Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	SAM8RC
Core Size	8-Bit
Speed	12MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	LCD, LVD, LVR, PWM, WDT
Number of I/O	26
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-WFQFN Exposed Pad
Supplier Device Package	32-ELP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/s3f8s39xzz-lo89

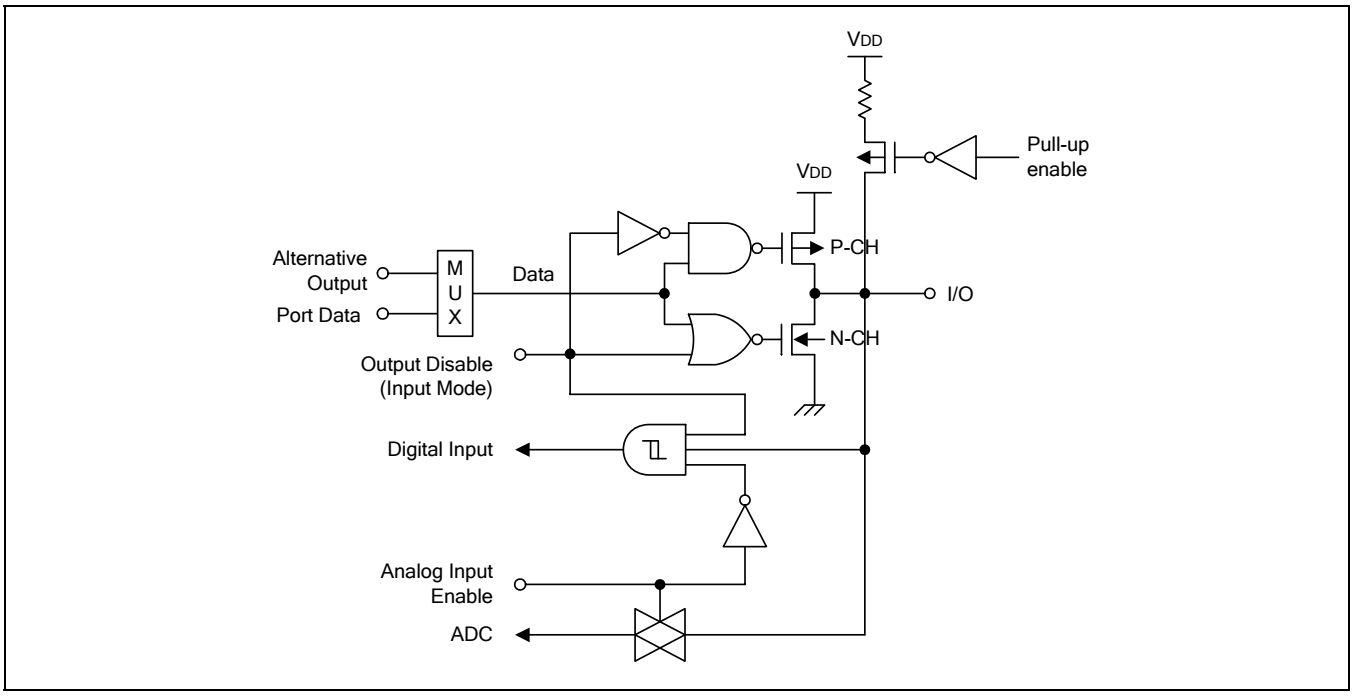


Figure 1-10 Pin Circuit Type E-1 (P0, P1)

Figure 1-11 illustrates the pin circuit type E-2 (P2.4-P2.7).

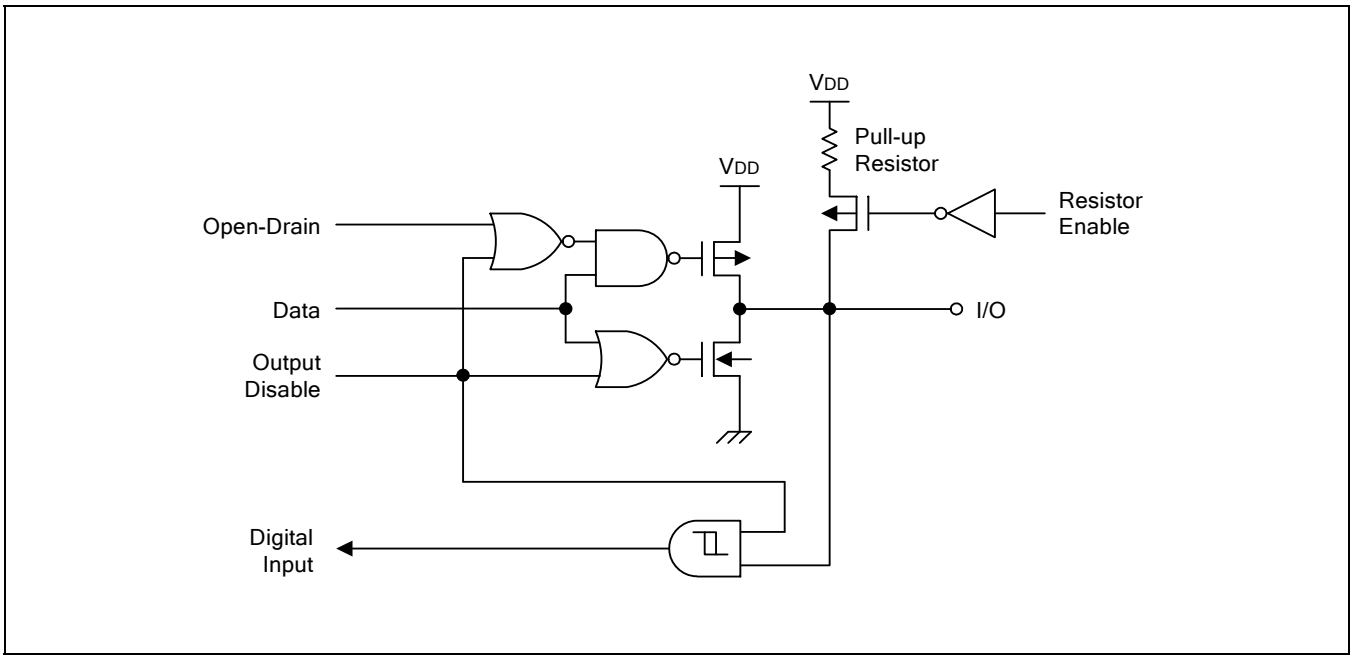


Figure 1-11 Pin Circuit Type E-2 (P2.4-P2.7)

Figure 3-6 illustrates the indirect working register addressing to program or data memory.

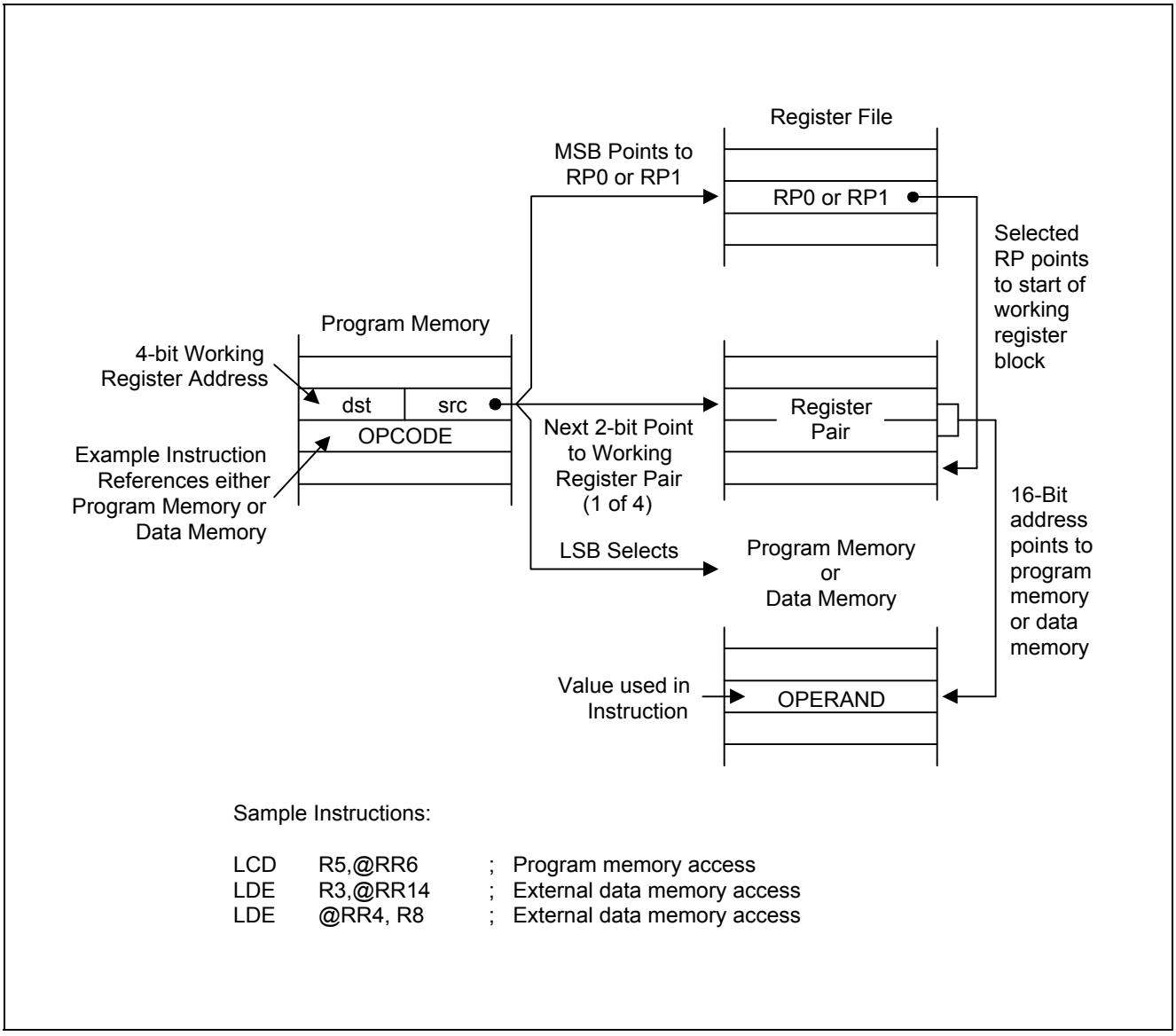


Figure 3-6 Indirect Working Register Addressing to Program or Data Memory

4.1.10 ICCR-Multi-Master IIC-Bus Clock Control Register (05H, PAGE8)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
nRESET Value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addressing Mode	Register addressing mode only							

.7	**Acknowledgement Enable Bit**	
	0	Acknowledgement disable mode
	1	Acknowledgement enable mode
.6	**Tx Clock Selection Bit**	
	0	fosc/16
	1	fosc/512
.5	**Multi-master IIC-Bus Tx/Rx Interrupt Enable Bit**	
	0	Disable
	1	Enable
.4	**IIC Interrupt Pending Bit**	
	0	Interrupt request is not pending; (when read) pending bit clear when write 0
	1	Interrupt request is pending (when read)
.3– .0	**ICCR.3-0: Transmit Clock 4-Bit Prescaler Bits**	
	SCL clock = IICLK/(CCR < 3:0 > + 1) where, IICLK = fosc/16 when IICR.6 is "0", IICLK = fosc/512 when ICCR.6 is "1"	

4.1.24 P2INT-Port 2 Interrupt Control Register (E6H, BANK1)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
nRESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addressing Mode	Register addressing mode only							

.7– .6	P2.3 (INT3) External Interrupt Configuration Bit		
	0	0	Disable interrupt
	0	1	Enable interrupt; interrupt on falling edge
	1	0	Enable interrupt; interrupt on rising edge
	1	1	Enable interrupt; interrupt on rising or falling edge
.5– .4	P2.2 (INT2) External Interrupt Configuration Bit		
	0	0	Disable interrupt
	0	1	Enable interrupt; interrupt on falling edge
	1	0	Enable interrupt; interrupt on rising edge
	1	1	Enable interrupt; interrupt on rising or falling edge
.3– .2	P2.1 (INT1) External Interrupt Configuration Bit		
	0	0	Disable interrupt
	0	1	Enable interrupt; interrupt on falling edge
	1	0	Enable interrupt; interrupt on rising edge
	1	1	Enable interrupt; interrupt on rising or falling edge
.1– .0	P2.0 (INT0) External Interrupt Configuration Bit		
	0	0	Disable interrupt
	0	1	Enable interrupt; interrupt on falling edge
	1	0	Enable interrupt; interrupt on rising edge
	1	1	Enable interrupt; interrupt on rising or falling edge

4.1.33 RESETID-Reset Source Indicating Register (00H, PAGE8)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
Read/Write	–	–	–	R/W	–	R/W	R/W	R/W
Addressing Mode	Register addressing mode only							
.7– .3	Not used							
.2	nReset pin Indicating Bit							
	0	Reset is not generated by nReset pin (when read)						
	1	Reset is generated by nReset pin (when read)						
.1	Basic Timer Reset Indicating Bit							
	0	Reset is not generated by Basic Timer (when read)						
	1	Reset is generated by Basic Timer (when read)						
.0	LVR Indicating Bit							
	0	Reset is not generated by LVR (when read)						
	1	Reset is generated by LVR (when read)						

State of RESETID Depends on Reset Source								
	.7	.6	.5	.4	.3	.2	.1	.0
LVR	–	–	–	–	–	0	0	1
Basic Timer or nReset pin	–	–	–	–	–	(3)	(3)	(2)

NOTE:

1. To clear an indicating register, write a "0" to indicating flag bit; writing a "1" to a reset indicating flag (RESETID.0-2) has no effect.
2. When a LVR reset occurs, it sets RESETID.0 and it clears all the other bits to "0" simultaneously.
3. When a basic timer or nRESET pin reset occurs, the corresponding bit will be set, but all other indicating bits won't be changed.

4.1.42 SWTCON-Stop Wake-Up Timer Control Register (FAH, Bank 0)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	–	0	–	0	0	0	0
Read/Write	R/W	–	R/W	–	R/W	R/W	R/W	R/W

.7	**Stop Wake-Up Timer Enable bit**	
	0	Disable Stop Wake-Up Timer
	1	Enable Stop Wake-Up Timer
.6	**Not used**	
.5	**Stop Wake-Up Timer Interrupt Enable bit**	
	0	Disable interrupt
	1	Enable interrupt
.4	**Not used**	
.3– .0	**Stop Wake-Up Timer prescaler bits (SWTPSB)**	
	SWTCLK = $F_{CLK}/(2^{SWTPSB[3-0]})$	

NOTE:

1. Prescaler values (SWTPSB) above 12 are NOT valid.
2. Bit 6 and .4 must kept as default value "0".

4.1.57 UART1CONL-UART 1 Control Register Low Byte (FAH, BANK1)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
nRESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Addressing Mode	Register addressing mode only							

.7	**UART 1 Transmit Parity bit Autogeneration Enable Bit**	
	0	Disable parity bit autogeneration
	1	Enable parity bit autogeneration
.6	**UART 1 Transmit Parity bit Selection Bit (for modes 2 and 3 only)**	
	0	Even parity bit
	1	Odd parity bit
	NOTE: If the UART1CONL.7 = 0, This bit is "don't care".	
.5	**UART 1 Receive Parity bit Selection Bit (for modes 2 and 3 only)**	
	0	Even parity bit check
	1	Odd parity bit check
	NOTE: If the UART1CONL.7 = 0, This bit is "don't care".	
.4	**UART 1 Receive Parity bit Error Status Bit (For modes 2 and 3 only)**	
	0	No parity bit error
	1	Parity bit error
	NOTE: If the UART1CONL.7 = 0, This bit is "don't care".	

.3-.2	**UART 1 Clock Selection Bits**		
	0	0	fxx/8
	0	1	fxx/4
	1	0	fxx/2
	1	1	fxx/1

.1	**UART 1 Transmit Interrupt Enable Bit**	
	0	Disable Tx interrupt
	1	Enable Tx interrupt
.0	**UART 1 Transmit Interrupt Pending Bit**	
	0	No interrupt pending (when read), clears pending bit (when write)
	1	Interrupt is pending (when read)

5.3.1 Interrupt Vector Addresses

All interrupt vector addresses for the S3F8S39/S3F8S35 interrupt structure is stored in the vector address area of the first 256 bytes of the program memory (ROM).

You can allocate unused locations in the vector address area as normal program memory. If you do so, ensure not overwrite any of the stored vector addresses.

The default program reset address in the ROM is 0100H.

Figure 5-3 illustrates the ROM vector address area.

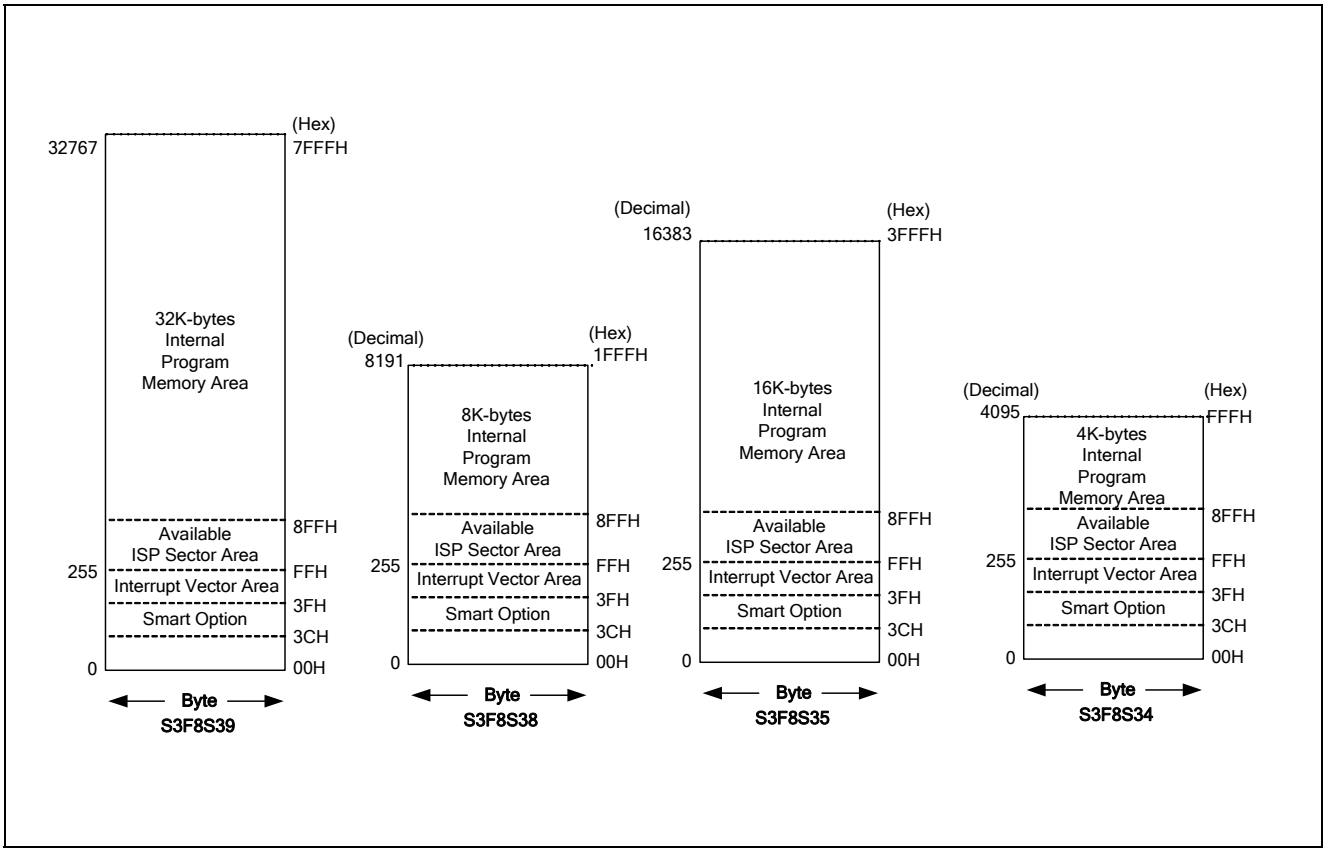


Figure 5-3 ROM Vector Address Area

5.3.2 Enable/Disable Interrupt Instructions

Executing the enable interrupts (EI) instruction globally enables the interrupt structure. All interrupts are then serviced as they occur according to the established priorities.

NOTE: The system initialization routine executed after a reset should always contain an EI instruction to globally enable the interrupt structure.

During the normal operation, you can execute the disable interrupt (DI) instruction at any time to globally disable interrupt processing. The EI and DI instructions change the value of bit 0 in the SYM register.

5.11 Interrupt Pending Function Types

This section includes:

- Overview
- Pending bits cleared automatically by hardware
- Pending bits cleared by the service routine

5.11.1 Overview

There are two types of interrupt pending bits:

- One type that is automatically cleared by hardware after interrupt service routine is acknowledged and executed.
- Another type that must be cleared in the interrupt service routine.

5.11.2 Pending Bits Cleared Automatically by Hardware

For interrupt pending bits that are cleared automatically by hardware, interrupt logic sets the corresponding pending bit to "1" when a request occurs. It then issues an IRQ pulse to inform the CPU that an interrupt is waiting to be serviced. The CPU acknowledges the interrupt source by sending an IACK, executes the service routine, and clears the pending bit to "0". This type of pending bit is not mapped and cannot, therefore, be read or written by application software.

In S3F8S39/S3F8S35 interrupt structure, TimerA, Timer0, Timer1, and Timer2 overflow interrupts, TimerB match interrupt and LVD interrupt belong to this category of interrupts, in which pending bits can be cleared automatically by hardware.

5.11.3 Pending Bits Cleared by the Service Routine

The second type of pending bit is the one that should be cleared by program software. The service routine should clear the appropriate pending bit before a return-from-interrupt subroutine (IRET) occurs. To do this, a "0" must be written to the corresponding pending bit location in the source's mode or control register.

6 Instruction Set

6.1 Overview

The SAM8RC instruction set supports large register files that are typical of most SAM8 microcontrollers. There are 78 instructions.

The powerful data manipulation capabilities and features of the instruction set are:

- Supports a full complement of 8-bit arithmetic and logic operations that includes multiplication and division
- No special I/O instructions (I/O control/data registers are mapped directly into the register file)
- Includes decimal adjustment in binary-coded decimal (BCD) operations
- 16-bit (word) data can be incremented and decremented
- Provides flexible instructions for bit addressing, rotate, and shift operations

6.1.1 Data Types

The SAM8 CPU performs operations on bits, bytes, BCD digits, and two-byte words. Bits in the register file can be set, cleared, complemented, and tested. Bits within a byte are numbered from 7 to 0, where bit 0 is the least significant (right-most) bit.

6.1.2 Register Addressing

To access an individual register, an 8-bit address in the range 0-255 or the 4-bit address of a working register is specified. Paired registers can be used to construct 16-bit data or 16-bit program memory or data memory addresses. Refer to Chapter 2 "Address Spaces" for more information about register addressing.

6.1.3 Addressing Modes

There are seven explicit addressing modes, namely:

- Register (R)
- Indirect Register (IR)
- Indexed (X)
- Direct (DA)
- Relative (RA)
- Immediate (IM)
- Indirect (IA)

Refer to Chapter 3 "Addressing Modes" for more information on these addressing modes.

6.6.15 CLR-Clear

CLR dst

Operation: dst ← "0"

Clears the destination location to "0".

Flags: Does not affect any other flags..

Format:

						Bytes	Cycles	Opcode (Hex)	Addr Mode dst
	opc	dst				2	4	B0	R
							4	B1	IR

Examples: Given: Register 00H = 4FH, register 01H = 02H, and register 02H = 5EH:

```
CLR     00H     →       Register 00H = 00H
CLR     @01H    →       Register 01H = 02H, register 02H = 00H
```

In Register (R) addressing mode, the statement "CLR 00H" clears the destination register 00H value to 00H. In the second example, the statement "CLR @01H" uses Indirect Register (IR) addressing mode to clear the 02H register value to 00H.

6.6.22 DEC-Decrement

DEC dst

Operation: dst ← dst-1

The contents of the destination operand are decremented by one.

Flags:

C: Does not affect.

Z: Sets if the result is "0"; otherwise it clears.

S: Sets if result is negative; otherwise it clears.

V: Sets if arithmetic overflow occurred; otherwise it clears.

D: Does not affect.

H: Does not affect.

Format:

						Bytes	Cycles	Opcode (Hex)	Addr Mode dst
	opc	dst				2	4	00	R
							4	01	IR

Examples: Given: R1 = 03H and register 03H = 10H:

```
DEC     R1      →       R1 = 02H
DEC     @R1     →       Register 03H = 0FH
```

In the first example, if working register R1 contains the value 03H, the statement "DEC R1" decrements the hexadecimal value by1, leaving the value 02H. In the second example, the statement "DEC @R1" decrements the value 10H contained in the destination register 03H by 1, leaving the value 0FH.

8.1.1 MCU Initialization Sequence

These are the sequence of events that occur during a Reset operation:

- Disables all interrupts.
- Enables watchdog function (basic timer).
- Sets ports 0-4 to input mode
- Peripheral control and data registers reset to their initial values (Refer to *Table 8-1* for more information.)
- Loads the program counter with the ROM reset address, 0100H or other values set by the Smart Option.
- When the programmed oscillation stabilization time interval has elapsed, the address stored in the first and second bytes of reset address in ROM is fetched and executed.

Figure 8-2 illustrates the block diagram of Reset.

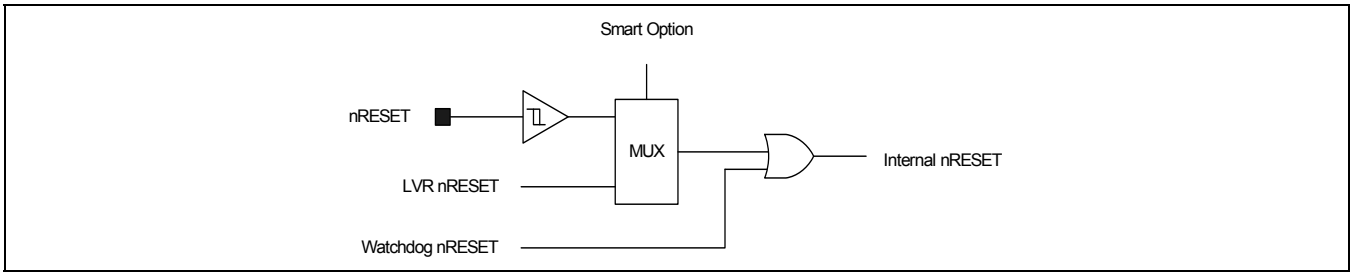


Figure 8-2 Reset Block Diagram

Figure 8-3 illustrates the timing for S3F8S39/S3F8S35 after Reset.

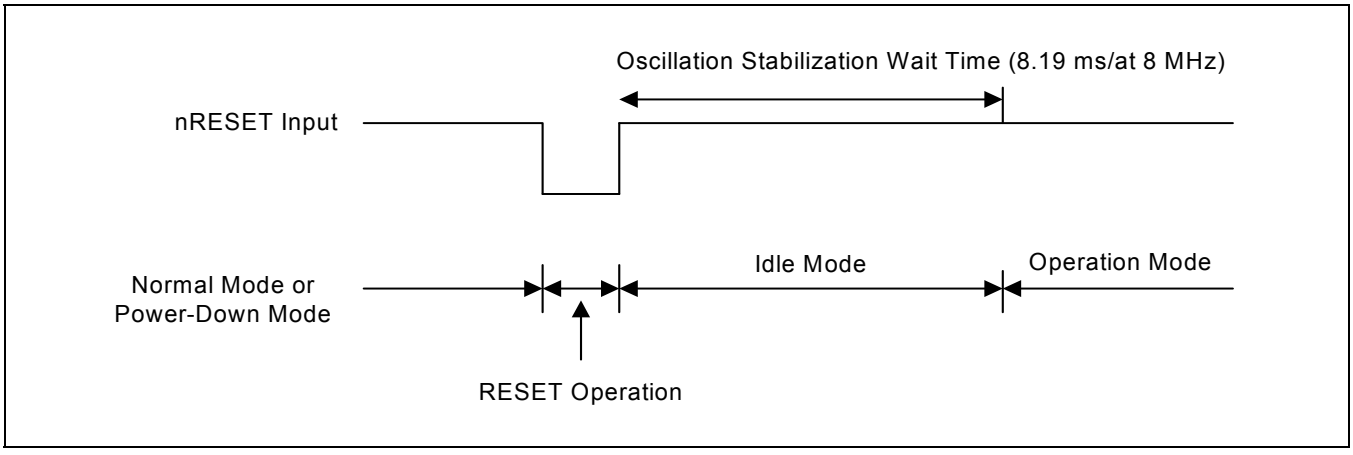


Figure 8-3 Timing for S3F8S39/S3F8S35 after reset

11.2 Timer A Control Register (TACON)

You use the Timer A control register, TACON, to

- Select the Timer A operating mode (interval timer, capture mode, or PWM Mode)
- Select the Timer A input clock frequency
- Clear the Timer A counter and TACNT
- Enable the Timer A overflow interrupt or Timer A match/capture interrupt

TACON is located in set 1, Bank 0 at address E2H, and is Read/Write addressable using Register addressing mode.

A reset clears TACON to "00H". This sets Timer A to normal interval timer mode, selects an input clock frequency of fxx, and disables all Timer A interrupts. You can clear the Timer A counter at any time during normal operation by writing a "1" to TACON.5.

The Timer A overflow interrupt (TAOVF) is interrupt level IRQ0 and has the vector address D0H. When a Timer A overflow interrupt occurs and is serviced by the CPU, the pending condition is cleared automatically by hardware or must be cleared by software.

To enable the Timer A match/capture interrupt (IRQ0, vector CEH), you should write TACON.3 to "1". To detect a match/capture interrupt pending condition, the application program polls TACON.1. When a "1" is detected, a Timer A match or capture interrupt is pending. When the interrupt request has been serviced, the pending condition must be cleared by software by writing a "0" to the Timer A match/capture interrupt pending bit, TACON.1.

*Figure 11-1*illustrates the Timer A control register.

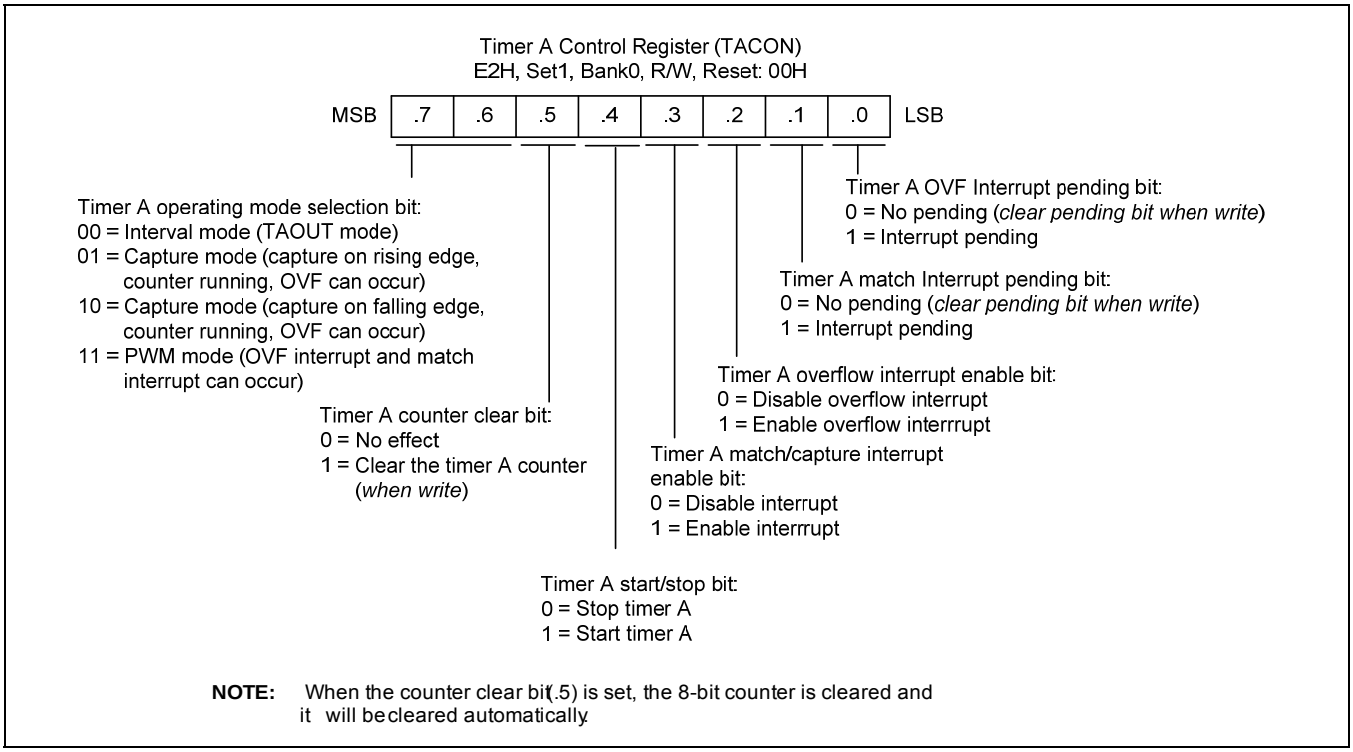


Figure 11-1 Timer A Control Register (TACON)

16.1.4 Conversion Timing

The A/D conversion process requires four steps (four clock edges) to convert each bit and 10 clocks to step-up A/D conversion. Therefore, it requires total of 50 clocks to complete a 10-bit conversion: With a 12 MHz CPU clock frequency, one clock cycle is 333 ns (4/fxx). If each bit conversion requires four clocks, it calculates the conversion rate as:

- 4 clocks/bit x 10-bits + step-up time (10 clock) = 50 clocks
- 50 clock × 333 μs = 16.65 μs at 12 MHz, 1 clock time = 4/fxx (assuming ADCON.2–1 = 10)

16.1.5 Internal A/D Conversion Procedure

The procedure for internal A/D Conversion is:

1. Analog input should remain between the voltage range of V_{SS} and V_{DD}.
2. Configure the analog input pins to input mode by making the appropriate settings in P0CONH, P0CONL, P1CONL, P3CONH, and P3CONL registers.
3. Before the conversion operation starts, you should select one of the sixteen input pins (ADC0–ADC15) by writing the appropriate value to the ADCON register.
4. When it completes the conversion, (16 clocks have elapsed), the EOC flag is set to "1", so that a check can be made to verify that the conversion was successful.
5. Loads the converted digital value to the output register, ADDATAH (8-bit) and ADDATAL (2-bit), and then the ADC module enters an idle state.
6. You can now read the digital conversion result from the ADDATAH and ADDATAL register.

Figure 16-5 illustrates the recommended A/D Converter circuit for highest absolute accuracy.

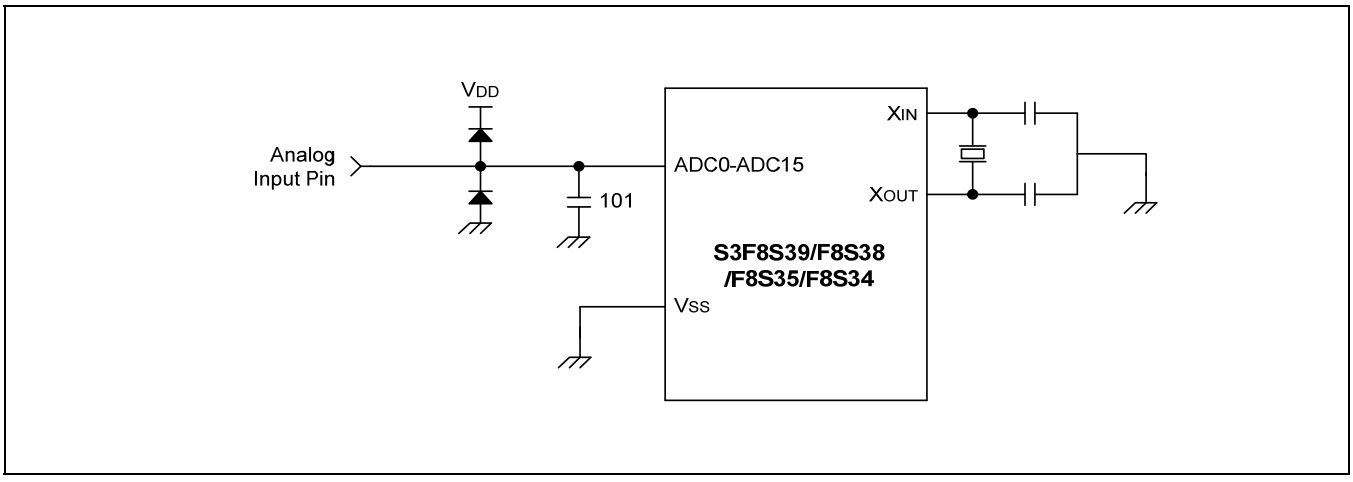


Figure 16-5 Recommended A/D Converter Circuit for Highest Absolute Accuracy

18.1.2 Multi-Master IIC-Bus Control/Status Register

The multi-master IIC-bus control/status register (ICSR) is located at address FDH, BANK1. 4 bits in this register, ICSR.3-ICSR.0, are read-only status flags.

ICSR register settings are used to control or monitor the IIC-bus functions (Refer to *Figure 18-2* for more information). The IIC-bus functions are:

- Selects master/slave transmit or receive mode Provide an IIC-bus busy status flag
- Enables/disables Serial output Provide failed bus arbitration procedure status flag
- Provide Slave address/address register match or general call received status flag
- Provide Slave address 00000000B (general call) received status flag
- Provide last received bit status flag (not ACK = "1", ACK = "0")

Figure 18-2 illustrates the multi-master IIC-bus control/status register.

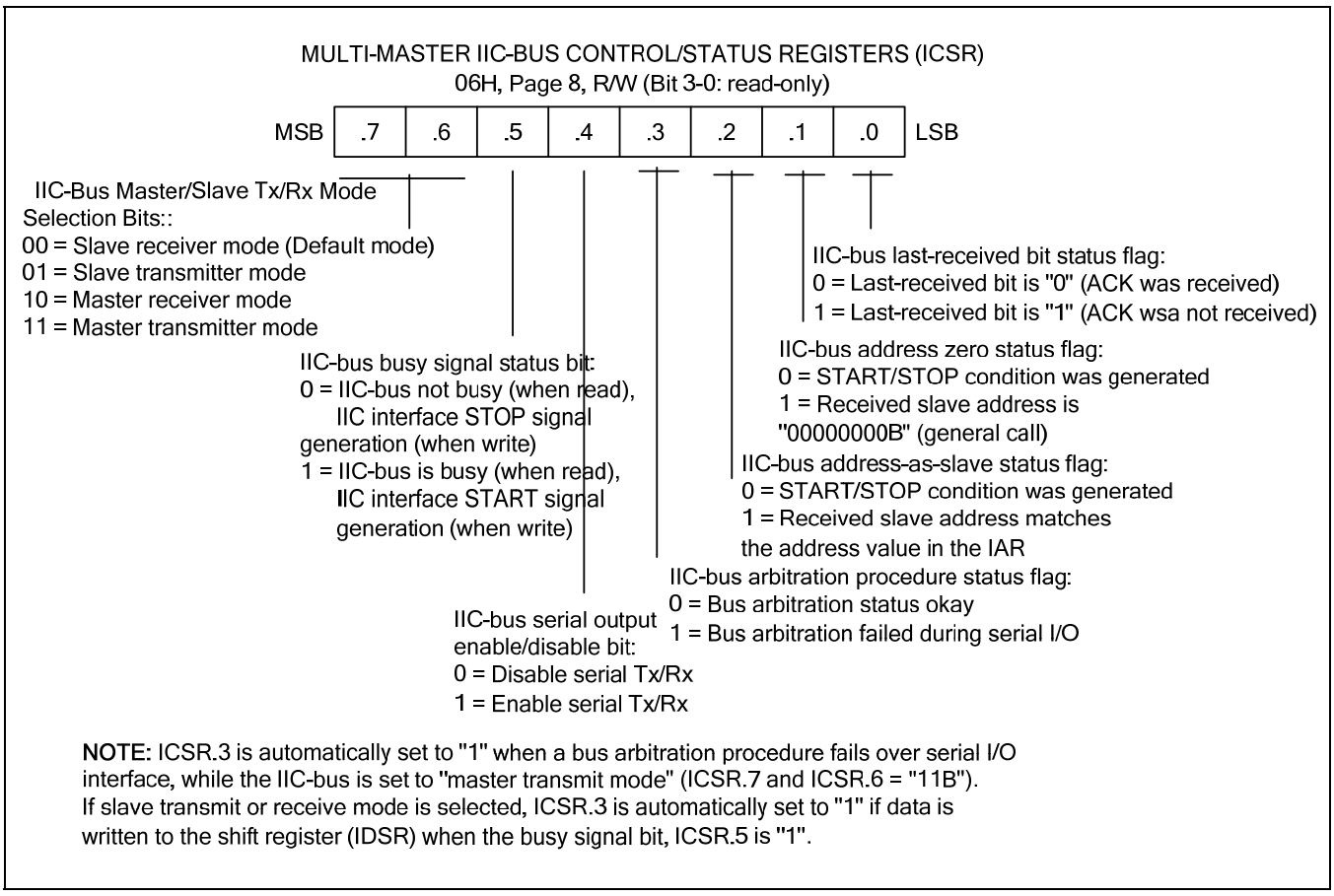


Figure 18-2 Multi-Master IIC-Bus Control/Status Register (ICSR)

18.2 Block Diagram

Figure 18-5 illustrates the block diagram of IIC-bus.

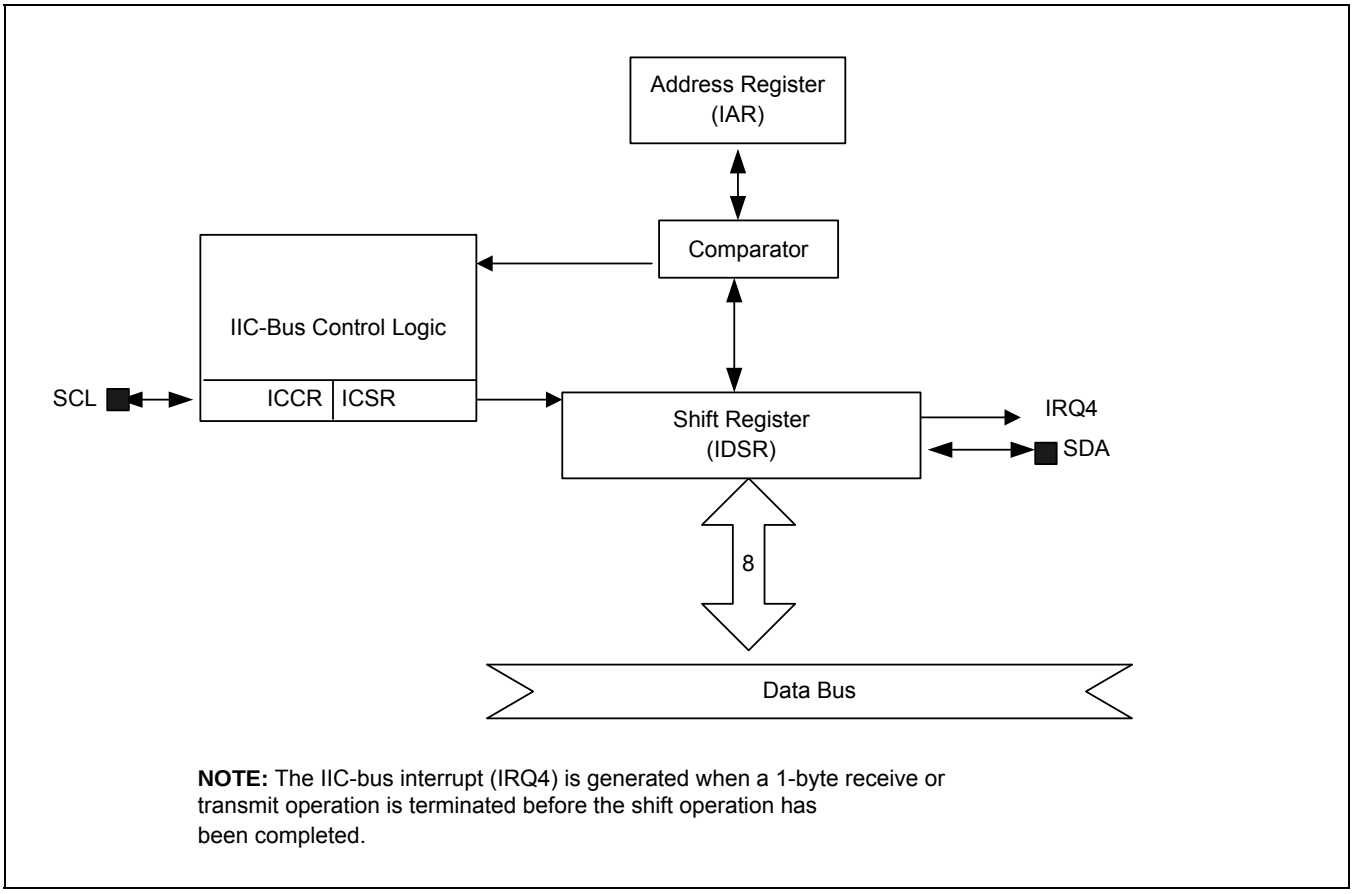


Figure 18-5 IIC-Bus Block Diagram

19.6 Serial Port Mode 1 Function Description

In mode 1, 10 bits are transmitted (through the TxD0 (P2.6) pin) or received (through the RxD0 (P2.7) pin).

The three components of each data frame are:

- Start bit ("0")
- 8 data bits (LSB first)
- Stop bit ("1")

The baud rate for mode 1 is variable.

19.6.1 Mode 1 Transmit Procedure

The steps of Mode 1 transmit procedure are:

1. Select the UART 0 clock, UART0CONL.3 and .2.
2. Set the UART 0 transmit parity bit autogeneration enable or disable bit (UART0CONL.7).
3. Select the baud rate that BRDATA0 generates.
4. Select mode 1 (8-bit UART) by setting UART0CONH bits 7 and 6 to "01B".
5. Write transmission data to the shift register UDATA0 (F0H, set 1, bank 0). The start and stop bits are generated automatically by hardware.

19.6.2 Mode 1 Receive Procedure

The steps of Mode 1 receive procedure are:

1. Select the UART 0 clock, UART0CONL.3 and .2.
2. Set the UART 0 transmit parity bit autogeneration enable or disable bit (UART0CONL.7).
3. Select the baud rate to be generated by BRDATA0.
4. Select mode 1 and set the Receive Enable (RE) bit in the UART0CONH register to "1".
5. The start bit low ("0") condition at the RxD0 (P2.7) pin will cause the UART 0 module to start the serial data receive operation.

25.3.2 Reference Table for Connection

Table 25-2 lists the reference table for connection.

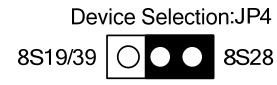
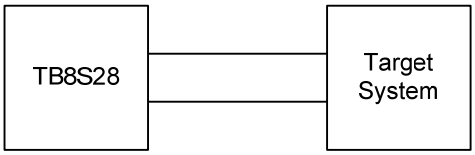
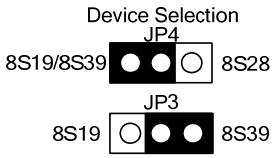
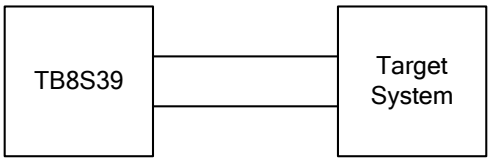
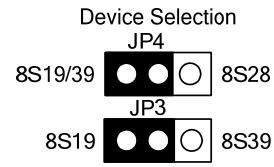
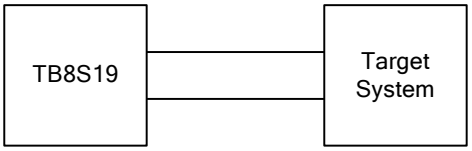
Table 25-2 Reference Table for Connection

Pin Name	I/O Mode in Applications	Resistor (Need)	Required Value
V_{PP} (TEST)	Input	Yes	R_{Vpp} is 10 kΩ to 50 kΩ. C_{Vpp} is 0.01 μF to 0.02 μF.
nRESET	Input	Yes	R_{nRESET} is 2 kΩ to 5 kΩ. C_{nRESET} is 0.01 μF to 0.02 μF.
SDAT (I/O)	Input	Yes	R_{SDAT} is 2 kΩ to 5 kΩ.
	Output	No (NOTE)	–
SCLK (I/O)	Input	Yes	R_{SCLK} is 2 kΩ to 5 kΩ.
	Output	No (NOTE)	–

NOTE: In the on-board writing mode, very high-speed signal will be provided to pin SCLK and SDAT. It also causes some damages to the application circuits that are connected to SCLK or SDAT port if the application circuit is designed as high speed response such as relay control circuit. If possible, set the I/O configuration of SDAT and SCLK pins to input mode.
The value of R and C in this table is a recommended value. It varies with circuit of system.

Table 26-2 lists the device selection settings for TB8S19/8S28/8S39.

Table 26-2 Device Selection Settings for TB8S19/8S28/8S39

"Device Selection" Settings	Operating Mode	Comments
		Operate with TB8S28
		Operate with TB8S39
		Operate with TB8S19

NOTE: The following symbol in the "8S28" Setting column indicates the electrical short (off) configuration: