19-5327; Rev 0; 6/10

System Management Microcontroller

MAX31782

General Description

The MAX31782 provides a complete solution for the monitoring and controlling of complex system physical health characteristics based on a high-performance, MAXQ20, 16-bit microcontroller core with generous amounts of flash program/data and RAM data memory.

I/O resources include an accurate measurement system for temperature and voltage, PWM outputs, timer inputs, and GPIO to support monitoring and controlling critical system parameters such as temperature, voltage, fan speed, and chassis intrusion. Direct connection of diode-connected transistors used as remote temperature sensors is supported as well as expansion to a virtually unlimited number of external digital temperature sensor ICs using the on-chip master I^{2}C interface. An independent slave I^{2}C interface facilitates communication to a host microprocessor in addition to password-protected in-system reprogramming of the on-chip flash.

Ease of development is supported with highly versatile C-compilers and development software that programs flash and performs in-circuit debug through the integrated JTAG interface and associated hardware.

All these features combined make the device a highly flexible platform, allowing the designer to easily create a customized complex system management solution.

Applications

Network Switches/Routers

Base Stations

Servers

Smart Grid Network Systems

Typical Operating Circuit appears at end of data sheet.

Features

- ♦ **MAXQ20, High-Performance, 16-Bit µC**
- ♦ **Efficient C-Language Programming**
- ♦ **36KWords Total Program Memory**
  - **32KWords Flash Program Memory**
  - **4KWords ROM Program Memory**
- ♦ **1KWords Data RAM**
- ♦ **12-Bit ADC with 7-Input Mux for Temperature and Voltage Monitoring**
- ♦ **Temperature Measurement Analog Front-End**
  - **0.125°C Resolution**
  - **Diode Series Resistance Cancellation**
- ♦ **Six Timer/Fan Tachometer Inputs**
- ♦ **Six 16-Bit PWM Outputs for Fan Speed or D/A Applications**
- ♦ **5-Bit GPIO Ports**
- ♦ **SMBus™/I^{2}C-Compatible Slave Interface for Host Communication with Password-Protected Flash Programming**
- ♦ **I^{2}C-Compatible Master Interface for Slave Expansion**
- ♦ **Power-On Reset and Brownout Monitors**
- ♦ **JTAG Port Supports In-System Debug and Flash Programming**
- ♦ **Internal Oscillator Requires No Crystal**
- ♦ **2.7V to 5.5V Operating Voltage Range**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX31782ETL+	-40°C to +85°C	40 TQFN-EP*
MAX31782ETL+T	-40°C to +85°C	40 TQFN-EP*

+Denotes a lead(Pb)-free/RoHS-compliant package.
T = Tape and reel.
**EP = Exposed pad.*

SMBus is a trademark of Intel Corp.

***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**.*

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

ABSOLUTE MAXIMUM RATINGS

V_{DD} to V_{SS} .... -0.5V to +5.5V
All Other Pins to V_{SS} except
REG18 and REG25 .... -0.5V to (V_{DD} + 0.5V)*
SCL, SDA, MSDA, MSCL, P6.0–P6.4
Continuous Sink Current .... 20mA each, 50mA total

P6.0–P6.4 Continuous Source Current ...20mA each, 50mA total
Operating Temperature Range .... -40°C to +85°C
Storage Temperature Range............................ -55°C to +125°C
Lead Temperature (soldering, 10s) ....+260°C
Soldering Temperature (reflow) ....+260°C

**Subject to not exceeding +5.5V.*

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.

RECOMMENDED OPERATING CONDITIONS

(T_A = -40°C to +85°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{DD} Operating Voltage Range	V_{DD}	(Note 1)	2.7		5.5	V
Input Logic 1	V_{IH}		0.7 x V_{DD}		V_{DD} + 0.3	V
Input Logic 0	V_{IL}		-0.3		+0.3 x V_{DD}	V
Input Logic-High: SCL, SDA, MSDA	V_{I2C_IH}	2.7V ≤ V_{DD} ≤ 3.6V (Note 1)	2.1		V_{DD} + 0.3	V
Input Logic-Low: SCL, SDA, MSDA	V_{I2C_IL}	2.7V ≤ V_{DD} ≤ 3.6V (Note 1)	-0.5		+0.8	V
Input Logic-High: GPIO (Including SCL, SDA, MSCL, and MSDA Under Full V_{DD} Range)	V_{IH1}	(Note 1)	0.7 x V_{DD}			V
Input Logic-Low: GPIO (Including SCL, SDA, MSCL, and MSDA Under Full V_{DD} Range)	V_{IL1}	(Note 1)			0.3 x V_{DD}	V

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current	I_{CPU}	Assuming 100% CPU duty cycle (Note 2)		1.73	2.34	mA
	I_{STOP}	(Note 2)		830	1250	µA
	$I_{PROGRAM}$			7		mA
Brownout Voltage	V_{BO}	Monitors V_{DD} (Note 1)	2.40	2.46	2.55	V
Brownout Hysteresis	V_{BOH}	Monitors V_{DD} (Note 1)		30		mV
Internal System Clock	f_{MOSC}			4.0		MHz
System Clock Error (Note 3)	$f_{ERR:MOSC}$	Initial tolerance, T_A = +25°C, V_{DD} = 5.5V	-1		+1	%
		+25°C ≤ T_A ≤ +85°C	-2		+1	
		-40°C ≤ T_A ≤ +25°C	-5.5		+0.6	

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
System Clock Startup	$t_{SU:MOSC}$	From POR, MOSC inactive		1000		MOSC Cycles
Output Logic-Low: SDA, MSCL, MSDA P6.0–P6.4, PWM.0–PWM.5	V_{OL1}	I_{OL} = 4mA (Note 1)			0.4	V
Output Logic-High: P6.0–P6.4, PWM.0–PWM.5	V_{OH1}	I_{OH} = -2mA (Note 1)	V_{DD} - 0.5			V
GPIO Mode Pullup Current	I_{PU}	$V_{PIN} = V_{SS}$, V_{DD} = 3.3V	38	55	107	µA
ADC Voltage Conversion Time	t_{CONV_V}	(Note 4)		128	150	µs
ADC Temperature Conversion Time	t_{CONV_T}	(Note 4)			7	ms
ADC Internal Reference				1.225		V
ADC Voltage Measurement Error	V_{ERR}		-1		+1	%
ADC Internal Reference Temperature Drift			-0.5		+0.5	%
ADC Internal Reference Initial Accuracy (+25°C)			-1		+1	mV
ADC External Reference Buffer Accuracy		(Note 5)		±0.25		%
ADC Operating Current	I_{ADC}	This current is in addition to I_{CPU}			2.2	mA
ADC Full-Scale Input Voltage (Note 6)	V_{FS}	ADGAIN = 0, factory set, internal reference	1.213	1.225	1.237	V
		ADGAIN = 1, factory set, internal reference	5.445	5.5	5.555	
ADC Measurement Resolution	V_{LSB}	ADGAIN = 0		300		µV
		ADGAIN = 1		1343		
ADC Bit Resolution			12			Bits
AD0P–AD5P Input Resistance	R_{IN}		15			MΩ
ADC Integral Nonlinearity	INL				±4	LSB
ADC Differential Nonlinearity	DNL				±1	LSB
ADC Offset	V_{OFFSET}			±2		LSB
Internal Temperature Measurement Error		T_A = -40°C to +85°C	-3		+3	°C
Remote Temperature Measurement Error (MAX31782 Error Only)		T_A = 0°C to +60°C, T_{DIODE} = +60°C to +120°C	-1.5		+1.5	°C
		T_A = 0°C to +60°C, T_{DIODE} = -45°C to +120°C	-1.75		+1.75	
		T_A = -40°C to +85°C, T_{DIODE} = +60 to +120°C	-2.75		+2.75	
		T_A = -40°C to +85°C, T_{DIODE} = -45°C to +120°C	-3.0		+3.0	

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{DD} = 3.3V, T_A = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Flash Erase Time	t_{ME}	Mass erase	20		40	ms
	t_{PE}	Page erase	20		40	
Flash Programming Time per Word	t_{PROG}		20		40	µs
Flash Endurance	N_{FLASH}	T_A = +50°C	20,000			Write Cycles
Data Retention		T_A = +50°C	100			Years

ELECTRICAL CHARACTERISTICS: I^{2}C-COMPATIBLE INTERFACE

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted.) (Figure 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}	Timeout not enabled (Note 7)	10		400	kHz
Bus Free Time Between a STOP and START Condition	t_{BUF}		1.3			µs
Hold Time (Repeated) START Condition	$t_{HD:STA}$	(Note 8)	0.6			µs
Low Period of SCL Clock	t_{LOW}		1.3			µs
High Period of SCL Clock	t_{HIGH}		0.6			µs
Setup Time for a (Repeated) START Condition	$t_{SU:STA}$		0.6			µs
Data Hold Time (Note 9)	$t_{HD:DAT}$	Receive	0			ns
		Transmit	300			
Data Setup Time	$t_{SU:DAT}$		100			ns
Rise Time of Both SDA and SCL Signals	t_R	(Note 10)	20 + 0.1C_B		300	ns
Fall Time of Both SDA and SCL Signals	t_F	(Note 10)	20 + 0.1C_B		300	ns
Setup Time for STOP Condition	$t_{SU:STO}$		0.6			µs
Spike Pulse Width That Can Be Suppressed by Input Filter	t_{SP}	(Note 11)	0		50	ns
SCL, SDA Capacitive Loading	C_B				400	pF
SMBus Timeout			25	30	35	ms

ELECTRICAL CHARACTERISTICS: JTAG INTERFACE

(V_{DD} = 2.7V to 5.5V, T_A = -40°C to +85°C, unless otherwise noted.) (Figure 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
JTAG Logic Reference	V_{REF}			$V_{DD}/2$		V
TCK High Time	t_{TH}		1			µs
TCK Low Time	t_{TL}		1			µs
TCK Low to TDO Output	t_{TLQ}				0.125	µs
TMS, TDI Input Setup to TCK High	t_{DVTH}		0.30			µs
TMS, TDI Input Hold after TCK High	t_{THDX}		0.25			µs

Note 1: All voltages are referenced to ground (V_{SS}). Currents entering the IC are specified positive and currents exiting the IC are negative.

Note 2: This value does not include current in SDA, SCL, and P6.0–P6.4.

Note 3: Guaranteed by design.

Note 4: ADCCLK = SYSCLK/16. This is following an initial conversion time of approximately 80µs.

Note 5: Base line accuracy of reference source + 0.25% introduced by the MAX31782.

Note 6: The voltage applied to the pins must not exceed their corresponding absolute maximum voltages.

Note 7: Minimum SCL frequency applies only when in I^{2}C master mode.

Note 8: After this period, the first clock pulse can be generated.

Note 9: This device internally provides a hold time of at least 25ns for the SDA signal (referenced to the V_{IHMIN} of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 10: C_B—Total capacitance of one bus line in pF.

Note 11: Filters on SDA and SCL suppress noise spikes at the input buffers and delay the sampling instant.

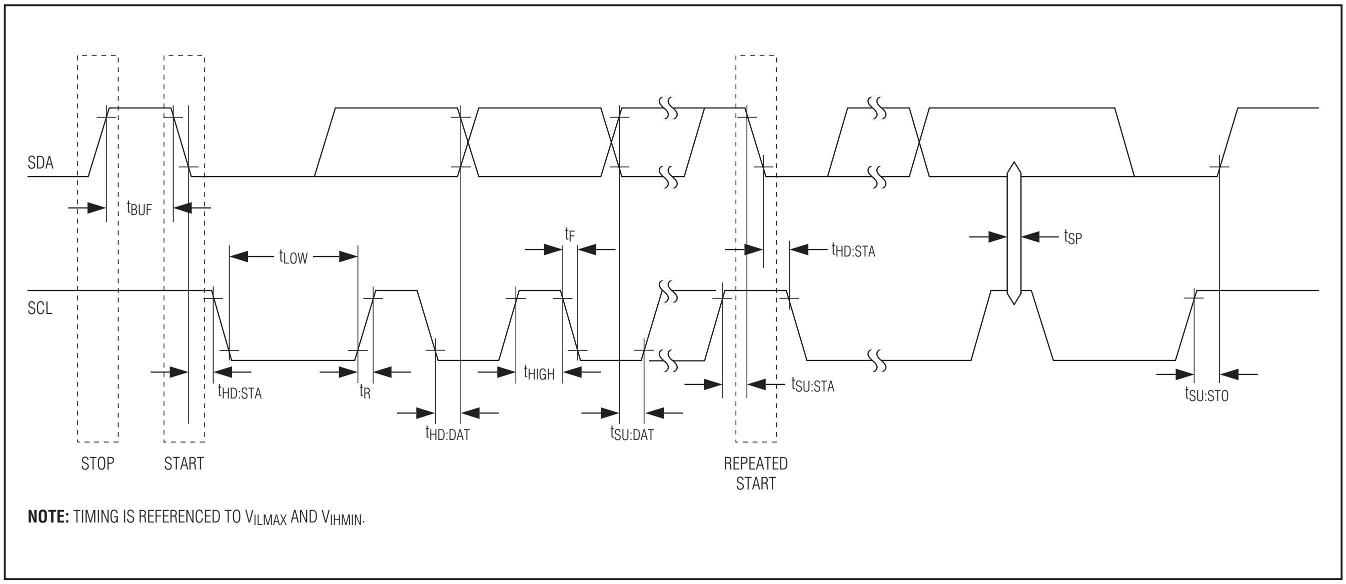


Figure 1. I²C-Compatible Bus Timing Diagram

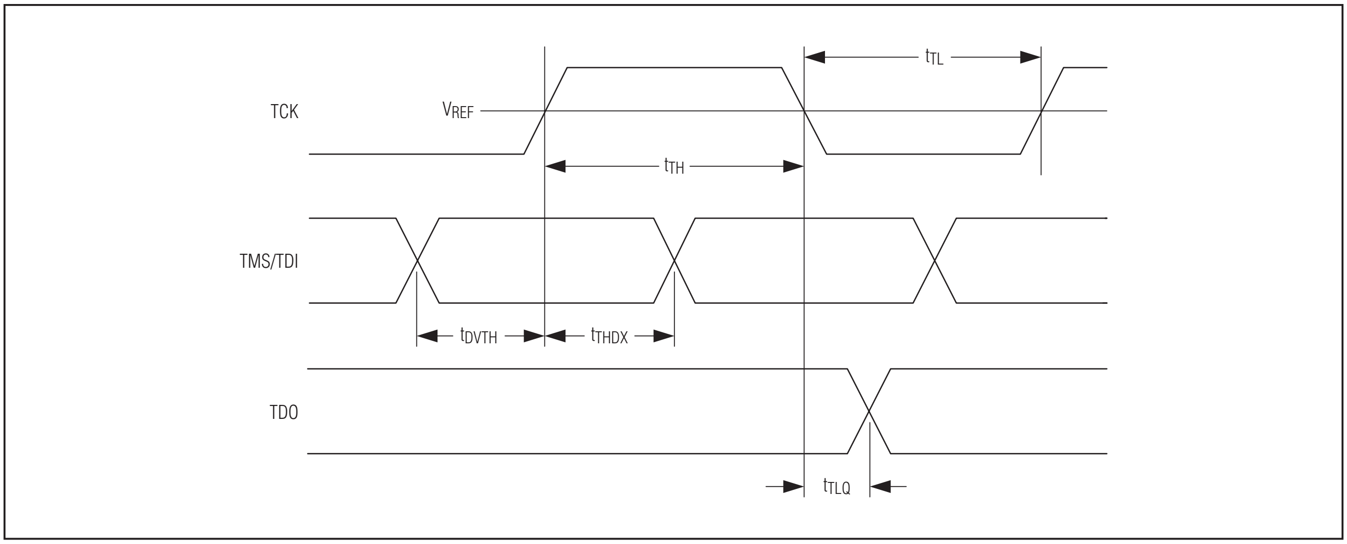
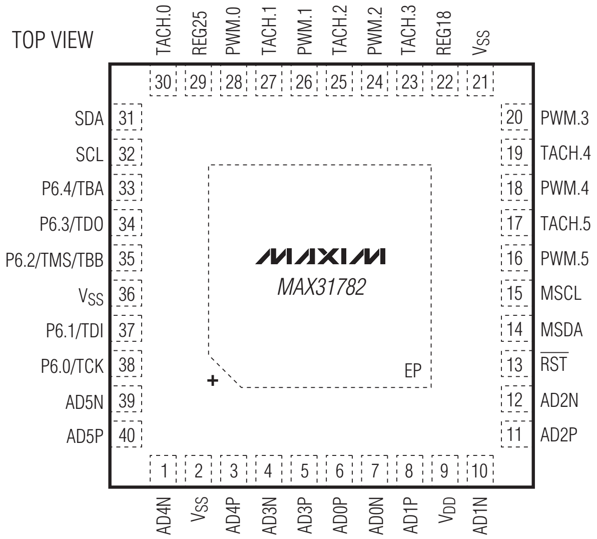


Figure 2. JTAG Timing Diagram

Pin Configuration



TQFN
(6mm × 6mm × 0.75mm)

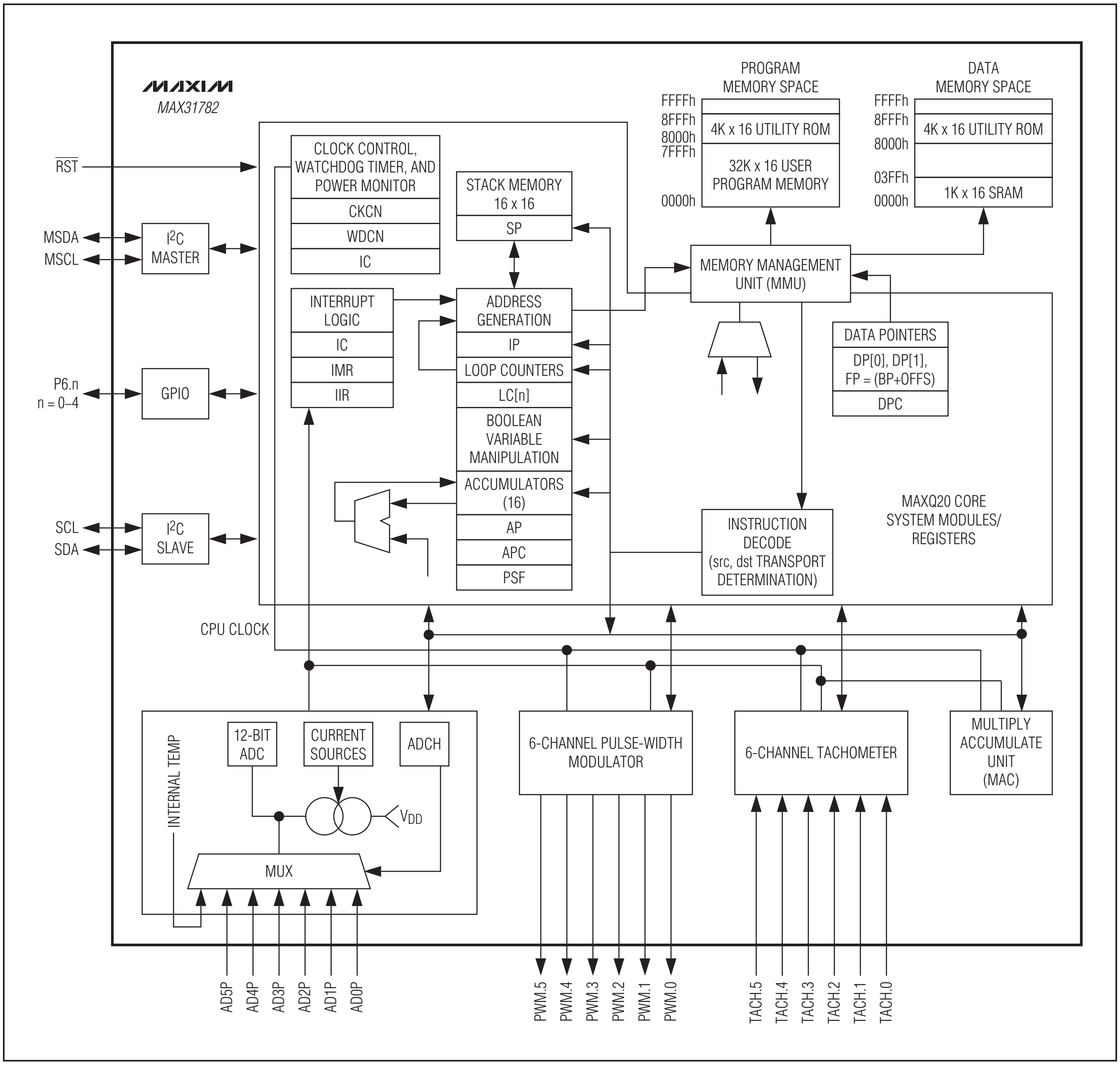
Pin Description

PIN	NAME	FUNCTION
1	AD4N	Ground Reference for ADC.4 Voltage Measurement
2, 21, 36	V_{SS}	Supply Return Node
3	AD4P	ADC Voltage-Sense Input, Measurement Relative to AD4N
4	AD3N	Ground Reference for ADC.3 Voltage Measurement. Connected to external reference pin when enabled.
5	AD3P	ADC Voltage-Sense Input, Measurement Relative to AD3N
6	AD0P	ADC Voltage-Sense Input, Measurement Relative to AD0N
7	AD0N	Ground Reference for ADC.0 Voltage Measurement
8	AD1P	ADC Voltage-Sense Input, Measurement Relative to AD1N
9	V_{DD}	Input Supply. +2.7V to +5.5V input range. Bypass V_{DD} to V_{SS} with a 0.1µF capacitor.
10	AD1N	Ground Reference for ADC.1 Voltage Measurement
11	AD2P	ADC Voltage-Sense Input, Measurement Relative to AD2N
12	AD2N	Ground Reference for ADC.2 Voltage Measurement
13	$\overline{RST}$	Active-Low Reset. A low-level voltage at this pin resets the IC.
14	MSDA	Master I^{2}C-Compatible Bidirectional Data Line. When disabled, this pin can be used as GPIO P2.7.
15	MSCL	Master I^{2}C-Compatible Clock. When disabled, this pin can be used as GPIO P2.6.
16	PWM.5	PWM Output No. 5. When disabled, this pin can be used as GPIO.
17	TACH.5	Tachometer Input No. 5. When disabled, this pin can be used as GPIO.
18	PWM.4	PWM Output No. 4. When disabled, this pin can be used as GPIO.
19	TACH.4	Tachometer Input No. 4. When disabled, this pin can be used as GPIO.

Pin Description (continued)

PIN	NAME	FUNCTION
20	PWM.3	PWM Output No. 3. When disabled, this pin can be used as GPIO.
22	REG18	Bypass REG18 to V_{SS} with 1µF and high-frequency 10nF capacitors. Do not use for external circuitry.
23	TACH.3	Tachometer Input No. 3. When disabled, this pin can be used as GPIO.
24	PWM.2	PWM Output No. 2. When disabled, this pin can be used as GPIO.
25	TACH.2	Tachometer Input No. 2. When disabled, this pin can be used as GPIO.
26	PWM.1	PWM Output No. 1. When disabled, this pin can be used as GPIO.
27	TACH.1	Tachometer Input No. 1. When disabled, this pin can be used as GPIO.
28	PWM.0	PWM Output No. 0. When disabled, this pin can be used as GPIO.
29	REG25	Bypass REG25 to V_{SS} with 1µF and high-frequency 10nF capacitors. Do not use for external circuitry.
30	TACH.0	Tachometer Input No. 0. When disabled, this pin can be used as GPIO.
31	SDA	Slave I^2C-Compatible Bidirectional Data Line. When disabled, this pin can be used as GPIO.
32	SCL	Slave I^2C-Compatible Clock. When disabled, this pin can be used as GPIO.
33	P6.4/TBA	Programmable I/O Pin. Alternate function: Timer/Counter TBA
34	P6.3/TDO	Programmable I/O Pin. Alternate function: JTAG TDO
35	P6.2/TMS/TBB	Programmable I/O Pin. Alternate functions: Timer/Counter TBB, JTAG TMS
37	P6.1/TDI	Programmable I/O Pin. Alternate function: JTAG TDI
38	P6.0/TCK	Programmable I/O Pin. Alternate function: JTAG TCK
39	AD5N	Ground Reference for ADC.5 Voltage Measurement
40	AD5P	ADC Voltage-Sense Input, Measurement Relative to AD5N
—	EP	Exposed Pad. Not electrically connected to IC. Connect to V_{SS}.

Block Diagram



Detailed Description

The MAX31782 incorporates the 16-bit MAXQ20 microcontroller core with 16 accumulators and 16-level hardware stack. Three memory blocks provide flash application code space, utility ROM code space, and RAM memory. Specialized peripherals are integrated to perform PWM control of fan speed, read fan tachometers, and perform temperature monitoring using diode-connected transistors. The device also features two I^{2}C-compatible communication peripherals. The slave I^{2}C-compatible peripheral is included to allow communication between a host system and the device. An I^{2}C-compatible master interface is also included to allow communication with remote I^{2}C digital temperature sensors or other I^{2}C devices. General-purpose I/O pins (GPIOs) are also provided to allow interrupt functions and control of other circuitry using the system management microprocessor. The MAXQ20 core, along with the specialized peripherals, provides a flexible solution for system and thermal management. Flexibility is further enhanced as the solution allows for upgrading the program and data flash contents over the I^{2}C-compatible interface. Updates to the program flash are protected against unauthorized writes by a 256-bit user password.

The following sections are an introduction to the primary features of the MAX31782 system management microcontroller. More detailed descriptions of the device features can be found in the user's guides described in the *Additional Documentation* section.

MAXQ20 Core Architecture

The device employs a MAXQ20 low-cost, high-performance, CMOS, fully static, 16-bit RISC microcontroller with flash memory. It is structured on a highly advanced, 16-accumulator-based, 16-bit RISC architecture. Fetch and execution operations are completed in one cycle without pipelining, since the instruction contains both the op code and data. The highly efficient core is supported by 16 accumulators and a 16-level hardware stack, enabling fast subroutine calling and task switching. Data can be quickly and efficiently manipulated with three internal data pointers. Multiple data pointers allow more than one function to access data memory without having to save and restore data pointers each time. The data pointers can automatically increment or decrement following an operation, eliminating the need for software intervention.

Instruction Set

The instruction set is composed of fixed-length, 16-bit instructions that operate on registers and memory locations. The instruction set is highly orthogonal, allowing arithmetic and logical operations to use any register along with the accumulator. Special-function registers control the peripherals and are subdivided into register modules. The family architecture is modular, so that new devices and modules can reuse code developed for existing products.

The architecture is transport-triggered. This means that writes or reads from certain register locations can also cause side effects to occur. These side effects form the basis for higher level op codes defined by the assembly, such as ADDC, OR, JUMP, etc. The op codes are implemented as MOVE instructions between certain register locations, while the assembler handles the encoding, which need not be a concern to the programmer. The 16-bit instruction word is designed for efficient execution.

Bit 15 indicates the format for the source field of the instruction. Bits 0–7 of the instruction represent the source for the transfer. Depending on the value of the format field, this can either be an immediate value or a source register. If this field represents a register, the lower 4 bits contain the module specifier and the upper 4 bits contain the register index in that module.

Bits 8–14 represent the destination for the transfer. This value always represents a destination register, with the lower 4 bits containing the module specifier and the upper 3 bits containing the register subindex within that module. Any time it is necessary to directly select one of the upper 24 registers as a destination, the prefix register, PFX, is needed to supply the extra destination bits. This prefix register write is inserted automatically by the assembler and requires only one additional execution cycle. Refer to the *MAXQ Family User's Guide* for complete instruction set information.

Memory Organization

The device incorporates several memory areas, including:

- 32KWords of flash memory for application program storage
- 1KWords of SRAM for storage of temporary variables
- 4KWords of utility ROM contain a debugger and program loader
- 16-level stack memory for storage of program return addresses and general-purpose use

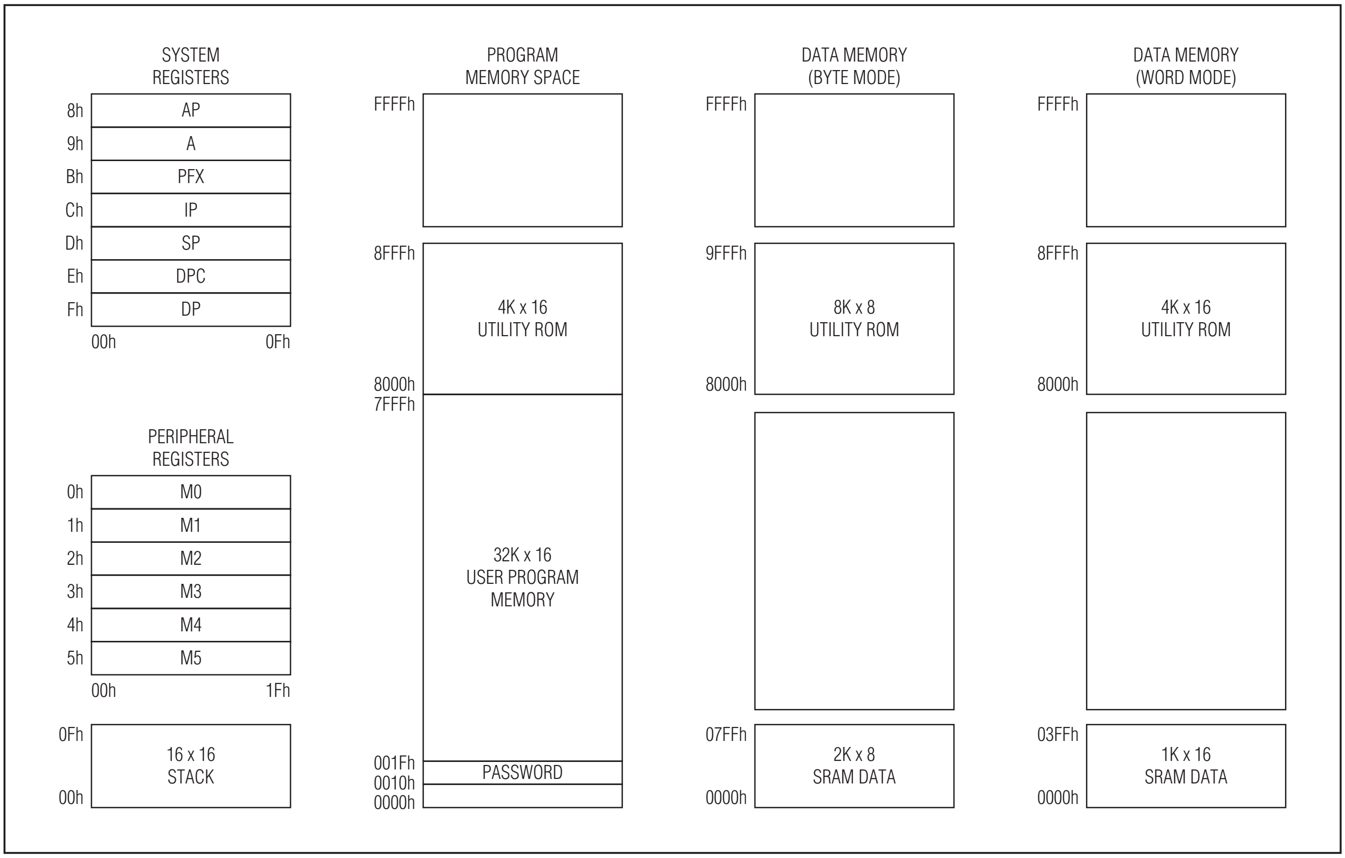


Figure 3. Memory Map

The memory is implemented using the Harvard architecture, with separate address spaces for program and data memory. A pseudo-Von Neumann memory map is also used placing ROM, application code, and data memory into a single contiguous memory map. The pseudo-Von Neumann memory map allows data memory to be mapped into program space, permitting code execution from data memory. In addition, program memory can be mapped into data space, permitting code constants to be accessed as data memory. Figure 3 shows the device's memory map when executing from program memory space. Refer to the *MAXQ Family User's Guide: MAX31782 Supplement* for memory map information when executing from data or ROM space.

The incorporation of flash memory allows field upgrade of the firmware. Flash memory is password protected with a 16-word key, denying access to program memory by unauthorized individuals.

Utility ROM

The utility ROM is a 4KWord block of internal ROM memory that defaults to a starting address of 8000h. The utility ROM consists of subroutines that can be called from application software. These include the following:

- In-system programming (bootstrap loader) over JTAG or I^{2}C-compatible interfaces
- In-circuit debug routines
- Callable routines for in-application flash programming

Following any reset, execution begins in the utility ROM. The ROM software determines whether the program execution should immediately jump to location 0000h, the start of application code, or to one of the special routines mentioned. Routines within the utility ROM are firmware-accessible and can be called as subroutines by the application software. More information on the utility ROM contents is contained in the *MAXQ Family User's Guide: MAX31782 Supplement*.

Password

The device is programmed with a default password prior to being shipped. The password is defined as the 16 words of physical program memory at addresses 0010h–001Fh. A single password lock bit (PWL) is implemented in the SC register. Once a new device is programmed, a password is defined (password is other than all zeros or all ones) and the PWL bit is set. If the PWL is zero, the device is deemed unprogrammed. The password is automatically set to all ones following a mass erase.

Stack Memory

A 16-bit, 16-level internal stack provides storage for program return addresses and general-purpose use. The stack is used automatically by the processor when the CALL, RET, and RETI instructions are executed and interrupts serviced. The stack can also be used explicitly to store and retrieve data by using the PUSH, POP, and POPI instructions.

On reset, the stack pointer, SP, initializes to the top of the stack (0Fh). The CALL, PUSH, and interrupt-vectoring operations increment SP, and then store a value at the location pointed to by SP. The RET, RETI, POP, and POPI operations retrieve the value at SP and then decrement SP.

Programming

The flash memory of the microcontroller can be programmed by one of three methods: in-system programming, in-application programming, and production programming. All three methods provide great flexibility in system design and reduce the life-cycle cost of the embedded system.

In-System Programming

An internal bootstrap loader allows the device to be programmed over the JTAG or I^{2}C-compatible interfaces. As a result, system software can be upgraded in-system, eliminating the need for a costly hardware retrofit when software updates are required.

Programming source select (PSS) bits in the ICDF register determine which interface is used for bootloading operation. The device supports JTAG and I^{2}C as an interface corresponding to 00 and 01 bits of PSS, respectively.

In-Application Programming

The in-application programming feature allows the microcontroller to modify its own flash program memory. This allows on-the-fly software updates in mission-critical applications that cannot afford downtime. Alternatively, it allows the application to develop custom loader software that can operate under the control of the application software. The utility ROM contains firmware-accessible flash programming functions that erase and program flash memory. These functions are described in detail in the *MAXQ Family User's Guide: MAX31782 Supplement*.

System Timing

The device generates its 4MHz instruction clock (MOSC) internally using a ring oscillator. On power-up, the output of the oscillator (which cannot be accessed externally) is disabled until V_{DD} rises above V_{BO}. Once this threshold is reached, 1000 cycles are counted (~ 250µs) and then the output is enabled, clocking the device.

System Reset

The device features several sources that can be used to reset the device.

Power-On Reset

An internal power-on-reset (POR) circuit is used to enhance system reliability. This circuit forces the device to perform a POR whenever a rising voltage on V_{DD} climbs above V_{POR}. When this happens the following events occur:

- All registers and circuits enter their reset state
- The POR flag (WDCN.7) is set to indicate the source of the reset
- Code execution begins at location 8000h when the reset condition is released

Brownout Detect/Reset

The device features a brownout-detect/reset function. Whenever the power monitor detects a brownout condition (when $V_{DD} < V_{BO}$), it immediately issues a reset and stays in that state as long as V_{DD} remains below V_{BO}. Once V_{DD} voltage rises above V_{BO}, the device waits for $t_{SU:MOSC}$ before returning to normal operation, also referred to as CPU state. If a brownout occurs during $t_{SU:MOSC}$, it again goes back to the brownout state. Otherwise, it enters into CPU state. In CPU state, the brownout detector is also enabled. On power-up, the device always enters into brownout state first and then follows the previously mentioned sequence. The reset issued by brownout is the same as POR. Whatever action happens on POR also happens on brownout reset. All the registers that are cleared on POR are also cleared on brownout reset.

Table 1. Power Modes

CKCN. STOP	SVM. SVMEN	SVM. SVMSTOP	CPU	LDO (1.8V)	INTERNAL OSCILLATOR	BROWNOUT DETECT	SVM MONITOR	LDO (1.8V) MONITOR	ADC	POWER MODE	LDO (2.5V)
0	0	X	On	On	On	On	Off	On	On/Off	CPU Mode	On
0	1	X	On	On	On	On	On	On	On/Off	CPU Mode	On
1	0	X	Off	On	On	On	Off	On	Off	Stop Mode	On
1	1	0	Off	On	On	On	Off	On	Off	Stop Mode	On
1	1	1	Off	On	On	On	On	On	Off	Stop Mode	On

Watchdog Timer Reset

The watchdog timer provides a mechanism to reset the processor in the case of undesirable code execution. The watchdog timer is a hardware timer designed to be periodically reset by the application software. If the software operates correctly, the timer is reset before it reaches its maximum count. However, if undesirable code execution prevents a reset of the watchdog timer, the timer reaches its maximum count and resets the processor.

The watchdog timer is controlled through 2 bits in the WDCN register (WDCN[5:4]: WD[1:0]). Its timeout period can be set to one of the four programmable intervals ranging from 2^{12} to 2^{21} system clock (MOSC) periods (1.024ms to 0.524s). The watchdog interrupt occurs at the end of this timeout period, which is 512 MOSC clock periods, or 128µs, before the reset. The reset generated by the watchdog timer lasts for four system clock cycles, which is 1µs. Software can determine if a watchdog time caused a reset by checking the watchdog timer reset flag (WTRF) in the WDCN register. Execution resumes at location 8000h following a watchdog timer reset.

External Reset

Asserting the $\overline{\text{RST}}$ pin low causes the device to enter the reset state. The external reset function is described in the *MAXQ Family User's Guide*. Execution resumes at location 8000h after the $\overline{\text{RST}}$ pin is released.

Internal System Reset

In I^{2}C bootload mode, the host can issue a BBh command to reset the communicating device using an I^{2}C slave address of 34h. This reset has the same effect as the external reset as far as the reset values of all registers are concerned. Also, an internal system reset can occur when the in-system programming is done (ROD = 1).

Power Modes

The device supports two modes of operation: CPU mode and stop mode. The device enters stop mode state after a CPU STOP (CKCN.STOP) is asserted. On entering stop mode, the digital core is inactive as its clock is turned off. All the analog circuits, except ADC (including SVM, LDOs, and monitor circuits), are still active. Stop mode is exited by any of the following: an external interrupt on port 6, an I^{2}C START interrupt, an SVM interrupt, or an external reset. For one of the mentioned interrupts to get the device out of stop mode, it must be enabled. The system returns to CPU mode within 10 system clocks. If an interrupt causes the system to come out of stop mode, the program execution starts from the point where stop mode was asserted. However, if an external reset is used to come out of stop mode, the program execution begins from ORG (starting point). Table 1 explains the state of analog/digital circuits during different modes.

Register Set

Most functions of the device are controlled by sets of registers. These registers provide a working space for memory operations as well as configuring and addressing peripheral registers on the device. Registers are divided into two major types: system registers (SPRs) and peripheral registers (SFRs). The common register set, also known as the system registers, includes the ALU, accumulator registers, data pointers, interrupt vectors and control, and stack pointer. The peripheral registers define additional functionality and the functionality is broken up into discrete modules. Both the system registers and the peripheral registers are illustrated in detail in the *MAXQ Family User's Guide: MAX31782 Supplement*.

Hardware Multiplier

The hardware multiplier (a multiply-accumulate, or MAC module) is a very powerful tool, especially for applications that require heavy calculations. This multiplier can execute the multiply, multiply-negate, or multiply-accumulate, multiply-subtract operation for signed or unsigned operands in a single machine cycle, and even faster for special cases. The MAC module uses eight SFRs, mapped as register 0h–07h in module M5.

System Interrupts

Multiple interrupt sources are available to respond to internal and external events. The MAXQ20 architecture uses a single interrupt vector (IV) and single interrupt-service routine (ISR) design. For maximum flexibility, interrupts can be enabled globally, individually, or by module. When an interrupt condition occurs, its individual flag is set, even if the interrupt source is disabled at the local, module, or global level. Interrupt flags must be cleared within the firmware-interrupt routine to avoid repeated interrupts from the same source. Application software must ensure a delay between the write to the flag and the RETI instruction to allow time for the interrupt hardware to remove the internal interrupt condition. Asynchronous interrupt flags require a one-instruction delay and synchronous interrupt flags require a two-instruction delay. When an enabled interrupt is detected, execution jumps to a user-programmable interrupt vector location. The IV register defaults to 0000h on reset or power-up, so if it is not changed to a different address, application firmware must determine whether a jump to 0000h came from a reset or interrupt source.

Once control has been transferred to the ISR, the Interrupt Identification register (IIR) can be used to determine if a system register or peripheral register was the source of the interrupt. In addition to IIR, MIIR registers are implemented to indicate which particular function under a peripheral module has caused the interrupt. The device contains six peripheral modules, M0–M5. An MIIR register is implemented under each module. The MIIR registers are 16-bit read-only registers and all of them default to all 0 on system reset. Once the module that causes the interrupt is singled out, it can then be interrogated for the specific interrupt source and software can take appropriate action. Interrupts are evaluated by application code allowing the definition of a unique interrupt priority scheme for each application. Interrupt sources are available from the watchdog timer, the ADC, the TACH.n pins, the programmable timer/counter, the I^{2}C-compatible master and slave interface, the SVM, and the port 6 I/O pins.

Programmable Timer/Counter

The device features a general-purpose programmable timer/counter commonly referred to as a Timer B module. The specification for this timer/counter block is the same as the Timer B specification. There are four registers associated with this timer/counter block: TB0CN (control register), TB0V (value register), TB0C (compare register), and TB0R (capture/reload value register). The timer/counter has two pins, TBA and TBB, that are multiplexed with pins P6.4 and P6.2, respectively. When TBA or TBB is enabled, the corresponding pin functions as a timer/counter pin instead of a GPIO. See the *I/O Port* section for more details. Detailed information regarding the timer/counter block can be found in the *MAXQ Family User's Guide: MAX31782 Supplement*.

I/O Port

The device includes a simple input/output (I/O) data port, port 6. Pins P6.0–P6.4 are primary GPIO pins with alternate functions. Each pin is multiplexed with at least one special function, such as interrupts, timer/counter I/O pins, or JTAG pins. Table 2 summarizes the functionality of the I/O pins. Figure 4 shows a block diagram of the I/O port.

Port 6 pins have Schmitt trigger receivers and full CMOS output drivers, and can support alternate functions. The port is accessed through six SFRs (PO6, PI6, PD6, EIE6, EIF6, and EIES6) in module 1 and each pin can be individually configured. The pin is either high impedance or a weak pullup when defined as an input, dependent on the state of the corresponding bit in the output register. In addition, each pin can function as external interrupt with individual enable, flag, and active edge selection, when programmed as input.

On power-up, pins P6.0–P6.3 default to JTAG. Clearing SC.TAP to 0 (1 is the power-up state) configures them as GPIO. Setting EIE6.n (n = 0–4, 6, 7) to 1 configures P6.n to an interrupt.

Pins P6.2 and P6.4 have special functions that are the timer/counter's TBB and TBA pins, respectively. When TBB or TBA or both are enabled, P6.2 or P6.4 or both are used as their special functions. P6.2 and P6.4 are independent when used as timer/counter pins, i.e., when either of them is used as a timer/counter pin, the other can still be used as GPIO if the corresponding special function is not enabled.

Table 2. I/O Port Pins

PORT INDEX	PRIMARY FUNCTION	ALTERNATE FUNCTION	INTERRUPTS	TAP (JTAG)	RESET STATE
P6.0	GPIO, P6.0	—	INT0	TCK	TCK
P6.1	GPIO, P6.1	—	INT1	TDI	TDI
P6.2	GPIO, P6.2	Timer B TBB Pin	INT2	TMS	TMS
P6.3	GPIO, P6.3	—	INT3	TDO	TDO
P6.4	GPIO, P6.4	Timer B TBA Pin	INT4	—	GPIO input with weak pullup

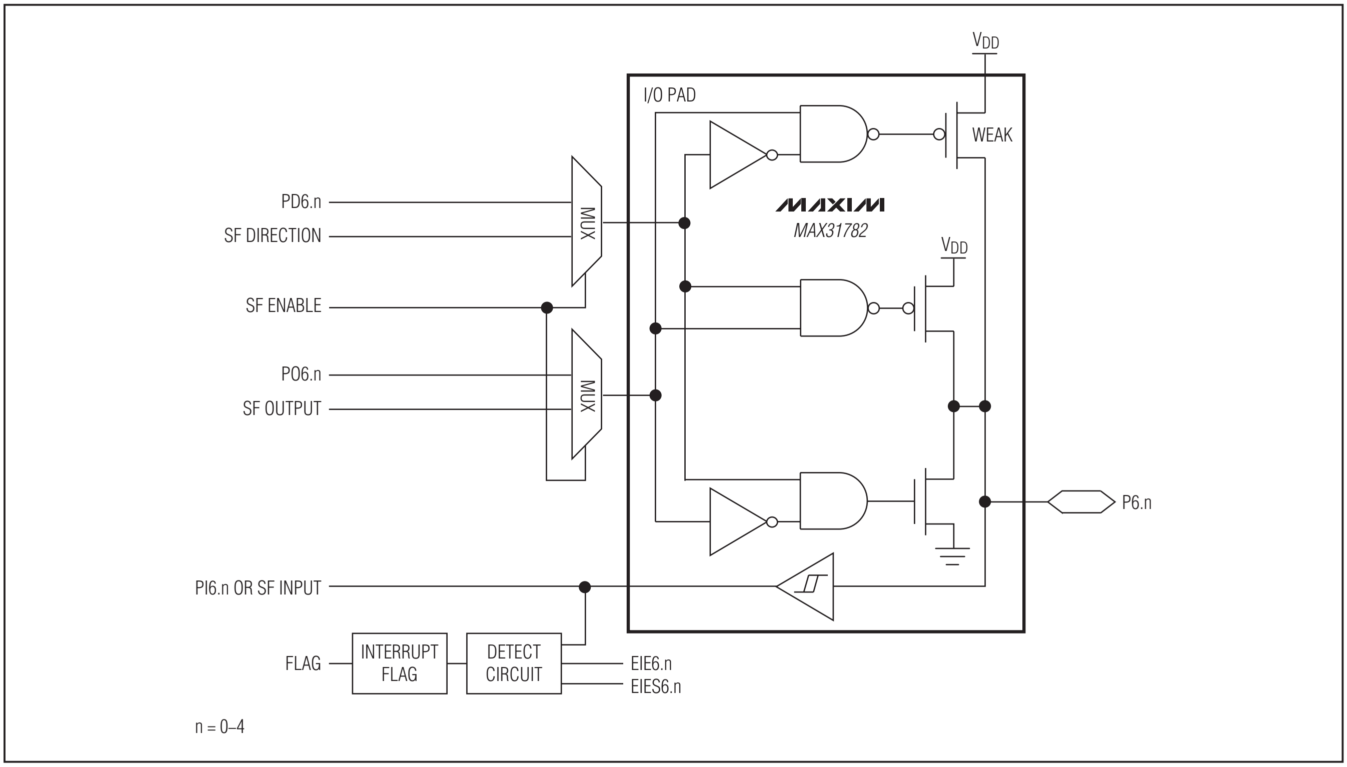


Figure 4. Port 6 I/O Block Diagram

PWM Outputs

The device provides six independent PWM outputs. Each PWM output is associated with four SFRs: PWMCNn, PWMVn, PWMRn, and PWMCn, where n = 0–5 is the channel number. The PWM clock is derived from the system clock with a division ratio defined by PWMCNn. The PWMCNn register also enables/disables the PWM output and selects the PWM polarity. The user can set the frequency and the duty cycle of each PWM output individually by configuring the corresponding PWMRn register and the PWMCn register, respectively.

When the PWM output functionality of a PWM.n pin is disabled, that pin can be used as a GPIO. When used as GPIO pins, PWM.n pins are accessed as Port 1 and through three SFRs: PO1, PI1, and PD1. Each PWM.n pin can be independently configured, and can be defined as an input with weak pullup, an input without pullup, or an output.

Tachometer Inputs

The device provides six pins for reading fan tachometer pulses. Each TACH.n pin functions independently and is associated with three SFRs: TACHCNn (the control register), TACHVn (the timer value register), and TACHRn (the timer capture register), where n = 0–5 is the channel number.

There is an internal timer for each TACH.n pin. The clock for the TACH.n timer is derived from the system clock with a division ratio defined by TACHCNn. The TACH.n timer, when initially enabled, begins counting up from the TACHVn value and upon overflow subsequently continues counting from 0000h to the FFFFh overflow, i.e., rolls over from FFFFh to 0000h if left enabled and running. If the capture function is enabled by configuring the TACHCNn register, a 1-to-0 transition on the prescaled tachometer pulses causes the value in the TACHVn register to be transferred into the TACHRn register and set the external trigger flag. Upon capture, TACHVn reloads 0000h and continues counting. The user can calculate the tachometer pulse period and the fan speed by reading the TACHRn register.

When the tachometer input functionality of a TACH.n pin is disabled, that pin can be used as a GPIO. When used as GPIO pins, TACH.n pins are accessed as port 2 and through three SFRs: PO2, PI2, and PD2. Each TACH.n pin can be independently configured, and can be defined as an input with weak pullup, an input without pullup, or an output.

I^{2}C-Compatible Interface Modules

The device provides two independent I^{2}C-compatible interfaces; one is a master and the other is a slave.

I^{2}C-Compatible Master Interface

The device features an internal I^{2}C-comaptible master interface for communication with a wide variety of external I^{2}C devices. The I^{2}C-compatible master bus is a bidirectional bus using two bus lines, the serial-data line (MSDA) and the serial-clock line (MSCL). For the I^{2}C-compatible master, the device has ownership of the I^{2}C bus, and drives the clock and generates the START and STOP signals. This allows the device to send data to a slave or receive data from a slave as required. Both the MSDA and MSCL lines must be driven as open-drain outputs. External pullup resistors are required to pull the lines to a logic-high state.

When the I^{2}C-compatible master interface is disabled, MSDA and MSCL can be used as GPIO pins. When used as GPIO pins, MSDA and MSCL can be used as pins P2.7 and P2.6, respectively, and are accessed through three SFRs: PO2, PI2, and PD2. Because these pins are open drain, external pullups are required to realize a logic-high.

I^{2}C-Compatible Slave Interface

The device also features an internal I^{2}C-comaptible slave interface for communication with a host. Furthermore, the device can be in-system programmed (bootloaded) through the I^{2}C-compatible slave interface. For the I^{2}C-compatible slave interface, the device relies on an externally generated clock to drive SCL and responds to data and commands only when requested by the I^{2}C master device.

SMBus Timeout

Both the I^{2}C-compatible master and slave interfaces can work in SMBus-compatible mode for communication with other SMBus devices. To achieve this, a 30ms timer has been implemented on the I^{2}C-compatible slave interface to make the interface SMBus compatible. The purpose of this timer is to issue a timeout interrupt and thus the firmware can reset the I^{2}C-compatible slave interface when the SCL is held low for longer than 30ms. The timer only starts when **none** of the following conditions is true:

- The I^{2}C-compatible slave interface is in the idle state and there is no communication on the bus.
- The I^{2}C-compatible slave interface is not working in SMBus-compatible mode.
- The SCL logic level is high.
- The I^{2}C-compatible slave interface is disabled.

When a timeout occurs, the timeout bit is set and an interrupt is generated, if enabled. If a timeout interrupt is generated, the firmware disables and reenables the I^{2}C-compatible slave interface. After this process, the SCL and SDA pins are set to high impedance. All the relevant I^{2}C slave SFRs should be reloaded by firmware.

Analog-to-Digital Converter (ADC)

The device contains a 12-bit analog-to-digital converter (ADC) with a 7-input mux (Figure 5). The mux selects the ADC input from six external channels and one internal channel. The six external channels can operate in fully differential voltage mode or in single-ended voltage mode. In addition, any of the six external channels can be configured to measure the temperature of an external diode. The internal channel is used exclusively to measure the die temperature. The ADC is controlled by SFR registers.

The ADC can be set up to continuously poll the input channels (continuous-sequence mode) or run a short burst of conversions and enter a shutdown mode to conserve power (single-sequence mode).

The six external channels can be individually configured to operate in external temperature mode. In external temperature mode, current is forced into an external diode that is connected between user-specified channel pins. The diode temperature is obtained by measuring the diode voltages at multiple bias currents. The device features a 3-point series resistance-cancellation algorithm to provide high-temperature measurement accuracy. The ADC is able to measure the external diode temperature immune to the loop resistance. For both external and internal temperature measurements, the internal reference is automatically selected and the full-scale (FS) value is fixed at 1.225V. The temperature measurement resolution is 0.125°C.

When the external channels are configured to operate in voltage mode, the voltage applied on the corresponding channel (differential or single-ended) is converted to a digital readout. In voltage mode, the reference can be either internal or external. If the internal reference is used, the FS can be set to 1.225V or 5.5V. These FS values can be trimmed by modifying the associated registers (ADCG1 and ADCG5), respectively.

In voltage mode, an ADC conversion takes 34 ADCCLK cycles to complete. The ADCCLK is derived from the system clock with division ratio defined by the ADC Control register. The fastest ADC sampling rate is SYSCLK/544. With a 4MHz system clock, this is theoretically equivalent to 7.35ksps. In applications where extending the acquisition time is desired, the sample can be acquired over a prolonged period determined by the ADC Control register.

The ADC has eight configuration registers. Each channel can have its own configuration, such as differential mode select, data alignment select, acquisition extension enable, ADC reference select, and external temperature mode select, etc. The ADC also has sixteen 13-bit circular data buffers for conversion result storage. The ADC data available interrupt flag (ADDAI) can be configured to trigger an interrupt following a predetermined number of samples. Once set, ADDAI can be cleared by software or at the start of a conversion process.

When the device is put into stop mode, any in-progress ADC conversion is aborted and the ADC start conversion bit (ADCONV) is reset to 0. The ADC is shut down completely to conserve power. On exiting stop mode, the ADC waits on ADCONV = 1. When ADCONV is set to 1, it counts 20 ADCCLK cycles before acquisition commences.

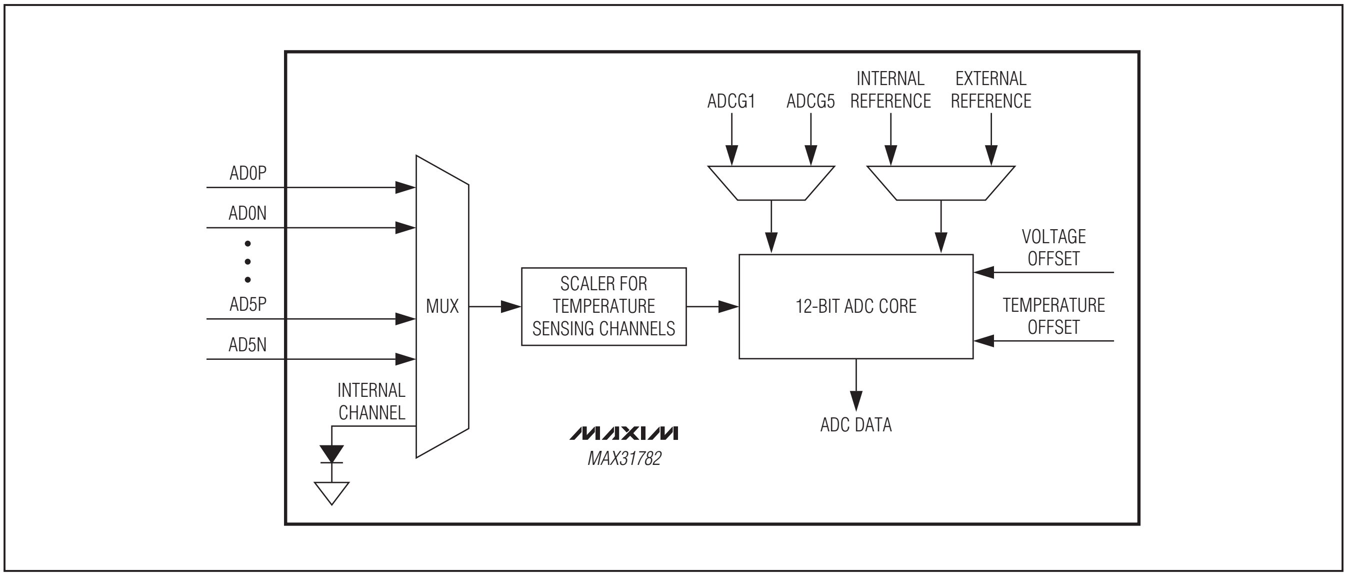


Figure 5. ADC Block Diagram

In-Circuit Debug

Embedded debugging capability is available through the JTAG-compatible test access port (TAP). Embedded debug hardware and embedded ROM firmware provide in-circuit debugging capability to the user application, eliminating the need for an expensive in-circuit emulator. Figure 6 shows a block diagram of the in-circuit debugger. The in-circuit debug features include the following:

- Hardware debug engine
- Set of registers able to set breakpoints on register, code, or data accesses (ICDA, ICDB, ICDC, ICDD, ICDF, ICDT0, and ICDT1)
- Set of debug service routines stored in the utility ROM

The embedded hardware debug engine is an independent hardware block in the microcontroller. The debug engine can monitor internal activities and interact with selected internal registers while the CPU is executing user code. Collectively, the hardware and software features allow two basic modes of in-circuit debugging: background and debug.

Background mode allows the host to configure and set up the in-circuit debugger while the CPU continues to execute the application software at full speed. Debug mode can be invoked from background mode.

Debug mode allows the debug engine to take control of the CPU, providing read/write access to internal registers and memory, and single-step trace operation.

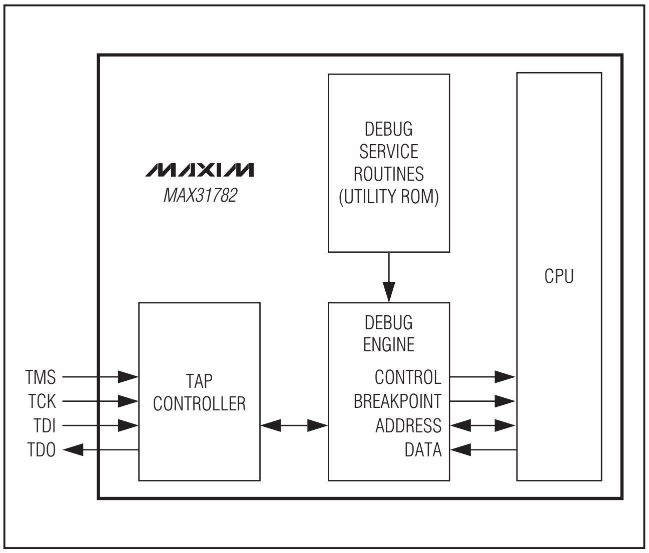


Figure 6. In-Circuit Debugger

Applications Information

Power-Supply Decoupling

To achieve the best results when using the device, decouple the V_{DD} power supply with a 0.1µF capacitor. Use a high-quality, ceramic, surface-mount capacitor if possible. Surface-mount components minimize lead inductance, which improves performance, and ceramic capacitors tend to have adequate high-frequency response for decoupling applications.

Decouple REG25 and REG18 using 1µF and 10nF capacitors (one each per output). **Note:** Do not use either of these pins for external circuitry.

Additional Documentation

Designers must have four documents to fully use all the features of this device. This data sheet contains pin descriptions, feature overviews, and electrical specifications. Errata sheets contain deviations from published specifications. The user's guides offer detailed information about device features and operation. The following documents can be downloaded from **www.maxim-ic.com**.

- The MAX31782 data sheet, which contains electrical/timing specifications and pin descriptions.
- The MAX31782 revision-specific errata sheet (**www.maxim-ic.com/errata**).
- The *MAXQ Family User's Guide*, which contains detailed information on core features and operation, including programming.
- The *MAXQ Family User's Guide: MAX31782 Supplement*, which contains detailed information on features specific to the MAX31782.

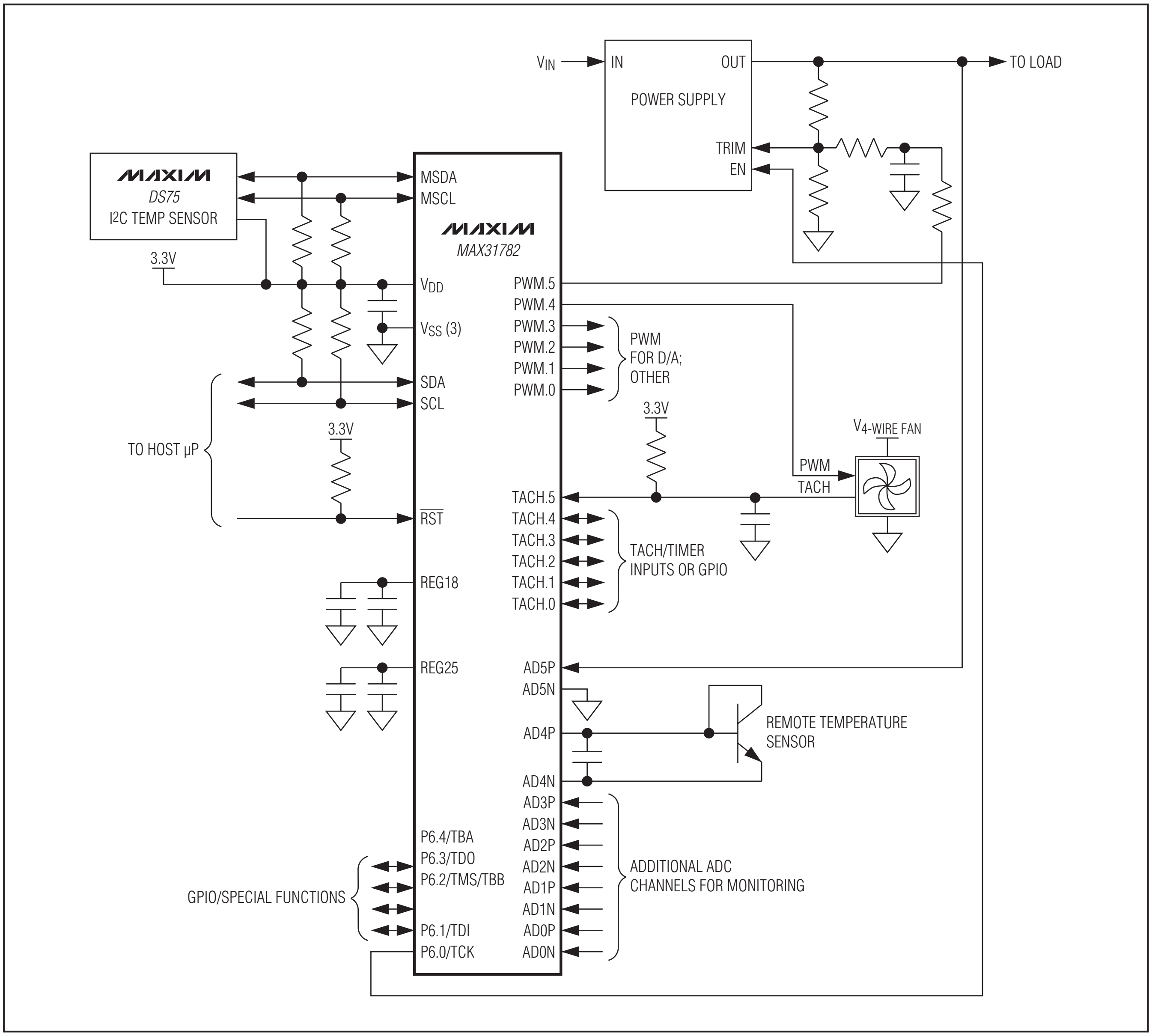
Development and Technical Support

Maxim and third-party suppliers provide a variety of highly versatile, affordably priced development tools for this microcontroller, including the following:

- Compilers (C and assembly)
- In-circuit debugger
- Integrated development environments (IDEs)
- Serial-to-JTAG converters for programming and debugging
- USB-to-JTAG converters for programming and debugging

Technical support is available through email at **mixedsignal.apps@maxim-ic.com**.

Typical Operating Circuit



Package Information

For the latest package outline information and land patterns, go to **www.maxim-ic.com/packages**. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
40 TQFN-EP	T4066+2	21-0141	90-0053

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	6/10	Initial release	—

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.