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	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p80c554sbbd-157

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

MODE	PROGRAM MEMORY	ALE	$\overline{\text{PSEN}}$	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 8xC554 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 $\overline{\text{PSEN}}$ 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 $\overline{\text{PSEN}}$ 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 $\overline{\text{EW}}$ 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)

AUXR1 Address = A2H Reset Value = 0000 00x0B

Not Bit Addressable

	ADC8	AIDL	SRST	GF2	WUPD	0	—	DSP
Bit:	7	6	5	4	3	2	1	0

Symbol	Function
DPS	Data Pointer Switch—switches between DPRT0 and DPTR1. **DPS** **Operating Mode** 0 DPTR0 1 DPTR1
WUPD	Enable wakeup from powerdown.
GF2	General Purpose Flag—set and cleared by the user.
SRST	Software Reset
AIDL	Enables the ADC during idle mode.
ADC8	ADC Mode Switch—switches between 10-bit conversion and 8-bit conversion. **ADC8** **Operating Mode** 0 10-bit conversion (50 machine cycles) 1 8-bit conversion (24 machine cycles)

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

SU01081

Figure 7. AUXR1: DPTR Control Register

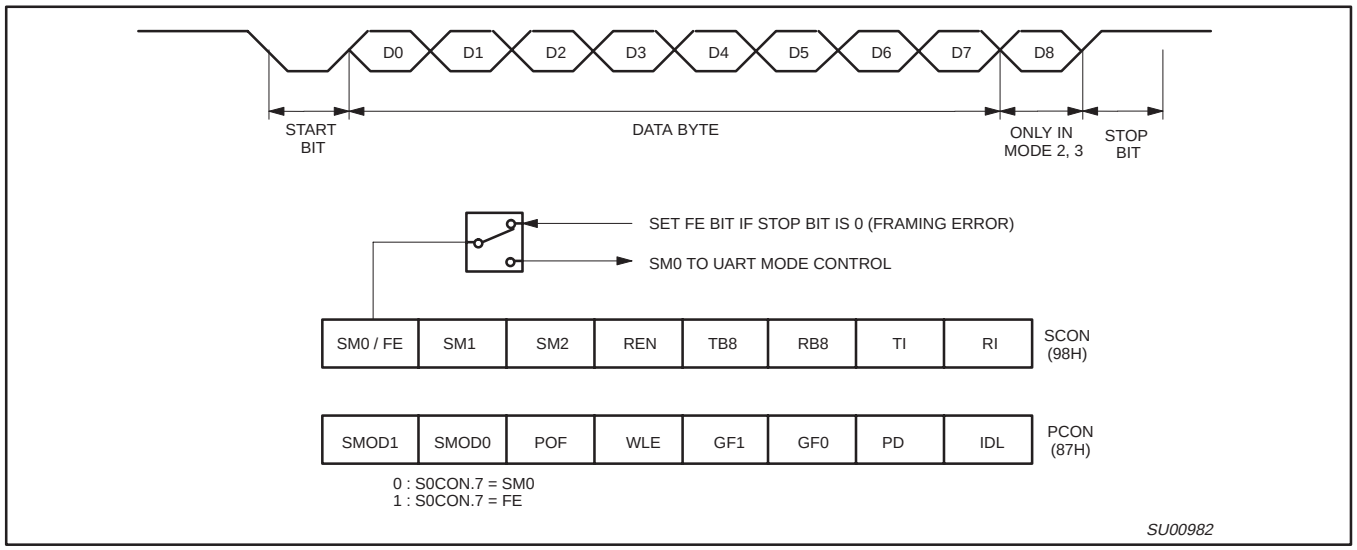

Figure 9. UART Framing Error Detection

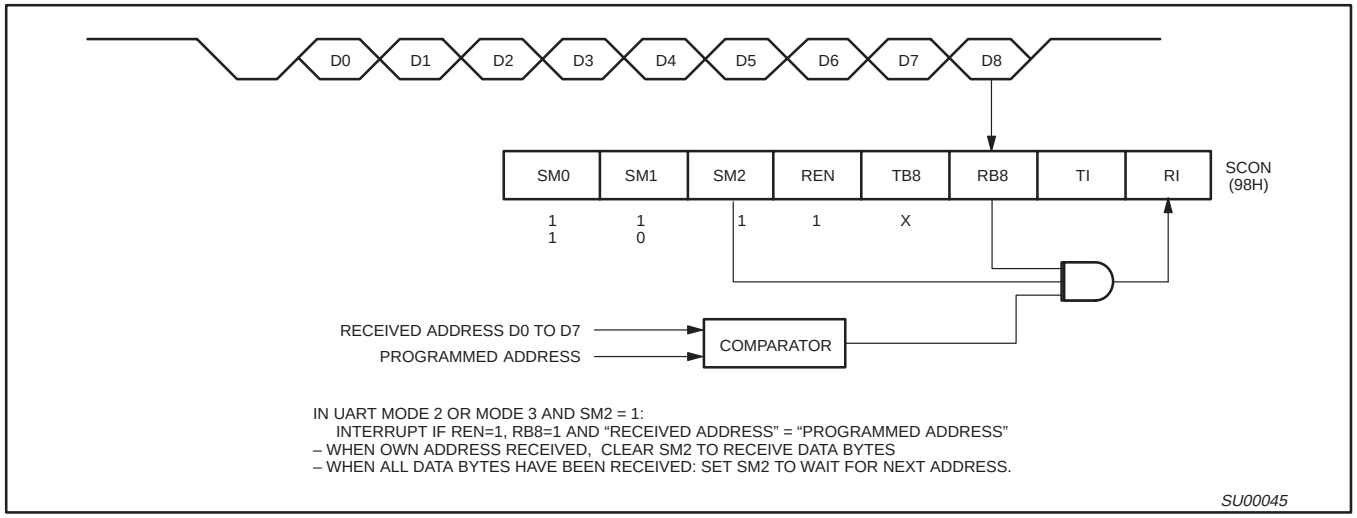

Figure 10. UART Multiprocessor Communication, Automatic Address Recognition

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. 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 b 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 slaves. Slave 0 requires a 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.

Reset Value = 00H

	7	6	5	4	3	2	1	0
CTCON (EBH)	CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN1	CTP0
	(MSB)							(LSB)

BIT	SYMBOL	CAPTURE/INTERRUPT ON:
CTCON.7	CTN3	Capture Register 3 triggered by a falling edge on CT3I
CTCON.6	CTP3	Capture Register 3 triggered by a rising edge on CT3I
CTCON.5	CTN2	Capture Register 2 triggered by a falling edge on CT2I
CTCON.4	CTP2	Capture Register 2 triggered by a rising edge on CT2I
CTCON.3	CTN1	Capture Register 1 triggered by a falling edge on CT1I
CTCON.2	CTP1	Capture Register 1 triggered by a rising edge on CT1I
CTCON.1	CTN0	Capture Register 0 triggered by a falling edge on CT0I
CTCON.0	CTP0	Capture Register 0 triggered by a rising edge on CT0I

SU01085

Figure 14. Capture Control Register (CTCON)

When a match with CM1 occurs, the controller resets bits 0-5 of port 4 if the corresponding bits of the reset/toggle enable register RTE are at logic 1 (see Figure 15 for RTE register function). If RTE is “0”, then P4.n is not affected by a match between CM1 or CM2 and Timer 2. When a match with CM2 occurs, the controller “toggles” bits 6 and 7 of port 4 if the corresponding bits of the RTE are at logic 1. The port latches of bits 6 and 7 are not toggled.
Two additional flip-flops store the last operation, and it is these flip-flops that are toggled.

Thus, if the current operation is “set,” the next operation will be “reset” even if the port latch is reset by software before the “reset” operation occurs. The first “toggle” after a chip RESET will set the port latch. The contents of these two flip-flops can be read at STE.6 and STE.7 (corresponding to P4.6 and P4.7, respectively). Bits STE.6 and STE.7 are read only (see Figure 16 for STE register function). A logic 1 indicates that the next toggle will set the port latch; a logic 0 indicates that the next toggle will reset the port latch. CM0, CM1, and CM2 are reset by the RST signal.

The modified port latch information appears at the port pin during S5P1 of the cycle following the cycle in which a match occurred. If the port is modified by software, the outputs change during S1P1 of the following cycle. Each port 4 bit can be set or reset by software at any time. A hardware modification resulting from a comparator match takes precedence over a software modification in the same cycle. When the comparator results require a “set” and a “reset” at the same time, the port latch will be reset.

Timer T2 Interrupt Flag Register TM2IR: Eight of the nine Timer T2 interrupt flags are located in special function register TM2IR (see Figure 17). The ninth flag is TM2CON.4.

The CT0I and CT1I flags are set during S4 of the cycle in which the contents of Timer T2 are captured. CT0I is scanned by the interrupt logic during S2, and CT1I is scanned during S3. CT2I and CT3I are set during S6 and are scanned during S4 and S5. The associated interrupt requests are recognized during the following cycle. If these flags are polled, a transition at CT0I or CT1I will be recognized one cycle before a transition on CT2I or CT3I since registers are read during S5. The CMI0, CMI1, and CMI2 flags are set during S6 of the cycle following a match. CMI0 is scanned by the interrupt logic during S2; CMI1 and CMI2 are scanned during S3 and S4. A match will be recognized by the interrupt logic (or by polling the flags) two cycles after the match takes place.

The 16-bit overflow flag (T2OV) and the byte overflow flag (T2BO) are set during S6 of the cycle in which the overflow occurs. These flags are recognized by the interrupt logic during the next cycle.

Special function register IP1 (Figure 17) is used to determine the Timer T2 interrupt priority. Setting a bit high gives that function a high priority, and setting a bit low gives the function a low priority. The functions controlled by the various bits of the IP1 register are shown in Figure 17.

Reset Value = 00H

	7	6	5	4	3	2	1	0
RTE (EFH)	TP47	TP46	RP45	RP44	RP43	RP42	RO41	RP40
	(MSB)							(LSB)

BIT	SYMBOL	FUNCTION
RTE.7	TP47	If “1” then P4.7 toggles on a match between CM1 and Timer T2
RTE.6	TP46	If “1” then P4.6 toggles on a match between CM1 and Timer T2
RTE.5	RP45	If “1” then P4.5 is reset on a match between CM1 and Timer T2
RTE.4	RP44	If “1” then P4.4 is reset on a match between CM1 and Timer T2
RTE.3	RP43	If “1” then P4.3 is reset on a match between CM1 and Timer T2
RTE.2	RP42	If “1” then P4.2 is reset on a match between CM1 and Timer T2
RTE.1	RP41	If “1” then P4.1 is reset on a match between CM1 and Timer T2
RTE.0	RP40	If “1” then P4.0 is reset on a match between CM1 and Timer T2

SU01086

Figure 15. Reset/Toggle Enable Register (RTE)

Timer T3, The Watchdog Timer

In addition to Timer T2 and the standard timers, a watchdog timer is also incorporated on the 8xC554. The purpose of a watchdog timer is to reset the microcontroller if it enters erroneous processor states (possibly caused by electrical noise or RFI) within a reasonable period of time. An analogy is the “dead man’s handle” in railway locomotives. When enabled, the watchdog circuitry will generate a system reset if the user program fails to reload the watchdog timer within a specified length of time known as the “watchdog interval.”

Watchdog Circuit Description: The watchdog timer (Timer T3) consists of an 8-bit timer with an 11-bit prescaler as shown in Figure 18. The prescaler is fed with a signal whose frequency is 1/6 the oscillator frequency (0.5 MHz with a 12-MHz oscillator). The 8-bit timer is incremented every “t” seconds, where:

$t = 6 \times 2048 \times 1/f_{OSC}$
(= 0.75 ms at f_{OSC} = 16 MHz; = 0.5 ms at f_{OSC} = 24 MHz)

If the 8-bit timer overflows, a short internal reset pulse is generated which will reset the 8xC554. A short output reset pulse is also generated at the RST pin. This short output pulse (3 machine cycles) may be destroyed if the RST pin is connected to a capacitor. This would not, however, affect the internal reset operation.

Watchdog operation is activated when external pin $\overline{EW}$ is tied low. When $\overline{EW}$ is tied low, it is impossible to disable the watchdog operation by software.

How to Operate the Watchdog Timer: The watchdog timer has to be reloaded within periods that are shorter than the programmed watchdog interval; otherwise the watchdog timer will overflow and a system reset will be generated. The user program must therefore continually execute sections of code which reload the watchdog timer. The period of time elapsed between execution of these sections of code must never exceed the watchdog interval. When using a 16-MHz oscillator, the watchdog interval is programmable between 0.75 ms and 196 ms. When using a 24-MHz oscillator, the watchdog interval is programmable between 0.5 ms and 127.5 ms.

In order to prepare software for watchdog operation, a programmer should first determine how long his system can sustain an erroneous processor state. The result will be the maximum watchdog interval. As the maximum watchdog interval becomes shorter, it becomes more difficult for the programmer to ensure that the user program always reloads the watchdog timer within the watchdog interval, and thus it becomes more difficult to implement watchdog operation.

The programmer must now partition the software in such a way that reloading of the watchdog is carried out in accordance with the above requirements. The programmer must determine the execution times of all software modules. The effect of possible conditional branches, subroutines, external and internal interrupts must all be taken into account. Since it may be very difficult to evaluate the execution times of some sections of code, the programmer should use worst case estimations. In any event, the programmer must make sure that the watchdog is not activated during normal operation.

The watchdog timer is reloaded in two stages in order to prevent erroneous software from reloading the watchdog. First PCON.4 (WLE) must be set. The T3 may be loaded. When T3 is loaded, PCON.4 (WLE) is automatically reset. T3 cannot be loaded if PCON.4 (WLE) is reset. Reload code may be put in a subroutine as it is called frequently. Since Timer T3 is an up-counter, a reload value of 00H gives the maximum watchdog interval (255 ms with a 12-MHz oscillator), and a reload value of 0FFH gives the minimum watchdog interval (1 ms with a 12-MHz oscillator).

In the idle mode, the watchdog circuitry remains active. When watchdog operation is implemented, the power-down mode cannot be used since both states are contradictory. Thus, when watchdog operation is enabled by tying external pin $\overline{EW}$ low, it is impossible to enter the power-down mode, and an attempt to set the power-down bit (PCON.1) will have no effect. PCON.1 will remain at logic 0.

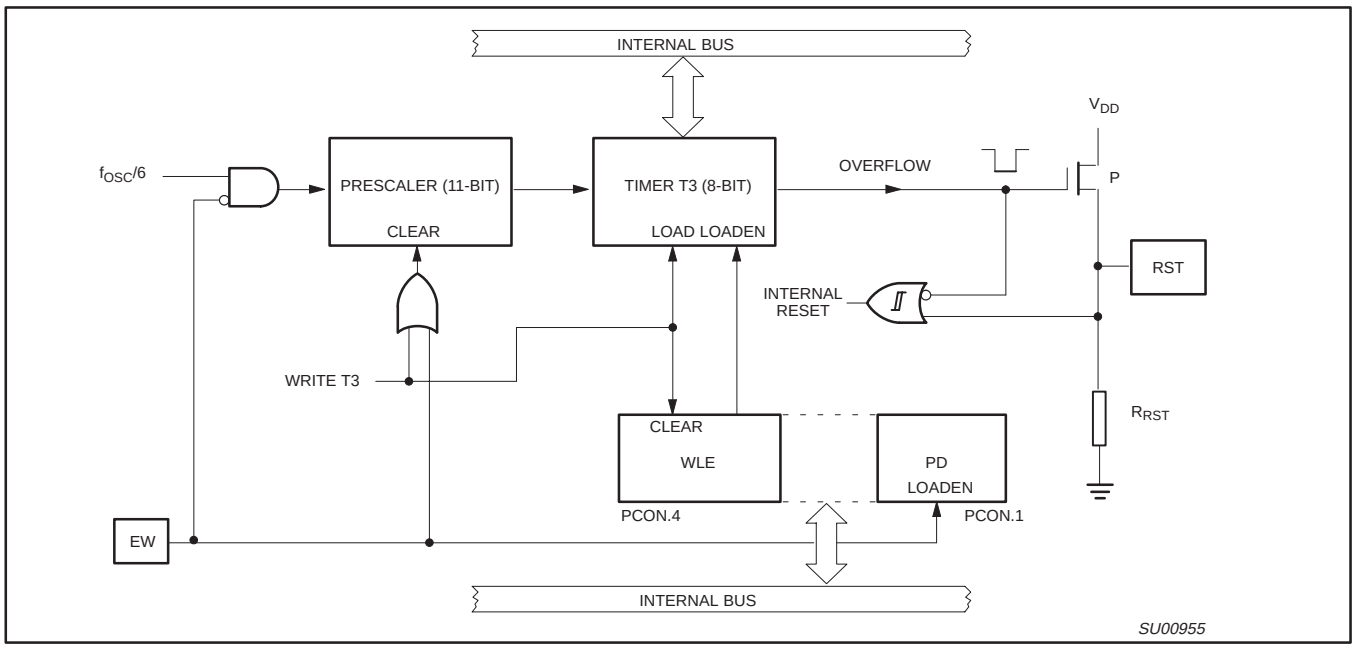

Figure 18. Watchdog Timer

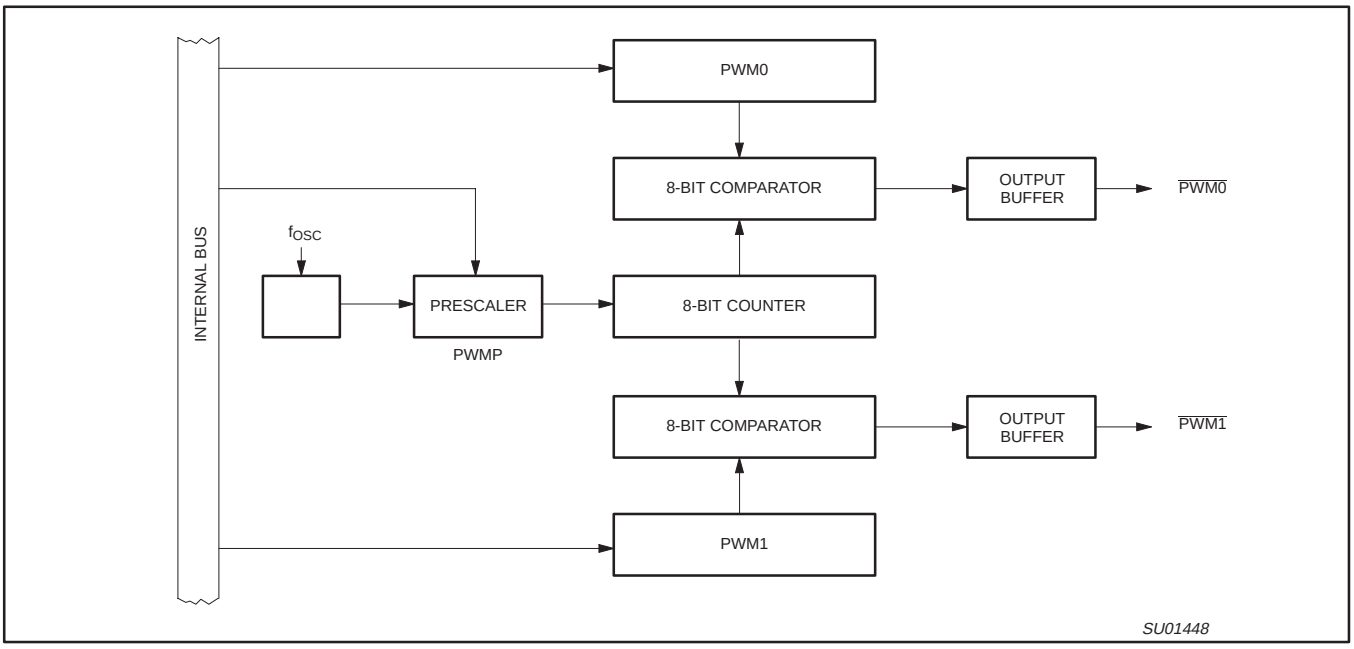

Figure 19. Functional Diagram of Pulse Width Modulated Outputs

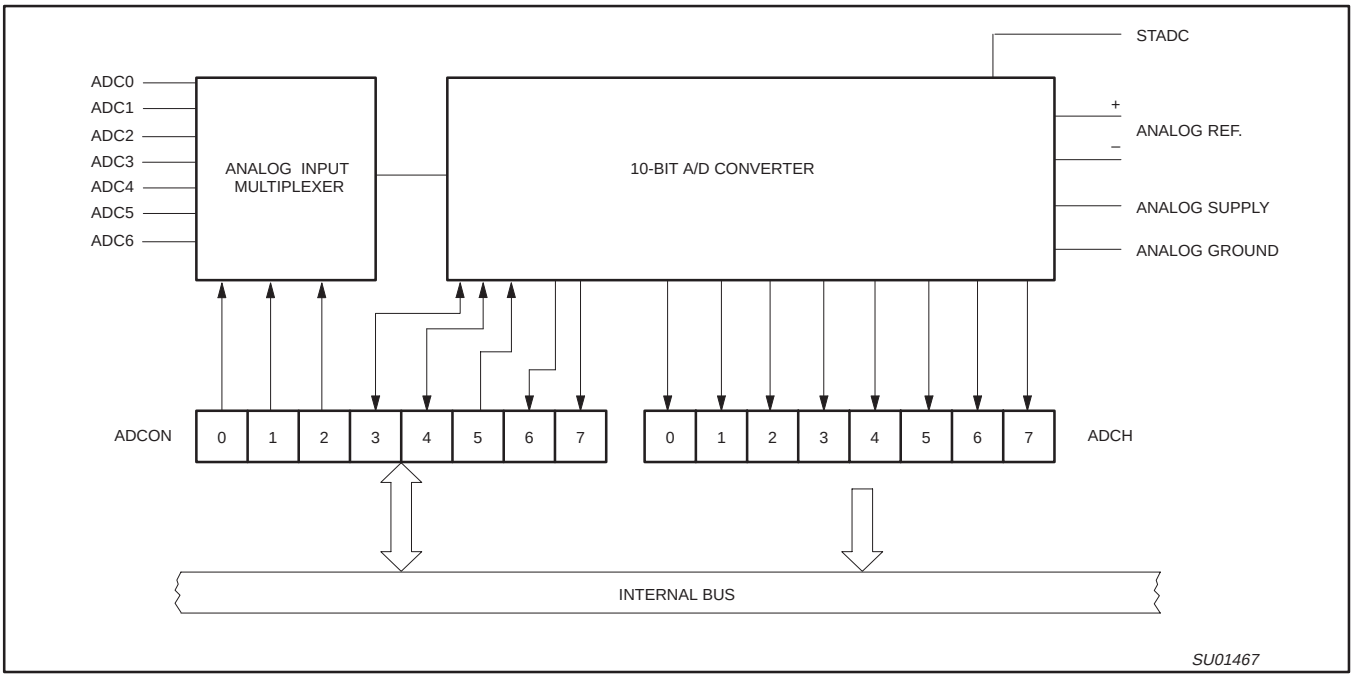

Figure 20. Functional Diagram of Analog Input Circuitry

10-Bit Analog-to-Digital Conversion: Figure 21 shows the elements of a successive approximation (SA) ADC. The ADC contains a DAC which converts the contents of a successive approximation register to a voltage (VDAC) which is compared to the analog input voltage (Vin). The output of the comparator is fed to the successive approximation control logic which controls the successive approximation register. A conversion is initiated by setting ADCS in the ADCON register. ADCS can be set by software only or by either hardware or software.

The software only start mode is selected when control bit ADCON.5 (ADEX) = 0. A conversion is then started by setting control bit ADCON.3 (ADCS). The hardware or software start mode is selected when ADCON.5 = 1, and a conversion may be started by setting ADCON.3 as above or by applying a rising edge to external pin STADC. When a conversion is started by applying a rising edge, a low level must be applied to STADC for at least one machine cycle followed by a high level for at least one machine cycle.

Reset Value = xx00 0000B

	7	6	5	4	3	2	1	0
ADCON (C5H)	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0
	(MSB)							(LSB)

Bit	Symbol	Function
ADCON.7	ADC.1	Bit 1 of ADC result
ADCON.6	ADC.0	Bit 0 of ADC result
ADCON.5	ADEX	Enable external start of conversion by STADC 0 = Conversion can be started by software only (by setting ADCS) 1 = Conversion can be started by software or externally (by a rising edge on STADC)
ADCON.4	ADCI	ADC interrupt flag: this flag is set when an A/D conversion result is ready to be read. An interrupt is invoked if it is enabled. The flag may be cleared by the interrupt service routine. While this flag is set, the ADC cannot start a new conversion. ADCI cannot be set by software.
ADCON.3	ADCS	ADC start and status: setting this bit starts an A/D conversion. It may be set by software or by the external signal STADC. The ADC logic ensures that this signal is HIGH while the ADC is busy. On completion of the conversion, ADCS is reset immediately after the interrupt flag has been set. ADCS cannot be reset by software. A new conversion may not be started while either ADCS or ADCI is high.

ADCI	ADCS	ADC Status
0	0	ADC not busy; a conversion can be started
0	1	ADC busy; start of a new conversion is blocked
1	0	Conversion completed; start of a new conversion requires ADCI=0
1	1	Conversion completed; start of a new conversion requires ADCI=0

If ADCI is cleared by software while ADCS is set at the same time, a new A/D conversion with the same channel number may be started.
But it is recommended to reset ADCI ***before*** ADCS is set.

ADCON.2	AADR2	Analogue input select: this binary coded address selects one of the eight analogue port bits of P5 to be input to the converter. It can only be changed when ADCI and ADCS are both LOW.
ADCON.1	AADR1	
ADCON.0	AADR0	

AADR2	AADR1	AADR0	Selected Analog Channel
0	0	0	ADC0 (P5.0)
0	0	1	ADC1 (P5.1)
0	1	0	ADC2 (P5.2)
0	1	1	ADC3 (P5.3)
1	0	0	ADC4 (P5.4)
1	0	1	ADC5 (P5.5)
1	1	0	ADC6 (P5.6)

SU01468

Figure 23. ADC Control Register (ADCON)

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 (I²C) 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 (I²C) 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)

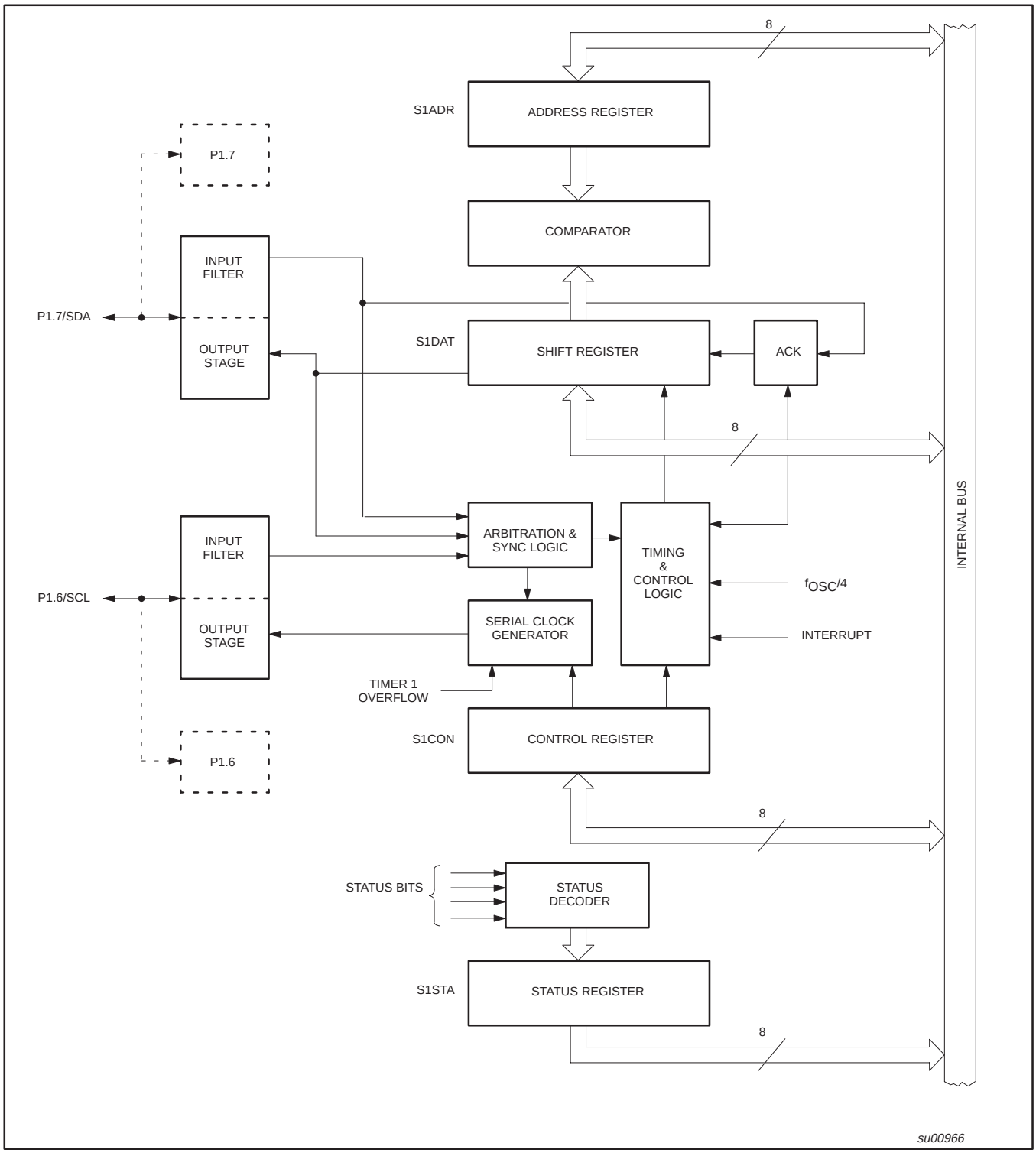

Figure 35. I2C Bus Serial Interface Block Diagram

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: $f_{OSC}/60$, $f_{OSC}/4800$, 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.

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 I²C bus status.

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.

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 I²C 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).

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.

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 microcontroller's own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.

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

The most significant bit corresponds to the first bit received from the I²C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I²C bus, and a logic 0 corresponds to a low level on the bus.

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.

	7	6	5	4	3	2	1	0
S1DAT (DAH)	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
	shift direction							

SD7 - SD0:

Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I²C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 38 shows how data in S1DAT is serially transferred to and from the SDA line.

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 39). 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.

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 I²C bus. The STO bit is also cleared when ENS1 = "0".

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

ENS1, the SIO1 Enable Bit

ENS1 = "0": When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. 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. P1.6 and P1.7 may be used as open drain I/O ports.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

ENS1 should not be used to temporarily release SIO1 from the I2C bus since, when ENS1 is reset, the I2C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

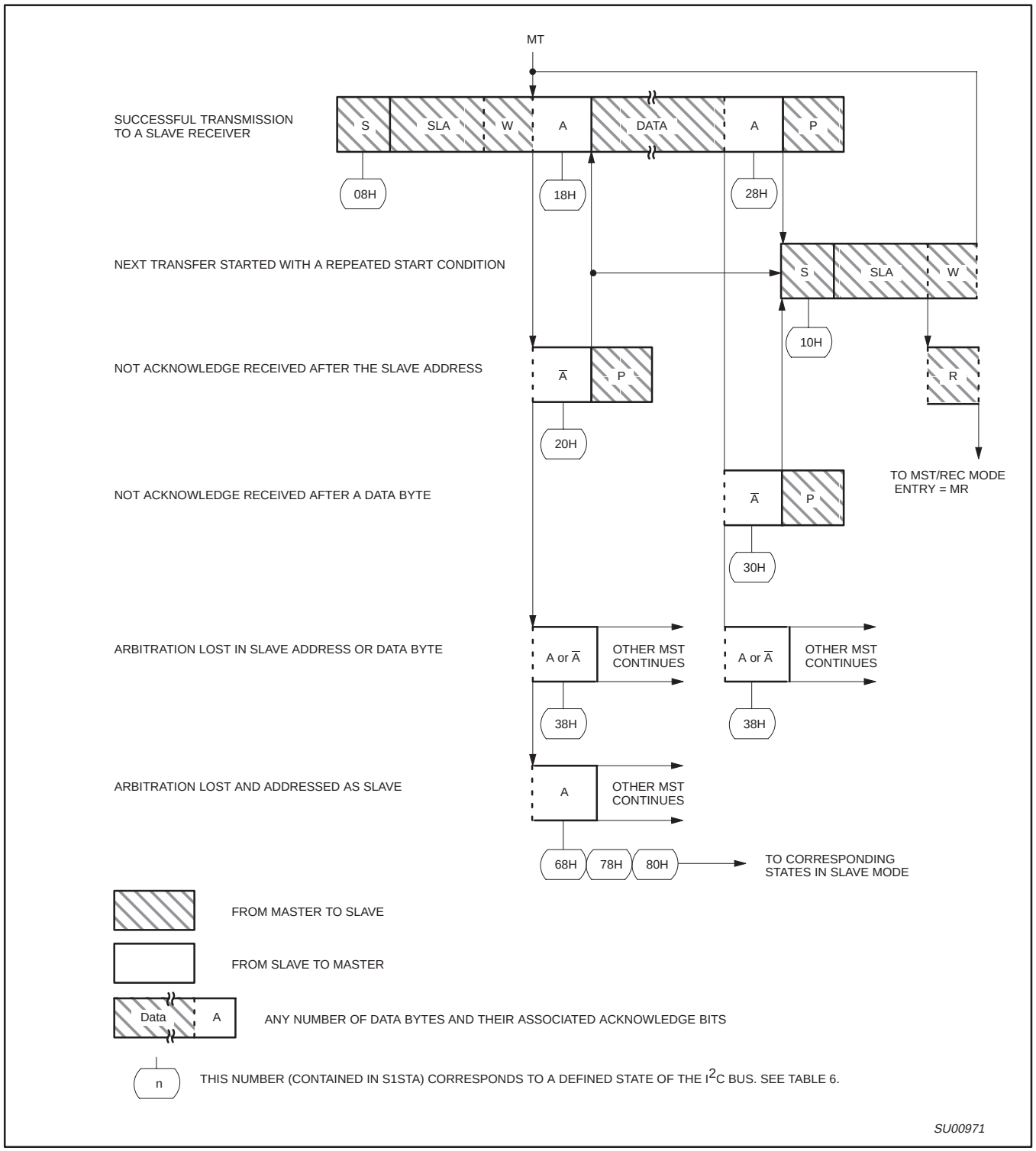

Figure 40. Format and States in the Master Transmitter Mode

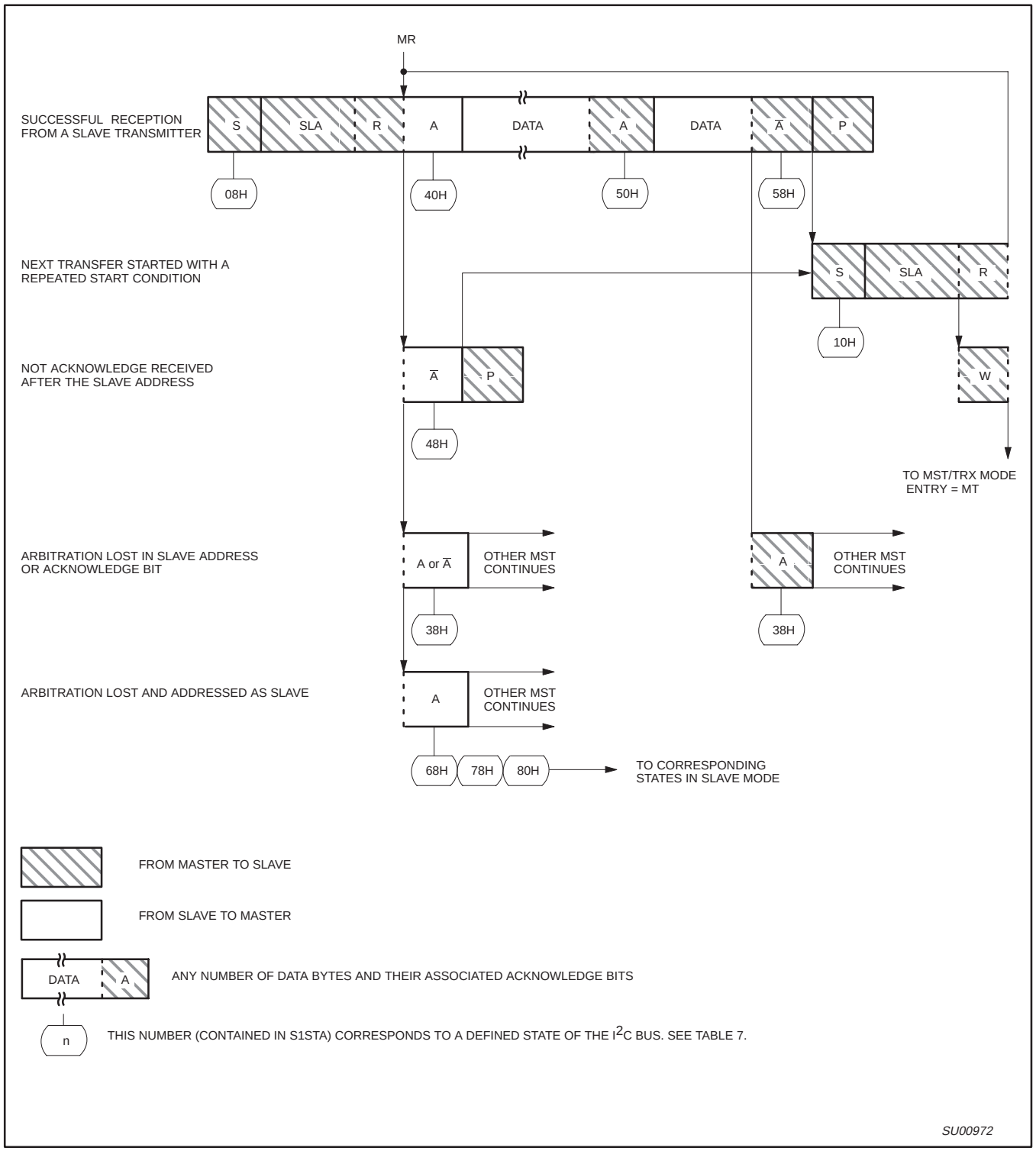

Figure 41. Format and States in the Master Receiver Mode

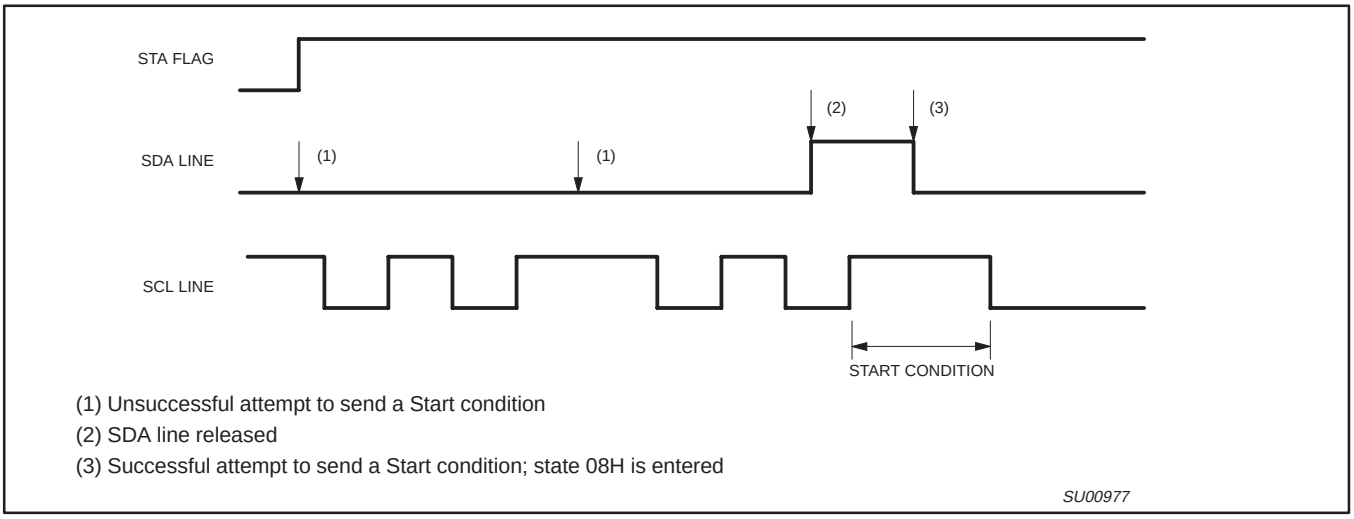

Figure 46. Recovering from a Bus Obstruction Caused by a Low Level on SDA

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

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 47. The initialization routine performs the following functions:

– S1ADR is loaded with the part's 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 I2C 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.

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.

THE STATE SERVICE ROUTINES

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.

```
                        !– – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – – –
                        ! 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
```

ABSOLUTE MAXIMUM RATINGS[1, 2, 3]

PARAMETER	RATING	UNIT
Storage temperature range	–65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ to V_{SS}	–0.5 to +13	V
Voltage on any other pin to V_{SS}	–0.5 to +6.5	V
Input, output DC current on any single I/O pin	5.0	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.0	W

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maxima.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

AC ELECTRICAL CHARACTERISTICS

V_{DD} and T_{amb} minimum and maximum, per device specifications table; V_{SS} = 0 V; C_L = 100 pF for Port 0, ALE and $\overline{PSEN}$; C_L = 80 pF for all other outputs unless otherwise specified.

SYMBOL	FIGURE	PARAMETER	16 MHz CLOCK		VARIABLE CLOCK		UNIT
			MIN	MAX	MIN	MAX	
External Program Memory							
$1/f_{CLK}$	49	System clock frequency, see Note 1			3.5	16	MHz
t_{LHLL}	49	ALE pulse width	37.5	–	t_{CLK}–25	–	ns
t_{AVLL}	49	Address valid to ALE LOW	6.25	–	0.5 t_{CLK}–25	–	ns
t_{LLAX}	49	Address hold after ALE LOW	6.25	–	0.5 t_{CLK}–25	–	ns
t_{LLIV}	49	ALE LOW to valid instruction in	–	60	–	2 t_{CLK}–65	ns
t_{LLPL}	49	ALE LOW to $\overline{PSEN}$ LOW	6.25	–	0.5 t_{CLK}–25	–	ns
t_{PLPH}	49	$\overline{PSEN}$ pulse width	48.75	–	1.5 t_{CLK}–45	–	ns
t_{PLIV}	49	$\overline{PSEN}$ LOW to valid instruction in	–	33.75	–	1.5 t_{CLK}–60	ns
t_{PXIX}	49	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXIZ}	49	Input instruction float after $\overline{PSEN}$	–	6.25	–	0.5 t_{CLK}–25	ns
t_{AVIV}	49	Address to valid instruction in	–	76.25	–	2.5 t_{CLK}–80	ns
t_{PLAZ}	49	$\overline{PSEN}$ LOW to address float	–	10	–	10	ns
External Data Memory							
t_{RLRH}	50, 51	$\overline{RD}$ pulse width	87.5	–	3 t_{CLK}–100	–	ns
t_{WLWH}	50, 51	$\overline{WR}$ pulse width	87.5	–	3 t_{CLK}–100	–	ns
t_{RLDV}	50, 51	$\overline{RD}$ LOW to valid data in	–	66.25	–	2.5 t_{CLK}–90	ns
t_{RHDX}	50, 51	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	50, 51	Data float after $\overline{RD}$	–	42.5	–	t_{CLK}–20	ns
t_{LLDV}	50, 51	ALE LOW to valid data in	–	100	–	4 t_{CLK}–150	ns
t_{AVDV}	50, 51	Address to valid data in	–	116.25	–	4.5 t_{CLK}–165	ns
t_{LLWL}	50, 51	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	43.75	143.75	1.5 t_{CLK}–50	1.5 t_{CLK}+50	ns
t_{AVWL}	50, 51	Address valid to $\overline{RD}$ low or $\overline{WR}$ LOW	50	–	2 t_{CLK}–75	–	ns
t_{QVWX}	50, 51	Data valid to $\overline{WR}$ transition	1.25	–	0.5 t_{CLK}–30	–	ns
t_{WHQX}	51	Data hold after $\overline{WR}$	6.25	–	0.5 t_{CLK}–25	–	ns
t_{QVWH}	50, 51	Data valid time $\overline{WR}$ HIGH	88.75	–	3.5 t_{CLK}–130	–	ns
t_{RLAZ}	50, 51	$\overline{RD}$ LOW to address float	–	0	–	0	ns
t_{WHLH}	50, 51	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	6.25	56.25	0.5 t_{CLK}–25	0.5 t_{CLK}+25	ns
External Clock							
t_{CHCX}	52	High time	33.3	50	$t_{CLK} \times 0.4$	$t_{CLK} \times 0.6$	ns
t_{CLCX}	52	Low time	33.3	50	$t_{CLK} \times 0.4$	$t_{CLK} \times 0.6$	ns
t_{CLCH}	52	Rise time	–	20	–	20	ns
t_{CHCL}	52	Fall time	–	20	–	20	ns
UART Timing – Shift Register Mode							
t_{XLXL}	53	Serial port clock cycle time	500	–	6 t_{CLK}	–	ns
t_{QVXH}	53	Output data setup to clock rising edge	179.5	–	5 t_{CLK} –133	–	ns
t_{XHQX}	53	Output data hold after clock rising edge	32.5	–	t_{CLK}–30	–	ns
t_{XHDX}	53	Input data hold after clock rising edge	0	–	0	–	ns
t_{XHDV}	53	Clock rising edge to input data valid	–	179.5	–	5 t_{CLK}–133	ns

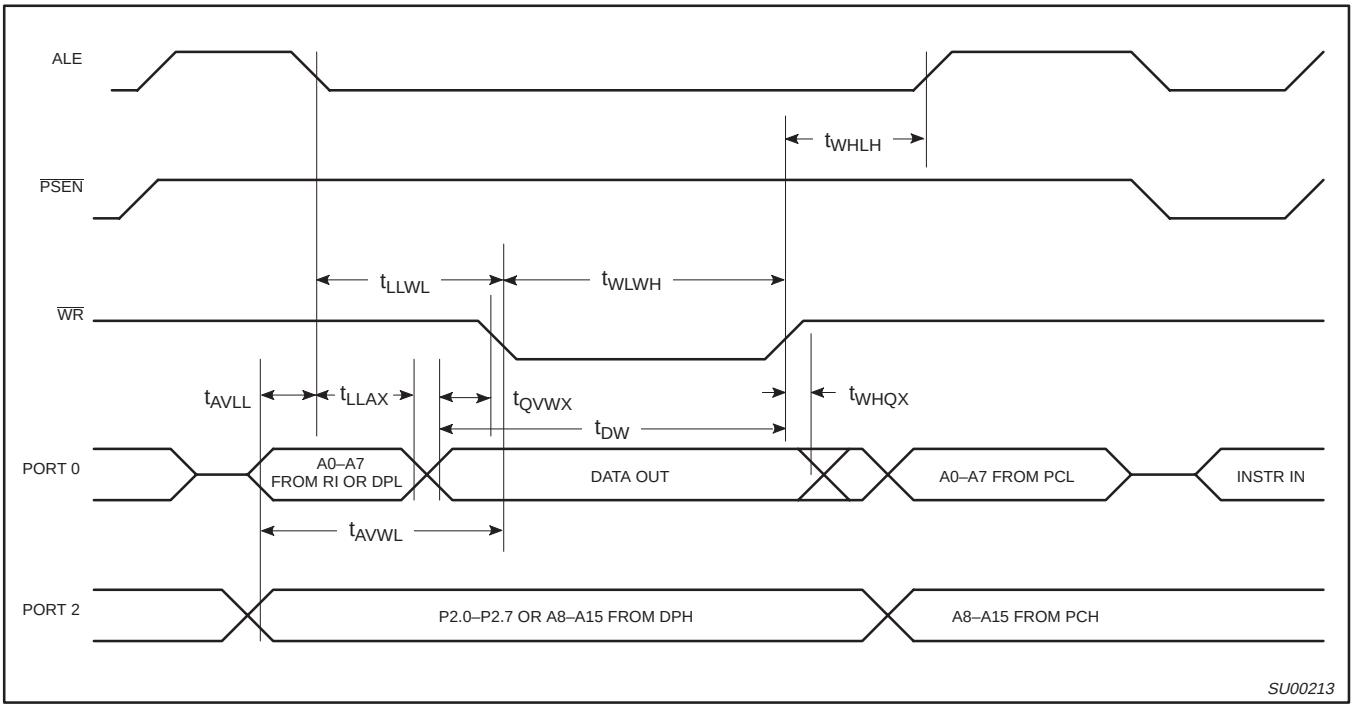

Figure 51. External Data Memory Write Cycle

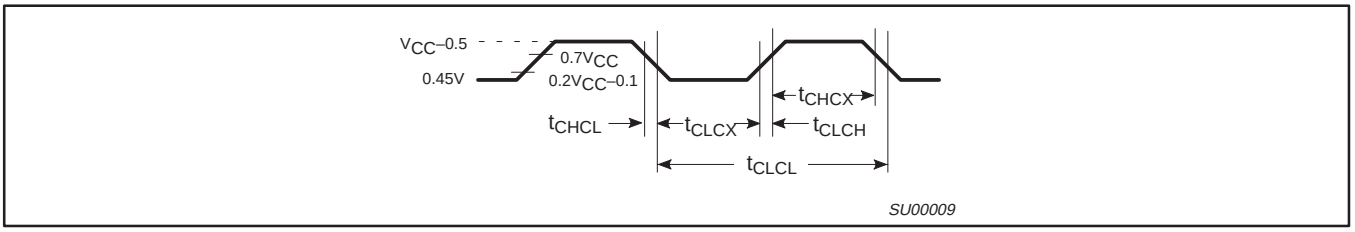

Figure 52. External Clock Drive XTAL1

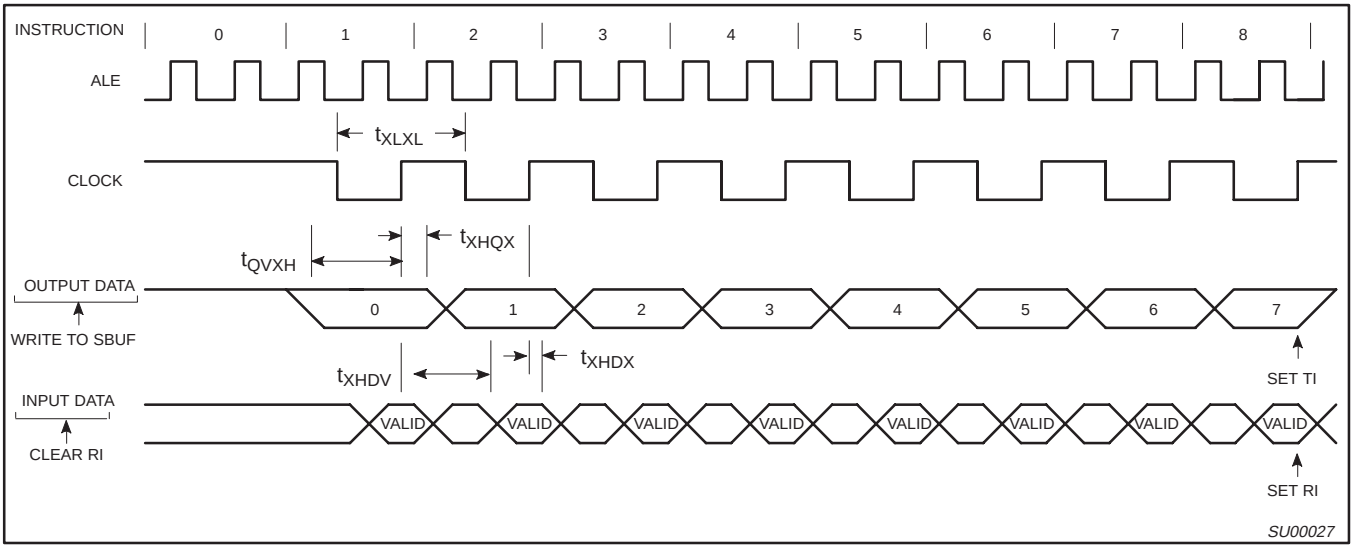

Figure 53. Shift Register Mode Timing

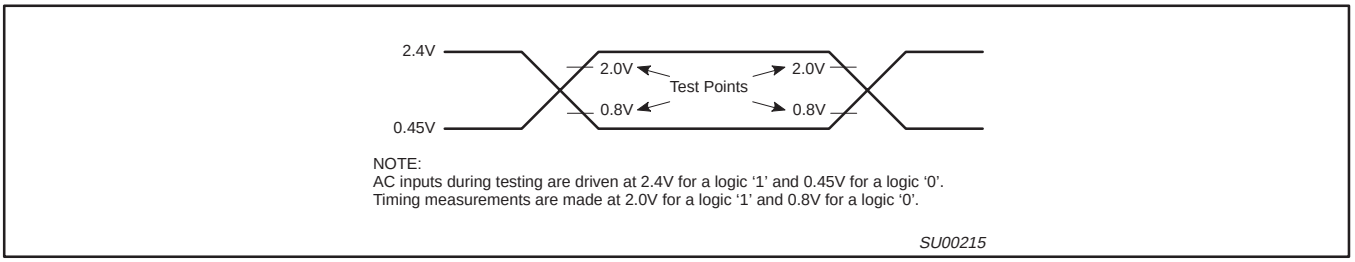

Figure 54. AC Testing Input/Output

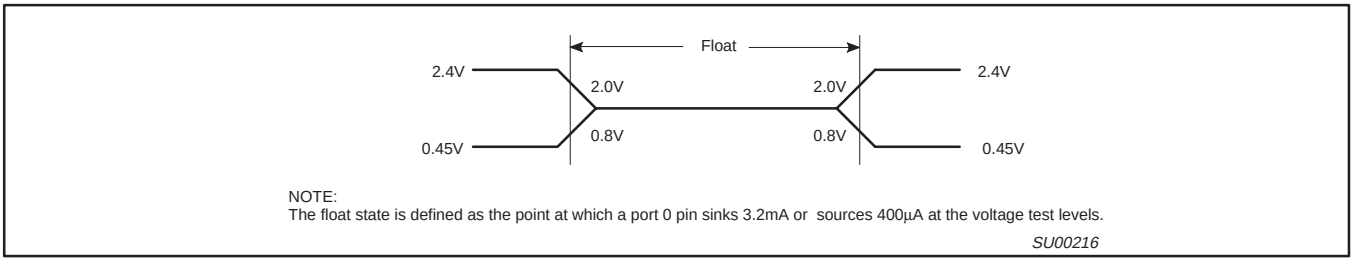

Figure 55. AC Testing Input, Float Waveform

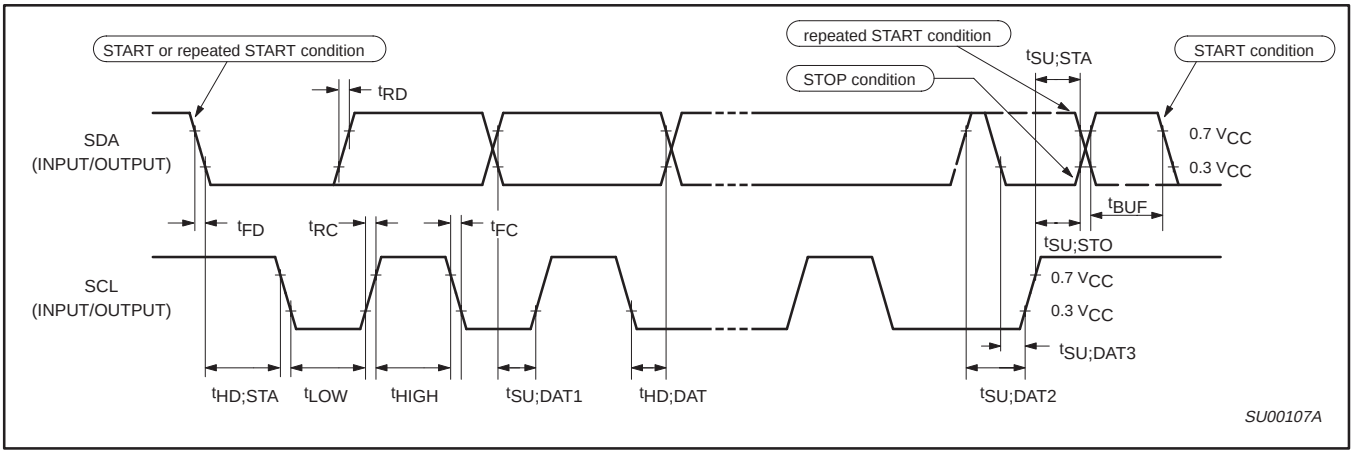

Figure 56. Timing SIO1 (I²C) Interface

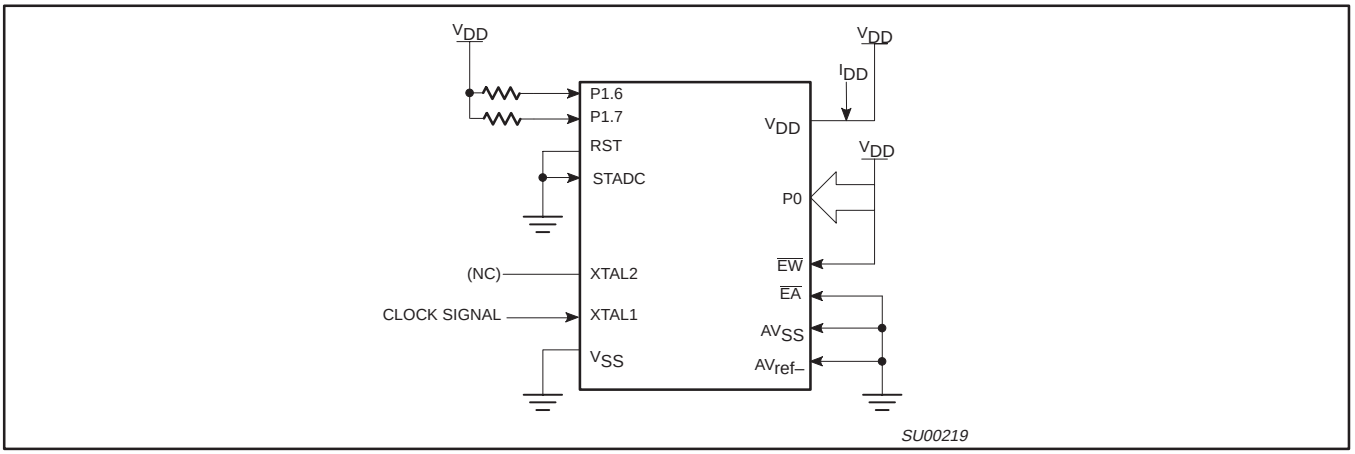

Figure 59. I_{DD} Test Condition, Idle Mode
All other pins are disconnected[2]

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.

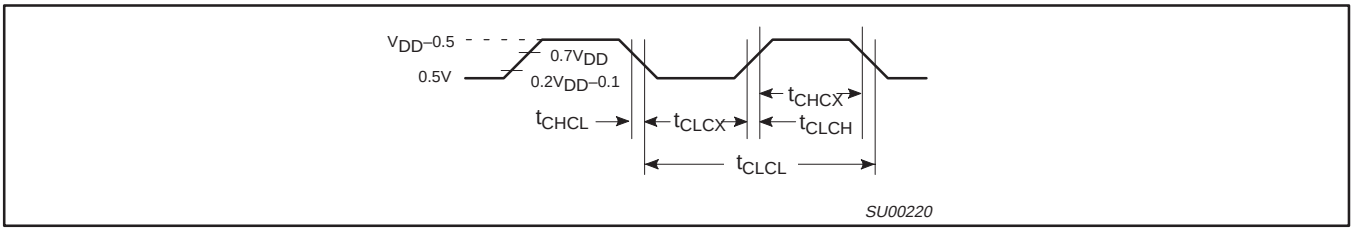

Figure 60. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes
$t_{CLCH} = t_{CHCL} = 5$ ns

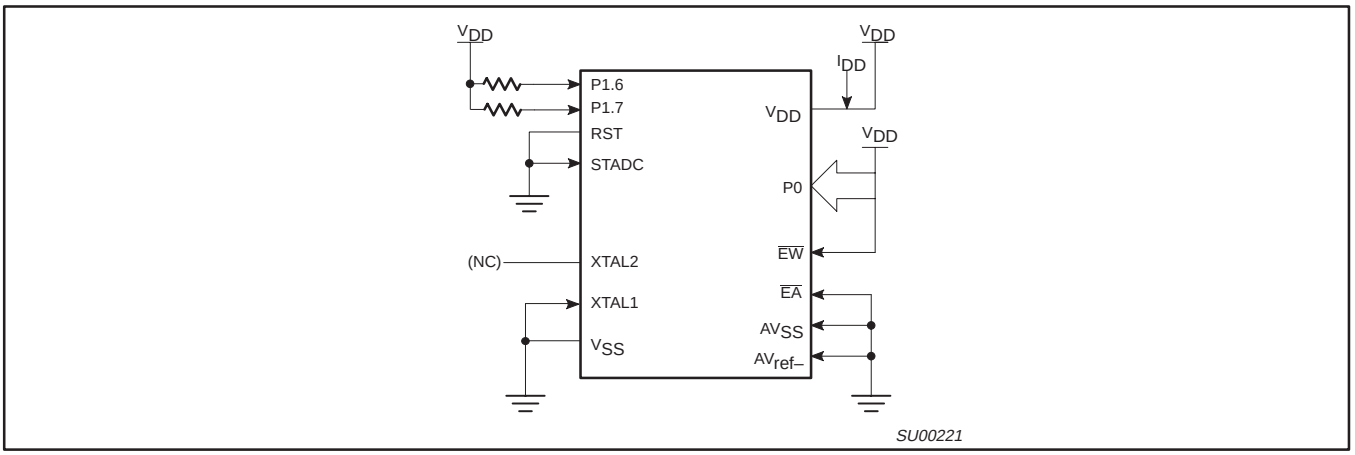

Figure 61. I_{DD} Test Condition, Power Down Mode
All other pins are disconnected. V_{DD} = 2 V to 5.5 V[3]

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.

Security Bits

With none of the security bits programmed the code in the program memory can be verified. When only security bit 1 (see Table 12) 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 12. 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.