Mobile Platform Controller plus 8 Mbit LPC Firmware Flash
SST79LF008

Advance Information

FEATURES:

- **8 Mbit LPC Firmware Memory SuperFlash device with integrated LPC Keyboard, System configuration, and Power Management controller**
- **ACPI 2.0 Compliant**
- **Conforms to LPC Interface Specification v1.1**
  - Includes support for Multi-byte Firmware Memory Read/Write Cycles
    - Firmware Memory 1-, 2-, 4-, 16-, and 128-byte Read Cycles
    - Firmware Memory 1-, 2-, and 4-byte Write Cycles
  - 15.7 MB/sec data transfer rate @ 33MHz clock for Multi-Byte Read
  - One ID pin for LPC Firmware Memory Device selection
- **LPC Firmware Memory**
  - 8 Mbit Single Block of on-chip SuperFlash memory with two Shared-ROM modes
    - Mode 1: 7 Mbit (896 KByte) for system BIOS and 1 Mbit (128 KByte) for 8051 firmware
    - Mode 2: 7.5 Mbit (960 KByte) for system BIOS and 0.5 Mbit (64 KByte) for 8051 firmware
  - Uniform 4 KByte Sectors and 64 KByte Blocks with Erase capability
  - 19 Lockable Blocks: one 16 KByte Boot Block, two 8 KByte Parameter Blocks, one 32 KByte Parameter Block, fifteen 64 KByte Main Blocks
    - Block Locking Registers for individual block Read-Lock, Write-Lock, and Lock Down protection
  - Lockable bottom 4 KByte sector for 8051 boot firmware
  - Erase-Suspend allowing Read or Program of the other blocks
  - Two-Cycle Command Set
- **Non-Volatile Registers (NVR)**
  - 64-bit SST Pre-Programmed Identifier
  - 192-bit OTP User Unique Identifier with Write-Lock protection
  - 3 KByte OTP User NVR area (UNVR)
  - 4 KByte Erasable NVR area (ENVR) with Write-/Read-Lock protection
- **Superior Reliability**
  - Endurance: 100,000 Cycles (typical)
  - Greater than 100 years Data Retention
- **Fast Erase/Program Operations**
  - Sector-Erase Time: 55 ms (typical)
  - Block-Erase Time: 55 ms (typical)
  - Word-Program Time: 15 µs (typical)
- **aLPC mode for Rapid Factory Programming**
  - Alternate LPC bus (aLPC) for in-system and factory programming
  - Auto Address Increment (AAI)
  - Multi-Byte Program
  - Chip Rewrite Time: 12 seconds (typical)
- **Embedded Enhanced 8051 MCU**
  - 3- or 6-clock (selectable) per-instruction cycle
  - Up to 33 MHz 8051 operating frequency
  - Up to 128 KByte Program Address Space
  - 256 Byte standard 8051 RAM
  - 2 KByte on-chip expanded Data RAM / Executable RAM (Scratch ROM)
  - Extended up to 2 KByte Stack Space
  - Four Levels of Interrupt Priorities and Twelve Interrupt Vectors
  - Power-saving IDLE and Power-Down modes
  - Multiple Maskable Hardware Wake-up Events (sources include: Hibernation timer, LPC, serial interfaces, all GPIOs, and others)
- **LPC Host Interfaces**
  - One 8042-style legacy KBC interface channel
  - Two ACPI EC interface channels
  - 32 8-bit LPC Host-to-8051 Mailbox Registers
  - Programmable Base addresses for all channels
- **System Interrupts**
  - IRQ1 and IRQ12 via serialized IRQ Interface
  - Two EC SCI event outputs
  - SMI via Serialized IRQ2 or SMI event output
- **Hardware GA20 and CPU Reset Outputs Control**
- **16 x 8 (24 pins) Key Scan Matrix expandable to 16 x 14 (30 pins)**
- **Three Independent PS/2 Ports**
  - Hardware driven receive and transmit protocols
  - Integrated time-out control
- **Two SMBus controllers/Three SMBus channels**
  - SMBus 2.0 compliant
  - Master and Slave operation
  - Internal multiplexer for SMBus channel selection
- **Full-Duplex Enhanced UART channel**
- **SPI Master/Slave channel**

Advance Information

- **Eight-channel ADC with 10-bit resolution**
- **Four-channel DAC with 8-bit resolution**
- **Two 8-bit Fan Tachometer channels with clock prescaler**
- **Three PWM channels with 8-bit resolution and independent prescaler**
- **Five direct LED control channels with blinking capability**
- **Watchdog Timer**
- **Hibernation Timer**
- **Three 16-bit Timers/Counters**
- **Configurable 5-Volt Tolerant General Purpose I/O Ports (GPIO)**
  - 112 GPIOs with 35 dedicated (non-multiplexed with alternative function)
  - Any dedicated GPIO or GPIO with disabled alternative function can be configured as Edge-Trigger maskable Interrupt and/or Wake Up event
- **Clocks**
  - Standard 32.768 KHz crystal oscillator
  - 10 to 20MHz fail-safe internal ring oscillator (automatic switch-on if Power-Good signal is de-asserted)
  - Up to 33MHz core clock derived directly from the external clock input or via internal PLL
- **Single 3.0-3.6V operation with 5V tolerant I/O (except LPC bus and analog I/O)**
- **Low Power Consumption**
  - Idle Mode supply current: 17mA (typical)
  - Power-Down mode supply current: 100 µA (typical)
- **Temperature Range: 0°C to 70°C**
- **Packages Available**
  - 176-lead LQFP
  - 176-ball TFBGA
- **All non-Pb (lead-free) devices are RoHS Compliant**

PRODUCT DESCRIPTION

The SST79LF008 is a high-performance LPC flash device with integrated PC Keyboard/Auxiliary device controller (KBC) and ACPI embedded controller (EC). This product is well suited for a wide range of mobile internet computing applications which require high integration (small form factor), superior power, and thermal management capability.

SST79LF008 includes 8Mbit of SuperFlash memory, which can be used to store system BIOS as well as KBC/EC firmware. Either 128 KByte (1 Mbit) or 64 KByte (0.5 Mbit) of the SuperFlash memory can be allocated for the KBC/EC code providing, respectively, 896 KByte (7.0 Mbit) or 960 KByte (7.5 Mbit) for the system BIOS memory. The SST79LF008 features in-system programming, which provides maximum flexibility in the manufacturing environment as well as a mechanism for updating the keyboard firmware code, the main system BIOS code, and adding new functionality in the end-user environment in order to meet the latest market demands. It also speeds up software development and improves the overall time-to-market.

The SST79LF008 is manufactured with SST's proprietary, high-performance SuperFlash technology. The split-gate cell design and thick-oxide tunneling injector attain better reliability and manufacturing capability compared with alternate approaches. The device significantly improves performance and reliability, while lowering the power consumption.

The SST79LF008 is designed to be compatible with any LPC bus compatible host controllers, such as the ICHx or other south-bridge devices of PC chipsets for PC-BIOS application. It provides several mechanisms controlled by KBC/EC firmware and/or LPC host for code and data storage protection. SST79LF008 also includes an additional 4 Kbyte of lockable, open-after-reset SuperFlash memory, which can be used as secure ENVR storage.

SST79LF008 on-chip peripherals, including PS/2 ports, Matrix scanner, SMBus controllers, and ADC/DAC/PWM with flexible GPIO configuration, provide necessary hardware support for the KBC/EC functions on the mobile PC platforms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 FUNCTIONAL BLOCKS

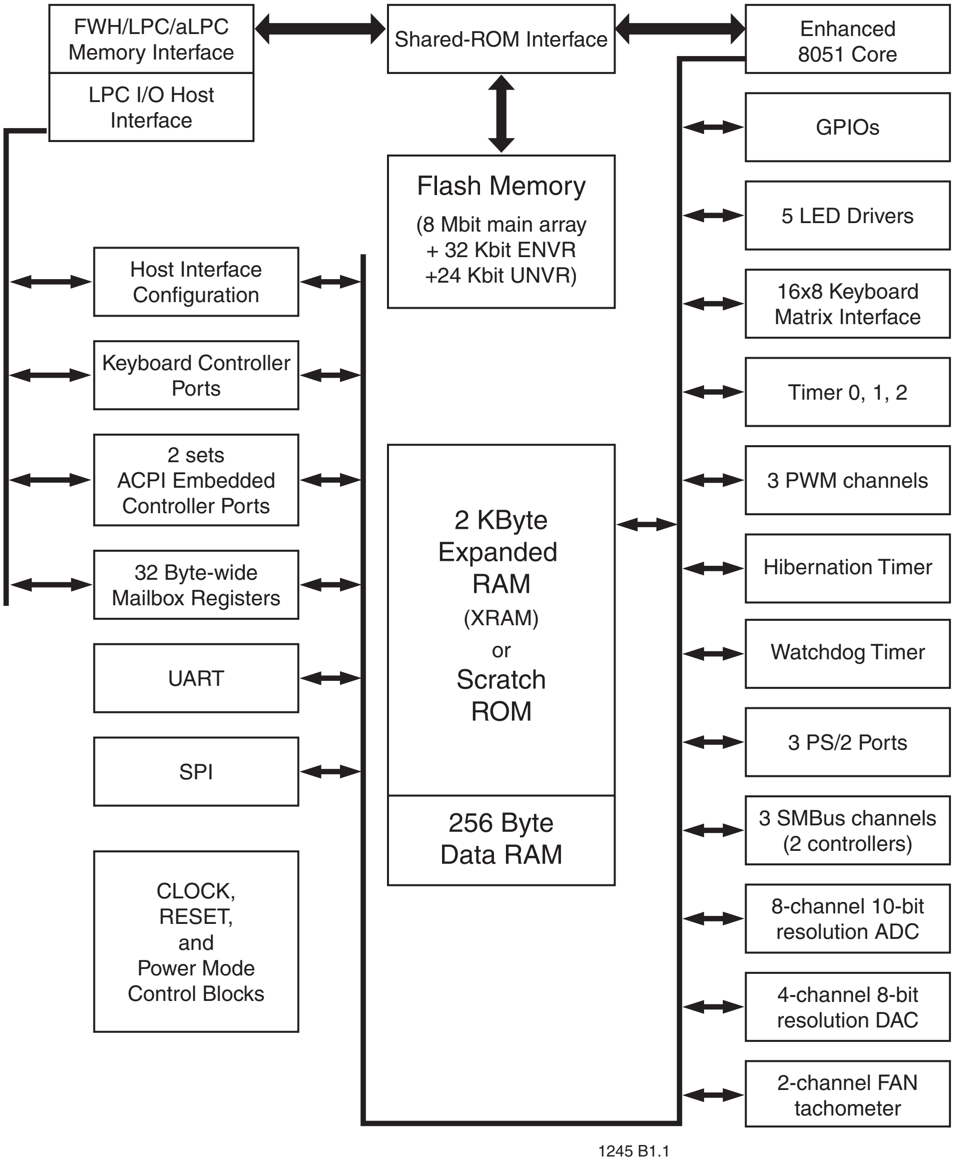
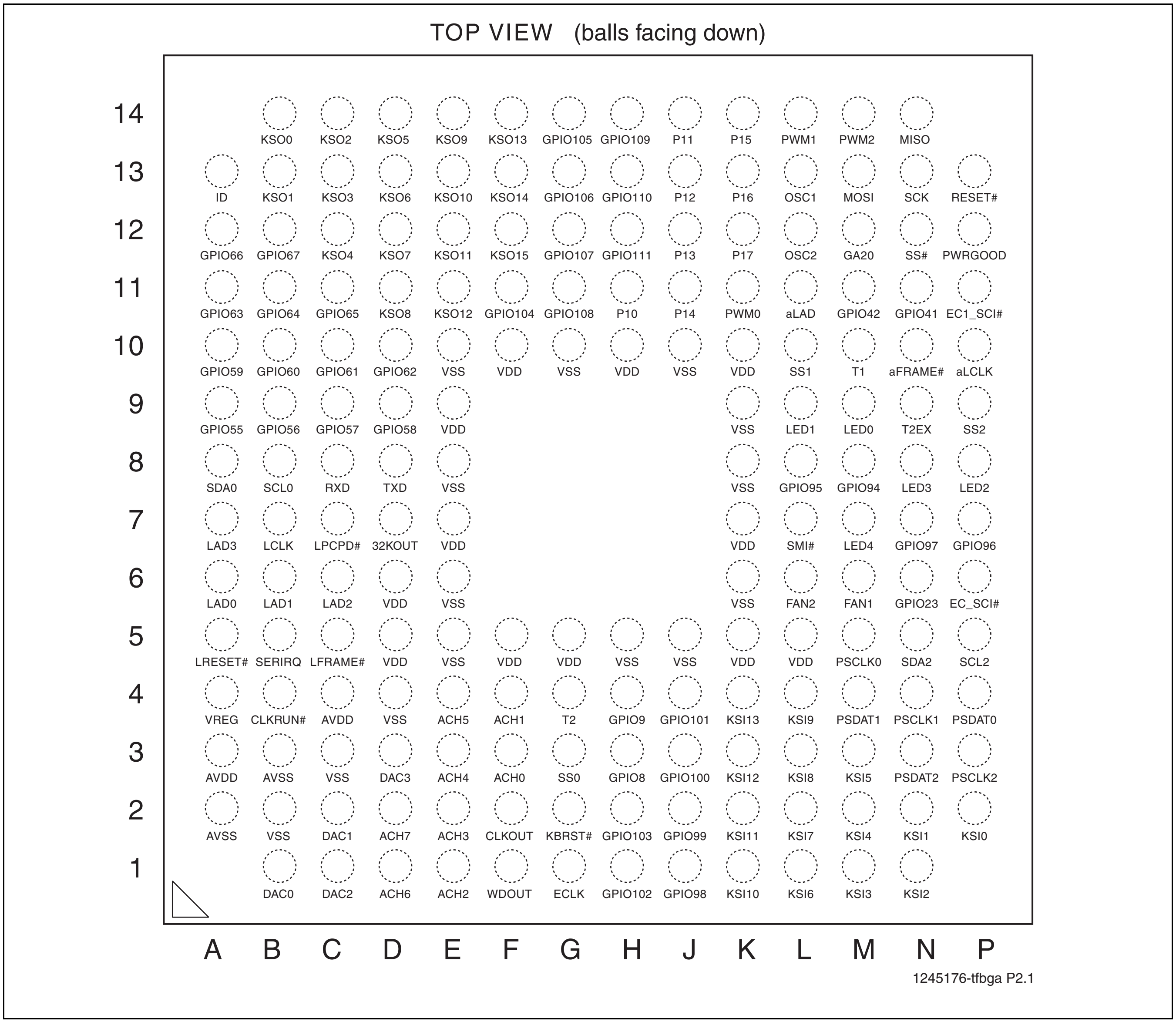

FWH/LPC/aLPC Memory Interface
LPC I/O Host Interface
Shared-ROM Interface
Enhanced 8051 Core
Flash Memory
(8 Mbit main array + 32 Kbit ENVR +24 Kbit UNVR)
Host Interface Configuration
Keyboard Controller Ports
2 sets ACPI Embedded Controller Ports
32 Byte-wide Mailbox Registers
UART
SPI
CLOCK, RESET, and Power Mode Control Blocks
2 KByte Expanded RAM (XRAM) or Scratch ROM
256 Byte Data RAM
GPIOs
5 LED Drivers
16x8 Keyboard Matrix Interface
Timer 0, 1, 2
3 PWM channels
Hibernation Timer
Watchdog Timer
3 PS/2 Ports
3 SMBus channels (2 controllers)
8-channel 10-bit resolution ADC
4-channel 8-bit resolution DAC
2-channel FAN tachometer
1245 B1.1

FIGURE 1-1: SST79LF008 Functional Block Diagram

2.0 PIN ASSIGNMENTS

The signal/pin assignments are listed in Table 2-1. Low active signals have a “#” suffix. I/O buffer types are listed in Table 2-2. Section 24.0 defines the DC characteristics for all input and output buffers.



TOP VIEW (balls facing down)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P
14		KSO0	KSO2	KSO5	KSO9	KSO13	GPIO105	GPIO109	P11	P15	PWM1	PWM2	MISO	
13	ID	KSO1	KSO3	KSO6	KSO10	KSO14	GPIO106	GPIO110	P12	P16	OSC1	MOSI	SCK	RESET#
12	GPIO66	GPIO67	KSO4	KSO7	KSO11	KSO15	GPIO107	GPIO111	P13	P17	OSC2	GA20	SS#	PWRGOOD
11	GPIO63	GPIO64	GPIO65	KSO8	KSO12	GPIO104	GPIO108	P10	P14	PWM0	aLAD	GPIO42	GPIO41	EC1_SCI#
10	GPIO59	GPIO60	GPIO61	GPIO62	VSS	VDD	VSS	VDD	VSS	VDD	SS1	T1	aFRAME#	aLCLK
9	GPIO55	GPIO56	GPIO57	GPIO58	VDD					VSS	LED1	LED0	T2EX	SS2
8	SDA0	SCL0	RXD	TXD	VSS					VSS	GPIO95	GPIO94	LED3	LED2
7	LAD3	LCLK	LPCPD#	32KOUT	VDD					VDD	SMI#	LED4	GPIO97	GPIO96
6	LAD0	LAD1	LAD2	VDD	VSS					VSS	FAN2	FAN1	GPIO23	EC_SCI#
5	LRESET#	SERIRQ	LFRAME#	VDD	VSS	VDD	VDD	VSS	VSS	VDD	VDD	PSCLK0	SDA2	SCL2
4	VREG	CLKRUN#	AVDD	VSS	ACH5	ACH1	T2	GPIO9	GPIO101	KSI13	KSI9	PSDAT1	PSCLK1	PSDAT0
3	AVDD	AVSS	VSS	DAC3	ACH4	ACH0	SS0	GPIO8	GPIO100	KSI12	KSI8	KSI5	PSDAT2	PSCLK2
2	AVSS	VSS	DAC1	ACH7	ACH3	CLKOUT	KBRST#	GPIO103	GPIO99	KSI11	KSI7	KSI4	KSI1	KSI0
1		DAC0	DAC2	ACH6	ACH2	WDOUT	ECLK	GPIO102	GPIO98	KSI10	KSI6	KSI3	KSI2	

1245176-tfbga P2.1

FIGURE 2-1: Pin Assignments for 176-ball TFBGA

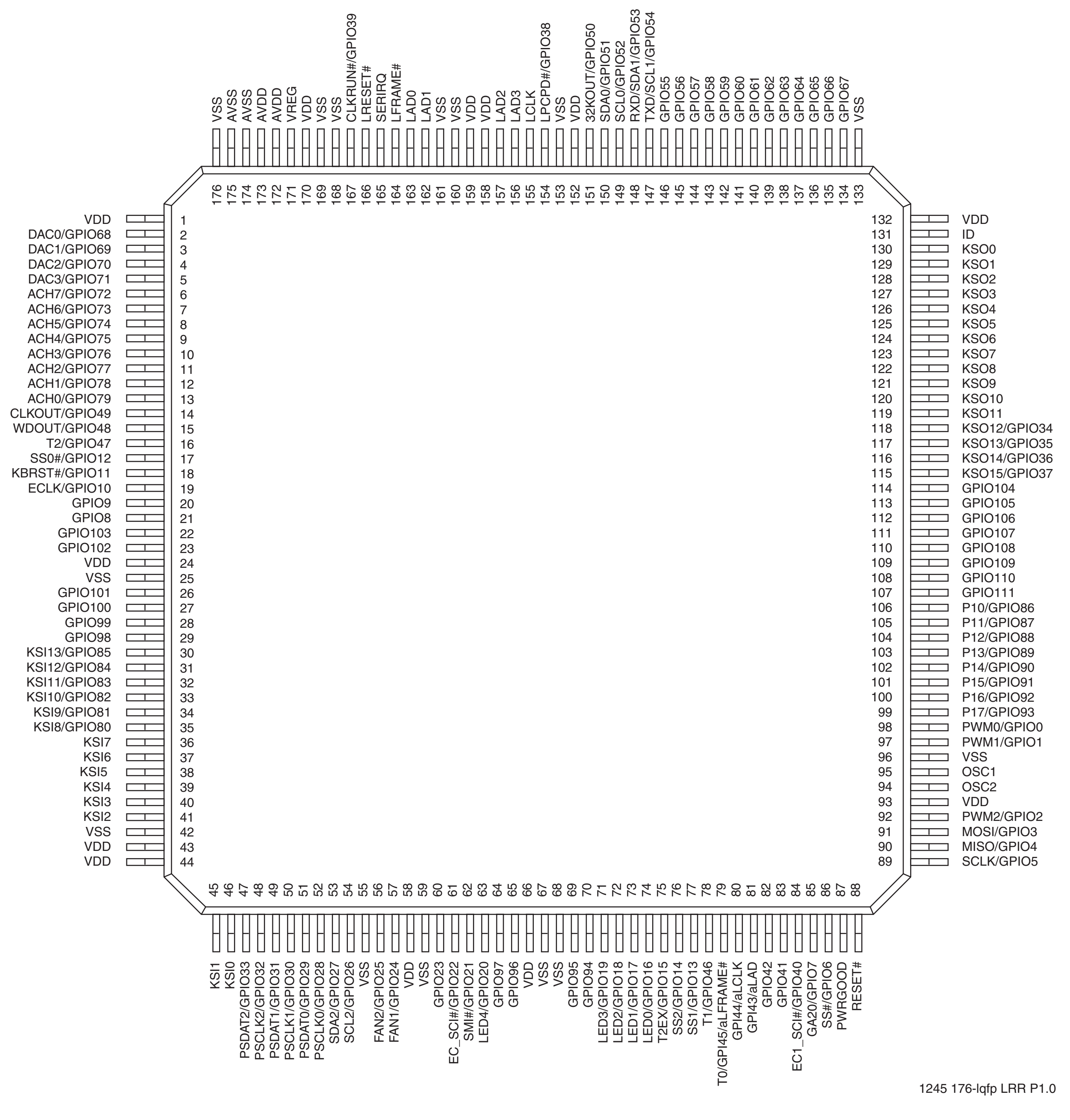


FIGURE 2-2: Pin Assignments for 176-lead LQFP

2.1 Pin Descriptions

The pins and functions of each pin in Table 2-1 are organized by each pin's major function. However, some pins are multiplexed with GPIO function.

TABLE 2-1: Pin Descriptions (1 of 6)

Symbol	I/O Buffer Type	176-leads or balls		Number of pins	Name and Functions
		LQFP	TFBGA		
LPC Bus Interface(10)					
LAD3 LAD2 LAD1 LAD0	IOPCI	156 157 162 163	A7 C6 B6 A6	4	**LPC Address/Data bus LAD[3:0]:** Multiplexed command, address and data bi-directional bus signals.
LCLK	IPCI	155	B7	1	**LPC Clock:** LPC bus clock input signal (commonly the same as PCI clock, up to 33MHz)
LFRAME#	IPCI	164	C5	1	**LPC Frame:** LPC bus control input signal. Low pulse indicates the start of the LPC transfer cycle when the bus is idle, or the termination (abort) of the broken LPC cycle already in progress.
LPCPD# / GPIO38	IOD6	154	C7	1	**LPC Power Down:** LPC Power down input signal, or GPIO Port. Indicates that power will be removed from the LPC bus.
CLKRUN# / GPIO39	IODPCI	167	B4	1	**Clock Run:** PCI Clock Control signal, or GPIO Port. Indicates when LPC clock is (about to be) stopped by the LPC host, and provides a mechanism for the LPC device to request clock re-start.
LRESET#	IPCI	166	A5	1	**LPC Reset:** LPC bus reset input signal. This signal is used to reset the LPC interface control logic.
SERIRQ	IOPCI	165	B5	1	**Serialized IRQ:** This signal is used to generate serialized interrupts from the SST79LF008 to the LPC host.
Keyboard Matrix Interface (30)					
KSO15/GPIO37 KSO14/GPIO36 KSO13/GPIO35 KSO12/GPIO34	IOD4	115 116 117 118	F12 F13 F14 E11	4	**KSO[15:12] Keyboard Scan Outputs:** Keyboard Scan output pins, or GPIO Port

TABLE 2-1: Pin Descriptions (Continued) (2 of 6)

Symbol	I/O Buffer Type	176-leads or balls		Number of pins	Name and Functions
		LQFP	TFBGA		
KSO11 KSO10 KSO9 KSO8 KSO7 KSO6 KSO5 KSO4 KSO3 KSO2 KSO1 KSO0	OD4	119 120 121 122 123 124 125 126 127 128 129 130	E12 E13 E14 D11 D12 D13 D14 C12 C13 C14 B13 B14	12	**KSO[11:0] Keyboard Scan Outputs:** Keyboard Scan output pins only
KSI13/GPIO85 KSI12/GPIO84 KSI11/GPIO83 KSI10/GPIO82 KSI9/GPIO81 KSI8/GPIO80	SIO4_PU	30 31 32 33 34 35	K4 K3 K2 K1 L4 L3	6	**KSI[13:8] Extended Keyboard Scan Inputs:** 6 individually selectable Keyboard Scan input pins (with internal pull-ups and wake up interrupts). Each of these pins can be also configured as GPIO with programmable pull-up resistors (disabled after reset).
KSI7 KSI6 KSI5 KSI4 KSI3 KSI2 KSI1 KSI0	SI_PU	36 37 38 39 40 41 45 46	L2 L1 M3 M2 M1 N1 N2 P2	8	**KSI[7:0] Keyboard Scan Inputs:** 8 Keyboard Scan input pins with internal pull-up resistors and wake up interrupts (pull-ups are always enabled).
PS/2 Interface (6)					
PSDAT2/GPIO33 PSDAT1/GPIO31 PSDAT0/GPIO29	SIOD15	47 49 51	N3 M4 P4	3	**PSDAT[2:0] PS/2**:Channel 2 to 0 data signal, or GPIO Port
PSCLK2/GPIO32 PSCLK1/GPIO30 PSCLK0/GPIO28	SIOD15	48 50 52	P3 N4 M5	3	**PSCLK[2:0] PS/2**: Channel 2 to 0 clock signal, or GPIO Port
SMBus and UART Interface (6)					
SDA0/GPIO51	SIO6	150	A8	1	**SMBus:** Channel 0 data signal, or GPIO port
SCL0/GPIO52	SIO6	149	B8	1	**SMBus:** Channel 0 clock signal, or GPIO port
RXD/SDA1/GPIO53	SIO6	148	C8	1	**UART:** Receive data input, or **SMBus** channel 1 data signal, or GPIO port
TXD/SCL1/GPIO54	SIO6	147	D8	1	**UART:** Transmit data output, or **SMBus** channel 1 clock signal, or GPIO port
SDA2/GPIO27	SIO6	53	N5	1	**SMBus:** Channel 2 data signal, or GPIO port
SCL2/GPIO26	SIO6	54	P5	1	**SMBus:** Channel 2 clock signal, or GPIO port

TABLE 2-1: Pin Descriptions (Continued) (3 of 6)

Symbol	I/O Buffer Type	176-leads or balls		Number of pins	Name and Functions
		LQFP	TFBGA		
SPI interface (4)					
MOSI/GPIO3	IO5	91	M13	1	**SPI Data:** Master data output line, slave data Input line, or GPIO port
MISO/GPIO4	IO5	90	N14	1	**SPI Data:** Master data input line, slave data output line, or GPIO Port
SCK/GPIO5	IO5	89	N13	1	**SPI Clock:** Master clock output line, slave clock input line, or GPIO Port
SS#/GPIO6	IO5	86	N12	1	**SPI Port Select:** Slave port select input, or GPIO Port
aLPC bus, Timers, PWMs and Fan Tachometers (12)					
GPI43/aLAD	AIO4	81	L11	1	**aLPC DATA:** Alternative LPC Address/Data Bus or GPI Port
GPI44/aLCLK	AIO4	80	P10	1	**aLPC Clock:** Alternative LPC clock or GPI Port
T0/GPI45/aLFRAME#	IO5	79	N10	1	**Timer0 Counter Input:** External count input to Timer/Counter 0, or GPI Port, or aLPC frame: alternative LPC Frame
T1/GPIO46	IO5	78	M10	1	**Timer1 Counter Input or Output:** External count input to Timer/Counter 1, or Clock output from Timer/Counter 1, or GPIO Port
T2/GPIO47	IO5	16	G4	1	**Timer2 Counter Input or Output:** External count input to Timer/Counter 2, or Clock output from Timer/Counter 2, or GPIO Port
WDOUT/GPIO48	IO5	15	F1	1	**Watchdog Timer Output:** Watchdog Timer Output, or GPIO Port
T2EX/GPIO15	IO5	75	N9	1	**Timer 2 External Input:** External interrupt input to Timer 2, or GPIO Port
PWM2/GPIO2 PWM1/GPIO1 PWM0/GPIO0	IO5	92 97 98	M14 L14 K11	3	**PWM[2:0] PWM Output:** PWM output 2 to 0, or GPIO Port
FAN2/GPIO25 FAN1/GPIO24	SIO6	56 57	L6 M6	2	**FAN[2:1] Fan Input:** Fan Tachometer input 2 to 1, or GPIO port
LED Drivers (5)					
LED4/GPIO20 LED3/GPIO19 LED2/GPIO18 LED1/GPIO17 LED0/GPIO16	SIOD15	63 71 72 73 74	M7 N8 P8 L9 M9	5	**LED[4:0] LED Driver** output, or GPIO Port

TABLE 2-1: Pin Descriptions (Continued) (4 of 6)

Symbol	I/O Buffer Type	176-leads or balls		Number of pins	Name and Functions
		LQFP	TFBGA		
Miscellaneous and Dedicated GPIOs (52)					
EC1_SCI#/GPIO40	SIOD15	84	P11	1	**Embedded Controller Interrupt:** Embedded controller interrupt output for port 68H/6CH Host interface, or GPIO Port
SMI#/GPIO21	SIOD15	62	L7	1	**System Management Interrupt:** System management interrupt output, or GPIO Port
EC_SCI#/GPIO22	SIOD15	61	P6	1	**Embedded Controller Interrupt:** Embedded controller interrupt output for port 62H/66H Host interface, or GPIO Port
GPIO23	SIO6	60	N6	1	**GPIO Port**
GA20/GPIO7	IO5	85	M12	1	**Gate A20:** GA20 output, or GPIO Port
KBRST#/GPIO11	IO5	18	G2	1	**KBC Reset to CPU:** Keyboard Controller reset output, or GPIO Port
SS2/GPIO14 SS1/GPIO13 SS0/GPIO12	IO5	76 77 17	P9 L10 G3	3	**SS[2:0] LPC Host Status Signals:** LPC IO Host interface status signal output, or GPIO Port
GPIO8 GPIO9 GPIO41 GPIO42 GPIO55	IO5 IO5 IO5 IO5 SIO6	21 20 83 82 146	H3 H4 N11 M11 A9	5	GPIO Port
GPIO67 GPIO66 GPIO65 GPIO64 GPIO63 GPIO62 GPIO61 GPIO60 GPIO59 GPIO58 GPIO57 GPIO56	IO5	134 135 136 137 138 139 140 141 142 143 144 145	B12 A12 C11 B11 A11 D10 C10 B10 A10 D9 C9 B9	12	GPIO Port
P17/GPIO93 P16/GPIO92 P15/GPIO91 P14/GPIO90 P13/GPIO89 P12/GPIO88 P11/GPIO87 P10/GPIO86	IO5	99 100 101 102 103 104 105 106	K12 K13 K14 J11 J12 J13 J14 H11	8	**P[17:10]** 8051 Port1 or GPIO Port

TABLE 2-1: Pin Descriptions (Continued) (5 of 6)

	I/O Buffer	176-leads or balls		Number	
Symbol	**Type**	**LQFP**	**TFBGA**	**of pins**	**Name and Functions**
GPIO111	SIO4_PU	107	H12	18	**GPIO Port:** with internal programmable pull-up resistors (enabled after reset).
GPIO110		108	H13		
GPIO109		109	H14		
GPIO108		110	G11		
GPIO107		111	G12		
GPIO106		112	G13		
GPIO105		113	G14		
GPIO104		114	F11		
GPIO103		22	H2		
GPIO102		23	H1		
GPIO101		26	J4		
GPIO100		27	J3		
GPIO99		28	J2		
GPIO98		29	J1		
GPIO97		64	N7		
GPIO96		65	P7		
GPIO95		69	L8		
GPIO94		70	M8		
Analog Interface (12)					
DAC3/GPIO71	AIO4	5	D3	4	**DAC[3:0] DAC:** Digital to Analog Converter outputs, or GPIO Port
DAC2/GPIO70		4	C1		
DAC1/GPIO69		3	C2		
DAC0/GPIO68		2	B1		
ACH7/GPIO72	AIO4	6	D2	8	**ACH[7:0] ADC:** Analog to Digital Converter input, or GPIO Port
ACH6/GPIO73		7	D1		
ACH5/GPIO74		8	E4		
ACH4/GPIO75		9	E3		
ACH3/GPIO76		10	E2		
ACH2/GPIO77		11	E1		
ACH1/GPIO78		12	F4		
ACH0/GPIO79		13	F3		
Clocks (5)					
OSC1	OSC	95	L13	1	**OSC1 and OSC2:** Input and Output of the internal inverting oscillator amplifier. These 2 pins are connected to the external 32.768KHz crystal circuit.
OSC2		94	L12	1	
32KOUT/GPIO50	IO5	151	D7	1	**32KHz Clock:** Output of 32.768Khz clock signal, or GPIO Port
ECLK/GPIO10	IO5	19	G1	1	**ECLK:** External Clock input, or GPIO Port
CLKOUT/GPIO49	IO5	14	F2	1	**Core Clock Output:** 8051 core clock output, or GPIO Port

TABLE 2-1: Pin Descriptions (Continued) (6 of 6)

Symbol	I/O Buffer Type	176-leads or balls		Number of pins	Name and Functions
		LQFP	TFBGA		
Reset, ID and Power (34)					
PWRGOOD	I	87	P12	1	**Power Good:** Input used to select clock source
RESET#	SI_PU	88	P13	1	**Reset:** By setting the RESET# pin low, the entire device is reset to a default state (internal pull-up resistor always enabled).
ID	I_PD	131	A13	1	**Firmware Memory ID:** LPC Firmware memory ID selection input (internal pull-down resistor always enabled).
V_{DD}	PWR	1, 24, 43, 44 58, 66, 93, 132, 152, 158, 159, 170	F5, G5, K5, L5, K7, K10, H10, F10, E9, E7, D6, D5	12	**Digital Power supply 3.3V:** (power supply voltage must be applied to all V_{DD} pins)
AV_{DD}	PWR	172,173	C4, A3	2	**Analog Power Supply 3.3V:** (analog power supply voltage must be applied to all AV_{DD} pins)
V_{SS}	PWR	25, 42, 55, 59, 67, 68, 96, 133, 153, 160, 161, 168, 169, 176	H5, J5, K6, K8, K9, J10, G10, E10, E8, E6, E5, D4, C3, B2	14	**Ground (GND):** 0V reference (ground plane must be connected to all V_{SS} pins)
AV_{SS}	PWR	174, 175	B3, A2	2	**Analog Ground:** For ADC and DAC (analog ground plane must be connected to all AV_{SS} pins as well as to GND plane)
VREG	PWR	171	A4	1	**Internal voltage regulator output:** An external ceramic capacitor (1µf) must be connected between this pin and system ground.

T2-1.1 1320

2.2 I/O Type Descriptions

I/O buffer types in Table 2-1 are described in Table 2-2. DC parameters are described in Section 24.0.

TABLE 2-2: I/O Buffer Types[1]

I/O Buffer Type	Description
AIO4	I/O with push-pull 4mA output multiplexed with Analog function
I	Input
I_PD	Input with internal pull-down
IO5	I/O with push-pull 5mA output
IOD4	Input, open drain output with 4mA sink capability
IOD6	Input, open drain output with 6mA sink capability
IODPCI	Input, open drain output, PCI compatible
IOPCI	I/O, PCI compatible
IPCI	Input, PCI compatible
OSC	32 kHz oscillator input and output
OD4	Open drain output with 4mA sink capability
SI_PU	Schmitt triggered input with internal pull-up
SIO4_PU	I/O with Schmitt triggered input, push-pull 4mA output, internal pull-up
SIO6	I/O with Schmitt triggered input, push-pull 6mA output
SIOD15	Schmitt triggered input, open drain output with 15mA sink capability
PWR	Power or Ground pin

T2-2.1 1320

[1] Any pin configured as input or OD high output without internal pull-up or pull-down resistor must not be left disconnected.

3.0 MEMORY ORGANIZATION

The on-chip 8051 MCU has separate address spaces for program and data memory, which are described in this section. The on-chip 8051 can also access, via Shared ROM Interface, a total of 1 MByte (8 Mbits) of the main Flash Memory array, 4 KByte of ENVR flash sector, and
3 KBytes of UNVR one-time programmable memory. The main Flash Memory array is divided into 16 blocks with 64 KByte block size, each block consists of 16 sectors with 4 KByte sector size. The entire main array as well as ENVR can be erased/programmed/read by the 8051 core using in-application programming mode commands specified in Section 4.0. However, only the two bottom blocks of the main array can be mapped into 8051 program space.

3.1 Program Memory

The SST79LF008 enhanced 8051 MCU can operate either in 16-bit addressing mode with firmware size up to 64 KByte, or in 17-bit addressing mode with firmware size up to 128 KByte. Flash memory Block 0 is mapped to 8051 program memory to store firmware code in 16-bit addressing mode, and Flash memory Blocks 0 and 1 are mapped to 8051 program memory to store firmware code in 17-bit addressing mode. For program memory organization, see Figure 3-1. The addressing mode is controlled by ACON register, described in Section 6.5.

Additionally, sections of the on-chip SRAM can be mapped into the 8051 program space providing a so called Scratch ROM area for code that can run during in-application programming. The size of Scratch ROM can be selected as 2 KByte, 1 KByte, 512 Byte, or 256 Byte. See Figures 3-2, 3-3, 3-4, and 3-5. Scratch ROM mapping and size is controlled by SCRROM register, defined in Section 4.4.1.

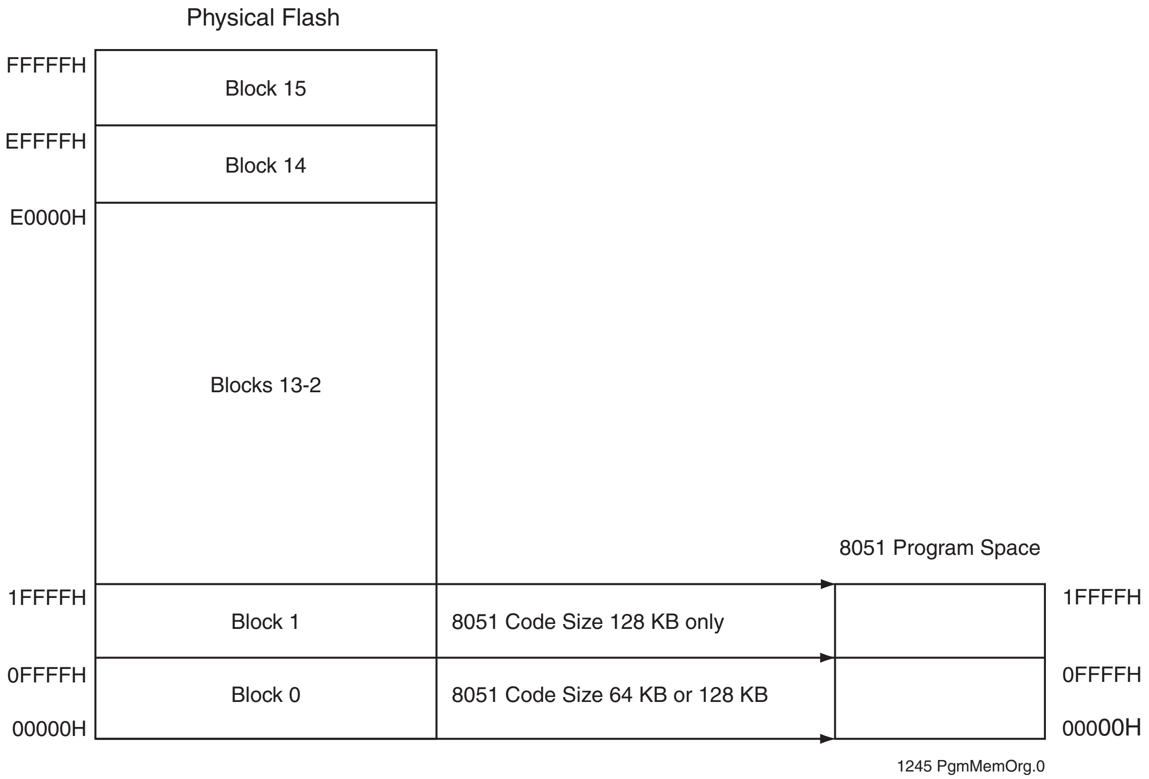


FIGURE 3-1: SST79LF008 Program Memory Organization

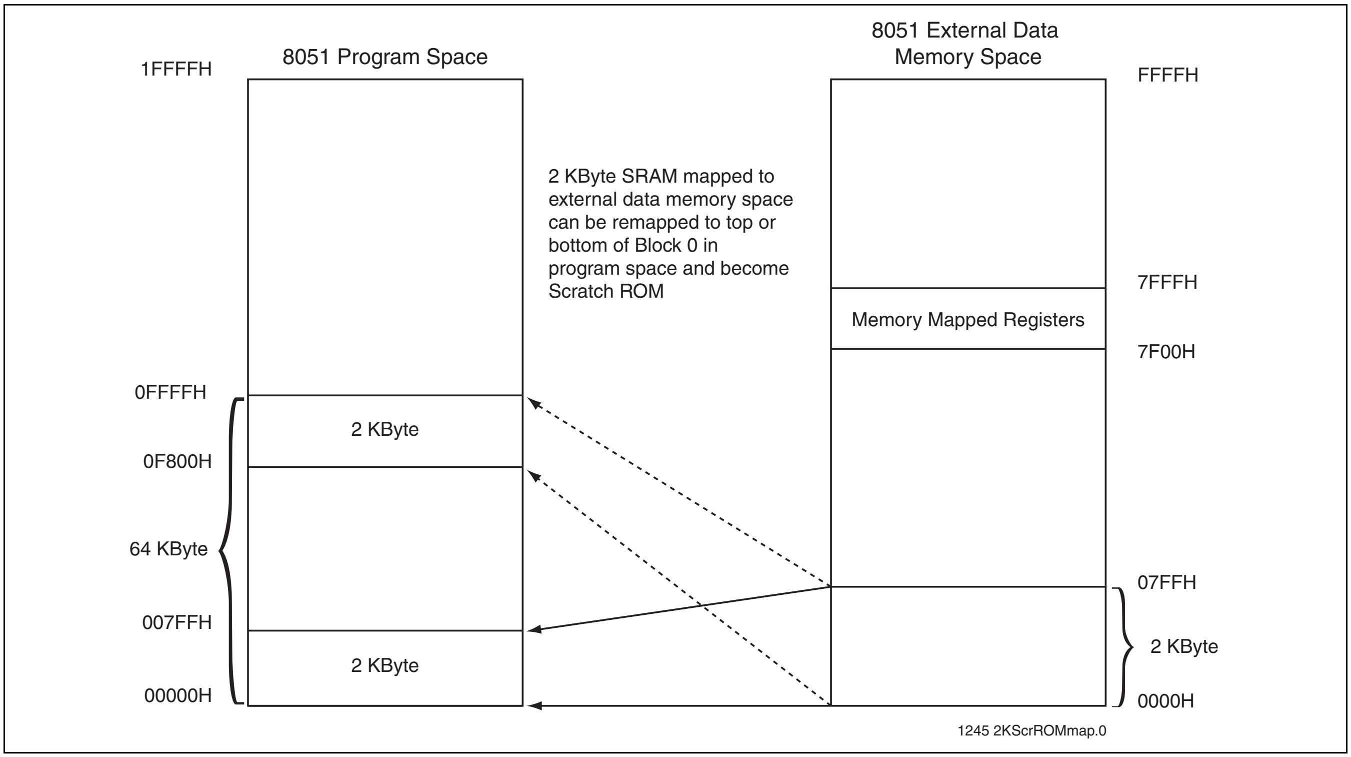

8051 Program Space
8051 External Data Memory Space
1FFFFH
0FFFFH
0F800H
64 KByte
007FFH
00000H
2 KByte
2 KByte
2 KByte SRAM mapped to external data memory space can be remapped to top or bottom of Block 0 in program space and become Scratch ROM
FFFFH
7FFFH
Memory Mapped Registers
7F00H
07FFH
2 KByte
0000H
1245 2KScrROMmap.0

FIGURE 3-2: 2 KByte Scratch ROM Mapping

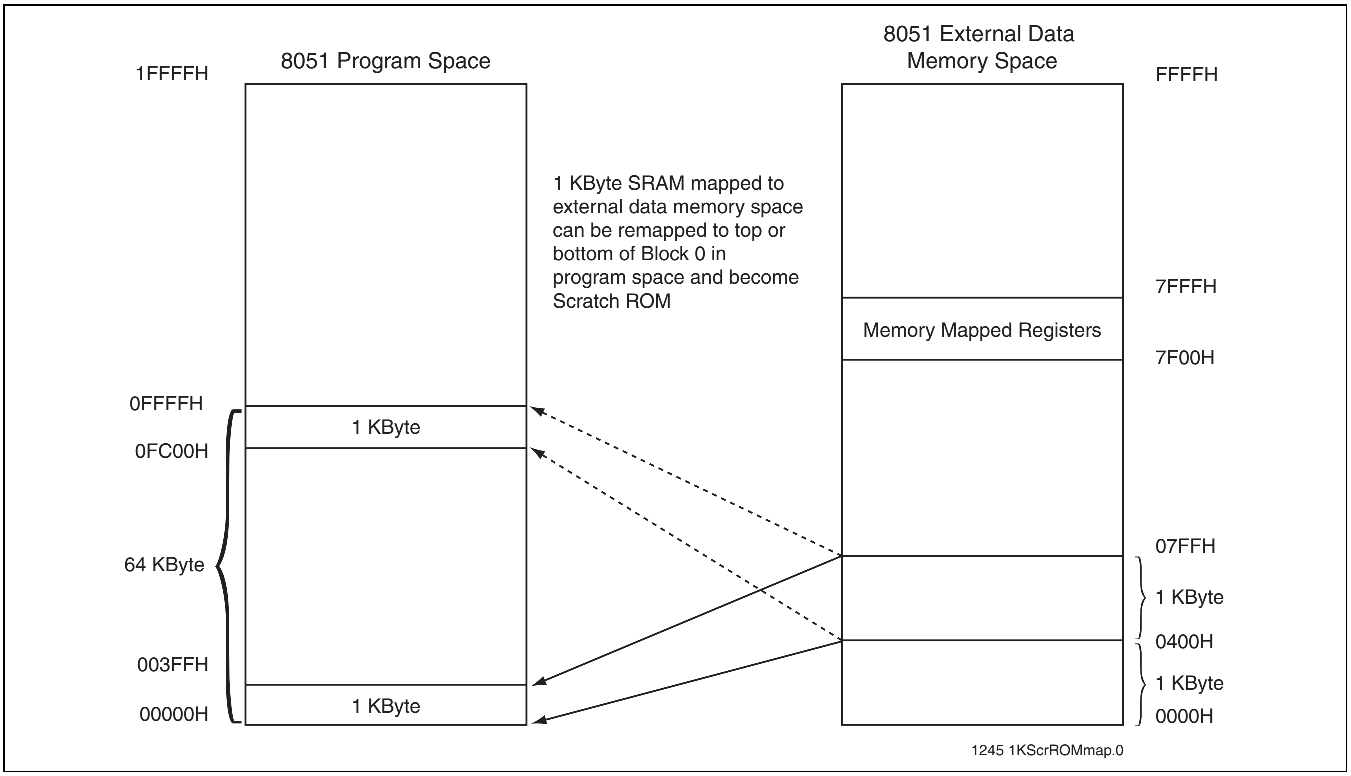

8051 Program Space
8051 External Data Memory Space
1FFFFH
0FFFFH
0FC00H
64 KByte
003FFH
00000H
1 KByte
1 KByte
1 KByte SRAM mapped to external data memory space can be remapped to top or bottom of Block 0 in program space and become Scratch ROM
FFFFH
7FFFH
Memory Mapped Registers
7F00H
07FFH
1 KByte
0400H
1 KByte
0000H
1245 1KScrROMmap.0

FIGURE 3-3: 1 KByte Scratch ROM Mapping

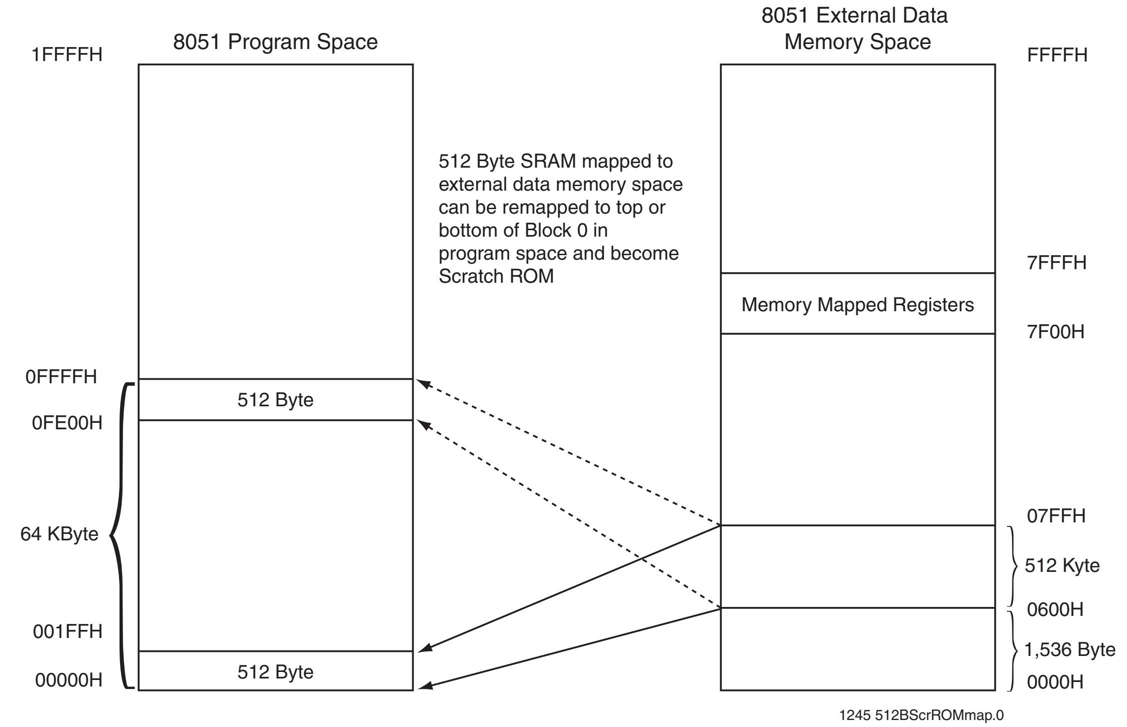

8051 Program Space
1FFFFH
0FFFFH
512 Byte
0FE00H
64 KByte
001FFH
512 Byte
00000H
512 Byte SRAM mapped to external data memory space can be remapped to top or bottom of Block 0 in program space and become Scratch ROM
8051 External Data Memory Space
FFFFH
7FFFH
Memory Mapped Registers
7F00H
07FFH
512 Kyte
0600H
1,536 Byte
0000H
1245 512BScrROMmap.0

FIGURE 3-4: 512 Byte Scratch ROM Mapping

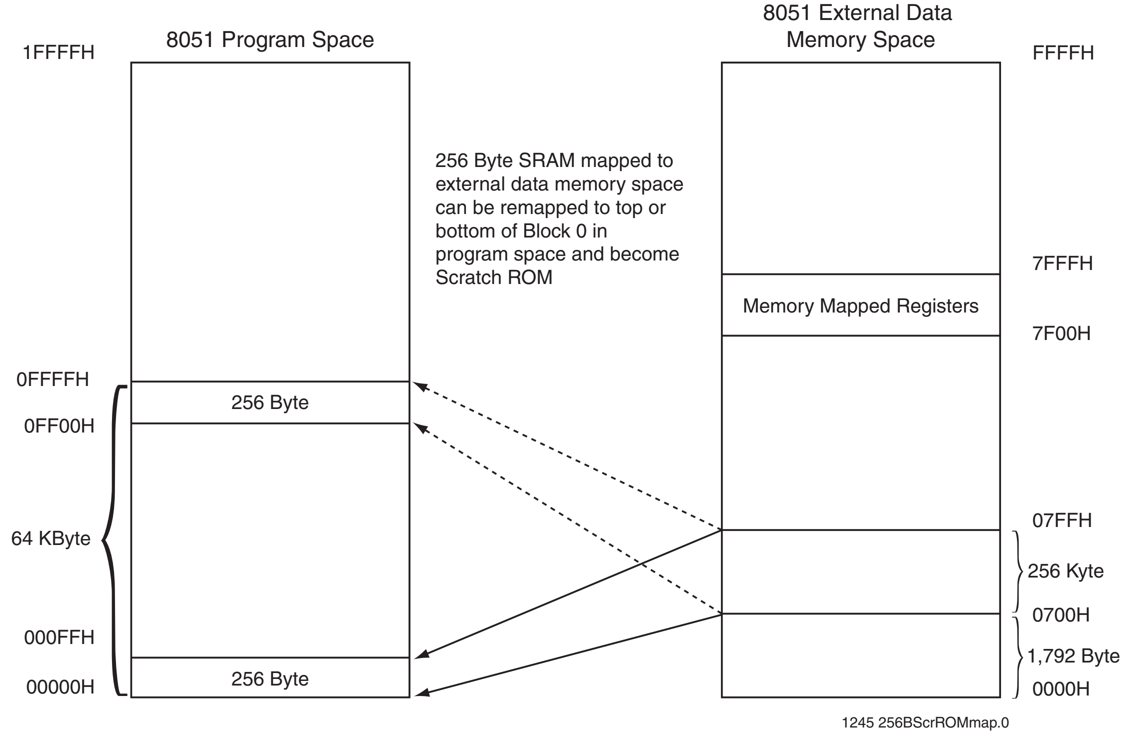

8051 Program Space
1FFFFH
0FFFFH
256 Byte
0FF00H
64 KByte
000FFH
256 Byte
00000H
256 Byte SRAM mapped to external data memory space can be remapped to top or bottom of Block 0 in program space and become Scratch ROM
8051 External Data Memory Space
FFFFH
7FFFH
Memory Mapped Registers
7F00H
07FFH
256 Kyte
0700H
1,792 Byte
0000H
1245 256BScrROMmap.0

FIGURE 3-5: 256 Byte Scratch ROM Mapping

3.2 Data Memory

The on-chip 2304-Byte SRAM can be divided into three sections:

1. Lower RAM - SRAM section mapped to the lower 128 Bytes of 8051 Internal Data Memory space (00H to 7FH), that are directly and indirectly addressable.
2. Upper RAM - SRAM section mapped into the higher 128 Bytes of 8051 Internal Data Memory space (80H to FFH), that are indirectly addressable only.
3. Expanded RAM - SRAM section mapped into 2048 bytes of 8051 External Data Memory space (0000H to 07FFH) that are indirectly addressable via 8051 external memory access instructions only. (As described in Section 3.1, 256, 512, 1024 or 2048 Bytes of this SRAM section can be also remapped to the 8051 program address space providing a Scratch ROM area.)

The on-chip Special Function Registers (SFRs) are mapped into the higher 128 Bytes of 8051 Internal Data Memory space (80H to FFH), and thus overlapped with Upper RAM. However, unlike Upper RAM, all SFRs are directly addressable only.

The on-chip Memory Mapped Configuration registers (MMCRs) are mapped into 256 bytes of 8051 External Data Memory space (7F00H to 7FFFH) that are indirectly addressable via 8051 external memory access instructions only.

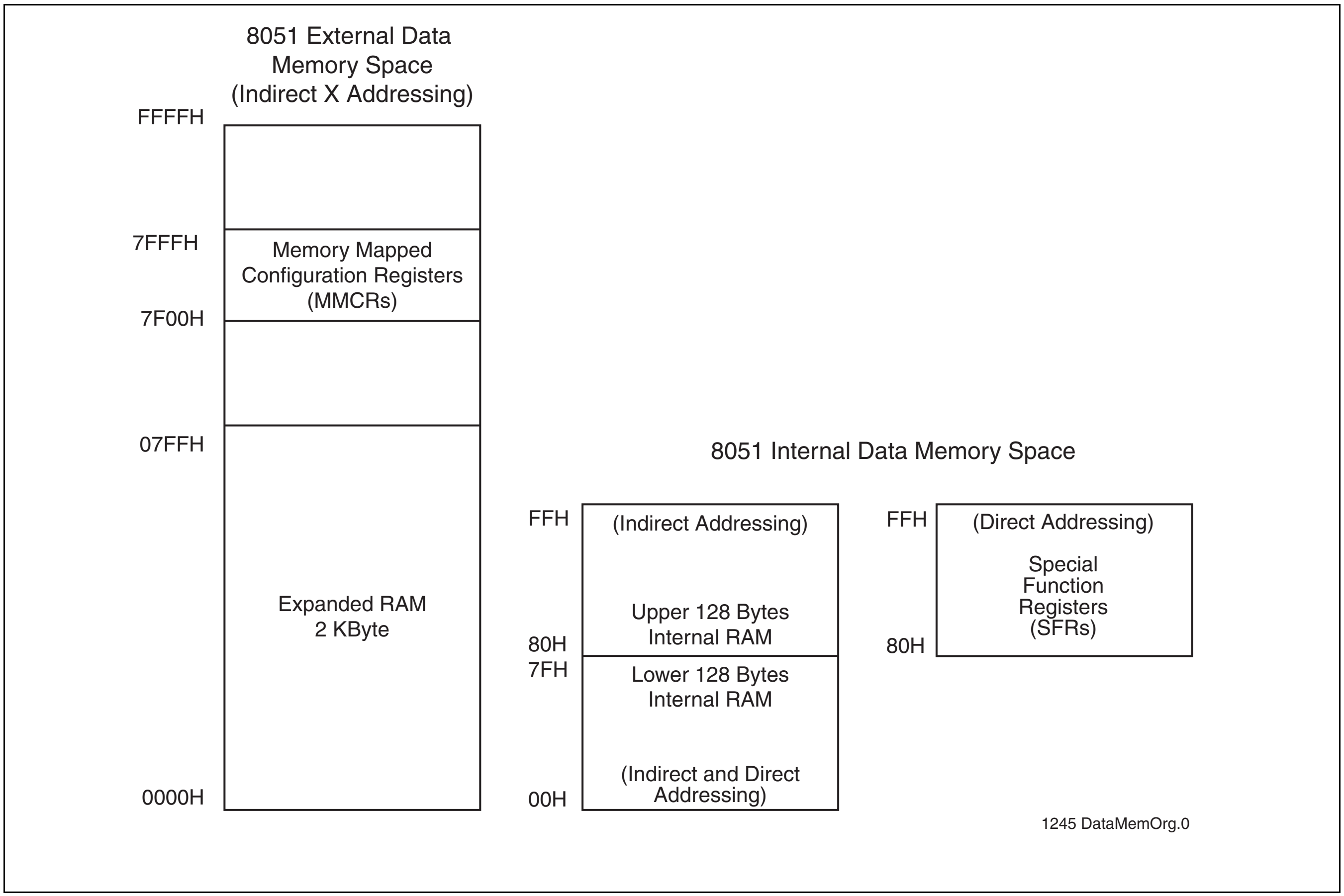


FIGURE 3-6: SST79LF008 Data Memory Organization

3.3 Data Memory Addressing Modes

The Lower RAM is accessed by all 8051 instructions that utilize both direct and indirect internal data memory addressing modes. In addition the lowest 32 bytes (00H-1FH) of Lower RAM are grouped into 4 banks of 8 registers, which can be accessed by 8051 register addressing mode. Bytes 20H-2FH provide a 128-bit addressable space, accessible by 8051 bit addressing mode (bit addresses 00H-7FH).

Because the Upper RAM occupies the same addresses as the SFRs, the respective data areas are distinguished by the type of addressing mode used: the Upper RAM is accessed via indirect internal data memory addressing mode, and the SFRs are accessed via direct internal data memory addressing mode. The SFRs that are located at addresses ended with 0H or 8H are also bit-addressable (bit addresses 80H-FFH).

The entire Expanded RAM and MMCRs are accessed by 8051 MOVX instructions that utilize indirect external data memory addressing mode and DPTR pointer. Using MOVX with R0/R1 indirect pointer can only access the lowest 256 bytes of total 2KB Expanded RAM.

The 8051 stack with default 8-bit addressing mode can be located anywhere within the 256 bytes of Lower and/or Upper RAM.

For additional details on 8051 addressing modes refer to the description of standard 8051 instruction set. For the SST79LF008 enhanced 8051 MCU features, see Section 6.0.

3.4 Special Function Registers (SFRs)

All Special Function Registers are located in 8051 internal data memory space, addresses 80H – FFH. For the detailed description of SFRs see the respective sections as specified in Tables 3-2, 3-3, 3-4, and 3-5. Some of SFRs contain reserved bits. On reads, software must not rely on reserved bits being any particular value. On writes, zeros should be written to reserved bits. When modifying a register with reserved bits the values read from the reserved bits can be written back to them. Software should not write to non used locations in SFR space.

3.4.1 SFR Map

TABLE 3-1: Special Function Register Memory Map

	8 BYTES								
F8H	IPA[1]								FFH
F0H	B[1]							IPAH	F7H
E8H	IEA[1]								EFH
E0H	ACC[1]								E7H
D8H									DFH
D0H	PSW[1]					SPCR			D7H
C8H	T2CON[1]	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CFH
C0H									C7H
B8H	IP[1]	SADEN							BFH
B0H	P3[1]							IPH	B7H
A8H	IE[1]	SADDR	SPSR	EXIF				CLKCON	AFH
A0H	P2[1]		AUXR1						A7H
98H	SCON[1]	SBUF		ESP		ACON			9FH
90H	P1[1]			DPX					97H
88H	TCON[1]	TMOD	TL0	TL1	TH0	TH1			8FH
80H	P0[1]	SP	DPL	DPH			SPDR	PCON	87H

T3-1.0 1320

[1] Bit addressable SFRs

Function and bit definitions for registers ACC, B, PSW, and P1 are the same as in standard 8051 MCU. Registers P0, P2, P3 are reserved. For the detailed description of all other SFRs see the respective sections as specified in the reference charts below.

3.4.2 SFR References

TABLE 3-2: Miscellaneous SFRs Reference

Register Name	Address	Reference Section	Bit Access Type								Reset Value[1]	Key
			7	6	5	4	3	2	1	0		
P1	90H	-	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
PSW	D0H	-	Q	R	Q	Q	Q	R	Q	R	00H	
ACC	E0H	-	Q	Q	Q	Q	Q	Q	Q	Q	00H	
B	F0H	-	Q	Q	Q	Q	Q	Q	Q	Q	00H	
CLKCON	AFH	6.3.1	-	-	-	-	Q	Q	Q	Q	xxxx0100b	
EXIF	ABH	8.3.1	-	-	-	-	R	R	R	R	x0H	
IE	A8H	8.3.2	Q	-	Q	Q	Q	Q	Q	Q	0x000000b	
IEA	E8H	8.3.3	-	-	-	-	Q	Q	Q	Q	x0H	
IP	B8H	8.3.4	Q	-	Q	Q	Q	Q	Q	Q	0x000000b	
IPH	B7H	8.3.5	Q	-	Q	Q	Q	Q	Q	Q	0x000000b	
IPA	F8H	8.3.6	-	-	-	-	Q	Q	Q	Q	x0H	
IPAH	F7H	8.3.7	-	-	-	-	Q	Q	Q	Q	x0H	
AUXR1	A2H	6.4.1	-	-	-	-	-	-	-	Q	xxxxxxx0b	
ACON	9DH	6.5.1	-	-	-	-	-	Q	Q	-	xxxxx00xb	
SP	81H	6.5.3	Q	Q	Q	Q	Q	Q	Q	Q	07H	
ESP	9BH	6.5.2	-	-	-	-	-	Q	Q	Q	xxxxx000b	
DPL	82H	6.4.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
DPH	83H	6.4.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
DPX	93H	6.4.4	-	-	-	-	-	-	-	Q	xxxxxxx0b	

T3-2.0 1320

1. All SFRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-3: Timer SFRs Reference

Register Name	Address	Reference Section	Bit Access Type								Reset Value[1]	Key
			7	6	5	4	3	2	1	0		
TCON	88H	10.3.1	Q	Q	Q	Q	R	R	R	R	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
TMOD	89H	10.3.2	R	Q	Q	Q	R	Q	Q	Q	00H	
TL0	8AH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
TL1	8BH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
TH0	8CH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
TH1	8DH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
T2CON	C8H	10.3.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
T2MOD	C9H	10.3.4	-	-	-	-	-	Q	Q	Q	x0H	
RCAP2L	CAH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
RCAP2H	CBH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
TL2	CCH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	
TH2	CDH	10-1	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-3.0 1320

1. All SFRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-4: UART SFRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
PCON	87H	11.4.1	Q	Q	Q	Q	Q	Q	Q	Q	00x[2]x[3]0000b	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
SADDR	A9H	11.4.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SADEN	B9H	11.4.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SBUF	99H	11.4.4	Q	Q	Q	Q	Q	Q	Q	Q	xxH	
SCON	98H	11.4.5	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-4.0 1320

1. All SFRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).
2. Bit 5 of PCON register (Brown-Out Flag) is cleared by Power-On reset, and it is not affected by other reset events.
3. Bit 4 of PCON register (Power-On Flag) is set by Power-On reset, and it is not affected by other reset events.

TABLE 3-5: SPI SFRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
SPCR	D5H	12.4.1	Q	-	Q	Q	Q	Q	Q	Q	0x000100b	Q: Read and Write -: Reserved Bit x: Indeterminate Value
SPSR	AAH	12.4.2	Q	Q	-	-	-	-	-	-	00xxxxxxb	
SPDR	86H	12.4.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-5.0 1320

1. All SFRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

3.5 Memory Mapped Configuration Registers (MMCR)

All Memory Mapped Configuration Registers are located in 8051 external data memory space, addresses 7F00H – 7FFFH. For the detailed description of MMCRs see the respective sections as specified Tables 3-6 through 3-25. Some of MMCRs contain reserved bits. On reads software must not rely on reserved bits being any particular value. On writes zeros should be written to reserved bits. When modifying a register with reserved bits the values read from the reserved bits can be written back to them. Software should not write to unused locations in MMCR space.

3.5.1 MMCR References

TABLE 3-6: GPIO Input MMCRs References

Register Name	Address	Reference Section	Bit Access Type								Reset Value[1]	Key
			7	6	5	4	3	2	1	0		
GPIOAIN	7F1AH	9.1.2	R	R	R	R	R	R	R	R	ppH	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit p: Pass through pin state
GPIOBIN	7F1DH	9.1.6	R	R	R	R	R	R	R	R	ppH	
GPIOCIN	7F20H	9.1.11	R	R	R	R	R	R	R	R	ppH	
GPIODIN	7F24H	9.1.15	R	R	R	R	R	R	R	R	ppH	
GPIOEIN	7FA2H	9.1.18	R	R	R	R	R	R	R	R	ppH	
GPIOFIN	7FA5H	9.1.22	R	R	R	R	R	R	R	R	ppH	
GPIOGIN	7F3BH	9.1.26	R	R	R	R	R	R	R	R	ppH	
GPIOHIN	7FE2H	9.1.30	R	R	R	R	R	R	R	R	ppH	
GPIOIIN	7FE5H	9.1.34	R	R	R	R	R	R	R	R	ppH	
GPIOJIN	7FE8H	9.1.39	R	R	R	R	R	R	R	R	ppH	
GPIOKIN	7FEBH	9.1.44	R	R	R	R	R	R	R	R	ppH	
GPIOLIN	7FA9H	9.1.48	R	R	R	R	R	R	R	R	ppH	
GPIOMIN	7F6BH	9.1.51	R	R	R	R	R	R	R	R	ppH	
GPIONIN	7F84H	9.1.55	R	R	R	R	R	R	R	R	ppH	

T3-6.0 1320

[1] All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-7: GPIO Output MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
GPIOAOUT	7F19H	9.1.3	Q	Q	Q	Q	Q	Q	Q	Q	FFH	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
GPIOBOUT	7F1CH	9.1.7	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOCOUT	7F1FH	9.1.12	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIODOUT	7F23H	9.1.16	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOEOUT	7FA1H	9.1.19	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOFOUT	7FA4H	9.1.23	Q	Q	-	-	-	Q	Q	Q	11xxx111b	
GPIOGOUT	7F3AH	9.1.27	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOHOUT	7FE1H	9.1.31	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOIOUT	7FE4H	9.1.35	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOJOUT	7FE7H	9.1.40	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOKOUT	7FEAH	9.1.45	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOLOUT	7FA8H	9.1.49	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIOMOUT	7F30H	9.1.52	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
GPIONOUT	7F82H	9.1.56	Q	Q	Q	Q	Q	Q	Q	Q	FFH	

T3-7.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-8: GPIO Direction MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
GPIOADIR	7F18H	9.1.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
GPIOBDIR	7F1BH	9.1.5	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOCDIR	7F1EH	9.1.10	Q	-	-	-	-	-	-	-	0xxxxxxxb	
GPIODDIR	7F22H	9.1.14	-	-	-	-	Q	Q	Q	Q	x0H	
GPIOFDIR	7FA3H	9.1.21	Q	Q	-	-	-	Q	Q	-	00xxx00xb	
GPIOGDIR	7F39H	9.1.25	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOHDIR	7FE0H	9.1.29	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOIDIR	7FE3H	9.1.33	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOJDIR	7FE6H	9.1.38	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOKDIR	7FE9H	9.1.43	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOLDIR	7FA7H	9.1.47	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-8.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-9: GPIO Function Selection MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
GPIOASEL	7F3DH	9.1.4	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
GPIOBSEL	7F40H	9.1.8	Q	Q	Q	Q	Q	Q	-	-	000000xxb	
GPIOCSEL	7FDCH	9.1.13	-	Q	Q	Q	Q	Q	Q	Q	x0000000b	
GPIODSEL	7FDDH	9.1.17	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESEL	7FDEH	9.1.20	Q	Q	Q	Q	Q	Q	Q	Q	C0H	
GPIOFSEL	7FA6H	9.1.24	Q	Q	Q	-	-	-	-	Q	000xxxx0b	
GPIOGSEL	7F3CH	9.1.28	-	Q	Q	Q	Q	Q	Q	Q	x0000000b	
GPIOHLOD	7FABH	9.1.32	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOISEL	7FF0H	9.1.36	Q	Q	Q	Q	-	-	-	-	0xH	
GPIOIOD	7FACH	9.1.37	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOJSEL	7FF5H	9.1.41	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOJOD	7FADH	9.1.42	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOKPU	7FB7H	9.1.46	-	-	Q	Q	Q	Q	Q	Q	xx000000b	
GPIOMPU	7F5BH	9.1.50	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOMOD	7FF6H	9.1.53	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIONPU	7F83H	9.1.54	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIONOD	7FFCH	9.1.57	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-9.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-10: GPIO Active Edge Selection MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value**[1]	**Key**
GPIOESA	7F57H	8.3.19	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
GPIOESB	7F58H	8.3.22	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESC	7F5CH	8.3.25	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESD	7F5DH	8.3.28	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESE	7F60H	8.3.31	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESF	7F61H	8.3.34	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESG	7F62H	8.3.37	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESH	7F6CH	8.3.40	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESI	7F6DH	8.3.43	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESJ	7F6EH	8.3.44	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESK	7F6FH	8.3.47	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESL	7FD0H	8.3.48	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESM	7FD1H	8.3.51	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESN	7FD2H	8.3.52	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOKINT	7FB8H	8.3.53	-	-	Q	Q	Q	Q	Q	Q	xx000000b	
GPIOESO	7FD3H	8.3.56	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESP	7FD4H	8.3.57	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESQ	7FD5H	8.3.60	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESR	7FD6H	8.3.61	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOESS	7FD7H	8.3.64	Q	Q	Q	Q	Q	Q	Q	Q	00H	
GPIOEST	7FD8H	8.3.65	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-10.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-11: Interrupt Source MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value**[1]	**Key**
INTSRCA	7F00H	8.3.8	R	R	R	R	-	R	Q	R	1000x000b	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
INTSRCAMSK	7F01H	8.3.9	Q	Q	Q	Q	-	Q	Q	Q	0000x000b	
INTSRCB	7F02H	8.3.10	R	R	R	R	R	R	R	R	90H	
INTSRCBMSK	7F03H	8.3.11	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-11.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-12: Wakeup Source MMCRs References

Register Name	Address	Reference Section	Bit Access Type								Reset Value[1]	Key
			7	6	5	4	3	2	1	0		
WSRCA	7F2AH	8.3.12	Q	Q	Q	Q	Q	-	R	Q	00000x10b	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
WSRCAMSK	7F2BH	8.3.13	Q	Q	Q	Q	Q	-	Q	Q	00000x00b	
WSRCB	7F2CH	8.3.14	R	R	-	-	Q	Q	Q	Q	00xx0000b	
WSRCBMSK	7F2DH	8.3.15	Q	Q	-	-	Q	Q	Q	Q	00xx0000b	
KEYWSRC	7F2FH	8.3.16	-	-	-	-	-	-	Q	Q	xxxxxx00b	
WSRCC	7F59H	8.3.17	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCCMSK	7F5AH	8.3.18	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCD	7F5EH	8.3.20	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCDMSK	7F5FH	8.3.21	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCE	7F63H	8.3.23	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCEMSK	7F66H	8.3.24	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCF	7F64H	8.3.26	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCFMSK	7F65H	8.3.27	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCG	7F55H	8.3.29	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCGMSK	7F56H	8.3.30	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCH	7FAEH	8.3.32	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCHMSK	7FAFH	8.3.33	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCI	7F3EH	8.3.35	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCIMSK	7F3FH	8.3.36	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCJ	7FC8H	8.3.38	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCJMSK	7FC9H	8.3.39	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCK	7FCAH	8.3.41	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCKMSK	7FCBH	8.3.42	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCL	7FCCH	8.3.45	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCLMSK	7FCDH	8.3.46	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCM	7FCEH	8.3.49	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCMMSK	7FCFH	8.3.50	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCN	7FB0H	8.3.54	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCNMSK	7FB1H	8.3.55	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCO	7FDBH	8.3.58	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCOMSK	7FECH	8.3.59	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCP	7FEDH	8.3.62	Q	Q	Q	Q	Q	Q	Q	Q	00H	
WSRCPMSK	7FEEH	8.3.63	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-12.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-13: Timer MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
WDTCSR	7F37H	10.4.1.1	Q	-	-	-	-	-	Q	Q	0xxxxx00b	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
WDTDAT	7F38H	10.4.1.2	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
HIBER	7FF3H	10.5.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-13.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-14: PWM MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
PWMPL0	7F25H	10.6.1.1	Q	Q	Q	Q	Q	Q	Q	Q	FFH	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value p: Pass through pin state
PWMPL1	7F26H	10.6.1.2	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMPL2	7F29H	10.6.1.3	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMPH0	7F97H	10.6.1.4	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMPH1	7F98H	10.6.1.5	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMPH2	7F99H	10.6.1.6	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMC0	7F9AH	10.6.1.7	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMC1	7F9BH	10.6.1.8	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMC2	7F9CH	10.6.1.9	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMD0	7F9DH	10.6.1.10	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMD1	7F9EH	10.6.1.11	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMD2	7F9FH	10.6.1.12	Q	Q	Q	Q	Q	Q	Q	Q	FFH	
PWMCR	7F96H	10.6.1.13	Q	Q	Q	Q	-	Q	Q	Q	0000x000b	
PWM555CR1	7F21H	10.6.1.14	-	Q	Q	R	Q	Q	Q	Q	x00p0000b	

T3-14.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-15: SMBus MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
SMCR0	7F31H	13.5.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value p: Pass through pin state
SMCR1	7F67H	13.5.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SMSR0	7F32H	13.5.3	Q	Q	Q	Q	R	R	R	R	00H	
SMSR1	7F68H	13.5.4	Q	Q	Q	Q	R	R	R	R	00H	
SAR0	7F33H	13.6.1	Q	Q	Q	Q	Q	Q	Q	-	0000000xb	
SAR1	7F69H	13.6.2	Q	Q	Q	Q	Q	Q	Q	-	0000000xb	
SDSR0	7F34H	13.6.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SDSR1	7F6AH	13.6.4	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SLSR	7F88H	13.7.2	-	-	R	R	R	R	R	R	xxppppppb	
SSCR	7F89H	13.7.1	Q	Q	-	-	-	Q	Q	Q	00xxx001b	

T3-15.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-16: PS/2 MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
PS2TX0	7F41H	14.4.1.1	W	W	W	W	W	W	W	W	-	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
PS2TX1	7F45H	14.4.1.2	W	W	W	W	W	W	W	W	-	
PS2TX2	7F49H	14.4.1.3	W	W	W	W	W	W	W	W	-	
PS2RCV0	7F41H	14.4.2.1	R	R	R	R	R	R	R	R	FFH	
PS2RCV1	7F45H	14.4.2.2	R	R	R	R	R	R	R	R	FFH	
PS2RCV2	7F49H	14.4.2.3	R	R	R	R	R	R	R	R	FFH	
PS2CR0	7F42H	14.4.3.1	Q	Q	Q	Q	Q	Q	Q	Q	40H	
PS2CR1	7F46H	14.4.3.2	Q	Q	Q	Q	Q	Q	Q	Q	40H	
PS2CR2	7F4AH	14.4.3.3	Q	Q	Q	Q	Q	Q	Q	Q	40H	
PS2STS0	7F43H	14.4.4.1	R	R	R	R	R	R	R	R	50H	
PS2STS1	7F47H	14.4.4.2	R	R	R	R	R	R	R	R	50H	
PS2STS2	7F4BH	14.4.4.3	R	R	R	R	R	R	R	R	50H	
APS2STS0	7FF7H	14.4.4.4	R	R	R	R	R	R	R	R	50H	
APS2STS1	7FF8H	14.4.4.5	R	R	R	R	R	R	R	R	50H	
APS2STS2	7FF9H	14.4.4.6	R	R	R	R	R	R	R	R	50H	
PS2TMOUT	7F44H	14.4.5.1	-	-	-	-	Q	Q	Q	Q	xxxx0000b	
PS2STATUS2	7F48H	14.4.5.2	R	R	R	R	R	R	R	-	0000000xb	

T3-16.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-17: Fan Tachometer MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
FANCNT1	7FBAH	15.3.1	R	R	R	R	R	R	R	R	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
FANCNT2	7FBBH	15.3.2	R	R	R	R	R	R	R	R	00H	
FAN1LD	7FBCH	15.3.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
FAN2LD	7FBDH	15.3.4	Q	Q	Q	Q	Q	Q	Q	Q	00H	
FANTIMEBASE	7FBEH	15.3.5	Q	Q	-	-	Q	Q	Q	Q	00xx0101b	

T3-17.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-18: ADC MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
ADDRA	7F8EH	16.2.1	R	R	R	R	R	R	R	R	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
ADDRB	7F8FH	16.2.2	R	R	R	R	R	R	R	R	00H	
ADDRC	7F90H	16.2.3	R	R	R	R	R	R	R	R	00H	
ADDRD	7F91H	16.2.4	R	R	R	R	R	R	R	R	00H	
ADDRL	7F92H	16.2.5	R	R	R	R	R	R	R	R	00H	
ADCSR	7F93H	16.2.6	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-18.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-19: DAC MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
DACDAT0	7F4CH	17.2.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
DACDAT1	7F4DH	17.2.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
DACDAT2	7F4EH	17.2.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
DACDAT3	7F4FH	17.2.4	Q	Q	Q	Q	Q	Q	Q	Q	00H	
DACCTRL	7F50H	17.2.5	-	-	-	Q	Q	Q	Q	Q	xxx00000b	

T3-19.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-20: KBC Host Interface MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
KBCDATA	7FF1H	18.2.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
KBCSTS	7FF2H	18.2.2	Q	Q	Q	Q	R	Q	R	R	00H	
AUXDATA	7FFAH	18.2.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
KBDCFG	7FF4H	18.2.4	Q	-	Q	Q	-	Q	Q	-	0x00x00xb	
PCOBF	7FFDH	18.2.5	-	-	-	-	-	-	-	Q	xxxxxxx0b	
KEYSCAN	7F04H	18.3.1	R	Q	Q	Q	Q	Q	Q	Q	20H	

T3-20.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-21: GA20 Control MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
GA20	7FFBH	19.2.1	-	-	-	-	Q	Q	-	Q	xxxx00x1b	Q: Read and Write -: Reserved Bit x: Indeterminate Value
SETGA20	7FFEH	19.2.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
RSTGA20	7FFFH	19.2.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-21.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-22: ACPI EC Interface MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
ECIDATA	7F53H	20.2.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
ECIDATA1	7F80H	20.2.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
ECISTS	7F54H	20.2.3	Q	Q	Q	Q	R	Q	R	R	00H	
ECISTS1	7F81H	20.2.4	Q	Q	Q	Q	R	Q	R	R	00H	
ECICFG	7F51H	20.3.1	-	-	Q	Q	Q	Q	Q	Q	xx000000b	
ECICFG1	7F52H	20.3.2	-	-	Q	Q	Q	Q	Q	Q	xx000000b	

T3-22.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-23: MailBox MMCRs References

			Bit Access Type									
Register Name	**Address**	**Reference Section**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Reset Value[1]**	**Key**
MBX0	7F08H	21.1.1	Q	Q	Q	Q	Q	Q	Q	Q	00H	Q: Read and Write R: Read Only W: Write Only -: Reserved Bit x: Indeterminate Value
MBX1	7F09H	21.1.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX2	7F0AH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX3	7F0BH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX4	7F0CH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX5	7F0DH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX6	7F0EH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX7	7F0FH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX8	7F10H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX9	7F11H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXA	7F12H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXB	7F13H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXC	7F14H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXD	7F15H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXE	7F16H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBXF	7F17H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX10	7F70H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX11	7F71H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX12	7F72H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX13	7F73H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX14	7F74H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX15	7F75H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX16	7F76H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX17	7F77H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX18	7F78H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX19	7F79H	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1A	7F7AH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1B	7F7BH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1C	7F7CH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1D	7F7DH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1E	7F7EH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX1F	7F7FH	21.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
MBX94	-	21.2.1	R	R	-	Q	R	-	-	Q	00x00xx0b	
MBX96	-	21.2.2	-	-	-	-	R	-	-	-	xxxx0xxxb	
MBX97	-	21.2.3	-	-	-	-	Q	-	-	-	xxxx0xxxb	

T3-23.0 1320

[1] All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-24: Configuration MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
LPCMON	7F8AH	23.1.1	Q	Q	Q	-	-	-	Q	R	001xxx0pb	Q: Read and Write R: Read Only -: Reserved Bit x: Indeterminate Value p: Pass through pin state
CFGINDEX	7F8CH	23.1.2	Q	Q	Q	Q	Q	Q	Q	Q	00H	
CFGDATA	7F8DH	23.1.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	

T3-24.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).

TABLE 3-25: Miscellaneous MMCRs References

			Bit Access Type									
Register Name	Address	Reference Section	7	6	5	4	3	2	1	0	Reset Value[1]	Key
MID	7F05H	3.5.2	R	R	R	R	R	R	R	R	BFH	Q: Read and Write R: Read Only -: Reserved Bit x: Indeterminate Value
DEVID	7F06H	3.5.3	R	R	R	R	R	R	R	R	F0H	
DEVREV	7F07H	3.5.4	R	R	R	R	R	R	R	R	01H	
CLKSRCCON[2]	7F27H	5.3.2.1	-	-	Q	Q	R	Q	Q	R	xx010100b	
SRCROM[2]	7F28H	4.4.1.1	-	-	-	-	Q	Q	Q	Q	x0H	
RSTCON[2]	7F2EH	5.2.3.1	-	-	-	-	-	Q	Q	Q	xxxxx001b	
SFCS	7FC0H	4.4.5.1	Q	-	-	-	Q	Q	R	Q	0xxx0x[3]10b	
SFCMD	7FC1H	4.4.5.2	Q	Q	-	-	Q	Q	Q	Q	00xx0000b	
SFAL	7FC2H	4.4.5.3	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SFAH	7FC3H	4.4.5.4	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SFAX	7FC6H	4.4.5.5	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SFDL	7FC4H	4.4.5.6	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SFDH	7FC5H	4.4.5.7	Q	Q	Q	Q	Q	Q	Q	Q	00H	
SFSEC	7FC7H	4.7.1	Q	Q	Q	Q	Q	Q	Q	Q	00[4]0000x[3]0b	
LPCSS	7FDFH	9.1.9	-	-	-	-	-	Q	Q	Q	xxxxx000b	
PLLM[2]	7F35H	5.3.2.2	-	-	Q	Q	Q	Q	Q	Q	xx100010b	
PLLPS[2]	7F36H	5.3.2.3	-	-	Q	Q	Q	Q	Q	Q	xx110010b	
ALPCBC	7F8BH	4.8.2	Q	-	-	-	-	-	-	-	0xxxxxxxb	

T3-25.0 1320

1. All MMCRs returned to their reset values specified in the charts above after the following reset events: Power-On reset, External reset, Watchdog timer reset, Brown-out reset, 8051 firmware Soft reset, LPC Soft reset, and aLPC Soft reset (see also Section 5.2).
2. These selected MMCRs are also returned to their reset values after the following reset events: 8051 firmware Soft reset, and LPC Soft reset (see also Section 5.2).
3. Bit 2 of SFCS register and Bit 1 of the SFSEC register are affected by reset events as follows:
   After Power-On Reset, External reset, Watchdog timer reset, Brown-out reset, and aLPC Soft reset
   SFCS[2] = SFSEC[1] = 0 if ENVR location at address 0FFFH contains 0FFH value
   SFCS[2] = SFSEC[1] = 1 if ENVR location at address 0FFFH contains any value other than 0FFH
   After 8051 firmware Soft reset, and LPC Soft reset SFCS[2] is always cleared (‘0’), and SFSEC[1] is preserved.
   These bits are not affected by other reset events.
4. Bit 6 of SFSEC register is reset only by Power-On reset, External reset, Brown-out reset, and LPC Interface reset.

3.5.2 JEDEC Registers

3.5.2.1 JEDEC Manufacturer ID Register (MID)

Location		7	6	5	4	3	2	1	0
7F05H	Read	MID7	MID6	MID5	MID4	MID3	MID2	MID1	MID0
	Write	-	-	-	-	-	-	-	-
	Reset	1	0	1	1	1	1	1	1

Symbol	Function
-	Not implemented, reserved for future use. **Note**: User should not write ‘1’s to reserved bits. The value read from a reserved bit is indeterminate.

3.5.2.2 JEDEC Device ID Register (DEVID)

Location		7	6	5	4	3	2	1	0
7F06H	Read	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	Write	-	-	-	-	-	-	-	-
	Reset	1	1	1	1	0	0	0	0

Symbol	Function
-	Not implemented, reserved for future use. **Note**: User should not write ‘1’s to reserved bits. The value read from a reserved bit is indeterminate.

3.5.2.3 Device Revision Register (DEVREV)

Location		7	6	5	4	3	2	1	0
7F07H	Read	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
	Write	-	-	-	-	-	-	-	-
	Reset	Device Revision Number[1]							

[1] Please contact SST to find out how this number corresponds to the SST79LF008 package markings.

Symbol	Function
-	Not implemented, reserved for future use. **Note**: User should not write ‘1’s to reserved bits. The value read from a reserved bit is indeterminate.

4.0 FLASH MEMORY PROGRAMMING

4.1 SuperFlash Memory Overview

The SST79LF008 flash memory is manufactured with SST's proprietary, high-performance SuperFlash technology. The split-gate cell design and thick-oxide tunneling injector attain greater reliability and manufacturability compared with alternative technology approaches. The SuperFlash technology significantly improves the performance and reliability of the flash memory while lowering power consumption.

The SST79LF008 allows flash Write operations (Program or Erase) in-system with a single 3.0-3.6V power supply, and it uses less energy during Erase and Program than alternative flash memory technologies for memory devices. The total energy consumed is a function of the applied voltage, current, and time of application. SuperFlash technology uses less current to program for any voltage range and has a shorter erase time than comparable technologies. This means that the total energy consumed during any Erase or Program operation is less than alternative flash memory technologies.

The SuperFlash technology provides fixed Erase and Program times, independent of the number of the performed Erase/Program cycles. This eliminates the need to calibrate or correlate the system software or hardware to the cumulative number of erase cycles, which is necessary with alternative flash memory technologies whose Erase and Program times increase with accumulated Erase/Program cycles. To protect against inadvertent write, the SST79LF008 provides on-chip write protection. The SST79LF008 flash memory is offered with a typical endurance of 100,000 cycles and data retention of greater than 100 years.

The SST79LF008 flash array is a (512K + 2K) x 16 sector erase, block erase, and word program embedded SuperFlash memory. It is organized as 512K words of Main Flash Memory array plus 2K words of erasable non-volatile registers (ENVR). In addition 1.5K words of user non-volatile registers (UNVR) can be one-time programmed.

Key Features of SST79LF008 Flash Memory:

- SuperFlash Technology
- Organized as (512K Main array + 2K ENVR) x16
- Single Voltage Read/Write Operations
- Endurance: 100,000 Cycles (typical)
- Greater than 100 years Data Retention
- Main array and ENVR write and read protection with a lock down/open after reset only option
- Uniform 2K Word sectors
- Uniform 32K Word blocks
- Sector-/Block-Erase Capability
- Fast Sector-/Block-Erase Time:
  – 55 ms typical
  – 60 ms max
- Fast Word-Program Time:
  – 15 µs typical
  – 60 µs max

4.2 Flash Memory Map

The map of SST79LF008 main flash memory array is shown in Figure 4-1. It includes a total 1 MByte (8 Mbit) flash memory space. A maximum of 128 KByte KBC firmware block can be located in the lower part of flash memory to store the 8051-specific keyboard and embedded controller code. The other flash memory area can be used to store the system BIOS related code and data. See Section 7.0 for the details on mapping SST79LF008 flash memory into the system LPC interface address space.

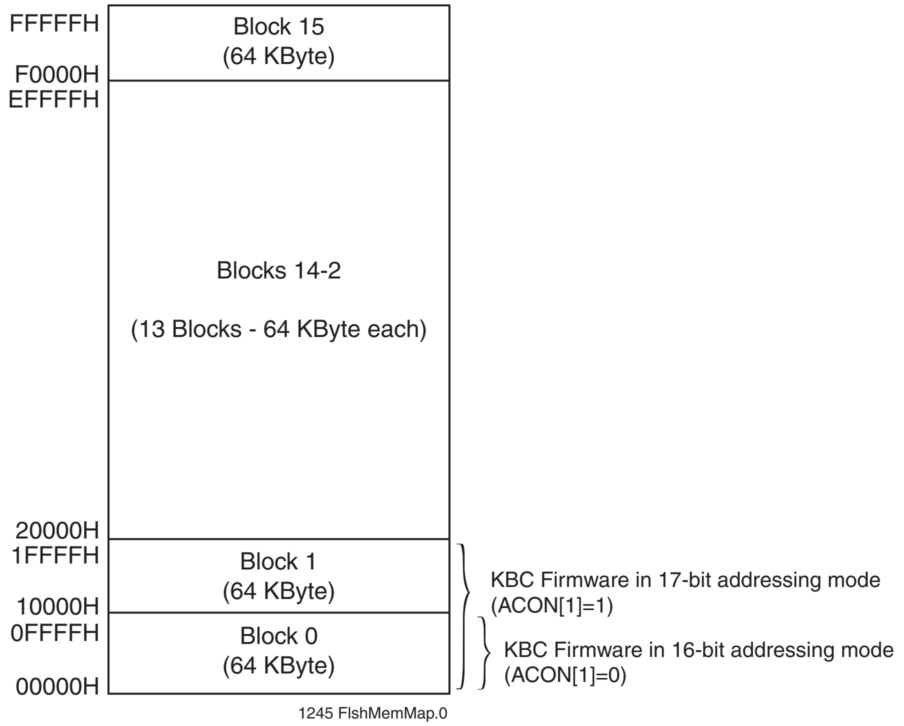


FIGURE 4-1: SST79LF008 Flash Memory Map

4.2.1 ENVR / UNVR Address Space

The maps of ENVR and UNVR address spaces are shown in tables below.

TABLE 4-1: ENVR Address Space

Address	Contents
FFFH	EnableBoot Byte (see Section 4.5)
000H-FFEH	Erasable NVR for user

T4-1.0 1320

TABLE 4-2: UNVR Address Space

Address	Contents
000H-BFFH	3K OTP for User NVR

T4-2.0 1320

4.2.2 Programming Modes

The SST79LF008 internal flash memory (including ENVR and UNVR spaces) can be programmed through a Shared ROM Interface using the following three methods:

- In-Application Programming Mode (8051 controlled)
- Remote aLPC Programming Mode (aLPC Host controlled)
- In-system LPC Programming Mode (LPC Host controlled)

The first two modes are described in this section. For LPC programming mode, see Section 7.0.

4.3 Shared ROM Interface

The SST79LF008 Shared ROM Interface (SRI) provides LPC Host, alternate LPC (aLPC) Host and the 8051 MCU with access to the entire 1 MByte of main Flash memory array as well as to ENVR and UNVR areas. See Figure 4-2.

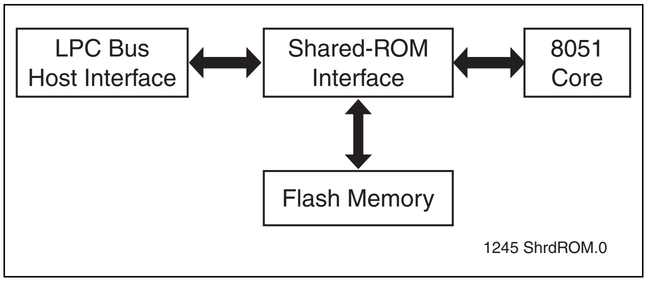


FIGURE 4-2: Shared ROM Interface

Both LPC and aLPC Hosts access flash memory via the internal LPC bus interface unit (see details on switching bus control between LPC Host and aLPC Host in Section 4.8). The aLPC Host flash access is exclusive as 8051 is forced into reset state while aLPC mode is enabled. The LPC Host and 8051 may access flash memory concurrently. For LPC Host read operations the arbitration between LPC Host and 8051 is completely handled by SRI hardware. For LPC Host program/erase operation flash memory access arbitration must be implemented in software via one of the following mechanisms:

1. 8051 firmware turns over the flash bus to the LPC Host by setting HOST_ACCESS bit in SFCS register, see Section 4.4.5, In this case 8051 firmware must run from the Scratch ROM, described in Section 4.4.1, just before and while the flash bus is released to the LPC Host.
2. 8051 firmware enters Idle mode, which allows the LPC Host software to take over the flash bus by setting the STP_CLK bit in MBX94 register, described in Section 21.2. In this case 8051 firmware can run from flash before entering Idle mode provided the LPC Host does not update the respective flash locations.

   For any mechanism above, the LPC Host software can confirm the bus turn around by checking the HOSTFLASH bit in MBX94 register. See Section 21.2.
3. 8051 firmware enables LPC Soft reset via LPCMON register, detailed in Section 23.1, which allows the LPC Host to take over the flash bus by sending a Force LPC Soft Reset command, detailed in Section 7.3. In this case 8051 will be kept in reset state and 8051 firmware will not run until the LPC Host sends a Release LPC Soft Reset command, which re-starts 8051 code execution. As KBC operation is aborted and then restarted, this mechanism is not recommended for LPC Host program/erase access, unless flash memory is blank or corrupted.

4.4 In-Application Programming Mode

During In-Application programming (IAP), the 8051 executes instructions from the scratch ROM, which is a portion of XRAM mapped to the top or bottom of the first 64 KByte (Block0) in the 8051 program space. The flash control registers SFCS, SFCMD, SFAL, SFAH, SFDL, and SFDH located among the memory-mapped registers, control and monitor the device's erase and program operations.

4.4.1 Scratch ROM Mapping Control

The IAP code must be executed from 8051 program memory space that is not on the flash. For this purpose the SST79LF008 allows a section of on-chip XRAM to be mapped to 8051 program space. The memory-mapped register SCRROM, described below, controls the mapping of XRAM to 8051 program address space. Mapping can affect the entire 2K expanded RAM, or only part of it, as indicated by SCRSIZE field. Table 4.3 details scratch ROM mapping to 8051 program space. See also Figures 3-2 to 3-6.

4.4.1.1 Scratch ROM Control Register (SCRROM)

Location		7	6	5	4	3	2	1	0
7F28H	Read/Write	-	-	-	-	SCRSIZE1	SCRSIZE0	SCRPOS	SCREN
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented, reserved for future use. **Note**: User should not write '1's to reserved bits. The value read from a reserved bit is indeterminate.
X	Not defined
SCRSIZE[1:0]	Scratch ROM size 00: 256 Bytes 01: 512 Bytes 10: 1024 Bytes 11: 2048 Bytes
SCRPOS	Scratch ROM position 1: Map to Block0 top 0: Map to Block0 bottom
SCREN	Scratch ROM mapping enable 1: Enable mapping 0: Disable mapping

TABLE 4-3: Scratch ROM Mapping

SCREN	SCRPOS	SCRSIZE[1:0]	XRAM Section Address	Program Space Address
0	X	XX	000H - 7FFH	No Mapping
1	0	00	700H - 7FFH	0000H - 00FFH
1	0	01	600H - 7FFH	0000H - 01FFH
1	0	10	400H - 7FFH	0000H - 03FFH
1	0	11	000H - 7FFH	0000H - 07FFH
1	1	00	700H - 07FFH	FF00H - FFFFH
1	1	01	600H - 7FFH	FE00H - FFFFH
1	1	10	400H - 7FFH	FC00H - FFFFH
1	1	11	000H - 7FFH	F800H - FFFFH

T4-3.0 1320

4.4.2 IAP Mode Control

The IAP enable bit, SFCS[7], enables in-application programming mode. Until this bit is set, all flash programming IAP commands will be ignored.

Table 4-4 contains the In-Application Programming commands that can be used when IAP mode is enabled. All IAP commands should be executed from the Scratch ROM by writing the command code to SFCMD register after address and data registers are properly loaded, see Figures 4-3 to 4-6. The command codes not listed in the table are reserved.

The same commands are used to access either main 1MByte flash array, or 4 KByte flash ENVR, or 3 KByte OTP UNVR. The selection of the target memory area is controlled by bits SFCMD[7:6].

TABLE 4-4: IAP Commands[1] for SST79LF008

Operation	SFCMD[3:0]	SFDH[7:0]	SFDL[7:0]	SFAX[7:0]	SFAH[7:0]	SFAL[7:0]
Reserved[2]	0000b, 1110b	X[3]	X	X	X	X
Sector-Erase	0001b	X	X	AX[4]	AH[5]	X
Block-Erase	0010b	X	X	AX[4]	AH[5]	X
Word- Program	0011b	DH[6]	DL[6]	AX[4]	AH[5]	AL[7]
Erase-Suspend	0100b	X	X	X	X	X
Erase-Resume	0101b	X	X	X	X	X
Word-Read	1111b	DH[6]	DL[6]	AX[4]	AH[5]	AL[7]
No Operation	Remaining Combinations	X[3]	X	X	X	X

T4-4.0 1320

1. SFCS[7] = 1 enables IAP commands; SFCS[7] = 0 disables IAP commands.
2. Do not use reserved values.
3. X = “Don’t care”
4. AX = Word Address most significant order byte (SFAX[7:3] = 0, SFAX[2:0] = Word address bits 18:16).
5. AH = Word Address high order byte (SFAH[7:0] = Word address bits 15:8).
6. DH = Data high byte (input or output); DL = Data low byte (input or output).
7. AL = Word Address low order byte (SFAL[7:0] = Word address bits 7:0).

4.4.3 Address Selection for IAP Commands

Only word access is supported in IAP mode. Three address registers (SFAX, SFAH and SFAL) are provided to specify 24-bit word address for any individual word location. Only 19 bits of these registers are significant for access to 512 KWord main flash array, bits 23:19 (SFAX[7:3]) must be always 0. Sixteen 32 KWord blocks in the main flash array can be erased independently. There are 16 sectors of 2 KWord in each block, and each sector can also be erased independently. In addition the 2 KWord ENVR flash sector can be erased in IAP mode. For block or sector selection SFAX, SFAH, and SFAL registers should be loaded with any valid word address within the respective block or sector.

4.4.4 IAP Mode Commands Description

4.4.4.1 No Operation

The No Operation command causes the flash controller to do nothing.

4.4.4.2 Sector-Erase

The Sector-Erase command erases all bytes in a sector. The sector size is 2 KWord (4 KByte). The selection of the sector to be erased is determined by the contents of SFAX and SFAH registers. For example, to erase sector FF000H-FFFFFH of the main flash array the following settings should be used.

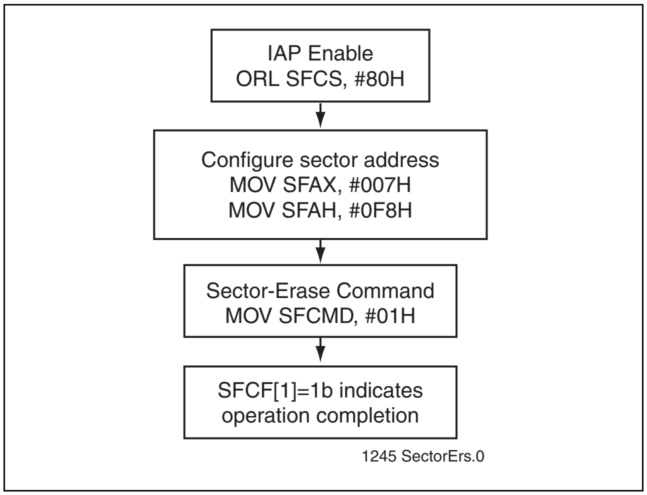


FIGURE 4-3: IAP Sector-Erase

4.4.4.3 Block-Erase

The Block-Erase command erases all bytes in a 32 KWord (64 KByte) memory block. The selection of the block to be erased is determined by the contents of SFAX and SFAH registers. For example, to erase block F0000H-FFFFFH of the main flash array the following settings should be used.

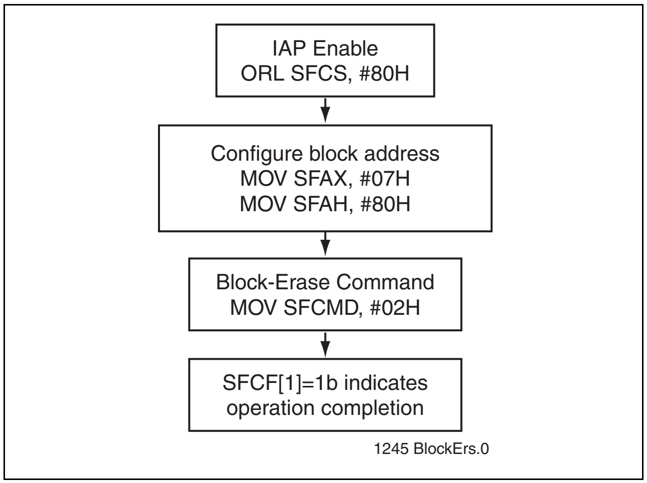


FIGURE 4-4: IAP Block-Erase

4.4.4.4 Word-Program

The Word-Program command programs data into a single word location in the flash memory. The word address is determined by the contents of SFAX, SFAH and SFAL registers. The data to be programmed is stored in registers SFDH (high byte) and SFDL (low byte). For example, to program word data to the main flash array at word address 7F801H (which results in storing high byte word data at byte address FF003H, and low byte word data at byte address FF002H) the following settings should be used.

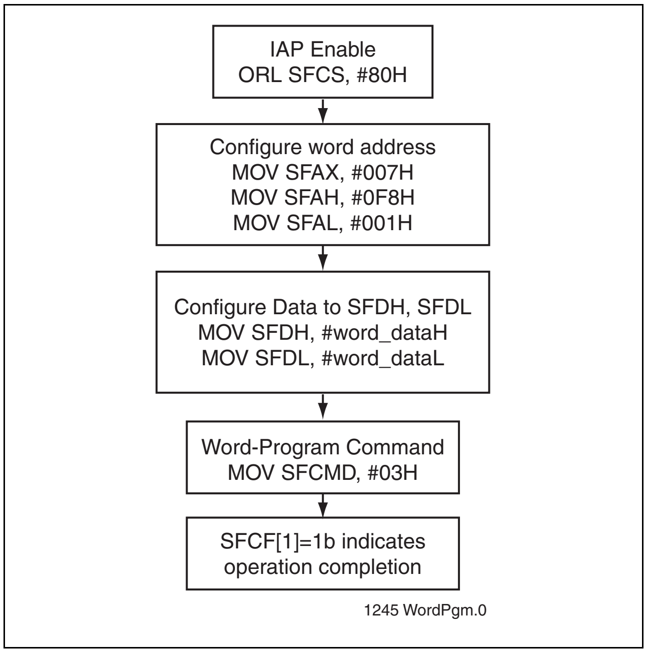


FIGURE 4-5: IAP Word-Program

4.4.4.5 Erase-Suspend

The Erase-Suspend command allows a Sector-Erase or Block-Erase operation interruption in order to read or program data into another block of memory. Once the Erase-Suspend command is executed, the device will suspend the on-going erase operation.

After a successful Erase-Suspend, a Word-Read or Word-Program command can be issued to read from or write to a different sector/block of flash memory than the one suspended. If a Word-Read command is issued to an address within the suspended sector or block, the read operation may return invalid data. If a Word-Program command is issued to an address within the suspended memory area, the command is acknowledged but rejected.

Suspended operations cannot be nested. That is, the system needs to complete/resume any previously suspended operation before a new operation can be suspended.

4.4.4.6 Erase-Resume

The Erase-Resume command resumes the erase process in the suspended sector or block. After the Erase-Resume command is issued, the device will resume the erase process. Erase cannot be resumed until the Word-Read or Word-Program operation already in progress has been completed.

4.4.4.7 Word-Read

The Word-Read command allows the user to verify that the device has correctly performed an Erase or Program command. Word-Read returns the data word in registers SFDH (high byte) and SFDL (low byte) if the command is successful. The user must check if the previous flash operation has been fully completed before issuing a Word-Read. Word-Read command execution time is short enough that there is no need to poll for command completion. The address is determined by the contents of SFAX, SFAH and SFAL registers. For example, to read word FF003H:FF002H of the main flash array the following settings should be used.

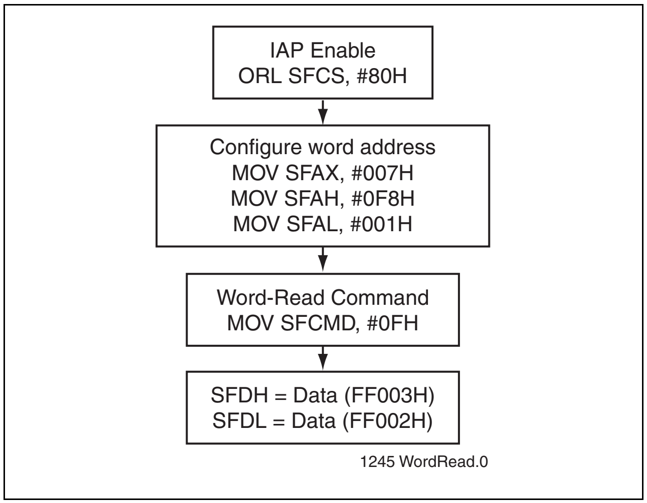


FIGURE 4-6: IAP Word-Read

4.4.5 SuperFlash Control and Status Registers

4.4.5.1 SuperFlash Control and Status Register (SFCS)

Location		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7FC0H	**Read**	IAPEN	-	-	-	SOFTRST	OVERLAY	FLASH_RDY	HOST_ACCESS
	Write							-	
	Reset	0	X	X	X	0	X	1	0

Symbol **Function**

-
Not implemented, reserved for future use.
Note: User should not write ‘1’s to reserved bits. The value read from a reserved bit is indeterminate.

X
Not defined

IAPEN
8051 controlled IAP mode Enable bit
1: Enable IAP mode
0: Disable IAP mode
This bit can be set by the 8051 firmware only while it is running from the Scratch ROM. The firmware must not restart running from flash until this bit is cleared.

SOFTRST
8051 controlled Soft Reset
A transition of this bit from 0 to 1 will generate an 8051 soft reset (see Section 5.2 for the effect of this reset event)

OVERLAY
BootRom Overlay bit
1: BootRom (0F000H? 0FFFFH) is mapped to bottom 4 KByte sector in 8051 program space (00000H - 00FFFH),
0: BootRom (0F000H - 0FFFFH) is NOT mapped to bottom 4 KByte sector in 8051 program space

The reset value of this bit is determined as follows.

1. After Power On Reset, External pin reset, Watchdog timer reset, Brown-out reset
   OVERLAY = 0 if the ENVR location at address 0FFFH contains 0FFH value.
   OVERLAY = 1 if the ENVR location at address 0FFFH contains any value other
   than 0FFH
2. After 8051 soft reset, LPC soft reset, and aLPC soft reset, OVERLAY = 0 always.
3. Once the byte at location 0FFFH of ENVR is programmed, it can only be restored by ENVR erase, but software can always change the value of OVERLAY bit at run time without the need of ENVR erase. OVERLAY bit affects only flash memory access addressed via 8051 program counter (PC). It does not affect physical addresses used in IAP mode for flash memory programming commands.

FLASH_RDY
Indicates program or erase completion in IAP mode.
1: Ready - IAP command is completed
0: Busy - IAP command is in progress

HOST_ACCESS
Flash bus ownership control bit.
1: 8051 released flash bus to LPC Host
0: 8051 owns the flash bus

4.4.5.2 SuperFlash Command Register (SFCMD)

Location		7	6	5	4	3	2	1	0
7FC1H	Read/ Write	UNVRSEL	ENVRSEL	-	-	FCM3	FCM2	FCM1	FCM0
	Reset	0	0	X	X	0	0	0	0

Symbol	Function
-	Not implemented, reserved for future use. **Note**: User should not write ‘1’s to reserved bits. The value read from a reserved bit is indeterminate.
X	Not defined
UNVRSEL	Select UNVR as target for IAP command.
ENVRSEL	Select ENVR as target for IAP command.

SFCMD[7:6]	Operation
1X	Select UNVR[1]
01	Select ENVR[2]
00	Select Main flash array

[1] Valid address for 3 KB UNVR area is from 0000H to 0BFFH

[2] Valid address for 4 KB ENVR area is from 0000H to 0FFFH

FCM[3:0] IAP mode flash command

SFCMD[3:0]	Operation
0000	No Operation
0001	Sector-Erase
0010	Block-Erase
0011	Word-Program
0100	Erase-Suspend
0101	Erase-Resume
1111	Word-Read

Note: All commands should be issued from the Scratch ROM while 8051 owns the flash bus.
The SFCMD[3:0] bits are automatically cleared by hardware during command execution, and read back operation returns 0 in FCM field. If command is issued while LPC Host owns the flash bus, it will be ignored and read back operation returns previously written command in FCM field.

4.4.5.3 SuperFlash Address Register (SFAL)

Location		7	6	5	4	3	2	1	0
7FC2H	Read/ Write	SFAL7	SFAL6	SFAL5	SFAL4	SFAL3	SFAL2	SFAL1	SFAL0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SFAL[7:0]	Flash Word Address low order byte.

4.4.5.4 SuperFlash Address Register (SFAH)

Location		7	6	5	4	3	2	1	0
7FC3H	Read/ Write	SFAH7	SFAH6	SFAH5	SFAH4	SFAH3	SFAH2	SFAH1	SFAH0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SFAH[7:0]	Flash Word Address high order byte.

4.4.5.5 SuperFlash Address Register (SFAX)

Location		7	6	5	4	3	2	1	0
7FC6H	Read/ Write	SFAX7	SFAX6	SFAX5	SFAX4	SFAX3	SFAX2	SFAX1	SFAX0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SFAX[7:0]	Flash Word Address most significant order byte.

Note:SFAX, SFAH and SFAL registers are cascaded to form a linear Word Address of the entire 8 Mbit on-chip flash memory (Word Address = (Byte Address)/2).

4.4.5.6 SuperFlash Data Low Register (SFDL)

Location		7	6	5	4	3	2	1	0
7FC4H	Read/ Write	SFDL7	SFDL6	SFDL5	SFDL4	SFDL3	SFDL2	SFDL1	SFDL0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SFDL[7:0]	Flash data low byte. For read flash command reading this register returns the low byte of word data from the flash. For write flash command SFDL contains the low byte of word data to be programmed. (SST79LF008 only supports word-wide access in IAP mode).

4.4.5.7 SuperFlash Data High Register (SFDH)

Location		7	6	5	4	3	2	1	0
7FC5H	Read/ Write	SFDH7	SFDH6	SFDH5	SFDH4	SFDH3	SFDH2	SFDH1	SFDH0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SFDH[7:0]	Flash data high byte. For read flash command reading this register returns the high byte of word data from the flash. For write flash command SFDH contain the high byte of word data to be programmed. (SST79LF008 only supports word-wide access in IAP mode).

4.5 BootRom Area

BootRom area physically occupies top 4KByte of flash memory Block0 at addresses from 0F000H to 0FFFFH. Bit 2 OVERLAY of the SFCS register controls logical mapping of the BootRom area.

When OVERLAY = 1, BootRom is enabled to overlay the bottom 4 KByte address of flash memory. The overlay mode only affects the memory access addressed via 8051 program counter (PC). It does not affect the physical addresses of flash memory locations used for memory access by flash programming commands. When OVERLAY = 0, BootRom overlay mode is disabled. The OVERLAY bit can be altered at run-time and any change takes effect immediately. Hence, it is recommended that the OVERLAY bit be changed by the code outside the overlapped bottom 4 KByte address range.

The Power-on reset, Brown-out reset, External pin reset, and Watchdog timer reset will set/clear OVERLAY bit according to the EnableBoot byte in ENVR as shown in Table 4-5.

8051 soft reset (via SFCS.3 bit), LPC and aLPC soft resets clear OVERLAY bit regardless of ENVR state and re-start KBC firmware always from 0000H physical address.

EnableBoot is a non-volatile flash byte, and it is located at address 0FFFH (last byte) of ENVR. After ENVR is erased, the default value is 0FFH. This non-volatile byte can be programmed by IAP command, or via LPC, or aLPC bus. Since SST79LF008 only supports word program, in order to program EnableBoot byte, the entire word 0FFFH:0FFEH has to be updated.

TABLE 4-5: OVERLAY Bit Value After RESET

EnableBoot Byte (ENVR Address 0FFFH)	OVERLAY (SFCS[2])	Description
0FFH (Default Value) (BootRom Disabled)	0	No overlay. After reset 8051 starts execution at physical address 0000H.
Non-0FFH (BootRom Enabled)	1	Overlay. 0F000H to 0FFFFH (top 4KB of 64KB) overlays to 0000H to 0FFFH (bottom 4KB of 64KB). After reset 8051 starts execution from BootRom at physical address 0F000H.

T4-5.0 1320

4.6 LPC Flash Programming Mode

The SST79LF008 flash memory array can be programmed independently by the LPC Host through the LPC bus interface. See Section 7.0 for the detailed description of all LPC Flash commands.

4.7 8051 Controlled Security

SST79LF008 provides an 8051 controlled read/write lock protection through a SuperFlash Security Control Register (SFSEC) in Section 4.71.

4.7.1 SuperFlash Security Control Register (SFSEC)

Location		7	6	5	4	3	2	1	0
7FC7H	Read/ Write	BTLK	STICKY_ LK	W_LOCK	R_LOCK	W_LOCK_ KBC	NO_MAP_ KBC	W_LOCK_ BTROM	NO_MAP_ BTROM
	Reset	0	0	0	0	0	0	X	0

Symbol	Function
X	Not defined
BTLK	8051 Boot sector lock bit (valid for 8051 IAP and LPC Flash programming modes) 1: Lock - 4 KByte sector at physical address 0000H - 0FFFH is write-protected 0: No Lock
STICKY_LK	ENVR sector lock bit (valid for 8051 IAP and LPC Flash programming modes). This bit can be set by 8051 firmware but it is cleared only by the following hardware reset events: Power On reset, Brown out reset, External pin reset, and LPC Reset. 1: Lock - ENVR is read-protected and write-protected (read returns 00H) 0: No Lock
W_LOCK	BIOS flash write lock bit (valid for LPC Flash programming mode only) 1: Lock - BIOS flash memory area is write-protected (BIOS flash area is Block2-Block15 = 7.0 Mbit, or Block1-Block15 = 7.5 Mbit depending on the status of ACON[1] bit. See Section 6.5) 0: No Lock
R_LOCK	BIOS flash read lock bit (valid for LPC Flash programming mode only) 1: Lock - BIOS flash memory area is read-protected (BIOS flash area is Block2-Block15 = 7.0 Mbit, or Block1-Block15 = 7.5 Mbit depending on the status of ACON[1] bit. See Section 6.5; LPC read access to BIOS area returns 00H). 0: No Lock
W_LOCK_KBC	KBC flash write lock bit (valid for 8051 IAP and LPC Flash programming modes). 1: Lock - KBC flash memory area is write-protected (KBC flash area is Block0-Block1 = 128 KByte, or Block0 = 64 KByte depending on the status of ACON[1] bit. See Section 6.5) 0: No Lock
NO_MAP_KBC	KBC flash mapping control bit 1: KBC area is hidden from the LPC host (KBC flash area is Block0-Block1 = 128 KByte, or Block0 = 64 KByte depending on the status of ACON[1] bit. See Section 6.5; LPC read access to KBC area returns 00H). 0: KBC area is visible to LPC host.
W_LOCK_BTROM	Write lock for BootRom area (valid for 8051 IAP and LPC programming modes). 1: Lock - BootRom area at physical address 0F000H - 0FFFFH is write-protected. 0: No Lock The reset value of this bit is determined as follows. After Power On Reset, External pin reset, Watchdog timer reset, Brown out reset W_LOCK_BTROM = 0 if the ENVR location at address 0FFFH contains 0FFH W_LOCK_BTROM = 1 if the ENVR location at address 0FFFH contains any value other than 0FFH
NO_MAP_BTROM	BootRom mapping control bit. 1: BootRom area at physical address 0F000H - 0FFFFH is hidden from the LPC host (LPC read access to BootRom area returns 00H). 0: BootRom area is visible to LPC host.

There are two sets of lock and mapping control bits that provide read, and write (program/erase) protection on a per-block basis. The first set is controlled by 8051 core, which can protect the BIOS memory space through the W_LOCK and R_LOCK bits, and KBC firmware space through W_LOCK_KBC, NO_MAP_KBC, BTLK, W_LOCK_BTROM, and NO_MAP_BTROM bits. The second set is controlled by the LPC host, which can set read or write protections of the flash blocks through block locking registers T_MINUS18_LK, …, T_BLOCK_LK described in Section 7.0. The LPC host also controls mapping of KBC firmware area via Bit 4 of Mailbox register 94 described in Section 21.0.

Only the protection status set by the 8051 core applies to 8051 firmware initiated Read or Write operations. For LPC host initiated Read or Write operations, both sets of the protection and mapping attributes take effect and the more restrictive one applies.

All of the above locks do not apply to aLPC programming mode.

4.8 aLPC MODE

4.8.1 Alternate LPC (aLPC) Interface

Besides the standard LPC bus, the SST79LF008 implements an alternate LPC (aLPC) bus to support remote in-system-programming of the on-chip Flash.

Either the LPC bus or the aLPC bus may be connected to the internal LPC bus interface unit (BIU) as shown on Figure 4-7.

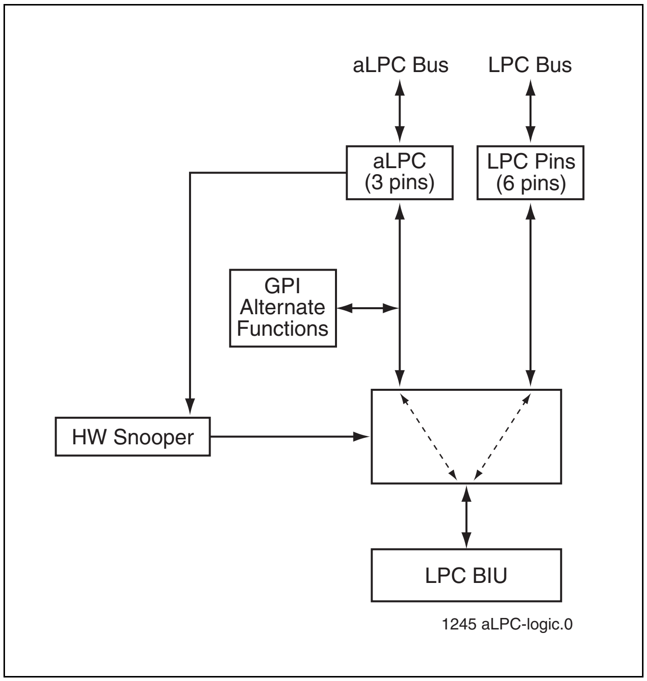


FIGURE 4-7: aLPC Logic Diagram

A 3-wire input hardware snooper circuit is used to switch the 3 aLPC signals (aLFRAME#, aLCLK and aLAD) from their default GPI mode to aLPC bus mode, upon detecting a unique non-random pattern stream of the "Enable_and_Poll" sequence described in Table 4-6.

TABLE 4-6: aLPC Snooper Command Sequences

	1st Bus Write Cycle[1]		2nd Bus Write Cycle[1]		3rd Bus Write Cycle[1]		4th Bus Write Cycle[1]	
Command Sequence	Addr	Data	Addr	Data	Addr	Data	Addr	Data
aLPC Mode Enable_and_Poll	55H	AAH	AAH	55H	55H	B7H	AAH	AFH
aLPC Mode Switch_and_Reset	55H	AAH	AAH	55H	55H	B7H	AAH	52H
aLPC Mode Exit	55H	C5H	AAH	B6H				

T4-6.0 1320

[1] Each Write Cycle is an aLPC I/O Write Cycle (See Section 4.8.6)

There are 3 states of the aLPC Snooper: IDLE, READY and SWITCHED to aLPC mode. After Power-on, Brown-out, External pin, and WDT resets, the SST79LF008 Snooper will start from IDLE state.

In IDLE state LPC bus is connected to the internal LPC interface unit, and hardware only snoops for Enable_and_Poll aLPC sequence, no other sequence will get the Snooper out of IDLE state. After successfully receiving a full Enable_and_Poll command, an interrupt is sent to the 8051 and the hardware Snooper will go to READY state.

In READY state, the hardware snoops for all command sequences:

- If the hardware snoops an aLPC Exit_and_Reset sequence, it will go to IDLE state.

- If the hardware snoops an Enable_and_Poll sequence, it will stay in READY state and return the status of RDY4ALPC bit on aLFRAME# line during SYNC phase of the fourth write cycle of the Enable_and_Poll sequence. (If the RDY4ALPC bit in the ALPCBC register is set, the Snooper will drive aLFRAME# low during SYNC phase, otherwise the Snooper will not drive aLFRAME# at all.) No interrupt to the 8051 is generated by the Snooper in the READY state.
- If hardware snoops a Switch_and_Reset sequence, it will proceed to SWITCHED state, which enable aLPC flash programming mode.

The aLPC Host can issue Switch_and_Reset sequence immediately after Enable_and_Poll to force the aLPC mode entry, or it can poll RDY4ALPC bit to make sure that 8051 is ready to switch. The former scenario is recommended when programming a blank chip, or if 8051 firmware is corrupted; the latter scenario provides handshaking with 8051 firmware for graceful entry to the aLPC mode.

In SWITCHED state (aLPC mode) 8051 is permanently kept in rest condition and aLPC bus is connected to internal LPC interface unit. Hardware only snoops for aLPC Exit_and_Reset sequence, which returns Snooper back to IDLE state. In SWITCHED state aLPC flash commands described in Section 4.8.6 are used to access the entire SST79LF008 flash memory.

All aLPC Snooper command sequences utilize aLPC I/O Write cycles described in Section 4.8.5. The following figure illustrates the relationship between three states of the aLPC snooper.

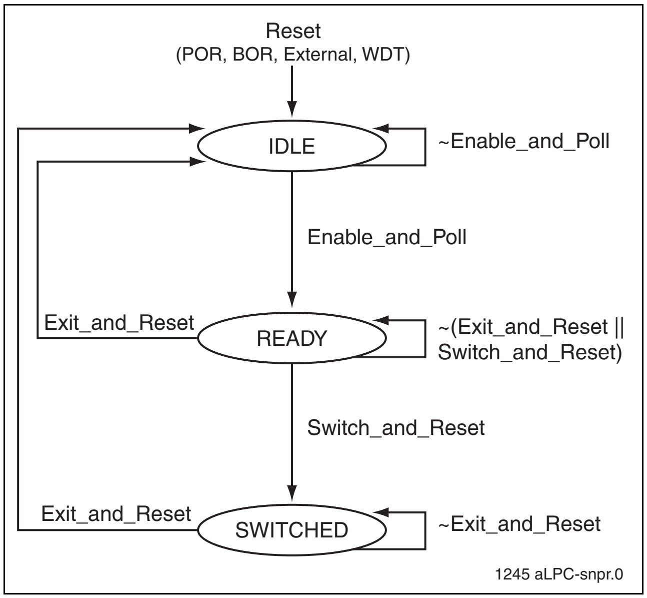


FIGURE 4-8: aLPC Snooper State Machine

aLPC Bus Control Register (ALPCBC)

Location		7	6	5	4	3	2	1	0
7F8BH	Read/ Write	RDY4ALPC	-	-	-	-	-	-	-
	Reset	0	X	X	X	X	X	X	X

Symbol	Function
-	Not implemented
X	Not defined
RDY4ALPC	8051 is ready for entering aLPC mode. While the Snooper is in IDLE or READY state 8051 can write to this bit. On entry into the SWITCHED state (aLPC mode) this bit is cleared by hardware. 1: 8051 is ready for aLPC mode. 0: 8051 is not ready for aLPC mode.

4.8.2 aLPC Access to BIOS and KBC Code

After entering SWITCHED state a downloader or hardware dongle which functions as an aLPC Host has access to the entire SST79LF008 Flash Memory including ENVR and UNVR. Both BIOS and KBC code can be programmed via the aLPC interface pins using aLPC Memory Write/Read cycles described in Sections 4.8.4 and 4.8.5, with aLPC flash commands listed in Section 4.8.6. For aLPC pin descriptions, see Table 4-7.

All flash memory protection mechanisms (block locking and mapping control) are disabled in aLPC mode.

TABLE 4-7: aLPC Pin Descriptions

Symbol	Pin Name	Type	Functions
aLCLK	Clock	I	To provide a clock input to the control unit (Host driven always)
aLAD	Address and Data	I/O	To provide aLPC bus information such as addresses, commands, and data (Host or device driven depending on the direction of the transfer.[1])
aLFRAME#	Frame	I/O	To indicate the start of an aLPC cycle (external aLPC Host driven), and the Ready or Busy status of the device (SST79LF008 driven[2]). Also used by the Host to abort an aLPC cycle in progress.

T4-7.0 1320

1. External pull-up resistor should be connected to the aLAD pin in order to maintain pin state during turn-around cycles.
2. The aLFRAME# signal is driven low to indicate Busy, and tri-stated to indicate Ready status. Hence, the external pull-up resistor should also be connected to the aLFRAME# pin.

4.8.3 aLPC Memory Write Operation with Auto-Address Increment and Multi-Byte Programming

The aLPC Mode with Auto-Address Increment (AAI) and Multi-Byte Programming features are provided for high-speed programming through the aLPC Host. The Auto-Address Increment in aLPC mode allows loading for only one address for each group of 1, 2, 4, 16, 128-byte, 4K, 64K, or 1M-byte of data. The Multi-Byte Programming feature supports multi-byte programming within one aLPC cycle.

All aLPC flash commands listed in Section 4.8.6 utilize aLPC Memory Write cycles described in this section. In aLPC mode aLAD is the only Data/Address input available, hence aLPC Memory Write cycles are similar to the standard LPC Firmware Memory Write cycles with the exception of using 4 clocks (instead of 1) to transfer each field as shown in Table 4-8-3. The aLPC interface also provides an extended handshaking mechanism between the aLPC Host and the ST79LF008 device for write cycles using the aLFRAME# pin.

When the aLPC Host starts a write cycle, the Host drives aLFRAME# low until the START field is transferred. Then the Host drives aLFRAME# high for four cycles. After those, the Host floats aLFRAME# and monitors the aLFRAME# signal, which is controlled by the SST79LF008 device in order to output Ready/Busy status. The SST79LF008 device would drive aLFRAME# low when the write-buffer is full or the flash memory is busy. When the SST79LF008 device is ready to receive data from the Host, aLFRAME# is switched to high.

Data in the aLPC Data Field is accepted until the internal buffer is full. At that point the device asserts the aLFRAME# low (busy) at the falling edge of aLCLK for the most significant bit of the last accepted nibble to indicate that the internal buffer is full and cannot accept any more data. The clock aLCLK must not be stopped during the program/erase procedure but any data present on aLAD line will be ignored while aLFRAME# is low. When the device is ready, the aLFRAME# signal is deasserted at the falling edge of aLCLK to indicate to the Host that more data (the next group of bytes/bits) can be accepted by the internal data buffer. At this point the host should generate 4 additional aLCLK clock pulses before sending any more data.

See Figure 4-9 for example of aLPC Memory Write cycle including handshaking operations.

Advance Information

TABLE 4-8: aLPC Memory Write Cycle Field Definitions

Clock Cycle	Field Name	Field Contents	SST79LF008 aLAD	Comments
0-3	START	0,1,1,1 LSb MSb	IN[1]	aLFRAME# must be active (low) for the part to respond. Only the last four clocks of the start field (before aLFRAME# transitioning high) should be recognized. The START field contents indicate aLPC Firmware Memory Write cycle (value = 1110). Order of bit transfer for this field: LS Bit first
4-7	IDSEL	0,0,0,0 (or 1,0,0,0) LSb MSb	IN[1]	ID selects SST79LF0008 device to respond. If the IDSEL field matches the device ID, then that particular device will respond to the whole bus cycle. Valid IDs = 0000 or 0001. Order of bit transfer for this field: LS Bit first
8-35	ADDR	28-bit address	IN[1]	These 28 clock cycles make up the 28-bit starting memory address A_{27}-A_0. Order of bit transfer for this field: MS Nibble first, LS Bit first: A24,A25,A26,A27, A20,A21,A22,A23, A16,A17,A18,A19, A12,A13,A14,A15, A8,A9,A10,A11, A4,A5,A6,A7, A0,A1,A2,A3
36-39	MSIZE	S0,S1,S2,S3 LSb MSb	IN[1]	Device will execute multi-byte write for N bytes. Valid field values S = 0, 1, 2, 4, 7, 12, 13, 14 For the respective N = 1, 2, 4, 16, 128, 4K, 64K, 1M bytes Order of bit transfer for this field: LS Bit first

40-(m-1)
m = 40+8*N+ # of wait cycles
	DATA	D0, D1, …, D(8*N)
		LSb MSb
			IN[1]	Data field consists of 8*N clock periods, where N is defined by MSIZE field. The host will insert wait cycles and pause the data stream when aLFRAME# goes low until it returns high, signifying that the chip is ready for more data.
				Order of bit transfer for this field: LS Nibble first, LS Bit first, thus DATA is transmitted starting with the least significant bit of Byte 0, sequentially to the most significant bit of byte (N-1).

(m)-(m+3)	TAR0	1,1,1,1	IN[1]	In these 4 clock cycles, the aLPC host has driven the aLAD pin to ‘1’. This is the first part of the bus turnaround.
(m+4)-(m+7)	TAR1	1,1,1,1 (float)	Float	The aLPC host floats the bus, and SST79LF008 takes control of the bus after these 4 cycles, completing bus turnaround.
(m+8)-(m+11)	RSYNC	0,0,0,0	OUT[2]	During these 4 clock cycles, the SST79LF008 generates a ready-sync (RSYNC) indicating that it has received data.
(m+12)-(m+15)	TAR0	1,1,1,1	OUT[2]	In these 4 clock cycles, SST79LF0008 has driven the bus to ‘1’. This is the first part of the bus turnaround.
(m+16)-(m+19)	TAR1	1,1,1,1 (float)	Float then IN[1]	SST79LF008 floats the bus, and the aLPC host resumes control of the bus after these 4 cycles, completing bus turnaround.

T4-8.0 1320

[1] SST79LF008 reads field contents on the falling edge of the present clock cycle

[2] Field contents are valid on the falling edge of the present clock cycle, and on the rising edge of the next clock cycle.

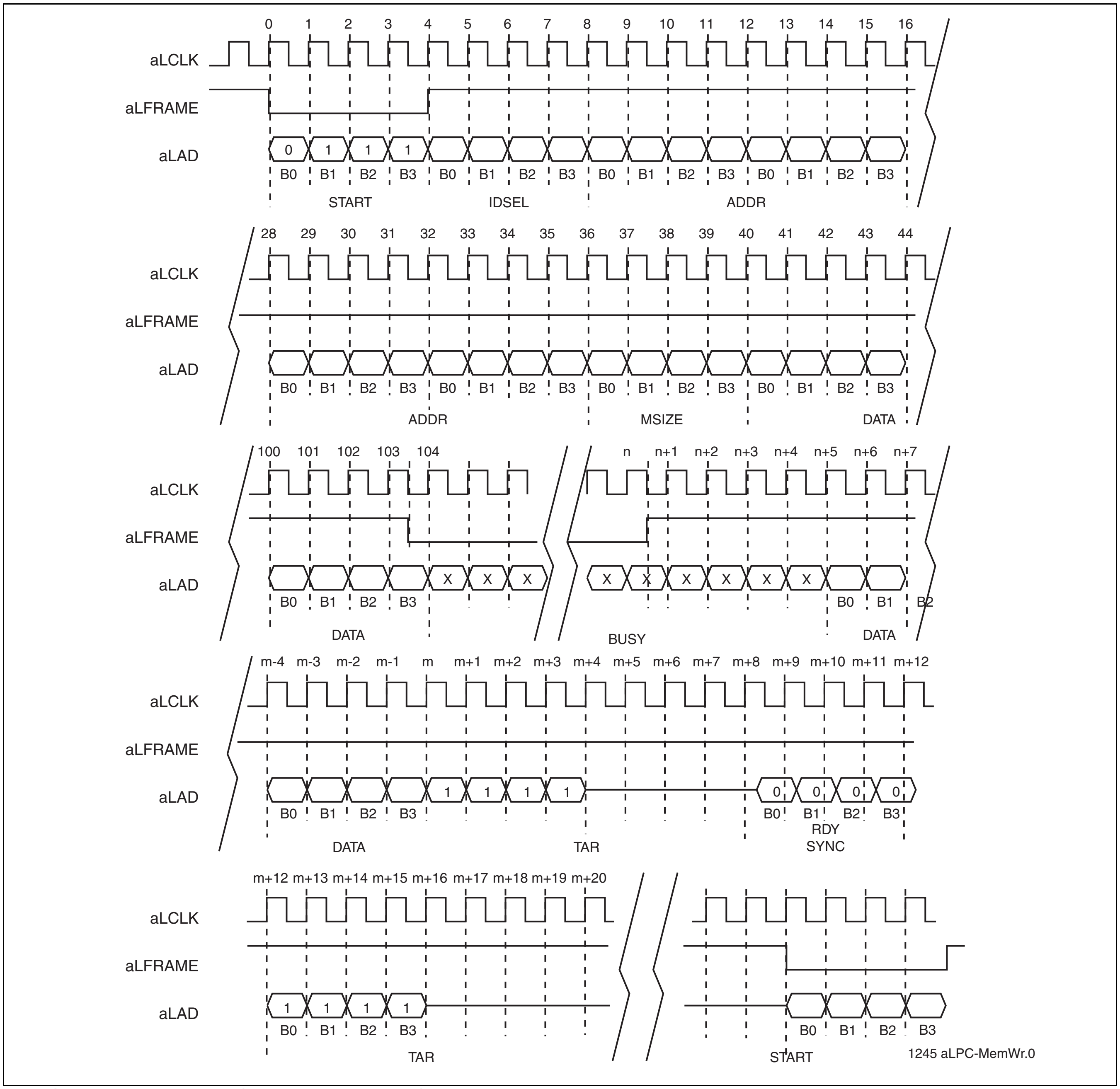


FIGURE 4-9: aLPC Memory Write Cycle

It is possible for SST79LF008 to signal busy status (drive aLFRAME# low) at the falling edge of the very last data bit (clock cycle m-1 on the diagram). This has no effect on the completion of current transaction, i.e., TAR-RSYNC-TAR sequence (clock cycles from m to m+19) is generated without any wait cycles inserted. However, the next LPC transaction must not be started by the Host until the aLFRAME# is returned to high level plus 4 spare clock cycles.

The aLPC Host can abort the transaction in progress by taking over and driving aLFRAME# low for at least 4 cycles with a START value of 1,1,1,1 (similar to the standard LPC abort mechanism). Because of the aLFRAME# extended functionality the following limitation for abort timing should be observed: the Host must not drive aLFRAME# in order to abort the transaction when aLFRAME# is asserted low by the SST79LF008 device as well as 4 clocks prior and 4 clocks after this event.

4.8.4 aLPC Memory Read Operation

In the aLPC mode the SST79LF008 device supports AAI and Multi-Byte Memory Read cycles, which allows to load only one address for reading a group of 1, 2, 4, 16, 128-bytes, 4K, 64K, or 1M-byte of data. These cycles are similar to the standard LPC Firmware Memory Read cycles with the exception of using 4 clocks (instead of 1) to transfer each field.

TABLE 4-9: aLPC Memory Read Cycle Field Definitions

Clock Cycle	Field Name	Field Contents	SST79LF008 aLAD	Comments
0-3	START	1,0,1,1 LSb MSb	IN[1]	aLFRAME# must be active (low) for the part to respond. Only the last four clocks of the start field (before aLFRAME# transitioning high) should be recognized. The START field contents indicate aLPC Firmware Memory Write cycle (value = 1101). Order of bit transfer for this field: LS Bit first
4-7	IDSEL	0,0,0,0 (or 1,0,0,0) LSb MSb	IN[1]	ID selects SST79LF0008 device to respond. If the IDSEL field matches the device ID, then that particular device will respond to the whole bus cycle. Valid IDs = 0000 or 0001. Order of bit transfer for this field: LS Bit first
8-35	ADDR	28-bit address	IN[1]	These 28 clock cycles make up the 28-bit starting memory address A$_{27}$-A$_0$. Order of bit transfer for this field: MS Nibble first, LS Bit first: A24,A25,A26,A27, A20,A21,A22,A23, A16,A17,A18,A19, A12,A13,A14,A15, A8,A9,A10,A11, A4,A5,A6,A7, A0,A1,A2,A3
36-39	MSIZE	S0,S1,S2,S3 LSb MSb	IN[1]	Device will execute multi-byte write for N bytes. Valid field values S = 0, 1, 2, 4, 7 For the respective N = 1, 2, 4, 16 Bytes Order of bit transfer for this field: LS Bit first
40-43	TAR0	1,1,1,1	IN[1]	In these 4 clock cycles, the aLPC host has driven the aLAD pin to ‘1’. This is the first part of the bus turnaround.
44-47	TAR1	1,1,1,1 (float)	Float	The aLPC host floats the bus, and SST79LF008 takes control of the bus after these 4 cycles, completing bus turnaround.
48-51	SYNC	0,1,1,0	OUT[2]	During these 4 clock cycles, the SST79LF008 generates a long-wait-sync (LWSYNC) indicating that data is not ready, yet.
52-(n-5)			OUT[2]	SST79LF008 continues to generate long-wait-sync.
(n-4)-(n-1)		0,0,0,0	OUT[2]	During these 4 clock cycles, SST79LF008 generates a ready-sync (RSYNC) indicating that the least-significant bit of the least significant data byte will be sent on the next clock cycle.

n-(m-1)
m = n+8*N
	DATA	D0, D1,
		…,
		D(8*N)
		LSb MSb
			OUT[2]	Data field consists of 8*N clock periods, where N is defined by MSIZE field. The host will insert wait cycles and pause the data stream when aLFRAME# goes low until it returns high, signifying that the chip is ready for more data.
				Order of bit transfer for this field: LS Nibble first, LS Bit first, thus DATA is transmitted starting with the least significant bit of Byte 0, sequentially to the most significant bit of Byte(N-1).

(m)-(m+3)	TAR0	1,1,1,1	OUT[2]	In these 4 clock cycles, SST79LF0008 has driven the bus to ‘1’ This is the first part of the bus turnaround.
(m+4)-(m+7)	TAR1	1,1,1,1 (float)	Float then IN[1]	The aLPC host floats the bus, and the aLPC host resumes control of the bus after these 4 cycles, completing bus turnaround.

T4-9.0 1320

[1] SST79LF008 reads field contents on the falling edge of the present clock cycle

[2] Field contents are valid on the falling edge of the present clock cycle, and on the rising edge of the next clock cycle.

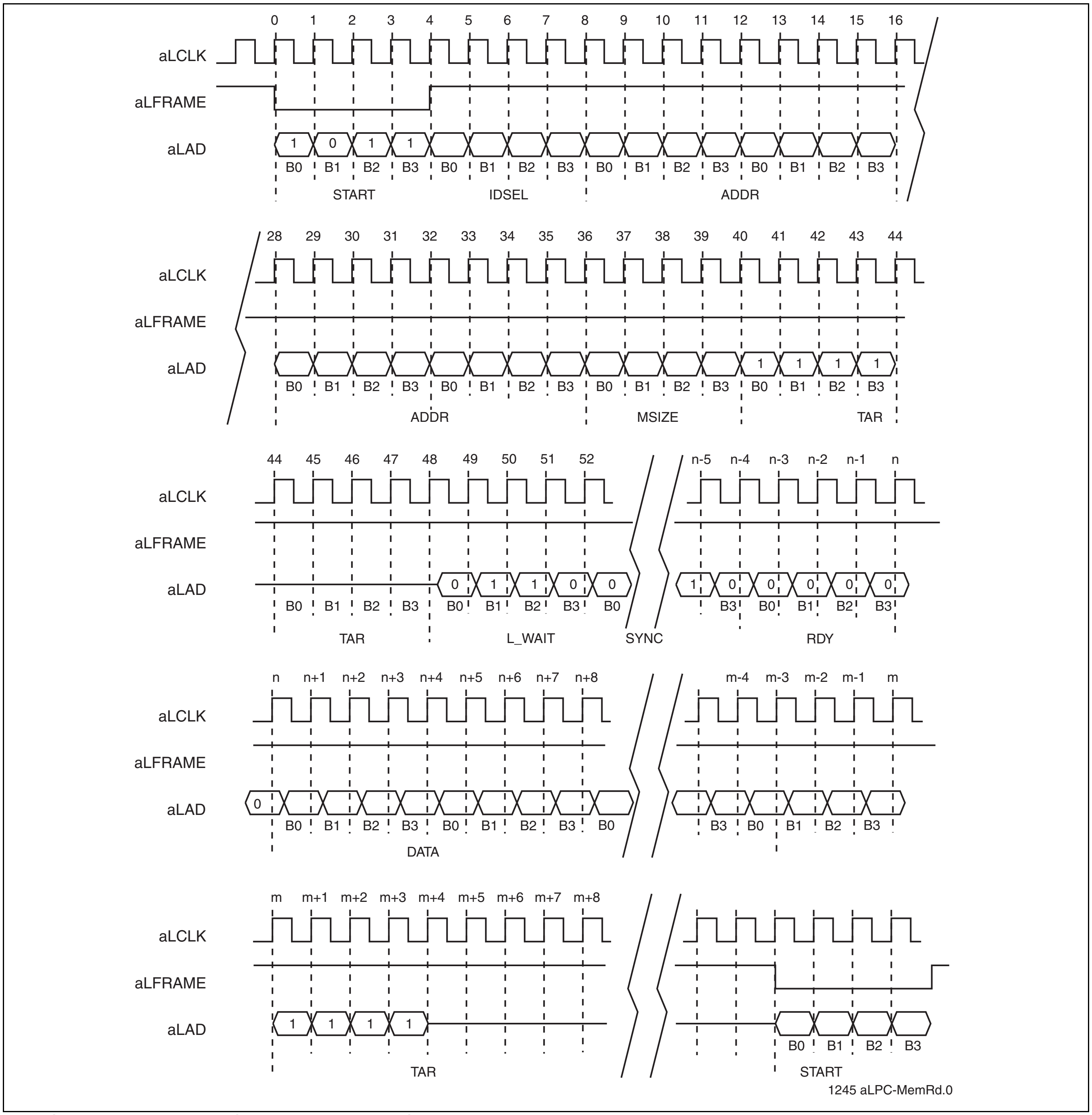


FIGURE 4-10: aLPC Memory Read Cycle

4.8.5 aLPC I/O Write Operation

When the SST79LF008 device is switched into aLPC mode it supports aLPC I/O Write and Read cycles in addition to the aLPC Memory Write and Read cycles previously described. However, in aLPC mode all LPC logical I/O devices are disabled (as all Configuration registers are reset - see Section 23.2). Therefore the only useful aLPC I/O operations are aLPC Write cycles utilized by the aLPC Host to issue aLPC Snooper commands described in Table 4-6. These cycles are similar to the standard LPC I/O Write cycles except 4 clocks (instead of 1) are used to transfer each field. The following exceptions with regards to aLAD and aLFRAME# control are also applied to the aLPC I/O cycles when the Snooper is in IDLE and READY states:

- In IDLE and READY states the Snooper tri-states the aLAD line and does not return RSYNC to the Host
- In READY state the Snooper returns the status of RDY4ALPC bit on aLFRAME# during SYNC phase of the fourth write cycle of the Enable_and_Poll sequence.

Examples of I/O Write cycles before and after entry to SWITCHED state (aLPC mode) are shown in Figures 4-11 and 4-12, respectively.

TABLE 4-10: aLPC I/O Write Cycle Field Definitions

Clock Cycle	Field Name	Field Contents	SST79LF008 aLAD	Comments
0-3	START	0,0,0,0 LSb MSb	IN[1]	aLFRAME# must be active (low) for the part to respond. Only the last four clocks of the start field (before aLFRAME# transitioning high) should be recognized. The START field contents indicate aLPC Firmware Memory Write cycle (value = 0000). Order of bit transfer for this field: LS Bit first
4-7	CYC_TYPE	0,1,0,0 LSb MSb	IN[1]	Cycle type field indicates aLPC I/O Write cycle (value - 0010) Order of bit transfer for this field: LS Bit first
8-23	ADDR	16-bit address	IN[1]	These 16 clock cycles make up the 16-bit I/O address A_{15}-A_0. Order of bit transfer for this field: MS Nibble first, LS Bit first: A12,A13,A14,A15, A8,A9,A10,A11, A4,A5,A6,A7, A0,A1,A2,A3
24-31	DATA	D0,D1,...D7 LSb MSb	IN[1]	These 8 clock cycles are used to transmit one Data byte. Order of bit transfer for this field: LS Bit first
32-35	TAR0	1,1,1,1	IN[1]	In these 4 clock cycles, the aLPC host has driven the aLAD pin to '1'. This is the first part of the bus turnaround.
36-39	TAR1	1,1,1,1 (float)	Float	The aLPC host floats the bus, and SST79LF008 takes control of the bus after these 4 cycles, completing bus turnaround.
40-43	SYNC	0,0,0,0	OUT[2,3]	During these 4 clock cycles, the SST79LF008 generates a ready-sync (RSYNC) indicating that data is not ready, yet.
44-47	TAR0	1,1,1,1	OUT[2,3]	In these 4 clock cycles, the aLPC host has driven the bus pin to '1'. This is the first part of the bus turnaround.
48-51	TAR1	1,1,1,1 (float)	Float then IN[1]	SST79LF008 floats the bus, and the aLPC host resumes control of the bus after these 4 cycles, completing bus turnaround.

T4-10.0 1320

[1] SST79LF008 reads field contents on the falling edge of the present clock cycle
[2] Field contents are valid on the falling edge of the present clock cycle, and on the rising edge of the next clock cycle.
[3] When the Snooper is in IDLE or READY state, aLAD is always floated in SST79LF008 (and SYNC = 1,1,1,1).

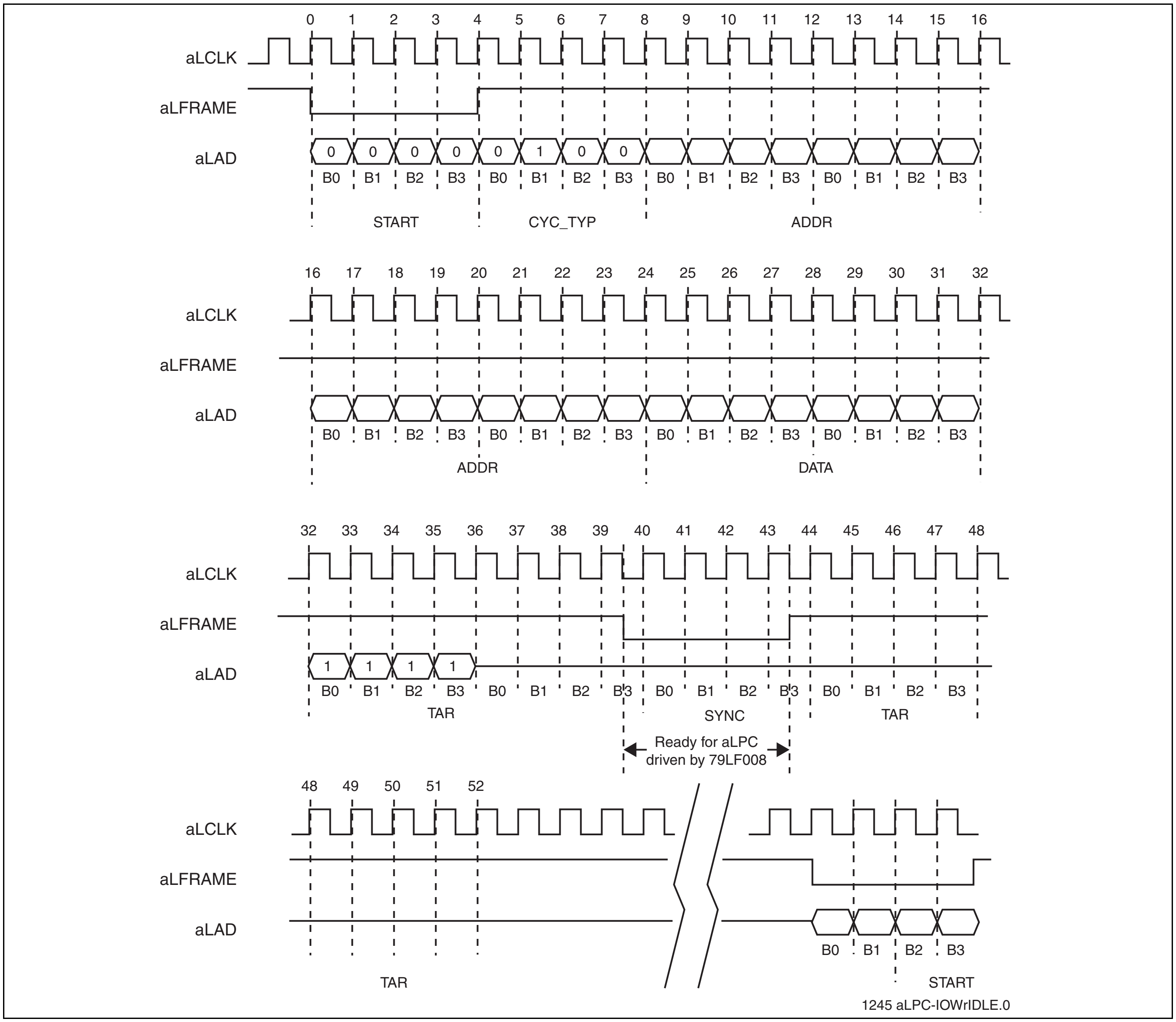


FIGURE 4-11: aLPC I/O Write Cycle (IDLE or READY state)

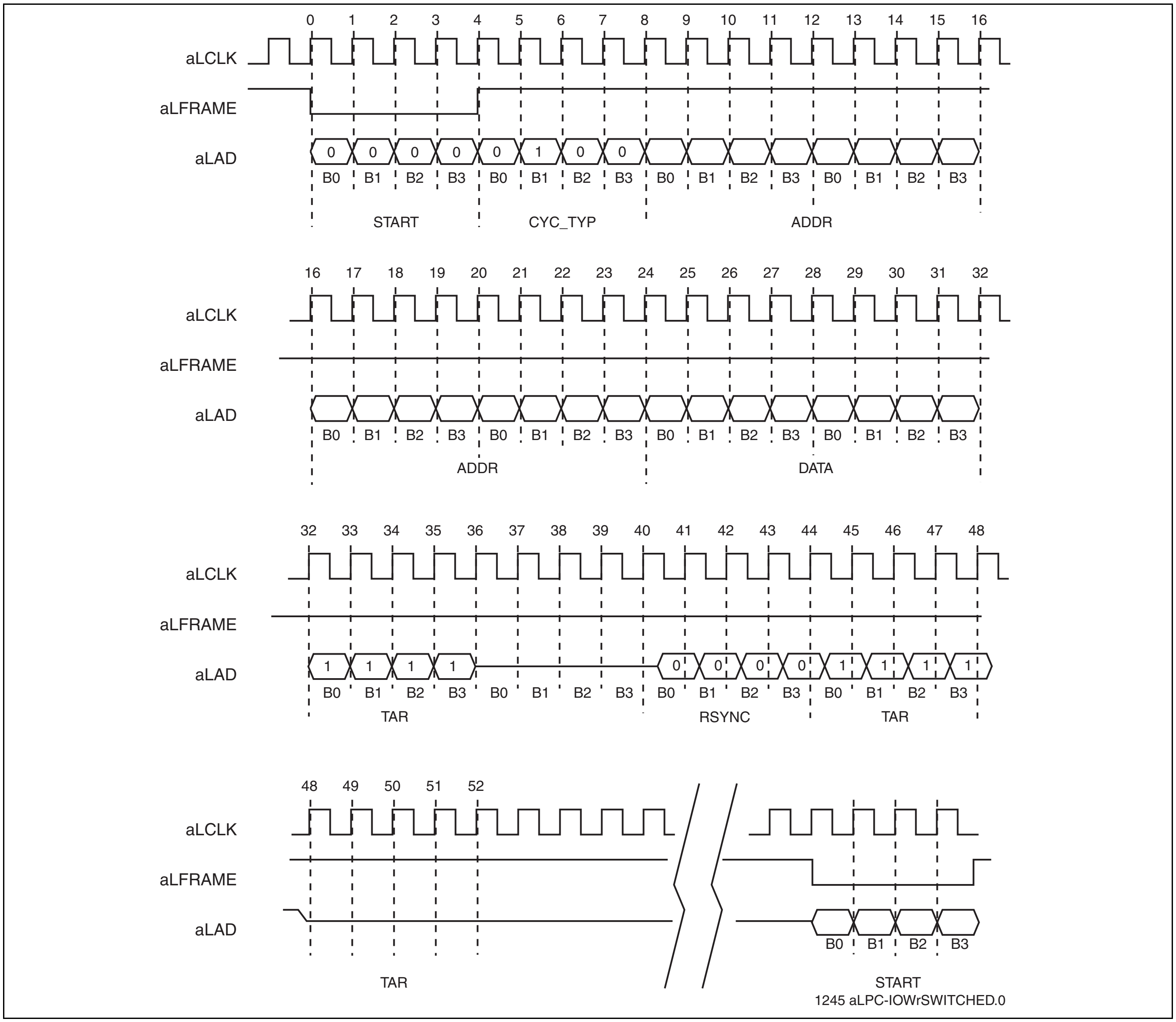


FIGURE 4-12: aLPC I/O Write cycle (SWITCHED state)

4.8.6 aLPC Flash Commands

When aLPC Mode is enabled (i.e., the aLPC Snooper is in SWITCHED state) all aLPC Memory Write operations are interpreted by the flash command interface. Commands consist of one or more sequential bus write cycles described in Section 4.8.4. The aLPC flash commands are summarized in Table 4-11. For a detailed description of each command, refer to Section 7.3, as aLPC and standard LPC flash command are functionally similar.

TABLE 4-11: aLPC Bus Flash Command Definitions

Command	Bus Cycles Required	First Bus Cycle			Second Bus Cycle		
		Oper	Addr	Data	Oper	Addr	Data
Read-Array/Reset	1	Write	X	FFH			
Read-Device-ID Read Unique ID	≥ 2	Write	X	90H	Read	IA	ID
Read-Status-Register	2	Write	X	70H	Read	X	SRD
Clear-Status-Register	1	Write	X	50H			
Sector-Erase	2	Write	X	30H	Write	SA	D0H
Block-Erase	2	Write	X	20H	Write	BA	D0H
Program	2	Write	X	40H or 10H	Write	WA	WD
Erase-Suspend	1	Write	X	B0H			
Erase-Resume	1	Write	X	D0H			
User-Unique-ID Program	2	Write	X	A5H	Write	WA	WD
User-Unique-ID Program-Lockout	2	Write	X	85H	Write	X	00H
Enter UNVR (3K OTP) / Enter ENVR	2	Write	X	60H	Write	X	76H

T4-11.0 1320

5.0 POWER, RESET AND CLOCK SOURCES

5.1 Power Planes

SST79LF008 has two power planes: V_{DD} and $AV_{DD.}$ The digital logic and on-chip memory are powered by $V_{DD;}$ the analog circuits in ADC and DAC are powered by AV_{DD}. Both power planes must be applied/removed simultaneously.

SST recommends a high frequency 0.1 µF ceramic capacitor to be placed as close as possible between each V_{DD} and V_{SS} pin, less than 1cm away from V_{DD} pin. Additionally, a low frequency tantalum capacitor (4.7 µf min.) from V_{DD} to V_{SS} should be placed on the common power/ground net as close as possible to the chip.

SST recommends a high frequency 0.1 µF ceramic capacitor to be placed as close as possible between each AV_{DD} and AV_{SS} pin, less than 1cm away from AV_{DD} pin. Additionally, a low frequency tantalum capacitor (4.7 µf min.) from AV_{DD} to AV_{SS} should be placed on the common analog power/ground net as close as possible to the chip.

Note also that a high frequency 1 µF ceramic capacitor must be placed as close as possible between SST79LF008 internal regulator output VREG and V_{SS} pin, less than 1 cm away from V_{REG} pin.

For correct SST79LF008 operation all power pins must be connected to the respective power/ground planes, see Section 2.1, “Pin Descriptions”.

5.2 Reset Sources

SST79LF008 has the following reset sources:

1. Power-On reset (POR)
2. External Chip reset (RESET# pin)
3. Brown-out detection reset (BOR)
4. Watchdog timer (WDT) reset
5. aLPC Soft reset
6. LPC Soft reset
7. 8051 Soft reset
8. Configuration Soft reset
9. LPC Interface reset (LRESET# pin)

TABLE 5-1: SST79LF008 Reset Sources

Reset Source	Reset Mechanism	Reset Modules	Restart 8051 code after reset	Maskable	Idle Wake up	PD Wake up
POR	V_{DD} is powered up	The SST79LF008 chip	Yes[1]	No	Yes	Yes
External Reset	RESET# pin is asserted low	The SST79LF008 chip	Yes[1]	No	Yes	Yes
BOR	V_{DD} is below Brown-Out threshold	The SST79LF008 chip	Yes[1]	Yes	Yes	No
WDT reset	Watchdog timer underflow	The SST79LF008 chip	Yes[1]	Yes	Yes	Yes
aLPC Soft Reset	(a) Entry to aLPC mode via aLPC Switch_and_Reset sequence (b) Exit from aLPC mode via aLPC Exit_and_Reset sequence	The SST79LF008 chip, except the aLPC snooper	Yes[1]	No	Yes	Yes
LPC Soft Reset	Force LPC Soft Reset command received via LPC bus	8051 MCU, selected MMCRs, and Configuration registers; clear SFSC[2] (OVERLAY) bit	Yes[2]	Yes	Yes	Yes
8051 Soft Reset	Transition of SOFTRST = SFCS[3] bit from ‘0’ to ‘1’	8051 MCU and selected MMCRs; clear SFSC[2] (OVERLAY) bit	Yes[2]	No	N/A	N/A
Configuration Soft Reset	Transition of Bit 0 in Chip Control register 0 from ‘0’ to ‘1’	Configuration registers	No	No	N/A	N/A
External LPC Interface reset	LRESET# pin is asserted low	The LPC bus interface and flash block locking registers	No	No	Yes	Yes

T5-1.0 1320

[1] Start address and OVERLAY bit state is determined by ENVR contents as described in Section4.4.5.

[2] Start address is 0000H regardless of ENVR contents and OVERLAY bit is always cleared.

The SST79LF008 reset block diagram is shown on Figure 5-1. POR, External Reset, BOR, and WDT resets are internally “or-ed” to form an internal reset signal to the whole SST79LF008 chip. This signal is asserted as soon as any of the above reset sources is asserted and will be de-asserted only when none of them is asserted. The 16-bit reset extender counter guarantees that the chip will be held in reset state for at least 65,536 clocks of RCLK after all these reset sources are de-asserted. The extended reset signal applied to aLPC Snooper and “or-ed” with aLPC Soft reset to form a flash memory reset. The 7-bit configuration extender counter keeps 8051 MCU and other peripheral modules in reset state for 128 clocks of RCLK, while the internal configuration is performed in hardware. After the configuration extender is expired and the internal Chip Logic Reset signal is de-asserted, all SFRs (see Section 3.4), MMCRs (see Section 3.5), flash block locking registers (see Section 7.6), and configuration registers (see Section 23.2) contain their respective reset values. The 8051 program execution will restart from either address 0000H or F000H (BootRom) depending on the ENVR contents as described in Section 4.5.

The 8051 firmware Soft Reset affects the 8051 MCU core including all SFRs, and selected MMCRs as specified in Section 3.5. After the 8051 Soft reset the OVERLAY bit is always cleared. Thus, the 8051 program execution will restart from address 0000H.

The LPC Soft reset affects the 8051 MCU core including all SFRs, selected MMCRs as specified in Section 3.5, and all Configuration registers. After the LPC Soft reset the OVERLAY bit is always cleared. Thus, the 8051 program execution will restart from address 0000H.

The Configuration Soft reset affects configuration registers only; it does not restart 8051 code execution.

The LPC Interface reset signal initializes LPC interface state machine (including Serialized IRQ, and CLKRUN# mechanisms), and resets flash memory command sequence described in Section 7.3 as well as all flash block locking registers described in Section 7.6. The LPC reset does not restart 8051 code execution.

Note. The internal SRAM is not affected by any type of reset. On power up, the SRAM content is indeterminate.

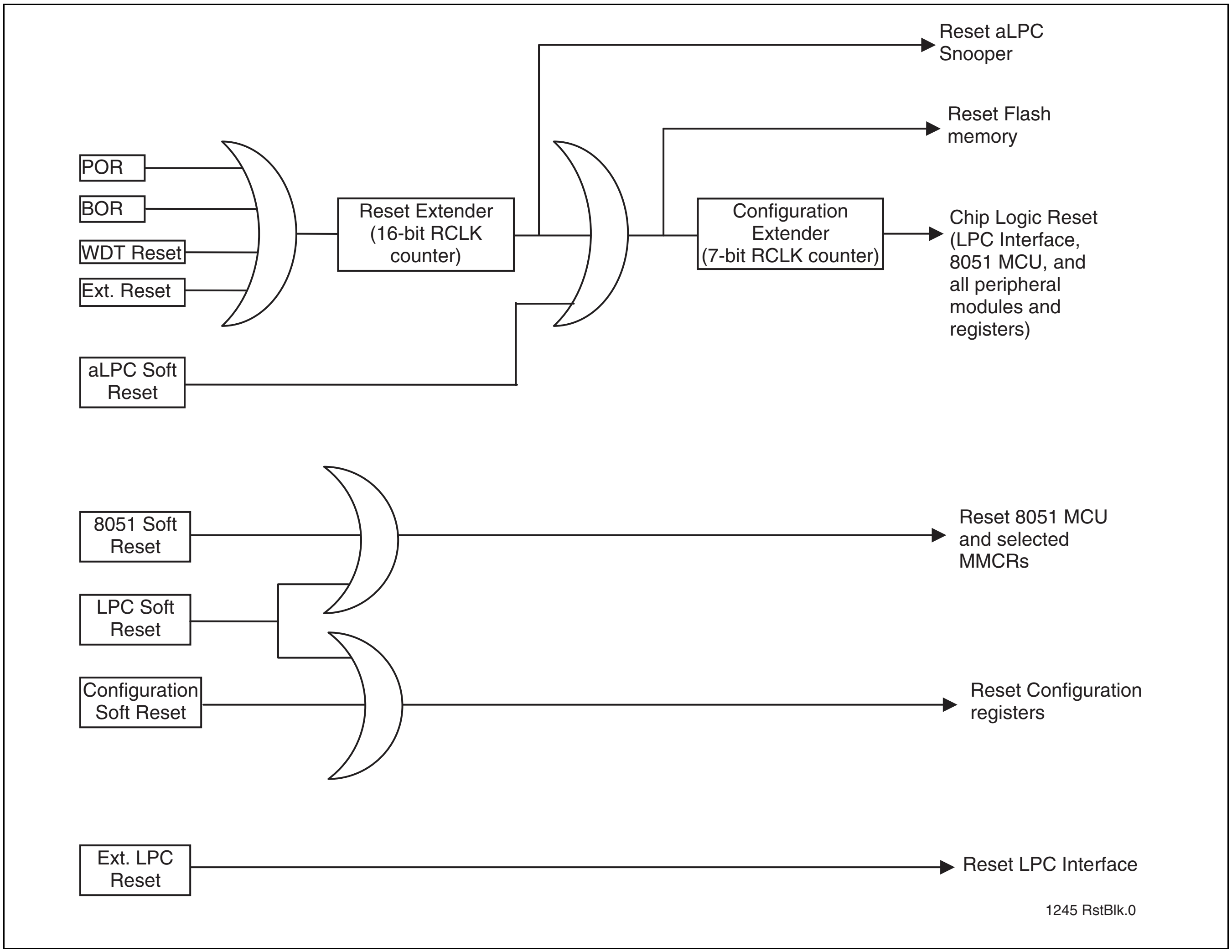

POR
BOR
WDT Reset
Ext. Reset
Reset Extender (16-bit RCLK counter)
aLPC Soft Reset
Reset aLPC Snooper
Reset Flash memory
Configuration Extender (7-bit RCLK counter)
Chip Logic Reset (LPC Interface, 8051 MCU, and all peripheral modules and registers)
8051 Soft Reset
LPC Soft Reset
Configuration Soft Reset
Reset 8051 MCU and selected MMCRs
Reset Configuration registers
Ext. LPC Reset
Reset LPC Interface
1245 RstBlk.0

FIGURE 5-1: Reset Block Diagram

5.2.1 Power-On Reset

The SST79LF008 provides an internal power-on reset circuit, which generates Power-On Reset when V_{DD} power is applied. The POR is extended internally until ring oscillator clock is stabilized and counted at least 65,536 times. See Figure 5-1. However, the 32.768 KHz oscillator has much larger power on stabilization time. Therefore, accuracy of any module in the XCLK domain (see Table 5-3) is not guaranteed immediately after POR. If necessary the external RC circuitry can be used to additionally extend POR, see Figure 5-2, or 8051 firmware can implement software delay before using XCLK controlled devices.

The POF flag in PCON register is set to indicate that Power-On Reset has occurred, see Section 11.4.

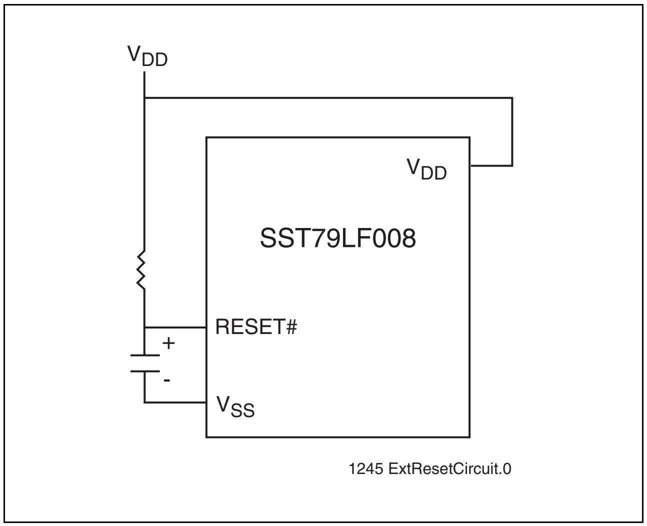


FIGURE 5-2: External Reset Circuit

5.2.2 External Reset

An External reset, derived from the external circuitry connected to the RESET# pin of SST79LF008, will reset the entire chip. In order to properly reset the device, a logic level low must be applied to the RESET# pin for at least 48 cycles of 8051 core clock (CCLK) once V_{DD} power is already applied. External reset can not be masked. If External reset occurs when the chip is in Idle or Power Down mode, it will cause the chip to exit the respective mode.

5.2.3 Brown-out Detection Reset

The device includes a Brown-Out detection circuit to protect the system from severe supply voltage V_{DD} fluctuations. When V_{DD} drops below the brown-out voltage threshold, the detector circuit generates a brown-out reset to reset the whole device. The BOR is de-asserted automatically after V_{DD} exceeds the brown-out voltage threshold. Brown-out reset can be masked via BOREN bit in RSTCON register (by default BOR is enabled). If BOR occurs when the chip is in Idle mode, it will cause the chip to exit the respective mode. BOR is not generated when chip is in Power Down mode.

The BOF flag in PCON register () is set when brown-out condition is detected, provided BOR is enabled. See Section 11.4.

5.2.3.1 Reset Control Register (RSTCON)

Location		7	6	5	4	3	2	1	0
7F2EH	Read	-	-	-	-	-	WDTPOL	WDTRSTEN	BOREN
	Write								
	Reset	X	X	X	X	X	0	0	1

Symbol	Function
-	Not implemented
X	Not defined
WDTPOL	WDT Output Polarity control bit 1: WDT output is High Active (when WDT underflows) 0: WDT output is Low Active (when WDT underflows)
WDTRSTEN	WDT Reset Enable bit 1: Enable WDT Reset 0: Disable WDT Reset
BOREN	BOR Enable bit (When in Power Down mode, BOR is disabled regardless of this bit state.) 1: Enable BOR 0: Disable BOR

5.2.4 Watchdog Timer (WDT) Reset

Watchdog Timer (WDT) reset is generated when WDT underflows (i.e., when firmware failed to reload WDT within the programmed watchdog interval). WDT reset affects the entire device. It can be masked by WDTRSTEN bit in RSTCON register (by default WDT reset is disabled). In addition WDT controls allow firmware to stop WDT completely, or only in Power Down mode. For more WDT operation details, see Section 10.0.

5.2.5 aLPC Soft Reset

A Switch_and_Reset sequence, sent over an aLPC bus, resets the whole chip with the exception of aLPC Snooper. Then the SST79LF008 device will enter aLPC mode with flash memory bus owned by the aLPC Host and 8051 MCU held in reset, until aLPC exit sequence terminates aLPC mode. After Exit_and_Reset sequence is received the SST79LF008 chip is reset again. The reset extender does not affect aLPC Soft reset duration. The aLPC Host is required to provide at least 8 spare aLPC clock cycles after the end Switch_and_Reset or Exit_and_Reset sequence. For the aLPC sequences details, see Section 4.8. The aLPC Soft reset can not be masked. If it occurs when the chip is in Idle or Power Down mode, the respective mode is terminated.

5.2.6 LPC Soft Reset

A Force LPC Soft Reset command sent over LPC bus will reset the 8051 MCU core, including all SFRs, selected MMCRs and all Configuration registers. After this command flash memory bus is owned by the LPC Host, and 8051 is held in reset state, until a Release LPC Soft Reset command is received. The reset extender and configuration extender counters do not affect LPC Soft reset. See Section 7.3.10 for command details and Section 3.5 for the list of affected MMCRs. The LPC Soft reset can be masked via LRSTCOREENB bit in LPC bus monitor register described in Section 23.1. If LPC Soft reset occurs when the chip is in Idle or Power Down mode, the respective mode is terminated.

5.2.7 8051 Firmware Soft Reset

A transition from 0 to 1 of SFCS[3] (SOFTRST) bit will generate an internal reset signal, which resets the 8051 MCU core and selected MMCRs. The reset extender and configuration extender counters do not affect 8051 Soft reset. See Section 3.5 for the list of affected MMCRs.

5.2.8 Configuration Soft Reset

When bit 0 of the chip control register 0 transitions from 0 to 1, only the configuration registers are reset. For more details, see Section 23.2.

5.2.9 LPC Interface Reset

LPC Interface reset is controlled by the LRESET# input. When the LRESET# signal is active low, the SST79LF008 device ignores the LFRAME# and LCLK inputs, and tri-states the address/data signals of LPC bus LAD[3:0], as well as SERIRQ and CLKRUN# outputs. The LRESET# resets LPC interface state machine (including SERIRQ and CLKRUN# mechanisms), flash command sequencer and flash block locking registers described in Section 7.6. During this process, the STICKY_LK bit in SFSEC register is also cleared. The reset extender and configuration extender counters do not affect LPC Interface reset. The LRESET# can not be masked, and it generates an interrupt to 8051, which can be configured as a wake up event from Idle or Power Down mode.

If LRESET# signal is activated while flash memory Erase is in progress, the SRI hardware automatically suspends Erase operation and switches flash memory to read array mode during the LPC Host read access after LRESET# is de-asserted. LPC read operation within any flash sector/ block, other than the suspended one, would complete normally in this case. The erase operation will automatically resume on completion of the LPC Host read. If LPC reset occurs when flash bus was turned over to the LPC Host (using mechanism (a) or (b) described in Section 4.3), it is LPC Host software and 8051 firmware's responsibility to restore 8051 flash bus ownership after LRESET# is de-asserted and erase operation is finished.

If LRESET# signal is activated after Force LPC Soft Reset command, the flash bus will be automatically released to 8051, and 8051 will restart code execution when LRESET# is de-asserted. In a case when erase operation was initiated during LPC Soft Reset, and aborted by LRESET# assertion (i.e., mechanism (c) in Section 4.3 was used), the 8051 will not be able to properly fetch flash memory contents, making code execution results unpredictable. Therefore mechanism (c), is not recommended for flash update unless flash memory is blank or corrupted.

The LRESET# signal has no effect in aLPC mode.

5.3 Clock Sources

5.3.1 Clock Input Options

The SST79LF008 has several clock sources which are listed in Table 5-2.

TABLE 5-2: SST79LF008 Clock Sources

Clock Source	Frequency	Destination
Internal Ring Oscillator Clock (RCLK)	10-20MHz	8051 CPU and core peripherals
External Clock (ECLK) signal	4-16MHz	PLL, 8051 CPU and core peripherals
Internal PLL-Generated Clock (PCLK), derived from ECLK	Up to 33 MHz	8051 CPU and core peripherals
External crystal connected to the internal oscillator circuit (XCLK)	32.768KHz	Hibernation timer, Timer0, Timer1, Watchdog timer, Fan tachometer, and PWM
External LPC Clock (LCLK)	Up to 33MHz	LPC Bus Interface
External aLPC Clock (aLCLK)	Up to 5MHz	aLPC Bus Interface

T5-2.0 1245

Internal clock sources include Ring Oscillator and PLL, which, however, depends on external clock ECLK. External clock sources include ECLK as well as LPC clocks LCLK and aLCLK. The 32.768 kHz clock is generated by the circuitry which combines an on-chip oscillator and an external crystal as shown on Figure 5-3.

5.3.1.1 Crystal Oscillator

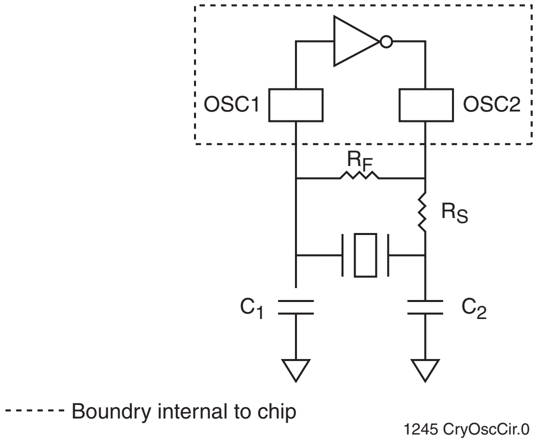


FIGURE 5-3: Crystal Oscillator Circuit

TABLE 5-3: Crystal Oscillator Circuit Components

R_F	R_S	C1,C2	Crystal Frequency
10MΩ	0 -- 200KΩ	10 -- 35pF	32.768KHz

T5-3.0 1245

5.3.2 Clock Selection Control and Clock Domains

The 8051 CPU and core peripherals clock (CCLK) are derived from three different clock sources—the on-chip ring oscillator (RCLK), the PLL clock (PCLK), and the external clock source (ECLK). The 8051 can be programmed to switch the clock source during normal code execution. The 8051 clock source can be selected through the Clock Source Control Register (CLKSRCCON). In addition peripheral modules Time0, Timer1, and PWM have their own clock selection control registers described in Section 10.0. The 8051 core clock, as well as some peripheral clocks, are stopped when 8051 enters either Idle mode or Power Down mode. for PCON register description, see Section 11.4.

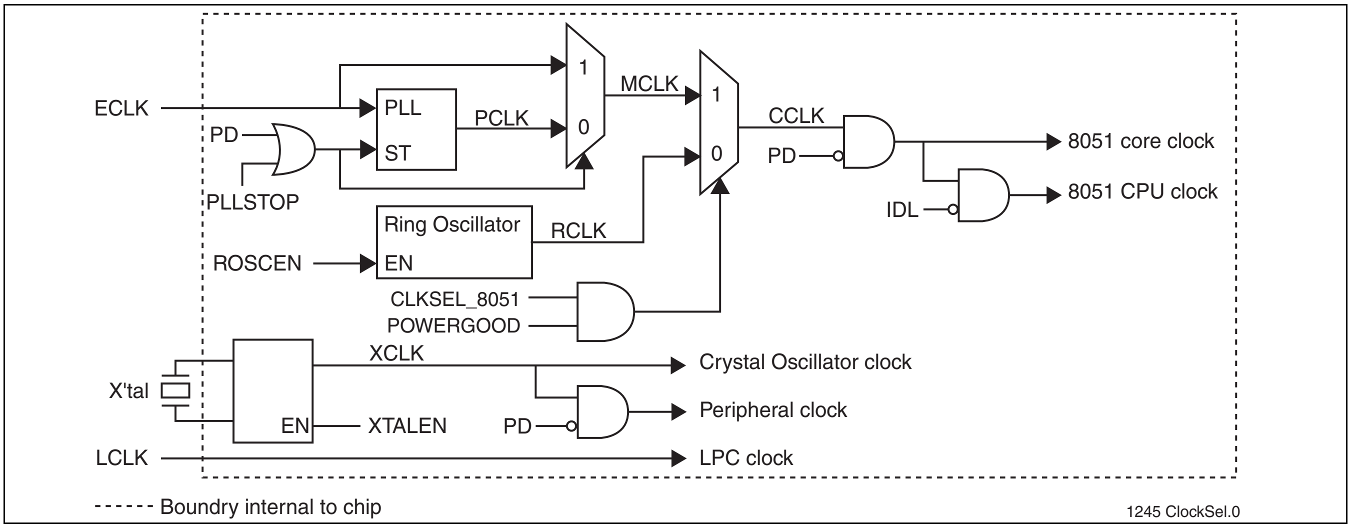


FIGURE 5-4: SST79LF008 Clock Selection

The SST79LF008 clock usage and clock domains are summarized in the Table 5-4.

TABLE 5-4: Clock Domains for SST79LF008 Modules

MODULE	Clock Domain	Clock status in Idle mode	Clock status in Power Down
8051	8051 CPU clock	stop	stop
Timer 0/1	8051 core clock	run	stop
	Peripheral clock	run	stop
Timer 2	8051 core clock	run	stop
Hibernation timer	Crystal Oscillator clock	run	run
WDT	Peripheral clock	run	run/stop depends on (WDTCSR[7])
Fan Tachometer	Peripheral clock	run	run/stop depends on (FANTIMEBASE[7:6])
PWM	8051 core clock	run	stop
	Crystal Oscillator clock	run	run
ADC	8051 core clock	run	stop
DAC	8051 core clock	run	stop
PWM LED	Crystal Oscillator clock	run	run
UART	8051 core clock	run	stop
SPI	8051 core clock	run	stop
SMBus	8051 core clock	run	stop
PS/2	8051 core clock	run	stop
GPIO	8051 core clock	run	stop (retain I/O state)
LPC Interface	LPC clock	X[1]	X[1]

T5-4.1245

[1] LPC clock status is controlled by the external LPC Host, it is not affected by 8051 Idle or Power Down mode.

5.3.2.1 Clock Source Control Register (CLKSRCCON)

Location		7	6	5	4	3	2	1	0
7F27H	Read	-	-	CLKSEL_ 8051	ROSCEN	PLLOK	XTALEN	PLLSTOP	ECLKOK
	Write					-			-
	Reset	X	X	0	1	0	1	0	0

Symbol **Function**

- Not implemented

X Not defined

CLKSEL_8051 8051 clock source selection bit
1: 8051 Clock source CCLK = MCLK (i.e., either PCLK or ECLK)
0: 8051 Clock source CCLK = RCLK

ROSCEN Ring Oscillator Enable bit
The 8051 firmware can read/write this bit. It is also reset by hardware on entry to Power Down mode, and set on wake up from Power Down mode
Enable Ring oscillator
0: Stop Ring oscillator

PLLOK PLL status bit
This bit is set when PLL output signal is stabilized. If firmware attempts to select PCLK as 8051 core clock source when this bit is cleared, hardware uses RCLK instead, until PLLOK is set. Only after PLLOK is set 8051 core clock is switched to PCLK. An interrupt is generated when PLLOK bit changes from 0 to 1.
1: PLL is stabilized (the PLL stabilization counter reaches 3F00H = 16,128)
0: PLL is not stabilized because of any of the following 3 conditions

PLL stopped (PLLSTOP =1, PLL stabilization counter is reset)

ECLK stopped (ECLKOK = 0, PLL stabilization counter is reset)

PLL stabilization time has not expired (PLLSTOP = 0, ECLKOK = 1, PLL stabilization counter is running but has not reached 3F00H, yet). This bit and PLL stabilization counter are cleared in hardware on entry to Power Down mode.

XTALEN Crystal Oscillator Enable bit
1: Enable 32.768KHz crystal oscillator
0: Disable 32.768KHz crystal oscillator

PLLSTOP PLL control and MCLK selection bit
1: Select MCLK = ECLK (PLL module is stopped, stabilization counter is reset)
0: Select MCLK = PCLK (PLL module is running)

ECLKOK External clock ECLK status bit
This bit is set when ECLK is applied. If firmware attempts to select ECLK as 8051 core clock when this bit is cleared, the hardware uses RCLK instead, until ECLKOK is set. Only after ECLKOK is set to 8051 core clock is switched to ECLK. An interrupt is generated when ECLKOK changes from 0 to 1 or from 1 to 0
1: ECLK is running
0: ECLK has stopped

5.3.2.2 PLL M Value Register (PLLM)

Location		7	6	5	4	3	2	1	0
7F35H	Read	-	-	PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0
	Write								
	Reset	X	X	1	0	0	0	1	0

5.3.2.3 PLL PS Value Register (PLLPS)

Location		7	6	5	4	3	2	1	0
7F36H	Read	-	-	PLLP1	PLLP0	PLLS3	PLLS2	PLLS1	PLLS0
	Write								
	Reset	X	X	1	1	0	0	1	0

Symbol	Function
-	Not implemented
X	Not defined
PLLM[5:0], PLLP[1:0], PLLS[3:0]	PLL Output frequency control*† $F_{PLLO} = F_{PLLI} * m/(p*s)$ $m = 2*(M+2)$ $p = (P+1)$ $s = 2*(S+2)$

F_{PLLI} (MHz) (ECLK)	P	M	S	F_{PLLO} (MHz) (PCLK)
4	0	34	7	16
4	0	34	4	24
4	0	30	2	32
8	1	34	7	16
8	1	34	4	24
8	1	30	2	32
12	2	34	7	16
12	2	34	4	24
12	2	30	2	32
14.318	3	38	7	15.9089
14.318	3	38	4	23.8633
14.318	3	34	2	32.2155
16	3	34	4	24
16	3	30	2	32

T5-4.1245

5.3.3 Clock Switching after Power On and Reset

When SST79LF008 is powered on, the default clock source is always the internal ring oscillator. The RCLK continues to be used until all of the following events have taken place.

1. The 65,536 + 128 counts of RCLK are completed
2. ECLK is running
3. PWRGOOD pin is high
4. RCLK is not selected by firmware as the core clock (i.e., CLKSEL_8051 = 1)
5. The PLL output clock (if enabled), has been through 16,128 (3F00H) counts

* When changing PLL output frequency (i.e. changing PLLM and/or PLLPS register) PCLK must not be selected as 8051 core clock

† When changing PLL output frequency (i.e. changing PLLM and/or PLLPS register) PCLK must not be selected as 8051 core clock

If all these five conditions are satisfied, a smooth transition from ring oscillator clock to the desired clock source ECLK or PCLK will automatically take place. Until then, the ring oscillator clock is supplied to the 8051 core. The 8051 starts execution as soon as it is released from reset using the oscillator clock, and then switches to another clock later on at run time. See Figure 5-4.

The described clock selection procedure is followed after External reset, WDT reset and aLPC Soft reset. Then, SST79LF008 will use the selected clock as long as software clock selection is not changed, ECLK is available, and PWRGOOD signal is high.

5.3.3.1 Power Good Signal

PWRGOOD is a signal that indicates the system power is ‘good’. The SST79LF008 device uses this signal as an external indicator of the availability of the ECLK clock. When PWRGOOD goes low, it indicates that the ECLK will be going away. Therefore SST79LF008 will automatically switch to internal ring oscillator as the clock source. In this case the ECLK must be available for at least two full cycles after PWRGOOD becomes low.

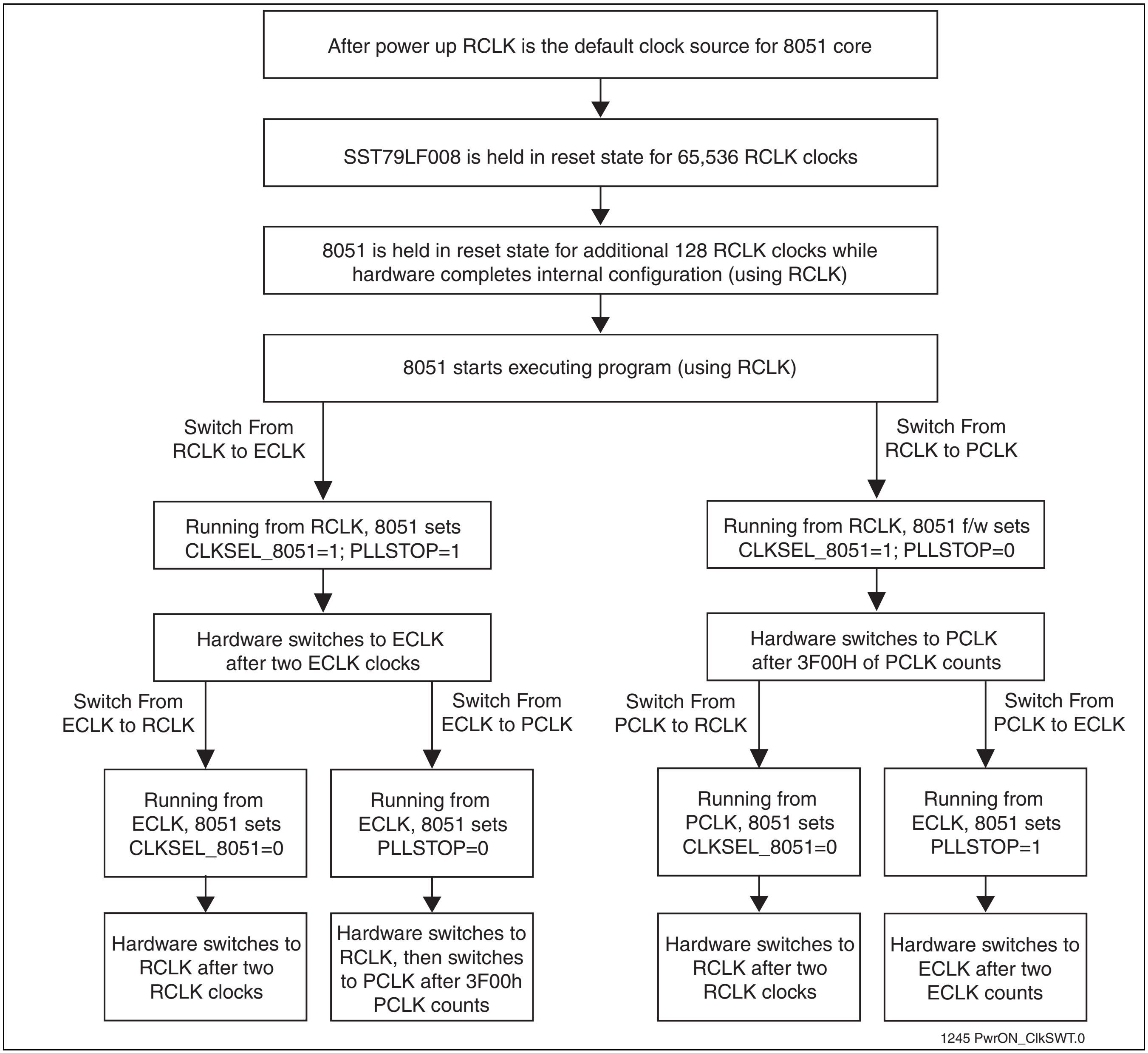

After power up RCLK is the default clock source for 8051 core
SST79LF008 is held in reset state for 65,536 RCLK clocks
8051 is held in reset state for additional 128 RCLK clocks while hardware completes internal configuration (using RCLK)
8051 starts executing program (using RCLK)
Switch From RCLK to ECLK
Switch From RCLK to PCLK
Running from RCLK, 8051 sets CLKSEL_8051=1; PLLSTOP=1
Running from RCLK, 8051 f/w sets CLKSEL_8051=1; PLLSTOP=0
Hardware switches to ECLK after two ECLK clocks
Hardware switches to PCLK after 3F00H of PCLK counts
Switch From ECLK to RCLK
Switch From ECLK to PCLK
Switch From PCLK to RCLK
Switch From PCLK to ECLK
Running from ECLK, 8051 sets CLKSEL_8051=0
Running from ECLK, 8051 sets PLLSTOP=0
Running from PCLK, 8051 sets CLKSEL_8051=0
Running from ECLK, 8051 sets PLLSTOP=1
Hardware switches to RCLK after two RCLK clocks
Hardware switches to RCLK, then switches to PCLK after 3F00h PCLK counts
Hardware switches to RCLK after two RCLK clocks
Hardware switches to ECLK after two ECLK counts
1245 PwrON_ClkSWT.0

FIGURE 5-5: Power-On Sequence and Core Clock Switching

5.3.4 Clock Switching in Low Power Modes

The SST79LF008 device has two low power modes: idle and Power Down mode.

Idle mode is entered by setting IDL bit in PCON register (Section 11.4). In idle mode the internal clock is gated off to the 8051CPU. In this mode, Clock selection controls cannot be changed and all peripheral modules continue to operate normally. Idle mode can be terminated by one of the reset events (see Table 5-2), or by any enabled interrupt (see Table 8-1). In any case the IDL bit is cleared in hardware on exit from idle mode. Clock selection, in the case when idle mode is terminated by a reset event, follows the procedure described in the Section 5.3.3. If Idle mode is terminated by an interrupt, 8051 uses the current clock selection and executes a call to the respective interrupt vector within one machine cycle after interrupt request is asserted. On return from the interrupt, the next instruction executed is the one following the instruction that sets IDL bit.

Power Down mode is entered by setting PD bit in PCON register (Section 11.4). In this mode the clock is stopped for 8051 CPU and most of the peripherals. At this time only modules in Crystal Oscillator and LPC Interface domains may continue to operate (see Table 5-3). After 8051 enters into Power Down mode, hardware stops PLL and Ring Oscillator automatically. Power Down mode can be terminated by one of the reset events (see Table 5-2), or by any enabled wake up event (see Table 8-1). In any case PD bit is cleared in hardware on exit from Power Down mode. Clock selection, in the case when Power Down mode is terminated by a reset event, follows the procedure described in Section 5.3.3. Clock selection, in the case when Power Down mode is terminated by a wake event, is shown on Figure 5-4. After waking up from Power Down mode, the 8051 always uses RCLK as a core clock, and executes a call to the respective wake up interrupt vector within two machines cycles after the wake event. On return from the interrupt, the next instruction executed is the one following the instruction that sets PD bit. Independent of code execution flow, 8051 is then automatically switched by clock selection hardware to another clock source, such as ECLK or PCLK, if the source is stable and was selected by software before Power Down mode entry.

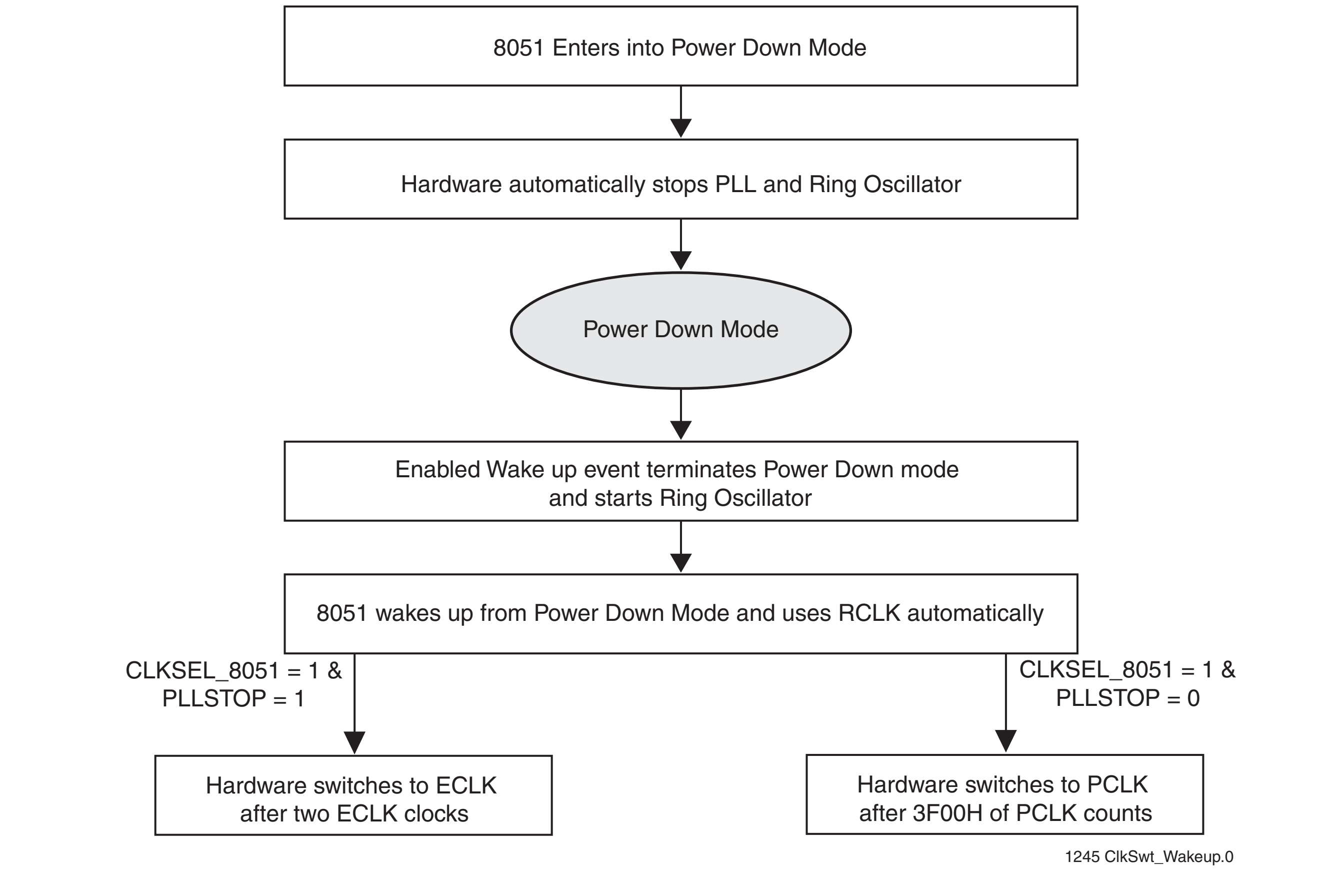

8051 Enters into Power Down Mode
Hardware automatically stops PLL and Ring Oscillator
Power Down Mode
Enabled Wake up event terminates Power Down mode and starts Ring Oscillator
8051 wakes up from Power Down Mode and uses RCLK automatically
CLKSEL_8051 = 1 & PLLSTOP = 1
CLKSEL_8051 = 1 & PLLSTOP = 0
Hardware switches to ECLK after two ECLK clocks
Hardware switches to PCLK after 3F00H of PCLK counts
1245 ClkSwt_Wakeup.0

FIGURE 5-6: Clock Switching after Waking up from Power Down Mode

6.0 8051 EMBEDDED MICROCONTROLLER

The SST79LF008 device includes a high performance 8051 embedded microcontroller unit (MCU) as shown on Figure 1-1.

6.1 8051 MCU Enhancement

The 8051 embedded controller in SST79LF008 provides the following major improvements over the traditional 8051 controller:

- Increased performance: Adopts a selectable 6- or 3- clock machine cycle in lieu of the conventional 12-clock machine cycle, and supports up to 33 MHz operating frequency
- Increased program address space: Up to 128 KByte instead of conventional 64 KByte
- Increased on-chip SRAM: 2 KByte plus standard 256 Byte data memory
- Increased stack address space: Up to 2 KByte instead of conventional 256 Byte stack
- Dual Data Pointers: provide extra programming flexibility for 8051 code development
- Power-saving Operation: Idle and Power Down modes with multiple maskable wake-up sources
- Programmable clock source: internal ring oscillator (RCLK), external input clock (ECLK), or PLL clock (PCLK derived from ECLK, programmable frequency)

6.2 8051 Addressing Modes

The SST79LF008 supports two different addressing modes selectable by the AM1 bit in the ACON register. The 8051 core operates in either the traditional 8051 16-bit address mode (64 KByte address space) or in a 17-bit contiguous address mode (128 KByte address space).

6.2.1 16-Bit Addressing Mode

In 16-bit addressing mode, code and data access is similar to the traditional 8051 memory access; all MCU instructions are opcode compatible and have identical byte count with the conventional 8051 controller. In this mode the MCU can access up to 64 KByte of program and data memory. The SST79LF008 defaults to 16-bit addressing mode following any reset event, which restarts 8051, as specified in Table 5-2.

6.2.2 17-Bit Contiguous Addressing Mode

In 17-bit contiguous addressing mode, code and data access utilizes expanded 17-bit program counter and 17-bit data pointer; several branching and transfer instructions are modified as described below. In this mode the MCU can access up to 128 KByte of program and data memory. Selection of this mode is controlled by AM1 bit in ACON register, See also “6.5.1 Address Control Register (ACON)” .

AM1 = ACON[1] = 1: Select 17-bit Contiguous Addressing Mode

AM1 = ACON[1] = 0: Select 16-bit Addressing Mode (reset value)

6.2.2.1 8051 Instruction Set Modifications

All instruction opcodes are the same for the 16-bit and 17-bit addressing modes. The operand size and encoding is also the same except for the ACALL, AJMP, LCALL and LJMP branch instructions and a MOV DPTR, #data instruction. These unique instructions will require the compiler to generate additional operands in 17-bit addressing mode relative to the conventional 16-bit addressing mode. Therefore 17-bit addressing mode support requires an assembler, compiler and linker to be specifically designed to properly handle the modified length of the above instructions.

The number of machine cycles per instruction may also be different because in 17-bit mode all branch instructions operates over entire 17-bit program counter, and all instructions which utilize the data pointer operates over 17-bit expanded DPTR = DPX[0]+DPH+DPL. However, changes in the number of machine cycles are transparent for the assembler and compiler.

The 8051 instructions modified in 17-bit addressing mode are specified in Table 6-1, which provides encoding, and byte numbers, as well as execution details for each instruction. The instruction assumes that extended 2 KByte stack mode is selected, see Section 6.5. Note, however, that 17-bit contiguous addressing mode can be also used with traditional 8051 256 Byte stack; in this case extended stack pointer ESP:SP in Table 6-1 is replaced by standard stack pointer SP.

Advance Information

TABLE 6-1: 17-bit Addressing Mode-Specific Instructions

	INSTRUCTION CODE	NUMBER	
MNEMONIC	**D7 D6 D5 D4 D3 D2 D1 D0**	**of BYTES**	**OPERATION**
ACALL addr19	a_{18} a_{17} a_{16} 1 0 0 0 1	3	(PC) <- (PC) + 3
	a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8		(ESP:SP) <- (ESP:SP) + 1
	a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		((ESP:SP)) <- (PC7-0)
			(ESP:SP) <- (ESP:SP) + 1
			((ESP:SP)) <- (PC15-8)
			(ESP:SP) <- (ESP:SP) + 1
			((ESP:SP)[0]) <- (PC16)
			(PC16-0) <- addr16-0
AJMP addr19	a_{18} a_{17} a_{16} 0 0 0 0 1	3	(PC) <- (PC) + 3
	a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8		(PC16-0) <- addr116-0
	a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		
INC DPTR	1 0 1 0 0 0 1 1	1	(DPTR) <- (DPTR) + 1
LCALL addr24	0 0 0 1 0 0 1 0	4	(PC) <- (PC) + 4
	a_{23} a_{22} a_{21} a_{20} a_{19} a_{18} a_{17} a_{16}		(ESP:SP) <- (ESP:SP) + 1
	a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8		((ESP:SP)) <- (PC7-0)
	a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		(ESP:SP) <- (ESP:SP) + 1
			((ESP:SP)) <- (PC15-8)
			(ESP:SP) <- (ESP:SP) + 1
			((ESP:SP)[0]) <- (PC16)
			(PC16-0) <- addr16-0
LJMP addr24	0 0 0 0 0 0 1 0	4	(PC16-0) <- addr16-0
	a_{23} a_{22} a_{21} a_{20} a_{19} a_{18} a_{17} a_{16}		
	a_{15} a_{14} a_{13} a_{12} a_{11} a_{10} a_9 a_8		
	a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0		
MOV DPTR, #data24	1 0 0 1 0 0 0 0	4	(DPTR) <- (#data16-0)
	d_{23} d_{22} d_{21} d_{20} d_{19} d_{18} d_{17} d_{16}		DPX[0] <- #data16
	d_{15} d_{14} d_{13} d_{12} d_{11} d_{10} d_9 d_8		DPH <- #data15-8
	d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0		DPL <- #data7-0
RET	0 0 1 0 0 0 1 0	1	(PC16) <- ((ESP:SP)[0])
			(ESP:SP) – 1
			(PC15-8) <- ((ESP:SP))
			(ESP:SP) – 1
			(PC7-0) <- ((ESP:SP))
			(ESP:SP) - 1
RETI	0 0 1 1 0 0 1 0	1	(PC16) <- ((ESP:SP)[0])
			(ESP:SP) – 1
			(PC15-8) <- ((ESP:SP))
			(ESP:SP) - 1
			(PC7-0) <- ((ESP:SP))
			(ESP:SP) - 1

T6-1.1320

6.3 8051 Machine Cycle Control

For any 8051 core clock CCLK source specified in Section 5.3.2, the number of CCLK clocks within each instruction cycle can be selected by clock mode bit in 8051 clock control register.

6.3.1 Clock Control Register (CLKCON)

Location		7	6	5	4	3	2	1	0
SFR	Read	-	-	-	-	SPR2	CLKMD	T1SEL32K	T0SEL32K
AFH	Write								
	Reset	X	X	X	X	0	1	0	0

Symbol	Function
-	Not implemented
X	Not defined
SPR2	SPI clock Rate select bit 2 This bit together with SPR[1:0] bits controls SPI clock rate in master mode (see Section 12.4.1)
CLKMD	8051 clock Mode control bit 1: 6-clock mode (6 clocks in one instruction cycle) 0: 3-clock mode (3 clocks in one instruction cycle)
T1SEL32K	Timer 1 clock Selection bit for event counter function 1: Timer 1 uses the 32.768KHz clock as event counter clock 0: Timer 1 uses external signal on the T1 input pin as event counter clock
T0SEL32K	Timer 0 clock Selection bit 1: Timer 0 uses the 32.768KHz clock as event counter clock 0: Timer 0 uses external signal on the T0 input pin as event counter clock

6.4 8051 Dual Data Pointers

SST79LF008 has two 17-bit data pointers sharing the same addresses in SFR space: both DPL registers share address 82H, both DPH registers share address 83H, and both DPX registers share address 93H. The DPTR selection bit (DPS) in AUXR1 register determines which of the two data pointers is accessed. When DPS = 0, DPTR0 is selected, and when DPS = 1, DPTR1 is selected. Quick switching between the two data pointers can be accomplished by a single INC AUXR1 instruction. When switching between DPTR0 and DPTR1, all three DPX, DPH and DPL registers are switched respectively. The non-selected DPTR registers retain the values they have prior to switch. Refer to Figure 6-1 for illustration of dual data pointer organization.

6.4.1 Auxiliary Register (AUXR1)

Location		7	6	5	4	3	2	1	0
SFR	Read	-	-	-	-	-	-	-	DPS
A2H	Write								
	Reset	X	X	X	X	X	X	X	0

Symbol	Function
-	Not implemented
X	Not defined
DPS	DPTR Registers Selection bit 1: DPTR1 is selected 0: DPTR0 is selected

6.4.2 Data Pointer Low Register (DPL)

Location		7	6	5	4	3	2	1	0
SFR 82H	Read	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DPL[7:0]	Low byte of DPTR

6.4.3 Data Pointer High Register (DPH)

Location		7	6	5	4	3	2	1	0
SFR 83H	Read	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DPH[7:0]	High byte of DPTR

6.4.4 Data Pointer Extended Register (DPX)

Location		7	6	5	4	3	2	1	0
SFR 93H	Read	-	-	-	-	-	-	-	DPX0
	Write								
	Reset	X	X	X	X	X	X	X	0

Symbol	Function
-	Not implemented
X	Not defined
DPX[0]	Most significant bit of DPTR when in 17-bit addressing mode

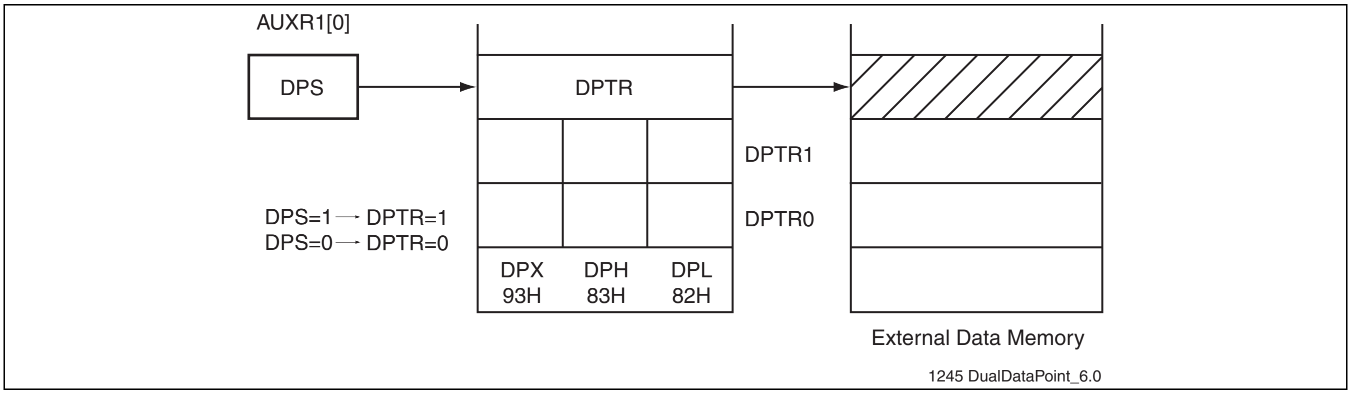


FIGURE 6-1: Dual Data Pointer Organization

6.5 8051 Stack Extension

The conventional 8051 stack is limited to 256Byte internal RAM. The SST79LF008 MCU provides either this conventional stack, or an extended 2 KByte stack (11-bit stack address). When extended stack is enabled by setting the Stack Address (SA) bit in the ACON register, the 2 KByte expanded RAM (XRAM) becomes the memory space used by all instructions that affect the stack. The 11-bit address is formed by concatenating the lower 3 bits of the Extended Stack Pointer (ESP), and the 8-bit Stack Pointer (SP).

When the SA bit is set, any overflow of SP from FFH to 00H will increment the ESP by 1, and any underflow of SP from 00h to FFH will decrement the ESP by 1. When SA = 0, ESP is ignored, but still read/write accessible, and SP is used as a conventional stack pointer.

6.5.1 Address Control Register (ACON)

Location		7	6	5	4	3	2	1	0
SFR	Read	-	-	-	-	-	SA	AM1	-
9DH	Write								
	Reset	X	X	X	X	X	0	0	X

Symbol	**Function**
-	Not implemented
X	Not defined
SA	Extended Stack Address Mode Enable bit 1: All stack instructions will utilize the 11-bit stack pointer ESP:SP formed by concatenating the 3 least significant bits of ESP register with the SP register 0: All stack instructions will utilize the traditional 8-bit 8051 SP register
AM1	Address Mode Control bit. 1: 17-bit Contiguous Addressing Mode (128 KByte 8051 flash area and 896 KByte/7.0Mbit BIOS flash area) 0: 16-bit Addressing Mode (64 KByte 8051 flash area and 960 KByte/7.5Mbit BIOS flash area)

6.5.2 Extended Stack Pointer Register (ESP)

Location		7	6	5	4	3	2	1	0
SFR	Read	-	-	-	-	-	ESP2	ESP1	ESP0
9BH	Write								
	Reset	X	X	X	X	X	0	0	0

Symbol	**Function**
-	Not implemented
X	Not defined
ESP[2:0]	Extended Stack Pointer This register contains the upper 3 bits of the 11-bit extended stack pointer. 11-bit stack pointer allows a stack depth of 2 KBytes. Note that as the stack reaches the top of the 2 KByte XRAM, it will wrap around to XRAM location 0.

6.5.3 Stack Pointer Register (SP)

Location		7	6	5	4	3	2	1	0
SFR	Read	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
81H	Write								
	Reset	0	0	0	0	0	1	1	1

Symbol	**Function**
SP[7:0]	Stack Pointer The stack pointer identifies current location of the stack. The stack pointer is incremented before every push operation (decremented after every pop operation). After reset SP defaults to 07H, and the stack starts at Internal RAM location 07H. Once the 11-bit stack is enabled (SA = 1), this register is combined with the extended stack pointer (ESP) to form the 11-bit address, and the stack will start at XRAM location 07H. Of course, software can relocate the stack to different portion of RAM as desired.

7.0 LPC INTERFACE

The SST79LF008 communicates with the host through the LPC bus. The SST79LF008 LPC interface implementation complies with LPC Interface Specification, Rev. 1.1, and always supports LPC I/O Read/Write cycle types. When LPCMODE bit of LPCMON register is '0', the SST79LF008 responds to Multi-Byte Firmware Memory Read/Write cycles on the LPC bus, and LPC Memory cycles are ignored. When LPCMODE bit of LPCMON register is '1', the SST79LF008 responds to Single-Byte LPC Memory Read/Write cycles on the LPC bus, and LPC Firmware Memory cycles are ignored.

SST79LF008 utilizes all required LPC signals: LAD[3:0], LFRAME#, LRESET#, and LCLK, as well as the following optional LPC signals: SERIRQ, CLKRUN#, and LPCPD# (note that CLKRUN#, and LPCPD# signals share pins with GPIO, and should be properly selected to enable the respective function).

The SST79LF008 flash memory can be read, written, erased and reprogrammed via LPC interface. The flash memory is divided into blocks and sectors that can be erased independently. Flash memory blocks can be protected to prevent accidental modification. All flash commands are interpreted by the device command interface. An on-chip memory controller simplifies the process of programming or erasing the memory by taking care of all of the special operations that are required to update the memory contents. The end of a program or erase operation can be detected by the LPC Host software via flash memory status register.

7.1 LPC Bus Transfer

The SST79LF008 LPC interface implementation complies with LPC Interface Specification, Rev. 1.1, and always supports LPC I/O Read/Write cycle types. When LPCMODE bit of LPCMON register is '0', the SST79LF008 responds to Multi-Byte Firmware Memory Read/Write cycles on the LPC bus, and LPC Memory cycles are ignored. When LPCMODE bit of LPCMON register is '1', the SST79LF008 responds to Single-Byte LPC Memory Read/Write cycles on the LPC bus, and LPC Firmware Memory cycles are ignored. Table 7-1 summarize the size of transfers supported by the SST79LF008.

TABLE 7-1: Transfer Size Supported by the SST79LF008

Cycle Type	Size of Transfer
Firmware Memory Read	1, 2, 4, 16, 128 bytes
Firmware Memory Write	1, 2, 4 bytes
LPC Memory Read	1 byte
LPC Memory Write	1 byte
I/O Read	1 byte
I/O Write	1 byte

T7-1.0 1320

The LPC bus transfer uses four data signals LAD[3:0], one control signal LFRAME#, and LPC clock LCLK. Reset signal LRESET# will put the LPC interface module into a known reset state. The data signals, control signal and clock are designed to be compatible with PCI electrical specifications. The LPC interface operates with a clock speed of 33 MHz.

7.2 LPC Bus Cycles

The start of any LPC cycle is indicated by the LPC Host via active low LFRAME# signal. The START value for LPC cycle determines whether it is Firmware Memory or LPC Memory/LPC I/O cycle — see Table 7-2 (the START value is the LAD[3:0] value latched on the last clock before the host chipset drives LFRAME# signal inactive from low-to-high).

TABLE 7-2: Firmware and LPC Memory Cycles START Field Definition

START Value	Definition
0000	Start of a LPC Memory Read/Write cycle or I/O Read/Write (next field specifies cycle type and direction)
1101	Start of a cycle for Firmware Memory Read
1110	Start of a cycle for Firmware Memory Write

T7-2.0 1320

See the following sections for detailed examples of Firmware Memory, LPC memory, and LPC I/O cycles. See Section 23 for the LPC related configuration options available in SST79LF008 device.

7.2.1 Firmware Memory Cycles

TABLE 7-3: Firmware Memory Read Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	1101	IN	LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized. The START field contents indicate a Firmware Memory Read cycle.
2	IDSEL	0000 or 0001	IN	Indicates which SST79LF008 device should respond. If the IDSEL (ID select) field matches the value specified by ID input pin, then that particular device will respond to the bus cycle.
3-9	MADDR	AAAA	IN	These seven clock cycles make up the 28-bit memory address. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first.
10	MSIZE	YYYY	IN	Indicates transfer size. Device will execute multi-byte read of 2^{YYYY} bytes. SST79LF008 supports only YYYY = 0, 1, 2, 4, 7 (i.e., read 1, 2, 4, 16, 128 bytes).
11	TAR0	1111	IN, then Float	In this clock cycle, the host has driven the bus to all ‘1’s and then floats the bus, prior to the next clock cycle. This is the first part of the bus “turnaround cycle.”
12	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle.
13	RSYNC[2]	0000	OUT	During this clock cycle, the SST79LF008 will generate a “ready-sync” (RSYNC) indicating that the least-significant nibble of the least-significant byte will be available during the next clock cycle.
14-A	DATA	DDDD	OUT	A = $(13+2^{n+1})$; n = MSIZE
				Least significant nibbles outputs first.
(A+1)	TAR0	1111	OUT, then Float	In this clock cycle, the SST79LF008 has driven the bus to all ones and then floats the bus prior to the next clock cycle. This is the first part of the bus “turnaround cycle”. A = $(13+2^{n+1})$; n = MSIZE
(A+2)	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle. A = $(13+2^{n+1})$; n = MSIZE

T7-3.0 1320

1. Field contents are valid on the rising edge of the present clock cycle
2. Between TAR1 and RSYNC cycles SST79LF008 may insert a number of “long- wait-sync” cycles (LWSYNC = 0110b), indicating that data is not ready, yet.

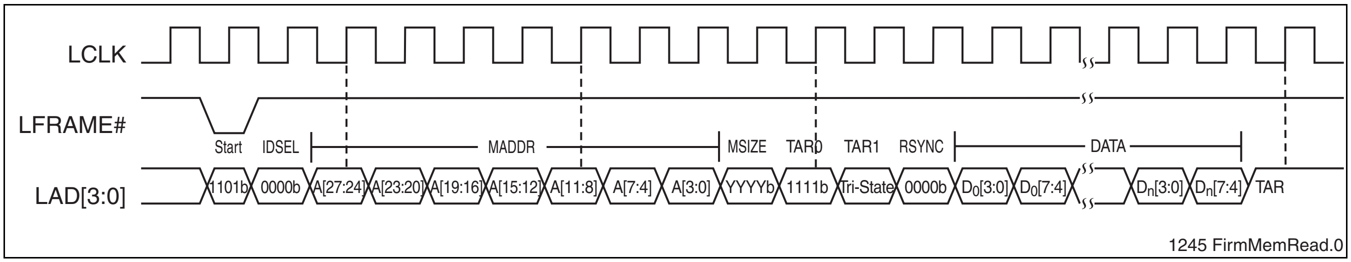


FIGURE 7-1: Firmware Memory Read Waveform

TABLE 7-4: Firmware Memory Write Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	1110	IN	LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized. The START field contents indicate a Firmware Memory Write cycle.
2	IDSEL	0000 or 0001	IN	Indicates which SST79LF008 device should respond. If the IDSEL (ID select) field matches the value specified by ID input pin, then that particular device will respond to the bus cycle.
3-9	MADDR	AAAA	IN	These seven clock cycles make up the 28-bit memory address. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first.
10	MSIZE	YYYY	IN	Indicates transfer size. Device will execute multi-byte write of 2^{YYYY} bytes. SST79LF008 supports only YYYY = 0, 1, 2 (i.e., write 1, 2, 4 bytes.
11-A	DATA	DDDD	IN	A = $(10+2^{n+1})$; n = MSIZE Least significant nibble entered first.
(A+1)	TAR0	1111	IN, then Float	In this clock cycle, the master drives the bus to all ‘1’s, and then floats the bus prior to the next clock cycle. This is the first part of the bus “turnaround cycle.” A = $(10+2^{n+1})$; n = MSIZE
(A+2)	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle. A = $(10+2^{n+1})$; n = MSIZE
(A+3)	RSYNC	0000	OUT	The SST79LF008 outputs the “ready-sync” value 0000b, indicating that it has received data or a flash command. A = $(10+2^{n+1})$; n = MSIZE
(A+4)	TAR0	1111	OUT, then Float	In this clock cycle, the SST79LF008 drives the bus to all ‘1’s, and then floats the bus prior to the next clock cycle. This is the first part of the bus “turnaround cycle”. A = $(10+2^{n+1})$; n = MSIZE
(A+5)	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle. A = $(10+2^{n+1})$; n = MSIZE

T7-4.0 1320

[1] Field contents are valid on the rising edge of the present clock cycle

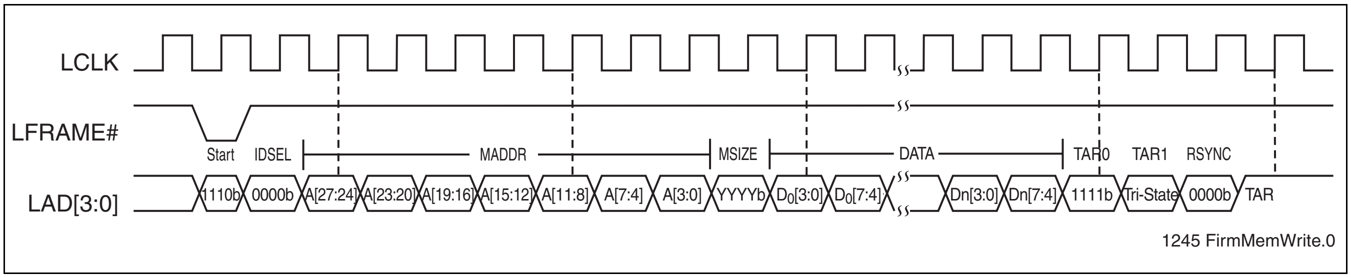


FIGURE 7-2: Firmware Memory Write Waveform

7.2.2 LPC Memory Cycles

TABLE 7-5: LPC Memory Read Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	0000	IN	Indicates start of a cycle. LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized.
2	CYCTYPE + DIR	010xb	IN	Type of cycle and direction of transfer. Field contents indicates an LPC Memory Read cycle.
3-10	ADDR	AAAA	IN	Address Phase for Memory Cycle. LPC protocol supports a 32-bit address phase. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first. The SST79LF008 encodes ID and register space access in the address fields.
11	TAR0	1111	IN, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
12	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle.
13	RSYNC[2]	0000	OUT	The SST79LF008 outputs the “ready-sync” value 0000b indicating that data will be available during the next clock cycle.
14	DATA	$D_3D_2D_1D_0$	OUT	This field is the least-significant nibble of the data byte.
15	DATA	$D_7D_6D_5D_4$	OUT	This field is the most-significant nibble of the data byte.
16	TAR0	1111	OUT, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
17	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle.

T7-5.0 1320

1. Field contents are valid on the rising edge of the present clock cycle
2. Between TAR1 and RSYNC cycles SST79LF008 may insert a number of “long- wait-sync” cycles (LWSYNC = 0110b), indicating that data is not ready, yet.

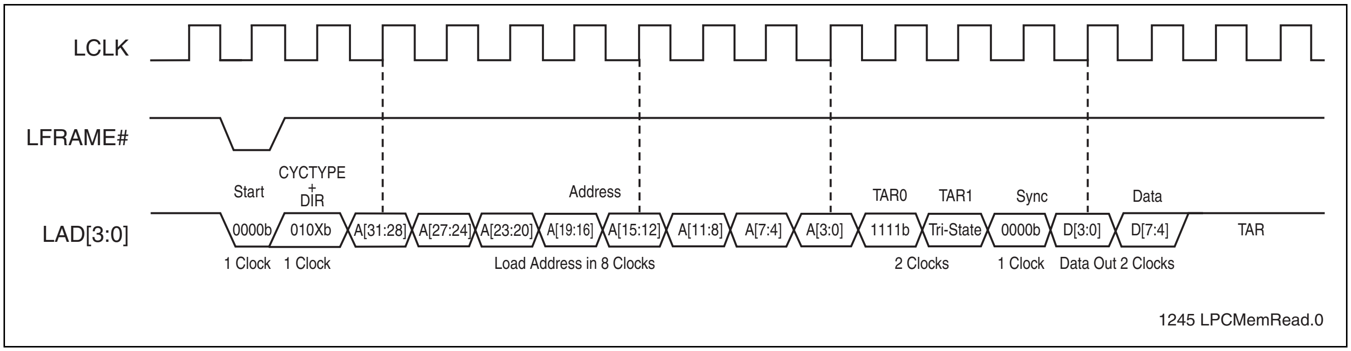


FIGURE 7-3: LPC Memory Read Cycle Waveform

TABLE 7-6: LPC Memory Write Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	0000	IN	Indicates start of a cycle. LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized.
2	CYCTYPE + DIR	011xb	IN	Type of cycle and direction of transfer. Field contents indicates an LPC Memory Write cycle.
3-10	ADDR	AAAA	IN	Address Phase for Memory Cycle. LPC protocol supports a 32-bit address phase. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first. The SST79LF008 encodes ID and register space access in the address fields.
11	DATA	$D_3D_2D_1D_0$	IN	This field is the least-significant nibble of the data byte.
12	DATA	$D_7D_6D_5D_4$	IN	This field is the most-significant nibble of the data byte.
13	TAR0	1111	IN, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
14	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle.
15	RSYNC	0000	OUT	The SST79LF008 outputs the “ready-sync” value 0000b indicating that it has received data or a flash command.
16	TAR0	1111	OUT, then Float	In this clock cycle, the host drives the bus to all ‘1’s and then floats the bus. This is the first part of the bus “turnaround cycle.”
17	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle.

T7-6.0 1320

[1] Field contents are valid on the rising edge of the present clock cycle

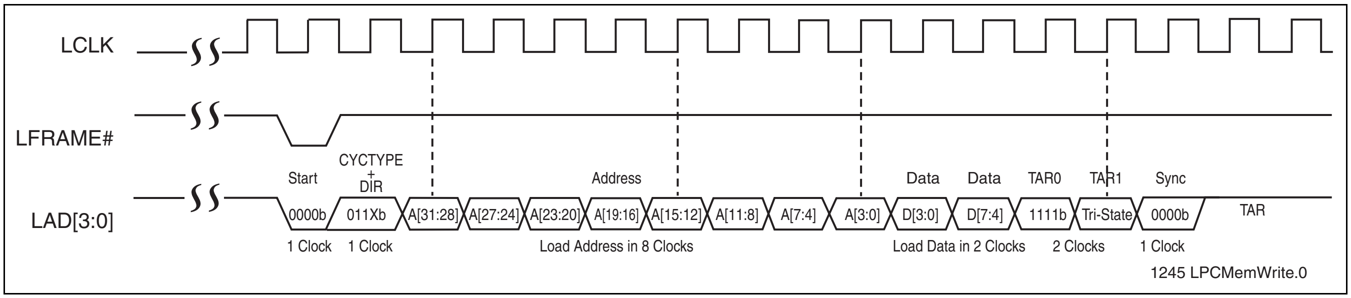


FIGURE 7-4: LPC Memory Write Cycle Waveform

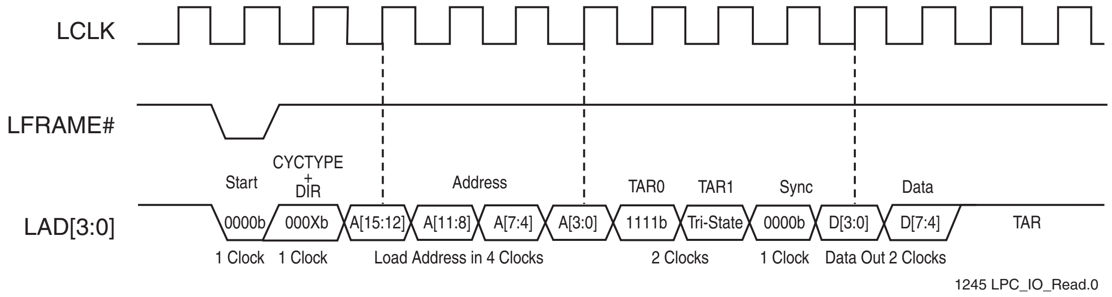
7.2.3 LPC I/O Cycles

TABLE 7-7: LPC I/O Read Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	0000	IN	Indicates start of a cycle. LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized.
2	CYCTYPE + DIR	000xb	IN	Type of cycle and direction of transfer. Field contents indicates an LPC I/O Read cycle.
3-6	ADDR	AAAA	IN	Address Phase for Memory Cycle. LPC protocol supports a 16-bit address phase. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first.
7	TAR0	1111	IN, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
8	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle.
9	RSYNC	0000	OUT	The SST79LF008 outputs the “ready-sync” value 0000b indicating that data will be available during the next clock cycle.
10	DATA	$D_3D_2D_1D_0$	OUT	This field is the least-significant nibble of the data byte.
11	DATA	$D_7D_6D_5D_4$	OUT	This field is the most-significant nibble of the data byte.
12	TAR0	1111	OUT, then Float	In this clock cycle, the host drives the bus to all ‘1’s and then floats the bus. This is the first part of the bus “turnaround cycle.”
13	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle.

T7-7.0 1320

[1] Field contents are valid on the rising edge of the present clock cycle



1245 LPC_IO_Read.0

FIGURE 7-5: LPC I/O Read Cycle Waveform

TABLE 7-8: LPC I/O Write Cycle Field Definitions

Clock Cycle	Field Name	Field Contents[1]	Direction LAD[3:0]	Comments
1	START	0000	IN	Indicates start of a cycle. LFRAME# must be active (low) for the part to respond. Only the last start field (before LFRAME# transitioning high) should be recognized.
2	CYCTYPE + DIR	001xb	IN	Type of cycle and direction of transfer. Field contents indicates an LPC I/O Write cycle.
3-6	ADDR	AAAA	IN	Address Phase for Memory Cycle. LPC protocol supports a 16-bit address phase. AAAA is one nibble of the entire address. Addresses are transferred most-significant nibble first.
7	DATA	$D_3D_2D_1D_0$	IN	This field is the least-significant nibble of the data byte.
8	DATA	$D_7D_6D_5D_4$	IN	This field is the most-significant nibble of the data byte.
9	TAR0	1111	IN, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
10	TAR1	1111 (float)	Float, then OUT	The SST79LF008 takes control of the bus during this cycle.
11	RSYNC	0000	OUT	The SST79LF008 outputs the “ready-sync” value 0000b indicating that it has received data.
12	TAR0	1111	OUT, then Float	In this clock cycle, the host drives the bus to all ‘1’s, and then floats the bus. This is the first part of the bus “turnaround cycle.”
13	TAR1	1111 (float)	Float, then IN	The host resumes control of the bus during this cycle.

T7-8.0 1320

[1] Field contents are valid on the rising edge of the present clock cycle

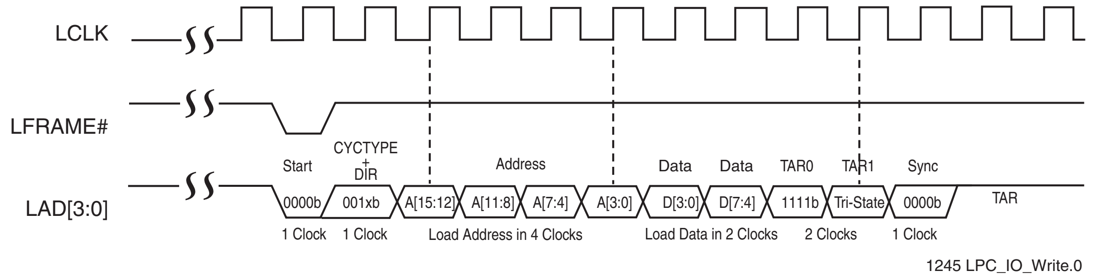


FIGURE 7-6: LPC I/O Write Cycle Waveform

7.3 LPC Flash Command Definitions

All memory write operations are interpreted by the LPC Flash command interface. Commands consist of one or more sequential bus write operations. After power-up or reset, the device enters into the read array mode. For power-up and reset details, see Section 5.2. The commands are summarized in Table 7-9. For detail of each command, refer to the sections below.

TABLE 7-9: LPC Flash Command Definitions[1]

Command	Bus Cycles Required	First Bus Cycle			Second Bus Cycle			Notes
		Oper	Addr	Data	Oper	Addr	Data	
Read Array/Reset	1	Write	X	FFH				
Read Device ID Read Unique ID	≥ 2	Write	X	90H	Read	IA	ID	2,3
Read Status Register	2	Write	X	70H	Read	X	SRD	3
Clear Status Register	1	Write	X	50H				
Sector Erase	2	Write	X	30H	Write	SA	D0H	4
Block Erase	2	Write	X	20H	Write	BA	D0H	4
Program	2	Write	X	40H or 10H	Write	WA	WD	4
Erase Suspend	1	Write	X	B0H				
Erase Resume	1	Write	X	D0H				
User Unique ID Program	2	Write	X	A5H	Write	WA	WD	5
User Unique ID Program Lockout	2	Write	X	85H	Write	X	00H	
Enter UNVR(3K OTP) / Enter ENVR	2	Write	X	60H	Write	X	76H	6
Force LPC Soft Reset	2	Write	X	65H	Write	X	8AH	7
Release LPC Soft Reset	2	Write	X	6AH	Write	X	79H	7

T7-9.0 1320

1. X = Any valid address within the device main flash memory array address space (FFF0 0000H to FFFF FFFFH).
   IA = Device Identification (ID) address/Unique ID address
   SA/BA = Address of sector or block being erased (any valid address within the respective sector/block)
   WA = Address of memory location to be written to
   SRD = Data read from Status Register
   WD = Data written to address WA
   ID = Data read from Device identifier codes or Unique ID
2. Read operations, following the Read Device/Unique ID command, access either Manufacturer ID, or Device ID, or Unique ID. Valid Manufacturer and Device ID addresses are FFFC 0000H and FFFC 0001H, respectively.
   Valid address range for SST Pre-Programmed 8-byte Factory Unique ID is from FFFC 0180H to FFFC 0187H.
   Valid address range for User Programmable 24-byte Unique ID is from FFFC 0188H to FFFC 019FH.
3. Subsequent reads continue to return ID or Status data until another valid command is issued.
4. The sector or block to be erased or programmed must not be write-locked, otherwise the operation will fail.
5. Valid User Programmable Unique ID addresses are from FFFC 0188H to FFFC 019FH.
6. After enter ENVR/UNVR command is executed
   Valid address for 4K ENVR area is from FFF0 0000H to FFF0 0FFFH.
   Valid address for 3K OTP UNVR area is from FFF0 1000H to FFF0 1BFFH.
7. These 2 commands are not standard flash memory control commands. Do not use these commands unless the KBC firmware and BIOS are corrupted.

Two LPC write cycles within 2-cycle command must be consecutive. The command sequence has to be restarted from the first cycle, if the second cycle data is incorrect, the second cycle is a read from memory array, or the second cycle is a read/write access to LPC memory mapped registers. For more information on LPC memory mapped registers, see Section 7.6. However, if the second cycle is aborted or contains invalid fields, it will be ignored, and the command sequence does not need to be restarted. See Section 7.4 for LPC abort mechanisms and invalid fields.

All address ranges defined for LPC flash commands are specified as 32-bit system memory addresses, and are valid when SST79LF008 is used as a boot firmware memory device (low level is applied to ID input pin). When the SST79LF008 is not a boot device, the respective addresses will be changed according to Multiple Device Selection mechanism described in Section 7.5.

7.3.1 Read Array Command

After Power-On, Brown-Out, External, WDT, or aLPC Soft Reset, the device defaults to the read array mode. The read operation can also be initiated by issuing the Read Array/Reset Command. The device remains available for main flash memory array reads until another command is written.

Once an internal Program/Erase operation starts, the device will not recognize the Read Array/Reset command until the operation is completed, unless the erase operation is suspended via an Erase Suspend command as described in Section 7.3.7.

7.3.2 Read Device Identifier Command

The Read ID operation is initiated by writing the Read Device ID command. Following the write of this command, the device outputs the manufacturer and device ID data from the addresses shown in Table 7-10. Write any other valid command to the device to terminate the Read ID operation.

.
TABLE 7-10: Product Identification

	Address	Data
Manufacturer's ID (SST)	FFFC 0000H	BFH
Device ID (SST79LF008)	FFFC 0001H	F0H

T7-10.0 1320

7.3.3 Read Status Register Command

The Status register provides information on the current or previous Program or Erase operation. The Status register may be read to determine when a program or sector/block erase command completes, and whether the operation completed successfully. The Status register may be read at any time by issuing the Read Status Register Command.

After writing this command, all subsequent read operations within the device main flash memory array will return the data from the status register until another valid command is written. The Status Register bits are summarized in Register 7.3.3.1.

7.3.3.1 Flash Memory Status Register

	7	6	5	4	3	2	1	0
Name	WSMS	ESS	-	-	-	-	BPS	-
Reset	1	0	0	0	0	0	0	0

Bit	Description
7	Write State Machine Status Check WSMS to determine erase or program completion. 1 = Ready 0 = Busy
6	Erase Suspend Status 1 = Erase Suspended 0 = Erase in progress/completed
1	Block Protect Status The Block Write-Lock bit is interrogated only after erase or program command is issued. It informs the system whether or not the selected block is locked. This bit does not provide a continuous indication of write-lock bit value. 1 = Write-lock bit is set (operation aborted) 0 = Block is unlocked
5:2, 0	Reserved for future use

T7-10.0 1320

7.3.4 Clear Status Register Command

The Clear Status Register command can be used to reset the BPS bit in the Status register to '0'. This bit does not automatically return to '0' when a new Program or Erase command is issued. Therefore, it should be cleared by issuing the Clear Status Register Command before attempting a new Program or Erase command. Device Power-On, Brown-Out, External, WDT, or aLPC Soft Reset will return Status Register to its reset value, and clear BPS to '0'.

7.3.5 Sector Erase Command and Block Erase Command

The Erase Command operates on one sector or block at a time. This command requires an arbitrary address within the targeted sector or block (SA or BA) to be specified in the second bus cycle. Note that a Sector/Block Erase operation changes all Sector/Block byte data to FFH. If a read operation is performed within the main flash memory array after issuing the erase command, the device will automatically output Status Register data. The system can poll the Status Register in order to verify the completion of the Sector/Block Erase operation. If a Sector/Block Erase is attempted on a locked block, the operation will fail and the data in the Sector/Block will not be changed. In this case, the Status Register will report the error (BPS = 1). During the Block Erase or Sector Erase operation, the device will only accept the Read Status Register or the Erase Suspend commands. All other commands will be ignored until the operation is completed.

7.3.6 Program Command

The Program Command writes data (WD) specified in the second bus cycle to the consecutive flash memory locations starting with the specified address (WA). The data size can be specified as 1, 2, or 4 bytes for Firmware Memory cycles, and 1 byte only for LPC memory cycles. After issuing Program command the device automatically outputs the Status Register data when read within the main flash memory array. The system can poll the Status Register in order to verify the completion of the Program operation. If a Program operation is attempted on a locked block, the operation will fail and the data in the addressed byte will not be changed. In this case, the Status Register will report the error (BPS = 1). During the Program operation, the device will only accept the Read Status Register command. All other commands will be ignored until the operation is completed.

7.3.7 Erase Suspend Command and Erase Resume Commands

The Erase Suspend command allows Sector or Block Erase interruption in order to read or program data in another block of memory. Once the Erase Suspend command is executed, the device will suspend any in-progress Erase operation within time T_{ES}. See Table 24-8. The device outputs status register data, when read within the main flash memory array, after the Erase Suspend command is written. After erase operation is actually suspended, the device will set the Status Register bit ESS = 1. Thus, the system can determine whether the erase operation has been suspended (WSMS = 1 and ESS = 1) or completed (WSMS = 1 and ES = 0) by polling the Status Register.

After a successful suspend, a Read Memory Array command may be issued to read data from a block other than the suspended block. A Program command may also be issued while Erase is suspended to program data in memory locations other than the sector or block currently in the Erase Suspend mode. If a Read Array command is written to an address within the suspended Sector/Block, this may result in reading invalid data. If a Program command is written to an address within the suspended Sector/Block, the command is acknowledged but ignored. Other valid commands while an erase is suspended include Read Status Register, Read Device ID, and Erase Resume.

The Erase Resume command resumes the erase operation in the suspended sector or block. After the Erase Resume command is written, the device will continue the erase operation. Erase cannot resume until any program operations initiated during erase suspend have completed. Suspended operations cannot be nested. That is, the system needs to complete/resume any previously suspended operation before a new operation can be suspended. Once the Erase Resume command is issued, the subsequent bus read operations within the main flash memory array read the status register.

7.3.8 User Unique ID Read, Program and Lockout Commands

The 256-bits (32 bytes) of the SST79LF008 Unique ID space are divided into two segments. One 64-bit segment is programmed at SST with a unique 64-bit number, which is unchangeable. The other 192-bit segment is a one time programmable segment (OTP) which is left blank for customers to program as desired. Once the customer segment is programmed, it can be locked to prevent reprogramming. Note that regardless of whether or not the Unique ID is locked, neither of the Unique ID segments can be erased.

In order to read the Unique ID information, the user can issue a Read Unique ID command to the device. At this point the device enters the Read Device/Unique ID mode. The Unique ID information can be read at the following memory addresses (IA):

- FFFC 0180H to FFFC 0187H – SST Factory Pre-Programmed ID (8 bytes – 64 bit).
- FFFC 0188H to FFFC 019FH – User Programmed Unique ID (24 bytes – 192bit).

A Read Array/Reset command must then be issued to the device in order to exit the “Read Device/Unique ID” mode and return to read array mode.

An alternative method to read the Unique ID information without switching from read array mode is to read the respective registers located in the firmware flash memory register space described in Section 7.6. In this case the Unique ID information can be retrieved in read array mode at the following register addresses:

- FFBC 0180H to FFBC 0187H – SST Pre-Programmed Device ID Segment (8 bytes – 64bit).
- FFBC 0188H to FFBC 019FH – User Programmed Unique ID Segment (24bytes – 192bit).

In order to Program the Unique ID, a Program Unique ID command should be issued with the address (WA) in the range of FFFC 0188H to FFFC 019FH. Processing of this command is similar to the main flash array program command described above. In order to protect Unique ID from corruption, Unique ID Program Lockout command should be used.

7.3.9 Enter UNVR (3K OTP) / Enter ENVR Commands

The Enter UNVR/ENVR access mode command is used to access 3 KByte OTP UNVR and 4 KByte flash ENVR. Once the Enter UNVR/Enter ENVR command is issued, the LPC Host can read ENVR information at addresses FFF0 0000H to FFF0 0FFFH in LPC address space and read UNVR at addresses FFF0 1000H to FFF0 1BFFH in LPC address space. The LPC Host can also erase/program ENVR as well as program UNVR using the same Sector Erase and Program commands as for the main flash array.

A Read Array/Reset command must be issued to the device in order to exit the UNVR/ENVR access mode and return to read main flash array mode.

7.3.10 Force / Release LPC Soft Reset Commands

The Force LPC Soft Reset command is used to put 8051 into LPC Soft Reset state and unconditionally release the flash memory bus to the LPC Host. This command aborts KBC operation, and it is not recommended, unless KBC firmware and BIOS code are both corrupted.

The Release LPC Soft Reset command is used to restart 8051 code execution and KBC operations after LPC Soft reset.

Note. In addition to Release command, the LPC Soft Reset state is also terminated by Power-On, Brown-Out, External, WDT or aLPC Soft Reset, as well as by LPC Interface Reset (LRESET#) and LPC Power Down (LPCPD#) signals.

7.4 LPC Abort Mechanism and Invalid Fields

If LFRAME# is driven low for one or more clock cycles after the start of a bus cycle, the cycle will be terminated. The host may drive the LAD[3:0] with 1111b (ABORT nibble) to return the interface to ready mode. The ABORT only affects the current bus cycle. For a multi-cycle command sequence, such as the Erase or Program commands, ABORT doesn’t interrupt the entire command sequence, only the current bus cycle of the command sequence. The host can re-send the bus cycle for the aborted command and continue the command sequence after the device is ready again.

During an on-going LPC bus cycle, the SST79LF008 will not explicitly indicate that it has received invalid field sequences. The response to specific invalid fields or sequences is explained in Sections 7.4.1 and 7.4.2.

7.4.1 Response to Invalid Fields for Firmware Memory Cycle

ID mismatch: The SST79LF008 compares ID bits in the IDSEL field with the ID value specified by the SST79LF008 input ID pin. If there is a mismatch, the device will ignore the cycle. See Multiple Device Selection, Section 7.5 for details.

Address out of range: The address sequence is 7 fields long (28 bits) for Firmware Memory bus cycles. Only address bits A_0 to A_{19} and A_{22} are decoded by the SST79LF008. Address A_{22} has the special function of directing reads and writes to the flash core ($A_{22} = 1$) or to the register space ($A_{22} = 0$).

Invalid MSIZE field: If the SST79LF008 receives an invalid size field during a Firmware Memory Read or Write operation, the device will ignore the cycle and no operation will be attempted. The device will not generate any kind of response in this situation. The SST79LF008 will only respond to MSIZE values listed in the Table 7-11.

TABLE 7-11: Valid MSIZE Field for Firmware Memory Cycle

Bits	Direction	Size of transfer
0000	R/W	1 Byte
0001	R/W	2 Byte
0010	R/W	4 Byte
0100	R	16 Byte
0111	R	128 Byte

T7-11.0 1320

Once valid START, IDSEL, and MSIZE are received, the SST79LF008 will always complete the bus cycle. However, if the device is busy performing a flash Erase or Program operation, no new internal memory write will be executed. As long as the states of LAD[3:0] and LFRAME# are known, the response of the SST79LF008 to signals received during the cycle is predictable.

Non-boundary-aligned address: The SST79LF008 accepts Multi-Byte transfers for both Read and Write operations. The device address space is thus divided into pages of uniform size 2, 4, 16, or 128 Byte wide, according to the MSIZE value. The host issues only one address in the MADDR field of the Firmware Memory Cycle but multiple bytes are read from, or written to, the device. For this reason the MADDR address should be page "boundary aligned".

Boundary aligned means that for a 2 Byte transfer the address should be aligned to a Word boundary ($A_0 = 0$), for a 4 Byte transfer the address should be aligned to a Double Word boundary ($A_0 = 0$, $A_1 = 0$), etc. If the address supplied by the host is not page "boundary aligned", the SST79LF008 will force a boundary alignment, starting the Multi-Byte Read or Write operation from the lower Byte of the addressed page.

7.4.2 Response to Invalid Fields for LPC Memory Cycle

ID mismatch: The SST79LF008 interprets address bits [A_{24}:A_{23}, A_{21}:A_{20}] as ID information and compares them with the complement of ID value specified by the SST79LF008 input ID pin. If there is a mismatch, the device will ignore the cycle. See Multiple Device Selection, Section 7.5, for details.

Address out of range: The address sequence is 8 fields long (32 bits). The address bits [A_{24}:A_{23}, A_{21}:A_{20}] for the SST79LF008 are used to select the device with proper IDs. The most significant address bits [A_{31}:A_{25}] must be "1's" for LPC memory cycle to be completed. Address A_{22} has the special function of directing reads and writes to the flash core (A_{22} = 1) or to the register space (A_{22} = 0).

For the boot device (with LPC protocol ID = 0), the SST79LF008 also decodes the physical addresses of the top 128 KByte blocks at two system memory ranges:

- FFFF FFFFH to FFFE 0000H—top of 4 GByte address space
- 000F FFFFH to 000E 0000H—top of legacy 1 MByte address space

Once valid START, CYCTYPE + DIR, and address range, including ID bits, are received, the SST79LF008 will always complete the bus cycle. However, if the device is busy performing a flash Erase or Program operation, no new internal memory write will be executed. As long as the states of LAD[3:0] and LFRAME# are known, the response of the SST79LF008 to signals received during the LPC cycle is predictable.

7.5 Multiple Device Selection

Multiple LPC firmware memory devices may be strapped to increase memory densities in a system. BIOS support, bus loading, or the attaching bridge may limit the number of connected devices. The boot device must respond to LPC protocol with ID of 0 (0000b); subsequent devices use incremental numbering. Equal density must be used with multiple devices.

With one ID input pin SST79LF008 can have two different LPC protocol ID values. Respectively, SST79LF008 flash memory array will be mapped into two different address ranges in the 4GByte system memory space depending on ID pin logic level. When V_{IL} level is applied to ID input pin, the valid LPC protocol ID is 0 (0000b). When V_{IH} level is applied to ID input pin, the valid LPC protocol ID is 1 (0001b).

7.5.1 Multiple Device Selection for Firmware Memory Cycle

For Firmware Memory Read/Write cycles, LPC protocol ID information is included into IDSEL field of every cycle. The ID value specified by SST79LF008 ID input pin (0000b or 0001b) must match the value in IDSEL field. If there is a mismatch the SST79LF008 will ignore the respective LPC Firmware Memory cycle.

7.5.2 Multiple Device Selection for LPC Memory Cycle

For LPC Memory Read/Write cycles, LPC protocol ID information is included in the address bits of every cycle. The address bits [A_{24}:A_{23}, A_{21}:A_{20}] are used to select the device with proper IDs. The ID bits in the address field must match the inverse of the ID value specified by SST79LF008 ID input pin (i.e., address bits should be 1111b when V_{IL} level is applied to ID input pin or 1110b when V_{IH} level is applied to ID input pin). If there is a mismatch the SST79LF008 will ignore the respective LPC Memory cycle.

7.6 LPC Memory Mapped Registers.

The LPC memory mapped registers can be accessed by LPC Firmware Memory cycles as well as by LPC Memory cycles with address bit A_{22} = 0. Four types of registers are implemented in SST79LF008: Block Locking registers, JEDEC ID Registers, Multi-byte Read/Write Configuration Registers, and Unique ID Registers. These registers appear at their respective addresses in the 4GByte system memory address space as specified in Table 7-12 for the boot device. They will appear elsewhere if SST79LF008 is not the boot device according to Multiple Device Selection mechanism described in Section 7.5. Read access to unused register locations will return 00H. Write access to these locations has no effect. Attempts to read or write any register during internal Program/Erase operation are completed normally.

7.6.1 Flash Memory Block Locking Registers

SST79LF008 provides software controlled lock protection through a set of Block Locking registers. These registers are read/write accessible via standard memory locations specified in Table 7-12.

TABLE 7-12: Block Locking Registers

Register	Protected Memory		Memory Map Register Address for Boot Device	Reset Value[1]	Access
	Size	Address Range			
T_BLOCK_LK	16K	0FFFFFH-0FC000H	FFBFC002H	01H	R/W
T_MINUS01_LK	8K	0FBFFFH-0FA000H	FFBFA002H	01H	R/W
T_MINUS02_LK	8K	0F9FFFH-0F8000H	FFBF8002H	01H	R/W
T_MINUS03_LK	32K	0F7FFFH-0F0000H	FFBF0002H	01H	R/W
T_MINUS04_LK	64K	0EFFFFH-0E0000H	FFBE0002H	01H	R/W
T_MINUS05_LK	64K	0DFFFFH-0D0000H	FFBD0002H	01H	R/W
T_MINUS06_LK	64K	0CFFFFH-0C0000H	FFBC0002H	01H	R/W
T_MINUS07_LK	64K	0BFFFFH-0B0000H	FFBB0002H	01H	R/W
T_MINUS08_LK	64K	0AFFFFH-0A0000H	FFBA0002H	01H	R/W
T_MINUS09_LK	64K	09FFFFH-090000H	FFB90002H	01H	R/W
T_MINUS10_LK	64K	08FFFFH-080000H	FFB80002H	01H	R/W
T_MINUS11_LK	64K	07FFFFH-070000H	FFB70002H	01H	R/W
T_MINUS12_LK	64K	06FFFFH-060000H	FFB60002H	01H	R/W
T_MINUS13_LK	64K	05FFFFH-050000H	FFB50002H	01H	R/W
T_MINUS14_LK	64K	04FFFFH-040000H	FFB40002H	01H	R/W
T_MINUS15_LK	64K	03FFFFH-030000H	FFB30002H	01H	R/W
T_MINUS16_LK	64K	02FFFFH-020000H	FFB20002H	01H	R/W
T_MINUS17_LK	64K	01FFFFH-010000H	FFB10002H	01H	R/W
T_MINUS18_LK	64K	00FFFFH-000000H	FFB00002H	01H	R/W

T7-12.0 1320

1. All block locking registers returned to their reset values specified above after any one of the following reset events: Power-On Reset, External Reset, Watchdog Timer Reset, Brown-Out Reset, aLPC Soft Reset, or external LPC Interface Reset (see also Section 5.2.)

In case of multi-byte register reads with Firmware Memory cycle, the device will return register data for the addressed register until the command finishes or is aborted.

Bit definitions for block locking registers are specified in Table 7-13.

TABLE 7-13: Block Locking Register Bits

Reserved Bit [7:3]	Read-Lock Bit [2][1]	Lock-Down Bit [1] [2]	Write-Lock Bit [0] [3]	Lock Status
00000	0	0	0	Full Access
00000	0	0	1	Write Locked (Default State after reset)
00000	0	1	0	Locked Open (Full Access Locked Down)
00000	0	1	1	Write Locked Down
00000	1	0	0	Block Read Locked (Registers alterable)
00000	1	0	1	Block Read & Write Lock (Registers not alterable)
00000	1	1	0	Block Read Locked Down (Register not alterable)
00000	1	1	1	Block Read & Write lock Down (Register not alterable)

T7-13.0 1320

1. Read Lock: The Read-Lock bit, bit 2, controls the read access. The default read status of all blocks after reset is read-unlocked. When a block's read lock bit is set, data cannot be read from that block. An attempted read from a read-locked block will result in the data 00h. The read lock status can be unlocked by clearing the read lock bit, provided that the block is not locked down. The current read lock status of a particular block can be determined by reading the corresponding read-lock bit.
2. Lock Down. The Lock-Down bit, bit 1, controls the Block Locking register. The default Lock Down status of all blocks after reset is not locked down. Once the Lock-Down bit is set, any future attempted changes to that Block Locking register will be ignored. The Lock-Down bit is only cleared upon a device reset. Current Lock Down status of a particular block can be determined by reading the corresponding Lock-Down bit. Once a block's Lock-Down bit is set, the Read- and Write-Lock bits for that block can no longer be modified, and the block is locked down in its current state of read/write accessibility.
3. Write-Lock: The Write-Lock bit, bit 0, controls the Program/Erase lock state. The default Write status of all blocks after reset is write locked. When bit 0 of the Block Locking register is set, Program and Erase operations for the corresponding block is prevented. Clearing the Write-Lock bit will unprotect the block. The Write-Lock bit must be cleared prior to starting a Program or Erase operation because it is sampled at the beginning of the operation.

Note: The registers (T_BLOCK_LK, T_MINUS01_LK, T_MINUS02_LK, and T_MINUS03_LK) protect memory areas within one 64 KByte flash memory Block15 (see Figure 4-1). Therefore, when any of these memory areas are write-protected the Block Erase command for Block15 is not accepted.

7.6.2 JEDEC ID Registers

The JEDEC ID registers are read-only registers and are accessible via memory locations specified in Table 7-14. In case of multi-byte register reads with Firmware Memory cycle, the device will return register data for the addressed register until the command finishes or is aborted.

TABLE 7-14: JEDEC ID Registers

Register	Register Address for Boot Device	Value	Access
Manufacturer ID	FFBC0000H	BFH	R
Device ID (SST79LF008)	FFBC0001H	F0H	R

T7-14.0 1320

7.6.3 Multi-byte Read/Write Configuration Registers

The multi-byte read/write configuration (MBR) registers are four 8-bit read-only registers located at addresses specified in Table 7-15. These registers are accessible using Firmware Memory Read cycle only. The device will return unused register space data (00H) if these registers are addressed via LPC memory read cycles. These registers contain information about multi-byte read and write access sizes that will be accepted for Firmware Memory multi-byte commands. In case of multi-byte register reads, device will return register data for addressed register until the command finishes or is aborted.

TABLE 7-15: Multi-byte Read/Write Configuration registers

Register	Register Address for Boot Device	Value	Access	Description
MULTI_BYTE_READ_L	FFBC0005H	0100 1011b	R	Device supports 1,2,4, 16, 128 byte reads
MULTI_BYTE_READ_H	FFBC0006H	0000 0000b	R	Future Expansion for read
MULTI_BYTE_WRITE_L	FFBC0007H	0000 0011b	R	Device supports 2,4 byte write
MULTI_BYTE_WRITE_H	FFBC0008H	0000 0000b	R	Future Expansion for write

T7-15.0 1320

7.6.4 Unique ID Registers

In addition to Read Unique ID command described in Section 7.3, the SST79LF008 allows the LPC Host to read Unique ID Information and its Write Lock/Unlock status via LPC memory mapped register space at addresses defined in Table 7-16. In case of multi-byte register reads with Firmware Memory cycle, for the all UID_BYTE registers, the device will return page aligned sequential register data with wrap-around until the command finishes or is aborted. Multi-byte read of UID_WRITE_LOCK register will return register data for the addressed register until the command finishes or is aborted.

All Unique ID registers are read-only registers. The Unique ID Program and Lockout commands shown in Table 7-16 can be used to write (program) and lock Unique ID.

TABLE 7-16: Unique ID Registers

Register	Register Address	Value	Access	Description
UID_WRITE_LOCK	FFBC017FH	0000 0000b 0000 0001b	R	Write Unlocked Write Locked
UID_BYTE_0	FFBC0180H		R	Factory Programmed
UID_BYTE_1	FFBC0181H		R	Factory Programmed
UID_BYTE_2	FFBC0182H		R	Factory Programmed
UID_BYTE_3	FFBC0183H		R	Factory Programmed
UID_BYTE_4	FFBC0184H		R	Factory Programmed
UID_BYTE_5	FFBC0185H		R	Factory Programmed
UID_BYTE_6	FFBC0186H		R	Factory Programmed
UID_BYTE_7	FFBC0187H		R	Factory Programmed
UID_BYTE_8	FFBC0188H		R	User Programmed
UID_BYTE_9	FFBC0189H		R	User Programmed
…	…		…	…
UID_BYTE_30	FFBC019EH		R	User Programmed
UID_BYTE_31	FFBC019FH		R	User Programmed

T7-16.0 1320

7.7 PCI CLOCK RUN CONTROL SUPPORT

The SST79LF008 supports the CLKRUN# input/open drain output signal according to the PCI Mobile Design Guide Rev 1.0 specification. This signal is used by the system to indicate the LPC clock status. When CLKRUN# is "high", the LPC clock is, or is about to be, stopped. When CLKRUN# is "low", the LPC clock is running.

The CLKRUN# is used by SST79LF008 to request restarting the stopped clock in order to report the Serialized IRQs status changes. If any module within the SST79LF008 asserts or de-asserts serialized interrupt to the system and the CLKRUN# input is "high", the SST79LF008 can request the restoration of the clock by the assertion of the CLKRUN# signal low. The SST79LF008 drives CLKRUN#

low until it detects two rising edges of the restarted LPC clock. After the second clock edge, the SST79LF008 disables its CLKRUN# open drain driver.

The SST79LF008 would not assert CLKRUN# if it is already driven low by the central resource or any other device on the bus. Also, the SST79LF008 would not assert CLKRUN# unless the line has been de-asserted for two successive clocks; for example, before the clock was stopped. Additionally, CLKRUN# is asserted if 8051 core tries to access any device 0, 1, or 3 configuration register and the device configuration register index 30H value is 00H (inactive). See Table 23-1 for Configuration Registers Map.

Refer to the PCI Mobile Design Guide Rev 1.0 for a detailed description of the CLKRUN# function.

7.8 LPC Power Down Protocol Support

The SST79LF008 supports the LPCPD# input signal. This signal is asserted by the system prior to going to low-power state. After LPCPD# is activated (with at least 30 microseconds delay) the LPC clock is stopped low, and the other host LPC I/F output signals being tri-stated or driven low.

Upon recognizing that LPCPD# is asserted, there will be no further transactions on the LPC interface. While LPCPD# is asserted, the SST79LF008 continues to drive IRQ1 and IRQ12 frames. After LPCPD# is de-asserted, the LPC interface may be reset depending on the characteristics of the system reset signal connected to LRESET# pin. The SST79LF008, however, resets internal LPC protocol state machine on exit from LPC low power state without LRESET# going active.

The SST79LF008 asynchronously recognizes LPCPD# state changes from active to inactive and vice versa. (Note that LPCPD# signal, may not meet setup times to LCLK, however, it can be sampled with LCLK, since the clock is running for at least 30 microseconds after LPCPD# goes low, and for more than 30 microseconds prior to LPCPD# going high.)

The state of the LPCPD# signal can be directly read via LPCMON register (see Section 23.1); LPCPD# signal transition generates interrupt request to 8051 core via WSRCG register (see Section 8.3).

Refer to the LPC Interface Specification, Rev 1.1 for a detailed description of the LPCPD# function.

8.0 INTERRUPTS AND WAKEUPS

8.1 SST79LF008 Interrupts

SST79LF008 has eleven interrupt sources under a four-level priority scheme. Table 8-1 and Figures 8-1, 8-2, 8-3, 8-4, and 8-5 summarize the supported interrupt structure. Interrupt 0 (INT0) is dedicated for matrix keyboard event. Interrupt 1 (INT1) combines interrupt source registers A and B. Interrupt 2 (INT2) combines wakeup event source registers A and B. Interrupt 3 (INT3) combines wakeup event source registers C,D, and E. Interrupt 4 (INT4) combines wakeup event source registers F,G, H and I. Interrupt 5 (INT5) combines wakeup event source registers J, K, L, M, N, O and P. For details on INT0-INT5 interrupt sources see Section 8.3.

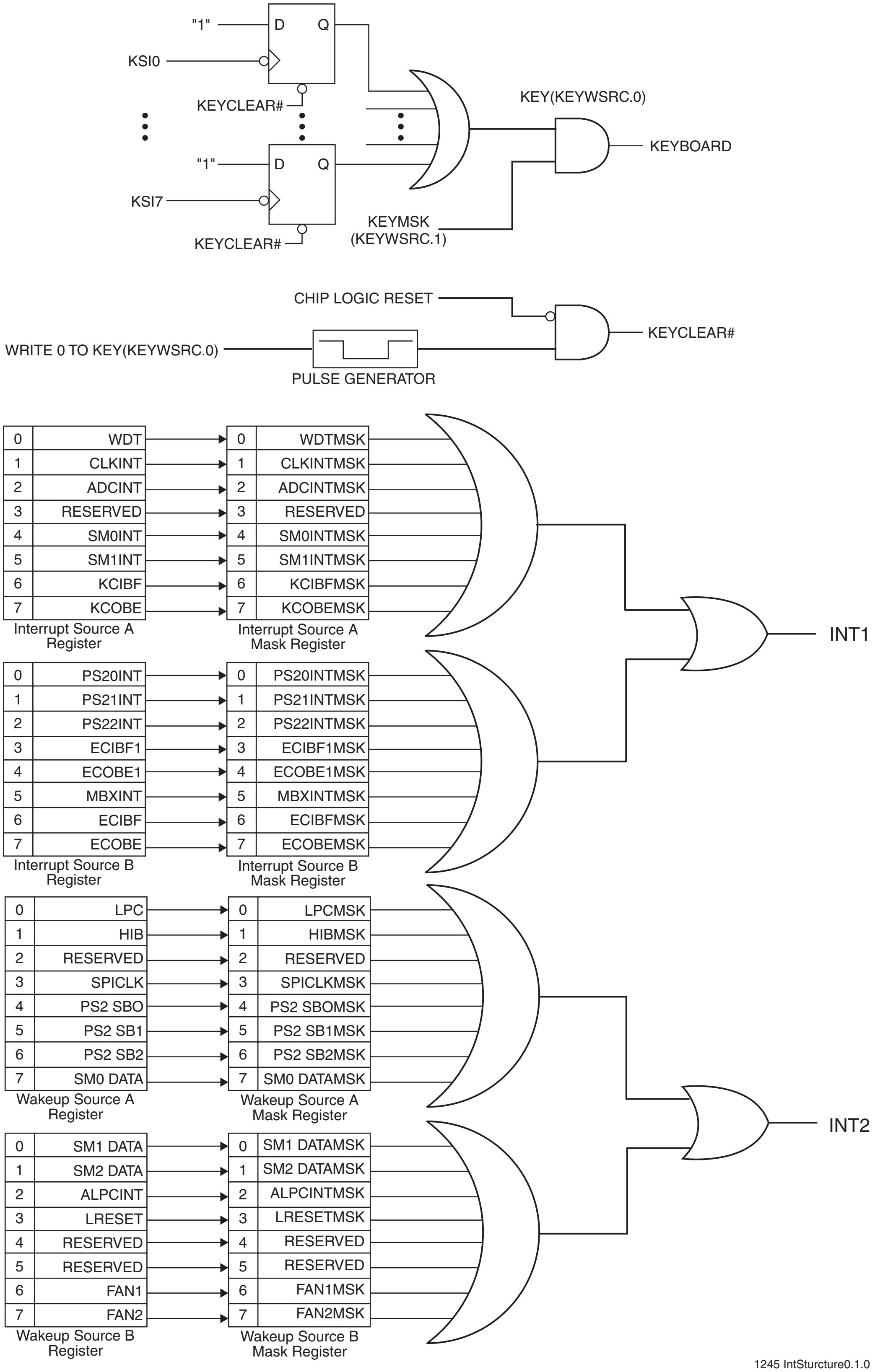
Timer 0 and Timer 1 (TF0, TF1) as well as serial interface interrupt (SPI) are dedicated interrupts. Timer 2 interrupt combines timer 2 overflow and external flag interrupts (TF2 or EXF2). Serial port UART interrupt combines transmit and receive interrupts (TI or RI). For details on Timer 0-2, UART and SPI interrupt sources see Sections 10.0, 11.0, 12.0.

TABLE 8-1: SST79LF008 Interrupt Sources

Interrupt	Description	Priority within level[1]	Interrupt request flag	ENABLE	Priority level control	Interrupt vector address	Wakeup Idle/Power Down
INT0	Keyboard	1	TCON.1	IE.0	IP.0,IPH.0	03H	Yes/Yes
TF0	Timer 0 interrupt	2	TCON.5	IE.1	IP.1,IPH.1	0BH	Yes/No
INT1	Interrupt source A and B	3	TCON.3	IE.2	IP.2,IPH.2	13H	Yes/No
TF1	Timer 1 interrupt	4	TCON.7	IE.3	IP.3,IPH.3	1BH	Yes/No
TI or RI	UART	5	SCON.0(RI), SCON.1(TI)	IE.4	IP.4,IPH.4	23H	Yes/No
TF2 or EXF2	Timer 2 interrupt	6	T2CON.7(TF2), T2CON.6(EXF2)	IE.5	IP.5,IPH.5	2BH	Yes/No
INT2	External interrupt 2	7	EXIF.0	IEA.0	IPA.0,IPAH.0	33H	Yes/Yes
INT3	External interrupt 3	8	EXIF.1	IEA.1	IPA.1,IPAH.1	3BH	Yes/Yes
INT4	External interrupt 4	9	EXIF.2	IEA.2	IPA.2,IPAH.2	43H	Yes/Yes
INT5	External interrupt 5	10	EXIF.3	IEA.3	IPA.3,IPAH.3	4BH	Yes/Yes
SPI	SPI interrupt	11	SPSR.7	SPCR.7	IP.7,IPH.7	53H	Yes/Yes

T8-1.1320

[1] Priority within level order -- 1(INTO)=highest/11(SPI)=lowest--is used to resolve simultaneous interrupt requests of the same level.



1245 IntSturcture0.1.0

FIGURE 8-1: SST79LF008 Interrupt Structure (int1-int2)

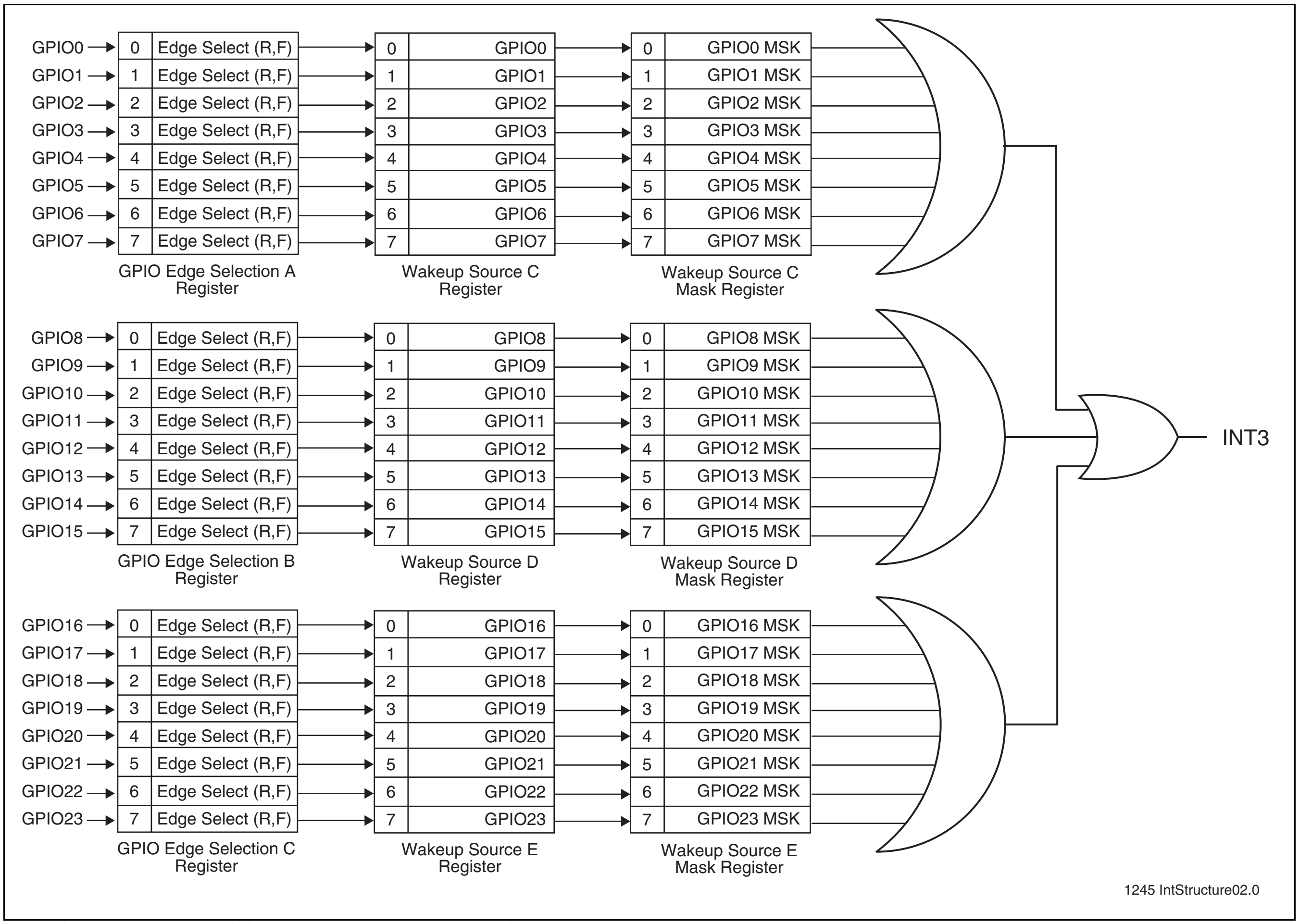


FIGURE 8-2: SST79LF008 Interrupt Structure (int3)

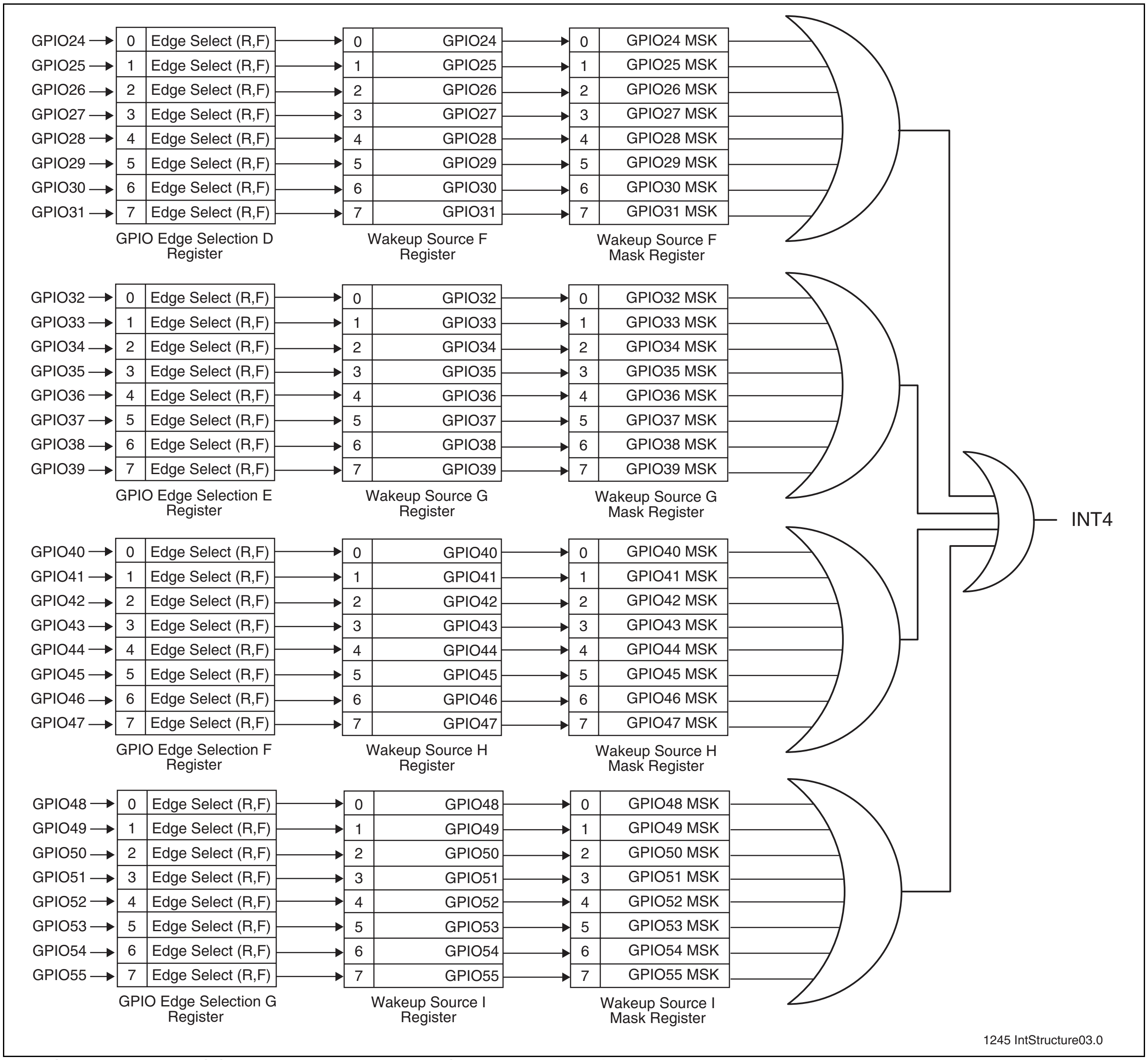


FIGURE 8-3: SST79LF008 Interrupt Structure (int4)

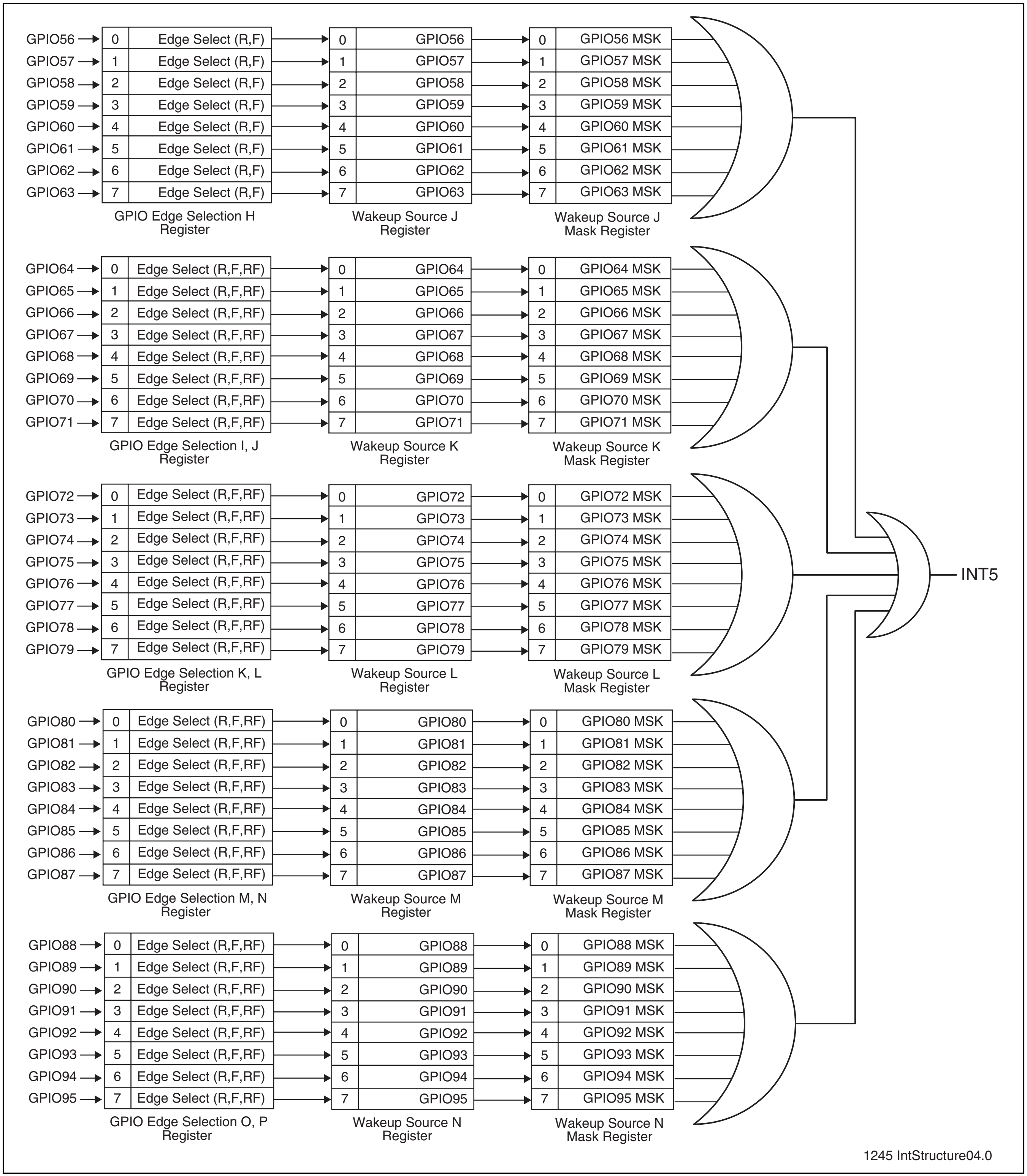

GPIO56 0 Edge Select (R,F) 0 GPIO56 0 GPIO56 MSK
GPIO57 1 Edge Select (R,F) 1 GPIO57 1 GPIO57 MSK
GPIO58 2 Edge Select (R,F) 2 GPIO58 2 GPIO58 MSK
GPIO59 3 Edge Select (R,F) 3 GPIO59 3 GPIO59 MSK
GPIO60 4 Edge Select (R,F) 4 GPIO60 4 GPIO60 MSK
GPIO61 5 Edge Select (R,F) 5 GPIO61 5 GPIO61 MSK
GPIO62 6 Edge Select (R,F) 6 GPIO62 6 GPIO62 MSK
GPIO63 7 Edge Select (R,F) 7 GPIO63 7 GPIO63 MSK
GPIO Edge Selection H Register
Wakeup Source J Register
Wakeup Source J Mask Register
GPIO64 0 Edge Select (R,F,RF) 0 GPIO64 0 GPIO64 MSK
GPIO65 1 Edge Select (R,F,RF) 1 GPIO65 1 GPIO65 MSK
GPIO66 2 Edge Select (R,F,RF) 2 GPIO66 2 GPIO66 MSK
GPIO67 3 Edge Select (R,F,RF) 3 GPIO67 3 GPIO67 MSK
GPIO68 4 Edge Select (R,F,RF) 4 GPIO68 4 GPIO68 MSK
GPIO69 5 Edge Select (R,F,RF) 5 GPIO69 5 GPIO69 MSK
GPIO70 6 Edge Select (R,F,RF) 6 GPIO70 6 GPIO70 MSK
GPIO71 7 Edge Select (R,F,RF) 7 GPIO71 7 GPIO71 MSK
GPIO Edge Selection I, J Register
Wakeup Source K Register
Wakeup Source K Mask Register
GPIO72 0 Edge Select (R,F,RF) 0 GPIO72 0 GPIO72 MSK
GPIO73 1 Edge Select (R,F,RF) 1 GPIO73 1 GPIO73 MSK
GPIO74 2 Edge Select (R,F,RF) 2 GPIO74 2 GPIO74 MSK
GPIO75 3 Edge Select (R,F,RF) 3 GPIO75 3 GPIO75 MSK
GPIO76 4 Edge Select (R,F,RF) 4 GPIO76 4 GPIO76 MSK
GPIO77 5 Edge Select (R,F,RF) 5 GPIO77 5 GPIO77 MSK
GPIO78 6 Edge Select (R,F,RF) 6 GPIO78 6 GPIO78 MSK
GPIO79 7 Edge Select (R,F,RF) 7 GPIO79 7 GPIO79 MSK
GPIO Edge Selection K, L Register
Wakeup Source L Register
Wakeup Source L Mask Register
INT5
GPIO80 0 Edge Select (R,F,RF) 0 GPIO80 0 GPIO80 MSK
GPIO81 1 Edge Select (R,F,RF) 1 GPIO81 1 GPIO81 MSK
GPIO82 2 Edge Select (R,F,RF) 2 GPIO82 2 GPIO82 MSK
GPIO83 3 Edge Select (R,F,RF) 3 GPIO83 3 GPIO83 MSK
GPIO84 4 Edge Select (R,F,RF) 4 GPIO84 4 GPIO84 MSK
GPIO85 5 Edge Select (R,F,RF) 5 GPIO85 5 GPIO85 MSK
GPIO86 6 Edge Select (R,F,RF) 6 GPIO86 6 GPIO86 MSK
GPIO87 7 Edge Select (R,F,RF) 7 GPIO87 7 GPIO87 MSK
GPIO Edge Selection M, N Register
Wakeup Source M Register
Wakeup Source M Mask Register
GPIO88 0 Edge Select (R,F,RF) 0 GPIO88 0 GPIO88 MSK
GPIO89 1 Edge Select (R,F,RF) 1 GPIO89 1 GPIO89 MSK
GPIO90 2 Edge Select (R,F,RF) 2 GPIO90 2 GPIO90 MSK
GPIO91 3 Edge Select (R,F,RF) 3 GPIO91 3 GPIO91 MSK
GPIO92 4 Edge Select (R,F,RF) 4 GPIO92 4 GPIO92 MSK
GPIO93 5 Edge Select (R,F,RF) 5 GPIO93 5 GPIO93 MSK
GPIO94 6 Edge Select (R,F,RF) 6 GPIO94 6 GPIO94 MSK
GPIO95 7 Edge Select (R,F,RF) 7 GPIO95 7 GPIO95 MSK
GPIO Edge Selection O, P Register
Wakeup Source N Register
Wakeup Source N Mask Register
1245 IntStructure04.0

FIGURE 8-4: SST79LF008 Interrupt Structure (int5)

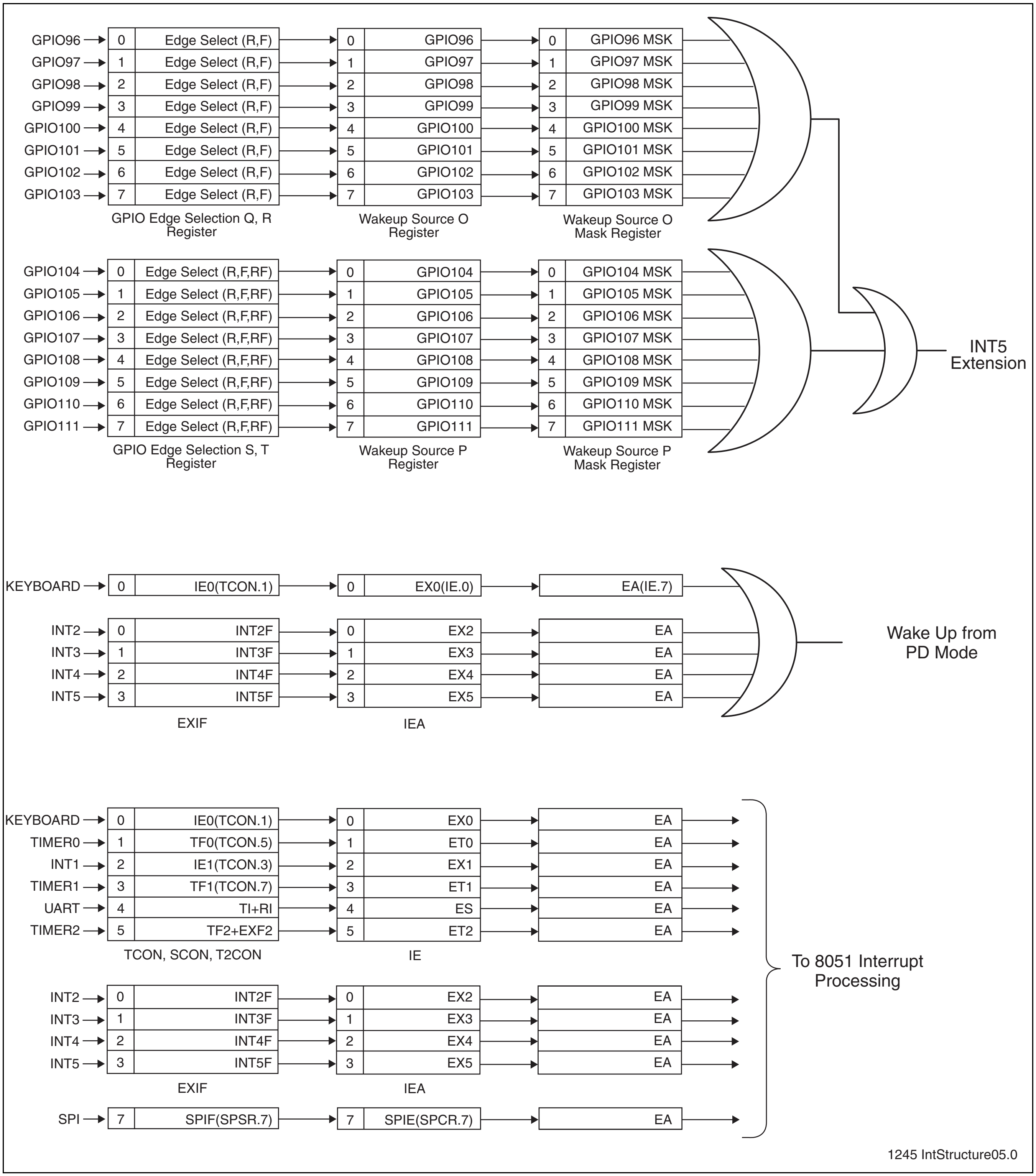


FIGURE 8-5: SST79LF008 Interrupt Structure (int5, pd mode wakeup, 8051 interrupt)

8.2 SST79LF008 Wakeups

SST79LF008 wake-up sources include LPC cycle start, LPCPD# signal transition, SPI slave clock change, SMBus start condition, PS2 clock falling edge, Matrix keyboard scanner input falling edge, Hibernation timer, FAN tachometers, and programmable GPIOs.

All GPIO wake up and interrupt sources are edge sensitive. Active edge for each wake up and interrupt is software specified via the Active Edge Selection registers. Unless explicitly stated otherwise, the GPIO wake up and interrupt requests are not asserted if the alternate function is selected for the respective pin.

For detailed information on Interrupt and Wake up Source registers and their associated Mask registers, as well as Active Edge Selection registers, refer to Section 8.3. Refer to Section 9.1 for Alternate Function Select registers. Changes in GPIO configuration, function, or edge selection may generate spurious interrupt requests, which are addressed by software.

8.3 INTERRUPT CONTROL REGISTERS

8.3.1 External Interrupt Flag Register (EXIF)

Location		7	6	5	4	3	2	1	0
ABH	Read	-	-	-	-	INT5F	INT4F	INT3F	INT2F
	Write					-	-	-	-
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
INT[5:2]F	Interrupt 5-2 flag 1: Interrupt pending 0: No Interrupt

8.3.2 Interrupt Enable Register (IE)

Location		7	6	5	4	3	2	1	0
A8H	Read	EA	-	ET2	ES	ET1	EX1	ET0	EX0
	Write								
	Reset	0	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
EA	Enable Global Interrupt
ET2	Enable Timer 2 Interrupt
ES	Enable UART Interrupt
ET1	Enable Timer 1 Interrupt
EX1	Enable Interrupt 1
ET0	Enable Timer 0 Interrupt
EX0	Enable Interrupt 0 1: Enable Interrupt 0: Disable Interrupt

8.3.3 Interrupt Enable Register A (IEA)

Location		7	6	5	4	3	2	1	0
E8H	Read	-	-	-	-	EX5	EX4	EX3	EX2
	Write								
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
EX[5:2]	Enable Interrupt 5-2 1: Enable Interrupt 0: Disable Interrupt

8.3.4 Interrupt Priority Register (IP)

Location		7	6	5	4	3	2	1	0
B8H	Read	PSPI	-	PT2	PS	PT1	PX1	PT0	PX0
	Write								
	Reset	0	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PSPI	SPI Interrupt Priority Bit
PT2	Timer 2 Interrupt Priority Bit
PS	UART Interrupt Priority Bit
PT1	Timer 1 Interrupt Priority Bit
PX1	Interrupt 1 Priority Bit
PT0	Timer 0 Interrupt Priority Bit
PX0	Interrupt 0 Priority Bit

8.3.5 Interrupt Priority High (IPH)

Location		7	6	5	4	3	2	1	0
B7H	Read	PSPIH	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
	Write								
	Reset	0	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PSPIH	SPI Interrupt Priority Bit High, with PSPI provides 4 Level Priority (11b = highest)
PT2H	Timer 2 Interrupt Priority Bit High, with PT2 provides 4 Level Priority (11b = highest)
PSH	UART Interrupt Priority Bit High, with PS provides 4 Level Priority (11b = highest)
PT1H	Timer 1 Interrupt Priority Bit High, with PT1 provides 4 Level Priority (11b = highest)
PX1H	Interrupt 1 Priority Bit High, with PX1 provides 4 Level Priority (11b = highest)
PT0H	Timer 0 Interrupt Priority Bit High, with PT0 provides 4 Level Priority (11b = highest)
PX0H	Interrupt 0 Priority Bit High, with PX0 provides 4 Level Priority (11b = highest)

8.3.6 Interrupt Priority Register A (IPA)

Location		7	6	5	4	3	2	1	0
F8H	Read	-	-	-	-	PX5	PX4	PX3	PX2
	Write								
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PX[5:2]	Interrupt 5-2 Priority bits

8.3.7 Interrupt Priority High Register A (IPAH)

Location		7	6	5	4	3	2	1	0
F7H	Read	-	-	-	-	PX5H	PX4H	PX3H	PX2H
	Write								
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PX[5:2]H	External Interrupt 5-2 Priority Bit High, with PX[5:2] provide 4 Level Priority (11b = highest)

8.3.8 Interrupt Source Register A (INTSRCA)

Location		7	6	5	4	3	2	1	0
7F00H	Read	KCOBE	KCIBF	SM1INT	SM0INT	-	ADCINT	CLKINT	WDT
	Write	-	-	-	-		-		-
	Reset	1	0	0	0	X	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
KCOBE	Keyboard controller OBE (Output Buffer Empty) interrupt flag. Set when OBF bit in KBCSTS register is ‘0’, cleared when OBF bit is ‘1’.
KCIBF	Keyboard controller IBF (Input Buffer Full) interrupt flag. Set when IBF bit in KBCSTS register is ‘1’, cleared when IBF bit is ‘0’.
SM1INT	SMBus channel 1 or 2 interrupt flag. Set when INT bit in SMCR1 register is ‘1’, cleared when INT bit is ‘0’.
SM0INT	SMBus channel 0 interrupt flag. Set when INT bit in SMCR0 register is ‘1’, cleared when INT bit is ‘0’.
ADCINT	A/D conversion completion interrupt flag. Set when ADF bit in ADCSR register is ‘1’, cleared when ADF bit is ‘0’.
CLKINT	Clock source change interrupt flag. Set when PLLOK bit changes from ‘0’ to ‘1’, or on any change of ECLOK bit. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
WDT	Watchdog timer underflow flag. Set when WDT underflows, cleared when WDT is reloaded or disabled.

8.3.9 Interrupt Source A Mask Register (INTSRCAMSK)

Location		7	6	5	4	3	2	1	0
7F01H	Read	KCOBE MSK	KCIBF MSK	SM1INT MSK	SM0INT MSK	-	ADCINT MSK	CLKINT MSK	WDT MSK
	Write								
	Reset	0	0	0	0	X	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
KCOBEMSK, …,	Interrupt Source Mask
WDTMSK	1: Enable Interrupt from the respective source in INTSRCA register 0: Mask Interrupt source

8.3.10 Interrupt Source Register B (INTSRCB)

Location		7	6	5	4	3	2	1	0
7F02H	Read	ECOBE	ECIBF	MBXINT	ECOBE1	ECIBF1	PS22INT	PS21INT	PS20INT
	Write	-	-	-	-	-	-	-	-
	Reset	1	0	0	1	0	0	0	0

Symbol	Function
-	Not implemented
ECOBE	ACPI ECI channel 0 Output Buffer Empty interrupt flag. Set when OBF bit in ECISTS register is ‘0’, cleared when OBF bit is ‘1’.
ECIBF	ACPI ECI channel 0 Input Buffer Full interrupt flag. Set when IBF bit in ECISTS register is ‘1’, cleared when IBF bit is ‘0’.
MBXINT	Mailbox interface System-to-8051interrupt. Set when the system writes to Mailbox register 0, cleared when 8051 reads Mailbox register 0.
ECOBE1	ACPI ECI channel 1 Output Buffer Empty interrupt flag. Set when OBF bit in ECISTS1 register is ‘0’, cleared when OBF bit is ‘1’.
ECIBF1	ACPI ECI channel 1 Input Buffer Full interrupt flag. Set when IBF bit in ECISTS1 register is ‘1’, cleared when IBF bit is ‘0’.
PS22INT	PS2 channel 2 interrupt If PS/2 h/w state machine is enabled, set by ‘0’ to ‘1’ transition of any of the following bits in PS2STS2 register: PS2STSn_RDATA_RDY, PS2STSn_XMIT_IDLE, PS2STSn_R_TIMEOUT or PS2STSn_T_TIMEOUT. If PS/2 h/w state machine is disabled, set by falling edge on PSCLK2 pin driven by the peripheral. Cleared by reading PS2STS2 register (in any mode of PS2 h/w state machine).
PS21INT	PS2 channel 1 interrupt If PS/2 h/w state machine is enabled, set by ‘0’ to ‘1’ transition of any of the following bits in PS2STS1 register: PS2STSn_RDATA_RDY, PS2STSn_XMIT_IDLE, PS2STSn_R_TIMEOUT or PS2STSn_T_TIMEOUT. If PS/2 h/w state machine is disabled, set by falling edge on PSCLK1 pin driven by the peripheral. Cleared by reading PS2STS1 register (in any mode of PS2 h/w state machine).
PS20INT	PS2 channel 0 interrupt If PS/2 h/w state machine is enabled, set by ‘0’ to ‘1’ transition of any of the following bits in PS2STS0 register: PS2STSn_RDATA_RDY, PS2STSn_XMIT_IDLE, PS2STSn_R_TIMEOUT or PS2STSn_T_TIMEOUT. If PS/2 h/w state machine is disabled Set by falling edge on PSCLK0 pin driven by the peripheral. Cleared by reading PS2STS0 register (in any mode of PS2 h/w state machine).

8.3.11 Interrupt Source B Mask Register (INTSRCBMSK)

Location		7	6	5	4	3	2	1	0
7F03H	Read	ECOBE MSK	ECIBF MSK	MBXINT MSK	ECOBE1 MSK	ECIBF1 MSK	PS22INT MSK	PS21INT MSK	PS20INT MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
ECOBEMSK, …, PS20INTMSK	Interrupt Source Mask 1: Enable Interrupt from the respective source in INTSRCB register 0: Mask Interrupt source

8.3.12 Wakeup Source A Register (WSRCA)

Location		7	6	5	4	3	2	1	0
7F2AH	Read	SMB0_ DATA	PS2_ SB2	PS2_ SB1	PS2_ SB0	SPICLK	-	HIB_TO	LPC
	Write							-	
	Reset	0	0	0	0	0	X	1	0

Symbol	Function
-	Not implemented
X	Not defined
SMB0_DATA	SMBus channel 0 start condition detection flag. Set when start condition is detected Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
PS2_SB2	PS2 channel 2 start bit detection flag. Set by falling edge on PSCLK2 pin driven by the peripheral (in any mode of PS2 h/w state machine). Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
PS2_SB1	PS1 channel 1 start bit detection flag. Set by falling edge on PSCLK1 pin driven by the peripheral (in any mode of PS2 h/w state machine). Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
PS2_SB0	PS2 channel 0 start bit detection flag. Set by falling edge on PSCLK0 pin driven by the peripheral (in any mode of PS2 h/w state machine). Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
SPICLK	SPI clock edge detection flag (in SPI slave mode). Set when any transition of SPI Clock (SCK) or SPI port Select (SS#) signal is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
HIB_TO	Hibernation timer time out flag. Set when Hibernation timer underflows, cleared when Hibernation timer is restarted.
LPC	LPC LFRAME# falling edge detection flag. Set when falling edge is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.

8.3.13 Wakeup Source A Wakeup Mask Register (WSRCAMSK)

Location		7	6	5	4	3	2	1	0
7F2BH	Read	SMB0_DATA_MSK	PS2_SB2_MSK	PS2_SB1_MSK	PS2_SB0_MSK	SPICLK_MSK	-	HIB_MSK	LPC_MSK
	Write								
	Reset	0	0	0	0	0	X	0	0

Symbol	Function
-	Not implemented
X	Not defined
SMB0_DATA_MSK,	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
PS2_SB2_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
PS2_SB1_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
PS2_SB0_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
SPICLK_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
HIB_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source
LPC_MSK	Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt from the respective source in WSRCA register 0: Mask Wakeup and Interrupt source

8.3.14 Wakeup Source B Register (WSRCB)

Location		7	6	5	4	3	2	1	0
7F2CH	Read	FAN2	FAN1	-	-	LRESET	aLPCINT	SMB2_ DATA	SMB1_ DATA
	Write	-	-						
	Reset	0	0	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
FAN[2:1]	FAN Tachometer [2:1] threshold detection flag. Set when FAN tachometer counter exceeds the threshold, cleared when counter is reloaded with value below threshold.
LRESET	LPC LRESET# falling edge detection flag. Set when falling edge is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
aLPCINT	aLPC Enable_and_Poll sequence detection flag. Set when Enable_and_Poll sequence is received over aLPC bus. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
SMB2_DATA	SMBus channel 2 start condition detection flag. Set when start condition is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.
SMB1_DATA	SMBus channel 1 start condition detection flag. Set when start condition is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.

8.3.15 Wakeup Source B Mask Register (WSRCBMSK)

Location		7	6	5	4	3	2	1	0
7F2DH	Read	FAN2MSK	FAN1MSK	-	-	LRESET_ MSK	aLPCIN MSK	SMB2_ MSK	SMB1_ MSK
	Write								
	Reset	0	0	X	X	0	X	0	0

Symbol	Function
-	Not implemented
X	Not defined
FAN2MSK, …,	Wakeup and Interrupt Source Mask
SMB1_MSK	1: Enable Wakeup and Interrupt from the respective source in WSRCB register 0: Mask Wakeup and Interrupt source

8.3.16 Keyboard Wakeup Control Register (KEYWSRC)

Location		7	6	5	4	3	2	1	0
7F2FH	Read	-	-	-	-	-	-	KEYMSK	KEY
	Write								
	Reset	X	X	X	X	X	X	0	0

Symbol	Function
-	Not implemented
X	Not defined
KEYMSK	Keystroke Wake up and Interrupt Source Mask 1: Enable Wake up and Interrupt when KEY bit is set 0: Mask and keystroke Wake up and Interrupt
KEY	Keystroke press detection flag. Set when falling edge on any scanner input lines KSI[7:0] is detected. Write ‘0’ to clear. Writing ‘1’ to this bit will be ignored.

8.3.17 Wakeup Source C Register (WSRCC)

Location		7	6	5	4	3	2	1	0
7F59H	Read	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GP102	GPIO1	GPIO0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[7:0]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.18 Wakeup Source C Mask Register (WSRCCMSK)

Location		7	6	5	4	3	2	1	0
7F5AH	Read	GPIO7_ MSK	GPIO6_ MSK	GPIO5_ MSK	GPIO4_ MSK	GPIO3_ MSK	GPIO2_ MSK	GPIO1_ MSK	GPIO0_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[7:0]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.19 GPIO Active Edge Selection Register A (GPIOESA)

Location		7	6	5	4	3	2	1	0
7F57H	Read	GPIO7_ ES	GPIO6_ ES	GPIO5_ ES	GPIO4_ ES	GPIO3_ ES	GPIO2_ ES	GPIO1_ ES	GPIO0_ ES
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[7:0]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.20 Wakeup Source D Register (WSRCD)

Location		7	6	5	4	3	2	1	0
7F5EH	Read	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[15:8]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.21 Wakeup Source D Mask Register (WSRCDMSK)

Location		7	6	5	4	3	2	1	0
7F5FH	Read	GPIO15_ MSK	GPIO14_ MSK	GPIO13_ MSK	GPIO12_ MSK	GPIO11_ MSK	GPIO10_ MSK	GPIO9_ MSK	GPIO8_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[15:8]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.22 GPIO Active Edge Selection Register B (GPIOESB)

Location		7	6	5	4	3	2	1	0
7F58H	Read	GPIO15_ ES	GPIO14_ ES	GPIO13_ ES	GPIO12_ ES	GPIO11_ ES	GPIO10_ ES	GPIO9_ ES	GPIO8_ ES
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[15:8]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.23 Wakeup Source E Register (WSRCE)

Location		7	6	5	4	3	2	1	0
7F63H	Read	GPIO23	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[23:16]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.24 Wakeup Source E Mask Register (WSRCEMSK)

Location		7	6	5	4	3	2	1	0
7F66H	Read	GPIO23_ MSK	GPIO22_ MSK	GPIO21_ MSK	GPIO20_ MSK	GPIO19_ MSK	GPIO18_ MSK	GPIO17_ MSK	GPIO16_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[23:16]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.25 GPIO Active Edge Selection Register C (GPIOESC)

Location		7	6	5	4	3	2	1	0
7F5CH	Read Write	GPIO23_ ES	GPIO22_ ES	GPIO21_ ES	GPIO20_ ES	GPIO19_ ES	GPIO18_ ES	GPIO17_ ES	GPIO16_ ES
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[23:16]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.26 Wakeup Source F Register (WSRCF)

Location		7	6	5	4	3	2	1	0
7F64H	Read Write	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[31:24]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.27 Wakeup Source F Mask Register (WSRCFMSK)

Location		7	6	5	4	3	2	1	0
7F65H	Read Write	GPIO31_ MSK	GPIO30_ MSK	GPIO29_ MSK	GPIO28_ MSK	GPIO27_ MSK	GPIO26_ MSK	GPIO25_ MSK	GPIO24_ MSK
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[31:24]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.28 GPIO Active Edge Selection Register D (GPIOESD)

Location		7	6	5	4	3	2	1	0
7F5DH	Read Write	GPIO31_ ES	GPIO30_ ES	GPIO29_ ES	GPIO28_ ES	GPIO27_ ES	GPIO26_ ES	GPIO25_ ES	GPIO24_ ES
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[31:24]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.29 Wakeup Source G Register (WSRCG)

Location		7	6	5	4	3	2	1	0
7F55H	Read	GPIO39	GPIO38/ LPCPD[1]	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
	Write								
	Reset	0	0	0	0	0	0	0	0

[1] This interrupt source bit is set when active edge is detected regardless of whether GPIO38 or alternate LPCPD# function is selected

Symbol	Function
GPIO[39:32]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.30 Wakeup Source G Mask Register (WSRCGMSK)

Location		7	6	5	4	3	2	1	0
7F56H	Read	GPIO39_ MSK	GPIO38_ MSK	GPIO37_ MSK	GPIO36_ MSK	GPIO35_ MSK	GPIO34_ MSK	GPIO33_ MSK	GPIO32_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[39:32]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.31 GPIO Active Edge Selection Register E (GPIOESE)

Location		7	6	5	4	3	2	1	0
7F60H	Read	GPIO39_ ES	GPIO38_ ES	GPIO37_ ES	GPIO36_ ES	GPIO35_ ES	GPIO34_ ES	GPIO33_ ES	GPIO32_ ES
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[39:32]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.32 Wakeup Source H Register (WSRCH)

Location		7	6	5	4	3	2	1	0
7FAEH	Read	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[47:40]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.33 Wakeup Source H Mask Register (WSRCHMSK)

Location		7	6	5	4	3	2	1	0
7FAFH	Read Write	GPIO47_ MSK	GPIO46_ MSK	GPIO45_ MSK	GPIO44_ MSK	GPIO43_ MSK	GPIO42_ MSK	GPIO41_ MSK	GPIO40_ MSK
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[47:40]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.34 GPIO Active Edge Selection Register F (GPIOESF)

Location		7	6	5	4	3	2	1	0
7F61H	Read Write	GPIO47_ ES	GPIO46_ ES	GPIO45_ ES	GPIO44_ ES	GPIO43_ ES	GPIO42_ ES	GPIO41_ ES	GPIO40_ ES
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[47:40]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.35 Wakeup Source I Register (WSRCI)

Location		7	6	5	4	3	2	1	0
7F3EH	Read Write	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[55:48]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.36 Wakeup Source I Mask Register (WSRCIMSK)

Location		7	6	5	4	3	2	1	0
7F3FH	Read Write	GPIO55_ MSK	GPIO54_ MSK	GPIO53_ MSK	GPIO52_ MSK	GPIO51_ MSK	GPIO50_ MSK	GPIO49_ MSK	GPIO48_ MSK
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[55:48]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.37 GPIO Active Edge Selection Register G (GPIOESG)

Location		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7F62H	Read	GPIO55_ ES	GPIO54_ ES	GPIO53_ ES	GPIO52_ ES	GPIO51_ ES	GPIO50_ ES	GPIO49_ ES	GPIO48_ ES
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
GPIO[55:48]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.38 Wakeup Source J Register (WSRCJ)

Location		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7FC8H	Read	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
GPIO[63:56]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.39 Wakeup Source J Mask Register (WSRCJMSK)

Location		7	6	5	4	3	2	1	0
7FC9H	Read	GPIO63_ MSK	GPIO62_ MSK	GPIO61_ MSK	GPIO60_ MSK	GPIO59_ MSK	GPIO58_ MSK	GPIO57_ MSK	GPIO56_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
GPIO[63:56]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.40 GPIO Active Edge Selection Register H (GPIOESH)

Location		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7F6CH	Read	GPIO63_ ES	GPIO62_ ES	GPIO61_ ES	GPIO60_ ES	GPIO59_ ES	GPIO58_ ES	GPIO57_ ES	GPIO56_ ES
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
GPIO[63:56]_ES	GPIO active edge control bit 1: Select Rising edge (Low to High transition) 0: Select Falling edge (High to Low transition)

8.3.41 Wakeup Source K Register (WSRCK)

Location		**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
7FCAH	Read	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
GPIO[71:64]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.42 Wakeup Source K Mask Register (WSRCKMSK)

Location		7	6	5	4	3	2	1	0
7FCBH	Read	GPIO71_ MSK	GPIO70_ MSK	GPIO69_ MSK	GPIO68_ MSK	GPIO67_ MSK	GPIO66_ MSK	GPIO65_ MSK	GPIO64_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[71:64]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.43 GPIO Active Edge Selection Register I (GPIOESI)

Location		7	6	5	4	3	2	1	0
7F6DH	Read	GPIO67_ ES1	GPIO67_ ES0	GPIO66_ ES1	GPIO66_ ES0	GPIO65_ ES1	GPIO65_ ES0	GPIO64_ ES1	GPIO64_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[67:64]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.44 GPIO Active Edge Selection Register J (GPIOESJ)

Location		7	6	5	4	3	2	1	0
7F6EH	Read	GPIO71_ ES1	GPIO71_ ES0	GPIO70_ ES1	GPIO70_ ES0	GPIO69_ ES1	GPIO69_ ES0	GPIO68_ ES1	GPIO68_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[71:68]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.45 Wakeup Source L Register (WSRCL)

Location		7	6	5	4	3	2	1	0
7FCCH	Read	GPIO79	GPIO78	GPIO77	GPIO76	GPIO75	GPIO74	GPIO73	GPIO72
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:72]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.46 Wakeup Source L Mask Register (WSRCLMSK)

Location		7	6	5	4	3	2	1	0
7FCDH	Read	GPIO79_ MSK	GPIO78_ MSK	GPIO77_ MSK	GPIO76_ MSK	GPIO75_ MSK	GPIO74_ MSK	GPIO73_ MSK	GPIO72_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:72]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.47 GPIO Active Edge Selection Register K (GPIOESK)

Location		7	6	5	4	3	2	1	0
7F6FH	Read	GPIO75_ ES1	GPIO75_ ES0	GPIO74_ ES1	GPIO74_ ES0	GPIO73_ ES1	GPIO73_ ES0	GPIO72_ ES1	GPIO72_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[75:72]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.48 GPIO Active Edge Selection Register L (GPIOESL)

Location		7	6	5	4	3	2	1	0
7FD0H	Read	GPIO79_ ES1	GPIO79_ ES0	GPIO78_ ES1	GPIO78_ ES0	GPIO77_ ES1	GPIO77_ ES0	GPIO76_ ES1	GPIO76_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:76]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.49 Wakeup Source M Register (WSRCM)

Location		7	6	5	4	3	2	1	0
7FCEH	Read	GPIO87	GPIO86	GPIO85	GPIO84	GPIO83	GPIO82	GPIO81	GPIO80
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[87:80]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.50 Wakeup Source M Mask Register (WSRCMMSK)

Location		7	6	5	4	3	2	1	0
7FCFH	Read	GPIO87_MSK	GPIO86_MSK	GPIO85_MSK	GPIO84_MSK	GPIO83_MSK	GPIO82_MSK	GPIO81_MSK	GPIO80_MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[87:80]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.51 GPIO Active Edge Selection Register M (GPIOESM)

Location		7	6	5	4	3	2	1	0
7FD1H	Read	GPIO83_ES1	GPIO83_ES0	GPIO82_ES1	GPIO82_ES0	GPIO81_ES1	GPIO81_ES0	GPIO80_ES1	GPIO80_ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[83:80]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.52 GPIO Active Edge Selection Register N (GPIOESN)

Location		7	6	5	4	3	2	1	0
7FD2H	Read	GPIO87_ES1	GPIO87_ES0	GPIO86_ES1	GPIO86_ES0	GPIO85_ES1	GPIO85_ES0	GPIO84_ES1	GPIO84_ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[87:84]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.53 GPIO K Interrupt Selection Register (GPIOKINT)

Location		7	6	5	4	3	2	1	0
7FB8H	Read	-	-	GPIO85INT	GPIO84INT	GPIO83INT	GPIO82INT	GPIO81INT	GPIO80INT
	Write								
	Reset	X	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO[85:80]INT	GPIO Interrupt control bit 1: GPIO active edge detection generates INT0 (combined with any KEY press detection interrupt) 0: GPIO active edge detection generates INT5 (as shown on Figure 8-1)

8.3.54 Wakeup Source N Register (WSRCN)

Location		7	6	5	4	3	2	1	0
7FB0H	Read	GPIO95	GPIO94	GPIO93	GPIO92	GPIO91	GPIO90	GPIO89	GPIO88
	Write								
	Reset	X	0	0	0	0	0	0	0

Symbol	Function
X	Not defined
GPIO[95:88]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.55 Wakeup Source N Mask Register (WSRCNMSK)

Location		7	6	5	4	3	2	1	0
7FB1H	Read	GPIO95_ MSK	GPIO94_ MSK	GPIO93_ MSK	GPIO92_ MSK	GPIO91_ MSK	GPIO90_ MSK	GPIO89_ MSK	GPIO88_ MSK
	Write								
	Reset	X	0	0	0	0	0	0	0

Symbol	Function
X	Not defined
GPIO[95:88]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.56 GPIO Active Edge Selection Register O (GPIOESO)

Location		7	6	5	4	3	2	1	0
7FD3H	Read	GPIO91_ ES1	GPIO91_ ES0	GPIO90_ ES1	GPIO90_ ES0	GPIO89_ ES1	GPIO89_ ES0	GPIO88_ ES1	GPIO88_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[91:88]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.57 GPIO Active Edge Selection Register P (GPIOESP)

Location		7	6	5	4	3	2	1	0
7FD4H	Read	GPIO95_ ES1	GPIO95_ ES0	GPIO94_ ES1	GPIO94_ ES0	GPIO93_ ES1	GPIO93_ ES0	GPIO92_ ES1	GPIO92_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[95:92]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.58 Wakeup Source O Register (WSRCO)

Location		7	6	5	4	3	2	1	0
7FDBH	Read	GPIO103	GPIO102	GPIO101	GPIO100	GPIO99	GPIO98	GPIO97	GPIO96
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[103:96]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.59 Wakeup Source O Mask Register (WSRCOMSK)

Location		7	6	5	4	3	2	1	0
7FECH	Read	GPIO103 _MSK	GPIO102 _MSK	GPIO101 _MSK	GPIO100 _MSK	GPIO99_ MSK	GPIO98_ MSK	GPIO97_ MSK	GPIO96_ MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[103:96]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.60 GPIO Active Edge Selection Register Q (GPIOESQ)

Location		7	6	5	4	3	2	1	0
7FD5H	Read	GPIO99_ ES1	GPIO99_ ES0	GPIO98_ ES1	GPIO98_ ES0	GPIO97_ ES1	GPIO97_ ES0	GPIO96_ ES1	GPIO96_ ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[99:96]_ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.61 GPIO Active Edge Selection Register R (GPIOESR)

Location		7	6	5	4	3	2	1	0
7FD6H	Read	GPIO103 _ES1	GPIO103 _ES0	GPIO102 _ES1	GPIO102 _ES0	GPIO101 _ES1	GPIO101 _ES0	GPIO100 _ES1	GPIO100 _ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[103:100] _ES[1:0]	GPIO active edge control bit 01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.62 Wakeup Source P Register (WSRCP)

Location		7	6	5	4	3	2	1	0
7FEDH	Read	GPIO111	GPIO110	GPIO109	GPIO108	GPIO107	GPIO106	GPIO105	GPIO104
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[111:104]	GPIO active edge detection flag. Set when selected edge is detected. Write 0 to clear. Writing 1 to this bit will be ignored.

8.3.63 Wakeup Source P Mask Register (WSRCPMSK)

Location		7	6	5	4	3	2	1	0
7FEEH	Read	GPIO111 _MSK	GPIO110 _MSK	GPIO109 _MSK	GPIO108 _MSK	GPIO107 _MSK	GPIO106 _MSK	GPIO105 _MSK	GPIO104 _MSK
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[111:104]_MSK	GPIO Wakeup and Interrupt Source Mask 1: Enable Wakeup and Interrupt when GPIO edge detection flag is set 0: Mask GPIO Wakeup and Interrupt

8.3.64 GPIO Active Edge Selection Register S (GPIOESS)

Location		7	6	5	4	3	2	1	0
7FD7H	Read	GPIO107 _ES1	GPIO107 _ES0	GPIO106 _ES1	GPIO106 _ES0	GPIO105 _ES1	GPIO105 _ES0	GPIO104 _ES1	GPIO104 _ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[107:104]	GPIO active edge control bit
_ES[1:0]	01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

8.3.65 GPIO Active Edge Selection Register T (GPIOEST)

Location		7	6	5	4	3	2	1	0
7FD8H	Read	GPIO111 _ES1	GPIO111 _ES0	GPIO110 _ES1	GPIO110 _ES0	GPIO109 _ES1	GPIO109 _ES0	GPIO108 _ES1	GPIO108 _ES0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[111:108]	GPIO active edge control bit
_ES[1:0]	01: Select Rising edge (Low to High transition) 10: Select Both Falling and Rising Edges 00 or 11: Select Falling edge (High to Low transition)

9.0 GPIO PORTS

The SST79LF008 has 112 general purpose input/output pins (GPIOs); 77 pins are multiplexed with alternate functions, as shown in to Figure 2-1, and 35 pins are dedicated GPIOs. All GPIO pins also generate 8051 Interrupt and Wake up events. See Section 8.0 for additional information on Interrupt and Wake up control.

Use the GPIO Function Selection Registers to select either the GPIO function or an alternate function for the corresponding pins. When an alternate function is selected the direction of the pin as well as output data is determined by the peripheral module that controls the alternate function.

When selecting the GPIO function, the direction of the pin is determined by the respective GPIO Direction Register, and the output data for output pins is specified by the respective Output Register. In other words, the status of the pin is controlled by the data in the GPIO Output Register when the GPIO function is selected and output direction is specified). No direction control is provided for GPI43-GPI45 pins, which are always configured as inputs. Also, no direction control is provided for the GPIO16-GPIO22, GPIO28-GPIO39, and GPIO40 pins with open drain buffers, which are always configured as outputs.

Note: When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

GPIO60-GPIO79, GPIO88-GPIO91, and GPIO96-GPIO111 with push-pull buffers can be used to emulate open drain configuration by tri-stating the push-pull buffer when output data is ‘1’. The respective open-drain/push-pull selection registers control buffer configuration for these pins.

GPIO80-GPIO85 and GPIO94-GPIO111 have programmable pull-up resistors, which can be enabled/disabled by the respective pull-up control registers.

For most pins, reading the GPIO input registers will return the status of the pins, regardless of selected function. The only exceptions being, GPIO68-GPIO71 and GPIO72-GPIO79 are multiplexed with DAC outputs or ADC inputs are the only exceptions. These pins can be read via input registers in the GPIO mode only. Reading the input register when DAC or ADC alternate function is enabled will return an indeterminate value.

For detailed information on GPIO control registers refer to Section 9.1. See also, the GPIO buffer types list in Table 2-1 and Table 2-2.

9.1 GPIO CONTROL REGISTERS

9.1.1 GPIO A Direction Register (GPIOADIR)

Location		7	6	5	4	3	2	1	0
7F18H	Read	GPIO7_ DiR	GPIO6_ DIR	GPIO5_ DIR	GPIO4_ DIR	GPIO3_ DIR	GPIO2_ DIR	GPIO1_ DIR	GPIO0_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[7:0]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.2 GPIO A Input Register (GPIOAIN)[1]

Location		7	6	5	4	3	2	1	0
7F1AH	Read	GPIO7_IN	GPIO6_IN	GPIO5_IN	GPIO4_IN	GPIO3_IN	GPIO2_IN	GPIO1_IN	GPIO0_IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO7_IN	GPIO6_IN	GPIO5_IN	GPIO4_IN	GPIO3_IN	GPIO2_IN	GPIO1_IN	GPIO0_IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[7:0]_IN	When read, returns the status of the pin.

9.1.3 GPIO A Output Register (GPIOAOUT)

Location		7	6	5	4	3	2	1	0
7F19H	Read	GPIO7_ OUT	GPIO6_ OUT	GPIO5_ OUT	GPIO4_ OUT	GPIO3_ OUT	GPIO2_ OUT	GPIO1_ OUT	GPIO0_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[7:0]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.4 GPIO A Function Select Register (GPIOASEL)

Location		7	6	5	4	3	2	1	0
7F3DH	Read	GPIO7_ SEL	GPIO6_ SEL	GPIO5_ SEL	GPIO4_ SEL	GPIO3_ SEL	GPIO2_ SEL	GPIO1_ SEL	GPIO0_ SEL
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO7_SEL	1: GA20 (Gate A20 output) 0: GPIO7 function
GPIO6_SEL	1: SS# (Slave port select input for SPI) 0: GPIO6 function
GPIO5_SEL	1: SCK (Master clock output, slave clock input pin for SPI) 0: GPIO5 function
GPIO4_SEL	1: MISO (Master data input pin, slave data output pin for SPI) 0: GPIO4 function
GPIO3_SEL	1: MOSI (Master data output pin, slave data input pin for SPI) 0: GPIO3 function
GPIO2_SEL	1: PWM2 (Pulse Width Modulator output 2) 0: GPIO2 function
GPIO1_SEL	1: PWM1 (Pulse Width Modulator output 1) 0: GPIO1 function
GPIO0_SEL	1: PWM0 (Pulse Width Modulator output 0) 0: GPIO0 function

9.1.5 GPIO B Direction Register (GPIOBDIR)

Location		7	6	5	4	3	2	1	0
7F1BH	Read	GPIO15_ DIR	GPIO14_ DIR	GPIO13_ DIR	GPIO12_ DIR	GPIO11_ DIR	GPIO10_ DIR	GPIO9_ DIR	GPIO8_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[15:8]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.6 GPIO B Input Register (GPIOBIN)[1]

Location		7	6	5	4	3	2	1	0
7F1DH	Read	GPIO15_ IN	GPIO14_ IN	GPIO13_ IN	GPIO12_ IN	GPIO11_ IN	GPIO10_ IN	GPIO9_ IN	GPIO8_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO15_ IN	GPIO14_ IN	GPIO13_ IN	GPIO12_ IN	GPIO11_ IN	GPIO10_ IN	GPIO9_ IN	GPIO8_ IN

[1] When using an open drain pin as an input, output data must be specified as '1'. However, specifying output data as '0' when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[15:8]_IN	When read, returns the status of the pin.

9.1.7 GPIO B Output Register (GPIOBOUT)

Location		7	6	5	4	3	2	1	0
7F1CH	Read	GPIO15_ OUT	GPIO14_ OUT	GPIO13_ OUT	GPIO12_ OUT	GPIO11_ OUT	GPIO10_ OUT	GPIO9_ OUT	GPIO8_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[15:8]_OUT	When written to, output data is updated. When read, returns previously written data.

.

9.1.8 GPIO B Function Select Register (GPIOBSEL)

Location		7	6	5	4	3	2	1	0
7F40H	Read	GPIO15_ SEL	GPIO14_ SEL	GPIO13_ SEL	GPIO12_ SEL	GPIO11_ SEL	GPIO10_ SEL	-	-
	Write								
	Reset	0	0	0	0	0	0	X	X

Symbol	Function
-	Not implemented
X	Not defined
GPIO15_SEL	1: T2EX (Timer 2 external interrupt input) 0: GPIO15 function
GPIO14_SEL	1: SS2 (LPC Host interface status signal output 2) 0: GPIO14 function
GPIO13_SEL	1: SS1 (LPC Host interface status signal output 1) 0: GPIO13 function
GPIO12_SEL	1: SS0 (LPC Host interface status signal output 0) 0: GPIO12 function
GPIO11_SEL	1: KBRST#. (Keyboard Controller reset to CPU output) 0: GPIO11 function
GPIO10_SEL	1: ECLK (External clock input) 0: GPIO10 function

9.1.9 LPC Status Signals Output Control Register (LPCSS)

Location		7	6	5	4	3	2	1	0
7FDFH	Read Write	-	-	-	-	-	SSEL2	SSEL1	SSEL0
	Reset	X	X	X	X	X	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
SSEL[2:0]	LPC Host status signals selection bits

TABLE 9-1: LPC Host Status Signals as a Function of SSEL[2:0]

SSEL2 SSEL1 SSEL0	SS0# Output	SS1# Output	SS2# Output
000	KBCSTS[IBF or OBF][1]	ECISTS[IBF or OBF][1]	Mailbox Interrupt Status[2]
001	KBCSTS[IBF or OBF]	Mailbox Interrupt Status	ECISTS[IBF or OBF]
010	ECISTS[IBF or OBF]	KBCSTS[IBF or OBF]	Mailbox Interrupt Status
011	ECISTS[IBF or OBF]	Mailbox Interrupt Status	KBCSTS[IBF or OBF]
100	Mailbox Interrupt Status	KBCSTS[IBF or OBF]	ECISTS[IBF or OBF]
101	Mailbox Interrupt Status	ECISTS[IBF or OBF]	KBCSTS[IBF or OBF]
110-111	Reserved	Reserved	Reserved

[1] Asserted when either IBF or OBF flag in the respective status register is set ‘1’ (see also Sections 18.0, 20.0).

[2] Asserted when either Host to 8051 or 8051 to Host interrupt is pending (see also Section 21.0).

9.1.10 GPIO C Direction Register (GPIOCDIR)

Location		7	6	5	4	3	2	1	0
7F1EH	Read Write	GPIO23_ DIR	-	-	-	-	-	-	-
	Reset	0	X	X	X	X	X	X	X

Symbol	Function-
-	Not implemented
X	Not defined
GPIO23_DIR	GPIO direction control bit 1: Output 0: Input

9.1.11 GPIO C Input Register (GPIOCIN)[1]

Location		7	6	5	4	3	2	1	0
7F20H	Read	GPIO23_ IN	GPIO22_ IN	GPIO21_ IN	GPIO20_ IN	GPIO19_ IN	GPIO18_ IN	GPIO17_ IN	GPIO16_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO23_ IN	GPIO22_ IN	GPIO21_ IN	GPIO20_ IN	GPIO19_ IN	GPIO18_ IN	GPIO17_ IN	GPIO16_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[23:16]_IN	When read, returns the status of the pin

9.1.12 GPIO C Output Register (GPIOCOUT)

Location		7	6	5	4	3	2	1	0
7F1FH	Read	GPIO23_ OUT	GPIO22_ OUT	GPIO21_ OUT	GPIO20_ OUT	GPIO19_ OUT	GPIO18_ OUT	GPIO17_ OUT	GPIO16_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[23:16]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.13 GPIO C Function Select Register (GPIOCSEL)

Location		7	6	5	4	3	2	1	0
7FDCH	Read	-	GPIO22_ SEL	GPIO21_ SEL	GPIO20_ SEL	GPIO19_ SEL	GPIO18_ SEL	GPIO17_ SEL	GPIO16_ SEL
	Write								
	Reset	X	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO22_SEL	1: EC_SCI# (ACPI EC channel 0 interrupt output) 0: GPIO22 function
GPIO21_SEL	1: SMI# (System Management Interrupt output) 0: GPIO21 function
GPIO20_SEL	1: LED4 output 0: GPIO20 function
GPIO19_SEL	1: LED3 output 0: GPIO19 function
GPIO18_SEL	1: LED2 output 0: GPIO18 function
GPIO17_SEL	1: LED1 output 0: GPIO17 function
GPIO16_SEL	1: LED0 output 0: GPIO16 function

9.1.14 GPIO D Direction Register (GPIODDIR)

Location		7	6	5	4	3	2	1	0
7F22H	Read	-	-	-	-	GPIO27_ DIR	GPIO26_ DIR	GPIO25_ DIR	GPIO24_ DIR
	Write								
	Reset	X	X	X	X	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO[27:24]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.15 GPIO D Input Register (GPIODIN)[1]

Location		7	6	5	4	3	2	1	0
7F24H	Read	GPIO31_IN	GPIO30_IN	GPIO29_IN	GPIO28_IN	GPIO27_IN	GPIO26_IN	GPIO25_IN	GPIO24_IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO31_IN	GPIO30_IN	GPIO29_IN	GPIO28_IN	GPIO27_IN	GPIO26_IN	GPIO25_IN	GPIO24_IN

1. When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[31:24]_IN	When read, returns the status of the pin.

9.1.16 GPIO D Output Register (GPIODOUT)

Location		7	6	5	4	3	2	1	0
7F23H	Read Write	GPIO31_OUT	GPIO30_OUT	GPIO29_OUT	GPIO28_OUT	GPIO27_OUT	GPIO26_OUT	GPIO25_OUT	GPIO24_OUT
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[31:24]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.17 GPIO D Function Select Register (GPIODSEL)

Location		7	6	5	4	3	2	1	0
7FDDH	Read Write	GPIO31_SEL	GPIO30_SEL	GPIO29_SEL	GPIO28_SEL	GPIO27_SEL	GPIO26_SEL	GPIO25_SEL	GPIO24_SEL
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO31_SEL	1: PSDAT1 (PS/2 channel 1 data pin) 0: GPIO31 function
GPIO30_SEL	1: PSCLK1 (PS/2 channel 1 clock pin) 0: GPIO30 function
GPIO29_SEL	1: PSDAT0 (PS/2 channel 0 data pin) 0: GPIO29 function
GPIO28_SEL	1: PSCLK0 (PS/2 channel 0 clock pin) 0: GPIO28 function
GPIO27_SEL	1: SDA2 (SMBus 2 data pin) 0: GPIO27 function
GPIO26_SEL	1: SCL2 (SMBus 2 clock pin) 0: GPIO26 function
GPIO25_SEL	1: FAN2 (Tachometer FAN 2 input) 0: GPIO25 function
GPIO24_SEL	1: FAN1 (Tachometer FAN 1 input) 0: GPIO24 function

9.1.18 GPIO E Input Register (GPIOEIN)[1]

Location		7	6	5	4	3	2	1	0
7FA2H	Read	GPIO39_ IN	GPIO38_ IN	GPIO37_ IN	GPIO36_ IN	GPIO35_ IN	GPIO34_ IN	GPIO33_ IN	GPIO32_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO39_ IN	GPIO38_ IN	GPIO37_ IN	GPIO36_ IN	GPIO35_ IN	GPIO34_ IN	GPIO33_ IN	GPIO32_ IN

[1] When using an open drain pin as an input, output data must be specified as '1'. However, specifying output data as '0' when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[39:32]_IN	When read, returns the status of the pin.

9.1.19 GPIO E Output Register (GPIOEOUT)

Location		7	6	5	4	3	2	1	0
7FA1H	Read Write	GPIO39_ OUT	GPIO38_ OUT	GPIO37_ OUT	GPIO36_ OUT	GPIO35_ OUT	GPIO34_ OUT	GPIO33_ OUT	GPIO32_ OUT
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[39:32]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.20 GPIO E Function Select Register (GPIOESEL)

Location		7	6	5	4	3	2	1	0
7FDEH	Read Write	GPIO39_ SEL	GPIO38_ SEL	GPIO37_ SEL	GPIO36_ SEL	GPIO35_ SEL	GPIO34_ SEL	GPIO33_ SEL	GPIO32_ SEL
	Reset	1	1	0	0	0	0	0	0

Symbol	Function
GPIO39_SEL	1: CLKRUN# function (PCI Clock Control signal) 0: GPIO39 function
GPIO38_SEL	1: LPCPD# input (LPC power down signal) 0: GPIO38 function
GPIO37_SEL	1: KSO15 output (Keyboard scan output 15) 0: GPIO37 function
GPIO36_SEL	1: KSO14 output (Keyboard scan output 14) 0: GPIO36 function
GPIO35_SEL	1: KSO13 output (Keyboard scan output 13) 0: GPIO35 function
GPIO34_SEL	1: KSO12 output (Keyboard scan output 12) 0: GPIO34 function
GPIO33_SEL	1: PSDAT2 (PS/2 channel 2 data pin) 0: GPIO33 function
GPIO32_SEL	1: PSCLK2 (PS/2 channel 2 clock pin) 0: GPIO32 function

9.1.21 GPIO F Direction Register (GPIOFDIR)

Location		7	6	5	4	3	2	1	0
7FA3H	Read	GPIO47_ DIR	GPIO46_ DIR	-	-	-	GPIO42_ DIR	GPIO41_ DIR	-
	Write								
	Reset	0	0	X	X	X	0	0	X

Symbol	Function
-	Not implemented
X	Not defined
GPIO[47:46]_DIR	GPIO direction control bit 1: Output 0: Input
GPIO[42:41]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.22 GPIO F Input Register (GPIOFIN)[1]

Location		7	6	5	4	3	2	1	0
7FA5H	Read	GPIO47_ IN	GPIO46_ IN	GPIO45_ IN	GPIO44_ IN	GPIO43_ IN	GPIO42_ IN	GPIO41_ IN	GPIO40_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO47_ IN	GPIO46_ IN	GPIO45_ IN	GPIO44_ IN	GPIO43_ IN	GPIO42_ IN	GPIO41_ IN	GPIO40_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[47:40]_IN	When read, returns the status of the pin.

9.1.23 GPIO F Output Register (GPIOFOUT)

Location		7	6	5	4	3	2	1	0
7FA4H	Read	GPIO47_ OUT	GPIO46_ OUT	-	-	-	GPIO42_ OUT	GPIO41_ OUT	GPIO40_ OUT
	Write								
	Reset	1	1	X	X	X	1	1	1

Symbol	Function
-	Not implemented
X	Not defined
GPIO[47:40]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.24 GPIO F Function Select Register (GPIOFSEL)

Location		7	6	5	4	3	2	1	0
7FA6H	Read	GPIO47_ SEL	GPIO46_ SEL	GPIO45_ SEL	-	-	-	-	GPIO40_ SEL
	Write								
	Reset	0	0	0	X	X	X	X	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO47_SEL	1: T2 (Timer2 counter input or output) 0: GPIO47 function
GPIO46_SEL	1: T1 (Timer1 counter input or output) 0: GPIO46 function
GPIO45_SEL	1: T0 (Timer0 counter input) 0: GPI45 function
GPIO40_SEL	1: EC1_SCI (ACPI EC channel 1 interrupt output) 0: GPIO40 function

9.1.25 GPIO G Direction Register (GPIOGDIR)

Location		7	6	5	4	3	2	1	0
7F39H	Read	GPIO55_ DIR	GPIO54_ DIR	GPIO53_ DIR	GPIO52_ DIR	GPIO51_ DIR	GPIO50_ DIR	GPIO49_ DIR	GPIO48_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[55:48]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.26 GPIO G Input Register (GPIOGIN)[1]

Location		7	6	5	4	3	2	1	0
7F3BH	Read	GPIO55_ IN	GPIO54_ IN	GPIO53_ IN	GPIO52_ IN	GPIO51_ IN	GPIO50_ IN	GPIO49_ IN	GPIO48_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO55_ IN	GPIO54_ IN	GPIO53_ IN	GPIO52_ IN	GPIO51_ IN	GPIO50_ IN	GPIO49_ IN	GPIO48_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[55:48]_IN	When read, returns the status of the pin.

9.1.27 GPIO G Output Register (GPIOGOUT)

Location		7	6	5	4	3	2	1	0
7F3AH	Read	GPIO55_ OUT	GPIO54_ OUT	GPIO53_ OUT	GPIO52_ OUT	GPIO51_ OUT	GPIO50_ OUT	GPIO49_ OUT	GPIO48_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[55:48]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.28 GPIO G Function Select Register (GPIOGSEL)

Location		7	6	5	4	3	2	1	0
7F3CH	Read	-	GPIO54_ SEL	GPIO53_ SEL	GPIO52_ SEL	GPIO51_ SEL	GPIO50_ SEL	GPIO49_ SEL	GPIO48_ SEL
	Write								
	Reset	X	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO54_SEL	1: SCL1 (SMBus 1 clock)/TXD (UART transmit output) selected by UART_SM bit 0: GPIO54 function
GPIO53_SEL	1: SDA1 (SMBus 1 data)/RXD (UART receive input) selected by UART_SM bit 0: GPIO53 function
GPIO52_SEL	1: SCL0 (SMBus 0 clock pin) 0: GPIO52 function
GPIO51_SEL	1: SDA0 (SMBus 0 data pin) 0: GPIO51 function
GPIO50_SEL	1: 32KCLKOUT (32.768KHz clock signal output) 0: GPIO50 function.
GPIO49_SEL	1: CLKOUT (8051 core clock output) 0: GPIO49 function.
GPIO48_SEL	1: WDOGOUT (Watchdog timer output) 0: GPIO48 function.

9.1.29 GPIO H Direction Register (GPIOHDIR)

Location		7	6	5	4	3	2	1	0
7FE0H	Read	GPIO63_ DIR	GPIO62_ DIR	GPIO61_ DIR	GPIO60_ DIR	GPIO59_ DIR	GPIO58_ DIR	GPIO57_ DIR	GPIO56_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[63:56]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.30 GPIO H Input Register (GPIOHIN)[1]

Location		7	6	5	4	3	2	1	0
7FE2H	Read	GPIO63_IN	GPIO62_IN	GPIO61_N	GPIO60_IN	GPIO59_IN	GPIO58_IN	GPIO57_IN	GPIO56_IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO63_IN	GPIO62_IN	GPIO61_N	GPIO60_IN	GPIO59_IN	GPIO58_IN	GPIO57_IN	GPIO56_IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[63:56]_IN	When read, returns the status of the pin.

9.1.31 GPIO H Output Register (GPIOHOUT)

Location		7	6	5	4	3	2	1	0
7FE1H	Read	GPIO63_OUT	GPIO62_OUT	GPIO61_OUT	GPIO60_OUT	GPIO59_OUT	GPIO58_OUT	GPIO57_OUT	GPIO56_OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[63:56]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.32 GPIO HL Open-Drain/Push-Pull Section Register (GPIOHLOD)

Location		7	6	5	4	3	2	1	0
7FABH	Read	GPIO63_OD	GPIO62_OD	GPIO61_OD	GPIO60_OD	GPIO91_OD	GPIO90_OD	GPIO89_OD	GPIO88_OD
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[63:60]_OD	GPIO output buffer configuration control bit 1: Open-Drain configuration 0: Push-Pull configuration
GPIO[91:88]_OD	GPIO output buffer configuration control bit 1: Open-Drain configuration 0: Push-Pull configuration

9.1.33 GPIO I Direction Register (GPIOIDIR)

Location		7	6	5	4	3	2	1	0
7FE3H	Read	GPIO71_DIR	GPIO70_DIR	GPIO69_DIR	GPIO68_DIR	GPIO67_DIR	GPIO66_DIR	GPIO65_DIR	GPIO64_DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[71:64]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.34 GPIO I Input Register (GPIOIIN)[1]

Location		7	6	5	4	3	2	1	0
7FE5H	Read	GPIO71_IN	GPIO70_IN	GPIO69_IN	GPIO68_IN	GPIO67_IN	GPIO66_IN	GPIO65_IN	GPIO64_IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO71_IN	GPIO70_IN	GPIO69_IN	GPIO68_IN	GPIO67_IN	GPIO66_IN	GPIO65_IN	GPIO64_IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[67:64]_IN	When read, returns the status of the pin.
GPIO[71:68]_IN	When read, returns the status of the pin in normal GPIO mode only.

9.1.35 GPIO I Output Register (GPIOIOUT)

Location		7	6	5	4	3	2	1	0
7FE4H	Read	GPIO71_OUT	GPIO70_OUT	GPIO69_OUT	GPIO68_OUT	GPIO67_OUT	GPIO66_OUT	GPIO65_OUT	GPIO64_OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[71:64]_OUT	When written to, output data is updated. When read, returns previously written data

9.1.36 GPIO I Function Select Register (GPIOISEL)

Location		7	6	5	4	3	2	1	0
7FF0H	Read	GPIO71_SEL	GPIO70_SEL	GPIO69_SEL	GPIO68_SEL	-	-	-	-
	Write								
	Reset	0	0	0	0	X	X	X	X

Symbol	Function
-	Not implemented
X	Not Defined
GPIO71_SEL	1: DAC3 (Digital to Analog converter channel 3 output) 0: GPIO71 function
GPIO70_SEL	1: DAC2 (Digital to Analog converter channel 2 output) 0: GPIO70 function
GPIO69_SEL	1: DAC1 (Digital to Analog converter channel 1 output) 0: GPIO71 function
GPIO68_SEL	1: DAC0 (Digital to Analog converter channel 0 output) 0: GPIO70 function

9.1.37 GPIO I Open-Drain/Push-Pull Section Register (GPIOIOD)

Location		7	6	5	4	3	2	1	0
7FACH	Read Write	GPIO71_ OD	GPIO70_ OD	GPIO69_ OD	GPIO68_ OD	GPIO67_ OD	GPIO66_ OD	GPIO65_ OD	GPIO64_ OD
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[71:64]_OD	GPIO output buffer configuration control bit. 1: Open-Drain configuration 0: Push-Pull configuration

9.1.38 GPIO J Direction Register (GPIOJDIR)

Location		7	6	5	4	3	2	1	0
7FE6H	Read Write	GPIO79_ DIR	GPIO78_ DIR	GPIO77_ DIR	GPIO76_ DIR	GPIO75_ DIR	GPIO74_ DIR	GPIO73_ DIR	GPIO72_ DIR
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:72]_DIR	GPIO direction control bit 1: Output 0: Input

9.1.39 GPIO J Input Register (GPIOJIN)[1]

Location		7	6	5	4	3	2	1	0
7FE8H	Read	GPIO79_ IN	GPIO78_ IN	GPIO77_ IN	GPIO76_ IN	GPIO75_ IN	GPIO74_ IN	GPIO73_ IN	GPIO72_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO79_ IN	GPIO78_ IN	GPIO77_ IN	GPIO76_ IN	GPIO75_ IN	GPIO74_ IN	GPIO73_ IN	GPIO72_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[79:72]_IN	When read, returns the status of the pin in normal GPIO mode only.

9.1.40 GPIO J Output Register (GPIOJOUT)

Location		7	6	5	4	3	2	1	0
7FE7H	Read Write	GPIO79_ OUT	GPIO78_ OUT	GPIO77_ OUT	GPIO76_ OUT	GPIO75_ OUT	GPIO74_ OUT	GPIO73_ OUT	GPIO72_ OUT
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[79:72]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.41 GPIO J Function Select Register (GPIOJSEL)

Location		7	6	5	4	3	2	1	0
7FF5H	Read	GPIO79_ SEL	GPIO78_ SEL	GPIO77_ SEL	GPIO76_ SEL	GPIO75_ SEL	GPIO74_ SEL	GPIO73_ SEL	GPIO72_ SEL
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:72]_SEL	1: ACH[0:7] Analog to Digital converter channel 0 to 7
	0: GPIO[79:72] function

9.1.42 GPIO J Open-Drain/Push-Pull Section Register (GPIOJOD)

Location		7	6	5	4	3	2	1	0
7FADH	Read	GPIO79_ OD	GPIO78_ OD	GPIO77_ OD	GPIO76_ OD	GPIO75_ OD	GPIO74_ OD	GPIO73_ OD	GPIO72_ OD
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[79:72]_OD	GPIO output buffer configuration control bit
	1: Open-Drain configuration
	0: Push-Pull configuration

9.1.43 GPIO K Direction Register (GPIOKDIR)

Location		7	6	5	4	3	2	1	0
7FE9H	Read	GPIO87_ DIR	GPIO86_ DIR	GPIO85_ DIR	GPIO84_ DIR	GPIO83_ DIR	GPIO82_ DIR	GPIO81_ DIR	GPIO80_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[87:80]_DIR	GPIO direction control bit
	1: Output
	0: Input

9.1.44 GPIO K Input Register (GPIOKIN)[1]

Location		7	6	5	4	3	2	1	0
7FEBH	Read	GPIO87_ IN	GPIO86_ IN	GPIO85_ IN	GPIO84_ IN	GPIO83_ IN	GPIO82_ IN	GPIO81_ IN	GPIO80_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO87_ IN	GPIO86_ IN	GPIO85_ IN	GPIO84_ IN	GPIO83_ IN	GPIO82_ IN	GPIO81_ IN	GPIO80_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[87:80]_IN	When read, returns the status of the pin.

9.1.45 GPIO K Output Register (GPIOKOUT)

Location		7	6	5	4	3	2	1	0
7FEAH	Read	GPIO87_ OUT	GPIO86_ OUT	GPIO85_ OUT	GPIO84_ OUT	GPIO83_ OUT	GPIO82_ OUT	GPIO81_ OUT	GPIO80_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[87:80]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.46 GPIO K Pull-up Control Register (GPIOKPU)

Location		7	6	5	4	3	2	1	0
7FB7H	Read	-	-	GPIO85 PU	GPIO84 PU	GPIO83 PU	GPIO82 PU	GPIO81 PU	GPIO80 PU
	Write								
	Reset	X	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
GPIO[85:80]PU	GPIO Internal Pull-up control bit 1: Enable Internal Pull-up 0: Disable Internal Pull-up

9.1.47 GPIO L Direction Register (GPIOLDIR)

Location		7	6	5	4	3	2	1	0
7FA7H	Read	GPIO95_ DIR	GPIO94_ DIR	GPIO93_ DIR	GPIO92_ DIR	GPIO91_ DIR	GPIO90_ DIR	GPIO89_ DIR	GPIO88_ DIR
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[95:88]_DIR	GPIO direction control bit 1: Output 0: Input When GPIO[95:94]_DIR=0, GPIO[95:94] are inputs with internal pull-ups enabled

9.1.48 GPIO L Input Register (GPIOLIN)[1]

Location		7	6	5	4	3	2	1	0
7FA9H	Read	GPIO95_ IN	GPIO94_ IN	GPIO93_ IN	GPIO92_ IN	GPIO91_ IN	GPIO90_ IN	GPIO89_ IN	GPIO88_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO95_ IN	GPIO94_ IN	GPIO93_ IN	GPIO92_ IN	GPIO91_ IN	GPIO90_ IN	GPIO89_ IN	GPIO88_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[95:88]_IN	When read, returns the status of the pin.

9.1.49 GPIO L Output Register (GPIOLOUT)

Location		7	6	5	4	3	2	1	0
7FA8H	Read	GPIO95_ OUT	GPIO94_ OUT	GPIO93_ OUT	GPIO92_ OUT	GPIO91_ OUT	GPIO90_ OUT	GPIO89_ OUT	GPIO88_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[95:88]_OUT	When written to, output data is updated. When read, returns previously written data.

9.1.50 GPIO M Pullup Control Register (GPIOMPU)

Location		7	6	5	4	3	2	1	0
7F5BH	Read	GPIO103 _PU	GPIO102 _PU	GPIO101 _PU	GPIO100 _PU	GPIO99_ PU	GPIO98_ PU	GPIO97_ PU	GPIO96_ PU
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[103:96]_PU	GPIO Internal Pullup control bit 1: Disable Internal Pullup 0: Enable Internal Pullup

9.1.51 GPIO M Input Register (GPIOMIN)[1]

Location		7	6	5	4	3	2	1	0
7F6BH	Read	GPIO103_ IN	GPIO102_ IN	GPIO101_ IN	GPIO100_ IN	GPIO99_ IN	GPIO98_ IN	GPIO97_ IN	GPIO96_ IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO103_ IN	GPIO102_ IN	GPIO101_ IN	GPIO100_ IN	GPIO99_ IN	GPIO98_ IN	GPIO97_ IN	GPIO96_ IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[103:96]_IN	When read, returns the status of the pin.

9.1.52 GPIO M Output Register (GPIOMOUT)

Location		7	6	5	4	3	2	1	0
7F30H	Read	GPIO103 _OUT	GPIO102 _OUT	GPIO101 _OUT	GPIO100 _OUT	GPIO99_ OUT	GPIO98_ OUT	GPIO97_ OUT	GPIO96_ OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[103:96]_OUT	When written to, output data is updated (see Table 9-2 below). When read, returns previously written data.

9.1.53 GPIO M Open-Drain/Push-Pull Section Register (GPIOMOD)

Location		7	6	5	4	3	2	1	0
7FF6H	Read	GPIO103 _OD	GPIO102 _OD	GPIO101 _OD	GPIO100 _OD	GPIO99_ OD	GPIO98_ OD	GPIO97_ OD	GPIO96_ OD
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[103:96]_OD	GPIO output buffer configuration control bit (see Table 9-2 below)

9.1.54 GPIO N Pullup Control Register (GPIONPU)

Location		7	6	5	4	3	2	1	0
7F83H	Read	GPIO111 _PU	GPIO110 _PU	GPIO109 _PU	GPIO108 _PU	GPIO107 _PU	GPIO106 _PU	GPIO105 _PU	GPIO104 _PU
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[111:104]_PU	GPIO Internal Pullup control bit 1: Disable Internal Pullup 0: Enable Internal Pullup

9.1.55 GPIO N Input Register (GPIONIN)[1]

Location		7	6	5	4	3	2	1	0
7F84H	Read	GPIO111 _IN	GPIO110 _IN	GPIO109 _IN	GPIO108 _IN	GPIO107 _IN	GPIO106 _IN	GPIO105 _IN	GPIO104 _IN
	Write	-	-	-	-	-	-	-	-
	Reset	GPIO111 _IN	GPIO110 _IN	GPIO109 _IN	GPIO108 _IN	GPIO107 _IN	GPIO106 _IN	GPIO105 _IN	GPIO104 _IN

[1] When using an open drain pin as an input, output data must be specified as ‘1’. However, specifying output data as ‘0’ when using an open drain pin as an input will damage the part.

Symbol	Function
-	Not implemented
GPIO[111:104]_IN	When read, returns the status of the pin.

9.1.56 GPIO N Output Register (GPIONOUT)

Location		7	6	5	4	3	2	1	0
7F82H	Read	GPIO111 _OUT	GPIO110 _OUT	GPIO109 _OUT	GPIO108 _OUT	GPIO107 _OUT	GPIO106 _OUT	GPIO105 _OUT	GPIO104 _OUT
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
GPIO[111:104]_OUT	When written to, output data is updated (see Table 9-2). When read, returns previously written data.

9.1.57 GPIO N Open-Drain/Push-Pull Section Register (GPIONOD)

Location		7	6	5	4	3	2	1	0
7FFCH	Read Write	GPIO111 _OD	GPIO110 _OD	GPIO109 _OD	GPIO108 _OD	GPIO107 _OD	GPIO106 _OD	GPIO105 _OD	GPIO104 _OD
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
GPIO[111:104]_OD	GPIO output buffer configuration control bit (see Table 9-2 below)

TABLE 9-2: GPIO96-GPIO111 Input/Output configuration control

GPIOn_OD	GPIOn_OUT	Input/Output
0	0	Output ‘0’
0	1	Input / Output ‘1’ open drain
1	0	Output ‘0’
1	1	Output ‘1’ push pull

T9-2.0 1320

10.0 TIMERS/COUNTERS, WATCHDOG TIMER AND PWM

10.1 Timers: T0, T1, T2

The SST79LF008 device has three 16-bit registers that can be used as either timers or event counters. The three timers/counters are denoted Timer 0 (T0), Timer 1 (T1), and Timer 2 (T2). Each is designated a pair of 8-bit registers in the Special Function Registers (SFRs). The pair of registers consists of a most significant (high) byte and least significant (low) byte. The respective registers are TL0, TH0, TL1, TH1, TL2, and TH2. See Table 10-4. Timer 2 also has the capture registers RCAP2L and RCAP2H. See Table 10-4.

10.2 Timer Operations

Refer to Section 10.3 for full description of the TCON, TMOD, T2CON and T2MOD registers that control timer operations.

10.2.1 Timer 1 and Timer 0

Timer 1 and Timer 0 operations are controlled by TMOD and TCON registers. Each timer can be configured to operate either as a timer or event counter depending on the value of the bits C/T#_T1 and C/T_/T0 in TMOD register. The clock source for timer function is 8051 core clock CCLK divided by 12. The clock source for the event counter function is either the T1 or T0 input pin respectively, or crystal oscillator clock XCLK as selected by CLKCON register (active falling edge for any source). Both the T1 and T0 timers count up. Each timer can be turned ON by setting, or turned OFF by clearing, the TR1/TR0 bit in TCON register.

There are four operating modes available in either timer or counter operations. In Modes 0, 1 and 2 both T0 and T1 operate similarly. In Mode 3, T0 and T1 operations are different. Table 10-1 and Table 10-2 provide the examples of TMOD values to be used to select operation modes for timers T1 and T0.

10.2.1.1 Mode 0

In Mode 0 each timer is configured as a 13-bit timer, which includes 8-bit counter (TH1/TH0) and a 5-bit prescaler (lower 5 bits of TL1/TL0). As the count overflows from all 1s to all 0s, the counter continues to count, and the respective timer overflow flag TF1/TF0 is set.

10.2.1.2 Mode 1

Mode 1 is similar to Mode 0, with the exception that each timer uses full 16-bit counter. The clock is applied to the combined high and low timer registers TH1:TL1/TH0:TL0.

10.2.1.3 Mode 2

In Mode 2 each timer is configured as an 8-bit counter with automatic reload. Timer 1 uses TL1 register as a counter; when overflow occurs, bit TF1 is set, and TL1 is reloaded with the contents of register TH1. Timer 0 uses TL0 register as a counter; when overflow occurs, bit TF0 is set, and TL0 is reloaded with the contents of register TH0. The reload does not modify TH1/TH0 value.

10.2.1.4 Mode 3

Timer 1 in Mode 3 is halted and holds its count. Timer 0 in Mode 3 is divided into two separate 8-bit counters TL0 and TH0. Timer 0 control bits: C/T#_T0, TR0, and TF0 are dedicated to TL0 operations only. Timer 1 control bits: TR1 and TF1 are dedicated to TH0 operations only, and TH0 is forced into timer mode which uses CCLK divided by 12 as a clock source.

TABLE 10-1: Timer 0 Operating Modes

	Mode	Function	TMOD[1]
Used as Timer	0	13-bit Timer	00H
	1	16-bit Timer	01H
	2	8-bit Auto-Reload	02H
	3	Two 8-bit Timers	03H
Used as Counter	0	13-bit Counter 8-bit Counter TH0 with TL0 as 5-bit prescaler	04H
	1	16-bit Counter	05H
	2	8-bit Auto-Reload	06H
	3	Two 8-bit Counters	07H

T10-1.0 1320

[1] The Timer is turned ON/OFF by setting/clearing bit TR0.

TABLE 10-2: Timer 1 Operating Modes

	Mode	Function	TMOD[1]
Used as Timer	0	13-bit Timer	00H
	1	16-bit Timer	10H
	2	8-bit Auto-Reload	20H
	3	Stopped	30H
Used as Counter	0	13-bit Counter 8-bit Counter TH1 with TL1 as 5-bit prescaler	40H
	1	16-bit Counter	50H
	2	8-bit Auto-Reload	60H
	3	Not Available	-

T10-2.0 1320

1. The Timer is turned ON/OFF by setting/clearing bit TR1.

10.2.2 Timer 2

Similar to Timer 1 and 0, Timer 2 can operate either as a timer or as an event counter, depending on the value of bit C/T2# in T2CON register. Timer 2 has three operating modes: capture, auto-reload, and baud rate generator. Refer to Table 10-3 examples of the respective settings.

10.2.2.1 16-bit Timer/Counter Capture Mode

In the capture mode, the EXEN2 bit in T2CON selects one of two options. If EXEN2 bit is cleared to 0, then Timer 2 is a 16-bit timer/counter. When the timer/counter overflow occurs, Timer 2 overflow bit TF2 will be set. If EXEN2 bit is set to 1, then Timer 2 operates the same way, but in addition a falling edge on the external input T2EX causes the current value in the Timer 2 registers TL2 and TH2 to be captured into the RCAP2L and RCAP2H registers respectively. The T2EX falling edge also sets the EXF2 bit in T2CON. Either TF2 or EXF2 flags can generate Timer 2 interrupt to 8051.

10.2.2.2 16-bit Timer/Counter Auto-reload Mode

In the auto-reload mode, the EXEN2 bit in T2CON also selects one of two options. If EXEN2 is cleared to 0, then when the 16-bit timer/counter overflow occurs. TF2 bit is set, and Timer 2 registers are reloaded with the contents of the RCAP2L and RCAP2H registers. If EXEN2 is set to 1, then Timer 2 operates the same way as above, but, in addition a falling edge on the external input, T2EX also triggers the counter reload and sets EXF2 bit. Either TF2 or EXF2 flags can generate Timer 2 interrupt to 8051.

10.2.2.3 Baud Rate Generator Mode

When RCLK or TCLK bit is set to 1, T2 output signal determines UART baud rates for receive and/or transmit as described in Section 11.0. This mode is similar to auto-reload mode with the following exceptions: TF2 flag is not set on overflow, and T2EX falling edge does not cause a reload.

TABLE 10-3: Timer 2 Operating Modes

		T2CON[1]	
	Mode	Internal Control[2]	External Control[3]
Used as Timer	16-bit Auto Reload	00H	08H
	16-bit Capture	01H	09H
	Baud rate generator Receive and Transmit	30H	38H
	Receive only	20H	28H
	Transmit only	10H	18H
Used as Counter	16-bit Auto Reload	02H	0AH
	16-bit Capture	03H	0BH

T10-3.0 1320

1. The Timer is turned ON/OFF by setting/clearing bit TR2.
2. Capture/Reload occurs only on timer or counter overflow.
3. Capture/Reload occurs on timer or counter overflow and a 1 to 0 transition on T2EX pin except when Timer 2 is used in the baud rate generating mode. The GPIO15 pin must be set as T2EX.

10.3 Timers/Counters SFRs

TABLE 10-4: Timer/Counters SFRs

Symbol	Description	Direct Address	Symbol								RESET Value
			MSB							LSB	
TMOD	Timer/Counter Mode Control	89H	Timer 1 Control bits				Timer 0 Control bits				00H
			GATE _T1	C/ T#_T1	M1_ T1	M0_ T1	GATE _T0	C/ T#_T0	M1_ T0	M0_ T0	
TCON[1]	Timer/Counter Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TH0	Timer 0 MSB	8CH	TH0[7:0]								00H
TL0	Timer 0 LSB	8AH	TL0[7:0]								00H
TH1	Timer 1 MSB	8DH	TH1[7:0]								00H
TL1	Timer 1 LSB	8BH	TL1[7:0]								00H
T2CON[1]	Timer / Counter 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN 2	TR2	C/T2#	CP/ RL2#	00H
T2MOD#	Timer 2 Mode Control	C9H	-	-	-	-	-	T1OE	T2OE	DCEN	X0H
TH2	Timer 2 MSB	CDH	TH2[7:0]								00H
TL2	Timer 2 LSB	CCH	TL2[7:0]								00H
RCAP2H	Timer 2 Capture MSB	CBH	RCAP2H[7:0]								00H
RCAP2L	Timer 2 Capture LSB	CAH	RCAP2L[7:0]								00H

T10-4.0 1320

[1] Bit x Addressable SFRs -8.0 555

10.3.1 Timer / Counter Control Register (TCON)

Location		7	6	5	4	3	2	1	0
88H	Read	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
	Write					-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt service routine, or it can be cleared in software.
TR1	Timer 1 run control bit. Set/cleared by software to turn on/off Timer/Counter
TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt service routine, or can be cleared in software.
TR0	Timer 0 run control bit. Set/cleared by software to turn on/off Timer/Counter
IE1	Interrupt INT1 request flag This is read only bit equal to the INT1 signal (see Figure 8-1 for INT1 signal sources)
IT1	Interrupt 1 type control bit. Always cleared. 0: INT1 is a level triggered interrupt
IE0	Interrupt INT0 request flag This is a read only bit equal to the INT0 signal (see Figure 8-1 for INT0 signal sources)
IT0	Interrupt 0 type control bit. Always cleared. 0: INT0 is a level triggered interrupt

10.3.2 Timer / Counter Mode Register (TMOD)

Location		7	6	5	4	3	2	1	0
89H	Read	GATE_T1	C/T#_T1	M1_T1	M0_T1	GATE_T0	C/T#_T0	M1_T0	M0_T0
	Write	-				-			
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
GATE_T1	Timer 1 gating control bit. Always cleared. 0: Gate function is disabled and Timer 1 is controlled by TR1 bit only.
C/T#_T1	Timer or Counter Selector bit (Timer 1). 1: Counter operation (input clock is selected by CLKCON register). 0: Timer operation (input clock frequency is FCCLK/12).
M1_T1	Mode bit 1 for T1.
M0_T1	Mode bit 0 for T1.
GATE_T0	Timer 0 gating control bit. Always cleared. 0: Gate function is disabled and Timer 0 is controlled by TR0 bit only.
C/T#_T0	Timer or Counter Selector bit (Timer 0) 1: Counter operation (input clock is selected by CLKCON register). 0: Timer operation (input clock frequency is FCCLK/12).
M1_T0	Mode bit 1 for T0.
M0_T0	Mode bit 0 for T0.

TABLE 10-5: Timer Operating Mode as a Function of Mode Bits

M1_Tn[1]	M0_Tn[1]	Mode	Operating mode
0	0	0	13-bit Timer Mode. 8-bit Timer/Counter THn with TLn as 5-bit prescaler.
0	1	1	16-bit Timer Mode. THn and TLn are cascaded into 16-bit Timer/Counter with no prescaler.
1	0	2	8-bit Auto Reload Mode. THn holds a value, which has to be reloaded into 8-bit auto-reload Timer/Counter TL1 each time it overflows.
1	1	3 (T1)	Timer/Counter 1 is stopped.
1	1	3 (T0)	TL0 is an 8-bit Timer/Counter controlled by the Timer 0 control bits. TH0 is an 8-bit Timer only controlled by Timer 1 control bits.

T10-5.1320

[1] n=0,1

10.3.3 Timer/Counter 2 Control Register (T2CON)

Location		7	6	5	4	3	2	1	0
C8H	Read	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set in baud rate generator mode (when RCLK or TCLK = 1) and in clock output mode (when T20E = 1 and C/T2# = 0).
EXF2	Timer 2 external flag set when either a capture or reload is caused by a falling edge on T2EX and EXEN2 = 1, and must be cleared by software. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock flag. When set, causes the UART to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock flag. When set, causes the UART to use Timer 2 overflow pulses to transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflow to be used for the transmit clock.
EXEN2	Timer 2 external enable control bit. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the UART. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/stop control for Timer 2. A logic ‘1’ starts the Timer 2. A logic ‘0’ stops Timer 2
C/T2#	Timer or counter select (Timer 2) 1: External event counter (T2 input pin falling edge-triggered) 0: Internal timer input clock frequency is F_{CCLK}/12 when RCLK = 0 and TCLK = 0 input clock frequency is F_{CCLK}/2 when RCLK = 1 or TCLK = 1
CP/RL2#	Capture/Reload flag. When set, captures occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads occur either when Timer 2 overflows or with 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.

10.3.4 Timer/Counter 2 Mode Control (T2MOD)

Location		7	6	5	4	3	2	1	0
C9H	Read	-	-	-	-	-	T1OE	T2OE	DCEN
	Write								
	Reset	X	X	X	X	X	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
T1OE	Timer 1 Output Enable bit 1: Enable Timer 1 output. Bit C/T#_T1 must be cleared. A 50% duty cycle will be output. T1OE bit does not affect T1 overflow interrupt 0: Disable Timer 1 output
T2OE	Timer 2 Output Enable bit 1: Enable Timer 2 output. Bit C/T2# must be cleared. A 50% duty cycle will be output. If T2OE = 1, TF2 overflow flag will not be set on Timer 2 roll-over, and T2 overflow interrupt will not be generated. If T2OE = 0, TF2 overflow interrupt is not affected. 0: Disable Timer 2 output
DCEN	Timer 2 Down Count Enable bit 1: Timer 2 can be configured as an up or down counter. The T2EX pin controls the count direction. A logic ‘1’ at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH-to-0000H transition and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into timer registers TH2 and TL2 respectively. Logic ‘0’ at T2EX makes Timer 2 count down. In this case the timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes FFFFH to be reloaded into the timer registers. The EXF2 bit toggles whenever Timer 2 overflows or underflows. This bit can be used as a 17th bit of resolution if desired. In this operating mode, EXF2 does not generate an interrupt. 0: Timer 2 is configured as up counter only

10.4 Watchdog Timer (WDT)

A watchdog timer (WDT) is a hardware timer that offers protection against software and/or hardware deadlock during normal operation. When the WDT is enabled, it will generate a chip reset or interrupt if the user program does not reload the WDTDAT register within a specified time interval. The frequency of the XCLK clock source for the WDT is 32.768 KHz. The WDT has a 9-bit prescaler, which makes the watchdog timer resolution equal to approximately 16.0ms. Hence, the 8-bit watchdog counter provides a watchdog time interval from 16ms to 4 seconds. Both the watchdog counter and the prescaler are reset when the unit is disabled.

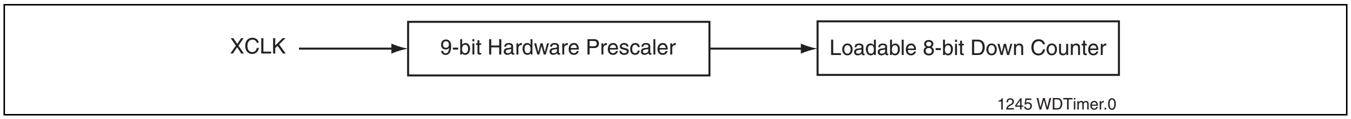


FIGURE 10-1: Watchdog Timer

After POR, BOR, External reset, WDT reset, or aLPC Soft reset the WDT is disabled. The WDT commences counting down from the loaded WDT data value as soon as enable bit WDTEN in WDTCSR register is set.

Once activated, the WDT must be reloaded periodically in software before the programmed WDT interval expires. Otherwise, the WDT will underflow and a WDT reset or interrupt will be generated.

WDT reset is controlled by the WDTRSTEN bit in RSTCON register, and WDT interrupt is controlled by the WDTMSK bit in INTSRCAMSK register. Both WDT reset and WDT interrupt will generate an output signal on GPIO48 if it is selected as WDT output pin. When WDT reset is enabled (WDTRSTEN = 1), this output will be asserted for one 32KHz clock cycle. When WDT reset is disabled, and WDT interrupt is enabled (WDTMSK = 1), this output will stay asserted until the interrupt is cleared; WDT will stay in underflow state and will not wrap around. When data register is reloaded or WDT is disabled by clearing WDTEN = 0, the underflow state is exited. On exit from underflow state WDT interrupt is cleared, and WDT output is deasserted. It is possible to prevent both WDT reset and WDT interrupt from happening when WDTRSTEN = WDTMSK = 0.

There are two methods to disable the WDT operation at run time: (1) clear the WDT enable bit directly, or (2) reload the WDT data register with 00H, which will automatically clear WDT enable bit.

10.4.1 Watchdog Timer MMCRs

10.4.1.1 Watchdog Timer Control / Status Register (WDTCSR)

Location		7	6	5	4	3	2	1	0
7F37H	Read	WDT- STOPEN	-	-	-	-	-	WDTEN	WLE
	Write								
	Reset	0	X	X	X	X	X	0	0

Symbol	Function
-	Not implemented
X	Not defined
WDTSTOPEN	WDT Power Down mode operation control bit 1: WDT keeps running when 8051 enters into Power Down mode provided WDT is enabled. 0: WDT is stopped when 8051 enters into Power Down mode. WDT automatically resumes running after Power Down exit staring with the default value 0FFH.
WDTEN	WDT Enable bit. Set by firmware to enable (start) the WDT. Automatically cleared when loading 00H to WDTDAT register. 1: Enable WDT operation 0: Disable WDT operation
WLE	Watchdog Load Enable bit. Set by firmware to enable writing to the WDTDAT register. Automatically cleared when writing any data to the WDTDAT register. 1: Enable writes to WDTDAT register (WDT is reloaded) 0: Disable writes to the WDTDAT register (write is ignored).

10.4.1.2 Watchdog Data Register (WDTDAT)

Location		7	6	5	4	3	2	1	0
7F38H	Read	WDTDAT7	WDTDAT6	WDTDAT5	WDTDAT4	WDTDAT3	WDTDAT2	WDTDAT1	WDTDAT0
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
WDTDAT[7:0]	WDT data When written to and WLE = 1, the WDT timer is initialized (counter is loaded with the respective data byte, and prescaler is reset). When read, it will return the current value. The watchdog timer (WDT) must be reloaded within the periods that are shorter than the programmed watchdog interval; otherwise the WDT will underflow.

Note: Recommended WDT start software sequence: set WLE = 1 first, next load WDTDAT, and finally set WDTEN = 1

10.5 Hibernation Timer

SST79LF008 has a hibernation timer which allows for hibernation time of up to 127.5 minutes in 30-second intervals. The XCLK clock source for the hibernation timer is derived from a 32.768 KHz crystal. In order to generate a 30-second time interval, there is a 20-bit prescaler which counts up from 0 to F0000H and then wraps around to zero. The hibernation time counter is a down-counter which does not wrap around after reaching zero value. Interrupt and wakeup events are generated when hibernation timer reaches zero value. When a non-zero value is written into HIBER register hibernation timer is re-started and the respective interrupt is cleared. The relationship between the prescaler and the down counter can be seen in Figure 10-2 below.

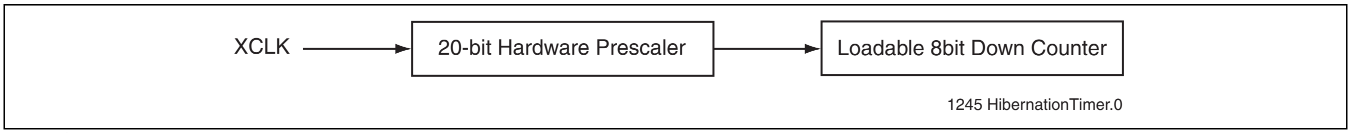


FIGURE 10-2: Hibernation Timer

10.5.1 Hibernation Timer Register (HIBER)

Location		7	6	5	4	3	2	1	0
7FF3H	Read	HIB7	HIB6	HIB5	HIB4	HIB3	HIB2	HIB1	HIB0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
HIB[7:0]	Hibernation time in 30 second intervals 00H: Hibernation time = 0 seconds 01H: Hibernation time = 30 seconds 02H: Hibernation time = 1 minute 03H: Hibernation time = 1.5 minutes... Continues to increment in 30 second intervals until FFH: Hibernation time = 127.5 minutes

10.6 Pulse Width Modulators (PWM)

The SST79LF008 device has three independent PWM channels. Each of them has 16-bit prescaler, 8-bit cycle time register, and 8-bit duty cycle control register. The PWM clock can be derived from 8051 core clock CCLK which causes the frequency to be $F_{PWM} = F_{CCLK}$ or from 32.768 KHz crystal oscillator clock XCLK which causes the frequency to be F_{PWM} = 32.768KHz (selected by PWM control register).

The prescaler output frequency is determined as $F_{PWMP} = F_{PWM}$ / ((PWMPHn:PWMPLn)+1). The PWM output cycle time can be found as T_{PWMC} = (PWMCn+1) / $F_{PWMP,}$ and the PWM output Duty Cycle = (PWMDn+1) / (PWMCn+1) x100%. The PWM Duty Cycle specifies the fraction of the PWM cycle for which the output signal is high. In the case where PWMDn = PWMCn the PWM output is always high. If PWMDn > PWMCn, then the PWM output is always low.

The device also includes an additional PWM timer with five "555-like" 300ms interval "blinking" outputs for LED control. The clock source for this PWM timer is 32.768 KHz XCLK signal.

10.6.1 PWM MMCRs

10.6.1.1 PWM Channel 0 Prescaler Register Low Byte (PWMPL0)

Location		7	6	5	4	3	2	1	0
7F25H	Read	PWMPL0 _7	PWMPL0 _6	PWMPL0 _5	PWMPL0 _4	PWMPL0 _3	PWMPL0 _2	PWMPL0 _1	PWMPL0 _0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.2 PWM Channel 1 Prescaler Register Low Byte (PWMPL1)

Location		7	6	5	4	3	2	1	0
7F26H	Read	PWMPL1 _7	PWMPL1 _6	PWMPL1 _5	PWMPL1 _4	PWMPL1 _3	PWMPL1 _2	PWMPL1 _1	PWMPL1 _0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.3 PWM Channel 2 Prescaler Register Low Byte (PWMPL2)

Location		7	6	5	4	3	2	1	0
7F29H	Read	PWMPL2 _7	PWMPL2 _6	PWMPL2 _5	PWMPL2 _4	PWMPL2 _3	PWMPL2 _2	PWMPL2 _1	PWMPL2 _0
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	**Function**
PWMPLn[7:0]	PWM channel 0-2 prescaler low byte. When PWMPLn (n=0-2) register is modified, the PWMn output can be unpredictable for no more than one PWMn cycle. The output frequency of the prescaler is F_{PWM}/((PWMPHn:PWMPLn)+1).

10.6.1.4 PWM Channel 0 Prescaler Register High Byte (PWMPH0)

Location		7	6	5	4	3	2	1	0
7F97H	Read	PWMPH0_7	PWMPH0_6	PWMPH0_5	PWMPH0_4	PWMPH0_3	PWMPH0_2	PWMPH0_1	PWMPH0_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.5 PWM Channel 1 Prescaler Register High Byte (PWMPH1)

Location		7	6	5	4	3	2	1	0
7F98H	Read	PWMPH1_7	PWMPH1_6	PWMPH1_5	PWMPH1_4	PWMPH1_3	PWMPH1_2	PWMPH1_1	PWMPH1_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.6 PWM Channel 2 Prescaler Register High Byte (PWMPHn)

Location		7	6	5	4	3	2	1	0
7F99H	Read	PWMPH2_7	PWMPH2_6	PWMPH2_5	PWMPH2_4	PWMPH2_3	PWMPH2_2	PWMPH2_1	PWMPH2_0
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
PWMPHn[7:0]	PWM channel 0-2 prescaler high byte. When PWMPHn (n=0-2) register is modified, the PWMn output can be unpredictable for no more than one PWMn cycle. The output frequency of the prescaler is F_{PWM}/((PWMPHn:PWMPLn)+1).

10.6.1.7 PWM Channel 0 Cycle Time Register (PWMC0)

Location		7	6	5	4	3	2	1	0
7F9AH	Read	PWMC0_7	PWMC0_6	PWMC0_5	PWMC0_4	PWMC0_3	PWMC0_2	PWMC0_1	PWMC0_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.8 PWM Channel 1 Cycle Time Register (PWMC1)

Location		7	6	5	4	3	2	1	0
7F9BH	Read	PWMC1_7	PWMC1_6	PWMC1_5	PWMC1_4	PWMC1_3	PWMC1_2	PWMC1_1	PWMC1_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.9 PWM Channel 2 Cycle Time Register (PWMC2)

Location		7	6	5	4	3	2	1	0
7F9CH	Read	PWMC2_7	PWMC2_6	PWMC2_5	PWMC2_4	PWMC2_3	PWMC2_2	PWMC2_1	PWMC2_0
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
PWMCn[7:0]	PWM channel 0-2 cycle time value. The 8-bit down-counter divides the prescaler output frequency by (PWMCn+1). When PWMCn (n=0-2) register is modified, the PWMn output can be unpredictable for no more than one PWMn cycle.

10.6.1.10 PWM Channel 0 Duty Cycle Time Register (PWMD0)

Location		7	6	5	4	3	2	1	0
7F9DH	Read	PWMD0_7	PWMD0_6	PWMD0_5	PWMD0_4	PWMD0_3	PWMD0_2	PWMD0_1	PWMD0_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.11 PWM Channel 1 Duty Cycle Time Register (PWMD1)

Location		7	6	5	4	3	2	1	0
7F9EH	Read	PWMD1_7	PWMD1_6	PWMD1_5	PWMD1_4	PWMD1_3	PWMD1_2	PWMD1_1	PWMD1_0
	Write								
	Reset	1	1	1	1	1	1	1	1

10.6.1.12 PWM Channel 2 Duty Cycle Time Register (PWMD2)

Location		7	6	5	4	3	2	1	0
7F9FH	Read	PWMD2_7	PWMD2_6	PWMD2_5	PWMD2_4	PWMD2_3	PWMD2_2	PWMD2_1	PWMD2_0
	Write								
	Reset	1	1	1	1	1	1	1	1

Symbol	Function
PWMDn[7:0]	PWM channel 0-2 duty cycle value. This value defines the number of prescaler clocks for which PWM output is high. When PWMDn (n=0-2) register is modified, the new duty cycle will be in effect starting with the next PWMn cycle. Duty Cycle = ((PWMDn+1) / (PWMCn+1)) x 100% If PWMDn=PWMCn, then PWM output is always high (Duty Cycle = 100%) If PWMDn > PWMCn PWM output is always low (Duty Cycle = 0%)

10.6.1.13 PWM Control Register (PWMCR)

Location		7	6	5	4	3	2	1	0
7F96H	Read	PWM4_LED1	PWM4_LED0	PWM3_LED1	PWM3_LED0	-	PWM2_SEL	PWM1_SEL	PWM0_SEL
	Write								
	Reset	0	0	0	0	X	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PWM[2:0]_SEL	PWM channel 2-0 clock source Selection bit 1: Select XCLK - F_{PWM} = 32.768KHz, PWM keeps running when 8051 is in Power Down mode. 0: Select 8051 core clock CCLK - F_{PWM} = F_{CCLK}. PWM is stopped when 8051 is in Power Down mode.
PWM[4:3]_LED[1:0]	LED 4-3 output control bits

10.6.1.14 PWM 555 Like LED Control Register (PWM555CR1)

Location		7	6	5	4	3	2	1	0
7F21H	Read	-	PWM2_LED1	PWM2_LED0	PWR-GOOD	PWM1_LED1	PWM1_LED0	PWM0_LED1	PWM0_LED0
	Write				-				
	Reset	X	0	0	PWR-GOOD	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
PWM[2:0]_LED[1:0]	LED 2-0 output control bits
PWRGOOD	Status of POWERGOOD Pin (reset value = pass through pin state).

TABLE 10-6: LED 4/2/0 Behavior

Control Bits:	Output Behavior
PWM4_LED[1:0] PWM2_LED[1:0] PWM0_LED[1:0]	LED4 LED2 LED0
00	Output high (LED is Off)
01	Output low = 0.125s, Output period = 1.0s
10	Output low =0.125s, Output period = 0.5s
11	Output low (LED is On)

T10-6.1320

TABLE 10-7: LED 3/1 Behavior

Control Bits:	Output Behavior
PWM3_LED[1:0] PWM1_LED[1:0]	LED3 LED1
00	Output high (LED is Off)
01	Output low = 0.125s, Output period = 3.0s
10	Output low =0.125, Output period = 1.5s
11	Output low (LED is On)

T10-7.1320

Note: 555, like PWM, uses 32.768KHz and runs in Power Down mode.

11.0 SERIAL I/O PORT (UART)

11.1 Full-Duplex, Enhanced UART

The SST79LF008 serial I/O port is a full-duplex, enhanced UART port that uses the transmit and receiver registers to simultaneously transmit and receive data in the hardware while the software is performing other tasks. 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, while reading from the SBUF register obtains the contents of the receive register. The enhanced UART features framing error detection and automatic address recognition.

The UART has four modes of operation which are selected by the Serial Port Mode selection bits (SM0 and SM1) 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) SFR 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.

11.2 Framing Error Detection

Framing Error Detection allows checking for valid stop bit during receive operation in serial modes 1, 2, and 3. An incorrect stop bit could be caused by noise in the serial line, simultaneous transmissions by two CPUs, or mismatched baud rates between transmitter and receiver. Serial mode 0 does not permit Framing Error Detection because it employs a synchronous serial protocol that does not use stop bit.

Framing Error Detection is selected in the PCON register by setting SMOD0 = 1, see Figure 11-1. If a stop bit is missing, the Framing Error bit (FE) at SCON[7] will be set. The software examines the FE bit after each received bit to check for data errors. After the FE bit is set, it can only be cleared by software. Valid stop bits do not clear the FE bit. When Framing Error Detection is enabled, RI rises on the stop bit, instead of the last data bit. See Figures 11-2 and 11-3.

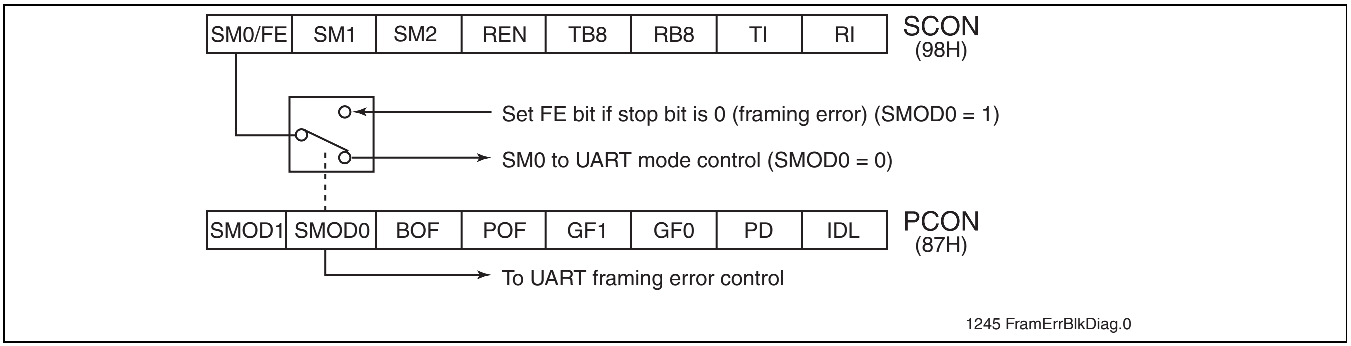

SM0/FE
SM1
SM2
REN
TB8
RB8
TI
RI
SCON
(98H)
Set FE bit if stop bit is 0 (framing error) (SMOD0 = 1)
SM0 to UART mode control (SMOD0 = 0)
SMOD1
SMOD0
BOF
POF
GF1
GF0
PD
IDL
PCON
(87H)
To UART framing error control
1245 FramErrBlkDiag.0

FIGURE 11-1: Framing Error Block Diagram

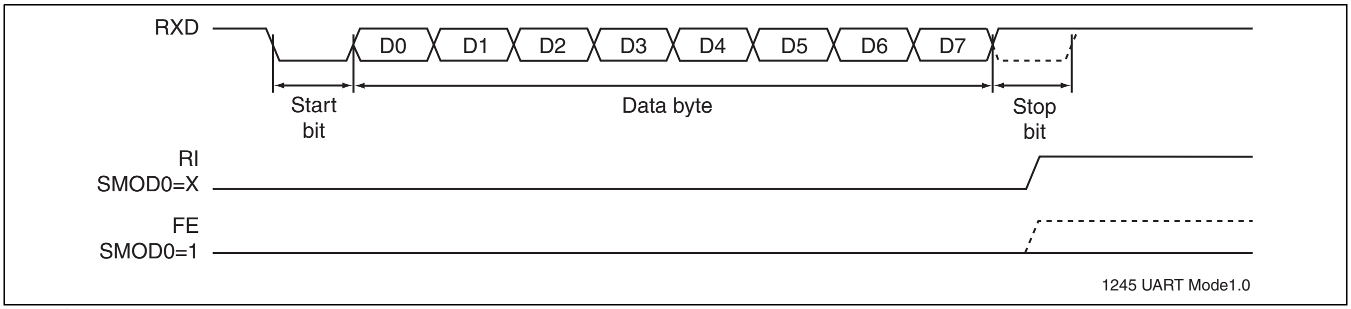


FIGURE 11-2: UART Timings in Mode 1

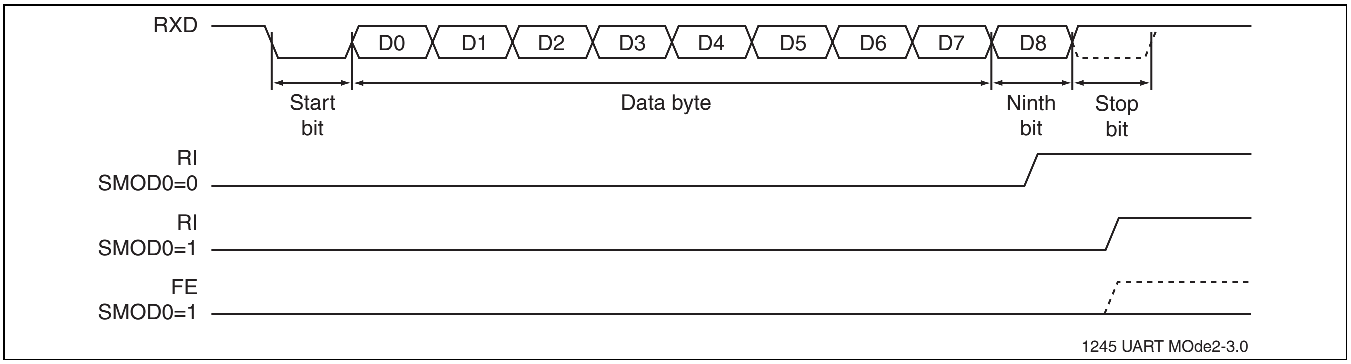


FIGURE 11-3: UART Timings in Modes 2 and 3

11.3 Automatic Address Recognition

Automatic Address Recognition helps reduce the time and power required to interface and communicate with multiple serial devices. All connected devices share the same serial link, and each device has its own address. In this configuration, a device is only interrupted when it receives its own address, group address, or broadcast address. This process eliminates the software overhead to compare addresses.

Automatic Address Recognition saves power by working in conjunction with the idle mode, thereby reducing the system's overall power consumption. Since there may be multiple slaves connected serially to one master, only one slave would have to be interrupted from the idle mode to respond to the master's transmission. During this transmission, Automatic Address Recognition (AAR) allows all the other slaves to remain in idle mode. Limiting the number of interruptions reduces the total current drawn from the system.

There are two ways to communicate with slaves—by group or all at once. To communicate with a group of slaves, the master sends out an address called the "given address". To communicate with all the slaves, the master sends out an address called the "broadcast" address.

Enable AAR in mode 2 or 3 (9-bit modes) by setting the SM2 bit in SCON. Each slave has its own SM2 bit set waiting for an address byte (9th bit = 1). The Receive Interrupt (RI) flag will only be set when the received byte matches either the given address or the broadcast address. Next, the slave then clears its SM2 bit to enable reception of the data bytes (9th bit = 0) from the master. When the 9th bit = 1, the master is sending an address. When the 9th bit = 0, the master is sending the actual data.

If mode 1 is used, the stop bit takes the place of the 9th bit. Bit RI is set only when the received command frame address matches the device's address, and is terminated by a valid stop bit. Note that AAR is not available in mode 0. Setting SM2 bit in the SCON register in mode 0 will have no effect.

11.3.1 Using the Given Address to Select Slaves

Given addresses are used to address an individual slave or a group of slaves. A slave may have one or more given addresses because of "don't care" bits. The given address is computed by a logical AND operation of the SADDR value and the SADEN value. Any bit masked off by a 0 from SADEN becomes a "don't care" represented by 'X' in the example below.

Slave 1			
SADDR	=	1111	0001
SADEN	=	1111	1010
GIVEN	=	1111	0X0X

Slave 2			
SADDR	=	1111	0011
SADEN	=	1111	1001
GIVEN	=	1111	0XX1

Because of the "don't care" bits, multiple slaves may respond to a given address as shown in Table 11-1. Continuing the above example, slave 1 has an individual address of 11110001 loaded into SADDR. The SADEN byte has been used to mask off bits for a given address to allow more combinations of selecting slave 1 and 2. In this case for the given addresses, the last bit (LSB) of slave 1 is a "don't care" and the last bit slave 2 is a 1. Similarly for slave 2, the second to last bit is a "don't care" and that of slave 1 is a 0. Thus to communicate with slave 1 and 2, the master would need to send an address with the last two bits equal to 01 (e.g. 1111 0001). To communicate with slave 1 only, the last two bits must be 00 (e.g. 1111 0000).

TABLE 11-1: Possible Addresses for Slaves 1 and 2

Target Slave	Given Address	All Possible Addresses
Slave 1 only	1111 0X0X	1111 0000 1111 0100
Slave 2 only	1111 0XX1	1111 0111 1111 0011
Slave 1 and 2	1111 0XXX	1111 0001 1111 0101

T11-1.0 1320

If the user added a third slave, the slave 3 example in Table 11-2 provides its Given Address calculation, and Table 11-2 indicates all possible addresses for Slave 3 and the combination of Slaves 2 and 3.

Slave 3

SADDR	=	1111	1001
SADEN	=	1111	0101
GIVEN	=	1111	X0X1

TABLE 11-2: Possible Addresses for Slave 3 and Slave 2/3 Combination

Target Slave	Given Address	All Possible Addresses
Slave 3 only	1111 X0X1	1111 1011 1111 1001
Slave 2 and 3 only	1111 XXX1	1111 0011

T11-2.0 1320

11.3.2 Using the Broadcast Address to Select Slaves

Using the Broadcast Address, the master can communicate with all the slaves at once. The broadcast address is formed by performing a logical OR of SADDR and SADEN with zeros in the result treated as “don’t cares”.

Slave 1

	1111 0001	= SADDR
+	1111 1010	= SADEN
	1111 1X11	= Broadcast

“Don’t cares” allow for a wider range in defining the Broadcast Address, but in most cases, the Broadcast Address will be FFH.

On reset, SADDR and SADEN are ‘0’. This produces a Given Address of all “don’t cares” as well as a Broadcast Address of all “don’t cares”. This effectively disables Automatic Addressing mode.

11.4 UART SFRs

11.4.1 Power Control Register (PCON)

Location		7	6	5	4	3	2	1	0
87H	Read Write	SMOD1	SMOD0	BOF[1]	POF[1]	GF1	GF0	PD	IDL
	Reset	0	0	0	1	0	0	0	0

[1] These bits are reset by Power-On reset only (all other reset events have no affect)

Symbol	Function
SMOD1	Double baud rate bit. If SMOD1 = 1, Timer 1 is used to generate the baud rate, and the serial port is used in modes 1, 2, and 3, then baud rate is doubled.
SMOD0	FE bit / SM0 bit selection control bit. 1: SCON[7] = FE bit 0: SCON[7] = SM0 bit
BOF	Brown-out detection status flag. Set when brown-out is detected, cleared by software. This bit is also cleared after Power-On reset. It is not affected by any other reset event.
POF	Power-on reset status flag. Set after POR, cleared by software. This bit is not affected by any other reset event.
GF1	General-purpose flag bit
GF0	General-purpose flag bit
PD	Power Down mode bit. Setting this bit activates Power Down mode (refer to Figure 5-3). It is cleared by hardware after exiting Power Down mode by one of the reset (see Table 5-1) or enabled wakeup event (see Figure 8-1 through Figure 8-5). 1: Activates Power Down mode 0: Power Down mode is not activated
IDL	Idle mode bit. Setting this bit activates Idle mode (refer to Table 5-3). It is cleared by hardware after exiting from Idle mode by one of the reset (see Table 5-1) or enabled interrupt event, see Figure 8-1 through Figure 8-5. 1: Activates idle mode 0: Idle mode is not activated

11.4.2 Slave Address Register (SADDR)

Location		7	6	5	4	3	2	1	0
A9H	Read Write	SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SADDR[7:0]	UART slave address

11.4.3 Slave Address Mask Register (SADEN)

Location		7	6	5	4	3	2	1	0
B9H	Read Write	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SDAEN[7:0]	UART slave address mask

11.4.4 Serial Port UART Data Register (SBUF)

Location		7	6	5	4	3	2	1	0
99H	Read Write	SBUF7	SBUF6	SBUF5	SBUF4	SBUF3	SBUF2	SBUF1	SBUF0
	Reset	X	X	X	X	X	X	X	X

Symbol	Function
X	Not defined
SBUF[7:0]	UART data buffer

11.4.5 Serial Port Control Register (SCON)

Location		7	6	5	4	3	2	1	0
98H	Read Write	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
FE	Framing Error bit (set SMOD0 = 1 to access FE bit) 1: Framing Error. Set by receiver when invalid stop bit is detected, cleared by software 0: No framing error detected
SM0	Serial Port Mode Bit 0 (clear SMOD0 = 0 to access SM0 bit)
SM1	Serial Port Mode Bit 1
SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. In these modes, if SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast address. In Mode 1, if SM2 = 1 then RI will not be set unless a valid stop bit has been received. In Mode 0, SM2 should be 0.
REN	Enables serial reception 1: Enable reception. 0: Disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. This bit is set or cleared by software as desired.
RB8	In Modes 2 and 3, RB8 is the 9th data bit that has been received. In Mode 1, if SM2 = 0, RB8 is the stop bit that has been received. In Mode 0, RB8 is not used.
TI	Transmit interrupt flag. 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, in any serial transmission. Must be cleared by software.
RI	Receive interrupt flag. 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, in any serial reception (except see SM2). Must be cleared by software.

TABLE 11-3: Serial Port Mode Description

SM0	SM1	Mode	Description	Baud Rate[1]
0	0	0	Shift Register	FCCLK / 12
0	1	1	8-bit UART	Variable[2,3]
1	0	2	9-bit UART	FCCLK /16 (SMOD1 =1) or FCCLK /32 (SMOD1 = 0)
1	1	3	9-bit UART	Variable[2,3]

T11-3.0 1320

1. FCCLK = 8051 core clock frequency
2. Variable baud rate if T1 is used as rate generator = (2^{SMOD1})*(T1 overflow rate)/32
3. Variable baud rate if T2 is used as rate generator = (T2 overflow rate)/16

12.0 SERIAL PERIPHERAL INTERFACE (SPI)

12.1 SPI Features

- Master or slave operation
- 16 MHz bit frequency (max) master mode
- 8 MHz bit frequency (max) slave mode
- LSB first or MSB first data transfer
- Four programmable bit rates
- End of transmission interrupt (SPIF)
- Write collision flag protection (WCOL)
- Wake up from Idle (master and slave modes)
- Wake up from Power Down modes (slave mode only)

12.2 SPI Description

The serial peripheral interface (SPI) allows full-duplex high-speed synchronous data transfer between the SST79LF008 and the peripheral devices.

Figure 12-1 shows the correspondence between master and slave SPI devices. The SCK (GPIO5) pin is the clock output for master mode and input for slave mode.

The SST79LF008 does not output SS#. If the SST79LF008 is the master and there is only one slave device, the slave's SS# input can be tied low. If there is more than one slave, N GPIOs can be used to select N slaves. Another option is external generation of the SS# inputs of multiple slave devices.

When SST79LF008 master mode is selected, the SPI clock generator will start following a write to the SST79LF008 device SPI data register. The written data is then shifted out of the MOSI (GPIO3) pin into the MOSI pin of the slave device. Following a complete transmission of one byte of data, the SPI clock generator is stopped and the SPIF flag is set. An SPI interrupt request will be generated if the SPI Interrupt Enable bit (SPIE) is set.

When SST79LF008 slave mode is selected, an external master generates SCK clock and drives the Slave Select input pin SS# (GPIO6) low to select the SST79LF008 SPI module as a slave. If the Slave Select input pin has not been driven low, then the SST79LF008 SPI unit is not active and the MOSI port can also be used as an input port pin.

Clock Phase control bit (CPHA) and Clock Polarity (CPOL) control the phase and polarity of the SPI clock. Figures 12-2 and 12-3 show the transfer formats with four possible combinations of these two bits.

To wake up the SST79LF008 from IDLE, whether in master or slave mode, the following conditions must be met: The SPIF bit is set to '1' upon completion of the data transfer, the SPIE is '1', and EA (Enable Global Interrupt bit in interrupt enable register) is '1'. These conditions generate an interrupt that wakes the device from IDLE mode.

In slave mode only, the SST79LF008 wakes up from Power Down when CPHA = 0 or CPHA = 1. When CPHA = 0, a transition from high to low on SS# pin wakes up the device from Power Down mode. When CPHA = 1, the clock edges of SCK wakes up the device from Power Down mode.

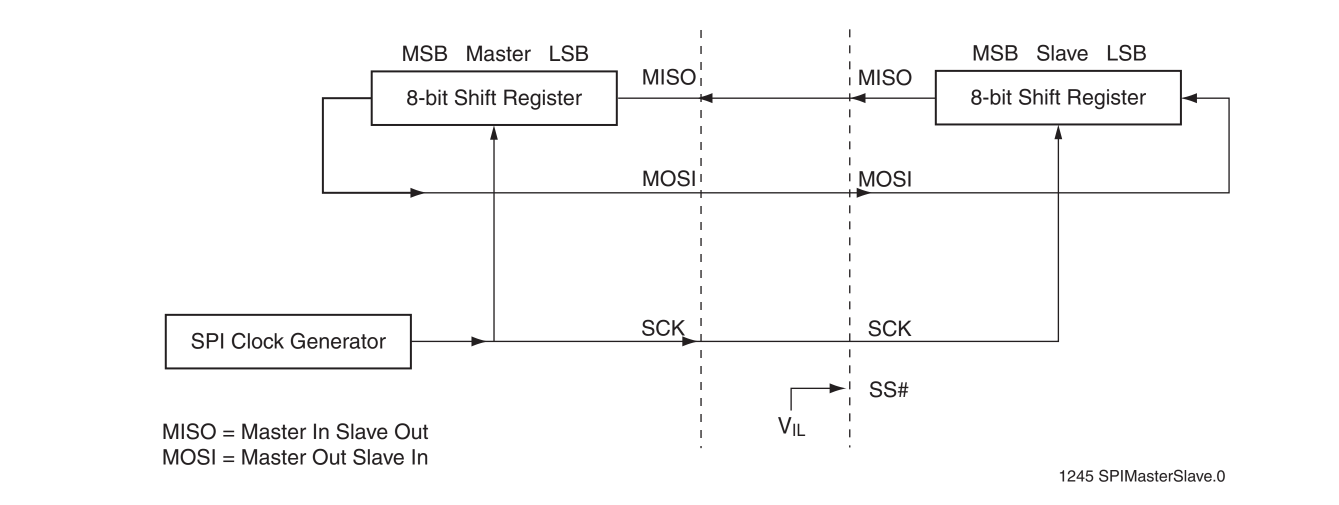


FIGURE 12-1: SPI Master-Slave Interconnection

12.3 SPI Transfer Formats

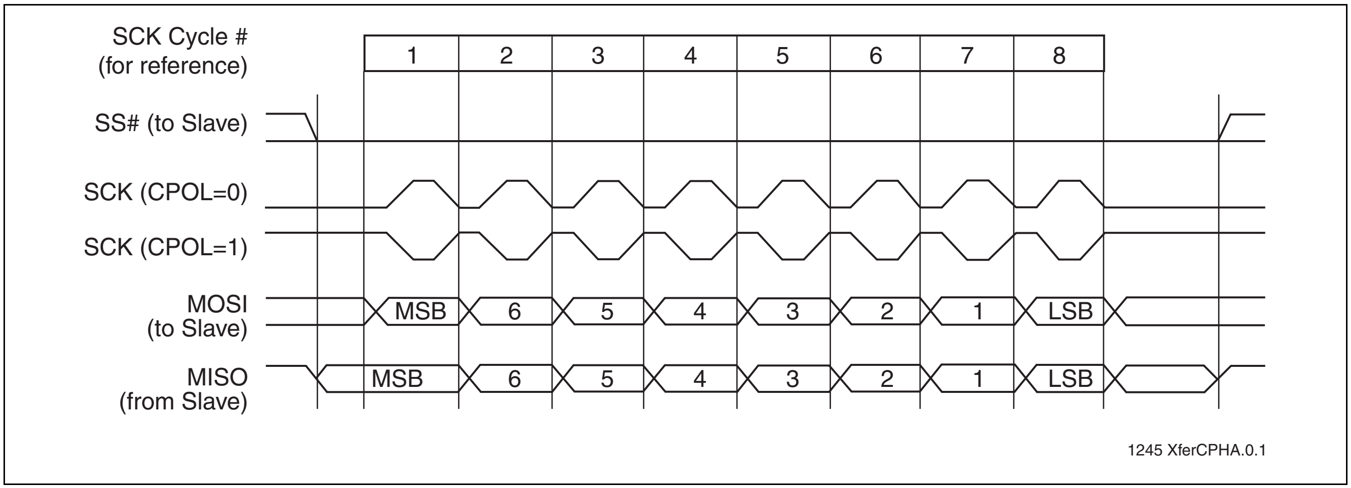


FIGURE 12-2: SPI Transfer Format (CPHA = 0)

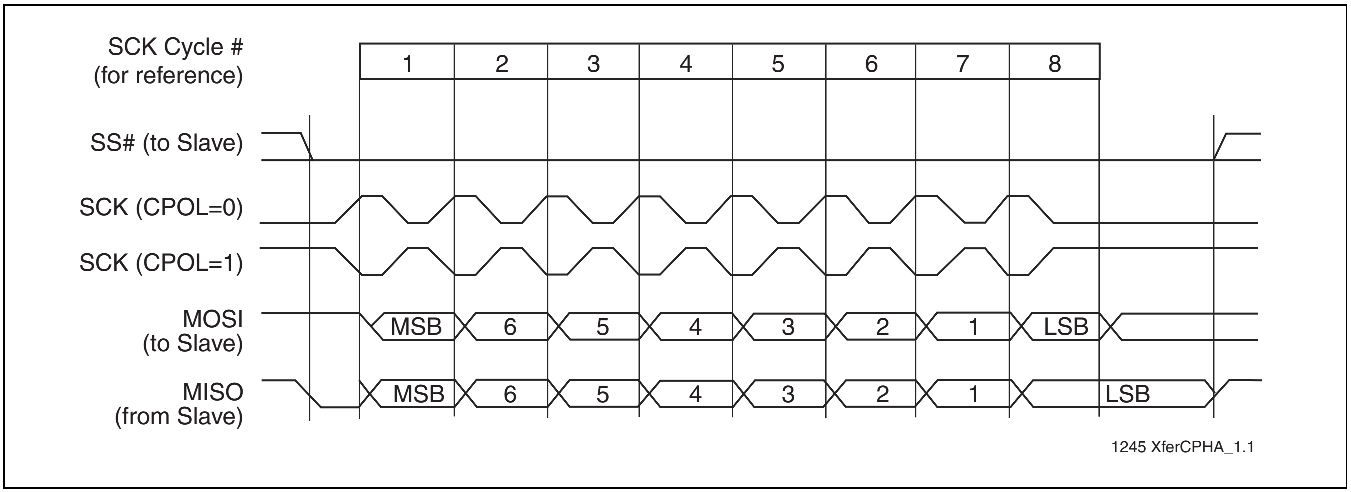


FIGURE 12-3: SPI Transfer Format (CPHA = 1)

12.4 SPI SFRs

12.4.1 SPI Control Register (SPCR)

Location		7	6	5	4	3	2	1	0
D5H	Read	SPIE	-	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
	Write								
	Reset	0	X	0	0	0	1	0	0

Symbol **Function**

- Not implemented

X Not defined

SPIE SPI end of transmission Interrupt Enable bit
1: Enable SPI interrupt
0: Disable SPI interrupt

DORD Data Transmission Order
1: LSB first in data transmission
0: MSB first in data transmission

MSTR Master/Slave select
1: Select Master mode
0: Select Slave mode

CPOL Clock Polarity
1: SCK is high when idle (Active Low)
0: SCK is low when idle (Active High)

CPHA Clock Phase control bit
1: Shift-in triggered on the trailing edge of the clock
0: Shift-in triggered on the leading edge of the clock

SPR1, SPR0 SPI Clock Rate Select bits
These two bits together with SPR2 bit in CLKCON register control the SCK rate of the device configured as a master. SPR2 bit has no effect on the slave.
The relationship between SCK and 8051 core clock frequency, F_{CCLK}, as shown in Tables 12-1 and 12-2.

TABLE 12-1: SCK Rate as a Function of SPI Clock Rate Select Bits (Master Only)

SPR2	SPR1	SPR0	SCK Frequency = FCCLK divided by
0	0	0	4
0	0	1	16
0	1	0	64
0	1	1	128
1	0	0	2
1	0	1	8
1	1	0	32
1	1	1	64

T12-1.1320

TABLE 12-2: SCK Rate as a Function of SPI Clock Rate Select Bits (Slave Only)

SPR1	SPR0	SCK Frequency = F_{CCLK} divided by
0	0	4
0	1	16
1	0	64
1	1	128

T12-2.

12.4.2 SPI Status Register (SPSR)

Location		7	6	5	4	3	2	1	0
AAH	Read	SPIF	WCOL	-	-	-	-	-	-
	Write								
	Reset	0	0	X	X	X	X	X	X

Symbol	Function
-	Not implemented
X	Not defined
SPIF	SPI Interrupt Flag. Set upon completion of data transfer. If SPIE = 1, an interrupt is then generated. Cleared by software. This bit is also automatically cleared by any access to SPDR after reading SPSR with SPIF=1.
WCOL	Write Collision Flag. Set if SPI data register is written during data transfer. Cleared by software. This bit is also automatically cleared by any access to SPDR after reading SPSR with WCOL=1.

12.4.3 SPI Data Register (SPDR)

Location		7	6	5	4	3	2	1	0
86H	Read	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SPD[7:0]	SPI data. When read, returns received data. When written to, the data is to be transmitted. Writing to this register during transfer will be ignored, and will set WCOL bit.

13.0 SMBUS INTERFACE

13.1 SMBus Features

- Compatible with SMBus 2.0 specification
- Two SMBus controllers and three SMBus channels
- Selection of SMBus channels through the internal multiplexer
- Support for SMbus master and slave operation
- Software-defined slave address and General Call address support
- Support for both polling and interrupt driven operation
- Wake up from Idle and Power Down—slave mode
- Wake up from Idle—master mode

13.2 SMBus Channels

The SST79LF008 includes two SMBus controllers, which support three separate two-wire SMBus channels. Each channel consists of Serial Data Line (SDAn, n=0-2) and the Serial Clock Line (SCLn, n=0-2). SMBus controller 0 controls the clock and data lines of the SMBus channel 0 (SCL0,SDA0), SMBus controller 1 controls via internal multiplexer the clock and data lines of SMBus channel 1 (SCL1, SDA1) and SMBus channel 2 (SCL2, SDA2). See Figure 13-1. SSCR, in Register 13.7.1, contains SM1_EN, SM0_EN, and CHSEL1

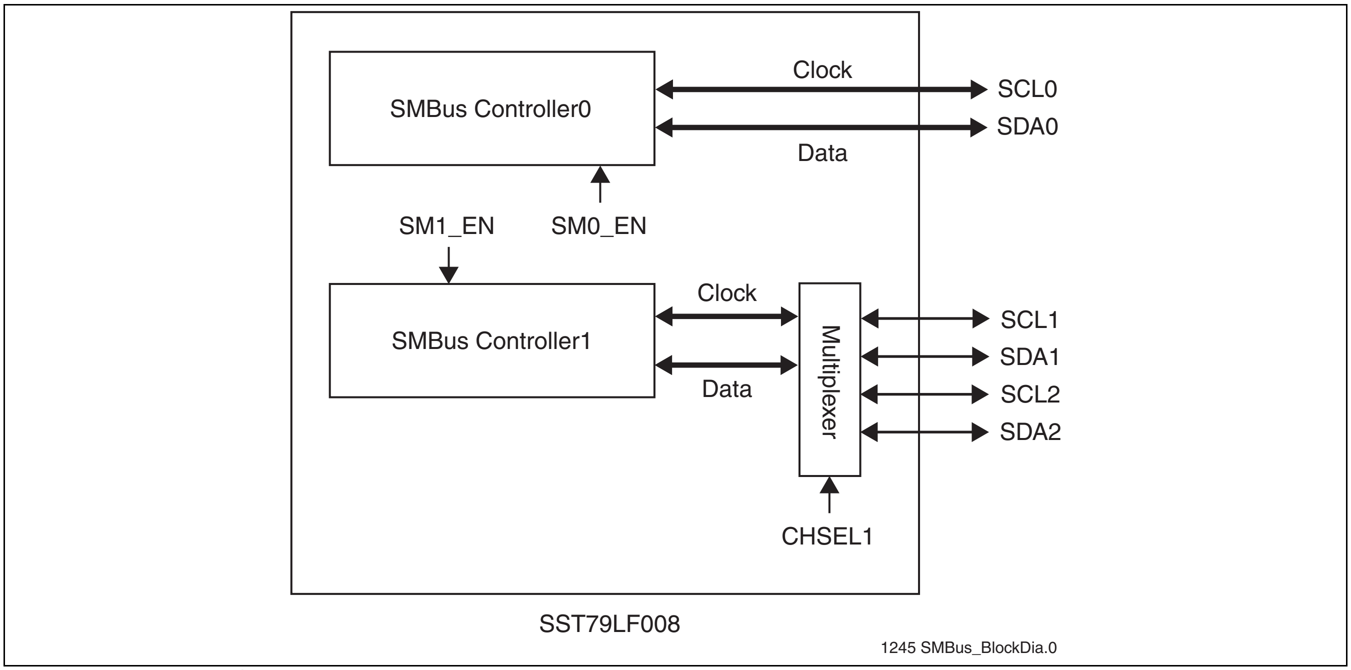


FIGURE 13-1: SMBus Module Block Diagram

13.3 SMBus Protocol Overview

SMBus is a bidirectional two-wire bus, which allows multiple master and slave devices to be connected simultaneously. All SMBus devices must have open drain outputs. The SMBus lines are connected externally to a positive voltage source (up to 5V) via pull-up circuits, and remain high when they are not driven by SMBus devices. The bus master device initiates the SMBus protocol, controls the bus clock, and terminates the transaction. The slave device responds and transmits the requested data back to the bus master device. Both master and slave devices on the bus can either operate as a transmitter or as a receiver.

Figures 13-2 and 13-3 illustrate bit and byte transfer layers of SMBus interface.

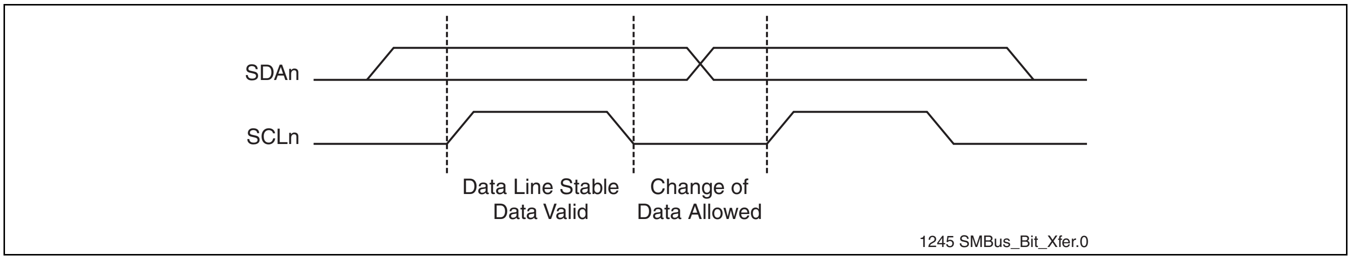


FIGURE 13-2: SMBus Relationship of SDA_n to SCL_n for Bit Transfer

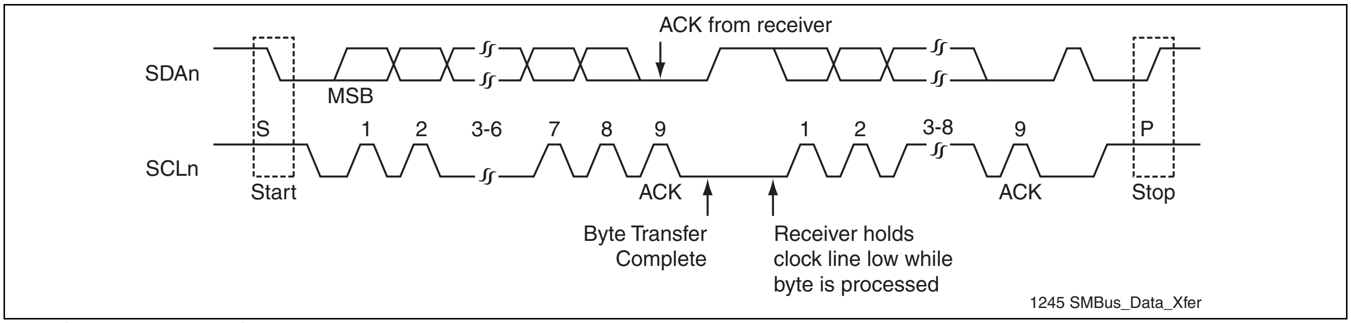


FIGURE 13-3: SMBus Byte Transfer

As shown on Figure 13-3, each SMBus transfer is initiated by the master with SMBus start condition (S), and terminated with the stop condition (P). The START condition is created by a high to low transition of the data line SDAn while the clock line SCLn is high. The STOP condition is created by a low to high transition of the data line SDAn while the clock line SCLn is high. The bus is considered to be “busy” after a START condition. The bus status will remain “busy” until a STOP condition is detected.

Between START and STOP condition, the SMBus protocol permits all information including address, command, and data to be transmitted on a serial data line, SDAn, synchronized with a serial clock SCLn. A single data bit is transferred per each SCLn clock pulse. As shown on Figure 13-2, throughout the SCLn clock’s high period, the data on SDAn line is kept stable by the transmitter, and can be sampled by the receiver. New data is sent by the transmitter only during the low state of the SCLn clock.

At each clock cycle, the slave can hold SCLn low for an extended period of time when the slave is still handling the previous data or is preparing a new data. There are typically two cases which may cause the slave to hold the bus: a byte received has not been processed, or the next byte to be transmitted is not ready. Under these circumstances, the slave will extend the clock low state.

The SMBus bytes are always transferred with the most significant bit first. An acknowledgement (ACK) cycle is appended at the end of each data byte (i.e., each byte transfer requires nine clock pulses with ACK being the ninth clock). The clock pulse for SMBus ACK clock cycle is always generated by the master device, but the ACK data bit is sent by the receiving device (master or slave). The transmitter must release the data line SDAn during ACK clock allowing the receiver to control the data line. The receiver asserts a low level on the data line during the ACK clock pulse to acknowledge that it has received the last data byte correctly.

If invalid command or data is detected or the receiver is busy, it may send a negative acknowledge (NACK) to indicate that it will not accept any additional data bytes. NACK is recognized when a high level is detected on the data line during the ACK clock pulse. There is an exception when NACK must be generated after a valid byte transfer. If the bus master is the receiver, it must indicate to the slave transmitter an end of data. It does this by responding with NACK to the last byte clocked out of the slave device.

Each slave device on the SMBus must have a unique slave address. The master transmits the address of the targeted slave device in the first seven bits after a START condition. The eighth bit, R/W#, specifies the direction of data transfer. See Figure 13-4. The slave device can operate as a transmitter or a receiver depending on the value of R/W# bit (1= slave transmitter, 0=slave receiver). The slave device must always acknowledge its own address.

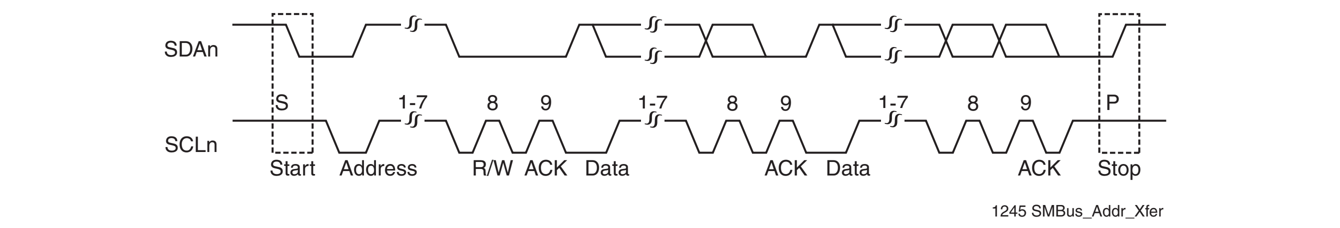


FIGURE 13-4: SMBus Address Transfer

It is possible for multiple masters to generate a START condition and then continue the transfer simultaneously. An arbitration mechanism is provided by the SMBus protocol for this case. Arbitration takes place on the SDAn line to prevent contention on the bus between masters while the SCLn line is at the high level. If a master attempts to send data bit ‘1’ over the bus, and detects SDAn at a low level driven by another master, it will abort the data transfer because the current level on the bus does not match its own. The master that loses the arbitration can generate clock pulses until the end of the last-transmitted data byte. If a master loses the arbitration during the addressing phase, it is possible that the master which has won the arbitration was attempting to address the master which has lost. In this case, the losing master must immediately switch to the slave receiver mode, and acknowledge its own slave address.

See System Management Bus specification, Version 2.0 for more details on SMBus protocol.

13.4 SMBus MMCRs

There are four MMC registers associated with each of the two SMBus controllers. These registers are Multi-master bus control register, Multi-master bus control/status register, Multi-master bus address register, and Multi-master bus transmit/receive data shift register. In addition there are two registers common for both controllers: SMBus switch register and SMBus line status register. All register types are explained in Table 13-1.

TABLE 13-1: SMBus MMCRs

Symbol	Description	Address	Bit Address, Symbol, or Alternative Port Function MSB							LSB	Reset Value
SMCR0	Multi-Master Bus Control	7F31H	ACKEN	CLKSEL	INTEN	INT	TXCLK3	TXCLK2	TXCLK1	TXCLK0	00H
SMSR0	Multi-Master Bus Control / Status	7F32H	SMSR0_ MOD1	SMSR0_ MOD0	SMSR0_ BSY	SMSR0_ SEREN	SMSR0_ ARB	SMSR0_ ADDRS	SMSR0_ ADDR0	SMSR0_ ACK	00H
SAR0	Multi-Master Bus Address	7F33H	SAR0[7:0]								0000000xb
SDSR0	Multi-Master Bus Transmit / Receive Data Shift	7F34H	SDSR0[7:0]								00H
SMCR1	Multi-Master Bus Control	7F67H	ACKEN	CLKSEL	INTEN	INT	TXCLK3	TXCLK2	TXCLK1	TXCLK0	00H
SMSR1	Multi-Master Bus Control / Status Register	7F68H	SMSR1_ MOD1	SMSR1_ MOD0	SMSR1_ BSY	SMSR1_ SEREN	SMSR1_ ARB	SMSR1_ ADDRS	SMSR1_ ADDR0	SMSR1_ ACK	00H
SAR1	Multi-Master Bus Address	7F69H	SAR1[7:0]								0000000xb
SDSR1	Multi-Master Bus Transmit / Receive Data Shift	7F6AH	SDSR1[7:0]								00H
SLSR[1]	SMBus Line Status Register	7F88H	-	-	SCL2	SDA2	SCL1	SDA1	SCL0	SDA0	-
SSCR	SMBus Switch Control Register	7F89H	SM1_EN	SM0_EN	-	-	-	P1_SEL	UART_ SM	CHSEL1	00xxx001b

T13-1.1245

[1] No reset value is specified for SLSR since this is just pass through register.

13.5 SMBus Multi-master Control and Status Registers

The SMBus control and status registers are used to control SMBus operations.

13.5.1 Multi-Master Bus Control Register (SMCR0)

Location		7	6	5	4	3	2	1	0
7F31H	Read	SMCR0_ ACKEN	SMCR0_ CLKSEL	SMCR0_ INTEN	SMCR0_ INT	SMCR0_ TXCLK3	SMCR0_ TXCLK2	SMCR0_ TXCLK1	SMCR0_ TXCLK0
	Write								
	Reset	0	0	0	0	0	0	0	0

13.5.2 Multi-Master Bus Control Register (SMCR1)

Location		7	6	5	4	3	2	1	0
7F67H	Read	SMCR1_ ACKEN	SMCR1_ CLKSEL	SMCR1_I NTEN	SMCR1_I NT	SMCR1_ TXCLK3	SMCR1_ TXCLK2	SMCR1_ TXCLK1	SMCR1_TX CLK0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

SMCRn_ACKEN
Acknowledge enable bit (for SST79LF008 operation as a receiver) (n = 0-1)
1: Enable ACK generation (SDAn is driven low during ACK cycle)
0: Disable ACK generation (SDAn is not driven = SDAn “floats” during ACK cycle)

SMCRn_CLKSEL
SCL clock prescaler control (n = 0-1)
1: $F_{PSCL} = F_{CCLK}/256$
0: $F_{PSCL} = F_{CCLK} /16$
F_{CCLK} = 8051 core clock frequency

SMCRn_INTEN
SMBus controller Interrupt Enable bit (must be always set for normal operation—use INTSRCAMSK register bits to mask SMBus interrupts if necessary) (n = 0-1)
1: Enabled Interrupt
0: Disabled Interrupt

SMCRn_INT
SMBus interrupt pending flag. This bit is set by hardware, and cleared when software is writing ‘0’. Writing ‘1’ to this bit will be ignored. (n = 0-1)
1: Interrupt is pending (SMBus is stalled as SCLn line is held low)
0: No interrupt pending
Events that set SMCR1_INT in hardware include:
When an 1-byte transmission or receiving operation is completed
When a general call or a slave address match occurs
When bus arbitration fails

SMCRn_TXCLK[3:0]
SMBus clock selection control (for SST79LF008 operation as a master) (n = 0-1)
SMBus clock frequency = F_{PSCL} /(SMCR1_TXCLK[3:0] + 1). Must not be set 000x if SMCR1_CLKSEL = 0.

13.5.3 Multi-Master Bus Control / Status Register 0 (SMSR0)

Location		7	6	5	4	3	2	1	0
7F32H	Read	SMSR0 _MOD1	SMSR0 _MOD0	SMSR0 _BSY	SMSR0 _SEREN	SMSR0 _ARB	SMSR0 _ADDRS	SMSR0 _ADDR0	SMSR0 _ACK
	Write					-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

- Not implemented

13.5.4 Multi-Master Bus Control / Status Register 1 (SMSR1)

Location		7	6	5	4	3	2	1	0
7F68H	Read	SMSR1 _MOD1	SMSR1 _MOD0	SMSR1 _BSY	SMSR1 _SEREN	SMSR1 _ARB	SMSR1 _ADDRS	SMSR1 _ADDR0	SMSR1 _ACK
	Write					-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

- Not implemented

SMSRn_MOD[1:0] SMBus controller mode selection bits (n = 0-1)

SMBus controller mode. Selection as a function of SMSRn_MOD bits.

00 Slave receive mode
01 Slave transmit mode
10 Master receive mode
11 Master transmit mode

Normally these bits should be modified by software only when SMBus is idle. If they are modified during data transfer, unexpected results could occur. One exception is when MODE[1:0] is changed in order to generate a REPEATED START condition after the last byte of the current transaction is received or transmitted.

Both SMSRn_MOD0 and SMSRn_MOD1 bits are cleared by hardware automatically when:

1) SMBus STOP condition is detected
2) When SEREN bit is cleared
3) SMBus arbitration failure is detected

SMSRn_MOD0 bit is also set/cleared by hardware equal to the value of R/W# bit of the address phase, when SMBus controller is in slave mode. (n = 0-1)

SMSRn_BSY When read, this bit indicates the bus busy status (SMBus is busy between START and STOP conditions). When written to, this bit controls generation of START and STOP conditions for SMBus controller in master mode.

1: When read: the bus is busy

When written: START condition is generated, regardless of current SMBus busy status

0: When read: the bus is not busy

When written, STOP condition is generated only if SMBus was already busy. To generate START condition when bus is idle (BSY='0'), software should write '1' to SEREN, SMSRn_MOD1, and BSY bits simultaneously. To generate a REPEATED START current when bus is already busy (BSY='1'), software should write '1' to SMSRn_MOD1, BSY, and SEREN bits simultaneously, and then clear INT bit to release the bus. To generate a STOP condition when bus is already busy (BSY='1') software should write '1' to bits SEREN and SMSRn_MOD1 while writing '0' to BSY bit simultaneously, and then clear INT bit to release the bus.
(n = 0-1)

SMSRn_SEREN	SMBus Serial output Enable bit 1: Enable SMBus Receive/Transmit over SDAn and SCLn 0: Disable SMBus Receive/Transmit: MOD[1:0] are cleared, SDAn, SCLn are floated The order of floating SDAn and SCLn is as follows: SCLn floats first, then SDAn floats. Therefore, disabling serial outputs when SST79LF008 is a transmitter may create a STOP condition on the bus if the last transmitted data bit was ‘0’. (n = 0-1)
SMSRn_ARB	Arbitration status flag 1: Bus arbitration failed 0: No bus arbitration failure detected This bit is automatically cleared when START/STOP condition is detected. (n = 0-1)
SMSRn_ADDRS	Slave Address match status 1: Received slave address matches the address value in SAR 0: No slave address match detected Both ARB and ADDRS bits will be set if arbitration failed during address phase, and address sent by the other master matches SAR setting.This bit is automatically cleared when START/STOP condition is detected. (n = 0-1)
SMSRn_ADDR0	Broadcast Address match status flag 1: Received slave address is the broadcast address 00H 0: No broadcast address match is detected Both ARB and ADDRS bits will be set if arbitration failed during address phase, and address sent by the other master is the broadcast address.This bit is automatically cleared when START/STOP condition is detected. (n = 0-1)
SMSRn_ACK	Last-received bit status flag (n = 0-1) 1: Last-received bit is ‘1’ (ACK was not received) 0: Last-received bit is ‘0’ (ACK was received)

13.6 Multi-master Bus Address and Data Shift Registers

The SMBus address register is used to store the slave address of the respective SMBus controller.

13.6.1 Multi-Master Bus Address Register 0 (SAR0)

Location		7	6	5	4	3	2	1	0
7F33H	Read	SAR0_7	SAR0_6	SAR0_5	SAR0_4	SAR0_3	SAR0_2	SAR0_1	-
	Write								
	Reset	0	0	0	0	0	0	0	X

13.6.2 Multi-Master Bus Address Register 1 (SAR1)

Location		7	6	5	4	3	2	1	0
7F69H	Read	SAR1_7	SAR1_6	SAR1_5	SAR1_4	SAR1_3	SAR1_2	SAR1_1	-
	Write								
	Reset	0	0	0	0	0	0	0	X

Symbol	Function
-	Not Implemented
X	Not defined
SARn[7:1]	These bits holds the 7-bit slave address of the respective SMBus controller in the slave mode, they are compared with the data received from the master in the address phase. When address match is detected slave address match ADDRS bit, and SMBus pending interrupt INT bit will be both set. This register can only be written when SEREN = 0. When SAR is read, it always return previously written data. (n = 0-1)

The SMBus transmit/receive data register acts as a serial shift register and read buffer for interfacing with the SMBus. This register performs all read and write operation from/to the SMBus. In the receiver mode, the SMBus data is shifted into the data register until the acknowledge phase. Further reception of data is inhibited (SCL held low) until the interrupt pending bit INT is cleared. In the transmitter mode, if the SEREN bit is set, the data is transmitted to the SMBus as soon as it is written to the data register. Further transmission of data is inhibited (SCL held low) until the interrupt pending bit INT is cleared.

13.6.3 Multi-Master Bus Transmit / Receive Data Shift Register 0 (SDSR0)

Location		7	6	5	4	3	2	1	0
7F34H	Read	SDSR0_7	SDSR0_6	SDSR0_5	SDSR0_4	SDSR0_3	SDSR0_2	SDSR0_1	SDSR0_0
	Write								
	Reset	0	0	0	0	0	0	0	0

13.6.4 Multi-Master Bus Transmit / Receive Data Shift Register 1 (SDSR1)

Location		7	6	5	4	3	2	1	0
7F6AH	Read	SDSR1_7	SDSR1_6	SDSR1_5	SDSR1_4	SDSR1_3	SDSR1_2	SDSR1_1	SDSR1_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
SDSRn[7:0]	Data shift register (n = 0-1) When written to, transmit data to the SMBus provided SEREN = 1 (write is ignored if SEREN = 0). When read, returns the latched value received from SMBus during the last read operation. In address phase of master mode for SMBus controller 0 (controller 1) bit SDSR0 in SDSR0 (SDSR1) register is equal to the inverse of SMSRn_MOD0 bit in the respective SMSR0 (SMSR1) register. In other words, if it is master transmit mode, then SDSR0 equals to '0'; if it is master receive mode, then SDSR0 equals to '1'.

13.7 SMBus Switch Control and Line Status Registers

The SMBus Switch Control register is used to enable/disable both SMBus controllers, as well as to control the SMBus multiplexer, as shown on Figure 13-1. The SMBus Line status register is used to monitor the SMBus line status.

13.7.1 SMBus Switch Control Register (SSCR)

Location		7	6	5	4	3	2	1	0
7F89H	Read	SM1_EN	SM0_EN	-	-	-	P1_SEL	UART_SM	CHSEL1
	Write								
	Reset	0	0	X	X	X	0	0	1

Symbol	Function
-	Not Implemented
X	Not defined
SM1_EN	SMBus controller 1 Enable bit 1: SMBus controller 1 is enabled 0: SMBus controller 1 is disabled and reset to the default state
SM0_EN	SMBus controller 0 Enable bit 1: SMBus controller 0 is enabled 0: SMBus controller 0 is disabled and reset to the default state
P1_SEL	Port1/GPIO86-GPIO93 Selection bit 1: Select 8051 Port1 function 0: Select normal GPIO86-GPIO93 function
UART_SM	UART and SMBus Channel 1 Selection bit 1: UART RXD/TXD function is selected as alternate function for GPIO53/GPIO54 0: SDA1/SCL1 function is selected as alternate function for GPIO53/GPIO54
CHSEL1	Channel Selection for SMBus controller 1 1: Select SMBus channel 1. SCL1 and SDA1 pins are driven by the SMBus controller 1, SCL2 and SDA2 pins are tri-stated 0: Select SMBus channel 2. SCL2 and SDA2 pins are driven by the SMBus controller 1. SCL1 and SDA1 pins are tri-stated.

13.7.2 SMBus Line Status Register (SLSR)

Location		7	6	5	4	3	2	1	0
7F88H	Read	-	-	SCL2	SDA2	SCL1	SDA1	SCL0	SDA0
	Write			-	-	-	-	-	-
	Reset	X	X	SCL2	SDA2	SCL1	SDA1	SCL0	SDA0

Symbol	Function
-	Not implemented
X	Not defined

13.8 SMBus Operations

After the SST79LF008 chip resets, both SMBus controllers are disabled. The 8051 firmware can enable each controller and configure SMBus channels via SSCR, GPIODSEL, and GPIOGSEL registers. The firmware can also specify the SMBus clock frequency and enable SMBus interrupts in SMCR0 or SMCR1 registers, and load slave addresses to the respective SAR0 or SAR1 registers. Then SST79LF008 is ready for SMBus transmit/receive transactions—a typical example is described in Figure 13-5.

13.8.1 Master Transmit Mode

In Master Transmit Mode, the master addresses the slave by sending the slave address, then the master will transmit data to the slave, and terminates the transfer after all data has been transmitted.

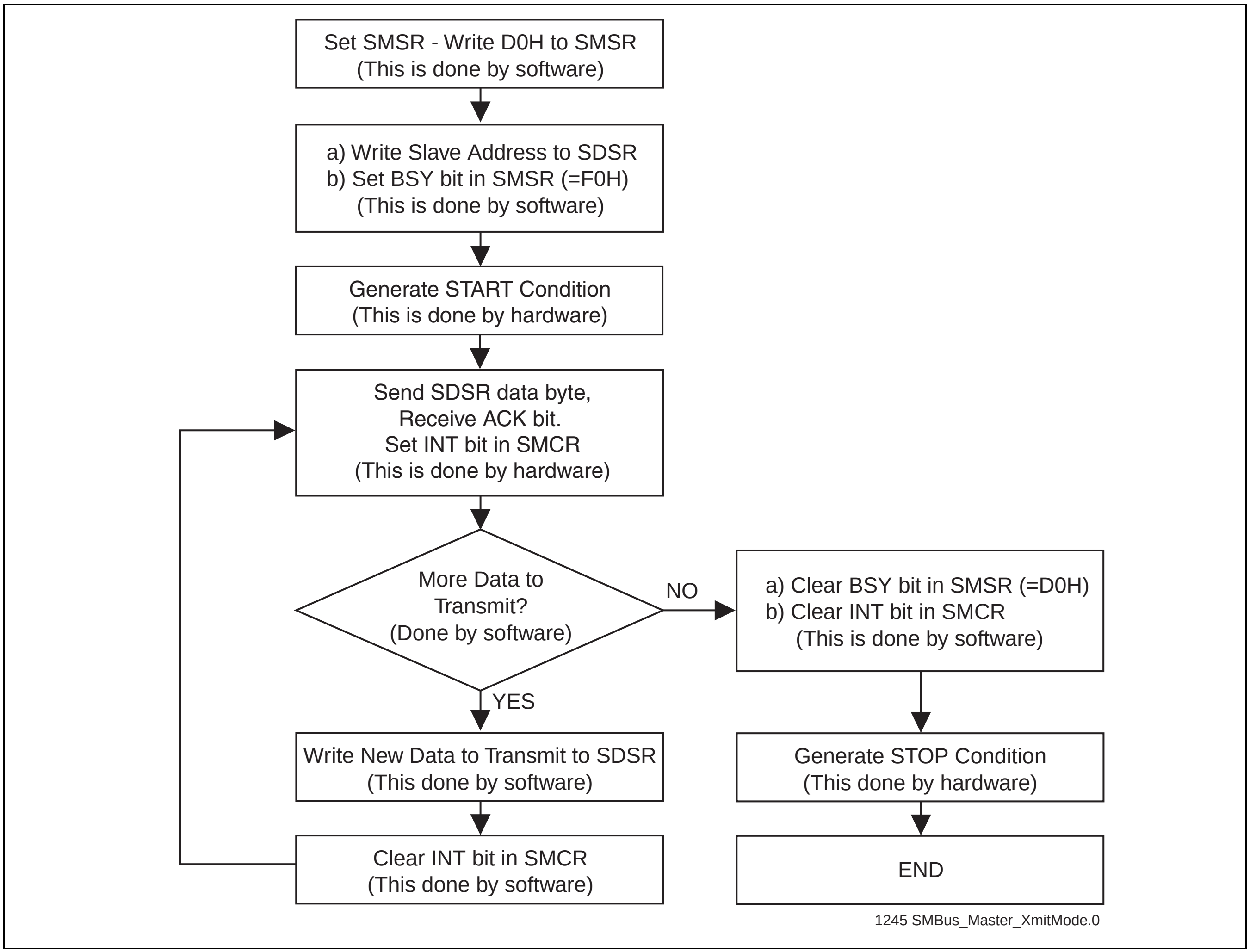

Set SMSR - Write D0H to SMSR
(This is done by software)
a) Write Slave Address to SDSR
b) Set BSY bit in SMSR (=F0H)
(This is done by software)
Generate START Condition
(This is done by hardware)
Send SDSR data byte,
Receive ACK bit.
Set INT bit in SMCR
(This is done by hardware)
More Data to
Transmit?
(Done by software)
NO
a) Clear BSY bit in SMSR (=D0H)
b) Clear INT bit in SMCR
(This is done by software)
YES
Write New Data to Transmit to SDSR
(This done by software)
Generate STOP Condition
(This done by hardware)
Clear INT bit in SMCR
(This done by software)
END
1245 SMBus_Master_XmitMode.0

FIGURE 13-5: SMBus Master Transmit Mode Operation

Figure 13-5 illustrates interaction between 8051 firmware and SMBus controller hardware in Master Transmit mode:

Master Transmit Mode

1. Write D0H to SMSR. This presets the SMBus controller for Master Transmit mode.
2. Write a 7-bit slave address to SDSR[7-1], and ‘0’ to SDSR[0] (R/W# bit). The R/W# bit determines the direction of the transfer—if R/W# = 0 then the master will send data to the slave. Set the BSY bit in SMSR register. SMBus controller will generate a START condition and automatically send SDSR data over the SMBus.
3. The hardware completes transmission of eight data bits, receives the ninth bit during ACK clock period, and sets interrupt pending—keeping SMBus on hold and allowing the software to process the transfer results and check errors.
   Note that in the case of successful transmission, allowing that arbitration has not failed, the slave is expected to send back an ACK to the master—lowering the SDAn line during ACK clock.
4. Check for more data to transfer.
   If the master has more data to transfer, the software will write the next data byte to SDSR and clear the interrupt pending bit in SMCR. This causes the data in SDSR to be sent automatically over the SMBus by controller hardware and returns to Step 3.
   If the master has no data to transfer, the software will clear the BSY bit in SMSR and clear the interrupt pending bit in SMCR. The controller hardware will generate a STOP condition and release the SMBus lines which completes the transaction.

13.8.2 Master Receive Mode

In Master Receive Mode, the master addresses the slave by sending the slave address. After which, the master will receives data from the slave and then terminates the transfer after all data has been received.

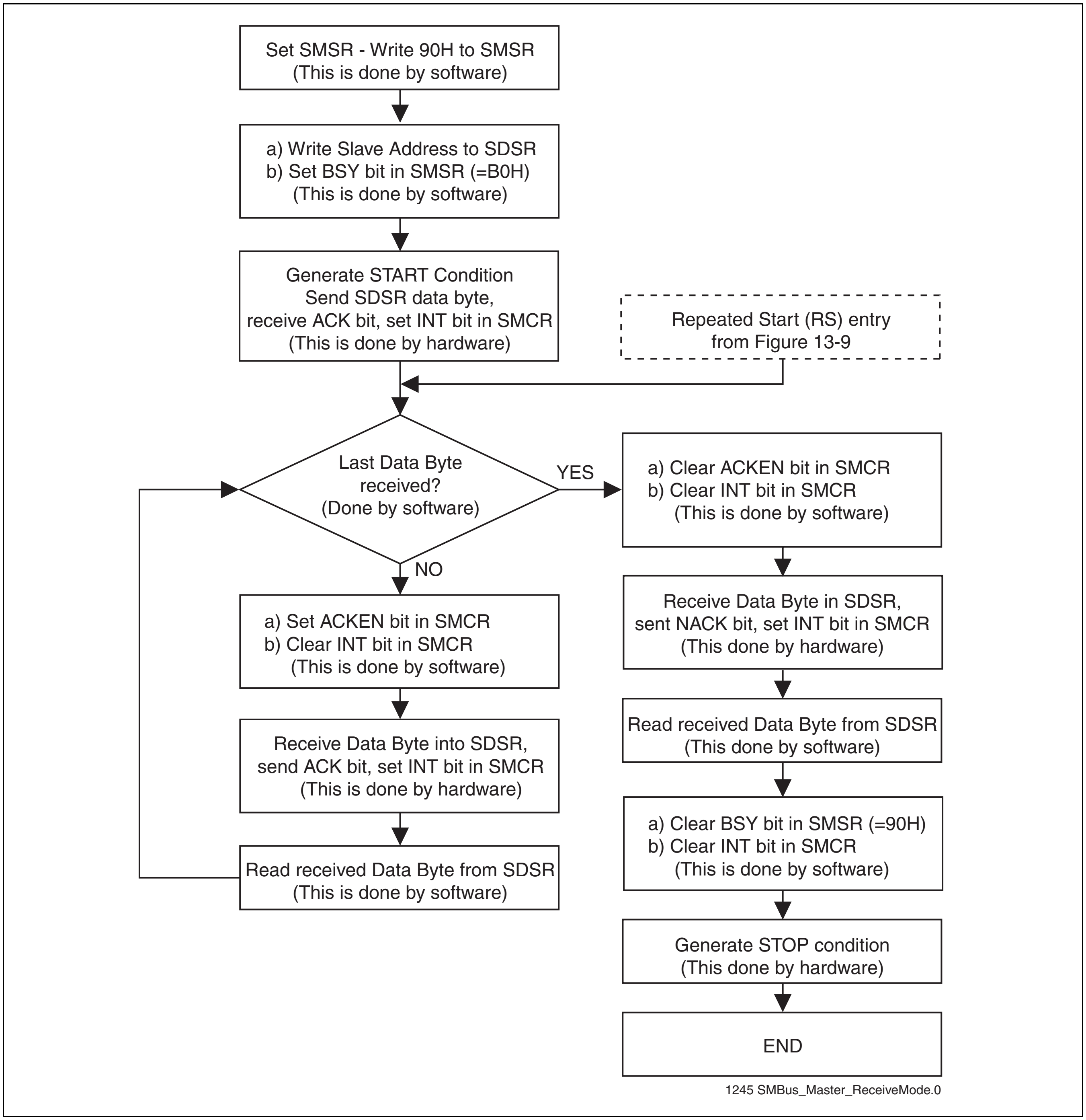

Set SMSR - Write 90H to SMSR
(This is done by software)
a) Write Slave Address to SDSR
b) Set BSY bit in SMSR (=B0H)
(This is done by software)
Generate START Condition
Send SDSR data byte,
receive ACK bit, set INT bit in SMCR
(This is done by hardware)
Repeated Start (RS) entry
from Figure 13-9
Last Data Byte
received?
(Done by software)
YES
NO
a) Clear ACKEN bit in SMCR
b) Clear INT bit in SMCR
(This is done by software)
Receive Data Byte in SDSR,
sent NACK bit, set INT bit in SMCR
(This done by hardware)
Read received Data Byte from SDSR
(This done by software)
a) Clear BSY bit in SMSR (=90H)
b) Clear INT bit in SMCR
(This is done by software)
Generate STOP condition
(This done by hardware)
END
a) Set ACKEN bit in SMCR
b) Clear INT bit in SMCR
(This is done by software)
Receive Data Byte into SDSR,
send ACK bit, set INT bit in SMCR
(This is done by hardware)
Read received Data Byte from SDSR
(This is done by software)
1245 SMBus_Master_ReceiveMode.0

FIGURE 13-6: SMBus Master Receive Mode Operation

Figure 13-6 illustrates interaction between 8051 firmware and SMBus controller hardware in Master Receive mode:

Master Receive mode:

1. Write 90H to SMSR. This will preset SMBus controller for Master Receive mode.
2. Write a 7-bit slave address to SDSR[7-1], and '1' to SDSR[0] (R/W# bit). The R/W# bit determines the direction of the transfer—if R/W# = 1 then the master will receive data from the slave. Set the BSY bit in SMSR register. SMBus controller will generate a START condition and automatically send SDSR data over the SMBus.
3. The hardware completes transmission of eight data bits, receives the ninth bit during ACK clock period, and sets interrupt pending. This keeps SMBus on hold and allows the software to process the transfer results and check errors.
   Note that in the case of successful transmission, allowing that arbitration has not failed, the slave is expected to send back an ACK to the master—lowering the SDAn line during ACK clock.
4. Determine whether the next transfer is the last data byte to transfer from the slave.
   If not, set the ACK enable bit and clear the interrupt pending bit in SMCR to release the SMBus allowing the slave to send the next data byte to the master. Proceed to Step 5 to finish receiving the next data byte.
   If yes, indicated by only one byte remaining, clear the ACK enable bit and clear the interrupt pending bit in SMCR. This releases the SMBus allowing the slave to send the last data byte to the master. Proceed to Step 7 to complete receiving the last data byte.
5. The hardware completes receipt of eight data bits, then sends ninth bit during ACK clock period, and sets interrupt pending. This keeps SMBus on hold and allows the software to read the received byte. The data sent in ACK clock period depends on the value of ACKEN bit in SMCR register. If ACKEN bit is set, then ACK is generated automatically—SDAn is '0' in the ACK clock period.
6. Read the received data byte from SDSR and return to Step 4.
7. The hardware completes receipt of eight data bits, sends ninth bit during ACK clock period, and sets interrupt pending. This keeps SMBus on hold and allows the software to read the received byte. The data sent in ACK clock period depends on the value of ACKEN bit in SMCR register. If ACKEN bit is cleared, then NACK is generated automatically—SDAn is floating = '1' in ACK clock period.
8. Read the last received data byte from SDSR, clear the BSY bit in SMSR, and then clear the interrupt pending bit in SMCR. The controller hardware will generate a STOP condition and release the SMBus channel which completes the transaction.

13.8.3 Slave Transmit Mode

In the Slave Transmit Mode, the slave compares the received address with SAR register contents. If the received address and SAR register matches, the slave will transmit data to the master until the master stops the transaction.

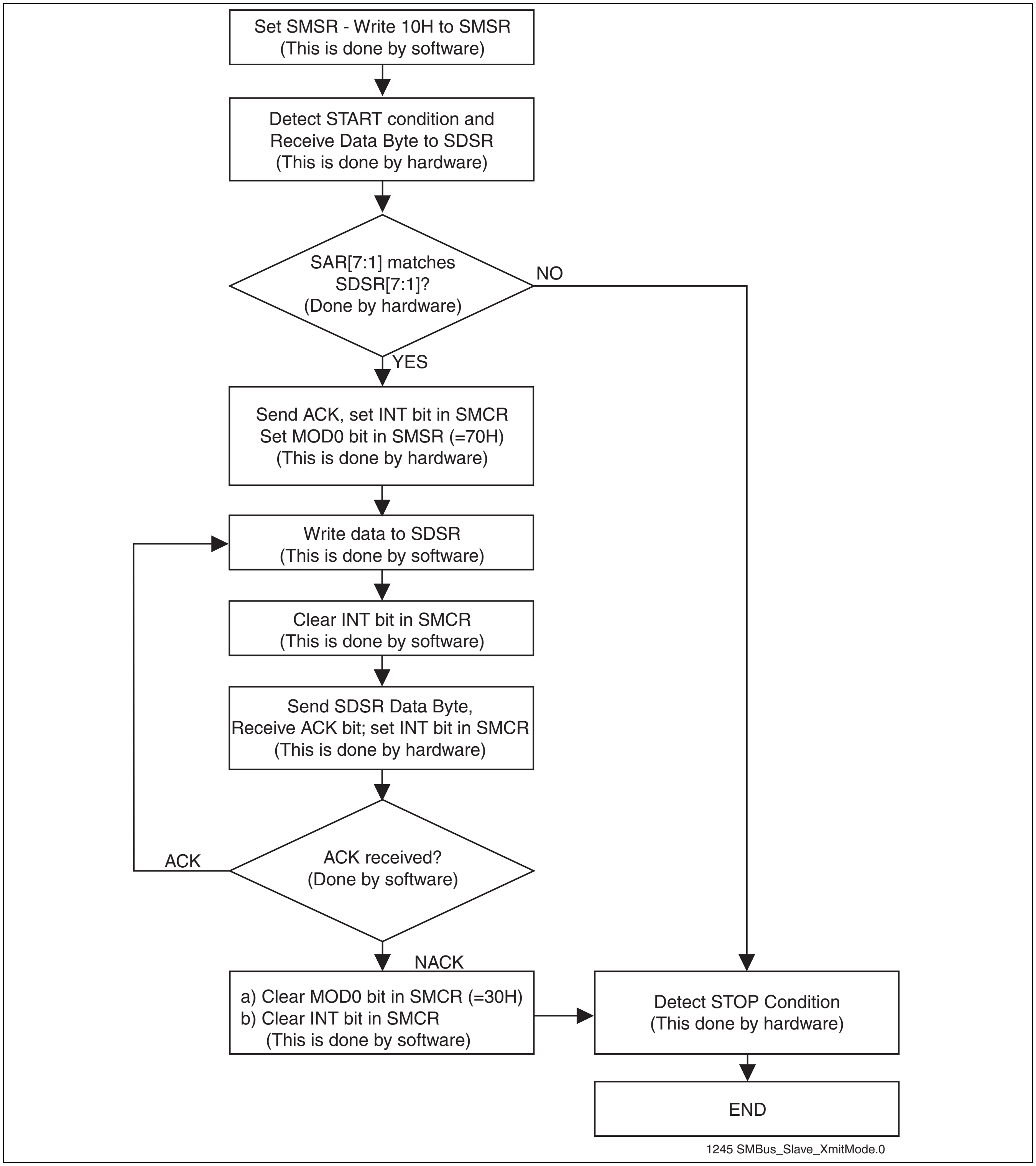

Set SMSR - Write 10H to SMSR
(This is done by software)
Detect START condition and
Receive Data Byte to SDSR
(This is done by hardware)
SAR[7:1] matches
SDSR[7:1]?
(Done by hardware)
NO
YES
Send ACK, set INT bit in SMCR
Set MOD0 bit in SMSR (=70H)
(This is done by hardware)
Write data to SDSR
(This is done by software)
Clear INT bit in SMCR
(This is done by software)
Send SDSR Data Byte,
Receive ACK bit; set INT bit in SMCR
(This is done by hardware)
ACK
ACK received?
(Done by software)
NACK
a) Clear MOD0 bit in SMCR (=30H)
b) Clear INT bit in SMCR
(This is done by software)
Detect STOP Condition
(This done by hardware)
END
1245 SMBus_Slave_XmitMode.0

FIGURE 13-7: SMBus Slave Transmit Mode Operation

Figure 13-7 illustrates interaction between 8051 firmware and SMBus controller hardware in Slave Transmit mode:

Slave Transmit mode:

1. Write 10H to SMSR. This will preset SMBus controller for Slave Receive mode.
2. The hardware detects the START bit, automatically sets the BSY bit, and receives the data byte from the bus.
   Then the hardware compares the value of SAR[7:1] with the data received in the SDSR[7:1]:
   If it matches, proceed to Step 3 to complete the address phase.
   If not, detect STOP condition when generated by the master. The STOP condition clears BSY bit in the hardware and completes the transaction.
3. The hardware sends ACK and sets the interrupt pending bit to keep SMBus on hold. If the value of the R/W# bit from the master (=SDSR[0]) is ‘1’, then SMSRn_MOD0 is set and SMBus controller automatically switches to Slave Transmit mode. For example, when the master requires data from the slave.
4. Write data to be transmitted to the SDSR and clear the interrupt pending bit in SMCR. This causes the data in SDSR to be sent automatically over SMBus by controller hardware.
5. The hardware completes transmission of eight data bits, receives the ninth bit during ACK clock period, and sets interrupt pending. This keeps SMBus on hold and allows the software to process the transfer results and check errors.
   Note that in the case of successful transmission, allowing that arbitration has not failed, the master is expected to send back an ACK to the slave—lowering the SDAn line during ACK clock—except for the last byte of the transaction.
6. Determine whether the last data byte has transmitted.
   If no, ACK was received. Return to Step 4 to start the next data byte transmit cycle.
   If yes, NACK was received. Clear SMSRn_MOD0 bit in SMSR register and clear the interrupt pending bit in SMCR. This will release the bus so that the master will be able to generate a STOP condition which clears BSY bit in hardware and completes the transaction.

13.8.4 Slave Receive Mode

In Slave Receive Mode, the slave compares the received address with the SAR register contents. If they match, the slave will receive data from the master until the master stops the transaction.

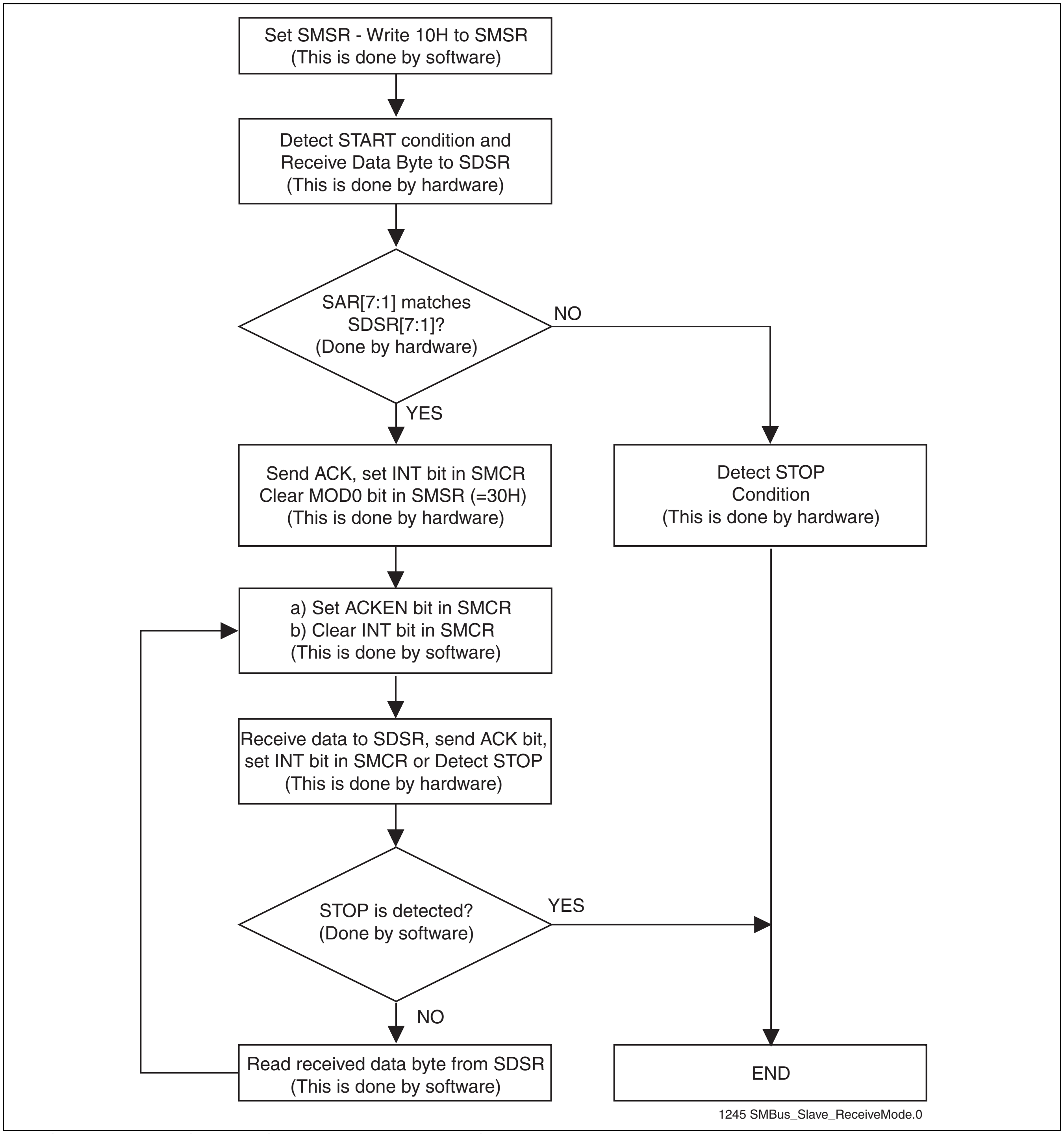

Set SMSR - Write 10H to SMSR
(This is done by software)
Detect START condition and
Receive Data Byte to SDSR
(This is done by hardware)
SAR[7:1] matches
SDSR[7:1]?
(Done by hardware)
NO
YES
Send ACK, set INT bit in SMCR
Clear MOD0 bit in SMSR (=30H)
(This is done by hardware)
Detect STOP
Condition
(This is done by hardware)
a) Set ACKEN bit in SMCR
b) Clear INT bit in SMCR
(This is done by software)
Receive data to SDSR, send ACK bit,
set INT bit in SMCR or Detect STOP
(This is done by hardware)
STOP is detected?
(Done by software)
YES
NO
Read received data byte from SDSR
(This is done by software)
END
1245 SMBus_Slave_ReceiveMode.0

FIGURE 13-8: SMBus Slave Receive Mode Operation

Figure 13-8 illustrates interaction between 8051 firmware and SMBus controller hardware in Slave Receive mode:

Slave Receive mode:

1. Write 0x10 to SMSR. This will preset SMBus controller for Slave Receive mode.
2. The hardware detects the START bit, automatically sets the BSY bit, and receives the data byte from the bus.
   Then the hardware compares the value of SAR[7:1] with the data received in the SDSR[7:1]:
   If it matches, proceed to Step 3 to complete the address phase.
   If not, detect STOP condition when generated by the master. The STOP condition clears BSY bit in the hardware and completes the transaction.
3. The hardware sends ACK and sets the interrupt pending bit to keep SMBus on hold. If the value of the R/W# bit from the master (=SDSR[0]) is ‘0’, then SMSRn_MOD0 bit in SMSR is kept unchanged and SMBus controller is in Slave Receive Mode. This happen, for example, when the master sends data to the slave.
4. Set ACKEN bit and clear the interrupt pending bit to release the SMBus, allowing the master to send data byte to the slave.
5. If the master device stops the transaction, the hardware detects a STOP condition and clears BSY bit.
   OR
   If the master device continues sending data, the hardware completes transmission of eight data bits, receives the ninth bit during ACK clock period, and sets interrupt pending. This keeps SMBus on hold and allows the software to read the received byte. The data sent in ACK clock period depends on the value of ACKEN bit in SMCR register. If ACKEN bit is set, then ACK is generated automatically (SDAn is ‘0’ in ACK clock period.
6. Determine whether the STOP bit is detected (BSY bit is cleared.)
   If no STOP condition is detected, go to Step 7 to read data.
   If master generates a STOP condition, then the transaction is complete.
7. Software reads the received data in SDSR and returns to Step 4 to start the next data byte receiving cycle.

13.8.5 Switching Between Master Transmit and Master Receive Modes

When switching from Master Transmit to Master Receive mode, a REPEATED START condition precedes the switch. This differs from most general mode switches when the switch occurs after a STOP condition and while the SMBus is idle. For example, when in Transmit Mode and after all necessary data is transmitted, the software writes a 7-bit slave address to SDSR[7:1] and '1' to the SDSR[0] (R/W# bit). The software then clears the SMSRn_MOD0 bit in SMSR and clears the interrupt pending bit in SMCR to release the SMBus, which causes the controller hardware to generate a REPEATED START condition and the SMBus operation continues in Master Receive mode, see Figure 13-9. Refer to Section 13.5 for Master Transmit details and Section 13.6 Master Receive details.

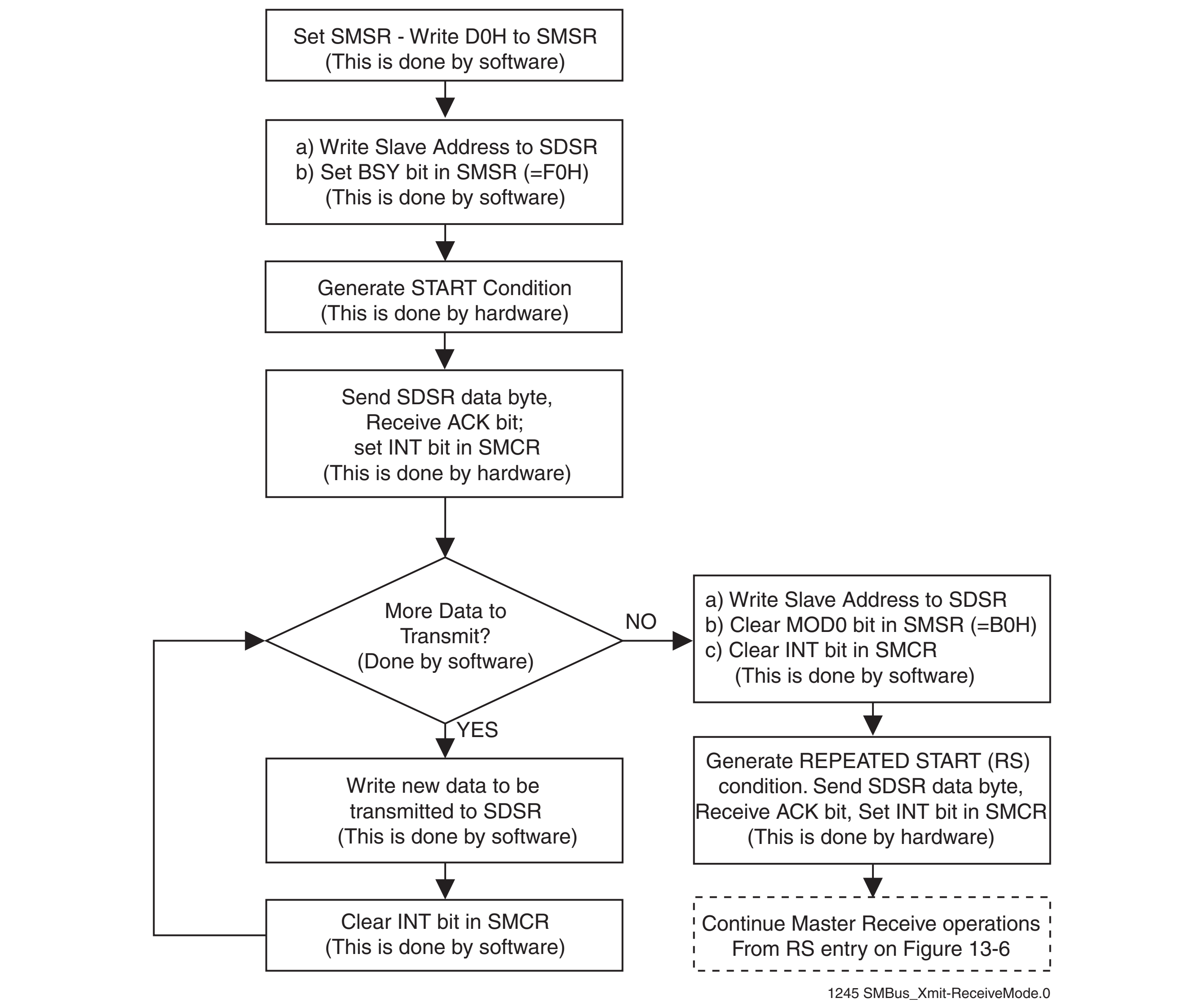

Set SMSR - Write D0H to SMSR
(This is done by software)
a) Write Slave Address to SDSR
b) Set BSY bit in SMSR (=F0H)
(This is done by software)
Generate START Condition
(This is done by hardware)
Send SDSR data byte,
Receive ACK bit;
set INT bit in SMCR
(This is done by hardware)
More Data to Transmit?
(Done by software)
NO
YES
a) Write Slave Address to SDSR
b) Clear MOD0 bit in SMSR (=B0H)
c) Clear INT bit in SMCR
(This is done by software)
Write new data to be transmitted to SDSR
(This is done by software)
Generate REPEATED START (RS) condition. Send SDSR data byte, Receive ACK bit, Set INT bit in SMCR
(This is done by hardware)
Clear INT bit in SMCR
(This is done by software)
Continue Master Receive operations From RS entry on Figure 13-6
1245 SMBus_Xmit-ReceiveMode.0

FIGURE 13-9: SMBus Transmit/Receive Mode Switch

Figure 13-9 illustrates interaction between 8051 firmware and SMBus controller hardware in Master Transmit to Master Receive mode switch:

Master Transmit Mode to Master Receive Mode Switch:

1. Write D0H to SMSR. This will preset SMBus controller for Master Transmit mode.
2. Write a 7-bit slave address to SDSR[7-1] and ‘0’ to SDSR[0] (R/W# bit). The R/W# bit determines the direction of the transfer. For example, if R/W# = ‘0’, then the master sends data to the slave. Set the BSY bit in the SMSR register. SMBus controller generates a START condition and automatically sends SDSR data over the SMBus.
3. The hardware completes the transmission of eight data bits, receives the ninth bit during ACK clock period, and sets the interrupt pending. This keeps SMBus on hold and allows the software to process the transfer results and check errors.

Note: In the case of a successful transmission, allowing that arbitration has not failed, the slave is expected to send back an ACK to the master—lowering the SDAn line during ACK clock.

4. Check for more data transfer.
   If the master has more data to transfer, the software will write the next data byte to SDSR and clear the interrupt pending bit in SMCR. This causes the data in SDSR to be sent automatically over the SMBus by controller hardware and returns to Step 3.
   If the master has no data to transfer, the software will write a 7-bit slave address to SDSR[7-1] and ‘1’ to SDSR[0] (R/W# bit). The R/W# bit determines the direction of the transfer. For example, if R/W# = ‘1’, then the master receives data from the slave. The software clears the SMSRn_MOD0 bit in SMSR and clears the interrupt pending bit in SMCR to release the SMBus. The controller hardware will generate a REPEATED START condition and the SMBus operation continues in Master Receive mode, see Figure 13-9. For Master Transmit details, see Section 13.5 For Master Receive details, see Section 13.6.

14.0 PS/2 INTERFACE

14.1 PS/2 Features

- IBM PS/2 standard compliant
- Three independent PS/2 channels
- PS/2 hardware state machine for each channel
- Embedded transfer time-out detection
- Support both polling and interrupt driven operation
- Optional support for software bit-banging control
- Wake up from Idle and Power Down modes

14.2 PS/2 Channels

The PS/2 interface is an industry standard interface for PC communication to an external keyboard, mouse, internal pointing device, and other PS/2 compatible auxiliary devices. The SST79LF008 provides three identical PS/2 serial channels that are used to interface directly with PS/2 peripherals (see Figure 14-1).

Each PS/2 channel consists of two signal lines: PS/2 clock (PSCLKn, n=0-2) and PS/2 data (PSDATn, n=0-2). The respective I/O buffers must have open drain outputs, so that either the SST79LF008 PS/2 host, or the PS/2 device can control PS/2 signals during bidirectional communications. However, the clock for both transmit and receive protocols is always generated by the PS/2 peripheral device. The PS/2 lines are connected externally to the positive 5V source via pull-up resistors (typically 10 KOhm).

The on-chip PS/2 hardware state machine for each channel supports standard IBM PS/2 compliant receive and transmit protocols. Embedded PS/2 time-out detection frees core timers from PS/2 interface control. For any PS/2 transfer, start-bit interrupt as well as transfer completion interrupt are generated.

The 8051 core controls PS/2 channels via memory mapped configuration registers. The PS/2 hardware state machine may be disabled. In this case, the respective PS/2 channel can operate in software controlled bit-banging mode. This allows communication to peripheral devices, which do not meet the standard PS/2 protocol timing.

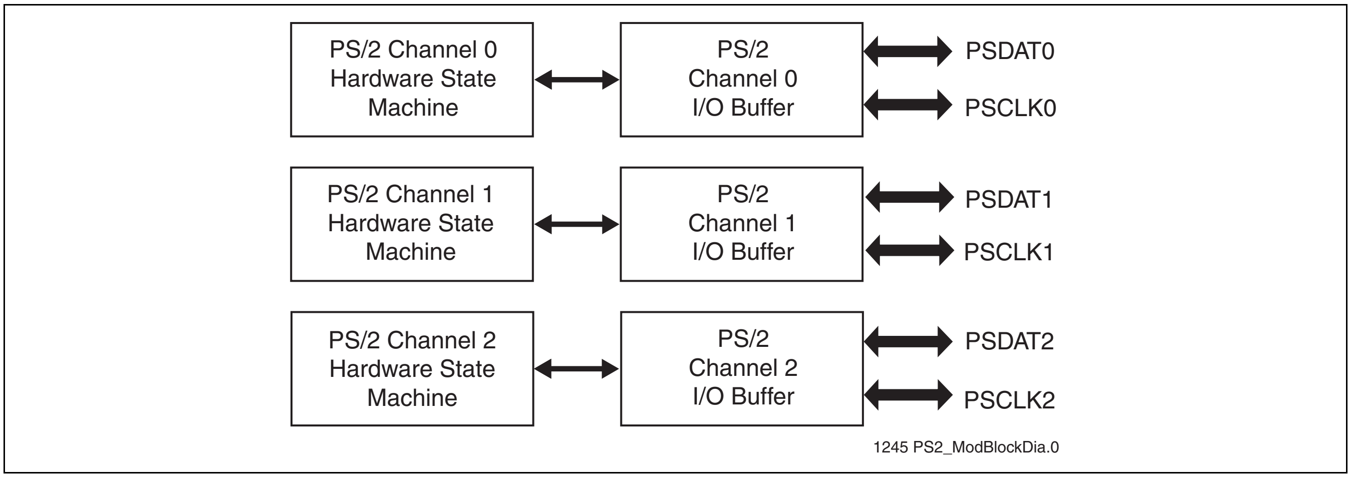

PS/2 Channel 0 Hardware State Machine
PS/2 Channel 0 I/O Buffer
PSDAT0
PSCLK0
PS/2 Channel 1 Hardware State Machine
PS/2 Channel 1 I/O Buffer
PSDAT1
PSCLK1
PS/2 Channel 2 Hardware State Machine
PS/2 Channel 2 I/O Buffer
PSDAT2
PSCLK2
1245 PS2_ModBlockDia.0

FIGURE 14-1: PS/2 Module Block Diagram

Operations of all PS/2 channels are independent except for the following scenario. If PS/2 hardware state machine is enabled—the PS2CRn_PS2_EN (n=0-2) bit is set,see Section 14.4.3—only the channel which detects the start bit first is selected for receiving. The transfer over all other enabled channels is automatically inhibited by resetting the respective control registers to the default values, which results in forcing low state on the PS/2 clock lines. Software can set the PS2CRn_PS2_EN (n = 0-2) bits for the inhibited channels at any time, but hardware will hold clock lines low until the active channel completes receiving data and active PS/2 channel interrupt is cleared. If multiple start bits are detected at the same time from more than one channel, the lower order channel will be selected. If PS/2 hardware state machine is disabled, the PS2CRn_PS2_EN (n = 0-2) bit is cleared. It is the responsibility of the firmware to select the active receiving channel. The selection of transmitting channel is always controlled by the firmware.

14.3 PS/2 Protocol Overview

The PS/2 protocol data stream transmits to/from the PS/2 device via each PS/2 channel consists of 11 bits: start bit (always 0), eight data bits (with least significant bit first), a parity bit (always odd parity), and a stop bit (always 1). The transmit protocol also includes a line control bit that serves as PS/2 device acknowledgement.

Normally the PS/2 interface is in the idle state with both clock and data lines floating (i.e., pulled-up high by external resistors). The receive start bit is created by a high to low transition of the clock signal PSCLKn while the data signal PSDATn is held low as shown in Figure 14-2. After the start bit, the PS/2 interface is in the active receive state. In this state, the PS/2 peripheral device generates clock signal PSCLKn and sends PS/2 data on the data line PSDATn. The SST79LF008 PS/2 host samples each data bit on the falling edge of the clock. The eight data bits are followed by the odd parity bit and a stop bit which completes the transfer.

The PS/2 transmit mode is initiated when the PS/2 host switches the PS/2 channel into the transmit idle, or request-to-send state. In this state the respective clock line PSCLKn is high while the data line PSDATn is forced low by the SST79LF008. In response the PS/2 device generates a falling edge on the clock line while the data signal PSDATn is still low. This indicates the start bit for the transmission as shown in Figure 14-3. After the start bit, the PS/2 interface is in the active transmit state. In this state, the PS/2 peripheral device generates clock signal PSCLKn, but the PS/2 data on the data line PSDATn is sent by the PS/2 host. Each data bit is shifted out of the SST79LF008 host on the falling edge of the clock. The eight data bits are followed by an odd parity bit and a stop bit. Then, the PS/2 device forces data line low and generates one more clock, called line-control bit, to complete the transfer.

When the PS/2 host is receiving data from the PS/2 peripheral device the data stream can be aborted by forcing the clock line of the respective channel low. If the PS/2 abort occurs prior to the falling edge of the 10th clock, then the received data is discarded and the peripheral device will re-transmit it later. If the PS/2 abort occurs following the falling edge of the 10th clock, then the received data must be accepted by the host, as the peripheral device will not re-transmit it.

See *IBM Personal System/2 Hardware Interface Technical Reference* for more details on PS/2 protocol.

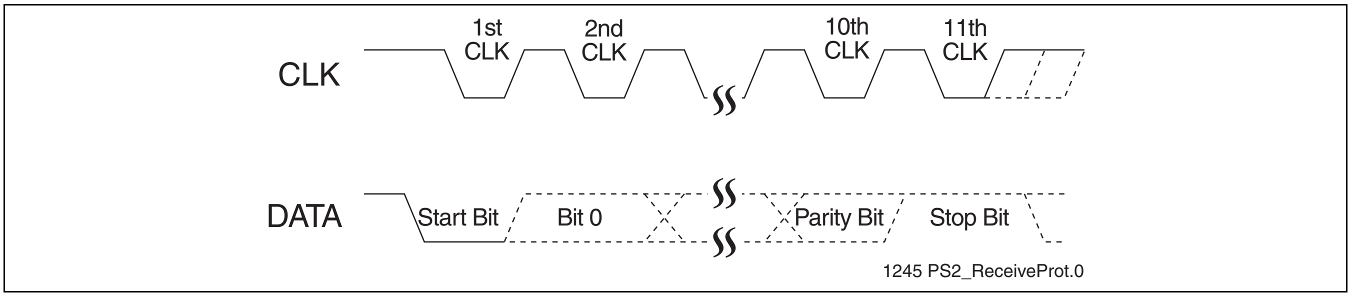


FIGURE 14-2: PS/2 Receive Protocol

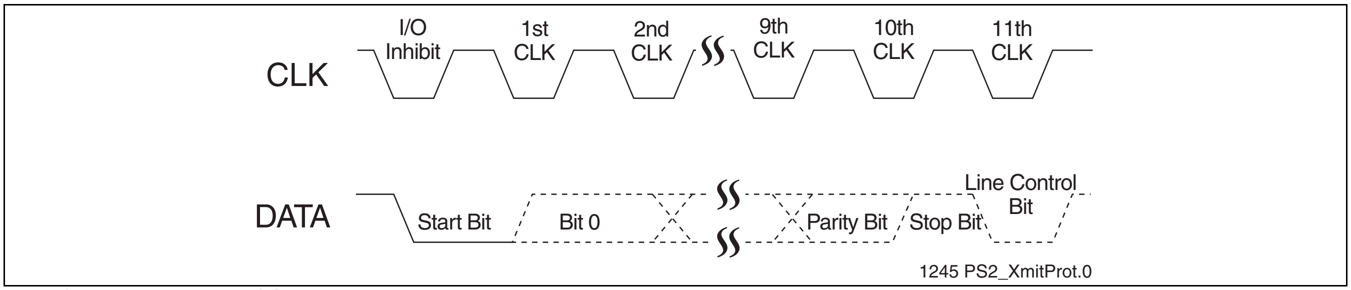


FIGURE 14-3: PS/2 Transmit Protocol

14.4 PS/2 MMCRs

There are five MMC registers associated with each PS/2 channel: PS/2 transmit register, PS/2 receive register, PS/2 control register, PS/2 status, and PS/2 alternate status registers. The transmit and receive registers are located at the same address. There are also two registers common for all channels: PS/2 time-out control register and PS/2 status 2 register.

14.4.1 PS/2 Transmit Registers

PS/2 transmit registers are write-only registers. To transmit a data byte over the PS/2 interface by the PS/2 hardware, it must be written while the PS2CRn_PS2_EN, n=0-2,and PS2CRn_PS2_T/R, n=0-2 bits in the PS/2 control register and the PS2STSn_XMIT_IDLE bit in the PS2 status register, or Alternate PS2 status register are set. If any one of the three bits, are cleared, PS2CRn_PS2_EN, PS2_T/R, or PS2STSn_XMIT_IDLE, then the data written to the transmit register is ignored.

When PS/2 transmission is initiated, the PS2STSn_XMIT_IDLE, n=0-2, and APS2STSn_XMIT_IDLE, n=0-2, bits are automatically cleared. After successful completion of the transmission, the PS2STSn_XMIT_IDLE and APS2STSn_XMIT_IDLE bits are set, and the PS2CRn_PS2_T/R bit is cleared in hardware. This automatically switches the respective PS/2 channel into the receive mode.

14.4.1.1 PS/2 Transmit Register 0 (PS2TX0)

Location		7	6	5	4	3	2	1	0
7F41H	Write	PS2TX0 _7	PS2TX0 _6	PS2TX0 _5	PS2TX0 _4	PS2TX0 _3	PS2TX0 _2	PS2TX0 _1	PS2TX0 _0
	Reset	0	0	0	0	0	0	0	0

14.4.1.2 PS/2 Transmit Register 1 (PS2TX1)

Location		7	6	5	4	3	2	1	0
7F45H	Write	PS2TX1 _7	PS2TX1 _6	PS2TX1 _5	PS2TX1 _4	PS2TX1 _3	PS2TX1 _2	PS2TX1 _1	PS2TX1 _0
	Reset	0	0	0	0	0	0	0	0

14.4.1.3 PS/2 Transmit Register 2 (PS2TX2)

Location		7	6	5	4	3	2	1	0
7F49H	Write	PS2TX2 _7	PS2TX2 _6	PS2TX2 _5	PS2TX2 _4	PS2TX2 _3	PS2TX2 _2	PS2TX2 _1	PS2TX2 _0
	Reset	0	0	0	0	0	0	0	0

Symbol	**Function**
PS2TXn[7:0]	PS/2 transmit, write only, register bits (n = 0-2)

14.4.2 PS/2 Receive Registers

PS/2 receive registers are read-only registers. When the PS2CRn_PS/2_EN (n = 0-2) bit is set and the PS2CRn_PS/2_T/R (n = 0-2) is cleared, the PS/2 hardware state machine places data, received from the peripheral device, into the receive register at the end of a successful receipt of data. At the same time, the respective PS/2 clock line is forced low by the PS/2 hardware to inhibit any further PS/2 transmission, and the PS2STSn_RDATA_RDY or APS2STSn_RDATA_RDY (n = 0-2) bits inthe status register or alternate status register are set indicating that data is ready to be read by the software.

Thus, data received over PS/2 interface can be read from the PS/2 receive register only when the PS2STSn_RDATA_RDY or APS2STSn_RDATA_RDY (n=0-2) bits are set. Reading this register while the PS2STSn_RDATA_RDY or APS2STSn_RDATA_RDY (n=0-2) bits are cleared always returns 0FFH.

The PS2STSn_RDATA_RDY or APS2STSn_RDATA_RDY bits must be cleared by reading the status register in order to allow the next PS/2 data reception or transmission.

14.4.2.1 PS/2 Receive Register 0 (PS2RCV0)

Location		7	6	5	4	3	2	1	0
7F41H	Read	PS2RCV0_7	PS2RCV0_6	PS2RCV0_5	PS2RCV0_4	PS2RCV0_3	PS2RCV0_2	PS2RCV0_1	PS2RCV0_0
	Reset	1	1	1	1	1	1	1	1

14.4.2.2 PS/2 Receive Register 1 (PS2RCV1)

Location		7	6	5	4	3	2	1	0
7F45H	Read	PS2RCV1_7	PS2RCV1_6	PS2RCV1_5	PS2RCV1_4	PS2RCV1_3	PS2RCV1_2	PS2RCV1_1	PS2RCV1_0
	Reset	1	1	1	1	1	1	1	1

14.4.2.3 PS/2 Receive Register 2 (PS2RCV2)

Location		7	6	5	4	3	2	1	0
7F49H	Read	PS2RCV2_7	PS2RCV2_6	PS2RCV2_5	PS2RCV2_4	PS2RCV2_3	PS2RCV2_2	PS2RCV2_1	PS2RCV2_0
	Reset	1	1	1	1	1	1	1	1

Symbol	**Function**
PS2RCVn[7:0]	PS/2 receive, read only, register bits (n = 0-2)

14.4.3 PS/2 Control Registers

14.4.3.1 PS/2 Control Register 0 (PS2CR0)

Location		7	6	5	4	3	2	1	0
7F42H	Read	PS2CR0_WR_CLK	PS2CR0_WR_DATA	PS2CR0_STOP1	PS2CR0_STOP0	PS2CR0_PARITY1	PS2CR0_PARITY0	PS2CR0_PS2_EN	PS2CR0_PS2_T/R
	Write								
	Reset	0	1	0	0	0	0	0	0

14.4.3.2 PS/2 Control Register 1 (PS2CR1)

Location		7	6	5	4	3	2	1	0
7F46H	Read	PS2CR1_WR_CLK	PS2CR1_WR_DATA	PS2CR1_STOP1	PS2CR1_STOP0	PS2CR1_PARITY1	PS2CR1_PARITY0	PS2CR1_PS2_EN	PS2CR1_PS2_T/R
	Write								
	Reset	0	1	0	0	0	0	0	0

14.4.3.3 PS/2 Control Register 2 (PS2CR2)

Location		7	6	5	4	3	2	1	0
7F4AH	Read	PS2CR2_ WR_CLK	PS2CR2_ WR_DATA	PS2CR2_ STOP1	PS2CR2_ STOP0	PS2CR2_ PARITY1	PS2CR2_ PARITY0	PS2CR2_ PS2_EN	PS2CR2_ PS2_T/R
	Write								
	Reset	0	1	0	0	0	0	0	0

Symbol **Function**

PS2CRn_WR_CLK
PS/2 clock line control bit (n = 0-2)
When PS2CRn_PS2_EN = 0 (bit-banging enabled)
1: Float the respective PS/2 channel's PSCLKn pin
0: Drive “low” the respective PS/2 channel's PSCLKn pin
When PS2CRn_PS2_EN = 1, this bit can be updated but has no effect on PSCLKn pin. Reading this bit always returns the written value.

PS2CRn_WR_DATA
PS/2 data line control bit (n = 0-2)
When PS2CRn_PS2_EN = 0 (bit-banging enabled)
1: = Float the respective PS/2 channel's PSDATn pin
0: = Drive “low” the respective PS/2 channel's PSDATn pin
When PS2CRn_PS2_EN = 1, this bit can be updated, but has no effect on PSDATn pin. Reading this bit always returns the written value.

PS2CRn_STOP[1:0]
PS/2 hardware state machine stop frame control bits. Valid only when PS2CRn_PS2_EN = 1. PS/2 transmit protocol with high level stop bit (n = 0-2)
00: PS/2 receive protocol with high level stop bit (PS/2 standard)
01: PS/2 receive protocol with low level stop bit
10: PS/2 receive protocol stop bit level is ignored (data for stop bit clock is not checked, however, it is still counted as the 11th clock of the receiving data stream).
11: Reserved

PS2CRn_PARITY[1:0]
PS/2 hardware state machine parity frame control bits. Valid only when PS2CRn_PS2_EN = 1. (n = 0-2)
00: PS/2 receive and transmit protocols with odd parity bit (PS/2 standard)
01: PS/2 receive and transmit protocols with even parity bit
10: PS/2 receive protocol parity bit level is ignored (data for parity bit clock is not checked, however, it is still counted as the 10th clock of the receiving data stream). PS/2 transmit protocol with odd parity bit
11: Reserved

PS2CRn_PS2_EN
PS2 Channel hardware state machine Enable bit (n = 0-2)
1: Enable PS/2 hardware state machine. The PS/2 hardware automatically receives or transmits data over the respective PS/2 channel depending on the PS2CRn_PS2_T/R bit.
0: Disable PS/2 hardware state machine. Enable bit-banging of PSCLKn and PSDATn lines under software control using PS2CRn_WR_CLK and PS2CRn_WR_DATA bits in this register as well as (A)PS2STSn_RD_DATA and (A)PS2STSn_RD_CLK flags in the status register.
When PS2CRn_PS2_EN bit switches from ‘0’ to ‘1’, the PS/2 hardware is enabled in receive or transmit mode depending on PS2CRn_PS2_T/R bit (default: receive mode with PS2CRn_PS2_T/R = 0).
When PS2CRn_PS2_EN bit switches from ‘1’ to ‘0’, the PS/2 lines are set according to PS2CRn_WR_CLK and PS2CRn_WR_DATA values, and PS/2 hardware is disabled.

Note: To abort a transfer from the PS/2 peripheral, the PS2CRn_WR_CLK and PS2CRn_PS2_EN bits should be cleared simultaneously prior to the falling edge of the 10th clock in the receiving data stream (parity bit), and then held for at least 100S. If PS2CRn_PS2_EN bit is cleared after the falling edge of the 10th clock, then the received data is saved in the receive register (provided there is no parity error), (A)PS2STSn_RDATA_RDY bit is set, and PSCLKn line is forced low.

PS2CRn_PS2_T/R

PS/2 Channel Transmit/Receive control bit. Valid only when PS2CRn_PS2_EN = 1. (n = 0-2)
1: Enable PS/2 channel to transmit data
0: Enable PS/2 channel to receive data

Transmit: Setting the PS/2_T/R bit in software causes the PS/2 hardware to inhibit PS/2 interface (PSCLKn line is driven low and PSDATn line is floated). The channel is inhibited until the next write to the transmit register, which switches the PS/2 channel into request-to-send state (PSDATn line is driven low and after that the PSCLKn line is floated). In response, the PS/2 device starts transmission by generating PS/2 clock pulses. During the transmission, the peripheral device drives PSCLKn line and the PS/2 hardware drives PSDATn line until the stop bit is sent. Then the PS/2 device generates line control bit clock and data. The PS/2_T/R bit is automatically cleared if transmit time-out is detected, or by the 11th clock rising edge when the line control bit successfully completes the transmission. In the latter case the PS/2 channel hardware automatically switches into receive mode.

Receive: When PS2CRn_PS2_T/R bit is cleared in software or after successful transmission the PS/2 channel is in receive mode with PSCLKn and PSDATn lines floating and ready to automatically shift data from the peripheral PS/2 device.

Note: The PS2CRn_PS2_T/R bit must not be cleared by software in the middle of the transmission. The PS2CRn_PS2_T/R bit can be set by software in the middle of receiving data, prior to the falling edge of the 10th clock (parity bit). In this case the received data is discarded. If this bit is set after the 10th clock, the received data is saved in the receive register (provided no parity error), the (A)PS2STSn_RDATA_RDY bit is set, and PSCLKn line is forced low. Neither transmit nor receive operation is possible if any of the bits (A)PS2STSn_RDATA_RDY, (A)PS2STSn_T_TIMEOUT, or (A)PS2STn_R_TIMEOUT in the status register are set, because in these cases the channel's PSCLKn line will be held low until the status register is read by software.

14.4.4 PS/2 Status Registers

There are six PS/2 status registers: one PS/2 Status Register and one Alternate PS/2 Status Register for each PS/2 channel. Reading the alternate status register, APS2STSn, does NOT clear the respective channel's interrupt sources and/or status bits. However, reading the status register, PS2STSn, clears PS/2 channel interrupt sources and the status bits PS2STSn_FE/APS2STSn_FE, n=0-2, PS2STSn_PE/APS2STSn_PE, n=0-2, PS2STSn_RDATA_RDY/APS2STSn_RDATA_RDY, n=0-2, PS2STSn_T_TIMEOUT/APS2STSn_T_TIMEOUT, n=0-2, PS2STSn_R_TIMEOUT/APS2STSn_R_TIMEOUT, n=0-2, PS2STSn_TXSB/APS2STSn_TXSB, n=0-2, and PS2STSn_RX_BUSY/APS2STSn_RX_BUSY, n=0-2, of the respective channel.

14.4.4.1 PS/2 Status Register 0 (PS2STS0)

Location		7	6	5	4	3	2	1	0
7F43H	Read	PS2STS0_ RD_CLK	PS2STS0_ RD_DATA	PS2STS0_T _TIMEOUT	PS2STS0_ XMIT_IDLE	PS2STS0_ FE	PS2STS0_ PE	PS2STS0_R _TIMEOUT	PS2STS0_R DATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

14.4.4.2 PS/2 Status Register 1 (PS2STS1)

Location		7	6	5	4	3	2	1	0
7F47H	Read	PS2STS1_ RD_CLK	PS2STS1_ RD_DATA	PS2STS1_T _TIMEOUT	PS2STS1_X MIT_IDLE	PS2STS1_ FE	PS2STS1_ PE	PS2STS1_R _TIMEOUT	PS2STS1_R DATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

14.4.4.3 PS/2 Status Register 2 (PS2STS2)

Location		7	6	5	4	3	2	1	0
7F4BH	Read	PS2STS2_ RD_CLK	PS2STS2_ RD_DATA	PS2STS2_T _TIMEOUT	PS2STS2_X MIT_IDLE	PS2STS2_ FE	PS2STS2_ PE	PS2STS2_R _TIMEOUT	PS2STS2_R DATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

14.4.4.4 Alternate PS/2 Status Register 0 (APS2STS0)

Location		7	6	5	4	3	2	1	0
7FF7H	Read	APS2STS0 _RD_CLK	APS2STS0 _RD_DATA	APS2STS0_ T_TIMEOUT	APS2STS0_ XMIT_IDLE	APS2STS0 _FE	APS2STS0 _PE	APS2STS0_ R_TIMEOUT	APS2STS0_ RDATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

14.4.4.5 Alternate PS/2 Status Register 1 (APS2STS1)

Location		7	6	5	4	3	2	1	0
7FF8H	Read	APS2STS1 _RD_CLK	APS2STS1 _RD_DATA	APS2STS1_ T_TIMEOUT	APS2STS1_ XMIT_IDLE	APS2STS1 _FE	APS2STS1 _PE	APS2STS1_ R_TIMEOUT	APS2STS1_ RDATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

14.4.4.6 Alternate PS/2 Status Register 2 (APS2STS2)

Location		7	6	5	4	3	2	1	0
7FF9H	Read	APS2STS2 _RD_CLK	APS2STS2 _RD_DATA	APS2STS2_ T_TIMEOUT	APS2STS2_ XMIT_IDLE	APS2STS2 _FE	APS2STS2 _PE	APS2STS2_ R_TIMEOUT	APS2STS2_ RDATA_RDY
	Write	-	-	-	-	-	-	-	-
	Reset	0	1	0	1	0	0	0	0

Symbol **Function**

- Not implemented

(A)PS2STSn_RD_CLK PS/2 clock line status flag (n = 0-2)
This bit reflects the current state of the PSCLKn line. The RD_CLK flag can be used in conjunction with PS2CRn_WR_CLK control bit for software bit-banging when PS2CRn_PS2_EN = 0. A high to low transition on PSCLKn pin caused by the peripheral device will generate the following:

- PS2 channel interrupt request in the INTSRCB register when PS2CRn_PS2_EN = 0
- PS2 start bit interrupt request in the WSRCA register, regardless of the PS2CRn_PS2_EN value.

Note: For proper bit-banging operations the (A)PS2STSn_RDATA_RDY, (A)PS2STn_R_TIMEOUT, and (A)PS2STSn_T_TIMEOUT flags in this registers must be cleared.

(A)PS2STSn_RD_DATA PS/2 data line status flag (n = 0-2)
This bit reflects the current state of the PSDATn line. The (A)PS2STSn_RD_DATA flag can be used in conjunction with PS2CRn_WR_DATA control bit for software bit-banging when PS2CRn_PS2_EN= 0

(A)PS2STSn_T_TIMEOUT Transmission Time-out flag (n = 0-2)
This bit is set when PS2CRn_PS2_EN = 1, and one of the following conditions is detected:

- The transmission bit time (time between clock falling edges) exceeds 300µs, if this time-out detection is enabled.
- The transmission start clock is not received within 25 ms of signaling a transmission request event (request-to-send state), if this time-out detection is enabled.
- The time from the 1st (start) clock falling edge to the rising edge of the 11th clock (line control bit) exceeds 2 ms, if this time-out detection is enabled.
- The response start bit is not received within 25ms after successful completion of the transmission, if this time-out detection is enabled.

The channel's PSCLKn pin is pulled down in hardware after the (A)PS2STSn_T_TIMEOUT bit is set and will be held low until (A)PS2STSn_T_TIMEOUT is cleared by reading the status register in software.
The PS/2 channel interrupt request is generated on the low to high transition of (A)PS2STSn_T_TIMEOUT.

(A)PS2STSn_XMIT_IDLE Transmitter idle status flag (n = 0-2)
This bit is cleared by writing to the transmit register when the channel’s hardware is in transmit mode (PS2CRn_PS2_EN = PS2CRn_PS2_T/R = 1). While (A)PS2STSn_XMIT_IDLE = 0, the PS/2 hardware state machine is transmitting data to the PS2 peripheral device.
This bit is set when one of the following events occurs:

- The rising edge of the 11th clock at the end of the successful transmission
- Transmission time-out is detected ((A)PS2STSn_T_TIMEOUT is set)
- When the PS2CRn_PS2_T/R bit is written ‘0’
- When the PS2CRn_PS2_EN bit is written ‘0’

If a transmission is completed successfully, the PS/2 channel will automatically switch into receiving mode.
The PS/2 channel interrupt request is generated on the low to high transition of (A)PS2STSn_XMIT_IDLE.

(A)PS2STSn_FE Framing Error flag (n = 0-2)
This flag (along with (A)PS2STn_R_TIMEOUT) is set following the falling edge of the 11th clock when PS/2 channel is receiving data, and the stop bit level sampled on the falling edge of the 11th clock does not match the stop bit polarity specified in the control register.
This flag is cleared by reading the status register.

(A)PS2STSn_PE Parity Error flag (n = 0-2)
This flag (along with (A)PS2STn_R_TIMEOUT) is set following the falling edge of the 10th clock when PS/2 channel is receiving data, and the parity bit level sampled on the falling edge of the 10th clock does not match either even or odd parity specified in the control register.
This flag is cleared by reading the status register.

(A)PS2STSn_R_TIMEOUT Receiving Time-out flag (n = 0-2)
This bit is set when PS2CRn_PS2_EN = 1, and one of following conditions is detected:

- The receiving bit time (time between clock falling edges) exceeds 300µs, if this time-out detection is enabled
- The time from the 1st (start) clock falling edge to the 11th (stop bit) clock falling edge exceeds 2 ms, if this time-out detection is enabled
- Parity error (A)PS2STn_PE is detected
- Framing error (A)PS2STSn_FE is detected

The channel's PSCLKn pin is pulled down in hardware after the (A)PS2STn_R_TIMEOUT bit is set and will be held low until the (A)PS2STn_R_TIMEOUT bit is cleared by reading the status register in software.
The PS/2 channel interrupt request is generated on the low to high transition of (A)PS2STn_R_TIMEOUT.

(A)PS2STSn_RDATA_RDY Received Data Ready flag (n = 0-2)
If PS2CRn_PS2_EN = 1 and receive protocol is completed with no time-out, parity, or framing errors this bit is set following the falling edge of the 11th clock. It is also set when software attempts to abort current reception. This is done by clearing the PS2CRn_PS2_EN bit or setting the PS2CRn_PS2_T/R bit after the falling edge of the 10th clock. In any case, this bit indicates that the PS/2 receive register contains the data received from the PS/2 device.
The channel's PSCLKn pin is pulled down in hardware after the (A)PS2STSn_RDATA_RDY bit is set and will be held low until the (A)PS2STSn_RDATA_RDY bit is cleared by reading the status register in software.
The PS/2 channel interrupt request is generated on the low to high transition of (A)PS2STSn_RDATA_RDY.

14.4.5 PS/2 Time-out and Status 2 Registers

The PS/2 Time-out control register is used to enable or disable PS/2 protocol time-out detection.

14.4.5.1 PS/2 Time-out Control Register (PS2TMOUT)

Location		7	6	5	4	3	2	1	0
7F44H	Read	-	-	-	-	TXTMSEL	TMRSPEN	TMOUTEN1	TMOUTEN0
	Write								
	Reset	X	X	X	X	0	0	0	0

Symbol **Function**

- Not implemented

X Not defined

TXTMSEL Transmit 2ms Time-out selection bit
1: Enable in transmit mode, 2ms time-out is from the start bit falling edge to the line control bit clock rising edge
0: Disable in transmit mode, 2ms time-out is from the start bit falling edge to the line control bit data falling edge.

TMRSPEN Enable 25ms Response Time-out detection bit
1: Enable response time-out detection after successful transmit completion
0: Disable response time-out detection after successful transmit completion

TMOUTEN1 Enable 2ms and 25ms Time-out detection bit
1: Enable 2ms time-out detection for transmit/receive and 25ms time-out for transmit only
0: Disable 2ms time-out detection for transmit/receive and 25ms time-out for transmit only
In transmit mode, a 2ms time-out is applied to the time interval from the start bit falling edge to the line control bit data, or clock edges, depending on the value of TXTMSEL. A 25ms time-out is applied to the time interval from the request-to-send state to the transmission start bit falling edge.
In receive mode, 2ms time-out is applied to the time interval from the start bit falling edge to the stop bit falling edge.

TMOUTEN0 Enable 300us bit transfer Time-out detection bit
1: Enable bit transfer 300µss time-out detection for transmit/receive
0: Disable bit transfer 300µs time-out detection for transmit/receive.

14.4.5.2 PS/2 Status 2 Register (PS2STATUS2)

The PS/2 Status 2 register is used to provide auxiliary status flags for all three PS/2 channels.

Location		7	6	5	4	3	2	1	0
7F48H	Read	TxSB2	TxSB1	TxSB0	RX_BUSY2	RX_BUSY1	RX_BUSY0	TxRsp	-
	Write	-	-	-	-	-	-	-	
	Reset	0	0	0	0	0	0	0	X

Symbol **Function**

- Not implemented

X Not defined

TxSB[2:0] PS/2 channels 2-0 Transmit Start Bit Time-out flag
This bit is set when the transmission start bit is not detected within 25 ms of the signal of a request-to-send state.
This bit is cleared by reading the respective status register.

RX_BUSY[2:0] PS/2 channel 2-0 Receiver Busy flag
This bit is set when the start bit is received from the peripheral device.
This bit is cleared by reading the respective status register.
To avoid line contention, the software should not start transmission while the respective channel receives data from the PS/2 device and the RX_BUSY bit is ‘1’. See also the PS2CRn_PS2_T/R bit description on page 183.

TxRsp PS/2 Response Time-out flag
This bit is set when no response start bit is detected within 25ms after completion of a successful transmission.
This bit is cleared by reading any channel’s status register.

Note: All PS/2 channel interrupt sources and the related status bits only can be cleared by reading the status register (PS2STS0, PS2STS1 or PS2STS2) of the respective channel. Neither interrupts nor status bits are affected by reading alternate status registers.

15.0 FAN TACHOMETERS

15.1 Fan Tachometer Features

- 8-bit resolution
- Two independent channels
- Clock prescaler
- Programmable preload value
- Threshold detector
- Support for both polling and interrupt driven operation
- Wake up from Idle and Power Down modes

The SST79LF008 has two independent pulse counters gated by external signals. The most common application for these counters is fan tachometer, where the external gate signal is a square wave signal from the fan, with its frequency proportional to the fan speed. By measuring the period of the square wave, a fan tachometer can monitor a fan speed and also detect when a fan has seized. Fan tachometer input pins are FAN1 and FAN2. These pins are multiplexed with GPIO24 and GPIO25 respectively.

Tachometers use the 32.768 KHz oscillator clock as the time base source. The clock can be pre-scaled before being presented to a tachometer. On each rising edge of the FANn input, the preload value of the respective tachometer will be loaded into the 8 bit counter. The tachometer counter will then count up at each rising edge of the prescaled clock. A tachometer interrupt is generated if the counter reaches count C0H (192 decimal).

15.2 Fan Tachometer Operation

Each fan tachometer includes a Clock Prescaler, a Pulse Counter, a Preload register, a Read Latch, and a Threshold Detector. See Figure 15-1.

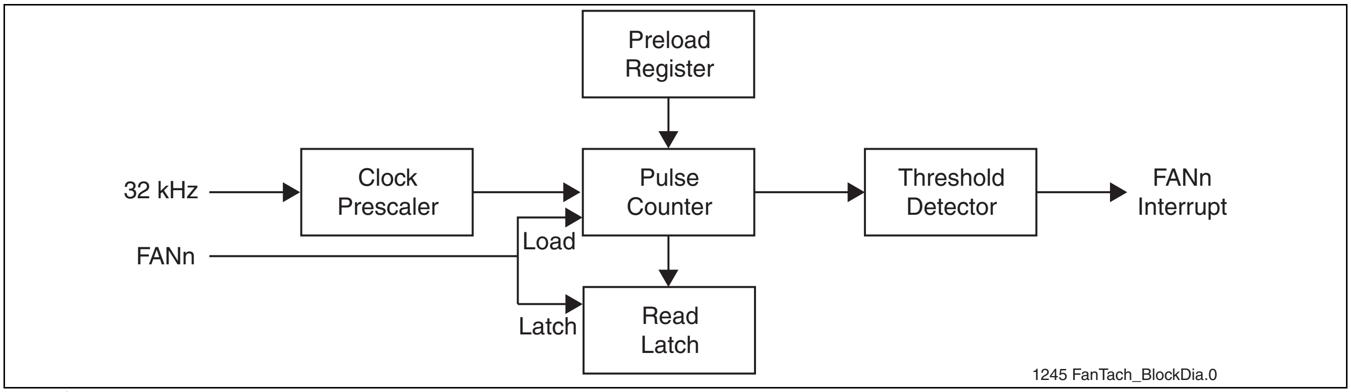


FIGURE 15-1: Fan Tachometer Block Diagram

The clock prescaler provides the fan tachometer time base. The frequency of the prescaler output clock is equal to 32.768 KHz times the prescaler ratio (1/2 by default). The prescaler ratio for each channel can be independently specified via a prescaler register from 1/1 to 1/8, which accommodates fans with a wide range of speed.

The prescaler output pulses are counted by the pulse counter during the FANn input signal period. The counter is incremented on the rising edge of the prescaler clock, and it does not wrap around when the maximum FFH (255 decimal) count is reached. On the rising edge of the FANn signal, the pulse counter value is latched into the read latch and the preload value is loaded into the pulse counter. Thus, the read latch contains the measurement of the last input signal period in time base units.

The threshold value for the fan speed threshold detector is fixed at C0H (192 decimal) count. However, the initial value for the pulse counter is programmable via the preload register. Hence, the pulse count can be scaled so that the threshold value corresponds to the desired lower limit of the fan speed. The preload value should be equal to 192 less the pulse count for the fan speed lower limit.

The counter is re-loaded with the preload value automatically on the FANn input signal rising edge, or when the software is writing to the preload register. Since software operations are asynchronous to the FANn signal, the tachometer reading may be incorrect until the second FANn rising edge after the write to the preload register.

When the pulse count equals or exceeds the threshold value, the respective FANn interrupt request is generated. The interrupt is cleared when preload register is written with the value below threshold.

The fan tachometers continue running in the Idle mode. Operation in the Power Down mode is controlled by software as shown Section 15.3.

15.3 Fan Tachometers MMCRs

15.3.1 Fan Tachometer 1 Read Register (FANCNT1)

Location		7	6	5	4	3	2	1	0
7FBAH	Read	FANCNT1_7	FANCNT1_6	FANCNT1_5	FANCNT1_4	FANCNT1_3	FANCNT1_2	FANCNT1_1	FANCNT1_0
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
FANCNT1[7:0]	Read-only register which returns the last latched Fan Tachometer 1 pulse count

15.3.2 Fan Tachometer 2 Read Register (FANCNT2)

Location		7	6	5	4	3	2	1	0
7FBBH	Read	FANCNT2_7	FANCNT2_6	FANCNT2_5	FANCNT2_4	FANCNT2_3	FANCNT2_2	FANCNT2_1	FANCNT2_0
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
FANCNT2[7:0]	Read-only register which returns the last latched Fan Tachometer 2 pulse count

15.3.3 Fan Tachometer 1 Preload Register (FAN1LD)

Location		7	6	5	4	3	2	1	0
7FBCH	Read	FAN1LD_7	FAN1LD_6	FAN1LD_5	FAN1LD_4	FAN1LD_3	FAN1LD_2	FAN1LD_1	FAN1LD_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
FAN1LD[7:0]	Preload value for Fan Tachometer 1 pulse counter. Pulse counter is loaded with this value when the preload register is written to by 8051 firmware, and each time on the rising edge of FAN1 input.

15.3.4 Fan Tachometer 2 Preload Register (FAN2LD)

Location		7	6	5	4	3	2	1	0
7FBDH	Read	FAN2LD_7	FAN2LD_6	FAN2LD_5	FAN2LD_4	FAN2LD_3	FAN2LD_2	FAN2LD_1	FAN2LD_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
FAN2LD[7:0]	Preload value for Fan Tachometer 2 pulse counter. Pulse counter is loaded with this value when the preload register is written to by 8051 firmware, and each time on the rising edge of FAN2 input.

15.3.5 Fan Tachometer Prescaler Register (FANTIMEBASE)

Location		7	6	5	4	3	2	1	0
7FBEH	Read	F2STOPEN	F1STOPEN	-	-	F2S1	F2S0	F1S1	F1S0
	Write								
	Reset	0	0	X	X	0	1	0	1

Symbol	Function
-	Not implemented
X	Not defined
F2STOPEN	Fan Tachometer 2 operation in Power Down mode control bit 1: Tachometer 2 keeps running when 8051 enters into Power Down mode 0: Tachometer 2 is stopped and loaded with preload value when 8051 enters into Power Down mode
F1STOPEN	Fan Tachometer 1 operation in Power Down mode control bit 1: Tachometer 1 keeps running when 8051 enters into Power Down mode 0: Tachometer 1 is stopped and loaded with preload value when 8051 enters into Power Down mode
F2S1, F2S0	Fan Tachometer 2 Prescaler ratio 00: prescaler ratio = 1/1 01: prescaler ratio = 1/2 10: prescaler ratio = 1/4 11: prescaler ratio = 1/8
F1S1, F1S0	Fan Tachometer 1 Prescaler ratio 00: prescaler ratio = 1/1 01: prescaler ratio = 1/2 10: prescaler ratio = 1/4 11: prescaler ratio = 1/8

16.0 ANALOG TO DIGITAL CONVERTER (ADC)

16.1 ADC Features

- 10-bit resolution
- Eight input channels
- Two conversion modes
  - Single mode: one-time A/D conversion of one channel
  - Continuous mode: continuous cyclical conversion on one to four channels
- Sample-and-hold input circuit
- A/D interrupt requested at the end of conversion
- Standby mode with low power consumption
- Automatic entry into standby mode when 8051 is in Power Down mode

An ADC block diagram is shown in Figure 16-1. In addition to being the analog power supply voltage, AV_{DD} is also used as the reference voltage for the A/D conversion. There are five 8-bit data registers that store up to four conversion results simultaneously. The eight analog input pins are divided into two groups: group 0 (ACH0 to ACH3), and group 1 (ACH4 to ACH7). Table 16-1specifies the relation between analog input channels and data registers.

TABLE 16-1: Analog Input Channels/Data Registers relationship

GROUP0	GROUP1	ADC Data Register
ACH0	ACH4	ADDRA[7:0]:ADDRL[1:0]
ACH1	ACH5	ADDRB[7:0]:ADDRL[3:2]
ACH2	ACH6	ADDRC[7:0]:ADDRL[5:4]
ACH3	ACH7	ADDRD[7:0]:ADDRL[7:6]

T16-1.0 1320

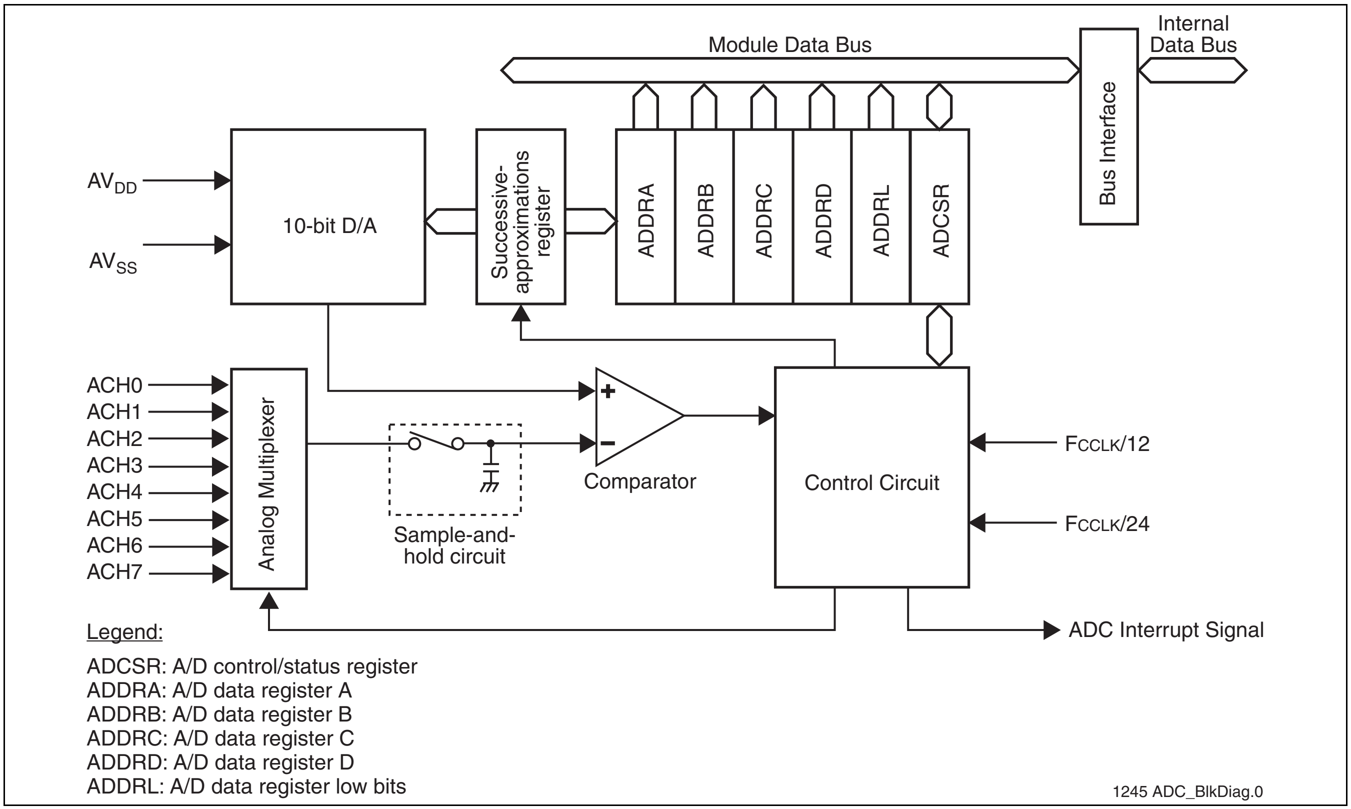


FIGURE 16-1: ADC Block Diagram

16.2 ADC MMCRs

There are six 8-bit read only A/D data registers that are used to store the A/D conversion results.

16.2.1 ADC Data register A (ADDRA)

Location		7	6	5	4	3	2	1	0
7F8EH	Read	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
AD[9:2]	The most significant 8 bits of A/D conversion result for channel ACH0 or ACH4

16.2.2 ADC Data Register B (ADDRB)

Location		7	6	5	4	3	2	1	0
7F8FH	Read	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
BD[9:2]	The most significant 8 bits of A/D conversion result for channel ACH1 or ACH5

16.2.3 ADC Data Register C (ADDRC)

Location		7	6	5	4	3	2	1	0
7F90H	Read	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
CD[9:2]	The most significant 8 bits of A/D conversion result for channel ACH2 or ACH6

16.2.4 ADC Data Register D (ADDRD)

Location		7	6	5	4	3	2	1	0
7F91H	Read	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
DD[9:2]	The most significant 8 bits of A/D conversion result for channel ACH3 or ACH7

16.2.5 ADC Data Register Lower Bits (ADDRL)

Location		7	6	5	4	3	2	1	0
7F92H	Read	DD1	DD0	CD1	CD0	BD1	BD0	AD1	AD0
	Write	-	-	-	-	-	-	-	-
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

- Not implemented

DD[1:0] The least significant 2 bits of A/D conversion result for channel ACH3 or ACH7 (the most significant 8 bits of A/D conversion result are stored in register ADDRD)

CD[1:0] The least significant 2 bits of A/D conversion result for channel ACH2 or ACH6 (the most significant 8 bits of A/D conversion result are stored in register ADDRC)

BD[1:0] The least significant 2 bits of A/D conversion result for channel ACH1 or ACH5 (the most significant 8 bits of A/D conversion result are stored in register ADDRB)

AD[1:0] The least significant 2 bits of A/D conversion result for channel ACH0 or ACH4 (the most significant 8 bits of A/D conversion result are stored in register ADDRA)

16.2.6 ADC Control and Status Register (ADCSR)

Location		7	6	5	4	3	2	1	0
7F93H	Read	ADF	ADCEN	ADST	SCAN	CKS	CH2	CH1	CH0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

ADF A/D conversion completion flag
Set by hardware in single mode when A/D conversion for the selected channel is completed. Set by hardware in continuous mode when A/D conversion cycle for all selected channels is completed. After ADF is set, the software must read ADCSR register first, then write 0 to ADF in order to clear this bit. Writing 1 to ADF bit will be ignored.

ADCEN Enable ADC bit
1: Enable ADC
0: Disable ADC, and switch ADC into standby mode
If this bit is set, ADC enters/exits standby mode automatically when 8051 enters/exits Power Down mode.

ADST A/D conversion Start bit
1: ADC conversion is started (in progress)
0: ADC conversion is stopped
This bit can be set or cleared by software in either single or continuous mode. It is cleared by hardware when conversion is completed in single mode only. The ADST bit is also automatically cleared when ADCEN bit is cleared.

SCAN A/D conversion mode selection bit
1: Continuous mode
0: Single mode

CKS A/D clock selection bit (A/D conversion time = 5 periods of ADC clock)
1: The frequency of ADC clock is FCCLK/12, FCCLK – 8051 core clock frequency
0: The frequency of ADC clock is FCCLK /24, FCCLK – 8051 core clock frequency
Note: ADC clock frequency must not exceed 2.0 MHz

CH[2:0] Analog input channels selection bits

16.3 ADC Operations

ADC implements a successive approximation algorithm with 10-bit resolution. It has two operating modes: single mode and continuous mode. To prevent incorrect results of the A/D conversion, the conversion mode, clock, and channel selection bits must be changed only when conversion is stopped (ADST = 0). It is acceptable to simultaneously write new values for the selection bits and to set the ADST bit in order to start a conversion.

TABLE 16-2: Channel and Mode Selection

Group selection bit	Channel selection bits		Selected Input channels	
CH2	**CH1**	**CH0**	**Single mode**	**Continuous mode**
0	0	0	ACH0	ACH0
		1	ACH1	ACH0-ACH1
	1	0	ACH2	ACH0-ACH2
		1	ACH3	ACH0-ACH3
1	0	0	ACH4	ACH4
		1	ACH5	ACH4-ACH5
	1	0	ACH6	ACH4-ACH6
		1	ACH7	ACH4-ACH7

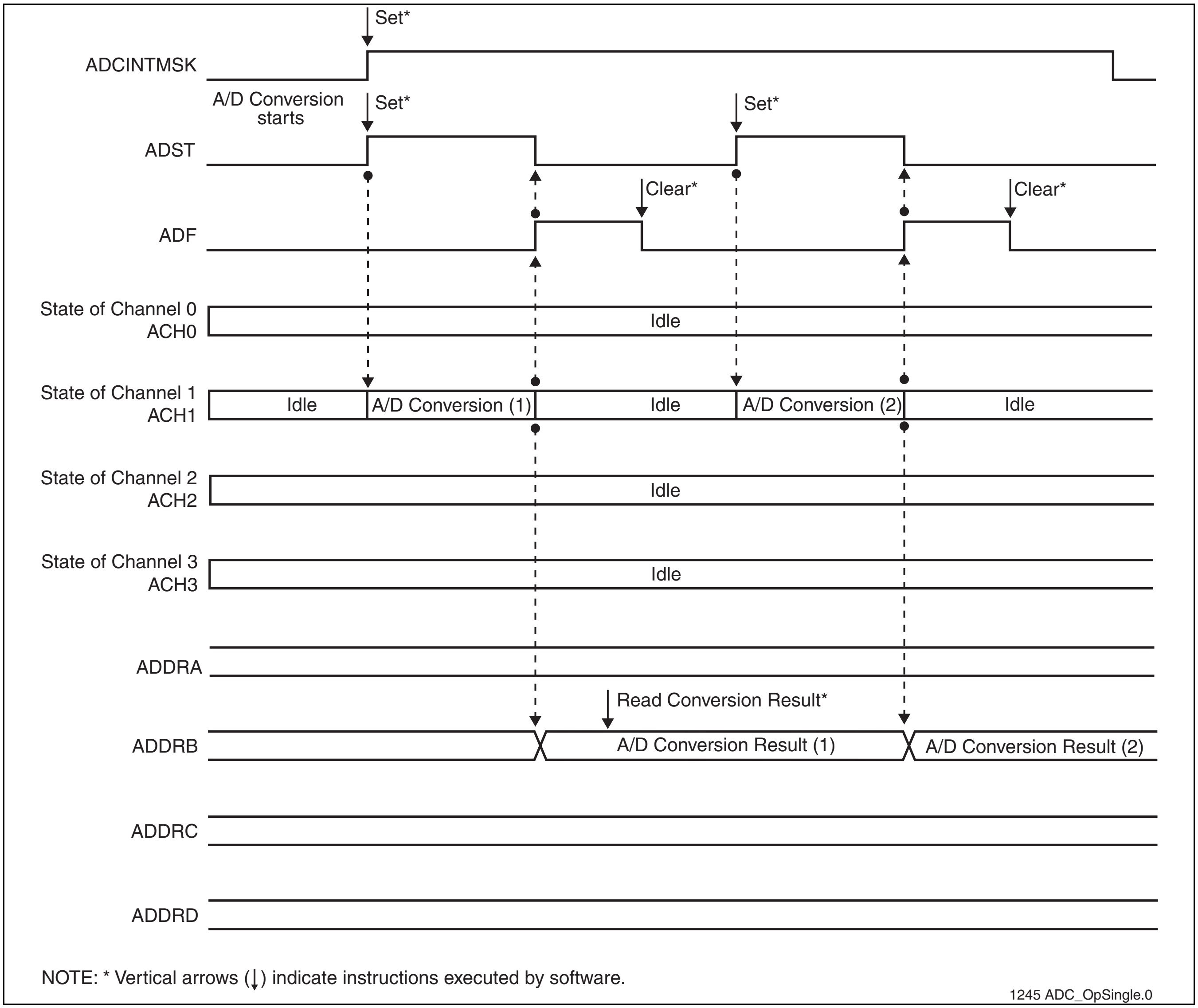
T16-2.0 1320

16.3.1 Single Mode

Single mode is used only when a one-time A/D conversion on one channel is required. A typical, single mode conversion, with channel 1 (ACH1) selected, is described below. It is assumed that ADC is enabled (ADCEN = 1).

1. The software enables the A/D interrupt (ADCINTMSK = 1), selects a single mode (SCAN = 0) with ACH1 as the input channel (CH2 = CH1 = 0, CH0 = 1), and starts the A/D conversation (ADST = 1).
2. When the A/D conversion is complete, the result is transferred by hardware into ADDRB and ADDRL[3:2] registers. Simultaneously, the ADF flag is set to 1, the ADST bit is cleared to 0, and the ADC is stopped.
3. An ADC interrupt is generated since ADF = 1 and ADCINTMSK = 1.
4. In response to the ADC interrupt, the software interrupt service routine reads ADCSR, and then writes 0 in the ADF flag. After which, the software reads and processes, if necessary, the conversion result for the ACH1 channel.

Using the software, the ADST bit can be set to 1 to start the next A/D conversion, and steps 2 through 4 are repeated. See Figure 16-2 for a timing diagram of this example.



NOTE: * Vertical arrows (↓) indicate instructions executed by software.

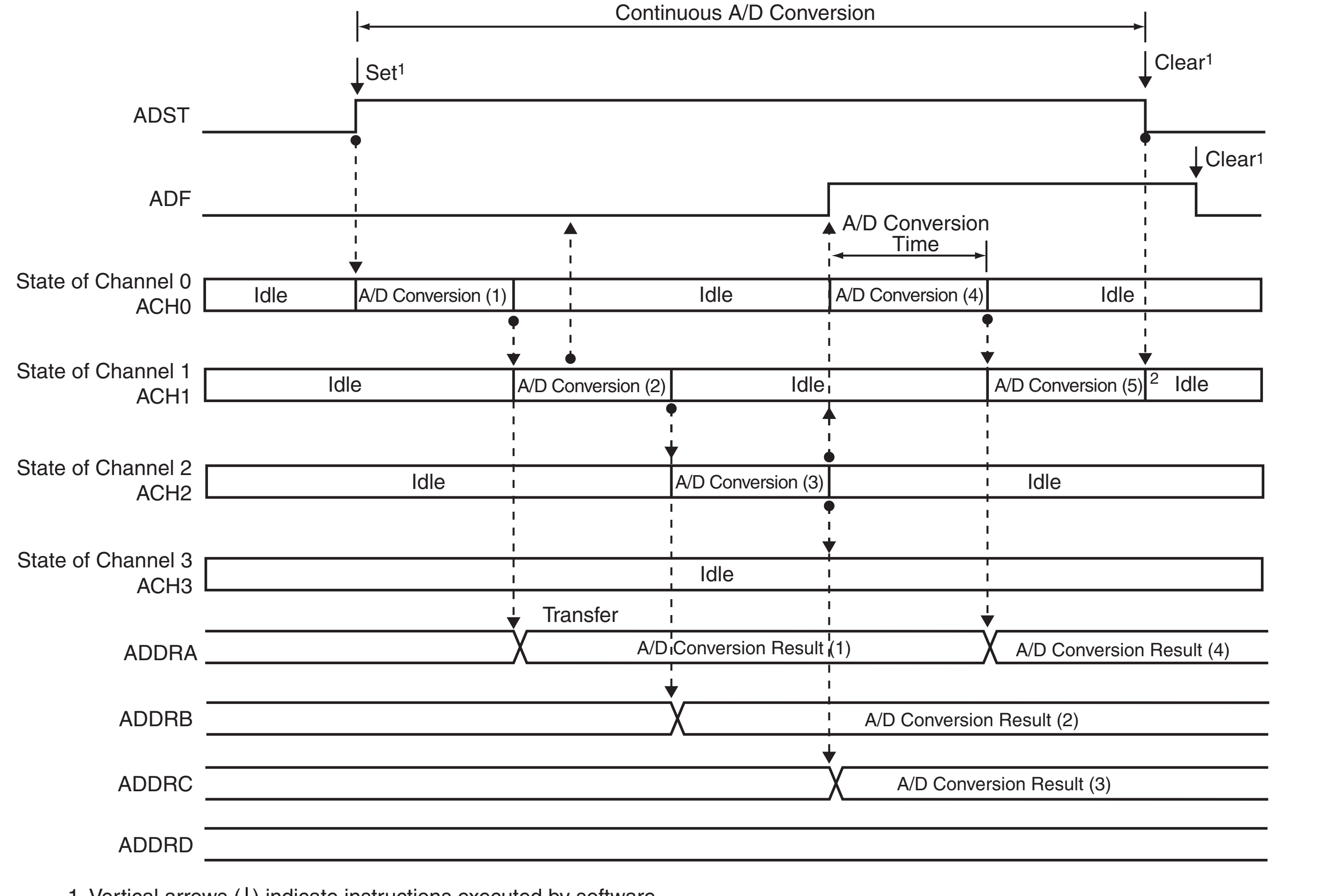
FIGURE 16-2: Example of ADC Operation (Single Mode)

16.3.2 Continuous Mode

Continuous mode is used to monitor analog inputs in a group of channels, with up to four channels per group. When the software sets the ADST bit to 1, the A/D conversion starts with the first channel in the group (ACH0 when CH2 = 0, ACH4 when CH2 = 1). If two or more channels are selected, the conversion of the second channel (ACH1 or ACH5) starts immediately after the conversion of the first channel is complete. The A/D conversion continues cyclically on all selected channels until the ADST bit is cleared to 0 by the software. The conversion results are stored in the ADC data registers. A typical continuous mode conversion, with three channels in group 0 (ACH0 to ACH2) selected, is described below. It is assumed that ADC is already enabled (ADCEN = 1).

1. The software enables the A/D interrupt (ADCINTMSK = 1), selects continuous mode (SCAN = 1), and scans group 0 (CH2 = 0) with input channels ACH0 to ACH2 (CH1 = 1, CH0 = 0), and then starts the A/D conversion (ADST = 1).
2. After the A/D conversion of the first channel (ACH0) is complete, the conversion result is transferred by hardware to ADDRA and ADDRL[1:0] registers.
3. The conversion of the second channel (ACH1) is started automatically and the results are stored in ADDRC and ADDRL[3:2] registers.
4. The conversion cycle proceeds similarly for the third channel (ACH2) with results stored in ADDRC and ADDRL[5:4] registers. When the conversion of all selected channels (ACH0 through ACH2) is complete, the ADF flag is set to 1 and the conversion of the first channel (ACH0 starts again from step 2.
5. An ADC interrupt is generated because ADF = 1 and ADCINTMSK = 1.
6. In response to the ADC interrupt, the software reads ADCSR and writes 0 in the ADF flag. After which, the software reads and processes, if necessary, the conversion results for the three selected channels.
7. If the ADST bit remains set to 1, steps 2 through 6 are repeated automatically. When the ADST bit is cleared to zero in the software, the A/D conversion stops.

Using the software, the ADST bit can be set to 1 to restart the A/D conversion cycle from the first channel (ACH0), beginning with step 2. See Figure 16-3 for a timing diagram of this example.



1. Vertical arrows (↓) indicate instructions executed by software.
2. Currently being converted is ignored.

1245 ADC_OpContinuous.0

FIGURE 16-3: Example of ADC Operation (Continuous Mode)

17.0 DIGITAL TO ANALOG CONVERTOR (DAC)

17.1 DAC Features

- 8-bit resolution
- Four D/A channels with one shared R-string (resistor ladder network) converter
- Sample-and-hold output circuits
- Independent enable/disable control for each channel
- 0V output for disabled channel
- Standby mode with low power consumption (all channels and shared R-string converter are disabled)
- Automatic entry into standby mode when 8051 is in Power Down mode

The DAC has one R-string digital-to-analog converter shared by 4 channels, see Figure 17-1. Each DAC channel generates an 8-bit resolution output and drives the corresponding output pin DAC0 to DAC3 via the sample-and-hold circuit.

The output voltage is determined by the value written to the respective DAC data register when the DAC channel is enabled. The output voltage is 0V, regardless of the value in the respective DAC data register, when the DAC channel is disabled to reduce power consumption.

After reset and in power down mode, all four channels, as well as the shared R-string, are disabled and the voltages on the DAC0-3 outputs are 0V. AV_{DD} analog power supply voltage is the reference voltage of the converter.

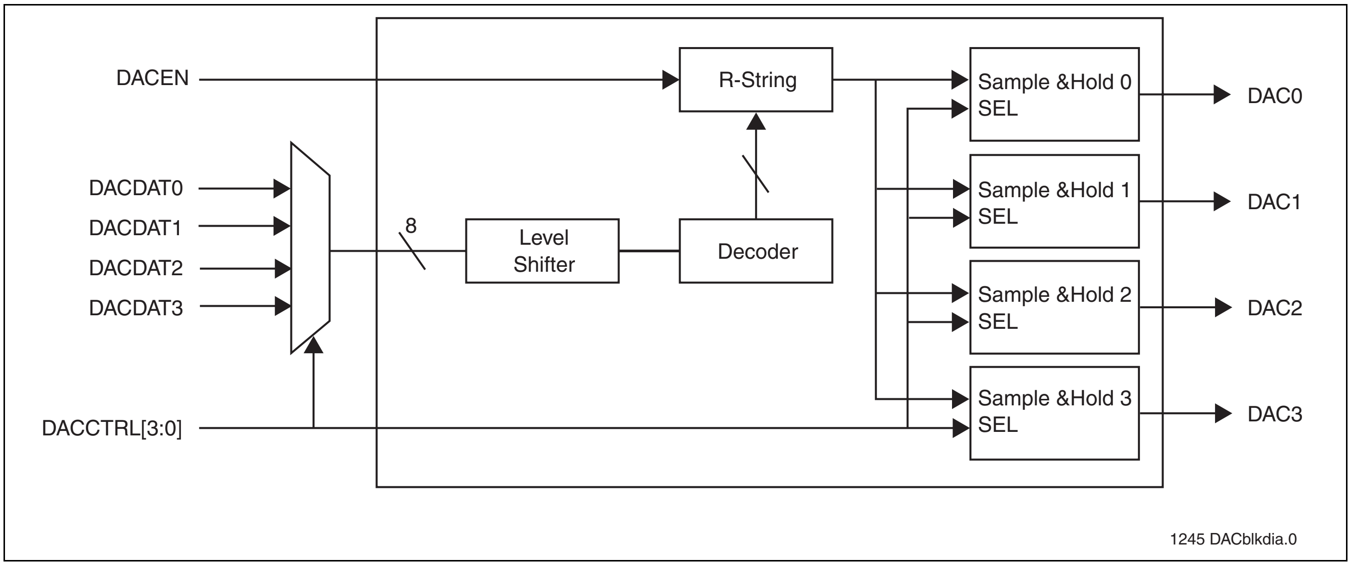


FIGURE 17-1: DAC Block Diagram

17.2 DAC MMCRs

17.2.1 DAC Data Channel Register 0 (DACDAT0)

Location		7	6	5	4	3	2	1	0
7F4CH	Read	DACDAT0_7	DACDAT0_6	DACDAT0_5	DACDAT0_4	DACDAT0_3	DACDAT0_2	DACDAT0_1	DACDAT0_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DACDAT0[7:0]	DAC channel 0 input data

17.2.2 DAC Data Channel Register 1 (DACDAT1)

Location		7	6	5	4	3	2	1	0
7F4DH	Read	DACDAT1_7	DACDAT1_6	DACDAT1_5	DACDAT1_4	DACDAT1_3	DACDAT1_2	DACDAT1_1	DACDAT1_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DACDAT1[7:0]	DAC channel 1 input data

17.2.3 DAC Data Channel Register 2 (DACDAT2)

Location		7	6	5	4	3	2	1	0
7F4EH	Read	DACDAT2_7	DACDAT2_6	DACDAT2_5	DACDAT2_4	DACDAT2_3	DACDAT2_2	DACDAT2_1	DACDAT2_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DACDAT2[7:0]	DAC channel 2 input data

17.2.4 DAC Data Channel Register 3 (DACDAT3)

Location		7	6	5	4	3	2	1	0
7F4FH	Read	DACDAT3_7	DACDAT3_6	DACDAT3_5	DACDAT3_4	DACDAT3_3	DACDAT3_2	DACDAT3_1	DACDAT3_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
DACDAT3[7:0]	DAC channel 3 input data

17.2.5 DAC Control Register (DACCTRL)

Location		7	6	5	4	3	2	1	0
7F50H	Read	-	-	-	DACEN	DACEN3	DACEN2	DACEN1	DACEN0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

DACEN DAC R-string Enable bit
1: Enable DAC R-string
0: Disable DAC R-string (standby mode DAC0-DAC3 output voltage is 0V)
If this bit is set, DAC also enters/exits standby mode automatically when 8051 enters/exits Power Down mode.

DACEN3 DAC Channel 3 Enable bit
1: Enable – DAC3 output voltage level is specified by the value in DACDAT3 register
0: Disable – DAC3 output voltage is 0V

DACEN2 DAC Channel 2 Enable bit
1: Enable – DAC2 output voltage level is specified by the value in DACDAT2 register
0: Disable – DAC2 output voltage is 0V

DACEN1 DAC Channel 1 Enable bit
1: Enable – DAC1 output voltage level is specified by the value in DACDAT1 register
0: Disable – DAC1 output voltage is 0V

DACEN0 DAC Channel 0 Enable bit
1: Enable – DAC0 output voltage level is specified by the value in DACDAT0 register
0: Disable – DAC0 output voltage is 0V.

17.3 DAC Operations

17.3.1 Output Voltage

For each channel, the DAC converts the input digital value stored in the respective DAC data register into an analog output voltage, relative to the analog ground pin (AV_{SS}). The analog power supply voltage AV_{DD} is used as the reference voltage of the converter. The output voltage level on DACn pin for enabled channel n = 0-3 can be found as follows: V_{OUT} = (DACDATn[7:0]) * (AV_{DD} / 256)

17.3.2 Conversion Cycle

Since there is only one R-string, an interleaving scheme is used to periodically convert each of the DAC data registers and refresh the DAC output Sample-and-hold circuits. When a DAC channel is selected for conversion, the content of the respective data register is copied into an 8-bit intermediate register, which holds the value throughout the entire conversion time slot allocated for this channel. At the end of the conversion time the corresponding DAC Sample-and-hold circuit is updated.

Each channel is allocated 24 core clocks for D/A conversion. A disabled channel will not be converted, but it will still occupy its time slot. Hence, the total conversion cycle for all 4 channels always occupies 96 core clock periods, i.e., conversion cycle time is equal to (96 * T_{CCLK}.).

17.3.3 DAC Channel Control

After SST79LF008 chip reset, the R-string and all DAC channels are disabled (DACCTRL register is cleared to 00H). In this state, no D/A conversion is performed, and the DAC is in standby mode with minimum current consumption.

The shared DAC R-string is enabled via the DACEN bit in the DACCTRL register. Each DAC channel can be enabled by setting the corresponding DACENn (n = 0 to 3) bit in the DACCTRL register. Once the DAC channel is enabled, D/A converter outputs the voltage level, which is specified by the respective data register DACDATn.

Each DAC channel can be independently disabled by clearing the DACENn bit in the DACCTRL register. If the DAC channel is disabled, its output voltage is 0V regardless of the value stored in the channel's data register.

17.3.4 Standby Mode

The DAC standby mode, with all DAC channels and the shared R-string disabled, provides the minimum possible DAC power consumption. All DAC0-DAC3 output voltages are forced to 0V in the standby mode.

The DAC switches to the standby mode automatically, when 8051 is in Power Down mode. This happens regardless of the state of the DACENn bits in the DACCTRL register.

18.0 KEYBOARD CONTROLLER HOST INTERFACE

18.1 Keyboard Controller Interface Overview

The SST79LF008 provides a 8042-style keyboard controller (KBC) host interface which is accessible via an LPC bus at standard I/O addresses 60H and 64H. This guarantees compatibility with the PC system BIOS and OS keyboard/ mouse services, as well as with any legacy applications that access keyboard ports directly.

The KBC interface includes the following 8-bit registers: KBC data write register, KBC data read register, KBC command write register, and KBC status register. The host processor accesses KBC interface registers at two addresses in the LPC I/O space. The 8051 core accesses the KBC interface registers at three addresses in external data memory space. Table 18-1 describes the register mapping to the host I/O space and 8051 memory space, as well as the access type for each register.

TABLE 18-1: Keyboard Controller Interface Mapping

LPC Host I/O Address and Access Type		Function	8051 Memory mapped Address and Access Type	
Address[1]	Access		Access	Address (MMCR register)
60H	Write	Host-to-KBC data write	Read	7FF1H (KBCDATA)
	Read	Host-from-KBC data read	Write	7FF1H (KBCDATA) or 7FFAH (AUXDATA)
64H	Write	Host-to-KBC command write	Read	7FF1H (KBCDATA)
	Read	Host-from-KBC status read	Write/Read	7FF2H (KBCSTS)

T18-1.1320

1. The default base address for KBC host interface ports can be changed via SST79LF008 configuration registers (see Section 23). For simplicity the description in Section 18.1 refers to default addresses.

When the Host writes a command byte to the KBC through port 64H, this sets the C/D bit in the KBC status register and the IBF bit in the KBC status register. When the Host writes a data byte to the KBC through port 60H, this clears the C/D bit in the KBC status register and sets the IBF bit in the KBC status register. When the Host reads data from the KBC through port 60H, the OBF bit in the KBC status register is cleared. See detailed bit description for the KBC status register in the next section.

The KBC interface also includes a mechanism for the generation of IRQ1 and IRQ12 interrupts to the LPC Host when data from the keyboard or mouse is ready to be read by the system.

18.2 Keyboard Controller Interface MMCRs

18.2.1 Keyboard Data Register (KBCDATA)

Location		7	6	5	4	3	2	1	0
7FF1H	Read Write	KBCDATA _7	KBCDATA _6	KBCDATA _5	KBCDATA _4	KBCDATA _3	KBCDATA _2	KBCDATA _1	KBCDATA _0
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
KBCDATA[7:0]	When the 8051 core reads from this register, the data returned is the last data byte written by the LPC Host to port 60H (if C/D bit is 0), or the last command byte written by the LPC Host to port 64H (if C/D bit is 1). The IBF bit in the status register is also cleared when the 8051 core reads this register. The 8051 core writes to this register the data byte from the KBC or keyboard, which will be returned to the LPC Host on the next read from port 60H. The OBF bit in the status register is set, and the internal KOBF signal is asserted when the 8051 core writes to this register. See Table 18-2.

18.2.2 Keyboard Status Register (KBCSTS)

Location		7	6	5	4	3	2	1	0
7FF2H	Read	UD	UD	AUXOBF/ UD	UD	C/D	UD	IBF	OBF
	Write					-		-	-
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

- Not implemented

UD User defined bits. Can be written/read by the 8051 core.

AUXOBF/UD Auxiliary Output Buffer Full flag (if AUXSEL=1) or User defined bit (if AUXSEL=0)
If AUXSEL=1 in KBDCFG register this read only bit is controlled by hardware as follows:
Set to ‘1’ when 8051 writes into the AUXDATA register at 7FFAH
Cleared to ‘0’ when 8051 writes into the KBCDATA register at 7FF1H
C/D Command/Data flag (read only)
Set to ‘1’ when the LPC Host writes command byte to port 64H
Cleared to ‘0’ when the LPC Host writes data byte to port 60H

IBF Input Buffer Full flag (read only)
Set when the LPC Host writes data or command to port 60H or 64H
Cleared when the 8051 core reads KBCDATA register at address 7FF1H
Interrupt request KCIBF to 8051 is asserted when this bit is set

OBF Output Buffer Full flag (read only)
Set when the 8051 core writes into KBCDATA register at address 7FF1H or into AUXDATA register at address 7FFAH
Cleared when the LPC Host reads port 60H
Interrupt request KCOBE to 8051 is asserted when this bit is cleared

18.2.3 Keyboard Auxiliary Data Register (AUXDATA)

Location		7	6	5	4	3	2	1	0
7FFAH	Read	AUXDATA _7	AUXDATA _6	AUXDATA _5	AUXDATA _4	AUXDATA _3	AUXDATA _2	AUXDATA _1	AUXDATA _0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

AUXDATA[7:0] When the 8051 core reads from this register the data returned is the last 8051 written data.
The 8051 core writes to this register the data byte from Mouse/Auxiliary Device which will be returned to the LPC Host on the next read from port 60H. The OBF bit in the status register is set, and the internal MOBF signal is asserted, when the 8051 core writes to this register. See Table 18-2.

18.2.4 Keyboard Controller Configuration Register (KBDCFG)

Location		7	6	5	4	3	2	1	0
7FF4H	Read	AUXSEL	-	OBFEN	AUX-OBFEN	-	PCOBFEN	SAEN	-
	Write								
	Reset	0	X	0	0	X	0	0	X

Symbol	Function
-	Not implemented
X	Not defined
AUXSEL	Enable hardware control of Auxiliary Output buffer full status flag 1: AUXOBF bit in the status register is controlled by hardware write at address 7FFAH or 7FF1H 0: AUXOBF bit in the status register is a user defined bit (UD) controlled by 8051 software.
OBFEN	KIRQ control bit 1: KIRQ follows PCOBF state 0: KIRQ is forced low (de-asserted) PCOBF is an internal signal, which reflects the status of 8051 and writes at address 7FF1H or at address 7FFDH. KIRQ is an internal source of IRQ1 interrupt for Serial IRQ transmission. See Figure 18-1 for KBC Interrupt Control diagram.
AUXOBFEN	MIRQ control bit 1: MIRQ follows MOBF state 0: MIRQ is forced low (de-asserted) MOBF is an internal signal, which reflects the status of 8051 and writes at address 7FFAH. MIRQ is an internal source of IRQ12 interrupt for Serial IRQ transmission. See Figure 18-1 for KBC Interrupt Control diagram.
PCOBFEN	Select PCOBF source 1: PCOBF signal reflects the value of bit 0 in the PCOBF register at address 7DDFH (PCOBF = PCOBFL) 0: PCOBF reflects the status of 8051 and writes to KBCDATA register at address 7FF1H (PCOBF = KOBF)
SAEN	GA20 Software control Enable bit 1: Enable software control of GA20 pin 0: Enable hardware control of GA20 pin

18.2.5 PCOBF Register (PCOBF)

Location		7	6	5	4	3	2	1	0
7FFDH	Read	-	-	-	-	-	-	-	PCOBFL
	Write								
	Reset	X	X	X	X	X	X	X	0

Symbol	Function
-	Not implemented
X	Not defined
PCOBFL	PCOBF firmware controlled latch PCOBF signal reflects the value of this bit provided PCOBFEN=1. See Figure 18-1

18.2.6 IRQ1 and IRQ12 Control

After SST79LF008 chip reset, all the KBC status flags and the internal control signals PCOBF, KOBF, and MOBF are reset to 0. Respectively, KIRQ and MIRQ requests are deasserted low, which results in both IRQ1 and IRQ12 being reported to the LPC Host as “low” via Serialized IRQ bus.

At run time the status flags and the internal control signals are changed in hardware according to Table 18-2.

TABLE 18-2: KBC Output Buffer Flags Control

Operation	MOBF@7FFA	KOBF@7FF1	OBF (KBCSTS[0])	AUXOBF[1] (KBCSTS[5])
8051 Writes to KBCDATA register at address 7FF1H	No Change	1	1	0
8051 Writes to AUXDATA register at address 7FFAH	1	No Change	1	1
LPC Host reads Port 60H	0	0	0	No Change

T18-2.1245

[1] This flag is controlled by hardware only if AUXSEL = 1

KIRQ and MIRQ interrupt requests reflect changing in the status flags according to Tables 18-3 and 18-4 and Figure 18-1. Whenever KIRQ/MIRQ is asserted, IRQ1/IRQ12 interrupts are reported to the LPC Host as “high” via Serialized IRQ bus

TABLE 18-3: KBC Interrupt Control

OBFEN (KBDCFG[5])	PCOBFEN (KBDCFG[2])	KIRQ
0	X[1]	KIRQ is inactive and driven low
1	0	KIRQ = KOBF
1	1	KIRQ = PCOBFL

T18-3.0 1245

[1] X = Not defined

TABLE 18-4: Mouse Interrupt Control

AUXOBFEN (KBDCFG[4])	MIRQ
0	MIRQ is inactive and driven low
1	MIRQ = MOBF

T18-4.0 1245

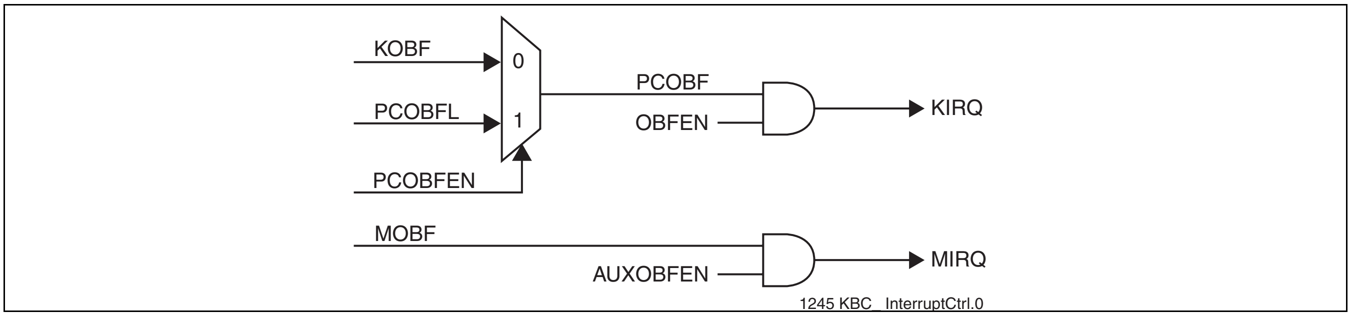


FIGURE 18-1: KBC Interrupt Control

18.3 Keyboard Matrix Scan Control

The SST79LF008 provides 16 scan outputs KSO[15:0] and 8 scan inputs KSI[7:0] for standard notebook keyboard matrix scanning. These lines can be accessed by 8051 firmware via the KEYSCAN register described below.

18.3.1 Keyboard Scan-In / Scan-Out Register (KEYSCAN)

Location		7	6	5	4	3	2	1	0
7F04H	Read	KSI7	KSI6	KSI5	KSI4	KSI3	KSI2	KSI1	KSI0
	Write	-	KSOINV	KSEN	KSOLOW	KSOC3	KSOC2	KSOC1	KSOC0
	Reset	0	0	1	0	0	0	0	0

Symbol	Function
-	Not implemented
KSI[7:0]	When the KEYSCAN register is read, it returns the state of KSI[7:0] pins via these read-only bits. Any KSI pin transitions from high to low will assert KEY interrupt request to 8051.
KSOINV	Scanner output polarity control bit (write-only)
KSEN	Scanner enable mask (write-only)
KSOLOW	Scanner low output control bit (write-only)
KSOC[3:0]	Scanner output selection bits (write-only)

TABLE 18-5: KSO[15:0] Control

KSEN	KSOLOW	KSOINV	KSOC[3:0]	KSO[15:0]
1	X[1]	X	X	All lines are high
0	1	1	X	All lines are high
0	1	0	X	All lines are low
0	0	1	n	KSO[n] line is high and all other lines are low
0	0	0	n	KSO[n] line is low, and all other lines are high (n = 15-0)

T18-5.0 1245

[1] X = Not defined

19.0 GA20 AND CPU RESET HARDWARE CONTROL

The keyboard controller GA20 output is a PC legacy feature, which provides capability to mask the address line A20 in order to emulate 8086 20-bit address space. Similarly the KBC to CPU reset output, KBRST#, is a legacy host CPU reset signal which can be triggered via the KBC. The SST79LF008 device contains on-chip logic to provide the host processor with direct control of GA20 and KBRST# outputs. This control is implemented via specific command/data sequences sent over LPC interface to ports 64H and 60H. Optionally, this on-chip logic can be disabled via SAEN bit in the KBDCFG register and SKBEN bit in the GA20 register, requesting control of the GA20 and/or KBRST# outputs to the 8051 firmware.

19.1 GA20 State Machine

Table 19-1 lists typical GA20 command sequences sent by the LPC Host to control the GA20 output from KBC. The hardware GA20 state machine, which interprets these sequences when GA20 hardware is enabled, is shown in Figure 19-1. After SST79LF008 chip reset, the state machine is in S0 state. Note that during a valid GA20 command sequence the IBF flag in the KBC status register is not ‘1’. This makes the hardware GA20 control (when SAEN = 0) transparent to 8051 firmware.

TABLE 19-1: GA20 Command Sequences

I/O Port	R/W	VALUE	IBF	GA20	Functions
64H 60H	W W	D1 DF	0 0	Q[1] 1	Set GA20 command
64H 60H	W W	D1 DD	0 0	Q 0	Clear GA20 command
64H 60H 64H	W W W	D1 DF FF	0 0 0	Q[1] 1 1	Extended Set GA20 command
64H 60H 64H	W W W	D1 DD FF	0 0 0	Q 0 0	Extended Clear GA20 command
64H 64H 64H	W W W	D1 XX[2] FF	0 1 1	Q Q 0	Invalid Sequence

T19-1.1320

[1] Q means no changes

[2] XX means any command code except D1

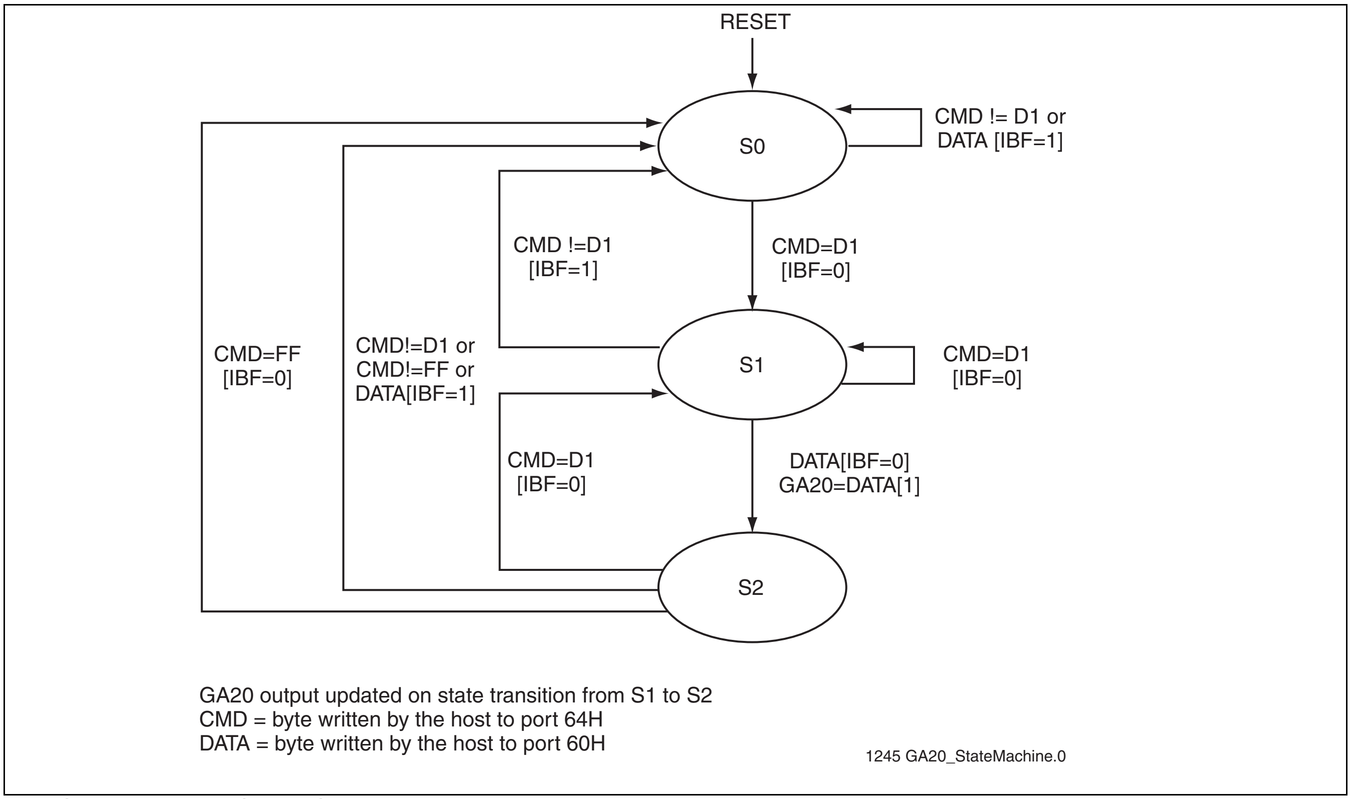


FIGURE 19-1: GA20 State Machine

19.2 GA20 and KBRST# MMCRs

19.2.1 GA20 Output Register (GA20)

Location		7	6	5	4	3	2	1	0
7FFBH	Read	-	-	-	-	SKBEN	KBRST	-	GA20
	Write								GA20_SW
	Reset	X	X	X	X	0	0	X	1

Symbol	Function
-	Not implemented
X	Not defined
SKBEN	KBRST# Software control Enable bit 1: Enable KBRST# software control 0: Enable KBRST# hardware control
KBRST	KBRST# Pulse generation control bit Set/cleared by software. When SKBEN = 1, the ‘0’ to ‘1’ transition of this bit generates a pulse on KBRST# output (low going pulse with duration more than 6µs).
GA20	GA20 software control bit/Status flag Reading this bit returns the present state of the GA20 output signal
GA20_SW	Writing to this bit set/reset software controlled GA20_SW signal (which is output to GA20 pin provided SAEN = 1)

When SKBEN = 0, KBRST# hardware control is enabled, a KBRST# low going pulse is generated automatically in response to the LPC Host command FEH written to port 64H (pulse duration is 200 cycles of LPC clock LCLK). In this case the FEH command does not set the IBF flag in the KBC status register. When SKBEN = 1, 8051 firmware controls KBRST# pin, and it can generate KBRST# low going pulse by writing ‘0’, and then ‘1’ to the KBRST bit (pulse duration is 200 cycles of 8051 core clock CCLK).

When SAEN’=’0, GA20 hardware control is enabled, the GA20 output is controlled by the LPC Host command/data sequences written to ports 64H/60H as shown on Figure 19-1 and in Table 19-1. Additionally, in this mode, the 8051 core can set/reset GA20 line via SETGA20 and RSTGA20 registers described below. Since the LPC Host GA20 sequence and 8051 writes to SETGA20/RSTGA20 registers, they asynchronously control the same GA20 output. It is necessary for 8051 to read back the GA20 status via GA20 register to confirm the actual GA20 state.

19.2.2 Set GA20 Register (SETGA20)

Location		7	6	5	4	3	2	1	0
7FFEH	Read	SGA207	SGA206	SGA205	SGA204	SGA203	SGA20	SGA201	SGA200
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
SGA20[7:0]	Any writes to this register sets hardware controlled GA20 asynchronously. Read from this register always returns 00H.

19.2.3 Reset GA20 Register (RSTGA20)

Location		7	6	5	4	3	2	1	0
7FFFH	Read	RGA207	RGA206	RGA205	RGA204	RGA203	RGA20	RGA201	RGA200
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
RGA20[7:0]	Any writes to this register resets hardware controlled GA20 asynchronously. Read from this register always returns 00H.

When SAEN = 1, only the 8051 core controls GA20 pin via direct writes to bit 0 of GA20 register. The LPC Host command detection is disabled when SAEN = 1, but writing to SETGA20 and RSTGA20 registers still affects the hardware controlled GA20_HW signal, even though it is not output to GA20 pin. Note that hardware and software controlled GA20 signals are independent. Figure 19-2 shows how the LPC Host and KBC firmware controls the GA20 output.

After SST79LF008 chip reset, the KBRST# signal is ‘1’, hardware controlled GA20_HW signal is ‘1’, and software controlled GA20_SW signal is ‘0’. Reset value of
SAEN = 0, thus, GA20 status is returned by default as GA20_HW = 1. Note that both GA20 and KBRST# signals are multiplexed with GPIO pins, and should be properly selected by 8051 firmware in order to utilize the respective functions.

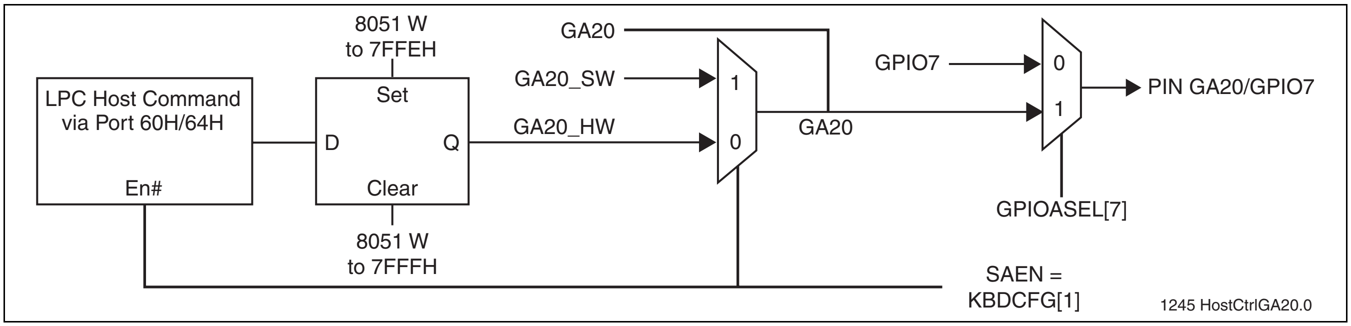


FIGURE 19-2: : Host and 8051 Control of GA20

20.0 ACPI EMBEDDED CONTROLLER INTERFACE

20.1 ACPI Embedded Controller Interface Overview

ACPI specification defines a hardware and software interface between the operating system and an embedded controller (EC). This interface can be used by the standard operating system driver to directly communicate with the embedded controller. SST79LF008 provides two2 ACPI compliant EC interfaces ECI0 and ECI1.

EC interface includes the following 8-bit registers: EC data write register, EC data read register, EC command write register, and EC status register. The host processor accesses EC interface registers at two addresses in the LPC I/O space. The 8051 core accesses EC interface registers at two addresses in the external data memory space. Figure 20-1 describes the register mapping to the host I/O space and 8051 memory space as well as access type for each register.

TABLE 20-1: Embedded Controller Interface Mapping

LPC Host I/O Address and Access Type		Function	8051 Memory mapped Address and Access Type	
Address[1]	Access		Access	Address (MMCR register)
62H for ECI0 68H for ECI1	Write	Host-to-EC data write	Read	7F53H (ECIDATA) for ECI0 7F80H (ECIDATA1) for ECI1
	Read	Host-from-EC data read	Write	
66H for ECI0 6CH for ECI1	Write	Host-to-EC command write	Read	7F53H (ECIDATA) for ECI0 7F80H (ECIDATA1) for ECI1
	Read	Host-from-EC status read	Write/Read	7F54H (ECISTS) for ECI0 7F81H (ECISTS1) for ECI1

T20-1.1245

[1] The default base address for EC host interface ports can be changed via SST79LF008 configuration registers (see Section 23.0). For simplicity the description in Section 20.1 refers to default addresses.

When the Host writes command byte to EC through port 66H (6CH), the ECISTSn_C/D bit is set and the ECISTSn_IBF bit is set in the respective EC status register. When the Host writes data byte to EC through port 62H (68H), the ECISTSn_C/D bit is cleared and the ECISTSn_IBF bit is set in the respective EC status register. When the Host reads data from EC through port 62H (68H), the ECISTSn_OBF bit in the respective EC status register is cleared. See detailed bit description for EC status registers in Section 20.2.

20.2 Embedded Controller Interface MMCRs

20.2.1 ECI Data Register (ECIDATA)

Location		7	6	5	4	3	2	1	0
7F53H	Read	ECIDATA_7	ECIDATA_6	ECIDATA_5	ECIDATA_4	ECIDATA_3	ECIDATA_2	ECIDATA_1	ECIDATA_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
ECIDATA[7:0]	When 8051 core reads from this register, data returned is the last data byte written by the LPC Host to port 62H (if ECISTSn_C/D bit in ECISTS register is 0), or the last command byte written by the LPC Host to port 66H (if ECISTSn_C/D bit is 1). The ECISTSn_IBF bit in ECISTS register is also cleared when 8051 core reads this register. When 8051 core writes to this register, it provides data to be returned to the LPC Host on the next read from port 62H. The ECISTSn_OBF bit in ECISTS register is also set when 8051 core writes to this register.

20.2.2 ECI Data Register 1 (ECIDATA1)

Location		7	6	5	4	3	2	1	0
7F80H	Read	ECIDATA1_7	ECIDATA1_6	ECIDATA1_5	ECIDATA1_4	ECIDATA1_3	ECIDATA1_2	ECIDATA1_1	ECIDATA1_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
ECIDATA1[7:0]	When 8051 core reads from this register, data returned is the last data byte written by the LPC Host to port 68H (if ECISTSn_C/D bit in ECISTS1 register is 0), or the last command byte written by the LPC Host to port 6CH (if ECISTSn_C/D bit is 1). The ECISTSn_IBF bit in ECISTS1 register is also cleared when 8051 core reads this register. When 8051 core writes to this register, it provides data to be returned to the LPC Host on the next read from port 68H. The ECISTSn_OBF bit in ECISTS1 register is also set when 8051 core writes to this register.

20.2.3 ECI Status Register (ECISTS)

Location		7	6	5	4	3	2	1	0
7F54H	Read	UD	ECISTS_ SMI_EVT	ECISTS_ SCI_EVT	ECISTS_ BURST	ECISTS_ C/D	UD	ECISTS_ IBF	ECISTS_ OBF
	Write					-		-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
UD	User defined bits. Can be written/read by 8051
ECISTS_SMI_EVT	SMI Event flag 1: SMI event is pending 0: No outstanding SMI events The ECISTS_SMI_EVT bit is a software controlled flag. Typically it is set when the embedded controller has detected an internal event that is to be processed by the system management interrupt SMI handler.
ECISTS_SCI_EVT	SCI Event flag 1: SCI event is pending 0: No outstanding SCI events The ECISTS_SCI_EVT bit is a software controlled flag. Typically it is set when the embedded controller has detected an internal event that is to be processed by the operating system driver that handles system control interrupt SCI.
ECISTS_BURST	Burst Mode flag 1: EC is in Burst Mode for polled command processing 0: EC is in normal mode for interrupt-driven command processing The ECISTSn_BURST bit is a software only controlled flag. It indicates the embedded controller resources are dedicated to processing EC command/data stream. Burst Mode speeds up communication with the operating system driver as it eliminates the overhead of SCIs processing.
ECISTS_C/D	Command/Data flag (read only) Set to ‘1’ when the LPC Host writes command byte to port 66H Cleared to ‘0’ when the LPC Host writes data byte to port 62H
ECISTS_IBF	Input Buffer Full flag (read only) Set when the LPC Host writes data or command to port 62H or 66H Cleared when 8051 reads ECIDATA register at address 7F53H Interrupt request ECIBF to 8051 is asserted when this bit is set
ECISTS_OBF	Output Buffer Full flag (read only) Set when 8051 writes into ECIDATA register at address 7F53H Cleared when the LPC Host reads port 62H Interrupt request ECOBE to 8051 is asserted when this bit is cleared

20.2.4 ECI Status Register 1 (ECISTS1)

Location		7	6	5	4	3	2	1	0
7F81H	Read	UD	ECISTS1_ SMI_EVT	ECISTS1_ SCI_EVT	ECISTS1 _BURST	ECISTS1 _C/D	UD	ECISTS1 _IBF	ECISTS1 _OBF
	Write					-		-	-
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
-	Not implemented
UD	User defined bits. Can be read/written by 8051.
ECISTS1_SMI_EVT	SMI Event flag 1: SMI event is pending 0: No outstanding SMI events The SMI_EVT bit is a software only controlled flag. Typically it is set when the embedded controller detects an internal event to be processed by the system management interrupt SMI handler.
ECISTS1_SCI_EVT	SCI Event flag 1: SCI event is pending 0: No outstanding SCI events The ECISTS_SCI_EVT bit is a software-only controlled flag. Typically it is set when the embedded controller has detected an internal event that is to be processed by the operating system driver that handles system control interrupt SCI.
ECISTS1_BURST	Burst Mode flag 1: EC is in Burst Mode for polled command processing 0: EC is in normal mode for interrupt-driven command processing. The ECISTSn_BURST bit is a software-only controlled flag. It indicates the embedded controller resources are dedicated to processing EC command/data stream. Burst Mode speeds up communication with the operating system driver as it eliminates the overhead of SCIs processing.
ECISTS1_C/D	Command/Data flag (read only) Set to ‘1’ when the LPC Host writes command byte to port 6CH Cleared to ‘0’ when the LPC Host writes data byte to port 68H
ECISTS1_IBF	Input Buffer Full flag (read only) Set when the LPC Host writes data or command to port 68H or 6CH Cleared when 8051 reads ECIDATA1 register at address 7F80H Interrupt request ECIBF1 to 8051 is asserted when this bit is set
ECISTS1_OBF	Output Buffer Full flag (read only) Set when 8051 writes into ECIDATA1 register at address 7F80H Cleared when the LPC Host reads port 68H Interrupt request ECOBE1 to 8051 is asserted when this bit is cleared.

20.3 SMI and SCI Control

In addition to the standard ACPI EC registers described in the Section 20.2, the SST79LF008 provides ECI configuration registers, which control SMI and SCI hardware generation from EC interface. As shown on Figure 20-1, both SMI and SCI interrupts are generated as active low level signals. A single SMI pin is shared by both ECI interfaces and mailbox interface as detailed in Section 21.0, and two SCI pins (EC_SCI and EC1_SCI) are dedicated to EC interfaces (ECI0 and ECI1, respectively).

20.3.1 ECI Configuration Register (ECICFG)

Location		7	6	5	4	3	2	1	0
7F51H	Read	-	-	ECICFG_SMIW	ECICFG_SMISEL	ECICFG_SCIW	ECICFG_SCISEL	ECICFG_SCIEN	ECICFG_SMIEN
	Write								
	Reset	X	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
ECICFG_SMIW	SMI generation control bit 1: Generate SMI from ECI0 0: Do not generate SMI from ECI0
ECICFG_SMISEL	SMI source selection bit 1: Select ECISTSn_OBF (ECISTS.0) as SMI source (ECISTSn_OBF = 1 will cause SMI if ECICFGn_SMIEN is set) 0: Select ECICFGn_SMIW (ECICFG.5) as SMI source (ECICFGn_SMIW = 1 will cause SMI if ECICFGn_SMIEN is set)
ECICFG_SCIW	SCI generation control bit 1: Generate SCI from ECI0 0: Do not generate SCI from ECI0
ECICFG_SCISEL	SCI source selection bit 1: Select ECISTSn_OBF (ECISTS.0) as SCI source (ECISTSn_OBF = 1 will cause SCI if ECICFGn_SCIEN is set) 0: Select ECICFGn_SCIW (ECICFG.3) as SCI source (ECICFGn_SCIW = 1 will cause SCI if ECICFGn_SCIEN is set)
ECICFG_SCIEN	SCI generation enable bit 1: Enable SCI generation from ECI0 0: Disable SCI generation from ECI0
ECICFG_SMIEN	SMI generation enable bit 1: Enabled SMI generation from ECI0 0: Disable SMI generation from ECI0

20.3.2 ECI Configuration Register 1(ECICFG1)

Location		7	6	5	4	3	2	1	0
7F52H	Read	-	-	ECICFG1 _SMIW	ECICFG1 _SMISEL	ECICFG1 _SCIW	ECICFG1 _SCISEL	ECICFG1 _SCIEN	ECICFG1_ SMIEN
	Write								
	Reset	X	X	0	0	0	0	0	0

Symbol	Function
-	Not implemented
X	Not defined
ECICFG1_MIW	SMI generation control bit 1: Generate SMI from ECI1 0: Do not generate SMI from ECI1
ECICFG1_SMISEL	SMI source selection bit 1: Select ECISTSn_OBF (ECISTS1.0) as SMI source (ECISTSn_OBF = 1 will cause SMI if ECICFGn_SMIEN is set) 0: Select ECICFGn_SMIW (ECICFG1.5) as SMI source (ECICFGn_SMIW = 1 will cause SMI if ECICFGn_SMIEN is set)
ECICFG1_SCIW	SCI generation control bit 1: Generate SCI from ECI1 0: Do not generate SCI from ECI1
ECICFG1_SCISEL	SCI source selection bit 1: Select ECISTSn_OBF (ECISTS1.0) as SCI source (ECISTSn_OBF = 1 will cause SCI if ECICFGn_SCIEN is set) 0: Select ECICFGn_SCIW (ECICFG1.3) as SCI source (ECICFGn_SCIW = 1 will cause SCI if ECICFGn_SCIEN is set)
ECICFG1_SCIEN	SCI generation enable bit 1: Enable SCI generation from ECI1 0: Disable SCI generation from ECI1
ECICFG1_SMIEN	SMI generation enable bit 1: Enabled SMI generation from ECI1 0: Disable SMI generation from ECI1

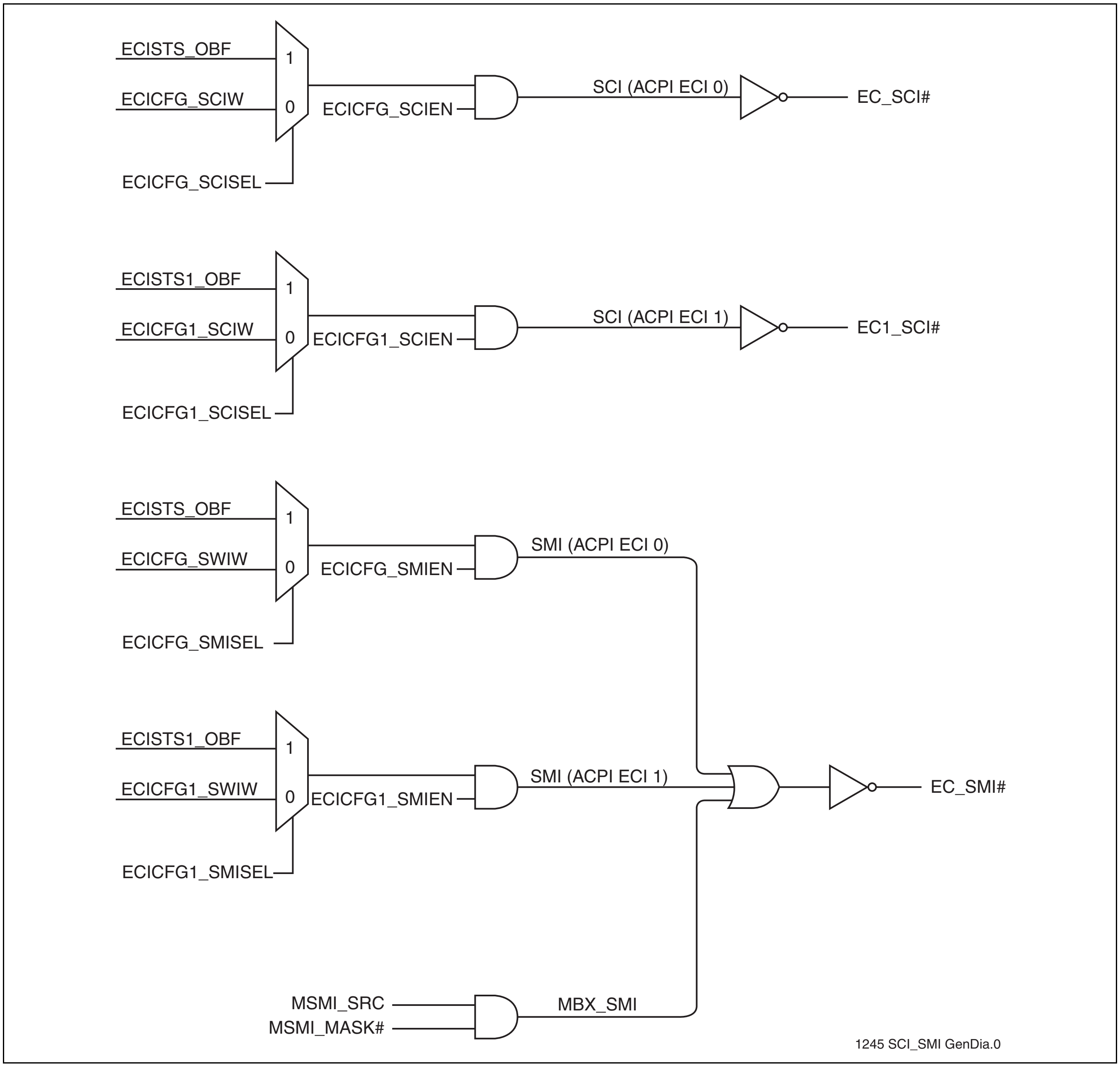


FIGURE 20-1: SCI and SMI Generation Diagram

21.0 MAILBOX INTERFACE AND DATA TRANSFER BLOCK

The SST79LF008 mailbox interface provides the LPC Host and 8051 with an additional mechanism for software controlled communications. The mailbox (MBX) interface includes 32 command/data transfer registers, and 3 control registers.

Mailbox registers are accessed by the Host via a pair of Index/Data ports mapped into the LPC I/O space. The default LPC I/O address of the MBX Index port is 00H and of the MBX Data port is 01H. The default addresses can be changed by 8051 firmware during the SST79LF008 initial configuration as described in Section 23.0. In order to access a mailbox register, the Host must write the respective access index to the MBX Index port and then read/ write data from/to the MBX data port. The 8051 can access any mailbox transfer register directly at the assigned address in the 8051 external data memory space. Refer to Table 21-1 and Table 21-2 for mailbox registers Host access indexes and 8051 memory addresses.

21.1 Mailbox Command/Data Transfer Registers

TABLE 21-1: Mailbox Command/Data Transfer Registers Map

Mailbox name	Host Acess Index	8051 Address	Function
Mailbox register 0	82H	7F08H	Host-to-8051 Mailbox command register
Mailbox register 1	83H	7F09H	8051-to-Host Mailbox command register
Mail box register 2	84H	7F0AH	Mailbox data transfer register
Mail box register 3	85H	7F0BH	Mailbox data transfer register
Mailbox register 4	86H	7F0CH	Mailbox data transfer register
Mailbox register 5	87H	7F0DH	Mailbox data transfer register
Mailbox register 6	88H	7F0EH	Mailbox data transfer register
Mailbox register 7	89H	7F0FH	Mailbox data transfer register
Mailbox register 8	8AH	7F10H	Mailbox data transfer register
Mailbox register 9	8BH	7F11H	Mailbox data transfer register
Mailbox register A	8CH	7F12H	Mailbox data transfer register
Mailbox register B	8DH	7F13H	Mailbox data transfer register
Mailbox register C	8EH	7F14H	Mailbox data transfer register
Mailbox register D	8FH	7F15H	Mailbox data transfer register
Mailbox register E	90H	7F16H	Mailbox data transfer register
Mailbox register F	91H	7F17H	Mailbox data transfer register
Mailbox register 10	A0H	7F70H	Mailbox data transfer register
Mailbox register 11	A1H	7F71H	Mailbox data transfer register
Mailbox register 12	A2H	7F72H	Mailbox data transfer register
Mailbox register 13	A3H	7F73H	Mailbox data transfer register
Mailbox register 14	A4H	7F74H	Mailbox data transfer register
Mailbox register 15	A5H	7F75H	Mailbox data transfer register
Mailbox register 16	A6H	7F76H	Mailbox data transfer register
Mailbox register 17	A7H	7F77H	Mailbox data transfer register
Mailbox register 18	A8H	7F78H	Mailbox data transfer register
Mailbox register 19	A9H	7F79H	Mailbox data transfer register
Mailbox register 1A	AAH	7F7AH	Mailbox data transfer register
Mailbox register 1B	ABH	7F7BH	Mailbox data transfer register
Mailbox register 1C	ACH	7F7CH	Mailbox data transfer register
Mailbox register 1D	ADH	7F7DH	Mailbox data transfer register

TABLE 21-1: Mailbox Command/Data Transfer Registers Map (Continued)

Mailbox name	Host Acess Index	8051 Address	Function
Mailbox register 1E	AEH	7F7EH	Mailbox data transfer register
Mailbox register 1F	AFH	7F7FH	Mailbox data transfer register

T21-1.1320

21.1.1 Mailbox register 0 (MBX0)

Location		7	6	5	4	3	2	1	0
7F08H	Read	MBX0_7	MBX0_6	MBX0_5	MBX0_4	MBX0_3	MBX0_2	MBX0_1	MBX0_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
MBX0[7:0]	Host-to-8051 Mailbox command register When the Host writes to this register, an 8051 interrupt request MBXINT is asserted. The interrupt request is cleared when the 8051 reads data from this register. When the 8051 writes to this register, the data is ignored and the register is reset to 00H.

21.1.2 Mailbox register 1 (MBX1)

Location		7	6	5	4	3	2	1	0
7F09H	Read	MBX1_7	MBX1_6	MBX1_5	MBX1_4	MBX1_3	MBX1_2	MBX1_1	MBX1_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
MBX1[7:0]	8051-to-Host Mailbox command register When the 8051 writes to this register, a mailbox SMI source bit, MSMI_SRC is asserted. The SMI source is cleared when the Host reads data from this register. When the Host writes to this register, the data is ignored and the register is reset to 00H.

The mailbox registers 0 and 1 can be used by the Host software and 8051 firmware to create mailbox command protocol(s), and to provide a hand-shaking mechanism for shared access to the other 30 mailbox data transfer registers.

21.1.3 Mailbox registers 2-1F (MBX2 – MBX1F)

Location		7	6	5	4	3	2	1	0
(see Table 21-1)	Read	MBXn_7	MBXn_6	MBXn_5	MBXn_4	MBXn_3	MBXn_2	MBXn_1	MBXn_0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
MBXn[7:0]	Mailbox data transfer register (n = 2-1F) General purpose data transfer registers. Can be read/written by both Host and 8051

21.2 Mailbox Control Registers

Only the LPC Host can access the Mailbox control registers. These registers are not mapped into 8051 memory space, and therefore can only be identified by the Host access indexes. These control registers are shown in Table 21-2.

TABLE 21-2: Mailbox Control Registers Map

Mailbox name	HOST ACCESS INDEX	FUNCTION
Mailbox register 94	94H	Host control of 8051 clock and shared flash access
Mailbox register 96	96H	MBX SMI source register
Mailbox register 97	97H	MBX SMI mask register

T21-2.1245

21.2.1 Mailbox register 94 (MBX94)

Location		7	6	5	4	3	2	1	0
(see Table 21-2)	Read	IDLE	HOST-FLASH	-	MAP	EXECU-TION	-	-	STP_CLK
	Write	-	-			-			
	Reset	0	0	X	0	0	X	X	0

Symbol	Function
-	Not implemented
X	Not defined
IDLE	8051 Idle mode status flag 1: 8051 is in Idle mode 0: 8051 is not in Idle mode
HOSTFLASH	Flash interface ownership flag 1: LPC Host owns flash interface 0: 8051 owns flash interface This bit is set when the shared flash interface is released to the host because either (a) 8051 is running from the scratch ROM and the HOST_ACCESS bit in the SFSC register is set, or (b) 8051 is in Idle mode and 8051 core clock is stopped by setting the STP_CLK bit in MBX94 (this register).
MAP	KBC flash mapping control bit 1: KBC flash area is mapped to LPC space 0: KBC flash area is not mapped to LPC space (LPC read access to KBC area returns 00H) The KBC flash area is Block0-Block1 = 128 KByte, or Block0 = 64 KByte depending on the status of ACON[1] bit – see Section 6.5. Note that if the 8051 doesn’t map KBC flash area to the LPC space, then this bit is ignored. If the 8051 maps the KBC flash to the LPC space, then this bit controls whether the KBC flash area is visible to the LPC Host or not.
EXECUTION	LPC program/erase operation acceptance indicator 1: Last LPC Host triggered program/erase operation is accepted 0: Last LPC Host triggered program/erase operation has been ignored
STP_CLK	Stop 8051 clock request bit Set by the LPC Host to stop 8051 clock in order to gain access to the shared flash interface (this bit should be set only after 8051 has been put into Idle mode) Cleared by the LPC Host to signal the release of shared flash interface

21.2.2 Mailbox register 96 (MBX96)

Location		7	6	5	4	3	2	1	0
(see Table 21-2)	Read	-	-	-	-	MSMI_SRC	-	-	-
	Write					-			
	Reset	X	X	X	X	0	X	X	X

Symbol	Function
-	Not implemented
X	Not defined
MSMI_SRC	Mailbox SMI source bit Set when 8051 writes to 8051-to-Host Mailbox register 1 Cleared when the LPC Host reads Mailbox register 1

21.2.3 Mailbox register 97 (MBX97)

Location		7	6	5	4	3	2	1	0
(see Table 21-2)	Read	-	-	-	-	MSMI_MSK	-	-	-
	Write								
	Reset	X	X	X	X	0	X	X	X

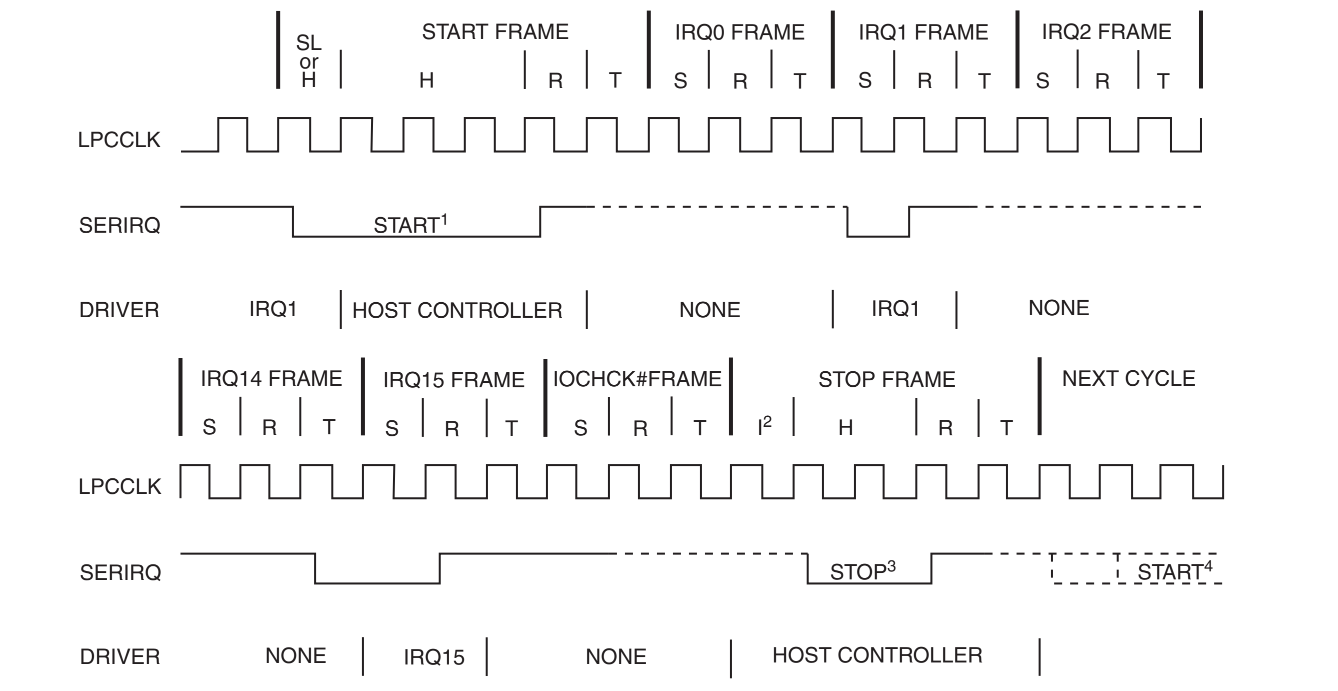
Symbol	Function
-	Not implemented
X	Not defined
MSMI_MSK	Mailbox SMI mask 1: Mask Mailbox SMI 0: Enable Mailbox SMI This bit affects both possible mechanisms for Mailbox SMI reporting (SMI# pin and Serialized IRQ2).

22.0 SERIALIZED INTERRUPTS

The SST79LF008 device provides serialized interrupt output, SERIRQ, which can be used to report interrupts from SST79LF008 to the LPC Host according to the *Serialized IRQ Specification for PCI Systems, Revision 6.0*.

22.1 Serialized IRQ Cycle Overview

An example of the Serialized IRQ cycle is shown on Figure 22-1. The cycle always begins with the Start frame and ends with the Stop frame. There are maximum 32 IRQ/ Data frames between the start and stop frame. Each of the data frames includes three phases: Sample phase, Recovery phase, and Turn-around phase. The SERIRQ is considered *Idle* between Stop and Start Frames. The SERIRQ is *Active* between Start and Stop Frames.



Legend: H = Host Control SL = Slave Control R = Recovery T = Turn-around S = Sample

1. Start Frame pulse can be 4-8 clocks wide.
2. There may be none, or one or more idles states (I) during the STOP frame.
3. Stop Frame is two clocks wide for Quiet mode, three clocks wide for continuous mode.
4. The next SERIRQ cycle's Start Frame pulse may or may not start immediately after the turn-around of the Stop frame.

1245 SerialIRQ cycle.0

FIGURE 22-1: Serialized IRQ cycle.

22.2 Serialized IRQ Start Frame

There are two modes of operation for the generation of SERIRQ Start Frame: Continuous and Quiet mode.

In Continuous mode, the SST79LF008 does not generate a Start Frame. The device just monitors SERIRQ input and waits for the LPC Host to initiate the Start Frame by driving SERIRQ line low for four to eight clocks.

In Quite mode, the SST79LF008 generates the Start Frame when it detects any transition of the internal IRQ/ Data signals associated with serialized IRQs, see Section 22.3. The device will not generate the start frame if the SERIRQ is already Active, and the IRQ/Data transition can be reported in the current SERIRQ Cycle.

In order to initiate the Start Frame, the SST79LF008 drives the SERIRQ line low for one clock, and then immediately tri-states the line. The Host controller takes over driving the SERIRQ low during the next clock and will continue driving it for a period of three to seven clocks. Thus a total Start Frame low pulse width is from four to eight clocks.

After SST79LF008 chip reset, as well as after LPC Interface reset, the SST79LF008 is in the Continuous mode, therefore only the LPC Host can initiate the first Start Frame. A SERIRQ mode transition can only occur during the Stop Frame as described in Section 22.4.

22.3 Serialized IRQ Data Frame

After a Start Frame low going edge has been initiated, SST79LF008 waits for the rising edge of the start pulse in order to start counting IRQ/Data Frames. Each IRQ/Data Frame has three clocks: Sample phase, Recovery phase, and Turn-around phase. During the sample phase, the SST79LF008 drives the SERIRQ pin low provided the last sampled IRQ/Data value associated with the respective frame was low. If the last sampled IRQ/Data value was high, or no IRQ/Data signal is associated with the respective frame, the SERIRQ line is left tri-stated. During the recovery phase, the SST79LF008 drives the SERIRQ high, if and only if, it had driven the SERIRQ low during the previous sample phase. During the turn-around phase, the SST79LF008 tri-states the SERIRQ. The above rules of Data Frame control are followed by the SST79LF008 regardless of which device has initiated the Start Frame.

The Sample phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame number times three, minus one. The only three internal IRQ/Data signals that are actually sampled by the SST79LF008 are: KIRQ (associated typically with IRQ1 frame via configuration registers described in Section 23.0), MIRQ (associated typically with IRQ12 frame via configuration registers described in Section 23.0), and Mailbox SMI (associated with IRQ2 frame if SMIEN_IRQ2 bit in configuration space is set). Table 22-1 shows the default SERIRQ sampling periods for these three internal IRQ/Data signals. For all other non-associated frames the SERIRQ line is left tri-stated by the SST79LF008.

TABLE 22-1: SST79LF008 SERIRQ Sampling Periods

IRQ/Data Frame number	Reported System IRQ	SST79LF008 Signal Sampled	Number of clocks past Start
2	IRQ1	KIRQ (Section 18.0)	5
3	IRQ2	MSMI_SRC (Section 20.0)	8
13	IRQ12	MIRQ (Section 18.0)	38

T22-1.1245

22.4 Serialized IRQ Stop Frame

After all IRQ/Data Frames are completed the SERIRQ cycle is terminated by a Stop Frame, which is indicated by the SERIRQ line being kept low for two or three clocks. Only the Host controller can initiate the Stop Frame. If the Stop Frame low time is two clocks, then the next SERIRQ cycle is in the Quiet mode, and the SST79LF008 may initiate a Start Frame. If the Stop Frame low time is three clocks, then the next SERIRQ cycle is in the Continuous mode, and only the Host may initiate a Start Frame. In any mode, the next Start Frame can be initiated once two or more clocks have occurred after the rising edge of the Stop Frame's pulse.

23.0 SST79LF008 CONFIGURATION

The SST79LF008 configuration module provides 8051 firmware with a flexible mechanism to relocate I/O interfaces within the LPC Host I/O address space.

23.1 Access to Configuration Registers

The 8051 access to configuration space is controlled by the SELCFG bit in the LPC bus monitor register, LPCMON. Any configuration register is accessed via a pair of CFGINDEX and CFGDATA ports. The LPCMON, CFGINDEX PORT and CFGDATA PORT registers are mapped into the 8051 external data memory address space as shown in Registers 23.1.1-23.1.3.

23.1.1 LPC Bus Monitor Register (LPCMON)

Location		7	6	5	4	3	2	1	0
7F8AH	Read	SELCFG	LPC-MODE	-	-	-	-	LRST-COREENB	LPCPD
	Write								-
	Reset	0	0	X	X	X	X	0	-

Symbol	Function
-	Not implemented
X	Not defined
SELCGF	Configuration space access control bit 1: Reserved. Do not use this setting. 0: Enable 8051 access to configuration registers (must always be enabled).
LPCMODE	LPC Memory cycle control bit 1: SST79LF008 responds to LPC Memory cycles on LPC bus. Firmware Memory cycles are ignored. 0: SST79LF008 responds to Firmware Memory cycles on LPC bus. LPC Memory cycles are ignored.
LRSTCOREENB	LPC Soft reset control bit 1: LPC commands “Force/Release LPC Soft Reset” are ignored. 0: LPC commands “Force/Release LPC Soft Reset” are accepted. This bit can be set by 8051 firmware, but it is cleared only by SST79LF008 chip reset.
LPCPD	LPCPD signal status flag (read only) This bit is equal to the inverse of LPCPD# input pin (reset value is not specified as it is passed through pin state).

23.1.2 Configuration INDEX PORT Register (CFGINDEX)

Location		7	6	5	4	3	2	1	0
7F8CH	Read	CFGINDEX _7	CFGINDEX _6	CFGINDEX _5	CFGINDEX _4	CFGINDEX _3	CFGINDEX _2	CFGINDEX _1	CFGINDEX _0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol	Function
CFGINDEX[7:0]	Configuration register index

23.1.3 Configuration DATA PORT Register (CFGDATA)

Location		7	6	5	4	3	2	1	0
7F8DH	Read	CFGDATA _7	CFGDATA _6	CFGDATA _5	CFGDATA _4	CFGDATA _3	CFGDATA _2	CFGDATA _1	CFGDATA _0
	Write								
	Reset	0	0	0	0	0	0	0	0

Symbol **Function**

CFGDATA[7:0] Configuration register data

Configuration Registers Map is shown in the Table 23-1.

To access any Global Configuration register (index 00H-2FH) the 8051 firmware should do the following.

1. Write the index of the configuration register into the CFGINDEX PORT.
2. Write/read the Global Configuration register through the CFGDATA PORT.

To access any Device Configuration register (index 30H and above) the 8051 firmware should do one of the following steps.

a)

1. Write 07H (the index of the Logical Device Number register) to the CFGINDEX PORT.
2. Write the number of the targeted logical device to the CFGDATA PORT.

b)

1. Write the address of the Device Configuration register to the CFGINDEX PORT.
2. Write/read the Device Configuration register through the CFGDATA PORT.

23.2 Configuration Registers Description

TABLE 23-1: Configuration Registers Map[1]

Index	Access Type	Reset Value[2]	Configuration Register Name
GLOBAL CONFIGURATION REGISTERS			
07H	R/W	00H	Logical Device Number[3]
20H	R	BFH	Manufacturer ID (read only)
21H	R	F0H	Device ID (read only)
22H	R	Revision Number	Device Revision (read only)
25H	W	00H	Chip Control register 0 Bits [7:1] are reserved Bit 0 = 1: Soft reset of Configuration registers[4]
26H	R/W	00H	Chip Control register 1 Bits [7:1] are reserved Bit 0 = SMIEN_IRQ2 1: Frame 3 of the SERIRQ cycle (IRQ2) is used to report Mailbox SMI. 0: SMI# pin is used to report Mailbox SMI
LOGICAL DEVICE 0 CONFIGURATION REGISTERS (KEYBOARD CONTROLLER)			
30H	R/W	00H	01H = Device is active 00H = Device is inactive. The address of the device is not decoded. LPC I/O read and write cycles to the device are ignored. No SERIRQ data frames is associated with keyboard and mouse interrupts.
60H[5],61H	R/W	00H,60H	Keyboard Data port LPC I/O address = 0000:A[11:3]:000 Command/Status port address = Data port address + 4
70H	R/W	01H	Keyboard Interrupt selection (no effect if Device is inactive)
72H	R/W	0CH	Mouse Interrupt selection (not effect if Device is inactive)
LOGICAL DEVICE 1 CONFIGURATION REGISTERS (ACPI ECI0)			
30H	R/W	00H	01H = Device is active 00H = Device is inactive. The address of device is not decoded. LPC I/O read and write cycles to the device are ignored.
60H[5],61H	R/W	00H,62H	ACPI ECI0 Data port LPC I/O address = 0000:A[11:3]:0:A1:0 Command/Status port address = Data port address + 4
LOGICAL DEVICE 2 CONFIGURATION REGISTERS (Mailbox 32 Byte Data Block)			
30H	R/W	00H	01H = Device is active 00H = Device is inactive. The address of device is not decoded. LPC I/O read and write cycles to the device are ignored.
60H[5],61H	R/W	00H,00H	Mailbox Index port LPC I/O address = 0000:A[11:1]:0 Mailbox Data port address = Index port address + 1

TABLE 23-1: Configuration Registers Map[1] (Continued)

Index	Access Type	Reset Value[2]	Configuration Register Name
LOGICAL DEVICE 3 CONFIGURATION REGISTERS (ACPI ECI1)			
30H	R/W	00H	01H = Device is active. 00H = Device is inactive; the address of device is not decoded; LPC I/O read and write cycles to the device are ignored.
60H[5], 61H	R/W	00H, 68H	ACPI ECI1 Data port LPC I/O address 0000:A[11:3]:0:A1:0 Command/Status port address = Data port address + 4

T23-1.1320

1. Register at indexes not listed in the map are reserved and must not be written to by software.
2. All configuration registers are returned to their reset values specified above after the following reset events: Power-On Reset, External reset, Watchdog timer reset, Brown-Out reset, aLPC Soft reset, LPC Soft reset, and Configuration Soft reset (see also Section 5.2).
3. A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device. All accesses to device specific configuration registers with index above 30H (including the activate command) operate only on the selected logical device.
4. The hardware automatically clears this bit after soft reset is completed; there is no need for software to clear this bit. This soft reset only affects configuration registers.
5. Register at index 60H contains high byte of LPC I/O address - bits A[15:8], and register at index 61H contains low byte of LPC I/O address- bits A[7:0].

24.0 ELECTRICAL SPECIFICATION

24.1 Absolute Maximum Stress Ratings

Absolute Maximum Stress Ratings (Applied conditions greater than those listed under "Absolute Maximum Stress Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions or conditions greater than those defined in the operational sections of this datasheet is not implied. Exposure to absolute maximum stress rating conditions may affect device reliability.)

Temperature Under Bias-55°C to +125°C
Storage Temperature65°C to +150°C
Supply Voltage on V_{DD} and AV_{DD} Pins to Ground Potential-0.3V to 3.8V
D.C. Voltage on Any Pin with IPCI, IOPCI and IODPCI buffer type[1] to Ground Potential-0.5V to VDD+0.5V
Transient Voltage (<20 ns) on Any Pin with IOPCI and IODPCI buffer type to Ground Potential-2.0V to VDD+2.0V
Voltage on Any Pin with AIO4 buffer type to Ground Potential-0.3V to AVDD
Voltage on Any Other Pin to Ground Potential-0.3V to 5.5V
Package Power Dissipation Capability (Ta=25°C)1.0W
Surface Mount Solder Reflow Temperature 260°C for 10 seconds
Output Short Circuit Current[2]50 mA

1. Refer to Table 2-1 for pin buffer type assignments.
2. Outputs shorted for no more than one second. No more than one output shorted at a time.

24.2 Operating Conditions

TABLE 24-1: Operating Range

Range	Ambient Temp	VDD	AVDD	FOSC	FECLK
Commercial	0°C to + 70°C	3.0-3.6V	3.15-3.45V[1]	32.768KHz	4-16 MHz[2] 8-33 MHz[3]

T24-1.1245

1. If accuracy of analog operations is not relevant for the particular application, AVDD range can be expanded to 3.0 - 3.6V, which would allow direct connection to VDD for the entire operating range.
2. If external clock is used as PLL input clock.
3. If external clock is used directly as 8051 core clock.

TABLE 24-2: AC Condition of Test

Input Rise/Fall Time	3 ns
Output Load	CL = 30 pF
See Figure 24-1 and Figure 24-2	

T24-2.1245

TABLE 24-3: Recommended System Power-up Timing

Symbol	Parameter	Minimum	Units
$T_{PU-READ}$[1]	Power-up (VDD = VDD Min) to LPC Host Read Operation	10	ms
$T_{PU-WRITE}$[1]	Power-up (VDD = VDD Min) to LPC Host Write Operation	10	ms

T24-3.1245

1. This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

TABLE 24-4: Pin Capacitance (T_A = 25°C; fc = 1MHz; V_{DD} = AVDD = 3.3V; other pins open)

Parameter	Description	Test Condition	Maximum
C_{INc}[1]	Clock Input Capacitance	V_{INC} = 0V	12 pF
C_{IN}[1]	Input Capacitance	V_{IN} = 0V	12 pF
$C_{I/O}$[1]	I/O Pin Capacitance	$V_{I/O}$ = 0V	12 pF
C_{OUT}[1]	Output Pin Capacitance	V_{OUT} = 0V	12 pF

T24-4.1245

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

TABLE 24-5: Reliability Characteristics

Symbol	Parameter	Minimum Specification	Units	Test Method
N_{END}[1]	Endurance	10,000	Cycles	JEDEC Standard A117
T_{DR}[1]	Data Retention	100	Years	JEDEC Standard A103
I_{LTH}[1]	Latch Up	100 + I_{DD}	mA	JEDEC Standard 78

T24-5.1245

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

24.3 DC Electrical Characteristics

TABLE 24-6: DC Characteristics (TA=0 to 70°C, AVDD = VDD = 3.0 to 3.6V, AVSS = VSS = 0V) (1 of 3)

Symbol	Type[2]	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IL}	AIO4	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to $V_{DD,}$ V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=4mA
V_{OH}		Output High Voltage	2.4	-	-	V	I_{OH}=-4mA
V_{IL}	I	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to $V_{DD,}$ V_{DD}=V_{DD} Max
V_{IL}	I_PD	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	10	66	110	µA	V_{IN} = GND to $V_{DD,}$ V_{DD}=V_{DD} Max
V_{IL}	IO5	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to $V_{DD,}$ V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=5mA
V_{OH}		Output High Voltage	2.4	-	-	V	I_{OH}=-5mA

TABLE 24-6: DC Characteristics (TA=0 to 70°C, AVDD = VDD = 3.0 to 3.6V, AVSS = VSS = 0V) (2 of 3)

Symbol	Type[2]	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IL}	IOD4	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=4mA
V_{IL}	IOD5	Input Low Voltage	-	-	0.8	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	2.0	-	-		AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4		I_{OL}=5mA
V_{IL}	IODPCI	Input Low Voltage	-0.5		0.3 V_{DD}	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	0.5 V_{DD}		V_{DD}+0.5	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.1V_{DD}		Iout=1.5mA
V_{IL}	IOPCI	Input Low Voltage	-0.5		0.3 V_{DD}	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	0.5 V_{DD}		V_{DD}+0.5	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.1V_{DD}		Iout=1.5mA
V_{OH}		Output High Voltage	0.9V_{DD}	-	-	V	Iout=-0.5mA
V_{IL}	IPCI	Input Low Voltage	-0.5		0.3 V_{DD}	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage	0.5 V_{DD}		V_{DD}+0.5	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}	OD4	Output Low Voltage	-	-	0.4	V	I_{OL}=4mA
V_T	SIO4_PU	Switching threshold		1.4		V	
V_{T-}		Schmitt trigger, negative-going threshold	0.8			V	AV_{DD}=V_{DD}=V_{DD} Min
V_{T+}		Schmitt trigger, positive-going threshold			2.0	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current with pull-up disabled with pull-up enabled	 -10 -110	 -66	 10 -10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=4mA
V_{OH}		Output High Voltage	2.4	-	-	V	I_{OH}=-4mA
V_T	SIO5	Switching threshold		1.4		V	
V_{T-}		Schmitt trigger, negative-going threshold	0.8			V	AV_{DD}=V_{DD}=V_{DD} Min
V_{T+}		Schmitt trigger, positive-going threshold			2.0	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=5mA
V_{OH}		Output High Voltage	2.4	-	-	V	I_{OH}=-5mA

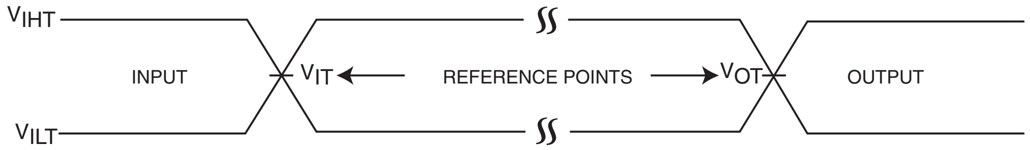
TABLE 24-6: DC Characteristics (TA=0 to 70ºC, AVDD = VDD = 3.0 to 3.6V, AVSS = VSS = 0V) (3 of 3)

Symbol	Type[2]	Parameter	Min	Typ	Max	Unit	Test Conditions
V_T	SIOD15	Switching threshold		1.4		V	
V_{T-}		Schmitt trigger, negative-going threshold	0.8			V	AV_{DD}=V_{DD}=V_{DD} Min
V_{T+}		Schmitt trigger, positive-going threshold			2.0	V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-10		10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{OL}		Output Low Voltage	-	-	0.4	V	I_{OL}=15mA
V_T	SI_PU	Switching threshold		1.4		V	
V_{T+}		Schmitt trigger, positive-going threshold			2.0	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{T-}		Schmitt trigger, negative-going threshold	0.8			V	AV_{DD}=V_{DD}=V_{DD} Max
I_{IL}		Input Leakage Current	-110	-66	-10	µA	V_{IN} = GND to V_{DD}, V_{DD}=V_{DD} Max
V_{IL}	OSC[1]	Input Low Voltage (OSC1)	V_{SS}	-	0.2 x V_{DD}	V	AV_{DD}=V_{DD}=V_{DD} Min
V_{IH}		Input High Voltage(OSC1)	0.8 x V_{DD}	-	-	V	AV_{DD}=V_{DD}=V_{DD} Max
V_{BOD}		Brown-out Detection Voltage	2.05	2.5	2.85	V	
I_{DD}	PWR	V_{DD} Supply Current in					V_{DD} = V_{DD} Max LCLK=V_{ILT}/ V_{IHT} at
		Active Mode without Flash program/erase operation			35	mA	f=33MHz ECLK=V_{ILT}/ V_{IHT} at f=14.318 MHz
		Active Mode with Flash program/erase operation			45	mA	PLL running at f=32.2155 MHz
		Idle Mode			25	mA	f=32.768 KHz All outputs and inputs with pull-ups or pull-downs are open. All other inputs = $V_{DD.}$
		Power Down Mode			150	uA	V_{DD} = V_{DD} Max PLL stopped. Oscillator disabled. All outputs and inputs with pull-ups or pull-downs are open. All other inputs = V_{DD}.
I_{DDA}	PWR	AVDD Supply Current					AV_{DD} = AV_{DD} Max ADC clock frequency = 2MHz No external load on DAC outputs
		Active Mode ADC operation			4	mA	
		Active Mode DAC operation		18	25	mA	
		Standby Mode (ADC and DAC are disabled)		30	70	uA	

T24-6.1245

[1] Connect crystal oscillator circuitry to oscillator input and output pins according to the schematic on Figure 5-3.
[2] See I/O buffer types description in Table 2-2.

24.4 AC Electrical Characteristics



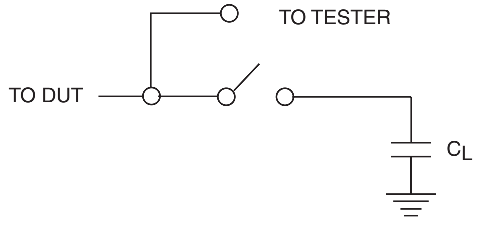
AC test inputs are driven at V_{IHT} (0.9 V_{DD}) for a logic “1” and V_{ILT} (0.1 V_{DD}) for a logic “0”. Measurement reference points for inputs and outputs are V_{IT} (0.5 V_{DD}) and V_{OT} (0.5 V_{DD}). Input rise and fall times (10%↔90%) are <3 ns.

Note: V_{IT} = V_{INPUT} Test
V_{0T} = V_{OUTPUT} Test
V_{IHT} = V_{INPUT} HIGH Test
V_{ILT} = V_{INPUT} LOW Test

1245 ACInOut Waveforms.0

FIGURE 24-1: AC Input/Output Reference Waveforms

Note: The above reference points apply to all AC measurements specified in this section unless explicitly stated otherwise. For AC condition of test and operating range see Table 24-1 and Table 24-2.



1245 TestLoad Ex.0

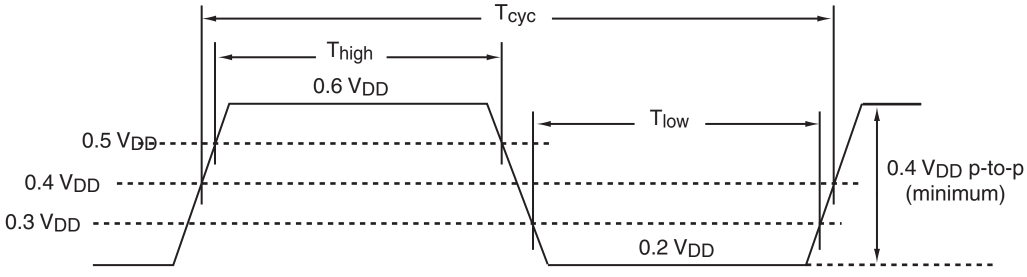
FIGURE 24-2: A Test Load Example

24.4.1 LPC Interface and Firmware Memory Timing

TABLE 24-7: LPC Clock Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{CYC}	LCLK Cycle Time	30		ns
T_{HIGH}	LCLK High Time	11		ns
T_{LOW}	LCLK Low Time	11		ns
-	LCLK Slew Rate (peak-to-peak)	1	4	V/ns
-	LRESET# Slew Rate	50		mV/ns

T24-7.1245



1245 LCLK WaveForm.0

FIGURE 24-3: LCLK Wave Form

TABLE 24-8: LPC Read/Write Cycle Timing Parameters

Symbol	Parameter	Min	Max	Units
T_{CYC}	Clock Cycle Time	30		ns
T_{SU}	Data Set Up Time to Clock Rising	7		ns
T_{DH}	Clock Rising to Data Hold Time	0		ns
T_{VAL}[1]	Clock Rising to Data Valid	2	11	ns
T_{BP}	Byte Programming Time		60	µs
T_{SE}	Sector-Erase Time		60	ms
T_{BE}	Block-Erase Time		60	ms
T_{ES}	Erase Suspend Latency Time		20	µs
T_{ON}	Clock Rising to Active (Float to Active Delay)	2		ns
T_{OFF}	Clock Rising to Inactive (Active to Float Delay)		28	ns

T24-8.1245

[1] Minimum and maximum time have different loads. See PCI spec.

TABLE 24-9: LPC AC Input/Output Specifications[1]

Symbol	Parameter	Min	Max	Units	Conditions
I_{OH}(AC)	Switching Current High	-12 V_{DD}		mA	0 < V_{OUT} ≤0.3 V_{DD}
		-17.1(V_{DD}-V_{OUT}		mA	0.3 V_{DD} < V_{OUT} < 0.9 V_{DD}
			Equation C[2]		0.7 V_{DD} < V_{OUT} < V_{DD}
	(Test Point)		-32 V_{DD}	mA	V_{OUT} = 0.7 V_{DD}
I_{OL}(AC)	Switching Current Low	16 V_{DD}		mA	V_{DD} >V_{OUT} ≥0.6 V_{DD}
		26.7 V_{OUT}		mA	0.6 V_{DD} > V_{OUT} > 0.1 V_{DD}
			Equation D[2]		0.18 V_{DD} > V_{OUT} > 0
	(Test Point)	26.7 V_{OUT}	38 V_{DD}	mA	V_{OUT} = 0.18 V_{DD}
I_{CL}	Low Clamp Current	-25+(V_{IN}+1)/0.015		mA	-3 < V_{IN} ≤-1
I_{CH}	High Clamp Current	25+(V_{IN}-V_{DD}-1)/0.015		mA	V_{DD}+4 > V_{IN} ≥V_{DD}+1
$SLEW_R$	Output Rise Slew Rate	1	4	V/ns	0.2 V_{DD}-0.6 V_{DD} load
$SLEW_F$[3]	Output Fall Slew Rate	1	4	V/ns	0.6 V_{DD}-0.2 V_{DD} load

T24-9.1245

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

[2] See PCI spec

[3] PCI specification output load is used

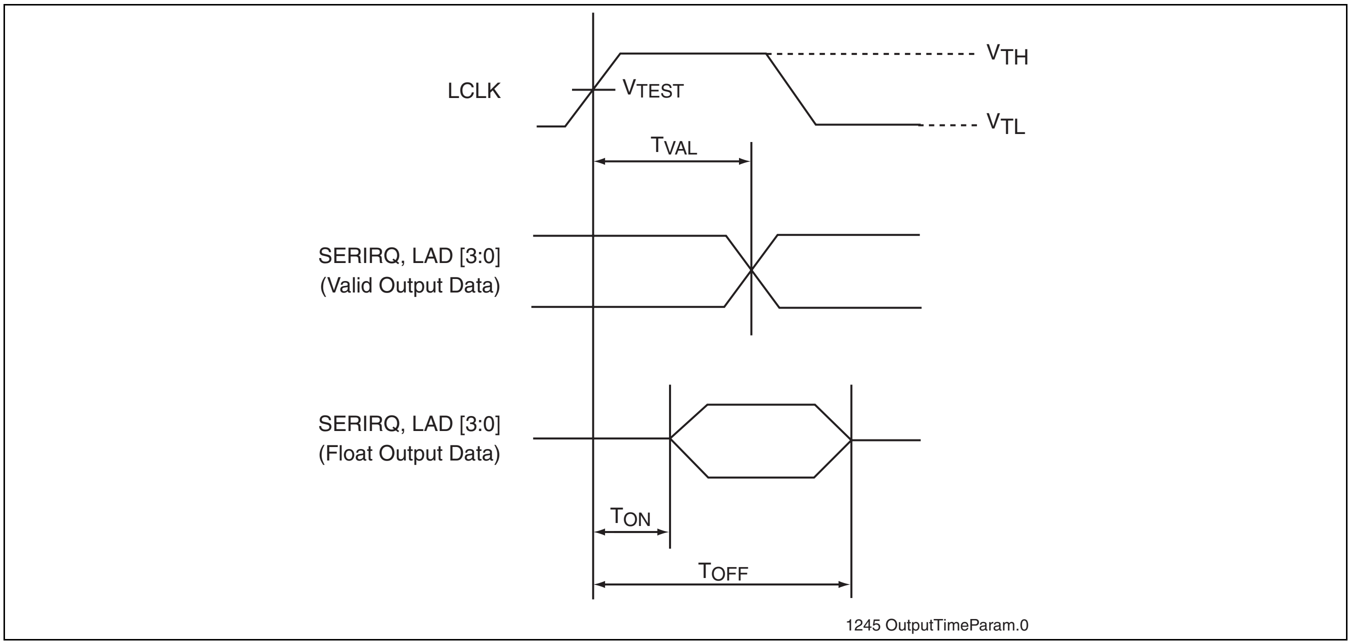


FIGURE 24-4: LPC Output Timing Parameters

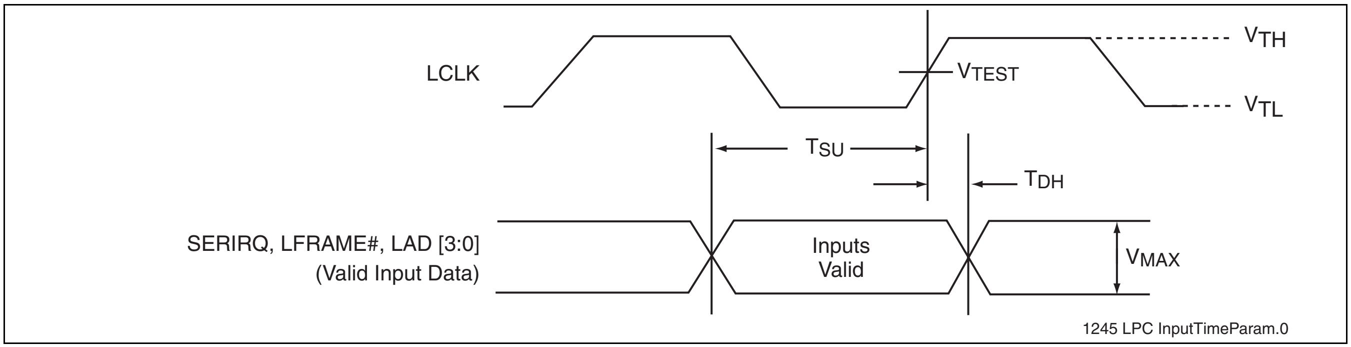


FIGURE 24-5: LPC Input Timing Parameters

TABLE 24-10: LPC Interface Measurement Condition Parameters

Symbol	Value	Units
V_{TH}[1]	0.6 V_{DD}	V
V_{TL}[1]	0.2 V_{DD}	V
V_{TEST}	0.4 V_{DD}	V
V_{MAX}[1]	0.4 V_{DD}	V
Input Signal Edge Rate	1	V/ns

T24-10.1245

1. The input test environment is done with 0.1 VDD of overdrive over VIH and VIL. Timing parameters must be met with no more overdrive than this. VMAX specified the maximum peak-to-peak waveform allowed for measuring input timing. Production testing may use different voltage values, but must correlate results back to these parameters.

TABLE 24-11: LPC Reset Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T_{LRSTP}	LRESET# Pulse Width	100		ns
T_{LRSTF}	LRESET# Low to Output Float		48	ns
T_{LRSTD}[2]	LRESET# High to LFRAME# or SERIRQ Low (1st LPC Host access delay after LPC Reset)	150		ns
T_{LRSTE}[3]	LRESET# Low to LFRAME# Low if reset during Sector-/Block-Erase or Program	60		µs

T24-11.1245

1. Guaranteed by design
2. LPC Reset NOT during Program or Erase operation
3. LPC Reset during Program or Erase operation

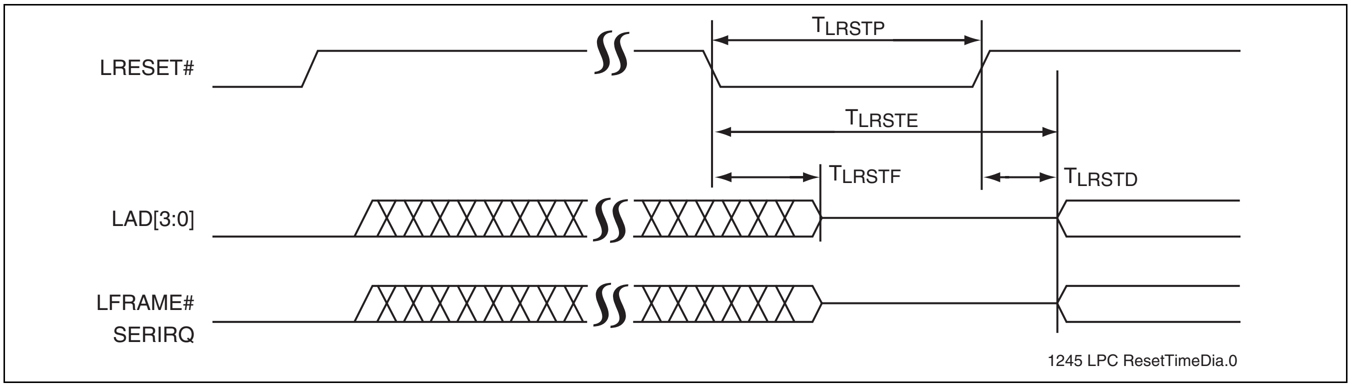


FIGURE 24-6: LPC Reset Timing Diagram

24.4.2 External Clocks and Reset Timing

TABLE 24-12: External Clocks and Reset Timing Parameters[1]

SYMBOL	PARAMETER	Min	Typ	Max	Units
$F_{OSC} = 1/T_{OSC}$	Crystal Oscillator Frequency		32.768		KHz
T_{OSCSU}	Crystal Oscillator Start Up time		1	5	s
$1/T_{ECLK} = F_{ECLK} = F_{CCLK}$	External Clock Frequency if used as 8051 clock	8		33	MHz
$1/T_{ECLK} = F_{ECLK} = F_{PLLI}$	External Clock Frequency if used as PLL input	4		16	MHz
T_{ECLKH}/T_{ECLK}	External Clock Duty Cycle	40		60	%
T_{ECLKF}	External Clock Fall Time			5	ns
T_{ECLKR}	External Clock Rise Time			5	ns
F_{RCLK}	Ring Oscillator Frequency	8		24	MHz
T_{PLLON}	Time to switch 8051 core clock to PLL output			300 + 16128/ F_{PLLO}[2]	µs
T_{CCLK}	8051 Core Clock Period requirements	30		125	ns
T_{WLOW} (T_{WHIGH})	External interrupt input pulse low (high) time	2			T_{CCLK}
T_{WLOW} (T_{WHIGH})	External timer input pulse low (high) time	12			T_{CCLK}
T_{XRSTP}	External Reset# pulse width	48			T_{CCLK}
T_{XRSTD}	External Reset# to 1st LPC Host access delay External Reset# High to 8051 code start delay	10		 10	ms ms
T_{PURSTD}	Power-up to 1st LPC Host access delay Power-up to 8051 code start delay	10		 10	ms ms

T24-12.1245

[1] Guaranteed by design
[2] F_{PLLO} – PLL output clock frequency in MHz (see Section 5.3.2)

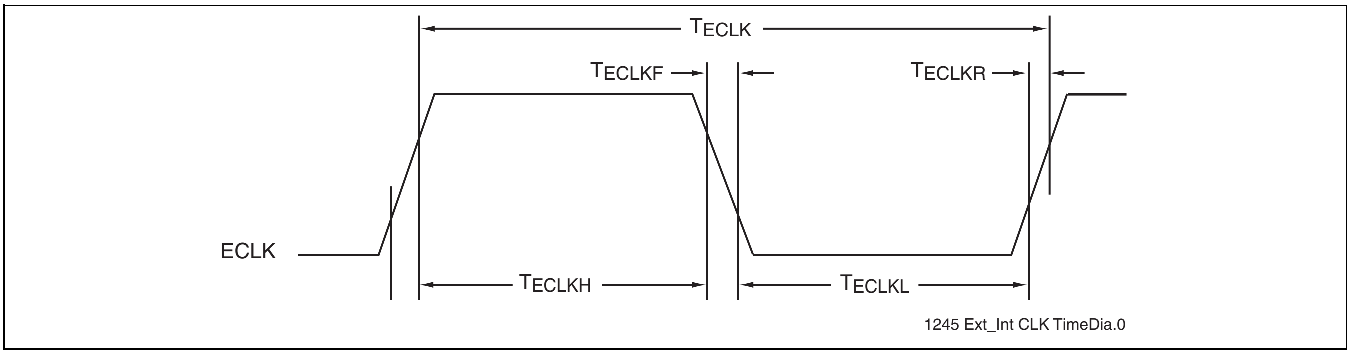


FIGURE 24-7: External Input Clock Timing Diagram

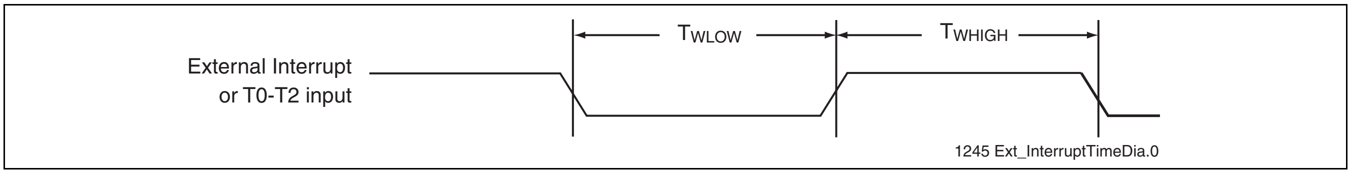
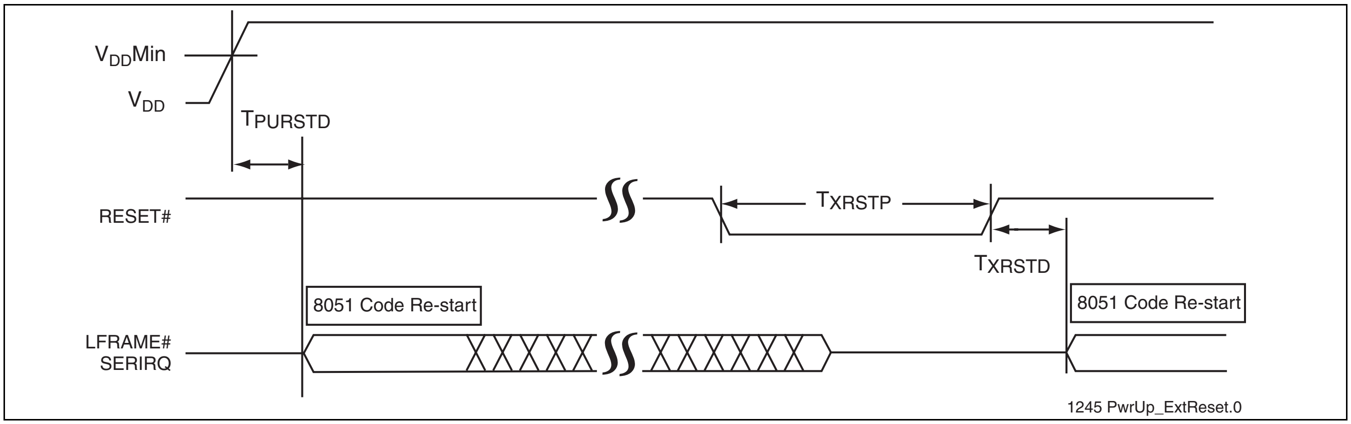


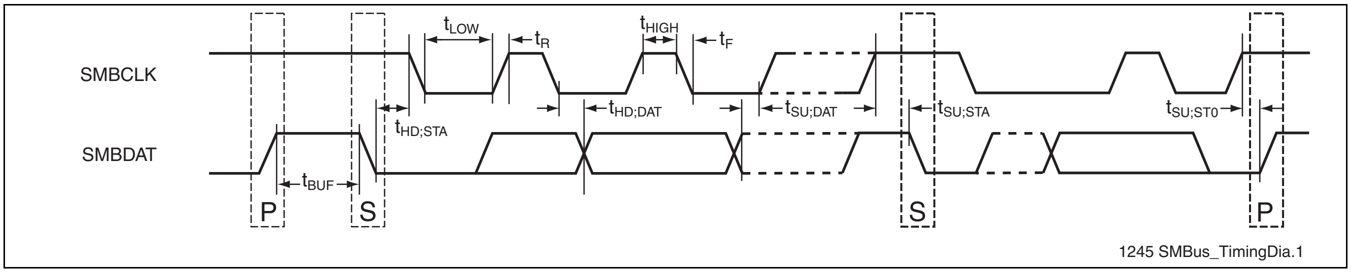
FIGURE 24-8: External Interrupt Timing Diagram



1245 PwrUp_ExtReset.0

FIGURE 24-9: Power Up and External Reset Timing Diagram

24.4.3 SMBus Interface Timing



1245 SMBus_TimingDia.1

FIGURE 24-10: SMBus Timing Diagram

TABLE 24-13: SMBus Interface Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
F_{SMB}	SMBus Operating Frequency		100	kHz
T_{BUF}	Bus Free Time Between Stop and Start Condition	4.7		µs
$T_{HD:STA}$	Hold time after (repeated) Start Condition. After this period, the first clock is generated	4.0		µs
$T_{SU:STA}$	Repeated Start Condition setup time	4.7		µs
$T_{SU:STO}$	Stop Condition Setup Time	4.0		µs
$T_{HD:DAT}$	Data Hold Time	90		ns
$T_{SU:DAT}$	Data Setup Time	250		ns
T_{LOW}	Clock Low Period	4.7		µs
T_{HIGH}	Clock High Period	4.0		µs
T_F	Clock/Data Fall Time		300	ns
T_R[2]	Clock/Data Rise Time		1000	ns

T24-13.1245

1. Guaranteed by design
2. Depends on pull-up value

TABLE 24-14: SMBus Interface Measurement Reference Points

Symbol	Value	Units
V_{TH}	V_{T+} Max + 0.25V	V
V_{TL}	V_{T-} Min - 0.15V	V

T24-14.1245

24.4.4 PS/2 Interface Timing

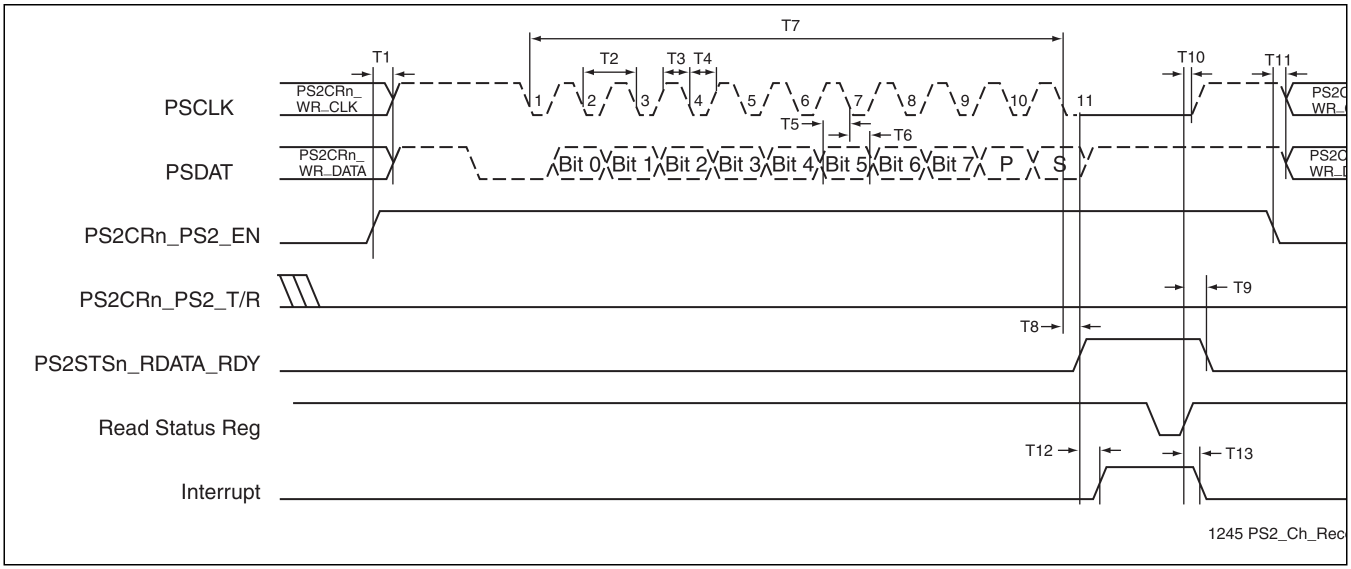


FIGURE 24-11: PS/2 Hardware State Machine Receive Timing Diagram

Note: Solid (dashed) line indicates that PS/2 interface signal is driven by SST79LF008 (PS/2 peripheral device).

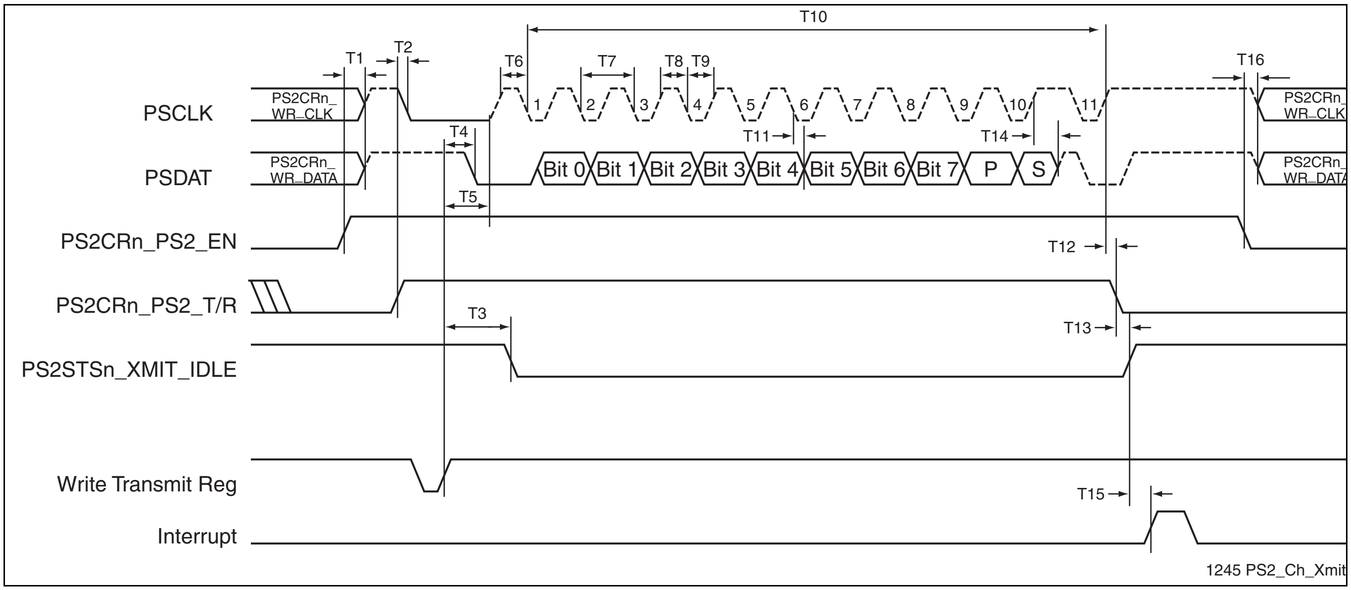


FIGURE 24-12: PS/2 Hardware State Machine Transmit Timing Diagram

Note: Solid (dashed) line indicates that PS/2 interface signal is driven by SST79LF008 (PS/2 peripheral device).

TABLE 24-15: PS/2 Receive Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T1	Time from PS/2 state machine enabled in receive mode (PS2CRn_PS2_EN = 1 and PS2CRn_PS2_T/R = 0) to SST79LF008 PSCLK and PSDAT outputs are in High-Z state	6	15	ns
T2	PSCLK period		300[2]	µs
T3	Duration of PSCLK active (high)	30		µs
T4	Duration of PSCLK inactive (low)	30		µs
T5	Setup time from input PSDAT transition to falling edge of PSCLK (SST79LF008 uses falling edge of PSCLK to sample PSDAT)	0		ns
T6	Hold time from falling edge of PSCLK to input PSDAT transition (SST79LF008 uses falling edge of PSCLK to sample PSDAT)	600		ns
T7	Time from falling edge of the 1st clock (Start bit) to falling edge of the 11th clock (Stop bit)		2[2]	ms
T8	Time from falling edge of the 11th clock (Stop bit) to SST79LF008 sets PS2STSn_RDATA_RDY bit and drives PSCLK low to inhibit the next transfer		600	ns
T9	Time from SST79LF008 Status Register read (trailing edge of the read signal) to PS2STSn_RDATA_RDY bit cleared		0	ns
T10	Time from SST79LF008 Status Register read (trailing edge of the read signal) to SST79LF008 PSCLK output is in High-Z state	30	150	ns
T11	Time from PS/2 state machine disabled (PS2CRn_PS2_EN = 0) to SST79LF008 PSCLK and PSDAT outputs are configured according to the PS2CRn_WR_CLK and PS2CRn_WR_DATA bits	6	15	ns
T12	Time from PS2STSn_RDATA_RDY bit low-to-high transition to PS/2 Channel interrupt generated		150	ns
T13	Time from SST79LF008 Status Register read (trailing edge of the read signal) to PS/2 interrupt cleared		0	ns

T24-15.1320

[1] Guaranteed by design
[2] These maximum limits applied by SST79LF008 hardware provided the respective time-out detection is enabled

TABLE 24-16: PS/2 Transmit Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T1	Time from PS/2 state machine enabled in receive mode (PS2CRn_PS2_EN = 1 and PS2CRn_PS2_T/R = 0) to SST79LF008 PSCLK and PSDAT outputs are in High-Z state	6	15	ns
T2	Time from PS/2 state machine switched into transmit mode (PS2CRn_PS2_T/R = 1) to PSCLK line driven low	60	300	ns
T3	Time from SST79LF008 transmit register write (trailing edge of the write signal) to PS2STSn_XMIT_IDLE bit cleared		T5 + 0	ns
T4	Time from SST79LF008 transmit register write (trailing edge of the write signal) to PSDAT line driven low	30	150	ns
T5	Time from SST79LF008 transmit register write (trailing edge of the write signal) to SST79LF008 PSCLK output is in High-Z state	T4+ 90	T4+ 450	ns
T6	Time from request-to-send state (PSCLK = 1, PSDAT = 0) to the 1st clock falling edge (Start bit) generated by the peripheral device	2	 25[2]	µs ms
T7	PSCLK period		300[2]	µs
T8	Duration of PSCLK active (high)	30		µs
T9	Duration of PSCLK inactive (low)	30		µs
T10	Time from falling edge of the1st clock (Start bit) to rising edge of the 11th clock (Line Control bit)		2[2]	ms
T11	Time from falling edge of PSCLK to SST79LF008 PSDATA output is in High-Z state to transmit ‘1’, or driven low to transmit ‘0’ (the peripheral device uses rising edge of PSCLK to sample PSDATA)	60	450	ns
T12	Time from rising edge of the 11th clock (Line Control bit) to PS_T/R bit cleared	90	450	ns
T13	Time from PS_T/R bit cleared to PS2STSn_XMIT_IDLE bit set	30	150	ns
T14	Time from rising edge of the 10th clock (Stop bit) to SST79LF008 PSDATA output is in High-Z state	30	600	ns
T15	Time from PS2STSn_XMIT_IDLE bit low-to-high transition to PS/2 Channel interrupt generated Interrupt is cleared by reading the Status Register same as in receive mode - not shown.	30	150	ns
T16	Time from PS/2 state machine disabled (PS2CRn_PS2_EN = 0) to SST79LF008 PSCLK and PSDAT outputs are configured according to the PS2CRn_WR_CLK and PS2CRn_WR_DATA bits	6	15	ns

T24-16.1245

[1] Guaranteed by design
[2] These maximum limits applied by SST79LF008 hardware provided the respective time-out detection is enabled

TABLE 24-17: PS/2 Interrupt Timing in bit-banging mode[1,2]

Symbol	Parameter	Min	Max	Units
T1	Time from falling edge of PSCLK to PS/2 Channel Interrupt generated	60	450	ns
T2	Time from SST79LF008 Status Register read (trailing edge of the read signal) to PS/2 interrupt cleared		0	ns

T24-17.1245

[1] In bit-banging mode PS/2 receive and transmit protocols are controlled by the 8051 firmware
[2] Guaranteed by design

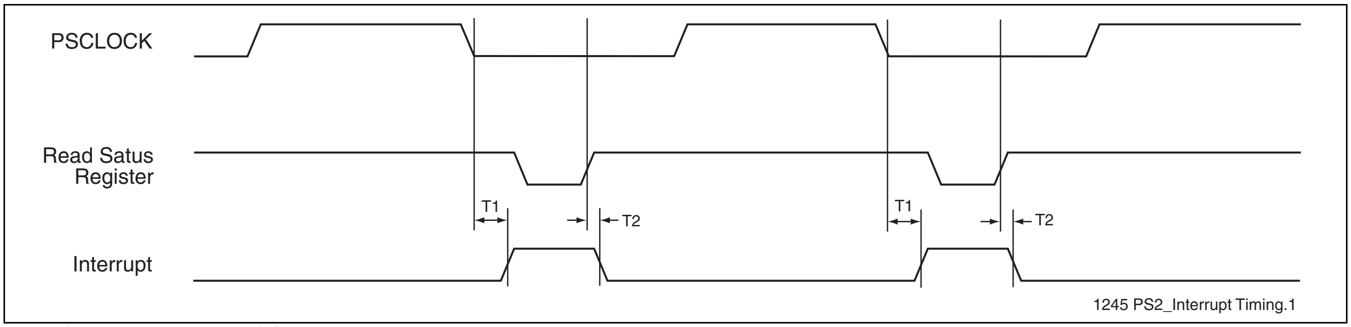


FIGURE 24-13: PS/2 Interrupt Timing in bit-banging mode

TABLE 24-18: PS/2 Interface Measurement Reference Points

Symbol	Value	Units
V_{TH}	V_{T+} Max	V
V_{TL}	V_{T-} Min	V

T24-18.1245

24.4.5 UART Timing

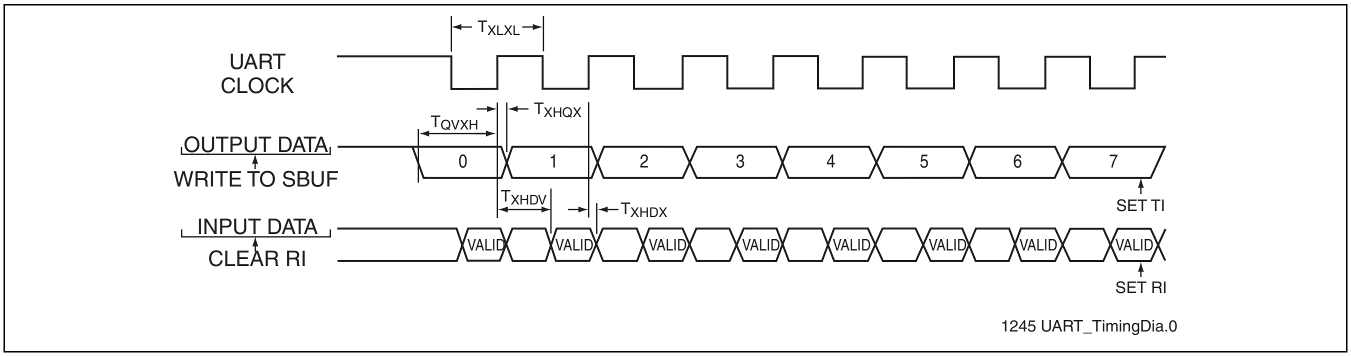


FIGURE 24-14: UART Timing Diagram (Shift Register Mode)

TABLE 24-19: UART Timing Parameters[1]

Symbol	Parameter	8051 core clock frequency						Units
		12MHz		32MHz		Variable		
		Min	Max	Min	Max	Min	Max	
T_{XLXL}	Serial Port Clock Cycle Time	1.0		0.375		12 T_{CCLK}[2]		us
T_{QVXH}	Output Data Setup to Clock Rising Edge	700		179		10 T_{CCLK}[2]– 133		ns
T_{XHQX}	Output Data Hold After Clock Rising Edge	50		0		2 T_{CCLK}[2]– 117		ns
T_{XHDXr}	Input data Hold After Clock Rising Edge	0		0		0		ns
T_{XHDV}	Clock Rising Edge to input Data Valid		700		179		10 T_{CCLK}[2] - 133	ns

T24-19.1245

[1] Guaranteed by design
[2] T_{CCLK} – 8051core clock period (see Table 24-12)

24.4.6 SPI Timing

TABLE 24-20: SPI Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T_{SU}	Data In Setup time	1		T_{CCLK}[2]
T_{DH}	Data In Hold time	1		T_{CCLK}[2]
T_V	Data Out Valid time		1	T_{CCLK}[2]
T_{SSS}	SS setup time.	1		T_{CCLK}[2]
T_{SSH}	SS hold time	1		T_{CCLK}[2]
T_{SCK}	Serial Clock cycle in master (slave) mode	2 (4)		T_{CCLK}[2]
T_{SCKH}	Serial Clock low time	1		T_{CCLK}[2]
T_{SCK}	Serial Clock high time	1		T_{CCLK}[2]

T24-20.1245

[1] Guaranteed by design
[2] TCCLK – 8051core clock period (see Table 24-12)

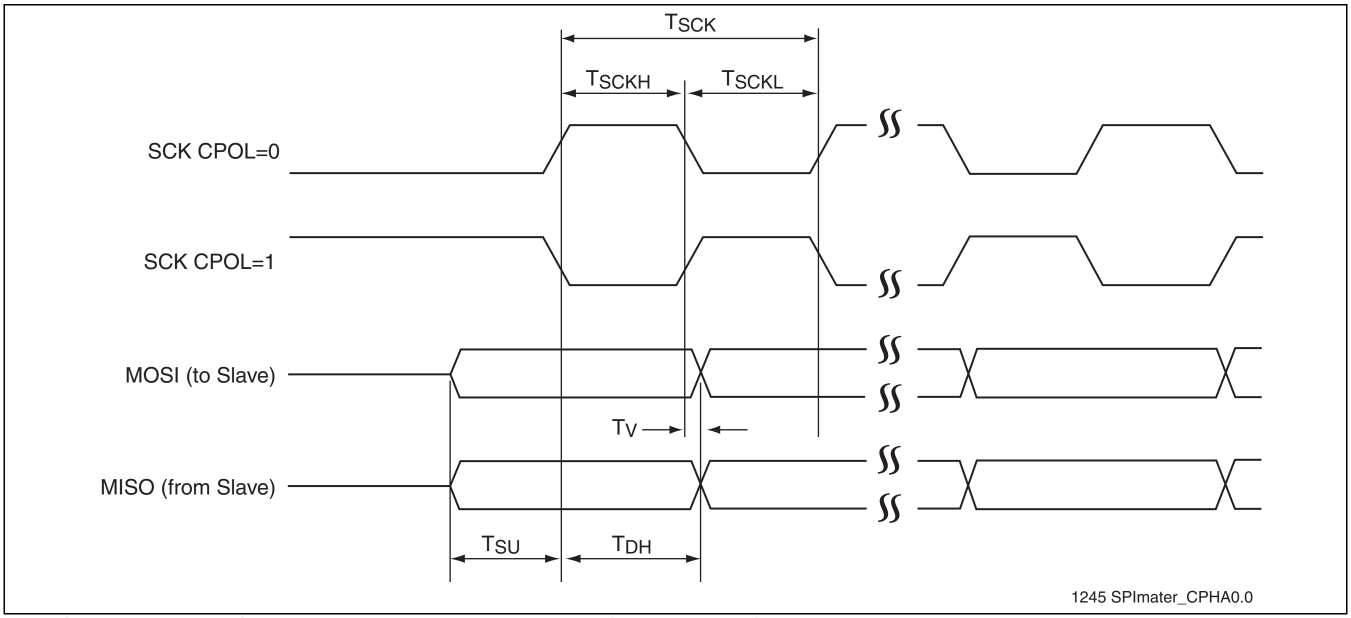


FIGURE 24-15: SPI Master Timing Diagram (CPHA=0, MSTR = 1)

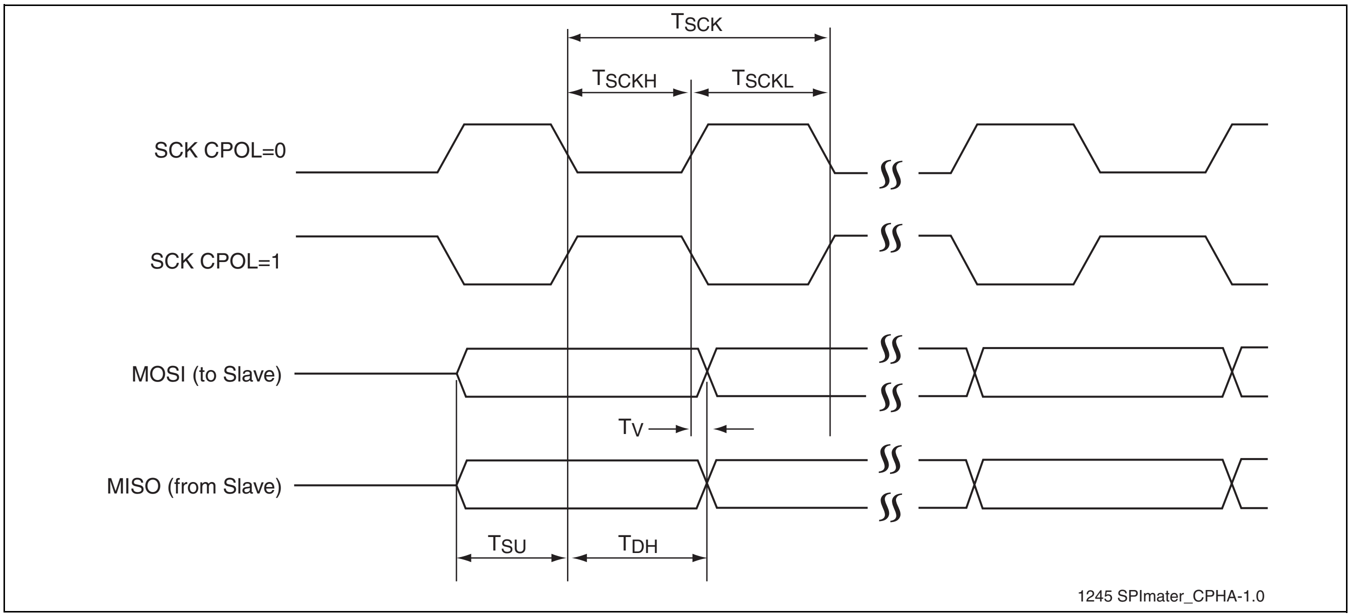


FIGURE 24-16: SPI Master Timing Diagram (CPHA=1, MSTR = 1)

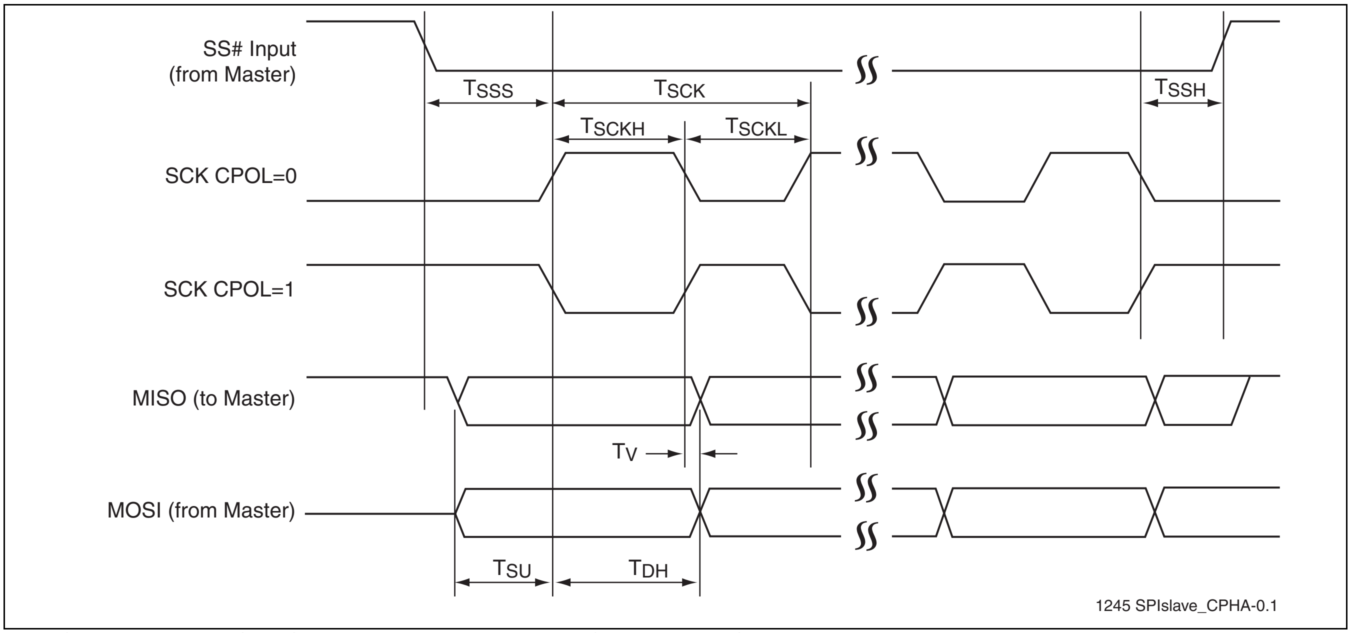


FIGURE 24-17: SPI Slave Timing Diagram (CPHA=0, MSTR = 0)

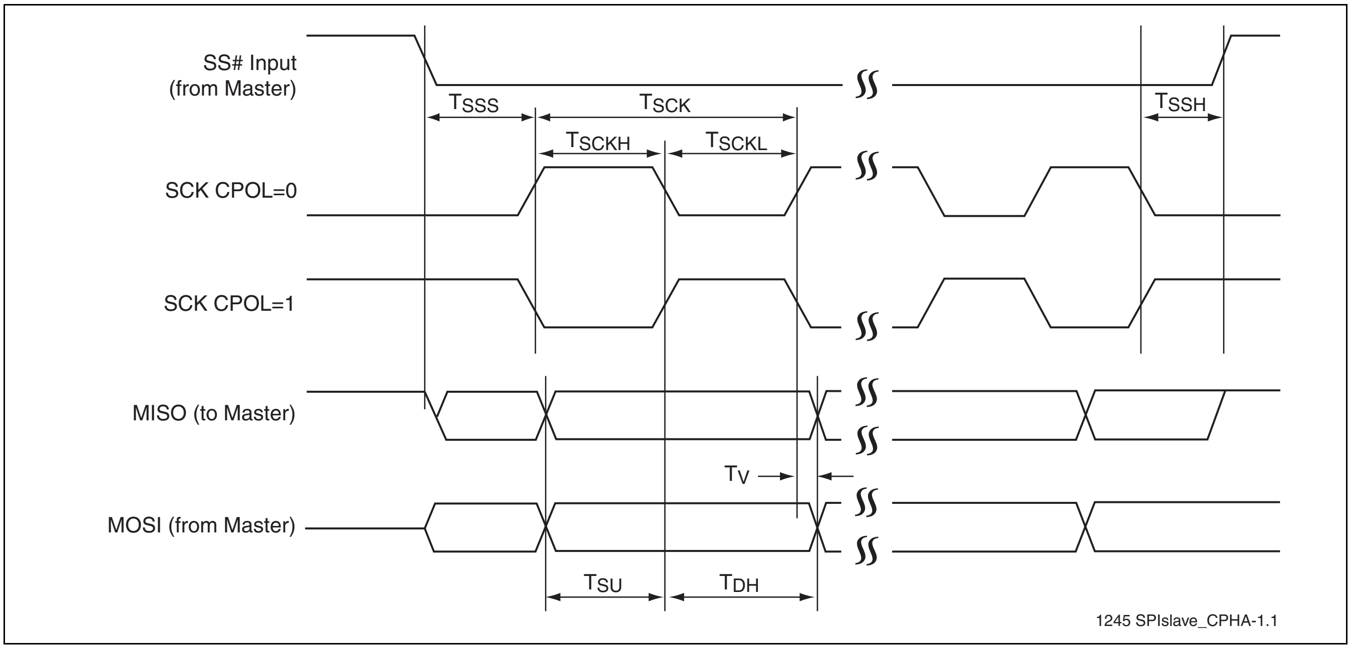


FIGURE 24-18: SPI Slave Timing Diagram (CPHA=1, MSTR = 0)

24.4.7 PWM and FAN Tachometer Timing

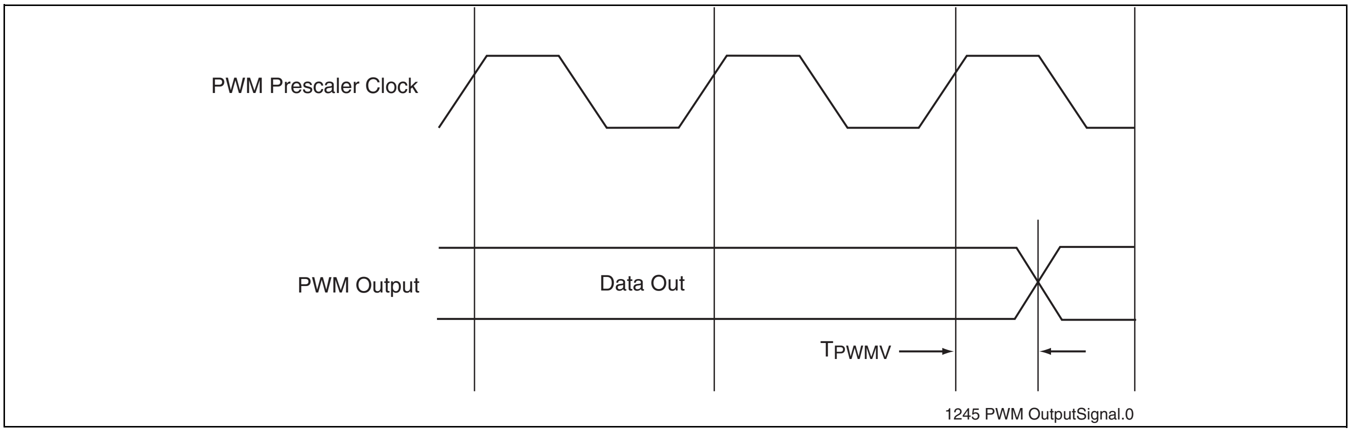


FIGURE 24-19: PWM Output Signals Timing Diagram

TABLE 24-21: PWM Output Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T_{PWMV}	PWM Output Valid Time	0	0.5	T_{CCLK}[2]

T24-21.1245

1. Guaranteed by design
2. T_{CCLK} – 8051core clock period (see Table 24-12)

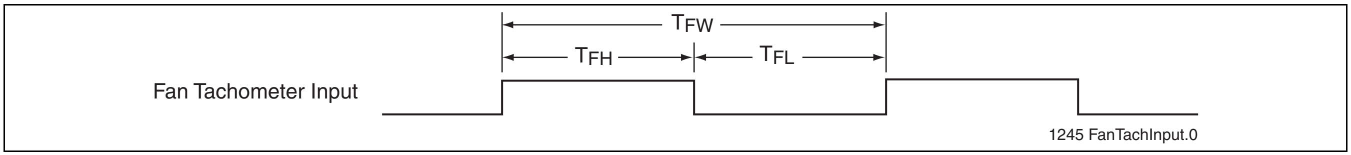


FIGURE 24-20: FAN Tachometer Input Timing Diagram

TABLE 24-22: FAN Tachometer Input Timing Parameters[1]

Symbol	Parameter	Min	Max	Units
T_{FW}	Fan Tachometer Input Pulse Width	4		T_{FTCLK}[2]
T_{FH}	Fan Tachometer Input Pulse High Time	3		T_{FTCLK}[2]
T_{FL}	Fan Tachometer Input Pulse Low Time	1		T_{FTCLK}[2]

T24-22.1245

1. Guaranteed by design
2. T_{FTCLK} is a period of the clock used for the tachometer counter (see Section 5.2)

24.4.8 aLPC Interface Timing

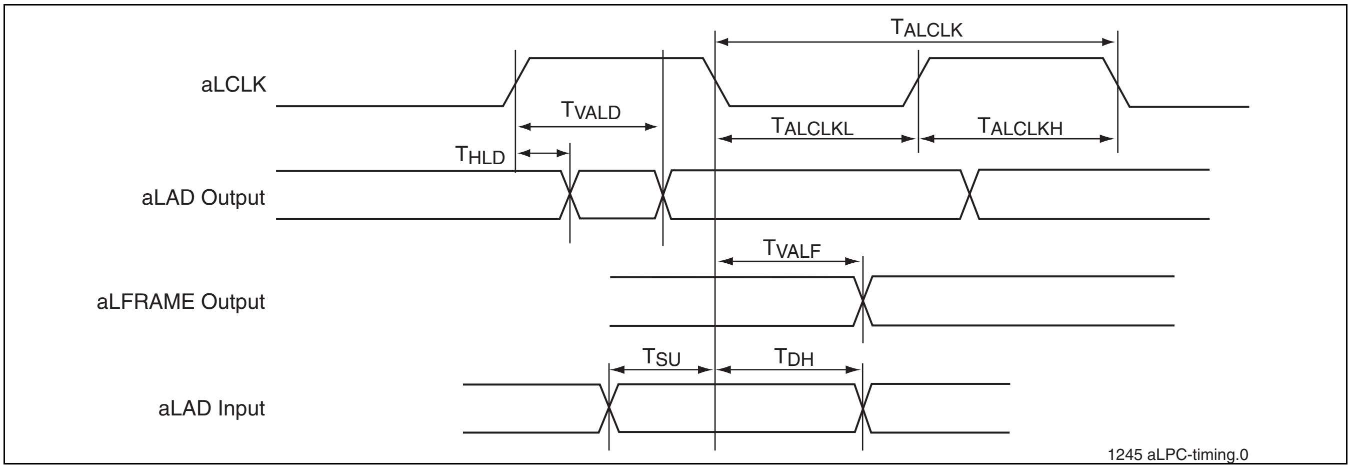


FIGURE 24-21: aLPC Timing Diagram

TABLE 24-23: aLPC Timing Parameters[1]

Symbol	Parameter	MIN	MAX	UNITS
T_{ALCLK}	aLCLK Clock Cycle Time before entry to the aLPC mode[2]	1250		ns
T_{ALCLK}	aLCLK Clock Cycle Time in the aLPC mode[3]	200		ns
T_{ALCLKH}	aLCLK Clock High Time	80		ns
T_{ACLKL}	aLCLK Clock Low Time	80		ns
T_{HLD}	Clock Rising to Output Data Hold	0		ns
T_{VALD}	Clock Rising to Output Data Valid	2	20	ns
T_{VALF}	Clock Falling to Output aLFARME Valid	5	20	ns
T_{SU}	Input Data Set Up Time to Clock Falling	10		ns
T_{DH}	Clock Falling to Input Data Hold Time	5		ns
T_{ARSTD}	aLPC mode entry to 1st aLPC Host access delay aLPC mode exit to 1st LPC Host access delay	15000 + 8*T_{ALCLK}		ns

T24-23.1245

1. Guaranteed by design
2. During Enable_and_Poll and Switch_and_Reset sequences followed by reset completion delay T_{ARSTD}
3. After T_{ARSTD} delay

24.5 Analog Characteristics

24.5.1 ADC Characteristics

TABLE 24-24: ADC Characteristics (TA=0 to 70°C, VDD = 3.0-3.6V, AVDD = 3.15-3.45V, AVSS = VSS = 0V)

Symbol	Parameter	Min	Typ	Max	Units	Conditions
Bit	Resolution		10		Bits	
AINT[1]	Analog Input Voltage	0		AV_{DD}	V	
DNL	Differential Non-Linearity Error		±0.8	±1	LSB	
INL	Integral Non-Linearity Error		±1.0	±2	LSB	
TOPOFF BOTOFF[1]	Offset Voltage	-8	3	8	LSB	
F_C	Maximum Conversion Rate			400	KSPS	ADC clock frequency = 2.0 MHz

T24-24.1245

[1] Guaranteed by design

24.5.2 DAC Characteristics

TABLE 24-25: DAC Characteristics (TA=0 to 70°C, VDD = 3.0-3.6V, AVDD = 3.15-3.45V, AVSS = VSS = 0V, RL >= 100k, CL <= 50pF)

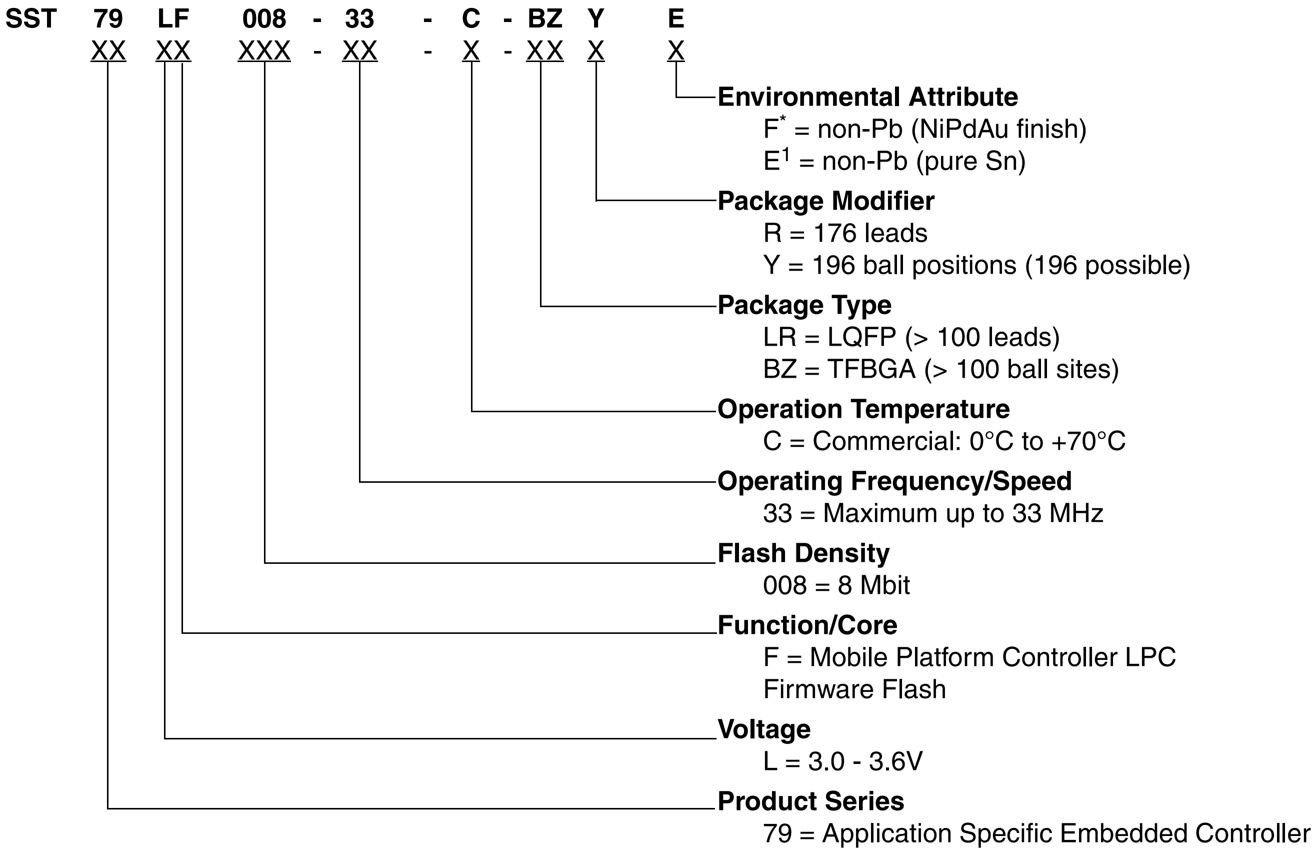
Symbol	Parameter	Min	Typ	Max	Units	Conditions	Comments
Bit	Resolution		8		Bits		
DNL	Differential Non-Linearity Error		0.3	1.0	LSB		
INL	Integral Non-Linearity Error		0.5	1.0	LSB		
V_{FS}	Full Scale Voltage	3.087	3.207	3.287	V	AV_{DD} = 3.3V	$V_{FS} = V_{OMAX}$ – VOUT(00H)[1]
V_{ZSE}	Zero Scale Error		40	100	mV		V_{ZSE} = VOUT(00H)[1]
V_{FSE}	Full Scale Voltage Error		40	100	mV		V_{FSE} = V_{OMAX} – (AV_{DD}*255/256)
V_{OMAX}[2]	Maximum Output Voltage	3.187	3.247	3.287	V	AV_{DD} = 3.3V	V_{OMAX} = VOUT(FFH)
V_{LSB}[2]	LSB Size	12.11	12.58	12.89	mV	AV_{DD} = 3.3V	V_{LSB} = (V_{OMAX} – VOUT(00H)[1])/255
	Channel Crosstalk[2,3]		-40	-30	dB		20*log(V_{PP} max of unselected channel / V_{FS} of selected channel)
T_S[2,4]	Analog Output Settling Time			1	us	C_L = 50pF R_L = 100k	
T_{ON}[2,5] T_{ONA}[2,6]	Analog Output Enable Time			3 100	us	C_L = 50pF R_L = 100k	

T24-25.1245

[1] VOUT(XXH) – actual DAC output voltage for input code XXH
[2] Guaranteed by design
[3] Peak-to-peak output voltage of unselected channel with input code 80H, when selected channel output voltage changes from VOUT(00H) to VOUT(FFH)
[4] Time from loading data to output voltage settling within an error of +/- 0.5LSB
[5] Time from the moment when DACENn = 1 in DACCTRL register to settling of the output voltage
[6] Time from the moment when DACEN = 1 in DACCTRL register to settling of the output voltage

Advance Information

25.0 PRODUCT ORDERING INFORMATION



SST 79 LF 008 - 33 - C - BZ Y E

XX XX XXX - XX - X - XX X X

Environmental Attribute
F* = non-Pb (NiPdAu finish)
E1 = non-Pb (pure Sn)

Package Modifier
R = 176 leads
Y = 196 ball positions (196 possible)

Package Type
LR = LQFP (> 100 leads)
BZ = TFBGA (> 100 ball sites)

Operation Temperature
C = Commercial: 0°C to +70°C

Operating Frequency/Speed
33 = Maximum up to 33 MHz

Flash Density
008 = 8 Mbit

Function/Core
F = Mobile Platform Controller LPC Firmware Flash

Voltage
L = 3.0 - 3.6V

Product Series
79 = Application Specific Embedded Controller

* Environmental suffix “F” or “E” denotes non-Pb solder. SST non-Pb solder devices are “RoHS Compliant”.

25.1 Valid Combinations

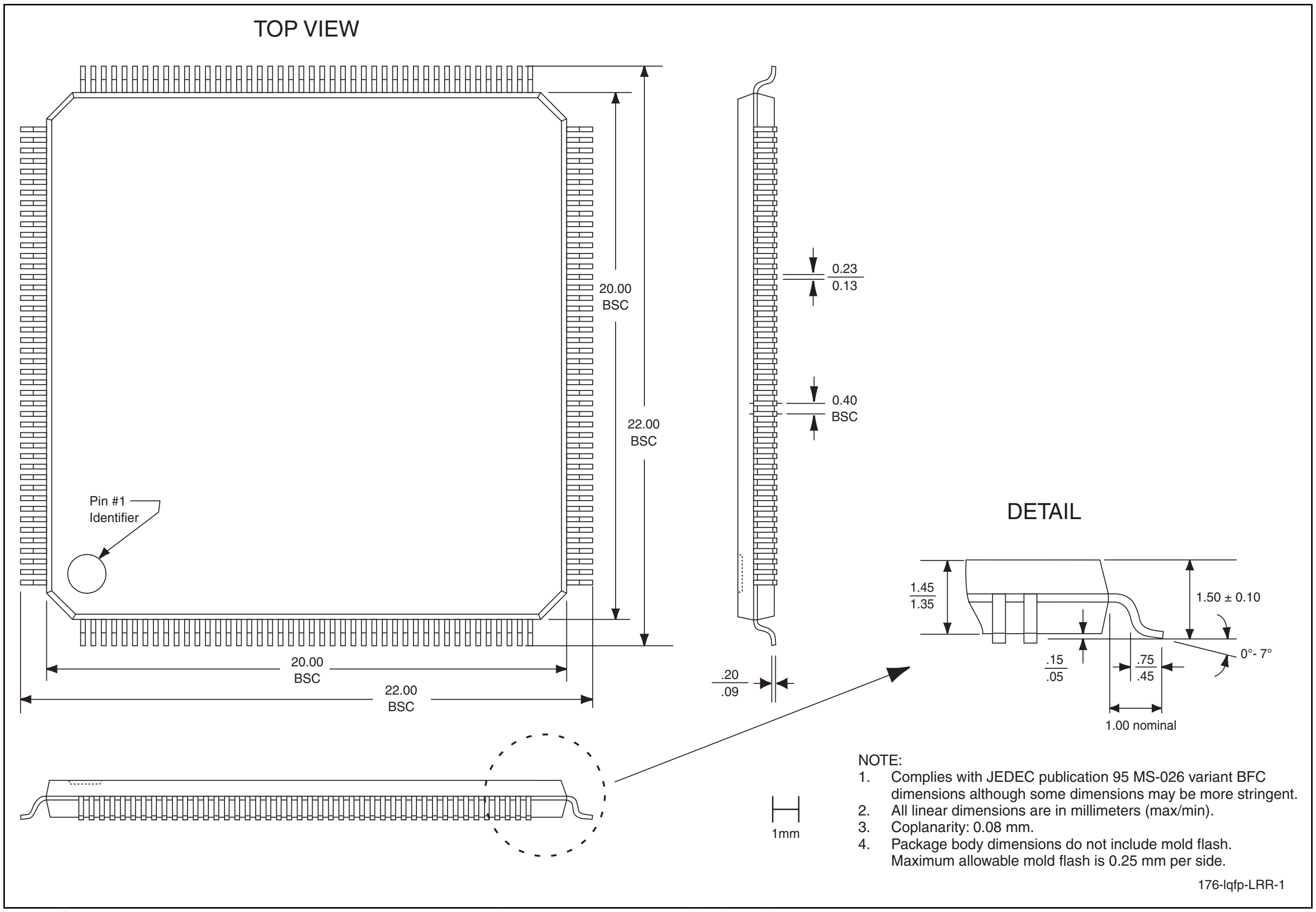
Valid combinations for SST79LF008

SST79LF008-33-C-LRRF

SST79LF008-33-C-BZYE

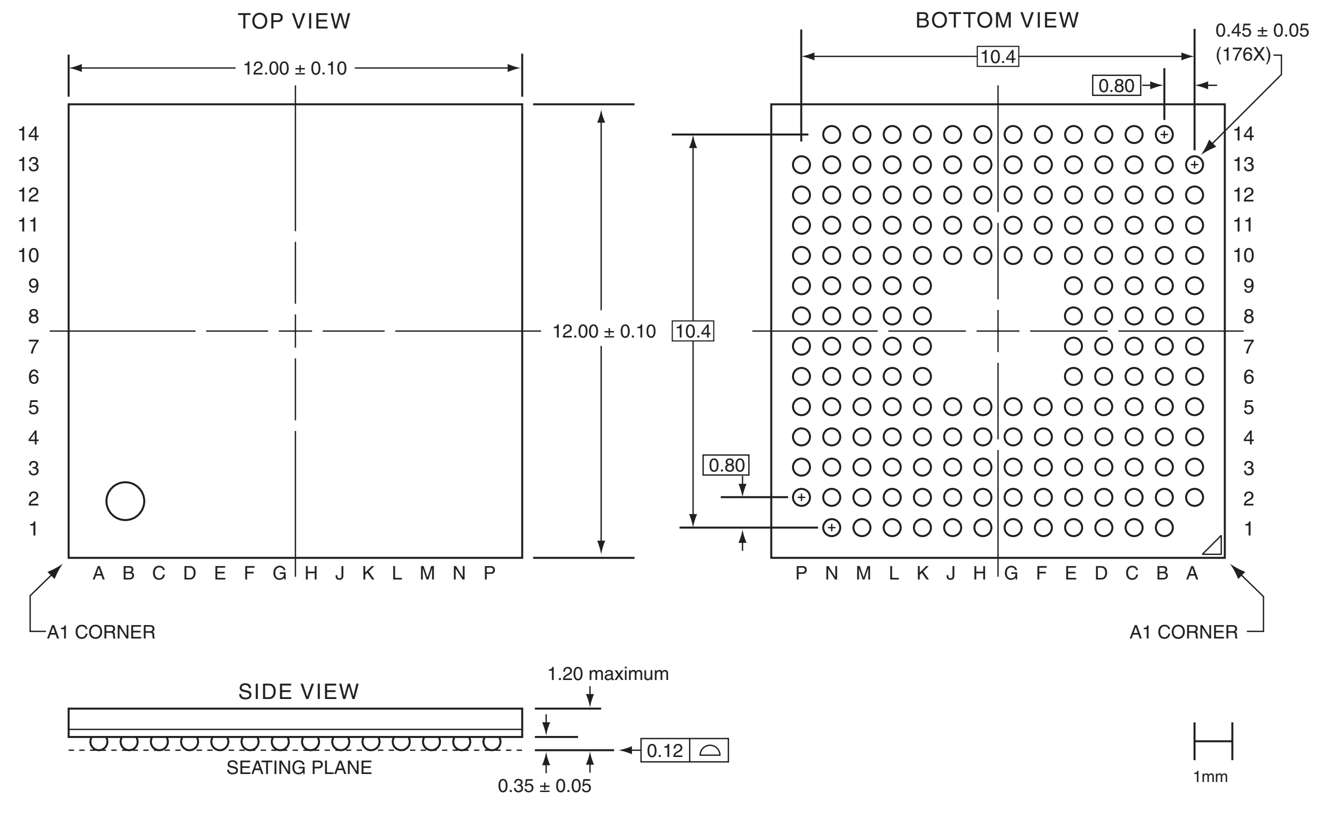
Note: Valid combinations are those products in mass production or will be in mass production. Consult your SST sales representative to confirm availability of valid combinations and to determine availability of new combinations.

26.0 PACKAGING DIAGRAMS



NOTE:
1. Complies with JEDEC publication 95 MS-026 variant BFC dimensions although some dimensions may be more stringent.
2. All linear dimensions are in millimeters (max/min).
3. Coplanarity: 0.08 mm.
4. Package body dimensions do not include mold flash. Maximum allowable mold flash is 0.25 mm per side.

**FIGURE 26-1: 176-lead Low-profile Quad Flat Pack (LQFP)
SST Package Code: LRR**



Note: 1. Although many dimensions are similar to those of JEDEC Publication 95, MO-210, this specific package is not registered.
2. All linear dimensions are in millimeters.
3. Coplanarity: 0.12 mm
4. Ball opening size is 0.38 mm (± 0.05 mm)

176-tfbga-BZY-12x12-450mic-2

FIGURE 26-2: 176-ball Thin-profile Fine-pitch Ball Grid Array (TFBGA) 12mm x 12mm SST Package Code: BZY

TABLE 26-1: Revision History

Number	Description	Date
00	• Initial Release of data sheet	May 2006
01	• In Figure 2-1: Pin Assignments, modified pin description for pins B1 thru B7. • In Figure 2-2: Pin Assignments, changed 052 to 152. • In Table 4-4: IAP Commands, changed No Operation to No Reserve in row 1, and added a new row at bottom for No Operation. • For Control Register “14.4.3.3 PS/2 Control Register 2 (PS2CR2)” on page 185, edited the PS2CRn_STOP[1:0] Function. • In Table 24-6: DC Characteristics, revised V_{IL} AND V_{IH} Min. and Max vaules. • Added paragraph (3rd) to “SPI Description” on page 159. • Revised SPI Transfer Formats, Figure 12-2 and Figure 12-3	Oct 2006