Silicon Storage Technology, Inc.

FlashFlex51 MCU
SST89F54 / SST89F58

8051 Compatible Multi-Purpose 8-bit Microcontroller Unit
with Embedded SuperFlash Memory for Flexibility

User's Manual

Preliminary

407-12 2/99

Table of Contents

Preliminary

Introduction

Silicon Storage Technology, Inc. (SST) designs, manufactures and sells a variety of Electrically Erasable Programmable Read Only Memory (EEPROMs) products manufactured with SST's proprietary SuperFlash EEPROM technology. These programmable, nonvolatile memory products retain data without applied power and are much more flexible to use than other competing nonvolatile memory solutions.

When compared with alternate solutions, SST's patented processes and designs allow for the creation of high performance, high reliability and high density EEPROMs at competitive prices. Founded in 1989, SST serves the manufacturers of personal computers, notebook computers, palm computers, PC peripherals, PCMCIA cards, cellular phones, video games, electronic organizers, digital cameras, and other commercial applications requiring low power and rugged reprogrammable nonvolatile memory.

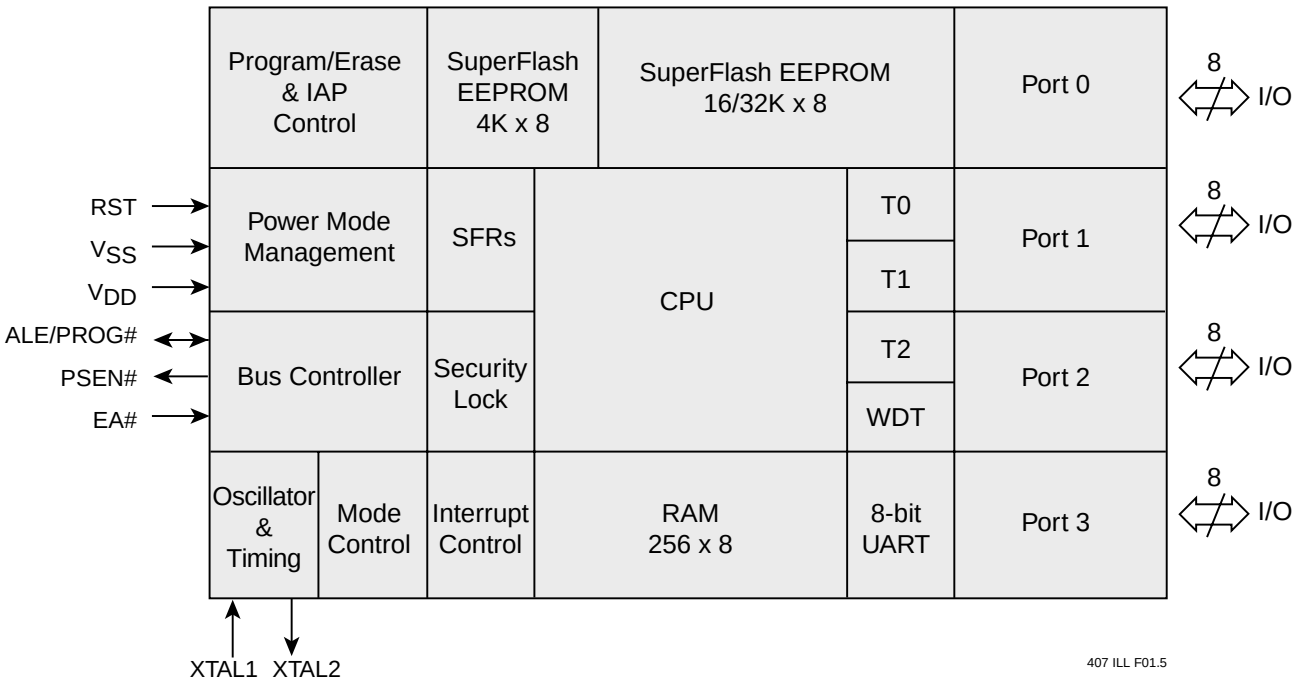
The growing use and popularity of SST's SuperFlash EEPROM technology in the flash embedded controller market has prompted the development of the SST89F54/58, the first members of the FlashFlex51 family of 8-bit, 8xC5x compatible microcontrollers. The FlashFlex51 family is a family of multi-purpose microcontroller products designed and manufactured with state-of-the-art SuperFlash CMOS semiconductor process technology. As members of the FlashFlex51 family, the SST89F54/58 use the same powerful instruction set, have the same architecture, and are pin-for-pin compatible with standard 8xC5x microcontroller devices.

The highly reliable, patented SuperFlash technology and memory cell have a number of important advantages for designing and manufacturing flash EEPROMs in the application of embedded controllers, when compared with other approaches. These advantages translate into significant cost and reliability benefits for our customers

SST has developed extensive partnerships with several major IC manufacturers. These partnerships provide SST with a guaranteed source of high quality, state-of-the-art wafers. SST agreements with large users provide the information and demand for SST to maintain a leadership position in the cost and performance driven embedded controller market.

Functional Block Diagram

The SST89F54/58 are fully compatible with the 8xC5x family of microcontrollers and may be used in applications that support the 8xC5x architecture. The functional block diagram for the SST89F54/58 is presented in the figure below. The following sections will provide a comprehensive overview of the hardware features presented in the functional block diagram.



FUNCTIONAL BLOCK DIAGRAM

Preliminary

Features

The FlashFlex51 family is a family of embedded controllers which are comprised of the same core features, including an 8-bit MCU (ALU and register set), three Counter/Timer peripherals, and parallel and serial I/O peripherals. The SST89F54/58 support all of the standard features of 8xC5x compatible microcontrollers along with new features. A short overview of the new (non-8xC5x standard) features that the SST89F54/58 support are presented in the New Product Features section. Below is a summary of the features that the SST89F54/58 support.

- **8051 Family Compatible Multi-Purpose 8-bit Microcontroller Unit (MCU) with Embedded SuperFlash Memory for Flexibility**
- **Fully software and development toolset compatibility as well as pin-for-pin compatibile with standard 8xC5x microcontrollers.**
- **256 Bytes register/data RAM**
- **20/36 KByte Embedded High Performance Flexible SuperFlash EEPROM**
  - One 16/32 KByte block (128-Byte sector size)
  - One 4 KByte block (64-Byte sector size)
  - Individual Block Security Lock
  - 87C5x Programmer Compatible
  - Concurrent Operation during In-Application Programming (IAP)
- **Supports External Address Range up to 64 KByte of Program and Data Memory**
- **High Current Drive on Port 1 (5,6,7) pins**
- **Three 16-bit Timer/Counters**
- **Programmable Serial Port (UART)**
- **Six Interrupt Sources at 2 Priority Levels**
- **Selectable Watchdog Timer (WDT)**
- **Four 8-bit I/O Ports (32 I/O Pins)**
- **TTL-and CMOS-Compatible Logic Levels**
- **0 to 33 MHz Operation at 5V ± 10% Supply**
- **0 to 12 MHz Operation at 3V ± 10% Supply**
- **Low Voltage (3V) Operation (0 to 12 MHz)**
- **PDIP-40, PLCC-44 and TQFP-44 Packages**
- **Temperature Ranges:**
  - Commercial (0°C to +70°C)
  - Industrial (-40°C to +85°C)

Product Description

The FlashFlex51 family is a family of embedded microcontroller products designed and manufactured on the state-of-the-art SuperFlash CMOS semiconductor process technology. As a member of the FlashFlex51 controller family, the SST89F54/58 use the same powerful instruction set, has the same architecture, and is pin-for-pin compatible with standard 8xC5x microcontroller devices.

MCU

The SST89F54/58 feature an enhanced 8-bit FlashFlex51 MCU, which is fully software and development toolset compatible. The MCU may operate at a frequency between 0 to 33 MHz with a 5 Volts supply.

Memory Organization

The SST89F54/58 memory architecture is organized around two separate address spaces, which include one area for program memory and one area for data memory (RAM). The SST89F54/58 support up to 64 KByte of program memory with a maximum of 20/36 KByte located on-chip and the remaining memory located externally. The SST89F54/58 support up to 64 KByte of external data memory, in addition to 256 Bytes of on-chip RAM. The SST89F54/58 have 256 x 8 bits of on-chip RAM.

The SST89F54/58 internal program memory consists of Silicon Storage Technology's highly reliable, high density SuperFlash EEPROMs. SST's patented programmable, nonvolatile memory products retain data without applied power and are much more flexible to use than other competing nonvolatile memory solutions. The SST89F54/58 internal program memory consists of two SuperFlash memory blocks, Block 1 and Block 0. Block 1 is the secondary 4 KByte SuperFlash EEPROM. Block 0 is the primary 16/32 KByte SuperFlash EEPROM.

Watchdog Timer

The SST89F54/58 provide an enhanced programmable watchdog timer for fail safe protection against software hang. When a software hang or a hardware based software error occurs, the Watchdog Timer (WDT) unit provides an automatic recovery of the system by means of an internally generated watchdog reset.

Serial I/O

The SST89F54/58 provide a full duplex Serial I/O Port that allows data to be transmitted and received simultaneously in hardware by the transmit and receive registers, respectively, while the software is performing other tasks. The Serial I/O port may operate in four different modes and perform the function of an UART (Universal Asynchronous Receiver/ Transmitter) chip.

Power-Saving Modes

The SST89F54/58 provide two power-saving modes of operation for applications where power consumption is critical. The two power-saving modes are Power Down and Standby (Stop Clock) modes. In the Power Down mode, the oscillator is frozen and the current draw is reduced to approximately 15% of the current drawn when the device is fully active while the supply voltage for the SST89F54/58 can be reduced to a V_{DD} of 2V. Standby mode, like Power Down mode, reduces device current drain to approximately 15 microamperes. It is controlled by hardware (gating on/off the system clock).

Parallel Ports

The SST89F54/58 provide four 8-bit parallel ports, which are accessed by the Port 0 (P0), Port 1 (P1), Port 2 (P2) and Port 3 (P3) special function registers. Each port consists of an output driver, an input buffer and an 8-bit latch, which is located in the port's special function register. The four parallel ports are bi-directional ports and may be used for either input or output. Port 1, Port 2 and Port 3 are multifunctional and can be used for general I/O or for alternate functions. A port pin serving in its alternate function cannot be used for normal I/O, however the alternate functions can be activated, only if the corresponding bit latch in the port special function register contains a "1". The function of the port pins also depends upon whether the system is in the External Host Mode or the In-Application Mode.

Preliminary

Timers/Counters

The SST89F54/58 provide three 16-bit registers that can be used as either timers or event counters. The three Timers/Counters are the Timer 0 (T0), Timer 1 (T1) and Timer 2 (T2) registers. These three registers are located in the SFR as pairs of 8-bit registers. The low byte of the T0 register is stored in the Timer 0 LSB (TL0) special function register and the high byte of the T0 register is stored in the Timer 0 MSB (TH0) special function register. The low byte of the T1 register is stored in the Timer 1 LSB (TL1) special function register and the high byte of the T1 register is stored in the Timer 1 MSB (TH1) special function register. The low byte of the T2 register is stored in the Timer 2 LSB (TL2) special function register and the high byte of the T2 register is stored in the Timer 2 MSB (TH2) special function register. The T0, T1 and T2 registers are alternate functions of port pins.

Interrupts

The SST89F54/58 provide 6 interrupt sources, which include two external interrupts (INT0# and INT1#), three Timer/Counter interrupts (TF0, TF1 and TF2), and one from the serial port (SI or TI). The Interrupt Enable (IE) special function register is the source of the interrupts. Each of the bits that generate the interrupts may be set or cleared by software with the same result as setting or clearing the bits through hardware. Therefore, interrupts may be generated or canceled by software. Individual interrupts can be enabled or disabled by setting or clearing individual bits of the IE register. The SST89F54/58 also contain a global enable bit which allows all of the interrupts to be enabled or disabled by setting or clearing the EA bit of the IE register.

New Product Features (SST Uniqueness)

SuperFlash Technology

The SST89F54/58 are members of the FlashFlex51 family of embedded 8-bit microcontrollers. The FlashFlex51 family is a family of embedded microcontroller products designed and manufactured on the state-of-the-art SuperFlash CMOS semiconductor process technology. As a member of the FlashFlex51 controller family, the SST89F54/58 use the same powerful instruction set, have the same architecture, and are pin-for-pin compatible with standard 8xC5x microcontroller devices.

SuperFlash Organization

SST89F54/58 come with 20/36 KByte of integrated on-chip flash EEPROM program memory, using the patented and proprietary SST CMOS SuperFlash EEPROM technology and the SST field enhancing tunneling injector split-gate memory cells. The Super Flash memory is partitioned into 2 independent program memory blocks. The primary 16/32 KByte SuperFlash memory occupies the standard 8xC5x's 16/32 KByte of internal ROM space and the secondary 4 KByte SuperFlash block occupies the upper most of the 64 KByte address space for the 8xC5x architecture.

SuperFlash Programmability

Typical 8xC5x microcontrollers, while referencing program storage, can only read instructions, even though the physical device used for external program storage can read/program, it is effectively read-only memory. This is due to the fact that the ROM is masked at the factory and once the ROM chips are masked, the program cannot be changed. However, SST solves this user restriction by implementing flash memory blocks that can be programmed via a standard 87C5x EPROM programmer or a standard flash EEPROM memory programmer fitted with a special adapter and firmware for SST89F54/58 devices. During power-on reset, the SST89F54/58 can be configured as a master for In-Application Programming (IAP) operation or as a slave to an external host. The SST89F54/58 are designed to be programmed "in-place" and "in-operation" on the printed circuit board assembly for maximum flexibility.

Security Lock

The SST89F54/58 provide a security lock mechanism, that prevents program code corruption resulting from accidental sector or block erasing/programming erroneous program code to the internal flash memory blocks. The security lock also prevents software piracy by disabling the read access of the internal flash memory contents. When the security lock is activated, the MOVC instructions executed from external program memory or flash memory are disabled from fetching code bytes from unlocked memory blocks.

SST89F54/58 Unique Special Function Registers

The SST89F54/58 not only provide the standard 8xC5x SFRs, but it also implements seven new registers, which include the SFDT, SFAL, SFAH, SFCM, SFCF, WDTD and WDTC registers. The SFDT, SFAL, SFAH, SFCM and SFCF support the SuperFlash EEPROM memory and the WDTC and WDTD registers support the Watchdog Timer.

MCU Timing

Machine Cycle

The SST89F54/58 machine cycle consists of six states, which are divided into two phases that are one clock period for each phase. Therefore, the SST89F54/58 machine cycle is 12 clock periods. Operating the SST89F54/58 with a low voltage, (3 Volts supply) limits the MCU to a frequency between 0 to 12 MHz and a machine cycle limited to a maximum of 1 ms.

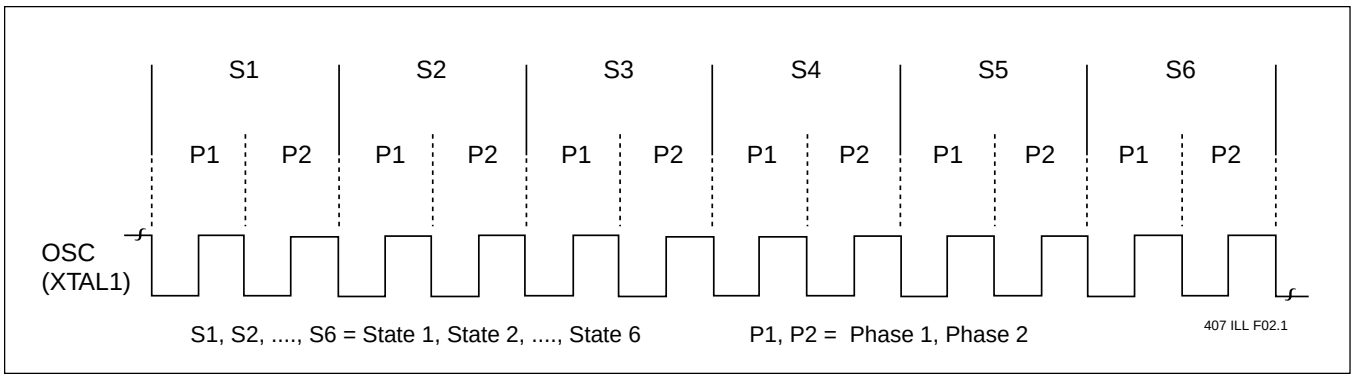


FIGURE 1: MACHINE CYCLE WAVEFORM

Timing Parameters

The timing characteristics for the PSEN#, ALE, RD# and WR# signals are presented in the AC Characteristics Table in the Appendix section. External Program Memory Read, External Data Memory Read, and External Data Memory Write Cycle Waveforms are also presented in the Appendix section.

Note that the timings published in the AC Characteristics Table of the Appendix section includes the propagation delays for the given testing specifications. The parameter values are subject to change without notice.

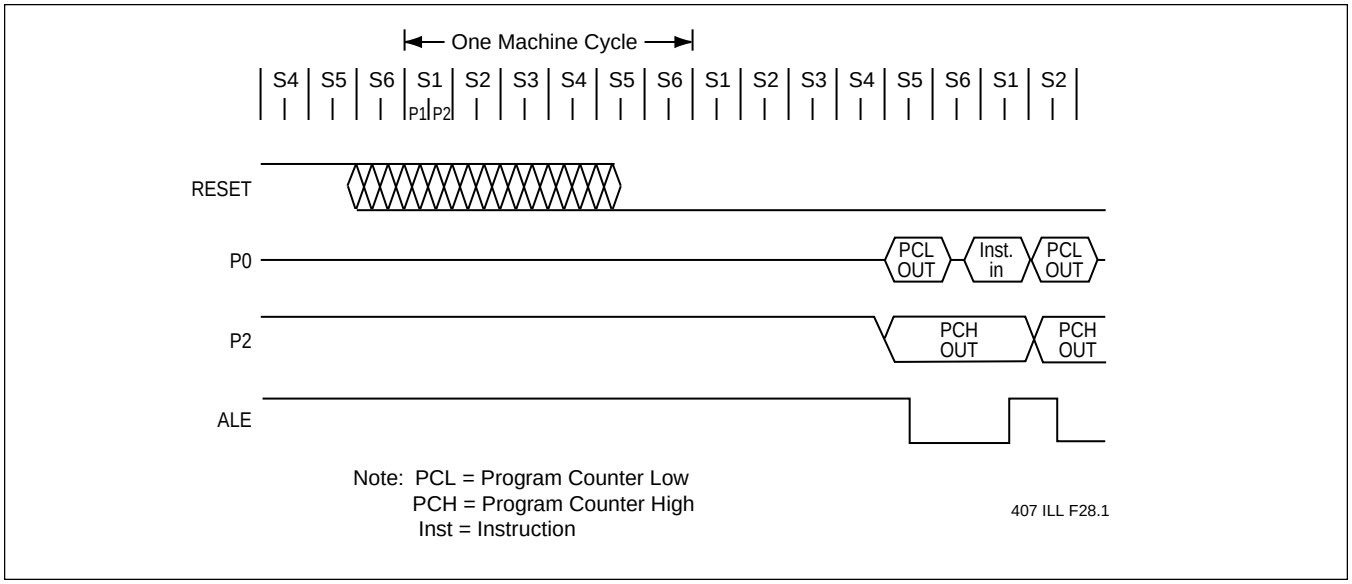


FIGURE 2: RESET WAVEFORM

Reset

A system reset initializes the MCU and begins program execution at program memory location 0000h. The reset input for the SST89F54/58 is the RST pin. In order to reset the SST89F54/58, a logic level high must be applied to the RST pin for at least two machine cycles (24 clocks), after the oscillator becomes stable. ALE, PSEN# are weakly pulled high during reset. During reset, ALE and PSEN output a high level in order to perform correct reset. This level must not be affected by external element. A system reset will not affect the 256 Bytes of on-chip RAM while the SST89F54/58 is running, however, the contents of the on-chip RAM during power up are indeterminate. All Special Function Registers (SFR) return to their reset values, which are outlined in the Special Function Registers section. The Machine Cycle Waveform figure below presents the reset timing waveform

After a successful reset is completed, if the PSEN# pin is driven by an input force with a high-to-low transition while the RST input pin is continually held high, the device will enter the External Host mode for the internal flash memory programming operation. Otherwise the device will enter the IAP mode.

Power-On Reset

At initial power up, the port pins will be in a random state until the oscillator has started and the internal reset algorithm has written one's to all the pins. Powering up the device without a valid reset could cause the CPU to start executing instructions from an indeterminate location. Such undefined states may inadvertently corrupt the code in the flash.

To ensure a good power-on reset, it is required that the V_{DD} rise time does not exceed 1 ms and the oscillator start up time does not exceed 10 ms. The Minimum V_{DD} to RST for Power-On Reset figure below shows the maximum delay time allowed between initial power up and reset. RST should lag no more than 10 ns behind V_{DD} at voltages above 1.4 V. A common method to extend the RST signal is to implement a RC circuit by connecting the RST pin to V_{CC} through a 10 µF capacitor and to V_{SS} through an 8.2K resistor as shown in the Power-On Reset circuit in Figure 4. This method maintains the necessary relationship between V_{DD} and RST to avoid programming at an indeterminate location which may cause code corruption in the flash.

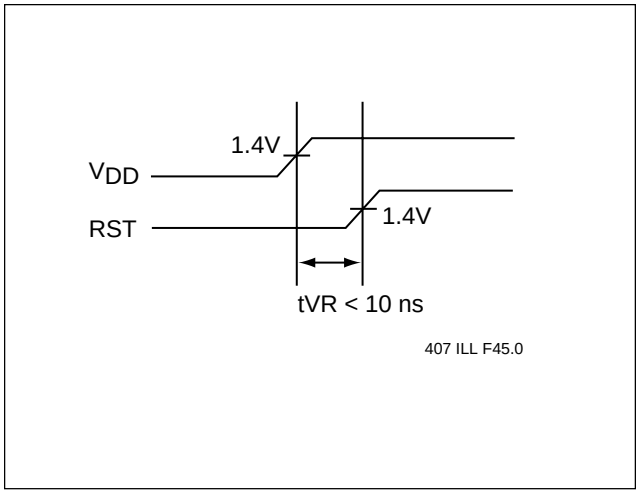


FIGURE 3: MINIMUM V_{DD} TO RST FOR POWER-ON RESET

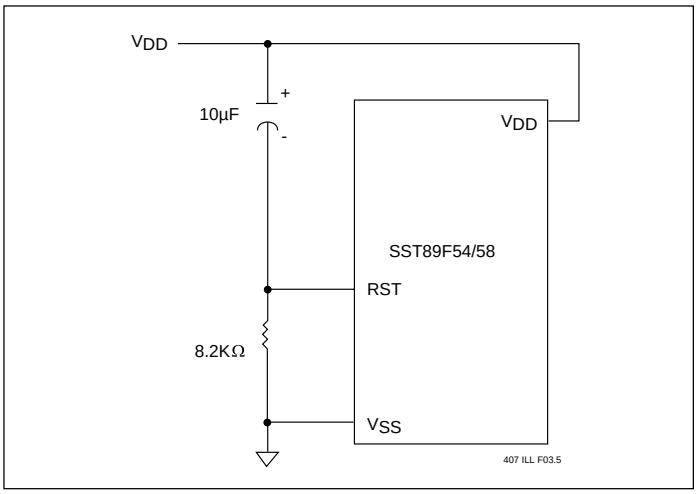


FIGURE 4: SAMPLE RESET CIRCUIT

Architecture

The MCU in the SST89F54/58 family is an enhanced 8-bit FlashFlex51 MCU, which is fully software and development toolset compatible as well as pin-for-pin compatible to standard 8xC5x microcontrollers. The MCU is composed of three sections, which consist of an arithmetic logic unit (ALU), a timing and a Timing and Control section, and Registers section.

ALU

Data byte manipulation is controlled by the ALU. The ALU performs all addition, subtraction, multiplication, division and logic operations such as logical AND and OR. The ALU section is internal to the microcontroller MCU and is not under the programmer's direct control.

Timing and Control

The Timing and Control section is responsible for synchronizing the flow of data into and out of the MCU. It coordinates the movement of data on the buses both internal and external to the microcontroller. The PSEN#, RD# and WR# signals are generated by the Timing and Control section. This section is internal to the microcontroller MCU and is not under the programmer's direct control. However, the operating frequency of the MCU may be changed by using an external oscillator.

Registers

The Registers section is composed of 8-bit latches which are used to hold and manipulate data. The register section of the SST89F54/58 are made up of two register groups, which include the 8-bit special function registers and the eight general-purpose 8-bit register banks. The only section of the MCU that is directly controllable by the programmer is the Registers section.

The Special Function Registers (SFR) are standard components of 8xC5x compatible microcontrollers, which perform specific, required tasks. Every normal processor register, except the eight general-purpose register banks, and special function corresponds to a special function register. Processor registers such as the accumulator (ACC), the program status word register (PSW), the data pointer to external memory (DPL and DPH) and the stack pointer (SP) are controlled by special function registers. Also, all internal features such as the Watchdog Timer, Power Mode Management, Timers/Counters, Parallel Ports, Serial Port, Security Lock and SuperFlash memory functions, are controlled by special function registers. The special function registers function much like accumulators, and programming them is simple and convenient.

The 8 general-purpose 8-bit register banks are called R0 through R7 and are grouped into four banks of eight bytes each. The general-purpose registers provide scratchpad RAM for temporary values.

Memory Organization

The SST89F54/58 memory architecture is organized around two separate address spaces, which include one area for program memory (ROM) and one area for data memory (RAM). The SST89F54/58 support up to 64 KByte of program memory with a maximum of 20/36 KByte located on-chip and the remaining memory located externally. The SST89F54/58 support up to 64 KByte of external data memory. The SST89F54/58 have 256 x 8 bits of on-chip RAM.

The Timing and Control (T/C) bus is used in order to distinguish between the overlapping program and data memory spaces. The control lines used to access external data memory are the RD# and WR# lines. The RD# line is active for an external data memory read and the WR# line is active for an external data memory write. The control line used to read program memory is the Program Store Enable (PSEN#) line. Note that there is no equivalent WR# line for program memory since there are no instructions that permit programming to program memory. Internal data memory is accessed through indirect addressing techniques.

Program Memory

The SST89F54/58 program memory structure is unique in that it provides two different options for memory organization. The figures below present the two options for program memory organization for the SST89F54 and SST89F58.

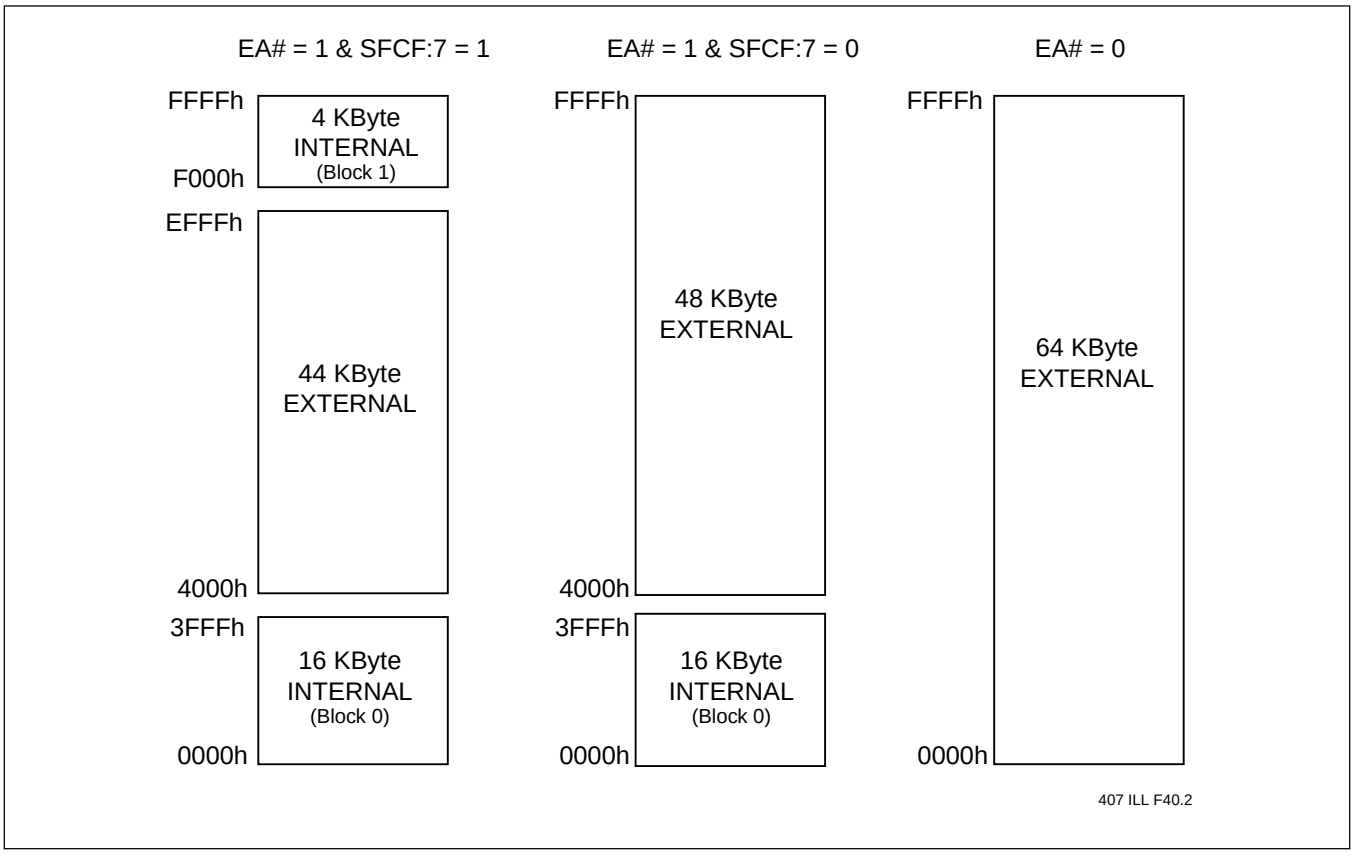


FIGURE 5A: SST89F54 PROGRAM MEMORY ORGANIZATION

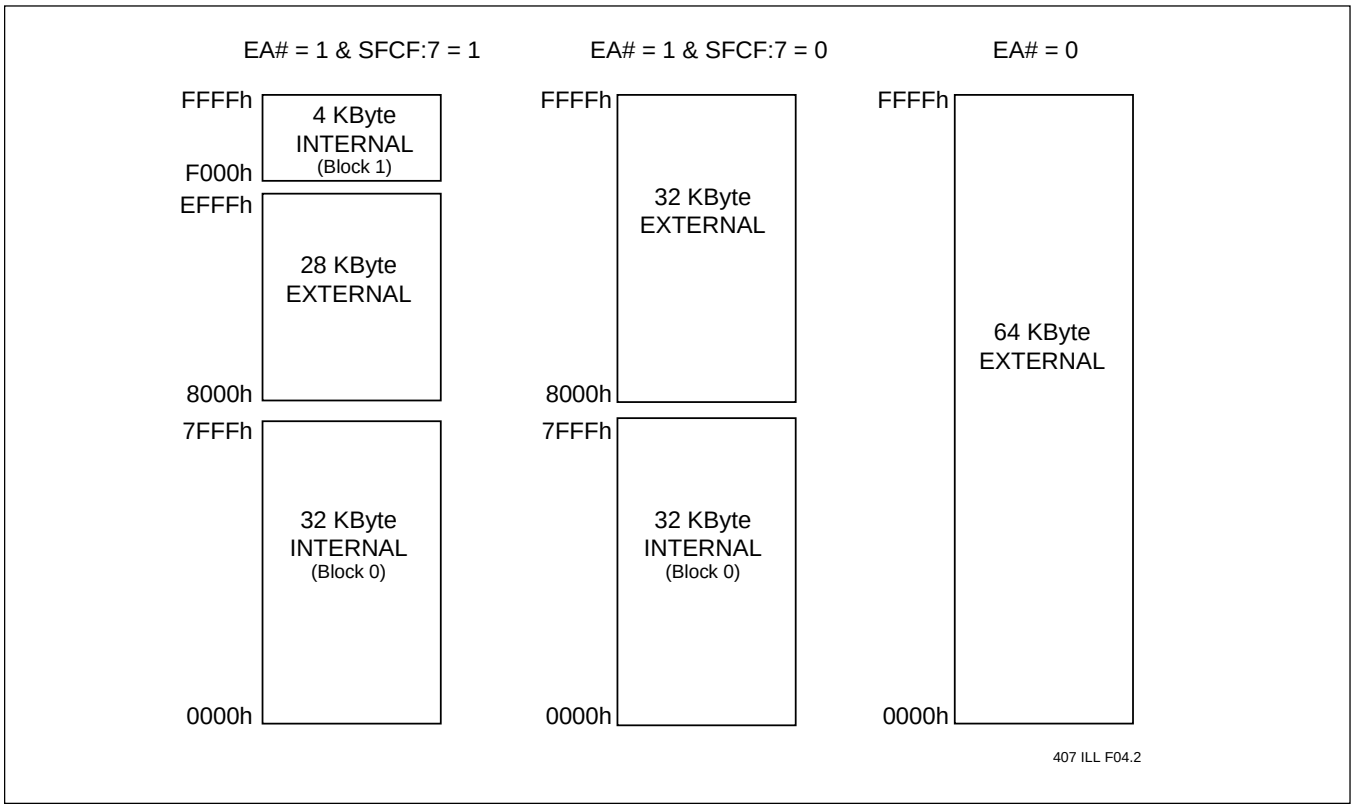


FIGURE 5B: SST89F58 PROGRAM MEMORY ORGANIZATION

As demonstrated in the figure, the Secondary Flash Block Visibility (VIS) bit of the SuperFlash Configuration (SFCF) register and the External Access (EA#) signal produced by the EA# pin of the bus controller determine the program memory organization. The EA# pin must be strapped to Vss in order to enable the SST89F54/58 to fetch code from External Program Memory locations starting at 0000h to FFFFh. Note, however, that if either of the Lock bits (SFCF[6:5]) of the SFCF register are read, the logic level at the EA# pin will be internally latched during reset. The various combinations of the Lock bits and the results are presented in the table below.

SFCF.6	SFCF.5	Security Lock Decoding
1	1	Flash Block 0 and Block 1 are locked
1	0	Both locked but they are accessible only through In-Application Programming
0	1	Flash Block 1 is locked
0	0	No Flash Blocks are locked

The EA# pin must be strapped to V_{DD} in order to enable the SST89F54/58 to fetch code from internal program memory. The VIS bit serves as an Secondary Flash Block Visibility selector for the 4KByte block of internal flash program memory. The VIS bit must be set in order for the 4 KByte block of internal flash program memory to be visible at the top of the secondary 64 KByte address range. The VIS bit must be cleared in order for the 4KByte block of internal flash memory to not be visible.

When the EA# line is inactive, the program memory for the SST89F54 is composed of two memory blocks. The primary memory block is 16 KByte of flash memory and occupies the address space 0000h to 3FFFh. The 48 KByte of external memory occupies the address space 4000h to FFFFh. When the 4 KByte flash memory is visible, it resides in the secondary memory block and occupies the address space F000h to FFFFh. The 16KByte primary SuperFlash block are organized as 128 sectors with sector address from A15 to A7. Each sector contains 2 rows with row address from A15 to A6. Each row has 64 Bytes with byte address from A5 to A0.

When the EA# line is inactive, the program memory for the SST89F58 is composed of two memory blocks. The primary memory block is 32 KByte of flash memory and occupies the address space 0000h to 7FFFh. The 32 KByte of external memory occupies the address space 8000h to FFFFh. When the 4 KByte flash memory is visible, it resides in the secondary 4KByte memory block and occupies the address space F000h to FFFFh. The 32 KByte primary SuperFlash block are organized as 256 sectors with sector address from A15 to A7. Each sector contains 2 rows with row address from A15 to A6. Each row has 64 bytes with byte address from A5 to A0.

When internal code operation is enabled (EA# = 1), the secondary 4 KByte flash memory block is selectively visible for code fetching. The secondary block is always accessible through the SuperFlash mailbox registers: SFCM, SFCF, SFAL, SFAH and SFDT. When bit 7 of the SuperFlash Configuration/Status mailbox register (SFCF:7), SFR address location F7h, is set, the secondary 4 KByte block will be visible by program counter. The 4K x 8 secondary SuperFlash block are organized as 64 sectors with sector address from A15 to A6. Each sector contains 2 rows with row address from A15 to A5. Each row contains 32 bytes with byte address from A4 to A0.

Note that after reset, the MCU begins fetching instructions starting at address 0000h. The physical location of address 0000h is either on-chip or external, depending on the value of the EA# line. If the EA# line is a logic low, address 0000h and all other program storage addresses will reference external program memory. If the EA# line is a logic high, address 0000h will reference on-chip program memory. Also note that each interrupt service routine is associated with an interrupt vector address located in the interrupt vector table, which starts at 0003h. The physical location of address 0003h will depend on the logic level of the EA# line as well.

Preliminary

Data Memory

The SST89F54/58 data memory structure consists of 256 Bytes of on-chip RAM and can address up to 64 KByte of external data memory. Note that the 256 Bytes of on-chip RAM is not affected by a system reset while the SST89F54/58 is running, however, the contents of the on-chip RAM during power up is indeterminate.

The 4 KByte of small sectors (64 Bytes per sector) on chip flash memory to be mapped either as program memory or data memory. When the 4KByte flash is mapped as data memory (SFCF:7=0) it is accessible via the mailbox registers.

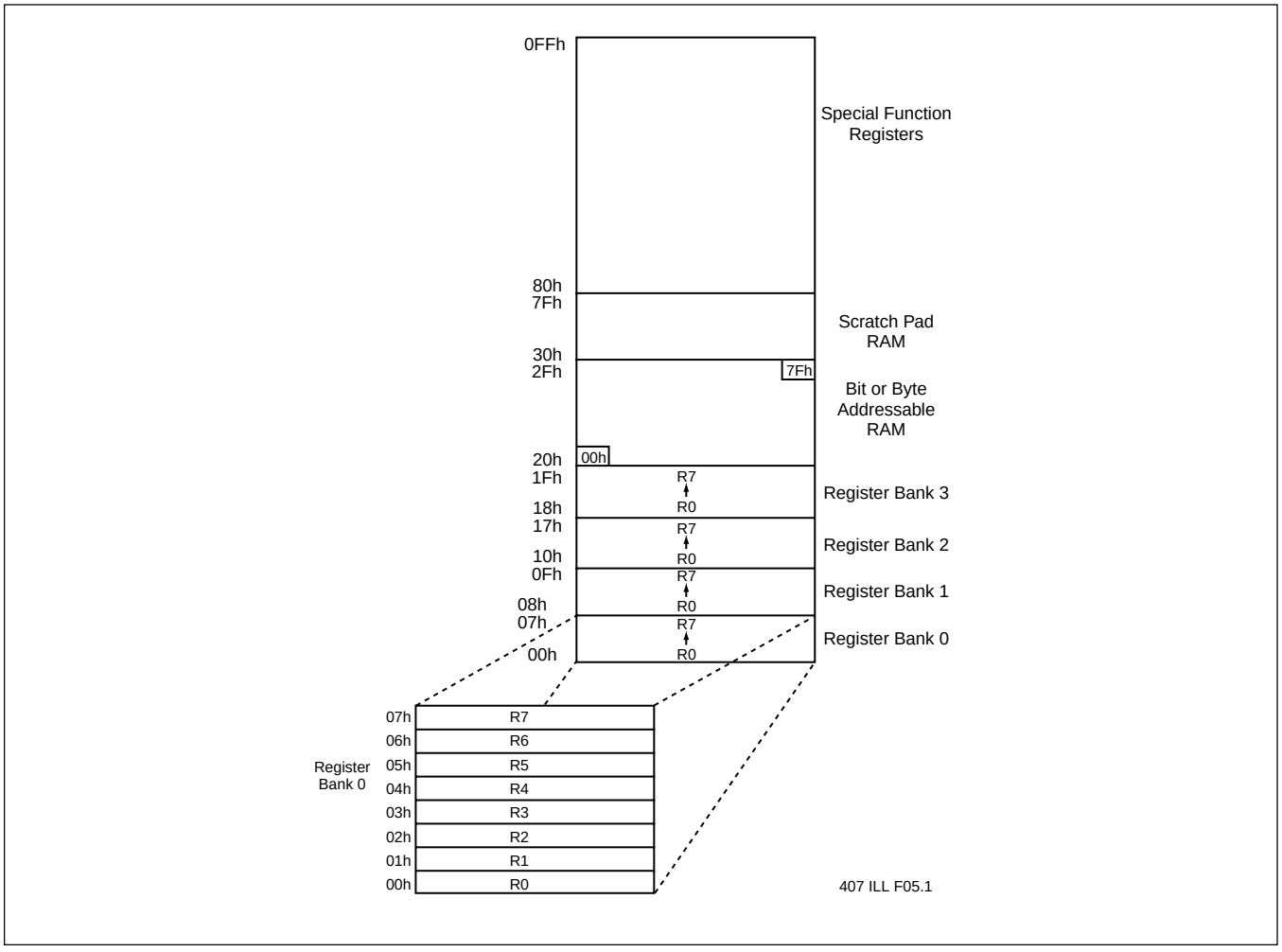


FIGURE 6: INTERNAL DATA MEMORY (256 BYTES RAM)

The secondary 128 Bytes of the 256 Bytes of on-chip RAM contain the Special Function Registers. The Special Function Registers reside at memory locations 80h through FFh in the on-chip RAM. More information about the Special Function Registers is presented in the Special Function Registers section.

The lowest 32 Bytes of the lower 128 Bytes are grouped into four banks of eight one-byte registers. The four banks reside at memory locations 00h through 1Fh in the on-chip RAM. The four banks are simply groups of eight bytes that may be used for general-purpose storage, however, only one bank is activated at a time. The active bank is selected by a combination of the RS1 and RS0 bits in the PSW register as outlined in the table below. Note that RS1 and RS0 can be set by program instructions.

REGISTER BANK SELECT TABLE

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00h-07h
0	1	1	08h-0Fh
1	0	2	10h-17h
1	1	3	18h-1Fh

The eight registers in the active bank are byte addressable. They can be read or written to by the program and may be referred to as R0 through R7. **Note:** after reset, the stack pointer (SP) is pointing to the top register of the lowest bank (address 07h), causing the stack to start at 8h. In order to avoid writing stack data into the registers, it is required that a new address is loaded into the SP before any CALL or PUSH instructions are used.

The next 16 Bytes of the lower 128 Bytes form a block that can be addressed as either bytes or as 128 individual bits. The block of bit or byte addressable RAM resides at memory locations 20h through 2Fh in the on-chip RAM. The byte addresses are 20h to 2Fh. The bit addresses are 00h to 7Fh. The format of the instruction determines whether the reference is to a byte or a bit.

The remaining 80 bytes of the lower 128 Bytes form a block that can be used as scratch pad RAM. The scratch pad RAM resides at memory locations 30h through 7Fh in the on-chip RAM.

The external data memory consists of up to 64 KByte of memory and resides at memory locations 0000h through FFFFh. The external data memory can be accessed by using indirect addressing techniques. The 16-bit Data Pointer (DPTR) register is used to store the address of the external memory location. The low byte of the DPTR register is stored in the Data Pointer Low (DPL) register and the high byte of the DPTR register is stored in the Data Pointer High (DPH) register.

Preliminary

External Memory Interface

The main function of the external bus interface is to connect the external memory to the MCU. The external bus also facilitates the expansion of the external data memory. The SST89F54/58 memory architecture supports up to 64 Kbyte of external data memory. The external data memory resides at memory locations 0000h through FFFFh can be accessed by using indirect addressing techniques. The 16-bit Data Pointer (DPTR) register is used to store the address of the external memory location. The low byte of the DPTR register is stored in the Data Pointer Low (DPL) register and the high byte of the DPTR register is stored in the Data Pointer High (DPH) register.

Accessing External Memory

The SST89F54/58 memory architecture is organized around two separate address spaces, which include one area for program memory (ROM) and one area for data memory (RAM). The SST89F54/58 support up to 64 KByte of program memory with a maximum of 20/36 KByte located on-chip and the remaining memory located externally. The SST89F54/58 support up to 64 KByte of external data memory, in addition to 256 Bytes of on-chip RAM.

The Timing and Control (T/C) bus is used in order to distinguish between the overlapping program and data memory spaces. External Program Memory is accessed when the EA# signal is active or the program counter (PC) contains an address that is larger than FFFFh. The control lines used to access external data memory are the RD# and WR# lines. The RD# line is an alternate function of the Port 3 bit P3.7 and the WR# line is an alternate function of the Port 3 bit P3.6. The RD# line is active for an external data memory read and the WR# line is active for an external data memory write. The control line used to read program memory is the Program Store Enable (PSEN#) line. Note that there is no equivalent WR# line for program memory since there are no instructions that permit programming to program memory. Internal data memory is accessed through indirect addressing techniques.

Instructions which access external program memory always use a 16-bit address. Instructions which access external data memory may use either an 8-bit (MOVX@RI) or a 16-bit (MOVX@DPTR) address, depending on the instruction being executed. An instruction that uses a 16-bit address outputs the high byte of the address to Port 2, where it is held for the duration of the read or write cycle. During this time the Port 2 latch (P2) does not have to contain 1's and the contents of Port 2 are not lost. If the external memory cycle is not immediately followed by another external memory read or write cycle, the prior contents of the Port 2 latches are restored after the memory access cycle. Note that if an 8-bit address is being used, the contents of the Port 2 latches are unchanged and remain at the Port 2 pins throughout the external memory cycle.

Instructions which access external program or data memory output the low byte of the address to Port 0. During this time, the Port 0 pins are connected to an internal active pull-up and they do not float. The ALE signal is used to capture the address byte into an external latch. The address byte is valid at the negative transition of the ALE signal. For a write instruction, the data byte to be written is sent to Port 0 just before the WR# signal is activated and remains there until after the WR# signal is deactivated. For a read instruction, the data byte to be read is read from Port 0 just before the read strobe is deactivated. During any access to external memory, the MCU writes 0FFh to Port 0, therefore, the prior contents of the Port 0 latches are lost.

Reset Operation of the External Memory

A system reset initializes the MCU and begins program execution at program memory location 0000h. The physical location of address 0000h is either on-chip or external, depending on the value of the EA# line. If the EA# line is a logic low, address 0000h and all other program storage addresses will reference external program memory. If the EA# line is a logic high, address 0000h will reference on-chip program memory.

The reset input for the SST89F54/58 is the RST pin. In order to reset the SST89F54/58, a logic level high must by applied to the RST pin for at least two machine cycles, while the oscillator is running. The reset signal is asynchronous to the system clock, however, the RST pin is sampled during the second phase of the fifth state of every machine cycle. A power-on reset also requires a logic level high to by applied to the RST pin for at least two machine cycles, providing that the V_{DD} rise time does not exceed a millisecond and the oscillator start up time does not exceed 10 milliseconds.

After a successful reset is completed, if the PSEN# pin is driven by an input force with a high-to-low transition while the RST input pin is continually held high, the device will enter the External Host mode for the internal flash memory programming operation, otherwise the device will enter normal operation. In normal operation, the content of the external RAM is not affected by a reset, as long as the power supply is not turned off. However, after a power-on reset, the content of the external RAM is undefined. If the reset occurs during a write instruction to the external RAM, the content of the external RAM memory location depends on the machine cycle in which the active reset signal is detected. In the case of the 2-cycle MOVX instruction, the external RAM will not be affected if the instruction is in its first cycle, however it is overwritten if the instruction is in its second cycle. Note that a system reset will not affect the on-chip RAM while the SST89F54/58 are running, however, the contents of the on-chip RAM during power up are indeterminate.

Special Function Registers

The Special Function Registers (SFRs) are standard components of 8xC5x compatible microcontrollers which perform specific, required tasks. The registers are eight bits each and are located in internal data memory at memory locations 80h - FFh in the secondary 128 Bytes of the on-chip 256 Bytes of RAM and may be addressed by program instructions. Some of the SFR locations are bit addressable as well as byte addressable. The addresses in the SFR map which are not occupied are reserved for use in future versions of the SST89F54/58 and should not be used in programs for the current version. The result of reading or writing to these registers is indeterminate. User programs should not write 1's to these addresses, since they may be used in future products to invoke new features.

The SST89F54/58 not only provide the standard 8xC5x SFRs, but it also implement seven new registers, which include the SFDT, SFAL, SFAH, SFCM, SFCF, WDTC and WDTD registers. Most of the unique features of the SST89F54/58 microcontroller family, such as the SuperFlash EEPROM memory are controlled by bits in these additional SFRs.

The SFRs for the SST89F54/58 are organized in the memory map shown below. Individual descriptions of each SFR and its eight bits are presented in the following sections, including the contents of each register after reset. Note that the individual bits of each register may be accessed individually by designating the name of the register followed by a decimal point followed by the number of the bit position within the register.

8 BYTES

F8	SFDT	SFAL	SFAH	SFCM					FF
F0	B*							SFCF	F7
E8									EF
E0	ACC*								E7
D8									DF
D0	PSW*								D7
C8	T2CON*		RCAP2L	RCAP2H	TL2	TH2			CF
C0	WDTC								C7
B8	IP*								BF
B0	P3*								B7
A8	IE*								AF
A0	P2*								A7
98	SCON*	SBUF							9F
90	P1*								97
88	TCON*	TMOD	TL0	TL1	TH0	TH1			8F
80	P0*	SP	DPL	DPH			WDTD	PCON	87

FlashFlex51 SFR Memory Map

* = BIT ADDRESSABLE
All addresses are hexadecimal

407 ILL F43.4

For the following SFR descriptions, "r/w" under the corresponding bit means read or write, while "r" means read only, "r/c" means read or clear, "r/s" means read or set.

Accumulator (ACC)

The Accumulator register serves as the accumulator for arithmetic operations. The accumulator is also used in some instructions as an index register. When referring to the accumulator as a location in the SFR, the mnemonic ACC is used, however accumulator-specific instructions refer to the accumulator as A. The following table provides a bit description of the Accumulator register.

Accumulator (ACC)

Location	7	6	5	4	3	2	1	0	Reset Value
E0h	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

B Register (B)

The B register serves as a second accumulator for certain arithmetic operations. Specifically, the B register functions as a divisor in the DIV AB instruction and as a multiplicand in the MUL AB instruction. B register-specific instructions refer to the B register as B. The B register is also used in some instructions as a general-purpose scratchpad register. The following table provides a bit description of the B register.

B Register (B)

Location	7	6	5	4	3	2	1	0	Reset Value
F0h	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Program Status Word (PSW)

The Program Status Word register serves as a status flag. The following table provides a bit description of the PSW register.

Program Status Word (PSW)

Location	7	6	5	4	3	2	1	0	Reset Value
D0h	CY	AC	F0	RS1	RS0	OV	F1	PARITY	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

The PSW contains the common status flags such as the CY, AC, OV and PARITY flags.

Carry Flag bit (CY) is set to 1 if the last arithmetic operation resulted in a carry into (during addition) or a borrow (during subtraction) from the high order byte. All other arithmetic operations clear the CY flag to 0.

Auxiliary Carry Flag bit (AC) is set to 1 if the last arithmetic operation resulted in a carry into (during addition), or a borrow (during subtraction) from the high order nibble. All other arithmetic operations clear the AC flag to 0.

Overflow Flag bit (OV) is set to 1 if the last arithmetic operation resulted in a carry (during addition), borrow (during subtraction), or overflow (during multiply or divide). All other arithmetic operations clear the OV flag to 0.

Parity Flag bit (PARITY) is set to 1 if the modulo-2 sum of the eight bits of the ACC register is 1 (odd-parity). If the modulo-2 sum of the eight bits of the ACC register is 0 (even-parity) then the PARITY flag is cleared to 0.

In addition to the usual status flags, the SST89F54/58 implement four control flags, which include the F0, F1, RS0 and RS1 flags and are not associated with any specific MCU state or function.

F0 and F1 flag bits are bit-addressable, general-purpose flags used for software control.

RS0 and RS1

In addition to the SFRs, the SST89F54/58 data memory supports a block of specialized internal data RAM that is grouped into four banks of eight one byte registers each. These registers are located in internal data memory at memory locations 00h-1Fh in the lowest 32 Bytes of the lower 128 Bytes of the on-chip 256 Bytes of RAM. Only one bank is activated at a time. The active bank is selected by a combination of the RS1 and RS0 bits in the PSW register as outlined in the table below. Note that RS1 and RS0 can be set by program instructions.

REGISTER BANK SELECT TABLE

RS1	RS0	REGISTER BANK	ADDRESS
0	0	0	00h-07h
0	1	1	08h-0Fh
1	0	2	10h-17h
1	1	3	18h-1Fh

The eight registers in the active bank are byte addressable. Note that after reset, the stack pointer (SP) is pointing to the top register of the lowest bank (address 07h), causing the stack to start at 8h. In order to avoid writing stack data into the registers, it is required that a new address be loaded into the SP before any CALL or PUSH instructions are used.

Stack Pointer (SP)

The SP register points to the location in data memory that becomes the top of the stack after executing PUSH, POP, RET and CALL instructions. Since the stack pointer is one byte wide, the stack may address a maximum of 256 Bytes. The stack grows upward through memory, therefore, the SP is incremented before the data is stored as a result of a PUSH or CALL instruction. The stack may reside anywhere in the on-chip 256 Bytes of RAM by loading the desired starting address into the SP. Note that after reset, the stack pointer (SP) is pointing to the top register of the lowest bank (address 07h), causing the stack to start at 8h. In order to avoid writing stack data into the registers, it is required that a new address be loaded into the SP before any CALL or PUSH instructions are used. The following table provides a bit description of the SP register.

Stack Pointer (SP)

Location	7	6	5	4	3	2	1	0	Reset Value
81h	SP[7:0]								07h
	r/w								

Data Pointer (DPTR)

The DPTR is a 16-bit data pointer, consisting of two 8-bit SFRs. The Data Pointer Low (DPL) register is the low byte of the FlashFlex51 DPTR and the Data Pointer High (DPH) register is the high byte of the FlashFlex51 DPTR. The DPTR is used to point to non-scratchpad data RAM and may be used to hold a 16-bit address for certain instructions. The DPTR can be used as a single 16-bit register or as two 8-bit registers. The following tables provide a bit description of the DPL and DPH registers.

Data Pointer Low 0 (DPL)

Location	7	6	5	4	3	2	1	0	Reset Value
82h	DPL[7:0]								00h
	r/w								

Data Pointer High 0 (DPH)

Location	7	6	5	4	3	2	1	0	Reset Value
83h	DPH[7:0]								00h
	r/w								

Interrupt Enable Register (IE)

Individual interrupts can be enabled or disabled by setting or clearing individual bits of the Interrupt Enable (IE) register. The IE register also contains a global enable bit which allows enabling or disabling all of the interrupts by setting or clearing the EA bit.

The following table provides a bit description of the IE register.

– Not implemented: Reserved for future use.*

Enable All (EA): Disables all interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.

IE.6: This bit is reserved for future use.

Timer 2 Enable (ET2): Enable or disable the Timer 2 overflow or capture interrupt.

Serial Port Enable (ES): Enable or disable the serial port interrupt.

Timer 1 Enable (ET1): Enable or disable the Timer 1 overflow interrupt.

External Interrupt 1 Enable (EX1): Enable or disable External Interrupt 1.

Timer 0 Enable (ET0): Enable or disable the Timer 0 overflow interrupt.

External Interrupt 0 Enable (EX0): Enable or disable External Interrupt 0.

Interrupt Enable (IE)

Location	7	6	5	4	3	2	1	0	Reset Value
A8h	EA	–	ET2	ES	ET1	EX1	ET0	EX0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Interrupt Priority Register (IP)

Priority levels for Timer/Counters, Serial Port and external interrupts may be set by setting or clearing individual bits of the Interrupt Priority (IP) register. A 0 value is designated a low priority and a 1 value is the high priority.

– Not implemented: Reserved for future use.*

IP.7: This bit is reserved for future use.

IP.6: This bit is reserved for future use.

Timer 2 Priority (PT2): Defines the Timer 2 interrupt priority level.

Serial Port Priority (PS): Defines the Serial Port interrupt priority level.

Timer 1 Priority (PT1): Defines the Timer 1 interrupt priority level.

External Interrupt Priority (PX1): Defines the External Interrupt 1 priority level.

Timer 0 Priority (PT0): Defines the Timer 0 interrupt priority level.

External Interrupt Priority (PX0): Defines the External Interrupt 0 priority level.

Interrupt Priority Register (IP)

Location	7	6	5	4	3	2	1	0	Reset Value
B8h	–	–	PT2	PS	PT1	PX1	PT0	PX0	xx000000
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Preliminary

Port Latches (P0, P1, P2, P3)

The 32 I/O pins are organized into four 8-bit registers designated as P0 - P3. Each port has an associated 8-bit latch, the outputs of which drive the matching I/O pins. The contents of the latches can be read from or written to in the P0, P1, P2 and P3 registers.

Port 0 (P0) functions as a time-multiplexed address/data bus during external memory accesses, and as a general purpose I/O port on devices which incorporate internal program memory. During external memory accesses, this port will contain the least-significant byte (LSB) of the address when the ALE signal is HIGH and the data bus when the ALE signal is LOW. When used as an I/O port, the port bits are open-drain and require pull-up resistors. Writing a logical 1 to any bit of the port will place it in a high impedance mode, which is necessary if the pin is to be used as an input. Pull-ups are not required when accessing external memory.

Port 0 (P0)

Location	7	6	5	4	3	2	1	0	Reset Value
80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Port 1 (P1) functions as a general purpose I/O port. In addition, two of the pins have alternative functions, which are controlled by several other SFRs. The alternative functions of the P1 bits are described below. Note that the corresponding P1 latch bit must contain a logic one before the pin can be used in its alternative function capacity.

Port 1 (P1)

Location	7	6	5	4	3	2	1	0	Reset Value
90h	-	-	-	-	-	-	T2EX	T2	FFh
							r/w	r/w	

– Not implemented: Reserved for future use.*

Timer 2 Capture/Reload Trigger (T2EX): A 1 to 0 transition on this pin will cause the value in the T2 registers to be transferred into the capture registers if enabled by EXEN2 (T2CON.3). When in auto-reload mode, a 1 to 0 transition on this pin will reload the timer 2 registers with the value in RCAP2L and RCAP2H if enabled by EXEN2 (T2CON.3).

Timer 2 External Input (T2): A 1 to 0 transition on this pin will cause timer 2 to increment or decrement depending on the timer configuration.

Port 2 (P2) functions as an address bus during external memory accesses, and as a general purpose I/O port on devices which incorporate internal program memory. During external memory cycles, this port will contain the MSB of the address.

Port 2 (P2)

Location	7	6	5	4	3	2	1	0	Reset Value
A0h	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Port 3 (P3) functions as a general purpose I/O port. In addition, all the pins have an alternative function, which is controlled by several other SFRs. The alternative functions of the P3 bits are described below. Note that the corresponding P3 latch bit must contain a logic one before the pin can be used in its alternative function capacity.

Port 3 (P3)

Location	7	6	5	4	3	2	1	0	Reset Value
B0h	RD#	WR#	T1	T0	INT1#	INT0#	TXD0	RXD0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

External Data Memory Read Strobe (RD#): provides an active low read strobe to an external memory device.

External Data Memory Write Strobe (WR#): provides an active low write strobe to an external memory device.

Timer/Counter 1 External Input (T1): a 1 to 0 transition on this pin will increment timer 1.

Timer/Counter 0 External Input (T0): a 1 to 0 transition on this pin will increment timer 0.

External Interrupt 1 (INT1#): a falling edge/low level on this pin will cause an external interrupt 1 if enabled.

External Interrupt 0 (INT0#): a falling edge/low level on this pin will cause an external interrupt 0 if enabled.

Serial Port 0 Transmit (TXD): transmits the serial port 0 data in serial port modes 1, 2, 3 and emits the synchronizing clock in serial port mode 0.

Serial Port 0 Receive (RXD): receives the serial port 0 data in serial port modes 1, 2, 3 and is a bi-directional data transfer pin in serial port mode 0.

Power Control Register (PCON)

The PCON register consists of the control bits for the Serial Port and the power reducing modes of operation.

Baud Rate Doubler Enable (SMOD): This bit enables/disables the serial baud rate doubling function.
1 = Baud rate will double that defined by baud rate generation equation when the serial port is used in modes 1, 2, and 3
0 = Baud rate will be that defined by baud rate generation equation

– Not implemented: Reserved for future use.*

General purpose flag bit (GF1): User defined.

General purpose flag bit (GF0): User defined.

Power Down bit (PD): Setting this bit activates Power Down operation in the SST89F54/58.

Idle Mode bit (IDL): Setting this bit activates Idle Mode operation in the SST89F54/58.

* User software should not write 1's to reserved bits, since these bits may be used in future microcontrollers. In the case of future use, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Power Control (PCON)

Location	7	6	5	4	3	2	1	0	Reset Value
87h	SMOD	–	–	–	GF1	GF0	PD	IDL	0xxx0000
					r/w				

Serial Data Buffer (SBUF)

The Serial Data Buffer functions as two separate registers, a transmit buffer and a receive buffer register, however it resides in a single address. The SBUF register may be used to either transmit or receive data. When data is written to the SBUF register, it must first pass through the transmit buffer where it is held for serial transmission (Moving a byte to the SBUF register is what initiates the transmission). When data is read from the SBUF register, it is read from the data receive buffer register. The following table provides a bit description of the SBUF register.

Serial Data Buffer (SBUF)

Location	7	6	5	4	3	2	1	0	Reset Value
99h	SBUF[7:0]								Indeterminate
	r/w								

Serial Port Control Register (SCON)

The SCON register consists of the following control bits for the Serial Port.

Serial Port mode specifier (SMO): (See table below)

Serial Port mode specifier (SM1): (See table below)

Multiprocessor Mode Enable (SM2): Enables the multiprocessor communication feature in modes 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Serial Port Setup Table below)

Serial Receive Enable (REN): Set/Cleared by software to Enable/Disable reception.

TB8: The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.

RB8: In modes 2 & 3, RB8 is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.

Transmit Interrupt Flag (TI): Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.

Receive Interrupt Flag (RI): Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

Serial Port Control Register (SCON)

Location	7	6	5	4	3	2	1	0	Reset Value
98h	SM0	SM1	SM2	REN	TB8	RB8	T1	R1	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Timer and Timer/Counter Control Registers

The Timer Registers are Timer/Counter 0, Timer/Counter 1 and Timer/Counter 2. The Timer Registers are 16-bit registers, consisting of two 8-bit SFRs. The TH0 and TL0 registers are the high and low bytes, respectively, of the 16-bit counting register for Timer/Counter 0. The TH1 and TL1 registers are the high and low bytes, respectively, of the 16-bit counting register for Timer/Counter 1. The TH2 and TL2 registers are the high and low bytes, respectively, for Timer/Counter 2. Also, the SST89F54/58 use two 8-bit capture registers (RCAP2H and RCAP2L) in conjunction with Timer 2 in order to hold copies of the TH2 and TL2 register contents, in response to a transition of the T2EX pin. The Timer 2 Capture LSB (RCAP2L) register is used to capture the TL2 value when timer 2 is configured in "Capture Mode." The Timer 2 Capture MSB (RCAP2H) register is used to capture the TH2 value when timer 2 is configured in "Capture Mode." The RCAP2H register is also used as the MSB of a 16-bit reload value when Timer 2 is configured in auto-reload mode. RCAP2L is also used as the LSB of a 16-bit reload value when Timer 2 is configured in auto-reload mode. The following tables provides a bit description of the TH0, TL0, TH1, TL1, TH2, TL2, RCAP2H and RCAP2L registers.

The Timer/Counter Control registers are the Timer Mode (TMOD), Timer Control (TCON), and the Timer 2 Control (T2CON) registers. These registers are also described below with the bit description on the tables.

Timer/Counter Mode Control Register (TMOD)

The upper nibble of the TMOD register controls Timer 1 and the lower nibble of the TMOD register controls Timer 0. The following bits are applicable for each Timer.

Gate: When TRx (in TCON) is set and GATE=1, TIMER/COUNTERx will run only while the INTx pin is HIGH (hardware control). When GATE=0, TIMER/COUNTERx will run only while TRx=1 (software control).

C/T#: Timer or Counter Selector. This bit is cleared for Timer operation (input from the internal system clock). This bit is set Counter operation (input from Tx input pin).

Mode Selector Bits: The Mode Selector Bits (M1 and M0) select the operating mode for the Timers. The table below describes the M1 and M0 bit values and the function of each operating mode.

M1	M0	Operating Mode	
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	**Timer 0:** TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	**Timer 1:** Timer/Counter 1 stopped.

Timer/Counter Mode Control Register (TMOD)

Location	7	6	5	4	3	2	1	0	Reset Value
	Timer 1				Timer 0				
89h	GATE	C/T#	M1	M0	GATE	C/T#	M1	M0	00h
	r/w								

Timer/Counter Control Register (TCON)

The TCON register consists of the overflow flag, run control, external interrupt edge flag and interrupt type control bits for Timer/Counter 0 and Timer/Counter 1.

Timer 1 Overflow Flag (TF1): Timer 1 Overflow Flag is set by hardware when the Timer/Counter 1 overflows. It is cleared by the hardware as the processor vectors to the interrupt service routine.

Timer 1 Run Control (TR1): Timer 1 Run Control bit is set/cleared by software to turn the Timer/Counter 1 ON/OFF.

Timer 0 Overflow Flag (TF0): Timer 0 Overflow Flag is set by hardware when the Timer/Counter 0 overflows. It is cleared by hardware as the processor vectors to the interrupt service routine.

Timer 0 Run Control (TR0): Timer 0 Run Control bit is set/cleared by software to turn the Timer/Counter 0 ON/OFF.

External Interrupt 1 Edge Flag (IE1): External Interrupt 1 Edge Flag. Set by hardware when External Interrupt edge is detected. It is cleared by hardware when the interrupt is processed.

Interrupt 1 Type Control (IT1): Interrupt 1 Type Control bit is set/cleared by software to specify a falling edge/low level triggered External Interrupt.

External Interrupt 0 Edge Flag (IE0): External Interrupt 0 Edge Flag is set by hardware when the External Interrupt edge is detected and cleared by hardware when the interrupt is processed.

Interrupt 0 Type Control (IT0): Interrupt 0 Type Control bit is set/cleared by software to specify falling edge/low level triggered External Interrupt

Timer/Counter Control Register (TCON)

Location	7	6	5	4	3	2	1	0	Reset Value
88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Timer 0 MSB (TH0)

Location	7	6	5	4	3	2	1	0	Reset Value
8Ch	TH0[7:0]								00h
	r/w								

Timer 0 LSB (TL0)

Location	7	6	5	4	3	2	1	0	Reset Value
8Ah	TL0[7:0]								00h
	r/w								

Timer 1 MSB (TH1)

Location	7	6	5	4	3	2	1	0	Reset Value
8Dh	TH1[7:0]								00h
	r/w								

Timer 1 LSB (TL1)

Location	7	6	5	4	3	2	1	0	Reset Value
8Bh	TL1[7:0]								00h
	r/w								

Timer/Counter 2 Control Register (T2CON)

The T2CON register consists of the control bits for Timer/Counter 2.

Timer 2 Overflow Flag (TF2): This flag is set by hardware and cleared by software. TF2 cannot be set when either (T2CON.7), RCLK = 1 or CLK = 1.

Timer 2 External flag (EXF2): This flag is set when either a capture or reload is caused by a negative transition on T2EX, and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the MCU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.

Receive Clock Flag (RCLK): When set, causes the Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 & 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.

Transmit Clock Flag (TCLK): When set, causes the Serial Port to use Timer 2 overflow pulses for its transmit clock in modes 1 & 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.

Timer 2 External Enable Flag (EXEN2): When set, allows a capture or reload to occur as a result of negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.

Timer 2 Run Control (TR2): Software START/STOP control for Timer 2. A logic 1 starts the Timer.

Timer or Counter Select (C/T2#): 0 = Internal Timer. 1 = External Event Counter (falling edge triggered).

Capture/Reload Flag (CP/RL2#): Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, Auto-Reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the Timer is forced to Auto-Reload on Timer 2 overflow.

Timer/Counter 2 Control Register (T2CON)

Location	7	6	5	4	3	2	1	0	Reset Value
C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Timer 2 MSB (TH2)

Location	7	6	5	4	3	2	1	0	Reset Value
CDh	TH2[7:0]								00h
	r/w								

Timer 2 LSB (TL2)

Location	7	6	5	4	3	2	1	0	Reset Value
CCh	TL2[7:0]								00h
	r/w								

Timer 2 Capture MSB (RCAP2H)

Location	7	6	5	4	3	2	1	0	Reset Value
CBh	RCAP2H[7:0]								00h
	r/w								

Timer 2 Capture LSB (RCAP2L)

Location	7	6	5	4	3	2	1	0	Reset Value
CAh	RCAP2L[7:0]								00h
	r/w								

Preliminary

Watchdog Registers

The Watchdog Timer feature of the SST89F54/58 is a mechanism which guards against software hang. The two Special Function Registers which form the Watchdog Timer are the Watchdog Timer Control (WDTC) register and the Watchdog Timer Data/Reload Register (WDTD).

Watchdog Timer Control Register (WDTC)

– Not implemented: Reserved for future use. *

Watchdog Timer Reset Enable (WDRE):
1: Enable Watchdog Timer Reset (not resetting WDTS flag)
0: Disable Watchdog Timer Reset

Watchdog Timer Reset Flag (WDTS):
1: Hardware sets the bit when a Watchdog timer overflow (reset, if enabled) occurs. Software can poll the bit for WDT overflow, if the WDT reset is disabled.
0: External hardware reset clears the flag. Software can also clear the flag by writing a “1” (SET bit) to the bit.

Watchdog Timer Refresh (WDT):
1: Software sets the bit to force a Watchdog timer refresh. It must be set directly before SWDT is set to prevent an unintentional refresh of the watchdog timer.
0: Hardware resets the bit when the refresh is done. Software cannot reset the bit to “0”.

Start Watchdog Timer (SWDT):
1: Software sets the bit to start the Watchdog timer running.
0: Software resets the bit to stop the Watchdog timer running

* User software should not write 1’s to reserved bits. These bits may be used in future SST89F54/58 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Watchdog Timer Control Register (WDTC)

Location	7	6	5	4	3	2	1	0	Reset Value
C0h	–	–	–	–	WDRE	WDTS	WDT	SWDT	x0h
					r/w	r/c	r/s	r/w	

Watchdog Timer Data/Reload Register (WDTD)

Watchdog Timer Reload Value (WDRL). The eight bit reload value is loaded into the high-byte of the watchdog timer when a refresh is triggered by a consecutive setting of the bits WDT (WDTC.1) and SWDT (WDTC.0).

Watchdog Timer Data/Reload Register (WDTD)

Location	7	6	5	4	3	2	1	0	Reset Value
86h	WDRL								00h
	r/w								

SST89F54/58 Unique Special Function Registers

The SST89F54/58 not only provide the standard 8xC5x SFRs, but it also implement five new registers, which include the SFDT, SFAL, SFAH, SFCM, and SFCF registers. Most of the unique features of the SST89F54/58 microcontroller family, such as the SuperFlash EEPROM memory are controlled by bits in these additional SFRs.

SuperFlash Data Register (SFDT)

SuperFlash Data Register (SFDT). Serves as the mailbox register for read and write data of the SuperFlash blocks.

SuperFlash Data Register (SFDT)

Location	7	6	5	4	3	2	1	0	Reset Value
B5h	SuperFlash Data Register								00h
	r/w								

SuperFlash Address Registers (SFAL, SFAH)

SuperFlash Address Low Register. Contains the low order address byte to the SuperFlash blocks.

SuperFlash Address Registers (SFAL, SFAH)

Location	7	6	5	4	3	2	1	0	Reset Value
B3h	SuperFlash Low Order Byte Address Register – A7 to A0 (SFAL)								00h
	r/w								

SuperFlash Address High Register. Contains the high order address byte to the SuperFlash blocks.

Location	7	6	5	4	3	2	1	0	Reset Value
B4h	SuperFlash High Order Byte Address Register – A15 to A8 (SFAH)								00h
	r/w								

Preliminary

SuperFlash Command Register (SFCM)

Flash Interrupt Enable (FIE):
0: not to re-map the second external interrupt source as a flash operation completion interrupt.
1: to re-map the second external interrupt source as a flash operation completion interrupt.

– Not implemented: Reserved for future use.*

Flash Operation Command (FCM):
1111: Block Erase
1110: Byte Program
1101: reserved
1100: Byte Verify
1011: Sector Erase
1010: Burst Program
1001: reserved
1000: reserved
0111: Chip Erase
0110: reserved
0101: reserved
0100: reserved
0011: reserved
0010: reserved
0001: reserved
0000: reserved

* User software should not write 1's to reserved bits. These bits may be used in future SST89F54/58 products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

SuperFlash Command Register (SFCM)

Location	7	6	5	4	3	2	1	0	Reset Value
B2h	FIE	–	–	–	FCM				00h
	r/w								

SuperFlash Configuration/Status Register (SFCF)

Upper Flash Block Visibility (VIS):
1: secondary 4 KBytes flash block is visible at the address range F000h - FFFFh
0: secondary 4 KBytes flash block is not visible

Security Lock Decoding (SECD):
11: both flash blocks are locked
10: both flash blocks are accessible only through In-Application Programming
01: only the secondary flash block is locked and the primary flash block is not locked
00: both flash blocks are not locked

– Not implemented: Reserved for future use *

Flash Operation Completion Polling Bit (BUSY):
1: device is busy working on the flash operation and device is not ready for next operation
0: flash operation is successfully completed and device is ready for the next operation or next byte

FREQ Input Clock Frequency Range: (setting change will not be reflected during a programming (SFCF[2:0]) operation)

111: 250 to 500 KHz
110: 0.5 to 1 MHz
101: 1 to 2 MHz
100: 2 to 4 MHz
011: 32 to 64 MHz
010: 16 to 32 MHz
001: 8 to 16 MHz
000: 4 to 8 MHz

* User software should not write 1's to reserved bits. These bits may be used in future SST89C5x products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

SuperFlash Configuration/Status Register (SFCF)

Location	7	6	5	4	3	2	1	0	Reset Value
B1h	VIS	SECD		-	BUSY	FREQ			00h
	r/w	r			r	r/w			

Flash Memory Programming

The SST89F54/58 internal flash memory may be programmed or erased using the following two methods:

- External Host Mode (parallel only)
- In-Application Programming (IAP) Mode (parallel or serial)

SST SuperFlash EEPROMs are single power supply re-programmable nonvolatile memories intended to be altered in-system with the use of one power supply. Similar to SRAMs, there exists the possibility of unintentional writes. The security lock prevents inadvertent alteration of data and code stored in the flash memory. **SST strongly recommends that the device remains locked at all times, unless the blocks are being intentionally programmed, to preserve data integrity.**

External Host Mode

The SST89F54/58 provide the user with a direct flash memory access that can be used for programming into the flash memory without using the MCU. The direct flash memory access is entered using the External Host Mode. While the reset input (RST) is continually held active (HIGH), if the PSEN# pin is forced by an input with a transition from high-to-low state, the device enters the External Host Mode arming state at this time. The MCU core is stopped from running and all the chip I/O pins are reassigned and become flash memory access and control pins. At this time the external host should initiate a “Read ID” operation. Once the device enters into External Host Mode, the internal flash memory blocks are accessed through the re-assigned I/O port pins as shown in Figure 7 below.

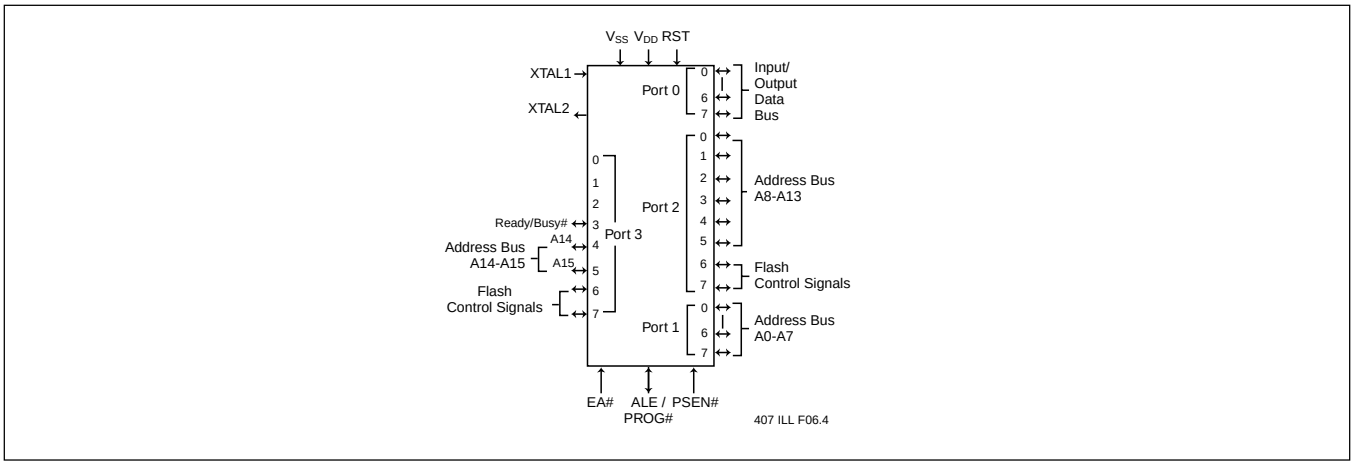


FIGURE 7: I/O PIN ASSIGNMENT FOR EXTERNAL HOST MODE

The I/O port pins are re-assigned by an external host, such as a printed circuit board tester, a PC controlled development board or an MCU programmer. The SST89F54/58 need a clock signal of 4-8 MHz on XTAL 1 to operate the External Host Mode. The clock signal is required to generate internal signals used to control Program and Erase operations.

When the chip is in the External Host Mode, Port 0 pins are assigned to be the parallel data input and output pins. Port 1 pins are assigned to be the non-muxed low order address bus signals for the internal flash memory (A0-A7). The first six bits of Port 2 pins (P2[0:5]) are assigned to be the non-muxed upper order address bus signals for the internal flash memory (A8-A13) along with two of the Port 3 pins (P3[6:7]). The RST, PSEN#, PROG#/ALE, EA# pins are assigned as the control signal pins. One of the Port 3 pins (P3.3) is assigned to be the Ready/Busy# status signal, which can be used for handshaking with the external host during a flash memory programming operation (Erase, Program, Verify, etc.). The P3.3 pin is internally self-timed and can be controlled by an external host asynchronously or synchronously.

External Host Mode Commands

The External Host Mode uses seven hardware commands, which are decoded from the control signal pins, in order to facilitate the internal flash memory erase, test and programming processes. The External Host Mode is enabled on the falling edge of PSEN#. The External Host Mode commands are enabled on the falling edge of ALE/PROG#. See the External Host Mode Commands Table for all signal logic assignments and the Flash Memory Programming/Verification Parameters Table for all timing parameter values for the External Host Mode Commands.

TABLE 1: EXTERNAL HOST MODE COMMANDS

Operation	RST	PSEN#	PROG# /ALE	EA#	P2.6	P2.7	P3.6	P3.7	P0[7:0]	P1[7:0]	P3[4:5] P2[5:0]
Read ID	H	L	H	H	L	L	L	L	DO	AL	AH
Chip Erase	H	L	⇓	H	H	H	H	L	X	X	X
Block Erase	H	L	⇓	H	H	H	H	H	X	X	AH
Sector Erase	H	L	⇓	H	H	H	L	H	X	AL	AH
Byte Program	H	L	⇓	H	L	H	H	H	DI	AL	AH
Burst Program	H	L	⇓	H	L	H	L	H	DI	AL	AH
Byte Verify	H	L	H	H	L	L	H	H	DO	AL	AH

407 PGM T1.6

Note: Symbol ⇓ signifies a negative pulse and the command is asserted during the low state of PROG#/ALE input. All other combinations of the above input pins are invalid and may result in unexpected behaviors.

Note: IntEn = Interrupt Enable for flash operation completion;
L = Logic low level; H = Logic high level; X = Don't care; AL = Address low order byte; AH = Address high order byte;
DI = Data Input; DO = Data Output

Preliminary

TABLE 2: FLASH MEMORY PROGRAMMING/VERIFICATION PARAMETERS

Parameter[1,2]	Symbol	4 MHz		8 MHz		Units
		Min	Max	Min	Max	
Programming Clock	CLK	250		125		ns
Reset Setup Time	T_{SU}	6		3		µs
Read ID Command Width	T_{RD}		500		250	ns
PSEN# Setup Time	T_{ES}	500		250		ns
Chip Erase Time	T_{CE}	8.5		4.3		ms
Block Erase Time	T_{BE}	8.5		4.3		ms
Sector Erase Time	T_{SE}	2.2		1.1		ms
Program Setup Time	T_{PROG}	1.8		0.9		ms
Byte Program Time [3]	T_{PB}	194		97		µs
Verify Command Delay Time	T_{OA}		45		45	ns
Verify High Order Address Delay Time	T_{AHA}		45		45	ns
Verify Low Order Address Delay Time	T_{ALA}		45		45	ns
First Burst Program Byte Time	T_{BUP1}	174	214	87	107	µs
Burst Program Time [3,4]	T_{BUP}	62	102	31	51	µs
Burst Program Recovery	T_{BUPRCV}		70		35	µs

Note: 407 PGM T2.6

1. All signals that align together in the timing diagrams should be derived from the same clock edge. Set up and hold times are not critical if they are within 10ns.
2. All timing measurements are from the 50% of the input to 50% of the output.
3. Don't program any byte twice before next erase.
4. The max time includes a 20µs burst program time-out limit.

The external host mode commands are the following:

1) Read ID

2 Chip Erase

3) Block Erase

4) Sector Erase

5) Byte Program

6) Busrt Program

7) Byte Verify

The following sections provide a brief description of the commands. The critical timing for all Erase and Program commands is dependent upon the 4-8 MHz external clock input at the XTAL1 pin. The high-to-low transition of the PROG# signal initiates the Erase and Program commands, which are synchronized internally. The Read commands are static reads, independent of both the PROG# signal level and clock input.

Product Identification

The Read ID command accesses the Signature Byte that identifies the device as a SST89F54/58 and the manufacturer as SST. External programmers primarily use these Signature Bytes, shown in Table 3, in the selection of programming algorithms. The Read ID command is selected by the byte code of 00h on P2[6:7] and P3[6:7]. See Figure 8 for the timing waveforms.

SIGNATURE BYTES TABLE

	Address	Data
Manufacturer's Code	0030h	BFh
SST89F54 Device Code	0031h	E3h
SST89F58 Device Code	0031h	E1h

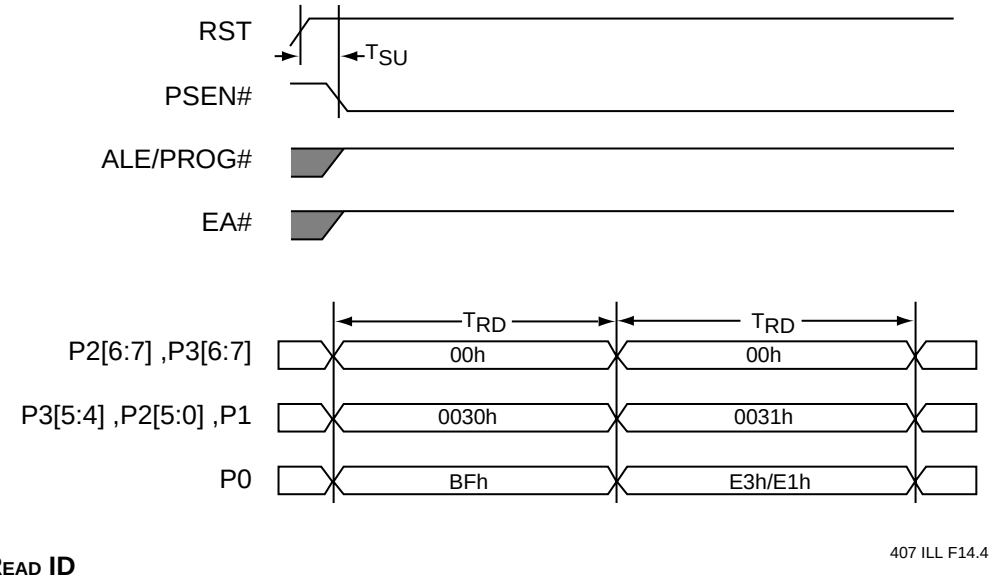


FIGURE 8: READ ID

Read chip signature and identification registers at the addressed location.

The following three commands are for erasing all or part of the memory array. All the data in the memory array will be erased to FFh. Memory addresses that are to be programmed must be in the erased state prior to programming. Selection of the Erase command to use, prior to programming the device, will be dependent upon the contents already in the array and the desired programming field size.

The Chip Erase command erases both flash memory blocks (16/32K and 4K) of the SST89F54/58. This command ignores the security lock status and will erase the Security Byte. The Chip Erase command is selected by the byte code of 0Eh on P2[6:7] and P3[6:7]. See Figure 9 for the timing waveforms.

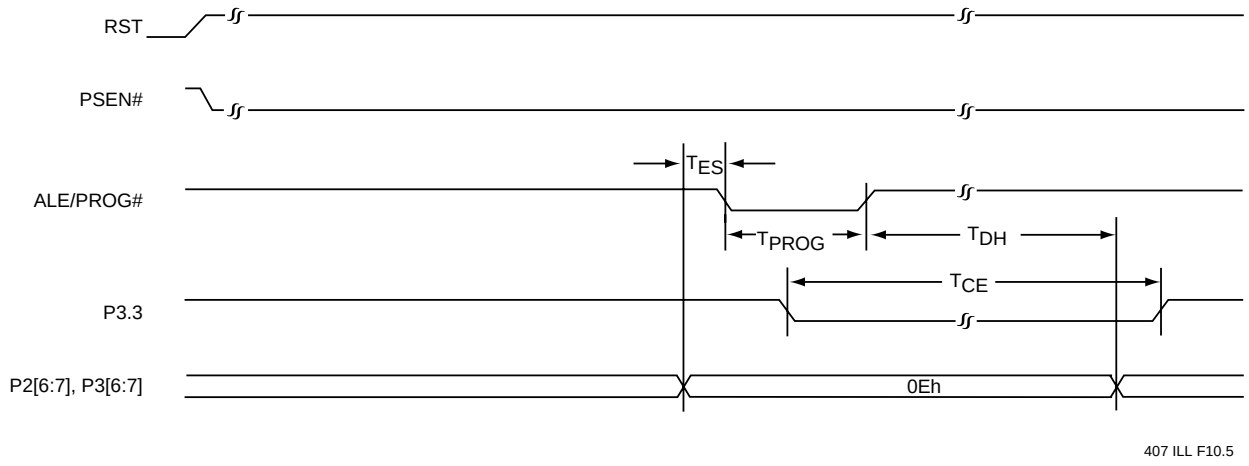


FIGURE 9: CHIP ERASE
Erase both flash memory blocks. Security lock is ignored and the security byte is erased too.

The Block Erase command erases one of the memory blocks (16/32K or 4K) of the SST89F54/58. This command will not enable if the selected memory block is security locked. The selection of the memory block to be erased is determined by P3[5:4] and P2[5:0]. If P3[5:4] and P2[5:0] is a “0Xh”, then the primary flash memory block (16/32K) is selected. If P3[5:4] and P2[5:0] is a “FXh”, then the secondary flash memory block (4K) is selected. The Block Erase command is selected by the byte code of 0Fh on P2[6:7] and P3[6:7]. See Figure 10 for the timing waveforms.

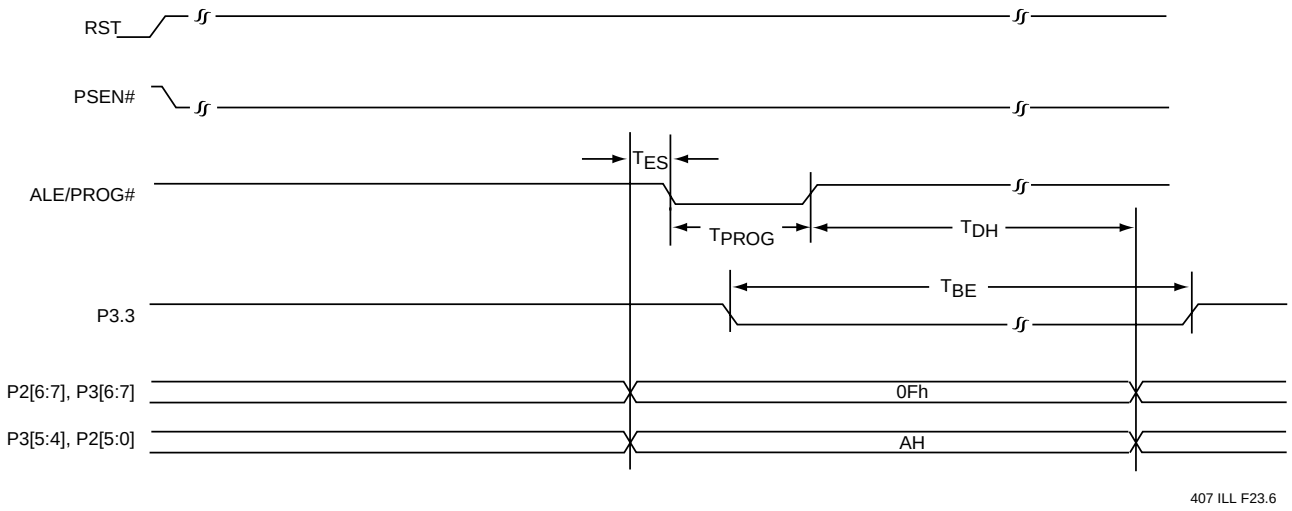


FIGURE 10: BLOCK ERASE
Erase one of the flash memory blocks, if the security lock is not activated on that flash memory block. The high address byte determines which block is erased. For example, if AH = 0Xh, primary flash memory block is erased.

The Sector Erase command erases a sector. The sector size for the primary flash memory block (Address locations 0-3FFFh/0-7FFFh) is 128 Bytes. The sector size for the secondary flash memory block (Address locations F000h-FFFFh) is 64 Bytes. This command will not enable if the selected memory block is security locked. The selection of the memory sector to be erased is determined by P2[0:5] (A8-A13) and P3[4:5] (A14 & A15). The Sector Erase command is selected by the byte code of 0Dh on P2[6:7] and P3[6:7]. See Figure 11 for the timing waveforms.

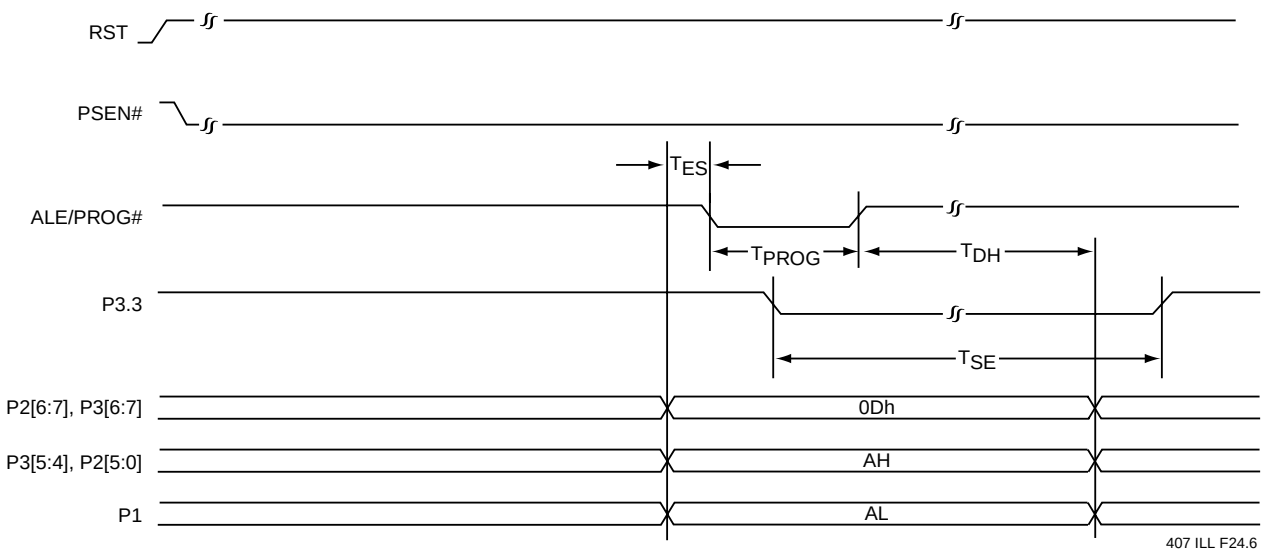


FIGURE 11: SECTOR ERASE
Erase the addressed sector if the security lock is not activated on that flash memory block.

Preliminary

The following two Program commands are for programming new data into the memory array. Selection of which Program command to use for programming will be dependent upon the desired programming field size. The Program commands will not enable if the selected memory block is security locked.

The Byte Program command programs data into a single byte. Ports P0[0:7] are used for data. The memory location is selected by P1[0:7], P2[0:5], and P3[4:5] (A0-A15). The Byte Program command is selected by the byte code of 07h on P2[6:7] and P3[6:7]. See Figure 12 for the timing waveforms.

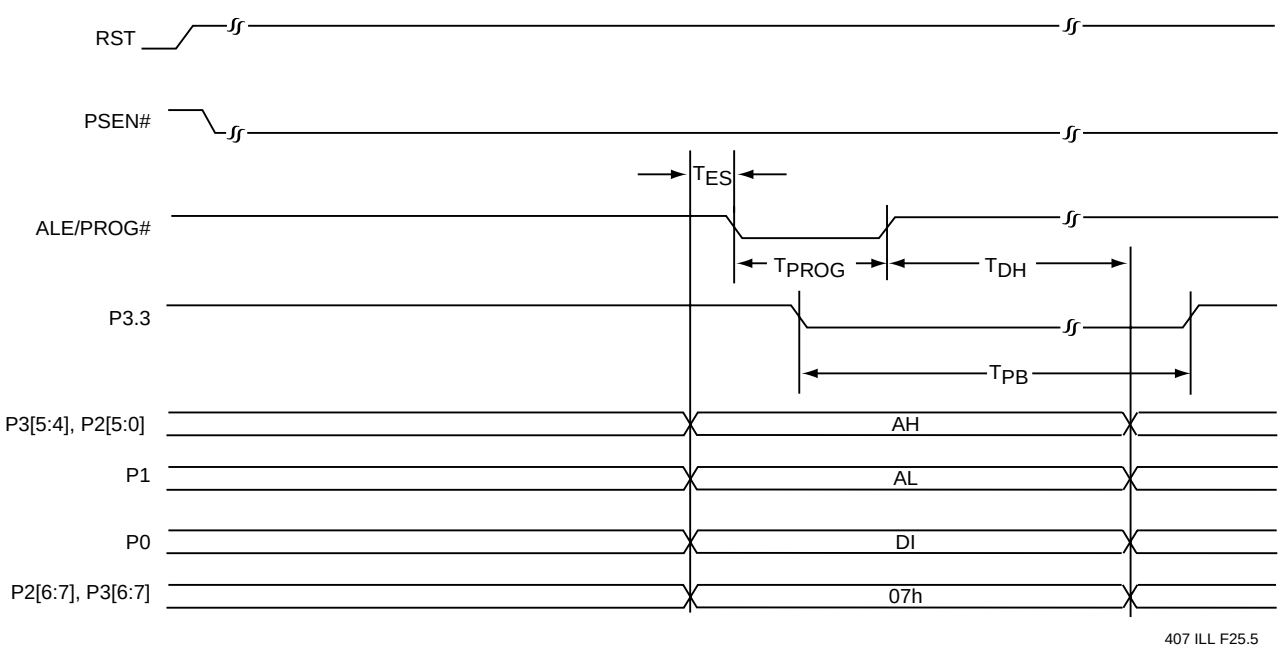


FIGURE 12: BYTE PROGRAM

Program the addressed code byte if the byte location has been successfully erased and not yet programmed. This operation is only allowed when the security lock is not activated on that flash memory block.

The Burst Program command programs data to an entire row, sequentially byte by byte. Ports P0[0:7] are used for data input. The memory location is selected by P1[0:7], P2[0:5], and P3[4:5] (A0-A15). The Burst Program command is selected by the byte code of 05h on P2[6:7] and P3[6:7]. See Figure 13 for the timing waveforms.

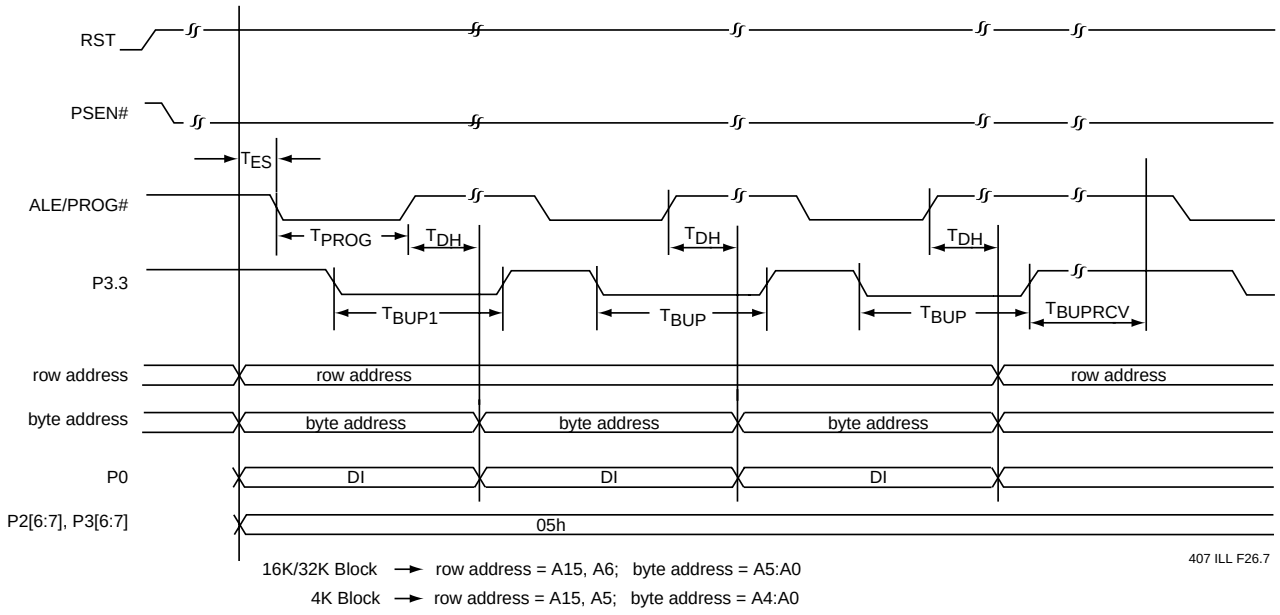


FIGURE 13: BURST PROGRAM

Program the entire addressed row by burst programming each byte sequentially within the row if the byte location has been successfully erased and not yet programmed. This operation is only allowed when the security lock is not activated on that flash memory block.

The termination of the Burst Program can be accomplished by: 1) Change to a new row-Addresses (Note: the row-Address range are different for the 4Kx8 flash block and for the 16/32K x 8 flash block.); 2) Change to a new command that requires a negative transition of the ALE/PROG# (i.e. any Erase or Program command); 3) Wait for time out limit expires (20 us); when programming the next byte.

Preliminary

The Byte Verify command allows the user to verify that the SST89F54/58 correctly performed an Erase or Program command. Ports P0[7:0] are used for data output. The memory location is selected by P1[7:0], P2[0:5], and P3[4:5] (A0-A15). This command will not enable if the Security Byte is enabled on the selected memory block. See Figure 14 for the timing waveforms.

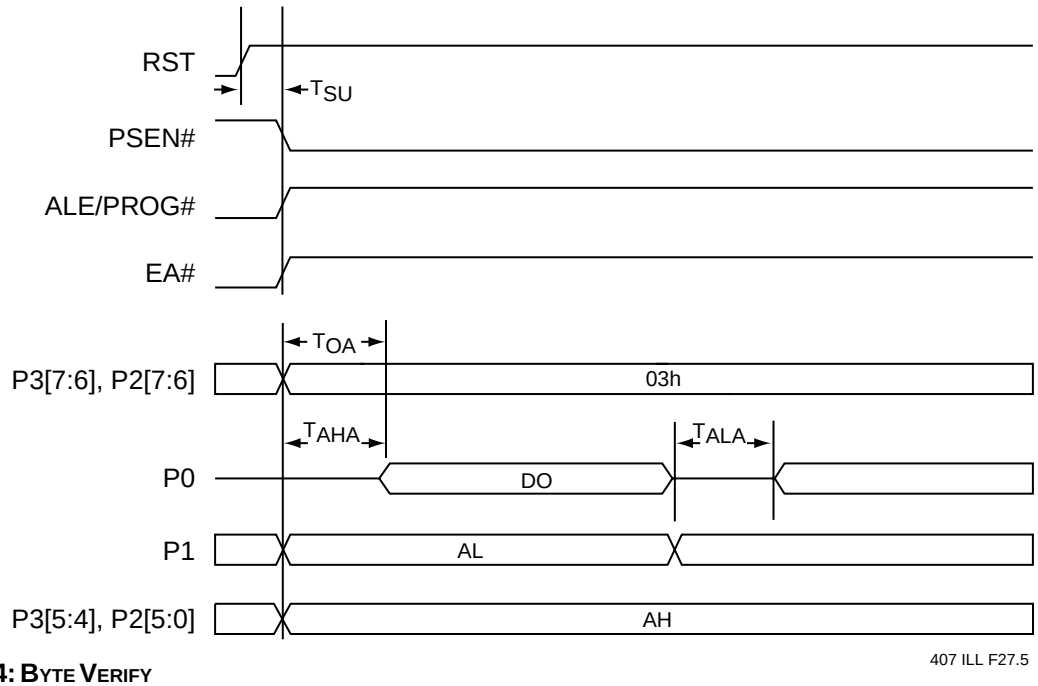


FIGURE 14: BYTE VERIFY

Read the code byte from the addressed flash memory location if the security lock is not activated on that flash memory block.

Programming a SST89F54/58

To program new data into the memory array, supply 5 volts to V_{DD} and RST, and perform the following steps.

1. Enable RST, and PSEN# in sequence per the appropriate timing diagram.
2. Raise EA# High (either V_{IH} or V_H).
3. Read the device and manufacturer ID, using the Read ID, command to ensure the correct programming algorithm.
4. Verify that the memory blocks or sectors for programming are in the erased state, FFh. If they are not erased, then erase them using the appropriate Erase command.
5. Select the memory location using the address lines (P1[0:7], P2[0:5], P3[4:5]).
6. Present the data in on P0[0:7] and the program command in P2[6:7] and P3[6:7].
7. Pulse ALE/PROG#.
8. Wait for low to high transition on READY/BUSY# (P3.3).
9. Continue steps 5 – 8 until programming is finished.
10. Verify the flash memory contents.

Flash Operation Status Detection (External Host Mode Handshake)

The SST89F54/58 provide two firmware means for an external host to detect the completion of a flash memory operation, therefore the external host can optimize the system program or erase cycle of the embedded flash memory. The end of a flash memory operation cycle (erase or program) can be detected by: 1) monitoring the Ready/Busy# bit at Port 3.3; 2) monitoring the Data# Polling bit at Port 0.7; These two detection mechanisms are described below.

Ready/Busy# (P3.3)

The progress of the flash memory programming can be monitored by the Ready/Busy# output signal. P3.3 is driven low, sometime after ALE/PROG# goes low during a flash memory operation to indicate the Busy# status of the flash programming controller. P3.3 is driven high when the flash programming operation is completed to indicate the Ready status.

During a Burst Program operation, P3.3 is driven high (Ready) in between each Byte Program to indicate the ready status to receive the next byte. When the external host detects the Ready status after a byte among the burst is programmed, it may then put the data/address (in the same page) of the next byte on the bus and drive ALE/PROG# low (pulse) immediately, before the 20 µs time-out limit expires

Data# Polling (P0.7)

SST89F54/58 also feature Data# Polling to indicate the flash memory programming operation status. The true data will be read from P0.7 via a Byte Verify flash operation, which can be asserted immediately after the initiation of a Program Byte or Burst Program operation.

The operational sequence is as follows:

1. A Byte Program or Burst Program write operation is initiated.
2. The software immediately issues a Byte Verify command, which internally writes the Data# Polling bit to P0.7.
3. The Data# Polling bit content initially shows the complements of the data programmed in Step 1 which indicates the Busy status of the flash block. When the program operation completes, then true data will appear at P0.7, which indicates the Ready status of the flash block.
4. Repeat Steps 1 to 3 for additional flash operations.

Preliminary

In-Application Programming Mode

The SST89F54/58 offers 20/36 KByte of in-application re-programmable flash memory. During In-Application Programming, the CPU of the microcontroller enters IAP Mode. The two blocks of flash memory allows the CPU to concurrently execute user code in one block, while the other is being reprogrammed. The CPU may also fetch code from external memory while all internal flash is being reprogrammed. The chip can start the In-Application Programming operation either with the external program code execution being enabled (EA# = L) or disabled (EA#=H). The mailbox registers (SFCM, SFAL, SFAH, SFDT and SFCF) located in the Special Function Register (SFR), control and monitor the device's erase and program process.

SST SuperFlash EEPROMs are single power supply re-programmable nonvolatile memories intended to be altered in-system with the use of one power supply. Similar to SRAMs, there exists the possibility of unintentional writes. The security lock prevents inadvertent alteration of data and code stored in the flash memory. **SST strongly recommends that the device remains locked at all times, unless the blocks are being intentionally programmed, to preserve data integrity.**

There are six (6) IAP Mode commands plus a Setup Command, which can be issued via the command mailbox register, SFCM at SFR location FBh. A pair of mailbox register addresses the memory array of the SuperFlash blocks: SFAL, low order address at SFR location F9h, and SFAH, the high order address at SFR location FAh. Data is latched through the SFDT register at SFR location F8h. SFCF, the configuration/status register at SFR location F7h, provides security lock status and program counter visibility to the secondary flash memory block. The list in Table 3 outlines all the commands and their associated bit settings of the mailbox registers.

In-Application Mode Commands

All of the following commands can only be initiated in the IAP Mode. In all situations, writing a command to the (SFCM) register will initiate all of the operations. All commands (except Chip Erase) will be ignored if the selected memory block is security locked. The critical timing for all Erase and Program commands is dependent upon the minimum frequency pre-scaling factor specified in the SuperFlash Configuration/Status Register (SFCF[2:0]).

The Chip Erase command erases both memory blocks (16/32K and 4K). This command ignores the security lock status and will erase the Security Byte. The Chip Erase command is initiated as follows:

1.) Set the operational frequency pre-scaling factor FREQ in (SFCF[2:0]).

2.) Move 55h to the SuperFlash Data Register (SFDT) (i.e. "MOV SFDT, #55h" where SFDT is the register address). This serves as a dual level precautionary measure to prevent accidental chip erasure.

3.) Move the Chip Erase command to the SuperFlash Command Register (SFCM) (i.e. "MOV SFCM, #87h" *or* "MOV SFCM, #07h"). If SFCM[7] is set, INT1# will interrupt the system when the erase is complete. Otherwise you must poll SFCF[3] to determine when the erase is complete.

The Block Erase command erases one of the two memory blocks (16/32K or 4K). The selection of the memory block to be erased is determined by the AH byte (SFAH) of the SuperFlash Address Register. If (SFAH) is a "0Xh", the primary flash memory block is selected (16/32K). If (SFAH) is a "FXh", the secondary flash memory block is selected (4K). The Block Erase command is initiated as follows:

1.) Set the operational frequency pre-scaling factor FREQ in (SFCF[2:0]).

2.) Move the block address to (SFAH) (i.e. "MOV SFAH, F0h" *or* "MOV SFAH, 00h").

3.) Move 55h to the SuperFlash Data Register (SFDT) (i.e. "MOV SFDT, #55h" where SFDT is the register address). This serves as a dual level precautionary measure to prevent accidental chip erasure.

4.) Move the Block Erase command to (SFCM) (i.e. "MOV SFCM, #8Fh" *or* "MOV SFCM, #0Fh"). If SFCM[7] is set, INT1# will interrupt the system when the erase is complete. Otherwise you must poll the SFCF[3] register to determine when the erase is complete.

The Sector Erase command erases a sector. The sector size for the primary flash memory block (Address locations 0-3FFFh/7FFFh) is 128 Bytes. The sector size for the secondary flash memory block (Address locations F000h-FFFFh) is 64 bytes. The Sector Erase command is initiated as follows:

1.) Set the operational frequency pre-scaling factor FREQ in (SFCF[2:0]).

2.) Move the sector address to (SFAH) and to (SFAL).

3.) Move the Sector Erase command to (SFCM) (i.e. "MOV SFCM, #8Bh" *or* "MOV SFCM, #0Bh"). If SFCM[7] is set, INT1# will interrupt the system when the erase is complete. Otherwise you must poll the SFCF[3] register to determine when the erase is complete.

The following two Program commands are for programming new data into the memory array. The portion of the memory array to be programmed should be in the erased state, FFh. If the memory is not erased, then erase it with the appropriate erase command. WARNING: do not write (program or erase) to a block that the code is currently fetching from. This can "hang" the CPU and may even corrupt program data as it is being executed.

The Byte Program command programs data into a single byte. The Byte Program command is initiated as follows:

1.) Set the operational frequency pre-scaling factor FREQ in (SFCF[2:0]).

2.) Move the high order address byte AH to (SFAH).

3.) Move the low order address byte AL to (SFAL).

4.) Move the data to the (SFDT).

5.) Move the Byte Program command to (SFCM) (i.e. "MOV SFCM, #8Eh" *or* "MOV SFCM, #0Eh"). If SFCM[7] is set, INT1# will interrupt the system when the program is complete. Otherwise you must poll the SFCF[3] register to determine when the program is complete.

Preliminary

The Burst Program command programs data to an entire row, sequentially byte by byte. The Burst Program command is initiated as follows:

1.) Set the operational frequency pre-scaling factor FREQ in (SFCF[2:0]).
2.) Move the high order address byte to (SFAH).
3.) Move the low order address byte to (SFAL).
4.) Move the data to (SFDT).
5.) Move the Burst Program command to (SFCM) (i.e. “MOV SFCM, #8Ah” *or* “MOV SFCM, #0Ah”). If SFCM[7] is set, INT1# will interrupt the system when the program is complete. Otherwise you must poll the SFCF[3] register to determine when the program is complete.
6.) Wait for interrupt or poll SFCF[3] before programming the next byte.
7.) If another Burst Program is needed return to step 4 (same row).

The Byte Verify command allows the user to verify that the SST89F54/58 has correctly performed an Erase or Program command. The Byte Verify command is initiated as follows:

1.) Move the high order address byte AH to (SFAH).
2.) Move the low order address byte AL to (SFAL).
3.) Move the Byte Verify command to (SFCM) (i.e. "MOV SFCM, #8Ch" *or* "MOV SFCM, #0Ch").

Polling

A command that uses the polling method to signify the completion of an operation must check the BUSY bit (SFST[3]). Copy the SFST register into temporary memory and mask the polling bit. Once it is isolated, the status of the BUSY bit can be checked.

MOVC instruction is used for verification of the Programming and Erase operation of the flash memory.

TABLE 3: IN-APPLICATION MODE COMMANDS

Operation	SFAH [7:0]	SFAL [7:0]	SFDT [7:0]	SFCM[1] [7:0]
Chip Erase	X	X	55h	87h/07h
Block Erase	FXh/0Xh	X	55h	8Fh/0Fh
Sector Erase	AH	AL	X	8Bh/0Bh
Byte Program	AH	AL	DI	8Eh/0Eh
Burst Program	AH	AL	DI	8Ah/0Ah
Byte Verify	AH	AL	DO	8Ch/0Ch

407 PGM T3.11

Notes: X = Don't care; AL = Address low order byte; AH = Address high order byte;
DI = Data Input; DO = Data Output
1. SFCM(7:0) 8X/0X = Interrupt/Polling Enable for flash operation completion

Power Mode Management

The SST89F54/58 offer two power-saving modes of operation for applications where power consumption is critical. The three power-saving modes are the Power Down, and the Standby modes. During these modes, backup power is supplied by V_{DD}. The Power Down mode is selected by the Power Down bit (PD) of the Power Control (PCON) special function register shown in the figure below. In the Power Down mode, the oscillator is frozen. In the Standby mode, an external hardware gates OFF the external clock input to the MCU. The following sections describe the Idle and Power Down, and Standby modes of operation.

Power Control (PCON)

Location	7	6	5	4	3	2	1	0	Reset Value
87h	SMOD	–	–	–	GF1	GF0	PD	–	0xxx0000
	r/w								

Power Down Mode

A Power Down mode can be invoked by software in order to save more power. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power Down mode is exited.

The Power Down Mode is selected by setting the Power Down bit (PD) of the Power Control (PCON) register. Consequently, the instruction that sets the PD bit is the last instruction that will be executed before the MCU goes into the power down mode of operation. The current draw is reduced to approximately 15µA and the supply voltage can be reduced to a V_{DD} of 2V. The Power Down Mode essentially leaves the MCU in a frozen state while preserving the values that the on-chip RAM and the Special Function Registers held before the Power Down Mode operation was initiated. The oscillator stops, thereby stopping the system clock and all functions. The ALE and PSEN# pins output logic low levels during the Power Down mode. The port pins output the values held by their respective Special Function Registers.

On the SST89F54/58, either a hardware reset or an external interrupt can cause an exit from Power Down. It is imperative that V_{DD} is not reduced before the Power Down Mode is initiated and that V_{DD} is restored to its normal operating level before the Power Down Mode is terminated. In order to properly terminate Power Down, the hardware reset or external interrupt should not be executed before V_{DD} is restored to its normal operating level, and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

A hardware reset will cause the PD bit to be cleared asynchronously (redefines all the SFR's to their reset value but does not change the on-chip RAM). After reset, the oscillator will restart and the MCU will then resume program execution at 0000h. Note that in order to reset the SST89F54/58, a high level must by applied to the RST pin for at least two machine cycles. Since the RST line must be held high for at least two machine cycles, two or three machine cycles of program execution may be executed before the internal reset algorithm takes control. The on-chip hardware prevents access to the internal 256 bytes of on-chip RAM during this time, however the port pins are accessible. In order to eliminate the possibility of unexpected outputs, the instruction following the one that invokes the Power Down Mode should not attempt to access a port pin or external or internal program memory during the system reset.

An external interrupt allows both the SFRs and on-chip RAM to retain their values. With an external interrupt, the interrupt to be used (INT0# or INT1#) must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

Standby (Stop Clock) Mode

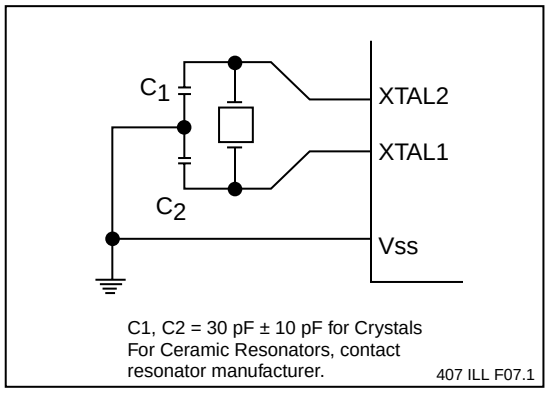
Standby Mode reduces device current drain to approximately 15 microamperes. It is controlled by hardware (gating on/off the system clock). It is initiated by an external hardware that gates OFF the external clock input to the MCU. This gating shall be synchronized with an input clock transition (low-to-high or high-to-low). The state of the MCU is totally preserved when the SST89F54/58 is in the Standby Mode. Standby Mode is exited by a gate ON external clock, and execution begins at the next clock in normal processing.

Standby Mode and Power Down Mode are similar, both reduce current drain. However, entry to the two modes is different, Power Down Mode is entered by software, while Standby Mode is controlled by hardware.

On-Chip Peripheral Components

Clock Input Options

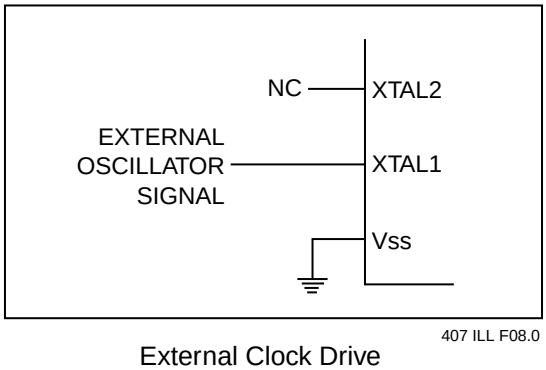
The SST89F54/58 clock may be driven by the internal or an external oscillator. The configuration required for using the internal oscillator is presented below.



Using the On-Chip Oscillator

FIGURE 15: INTERNAL OSCILLATOR CHARACTERISTICS

The configuration required for using an external oscillator is presented below.



External Clock Drive

FIGURE 16: EXTERNAL OSCILLATOR CHARACTERISTICS

Note that when driving the SST89F54/58 from an external clock source, XTAL2 should be left disconnected and XTAL1 should be connected to the external source. Also note that at start-up, the external oscillator may encounter a higher capacitive load at XTAL1 due to interaction between the amplifier and its feedback capacitance. However, the capacitance shall not exceed 15 pF once the external signal meets the V_{IL} and V_{IH} specifications.

Minimum and maximum high and low times specified on the data sheet must be observed when using an external oscillator, however, there are no requirements on the duty cycle of the external clock cycle. Table 4 and Figure 17 present the required minimum and maximum timing parameters for the external clock waveform.

TABLE 4: EXTERNAL CLOCK DRIVE

Symbol	Parameter	Oscillator						Units
		12 MHz		33 MHz		Variable		
		Min	Max	Min	Max	Min.	Max.	
1/tCLCL	Oscillator Frequency					0	33	MHz
tCHCX	High Time					0.35tCLCL	0.65tCLCL	ns
tCLCX	Low Time					0.35tCLCL	0.65tCLCL	ns
tCLCH	Rise Time		20		5			ns
tCHCL	Fall Time		20		5			ns

407 PGM T4.2

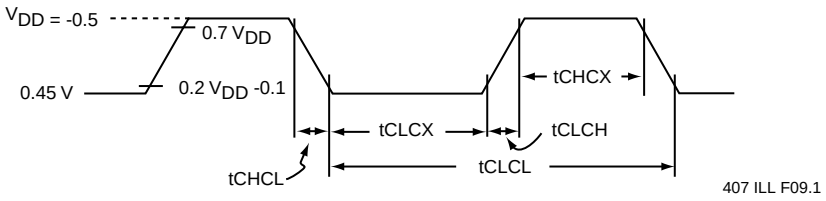


FIGURE 17: EXTERNAL CLOCK DRIVE WAVEFORM

Timers/Counters

The SST89F54/58 have three 16-bit registers that can be used as either timers or event counters. The three Timers/ Counters are the Timer 0 (T0), Timer 1 (T1) and Timer 2 (T2) registers. These three registers are located in the SFR as pairs of 8-bit registers. The low byte of the T0 register is stored in the Timer 0 LSB (TL0) special function register and the high byte of the T0 register is stored in the Timer 0 MSB (TH0) special function register. The low byte of the T1 register is stored in the Timer 1 LSB (TL1) special function register and the high byte of the T1 register is stored in the Timer 1 MSB (TH1) special function register. The low byte of the T2 register is stored in the Timer 2 LSB (TL2) special function register and the high byte of the T2 register is stored in the Timer 2 MSB (TH2) special function register. Note that the T0, T1 and T2 registers are alternate functions of port pins, as detailed in the Special Function Registers section under the Port 3 (P3) register description. The following sections describe the functions of Timer 0, Timer 1 and Timer 2 for the SST89F54/58.

Timer 0

The Timer/Counter Mode Control (TMOD) special function register determines the function of Timer 0. The Mode Selector bits (M1) and (M0) of the TMOD special function register are used to select one of four operating modes, mode 0, mode 1, mode 2 or mode 3 for Timer 0. The table below outlines the combinations for the M0 and M1 registers and the resulting operating mode.

M1	M0	Operating Mode	
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	**Timer 0:** TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	**Timer 1:** Timer/Counter 1 stopped.

Preliminary

Timer 0 may either function as a timer or as an event counter. The timer or counter function is selected by the Timer or Counter Select (C/T#) bit for Timer 0 in the TMOD register (TMOD.2). If the TMOD.2 bit is cleared, then Timer 0 will function as an internal timer with input from the internal system clock. If the TMOD.2 bit is set, then Timer 0 will function as an external event counter with input from the Timer 0 input pin. When used as a timer, the Timer 0 register is incremented once per machine cycle (once every 12 clock periods). When used as a counter, the Timer 0 register is incremented when a high-to-low (negative clock edge) transition is applied to the Timer 0 input pin. Two complete machine cycles are required for the MCU to see the high-to-low transition, therefore, the input must be held high for at least one clock cycle and then low for at least one clock cycle. The TMOD register is presented in the following figure.

Timer/Counter Mode Control Register (TMOD)

Location	7	6	5	4	3	2	1	0	Reset Value
	Timer 1				Timer 0				
89h	GATE	C/T#	M1	M0	GATE	C/T#	M1	M0	00h
	r/w								

Note that when the Timer 0 Run Control (TR0) bit of the Timer/Counter Control (TCON) register is set and the GATE bit for Timer 0 in the TMOD register (TMOD.3) is set, Timer 0 will run only while the INT0# pin is at a logic level high. When the TMOD.3 bit is cleared, Timer 0 will run only while TR0 is at a logic level high.

Timer/Counter Control Register (TCON)

Location	7	6	5	4	3	2	1	0	Reset Value
88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Timer 0 Mode 0

The following figure presents a functional overview of the Timer/Counter 0 Mode 0.

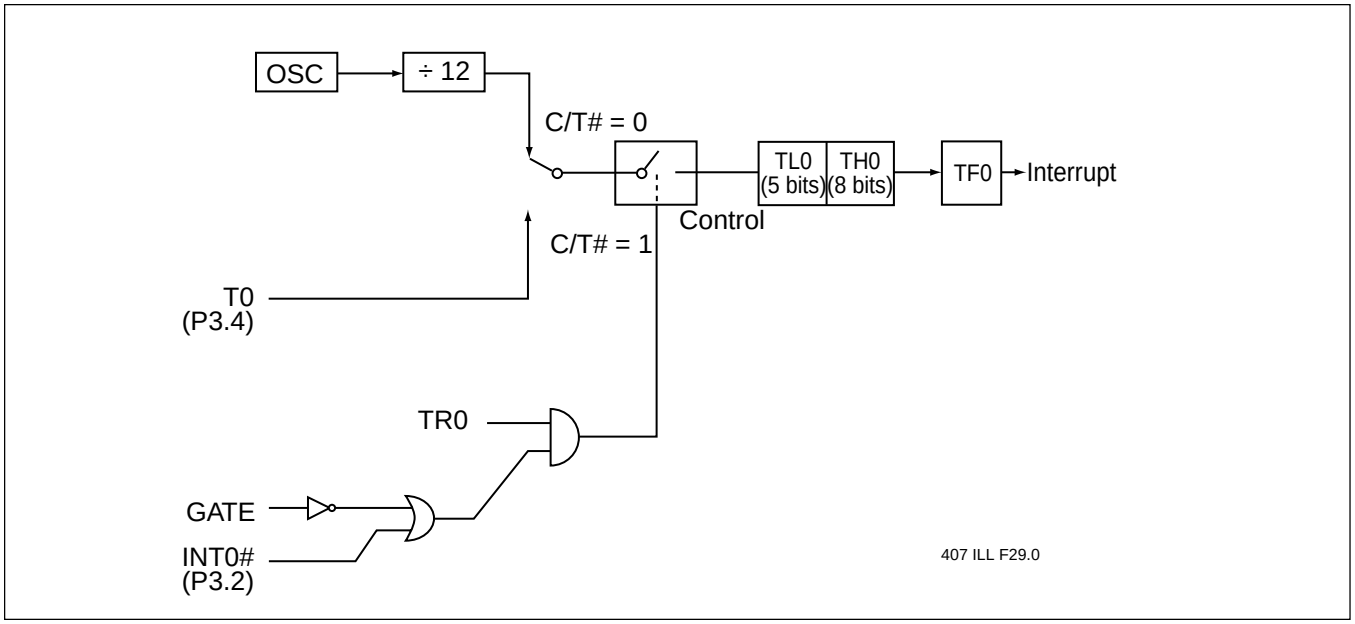


FIGURE 18: TIMER/COUNTER 0 MODE 0

In mode 0 timer 0 is configured as a 13-bit register that can be thought of as an 8-bit counter preceded by a 5-bit divide-by-32 prescaler. The 8-bit count is in the TH0 register and the 5-bit prescaler is the primary 5 bits of the TL0 register. The secondary 3 bits of the TL0 register are random and should be ignored. As the 13-bit count in the TL0 and TH0 registers goes from all 1's to all 0's, the Timer 0 Interrupt (TF0) flag is set.

The input source for the counter is selected by the TMOD.2 bit. The counting process can be enabled or disabled independently of the input. In order for counting to proceed, the TR0 bit must be at a logic level high while either the TMOD.3 bit is at a logic level low or the INT0# pin is held at a logic level low. The use of the TMOD.3 and the TR0 bits allows counting to be controlled by software and the use of the INT0# pin allows counting to be controlled by external hardware. Note that INT0# and Timer 0 are alternate functions of port 3 pins.

Timer 0 Mode 1

The following figure presents a functional overview of the Timer/Counter 0 Mode 1.

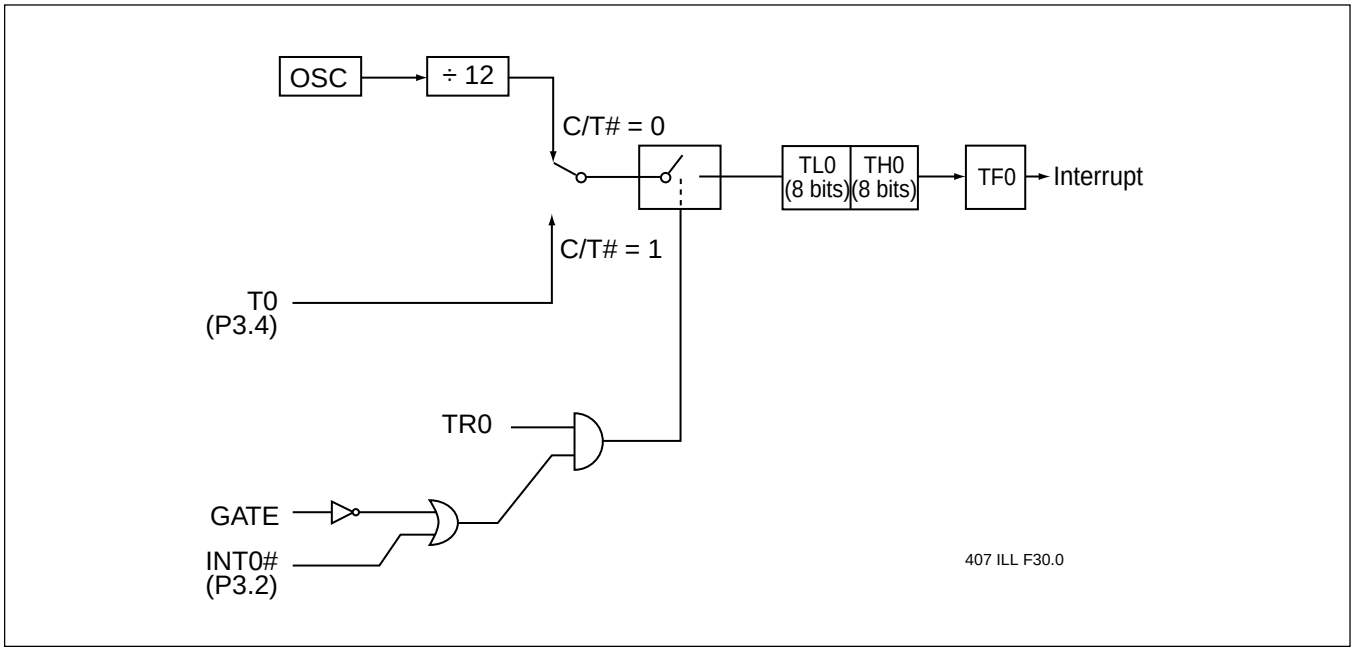


FIGURE 19: TIMER/COUNTER 0 MODE 1

Mode 1 for Timer 0 is the same as mode 0, however, the timer register is 16 bits, with all eight bits of TL0 being used.

Timer 0 Mode 2

The following figure presents a functional overview of the Timer/Counter 0 Mode 2.

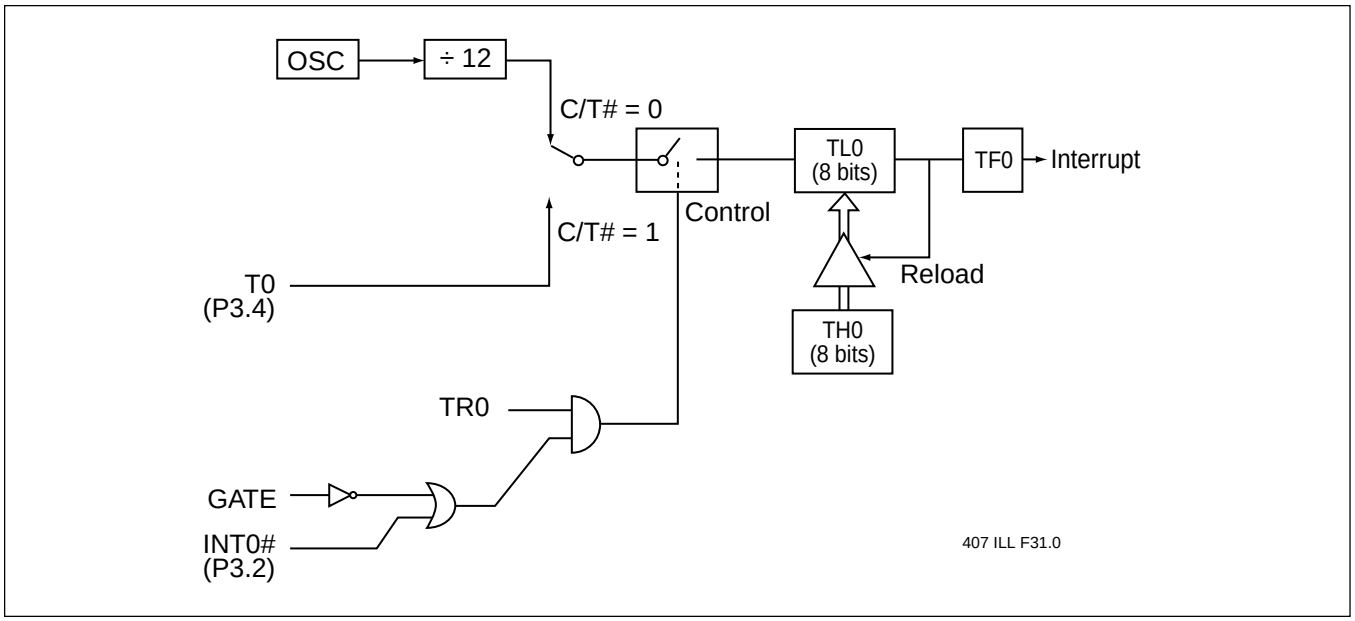


FIGURE 20: TIMER/COUNTER 0 MODE 2

Mode 2 operation is much the same as mode 0, except that the TL0 register is used as an 8-bit counter and the TH0 register is used to hold a preset number. When the contents of TL0 go from all 1s to all 0s, the interrupt TF0 is set and the contents of TH0 are transferred to TL0. The contents of TH0 remain unchanged. Since the contents of TH0 are under software control, the counter can be made to divide the count source by any number from 1 to 255 by means of the automatic reload of TH0 into TL0.

Timer 0 Mode 3

The following figure presents a functional overview of the Timer/Counter 0 Mode 3.

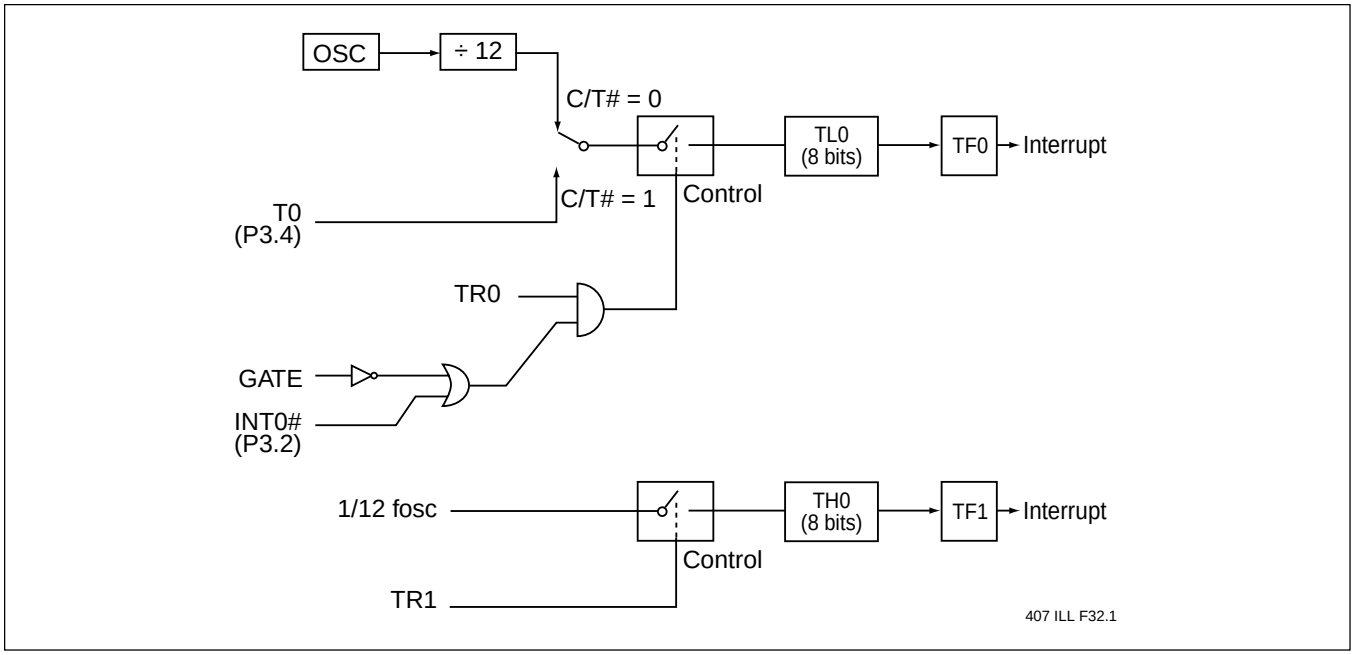


FIGURE 21: TIMER/COUNTER 0 MODE 3

In mode 3 operation, timer 0 is split into two separate counters. The first counter is the same as mode 0, except that TL0 is used as an 8-bit counter and there is no prescaler. The second counter uses TH0 as an 8-bit counter. The count source is the oscillator divided by 12 and the enable control is the TR1 bit. Note that the first counter sets the TF0 interrupt flag and the second counter sets the TF1 interrupt flag.

Timer 1

The Timer/Counter Mode Control (TMOD) special function register determines the function of Timer 1. The Mode Selector bits (M1) and (M0) of the TMOD special function register are used to select one of four operating modes, mode 0, mode 1, mode 2 or mode 3 for Timer 1. The table below outlines the combinations for the M0 and M1 registers and the resulting operating mode.

M1	M0	Operating Mode	
0	0	0	13-bit Timer (MCS-48 compatible)
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	**Timer 0:** TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	**Timer 1:** Timer/Counter 1 stopped.

Timer 1 may either function as a timer or as an event counter. The timer or counter function is selected by the Timer or Counter Select (C/T#) bit for Timer 1 in the TMOD register (TMOD.6). If the TMOD.6 bit is cleared, then Timer 1 will function as an internal timer with input from the internal system clock. If the TMOD.6 bit is set, then Timer 1 will function as an external event counter with input from the Timer 1 input pin. When used as a timer, the Timer 1 register is incremented once per machine cycle (once every 12 clock periods). When used as a counter, the Timer 1 register is incremented when a high-to-low (negative clock edge) transition is applied to the Timer 1 input pin. Two complete machine cycles are required for the MCU to see the high-to-low transition, therefore, the input must be held high for at least one clock cycle and then low for at least one clock cycle. The TMOD register is presented in the following figure.

Timer/Counter Mode Control Register (TMOD)

Location	7	6	5	4	3	2	1	0	Reset Value
	Timer 1				Timer 0				
89h	GATE	C/T#	M1	M0	GATE	C/T#	M1	M0	00h
	r/w								

Note that when the Timer 1 Run Control (TR1) bit of the Timer/Counter Control (TCON) register is set and the GATE bit for Timer 1 in the TMOD register (TMOD.7) is set, Timer 1 will run only while the INT1# pin is at a logic level high. When the TMOD.7 bit is cleared, Timer 1 will run only while TR1 is at a logic level high.

Timer/Counter Control Register (TCON)

Location	7	6	5	4	3	2	1	0	Reset Value
88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Preliminary

Timer 1 Mode 0

The following figure presents a functional overview of the Timer/Counter 1 Mode 0.

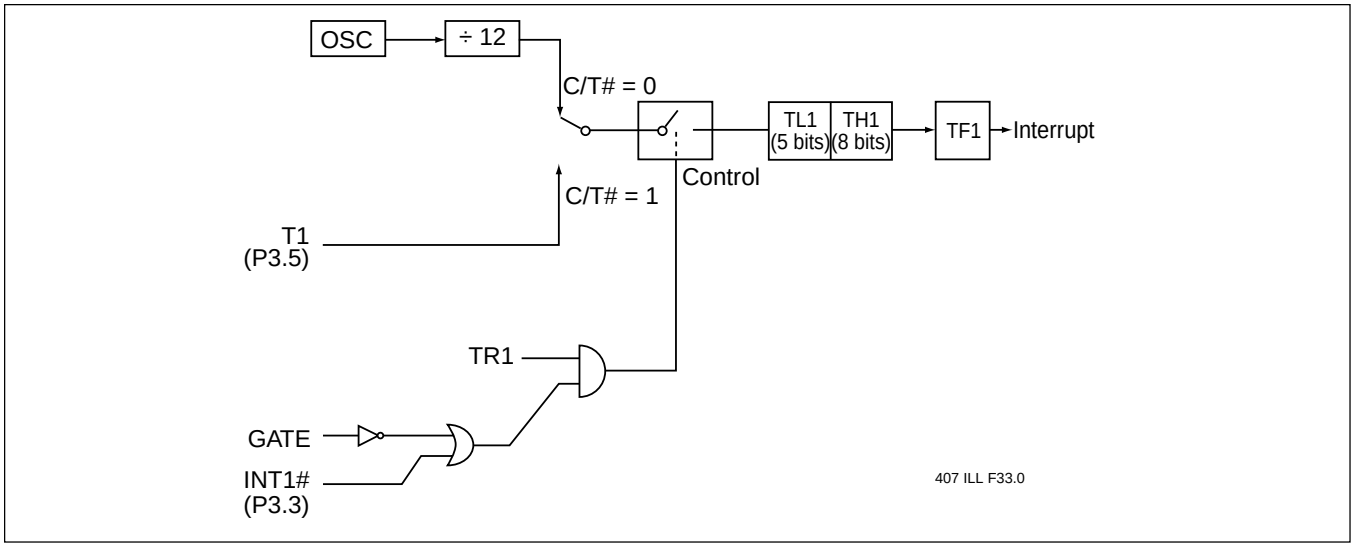


FIGURE 22: TIMER/COUNTER 1 MODE 0

In mode 0 timer 1 is configured as a 13-bit register that can be thought of as an 8-bit counter preceded by a 5-bit divide-by-32 prescaler. The 8-bit count is in the TH1 register and the 5-bit prescaler is the primary 5 bits of the TL1 register. The secondary 3 bits of the TL1 register are random and should be ignored. As the 13-bit count in TL1 and TH1 goes from all 1's to all 0's, the Timer 1 Interrupt (TF1) flag is set. The timer interrupt is the link between the counter hardware and the program software.

The input source for the counter is selected by the TMOD.6 bit. The counting process can be enabled or disabled independently of the input. In order for counting to proceed, the TR1 bit must be at a logic level high while either the appropriate TMOD.7 bit is at a logic level low or the INT1# pin is held at a logic level low. The use of the TMOD.7 and the TR1 bits allows counting to be controlled by software and the use of INT1# pin allows counting to be controlled by external hardware. Note that INT1# and Timer 1 are alternate functions of port 3 pins.

Timer 1 Mode 1

The following figure presents a functional overview of the Timer/Counter 1 Mode 1.

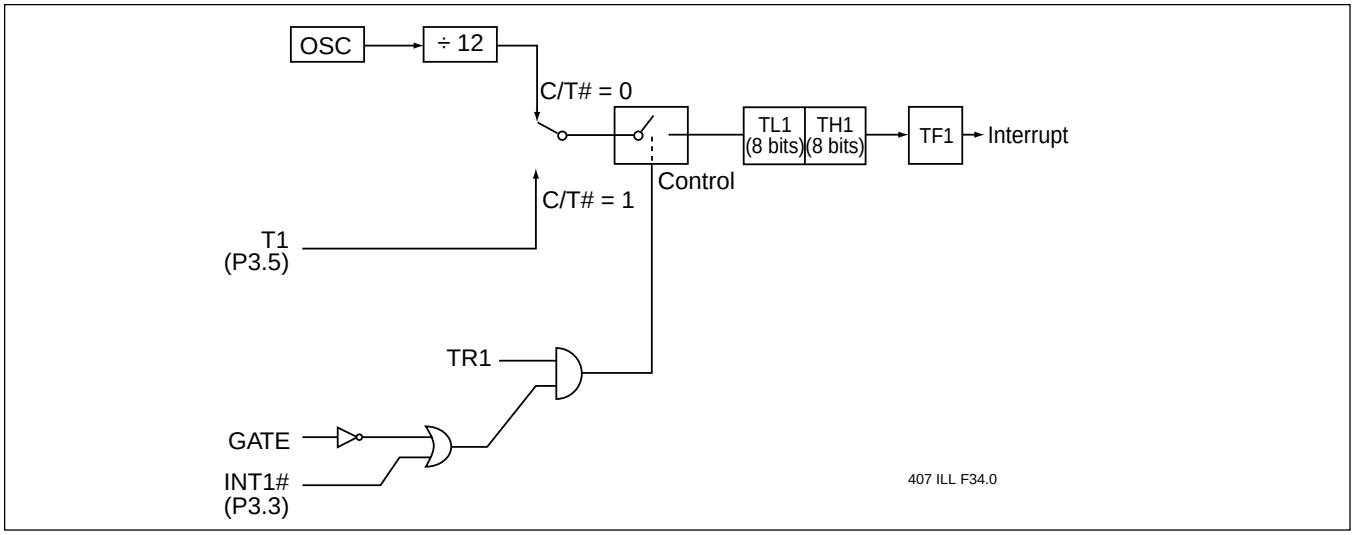


FIGURE 23: TIMER/COUNTER 1 MODE 1

Mode 1 for Timer 1 is the same as mode 0, however, the timer register is 16 bits, with all eight bits of TL1 being used.

Timer 1 Mode 2

The following figure presents a functional overview of the Timer/Counter 1 Mode 2.

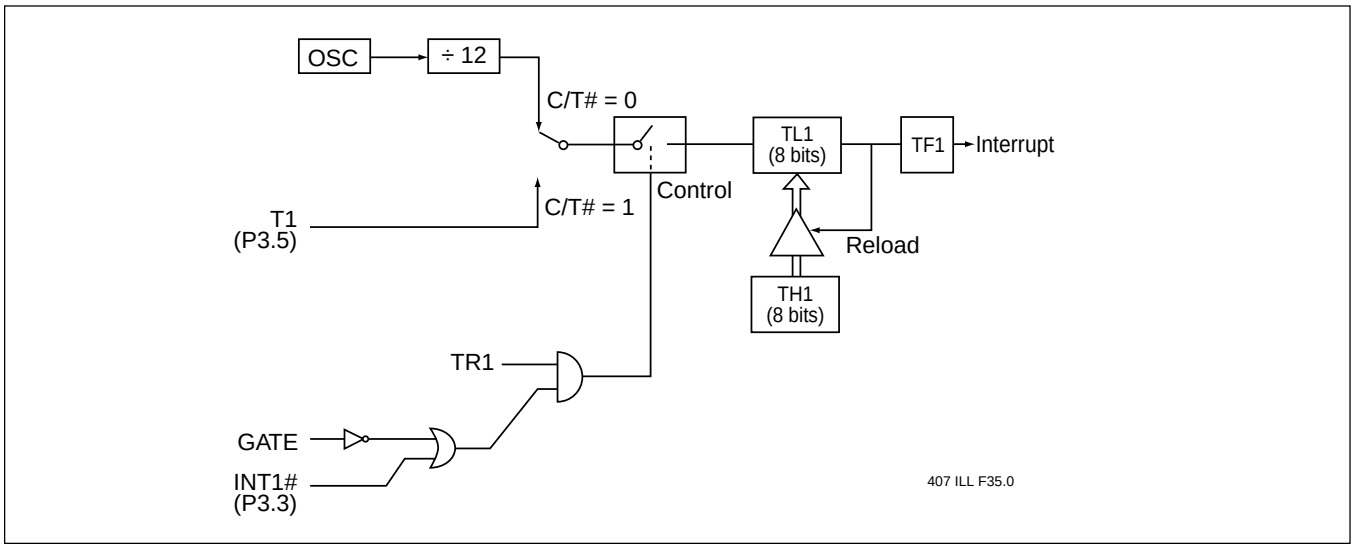


FIGURE 24: TIMER/COUNTER 1 MODE 2

Mode 2 operation is much the same as mode 0, except that the TL1 register is used as an 8-bit counter and the TH1 register is used to hold a preset number. When the contents of TL1 go from all 1's to all 0's, the interrupt TF1 is set and the contents of TH1 are transferred to TL1. The contents of TH1 remain unchanged. Since the contents of TH1 are under software control, the counter can be made to divide the count source by any number from 1 to 255 by means of the automatic reload of TH1 into TL1.

Timer 1 Mode 3

The following figure presents a functional overview of the Timer/Counter 1 Mode 3.

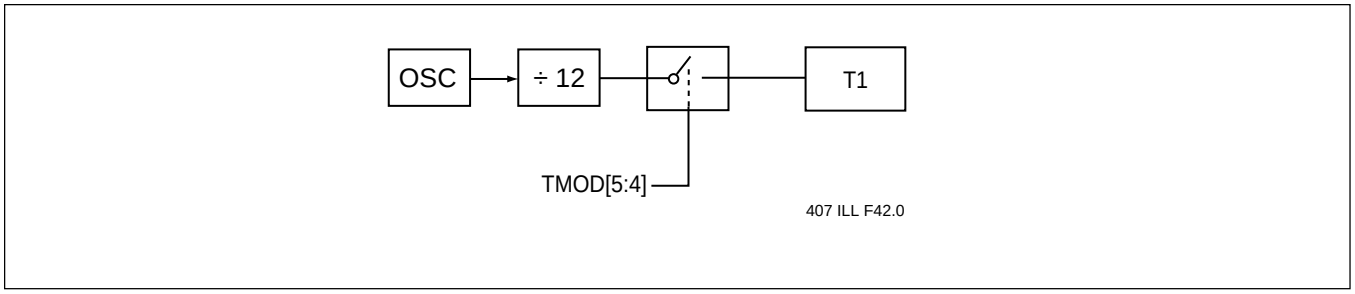


FIGURE 25: TIMER/COUNTER 1 MODE 3

In mode 3 operation, timer 1 is disabled, however it still holds its count.

When timer 0 is in Mode 3, timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial channel as a baud rate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

TMOD[5:4]

11 = off
00 = on
01 = on
10 = not available

Timer 2

Timer 2 is a 16-bit Timer/Counter. Timer 2 has three operating modes: capture, auto-load, and baud rate generator. The operating mode of Timer 2 is selected by the Timer/Counter 2 Control (T2CON) special function register shown below.

Timer/Counter 2 Control Register (T2CON)

Location	7	6	5	4	3	2	1	0	Reset Value
C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Timer 2 may either function as a timer or an event counter. The timer or counter function is selected by the Timer/Counter Select bit (C/T2#) of the T2CON register. If the C/T2# bit is set then Timer 2 will function as an internal timer. If the C/T2# bit is a 0 then Timer 2 will function as an external event counter, which is falling edge triggered.

The table below outlines the input combinations and the resulting operating modes for Timer 2.

RCLK+TCLK	CP/RL2#	TR2	Operating Mode
0	0	1	16-bit Auto-Reload
0	1	1	16-bit capture
1	X	1	Baud rate generator
X	X	0	(OFF)

The first two operating modes for Timer 2 are the capture and auto-reload mode. These two modes are selected by the Capture/Reload flag bit (CP/RL2#) of the T2CON register. When the CP/RL2# bit is set, Timer 2 is in the capture mode and captures will occur on negative transitions at the T2EX bit of the P1 register if the External Enable Flag bit (EXEN2) is set. When the CP/RL2# bit is cleared, Timer 2 is in the auto-reload mode and auto-reloads will occur either with Timer 2 overflows (the TF2 bit of the T2CON register is set) or negative transitions at the T2EX bit of the P1 register when the EXEN2 bit is set. When either the Receive Clock Flag bit (RCLK) or the Transmit Clock Flag bit (TCLK) is set, the CP/RL2# bit is ignored and Timer 2 is forced to auto-reload on Timer 2 overflow.

The following figure presents a functional overview of the Capture Mode.

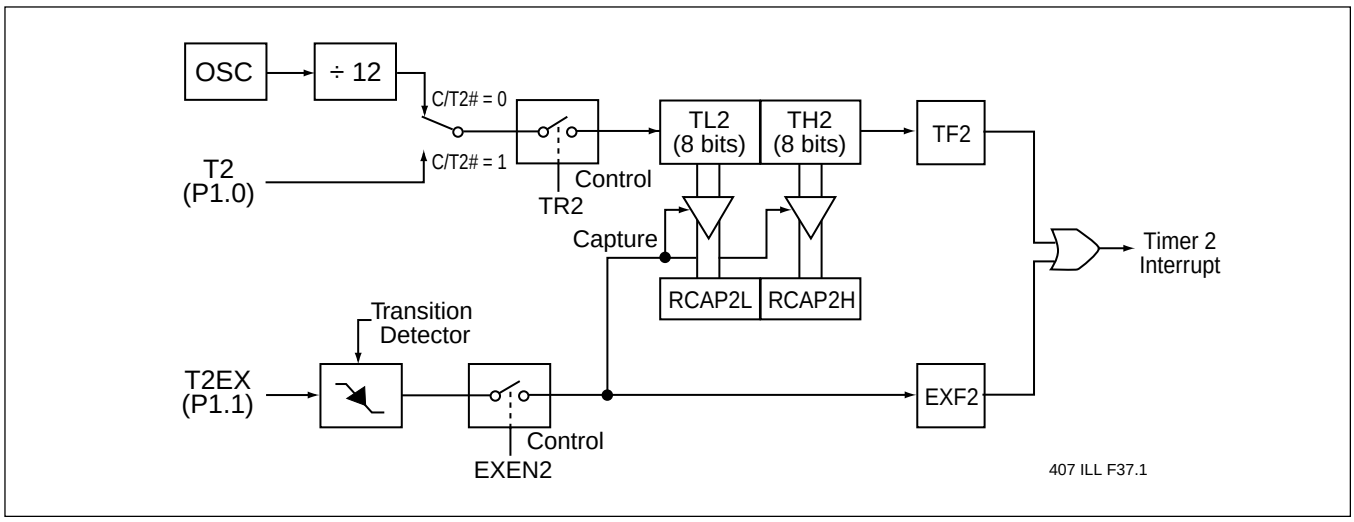


FIGURE 26: CAPTURE MODE

The capture mode provides two options, which are selected by the Timer 2 External Enable Flag bit (EXEN2) of the T2CON register. When the EXEN2 flag bit is cleared, then Timer 2 can be used as a 16-bit timer or counter and when Timer 2 overflows, it sets the Timer 2 Overflow flag bit (TF2) of the TCON register. Therefore, the TF2 bit may be used to generate an interrupt. When the EXEN2 flag bit is set, Timer 2 operates in the same manner as when the EXEN2 flag bit is cleared, however, a negative (1-to-0) transition on the T2EX bit of the P1 register causes the current value in the Timer 2 register (registers TL2 and TH2) to be captured into registers RCAP2L and RCAP2H, respectively. When the Timer 2 interrupt is enabled and the Timer 2 External Flag bit (EXF2) in the T2CON register is set, then the MCU will vector to the Timer 2 interrupt routine.

The following figure presents a functional overview of the Auto-Reload Mode.

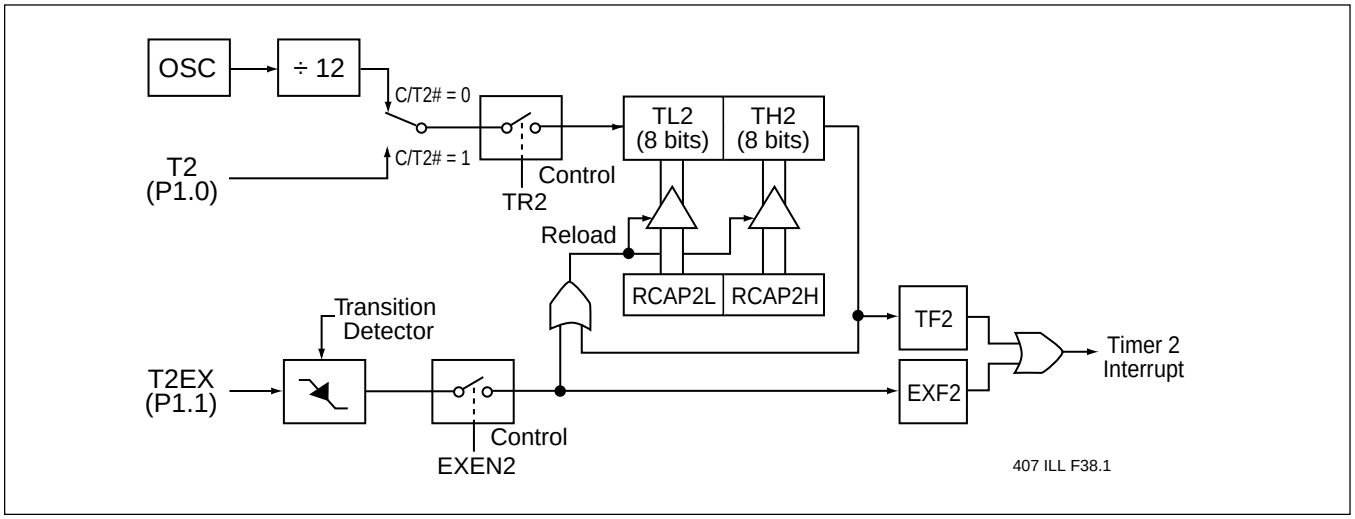


FIGURE 27: AUTO-RELOAD MODE

The auto-reload mode also provides two options, which are selected by the EXEN2 flag bit. When the EXEN2 flag bit is cleared and Timer 2 restarts its count, the TF2 bit is set and the Timer 2 registers (registers TL2 and TH2) are reloaded with the 8-bit values in the RCAP2L and RCAP2H registers, respectively. Note that the RCAP2L and RCAP2H registers are preset by software. When the EXEN2 flag bit is set, then Timer 2 still operates in the same manner as when the EXEN2 flag bit is cleared, however, a negative (1-to-0) transition on the T2EX bit of the P1 register causes the Timer 2 External flag bit (EXF2) bit to be set and the Timer 2 registers (registers TL2 and TH2) are reloaded with the 8-bit values in the RCAP2L and RCAP2H registers, respectively.

Preliminary

The following figure presents a functional overview of the Baud Rate Generator Mode.

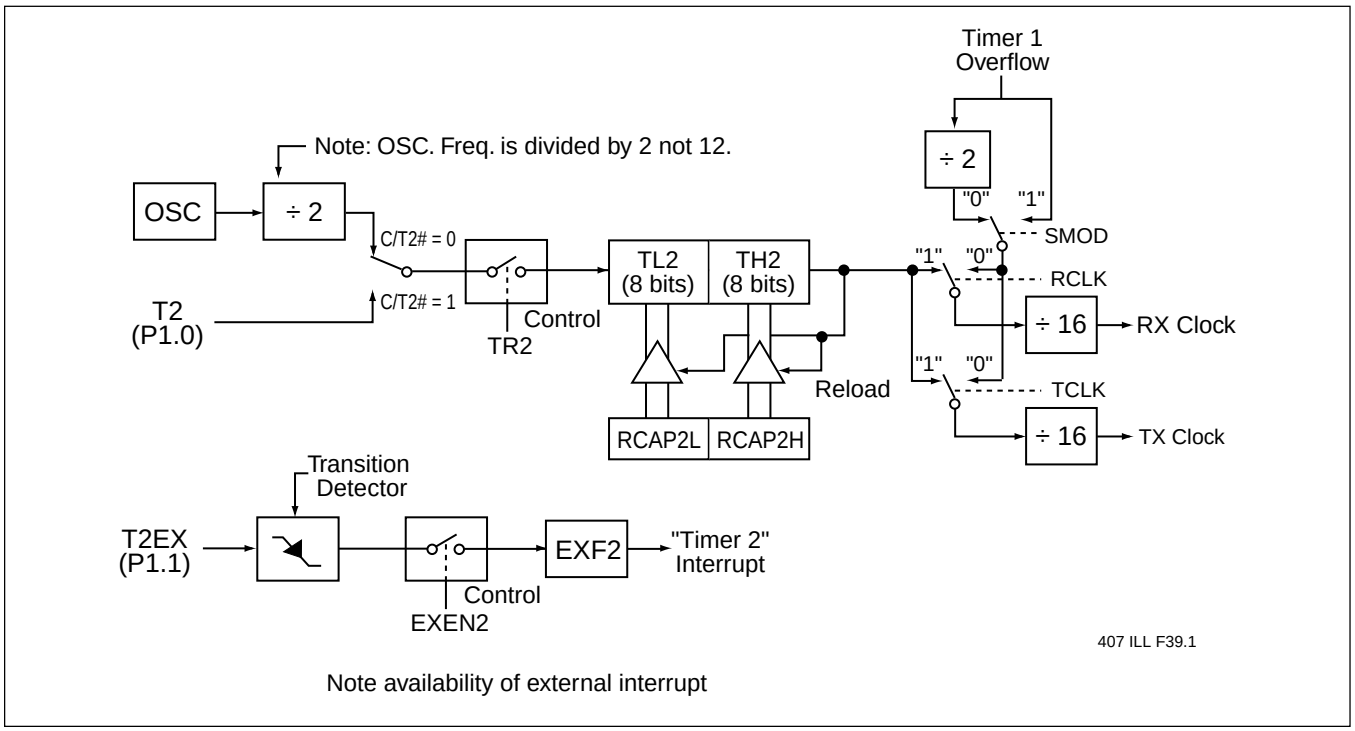


FIGURE 28: BAUD RATE GENERATOR MODE

The third operating mode for Timer 2 is the baud rate generator and is used in conjunction with the Serial Port. The baud rate generator mode is selected by the Transmit Clock flag bit (TCLK) and the Receive Clock flag bit (RCLK) of the T2CON register. When the RCLK bit is set, the Serial Port uses Timer 2 overflow pulses for its receive clock in mode 1 and mode 3. When the RCLK bit is cleared, the Serial Port uses Timer 1 overflow pulses for its receive clock in mode 1 and mode 3. When the TCLK bit is set, the Serial Port uses Timer 2 overflow pulses for its transmit clock in mode 1 and mode 3. When the TCLK bit is cleared, the Serial Port uses Timer 1 overflow pulses for its transmit clock in mode 1 and mode 3.

Parallel Ports

The SST89F54/58 have four 8-bit parallel ports, which are accessed by the Port 0 (P0), Port 1 (P1), Port 2 (P2) and Port 3 (P3) special function registers. Each port consists of an output driver, an input buffer and an 8-bit latch, which is located in the port's special function register. The four parallel ports are bi-directional ports and may be used for either input or output. Port 1, Port 2 and Port 3 are multifunctional and can be used for general I/O or for alternate functions. A port pin serving in its alternate function can not also be used for normal I/O, however the alternate functions can only be activated if the corresponding bit latch in the port special function register contains a 1. The function of the port pins also depends upon whether the system is in the External Host Mode or the IAP Mode. The pin assignments for the parallel ports in the External Host Mode and IAP Mode are outlined in the figures below. The following sections describe the operation and function of each of the parallel ports.

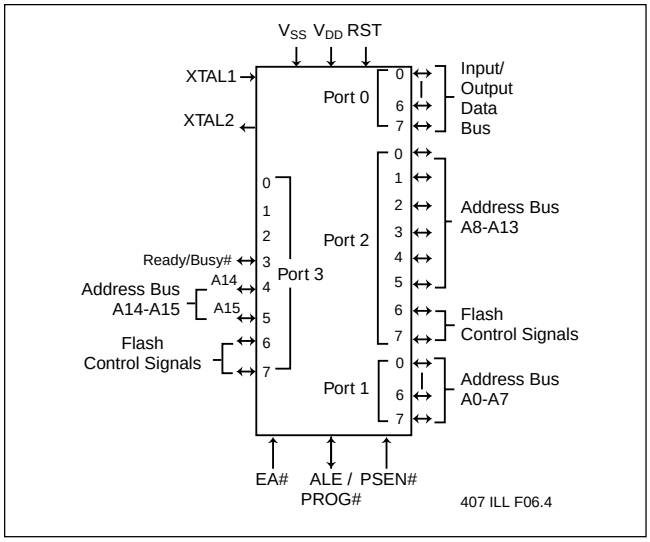


FIGURE 29: I/O PIN ASSIGNMENT FOR EXTERNAL HOST MODE

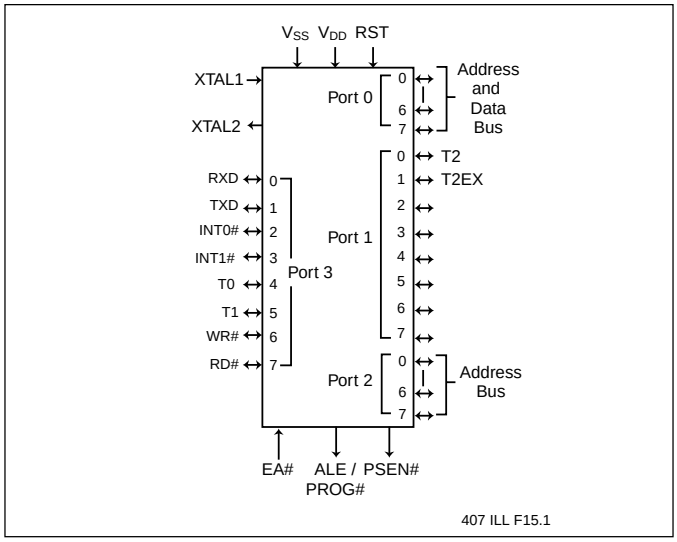


FIGURE 30: I/O PIN ASSIGNMENT FOR IN-APPLICATION PROGRAMMING MODE

Port 0

Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 consists of an output driver, an input buffer and an 8-bit latch, which is located in the Port 0 (P0) special function register. The P0 special function register is shown in the figure below.

Port 0 (P0)

Location	7	6	5	4	3	2	1	0	Reset Value
80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

When the SST89F54/58 are in the IAP Mode, Port 0 functions as a time-multiplexed address/data bus during external memory accesses, and as general purpose I/O port on devices which incorporate internal program memory. As an output port, each pin can sink eight LS TTL inputs. During external memory accesses, Port 0 contains the low byte (LSB) of the address bus when the ALE signal is at a logic level high and the data bus when the ALE signal is at a logic level low. When used as an I/O port, the port bits are open-drain and require pull-up resistors. Writing a logical 1 to any bit of the port places it in a high impedance mode, which is necessary if the pin is to be used as an input. Pull-ups are not required when accessing external memory.

When the SST89F54/58 are in the External Host Mode, Port 0 functions as a parallel data input and output port. Port 0 also receives the code bytes during Flash Memory programming, and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 consists of an output driver, an input buffer and an 8-bit latch, which is located in the Port 1 (P1) special function register. The P1 special function register is shown in the figure below.

Port 1 (P1)

Location	7	6	5	4	3	2	1	0	Reset Value
90h	-	-	-	-	-	-	T2EX	T2	FFh
							r/w	r/w	

Preliminary

When the SST89F54/58 are in the IAP Mode, Port 1 functions as a general purpose I/O port. The Port 1 output buffers can drive four LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. When a 1 is written to a Port 1 bit, either the processor port latch or the external circuitry can pull the port bit low. That is why it is necessary to assure that a logic 1 is written to the latch of any bit that is to be used as an input. In this way the input port pin's state is controlled by the external circuitry.

As inputs, Port 1 pins that are externally pulled low will source current due to the internal pull-ups. In addition, all the pins have an alternative function, which is listed in the figure above. Each of the functions is controlled by several other special function registers. The associated Port 1 latch bit must contain a logic one before the pin can be used in its alternate function.

When the SST89F54/58 are in the External Host Mode, Port 1 functions as a non-muxed lower order address bus for the internal flash memory (A0 - A7). Port 1 receives the low-order address bytes during FLASH MEMORY programming and program verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 consists of an output driver, an input buffer and an 8-bit latch, which is located in the Port 2 (P2) special function register. The P2 special function register is shown in the figure below.

Port 2 (P2)

Location	7	6	5	4	3	2	1	0	Reset Value
A0h	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

When the SST89F54/58 are in the IAP Mode, Port 2 functions as an address bus during external memory accesses, and as a general purpose I/O port on devices which incorporate internal program memory. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. When a 1 is written to a Port 2 bit, either the processor port latch or the external circuitry can pull the port bit low. That is why it is necessary to assure that a logic 1 is written to the latch of any bit that is to be used as an input. In this way the input port pin's state is controlled by the external circuitry.

As inputs, Port 2 pins that are externally pulled low will source current because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX@DPTR). In this application it uses strong internal pull-ups when emitting 1's. During accesses to external data memory that use 8-bit addresses (MOVX@Ri), Port 2 emits the contents of the P2 special function register.

When the device is in the External Host Mode, the first 6 bits of Port 2 pins (P2[0:5]) are assigned to be the non-muxed upper order address bus signals for the internal flash memory (A8 - A13). The two upper Port 2 pins (P2[6:7]) and are assigned as the control signal pins for FLASH MEMORY programming and program verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 consists of an output driver, an input buffer and an 8-bit latch, which is located in the Port 3 (P3) special function register. The P3 special function register is shown in the figure below.

Port 3 (P3)

Location	7	6	5	4	3	2	1	0	Reset Value
B0h	RD#	WR#	T1	T0	INT1#	INT0#	TXD0	RXD0	FFh
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

When the SST89F54/58 are in the IAP Mode, Port 3 functions as a general purpose I/O port. The Port 3 output buffers can drive four LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. When a 1 is written to a Port 3 bit, either the processor port latch or the external circuitry can pull the port bit low. That is why it is necessary to assure that a logic 1 is written to the latch of any bit that is to be used as an input. In this way the input port pin's state is controlled by the external circuitry.

As inputs, Port 3 pins that are externally pulled low will source current, due to the pull-ups. In addition, all the pins have an alternative function, which is listed in the figure above. Each of the functions is controlled by several other special function registers. The associated Port 3 latch bit must contain a logic one before the pin can be used in its alternate function.

When the SST89F54/58 is in the External Host Mode, Port 3 receives some control signals during Flash Memory programming and program verification. Two of the Port 3 pins are assigned to be part of the non-muxed upper order address bus signals for the internal flash memory (P3.4 as A14 and P3.5 as A15). Also the two upper order Port 3 pins (P3[6:7]) are assigned as the control signal pins.

Reading and Writing

An instruction that writes a value to a port latch is executed during the second phase of the sixth state of the final cycle of the instruction. However, port latches are sampled by their output buffers during the first phase of any clock period, consequently, during the second phase the output buffer contains the value it held during the first phase. The new value in the port latch is not sent to the output pin until the first phase of the first state of the next machine cycle.

Ports 1, 2 and 3 have internal pull-up resistors. When Ports 1, 2 and 3 are used as inputs, their pins are high when open-circuited and source current when pulled low by an external device. Port 0 is an open drain port and does not have pull-up resistors. When Port 0 is used as an input, its pins are floating. Instructions that read the port pins may either read the actual levels on the pins or the levels in the latch. In some cases the port is used to output a logic level high, but the external load attached to the port pin is of such a low resistance that the voltage on the pin is at a logic level low. Reading the latch would return a logic level high, however reading the pin would show the misinterpreted value as a logic level low.

The instructions that read the latch rather than the pin are called read-modify-write instructions. Read-modify-write instructions read a value from the port bit latch, possibly change it, and then rewrite the new value to the latch. The following table presents an overview of some of the instructions which may be used with the ports. Note that a more comprehensive description of the SST89F54/58 instruction set is presented in **Appendix B**.

ANL	logical AND e.g. ANL P1, A
ORL	logical OR e.g. ORL P1, A
XRL	logical XOR e.g. XRL P1, A
JBC	Jump if bit = 1 and clear bit e.g. JBC P1.1, LABEL
CPL	complement bit e.g. CPL P3.0
INC	increment e.g. INC P1
DEC	decrement e.g. DEC P1
DJNZ	decrement and jump if not zero e.g. DJNZ P1, LABEL
MOV, PX, Y, C	move carry bit to bit Y of Port X
CLR PX, Y	clear bit Y of Port X
SET PX, Y	set bit Y of Port X

Preliminary

Serial I/O (UART)

The SST89F54/58 Serial I/O port is a full duplex port that allows data to be transmitted and received simultaneously in hardware by the transmit and receive registers, respectively, while the software is performing other tasks. The Serial I/O port performs the function of an UART (Universal Asynchronous Receiver/Transmitter) chip. The transmit and receive registers are both located in the Serial Data Buffer (SBUF) special function register. Writing to the SBUF register loads the transmit register, and reading from the SBUF register obtains the contents of the receive register. The SBUF special function register is presented in the figure below.

Serial Data Buffer (SBUF)

Location	7	6	5	4	3	2	1	0	Reset Value
99h	SBUF[7:0]								Indeterminate
	r/w								

When reading or writing to the Serial port, a serial port interrupt is generated by the hardware in order to get the attention of the program. The receiver hardware is double buffered and can hold a received frame of data for reading while a second frame is being received. Double buffering allows the receiver interrupt service routine to be less time critical, however the stored frame must be read before reception of the second frame is complete or the first frame will be overwritten and lost.

The Serial I/O port has four modes of operation which are selected by the Serial Port Mode Specifier (SM0 and SM1) bits of the Serial Port Control (SCON) special function register. In all four modes, transmission is initiated by any instruction that uses the SBUF register as a destination register. Reception is initiated in mode 0 when the Receive Interrupt (RI) flag bit of the Serial Port Control (SCON) special function register is cleared and the Reception Enable/Disable (REN) bit of the SCON register is set. Reception is initiated in the other modes by the incoming start bit if the REN bit of the SCON register is set. The following sections describe the four modes of operation for the Serial I/O port.

Selecting the Mode

The operating mode for the Serial I/O port is selected by the Serial Port Mode Specifier (SM0 and SM1) bits of the Serial Port Control (SCON) special function register. The operating mode is selected by a combination of the SM0 and SM1 bits in the SCON register as outlined in the table below.

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

The SCON register functions as the serial port control and status register as well. The SCON register contains the ninth data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI). The SM2 bit is used in a multiprocessor system and enables the multiprocessor communication feature in modes 2 and 3. The SCON special function register is presented in the figure below.

Serial Port Control Register (SCON)

Location	7	6	5	4	3	2	1	0	Reset Value
98h	SM0	SM1	SM2	REN	TB8	RB8	T1	R1	00h
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

Baud Rates

The baud rate for mode 0 is set at 1/12 the oscillator frequency. The baud rate for mode 0 is given by the formula:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator frequency}}{12}$$

The baud rates for mode 1, mode 2 and mode 3 depends upon the Baud Rate Doubler Enable (SMOD) bit of the Power Control (PCON) special function register. The serial baud rate doubling function is enabled by setting the SMOD bit and disabled by clearing the SMOD bit. When the SMOD bit is cleared, which is the default setting for the SMOD bit, the baud rate for mode 1 and mode 3 is the baud rate that is defined by the baud rate generation equation and the baud rate for mode 2 is 1/64 the oscillator frequency. When the SMOD bit is set, the baud rate for mode 1 and mode 3 is double the baud rate defined by the baud rate generation equation and the baud rate for mode 2 is 1/32 the oscillator frequency. The baud rate for mode 2 is then given by the formula:

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD}}{64} \times \text{Oscillator frequency}$$

The baud rates for mode 1 and mode 3 can be determined by Timer 1, Timer 2 or by both Timers 1 and 2, one timer for transmit and the other for receive.

Timer/Counter 1 as a Baud Rate Generator

When Timer/Counter 1 is used as a baud rate generator, the baud rates for Serial Port mode 1 and mode 3 are determined by the Timer/Counter 1 overflow rate and the value of the Baud Rate Doubler Enable (SMOD) bit of the Power Control (PCON) special function register. The general equation for the baud rate for mode 1 and mode 3 is given by the formula:

$$\text{Mode 1 and Mode 3 Baud Rate} = \frac{2^{SMOD}}{32} \times \text{Timer 1 overflow rate}$$

Timer/Counter 1 may be used in any of its modes, however, its most common implementation is as a timer in mode 2, the auto-reload mode. When the Timer 1 is configured to run as a 16-bit timer with the Timer 1 interrupt enabled, Timer 1 may be used to produce low baud rates. When Timer 1 is configured to run in mode 2, the baud rate for mode 1 and mode 3 is given by the formula:

$$\text{Mode 1 and Mode 3 Baud Rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator frequency}}{12 \times [256-(TH1)]}$$

The term (TH1) represents the contents of the Timer 1 MSB (TH1) special function register. Note that since the value stored in the TH1 register must, by definition, be an integer number, the oscillator frequency must be an integer multiple of the baud rate in order to obtain an exact baud rate using the above expression.

Timer/Counter 2 as a Baud Rate Generator

Timer/Counter 2 is selected as a baud rate generator by setting the Transmit Clock (TCLK) and/or the Receive Clock (RCLK) flag bits in the Timer/Counter 2 Control (T2CON) special function register. Therefore, the baud rates for transmit and receive can be simultaneously different.

The Timer/Counter 2 baud rate generator mode is similar to the auto-reload mode, in that a rollover in the Timer 2 MSB (TH2) special function register causes the Timer 2 registers to be reloaded with the 16-bit value in the Timer 2 Capture MSB (RCAP2H) and Timer 2 Capture LSB (RCAP2L) special function registers. The RCAP2L and RCAP2H registers are set by software. The general equation for the baud rate for mode 1 and mode 3 is given by the formula:

$$\text{Modes 1 and 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

Preliminary

Timer/Counter 2 can be configured for either timer or counter operation, however, its most common implementation is as a timer. As a baud rate generator, Timer 2 increments during every state at ½ the oscillator frequency. When Timer/Counter 2 is configured as a timer, the baud rate for mode 1 and mode 3 is given by the formula:

$$\text{Mode 1 and Mode 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

The term (RCAP2H, RCAP2L) represents the contents of the RCAP2H and RCAP2L special function registers.

Note that a rollover in TH2 will not generate an interrupt since it does not set the Timer 2 Overflow (TF2) flag bit of the T2CON special function register. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. If the Timer 2 External Enable (EXEN2) flag bit of the T2CON register is set, a 1-to-0 transition in the Timer 2 Capture/Reload Trigger (T2EX) bit of the Port 1 (P1) special function register will set the Timer 2 External (EXF2) flag bit of the T2CON special function register, however it will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2).

Also note that when Timer 2 is operating in the timer function in the baud rate generator mode, the TH2 and TL2 special function registers should not be accessed. While Timer 2 is in the baud rate generator mode the Timer 2 increments during every state time and the results of a read or write to these registers may not be accurate. The RCAP2H and RCAP2L special function registers may be read, however they should not be written to, since a write might overlap a reload and cause write or reload errors to occur. Therefore, it is necessary to turn the Timer 2 off before accessing the TH2, TL2, RCAP2H and RCAP2L registers.

Mode 0

Mode 0 is a half-duplex synchronous mode. Data is sent and received through the Serial Port Receive (RXD) bit of the Port 3 (P3) special function register in 8-bit frames, LSB first. Data may not be sent and received simultaneously in mode 0 as in the other modes. The baud rate for mode 0 is fixed at 1/12 the oscillator frequency. In mode 0 the Serial Port functions as a shift register. The shift clock, which is the same frequency as the baud rate, is sent out the Serial Port Transmit (TXD) bit of the Port 3 (P3) special function register during both transmission and reception and is used to synchronize the receiver with the sender. A shift clock edge occurs during the valid state of each data bit. Note that TXD and RXD are alternate functions of the Port 3 pins P3.1 and P3.0, respectively.

Transmission of data is initiated by any instruction that uses the Serial Data Buffer (SBUF) special function register as a destination register. A Serial Port transmission sends the output of the transmit shift register to the RXD pin and sends the output of the shift clock to the TXD pin. The shift clock is low during state 3, state 4 and state 5 of every machine cycle in which the Serial Port is transmitting data. A complete machine cycle will elapse between the instruction that initiates the Serial Port transmission and the actual transmission.

When a transmission is initiated, the contents of the Serial Data Buffer (SBUF) special function register are loaded into the transmit shift register and a 1 is loaded into the ninth bit position of the transmit shift register, during the second phase of the sixth state. The contents of the transmit shift register are shifted to the right one position during each machine cycle. As data bits shift out to the right, zeroes are shifted in from the left. When the MSB of the data byte finally reaches the output position of the shift register, the 1 that was initially loaded into the ninth bit position is located just to the left of the MSB and all positions to the left of that contain zeroes. This condition flags one last shift and then deactivates the transmit shift register and sets the Transmit Interrupt (TI) flag bit of the Serial Port Control (SCON) special function register. The last shift occurs at the first phase of the first state of the tenth machine cycle after the Serial Port transmission was activated.

Reception of data is initiated by any instruction that uses the Serial Data Buffer (SBUF) special function register as a source register when the Reception Enable/Disable (REN) bit of the SCON register is set and the Receive Interrupt (RI) flag bit is cleared. A Serial Port reception reads the contents of the receive shift register and sends the output of the shift clock to the TXD pin. The shift clock makes transitions at the first phase of the third state and the first phase of the sixth state of every machine cycle in which the Serial Port is receiving data. A complete machine cycle will elapse between the instruction that initiates the Serial Port reception and the actual reception.

When a reception is initiated the bits 11111110 are written to the receive shift register during the second phase of the sixth state. The contents of the receive shift register are shifted to the left one position during the second phase of the sixth state of each machine cycle. As data bits shift in from the right, 1's are shifted out to the left.

The data bits that shift in from the right are the values that were sampled at the RXD pin during the second phase of the fifth state of the same machine cycle. When the 0 that was initially loaded into the LSB finally reaches the leftmost position in the shift register, the 0 that was initially loaded into the LSB is located at the MSB position and all positions to the right of that contain data bits. This condition flags one last shift and then loads the Serial Data Buffer (SBUF) special function register. The last shift occurs at the first phase of the first state of the tenth machine cycle after the Serial Port reception was activated.

Mode 1

Mode 1 is a full duplex asynchronous mode. Data is sent out through the Serial Port Transmit (TXD) bit of the Port 3 (P3) special function register in 8-bit frames, LSB first. Data is received through the Serial Port Receive (RXD) bit of the Port 3 (P3) special function register in 8-bit frames, LSB first. A complete frame consists of a start bit (always a 0), followed by 8 data bits from the Serial Data Buffer (SBUF) special function register, followed by a stop bit (always a 1). The start and stop bits are added by hardware. The 8-bit data byte in the SBUF register is read or written to by software. The baud rate (bit time) is variable and can be obtained by using Timer/Counter 1 or Timer/Counter 2 or both (one for transmit and the other for receive) as a baud rate generator. Note that TXD and RXD are alternate functions of the Port 3 pins P3.1 and P3.0, respectively.

Transmission of data is initiated by any instruction that uses the SBUF register as a destination register. A Serial Port transmission sends the output of the transmit shift register to the TXD pin. A complete machine cycle will elapse between the instruction that initiates the Serial Port transmission and the actual transmission.

When a transmission is initiated, the contents of the SBUF special function register are loaded into the transmit shift register and a 1 is loaded into the ninth bit position of the transmit shift register. The first transmission of data, which loads the start bit to the TXD pin, occurs during the first phase of the first state of the machine cycle following the next rollover in the divide-by-16 counter. One bit time later, the first data bit of the transmit shift register is loaded to the TXD pin. The contents of the transmit shift register are shifted to the right one position during each shift pulse. As data bits shift out to the right, zeroes are clocked in from the left. When the MSB of the data byte finally reaches the output position of the shift register, the 1 that was initially loaded into the ninth bit position is located just to the left of the MSB and all positions to the left of that contain zeroes. This condition flags one last shift and then deactivates the transmit shift register and sets the Transmit Interrupt (TI) flag bit of the Serial Port Control (SCON) special function register. The last shift occurs at the tenth divide-by-16 rollover after the Serial Port transmission was activated.

Reception of data is initiated by any instruction that uses the Serial Data Buffer (SBUF) special function register as a source register when a 1-to-0 transition is detected at the RXD pin and the Reception Enable/Disable (REN) bit of the Serial Port Control (SCON) special function register is set. In order to check for a transition, the RXD pin is sampled at a rate of 16 times the baud rate that has been established for mode 1. When a transition is detected, the divide-by-16 counter is immediately reset and 1FFh is written to the receive shift register. The divide-by-16 counter is reset in order to align its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. During the seventh, eighth, and ninth counter states of each bit time, the value at the RXD pin is sampled. The value that is accepted is the value that was seen in at least two of the three samples. If the value accepted during the first bit time is not 0, the start bit, then the receive circuits are reset and the another 1-to-0 transition must be detected. This is to provide rejection of false start bits. If the start bit is valid, it is shifted into the input shift register, and reception of the frame will proceed.

As data bits shift in from the right during each bit time, 1's are shifted out to the left. The data bits that shift in from the right are the values that were sampled at the RXD pin during the seventh, eighth and ninth counter states of each bit time. When the start bit finally reaches the leftmost position in the shift register, which in mode 1 is a 9-bit register, all positions to the right contain data bits. This condition flags one last shift and then loads the SBUF register with the 8 data bits and the RB8 bit of the SCON register with the stop bit, and sets the Receive Interrupt (RI) flag bit. The signal to load SBUF and RB8 and to set RI, will only be generated if the RI bit is cleared and either the SM2 bit of the SCON register or the RB8 bit is set during the final shift pulse. If either of these two conditions is not met, the received frame is lost.

Mode 2

Mode 2 is a full duplex asynchronous mode. Data is sent out through the Serial Port Transmit (TXD) bit of the Port 3 (P3) special function register in 11-bit frames, LSB first. Data is received through the Serial Port Receive (RXD) bit of the Port 3 (P3) special function register in 11-bit frames, LSB first. A complete frame consists of a start bit (always a 0), followed by 8 data bits from the Serial Data Buffer (SBUF) special function register, a programmable ninth data bit, and a stop bit (always a 1). When the Serial Port is transmitting data, the ninth bit is obtained from the TB8 bit in the Serial Port Control (SCON) special function register and can be programmed with a 0 or 1. When the Serial Port is receiving data, the ninth bit can be read from RB8 in SCON, while the stop bit is ignored. The ninth data bit is used as a parity bit for the 8-bit data byte. The start and stop bits are added by hardware. The 8-bit data byte in the SBUF register is read or written to by software. The baud rate is either 1/32 or 1/64 the oscillator frequency, depending on the value of the Baud Rate Doubler Enable (SMOD) bit of the Power Control (PCON) special function register. If the SMOD bit is set, the baud rate is 1/32 the oscillator frequency and if the SMOD bit is cleared, the baud rate is 1/64 the oscillator frequency. Note that TXD and RXD are alternate functions of the Port 3 pins P3.1 and P3.0, respectively.

Transmission of data is initiated by any instruction that uses the SBUF register as a destination register. A Serial Port transmission sends the output of the transmit shift register to the TXD pin. A complete machine cycle will elapse between the instruction that initiates the Serial Port transmission and the actual transmission.

When a transmission is initiated, the contents of the SBUF special function register is loaded into the transmit shift register and TB8 is loaded into the ninth bit position of the transmit shift register. One bit time after the transmission is initiated, the first shift pulse occurs and clocks a 1 (the stop bit) into the ninth bit position of the shift register. The first transmission of data, which loads the start bit to the TXD pin, occurs during the first phase of the first state of the machine cycle following the next rollover in the divide-by-16 counter. The contents of the transmit shift register are shifted to the right one position during each shift pulse. As data bits shift out to the right, zeroes are clocked in from the left. When the MSB of the data byte finally reaches the output position of the shift register, the 1 that was initially loaded into the ninth bit position is located just to the left of the MSB and all positions to the left of that contain zeroes. This condition flags one last shift and then deactivates the transmit shift register and sets the Transmit Interrupt (TI) flag bit of the Serial Port Control (SCON) special function register. The last shift occurs at the tenth divide-by-16 rollover after the Serial Port transmission was activated.

Reception of data is initiated by any instruction that uses the Serial Data Buffer (SBUF) special function register as a source register when a 1-to-0 transition is detected at the RXD pin and the Reception Enable/Disable (REN) bit of the SCON register is set. In order to check for a transition, the RXD pin is sampled at a rate of 16 times the baud rate that has been established for mode 1. When a transition is detected, the divide-by-16 counter is immediately reset and 1FFh is written to the receive shift register. The divide-by-16 counter is reset in order to align its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. During the seventh, eighth, and ninth counter states of each bit time, the value at the RXD pin is sampled. The value that is accepted is the value that was seen in at least two of the three samples. If the value accepted during the first bit time is not 0, the start bit, then the receive circuits are reset and the another 1-to-0 transition must be detected. This is to provide rejection of false start bits. If the start bit is valid, it is shifted into the input shift register, and reception of the frame will proceed.

As data bits shift in from the right during each bit time, 1's are shifted out to the left. The data bits that shift in from the right are the values that were sampled at the RXD pin during the seventh, eighth and ninth counter states of each bit time. When the start bit finally reaches the leftmost position in the shift register, which in mode 2 is an 11-bit register, all positions to the right contain data bits. This condition flags one last shift and then loads the SBUF register with the 8 data bits and the RB8 bit of the SCON register with the stop bit, and sets the Receive Interrupt (RI) flag bit. The signal to load SBUF and RB8 and to set RI, will only be generated if the RI bit is cleared and either the SM2 bit of the SCON register or the RB8 bit is set during the final shift pulse. If either of these two conditions is not met, the received frame is lost.

Mode 3

Mode 3 is a full duplex asynchronous mode. Data is sent out through the Serial Port Transmit (TXD) bit of the Port 3 (P3) special function register in 11-bit frames, LSB first. Data is received through the Serial Port Receive (RXD) bit of the Port 3 (P3) special function register in 11-bit frames, LSB first. A complete frame consists of a start bit (always a 0), followed by 8 data bits from the Serial Data Buffer (SBUF) special function register, a programmable ninth data bit, and a stop bit (always a 1). When the Serial Port is transmitting data, the ninth bit is obtained from the TB8 bit in the Serial Port Control (SCON) special function register and can be programmed with a 0 or 1.

When the Serial Port is receiving data, the ninth bit can be read from RB8 in SCON, while the stop bit is ignored. The ninth data bit is used as a parity bit for the 8-bit data. The start and stop bits are added by hardware. The 8-bit data byte in the SBUF register is read or written to by software. The baud rate (bit time) is variable and can be obtained by using Timer/Counter 1 or Timer/Counter 2 or both (one for transmit and the other for receive) as a baud rate generator. Note that TXD and RXD are alternate functions of the Port 3 pins P3.1 and P3.0, respectively.

Transmission of data is initiated by any instruction that uses the SBUF register as a destination register. A Serial Port transmission sends the output of the transmit shift register to the TXD pin. A complete machine cycle will elapse between the instruction that initiates the Serial Port transmission and the actual transmission.

When a transmission is initiated, the contents of the SBUF special function register are loaded into the transmit shift register and TB8 is loaded into the ninth bit position of the transmit shift register. One bit time after the transmission is initiated, the first shift pulse occurs and clocks a 1 (the stop bit) into the ninth bit position of the shift register. The first transmission of data, which loads the start bit to the TXD pin, occurs during the first phase of the first state of the machine cycle following the next rollover in the divide-by-16 counter. The contents of the transmit shift register are shifted to the right one position during each shift pulse. As data bits shift out to the right, zeroes are clocked in from the left. When the MSB of the data byte finally reaches the output position of the shift register, the 1 that was initially loaded into the ninth bit position is located just to the left of the MSB and all positions to the left of that contain zeroes. This condition flags one last shift and then deactivates the transmit shift register and sets the Transmit Interrupt (TI) flag bit of the Serial Port Control (SCON) special function register. The last shift occurs at the tenth divide-by-16 rollover after the Serial Port transmission was activated.

Reception of data is initiated by any instruction that uses the Serial Data Buffer (SBUF) special function register as a source register when a 1-to-0 transition is detected at the RXD pin and the Reception Enable/Disable (REN) bit of the SCON register is set. In order to check for a transition, the RXD pin is sampled at a rate of 16 times the baud rate that has been established for mode 1. When a transition is detected, the divide-by-16 counter is immediately reset and 1FFh is written to the receive shift register. The divide-by-16 counter is reset in order to align its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. During the seventh, eighth, and ninth counter states of each bit time, the value at the RXD pin is sampled. The value that is accepted is the value that was seen in at least two of the three samples. If the value accepted during the first bit time is not 0, the start bit, then the receive circuits are reset and the another 1-to-0 transition must be detected. This is to provide rejection of false start bits. If the start bit is valid, it is shifted into the input shift register, and reception of the frame will proceed.

As data bits shift in from the right during each bit time, 1's are shifted out to the left. The data bits that shift in from the right are the values that were sampled at the RXD pin during the seventh, eighth and ninth counter states of each bit time. When the start bit finally reaches the leftmost position in the shift register, which in mode 3 is an 11-bit register, all positions to the right contain data bits. This condition flags one last shift and then loads the SBUF register with the 8 data bits and the RB8 bit of the SCON register with the stop bit, and sets the Receive Interrupt (RI) flag bit. The signal to load SBUF and RB8 and to set RI, will only be generated if the RI bit is cleared and either the SM2 bit of the SCON register or the RB8 bit is set during the final shift pulse. If either of these two conditions is not met, the received frame is lost.

Preliminary

Multiprocessor Systems

Mode 2 and mode 3 may be used in multiprocessor systems. In mode 2 and mode 3, the frame is 11 bits, which consist of a start bit, 9 data bits and a stop bit. The ninth data bit is stored in the RB8 bit of the Serial Port Control (SCON) special function register. The Serial Port can be programmed to generate a serial port interrupt when the stop bit is received and the RB8 bit of the SCON register is set.

When the master processor transmits a block of data to one or more of its slaves, it first sends out an address byte, which identifies the target slave. An address byte is different from a data byte in that the ninth bit is set for an address byte and cleared for a data byte. This multiprocessor system feature is enabled by setting the SM2 bit of the SCON register. While the SM2 bit is set, the slaves are not interrupted by a data byte, however, an address byte will interrupt all the slaves, so that each slave can examine the received byte and determine whether it is being addressed. The addressed slave then clears its SM2 bit and receives the data bytes. The slaves that are not addressed leave their SM2 bits set and do not receive the data bytes.

SERIAL PORT SET-UP TABLE

MODE	SCON	SM2 VARIATION
0	10h	
1	50h	Single Processor
2	90h	Environment
3	D0h	(SM2 = 0)
0	NA	
1	70h	Multiprocessor
2	B0h	Environment
3	F0h	(SM2 = 1)

Interrupt System

The SST89F54/58 provide 6 interrupt sources, which include two external interrupts (INT0# and INT1#), three Timer/Counter interrupts (TF0, TF1 and TF2), and one from the serial port (SI or TI). The Interrupt Enable (IE) special function register, the source of the interrupts, is presented in the figure below. A description of each of the interrupt types is presented in the following sections. Note that each of the bits that generate the following interrupts may be set or cleared by software with the same result as setting or clearing the bits through hardware. Therefore, interrupts may be generated or canceled by software. Also note that individual interrupts can be enabled or disabled by setting or clearing individual bits of the IE register. The SST89F54/58 also contain a global enable bit which allows all of the interrupts to be enabled or disabled by setting or clearing the EA bit of the IE register.

Interrupt Enable (IE)

Location	7	6	5	4	3	2	1	0	Reset Value
A8h	EA	–	ET2	ES0	ET1	EX1	ET0	EX0	00h
	r/w		r/w	r/w	r/w	r/w	r/w	r/w	

The SST89F54/58 support 2 external interrupt sources, INT0# and INT1#. INT0# and INT1# may be either level-activated (low level triggered) or transition-activated, based upon the Interrupt 0 Type (IT0) and Interrupt 1 Type (IT1) control bits of the Timer/Counter Control (TCON) register. The IT0 and IT1 control bits of the TCON register may be set or cleared by software in order to specify a falling edge or a low level triggered external interrupt, respectively. The flags that generate the INT0# and INT1# interrupts are the External Interrupt 0 (IE0) and External Interrupt 1 (IE1) Edge flag bits of the TCON register. The IE0 flag bit is set by hardware when an external interrupt edge is detected on the INTO# line and the IE1 flag bit is set by hardware when an external interrupt edge is detected on the INT1# line. The IE0 and IE1 flag bits are cleared by hardware when the external interrupt is processed (the MCU vectors to the interrupt service routine).

The Timer 0 and Timer 1 interrupts are generated by the Timer 0 Overflow (TF0) and Timer 1 Overflow (TF1) flag bits of the TCON register. The TF0 bit is set by hardware when Timer 0 overflows and the TF1 bit is set by hardware when Timer 1 overflows. When a Timer 0 interrupt is generated, the TF0 bit is cleared by hardware while the interrupt is processed (the MCU vectors to the interrupt service routine). When a Timer 1 interrupt is generated, the TF1 bit is cleared by hardware when the interrupt is processed (the MCU vectors to the interrupt service routine).

The Timer/Counter 2 interrupt is generated when either the Timer 2 Overflow (TF2) or the Timer 2 External (EXF2) flag bits of the Timer/Counter 2 Control (T2CON) register are active. The TF2 flag bit is set by hardware only when the RCLK and the TCLK bits of the T2CON register are cleared. The EXF2 flag bit is set by hardware when either a capture or reload is caused by a negative transition on T2EX, and the Timer 2 External Enable (EXEN2) flag bit of the T2CON register is set. When the Timer 2 interrupt is enabled and the EXF2 bit of the T2CON register is set, the MCU will vector to the Timer 2 interrupt service routine. Neither the TF2 nor the EXF2 bits are cleared by hardware and, therefore, must be cleared by software.

The serial port interrupt is generated when either the Receive Interrupt (RI) or Transmit (TI) flag bits of the Serial Port Control (SCON) register are active. The TI flag bit is set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. The RI flag bit is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, except the SM2 variation. Neither the RI nor TI bits are cleared by hardware when the interrupt is processed (the MCU vectors to the interrupt service routine). Normally, the service routine will have to determine which bit generated the interrupt and that bit will have to be cleared by software.

Preliminary

Interrupt Priority Levels

Individual interrupts can be programmed as a low-priority or a high-priority interrupt by setting or clearing the corresponding bit in the Interrupt Priority (IP) special function register. A 0 value is designated a low priority and a 1 value is the high priority. The IP register table is presented in the figure below.

Interrupt Priority Register (IP)

Location	7	6	5	4	3	2	1	0	Reset Value
B8h	–	–	PT2	PS	PT1	PX1	PT0	PX0	xx000000
			r/w	r/w	r/w	r/w	r/w	r/w	

A low-priority interrupt can be interrupted by a higher priority interrupt, however, it may not be interrupted by an interrupt of the same priority. A high-priority interrupt may not be interrupted by a low-priority interrupt or an interrupt of the same priority. When two interrupt requests of different priority levels are received simultaneously, the request with the higher priority level will be serviced first. When two interrupt requests of the same priority level are received simultaneously, an internal polling sequence will be used to determine which request will be serviced first. Therefore, within each priority level there is a second priority structure, which is used by the polling sequence. This secondary priority structure is organized as follows:

	Source	Priority Within the Primary Priority Level
1)	IE0	(highest)
2)	TF0	
3)	IE1	
4)	TF1	
5)	RI + TI	
6)	TF2 + EXF2	(lowest)

Interrupt Execution

Every interrupt is associated with a fixed program memory, starting at address 0003H, which contains the corresponding interrupt vector address. Enabling an interrupt causes the MCU to jump to the interrupt vector address, which is the start of the interrupt service routine. The six interrupt vector addresses corresponding to the six interrupt sources are stored in program memory addresses 03h to 2Bh in the on-chip primary flash memory block and each interrupt vector is 8 bytes. The following figure outlines the data memory allocated for interrupt vectors.

Interrupt Source	Vector Address
IE0	0003h
TF0	000Bh
IE1	0013h
TF1	001Bh
RI + TI	0023h
TF2 + EXF2	002Bh

The SST89F54/58 machine cycle has six states with two phases each. The interrupt flags are sampled at the 2nd phase of the 5th state of each machine cycle. The samples are polled during the next machine cycle. The interrupt flag must have been set before and during the 5th state and 2nd phase of the previous machine cycle in order for the interrupt to be recognized. When an interrupt is detected by the polling cycle, the interrupt system will generate an LCALL to the appropriate interrupt service routine. However, an LCALL may be blocked under one of the following conditions:

1) The MCU is already busy with an interrupt of equal or higher priority.

2) The MCU is busy with the current instruction.

3) The current instruction is RETI.

4) The current instruction is writing to the IE or IP special function registers.

The above conditions ensure that the interrupt does not interfere with the current interrupt, and allows the MCU to complete the current instruction. Note that if the interrupt flag was blocked while it was active, then it must still be active after the blocking condition is removed, in order for the interrupt to be recognized. The polling sequence will not remember the status of the flags from the previous polling sequence

When the interrupt occurs, the MCU acknowledges the request by executing an LCALL (generated by hardware) to the appropriate servicing routine. The LCALL pushes the contents of the Program Counter (PC) onto the stack, however, it does not push the contents of the PSW. The LCALL then loads the PC with the vector address of the corresponding Interrupt Service Routine.

The MCU fetches its next instruction from this address and continues until it encounters the RETI instruction. The RETI instruction is used to end the ISR and prompts the MCU to pop the first two bytes from the stack and reloads the PC with these two bytes, which were the contents of the PC register before the ISR was executed. The program then continues from where it was interrupted.

Preliminary

Security Lock

The Security feature protects against software piracy and prevents the contents of the flash from being read by unauthorized parties. It also protects against code corruption resulting from accidental erasing and programming to the internal flash memory locations. The security byte is located in the highest address location, FFFFh, of the SST89F54/58 program memory space.

When the security lock is activated, the MOVC instructions executed from external program memory or unlocked flash memory are disabled from fetching code bytes from locked memory blocks (See Table 5). The security lock can either "hard" lock both flash memory blocks or just "hard" lock the secondary flash memory block (Block 1) independently. When the memory blocks are locked, the following commands are not allowed on the locked flash memory blocks:

- Sector Erase
- Block Erase
- Program Erase
- Burst Program
- Byte Verify

Both memory blocks may be "soft" locked, allowing In-Application Programming. This feature allows the device to enter IAP Mode executing from internal memory, but inhibits the device from entering IAP Mode executing from external memory or External Host Mode. Table 9 lists the security lock options and commands allowed for each option.

Activation and deactivation of the security lock

For both External Host and In-Application Programming modes, the security byte, 55h, F5h or 05h, must first be programmed into the address location FFFFh using the Byte or Burst Program operation. After the security byte has been programmed into address location FFFFh, the security lock is activated following a successful system reset.

To deactivate the security lock, the security byte at location FFFFh is programmed with a value of, FFh, via the Chip Erase operation. The default value of the security byte is FFh.

TABLE 5: INTERNAL AND EXTERNAL PROGRAM MEMORY ACCESS WITH SECURITY LOCK ACTIVATED

MOVC INSTRUCTIONS EXECUTED FROM	ACCESS TO LOCKED PROGRAM MEMORY	ACCESS TO UNLOCKED OR EXTERNAL PROGRAM MEMORY
locked program memory	YES	YES
unlocked or external program memory	NO	YES

407 PGM T6.2

TABLE 6: SECURITY LOCK OPTIONS

Sec Byte	SFCF [6:5]	EA#	blk Sel	Block Erase	Sector Erase	Byte Program	Burst Program	Byte Verify	Description
FF	00	X	X	Y	Y	Y	Y	Y	no lock, (default)
55	11	X	X	N	N	N	N	N	both locked (Hard Lock)
F5	01	X	0	Y	Y	Y	Y	Y	only block 1 (4KB) is locked
		X	1	N	N	N	N	N	
05	10	0	0/1	N	N	N	N	N	blocks are accessible only through In-Application Programming (Soft Lock)
		1	0/1	Y	Y	Y	Y	Y	

407 PGM T5.6

NOTE: **SecByte** = Value of Security Byte at location FFFFh; **SFCF(6:5)** = Bit 5 and 6 of SFCF register; **EA#** = Ext. Access enable input pin: 1 – running code from internal memory, 0 – running code from external enable; **blkSel** = Block Select signal (internal): 1 – block 1 (4Kx8), 0 – block 0 (32Kx8); **X** = don't care; **Y** = command allowed; **N** = command not allowed.

In-Application Programming Mode

The security lock option, 05h, "soft" locks both flash memory blocks. This lock option allows the user to update the code in the locked flash memory blocks under a pre-determined secure environment. When both flash memory blocks are "soft" locked, the software code executing from one internal flash memory block can perform In-Application Programming on the other block. In other words, code residing in Block 1 may program to Block 0 and vice versa. The following IAP Mode commands issued through the command mailbox register, SFCM, and executed from the internal program memory can be operated on both Block 0 and Block 1: Block Erase, Sector Erase, Byte Program, Burst Program and Byte Verify .

The security byte, 55h (both blocks "hard" locked), prohibits In-Application Programming (IAP) to the flash memory blocks. Only the Chip Erase operation will erase both blocks including the security byte. If the security byte, F5h (only Block 1 is locked), In-Application Programming is only allowed in Block 0.

A CHIP ERASE operation can deactivate, or change the level of, the security lock after it is set. Chip Erase will set the security byte to the value of FFh. The In-Application Programming operation with program code execution from either internal flash memory or external program code storage can write a new lock option to the security byte, and the new security lock will activate on the next system reset.

External Host Mode

If the security lock is activated, the following External Host Mode commands are not allowed on the locked flash memory blocks: Sector Erase, Block Erase, Byte Program, Burst Program, and Byte Verify. Only the Chip Erase operation can deactivate, or change level of, the security lock after it is set. CHIP ERASE will set the security byte to the value of FFh.

Watchdog Timer

The SST89F54/58 offer an enhanced programmable watchdog timer for fail safe protection against software hang. When a software hang or a hardware based software error occurs, the Watchdog Timer unit provides an automatic recovery of the system by means of an internally generated watchdog reset. The following sections describe the activation and operation of the Watchdog Timer.

Watchdog Timer Functional Block Diagram

The functional block diagram for the programmable Watchdog Timer is given in the figure below.

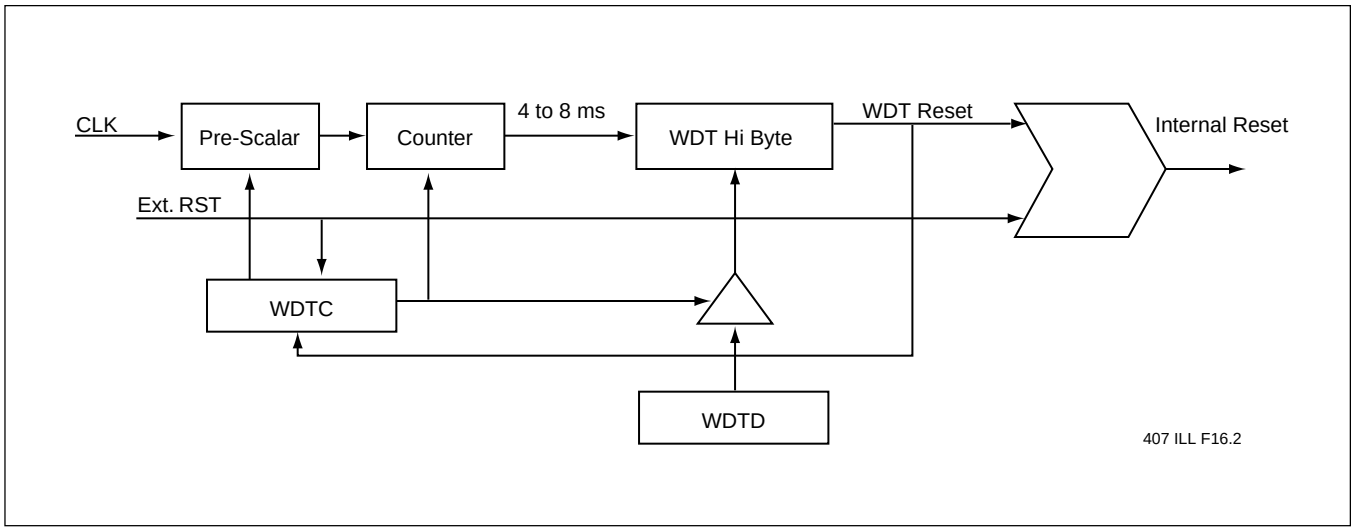


FIGURE 31: BLOCK DIAGRAM OF PROGRAMMABLE WATCHDOG TIMER

The Watchdog Timer is essentially an 8-bit timer. It uses the same time base as the flash operation controller. When the flash operation controller is operating, the time base is re-started by the hardware periodically; therefore, the flash operation controller time base provides an elongated time-out period for the watchdog timer. The time base is then divided by a pre-scalar, which is enabled by the Watchdog Timer Control (WDTC) special function operation controller. The 8-bit Watchdog Timer count value can be programmed with the value stored in the Watchdog Timer Data/Reload (WDTD) special function register. When a software hang or a hardware based software error occurs, the Watchdog Timer unit initiates an automatic recovery of the system by means of an internally generated watchdog reset.

The following figures outline the functions of the WDTD and WDTC special function registers, which are discussed in further detail in the following sections.

Watchdog Timer Control Register (WDTC)

Location	7	6	5	4	3	2	1	0	Reset Value
C0h	–	–	–	–	WDRE	WDTS	WDT	SWDT	x0h
					r/w	r/c	r/s	r/w	

Watchdog Timer Data/Reload Register (WDTD)

Location	7	6	5	4	3	2	1	0	Reset Value
86h	WDRL								00h
	r/w								

Detecting a Software Hang

In order to protect the system against software hang, the user's program must refresh the watchdog timer within a time period previously set by the WDTD register. If the program fails to perform this periodical refresh, the Watchdog Timer will assume that a software hang had occurred and an internally generated watchdog reset will be initiated. The refresh software may also be designed such that the Watchdog Timer times out if the program does not work properly, which includes software errors based on hardware related problems.

Refreshing the Watchdog Timer

The 8-bit Watchdog Timer Data/Reload (WDTD) special function register stores the 8-bit reload value for the Watchdog Timer. When the Watchdog Timer is activated, its high byte is initialized to the reload value stored in the WDTD register. When a refresh is initiated, the reload value is loaded into the high byte of the watchdog timer.

Note that the WDTD register can be written to at any time, therefore an erroneous reload value stored in the WDTD register can be corrected by software.

A refresh is triggered by a consecutive setting of the Watchdog Timer Refresh (WDT) and Start Watchdog Timer (SWDT) bits of the Watchdog Timer Control (WDTC) special function register. Note that the WDT bit must be set directly before the SWDT bit is set in order to prevent an unintentional refresh of the watchdog timer. The WDT bit is cleared by hardware when the refresh is done. The WDT bit may not be cleared by software, however, the WDT bit may be set by software in order to force a Watchdog Timer refresh.

Activating the Watchdog Timer

The Watchdog Timer may be stopped and started by software by setting and clearing the Start Watchdog Timer (SWDT) bit of the Watchdog Timer Control (WDTC) special function register. Program code may be used to set the SWDT bit in order to start the Watchdog Timer and to clear the SWDT bit in order to stop the Watchdog Timer.

Watchdog Timer Reset

When the program fails to refresh the watchdog timer before time-out, an internally generated watchdog reset is initiated, lasting for 20 clocks. The internal watchdog reset differs from an external reset in so far as the Watchdog Timer is not disabled and the Watchdog Timer Reset (WDTS) flag bit of the WDTC register is set. The WDTS flag bit is set by hardware when a watchdog reset occurs and can be cleared by an external hardware reset or by software. The WDTS flag bit keeps track of the source that activates the reset and the program can determine the reset source (external reset or watchdog reset) by examining WDTS flag bit.

The watchdog reset may be enabled or disabled by setting or clearing the Watchdog Timer Reset Enable (WDRE) bit of the WDTC register. Setting the WDRE bit enables the watchdog reset and clearing the WDRE bit disables the watchdog reset. Note that the Watchdog Timer is halted during the Idle and Power Down Modes of operation. Therefore, it is possible to use the Idle Mode in combination with the watchdog timer function, however, the watchdog reset may not be used to exit the Idle or Power Down Modes of operation.

Preliminary

AC ELECTRICAL CHARACTERISTICS

AC Characteristics: (Over Operating Conditions; Load Capacitance for Port 0, ALE, and PSEN# = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

TABLE 7: AC ELECTRICAL CHARACTERISTICS

T_{AMB} = 0°C TO +70°C OR -40°C TO +85°C, V_{DD} = 3V ±10% @ 12 MHZ, 5V ±10% @ 33 MHZ, V_{SS} = 0

Symbol	Parameter	Oscillator						Units
		12 MHz		33 MHz		Variable		
		Min	Max	Min	Max	Min.	Max.	
1/tCLCL	Oscillator Frequency					0	33	MHz
tLHLL	ALE Pulse Width	127		20		2tCLCL - 40		ns
tAVLL	Address Valid to ALE Low	43		5		tCLCL - 40 tCLCL - 25		ns ns
tLLAX	Address Hold After ALE Low	53		5		tCLCL - 30 tCLCL - 25		ns ns
tLLIV	ALE Low to Valid Instr In		234		56		4tCLCL - 100 4tCLCL - 65	ns ns
tLLPL	ALE Low to PSEN# Low	53		5		tCLCL - 30 tCLCL - 25		ns ns
tPLPH	PSEN# Pulse Width	205		46		3tCLCL - 45		ns
tPLIV	PSEN# Low to Valid Instr In		145		35		3tCLCL - 105 3tCLCL - 55	ns ns
tPXIX	Input Instr Hold After PSEN#					0		ns
tPXIZ	Input Instr Float After PSEN#		59		5		tCLCL - 25 tCLCL - 25	ns ns
tAVIV	Address to Valid Instr In		312		71		5tCLCL - 105 5tCLCL - 80	ns ns
tPLAZ	PSEN# Low to Address Float		10		10		10	ns
tRLRH	RD# Pulse Width	400		82		6tCLCL - 100		ns
tWLWH	Write Pulse Width (WE#)	400		82		6tCLCL - 100		ns
tRLDV	RD# Low to Valid Data In		252		61		5tCLCL - 165 5tCLCL - 90	ns ns
tRHDX	Data Hold After RD#	0		0		0		ns
tRHDZ	Data Float After RD#		107		35		2tCLCL - 60 2tCLCL - 25	ns ns
tLLDV	ALE Low to Valid Data In		517		150		8tCLCL - 150 8tCLCL - 90	ns ns
tAVDV	Address to Valid Data In		585		180		9tCLCL - 165 9tCLCL - 90	ns ns
tLLWL	ALE Low to RD# or WR# Low	200	300	40	140	3tCLCL - 50	3tCLCL + 50	ns
tAVWL	Address to RD# or WR# Low	203		46		4tCLCL – 130 4tCLCL – 75		ns ns
tQVWX	Data Valid to WR# Transition	33		0		tCLCL - 50 tCLCL - 30		ns ns
tWHQX	Data Hold After WR#	33		3		tCLCL - 50 tCLCL - 27		ns ns
tQVWH	Data Valid to WR# High	433		140		7tCLCL - 150 7tCLCL - 70		ns ns
tRLAZ	RD# Low to Address Float		0		0		0	ns
tWHLH	RD# or WR# High to ALE High	43	123	5	55	tCLCL - 40 tCLCL - 25	tCLCL + 40 tCLCL + 25	ns ns
tVR[1]	V_{DD} to Reset		10		10		10	ns

Note: 1. Refer to Figure 3 for minimum timing requirements for Power-On Reset.

407 PGM T7.3

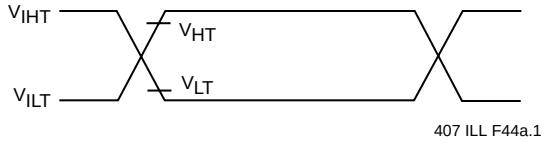
Explanation of Symbols

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A: Address
C: Clock
D: Input data
H: Logic level HIGH
I: Instruction (program memory contents).
L: Logic level LOW or ALE
P: PSEN#
Q: Output data
R: RD# signal
T: Time
V: Valid
W: WR# signal
X: No longer a valid logic level
Z: High Impedance (Float)

For example:

tAVLL=Time from Address Valid to ALE Low
tLLPL=Time from ALE Low to PSEN# Low

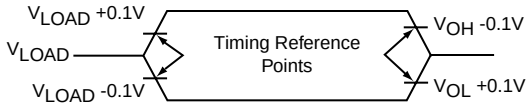


407 ILL F44a.1

AC Inputs during testing are driven at V_{IHT} (V_{DD} -0.5V) for Logic "1" and V_{ILT} (0.45V) for a Logic "0". Measurement reference points for inputs and outputs are at V_{HT} (0.2 V_{DD} + 0.9) and V_{LT} (0.2V_{DD} - 0.1)

Note: V_{HT}- V_{HIGH} Test
V_{LT}- V_{LOW} Test
V_{IHT}-V_{INPUT} HIGH Test
V_{ILT}- V_{INPUT} LOW Test

AC Testing Input/Output

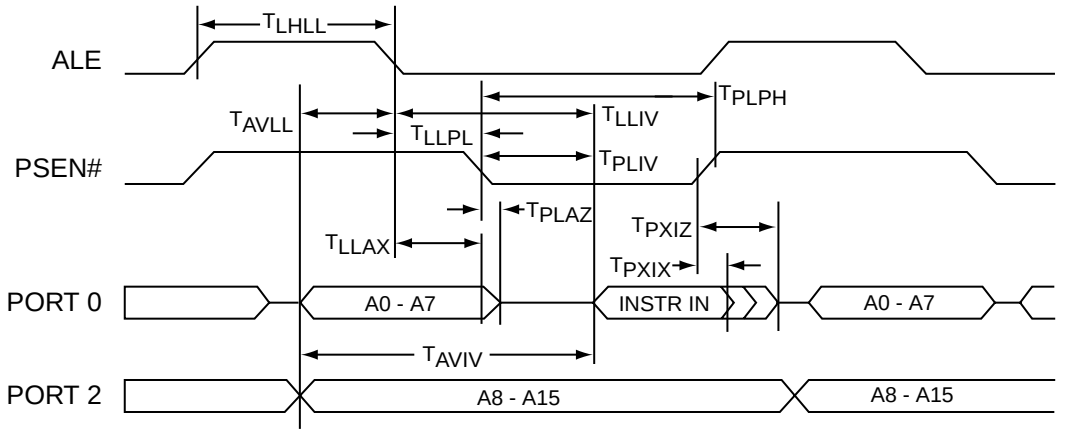


407 ILL F44b.0

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. I_{OL}/I_{OH} = ± 20mA.

Float Waveform

AC Testing Input/Output, Float Waveform



407 ILL F17.2

Figure 32: External Program Memory Read Cycle

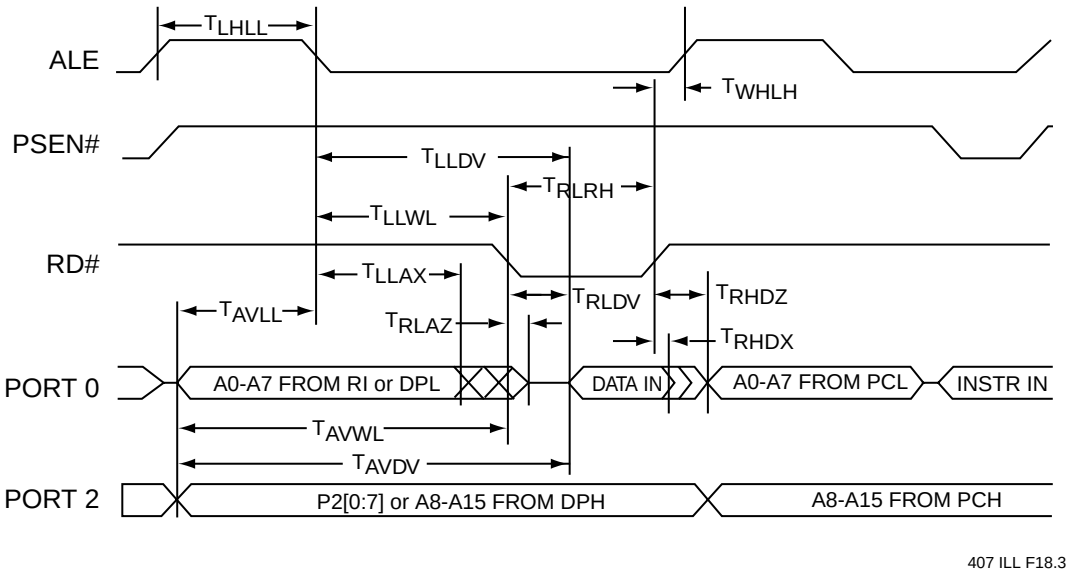


FIGURE 33: EXTERNAL DATA MEMORY READ CYCLE

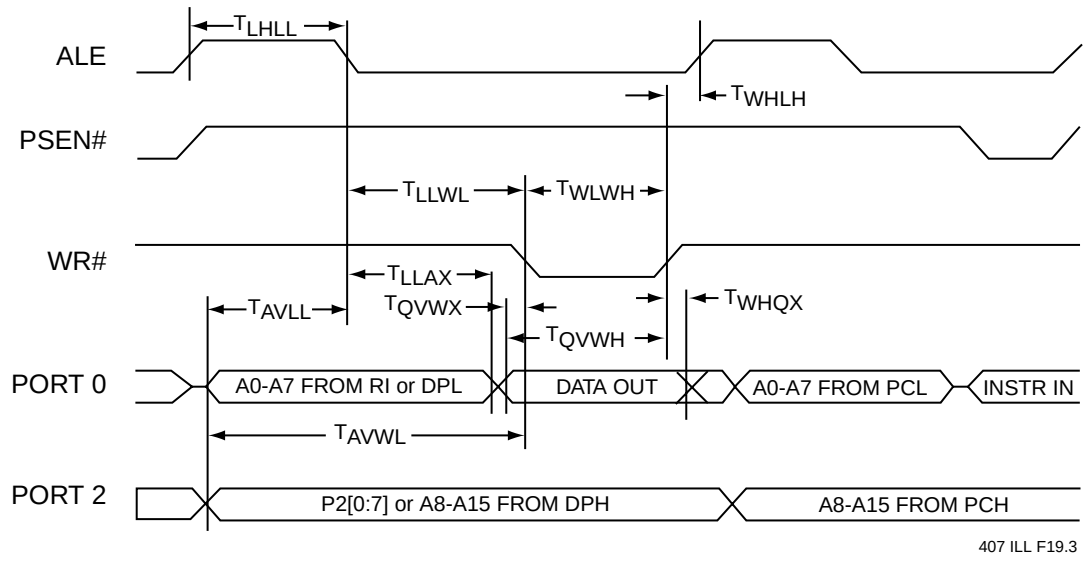


FIGURE 34: EXTERNAL DATA MEMORY WRITE CYCLE

SST89F54/58 Pin Descriptions

Below is a summary of each of the pins of the SST89F54/58. Note that the SST89F54/58 are pin-for-pin compatible with the industry standard 8xC5x microcontroller.

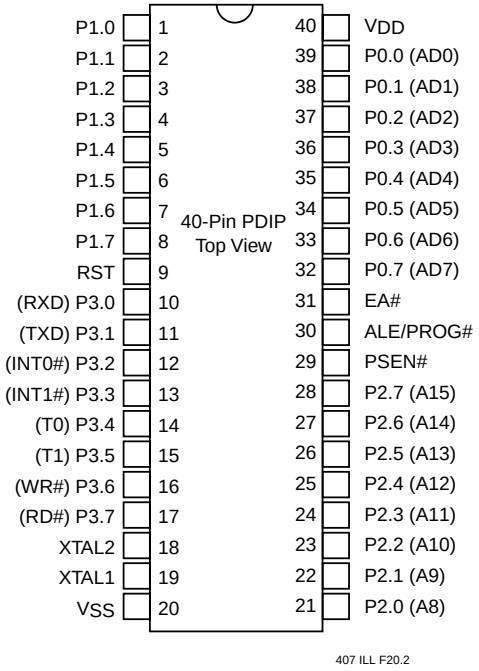


FIGURE 35: PIN ASSIGNMENTS FOR 40-PIN PLASTIC DIP

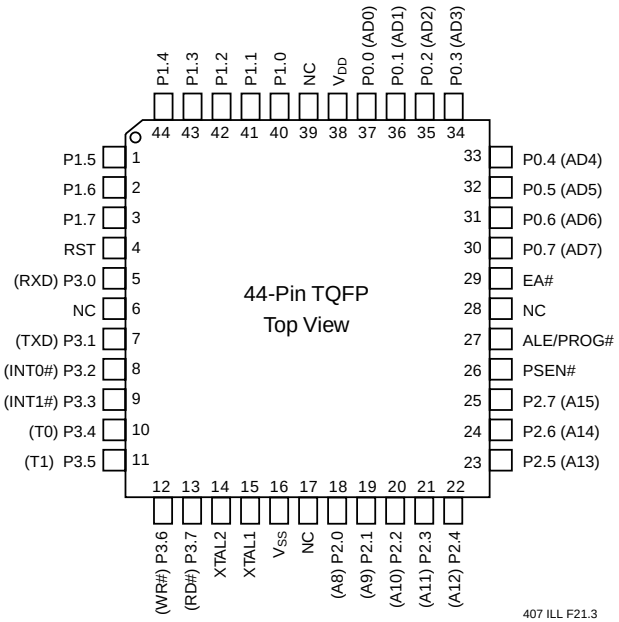


FIGURE 36: PIN ASSIGNMENTS FOR 44-PIN TQFP

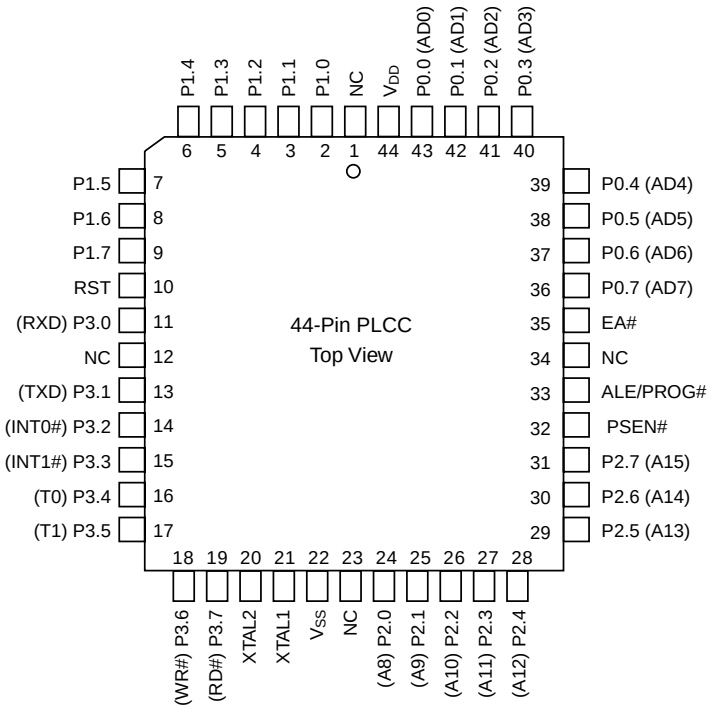


FIGURE 37: PIN ASSIGNMENTS FOR 44-PIN PLCC

Preliminary

TABLE 8: PIN DESCRIPTIONS

Symbol	Type[1]	Name and Functions
P0[7:0]	I/O[1]	***Port 0:*** Port 0 is an 8-bit open drain bi-directional I/O port. As an output port each pin can sink several LS TTL inputs. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external memory. In this application it uses strong internal pull-ups when transitioning to 1's. Port 0 also receives the code bytes during FLASH MEMORY programming, and outputs the code bytes during program verification. External pull-ups are required during program verification.
P1[7:0]	I/O with internal pull-ups	***Port 1:*** Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can drive LS TTL inputs. Port 1 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current (I_{IL}, on the data sheet) because of the internal pull-ups. P1(5,6,7) pins have high current drive of 16 mA. Port 1 also receives the low-order address bytes during FLASH MEMORY programming and program verification.
P2[7:0]	I/O with internal pull-ups	***Port 2:*** Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current (I_{IL}, on the data sheet) because of the internal pull-ups. Port 2 sends the high-order address byte during fetches from external Program memory and during accesses to external Data Memory that use 16-bit address (MOVX@DPTR). In this application it uses strong internal pull-ups when outputting 1's. During accesses to external Data Memory that use 8-bit addresses (MOVX@Ri), Port 2 sends the contents of the P2 Special Function Register. Port 2 also receives some control signals and a partial of high-order address bits during FLASH MEMORY programming and program verification.
P3[7:0]	I/O with internal pull-ups	***Port 3:*** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers could drive LS TTL inputs. Port 3 pins that have 1's written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally pulled low will source current (I_{IL}, on the data sheet) because of the pull-ups. Port 3 also serves the functions of various special features of the FlashFlex51 Family. Port 3 also receives some control signals and a partial of high-order address bits during FLASH MEMORY programming and program verification.
P3.0	I	***RXD:*** Serial input line
P3.1	O	***TXD:*** Serial output line
P3.2	I	***INT0#:*** External Interrupt 0
P3.3	I	***INT1#:*** External Interrupt 1
P3.4	I	***T0:*** Timer 0 external input
P3.5	I	***T1:*** Timer 1 external input
P3.6	O	***WR#:*** External Data Memory Write strobe
P3.7	O	***RD#:*** External Data Memory Read strobe
PSEN#	O/I	***Program Store Enable:*** PSEN# is the Read strobe to External Program Memory. When the SST89F54/58 are executing from Internal Program Memory, PSEN# is inactive (high). When the device is executing code from External Program Memory, PSEN# is activated twice each machine cycle, except that two PSEN# activation are skipped during each access to External Data Memory. While the RST input is continually held high (for more than ten machine cycles), a forced high-to-low input transition on the PSEN# pin will bring the device into the "External Host" mode for the internal flash memory programming operation.

PIN DESCRIPTIONS (CONTINUED)

Symbol	Type[1]	Name and Functions
RST	I	***Reset:*** A high on this pin for two machine cycles while the oscillator is running resets the device. The port pins will be driven to their reset condition when a minimum V_{IH1} voltage is applied whether the oscillator is running or not. After a successful reset is completed, if the PSEN# pin is driven by an input force with a high-to-low transition while the RST input pin is continually held high, the device will enter the "External Host" mode for the internal flash memory programming operation, otherwise the device will enter the "Normal" operation mode.
EA#	I	***External Access Enable:*** EA# must be strapped to V_{SS} in order to enable the SST89F54/58 to fetch code from External Program Memory locations starting at 0000h up to FFFFh. Note, however, that if the Security Lock is activated on either block, the logic level at EA# is internally latched during reset. EA# must be strapped to V_{DD} for internal program execution.The EA# pin can tolerate a high voltage[2] of 12V (see Electrical Specifications)
ALE/PROG#	I/O	***Address Latch Enable:*** ALE is the output signal for latching the low byte of the address during accesses to external memory. This pin is also the programming pulse input (PROG#).
$XTAL_1$ $XTAL_2$	I O	***Oscillator:*** Input and output to the inverting oscillator amplifier. XTAL1 is input to internal clock generation circuits from an external clock source.
V_{DD}	I	***Power Supply:*** Supply voltage
Vss	I	***Ground:*** Circuit ground. (0V reference)

Note: 1) I = Input
O = Output
2) It is not necessary to receive a 12V programming supply voltage during flash programming.

407 PGM T8.6

Appendix A- Programming Examples

Example 1: Using the 4K Flash Block

In this example, the assembly language program exercises the 8-LEDs, located at Port 1 on the Baseboard from the SST89C5x Starter Kit. Visually, the software routine causes two adjacent LEDs to light up, and proceed to shift right one LED position at a specific time interval until they reach the right most LED position, then they shift left in the same manner. This right-left sequence is continuous. The program executes from the 4K Flash block in internal memory. The program code is divided into the following sections:

(1) Equates to assign specific values in the program

(2) Starting code at hardware or power-on RESET (memory address 0000h) modifies bit 7 in the SFCF register to make the secondary flash block (4K block) visible in the 64K memory space

(3) Main program (in 4K block)

(4) Software time delay subroutine (in 4K block).

```
*****************************************************************************
Program begins here...
*****************************************************************************
;.............................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;       EQUATES             ;
;.............................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

init_val        equ     3Fh         ; initial display value
dsply_port      equ     90h         ; port P1
sfcf_port       equ     0F7h        ; port SFCF


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;   RESET (Start of Code)  ;
;.............................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                org     0000h
init:
                mov     a,sfcf_port     ; get SFCF register contents
                orl     a, #80h         ; set VIS bit=1 in SFCF reg.
                mov     sfcf_port, a    ; to enable upper 4K block
                ljmp    main            ; jump to 4K block
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;    MAIN Program Loop     ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                  org         0F000h
main:
                  mov         a, #init_val       ; load initial value to display
                  mov         r3, #06h           ; load # of right shifts

lup1:
                  mov         dsply_port, a      ; display value
                  mov         r0, #00h           ; load time delay regs
                  mov         r1, #00h
                  mov         r2, #02h
                  acall       dly                ; insert a time delay

                  rr          a                  ; shift display value 1 bit right
                  dec         r3                 ; decrement shift count
                  cjne        r3, #00h, lup1     ; repeat right shifts until count = 0

                  mov         r3, #06h           ; load # of left shifts
lup2:
        mov     dsply_port, a   ; display value
                  mov         r0, #00h           ; load time delay regs
                  mov         r1, #00h
                  mov         r2, #02h
                  acall       dly                ; insert a time delay

                  rl          a                  ; shift display value 1 bit left
                  dec         r3                 ; decrement shift count
                  cjne        r3, #00h, lup2     ; repeat left shifts until count = 0

; Go to top of the code and start the entire code sequence over again.
; This will be a Do Forever loop.
;
        ajmp    main

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;   Software Time Delay    ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; Note that the time delay in the following delay
; subroutine will be dependent upon the data values
; inserted in registers r0, r1 and r2 prior to calling
; this subroutine. Each instruction has its own time
; of execution, and therefore a calculation of the time
; related to number of machine cycles must be made.
; The number of machine cycles multiplied by 12 clocks/
; machine cycle, then multiplied by the clock period
; will provide the expected time delay.
```

Preliminary

```
;
; In the Delay Subroutine, let's call the first 5
; instructions Loop1 (based on register R0), the next
; 5 Loop2 (based on R1), the next 5 Loop3 (based on R2),
; and last instruction RET.  DEC and NOP instructions
; each take 1 machine cycle to execute, while the CJNE
; and RET take 2 cycles.  Thus, the calculation for the
; number of machine cycles (#MC) is as follows:
; #MC  = 5*R0                              (1st time thru Loop1)
;              + [(256*5) + 5]R1 (combines Loops 1 & 2)
;              + [((256*5) + 5)256 + 5]R2    (combines Loops 1-3)
;              + 2                          (for RET instr.)
;         = 5*R0 + 1285*R1 + 328,965*R2 + 2
; (Note that the R2 component provides the largest part
;  of the delay.)
;
; Delay time = #MC * 12 Clks/MC * 1/Clk Frequency
;
; In an example r0=r1=r2=00h (note: 00h = 256),
; therefore,
; #MC  = 5*256 + [(256*5) + 5]256 + [((256*5) + 5)256
;              + 5]256 +2
;         = 1280 + 328960 + 84215040 + 2
;         = 84545282
;
; Assuming a 12 MHz system clock,
;    Delay time = 84545282 * 12 * 1/12 MHz
;                  = 84.55 seconds, which corresponds
;                 to the max. possible time delay.

; If r2 was changed to the value 2, then the delay
; would be equal to 0.99 seconds. If r2 = 6, then
; the delay = 2.94 seconds.

dly:                                    ; delay subroutine
          dec     r0
          nop
          nop
          cjne    r0, #00h, dly
          dec     r1
          nop
          nop
          cjne    r1, #00h, dly
          dec     r2
          nop
          nop
          cjne    r2, #00h, dly
          ret

          end
```

Example 2: Watchdog Timer

The Watchdog Timer (WDT) is programmed in this example to show how this specific component can be used in a design.

```
********************************************************************************
Program begins here...
********************************************************************************
;.............................................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;       Watchdog Timer EQUATES        ;
;.............................................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

wdtc            equ     0C0h    ; watchdog timer control register
wdtd            equ     86h     ; watchdog timer data/reload register
wdre_H          equ     08h     ; watchdog timer reset enable
wdre_L          equ     0F7h    ; watchdog timer reset disable
wdts            equ     04h     ; watchdog timer reset flag bit
wdt             equ     02h     ; watchdog timer timer refresh bit
swdt_H          equ     01h     ; start watchdog timer data
swdt_L          equ     0FEh    ; stop watchdog timer data
wdt_hbyt        equ     0F0h    ; programmed wdt hi-byte


;.............................................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;   Watchdog Timer Initialization     ;
;.............................................................
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; Set up the Watchdog Timer for normal operation.  During
; initialization for the Watchdog Timer, disable the Watchdog Timer
; function by writing a "0" to the Watchdog Timer Reset Enable bit
; (bit 3) in the Watchdog Timer Control Register (WDTC).

        mov     a, wdtc         ; disable watchdog timer
        anl     a, #wdre_L
        mov     wdtc, a
```

```
; Now load the desired reset timer value in the Watchdog Timer Data/Reload
; Register. Note that loading F0h would correspond to an overflow in 10h counts,
; loading E0h overflows in 20h counts, etc. Note that loading 00h would provide
; 256 counts, which in turn, issues a WDT reset in approximatly 1.6 second
; for the 6.4 millisecond clock. After
; initializing the timer, enable the watchdog timer.

        mov     wdtd, #wdt_hbyt         ; load timer value
        mov     a, wdtc                 ; enable watchdog timer
        orl     a, wdre_H
        mov     wdtc, a



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;       To Start WDT Operation      ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; The Watchdog Timer is started by writing a "1" to the Start
; Watchdog Timer bit, SWDT, which is bit 0 in the Watchdog Timer
; Control Register (WDTC). This code should be placed
; at the end of startup initialization.

        mov     a, wdtc
        orl     a, #swdt_H
        mov     wdtc, a
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;           To Stop WDT Operation       ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; The Watchdog Timer is stopped by writing a "0" to the Start
; Watchdog Timer bit, SWDT, which is bit 0 in the Watchdog Timer
; Control Register (WDTC).

            mov     a, wdtc
            anl     a, #swdt_L
            mov     wdtc, a


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;           To Software Refresh the WDT   ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; The Watchdog Timer must be refresh at regular timing intervals
; within the software prior to the WDT timing out.  The following
; instructions must be inserted into the main body of the code:

            mov     a, wdtc
            orl     a, #wdt
            mov     wdtc, a


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;           To Clear WDT Reset Flag                    ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; If the Watchdog Timer Reset is enabled (WDRE = WDTC.3) and a WDT
; reset (or overflow) occurs, then the WDT Reset Flag (WDTS = WDTC.2)
; is set when the WDT reset generates an internal reset within the SST89C5x chip.
; The WDT Reset Flag  can be used by the software to determine the source of the
; reset. Since this flag bit is not cleared by a WDT reset signal, it is cleared by
; the software writing a "1" to the WDT Reset Flag bit.

            mov     a, wdtc             ; get contents of WDTC register
            orl     a, #wdts            ; clear the WDT reset flag bit
            mov     wdtc, a             ; update WDTC register

            end
```

Appendix B- Instruction Set

Arithmetic Operations

Instruction		Description	Byte	Cycle	C	AC	OV
ADD	A,Rn	Add Rn to ACC	1	1	x	x	x
ADD	A,direct	Add direct to ACC	2	1	x	x	x
ADD	A,@Ri	Add indirect to ACC	1	1	x	x	x
ADD	A,#data	Add immediate to ACC	2	1	x	x	x
ADDC	A,Rn	Add Rn with Carry	1	1	x	x	x
ADDC	A,direct	Add direct with Carry	2	1	x	x	x
ADDC	A,@Ri	Add indirect with Carry	1	1	x	x	x
ADDC	A,#data	Add immediate with Carry	2	1	x	x	x
SUBB	A,Rn	Subtract Rn with Borrow	1	1	x	x	x
SUBB	A,direct	Subtract dir. with Borrow	2	1	x	x	x
SUBB	A,@Ri	Subtract ind. with Borrow	1	1	x	x	x
SUBB	A,#data	Subtract imm. with Borrow	2	1	x	x	x
INC	A	Increment ACC	1	1	-	-	-
INC	Rn	Increment Rn	1	1	-	-	-
INC	direct	Increment direct	2	1	-	-	-
INC	@Ri	Increment indirect	1	1	-	-	-
DEC	A	Decrement ACC	1	1	-	-	-
DEC	Rn	Decrement Rn	1	1	-	-	-
DEC	direct	Decrement direct	2	1	-	-	-
DEC	@Ri	Decrement indirect	1	1	-	-	-
INC	DPTR	Increment DPTR	1	2	-	-	-
MUL	AB	Multiply A and B	1	4	0	x	-
DIV	AB	Divide A by B	1	4	0	x	-
DA	A	Decimal-adjust ACC	1	1	x	-	-

Logical Operations

Instruction		Description	Byte	Cycle	C	AC	OV
ANL	A,Rn	AND Rn to ACC	1	1	-	-	-
ANL	A,direct	AND direct to ACC	2	1	-	-	-
ANL	A,@Ri	AND indirect to ACC	1	1	-	-	-
ANL	A,#data	AND immediate to ACC	2	1	-	-	-
ANL	direct,A	AND ACC to direct	2	1	-	-	-
ANL	direct,#data	AND immediate to direct	3	2	-	-	-
ORL	A,Rn	OR Rn to ACC	1	1	-	-	-
ORL	A,direct	OR direct to ACC	2	1	-	-	-
ORL	A,@Ri	OR indirect to ACC	1	1	-	-	-
ORL	A,#data	OR immediate to ACC	2	1	-	-	-
ORL	direct,A	OR ACC to direct	2	1	-	-	-
ORL	direct,#data	OR immediate to direct	3	2	-	-	-
XRL	A,Rn	XOR Rn to ACC	1	1	-	-	-
XRL	A,direct	XOR direct to ACC	2	1	-	-	-
XRL	A,@Ri	XOR indirect to ACC	1	1	-	-	-
XRL	A,#data	XOR immediate to ACC	2	1	-	-	-
XRL	direct,A	XOR ACC to direct	2	1	-	-	-
XRL	direct,#data	XOR immediate to direct	3		-	-	-
CLR	A	Clear ACC	1	1	-	-	-
CPL	A	Complement ACC	1	1	-	-	-
RL	A	Rotate ACC left	1	1	-	-	-
RLC	A	(through C)	1	1	x	-	-
RR	A	Rotate ACC right	1	1	-	-	-
RRC	A	(through C)	1	1	x	-	-
SWAP	A	Swap nibbles within ACC	1	1	-	-	-

Data Transfer Operations

Instruction		Description	Byte	Cycle	C	AC	OV
MOV	A,Rn	Move Rn to ACC	1	1	-	-	-
MOV	A,direct	Move direct to ACC	2	1	-	-	-
MOV	A,@Ri	Move indirect to ACC	1	1	-	-	-
MOV	A,#data	Move immediate to ACC	2	1	-	-	-
MOV	Rn,A	Move ACC to Rn	1	1	-	-	-
MOV	Rn,direct	Move direct to Rn	2	2	-	-	-
MOV	Rn,#data	Move immediate to Rn	2	1	-	-	-

MOV	direct,A	Move ACC to direct	2	1	-	-	-
MOV	direct,Rn	Move Rn to direct	2	2	-	-	-
MOV	direct,direct	Move direct to direct	3	2	-	-	-
MOV	direct,@Ri	Move indirect to direct	2	2	-	-	-
MOV	direct,#data	Move immediate to direct	3	2	-	-	-
MOV	@Ri,A	Move ACC to indirect	1	1	-	-	-
MOV	@Ri,direct	Move direct to indirect	2	2	-	-	-
MOV	@Ri,#data	Move immediate to indir.	2	1	-	-	-
MOV	DPTR,#data16	Move 16-bit imm. to DPTR	3	2	-	-	-
MOVC	A,@A+DPTR	Move code or constant pointed by A+DPTR to ACC	1	2	-	-	-
MOVC	A,@A+PC	Move code or constant pointed by A+PC to ACC	1	2	-	-	-
MOVX	A,@Ri	Move external indirect pointed by Rn to ACC	1	2	-	-	-
MOVX	A,@DPTR	Move external indirect pointed by DPTR to ACC	1	2	-	-	-
MOVX	@Ri,A	Move ACC to external indirect pointed by Rn	1	2	-	-	-
MOVX	@DPTR,A	Move ACC to external indirect pointed by DPTR	1	2	-	-	-
PUSH	direct	Push direct onto stack	2	2	-	-	-
POP	direct	Pop from stack to direct	2	2	-	-	-
XCH	A,direct	Exchange direct with ACC	2	1	-	-	-
XCH	A,@Ri	Exchange indir. with ACC	1	1	-	-	-
CHD	A,@Ri	Exchange ACC lower-nibble with indirect memory	1	1	-	-	-

Boolean Variable Manipulation

Instruction		Description	Byte	Cycle	C	AC	OV
CLR	C	Clear Carry	1	1	0	-	-
CLR	bit	Clear direct bit	2	1	-	-	-
SETB	C	Set Carry	1	1	1	-	-
SETB	bit	Set direct bit	2	1	-	-	-
CPL	C	Complement Carry	1	1	x	-	-
CPL	bit	Complement direct bit	2	1	-	-	-

ANL	C,bit	AND direct bit to Carry	2	2	x	-	-
ANL	C,/bit	AND complememt bit to C	2	2	x	-	-
ORL	C,bit	OR direct bit to Carry	2	2	x	-	-
ORL	C,/bit	OR complememt bit to C	2	2	x	-	-
MOV	C,bit	Move direct bit to Carry	2	1	x	-	-
MOV	bit,C	Move Carry to direct bit	2	2	-	-	-
JC	rel	Jump if Carry is set	2	2	-	-	-
JNC	rel	Jump if Carry is not set	2	2	-	-	-
JB	bit,rel	Jump if direct bit is set	3	2	-	-	-
JNB	bit,rel	Jump if dir bit is not set	3	2	-	-	-
JBC	bit,rel	Jump if direct bit is set and clear it	3	2	-	-	-

Program Branching

Instruction		Description	Byte	Cycle	C	AC	OV
ACALL	addr11	Absolute subroutine call	2	2	-	-	-
LCALL	addr16	Long subroutine call	3	2	-	-	-
RET		Return from subroutine	1	2	-	-	-
RETI		Return from interrupt	1	2	-	-	-
AJMP	addr11	Absolute jump	2	2	-	-	-
LJMP	addr16	Long jump	3	2	-	-	-
SJMP	rel	Short jump	2	2	-	-	-
JMP	@A+DPTR	Jump to indirect pointed by A+DTPR	1	2	-	-	-
JZ	rel	Jump if ACC is zero	2	2	-	-	-
JNZ	rel	Jump if ACC is not zero	2	2	-	-	-
CJNE	A,direct,rel	Compare ACC with direct and jump if not equal	3	2	x	-	-
CJNE	A,#data,rel	Compare ACC with immediate and jump if not equal	3	2	x	-	-
CJNE	Rn,#data,rel	Compare Rn with immediate and jump if not equal	3	2	x	-	-
CJNE	@Ri,#data,rel	Compare indirect with imm. and jump if not equal	3	2	x	-	-
DJNZ	Rn,rel	Decrement Rn and jump if not zero	2	2	-	-	-
DJNZ	direct,rel	Decrement direct and jump if not zero	3	2	-	-	-
NOP	No Operation		1	1	-	-	-

Appendix C- Ordering Information

Product Ordering Information

Product Identification Descriptor

Device **Speed** **Suffix 1** **Suffix 2**

SST89F5 X - 33 - X X - X X - XXXXXA

C-Spec Identification Code
C-Spec Revision
C-Spec Tracking No.

Package Modifier
I = 40 pins
J = 44 pins

Package Type
P = PDIP
N = PLCC
TQ = TQFP

Operation Temperature
C = Commercial = 0° to 70°C
I = Industrial = -40° to 85°C

Release ID
Blank = Initial release
A = First enhancement

Operating Frequency
33 = 0-33 MHz

Memory Size
4***** = 16(20) KByte
8***** = 32(36) KByte
* = 4K additional flash can be enabled via VISIBLE bit in SFCF

Core Type
51 Core

Voltage Range
2.7-5.5V

Device Family

Part Number Valid Combinations

SST89F54 Valid combinations

Part Number	*Package*	*Pins*	*V_{DD}*	*Speed*	*Temperature*
SST89F54-33-C-PI	PDIP	40	2.7-5.5	0-33MHz	Commercial
SST89F54-33-C-NJ	PLCC	44	2.7-5.5	0-33MHz	Commercial
SST89F54-33-C-TQJ	TQFP	44	2.7-5.5	0-33MHz	Commercial
SST89F54-33-I-PI	PDIP	40	2.7-5.5	0-33MHz	Industrial
SST89F54-33-I-NJ	PLCC	44	2.7-5.5	0-33MHz	Industrial
SST89F54-33-I-TQJ	TQFP	44	2.7-5.5	0-33MHz	Industrial

SST89F58 Valid combinations

Part Number	*Package*	*Pins*	*V_{DD}*	*Speed*	*Temperature*
SST89F58-33-C-PI	PDIP	40	2.7-5.5	0-33MHz	Commercial
SST89F58-33-C-NJ	PLCC	44	2.7-5.5	0-33MHz	Commercial
SST89F58-33-C-TQJ	TQFP	44	2.7-5.5	0-33MHz	Commercial
SST89F58-33-I-PI	PDIP	40	2.7-5.5	0-33MHz	Industrial
SST89F58-33-I-NJ	PLCC	44	2.7-5.5	0-33MHz	Industrial
SST89F58-33-I-TQJ	TQFP	44	2.7-5.5	0-33MHz	Industrial

Example: Valid combinations are those products in mass production or will be in mass production. Consult your SST sales representative to confirm availability and to determine availability of new combinations.

Part Number Cross-Reference Guide

Intel		SST		package
i87C54	16 KB EPROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
i87C58	32 KB EPROM & 256B RAM	SST89F58	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
i87L54	16 KB ROM (OTP) & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	L Q
i87L58	32 KB ROM (OTP) & 256B RAM	SST89F58	4 KB Flash, 32 KB Flash & 256B RAM	L Q
i87C51FB	16 KB EPROM & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
i87C51FC	32 KB EPROM & 256B RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	D L Q

Atmel		SST		package
AT89C52	8 KB Flash & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
AT89LV52	8 KB Flash & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
AT89S53	12 KB Flash & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
AT89LS53	12 KB Flash & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
AT89C55	20 KB Flash & 256B RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
AT89LV55	20 KB Flash & 256B RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	D L Q

Temic		SST		package
80C51	4 KB ROM & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
80C52	8 KB ROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
83C154	16 KB ROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
83C154D	32 KB ROM & 256B RAM	SST89F58	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
87C51	4 KB EPROM & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
87C52	8 KB EPROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q

Philips		SST		package
P80C54	16 KB ROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
P80C58	32 KB ROM & 256B RAM	SST89F58	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
P87C54	16 KB EPROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
P87C58	32 KB EPROM & 256B RAM	SST89F58	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
P87C524	16 KB EPROM & 512B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L Q
P87C528	32 KB EPROM & 512B RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
P83C524	16 KB ROM & 512B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	D L
P83C528	32 KB MROM & 512B RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	D L Q
P89CE558	32 KB Flash & 1K RAM	SST89F58*	4 KB Flash, 32 KB Flash & 256B RAM	

Siemens		SST		package
C501-1R	8 KB ROM & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L
C501-1E	8 KB ROM (OTP) & 256B RAM	SST89F54	4 KB Flash, 16 KB Flash & 256B RAM	D L
C513A-H	12 KB EPROM & 512B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	L
C503-1R	8 KB ROM & 256B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	L
C504-2R	16 KB ROM & 512B RAM	SST89F54*	4 KB Flash, 16 KB Flash & 256B RAM	Q

D: PDIP
L: PLCC
Q: TQFP

NOTE: The SST89F58 can be substituted for any SST89F54 listing above.
* Indicates SST similar function and not direct replacement/socket compatible.

Appendix D- Third Party Development Tools

For the most up-to-date development tools information, visit the SST web site (www.ssti.com) or manufacturer web sites listed.

Programmers

Company	Tools	Availability
Advantech Equipment Corporation Fl.5 108-3, Ming-Chuan Road Shing-Tien City Taipei, Taiwan, R.O.C. Phone: 886.2.218.4567 Fax: 886.2.218.2435 BBS: 886.2.218.5434 Web: www.advantech.com.tw	Labtool-48	Now
Advin Systems 1050-L East Duane Avenue Sunnyvale, CA 94086 Phone: 408.243.7000 Fax: 408.736.2503 BBS: 408.737.9200 Email: sales@advin.com Web: www.advin.com	Pilot-146, MVP, U128+, U44+, U84+	Now
Electronic Engineering Tools 544 Weddell Drive, Suite 6 Sunnyvale, CA 94089 Phone: 408.734.8184 Fax: 408.734.8185 Email: support@eetools.com Web: www.eetools.com	TopMax	Now
Hi-Lo Systems Research Co. Ltd. 4F, No. 2, Sec. 5, Ming Shen E. Road Taipei, Taiwan R.O.C. Phone: 886.2.7640215 Fax: 886.2.7566403 TW BBS: 886.2.7690881 US BBS: 510.623.0430 Email: hilosale@hilosystems.com.tw Web: www.hilosystems.com.tw	ALL-07	Now
Leap Electronics Co. Ltd. 6th Fl-4 No. 4, Lane 609, Sec 5 Chunghsin Road, Sanchung City, Taipei Hsien, Taiwan, R.O.C. Phone: 886.2.9991860 Fax: 886.2.9990015 BBS: N/A Email: service@leap.com.tw Web: www.leap.com.tw	Leaper-10, LP-U4	Now

Company	Tools	Availability
Needham's Electronics, Inc. 4630 Beloit Drive, Suite 20 Sacramento, CA 95838 Phone: 916.924.8037 Fax: 916.924.8065 BBS: 916.924.8094 Web: www.needhams.com FTP: ftp.needhams.com	EMP 30	Late 1Q99
Phyton, Inc. 7206 Bay Parkway, 2nd Floor Brooklyn, NY 11204 Phone: 718.259.3191 Fax: 718.259.3191 Email: igs@phyton.com Web: www.phyton.com	ChipProg	Now
Stag Programmers Ltd. Silver Court, Watchmead Welwyn Garden City, Herts AL7 1LT, UK Phone: 44.1707.332148 Fax: 44.1707.371503 BBS: 408.988.1768	Eclipse	Late 1Q99
System General Corporation 3F, No. 1, Alley 8 Lane 45, Bao Shing Road, Shin Dian, Taipei, Taiwan, R.O.C. Phone: 886.2.917.3005 Fax: 886.2.911.1283 Web: www.sg.com.tw **System General Corporation** 1623 South Main Street Milpitas, CA 95035 Phone: 408.263.6667 Fax: 408.263.9220 BBS: 408.262.6438	Turpro 848	Now

Programmer Adapters

Company	Tools	Availability
EDI Corporation 2611 South Highland Drive Las Vegas, NV 89109 Phone: 702.735.4997 Fax: 702.735.8339 Web: www.edi-adapters.com	Programmer, emulator and logic analyzer adapters	Now
Emulation Technology, Inc. 2344 Walsh Avenue, Bldg. F Santa Clara, CA 95051 Phone: 408.982.0660 Toll Free: 800.ADAPTER Fax: 408.982.0664	Programmer, emulator and logic analyzer adapters	Now

Software: Compilers, Assemblers, Simulators

Company	Tools	Availability
Avocet Systems, Inc. 120 Union Street Rockport, MA 04856 Phone: 207.236.9055 Toll Free: 800.448.8500 Fax: 207.236.6713 Web: www.midcoast.com/~avocet	C compiler/assembler, C source-level simulator, Assembly-level simulator, Integrated Development Environment	Now
BSO/Tasking, Inc. 333 Elm Street Dedham, MA 02026 Phone: 617.320.9400 Fax: 617.320.9212 Web: www.tasking.ml	C compiler/assembler, C source-level simulator, Assembly-level simulator, Integrated Development Environment	Now
Keil Software 16990 Dallas Pkwy, Suite 120 Dallas, TX 75248 Phone: 972.735.8052 Toll Free: 800.348.8051 Fax: 972.735.8055 Web: www.keil.com **Keil Elektronik GmbH** Bretonischer Ring 15 D-85630 Grasbrun Germany Phone: ++49.89.456040.0 Fax: ++49.89.468162 Web: www.keil.com/~market	C compiler/assembler, Hi-level simulator, Integrated Development Environment	Now

Preliminary

In-Circuit Emulators (ICE)

Company	Tools	Availability
Hitex Development Tools 710 Lakeway Drive, Suite 280 Sunnyvale, CA 94086 Phone: 408.733.7080 Toll Free: 800.45.HITEX Fax: 408.733.6320 Web: www.hitex.com **Hitex-Systementwicklung** Greschbachstrasse 12 D-76229 Karlsruhe, Germany Phone: +49.721.9628.133 Fax: +49.721.9628.189 Web: www.hitex.de	MX51A (to 40 MHz)	Now
Metalink Corporation 325 E. Elliot Road, Suite 23 Chandler, AZ 85225 Phone: 602.926.0797 Fax: 602.926.1198 Web: http://metaice.com	iceMaster-PE (to 16 MHz) iceMaster-AA (to 24 MHz) iceMaster-SF (to 33 MHz)	Now Now 2Q99
Nohau Corporation 51 E. Campbell Avenue Campbell, CA 95008 Phone: 408.866.1820 Fax: 408.378.7869 Web: www.nohau.com	EMUL51 (to 42 MHz)	Now
Phyton, Inc. 7206 Bay Parkway, 2nd Floor Brooklyn, NY 11204 Phone: 718.259.3191 Fax: 718.259.1539 Web: www.phyton.com	PICE-51 (to 40 MHz)	Now

Evaluation Kits

Company	Tools	Availability
PHYTEC America LLC 755 Winslow Way E, Suite 302 Bainbridge Island, WA 98110 Phone: 206.780.9047 Toll Free: 800.278.9913 Fax: 206.780.9135 Web: www.phytec.com **PHYTEC Meßtechnik GmbH** Robert-Koch-Straße 39 D-55129 Mainz, Germany Phone: +49.6131.9221.0 Fax: +49.6131.9221.33	KitCON-FlashFlex51	Now