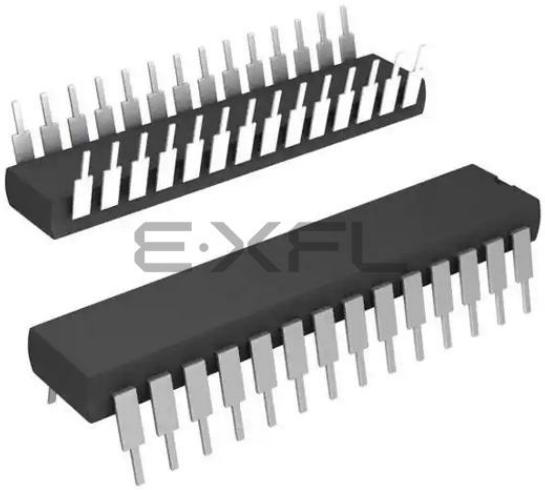


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
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-g

1.2 Key Features

50 MIPS Performance

- SX20AC/SX28AC: DC - 75 MHz
- SX20AC/SX28AC: 13.3 ns instruction cycle, 39.9 ns internal interrupt response
- 1 instruction per clock (branches 3)

EE/FLASH Program Memory and SRAM Data Memory

- Access time of < 13.3 ns provides single cycle access
- EE/Flash rated for > 10,000 rewrite cycles
- 2048 Words EE/Flash program memory
- 136x8 bits SRAM data memory

CPU Features

- Compact instruction set
- All instructions are single cycle except branch
- Eight-level push/pop hardware stack for subroutine linkage
- Fast table lookup capability through run-time readable code (IREAD instruction)
- Totally predictable program execution flow for hard real-time applications

Fast and Deterministic Interrupt

- Jitter-free 3-cycle internal interrupt response
- Hardware context save/restore of key resources such as PC, W, STATUS, and FSR within the 3-cycle interrupt response time
- External wakeup/interrupt capability on Port B (8 pins)

Flexible I/O

- All pins individually programmable as I/O
- Inputs are TTL or CMOS level selectable
- All pins have selectable internal pull-ups
- Selectable Schmitt Trigger inputs on Ports B, and C
- All outputs capable of sourcing/sinking 30 mA
- Port A outputs have symmetrical drive
- Analog comparator support on Port B (RB0 OUT, RB1 IN-, RB2 IN+)
- Selectable I/O operation synchronous to the oscillator clock

Hardware Peripheral Features

- One 8-bit Real Time Clock/Counter (RTCC) with programable 8-bit prescaler
- Watchdog Timer (shares the RTCC prescaler)
- Analog comparator
- Brown-out detector
- Multi-Input Wakeup logic on 8 pins
- Internal RC oscillator with configurable rate from 31.25 kHz to 4 MHz
- Power-On-Reset

Packages

- 20-pin SSOP, 28-pin DIP/SSOP

Programming and Debugging Support

- On- chip in-system programming support with serial and parallel interfaces
- In-system serial programming via oscillator pins
- On-chip in-System debugging support logic
- Real-time emulation, full program debug, and integrated development environment offered by third party tool vendors

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^2C™, 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.

2.3 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

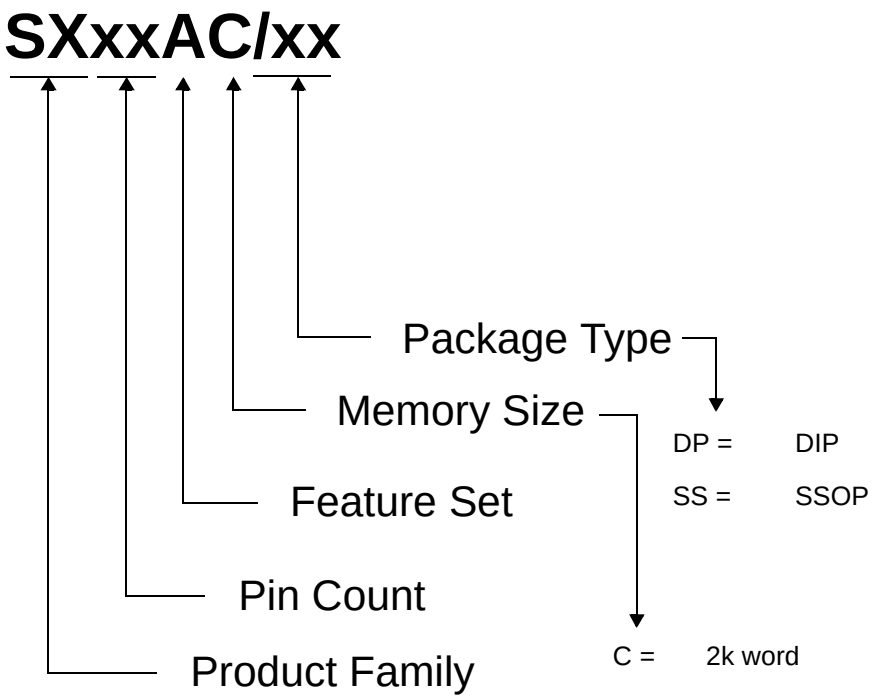


Figure 2-1. Part Number Reference Guide

3.1.1 Read-Modify-Write Considerations

Caution must be exercised when performing two successive read-modify-write instructions (SETB or CLRB operations) on I/O port pin. Input data used for an instruction must be valid *during* the time the instruction is executed, and the output result from an instruction is valid only *after* that instruction completes its operation. Unexpected results from successive read-modify-write operations on I/O pins can occur when the device is running at high speeds. Although the device has an internal write-back section to prevent such conditions, it is still recommended that the user program include a NOP instruction as a buffer between successive read-modify-write instructions performed on I/O pins of the same port.

Also note that reading an I/O port is actually reading the pins, not the output data latches. That is, if the pin output driver is enabled and driven high while the pin is held low externally, the port pin will read low.

3.2 Port Configuration

Each port pin offers the following configuration options:

- data direction
- input voltage levels (TTL or CMOS)
- pullup type (pullup resistor enable or disable)
- Schmitt trigger input (for Port B and Port C only)

Port B offers the additional option to use the port pins for the Multi-Input Wakeup/Interrupt function and/or the analog comparator function.

Port configuration is performed by writing to a set of control registers associated with the port. A special-purpose instruction is used to write these control registers:

- mov !RA,W (move W to Port A control register)
- mov !RB,W (move W to Port B control register)
- mov !RC,W (move W to Port C control register)

Each one of these instructions writes a port control register for Port A, Port B, or Port C. There are multiple control registers for each port. To specify which one you want to access, you use another register called the MODE register.

3.2.1 MODE Register

The MODE register controls access to the port configuration registers. Because the MODE register is not memory-mapped, it is accessed by the following special-purpose instructions:

- mov M, #lit (move literal to MODE register)
- mov M,W (move W to MODE register)
- mov W,M (move MODE register to W)

The value contained in the MODE register determines which port control register is accessed by the “mov !rx,W” instruction as indicated in Table 3-3. MODE register values not listed in the table are reserved for future expansion and should not be used. Therefore, the MODE register should always contain a value from 08h to 0Fh. Upon reset, the MODE register is initialized to 0Fh, which enables access to the port direction registers.

After a value is written to the MODE register, that setting remains in effect until it is changed by writing to the MODE register again. For example, you can write the value 0Eh to the MODE register just once, and then write to each of the three pullup configuration registers using the three “mov !rx,W” instructions.

Table 3-3. MODE Register and Port Control Register Access

MODE Reg.	mov !RA,W	mov !RB,W	mov !RC,W
08h	not used	CMP_B	not used
09h	not used	WKPND_B	not used
0Ah	not used	WKED_B	not used
0Bh	not used	WKEN_B	not used
0Ch	not used	ST_B	ST_C
0Dh	LVL_A	LVL_B	LVL_C
0Eh	PLP_A	PLP_B	PLP_C
0Fh	RA Direction	RB Direction	RC Direction

The following code example shows how to program the pullup control registers.

```
mov  M,#$0E   ;MODE=0Eh to access port pullup
              ;registers

mov  W,#$03   ;W = 0000 0011
mov  !RA,W    ;disable pullups for A0 and A1

mov  W,#$FF   ;W = 1111 1111
mov  !RB,W    ;disable all pullups for B0-B7

mov  W,#$00   ;W = 0000 0000
mov  !RC,W    ;enable all pullups for C0-C7
```

First the MODE register is loaded with 0Eh to select access to the pullup control registers (PLP_A, PLP_B, and PLP_C). Then the MOV !rx,W instructions are used to specify which port pins are to be connected to the internal pullup resistors. Setting a bit to 1 disconnects the corresponding pullup resistor, and clearing a bit to 0 connects the corresponding pullup resistor.

3.2.2 Port Configuration Registers

The port configuration registers that you control with the MOV !rx,W instruction operate as described below.

RA, RB, and RC Data Direction Registers (MODE=0Fh)

Each register bit sets the data direction for one port pin. Set the bit to 1 to make the pin operate as a high-impedance input. Clear the bit to 0 to make the pin operate as an output.

PLP_A, PLP_B, and PLP_C: Pullup Enable Registers (MODE=0Eh)

Each register bit determines whether an internal pullup resistor is connected to the pin. Set the bit to 1 to disconnect the pullup resistor or clear the bit to 0 to connect the pullup resistor.

LVL_A, LVL_B, and LVL_C: Input Level Registers (MODE=0Dh)

Each register bit determines the voltage levels sensed on the input port, either TTL or CMOS, when the Schmitt trigger option is disabled. Program each bit according to the type of device that is driving the port input pin. Set the bit to 1 for TTL or clear the bit to 0 for CMOS.

ST_B and ST_C: Schmitt Trigger Enable Registers (MODE=0Ch)

Each register bit determines whether the port input pin operates with a Schmitt trigger. Set the bit to 1 to disable Schmitt trigger operation and sense either TTL or CMOS voltage levels; or clear the bit to 0 to enable Schmitt trigger operation.

WKEN_B: Wakeup Enable Register (MODE=0Bh)

Each register bit enables or disables the Multi-Input Wakeup/Interrupt (MIWU) function for the corresponding Port B input pin. Clear the bit to 0 to enable MIWU operation or set the bit to 1 to disable MIWU operation. For more information on using the Multi-Input Wakeup/Interrupt function, see Section 7.0.

WKED_B: Wakeup Edge Register (MODE=0Ah)

Each register bit selects the edge sensitivity of the Port B input pin for MIWU operation. Clear the bit to 0 to sense rising (low-to-high) edges. Set the bit to 1 to sense falling (high-to-low) edges.

WKPND_B: Wakeup Pending Bit Register (MODE=09h)

When you access the WKPND_B register using MOV !RB,W, the CPU does an exchange between the contents of W and WKPND_B. This feature lets you read the WKPND_B register contents. Each bit indicates the status of the corresponding MIWU pin. A bit set to 1 indicates that a valid edge has occurred on the corresponding MIWU pin, triggering a wakeup or interrupt. A bit set to 0 indicates that no valid edge has occurred on the MIWU pin.

CMP_B: Comparator Register (MODE=08h)

When you access the CMP_B register using MOV !RB,W, the CPU does an exchange between the contents of W and CMP_B. This feature lets you read the CMP_B register contents. Clear bit 7 to enable operation of the comparator. Clear bit 6 to place the comparator result on the RB0 pin. Bit 0 is a result bit that is set to 1 when the voltage on RB2 is greater than RB1, or cleared to 0 otherwise. (For more information using the comparator, see Section 11.0.)

3.2.3 Port Configuration Upon Reset

Upon reset, all the port control registers are initialized to FFh. Thus, each pin is configured to operate as a high-impedance input that senses TTL voltage levels, with no internal pullup resistor connected. The MODE register is initialized to 0Fh, which allows immediate access to the data direction registers using the “MOV !rx,W” instruction.

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 CF bit is cleared, Carry bit works as input for ADD and SUB instructions.

5.0 DEVICE CONFIGURATION REGISTERS

The SX device has three registers (FUSE, FUSEX, DEVICE) that control functions such as operating the device in Turbo mode, extended (8-level deep) stack operation, and speed selection for the internal RC oscillator. These registers are not programmable “on the fly” during normal device operation. Instead, the FUSE and FUSEX registers can only be accessed when the SX device is being programmed. The DEVICE register is a read-only, hard-wired register, programmed during the manufacturing process.

5.1 FUSE Word (Read/Program at FFFh in main memory map)

$\overline{\text{TURBO}}$	$\overline{\text{SYNC}}$	Reserved	Reserved	$\overline{\text{IRC}}$	DIV1/ IFBD	DIV0/ FOSC2	Re-served	$\overline{\text{CP}}$	WDTE	FOSC1	FOSC0
Bit 11											**Bit 0**

$\overline{\text{TURBO}}$ Turbo mode enable:
- 0 = turbo (instruction clock = osc/1)
- 1 = instr clock = osc/4

$\overline{\text{SYNC}}$ Synchronous input enable (for turbo mode): This bit synchronizes the signal presented at the input pin to the internal clock through two internal flip-flops.
- 0 = enabled
- 1 = disabled

$\overline{\text{IRC}}$ Internal RC oscillator enable:
- 0 = enabled - OSC1 pulled low by weak pullup, OSC2 pulled high by weak pullup
- 1 = disabled - OSC1 and OSC2 behave according to FOSC2: FOSC0

DIV1: DIV0 Internal RC oscillator divider:
- 00b = 4 MHz
- 01b = 1 MHz
- 10 = 128 KHz
- 11b = 32 KHz

$\overline{\text{IFBD}}$ Internal crystal/resonator oscillator feedback resistor (1 MΩ):
- 0= disabled Internal feedback resistor disable (external feedback required)
- 1= enabled Internal feedback resistor enabled (valid when $\overline{\text{IRC}}$ = 1)

$\overline{\text{CP}}$ Code protect enable:
- 0 = enabled (FUSE, code, and ID memories read back as garbled data)
- 1 = disabled (FUSE, code, and ID memories can be read normally)

WDTE Watchdog timer enable:
- 0 = disabled
- 1 = enabled

FOSC2: FOSC0 External oscillator configuration (valid when $\overline{\text{IRC}}$ = 1):
- 000b = LP1 – low power crystal (32KHz)
- 001b = LP2 – low power crystal/resonator (32 KHz to 1 MHz)
- 010b = XT1 – normal crystal/resonator (32 KHz to 10 MHz)
- 011b = XT2 – normal crystal/resonator (1MHz to 24 MHz)
- 100b = HS1 – high speed crystal/resonator/external crystal oscillator (1MHz to 50 MHz)
- 101b = HS2 – high speed crystal/resonator/external crystal oscillator (1 MHz to 50 MHz)
- 110b = HS3 – high speed crystal/resonator/external crystal oscillator (1 MHz to 75 MHz)
- 111b = RC network - OSC2 is pulled high with a weak pullup (no CLKOUT output)
- Note: The frequencies are target values.

5.2 FUSEX Word (Read/Program via Programming Command)

IRCTRIM2	PINS	IRCTRIM1	IRCTRIM0	$\overline{\text{OPTIONX}}$/ STACKX	$\overline{\text{CF}}$	BOR1	BOR0	BORTRIM1	BORTRIM0	BP1	BP0
Bit 11											**Bit 0**

IRCTRIM2: IRCTRIM0 — Internal RC oscillator trim bits. This 3-bit field adjusts the operation of the internal RC oscillator to make it operate within the target frequency range 4 MHz plus or minus 8%. Parts are shipped from the factory untrimmed. The device relies on the programming toll to provide the trimming function.

000b = minimum frequency

111b = maximum frequency

each step about 3%

PINS — Selects the number of pins.

0 = 18/20 pins

1 = 28 pins

$\overline{\text{OPTIONX}}$/ $\overline{\text{STACKX}}$ — OPTION Register Extension and Stack Extension. Set to 1 to disable the programmability of bit 6 and bit 7 in the OPTION register, the RTW and RTE_IE bits (in other words, to force these two bits to 1) and to limit the program stack size to two locations. Clear to 0 to enable programming of the RTW and RTE_IE bits in the OPTION register, and to extend the stack size to eight locations.

$\overline{\text{CF}}$ — active low – makes carry bit input to ADD and SUB instructions.

BOR1: BOR0 — Brown-Out Reset;These bits enable or disable the brown-out reset function and set the brown-out threshold voltage as follows:

00b = 4.2V

01b = 2.6V

10b = 2.2V

11b = Brown-Out disabled

BORTRIM1: BORTRIM0 — Brown-Out trim bits (parts are shipped out of factory untrimmed).

BP1:BP0 — Configure Memory Size:

00b = 1 page, 1 bank

01b = 2 pages, 1 bank

10b = 4 pages, 4 banks

11b = 4 pages, 8 banks (default configuration)

5.3 DEVICE Word (Hard-Wired Read-Only)

1	1	1	1	1	1	0	0	1	1	1	0
Bit 11											**Bit 0**

6.0 MEMORY ORGANIZATION

6.1 Program Memory

The program memory is organized as 2K, 12-bit wide words. The program memory words are addressed sequentially by a binary program counter. The program counter starts at zero. If there is no branch operation, it will increment to the maximum value possible for the device and roll over and begin again.

Internally, the program memory has a semi-transparent page structure. A page is composed of 512 contiguous program memory words. The lower nine bits of the program counter are zeros at the first address of a page and ones at the last address of a page. This page structure has no effect on the program counter. The program counter will freely increment through the page boundaries.

6.1.1 Program Counter

The program counter contains the 11-bit address of the instruction to be executed. The lower eight bits of the program counter are contained in the PC register (02h) while the upper bits come from the upper three bits of the STATUS register (PA0, PA1, PA2). This is necessary to cause jumps and subroutine calls *across* program memory page boundaries. Prior to the execution of a branch operation, the user program must initialize the upper bits of the STATUS register to cause a branch to the desired page. An alternative method is to use the PAGE instruction, which automatically causes branch to the desired page, based on the value specified in the operand field. Upon reset, the program counter is initialized with 07FFh.

6.1.2 Subroutine Stack

The subroutine stack consists of eight 11-bit save registers. A physical transfer of register contents from the program counter to the stack or vice versa, and within the stack, occurs on all operations affecting the stack, primarily calls and returns. The stack is physically and logically separate from data RAM. The program cannot read or write the stack.

6.2 Data Memory

The data memory consists of 136 bytes of RAM, organized as eight banks of 16 registers plus eight registers which are not banked. Both banked and non-banked memory locations can be addressed directly or indirectly using the FSR (File Select Register). The special-function registers are mapped into the data memory.

6.2.1 File Select Register (04h)

Instructions that specify a register as the operand can only express five bits of register address. This means that only registers 00h to 1Fh can be accessed. The File Select Register (FSR) provides the ability to access registers beyond 1Fh.

Figure 6-1 shows how FSR can be used to address RAM locations. The three high-order bits of FSR select one of eight SRAM banks to be accessed. The five low-order bits select one of 32 SRAM locations within the selected bank. For the lower 16 addresses, Bank 0 is always accessed, irrespective of the three high-order bits. Thus, RAM register addresses 00h through 0Fh are “global” in that they can always be accessed, regardless of the contents of the FSR.

The entire data memory (including the dedicated-function registers) consists of the lower 16 bytes of Bank 0 and the upper 16 bytes of Bank 0 through Bank 7, for a total of (1+8)*16 = 144 bytes. Eight of these bytes are for the function registers, leaving 136 general-purpose memory locations. In the 18-pin SX packages, register RC is not used, which makes address 07h available as an additional general-purpose memory location.

Below is an example of how to write to register 10h in Bank 4:

```
mov     FSR,#$90    ;Select Bank 4 by
                    ;setting FSR<7:5>
mov     $10,#$64    ;load register 10h with
                    ;the literal 64h
```

8.0 INTERRUPT SUPPORT

The device supports both internal and external maskable interrupts. The internal interrupt is generated as a result of the RTCC rolling over from 0FFh to 00h. This interrupt source has an associated enable bit located in the OPTION register. There is no pending bit associated with this interrupt.

Port B provides the source for eight external software selectable, edge sensitive interrupts. These interrupt sources share logic with the Multi-Input Wakeup circuitry. The WKEN_B register allows interrupt from Port B to be individually enabled or disabled. Clearing a bit in the WKEN_B register enables the interrupt on the corresponding Port B pin. The WKED_B selects the transition edge to be either positive or negative. The WKEN_B and WKED_B registers are set to FFh upon reset. Setting a bit in the WKED_B register selects the falling edge while clearing the bit selects the rising edge on the corresponding Port B pin.

The WKPND_B register serves as the external interrupt pending register.

The WKPND_B register comes up a with random value upon reset. The user program must clear the WKPND_B register prior to enabling the interrupt. The proper sequence is described in Section 7.2 .

Figure 8-1 shows the structure of the interrupt logic.

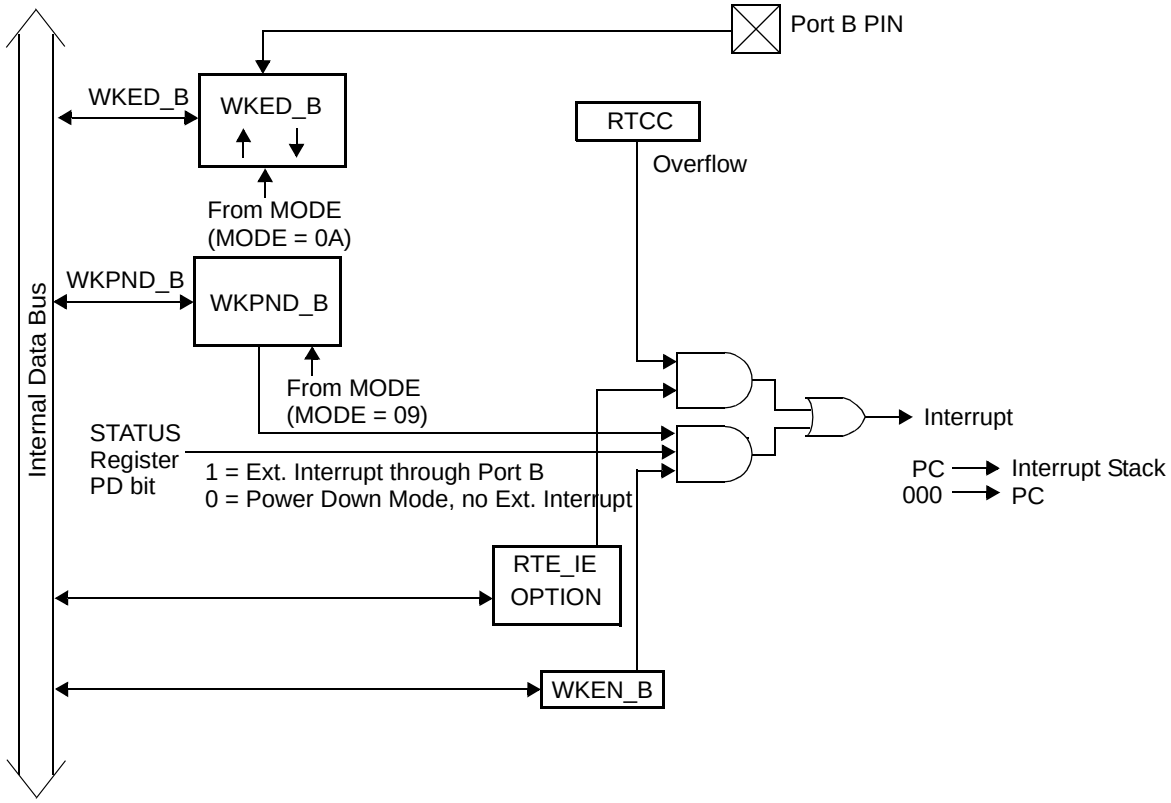


Figure 8-1. Interrupt Structure

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.

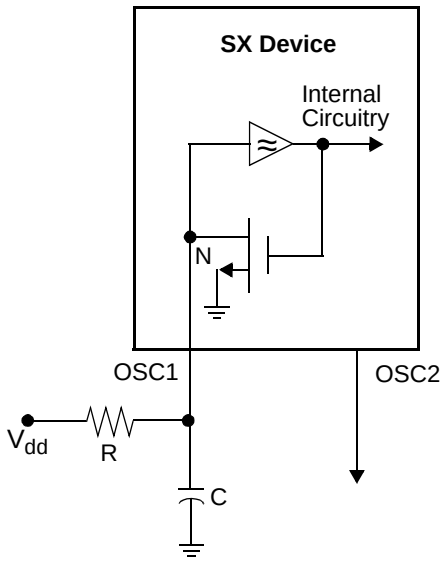


Figure 9-3. RC Oscillator Mode

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.

10.2 Watchdog Timer

The watchdog logic consists of a Watchdog Timer which shares the same 8-bit programmable prescaler with the RTCC. The prescaler actually serves as a postscaler if used in conjunction with the WDT, in contrast to its use as a prescaler with the RTCC.

10.3 The Prescaler

The 8-bit prescaler may be assigned to either the RTCC or the WDT through the PSA bit (bit 3 of the OPTION register). Setting the PSA bit assigns the prescaler to the WDT. If assigned to the WDT, the WDT clocks the prescaler and the prescaler divide rate is selected by the PS0, PS1, and PS2 bits located in the OPTION register. Clearing the PSA bit assigns the prescaler to the RTCC. Once assigned to the RTCC, the prescaler clocks the RTCC and the divide rate is selected by the PS0, PS1, and PS2 bits in the OPTION register. The prescaler is not mapped into the data memory, so run-time access is not possible.

The prescaler cannot be assigned to both the RTCC and WDT simultaneously.

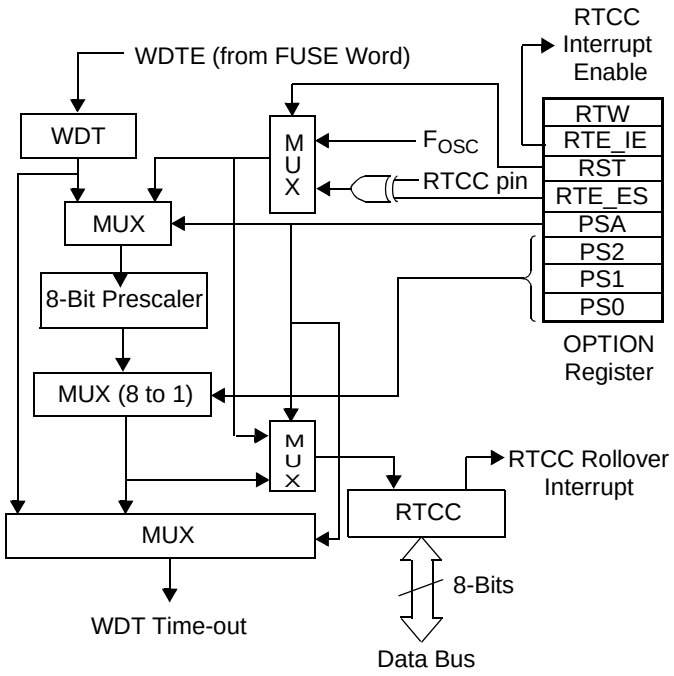
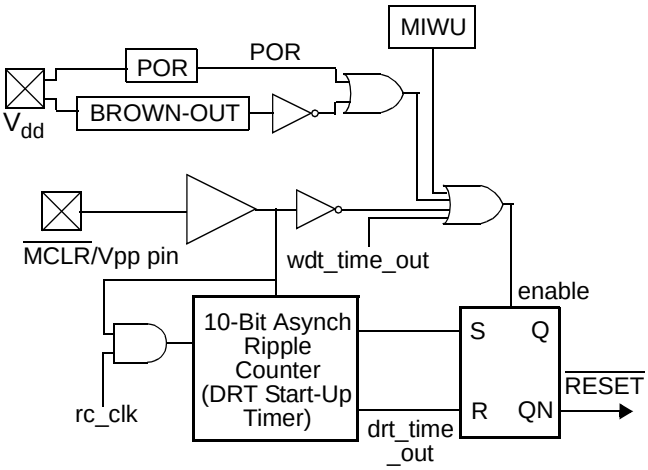


Figure 10-1. RTCC and WDT Block Diagram

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 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.



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 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.

.

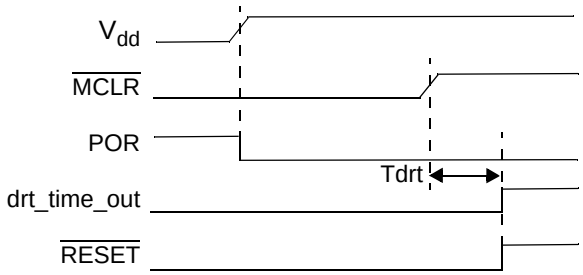


Figure 12-2. Time-Out Sequence on Power-Up (MCLR not tied to V_{dd})

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 $\overline{CF}$ 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 $\overline{CF}$ 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 $\overline{CF}$ 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
Bitwise Operations					
CLRB fr.bit	Clear Bit in fr (fr.bit = 0)	1	1	0100 bbbf ffff	none
SB fr.bit	Test Bit in fr and Skip if Set (test fr.bit and skip next instruction if bit is 1)	1 or 2 (skip)	1 or 2 (skip)	0111 bbbf ffff	none
SETB fr.bit	Set Bit in fr (fr.bit = 1)	1	1	0101 bbbf ffff	none
SNB fr.bit	Test Bit in fr and Skip if Clear (test fr.bit and skip next instruction if bit is 0)	1 or 2 (skip)	1 or 2 (skip)	0110 bbbf ffff	none
Data Movement Instructions					
MOV fr,W	Move W to fr (fr = W)	1	1	0000 001f ffff	none
MOV W,fr	Move fr to W (W = fr)	1	1	0010 000f ffff	Z
MOV W,fr-W	Move (fr-W) to W (W = fr - W); complement of carry bit is subtracted if CF bit in FUSEX register is cleared to 0	1	1	0000 100f ffff	C, DC, Z
MOV W,#lit	Move Literal to W (W = lit)	1	1	1100 kkkk kkkk	none
MOV W,/fr	Move Complement of fr to W (W = fr ^ FFh)	1	1	0010 010f ffff	Z
MOV W,--fr	Move (fr-1) to W (W = fr - 1)	1	1	0000 110f ffff	Z
MOV W,++fr	Move (fr+1) to W (W = fr + 1)	1	1	0010 100f ffff	Z
MOV W,<<fr	Rotate fr Left through Carry and Move to W (W = << fr)	1	1	0011 010f ffff	C
MOV W,>>fr	Rotate fr Right through Carry and Move to W (W = >> fr)	1	1	0011 000f ffff	C
MOV W,<>fr	Swap High/Low Nibbles of fr and move to W (W = <> fr)	1	1	0011 100f ffff	none
MOV W,M	Move MODE Register to W (W = MODE), high nibble is cleared	1	1	0000 0100 0010	none
MOVSZ W,--fr	Move (fr-1) to W and Skip if Zero (W = fr -1 and skip next instruction if result is zero)	1 or 2 (skip)	1 2 (skip)	0010 110f ffff	none
MOVSZ W,++fr	Move (fr+1) to W and Skip if Zero (W = fr + 1 and skip next instruction if result is zero)	1 or 2 (skip)	1 2 (skip)	0011 110f ffff	none
MOV M,W	Move W to MODE Register (MODE = W)	1	1	0000 0100 0011	none
MOV M,#lit	Move Literal to MODE Register (MODE = lit)	1	1	0000 0101 kkkk	none
MOV !rx,W	Move W to Port Rx Control Register:rx <=> W (exchange W and WKPND_B or CMP_B) *or* rx = W (move W to rx for all other port control registers)	1	1	0000 0000 0fff	none
MOV !OPTION, W	Move W to OPTION Register (OPTION = W)	1	1	0000 0000 0010	none
TEST fr	Test fr for Zero (fr = fr to set or clear Z bit)	1	1	0010 001f ffff	Z

17.3 AC 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	Min	Typ	Max	Units	Conditions
F_{osc}	External CLKIN Frequency	DC	-	32 1.0 4.0 10 24 50 75	KHz MHz MHz MHz MHz MHz MHz	LP1 LP2 RC XT1 XT2 HS1/HS2/HS3 HS3
	Oscillator Frequency	DC 0.032 DC 0.032 1.0 1.0 1.0	-	32 1.0 4.0 10.0 24.0 50 75	KHz MHz MHz MHz MHz MHz MHz	LP1 LP2 RC XT1 XT2 HS1/HS2/HS3 HS3
T_{osc}	External CLKIN Period	31.25 1.0 250 100 41.7 20 13.3	-	-	µs µs ns ns ns ns	LP1 LP2 RC XT1 XT2 HS1/HS2/HS3
	Oscillator Period	31.25 1.0 250 0.1 41.7 20 13.3	-	- 31.25 - 31.25 1000.0 1000.0 -	µs µs µs µs ns ns ns	LP1 LP2 RC XT1 XT2 HS1/HS2/HS3 HS3
T_{osL}, T_{osH}	Clock in (OSC1) Low or High Time	2.0 50 8.0 5.3	-	-	µs ns ns ns	LP1/LP2 XT1/XT2 HS1/HS2/HS3 HS3

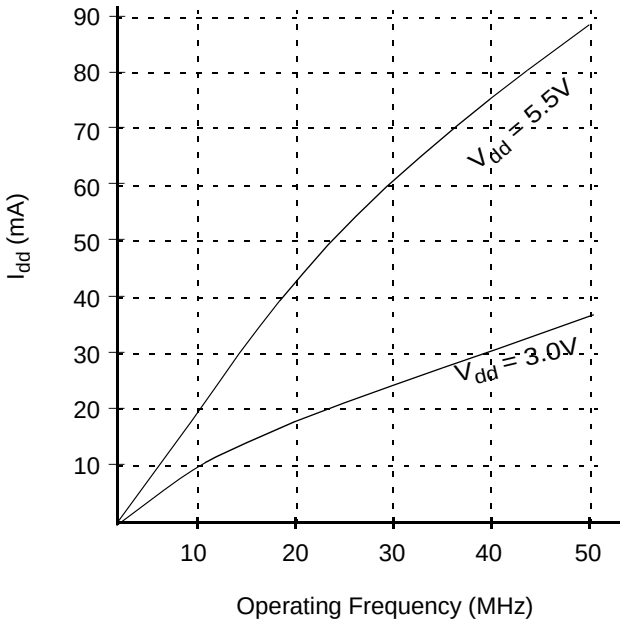
Note: Data in the Typical (“TYP”) column is at 25°C unless otherwise stated.

17.4 Comparator DC and AC Specifications (50 MHz and 75 MHz Operation)

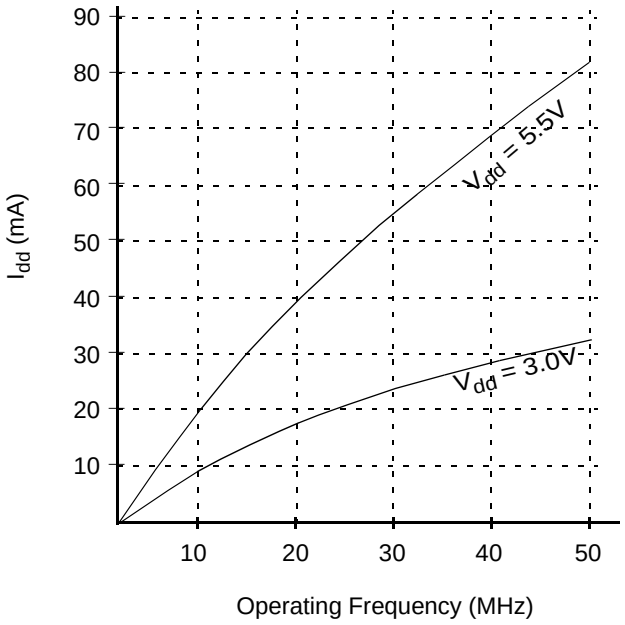
Parameter	Conditions	Min	Typ	Max	Units
Input Offset Voltage	0.4V < Vin < Vdd – 1.5V		+/- 10	+/- 25	mV
Input Common Mode Voltage Range		0.4		Vcc – 1.3	V
Voltage Gain			300k		V/V
DC Supply Current (enabled)	Vdd = 5.5V			120	µA
Response Time	$V_{overdrive}$ = 25mV			250	ns

17.5 Typical Performance Characteristics (25°C)

Active Supply Current Vs Operating Frequency (Crystal Clock)



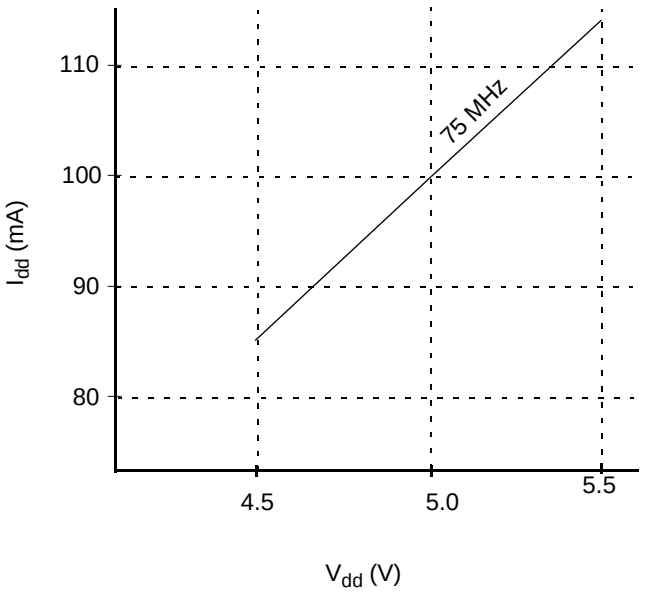
Active Supply Current Vs Operating Frequency (External Clock)



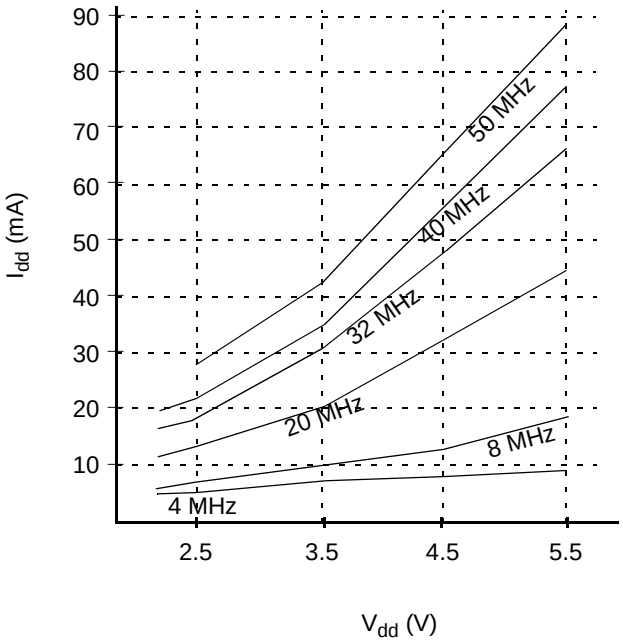
17.7 Typical Performance Characteristics (Continued)

Active Supply Current Vs Operating Frequency (External Clock)

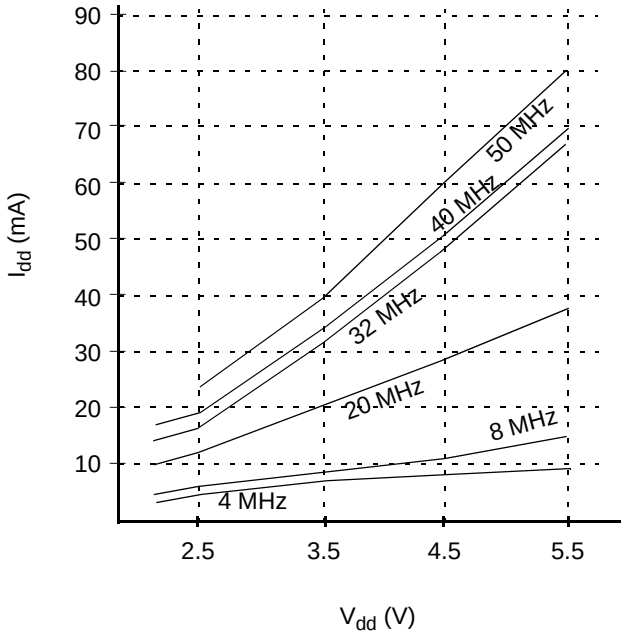
SX18AC75/SX20AC75/SX28AC75



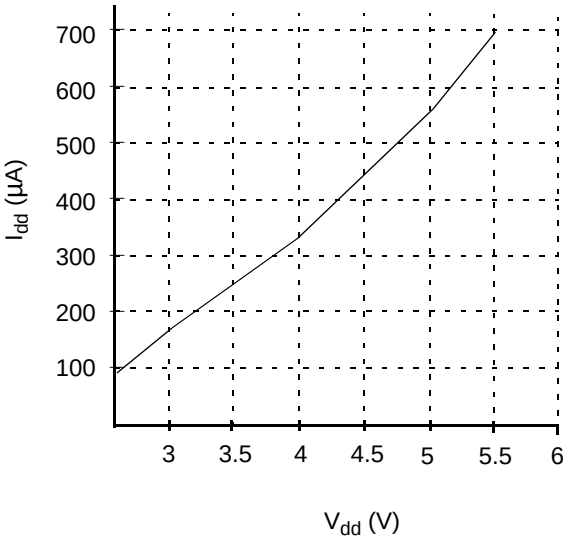
Active Supply Current Vs V_{dd} (Crystal Clock)



Active Supply Current Vs V_{dd} (External Clock)

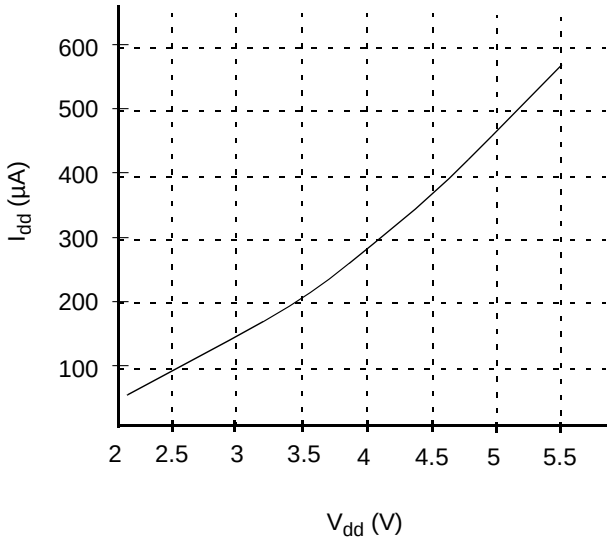


Active Supply Current Vs V_{dd} (32 kHz Crystal Clock)

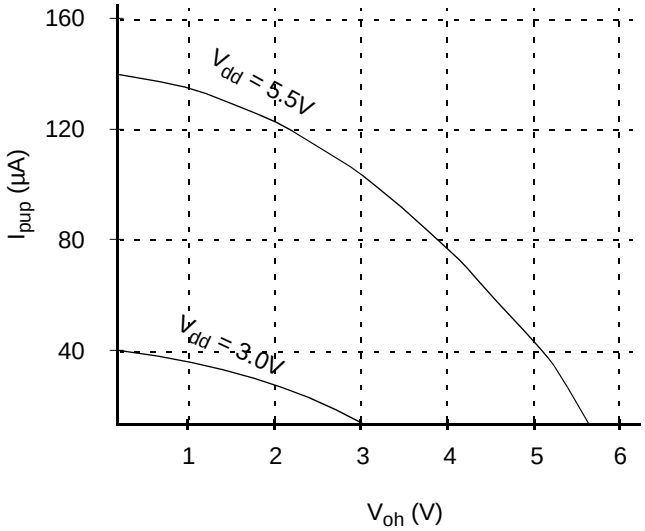


17.7 Typical Performance Characteristics (25°C)

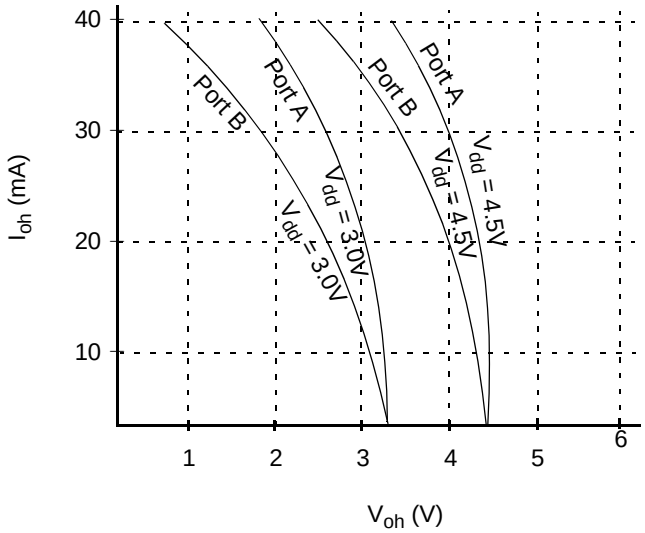
Active Supply Current Vs V_{dd} (32 kHz External Clock)



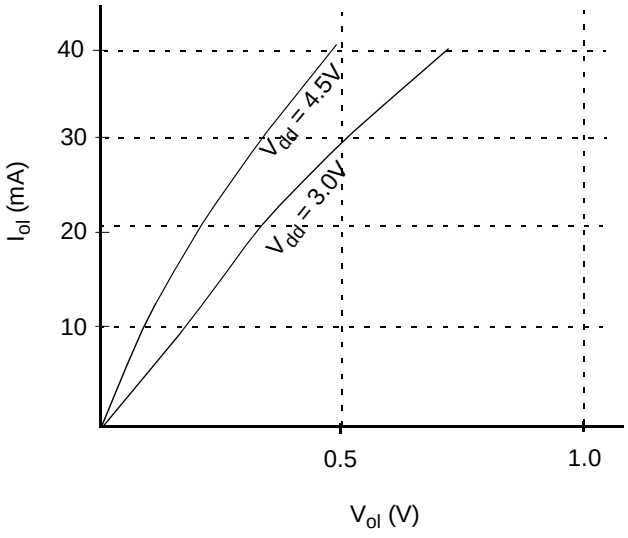
Port A/B/C Weak Pull-Up Source Current



Port A/B/C Source Current



Port A/B/C Sink Current



SX28AC/DP