Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. All of the slaves may be contacted by using the Broadcast address. Two Special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0 SADDR = 1100 0000
SADEN = 1111 1101
Given = 1100 00X0

Slave 1 SADDR = 1100 0000
SADEN = 1111 1110
Given = 1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two salves. Slave 0 requires as 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for Slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for Slave 0) and bit 1 = 0 (for Slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select Slaves 1 and 2 while excluding Slave 0:

Slave 0 SADDR = 1100 0000
SADEN = 1111 1001
Given = 1100 0XX0

Slave 1 SADDR = 1110 0000
SADEN = 1111 1010
Given = 1110 0X0X

Slave 2 SADDR = 1110 0000
SADEN = 1111 1100
Given = 1110 00XX

In the above example the differentiation among the 3 Slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude Slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are trended as don't cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are leaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

15 SIO1, I^{2}C SERIAL IO

The I^{2}C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Bidirectional data transfer between masters and slaves
- Multimaster bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- The I^{2}C bus may be used for test and diagnostic purposes

The I/O pins P1.6 and P1.7 must be set to Open Drain (SCL and SDA).

The 8xC591 on-chip I^{2}C logic provides a serial interface that meets the I^{2}C bus specification. The SIO1 logic handles bytes transfer autonomously. It also keeps track of serial transfers, and a status register (S1STA) reflects the status of SIO1 and the I^{2}C bus.

The CPU interfaces to the I^{2}C logic via the following four special function registers: S1CON (SIO1 control register), S1STA (SIO1 status register), S1DAT (SIO1 data register), and S1ADR (SIO1 slave address register). The SIO1 logic interfaces to the external I^{2}C bus via two port 1 pins: P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I^{2}C bus configuration is shown in Figure 30, and Figure 31 shows how a data transfer is accomplished on the bus. Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I^{2}C bus:

1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a not acknowledge is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I^{2}C bus will not be released.

15.1 Modes of Operation

The on-chip SIO1 logic may operate in the following four modes:

1. **Master Transmitter Mode:**
   Serial data output through P1.7/SDA while P1.6/SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and we say that a “W” is transmitted. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.
2. **Master Receiver Mode:**
   The first byte transmitted contains the slave address of the transmitting device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 1, and we say that an “R” is transmitted. Thus the first byte transmitted is SLA+R. Serial data is received via P1.7/SDA while P1.6/SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.
3. **Slave Receiver Mode:**
   Serial data and the serial clock are received through P1.7/SDA and P1.6/SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.
4. **Slave Transmitter Mode:**
   The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via P1.7/SDA while the serial clock is input through P1.6/SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In a given application, SIO1 may operate as a master and as a slave. In the slave mode, the SIO1 hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, SIO1 switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

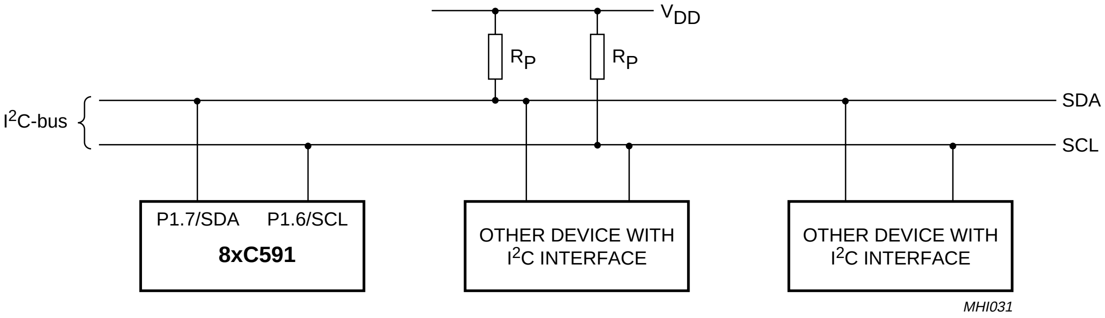

V_DD
R_P
R_P
I2C-bus
SDA
SCL
P1.7/SDA
P1.6/SCL
8xC591
OTHER DEVICE WITH I2C INTERFACE
OTHER DEVICE WITH I2C INTERFACE
MHI031

Fig.30 Typical I^{2}C Bus configuration.

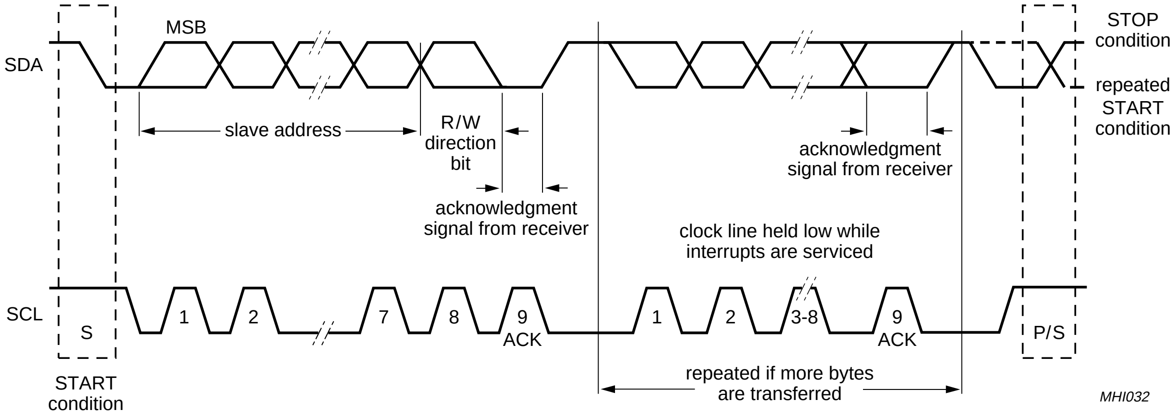

SDA
SCL
MSB
slave address
R/W direction bit
acknowledgment signal from receiver
acknowledgment signal from receiver
clock line held low while interrupts are serviced
STOP condition
repeated START condition
S
1
2
7
8
9
ACK
1
2
3-8
9
ACK
P/S
START condition
repeated if more bytes are transferred
MHI032

Fig.31 Data Transfer on the I^{2}C Bus.

15.2 SIO1 Implementation and Operation

Figure 32 shows how the on-chip I^{2}C bus interface is implemented, and the following text describes the individual blocks.

15.2.1 INPUT FILTERS AND OUTPUT STAGES

The input filters have I^{2}C compatible input levels. If the input voltage is less than 1.5 V, the input logic level is interpreted as 0; if the input voltage is greater than 3.0 V, the input logic level is interpreted as 1. Input signals are synchronized with the internal clock ($f_{CLK}/4$), and spikes shorter than three oscillator periods are filtered out.

The output stages consist of open drain transistors that can sink 3 mA at $V_{OUT} < 0.4$ V. These open drain outputs do have clamping diodes to V_{DD}. Thus, precautions have to be considered, if a powered-down 8xC591 on one board clamps the I^{2}C bus externally.

15.2.2 ADDRESS REGISTER, S1ADR

This 8-bit special function register may be loaded with the 7-bit slave address (7 most significant bits) to which SIO1 will respond when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable general call address (00H) recognition.

15.2.3 COMPARATOR

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in S1ADR). It also compares the first received 8-bit byte with the general call address (00H). If an equality is found, the appropriate status bits are set and an interrupt is requested.

15.2.4 SHIFT REGISTER, S1DAT

This 8-bit special function register contains a byte of serial data to be transmitted or a byte which has just been received. Data in S1DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

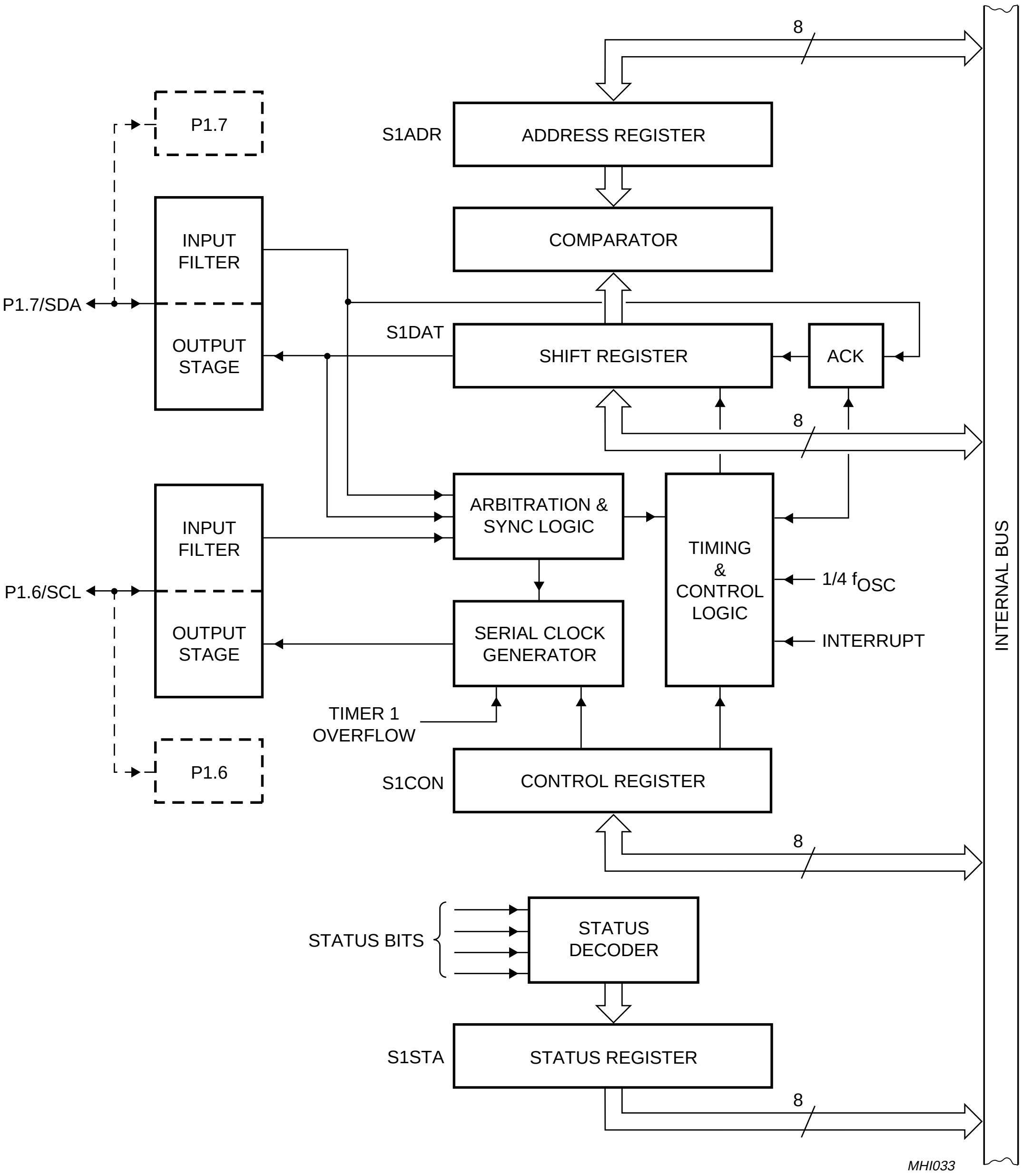


Fig.32 I^{2}C Bus Interface Block Diagram.

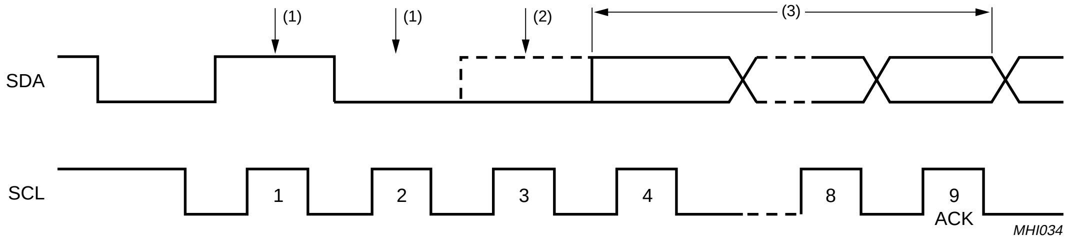
15.2.5 ARBITRATION AND SYNCHRONIZATION LOGIC

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I^{2}C bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a not acknowledge: (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses. Figure 33 shows the arbitration procedure.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the mark duration is determined by the device that generates the shortest marks, and the space duration is determined by the device that generates the longest spaces. Figure 34 shows the synchronization procedure.

A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

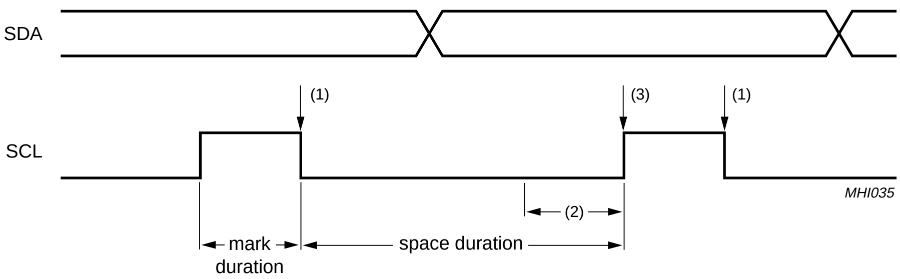


(1) Another device transmits identical serial data.

(2) Another device overrules a logic 1 (dotted line) transmitted by SIO1 (master) by pulling the SDA line low. Arbitration is lost, and SIO1 enters the slave receiver mode.

(3) SIO1 is in the slave receiver mode but still generates clock pulses until the current byte has been transmitted. SIO1 will not generate clock pulses for the next byte. Data on SDA originates from the new master once it has won arbitration.

Fig.33 Arbitration Procedure.



(1) Another service pulls the SCL line low before the SIO “mask” duration is complete. The serial clock generator is immediately reset and commences with the “space” duration by pulling SCL low.

(2) Another device still pulls the SCL line low after SIO1 releases SCL. The serial clock generator is forced into the wait state until the SCL line is released.

(3) The SCL line is released, and the serial clock generator commences with the mark duration.

Fig.34 Serial Clock Synchronization.

15.2.6 SERIAL CLOCK GENERATOR

This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the master transmitter or master receiver mode. It is switched off when SIO1 is in a slave mode. The programmable output clock frequencies are: $f_{CLK}/120$, $f_{CLK}/9600$, and the Timer 1 overflow rate divided by eight. The output clock pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

15.2.7 TIMING AND CONTROL

The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for S1DAT, enables the comparator, generates and detects start and stop conditions, receives and transmits acknowledge bits, controls the master and slave modes, contains interrupt request logic, and monitors the I^2C bus status.

15.2.8 CONTROL REGISTER, S1CON

This 7-bit special function register is used by the microcontroller to control the following SIO1 functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

15.2.9 STATUS DECODER AND STATUS REGISTER

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I^2C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines (see the software example in this section).

15.2.10 THE FOUR SIO1 SPECIAL FUNCTION REGISTERS

The microcontroller interfaces to SIO1 via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

15.2.10.1 The Address Register, S1ADR

The CPU can read from and write to this 8-bit, directly addressable SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a master mode. In the slave modes, the seven most significant bits must be loaded with the microcontrollers own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.

The most significant bit corresponds to the first bit received from the I^{2}C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I^{2}C bus, and a logic 0 corresponds to a low level on the bus.

Table 51 Address Register S1ADR (address DBH)

7	6	5	4	3	2	1	0
X	X	X	X	X	X	X	GC

Table 52 Description of S1ADR (DBH) bits

BIT	SYMBOL	DESCRIPTION
7 to 1	X	Own slave address.
0	GC	0 = general call address is not recognized. 1 = general call address is recognized.

15.2.11 THE DATA REGISTER, S1DAT

S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer (BSD7) on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, BSD7 is loaded with the content of S1DAT.7, which is the first bit to be transmitted to the SDA line (see Figure 36). After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

Table 53 Address Register S1DAT (address DAH)

7	6	5	4	3	2	1	0
SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Table 54 Description of S1DAT (DAH) bits

BIT	SYMBOL	DESCRIPTION
7 to 0	SD7 to SD0	Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I^{2}C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 35 shows how data in S1DAT is serially transferred to and from the SDA line.

15.2.12 THE CONTROL REGISTER, S1CON

The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^2C bus. The STO bit is also cleared when ENS1 = 0.

Table 55 Address Register S1CON (address D8H)

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0

Table 56 Description of S1CON (D8H) bits

BIT	SYMBOL	DESCRIPTION
7	CR2	Clock rate bit 2, see Table 57.
6	ENS1	**Enable serial I/O**. ENS1 = 0: I^2C I/O disabled and reset. ENS1 = 1: serial I/O enabled.
5	STA	**START flag.** When this bit is set in slave mode, the hardware checks the I^2C-bus and generates a START condition if the bus is free or after the bus becomes free. If the device operates in master mode it will generate a repeated START condition.
4	STO	**STOP flag**. If this bit is set in a master mode a STOP condition is generated. A STOP condition detected on the I^2C-bus clears this bit. This bit may also be set in slave mode in order to recover from an error condition. In this case no STOP condition is generated to the I^2C-bus, but the hardware releases the SDA and SCL lines and switches to the not selected receiver mode. The STOP flag is cleared by the hardware.
3	SI	**Serial Interrupt flag.** This flag is set and an interrupt request is generated, after any of the following events occur: • A START condition is generated in master mode. • The own slave address has been received during AA = 1. • The general call address has been received while S1ADR.0 and AA = 1. • A data byte has been received or transmitted in master mode (even if arbitration is lost). • A data byte has been received or transmitted as selected slave. • A STOP or START condition is received as selected slave receiver or transmitter. While the SI flag is set, SCL remains LOW and the serial transfer is suspended. SI must be reset by software.
2	AA	**Assert Acknowledge flag.** When this bit is set, an acknowledge is returned after any one of the following conditions: • Own slave address is received. • General call address is received (S1ADR.0 = 1). • A data byte is received, while the device is programmed to be a master receiver. • A data byte is received. while the device is a selected slave receiver. When the bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own address or general call address is received.
1	CR1	Clock rate bits 1 and 0; see Table 57.
0	CR0	

15.2.12.1 ENS1, the SIO1 enable bit

ENS1 = “0”: When ENS1 is “0”, the SDA and SCL input signals are ignored, SIO1 is in the not addressed slave state, and the STO bit in S1CON is forced to 0. No other bits are affected.

ENS1 = “1”: When ENS1 is 1, I^{2}C is enabled. Note, that P1.6 and P1.7 have to set to Open Drain by writing the Port mode registers P1M1.x and P1M2.x bits 6 and 7 with a 1 (see Section 6.2 “Pin description”).

ENS1 should not be used to temporarily release SIO1 from the I^{2}C bus since, when ENS1 is reset, the I^{2}C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

In the following text, it is assumed that ENS1 = 1.

15.2.12.2 STA, the START flag

STA = “1”: When the STA bit is set to enter a master mode, the SIO1 hardware checks the status of the I^{2}C bus and generates a START condition if the bus is free. If the bus is not free, then SIO1 waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal serial clock generator.

If STA is set while SIO1 is already in a master mode and one or more bytes are transmitted or received, SIO1 transmits a repeated START condition. STA may be set at any time. STA may also be set when SIO1 is an addressed slave.

STA = “0”: When the STA bit is reset, no START condition or repeated START condition will be generated.

15.2.12.3 STO, the STOP Flag

STO = “1”: When the STO bit is set while SIO1 is in a master mode, a STOP condition is transmitted to the I^{2}C bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I^{2}C bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined not addressed slave receiver mode. The STO flag is automatically cleared by hardware.

If the STA and STO bits are both set, the a STOP condition is transmitted to the I^{2}C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.

STO = “0”: When the STO bit is reset, no STOP condition will be generated.

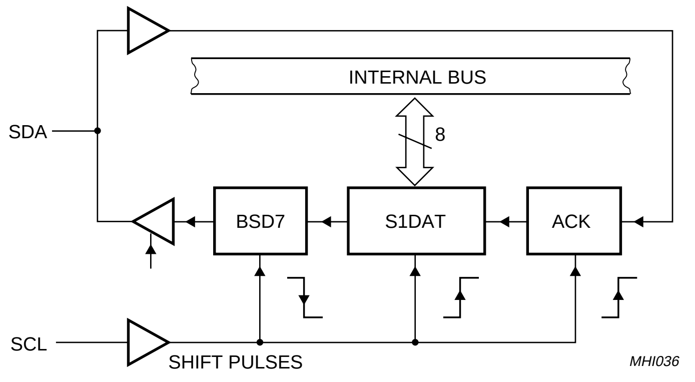
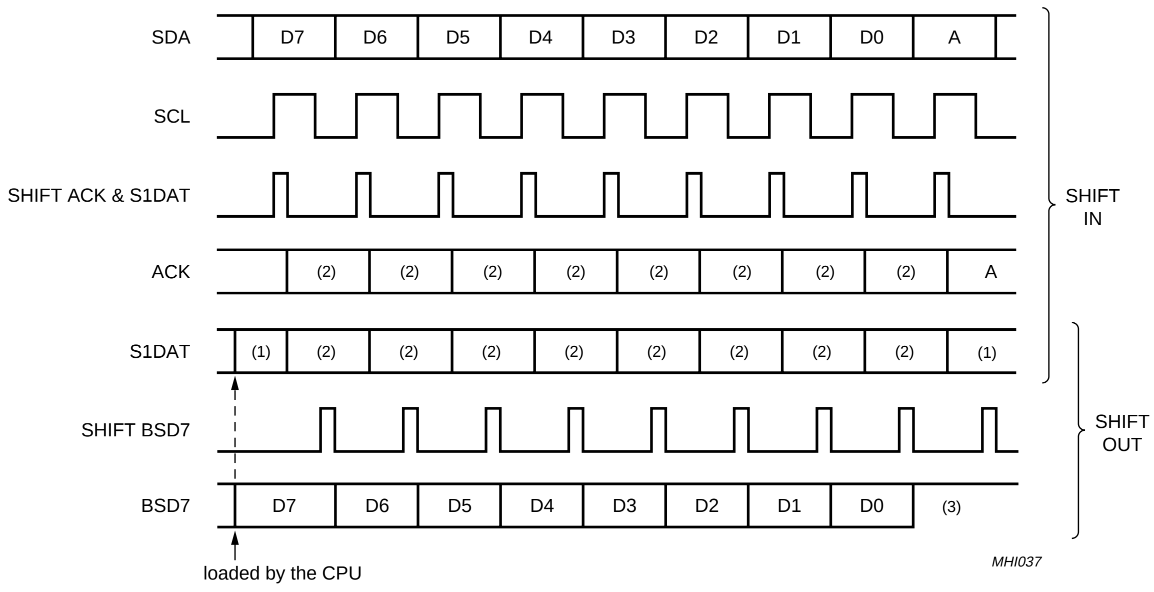


Fig.35 Serial Input/Output Configuration.



(1) Valid data in S1DAT.
(2) Shifting data in S1DAT and ACK.
(3) High level on SDA.

Fig.36 Shift-in and Shift-out Timing.

15.2.12.4 SI, the Serial Interrupt Flag

SI = “1”: When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = 0: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

15.2.12.5 AA, the Assert Acknowledge flag

AA = “1”: If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

- The “own slave address” has been received
- The general call address has been received while the general call bit (GC) in S1ADR is set
- A data byte has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:

- A data has been received while SIO1 is in the master receiver mode
- A data byte has been received while SIO1 is in the addressed slave receiver mode

When SIO1 is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 40). When SI is cleared, SIO1 leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I^2C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the parts own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

15.2.12.6 CR0, CR1, and CR2, the Clock Rate Bits

These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 57.

A 12.5 kHz bit rate may be used by devices that interface to the I^2C bus via standard I/O port lines which are software driven and slow. 100kHz is usually the maximum bit rate and can be derived from a 16 MHz, 12 MHz, or a 6 MHz oscillator. A variable bit rate (0.5 kHz to 62.5 kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 57 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100 kHz.

15.2.13 THE STATUS REGISTER, S1STA

S1STA is an 8-bit read-only special function register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains F8H, no relevant state information is available and no serial interrupt is requested. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = “1”). A valid status code is present in S1STA one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

Table 57 Serial clock rate

CR2	**CR1**	**CR0**	**BIT FREQUENCY (kHz) at f_{CLK}**			**f_{CLK} DIVIDED BY**
			6 MHz	**8 MHz**	**12 MHz**	
0	0	0	47	62.5	94	128
0	0	1	54	71	107(1)	112
0	1	0	63	83.3	125(1)	96
0	1	1	75	100	150(1)	80
1	0	0	12.5	17	25	480
1	0	1	100	133(1)	200(1)	60
1	1	0	200	267(1)	400(1)	30
1	1	1	0.49 > 62.5 0 < 254	0.65 < 55.6 0 < 253	0.98 < 50.0 0 < 251	48 x (256 (reload value Timer 1)) Reload value Timer 1 in Mode 2.

Note

1. These frequencies exceed the upper limit of 100 kHz of the standard I^{2}C-bus specification.

15.2.14 MORE INFORMATION ON SIO1 OPERATING MODES

The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in Figures 37 to 40. These figures contain the following abbreviations:

Abbreviation	***Explanation***
S	*Start condition*
SLA	*7-bit slave address*
R	*Read bit (high level at SDA)*
W	*Write bit (low level at SDA)*
A	*Acknowledge bit (low level at SDA)*
$\overline{A}$	*Not acknowledge bit (high level at SDA)*
Data	*8-bit data byte*
P	*Stop condition*

In Figures 37 to 40, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Tables 61 to 65.

15.2.14.1 Master Transmitter Mode:

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 37). Before the master transmitter mode can be entered, S1CON must be initialized as in Table 58.

CR0, CR1, and CR2 define the serial bit rate. ENS1 must be set to logic 1 to enable SIO1. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO0 cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO1 logic will now test the I^{2}C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (S1STA) will be 08H. This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA+W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 18H, 20H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 61. After a repeated start condition (state 10H). SIO1 may switch to the master receiver mode by loading S1DAT with SLA+R).

Table 58 Address Register S1CON (address D8H)

7	6	5	4	3	2	1	0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0
bit rate	1	0	0	0	X	bit rate	

15.2.14.2 Master Receiver Mode

In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 38). The transfer is initialized as in the master transmitter mode. When the start condition has been transmitted, the interrupt service routine must load S1DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in S1CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. These are 40H, 48H, or 38H for the master mode and also 68H, 78H, or B0H if the slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 62. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 62. After a repeated start condition (state 10H), SIO1 may switch to the master transmitter mode by loading S1DAT with SLA+W.

15.2.14.3 Slave Receiver Mode:

In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 39). To initiate the slave receiver mode, S1ADR and S1CON must be loaded as in Table 59.

The upper 7 bits are the address to which SIO1 will respond when addressed by a master. If the LSB (GC) is set, SIO1 will respond to the general call address (00H); otherwise it ignores the general call address.

CR0, CR1, and CR2 do not affect SIO1 in the slave mode. ENS1 must be set to logic 1 to enable SIO1. The AA bit must be set to enable SIO1 to acknowledge its own slave address or the general call address. STA, STO, and SI must be reset.

When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for SIO1 to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 63. The slave receiver mode may also be entered if arbitration is lost while SIO1 is in the master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^{2}C bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^{2}C bus.

Table 59 Address Register S1ADR (DBH) (address 00H)

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	X	X	X	X	X	GC
own slave address							

Table 60 Address Register S1CON (D8H) (address 00H)

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CR2	ENS1	STA	STO	SI	AA	CR1	CR0
X	1	0	0	0	1	X	X

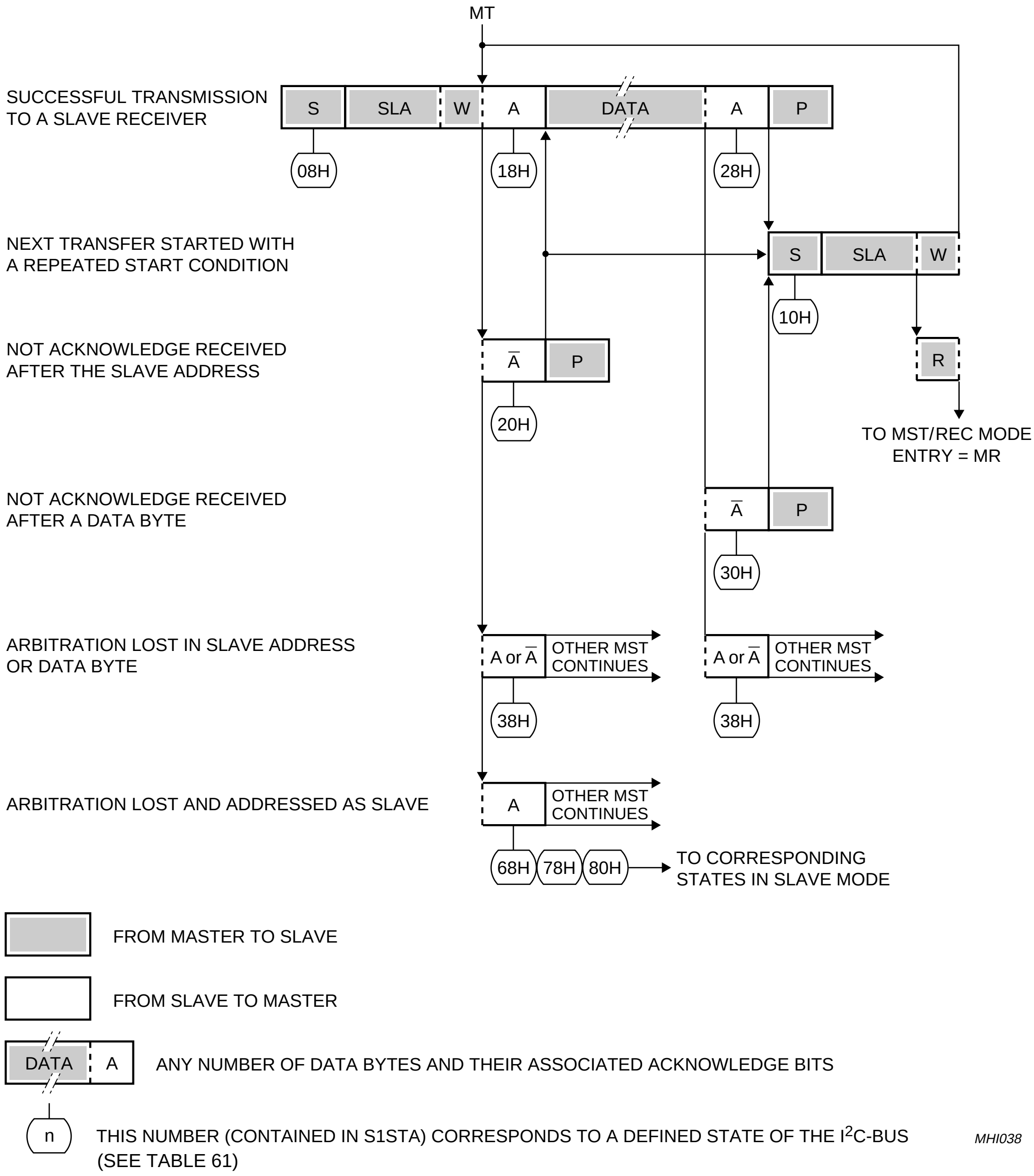

MT
SUCCESSFUL TRANSMISSION TO A SLAVE RECEIVER
S
SLA
W
A
DATA
A
P
08H
18H
28H
NEXT TRANSFER STARTED WITH A REPEATED START CONDITION
S
SLA
W
10H
NOT ACKNOWLEDGE RECEIVED AFTER THE SLAVE ADDRESS
A
P
20H
R
TO MST/REC MODE ENTRY = MR
NOT ACKNOWLEDGE RECEIVED AFTER A DATA BYTE
A
P
30H
ARBITRATION LOST IN SLAVE ADDRESS OR DATA BYTE
A or A
OTHER MST CONTINUES
38H
A or A
OTHER MST CONTINUES
38H
ARBITRATION LOST AND ADDRESSED AS SLAVE
A
OTHER MST CONTINUES
68H
78H
80H
TO CORRESPONDING STATES IN SLAVE MODE
FROM MASTER TO SLAVE
FROM SLAVE TO MASTER
DATA
A
ANY NUMBER OF DATA BYTES AND THEIR ASSOCIATED ACKNOWLEDGE BITS
n
THIS NUMBER (CONTAINED IN S1STA) CORRESPONDS TO A DEFINED STATE OF THE I²C-BUS (SEE TABLE 61)
MHI038

Fig.37 Format and States in the Master Transmitter Mode.

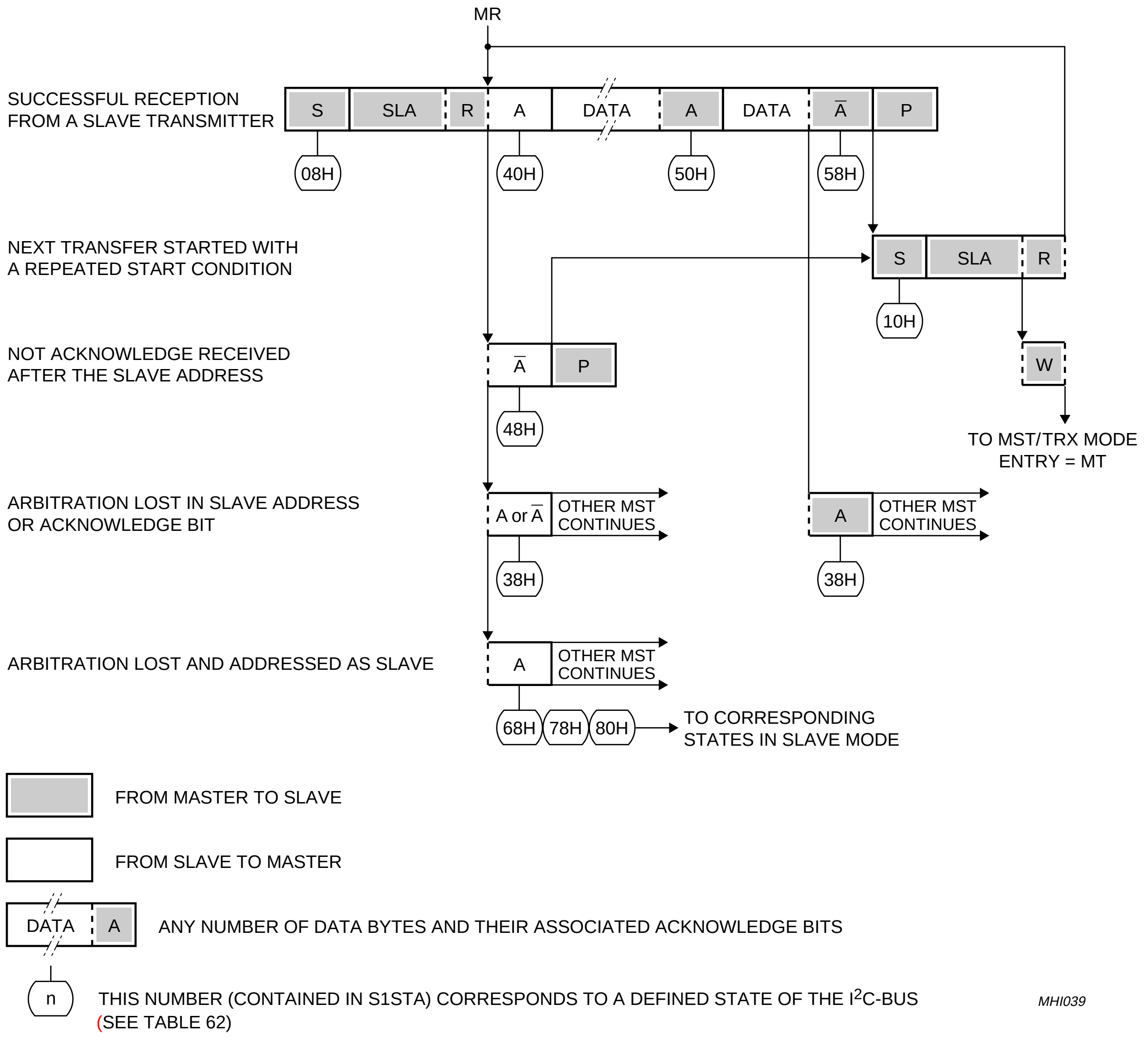

MR
SUCCESSFUL RECEPTION FROM A SLAVE TRANSMITTER
S
SLA
R
A
DATA
A
DATA
$\overline{A}$
P
08H
40H
50H
58H
NEXT TRANSFER STARTED WITH A REPEATED START CONDITION
S
SLA
R
10H
NOT ACKNOWLEDGE RECEIVED AFTER THE SLAVE ADDRESS
$\overline{A}$
P
48H
W
TO MST/TRX MODE ENTRY = MT
ARBITRATION LOST IN SLAVE ADDRESS OR ACKNOWLEDGE BIT
A or $\overline{A}$
OTHER MST CONTINUES
38H
A
OTHER MST CONTINUES
38H
ARBITRATION LOST AND ADDRESSED AS SLAVE
A
OTHER MST CONTINUES
68H
78H
80H
TO CORRESPONDING STATES IN SLAVE MODE
FROM MASTER TO SLAVE
FROM SLAVE TO MASTER
DATA
A
ANY NUMBER OF DATA BYTES AND THEIR ASSOCIATED ACKNOWLEDGE BITS
n
THIS NUMBER (CONTAINED IN S1STA) CORRESPONDS TO A DEFINED STATE OF THE I²C-BUS (SEE TABLE 62)
MHI039

Fig.38 Format and States in the Master Receiver Mode.

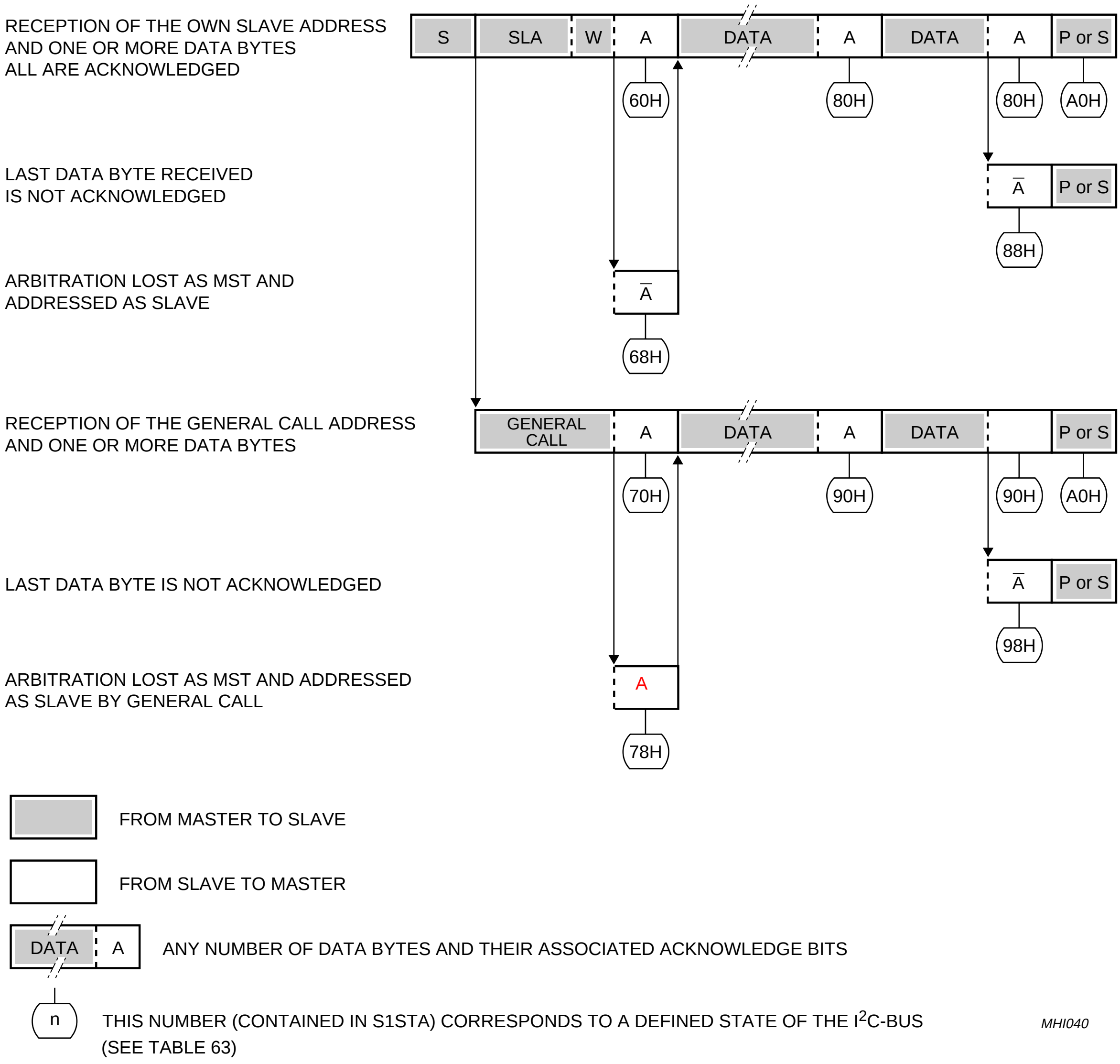

RECEPTION OF THE OWN SLAVE ADDRESS AND ONE OR MORE DATA BYTES ALL ARE ACKNOWLEDGED
S
SLA
W
A
DATA
A
DATA
A
P or S
60H
80H
80H
A0H
LAST DATA BYTE RECEIVED IS NOT ACKNOWLEDGED
A̅
P or S
88H
ARBITRATION LOST AS MST AND ADDRESSED AS SLAVE
A̅
68H
RECEPTION OF THE GENERAL CALL ADDRESS AND ONE OR MORE DATA BYTES
GENERAL CALL
A
DATA
A
DATA
P or S
70H
90H
90H
A0H
LAST DATA BYTE IS NOT ACKNOWLEDGED
A̅
P or S
98H
ARBITRATION LOST AS MST AND ADDRESSED AS SLAVE BY GENERAL CALL
A
78H
FROM MASTER TO SLAVE
FROM SLAVE TO MASTER
DATA
A
ANY NUMBER OF DATA BYTES AND THEIR ASSOCIATED ACKNOWLEDGE BITS
n
THIS NUMBER (CONTAINED IN S1STA) CORRESPONDS TO A DEFINED STATE OF THE I^{2}C-BUS (SEE TABLE 63)
MHI040

Fig.39 Format and States in the Slave Receiver Mode.

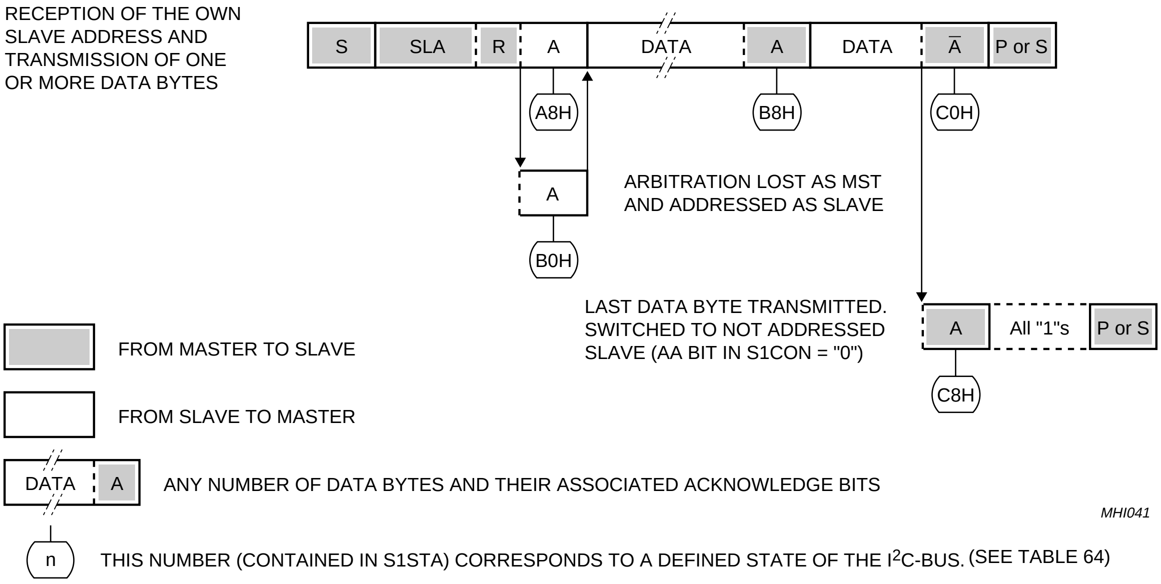

RECEPTION OF THE OWN SLAVE ADDRESS AND TRANSMISSION OF ONE OR MORE DATA BYTES
S
SLA
R
A
DATA
A
DATA
$\overline{A}$
P or S
A8H
B8H
C0H
A
ARBITRATION LOST AS MST AND ADDRESSED AS SLAVE
B0H
LAST DATA BYTE TRANSMITTED. SWITCHED TO NOT ADDRESSED SLAVE (AA BIT IN S1CON = "0")
A
All "1"s
P or S
C8H
FROM MASTER TO SLAVE
FROM SLAVE TO MASTER
DATA
A
ANY NUMBER OF DATA BYTES AND THEIR ASSOCIATED ACKNOWLEDGE BITS
MHI041
n
THIS NUMBER (CONTAINED IN S1STA) CORRESPONDS TO A DEFINED STATE OF THE I^{2}C-BUS. (SEE TABLE 64)

Fig.40 Format and States of the Slave Transmitter Mode.

Table 61 Master Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK bit will received
10H	A repeated START condition has been transmitted	Load SLA+W or	X	0	0	X	As above
		Load SLA+R	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received been received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
28H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in S1DAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R/$\overline{W}$ or Data bytes	No S1DAT action or	0	0	0	X	I^2C bus will be released; not addressed slave will be entered
		No S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

Table 62 Master Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+WR	X	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or	X	0	0	X	As above
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	no S1DAT action or	0	0	0	X	I^2C bus will be released; SIO1 will enter a slave mode
		no S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	no S1DAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		no S1DAT action	0	0	0	1	Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	no S1DAT action or	1	0	0	X	Repeated START condition will be transmitted
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
50H	Data byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned
58H	Data byte has been received; ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset

Table 63 Slave Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
60H	Own SLA+W has been received; ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
68H	Arbitration lost in SLA+R/W as master; Own SLA+W has been received, ACK returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
70H	General call address (00H) has been received; ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
78H	Arbitration lost in SLA+R/$\overline{W}$ as master; General call address has been received, ACK has been returned	No S1DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		no S1DAT action	X	0	0	1	Data byte will be received and ACK will be returned
80H	Previously addressed with own SLV address; DATA has been received; ACK has been returned	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		read data byte	X	0	0	1	Data byte will be received and ACK will be returned
88H	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
90H	Previously addressed with General Call; DATA byte has been received; ACK has been returned	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned
		read data byte	X	0	0	1	Data byte will be received and ACK will be returned

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
98H	Previously addressed with General Call; DATA byte has been received; NOT ACK has been returned	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No STDAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		No STDAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		No STDAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		No STDAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 64 Slave Transmitter Mode

STATUS CODE (S1STA)	STATUS OF THE I^{2}C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
B8H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
C0H	Data byte in S1DAT has been transmitted; NOT ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
C8H	Last data byte in S1DAT has been transmitted (AA = 0); ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 65 Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I²C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
F8H	No relevant state information available; SI = 0	No S1DAT action	No S1CON action				Wait or proceed current transfer
00H	Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.	No S1DAT action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the not addressed SLV mode. STO is reset.

15.2.14.4 Slave Transmitter Mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 40). Data transfer is initialized as in the slave receiver mode. When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be “1” (R) for SIO1 to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 64. The slave transmitter mode may also be entered if arbitration is lost while SIO1 is in the master mode (see state B0H).

If the AA bit is reset during a transfer, SIO1 will transmit the last byte of the transfer and enter state C0H or C8H. SIO1 is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^{2}C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^{2}C bus.

15.2.14.5 Miscellaneous States

There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 65). These are discussed below.

S1STA = F8H:
This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when SIO1 is not involved in a serial transfer.

S1STA = 00H:
This status code indicates that a bus error has occurred during an SIO1 serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal SIO1 signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes SIO1 to enter the not addressed slave mode (a defined state) and to clear the STO flag (no other bits in S1CON are affected). The SDA and SCL lines are released (a STOP condition is not transmitted).

15.2.15 SOME SPECIAL CASES

The SIO1 hardware has facilities to handle the following special cases that may occur during a serial transfer:

Simultaneous Repeated START Conditions from Two Masters.

A repeated START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 41). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data. If the SIO1 hardware detects a repeated START condition on the I^{2}C bus before generating a repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, SIO1 will transmit a normal START condition (state 08H), and a retry of the total serial data transfer can commence.

15.2.15.1 Data Transfer after loss of Arbitration

Arbitration may be lost in the master transmitter and master receiver modes (see Figure 33). Loss of arbitration is indicated by the following states in S1STA; 38H, 68H, 78H, and B0H (see Figures 37 and 38).

If the STA flag in S1CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08H) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

15.2.15.2 Forced Access to the I^{2}C bus

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I^{2}C bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I^{2}C bus is possible. This is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The SIO1 hardware behaves as if a STOP condition was received and is able to transmit a START condition. The STO flag is cleared by hardware (see Figure 42).

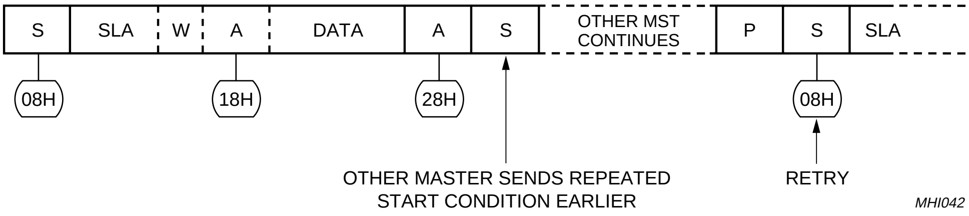


Fig.41 Simultaneous repeated START conditions from 2 Masters.

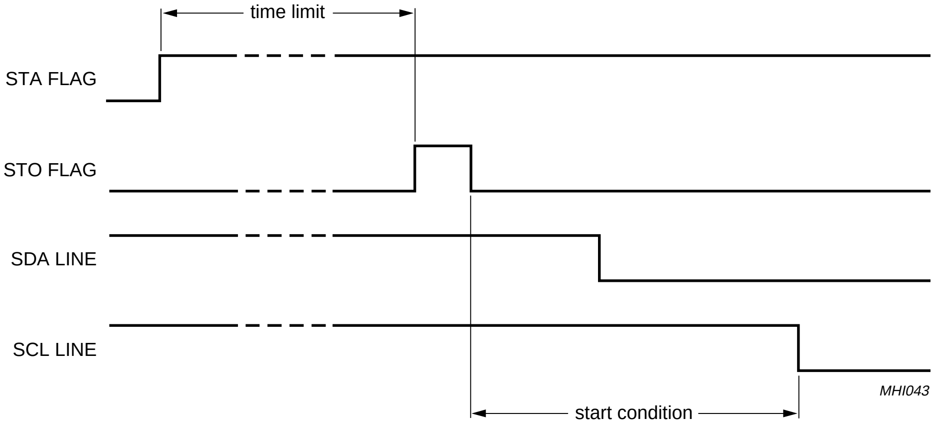


Fig.42 Forces access to a busy I^2C bus.

15.2.15.3 I²C bus obstructed by a low level on SCL and SDA

An I²C bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the SIO1 hardware cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 43). The SIO1 hardware transmits additional clock pulses when the STA flag is set, but no START condition can be generated because the SDA line is pulled LOW while the I²C bus is considered free.

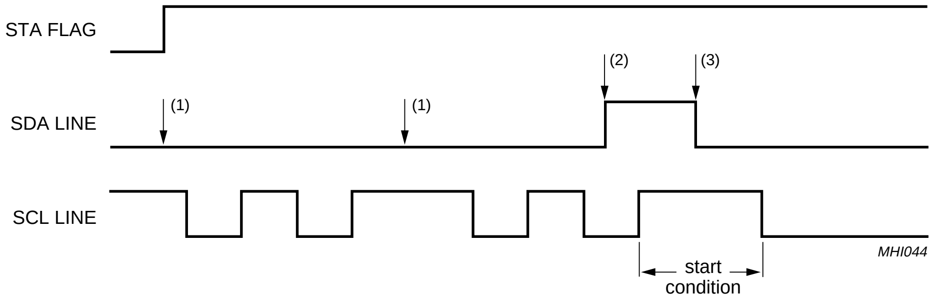
The SIO1 hardware attempts to generate a START condition after every two additional clock pulses on the SCL line. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transfer continues.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the SIO1 hardware performs the same action as described above. In each case, state 08H is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

15.2.15.4 Bus error

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data or an acknowledge bit.

The SIO1 hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, SIO1 immediately switches to the not addressed slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00H. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 65.



(1) Unsuccessful attempt to send a Start condition.
(2) SDA line released.
(3) Successful attempt to send a Start condition; state D8H is centered.

Fig.43 Recovering from a bus obstruction caused by a low level on SDA.

15.3 Software Examples of SIO1 Service Routines

This section consists of a software example for:

- Initialization of SIO1 after a RESET
- Entering the SIO1 interrupt routine
- The 26 state service routines for the
  - Master transmitter mode
  - Master receiver mode
  - Slave receiver mode
  - Slave transmitter mode

15.3.1 Initialization

In the initialization routine, SIO1 is enabled for both master and slave modes. For each mode, a number of bytes of internal data RAM are allocated to the SIO to act as either a transmission or reception buffer. In this example, 8 bytes of internal data RAM are reserved for different purposes. The data memory map is shown in Figure 44. The initialization routine performs the following functions:

- S1ADR is loaded with the parts own slave address and the general call bit (GC)
- P1.6 and P1.7 bit latches are loaded with logic 1s
- RAM location HADD is loaded with the high-order address byte of the service routines
- The SIO1 interrupt enable and interrupt priority bits are set
- The slave mode is enabled by simultaneously setting the ENS1 and AA bits in S1CON and the serial clock frequency (for master modes) is defined by loading CR0 and CR1 in S1CON. The master routines must be started in the main program.

The SIO1 hardware now begins checking the I^{2}C bus for its own slave address and general call. If the general call or the own slave address is detected, an interrupt is requested and S1STA is loaded with the appropriate state information. The following text describes a fast method of branching to the appropriate service routine.

15.3.2 SIO1 Interrupt Routine

When the SIO1 interrupt is entered, the PSW is first pushed on the stack. Then S1STA and HADD (loaded with the high-order address byte of the 26 service routines by the initialization routine) are pushed on to the stack. S1STA contains a status code which is the lower byte of one of the 26 service routines. The next instruction is RET, which is the return from subroutine instruction. When this instruction is executed, the high and low order address bytes are popped from stack and loaded into the program counter.

The next instruction to be executed is the first instruction of the state service routine. Seven bytes of program code (which execute in eight machine cycles) are required to branch to one of the 26 state service routines.

```
SI  PUSH  PSW     Save PSW
    PUSH  S1STA   Push status code (low order
                  address byte)
    PUSH  HADD    Push high order address byte
    RET           Jump to state service routine
```

The state service routines are located in a 256-byte page of program memory. The location of this page is defined in the initialization routine. The page can be located anywhere in program memory by loading data RAM register HADD with the page number. Page 01 is chosen in this example, and the service routines are located between addresses 0100H and 01FFH.

15.3.3 The State Service Routine

The state service routines are located 8 bytes from each other. Eight bytes of code are sufficient for most of the service routines. A few of the routines require more than 8 bytes and have to jump to other locations to obtain more bytes of code. Each state routine is part of the SIO1 interrupt routine and handles one of the 26 states. It ends with a RETI instruction which causes a return to the main program.

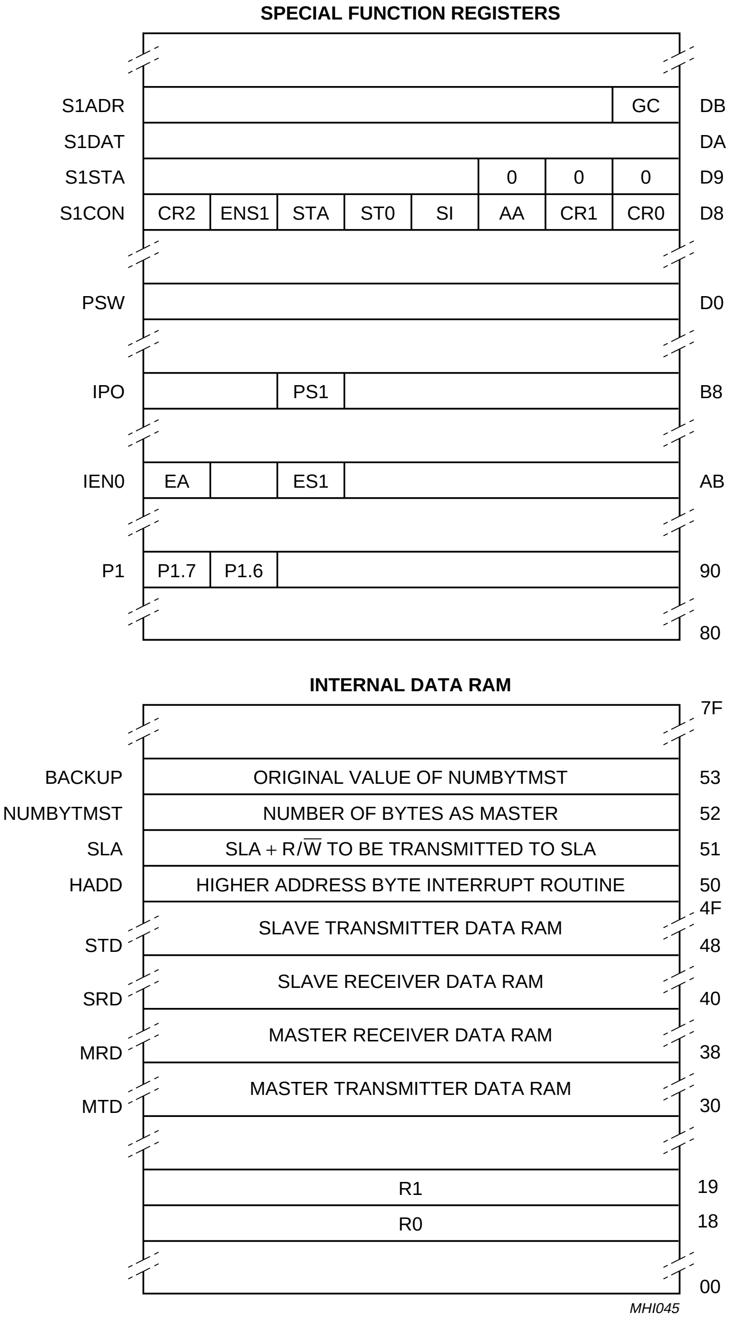


Fig.44 SIO1 Data Memory Map.

15.3.4 MASTER TRANSMITTER AND MASTER RECEIVER MODES

The master mode is entered in the main program. To enter the master transmitter mode, the main program must first load the internal data RAM with the slave address, data bytes, and the number of data bytes to be transmitted. To enter the master receiver mode, the main program must first load the internal data RAM with the slave address and the number of data bytes to be received. The R/W bit determines whether SIO1 operates in the master transmitter or master receiver mode.

Master mode operation commences when the STA bit in S1CION is set by the SETB instruction and data transfer is controlled by the master state service routines in accordance with Table 61, Table 62, Figure 37 and Figure 38. In the example below, 4 bytes are transferred. There is no repeated START condition. In the event of lost arbitration, the transfer is restarted when the bus becomes free. If a bus error occurs, the I^{2}C bus is released and SIO1 enters the not selected slave receiver mode. If a slave device returns a not acknowledge, a STOP condition is generated.

A repeated START condition can be included in the serial transfer if the STA flag is set instead of the STO flag in the state service routines vectored to by status codes 28H and 58H. Additional software must be written to determine which data is transferred after a repeated START condition.

15.3.5 SLAVE TRANSMITTER AND SLAVE RECEIVER MODES

After initialization, SIO1 continually tests the I^{2}C bus and branches to one of the slave state service routines if it detects its own slave address or the general call address (see Table 63, Table 64, Figure 39, and Figure 40). If arbitration was lost while in the master mode, the master mode is restarted after the current transfer. If a bus error occurs, the I^{2}C bus is released and SIO1 enters the not selected slave receiver mode.

In the slave receiver mode, a maximum of 8 received data bytes can be stored in the internal data RAM. A maximum of 8 bytes ensures that other RAM locations are not overwritten if a master sends more bytes. If more than 8 bytes are transmitted, a not acknowledge is returned, and SIO1 enters the not addressed slave receiver mode. A maximum of one received data byte can be stored in the internal data RAM after a general call address is detected. If more than one byte is transmitted, a not acknowledge is returned and SIO1 enters the not addressed slave receiver mode.

In the slave transmitter mode, data to be transmitted is obtained from the same locations in the internal data RAM that were previously loaded by the main program. After a not acknowledge has been returned by a master receiver device, SIO1 enters the not addressed slave mode.

15.3.6 ADAPTING THE SOFTWARE FOR DIFFERENT APPLICATIONS

The following software example shows the typical structure of the interrupt routine including the 26 state service routines and may be used as a base for user applications. If one or more of the four modes are not used, the associated state service routines may be removed but, care should be taken that a deleted routine can never be invoked.

This example does not include any time-out routines. In the slave modes, time-out routines are not very useful since, in these modes, SIO1 behaves essentially as a passive device. In the master modes, an internal timer may be used to cause a time-out if a serial transfer is not complete after a defined period of time. This time period is defined by the system connected to the I^{2}C bus.

SI01 EQUATE LIST

```
LOC     OBJ     SOURCE
                !*********************************************************************************************************************************
                ! LOCATIONS OF THE SI01 SPECIAL FUNCTION REGISTERS!
                !*********************************************************************************************************************************

00D8            S1CON       -0xd8
00D9            S1STA       -0xd9
00DA            S1DAT       -0xda
00DB            S1ADR       -0xdb

00A8            IEN0        -0xa8
00B8            IP0         02b8
                !*********************************************************************************************************************************
                ! BIT LOCATIONS
                !*********************************************************************************************************************************

00DD            STA         -0xdd                              ! STA bit in S1CON
00BD            SI01HP      -0xbd                              ! IP0, SI01 Priority bit
                !*********************************************************************************************************************************
                ! IMMEDIATE DATA TO WRITE INTO REGISTER S1CON
                !*********************************************************************************************************************************

00D5            ENS1_NOTSTA_STO_NOTSI_AA_CR0                   -0xd5     ! Generates STOP
                                                                         ! (CR0 = 100kHz @ fOSC =
                                                                         !  6 MHz)
00C5            ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0                -0xc5     ! Releases BUS and ACK
                                                                         !
00C1            ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0             -0xc1     ! Releases BUS and
                                                                         ! NOT ACK
00E5            ENS1_STA_NOTSTO_NOTSI_AA_CR0                   -0xe5     ! Releases BUS and set
                                                                         ! STA
                !*********************************************************************************************************************************
                ! GENERAL IMMEDIATE DATA
                !*********************************************************************************************************************************

0031            OWNSLA      -0x31                              ! Own SLA+General Call
                                                               ! must be written into S1ADR
00A0            ENSI01      -0xa0                              ! EA+ES1, enable SIO1 interrupt
                                                               ! must be written into IEN0
0001            PAG1        -0x01                              ! select PAG1 as HADD
00C0            SLAW        -0xc0                              ! SLA+W to be transmitted
00C1            SLAR        -0xc1                              ! SLA+R to be transmitted
0018            SELRB3      -0x18                              ! Select Register Bank 3
```

```
LOC     OBJ       SOURCE
                  !******************************************************************************************
                  ! LOCATIONS IN DATA RAM
                  !******************************************************************************************
0030              MTD           -0x30                                  ! MST/TRX/DATA base address
0038              MRD           -0x38                                  ! MST/REC/DATA base address
0040              SRD           -0x40                                  ! SLV/REC/DATA base address
0048              STD           -0x48                                  ! SLV/TRX/DATA base address

0053              BACKUP            -0x53                              ! Backup from NUMBYTMST
                                                                       ! To restore NUMBYTMST in case
                                                                       ! of an Arbitration Loss.
0052              NUMBYTMST         -0x52                              ! Number of bytes to transmit
                                                                       ! or receive as MST.
0051              SLA               -0x51                              ! Contains SLA+R/W to be
                                                                       ! transmitted.
0050              HADD              -0x50                              ! High Address byte for STATE 0f
                                                                       ! till STATE 25.
                  !******************************************************************************************
                  ! INITIALIZATION ROUTINE
                  ! Example to initialize IIC Interface as slave receiver or slave transmitter and start a MASTER TRANSMIT
                  ! or a MASTER RECEIVE function. 4 bytes will be transmitted or received.
                  !******************************************************************************************
                  .sect    strt
                  .base    0x00
0000    4100                    ajmp       INIT                        ! RESET
                  .sect    initial
                  .base    0x200
0200    75DB31    INIT:         mov        S1ADR,#OWNSLA               ! Load own SLA + enable
                                                                       ! general call recognition
0203    D296                    setb       P1(6)                       ! P1.6 High level.
0205    D297                    setb       P1(7)                       ! P1.7 High level.
0207    755001                  mov        HADD,#PAG1
020A    43A8A0                  orl        IEN0,#ENSI01                ! Enable SI01 interrupt
020D    C2BD                    clr        SI01HP                      ! SI01 interrupt low priority
020F    75D8C5                  mov        S1CON, #ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                       ! Initialize SLV funct.
                  !******************************************************************************************
                  ! START MASTER TRANSMIT FUNCTION
                  !******************************************************************************************
0212    755204                  mov        NUMBYTMST,#0x4              ! Transmit 4 bytes.
0215    7551C0                  mov        SLA,#SLAW                   ! SLA+W, Transmit funct.
0218    D2DD                    setb       STA                         ! set STA in S1CON!
                  !******************************************************************************************
                  ! START MASTER RECEIVE FUNCTION
                  !******************************************************************************************
021A    755204                  mov        NUMBYTMST,#0x4              ! Receive 4 bytes.
021D    7551C1                  mov        SLA,#SLAR                   ! SLA+R, Receive funct.
0220    D2DD                    setb       STA                         ! set STA in S1CON
```

```
LOC     OBJ         SOURCE
                    !*********************************************************************************************************************
                    ! SI01 INTERRUPT ROUTINE
                    !*********************************************************************************************************************
                    .sect      intvec                                       ! SI01 interrupt vector
                    .base      0x00

                    ! S1STA and HADD are pushed onto the stack.
                    ! They serve as return address for the RET instruction.
                    ! The RET instruction sets the Program Counter to address HADD,
                    ! S1STA and jumps to the right subroutine.

002B    C0D0                  push        psw                               ! save psw
002D    C0D9                  push        S1STA
002F    C050                  push        HADD
0031    22                    ret                                           ! JMP to address HADD,S1STA.
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    ! STATE   : 00, Bus error.
                    ! ACTION : Enter not addressed SLV mode and release bus. STO reset.
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    .sect       st0
                    .base      0x100
0100    75D8D5                mov         S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0           ! clr SI
                                                                                        ! set STO,AA
0103    D0D0                  pop         psw
0105    32                    reti
                    !*********************************************************************************************************************
                    ! MASTER STATE SERVICE ROUTINES
                    !*********************************************************************************************************************

                    !*********************************************************************************************************************
                    ! State 08 and State 10 are both for MST/TRX and MST/REC.
                    ! The R/W bit decides whether the next state is within
                    ! MST/TRX mode or within MST/REC mode.
                    !*********************************************************************************************************************

                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    ! STATE   : 08, A, START condition has been transmitted.
                    ! ACTION : SLA+R/W are transmitted, ACK bit is received.!
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    .sect      mts8
                    .base       0x108
0108    8551DA                mov         S1DAT,SLA                         ! Load SLA+R/W
010B    75D8C5                mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                            ! clr SI
010E    01A0                  ajmp        INITBASE1
```

```
LOC     OBJ         SOURCE

                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    ! STATE   : STATE : 10, A repeated START condition has been transmitted.
                    ! ACTION : SLA+R/W are transmitted, ACK bit is received.!
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    .sect     mts10
                    .base     0x110
0110    8551DA                          mov       S1DAT,SLA                       ! Load SLA+R/W
0113    75D8C5                          mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                                  ! clr SI
010E     01A0                           ajmp      INITBASE1
                    .sect     ibase1
                    .base     0xa0
00A0    75D018      INITBASE1:          mov       psw,#SELRB3
00A3    7930                            mov        r1,#MTD
00A5    7838                            mov        r0,#MRD
00A7    855253                          mov       BACKUP,NUMBYTMST                ! Save initial value
00AA    D0D0                            pop       psw
00AC    32                              reti
                    !************************************************************************************************************************
                    ! MASTER TRANSMITTER STATE SERVICE ROUTINES
                    !************************************************************************************************************************

                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    ! STATE   : 18, Previous state was STATE 8 or STATE 10, SLA+W have been transmitted,  ACK been received. !
                    ACTION : First DATA is transmitted, ACK bit is received.
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -dc
                    .sect     mts18
                    .base     0x118
0118    75D018                          mov       psw,#SELRB3
011B    87DA                            mov       S1DAT,@r1
011D    01B5                            ajmp      CON
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    ! STATE   : 20, SLA+W have been transmitted, NOT ACK has been received
                    ! ACTION : Transmit STOP condition.!
                    !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                    .sect     mts20
                    .base      0x120
0120    75D8D5                          mov       S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                                  ! set STO, clr SI
0123    D0D0                            pop       psw
0125    32                              reti
```

```
LOC     OBJ        SOURCE
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   : 28, DATA of S1DAT have been transmitted, ACK received.
                   ! ACTION : If Transmitted DATA is last DATA then transmit a STOP condition, else transmit next DATA.
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect     mts28
                   .base     0x128
0128    D55285                       djnz        NUMBYTMST,NOTLDAT1          ! JMP if NOT last DATA
012B    75D8D5                       mov         S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                             ! clr SI, set AA
012E    01B9                         ajmp        RETmt
                   .sect     mts28sb
                   .base     0x0b0
00B0    75D018      NOTLDAT1:        mov         psw,#SELRB3
00B3    87DA                         mov         S1DAT,@r1
00B5    75D8C5     CON:              mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                             ! clr SI, set AA
00B8    09                           inc         r1
00B9    D0D0       RETmt :           pop         psw
00BB    32                           reti
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   : 30, DATA of S1DAT have been transmitted, NOT ACK received.
                   ! ACTION : Transmit a STOP condition.
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect     mts30
                   .base     0x130
0130    75D8D5                       mov         S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                             ! set STO, clr SI
0133    D0D0                         pop         psw
0135    32                           reti!
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   : 38, Arbitration lost in SLA+W or DATA.
                   ! ACTION : Bus is released, not addressed SLV mode is entered. A new START condition is
                   !                transmitted when the IIC bus is free again.!
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect     mts38
                   .base     0x138
0138    75D8E5                       mov         S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
013B    855352                       mov         NUMBYTMST,BACKUP
013E    01B9                         ajmp        RETmt
                   !********************************************************************************************************
                   ! MASTER RECEIVER STATE SERVICE ROUTINES
                   !********************************************************************************************************
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   : 40, Previous state was STATE 08 or STATE 10, SLA+R have been transmitted, ACK received.
                   ! ACTION : DATA will be received, ACK returned.
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect     mts40
                   .base     0x140
0140    75D8C5                       mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                             ! clr STA, STO, SI set AA
0143    D0D0                         pop         psw
        32                           reti
```

```
LOC    OBJ       SOURCE
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 ! STATE   : 48, SLA+R have been transmitted, NOT ACK received.
                 ! ACTION : STOP condition will be generated.
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 .sect    mts48
                 .base    0x148
0148   75D8D5    STOP:              mov          S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                         ! set STO, clr SI
014B   D0D0                         pop          psw
014D    32                          reti
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 ! STATE   : 50, DATA have been received, ACK returned.
                 ! ACTION : Read DATA of S1DAT. DATA will be received, if it is last DATA then NOT ACK will be returned
                 !              else ACK will be returned.
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 .sect    mrs50
                 .base    0x150
0150   75D018                       mov          psw,#SELRB3
0153   A6DA                         mov          @r0,S1DAT                   ! Read received DATA
0155   01C0                         ajmp         REC1
                 .sect    mrs50s
                 .base    0xc0

00C0   D55205    REC1:              djnz         NUMBYTMST,NOTLDAT2
00C3   75D8C1                       mov           S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                                         ! clr SI,AA
00C6   8003                         sjmp         RETmr
00C8   75D8C5    NOTLDAT2:          mov           S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                         ! clr SI, set AA
00CB   08        RETmr:             inc          r0
00CC   D0D0                         pop          psw
00CE   32                           reti
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 ! STATE   : 58, DATA have been received, NOT ACK returned.
                 ! ACTION : Read DATA of MASTER STATE SERVICE ROUTINESS1DAT and generate a STOP condition.
                 !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                 .sect    mrs58
                 .base    0x158
0158   75D018                       mov          psw,#SELRB3
015B   A6DA                         mov          @R0,S1DAT
015D   80E9                         sjmp         STOP
```

```
LOC     OBJ       SOURCE
                  !*********************************************************************************************************************************
                  ! SLAVE RECEIVER STATE SERVICE ROUTINES
                  !*********************************************************************************************************************************

                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 60, Own SLA+W have been received, ACK returned.
                  ! ACTION : DATA will be recMASTER STATE SERVICE ROUTINESeived and ACK returned.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs60
                  .base     0x160
0160    75D8C5                         mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                             ! clr SI, set AA
0163    75D018                         mov         psw,#SELRB3
0166    01D0                           ajmp        INITSRD
                  .sect     insrd
                  .base     0xd0

00D0    7840      INITSRD:             mov         r0,#SRD
00D2    7908                           mov         r1,#8
00D4    D0D0                           pop         psw
00D6    32                             reti
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 68, Arbitration lost in SLA and R/W as MST Own SLA+W have been received, ACK returned
                  ! ACTION : DATA will be received and ACK returned. STA is set to restart MST mode after the bus is free again.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs68
                  .base     0x168
0168    75D8E5                         mov         S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
016B    75D018                         mov         psw,#SELRB3
016E    01D0                           ajmp        INITSRD
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 70, General call has been received, ACK returned.
                  ! ACTION : DATA will be received and ACK returned.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs70
                  .base     0x170
0170    75D8C5                         mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                             ! clr SI, set AA
0173    75D018                         mov         psw,#SELRB3               ! Initialize SRD counter
0176    01D0                           ajmp        initsrd
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 78, Arbitration lost in SLA+R/W as MST. General call has been received, ACK returned.
                  ! ACTION : DATA will be received and ACK returned. STA is set to restart MST mode after the bus is free again.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs78
                  .base     0x178
0178    75D8E5                         mov         S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
017B    75D018                         mov         psw,#SELRB3               ! Initialize SRD counter
017E    01D0                           ajmp        INITSRD
```

```
LOC     OBJ       SOURCE
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 80, Previously addressed with own SLA. DATA received, ACK returned.
                  ! ACTION : Read DATA.
                  !            IF received DATA was the last
                  !            THEN superfluous DATA will be received and NOT ACK returned
                  !            ELSE next DATA will be received and ACK returned.!
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs80
                  .base     0x180
0180    75D018                        mov       psw,#SELRB3
0183    A6DA                          mov       @r0,S1DAT                      ! Read received DATA
0185    01D8                          ajmp      REC2

                  .sect     srs80s
                  .base      0xd8

00D8    D906      REC2:               djnz       r1,NOTLDAT3
00DA    75D8C1    LDAT:               mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_NOTAA_CR0
                                                                               ! clr SI,AA
00DD    D0D0                          pop       psw
00DF    32                            reti
00E0    75D8C5    NOTLDAT3:           mov        S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                               ! clr SI, set AA
00E3    08                            inc        r0
00E4    D0D0      RETsr:              pop       psw
00E6    32                            reti
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 88, Previously addressed with own SLA. DATA received NOT ACK returned.
                  ! ACTION : No save of DATA, Enter NOT addressed SLV mode.
                  !            Recognition of own SLA. General call recognized, if S1ADR. 01.!
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs88
                  .base     0x188
0188    75D8C5                         mov       S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                               ! clr SI, set AA
018B    01E4                          ajmp      RETsr
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  ! STATE   : 90, Previously addressed with general call. DATA has been received, ACK has been returned.
                  ! ACTION : Read DATA.
                  !            After General call only one byte will be received with ACK the second DATA
                  !            will be received with NOT ACK.
                  !            DATA will be received and NOT ACK returned.
                  !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                  .sect     srs90
                  .base     0x190
0190    76D018                        mov       psw,#SELRB3
0193    A6DA                          mov       @r0,S1DAT                      ! Read received DATA
0195    01DA                          ajmp      LDAT
```

```
LOC     OBJ        SOURCE
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   :  98, Previously addressed with general call.
                   !            DATA has been received, NOT ACK has been returned.
                   ! ACTION : No save of DATA, Enter NOT addressed SLV mode.
                   !            Recognition of own SLA. General call recognized, if S1ADR. 01.!
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect      srs98
                   .base      0x198
0198    75D8C5                         mov        S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                           ! clr SI, set AA
019B    D0D0                           pop        psw
019D    32                             reti
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   :  A0, A STOP condition or repeated START has been received, while still addressed as
                   !            SLV/REC or SLV/TRX.
                   ! ACTION : No save of DATA, Enter NOT addressed SLV mode.
                   !            Recognition of own SLA. General call recognized, if S1ADR. 01.
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect      srsA0
                   .base      0x1a0
01A0    75D8C5                         mov        S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                           ! clr SI, set AA
01A3    D0D0                           pop        psw
01A5    32                             reti
                   !*********************************************************************************************
                   ! SLAVE TRANSMITTER STATE SERVICE ROUTINES
                   !*********************************************************************************************
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   ! STATE   : A8, Own SLA+R received, ACK returned.
                   ! ACTION : DATA will be transmitted, A bit received.
                   !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                   .sect      stsa8
                   .base       0x1a8
01A8    8548DA                         mov        S1DAT,STD                    ! load DATA in S1DAT
01AB    75D8C5                         mov         S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                           ! clr SI, set AA
01AE    01E8                           ajmp       INITBASE2
                   .sect      ibase2
                   .base       0xe8
00E8    75D018     INITBASE2:          mov        psw,#SELRB3
00EB    7948                           mov        r1, #STD
00ED    09                             inc        r1
00EE    D0D0                           pop        psw
00F0    32                             reti
```

```
LOC       OBJ           SOURCE
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        ! STATE   :  B0, Arbitration lost in SLA and R/W as MST. Own SLA+R received, ACK returned.
                        ! ACTION : DATA will be transmitted, A bit received.
                        !              STA is set to restart MST mode after the bus is free again.
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        .sect     sstsb0
                        .base     0x1b0
01B0      8548DA                            mov            S1DAT,STD                        ! load DATA in S1DAT
01B3      75D8E5                            mov            S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
01B6      01E8                              ajmp           INITBASE2
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        ! STATE   : B8, DATA has been transmitted, ACK received.
                        ! ACTION : DATA will be transmitted, ACK bit is received.
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        .sect     stsb8
                        .base     0x1b8
01B8      75D018                            mov            psw,#SELRB3
01BB      87DA                              mov            S1DAT,@r1
01BD      01F8                              ajmp           SCON
                        .sect     scn
                        .base     0xf8
00F8      75D8C5        SCON:               mov            S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                                            ! clr SI, set AA
00FB      09                                inc             r1
00FC      D0D0                              pop             psw
00FE      32                                reti
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        ! STATE   : C0, DATA has been transmitted, NOT ACK received.
                        ! ACTION : Enter not addressed SLV mode.
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        .sect     stsc0
                        .base     0x1c0
01C0      75D8C5                            mov            S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                                            ! clr SI, set AA
01C3      D0D0                              pop            psw
01C5      32                                reti
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        ! STATE   : C8, Last DATA has been transmitted (AA=0), ACK received.
                        ! ACTION : Enter not addressed SLV mode
                        !- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                        .sect     stsc8
                        .base     0x1c8
01C8      75D8C5                            mov            S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                                            ! clr SI, set AA
01CB      D0D0                              pop            psw
01CD      32                                reti
                        !********************************************************************************************************************
                        ! END OF SI01 INTERRUPT ROUTINE
                        !********************************************************************************************************************
```

16 TIMER 2

16.1 Features of Timer 2

Timer T2 is a 16-bit timer consisting of two registers TMH2 (HIGH byte) and TML2 (LOW byte). The 16-bit timer/counter can be switched off or clocked via a prescaler from one of two sources: $f_{CLK}/6$ or an external signal. When Timer T2 is configured as a counter, the prescaler is clocked by an external signal on T2 (P3.O). A rising edge on T2 increments the prescaler, and the maximum repetition rate is one count per machine cycle (1 MHz with a 6 MHz oscillator).

The maximum repetition rate for Timer T2 is twice the maximum repetition rate for Timer 0 and Timer 1. T2 (P3.0) is sampled at S2P1 and again at S5P1 (i.e., twice per machine cycle). A rising edge is detected when T2 is LOW during one sample and HIGH during the next sample. To ensure that a rising edge is detected, the input signal must be LOW for at least ½ cycle and then HIGH for at least ½ cycle. If a rising edge is detected before the end of S2P1, the timer will be incremented during the following cycle; otherwise it will be incremented one cycle later. The prescaler has a programmable division factor of 1, 2, 4, or 8 and is cleared if its division factor or input source is changed, or if the timer/counter is reset.

Timer T2 may be read "on the fly" but possesses no extra read latches, and software precautions may have to be taken to avoid misinterpretation in the event of an overflow from least to most significant byte while Timer T2 is being read. Timer T2 is not loadable and is reset by the $\overline{RST}$ signal or by a rising edge on the input signal RT2, if enabled. RT2 is enabled by setting bit T2ER (TM2CON.5).

When the least significant byte of the timer overflows or when a 16-bit overflow occurs, an interrupt request may be generated. Either or both of these overflows can be programmed to request an interrupt. In both cases, the interrupt vector will be the same. When the lower byte (TML2) overflows, flag T2B0 (TM2CON) is set and flag T20V (TM2IR) is set when TMH2 overflows. These flags are set one cycle after an overflow occurs. Note that when T20V is set, T2B0 will also be set. To enable the byte overflow interrupt, bits ET2 (IEN1.7, enable overflow interrupt, see Table 67) and T2IS0 (TM2CON.6, byte overflow interrupt select) must be set. Bit TWBO (TM2CON.4) is the Timer T2 byte overflow flag. To enable the 16-bit overflow interrupt, bits ET2 (IE1.7, enable overflow interrupt) and T2IS1 (TM2CON.7, 16-bit overflow interrupt select) must be set. Bit T2OV (TM2IR.7) is the Timer T2 16-bit overflow flag. All interrupt flags must be reset by software. To enable both byte and 16-bit overflow, T2IS0 and T2IS1 must be set and two interrupt service routines are required. A test on the overflow flags indicates which routine must be executed. For each routine, only the corresponding overflow flag must be cleared. Timer T2 may be reset by a rising edge on RT2 (P3.1) if the Timer T2 external reset enable bit (T2ER) in TM2CON is set. This reset also clears the prescaler. In the Idle mode, the timer/counter and prescaler are reset and halted. Timer T2 is controlled by the TM2CON special function register (see Section 16.1.1).

Table 66 Timer T2 Interrupt Enable Register IEN1 (address E8H)

7	6	5	4	3	2	1	0
ET2	ECAN	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

Table 67 Description of interrupt Enable Register IEN1 bits

BIT	SYMBOL	FUNCTION
7	ET2	Enable Timer T2 overflow interrupt(s).
6	ECAN	Enable CAN interrupt.
5	ECM1	Enable T2 Comparator 1 interrupt.
4	ECM0	Enable T2 Comparator 0 interrupt.
3	ECT3	Enable T2 Capture register 3 interrupt.
2	ECT2	Enable T2 Capture register 2 interrupt.
1	ECT1	Enable T2 Capture register 1 interrupt.
0	ECT0	Enable T2 Capture register 0 interrupt.