Parallax Inc. - SX28AC/SS Datasheet

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	8-Bit
Speed	75MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	20
Program Memory Size	3KB (2K x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	136 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/parallax/sx28ac-ss

1.3 Architecture

The SX devices use a modified Harvard architecture. This architecture uses two separate memories with separate address buses, one for the program and one for data, while allowing transfer of data from program memory to SRAM. This ability allows accessing data tables from program memory. The advantage of this architecture is that instruction fetch and memory transfers can be overlapped with a multi-stage pipeline, which means the next instruction can be fetched from program memory while the current instruction is being executed using data from the data memory.

Ubicom has developed a revolutionary RISC-based architecture and memory design techniques that is 20 times faster than conventional MCUs, deterministic, jitter free, and totally reprogramable.

The SX family implements a four-stage pipeline (fetch, decode, execute, and write back), which results in execution of one instruction per clock cycle. For example, at the maximum operating frequency of 75 MHz, instructions are executed at the rate of one per 13.3ns clock cycle.

1.3.1 The Virtual Peripheral Concept

Virtual Peripheral concept enables the “software system on a chip” approach. Virtual Peripheral, a software module that replaces a traditional hardware peripheral, takes advantage of the Ubicom architecture’s high performance and deterministic nature to produce same results as the hardware peripheral with much greater flexibility.

The speed and flexibility of the Ubicom architecture complemented with the availability of the Virtual Peripheral library, simultaneously address a wide range of engineering and product development concerns. They decrease the product development cycle dramatically, shortening time to production to as little as a few days.

Ubicom’s time-saving Virtual Peripheral library gives the system designers a choice of ready-made solutions, or a head start on developing their own peripherals. So, with Virtual Peripheral modules handling established functions, design engineers can concentrate on adding value to other areas of the application.

The concept of Virtual Peripheral combined with in-system re-programmability provides a power development platform ideal for the communications industry because of the numerous and rapidly evolving standards and protocols.

Overall, the concept of Virtual Peripheral provides benefits such as using a more simple device, reduced component count, fast time to market, increased flexibility in design, customization to your application, and ultimately overall system cost reduction.

Some examples of Virtual Peripheral modules are:

- Communication interfaces such as I^{2}C™, Microwire™ (µ-Wire), SPI, IrDA Stack, UART, and Modem functions
- Frequency generation and measurement
- PPM/PWM output
- Delta/Sigma ADC
- DTMF generation/detection
- PSK/FSK generation/detection
- FFT/DFT based algorithms

1.3.2 The Communications Controller

The combination of the Ubicom hardware architecture and the Virtual Peripheral concept create a powerful, creative platform for the communications design communities: SX communications controller. Its high processing power, recofigurability, cost-effectiveness, and overall design freedom give the designer the power to build products for the future with the confidence of knowing that they can keep up with innovation in standards and other areas.

1.4 Programming and Debugging Support

The SX devices are currently supported by third party tool vendors. On-chip in-system debug capabilities have been added, allowing tools to provide an integrated development environment including editor, macro assembler, debugger, and programmer. Un-obtrusive in-system programming is provided through the OSC pins. There is no need for a bon-out chip, so the user does not have to worry about the potential variations in electrical characteristics of a bond-out chip and the actual chip used in the target applications. the user can test and revise the fully debugged code in the actual SX, in the actual application, and get to production much faster.

1.5 Applications

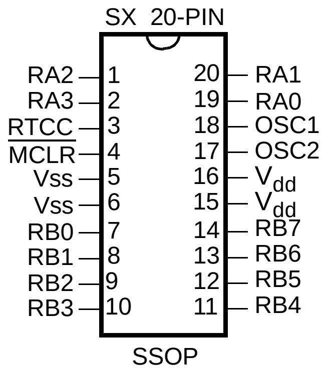
Emerging applications and advances in existing ones require higher performance while maintaining low cost and fast time-to-production.

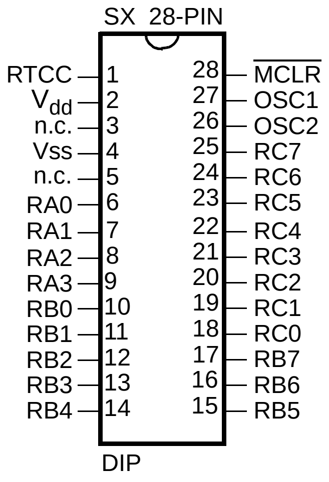
The device provides solutions for many familiar applications such as process controllers, electronic appliances/tools, security/monitoring systems, consumer automotive, sound generation, motor control, and personal communication devices. In addition, the device is suitable for applications that require DSP-like capabilities, such as closed-loop servo control (digital filters), digital answering machines, voice notation, interactive toys, and magnetic-stripe readers.

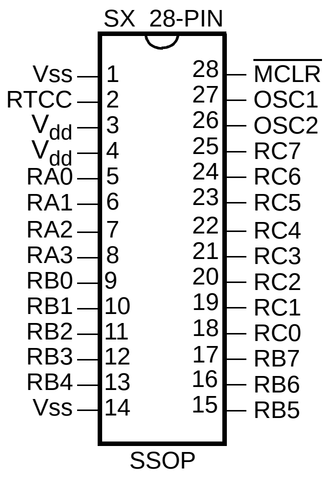
Furthermore, the growing Virtual Peripheral library features new components, such as the Internet Protocol stack, and communication interfaces, that allow design engineers to embed Internet connectivity into all of their products at extremely low cost and very little effort.

2.0 CONNECTION DIAGRAMS

2.1 Pin Assignments







2.2 Pin Descriptions

Name	Pin Type	Input Levels	Description
RA0	I/O	TTL/CMOS	Bidirectional I/O Pin; symmetrical source / sink capability
RA1	I/O	TTL/CMOS	Bidirectional I/O Pin; symmetrical source / sink capability
RA2	I/O	TTL/CMOS	Bidirectional I/O Pin; symmetrical source / sink capability
RA3	I/O	TTL/CMOS	Bidirectional I/O Pin; symmetrical source / sink capability
RB0	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; comparator output; MIWU/Interrupt input
RB1	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; comparator negative input; MIWU/Interrupt input
RB2	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; comparator positive input; MIWU/Interrupt input
RB3	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; MIWU/Interrupt input
RB4	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; MIWU/Interrupt input
RB5	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; MIWU/Interrupt input
RB6	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; MIWU/Interrupt input
RB7	I/O	TTL/CMOS/ST	Bidirectional I/O Pin; MIWU/Interrupt input
RC0	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC1	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC2	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC3	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC4	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC5	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC6	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RC7	I/O	TTL/CMOS/ST	Bidirectional I/O pin
RTCC	I	ST	Input to Real-Time Clock/Counter
$\overline{\text{MCLR}}$	I	ST	Master Clear reset input – active low
OSC1/In/Vpp	I	ST	Crystal oscillator input – external clock source input
OSC2/Out	O	CMOS	Crystal oscillator output – in R/C mode, internally pulled to V_{dd} through weak pull-up
V_{dd}	P	–	Positive supply pin
Vss	P	–	Ground pin
Note: I = input, O = output, I/O = Input/Output, P = Power, TTL = TTL input, CMOS = CMOS input, ST = Schmitt Trigger input, MIWU = Multi-Input Wakeup input			

2.3 Part Numbering

Table 2-1. Ordering Information

Device	Pins	I/O	Operating Frequency (MHz)	EE/Flash (Words)	RAM (Bytes)	Operating Temp. (°C)
SX20AC/SS	20	12	50	2K	136	-40°C to +85°C
SX20AC/SS	20	12	75	2K	136	0°C to +70°C
SX28AC/DP	28	20	50	2K	136	-40°C to +85°C
SX28AC/DP	28	20	75	2K	136	0°C to +70°C
SX28AC/SS	28	20	50	2K	136	-40°C to +85°C
SX28AC/SS	28	20	75	2K	136	0°C to +70°C

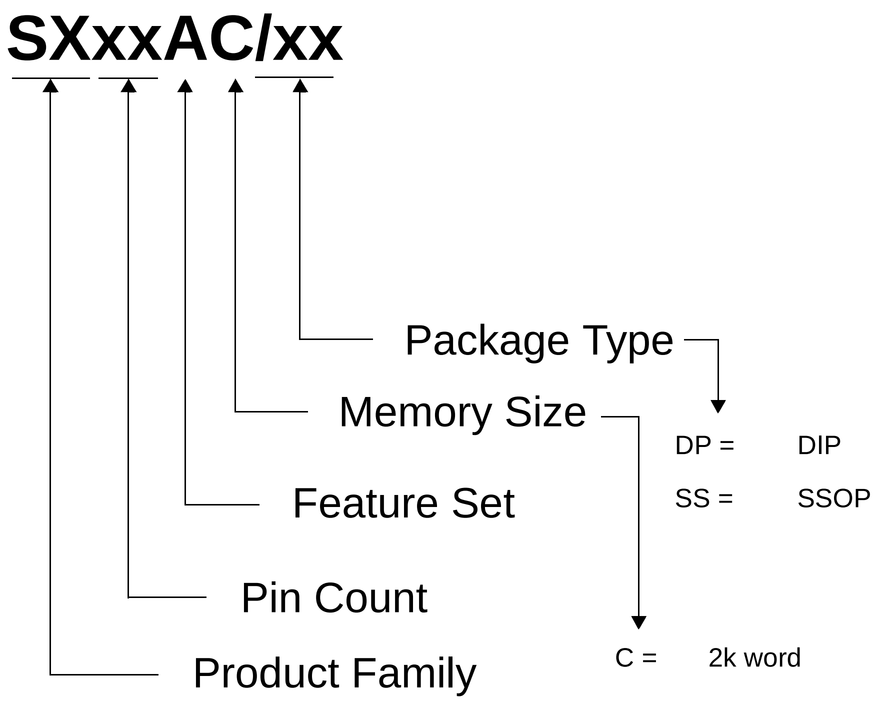


Figure 2-1. Part Number Reference Guide

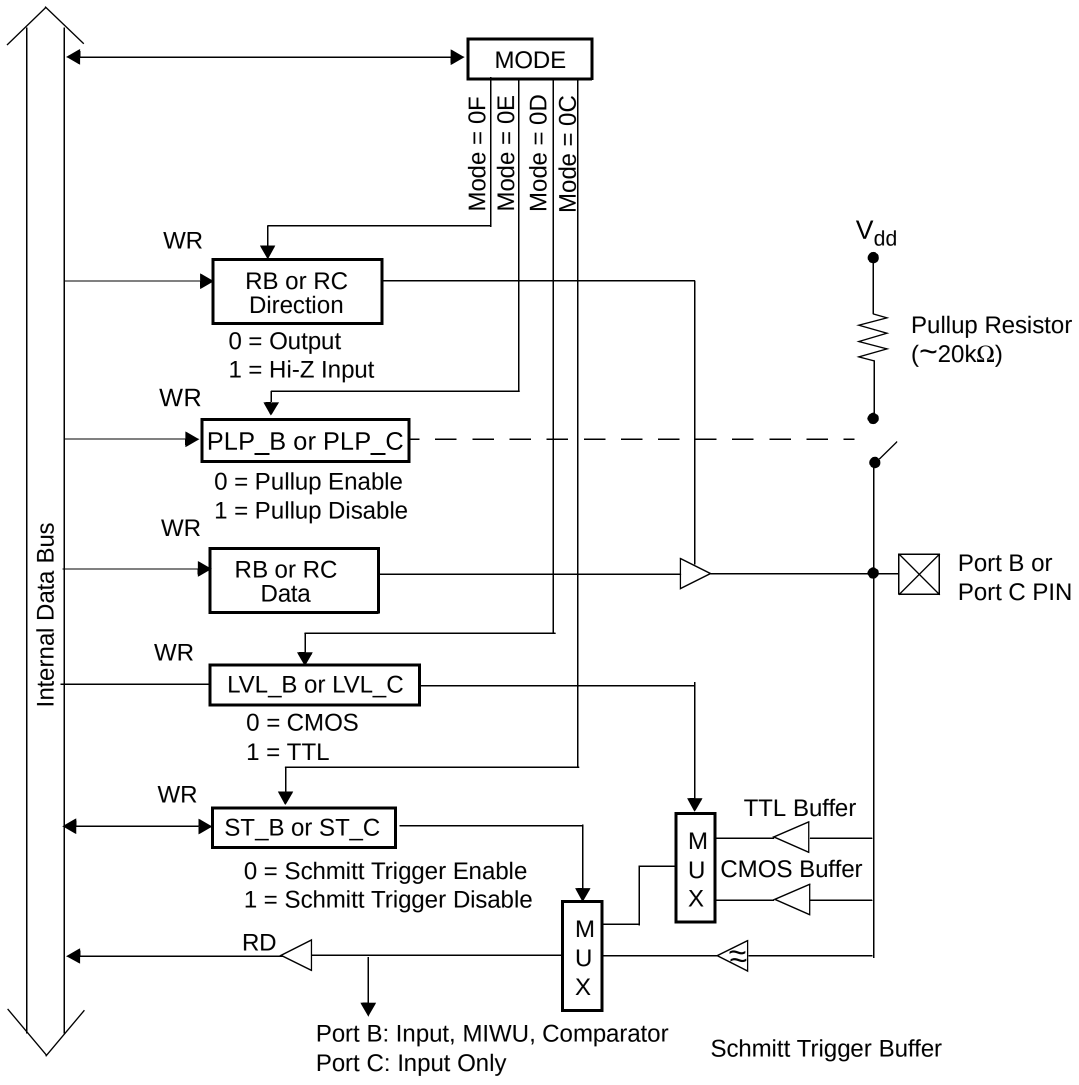


Figure 3-2. Port B, Port C Configuration

For example, suppose all four Port A pins are configured as outputs and with RA0 and RA1 to be high, and RA2 and RA3 to be low:

```
mov   W,#$03    ;load W with the value 03h
                ;(bits 0 and 1 high)
mov   $05,W     ;write 03h to Port A data
                ;register
```

The second “mov” instruction in this example writes the Port A data register (RA), which controls the output levels of the four Port A pins, RA0 through RA3. Because Port A has only four I/O pins, only the four least significant bits of this register are used. The four high-order register bits are “don’t care” bits. Port B and Port C are both eight bits wide, so the full widths of the RB and RC registers are used.

When a write is performed to a bit position for a port that has been configured as an input, a write to the port data register is still performed, but it has no immediate effect on the pin. If later that pin is configured to operate as an output, it will reflect the value that has been written to the data register.

When a read is performed from a bit position for a port, the operation is actually reading the voltage level on the pin itself, not necessarily the bit value stored in the port data register. This is true whether the pin is configured to operate as an input or an output. Therefore, with the pin configured to operate as an input, the data register contents have no effect on the value that you read. With the pin configured to operate as an output, what is read generally matches what has been written to the register.

4.0 SPECIAL-FUNCTION REGISTERS

The CPU uses a set of special-function registers to control the operation of the device.

The CPU registers include an 8-bit working register (W), which serves as a pseudo accumulator. It holds the second operand of an instruction, receives the literal in immediate type instructions, and also can be program-selected as the destination register.

A set of 31 file registers serves as the primary accumulator. One of these registers holds the first operand of an instruction and another can be program-selected as the destination register. The first eight file registers include the Real-Time Clock/Counter register (RTCC), the lower eight bits of the 11-bit Program Counter (PC), the 8-bit STATUS register, three port control registers for Port A, Port B, Port C, the 8-bit File Select Register (FSR), and INDF used for indirect addressing.

The five low-order bits of the FSR register select one of the 31 file registers in the indirect addressing mode. Calling for the file register located at address 00h (INDF) in any of the file-oriented instructions selects indirect addressing, which uses the FSR register. It should be noted that the file register at address 00h is not a physically implemented register. The CPU also contains an 8-level, 11-bit hardware push/pop stack for subroutine linkage.

Table 4-1. Special-Function Registers

Addr	Name	Function
00h	INDF	Used for indirect addressing
01h	RTCC	Real Time Clock/Counter
02h	PC	Program Counter (low byte)
03h	STATUS	Holds Status bits of ALU
04h	FSR	File Select Register
05h	RA	Port RA Control register
06h	RB	Port RB Control register
07h	RC*	Port RC Control register

*In the SX18 package, Port C is not used, and address 07h is available as a general-purpose RAM location.

4.1 PC Register (02h)

The PC register holds the lower eight bits of the program counter. It is accessible at run time to perform branch operations.

4.2 STATUS Register (03h)

The STATUS register holds the arithmetic status of the ALU, the page select bits, and the reset state. The STATUS register is accessible during run time, except that bits PD and TO are read-only. It is recommended that only SETB and CLRB instructions be used on this register. Care should be exercised when writing to the STATUS register as the ALU status bits are updated upon completion of the write operation, possibly leaving the STATUS register with a result that is different than intended.

PA2	PA1	PA0	TO	PD	Z	DC	C
Bit 7							**Bit 0**

Bit 7-5: Page select bits PA2:PA0

000 = Page 0 (000h – 01FFh)

001 = Page 1 (200h – 03FFh)

010 = Page 2 (400h – 05FFh)

011 = Page 3 (600h – 07FFh)

Bit 4: Time Out bit, TO

1 = Set to 1 after power up and upon execution of CLRWDT or SLEEP instructions

0 = A watchdog time-out occurred

Bit 3: Power Down bit, PD

1= Set to a 1 after power up and upon execution of the CLRWDT instruction

0 = Cleared to a ‘0’ upon execution of SLEEP instruction

Bit 2: Zero bit, Z

1 = Result of math operation is zero

0 = Result of math operation is non-zero

Bit 1: Digit Carry bit, DC

After Addition:

1 = A carry from bit 3 occurred

0 = No carry from bit 3 occurred

After Subtraction:

1 = No borrow from bit 3 occurred

0 = A borrow from bit 3 occurred

Bit 0: Carry bit, C

After Addition:

1 = A carry from bit 7 of the result occurred

0 = No carry from bit 7 of the result occurred

After Subtraction:

1 = No borrow from bit 7 of the result occurred

0 = A borrow from bit 7 of the result occurred

Rotate (RR or RL) Instructions:

The carry bit is loaded with the low or high order bit, respectively. When $\overline{CF}$ bit is cleared, Carry bit works as input for ADD and SUB instructions.

9.0 OSCILLATOR CIRCUITS

The device supports several user-selectable oscillator modes. The oscillator modes are selected by programming the appropriate values into the FUSE Word register. These are the different oscillator modes offered:

LP:	Low Power Crystal
XT:	Crystal/Resonator
HS:	High Speed Crystal/Resonator
RC:	External Resistor/Capacitor
	Internal Resistor/Capacitor

9.1 XT, LP or HS modes

In XT, LP or HS, modes, you can use either an external resonator network or an external clock signal as the device clock.

To use an external resonator network, you connect a crystal or ceramic resonator to the OSC1/CLKIN and OSC2/CLKOUT pins according to the circuit configuration shown in Figure 9-1. A parallel resonant crystal type is recommended. Use of a series resonant crystal may result in a frequency that is outside the crystal manufacturer specifications. Table 9-1 shows the recommended external components associated with a crystal-based oscillator. Table 9-2 shows the recommended external component values for a resonator-based oscillator.

Bits 0, 1 and 5 of the FUSE register (FOSC1:FOSC2) are used to configure the different external resonator/crystal oscillator modes. These bits allow the selection of the appropriate gain setting for the internal driver to match the desired operating frequency. If the XT, LP, or HS mode is selected, the OSC1/CLKIN pin can be driven by an external clock source rather than a resonator network, as long as the clock signal meets the specified duty cycle, rise and fall times, and input levels (Figure 9-2). In this case, the OSC2/CLKOUT pin should be left open.

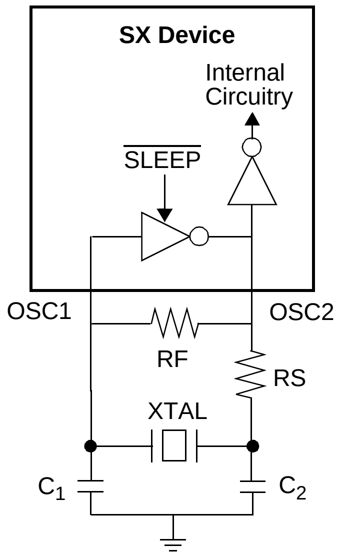


Figure 9-1. Crystal Operation (or Ceramic Resonator) (HS, XT or LP OSC Configuration)

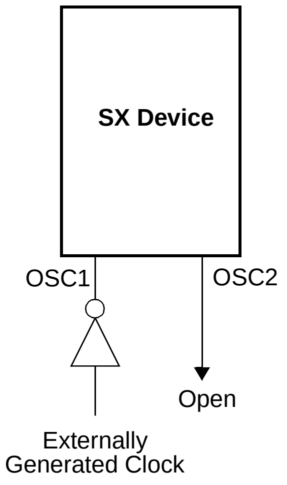


Figure 9-2. External Clock Input Operation (HS, XT or LP OSC Configuration)

Table 9-1. External Component Selection for Crystal Oscillator (Vdd=5.0V), Rs = 0 Ω

FOSC2:FOSC0	Crystal Frequency	C1	C2	R_F
010	4 MHz	15 pF	22 pF	1 MΩ
011	8 MHz	56 pF	33 pF	1 MΩ
011	20 MHz	33 pF	22 pF	1 MΩ
011	32 MHz	15 pF	22 pF	1 MΩ
100	50* MHz	15 pF	15 pF	1 MΩ

* 50 MHz fundamental crystal

Table 9-2. External Component Selection for Murata Ceramic Resonators (Vdd=5.0V)

FOSC2:FOSC0	Resonator Frequency	Resonator Part Number	C1	C2	R_F	R_S
011	4 MHz	CSA4.00MG	30 pF	30 pF	1MΩ	0 Ω
011	4 MHz	CST4.00MGW	Internal (30 pF)	Internal (30 pF)	1 MΩ	0 Ω
011	4 MHz	CSTCC4.00G0H6	Internal (47 pF)	Internal (47 pF)	1 MΩ	0 Ω
011	8 MHz	CSA8.00MTZ	30 pF	30 pF	1 MΩ	0 Ω
011	8 MHz	CST8.00MTW	Internal (30 pF)	Internal 30 pF)	1 MΩ	0 Ω
011	8 MHz	CSTCC8.00MG0H6	Internal (47 pF)	Internal 47pF)	1 MΩ	0 Ω
011	20 MHz	CSA20.00MXZ040	5 pF	5 pF	1 MΩ	0 Ω
011	20 MHz	CST20.00MXW0H1	Internal (5 pF)	Internal (5 pF)	1 MΩ	0 Ω
011	20 MHz	CSACV20.00MXJ040	5 pF	5 pF	22 kΩ	0 Ω
011	20 MHz	CSTCV20.00MXJ0H1	Internal (5 pF)	Internal (5 pF)	22 kΩ	0 Ω
100	33 MHz	CSA33.00MXJ040	5 pF	5 pF	1 MΩ	0 Ω
100	33 MHz	CST33.00MXW040	Internal (5 pF)	Internal (5 pF)	1 MΩ	0 Ω
100	33 MHz	CSACV33.00MXJ040	5 pF	5 pF	1 MΩ	0 Ω
100	33 MHz	CSTCV33.00MXJ040	Internal (5 pF)	Internal (5 pF)	1 MΩ	0 Ω
101	50 MHz	CSA50.00MXZ040	15 pF	15 pF	10 kΩ	0 Ω
101	50 MHz	CST50.00MXW0H3	Internal (15 pF)	Internal (15 pF)	10 kΩ	0Ω
101	50 MHz	CSACV50.00MXJ040	15 pF	15 pF	10 kΩ	0 Ω
101	50 MHz	CSTCV50.00MXJ0H3	Internal (15 pF)	Internal (15 pF)	10 kΩ	0 Ω

9.2 External RC Mode

The external RC oscillator mode provides a cost-effective approach for applications that do not require a precise operating frequency. In this mode, the RC oscillator frequency is a function of the supply voltage, the resistor (R) and capacitor (C) values, and the operating temperature. In addition, the oscillator frequency will vary from unit to unit due to normal manufacturing process variations. Furthermore, the difference in lead frame capacitance between package types also affects the oscillation frequency, especially for low C values. The external R and C component tolerances contribute to oscillator frequency variation as well.

Figure 9-3 shows the external RC connection diagram. The recommended R value is from 3kΩ to 100kΩ. For R values below 2.2kΩ, the oscillator may become unstable, or may stop completely. For very high R values (such as 1 MΩ), the oscillator becomes sensitive to noise, humidity, and leakage.

Although the oscillator will operate with no external capacitor (C = 0pF), it is recommended that you use values above 20 pF for noise immunity and stability. With no or small external capacitance, the oscillation frequency can vary significantly due to variation in PCB trace or package lead frame capacitances.

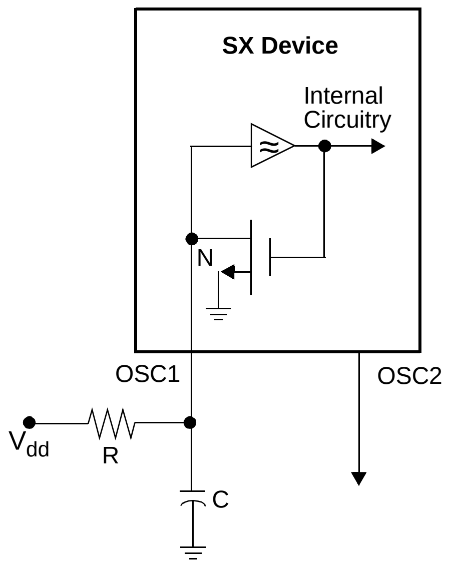


Figure 9-3. RC Oscillator Mode

9.3 Internal RC Mode

The internal RC mode uses an internal oscillator, so the device does not need any external components. At 4 MHz, the internal oscillator provides typically +/–8% accuracy over the allowed temperature range. The internal clock frequency can be divided down to provide one of eight lower-frequency choices by selecting the desired value in the FUSE Word register. The frequency range is from 31.25 KHz to 4 MHz. The default operating frequency of the internal RC oscillator may not be 4 MHz. This is due to the fact that the SX device requires trimming to obtain 4 MHz operation. The parts shipped out of the factory are not trimmed. The device relies on the programming tool provided by the third party vendors to support trimming.

10.0 REAL TIME CLOCK (RTCC)/WATCHDOG TIMER

The device contains an 8-bit Real Time Clock/Counter (RTCC) and an 8-bit Watchdog Timer (WDT). An 8-bit programmable prescaler extends the RTCC to 16 bits. If the prescaler is not used for the RTCC, it can serve as a postscaler for the Watchdog Timer. Figure 10-1 shows the RTCC and WDT block diagram.

10.1 RTCC

RTCC is an 8-bit real-time timer that is incremented once each instruction cycle or from a transition on the RTCC pin. The on-board prescaler can be used to extend the RTCC counter to 16 bits.

The RTCC counter can be clocked by the internal instruction cycle clock or by an external clock source presented at the RTCC pin.

To select the internal clock source, bit 5 of the OPTION register should be cleared. In this mode, RTCC is incremented at each instruction cycle unless the prescaler is selected to increment the counter.

To select the external clock source, bit 5 of the OPTION register must be set. In this mode, the RTCC counter is incremented with each valid signal transition at the RTTC pin. By using bit 4 of the OPTION register, the transition can be programmed to be either a falling edge or rising edge. Setting the control bit selects the falling edge to increment the counter. Clearing the bit selects the rising edge.

The RTCC generates an interrupt as a result of an RTCC rollover from 0FF to 000. There is no interrupt pending bit to indicate the overflow occurrence. The RTCC register must be sampled by the program to determine any overflow occurrence.

11.0 COMPARATOR

The device contains an on-chip differential comparator. Ports RB0-RB2 support the comparator. Ports RB1 and RB2 are the comparator negative and positive inputs, respectively, while Port RB0 serves as the comparator output pin. To use these pins in conjunction with the comparator, the user program must configure Ports RB1 and RB2 as inputs and Port RB0 as an output. The CMP_B register is used to enable the comparator, to read the output of the comparator internally, and to enable the output of the comparator to the comparator output pin.

The comparator enable bits are set to “1” upon reset, thus disabling the comparator. To avoid drawing additional current during the power down mode, the comparator should be disabled before entering the power down mode. Here is an example of how to setup the comparator and read the CMP_B register.

```
mov  M,#$08   ;set MODE register to access
              ;CMP_B
mov  W,#$00   ;clear W
mov  !RB,W    ;enable comparator and its
              ;output
...           ;delay after enabling
              ;comparator for response
mov  M,#$08   ;set MODE register to access
              ;CMP_B
mov  W,#$00   ;clear W
mov  !RB,W    ;enable comparator and its
              ;output and also read CMP_B
              ;(exchange W and CMB_B)
and  W,#$01   ;set/clear Z bit based on
              ;comparator result
snb  $03.2    ;test Z bit in STATUS reg
              ;(0 => RB2<RB1)
jmp  rb2_hi   ;jump only if RB2>RB1
...
```

The final “mov” instruction in this example performs an exchange of data between the working register (W) and the CMP_B register. This exchange occurs only with Port B accesses. Otherwise, the “mov” instruction does not perform an exchange, but only moves data from the source to the destination.

Figure 11-1 shows the comparator block diagram.

CMP_B - Comparator Enable/Status Register

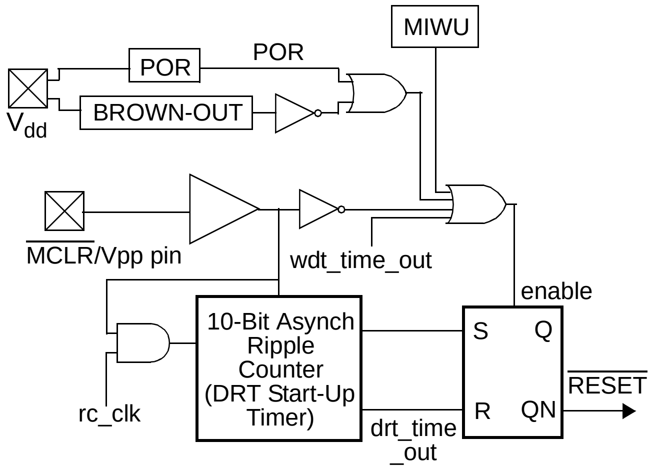
$\overline{\text{CMP\_EN}}$	$\overline{\text{CMP\_OE}}$	Reserved	CMP_RES
Bit 7	Bit 6	Bits 5–1	Bit 0

CMP_RES	Comparator result: 1 for RB2>RB1 or 0 for RB2<RB1. Comparator must be enabled (CMP_EN = 0) to read the result. The result can be read whether or not the CMP_OE bit is cleared.
$\overline{\text{CMP\_OE}}$	When cleared to 0, enables the comparator output to the RB0 pin.
$\overline{\text{CMP\_EN}}$	When cleared to 0, enables the comparator.

12.0 RESET

Power-On-Reset, Brown-Out reset, watchdog reset, or external reset initializes the device. Each one of these reset conditions causes the program counter to branch to the top of the program memory. For example, on the device with 2048K words of program memory, the program counter is initialized to 07FF.

The device incorporates an on-chip Power-On Reset (POR) circuit that generates an internal reset as V_{dd} rises during power-up. Figure 12-1 is a block diagram of the circuit. The circuit contains an 10-bit Delay Reset Timer (DRT) and a reset latch. The DRT controls the reset time-out delay. The reset latch controls the internal reset signal. Upon power-up, the reset latch is set (device held in reset), and the DRT starts counting once it detects a valid logic high signal at the $\overline{\text{MCLR}}$ pin. Once DRT reaches the end of the timeout period (typically 72 msec), the reset latch is cleared, releasing the device from reset state.


MIWU
POR
POR
BROWN-OUT
V_{dd}
$\overline{\text{MCLR}}$/Vpp pin
wdt_time_out
enable
10-Bit Asynch Ripple Counter (DRT Start-Up Timer)
S
Q
R
QN
$\overline{\text{RESET}}$
rc_clk
drt_time_out

Note:Ripple counter is 10 bits for Power on Reset (POR) only.

Figure 12-1. Block Diagram of On-Chip Reset Circuit

Figure 12-2 shows a power-up sequence where $\overline{\text{MCLR}}$ is not tied to the V_{dd} pin and V_{dd} signal is allowed to rise and stabilize before MCLR pin is brought high. The device will actually come out of reset T_{drt} msec after MCLR goes high.

The brown-out circuitry resets the chip when device power (V_{dd}) dips below its minimum allowed value, but not to zero, and then recovers to the normal value.

.

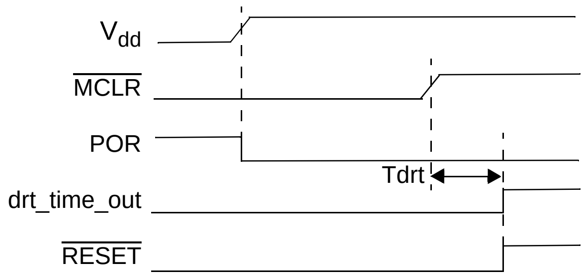


Figure 12-2. Time-Out Sequence on Power-Up (MCLR not tied to V_{dd})

15.0 INSTRUCTION SET

As mentioned earlier, the SX family of devices uses a modified Harvard architecture with memory-mapped input/output. The device also has a RISC type architecture in that there are 43 single-word basic instructions. The instruction set contains byte-oriented file register, bit-oriented file register, and literal/control instructions.

Working register W is one of the CPU registers, which serves as a pseudo accumulator. It is a pseudo accumulator in a sense that it holds the second operand, receives the literal in the immediate type instructions, and also can be program-selected as the destination register. The bank of 31 file registers can also serve as the primary accumulators, but they represent the first operand and may be program-selected as the destination registers.

15.1 Instruction Set Features

1. All single-word (12-bit) instructions for compact code efficiency.
2. All instructions are single cycle except the jump type instructions (JMP, CALL) and failed test instructions (DECSZ fr, INCSZ fr, SB bit, SNB bit), which are two-cycle.
3. A set of File registers can be addressed directly or indirectly, and serve as accumulators to provide first operand; W register provides the second operand.
4. Many instructions include a destination bit which selects either the register file or the accumulator as the destination for the result.
5. Bit manipulation instructions (Set, Clear, Test and Skip if Set, Test and Skip if Clear).
6. STATUS Word register memory-mapped as a register file, allowing testing of status bits (carry, digit carry, zero, power down, and timeout).
7. Program Counter (PC) memory-mapped as register file allows W to be used as offset register for indirect addressing of program memory.
8. Indirect addressing data pointer FSR (file select register) memory-mapped as a register file.
9. IREAD instruction allows reading the instruction from the program memory addressed by W and upper four bits of MODE register.
10. Eight-level, 11-bit push/pop hardware stack for subroutine linkage using the Call and Return instructions.
11. Six addressing modes provide great flexibility.

15.2 Instruction Execution

An instruction goes through a four-stage pipeline to be executed (Figure 15-1). The first instruction is fetched from the program memory on the first clock cycle. On the second clock cycle, the first instruction is decoded and the second instruction is fetched. On the third clock cycle, the first instruction is executed, the second instruction is decoded, and the third instruction is fetched. On the fourth clock cycle, the first instruction's results are written to its destination, the second instruction is executed, the third instruction is decoded, and the fourth instruction is fetched. Once the pipeline is full, instructions are executed at the rate of one per clock cycle.

Instructions that directly affect the contents of the program counter (such as jumps and calls) require that the pipeline be cleared and subsequently refilled. Therefore, these instruction take more than one clock cycle.

The instruction execution time is derived by dividing the oscillator frequency by either one (turbo mode) or four (non-turbo mode). The divide-by factor is selected through the FUSE Word register.

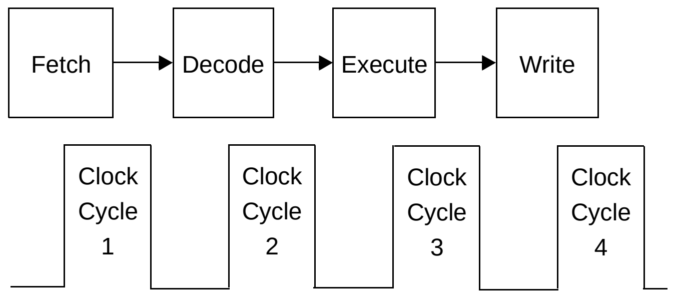


Figure 15-1. Pipeline and Clock Scheme

15.3 Addressing Modes

The device support the following addressing modes:

Data Direct
Data Indirect
Immediate
Program Direct
Program Indirect
Relative

Both direct and indirect addressing modes are available. The INDF register, though physically not implemented, is used in conjunction with the indirect data pointer (FSR) to perform indirect addressing. An instruction using INDF as its operand field actually performs the operation on the register pointed by the contents of the FSR. Consequently, processing two multiple-byte operands requires alternate loading of the operand addresses into the FSR pointer as the multiple byte data fields are processed.

Examples:

Direct addressing:

```
mov   RA,#01            ;move “1” to RA
```

Indirect Addressing:

```
mov   FSR,#RA           ;FSR = address of RA
mov   INDF,#$01         ;move “1” to RA
```

15.13 Comparison and Conditional Branch Instructions

The instruction set includes instructions such as DECSZ fr (decrement file register and skip if zero), INCSZ fr (increment file register and skip if zero), SNB bit (bit test file register and skip if bit clear), and SB bit (bit test file register and skip if bit set). These instructions will cause the next instruction to be skipped if the tested condition is true. If a skip instruction is immediately followed by a PAGE or BANK instruction (and the tested condition is true) then two instructions are skipped and the operation consumes three cycles. This is useful for conditional branching to another page where a PAGE instruction precedes a JMP. If several PAGE and BANK instructions immediately follow a skip instruction then they are all skipped plus the next instruction and a cycle is consumed for each.

15.14 Logical Instruction

The instruction set contain a full complement of the logical instructions (AND, OR, Exclusive OR), with the W register and a selected memory location (using either direct or indirect addressing) serving as the two operands.

15.15 Shift and Rotate Instructions

The instruction set includes instructions for left or right rotate-through-carry.

15.16 Complement and SWAP

The device can perform one's complement operation on the file register (fr) and W register. The MOV W,<>fr instruction performs nibble-swap on the fr and puts the value into the W register.

15.17 Key to Abbreviations and Symbols

Symbol	Description
W	Working register
fr	File register (memory-mapped register in the range of 00h to FFh)
PC	Lower eight bits of program counter (file register 02h)
STATUS	STATUS register (file register 03h)
FSR	File Select Register (file register 04h)
C	Carry bit in STATUS register (bit 0)
DC	Digit Carry bit in STATUS register (bit 1)
Z	Zero bit in STATUS register (bit 2
PD	Power Down bit in STATUS register (bit 3)
TO	Watchdog Timeout bit in STATUS register (bit 4)
PA2:PA0	Page select bits in STATUS register (bits 7:5)
OPTION	OPTION register (not memory-mapped)
WDT	Watchdog Timer register (not memory-mapped)
MODE	MODE register (not memory-mapped)
rx	Port control register pointer (RA, RB, or RC)
!	Non-memory-mapped register designator
f	File register address bit in opcode
k	Constant value bit in opcode
n	Numerical value bit in opcode
b	Bit position selector bit in opcode
.	File register / bit selector separator in assembly language instruction
#	Immediate literal designator in assembly language instruction
lit	Literal value in assembly language instruction
addr8	8-bit address in assembly language instruction
addr9	9-bit address in assembly language instruction
addr12	12-bit address in assembly language instruction
/	Logical 1's complement
|	Logical OR
^	Logical exclusive OR
&	Logical AND
<>	Swap high and low nibbles (4-bit segments)
<<	Rotate left through carry bit
>>	Rotate right through carry bit
- -	Decrement file register
++	Increment file register

16.0 INSTRUCTION SET SUMMARY TABLE

Table 16-1 lists all of the instructions, organized by category. For each instruction, the table shows the instruction mnemonic (as written in assembly language), a brief description of what the instruction does, the number of instruction cycles required for execution, the binary opcode, and the status bits affected by the instruction.

The “Cycles” column typically shows a value of 1, which means that the overall throughput for the instruction is one per clock cycle. In some cases, the exact number of cycles depends on the outcome of the instruction (such as the test-and-skip instructions) or the clocking mode (Compatible or Turbo). In those cases, all possible numbers of cycles are shown in the table.

The instruction execution time is derived by dividing the oscillator frequency by either one (Turbo mode) or four (Compatible mode). The divide-by factor is selected through the FUSE Word register.

Table 16-1. The SX Instruction Set

Mnemonic, Operands	Description	Cycles (Compatible)	Cycles (Turbo)	Opcode	Bits Affected
Logical Operations					
AND fr, W	AND of fr and W into fr (fr = fr & W)	1	1	`0001 011f ffff`	Z
AND W, fr	AND of W and fr into W (W = W & fr)	1	1	`0001 010f ffff`	Z
AND W,#lit	AND of W and Literal into W (W = W & lit)	1	1	`1110 kkkk kkkk`	Z
NOT fr	Complement of fr into fr (fr = fr ^ FFh)	1	1	`0010 011f ffff`	Z
OR fr,W	OR of fr and W into fr (fr = fr | W)	1	1	`0001 001f ffff`	Z
OR W,fr	OR of W and fr into fr (W = W | fr)	1	1	`0001 000f ffff`	Z
OR W,#lit	OR of W and Literal into W (W = W | lit)	1	1	`1101 kkkk kkkk`	Z
XOR fr,W	XOR of fr and W into fr (fr = fr ^ W)	1	1	`0001 101f ffff`	Z
XOR W,fr	XOR of W and fr into W (W = W ^ fr)	1	1	`0001 100f ffff`	Z
XOR W,#lit	XOR of W and Literal into W (W = W ^ lit)	1	1	`1111 kkkk kkkk`	Z
Arithmetic and Shift Operations					
ADD fr,W	Add W to fr (fr = fr + W); carry bit is added if $\overline{\text{CF}}$ bit in FUSEX register is cleared to 0	1	1	`0001 111f ffff`	C, DC, Z
ADD W,fr	Add fr to W (W = W + fr); carry bit is added if $\overline{\text{CF}}$ bit in FUSEX register is cleared to 0	1	1	`0001 110f ffff`	C, DC, Z
CLR fr	Clear fr (fr = 0)	1	1	`0000 011f ffff`	Z
CLR W	Clear W (W = 0)	1	1	`0000 0100 0000`	Z
CLR !WDT	Clear Watchdog Timer, clear prescaler if assigned to the Watchdog (TO = 1, PD = 1)	1	1	`0000 0000 0100`	TO, PD
DEC fr	Decrement fr (fr = fr - 1)	1	1	`0000 111f ffff`	Z
DECSZ fr	Decrement fr and Skip if Zero (fr = fr - 1 and skip next instruction if result is zero)	1 or 2 (skip)	1 or 2 (skip)	`0010 111f ffff`	none
INC fr	Increment fr (fr = fr + 1)	1	1	`0010 101f ffff`	Z
INCSZ fr	Increment fr and Skip if Zero (fr = fr + 1 and skip next instruction if result is zero)	1 or 2 (skip)	1 or 2 (skip)	`0011 111f ffff`	none
RL fr	Rotate fr Left through Carry (fr = << fr)	1	1	`0011 011f ffff`	C
RR fr	Rotate fr Right through Carry (fr = >> fr)	1	1	`0011 001f ffff`	C
SUB fr,W	Subtract W from fr (fr = fr - W); complement of the carry bit is subtracted if $\overline{\text{CF}}$ bit in FUSEX register is cleared to 0	1	1	`0000 101f ffff`	C, DC, Z
SWAP fr	Swap High/Low Nibbles of fr (fr = <> fr)	1	1	`0011 101f ffff`	none

Table 16-1. The SX Instruction Set (Continued)

Mnemonic, Operands	Description	Cycles (Compatible)	Cycles (Turbo)	Opcode	Bits Affected
Program Control instruction					
CALL addr8	Call Subroutine: top-of-stack = program counter + 1 PC(7:0) = addr8 program counter (8) = 0 program counter (10:9) = PA1:PA0	2	3	`1001 kkkk kkkk`	none
JMP addr9	Jump to Address: PC(7:0) = addr9(7:0) program counter (8) = addr9(8) program counter (10:9) = PA1:PA0	2	3	`101k kkkk kkkk`	none
NOP	No Operation	1	1	`0000 0000 0000`	none
RET	Return from Subroutine (program counter = top-of-stack)	2	3	`0000 0000 1100`	none
RETP	Return from Subroutine Across Page Boundary (PA1:PA0 = top-of-stack (10:9) and program counter = top-of-stack)	2	3	`0000 0000 1101`	PA1, PA0
RETI	Return from Interrupt (restore W, STATUS, FSR, and program counter from shadow registers)	2	3	`0000 0000 1110`	all STATUS, except TO, PD
RETIW	Return from Interrupt and add W to RTCC (restore W, STATUS, FSR, and program counter from shadow registers; and add W to RTCC)	2	3	`0000 0000 1111`	all STATUS, except TO, PD
RETW lit	Return from Subroutine with Literal in W (W = lit and program counter = top-of-stack)	2	3	`1000 kkkk kkkk`	none
System Control Instructions					
BANK addr8	Load Bank Number into FSR(7:5) FSR(7:5) = addr8(7:5)	1	1	`0000 0001 1nnn`	none
IREAD	Read Word from Instruction Memory MODE:W = data at (MODE:W)	1	4	`0000 0100 0001`	none
PAGE addr12	Load Page Number into STATUS(7:5) STATUS(7:5) = addr12(11:9)	1	1	`0000 0001 0nnn`	PA1, PA0
SLEEP	Power Down Mode WDT = 00h, TO = 1, stop oscillator (PD = 0, clears prescaler if assigned)	1	1	`0000 0000 0011`	TO, PD

16.1 Equivalent Assembler Mnemonics

Some assemblers support additional instruction mnemonics that are special cases of existing instructions or alternative mnemonics for standard ones. For example, an assembler might support the mnemonic "CLC" (clear carry), which is interpreted the same as the instruction "clrb $03.0" (clear bit 0 in the STATUS register). Some of the commonly supported equivalent assembler mnemonics are described in Table 16-2.

Table 16-2. Equivalent Assembler Mnemonics

Syntax	Description	Equivalent	Cycles
CLC	Clear Carry bit	CLRB $03.0	1
CLZ	Clear Zero bit	CLRB $03.2	1
JMP W	Jump Indirect W	MOV $02,W	4 or 3 (note 1)
JMP PC+W	Jump Indirect W Relative	ADD $02,W	4 or 3 (note 1)
MODE imm4	Move Immediate to MODE Register	MOV M,#lit	1
NOT W	Complement W	XOR W,#$FF	1
SC	Skip if Carry bit Set	SB $03.0	1 or 2 (note 2)
SKIP	Skip Next Instruction	SNB $02.0 or SB $02.0	4 or 2 (note 3)

Note 1: The JMP W or JMP PC+W instruction takes 4 cycles in the "compatible" clocking mode or 3 cycles in the "turbo" clocking mode.

Note 2: The SC instruction takes 1 cycle if the tested condition is false or 2 cycles if the tested condition is true.

Note 3: The assembler converts the SKIP instruction into a SNB or SB instruction that tests the least significant bit of the program counter, choosing SNB or SB so that the tested condition is always true. The instruction takes 4 cycles in the "compatible" clocking mode or 2 cycles in the "turbo" clocking mode.

17.0 ELECTRICAL CHARACTERISTICS

17.1 Absolute Maximum Ratings

Ambient temperature under bias	-40°C to +85°C
Storage temperature	-65°C to +150°C
Voltage on V_{dd} with respect to V_{ss}	0 V to +7.0V
Voltage on OSC1 with respect to V_{ss}	0 V to +13.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V_{ss}	0 V to +13.5V
Voltage on all other pins with respect to V_{ss}	-0.6 V to (V_{dd} + 0.6V)V
Total power dissipation	700 mW
Max. current out of V_{ss} pin	130 mA
Max. current into V_{dd} pin	130 mA
Max. DC current into an input pin (with internal protection diode forward biased)	$\pm$500 μA
Max. allowable sink current per I/O pin	45 mA
Max. allowable source current per I/O pin	45 mA

17.2 DC Characteristics

SX20/28AC at 75MHz(Temp Range: 0°C <= Ta <= +70°C)

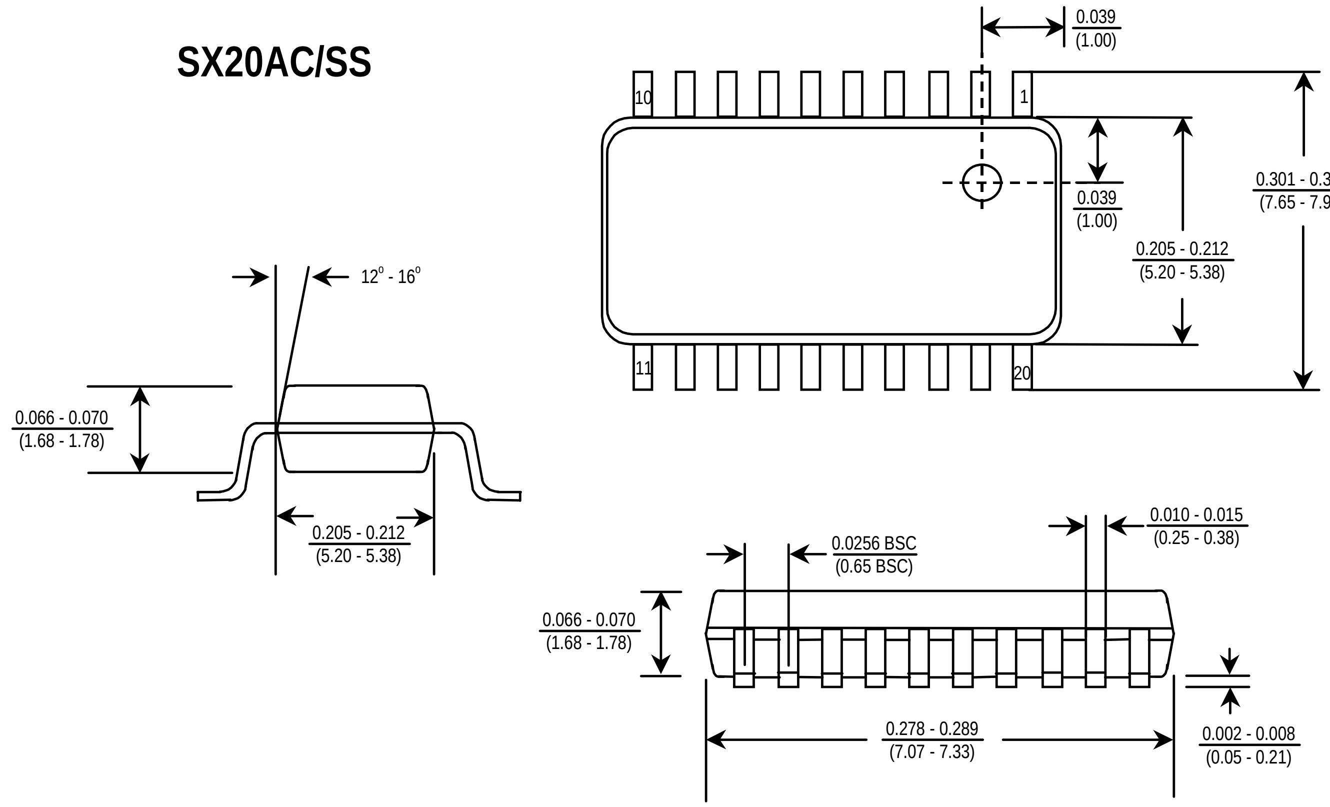
SX20/28AC at 50MHz (Temp Range: -40°C <= Ta <= +85°C)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{dd}	Supply Voltage (Note 1)	F_{osc}= 32 MHz F_{osc}= 50 MHz F_{osc}= 75 MHz	2.7 3.0 4.5	- - -	5.5 5.5 5.5	V V V
S_{Vdd}	V_{dd} rise rate (Note 1)		0.05	-	-	V/ms
I_{dd}	Supply Current, active	V_{dd} = 5.0V, F_{osc} = 75 MHz (External OSC) V_{dd} = 5.0V, F_{osc} = 50 MHz (Crystal) V_{dd} = 5.0V, F_{osc} = 4 MHz (Crystal) V_{dd} = 2.7V, F_{osc} = 20 MHz (Crystal)	- - - -	100 77 7.5 17	105 82 8 18	mA mA mA mA
I_{pd}	Supply Current, power down	V_{dd} = 3.0V, WDT enabled (before timeout) V_{dd} = 3.0V, WDT disabled V_{dd} = 4.5V, WDT enabled V_{dd} = 4.5V, WDT disabled	-	10 1.0	20 9.0 110 100	µA µA µA µA
V_{ih}, V_{il}	Input Levels $\overline{\text{MCLR}}$, OSC1, RTCC Logic High Logic Low All Other Inputs CMOS Logic High Logic Low TTL Logic High Logic Low		 $0.8V_{dd}$ V_{ss} $0.7V_{dd}$ V_{ss} 2.0 V_{ss}		 V_{dd} $0.2V_{dd}$ V_{dd} $0.3V_{dd}$ V_{dd} 0.8	 V V V V V V
I_{il}	Input Leakage Current	V_{in} = V_{dd} or V_{ss}	-1.0		+1.0	µA
I_{ip}	Weak Pullup Current	V_{dd} = 5.5V, V_{in} = 0V V_{dd} = 3.0V, V_{in} = 0V	100 25		190 50	µA µA
V_{oh}	Output High Voltage Ports B, C Port A	 Ioh = 20mA, Vdd = 4.5V Ioh = 12mA, Vdd = 3.0V Ioh = 30mA, Vdd = 4.5 Ioh = 20mA, Vdd = 3.0V	 Vdd-0.7 Vdd-0.7 Vdd-0.7 Vdd-0.7			 V V V V
V_{ol}	Output Low Voltage All Ports	Iol = 30mA, Vdd = 4.5V Iol = 20mA, Vdd = 3.0V			0.6 0.6	V V

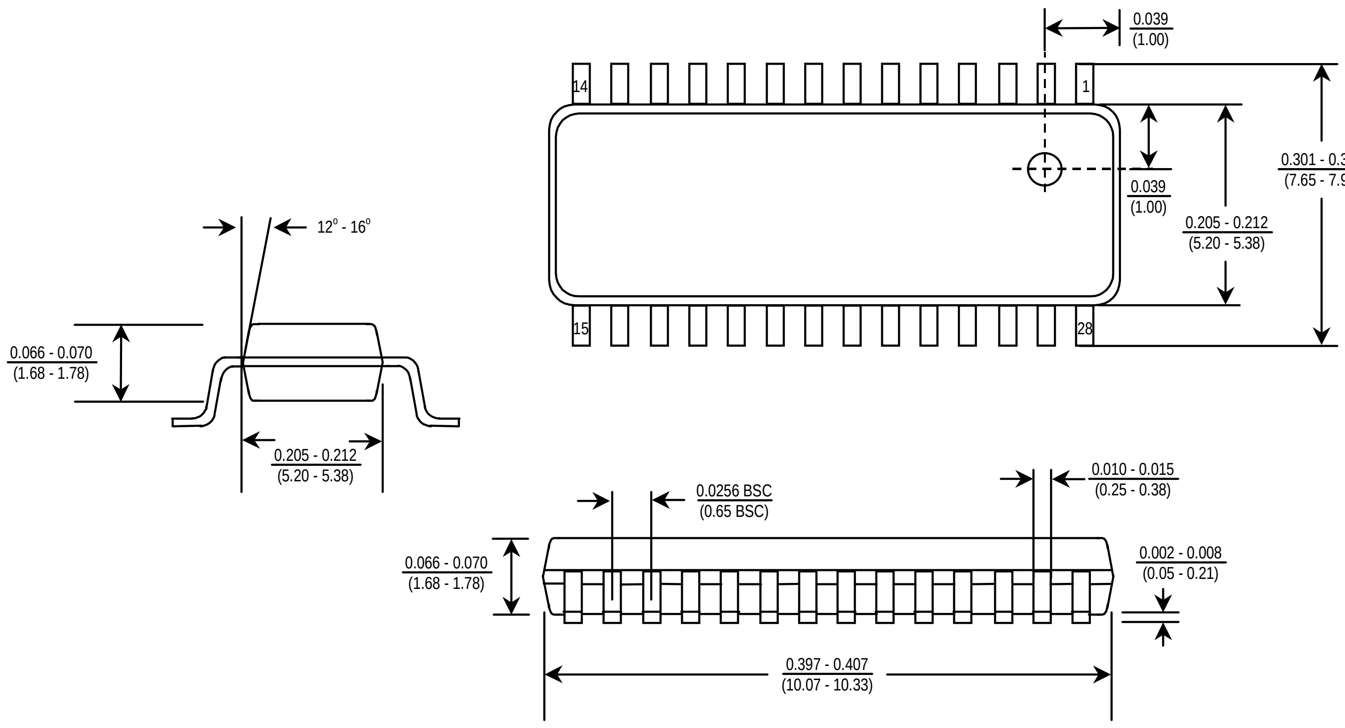
Note 1: Vdd must start rising from Vss to ensure proper Power-On-Reset when relying on the internal Power-On-Reset circuitry.

18.0 PACKAGE DIMENSIONS (DIMENSIONS ARE IN INCHES/(MILLIMETERS)

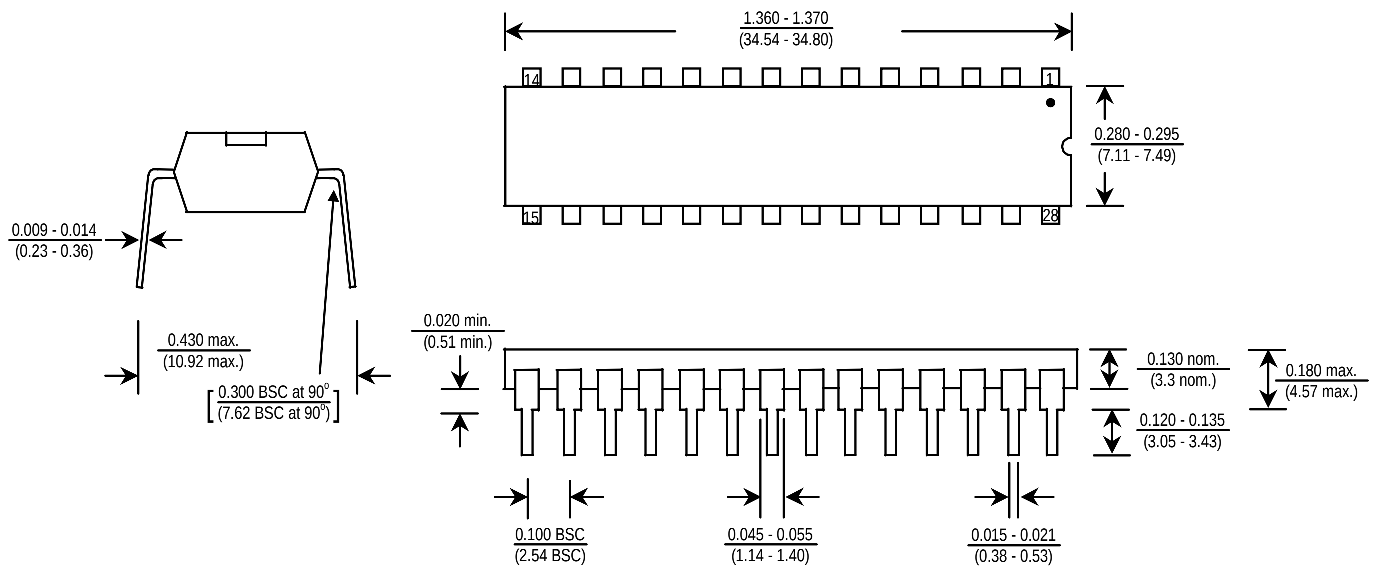
SX20AC/SS



SX28AC/SS



SX28AC/DP


1.360 - 1.370
(34.54 - 34.80)
14
1
15
28
0.280 - 0.295
(7.11 - 7.49)
0.009 - 0.014
(0.23 - 0.36)
0.430 max.
(10.92 max.)
0.300 BSC at 90°
(7.62 BSC at 90°)
0.020 min.
(0.51 min.)
0.130 nom.
(3.3 nom.)
0.180 max.
(4.57 max.)
0.120 - 0.135
(3.05 - 3.43)
0.100 BSC
(2.54 BSC)
0.045 - 0.055
(1.14 - 1.40)
0.015 - 0.021
(0.38 - 0.53)