Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

19-4668; Rev 1; 12/09

Low-Power, Active Energy, Polyphase AFE

General Description

The MAXQ3181 is a dedicated electricity measurement front-end that collects and calculates polyphase voltage, current, active power and energy, and many other metering parameters of a polyphase load. The computed results can be retrieved by an external master through the on-chip serial peripheral interface (SPI™) bus. This bus is also used by the external master to configure the operation of the MAXQ3181 and monitor the status of operations.

The MAXQ3181 performs voltage and current measurements using an integrated ADC that can measure up to seven external differential signal pairs. An eighth differential signal pair is used to measure the die temperature. An internal amplifier automatically adjusts the current channel gain to compensate for low-current channel-signal levels.

Applications

3-Phase Active Energy Electricity Meters

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAXQ3181-RAN+	-40°C to +85°C	28 TSSOP

+Denotes a lead(Pb)-free/RoHS-compliant package.

Pin Configuration and Typical Application Circuit appear at end of data sheet.

Features

- **Compatible with 3-Phase/3-Wire, 3-Phase/4-Wire, and Other 3-Phase Services**
- **0.1% Active Power and Energy Linearity Error**
- **0.5% Apparent Power and Energy Linearity Error**
- **0.5% Linearity Errors for RMS Voltage and RMS Current**
- **Neutral Line Current Measurement**
- **Line Frequency (Hz)**
- **Power Factors**
- **Phase Sequence Indication**
- **Phase Voltage Absence Detection**
- **Programmable Pulse Width**
- **Programmable No-Load Current Threshold**
- **Programmable Meter Constant**
- **Programmable Thresholds for Undervoltage and Overvoltage Detection**
- **Programmable Threshold for Overcurrent Detection**
- **Amp-Hours in Absence of Voltage Signals**
- **On-Chip Digital Temperature Sensor**
- **Precision Internal Voltage Reference 2.048V (30ppm/°C typical), Also Supports An External Voltage Reference**
- **Active Power and Energy of Each Phase and Combined 3-Phase (kWh), Positive and Negative**
- **Apparent Power and Energy of Each Phase and Combined 3-Phase**
- **Supports Software Meter Calibration**
- **Up to 3-Point Multipoint Calibration to Compensate for Transducer Nonlinearity**
- **Power-Fail Detection**
- **Bidirectional Reset Input/Output**
- **SPI-Compatible Serial Interface with Interrupt Request ($\overline{IRQ}$) Output**
- **Single 3.3V Supply, Low Power (35mW typical)**

Note: *Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, go to:* ***www.maxim-ic.com/errata****.*

Low-Power, Active Energy, Polyphase AFE

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

Low-Power, Active Energy, Polyphase AFE

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

Low-Power, Active Energy, Polyphase AFE

ABSOLUTE MAXIMUM RATINGS

Voltage Range on DVDD Relative to DGND .....-0.3V to +4.0V
Voltage Range on AVDD Relative to AGND.....-0.3V to +4.0V
Voltage Range on AGND Relative to DGND.....-0.3V to +0.3V
Voltage Range on AVDD Relative to DVDD .....-0.3V to +0.3V
Voltage Range on Any Pin Relative to
DGND except VxP, IxN Pins.....-0.3V to +4.0V
Voltage Range on VxP, IxN Relative to AGND.....-0.3V to +4.0V
Operating Temperature Range .....-40°C to +85°C
Junction Temperature.....+150°C
Storage Temperature Range .....-65°C to +150°C
Lead Soldering Temperature.....Refer to the IPC/
JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

METERING SPECIFICATIONS

(V_{AVDD} = V_{DVDD} = V_{RST} to 3.6V, Current Channel Dynamic Range 1000:1 at T_A = +25°C, unless otherwise noted.) (Note 1)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Active Energy Linearity Error	DR 1000:1		0.1		%
Apparent Energy Linearity Error	DR 1000:1		0.5		%
RMS Voltage Linearity Error	DR 20:1		0.5		%
RMS Current Linearity Error	DR 500:1		1.0		%
	DR 20:1		0.5		
Line Frequency Error			0.5		%
Power Factor Error			1.0		%

ELECTRICAL CHARACTERISTICS

(V_{AVDD} = V_{DVDD} = V_{RST} to 3.6V, T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER-SUPPLY SPECIFICATIONS						
Digital Supply Voltage	V_{DVDD}		V_{RST}		3.6	V
Power-Fail Interrupt Trip Point	V_{PFW}	Active mode, EPWRF = 1	2.84		3.13	V
Power-Fail Reset Trip Point	V_{RST}	Active mode	2.70		2.99	V
Analog Supply Voltage	V_{AVDD}		V_{RST}		3.6	V
Analog Supply Current	I_{AVDD}	f_{CLK} = 8MHz		0.9	1.8	mA
Digital Supply Current	I_{DVDD}	f_{CLK} = 8MHz		8.5	13	mA
Low-Power Measurement Mode Current	I_{LOWPM}	LOWPM = 1 (Note 1)		4.2		mA
Stop-Mode Current				0.2	12	µA
DIGITAL I/O SPECIFICATIONS						
Input High Voltage	V_{IH}		0.7 x V_{DVDD}			V
Input Low Voltage	V_{IL}				0.3 x V_{DVDD}	V
Input Hysteresis	V_{IHYS}	V_{DVDD} = 3.3V		500		mV
Input Leakage	I_L	V_{IN} = DGND or V_{DVDD}, pullup off		±0.01	±1	µA

ELECTRICAL CHARACTERISTICS (continued)

(V_{AVDD} = V_{DVDD} = V_{RST} to 3.6V, T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Low Current	I_{IL}	V_{IN} = 0.4V, weak pullup on	-50			µA
$\overline{RESET}$ Pullup Resistance	R_{RESET}		50	150	200	kΩ
Output High Voltage (Except $\overline{RESET}$)	V_{OH}	I_{OH} = -4mA	V_{DVDD} - 0.4			V
		I_{OH} = -6mA	V_{DVDD} - 0.5			
Output Low Voltage	V_{OL}	I_{OL} = 4mA			0.4	V
		I_{OL} = 6mA			0.5	
SYSTEM CLOCK SOURCES						
External Clock Input Frequency			0		8.12	MHz
External Clock Input Duty Cycle			45		55	%
External HF Crystal Frequency		Fundamental mode			8.12	MHz
XTAL1, XTAL2 Internal Load Capacitance				16		pF
Internal RC Oscillator Frequency			7.4	7.6	8.6	MHz
Internal RC Oscillator Accuracy				±2		%
Internal RC Oscillator Current				50	120	µA
Internal RC Oscillator Startup Delay		(Note 1)		0.45		µs
ANALOG-TO-DIGITAL CONVERTER						
Input Voltage Range			0		V_{REF}	V
Common-Mode Bias	V_{COMM}			1.14		V
Offset Error				±2		mV
Offset Error Drift				±8		µV/°C
Gain Error (G = 1)				0.05		%
Spurious-Free Dynamic Range	SFDR			90		dB
Total Harmonic Distortion	THD			90		dB
Input Bandwidth (-3dB)		(Note 1)		7		kHz
INTERNAL VOLTAGE REFERENCE						
Temperature Coefficient		(Note 1)		30		ppm/°C
Output Voltage	V_{REF}			2.048		V
INTERNAL TEMPERATURE SENSOR						
Temperature Error		(Note 1)	-4		+4	°C
SPI SLAVE-MODE INTERFACE TIMING						
Maximum SPI Clock Rate		(Note 3)			f_{SYS}/4	MHz
SCLK Input Pulse-Width High	t_{SCH}	(Note 3)	4 x t_{SYS}			ns
SCLK Input Pulse-Width Low	t_{SCL}	(Note 3)	4 x t_{SYS}			ns

Low-Power, Active Energy, Polyphase AFE

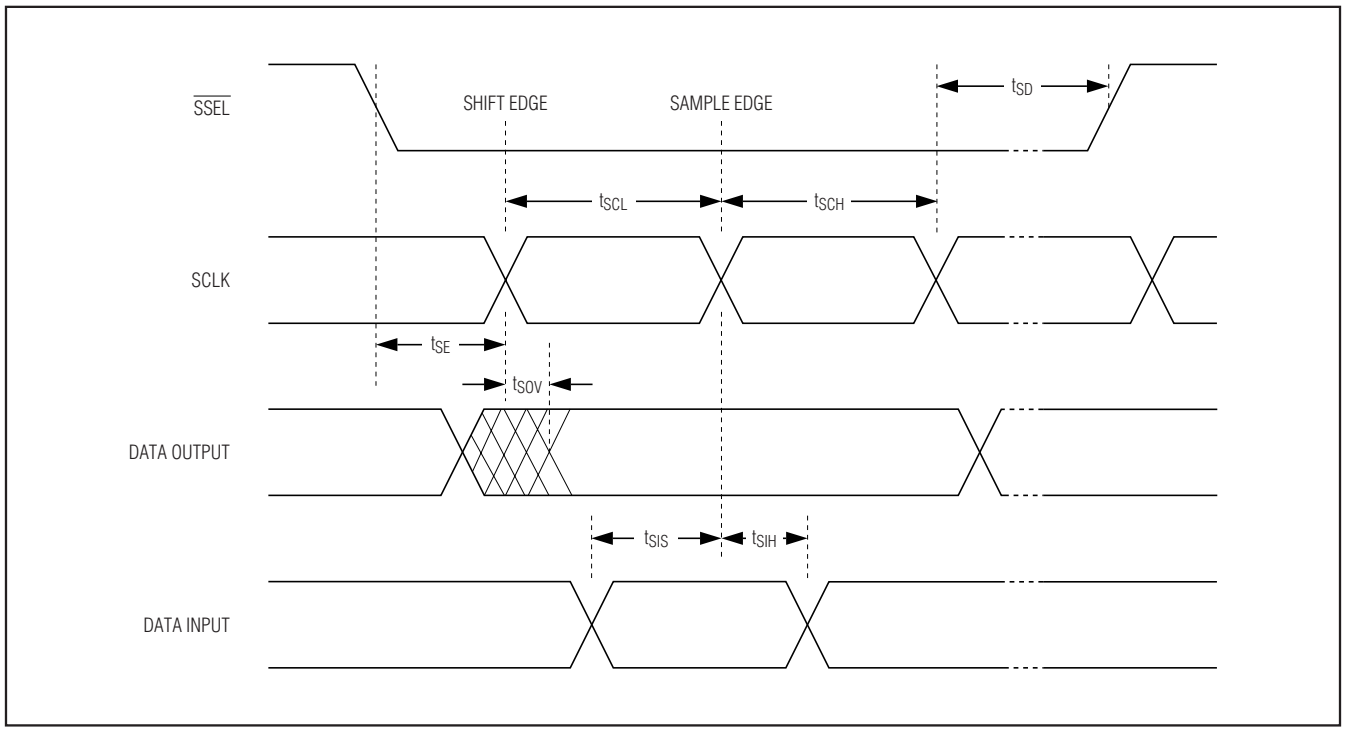
ELECTRICAL CHARACTERISTICS (continued)

(V_{AVDD} = V_{DVDD} = V_{RST} to 3.6V, T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

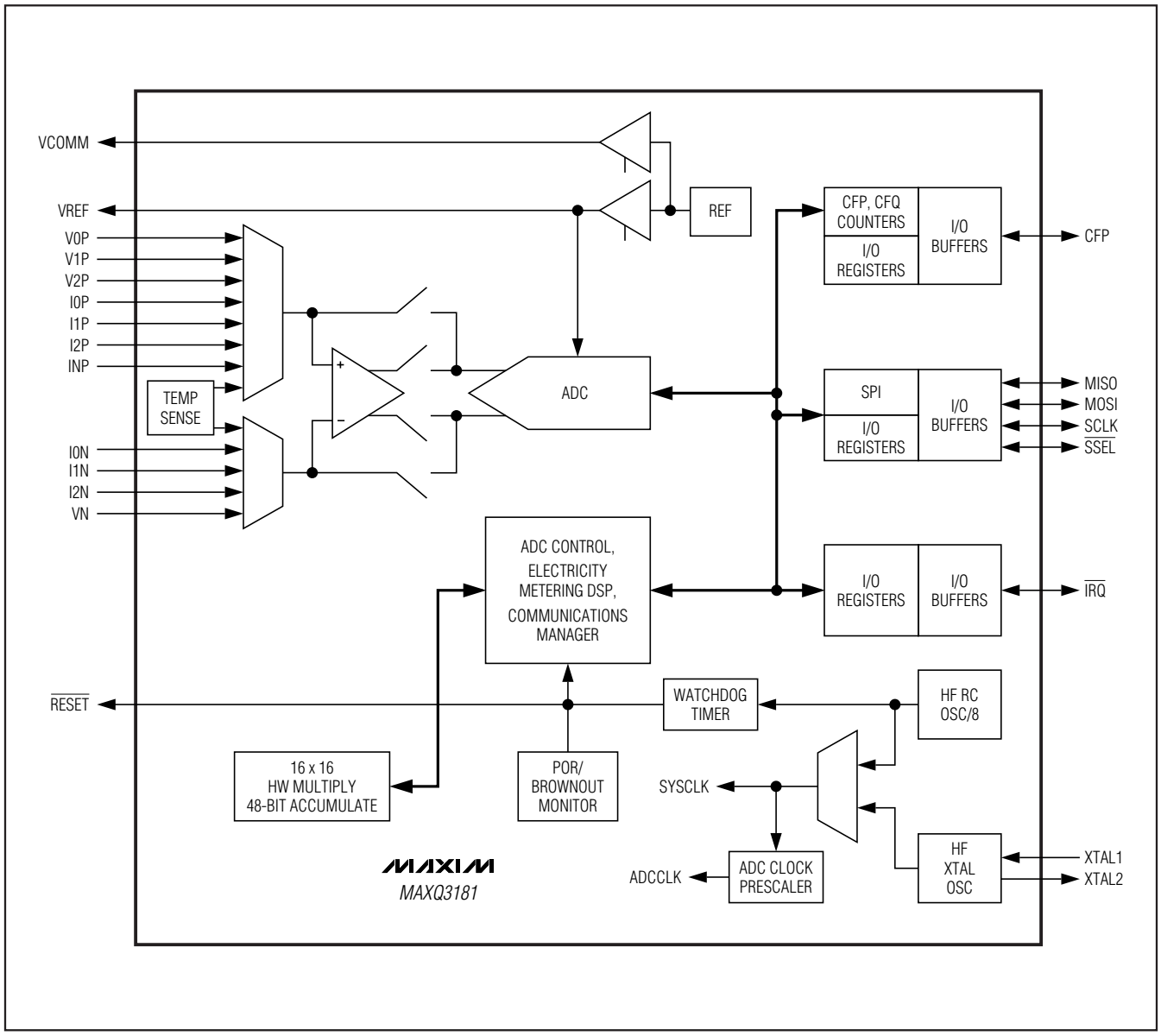
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
$\overline{SSEL}$ Low to First SCLK Edge (Slave Enable)	t_{SE}	(Note 3)	$4/t_{SYS}$			ns
Last SCLK Edge to $\overline{SSEL}$ High (Slave Disable)	t_{SD}		t_{SYS} + 5			ns
MOSI Valid to SCLK Sample Edge (MOSI Setup)	t_{SIS}		5			ns
SCLK Sample Edge to MOSI Change (MOSI Hold)	t_{SIH}		t_{SYS} + 5			ns
SCLK Shift Edge to MISO Valid (MISO Hold)	t_{SOV}				$3t_{SYS}$ + 5	ns

Note 1: Specifications guaranteed by design but not production tested.
Note 2: Specifications to -40°C are guaranteed by design and are not production tested.
Note 3: $t_{SYS} = 1/f_{SYS}$, where f_{SYS} is the system clock frequency, external or internal.

SPI Slave Mode Timing



Block Diagram



Low-Power, Active Energy, Polyphase AFE

Pin Description

PIN	NAME	FUNCTION
		POWER PINS
17, 22	DVDD	Digital Supply Voltage
25	AVDD	Analog Supply Voltage
18	DGND	Digital Ground
9	AGND	Analog Ground
23	VCOMM	Voltage Bias. This pin can be used to create an input common-mode DC offset for ADC channel conversions.
24	VREF	Voltage Reference. Reference voltage for the ADC. An external reference voltage can be connected to this pin when extremely high accuracy is required.
		VOLTAGE AND CURRENT PINS
26, 3, 4	V0P, I0P, I0N	Phase A Voltage and Current Analog Inputs
27, 5, 6	V1P, I1P, I1N	Phase B Voltage and Current Analog Inputs
28, 7, 8	V2P, I2P, I2N	Phase C Voltage and Current Analog Inputs
1	VN	Analog Input for Common Voltage
2	INP	Analog Input for Neutral Current
		CLOCK PINS
10	XTAL2	High-Frequency Crystal Input/Output. When using an external high-frequency crystal, the crystal oscillator circuit should be connected between XTAL1 and XTAL2. When using an externally driven clock (EXTCLK = 1), the clock should be input at XTAL1, with XTAL2 left unconnected.
11	XTAL1	
12	$\overline{IRQ}$	Interrupt Request Output. This line is driven low by the device to indicate to the master that an unmasked interrupt has occurred.
13	$\overline{SSEL}$	Slave Select Input. This line is the active-low slave select input for the SPI interface.
14	SCLK	Slave Clock Input. This line is the clock input for the SPI interface.
15	MOSI	Master Out-Slave In Input. This line is used by the master to transmit data to the slave (the MAXQ3181) over the SPI interface.
16	MISO	Master In-Slave Out Output. This line is used by the MAXQ3181 (the slave) to transmit data back to the master over the SPI interface.
19	CFP	Pulse Output. Configurable to represent energy or RMS voltage or current.
21	$\overline{RESET}$	Active-Low Reset Input/Output. An external master can reset the MAXQ3181 by driving this pin low. This pin includes a weak pullup resistor to allow for a combination of wired-OR external reset sources. An RC circuit is not required for power-up, as this function is provided internally. This pin also acts as a reset output when the source of the reset is internal to the device (power-fail, watchdog reset, etc.). In this case, the $\overline{RESET}$ pin is held low by the device until it exits the reset state, then the $\overline{RESET}$ pin is released.
		NO CONNECTION PINS
20	N.C.	No Connection

Detailed Description

The MAXQ3181 contains four major subsections: the analog front-end, the digital signal processor, the precision pulse generators, and an SPI peripheral for communication to the host processor.

Analog Front-End

The analog front-end (AFE) is an 8-channel analog-to-digital converter (ADC). It operates autonomously in the standard configuration, assigning three channels to phase A, B, and C voltage; three channels to phase A, B, and C current; one channel to neutral current; and the last channel to a temperature sensor.

Each channel also contains a programmable-gain amplifier capable of providing a gain of 1, 2, 4, 8, 16, or 32 incoming signals. Only the voltage channels permit gain scaling by the host processor. The MAXQ3181 DSP firmware automatically sets the gain on current channels.

Digital Signal Processor

The DSP code is permanently embedded in masked ROM and accepts raw current and voltage samples for each of three phases and continuously calculates a host of values including RMS volts, RMS amps, real energy, apparent energy, and power factor.

The MAXQ3181 DSP core processes incoming samples from the analog front-end according to user configurations. The host sets these operating parameters by specifying addresses within the device RAM space. When a calculation cycle is complete, the results are placed back into RAM as well. Thus, the DSP core uses the RAM block as both its input (for operating parameters) and output (for calculation results) medium. See the *SPI Peripheral* section for how the host writes operating parameters and reads results from the RAM.

The DSP also calculates certain values such as line frequency and active power only when demanded by the host.

Precision Pulse Generators

The MAXQ3181 includes a precision pulse generator that generates a pulse whenever certain conditions are met. In the MAXQ3181, many meter quantities can be selected for conversion to meter pulses including absolute energy, net energy, voltage, and current.

The pulse generator is an accumulator. On each DSP cycle, whatever quantity is being measured—real energy, current, or something else—is added to the pulse accumulator. The pulse accumulator is then tested to determine if the value in the accumulator is greater than the threshold. If it is greater, the threshold value is subtracted from the accumulator value and the meter pulse starts.

SPI Peripheral

The SPI controller is a slave-only device that can read or write any location in the data RAM. Additionally, it can request data from on-demand registers.

The MAXQ3181 implements a truly full-duplex communication, rather than the pseudo half-duplex mode used by other SPI peripherals. That is, each time a character is received by the MAXQ3181, a meaningful character is returned to the host. Often, this is a protocol character. In this way, the host can be assured that the command has been received and is valid. Optional error checking can also be enabled to further guarantee proper operation.

Operating Modes

The MAXQ3181 has two basic modes of operation, each of which is described in the following sections. The Initialization Mode is the default mode upon power-up or following reset; entry to and exit from the other operating modes is only performed as a result of commands sent by the master.

Run Mode

This mode is the normal operating mode for the MAXQ3181. In this mode, the MAXQ3181 continuously executes the following operations:

- Scans analog front-end channels and collects raw voltage and current samples.
- Processes voltage and current samples through DSP filters as enabled and configured.
- Calculates power, energy, and other required quantities and stores these values in RAM registers.
- Responds to register write and read commands from the master.
- Outputs power pulse on CFP as configured.
- Drives $\overline{IRQ}$ when an interrupt condition has been detected and the interrupt is not masked.

Stop Mode

This mode places the MAXQ3181 into a power-saving state where it consumes the least possible amount of current. In Stop Mode, all functions are suspended, including the ADC and power and voltage measurement and processing. The MAXQ3181 does not respond to any commands from the master in this operating state.

Low-Power, Active Energy, Polyphase AFE

Entry into Stop Mode only occurs at the request of the master. To place the MAXQ3181 into Stop Mode, the master must read the ENTER STOP (0xC02) register. Once this register has been read, the MAXQ3181 enters Stop Mode immediately, before the transmission of the final ACK byte by the MAXQ3181.

There are three possible ways to bring the MAXQ3181 back out of Stop Mode.

- **Power Cycle.** The MAXQ3181 automatically exits Stop Mode if a power-on reset occurs. Following exit from Stop Mode, all registers are cleared back to their default states, and the MAXQ3181 transitions to Initialization Mode.
- **External Reset.** The MAXQ3181 exits Stop Mode if an external reset is triggered by driving $\overline{\text{RESET}}$ low. Once the $\overline{\text{RESET}}$ pin is released and allowed to return to a high state, the MAXQ3181 comes out of reset and goes into Initialization Mode. All registers are cleared to their default states when exiting Stop Mode in this manner.
- **External Interrupt.** Driving the $\overline{\text{SSEL}}$ pin low causes the MAXQ3181 to exit Stop Mode without undergoing a reset cycle. When exiting Stop Mode in this manner, all register and configuration settings are retained, and the MAXQ3181 automatically resumes electric-metering functions and sample processing.

Note that when the master is communicating with the MAXQ3181, the $\overline{\text{SSEL}}$ line is normally driven low at the beginning of each SPI command. This means that if the master sends an SPI command after the MAXQ3181 enters Stop Mode, the MAXQ3181 automatically exits Stop Mode.

Reset Sources

There are several different sources that can cause the MAXQ3181 to undergo a reset cycle. For any type of hardware reset, the $\overline{\text{RESET}}$ pin is driven low when a reset occurs.

External Reset

This hardware reset is initiated by an external source (such as the master controller or a manual pushbutton press) driving the $\overline{\text{RESET}}$ pin on the MAXQ3181 low. The $\overline{\text{RESET}}$ line must be held low for at least four cycles of the currently selected clock for the external reset to take effect. Once the external reset takes effect, it remains in effect indefinitely as long as $\overline{\text{RESET}}$ is held low. Once the external reset has been released, the MAXQ3181 clears all registers to their default states and resumes execution in Initialization Mode.

When an external reset occurs outside of Stop Mode, execution (in Initialization Mode) resumes after four cycles of the currently selected clock (external high-frequency crystal for Run Mode, 1MHz internal RC oscillator for LOWPM Mode). As the MAXQ3181 enters Initialization Mode, the LOWPM bit is always cleared

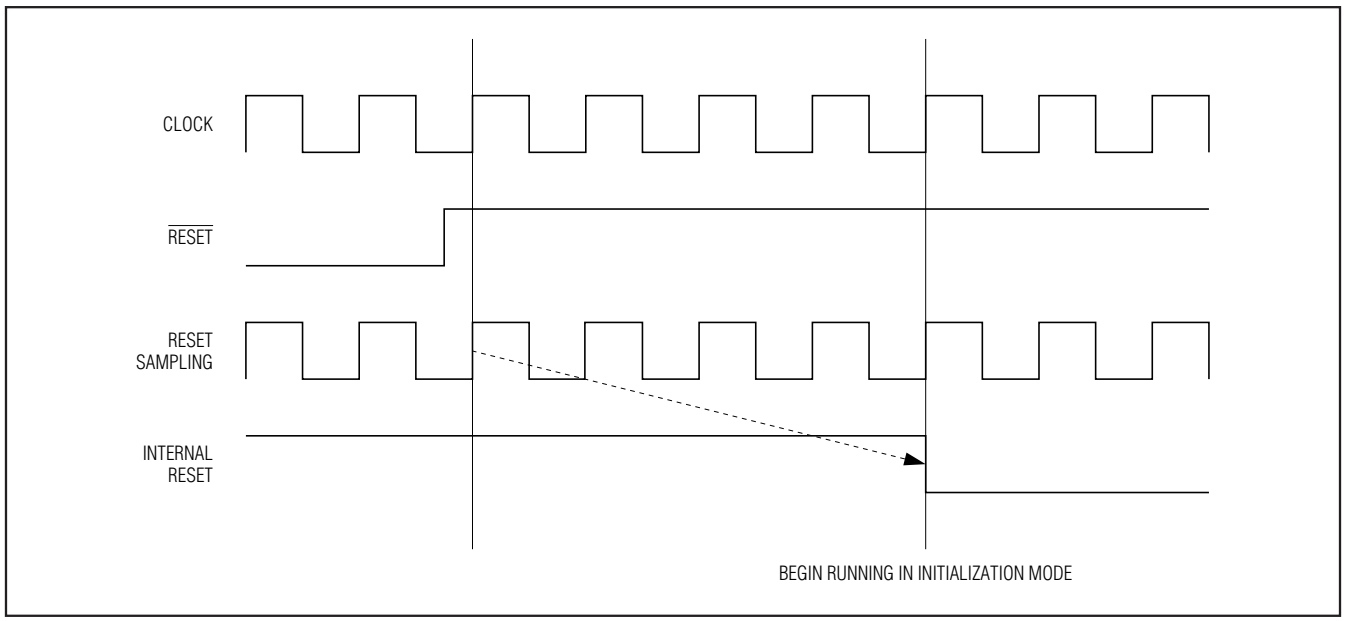


Figure 1. External Reset

to 0, meaning that the MAXQ3181 always switches to the high-frequency clock before it begins accepting commands in Initialization Mode.

When an external reset occurs from Stop Mode, execution (in Initialization Mode) resumes after 128 cycles of the internal RC oscillator (or approximately 128µs).

Power-On Reset

When the MAXQ3181 is first powered up, or when the power supply, V_{DVDD}, drops below the V_{RST} power-fail trip point (outside of Stop Mode), the MAXQ3181 is held in power-on reset. Once the power supply rises above the V_{RST} level, the power-on reset state is released and all registers are reset to their defaults and execution resumes in Initialization Mode. The high-frequency external crystal (LOWPM = 0) is always selected as the clock source following any power-on or brownout reset.

In Stop Mode brownout detection is disabled, so a power-on reset does not occur until V_{DVDD} drops to a lower level (V_{POR}). From the master's perspective, power-on resets and brownout resets both cause the MAXQ3181 to reset in the same way.

Watchdog Reset

The MAXQ3181 includes a hardware watchdog timer that is armed and periodically reset automatically during normal operation. Under normal circumstances, the MAXQ3181 always resets the watchdog timer often enough to prevent it from expiring. However, if an internal error of some kind causes the MAXQ3181 to lock up or enter an endless execution loop, the watchdog timer expires and triggers an automatic hardware reset. There is no register flag to indicate to the master that a watchdog reset has occurred, but the $\overline{\text{RESET}}$ line strobes low briefly.

The watchdog timer does not run during Stop Mode.

Software Reset

The master initiates a software reset by setting the SWRES (OPMODE0.3) bit to 1. When a software reset occurs, the MAXQ3181 clears all registers to their default states and returns to Initialization Mode, in the

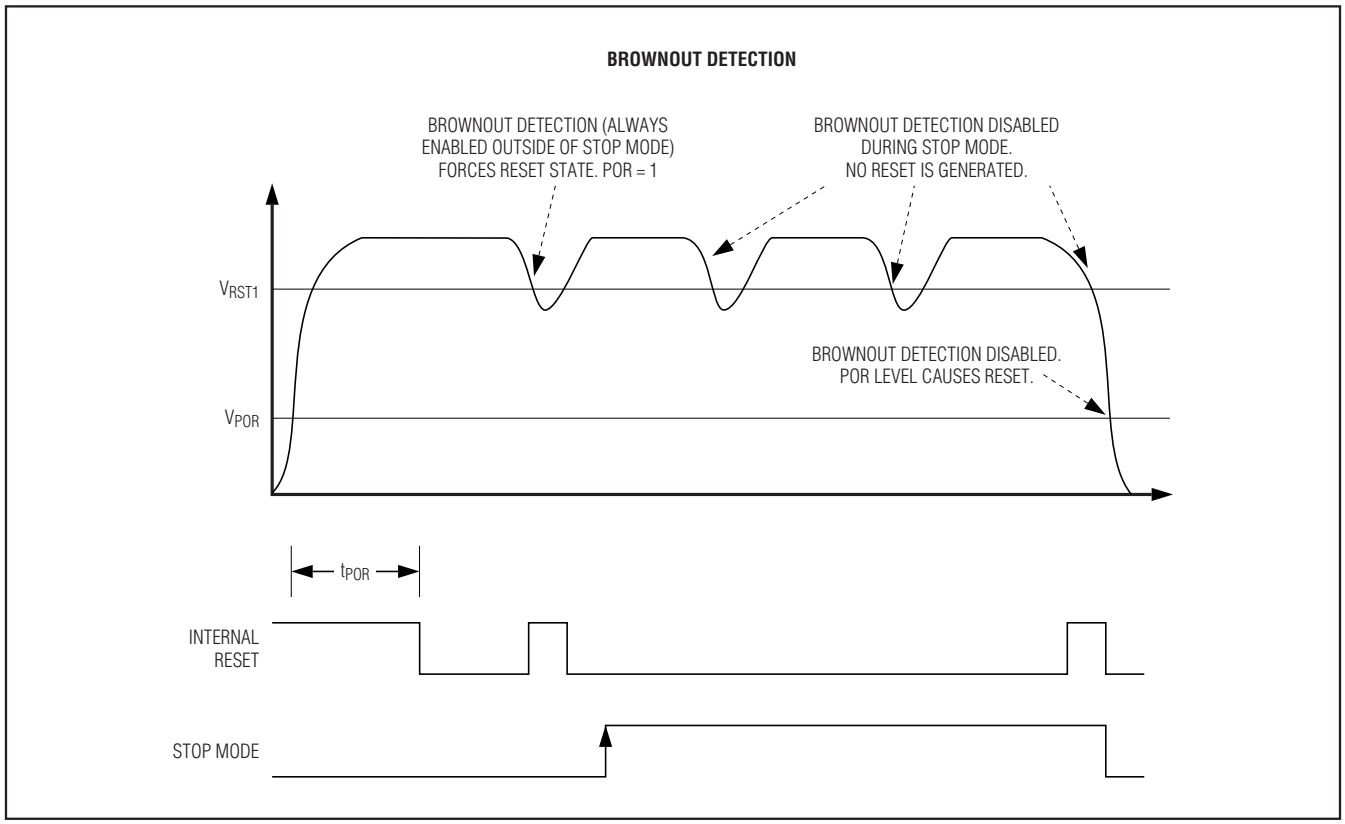


Figure 2. Brownout Reset

Low-Power, Active Energy, Polyphase AFE

same manner as if an external reset had taken place. Unlike a hardware reset, however, a software reset does not cause the MAXQ3181 to drive the RESET line low.

Power-Supply Monitoring

In addition to the hardware reset provided by the power-on reset and brownout reset circuits, the MAXQ3181 includes the capability to detect a low power supply on the DVDD pin and alert the master through the interrupt (IRQ) mechanism before a hardware reset occurs. This function, which is always enabled outside of Stop Mode, causes the RAM status register flag PWRF (IRQ_FLAG.0) to be set to 1 whenever V_{DVDD} drops below the V_{PFW} trip point. Once PWRF has been set to 1 by hardware, it can only be cleared by the master (or by a system reset). Whenever PWRF = 1, if the EPWRF interrupt masking bit is also set to 1, the MAXQ3181 drives IRQ low to signal to the master that an interrupt condition (in this case, a power-fail warning) exists and requires attention.

Clock Sources

All operations including ADC sampling and SPI communications are synchronized to a single system clock. This clock can be obtained from any one of three selectable sources, as shown in Figure 3.

External High-Frequency Crystal

The default system clock source for the MAXQ3181 is an external high-frequency crystal oscillator circuit connected between XTAL1 and XTAL2. When clocked with an external crystal, a parallel-resonant, AT-cut crystal oscillating in the fundamental mode is required.

When using a high-frequency crystal, the fundamental oscillation mode of the crystal operates as inductive reactance in parallel resonance with external capacitors C1 and C2. The typical values of these external capacitors vary with the type of crystal being used and should be selected based on the load capacitance as suggested by the crystal manufacturer.

Since noise at XTAL1 and XTAL2 can adversely affect device timing, the crystal and capacitors should always be placed as close as possible to the XTAL1 and XTAL2 pins, with connection traces between the crystal and the device kept as short and direct as possible. In multiple layer boards, avoid running other high-speed digital signals underneath the crystal oscillator circuit if possible, as this could inject unwanted noise into the clock circuit.

Following power-up or any system reset, the high-frequency clock is automatically selected as the system clock source. However, before this clock can be used

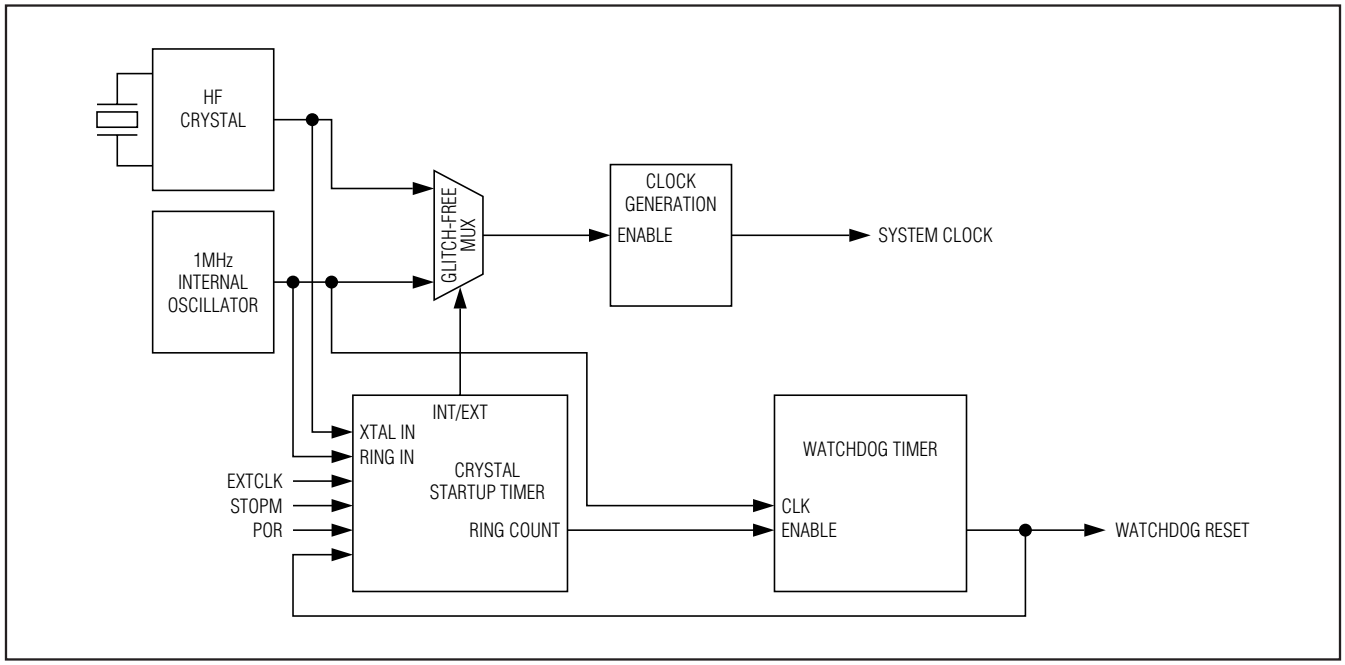


Figure 3. Simplified Clock Sources

for system execution, a crystal warmup timer must count 65,536 cycles of the high-frequency clock. While this warmup time period is in effect, execution continues using the internal 1MHz oscillator. Once the 65,536-cycle count completes (which requires approximately 8.2ms at 8MHz), the device automatically switches over to the high-frequency clock. This crystal warmup timer is also activated upon exit from Stop Mode, since the high-frequency crystal oscillator is shut down during Stop Mode.

External High-Frequency Clock

Instead of using a crystal oscillator to generate the high-frequency clock, it is also possible to input a high-frequency clock that has been generated by another source (such as a digital oscillator IC) directly into the XTAL1 pin of the MAXQ3181.

To use an external high-frequency clock as the system clock source, the XTAL1 pin should be used as the clock input and the XTAL2 pin should be left unconnected. The master should also shut down the internal crystal oscillator circuit by setting the EXTCLK bit (OPMODE0.4) to 1. This bit is only cleared by the MAXQ3181 if a power-on or brownout reset occurs and is unaffected by other resets.

When using an external high-frequency clock, the clock signal should be generated by a CMOS driver. If the clock driver is a TTL gate, its output must be connected to DVDD through a pullup resistor to ensure that the correct logic levels are generated. To minimize system noise in the clock circuitry, the external clock source must meet the maximum rise and fall times and the minimum high and low times specified for the clock source in the *Electrical Characteristics* table.

Internal RC Oscillator

When the external high-frequency crystal is warming up, or when the MAXQ3181 is placed into LOWPM mode, the system clock is sourced from an internal RC oscillator. This internal oscillator is designed to provide the system approximately 1MHz, although the exact frequency varies over temperature and supply voltage.

If no external crystal circuit or high-frequency clock will be used, the MAXQ3181 can be forced to operate infinitely from the internal oscillator by grounding XTAL1. This ensures that the crystal warmup count never completes, so the MAXQ3181 runs from the internal oscillator in all active modes.

Master Communications

Before the MAXQ3181 can begin performing electric-metering operations, the master must initialize a number of configuration parameters. Since the MAXQ3181 does not contain internal nonvolatile memory, these parameters (stored in internal registers) must be set by the master each time a power-up or reset cycle occurs, or each time a switch is made between LOWPM Mode and Run Mode.

The external master communicates with the MAXQ3181 over a standard SPI bus, using commands to read and write values to internal registers on the MAXQ3181. These registers include, among many other items:

- Operating mode settings (Stop Mode, LOWPM Mode, external clock mode, etc.)
- Status and interrupt flags (power-supply failure, overcurrent/overvoltage detection, etc.)
- Masking control for interrupts to determine which conditions cause $\overline{IRQ}$ to be driven low
- Configuration settings for analog channel scanning
- Power pulse output configuration
- Filter coefficients and configuration
- Read-only registers containing accumulated power and energy data

As the MAXQ3181 obtains voltage and current measurements in Run Mode or LOWPM Mode, it accumulates, filters, and performs a number of calculations on the collected data. Many of these operations (including the various filtering stages) are configured by settings in registers written by the master. The output results can then be read by the master from various read-only registers in parallel with the ongoing measurement and processing operations.

SPI Communications Rate and Format

The SPI is an interdevice bus protocol that provides fast, synchronous, full-duplex communications between a designated master device and one or more slave devices. In a MAXQ3181-based design, the MAXQ3181 would be the slave device connected to a designated master microcontroller.

The external master initiates all communications transfers. The interrupt request line $\overline{IRQ}$, while not technically part of the SPI bus interface, is also used for master/slave communications because it allows the MAXQ3181 to notify the master that an interrupt condition exists. Some SPI peripherals sacrifice speed in favor of simulating a half-duplex operation. This is not the case with the MAXQ3181; it is truly a full-duplex SPI slave.

Low-Power, Active Energy, Polyphase AFE

During an SPI transfer, data is simultaneously transmitted and received over two serial data lines (MISO and MOSI) with respect to a single serial shift clock (SCLK). The polarity and phase of the serial shift clock are the primary components in defining the SPI data transfer format. The polarity of the serial clock corresponds to the idle logic state of the clock line and, therefore, also defines which clock edge is the active edge. To define a serial shift clock signal that idles in a logic-low state (active clock edge = rising), the clock polarity select (CKPOL; R_SPICF.0) bit should be configured to a 0, while setting CKPOL = 1 causes the shift clock to idle in a logic-high state (active clock edge = falling). The phase of the serial clock selects which edge is used to sample the serial shift data. The clock phase select (CKPHA; R_SPICF.1) bit controls whether the active or inactive clock edge is used to latch the data. When CKPHA is set to a logic 1, data is sampled on the inactive clock edge (clock returning to the idle state). When CKPHA is set to a logic 0, data is sampled on the active clock edge (clock transition to the active state). Together, the CKPOL and CKPHA bits allow four possible SPI data transfer formats.

Transfers over the SPI interface always start with the most significant bit and end with the least significant bit. All SPI data transfers to and from the MAXQ3181 are always 8 bits (one byte) in length. The MAXQ3181 SPI interface does not support 16-bit character lengths.

The default format (upon power-up or system reset) for the MAXQ3181 SPI interface is represented in Figure 4a (CKPOL = 0; CKPHA = 0). In this format, the

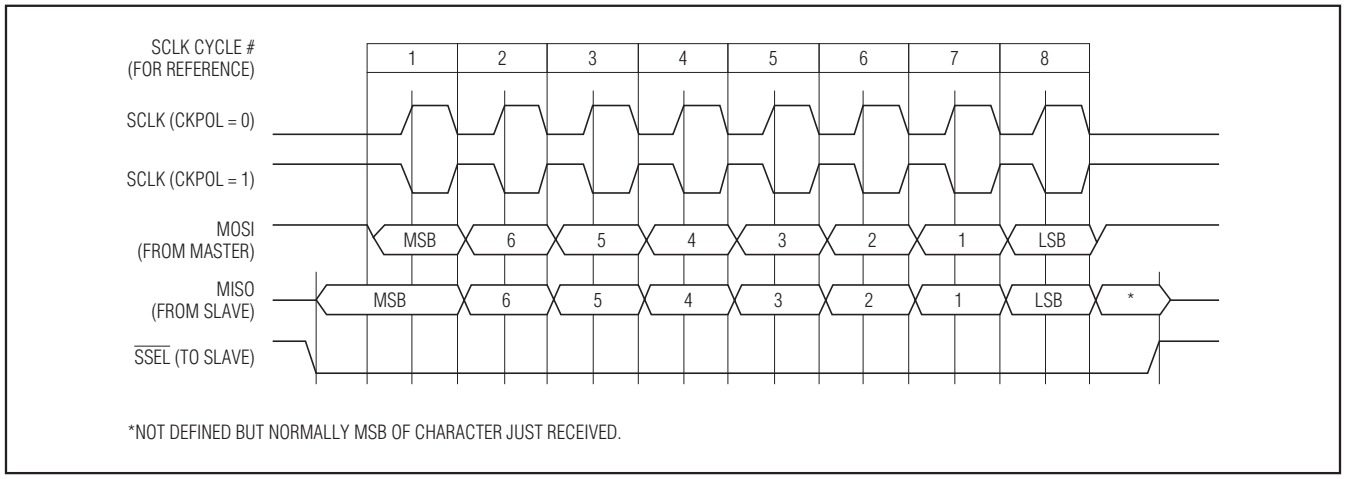


Figure 4a. SPI Interface Timing (CKPHA = 0)

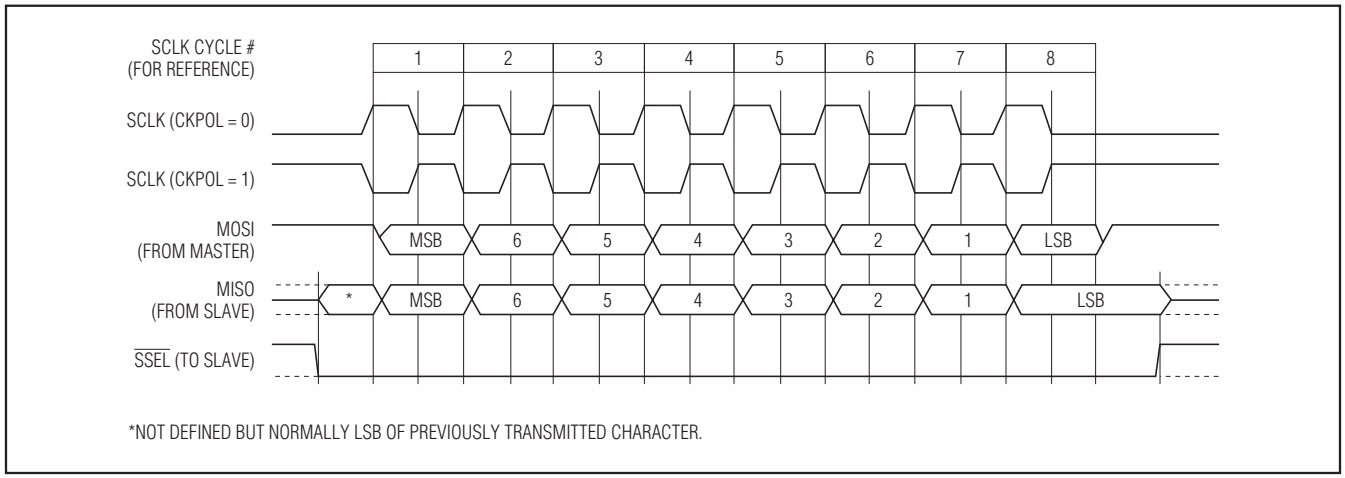


Figure 4b. SPI Interface Timing (CKPHA = 1)

SPI clock idle state is low, and data is shifted in and out on the rising edge of SCLK. Once SPI communication with the MAXQ3181 has been established, it is possible to alter the CKPOL and CKPHA format settings (as well as changing the $\overline{SSEL}$ signal from active low to active high) if desired by writing to the R_SPICF mirror register and then reading from the special command register UPD_SFR to copy the R_SPICF value into the internal SPI configuration register.

Whenever the active clock edge is used for sampling (CKPHA = 0), the transfer cycle must be started with assertion of the $\overline{SSEL}$ signal. This requirement means that the $\overline{SSEL}$ signal be deasserted and reasserted between successive transfers. Conversely, when the inactive edge is used for sampling (CKPHA = 1), the $\overline{SSEL}$ signal may remain low through successive transfers, allowing the active clock edge to signal the start of a new transfer.

The clock rate used for the SPI interface is determined by the bus master, since the MAXQ3181 always operates as an SPI slave device. However, the maximum clock rate is limited by the system clock frequency of the MAXQ3181. For proper communications operation, the SPI clock frequency used by the master must be less than or equal to the MAXQ3181's clock frequency divided by 4. For example, when the MAXQ3181 is running at 8MHz, the SPI clock frequency must be 2MHz or less. And if the MAXQ3181 is running in LOWPM Mode (or if the crystal is still warming up), the SPI clock frequency must remain at 250kHz or less for proper communications operation.

In addition to limiting the overall SPI bus clock rate, the master must also include a communications delay following each byte transmit/receive cycle. This delay, which provides the MAXQ3181 with time to process an ADC sample, should be a minimum of 400 system clocks. With default settings and running at 8MHz, this delay time is 50µs. Reducing the system clock frequency to 1MHz (LOWPM mode) would increase this delay period by a factor of 8 to 400µs.

SPI Communications Protocol

All transactions between the master and the MAXQ3181 consist of the master writing to or reading from one of the MAXQ3181's registers. To the host, the MAXQ3181 looks like a memory array that consists of both RAM and ROM. This is because the ROM firmware in the MAXQ3181 reads its operational parameters from RAM and places its results in RAM. Consequently, configuring a MAXQ3181 is as simple as performing a block write to its RAM locations.

Some read-only memory locations in the MAXQ3181 trigger actions within the device to calculate electricity-metering results on the fly. The specific function and purpose of RAM and virtual ROM locations are given in the register map. There are several different categories of internal registers on the MAXQ3181.

- **RAM Registers.** The values of these registers are stored in the internal RAM of the MAXQ3181. Some can be read and written by the master, while others are read-only. RAM registers are either 2 or 4 bytes long (16 or 32 bits), although in some registers not all the bits have defined values. Read/write registers are generally either status/flag registers (which can be written by either the MAXQ3181 or the master), configuration registers (which are written by the master and read by the MAXQ3181 firmware), or data registers (which are read-only and are written by the MAXQ3181 firmware and read by the master).
- **Virtual Registers.** These read-only registers are not stored in RAM; instead, they contain values that are calculated on the fly by the MAXQ3181 firmware when the master reads them. These registers are used by the master to obtain values such as phase A, B, and C active and apparent power; power factor; and RMS voltage and current, which are calculated from currently collected data on an as-needed basis. Most virtual registers are 8 bytes in length.
- **Hardware Registers.** These registers control core functions of the MAXQ3181 including the ADC and the SPI slave bus controller. Each of these registers (R_ACFG, R_ADCRATE, R_ADCACQ, R_SPICF, and OPMODE0 (bit 4, EXTCLK only)) has a register location in RAM that "shadows" the value of the hardware register. To read from a hardware register, the master must first read from the special command register UPD_MIR (A00h) to copy the values from the hardware registers to the mirror registers in RAM, and then the mirror register in RAM can be read. To write to a hardware register, the master reverses the process by writing to the mirror RAM register and then reading from the special command register UPD_SFR (900h) to copy the values from the mirror registers to the hardware registers.
- **Special Command Registers.** These registers (UPD_SFR and UPD_MIR) do not return meaningful data when read but instead trigger an operation. Reading UPD_SFR causes values to be copied from the mirror registers to hardware, and reading UPD_MIR causes values to be copied from the hardware to mirror registers.

Low-Power, Active Energy, Polyphase AFE

Every defined register on the MAXQ3181 has a 12-bit address (from 0 to 4095). This address is used when addressing the register for either a read or write operation. Addresses 0 to 1023 (000h to 3FFh) are used to address RAM registers. Registers with addresses from 1024 to 4095 (400h to FFFh) are used for virtual registers and special command registers.

Each command consists of a read/write command code, a data length (1, 2, 4, or 8 bytes), a 12-bit register address, and the specified number of data bytes followed optionally by a cyclic redundancy check (CRC). Since SPI is a full-duplex interface, the master and slave must both transmit the same number of bytes during the command. When a multiple-byte register is read or written (2/4/8 byte length), the least significant byte is read or written first in the command.

Every transaction begins with the master sending 2 bytes that contain the command (read or write), the address to access, and the number of bytes to transfer. Every SPI peripheral must return 1 byte for every byte it receives. If the master is reading 1 or more bytes from the MAXQ3181, it must send dummy bytes during the cycles when it is receiving a multibyte response to a request, meeting the "send a byte to get a byte" requirement. But the MAXQ3181 could require time to calculate the result, and thus might not have it ready when the master sends the dummy byte. For this reason, the MAXQ3181 always sends zero or more bytes of a NAK character (0x4E or ASCII 'N') followed by an ACK character (0x41, or ASCII 'A') before sending the data.

If the master is writing 1 or more bytes, it sends the data to be written immediately after sending the command. The MAXQ3181 returns ACK (0x41) for each data byte. It then returns NAK (0x4E) until the write cycle is complete, after which it returns a final ACK.

Immediately after the final ACK, the MAXQ3181 is ready to begin the next transaction; there is no need to wait for any other event. It is not even necessary to toggle $\overline{SSEL}$ to begin the next transaction. The MAXQ3181 knows that the first transaction is over and is ready for the next.

If, for whatever reason, it is necessary to reset the communications between the host and the MAXQ3181 (for

Table 1. Command Format for SPI Register Read

BYTE	TRANSFERS	BIT	DESCRIPTION
1st byte	Master sends command; Slave sends 0xC1 byte	7:6	Command Code: 00 Read 01 Reserved 10 Write 11 Reserved
		5:4	Data Length: 00 1 Byte 01 2 Bytes 10 4 Bytes 11 8 Bytes
		3:0	MSB portion of data address.
2nd byte	Master sends address; Slave sends 0xC2 byte	7:0	LSB portion of data address.
Sync bytes	Master sends dummy; Slave sends ACK (0x41) or NACK (0x4E) byte	7:0	Master sends dummy byte; Slave responds with NACK if busy, or with ACK when processing complete. Master must receive ACK, then receive data.
3rd byte (1st data byte)	Master sends dummy; Slave sends data	7:0	Data, LSB
...	...	...	...
Nth byte (Last data byte)	Master sends dummy; Slave sends data	7:0	Data, MSB
(N + 1) byte	Master sends dummy; Slave sends CRC	7:0	Optional CRC

Table 2. Command Format for SPI Register Write

BYTE	TRANSFERS	BIT	DESCRIPTION
1st byte	Master sends command; Slave sends 0xC1 byte	7:6	Command code: 00 Read 01 Reserved 10 Write 11 Reserved
		5:4	Data Length: 00 1 Byte 01 2 Bytes 10 4 Bytes 11 8 Bytes
		3:0	MSB portion of data address.
2nd byte	Master sends address; Slave sends 0xC2 byte	7:0	LSB portion of data address.
3rd byte (1st data byte)	Master sends data; Slave sends ACK (0x41)	7:0	Data, LSB
...	...	...	...
Nth byte (Last data byte)	Master sends data; Slave sends ACK (0x41)	7:0	Data, MSB
(N + 1) byte	Master sends CRC; Slave sends ACK (0x41)	7:0	Optional CRC
Sync bytes	Master sends dummy; Slave sends ACK (0x41) or NACK (0x4E) byte	7:0	Master sends dummy byte; Slave responds with NACK if busy, or with ACK when processing complete. Master must receive ACK before starting the next transaction.

example, if synchronization is lost), the host only needs to wait for the SPI to time out before restarting communication from the first command byte. SPI timeout count starts after receiving the first command byte from the master (after the 8th SPI clock of the first byte). The count stops and clears after receiving the last byte of a transaction (after the 8th SPI clock of the last byte).

If the timeout count expires (exceeds COM_TIMO) before the transaction completes, the MAXQ3181 abandons the unfinished transaction and resets the SPI logic to be ready for the next transaction. The default SPI timeout is 320ms.

Optionally, a CRC byte can be appended to each transaction. For write commands, the CRC byte is sent by the master, and for read commands the CRC byte is sent by the MAXQ3181. The CRC mode is enabled when the CRCEN bit is set to 1 in OPMODE1 register. Otherwise, the MAXQ3181 assumes no CRC byte is used. The 8-bit CRC is calculated for all bytes in a transaction, from the first command byte sent by the master through the last data byte excluding sync bytes, using the polynomial $P = x^8 + x^5 + x^4 + 1$. If the transmitted CRC byte does not match the calculated CRC byte (for a write command), the MAXQ3181 ignores the command.

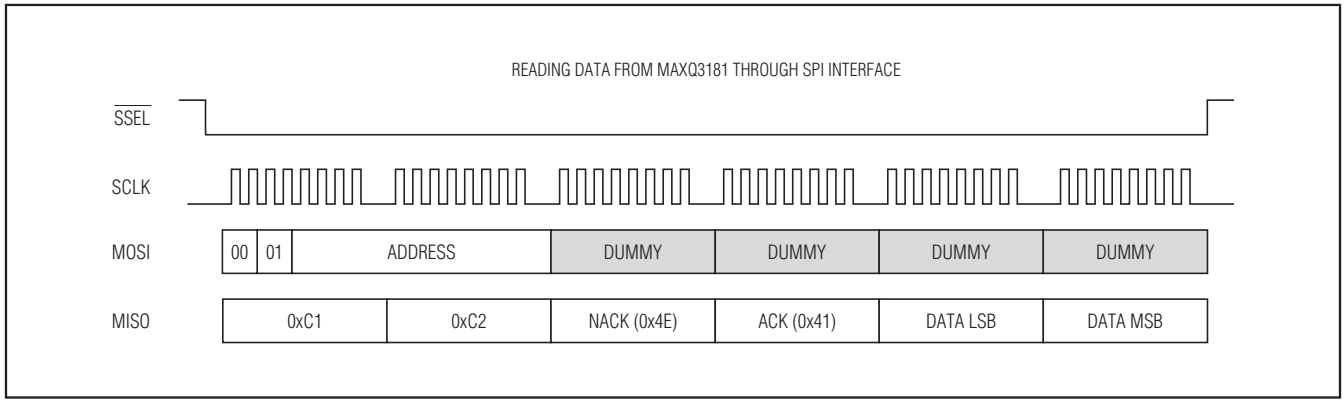


Figure 5. Read SPI Transfer

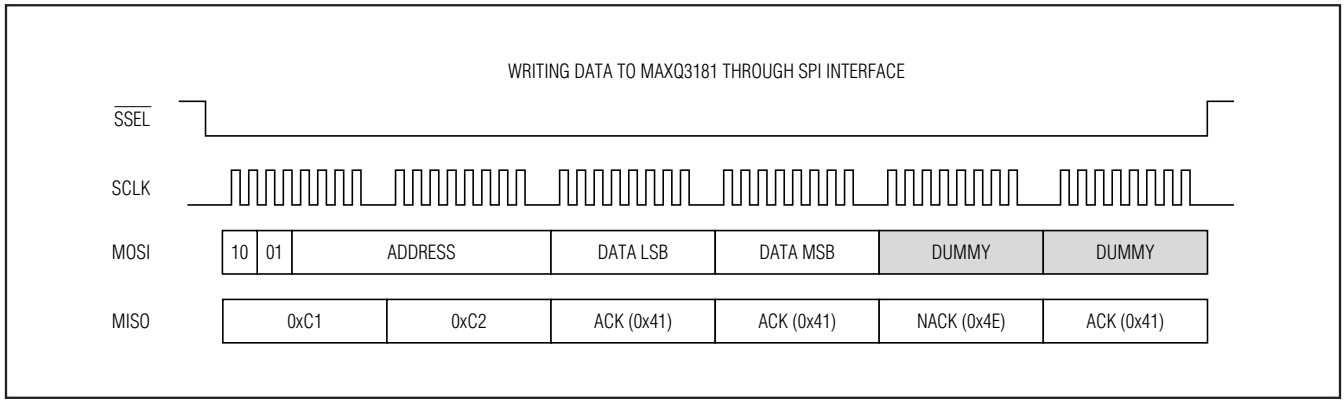


Figure 6. Write SPI Transfer

The length of the transfer is defined by the first command byte and the status of the CRCEN bit in the OPMODE1 register. There is no special synchronization mechanism provided in this simple protocol. Therefore, the master is responsible for sending/receiving the correct number of bytes. If the master mistakenly sends more bytes than are required by the current command, the extra bytes are either ignored (if the MAXQ3181 is busy processing the previous command) or are interpreted as the beginning of a new command. If the master sends fewer bytes than are required by the current command, the MAXQ3181 waits for SPI timeout, then drops the transaction and resets the communication channel. The duration of the timeout can be configured through the COM_TIMO register.

Figures 5 and 6 show typical 2-byte reading and writing transfers (without CRC byte).

Host Software Design

Individual message bytes sent through the SPI are processed in a software routine contained in the ROM firmware. For this reason, it is necessary to provide a delay between successive bytes. This byte spacing must be no less than 400 system clocks to ensure that the MAXQ3181 has a chance to read and process the byte before the arrival of the next one. It is strongly recommended that CRC be enabled for both read and write to achieve reliable communications.

Register Set

Data and device command and control information are located in internal registers. Registers range from 8 to 64 bits in length and are divided into RAM-based registers and virtual registers. The RAM-based registers contain both operating parameters and measurement results.

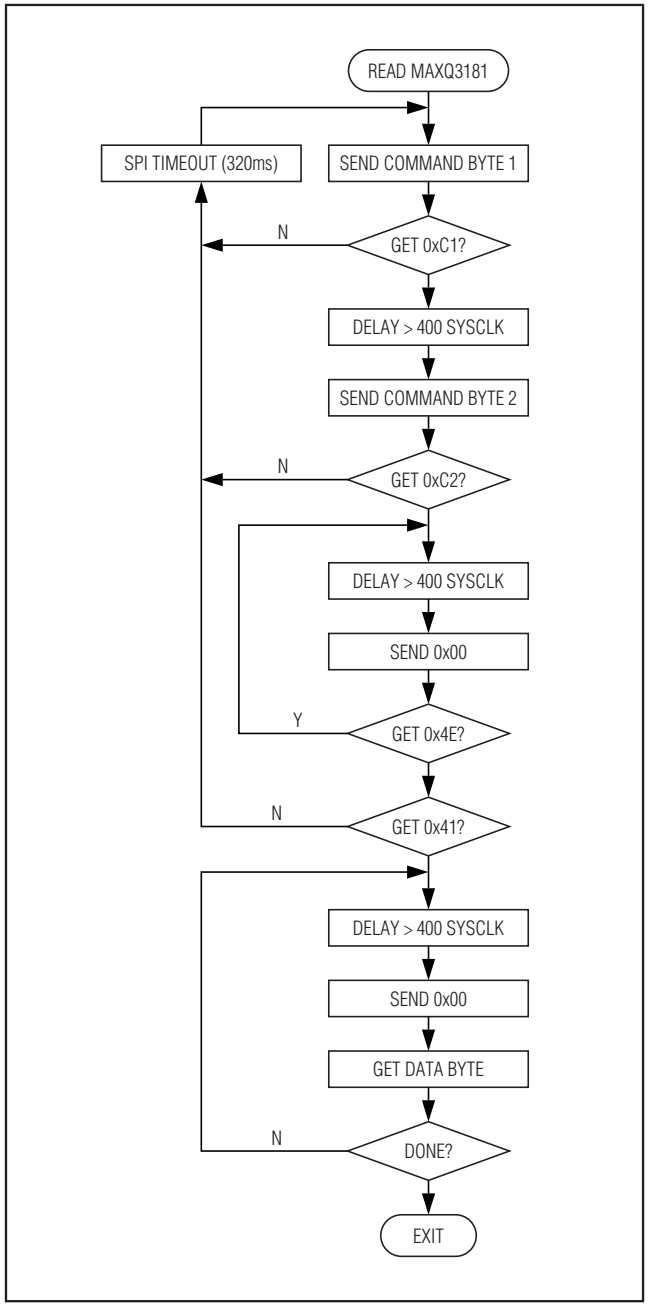


Figure 7. Flowchart for Reading from MAXQ3181

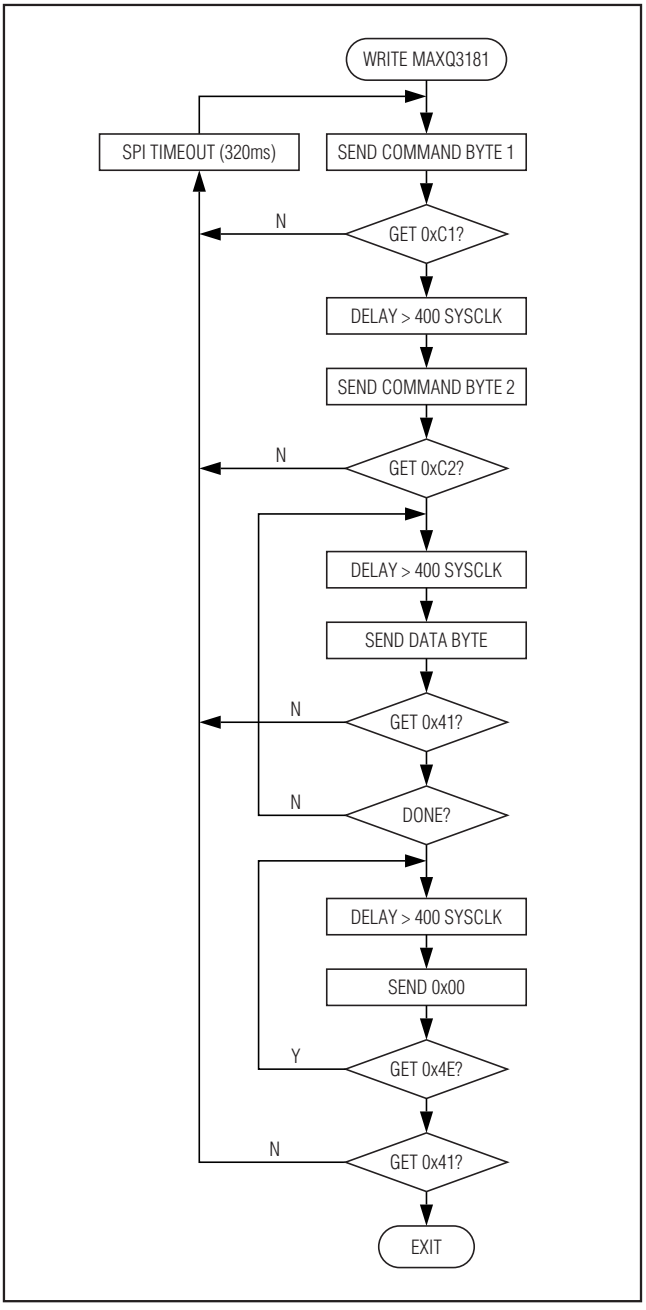


Figure 8. Flowchart for Writing to MAXQ3181

The virtual registers contain calculated values derived from one or more real registers. They are calculated at the time they are requested, and thus can involve additional time to return a value. Most virtual registers are 8 bytes in length and are delivered least significant byte first.

Low-Power, Active Energy, Polyphase AFE

Table 3. RAM Register Map

	x0h	x1h	x2h	x3h	x4h	x5h	x6h	x7h	x8h	x9h	xAh	xBh	xCh	xDh	xEh	xFh
0x00	STATUS	OP MODE1	OP MODE2	OP MODE3	IRQ_FLAG		IRQ_MASK		SCAN _IA	SCAN _VA	SCAN _IC	SCAN _VC	SCAN _IB	SCAN _VB	SCAN _IN	SCAN _TE
0x01	AUX_CFG		SYS_KHZ		VOLT_CC		AMP_CC		PWR_CC		ENR_CC		CYCNT		PLSCFG 1	
0x02	PLS1_WD		THR1										REJ_NS		AVG_NS	
0x03	AVG_C		HPF_C													
0x04	NS				OCLVL		OVLVL		UVLVL		NOLOAD		R_ACFG		R_ADCRATE	
0x05	R_ADCACQ		R_ SPICF		NZX_TIMO		COM_TIMO		ACC_TIMO		ZC_LPF		I1THR		I2THR	
0x06	CHKSUM		LINEFR													
0x11													N.IRMS			
0x12															N.I_GAIN	
PHASE A CONFIGURATION AND STATUS REGISTERS																
0x13	A.I_GAIN		A.V_GAIN		A.E_GAIN				A.OFFS_HI		A.GAIN_LO		A.OFFS_LO		A.PA0	
0x14	A.PA1		A.PA2		A. FLAGS	A. MASK	A. EOVER									
PHASE B CONFIGURATION AND STATUS REGISTERS																
0x21													B.I_GAIN		B.V_GAIN	
0x22	B.E_GAIN				B.OFFS_HI		B.GAIN_LO		B.OFFS_LO		B.PA0		B.PA1		B.PA2	
0x23	B. FLAGS	B. MASK	B. EOVER													
PHASE C CONFIGURATION AND STATUS REGISTERS																
0x30									C.I_GAIN		C.V_GAIN		C.E_GAIN			
0x31	C.OFFS_HI		C.GAIN_LO		C.OFFS_LO		C.PA0		C.PA1		C.PA2		C. FLAGS	C. MASK	C. EOVER	
0x32																
PHASE A MEASUREMENT REGISTERS*																
0x1C							A.PF		A.VRMS				A.IRMS			
0x1D	A.ACT								A.APP							
0x1E									A.EAPOS				A.EANEG			
0x1F									A.ES							
0x20																
PHASE B MEASUREMENT REGISTERS*																
0x2B			B.PF		B.VRMS				B.IRMS				B.ACT			
0x2C					B.APP											
0x2D					B.EAPOS				B.EANEG							
0x2E					B.ES											
0x2F																
PHASE C MEASUREMENT REGISTERS*																
0x39															C.PF	
0x3A	C.VRMS				C.IRMS				C.ACT							
0x3B	C.APP															
0x3C	C.EAPOS				C.EANEG											
0x3D	C.ES															
0x3E																

**Read-only.*

Table 4. Virtual Register Map

	x0	x1	x2	x3	x4	x5	x6	x7
0x80		PWRP.A	PWRP.B		PWRP.C			PWRP.T
0x81								
0x82		PWRS.A	PWRS.B		PWRS.C			PWRS.T
0x83		V.A	V.B		V.C			
0x84	I.N	I.A	I.B		I.C			
0x85								
0x86								PF.T
0x87		ENRS.A	ENRS.B		ENRS.C			ENRS.T
0x88								
0x89								
0x8A								
0x8B								
0x8C		ENRP.A	ENRP.B		ENRP.C			ENRP.T
0x8D								
0x8E								
0x8F								
SPECIAL FUNCTION REGISTERS								
0xC0	DSPVER	RAWTEMP	ENTER STOP	ENTER LOWPM	EXIT LOWPM			

Note: *All virtual registers are read-only.*