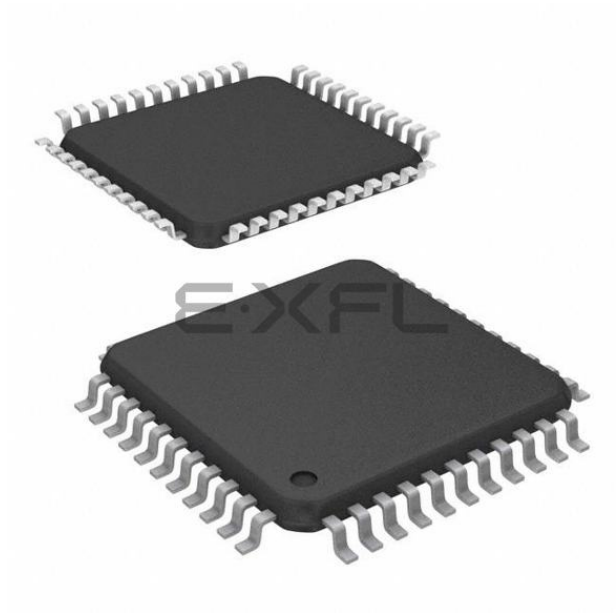
Analog Devices Inc./Maxim Integrated - DS89C420-ENL Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	33MHz
Connectivity	SIO, UART/USART
Peripherals	Power-Fail Reset, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/ds89c420-enl

AC CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = -40°C to +85°C)* (Figure 1, Figure 2, and Figure 3)

PARAMETER		SYMBOL	1 CYCLE PAGE MODE 1		2 CYCLE PAGE MODE 1		4 CYCLE PAGE MODE 1		PAGE MODE 2		NON-PAGE MODE		UNITS
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
System Clock (Note 1)	External Oscillator (25MHz, 33MHz)	1 / t_{CLCL}	0	25	0	25	0	25	0	25	0	25	MHz
			0	33	0	33	0	33	0	33	0	33	
	External Crystal (25MHz, 33MHz)		1	25	1	25	1	25	1	25	1	25	
			1	33	1	33	1	33	1	33	1	33	
ALE Pulse Width (Note 2)		t_{LHLL}	$0.5t_{CLCL}$ - 2 + t_{STC3}		t_{CLCL} - 2 + t_{STC3}		$2t_{CLCL}$ - 4 + t_{STC3}		$1.5t_{CLCL}$ - 5 + t_{STC3}		$1.5t_{CLCL}$ - 5 + t_{STC3}		ns
Port 0 Instruction Address Valid to ALE Low		t_{AVLL}							t_{CLCL} - 2		$0.5t_{CLCL}$ - 2		ns
Port 2 Instruction Address Valid to ALE Low		t_{AVLL2}	$0.5t_{CLCL}$ - 4		$0.5t_{CLCL}$ - 4		$1.5t_{CLCL}$ - 5		$0.5t_{CLCL}$ - 2		t_{CLCL} - 2		ns
Port 0 Data AddressValid to ALE Low		t_{AVLL3}							t_{CLCL} - 2 + t_{STC3}		$0.5t_{CLCL}$ - 2 + t_{STC3}		ns
Program Address Hold After ALE Low		t_{LLAX}	$0.5t_{CLCL}$ - 8		$1.5t_{CLCL}$ - 8		$2.5t_{CLCL}$ - 8		$0.5t_{CLCL}$ - 8		$0.5t_{CLCL}$ - 8		ns
Address Hold After ALE Low MOVX Write		t_{LLAX2}	$0.5t_{CLCL}$ - 8 + t_{STC4}		$1.5t_{CLCL}$ - 8 + t_{STC4}		$2.5t_{CLCL}$ - 8 + t_{STC4}		$0.5t_{CLCL}$ - 8 + t_{STC4}		$0.5t_{CLCL}$ - 8 + t_{STC4}		ns
Address Hold After ALE Low MOVX Read		t_{LLAX3}	$0.5t_{CLCL}$ - 8 + t_{STC4}		$1.5t_{CLCL}$ - 8 + t_{STC4}		$2.5t_{CLCL}$ - 8 + t_{STC4}		$0.5t_{CLCL}$ - 8 + t_{STC4}		$0.5t_{CLCL}$ - 8 + t_{STC4}		ns
ALE Low to Valid Instruction In		t_{LLIV}								$2.5t_{CLCL}$ - 20		$2.5t_{CLCL}$ - 20	ns
ALE Low to $\overline{PSEN}$ Low		t_{LLPL}							$1.5t_{CLCL}$ - 6		$0.5t_{CLCL}$ - 6		ns
$\overline{PSEN}$ Pulse Width for Program Fetch		t_{PLPH}	t_{CLCL} - 5		t_{CLCL} - 5		$2t_{CLCL}$ - 5		t_{CLCL} - 5		$2t_{CLCL}$ - 5		ns

PARAMETER	SYMBOL	1 CYCLE PAGE MODE 1		2 CYCLE PAGE MODE 1		4 CYCLE PAGE MODE 1		PAGE MODE 2		NON-PAGE MODE		UNITS
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
$\overline{PSEN}$ Low to Valid Instruction In	t_{PLIV}		t_{CLCL} - 18		t_{CLCL} - 18		$2t_{CLCL}$ - 18		t_{CLCL} - 18		$2t_{CLCL}$ - 18	ns
Input Instruction Hold After $\overline{PSEN}$	t_{PXIX}	0		0		0		0		0		ns
Input Instruction Float After $\overline{PSEN}$	t_{PXIZ}								t_{CLCL} - 5		t_{CLCL} - 5	ns
Port 0 Address to Valid Instruction In	t_{AVIV0}								$1.5t_{CLCL}$ - 20		$3t_{CLCL}$ - 20	ns
Port 2 Address to Valid Instruction In	t_{AVIV2}		t_{CLCL} - 18		$1.5t_{CLCL}$ - 18		$2.5t_{CLCL}$ - 18		$3t_{CLCL}$ - 20		$3.5t_{CLCL}$ - 20	ns
$\overline{PSEN}$ Low to Port 0 Address Float	t_{PLAZ}								0		0	ns
$\overline{RD}$ Pulse Width (P3.7) (Note 2)	t_{RLRH}	t_{CLCL} - 5 + t_{STC1}		t_{CLCL} - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		ns
$\overline{WR}$ Pulse Width (P3.6) (Note 2)	t_{WLWH}	t_{CLCL} - 5 + t_{STC1}		t_{CLCL} - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		$2t_{CLCL}$ - 5 + t_{STC1}		ns
$\overline{RD}$ (P3.7) Low to Valid Data In (Note 2)	t_{RLDV}		t_{CLCL} - 15 + t_{STC1}		t_{CLCL} - 15 + t_{STC1}		$2t_{CLCL}$ - 15 + t_{STC1}		$2t_{CLCL}$ - 15 + t_{STC1}		$2t_{CLCL}$ - 15 + t_{STC1}	ns
Data Hold After $\overline{RD}$ (P3.7)	t_{RHDX}	0		0		0		0		0		ns
Data Float After $\overline{RD}$ (P3.7)	t_{RHDZ}								t_{CLCL} - 5		t_{CLCL} - 5	ns
MOVX ALE Low to Input Data Valid (Note 2)	t_{LLDV}								$2.5t_{CLCL}$ - 20 + t_{STC1}		$2.5t_{CLCL}$ - 20 + t_{STC1}	ns
Port 0 Address to Valid Data In (Note 2)	t_{AVDV0}								$3t_{CLCL}$ - 20 + t_{STC1}		$3t_{CLCL}$ - 20 + t_{STC1}	ns

PARAMETER	SYMBOL	1 CYCLE PAGE MODE 1		2 CYCLE PAGE MODE 1		4 CYCLE PAGE MODE 1		PAGE MODE 2		NON-PAGE MODE		UNITS
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
Port 2 Address to Valid Data In (Note 2)	t_{AVDV2}		t_{CLCL} - 16 + t_{STC1}		1.5t_{CLCL} - 16 + t_{STC1}		3.5t_{CLCL} - 16 + t_{STC1}		3.0t_{CLCL} - 16 + t_{STC1}		3.5t_{CLCL} - 20 + t_{STC1}	ns
ALE Low to $\overline{RD}$ or $\overline{WR}$ Low (Note 2)	t_{LLRL} (t_{LLWL})	0.5t_{CLCL} - 8 + t_{STC2}	0.5t_{CLCL} + 1 + t_{STC2}	2t_{CLCL} - 8 + t_{STC2}	2t_{CLCL} + 8 + t_{STC2}	4t_{CLCL} - 8 + t_{STC2}	4t_{CLCL} + 8 + t_{STC2}	0.5t_{CLCL} - 8 + t_{STC2}	0.5t_{CLCL} + 4 + t_{STC2}	0.5t_{CLCL} - 8 + t_{STC2}	0.5t_{CLCL} + 4 + t_{STC2}	ns
Port 0 Address Valid to $\overline{RD}$ or $\overline{WR}$ Low (Note 2)	t_{AVRL0} (t_{AVWL0})							1.5t_{CLCL} - 5 + t_{STC2}		t_{CLCL} - 5 + t_{STC2}		ns
Port 2 Address Valid to $\overline{RD}$ or $\overline{WR}$ Low (Note 2)	t_{AVRL2} (t_{AVWL2})	0 + t_{STC5} - 5		0.5t_{CLCL} - 5 + t_{STC5}		1.5t_{CLCL} - 5 + t_{STC5}		t_{CLCL} - 5 + t_{STC5}		1.5t_{CLCL} - 5 + t_{STC5}		ns
Data Out Valid to $\overline{WR}$ Transition (Note 1)	t_{QVWX}	-5		-5		-5		-5		-5		ns
Data Hold After $\overline{WR}$ (Note 1)	t_{WHQX}	20		20		20		20		20		ns
$\overline{RD}$ or $\overline{WR}$ High to ALE High (Note 1)	t_{RHLH} (t_{WHLH})	t_{STC2} - 2	t_{STC2} + 6	t_{STC2} - 2	t_{STC2} + 6	t_{STC2} - 2	t_{STC2} + 6	t_{STC2} - 2	t_{STC2} + 6	t_{STC2} - 2	t_{STC2} + 6	ns

**Specifications to -40°C are guaranteed by design and not production tested.*

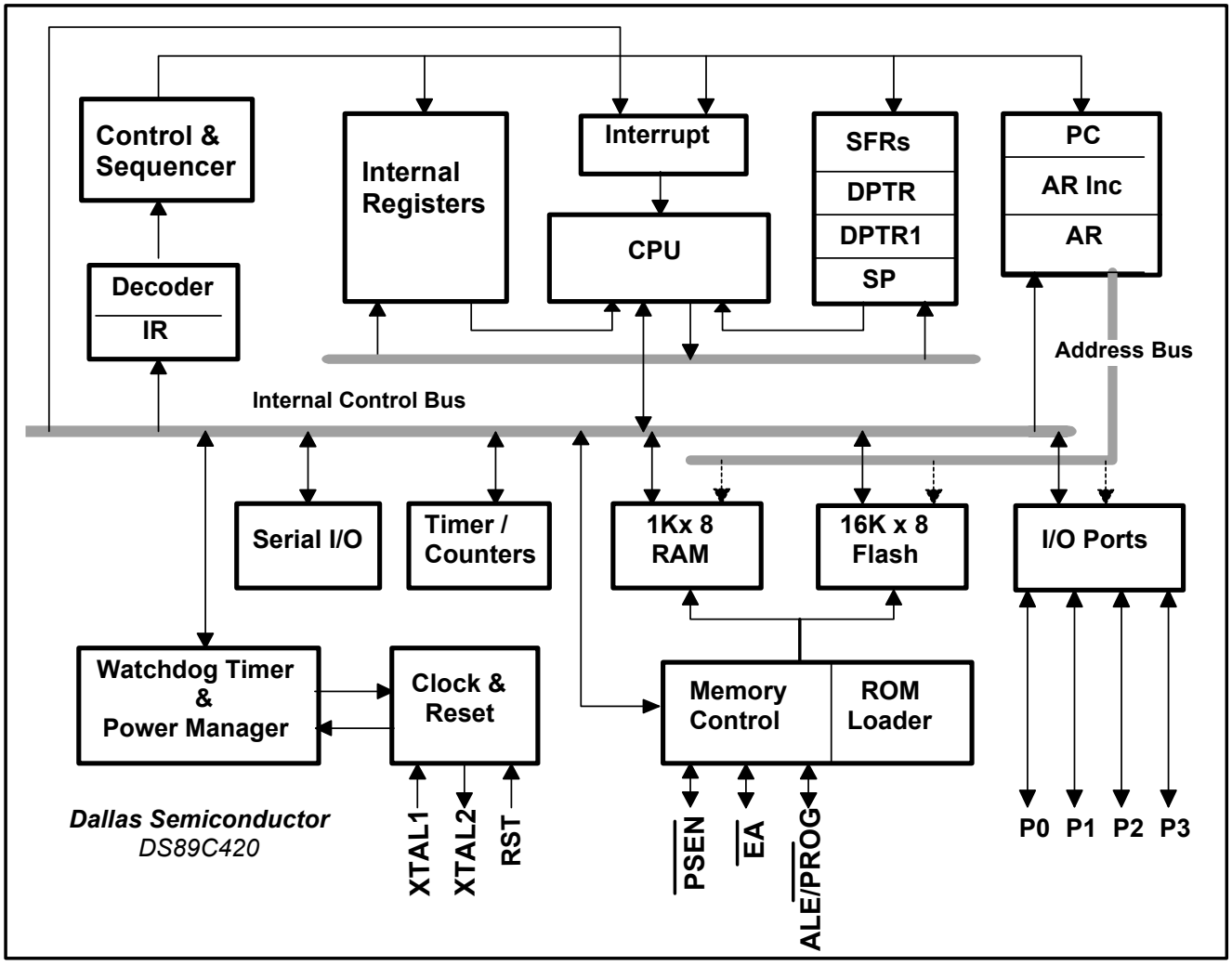
PIN DESCRIPTION

PIN			NAME	FUNCTION		
DIP	PLCC	TQFP				
40	12, 44	6, 38	V_{CC}	V_{CC} - +5V		
20	1, 22, 23, 34	16, 17, 28, 39	GND	Logic Ground		
9	10	4	RST	External Reset. The RST input pin is bidirectional and contains a Schmitt trigger to recognize external active-high reset inputs. The pin also employs an internal pulldown resistor to allow for a combination of wire-ORed external reset sources. An RC is not required for power-up, since the device provides this function internally.		
19	21	15	XTAL1	XTAL1, XTAL2. The crystal oscillator pins XTAL1 and XTAL2 provide support for fundamental mode parallel resonant, AT cut crystals. XTAL1 also acts as an input if there is an external clock source in place of a crystal. XTAL2 serves as the output of the crystal amplifier.		
18	20	14	XTAL2			
29	32	26	$\overline{PSEN}$	Program Store Enable. This signal is commonly connected to optional external program memory as a chip enable. $\overline{PSEN}$ provides an active-low pulse and is driven high when external program memory is not being accessed. In 1-cycle page mode 1, $\overline{PSEN}$ remains low for consecutive page hits.		
30	33	27	ALE/$\overline{PROG}$	Address Latch Enable. Functions as a clock to latch the external address LSB from the multiplexed address/data bus on Port 0. This signal is commonly connected to the latch enable of an external 373 family transparent latch. In default mode, ALE has a pulse width of 1.5 XTAL1 cycles and a period of four XTAL1 cycles. In page mode, the ALE pulse width is altered according to the page mode selection. In traditional 8051 mode, ALE is high when using the EMI reduction mode and during a reset condition. ALE can be enabled by writing ALEON = 1 (PMR.2). Note that ALE operates independently of ALEON during external memory accesses. As an alternate mode, this pin ($\overline{PROG}$) is used to execute the parallel program function.		
39	43	37	P0.0 (AD0)	Port 0 (AD0–7), I/O. Port 0 is an open-drain 8-bit, bidirectional I/O port. As an alternate function, Port 0 can function as the multiplexed address/data bus to access off-chip memory. During the time when ALE is high, the LSB of a memory address is presented. When ALE falls to a logic 0, the port transitions to a bidirectional data bus. This bus is used to read external program memory and read/write external RAM or peripherals. When used as a memory bus, the port provides weak pullups for logic 1 outputs. The reset condition of Port 0 is three-state. Pullup resistors are required when using Port 0 as an I/O port.		
38	42	36	P0.1 (AD1)			
37	41	35	P0.2 (AD2)			
36	40	34	P0.3 (AD3)			
35	39	33	P0.4 (AD4)			
34	38	32	P0.5 (AD5)			
33	37	31	P0.6 (AD6)			
32	36	30	P0.7 (AD7)			
1–8	2–9	40–44, 1–3	P1.0–P1.7	Port 1, I/O. Port 1 functions as both an 8-bit, bidirectional I/O port and an alternate functional interface for timer 2 I/O, new external interrupts, and new serial port 1. The reset condition of port 1 is with all bits at logic 1. In this state, a weak pullup holds the port high. This condition also serves as an input state, since any external circuit that writes to the port overcomes the weak pullup. When software writes a 0 to any port pin, the DS89C420 activates a strong pulldown that remains on until either a 1 is written or a reset occurs. Writing a 1 after the port has been at 0 causes a strong transition driver to turn on, followed by a weaker sustaining pullup. Once the momentary strong driver turns off, the port again becomes the output high (and input) state. The alternate functions of Port 1 are outlined below.		
				PORT	**ALTERNATE**	**FUNCTION**
1	2	40		P1.0	T2	External I/O for Timer/Counter 2
2	3	41		P1.1	T2EX	Timer 2 Capture/Reload Trigger
3	4	42		P1.2	RXD1	Serial Port 1 Receive
4	5	43		P1.3	TXD1	Serial Port 1 Transmit
5	6	44		P1.4	INT2	External Interrupt 2 (Positive Edge Detect)
6	7	1		P1.5	$\overline{INT3}$	External Interrupt 3 (Negative Edge Detect)
7	8	2		P1.6	INT4	External Interrupt 4 (Positive Edge Detect)
8	9	3		P1.7	$\overline{INT5}$	External Interrupt 5 (Negative Edge Detect)

PIN DESCRIPTION (continued)

<table>
<tr><th colspan="3">PIN</th><th rowspan="2">NAME</th><th colspan="3">FUNCTION</th></tr>
<tr><th>DIP</th><th>PLCC</th><th>PDIP</th><th></th><th></th><th></th></tr>
<tr><td>21</td><td>24</td><td>18</td><td>P2.0 (A8)</td><td colspan="3" rowspan="8">Port 2 (A8–15), I/O. Port 2 is an 8-bit, bidirectional I/O port. The reset condition of port 2 is logic high. In this state, a weak pullup holds the port high. This condition also serves as an input mode, since any external circuit that writes to the port overcomes the weak pullup. When software writes a 0 to any port pin, the DS89C420 activates a strong pulldown that remains on until either a 1 is written or a reset occurs. Writing a 1 after the port has been at 0 causes a strong transition driver to turn on, followed by a weaker sustaining pullup. Once the momentary strong driver turns off, the port again becomes both the output high and input state. As an alternate function, port 2 can function as the MSB of the external address bus when reading external program memory and read/write external RAM or peripherals. In page mode 1, port 2 provides both the MSB and LSB of the external address bus; in page mode 2, it provides the MSB and data.</td></tr>
<tr><td>22</td><td>25</td><td>19</td><td>P2.1 (A9)</td></tr>
<tr><td>23</td><td>26</td><td>20</td><td>P2.2 (A10)</td></tr>
<tr><td>24</td><td>27</td><td>21</td><td>P2.3 (A11)</td></tr>
<tr><td>25</td><td>28</td><td>22</td><td>P2.4 (A12)</td></tr>
<tr><td>26</td><td>29</td><td>23</td><td>P2.5 (A13)</td></tr>
<tr><td>27</td><td>30</td><td>24</td><td>P2.6 (A14)</td></tr>
<tr><td>28</td><td>31</td><td>25</td><td>P2.7 (A15)</td></tr>
<tr><td rowspan="2">10–17</td><td rowspan="2">11, 13–19</td><td rowspan="2">5, 7–13</td><td rowspan="2">P3.0–P3.7</td><td colspan="3">Port 3, I/O. Port 3 functions as both an 8-bit, bidirectional I/O port and an alternate functional interface for external interrupts, serial port 0, timer 0 and 1 inputs, and $\overline{RD}$ and $\overline{WR}$ strobes. The reset condition of port 3 is with all bits at logic 1. In this state, a weak pullup holds the port high. This condition also serves as an input mode, since any external circuit that writes to the port overcomes the weak pullup. When software writes a 0 to any port pin, the DS89C420 activates a strong pulldown that remains on until either a 1 is written or a reset occurs. Writing a 1 after the port has been at 0 causes a strong transition driver to turn on, followed by a weaker sustaining pullup. Once the momentary strong driver turns off, the port again becomes both the output high and input state. The alternate modes of Port 3 are outlined below.</td></tr>
<tr><td>PORT</td><td>ALTERNATE</td><td>FUNCTION</td></tr>
<tr><td>10</td><td>11</td><td>5</td><td>P3.0</td><td>P3.0</td><td>RXD0</td><td>Serial Port 0 Receive</td></tr>
<tr><td>11</td><td>13</td><td>7</td><td>P3.1</td><td>P3.1</td><td>TXD0</td><td>Serial Port 0 Transmit</td></tr>
<tr><td>12</td><td>14</td><td>8</td><td>P3.2</td><td>P3.2</td><td>$\overline{INT0}$</td><td>External Interrupt 0</td></tr>
<tr><td>13</td><td>15</td><td>9</td><td>P3.3</td><td>P3.3</td><td>$\overline{INT1}$</td><td>External Interrupt 1</td></tr>
<tr><td>14</td><td>16</td><td>10</td><td>P3.4</td><td>P3.4</td><td>T0</td><td>Timer 0 External Input</td></tr>
<tr><td>15</td><td>17</td><td>11</td><td>P3.5</td><td>P3.5</td><td>T1</td><td>Timer 1 External Input</td></tr>
<tr><td>16</td><td>18</td><td>12</td><td>P3.6</td><td>P3.6</td><td>$\overline{WR}$</td><td>External Data Memory Write Strobe</td></tr>
<tr><td>17</td><td>19</td><td>13</td><td>P3.7</td><td>P3.7</td><td>$\overline{RD}$</td><td>External Data Memory Read Strobe</td></tr>
<tr><td>31</td><td>35</td><td>29</td><td>$\overline{EA}$</td><td colspan="3">External Access. Allows selection of internal or external program memory. Connect to ground to force the DS89C420 to use an external memory-program memory. The internal RAM is still accessible as determined by register settings. Connect to V_{CC} to use internal flash memory.</td></tr>
</table>

Figure 5. Block Diagram



DETAILED DESCRIPTION

The DS89C420 is pin compatible with all three packages of the standard 8051 and includes standard resources such as three timer/counters, four 8-bit I/O ports, and a serial port. It features 16kB of in-system programmable flash memory, which can be programmed in-system from an I/O port using a built-in program memory loader. It can also be loaded externally using standard commercially available programmers.

Besides greater speed, the DS89C420 includes 1kB of data RAM, a second full-hardware serial port, seven additional interrupts, two more levels of interrupt priority, programmable watchdog timer, brownout monitor, and power-fail reset. The device also provides dual data pointers (DPTRs) to speed up block-data memory moves. This feature is further enhanced with a new selectable automatic increment/decrement and toggle-select operation. The speed of MOVX data memory access can be adjusted by adding stretch values up to 10 machine cycle times for flexibility in selecting external memory and peripherals.

A power management mode (PMM) significantly consumes less power by slowing the CPU execution rate from 1 clock period per cycle to 1024 clock periods per cycle. A selectable switchback feature can automatically cancel this mode to enable a normal speed response to interrupts.

The EMI reduction feature disables the ALE signal when the processor is not accessing external memory.

All standard SFR locations from the 8051 are duplicated in the DS89C420 and several SFRs have been added for the unique features of the DS89C420. Most of these features are controlled by bits in SFRs located in unused locations in the 8051 SFR map. This allows for increased functionality while maintaining complete instruction set compatibility. Table 1 summarizes the SFRs and their locations. Table 2 specifies the default reset condition for all SFR bits.

DATA POINTERS

The data pointers (DPTR and DPTR1) are used to assign a memory address for the MOVX instructions. This address can point to a MOVX RAM location (on-chip or off-chip), or a memory-mapped peripheral. Two pointers are useful when moving data from one memory area to another, or when using a memory-mapped peripheral for both source and destination addresses. The user selects the active pointer through a dedicated SFR bit (Sel = DPS.0), or activates an automatic toggling feature for altering the pointer selection (TSL = DPS.5). An additional feature, if selected, provides automatic incrementing or decrementing of the current DPTR.

STACK POINTER

The stack pointer denotes the register location at the top of the stack, which is the last used value. The user can place the stack anywhere in the scratchpad RAM by setting the stack pointer to the desired location, although the lower bytes are normally used for working registers.

I/O PORTS

The DS89C420 offers four 8-bit I/O ports. Each I/O port is represented by an SFR location, and can be written or read. The I/O port has a latch that contains the value written by software.

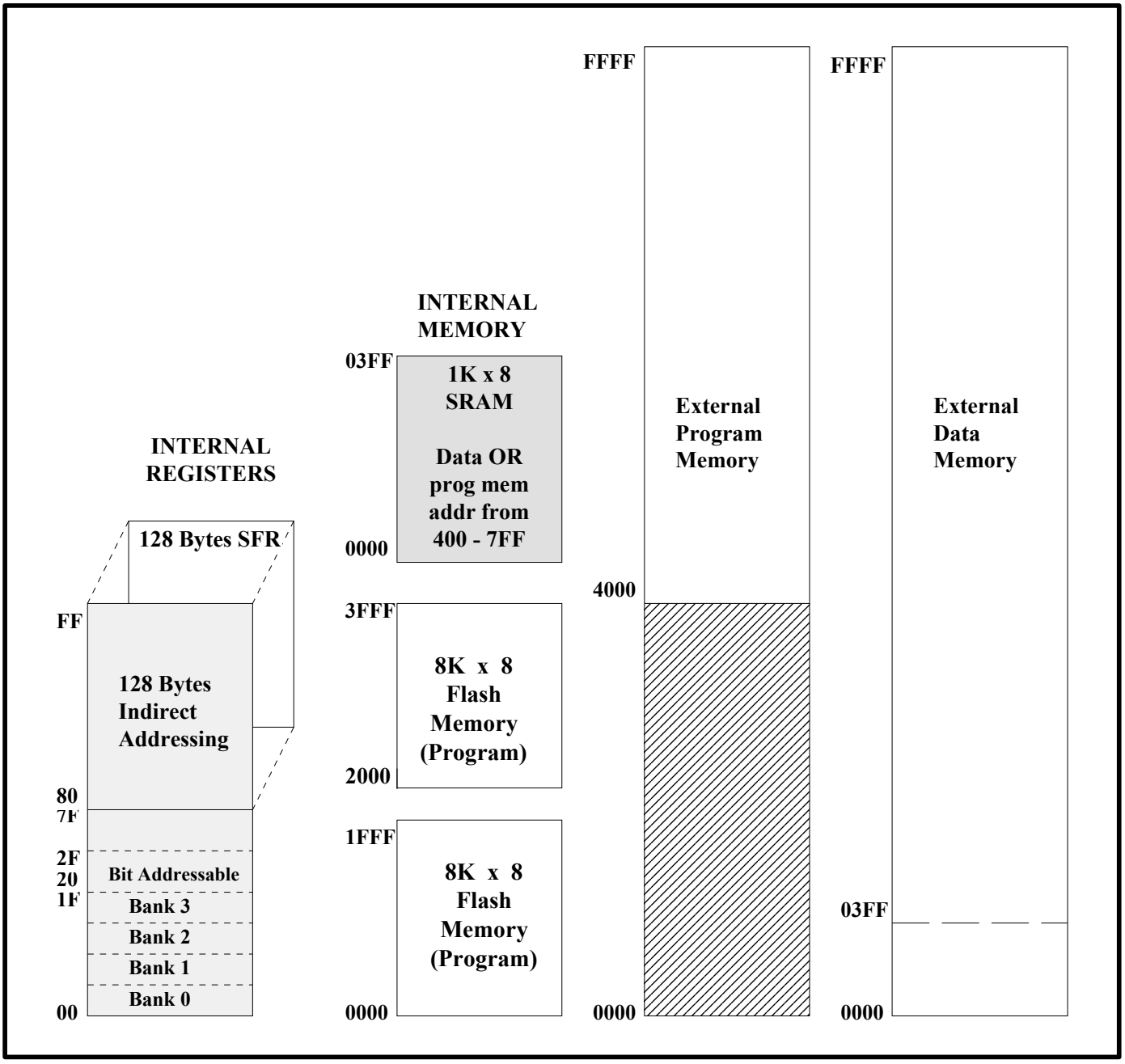
COUNTER/TIMERS

Three 16-bit timer/counters are available in the DS89C420. Each timer is contained in two SFR locations that can be read or written by software. The timers are controlled by other SFRs described in the “SFR Bit Description” section of the *Ultra-High-Speed Flash Microcontroller User’s Guide*.

SERIAL PORTS

The DS89C420 provides two UARTs that are controlled and accessed by SFRs. Each UART has an address that is used to read and write the UART. The same address is used for read and write operations, which are distinguished by the instruction. Its own SFR control register controls each UART.

Figure 6. Memory Map



The reset default condition is a maximum on-chip program-memory address of 16kB. When accessing external program memory, the first 16kB would be inaccessible. To select a smaller effective program memory size, software must alter bits RMS2–RMS0. Altering these bits requires a timed access procedure as explained later.

Care should be taken so that changing the ROMSIZE register does not corrupt program execution. For example, assume that a DS89C420 is executing instructions from internal program memory near the 12kB boundary (~3000h) and that the ROMSIZE register is currently configured for a 16kB internal program space. If software reconfigures the ROMSIZE register to 4kB (0000h–0FFFh) in the current state, the device immediately jumps to external program execution because program code from 4kB to 16kB (1000h–3FFFh) is no longer located on-chip. This could result in code misalignment and execution of an invalid instruction. The recommended method is to modify the ROMSIZE register from a location in memory that is internal (or external) both before and after the operation. In the above example, the instruction that modifies the ROMSIZE register should be located below the

Table 3. Flash Memory Lock Bits

LEVEL	LB1	LB2	LB3	PROTECTION
1	1	1	1	No program lock. Encrypted verify if encryption array is programmed.
2	0	1	1	Prevent MOVC in external memory from reading program code in internal memory. $\overline{EA}$ is sampled and latched on reset. Allow no further parallel or program memory loader programming.
3	X	0	1	Level 2 plus no verify operation. Also prevent MOVX in external memory from reading internal SRAM.
4	X	X	0	Level 3 plus no external execution.

The DS89C420 provides user-selectable options that must be set before beginning software execution. The option control register uses flash bits rather than SFRs, and is individually erasable and programmable as a byte-wide register. Bit 3 of this register is defined as the watchdog POR default. Setting this bit to 1 disables the watchdog-reset function on power-up, and clearing this bit to 0 enables the watchdog-reset function automatically. Other bits of this register are undefined and are at logic 1 when read. The value of this register can be read at address FCh in parallel programming mode or when executing a verify-option control-register instruction in ROM loader mode.

The signature bytes can be read in ROM loader mode or in parallel programming mode. Reading data from addresses 30h, 31h, and 60h provides signature information about manufacturer, part, and extension as follows:

ADDRESS VALUE	FUNCTION
30h DAh	Manufacturer ID
31h 42h	DS89C420 Device ID
60h 01h	Device Extension

ROM LOADER

The full 16kB of on-chip flash program-memory space, security flash block, and external SRAM can be programmed in-system from an external source through serial port 0 under the control of a built-in ROM loader. The ROM loader also has an auto-baud feature that determines which baud rate frequencies are being used for communication and sets up the baud rate generator for communication at that frequency.

When the DS89C420 is powered up and has entered its user operating mode, the ROM loader mode can be invoked at any time by forcing RST = 1, $\overline{EA}$ = 0, and $\overline{PSEN}$ = 0. It remains in effect until power-down or when the condition (RST = 1 and $\overline{PSEN}$ = $\overline{EA}$ = 0) is removed. Entering the ROM loader mode forces the processor to start fetching from the 2kB internal ROM for program memory initialization and other loader functions.

The read/write accessibility is determined by the state of the lock bits, which can be verified directly by the ROM loader. In the ROM loader mode, a mass-erase operation also erases the memory bank select and sets it to the default state. Otherwise, the memory bank select cannot be altered in the ROM loader mode.

Flash programming is executed by a series of internal flash commands that are derived (by the built-in ROM loader) from data transmitted over the serial interface from a host PC. PC-based software tools that configure and load the microcontrollers are available at www.maxim-ic.com/micros/ftpinfo.html.

Full details of the ROM loader software and its implementation are given in the *Ultra-High-Speed Flash Microcontroller User's Guide.*

Figure 9. Non-Page Mode, External Data-Memory Access (Stretch = 0, CD1:CD2 = 10)

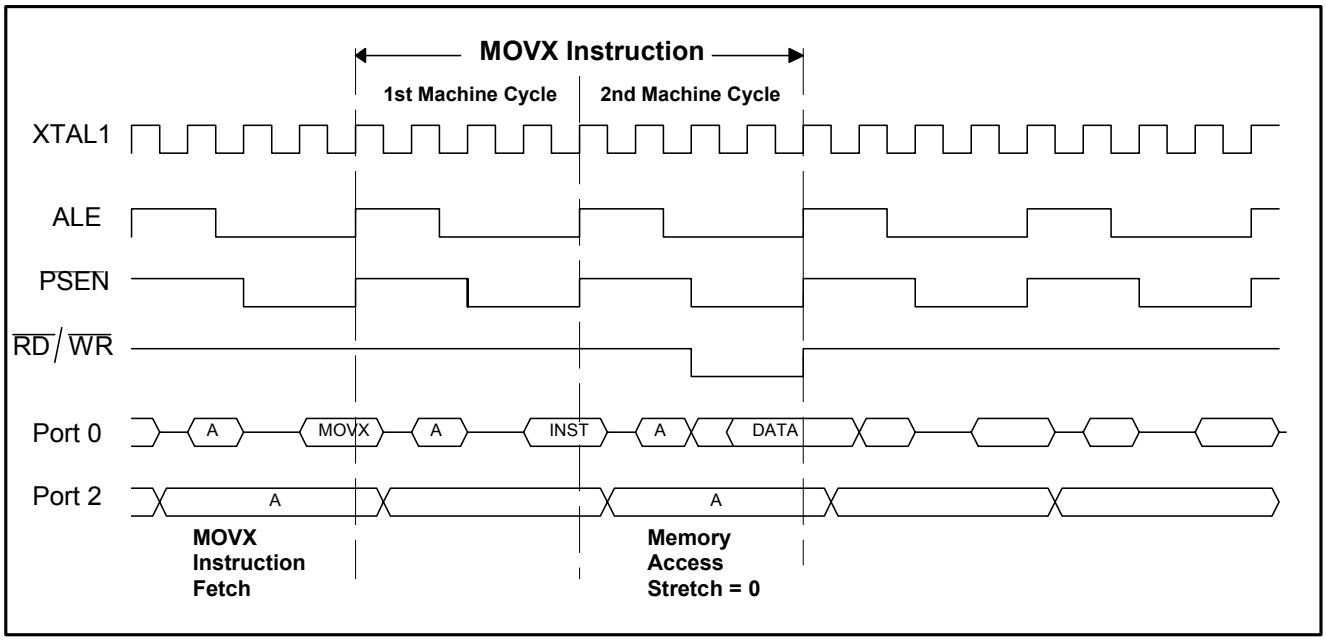
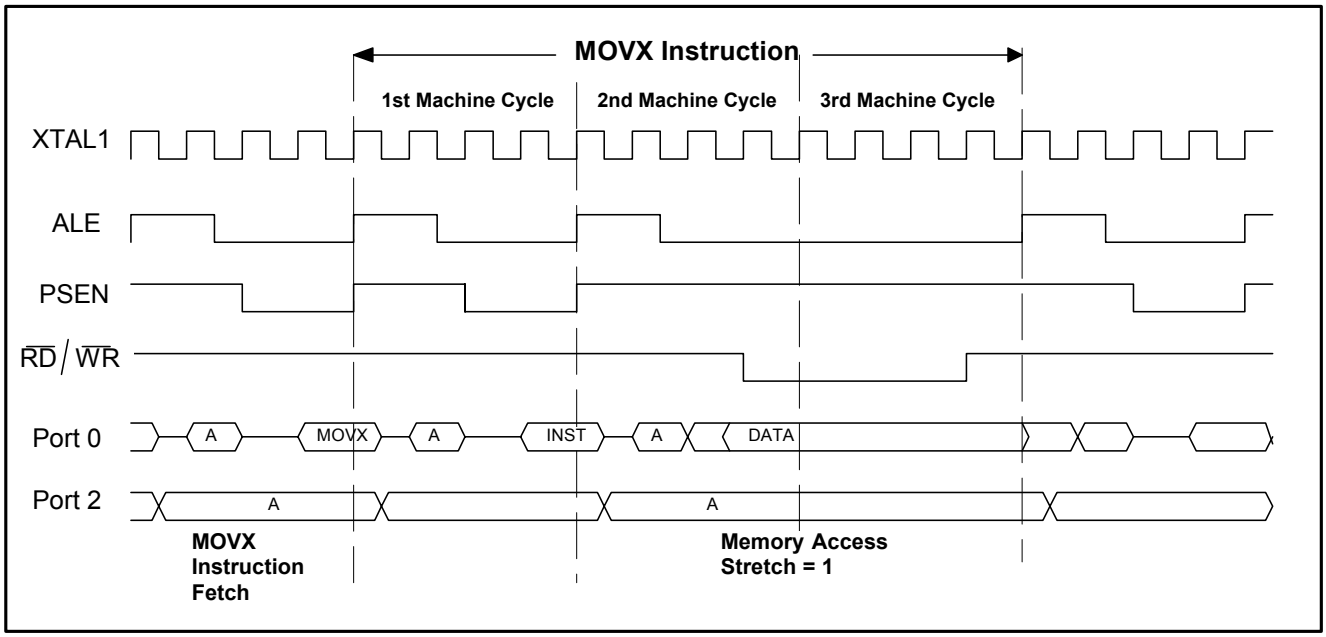


Figure 10. Non-Page Mode, External Data-Memory Access (Stretch = 1, CD1:CD2 = 10)



PAGE MODE, EXTERNAL MEMORY CYCLE

Page mode retains the basic circuitry requirement for original 8051 external memory interface, but alters the configuration of P0 and P2 for the purposes of address output and data I/O during external memory cycles. Additionally, the functions of ALE and $\overline{PSEN}$ are altered to support this mode of operation.

Setting the PAGEE (ACON.7) bit to logic 1 enables page mode. Clearing the PAGEE bit to a logic 0 disables the page mode and the external bus structure defaults to the original 8051 expanded bus configuration (non-page mode). The DS89C420 supports page mode in two external bus structures. The logic value of the page mode select bits in the ACON register determines the external bus structure and the basic memory cycle in the number of system clocks. Table 6 summarizes this option. The first three selections use the same bus structure but with a different memory cycle time. Setting the select bits to 11b selects another bus structure. Write access to the ACON register requires a timed access.

Table 6. Page Mode Select

PAGES1:PAGES0	CLOCKS PER MEMORY CYCLE		EXTERNAL BUS STRUCTURE
	PAGE HIT	PAGE MISS	
00	1	2	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of the address.
01	2	4	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of the address.
10	4	8	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of the address.
11	2	4	P0: Lower address byte. P2: The upper address byte is multiplexed with the data byte. **Note:** This setting affects external code fetches only; accessing the external data memory requires 4 clock cycles, regardless of page hit or miss.

The first page mode (page mode 1) external bus structure uses P2 as the primary address bus, (multiplexing both the most significant byte (MSB) and least significant byte (LSB) of the address for each external memory cycle) and P0 is used as the primary data bus. During external code fetches, P0 is held in a high-impedance state by the processor. Op codes are driven by the external memory onto P0 and latched at the end of the external fetch cycle at the rising edge of $\overline{PSEN}$. During external data read/write operations, P0 functions as the data I/O bus. It is held in a high-impedance state for external reads from data memory, and driven with data during external writes to data memory.

- A page miss occurs when the MSB of the subsequent address is different from the last address. The external memory machine cycle can be 2, 4, or 8 system clocks in length for a page miss.
- A page hit occurs when the MSB of the subsequent address does not change from the last address. The external memory machine cycle can be 1, 2, or 4 system clocks in length for a page hit.

During a page hit, P2 drives Addr0–7 of the 16-bit address while the most significant address byte is held in the external address latches. $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$ strobe accordingly for the appropriate operation on the P0 data bus. There is no ALE assertion for page hits.

During a page miss, P2 drives the Addr [8:15] of the 16-bit address and holds it for the duration of the first half of the memory cycle to allow the external address latches to latch the new most significant address byte. ALE is asserted to strobe the external address latches. During this operation, $\overline{PSEN}$, $\overline{RD}$, and $\overline{WR}$ are held in inactive states and P0 is in a high-impedance state. The second half of the memory cycle is executed as a page-hit cycle and the appropriate operation takes place.

A page miss can occur at set intervals or during external operations that require a memory access into a page of memory that has not been accessed during the last external cycle. Generally, the first external memory access causes a page miss. The new page address is stored internally, and is used to detect a page miss for the current external memory cycle.

Note that there are a few exceptions for this mode of operation when PAGES1 and PAGES2 are set to 00b:

- $\overline{\text{PSEN}}$ is asserted for both page hit and page miss for a full clock cycle.
- The execution of external MOVX instruction causes a page miss.
- A page miss occurs when fetching the next external instruction following the execution of an external MOVX instruction.

Figure 11 shows the external memory cycle for this bus structure. The first case illustrates a back-to-back execution sequence for 1-cycle page mode (PAGES1 = PAGES0 = 0b). $\overline{\text{PSEN}}$ remains active during page-hit cycles, and page misses are forced during and after MOVX executions, independent of the most significant byte of the subsequent addresses. The second case illustrates a MOVX execution sequence for 2-cycle page mode (PAGES1 = 0 and PAGES0 = 1). $\overline{\text{PSEN}}$ is active for a full clock cycle in code fetches. Note that changing the MSB of the data address causes the page misses in this sequence. The third case illustrates a MOVX execution sequence for 4-cycle page mode (PAGES1 = 1 and PAGES0 = 0). There is no page miss in this execution cycle because the most significant byte of the data address is assumed to match the last program address.

The second page mode (page mode 2) external bus structure multiplexes the most significant address byte with data on P2, and uses P0 for the least significant address byte. This bus structure is used to speed up external code fetches only. External data-memory access cycles are identical to the non-page mode except for the different signals on P0 and P2. Figure 12 illustrates the memory cycle for external code fetches.

Figure 11. Page Mode 1, External Memory Cycle (CD1:CD0 = 10)

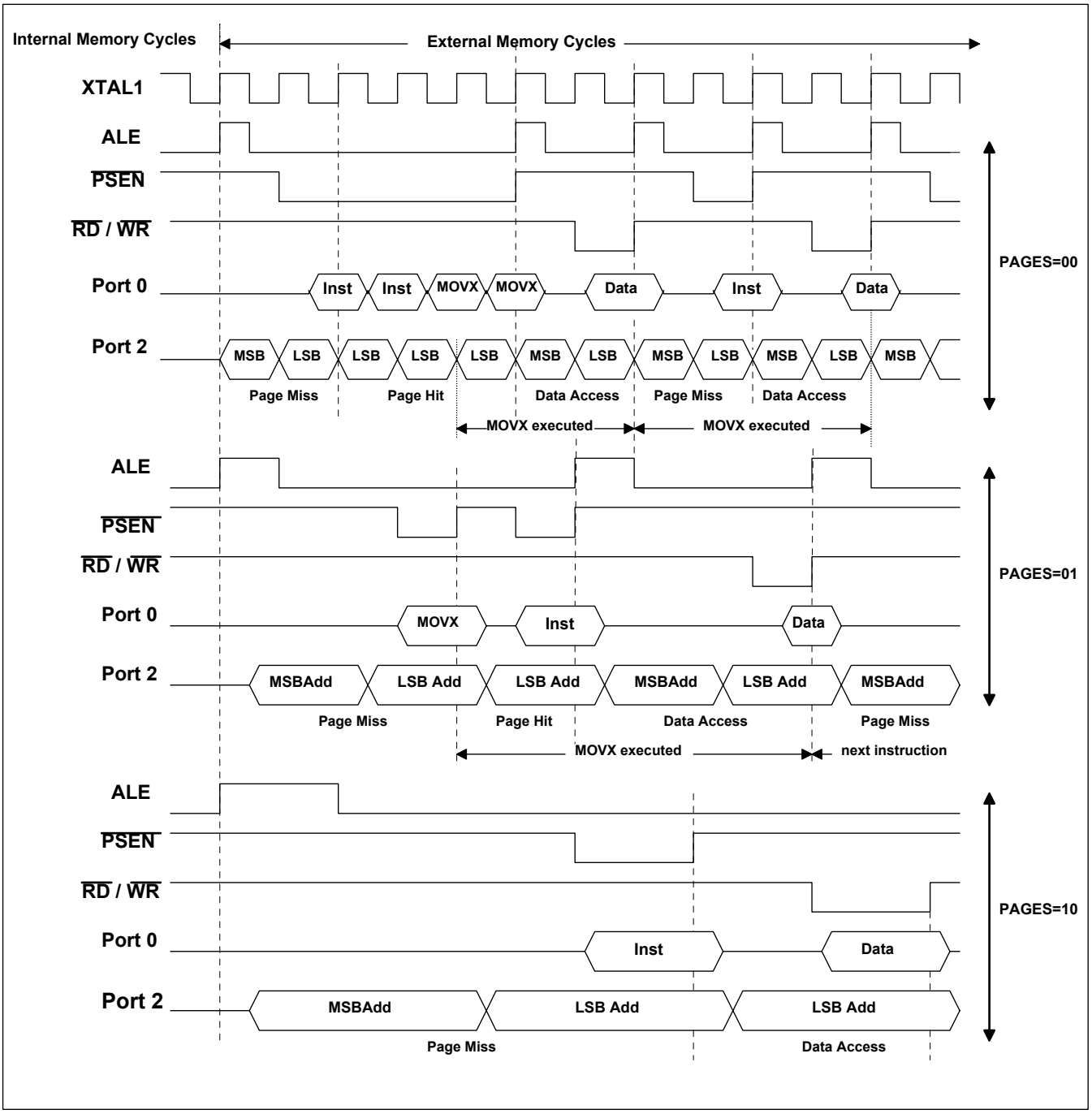
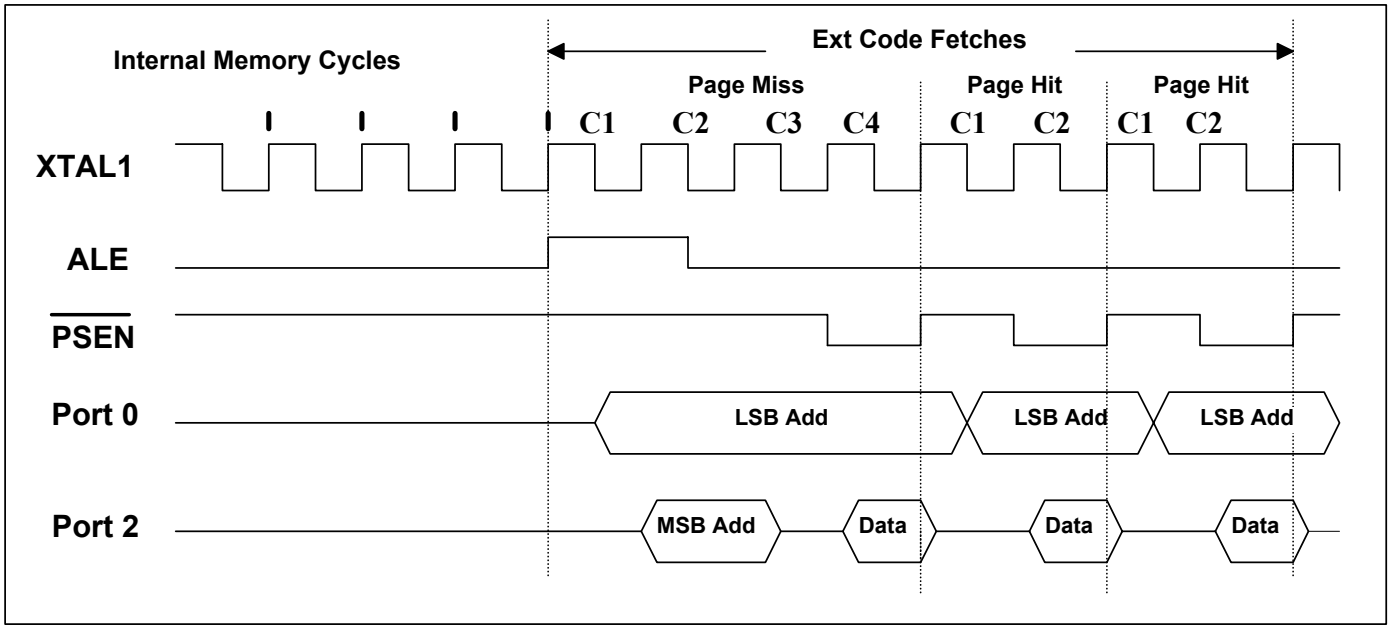


Figure 12. Page Mode 2, External Code Fetch Cycle (CD1:CD0 = 10)



STRETCH EXTERNAL DATA MEMORY CYCLE IN PAGE MODE

The DS89C420 allows software to adjust the speed of external data memory access by stretching the memory bus cycle in page mode operation just like non-page mode operation. The following tables summarize the stretch values and their effects on the external MOVX-memory bus cycle and the control signals' pulse width in terms of the number of oscillator clocks. A stretch machine cycle always contains four system clocks, independent of the logic value of the page mode select bits.

Table 7. Page Mode 1, Data Memory Cycle Stretch Values (Pages1:Pages0 = 00)

MD2:MD0	STRETCH CYCLES	$\overline{RD}/\overline{WR}$ PULSE WIDTH (IN NUMBER OF OSCILLATOR CLOCKS)			
		4X/$\overline{2X}$, CD1, CD0 = 100	4X/$\overline{2X}$, CD1, CD0 = 000	4X/$\overline{2X}$, CD1, CD0 = X10	4X/$\overline{2X}$, CD1, CD0 = X11
000	0	0.25	0.5	1	1024
001	1	0.75	1.5	3	3072
010	2	1.75	3.5	7	7168
011	3	2.75	5.5	11	11,264
100	7	3.75	7.5	15	15,360
101	8	4.75	9.5	19	19,456
110	9	5.75	11.5	23	23,552
111	10	6.75	13.5	27	27,648

Table 8. Page Mode 1, Data Memory Cycle Stretch Values (Pages1:Pages0 = 01)

MD2:MD0	STRETCH CYCLES	$\overline{RD}/\overline{WR}$ PULSE WIDTH (IN NUMBER OF OSCILLATOR CLOCKS)			
		4X/$\overline{2X}$, CD1, CD0 = 100	4X/$\overline{2X}$, CD1, CD0 = 000	4X/$\overline{2X}$, CD1, CD0 = X10	4X/$\overline{2X}$, CD1, CD0 = X11
000	0	0.25	0.5	1	1024
001	1	0.75	1.5	3	3072
010	2	1.75	3.5	7	7168
011	3	2.75	5.5	11	11,264
100	7	3.75	7.5	15	15,360
101	8	4.75	9.5	19	19,456
110	9	5.75	11.5	23	23,552
111	10	6.75	13.5	27	27,648

Table 9. Page Mode 1, Data Memory Cycle Stretch Values (Pages1:Pages0 = 10)

MD2:MD0	STRETCH CYCLES	$\overline{RD}/\overline{WR}$ PULSE WIDTH (IN NUMBER OF OSCILLATOR CLOCKS)			
		4X/$\overline{2X}$, CD1, CD0 = 100	4X/$\overline{2X}$, CD1, CD0 = 000	4X/$\overline{2X}$, CD1, CD0 = X10	4X/$\overline{2X}$, CD1, CD0 = X11
000	0	0.5	1	2	2048
001	1	1	2	4	4096
010	2	2	4	8	8192
011	3	3	6	12	12,288
100	7	4	8	16	16,384
101	8	5	10	20	20,480
110	9	6	12	24	24,576
111	10	7	14	28	28,672

Table 10. Page Mode 2, Data Memory Cycle Stretch Values (Pages1:Pages0 = 11)

MD2:MD0	STRETCH CYCLES	$\overline{RD}/\overline{WR}$ PULSE WIDTH (IN NUMBER OF OSCILLATOR CLOCKS)			
		4X/$\overline{2X}$, CD1, CD0 = 100	4X/$\overline{2X}$, CD1, CD0 = 000	4X/$\overline{2X}$, CD1, CD0 = X10	4X/$\overline{2X}$, CD1, CD0 = X11
000	0	0.5	1	2	2048
001	1	1	2	4	4096
010	2	2	4	8	8192
011	3	3	6	12	12,288
100	7	4	8	16	16,384
101	8	5	10	20	20,480
110	9	6	12	24	24,576
111	10	7	14	28	28,672

As shown in the previous tables, the stretch feature supports eight stretched external data-memory access cycles that can be categorized into three timing groups. When the stretch value is cleared to 000b, there is no stretch on external data-memory access and a MOVX instruction is completed in two basic memory cycles. When the stretch value is set to 1, 2, or 3, the external data memory access is extended by 1, 2, or 3 stretch memory cycles, respectively. Note that the first stretch value does not result in adding four system clocks to the control signals. This is because the first stretch uses one system clock to create additional address setup and data bus float time, and one system clock to create additional address and data hold time. When using very slow RAM and peripherals, a larger stretch value (4–7) can be selected. In this stretch category, two stretch cycles are used to create additional setup (the ALE pulse width is also stretched by one stretch cycle for page miss) and one stretch cycle is used to create additional hold time. The following timing diagrams illustrate the external data-memory access at divide-by-1 system clock mode (CD1:CD0 = 10b).

Figure 13. Page Mode 1, External Data Memory Access (Pages = 01, Stretch = 1, CD = 10)

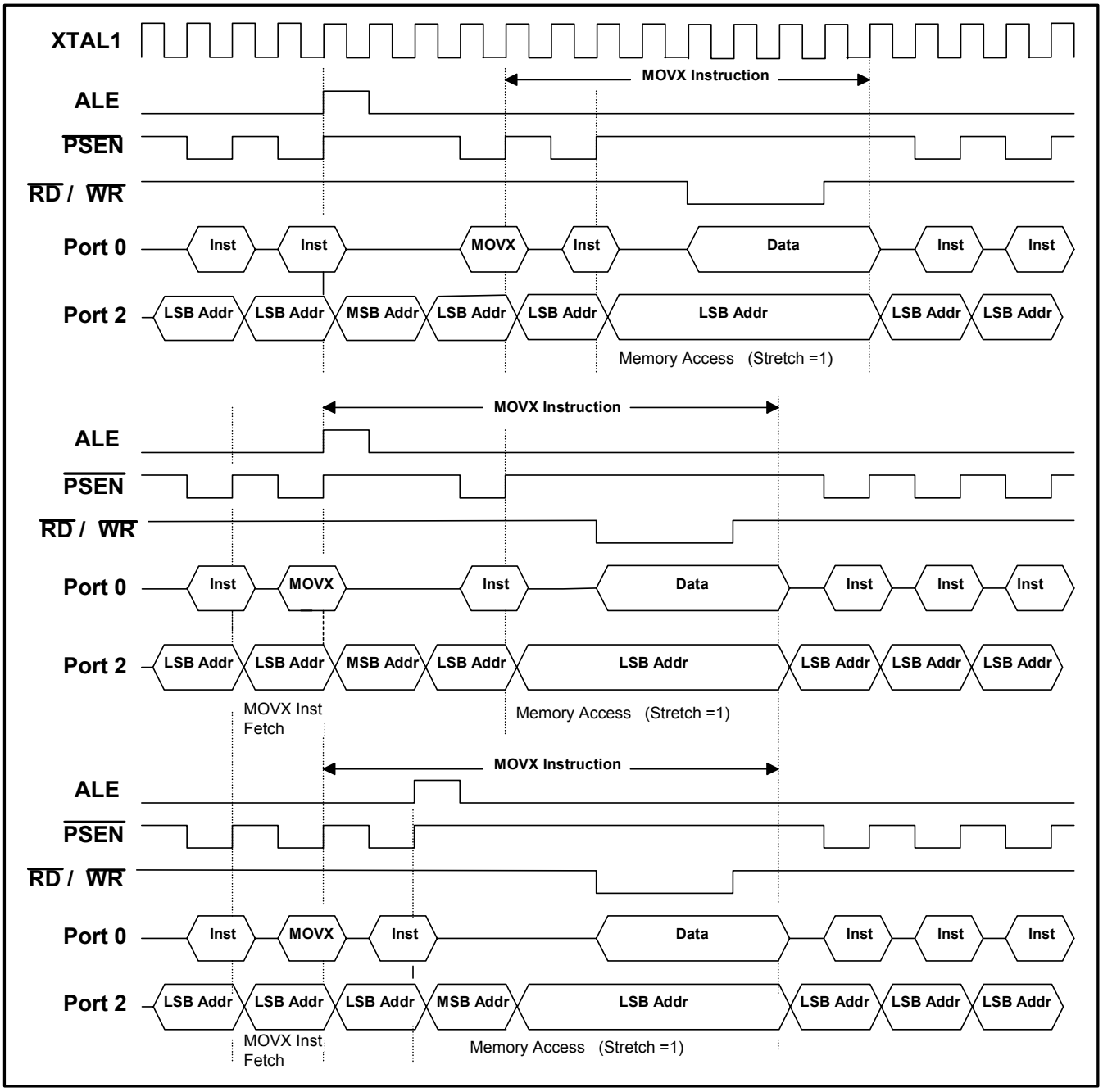


Figure 13 illustrates the external data-memory stretch cycle timing relationship when PAGEE = 1 and PAGES1:PAGES0 = 01. The stretch cycle shown is for a stretch value of 1 and is coincident with a page miss. Note that the first stretch value does not result in adding four system clocks to the $\overline{RD}/\overline{WR}$ control signals. This is because the first stretch uses one system clock to create additional setup and one system clock to create additional hold time.

timer/counters without auto-reload. Each timer can also be used as a counter of external pulses on the corresponding T0/T1 pin for 1-to-0 transitions. The timer mode (TMOD) register controls the operation mode. Each timer consists of a 16-bit register in 2 bytes, which can be found in the SFR map as TL0, TH0, TL1, and TH1. The timer control (TCON) register enables Timers 0 and 1.

Table 12. Timer Functions

FUNCTIONS	TIMER 0	TIMER 1	TIMER 2
Timer/Counter	13/16/8*/2 x 8 bit	13/16/8* bit	16 bit
Timer with Capture	No	No	Yes
External Control-Pulse Counter	Yes	Yes	No
Up/Down Auto-Reload Timer/Counter	No	No	Yes
Baud Rate Generator	No	Yes	Yes
Timer-Output Clock Generator	No	No	Yes

**8-bit timer/counter includes auto-reload feature; 2- x 8-bit mode does not.*

Timer 2 is a true 16-bit timer/counter that, with a 16-bit capture (RCAP2L and RCAP2H) register, is able to provide some unique functions like up/down auto-reload timer/counter and timer-output clock generation. Timer 2 (registers TL2 and TH2) is enabled by the T2CON register, and its mode of operation is selected by the T2MOD register.

Each timer has a selectable time base (Table 14). Following a reset, the timers default to divide-by-12 to maintain drop-in compatible with the 8051. If Timer 2 is used as a baud rate generator or clock output, its time base is fixed at divide by 2, regardless of the setting of its timer mode bits.

For details of operation, refer to “Programmable Timers” in the *Ultra-High-Speed Flash Microcontroller User’s Guide*.

TIMED ACCESS

The timed access function provides control verification to system functions. The timed access function prevents an errant CPU from making accidental changes to certain SFR bits that are considered vital to proper system operation. This is achieved by using software control when accessing the following SFR control bits:

WDCON.0	RWT	Reset Watchdog Timer
WDCON.1	EWT	Watchdog Reset Enable
WDCON.3	WDIF	Watchdog Interrupt Flag
WDCON.6	POR	Power-On Reset Flag
EXIF.0	BGS	Bandgap Select
ACON.5	PAGES0	Page Mode Select Bit 0
ACON.6	PAGES1	Page Mode Select Bit 1
ACON.7	PAGEE	Page Mode Enable
ROMSIZE.0	RMS0	Program Memory Size Select Bit 0
ROMSIZE.1	RMS1	Program Memory Size Select Bit 1
ROMSIZE.2	RMS2	Program Memory Size Select Bit 2
ROMSIZE.3	PRAME	Program RAM Enable
FCNTL.0	FC0	Flash Command Bit 0
FCNTL.1	FC1	Flash Command Bit 1
FCNTL.2	FC2	Flash Command Bit 2
FCNTL.3	FC3	Flash Command Bit 3

WATCHDOG TIMER

The watchdog timer functions as the source of both the watchdog interrupt and the watchdog reset. When the clock divider is set to 10b, the interrupt timeout has a default divide ratio of 2^{17} of the crystal oscillator clock, with the watchdog reset set to timeout 512 system clock cycles later. This results in a 33MHz crystal oscillator producing an interrupt timeout every 3.9718ms, followed 15.5µs later by a watchdog reset. The watchdog timer is reset to the default divide ratio following any reset. Using the WD0 and WD1 bits in the clock control (CKCON.6 and 7) register, other divide ratios can be selected for longer watchdog interrupt periods. Table 13 summarizes the watchdog bit settings and the timeout values.

Note: All watchdog-timer reset timeouts follow the programmed interrupt timeouts by 512 system clock cycles, which equates to varying numbers of oscillator cycles depending on the clock-divide (CD1:0) and crystal multiplier settings.

Table 13. Watchdog Timeout Value (in Number of Oscillator Clocks)

4X/$\overline{2X}$	CD1:0	WATCHDOG INTERRUPT TIMEOUT				WATCHDOG RESET TIMEOUT			
		WD1:0 = 00	WD1:0 = 01	WD1:0 = 10	WD1:0 = 11	WD1:0 = 00	WD1:0 = 01	WD1:0 = 10	WD1:0 = 11
1	00	2^{15}	2^{18}	2^{21}	2^{24}	2^{15} + 128	2^{18} + 128	2^{21} + 128	2^{24} + 128
0	00	2^{16}	2^{19}	2^{22}	2^{25}	2^{16} + 256	2^{19} + 256	2^{22} + 256	2^{25} + 256
x	01	2^{17}	2^{20}	2^{23}	2^{26}	2^{17} + 512	2^{20} + 512	2^{23} + 512	2^{26} + 512
x	10	2^{17}	2^{20}	2^{23}	2^{26}	2^{17} + 512	2^{20} + 512	2^{23} + 512	2^{26} + 512
x	11	2^{27}	2^{30}	2^{33}	2^{36}	2^{27} + 524,288	2^{30} + 524,288	2^{33}+ 524,288	2^{36} + 524,288

A watchdog control (WDCON) SFR is used for programming the functions. EWT (WDCON.1) is the enable for the watchdog-timer reset function and RWT (WDCON.0) is the bit used to restart the watchdog timer. Setting the RWT bit restarts the timer for another full interval. If the watchdog timer reset function is masked by the EWT bit and no resets are issued to the timer through the RWT bit, the watchdog timer generates interrupt timeouts at a rate determined by the programmed divide ratio. WDIF (WDCON.3) is the interrupt flag set at timer termination and WTRF (WDCON.2) is the reset flag set following a watchdog-reset timeout. The watchdog interrupt is enabled by the EWDI bit (EIE.4) when it is set to 1. The watchdog timer reset and interrupt timeouts are measured by counting system clock cycles.

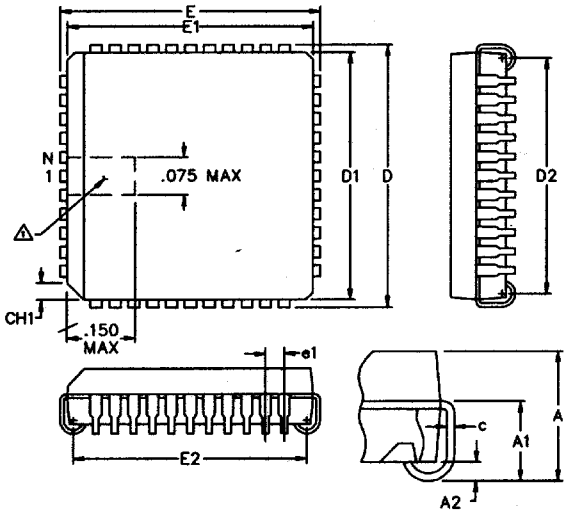
An independent watchdog timer functions as the crystal startup counter to count 65,536 crystal clock cycles before allowing the crystal oscillator to function as the system clock. This warmup time is verified by the watchdog timer following each power-up as well as each time the crystal is restarted following a stop mode. The watchdog is also used to establish a startup time whenever the CTM in the PMR register is set to enable the crystal multiplier (4X/$\overline{2X}$).

One of the applications of the watchdog timer is for the watchdog to wake up the system from idle mode. The watchdog interrupt can be programmed to allow a system to wake up periodically to sample the external world.

EXTERNAL RESET

If the RST input is taken to a logic 1, the device is forced into a reset state. An external reset is accomplished by holding the RST pin high for at least 3 clock cycles while the oscillator is running. Once the reset state is invoked, it is maintained as long as RST is pulled to logic 1. When the RST is removed, the processor exits the reset state within 4 clock cycles and begins execution at address 0000h. If a RST is applied while the processor is in stop mode, the RST causes the oscillator to begin running and forces the program counter to 0000h. There is a reset delay of 65,536 clock cycles to allow the oscillator to stabilize.

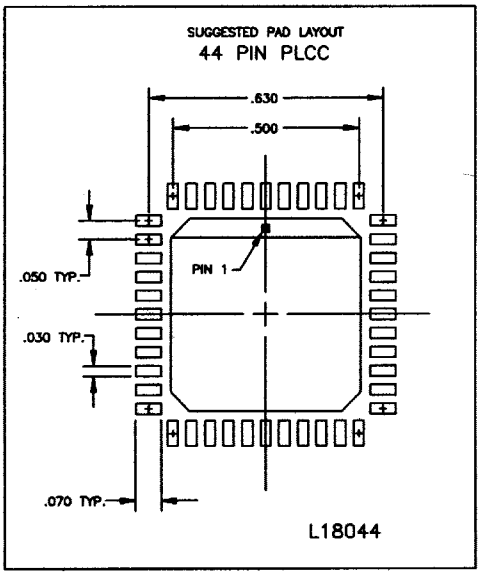
The RST pin is a bidirectional I/O. If a reset is caused by a power-fail reset, a watchdog timer reset, or an internal system reset, an output-reset pulse is also generated at the RST pin. This reset pulse is asserted as long as an internal reset is asserted and may not be able to drive the reset signal out if the RST pin is connected to an RC circuit. Connecting the RST pin to a capacitor does not affect the internal reset condition.



Note 1: Pin 1 identifier to be located in zone indicated.
Note 2: Controlling dimensions are in inches (in).

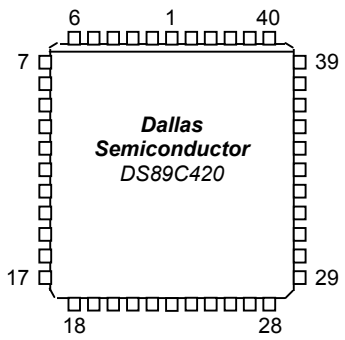
PKG	44–PIN	
DIM	MIN	MAX
A	0.165	0.180
A1	0.090	0.120
A2	0.020	–
B	0.026	0.033
B1	0.013	0.021
c	0.009	0.012
CH1	0.042	0.048
D	0.685	0.695
D1	0.650	0.656
D2	0.590	0.630
E	0.685	0.695
E1	0.650	0.656
E2	0.590	0.630
e1	0.050 BSC	
N	44	–

56–G4003–001



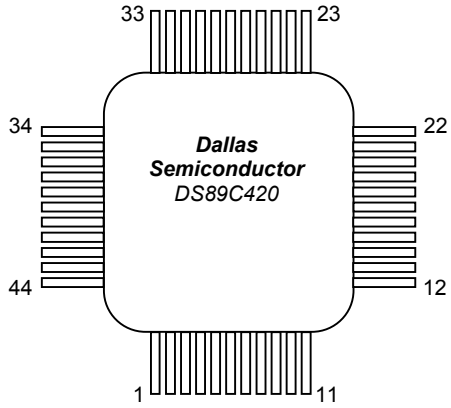
PIN CONFIGURATIONS

TOP VIEW



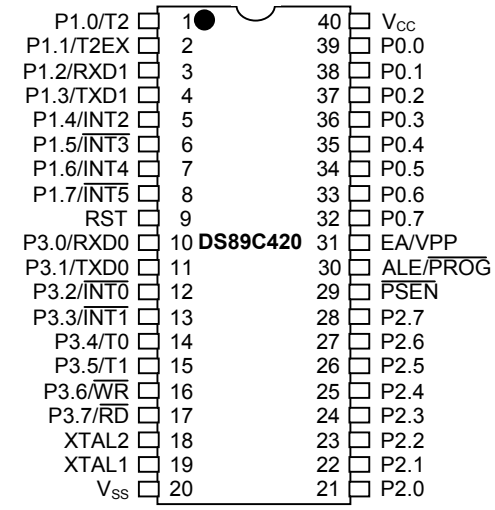
PLCC

TOP VIEW



TQFP

TOP VIEW



DIP