PRELIMINARY

DALLAS SEMICONDUCTOR

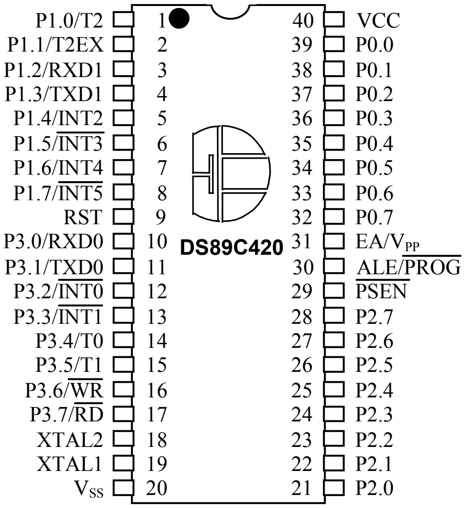
DS89C420
Ultra High-Speed Microcontroller

www.maxim-ic.com

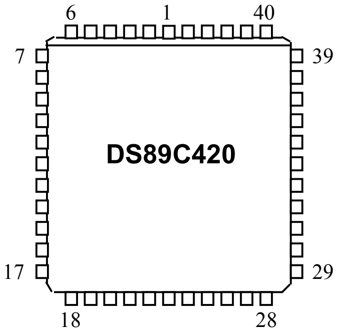
FEATURES

- 80C52 compatible
  - 8051 pin and instruction set compatible
  - Four bidirectional I/O ports
  - Three 16 bit timer counters
  - 256 bytes scratchpad RAM
- On-chip memory
  - 16k Flash memory
  - In-Application programmable
  - In System programmable via serial port
  - 1k SRAM for MOVX
- ROMSIZE feature
  - Selects internal program memory size from 0 to16k
  - Allows access to entire external memory map
  - Dynamically adjustable by software
- High Speed Architecture
  - One clock per machine cycle
  - DC to 50 MHz operation
  - Single cycle instruction in 20 ns
  - Optional variable length MOVX to access fast/slow peripherals
  - Dual data pointers with Auto Increment/Decrement and Toggle Select
  - Supports four paged modes
- Power Management Mode
  - Programmable clock divider
  - Automatic hardware and software exit
- Two full duplex serial ports
- Programmable Watchdog timer
- Thirteen interrupt sources (six external)
- Five levels of interrupt priority
- Power Fail Reset
- Early Warning Power Fail Interrupt

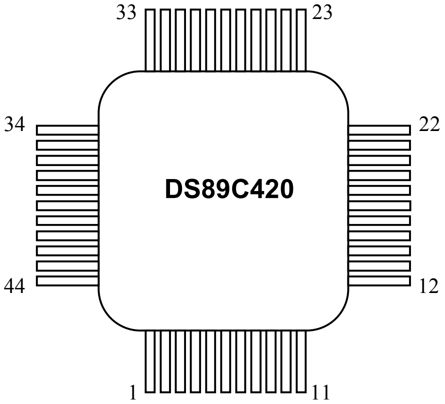
PIN ASSIGNMENT



40-Pin DIP



44-Pin PLCC



44-Pin TQFP

Note: Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, click here: http://dbserv.maxim-ic.com/errata.cfm.

DESCRIPTION

The DS89C420 offers the highest performance available in 8051-compatible microcontrollers. It features a redesigned processor core which executes every 8051 instruction up to 12 times faster than the original for the same crystal speed, thus allowing very significant improvements using the same code and crystal. The DS89C420 offers a maximum crystal speed of 50 MHz, achieving execution rates up to 50 MIPs for short instructions.

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

Besides greater speed, the DS89C420 includes 1 kbits of data RAM, a second full hardware serial port, seven additional interrupts, two more levels of interrupt priority, programmable Watchdog timer, Brown-out Monitor, and Power-Fail Reset. The device also provides dual data pointers (DPTRs) to speed-up block data memory moves and 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 ten machine cycle times for flexibility in selecting external memory and peripherals.

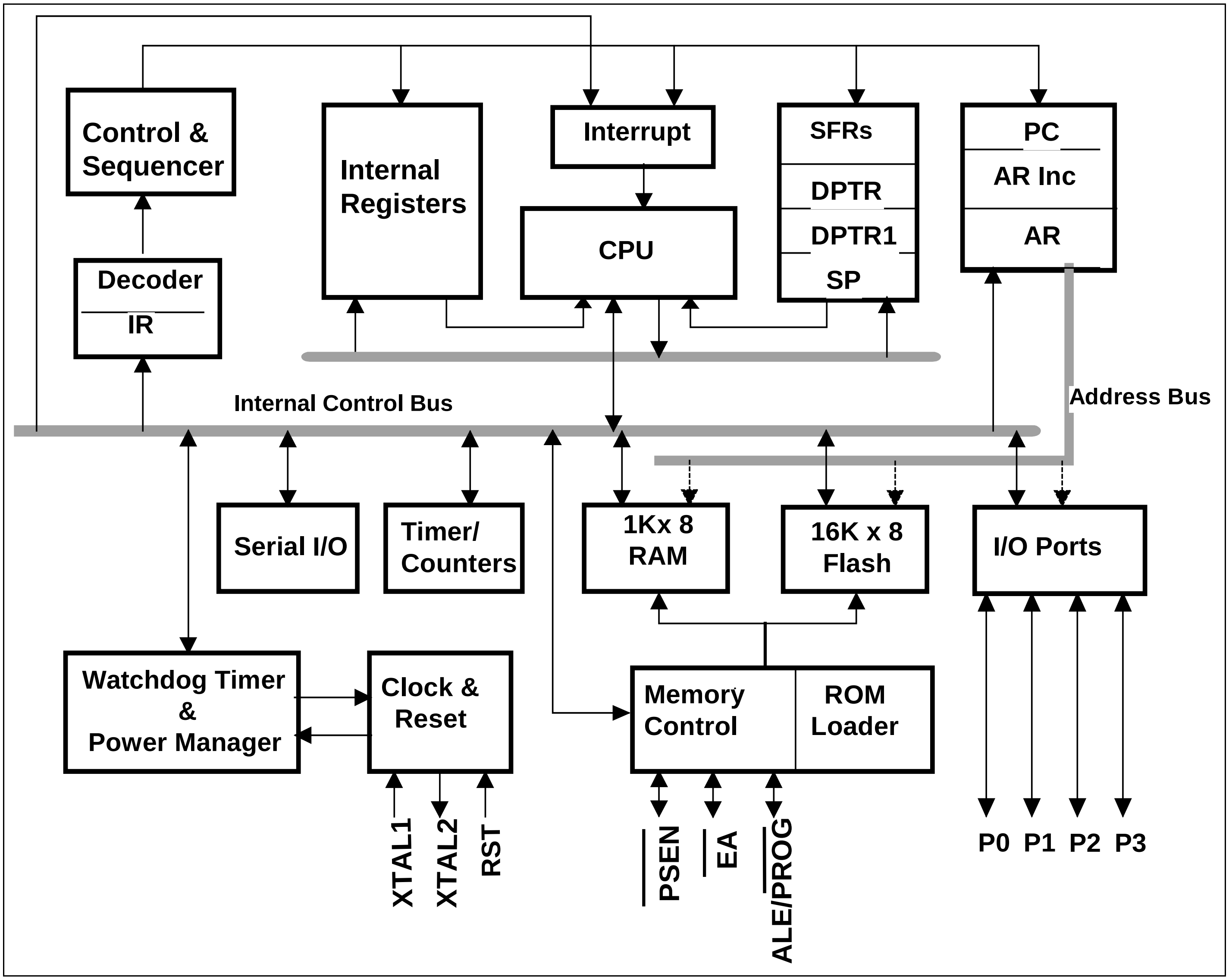
A Power Management Mode (PMM) uses significantly lower power consumption by slowing the CPU execution rate from 1 clock period per cycle to 1024 clock periods per cycle. A selectable switchback feature can automatically cancel this mode to enable a normal speed response to interrupts.

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

ORDERING INFORMATION:

PART NUMBER	PACKAGE	MAX. CLOCK SPEED	TEMPERATURE RANGE
DS89C420 - MCS	40-pin plastic DIP	50 MHz	0^{O}C to+70^{O}C
DS89C420 - QCS	44-pin PLCC	50 MHz	0^{O}C to+70^{O}C
DS89C420 - ECS	44-pin TQFP	50 MHz	0^{O}C to+70^{O}C
DS89C420 - MNS	40-pin plastic DIP	50 MHz	-40^{O}C to $+85^{O}$C
DS89C420 - QNS	44-pin PLCC	50 MHz	-40^{O}C to $+85^{O}$C
DS89C420 - ENS	44-pin TQFP	50 MHz	-40^{O}C to $+85^{O}$C

DS89C420 BLOCK DIAGRAM Figure 1



PIN DESCRIPTION Table 1

DIP	PLCC	TQFP	SIGNAL NAME	DESCRIPTION
40	12, 44	6, 38	V_{CC}	**V_{CC}** - +5V
20	1 , 22, 23, 34	16, 17, 28, 39	GND	**GND** – Logic Ground.
9	10	4	RST	**External Reset**. The RST input pin is bi-directional 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 OR'd external reset sources. An RC is not required for power-up, as the device provides this function internally.
19 18	21 20	15 14	XTAL1 XTAL2	**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.
29	32	26	$\overline{\text{PSEN}}$	**Program Store Enable output**. This signal is commonly connected to optional external program memory memory as a chip enable. $\overline{\text{PSEN}}$ provides an active low pulse and is driven high when external program memory is not being accessed.
30	33	27	ALE/ $\overline{\text{PROG}}$	**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. ALE is forced 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 ($\overline{\text{PROG}}$) is used to execute the parallel program function
39 38 37 36 35 34 33 32	43 42 41 40 39 38 37 36	37 36 35 34 33 32 31 30	P0.0 (AD0) P0.1 (AD1) P0.2 (AD2) P0.3 (AD3) P0.4 (AD4) P0.5 (AD5) P0.6 (AD6) P0.7 (AD7)	**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 pull-ups for logic 1 outputs. The reset condition of Port 0 is tri-state. Pullup resistors are required when using Port 0 as an I/O port.
1-8	2-9	40-44	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 a 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 will overcome

DIP	PLCC	TQFP	SIGNAL NAME	DESCRIPTION
				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	P1.0 T2 External I/O for timer/Counter2
2	3	41	P1.1	P1.1 T2EX Timer 2 Capture/Reload Trigger
3	4	42	P1.2	P1.2 RXD1 Serial Port 1 Receive Input
4	5	43	P1.3	P1.3 TXD1 Serial Port 1 Transmit Output
5	6	44	P1.4	P1.4 INT2 External Interrupt 2 (Positive Edge Detect)
6	7	1	P1.5	P1.5 $\overline{\text{INT3}}$ External Interrupt 3 (NegativeEdge Detect)
7	8	2	P1.6	P1.6 INT4 External Interrupt 4 (Positive Edge Detect)
8	9	3	P1.7	P1.7 $\overline{\text{INT5}}$ External Interrupt 5 (NegativeEdge Detect)

21
22
23
24
25
26
27
28
	24
	25
	26
	27
	28
	29
	30
	31
		18
		19
		20
		21
		22
		23
		24
		25
			P2.0 (A8)
			P2.1 (A9)
			P2.2(A10)
			P2.3(A11)
			P2.4(A12)
			P2.5(A13)
			P2.6(A14)
			P2.7(A15)
				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 will overcome 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 and in Page Mode 2, it provides the MSB and data.

DIP	PLCC	TQFP	SIGNAL NAME	DESCRIPTION
10-17	11,13-19	5, 7-13	P3.0-P3.7	Port 3 - I/O. Port 3 functions as both an 8-bit bi-directional 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 a 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 will overcome 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 :
				Port Alternate Function
10	11	5		P3.0 RXD0 Serial Port 0 Receive Input
11	13	7		P3.1 TXD0 Serial Port 0 Transmit Output
12	14	8		P3.2 $\overline{INT0}$ External Interrupt 0
13	15	9		P3.3 $\overline{INT1}$ External Interrupt 1
14	16	10		P3.4 T0 timer 0 External Input
15	17	11		P3.5 T1 timer 1 External Input
16	18	12		P3.6 $\overline{WR}$ External Data Memory Write Strobe
17	19	13		P3.7 $\overline{RD}$ External Data Memory Read Strobe
31	35	29	$\overline{EA}$	**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.

COMPATIBILITY

The DS89C420 is a fully static CMOS 8051-compatible microcontroller similar in functional features to the DS87C520 but with much higher performance. In most cases the DS89C420 can drop into an existing socket for the 8xc51 family to improve the operation significantly. While remaining familiar to 8051 family users, it has many new features. The DS89C420 runs the standard 8051 family instruction set and is pin-compatible with DIP, PLCC or TQFP packages. In general, software written for existing 8051 based systems works without modification on the DS89C420 with the exception of critical timing routines since the DS89C420 performs its instructions much faster than the original for any given crystal selection.

The DS89C420 provides three 16–bit timer/counters, full–duplex serial port (2), 256 bytes of direct RAM plus 1 kbits of extra MOVX RAM. I/O ports can operate as in standard 8051 products. Timers will default to a 12 clocks per cycle operation to keep their timing compatible with original 8051 family systems. However, timers are individually programmable to run at the new 1 clock per cycle if desired. The DS89C420 provides several new hardware features implemented by new SFRs.

PERFORMANCE OVERVIEW

The DS89C420 features a completely redesigned high-speed 8051 compatible core and allows operation at a higher clock frequency. This updated core does not have the dummy memory cycles that are present in a standard 8051. A conventional 8051 generates machine cycles using the clock frequency divided by 12. In the DS89C420, the same machine cycle takes one clock. Thus the fastest instructions execute twelve times faster for the same crystal frequency. It should be noted that this speed improvement will be reduced when using external memory access modes requiring more than one clock per cycle.

Improvement of individual programs will depend on the actual instructions used. Speed sensitive applications would make the most use of instructions that are 12 times faster. However, the sheer number of 12-to-1 improved opcodes makes dramatic speed improvements likely for any code. These architecture improvements produce instruction cycle times as low as 20 ns (50 MIPs). The Dual Data Pointer feature also allows the user to eliminate wasted instructions when moving blocks of memory. The new Page Modes allow for increased efficiency in external memory accesses.

INSTRUCTION SET SUMMARY

All instructions perform the same functions as their 8051 counterparts. Their effect on bits, flags, and other status functions is identical. However, the timing of each instruction is different. This applies both in absolute and relative number of clocks.

For absolute timing of real–time events, the timing of software loops can be calculated using information given in the Instruction Set table. However, counter/timers default to run at the older 12 clocks per increment. In this way, timer–based events occur at the standard intervals with software executing at higher speed. Timers optionally can run at lower numbers of clocks per increment to take advantage of faster processor operation.

The relative time of some instructions might be different in the new architecture than it was previously. For example, in the original architecture, the “MOVX A, @DPTR” instruction and the “MOV direct, direct” instruction used two machine cycles or 24 oscillator cycles. Therefore, they required the same amount of time. In the DS89C420, the MOVX instruction takes as little as two machine cycles or two oscillator cycles but the “MOV direct, direct” uses three machine cycles or 3 oscillator cycles. While both are faster than their original counterparts, they now have different execution times. This is because the DS89C420 usually uses one machine cycle for each instruction byte and requires one cycle for execution. The user concerned with precise program timing should examine the timing of each instruction to become familiar with the changes.

Many instructions require only one machine cycle, but others require more. In the original architecture, all were one or two machine cycles except for MUL and DIV.

SPECIAL FUNCTION REGISTERS (SFRs)

All peripherals and operations that are not explicit instructions in the DS89C420 are controlled via SFRs. The most common features basic to the architecture, are mapped to the SFRs. These include the CPU registers (ACC, B and PSW), data pointers (DPTRs), stack pointer, I/O ports, timer/counters and serial ports. In many cases, an SFR will control an individual function or report the function's status. The SFRs reside in register locations 80h-FFh and are only accessible by direct addressing. SFRs whose addresses end in 0h or 8h are bit-addressable.

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 2, "Special Function Registers", summarizes the SFRs and their locations. Table 3, "SFR Reset Value", 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 can select the active pointer via a dedicated SFR bit (Sel = DPS.0), or can activate 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. In general, software reads the state of external pins during a read operation.

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 "SFR Bit Description," in the DS89C420 User's Guide.

SERIAL PORTS

The DS89C420 provides two UARTs which 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 both read and write operations, and the read and write operations are distinguished by the instruction. Each UART is controlled by its own SFR control register.

SPECIAL FUNCTION REGISTERS Table 2

Register	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0	80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
SP	81h								
DPL	82h								
DPH	83h								
DPL1	84h								
DPH1	85h								
DPS	86h	ID1	ID0	TSL	AID	-	-	-	SEL
PCON	87h	SMOD_0	SMOD0	OFDF	OFDE	GF1	GF0	STOP	IDLE
TCON	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89h	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
TL0	8Ah								
TL1	8Bh								
TH0	8Ch								
TH1	8Dh								
CKCON	8Eh	WD1	WD0	T2M	T1M	T0M	MD2	MD1	MD0
P1	90h	$\overline{\text{INT5}}$	INT4	$\overline{\text{INT3}}$	INT2	TXD1	RXD1	T2EX	T2
EXIF	91h	IE5	IE4	IE3	IE2	CKRY	RGMD	RGSL	BGS
CKMOD	96h			T2MH	T1MH	T0MH	-	-	-
SCON0	98h	SM0/FE_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0
SBUF0	99h								
ACON	9Dh	PAGEE	PAGES1	PAGES0	-	-	-	-	-
P2	A0h	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
IE	A8h	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
SADDR0	A9h								
SADDR1	AAh								
P3	B0h	$\overline{\text{RD}}$	$\overline{\text{WR}}$	T1	T0	$\overline{\text{INT1}}$	$\overline{\text{INT0}}$	TXD0	RXD0
IP1	B1h	-	MPS1	MPT2	MPS0	MPT1	MPX1	MPT0	MPX0
IP0	B8h	-	LPS1	LPT2	LPS0	LPT1	LPX1	LPT0	LPX0
SADEN0	B9h								
SADEN1	BAh								
SCON1	C0h	SM0/FE_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1
SBUF1	C1h								
ROMSIZE	C2h	HBPF	BPF	TE	MOVCX	PRAME	RMS2	RMS1	RMS0
BP2	C3h	-	-	LB3	LB2	LB1	MS2	MS1	MS0

Register	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PMR	C4h	CD1	CD0	SWB	CTM	4X/$\overline{2X}$	ALEON	DME1	DME0
STATUS	C5h	PIS2	PIS1	PIS0	-	SPTA1	SPRA1	SPTA0	SPRA0
TA	C7h								
T2CON	C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
T2MOD	C9h							T2OE	DCEN
RCAP2L	CAh								
RCAP2H	CBh								
TL2	CCh								
TH2	CDh								
PSW	D0h	CY	AC	F0	RS1	RS0	OV	F1	P
FCNTL	D5h	$\overline{FBUSY}$	FERR			FC3	FC2	FC1	FC0
FDATA	D6h								
WDCON	D8h	SMOD_1	POR	EPFI	PFI	WDIF	WTRF	EWT	RWT
BPA1	DBh								
BPA2	DCh								
ACC	E0h								
EIE	E8h	-	-	-	EWDI	EX5	EX4	EX3	EX2
B	F0h								
EIP1	F1h	-	-	-	MPWDI	MPX5	MPX4	MPX3	MPX2
EIP0	F8h	-	-	-	LPWDI	LPX5	LPX4	LPX3	LPX2

SFR RESET VALUE Table 3

Register	Addr	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0	80h	1	1	1	1	1	1	1	1
SP	81h	0	0	0	0	0	1	1	1
DPL	82h	0	0	0	0	0	0	0	0
DPH	83h	0	0	0	0	0	0	0	0
DPL1	84h	0	0	0	0	0	0	0	0
DPH1	85h	0	0	0	0	0	0	0	0
DPS	86h	0	0	0	0	0	1	0	0
PCON	87h	0	0	Special	Special	0	0	0	0
TCON	88h	0	0	0	0	0	0	0	0
TMOD	89h	0	0	0	0	0	0	0	0
TL0	8Ah	0	0	0	0	0	0	0	0
TL1	8Bh	0	0	0	0	0	0	0	0
TH0	8Ch	0	0	0	0	0	0	0	0
TH1	8Dh	0	0	0	0	0	0	0	0
CKCON	8Eh	0	0	0	0	0	0	0	1
P1	90h	1	1	1	1	1	1	1	1
EXIF	91h	0	0	0	0	Special	Special	Special	0
CKMOD	96h	0	0	0	0	0	1	1	1
SCON0	98h	0	0	0	0	0	0	0	0

Register	**Addr**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
SBUF0	99h	0	0	0	0	0	0	0	0
ACON	9Dh	0	0	0	1	1	1	1	1
P2	A0h	1	1	1	1	1	1	1	1
IE	A8h	0	0	0	0	0	0	0	0
SADDR0	A9h	0	0	0	0	0	0	0	0
SADDR1	AAh	0	0	0	0	0	0	0	0
P3	B0h	1	1	1	1	1	1	1	1
IP1	B1h	1	0	0	0	0	0	0	0
IP0	B8h	1	0	0	0	0	0	0	0
SADEN0	B9h	0	0	0	0	0	0	0	0
SADEN1	BAh	0	0	0	0	0	0	0	0
SCON1	C0h	0	0	0	0	0	0	0	0
SBUF1	C1h	0	0	0	0	0	0	0	0
ROMSIZE	C2h	1	1	1	1	0	1	0	1
BP2	C3h	1	1	-	-	-	-	-	-
PMR	C4h	1	0	0	0	0	0	0	0
STATUS	C5h	0	0	0	1	0	0	0	0
TA	C7h	1	1	1	1	1	1	1	1
T2CON	C8h	0	0	0	0	0	0	0	0
T2MOD	C9h	1	1	1	1	1	1	0	0
RCAP2L	CAh	0	0	0	0	0	0	0	0
RCAP2H	CBh	0	0	0	0	0	0	0	0
TL2	CCh	0	0	0	0	0	0	0	0
TH2	CDh	0	0	0	0	0	0	0	0
PSW	D0h	0	0	0	0	0	0	0	0
FCNTL	D5h	1	0	1	1	0	0	0	0
FDATA	D6h	0	0	0	0	0	0	0	0
WDCON	D8h	0	Special	0	Special	0	Special	Special	0
BPA1	DBh	0	0	0	0	0	0	0	0
BPA2	DCh	0	0	0	0	0	0	0	0
ACC	E0h	0	0	0	0	0	0	0	0
EIE	E8h	1	1	1	0	0	0	0	0
B	F0h	0	0	0	0	0	0	0	0
EIP1	F1h	1	1	1	0	0	0	0	0
EIP0	F8h	1	1	1	0	0	0	0	0

MEMORY ORGANIZATION

There are three distinct memory areas in the DS89C420: scratch-pad registers, program memory and data memory. All registers are located on-chip but the program and data memory spaces can be either on-chip, off-chip, or both. There are 16 kbytes of on-chip program memory implemented in Flash memory, and 1 kbytes of on-chip data memory space which can be configured as program space using the ROMSIZE feature. The DS89C420 uses a memory addressing scheme that separates program memory from data memory. The program and data segments can be overlapped since they are accessed in different ways. If the maximum address of on-chip program or data memory is exceeded, the DS89C420 will perform an external memory access using the expanded memory bus. The $\overline{\text{PSEN}}$ signal will go active low to serve as a chip enable or output enable when performing a code fetch from external program memory. MOVX instructions will activate the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal for external data memory access. The lower 128 bytes of on-chip Flash memory are used to store reset and interrupt vectors. The programmable on-chip program memory feature allows software to dynamically configure the maximum address of on-chip program memory. This allows the DS89C420 to act as a boot loader for an external Flash or nonvolatile SRAM. It also enables the use of the overlapping external program spaces.

256 bytes of on-chip RAM serve as a register area and program stack, which are separated from the data memory.

REGISTER SPACE

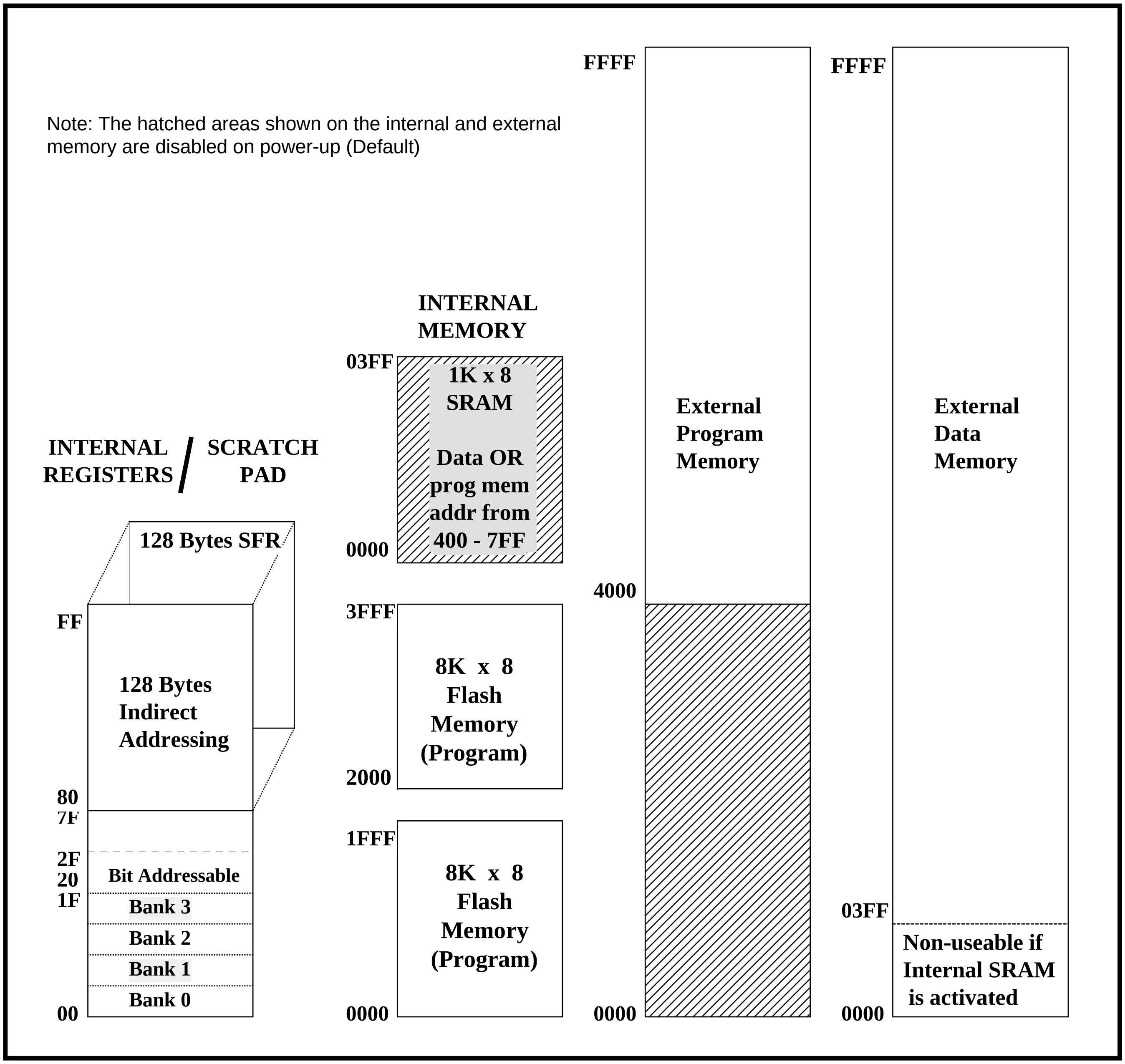
Registers are located in the 256 bytes of on-chip RAM, which can be divided into two sub-areas of 128 bytes each as illustrated in Figure 2, “Memory Map”. Separate classes of instructions are used to access the registers and the program/data memory. The upper 128 bytes are overlapped with the 128 bytes of SFRs in the memory map. The upper 128 bytes of Scratchpad RAM are accessed by indirect addressing and the SFR area is accessed by direct addressing. The lower 128 bytes can be accessed by direct or indirect addressing.

In the lower 128 bytes of Scratchpad RAM, there are four banks of eight Working Registers each. The Working Registers are general purpose RAM locations that can be addressed within the selected bank by any instructions that use R0-R7. The register bank selection is controlled via the Program Status Register in the SFR area. The contents of the Working Registers can be used for indirect addressing of the upper 128 bytes of Scratchpad RAM.

To support the Boolean operations, there are individually addressable bits in both the RAM and SFR areas. In the Scratchpad RAM area, registers 20h to 2Fh are bit addressable by software using Boolean operation instructions.

Another use of the scratchpad RAM area is for the Stack. The Stack Pointer in the SFRs is used to select storage locations for program variables and for return addresses of control operations.

MEMORY MAP Figure 2


Note: The hatched areas shown on the internal and external memory are disabled on power-up (Default)
INTERNAL REGISTERS / SCRATCH PAD
128 Bytes SFR
FF
128 Bytes Indirect Addressing
80
7F
2F
20
1F
Bit Addressable
Bank 3
Bank 2
Bank 1
Bank 0
00
INTERNAL MEMORY
03FF
1K x 8 SRAM
Data OR prog mem addr from 400 - 7FF
0000
3FFF
8K x 8 Flash Memory (Program)
2000
1FFF
8K x 8 Flash Memory (Program)
0000
FFFF
External Program Memory
4000
0000
FFFF
External Data Memory
03FF
Non-useable if Internal SRAM is activated
0000

MEMORY CONFIGURATION

As illustrated in Figure 2, “Memory Map,” the DS89C420 incorporates two 8 kbytes Flash memories for on-chip program memory and a 1 kbytes of SRAM for on-chip data memory or a particular range (400 –7FF) of “alternate” program memory space The DS89C420 uses an address scheme that separates program memory from data memory, such that the 16-bit address bus can address each memory area up to 64 kbytes.

PROGRAM MEMORY ACCESS

On-chip program memory begins at address 0000h and is contiguous through 3FFFh (16 kbits). Exceeding the maximum address of on-chip program memory causes the device to access off-chip memory. However, the maximum on-chip decoded address is selectable by software using the ROMSIZE feature. Software can cause the DS89C420 to behave like a device with less on-chip memory. This is beneficial when overlapping external memory, such as Flash, is used. The maximum memory size is dynamically variable. Thus a portion of memory can be removed from the memory map to access off-chip memory, then restored to access on-chip memory. In fact, all of the on-chip memory can be removed from the memory map allowing the full 64 kbits memory space to be addressed from off-chip memory. Program memory addresses that are larger than the selected maximum are automatically fetched from outside the part via Ports 0 and 2. A depiction of the memory map is shown in Figure 2.

The ROMSIZE register is used to select the maximum on-chip decoded address for program memory. Bits RMS2, RMS1, RMS0 have the following effect:

RMS2	**RMS1**	**RMS0**	**Maximum on-chip program memory Address**
0	0	0	0k
0	0	1	1k/03FFh
0	1	0	2k/07FFh
0	1	1	4k/0FFFh
1	0	0	8k/1FFFh
1	0	1	16k (default)/3FFFh
1	1	0	Invalid - reserved
1	1	1	Invalid - reserved

The reset default condition is a maximum on-chip program memory address of 16 kbits. When accessing external program memory, the first 16 kbits 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 12 kbits boundary (~3000h) and that the ROMSIZE register is currently configured for a 16 kbits internal program space. If software reconfigures the ROMSIZE register to 4 kbits (0000h-0FFFh) in the current state, the device will immediately jump to external program execution because program code from 4 kbits to 16 kbits (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 will be internal (or external) both before and after the operation. In the above example, the instruction which modifies the ROMSIZE register should be located below the 4 kbits (1000h) boundary, so that it will be unaffected by the memory modification. The same precaution should be applied if the internal program memory size is modified while executing from external program memory.

For non- page mode operations, off-chip memory is accessed using the multiplexed address/data bus on P0 and the MSB address on P2. While serving as a memory bus, these pins are not I/O ports. This convention follows the standard 8051 method of expanding on-chip memory. Off-chip program memory access also occurs if the $\overline{\text{EA}}$ pin is a logic 0. $\overline{\text{EA}}$ overrides all bit settings. The $\overline{\text{PSEN}}$ signal will go active (low) to serve as a chip enable or output enable when Ports 0 and 2 fetch from external program memory.

The $\overline{RD}$ and $\overline{WR}$ signals are used to control the external data memory device. Data memory is accessed by MOVX instructions. The MOVX@Ri instruction uses the value in the designated working register to provide the LSB of the address, while Port 2 supplies the address MSB. The MOVX@DPTR instruction uses one of the two data pointers to move data over the entire 64 kbits external data memory space. Software selects the data pointer to be used by writing to the SEL bit (DPS.0).

The DS89C420 also provides a user option for high-speed external memory access by reconfiguring the external memory interface into page mode operation.

NOTE:

When using the original 8051 expanded bus structure the throughput is reduced by 75% compared with that of internal operations. This is due to the CPU being stalled for three out of four clocks waitng for the data fetch which takes four clocks. Page mode 1 is the only external addressing mode where the CPU does not stall for external memory access.

ON-CHIP PROGRAM MEMORY

The full on-chip program memory range can be fetched by the processor automatically. The reset routines and all interrupt vectors are located in the lower 128 bytes of the on-chip program memory area.

On-chip program memory is logically divided into two 8 kbits Flash memory banks to support in-application programming. The on-chip program memory is designed to be programmed in-application with the standard 5 volt V_{CC} supply under the control of the user software or by using a built-in program memory Loader. It can also be programmed in standard Flash or EPROM programmers. The DS89C420 incorporates a Memory Management Unit (MMU) and other hardware to support any of the three programming methods. The MMU controls program and data memory access, and provides sequencing and timing controls for programming of the on-chip program memory. There is also a separate Security Flash block which is used to support a standard three-level lock, a 64-byte encryption array and other Flash options.

SECURITY FEATURES

The DS89C420 incorporates a 64-byte encryption array, allowing the user to verify program codes while viewing the data in encrypted form. The encryption array is implemented in a Security Flash memory Block which has the same electrical and timing characteristics as the on-chip program memory. Once the encryption array is programmed to non-FFh, the data presented in the verify mode is encrypted. Each byte of data is XNOR'ed with a byte in the encryption array during verification.

There is also a three-level lock that restricts viewing of the internal program and data memory contents. By programming the three lock bits, the user may select a level of security as specified in Table 4, "Flash Memory Lock Bits".

NOTE:

The read/write accessibility of the Flash memory during in-application programming is not affected by the state of the lock bits. However, the lock bits do affect the read/write accessibility in program memory Loader and parallel programming modes.

Once a security level is selected and programmed, the setting of the lock bits remains. Only a mass erase will erase these bits and allow reprogramming the security level to a less restricted protection.

FLASH MEMORY LOCK BITS Table 4

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 will be at logic 1 when read. The value of this register can be read at address FCh in Parallel Programming mode or executing a Verify Option Control Register instruction in ROM Loader or In-Application Programming mode.

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

Address	Value	Meaning
30h	DAh	Manufacturer ID
31h	42h	DS89C420 Device ID
60h	01h	Device extension

IN-APPLICATION PROGRAMMING BY USER SOFTWARE

The DS89C420 supports in-application programming of on-chip Flash memory by user software. In-application programming is initiated by writing a Flash command into the Flash Control (FCNTL:D5h) register to enable the Flash memory for erase/program/verify operations. Address and data are input into the MMU via the Flash Data (FDATA:D6h) register. The Flash command also enables read/write accesses to the FDATA. The MMU's sequencer provides the operation sequences and control functions to the Flash memory. The MMU is designed to operate independently from the processor except for read/write access to the SFRs.

Only the upper 8 kilobytes of the on-chip program memory can be in-application programmed by the user software. The lower 8 kilobytes of the on-chip program memory contain system hardware dependent codes that are crucial to system operation and should not be altered during in-application programming. To update the lower 8 kilobytes, the user software must first update new codes in the upper 8 kilobyte memory bank. Once the new codes are updated and verified, the user software must complement the logic state of the Memory Bank Select before forcing a reset. The Memory Bank Select is a nonvolatile memory cell which can be set or reset by the MMU, and its logic state determines which memory bank is to be used as the lower bank. A reset automatically configures the memory banks in an addressing order defined by the logic state of the Memory Bank Select. The System Reset command can be used by software to force a system reset after changing the Memory Bank Select. A reset effectively replaces the original codes with the newly updated codes by switching the memory banks. The original upper memory bank becomes the lower memory bank with starting address at 0000h. The original lower program memory bank is now the upper program memory bank (starting address at 2000h) and can be erased and reprogrammed as needed.

All Flash operations are self-timed. The progress of an erase or programming operation can be monitored by the user software via the Flash Busy (FBUSY;FCNTL.7) bit with a reset value at logic 1. A selected operation automatically starts when required data is written to the FDATA SFR. The MMU clears the FBUSY bit to indicate the start of a write/erase operation. This bit is held low until either the end of the operation, or an error indicator is returned. A Flash operating failure terminates the current operation and sets the Flash Error Flag (FERR;FCNTL.6) to a logic 1. Both the Busy and Error Flags are read-only bits.

Read/write access during in-application programming is not affected by the state of the lock bits.

The Flash Command (FC3-FC0;FCNTL.3:0) bits provide Flash commands as listed in Table 5.

IN-APPLICATION PROGRAMMING COMMANDS Table 5

FC3:FC0	Command	Operation
0000	Read Mode	Default state. All Flash blocks are in read mode.
0001	Verify Option Control Register	Read data from the Option Control Register. Data is available in the FDATA at the end of the following machine cycle. FDATA.3 is the logic value of the Watchdog POR default setting.
0010	Verify Security Block	Read a byte of data from the Security Block. After the address byte is written to the FDATA, data is available in the FDATA at the end of the following machine cycle. (Lock bits are addressed at 40h and FDATA.5:3 are the logic value of LB1, LB2 and LB3, respectively.)
0011	Verify Upper Program Memory Bank	Read a byte of data from upper Flash memory bank (address range from 2000h to 3FFFh). The first and second byte writes to the FDATA are the upper and lower byte of the address. Data is available in the FDATA at the end of the following machine cycle after the second address byte is written.
0100	Verify Bank Select	Read the logic value of the Bank Select. Data is available in the FDATA at the end of the following machine cycle and the LSB (FDATA.4) is the data output for Bank Select.
1000	Complement Memory Bank Select	Complement the logic value of the program Memory Bank Select.
1001	Write Option Control Register	Write to the Option Control Register as data is written to FDATA. Bit 3 of the data byte represents the Watchdog POR default setting.

FC3:FC0	Command	Operation
1010	Write Security Block	Write a byte of data to the Security Block at a selected location as addressed by the first byte write to the FDATA. The second write to the FDATA is the data byte. (Lock bits are addressed at 40h and the FDATA 5:3 represents lock bit LB3, LB2 and LB1, respectively.)
1011	Write Upper Program Memory Bank	Write a byte of code to the upper Flash memory bank (address range from 2000h to 3FFFh). The first and second byte writes to the FDATA are the upper byte and the lower byte of the address. The third write to the FDATA is the data byte.
1100	Erase Option Control Register	Erase the Option Control Register. The contents of this register will be returned to FFh. This operation will disable the Watchdog reset function on power-up.
1101	Erase Security Block	Erase the Security Flash block that contains the 64-byte Encryption Array and the lock bits. The content of every memory location will be turned into FFh.
1110	Erase Upper Program Memory Bank	Erase the upper 8 kilobytes of Flash memory bank. The contents of every memory location will be returned to FFh.
1111	System Reset	System reset. This command is used to cause a system reset which effectively switches the order of the internal memory banks if the Memory Bank Select has been complemented.

The Flash Command bits are cleared to 0 on all forms of reset, and it is important for the user software to clear these bits to 0 to return the flash memory to read mode from erase/program operation. This setting is a “no operation” condition for the MMU which allows the processor to return to its normal execution. Note that the Busy and Error flags have no function in normal Flash read mode.

The FCNTL SFR can only be written using timed access. This procedure provides protection against inadvertent erase/program operation on the Flash memory. Any command written to the FCNTL during a Flash operation is ignored (FBUSY=0). To ensure data integrity, an erase command sequence should be reinitiated if an erase or program operation is interrupted by a reset.

ROM LOADER

The full 16 kbits of on-chip Flash program memory space, Security Flash block and external SRAM can be programmed in-system from an external source via serial port 0 under the control of a built-in ROM Loader. The ROM Loader Loader also has an auto-baud feature which 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”, $\overline{\text{EA}}$ = “0”and $\overline{\text{PSEN}}$ to transition from a “1” to “0”. It will remain in effect until power-down or when the condition (RST=1 and $\overline{\text{PSEN}}$ = $\overline{\text{EA}}$ = “0”) is removed. Entering the ROM Loader mode will force the processor to start fetching from the 2k 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 which are derived (by the built-in ROM Loader) from data transmitted over the serial interface from a host PC. ROM Loader software for the creation of required commands (for flash or external data memory) from the host PC is available from Dallas Semiconductor, titled KIT420.

Full details of the ROM Loader software and its implementation is given in the *DS89C420 User's Guide.*

PARALLEL PROGRAMMING

The DS89C420 allows parallel programming of its internal Flash memory compatible with standard Flash or EPROM programmers. In parallel programming mode, a mass erase command is used to erase all memory locations in the 16 kbits program memory, the Security block and the Memory Bank Select. Erasing of the Memory Bank Select sets it to the default state; the Memory Bank Select cannot be altered otherwise. If lock bit LB2 has not been programmed, the program code can be read back for verification. The state of the lock bits can also be verified directly in the parallel programming mode. One instruction is used to read signature information (at addresses 30, 31 and 60h). Separate instructions are used for the Option Control Register.
The following sequence can be used to program the flash memory in the parallel programming mode:

1. The DS89C420 is powered up and running at a clock speed between 4 and 6 MHz.
2. Set RST = $\overline{\text{EA}}$ = 1 and $\overline{\text{PSEN}}$ = 0.
3. Apply the appropriate logic combination to pins P2.6, P2.7, P3.6 and P3.7 to select one of the Flash instructions shown in Table 8.
   - For program operation, apply the desired address to pins P1.7:0 and P2.5:0. Data is written to port 0.
   - For verify operation, apply the desired address to pins P1.7:0 and P2.5:0. Data is read at port 0.
4. Pulse ALE/$\overline{\text{PROG}}$ once to perform an erase/program operation.
5. Repeat steps 3 and 4 as necessary.

PARALLEL PROGRAMMING INSTRUCTION SET Table 7

Instruction	P2.5:0, P1.7:0	P0.7:0	$\overline{\text{PROG}}$	P2.6	P2.7	P3.6	P3.7	Operation
Mass Erase	Don't care	Don't care	PL[1]	H	L	L	L	Mass erase the 16k x 8 program memory, the Security block and the Bank Select. The contents of every memory location will be returned to FFh
Write Program Memory	ADDR	DIN	PL[3]	L	H	H	H	Program the 16k program memory
Read Program Memory	ADDR	DOUT	H[4]	L	L	H	H	Verify the 16k program memory
Write Encryption Array	ADDR	DIN	PL[3]	L	H	L	H	Program the 64 Byte Encryption Array
Write LB1	Don't care	Don't care	PL[3]	H	H	H	H	Program LB1 to logic 0
Write LB2	Don't care	Don't care	PL[3]	H	H	L	L	Program LB2 and LB1 to 00b
Write LB3	Don't care	Don't care	PL[3]	H	L	H	L	Program LB3, LB2 and LB1 to 000b
Read Lock Bits	Don't care	DOUT	H[4]	L	L	L	H	Verify the lock bits. The lock bits are at address 40h and the three LSBs of the DOUT are the logic value of the lock bits LB3, LB2 and LB1, respectively
Write Option Control Register	Don't care	DIN	PL[3]	L	H	L	L	Program the Option Control Register. Bit 3 of the DIN represents the Watchdog POR default setting.
Erase Option Control Register	Don't care	Don't care	PL[2]	H	L	L	H	Erase the Option Control Register. This operation will disable the Watchdog reset function on power-up.
Read Address 30, 31, 60, FC	ADDR	DOUT	H[4]	L	L	L	L	30h=Manufacturer ID 31h=Device ID 60h=Device extension FCh= Verify the Option Control Register. Bit 3 of the DOUT is the logic value of the Watchdog POR

NOTES:

1. Mass erase requires an active low $\overline{\text{PROG}}$ pulse width of 828 ms

2. Erase Option Control Register requires an active low $\overline{\text{PROG}}$ pulse width of 828 ms.

3. Byte program requires an active low $\overline{\text{PROG}}$ pulse width of 100µs max.

4. $\overline{\text{PROG}}$ is weakly pulled to a high internally.

5. P3.2 is pulled low during programming to indicate Busy. P3.2 is pulled high again when programming is completed to indicate Ready.

6. P3.0 is pulled high during programming to indicate an error.

ON-CHIP DATA MEMORY

On-chip data memory is provided by the 1 kbits SRAM and occupies addresses 0000h through 03FFh. The internal data memory is disabled after a power-on reset, and any MOVX instruction will direct the data memory access to the external data memory. To enable the internal data memory, software must configure the Data Memory Enable bits DME1 and DME0(PMR.1-0). See SFR Bit Descriptions in the *DS89C420 User's Guide* for data memory configurations. Once enabled, MOVX instructions with addresses inside the 1k range will access the on-chip data memory, and addresses exceeding the 1k range will automatically access external data memory.

An internal data memory cycle spans only one system clock period to support fast internal execution.

DATA POINTER DECREMENT AND OPTIONS

The DS89C420 incorporates a hardware feature to assist applications that require data pointer decrement. Data pointer Increment/Decrement bits ID0 and ID1 (DPS.6 and DPS.7) define how the INC DPTR instruction functions in relation to the active DPTR (selected by the SEL bit). Setting ID0=1 and SEL=0 enables the decrement operation for DPTR, and execution of the INC DPTR instruction decrements the DPTR contents by 1. Similarly, setting ID1=1 and SEL=1 enables the decrement operation for DPTR1, and execution of the INC DPTR instruction decrements the DPTR1 contents by 1. With this feature, the user can configure the data pointers to operate in four ways for the INC DPTR instruction:

ID1	ID0	SEL = 0	SEL = 1
0	0	Inc. DPTR	Inc. DPTR1
0	1	Dec DPTR	Inc. DPTR1
1	0	Inc. DPTR	Dec DPTR1
1	1	Dec DPTR	Dec DPTR1

The active data pointer is always selected by the SEL (DPS.0) bit. The DS89C420 offers a programmable option that allows any instructions related to data pointer to toggle the SEL bit automatically. This option is enabled by setting the Toggle Select Enable Bit (TSL-DPS.5) to a logic 1. Once enabled, the SEL bit is automatically toggled AFTER the execution of one of the following five DPTR related instructions:

```
INC DPTR
MOV DPTR #data16
MOVC A, @A+DPTR
MOVX A, @DPTR
MOVX @DPTR, A
```

The DS89C420 also offers a programmable option that automatically increases (or decreases) the contents of the selected data pointer by 1 AFTER the execution of a DPTR related instruction. The actual function (increment or decrement) is dependent on the setting of the ID1 and ID0 bits. This option is enabled by setting the Automatic Increment/Decrement Enable (AID-DPS.4) to a logic 1 and will be affected by one of the following three instructions:

```
MOVC A, @A+DPTR
MOVX A, @DPTR
MOVX @DPTR, A
```

EXTERNAL MEMORY

The DS89C420 executes external memory cycles for code fetches and read/writes of external program and data memory. A non-page external memory cycle is four times slower than the internal memory cycles (i.e., an external memory cycle contains four system clocks). However, a page mode external memory cycle can be completed in one, two or four system clocks for a page hit and two, four or eight system clocks for a page miss, depending on user selection. The DS89C420 also supports a second page mode operation with a different external bus structure that provides for fast external code fetches but uses four system clock cycles for data memory access.

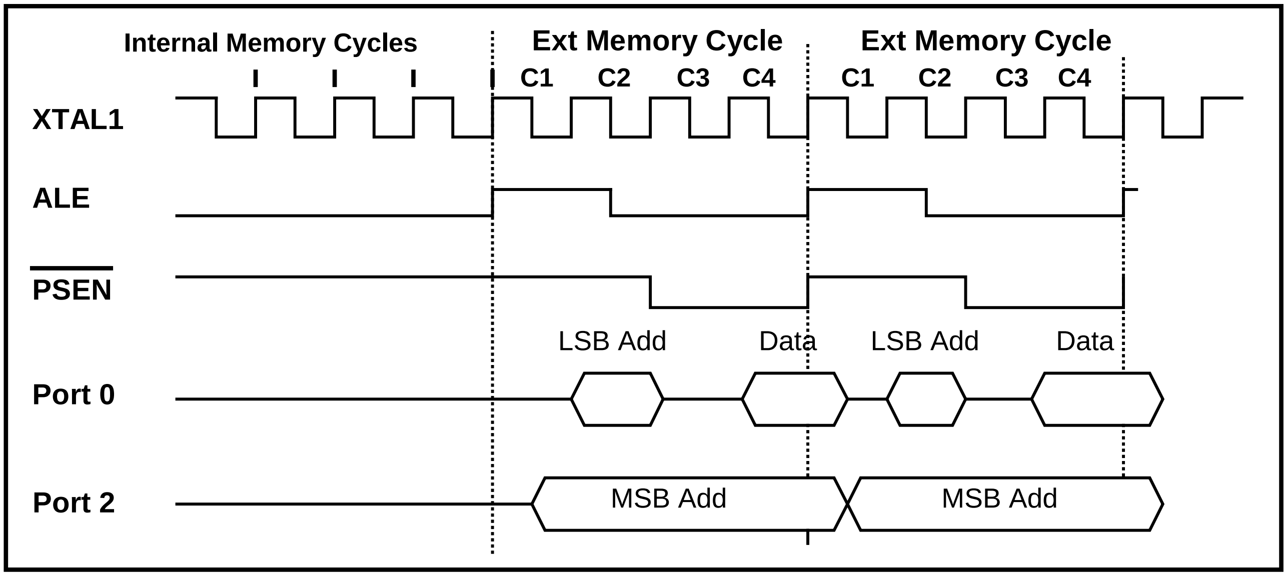
EXTERNAL PROGRAM MEMORY INTERFACE (NON-PAGE MODE)

Figure 3, “External Program Memory Access (non-page mode and CD1:CD0=10),” shows the timing relationship for internal and external code fetches when CD1 and CD0 are set to 10b, assuming the Microcontroller is in non-page mode for external fetches. Note that an external program fetch takes four system clocks, and an internal program fetch requires only one system clock.

As illustrated in Figure 3, ALE is de-asserted when executing an internal memory fetch. The DS89C420 provides a programmable user option to turn on the ALE during internal program memory operation. The ALE is automatically enabled for code fetch externally, independent of the setting of this option.

$\overline{\text{PSEN}}$ is only asserted for external code fetches, and is inactive during internal execution.

EXTERNAL PROGRAM MEMORY ACCESS (NON-PAGE MODE AND CD1:CD0=10) Figure 3



EXTERNAL DATA MEMORY INTERFACE IN NON-PAGE MODE OPERATION

Just like the program memory cycle, the external data memory cycle is four times slower than the internal data memory cycle in non-page mode. A basic internal memory cycle contains one system clock and a basic external memory cycle contains four system clocks for non-page mode operation.

The DS89C420 allows software to adjust the speed of external data memory access by stretching the memory bus cycle. CKCON (8Eh) provides an application selectable stretch value for this purpose. Software can change the stretch value dynamically by changing the setting of CKCON.2-CKCON.0. Table 8, "Data Memory Cycle Stretch Values," shows the data memory cycle stretch values and their effect on the external MOVX memory bus cycle and the control signal pulse width in terms of the number of oscillator clocks. A stretch machine cycle always contains four system clocks.

DATA MEMORY CYCLE STRETCH VALUES Table 8

MD2:MD0	Stretch Cycles	RD/WR Pulse Width (in number of oscillator clocks)			
		4X/2X,CD1,CD0 =100	4X/2X,CD1,CD0= 000	4X/2X,CD1,CD0= X10	4X/2X,CD1,CD0= X11
000	0	0.5	1	2	512
001	1	1	2	4	1024
010	2	2	4	8	2048
011	3	3	6	12	3072
100	7	4	8	16	4096
101	8	5	10	20	5120
110	9	6	12	24	6144
111	10	7	14	28	7168

As shown in Table 8, the stretch feature supports eight stretched external data memory access cycles which 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 machine 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, one stretch machine cycle (4 system clocks) is used to stretch the ALE pulse width, one stretch machine cycle is used to create additional setup and one stretch machine cycle is used to create additional hold time.

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 $\overline{\text{PSEN}}$ are altered to support this mode of operation.

Page mode is enabled by setting the PAGEE (ACON.7) bit to a logic 1. 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 number of system clocks. Table 9, “Page Mode Select,” summarizes this option. The first three selections use the same bus structure but with different memory cycle time. Setting the select bits to 11b selects another bus structure. Write access to the ACON register requires a timed access.

PAGE MODE SELECT Table 9

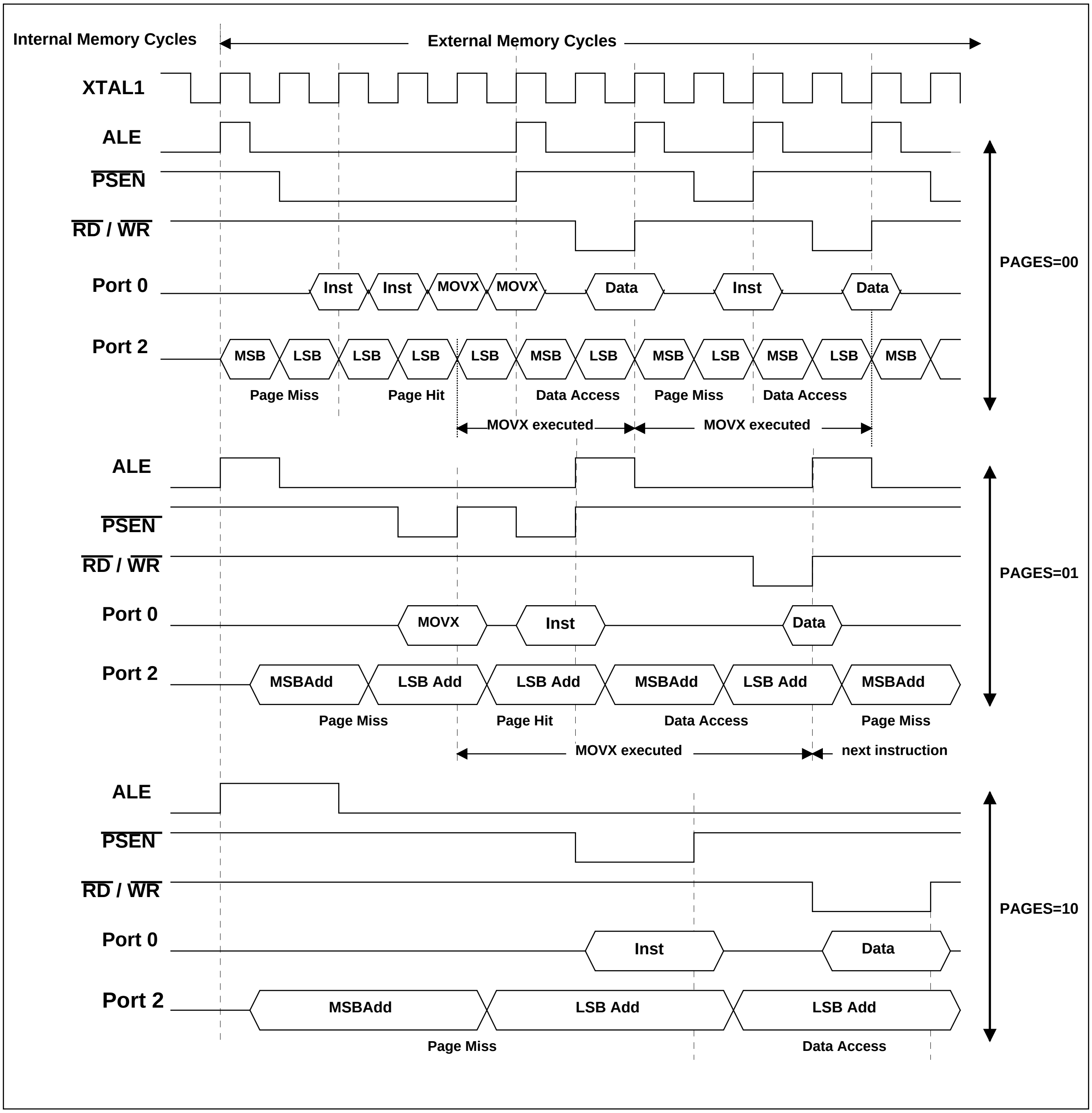
PAGES1:PAGES0	Clocks per Memory Cycle		External Bus Structure
	Page-Hit	Page-Miss	
00	1	2	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of address.
01	2	4	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of address.
10	4	8	P0: Primary data bus. P2: Primary address bus, multiplexing both the upper byte and lower byte of 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 four 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 and least significant byte of the address for each external memory cycle) and P0 is used as the primary data bus. During external code fetches, P0 will be held in a high-impedance state by the processor. Opcodes are driven by the external memory onto P0 and latched at the end of the external fetch cycle at the rising edge of $\overline{\text{PSEN}}$. 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 most significant byte 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 most significant byte 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 Addr 0-7 of the 16-bit address while the most significant address byte is held in the external address latches. $\overline{\text{PSEN}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ will strobe accordingly for the appropriate operation on the P0 data bus. There is no ALE assertion for page hits.

PAGE MODE 1 EXTERNAL MEMORY CYCLE (CD1:CD0=10) Figure 4



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, $\overline{\text{PSEN}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ are all held in inactive states and P0 is in a high-impedance state. The following half memory cycle is executed as a page hit cycle and the appropriate operation takes place.

A page miss may 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.

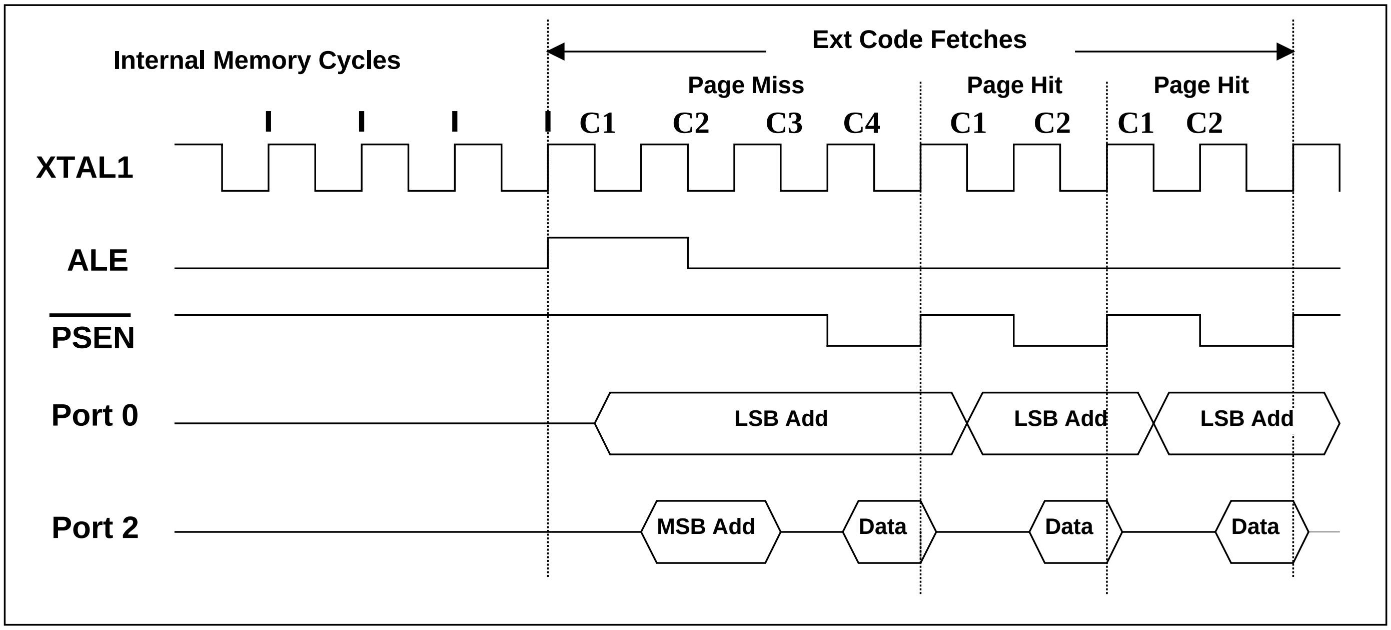
Note that there are a few exceptions for this mode of operation when PAGES1 and PAGES2 are set to 00b:

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

Figure 4, "Page Mode 1 External Memory Cycle (CD1:CD0=10)," shows the external memory cycle for this bus structure. The first case illustrates a back to back execution sequence for one 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 two cycle page mode (PAGES1=0 and PAGES0=1). $\overline{\text{PSEN}}$ is active for a full clock cycle in code fetches. Note that the page misses in this sequence are caused by changing of the most significant byte of the data address. The third case illustrates a MOVX execution sequence for four cycle page mode (PAGES1=1 and PAGES0=0). There is no page miss in this execution cycle as 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 5, "Page Mode 2 External Code Fetch Cycle (CD1:CD0=10)," illustrates the memory cycle for external code fetches.

PAGE MODE 2 EXTERNAL CODE FETCH CYCLE (CD1:CD0=10) Figure 5



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 effect 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.

PAGE MODE 1 DATA MEMORY CYCLE STRETCH VALUES (PAGES1:PAGES0=00) Table 10

		RD/WR Pulse Width (in number of oscillator clocks)			
MD2:MD0	**Stretch Cycles**	**4X/2X,CD1,CD0 =100**	**4X/2X,CD1,CD0 =000**	**4X/2X,CD1,CD0 =X10**	**4X/2X,CD1,CD0 =X11**
000	0	0.125	0.25	0.5	512
001	1	0.625	1.25	2.5	2560
010	2	1.625	3.25	6.5	6656
011	3	2.625	5.25	10.5	10752
100	7	3.625	7.25	14.5	14848
101	8	4.625	9.25	18.5	18944
110	9	5.625	11.25	22.5	23040
111	10	6.625	13.25	26.5	27136

PAGE MODE 1 DATA MEMORY CYCLE STRETCH VALUES (PAGES1:PAGES0=01) Table 11

		RD/WR Pulse Width (in number of oscillator clocks)			
MD2:MD0	**Stretch Cycles**	**4X/2X,CD1,CD0 =100**	**4X/2X,CD1,CD0 =000**	**4X/2X,CD1,CD0 =X10**	**4X/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	11264
100	7	3.75	7.5	15	15360
101	8	4.75	9.5	19	19456
110	9	5.75	11.5	23	23552
111	10	6.75	13.5	27	27648

PAGE MODE 1 DATA MEMORY CYCLE STRETCH VALUES (PAGES1:PAGES0=10) Table 12

		RD/WR Pulse Width (in number of oscillator clocks)			
MD2:MD0	**Stretch Cycles**	**4X/2X,CD1,CD0 =100**	**4X/2X,CD1,CD0 =000**	**4X/2X,CD1,CD0 =X10**	**4X/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	12288
100	7	4	8	16	16384
101	8	5	10	20	20480
110	9	6	12	24	24576
111	10	7	14	28	28672

PAGE MODE 2 DATA MEMORY CYCLE STRETCH VALUES (PAGES1:PAGES0=11) Table13

		RD/WR Pulse Width (in number of oscillator clocks)			
MD2:MD0	**Stretch Cycles**	**4X/2X,CD1,CD0 =100**	**4X/2X,CD1,CD0 =000**	**4X/2X,CD1,CD0 =X10**	**4X/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	12288
100	7	4	8	16	16384
101	8	5	10	20	20480
110	9	6	12	24	24576
111	10	7	14	28	28672

As shown in the above tables, the stretch feature supports eight stretched external data memory access cycles which 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).

PAGE MODE 1 EXTERNAL DATA MEMORY ACCESS (PAGES=01, STRETCH=1,CD=10) Figure 6

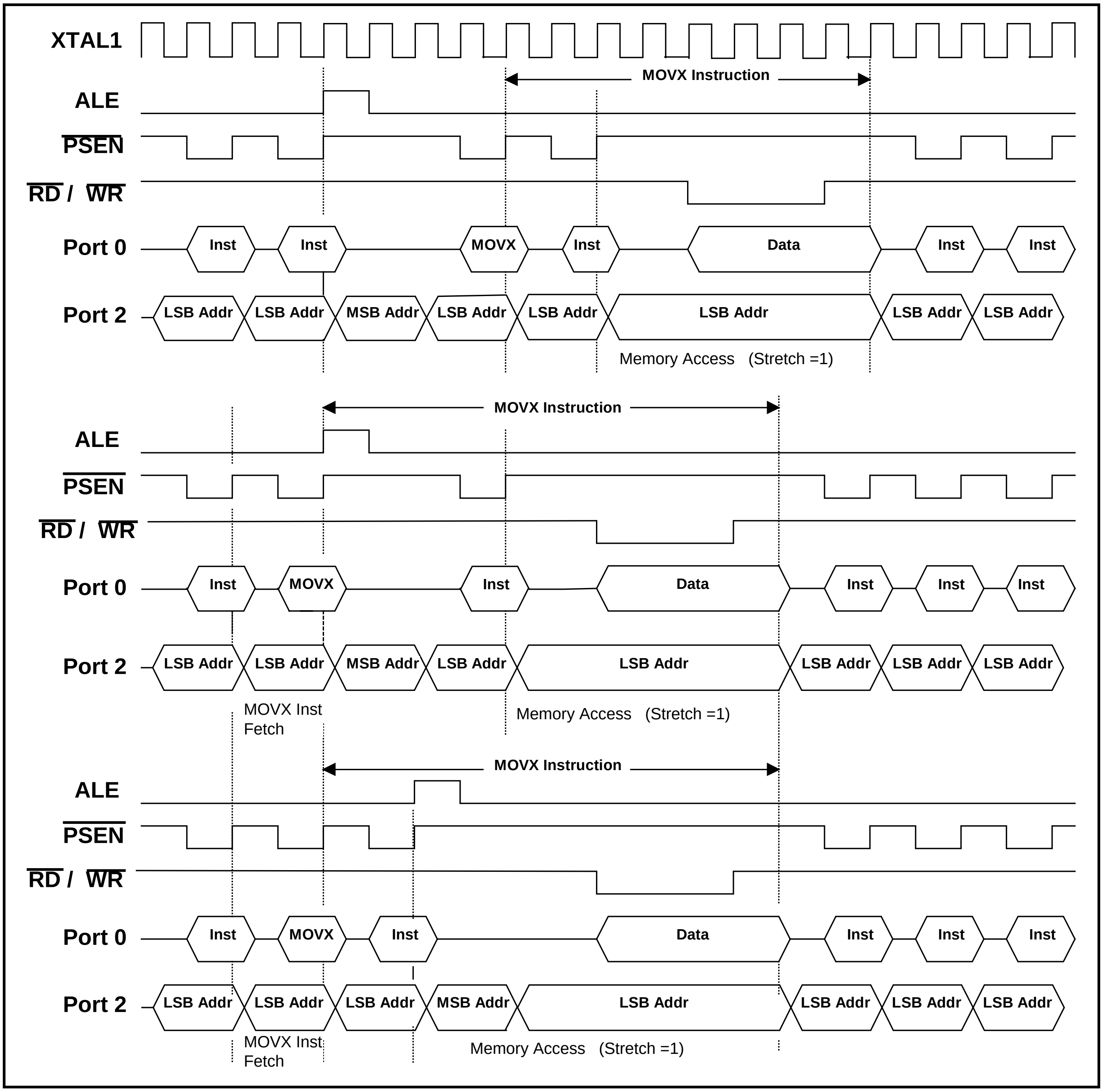


Figure 6, “Page Mode 1 External Data Memory Access (PAGES=01, Stretch=1,CD=10),” illustrates the external data memory stretch cycle timing relationship when PAGEE=1 and PAGES1:PAGES0=01. The stretched cycle shown is for a stretch value of 1 and is coincident with a page miss. Note that the first stretch 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 set-up and one system clock to create additional hold time.

PAGE MODE 1 EXTERNAL DATA MEMORY ACCESS (PAGES=01,STRETCH=4,CD=10) Figure 7

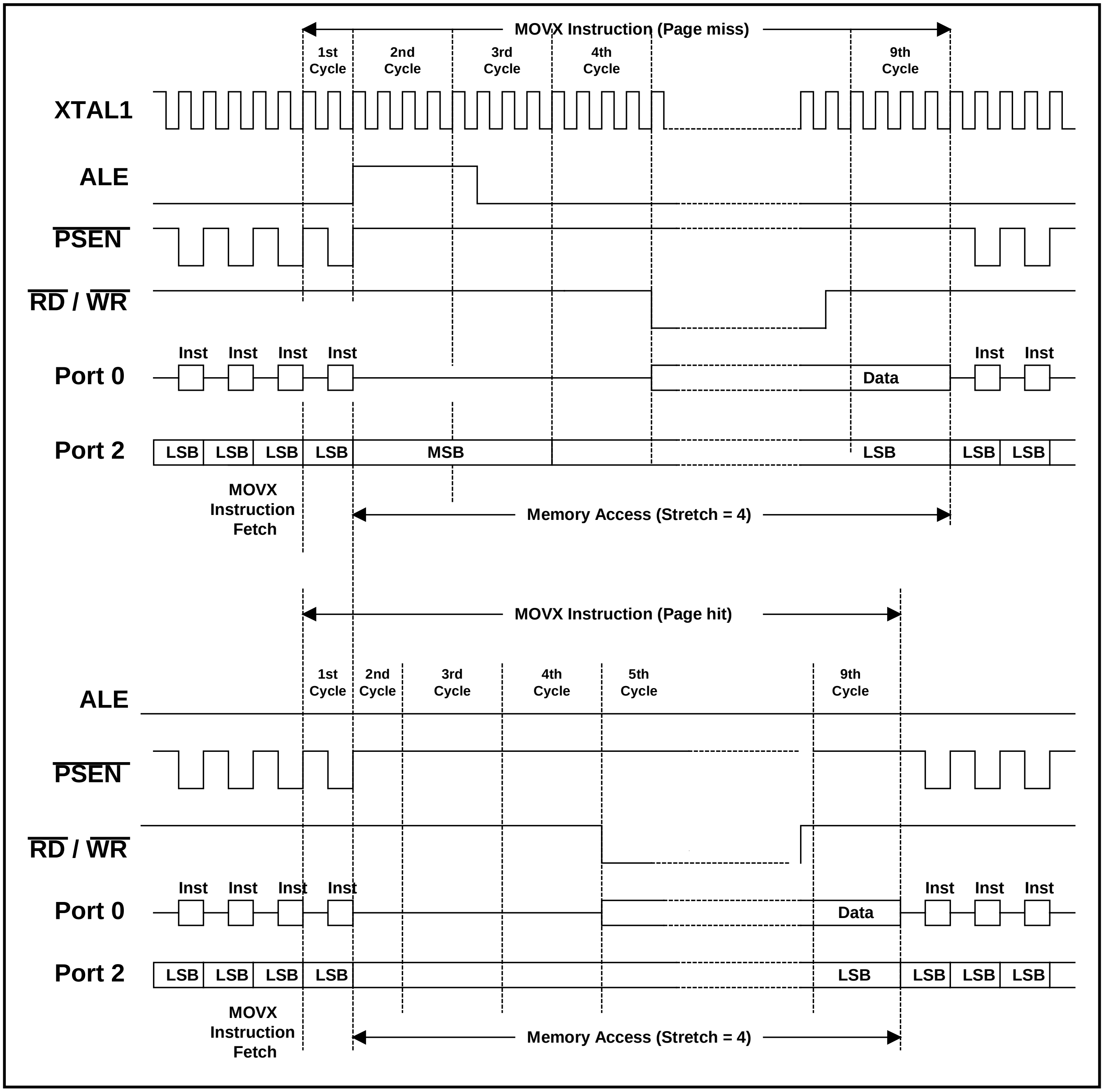


Figure 7, “Page Mode 1 External Data Memory Access (PAGES=01,Stretch=4,CD=10),” shows the timing relationship for a slow peripheral interface (Stretch value = 4). Note that a page hit data memory cycle is shorter than a page miss data memory cycle. The ALE pulse width is also stretched by a stretch cycle in the case of page miss.

The stretched data memory bus cycle timing relationship for PAGES=11 is identical to non-page mode operation since the basic data memory cycle always contains four system clocks in this page mode operation.

INTERRUPTS

The DS89C420 provides 13 interrupt vector sources. All interrupts, with the exception of the Power Fail, are controlled by a series combination of individual enable bits and a global enable (EA) in the Interrupt Enable register (IE.7). Setting EA to a logic 1 allows individual interrupts to be enabled. Setting EA to a logic 0 disables all interrupts regardless of the individual interrupt enable settings. The Power Fail interrupt is controlled by its individual enable only.

The interrupt enables and priorities are functionally identical to those of the 80C52 except that the DS89C420 supports five levels of interrupt priorities instead of the original two.

INTERRUPT PRIORITY

There are five levels of interrupt priority: Level 4 to 0. The highest interrupt priority is level 4, which is reserved for the Power Fail interrupt. All other interrupts have individual priority bits in the interrupt priority registers to allow each interrupt to be assigned a priority level from 3 to 0. The Power Fail interrupt always has the highest priority if it is enabled. All interrupts also have a natural hierarchy. In this manner, when a set of interrupts has been assigned the same priority, a second hierarchy determines which interrupt is allowed to take precedence. The natural hierarchy is determined by analyzing potential interrupts in a sequential manner with the order listed in Table 14.

INTERRUPT SUMMARY Table 14

Interrupt	Vector	Natural Order	Flag	Enable	Priority Control
Power Fail	33h	0 (Highest)	PFI (WDCON.4)	EPFI(WDCON.5)	N/A
Ext. Interrupt 0	03h	1	IE0 (TCON.1)**	EX0 (IE.0)	LPX0 (IP0.0) MPX0 (IP1.0)
Timer 0 overflow	0Bh	2	TF0 (TCON.5)*	ET0 (IE.1)	LPT0 (IP0.1) MPT0 (IP1.1)
Ext. Interrupt 1	13h	3	IE1 (TCON.3)**	EX1 (IE.2)	LPX1 (IP0.2) MPX1 (IP1.2)
Timer 1 overflow	1Bh	4	TF1 (TCON.7)*	ET1 (IE.3)	LPT1 (IP0.3) MPT1 (IP1.3)
Serial Port 0	23h	5	RI_0 (SCON0.0) TI_0 (SCON0.1)	ES0 (IE.4)	LPS0 (IP0.4) MPS0 (IP1.4)
Timer 2 overflow	2Bh	6	TF2 (T2CON.7) EXF2 (T2CON.6)	ET2 (IE.5)	LPT2 (IP0.5) MPT2 (IP1.5)
Serial Port 1	3Bh	7	RI_1 (SCON1.0) TI_1 (SCON1.1)	ES1 (IE.6)	LPS1 (IP0.6) MPS1 (IP1.6)
Ext. Interrupt 2	43h	8	IE2 (EXIF.4)	EX2 (EIE.0)	LPX2 (EIP0.0) MPX2 (EIP1.0)
Ext. Interrupt 3	4Bh	9	IE3 (EXIF.5)	EX3 (EIE.1)	LPX3 (EIP0.1) MPX3 (EIP1.1)
Ext. Interrupt 4	53h	10	IE4 (EXIF.6)	EX4 (EIE.2)	LPX4 (EIP0.2) MPX4 (EIP1.2)
Ext. Interrupt 5	5Bh	11	IE5 (EXIF.7)	EX5 (EIE.3)	LPX5 (EIP0.3) MPX5 (EIP1.3)
Watchdog	63h	12 (Lowest)	WDIF (WDCON.3)	EWDI (EIE.4)	LPWDI (EIP0.4) MPWDI (EIP1.4)

* Cleared automatically by hardware when the service routine is vectored to.

** If the interrupt is edge triggered, cleared automatically by hardware when the service routine is vectored to. If the interrupt is level triggered, the flag follows the state of the pin.

The processor indicates that an interrupt condition occurred by setting the respective flag bit. This bit is set regardless of whether the interrupt is enabled or not. Unless marked in Table 15, all of these flags must be cleared by software.

TIMER/COUNTERS

Three 16-bit timers are incorporated in the DS89C420. All three timers can be used as either counters of external events, where 1 to 0 transitions on a port pin are monitored and counted, or timers that count oscillator cycles. Table 16, “Timer Functions,” summarizes the timer functions.

Timers 0 and 1 both have three modes of operations. They can each be used as a 13-bit timer/counter, a 16-bit timer/counter, or an 8-bit timer/counter with auto-reload. Timer 0 has a fourth operating mode as two 8-bit 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 mode of operation is controlled by the timer Mode (TMOD) register. Each timer consists of a 16-bit register in two bytes, which can be found in the SFR map as TL0, TH0, TL1 and TH1. Timers 0 and 1 are enabled by the timer Control (TCON) register.

TIMER FUNCTIONS Table 15

Functions	Timer 0	Timer 1	Timer 2
Timer/Counter	13/16/8*/2x8 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. 2x8 bit mode does not.

Timer 2 is a true 16-bit timer/counter which, 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 its selectable timebase, (refer to Table 18, “Effect of Clock Mode on timer Operation in number of oscillator clocks”, for details). 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 timebase is fixed at divide by 2, regardless of the setting of its timer mode bits.

For details of operation, refer to the High-Speed Microcontroller User’s Guide, Section 11: Programmable timers.

TIMED ACCESS

The timed access function provides system control verification to system functions. The timed access function prevents an errant CPU from making accidental changes to certain SFR bits which are considered vital to proper system operation. This is achieved by the use of 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	RMS3	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

Before these bits can be altered, the processor must execute the timed access sequence. This sequence consists of writing an AAh to the Timed Access (TA, C7h) register followed by writing a 55h to the same register within three machine cycles. This timed sequence of steps then allows any of the timed access-protected SFR bits to be altered during the three machine cycles following the writing of the 55h. Writing to a timed access-protected bit outside of these three machine cycles has no effect on the bit.

The timed access process is address, data, and time dependent. A processor running out of control and not executing system software will statistically not be able to perform this timed sequence of steps, and as such will not accidentally alter the protected bits. It should be noted that this method should be used in the main body of the system software and never used in an interrupt routine in conjunction with the Watchdog reset. Interrupt routines using the timed access Watchdog reset bit (RWT) can recover a lost system and allow the resetting of the Watchdog, but the system will return to a lost condition once the RETI is executed. Also it is advisable that interrupts be disabled (EA=0) when executing the timed access sequence, since an interrupt during the sequence will add time, making the timed access attempt fail.

POWER MANAGEMENT AND CLOCK DIVIDE CONTROL

The DS89C420 incorporates power management features that monitor the power supply voltage levels and support low power operation with three power saving modes. Such features include a bandgap voltage monitor, Watchdog timer, selectable internal ring oscillator, and programmable system clock speed. The SFRs that provide control and application software access are the Watchdog Control (WDCON,D8h), Extended Interrupt Enable (EIE, E8h), Extended Interrupt Flag (EXIF, 91h) and Power Control (PCON, 87h) registers.

SYSTEM CLOCK DIVIDE CONTROL

The programmable Clock Divide Control bits (CD1 and CD0) provide the processor with the ability to adapt to different crystals and also to slow the system clocks providing lower power operation when required. An on-chip crystal multiplier allows the DS89C420 to operate at two or four times the crystal frequency by setting the $4X/\overline{2X}$ bit and is enabled by setting the CTM bit to a logic 1. An additional circuit provides a clock source at divide by 1024. When used with a 10 MHz crystal, for example, the processor executes machine cycle in times ranging from 25 ns (divide by 0.25) to 102.4 µs (divide by 1024) and maintains a highly accurate serial port baud rate while allowing the use of more cost effective lower frequency crystals. Although the Clock Divide Control bits may be written at any time, certain hardware features have been provided to enhance the use of these clock controls to guarantee proper serial port operation, and also to allow for a high speed response to an external interrupt. The 01b setting of CD1 and CD0 is reserved, and has the same effect as the setting of 10b which forces the system clock into a divide by 1 mode. The DS89C420 defaults to divide by 1 clock mode on all forms of reset.

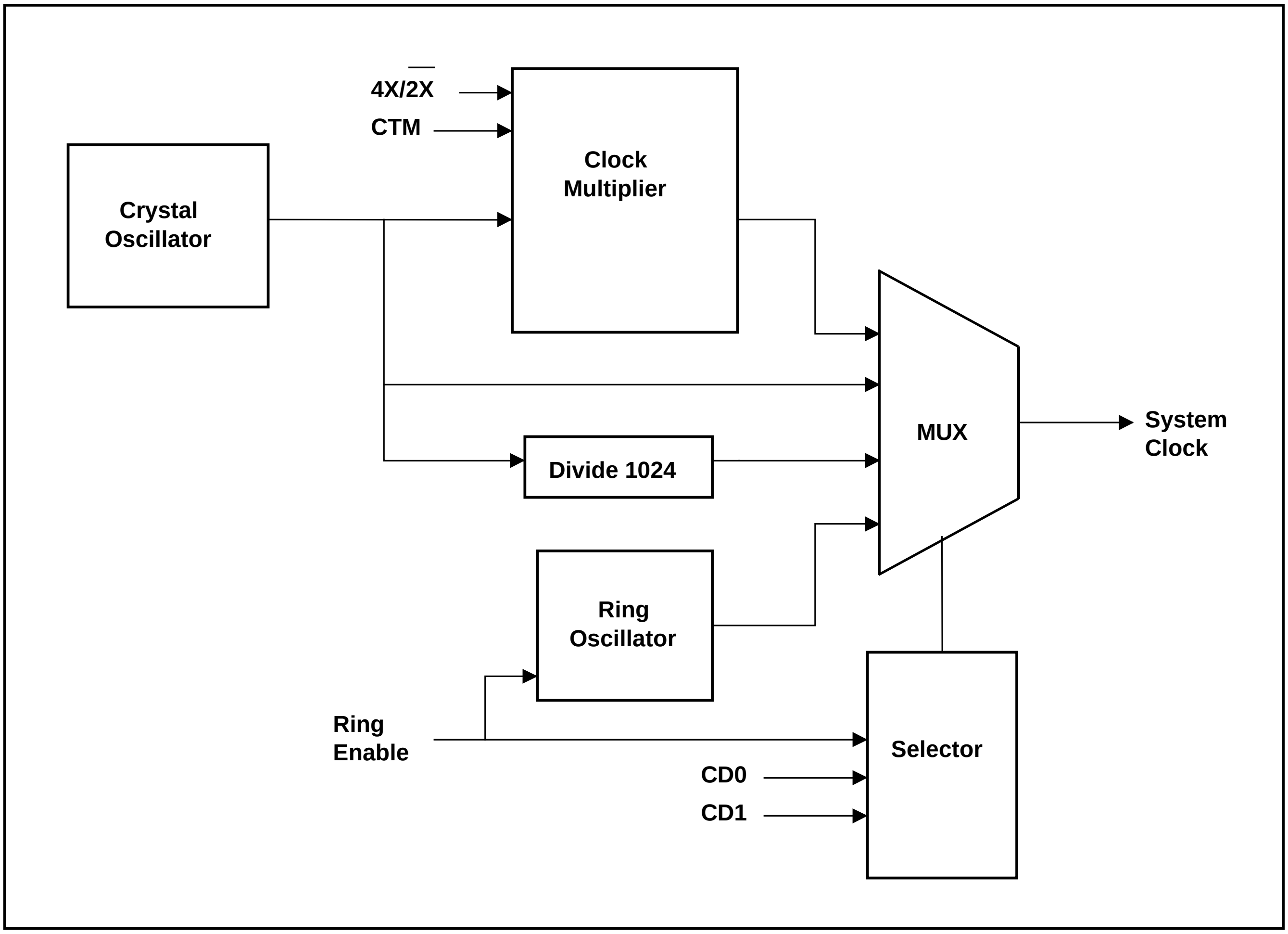
When programmed to the divide by 1024 mode, and the switchback bit (PMR.5: SWB) is also set, the system forces the Clock Divide Control bits to reset automatically to the divide by 1 mode whenever the system has detected externally enabled interrupts.

The oscillator divide ratios of 0.25, 0.5 and 1 are also used to provide standard baud rate generation for the serial ports through a forced divide by 12 input clock (TxMH,TxM = 00b, x = 1, 2, or 3) to the timers.

When in divide by 1024 mode, in order to allow a quick response to incoming data on a serial port, the system utilizes the switchback mode to automatically revert to divide by 1 mode whenever a start bit is detected. This automatic switchback is only enabled during divide by 1024 mode and all other clock modes are unaffected by interrupts and serial port activity. See Power Management Mode for more details.

Use of the divide by 0.25 or 0.5 option via the Clock Divide Control bits requires that the Crystal Multiplier be enabled and the specific system clock multiply value be established by the $4X/\overline{2X}$ bit in the PMR register. The Multiplier is enabled via the CTM (PMR.4) bit but cannot be automatically selected until a start-up delay has been established via the CKRY bit in the status register. The $4X/\overline{2X}$ bit can only be altered when the CTM bit is cleared to a logic 0. This prevents the system from changing the multiplier until the system has moved back to the divide by 1 mode and the multiplier has been disabled via the CTM bit. The CTM bit can only be altered when the CD1 and CD0 bits are set to divide by 1 mode and the RGMD bit is cleared to 0. Setting the CTM to a logic 1 from a previous logic 0 automatically clears the CKRY bit in the status register and starts the multiplier start-up time-out in the multiplier start-up counter. During the multiplier start-up period the CKRY bit will remain cleared and the CD1 and CD0 clock controls cannot be set to 00b. The CTM bit is cleared to a logic 0 on all resets. Figure 8 "System Clock sources" gives a simplified description of the generation of the system clocks. Specifics of hardware restrictions associated with the use of the $4X/\overline{2X}$ CTM, CKRY, CD1 and CD0 bits are outlined in the SFR description.

SYSTEM CLOCK SOURCES Figure 8



BANDGAP- MONITORED INTERRUPT AND RESET GENERATION

The power monitor in the DS89C420 monitors the Vcc pin in relation to the on-chip bandgap voltage reference. Whenever V_{CC} falls below V_{PFW}, an interrupt is generated if the corresponding Power Fail interrupt enable bit EPFI (WDCON.5) is set, causing the device to vector to address 33h. The Power-Fail interrupt status bit PFI (WDCON.4) will be set anytime V_{CC} transitions below V_{PFW}, and can only be cleared by software once set. Similarly, as V_{CC} falls below V_{RST}, a reset is issued internally to halt program execution. Following power-up, a power-on reset will initiate a power-on reset time-out before starting program execution. When Vcc is first applied to the DS89C420, the processor is held in reset until $V_{CC} > V_{RST}$ and a delay of 65,536 oscillator cycles has elapsed, to ensure that power is within tolerance and the clock source has had time to stabilize. Once the reset time-out period has elapsed, the reset condition is removed automatically and software execution begins at the reset vector location of 0000h. The Power-On Reset flag POR (WDCON.6) is set to logic 1 to indicate a power-on reset has occurred, and can only be cleared by software.

When the DS89C420 enters Stop mode, the bandgap, reset comparator and power fail interrupt comparator are automatically disabled to conserve power if the BGS (EXIF.0) bit is set to a logic 0. This is the lowest power mode resulting in a current of approximately 1 µA. If BGS is set to a logic 1, the Stop mode current will be approximately 50 uA. In this mode the Bandgap reference, reset comparator and the power fail comparator are powered up, although in a reduced fashion while in Stop mode.

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 time-out has a default divide ratio of 2^{17} of the crystal oscillator clock, with the Watchdog reset set to time-out 512 clock cycles later. This results in a 33 MHz crystal oscillator producing an interrupt time-out every 3.9718 ms, followed 15.5 µs 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. All Watchdog timer reset time-outs follow the programmed interrupt time-outs by 512 oscillator cycles. Table 17, “Watchdog Time-Out Value (in number of oscillator clocks),” summarizes the Watchdog bits settings and the time-out values.

WATCHDOG TIME-OUT VALUE (IN NUMBER OF OSCILLATOR CLOCKS) Table 16

		Watchdog Interrupt Time-out				**Watchdog Reset Time-out**			
4X/2X	**CD1:0**	**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}+512	2^{18}+512	2^{21}+512	2^{24}+512
0	00	2^{16}	2^{19}	2^{22}	2^{25}	2^{16}+512	2^{19}+512	2^{22}+512	2^{25}+512
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}+512	2^{30}+512	2^{33}+512	2^{36}+512

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 via the RWT bit, the Watchdog timer will generate interrupt time-outs 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 time-out. The Watchdog interrupt is enabled by the EWDI bit (EIE.4) when it is set to 1. The Watchdog timer reset and interrupt time-outs are measured by counting system clock cycles.

An independent Watchdog timer functions as the crystal start-up counter to count 65,536 crystal clock cycles before allowing the crystal oscillator to function as the system clock. This warm-up 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 start-up time whenever the CTM in the PMR register is set to enable the Crystal $4X/\overline{2X}$ Multiplier.

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.

INTERNAL SYSTEM RESET

An internal system reset can be initiated during in-application programming by writing a system reset command to the FCNTL register. The reset state will be maintained for TBD clock cycles. Once the reset is removed, the on-chip memory banks are reconfigured in an addressing order determined by the logic value of the bank select bit, and the processor resumes execution at 0000h.

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 will be maintained as long as RST is pulled to logic 1. When the RST is removed, the processor exits the reset state within TBD 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 will not affect the internal reset condition.

OSCILLATOR FAIL DETECT

The DS89C420 incorporates an oscillator fail detect circuit which, when enabled, causes a reset if the crystal oscillator frequency falls below 30 kHz and holds the chip in reset with the ring oscillator operating. The circuit is enabled by setting the OFDE (PCON.4) bit to a logic 1. The OFDE bit is only cleared from a logic 1 to a logic 0 by a power fail reset or by software. A reset caused by an oscillator failure also sets the OFDF (PCON.5) to a logic 1. This flag is cleared by software or power-on reset.

Note that this circuit will not force a reset when the oscillator is stopped via the software enabled Stop mode.

POWER MANAGEMENT MODE

The power management mode offers a software controllable power saving scheme by providing a reduced instruction cycle speed, which allows the DS89C420 to continue to operate while using an internally divided version of the clock source to save power. Power management mode is invoked by software setting the Clock Divide Control bits CD1 and CD0 (PMR.7-6) bits to 11b, which sets an operating rate of 1024 oscillator cycles for one machine cycle. On all forms of reset, the Clock Divide Control bits default to 10b which selects one oscillator cycle per machine cycle.

Since the clock speed choice affects all functional logic including timers, the DS89C420 implements several hardware switchback features that allow the clock speed to automatically return to the divide by 1 mode from a reduced cycle rate. This switchback function is enabled by setting the SWB (PMR.5) bit to a 1 in software.

When CD1 and CD0 are programmed to the divide by 1024 mode and the SWB bit is also enabled, the system will force the Clock Divide Control bits to automatically reset to the divide by 1 mode whenever the system detects an externally enabled (and allowed via nesting priorities) interrupt. The switchback will occur whenever one of the two following conditions occur. The first switchback condition is initiated by the detection of a low on either $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT3}}$, $\overline{\text{INT5}}$, or a high on INT2 or INT4 when the respective pin has been programmed and allowed (via nesting priorities) to issue an interrupt. The second switchback condition occurs when either serial port is enabled to receive data and is found to have an active low transition on the respective receive input pin. Serial port transmit activity also forces a switchback if the SWB is set. Note that the serial port activity, as related to the switchback is independent of the serial port interrupt relationship. Any attempt to change the clock divider to the divide by 1024 mode while the serial port is either transmitting or receiving will have no effect, leaving the clock control in the divide by 1 mode. Note also that the switchback interrupt relationship requires that the respective external interrupt source is allowed to actually generate an interrupt as defined by the priority of the interrupt and the state of the nested interrupts, before the switchback can actually occur. An interrupt by the serial port is not required, nor is the setting of Serial Port Enable. Disabling external interrupts and serial port receive/transmission mode will disable the automatic switchback mode. Clearing the SWB bit also disables the switchback, and all interrupt and serial port controls of the clock divider are disabled. All other clock modes will ignore the switchback relationship and are unaffected by interrupts and serial port activity.

The basic divide by 12 mode for the timers (TxMH,TxM = 00b), as well as the divide by 32 and 64 for mode 2 on the Serial Ports, has been maintained when running the processor with the oscillator divide ratio of 0.25, 0.5 and 1. Serial ports and timers track the oscillator cycles per machine cycle when the higher divide ratio of 1024 is selected, and require the switchback function to automatically return to the divide by 1 mode for proper operation when a qualified event occurs. Table 18, "Effect of Clock Mode on Timer Operation (in number of oscillator clocks)," summaries the effect of clock mode on timer operation.

It is possible to enable a receive function on a serial port when incoming data is not present and then change to the higher divide ratio. An inactive serial port receive/transmition mode requires the receive input pin to remain high and all outgoing transmissions to be completed. During this inactive receive mode it is possible to change the Clock Divide Control bits from a divide by 1 to a 1024 divide ratio. In the case when the serial port is being used to receive or transmit data it is very important to validate an attempted change in the Clock Divide Control bits (read CD1 and CD0 to verify write was allowed) before proceeding with low power program functions.

EFFECT OF CLOCK MODE ON TIMER OPERATION (IN NUMBER OF OSCILLATOR CLOCKS) Table 19

	Osc. cycles per machine cycle	**Osc. cycles per timers (0,1,2) clock**			**Osc. cycles per timer 2 clock**	**Osc. cycles per serial port clock Mode 0**		**Osc. cycles per serial port clock Mode 2**	
		TxMH,TxM=			**Baud Rate Gen.**				
4X/2X,CD1,CD0		**00**	**01**	**1x**	**T2MH,T2M=xx**	**SM2=0**	**SM2=1**	**SMOD=0**	**SMOD=1**
100	0.25	12	1	0.25	2	12	1	64	32
000	0.5	12	2	0.5	2	12	2	64	32
x01	1(reserved)								
x10	1 (default)	12	4	1	2	12	4	64	32
x11	1,024	12,288	4,096	1,024	2048	12,288	4096	65536	32768

Note:

x = don't care

RING OSCILLATOR

The DS89C420 incorporates a ring oscillator to allow the processor to recover instantly from the Stop mode. The oscillator typically runs at 10 MHz.

When the system is in Stop mode the crystal is disabled. When Stop mode is removed, the crystal requires a period of time to start up and stabilize. To allow the system to begin immediate execution of software following the removal of the Stop mode, the ring oscillator is used to supply a system clock until the crystal start-up time is satisfied. Once this time has passed the ring oscillator is switched off and the system clock is switched over to the crystal oscillator. This function is programmable and is enabled by setting the RGSL bit (EXIF.1) to logic 1. When it is logic 0, the processor will delay software execution until after the 65,536 crystal clock periods. To allow the processor to know whether it is being clocked by the ring or the crystal oscillator, an additional bit termed the RGMD bit indicates which clock source is being used. When the processor is running from the ring, the Clock Divide Control bits (CD1 and CD0 in the PMR register) are locked into the divide by 1 mode (CD1:CD0=10b). The Clock Divide Control bits cannot be changed from this state until after the system clock transitions to the crystal oscillator (RGMD=0).

Note:

The Watchdog is permanently connected to the crystal oscillator and will continue to run at the external clock rate. It is not driven by the ring oscillator.

IDLE MODE

Idle mode suspends the processor by holding the program counter in a static state. No instructions are fetched and no processing occurs. Setting the IDLE bit (PCON.0) to logic 1 invokes the Idle mode. The instruction that executes this step is the last instruction prior to freezing the program counter. Once in Idle mode, all resources are preserved but all peripheral clocks remain active and the timers, Watchdog, Serial Ports and Power Monitor functions continue to operate, so that the processor can exit the Idle Mode using any interrupt sources that are enabled. The oscillator detect circuit also continues to function when enabled. The IDLE bit is cleared automatically once the Idle mode is exited. On returning from the interrupt vector using the RETI instruction, the next address will be the one that immediately follows the instruction that invoked the Idle Mode. Any reset of the processor will also remove the Idle mode.

STOP MODE

The Stop mode disables all circuits within the processor. All on-chip clocks, timers and serial port communication are stopped, and no processing is possible.

Stop mode is invoked by setting the STOP bit (PCON.1) to logic 1. The processor enters the Stop mode on the instruction that sets the bit. The processor can exit Stop mode by using any of the six external interrupts that are enabled.

An external reset via the RST pin will unconditionally exit the processor from Stop mode. If the BGS bit is set to logic 1, the bandgap will provide a reset while in Stop mode if Vcc should drop below the Vrst level. If BGS is 0, no reset will be generated if Vcc drops below Vrst.

When the stop mode is removed, the processor will wait for TBD us for the internal Flash memory to warm-up before starting normal execution. Also, the processor waits for the crystal warm-up period if not using the ring oscillator.

SERIAL I/O

The DS89C420 provides a serial port (UART) that is identical to the 80C52. In addition it includes a second hardware serial port that is a full duplicate of the standard one. This port optionally uses pins P1.2 (RXD1) and P1.3 (TXD1). It has duplicate control functions included in new SFR locations.

Both ports can operate simultaneously but can be at different baud rates or even in different modes. The second serial port has similar control registers (SCON1 at C0h, SBUF1 at C1h) to the original. The new serial port can only use timer 1 for timer generated baud rates.

Control for serial port 0 is provided by the SCON0 register while its I/O buffer is SBUF0. The registers SCON1 and SBUF1 provide the same functions for the second serial port. A full description of the use and operation of both serial ports may be found in the *"High-Speed Microcontroller User's Guide."*

INSTRUCTION SET

The DS89C420 instructions are 100% binary compatible with the industry standard 8051, and are only different in the number of machine cycles used for the instructions. There are some special conditions and features to be considered when analyzing the DS89C420 instruction set. Full details are given in the *DS89C420 User's Guide.*

Table 18, "Instruction Set," lists the instructions and the number of clock cycles required for execution in the DS89C420. Note that one additional clock cycle may be required if the PSW, SP, DPS, IE, EIE, IP0, IP1, EIP0, EIP1or IP register is ACTUALLY WRITTEN to by certain DIRECT ADDRESSING instructions. This applies to the instructions marked with an "*". The "JBC Bit" instruction requires one additional clock cycle to clear a bit (if that bit resides in one of the aformentioned SFRs) and if the jump is actually taken..

INSTRUCTION SET Table18

	Instruction Code											
Instruction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Hex**	**Bytes**	**Cycles**	**Explanation**
Arithmetic:												
ADD A, Rn	0	0	1	0	1	n2	n1	n0	28-2F	1	1	(A)=(A)+(Rn)
ADD A, direct	0	0	1	0	0	1	0	1	25	2	2	(A)=(A)+(direct)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
ADD A, @Ri	0	0	1	0	1	1	1	i	26-27	1	2	(A)=(A)+((Ri))
ADD A, #data	0	0	1	0	0	1	0	0	24	2	2	(A)=(A)+#data
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
ADDC A, Rn	0	0	1	1	1	n2	n1	n0	38-3F	1	1	(A)=(A)+(Rn)+(C)
ADDC A, direct	0	0	1	1	0	1	0	1	35	2	2	(A)=(A)+(direct)+(C)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
ADDC A, @Ri	0	0	1	1	0	1	1	i	36-37	1	2	(A)=(A)+((Ri))+(C)
ADDC A, #data	0	0	1	1	0	1	0	0	34	2	2	(A)=(A)+#data+(C)
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
SUBB A, Rn	1	0	0	1	1	n2	n1	n0	98-9F	1	1	(A)=(A)-(Rn)-(C)
SUBB A, direct	1	0	0	1	0	1	0	1	95	2	2	(A)=(A)-(direct)-(C)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
SUBB A, @Ri	1	0	0	1	0	1	1	i	96-97	1	2	(A)=(A)-((Ri))-(C)
SUBB A, #data	1	0	0	1	0	1	0	0	94	2	2	(A)=(A)-#data-(C)
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
INC A	0	0	0	0	0	1	0	0	04	1	1	(A)=(A)+1
INC Rn	0	0	0	0	1	n2	n1	n0	08-0F	1	1	(Rn)=(Rn)+1
INC direct	0	0	0	0	0	1	0	1	05	2	2*	(direct)=(direct)+1
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
INC @Ri	0	0	0	0	0	1	1	i	06-07	1	2	((Ri))=((Ri))+1
INC DPTR	1	0	1	0	0	0	1	1	A3	1	1	(DPTR)=(DPTR)+1
DEC A	0	0	0	1	0	1	0	0	14	1	1	(A)=(A)-1
DEC Rn	0	0	0	1	1	n2	n1	n0	18-1F	1	1	(Rn)=(Rn)-1
DEC direct	0	0	0	1	0	1	0	1	15	2	2*	(direct)=(direct)-1
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
DEC @Ri	0	0	0	1	0	1	1	i	16-17	1	2	((Ri))=((Ri))-1
MUL AB	1	0	1	0	0	1	0	0	A4	1	9	(B15-8), (A7-0) =(A)x(B)
DIV AB	1	0	0	0	0	1	0	0	84		10	(B15-8), (A7-0) =(A)/(B)
DA A	1	1	0	1	0	1	0	0	D4	1	2	If [(A3-0)>9 or (AC)=1], then (A3-0)=(A3-0)+6 And If [(A7-4)>9 or ©=1], then (A7-4)=(A7-4)+6

Instruction	Instruction Code D7 D6 D5 D4 D3 D2 D1 D0	Hex	Bytes	Cycles	Explanation
Logical:					
ANL A, Rn	0 1 0 1 1 n2 n1 n0	58-5F	1	1	(A)=(A) AND (Rn)
ANL A, direct	0 1 0 1 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	55 Byte 2	2	2	(A)=(A) AND (direct)
ANL A, @Ri	0 1 0 1 0 1 1 i	56-57	1	2	(A)=(A) AND ((Ri))
ANL A, #data	0 1 0 1 0 1 0 0 d7 d6 d5 d4 d3 d2 d1 d0	54 Byte 2	2	2	(A)=(A) AND #data
ANL direct, A	0 1 0 1 0 0 1 0 a7 a6 a5 a4 a3 a2 a1 a0	52 Byte 2	2	2*	(direct)=(direct) AND A
ANL direct, #data	0 1 0 1 0 0 1 1 a7 a6 a5 a4 a3 a2 a1 a0 d7 d6 d5 d4 d3 d2 d1 d0	53 Byte 2 Byte 3	3	3	(direct)= (direct) AND #data
ORL A, Rn	0 1 0 0 1 n2 n1 n0	48-4F	1	1	(A)=(A) OR (Rn)
ORL A, direct	0 1 0 0 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	45 Byte 2	2	2	(A)=(A) OR (direct)
ORL A, @Ri	0 1 0 0 0 1 1 i	46-47	1	2	(A)=(A) OR ((Ri))
ORL A, #data	0 1 0 0 0 1 0 0 d7 d6 d5 d4 d3 d2 d1 d0	44 Byte 2	2	2	(A)=(A) OR #data
ORL direct, A	0 1 0 0 0 0 1 0 a7 a6 a5 a4 a3 a2 a1 a0	42 Byte 2	2	2*	(direct)=(direct) OR A
ORL direct, #data	0 1 0 0 0 0 1 1 a7 a6 a5 a4 a3 a2 a1 a0 d7 d6 d5 d4 d3 d2 d1 d0	43 Byte 2 Byte 3	3	3	(direct)= (direct) OR #data
XRL A, Rn	0 1 1 0 1 n2 n1 n0	68-6F	1	1	(A)=(A) XOR (Rn)
XRL A, direct	0 1 1 0 0 1 0 1 a7 a6 a5 a4 a3 a2 a1 a0	65 Byte 2	2	2	(A)=(A) XOR (direct)
XRL A, @Ri	0 1 1 0 0 1 1 i	46-67	1	2	(A)=(A) XOR ((Ri))
XRL A, #data	0 1 1 0 0 1 0 0 d7 d6 d5 d4 d3 d2 d1 d0	64 Byte 2	2	2	(A)=(A) XOR #data
XRL direct, A	0 1 1 0 0 0 1 0 a7 a6 a5 a4 a3 a2 a1 a0	62 Byte 2	2	2*	(direct)=(direct) XOR A
XRL direct, #data	0 1 1 0 0 0 1 1 a7 a6 a5 a4 a3 a2 a1 a0 d7 d6 d5 d4 d3 d2 d1 d0	63 Byte 2 Byte 3	3	3	(direct)= (direct) XOR #data
CLR A	1 1 1 0 0 1 0 0	E4	1	1	(A)=0
CPL A	1 1 1 1 0 1 0 0	F4	1	1	(A)=($\overline{A}$)
RL A	0 0 1 0 0 0 1 1	23	1	1	(A7-0)=(A6-0,7)
RLC A	0 0 1 1 0 0 1 1	33	1	1	(A7-0)=(A6-0),© And ©=(A7)
RR A	0 0 0 0 0 0 1 1	03	1	1	(A7-0)=(A0,7-1)
RRC A	0 0 0 1 0 0 1 1	13	1	1	(A7-0)=©,(A7-1) And ©=(A0)

	Instruction Code											
Instruction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Hex**	**Bytes**	**Cycles**	**Explanation**
SWAP A	1	1	0	0	0	1	0	0	C4	1	1	(A3-0) <-> (A7-4)
Data Transfer:												
MOV A, Rn	1	1	1	0	1	n2	n1	n0	E8-EF	1	1	(A)=(Rn)
MOV A, direct	1	1	1	0	0	1	0	1	E5	2	2	(A)=(direct)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
MOV A, @Ri	1	1	1	0	0	1	1	i	E6-E7	1	2	(A)=((Ri))
MOV A, #data	0	1	1	1	0	1	0	0	74	2	2	(A)=#data
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
MOV Rn, A	1	1	1	1	1	n2	n1	n0	F8-FF	1	1	(Rn)=(A)
MOV Rn direct	1	0	1	0	1	n2	n1	n0	A8-AF	2	2	(Rn)=(direct)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
MOV Rn, #data	0	1	1	1	1	n2	n1	n0	78-7F	2	2	(Rn)=#data
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
MOV direct, A	1	1	1	1	0	1	0	1	F5	2	2*	(direct)=(A)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
MOV direct, Rn	1	0	0	0	1	n2	n1	n0	88-8F	2	2*	(direct)=(Rn)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
MOV direct1, direct2	1	0	0	0	0	1	0	1	85	3	3*	(direct1)=(direct2)
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 3			
MOV direct, @Ri	1	0	0	0	0	1	1	i	86-87	2	2*	(direct)=((Ri))
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
MOV direct, #data	0	1	1	1	0	1	0	1	75	3	3	(direct)=#data
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 3			
MOV @Ri, A	1	1	1	1	0	1	1	i	F6-F7	1	1	((Ri))=(A)
MOV @Ri, direct	1	0	1	0	0	1	1	i	A6-A7	2	2	((Ri))=(direct)
MOV @Ri, #data	0	1	1	1	1	1	1	i	76-77	2	2	((Ri))=#data
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 2			
MOV DPTR, #data16	1	0	0	1	0	0	0	0	90	3	3	(DPH)=#data15-8
	d15	d14	d13	d12	d11	d10	d9	d8	Byte 2			(DPL)=#data7-0
	d7	d6	d5	d4	d3	d2	d1	d0	Byte 3			
MOVC A, @A+DPTR	1	0	0	1	0	0	1	1	93	1	3	(A)=((A)+(DPTR))
MOVC A, @A+PC	1	0	0	0	0	0	1	1	83	1	3	(A)=((A)+(PC))
MOVX A, @Ri	1	1	1	0	0	0	1	i	E2-E3	1	2	(A)=((Ri)), Data transfer
MOVX A,@DPTR	1	1	1	0	0	0	0	0	E0	1	2	(A)=((DPTR)), Data transfer
MOVX @Ri, A	1	1	1	1	0	0	1	i	F2-F3	1	2	((Ri))=(A), Data transfer
MOVX @DPTR,A	1	1	1	1	0	0	0	0	F0	1	2	((DPTR))=(A), Data transfer

	Instruction Code										
Instruction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Hex**	**Bytes**	**Cycles**	**Explanation**
PUSH direct	1 a7	1 a6	0 a5	0 a4	0 a3	0 a2	0 a1	0 a0	C0 Byte 2	2	2	(SP)=(SP)+1, ((SP))=(direct)
POP direct	1 a7	1 a6	0 a5	1 a4	0 a3	0 a2	0 a1	0 a0	D0 Byte 2	2	2*	(direct)=((SP)), (SP)=(SP)-1
XCH A, Rn	1	1	0	0	1	n2	n1	n0	C8-CF	1	2	(A) ↔ (Rn)
XCH A, direct	1 a7	1 a6	0 a5	0 a4	0 a3	1 a2	0 a1	1 a0	C5 Byte 2	2	3	(A) ↔ (direct)
XCH A, @Ri	1	1	0	0	0	1	1	i	C6-C7	1	3	(A) ↔ ((Ri))
XCHD A, @Ri	1	1	0	1	0	1	1	i	D6-D7	1	3	(A3-0) ↔ ((Ri 3-0))
Boolean:												
CLR C	1	1	0	0	0	0	1	1	C3	1	1	(C)=0
CLR bit	1 b7	1 b6	0 b5	0 b4	0 b3	0 b2	1 b1	0 b0	C2 Byte 2	2	2*	(bit)=0
SETB C	1	1	0	1	0	0	1	1	D3	1	1	(C)=1
SETB bit	1 b7	1 b6	0 b5	1 b4	0 b3	0 b2	1 b1	0 b0	D2 Byte 2	2	2*	(bit)=1
CPL C	1	0	1	1	0	0	1	1	B3	1	1	(C) = $\overline{(C)}$
CPL bit	1 b7	0 b6	1 b5	1 b4	0 b3	0 b2	1 b1	0 b0	B2 Byte 2	2	2*	(bit)=($\overline{\text{bit}}$)
ANL C, bit	1 b7	0 b6	0 b5	0 b4	0 b3	0 b2	1 b1	0 b0	82 Byte 2	2	2	(C)=(C) AND (bit)
ANL C, $\overline{\text{bit}}$	1 b7	0 b6	1 b5	1 b4	0 b3	0 b2	0 b1	0 b0	B0 Byte 2	2	2	(C)=(C) AND (bit)
ORL C, bit	0 b7	1 b6	1 b5	1 b4	0 b3	0 b2	1 b1	0 b0	72 Byte 2	2	2	(C)=(C) OR (bit)
ORL C, $\overline{\text{bit}}$	1 b7	0 b6	1 b5	0 b4	0 b3	0 b2	0 b1	0 b0	A0 Byte 2	2	2	(C)=(C) OR (bit)
MOV C, bit	1 b7	0 b6	1 b5	0 b4	0 b3	0 b2	1 b1	0 b0	A2 Byte 2	2	2	(C)=(bit)
MOV bit, C	1	0	0	1	0	0	1	0	92 Byte 2	2	2	(bit)=(C)
Branching:												
ACALL addr11	a10 a7	a9 a6	a8 a5	1 a4	0 a3	0 a2	0 a1	1 a0	Byte 1 Byte 2	2	2	(PC)=(PC)+2 (SP)=(SP)+1 ((SP))=(PC7-0) (SP)=(SP)+1 ((SP))=(PC15-8) (PC10-0)=addr11

	Instruction Code											
Instruction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Hex**	**Bytes**	**Cycles**	**Explanation**
LCALL addr16	0	0	0	1	0	0	1	0	12	3	3	(PC)=(PC)+3
	a15	a14	a13	a12	a11	a10	a9	a8	Byte 2			(SP)=(SP)+1
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 3			((SP))=(PC7-0)
												(SP)=(SP)+1
												((SP))=(PC15-8)
												(PC)=addr16
RET	0	0	1	0	0	0	1	0	22	1	3	(PC15:8)=((SP))
												(SP)=(SP)-1
												(PC7:0)=((SP))
												(SP)=(SP)-1
RETI	0	0	1	1	0	0	1	0	32	1	3	(PC15:8)=((SP))
												(SP)=(SP)-1
												(PC7:0)=((SP))
												(SP)=(SP)-1
AJMP addr11	a10	a9	a8	0	0	0	0	1	Byte 1	2	2	(PC)=(PC)+2
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 2			(PC10-0)=addr11
LJMP addr16	0	0	0	0	0	0	1	0	02	3	3	(PC)=addr16
	a15	a14	a13	a12	a11	a10	a9	a8	Byte 2			
	a7	a6	a5	a4	a3	a2	a1	a0	Byte 3			
SJMP rel	1	0	0	0	0	0	0	0	80	2	3	(PC)=(PC)+2
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 2			(PC)=(PC)+rel
JMP @A+DPTR	0	1	1	1	0	0	1	1	73	1	3	(PC)=(A)+(DPTR)
JZ rel	0	1	1	0	0	0	0	0	60	2	3	(PC)=(PC)+2,
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 2			If (A)=0,
												then (PC)=(PC)+rel
JNZ rel	0	1	1	1	0	0	0	0	70	2	3	(PC)=(PC)+2,
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 2			If (A)<>0,
												then (PC)=(PC)+rel
JC rel	0	1	0	0	0	0	0	0	40	2	3	(PC)=(PC)+2,
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 2			If ©=1,
												then (PC)=(PC)+rel
JNC rel	0	1	0	1	0	0	0	0	50	2	3	(PC)=(PC)+2,
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 2			If ©=0,
												then (PC)=(PC)+rel
JB bit, rel	0	0	1	0	0	0	0	0	20	3	4	(PC)=(PC)+3,
	b7	b6	b5	b4	b3	b2	b1	b0	Byte 2			If (bit)=1,
	r7	r6	r5	r4	r3	r2	r1	r0	Byte 3			then (PC)=(PC)+rel
JNB bit, rel	0	0	1	1	0	0	0	0	30	3	4	(PC)=(PC)+3,
	b7	b6	b5	b4	b3	b2	b1	b0	Byte 2			If (bit)=0,
	r7	r6	r5	r4	r3	r2	r1	r0				then (PC)=(PC)+rel

	Instruction Code											
Instruction	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Hex**	**Bytes**	**Cycles**	**Explanation**
JBC bit, rel	0 b7 r7	0 b6 r6	0 b5 r5	1 b4 r4	0 b3 r3	0 b2 r2	0 b1 r1	0 b0 r0	10 Byte 2 Byte 3	3	4*	(PC)=(PC)+3, If (bit)=1, then (bit)=0 and (PC)=(PC)+rel
CJNE A, direct, rel	1 a7 r7	0 a6 r6	1 a5 r5	1 a4 r4	0 a3 r3	1 a2 r2	0 a1 r1	1 a0 r0	B5 Byte 2 Byte 3	3	5	(PC)=(PC)+3, If (direct)<(A), then ©=0 and (PC)=(PC)+rel, Or if (direct)>(A) then ©=1 and (PC)=(PC)+rel
CJNE A, #data, rel	1 d7 r7	0 d6 r6	1 d5 r5	1 d4 r4	0 d3 r3	1 d2 r2	0 d1 r1	0 d0 r0	B4 Byte 2 Byte 3	3	4	(PC)=(PC)+3, If #data<(A), then ©=0 and (PC)=(PC)+rel, Or if #data>(A) then ©=1 and (PC)=(PC)+rel
CJNE Rn, #data, rel	1 d7 r7	0 d6 r6	1 d5 r5	1 d4 r4	0 d3 r3	n2 d2 r2	n1 d1 r1	n0 d0 r0	B8-BF Byte 2 Byte 3	3	4	(PC)=(PC)+3, If #data<(Rn), then ©=0 and (PC)=(PC)+rel, Or if (#data)>(Rn) then ©=1 and (PC)=(PC)+rel
CJNE Ri, #data, rel	1 d7 r7	0 d6 r6	1 d5 r5	1 d4 r4	0 d3 r3	1 d2 r2	1 d1 r1	i d0 r0	B6-B7 Byte 2 Byte 3	3	5	(PC)=(PC)+3, If (#data)<((Ri)), then ©=0 and (PC)=(PC)+rel, Or if (#data)>((Ri)) then ©=1 and (PC)=(PC)+rel
DJNZ Rn, rel	1 r7	1 r6	0 r5	1 r4	1 r3	n2 r2	n1 r1	n0 r0	D8-DF Byte 2	2	4	(PC)=(PC)+2, (Rn)=(Rn)-1, If (Rn)<>0, then (PC)=(PC)+rel
DJNZ direct, rel	1 a7 r7	1 a6 r6	0 a5 r5	1 a4 r4	0 a3 r3	1 a2 r2	0 a1 r1	1 a0 r0	D5 Byte 2 Byte 3	3	5	(PC)=(PC)+3, (direct)=(direct)-1, If (direct)<>0, then (PC)=(PC)+rel
NOP	0	0	0	0	0	0	0	0	00	1	1	(PC)=(PC)+1

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.3V to (Vcc + 0.5V)
Voltage on Vcc relative to Ground	-0.3V to +6.0V
Operating Temperature	-40^{O}C to + 85^{O}C
Storage Temperature	-55^{O}C to + 125^{O}C
Soldering Temperature	260^{O}C for 10 Seconds

* This is a stress rating only and functional operation of the device at these or other conditions above those indicated in the operations sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS Table 29

Parameter	Symbol	Min	Typ	Max	Units	Notes
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	1
Power Fail Warning	V_{PFW}	4.25	4.38	4.5	V	1
Min Operating Voltage	V_{RST}	4.0	4.13	4.25	V	1
Supply Current Active Mode @ 50 MHz	I_{CC}		100		mA	2
Supply Current Idle Mode @ 50 MHz	I_{IDLE}		15		mA	3
Supply Current Stop Mode Band-gap Disabled	I_{STOP}		10		µA	4
Supply Current Stop Mode Bandgap Enabled	I_{SPBG}		75		µA	4
Input Low Level	V_{IL}	-0.3		+0.8	V	1
Input High Level	V_{IH}	2.0		V_{DD}+0.3	V	1
Input High Level XTAL and RST	V_{IH2}	3.5		V_{DD}+0.3	V	1
Output Low Voltage, Port 1 and 3 @ IOL=1.6 mA	V_{OL1}		0.15	0.45	V	1
Output Low Voltage, Port 0 and 2, ALE, $\overline{\text{PSEN}}$ @ IOL=3.2 mA	V_{OL2}		0.15	0.45	V	1
Output High Voltage, Port 1, 2 and 3, ALE, $\overline{\text{PSEN}}$ @ IOH=-50 uA	V_{OH1}	2.4			V	1, 6
Output High Voltage, Port 1, 2 and 3 IOH=-1.5 mA	V_{OH2}	2.4			V	1, 7
Output High Voltage, Port 0 in Bus Mode @ IOH=-8 mA	V_{OH3}	2.4			V	1, 5
Input Low Current, Port 1, 2 and 3, @ 0.45V	I_{IL}			-55	µA	
Transition Current from 1 to 0, Port 1, 2 and 3 @ 2V	I_{TL}			-650	µA	8
Input Leakage Current, Port 0 in I/O Mode and $\overline{\text{EA}}$	I_{L}	-10		+10	µA	10
Input Leakage Current, Port 0 in Bus Mode	I_{L}	-300		+300	µA	9
RST Pull-down Resistance	R_{RST}	50		170	kΩ	10

NOTES:

1. All voltages are referenced to ground.
2. Active current is measured with a 50 MHz clock source driving XTAL1, Vcc= $\overline{\text{RST}}$ =5.5V. All other pins disconnected.
3. Idle Mode current measured with a 50 MHz clock source driving XTAL1, Vcc =5.5V, $\overline{\text{RST}}$ at ground. All other pins disconnected.
4. Stop Mode measured with XTAL and $\overline{\text{RST}}$ grounded, Vcc = 5.5V. All other pins disconnected. This value is not guaranteed. Users who are sensitive to this specification should contact Dallas Semiconductor for more information.
5. When addressing external memory.
6. $\overline{\text{RST}}$ =5.5V. This condition mimics the operation of pins in I/O mode. Port 0 is tri-state in reset and when in a logic high state during I/O mode.
7. During a 0 to 1 transition, a one shot drives the ports hard for two clock cycles. This measurement reflects a port pin in transition mode.
8. Ports 1, 2 and 3 source transition current when being pulled down externally. The current reaches its maximum at approximately 2V.
9. 0.45V<Vin <Vcc. Not a high impedence input. This port is a weak address holding latch in bus mode. Peak current occurs near the input transition point of the holding latch at approximately 2V.
10. 0.45V<Vin <Vcc. RST = 5.5V. This condition mimics the operation of pins in I/O mode.

AC CHARACTERISTICS Table 20

Parameter	Symbol	Min	Max	Units
Operating Clock Frequency	$1/t_{CLCL}$	0	50	MHz
ALE Pulse Width	t_{LHLL}	$1.5t_{CLCL}$ -5		ns
Port 0 Address Valid to ALE Low	t_{AVLL}	$0.5t_{CLCL}$ -5		ns
Address hold after ALE Low	t_{LLAX1}	$0.5t_{CLCL}$ -5		ns
ALE Low to $\overline{\text{PSEN}}$ Low	t_{LLPL}	$0.5t_{CLCL}$ -5		ns
ALE Low to Valid Instruction In	t_{LLIV}		$2.5t_{CLCL}$ -20	ns
$\overline{\text{PSEN}}$ Pulse Width	t_{PLPH}	2 t_{CLCL} -5		ns
$\overline{\text{PSEN}}$ Low to Valid Instruction In	t_{PLIV}		$2t_{CLCL}$ -20	ns
Input Instruction Hold after $\overline{\text{PSEN}}$	t_{PXIX}	0		ns
Input Instruction Float after $\overline{\text{PSEN}}$	t_{PXIZ}		t_{CLCL} -5	ns
Port 0 Address to Valid Instruction In	t_{AVIV}		$3t_{CLCL}$ -20	ns
Port 2 Address to Valid Instruction In	t_{AVIV2}		$3.5t_{CLCL}$ -25	ns
$\overline{\text{PSEN}}$ Low to Address Float	t_{PLAZ}		0	ns

- The data given in Table 20 is for 4 clocks per cycle operation. Values for other modes are detailed in the *DS89C420 User's Guide*.
- All parameters apply to both commercial and industrial temperature operation unless otherwise noted.
- All signals characterized with load capacitance of 80 pF except port 0, ALE, $\overline{\text{PSEN}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ with 100 pF.
- Interfacing to memory devices with float times (turn off times) over 35 ns may cause bus contention. This will not damage the part but will cause an increase in operating current.

MOVX AC CHARACTERISTICS Table 21

Parameter	Symbol	Min	Max	Units	Stretch
Data Access ALE Pulse Width	t_{LHLL2}	1.5 t_{CLCL} -5		ns	
Address Hold after ALE Low for MOVX Write	t_{LLAX2}	0.5 t_{CLCL} -5		ns	
$\overline{RD}$ Pulse Width	t_{RLRH}	2 t_{CLCL} -5 (2SV) t_{CLCL} -10		ns	SV=0 SV>1
$\overline{WR}$ Pulse Width	t_{WLWH}	2 t_{CLCL} -5 (2SV) t_{CLCL} -10		ns	SV=0 SV>1
$\overline{RD}$ Low to Valid Data In	t_{RLDV}		t_{MCS} -20	ns	
Data Hold after Read	t_{RHDX}	0		ns	
Data Float after Read	t_{RHDZ}		t_{CLCL} -5 (SV+1) t_{CLCL} -5	ns	SV=0 SV>1
ALE Low to Valid Data In	t_{LLDV}		2.5 t_{CLCL} -20 (SV+2.5) t_{CLCL} -40	ns	SV=0 SV>1
Port 0 Address to Valid Data In	t_{AVDV1}		3 t_{CLCL} -20 (SV+2.5) t_{CLCL} -20	ns	SV=0 SV>1
Port 2 Address to Valid Data In	t_{AVDV2}		3.5 t_{CLCL} -20 (SV+3) t_{CLCL} -20	ns	SV=0 SV>1
ALE Low to $\overline{RD}$ / $\overline{WR}$ Low	t_{LLWL}	0.5 t_{CLCL} -5 (SV) t_{CLCL} -5	0.5 t_{CLCL} +5 (SV) t_{CLCL} +5	ns	SV=0 SV>1
Port 0 Address to $\overline{RD}$ / $\overline{WR}$ Low	t_{AVWL1}	t_{CLCL} -5 (SV+1) t_{CLCL} -5		ns	SV=0 SV>1
Port 2 Address to $\overline{RD}$ / $\overline{WR}$ Low	t_{AVWL2}	1.5tCLCL-10 (SV+1.5) t_{CLCL} -10		ns	SV=0 SV>1
Data Valid to WR Transition	t_{QVWX}	-5		ns	
Data Hold after Write	t_{WHQX}	t_{CLCL} -5 (SV+1) t_{CLCL} -5		ns	SV=0 SV>1
$\overline{RD}$ Low to Address Float	t_{RLAZ}		-0.5 t_{CLCL} -5	ns	
$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{WHLH}	0 (SV) t_{CLCL} -5	10 (SV) t_{CLCL} +5	ns	SV=0 SV>1

NOTE:

SV is the Stretch Value represented by the setting of MD2:0 bits for the stretch data memory cycle as shown in Table 22.

STRETCH VALUES Table 22

MD2	MD1	MD0	Stretch Value (SV)
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

EXTERNAL CLOCK CHARACTERISTICS Table 23

Parameter	Symbol	Min	Max	Units
Clock High Time	t_{CHCX}	6		ns
Clock Low Time	t_{CLCX}	6		ns
Clock Rise Time	t_{CLCH}		4	ns
Clock Fall Time	t_{CHCL}		4	ns

SERIAL PORT MODE 0 TIMING CHARACTERISTICS Table 24

Parameter	Symbol	Min	Typ	Max	Units
Clock Cycle Time SM2=0 SM2=1	t_{XLXL}		 $12t_{CLCL}$ $4t_{CLCL}$		ns
Output Data Setup to Clock Rising SM2=0 SM2=1	t_{QVXH}		 $12t_{CLCL}$ $4t_{CLCL}$		ns
Output Data Hold to Clock Rising SM2=0 SM2=1	t_{XHQX}		 $12t_{CLCL}$ $4t_{CLCL}$		ns
Input Data Hold after Clock Rising SM2=0 SM2=1	t_{XHDX}		 $12t_{CLCL}$ $4t_{CLCL}$		ns
Clock Rising Edge to Input Data Valid SM2=0 SM2=1	t_{XHDV}		 $12t_{CLCL}$ $4t_{CLCL}$		ns

NOTE:

SM2 is the Serial Port 0 Mode Bit 2. When Serial Port 0 is operating in mode 0 (SM0=SM1=0), SM2 determines the number of crystal clocks in a serial port clock cycle.

POWER CYCLE TIMING CHARACTERISTICS Table 26

Parameter	Symbol	Min	Typ	Max	Units	Notes
Crystal Start-up Time	t_{CSU}		1.8		ms	1
Power-on Reset Delay	t_{POR}			65536	t_{CLCL}	2

NOTES:

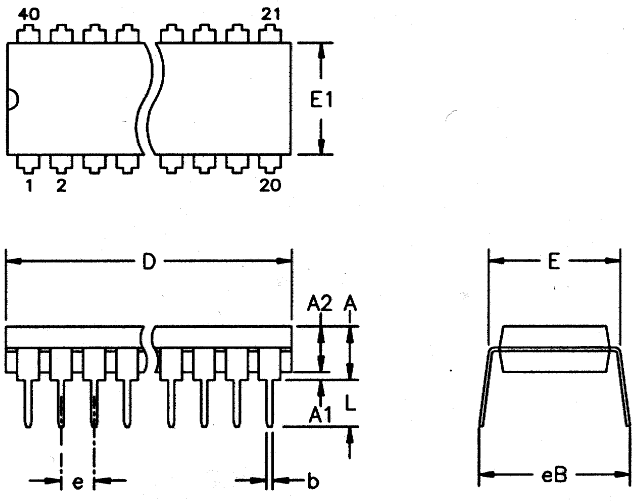
1. Start-up time for a crystal varies with load capacitance and manufacturer. Time shown is for an 11.0592 MHz crystal manufactured by Fox Electronics.

2. Reset delay is a synchronous counter of crystal oscillations during crystal start-up. At 50 MHz, this time is 1.310 ms.

FLASH MEMORY CHARACTERISTICS Table 27

Parameter	Symbol	Min	Typ	Max	Units
Programming Voltage	V_{PP}	4.5		5.5	V
Programming Supply Current	I_{PP}			40	mA
Address Setup to $\overline{\text{PROG}}$ Low	t_{AVGL}	20			ns
Address Hold after $\overline{\text{PROG}}$	t_{GHAX}	20			µs
Data Setup to $\overline{\text{PROG}}$ Low	t_{DVGL}	20			ns
Data Hold after $\overline{\text{PROG}}$	t_{GHDX}	20			ns
$\overline{\text{PROG}}$ Pulse Width	t_{GLGH}	85	16	100	µs
Address to Data Valid	t_{AVQV}			15	µs
Enable Low to Data Valid	t_{ELQV}			15	ns
Data Float after Enable	t_{EHQZ}	0		2	µs
$\overline{\text{PROG}}$ High to $\overline{\text{PROG}}$ Low	t_{GHGL}	6			µs

40-PIN PDIP (600-MIL)



DIMENSIONS ARE IN INCHES.

PKG	40-PIN	
DIM	MIN	MAX
A	-	0.200
A1	0.015	-
A2	0.140	0.160
b	0.014	0.022
c	0.008	0.012
D	1.980	2.085
E	0.600	0.625
E1	0.530	0.555
e	0.090	0.110
L	0.115	0.145
eB	0.600	0.700

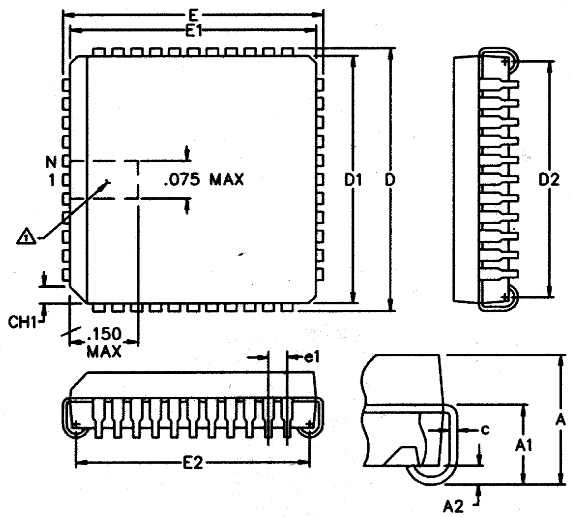
56-G5000-000

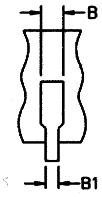
44-PIN PLCC

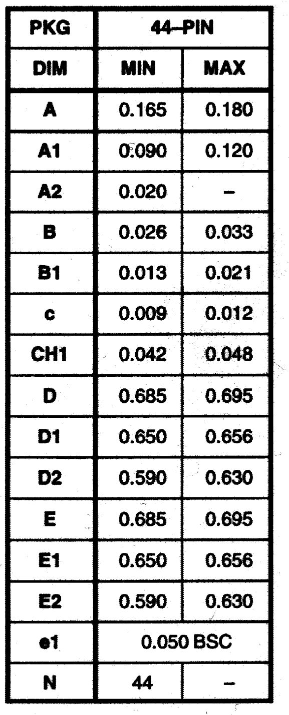
NOTES:

△1. Pin-1 identifier to be located in zone indicated

2. Controlling dimensions are in inches

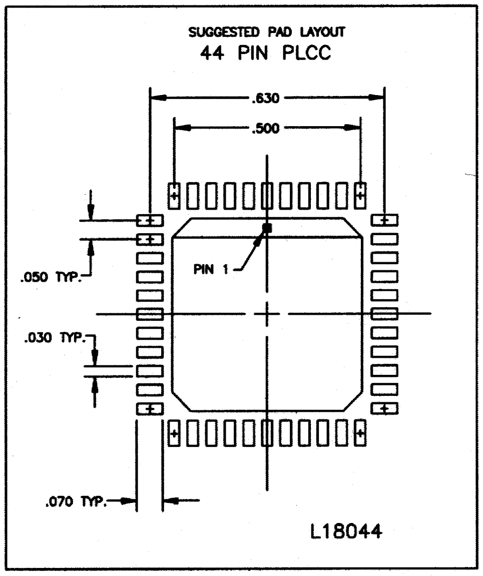




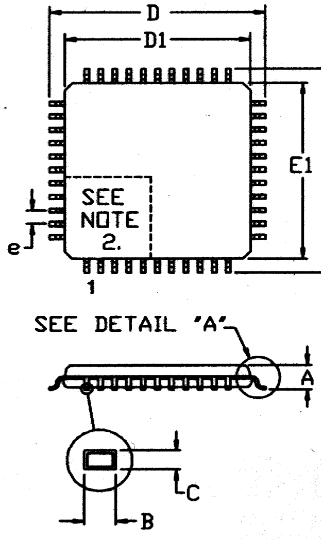


PKG	44–PIN	
DIM	MIN	MAX
A	0.165	0.180
A1	0.090	0.120
A2	0.020	–
B	0.026	0.033
B1	0.013	0.021
c	0.009	0.012
CH1	0.042	0.048
D	0.685	0.695
D1	0.650	0.656
D2	0.590	0.630
E	0.685	0.695
E1	0.650	0.656
E2	0.590	0.630
e1	0.050 BSC	
N	44	–

56–G4003–001

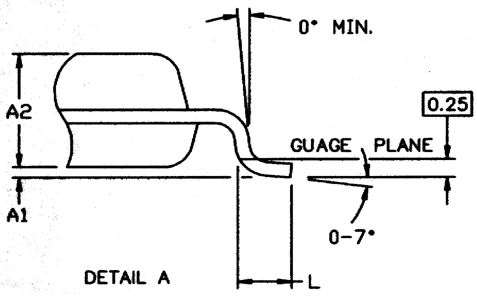


44 PIN TQFP



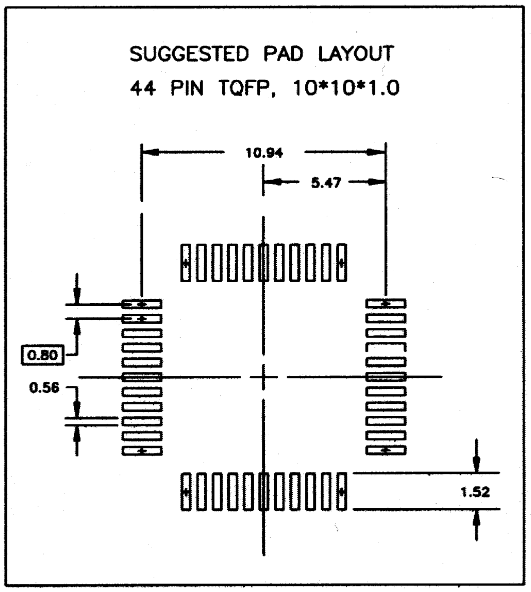
NOTES:

1. DIMENSIONS D1 AND E1 INCLUDE MOLD MISMATCH, BUT DO NOT INCLUDE MOLD PROTRUSION; ALLOWABLE PROTRUSION IS 0.25 MM PER SIDE.
2. DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL BUT MUST BE LOCATED WITHIN THE ZONE INDICATED.
3. ALLOWABLE DAMBAR PROTRUSION IS 0.08 MM TOTAL IN EXCESS OF THE B DIMENSION; AT MAXIMUM MATERIAL CONDITION. PROTRUSION NOT TO BE LOCATED ON LOWER RADIUS OR FOOT OF LEAD.
4. CONTROLLING DIMENSIONS: MILLIMETERS.



PKG	44-PIN	
DIM	MIN	MAX
A	–	1.20
A1	0.05	0.15
A2	0.95	1.05
D	11.80	12.20
D1	10.00 BSC	
E	11.80	12.20
E1	10.00 BSC	
L	0.45	0.75
e	0.80 BSC	
B	0.30	0.45
C	0.09	0.20

56–G4012–001



REVISION HISTORY

September 22, 2000 Original Issue

DALLAS SEMICONDUCTOR
ERRATA SHEET　DS89C420　Revision A2　06/15/01

Revision A2 may be identified by the date/revision brand yywwA2, where yy and ww are the year and workweek of manufacture, respectively. This errata sheet is valid only when used in conjunction with the most current version of the data sheet available from Dallas Semiconductor via the Internet.

This document contains the following types of information:

Errata: These are design errors that deviate from published specifications. Errata are intended to be fixed in subsequent revisions of the device.

Specification Modifications: These are changes to the published specifications and will be reflected in the next update of the documentation and apply to all subsequent revisions of the device.

Documentation Changes: This information includes typographical mistakes, errors, omissions or clarifications of device operation. Items listed in this section will be reflected in the next update of the documentation.

ERRATA

1. The maximum operating speed of the microcontroller core over the temperature range and voltage limits is 33 MHz. If the clock multiplier is used, the external oscillator or crystal frequency must be scaled appropriately so that the internal clock speed does not exceed 33 MHz.

   Work Around: None. Operating speed will be increased in later revisions.

2. The In-Application Programming feature is not functional when operating with program in internal flash memory or SRAM.

   Work Around: None. However this feature does operate normally in an external program memory

3. To enter loader mode, port pin P3.7 must be pulled high when the pin biasing combination of RST=1, $\overline{\text{PSEN}} = \overline{\text{EA}}$ =0 is applied.

   Work Around: None.

4. MOVC instructions executed from internal SRAM (locations 0400h-07FFh) when the internal SRAM is configured as program memory (PRAME=1) will not execute correctly.

   Work Around: None.

5. Under some circumstances, writes to parallel I/O ports can be delayed one clock cycle. In most cases, this will have no effect on system operation. If the immediately following instruction reads that modified port, however, the read instruction may read the port before its output value has changed. This behavior would differ from that of the traditional 8051.

Work Around: In most cases, none will be necessary. If the application software incorporates the unlikely occurrence of a port write immediately followed by a read of that port, a NOP can be inserted between the two instructions to incorporate sufficient delay to obviate the problem. This erratum will be fixed in the next revision.

6. The device will not fetch from internal flash memory if all three port pins P2.5, P2.6, and P2.7 are held low during a power-on reset.

   Work around: When internal code fetching is desired ($\overline{EA}$ =1), make certain that at least one of the three pins P2.5, P2.6, or P2.7 is not being held low during a power-on reset.

7. Any Read-Modify-Write (RMW) instruction that has P0, P1, P2, or P3 as its destination (such as ANL P0, #data) can incorrectly activate the strong internal pull-up for two clock cycles. This will only occur if a pin is at a logic 0 state and the result of the RMW instruction writes a logic 1 to the corresponding bit in the port latch, even if the bit was previously 1. If external logic connected to that pin is holding it at a logic 0, the strong internal driver can cause the pin to glitch momentarily to a logic 1. This behavior contradicts the data sheet which indicates that the strong internal pull-up should only be activated when a 0-to-1 transition is required on a port pin. A complete list of RMW instructions is contained in the DS89C420 User's Guide.

   Work around: When using a port pin as an input, always make certain that any external device driving a logic 0 can sink sufficient current to keep the pin voltage below +0.8V (V_{IL}) during the temporary activation of the internal pull-up (V_{OH2}).

8. Bootstrap loader mode will not operate correctly if any of the lock bits (LB1, LB2, LB3) are set.

   Work around: If any of the lock bits had been set by a previous session, erase the part using the K command, exit and reenter the bootstrap loader mode. The part will now accept bootstrap loader commands correctly.

SPECIFICATION MODIFICATIONS

1. NONE

DOCUMENTATION CHANGES

1. NONE

DALLAS SEMICONDUCTOR ERRATA SHEET DS89C420 Revision A1 06/15/01

Revision A1 may be identified by the date/revision brand yywwA1, where yy and ww are the year and workweek of manufacture, respectively. This errata sheet is valid only when used in conjunction with the most current version of the data sheet available from Dallas Semiconductor via the Internet.

This document contains the following types of information:

Errata: These are design errors that deviate from published specifications. Errata are intended to be fixed in subsequent revisions of the device.

Specification Modifications: These are changes to the published specifications and will be reflected in the next update of the documentation and apply to all subsequent revisions of the device.

Documentation Changes: This information includes typographical mistakes, errors, omissions or clarifications of device operation. Items listed in this section will be reflected in the next update of the documentation.

ERRATA

1. The execution of instructions with a cycle count that exceeds the byte count can cause incorrect device operation in the following conditions:
   a. The device is executing from internal (flash) memory, and
   b. an instruction opcode falls on a 256-byte page boundary (xxFFh), and
   c. the previous instruction is one of the following:

   ADDC A, @Ri
   MUL AB
   DA A
   ANL A, @Ri
   MOVC (all forms)
   XCH A, direct

   DEC @Ri
   DIV AB
   ORL A, @Ri
   XRL A, @Ri
   XCH A, Rn
   XCHD A, @Ri

   INC direct*
   ANL direct*, A
   XRL direct*, A
   MOV direct*, Rn
   MOV direct*, @Ri
   CLR bit
   CPL bit

   DEC direct*
   ORL direct*, A
   MOV direct*, A
   MOV direct1*, direct2
   POP direct*
   SETB bit

   *These instructions will cause a failure only when when the SFR is one of these:
   PSW, SP, DPS, IE, EIE, IP0, IP1, EIP0, or EIP1.
   Work Around: If programming in assembly language, write the software so that the code does not allow any of the above instructions to occur such that the next opcode falls on the last byte of

256-byte page (xxFFh) boundary. If writing in C, make all routines and functions less than 256 bytes in length and use byte constants placed at xxFFh intervals in the software. This condition will be corrected on revision A2 coming shortly.

2. The auto-baud routine of the bootstrap loader does not always function reliably (observed as garbled information instead of loader banner). Functionality is better at lower baud rates. Experiments show that the maximum reliable baud rate for any given crystal frequency is given by the equation: $f_{OSC}/1152$ = Max Baud

   Work Around: Use lower baud rates. This condition will be corrected on revision A2 coming shortly.

3. The signal $\overline{\text{PSEN}}$ is driven by a strong pull-up internally, and the component used to pull it down to enter loader mode must be capable of sinking 100 mA of current to ground.

   Work Around: None. This condition will be corrected on revision A2 coming shortly.

4. To enter loader mode, $\overline{\text{PSEN}}$ is sampled after Reset is pulled high and /EA is pulled low. This dictates that the transition of $\overline{\text{PSEN}}$ from high to low should be delayed from the onset of Reset and /EA by eight clock cycles. This timing relationship will be eliminated in future revisions (i.e., signals can be set simultaneously).

   Work Around: Use individual manual switches, passive circuitry (RC combinations), or active counting circuits (driven off of one of the processor's available clocks: ALE preferred) to delay the falling edge of $\overline{\text{PSEN}}$ so it will be sampled correctly (i.e., 8 oscillator clocks or more after Reset and $\overline{\text{EA}}$ are set). This condition will be corrected on revision A2 coming shortly.

5. If the oscillator is stopped or fails for any reason, the oscillator fail detect circuitry will cause a long delay in processor restart even though the oscillator fail detect reset function is disabled.

   Work Around: None. This condition will be corrected on revision A2 coming shortly.

6. Presently available sample devices (labeled ES) were tested successfully at 5.0 volts and at room temperature at a clock frequency of 45.0 MHz.

   Work Around: None. Operate these devices as close to these conditions as is reasonably possible.

7. The In-Application Programming feature is not functional. A design error causes bank 0 to read zeros while bank 1 is being erased or programmed.

   Work Around: None. This condition will be corrected on revision A2 coming shortly.

8. The In-System Programming feature is very frequency dependent. Reduced frequency will yield a high probability of success. Additionally, the feature appears to have some voltage and/or temperature dependencies. No specifics are available at this time.

   Work Around: Perform program and erase functions at lower crystal frequencies for greater success. Repeating the program or erase functions after a complete power down of the part seems to produce better results.

9. The Erase Flash command ("K") of the bootstrap loader can only be used once per bootloader session.

   Work Around: Reset the microcontroller before executing another Erase Flash command.

10. The Erase Flash command ("K") of the bootstrap loader may fail to completely erase the Flash memory under certain circumstances.

    Work Around: Use the CRC command ("C") after the Erase Flash command has completed to verify total erasure of the Flash. If completely erased, the CRC command should respond with "CRC = 4400". If the erasure was unsuccessful, reset the microcontroller and repeat the procedure.

11. To enter loader mode, port pin P3.7 must be pulled high when the pin biasing combination of RST=1, PSEN\=0, EA\=0 is applied.

    Work Around: None

12. The ROM firmware (version 0.9) does not support the ‘LB’ command nor does it allow the ‘R’ead or ‘W’rite of the ACON, CKCON, or PMR registers.

    Work Around: None

13. MOVC instructions executed from internal SRAM (locations 0400h-07FFh), when the internal SRAM is configured as program memory (PRAME=1), will not execute correctly.

    Work Around: None

14. In Bootstrap ROM loader mode, the ROM code can indicate that the chip is locked when it is not.

    Work around: Reset the device, press and release the enter key to obtain the 'sign-on' banner and then erase the flash (K command). If this does not clear the problem, remove and re-apply power to the device and then perform the steps listed in the preceding sentence. Repeat as needed. This condition will be corrected in the A2 revision.

15. The device will not fetch from internal flash memory if all three port pins P2.5, P2.6, and P2.7 are held low during a power-on reset.

    Work around: When internal code fetching is desired (EA\=1), make certain that at least one of the three pins P2.5, P2.6, or P2.7 is not being held low during a power-on reset.

16. Any Read-Modify-Write (RMW) instruction that has P0, P1, P2, or P3 as its destination (such as ANL P0, #data) can incorrectly activate the strong internal pullup for two clock cycles. This will only occur if a pin is at a logic 0 state and the result of the RMW instruction writes a logic 1 to the corresponding bit in the port latch, even if the bit was previously 1. If external logic connected to that pin is holding it at a logic 0, the strong internal driver can cause the pin to glitch momentarily to a logic 1. This behavior contradicts the data sheet which indicates that the strong internal pullup should only be activated when a 0-to-1 transition is required on a port pin. A complete list of RMW instructions is contained in the DS89C420 User's Guide.

Work around: When using a port pin as an input, always make certain that any external device driving a logic 0 can sink sufficient current to keep the pin voltage below +0.8V (VIL) during the temporary activation of the internal pullup (VOH2).

SPECIFICATION MODIFICATIONS

1. NONE

DOCUMENTATION CHANGES

1. NONE