Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	16MHz
Connectivity	EBI/EMI, I²C, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	40
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.18x24.18)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p87c554sbaa-512

80C51 8-bit microcontroller – 12 clock operation 16K/512 OTP/RAM, 8 channel 10-bit A/D, I²C, PWM, capture/compare, high I/O

P87C554

DESCRIPTION

The P87C554 Single-Chip 8-Bit Microcontroller is manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The 87C554 has the same instruction set as the 80C51.

The 87C554 contains a 16k × 8 non-volatile EPROM, a 512 × 8 read/write data memory, five 8-bit I/O ports, one 8-bit input port, two 16-bit timer/event counters (identical to the timers of the 80C51), an additional 16-bit timer coupled to capture and compare latches, a 15-source, four-priority-level, nested interrupt structure, an 8-input ADC, a dual DAC pulse width modulated interface, two serial interfaces (UART and I²C-bus), a "watchdog" timer and on-chip oscillator and timing circuits. For systems that require extra capability, the P87C554 can be expanded using standard TTL compatible memories and logic.

In addition, the P87C554 has two software selectable modes of power reduction—idle mode and power-down mode. The idle mode freezes the CPU while allowing the RAM, timers, serial ports, and interrupt system to continue functioning. Optionally, the ADC can be operated in Idle mode. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

The device also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 45 two-byte, and 17 three-byte. With a 16 MHz crystal, 58% of the instructions are executed in 0.75 μs and 40% in 1.5 μs. Multiply and divide instructions require 3 μs.

FEATURES

- 80C51 central processing unit
- 16k × 8 EPROM expandable externally to 64k bytes
- An additional 16-bit timer/counter coupled to four capture registers and three compare registers
- Two standard 16-bit timer/counters
- 512 × 8 RAM, expandable externally to 64k bytes
- Capable of producing eight synchronized, timed outputs
- A 10-bit ADC with eight multiplexed analog inputs
- Fast 8-bit ADC option
- Two 8-bit resolution, pulse width modulation outputs
- Five 8-bit I/O ports plus one 8-bit input port shared with analog inputs
- I²C-bus serial I/O port with byte oriented master and slave functions
- On-chip watchdog timer
- Extended temperature ranges
- Full static operation – 0 to 16 MHz
- Operating voltage range: 2.7 V to 5.5 V (0 to 16 MHz) and 4.5 V to 5.5 V (16 to 33 MHz)
- Three security bits
- Encryption array – 64 bytes
- 4 level priority interrupt
- 15 interrupt sources
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Second DPTR register
- ALE inhibit for EMI reduction
- Programmable I/O pins
- Wake-up from power-down by external interrupts
- Software reset
- Power-on detect reset
- ADC charge pump disable
- ONCE mode
- ADC active in Idle mode

Table 2. External Pin Status During Idle and Power-Down Modes

MODE	PROGRAM MEMORY	ALE	$\overline{PSEN}$	PORT 0	PORT 1	PORT 2	PORT 3	PORT 4	PWM0/ PWM1
Idle	Internal	1	1	Data	Data	Data	Data	Data	High
Idle	External	1	1	Float	Data	Address	Data	Data	High
Power-down	Internal	0	0	Data	Data	Data	Data	Data	High
Power-down	External	0	0	Float	Data	Data	Data	Data	High

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

POWER OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the V_{CC} level on the P87C554 rises from 0 to 5 V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The V_{CC} level must remain above 3 V for the POF to remain unaffected by the V_{CC} level.

Design Consideration

- When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems without the device having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Reduced EMI Mode

The ALE-Off bit, AO (AUXR.0) can be set to disable the ALE output. It will automatically become active when required for external memory accesses and resume to the OFF state after completing the external memory access.

	7	6	5	4	3	2	1	0
PCON (87H)	SMOD1	SMOD0	POF	WLE	GF1	GF0	PD	IDL
	(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
PCON.7	SMOD1	Double Baud rate bit. When set to logic 1, the baud rate is doubled when the serial port SIO0 is being used in modes 1, 2, or 3.
PCON.6	SMOD0	Selects SM0/FE for SCON.7 bit.
PCON.5	POF	Power Off Flag
PCON.4	WLE	Watchdog Load Enable. This flag must be set by software prior to loading timer T3 (watchdog timer). It is cleared when timer T3 is loaded.
PCON.3	GF1	General-purpose flag bit.
PCON.2	GF0	General-purpose flag bit.
PCON.1	PD	Power-down bit. Setting this bit activates the power-down mode. It can only be set if input $\overline{EW}$ is high.
PCON.0	IDL	Idle mode bit. Setting this bit activates the Idle mode.

If logic 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (00X00000).

SU00954

Figure 3. Power Control Register (PCON)

IEN1 (E8H)

7	6	5	4	3	2	1	0
ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0
(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
IEN1.7	ET2	Enable Timer T2 overflow interrupt(s)
IEN1.6	ECM2	Enable T2 Comparator 2 interrupt
IEN1.5	ECM1	Enable T2 Comparator 1 interrupt
IEN1.4	ECM0	Enable T2 Comparator 0 interrupt
IEN1.3	ECT3	Enable T2 Capture register 3 interrupt
IEN1.2	ECT2	Enable T2 Capture register 2 interrupt
IEN1.1	ECT1	Enable T2 Capture register 1 interrupt
IEN1.0	ECT0	Enable T2 Capture register 0 interrupt

SU00755

In all cases, if the enable bit is 0, then the interrupt is disabled, and if the enable bit is 1, then the interrupt is enabled.

Figure 28. Interrupt Enable Register (IEN1)

IP0 (B8H)

7	6	5	4	3	2	1	0
–	PAD	PS1	PS0	PT1	PX1	PT0	PX0
(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
IP0.7	–	Unused
IP0.6	PAD	ADC interrupt priority level
IP0.5	PS1	SIO1 (I^2C) interrupt priority level
IP0.4	PS0	SIO0 (UART) interrupt priority level
IP0.3	PT1	Timer 1 interrupt priority level
IP0.2	PX1	External interrupt 1 priority level
IP0.1	PT0	Timer 0 interrupt priority level
IP0.0	PX0	External interrupt 0 priority level

SU00763

Figure 29. Interrupt Priority Register (IP0)

IP0H (B7H)

7	6	5	4	3	2	1	0
–	PADH	PS1H	PS0H	PT1H	PX1H	PT0H	PX0H
(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
IP0H.7	–	Unused
IP0H.6	PADH	ADC interrupt priority level high
IP0H.5	PS1H	SIO1 (I^2C) interrupt priority level high
IP0H.4	PS0H	SIO0 (UART) interrupt priority level high
IP0H.3	PT1H	Timer 1 interrupt priority level high
IP0H.2	PX1H	External interrupt 1 priority level high
IP0H.1	PT0H	Timer 0 interrupt priority level high
IP0H.0	PX0H	External interrupt 0 priority level high

SU00983

Figure 30. Interrupt Priority Register High (IP0H)

SIO1, I2C Serial I/O: The I2C 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 I2C bus may be used for test and diagnostic purposes

The output latches of P1.6 and P1.7 must be set to logic 1 in order to enable SIO1.

The P87C554 on-chip I2C logic provides a serial interface that meets the I2C bus specification and supports all transfer modes (other than the low-speed mode) from and to the I2C bus. 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 I2C bus.

The CPU interfaces to the I2C 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 I2C bus via two port 1 pins: P1.6/SCL (serial clock line) and P1.7/SDA (serial data line).

A typical I2C bus configuration is shown in Figure 33, and Figure 34 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 I2C 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 I2C bus will not be released.

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/$\overline{W}$) 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/$\overline{W}$) 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.

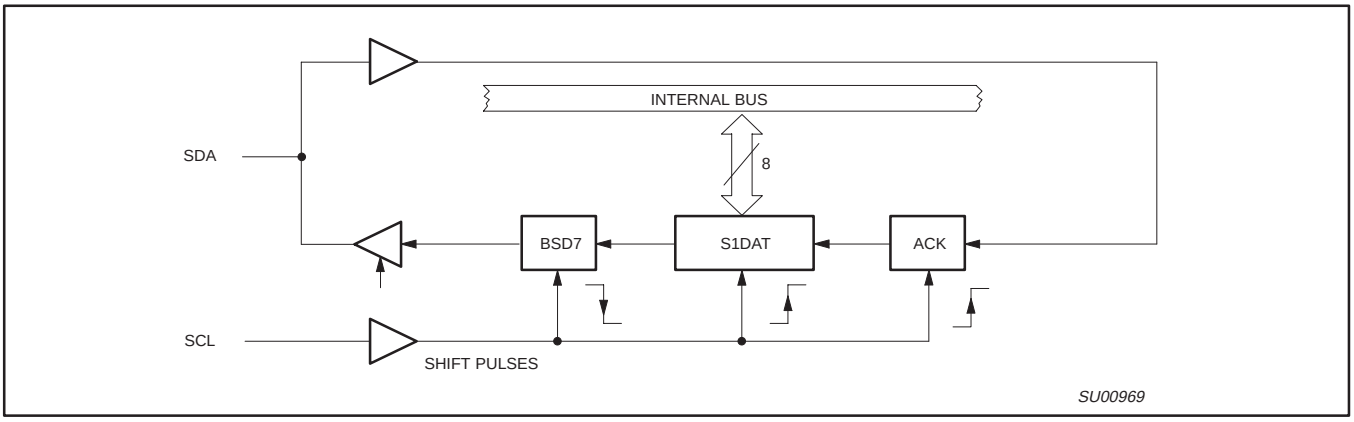


Figure 38. Serial Input/Output Configuration

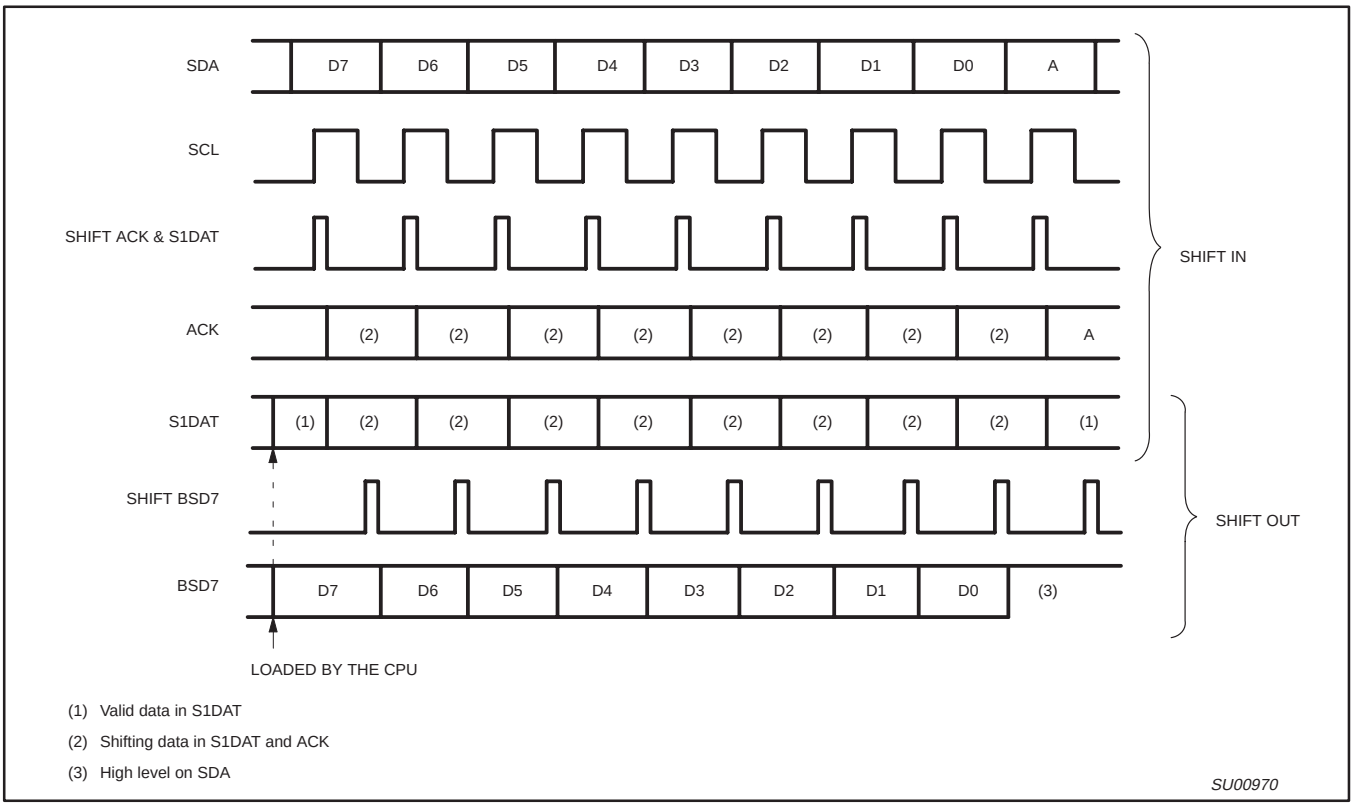


Figure 39. Shift-in and Shift-out Timing

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

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 I2C 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.

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 I2C 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 I2C 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.

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 40–43. 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 40-43, 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 6-10.

Master Transmitter Mode: In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 40). Before the master transmitter mode can be entered, S1CON must be initialized as follows:

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	bit rate	1	0	0	0	X	bit rate	

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 I²C 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 6. After a repeated start condition (state 10H). SIO1 may switch to the master receiver mode by loading S1DAT with SLA+R).

Master Receiver Mode: In the master receiver mode, a number of data bytes are received from a slave transmitter (see Figure 41). 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 7. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 7. After a repeated start condition (state 10H), SIO1 may switch to the master transmitter mode by loading S1DAT with SLA+W.

Slave Receiver Mode: In the slave receiver mode, a number of data bytes are received from a master transmitter (see Figure 42). To initiate the slave receiver mode, S1ADR and S1CON must be loaded as follows:

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

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.

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	X	1	0	0	0	1	X	X

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 8. 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 I²C 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 I²C bus.

Table 8. Slave Receiver Mode

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	
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/$\overline{W}$ 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
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.

Table 8. Slave Receiver Mode (Continued)

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	
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 9. Slave Transmitter Mode

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	
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/$\overline{W}$ 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	01	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 10. 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.

Slave Transmitter Mode: In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 43). 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 9. 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 I²C 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 I²C bus.

Miscellaneous States: There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 10). 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).

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 44). 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 I²C 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.

Data Transfer After Loss of Arbitration
Arbitration may be lost in the master transmitter and master receiver modes (see Figure 36). Loss of arbitration is indicated by the following states in S1STA; 38H, 68H, 78H, and B0H (see Figures 40 and 41).

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.

Forced Access to the I²C 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 I²C 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 I²C 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 45).

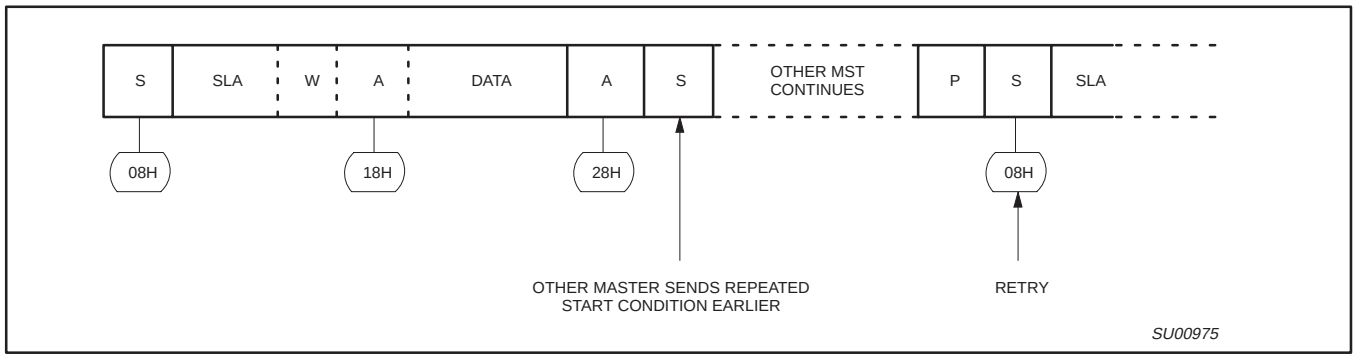


Figure 44. Simultaneous Repeated START Conditions from 2 Masters

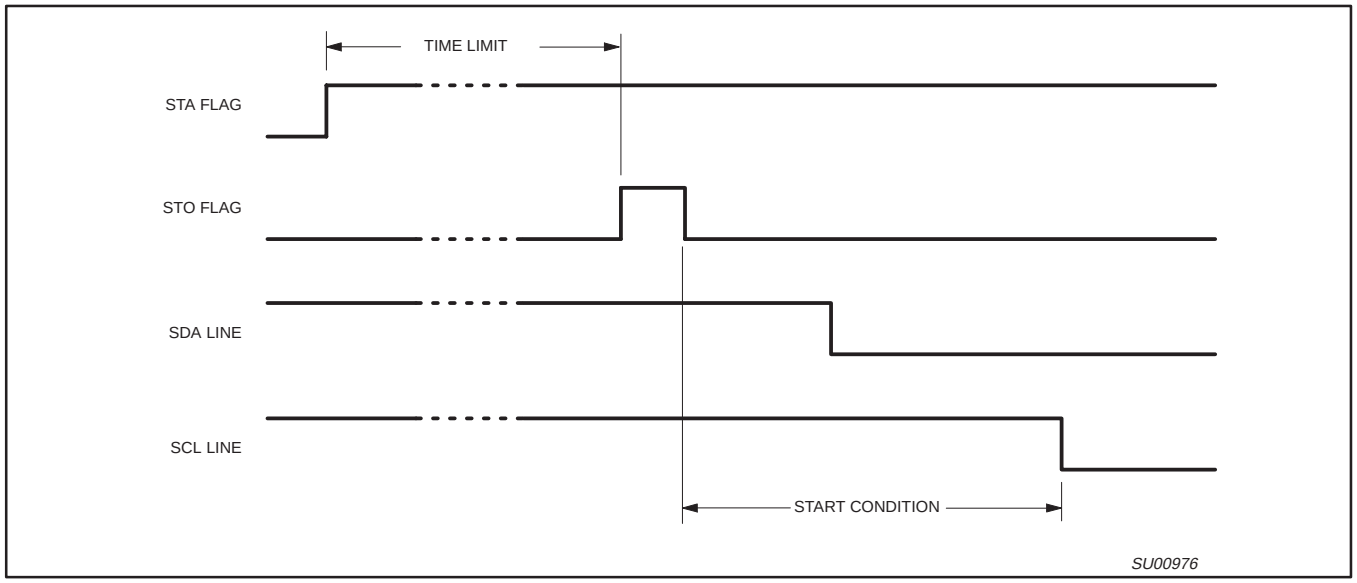


Figure 45. Forced Access to a Busy I2C Bus

I2C BUS OBSTRUCTED BY A LOW LEVEL ON SCL OR 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 46). 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.

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 10.

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/$\overline{W}$ 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 6, Table 7, Figure 40, and Figure 41. 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 I2C 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.

Slave Transmitter and Slave Receiver Modes

After initialization, SIO1 continually tests the I2C 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 8, Table 9, Figure 42, and Figure 43). If arbitration was lost while in the master mode, the master mode is restarted after the current transfer. If a bus error occurs, the I2C 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.

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 I2C bus.

```
                        !*************************************************************************************************
                        ! SI01 EQUATE LIST
                        !*************************************************************************************************
                        !*************************************************************************************************
                        ! 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)
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

                        !*************************************************************************************************
                        ! 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 0
                                                           ! 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


                        !*************************************************************************************************
                        ! 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
```

```
                        !– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
                        ! 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. 0–1.
                        !– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
                        .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

                        !– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
                        ! 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      stsb0
                        .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
```

DC ELECTRICAL CHARACTERISTICS

V_{SS}, AV_{SS} = 0 V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS		UNIT
			MIN	MAX	
I_{DD}	Supply current operating	See notes 1 and 2 f_{OSC} = 16 MHz		16	mA
I_{ID}	Idle mode	See notes 1 and 3 f_{OSC} = 16 MHz		4	mA
I_{PD}	Power-down current	See notes 1 and 4; 2 V < V_{PD} < V_{DD} max		50	µA
Inputs					
V_{IL}	Input low voltage, except $\overline{EA}$, P1.6, P1.7		–0.5	$0.2V_{DD}$–0.1	V
V_{IL1}	Input low voltage to $\overline{EA}$		–0.5	$0.2V_{DD}$–0.3	V
V_{IL2}	Input low voltage to P1.6/SCL, P1.7/SDA[5]		–0.5	$0.3V_{DD}$	V
V_{IH}	Input high voltage, except XTAL1, RST		$0.2V_{DD}$+0.9	V_{DD}+0.5	V
V_{IH1}	Input high voltage, XTAL1, RST		$0.7V_{DD}$	V_{DD}+0.5	V
V_{IH2}	Input high voltage, P1.6/SCL, P1.7/SDA[5]		$0.7V_{DD}$	6.0	V
I_{IL}	Logical 0 input current, ports 1, 2, 3, 4, except P1.6, P1.7	V_{IN} = 0.45 V		–50	µA
I_{TL}	Logical 1-to-0 transition current, ports 1, 2, 3, 4, except P1.6, P1.7	See note 6		–650	µA
$\pm I_{IL1}$	Input leakage current, port 0, $\overline{EA}$, STADC, $\overline{EW}$	0.45 V < V_I < V_{DD}		10	µA
$\pm I_{IL2}$	Input leakage current, P1.6/SCL, P1.7/SDA	0 V < V_I < 6 V 0 V < V_{DD} < 5.5 V		10	µA
$\pm I_{IL3}$	Input leakage current, port 5	0.45 V < V_I < V_{DD}		1	µA
$\pm I_{IL4}$	Input leakage current, ports 1, 2, 3, 4 in high impedance mode	0.45 V < V_{in} < V_{DD}		10	µA
Outputs					
V_{OL}	Output low voltage, ports 1, 2, 3, 4, except P1.6, P1.7	I_{OL} = 1.6mA[7]		0.4	V
V_{OL1}	Output low voltage, port 0, ALE, $\overline{PSEN}$, $\overline{PWM0}$, $\overline{PWM1}$	I_{OL} = 3.2mA[7]		0.4	V
V_{OL2}	Output low voltage, P1.6/SCL, P1.7/SDA	I_{OL} = 3.0mA[7]		0.4	V
V_{OH}	Output high voltage, ports 1, 2, 3, 4, except P1.6/SCL, P1.7/SDA	V_{CC} = 2.7 V I_{OH} = –20 µA	V_{CC} – 0.7		V
		V_{CC} = 4.5 I_{OH} = –30 µA	V_{CC} – 0.7		V
V_{OH1}	Output high voltage (port 0 in external bus mode, ALE, $\overline{PSEN}$, $\overline{PWM0}$, $\overline{PWM1}$)[8]	V_{CC} = 2.7 V I_{OH} = –3.2mA	V_{CC} – 0.7		V
V_{OH2}	Output high voltage (RST)	$-I_{OH}$ = 400 µA $-I_{OH}$ = 120 µA	2.4 $0.8V_{DD}$		V V
R_{RST}	Internal reset pull-down resistor		40	225	kΩ
C_{IO}	Pin capacitance	Test freq = 1 MHz, T_{amb} = 25°C		10	pF
Analog Inputs					
AV_{DD}	Analog supply voltage: 87C554[9]	AV_{DD} = V_{DD}±0.2 V	2.7	5.5	V
AI_{DD}	Analog supply current: operating:	Port 5 = 0 to AV_{DD}		1.2	mA
AI_{ID}	Idle mode: 87C554			50	µA
AI_{PD}	Power-down mode: 87C554	2 V < AV_{PD} < AV_{DD} max		50	µA

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address
C – Clock
D – Input data
H – Logic level high
I – Instruction (program memory contents)
L – Logic level low, or ALE
P – $\overline{PSEN}$
Q – Output data
R – $\overline{RD}$ signal
t – Time
V – Valid
W – $\overline{WR}$ signal
X – No longer a valid logic level
Z – Float

Examples: t_{AVLL} = Time for address valid to ALE low.
t_{LLPL} = Time for ALE low to $\overline{PSEN}$ low.

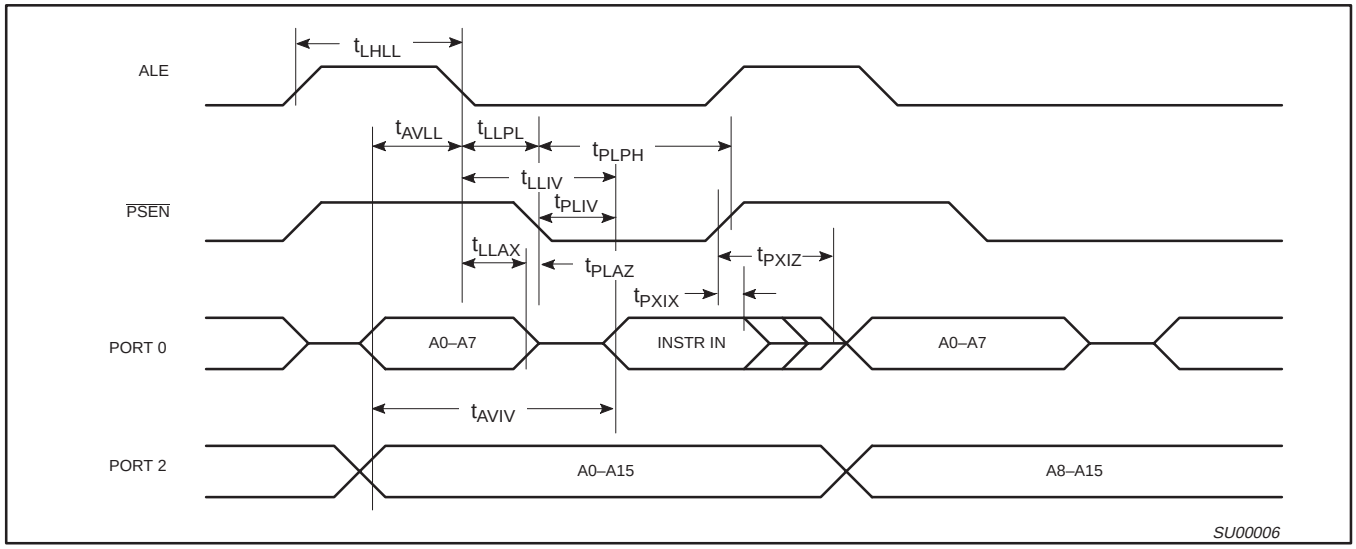


Figure 49. External Program Memory Read Cycle

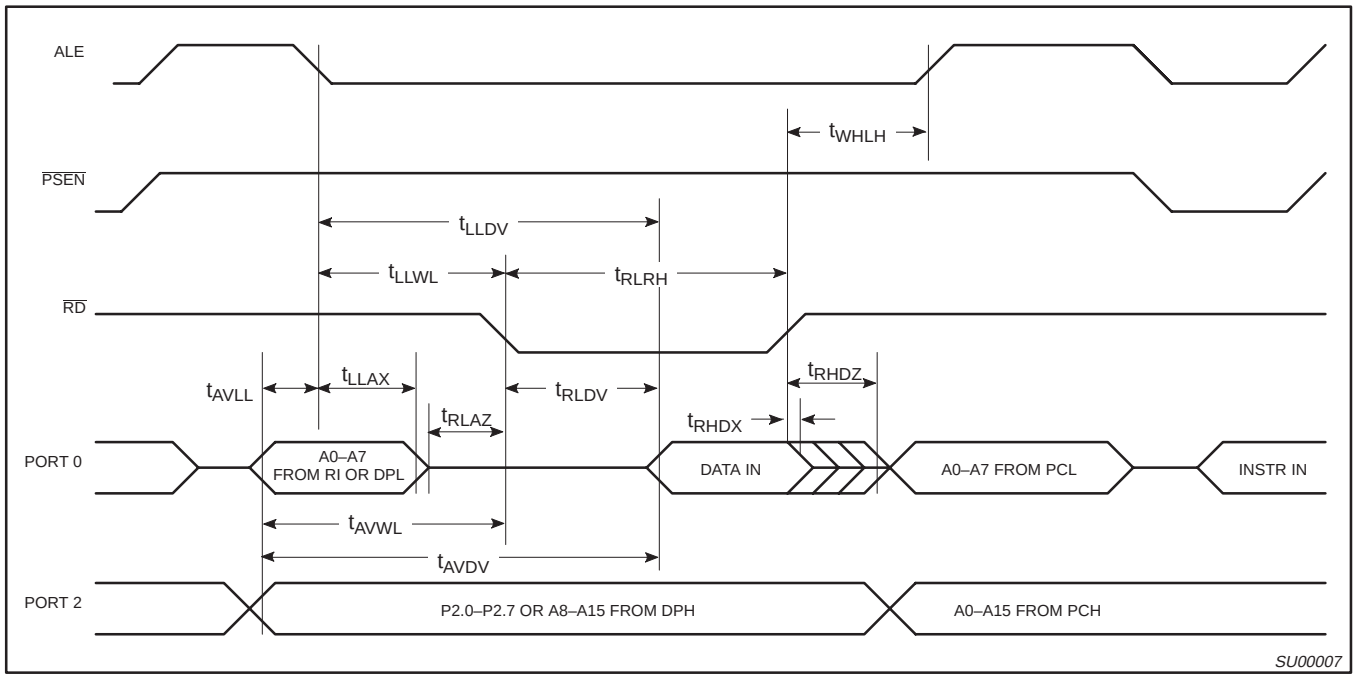
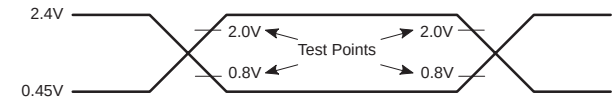


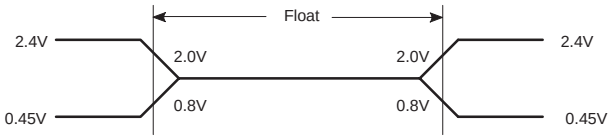
Figure 50. External Data Memory Read Cycle



NOTE:
AC inputs during testing are driven at 2.4V for a logic '1' and 0.45V for a logic '0'.
Timing measurements are made at 2.0V for a logic '1' and 0.8V for a logic '0'.

SU00215

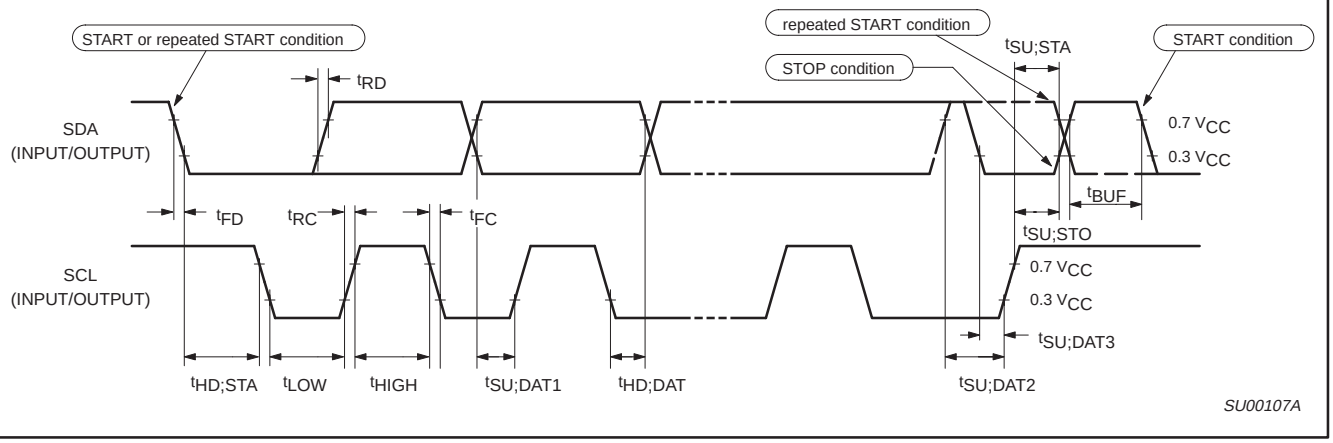
Figure 54. AC Testing Input/Output



NOTE:
The float state is defined as the point at which a port 0 pin sinks 3.2mA or sources 400μA at the voltage test levels.

SU00216

Figure 55. AC Testing Input, Float Waveform



SU00107A

Figure 56. Timing SIO1 (I²C) Interface

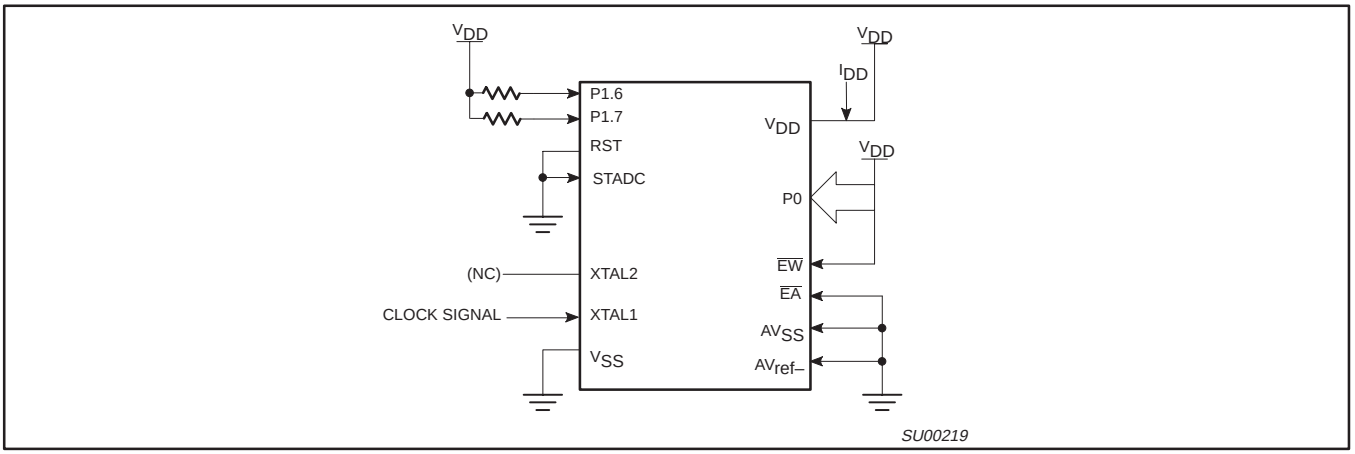


Figure 59. I_{DD} Test Condition, Idle Mode
All other pins are disconnected[2]

2. Idle Mode:
   a. The following pins must be forced to V_{DD}: Port 0 and $\overline{EW}$.
   b. The following pins must be forced to V_{SS}: RST, STADC, AV_{ss}, AV_{ref-}, and $\overline{EA}$.
   c. Ports 1.6 and 1.7 should be connected to V_{DD} through resistors of sufficiently high value such that the sink current into these pins cannot exceed the I_{OL1} spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
   d. The following pins must be disconnected: XTAL2 and all pins not specified above.

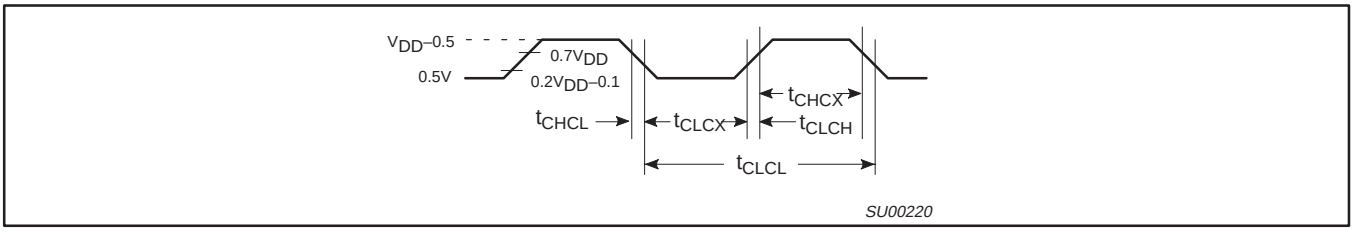


Figure 60. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes
$t_{CLCH} = t_{CHCL} = 5ns$

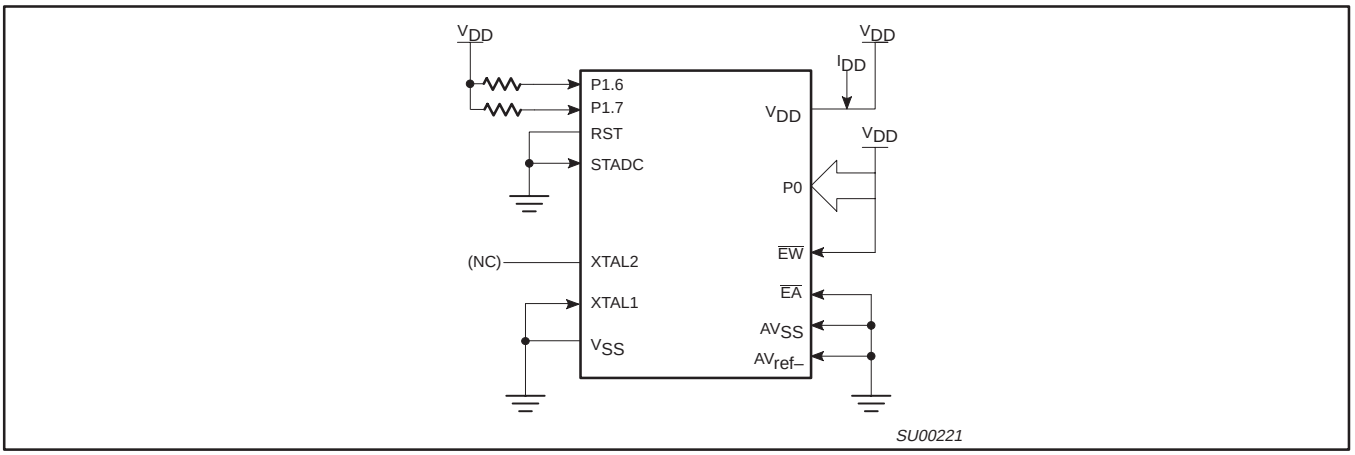


Figure 61. I_{DD} Test Condition, Power Down Mode
All other pins are disconnected. V_{DD} = 2 V to 5.5 V[3]

3. Power Down Mode:
   a. The following pins must be forced to V_{DD}: Port 0 and $\overline{EW}$.
   b. The following pins must be forced to V_{SS}: RST, STADC, XTAL1, AV_{ss}, AV_{ref-}, and $\overline{EA}$.
   c. Ports 1.6 and 1.7 should be connected to V_{DD} through resistors of sufficiently high value such that the sink current into these pins cannot exceed the I_{OL1} spec of these pins. These pins must not have logic 0 written to them prior to this measurement.
   d. The following pins must be disconnected: XTAL2 and all pins not specified above.

EPROM CHARACTERISTICS

The 87C554 contains three signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C554 manufactured by Philips:

(030H) = 15H indicates manufactured by Philips Components
(031H) = 93H indicates 87C554
(60H) = 01H

Program Verification

If security bits 2 or 3 have not been programmed, the on-chip program memory can be read out for program verification.

If the encryption table has been programmed, the data presented at port 0 will be the exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the encryption table contents in order to correctly decode the verification data. The encryption table itself cannot be read out.

Security Bits

With none of the security bits programmed the code in the program memory can be verified. If the encryption table is programmed, the code will be encrypted when verified. When only security bit 1 (see Table 11) is programmed, MOVC instructions executed from external program memory are disabled from fetching code bytes from the internal memory, $\overline{EA}$ is latched on Reset and all further programming of the EPROM is disabled. When security bits 1 and 2 are programmed, in addition to the above, verify mode is disabled.

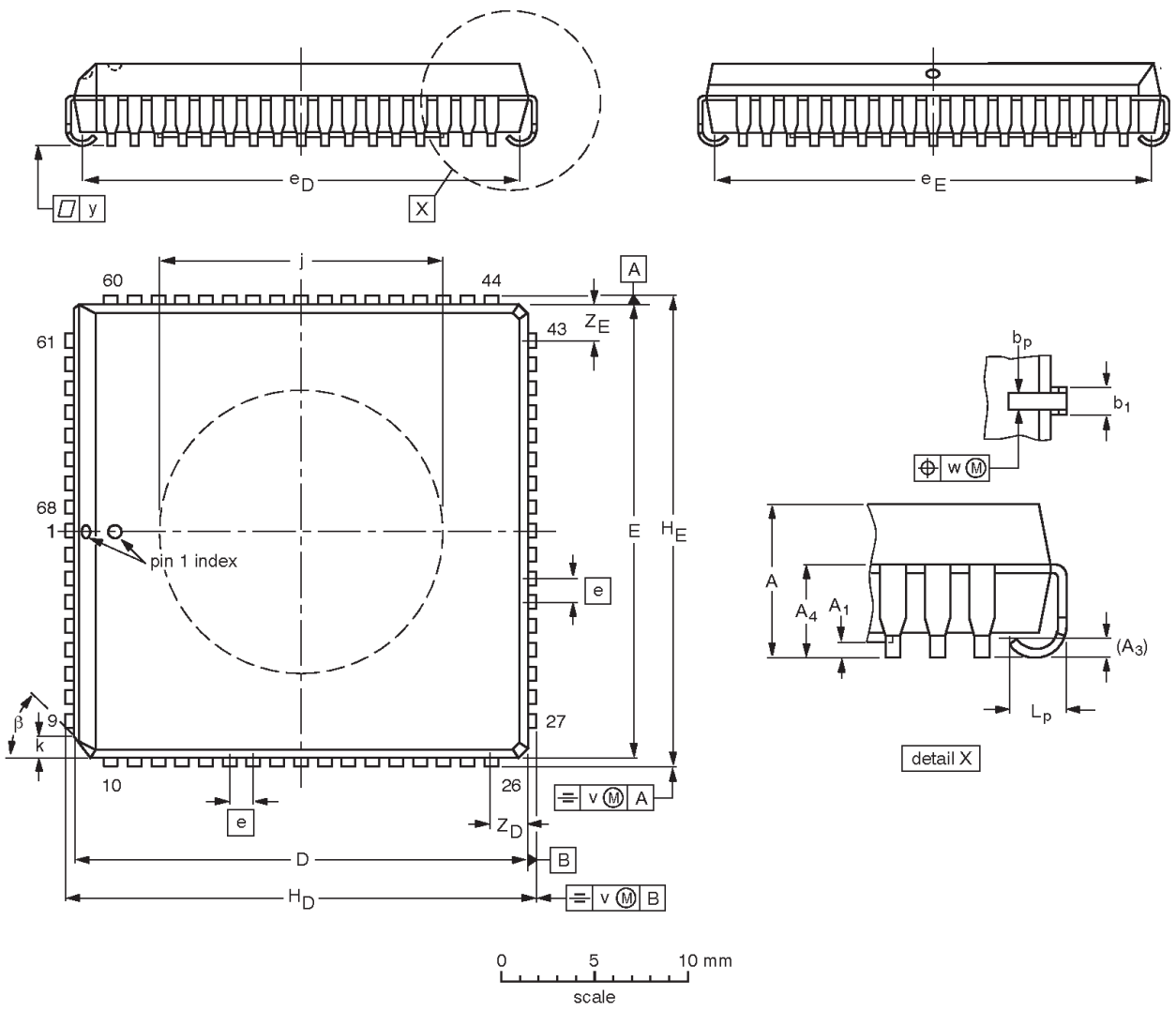
When all three security bits are programmed, all of the conditions above apply and all external program memory execution is disabled.

Table 11. Program Security Bits for EPROM Devices

PROGRAM LOCK BITS[1, 2]				
	SB1	SB2	SB3	PROTECTION DESCRIPTION
1	U	U	U	No Program Security features enabled. (Code verify will still be encrypted by the Encryption Array if programmed.)
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on Reset, and further programming of the EPROM is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, external execution is disabled.

NOTES:
1. P – programmed. U – unprogrammed.
2. Any other combination of the security bits is not defined.

PLCC68: plastic leaded chip carrier; 68 leads; pedestal SOT188-3



DIMENSIONS (mm dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	j	k	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.13	0.25	3.05	0.53 0.33	0.81 0.66	24.33 24.13	24.33 24.13	1.27	23.62 22.61	23.62 22.61	25.27 25.02	25.27 25.02	15.34 15.19	1.22 1.07	1.44 1.02	0.18	0.18	0.1	2.06	2.06	45°
inches	0.180 0.165	0.005	0.01	0.12	0.021 0.013	0.032 0.026	0.958 0.950	0.958 0.950	0.05	0.93 0.89	0.93 0.89	0.995 0.985	0.995 0.985	0.604 0.598	0.048 0.042	0.057 0.040	0.007	0.007	0.004	0.081	0.081	

Note

1. Plastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT188-3	112E10	MO-047	- - -			~~99-12-27~~ 01-11-14