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

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

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

Contact us

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

DS2790
Programmable 1-Cell Li-Ion Fuel Gauge and Protector

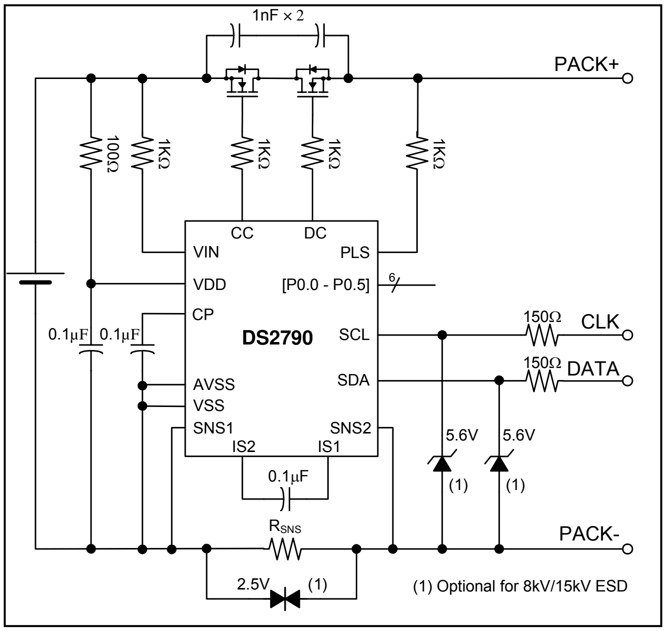
www.maxim-ic.com

GENERAL DESCRIPTION

The DS2790 provides a complete fuel gauging and protection solution for single cell Li-Ion 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 fuelgauge algorithms. The 2-wire interface provides an I^{2}C- or SMBus™-compatible communication path between the host and battery pack, while providing password protected programming of the fuel-gauging firmware. EEPROM data memory supports nonvolatile in-pack storage of charge parameters, cell characteristics, usage history, and manufacturing/lot tracking data.

An autonomous state machine performs voltage, current, and temperature related protection functions. This capability increases reliability of the whole system by eliminating dependence on the CPU for protection. The DS2790 supports Li-Ion batteries in a wide range of applications.

TYPICAL OPERATING CIRCUIT



PIN CONFIGURATION

See last page for TSSOP and TDFN packages.

FEATURES

- **Accurate Current Measurement for Coulomb Counting (Current Accumulation)**
  1.5% ±7.8µV over ± 64mV Input Range
  1.5% ±520µA over ±4.2A Range Using an External 15mΩ Series Resistor
- **High Resolution Current Reporting**
  12-bit + Sign Average Every 88ms
  15-bit + Sign Average Every 2.8s
- **Voltage Measurement**
  10-bit Average
- **Temperature Measurement**
  10-bit Using On-Chip Sensor
- **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
- **State Machine-Driven Protection**
  Protection Independent of CPU Operation
  Programmable Levels for:
  – Overvoltage/Undervoltage
  – Overcurrent
  – Temperature Limits
- **Lithium-Ion Protector Drives Highside N-FETs**
- **Industry Standard 400kHz 2-Wire interface**
  Password Protected Programming
- **Operates as Low as 2.5V Input on V_{DD}**
- **SHA-1 Hash Algorithm in ROM**
- **Internal Oscillator—No Crystal Required**
- **Low Power Consumption**
  3.3mA CPU Mode (1MHz), 280µA Analog Mode, 4.5µA Sleep Mode

ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE
DS2790E+	-20°C to +70°C	TSSOP-28
DS2790G+	-20°C to +70°C	TDFN-28

Contact factory concerning Mask ROM devices.
+ Denotes lead-free package.
MAXQ is a registered trademark of Maxim Integrated Products, Inc.
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, click here: www.maxim-ic.com/errata.*

ABSOLUTE MAXIMUM RATINGS

PLS to V_{SS}	-0.3V to +18V
CP to V_{SS}	-0.3V to +12V
DC to V_{SS}	-0.3V to CP+0.3V
CC to V_{SS}	V_{DD} -0.3V to CP+0.3V
P0.4, P0.5 to V_{SS}	-0.3V to V_{DD} +0.3V
AVSS to V_{SS}	-0.3V to +0.3V
All other pins to V_{SS}	-0.3V to +6V
SCL, SDA, P0.0–P0.5 Continous Sink Current	20mA Each, 50mA Total
P0.4, P0.5 Continous Source Current	20mA Each, 50mA Total
CC, DC Continuous Source/Sink Current	5mA
Operating Temperature Range	-40ºC to +85ºC
Storage Temperature Range	-55ºC to +125ºC
Soldering Temperature	See IPC/JEDEC J-STD-020A 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.

DC ELECTRICAL SPECIFICATIONS

(V_{DD} = 2.5V to 5.5V, T_A = -20°C to +70°C unless otherwise noted. Typical values are at V_{DD} = 3.7, T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current	I_{CPU}	CPU Mode (Note 1, 2)		1.5	3.3	mA
	I_{ANALOG}	ANALOG Mode (Note 2)		160	280	µA
	I_{SLEEP}	SLEEP Mode, (Note 2)			12.0	µA
		SLEEP Mode, (Note 2) V_{DD} = 4.2V, $T_A \le 50°C$		2.5	4.5	
		SLEEP Mode, (Note 2) V_{DD} = 2.5V, $T_A \le 50°C$		1.7	3.5	
Brownout Voltage	V_{BO}	(Note 3)	2.0		2.4	V
Power-On Reset Voltage	V_{POR}	(Note 3)		1.5		V
Internal System Clock	f_{OSCI}			1.0		MHz
System Clock Error	$f_{ERR:OSCI}$				±20	%
System Clock Startup	$t_{SU:OSCI}$	OSCA Active		1.0		µs
		From SLEEP, OSCA Inactive		700		
PLS Voltage Range		(Note 3)	-0.3		15	V
P0.4–P0.5 Voltage Range		(Note 3)	-0.3		V_{DD} + 0.3	V
P0.0–P0.3, SCL, SDA Voltage Range		(Note 3)	-0.3		+5.5	V
SCL, SDA, Input Logic High	V_{IH1}	(Note 3)	1.5			V
SCL, SDA, Input Logic Low	V_{IL1}	(Note 3)			0.6	V
P0.0 - P0.5, Input Logic High	V_{IH2}	(Note 3)	0.7 × V_{DD}			V
P0.0 - P0.5, Input Logic Low	V_{IL2}	(Note 3)			0.3 × V_{DD}	V
SCL, SDA, P0.0–P0.5 Output Logic Low:	V_{OL1}	I_{OL} = 4mA, (Note 3)			0.4	V
P0.4–P0.5 Output Logic High:	V_{OH1}	I_{OH} = -4mA, PPU[5:4] set, (Note 3)	V_{DD} – 0.5			V
SCL, SDA Pulldown Current	I_{PD1}	V_{PIN} = V_{IL1}, PPU[7:6] clear	0.3	1.2	3.0	µA
SCL, SDA Pullup Current	I_{PU1}	V_{PIN} = V_{IH1}, PPU[7:6] set	0.3	1.2	3.0	µA
P0.0–P0.3 Pullup Current	I_{PU2}	V_{PIN} = V_{IH2}, PPU[3:0] set	0.15	4	22	µA

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
P0.0–P0.5 Pulse Rejection	t_{PR}	Rising and Falling Edges		100		ns
Current Measurement Input Range (Full Scale)	V_{IS1}–V_{IS2}		-64		+64	mV
Current Measurement Resolution	I_{LSB}			15.625		µV/R_{SNS}
Current Measurement Offset Error	I_{OERR}		-7.8		+7.8	µV/R_{SNS}
Current Measurement Gain Error	I_{GERR}		-0.8		+0.8	% Full Scale
Accumulated Current Offset	q_{CA}	OBEN = 1	-188		0	µVh/day
		OBEN = 1, RSNS = 0.015Ω	-12.6		0	mAh/day
Temperature Measurement Resolution	T_{LSB}			0.125		°C
Temperature Measurement Error	T_{ERR}		-3		+3	°C
Voltage Full Scale	V_{FS}	(Note 4)	0		4.992	V
Voltage Measurement Resolution	V_{LSB}			4.88		mV
Voltage Measurement Error	V_{ERR}		-20		+20	mV
VIN Input Resistance	R_{IN}		15			MΩ
Current Measurement Sample Frequency	f_{SAMP}			1456		Hz
Analog Timebase Frequency	f_{OSCA}			70		KHz
Analog Timebase Error	$f_{ERR:OSCA}$	$V_{DD} \leq 4.5V$, $T_A = 25^{o}C$	-0.7		+0.7	%
			-2		+2	
Filter Resistors IS1 to SNS1, IS2 to SNS2	R_{KS}			10		kΩ
EEPROM Copy Time	t_{EEC}	$V_{DD} \geq 2.8V$		10	15	ms
EEPROM Copy Endurance Data EEPROM	N_{EECD}	$V_{DD} \geq 2.8V$, $T_A = 50^{o}C$	50,000			cycles
EEPROM Copy Endurance Program EEPROM	N_{EECP}	$V_{DD} \geq 2.8V$, $T_A = 50^{o}C$	1000			cycles

ELECTRICAL CHARACTERISTICS: PROTECTION CIRCUITRY

($2.5V \leq V_{DD} \leq 5.5V$, T_A = 0°C to +50°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Low: CC	V_{OLCC}	I_{OL} = 0.1mA, (Note 3)			V_{DD} + 0.1	V
Output Low: DC	V_{OLDC}	I_{OL} = 0.1mA, (Note 3)			0.1	V
Output High: CC	V_{OHCC}	I_{OH} = -0.1mA, (Note 3)	V_{OCP} - 0.25			V
Output High: DC	V_{OHDC}	I_{OH} = -0.1mA, (Note 3)	V_{OCP} - 0.25			V
Output Resistance: CC, DC	R_O	V_{OCP} = 9V, $V_{PIN} = V_{SS}$			2.0	kΩ
Output Voltage: CP	V_{OCP}	$I_{CC} + I_{DC} \leq 0.9\mu A$, (Note 3)	8.5	9.0	9.5	V
Overvoltage Detect	V_{OV}	OV = 01010b, (Note 3)	4.330	4.350	4.370	V
Charge Enable	V_{CE}	OV = 01010b, (Note 3)	4.230	4.250	4.270	V

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Undervoltage Detect	V_{UV}	UVF = 10111b, (Note 3)	2.430	2.450	2.470	V
Charge and Discharge Overcurrent Detect (Limits for Charge Thresholds are Positive, While Discharge is Negative.)	V_{OC}	COCT = DOCT = 00b	15.6	16	16.4	mV
		COCT = DOCT = 01b	31.2	32	32.8	mV
		COCT = DOCT = 10b	47.0	48	49.0	mV
		COCT = DOCT = 11b	62.7	64	65.3	mV
Short-Circuit Detect	V_{SC}	DOCT = 00b	75	100	125	mV
		DOCT = 01b	105	140	175	mV
		DOCT = 10b	135	180	225	mV
		DOCT = 11b	165	220	275	mV
Overvoltage Delay	t_{OVD}		0.8	1	1.2	s
Undervoltage Delay	t_{UVD}		75	100	125	ms
Overcurrent Delay	t_{OCD}		15	20	25	ms
Short-Circuit Delay	t_{SCD}	SCDT = 1	1.5	2	2.5	ms
		SCDT = 0	187	250	313	µs
Secondary Short-Circuit Delay	t_{SSCD}	(Note 5)	20		200	µs
Test Threshold	V_{TP}		0.3	1.0	1.5	V
Test Current	I_{TST}		10	20	40	µA
Pulldown Current, PLS	I_{PD}	Sleep Mode		200		µA
Recovery Charge Current	I_{RC}	VPLS = 5.0V, V_{DD} = 2.0V	0.5	1	2	mA

ELECTRICAL CHARACTERISTICS: 2-WIRE INTERFACE

(2.5V ≤ V_{DD} ≤ 5.5V, T_A = -20°C to +70°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}		0		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 6)	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	$t_{HD:DAT}$	(Note 7, 8)	0		0.9	µs
Data Setup Time	$t_{SU:DAT}$	(Note 7)	100			ns
Rise Time of both SDA and SCL Signals	t_R	(Note 9)	20+0.1C_B		300	ns
Fall Time of both SDA and SCL Signals	t_F	(Note 9)	20+0.1C_B		300	ns

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Setup Time for STOP Condition	$t_{SU:STO}$		0.6			µs
Spike Pulse Width that can be Suppressed by Input Filter	t_{SP}	(Note 10)	0		50	ns
Clock Low Time-Out	$t_{TIMEOUT}$	TTO_DIS = 0, (Note 11)	25		35	ms
Cumulative Clock Low Extend Time for Slave Device	$t_{LOW:SEXT}$	TLS_DIS = 0, (Note 12)			25	ms
Cumulative Clock Low Extend Time for Bus Master	$t_{LOW:MEXT}$	TTO_DIS = 0, TLS_DIS = 0 (Note 13)			10	ms
SCL, SDA Input Capacitance	C_{BIN}				60	pF

ELECTRICAL CHARACTERISTICS: JTAG INTERFACE

($2.5V \leq V_{DD} \leq 5.5V$, T_A = -20°C to +70°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
JTAG Logic Reference	V_{REF}			$V_{DD} \div 2$		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: Maximum current assumuing 100% CPU duty cycle.

Note 2: This value does not include current in SDA, SCL, and P0.0–P0.5.

Note 3: All Voltages referenced to V_{SS}.

Note 4: Voltage register can report up to 4.992V, however VIN pin input saturation occurs at 4.75V minimum.

Note 5: The secondary short circuit delay is measured from the falling transition on V_{DD} to the resultant falling transition on DC. The delay is measured from the time V_{DD} reaches V_{POR} - 0.5V to the time DC reaches 50% of V_{CP} (4.5V).

Note 6: f_{SCL} must meet the minimum clock low time plus the rise/fall times.

Note 7: The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.

Note 8: This device internally provides a hold time of at least 75ns for the SDA signal (referred to the VIHmin of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 9: C_B—total capacitance of one bus line in pF.

Note 10: Filters on SDA and SCL suppress noise spikes at the input buffers and delay the sampling instant.

Note 11: Devices participating in data transfer will timeout when any clock low exceeds the minimum $t_{TIMEOUT}$ of 25ms. Devices that have detected a timeout condition must reset the communication no later than the maximum $t_{TIMEOUT}$ of 35ms. The maximum value specified must be adhered to by both devices as it incorporates the cumulative stretch limit for the master (10ms) and slave device (25ms).

Note 12: $t_{LOW:SEXT}$ is the cumulative time the slave is allowed to extend the clock from the initial START to the STOP. If the DS2790 exceeds this time, it will release both SDA and SCL and reset the communication interface.

Note 13: $t_{LOW:MEXT}$ is the cumulative time the master is allowed to extend the clock cycles within each byte of a communication sequence. If the bus master exceeds this time it is possible for the DS2790 to violate $t_{TIMEOUT}$ without having violated $t_{LOW:SEXT}$.

Figure 1. 2-Wire Bus Timing Diagram

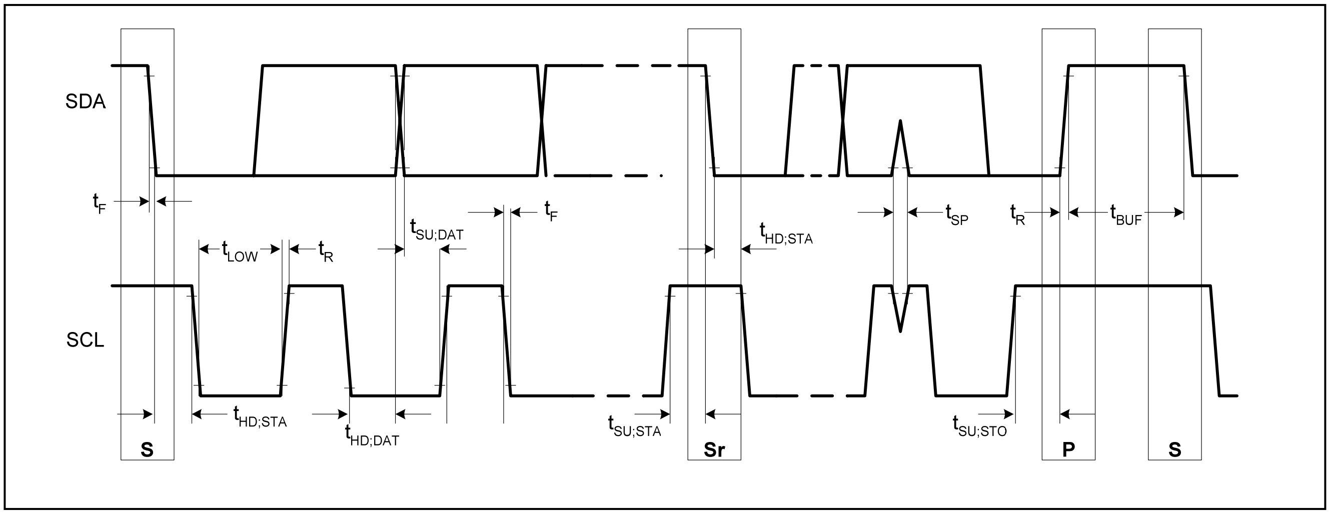
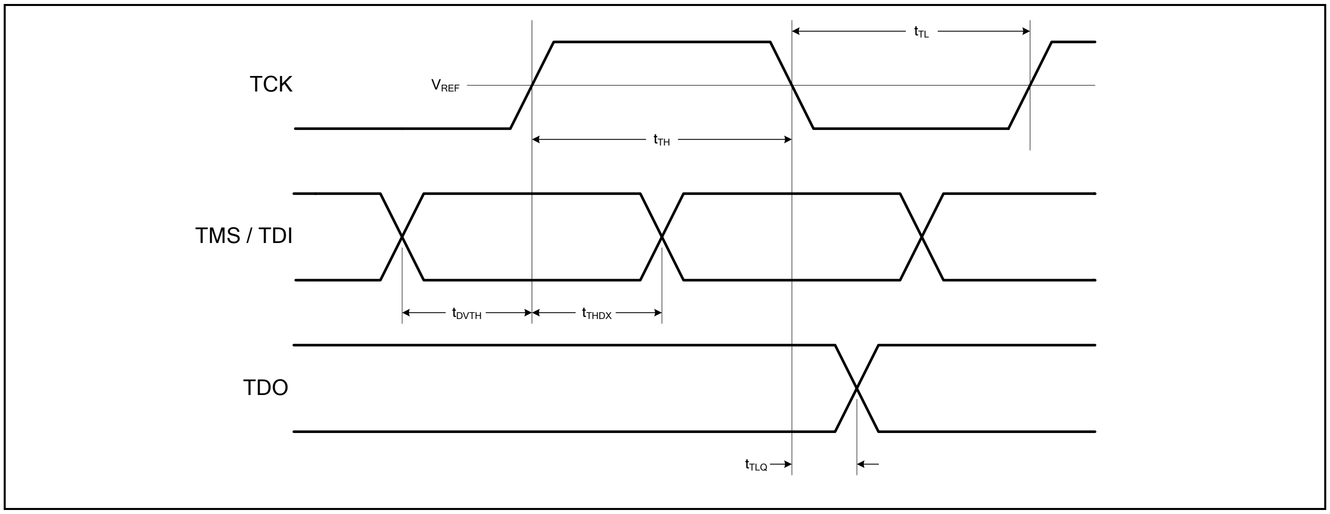


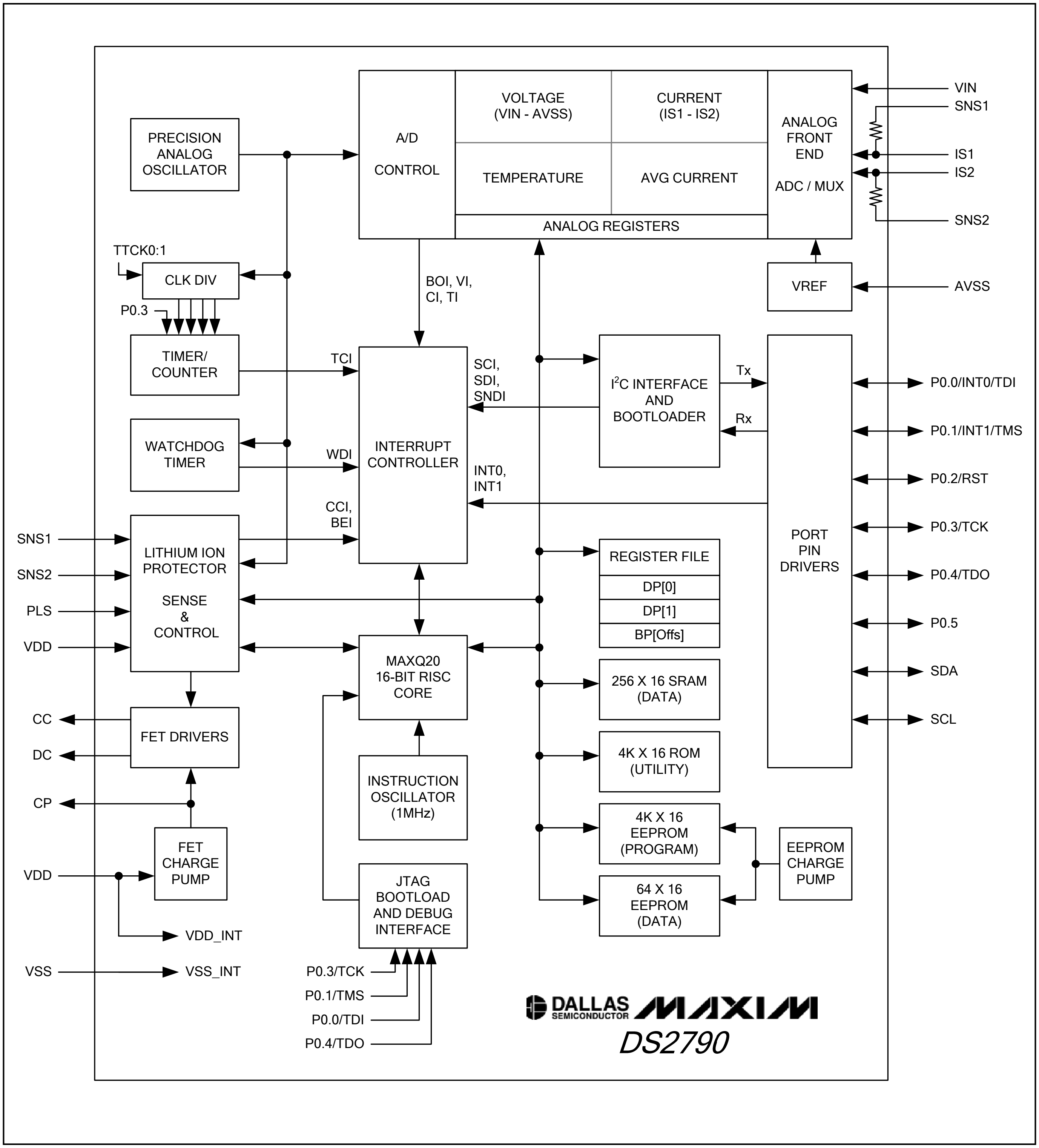
Figure 2. JTAG Timing Diagram



PIN DESCRIPTION

PIN	NAME	DESCRIPTION
1	N.C.	No Connection
2	N.C.	No Connection
3	CP	Charge Pump Output. Bypass CP to V_{SS} with 0.1µF.
4	PLS	Pack Plus. Positive pack terminal connection.
5	DC	Discharge Control. Discharge FET gate drive output.
6	CC	Charge Control. Charge FET gate drive output.
7	SCL	2-wire Serial Interface Clock Input and Output
8	SDA	2-wire Serial Interface Data Input and Output
9	P0.0	Programmable I/O Pin. Alternate functions: external interrupt input INT0, [JTAG TDI].
10	P0.1	Programmable I/O Pin. Alternate functions: External interrupt input INT1, [JTAG TMS].
11	SNS2	Current Sense Input. SNS2 attaches to pack end of current sense resistor.
12	IS2	Current Filter Input 2
13	N.C.	No Connection
14	N.C.	No Connection
15	N.C.	No Connection
16	N.C.	No Connection
17	IS1	Current Filter Input 1
18	SNS1	Current Sense Input. SNS1 attaches to battery end of current sense resistor and V_{SS}.
19	AVSS	Analog Supply Return Node. AVSS attaches to negative battery terminal.
20	V_{SS}	Digital Supply Return Node. V_{SS} attaches to negative battery terminal.
21	P0.2	Programmable I/O Pin. Alternate functions: Reset input pin $\overline{\text{RST}}$.
22	P0.3	Programmable I/O Pin. Alternate functions: Timer/Counter input pin TCK, [JTAG TCK].
23	P0.4	Programmable I/O Pin. Alternate function: [JTAG TDO]
24	P0.5	Programmable I/O Pin
25	V_{DD}	Input Supply: +2.5V to +5.5V input range. Bypass V_{DD} to V_{SS} with 0.1µF.
26	V_{IN}	Battery voltage sense input, measurement relative to AVSS.
27	N.C.	No Connection
28	N.C.	No Connection
—	PAD	Exposed PAD (TDFN only). Not electrically connected to IC. Connect to V_{SS} or leave floating.

FUNCTIONAL DIAGRAM



DETAILED DESCRIPTION

The following is an introduction to the primary features of the DS2790 Programmable 1-Cell Li-Ion Fuel Gauge and Protector. 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.

DS2790 Overview

The DS2790 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, Lithium-Ion protection, and 2-wire communication functions. The MAXQ20 core along with the specialized peripherals provide a flexible solution for fuel gauging and protection of Lithium-Ion battery packs. Flexibility is further enhanced as the solution allows for upgrading of the program and data EEPROM contents over the 2-wire 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 DS2790 employs a MAXQ20 low-cost, high-performance, CMOS, fully static, 16-bit RISC microcontroller with EEPROM 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 assembler, 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 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. See the *MAXQ Family User's Guide* for complete instruction set information.

Memory Organization

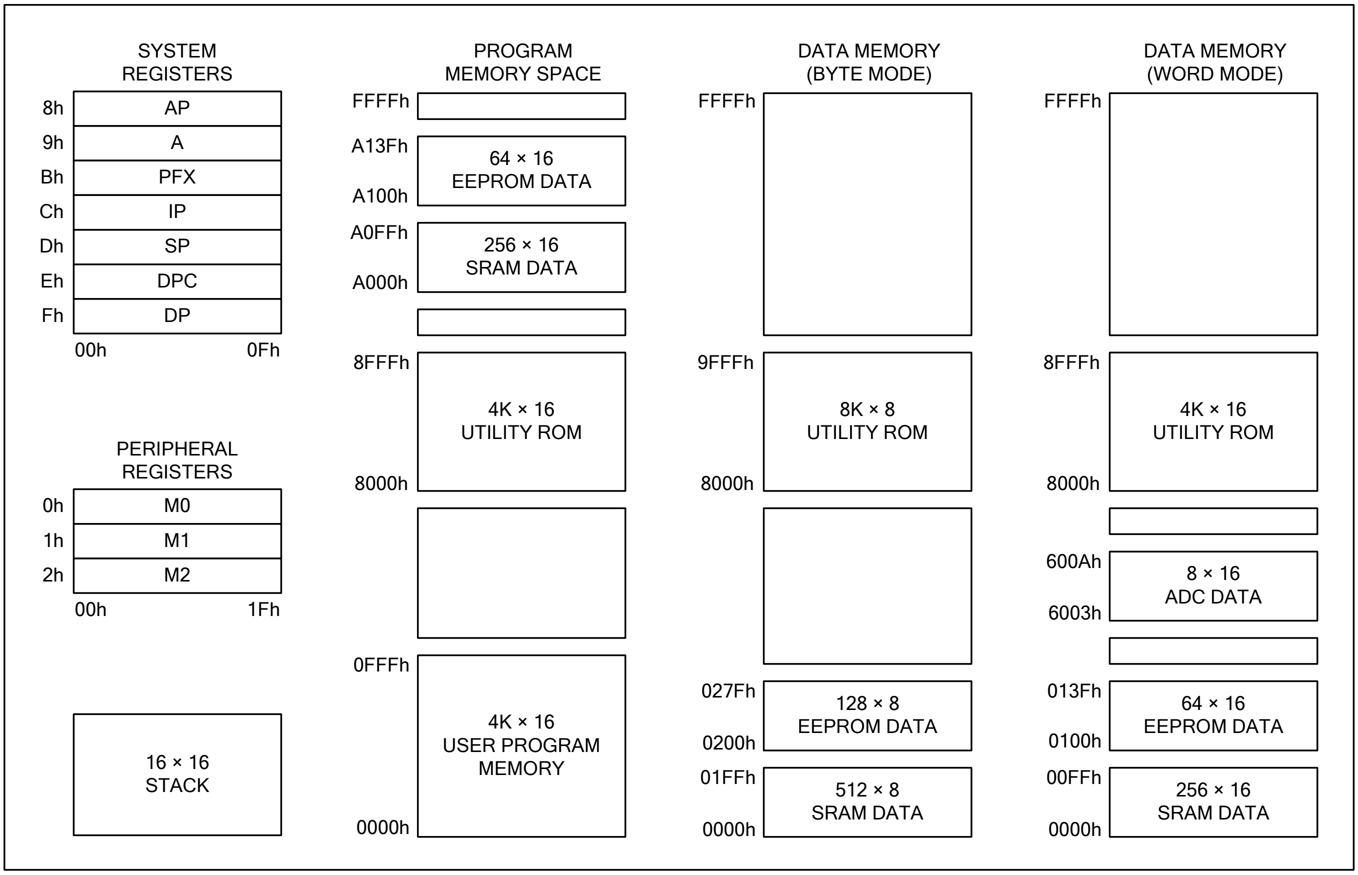
The DS2790 incorporates several memory areas:

- 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
- 8 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 Neumann memory map allows data memory to be mapped into program space, permitting code execution from data memory. In addition program memory may be mapped into data space, permitting code constants to be accessed as data memory. Figure 3 shows the DS2790's memory map when executing from program memory space. See the *MAXQ Family User's Guide: DS2790 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. DS2790 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 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, 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.

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 2-wire 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: DS2790 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 zero’s or all one’s), the PWL bit remains set, which prohibits access to commands to read memory contents over the JTAG and 2-wire 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 2-wire 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, available from Maxim Integrated Products. The 2-wire interface hardware can be an I^2C connection to another microcontroller, or a connection to a PC USB port using a USB to I^2C 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, the bootstrap loader can clear the SPE bit and reset the device, allowing the device to bypass the utility ROM and begin execution of the application software.

Performing a program request over the 2-wire 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 2-wire interface. When programming is complete, the exit loader function is used to reset the DS2790 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

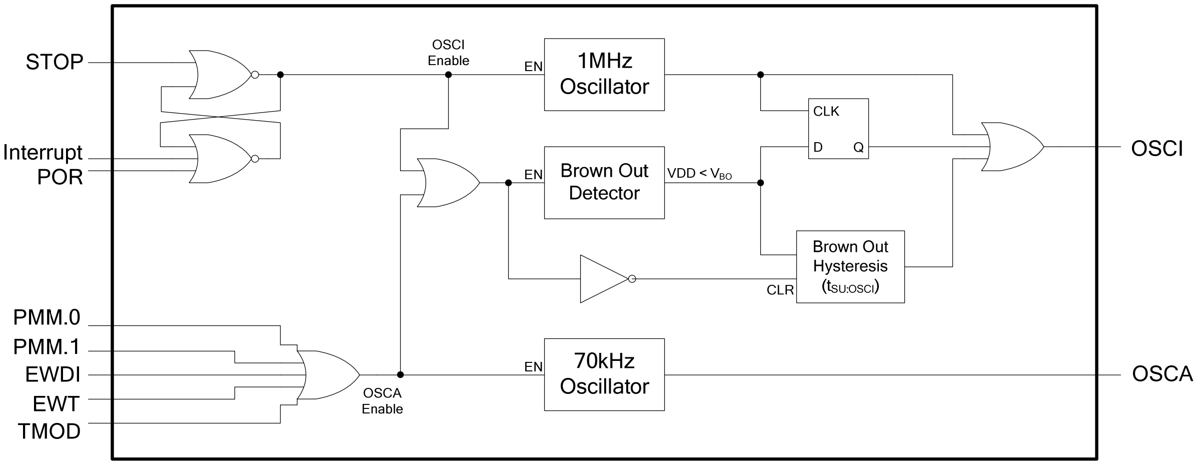
The in-application programming feature allows the microcontroller to modify its own EEPROM program memory. This allows on-the-fly software updates in mission-critical applications that cannot afford downtime. Alternatively, it allows the application to develop custom loader software that can operate 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: DS2790 Supplement*.

SYSTEM TIMING

The DS2790 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 band-gap based RC oscillator that is trimmed to better than 2% accuracy. The analog clock runs independent of OSCI and serves as the clock source for the ADC, watchdog timer, interval timer, and 2-wire timeouts.

OSCI is enabled through either a system interrupt or system POR and disabled through a system STOP. A voltage brown out 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. OSCA is enabled by the Watchdog Timer signals EWDI or EWT, the Timer / Counter (TMOD), or by the protection circuitry (PMM[1:0]).

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 - A few differences exist between the watchdog timer in the DS2790 and the one described in the *MAXQ Family User's Guide* as described in the Watchdog Timer section. 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 $\overline{\text{RST}}$ (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 $\overline{\text{RST}}$ pin is released.

MAXQ20 CORE POWER MANAGEMENT

The DS2790 is designed for low power battery monitoring applications. The peripherals have been designed with the ability to wake the processor from Stop 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 Stop mode. Processing resumes from stop mode via any of the following sources (when enabled):

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

No division of the internal system clock is supported, subsequently the PMME and CD[1:0] bits described in the *MAXQ users guide* are not implemented in the DS2790.

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 DS2790 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.

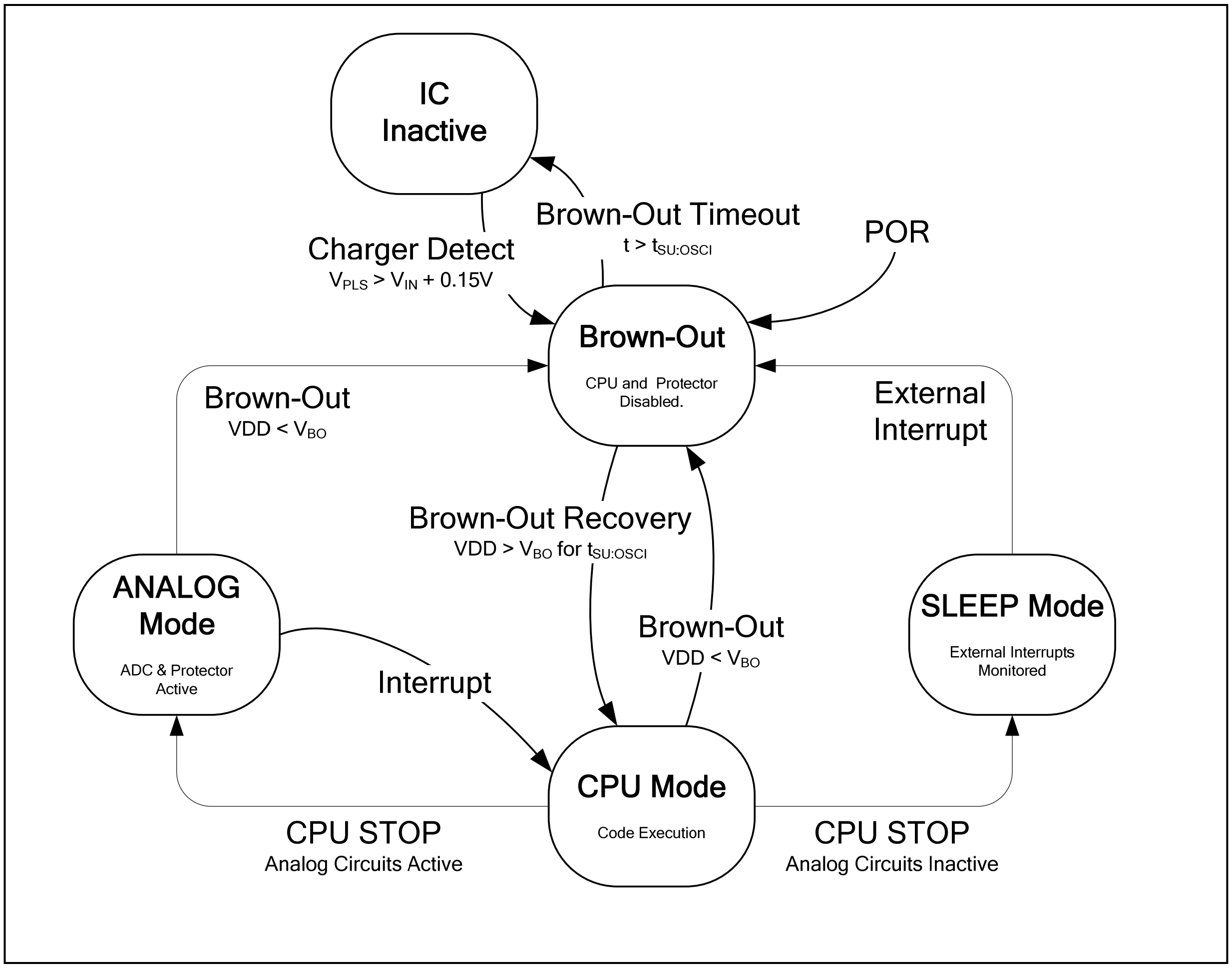
DS2790 POWER MODES

When power is first applied to the DS2790, a Power-on-Reset (POR) circuit transitions the IC to Brown-Out State where cell voltage is monitored. If V_{DD} voltage is above the brown-out threshold V_{BO}, the DS2790 enters CPU state and begins code execution. Firmware determines if the IC switches to ANALOG State or low-power SLEEP States when a STOP halts CPU operation.

The DS2790 enters SLEEP state after a CPU STOP if the ADC, the protector, and all internal timers are disabled. In SLEEP State, all IC operation becomes inactive except for external activity interrupts. Brown-Out detection does not occur in SLEEP State. Any interrupt generated by 2-wire port communication, external input on ports P0.0 or P0.1, or a charger detection on PLS will transition the DS2790 from SLEEP to Brown-Out to verify cell voltage before returning to CPU State. The DS2790 enters ANALOG State after a CPU STOP if any one of the following is active: the ADC, the protector, the interval timer or the watchdog timer. An external interrupt or an interrupt from any active internal circuit causes the DS2790 to transition back to CPU State to service the condition.

If the DS2790 is in ANALOG or CPU State, and V_{DD} falls below V_{BO}, a brown-out condition occurs and the DS2790 enters the Brown-Out State. In Brown-Out 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 DS2790 returns to CPU state and generates a brown-out interrupt (if enabled). Otherwise, if V_{DD} remains below V_{BO} for $t_{SU:OSCI}$, the DS2790 enters an inactive state where it waits for a charger to be applied. When charge voltage is sensed on PLS, the DS2790 returns to the Brown-Out State where V_{DD} voltage is verified before returning to CPU State.

Figure 5. DS2790 State Diagram



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. Table 1 shows the DS2790 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									—	—	—	—	—	—	INS	IGE
IMR									IMS	—	—	—	—	—	IM1	IM0
SC									TAP	—	—	—	—	ROD	PWL	—
IIR									IIS	—	—	—	—	—	II1	II0
CKCN									—	—	—	STOP	—	—	—	—
WDCN									POR	EWDI	WD1	WD0	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									GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0
BP	BP (16 bits)															
GRS	GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0	GR.15	GR.14	GR.13	GR.12	GR.11	GR.10	GR.9	GR.8
GRH									GR.15	GR.14	GR.13	GR.12	GR.11	GR.10	GR.9	GR.8
GRXL	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.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	0	0	0	0	0	0	0
WDCN									s	s	0	0	0	s	s	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	0	0	0	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	**TWSINT**	—	10h	—	—	—
01h	PPU	**TWSIM**	—	11h	—	—	—
02h	PAF	*TWSCMD*	—	12h	—	—	—
03h	**EIC**	**TWSCFG**	—	13h	—	—	—
04h	**EINT**	TWSTXD/RXD	—	14h	—	—	—
05h	**PROT**	—	—	15h	—	—	—
06h	**TC**	—	—	16h	—	—	—
07h	TCC	—	—	17h	—	—	—
08h	*PI*	—	—	18h	**ICDT0**	—	—
09h	—	*TWSFIF*	—	19h	**ICDT1**	—	—
0Ah	—	—	—	1Ah	*ICDC*	—	—
0Bh	—	—	—	1Bh	ICDF	—	—
0Ch	—	—	—	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									SDA_PU	SCL_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	MSNDI	MCCI	MBEI	MVI	MCI	MTI	MTCI	PIP.1	PIP.0	PIT.1	PIT.0	PIE.1	PIE.0
EINT	BOI	SCI	SDI	SNDI	CCI	BEI	VI	CI	TI	TCI	—	—	—	$\overline{\text{RST}}$	INT.1	INT.0
PROT	COCF	DOCF	SCF	OVF	UVF	—	*CC*	*DC*	—	—	—	—	CE	DE	PMM.1	PMM.0
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									*SDA*	*SCL*	*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*
TWSINT	—	—	—	—	TIMEOUT	STOP	RESTART_READ	RESTART_WRITE	START	TXD_BYTE	TXD_EMPTY	TXD_FULL	RXD_CMD	RXD_BYTE	RXD_EMPTY	RXD_FULL
TWSIM	—	—	—	—	TIMEOUT_MASK	STOP_MASK	RESTART_READ_MASK	RESTART_WRITE_MASK	START_MASK	TXD_BYTE_MASK	TXD_EMPTY_MASK	TXD_FULL_MASK	RXD_CMD_MASK	RXD_BYTE_MASK	RXD_EMPTY_MASK	RXD_FULL_MASK
TWSCMD									*TWSCMD.7*	*TWSCMD.6*	*TWSCMD.5*	*TWSCMD.4*	*TWSCMD.3*	*TWSCMD.2*	*TWSCMD.1*	*TWSCMD.0*
TWSCFG	ADDR.6	ADDR.5	ADDR.4	ADDR.3	ADDR.2	ADDR.1	ADDR.0	0	0	0	0	TOUT_LONG	TLS_DIS	TTO_DIS	CMD_HM	CMD_HM_DIS
TWSTXD/RXD									TXD/RXD.7	TXD/RXD.6	TXD/RXD.5	TXD/RXD.4	TXD/RXD.3	TXD/RXD.2	TXD/RXD.1	TXD/RXD.0
TWSFIF									*LTX.3*	*LTX.2*	*LTX.1*	*LTX.0*	*LRX.3*	*LRX.2*	*LRX.1*	*LRX.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
PROT	0	0	0	0	0	0	0	0	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
TWSINT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TWSIM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TWSCMD									0	0	0	0	0	0	0	0
TWSCFG	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
TWSTXD/RXD									0	0	0	0	0	0	0	0
TWSFIF									0	0	0	0	0	0	0	0

Note: s indicates bit reflects pin state.

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 TWSINT Register described in the 2-Wire Interface section, and the EINT Register as shown in Figure 6.

EINT Register

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

Figure 6. 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 all serial lines become high.	Every time all lines are high after any of them were low.
	SDI	MSDI	The **serial disconnect interrupt** is generated when all serial lines are low for at least 220ms.	Once every 220ms if all serial lines are held low. The first interrupt may take up to 440ms from the time all lines go low. Interrupt will not trigger if the ADC is off.
	SNDI	MSNDI	The **serial not disconnected interrupt** is generated when only one serial line goes high.	Every time any line goes high after all of them were low.
	CCI	MCCI	The **charger connection interrupt** is generated when V_{PLS} increases above V_{IN} and creates a charger detection condition.	Each time the charger detection condition evaluates to true after it was false.
Brown-Out Detector	BOI	MBOI	The **brown-out 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 brown-out.
Protection Logic	BEI	MBEI	The **battery event interrupt** is an interrupt for a collection of events that initiate the various battery conditions that are handled by the protection logic: overvoltage, undervoltage, charge overcurrent, discharge overcurrent and short-circuit. The battery conditions are available as flags in the MAS register of module 0.	Each entry into a protection violation.
A/D Converter	VI	MVI	The **voltage interrupt** indicates the voltage register in the data peripheral memory block has a fresh voltage average.	Once every 6.9ms. Never if the ADC is off.
	CI	MCI	The **current interrupt** indicates that the quick average current register 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 DS2790 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	$\overline{\text{RST}}$*		Configurable, [In]	-	Configurable, [Off]	-
P0.3	-	TCK*	Configurable, [In]	-	Configurable, [Off]	-
P0.4	-	TDO*	Configurable, [In]	-	-	Configurable, [Off]
P0.5	-	-	Configurable, [In]	-	-	Configurable, [Off]
SDA	-	-	Yes	Configurable, [On]	Configurable, [Off]	-
SCL	-	-	Yes	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 hi-Z 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.

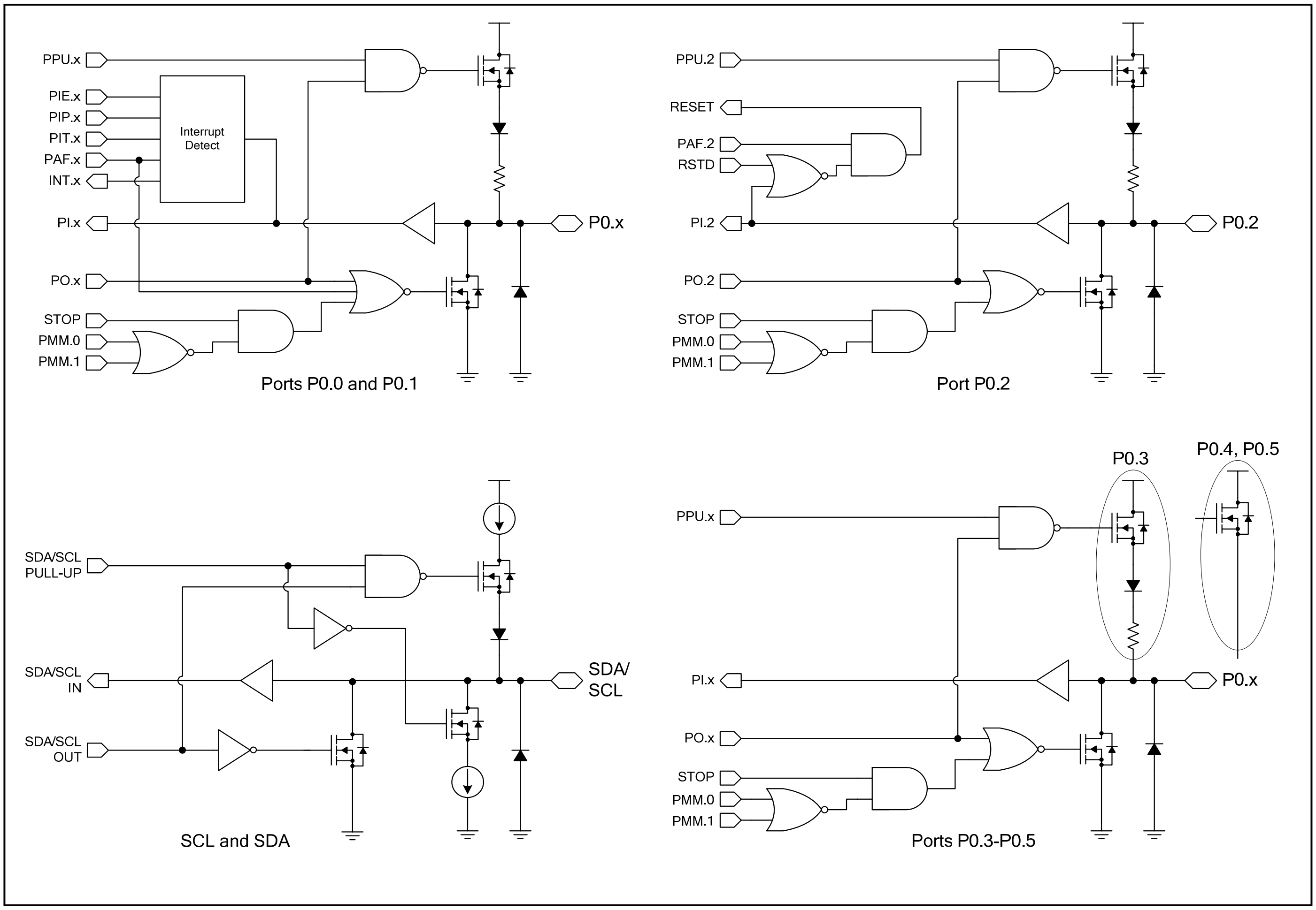
PAF register: The PAF register enables or disables the alternate functions of P0.0-P0.2. When a pin's PAF bit is cleared, the pin is controlled by the PI, PO, PPU, and EIC registers. When the PAF bit is set, the pin operates in its alternate function mode. The $\overline{\text{RST}}$ function of P0.2 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

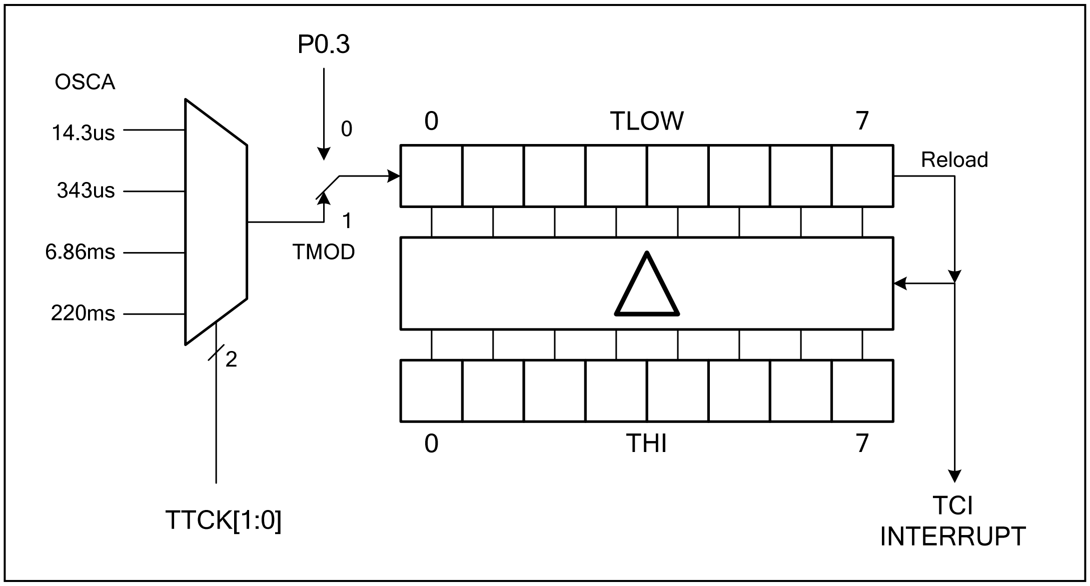
Figure 7. Port Pin Schematics



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 brown-out detectors continue to run if the CPU is stopped.

Figure 8. 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. The following table describes the possible resolution and range of the timer.

Table 9. Programmable Timer Configuration

TMOD	TTCK[1:0]	CLOCK PERIOD	TIMER RANGE (t * 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	

2-WIRE SLAVE PERIPHERAL INTERFACE MODULE

A 2-wire serial-peripheral interface for interconnection with external devices is incorporated into the DS2790. The 2-Wire Slave (TWS) peripheral allows interrupt driven I^2C or SMBus device communication with a minimal amount of CPU overhead. A Transmit/Recieve Data register (TWSTXD/RXD) handles byte level data transfers and the TWS FIFO register (TWSFIF) monitors the usage of the transmit and receive buffers. The 2-Wire Slave Command register (TWSCMD) maintains the command byte of every communication sequence for use by the MAXQ20 core.

Configuration of the 2-wire interface is handled through the TWS Configuration register (TWSCFG) allowing system software to change the DS2790's slave address, control handshaking on the clock line, and control bus timeout settings. The asynchronous interface between the TWS and MAXQ20 core is handled by TWS generated interrupts reported in the Interrupt register (TWSINT) and controlled in Interrupt Mask Register (TWSIM).