DS2792
Programmable Fuel Gauge with UART Interface

www.maxim-ic.com

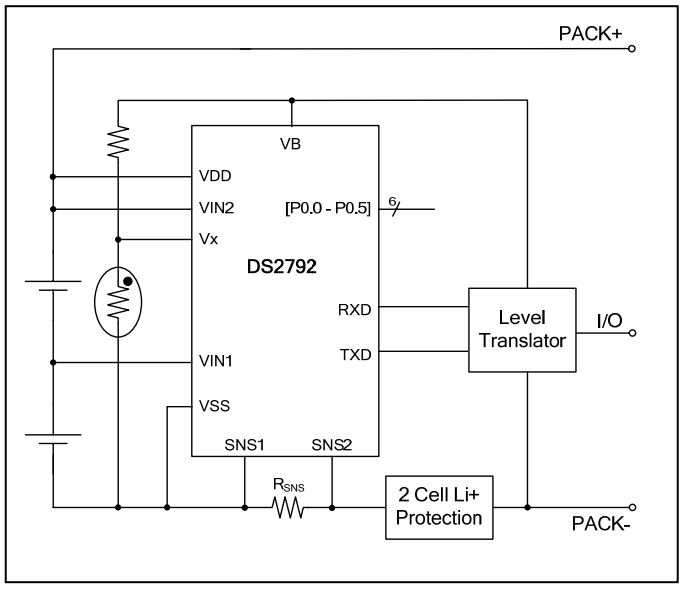
GENERAL DESCRIPTION

The DS2792 provides a user-programmable fuel gauge solution for Li-Ion and NiMH chemistry battery packs. A low-power, 16-bit MAXQ20 microcontroller with generous program and data memory, combined with an accurate measurement system for battery current, voltage, and temperature provide the ideal platform for customized fuel-gauge algorithms. EEPROM data memory supports nonvolatile (NV) in-pack storage of charge parameters, cell characteristics, usage history, and manufacturing/lot tracking data.

APPLICATIONS

Digital Video Cameras
SLR Digital Still Cameras
Subnotebook PCs and Ultra-Portable PCs
Industrial PDAs, Handheld Computers, and GPS

FUNCTIONAL DIAGRAM



MAXQ is a registered trademark of Maxim Integrated Products, Inc.

FEATURES

- **Accurate Current Measurement for Coulomb Counting (Current Accumulation)**
  1.5% ±4µV Over ±64mV Input Range
  1.5% ±267µA Over ±4.2A Range Using an External 15mΩ Series Resistor
- **High-Resolution Current Reporting**
  12-Bit + Sign Average Every 0.88ms
  15-Bit + Sign Average Every 2.8s
- **Three Voltage Measurement Sources**
  10-Bit Average from VIN1, VIN2–VIN1, and Vx Inputs
- **Temperature Measurement**
  10-Bit Using On-Chip Sensor
  Ratiometric Input for External Thermistor (Vx)
- **16-Bit MAXQ20 Low-Power Microcontroller**
  Efficient C-Language Programming
  8k Words Total Program Memory:
  4k Words EEPROM Program Memory
  4k Words ROM Program Memory
  64 Words Data EEPROM
  256 Words Data RAM
  Password-Protected Programming
- **On-Chip, Low Drop-Out Regulator**
  2.5V to 10V Operating Range
- **SHA-1 Hash Algorithm in ROM**
- **19.2kbps UART Interface**
- **Internal Oscillator: No Crystal Required**
- **Low-Power Consumption**
  1.5mA CPU Mode (1MHz), 145µA ANALOG Mode, 50µA SLEEP Mode

ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE
DS2792G+	-20ºC to +70ºC	TDFN-28

+ Denotes lead-free package.
Contact factory concerning Mask ROM devices.

Pin Configuration appears at end of data sheet.

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, click here: www.maxim-ic.com/errata.*

ABSOLUTE MAXIMUM RATINGS

V_{DD}, VIN2 to V_{SS} ..... -0.3V to +12V
P0.4, P0.5 to V_{SS} ..... -0.3V to V_B +0.3V
AV_{SS} to V_{SS} ..... -0.3V to +0.3V
All Other Pins to V_{SS} ..... -0.3V to +6V
TXD, P0.0–P0.5 Continous Sink Current ..... 20mA Each, 50mA Total
Operating Temperature Range ..... -40ºC to +85ºC
Storage Temperature Range ..... -55ºC to +125ºC
Soldering Temperature ..... See 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 beyone those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device.

RECOMMENDED DC OPERATING CHARACTERISTICS

(V_{DD} = 2.5V to 10V, T_A = -20°C to +70°C, unless otherwise noted. Typical values are at T_A = +25°C, V_{DD} = 5.0V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	(Note 1)	+2.5		+10	V
Voltage Range: VIN1, TXD, RXD, P0.0–P0.3, SNS1, SNS2		(Note 1)	-0.3		5.5	V
Voltage Range: P0.4–P0.5		(Note 1)	-0.3		V_B + 0.3	V
Voltage Range: VIN2		(Note 1)	-0.3		+10	V
Voltage Range: Vx		(Note 1)	-0.3		V_B + 0.3	V
Output Voltage: V_B	V_{VB}	V_{DD} > 3.5V, I_O = 2mA, (Note 1)	3.0	3.3	3.6	V

ELECTRICAL CHARACTERISTICS

(V_{DD} = 2.5V to 10V, T_A = -20°C to +70°C, unless otherwise noted. Typical values are at T_A = +25°C, V_{DD} = 5.0V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current	I_{SLEEP}	SLEEP mode (Note 2)		25	50	µA
	I_{SUSP}	SUSPEND mode (Note 3)		25	50	µA
	I_{ANALOG}	ANALOG mode (Note 4)		110	145	µA
	I_{CPU}	CPU mode (Note 5)		0.8	1.5	mA
Power-On Reset Threshold	V_{RESET}		1.0	1.6	2.2	V
Brownout Threshold	V_{BO}	(Note 10)	2.0	2.2	2.4	V
Regulator Drop-Out	$V_{DO:VB}$	V_{DD} = 2.5V, I_{VB} = 2.0mA, (Note 6)			0.15	V
Current Measurement Input Range	I_{FS}	V_{IS1}–V_{IS2}	-64		+64	mV
Current Measurement Resolution	I_{LSB}			15.625		$\mu V/R_{SNS}$
Current Measurement Gain Error	I_{GERR}	0°C ≤ T_A ≤ +50°C	-0.5		+0.5	% Full Scale
			-1		+1	
Current Measurement Offset Error	I_{OERR}		-7.8		+7.8	$\mu V/R_{SNS}$
Accumulated Current Range	q_{FS}		-204.8		+204.8	mVh/R_{SNS}
Accumulated Current Resolution	q_{LSB}			6.25		$\mu Vh/R_{SNS}$
Accumulated Current Offset	q_{CA}	OBEN = 1	-94		0	µVh/Day
		OBEN = 1, R_{SNS} = 0.015Ω	-6.3		0	mAh/Day
Temperature Measurement Range	T_{FS}		-40		+85	°C
Temperature Measurement LSb	T_{LSB}			0.125		°C
Temperature Measrement Error	T_{ERR}		-3		+3	°C

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
VIN1 Input Range (VIN1–V_{SS})	V_{FS1}	(Notes 1, 7)	0		4.99	V
VIN1 LSb	V_{LSB1}			4.88		mV
VIN2 Input Range (VIN2–VIN1)	V_{FS2}	(Notes 1, 7)	0		4.99	V
VIN2 LSb	V_{LSB2}			4.88		mV
VIN1, VIN2 Gain Error	V_{GERR}		-1		+1	%
VIN1, VIN2 Offset	V_{OERR}		-1		+1	LSb
Vx Input Range (Vx–V_{SS})	V_{FSX}	(Note 1)	0		V_B	V
Vx LSb	V_{LSBX}			V_B/1024		—
Vx Error	V_{ERRX}		-1		+1	% Full Scale
VIN1, VIN2, Vx Input Resistance	R_{IN}		15			MΩ
Current Measurement Sample Frequency	f_{SAMPLE}	(Note 8)		1456		Hz
Analog System Clock Frequency	f_{OSCA}			69.9		kHz
Analog System Clock Error	$f_{ERR:OSCA}$	V_{DD} > 2.7V, T_A = +25°C	-0.7		+0.7	%
		V_{DD} > 2.7V, 0°C ≤ T_A ≤ +50°C	-2		+2	
			-5		+5	
CPU System Clock Startup Time	$t_{SU:OSCI}$	OSCA active		1		µS
		From SLEEP, OSCA inactive		700		
CPU System Clock Frequency	f_{OSCI}	OSCA inactive		1000		kHz
		OSCA active		14 x f_{OSCA}		kHz
CPU System Clock Error	$f_{ERR:OSCI}$	OSCA inactive	-20		+20	%
		OSCA active		$f_{ERR:OSCA}$		
Suspend Period Error	$t_{ERR:SUS}$		-30		+30	%
Filter Resistors IS1 to SNS1, IS2 to SNS2	R_{KS}		7	10	13	kΩ
Input Logic High: RXD	$V_{IH:RXD}$	(Note 1)	1.5			V
Input Logic Low: RXD	$V_{IL:RXD}$	(Note 1)			0.6	V
Input Logic High: P0.0–P0.5	$V_{IH:P0}$	(Note 1)	0.7 x V_B			V
Input Logic Low: P0.0–P0.5	$V_{IL:P0}$	(Note 1)			0.3 x V_B	V
Output Logic Low: TXD, P0.X	V_{OL}	IOL = 4mA (Note 1)			0.4	V
P0.0–P0.3 Weak Pullup Current	$I_{PU:P0}$	V_{PIN} = V_{IH}, V_{DD} > 2.7V Bits PPU:0,1,2,3 set	0.15		22.0	µA
Output Logic High: P0.4–P0.5	V_{OH}	I_{PIN} = 1mA Bits PPU:4,5 set	V_B - 0.4			V
RXD, TXD Pulldown Current	$I_{PD:UART}$	V_{PIN} = V_{IL}, Bits PPU:6,7 clear	0.3	1.2	3	µA
RXD, TXD Pullup Current	$V_{PU:UART}$	V_{PIN} = V_{IL}, Bits PPU:6,7 set	0.3	1.2	3	µA
RXD, TXD Capacitance	C_{UART}			50		pF
RXD Pulse Rejection	$t_{SP:UART}$	Rising and falling edges (Note 9)			50	ns
P0.0–P0.5 Pulse Rejection	$t_{SP:P0}$	Rising and falling edges			10	ns

EEPROM RELIABILITY SPECIFICATION

(V_{DD} = 2.5V to 10V, T_A = -20°C to +70°C, unless otherwise noted. Typical values are at T_A = +25°C, V_{DD} = 5.0V.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
EEPROM Copy Time	t_{EEC}			10	15	ms
EEPROM Copy Endurance	N_{EEC}	T_A = +50°C	50,000			Cycles

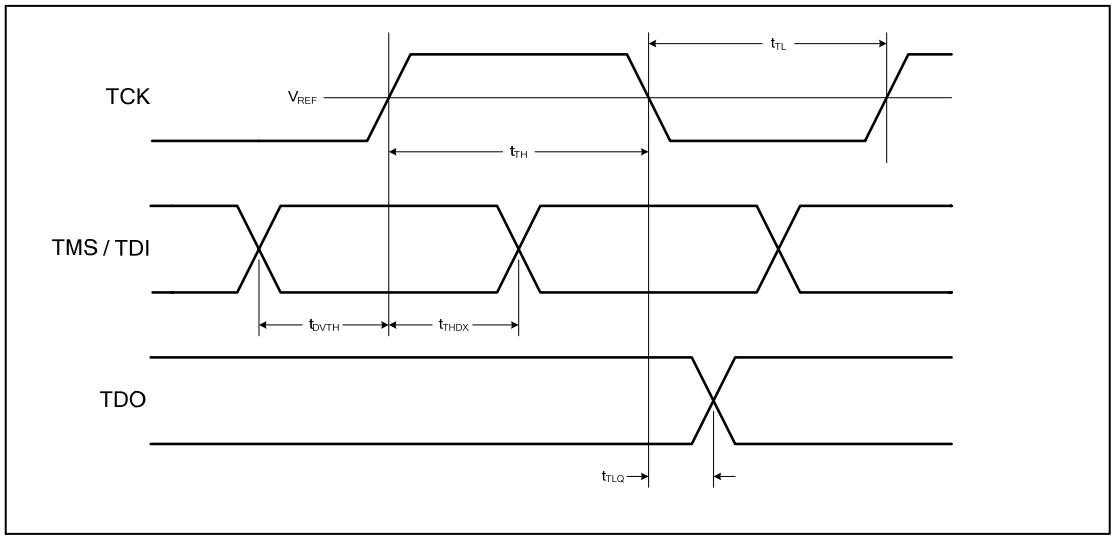
ELECTRICAL CHARACTERISTICS: JTAG INTERFACE

(V_{DD} = 2.5V to 10V, T_A = -20°C to +70°C, unless otherwise noted. Typical values are at T_A = +25°C, V_{DD} = 5.0V.)
(See Figure 1.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
JTAG Logic Reference	V_{REF}	(Note 1)		$V_B \div 2$		V
TCK High Time	t_{TH}		4.0			µs
TCK Low Time	t_{TL}		4.0			µs
TCK Low to TDO Output	t_{TLQ}				1.0	µs
TMS, TDI Input Setup to TCK High	t_{DVTH}		1.0			µs
TMS, TDI Input Hold after TCK High	t_{THDX}		4.0			µs

Note 1: All voltages referenced to V_{SS}.
Note 2: Internal voltage regulator remains active in SLEEP mode. RAM and registers are powered to maintain contents. RXD and internal interrupts can be armed by firmware.
Note 3: Internal voltage regulator and suspend timer are active in SUSPEND mode. RAM and registers are powered to maintain contents.
Note 4: Internal voltage regulator and ADC are active in ANALOG mode. RAM and registers are powered to maintain contents. ADC data is collected and updated to registers including current accumulation to ACR register.
Note 5: MAXQ core fetches and executes instructions in CPU mode.
Note 6: Parameters guaranteed by design.
Note 7: Voltage A/D readings saturate at 4.85V.
Note 8: $f_{OSCA} = 48 \times f_{SAMPLE}$.
Note 9: The filter on RXD suppresses noise spikes at the input buffers and delays the sampling instant.
Note 10: V_{RESET} and V_{BO} will never overlap.

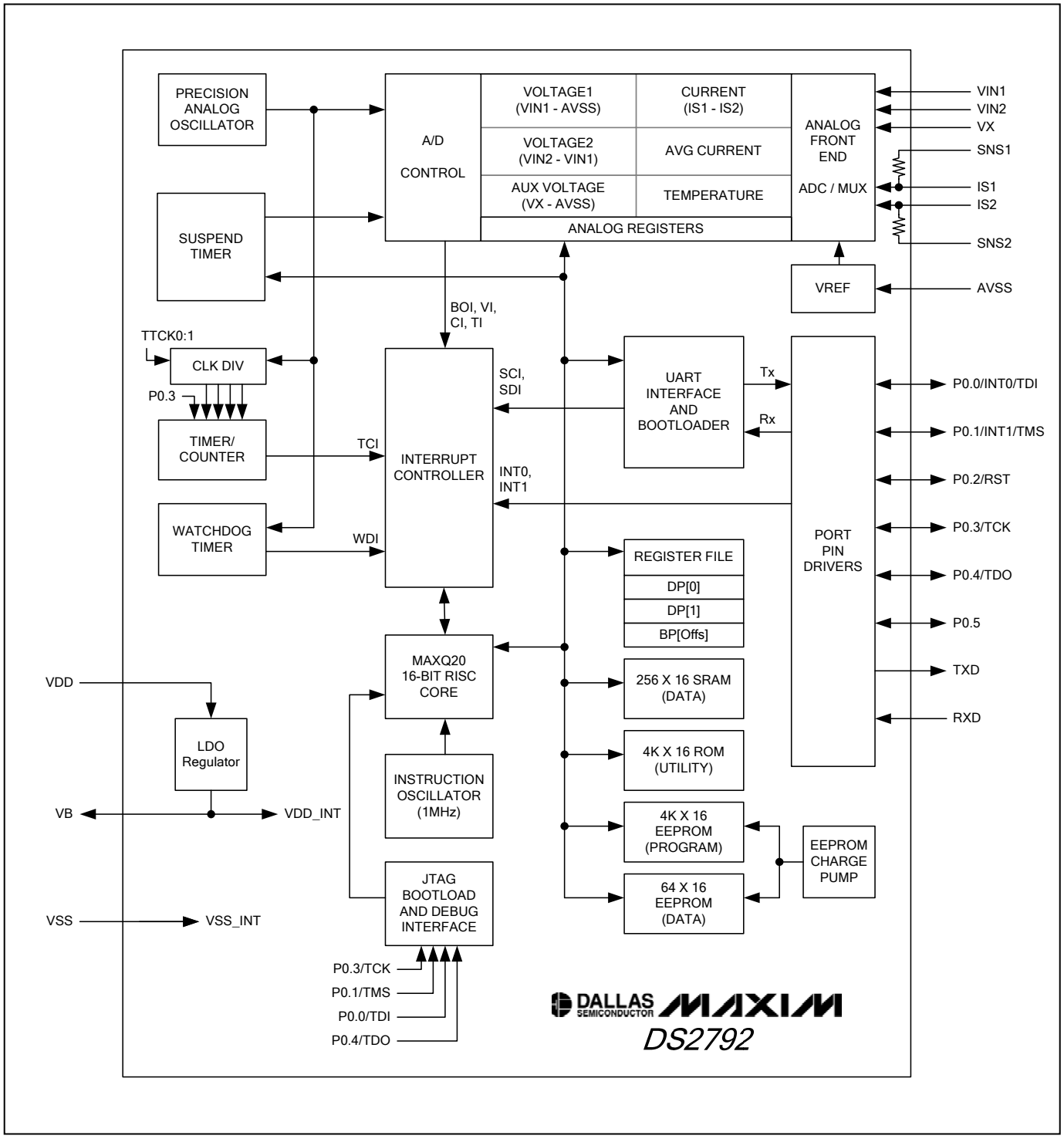
Figure 1. JTAG Timing Diagram



PIN DESCRIPTION

PIN	NAME	FUNCTION
1, 7–10, 21, 25, 26, 28	N.C.	**No Connection**
2	VIN2	**Battery Voltage Sense Input 2**. Voltage measurement on VIN2 is relative to VIN1.
3	VIN1	**Battery Voltage Sense Input 1**. Voltage measurement on VIN1 is relative to AV_{SS}.
4	Vx	**Auxiliary ADC Input**. Voltage measured ratiometrically with respect to V_B pin voltage or absolutely with respect to internal reference.
5	TXD	**Serial Interface Data Transmit**. Maximum of 19200bps.
6	RXD	**Serial Interface Data Recieve**. Maximum of 19200bps.
11	P0.0	**Programmable I/O Pin**. Alternate functions: external interrupt input INT0, [JTAG TDI].
12	P0.1	**Programmable I/O Pin**. Alternate functions: external interrupt input INT1, [JTAG TMS].
13	SNS2	**Current-Sense Input**. SNS2 attaches to the pack end of current-sense resistor.
14	IS2	**Current Filter Input 2**
15	IS1	**Current Filter Input 1**
16	SNS1	**Current-Sense Input**. SNS1 attaches to the battery end of current-sense resistor and V_{SS}.
17	AV_{SS}	**Analog Supply Return Node**. AV_{SS} attaches to negative battery terminal.
18	V_{SS}	**Digital Supply Return Node**. V_{SS} attaches to negative battery terminal.
19	P0.2	**Programmable I/O Pin**. Alternate functions: reset input pin $\overline{RST}$.
20	P0.3	**Programmable I/O Pin**. Alternate functions: timer/counter input pin TCK, [JTAG TCK].
22	P0.4	**Programmable I/O Pin**. Alternate function: [JTAG TDO]
23	P0.5	**Programmable I/O Pin**
24	V_B	**Bias Supply Output**. Internally regulated to 3.3V. Bypass V_B to V_{SS} with 0.1µF.
27	V_{DD}	**Input Supply**. +2.5V to +10.0V input range. Bypass V_{DD} to V_{SS} with 0.1µF.
—	PAD	**Exposed PAD.** Not electrically connected to IC. Connect to V_{SS} or leave floating.

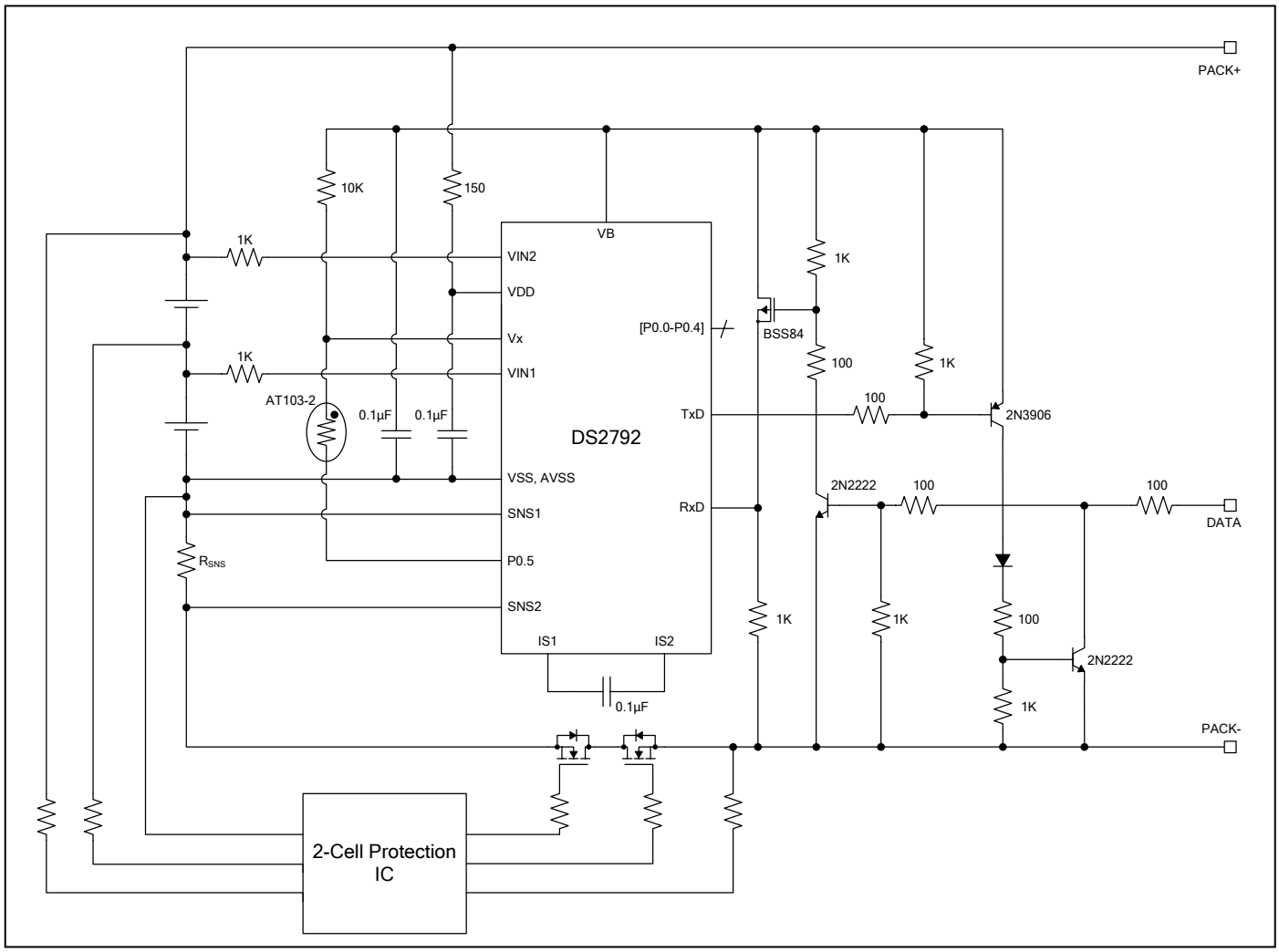
FUNCTIONAL DIAGRAM



TYPICAL OPERATING CIRCUIT

In Figure 2, the DS2792 is connected to the battery side of the pack protector. V_{DD} is isolated with a 150Ω resistor. Both V_{DD} and the internally regulated voltage V_B have 0.1µF bypass capacitors. The VIN1 and VIN2 pins sample the voltage of each cell. Current flow through the cell pack is monitored by measuring the voltage drop across the sense resistor R_{SNS}. Cell temperature is measured ratiometrically through the general-purpose voltage input Vx. The GPIO pin P0.5 gates the thermistor circuit to limit current flow between measurements. The UART signals RXD and TXD are combined into a single 3.3V bidirectional I/O line. The I/O circuit causes no extra current drain when idle.

Figure 2. Example Pack Circuit with Regulated I/O



DETAILED DESCRIPTION

The following is an introduction to the primary features of the DS2792 programmable 2-cell Li-Ion fuel gauge. More detailed descriptions of the device features can be found in the errata sheets and user's guides described later in the *Additional Documentation* section.

DS2792 Overview

The DS2792 incorporates the 16-bit MAXQ20 microcontroller core with 16 accumulators and 16-level hardware stack. Four memory blocks provide application code space, utility code space, RAM memory, and EEPROM memory. Specialized peripherals are integrated to perform battery monitoring, coulomb counting, and UART communication functions. The MAXQ20 core along with the specialized peripherals provide a flexible solution for fuel gauging of Li-Ion or NiMH battery packs. Flexibility is further enhanced as the solution allows for upgrading of

the program and data EEPROM contents over the UART interface. Updates to the program and data EEPROM are protected against unauthorized writes by a 256-bit user password. A read protection bit is provided to prevent reading either EEPROM.

MAXQ20 Core Architecture

The DS2792 employs a MAXQ20 low-cost, high-performance, CMOS, fully static, 16-bit RISC microcontroller with EEPROM memory. 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, such as ADDC, OR, JUMP, etc. The op codes are implemented as MOVE instructions between 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 to 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 four bits contain the module specifier and the upper four bits contain the register index in that module.

Bits 8 to 14 represent the destination for the transfer. This value always represents a destination register, with the lower four bits containing the module specifier and the upper three bits containing the register subindex within that module. Any time that 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

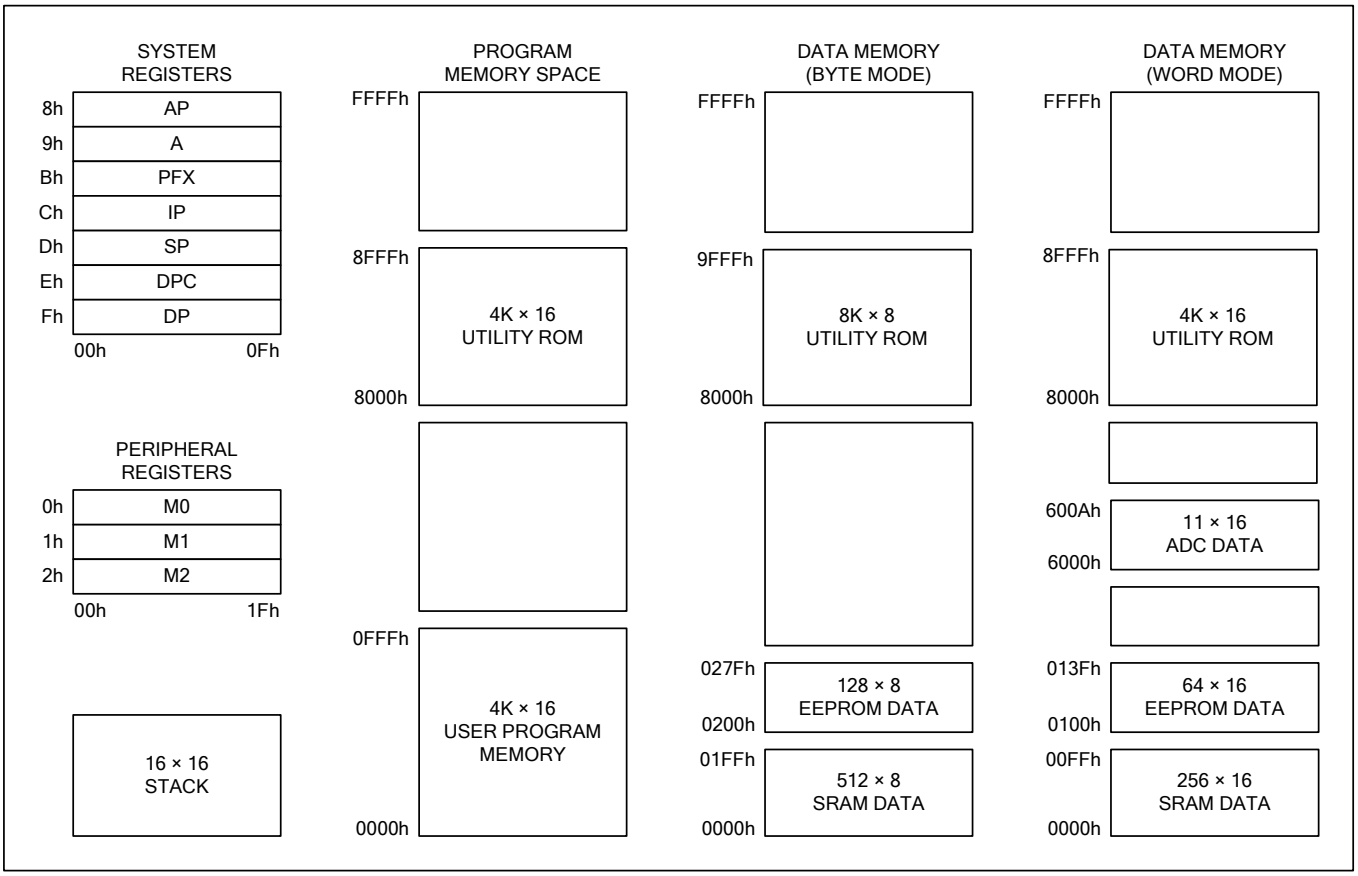
The DS2792 incorporates several memory areas:

- 4k Words of utility ROM contain a debugger, program loader, and SHA-1 routines
- 4k Words of EEPROM memory for application program storage
- 256 Words of SRAM for storage of temporary variables
- 64 Words of EEPROM memory for data storage
- 10 Words of ADC conversion data information
- 16-level stack memory for storage of program return addresses and general-purpose use

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 utilized placing ROM, application code, and data memory into a single contiguous memory map. The pseudo-Von Neuman 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 DS2792's memory map when executing from program memory space. Refer to the *MAXQ Family User's Guide: DS2792 Supplement* for memory map information when executing from data or ROM space.

The incorporation of EEPROM memory allows field upgrade of the firmware. EEPROM memory can be password protected with a 16-word key, denying access to program memory by unauthorized individuals. ROM memory is also available for high-volume, low-cost applications. Contact Dallas Semiconductor for more information on the availability of ROM-based devices.

Figure 3. DS2792 Memory Map



Stack Memory

A 16-bit, 16-level internal stack provides storage for program return addresses and general-purpose use. The stack is used automatically 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, then store a value at the location pointed to by SP. The RET, RETI, POP, and POPI operations retrieve the value pointed to by SP, and then decrement SP.

Utility ROM

The utility ROM is a 4k Word 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:

- In-system programming (bootstrap loader) over JTAG or serial interfaces
- In-circuit debug routines
- Internal self-test routines
- Callable routines for in-application EEPROM programming and SHA-1 calculations

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: DS2792 Supplement*.

Some applications require protection against unauthorized viewing of program code memory. For these applications, access to in-system programming, in-application programming, or in-circuit debugging functions is prohibited until a password has been supplied. The password is defined as the 16 words of physical program memory at addresses x0010h to x001Fh. Upon startup, code in the ROM examines the password, if a password is defined (password is other than all zeros or all ones), and the PWL bit remains set, which prohibits access to commands to read memory contents over the JTAG and serial interfaces.

A single password lock (PWL) bit is implemented in the SC register. When the PWL is set to one (power-on reset default), the password is required to access the utility ROM, including in-circuit debug and in-system programming routines that allow reading or writing of internal memory. When PWL is cleared to zero, these utilities are fully accessible without password. The password is automatically set to all ones following a mass erase.

PROGRAMMING

The EEPROM memory of the microcontroller can be programmed by two different methods: in-system programming and in-application programming. Both methods afford great flexibility in system design as well as reduce the life-cycle cost of the embedded system. These features can be password protected to prevent unauthorized access to code memory.

In-System Programming

An internal bootstrap loader allows the device to be programmed over the JTAG or serial interfaces. As a result, system software can be upgraded in-system, eliminating the need for a costly hardware retrofit when software updates are required. Remote software uploads are possible that enable physically inaccessible applications to be frequently updated. The JTAG interface hardware can be a JTAG connection to another microcontroller, or a connection to a PC serial port using a serial to JTAG converter such as the MAXQJTAG-001 (3.3V reference voltage required), available from Maxim Integrated Products. The UART interface hardware can be a connection to another microcontroller, or a connection to a PC USB port using a USB to UART converter such as the DS9123O, available from Dallas Semiconductor. A commercial gang programmer can also be used for programming.

Activating the JTAG interface and loading the test access port (TAP) with the system programming instruction invokes the bootstrap loader for use over the JTAG interface. Setting the SPE bit to 1 during reset through the JTAG interface executes the bootstrap-loader-mode program that resides in the utility ROM. When programming is complete and the bootstrap loader exited, the SPE bit will clear and the IC will reset, allowing execution of the application software.

Performing a program request over the serial interface also invokes the bootstrap loader. The user must successfully complete a password match (if PWL = 1). The bootstrap loader functions are then fully supported over the serial interface. When programming is complete, the exit loader function is used to reset the DS2792 and begin execution of the application software.

The following bootstrap loader functions are supported:

- Information commands
- Load EEPROM code and data
- Dump EEPROM code and data
- CRC EEPROM code and data
- Verify EEPROM code and data
- Erase EEPROM code and data

In-Application Programming

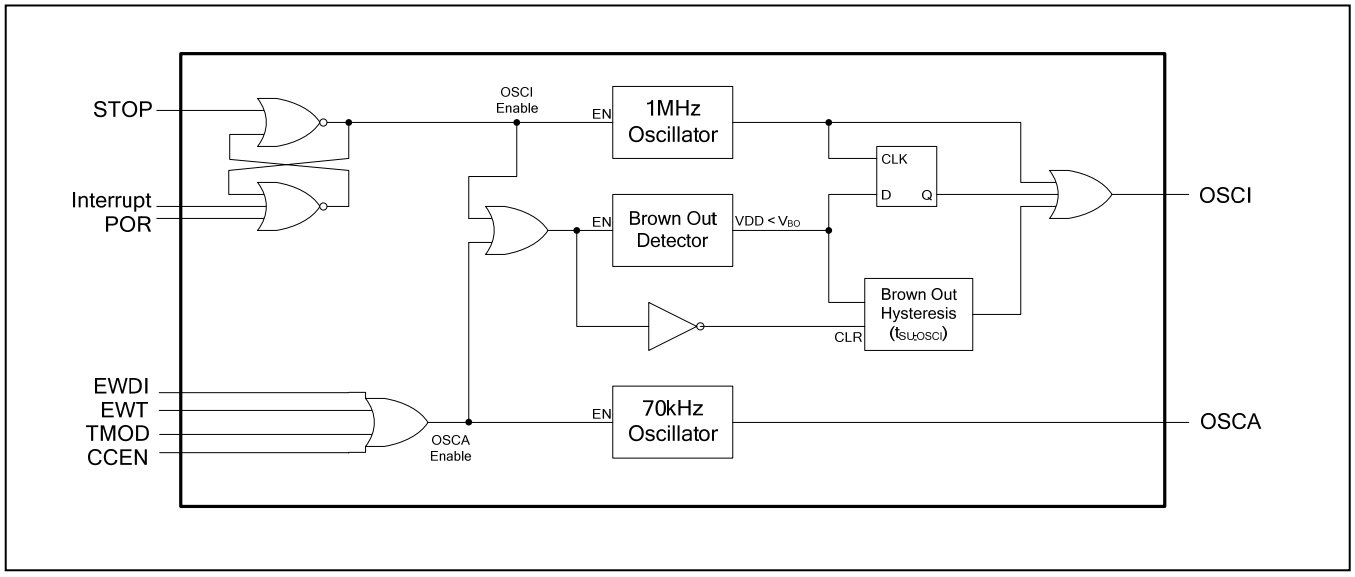
The in-application programming feature allows the microcontroller to modify its own EEPROM program memory. This allows on-the-fly software updates in mission-critical applications that cannot afford downtime. Alternatively, it allows the application to develop custom loader software that can operate under the control of the application software. The utility ROM contains firmware-accessible EEPROM programming functions that erase and program EEPROM memory. These functions are described in detail in the *MAXQ Family User's Guide: DS2792 Supplement*.

SYSTEM TIMING

The DS2792 generates its 1MHz instruction clock (OSCI) internally. This quick starting oscillator is used for instruction fetch and execution by the MAXQ20 core. The analog oscillator (OSCA) is a bandgap-based RC oscillator that is trimmed to better than 2% accuracy ($f_{ERR:OSCA}$). The analog clock runs independent of OSCI and serves as the clock source for the ADC, watchdog timer, and interval timer. OSCA is enabled by the watchdog timer signals EWDI or EWT, by the timer/counter (TMOD), or by the coulomb counter (CCEN).

OSCI is enabled through either a system interrupt or system POR and disabled through a system stop. A voltage brownout-detection circuit disables OSCI if V_{DD} falls below V_{BO}. Once V_{DD} raises above V_{BO}, a hysteresis circuit waits $t_{SU:OSCI}$ before re-enabling OSCI. To improve overall system timing and meet UART timing requirements, OSCI is slaved to OSCA when OSCA is active.

Figure 4. System Clocks



SYSTEM RESET

Several reset sources are provided for microcontroller control. Although code execution is halted in the reset state, OSCI continues to run.

Power-On Reset: An internal power-on reset circuit enhances system reliability. This circuit forces the device to perform a power-on reset whenever a rising voltage on V_{DD} climbs above V_{POR}. At this point 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.

Watchdog Timer Reset: Software can determine if a reset is caused by a watchdog timeout by checking the watchdog timer reset flag (WTRF) in the WDCN register. Execution resumes at location 8000h following a watchdog timer reset.

External System Reset: Asserting the external $\overline{RST}$ (port P0.2) 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 $\overline{RST}$ pin is released.

MAXQ20 CORE POWER MANAGEMENT

The DS2792 is designed for low-power battery-monitoring applications. The peripherals have been designed with the ability to wake the processor from SLEEP or ANALOG mode any time software intervention is needed. Power management is optimized in the applications by performing any necessary processing as quickly as possible, and re-entering the low power SLEEP or ANALOG mode. Processing resumes from SLEEP or ANALOG mode via any of the following sources (when enabled):

- An external interrupt is triggered.
- An external reset signal is applied to the $\overline{RST}$ pin.
- A watchdog timer interrupt occurs.
- An internal interrupt event occurs.

Note: No division of the internal system clock is supported, subsequently the PMME and CD[1:0] bits described in the *MAXQ Family User's Guide* are not implemented in the DS2792.

WATCHDOG TIMER

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 undesireable code execution prevents a reset of the watchdog timer, the timer reaches its maximum count and resets the processor.

The watchdog timer in the DS2792 differs in two respects from the one described in the *MAXQ Family User's Guide*: 1) the clock used by the timer is the 70kHz OSCA clock that runs independently of the 1MHz OSCI (or system) clock, and 2) the watchdog interrupt is an asynchronous interrupt that can bring the processor out of stop mode.

The watchdog timer is controlled through bits in the WDCN register. Its timeout period can be set to one of the four programmable intervals ranging from 2^{12} to 2^{21} OSCA clock periods (59ms up to 30s). The watchdog interrupt occurs at the end of this timeout period, which is 512 OSCA clock periods, or 7.3ms, before the reset.

DS2792 POWER MODES

When power is first applied to the DS2792, a power-on-reset (POR) circuit transitions the IC to brownout state where cell voltage is monitored. If V_{DD} voltage is above the brownout threshold V_{BO}, the DS2792 enters CPU mode and begins code execution. Firmware determines if the IC switches to ANALOG mode or low-power SLEEP mode when a STOP halts CPU operation.

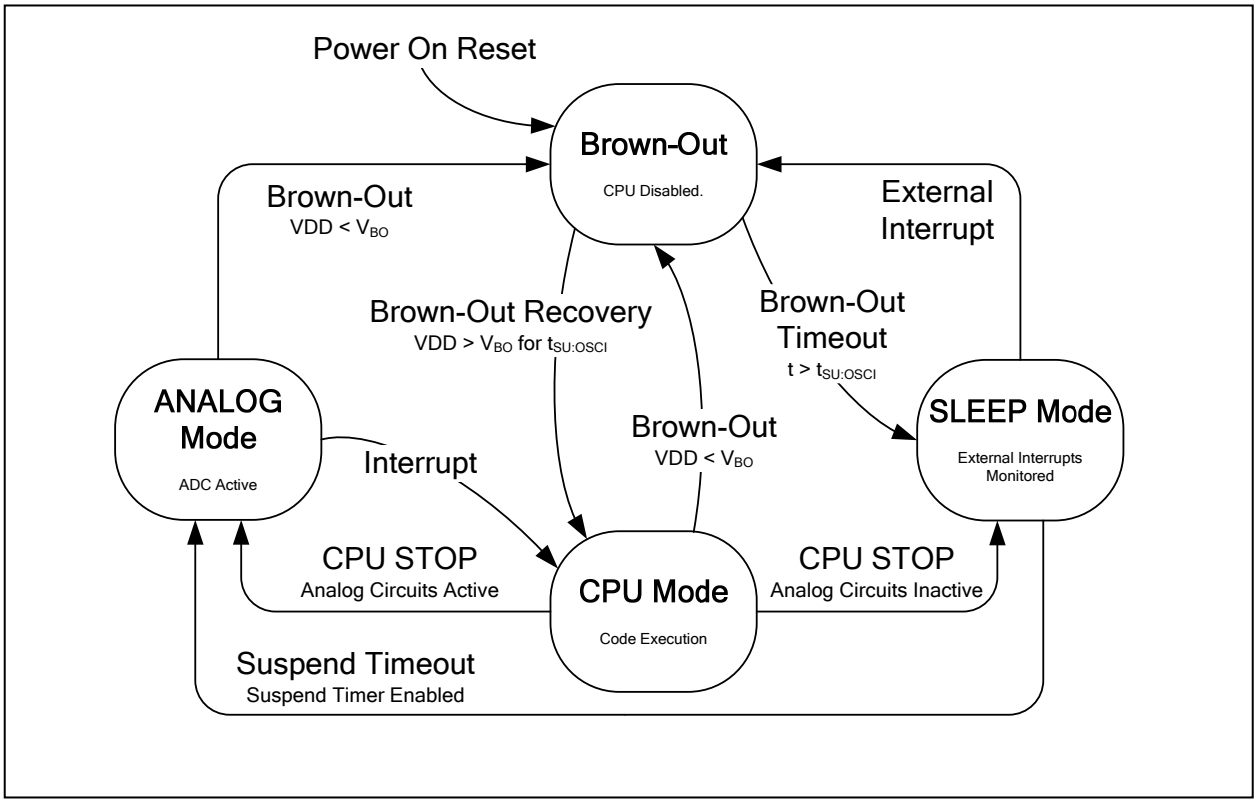
The DS2792 enters SLEEP mode after a CPU STOP if the ADC and all internal timers are disabled and the suspend timer is off. In SLEEP mode, all IC operation becomes inactive except for external activity interrupts. Brownout detection does not occur in SLEEP mode. Any interrupt generated by UART communication or external input on ports P0.0 or P0.1 will transition the DS2792 from SLEEP to brownout to verify cell voltage before returning to CPU mode. If the suspend timer is enabled, the IC will transition from SLEEP to ANALOG mode if a suspend timeout occurs.

The DS2792 enters ANALOG mode after a CPU STOP if any one of the following is active: the ADC, the interval timer, or the watchdog timer. An external interrupt or an interrupt from any active internal circuit causes the DS2792 to transition back to CPU mode to service the condition.

If the DS2792 is in ANALOG or CPU mode, and V_{DD} falls below V_{BO}, a brownout condition occurs and the DS2792 enters the brownout state. In brownout state, the processor is halted without changing the instruction pointer. If V_{DD} voltage rises above V_{BO} within a time of $t_{SU:OSCI}$, the DS2792 returns to CPU mode and generates a brown-out interrupt (if enabled). Otherwise, if V_{DD} remains below V_{BO} for $t_{SU:OSCI}$, the DS2792 again enters SLEEP mode and waits to receive an interrupt.

Note that the supply rise time at initial power up of the IC must be faster than $t_{SU:OSCI}$ to prevent the DS2792 from entering SLEEP mode prior to software enabling external interrupts. If this situation occurs, the DS2792 remains in SLEEP until power cycled.

Figure 5. DS2792 State Diagram



SUSPEND TIMER

The suspend timer allows the DS2792 to periodically monitor the cell while spending the majority of its time in a low-power state. Suspend operation is managed through the coulomb counter control register (CCCON). A timout value between 0.5s and 2.0s can be selected through the suspend timeout bits (CCSUS). When the suspend timer is enabled and a timeout occurs, the DS2792 will transition from SLEEP mode to ANALOG mode, and eventually to CPU mode. Upon transition to ANALOG mode, the coulomb counter enable bit (CCEN) is automatically set and the ADC block becomes active. The DS2792 may then perform all needed cell-monitoring operations before clearing the CCEN bit and returned to SLEEP mode. The process repeats after each timeout.

Figure 6. CCCON Register Format

FIELD	BIT	FORMAT	ALLOWABLE VALUES
Reserved	7:3	Read Only	Undefined
CCSUS	2:1	R/W	Suspend Timeout 0 0 = Suspend Timeout Disabled 0 1 = 0.5s timeout 1 0 = 1.0s timeout 1 1 = 2.0s timeout
CCEN	0	R/W	Coulomb Counter Enable 0 = ADC block disabled 1 = ADC block enabled

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 and peripheral registers. 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 that may be included by different products based on the MAXQ20 architecture. This functionality is broken up into discrete modules so that only the features required for a given product need to be included. Tables 1 through 6 detail the DS2792 register set.

Table 1. System Register Map

REGISTER INDEX	MODULE NAME (BASE SPECIFIER)						
	AP (8h)	**A (9h)**	**PFX (Bh)**	**IP (Ch)**	**SP (Dh)**	**DPC (Eh)**	**DP (Fh)**
00h	AP	**A[0]**	**PFX**	**IP**	—	—	—
01h	APC	**A[1]**	—	—	**SP**	—	—
02h	—	**A[2]**	—	—	**IV**	—	—
03h	—	**A[3]**	—	—	—	Offs	**DP0**
04h	PSF	**A[4]**	—	—	—	**DPC**	—
05h	IC	**A[5]**	—	—	—	**GR**	—
06h	IMR	**A[6]**	—	—	**LC0**	GRL	—
07h	—	**A[7]**	—	—	**LC1**	**BP**	**DP1**
08h	SC	**A[8]**	—	—	—	***GRS***	—
09h	—	**A[9]**	—	—	—	GRH	—
0Ah	—	**A[10]**	—	—	—	***GRXL***	—
0Bh	*IIR*	**A[11]**	—	—	—	***FP***	—
0Ch	—	**A[12]**	—	—	—	—	—
0Dh	—	**A[13]**	—	—	—	—	—
0Eh	CKCN	**A[14]**	—	—	—	—	—
0Fh	WDCN	**A[15]**	—	—	—	—	—

Note: *Names that appear in italics indicate that all bits of a register are read-only. Names that appear in bold indicate that a register is 16 bits wide. Registers in module AP are bit addressable.*

Table 2. System Register Bit Functions

REGISTER	REGISTER BIT NUMBER															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AP									—	—	—	—	AP (4 bits)			
APC									CLR	IDS	—	—	—	MOD2	MOD1	MOD0
PSF									Z	S	—	GPF1	GPF0	OV	C	E
IC									—	—	CGDS	—	—	—	INS	IGE
IMR									IMS	—	—	—	—	—	IM1	IM0
SC									TAP	—	—	—	—	ROD	PWL	—
IIR									IIS	—	—	—	—	—	II1	II0
CKCN									—	—	—	—	—	—	—	—
WDCN									POR	EWDI	—	—	WDIF	WTRF	EWT	RWT
A[n] (0..15)	A[n] (16 bits)															
PFX	PFX (16 bits)															
IP	IP (16 bits)															
SP	—	—	—	—	—	—	—	—	—	—	—	—	SP (4 bits)			
IV	IV (16 bits)															
LC[0]	LC[0] (16 bits)															
LC[1]	LC[1] (16 bits)															
Offs									Offs (8 bits)							
DPC	—	—	—	—	—	—	—	—	—	—	—	WBS2	WBS1	WBS0	SDPS1	SDPS0
GR	GR.15	GR.14	GR.13	GR.12	GR.11	GR.10	GR.9	GR.8	GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0
GRL									GRL.7	GRL.6	GRL.5	GRL.4	GRL.3	GRL.2	GRL.1	GRL.0
BP	BP (16 bits)															
GRS	GRS.15	GRS.14	GRS.13	GRS.12	GRS.11	GRS.10	GRS.9	GRS.8	GRS.7	GRS.6	GRS.5	GRS.4	GRS.3	GRS.2	GRS.1	GRS.0
GRH									GRH.7	GRH.6	GRH.5	GRH.4	GRH.3	GRH.2	GRH.1	GRH.0
GRXL	GRXL.15	GRXL.14	GRXL.13	GRXL.12	GRXL.11	GRXL.10	GRXL.9	GRXL.8	GRXL.7	GRXL.6	GRXL.5	GRXL.4	GRXL.3	GRXL.2	GRXL.1	GRXL.0
FP	FP (16 bits)															
DP[0]	DP[0] (16 bits)															
DP[1]	DP[1] (16 bits)															

Table 3. System Register Bit Reset Values

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AP									0	0	0	0	0	0	0	0
APC									0	0	0	0	0	0	0	0
PSF									1	0	0	0	0	0	0	0
IC									0	0	0	0	0	0	0	0
IMR									0	0	0	0	0	0	0	0
SC									0	0	0	0	0	0	s	0
IIR									0	0	0	0	0	0	0	0
CKCN									0	s	s	0	0	0	0	0
WDCN									s	s	0	0	0	0	0	0
A[n] (0..15)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PFX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IP	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SP	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
IV	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LC[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LC[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Offs									0	0	0	0	0	0	0	0
DPC	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
GR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRL									0	0	0	0	0	0	0	0
BP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRH									0	0	0	0	0	0	0	0
GRXL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DP0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DP1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: *s indicates bit reflects pin state.*

Table 4. Peripheral Register Map

REGISTER INDEX	MODULE			REGISTER INDEX	MODULE		
	M0 (0h)	M1 (1h)	M2 (2h)		M0 (0h)	M1 (1h)	M2 (2h)
00h	PO	**SINT**	—	10h	—	—	—
01h	PPU	**SMASK**	—	11h	—	—	—
02h	PAF	—	—	12h	—	—	—
03h	**EIC**	**SMD**	—	13h	—	—	—
04h	**EINT**	SBUF	—	14h	—	—	—
05h	**CCCON**	SCON	—	15h	—	—	—
06h	**TC**	**PR**	—	16h	—	—	—
07h	TCC	—	—	17h	—	—	—
08h	*PI*	—	—	18h	**ICDT0**	—	—
09h	—	—	—	19h	**ICDT1**	—	—
0Ah	—	SADEN	—	1Ah	*ICDC*	—	—
0Bh	—	SADDR	—	1Bh	ICDF	—	—
0Ch	—	**BRE**	—	1Ch	ICDB	—	—
0Dh	—	—	ECNTL	1Dh	***ICDA***	—	—
0Eh	—	—	**EADDR**	1Eh	***ICDD***	—	—
0Fh	—	—	**EDATA**	1Fh	—	—	—

Note: *Names that appear in italics indicate that all bits of a register are read-only. Names that appear in bold indicate that a register is 16 bits. All locations are bit addressable.*

Table 5. Peripheral Register Bit Functions

REGISTER	REGISTER BIT NUMBER															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO									—	—	PO.5	PO.4	PO.3	PO.2	PO.1	PO.0
PPU									TxD_PU	RxD_PU	PPU.5	PPU.4	PPU.3	PPU.2	PPU.1	PPU.0
PAF									—	RSTD	PAF.5	PAF.4	PAF.3	PAF.2	PAF.1	PAF.0
EIC	MBOI	MSCI	MSDI	—	—	—	MVI	MCI	MTI	MTCI	PIP.1	PIP.0	PIT.1	PIT.0	PIE.1	PIE.0
EINT	BOI	SCI	SDI	—	—	—	VI	CI	TI	TCI	—	—	—	$\overline{RST}$	INT.1	INT.0
CCCON									—	—	—	—	—	CCSUS.1	CCSUS.0	CCEN
TC	THI.7	THI.6	THI.5	THI.4	THI.3	THI.2	THI.1	THI.0	TLOW.7	TLOW.6	TLOW.5	TLOW.4	TLOW.3	TLOW.2	TLOW.1	TLOW.0
TTC									—	—	—	—	—	TTCK.1	TTCK.0	TMOD
PI									—	*RxDI*	*PI.5*	*PI.4*	*PI.3*	*PI.2*	*PI.1*	*PI.0*
ICDT0	ICDT0.15	ICDT0.14	ICDT0.13	ICDT0.12	ICDT0.11	ICDT0.10	ICDT0.9	ICDT0.8	ICDT0.7	ICDT0.6	ICDT0.5	ICDT0.4	ICDT0.3	ICDT0.2	ICDT0.1	ICDT0.0
ICDT1	ICDT1.15	ICDT1.14	ICDT1.13	ICDT1.12	ICDT1.11	ICDT1.10	ICDT1.9	ICDT1.8	ICDT1.7	ICDT1.6	ICDT1.5	ICDT1.4	ICDT1.3	ICDT1.2	ICDT1.1	ICDT1.0
ICDC									*DME*	—	*REGE*	—	*CMD.3*	*CMD.2*	*CMD.1*	*CMD.0*
ICDF									—	—	—	—	PSS.1	PSS.0	SPE	TXC
ICDB									ICDB.7	ICDB.6	ICDB.5	ICDB.4	ICDB.3	ICDB.2	ICDB.1	ICDB.0
ICDA	*ICDA.15*	*ICDA.14*	*ICDA.13*	*ICDA.12*	*ICDA.11*	*ICDA.10*	*ICDA.9*	*ICDA.8*	*ICDA.7*	*ICDA.6*	*ICDA.5*	*ICDA.4*	*ICDA.3*	*ICDA.2*	*ICDA.1*	*ICDA.0*
ICDD	*ICDD.15*	*ICDD.14*	*ICDD.13*	*ICDD.12*	*ICDD.11*	*ICDD.10*	*ICDD.9*	*ICDD.8*	*ICDD.7*	*ICDD.6*	*ICDD.5*	*ICDD.4*	*ICDD.3*	*ICDD.2*	*ICDD.1*	*ICDD.0*
SINT									—	—	—	—	—	—	TINT	RINT
SMASK									—	—	—	—	—	—	MTI	MRI
SMD									—	—	TXDi	RXDi	—	—	SMOD	FEDE
SBUF									SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SCON									FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI
PR	PR.15	PR.14	PR.13	PR.12	PR.11	PR.10	PR.9	PR.8	PR.7	PR.6	PR.5	PR.4	PR.3	PR.2	PR.1	PR.0
SADEN									SAME.7	SAME.6	SAME.5	SAME.4	SAME.3	SAME.2	SAME.1	SAME.0
SADDR									SAD.7	SAD.6	SAD.5	SAD.4	SAD.3	SAD.2	SAD.1	SAD.0
BRE	BRE.15	BRE.14	BRE.13	BRE.12	BRE.11	BRE.10	BRE.9	BRE.8	BRE.7	BRE.6	BRE.5	BRE.4	BRE.3	BRE.2	BRE.1	BRE.0

Note: *Names that appear in italics indicate a read-only register bit.*

Table 6. Peripheral Register Reset Values

REGISTER	REGISTER BIT NUMBER															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO									0	0	1	1	1	1	1	1
PPU									0	0	0	0	0	1	0	0
PAF									0	0	0	0	0	1	0	0
EIC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EINT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CCCON									0	0	0	0	0	0	0	0
TC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TTC									0	0	0	0	0	0	0	0
PI									s	s	s	s	s	s	s	s
ICDT0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ICDT1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ICDC									0	0	0	0	0	0	0	0
ICDF									0	0	0	0	0	0	0	0
ICDB									0	0	0	0	0	0	0	0
ICDA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ICDD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SINT									0	0	0	0	0	0	0	0
SMASK									0	0	0	0	0	0	0	0
SMD									0	0	0	0	0	0	0	0
SBUF									0	0	0	0	0	0	0	0
SCON									0	0	0	0	0	0	0	0
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SADEN									0	0	0	0	0	0	0	0
SADDR									0	0	0	0	0	0	0	0
BRE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: *s indicates bit reflects pin state.*

ON-CHIP REGULATOR

The DS2792 provides a regulated 3.3V output on the V_B pin capable of supplying up to 2mA of current to external circuitry. The regulated supply can be used to level shift the I/O lines and/or provide a reference voltage for the Vx pin to measure a pack ID resistor or thermistor. The regulator output is always active regardless of the DS2792's mode of operation.

SYSTEM INTERRUPTS

Multiple interrupt sources are available for quick response to internal and external events. The MAXQ20 architecture uses a single interrupt vector (IV), 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. The specified module 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 described in the *MAXQ Users Guide*, the SINT register described in the *UART Interrupts* section, and the EINT register as shown in Figure 7.

EINT Register

The EINT register contains interrupts generated by the ADC, the timer counter, the general-purpose port pins, and the UART RXD pin. Their masks and their configuration bits, along with the $\overline{RST}$ pin status and control, are present in the EIC and PAF registers of module 0.

Figure 7. EINT Register Interrupt Sources

GENERATOR	INTERRUPT	MASK	DESCRIPTION	FREQUENCY
Ports and Pins	INT0	PAF.0/PIE.0	The **interrupt from pin P0.0** is configurable via the PAF.0, PIT.0, and PIP.0 bits.	Dependent on external conditions.
	INT1	PAF.1/PIE.1	The **interrupt from pin P0.1** is configurable via the PAF.1, PIT.1, and PIP.1 bits.	Dependent on external conditions.
	SCI	MSCI	The **serial connect interrupt** is generated when RXD becomes high.	Rising edge on RXD.
	SDI	MSDI	The **serial disconnect interrupt** is generated when RXD is low for at least 220ms.	Once every 220ms if RXD is held low. The first interrupt may take up to 440ms from the time RXD goes low. Interrupt will not trigger if the ADC is off.
Brownout Detector	BOI	MBOI	The **brownout interrupt** indicates that V_{DD} was below V_{BO} in the past. It will not terminate the microcontroller's stop mode. It will interrupt the microcontroller, if MBOI is 1, after a charger brings V_{DD} above V_{BO} and causes the microcontroller to run.	Every time after exiting brownout.
A/D Converter	VI	MVI	The **voltage interrupt** indicates the voltage registers in the data peripheral memory block have fresh voltage averages.	Once every 13.8ms. Never if the ADC is off.
	CI	MCI	The **current interrupt** indicates that the current register (6004h) in the data peripheral memory block has a fresh reading and that the ACR has also been updated.	Once every 88ms. Never if the ADC is off.
	TI	MTI	The **temperature interrupt** indicates that the temperature register in the data peripheral memory block has a fresh average.	Once every 220ms. Never if the ADC is off.
Timer/ Counter	TCI	MTCI	The **timer/counter interrupt** indicates that the timer/counter has been reloaded after reaching its end-count.	Dependent on TMOD and TTCK[1:0].

I/O PORTS

The DS2792 includes a simple input/output (I/O) data port. From a software perspective, the port appears as a group of special-function registers within module M0. The simple I/O port defined for this product is described below:

- CMOS input buffers
- Four open-drain output drivers with selectable tri-state or weak pullups
- Two selectable open-drain or push-pull output drivers with selectable tri-state
- Support alternate functions and TAP controller interface signals
- Two pins have interrupt capability

The port is accessed through five peripheral registers (PO, PI, PAF, PPU, and EIC) addressed either by byte or by individual bit locations. The I/O port is designed to provide programming flexibility for the application. All individual I/O pins are independently configured; and can be defined as an input, output, or alternate function. Table 7 summarizes the functionality of the I/O pins.

Table 7. I/O Port Pins

FUNCTIONS			CHARACTERISTICS			
Primary	Alternate	TAP*	Bidirectional	Weak Passive Pulldown	Weak Active Pullup	Strong Active Pullup
P0.0	INT0	TDI*	Configurable, [In]	—	Configurable, [Off]	—
P0.1	INT1	TMS*	Configurable, [In]	—	Configurable, [Off]	—
P0.2	$\overline{RST}$*	—	Configurable, [In]	—	Configurable, [Off]	—
P0.3	—	TCK*	Configurable, [In]	—	Configurable, [Off]	—
P0.4	—	TDO*	Configurable, [In]	—	—	Configurable, [Off]
P0.5	—	—	Configurable, [In]	—	—	Configurable, [Off]
TXD	—	—	Output	Configurable, [On]	Configurable, [Off]	—
RXD	—	—	Input	Configurable, [On]	Configurable, [Off]	—

Note: *Reset values are denoted with an * and by [ ].*

PI Register: The PI register is a read-only input of the I/O pins. When the register is read, the logic level of each pin is reported in the corresponding bit locations. Reading a logic-low or high on a pin does not change the output drive on that pin.

PO Register: The PO register controls the output state of the I/O pins. Data written to this register determines the pin output drive. When a bit is written to a “0” (cleared), the n-channel output drive transistor is enabled, and the pullup is disabled. When bit is written to a “1” (set), the n-channel output drive transistor is disabled, and the pullup enabled (if so configured). The PO bits are set asynchronously during power-on reset to disable the N-channel output drive. PO bits are not altered in SLEEP mode, however drive to the n-channel is disabled.

PPU Register: The PPU register contains independent bits that define each pin as high impedance or pulled up when its n-channel output drive transistor is disabled. P0.0 through P0.3 have weak pullups, P0.4 and P0.5 have strong pullups. When the output is disabled and the PPU bit is cleared, the pin is high impedance. When the output is disabled and the PPU bit is set, the pin’s weak or strong pullup is enabled. When the PPU bit is set and the device enters STOP mode, the weak pullup remains enabled.

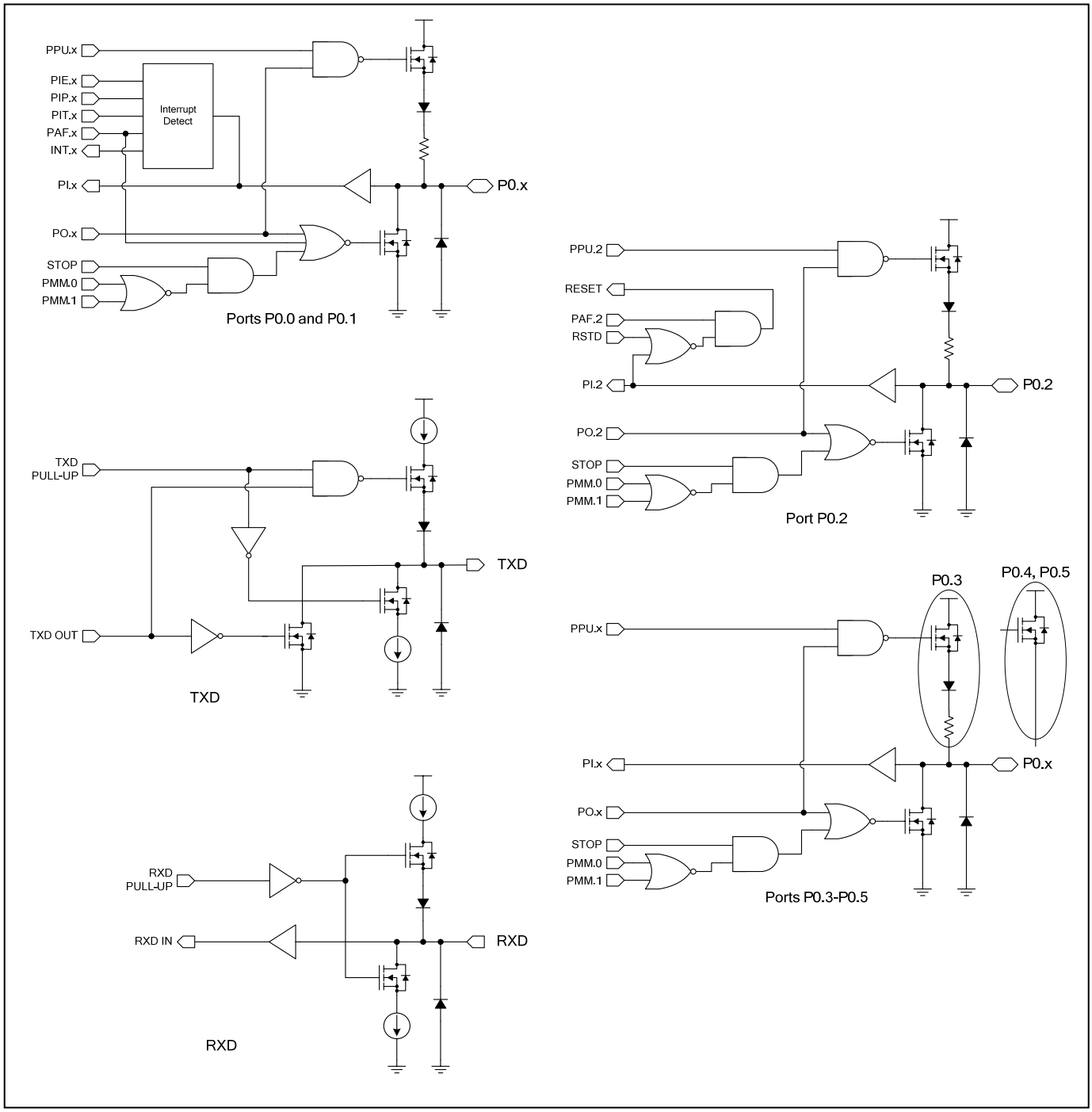
PAF Register: The PAF register enables or disables the alternate functions of P0.0–P0.2. When a pin’s PAF bit is cleared, the pin is controlled by the PI, PO, PPU, and EIC registers. When the PAF bit is set, the pin operates in it’s alternate function mode. The $\overline{RST}$ function of P0.2 can be disabled by writing the RSTD bit to 1.

EIC Register: The lower six bits of the EIC register are the port interrupt control bits. The port interrupt control bits are used to enable and configure detection of external interrupts. Interrupt enable bits, PIE.0 and PIE.1, enable detection of an interrupt on pins P0.0 and P0.1, respectively. Interrupt type bits, PIT.0 and PIT.1, define the type (level or edge) of interrupt on pins P0.0 and P0.1, respectively. Interrupt polarity bits, PIP.0 and PIP.1, determine the interrupt polarity on pins P0.0 and P0.1, respectively.

Table 8. P0 Interrupt Configuration

PIE.x	PIT.x	PIP.x	RESULT
0	X	X	Interrupt Disabled
1	0	0	Interrupt Enabled, Triggered on Logic-Low
1	0	1	Interrupt Enabled, Triggered on Logic-High
1	1	0	Interrupt Enabled, Triggered on Falling Edge
1	1	1	Interrupt Enabled, Triggered on Rising Edge

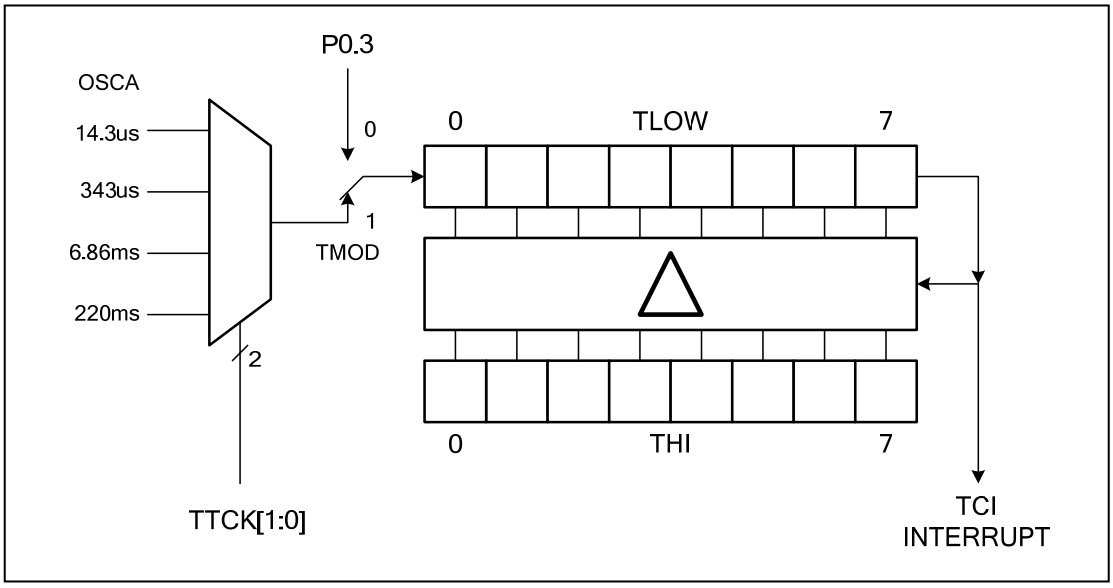
Figure 8. Port Pin Schematics



PROGRAMMABLE TIMER/COUNTER

The timer/counter block operates as a simple 8-bit interval timer or counter. The start value is programmable and is automatically reloaded when a rollover occurs. The TMOD bit in the TCC register selects between the counter and timer modes. In the counter mode, external events on the P0.3 pin are counted. In the timer mode, OSCA clock source cycles are counted. The OSCA clock and brownout detectors continue to run if the CPU is stopped.

Figure 9. Timer/Counter Block Diagram



The timer low byte (TLOW) is used to count input events, while the timer high byte (THI) is used to store the reload value. Firmware must initialize TLOW and THI with the same value for the first count to be the same as succeeding counts. TLOW counts up until FFh is reached, it is then automatically reloaded with the value in THI. THI remains unchanged unless modified by firmware. The clock source is selected with TTCK[1:0] bits. Table 9 describes the possible resolution and range of the timer.

Table 9. Programmable Timer Configuration

TMOD	TTCK[1:0]	CLOCK PERIOD	TIMER RANGE ($t \times 2^8$)
1	0 0	14.3µs	3.66ms
1	0 1	343µs	87.9ms
1	1 0	6.86ms	1.76s
1	1 1	220ms	56.3s
0	N/A	Counter Mode	

SERIAL INTERFACE MODULE

The DS2792 supports a universal asynchronous receiver/transmitter (UART) module for serial communication with framing error detection and automatic address recognition. The UART is initialized through a configuration register (SMD). Communication is managed through a control register (SCON) and a transmit/receive register (SBUF). The SBUF register location provides access to both transmit and receive registers, where a read is directed to the receive buffer and a write is directed to the transmit buffer. SBUF allows the serial interface to receive an incoming word before software has read the previous one provided SBUF is read before the stop bit of the next word is recieved.

If the DS2792 detects a start bit of an incoming transmission while in SLEEP mode, the instruction clock will be enabled to allow reception of data. Note that the instruction clock will not achieve accuracy tolerance required for UART communication unless the analog oscillator is also active. All outgoing transmission must be completed, and the RXD pin must be in the inactive logic state (see RXDi bit), before the DS2792 will transition back to SLEEP mode.

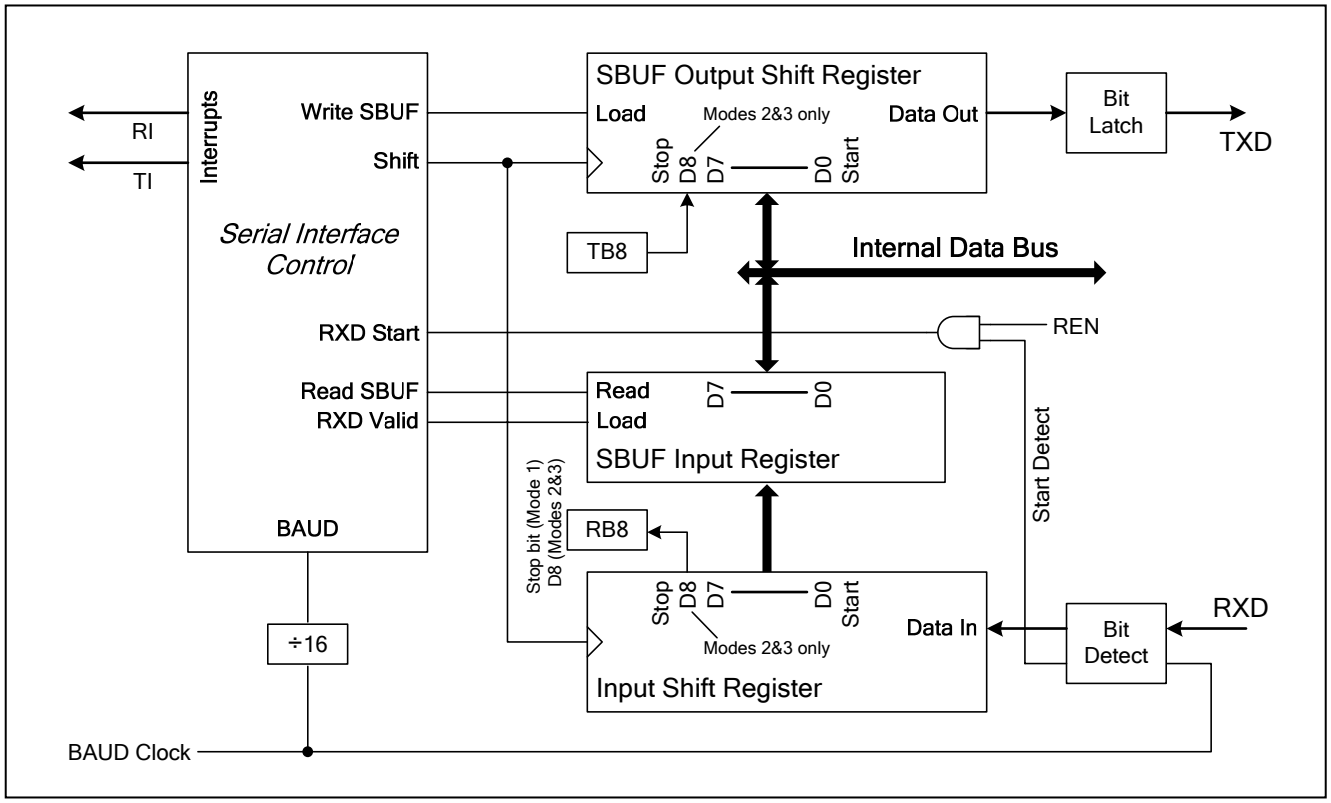
Figure 10. Serial Interface Control Register (SCON)

FIELD	BIT	FORMAT	ALLOWABLE VALUES
FE/SM0	7	R/W	**Serial Interface Mode Bit 0/Framing Error Flag** The operation of this bit depends on the state of the FEDE bit of the SMD register. When FEDE is cleared to 0, this bit functions as SM0 and controls the operating mode. See *Serial Interface Modes*, Table 10. When FEDE is set to 1, this bit functions as a framing error flag and will automatically be set to 1 when the incoming stop bit has been received as a logic 0. When used as FE, this flag must be cleared to 0 by software.
SM1	6	R/W	**Serial Interface Mode Bit 1** See *Serial Interface Modes*, Table 10
SM2	5	R/W	**Serial Interface Mode Bit 2** Mode 1 operation: 0 = Normal operation 1 = RXI flag not set if incoming stop is logic 0 Modes 2 and 3 operation: 0 = Multiprocessor mode disabled 1 = Multiprocessor mode enabled
REN	4	R/W	**Reciever Enable** 0 = Reciever enabled 1 = Reciever disabled
TB8	3	R/W	**Transmit Data Bit 8** When in mode 2 or 3, this bit determines the state of the ninth transmitted data bit. This bit is not used in mode 1.
RB8	2	R/W	**Received Data Bit 8** When in mode 2 or 3, this bit indicates the state of the incoming ninth data bit. In mode 1, this bit indicates the state of the stop bit.
TI	1	R/W	**Transmit Interrupt Flag** Set to 1 at the end of the last data bit transmitted. TI must be cleared by software.
RI	0	R/W	**Recieve Interrupt Flag** Set to 1 at the end of the stop bit sampling window when in mode 1. Set to 1 at the end of the sampling window of the ninth data bit when in mode 2 or 3. RI must be cleared by software.

Figure 11. Serial Interface Configuration Register (SMD)

FIELD	BIT	FORMAT	ALLOWABLE VALUES
Reserved	7:6	Read Only	**Undefined**
TXDi	6	R/W	**TXD Invert Bit** 0 = Normal operation 1 = TXD output signal inverted
RXDi	5	R/W	**RXD Invert Bit** 0 = Normal operation 1 = RXD input data inverted
Reserved	3:2	Read Only	**Undefined**
SMOD	1	R/W	**Baud Rate Doubler** 0 = Baud rate divided by 2 (mode 2) 0 = Baud rate divided by 4 (modes 1 and 3) 1 = Baud rate unchanged
FEDE	0	R/W	**Framing Error Enable Flag** 0 = SICON bit 7 controls communication mode (SM0) 1 = SICON bit 7 flags framing errors (FE)

Figure 12. Serial Interface Block Diagram



Serial Interface Modes of Operation

The UART operates in one of three asynchronous full-duplex communication modes with different protocols and baud rates. Two mode bits (SM1, SM0 in the SCON register) are used to select the modes of operation. The operating modes are summarized in Table 10.

Table 10. Serial Interface Modes

SM1	SM0	MODE	FUNCTION	CLOCK	LENGTH	START/STOP	9th BIT FUNCTION
0	1	1	Asynchronous Full Duplex	Phase Delta Register	8	1 Start, 1 Stop	N/A
1	0	2	Asynchronous Full Duplex	Instruction Clock	9	1 Start, 1 Stop	TB8, RB8
1	1	3	Asynchronous Full Duplex	Phase Delta Register	9	1 Start, 1 Stop	TB8, RB8

Mode 1

Mode 1 is asynchronous, full duplex with a 10-bit data stream consisting of a logic 0 start bit, eight data bits, and logic 1 stop bit. The data is transferred least significant bit first. The UART begins transmission several cycles after the first baud clock of the baud rate generator following a write to SBUF. Transmission on the TXD pin begins with the start bit, data then is shifted out on the pin, least significant bit first, followed by the stop bit. The TI bit is set two clock cycles after the stop bit is transmitted. All bits are shifted out at the rate determined by the baud rate generator. Baud rate generation is discussed in the *Baud Rate Generation* section.

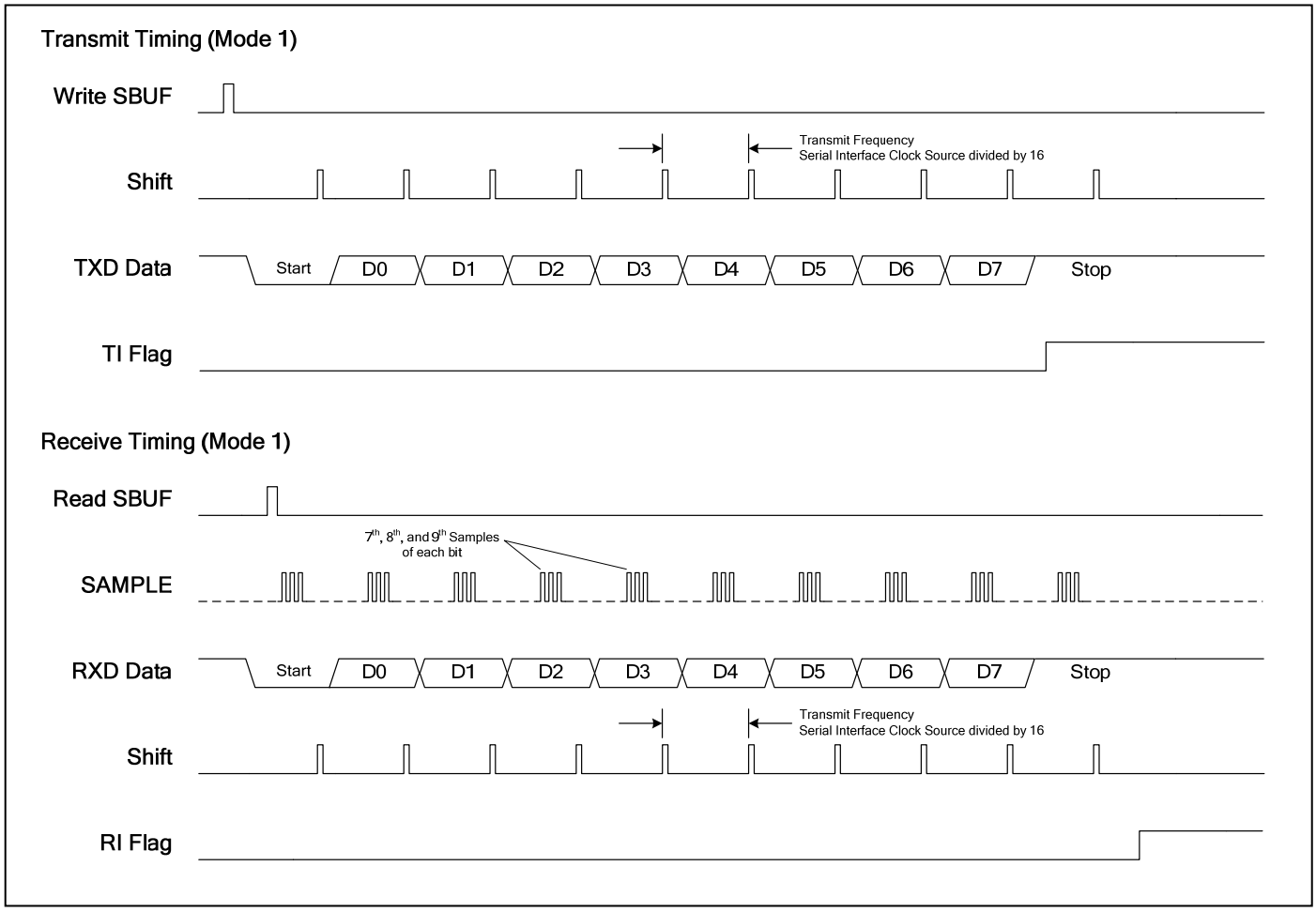
Once the baud rate generator is active, reception can begin at any time. The REN bit in SCON must be set to logic 1 to enable the reception. The falling edge of a start bit on the RXD pin will begin the reception process. Data will be shifted in at the selected baud rate. At the middle of the stop bit time slot, certain conditions must be met to load SBUF with the received data from the receive shift register:

- RI must be 0, and
- if SM2 is 0, the state of the stop bit does not matter, or
- if SM2 is 1, the state of the stop bit must be 1

If these conditions are true, the SBUF register will be loaded with the received byte, the RB8 bit will be loaded with the stop bits and the RI bit will be set. If these conditions are false, then SBUF and RB8 will not be loaded, the received data will be lost and the RI bit will not be set. Regardless of the receive word status, the receive logic will go back to looking for a 1 to 0 transition on the RXD pin after the middle of the stop bit time.

Each data bit received is sampled on the 7th, 8th, and 9th clock from the baud rate generator (running 16 times faster than communication speed). Using majority voting, two equal samples out of the three determine the logic value for each received bit. If the start bit was determined to be invalid, then the receive logic returns to looking for a 1 to 0 transition on the RXD pin in order to start the reception again.

Figure 13. Timing Diagram for Serial Interface Mode 1



Mode 2

Mode 2 is asynchronous, full-duplex communication with an 11-bit data stream consisting of a logic 0 start bit, eight data bits, a programmable 9th bit, and a logic 1 stop bit. The data is transferred least significant bit first. Similar to mode 1, transmissions occur on the TXD pin and reception on the RXD pin. For transmission purposes, the 9th bit can be configured as logic 0 or 1. A common use is to load this location with a parity bit. The 9th bit is transferred from the TB8 bit position in the SCON register during the write to SBUF. The UART will begin transmission a few cycles after the first rollover of the clock source divided by 16 following a write to SBUF. Transmission begins with the start bit being placed on the TXD pin, data is then shifted onto the pin, least significant bit first, followed by the 9th bit, and finally the stop bit. The TI bit is set at the same time the stop bit is transmitted on the TXD pin. All bits are shifted out at the rate determined by the baud rate generator. Baud rate generation is discussed in the *Baud Rate Generation* section.

Once the baud rate generator is active, reception can begin at any time. The REN bit must be set to a logic 1 to enable the reception. The falling edge of a start bit on the RXD pin will begin the reception process. Data will be shifted in at the selected baud rate. At the middle point of the 9th bit time slot, certain condition must be met to load SBUF with the received data from the receive shift register:

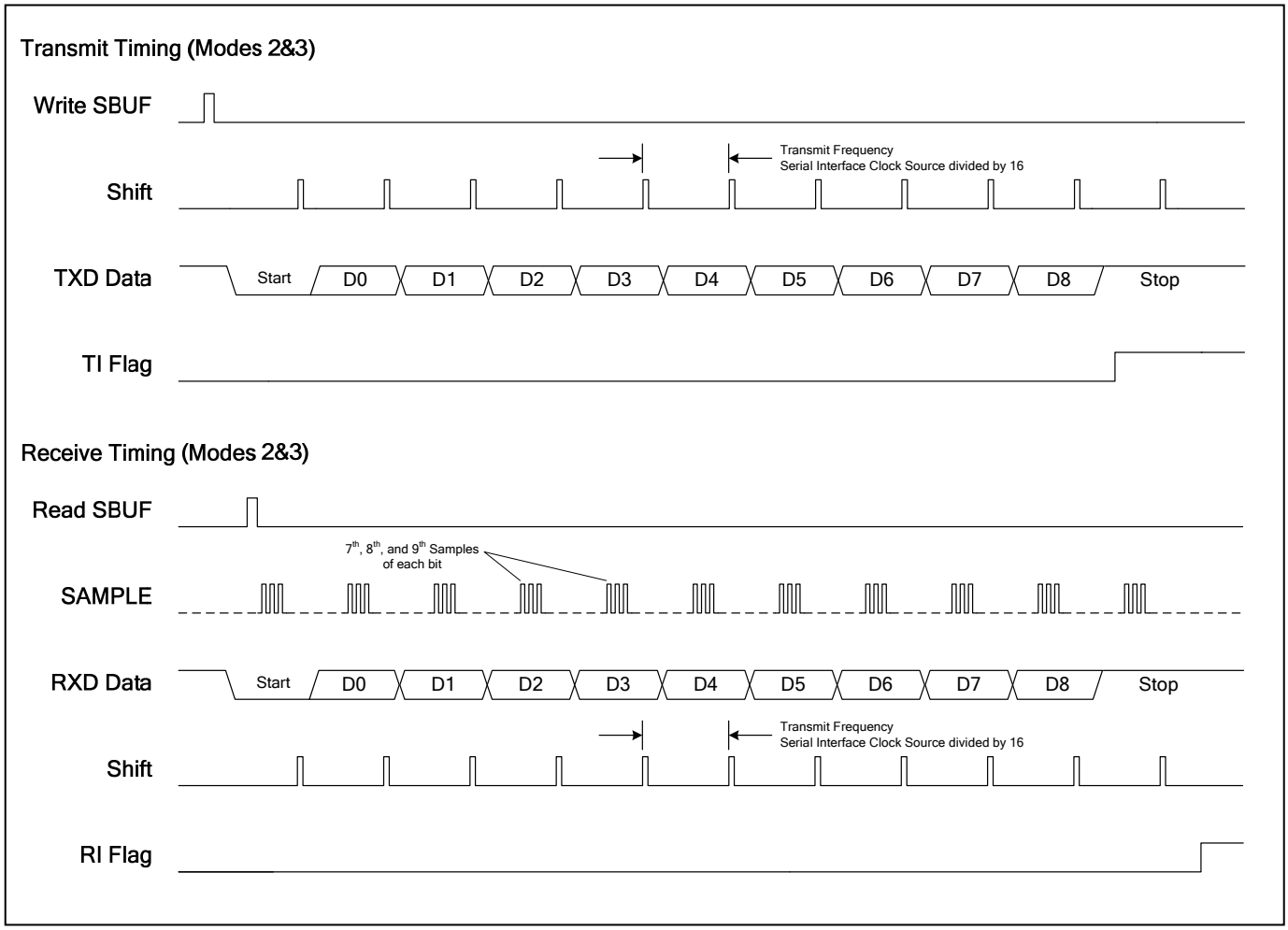
- RI must be 0, and
- if SM2 is 0, the state of the stop bit does not matter, or
- if SM2 is 1, the state of the 9th bit must be 1

If these conditions are true, then SBUF will be loaded with the received byte, the RB8 bit will be loaded with the 9th bit, and the RI bit will be set. If these conditions are false, then SBUF and RB8 will not be loaded, the received data will be lost, and RI bit will not be set. Regardless of the receive word status, the receive logic will go back to looking for a 1 to 0 transition on the RXD pin after the middle point of the stop bit time slot.

Each data bit received is sampled on the 7th, 8th, and 9th clock from the baud rate generator (running 16 times faster than communication speed). Using majority voting, two equal samples out of the three determine the logic value for each received bit. If the start bit was determined to be invalid, then the receive logic goes back to looking for a 1 to 0 transition on the RXD pin in order to start the reception again.

Mode 3
Operation of mode 3 is identical to mode 2 except for the baud rate source. Baud rate generation is discussed in the *Baud Rate Generation* section.

Figure 14. Timing Diagram for Serial Interface Modes 2 and 3



Detection of Framing Errors

A framing error occurs during communication when a valid stop bit is not detected. The result is a possible improper reception of the serial word. The serial inteface will notify the system by setting the framing error (FE) bit in the SCON register to 1. When set, the FE bit must be cleared by software or a system reset. Receiving a properly framed serial word will not clear the FE bit.

Note that both the FE and the serial mode bit 0 (SM0) share the same bit location in the SCON register, but this information is actually stored in different registers. The setting of FEDE in the SMD register determines which register is accessed: logic 0 allows access to the SM0 bit and a logic 1 allows access to the FE bit. The FEDE must be set to 1 while reading or writing the FE bit.

Multiprocessor Communication Mode

Multiprocessor communication mode makes special use of the 9th data bit in modes 2 and 3 if the SM2 bit of the SCON register is set. If enabled, the 9th data bit is used to signify that the incoming byte is an address. This allows the processor to be interrupted only if the correct address is received as defined by the serial address (SADDR) and serial address enable (SADEN) registers. The receive interrupt, if enabled, will only occur when a recognized address is received.

When multiprocessor mode is enabled and a serial word is received with the 9th bit set, the byte will be assumed to be an address. The address will be compared to an internally stored address. If it matches, a receive interrupt will occur. The internal address is derived from the SADDR and SADEN registers. The SADDR register specifies an absolute address. This is the user specified address of the device. The SADEN register indicates which address bit(s) will be used in the comparison. This allows broadcast transmissions that reach multiple microcontrollers or all microcontrollers on the serial interface. The user defines this protocol.

Software will write an 8-bit address to the SADDR register. This is the microcontroller's individual address. Any bit in SADEN that contains logic 0 will cause the corresponding bit in SADDR to be ignored in comparison. Thus, logic 0 bits in SADEN create don't care bit states for address comparisons. When an address is received, each address bit that is not masked by a don't care will be compared to the SADDR. The microcontroller will interrupt on any address that matches this comparison. Any address that meets this comparison is called a given address. The following example shows how one address can be directed to an individual processor or two out of three.

Micro 1	Micro 2	Micro 3
SADDR 11110000	SADDR 11110001	SADDR 11110010
SADEN 11111010	SADEN 11111001	SADEN 11111010
Given 11110x0x	Given 11110xx1	Given 11110x1x

Note that an address of 11110000 reaches only microcontroller 1. An address of 11110001 reaches microcontroller 1 and microcontroller 2. An address of 11110010 reaches only microcontroller 3. The microcontroller also matches on any address that corresponds to the broadcast address. This is the logical OR of the SADDR and SADEN registers, with any 0s defined as don't cares.

UART Interrupts

Interrupts generated by the serial interface are controlled through the serial interface interrupt register (SINT) and the serial interface interrupt mask register (SMASK). Writing either the receive interrupt mask (MRI) bit or transmit interrupt mask (MTI) bit to 1 will enable interrupts to occur whenever the corresponding transaction successfully completes. Writing either of these bits to 0 disables the corresponding interrupt. The RINT, TINT bits of the SMASK are a logical AND of the RI flag, TI flag bits of the control register, and the corresponding interrupt mask bits.

Upon successful reception of a data word on RXD, the RI flag of the SCON register will be set. If MRI is set to 1, the RINT flag will be set and an interrupt will be generated. System software can then read the SINT register to determine the source of the interrupt. The RINT flag and the interrupt are cleared by writing the RI flag to 0. After transmission of a data word on TXD, the TI flag of the SCON register will be set. If MTI is set to 1, the TINT flag will be set and an interrupt will be generated. System software can then read the SINT register to determine the source of the interrupt. The TINT flag and the interrupt are cleared by writing the TI flag to 0.

Command Codes

The DS2792 has two reserved command codes: a software power-on reset (POR) of the IC and an instruction to begin program loading over the serial interface. Each command code is enabled by transmitting a <break>, 0xFE, <break>, followed directly by the command instruction. There are no associated data bytes with either command. Any other serial communication between the instructions negates the operation. See the *MAXQ Family User's Guide: DS2792 Supplement* for the serial interface programming procedure. These command codes are fixed inside the DS2792 and cannot be altered. System firmware should avoid using FEh as a command code during device operation. A <break> instruction is defined as holding the RXD line low until a framing error is generated.

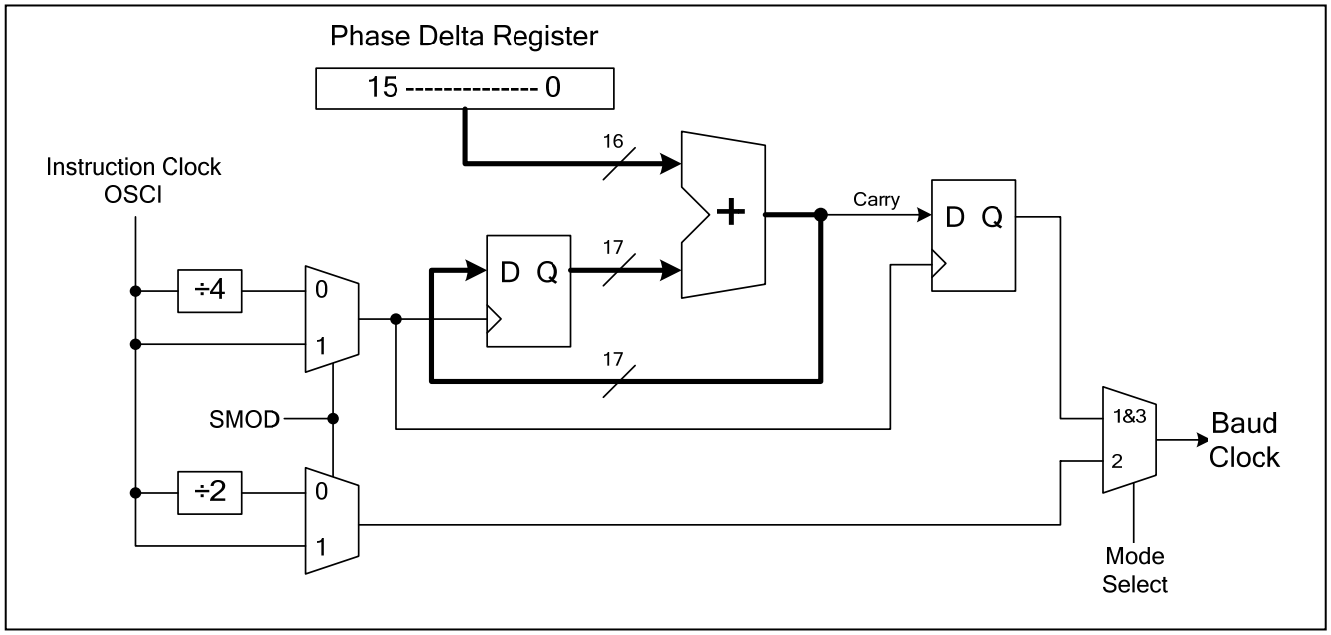
Table 11. Serial Interface Command Codes

COMMAND	HEX CODE	PURPOSE
Command Enable	<break>, FEh, <break>	Enable soft POR or program command.
Soft POR	FEh	Causes a reset of the IC.
Request Programming	FDh	Initiates programming over the serial interface.
Available	00h–FCh, FFh	Defined by application firmware.

Baud Rate Generation

The baud rate for serial communication is derived directly from the instruction oscillator OSCI. The method for baud rate generation depends on the serial interface's operating mode. In operating modes 1 and 3, the baud rate is determined by the value stored in the phase delta (PR) register. In operating mode 2, the OSCI oscillator sets the baud rate directly. The baud rate doubler (SMOD) bit in the SMD register can be used to double the baud rate in mode 2 or quadruple the baud rate in modes 1 and 3. Note that the baud clock frequency generated by this circuit will be the sampling frequency used by the serial interface and is therefore 16 times the actual communication rate.

Figure 15. Baud Rate Generation Circuit



In mode 2, baud rates are generated directly from the system clock. The baud rate for mode 2 is given by the formula: Baud Rate = $(2^{SMOD} \div 32) \times f_{OSCI}$. The result of this formula generates a baud rate of either 1/16 or 1/32 of the system clock frequency. In the above formula, the numerator is expressed as two to the power of SMOD, where SMOD is either a 0 or 1. The SMOD bit effectively doubles the baud rate when set to logic 1. The SMOD bit is set to logic 0 on reset, which gives the lower speed baud rate.

In modes 1 and 3, the baud rate generator creates a phase accumulator that generates a baud clock as the result of phase overflow into the most significant bit of the phase shift circuitry. As shown in Figure 15, a 16-bit phase delta register (PR) is used to select a suitable phase delta for its baud clock. The phase delta is used to provide a predetermined phase increment to the phase accumulator. The phase accumulator is formed by a 17-bit register and a 16-bit adder. When the baud rate generator is enabled, the content of the phase delta register adds to the value of the 17-bit register triggered by the system clock, essentially performing a phase accumulation. The baud clock is the result of the adder carry output to the most significant bit of the 17-bit register. The baud rate for modes 1 and 3 are given by the formula: baud rate = $(PR \times f_{OSCI}) \div 2^{23} \times 2^{(SMOD \times 2)}$. The SMOD bit effectively quadruples the baud rate when set to logic 1. The SMOD bit is set to logic 0 on reset, which gives the lower speed baud rate.

Table 12. Sample Baud Rate Settings

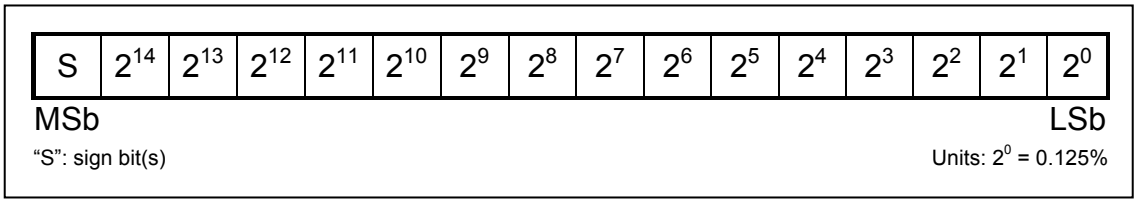
MODE	SMOD	PR REGISTER	BAUD RATE
2	0	N/A	30583 baud
2	1	N/A	61166 baud
1 or 3	1	1416h	2400 baud
1 or 3	1	282Dh	4800 baud
1 or 3	1	505Bh	9600 baud
1 or 3	1	A0B7h	19200 baud
1 or 3	0	282Dh	1200 baud

Baud rate calculations are based on a system clock frequency of 978.656KHz. Note that the analog oscillator must be active for reliable serial communication. If OSCA is inactive, the OSCI timebase error $f_{ERR:OSCI}$ may be greater than the UART communication standard of 3%.

Baud Rate Error Adjustment

The baud rate error (BRE) register provides error information that can be used by software to correct small errors in the DS2792's baud rate compared to the transmitting IC's baud rate by adjusting the value loaded in the phase delta register. At the end of each data word successfully received, the BRE register is updated with a 0.125% resolution two's complement value representing the percent difference between the expected final logic transition on TXD compared to the final logic transition as measured by the bit sampler. A positive error means the UART is running too slow and a negative error means the UART is running too fast.

Figure 16. Baud Rate Error Register Format



For example: Communication is occurring at 9600 baud (PR register contains 505Bh) in mode 1, and a value of C5h is received by the UART. The final RXD transition between bit D7 (logic 0) and the stop bit (logic 1) is expected to occur between sample 16 of the 7th bit and sample 1 of the 8th bit. After transmission is complete, the BRE register is found to contain a value of 08h indicating the final logic transition occurred 1.0% early. The master is transmitting data at a rate of roughly 1.0% faster than the DS2792's present baud rate. The PR register is updated to a value of 5129h (101% of 505Bh) to correct the transmission rate difference. It is recommended that the PR register is adjusted by only a fraction of the calculated error amount after each byte to prevent oscillation of the frequency setting.

Note that for dynamic baud rate error adjustment to be successful, the starting frequency difference must be within ±½ bit period at the final logic transition point (±50% if the final transition is between start and D0, ±25% if the final transition is between D0 and D1, etc). The BRE register is read only and is cleared on reset.

ANALOG-TO-DIGITAL CONVERSION

The DS2792 performs real-time measurements of system temperature, voltage, current, and accumulated current. The DS2792's ADC is controlled by an internal state machine that sequences the measurements, and stores the results in memory. The conversion results of the ADC are mapped into data memory starting at word address 6000h, as shown in Table 13.

The DS2792 current measurement system is designed to provide timely data on charge and discharge current at a moderate resolution level while simultaneously accumulating high resolution average data to support accurate coulomb counting. Current is measured by sampling the voltage drop across a series sense resistor, R_{SNS}, connected between SNS1 and SNS2. Individual current samples are taken every $1/f_{SAMPLE}$ (687µs). All samples are averaged to report current, average current, and accumulated current values.

The DS2792 measures voltage on three separate inputs. The voltage of the lower cell, VIN1, is reported as the measured difference between the VIN1 pin and analog ground pin AV_{SS}. The voltage of the upper cell, VIN2, is reported as the difference between the VIN2 and VIN1 pins. The auxiliary voltage, Vx, is reported both as an absolute value of the difference between the Vx pin and AVSS, and as a ratio metric percentage of the voltage of the V_B pin. Individual voltage samples are taken approximately every $5/f_{SAMPLE}$ (3.43ms). Multiple samples are averaged to update the voltage measurement registers.

The DS2792 measures temperature directly on chip. Individual temperature samples are taken every $10/f_{SAMPLE}$ (6.87ms). Multiple samples are averaged to update the average temperature register.

Table 13. ADC Related Registers

WORD ADDRESS	ACCESS	DESCRIPTION
6000h	Read Only	Vx Voltage Ratiometric
6001h	Read Only	Vx Voltage Absolute
6002h	Read Only	VIN2 Voltage
6003h	Read Only	VIN1 Voltage
6004h	Read Only	Current
6005h	Read Only	Temperature
6006h	R/W	Accumulated Current
6007h	Read Only	Accumulated Current (Middle Word)
6008h	Read Only	Accumulated Current (Lower Word)
6009h	Read Only	Average Current
600Ah	R/W	ADC Configuration

Current Measurement

The voltage signal developed across the sense resistor (between SNS1 and SNS2) is differentially sampled by the ADC inputs via internal 10kΩ resistors connected between SNS1 and IS1, and SNS2 and IS2. Isolating the ADC inputs (IS1 and IS2 pins) from the sense resistor with 10kΩ facilitates the use of an RC filter by adding a single external capacitor. The RC filter extends the effective input range beyond ±64mV in pulse load or pulse charge applications. The ADC accurately measures large peak signals as long as the differential signal level at IS1 and IS2 does not exceed ±64mV.

The current register reports the average of 128 individual current samples every 88ms. The reported value represents the average current during the 88ms measurement period. Figure 17 specifies the update interval and resolution of the current register. Values are posted in two's complement format. Positive values represent charge currents ($V_{IS1} > V_{IS2}$) and negative values represent discharge currents ($V_{IS2} > V_{IS1}$). Positive currents above the maximum register value are reported at the maximum value, 7FFFh. Negative currents below the minimum register value are reported at the minimum value, 8000h.

Figure 17. Current Register Format

12-bit + sign resolution (13-bit), 88ms update interval

Word Address 6004h

S	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	X	X	X

MSb LSb

"S": sign bit(s)

Units: $2^0 = 15.625\mu V/R_{SNS}$

CURRENT RESOLUTION (1 LSB)				
$\lvert V_{SS} - V_{SNS} \rvert$	R_{SNS}			
	25mΩ	20mΩ	15mΩ	10mΩ
15.625µV	625µA	781µA	1.042mA	1.56mA

CURRENT INPUT RANGE				
$V_{SS} - V_{SNS}$	R_{SNS}			
	25mΩ	20mΩ	15mΩ	10mΩ
±64mV	±2.56A	±3.2A	±4.27A	±6.40A

The average current register reports the average of 4096 current samples over a 2.8s measurement period. Figure 18 specifies the update interval and resolution of the average current register. Values are posted in the same format as the current register, but with an additional 3 bits of resolution. Positive currents above the maximum register value are reported at the maximum value, 7FFFh. Negative currents below the minimum register value are reported at the minimum value, 8000h.

Figure 18. Average Current Register Format

15-bit + sign resolution (16-bit), 2.8s update interval

Word Address 6009h

S	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

MSb LSb

"S": sign bit(s)

Units: $2^0 = 1.953\mu V/R_{SNS}$

AVERAGE CURRENT RESOLUTION (1 LSB)				
$\lvert V_{SS} - V_{SNS} \rvert$	R_{SNS}			
	25mΩ	20mΩ	15mΩ	10mΩ
1.953µV	78.13µA	97.66µA	130.2µA	195.3µA

Automatic Current Offset Correction

Continuous offset cancellation is performed automatically to correct for offsets in the current measurement system. Individual values reported by the current register have a maximum offset of ±0.5 bits (±7.8125µV). Individual values reported in the average current register have a maximum offset of ±4 bits (±7.8125µV).

Current Measurement Bias

Systematic errors or an application preference can require the application of an arbitrary bias to the current measurement process. The current measurement bias value sets a user-programmed positive or negative bias to the current measurement. The current measurement bias value can be used to estimate battery currents that do not flow through the sense resistor, estimate battery self-discharge, or correct for offset error in the current registers. The user-programmed two's complement value is added to the current register once per current sample. The bias control is applied in 0.98μV increments over a ±125μV range. When using a 15mΩ sense resistor, the bias control can be adjusted in 65.3μA increments over a ±8.33mA range. The current measurement bias bit field is located in the upper byte of the ADC configuration register.

Figure 19. Current Measurement Bias Field

Upper Byte of Word Address 600Ah

S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

MSb LSb

"S": sign bit

Units: $2^0 = 0.98\mu V/R_{SNS}$

Current Accumulation

Current measurements are internally accumulated and displayed in the accumulated current register (ACR). The accuracy of the ACR is dependent on both the current measurement and the accumulation time base. The 16-bit ACR has a range of ±204.8mVh with a resolution of 6.25μVh. Accumulation of charge current above the maximum register value is reported at the maximum value; conversely, accumulation of discharge current below the minimum register value is reported at the minimum value. Read and write access is allowed to the ACR.

Figure 20. Accumulated Current Register Format

Word Address 6006h

S	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

MSb LSb

"S": sign bit(s)

Units: $2^0 = 6.25\mu Vh/R_{SNS}$

ACR RESOLUTION (1 LSB)				
$\lvert V_{SS} - V_{SNS} \rvert / R_{SNS}$	R_{SNS}			
	25mΩ	20mΩ	15mΩ	10mΩ
6.25μVh	0.25mAh	0.312mAh	0.416mAh	0.625mAh

ACR ACCUMULATION RANGE				
$\lvert V_{SS} - V_{SNS} \rvert / R_{SNS}$	R_{SNS}			
	25mΩ	20mΩ	15mΩ	10mΩ
±204.8mVh	±8.192Ah	±10.24Ah	±13.65Ah	±20.48Ah

The lower 32 bits of ACR resolution (bits 2^{-1} to 2^{-32}) can be read by firmware from address locations 6007h and 6008h respectively. Note that since the lower bits are from separate address words it cannot be guaranteed that they will contain data from the same measurement as the main ACR register at the time of reading. However, two consecutive reads from addresses 6006h–6008h that contain the same data ensures that all data is from the same measurement. When the ACR register is written, the lower ACR bits are automatically cleared.

Accumulation Blanking

In order to avoid the accumulation of small positive offset errors over long periods, an offset blanking filter is provided. The blanking filter is enabled by setting the OBEN bit in the ADC configuration register. When OBEN is set, charge currents (positive values from the current register) less than 62.5μV are not accumulated in the ACR. The minimum charge current accumulated in the ACR is 4.167mA for R_{SNS} = 0.015Ω.

Voltage Measurements

The DS2792 continually measures the voltage between pins VIN1 and AV_{SS} and between pins VIN2 and VIN1 over a 0.0V to V_{FS} range. The VIN1 and VIN2 voltage registers are updated in two's complement format every 13.8ms with a resolution of 4.88mV. Voltages above the maximum register value are reported as the maximum value.

Figure 21. VIN1 Voltage Register Format

Word Address 6003h

S	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	X	X	X	X	X

MSb LSb

"S": sign bit

Units: 2^0 = 4.88 mV

Figure 22. VIN2 Voltage Register Format

Word Address 6002h

S	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	X	X	X	X	X

MSb LSb

"S": sign bit

Units: 2^0 = 4.88 mV

The DS2792 continually measures the voltage between pins Vx and V_{SS}. The result is reported in two's complement absolute and ratiometric formats every 20ms. The Vx ratiometric voltage result is a ratio of voltage on the Vx pin relative to the voltage on the V_B pin. The LSb weighting is V_B/1024. The Vx absolute voltage result uses the internal reference and has a resolution of 4.88mV. Voltages above the maximum register value are reported as the maximum register value.

Figure 23. Vx Ratiometric Register Format

Word Address 6000h

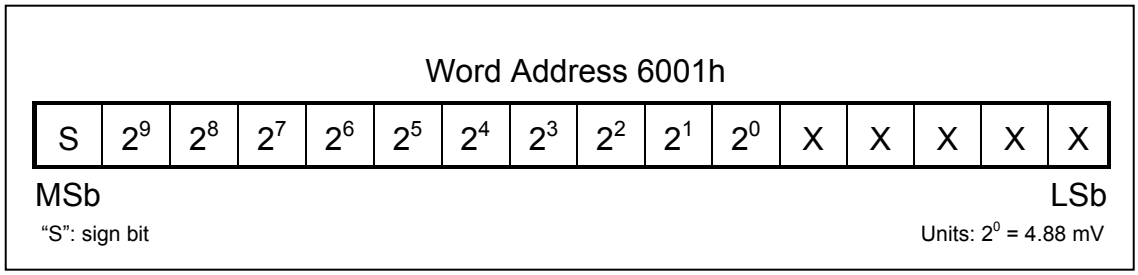
S	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	X	X	X	X	X

MSb LSb

"S": sign bit

Units: $2^0 = V_B/1024$

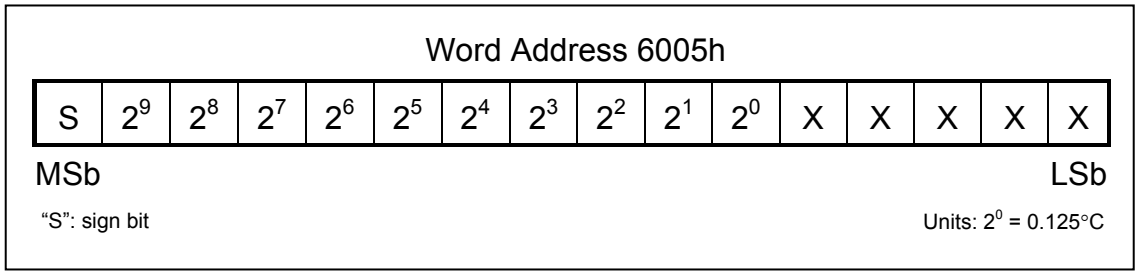
Figure 24. Vx Absolute Voltage Register Format



Temperature Measurement

The DS2792 uses an integrated temperature sensor to continually measure battery temperature. Temperature measurements are updated in the temperature register every 220ms in two's complement format with a resolution of 0.125°C over a ±127°C range. The temperature register format is shown in Figure 25.

Figure 25. Temperature Register Format



ADC Configuration Register

The ADC configuration register located at word address 600Ah controls current measurement bias and offset blanking. ADC configuration register bits are read and write accessable by application code. IBIAS bits are described in the current measurement bias section above. Offset blanking is controlled by the OBEN bit. If offset blanking is enabled, the DS2792 assumes small positive currents do not occur in the application. Small positive readings are considered A/D measurement error and are not accumulated.

Figure 26. ADC Configuration Register Format

ADDRESS	600Ah		BIT DEFINITION
Field	**Bit**	**Format**	**Allowable Values**
IBIAS	15:8	R/W	**Current Measurement Bias** 8-bit two's complement value that is added to the current measurement on every update.
Reserved	7:1	Read Only	**Undefined**
OBEN	0	R/W	**Offset Blanking Enable** 0 = All current measurements are accumulated into the ACR. 1 = Positive current measurements less than 62.5µV/R_{SNS} are not accumulated into the ACR.

PASSWORD-PROTECTED USER TRIM

System software can change current measurement gain, power-up state of the UART, and undervoltage threshold of the IC through the user trim in program EEPROM (word addresses 001Eh–001Fh). The user trim values are enabled through the trim key in the lower byte of address 001Fh. If the trim key is set to 76h, the user-trim values replace the default trim values, if the trim key is set to any other value, default trim is selected. Note that either all user-trim values are enabled or none are enabled. Figure 27 shows the format of all values that can be adjusted and their default trim values.

Figure 27. User Trim Registers

ADDRESS	001Eh		BIT DEFINITION
Field	**Bit**	**Format**	**Allowable Values**
IG	15:8	R/W	**Current Gain Trim** These bits adjust the current gain by ±25%. The most significant bit is the two's complement sign bit, 1 LSB = 0.195%. Example: A4h (-92d) adjusts the trim by -17.94%. Valid only if trim key = 76h Default = factory trim value
SRTC	7:0	R/W	**Sense Resistor Temperature Coefficient** These bits adjust the current gain based on temperature of the sense resistor. 1 LSB = 30.5ppm/ºC. Example: 1Ah (26d) adjusts current measurements for a sense resistor with a 793ppm/ºC temperature coefficient. Valid only if trim key = 76h Default = 00h

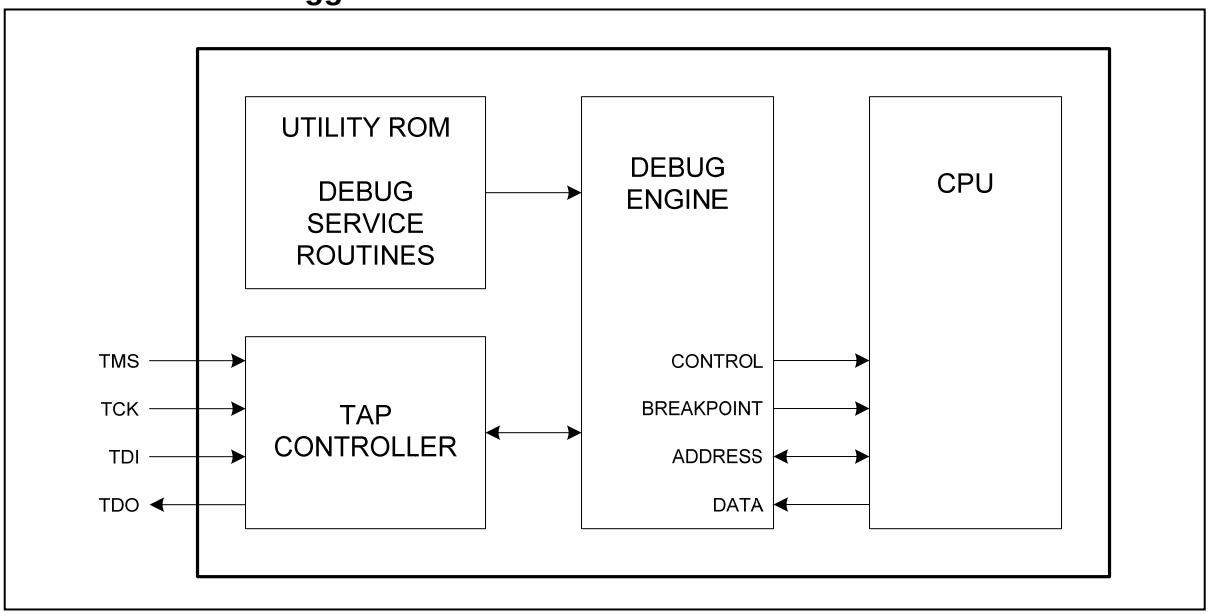
ADDRESS	001Fh		BIT DEFINITION
Field	**Bit**	**Format**	**Allowable Values**
BD9600	15	R/W	**Force UART BAUD Rate at Power-On Reset** 0 = 19200 BAUD 1 = 9600 BAUD Valid only if trim key = 76h Default = 0
TXDi	14	R/W	**TXDi Bit Power-On Reset Value** 0 = Normal Operation 1 = TXD Output Signal Inverted Valid only if trim key = 76h Default = 0
RXDi	13	R/W	**RXDi Bit Power-On Reset Value** 0 = Normal Operation 1 = RXD Input Data Inverted Valid only if trim key = 76h Default = 0
UVT	12:8	R/W	**Undervoltage Threshold** The undervoltage threshold ranges from 4.60V to 5.80V and is calculated by the equation: V_{UV} = 5.80V - 0.039V × UVT[4:0] Valid only if trim key = 76h Default = 17h (2.45V)
Trim Key	7:0	R/W	**Trim Key** Enables or disables all other user-trim values. 76h = all user trim values valid Other = all user trim values invalid (default trim used)

IN-CIRCUIT DEBUG

Embedded debugging capability is available through the JTAG-compatible test access port. 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 28 shows a block diagram of the in-circuit debugger. The in-circuit debug features include:

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

Figure 28. In-Circuit Debugger



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 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.

APPLICATIONS

The low-power, high-performance RISC architecture of the DS2792 makes it an excellent fit for many portable or battery-powered applications that require cost-effective computing and analog measurement capability. The high-throughput core is programmable in circuit over the UART and JTAG interfaces, allowing for firmware upgrades and ease of code development. Applications benefit from UART and GPIO peripheral interfaces allowing the microcontroller to communicate with many external devices. The DS2792's high level of integration reduces component count and board space, critical factors in the design of portable systems.

The DS2792 is ideally suited for applications such as fuel gauging, sensor conditioning, and data collection.

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/DS2792.

- The DS2792 data sheet, which contains electrical/timing specifications and pin descriptions, available at www.maxim-ic.com/DS2792.
- The DS2792 errata sheet, available at www.maxim-ic.com/errata.
- The *MAXQ Family User's Guide*, which contains detailed information on core features and operation, including programming at www.maxim-ic.com/products/MAXQUG.
- The *MAXQ Family User's Guide: DS2792 Supplement*, which contains detailed information on features specific to the DS2792 at www.maxim-ic.com/DS2792UG.

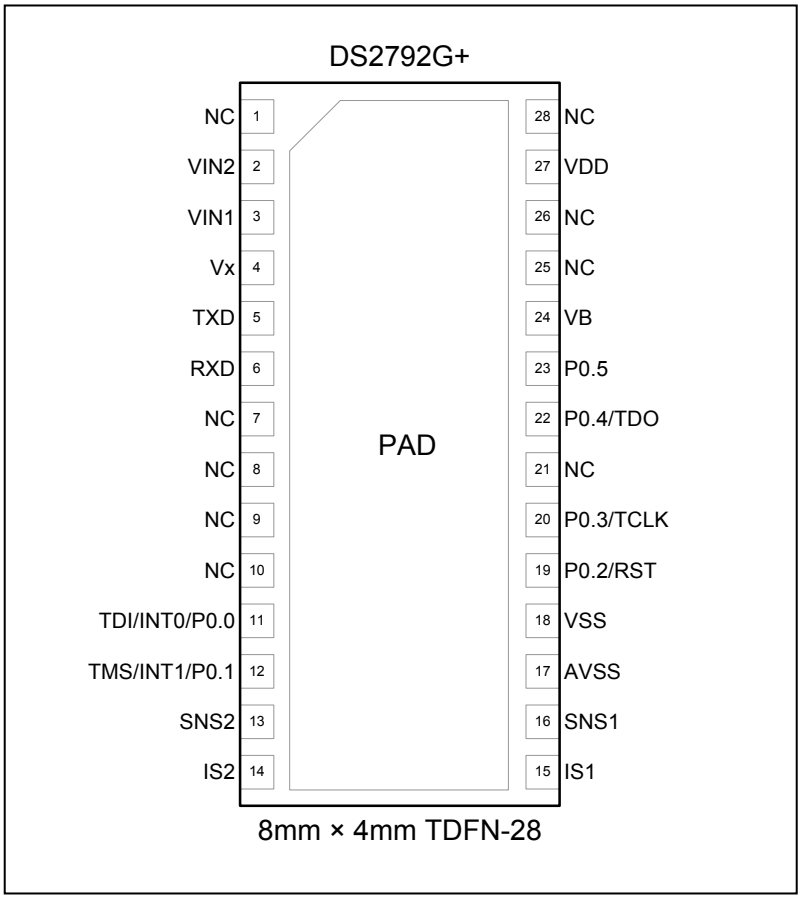
DEVELOPMENT AND TECHNICAL SUPPORT

A variety of highly versatile, affordably priced development tools for this microcontroller are available from Maxim/Dallas Semiconductor and third-party suppliers, including:

- Compilers
- In-circuit emulators
- 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 batterymanagement.support@dalsemi.com.

PIN CONFIGURATION



PACKAGE INFORMATION

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information, go to **www.maxim-ic.com/DallasPackInfo**.)