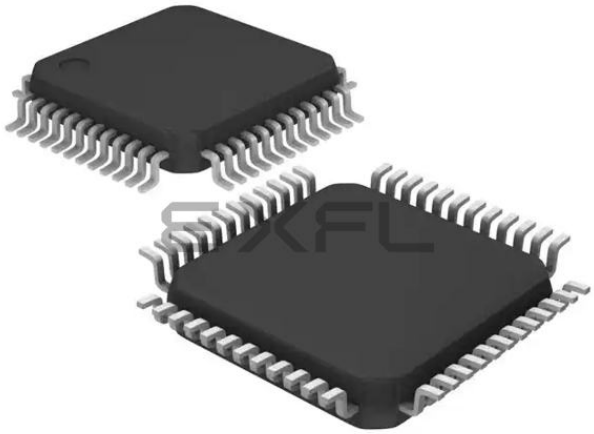


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

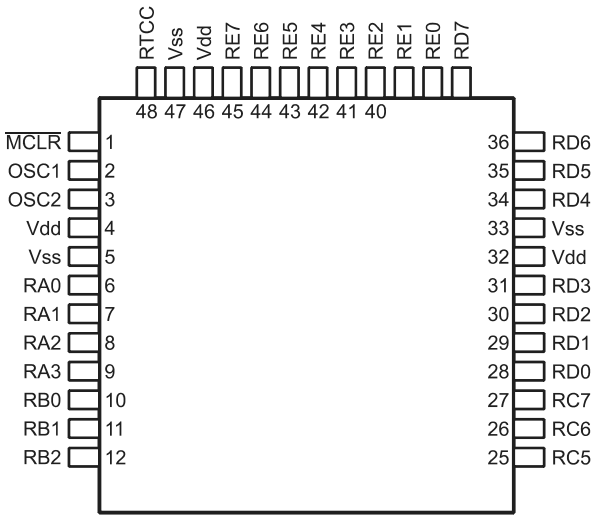
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	50MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	6KB (4K x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	262 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	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/parallax/sx48bd-g

2.0 CONNECTION DIAGRAMS

2.1. Pin Assignments



Top View

11.1. Timer Registers

Each timer consists of several registers.

Timer T1 registers:

T1CPL - Lower byte of Timer T1 capture register
T1CPH - Higher byte of Timer T1 capture register
T1R1CML - Lower byte of Timer T1 compare register 1
T1R1CMH - Higher byte of Timer T1 compare register 1
T1R2CML - Lower byte of Timer T1 compare register 2
T1R2CMH - Higher byte of Timer T1 compare register 2
T1CNTA - Timer T1 control register A
T1CNTB - Timer T1 control register B

Timer T2 registers:

T2CPL - Lower byte of Timer T2 capture register
T2CPH - Higher byte of Timer T2 capture register
T2R1CML - Lower byte of Timer T2 compare register 1
T2R1CMH - Higher byte of Timer T2 compare register 1
T2R2CML - Lower byte of Timer T2 compare register 2
T2R2CMH - Higher byte of Timer T2 compare register 2
T2CNTA - Timer T1 control register A
T2CNTB - Timer T1 control register B

11.2. Timer Operating Modes

Each timer can be configured to operate in one of the following modes:

- Pulse Width Modulation (PWM) mode
- Software Timer mode
- External Event mode
- Capture/Compare mode

11.2.1. PWM Mode

In the Pulse Width Modulation (PWM) mode, the timer generates an output signal having a programmable frequency and duty cycle. To use this mode, you load two 16-bit comparison registers, R1 and R2, with the number of timer clock cycles that you want the output signal to be high and low. The contents of R1 define the PWM low time while the contents of R2 define the PWM high time.

After the “Clear Timer” command is initiated through the MODE register, the timer starts from zero and counts up until it reaches the value in R1. At that point, it generates an interrupt (if enabled), toggles the output signal to a logic high level, and starts counting from zero again. The second time, it counts up until it reaches the value in R2. At that point, it again generates an interrupt (if enabled), toggles the output signal to a logic low level, and starts counting from zero again. This process is repeated continuously, alternating between R1 and R2 to obtain the value at which to toggle the output signal and return the counter to zero. The values of R1 and R2 establish the duty cycle and frequency of the output signal. If R1 and R2 contain the same value, the resulting output signal is a square wave. If R1 is changed to a value less than the timer count while the timer is counting to match R1, the timer will continue to count through FFFFh, and back up to the R1 value, while the output is low. Same is true for R2, except the output signal will be high.

Upon reset, the timer/counter is initialized to 0000.

In the PWM mode, the timer is clocked by the on-chip system clock divided by an 8-bit prescaler value. The divide-by factor can be set to any power-of-2 from 1 to 256. Thus, the period of the timer clock can be set from 1 to 256 times the system clock period.

Upon entering the PWM mode, the internally generated PWM signal is connected to the designated PWM output pin. The PWM mode bypasses the port data register (does not affect the contents of the data register). For the PWM output signal to appear on the pin (RB6 for T1, RC2 for T2), the corresponding port pin direction register must be configured for output.

11.2.2. Software Timer Mode

The Software Timer mode is the same as the PWM mode, except that the timer does not toggle the output signal. Instead, the application program takes action in response to the internally generated PWM signal upon each match between the counter and either contents of the active comparison value in either R1 or R2. The software can determine the cause of each interrupt by checking the timer interrupt pending flags. There are different flag bits associated with each type of event (R1 match, R2 match, and overflow).

11.2.3. External Event Mode

The External Event mode is the same as the PWM mode, except that the counter register is clocked by an external signal provided on an input pin (RB7 for T1 and RC3 for T2) rather than by the system clock. This mode can be used to count the occurrences of external events. The input pin can be configured to sense either rising or falling edges.

11.2.4. Capture/Compare Mode

In the Capture/Compare mode, the counter counts upward continuously without interruption. A valid transition received on either of two input pins causes the current value of the counter to be captured in an associated capture register. This capture feature can be used to keep track of the elapsed time between successive external events. In addition, the timer continuously compares the counter value against the value programmed into the R1 register. Each time a match occurs, it toggles the timer output pin, generates an interrupt (if enabled) and sets an associated interrupt pending flag. The timer continues to

Timer T1 Control B Register (T1CNTB)

RTCCOV	T1CPEDG	T1EXEDG	T1PS2 – T1PSO			T1MC1 – T1MC0	
7	6	5	4	3	2	1	0

RTCCOV RTCC Overflow Flag. This flag is automatically set to 1 when the Real-Time Clock/Counter (RTCC) overflows from FFh to 00h. This flag stays set until it is cleared by software. Note that this flag is not related to Multi-function timers T1 and T2.

T1CPEDG Timer T1 Capture Edge. This bit sets the edge sensitivity of the Timer T1 input capture pins, Capture 1 and Capture 2 (RB4 and RB5). Set this bit to 1 to sense positive-going (low-to-high) edges. Clear this bit to 0 to sense negative-going (high-to-low) edges.

T1EXEDG Timer T1 External Event Clock Edge. This bit sets the edge sensitivity of the Timer T1 input used to count external events (RB7). Set this bit to 1 to use positive-going (low-to-high) edges. Clear this bit to 0 to sense negative-going (high-to-low) edges.

T1PS2-T1PSO Timer T1 Prescaler Divider field. This 3-bit field specifies the divide-by factor for generating the timer T1PS0 clock from the on-chip system clock:

000 = divide by 1

001 = divide by 2

010 = divide by 4

011 = divide by 8

100 = divide by 16

101= divide by 32

110 = divide by 64

111 = divide by 128

For example, setting this field to 010 sets the divide-by factor to 4, which means that the T1 counter register is incremented once every four system clock cycles.

T1MC1-T1MCO Timer T1 Mode Control field. This 2-bit field specifies the Timer T1 operating mode as follows:

00 = Software Timer mode

01 = PWM mode

10 = Capture/Compare mode

11 = External Event mode

Timer T2 Control b Register (T2CNTb)

PRTRD	T2CPEDG	T2EXEDG	T2PS2 – T2PS0			T2MC1 – T2MC0	
7	6	5	4	3	2	1	0

PORTRD Port Read mode. This bit determines how the device reads data from its I/O ports (Port A through Port E). Clear this bit to 0 to have the device read data from the I/O pins directly. Set this bit to 1 to have the device read data from the port data registers latch. Under normal (output mode) conditions, it should not matter which method you use to read the port data. However, if a port pin is configured as an output and an external circuit forces the pin to the wrong value, the value read from the port will depend on the reading mode used. Note that this control bit is not related to multi-function timers T1 and T2.

T2CPEDG Timer T2 Capture Edge. This bit sets the edge sensitivity of the Timer T2 input capture pins, Capture 1 and Capture 2 (RC0 and RC1). Set this bit to 1 to sense positive-going (low-to-high) edges. Clear this bit to 0 to sense negative-going (high-to-low) edges.

T2EXEDG Timer T2 External Event Clock Edge. This bit sets the edge sensitivity of the Timer T2 input used to count external events (RC3). Set this bit to 1 to use positive-going (low-to-high) edges. Clear this bit to 0 to sense negative-going (high-to-low) edges.

T2PS2-T2PS0 Timer T2 Prescaler Divider field. This 3-bit field specifies the divide-by factor for generating the timer clock from the on-chip system clock:

000 = divide by 1

001 = divide by 2

010 = divide by 4

011 = divide by 8

100 = divide by 16

101 = divide by 32

110 = divide by 64

111 = divide by 128

For example, setting this field to 010 sets the divide-by factor to 4, which means that the T2 counter register is incremented once every four system clock cycles.

T2MC1- T2MC0 Timer T2 Mode Control field. This 2-bit field specifies the Timer T1 operating mode as follows:

00 = Software Timer mode

01 = PWM mode

10 = Capture/Compare mode

11 = External Event mode

16.0 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.

16.1. Instruction Set Features

- All single-word (12-bit) instructions for compact code efficiency.
- 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.
- 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.
- Many instructions include a destination bit which selects either the register file or the accumulator as the destination for the result.
- Bit manipulation instructions (Set, Clear, Test and Skip if Set, Test and Skip if Clear).
- STATUS Word register memory-mapped as a register file, allowing testing of status bits (carry, digit carry, zero, power down, and timeout).
- Program Counter (PC) memory-mapped as register file allows W to be used as offset register for indirect addressing of program memory.
- Indirect addressing data pointer FSR (file select register) memory-mapped as a register file.
- IREAD instruction allows reading the instruction from the program memory addressed by W and upper four bits of MODE register.
- Eight-level, 12-bit push/pop hardware stack for subroutine linkage using the Call and Return instructions.
- Seven addressing modes provide great flexibility.

16.2. Instruction Execution

An instruction goes through a four-stage pipeline to be executed (Figure 16-1). 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 instructions take more than one clock cycle.

The instruction execution time is derived by dividing the oscillator frequency by one (bit 11 of the FUSE Word register must be initialized to 0).

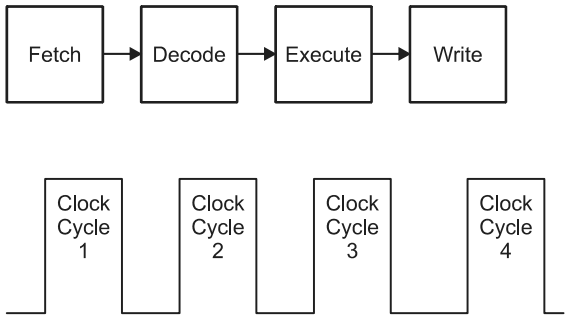


Figure 16-1: Pipeline and Clock Scheme

16.3. Addressing Modes

The device support the following addressing modes:

- Data Direct
- Data Indirect
- Data Semidirect
- 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. Any 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     W,#1
mov     RA,W            ;move "1" to RA
```

Indirect Addressing:

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

Semidirect Addressing:

```
mov     W,#$00
mov     FSR,W           ;FSR = bank 0 address
inc     $1F             ;increment file
                        ;register 0Fh
```

16.4. The Bank Instruction

Often it is desirable to set the bank select bits of the FSR register in one instruction cycle. The Bank instruction provides this capability. This instruction sets the upper 3 bits 4, 5 and 6 of the FSR to point to a specific RAM bank without affecting the lower 4 FSR bits, in preparation for using direct or semidirect addressing. Bit 7 of the FSR register is used to select the lower or upper block of banks.

Example:

```
bank    $E0     ;Select Bank E in FSR
inc     $1F     ;increment file register
                ;EFh using semidirect addressing
```

16.5. Bit Manipulation

The instruction set contains instructions to set, reset, and test individual bits in data memory. The device is capable of bit addressing anywhere in data memory.

16.6. Input/Output Operation

The device contains three registers associated with each I/O port. The first register (Data Direction Register), configures each port pin as a Hi-Z input or output. The second register (TTL/CMOS Register), selects the desired input level for the input. The third register (Pull-Up Register), enables a weak pull-up resistor on the pin configured as a input. To read or write these registers, you must first write an appropriate value into the MODE register to select the desired register set, and then use the "mov !rx,W" instruction to read or write the register.

16.6.1. Read-Modify-Write Considerations

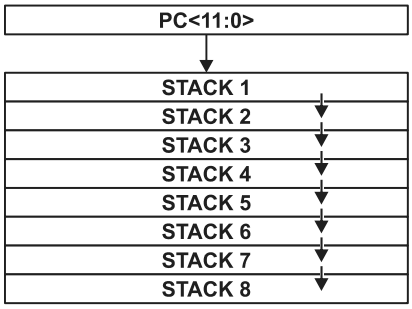
When two successive instructions are used on the same I/O port (except "mov Rx, W") with a very high clock rate, the "write" part of one instruction might not occur soon enough before the "read" part of the very next instruction, resulting in getting "old" data for the second instruction. To ensure predictable results, avoid using two successive read-modify-write instructions that access the same port data register if the clock rate is high or, insert 3 NOP instructions between the successive read-modify-write instructions (if SYNC bit in the FUSE register is enabled, 5 NOP instructions are required). For operating frequencies of 50 Mhz or lower, if bit 7 of the T2CNTB (PORTRD) is set, the port reads data from the data register instead of port pins. In this case, the NOP instructions are not required.

In the default device configuration, when a read is performed from a port bit position, 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. PORTRD of the T2CNTB register determines how the device reads data from its I/O ports (Port A through Port E). Clear this bit to 0 to have the device read data from the port I/O pins directly. Set this bit to 1 to have the device read data from the port data registers. Under normal output mode conditions, it should not matter which method you use to read the port data. However, if a port pin is configured as an output and an external circuit forces the pin to the opposite value, the value read from the port will depend on the reading mode used. Note that this control bit is not related to multi-function timers T1 and T2.

16.11. Subroutine Operation

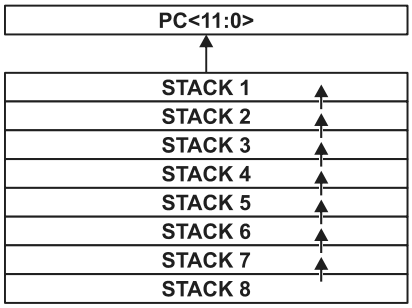
16.11.1. Push Operation

When a subroutine is called, the return address is pushed onto the subroutine stack. Specifically, each address in the stack is moved to the next lower level in order to make room for the new address to be stored. Stack 1 receives the contents of the program counter. Stack 8 is overwritten with what was in Stack 7. The contents of stack 8 are lost.



16.11.2. Pop Operation

When a return instruction is executed the subroutine stack is popped. Specifically, the contents of Stack 1 are copied into the program counter and the contents of each stack level are moved to the next higher level. For example, Stack 1 receives the contents of Stack 2, etc., until Stack 7 is overwritten with the contents of Stack 8. Stack 8 is left unchanged, so the contents of Stack 8 are duplicated in Stack 7.



16.12. 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.

16.13. Logical Instruction

The instruction set contains 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.

16.14. Shift and Rotate Instructions

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

16.15. 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.

Table 17-5: Native SX Instruction Set: Program Control Instructions				
Mnemonic, Operands	Description	Clock Cycles	Opcode	Flags Affected
CALL addr8	Call Subroutine: Top-of-stack = program counter + 1 PC(7:0) = addr8 Program counter (8) = 0 Program counter (10:9) = PA1:PA0	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	3	101k kkkk kkkk	none
NOP	No Operation	1	0000 0000 0000	none
RET	Return from subroutine (program counter = top-of-stack)	3	0000 0000 1100	none
RETP	Return from subroutine across Page boundary (PA1:PA0 = top-of-stack (10:19) and program counter = top-of-stack)	3	0000 0000 1101	PA1, PA0
RETI	Return from Interrupt (restore W, STATUS, FSR, and program counter from shadow registers)	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)	3	0000 0000 1111	All STATUS except TO, PD
RETW lit	Return from Subroutine with Liter in W (W = lit and program counter = top-of-stack)	3	1000 kkkk kkkk	none

Table 17-6: Native SX Instruction Set: System Control Instructions				
Mnemonic, Operands	Description	Clock Cycles	Opcode	Flags Affected
BANK addr8	Load Bank number into FSR(7:5) FSR(7:5) = addr8(7:5)	1	0000 0001 1nnn	none
IREAD	Read word from Instruction memory MODE:W = data at (MODE:W)	4	0000 0100 0001	none
PAGE addr12	Load Page number into STATUS(7:5) STATUS(7:5) = addr12(11:9)	1	0000 0001 0nnn	PA1, PA0
SLEEP	Power down mode WDT = 00h, TO = 1, stop oscillator (PD = 0, clears prescaler if assigned)	1	0000 0000 0011	TO, PD

17.1. Equivalent Assembler Mnemonics

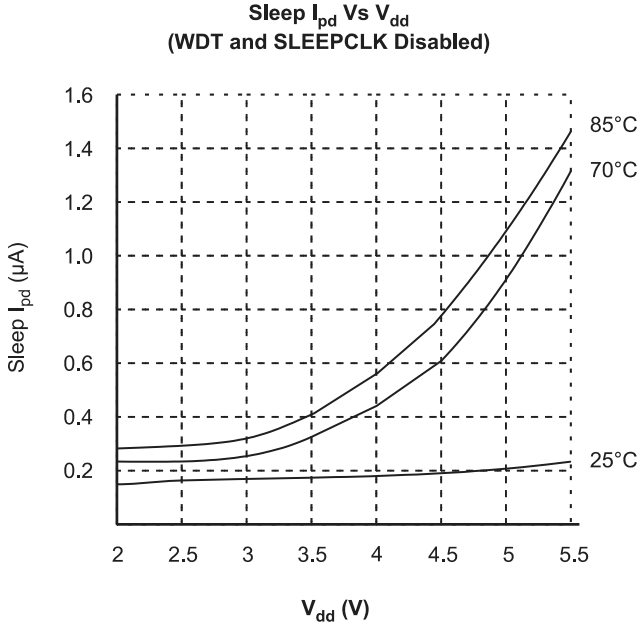
Some assemblers support additional instruction mnemonics that are special cases of existing instructions or alternative mnemonics for standard ones. For example, an assembler might support the mnemonic “CLC” (clear carry), which is interpreted the same as the instruction “clrb $03.0” (clear bit 0 in the STATUS register). Some of the commonly supported equivalent assembler mnemonics are described in Table 17-7.

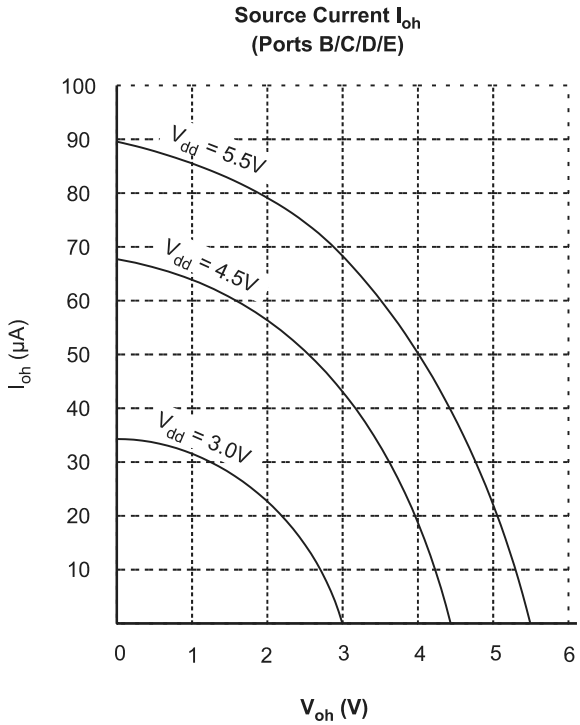
Table 17-7: SX Equivalent Assembler Mnemonics

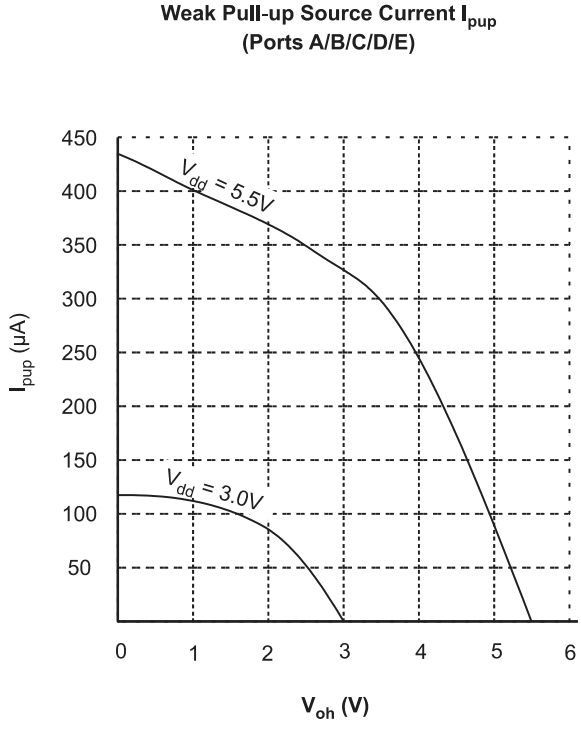
Syntax	Description	Equivalent	Cycles
CLC	Clear Carry bit	CLRB $03.0	1
CLZ	Clear Zero bit	CLRB $03.2	1
JMP W	Jump Indirect W	MOV $02,W	3
JMP PC+W	Jump Indirect W Relative	ADD $02,W	3
MODE imm4	Move Immediate to MODE Register	MOV M,#lit	1
NOT W	Complement of W	XOR W,#$FF	1
SC	Skip if Carry bit set	SB $03.0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

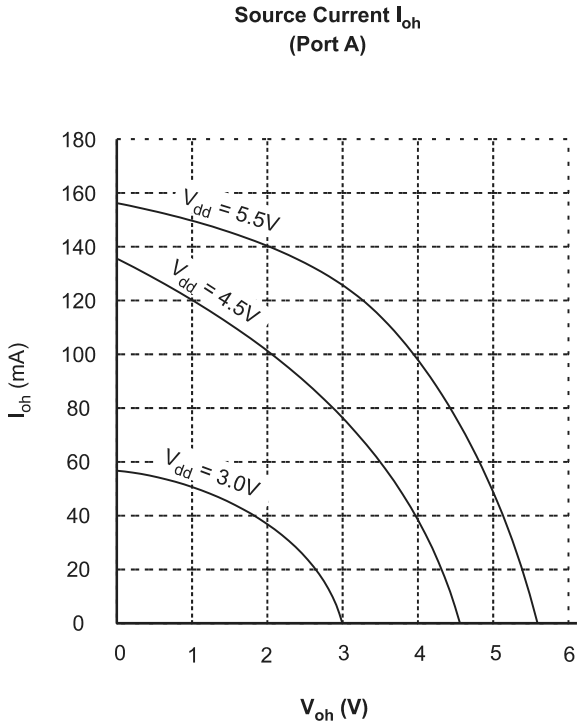
Note 1: The SC instruction takes 1 clock cycle if the tested condition is false or 2 cycles if the tested condition is true.

17.5. Typical Performance Characteristics (25 °C) (Continued)

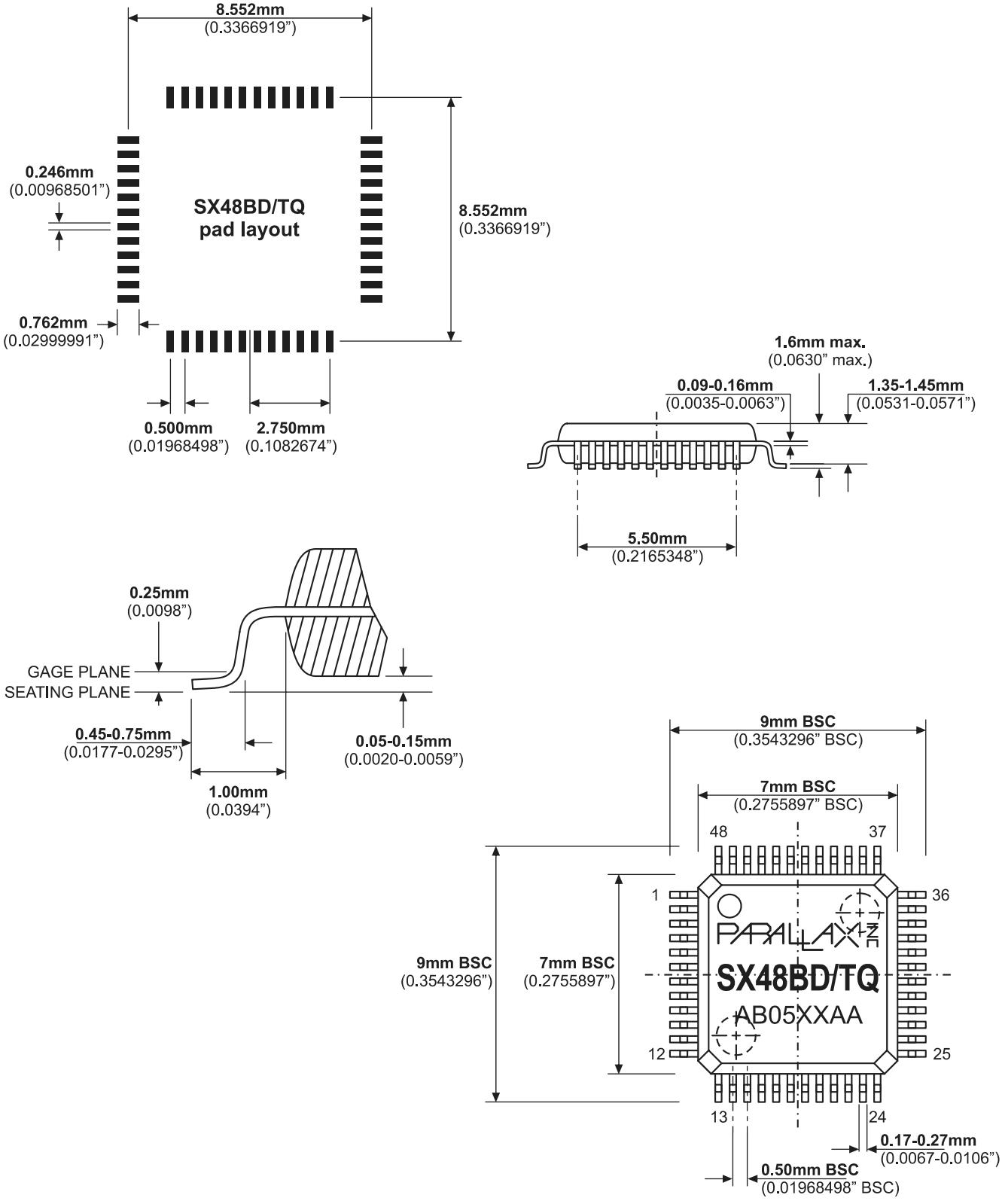








19.0 PACKAGE DIMENSIONS



20.0 MANUFACTURING INFORMATION

20.1. Reflow Peak Temperature

Package Type	Reflow Peak Temp.
Leaded	235+5/-0 °C
Green/RoHS	255+5/-0 °C

20.2. MSL3 Compliance

Chips shipped in production quantities are MSL3 compliant. For chips shipped in sample quantities or stored in compromised packaging, you may want to remove excess moisture before assembly by baking at 93 °C for 12 hours immediately before commencing soldering production.

20.3. Green/RoHS Compliance

All SX part numbers ending in "-G" are certified Green/RoHS Compliant. The certificates of Compliance are appended to this datasheet; full test reports can be obtained by contacting the Parallax Sales Team.

20.4. Stress Testing Data Summary

The Parallax SX chip is packaged by a different supplier than previously used for the Ubicom SX. For this reason Parallax engaged in a series of stress tests useful for a change of packaging suppliers. Parallax SX chips packaged at Greatek in Taiwan were exposed to these stresses between June and September 2006. The stresses were provided by a separate stress testing reliability qualification services firm (Nano Measurements in Santa Clara, California).

The stress tests chosen are common for a change of packaging suppliers. The stress testing firm conducted the tests in accordance with the applicable Joint Electron Device Engineering Council (www.jedec.com) guidelines.

The following stresses were chosen:

- Preconditioning, Level 3, 192 hours 30° C, 60% RH, preflow peak leaded and unleaded (depending on the package)
- JESD22-A118, unbiased HAST 192 hours, 110°-130° C, 85% RH, 10-20 PSIG
- JESD22-A102, autoclave, 121° C, 100% RH, 15 PSIG, non-biased, 96 hours

RoHS-compliant and standard chips were exposed to the stresses separately to further identify if failures would be associated with a particular package type. Between each stress, the parts were sent back to Parallax for our standard production test which exercises the I/O, RAM and EE/Flash memory, operating voltage and current draw.

The results are that 100% of the chips passed the Parallax standard production test, in between and after the stresses were applied to the chips.

Stock Code	Tested	Passed
SX20AC/SS-G	20	20
SX28AC/SS-G	20	20
SX28AC/DP-G	17	17
SX48BD-G	20	20
Subtotal	77	77
SX20AC/SS-G	20	20
SX28AC/SS-G	20	20
SX28AC/DP-G	17	17
SX48BD-G	20	20
Subtotal	77	77

Other stress test data are available upon request from the Parallax Research and Development team, contact Parallax Sales or Tech Support.

Parallax Sales and Tech Support Contact Information

For the latest information on SX programming tools, development boards, compilers, instructional materials, and application examples, please visit www.parallax.com/sx.

Parallax, Inc.
599 Menlo Drive, Suite 100
Rocklin, CA 95765

Sales/Tech Support: (916) 624-8333
Toll Free in the US: 1-888-512-1024
Sales: sales@parallax.com
Tech Support: support@parallax.com