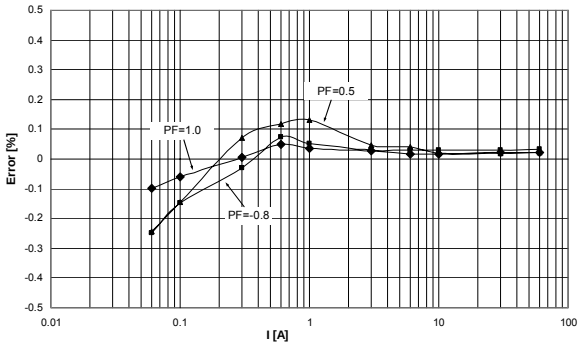
Graph 6: Error as a % of reading with variation in VDD – Channel I2



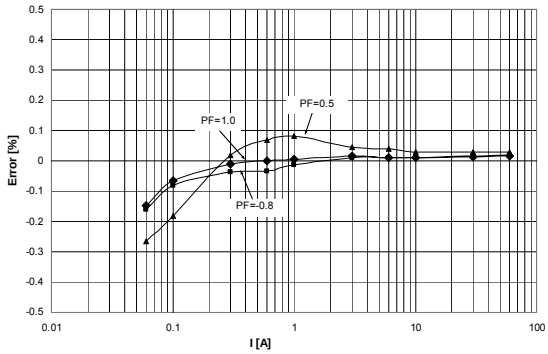
Graph 7: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at -40°C – Channel I1



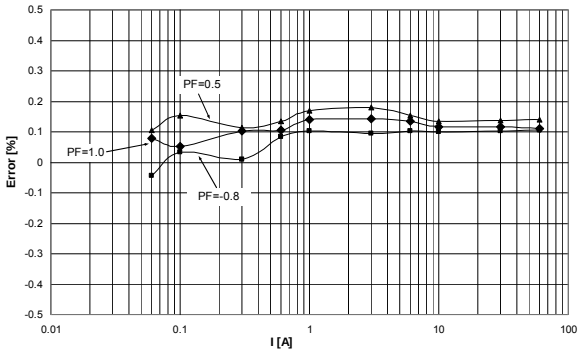
Graph 8: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at -40°C – Channel I2



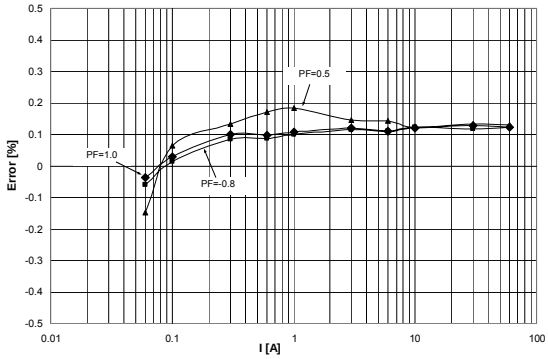
Graph 9: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at 25°C – Channel I1



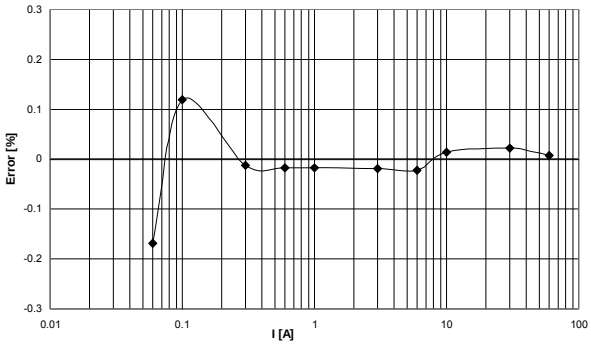
Graph 10: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at 25°C – Channel I2



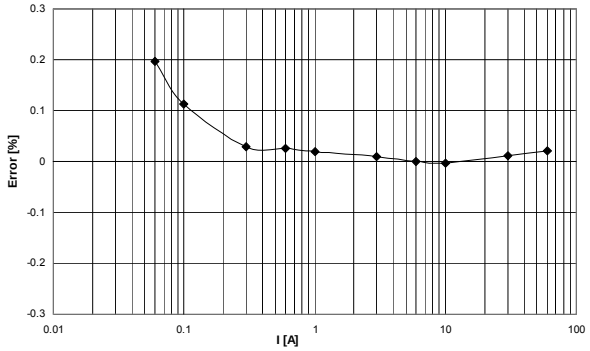
Graph 11: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at 85°C – Channel I1



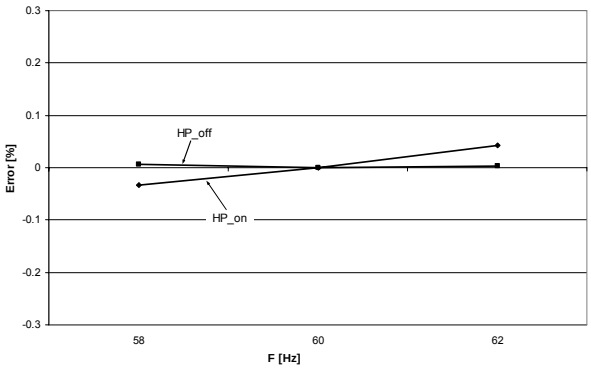
Graph 12: Error as a % of reading for PF=1, PF=-0.8, PF=0.5 at 85°C – Channel I2



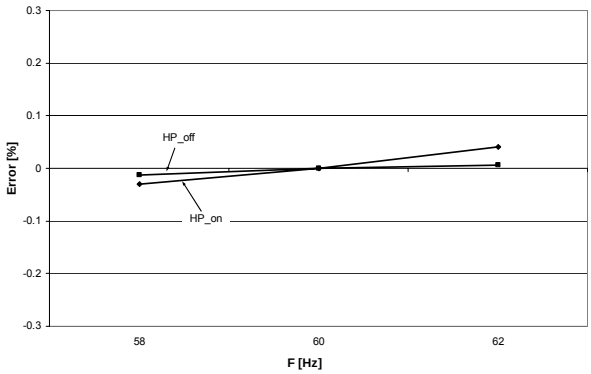
Graph 13: Error as a % of reading using vconst for mains voltage value – Channel I1



Graph 14: Error as a % of reading using vconst for mains voltage value – Channel I2



Graph 15: Error as a % of reading with variation in line frequency – Channel I1



Graph 16: Error as a % of reading with variation in line frequency – Channel I2

Note:
All measurements taken for the compilation of the graphs above were made using a reference meter design using the application circuit depicted on page 3 of this data sheet and incorporating the AS8267 / AS8268 integrated circuits.

For all graphical measurements, where the temperature was not specified, measurements were made at ambient (25 °C).

8. Detailed Functional Description

The AS8267 / AS8268 integrated circuits have a dedicated measurement front end, which is capable of measuring active and reactive energy, RMS mains voltage, RMS mains current as well as power factor. There are two completely separate differential current channel inputs, for measurement of both the Live and Neutral currents. The two current inputs may be connected to a shunt resistor (I1) and a current transformer (I2). Both current channels have programmable gains; thus it is possible to connect the shunt resistor to any of the two differential current inputs. The option to use two current transformers is also available. The AS8267 / AS8268 ICs may be programmed to accept either of the two measured currents for the energy calculation, or may be programmed to accept the larger of the two currents for the energy calculation.

The AS8267 / AS8268 ICs may also be used for conventional 1-phase single current measurement applications, where only the Live current is measured. In this case, the I2P and I2N pins are left unconnected and the second current channel modulator can be powered down.

The voltage channel input for measurement of the line voltage is also differential and is connected to a tap of a resistive divider of the line voltage. The resistive divider can be set to accommodate any line voltage standard (V_{mains}) including 100V, 110V, 220V, 230V or 240V.

A 3.0 to 4.0MHz low power oscillator generates the system clock for the AS8267 / AS8268 ICs. The absolute clock frequency may be calibrated on-chip. A low power divider is used to generate a 1Hz clock for the on-chip real time clock/calendar (RTC). The supply voltage to the low power oscillator, the low power divider and the RTC may be buffered with an external battery in case of mains power drops or failures.

The integrated temperature sensor can be used to compensate for temperature drift of the quartz crystal to improve measurement accuracy.

The LCD driver (LCDD) signals LSD0 ... LSD23 and LBP0 ... LBP3 can be directly connected to a liquid crystal display (LCD), which is used to display the various measured parameters. A total of 80 LCD segments may be driven by the AS8267 IC and 96 segments may be driven by the AS8268 IC.

A maximum of 12 programmable multi-purpose input/output (MPIO) pins are available for various meter functions, for example light-emitting diodes (LED) to signal energy consumption, energy direction, fault condition, etc. These I/O pins may also be programmed for use as bi-directional communication channels such as an optical interface or an additional Universal Asynchronous Receiver/Transmitter (UART2) Interface, should it so be required. The AS8267 has 9 x MPIO pins, while the AS8268 has 12 x MPIO pins.

A dedicated Serial Peripheral Interface (SPI) which can be configured as master or slave is also provided.
In master mode an external EEPROM (1kByte up to 32kByte) with a compatible serial peripheral interface can be connected if required.
In slave mode the interface allows direct access to the internal Flash memory.

The on-chip 8051 compatible microcontroller performs all the required calculations and enables the user to customize the input and output configuration of the meter. The microcontroller has a 1kB data memory, a square root calculation facility and a second UART (UART2) for debugging purposes.

The highly reliable 32kByte Flash memory allows storage of program and data. With the integrated security concept Flash Data can be protected against unauthorised access. The security concept offers password protection as well as an attack counter which blocks the Flash after five unauthorised attacks.

A programmable watchdog timer is provided to automatically initiate a system reset when a regular hold-off signal is not detected by the watchdog timer. The watchdog timer is an optional function which is software enabled.

A dedicated serial Universal Asynchronous Receiver/Transmitter (UART1) Interface within the System Control is provided to communicate with the AS8267 / AS8268 ICs and perform all the required programming and reading of data, especially during the meter production process.

The AS8267 / AS8268 ICs supply voltages (2 x VDDD and VDDA) are typically 3.3 Volts. These supply voltages should be derived from the V_{mains} with the use of a standard voltage regulated power supply circuit.

An on-chip power supply monitor (PSM) ensures that a reset is generated independently of the supply voltage rise and fall times. Monitoring of the V_{mains} is provided to ensure early power-down detection. A reset pin (RES_N) is also available for external system reset. The RES_N pin can be left unconnected if not required.

The individual functional elements of the AS8267 / AS8268 ICs, as well as the relationships between the various functional blocks are shown in the following block diagram. A detailed description of the AS8267 / AS8268 ICs system and the flexibility available to the kWh meter designer, through the system programmability is also described below:

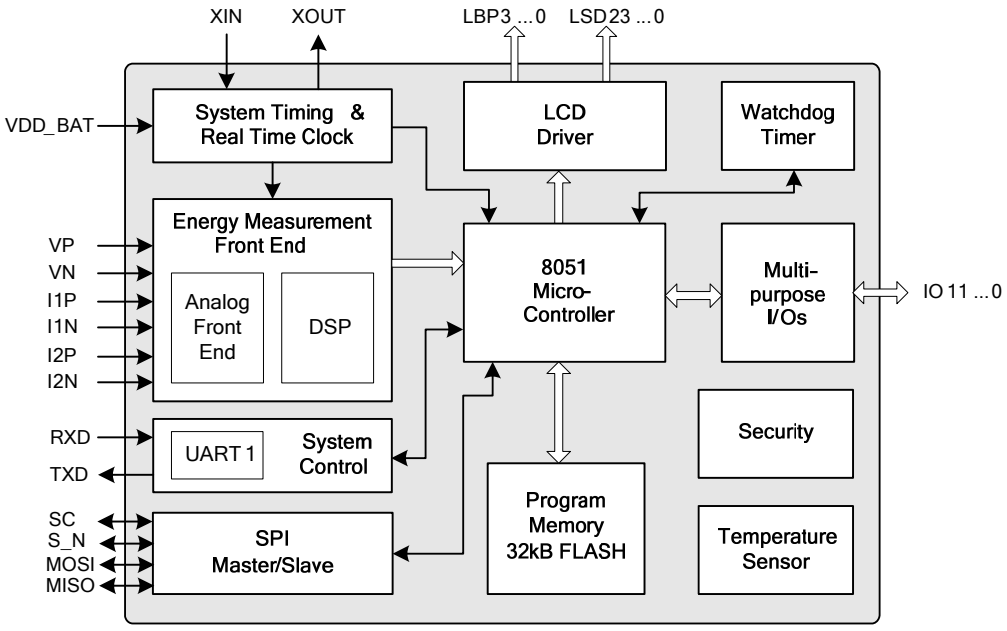


Figure 2: AS8267 / AS8268 block diagram

8.1 Energy Measurement Front End (Including DSP)

The Energy Measurement Front End is made up of the analog front end and the digital signal processing block (DSP), which performs the active energy measurement calculations for the microcontroller.

The analog front end comprises of the three Sigma-Delta modulators for the sampling of the mains voltage, Line current and a second current channel, for the optional measurement of the Neutral current. Also included in the analog front end is the voltage reference, which provides the temperature stability to the Sigma-Delta modulators. Setting up for the optimum input conditions for the voltage and current channels is also described in this section.

The digital signal processing block (DSP) provides the filtering and processing of the output data from the sigma-delta modulators and ensures that the specified measurement accuracy is provided by the AS8267 / AS8268. The DSP offers programming of measurement parameters and provides for fast and efficient meter production calibration procedures.

A power supply monitor (PSM) ensures that a reset is generated independently of the rise and fall times of the supply voltage (VDD). The PSM is also described in this section.

Analog Front End

The analog front end comprises of three identical Sigma-Delta modulators, which convert the differentially connected analog voltage and current inputs into digital signals. The two current inputs are gain adjustable to accommodate both directly connected or galvanically isolated current sensors.

The on-chip voltage reference (VREF) is the most important contributor to the accuracy of the AS8267 / AS8268 ICs due to it providing temperature stability to the circuit. Considering that the voltage and current signals are multiplied to derive the energy value, errors introduced prior to multiplication function results in errors being multiplied. Thus the introduction of errors into the voltage and the current channel inputs will result in a doubling of the percentage error after multiplication at the energy output.

The temperature coefficient of the VREF is specified at 15 ppm/K typical (30 ppm/K max.).

Current Inputs for Energy Calculation

The AS8267 / AS8268 ICs have 2 identical current inputs, I1P/I1N and I2P/I2N, for measurement of both the Live and Neutral currents. Either of the two current inputs may be selected for calculating the energy value.

These two differential current inputs are second order Sigma-Delta modulators, with each of the inputs being provided with selectable gains of 4, 16 and 20. The selectable gains are provided so that the AS8267 / AS8268 ICs may be easily adapted for use with either 2 current transformers or alternatively a shunt resistor and a current transformer for current sensing. The AS8267 / AS8268 ICs may also be used in a conventional single current configuration with either a current transformer or shunt resistor being used for current sensing.

The current input signal levels may be programmed by means of on-chip programmable gain settings. The required gain setting is selected as follows:

Current Input Gain Settings

Gain	Input Voltage	Comments
Current Inputs I1P, I1N		
20	-30mV≤V_{I1P}≤30mV	Shunt mode; default setting
16	-38mV≤V_{I1P}≤38mV	CT mode or shunt mode
4	-150mV≤V_{I1P}≤150mV	CT mode
Current Inputs I2P, I2N		
20	-30mV≤V_{I2P}≤30mV	Shunt mode
16	-38mV≤V_{I2P}≤38mV	CT mode or shunt mode
4	-150mV≤V_{I2P}≤150mV	CT mode; default setting

Notes:
1) Refer to the Settings Register (SREG) in the DSP section for programming of the Gain Settings.

For optimum operating conditions, the input signal at the Maximum Current (I_{max}) condition should be set at ±30mVp, when the Gain = 20, or ±150mVp, when the Gain = 4.
The default Gain, the AS8267 / AS8268 ICs current input gain settings without any programming required, is Gain = 20 for the I1 input and Gain = 4 for the I2 input.
The value of an ideal shunt resistor, may be calculated as follows:
Assuming an I_{max} rating of 60A (rms) → 84.85A (peak), then a shunt value of 350µΩ would be suitable.

$$R_{shunt} = \frac{30mV_p}{84.85A_p} = 354\mu\Omega$$ thus a standard 350µΩ shunt resistor may be selected.

The mains currents are sampled at 3.4956kHz, assuming that the recommended crystal oscillator frequency of 3.5795MHz, is used.

The current transformer(s) must be terminated with a voltage setting resistor (R_{VS}) to ensure the optimum voltage input level to the current input(s) of the AS8267 / AS8268 ICs. The value of R_{VS} is calculated as follows:

$$R_{VS} = \frac{V_{in(p)}}{I_L \sqrt{2}}$$

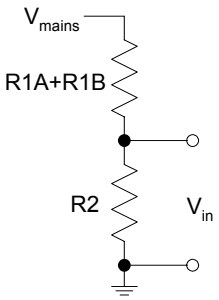
where I_L = CT RMS secondary current at rated conditions (V_{mains}; I_{max})
$V_{in(p)}$ = The peak input voltage to the IC at rated conditions (V_{mains}; I_{max}). For example, if Gain = 4, $V_{in(p)}$ should be set at 150mVpeak.

Example: A current transformer is specified at 60A/24mA and the Gain = 4:

$$R_{VS} = \frac{V_{in(p)}}{I_L \sqrt{2}} = \frac{150mV}{24mA \sqrt{2}} = 4.42\Omega \Rightarrow 4.3\Omega$$ thus a 4.3Ω Burden resistor may be selected

Voltage Input for Energy Calculation

The voltage channel input consisting of inputs VP and VN which are differential, with VP connected to the tap of a resistor divider circuit of the line voltage and VN connected to VSSA. For optimum operating conditions, the input signal at VP should be set at 100mVp for the rated voltage condition.
The resistor values for an ideal voltage divider may be calculated as follows:
Assuming a V_{mains} of 230V (rms) → 325V (peak) and R2 = 470Ω (according to the voltage divider shown below), the value of R1A+R1B may be calculated as follows:



$$R1A + R1B = R2 \times \frac{(V_{mains} - V_{in(P)})}{V_{in(P)}} = 470\Omega \times \frac{325V - 100mV}{100mV} = 1.53M\Omega$$

thus R1A = 820kΩ and R1B = 750kΩ resistors may be selected.

The mains voltage is also sampled at 3.4956kHz, assuming that the recommended crystal oscillator frequency of 3.5795MHz is used.

Digital Signal Processing Block (DSP)

The digital signal processing (DSP) block provides the signal processing required to ensure that the specified measured accuracy is performed and that the microcontroller (MCU) is provided with the appropriate data and protocol to perform all the required meter functions. For the description below, please refer to the following block diagram (Figure 3).

The DSP makes allowance for phase correction of the two current channels (i1 and i2) within the $Sinc^3$ decimation filters in the phase correction block. The applicable phase correction setting (pcorr_i1 or pcorr_i2) is selected (sel_i), depending upon which current (i1 or i2) is being used for the power calculation.

The equalization filters on the voltage and current channels which may be by-passed (sel_equ), correct for the attenuation introduced by the decimation filters at the edge of the input frequency band, while the high pass filters, which may also be by-passed (sel_hp), eliminate any DC offsets introduced into the input channels.

Independent calibration of the voltage (cal_v) and current signals (cal_i1 and cal_i2) is done after the voltage and current signals are provided for power calculation. This ensures that calibration of the voltage (sos_v), current channel 1 (sos_i1), current channel 2 (sos_i2) has no influence on the power (np) calibration.

The iMux (current multiplexer) allows the selection of the applicable current for power calculation (sel_i), while the vMux (voltage multiplexer) allows the selection of either the mains voltage data, or a constant voltage value, vconst (sel_v). The multiplication of the appropriately selected voltage and current signals is then performed.

After multiplication, the next multiplexer (sel_p) enables the selection of either instantaneous power or real power, which is derived through low pass filtering, PLP. The direction indicator output (diro) is derived from the output of the power low pass filter (PLP).

The following multiplexer (creep) allows the selection of the power signal, or blocks the power signal, depending on the required anti-creep and starting current thresholds, which may be set in the microcontroller.

Only when constant voltage value (vconst) is selected by the vMux (voltage multiplexer) or when diro=1, it is necessary to derive the absolute power value, for measurement (Abs).

The first pulse generator (Fast Pulse Gen) produces fast internal pulses, with the number of pulses being proportional to the measured energy. The multiplexer enables the selection of the appropriate pulse level (pulselev_i1 or pulselev_i2) depending on the current being used for energy measurement (sel_i). The output of the Fast Pulse Gen is always directly proportional to the LED pulse output, generated in the LED Pulse Gen. The LED output pulse rate is selectable (mconst). The polarity of the LED output pulses is also selectable (ledpol).

To ensure that the power data transferred to the microcontroller (MCU) is identical to that of the LED pulses, the power accumulator (P_ACCU) counts the pulses generated by the Fast Pulse Gen. After a defined number of sampling periods (nsamp), an interrupt is sent to the MCU, for the MCU to collect the accumulated energy data.

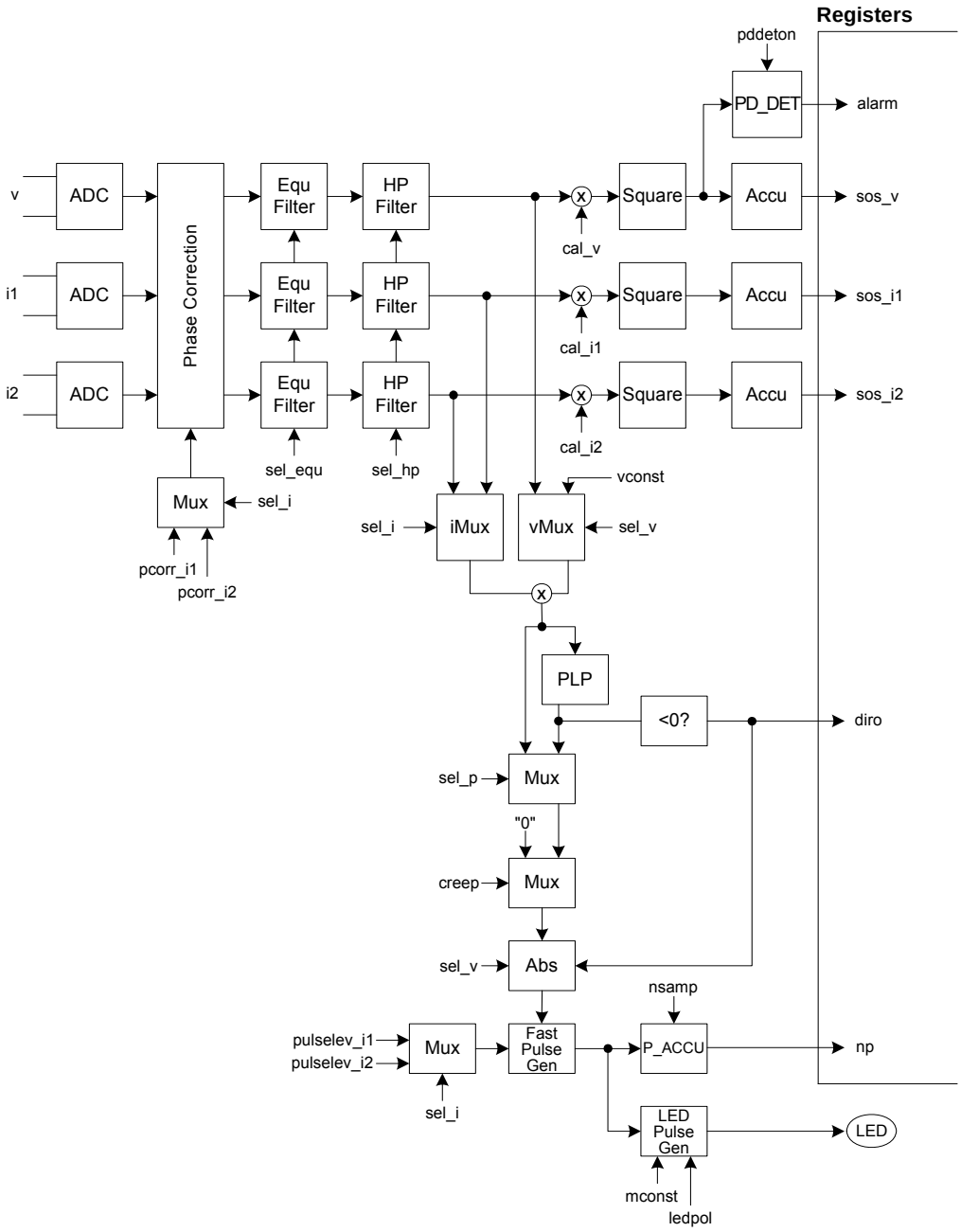


Figure 3: AFE block diagram

Phase Correction

The DSP provides phase correction of the two current channels (i1 and i2) by means of the Sinc^3 decimation filters in the phase correction block. Only one of the phase correction settings (pcorr_i1 or pcorr_i2) is valid at a time, depending on which current (i1 or i2) has been selected for the power calculation (sel_i).

The phase correction step size is dependent upon the main oscillator frequency selected (f_{osc}) and the mains frequency (f_{mains}). Assuming a 3.579545MHz crystal oscillator frequency and 50Hz mains frequency, the phase can be corrected in steps of 2.41' or 0.04 degrees.

pcorr									Phase Correction [unit(s)]
Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	1	1	1	1	1	1	1	1	255
0	1	1	1	1	1	1	1	0	254
									…
0	0	1	1	1	1	1	1	1	127
0	0	1	1	1	1	1	1	0	126
									…
0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	-1
1	1	1	1	1	1	1	1	0	-2
									…
1	1	0	0	0	0	0	0	1	-127
1	1	0	0	0	0	0	0	0	-128
									…
1	0	0	0	0	0	0	0	1	-255
1	0	0	0	0	0	0	0	0	-256

One 'unit' equals a certain phase shift related to the mains frequency:

$$1\ \text{unit} = 360° \times \frac{f_{mains}}{f_{osc}/8}$$

$$\text{Phase Correction}\left[°\right] = \#\text{units} \times 360° \times \frac{f_{mains}}{f_{osc}/8}$$

where f_{mains} is the mains frequency and f_{OSC} is the oscillator frequency.

Example:

$$1\,\text{unit} = 360° \times \frac{f_{mains}}{f_{osc}/8} = 0.04023° = 2.41°$$

255 units = 255 x 0.04023° = 10.26°

Calculating Phase Correction Factors

The measured phase_error in percentage is defined by the following formula

$$phase\_error = \frac{\cos(60° + phase\_shift) - \cos(60°)}{\cos(60°)} \times 100 \quad [\%]$$

while the phase_shift in degrees, is calculated as follows:

$$phase\_shift = \arccos\left[\left(1 + \frac{phase\_error[\%]}{100}\right) \times \cos(60°)\right] - 60°$$

phase_correction= –phase_shift

The required phase correction factor can be determined from error measurements with a power factor (PF) less than 1.
Assuming that at PF = 1 the meter has been calibrated and the error is approximately 0 for I_{cal} (calibration current), the PF is reduced and the effect of phase differences results in an increased error ('phase_error').

Example: The phase_error at PF = 0.5 (φ = 60°) is measured to be 9.2 %.

The related phase shift can be calculated using the following formula:

$$phase\_shift = \arccos\left[\left(1 + \frac{phase\_error[\%]}{100}\right) \times \cos 60°\right] - 60°$$

where the phase_error is the measured error in percentage and cosΦ is the phase angle.

For phase_error = 9.2[%] the phase_shift is -3.0° and therefore the phase correction is 3.0°.

If f_{osc} = 3.579545MHz and f_{mains} = 50Hz, one phase correction unit represents 2.41', which is 0.04023°.

Thus the phase correction factor must be set to

$$\frac{3.0°}{0.04023°} = 74.57 \; units$$

= 75 units.

The pcorr register has to be set to 4Bh.

Equalization Filters

The equalization filters in the voltage and current channels correct for the attenuation effects introduced by the decimation filters around the frequency band limit. The resulting transfer curve after the equalization filter has approximately 0dB attenuation over the entire frequency band.

The equalization filters may be by-passed (sel_equ), if required.

High-Pass Filters

The high pass filters in the voltage and current channels, with corner frequencies of <10Hz, correct for DC offsets introduced into the input channels.

Each of the voltage and current channels has a separate high pass filter in order to avoid any phase shift being introduced between the voltage and the two current channels.

The high pass filters may also be by-passed (sel_hp), if so desired.

Corner frequency: <10Hz

RMS Calculations

The DSP provides the voltage and current channel data in 'sum-of-squares' format. To calculate RMS values from the voltage (sos_v) and current (sos_i1 and sos_i2), the following formula should be applied for the voltage and current respectively:

$$V_{rms} = \sqrt{\frac{1}{nsamp}\sum_{i=1}^{nsamp} V_i^2}\ , \qquad \text{where } \sum_{i=1}^{nsamp} V_i^2 \text{ is the sos\_v value}$$

$$I_{rms} = \sqrt{\frac{1}{nsamp}\sum_{i=1}^{nsamp} I_i^2}\ , \qquad \text{where } \sum_{i=1}^{nsamp} I_i^2 \text{ is the sos\_i value}$$

nsamp should be selected in order to achieve coherent sampling as close as possible:

e.g. f_s = 3.4956kHz (f_{osc} = 3.579545MHz) ⇒ nsamp = 3496 should be selected if the MCU has to be interrupted every 1 second.

Refer to Squareroot Block (SQRT) for a detailed description of the programming sequence of the squareroot input operand.

Calibration of V and I Channels

The single channel data may be corrected with a 16-bit calibration value.
The calibration range is [1 LSB; 2 – 1 LSB], step size (1 LSB): 3.052 x 10^{-5}.

Calibration Register Setting	Value
0000h	0
0001h	0.00003052 (= 1 LSB)
…	…
2000h	0.25
…	…
4000h	0.5
…	…
8000h	1.0
…	…
FFFFh	1.99996948 (2 – 1LSB)

The V and I channel RMS calculation and calibration is described below (V and I channel are identical, thus only the I channel is shown):

The ideal values after RMS calculations of voltage and current are:

RMS_V(ideal) = 479(rms)
RMS_I(ideal) = 292,110(rms)

These values assume ideal input conditions with V_{in} = 100mVp at rated conditions and I_{in} = 30mVp (Gain = 20) at rated conditions.

Due to non-ideal components a different RMS value is calculated: RMS_I(actual). From this, the required calibration factor is calculated using the following formula:

$$cal\_i = \frac{RMS\_I(ideal)}{RMS\_I(actual)}$$

The following formula calculates the actual value to be programmed into the calibration registers (cal_v; cal_i1; cal_i2):

$$cal\_i(reg) = hex(round(cal\_i \times 32{,}768))$$

e.g.

RMS_I(ideal) = 292,110 at 40A

$$I_{cal} = 10A \rightarrow RMS\_I(ideal) = \frac{292{,}110}{4} = 73{,}027 \text{ at } 10A$$

$$I_{actual} = 9.2A \rightarrow RMS\_I(actual) = 67{,}185$$

$$cal\_i1(reg) = hex\left(round\left(\frac{73{,}027}{67{,}185} \times 32{,}768\right)\right)$$

$$= hex(round(35{,}617.31))$$

$$= hex(35{,}617)$$

$$= 8B21h$$

Constant Voltage Register (vconst)

The vconst registers (9334h and 9335h) provide a predefined voltage value that can be used for calculating energy when the V_{mains} is not available.

The default value of vconst is 2877 (0B3Dh) which translates into an equivalent V_{mains} value of 311V.

The energy is calculated using vconst and the selected current (i1 or i2) when sel_v in the SREG/Select register is set to '1'.
The vconst value may be calculated according to the formula:

$$vconst = RMS\_V \times \sqrt{2} \times \pi$$

Example: RMS_V = 479

$$\Rightarrow vconst = 479 \times \sqrt{2} \times \pi$$

$$= 2{,}128$$

(Once the voltage channel has been calibrated, 479 is the typical value when V_{mains} = 230V)

Note: When vconst is used for the calculation of energy, sel_p must be set to '0'.

Low Pass Filter for Real Power (PLP)

When the instantaneous power is low pass filtered the result is practically a DC value for the power, which is termed real power. It is generally preferred to use real power to generate pulses for the calibration, as the duration between pulses is more constant (pulse jitter).

Corner frequency: 18.6Hz

The low pass filter ensures that the power output pulse jitter is minimised.

Direction Indicator (DIRO)

The direction indicator (DIRO) situated in the Status Register (Bit 4) defines the direction of the measured power. The direction is determined by the phase relationship between the Mains voltage and selected Mains current (i1 or i2).

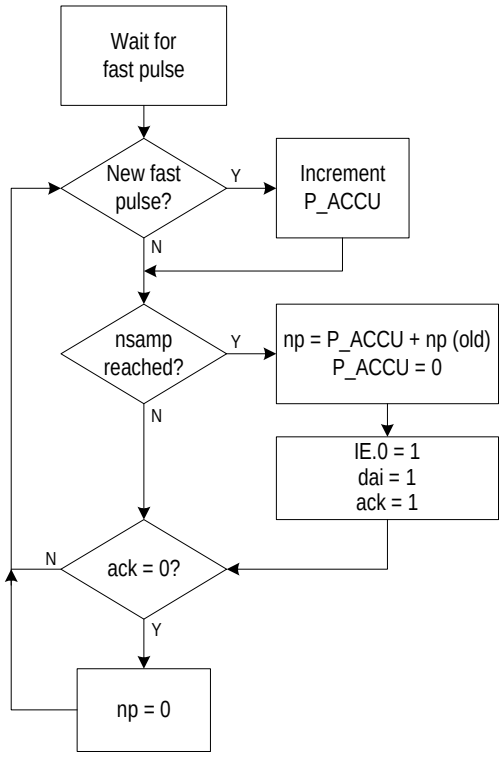
When bit 4 in the Status Register is '0', the Mains voltage and selected Mains current are in phase, thus indicating positive energy flow. When bit 4 in the Status Register is '1', the Mains voltage and the selected Mains current have a phase reversal, indicating negative energy flow. The energy calculation (np) is generated

from positive energy, thus when DIRO = 1, the negative energy is converted to positive energy by the ‘Abs’ block shown in Figure 3: AFE block diagram.

Should the meter application require unidirectional energy measurement, the MCU can separately derive both the positive and negative energy values, depending on the status of the DIRO bit.

Accumulator for Real Power (P_ACCU)

To ensure that the power information transferred to the MCU is identical to that of the LED pulses, the P_ACCU counts the pulses generated by Fast Pulse Gen. After ‘nsamp’ (nyquist) sampling periods an interrupt is sent to the MCU requesting to fetch the new energy information. (Interrupt line ‘IE.0’ goes high and the ‘data available interrupt’ (dai) flag in the SREG/Status register is set). The ‘ack’ bit in the SREG/Status register is also set to 1. If the MCU takes the energy information, it has to reset the ‘ack’ bit signalling that the energy information has been taken. If the ‘ack’ bit is not reset the P_ACCU will add the ‘old’ energy information to the ‘new’ energy information accumulated in the following cycle.
In any event, the MCU must reset the dai flag in order to clear the interrupt.



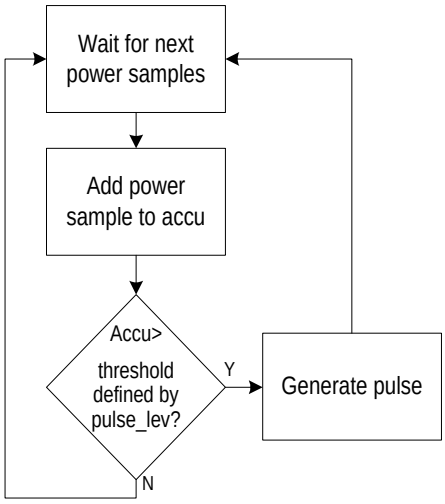
Note: The above flow chart assumes that the dai flag is always reset in time before the next interrupt is generated.

Pulse Generation

Two pulse generators are provided to ensure that virtually any LED pulse rate output can be programmed for display and calibration purposes. The first pulse generator (Fast Pulse Gen) produces fast internal pulses. These fast pulses are accumulated in the power accumulator (P_ACCU) for energy data transfer to the MCU.

The second pulse generator (LED Pulse Gen) produces the LED output pulses (meter constant) from the fast internal pulses. This type of data interface ensures that the MCU receives exactly the same energy information as is displayed by the LED pulses.

In case of 'creep', the power samples to be added will be set to 0.
The following flow chart shows the basic flow diagram for pulse generation:



The Fast Pulse Gen output pulse rate always has the same relationship with the LED pulse rate defined by mconst. Only if LED is calibrated to a meter constant different from those provided in the mconst table, will the fast internal pulse rate be different.

Formula for fast internal pulse rate (PR_{int}):

$$PR_{int} = 204{,}800 \times \frac{Target\,Pulse\,Rate}{mconst}\left[i/kWh\right]$$ where mconst is the meter constant.

$$1i = \frac{1{,}000 \times 3{,}600}{PR_{int}}\left[Ws\right]$$ when 1i is one impulse representing an energy equivalent.

e.g.

TargetPulseRate = mconst = 3200i/kWh

PR_{int} = 204,800 × 1 [i/kWh]

$$1i = \frac{1{,}000 \times 3{,}600}{204{,}800} = 17.58\left[Ws\right]$$

Active Power Calibration (Pulse_lev)

This paragraph describes how the active power measurement within the AS8267 / AS8268 ICs is calibrated.
The parameter Pulse_lev is the main parameter which determines the output frequency of the Fast Pulse Gen. This frequency relates to the measured power and is the basis from which the output pulse rate is derived.

Prior to system calibration, the appropriate value for the parameter Pulse_lev must be calculated to produce the required output pulse rate. The calibration exercise must accommodate all non-idealities that are present in the meter system.

The Pulse_lev is specified such that a typical pulse rate of 204,800i/kWh can be achieved.
During energy pulse calibration the correct Pulse_lev is determined in order to get the desired pulse rate.
The default value for Pulse_lev is defined for I_{max}=40A and V_{mains}=230V.

Default Pulse_lev: 570,950 (Pulse_lev(default))

Example for Pulse_lev calculations:

$$Pulse\_lev(ideal) = \frac{230V}{V_{mains}} \times \frac{40A}{I_{max}} \times Pulse\_lev(default)$$

V_{mains} (V)	I_{max} (A)	Pulse_lev (ideal)	Notes
230	100	228,380	
230	80	285,475	
230	60	380,633	
230	40	570,950	Default setting
230	20	1,141,900	
230	10	2,283,800	
240	100	218,864	
240	80	273,580	
240	60	364,774	
240	40	547,160	
240	20	1,094,321	
240	10	2,188,642	
Pulse_lev(ideal) = 230/V_{mains} x 40/I_{max} x 570,950			

Comparison Calibration Method

The most common calibration method is the comparison of energy reading of the meter under test (MUT) against a standard or reference meter. Normally, the standard, or reference meter has a considerably higher pulse rate than the meter under calibration. Reference meter output pulses are then counted between consecutive led pulses. To facilitate the calibration procedure, a pulse counter is provided in the MPIO block. In this case, the absolute calibration time and the calibration current are not relevant for the calibration cycle. The basic calibration setup is shown below:

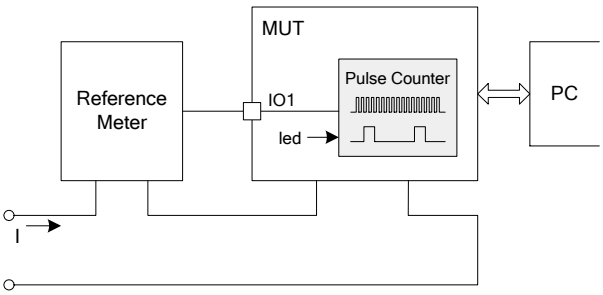


Figure 4: Basic setup for comparison calibration method (using IO1 as example input)

Note: An I/O used as push-button input can be used for the input of the reference meter pulses during calibration.

The standard or reference meter pulses are counted between two pulses from the meter to be calibrated. Ideally the sum of the pulses would exactly be the ratio between standard meter or reference pulse rate and the pulse rate of the meter under test. From the deviation the corrected Pulse_lev may be calculated.

$$Pulse\_lev(corrected) = Pulse\_lev(ideal) \times \frac{Ni}{Na},$$

where Ni is the ideal number of pulses and Na is the actual number of pulses (PCNT register in MPIO).
The actual number of pulses is available in the pulse count register (PCNT).

The ideal number of pulses Ni is the ratio between the pulse rates of the reference meter and the meter under test, which is always >1. The formula for Ni is as follows:

$$Ni = \frac{PR(ref)}{LED\,Pulse\,Rate(mconst)},$$

where PR(ref) is the reference meter constant.

The Pulse_lev (ideal) is calculated using the following formula:

$$Pulse\_lev(ideal) = \frac{230V}{V_{mains}} \times \frac{40A}{I_{max}} \times Pulse\_lev(default)$$

Example

The reference meter has a pulse rate, which is 10,000 times greater than the pulse rate of the AS8267 / AS8268 LED output.

LED Pulse Rate = 3,200i/kWh

PR(ref) = 3,200 x 10,000

$\Rightarrow$ Ni = 10,000

During a calibration cycle we measure 11,000 pulses between two LED pulses.

$\Rightarrow$ Na = 11,000

Assuming a meter with V_{mains} = 230V I_{max} = 60A

$$Pulse\_lev(ideal) = \frac{230V}{230V} \times \frac{40A}{60A} \times 570{,}950$$

$$= 380{,}633$$

$$Pulse\_lev(corrected) = 380{,}633 \times \frac{10{,}000}{11{,}000} = 346{,}030$$

LED Meter Constant Selection (mconst, 9330h)

The LED pulses are derived directly from the fast internal pulses (204,800i/kWh).
The 'mconst' register in SREG specifies the LED pulse rate:

MSB							LSB
-	-	-	-	mconst[3]	mconst[2]	mconst[1]	mconst[0]

Bit	Symbol	Function					
7	-	Not used					
6	-	Not used					
5	-	Not used					
4	-	Not used					
3	mconst[3]		Bit3	Bit2	Bit1	Bit0	LED Pulse Rate
			0	0	0	0	204,800
			0	0	0	1	102,400
			0	0	1	0	51,200
2	mconst[2]		0	0	1	1	25,600
			0	1	0	0	12,800
			0	1	0	1	6,400
			0	1	1	0	3,200
1	mconst[1]		0	1	1	1	1,600
			1	0	0	0	800
			1	0	0	1	400
0	mconst[0]		1	0	1	0	200
			1	0	1	1	100
			1	1	X	X	100

If the target meter constant is different from one of the selectable (mconst) meter constants defined above:

e.g. 1,000i/kWh (Target Pulse Rate)

The same formula $Ni = \frac{PR(ref)}{LED\,Pulse\,Rate(mconst)}$ can be used, but Ni is calculated using the Target Pulse Rate:

$$Ni = \frac{PR(ref)}{Target\,Pulse\,Rate}$$

(Important: Select a pulse rate which is close to mconst, for the Target Pulse Rate, so that the Pulse_lev stays within reasonable limits.)

After this calibration the energy equivalent of 1 fast pulse (1i) is different!

Standard: internal pulse rate: 204,800i/kIWh

$$\Rightarrow 1i = \frac{1{,}000 \times 3{,}600[Ws]}{204{,}800} = 17.58Ws$$

When a special pulse rate is required, the following formula applies:

$$\Rightarrow 1i = \frac{1{,}000 \times 3{,}600}{204{,}800} \times \frac{LED\,PulseRate}{Target PulseRate}[Ws]$$

Example:
Assuming a pulse rate of 1,000 is required:

1,600 → 204,800
1,000 → 204,800 x 1,000/1,600

$$\Rightarrow 1i = \frac{1{,}000 \times 3{,}600[Ws]}{204{,}800 \times 1{,}000 / 1{,}600} = 28.13Ws$$

Mains Current Leads/Lags Mains Voltage

The i_lead flag in the SREG/Status register determines if the mains current leads the mains voltage or lags the mains voltage. The data is provided for reactive power calculation, to establish if the measured power is capacitive or inductive.

LED Output Timing

The pulses on the LED output indicate the amount of energy that has been consumed over a certain period of time. Each pulse has an equivalent that can be set in the SREG/mconst register exactly. The unit is impulses per kWh (i/kWh).
This output may be used for calibration.
The polarity of the LED pulses may be selected via the ledpol bit in the SREG/Select Register for either positive or negative going pulses.

Timing Diagram

Timing Parameters

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Pulse width	t1		80		ms	50% duty cycle is enabled when the LED period is less than 160ms. For mconst=0, t1 will be 17.9µs.

Register Interface to MCU

One register block contains the data for the Meter Data Register (MDR) and the Settings Register (SREG), hence only one interface to the MCU is required.

Meter Data Register (MDR)

The meter data register is updated after ‘nsamp’ samples. Then an interrupt is issued to the MCU, which may take the energy data and process them further on. When an interrupt is generated the ‘ack’ bit in the SREG/Status register is set. If the MCU takes the data, it has to reset the ‘ack’ bit.
If the ‘ack’ bit has not been reset by the MCU when a new set of data is ready, the previous np value will be added to the new one.
In any case the dai flag in the SREG/Status register must be reset in order to clear the interrupt.

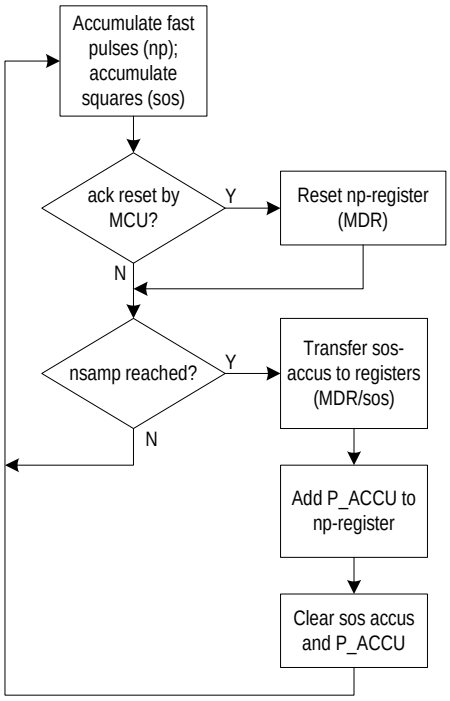
The following table shows the data which is available in the MDR:

Register Name	Address	Reset Value	Description
samptoend[7:0]	9300h	FFh	Indicates how many samples are left (until nsamp), before the next interrupt is generated. Using this information the MCU can determine if it still has time to transfer the MDR data to the MCU memory.
samptoend[15:8]	9301h	FFh	
np[7:0]	9302h	00h	number of fast pulses, equivalent to energy information accumulated during nsamp samples
np[15:8]	9303h	00h	
np[23:16]	9304h	00h	
np[31:24]	9305h	00h	
sos_v[7:0]	9306h	00h	sum of squares of voltage channel samples
sos_v[15:8]	9307h	00h	
sos_v[23:16]	9308h	00h	
sos_v[31:24]	9309h	00h	
sos_v[35:32]	930Ah	00h	
sos_i1[7:0]	930Bh	00h	sum of squares of current channel 1 samples
sos_i1[15:8]	930Ch	00h	
sos_i1[23:16]	930Dh	00h	
sos_i1[31:24]	930Eh	00h	
sos_i1[39:32]	930Fh	00h	
sos_i1[47:40]	9310h	00h	
sos_i1[53:48]	9311h	00h	
sos_i2[7:0]	9312h	00h	sum of squares of current channel 2 samples
sos_i2[15:8]	9313h	00h	
sos_i2[23:16]	9314h	00h	
sos_i2[31:24]	9315h	00h	
sos_i2[39:32]	9316h	00h	
sos_i2[47:40]	9317h	00h	
sos_i2[53:48]	9318h	00h	

Notes:

1) MDR is read-only for MCU. (except for ‘MCU debug mode’, then you can set the register values as described.)
2) Unused addresses will simply be ignored.

The following flowchart describes how accumulators and registers work together:



Calculation of Apparent Power

$$S[VA] = V_{rms} \times I_{rms} = \sqrt{\frac{1}{nsamp} \sum_{i=1}^{nsamp} \times I_i^2} \times \sqrt{\frac{1}{nsamp} \sum_{i=1}^{nsamp} \times V_i^2}$$

$$= \sqrt{\frac{1}{nsamp} \times sos\_v} \times \sqrt{\frac{1}{nsamp} \times sos\_i}$$

Calculation of Real Power

$$P[W] = np \times 1_{i,fast\,pulse\,rate}$$

$$1_{i,fast\,pulse\,rate} = \frac{1{,}000 \times 3{,}600}{204{,}800}[Ws]$$

Calculation of Reactive Power

$$Q[VAr] = \sqrt{S^2 - P^2}$$

Calculation of cos(φ)

$$\cos(\varphi) = \frac{P}{S}$$

Settings Register (SREG)

The settings register contains data stored by the MCU, which are used, for example, for calibration purposes, but also for general settings like input gain.

Register Name	Address	Reset Value	Description
pcorr_i1[7:0]	9320h	00h	Sets the phase correction for current channel i1.
pcorr_i1[8]	9321h	00h	
pcorr_i2[7:0]	9322h	00h	Sets the phase correction for current channel i2.
pcorr_i2[8]	9323h	00h	
cal_v[7:0]	9324h	00h	Calibration factor for voltage channel. Only acts on sos_v data.
cal_v[15:8]	9325h	80h	
cal_i1[7:0]	9326h	00h	Calibration factor for current channel i1. Only acts on sos_i1 data.
cal_i1[15:8]	9327h	80h	
cal_i2[7:0]	9328h	00h	Calibration factor for current channel i2. Only acts on sos_i2 data.
cal_i2[15:8]	9329h	80h	
pulselev_i1[7:0]	932Ah	46h	Pulse_lev for fast pulse generation if current channel i1 is selected (sel_i).
pulselev_i1[15:8]	932Bh	B6h	
pulselev_i1[23:16]	932Ch	08h	
pulselev_i2[7:0]	932Dh	46h	Pulse_lev for fast pulse generation if current channel i2 is selected (sel_i).
pulselev_i2[15:8]	932Eh	B6h	
pulselev_i2[23:16]	932Fh	08h	
mconst[3:0]	9330h	06h	Meter constant for LED pulse generation
-	9331h	-	Not used
nsamp[7:0]	9332h	ACh	Sets number of samples before next update of MDR.
nsamp[15:8]	9333h	0Dh	
vconst[7:0]	9334h	3Dh	A predefined voltage value which may be used for energy calculation in the event of V_{mains} not being available.
vconst[13:8]	9335h	0Bh	
Select	9336h	80h	Select register
Gains	9337h	03h	Gain settings register
Status	9338h	00h	Status register

Note: Unused addresses will simply be ignored. Unspecified bits will also be ignored.

Select Register (Select, 9336h)

MSB							LSB
ledpol	-	-	sel_p	sel_i	sel_v	sel_hp	sel_equ

Bit	Symbol	Function
7	ledpol	Selects polarity of LED pulses: 0: negative going pulses 1: positive going pulses (default)
6	-	Not used
5	-	Not used
4	sel_p	Select between instantaneous and real power for pulse generation 0: instantaneous power 1: real power (low-pass filtered instantaneous power)
3	sel_i	Select current channel for power calculation (Fast Pulse Gen) 0: i1 1: i2
2	sel_v	Select voltage channel data 0: selects voltage channel analog input 1: selects the predefined constant 'vconst'
1	sel_hp	Select high-pass filter 0: high-pass 1: no high-pass
0	sel_equ	Select equalisation filter 0: equalizer 1: no equalizer

Gain Settings Register (Gains, 9337h)

MSB							LSB
-	-	-	-	gain_i2[1]	gain_i2[0]	gain_i1[1]	gain_i1[0]

Bit	Symbol	Function			
7	-	Not used			
6	-	Not used			
5	-	Not used			
4	-	Not used			
3	gain_i2[1]	Gain setting for current channel 2 modulator	Bit1	Bit0	Gain
			0	0	4
			0	1	16
2	gain_i2[0]		1	0	16
			1	1	20
1	gain_i1[1]	Gain setting for current channel 1 modulator	Bit1	Bit0	Gain
			0	0	4
			0	1	16
0	gain_i1[0]		1	0	16
			1	1	20

Status Register (Status, 9338h)

MSB							LSB
creep	mdm	i_lead	diro	pddeton	alarm	dai	ack

Bit	Symbol	Function
7	creep	Indicator for creep situation, used as disable signal for LED pulse generation 0: no creep 1: creep
6	mdm	MCU Debug Mode flag Enables the MDR to be written by the MCU. This is useful for debugging when the programmer wants to know exactly what is received from the DSP block. 0: normal mode 1: debug mode as described later in the data sheet.
5	i_lead	Indicates if the mains current leads or lags the mains voltage. 0: mains current lags (inductive) 1: mains current leads (capacitive)
4	diro	DIRO indicator, signals when voltage and current are out of phase by 180° 0: 0° phase difference 1: 180° phase difference Can only be read by MCU.
3	pddeton	Enables the power-down detector functionality 0: no PD_DET functionality 1: PD_DET on
2	alarm	Indicates when the V_{mains} is falling below a predefined threshold. If this happens an interrupt is generated and the alarm flag is set. The interrupt will be reset only when the alarm flag is reset. 0: no alarm 1: alarm that V_{mains} is low
1	dai	Data Available Interrupt flag Indicates that an interrupt has been generated because new meter data are available. 0: no interrupt 1: interrupt due to new data Set only by DSP. Resetable only by software (MCU). A clear of 'dai' means that the irq is set back to 0.
0	ack	Acknowledge bit, indicates if MCU has transferred newly available data to its memory 1: Set by DSP, when data are ready on MDR. (not settable by MCU!) 0: Reset by MCU, when data have been taken. When ack gets reset the contents of MDR-np is set to zero. The 'P_ACCU' always adds the contents of MDR-np to the last value just before it transfers new data to the MDR. Thus, if ack=0 the MDR-np is reset and nothing is added to the P_ACCU. If ack has not been cleared the np data is still available and is added to the P_ACCU.

Current Channel Comparison

The two current channels can be compared by the microcontroller (MCU), if the greater of the two currents is required for energy calculation. This is done by comparing the calculated RMS values of the two currents. The threshold for changing from I1 to I2 (or visa versa) can also be set in the MCU software.

Creep Detection

The standards specify that no pulses must be generated when there is no current flow ('creep'). Additionally there is a threshold for current when the meter must generate pulses in any case ('starting current'). Therefore a detection circuit must guarantee that these two situations are under control.
The AS8267 / AS8268 current channel data are evaluated in the MCU to find out if there is a 'creep' situation. The related signal is used to stop the pulse generation if required.

MCU Debug Mode

When mdm flag of SREG/Status register is set, the DSP block enters the MCU debug mode. Here the MDR can be written through the MCU interface. In this mode the DSP block is not allowed to write to the MDR.
Special functionality:
1) ack set to 0 → np is set to 0 (i.e. must be set again by MCU)
2) when ack is not reset by the MCU the np value is doubled, i.e. a shift left is done.

Note: Also in debug mode an interrupt is generated after nsamp samples.

Power Supply Monitor (PSM)

The AS8267 / AS8268 ICs have an on-chip power supply monitor (PSM) that ensures a reset is generated independently of the supply voltage (VDD) rise and fall times.

A built in hysteresis is provided to accommodate slow changes on the VDD, to ensure clean signal switching.

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Threshold positive edge	$V_{th,pos}$	2.6		2.9	V	
Threshold negative edge	$V_{th,neg}$	2.2		2.8	V	
Hysteresis	Hyst	100			mV	

Table 1: Power supply monitor: Power-on reset specifications

To ensure sufficient time is available to store the meter data in an EEPROM during power-down, it is necessary to detect the falling supply voltage as fast as possible. Should only the VDD be monitored, an external capacitor in the 3.3V power supply could sustain the VDD supply voltage even after the V_{mains} has begun to fall. For this reason, the AS8267 / AS8268 ICs allow the monitoring of the V_{mains} to ensure early power-down detection. The power-down detector function (PD_DET) is enabled in the SREG/Status register.

An alarm signal is generated, when the V_{mains} falls below a specified mains voltage threshold, which enables the MCU to react with sufficient time. It is also possible to calculate energy during power-down detection, taking a constant voltage value for calculation of the energy value.

The mains voltage threshold is calculated as follows:

$$V_{mains}(\text{alarm}) = \frac{V_{mains} \times 512}{\text{RMS\_V(ideal)} \times \sqrt{2}}$$

$$= \frac{230 \times 512}{479 \times \sqrt{2}}$$

$$= 173.8 \text{ VAC}$$

External System Reset (RES_N)

An external reset pin (RES_N) is provided for system reset. RES_N is active LOW (i.e. logic ‘0’ will initiate a system reset). A system reset via the RES_N pin is OR-ed with the main system power-on reset generated by the power supply monitor PSM.
RES_N is internally pulled high.

System Timing and Real Time Clock (RTC)

A low power crystal oscillator using a 3.0 to 4.0MHz crystal provides the AS8267 / AS8268 system timing. The low power oscillator is internally connected to a low power divider, which provides a 1Hz signal to the real time clock, which may be trimmed. The RTC circuit may be battery powered to continue operating even when V_{mains} is interrupted.

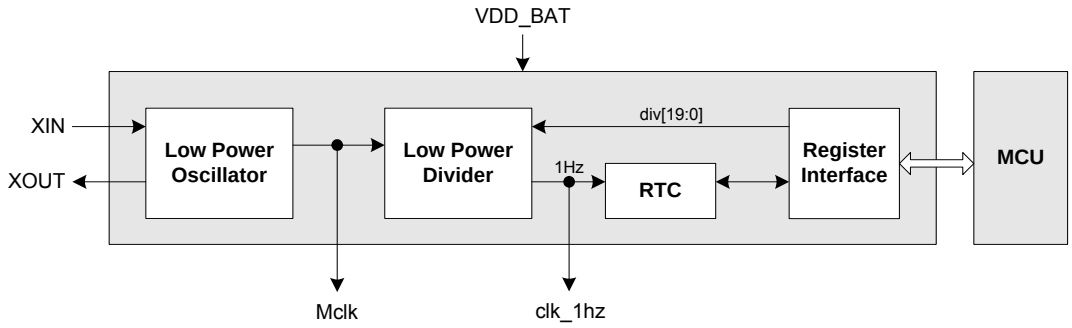


Figure 5: System timing and RTC block diagram

Low Power Oscillator (LP_OSC)

The low power oscillator is connected to an external 3.0 to 4.0MHz crystal. The oscillator can be operated in normal mode or low power mode.

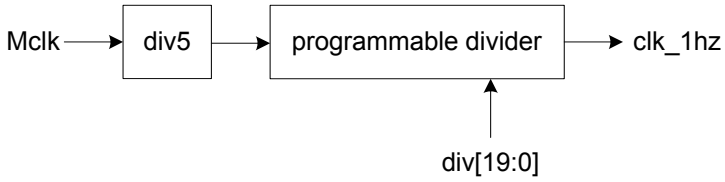
Should a suitable external clock signal be preferred, this may be directly connected to the XIN pin, which is fed through to output 'Mclk'. In this case, XOUT is left unconnected.

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Current consumption, normal mode	$I_{osc,norm}$		20		µA	VDD supply
Current consumption, low power mode	$I_{osc,bat}$		7		µA	VDD_BAT = 2VDC @ 25°C
Frequency range	f_{osc}	3.0	3.579545	4.0	MHz	
Supply voltage range	VDD_BAT	2.0	3.3	3.6	V	
Duty cycle	duty_cyc	45		55	%	

Low Power Divider (LP_DIV, 9130h – 9132h)

The main oscillator output frequency (Mclk) is divided down to 1Hz for the real time clock (RTC). The option to use alternative crystal frequencies and still derive a 1Hz clock signal for the real time clock (RTC), is provided through this internally programmable divider.

For power-saving reasons, the fast oscillator clock is first divided down by a fixed ratio (divide by 5) and then the programmable divider follows.



Parameter	Symbol	Min	Typ	Max	Unit	Notes
Input frequency range	f_{Mclk}	3.0	3.579545	4.0	MHz	
Division factor	n			1,048,575		1)

The setting of div[19:0] is located in the RTC registers (addresses: 9132h – 9130h)

Note:

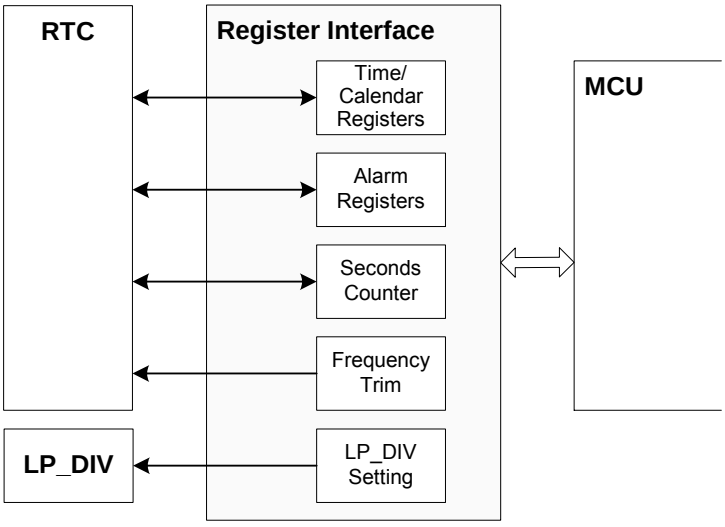
1) The division factor n is effective on the frequency Mclk/5. It represents the actual division factor minus 1.
   Example: Calculate n for oscillator frequency 3,579,545Hz. The frequency after the div5 is 715,909Hz.
   Therefore, n must be 715,909 – 1 = 715,908 to get 1Hz.
2) Setting n = 1 means a division factor of 2.

Real-Time Clock (RTC)

The RTC can be directly accessed from the MCU via a dedicated interface register.

Two alarm registers are provided to indicate a certain time instance, such as the start of a new month. In that case an interrupt is sent to the MCU. Constant frequency deviations of the crystal that is used can be trimmed to an accuracy of better than +/-1.4ppm.

A seconds counter is provided which may be used for certain meter calculations. There is only one interrupt output. The source of the interrupt is indicated in the Control/Status 2 register.



RTC Registers

Register Name	Address	Reset Value	Notes
Seconds / VL	9100h	80h	
Minutes	9101h	00h	
Hours	9102h	00h	
Days	9103h	01h	
Day of the Week	9104h	00h	
Month / Century	9105h	01h	
Years	9106h	00h	
Control / Status 1	9110h	10h	
Control / Status 2	9111h	00h	
Seconds Timer Byte 0	9112h	01h	
Seconds Timer Byte 1	9113h	00h	
Minute Alarm 1	9114h	00h	
Hour Alarm 1	9115h	00h	
Day Alarm 1	9116h	01h	
Month Alarm 1	9117h	01h	
Years Alarm 1	9118h	00h	
Minute Alarm 2	9119h	00h	
Hour Alarm 2	911Ah	00h	
Day Alarm 2	911Bh	01h	
Month Alarm 2	911Ch	01h	
Years Alarm 2	911Dh	00h	
Divider Register Byte 0	9130h	84h	[7:0] = div [7:0] (LP_DIV)
Divider Register Byte 1	9131h	ECh	[7:0] = div [15:8] (LP_DIV)
Divider Register Byte 2	9132h	0Ah	[3:0] = div [19:16] (LP_DIV)
Frequency Trim	9133h	00h	

Notes:

1) If illegal values (i.e. not defined in the following tables, e.g. '0', no BCD code, not correct last day of month, not correct leap year) are written to the time/date registers (9000h – 9006h), they are corrected to the first valid number ('automatic correction')! Then an interrupt is generated and the TSA flag in the Control/Status 2 register is set.
2) All other registers are not corrected (e.g. alarm info incorrect → alarm is not met).
3) After power-up of VDD_BAT the time/date registers are stopped, the WAIT flag (Control/Status 1) is set.
4) Unused addresses will simply be ignored.

Control / Status 1 Register (9110h)

MSB							LSB
-	-	-	WAIT	-	-	-	-

Bit	Symbol	Function
7	-	Not used
6	-	Not used
5	-	Not used
4	WAIT	Indicates that RTC is waiting for a start signal. The start signal is WAIT being reset to 0. WAIT = 0 ➔ RTC running normally. (Clear by MCU.) WAIT = 1 ➔ Is set when time/calendar information is changed (access to registers 9100h to 9106h). While RTC is waiting for a start signal, 1Hz clock is still gated to the MPIOs.
3	-	Not used
2	-	Not used
1	-	Not used
0	-	Not used

Register bit assignment: (Unassigned bits in the registers are marked with '-'. If these bits are read they will return zero value. Writing these bits has no effect.)

Control / Status 2 Register (9111h)

MSB							LSB
TSA	-	A2F	A1F	STF	AIE2	AIE1	SIE

Bit	Symbol	Function
7	TSA	Time Setting Alarm: Indicates when an impossible time/date has been set and it has been corrected by the RTC automatically, e.g. 31st February → 1st February. An interrupt will be generated and TSA is set to 1. The interrupt is cleared by setting TSA=0 (done by MCU (software)).
6	-	Not used
5	A2F	Set to logic 1 when an alarm 2 occurs and maintains this value until software clears it. Indicates the source of the interrupt. Cannot be set by software. When the flag is cleared, also the interrupt is cleared.
4	A1F	Set to logic 1 when an alarm 1 occurs and maintains this value until software clears it. Indicates the source of the interrupt. Cannot be set by software. When the flag is cleared, also the interrupt is cleared.
3	STF	Set to logic 1 when a seconds timer interrupt occurs and maintains this value until software clears it. Indicates the source of the interrupt. Cannot be set by software. When the flag is cleared, also the interrupt is cleared.
2	AIE2	AIE2 = 0; alarm 2 interrupt disabled AIE2 = 1; alarm 2 interrupt enabled
1	AIE1	AIE1 = 0; alarm 1 interrupt disabled AIE1 = 1; alarm 1 interrupt enabled
0	SIE	SIE = 0; seconds counter interrupt disabled SIE = 1; seconds counter interrupt enabled

Note: Alarm interrupts are only generated on rising clk_1hz edges (using system clock for detection). This means that enabling an alarm after that will not generate an interrupt.

Seconds / VL Register (9100h)

MSB							LSB
VL	sec.6	sec.5	sec.4	sec.3	sec.2	sec.1	sec.0

Bit	Symbol	Function
7	VL	VL = 0; reliable clock / calendar information guaranteed. VL = 1; clock / calendar information is NOT guaranteed. This bit is set after power-up of VDD_BAT. It can be cleared by software only.
6	sec.6	These bits represent current seconds value encoded in BCD format (values from 0 to 59).
5	sec.5	
4	sec.4	
3	sec.3	
2	sec.2	
1	sec.1	
0	sec.0	

Minutes Register (9101h)

MSB							LSB
0	min.6	min.5	min.4	min.3	min.2	min.1	min.0

These bits represent current minute value encoded in BCD format (values from 0 to 59).

Hours Register (9102h)

MSB							LSB
0	0	hour.5	hour.4	hour.3	hour.2	hour.1	hour.0

These bits represent the current hours value encoded in BCD format (values from 0 to 23).

Days Register (9103h)

MSB							LSB
0	0	day.5	day.4	day.3	day.2	day.1	day.0

These bits represent current day value encoded in BCD format (values from 1 to 31).

Note on leap years: ‘00’ years in general are no leap years unless the complete year can be divided by 400 (e.g. 2000). Since the year 2000 has passed already, this chip will not consider a leap year for ‘00’ years.

Day of the Week Register (9104h)

MSB							LSB
0	0	0	0	0	weekd.2	weekd.1	weekd.0

Bit	Symbol	Function
7	-	Not used
6	-	Not used
5	-	Not used
4	-	Not used
3	-	Not used
2	weekd.2	These bits represent the current weekday value.
1	weekd.1	
0	weekd.0	

Bit2	Bit1	Bit0	Day
0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

Month / Century Register (9105h)

MSB							LSB
C	0	0	month.4	month.3	month.2	month.1	month.0

Bit	Symbol	Function
7	C	Century bit. C = 0; indicates the year is 20xx C = 1; indicates the year is 21xx 'xx' indicates the value held in Years register. This bit is modified when Years register overflows from 99 to 00.
6	-	Not used
5	-	Not used
4	month.4	These bits represent the current month value encoded in BCD format.
3	month.3	

Bit4	Bit3	Bit2	Bit1	Bit0	Month
0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May

Bit	Symbol	Function						
2	month.2		0	0	1	1	0	June
			0	0	1	1	1	July
			0	1	0	0	0	August
1	month.1		0	1	0	0	1	September
			1	0	0	0	0	October
0	month.0		1	0	0	0	1	November
			1	0	0	1	0	December

Year Register (9106h)

MSB							LSB
year.7	year.6	year.5	year.4	year.3	year.2	year.1	year.0

These bits represent current year value encoded in BCD format (value from 0 to 99).

The Alarm 1 or 2 is generated when the programmed time has been reached (seconds = 0!).

Minute Alarm Register (1/2) (9114h/9119h)

MSB							LSB
0	mina.6	mina.5	mina.4	mina.3	mina.2	mina.1	mina.0

These bits represent minute alarm information encoded in BCD format (values from 0 to 59).

Hour Alarm Register (1/2) (9115h/911Ah)

MSB							LSB
0	0	houra.5	houra.4	houra.3	houra.2	houra.1	houra.0

These bits represent hour alarm information encoded in BCD format (values from 0 to 23).

Day Alarm Register (1/2) (9116h/911Bh)

MSB							LSB
0	0	daya.5	daya.4	daya.3	daya.2	daya.1	daya.0

These bits represent day alarm information encoded in BCD format (values from 1 to 31).

Month / Century Alarm Register (1/2) (9117h/911Ch)

MSB							LSB
C	0	0	mona.4	mona.3	mona.2	mona.1	mona.0

These bits represent current month alarm value encoded in BCD format (value from 1 to 12). Please see also the ‘month assignments’ table above.

Year Alarm Register (1/2) (9118h/911Dh)

MSB							LSB
yeara.7	yeara.6	yeara.5	yeara.4	yeara.3	yeara.2	yeara.1	yeara.0

These bits represent the year alarm value encoded in BCD format (value from 0 to 99).

Setting the Time

The time can be set by writing to the respective time and calendar registers. When this is done the clock stops, the WAIT bit is set and the control waits for the WAIT bit to be reset.
When the WAIT bit is reset the clock gate will be opened and the RTC starts running.

Alarms

When time and one of the alarm registers match (seconds = 0), an interrupt is generated. The source of the interrupt is indicated in the A[1|2]F register bits in the Control/Status 2 register.
The alarm generation can be disabled using the AIE1/2 bits.
When the rest of the chip is off, there is no clock for the MCU interface, hence no alarm will be generated.
The MCU interface is reset with the ‘res’ signal which is coming from the PSM, i.e. all Status 2 bits are reset to default, which means that after MCU power-up it has to set the appropriate alarms again. (After power-up the MCU has to check what the time is, and has to decide what the next appropriate alarms will be.)

Seconds Timer (9112h, 9113h)

The seconds counter block, if enabled (SIE bit of Control/Status 2 register), generates an interrupt every n seconds. ‘n’ is the number of seconds specified in the Seconds Timer registers 9112h (Byte 0) and 9113h (Byte 1). When an interrupt is sent, the flag STF is set.
The Seconds Timer register is **not BCD** coded.

Seconds counter start value:	0000h
Seconds counter count direction:	up
Condition for interrupt generation:	Seconds counter register value = Seconds Timer register value

Note: 0000h in the timer register means that no interrupt must be generated.

RTC Calibration (clk_1Hz)

When using the real-time clock (RTC) it is essential that the 1Hz signal to the real-time clock is accurate. There are many possible external influences on the crystal oscillator frequency including the absolute crystal frequency itself and the parasitic and oscillator capacitor values.

These influences alone can contribute to a significant change in the oscillator frequency. In this case, it is necessary to perform a calibration of the 1Hz signal through the ‘Programmable Divider’ located in the 'Low Power Divider'.

The procedure for trimming the RTC via the 'Programmable Divider' is explained below:

Assuming a crystal frequency of 3.579545 MHz

The Programmable Divider follows a fixed 'Divide by 5' divider, thus the default value to the Programmable Divider is:
3.579545 / 5 = 715909 (default value to Programmable Divider)
Therefore: A change of 1Hz in this default value is equal to:
1 / 715909 = 1.397 ppm
Measure the deviation in the clk_1Hz frequency output provided by the AS8267 / AS8268 ICs
Assuming an error of +690 ppm is measured (faster than real-time)
Thus +690 / 1.397 = 493.915 ≅ 494
Therefore 494 must be added to the default value:
715908 + 494 = 716402 (dec) = 0A EE 72 (hex)
Divider Register Byte 2 = 0A
Divider Register Byte 1 = EE
Divider Register Byte 0 = 72
The RTC is then calibrated to within +/- 1.4 ppm

Frequency Trimming (9133h)

A further option for clk_1Hz frequency trimming is available. In this case only the 5 lower bits of register 'Frequency Trim' (9133h) are used as defined in the following table.

FREQ_TRIM[4:0]					Correction [ppm]	Seconds per Day 1	Seconds per Day 2
0	1	1	1	1	87.0	7	8
0	1	1	1	0	81.2	7	7
0	1	1	0	1	75.4	6	7
0	1	1	0	0	69.6	6	6
0	1	0	1	1	63.8	5	6
0	1	0	1	0	58.0	5	5
0	1	0	0	1	52.2	4	5
0	1	0	0	0	46.4	4	4
0	0	1	1	1	40.6	3	4
0	0	1	1	0	34.8	3	3
0	0	1	0	1	29.0	2	3
0	0	1	0	0	23.2	2	2
0	0	0	1	1	17.4	1	2
0	0	0	1	0	11.6	1	1
0	0	0	0	1	5.8	0	1
0	0	0	0	0	0	0	0
1	1	1	1	1	-5.8	0	-1
1	1	1	1	0	-11.6	-1	-1
1	1	1	0	1	-17.4	-1	-2
1	1	1	0	0	-23.2	-2	-2
1	1	0	1	1	-29.0	-2	-3

FREQ_TRIM[4:0]					Correction [ppm]	Seconds per Day 1	Seconds per Day 2
1	1	0	1	0	-34.8	-3	-3
1	1	0	0	1	-40.6	-3	-4
1	1	0	0	0	-46.4	-4	-4
1	0	1	1	1	-52.2	-4	-5
1	0	1	1	0	-58.0	-5	-5
1	0	1	0	1	-63.8	-5	-6
1	0	1	0	0	-69.6	-6	-6
1	0	0	1	1	-75.4	-6	-7
1	0	0	1	0	-81.2	-7	-7
1	0	0	0	1	-87.0	-7	-8
1	0	0	0	0	-92.8	-8	-8

The table specifies 2 successive days with (possibly) a different number of seconds that have to be added or subtracted per day. ‘day1/day2’ are repeated continuously.

The RTC is always adjusted at the same time: **00:00 a.m. and 30 seconds**. (The 30 seconds is required to avoid conflicts with alarm settings, which are defined to occur at 0 seconds.)

Subtraction means that the specified number of 1Hz pulses is ignored. This has the effect that the clock stands still for the specified number of seconds.

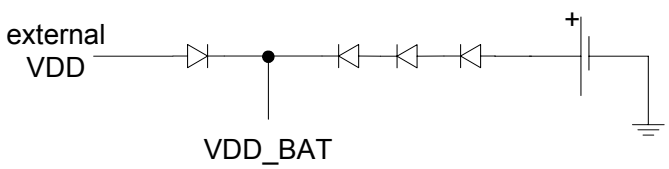
Example: A crystal has a frequency that is 30ppm higher than specified. Therefore the RTC will run faster. Thus, the RTC has to correct in the negative direction, by subtracting seconds. Value ‘11011’ (-29.0) will be chosen which means that on day1, 2 seconds are subtracted, then on the next day 3 seconds are subtracted, then 2 seconds again and so on.

Battery Backup Operation

The AS8267 / AS8268 ICs contain a real-time clock (RTC) circuit, which must continue to operate even when the mains supply voltage (V_{mains}) is interrupted. A battery backup facility is provided for this purpose at pin VDD_BAT.

The low power oscillator (LP_OSC), low power divider (LP_DIV) and the real time clock (RTC) are all supplied from the VDD_BAT pin.

The recommended battery backup circuit is shown below. The battery is connected to the VDD_BAT pin via one or two diodes. The external VDD is also connected to the VDD_BAT pin via a diode, with the battery backup only providing supply to the AS8267 / AS8268 ICs when the external VDD is interrupted.



8.2 Temperature Sensor

The AS8267 / AS8268 ICs include an on-chip temperature sensor which allows for temperature correction over the entire operating temperature range of the device:

Parameter	Symbol	Min	Typ	Max	Unit	Note
Absolute Error (trimmed)		-5		+5	°C	from -40°C to 85°C
Relative Error (trimmed)		-2		+2	°C	from -40°C to 85°C
Temperature Range		-40		85	°C	
Resolution			0.193		°C/LSB	

Note:
The temperature sensor is activated by the MCU for a single measurement.

A temperature measurement is initiated by setting the measure bit in the TS_Status register to 1. The result of the measurement is then available in the TS_Result0/TS_Result1 register.
The registers TS_OffsetCorr0/TS_OffsetCorr1 hold the offset correction value which is derived during production process.
The actual temperature value is calculated by the means of the following formula:

$$\mathrm{Temp}[°\mathrm{C}] = (\mathrm{Temp\_corrected} \times 0.193) - 75$$

with $\mathrm{Temp\_corrected} = \mathrm{TS\_Result}[15:0] + \mathrm{TS\_OffsetCorr}[15:0]$

Register Name	Address	Reset Value	Note
TS_Status	9600h	00h	
TS_Result0	9601h	00h	
TS_Result1	9602h	00h	
TS_OffsetCorr0	9603h	00h	
TS_OffsetCorr1	9604h	00h	

TS_Status (9600h)

MSB							LSB
0	0	0	0	0	0	0	Measure

Note:
MEASURE: - flag is set by the MCU and reset by the temperature sensor itself.
0: no operation; or: temperature measurement finished
1: if set by MCU the measurement starts; or: indicates an ongoing measurement

TS_Result0/TS_Result1 (9601h/9602h)

TS_Result0

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

TS_Result1

MSB							LSB
0	0	0	0	0	0	D9	D8

TS_OffsetCorr0/TS_OffsetCorr1 (9603h/9604h)

Signed offset correction value [15:0]; read only.

TS_OffsetCorr0

MSB							LSB
OC7	OC6	OC5	OC4	OC3	OC2	OC1	OC0

TS_OffsetCorr1

MSB							LSB
OC15	OC14	OC13	OC12	OC11	OC10	OC9	OC8

Example for temperature calculation:

$$Temp[°C] = (Temp\_corrected \times 0.193) - 75$$

with $Temp\_corrected = TS\_Result[15:0] + TS\_OffsetCorr[15:0]$

TS_Result[15:0] = 00A2h

TS_OffsetCorr[15:0] = 0101h

Temp_corrected = 01A3h = 419d

Temp[°C] = (419 × 0.193) – 75 = 5.9[°C]

8.3 LCD Driver (LCDD)

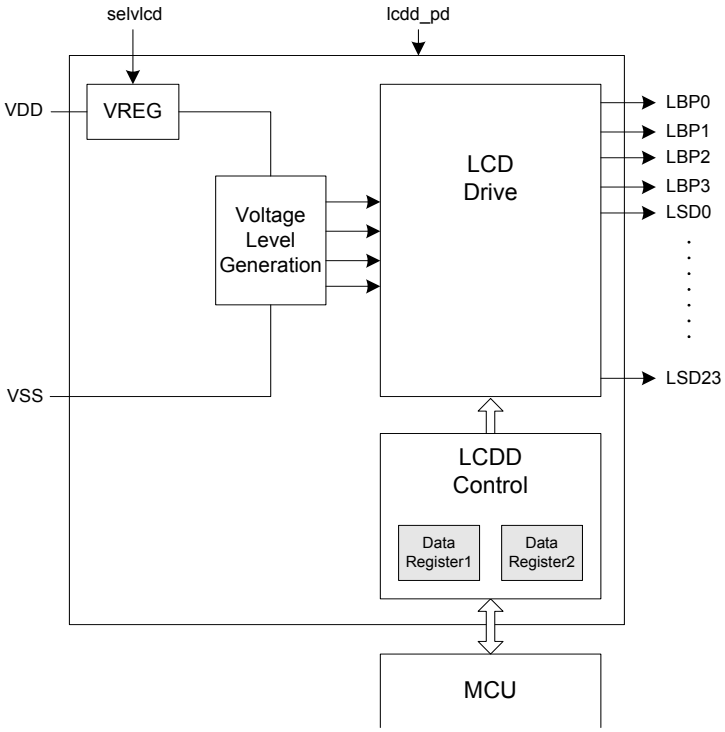


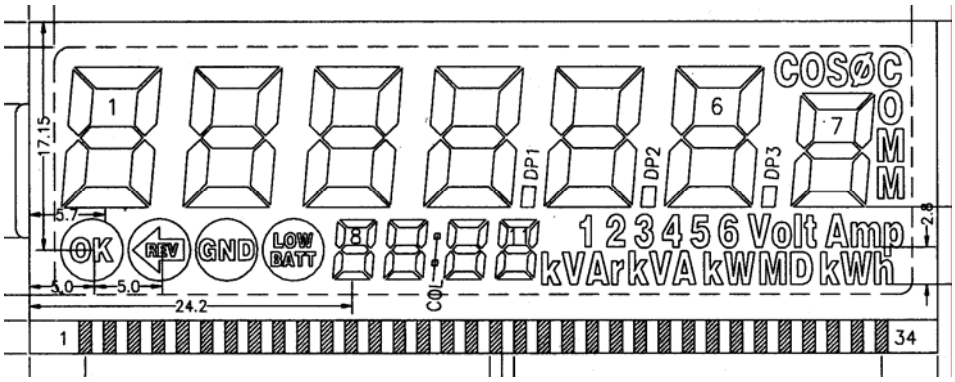
Figure 6: LCD driver block diagram

The on-chip LCD driver (LCDD) is a peripheral block, which interfaces to almost any liquid crystal display (LCD) having a multiplex rate of 4. It generates the drive signals to directly drive multiplexed LCDs containing up to four backplanes and up to 24 segments per backplane. The AS8267 has a 20 x 4 LCDD, while the AS8268 has a 24 x 4 LCDD.

The data registers receive and store the display information, which is to be sent to the display. The LCDD control block decodes the information into the select lines for the single segments using a specific timing.

The LCD voltage can be selected to adjust the contrast of the display, as required. The selvlcd[2:0] register bits enable the setting of the LCD contrast by selecting one of the defined LCD voltage levels. The contrast can be improved with a higher voltage, however the contrast is also dependent upon the crystal frequency.
With the lcdd_pd bit the LCDD analog part can be switched off.

Typical Display



The LCD above is a typical example of those used in electricity meter applications and consists of a number of digits (generally up to 8 digits) including decimal points. Typically, annunciators ('kWh', 'Volt', etc.) are also included to signify the type of data on display.

LCD Drive (LCD_DRIVE)

- LCD drive mode is 1/4duty, 1/3bias.
- 4 back planes
- 24 segment drives (maximum)

All other parameters are listed in the table below:

Parameter	Symbol	Min	Typ	Max	Unit	Notes
LCD frame frequency	f_{LCD}	33	39.4	44	Hz	1)
LCD voltage	V_{LCD}	2.3	2.5	2.75	V	for selvlcd='000'
LCD segment and back plane drive voltages	V_3	0.95 x V_{LCD}	V_{LCD}	1.05 x V_{LCD}	V	
	V_2	0.95 x 2/3V_{LCD}	2/3V_{LCD}	1.05 x 2/3V_{LCD}	V	
	V_1	0.95 x 1/3V_{LCD}	1/3V_{LCD}	1.05 x 1/3V_{LCD}	V	
	V_0		VSS			
LCD DC component	V_{DCLCD}	-20	0	20	mV	
LCD drive impedance	R_{LCD}			100	kΩ	
LCD load on each driver pin	C_{load}			300	pF	

Note:

1) These frequencies are derived from the master clock (3MHz; 3.58MHz; 4MHz) using a divider of 90,909.

LCDD Control (LCDD_CTRL) including Input and Config Registers

In the control block of the LCD driver there are two registers. Each of these registers may contain data to be displayed. With a special bit (921Eh, bit 0), it is possible to select one of the two register banks for display. Each register defines the settings for the different segment and plane select lines. The following table specifies the allocation of the register bits:

	LSD0	LSD1	LSD2	LSD3	LSD4	LSD5	LSD6	LSD7	LSD8	LSD9
LBP0	reg[0]	reg[4]	reg[8]	reg[12]	reg[16]	reg[20]	reg[24]	reg[28]	reg[32]	reg[36]
LBP1	reg[1]	reg[5]	reg[9]	reg[13]	reg[17]	reg[21]	reg[25]	reg[29]	reg[33]	reg[37]
LBP2	reg[2]	reg[6]	reg[10]	reg[14]	reg[18]	reg[22]	reg[26]	reg[30]	reg[34]	reg[38]
LBP3	reg[3]	reg[7]	reg[11]	reg[15]	reg[19]	reg[23]	reg[27]	reg[31]	reg[35]	reg[39]

	LSD10	LSD11	LSD12	LSD13	LSD14	LSD15	LSD16	LSD17	LSD18	LSD19
LBP0	reg[40]	reg[44]	reg[48]	reg[52]	reg[56]	reg[60]	reg[64]	reg[68]	reg[72]	reg[76]
LBP1	reg[41]	reg[45]	reg[49]	reg[53]	reg[57]	reg[61]	reg[65]	reg[69]	reg[73]	reg[77]
LBP2	reg[42]	reg[46]	reg[50]	reg[54]	reg[58]	reg[62]	reg[66]	reg[70]	reg[74]	reg[78]
LBP3	reg[43]	reg[47]	reg[51]	reg[55]	reg[59]	reg[63]	reg[67]	reg[71]	reg[75]	reg[79]

	LSD20	LSD21	LSD22	LSD23
LBP0	reg[80]	reg[84]	reg[88]	reg[92]
LBP1	reg[81]	reg[85]	reg[89]	reg[93]
LBP2	reg[82]	reg[86]	reg[90]	reg[94]
LBP3	reg[83]	reg[87]	reg[91]	reg[95]
	AS8268 only			

Notes:

1) Each of the register bits represents one of the segments of the digits or a decimal point or one of the annunciators.
2) reg[x]=0: Segment is turned off; reg[x]=1: Segment is turned on.

The complete register is organized in bytes according to following table:

Register Name	Address	Reset Value	Description
reg1[7:0]	9200h	00h	
reg1[15:8]	9201h	00h	
reg1[23:16]	9202h	00h	
reg1[31:24]	9203h	00h	
reg1[39:32]	9204h	00h	
reg1[47:40]	9205h	00h	
reg1[55:48]	9206h	00h	
reg1[63:56]	9207h	00h	
reg1[71:64]	9208h	00h	
reg1[79:72]	9209h	00h	
reg1[87:80]	920Ah	00h	AS8268 only
reg1[95:88]	920Bh	00h	
reg2[7:0]	9210h	00h	
reg2[15:8]	9211h	00h	
reg2[23:16]	9212h	00h	
reg2[31:24]	9213h	00h	
reg2[39:32]	9214h	00h	
reg2[47:40]	9215h	00h	
reg2[55:48]	9216h	00h	
reg2[63:56]	9217h	00h	
reg2[71:64]	9218h	00h	
reg2[79:72]	9219h	00h	
reg2[87:80]	921Ah	00h	AS8268 only
reg2[95:88]	921Bh	00h	
use_reg	921Eh	00h	Bit 0: Selects register to be used. 0: Data Register 1 1: Data Register 2
selvlcd[2:0]	921Fh	00h	Select V_{LCD} level. See table in LCD Voltage Select Register.
lcdd_pd	9220h	01h	Bit 0: Power-down of the LCDD analog part. 0: Display on 1: Display off

Notes:

1) Unused registers will simply be ignored.
2) All the registers are write only. Read operations always return 0.

LCD Display Data Select Register (USE_REG, 921Eh)

The use_reg register selects either Data Register 1 or Data Register 2 for display on the LCD. Select '0' for Data Register 1 and '1' for Data Register 2.

MSB							LSB
-	-	-	-	-	-	-	use_reg

LCD Voltage Select Register (SELVLCD, 921Fh)

The LCD voltage select register, SELVLCD enables variation of the LCD contrast by selecting on of the 8 pre-set voltage levels.

MSB							LSB
-	-	-	-	-	selvlcd.2	selvlcd.1	selvlcd.0

Bit	Symbol	Function	Bit2	Bit1	Bit0	VLCD
7	-	Not used				
6	-	Not used				
5	-	Not used				
4	-	Not used				
3	-	Not used				
2	selvlcd.2	These bits set the LCD voltage level for the LCD contrast setting.	0	0	0	2.5V
			0	0	1	2.5714V
1	selvlcd.1		0	1	0	2.6428V
			0	1	1	2.7142V
			1	0	0	2.7856V
0	selvlcd.0		1	0	1	2.8570V
			1	1	0	2.9284V
			1	1	1	3.0V

LCD Power-Down (LCDD_PD, 9220h)

The lcdd_pd register enables the analog part of the LCDD to be powered-down. Select '0' for LCD display on and '1' for LCD display off.

MSB							LSB
-	-	-	-	-	-	-	lcdd_pd

8.4 Programmable Multi-Purpose I/Os (MPIO)

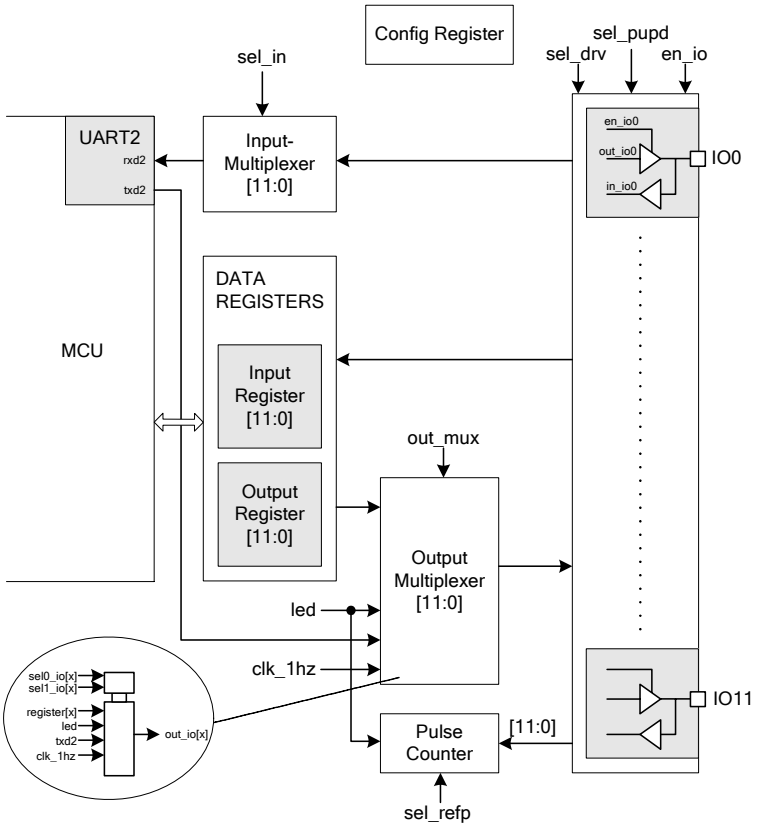


Figure 7: MPIO block diagram

A total of 9 bidirectional multi-purpose I/O pins (MPIO) are provided with the AS8267 and 12 bidirectional multi-purpose I/O pins with the AS8268, which may be used for a variety of purposes. All the I/Os can be freely programmed as inputs or outputs, with the option of either a pull-up or pull-down resistor. The drive strength of the individual I/O pins may also be programmed. On start-up all the I/O pins are disabled.
Furthermore, a pulse counter is available, which can be used for calibration purposes ('comparison calibration method': Between two LED pulses the pulses from a reference meter with much higher pulse rate are counted. The result is used to calculate the calibration factor.).

MPIO Registers

All the MPIO registers are listed in the table below. The individual register functions are then described in detail.

Register Name	Address	Reset Value	Notes
Config			
MAKE_IRQ0	9500h	00h	
MAKE_IRQ1	9501h	00h	
OUT_MUX0	9502h	00h	
OUT_MUX1	9503h	00h	

Register Name	Address	Reset Value	Notes
OUT_MUX2	9504h	00h	
SET_EN0	9505h	00h	
SET_EN1	9506h	00h	
SEL_DRV0	9507h	00h	
SEL_DRV1	9508h	00h	
SEL_PUPD0	9509h	00h	
SEL_PUPD1	950Ah	00h	
SEL_IN_RXD2	950Bh	04h	
SEL_IN_REFP	950Ch	03h	
Input			
IN0	950Dh	00h	
IN1	950Eh	00h	
Output			
OUT0	950Fh	00h	
OUT1	9510h	00h	
OUT2	9511h	00h	
OUT3	9512h	00h	
OUT4	9513h	00h	
OUT5	9514h	00h	
OUT6	9515h	00h	
OUT7	9516h	00h	
OUT8	9517h	00h	
OUT9	9518h	00h	AS8268 only
OUT10	9519h	00h	
OUT11	951Ah	00h	
Pulse counter			
PCNT0	951Bh	00h	
PCNT1	951Ch	00h	
PCNT2	951Dh	00h	
Status			
STATUS0	951Eh	00h	
STATUS1	951Fh	00h	

Note: Unused addresses are ignored.

MAKE_IRQ0/MAKE_IRQ1 (9500h/9501h)

The MAKE_IRQ registers specify if an interrupt should be generated after the related I/O input has changed. The I/O pin, which caused the interrupt, will be indicated in the STATUS0/STATUS1 flag registers.
IOx: 0: no interrupt on signal change 1: generate an interrupt on signal change

MAKE_IRQ0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

MAKE_IRQ1

MSB							LSB
0	0	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

OUT_MUX0/OUT_MUX1/OUT_MUX2 (9502h/9503h/9504h)

The OUT_MUX registers specify the source signal for each of the I/O outputs. Every 2 bits are used as select signals for the 4-way output multiplexer of the designated I/O.

OUT_MUX0

MSB							LSB
IO3: sel1	IO3: sel0	IO2: sel1	IO2: sel0	IO1: sel1	IO1: sel0	IO0: sel1	IO0: sel0

OUT_MUX1

MSB							LSB
IO7: sel1	IO7: sel0	IO6: sel1	IO6: sel0	IO5: sel1	IO5: sel0	IO4: sel1	IO4: sel0

OUT_MUX2

MSB							LSB
IO11: sel1	IO11: sel0	IO10: sel1	IO10: sel0	IO9: sel1	IO9: sel0	IO8: sel1	IO8: sel0
AS8268 only							

The following table shows the settings for the output signal options:

sel1	sel0	Output Signal	Notes
0	0	register[x]	
0	1	led	80ms pulse width
1	0	txd2	
1	1	clk_1hz	

SET_EN0/SET_EN1 (9505h/9506h)

The SET_EN registers set the en_io signal of the related I/O pin. The en_io enables the tri-state output buffer so that the I/O pins operate as outputs.
IOx: 0: disable output (I/O used as input) 1: enable output

SET_EN0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

SET_EN1

MSB							LSB
0	0	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

SEL_DRV0/SEL_DRV1 (9507h/9508h)

The SEL_DRV registers select the current drive strength for all the I/Os that have been selected as outputs:
IOx: 0: 4mA 1: 8mA

SEL_DRV0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

SEL_DRV1

MSB							LSB
0	0	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

SEL_PUPD0/SEL_PUPD1 (9509h/950Ah)

The SEL_PUPD registers select either a pull-up or pull-down resistor for each of the I/O pins:
IOx: 0: pull-down 1: pull-up

SEL_PUPD0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

SEL_PUPD1

MSB							LSB
0	0	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

SEL_IN_RXD2 (950Bh)

The SEL_IN_RXD2 register selects which I/O input is used for the special input signal ‘rxd2’ (UART2 receive input). Any one of the I/Os from IO0 to IO11 may be selected for this purpose.
The select bits are defined in the following table:

MSB							LSB
0	0	0	0	sel3	sel2	sel1	sel0

MSB							LSB	Input
0	0	0	0	0	0	0	0	**IO0**
0	0	0	0	0	0	0	1	**IO1**
0	0	0	0	0	0	1	0	**IO2**
0	0	0	0	0	0	1	1	**IO3**
0	0	0	0	0	1	0	0	**IO4**
0	0	0	0	0	1	0	1	**IO5**
0	0	0	0	0	1	1	0	**IO6**
0	0	0	0	0	1	1	1	**IO7**
0	0	0	0	1	0	0	0	**IO8**
0	0	0	0	1	0	0	1	**IO9**
0	0	0	0	1	0	1	0	**IO10**
0	0	0	0	1	0	1	1	**IO11**
AS8268 only								

SEL_IN_REFP (950Ch)

The SEL_IN_REFP register selects which I/O input is to be used for reference pulses. Any one of the I/Os from IO0 to IO11 may be selected for this purpose.
The select bits are defined in the following table:

MSB							LSB
0	0	0	0	sel3	sel2	sel1	sel0

MSB							LSB	Input
0	0	0	0	0	0	0	0	**IO0**
0	0	0	0	0	0	0	1	**IO1**
0	0	0	0	0	0	1	0	**IO2**
0	0	0	0	0	0	1	1	**IO3**
0	0	0	0	0	1	0	0	**IO4**
0	0	0	0	0	1	0	1	**IO5**
0	0	0	0	0	1	1	0	**IO6**

MSB							LSB	Input
0	0	0	0	0	1	1	1	**IO7**
0	0	0	0	1	0	0	0	**IO8**
0	0	0	0	1	0	0	1	**IO9**
0	0	0	0	1	0	1	0	**IO10**
0	0	0	0	1	0	1	1	**IO11**
AS8268 only								

IN0/IN1 (950Dh/950Eh)

The IN registers (input registers) store the input data from the I/O pins. These registers are continuously updated by the 'Mclk' (main clock).

IN0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

IN1

MSB							LSB
0	0	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

OUT0 ... OUT11 (950Fh – 951Ah)

The OUT registers (output registers) contain the output data to be sent to the I/O pins (through the multiplexers).

OUT0

MSB							LSB
0	0	0	0	0	0	0	IO0

OUT1

MSB							LSB
0	0	0	0	0	0	0	IO1

OUT2

MSB							LSB
0	0	0	0	0	0	0	IO2

OUT3

MSB							LSB
0	0	0	0	0	0	0	IO3

OUT4

MSB							LSB
0	0	0	0	0	0	0	IO4

OUT5

MSB							LSB
0	0	0	0	0	0	0	IO5

OUT6

MSB							LSB
0	0	0	0	0	0	0	IO6

OUT7

MSB							LSB
0	0	0	0	0	0	0	IO7

OUT8

MSB							LSB
0	0	0	0	0	0	0	IO8

OUT9

MSB							LSB
0	0	0	0	0	0	0	IO9
AS8268 only							

OUT10

MSB							LSB
0	0	0	0	0	0	0	IO10
AS8268 only							

OUT11

MSB							LSB
0	0	0	0	0	0	0	IO11
AS8268 only							

PCNT0/PCNT1/PCNT2 (951Bh/951Ch/951Dh)

The PCNT registers (pulse counter registers) contain the result of the pulse counting for calibration purposes.

PCNT0

MSB							LSB
b7	b6	b5	b4	b3	b2	b1	b0

PCNT1

MSB							LSB
b15	b14	b13	b12	b11	b10	b9	b8

PCNT2

MSB							LSB
b23	b22	b21	b20	b19	b18	b17	b16

The maximum reference pulse frequency is defined below:

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Reference pulse frequency	f_{refp}			120	kHz	

STATUS0/STATUS1 (951Eh/951Fh)

The STATUS registers contain the irq flag register bits for each of the I/Os, the COUNT register bit which signals when a pulse counting should be started and the CINT flag bit which indicates when pulse counting has been completed.
The irq flag registers are cleared by software only (MCU), but they cannot be set by software.
The COUNT register bit can be set and reset by software (MCU). The COUNT register bit is cleared, when the pulse counter is finished and an interrupt has been generated.

STATUS0

MSB							LSB
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0

STATUS1

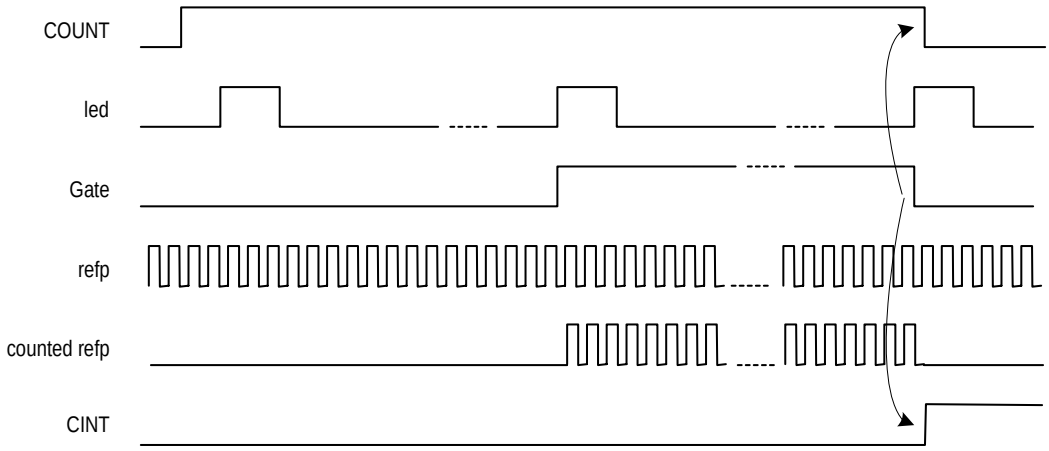
MSB							LSB
COUNT	CINT	0	0	IO11	IO10	IO9	IO8
				AS8268 only			

Notes:
1) IOx: 0: no change on input x 1: input x has changed
2) When an interrupt on an IO change has been generated, the signal irq is reset after the related flag has been cleared.
3) CINT is a flag, which indicates that the pulse counting has finished. When CINT is cleared, the irq is cleared.

Pulse Counter

A synchronous pulse counter is used. It is started after the COUNT bit has been set. The first led pulse is used for synchronisation. The second led pulse starts counting the reference pulses from the specified I/O input.

Timing:



Notes:
1) The COUNT signal is synchronized with 'led'.
2) COUNT is reset and CINT is set using clk and checking for falling edge on Gate.
3) The PCNT register is only updated when counting is finished.

Example:
Select IO3 as the pulse reference input.
Meter is $220V_{mains}$; I_{max} = 20A
Meter constant: 1,600imp/kWh
Reference meter constant: 16 million imp/kWh

Register settings:

MPIO	SET_EN0	(9505h):	IO3 bit = 0 (output disabled)
	SEL_IN_REFP	(950Ch):	02h
DSP	mconst	(9330h):	07h

(Ideal) Pulse_lev (932Ch – 932Ah) = $570{,}950 \times \frac{230V}{220V} \times \frac{40A}{20A}$

= 1,193,804 ⇒ 12374Ch

⇒ 932Ah: 4Ch
932Bh: 37h
932Ch: 12h

Procedure:
Status1 (951Fh): 80h → starts pulse counting.
When pulse counting is completed → CINT bit in Status1 = 1.

The status of the CINT bit in Status1 may be checked to confirm that the pulse counting is complete. Alternatively, the time between 2 pulses may be calculated to determine the count cycle time (the first pulse is used for synchronization and the second pulse starts the count cycle).

Following the pulse counting cycle, the number of pulses counted can be read from PCNT0/PCNT1/PCNT2 (951Bh/951Ch/951Dh).

The ideal number of pulses counted assuming the meter is perfectly calibrated would be:

$$Ni = \frac{16{,}000{,}000}{1{,}600} = 10{,}000$$

Assuming that we count 11,000 pulses, the (Ideal) Pulse_lev must be changed by the factor:

10,000/11,000 = 0.909
The new Pulse_lev = 1,193,804 x 0.909
= 1,085,168 ⇒ 108EF0h

⇒ 932Ah: F0h
932Bh: 8Eh
932Ch: 10h

8.5 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) represents a synchronous, bit serial 4-wire interface for full-duplex data transfer.
Depending on the operating mode (selectable by the sel_spi2 bit in the SCT enable signals register (9001h)) the interface can act as SPI2 Master/Slave interface (e.g. when an external EEPROM is connected) or as SPI_FLASH interface (when used as interface to the Flash).

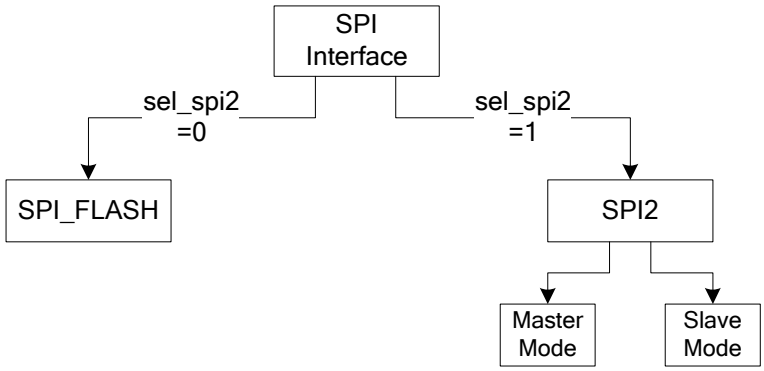


Figure 8: SPI Interface

8.5.1. SPI2 Master/Slave Mode

In this mode the SPI can operate in master mode, whereas the external EEPROM works in slave mode.
The external EEPROM memory must fulfil the requirements described below. The EEPROM is selectable in size from 1kByte to 32kByte in binary steps.

Key Features

- Standard 4 wire synchronous serial interface (MISO, MOSI, SC, S_N)
- Master/slave mode operation
- 8-bit word length (variable transmit/receive word optional)
- Shift clock SC high when idle
- MSB is always transmitted first
- Four selectable clocking schemes (clock idle state / clock phase)
- Selectable SPI clock rate divider (from mcu_clk/2 to mcu_clk/65536)
- Three maskable interrupts (transmission complete, overrun, collision)

SPI Registers

Register Name	Address	Description
SSPCON	9400h	Control register
SSPCLKDIV	9401h	Clock divider register
SSPSTAT	9402h	Status register
SSPBUF	9403h	Data register

Control Register (SSPCON, 9400h)

The control register is used for enabling the SPI-interrupts and to control the chip select of the SPI.

MSB							LSB
IETR	IEOV	IECO	IECS	CSO	-	AUTO	-

Bit	Symbol	Function
7	IETR	Transmit interrupt enable Issued after data register has been serially loaded with new data (slave mode) or if data has been shifted out after write access (master mode) 0: disable 1: enable
6	IEOV	Overrun interrupt enable Issued if ITRA still set and new data serially arrived 0: disable 1: enable
5	IECO	Write collision interrupt enable Issued if data register is written during transmission 0: disable 1: enable
4	IECS	Chip select interrupt enable Issued if chip-select pin is activated during master/slave mode 0: disable 1: enable
3	CSO	Chip select output state in master mode if AUTO = 0 Inverted state of output signal S_N 0: S_N = ‘1’ 1: S_N = ‘0’ (active)
2	ISOUT	Chip select output enable control in master mode
1	AUTO	1: Automatically activates the S_N (= ’0’) after data has been written to the data register and deactivates S_N (= ’1’) after transfer completed 0: S_N depends on the CSO bit (→ manual S_N setting)
0	-	Not used

Clock Divider Register (SSPCLKDIV, 9401h)

The clock divider register contains control bits to configure the clock-divider, to set-up the serial-clock SC, to enable the SPI and to select master or slave mode.

MSB							LSB
ENBL	CIDLE	CPHA	M/S	CLKDIV.3	CLKDIV.2	CLKDIV.1	CLKDIV.0

Bit	Symbol	Function
7	ENBL	SPI enable. Enables the SPI interface 1: enable 0: disable
6	CIDLE	Serial clock SC idle state 1: SC idles high 0: SC idles low
5	CPHA	Serial clock SC phase Data is samples and shifted out according to CIDLE/CPHA
4	M/S	Master/Slave mode 1: Master mode, must be '1' 0: Slave mode
3	CLKDIV.3	Clock divider exponent In master mode, SPI output clock SC is MCU_CLK / $2^{CLKDIV+1}$
2	CLKDIV.2	
1	CLKDIV.1	
0	CLKDIV.0	

CIDLE Bit6	CPHA Bit5	SC Idle	Data shifted out on SC	Input sampled on SC
0	0	0	falling	rising
0	1	0	rising	falling
1	0	1	rising	falling
1	1	1	falling	rising

Note: The CIDLE/CPHA set at 1 1 is used internally by most standard available EEPROMs

Bit3	Bit2	Bit1	Bit0	SC-Rate
0	0	0	0	1 : 2
0	0	0	1	1 : 4
0	0	1	0	1 : 8
0	0	1	1	1 : 16
0	1	0	0	1 : 32
0	1	0	1	1 : 64
0	1	1	0	1 : 128
0	1	1	1	1 : 256
1	0	0	0	1 : 512
1	0	0	1	1 : 1,024
1	0	1	0	1 : 2,048
1	0	1	1	1 : 4,096
1	1	0	0	1 : 8,192
1	1	0	1	1 : 16,384
1	1	1	0	1 : 32,768
1	1	1	1	1 : 65,536

The SPI output clock SC, which is derived from the mcu clock (mcu_clk) may be divided down as shown in the table above.

It is important to note, that the mcu_clk may also be divided down as described under MCUCLKDIV Register ('mcu_clk'). Therefore the SPI output clock SC is also dependent on the programming of the mcu_clk frequency.

Recommended programming for 3.579545MHz mcu clock rate, SPI enabled, SC clock phase = '11', master:

Value	SC Clock Rate
1.74MHz	F0h
0.895MHz	F1h
0.447MHz	F2h

Status Register (SSPSTAT, 9402h)

MSB							LSB
ITRA	IOVR	ICOL	-	CSI	-	-	-

Bit	Symbol	Function
7	ITRA[1]	Transmission complete interrupt issued. Issued after new data word is available in data-register (slave configuration), or if data-register has been shifted out after write access (master configuration)
6	IOVR[1]	Overrun interrupt issued. Issued if ITRA is still set from previous transmission and new data arrives (master and slave configuration)
5	ICOL[1]	Write-collision interrupt issued. Issued if data-register is written during receive (slave configuration) or transmit (master configuration)
4	-	Not used
3	CSI	Always '0', has no effect
2	-	Not used
1	-	Not used
0	-	Not used

Note:
1) Flag-bits change state independent of the state of the corresponding interrupt-enable bit of the control register.

The SPI interrupt status is captured in the SSPSTAT register. Each interrupt status bit can be masked by the SSPCON register, which is OR-ed to a single SPI interrupt request signal (SPI_IRQ).

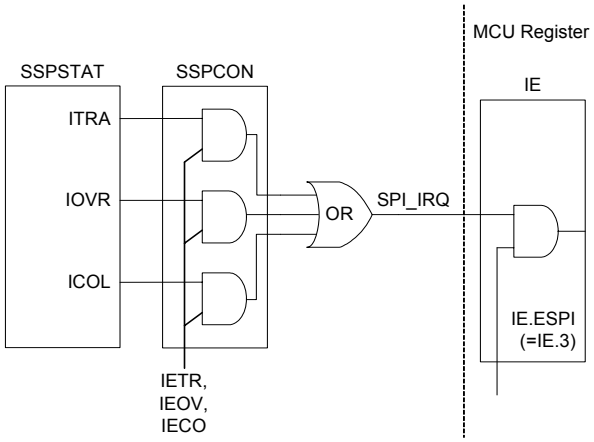
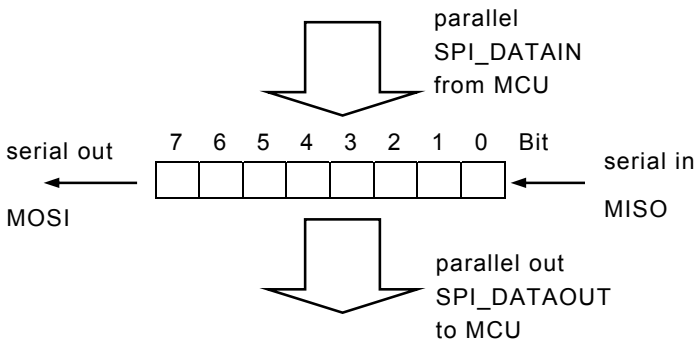


Figure 9: Block diagram

Data Register (SSPBUF, 9403h)

The data-register is an 8-bits wide shift-register with parallel load input and parallel output.



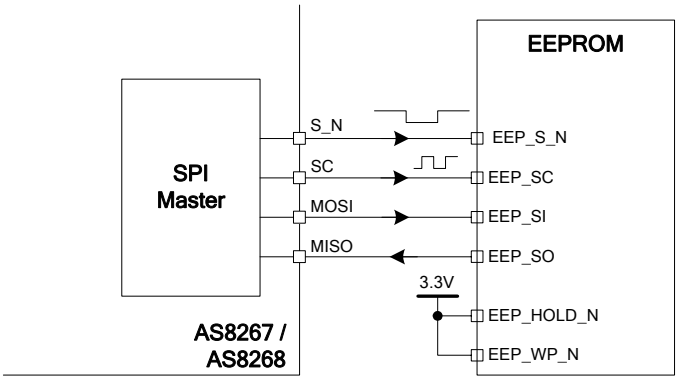


Figure 10: Typical SPI connection to an EEPROM

Many EEPROMs provide a HOLD_N (hold protocol) pin and a WP_N pin (write protect), which must be held '1', otherwise the operation is blocked.

8.5.2. SPI_FLASH Mode

In this mode the SPI operates in slave mode, in which the interface is used to communicate with the internal Flash memory.

The SPI_Flash block contains an 8-bit instruction register. It is accessed via the MISO pin, with data being clocked in on the rising edge of SC. The S_N pin must be low. The table below contains the list of the possible instruction bytes and format for SPI_Flash block operation. All instructions, addresses, and data are transferred MSB first, LSB last. Data is sampled on the first rising edge of SC after S_N goes low.

Instruction Set

Instruction Name	Instruction Code	Description
READ	03h	Read data from memory array beginning at selected address
PP	02h	Page Program, moves data to selected memory page
WREN	06h	Set write enable latch (enable write and erase operations)
WRDI	04h	Reset the write enable latch (disable write operations)
RDSR	05h	Read Flash status register (FLASH_STAT, 9700h)
WRSR	01h	Write Flash status register (FLASH_STAT, 9700h)
PE	D8h	Page Erase
ME	C7h	Mass Erase
RESET	E9h	Sets external reset pin (may be used to reset the whole device)

Functional Description

Arbitration

After a write, page erase or mass erase command the Flash memory is busy for some time and no command must be sent. To handle concurrent access from MCU and the MCU independent interfaces (UART1, SPI via SPI_Flash) to the Flash memory an arbitration procedure is necessary for the external interfaces:

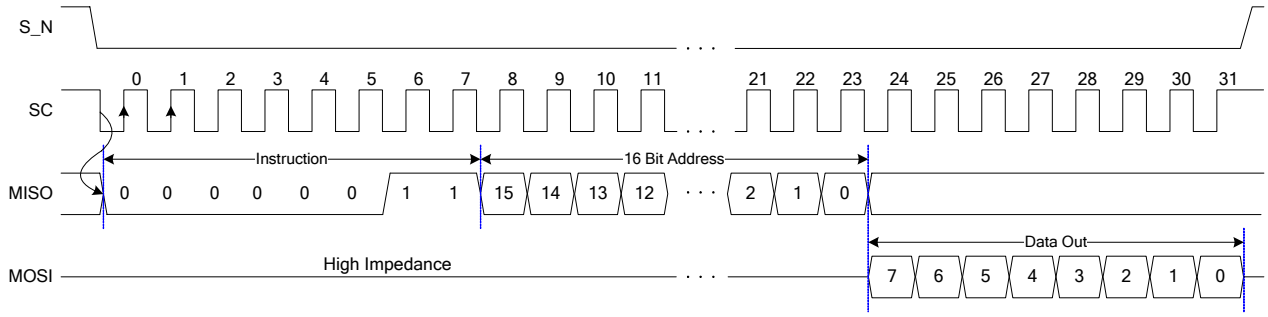
1. Send a request to the Flash (Set REQ bit in FLASH_STAT register)
2. Poll status of Flash until Flash is ready (WIP bit in FLASH_STAT register must be 0)
   -> Flash memory is reserved for external interface and MCU remains halted.
3. Send one or several Flash commands (Flash must not be busy before sending the next Flash command)
4. Release Flash request (Clear REQ bit in FLASH_STAT register)
   -> MCU is running again

Request is needed for all Flash commands
Concurrent access of both MCU independent interfaces is not allowed.

Read

The SPI_Flash is selected by pulling S_N low. The 8-bit read instruction is transmitted to the SPI_Flash followed by the 16-bit address with the MSB (address[15]) of the address word being don't care. After the correct read instruction and address are sent, the data stored in the memory at the selected address is shifted out on the MOSI pin. (After an access time of 13 system clocks the Flash data is available in the SPI transit register). The data stored in the memory at the next address can be read sequentially by continuing to provide clock pulses. The internal address pointer is automatically incremented to the next higher address after each byte of data is shifted out. When the highest address is reached (7FFFh), the address counter rolls over to address 0000h allowing the read cycle to be continued indefinitely. Rising the S_N pin terminates the read operation.

Timing



Example Read FLASH Sequence

1. 0x01 … … set REQ bit by writing 0x04 to status register
2. 0x04 … -"-
3. 0x05 … read Flash status register until Flash is ready
4. 0x03 … read Flash instruction
5. 0x00 … Start address, e.g. 80h
6. 0x80 … -"-
7. Dat0 … read byte 0 from address
8. Dat1 … read byte 1 from address + 1
9. … … read bytes as long as you want, after address overrun it restarts at address 0.
10. 0x01 … clear REQ bit by writing 0x00 to status register
11. 0x00 … -"-

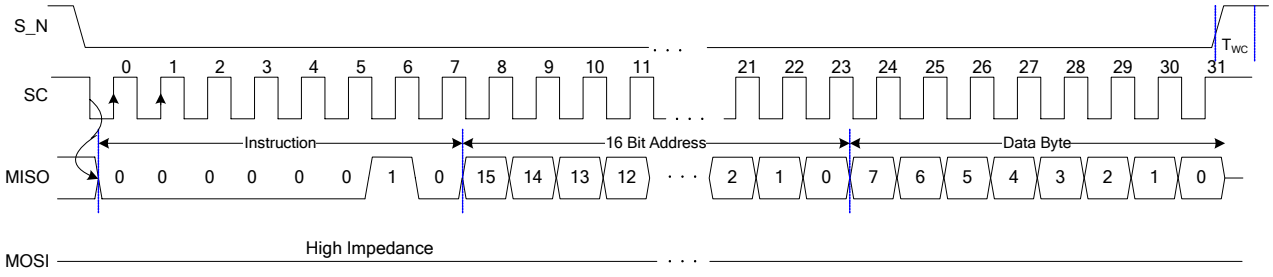
Page Program

The whole memory of 32K bytes is split into 512 pages with 64 bytes per page. Each page can be written with 64 bytes at once or, it can be written byte-wise or in groups of bytes. The addressing order is arbitrary. Prior to any attempt to write data to the SPI_Flash or status register, the write enable latch must be set by issuing the WREN instruction. Setting S_N low and then clocking out the proper instruction into the SPI_Flash does this. After all eight bits of the instruction are transmitted; the S_N must be brought high to set the write enable latch. If the write operation is initiated immediately after the WREN instruction without S_N being brought high, the data will not be written to the array because the write enable latch will not have been properly set.

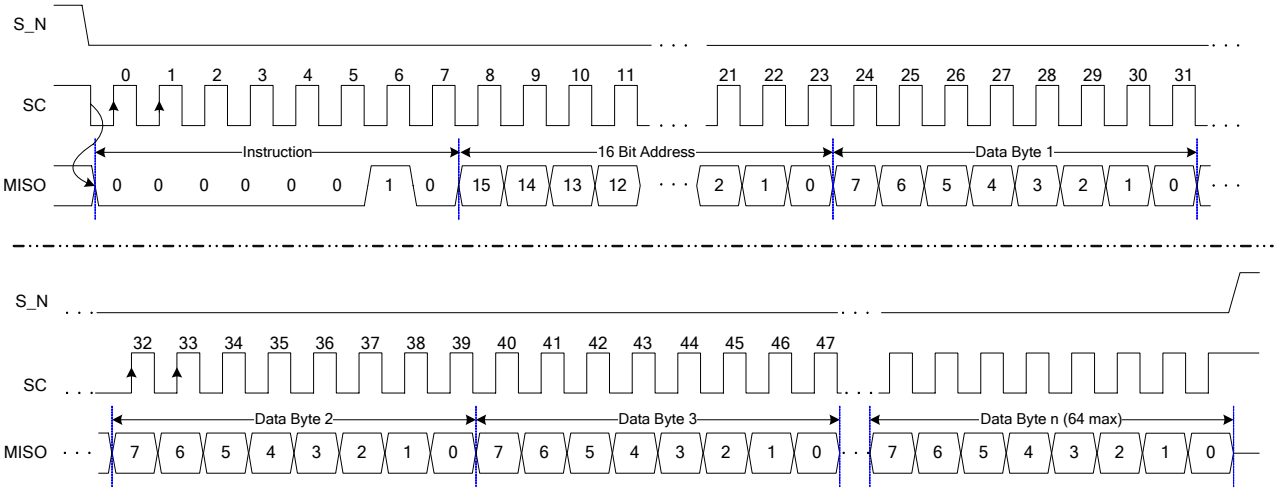
Once the write enable latch is set, the user may proceed by setting the S_N low, issuing a write instruction, followed by the address, and then the data to be written. Up to 64 bytes of data can be sent to the SPI_Flash before a program cycle is necessary. The only restriction is that all of the bytes must reside in the same page. An address consists of a page address (9 bits) and the address in page (6 bits), where the page address = address [15:6] and the address in page is address [5:0]. If the internal address counter reaches 0x7FFF and the clock continues, the counter will roll over to the first address 0x0000.

For the data to be actually written, the S_N must be brought high after the least significant bit (D0) of the n^{th} data byte has been clocked in. If S_N is brought high at any other time, the write operation will not be completed. While the write is in progress, the status register may be read to check the status. A read attempt of a memory array location will not be possible during a write cycle. When the write cycle is completed, the write enable latch is reset.

Byte Write



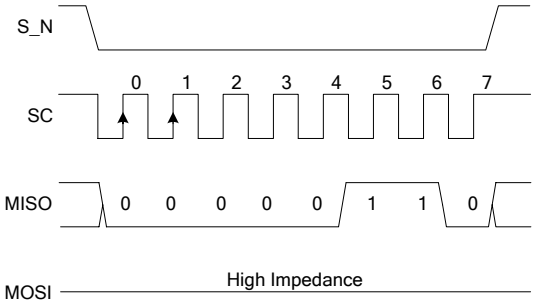
Page Program (max. 64 bytes)



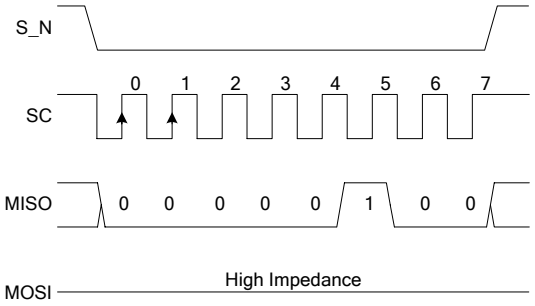
Example PROG Page Sequence

1. 0x01 … set REQ bit by writing 0x04 to status register
2. 0x04 … -“-
3. 0x05 … read Flash status register until Flash is ready
4. 0x06 … set Flash write enable
5. 0x02 … write Flash instruction
6. 0x00 … Start address, e.g. 80h
7. 0x80 … -“-
8. Dat0 … put byte 0 to address
9. Dat1 … put byte 1 to address + 1
10. … … max. 64 bytes (page size), then repeat sequence from step 2.
11. 0x01 … clear REQ bit by writing 0x00 to status register
12. 0x00 … -“-

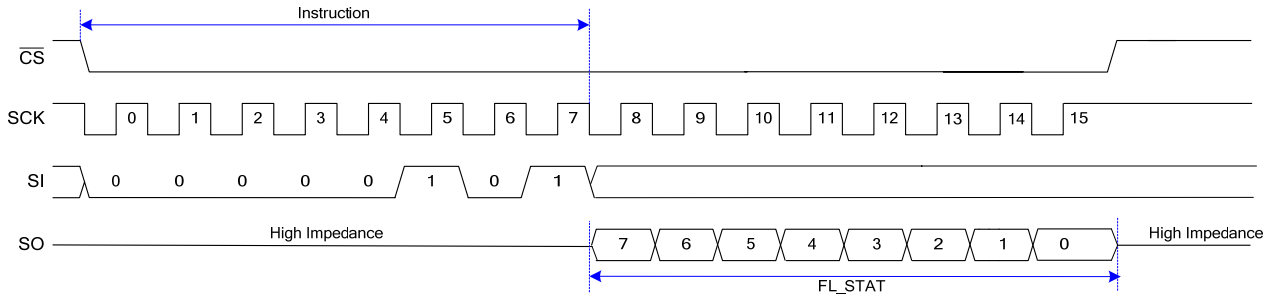
Write Enable Sequence



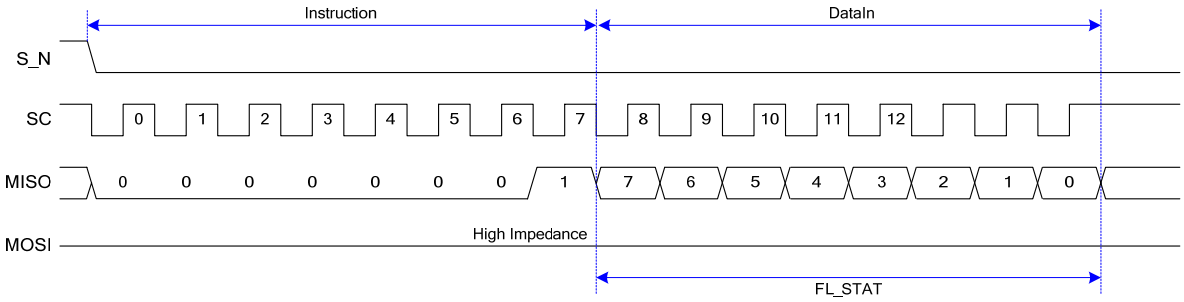
Write Disable Sequence



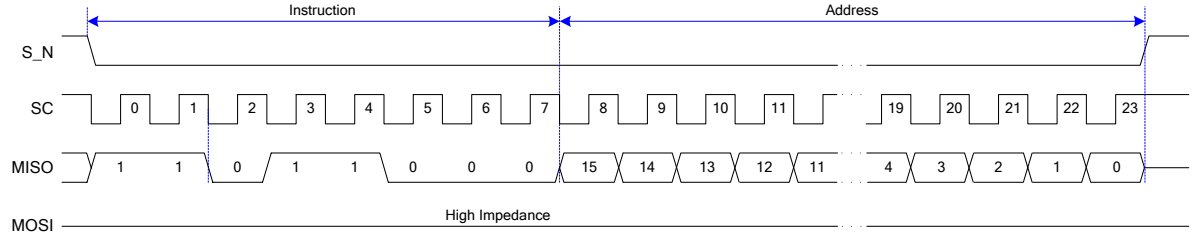
Read Status Register



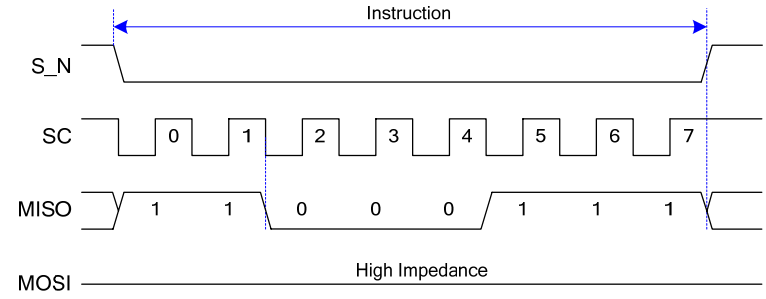
Write Status Register



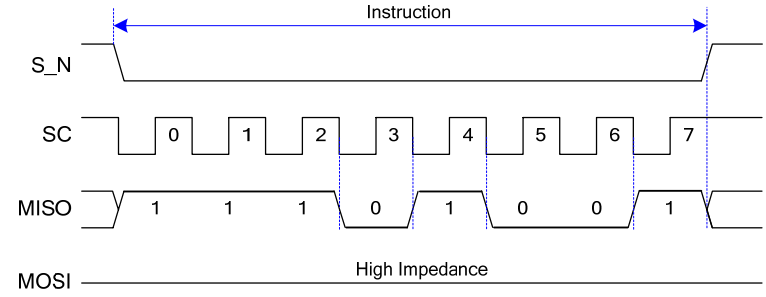
Page Erase



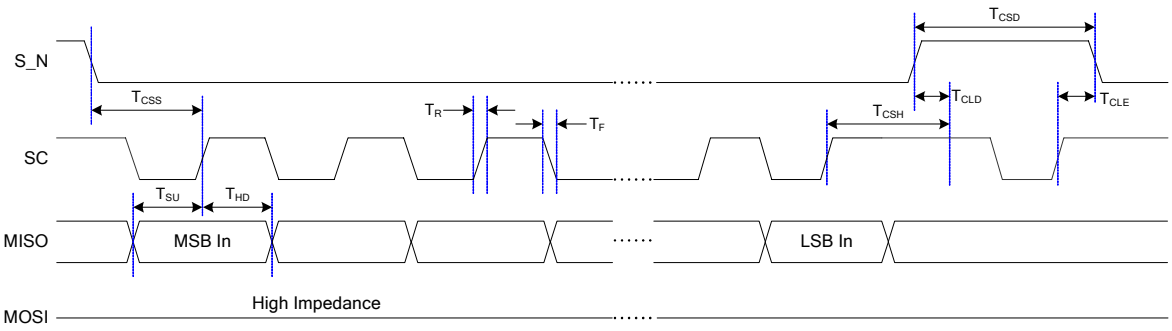
Mass Erase



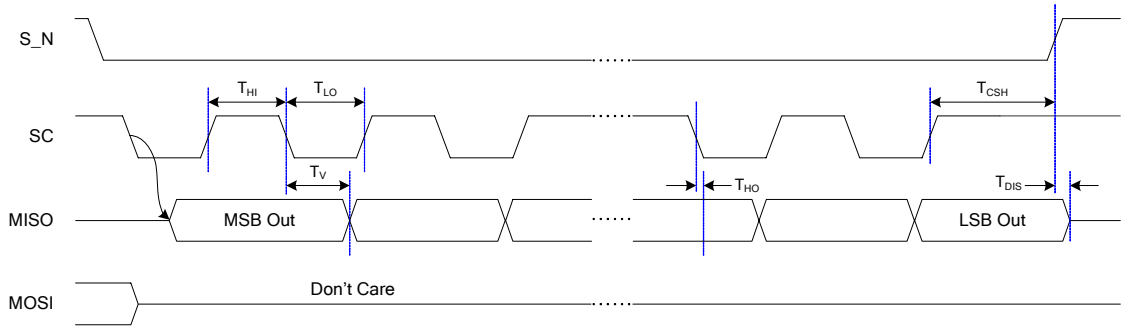
Reset



Serial Input Timing (MOSI)



Serial Output Timing (MISO)



Timing Characteristics

Parameter	Symbol	Min	Max	Unit	Note
S_N Setup Time	T_{CSS}	100	-	ns	
S_N Hold Time	T_{CSH}	150	-	ns	
S_N Disable Time	T_{CSD}	500	-	ns	
Data SetupTime	T_{SU}	30	-	ns	
Data Hold Time	T_{HD}	50	-	ns	
SC Rise Time	T_R	-	2	ns	
SC Fall Time	T_F	-	2	ns	
SC High Time	T_{HI}	150	-	ns	
SC Low Time	T_{LO}	150	-	ns	
SC Delay Time	T_{CLD}	50	-	ns	
SC Enable Time	T_{CLE}	50	-	ns	
Output Valid from Clock Low	T_V	-	150	ns	
Output Hold Time	T_{HO}	0	-	ns	
Output Disable Time	T_{DIS}	-	200	ns	

8.6 External EEPROM Requirements

An external EEPROM with SPI bus serial interface is used for non-volatile program and data storage. The SPI master block that communicates with the EEPROM is specified above. This section explains the requirement for Serial EEPROMs. It shows the most important figures and tables as a reference. For the details please turn to the data sheet of your specifically applied EEPROM.

The following minimum requirements must be fulfilled:

Pins

There must be at least the typical SPI pins like

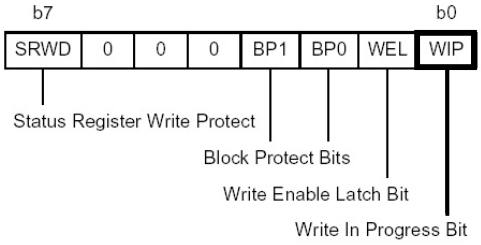
serial data input (EEP_SI), serial data output (EEP_SO)
serial clock input (EEP_SC), chip select input (EEP_S_N)

Clock Rate

The applicable clock rate pin EEP_SC must be ≥ 1MHz.

Status Register

must look like this:



Bit 0 must be the WIP bit, indicating that a write operation is in progress. Only this bit is polled during the EEPROM upload, means programming of the EEPROM. The Status register can be accessed via the RDSR instruction.

Data Protection

The write protection block size is given in the table below:

Status Register Bits		Protected Block	Array Addresses Protected
BP1	BP0		Example only
0	0	None	None
0	1	Upper quarter	6000h – 7FFFh
1	0	Upper half	4000h – 7FFFh
1	1	Whole memory	0000h – 7FFFh

Note: The array addresses must be referenced from the data sheet of the specific EEPROM used.

BP1, BP0 allows the selection of one out of 4 protection schemes.

In order to protect against inadvertent programming the user can see these bits. Please note that the protected range of EEPROM cannot be overwritten via an SCT command there anymore. Reprogramming must be done with a dedicated program then.

Instruction Set

Instruction Name	Instruction Format	Description
READ	03h	Read data from memory starting with selected address
WRITE	02h	Write data to memory beginning at selected address. Most EEPROMs allow page writing of pages 16, 32, 64 or even more bytes for faster device programming. Before every page write operation a WREN instruction must be applied – see also bootloading and uploading sequence for details.
WREN	06h	Write enable EEPROM, enables write operation
RDSR	05h	Read EEPROM Status register
WRDI	08h	Write disable EEPROM, disable write operation
WRSR	01h	Write EEPROM Status register

SPI Modes

These devices can be driven by a microcontroller with its SPI peripheral running in either of the two following modes:

- CPOL = 0, CPHA = 0
- CPOL = 1, CPHA = 1 (It is recommended to set CPOL = 1, CPHA = 1 in your program: The build-in bootloader uses this setting as well.)

For these two modes, input data is latched in on the rising edge of Serial Clock (SC), and output data is issued on the falling edge of Serial Clock (SC).
The recommended mode is shown in Figure 11. The clock polarity SC is ‘1’ when the bus master is in Stand-by mode and not transferring data (idle state):

- SC remains at 1 for (CPOL = 1, CPHA = 1)

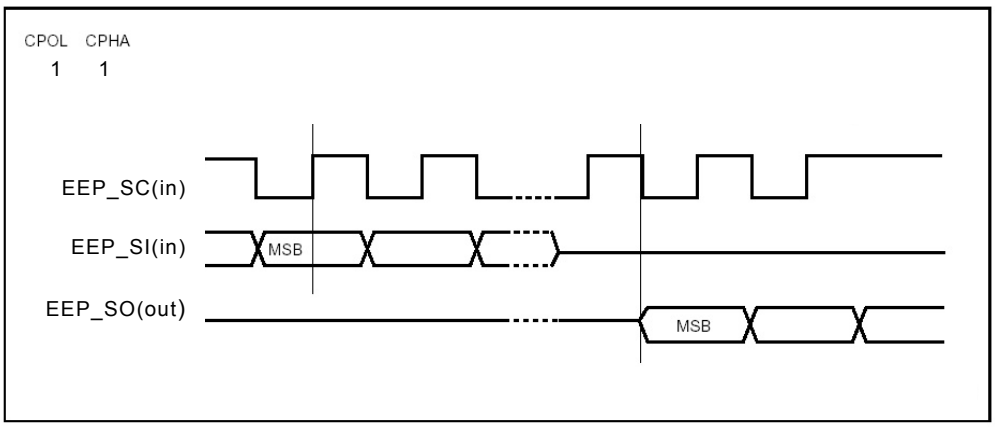


Figure 11: SPI modes recommended

Address Roll Over

When the highest address on the EEPROM is reached, e.g. 7FFFh for a 32kB device, then the address counter must roll over to 0000h.

Unused Upper Address Bits

Unused upper address bits must be ignored in any case. E.g. an 8kB device has a maximum address of 1FFFh must interpret 7FFFh as 1FFFh, ignoring the higher bits.

Example Pin List

Pin Name	Type	Functionality	Description
EEP_S_N	Input	Chip select, active low	When this input signal is High, the device is deselected and Serial Data Output (SO) is at high impedance. Unless an internal Write cycle is in progress, the device will be in the Standby mode. Driving Chip Select (S_N) Low enables the device, placing it in the active power mode.
EEP_SO	Output	Serial data output	This output signal is used to transfer data serially out of the device. Data is shifted out on the falling edge of Serial Clock (SC).
EEP_SI	Input	Serial data input	This input signal is used to transfer data serially into the device. It receives instructions, addresses and the data to be written. Values are latched on the rising edge of Serial Clock (SC).
EEP_SC	Input	Serial clock	This input signal provides the timing of the serial interface. Instructions, addresses or data present at Serial Data Input (SI) are latched on the rising edge of Serial Clock (SC). Data on Serial Data Output (SO) changes after the falling edge of Serial Clock (SC).
EEP_WP_N [1)]	Input	Write protect, active low	The main purpose of this input signal is to freeze the size of the area of memory that is protected against Write instructions (as specified by the values in the BP1 and BP0 bits of the Status register). This pin must be driven either High or Low and must be stable during all write operations.
EEP_HOLD_N [1)]	Input	Hold, active low	The Hold (HOLD_N) signal is used to pause any serial communications with the device without deselecting the device. During the Hold condition, the Serial Data Output (SO) is high impedance, and Serial Data Input (SI) and Serial Clock (SC) are Don't Care. To start the Hold condition, the device must be selected, with Chip Select (S_N) driven Low.
EEP_VCC	Supply	Positive supply voltage	
EEP_VSS	Supply	Negative supply voltage	

Note:

1) No Write Protect (EEP_WP_N) and Hold (EEP_HOLD_N) pins are available on the AS8267 / AS8268 ICs. These pins must be tied 'high' directly at the EEPROM device.

Instructions Timings

Write Enable (WREN)

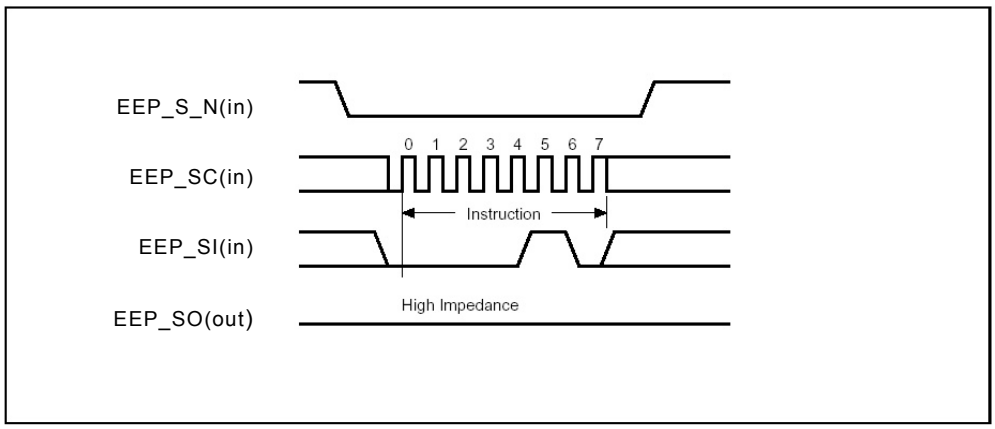


Figure 12: Write enable (WREN) sequence

Read Status Register (RDSR)

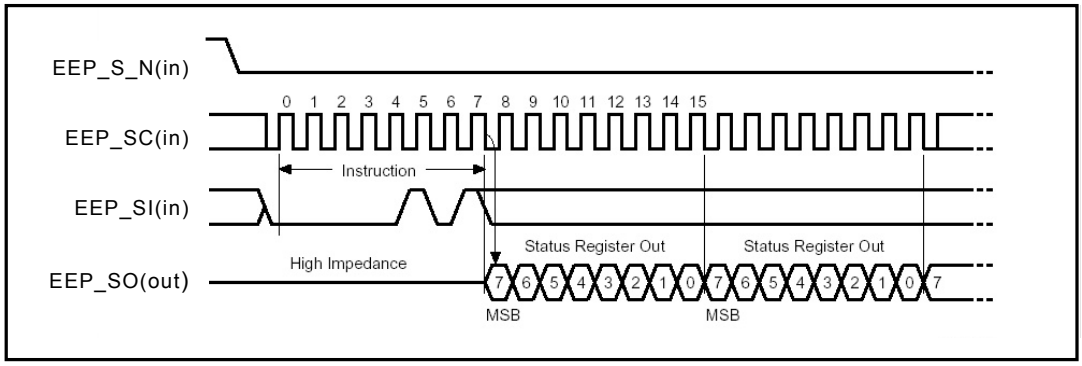


Figure 13: Read Status register (RDSR) sequence

Read from Memory Array (READ)

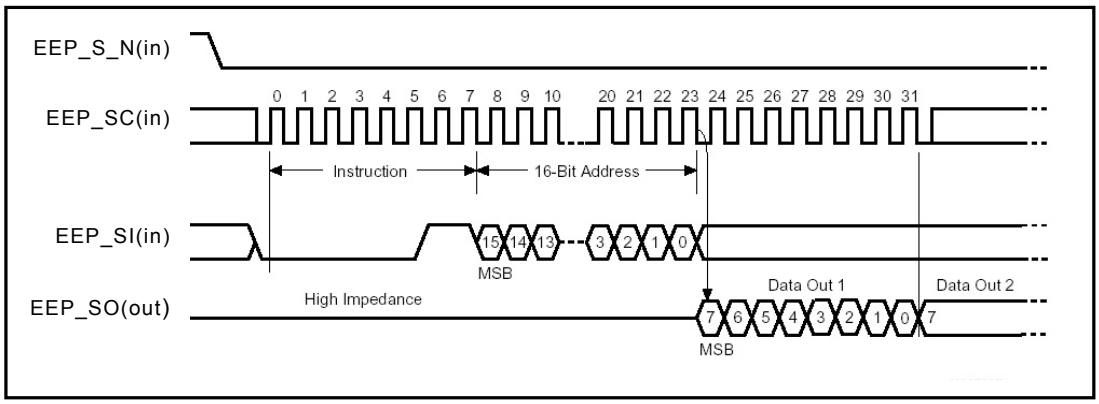


Figure 14: Read from memory array (READ) sequence

Write to Memory Array (WRITE)

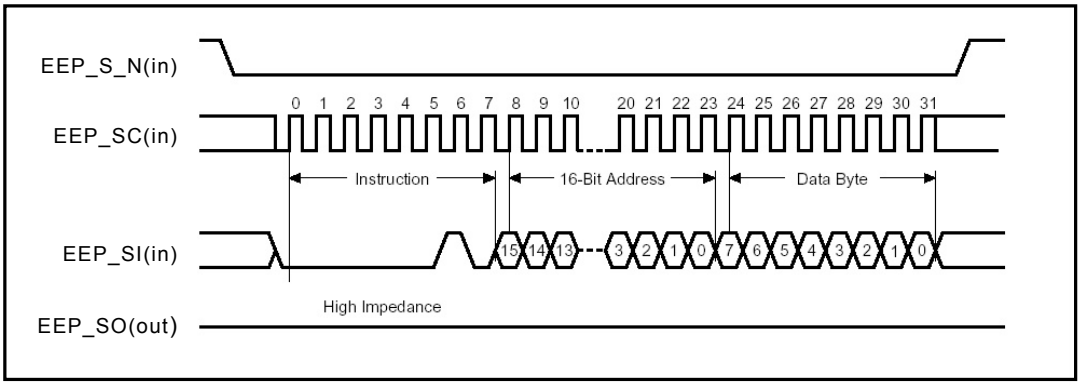


Figure 15: Byte write (WRITE) sequence

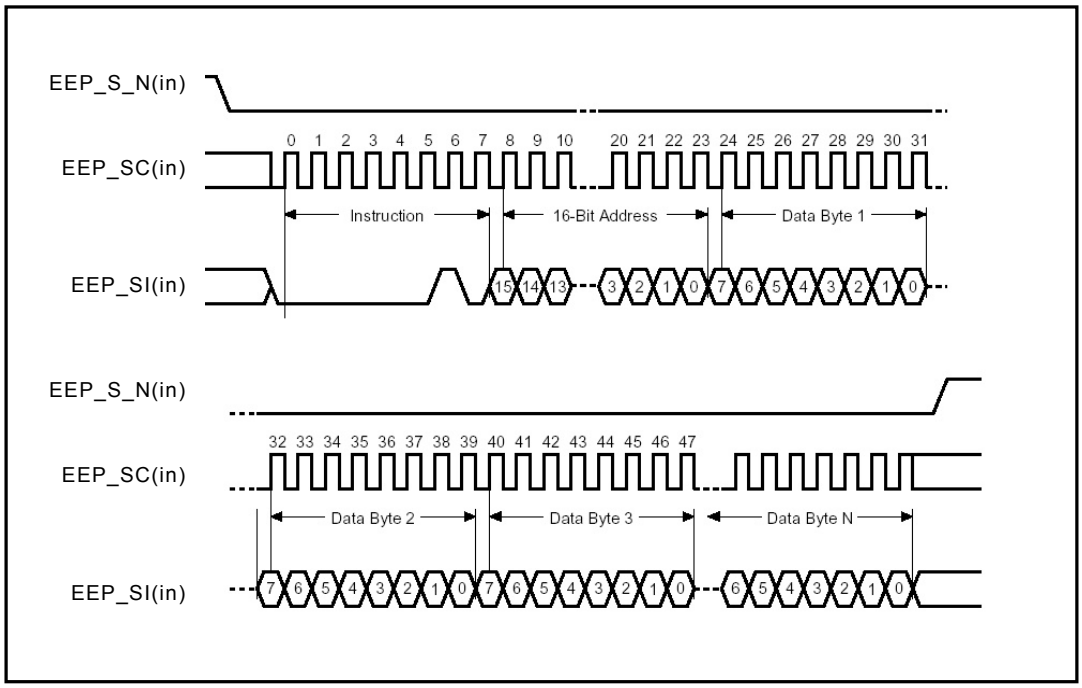


Figure 16: Page write (WRITE) sequence

8.7 FLASH Memory

The AS8267 / AS8268 provide a 32kByte Flash Memory for program and data. This Flash is organised in 512 pages with 64Bytes each. When the memory is erased all bytes are 0x00.

To speed up writing to the Flash Memory, a page write mode is available. A page program writes the data in the addressed memory page. It is also possible to write parts of a page or a single byte. Data that has not been addressed remains unchanged. Before data can be written to the memory, the addressed page or the whole memory must be erased. A mass erase sets all bit cells in the memory to logic '0'. A page erase sets all bit cells of an addressed page to logic '0'.

The Flash memory can be accessed via System Control in UART1 or the SPI_Flash commands.

FLASH Registers

Register Name	Address	Reset Value	Note
FLASH_STAT	9700h	0rra.0000	1)
FLASH_ATTACK	9701h	000s.ssss	1)

Note:

1) rr and s.ssss status bits are copied from Flash memory during boot sequence
'a' is dependent of LOCK bit after reset. If LOCK bit is set then 'a' becomes 0 (access denied) otherwise 'a' becomes 1 and access is granted.

FLASH Status Register

MSB							LSB
CPU_PE	LOCK	ATK_EN	ACCESS_EN	-	REQ	WEL	WIP

Bit	Symbol	Function
7	CPU_PE	CPU page erase, triggers a page erase process of the Flash on next CPU write access. Must be cleared by CPU afterwards (Not to be used by UART1) (read/write)
6	LOCK	Locks the Flash memory against unauthorized read access from outside using SET_PW(f7h) or SET_PW1(f8h) command via UART1. Stored in Flash memory (non-volatile) (read only)
5	ATK_EN	Enables the attack counter using SET_PW1 (f8h) command via UART1. Stored in Flash memory (non-volatile) (read only)
4	ACCESS_EN	Grant access to Flash if password entered correctly (read only)
3	-	Not used
2	REQ	External Flash Request (Not to be used by CPU due to dead lock!) (read/write)
1	WEL	Write enable latch (only writeable by SPI "WREN" command)
0	WIP	Write in progress (read only)

Note:

The WIP bit is set as soon as a Write, Page Erase or Mass Erase command is sent and reset when the Flash is ready again.

FLASH Attack Register

If enabled by the ATK_EN bit in the Flash status register, the attack register logs any unauthorized access to the device and stores it in the Flash memory. After five attacks the device disables the UART1 interface forever.

MSB							LSB
-	-	-	A5	A4	A3	A2	A1

Bit	Symbol	Function
7	-	Not used
6	-	Not used
5	-	Not used
4	A5	5th attack
3	A4	4th attack
2	A3	3rd attack
1	A2	2nd attack
0	A1	1st attack

Whenever the password is not entered correctly and the attack counter is enabled one bit of the attack register is set and its copy is updated in the Flash memory.

Data Organisation in the FLASH Memory

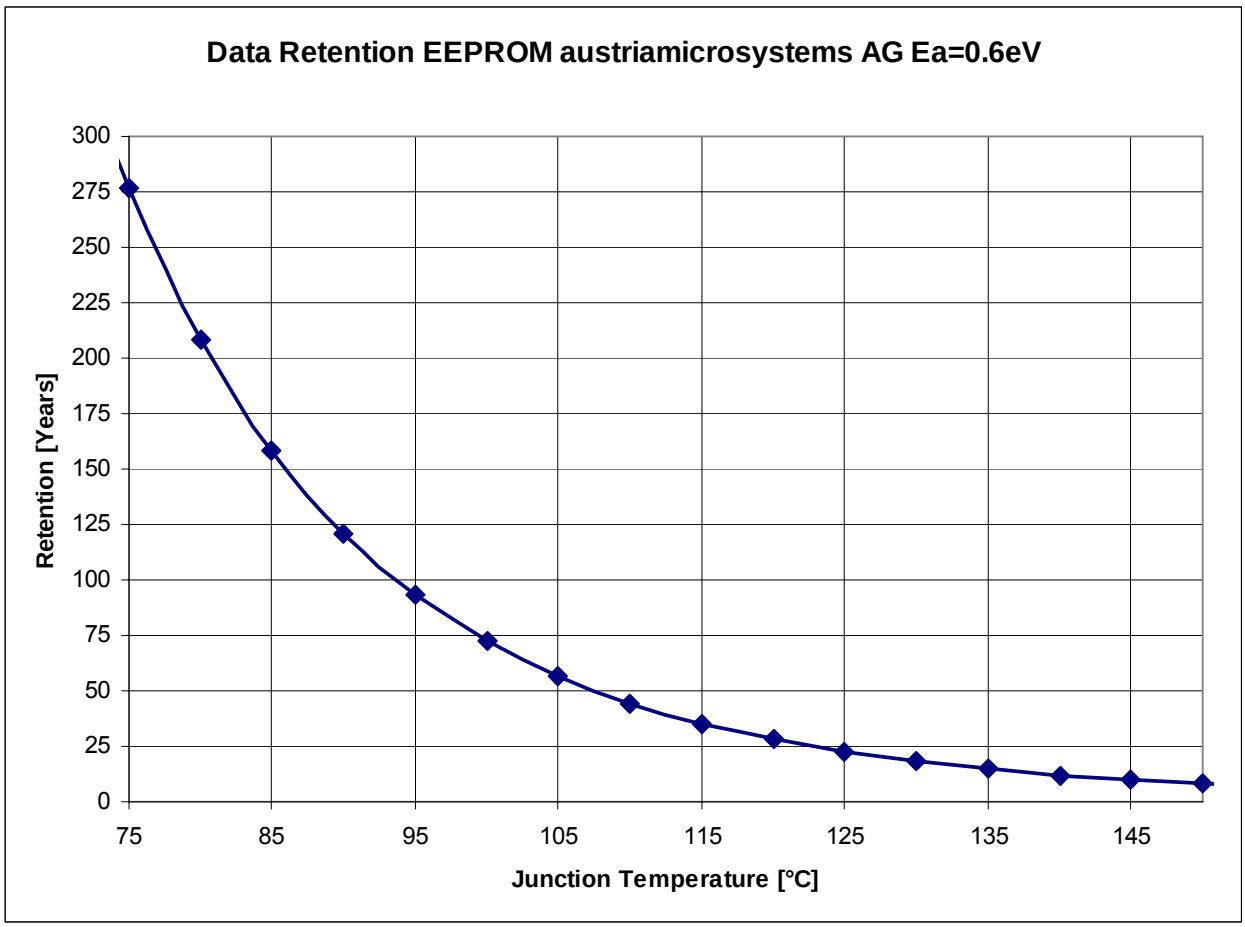
1. Program data are stored beginning at address 0000h.
2. Program data size must not be bigger than 32768 – 16 (0000h – 7FEFh).
3. The length of the program is stored at the two topmost bytes.
   For the 32k Flash memory this means: Length (takes 2 bytes) is stored at 7FFE to 7FFFh.
4. The 8byte password is stored at 7FF0h to 7FF7h.
5. Non-volatile Flash status flags are located at 7FF8h to 7FF9h.
6. Meter data and system parameters are stored in the remaining memory space. The allocation of memory space is totally up to the MCU program.
7. The program data will not be protected against overwriting processes from the MCU program.
8. In case there is no program stored in the Flash (boot loader detects all 0s or all 1s at the program length address) the boot loader forces the MCU to loop on address 0 (“SJMP $”, Hex code: 80FEh).

FLASH Timing

Parameter	Typ	Unit	Note
Program time	6.75	ms	fclk = 3.58MHz
Erase time	3.39	ms	fclk = 3.58MHz

FLASH Memory Reliability

The 32kByte Flash memory implemented in the AS8267 / AS8268 ICs provide an outstanding performance in respect to data retention time and endurance.
Endurance is the parameter that specifies the cumulative write/erase cycles of the memory cells within the Flash memory. The data retention time of a Flash memory is a critical end-of-life parameter. This parameter specifies the maximum period of time, after programming, that data can be expected to be retrieved valid from the memory.
According to the JEDEC A117 specification the Flash memory has a minimum endurance of 100,000 cycles and a retention time of 500 years at 65°C.



FLASH Security

The AS8267 / AS8268 comprise of two different possibilities to protect Flash content against copying and manipulation.
The detailed implementation will be explained in this chapter.

General Description

Based on the software development flow it only makes sense to lock the software after the development is finished. Therefore we can distinguish between access to the Flash during the software development and access to the Flash after the development is finished.
The protection is implemented in such a way that the external Flash memory access (UART1, SPI via SPI_Flash) is blocked.

Access during Software Development

During the development phase of the meter software each external access to the Flash must be enabled. This means that the listed commands are enabled

- READ
- WRITE Byte
- WRITE Page
- PAGE ERASE
- MASS ERASE

Access after Software Development

After the completion of the software development there are two possible modes of protecting the Flash content.

a) Protection via PASSWORD

If this mode is selected the listed commands are disabled:

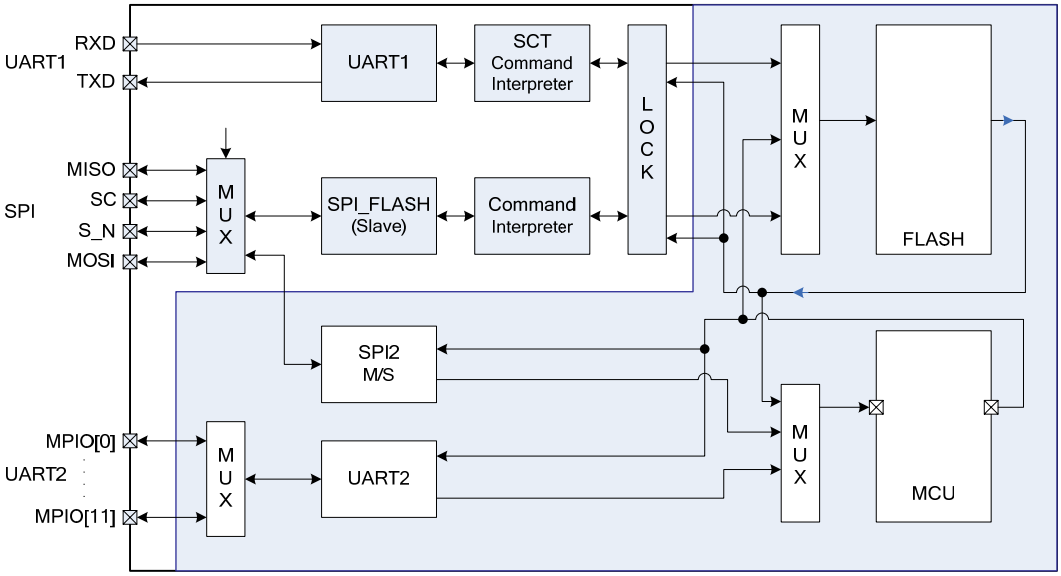
- READ
- WRITE Byte
- WRITE Page
- PAGE ERASE
- MASS ERASE

b) Protection via PASSWORD and ATTACK COUNTER

If this mode is selected the listed commands are disabled and the Attack Counter is enabled:

- READ
- WRITE Byte
- WRITE Page
- PAGE ERASE
- MASS ERASE

Block Diagram



The block diagram shows the main block involved in the security concept.
According to the block diagram there are four possibilities to get access to the internal Flash memory.

1. Access via UART1
   In this mode program/data can be read from or written in the Flash using commands defined in the SCT (System Control).

2. Access via SPI_FLASH Interface
   In this mode program/data can be read from or written to the Flash using the SPI_Flash interface. This path also includes a command interpreter which is able to handle different Flash access commands. Please refer to the SPI section in this document.
   To use this mode the SPI interface must be configured as slave (SPI_Flash).

3. Access via SPI2 Interface
   In this mode it is not possible to directly access the Flash memory from extern due to the missing command interpreter. A direct access to the Flash therefore would only be possible if there would be a program available in the mcu doing the command interpretation.

4. Access via UART2
   In this mode there is also no command interpreter in the communication path, so that a direct access to Flash is not possible.

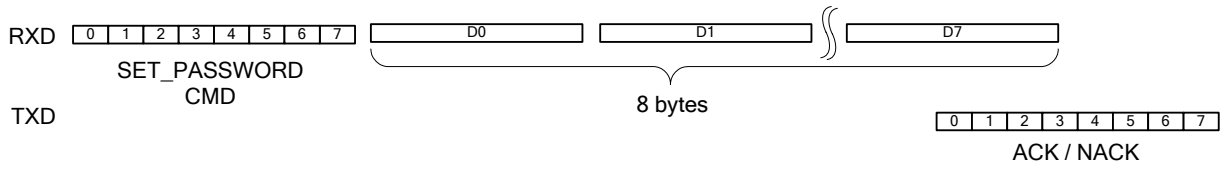
The Flash memory access via UART1 or SPI_Flash interface is an external Flash memory access and therefore protected by password.
The Flash memory access via SPI2 interface or UART2 interface is an internal Flash memory and therefore not protected by password. The protection of this path is up to the user.

Password Protection

If the protection schema “Protection via PASSWORD” is selected the listed commands are disabled:

- READ
- WRITE Byte
- WRITE Page
- PAGE ERASE
- MASS ERASE

The password is entered via the SET_PW command (instruction code F7h in the UART1 command interpreter) followed by the 8byte password.



Once a password is entered it is encrypted and stored in the Flash memory. At the same time the Flash memory ‘LOCK’ bit (bit6) is set in Flash status register (9700h). When ‘LOCK’ bit is set the top page of the Flash memory (storage of program length, password, non-volatile status flags) is blocked for page erase and write access even when access is granted. This also guarantees that the protection remains even the device is powered down and powered up again.

Based on the blocked write access of the top page of the Flash memory it is not possible to change an existing password. A new password can only be assigned after a MASS ERASE.

Once the correct password is entered the listed commands are enabled again.

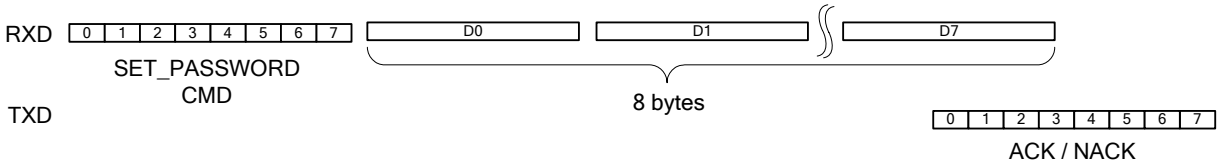
If the device memory is blank (e.g. after FAB-out) the access to the Flash memory is open and no password is required.

Password + Attack Counter Protection

If the protection schema “Protection via PASSWORD and ATTACK COUNTER” is selected the listed commands are disabled:

- READ
- WRITE Byte
- WRITE Page
- PAGE ERASE
- MASS ERASE

The password is entered in this case via the SET_PW1 command (instruction code F8h in the UART1 command interpreter) followed by the 8byte password.



In this mode the ATK_EN bit (bit5) is enabled in the Flash status register (9700h), and the attack register logs any unauthorized access to the device and stores it in the Flash memory. Also in this mode once a password is entered it is encrypted and stored in the Flash memory. At the same time the Flash memory 'LOCK' bit (bit6) is set in Flash status register. When 'LOCK' bit is set the top page of the Flash memory (storage of program length, password, non-volatile status flags) is blocked for page erase and write access even when access is granted. This also guarantees that the protection remains even the device is powered down and powered up again.

Based on the blocked write access of the top page of the Flash memory it is not possible to change an existing password. A new password can only be assigned after a MASS ERASE.

Once the correct password is entered the listed commands are enabled again.

In case an incorrect password is entered the Attack Counter is increased. After five attacks the UART1 will be disabled by switching off the internal clock for the UART1.

In this state the device is locked forever.

If the device memory is blank (e.g. after FAB-out) the access to the Flash memory is open and no password is required.

To give the user the possibility of reusing a blocked device he has to implement special functionality in his software.

The following example describes a possible implementation.

- Monitoring of one of the non blocked interfaces (SPI2 or UART2) within the customer specific MCU software
- If a specific (defined by the developer) sequence is applied the MCU can perform a page erase of the up most page (holds also program length) in the Flash memory.
- After a reset the device will now start with its default parameters and will not perform an automatic program load via the boot loader.
- In this operating mode it is than possible to write the program length again into the Flash memory. Also a new password can be entered.
- After a reset the meter will work again and also stored metering data can be accessed.

MCU Access to the FLASH Memory

The listed commands are available for the MCU access.

- READ
- WRITE Byte
- PAGE ERASE

Initiation of a PAGE ERASE by the MCU the CPU_PE bit in the FLASH Status Register (9700h) has to be set. After this a WRITE command with the selected address has to be performed.
After completion of the page erase procedure the CPU_PE bit has to be cleared by the MCU.

8.8 8051 Microcontroller (MCU)

The MCU is a derivative of the well-known 8051 microcontroller. The MCU block consists of an 8051 compatible microprocessor core, Flash memory, data memory (X_RAM), squareroot calculation unit and two UARTs for debugging and communication purposes. The Special Function Registers (SFR) section enfolds the standard blocks like the 16 bit timer (Timer 0), 128 bytes of internal data memory (I_RAM) and a serial interface (UART1). Furthermore, a squareroot block and a second serial interface (UART2) are also provided. Timer 1, Port 0 to 3 and the UART are not implemented exactly the same as in the original 8051. Instead, the bus extension (Port 0, 2 on single chip 8051) provides access to on-chip periphery, which comprises a serial peripheral interface (SPI), a real time clock (RTC), nine general purpose I/Os (MPIO), the LCD driver (LCDD), the DSP block that interfaces to the analog front end and system control registers (SCT). The MCU block is configured as Von Neumann architecture with the program in the Flash memory staring from 0000h and the data memory (X_RAM) and periphery section starting from 8000h up to FFFFh. All 64kB of memory can be accessed with both, the MOVC instruction (for program fetches and data read) and the MOVX instruction (for data read/store). The interrupt controller enfolds 7 internal interrupt sources, for having all necessary peripherals already on chip.

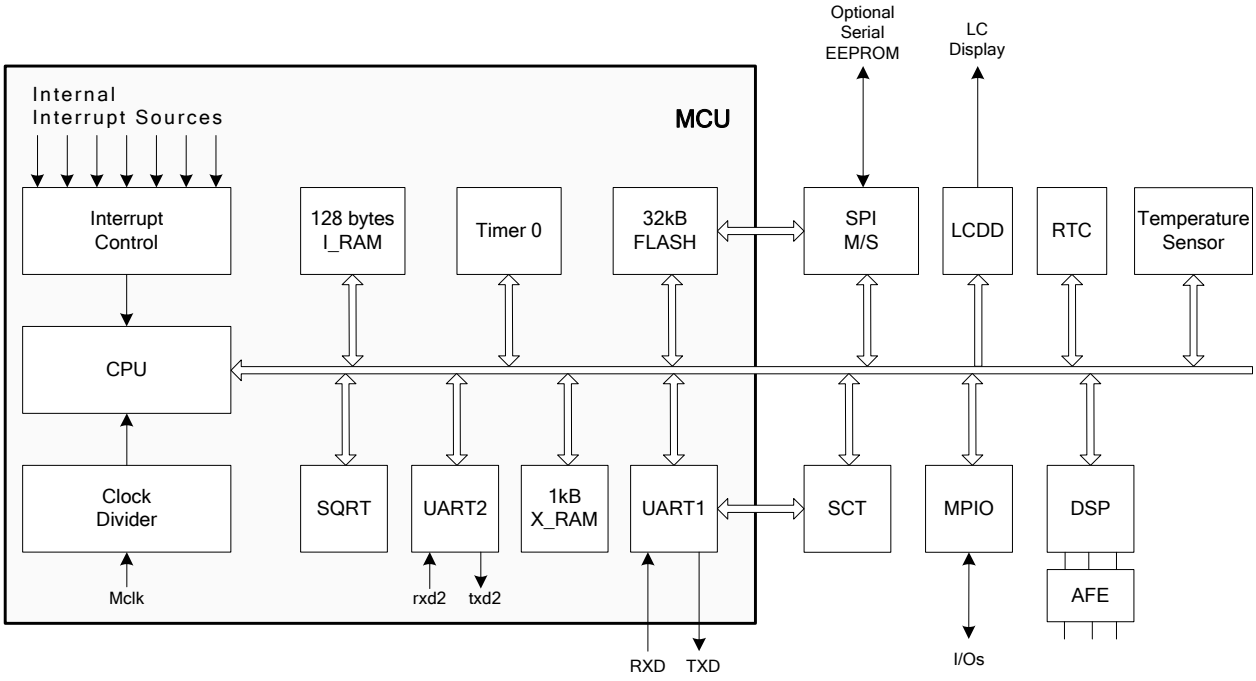


Figure 17: MCU block diagram

Legend

CPU .....8051 compatible microcontroller core
I_RAM .....128 bytes static RAM, range 00h to 7Fh of 8051
X_RAM.....1024 bytes static RAM, (extended) memory for data storage
FLASH .....32kB Flash memory, primarily for program storage, maybe used also for data
Timer 0.....16 bit timer (due to 8051 standard)
UART1 .....serial interface RS232 (due to 8051 standard) with extended baudrate generator
UART2 .....serial interface RS232 with extended baudrate generator
SQRT.....square root calculation out of 5 bytes (40 bits) input, 2.5 bytes (20 bits) output

SPI.....serial peripheral interface, used to access an external EEPROM
LCDD .....LCD driver block
RTC .....real time clock, time/data may be set via UART1 (SCT)
MPIO.....multi-purpose I/O pins, configurable inputs and outputs
DSP .....digital signal processing unit interfaces to analog front end (AFE)
AFE.....analog front end, includes amplifiers and A to D converters
SCT .....system control unit, combined with UART1 used for debugging/programming of the device

Key Features

- 8051 compatible 8 bit oriented microcontroller core
- 128 bytes of internal data memory (I_RAM)
- 32kB Flash memory
- 1kB data memory (X_RAM)
- Von Neumann architecture, shared program and data memory
- Cycle optimized compared to standard 8051, some instructions are executed in a single clock cycle
- 128 bytes of SFR range
- Standard SFRs: Timer 0, UART1 (with 16 baudrate reg.)
- Specific SFRs: UART2 (with 16 bit baudrate reg.), SQRT block
- Fully compatible 8051 instruction set including DA, MUL and DIV instruction
- 7 internal interrupt sources
- Ports P0, P1, P2, P3 are not implemented
- P0 and P2 are accessible as registers
- Register PCON is not implemented
- No idle mode via PCON
- Automatic bootload of application program after power-on reset
- 6 clock cycles per instruction (12 cycles in standard 8051)
- 1 data pointer DPTR

Instruction Set

The instruction set is fully compatible to the 8051 standard. This allows the use of commonly available software development tools for A51 Assembler, C-Compiler and code simulators. The instructions marked with the note [2)] are cycle optimised and execute in a single cycle compared to two cycles in standard 8051 controllers.

Hex Code	Mnemonic	Operands	B/C [1)]	Hex Code	Mnemonic	Operands	B/C [1)]
00	NOP		1/1	30	JNB	bit addr, code addr	3/2
01	AJMP	code addr	2/2	31	ACALL	code addr	2/2
02	LJMP	code addr	3/2	32	RETI		1/2
03	RR	A	1/1	33	RLC	A	1/1
04	INC	A	1/1	34	ADDC	A, #data	2/1
05	INC	dir	2/1	35	ADDC	A, dir	2/1
06	INC	@R0	1/1	36	ADDC	A, @R0	1/1
07	INC	@R1	1/1	37	ADDC	A, @R1	1/1
08	INC	R0	1/1	38	ADDC	A, R0	1/1
09	INC	R1	1/1	39	ADDC	A, R1	1/1
0A	INC	R2	1/1	3A	ADDC	A, R2	1/1
0B	INC	R3	1/1	3B	ADDC	A, R3	1/1
0C	INC	R4	1/1	3C	ADDC	A, R4	1/1
0D	INC	R5	1/1	3D	ADDC	A, R5	1/1
0E	INC	R6	1/1	3E	ADDC	A, R6	1/1
0F	INC	R7	1/1	3F	ADDC	A, R7	1/1
10	JBC	bit addr, code	3/2	40	JC	code addr	2/2
11	ACALL	code addr	2/2	41	AJMP	code addr	2/2
12	LCALL	code addr	3/2	42	ORL	dir, A	2/1
13	RRC	A	1/1	43	ORL	dir, #data	3/2
14	DEC	A	1/1	44	ORL	A, #data	2/1
15	DEC	dir	2/1	45	ORL	A, dir	2/1
16	DEC	@R0	1/1	46	ORL	A, @R0	1/1
17	DEC	@R1	1/1	47	ORL	A, @R1	1/1
18	DEC	R0	1/1	48	ORL	A, R0	1/1
19	DEC	R1	1/1	49	ORL	A, R1	1/1
1A	DEC	R2	1/1	4A	ORL	A, R2	1/1
1B	DEC	R3	1/1	4B	ORL	A, R3	1/1
1C	DEC	R4	1/1	4C	ORL	A, R4	1/1
1D	DEC	R5	1/1	4D	ORL	A, R5	1/1
1E	DEC	R6	1/1	4E	ORL	A, R6	1/1
1F	DEC	R7	1/1	4F	ORL	A, R7	1/1
20	JB	bit addr, code	3/2	50	JNC	code addr	2/2
21	AJMP	code addr	2/2	51	ACALL	code addr	2/2
22	RET		1/2	52	ANL	dir, A	2/1
23	RL	A	1/1	53	ANL	dir, #data	3/2
24	ADD	A, #data	1/1	54	ANL	A, #data	2/1
25	ADD	A, dir	2/1	55	ANL	A, dir	2/1
26	ADD	A, @R0	2/1	56	ANL	A, @R0	1/1
27	ADD	A, @R1	1/1	57	ANL	A, @R1	1/1
28	ADD	A, R0	1/1	58	ANL	A, R0	1/1
29	ADD	A, R1	1/1	59	ANL	A, R1	1/1
2A	ADD	A, R2	1/1	5A	ANL	A, R2	1/1
2B	ADD	A, R3	1/1	5B	ANL	A, R3	1/1
2C	ADD	A, R4	1/1	5C	ANL	A, R4	1/1
2D	ADD	A, R5	1/1	5D	ANL	A, R5	1/1
2E	ADD	A, R6	1/1	5E	ANL	A, R6	1/1
2F	ADD	A, R7	1/1	5F	ANL	A, R7	1/1

Hex Code	Mnemonic	Operands	B/C [1]	Hex Code	Mnemonic	Operands	B/C [1]
60	JZ	code addr	2/2	90	MOV	DPTR, #data	3/2
61	AJMP	code addr	2/2	91	ACALL	code addr	2/2
62	XRL	dir, A	2/1	92	MOV	bit addr, C	2/1[2]
63	XRL	dir, #data	3/2	93	MOVC	A, @A+DPTR	2/2
64	XRL	A, #data	2/1	94	SUBB	A, #data	2/1
65	XRL	A, dir	2/1	95	SUBB	A, dir	2/1
66	XRL	A, @R0	1/1	96	SUBB	A, @R0	1/1
67	XRL	A, @R1	1/1	97	SUBB	A, @R1	1/1
68	XRL	A, R0	1/1	98	SUBB	A, R0	1/1
69	XRL	A, R1	1/1	99	SUBB	A, R1	1/1
6A	XRL	A, R2	1/1	9A	SUBB	A, R2	1/1
6B	XRL	A, R3	1/1	9B	SUBB	A, R3	1/1
6C	XRL	A, R4	1/1	9C	SUBB	A, R4	1/1
6D	XRL	A, R5	1/1	9D	SUBB	A, R5	1/1
6E	XRL	A, R6	1/1	9E	SUBB	A, R6	1/1
6F	XRL	A, R7	1/1	9F	SUBB	A, R7	1/1
70	JNZ	code addr	2/2	A0	ORL	C, /bit addr	2/1[2]
71	ACALL	code addr	2/2	A1	AJMP	code addr	2/2
72	ORL	C, bit addr	2/1[2]	A2	MOV	C, bit addr	2/1
73	JMP	@A+DPTR	1/2	A3	INC	DPTR	1/1[2]
74	MOV	A, #data	2/1	A4	MUL	AB	1/4
75	MOV	dir, #data	2/1	A5	n/a	(reserved)	1/1
76	MOV	@R0, #data	1/1	A6	MOV	@R0, dir	2/1[2]
77	MOV	@R1, #data	1/1	A7	MOV	@R1, dir	2/1[2]
78	MOV	R0, #data	1/1	A8	MOV	R0, dir	2/1[2]
79	MOV	R1, #data	1/1	A9	MOV	R1, dir	2/1[2]
7A	MOV	R2, #data	1/1	AA	MOV	R2, dir	2/1[2]
7B	MOV	R3, #data	1/1	AB	MOV	R3, dir	2/1[2]
7C	MOV	R4, #data	1/1	AC	MOV	R4, dir	2/1[2]
7D	MOV	R5, #data	1/1	AD	MOV	R5, dir	2/1[2]
7E	MOV	R6, #data	1/1	AE	MOV	R6, dir	2/1[2]
7F	MOV	R7, #data	1/1	AF	MOV	R7, dir	2/1[2]
80	SJMP	code addr	2/2	B0	ANL	C, /bit addr	2/1[2]
81	AJMP	code addr	2/2	B1	ACALL	code addr	2/2
82	ANL	C, bit addr	2/1[2]	B2	CPL	bit addr	2/1
83	MOVC	A, @A+PC	2/2	B3	CPL	C	1/1
84	DIV	AB	1/4	B4	CJNE	A, #data, code	3/2
85	MOV	dir, dir	3/2	B5	CJNE	A, dir, code	3/2
86	MOV	dir, @R0	2/1[2]	B6	CJNE	@R0, #data, code	3/2
87	MOV	dir, @R1	2/1[2]	B7	CJNE	@R1, #data, code	3/2
88	MOV	dir, R0	2/1[2]	B8	CJNE	R0, #data, code	3/2
89	MOV	dir, R1	2/1[2]	B9	CJNE	R1, #data, code	3/2
8A	MOV	dir, R2	2/1[2]	BA	CJNE	R2, #data, code	3/2
8B	MOV	dir, R3	2/1[2]	BB	CJNE	R3, #data, code	3/2
8C	MOV	dir, R4	2/1[2]	BC	CJNE	R4, #data, code	3/2
8D	MOV	dir, R5	2/1[2]	BD	CJNE	R5, #data, code	3/2
8E	MOV	dir, R6	2/1[2]	BE	CJNE	R6, #data, code	3/2
8F	MOV	dir, R7	2/1[2]	BF	CJNE	R7, #data, code	3/2

Hex Code	Mnemonic	Operands	B/C [1]	Hex Code	Mnemonic	Operands	B/C [1]
C0	PUSH	dir	2/1[2]	E0	MOVX	A, @DPTR	2/2
C1	AJMP	code addr	2/2	E1	AJMP	code addr	2/2
C2	CLR	bit addr	2/1	E2	MOVX	A, @R0	2/2
C3	CLR	C	1/1	E3	MOVX	A, @R1	2/2
C4	SWAP	A	1/1	E4	CLR	A	1/1
C5	XCH	A, dir	2/1	E5	MOV	A, dir	2/1
C6	XCH	A, @R0	1/1	E6	MOV	A, @R0	1/1
C7	XCH	A, @R1	1/1	E7	MOV	A, @R1	1/1
C8	XCH	A, R0	1/1	E8	MOV	A, R0	1/1
C9	XCH	A, R1	1/1	E9	MOV	A, R1	1/1
CA	XCH	A, R2	1/1	EA	MOV	A, R2	1/1
CB	XCH	A, R3	1/1	EB	MOV	A, R3	1/1
CC	XCH	A, R4	1/1	EC	MOV	A, R4	1/1
CD	XCH	A, R5	1/1	ED	MOV	A, R5	1/1
CE	XCH	A, R6	1/1	EE	MOV	A, R6	1/1
CF	XCH	A, R7	1/1	EF	MOV	A, R7	1/1
D0	POP	dir	2/1[2]	F0	MOVX	@DPTR, A	1/2
D1	ACALL	code addr	2/2	F1	ACALL	code addr	2/2
D2	SETB	bit addr	2/1	F2	MOVX	@R0, A	1/2
D3	SETB	C	1/1	F3	MOVX	@R1, A	1/2
D4	DA	A	1/1	F4	CPL	A	1/1
D5	DJNZ	dir, code addr	3/2	F5	MOV	dir, A	2/1
D6	XCHD	A, @R0	1/1	F6	MOV	@R0, A	1/1
D7	XCHD	A, @R1	1/1	F7	MOV	@R1, A	1/1
D8	DJNZ	R0, code addr	2/2	F8	MOV	R0, A	1/1
D9	DJNZ	R1, code addr	2/2	F9	MOV	R1, A	1/1
DA	DJNZ	R2, code addr	2/2	FA	MOV	R2, A	1/1
DB	DJNZ	R3, code addr	2/2	FB	MOV	R3, A	1/1
DC	DJNZ	R4, code addr	2/2	FC	MOV	R4, A	1/1
DD	DJNZ	R5, code addr	2/2	FD	MOV	R5, A	1/1
DE	DJNZ	R6, code addr	2/2	FE	MOV	R6, A	1/1
DF	DJNZ	R7, code addr	2/2	FF	MOV	R7, A	1/1

dir ..... variable in I_RAM
code addr ... address in code memory
data ..... immediate data
bit addr..... address of a bit in bit-addressable I_RAM

Notes:
1) 'B' = number of bytes
'C' = number of cycles
2) Optimised execution in a single cycle; normally 2 cycles

Addressing Modes

The MCU comprises all standard 8051 addressing modes. For completeness they are listed here. There are five types. In two byte instructions the destination is specified first, then the source.

Mode	Examples	Notes
Register addressing	MOV A, R0	Register R0 in I_RAM one out of 4 banks selected
Direct addressing	MOV R0, A	Moves contents of A to R0
Register indirect addressing	MOV @R0, A MOVX @DPTR, A	Moves contents of A to location addressed by R0, or by DPTR
Immediate addressing	MOV R0, #data	Moves immediate #data to R0
Index addressing	MOVC A, @A+DPTR MOVC A, @A+PC	Moves contents of location addressed by A+DPTR, or A+PC to A. For reading lookup tables, applies to program memory only

Interrupt Controller

The 8051 core provides 7 interrupt sources: 2 of them are the same as in the standard 8051, the others are tied to specific internal sources. Each interrupt causes the program to jump to the corresponding interrupt vector if the interrupt is enabled in the interrupt enable register (IE). The interrupt priority can be controlled via the interrupt priority register (IP) in order to override the predefined priority, starting with IP.0 as highest. For further information on the interrupt sources refer to the appropriate chapters.

Interrupt Enable Register (IE)

Each of the interrupt sources can be individually enabled or disabled by setting the corresponding bit in the IE register. This register contains a global enable bit EA. By clearing this bit all interrupts can be disabled at once.

IE

MSB							LSB
EA	ERTC	ES2	ES	ESPI	EIOX	ET0	EDSP

Enable bit = 0 disables the interrupt
Enable bit = 1 enables the interrupt

Interrupt Priority Register (IP)

IP

MSB							LSB
-	PRTC	PS2	PS	PSPI	PIOX	PT0	PDSP

Priority bit = 1 assigns high priority
Priority bit = 0 assigns low priority

Interrupt Source	RTC	UART2	UART1	SPI	MPIO	Timer 0	DSP
Interrupt Vector	0033h	002Bh	0023h	001Bh	0013h	000Bh	0003h

Note:
Timer0 must have the highest priority in the IP register. No other interrupt should be assigned with a high priority.

Symbol	Position	Function	Priority
EA	IE.7 [1]	Disables all interrupt when 0. If EA = 1 each interrupt is individually enabled due to its enable bit.	
ERTC	IE.6	RTC real time clock, interrupt enable bit	Lowest
ES2	IE.5	UART2, serial port, interrupt enable bit	
ES	IE.4 [1]	UART1, serial port, interrupt enable bit	
ESPI	IE.3	SPI serial port, interrupt enable bit	
EIOX	IE.2	MPIO external pin, interrupt enable bit	
ET0	IE.1 [1]	Timer 0, interrupt enable bit	
EDSP	IE.0	DSP data available	Highest

Note:

1) Standard 8051 bits

Interrupt Priorities

Each interrupt source can be individually assigned one of two priority levels. A low priority interrupt can always be interrupted by a higher-priority interrupt, but not by another low priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.
If the corresponding IP bit is set then this interrupt is serviced first if another interrupt request occurs at the same time where the IP bit is zero.
Interrupt on the same priority level are serviced due to the internal polling sequence starting with DSP highest down to RTC lowest.

Symbol	Position	Function	Source Flags
	IP.7	-	
PRTC	IP.6	Real time clock, priority bit	TSA, STF, A1F, A2F
PS2	IP.5	UART2 serial port, priority bit	RI, TI
PS	IP.4 [1]	UART1 serial port, priority bit	RI, TI
PSPI	IP.3	SPI serial port, priority bit	ITRA
PIOX	IP.2	MPIO external pin, priority bit	in Status 0 / Status 1
PT0	IP.1 [1]	Timer 0, priority bit	TF0
PDSP	IP.0	DSP priority bit	dai, alarm

Note:

1) Standard 8051 bits

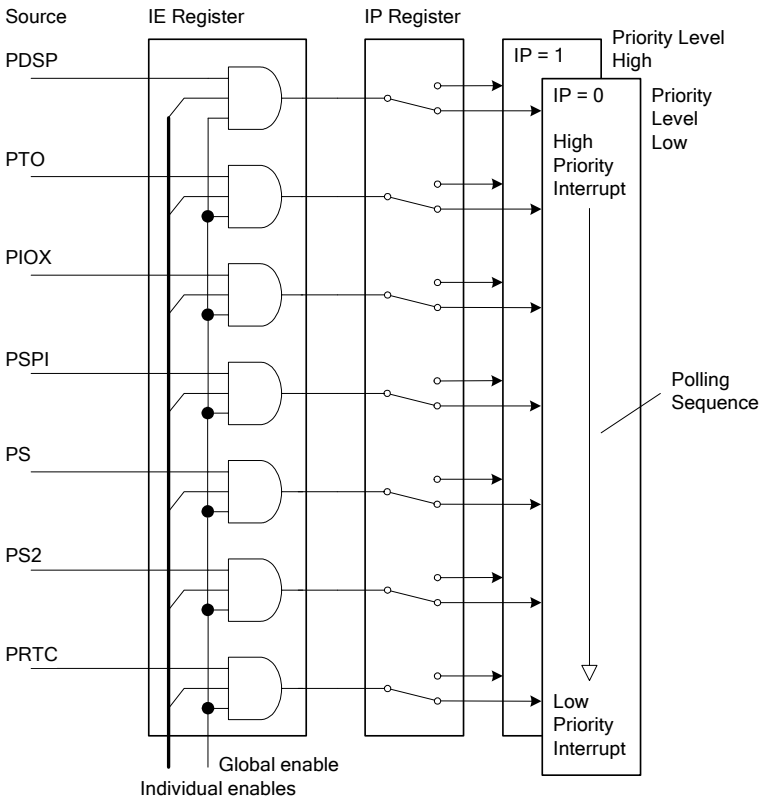
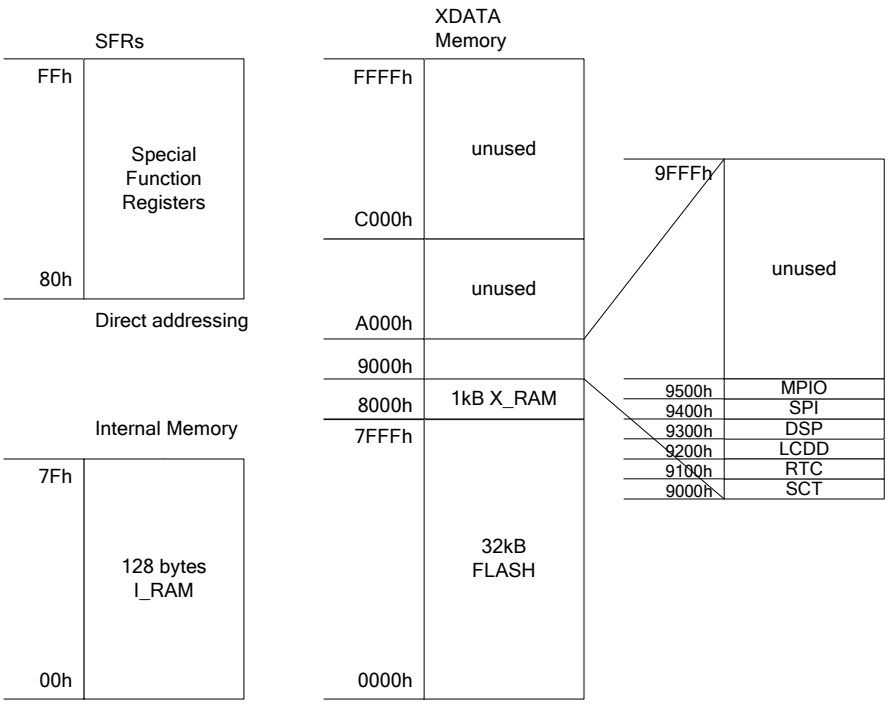


Figure 18: Interrupt control system

Memory Maps

The 8051 MCU is configured as Von Neumann architecture merging program and data range into one 64kB address space. This space is completely accessible via MOVX and partly accessible via MOVC (0000h – 5FFFh). Besides, there is the typical 8051 structure with 128 bytes of internal memory (I_RAM) and the special function registers (SFRs) also in a 128 byte address space.



FLASH Memory

The Flash memory shares address and output data lines with the X_RAM. 32kB out of 64kB addressable memory are used: 0000h – 7FFFh for program data storage.

Data Memory (X_RAM) and Block Interfaces

The following table shows the start (and stop) addresses for the X_RAM and the block interfaces. These locations can be accessed with the MOVX instruction.

Start Address	Stop Address	Contents
8000h	83FFh	X_RAM
9000h	9007h	SCT
9100h	9134h	RTC
9180h	9184h	WDT
9200h	9220h	LCDD
9300h	9338h	DSPREG (MDR/SREG)
9400h	9403h	EEP_SPI
9500h	951Fh	MPIO
9600h	9604h	TEMPSENS
9700h	9701h	FLASH (STAT/ATTACK)

Detailed Memory map:

Address	Contents	Address	Contents	Address	Contents	Address	Contents	
8000h	…	83FFh	X_RAM					X_RAM
9000h	-	9001h	enable signals	9002h	clkdiv[2:0]	9003h	-	SCT
9100h	Seconds/VL	9101h	Minutes	9102h	Hours	9103h	Days	RTC
9104h	Weekdays	9105h	Months/Cent.	9106h	Years	9107h	-	
9108h	-	9109h	-	910Ah	-	910Bh	-	
910Ch	-	910Dh	-	910Eh	-	910Fh	-	
9110h	Cont./Status1	9111h	Cont./Status2	9112h	Sec.Tim.B 0	9113h	Sec.Tim.B 1	
9114h	Min.Alarm 1	9115h	Hour Alarm 1	9116h	Day Alarm 1	9117h	Mon. Alarm 1	
9118h	YearsAlarm 1	9119h	Min.Alarm 2	911Ah	Hour Alarm 2	911Bh	Day Alarm 2	
911Ch	Mon. Alarm 2	911Dh	YearsAlarm 2	911Eh	-	911Fh	-	
9130h	DivReg B 0	9131h	DivReg B 1	9132h	DivReg B 2	9133h	Freq. Trim	
9180h	WDTE	9181h	WDTCLK					WDT
9200h	reg1[7:0]	9201h	reg1[15:8]	9202h	reg1[23:16]	9203h	reg1[31:24]	LCDD
9204h	reg1[39:32]	9205h	reg1[47:40]	9206h	reg1[55:48]	9207h	reg1[63:56]	
9208h	reg1[71:64]	9209h	reg1[79:72]	920Ah	reg1[87:80]	920Bh	reg1[95:88]	
920Ch	-	920Dh	-	920Eh	-	920Fh	-	
9210h	reg2[7:0]	9211h	reg2[15:8]	9212h	reg2[23:16]	9213h	reg2[31:24]	
9214h	reg2[39:32]	9215h	reg2[47:40]	9216h	reg2[55:48]	9217h	reg2[63:56]	
9218h	reg2[71:64]	9219h	reg2[79:72]	921Ah	reg2[87:80]	921Bh	reg2[95:88]	
921Ch	-	921Dh	-	921Eh	use_reg	921Fh	selvlcd[2:0]	

Address	Contents	Address	Contents	Address	Contents	Address	Contents	
9220h	lcdd_pd							
9300h	samptoend 0	9301h	samptoend 1	9302h	np[7:0]	9303h	np[15:8]	DSP
9304h	np[23:16]	9305h	np[31:24]	9306h	sos_v[7:0]	9307h	sos_v[15:8]	
9308h	sos_v[23:16]	9309h	sos_v[31:24]	930Ah	sos_v[35:32]	930Bh	sos_i1[7:0]	
930Ch	sos_i1[15:8]	930Dh	sos_i1[23:16]	930Eh	sos_i1[31:24]	930Fh	sos_i1[39:32]	
9310h	sos_i1[47:40]	9311h	sos_i1[53:48]	9312h	sos_i2[7:0]	9313h	sos_i2[15:8]	
9314h	sos_i2[23:16]	9315h	sos_i2[31:24]	9316h	sos_i2[39:32]	9317h	sos_i2[47:40]	
9318h	sos_i2[53:48]	9319h	-	931Ah	-	931Bh	-	
931Ch	-	931Dh	-	931Eh	-	931Fh	-	
9320h	pcorr_i1[7:0]	9321h	pcorr_i1[8]	9322h	pcorr_i2[7:0]	9323h	pcorr_i2[8]	
9324h	cal_v[7:0]	9325h	cal_v[15:8]	9326h	cal_i1[7:0]	9327h	cal_i1[15:8]	
9328h	cal_i2[7:0]	9329h	cal_i2[15:8]	932Ah	pulselev_i1 0	932Bh	pulselev_i1 1	
932Ch	pulselev_i1 2	932Dh	pulselev_i2 0	932Eh	pulselev_i2 1	932Fh	pulselev_i2 2	
9330h	mconst[3:0]	9331h	-	9332h	nsamp[7:0]	9333h	nsamp[15:8]	
9334h	vconst[7:0]	9335h	vconst[13:8]	9336h	Select	9337h	Gains	
9338h	Status	9339h	-	933Ah	-	933Bh	-	
9400h	SSPCON	9401h	SSPCLKDIV	9402h	SSPSTAT	9403h	SSPBUF	SPI2
9500h	make_irq0	9501h	make_irq1	9502h	out_mux0	9503h	out_mux1	MPIO
9504h	out_mux2	9505h	set_en0	9506h	set_en1	9507h	sel_drv0	
9508h	sel_drv1	9509h	sel_pupd0	950Ah	sel_pupd1	950Bh	sel_in	
950Ch	sel_refp	950Dh	in0	950Eh	in1	950Fh	out0	
9510h	out1	9511h	out2	9512h	out3	9513h	out4	
9514h	out5	9515h	out6	9516h	out7	9517h	out8	
9518h	out9	9519h	out10	951Ah	out11	951Bh	pcnt0	
951Ch	pcnt1	951Dh	pcnt2	951Eh	Status0	951Fh	Status1	
9600h	TS_Status	9601h	TS_Result0	9602h	TS_Result1	9603h	TS_OffsetCorr0	TS
9604h	TS_OffsetCorr1							
9700h	FLASH_STAT	9701h	FLASH_ATTACK	9702h	-	9703h	-	FLASH

Internal Memory (I_RAM)

128 bytes of I_RAM are provided, which can be accessed via 3 address modes.

- All memory 00h to 7Fh is directly addressable.
- 00h to 1Fh are register addressable in four banks. Bank switching is done in PSW (Program Status Word).
- 20h to 2Fh are bit addressable, which means that each bit of these registers can be set/cleared separately.

I_RAM Locations																
78h		79h		7Ah		7Bh		7Ch		7Dh		7Eh		7Fh		
70h		71h		72h		73h		74h		75h		76h		77h		
68h		69h		6Ah		6Bh		6Ch		6Dh		6Eh		6Fh		
60h		61h		62h		63h		64h		65h		66h		67h		
58h		59h		5Ah		5Bh		5Ch		5Dh		5Eh		5Fh		
50h		51h		52h		53h		54h		55h		56h		57h		
48h		49h		4Ah		4Bh		4Ch		4Dh		4Eh		4Fh		
40h		41h		42h		43h		44h		45h		46h		47h		
38h		39h		3Ah		3Bh		3Ch		3Dh		3Eh		3Fh		
30h		31h		32h		33h		34h		35h		36h		37h		
28h	bit 40-47	29h	bit 48-4F	2Ah	bit 50-57	2Bh	bit 58-5F	2Ch	bit 60-67	2Dh	bit 68-6F	2Eh	bit 70-77	2Fh	bit 78-7F	
20h	bit 00-07	21h	bit 08-0F	22h	bit 10-17	23h	bit 18-1F	24h	bit 20-27	25h	bit 28-2F	26h	bit 30-37	27h	bit 38-3F	
18h	R0	19h	R1	1Ah	R2	1Bh	R3	1Ch	R4	1Dh	R5	1Eh	R6	1Fh	R7	
10h	R0	11h	R1	12h	R2	13h	R3	14h	R4	15h	R5	16h	R6	17h	R7	
08h	R0	09h	R1	0Ah	R2	0Bh	R3	0Ch	R4	0Dh	R5	0Eh	R6	0Fh	R7	
00h	R0	01h	R1	02h	R2	03h	R3	04h	R4	05h	R5	06h	R6	07h	R7	

The first 4 x 8 bytes of the internal memory can be addressed via instructions using the register addressing mode (register bank 0, 1, 2, 3). The following 16 bytes (16 x 8 = 128 bits, address 20h to 2Fh) can be addressed via instructions using the direct-bit addressing mode. The address space from 30h to 7Fh is accessible via the direct addressing mode only. Gray-shaded R0 and R1 registers can be used for register indirect addressing.

Special Function Registers (SFR)

The following table shows the locations of the Special Function Registers. SFRs in bold style are original 8051 registers. SFRs in italic style are additional registers specific to the AS8267 / AS8268 ICs.

SFR Locations																
F8h		F9h		FAh		FBh		FCh		FDh		FEh		FFh		
F0h	**B**	F1h		F2h		F3h		F4h		F5h		F6h		F7h		
E8h	*SQRTIN0*	E9h	*SQRTIN1*	EAh	*SQRTIN2*	EBh	*SQRTIN3*	ECh	*SQRTIN4*	EDh	*SQRTOUT0*	EEh	*SQRTOUT1*	EFh	*SQRTOUT2*	
E0h	**ACC**	E1h		E2h		E3h		E4h		E5h		E6h		E7h		
D8h		D9h		DAh		DBh		DCh		DDh		DEh		DFh		
D0h	**PSW**	D1h		D2h		D3h		D4h		D5h		D6h		D7h		
C8h		C9h		CAh		CBh		CCh		CDh		CEh		CFh		
C0h	*SCON2*	C1h	*SBUF2*	C2h	*SBAUDL2*	C3h	*SBAUDH2*	C4h		C5h		C6h		C7h		
B8h	**IP**	B9h		BAh		BBh		BCh		BDh		BEh		BFh		
B0h	*SOVR2*	B1h		B2h		B3h		B4h		B5h		B6h		B7h		
A8h	**IE**	A9h		AAh		ABh		ACh		ADh		AEh		AFh		
A0h	**P2**	A1h		A2h		A3h		A4h		A5h		A6h		A7h		
98h	**SCON**	99h	**SBUF**	9Ah	*SBAUDL*	9Bh	*SBAUDH*	9Ch		9Dh		9Eh		9Fh		
90h	*SOVR*	91h		92h		93h		94h		95h		96h		97h		
88h	**TCON**	89h	**TMOD**	8Ah	**TL0**	8Bh		8Ch	**TH0**	8Dh		8Eh	*T0PRE*	8Fh		
80h	**P0**	81h	**SP**	82h	**DPL**	83h	**DPH**	84h		85h		86h		87h		

128 bytes of SFR address space is available using the direct addressing mode. The following table describes the use of the register bytes:

Symbol	Register Name	Address	Notes
Standard Registers			
ACC	Accumulator	E0h	
B	B Register	F0h	
PSW	Program Status Word	D0h	
SP	Stack Pointer	81h	
DPTR	Data Pointer 2 Bytes		
DPL	Low Byte	82h	
DPH	High Byte	83h	
P0	Port 0	80h	
P2	Port 2	A0h	
IP	Interrupt Priority Control	B8h	
IE	Interrupt Enable Control	A8h	
TMOD	Timer Mode Control	89h	
TCON	Timer Control	88h	
TH0	Timer 0 High Byte	8Ch	
TL0	Timer 0 Low Byte	8Ah	
SCON	Serial Control (UART1)	98h	
SBUF	Serial Data Buffer (UART1)	99h	
Custom Registers			
T0PRE	Timer 0 Prescaler	8Eh	
SOVR	Serial Overflow (UART1)	90h	
SBaudL	Serial Baudrate Low (UART1)	9Ah	
SBaudH	Serial Baudrate High (UART1)	9Bh	
SCON2	UART2 Control	C0h	
SBUF2	UART2 Serial Data Buffer	C1h	

Symbol	Register Name	Address	Notes
SBaudL2	UART2 Baudrate Low	C2h	
SBaudH2	UART2 Baudrate High	C3h	
SOVR2	UART2 Overflow	B0h	
SQRTIN0	Square Root Input [7:0]	E8h	Writing to this location triggers the squareroot calculation
SQRTIN1	Square Root Input [15:8]	E9h	
SQRTIN2	Square Root Input [23:16]	EAh	
SQRTIN3	Square Root Input [31:24]	EBh	
SQRTIN4	Square Root Input [39:32]	ECh	
SQRTOUT0	Square Root Output [7:0]	EDh	
SQRTOUT1	Square Root Output [15:8]	EEh	
SQRTOUT2	Square Root Output [23:16]	EFh	

Notes:
1) Ports P1 and P3 do not exist.
2) Timer 1 is not implemented (and the related SFRs).
3) Ports P0 and P2 are not connected to pins.
   P0 and P2 can be used as a register in general.
   However, P2 can be used for X_RAM access, when '@Ri' is used in the register indirect addressing mode (with Ri being either R0 or R1). In that case P2 will form the higher byte of the X_RAM address.
4) IE/IP: The sources for the interrupts are defined in interrupt controller section.
5) TCON, TMOD, TH0, TLO described in section Timer 0.
6) SCON, SBUF, SBaudL, SBaudH, SOVR are related to UART1 described in the UART section.
7) SCON2, SBUF2, SBaudL2, SBaudH2, SOVR2 are related to UART2 described in the UART2 section.

Squareroot Block (SQRT)

This SQRT block calculates the square root of a 40 bit input value (mapped to 5 eight bit input registers).
The output is a 20 bit number which is mapped to 3 eight bit output registers.
The calculation starts immediately after the least significant byte has been written (= address E8h).
For the square root calculation the Gypsi- or radicand algorithm is used, which produces one bit per clock cycle. Thus after 20 cycles the result is available in the SQRTOUT[2:0] registers.
Note: The interrupt signal is not connected to the interrupt controller of the MCU, because the result is available after a defined period of 4 machine cycles. The programmer has to take care for the correct timing. For instance, 4 NOP instructions must be inserted before reading out the result.
When writing SQRTIN[39:36] are don't care.
When reading SQRTOUT[23:20] those bits equal zero.

Data Registers

SFR-Address	Name	Description
E8h	SQRTIN0	Input value[7:0]
E9h	SQRTIN1	Input value[15:8]
EAh	SQRTIN2	Input value[23:16]
EBh	SQRTIN3	Input value[31:24]
ECh	SQRTIN4	Input value[39:32]
EDh	SQRTOUT0	Output value[7:0]
EEh	SQRTOUT1	Output value[15:8]
EFh	SQRTOUT2	Output value[19:16]

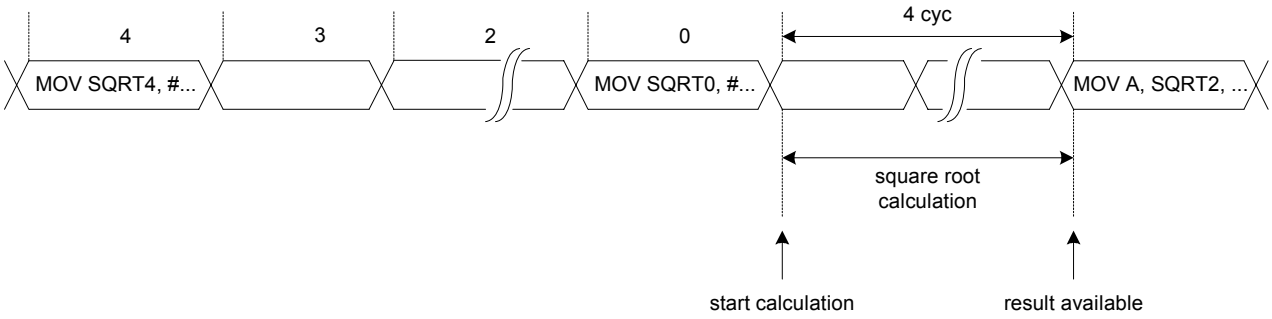


Figure 19: Timing diagram

During the time of calculation data must not be overwritten. As soon as the register SQRT0 is written, the calculation sequence is retriggered and the result is calculated from the latest contents of the 5 input registers.

Boot Loader (BOOTLOAD)

After power-up the boot loader checks if the program memory (32kB Flash) is blank or if there is a program available.
In case there is no program stored (no program length stored at 7FFFh and 7FFEh) the boot loader generates 'SJMP$' (Hex code: 80FEh) instruction address 0000h.
This guarantees a well defined behaviour after power-on.
In case there is a program stored in the Flash memory the boot load block loads also security information from the upmost page of the Flash memory.
After the boot load the MCU will start to work.
The loaded program will be executed.

Watchdog Timer (WDT)

A watchdog timer is provided on-chip to automatically initiate a system reset if a 'hold-off' signal is not detected within a predefined timeout period, by the watchdog.

The watchdog timer consists of a programmable timer driven either by the Mclk (main oscillator output frequency), or the MCU clock (microcontroller unit clock). The watchdog timer timeout period is dependent upon the programming of the WDTCLK register. When the watchdog times out, a reset signal is generated which is OR-ed with the main system reset. Thus a watchdog timer reset is identical to a power on reset.

If the watchdog timer function is required, the watchdog is enabled by setting the WDTE register LSB (Bit 0). As soon as this bit is enabled, the program must periodically access the WDTCLK register (either read or write) to prevent the watchdog timer from timeout and thus resetting the device.

Register Name	Address	Reset Value	Description
WDTE	9180h	xxxx.xxx0b	Enables or disables the watchdog timer function 0: watchdog disabled 1: watchdog enabled
WDTCLK[1:0]	9181h	xxxx.xxx00b	A read or write access clears the watchdog timer. Writing bits [1:0] selects the clock source.

x .....Don't care

Watchdog Timer Enable Register (WDTE)

MSB							LSB
-	-	-	-	-	-	-	WDTE0

Bit	Symbol	Function
7	-	Not used
6	-	Not used
5	-	Not used
4	-	Not used
3	-	Not used
2	-	Not used

Bit	Symbol	Function
1	-	Not used
0	WDTE0	Disables and enables the watchdog timer function 0: watchdog disabled 1: watchdog enabled

The watchdog timer has a selectable counter length of 18 bit, 20 bit or 22 bit for the Mclk and 18 bits for the mcu_clk. It should be noted that while the Mclk has a fixed frequency, depending on the crystal frequency, the MCU clock is programmable, being divisible by 1 to 128, in binary steps (see MCUCLKDIV Register ('mcu_clk')). The timeout periods below assume the Mclk = 3.579545MHz (fixed crystal frequency).

Watchdog Timer Clock Register (WDTCLK)

MSB							LSB
-	-	-	-	-	-	WDTCLK1	WDTCLK0

Bit	Symbol	Function				
7	-	Not used				
6	-	Not used				
5	-	Not used				
4	-	Not used				
3	-	Not used				
2	-	Not used				
1	WDTCLK1	Watchdog timeout period (Mclk = 3.579545MHz)	**Clock Source**	**Timeout Period (ms)**	**Bit1**	**Bit0**
			Mclk – default after reset	73.2	0	0
			Mclk	292.8	0	1
0	WDTCLK0		Mclk	1171.2	1	0
			Mcu_clk (div=1)	73.2	1	1
			Mcuclk (div=128)	9300		

2nd UART (UART2)

An additional serial interface, UART2 is provided for debugging purposes. UART2 is accessible via two of the multi-purpose I/Os (MPIO). The UART2 is functionally identical to UART1. The SFR addresses are defined as follows:

Register Name	Address	Description
SCON2	C0h	Serial port control register – see Serial Interface – UART1 for details.
SBUF2	C1h	Serial port buffer register – see Serial Interface – UART1 for details.
SBAUDL2	C2h	Baudrate reload register – Low address
SBAUDH2	C3h	Baudrate reload register – High address
SOVR2	B0h	'Serial overflow' register, which indicates when data in SBUF has been overwritten before being read. The flag is the LSB with the other 7 bits all being 0.

Below is an example how to configure the ports IO7 and IO6 as UART2s txd2 (IO7) and rxd2 (IO6) pins.

```
;-------------------------------------------------------------------------------
; Configure UART2 to the pins IO6 and IO7 with the Baudrate of 19200 Baud:
;-------------------------------------------------------------------------------
; map txd2 = IO7
; map rxd2 = IO6
;-------------------------------------------------------------------------------
xdata mem: OUTMUX1     (9503h) <- 80h ; maps txd2 to IO7
xdata mem: SET_ENO     (9505h) <- 80h ; enable output IO7
xdata mem: SEL_PUPDO   (9509h) <- 40h ; enable pullup for IO6
xdata mem: SEL_IN      (950Bh) <- 05h ; map rxd2 to IO6
idata mem: SBAUDL2     (0C2)   <- 11h ; set Baudrate register low
idata mem: SBAUDH2     (0C3)   <- 00h ; set Baudrate register high
idata mem: SCON2       (0C0)   <- 50h ; setup UART2 serial port for Rx and Tx.
;-------------------------------------------------------------------------------

;-------------------------------------------------------------------------------
; program fragment for enabling uart2 for serial communication.
;
; sfr locations --
;
SCON2             EQU    0C0h        ; Serial 2 Control Register
SBUF2             EQU    0C1h        ; Serial 2 Port Register
SBAUDL2           EQU    0C2h        ; Serial 2 Baudload LowByte
SBAUDH2           EQU    0C3h        ; Serial 2 Baudload HighByte
;
; variables --
;
BaudrateLO        EQU    11          ; Baudrate Value for 19200 baud,
BaudrateHI        EQU    0           ; mcu_clk = 3.58MHz
;
; memory map for the uart2 configurations --
;
OUTMUX1           EQU    09503H      ; need to be set as 0x80
SET_ENO           EQU    09505H      ; need to be set as 0x80
SEL_PUPDO         EQU    09509H      ; need to be set as 0x40
SEL_IN            EQU    0950BH      ; need to be set as 0x05
;
; instruction code fragment
; ...
               MOV      IE,#0A0h          ; enable serial interrupt UART2 0xA0
               MOV      DPTR,#OUTMUX1     ; 09503H <- 80h ; maps txd2 to IO7
               MOV      A,#080h
               MOVX     @DPTR,A
               MOV      DPTR,#SET_ENO     ; 09505H <- 80h ; enable output IO7
               MOV      A,#080h
               MOVX     @DPTR,A
               MOV      DPTR,#SEL_PUPDO   ; 09509H <- 40h ; enable pullup for IO6
               MOV      A,#040h
               MOVX     @DPTR,A
               MOV      DPTR,#SEL_IN      ; 0950BH <- 05h ; map rxd2 to IO6
               MOV      A,#005h
               MOVX     @DPTR,A
               MOV      SBAUDL2,#BaudrateLO ; set Baudrate (16 bits)
               MOV      SBAUDH2,#BaudrateHI
               MOV      SCON2,#050h      ;  Set up uart2 serial port for Rx and Tx.
; ...
;-------------------------------------------------------------------------------
```

Timer 0

There is only Timer 0 present, Timer 1 is not implemented except for some flags in the TCON register, which are also used for Timer 0. Furthermore, there is no counter mode available as the inputs T1 and INTO of the standard 8051 are not mapped to external pins. The connection of Timer 0 in each of its four operating modes is shown below.

There are five special function registers (SFR) related to the Timer 0:

Register Name	Address	Description
TMOD	89h	Timer mode register
TCON	88h	Timer control register
T0PRE	8Eh	Timer 0 8 bit prescaler register
TH0	8Ch	Timer 0 higher byte
TL0	8Ah	Timer 0 lower byte

TMOD

MSB							LSB
0	0	0	0	GATE	C/T_N	M1	M0

Bit	Symbol	Function				
7	-	Not used				
6	-	Not used				
5	-	Not used				
4	-	Not used				
3	GATE	Has no effect on Timer 0 operation can be used as register bit				
2	C/T_N	Acts like an enable signal				
1	M1	Mode select	Mode	Description	Bit1	Bit0
			0	13 bit timer (MCS-48 compatible)	0	0
			1	Same as mode 0 but 16 bit timer	0	1
0	M0		2	Configures Timer 0 as 8 bit autoreload timer. Overflow from TL0 sets TF0 and reloads TL0 with the value of TH0.	1	0
			3	Two 8 bit timers, TL0 controlled by Timer 0 standard bits, TH0 controlled by Timer 1 control bits but no interrupt	1	1

TCON

MSB							LSB
TF1	TR1	TF0	TR0	-	-	-	-

Bit	Symbol	Function
7	TF1	Timer 0 (Mode 3) TH0 overrun flag, generates no interrupt, flag can be polled by software.
6	TR1	Timer 0 (Mode 3) TH0 enable flag, TH0 runs if ‘1’ in all other modes the flag has no effect.
5	TF0	Timer 0 overrun flag, generates an interrupt. Flag is cleared by hardware when the processor jumps to the interrupt routine
4	TR0	Timer 0 run control bit. Timer runs if ‘1’. Cleared/set by software.
3	-	Not used
2	-	Not used
1	-	Not used
0	-	Not used

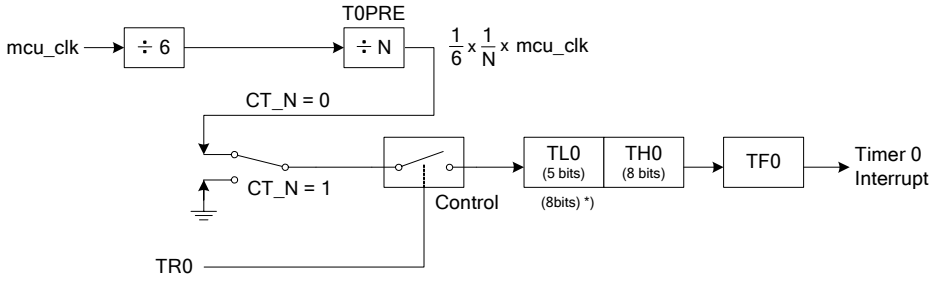


Figure 20: Timer 0 Mode 0 and Mode 1*): 13 bit counter

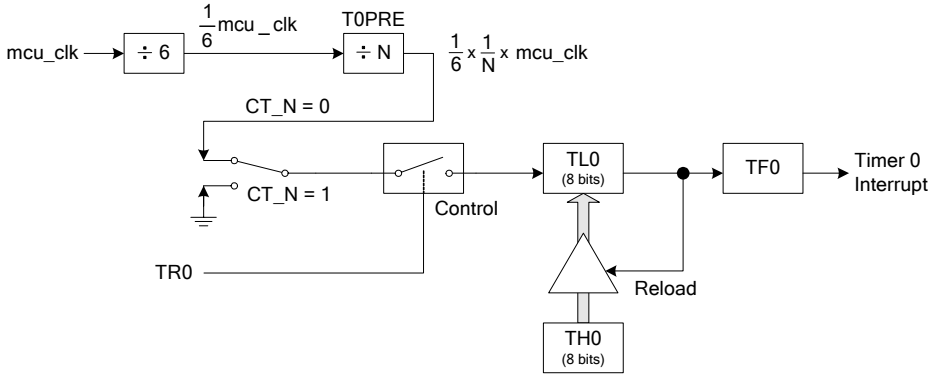


Figure 21: Timer 0 Mode 2: 8 bit counter

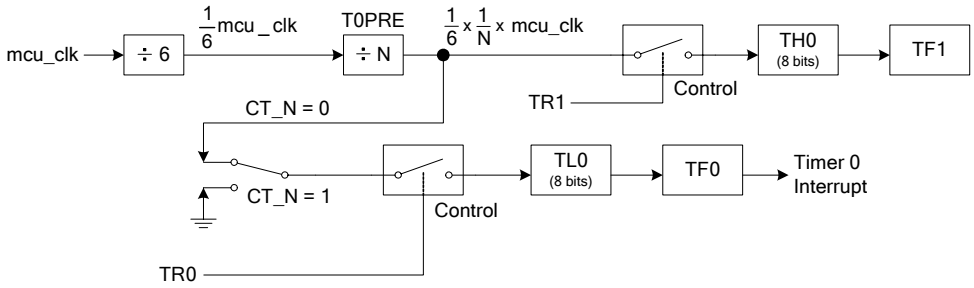


Figure 22: Timer 0 Mode 3: two 8 bit counters

T0PRE

Unlike the standard 8051 there is a 8-bit prescaler register available for timer 0. Values of 0x00 (default after reset) and 0x01 do not have any effect. For all other values the timer input frequency is divided according to the value (ranging from 2 up to 255).

8.9 System Control (SCT)

The system control is responsible for handling different modes of operation such as normal mode (metering functions) and test mode.
The clock generation and reset control are also available in System Control (SCT).

Modes of Operation

Power off

In this mode everything is off including the 'System Timing and RTC' block, provided that no battery is connected to VDD_BAT or the battery is discharged. Nothing happens.

RTC on, Rest of the Chip off

In this mode the 'System Timing and RTC' block is supplied by a battery, the RTC is working, but no interrupts are generated.
At the moment the battery is inserted, a power-on reset just for the RTC will be generated. The reset will be set to 0 after the first clock edges arrive.

Power-up Phase

When the power is switched on (for the 'rest' of the chip), there is a power-on reset first and the reset is held until the BOOTLOAD block has finished operation. The BOOTLOAD block will load information from the upmost page (program available, security) from the internal Flash memory.
After BOOTLOAD the MCU will start executing the program in the Flash memory. It is assumed that at the beginning of the program various system parameters are set (sel_i, sel_v, sel_p, creep etc.)

FLASH not programmed, must be loaded from outside

If the Flash does not contain a program the MCU will run in idle mode. It is necessary to write a program to the Flash next.

Reset Chip: Externally triggered BOOTLOAD

When a (new) program has been written to the Flash it will be necessary to trigger a new BOOTLOAD sequence. This can be done by generating a chip reset. After the related command has been detected by the SCT on the UART1 interface the reset/boot sequence will start. Next the MCU will run the program from the beginning.
This command can also be used for a simple reset.

Normal Operation

The MCU is working through its program, access to certain blocks/functions may be done via the UART interfaces. For example, it may be required to read data from an external EEPROM or from the RTC block. During these operations the MCU is not reset!

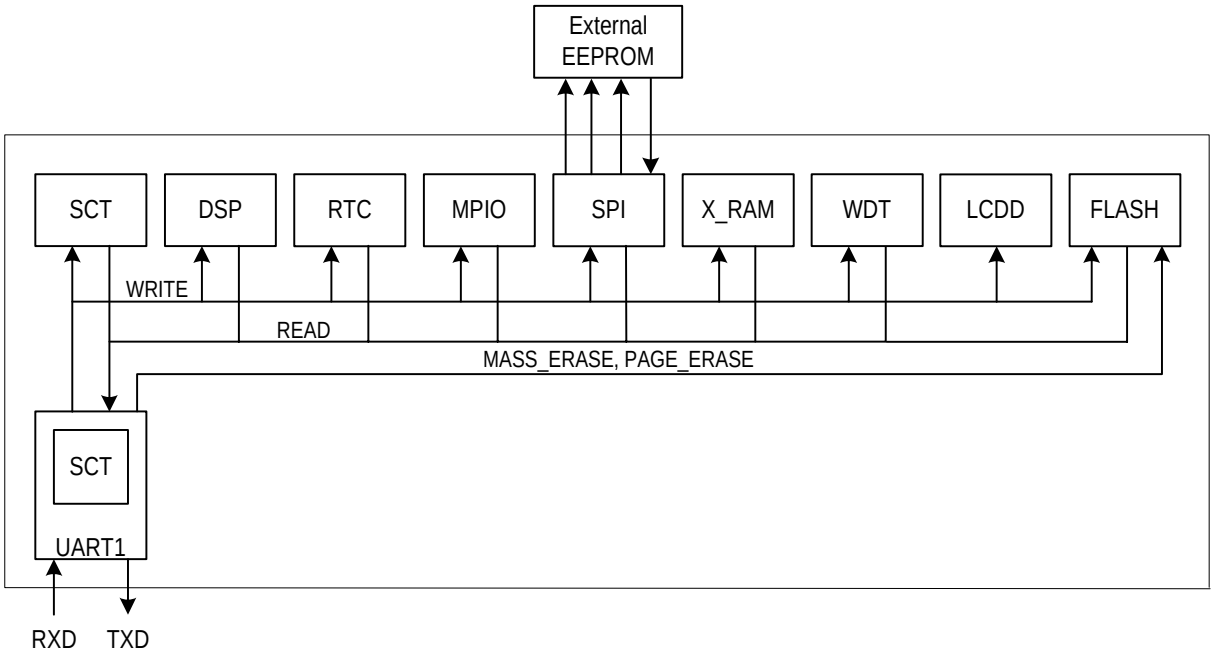
Program Debugging

Program debugging can be done using a so-called monitor program, which may communicate with a PC using the UART2 interface or the UART1 interface in direct access mode. In the AS8267 / AS8268 the SPI interface (SPI2 and MCU) can also be used to access the whole XDATA address space.

Note: Direct access mode ('dam' register bit) turns off the command interpreter. If the MCU program performs this operation, it must be able to clear the dam bit after the operation, or the SPI2 access must have the ability to do this. If both possibilities are blocked there is no way out and the device is permanently locked.
To prevent this situation UART2 is recommended to be used as debug interface.

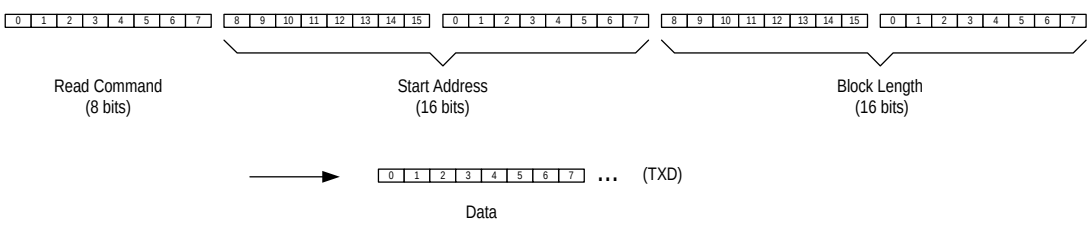
Read from XDATA Address

This command is used for Flash and external EEPROM read access.
Mainly for evaluation purposes it is possible to read all 64k of the XDATA address space. This includes all registers, the X_RAM memory and the Flash memory. A dedicated command is reserved for this. The following diagram shows the main blocks involved here:



Transmission Protocol:

In order to make the data transfer easier for the system control a defined protocol is used for talking to the UART1, where also the length of the data to be read from the XDATA address space is specified at the beginning of the transmission. It looks like this:



Notes:

1) When ‘enable_crc’ is set to 1 a 16-bit checksum word will be sent after the data stream. It can be used to validate the received data.

Before accessing the Flash a request has to be sent (see

2) Arbitration)

Write to XDATA Address

This command is used for Flash and external EEPROM write access.
For evaluation, but also for setting the RTC it will be required to write to registers located in the XDATA address space. Also for this an SCT command is prepared. The diagram in the section before shows the main blocks involved. Again the whole XDATA range of 64k bytes is visible to the WRITE_X instruction. However, for the Flash memory (including addresses 0000h to 7FFFh) there are some considerations to take when programming the device. Besides the ‘byte write’ command, there are also ‘page write’, ‘page_erase’ and ‘mass erase’.

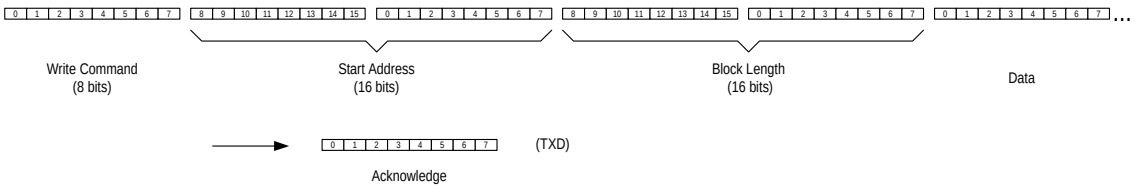
Notes:

Before accessing the Flash a request has to be sent (see

1) Arbitration)
2) In principle it is possible that a value, which has been modified using this write-command, immediately gets overwritten by the MCU. Therefore this command has to be used in an intelligent way.

Transmission Protocol:

In order to make the data transfer easier for the system control a defined protocol is used for talking to the UART1, where also the length of the data to be written to the XDATA address space is specified at the beginning of the transmission. It looks like this:



Notes:

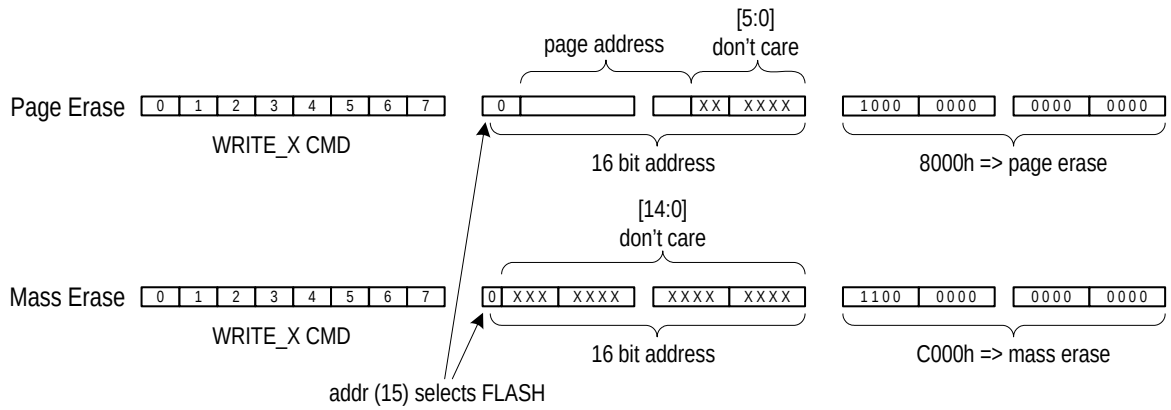
1) When ‘enable_crc’ is set to 0 the UART1 only sends back the acknowledge word (FAh).
2) When ‘enable_crc’ is set to 1 a 16-bit checksum has to be transferred to the UART1 at the end of the data stream. The SCT will calculate the checksum and depending on the result it will send back the acknowledge word (FAh) or the not-acknowledge word (FBh).

Transmission Protocol for FLASH BYTE PROGRAM and FLASH PAGE PROGRAM:

The protocol is the same as for a manual write operation to a register. The only difference is that after reaching the desired block length, a Flash program process is triggered, which takes 6ms maximum. The device detects automatically a Flash programming process when selecting an address lower than 8000h. The user is responsible for the correct programming and handling of Flash pages.

Transmission Protocol for FLASH PAGE ERASE and FLASH MASS ERASE:

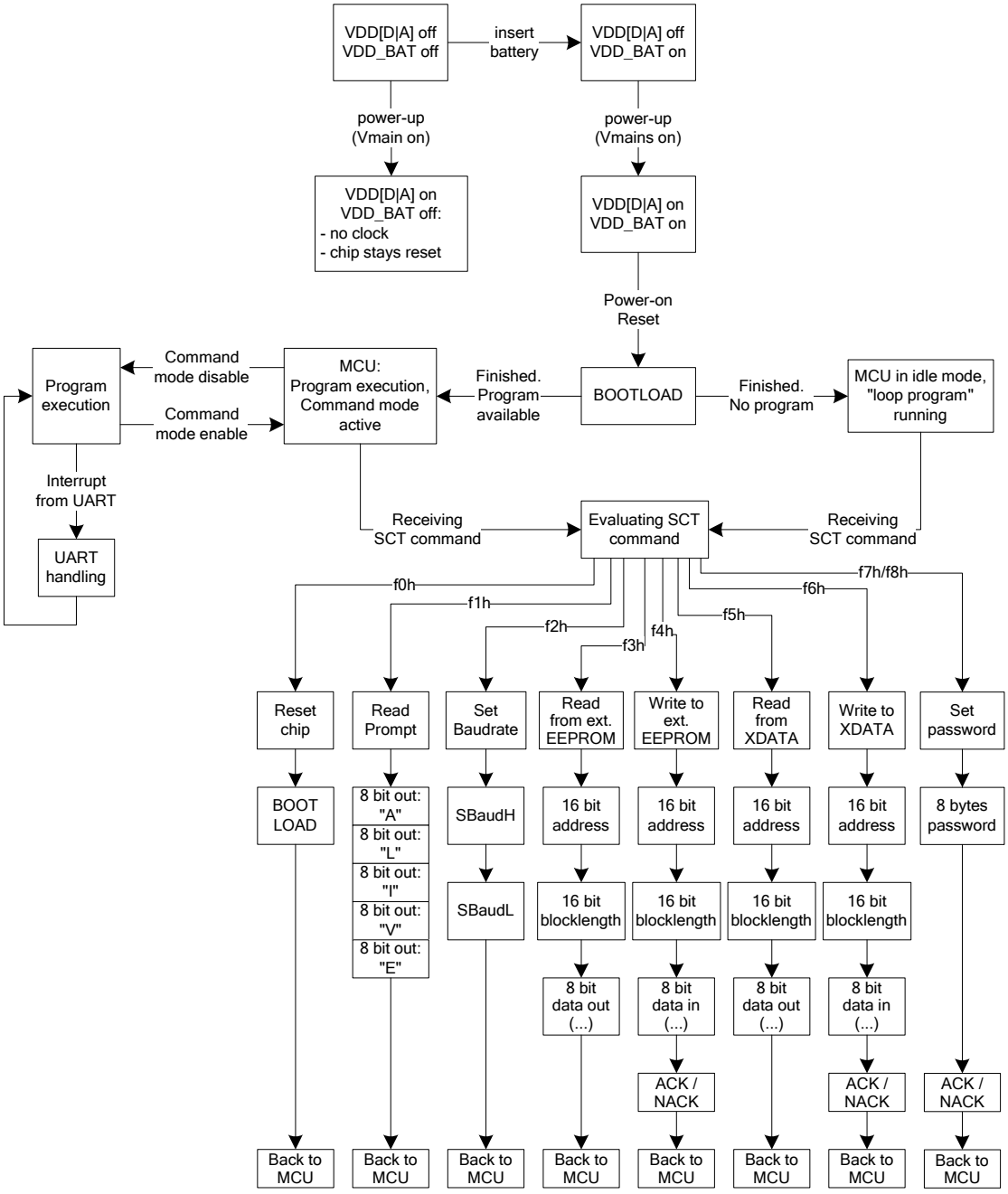
For ‘page erase’ and ‘mass erase’ there is also deployed the WRITE_X command.



For ‘page erase’ it is required to define the page to be erased. The erase code is 8000h in the block length field.

For mass erase, the address field is recommended to be set to 0000h and the block length field to C000h.

Flow Diagram of Operational Modes



Command Interpreter

The command interpreter is continuously looking at the UART1 input and detects, if a command has been sent, i.e. a specific byte that is defined to initiate a dedicated mode of operation (see the flow diagram above).
The commands have been specified to lie outside the “normal” ASCII range. All codes not specified within the following table can directly be transferred to the MCU without any interference by the SCT.

Command Name	Code	Description
SOFT_RESET	F0h	Resets the chip and initiates a BOOTLOAD sequence, then the MCU program is started.
RD_PROMPT	F1h	The chip sends a specific signature, “ALIVE”. This can be used to test the UART1/SCT interface.
SBAUD	F2h	Makes it possible to set the UART1 baudrate from outside the chip by directly accessing the SFRs “SBaudL” and SBaudH”. Default setting: 11 (3.5795MHz crystal and 19200 Baud)
READ_EE	F3h	Enables reading of data from ext. EEPROM
WRITE_EE	F4h	Data can be written to the ext. EEPROM
READ_X	F5h	Data from the XDATA address space can be read.
WRITE_X	F6h	Data can be written to any location in the XDATA address space.
SET_PW	F7h	Set password
SET_PW1	F8h	Set password and enable attack counter
ACK	FAh	Acknowledge
NACK	FBh	No Acknowledge

SCT Registers

The system control (SCT) registers provide for the setting of various enables signals and selection of the MCU clock (MCU_CLK) frequency.

Register Name	Address	Reset Value	Description
-	9000h	-	Not used
enable signals	9001h	000b	See table below
mcuclkdiv[2:0]	9002h	000b	See table below

Enable Signals Register

The enable signals register includes power-down signals and other control signals.

MSB							LSB
-	sel_spi2	enable_crc	u2clkoff	u1clkoff	dam	sdmi2_pd	afe_pd

Bit	Symbol	Function
7	-	Not used
6	sel_spi2	Selects the SPI path 0: selects path to SPI_Flash (slave mode) 1: selects path to SPI2 (master mode)
5	enable_crc	Enables checksum functionality during read/write accesses to XDATA address space or EEPROM. If enabled, a 16-bit checksum word (see Note below) is sent after the data, which is checked by the SCT (in case of 'write') or can be checked by an external component (in case of 'read'). 0: checksum disabled 1: checksum enabled
4	u2clkoff	Switches off the UART2 clock (which is also running at the highest system frequency 'mclk'): 0: clock active 1: clock switched off
3	u1clkoff	Switches off the UART1 clock (which is running at the highest system frequency 'mclk'): 0: clock active 1: clock switched off
2	dam	Select direct access mode for UART1; in case of 'dam' input data will no longer be interpreted as commands. 0: direct access mode off 1: dam on
1	sdmi2_pd	Set power-down for current channel 2 (active high) 0: no power-down 1: I2 powered down
0	afe_pd	Set power-down for the entire analog front end (AFE) 0: AFE powered up 1: AFE powered down

Note: The checksum is calculated using the following formula: $g(x) = x^{16} + x^{12} + x^5 + 1$
In dam mode no interrupt will be triggered, therefore the SCON register has to be polled.

MCUCLKDIV Register ('mcu_clk')

The MCU clock divider (mcuclkdiv) divides down the main clock (Mclk) which is the output from the low power oscillator.

Division of the mcu_clk frequency is provided to enable low power operating modes, for example when the AS8267 / AS8268 ICs are in a battery operating mode, when VDDD is connected to a battery.

MSB							LSB
-	-	-	-	-	mcuclkdiv.2	mcuclkdiv.1	mcuclkdiv.0

Bit	Symbol	Function
7	-	Not used
6	-	Not used
5	-	Not used
4	-	Not used
3	-	Not used

Bit	Symbol	Function	Bit2	Bit1	Bit0	Division
2	mcuclkdiv.2	These bits set the mcu_clk frequency by dividing down the main clock (Mclk).	0	0	0	Mclk : 1
			0	0	1	Mclk : 2
1	mcuclkdiv.1		0	1	0	Mclk : 4
			0	1	1	Mclk : 8
			1	0	0	Mclk : 16
0	mcuclkdiv.0		1	0	1	Mclk : 32
			1	1	0	Mclk : 64
			1	1	1	Mclk : 128

8.10 Serial Interface – UART1

- Extended version of the standard 8051 UART1
- SBUF and SCON are compatible with standard 8051
- Built-in 16 bit baudrate generator (SBAUDH, SBAUDL)
- Additional SOVR receiver overflow indicator register

UART1 is used to communicate externally. UART1 requires only two pins to receive and transmit information. UART1 is compatible to the Serial Interface of the 8051 microcontroller family, with the exception of the baudrate generation. UART1 is functionally identical to UART2. Thus the instructions below are also valid for UART2.

UART1 is segmented into three main functional blocks, namely *Baudrate*, *Transmission* and *Reception*, as shown in the block diagram below:

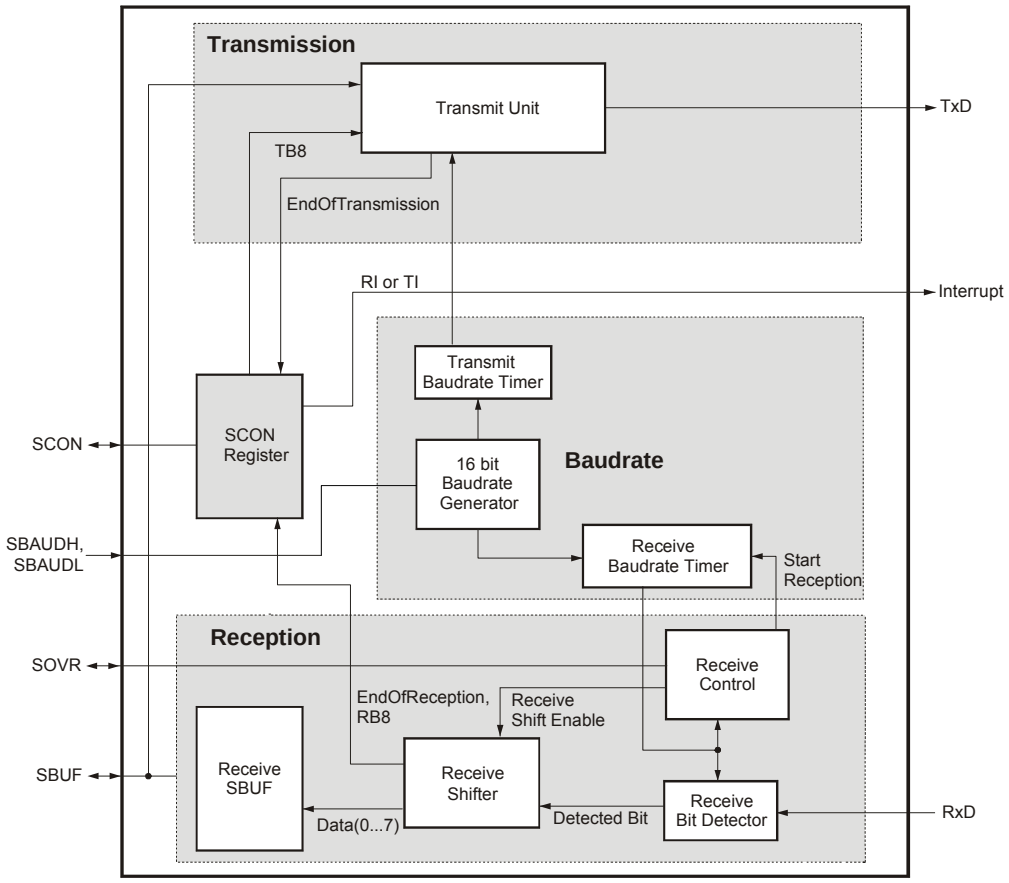


Figure 23: UART1 block diagram

There is no direct dependency on osc clock (Mode 0, 3). Instead there is a built-in 16 bit wide baudrate generator for higher flexibility.

UART is dedicated to the SCT block for writing to and reading from other functional blocks such as RTC, LCDD; besides, it is used for selection of different modes of operation.

SFRs of UART1

There are five special function registers dedicated to the UART1.

Register Name	Address	Description	Read/Write from MCU
SCON	98h	Serial port control register	read & write
SBUF	99h	Serial port buffer register	read & write (separate)
SBAUDL	9Ah	Baudrate reload register – Low address	write only
SBAUDH	9Bh	Baudrate reload register – High address	write only
SOVR	90h	Serial receive buffer overflow register	read & write, only one bit (=bit0) available

SOVR Register

Serial receive buffer overflow register. If data is received before it has been read out of SBUF then the bit SOV is set. It can the cleared by software. All other bits of SOVR are 0.

SOVR

MSB							LSB
0	0	0	0	0	0	0	SOV

Note:
Overflow flag. If ‘1’ then a receiver buffer overflow occurred. The old buffer value has been overwritten by new incoming data. Set by overflow, cleared by MCU.

SCON Register

The SFR Serial Port Control Register (SCON) is used to configure the UART1 and to check the status of the transmission.

SCON

MSB							LSB
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

<table>
<tr><th>Bit</th><th>Symbol</th><th colspan="4">Function</th></tr>
<tr><td rowspan="4">7
6</td><td rowspan="4">SM0
SM1</td><td rowspan="4">Mode select flags</td><td>Mode</td><td>Bit7</td><td>Bit6</td></tr>
<tr><td>Mode 0: same as Mode 1, (Mode 0 is not implemented due to standard 8051)</td><td>0</td><td>0</td></tr>
<tr><td>Mode 1: 8-bit UART1, variable Baudrate. - The serial transmission is set to 8 data bits. However up to 10 bits can be sent at port TxD and received at port RxD: start bit (always ‘0’), eight data bits (LSB first), and a stop bit (always ‘1’). The value of a received stop bit is transferred to SCON.RB8 and can be evaluated by the software.</td><td>0</td><td>1</td></tr>
<tr><td>Mode 2: 9-bit UART1, variable Baudrate. - The serial transmission is set to 9 data bits. However up to 11 bits can be sent at port TxD and received at port RxD: start bit (always ‘0’), nine data bits (LSB first), and a stop bit (always ‘1’). The value of SCON.TB8 is used for transmitting the ninth data bit (usually as parity bit). The value of the received ninth data bit is transferred to SCON.RB8 and can be evaluated by the software.
Mode 3: 9-bit UART1, variable Baudrate. – same as Mode 2.</td><td>1
1</td><td>0
1</td></tr>
<tr><td>5</td><td>SM2</td><td colspan="4">Mode Select Flag 2: Enables the multiprocessor communications feature in Mode 2.
Mode 0: SM2 is not used.
Mode 1: When SM2=’1’, RI is not set and SBUF is not loaded if the received stop bit is ‘0’.
Mode 2: When SM2=’1’, RI is not set and SBUF is not loaded if the received ninth data bit is ‘0’. Please refer also to section Multiprocessor Communications.</td></tr>
<tr><td>4</td><td>REN</td><td colspan="4">Receiver Enable Flag. With REN=’0’ the receiver is disabled, otherwise enabled. REN is to be set and cleared by the software.</td></tr>
<tr><td>3</td><td>TB8</td><td colspan="4">Value of the ninth data bit to be sent when in mode 2.</td></tr>
<tr><td>2</td><td>RB8</td><td colspan="4">Value of the ninth data bit received when in mode 2 or value of the stop bit received when in mode 1. Not used in mode 0.</td></tr>
<tr><td>1</td><td>TI</td><td colspan="4">Transmit Interrupt Flag. This flag is set by the UART1 at the end of transmitting. In mode 0 flag TI is set at the end of the eighth data bit, in all others modes at the beginning of the stop bit. Flag TI must be cleared by the software.</td></tr>
<tr><td>0</td><td>RI</td><td colspan="4">Receive Interrupt Flag. This flag is set by the hardware at the end of receiving. In mode 0 flag RI is set at the end of the eighth data bit, in mode 1 at the middle of the stop bit, and in mode 2 at the middle of the ninth data bit. Flag RI must be cleared by the software.</td></tr>
</table>

Note:

1) Mode Select Flags 0/1: These bits are used to select one of four transmission modes. In all four modes the baudrate is determined by the Baudrate Generator.

SBUF Registers

The 8-bit register SBUF is the data buffer register which actually consists of two registers for both transmitting and receiving data. Both are accessed by the same address SBUF. A write access to SBUF is redirected into the internal register TransmitSBUF, a read access to SBUF is redirected to the internal register ReceiveSBUF. Remark: The (optional) ninth data bit is defined in SCON.TB8/RB8.

SBUF

MSB							LSB
Data 7	Data 6	Data 5	Data 4	Data 3	Data 2	Data 1	Data 0

Note:
The SBUF register is split up within the UART1 into the internal registers TransmitSBUF (when writing to the SFR) and ReceiveSBUF (when reading from the SFR)

A write access to SBUF starts a transmission, according to the selected mode. A write access during an ongoing transmission results in discarding the byte without disturbing the process of transmission. If there is a series of bytes to be transmitted, the software has to wait until the previous byte has been sent (SCON.TI = '1'), before writing to SBUF. The shift sequence (serialization) is handled by means of the internal register TransmitSBUF that holds up to 12 bits (depending on the mode used): hardcoded '1'-bit + start bit + 8 data bits + optional ninth data bit + stop bit. During transmitting the content of TransmitSBUF is shift right, thus transmission is done with LSB first.

A read access to SBUF delivers the latest byte received by the UART1. Bit SCON.RI has to be cleared (to '0') by the software after fetching a byte from SBUF, thus enabling the UART1 to receive further bytes. If SCON.RI is not '0' when a new byte is received, the new byte will be discarded (and thus is lost) and SBUF will keep its old value.

SBAUDH, SBAUDL Baudrate Reload Registers

	MSB							LSB
SBAUDL	BR7	BF6	BR5	BR4	BR3	BR2	BR1	BR0
SBAUDH	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8

Note:
The SBAUDL and SBAUDH are merged into a 16 bit reload value:
SBAUDL = Baudrate value (7:0)
SBAUDH = Baudrate value (15:8)

Baudrate Generator

Unlike the original 8051 architecture, the UART1 incorporates a built-in baudrate generator. The baudrate is generated by a counter, which is decremented every clock cycle. When reaching the value 0, the counter is automatically reloaded. The reload value is a programmable value stored in the 16 bit register formed by SBAUDH and SBAUDL. A serial bit (during transmit and receive) is further divided into 16 time slices for accurate sampling. Due to the full-duplex operation there is a separate Transmit Baudrate Timer und Receive Baudrate Timer implemented for this task.

Baudrate Reload Register

The following table shows some selected values to be loaded to the BRReloadRegister (SBaudH + SBaudL) at a given clock frequency that may be used with the AS8267 / AS8268 ICs. The smaller the value, the more difficult it is to meet a demanded baudrate within a given tolerance. If the error is greater than 5% the baudrate is not appropriate for error-free communication.

Baudrate [Baud]	3.00MHz	error [%]	3.58MHz (3579545Hz)	error [%]	4.00MHz	error [%]
110	1704	0.0267	2033	0.0082	2272	0.0120
150	1249	0	1491	-0.0320	1666	0.0200

Baudrate [Baud]	3.00MHz	error [%]	3.58MHz (3579545Hz)	error [%]	4.00MHz	error [%]
300	624	0	745	0.0351	832	-0.0400
600	312	0.1597	372	0.0350	416	0.0799
1200	155	-0.1603	185	-0.2337	207	-0.1603
2400	77	-0.1603	92	-0.2337	103	-0.1603
4800	38	-0.1603	46	0.8326	51	-0.1603
9600	19	2.3438	22	-1.3232	25	-0.1603
19200	9	2.3438	11	2.8986	12	-0.1603
38400	4	2.3438	5	2.8986	(6)	→6.9940
57600	(2)	→-8.5069	3	2.8986	(3)	→-8.5069
76800	(1)	→-22.0703	2	2.8986	(2)	→-8.5069
115200	(1)	→18.6198	1	2.8986	(1)	→-8.5069

Below there are the formulas for calculating Baudrate and BaudrateReloadRegister, but also the error. You have to divide through 16 because the serial bit is further divided into 16 time slices.

$$\text{Baudrate} = \frac{\text{ClockFrequency}}{16 \times (1 + \text{BaudrateReloadRegister})}$$

$$\text{BaudrateReloadRegister} = \frac{\text{ClockFrequency}}{16 \times \text{Baudrate}} - 1$$

$$\text{error} = \frac{\text{desired\_baudrate} - \frac{\text{ClockFrequency}}{16 \times (1 + \text{BaudrateReloadRegister})}}{\text{desired\_baudrate}}$$

Transmission

A write access to register SBUF invokes the transmission of a byte. If there is already an ongoing transmission then the written byte is discarded. At the end of transmission flag SCON.TI is set, indicating the software that the next byte can be written to SBUF.

Reception

The process of receiving is initiated by setting SCON.REN to '1' and SCON.RI to '0' by software. After reception the UART1 sets SCON.RI to '1' and the data bits can be fetched from SBUF. Each data bit of the serial data stream is probed three times (in the middle of the bit time) to achieve noise immunity.

Interrupt

Per default the UART1 is operated in the 'command mode' (as described in section SCT) and an interrupt is asserted to the MCU except when a command is detected. The UART1 can also be used in the direct access mode (bit dam = '1' in SCT register 9001h). This setting allows the MCU to operate the UART1 as defined in the standard 8051 configuration.

The UART1 asserts an interrupt whenever flag SCON.RI is ‘1’ or SCON.TI is ‘1’. These flags are set if a successful receive or transmit operation has taken place. The flags to SCON.RI and SCON.TI must be cleared by software. The MCU program branches to the interrupt routine if the serial interrupt is enabled in the IE register, with IE.4 (= ES) = ‘1’. Since SCON.RI and SCON.TI are linked together (logic-or), there is a common interrupt service routine for both transmitting and receiving. The interrupt service routine has to decide which event triggered the interrupt request (by querying the flags RI and TI). It is important to clear the flags before leaving the interrupt service routine.

Multiprocessor Communications

Mode 2 has a special provision for multiprocessor communications. In this mode, a 9th data bit is received and goes into RB8. Then a stop bit follows.
The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:
When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the ninth bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

Modes

The UART1 can be used in two different modes: mode 0 (= mode 1) and mode 2 (= mode 3). The mode selection is due the bits SM0, SM1 in the SCON register.

Mode 0 and 1

8 bit UART1 with variable baudrate controlled by the baudrate generator.

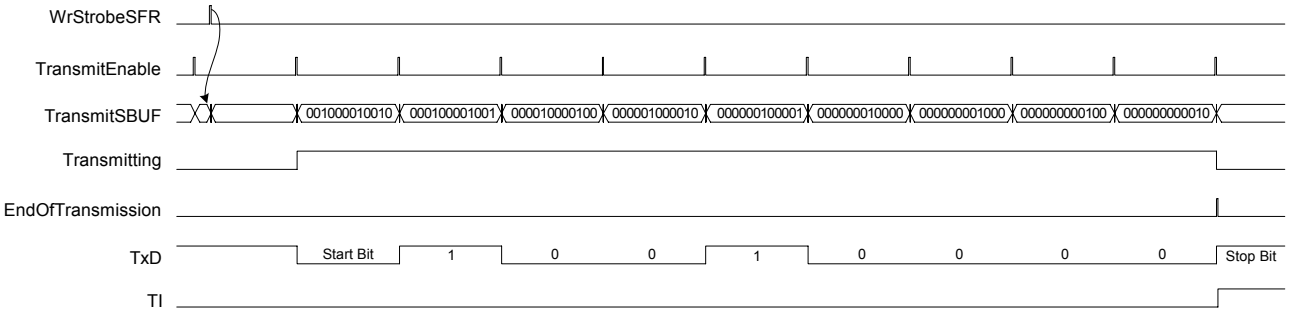


Figure 24: Transmitting in mode 1: here ‘09h’ is sent. The resulting bit stream on the TxD line is: start bit (=‘0’) + ‘10010000’, for LSB is sent first.

The process of transmitting is initiated by writing to SBUF. The byte written to SBUF is held in register TransmitSBUF. The transmission starts with the next 1-pulse on the internal signal TransmitEnable. Output TxD is driven with a start bit (‘0’), eight data bits with the LSB first shifted out from TransmitSBUF, and a stop bit (‘1’). At

begin of the stop bit the internal signal EndOfTransmission is activated, causing flag SCON.TI going to high and thus indicating the end of the transmission.

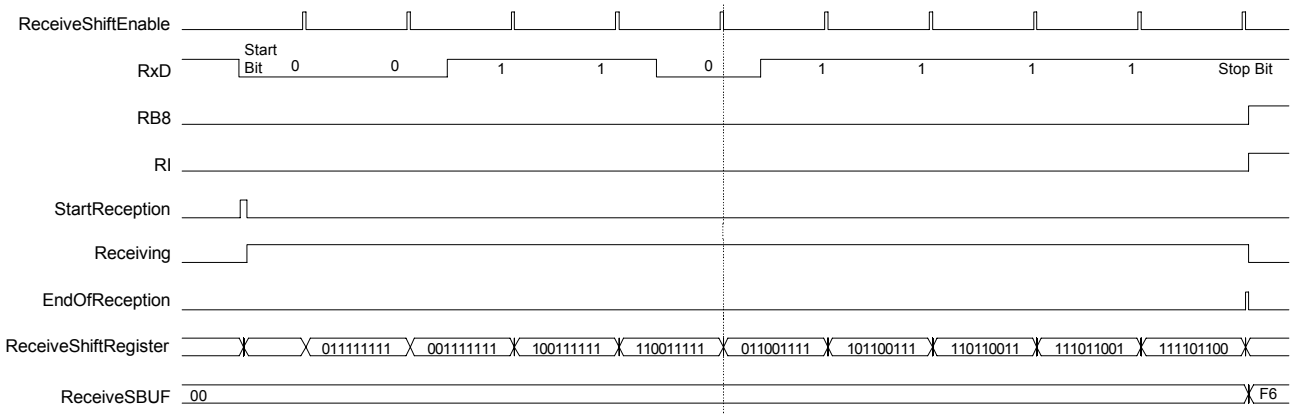


Figure 25: Receiving in mode 1 and mode 2: Here the bit stream '0'+'01101111' is received (see signal ReceivedDataBit), that is: start bit + F6*h*. The start bit is the first 0-pulse of signal RxD, that is when signal ReceivingStartbit is active.

Receiving is only possible when SCON.REN = '1'. The process of receiving is started with a falling edge on RxD (internal signal StartReception is activated) and controlled by a 4-bit counter, that means a bit time is divided into 16 time slices. The counter is reset when identifying a falling edge on RxD and is consequently synchronized. The value of RxD is probed three times at the counter stage 6, 7, and 8 (counter range is from 0 to 15). The final value (ReceivedData-Bit) is determined by majority. The multiple probing ensures a more robust serial connection.

At counter = 9 the received bit is transferred into the shift register (ReceiveShiftRegister). If the first received bit (stop bit) is not '0', then the process is aborted and the UART1 waits for the next falling edge on RxD. Due to this procedure all data packets with an invalid start bit are automatically discarded.

When receiving the stop bit (EndOfReception = '1') the following condition is checked:

```
SCON.RI='0' and (SCON.SM2='0' or Received_Stop_Bit='1')
```

If this condition is true, then all eight data bits are transferred to ReceiveSBUF, the stop bit is written to SCON.RB8, and SCON.RI is set to '1'. Otherwise all the received data is discarded and the receiver waits for the next falling edge on RxD.

Mode 2 and 3

9 bit UART1 with variable baudrate controlled by the baudrate generator.
Mode 2 is very similar to mode 1 except that nine data bits are processed. The subsequent text deals only the differences to mode 1. Mode 3 is the same as Mode 2.

The ninth data bit during transmission is taken from SCON.TB8 and is sent after the eight bits from SBUF.

When receiving the ninth data bit (EndOfReception = '1') the following condition is checked:

```
SCON.RI='0' and (SCON.SM2='0' or Ninth_Data_Bit='1')
```

If this condition is true, then all eight data bits are transferred to ReceiveSBUF, the ninth data bit is transferred to SCON.RB8, and SCON.RI is set to '1'. Otherwise all the received data is discarded and the receiver waits for the next falling edge on RxD.

Assembler Code

The following code fragments demonstrate the programming of the UART1.

Adjusting the Baudrate (for all modes)

```
       SBL equ 9AH             ; Serial BaudrateReload LowByte
       SBH equ 9BH             ; Serial BaudrateReload HighByte

       mov SBL,#38             ; 9600 baud, 6MHz
       mov SBH,#0
```

Using Mode 0

```
       mov SCON,#00H           ; mode 0, REN=0, RI=0, TI=0
       mov A,#53H
       clr TI                  ; clear transmit flag
       mov SBUF,A              ; transmit 53H in mode 0
wait:
       jnb TI,wait             ; wait until data is sent

       mov SCON,#10H           ; mode 0, REN=1 - start reception
       clr REN                 ; REN=0
wait:
       jnb RI,wait             ; wait until data is received
       mov A,SBUF              ; move received byte into the accu
```

Transmitting in Mode 0

```
       mov SCON,#50H           ; mode 1, REN=1, RI=0, TI=0
       mov A,#53H
       clr TI                  ; clear transmit flag
       mov SBUF,A              ; transmit 53H in mode 1
wait:
       jnb TI,wait             ; wait until data is sent
```

Receiving in Mode 1 (only bytes with valid stop bit)

```
       mov SCON,#70H           ; mode 1, SM2=1, REN=1, RI=0, TI=0
wait:
       jnb RI,wait             ; wait until data is received
       clr RI                  ; enable another reception
       mov A,SBUF              ; move received byte to accu
```

Transmitting in Mode 2 (ninth data bit as parity bit)

```
       mov A,#0A4h             ; move data to accu
       mov C,P                 ; parity information to carry flag
       mov TB8,C               ; parity information to ninth data bit
       mov SBUF,A              ; transmit A4H in mode 2
wait:
       jnb TI,wait             ; wait until data is sent
```

Interrupt Based Receiving

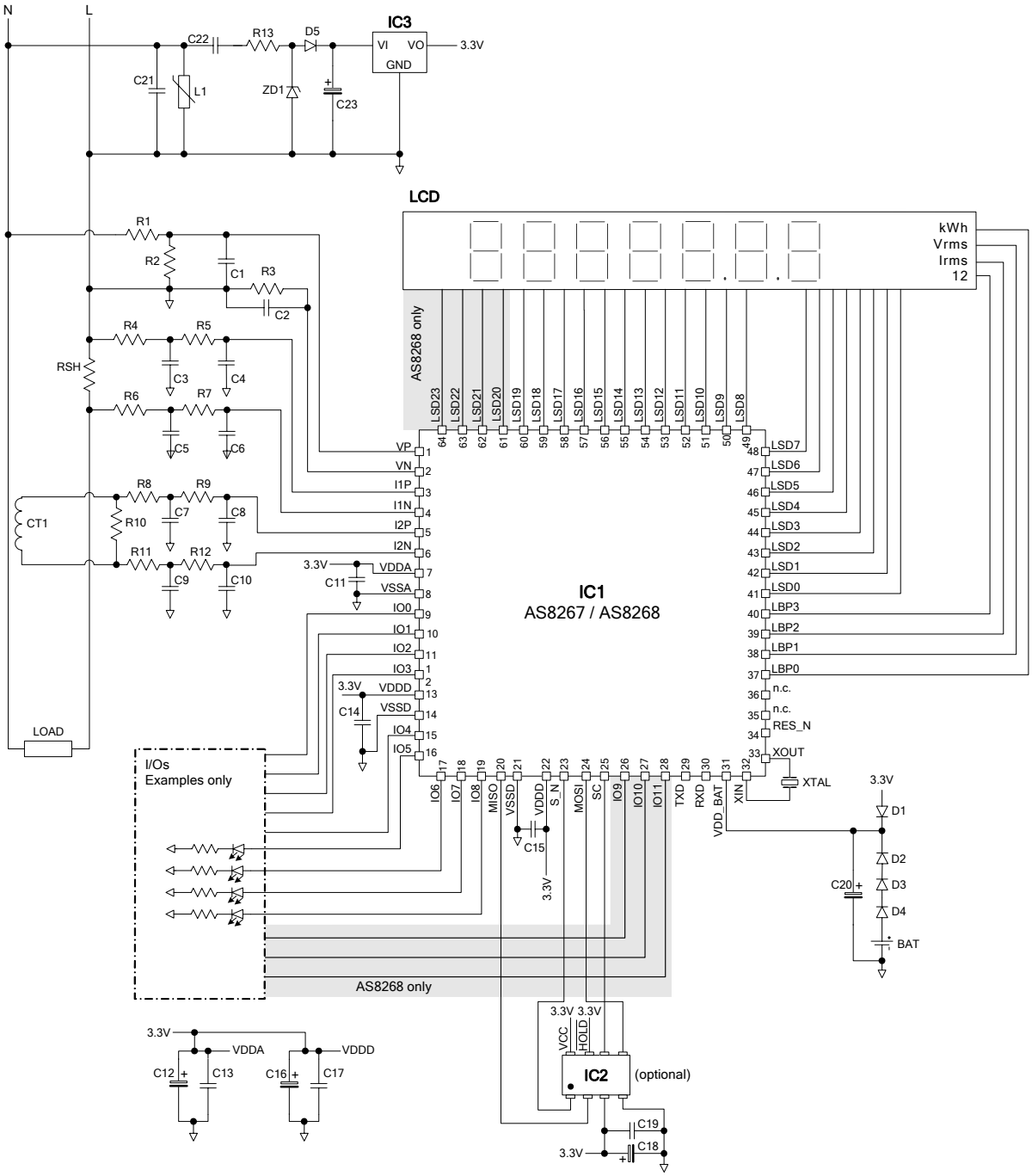
```
       org 0h                  ; reset vector
```

```
       ljmp program_start

       org 023h                ; serial interrupt vector
       ljmp SerialInterrupt

       org 100                 ; begin of main program
SerialInterrupt:
       clr RI                  ; clear the RI bit (since we know that was
                               ; the bit that caused the interrupt)
       mov P1, SBUF            ; move the received data out to port one
       reti

program_start:
       setb EA                 ; enable interrupts generally
       setb ES                 ; enable serial interrupts

       mov SCON,#50H           ; mode 1, REN = 1
       mov SBUF,#2FH
       clr RI                  ; ensure that RI is cleared
LOOP:
       jmp LOOP                ; endless loop
```

Interrupt Based Transmitting

```
       org 0h                  ; reset vector
       ljmp program_start

       org 023h                ; serial interrupt vector
       ljmp SerialInterrupt

       org 100                 ; begin of main program
SerialInterrupt:
       ...
       ...
       clr TI
       reti

       program_start:
       setb EA                 ; enable interrupts generally
       setb ES                 ; enable serial interrupts

       mov SCON,#40H           ; mode 1, REN = 0
       mov SBUF,#2FH
```

9. Circuit Diagram



10. Parts List

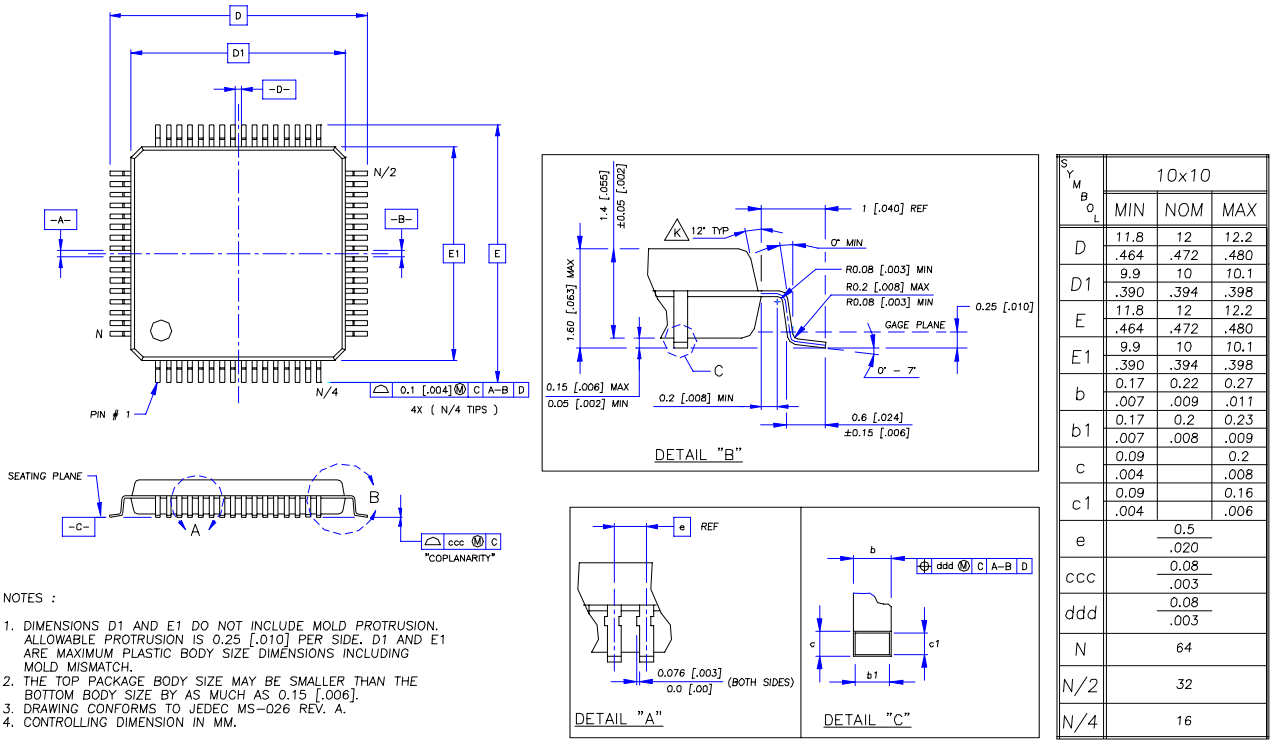
Designation	Value	Unit	Description
IC1			AS8267 / AS8268 Metering Integrated Circuits
IC2			Up to 32kB SPI Bus EEPROM (selectable in binary steps)
IC3	3.3	V	Voltage regulator LE33CZ
RSH	300	µOhm	Shunt resistor (see 'Analog Front End')
CT1			Current transformer
R1			Resistor (see 'Analog Front End')
R2	470	Ohm	Resistor (see 'Analog Front End')
R3	470	Ohm	Resistor
R4	680	Ohm	Resistor
R5	680	Ohm	Resistor
R6	680	Ohm	Resistor
R7	680	Ohm	Resistor
R8	680	Ohm	Resistor
R9	680	Ohm	Resistor
R10	4.7	Ohm	Resistor (see 'Analog Front End')
R11	680	Ohm	Resistor
R12	680	Ohm	Resistor
R13	470	Ohm	Resistor
C1	100	nF	Capacitor
C2	100	nF	Capacitor
C3	33	nF	Capacitor
C4	33	nF	Capacitor
C5	33	nF	Capacitor
C6	33	nF	Capacitor
C7	33	nF	Capacitor
C8	33	nF	Capacitor
C9	33	nF	Capacitor
C10	33	nF	Capacitor
C11	100	nF	Capacitor
C12	220	µF	Capacitor
C13	100	nF	Capacitor
C14	10	nF	Capacitor
C15	10	nF	Capacitor
C16	1.0	µF	Capacitor
C17	100	nF	Capacitor

Designation	Value	Unit	Description
C18	1.0	µF	Capacitor
C19	100	nF	Capacitor
C20	1.0	µF	Capacitor
C21	10	nF	Capacitor
C22	0.47	µF	Capacitor
C23	470	µF	Capacitor
D1			Diode 1N4148
D2			Diode 1N4148
D3			Diode 1N4148
D4			Diode 1N4148
D5			Diode 1N4004
ZD1	15	V	Zener diode BZV85–C15
L1			Varistor
BAT	3.0	V	Lithium battery
LCD			Liquid crystal display
XTAL	3.579545	MHz	Crystal

Note: The external components for the programmable multi-purpose I/Os (MPIO) are not included in the above parts list, as they depend on the specific meter functional requirements.

11. Packaging

LQFP64



SYMBOL	10x10		
	MIN	NOM	MAX
D	11.8	12	12.2
	.464	.472	.480
D1	9.9	10	10.1
	.390	.394	.398
E	11.8	12	12.2
	.464	.472	.480
E1	9.9	10	10.1
	.390	.394	.398
b	0.17	0.22	0.27
	.007	.009	.011
b1	0.17	0.2	0.23
	.007	.008	.009
c	0.09		0.2
	.004		.008
c1	0.09		0.16
	.004		.006
e	0.5		
	.020		
ccc	0.08		
	.003		
ddd	0.08		
	.003		
N	64		
N/2	32		
N/4	16		

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. A.
4. CONTROLLING DIMENSION IN MM.

12. Product Ordering Guide

Device Number	MPIO	LCDD	Temperature	Package	Packing
AS8267 BLQS	9	20 x 4	-40°C to 85°C	LQFP64	Tray in DryPack
AS8267 BLQW	9	20 x 4	-40°C to 85°C	LQFP64	T & R in DryPack
AS8268 BLQS	12	24 x 4	-40°C to 85°C	LQFP64	Tray in DryPack
AS8268 BLQW	12	24 x 4	-40°C to 85°C	LQFP64	T & R in DryPack

13. Collection of Formulae

Shunt resistor for mains current sensing:

$$R_{shunt} = \frac{V_p}{I_p}$$

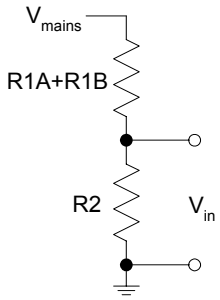
Where V_p is the peak input voltage to the IC at rated conditions and I_p the peak Imax value of the meter.

CT voltage setting termination resistor for mains current sensing:

$$R_{VS} = \frac{V_{in(p)}}{I_L \sqrt{2}}$$

Where $V_{in(p)}$ is the peak input voltage to the IC at rated conditions (V_{mains}; I_{max}). i.e.: If Gain = 4 then $V_{in(p)}$ must be set at 150mVpeak and I_L is the CT RMS secondary current at rated conditions (V_{mains}; I_{max})

Voltage divider for the V_{mains} input for the energy calculation:



$$R1A + R1B = R2 \times \frac{(V_{mains} - V_{in(P)})}{V_{in(P)}}$$

Where V_{mains} is the peak mains voltage and $V_{in(P)}$ is the is the peak input voltage to the IC at rated conditions.

Phase shift value of 1 unit of phase correction relative to the mains frequency:

$$1\,unit = 360° \times \frac{t_{ovs}}{t_{mains}} = 360° \times \frac{f_{mains}}{f_{ovs}} = 360° \times \frac{f_{mains}}{f_{osc}/8}$$

$$Phase = \#\,unit \times 360° \times \frac{f_{mains}}{f_{osc}/8}$$

Where f_{mains} is the mains frequency and f_{OSC} is the oscillator frequency.

Phase correction factor with a power factor (PF) of less than 1:

The meter has been calibrated at PF = 1 and the error is approximately 0 for I_{cal} (calibration current). If the PF is reduced, the effect of phase differences results in an increased error ('phase_error'):

First, the related phase shift in degrees can be calculated using the following formula:

$$phase\_shift = \arccos\left(\left(1+\frac{phase\_error[\%]}{100}\right)\times\cos 60^\circ\right) - 60^\circ$$

Where the phase_error is the measured error in percentage and cosΦ is the phase angle.

For phase_error = 9.2[%] the phase_shift is 3.0°.

For f_{osc} = 3.579545MHz and f_{mains} = 50Hz one phase correction unit represents 2.41', which is 0.04023°.

Thus the phase correction factor must be set to

$$\frac{3.0^\circ}{0.04023^\circ} = 74.57 \text{ units}$$

= 75 units.

The pcorr register has to be set to 4bh.

RMS values from the voltage (sos_v) and current (sos_i1 and sos_i2):

$$V_{rms} = \sqrt{\frac{1}{nsamp}\sum_{i=1}^{nsamp} V_i^2}\ , \text{ where } \sum_{i=1}^{nsamp} V_i^2 \text{ is the sos\_v value}$$

$$I_{rms} = \sqrt{\frac{1}{nsamp}\sum_{i=1}^{nsamp} I_i^2}\ , \quad \text{where } \sum_{i=1}^{nsamp} I_i^2 \text{ is the sos\_i value}$$

Where nsamp, the number of samples before an update rate of the MDR (meter data register), is selected to achieve coherent sampling.

16-bit calibration values for the voltage (V) and current (I) channels:

The ideal values after RMS calculations of voltage (V_{in} of 100mVp at rated conditions) and current and (I_{in} of 30mVp at rated conditions when Gain = 20) are:

RMS_V(ideal) = 479 (rms)

RMS_I (ideal) = 292,100 (rms)

Due to non-ideal components a different RMS value is calculated: RMS_I(actual). From this, the required calibration factor is calculated using the following formula:

$$cal\_i = \frac{RMS\_I(ideal)}{RMS\_I(actual)}$$

The following formula calculates the actual value to be programmed into the calibration registers (cal_v; cal_i1; cal_i2):

$$cal\_i(reg) = hex(round(cal\_i \times 32{,}768))$$

Fast internal pulse rate (PR_{int}):

The Fast Pulse Gen output always has the same relationship with the LED pulse rate, which is defined by mconst. Only if LED is calibrated to a meter constant different from those provided in the mconst table, will the fast internal pulse rate be different.

$$PR_{int} = 204{,}800 \times \frac{TargetPulseRate}{mconst} [i/kWh]$$

Where mconst is the meter constant.

$$1i = \frac{1{,}000 \times 3{,}600}{PR_{int}} [Ws]$$

Active power calibration (Pulse_lev):

The Pulse_lev is specified such that a typical pulse rate of 204,800i/kWh can be achieved.
During energy pulse calibration the correct Pulse_lev is determined in order to get the desired pulse rate.
The IC default value for Pulse_lev is defined for I_{max}=40A and V_{mains}=230V.

Default Pulse_lev: 570,950

The formula for calculating the ideal Pulse_lev is as follows:

$$Pulse\_lev(ideal) = \frac{230V}{V_{mains}} \times \frac{40A}{I_{max}} \times Pulse\_lev(default)$$

The standard or reference meter pulses are counted between two pulses from the meter under test. From the deviation the corrected Pulse_lev may be calculated.

$$Pulse\_lev(corrected) = Pulse\_lev(ideal) \times \frac{Ni}{Na},$$

Where Ni is the ideal number of pulses and Na is the actual number of pulses (pcnt register in MPIO).

The ideal number of pulses Ni is the ratio between the pulse rates, which is always >1. The formula for Ni is as follows:

$$Ni = \frac{PR(ref)}{LED\,Pulse\,Rate(mconst)}$$

Where PR(ref) is the reference meter constant.

LED Meter Constant (non-standard):

The LED pulses are derived directly from the fast internal pulses (204,800i/kWh) and is specified using the parameter 'mconst' of SREG. If the target meter constant is different from one of the selectable (mconst) meter constants, the following formula applies:

$$(IdealPulse\_lev \times \frac{Ni}{Na})$$

Where Ni is calculated using the Target Pulse Rate: $$Ni = \frac{Reference\,Meter\,Constant}{Target\,Pulse\,Rate}$$

NB: For mconst, select a pulse rate close to Target Pulse Rate, so that the Pulse_lev stays within reasonable limits.

Temperature Sensor:

The AS8267 / AS8268 ICs include an on-chip temperature sensor which allows for temperature correction over the entire operating temperature range of the device.

The actual temperature value is calculated by the means of the following formula:

$$Temp[°C] = (Temp\_corrected \times 0.193) - 75$$

$$Temp\_corrected = TS\_Result[15:0] + TS\_OffsetCorr[15:0]$$

14. Terminology

AFE	-	Analog front end
CT	-	Current transformer
DSP	-	Digital signal processor
EEPROM	-	Electrically erasable programmable read only memory
I_RAM	-	8051 internal memory
kB	-	kilobyte
LBPx	-	LCD back-plane driver pin
LCD	-	Liquid crystal display
LCDD	-	Liquid crystal display driver
LED	-	Light-emitting diode
LP_DIV	-	Low power divider
LP_OSC	-	Low power oscillator
LSB	-	Least significant bit
LSDx	-	LCD segment driver pin
MCU	-	Microcontroller unit
MDR	-	Meter data register (in DSP block)
MPIO	-	Programmable multi-purpose input/output
MSB	-	Most significant bit
P_ACCU	-	Power accumulator for real power
PF	-	Power factor
PLP	-	Power low pass filter
PSM	-	Power-supply monitor
PSW	-	Program status word
RAM	-	Random access memory
RES_N	-	System reset pin
RMS	-	Root mean square
RTC	-	Real time clock
SCT	-	System control
SDM	-	Sigma-delta modulator
SFR	-	Special function register
SQRT	-	Square root block
SREG	-	Settings register (in DSP block)
SPI	-	Serial peripheral interface
UART	-	Universal asynchronous receiver/transmitter
VREF	-	Voltage reference
WDT	-	Watchdog timer
X_RAM	-	8051 external data memory
X_DATA	-	64kByte address space

15. Revision

Revision	Date	Owner	Description
1.0	19-Jun-07	hza	

16. Copyright

17. Disclaimer

Devices sold by austriamicrosystems AG are covered by the warranty and patent indemnification provisions appearing in its Terms of Sale. austriamicrosystems AG makes no warranty, express, statutory, implied, or by description regarding the information set forth herein or regarding the freedom of the described devices from patent infringement. austriamicrosystems AG reserves the right to change specifications and prices at any time and without notice. Therefore, prior to designing this product into a system, it is necessary to check with austriamicrosystems AG for current information. This product is intended for use in normal commercial applications. Applications requiring extended temperature range, unusual environmental requirements, or high reliability applications, such as military, medical life-support or life-sustaining equipment are specifically not recommended without additional processing by austriamicrosystems AG for each application.

The information furnished here by austriamicrosystems AG is believed to be correct and accurate. However, austriamicrosystems AG shall not be liable to recipient or any third party for any damages, including but not limited to personal injury, property damage, loss of profits, loss of use, interruption of business or indirect, special, incidental or consequential damages, of any kind, in connection with or arising out of the furnishing, performance or use of the technical data herein. NO obligation or liability to recipient or any third party shall arise or flow out of austriamicrosystems AG rendering of technical or other services.

18. Contact

austriamicrosystems AG
A 8141 Schloss Premstaetten, Austria
T. +43 3136 500 0
F. +43 3136 525 01
info@austriamicrosystems.com