Rev 10; 7/08

EVALUATION KIT AVAILABLE

DALLAS SEMICONDUCTOR MAXIM

Low-Power LCD Microcontroller

MAXQ2000

General Description

The MAXQ2000 microcontroller is a low-power, 16-bit device that incorporates a liquid-crystal display (LCD) interface that can drive up to 100 (-RBX/-RBX+) or 132 (-RAX/-RAX+/-RFX/-RFX+) segments. The MAXQ2000 is uniquely suited for the blood-glucose monitoring market, but can be used in any application that requires high performance and low-power operation. The device can operate at a maximum of either 14MHz (V_{DD} > 1.8V) or 20MHz (V_{DD} > 2.25V). The MAXQ2000 has 32kWords of flash memory, 1kWord of RAM, three 16-bit timers, and one or two universal synchronous/asynchronous receiver/transmitters (UARTs). Flash memory aids prototyping and low-volume production. The microcontroller core is powered by a 1.8V supply, with a separate I/O supply for optimum flexibility. An ultra-low-power sleep mode makes these parts ideal for battery-powered, portable equipment.

Applications

Medical Instrumentation
Battery-Powered and Portable Devices
Electrochemical and Optical Sensors
Industrial Control
Data-Acquisition Systems and Data Loggers
Home Appliances
Consumer Electronics
Thermostats/Humidity Sensors
Security Sensors
Gas and Chemical Sensors
HVAC
Smart Transmitters

Typical Operating Circuit, Pin Configurations, and Ordering Information appear at end of data sheet.

Features

- **High-Performance, Low-Power, 16-Bit RISC Core**
  - DC to 20MHz Operation, Approaching 1MIPS per MHz
  - Dual 1.8V Core/3V I/O Enables Low Power/Flexible Interfacing
  - 33 Instructions, Most Single Cycle
  - Three Independent Data Pointers Accelerate Data Movement with Automatic Increment/Decrement
  - 16-Level Hardware Stack
  - 16-Bit Instruction Word, 16-Bit Data Bus
  - 16 x 16-Bit, General-Purpose Working Registers
  - Optimized for C-Compiler (High-Speed/Density Code)
- **Program and Data Memory**
  - 32kWords Flash Memory, Mask ROM for High-Volume Applications
  - 10,000 Flash Write/Erase Cycles
  - 1kWord of Internal Data RAM
  - JTAG/Serial Boot Loader for Programming
- **Peripheral Features**
  - Up to 50 General-Purpose I/O Pins
  - 100/132 Segment LCD Driver
    - Up to 4 COM and 36 Segments
    - Static, 1/2, and 1/3 LCD Bias Supported
    - No External Resistors Required
  - SPI™ and 1-Wire® (-RAX/-RAX+/-RFX/-RFX+ Only) Hardware I/O Ports
  - One or Two Serial UARTs
  - One-Cycle, 16 x 16 Hardware Multiply/Accumulate with 48-Bit Accumulator
  - Three 16-Bit Programmable Timers/Counters
  - 8-Bit, Subsecond, System Timer/Alarm
  - 32-Bit, Binary Real-Time Clock with Time-of-Day Alarm
  - Programmable Watchdog Timer
- **Flexible Programming Interface**
  - Bootloader Simplifies Programming
  - In-System Programming Through JTAG
  - Supports In-Application Programming of Flash Memory
- **Ultra-Low-Power Consumption**
  - 190µA typ at 8MHz Flash Operation, PMM1 at 2.2V
  - 700nA typ in Lowest Power Stop Mode
  - Low-Power 32kHz Mode and Divide-by-256 Mode

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

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

Low-Power LCD Microcontroller

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground Except V_{DD} .....-0.5V to (V_{DDIO} + 0.5)V
Voltage Range on V_{DD} Relative to Ground .....-0.5V to +2.75V
Voltage Range on V_{DDIO} Relative to Ground.....-0.5V to +3.6V

Operating Temperature Range .....-40°C to +85°C
Storage Temperature Range .....-65°C to +150°C
Soldering Temperature .....Refer to the IPC/JEDEC J-STD-020 Specification.

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

ELECTRICAL CHARACTERISTICS

(V_{DD} = $V_{DD(MIN)}$ to $V_{DD(MAX)}$, V_{DDIO} = 2.7V to 3.6V, T_A = -40°C to +85°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Core Supply Voltage	V_{DD}	32k x 16 flash		1.8	2.5	2.75	V
		Flash programming		2.25	2.5	2.75	
I/O Supply Voltage	V_{DDIO}			V_{DD}		3.6	V
V_{DD} Slew Rate		V_{DD} rising (Note 2)		225			mV/ms
Active Current, f_{HFIN} = 14MHz (Note 3)	I_{DD1}	/1 mode			6.0	9.2	mA
	I_{DD2}	/2 mode			5.6	8.6	
	I_{DD3}	/4 mode			3.4	5.1	
	I_{DD4}	/8 mode			1.9	2.9	
	I_{DD5}	PMM1 mode			0.5	0.7	
	I_{DD6}	PMM2 mode; 32KIN = 32.768kHz	Rev A2		4.8	7.6	
			Rev A3		0.1	0.95	
Active Current, f_{HFIN} = 20MHz (Note 3)	I_{DD1}	/1 mode			6.5	10.4	mA
	I_{DD2}	/2 mode			5.9	9.6	
	I_{DD3}	/4 mode			3.8	6.2	
	I_{DD4}	/8 mode			2.2	3.8	
	I_{DD5}	PMM1 mode			0.6	1.4	
	I_{DD6}	PMM2 mode; 32KIN = 32.768kHz	Rev A2		4.8	7.6	
			Rev A3		0.1	0.95	
Active Current		Execution from flash memory, 20MHz, V_{DD} = 2.2V, T_A = +25°C			5.1		mA
		Execution from flash memory, 8MHz, /8 mode, V_{DD} = 2.2V, T_A = +25°C			0.85		
		Execution from flash memory, 8MHz, PMM1 mode, V_{DD} = 2.2V, T_A = +25°C			0.19		
		Execution from RAM, 8MHz, /8 mode, V_{DD} = 2.2V, T_A = +25°C			0.30		
		Execution from RAM, 1MHz, /1 mode, V_{DD} = 2.2V, T_A = +25°C			0.14		
Stop-Mode Current	$I_{STOP(VDD)}$	-40°C < T_A < +25°C			0.7	55	µA
		T_A = +85°C			20	550	
Digital I/O Supply Current	I_{DDIO}	RTC enabled; HFIN ≥ 14MHz; all I/O disconnected			1	50	µA

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = $V_{DD(MIN)}$ to $V_{DD(MAX)}$, V_{DDIO} = 2.7V to 3.6V, T_A = -40°C to +85°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input High Voltage: HFIN and 32KIN	V_{IH1}		0.8 x V_{DDIO}		V_{DDIO}	V
Input High Voltage: P6.4–P6.5 and P7.0–P7.1	V_{IH2}	SVS on, V_{LCD} = 3.3V	0.8 x V_{LCD}		V_{LCD}	V
Input High Voltage: All Other Pins	V_{IH3}		0.8 x V_{DDIO}		V_{DDIO}	V
Input Low Voltage: HFIN and 32KIN	V_{IL1}		0		0.2 x V_{DDIO}	V
Input Low Voltage: All Other Pins	V_{IL2}		0		0.2 x V_{DDIO}	V
Output High Voltage: P6.4–P6.5 and P7.0–P7.1	V_{OH1}	SVS on; $I_{OH(MAX)}$ = 0.75mA; V_{LCD} = 2.7V	V_{LCD} - 0.2			V
Output High Voltage: All Other Pins	V_{OH2}	$I_{OH(MAX)}$ = 0.75mA; V_{DDIO} =1.8V	V_{DDIO} - 0.2			V
Output Low Voltage for All Other Pins	V_{OL1}	I_{OL} = 1.0mA; V_{DDIO} = 1.8V	GND		0.2	V
Output Low Voltage for P6.4–P6.5 and P7.0–P7.1	V_{OL2}	I_{OL} = 1.4mA; V_{DDIO} = 2.7V	GND		0.2	V
Input Leakage Current	I_L	Internal pullup disabled	-100		+100	nA
Input Pullup Current	I_{IP}	Internal pullup enabled	-20		-5	µA
LCD INTERFACE						
LCD Reference Voltage	V_{LCD}		2.7	3.3	3.6	V
LCD Bias Voltage 1	V_{LCD1}	1/3 bias	V_{ADJ} + 2/3 (V_{LCD} - V_{ADJ})			V
LCD Bias Voltage 2	V_{LCD2}	1/3 bias	V_{ADJ} + 1/3 (V_{LCD} - V_{ADJ})			V
LCD Adjustment Voltage	V_{ADJ}	Guaranteed by design	0		0.4 x V_{LCD}	V
LCD Bias Resistor	R_{LCD}			100		kΩ
LCD Adjustment Resistor	R_{LADJ}	LRA4:LRA0 = 11111b		200		kΩ
LCD Segment Voltage	V_{SEGxx}	When segment is driven at V_{LCD} level; V_{LCD} = 3V; I_{SEGxx} = -3µA; guaranteed by design	V_{LCD} - 0.02		V_{LCD}	V
		When segment is driven at V_{LCD1} level; V_{LCD1} = 2V; I_{SEGxx} = -3µA; guaranteed by design	V_{LCD1} - 0.02		V_{LCD1}	
		When segment is driven at V_{LCD2} level; V_{LCD2} = 1V; I_{SEGxx} = -3µA; guaranteed by design	V_{LCD2} - 0.02		V_{LCD2}	
		When segment is driven at V_{ADJ} level; V_{ADJ} = 0V; I_{SEGxx} = 3µA; guaranteed by design	V_{ADJ}		0.1	

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = $V_{DD(MIN)}$ to $V_{DD(MAX)}$, V_{DDIO} = 2.7V to 3.6V, T_A = -40°C to +85°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
EXTERNAL CLOCK SOURCE						
External-Clock Frequency	f_{HFIN}	External oscillator, $V_{DD} \geq 2.25V$	0		20	MHz
		External oscillator, $V_{DD} < 2.25V$	0		14	
		External crystal, $V_{DD} \geq 2.25V$	3		20	
		External crystal, $V_{DD} < 2.25V$	3		14	
		Flash programming, $V_{DD} \geq 2.25V$	2		20	
		Flash programming, $V_{DD} < 2.25V$	2		14	
External-Clock Period	t_{CLCL}	48% minimum duty cycle	50			ns
System-Clock Frequency	f_{CK}	$2.25V \leq V_{DD} \leq 2.75V$	0		20	MHz
		$1.8V \leq V_{DD} \leq 2.75V$	0		14	
System-Clock Period	t_{CK}		50			ns
REAL-TIME CLOCK						
RTC Input Frequency	f_{32KIN}	32kHz watch crystal		32.768		kHz
JTAG/FLASH PROGRAMMING						
Flash Erase Time		Mass erase	200			ms
		Page erase	20			
Flash Programming Time			2.5		5.0	ms
Write/Erase Cycles		T_A = +25°C	10,000			cycles
Data Retention		T_A = +25°C	100			years
SPI TIMING						
SPI Master Operating Frequency	$1/t_{MCK}$				f_{CK} / 2	MHz
SPI Slave Operating Frequency	$1/t_{SCK}$				f_{CK} / 8	MHz
SCLK Output Pulse-Width High/Low	t_{MCH}, t_{MCL}		t_{MCK} / 2 - 25			ns
SCLK Input Pulse-Width High/Low	t_{SCH}, t_{SCL}			t_{SCK} / 2		ns
MOSI Output Hold Time after SCLK Sample Edge	t_{MOH}	C_L = 50pF	t_{MCK} / 2 - 25			ns
MOSI Output Valid to Sample Edge	t_{MOV}		t_{MCK} / 2 - 25			ns
MISO Input Valid to SCLK Sample Edge Rise/Fall Setup	t_{MIS}		30			ns
MISO Input to SCLK Sample Edge Rise/Fall Hold	t_{MIH}		0			ns

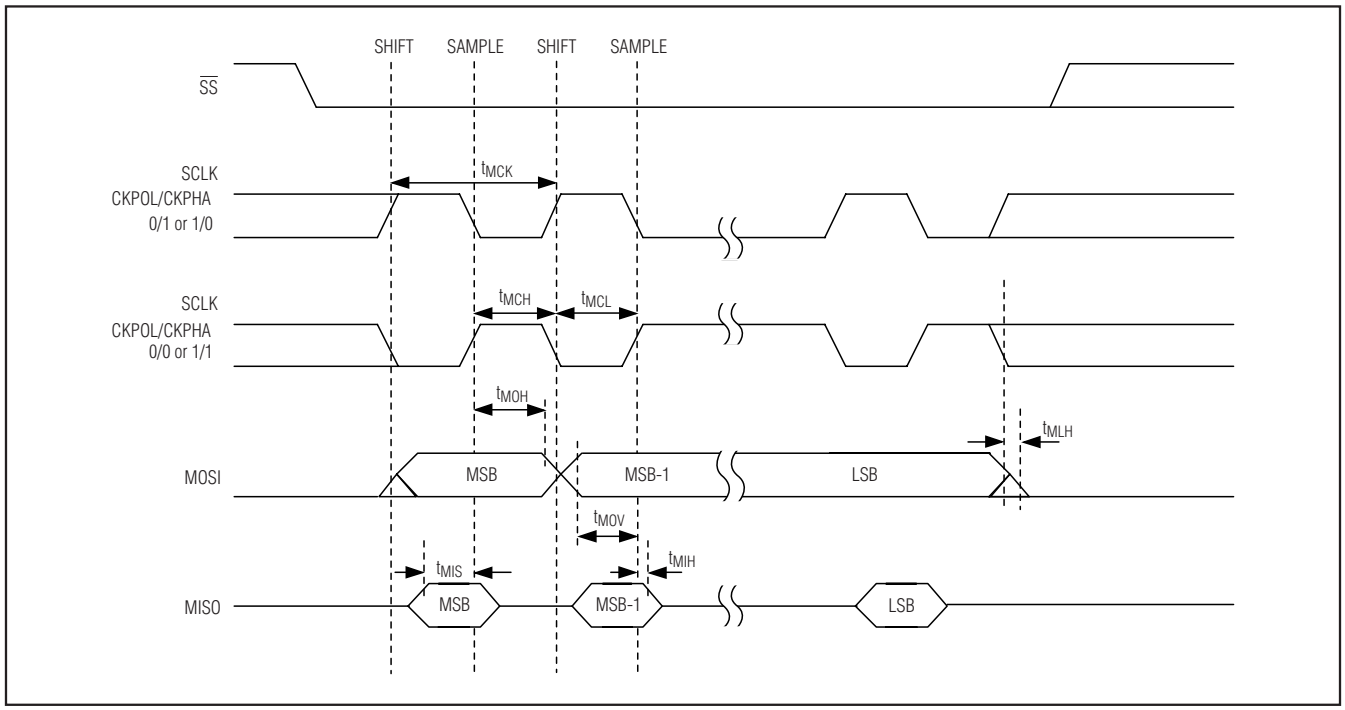
ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = $V_{DD(MIN)}$ to $V_{DD(MAX)}$, V_{DDIO} = 2.7V to 3.6V, T_A = -40°C to +85°C.) (Note 1)

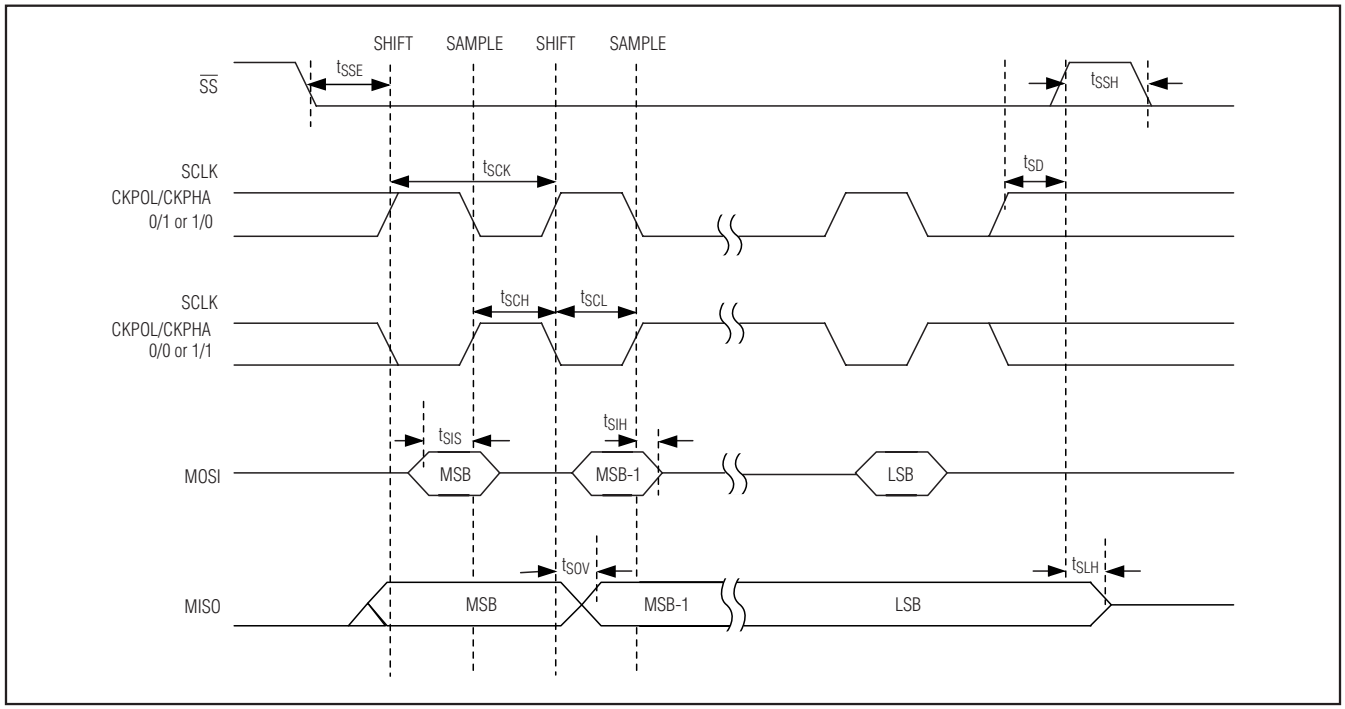
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCLK Inactive to MOSI Inactive	t_{MLH}		t_{MCK} / 2 - 25			ns
MOSI Input to SCLK Sample Edge Rise/Fall Setup	t_{SIS}		30			ns
MOSI Input from SCLK Sample Edge Transition Hold	t_{SIH}		t_{CK} + 25			ns
MISO Output Valid after SCLK Shift Edge Transition	t_{SOV}				$3t_{CK}$ + 25	ns
$\overline{SS}$ Inactive	t_{SSH}		t_{CK} + 25			ns
SCLK Inactive to $\overline{SS}$ Rising	t_{SD}		t_{CK} + 25			ns
MISO Output Disabled after $\overline{CS}$ Edge Rise	t_{SLH}				$2t_{CK}$ + 50	ns
$\overline{SS}$ Active to First Shift Edge	t_{SSE}		$4t_{CK}$			ns

Note 1: Specifications to -40°C are guaranteed by design and not production tested.
Note 2: Guaranteed by design.
Note 3: Measured on the V_{DD} pin with V_{DD} = 2.75V and not in reset.

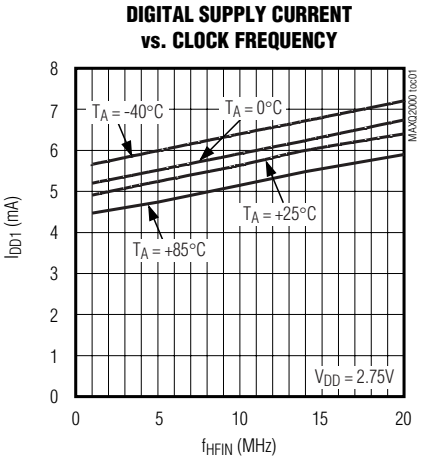
SPI Master Timing



SPI Slave Timing



Typical Operating Characteristics



Pin Description

PIN			NAME	FUNCTION
TQFN-EP	QFN-EP	LQFP		
40	49	70	V_{DD}	**Digital Supply Voltage**
22	27	36, 62	V_{DDIO}	**I/O Supply Voltage**
23, 35	28, 42	39, 63	GND	**Ground**
45	54	83	V_{LCD}	**LCD Bias-Control Voltage.** Highest LCD drive voltage used with static bias. Connected to an external source.
46	55	84	V_{LCD1}	**LCD Bias, Voltage 1.** LCD drive voltage used with 1/2 and 1/3 LCD bias. An internal resistor- divider sets the voltage. External resistors and capacitors can be used to change the LCD voltage or drive capability at this pin.
47	56	85	V_{LCD2}	**LCD Bias, Voltage 2.** LCD drive voltage used with 1/3 LCD bias. An internal resistor-divider sets the voltage. External resistors and capacitors can be used to change LCD voltage or drive capability at this pin.
48	57	86	V_{ADJ}	**LCD Adjustment Voltage.** Connect to an external resistor to provide external control of the LCD contrast. Leave disconnected for internal contrast adjustment.
28	33	50	$\overline{RESET}$	**Digital, Active-Low, Reset Input/Output.** The CPU is held in reset when this is low and begins executing from the reset vector when released. The pin includes pullup current source and should be driven by an open-drain, external source capable of sinking in excess of 2mA. This pin is driven low as an output when an internal reset condition occurs.
42	51	76	HFXIN	**High-Frequency Crystal Input.** Connect an external crystal or resonator between HFXIN and HFXOUT as the high-frequency system clock. Alternatively, HFXIN is the input for an external, high-frequency clock source when HFXOUT is floating.
41	50	71	HFXOUT	**High-Frequency Crystal Output/Input.** Connect an external crystal or resonator between HFXIN and HFXOUT as the high-frequency system clock. Alternatively, float HFXOUT when an external, high-frequency clock source is connected to the HFXIN pin.

Low-Power LCD Microcontroller

Pin Description (continued)

<table>
<tr><th colspan="3">PIN</th><th rowspan="2">NAME</th><th rowspan="2">FUNCTION</th></tr>
<tr><th>TQFN-EP</th><th>QFN-EP</th><th>LQFP</th></tr>
<tr><td>29</td><td>34</td><td>52</td><td>32KIN</td><td>32kHz Crystal Input. Connect an external, 32kHz watch crystal between 32KIN and 32KOUT as the low-frequency system clock. Alternatively, 32KIN is the input for an external, 32kHz clock source when 32KOUT is floating.</td></tr>
<tr><td>30</td><td>35</td><td>53</td><td>32KOUT</td><td>32kHz Crystal Output/Input. Connect an external, 32kHz watch crystal between 32KIN and 32KOUT as the low-frequency system clock. Alternatively, float 32KOUT when an external, 32kHz clock source is connected to the 32KIN pin.</td></tr>
<tr><td>1–8</td><td>66, 67, 68; 1–5</td><td>97, 98, 3–8</td><td>P1.0–P1.7; SEG8–SEG15</td><td>General-Purpose, 8-Bit, Digital, I/O, Type-C Port; LCD Segment-Driver Output. These port pins function as both bidirectional I/O pins and LCD segment-drive outputs. All port pins are defaulted as input with weak pullup after a reset. Enabling a pin’s LCD function disables the general-purpose I/O on the pin. Setting the PCF1 bit enables the LCD for all pins on this port and disables the general-purpose I/O function on all pins.
<table>
<tr><th>56-PIN</th><th>68-PIN</th><th>100-PIN</th><th>PORT</th><th>ALTERNATE FUNCTION</th></tr>
<tr><td>1</td><td>66</td><td>97</td><td>P1.0</td><td>SEG8</td></tr>
<tr><td>2</td><td>67</td><td>98</td><td>P1.1</td><td>SEG9</td></tr>
<tr><td>3</td><td>68</td><td>3</td><td>P1.2</td><td>SEG10</td></tr>
<tr><td>4</td><td>1</td><td>4</td><td>P1.3</td><td>SEG11</td></tr>
<tr><td>5</td><td>2</td><td>5</td><td>P1.4</td><td>SEG12</td></tr>
<tr><td>6</td><td>3</td><td>6</td><td>P1.5</td><td>SEG13</td></tr>
<tr><td>7</td><td>4</td><td>7</td><td>P1.6</td><td>SEG14</td></tr>
<tr><td>8</td><td>5</td><td>8</td><td>P1.7</td><td>SEG15</td></tr>
</table></td></tr>
<tr><td>9–12</td><td>6–13</td><td>9, 10, 11, 14–18</td><td>P2.0–P2.7; SEG16–SEG23</td><td>General-Purpose, 8-Bit, Digital, I/O, Type-C Port; LCD Segment-Driver Output. These port pins function as both bidirectional I/O pins and LCD segment-drive outputs. All port pins are defaulted as input with weak pullup after a reset. Enabling a pin’s LCD function disables the general-purpose I/O on the pin. Setting the PCF2 bit enables the LCD for all pins on this port and disables the general-purpose I/O function on all pins.
<table>
<tr><th rowspan="2">56-PIN</th><th rowspan="2">68-PIN</th><th rowspan="2">100-PIN</th><th rowspan="2">PORT</th><th colspan="2">ALTERNATE FUNCTIONS</th></tr>
<tr><th>56-PIN</th><th>68-PIN</th></tr>
<tr><td>—</td><td>6</td><td>9</td><td>P2.0</td><td>—</td><td>SEG16</td></tr>
<tr><td>—</td><td>7</td><td>10</td><td>P2.1</td><td>—</td><td>SEG17</td></tr>
<tr><td>—</td><td>8</td><td>11</td><td>P2.2</td><td>—</td><td>SEG18</td></tr>
<tr><td>—</td><td>9</td><td>14</td><td>P2.3</td><td>—</td><td>SEG19</td></tr>
<tr><td>9</td><td>10</td><td>15</td><td>P2.4</td><td>SEG16</td><td>SEG20</td></tr>
<tr><td>10</td><td>11</td><td>16</td><td>P2.5</td><td>SEG17</td><td>SEG21</td></tr>
<tr><td>11</td><td>12</td><td>17</td><td>P2.6</td><td>SEG18</td><td>SEG22</td></tr>
<tr><td>12</td><td>13</td><td>18</td><td>P2.7</td><td>SEG19</td><td>SEG23</td></tr>
</table></td></tr>
</table>

Pin Description (continued)

<table>
<tr><th colspan="3">PIN</th><th rowspan="2">NAME</th><th rowspan="2">FUNCTION</th></tr>
<tr><th>TQFN-EP</th><th>QFN-EP</th><th>LQFP</th></tr>
<tr><td>13–16</td><td>14–21</td><td>19–23, 27, 28, 29</td><td>P3.0–P3.7; SEGx; INT4–INT7</td><td>General-Purpose, Digital, I/O, Type-D Port; LCD Segment-Driver Output; External Edge-Selectable Interrupt. This port functions as both bidirectional I/O pins and LCD segment-drive outputs. All port pins are defaulted as inputs with weak pullups after a reset. The port pads can be configured as an external interrupt for pins 7 to 4. If the external interrupt is enabled, the LCD function on the associated pin is disabled. Setting the PCF3 bit enables the LCD for all pins on this port and disables the general-purpose I/O function on all pins.

It is possible to mix the LCD and interrupt functions on the same port. To do this, the interrupt enable must be established prior to setting the PCF0 bit. Care must be taken not to enable the external interrupt while the LCD is in normal operational mode, as this could result in potentially harmful contention between the LCD controller output and the external source connected to the interrupt input.
<table>
<tr><th rowspan="2">56-PIN</th><th rowspan="2">68-PIN</th><th rowspan="2">100-PIN</th><th rowspan="2">PORT</th><th colspan="2">ALTERNATE FUNCTIONS</th></tr>
<tr><th>56-PIN</th><th>68-PIN</th></tr>
<tr><td>—</td><td>14</td><td>19</td><td>P3.0</td><td>—</td><td>SEG24</td></tr>
<tr><td>—</td><td>15</td><td>20</td><td>P3.1</td><td>—</td><td>SEG25</td></tr>
<tr><td>—</td><td>16</td><td>21</td><td>P3.2</td><td>—</td><td>SEG26</td></tr>
<tr><td>—</td><td>17</td><td>22</td><td>P3.3</td><td>—</td><td>SEG27</td></tr>
<tr><td>13</td><td>18</td><td>23</td><td>P3.4</td><td>SEG20/INT4</td><td>SEG28/INT4</td></tr>
<tr><td>14</td><td>19</td><td>27</td><td>P3.5</td><td>SEG21/INT5</td><td>SEG29/INT5</td></tr>
<tr><td>15</td><td>20</td><td>28</td><td>P3.6</td><td>SEG22/INT6</td><td>SEG30/INT6</td></tr>
<tr><td>16</td><td>21</td><td>29</td><td>P3.7</td><td>SEG23/INT7</td><td>SEG31/INT7</td></tr>
</table></td></tr>
<tr><td>17–21</td><td>22–26</td><td>30–34</td><td>SEGx; COM3–COM0</td><td>LCD Segment-Driver Output; LCD Common-Drive Output. The selection of a pin function as either segment or its alternative common-mode signal is controlled by the choice of duty cycle (DUTY1:0).
<table>
<tr><th rowspan="2">56-PIN</th><th rowspan="2">68-PIN</th><th rowspan="2">100-PIN</th><th colspan="3">FUNCTION</th><th rowspan="2">ALTERNATE FUNCTIONS</th></tr>
<tr><th>56-PIN</th><th>68-PIN</th><th>100-PIN</th></tr>
<tr><td>17</td><td>22</td><td>30</td><td>SEG24</td><td>SEG32</td><td>SEG32</td><td>—</td></tr>
<tr><td>18</td><td>23</td><td>31</td><td>SEG25</td><td>SEG33</td><td>COM3</td><td>COM3</td></tr>
<tr><td>19</td><td>24</td><td>32</td><td>SEG26</td><td>SEG34</td><td>COM2</td><td>COM2</td></tr>
<tr><td>20</td><td>25</td><td>33</td><td>SEG27</td><td>SEG35</td><td>COM1</td><td>COM1</td></tr>
<tr><td>21</td><td>26</td><td>34</td><td>—</td><td>COM0</td><td>COM0</td><td>—</td></tr>
</table></td></tr>
<tr><td>24–27</td><td>29–32</td><td>40–43</td><td>P4.0–P4.3; TCK/TDI/TMS/TDO; INT8, INT9</td><td>General-Purpose, Digital, I/O, Type-D Port; Debug Port Signal; External Edge-Selectable Interrupt. Pins default to JTAG on POR; other functions must be enabled from software.
<table>
<tr><th>56-PIN</th><th>68-PIN</th><th>100-PIN</th><th>PORT</th><th colspan="2">ALTERNATE FUNCTIONS</th></tr>
<tr><td>24</td><td>29</td><td>40</td><td>P4.0</td><td>TCK</td><td>INT8</td></tr>
<tr><td>25</td><td>30</td><td>41</td><td>P4.1</td><td>TDI</td><td>INT9</td></tr>
<tr><td>26</td><td>31</td><td>42</td><td>P4.2</td><td>TMS</td><td>—</td></tr>
<tr><td>27</td><td>32</td><td>43</td><td>P4.3</td><td>TDO</td><td>—</td></tr>
</table></td></tr>
</table>

Low-Power LCD Microcontroller

Pin Description (continued)

PIN			NAME	FUNCTION
TQFN-EP	QFN-EP	LQFP		
—	36	54	P5.2/RX1/ INT10	**General-Purpose, Digital, I/O, Type-D Port; Serial Port 1 Receive; External Edge-Selectable Interrupt 10**
—	37	56	P5.3/TX1/ INT11	**General-Purpose, Digital, I/O, Type-D Port; Serial Port 1 Transmit; External Edge-Selectable Interrupt 11**
31	38	57	P5.4/$\overline{SS}$	**General-Purpose, Digital, I/O, Type-C Port; Active-Low, SPI, Slave-Select Input.** Becomes the slave-select input in SPI mode.
32	39	58	P5.5; MOSI	**General-Purpose, Digital, I/O, Type-C Port; SPI, Master-Out Slave-In Output.** Data is clocked out of the microcontroller on SCLK's falling edge and into the slave device on SCLK's rising edge. Becomes MOSI input in SPI mode.
33	40	59	P5.6; SCLK	**General-Purpose, Digital, I/O, Type-C Port; SPI, Clock Output.** Becomes SCLK input in slave mode but limited to SYSCLK / 8.
34	41	60	P5.7/ MISO	**General-Purpose, Digital, I/O, Type-C Port; SPI, Master-In Slave-Out Input.** Data is clocked out of the slave on SCLK's falling edge and into the microcontroller on SCLK's rising edge. Becomes MISO output in slave mode.
36	43	64	P6.0/T1B/ INT12	**General-Purpose, Digital, I/O, Type-D Port; Timer 1 Alternative Output (PWM); External Edge-Selectable Interrupt 12**
37	44	65	P6.1/T1/ INT13	**General-Purpose, Digital, I/O Type-D Port; Timer 1 Output (PWM); External Edge-Selectable Interrupt 13**
—	45	66	P6.2/T2B/ OW_OUT	**General-Purpose, Digital, I/O, Type-D Port; Timer 2 Alternative Output (PWM); 1-Wire Data Output**
—	46	67	P6.3/T2/ OW_IN	**General-Purpose, Digital, I/O, Type-D Port; Timer 2 Output (PWM); 1-Wire Data Input**
38	47	68	P6.4/T0B/ WKOUT0	**General-Purpose, Digital, I/O, Type-C Port; Timer 0 Alternative Output (PWM); Wakeup Output 0**
39	48	69	P6.5/T0/ WKOUT1	**General-Purpose, Digital, I/O, Type-C Port; Timer 0 Output (PWM); Wakeup Output 1**
43	52	81	P7.0/TX0/ INT14	**General-Purpose, Digital, I/O, Type-D Port; Serial Port 0 Transmit; External, Edge-Selectable Interrupt 14**
44	53	82	P7.1/RX0/ INT15	**General-Purpose, Digital, I/O, Type-D Port; Serial Port 0 Receive; External Edge-Selectable Interrupt 15**

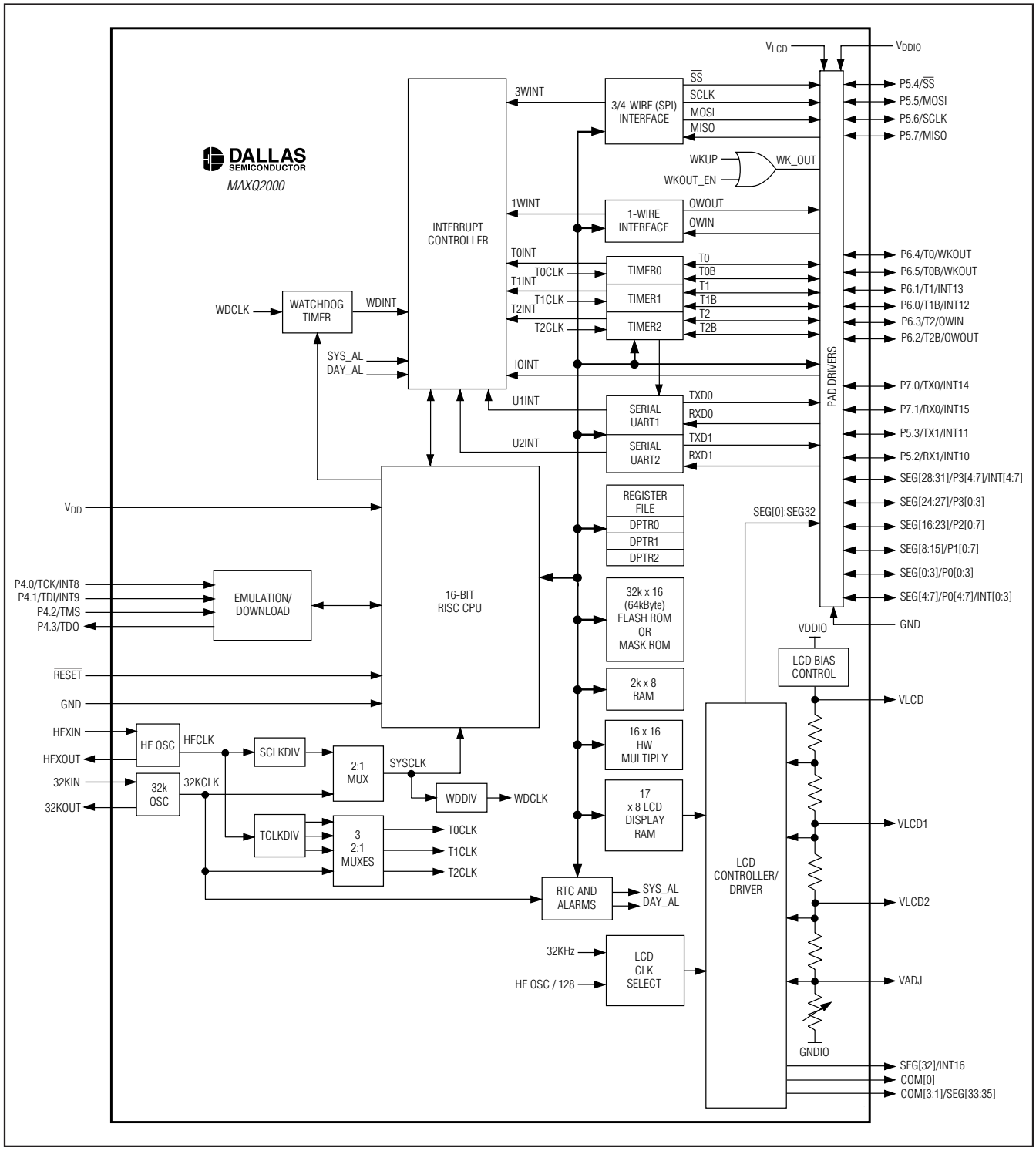
Pin Description (continued)

<table>
<tr><th colspan="3">PIN</th><th rowspan="2">NAME</th><th rowspan="2">FUNCTION</th></tr>
<tr><th>TQFN-EP</th><th>QFN-EP</th><th>LQFP</th></tr>
<tr><td>49–56</td><td>58–65</td><td>89–96</td><td>P0.0–P0.7; SEG0–SEG7; INT0–INT3</td><td>General-Purpose, Digital, I/O, Type-D Port; LCD Segment-Driver Output; External Edge-Selectable Interrupt. This port functions as both bidirectional I/O pins and LCD segment-drive outputs. All port pins are defaulted as input with weak pullup after a reset. The port pads can be configured as an external interrupt for pins 7 to 4. If the external interrupt is enabled, the LCD function on the associated pin is disabled. Setting the PCF0 bit enables the LCD for all pins on this port and disables the general-purpose I/O function on all pins.

It is possible to mix the LCD and interrupt functions on the same port. To do this, the interrupt enable must be established prior to setting the PCF0 bit. Care must be taken not to enable the external interrupt while the LCD is in normal operational mode, as this could result in potentially harmful contention between the LCD controller output and the external source connected to the interrupt input.
<table>
<tr><th>56-PIN</th><th>68-PIN</th><th>100-PIN</th><th>PORT</th><th colspan="2">ALTERNATE FUNCTIONS</th></tr>
<tr><td>49</td><td>58</td><td>89</td><td>P0.0</td><td>SEG0</td><td>—</td></tr>
<tr><td>50</td><td>59</td><td>90</td><td>P0.1</td><td>SEG1</td><td>—</td></tr>
<tr><td>51</td><td>60</td><td>91</td><td>P0.2</td><td>SEG2</td><td>—</td></tr>
<tr><td>52</td><td>61</td><td>92</td><td>P0.3</td><td>SEG3</td><td>—</td></tr>
<tr><td>53</td><td>62</td><td>93</td><td>P0.4</td><td>SEG4</td><td>INT0</td></tr>
<tr><td>54</td><td>63</td><td>94</td><td>P0.5</td><td>SEG5</td><td>INT1</td></tr>
<tr><td>55</td><td>64</td><td>95</td><td>P0.6</td><td>SEG6</td><td>INT2</td></tr>
<tr><td>56</td><td>65</td><td>96</td><td>P0.7</td><td>SEG7</td><td>INT3</td></tr>
</table>
</td></tr>
<tr><td>—</td><td>—</td><td>1, 2, 12, 13, 24, 25, 26, 35, 37, 38, 44–49, 51, 55, 61, 72–75, 77–80, 87, 88, 99, 100</td><td>N.C.</td><td>No Connection. These pins should not be connected.</td></tr>
<tr><td>—</td><td>—</td><td>—</td><td>EP</td><td>Exposed Paddle. Exposed paddle is on the under side of the package. It should be left unconnected.</td></tr>
</table>

Low-Power LCD Microcontroller

Block Diagram



Detailed Description

The following is an introduction to the primary features of the microcontroller. More detailed descriptions of the device features can be found in the data sheets, errata sheets, and user's guides described later in the *Additional Documentation* section.

MAXQ Core Architecture

The MAXQ2000 is a low-cost, high-performance, CMOS, fully static, 16-bit RISC microcontroller with flash memory and an integrated 100- or 132-segment LCD controller. It is structured on a highly advanced, accumulator-based, 16-bit RISC architecture. Fetch and execution operations are completed in one cycle without pipelining, because the instruction contains both the op code and data. The result is a streamlined 20 million instructions-per-second (MIPS) microcontroller.

The highly efficient core is supported by 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. As a result, application speed is greatly increased.

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 the higher-level op codes defined by the assembler, such as ADDC, OR, JUMP, etc. The op codes are actually 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.

Memory Organization

The device incorporates several memory areas:

- 4kB utility ROM,
- 32kWords of flash memory for program storage,
- 1kWord of SRAM for storage of temporary variables, and
- 16-level stack memory for storage of program return addresses and general-purpose use.

The memory is arranged by default in a Harvard architecture, with separate address spaces for program and data memory. A special mode allows data memory to be mapped into program space, permitting code execution from data memory. In addition, another mode allows program memory to be mapped into data space, permitting code constants to be accessed as data memory.

The incorporation of flash memory allows the devices to be reprogrammed, eliminating the expense of throwing away one-time programmable devices during development and field upgrades. Flash memory can be password protected with a 16-word key, denying access to program memory by unauthorized individuals.

A pseudo-Von Neumann memory map can also be enabled. This places the utility ROM, code, and data memory into a single contiguous memory map. This is useful for applications that require dynamic program modification or unique memory configurations.

Stack Memory

A 16-bit-wide 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.

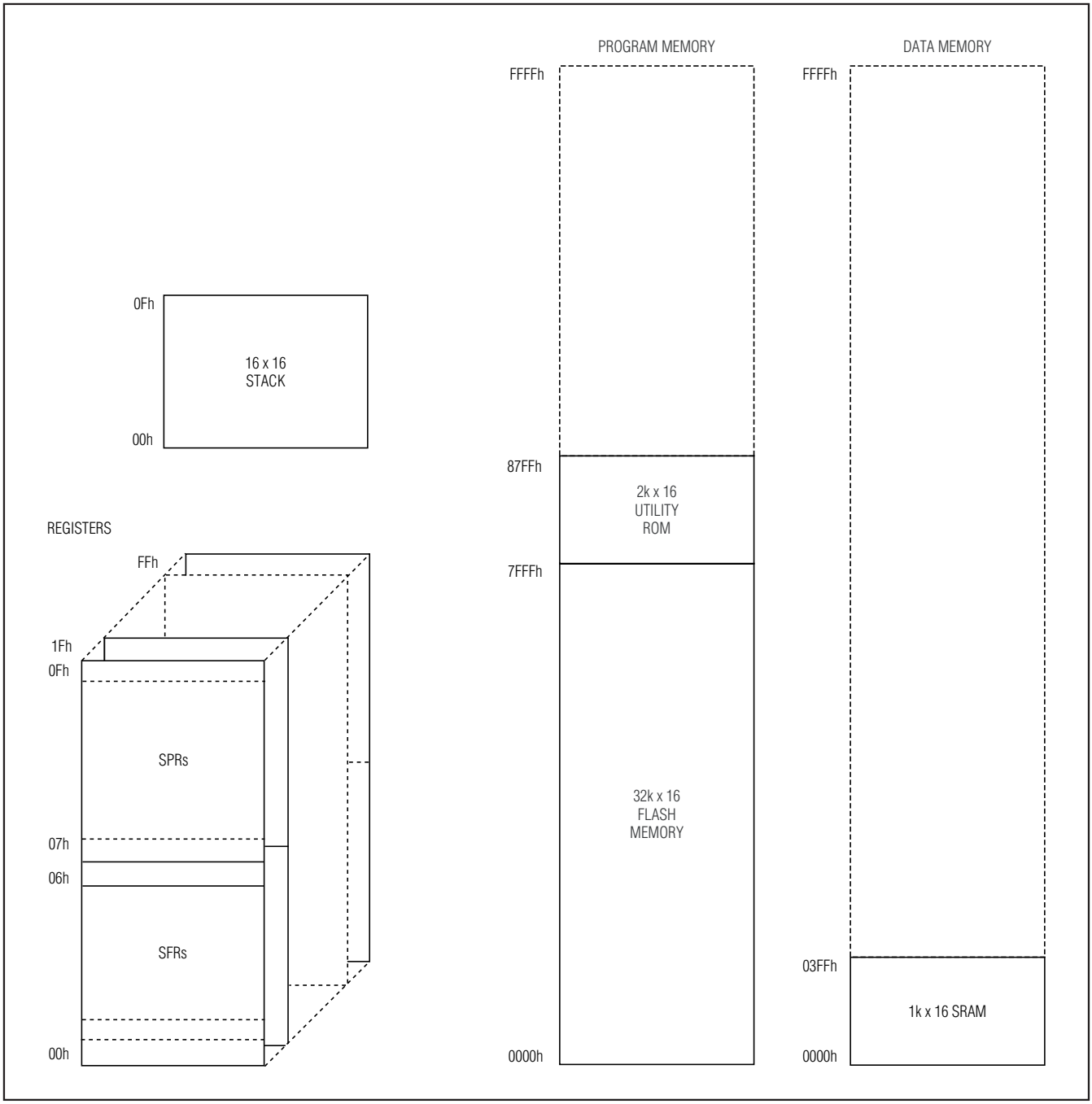


Figure 1. Memory Map

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 @SP and then decrement SP.

Utility ROM

The utility ROM is a 4kB 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 UART interfaces
- In-circuit debug routines
- Test routines (internal memory tests, memory loader, etc.)
- User-callable routines for in-application flash programming and fast table lookup

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 user-application code, or to one of the special routines mentioned. Routines within the utility ROM are user-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: MAXQ2000 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.

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 flash 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 reloaded over a simple JTAG interface. As a result, software can be upgraded in-system, eliminating the need for a costly hardware retrofit when updates are required. Remote software uploads are possible that enable physically inaccessible applications to be frequently updated. The 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. If in-system programmability is not required, a commercial gang programmer can be used for mass programming.

Activating the JTAG interface and loading the test access port (TAP) with the system programming instruction invokes the bootstrap loader. 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.

The following bootstrap loader functions are supported:

- Load
- Dump
- CRC
- Verify
- Erase

Optionally, the bootstrap loader can be invoked by the application code. In this mode, the application software would configure the SPE and PSS bits for UART communication, then jump to the start of the utility ROM. In this way, the bootstrap loader can be accessed through another UART-enabled peripheral, or a PC serial port through an RS-232 transceiver such as the MAX232. Because the bootstrap loader defaults to the JTAG configuration on reset, the UART versus JTAG selection must be made from the application code. As a result, bootstrap loader access through the UART is not possible in an unprogrammed device.

Low-Power LCD Microcontroller

In-Application Programming

The in-application programming feature allows the microcontroller to modify its own flash program memory while simultaneously executing its application software. 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 user-accessible flash programming functions that erase and program flash memory. These functions are described in detail in the user's guide supplement for this device.

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 MAXQ 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 and 4 show the MAXQ2000 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)**
0xh	AP	**A[0]**	**PFX**	**IP**	—	—	—
1xh	APC	**A[1]**	—	—	**SP**	—	—
2xh	—	**A[2]**	—	—	**IV**	—	—
3xh	—	**A[3]**	—	—	—	Offs	**DP0**
4xh	PSF	**A[4]**	—	—	—	**DPC**	—
5xh	IC	**A[5]**	—	—	—	**GR**	—
6xh	IMR	**A[6]**	—	—	**LC0**	GRL	—
7xh	—	**A[7]**	—	—	**LC1**	**BP**	**DP1**
8xh	SC	**A[8]**	—	—	—	***GRS***	—
9xh	—	**A[9]**	—	—	—	GRH	—
Axh	—	**A[10]**	—	—	—	***GRXL***	—
Bxh	*IIR*	**A[11]**	—	—	—	**FP**	—
Cxh	—	**A[12]**	—	—	—	—	—
Dxh	—	**A[13]**	—	—	—	—	—
Exh	CKCN	**A[14]**	—	—	—	—	—
Fxh	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															
	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	—	—	IM4	IM3	IM2	IM1	IM0
SC									TAP	—	—	CDA0	—	ROD	PWL	—
IIR									IIS	—	—	II4	II3	II2	II1	II0
CKCN									—	RGSL	RGMD	STOP	SWB	PMME	CD1	CD0
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	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

Table 4. Peripheral Register Map

REGISTER INDEX	MODULE NAME (BASE SPECIFIER)					
	M0 (x0h)	M1 (x1h)	M2 (x2h)	M3 (x3h)	M4 (x4h)	M5 (x5h)
0xh	PO0	PO4	MCNT	T2CNA0	T2CNA1	—
1xh	PO1	PO5	**MA**	T2H0	T2H1	—
2xh	PO2	PO6	**MB**	T2RH0	T2RH1	—
3xh	PO3	PO7	**MC2**	T2CH0	T2CH1	—
4xh	—	—	**MC1**	—	T2CNA2	—
5xh	—	—	**MC0**	**SPIB**	T2H2	—
6xh	EIF0	EIF1	SCON0	SCON1	T2RH2	—
7xh	EIE0	EIE1	SBUF0	SBUF1	T2CH2	—
8xh	*PI0*	*PI4*	SMD0	SMD1	T2CNB1	—
9xh	*PI1*	*PI5*	**PR0**	**PR1**	**T2V1**	—
Axh	*PI2*	*PI6*	—	—	**T2R1**	—
Bxh	*PI3*	*PI7*	***MC1R***	—	**T2C1**	—
Cxh	EIES0	EIES1	***MC0R***	T2CNB0	T2CNB2	—
Dxh	—	—	**LCRA**	**T2V0**	**T2V2**	—
Exh	—	—	LCFG	**T2R0**	**T2R2**	—
Fxh	—	—	LCD16	**T2C0**	**T2C2**	—
10xh	PD0	PD4	LCD0	T2CFG0	T2CFG1	—
11xh	PD1	PD5	LCD1	—	T2CFG2	—
12xh	PD2	PD6	LCD2	—	—	—
13xh	PD3	PD7	LCD3	OWA	—	—
14xh	—	—	LCD4	OWD	—	—
15xh	—	—	LCD5	SPICN	—	—
16xh	—	—	LCD6	SPICF	—	—
17xh	—	—	LCD7	SPICK	—	—
18xh	—	—	LCD8	**ICDT0**	—	—
19xh	**RCNT**	—	LCD9	**ICDT1**	—	—
1Axh	RTSS	—	LCD10	ICDC	—	—
1Bxh	**RTSH**	—	LCD11	ICDF	—	—
1Cxh	**RTSL**	—	LCD12	ICDB	—	—
1Dxh	RSSA	—	LCD13	**ICDA**	—	—
1Exh	RASH	SVS	LCD14	**ICDD**	—	—
1Fxh	**RASL**	WKO	LCD15	**TM**	—	—

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.*

Table 5. Peripheral Register Bit Functions

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO0									PO0 (8 bits)							
PO1									PO1 (8 bits)							
PO2									PO2 (8 bits)							
PO3									PO3 (8 bits)							
EIF0									IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
EIE0									EX7	EX6	EX5	EX4	EX3	EX2	EX1	EX0
PI0									PI0 (8 bits)							
PI1									PI1 (8 bits)							
PI2									PI2 (8 bits)							
PI3									PI3 (8 bits)							
EIES0									IT7	IT6	IT5	IT4	IT3	IT2	IT1	IT0
PD0									PD0 (8 bits)							
PD1									PD1 (8 bits)							
PD2									PD2 (8 bits)							
PD3									PD3 (8 bits)							
RCNT	WE	X32D	ACS	—	—	—	—	—	ALSF	ALDF	RDYE	RDY	BUSY	ASE	ADE	RTCE
RTSS									RTSS (8 bits)							
RTSH	RTSH (16 bits)															
RTSL	RTSL (16 bits)															
RSSA									RSSA (8 bits)							
RASH									RASH (8 bits)							
RASL	RASL (16 bits)															
PO4									—	—	—	PO4 (5 bits)				
PO5									PO5 (8 bits)							
PO6									PO6 (8 bits)							
PO7									—	—	—	—	—	—	PO7 (2 bits)	
EIF1									IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8
EIE1									EX15	EX14	EX13	EX12	EX11	EX10	EX9	EX8
PI4									—	—	—	PI4 (5 bits)				
PI5									PI5 (8 bits)							
PI6									PI6 (8 bits)							
PI7									—	—	—	—	—	—	PI7 (2 bits)	
EIES1									IT15	IT14	IT13	IT12	IT11	IT10	IT9	IT8
PD4									—	—	—	PD4 (5 bits)				
PD5									PD5 (8 bits)							
PD6									PD6 (8 bits)							

Table 5. Peripheral Register Bit Functions (continued)

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD7									—	—	—	—	—	—	PD7 (2 bits)	
SVS									SV67	SV66	SV65	SV64	—	—	SV71	SV70
WKO									—	—	—	—	—	WKL	WKE1	WKE0
MCNT									OF	MCW	CLD	SQU	OPCS	MSUB	MMAC	SUS
MA	MA (16 bits)															
MB	MB (16 bits)															
MC2	MC2 (16 bits)															
MC1	MC1 (16 bits)															
MC0	MC0 (16 bits)															
SCON0									SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF0									SBUF0 (8 bits)							
SMD0									—	—	—	—	—	ESI0	SMOD0	FEDE0
PR0	PR0 (16 bits)															
MC1R	MC1R (16 bits)															
MC0R	MC0R (16 bits)															
LCRA	—	—	—	DUTY1	DUTY0	FRM3	FRM2	FRM1	FRM0	LCCS	LRIG	LRA4	LRA3	LRA2	LRA1	LRA0
LCFG									PCF3	PCF2	PCF1	PCF0	—	—	OPM	DPE
LCD[0..15]									LCD[0..15] (8 bits)							
T2CNA0									ET2	T2OE0	T2POL0	TR2L	TR2	CPRL2	SS2	G2EN
T2H0									T2V0.15	T2V0.14	T2V0.13	T2V0.12	T2V0.11	T2V0.10	T2V0.9	T2V0.8
T2RH0									T2R0.15	T2R0.14	T2R0.13	T2R0.12	T2R0.11	T2R0.10	T2R0.9	T2R0.8
T2CH0									T2C0.15	T2C0.14	T2C0.13	T2C0.12	T2C0.11	T2C0.10	T2C0.9	T2C0.8
SPIB	SPIB (16 bits)															
SCON1									SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF1									SBUF1 (8 bits)							
SMD1									—	—	—	—	—	ESI1	SMOD1	FEDE1
PR1	PR1 (16 bits)															
T2CNB0									ET2L	T2OE1	T2POL1	TR2L	TF2	TF2L	TCC2	TC2L
T2V0	T2V0.15	T2V0.14	T2V0.13	T2V0.12	T2V0.11	T2V0.10	T2V0.9	T2V0.8	T2V0.7	T2V0.6	T2V0.5	T2V0.4	T2V0.3	T2V0.2	T2V0.1	T2V0.0
T2R0	T2R0.15	T2R0.14	T2R0.13	T2R0.12	T2R0.11	T2R0.10	T2R0.9	T2R0.8	T2R0.7	T2R0.6	T2R0.5	T2R0.4	T2R0.3	T2R0.2	T2R0.1	T2R0.0
T2C0	T2C0.15	T2C0.14	T2C0.13	T2C0.12	T2C0.11	T2C0.10	T2C0.9	T2C0.8	T2C0.7	T2C0.6	T2C0.5	T2C0.4	T2C0.3	T2C0.2	T2C0.1	T2C0.0
T2CFG0									T2CI	DIV2	DIV1	DIV0	T2MD	CCF1	CCF0	C/T2
OWA									—	—	—	—	—	A2	A1	A0
OWD									OWD (8 bits)							
SPICN									STBY	SPIC	ROVR	WCOL	MODF	MODFE	MSTM	SPIEN

Table 5. Peripheral Register Bit Functions (continued)

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPICF									ESPI1	—	—	—	—	CHR	CKPHA	CKPOL
SPICK									CKR7	CKR6	CKR5	CKR4	CKR3	CKR2	CKR1	CKR0
ICDC									DME	—	REGE	—	CMD3	CMD2	CMD1	CMD0
ICDF									—	—	—	—	PSS1	PSS0	SPE	TXC
ICDB									ICDB (8 bits)							
ICDA	ICDA (16 bits)															
ICDD	ICDD (16 bits)															
T2CNA1									ET2	T2OE0	T2POL0	TR2L	TR2	CPRL2	SS2	G2EN
T2H1									T2V1.15	T2V1.14	T2V1.13	T2V1.12	T2V1.11	T2V1.10	T2V1.9	T2V1.8
T2RH1									T2R1.15	T2R1.14	T2R1.13	T2R1.12	T2R1.11	T2R1.10	T2R1.9	T2R1.8
T2CH1									T2C1.15	T2C1.14	T2C1.13	T2C1.12	T2C1.11	T2C1.10	T2C1.9	T2C1.8
T2CNA2									ET2	T2OE0	T2POL0	TR2L	TR2	CPRL2	SS2	G2EN
T2H2									T2V2.15	T2V2.14	T2V2.13	T2V2.12	T2V2.11	T2V2.10	T2V2.9	T2V2.8
T2RH2									T2R2.15	T2R2.14	T2R2.13	T2R2.12	T2R2.11	T2R2.10	T2R2.9	T2R2.8
T2CH2									T2C2.15	T2C2.14	T2C2.13	T2C2.12	T2C2.11	T2C2.10	T2C2.9	T2C2.8
T2CNB1									ET2L	T2OE1	T2POL1	TR2L	TF2	TF2L	TCC2	TC2L
T2V1	T2V1.15	T2V1.14	T2V1.13	T2V1.12	T2V1.11	T2V1.10	T2V1.9	T2V1.8	T2V1.7	T2V1.6	T2V1.5	T2V1.4	T2V1.3	T2V1.2	T2V1.1	T2V1.0
T2R1	T2R1.15	T2R1.14	T2R1.13	T2R1.12	T2R1.11	T2R1.10	T2R1.9	T2R1.8	T2R1.7	T2R1.6	T2R1.5	T2R1.4	T2R1.3	T2R1.2	T2R1.1	T2R1.0
T2C1	T2C1.15	T2C1.14	T2C1.13	T2C1.12	T2C1.11	T2C1.10	T2C1.9	T2C1.8	T2C1.7	T2C1.6	T2C1.5	T2C1.4	T2C1.3	T2C1.2	T2C1.1	T2C1.0
T2CNB2									ET2L	T2OE1	T2POL1	TR2L	TF2	TF2L	TCC2	TC2L
T2V2	T2V2.15	T2V2.14	T2V2.13	T2V2.12	T2V2.11	T2V2.10	T2V2.9	T2V2.8	T2V2.7	T2V2.6	T2V2.5	T2V2.4	T2V2.3	T2V2.2	T2V2.1	T2V2.0
T2R2	T2R2.15	T2R2.14	T2R2.13	T2R2.12	T2R2.11	T2R2.10	T2R2.9	T2R2.8	T2R2.7	T2R2.6	T2R2.5	T2R2.4	T2R2.3	T2R2.2	T2R2.1	T2R2.0
T2C2	T2C2.15	T2C2.14	T2C2.13	T2C2.12	T2C2.11	T2C2.10	T2C2.9	T2C2.8	T2C2.7	T2C2.6	T2C2.5	T2C2.4	T2C2.3	T2C2.2	T2C2.1	T2C2.0
T2CFG1									T2CI	DIV2	DIV1	DIV0	T2MD	CCF1	CCF0	C/T2
T2CFG2									T2CI	DIV2	DIV1	DIV0	T2MD	CCF1	CCF0	C/T2

Table 6. Peripheral Register Reset Values

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO0									1	1	1	1	1	1	1	1
PO1									1	1	1	1	1	1	1	1
PO2									1	1	1	1	1	1	1	1
PO3									1	1	1	1	1	1	1	1
EIF0									0	0	0	0	0	0	0	0
EIE0									0	0	0	0	0	0	0	0
PI0									s	s	s	s	s	s	s	s
PI1									s	s	s	s	s	s	s	s
PI2									s	s	s	s	s	s	s	s
PI3									s	s	s	s	s	s	s	s
EIES0									0	0	0	0	0	0	0	0
PD0									0	0	0	0	0	0	0	0
PD1									0	0	0	0	0	0	0	0
PD2									0	0	0	0	0	0	0	0
PD3									0	0	0	0	0	0	0	0
RCNT	0	s	s	0	0	0	0	0	s	s	0	0	1	s	s	s
RTSS									s	s	s	s	s	s	s	s
RTSH	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
RTSL	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
RSSA									0	0	0	0	0	0	0	0
RASH									0	0	0	0	0	0	0	0
RASL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PO4									0	0	0	1	1	1	1	1
PO5									1	1	1	1	1	1	1	1
PO6									1	1	1	1	1	1	1	1
PO7									0	0	0	0	0	0	1	1
EIF1									0	0	0	0	0	0	0	0
EIE1									0	0	0	0	0	0	0	0
PI4									0	0	0	s	s	s	s	s
PI5									s	s	s	s	s	s	s	s
PI6									s	s	s	s	s	s	s	s
PI7									0	0	0	0	0	0	s	s
EIES1									0	0	0	0	0	0	0	0
PD4									0	0	0	0	0	0	0	0
PD5									0	0	0	0	0	0	0	0
PD6									0	0	0	0	0	0	0	0
PD7									0	0	0	0	0	0	0	0
SVS									0	0	0	0	0	0	0	0
WKO									0	0	0	0	0	0	0	0

Table 6. Peripheral Register Reset Values (continued)

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCNT									0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SCON0									0	0	0	0	0	0	0	0
SBUF0									0	0	0	0	0	0	0	0
SMD0									0	0	0	0	0	0	0	0
PR0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC1R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC0R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCRA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCFG									0	0	0	0	0	0	0	0
LCD[0..15]									0	0	0	0	0	0	0	0
T2CNA0									0	0	0	0	0	0	0	0
T2H0									0	0	0	0	0	0	0	0
T2RH0									0	0	0	0	0	0	0	0
T2CH0									0	0	0	0	0	0	0	0
SPIB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SCON1									0	0	0	0	0	0	0	0
SBUF1									0	0	0	0	0	0	0	0
SMD1									0	0	0	0	0	0	0	0
PR1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2CNB0									0	0	0	0	0	0	0	0
T2V0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2C0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2CFG0									0	0	0	0	0	0	0	0
OWA									0	0	0	0	0	0	0	0
OWD									0	0	0	0	0	0	0	0
SPICN									0	0	0	0	0	0	0	0
SPICF									0	0	0	0	0	0	0	0
SPICK									0	0	0	0	0	0	0	0
ICDC									s	s	s	s	s	s	s	s
ICDF									s	s	s	s	s	s	s	s
ICDB									s	s	s	s	s	s	s	s
ICDA	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
ICDD	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s
T2CNA1									0	0	0	0	0	0	0	0

Table 6. Peripheral Register Reset Values (continued)

REGISTER	REGISTER BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T2H1									0	0	0	0	0	0	0	0
T2RH1									0	0	0	0	0	0	0	0
T2CH1									0	0	0	0	0	0	0	0
T2CNA2									0	0	0	0	0	0	0	0
T2H2									0	0	0	0	0	0	0	0
T2RH2									0	0	0	0	0	0	0	0
T2CH2									0	0	0	0	0	0	0	0
T2CNB1									0	0	0	0	0	0	0	0
T2V1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2CNB2									0	0	0	0	0	0	0	0
T2V2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2R2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2CFG1									0	0	0	0	0	0	0	0
T2CFG2									0	0	0	0	0	0	0	0

Low-Power LCD Microcontroller

System Timing

For maximum versatility, the MAXQ2000 generates its internal system clock from one of five possible sources:

- Internal ring oscillator
- External high-frequency crystal or ceramic resonator, using an internal oscillator
- External high-frequency clock source
- External 32kHz crystal or ceramic resonator, using an internal oscillator
- External 32kHz clock source

A crystal warmup counter enhances operational reliability. Each time the external crystal oscillation must restart, such as after exiting Stop mode, the device initiates a crystal warmup period of 65,536 oscillations. This allows time for the crystal amplitude and frequency to stabilize before using it as a clock source. While in the warmup mode, the device can begin operation from the internal ring oscillator and automatically switch back to the crystal as soon as it is ready.

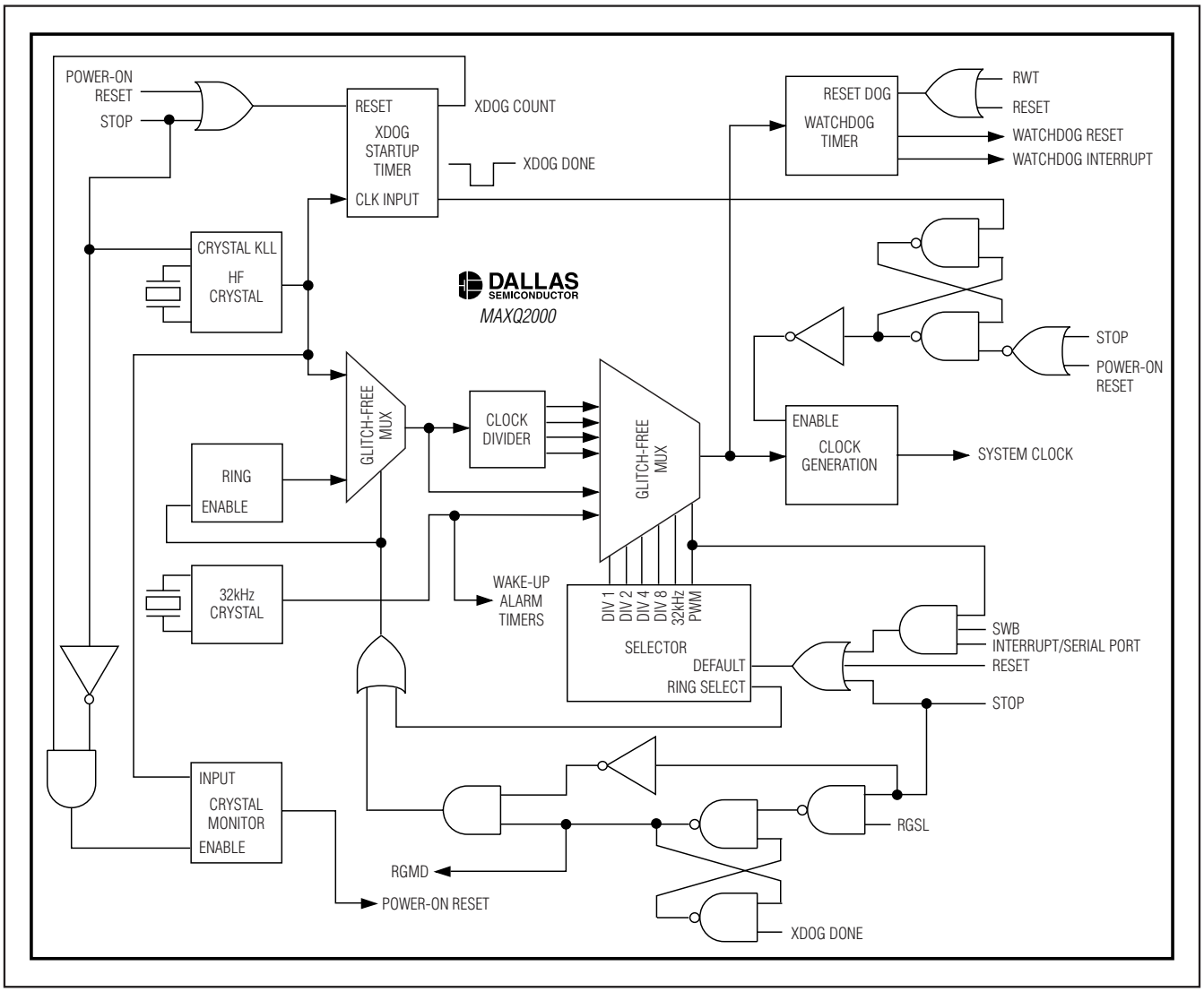


Figure 2. Clock Sources

Power Management

Advanced power-management features minimize power consumption by dynamically matching the processing speed of the device to the required performance level. This means device operation can be slowed and power consumption minimized during periods of reduced activity. When more processing power is required, the microcontroller can increase its operating frequency. Software-selectable clock-divide operations allow flexibility, selecting whether a system clock cycle is 1, 2, 4, or 8 oscillator cycles. By performing this function in software, a lower power state can be entered without the cost of additional hardware.

For extremely power-sensitive applications, three additional low-power modes are available:

- PMM1: divide-by-256 power-management mode (PMME = 1, CD1:0 = 00b)
- PMM2: 32kHz power-management mode (PMME = 1, CD1:0 = 11b)
- Stop mode (STOP = 1)

In PMM1, one system clock is 256 oscillator cycles, significantly reducing power consumption while the microcontroller functions at reduced speed. In PMM2, the device can run even slower by using the 32kHz oscillator as the clock source. The optional switchback feature allows enabled interrupt sources including external interrupts, UARTs, and the SPI module to quickly exit the power-management modes and return to a faster internal clock rate.

Power consumption reaches its minimum in Stop mode. In this mode, the external oscillator, system clock, and all processing activity is halted. Stop mode is exited when an enabled external interrupt pin is triggered, an external reset signal is applied to the $\overline{\text{RESET}}$ pin, or the RTC time-of-day alarm is activated. Upon exiting Stop mode, the microcontroller can choose to wait for the external high-frequency crystal to complete its warmup period, or it can start execution immediately from its internal ring oscillator while the warmup period completes.

Interrupts

Multiple interrupt sources are available for quick response to internal and external events. The MAXQ 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 user-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, software 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, the user program must determine whether a jump to 0000h came from a reset or interrupt source.

Once software 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. Because the interrupts are evaluated by user software, the user can define a unique interrupt priority scheme for each application. The following interrupt sources are available. Sources marked with an asterisk are not available on the 56-pin version.

- Watchdog Interrupt
- External Interrupts 0 to 15 (INT10*, INT11*)
- RTC Time-of-Day and Subsecond Alarms
- Serial Port 0 Receive and Transmit Interrupts
- Serial Port 1 Receive and Transmit Interrupts*
- SPI Mode Fault, Write Collision, Receive Overrun, and Transfer Complete Interrupts
- Timer 0 Low Compare, Low Overflow, Capture/Compare, and Overflow Interrupts
- Timer 1 Low Compare, Low Overflow, Capture/Compare, and Overflow Interrupts
- Timer 2 Low Compare, Low Overflow, Capture/Compare, and Overflow Interrupts
- 1-Wire Presence Detect, Transmit Buffer Empty, Transmit Shift Register Empty, Receive Buffer Full, and Shift Register Full, Short, and Low Interrupts*

Reset Sources

Several reset sources are provided for microcontroller control. Although code execution is halted in the reset state, the high-frequency oscillator and the ring oscillator continue to oscillate. Internal resets such as the power-on and watchdog resets assert the $\overline{\text{RESET}}$ pin low.

Low-Power LCD Microcontroller

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_{DDIO} climbs above approximately 1.8V. 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
- The ring oscillator becomes the clock source and
- Code execution begins at location 8000h

Watchdog Timer Reset

The watchdog timer functions are described in the *MAXQ Family User's Guide*. Execution resumes at location 8000h following a watchdog timer reset.

External System Reset

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

I/O Ports

The microcontroller uses the type C and type D bidirectional I/O ports described in the *MAXQ Family User's Guide*. The use of two port types allows for maximum flexibility when interfacing to external peripherals. Each port has eight independent, general-purpose I/O pins and three configure/control registers. Many pins support alternate functions such as timers or interrupts, which are enabled, controlled, and monitored by dedicated peripheral registers. Using the alternate function automatically converts the pin to that function.

Type-C port pins have Schmitt Trigger receivers and full CMOS output drivers, and can support alternate functions. The pin is either tri-stated or weak pullup when defined as an input, dependent on the state of the corresponding bit in the output register.

Type-D port pins have Schmitt Trigger receivers and full CMOS output drivers, and can support alternate functions. The pin is either tri-stated or weak pullup when defined as an input, dependent on the state of the corresponding bit in the output register. All type-D pins also have interrupt capability.

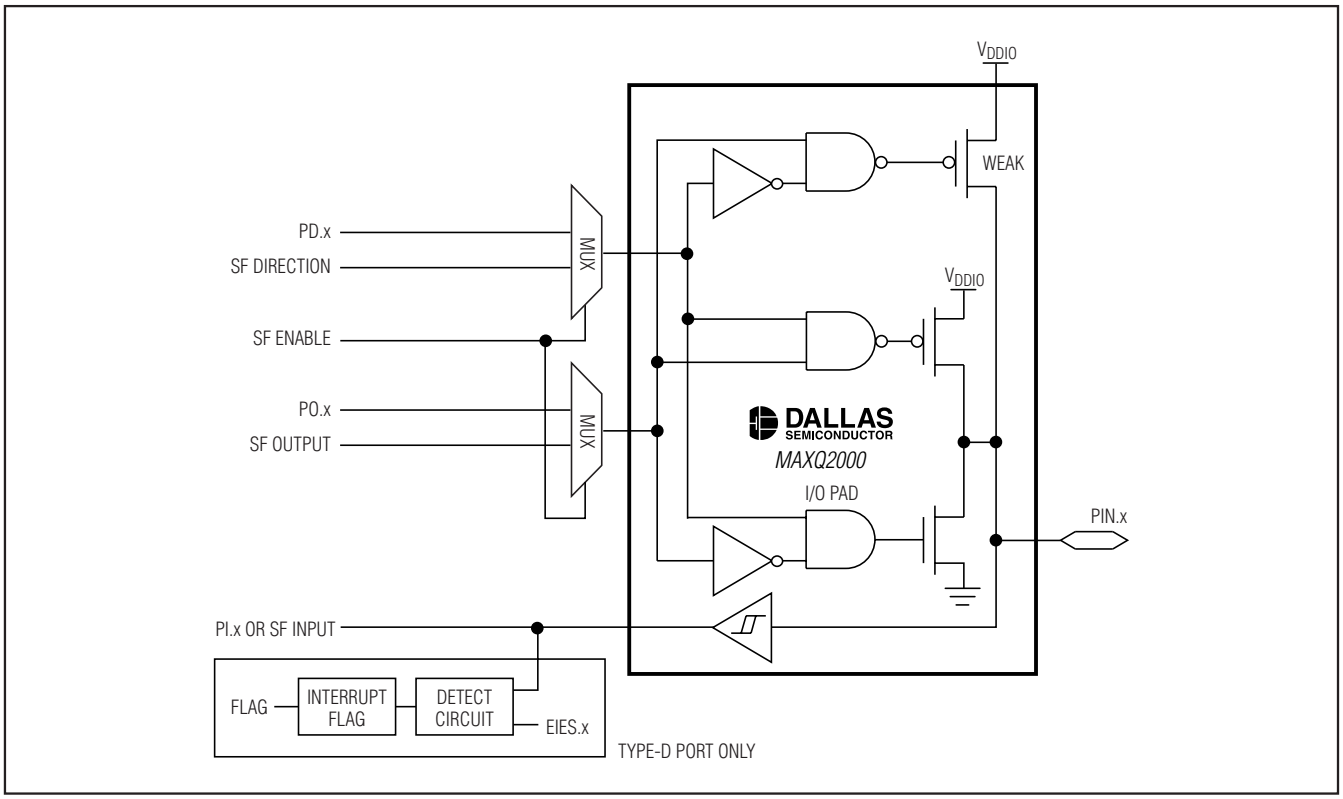


Figure 3. Type-C/D Port Pin Schematic

High-Speed Hardware Multiplier

The hardware multiplier module performs high-speed multiply, square, and accumulate operations, and can complete a 16-bit x 16-bit multiply-and-accumulate operation in a single cycle. The hardware multiplier consists of two 16-bit parallel-load operand registers (MA, MB), an accumulator that is formed by up to three 16-bit parallel registers (MC2, MC1, and MC0), and a status/control register (MCNT). Loading the registers can automatically initiate the operation, saving time on repetitive calculations. The accumulate function of the hardware multiplier is an essential element of digital filtering, signal processing, and PID control systems.

The hardware multiplier module supports the following operations:

- Multiply unsigned (16 bit x 16 bit)
- Multiply signed (16 bit x 16 bit)
- Multiply-Accumulate unsigned (16 bit x 16 bit)
- Multiply-Accumulate signed (16 bit x 16 bit)
- Square unsigned (16 bit)
- Square signed (16 bit)
- Square-Accumulate unsigned (16 bit)
- Square-Accumulate signed (16 bit)

Real-Time Clock

A binary real-time clock keeps the time of day in absolute seconds with 1/256-second resolution. The 32-bit second counter can count up to approximately 136 years and be translated to calendar format by the application software. A time-of-day alarm and independent subsecond alarm can cause an interrupt, or wake the device from Stop mode.

The independent subsecond alarm runs from the same RTC, and allows the application to perform periodic interrupts up to ones with a granularity of approximately 3.9ms. This creates an additional timer that can be used to measure long periods without performance degradations. Traditionally, long time periods have been measured using multiple interrupts from shorter programmable timers. Each timer interrupt required servicing, with each accompanying interruption slowing system operation. By using the RTC subsecond timer as a long-period timer, only one interrupt is needed, eliminating the performance hit associated with using a shorter timer.

An internal crystal oscillator clocks the RTC using integrated 6pF load capacitors, and give the best performance when mated with a 32.768kHz crystal rated for a 6pF load. No external load capacitors are required. Higher accuracy can be obtained by supplying an external clock source to the RTC. The frequency accuracy of a crystal-based oscillator circuit is dependent upon crystal accuracy, the match between the crystal and the oscillator capacitor load, ambient temperature, etc. An error of 20ppm is equivalent to approximately 1 minute per month.

Programmable Timers

The microcontroller incorporates three 16-bit programmable instances of the Timer 2 peripheral, denoted TR2A, TR2B, and TR2C. These timers can be used in counter/timer/capture/compare/PWM functions, allowing precise control of internal and external events. Timer 2 supports optional single-shot, external gating, and polarity control options.

Timer 2

The Timer 2 peripheral includes the following:

- 16-bit autoreload timer/counter
- 16-bit capture
- 16-bit counter
- 8-bit capture and 8-bit timer
- 8-bit counter and 8-bit timer

Watchdog Timer

An internal watchdog timer greatly increases system reliability. The timer resets the device if software execution is disturbed. The watchdog timer is a free-running counter designed to be periodically reset by the application software. If software is operating correctly, the counter will be periodically reset and never reach its maximum count. However, if software operation is interrupted, the timer does not reset, triggering a system reset and optionally a watchdog timer interrupt. This protects the system against electrical noise or electrostatic discharge (ESD) upsets that could cause uncontrolled processor operation. The internal watchdog timer is an upgrade to older designs with external watchdog devices, reducing system cost and simultaneously increasing reliability.

Low-Power LCD Microcontroller

The watchdog timer is controlled through bits in the WDCN register. Its timeout period can be set to one of four programmable intervals ranging from 2^{12} to 2^{21} system clocks in its default mode, allowing flexibility to support different types of applications. The interrupt occurs 512 system clocks before the reset, allowing the system to execute an interrupt and place the system in a known, safe state before the device performs a total system reset. At 16MHz, watchdog timeout periods can be programmed from 256µs to 33.5s, depending on the system clock mode.

Serial Peripherals

The microcontroller incorporates several common serial-peripheral interfaces for interconnection with popular external devices. Multiple formats provide maximum flexibility and lower cost when designing a system.

UARTs

Serial interfacing is provided through one (-RBX/-RBX+) or two (-RAX/-RAX+/-RFX/-RFX+) 8051-style universal synchronous/asynchronous receiver/transmitters. The UART allows the device to conveniently communicate with other RS-232 interface-enabled devices, as well as PCs and serial modems when paired with an external RS-232 line driver/receiver. The dual independent UARTs can communicate simultaneously at different baud rates with two separate peripherals. The UART can detect framing errors and indicate the condition through a user-accessible software bit.

The time base of the serial ports is derived from either a division of the system clock or the dedicated baud clock generator. The following table summarizes the operating characteristics as well as the maximum baud rate of each mode:

1-Wire Bus Master

The MAXQ2000-RAX/-RAX+/-RFX/-RFX+ include a Dallas Semiconductor 1-Wire bus master, which communicates to other 1-Wire peripherals, including iButton® products, through a simple bidirectional signaling scheme over a single electrical connection. The bus master provides complete control of the 1-Wire bus and transmit and receive activities, and generates all timing and control sequences of the 1-Wire bus. Communication between the CPU and the bus master is achieved through read/write access of the 1-Wire master address (OWA) and 1-Wire master data (OWD) peripheral registers. Detailed operation of the 1-Wire bus is described in the *Book of iButton Standards* (**www.maxim-ic.com/iButtonbook**).

Serial-Peripheral Interface (SPI) Module

The SPI port is a common, high-speed, synchronous peripheral interface that shifts a bit stream of variable length and data rate between the microcontroller and other peripheral devices. The SPI can be used to communicate with other microcontrollers, serial shift registers, or display drivers. Multiple master and slave modes permit communication with multiple devices in the same system. Programmable clock frequency, character lengths, polarity, and error handling enhance the usefulness of the peripheral. The maximum baud rate of the SPI interface is 1/2 the system clock for master mode operation and 1/8 the system clock for slave mode operation.

MODE	TYPE	START BITS	DATA BITS	STOP BIT	MAX BAUD RATE AT 16MHz
Mode 0	Synchronous	N/A	8	N/A	4Mbps
Mode 1	Asynchronous	1	8	1	500kbps
Mode 2	Asynchronous	1	8 + 1	1	500kbps
Mode 3	Asynchronous	1	8 + 1	1	500kbps

iButton is a registered trademark of Dallas Semiconductor Corp., a wholly owned subsidiary of Maxim Integrated Products, Inc.

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 4 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, and
- a set of debug service routines stored in the utility ROM.

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

- Background mode allows the host to configure and set up the in-circuit debugger while the CPU continues to execute the application software at full speed. Debug mode can be invoked from background mode.
- Debug mode allows the debug engine to take control of the CPU, providing read/write access to internal registers and memory, and single-step trace operation.

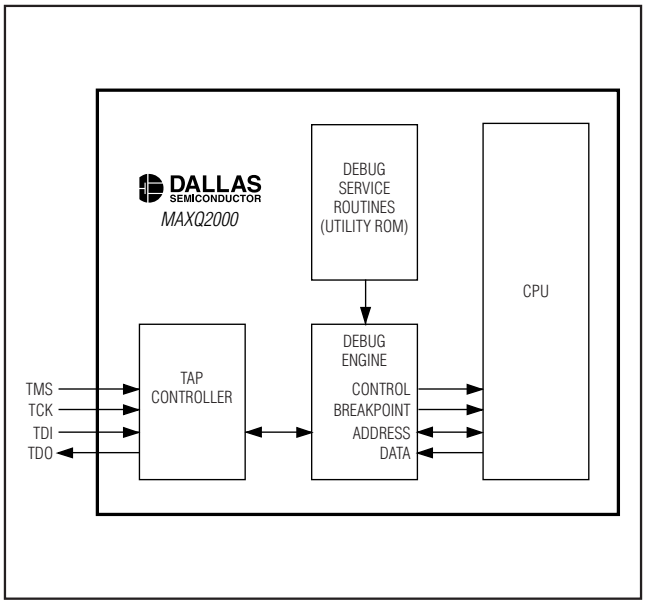


Figure 4. In-Circuit Debugger

LCD Controller

The MAXQ2000 microcontroller incorporates an LCD controller that interfaces to common low-voltage displays. By incorporating the LCD controller into the microcontroller, the design requires only an LCD glass rather than a considerably more expensive LCD module. Every character in an LCD glass is composed of one or more segments, each of which is activated by selecting the appropriate segment and common signal. The microcontroller can multiplex combinations of up to 33 segment (SEG0–SEG32) outputs and four common signal outputs (COM0–COM3). Unused segment outputs can be used as general-purpose port pins.

The segments are easily addressed by writing to dedicated display memory. Once the LCD controller settings and display memory have been initialized, the 17-byte display memory is periodically scanned, and the segment and common signals are generated automatically at the selected display frequency. No additional processor overhead is required while the LCD controller is running. Unused display memory can be used for general-purpose storage.

The design is further simplified and cost-reduced by the inclusion of software-adjustable internal voltage dividers to control display contrast, using either V_{DDIO} or an external voltage. If desired, contrast can also be controlled with an external resistance. The features of the LCD controller include the following:

- Automatic LCD segment and common-drive signal generation
- Four display modes supported:
  Static (COM0)
  1/2 duty multiplexed with 1/2 bias voltages (COM0, COM1)
  1/3 duty multiplexed with 1/3 bias voltages (COM0, COM1, COM2)
  1/4 duty multiplexed with 1/3 bias voltages (COM0, COM1, COM2, COM3)
- Up to 36 segment outputs and four common-signal outputs
- 17 bytes (136 bits) of display memory
- Flexible LCD clock source, selectable from 32kHz or HFClk / 128
- Adjustable frame frequency
- Internal voltage-divider resistors eliminate requirement for external components
- Internal adjustable resistor allows contrast adjustment without external components

Low-Power LCD Microcontroller

- Flexibility to use external resistors to adjust drive voltages and current capacity

A simple LCD-segmented glass interface example demonstrates the minimal hardware required to interface to a MAXQ2000 microcontroller. A two-character LCD is controlled, with each character containing seven segments plus decimal point. The LCD controller is configured for 1/2 duty cycle operation, meaning the active segment is controlled using a combination of segment signals, and COM0 or COM1 signals are used to select the active display.

Applications

The low-power, high-performance RISC architecture of the MAXQ2000 makes it an excellent fit for many portable or battery-powered applications that require cost-effective computing. The high-throughput core is complemented by a 16-bit hardware multiplier-accumulator, allowing the implementation of sophisticated computational algorithms. Applications benefit from a wide range of peripheral interfaces, allowing the microcontroller to communicate with many external devices. With integrated LCD support of up to 100 or 132 segments, applications can support complex user interfaces. Displays are driven directly with no additional external hardware required. Contrast can be adjusted using a built-in, adjustable resistor. The simplified architecture reduces component count and board space, critical factors in the design of portable systems.

The MAXQ2000 is ideally suited for applications such as medical instrumentation, portable blood glucose equipment, and data collection devices. For blood glucose measurement, the microcontroller integrates an SPI interface that directly connects with analog front ends for measuring test strips.

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/microcontrollers**.

- The MAXQ2000 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.
- The *MAXQ Family User's Guide: MAXQ2000 Supplement*, which contains detailed information on features specific to the MAXQ2000.

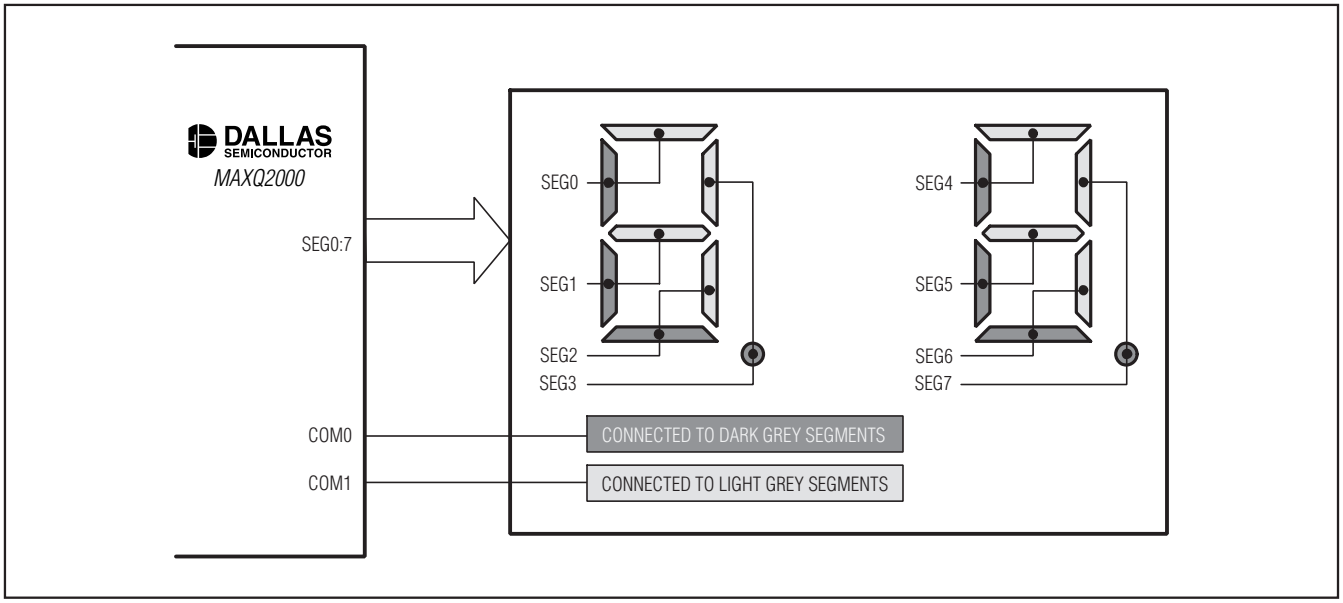


Figure 5. Two-Character, 1/2 Duty, LCD Interface Example

Development and Technical Support

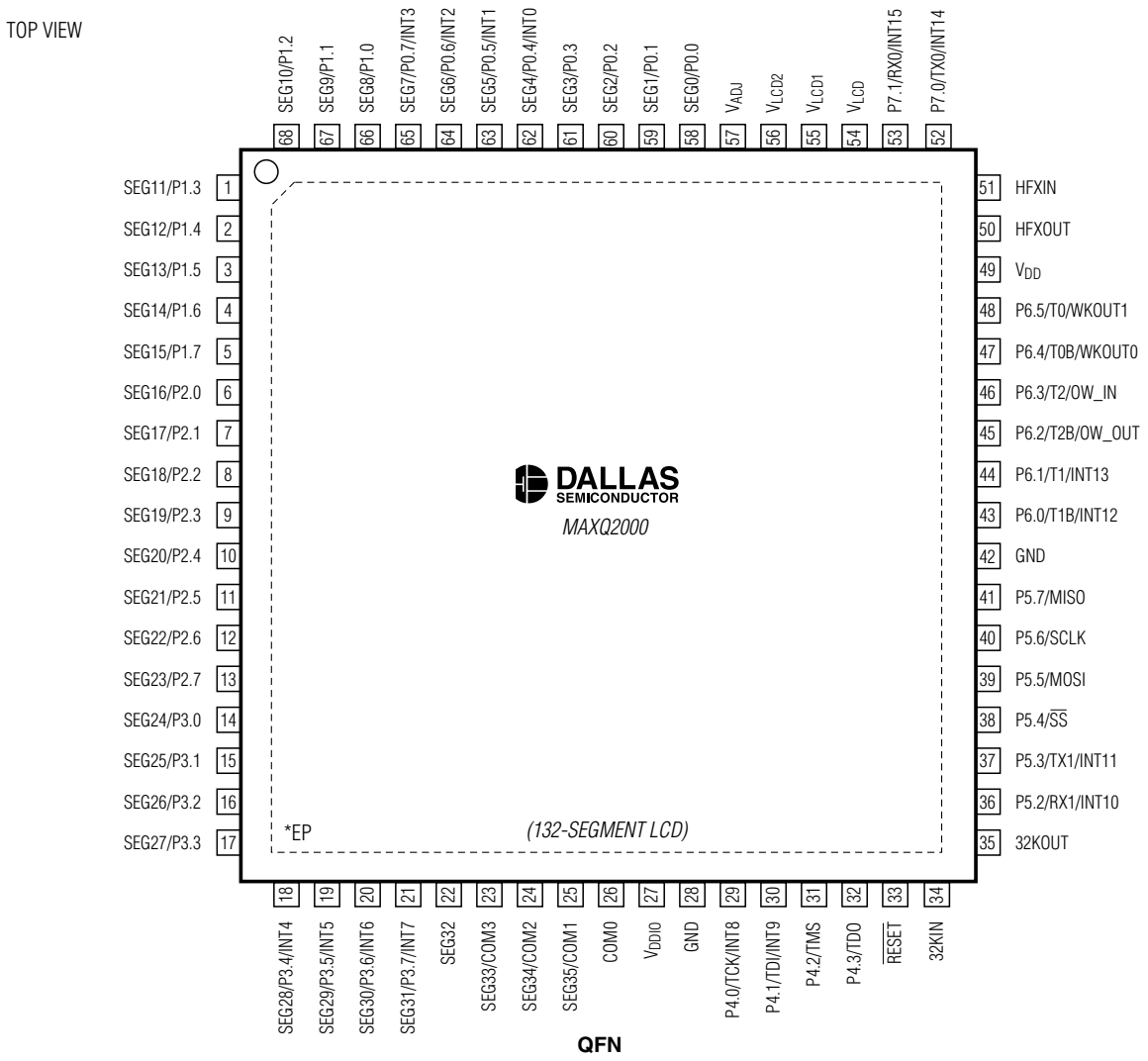
A variety of highly versatile, affordably priced development tools for this microcontroller are available from Maxim and third-party suppliers, including:

- Compilers
- In-circuit emulators
- Integrated development environments (IDEs)
- JTAG-to-serial converters for programming and debugging

A partial list of development tool vendors can be found on our website at **www.maxim-ic.com/MAXQ_tools**.

For technical support, go to **www.maxim-ic.com/support**.

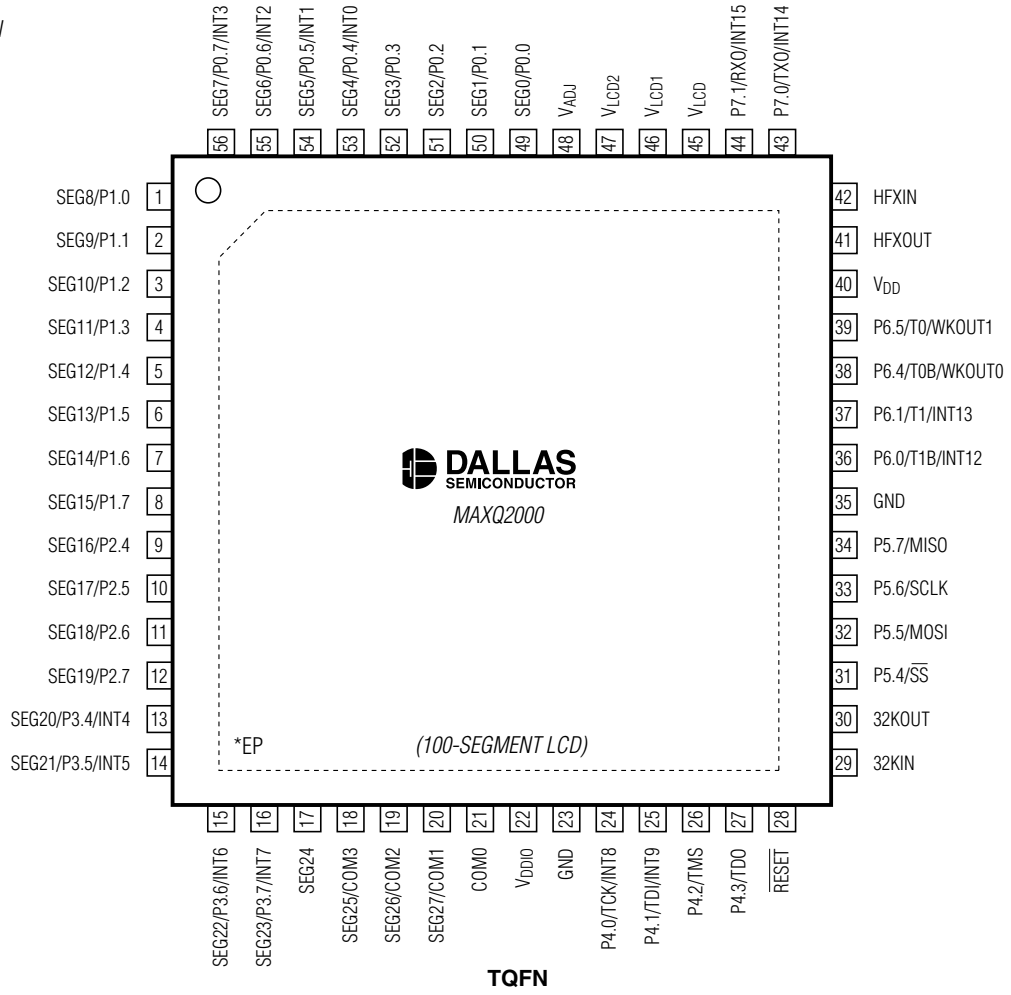
Pin Configurations



*EP = EXPOSED PAD.

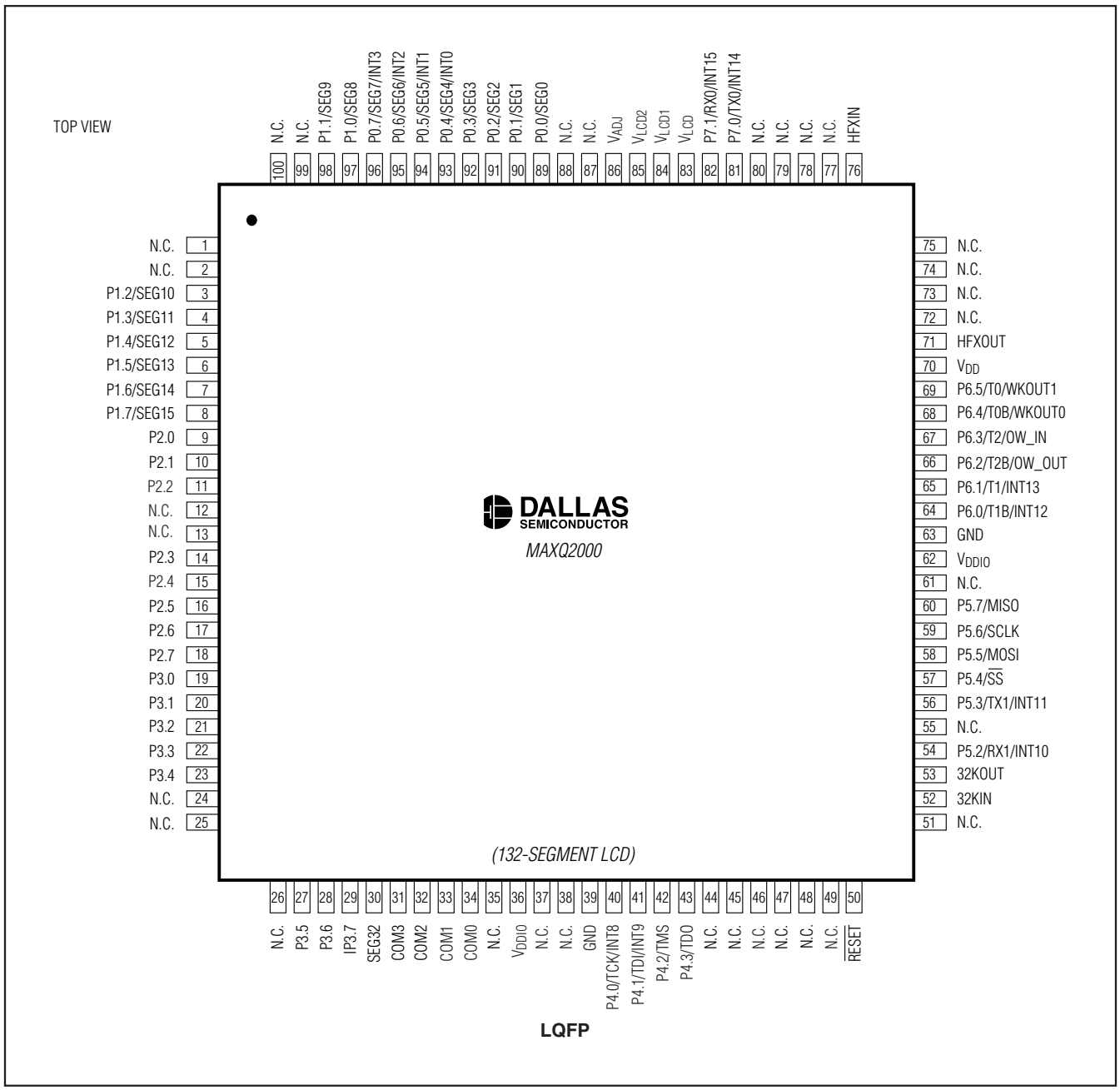
Low-Power LCD Microcontroller

Pin Configurations (continued)



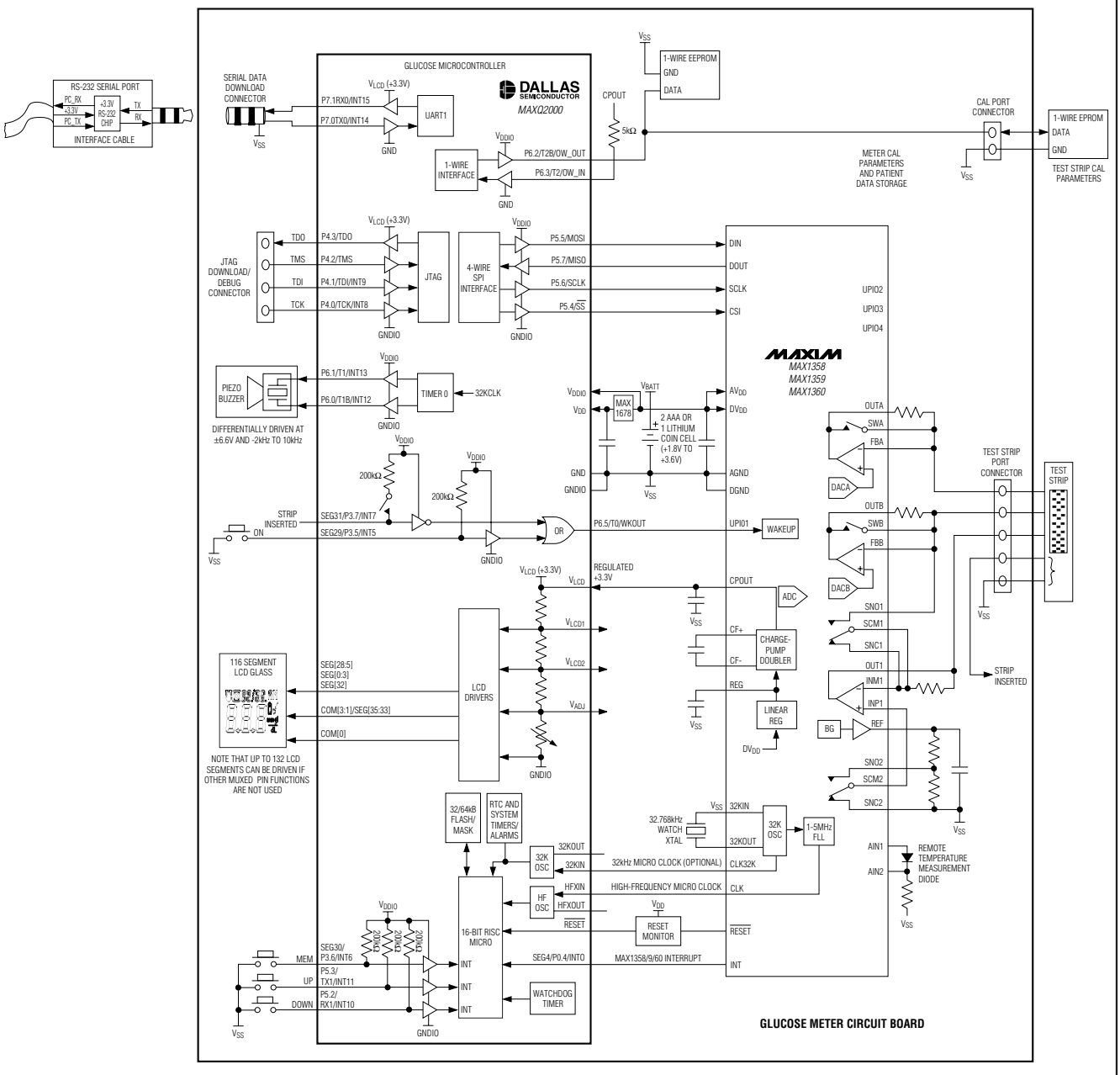
*EP = EXPOSED PAD.

Pin Configurations (continued)



Low-Power LCD Microcontroller

Typical Operating Circuit



Ordering Information

PART	PROGRAM MEMORY	DATA MEMORY	LCD SEGMENTS	EXTERNAL INTERRUPTS	UARTS	PIN-PACKAGE	PKG CODE
MAXQ2000-RAX	32kWord Flash	1kWord SRAM	132	16	2	68 QFN	G6800-4
MAXQ2000-RAX+	32kWord Flash	1kWord SRAM	132	16	2	68 QFN	G6800+4
MAXQ2000-RBX	32kWord Flash	1kWord SRAM	100	14	1	56 TQFN	T5688+2
MAXQ2000-RBX+	32kWord Flash	1kWord SRAM	100	14	1	56 TQFN	T5688-2
MAXQ2000-RFX	32kWord Flash	1kWord SRAM	132	16	2	100 LQFP	—
MAXQ2000-RFX+	32kWord Flash	1kWord SRAM	132	16	2	100 LQFP	—

Note: *All devices are specified over the -40°C to +85°C operating temperature range.*
+Denotes a Pb-free/RoHS-compliant package.

Package Information

For the latest package outline information and land patterns, go to **www.maxim-ic.com/packages**.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
68 QFN	G6800-4	**21-0122**
56 TQFN	T5688-2	**21-0135**
100 LQFP	—	**21-0297**

Low-Power LCD Microcontroller

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	10/04	Initial release for QFN package variant.	—
1	10/04	New product release for TQFN package variant.	1
2	12/04	In the *Features* section under *Peripheral Features*, corrected accumulator to show 48 bits (not 40 bits).	1
		In the *Electrical Characteristics* table, added Active Current line for 2.2V, 20MHz flash operation; $V_{IH2(MIN)}$ changed from 0.8 x V_{DDIO} to 0.75 x V_{DDIO}; updated V_{IH}, V_{IL}, V_{OH}, and V_{OL} data to match GBD/FTEC data.	2, 3
		Replaced the package drawing for 56-pin package.	38
3	6/05	Added lead-free part numbers to the *Ordering Information* table.	1
		In the *Electrical Characteristics* table under LCD Segment Voltage, clarified wording on V_{ADJ} spec to $V_{ADJ(MIN)}$ = V_{ADJ} and $V_{ADJ(MAX)}$ = 0.1V; changed ISEGxx to 3µA.	3
4	10/05	Clarified that flash memory write/erase cycles and data retention specifications are at +25°C.	4
5	1/06	Clarified V_{IH1}/V_{IH3} specifications, matching presented values to test program values (0.8 x V_{DDIO}); clarified VIH2 specification, matching presented values to test program values (0.8 x V_{LCD}); clarified V_{IL2} specification, matching presented values to test program values (0.2 x V_{DDIO}). These changes do not affect the testing or operation of the device.	2, 3
		Corrected typo on pin 38 (*Pin Configuration*) from P4/$\overline{SS}$ to P5.4/$\overline{SS}$.	33
6	3/06	Corrected Pb-free package number denotations. Should be MAXQ2000-RAX+ and MAXQ2000-RBX+.	1, 30, 34
7	6/06	Added Revision A3 typ and max conditions to I_{DD6} in the *Electrical Characteristics* table.	2
8	12/06	Added 100-pin LQFP package.	1, 7–11, 30, 35, 42
		Added EP lines and note to QFN and TQFN pin configurations.	33, 34
		Changed 4kWords Utility ROM (*Memory Organization* section) to 4kB.	13
		Changed 4k x 16 Utility ROM (Figure 1) to 2k x 16 Utility ROM, changed 8FFFh to 87FFh.	14
		Changed 4kWord to 4kB (*Utility ROM* section).	15
9	3/08	Added V_{DD} slew rate specification to *Electrical Characteristics* table.	2
		Corrected references of $\overline{SSEL}$ to $\overline{SS}$.	5, 6
		In the *Typical Operating Circuit*, added the reset monitor to ensure the V_{DD} slew rate specification is met.	36
		Added QFN and TQFN package codes to the *Ordering Information* table; removed package drawings and replaced with *Package Information* table.	37

Revision History (continued)

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
10	7/08	In the *Electrical Characteristics* table, changed the conditions for R_{LADJ} from LRA4:LRA0 = 0 to LRA4:LRA0 = 11111b; added the t_{SSE} parameter to the *SPI Timing* section.	3, 5
		Adjusted the location of "t_{MOV}" in the *SPI Master Timing* figure.	6