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	-40°C ~ 85°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/p80c554sfbd-157

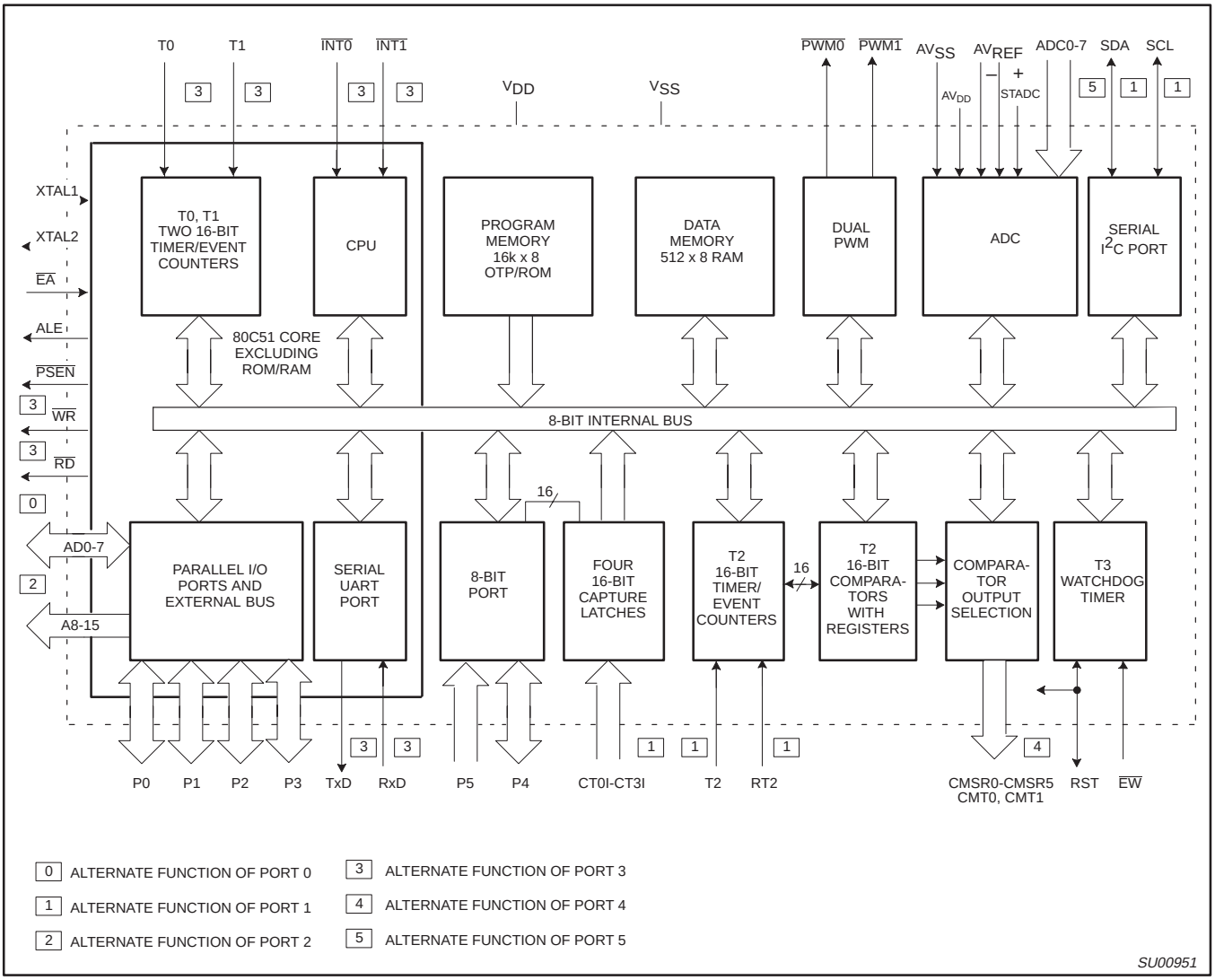
ORDERING INFORMATION

OTP/EPROM	ROMless	TEMPERATURE °C AND PACKAGE	FREQ. (MHz)	DRAWING NUMBER
P87C554SBBD	P80C554SBBD	0 to +70, Low Profile Quad Flat Package	16	SOT314–2
P87C554SFBD	P80C554SFBD	–40 to +85, Low Profile Quad Flat Package	16	SOT314–2

PART NUMBER DERIVATION

DEVICE NUMBER	OPERATING FREQUENCY MAX	TEMPERATURE RANGE	PACKAGE
P87C554 OTP P80C554 ROMless	S = 16 MHz	B= 0°C to 70°C F = –40°C to +85°C	BD=64L LQFP

BLOCK DIAGRAM



SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
P1M1	Port 1 output mode 1	92H									xx000000B
P1M2	Port 1 output mode 2	93H									xx000000B
P2M1	Port 2 output mode 1	94H									00H
P2M2	Port 2 output mode 2	95H									00H
P3M1	Port 3 output mode 1	9AH									00H
P3M2	Port 3 output mode 2	9BH									00H
P4M1	Port 4 output mode 1	9CH									00H
P4M2	Port 4 output mode 2	9DH									00H
PCON	Power control	87H	SMOD1	SMOD0	POF	WLE	GF1	GFO	PD	IDL	00x00000B
PSW	Program status word	D0H	CY	AC	FO	RS1	RS0	OV	F1	P	00H
PWMP#	PWM prescaler	FEH									00H
PWM1#	PWM register 1	FDH									00H
PWM0#	PWM register 0	FCH									00H
RTE#	Reset/toggle enable	EFH	TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40	00H
S0ADDR	Serial 0 slave address	F9H									00H
S0ADEN	Slave address mask	B9H									00H
S0BUF	Serial 0 data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
S0CON*	Serial 0 control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S1ADR#	Serial 1 address	DBH	SLAVE ADDRESS							GC	00H
SIDAT#	Serial 1 data	DAH									00H
S1STA#	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
SICON#*	Serial 1 control	D8H	CR2	ENS1	STA	ST0	SI	AA	CR1	CR0	00H
SP	Stack pointer	81H									07H
STE#	Set enable	EEH	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	C0H
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TL0	Timer low 0	8AH									00H
TMH2#	Timer high 2	EDH									00H
TML2#	Timer low 2	ECH									00H
TMOD	Timer mode	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TM2CON#	Timer 2 control	EAH	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
TM2IR#*	Timer 2 int flag reg	C8H	T20V	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	00H
T3#	Timer 3	FFH									00H

* SFRs are bit addressable.
\# SFRs are modified from or added to the 80C51 SFRs.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator, as shown in the logic symbol.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. The minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by either (1) externally holding the RST pin high for at least two machine cycles (12 oscillator periods) or (2) internally by an on-chip power-on detect (POD) circuit which detects V_{CC} ramping up from 0 V.

To insure a good external power-on reset, the RST pin must be high long enough for the oscillator to start up (normally a few milliseconds) plus two machine cycles. The voltage on V_{DD} and the RST pin must come up at the same time for a proper startup.

For a successful internal power-on reset, the V_{CC} voltage must ramp up from 0 V smoothly at a ramp rate greater than 5 V/100 ms.

The RST line can also be pulled HIGH internally by a pull-up transistor activated by the watchdog timer T3. The length of the output pulse from T3 is 3 machine cycles. A pulse of such short duration is necessary in order to recover from a processor or system fault as fast as possible.

Note that the short reset pulse from Timer T3 cannot discharge the power-on reset capacitor (see Figure 2). Consequently, when the watchdog timer is also used to set external devices, this capacitor arrangement should not be connected to the RST pin, and a different circuit should be used to perform the power-on reset operation. A timer T3 overflow, if enabled, will force a reset condition to the 8xC554 by an internal connection, independent of the level of the RST pin.

A reset may be performed in software by setting the software reset bit, SRST (AUXR1.5).

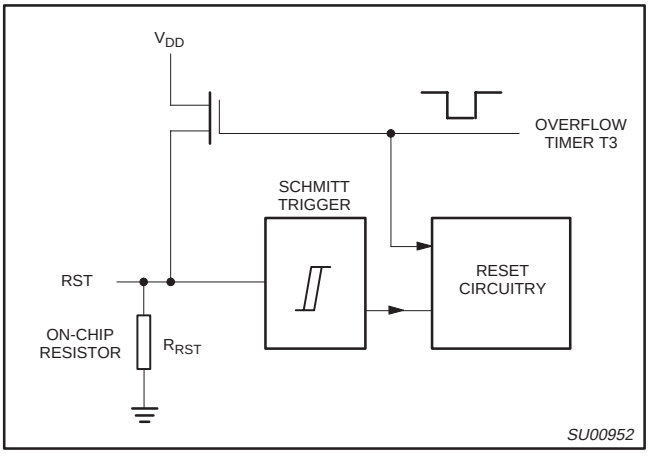


Figure 1. On-Chip Reset Configuration

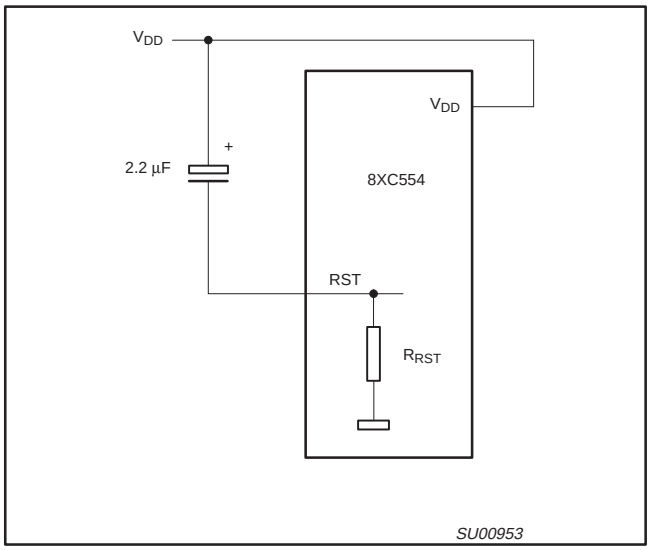


Figure 2. Power-On Reset

LOW POWER MODES

Stop Clock Mode

The static design enables the clock speed to be reduced down to 0 MHz (stopped). When the oscillator is stopped, the RAM and Special Function Registers retain their values. This mode allows step-by-step utilization and permits reduced system power consumption by lowering the clock frequency down to any value. For lowest power consumption the Power Down mode is suggested.

Idle Mode

In the idle mode (see Table 2), the CPU puts itself to sleep while some of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

Power-Down Mode

To save even more power, a Power Down mode (see Table 2) can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values down to 2.0 V and care must be taken to return V_{CC} to the minimum specified operating voltages before the Power Down Mode is terminated.

Either a hardware reset or external interrupt can be used to exit from Power Down. The Wake-up from Power-down bit, WUPD (AUXR1.3) must be set in order for an external interrupt to cause a wake-up from power-down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

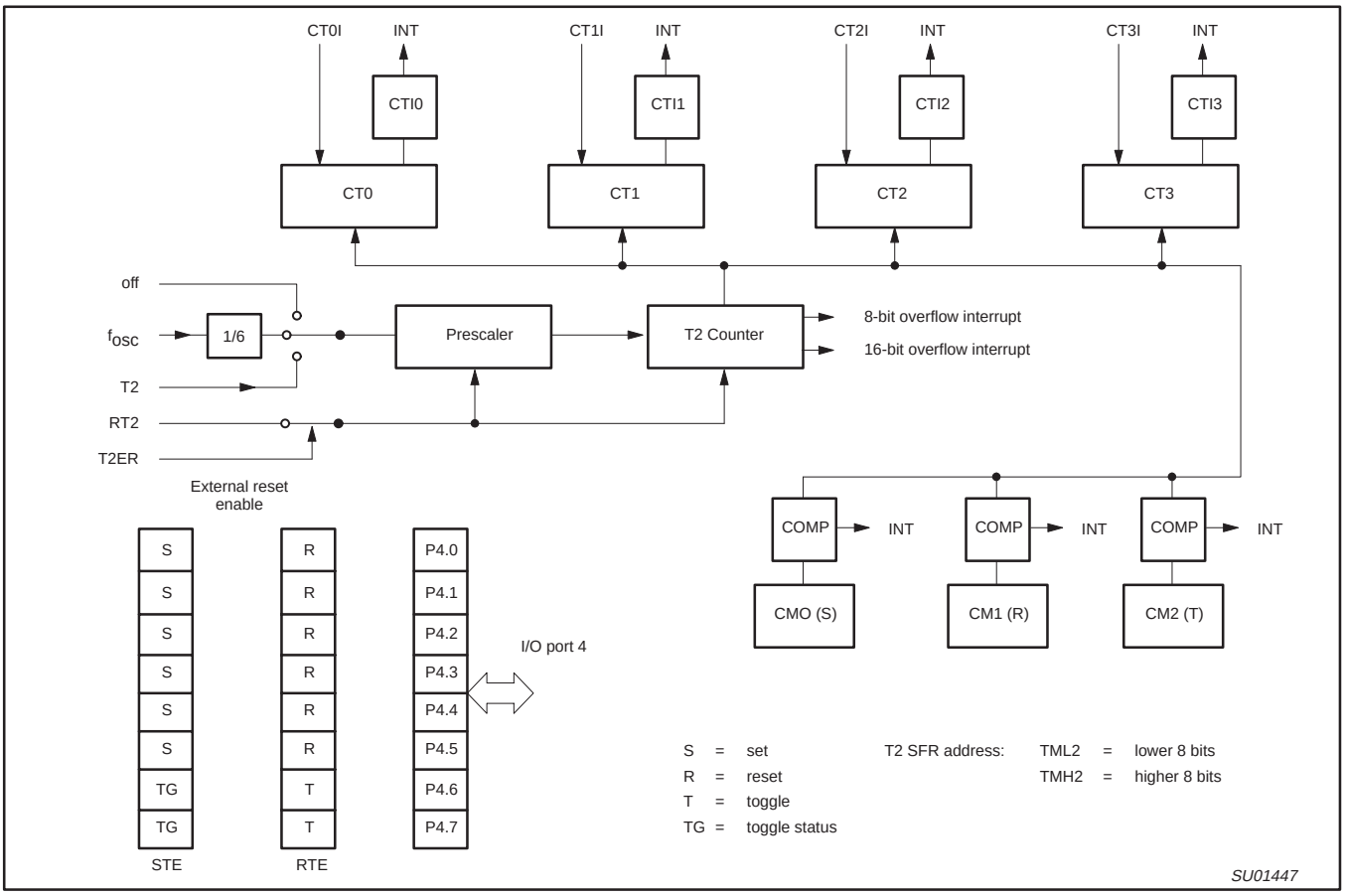


Figure 13. Block Diagram of Timer 2

Capture Logic: The four 16-bit capture registers that Timer T2 is connected to are: CT0, CT1, CT2, and CT3. These registers are loaded with the contents of Timer T2, and an interrupt is requested upon receipt of the input signals CT0I, CT1I, CT2I, or CT3I. These input signals are shared with port 1. The four interrupt flags are in the Timer T2 interrupt register (TM2IR special function register). If the capture facility is not required, these inputs can be regarded as additional external interrupt inputs.

Using the capture control register CTCON (see Figure 14), these inputs may capture on a rising edge, a falling edge, or on either a rising or falling edge. The inputs are sampled during S1P1 of each cycle. When a selected edge is detected, the contents of Timer T2 are captured at the end of the cycle.

Measuring Time Intervals Using Capture Registers: When a recurring external event is represented in the form of rising or falling edges on one of the four capture pins, the time between two events can be measured using Timer T2 and a capture register. When an event occurs, the contents of Timer T2 are copied into the relevant capture register and an interrupt request is generated. The interrupt service routine may then compute the interval time if it knows the previous contents of Timer T2 when the last event occurred. With a 12-MHz oscillator, Timer T2 can be programmed to overflow every 524 ms. When event interval times are shorter than this, computing the interval time is simple, and the interrupt service routine is short. For longer interval times, the Timer T2 extension routine may be used.

Compare Logic: Each time Timer T2 is incremented, the contents of the three 16-bit compare registers CM0, CM1, and CM2 are compared with the new counter value of Timer T2. When a match is found, the corresponding interrupt flag in TM2IR is set at the end of the following cycle. When a match with CM0 occurs, the controller sets bits 0-5 of port 4 if the corresponding bits of the set enable register STE are at logic 1.

During the early stages of software development/debugging, the watchdog may be disabled by tying the $\overline{EW}$ pin high. At a later stage, $\overline{EW}$ may be tied low to complete the debugging process.

Watchdog Software Example: The following example shows how watchdog operation might be handled in a user program.

```
;at the program start:

T3          EQU  0FFH ;address of watchdog timer T3
PCON        EQU  087H ;address of PCON SFR
WATCH-INTV  EQU  156  ;watchdog interval (e.g., 100 ms)

;to be inserted at each watchdog reload location within
;the user program:

LCALL WATCHDOG

;watchdog service routine:

WATCHDOG:  ORL  PCON,#10H  ;set condition flag (PCON.4)
           MOV T3,WATCH-INV ;load T3 with watchdog interval
           RET
```

If it is possible for this subroutine to be called in an erroneous state, then the condition flag WLE should be set at different parts of the main program.

Serial I/O

The 8xC554 is equipped with two independent serial ports: SIO0 and SIO1. SIO0 is a full duplex UART port and is similar to the Enhanced UART serial port. SIO1 accommodates the I²C bus.

SIO0: SIO0 is a full duplex serial I/O port identical to that of the Enhanced UART except Time 2 cannot be used as a baud rate generator. Its operation is the same, including the use of timer 1 as a baud rate generator.

Port 5 Operation

Port 5 may be used to input up to 8 analog signals to the ADC. Unused ADC inputs may be used to input digital inputs. These inputs have an inherent hysteresis to prevent the input logic from drawing excessive current from the power lines when driven by analog signals. Channel to channel crosstalk (Ct) should be taken into consideration when both analog and digital signals are simultaneously input to Port 5 (see, D.C. characteristics in data sheet).

Port 5 is not bidirectional and may not be configured as an output port. All six ports are multifunctional, and their alternate functions are listed in the Pin Descriptions section of this datasheet.

Pulse Width Modulated Outputs

The 8xC554 contains two pulse width modulated output channels (see Figure 19). These channels generate pulses of programmable length and interval. The repetition frequency is defined by an 8-bit prescaler PWMP, which supplies the clock for the counter. The prescaler and counter are common to both PWM channels. The 8-bit counter counts modulo 255, i.e., from 0 to 254 inclusive. The value of the 8-bit counter is compared to the contents of two registers: PWM0 and PWM1. Provided the contents of either of these registers is greater than the counter value, the corresponding $\overline{PWM0}$ or $\overline{PWM1}$ output is set LOW. If the contents of these registers are equal to, or less than the counter value, the output will be HIGH. The pulse-width-ratio is therefore defined by the contents of the registers PWM0 and PWM1. The pulse-width-ratio is in the range of 0 to 1 and may be programmed in increments of 1/255.

Buffered PWM outputs may be used to drive DC motors. The rotation speed of the motor would be proportional to the contents of PWMn. The PWM outputs may also be configured as a dual DAC. In this application, the PWM outputs must be integrated using conventional operational amplifier circuitry. If the resulting output voltages have to be accurate, external buffers with their own analog supply should be used to buffer the PWM outputs before they are integrated. The repetition frequency f_{PWM}, at the PWMn outputs is give by:

$$f_{PWM} = \frac{f_{OSC}}{(1 + PWMP) \times 255}$$

This gives a repetition frequency range of 246 Hz to 62.8 kHz (f_{OSC} = 16 MHz). At f_{OSC} = 24 MHz, the frequency range is 368 Hz to 83.4 Hz. By loading the PWM registers with either 00H or FFH, the PWM channels will output a constant HIGH or LOW level, respectively. Since the 8-bit counter counts modulo 255, it can never actually reach the value of the PWM registers when they are loaded with FFH.

When a compare register (PWM0 or PWM1) is loaded with a new value, the associated output is updated immediately. It does not have to wait until the end of the current counter period. Both $\overline{PWMn}$ output pins are driven by push-pull drivers. These pins are not used for any other purpose.

Prescaler frequency control register PWMP Reset Value = 00H

	7	6	5	4	3	2	1	0
PWMP (FEH)	MSB							LSB

PWMP.0-7 Prescaler division factor = PWMP + 1.

Reading PWMP gives the current reload value. The actual count of the prescaler cannot be read.

Reset Value = 00H

	7	6	5	4	3	2	1	0
PWM0 (FCH) PWM1 (FDH)	MSB							LSB

PWM0/1.0-7} Low/high ratio of $\overline{PWMn} = \frac{(PWMn)}{255 - (PWMn)}$

Analog-to-Digital Converter

The analog input circuitry consists of an 8-input analog multiplexer and a 10-bit, straight binary, successive approximation ADC. The A/D can also be operated in 8-bit mode with faster conversion times by setting bit ADC8 (AUXR1.7). The 8-bit results will be contained in the ADCH register. The analog reference voltage and analog power supplies are connected via separate input pins. For 10-bit accuracy, the conversion takes 50 machine cycles, i.e., 18.75 µs at an oscillator frequency of 16 MHz, 12.5 µs at an oscillator frequency of 24 MHz. For the 8-bit mode, the conversion takes 24 machine cycles. Input voltage swing is from 0 V to +5 V. Because the internal DAC employs a ratiometric potentiometer, there are no discontinuities in the converter characteristic. Figure 20 shows a functional diagram of the analog input circuitry.

The ADC has the option of either being powered off in idle mode for reduced power consumption or being active in idle mode for reducing internal noise during the conversion. This option is selected by the AIDL bit of AUXR1 register (AUXR1.6). With the AIDL bit set, the ADC is active in the idle mode, and with the AIDL bit cleared, the ADC is powered off in idle mode.

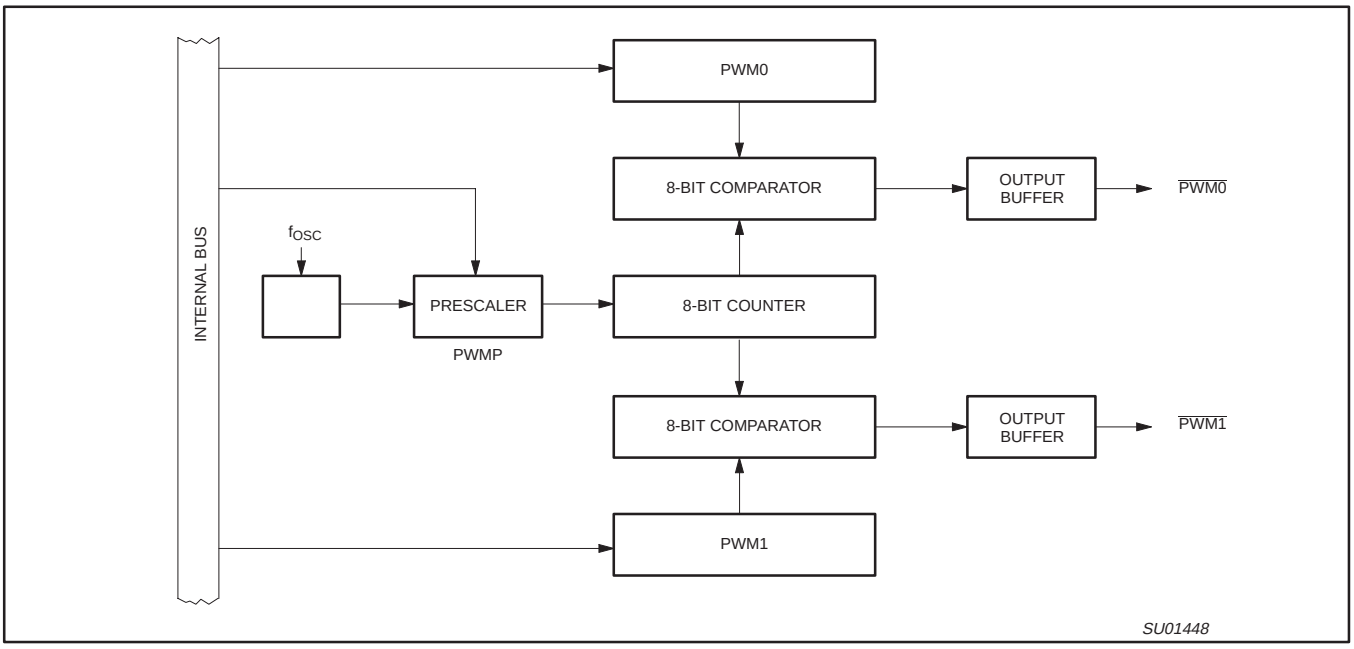


Figure 19. Functional Diagram of Pulse Width Modulated Outputs

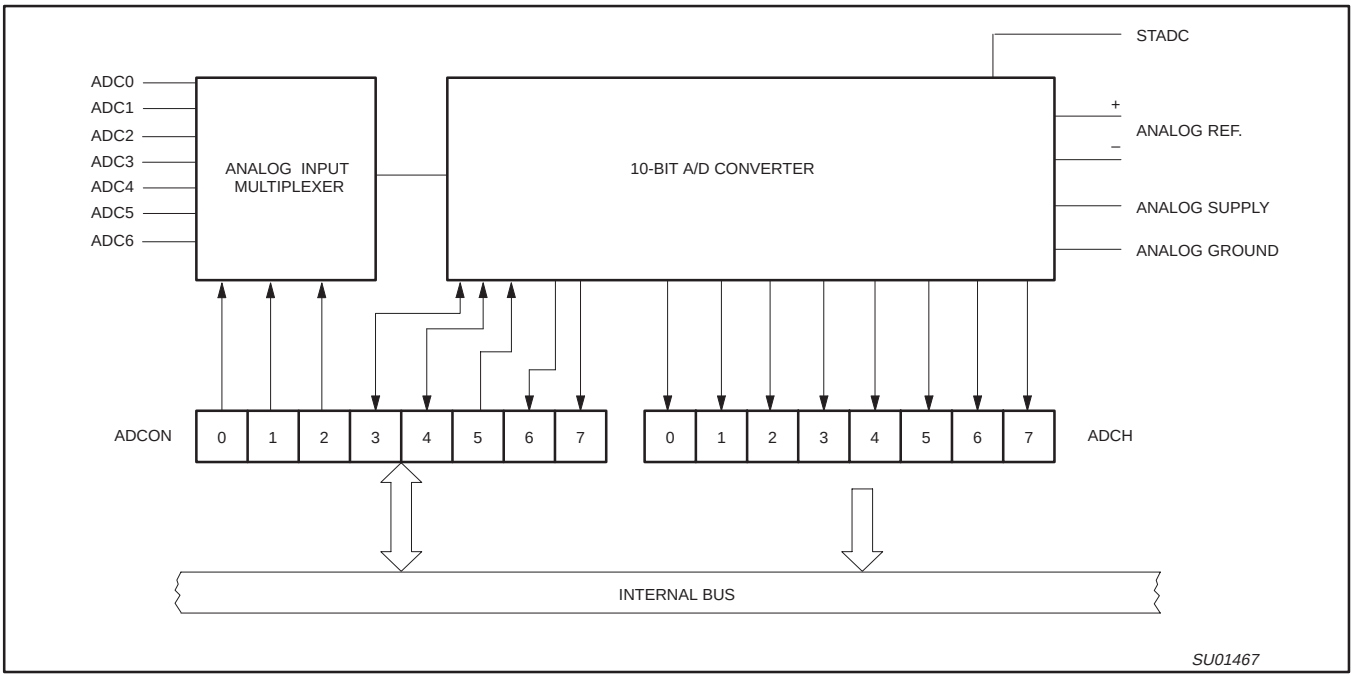


Figure 20. Functional Diagram of Analog Input Circuitry

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)

If the STA and STO bits are both set, the a STOP condition is transmitted to the I2C bus if SIO1 is in a master mode (in a slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.

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

SI, THE SERIAL INTERRUPT FLAG

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

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

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

AA, THE ASSERT ACKNOWLEDGE FLAG

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

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

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

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

When SIO1 is in the addressed slave transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 43). When SI is cleared, SIO1 leaves state C8H, enters the not addressed slave receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I2C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part's own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

CR0, CR1, AND CR2, THE CLOCK RATE BITS

These three bits determine the serial clock frequency when SIO1 is in a master mode. The various serial rates are shown in Table 5.

A 12.5 kHz bit rate may be used by devices that interface to the I2C bus via standard I/O port lines which are software driven and slow. 100 kHz is usually the maximum bit rate and can be derived from a 8 MHz, 6 MHz, or a 3-MHz oscillator. A variable bit rate (0.24 kHz to 62.5 kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a master mode.

The frequencies shown in Table 5 are unimportant when SIO1 is in a slave mode. In the slave modes, SIO1 will automatically synchronize with any clock frequency up to 100 kHz.

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

Table 5. Serial Clock Rates

CR2	CR1	CR0	BIT FREQUENCY (kHz) AT f_{OSC} 3 MHz	6 MHz	8 MHz	12 MHz[2]	15 MHz[2]	f_{OSC} DIVIDED BY
0	0	0	23	47	62.5	94	117[1]	128
0	0	1	27	54	71	107[1]	134[1]	112
0	1	0	31	63	83.3	125[1]	156[1]	96
0	1	1	37	75	100	150[1]	188[1]	80
1	0	0	6.25	12.5	17	25	31	48
1	0	1	50	100	133[1]	200[1]	250[1]	60
1	1	0	100	200	267[1]	400[1]	500[1]	30
1	1	1	0.24 < 62.5 0 < 255	0.49 < 62.5 0 < 254	0.65 < 55.6 0 < 253	0.98 < 50.0 0 < 251	1.22 < 52.1 0 < 250	48 × (256 – (reload value Timer 1)) Reload value Timer 1 in Mode 2.

NOTES:
1. These frequencies exceed the upper limit of 100 kHz of the I2C-bus specification and cannot be used in an I2C-bus application.
2. At f_{OSC} = 12 MHz/15 MHz the maximum I2C bus rate of 100 kHz cannot be realized due to the fixed divider rates.

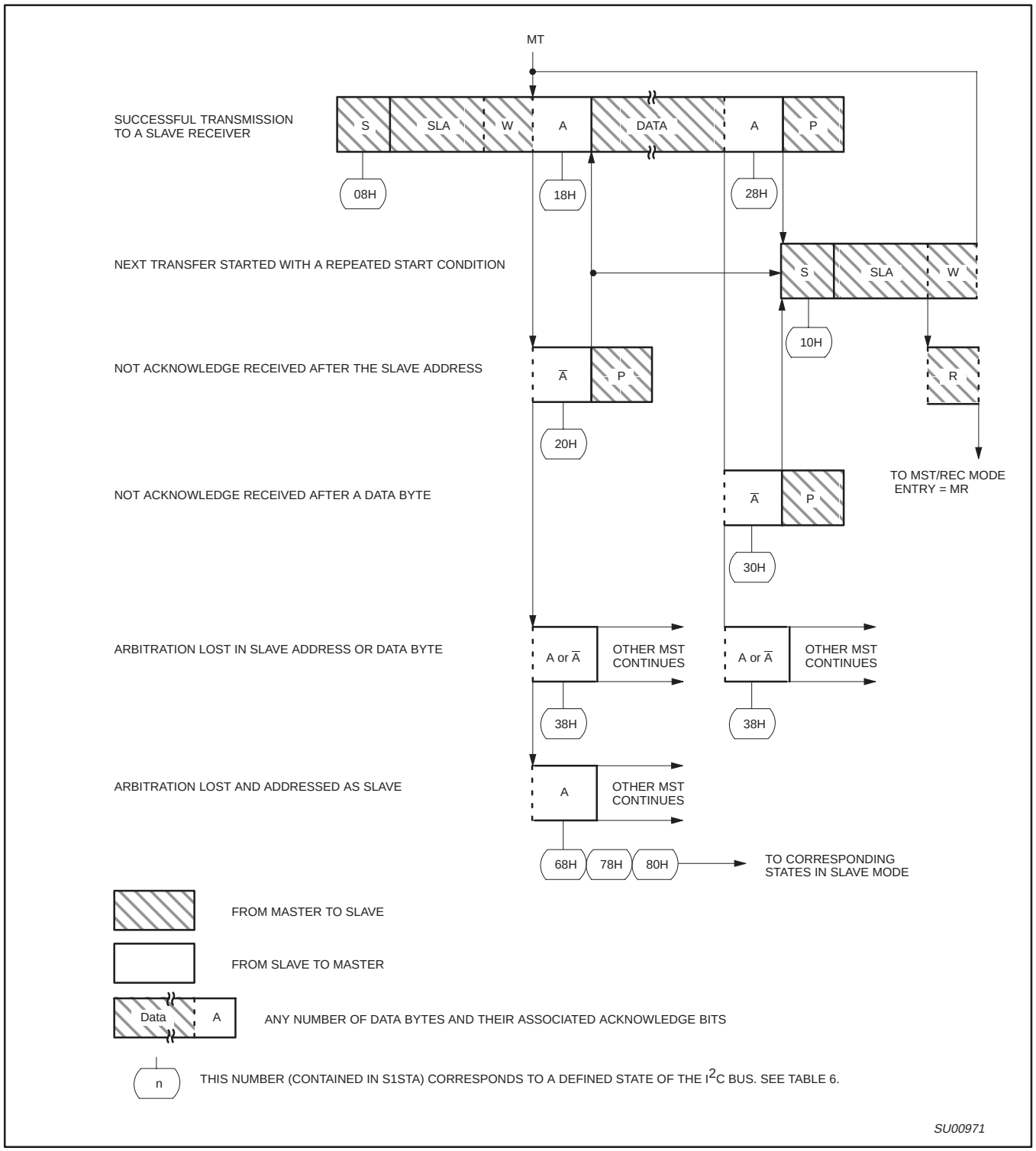


Figure 40. Format and States in the Master Transmitter Mode

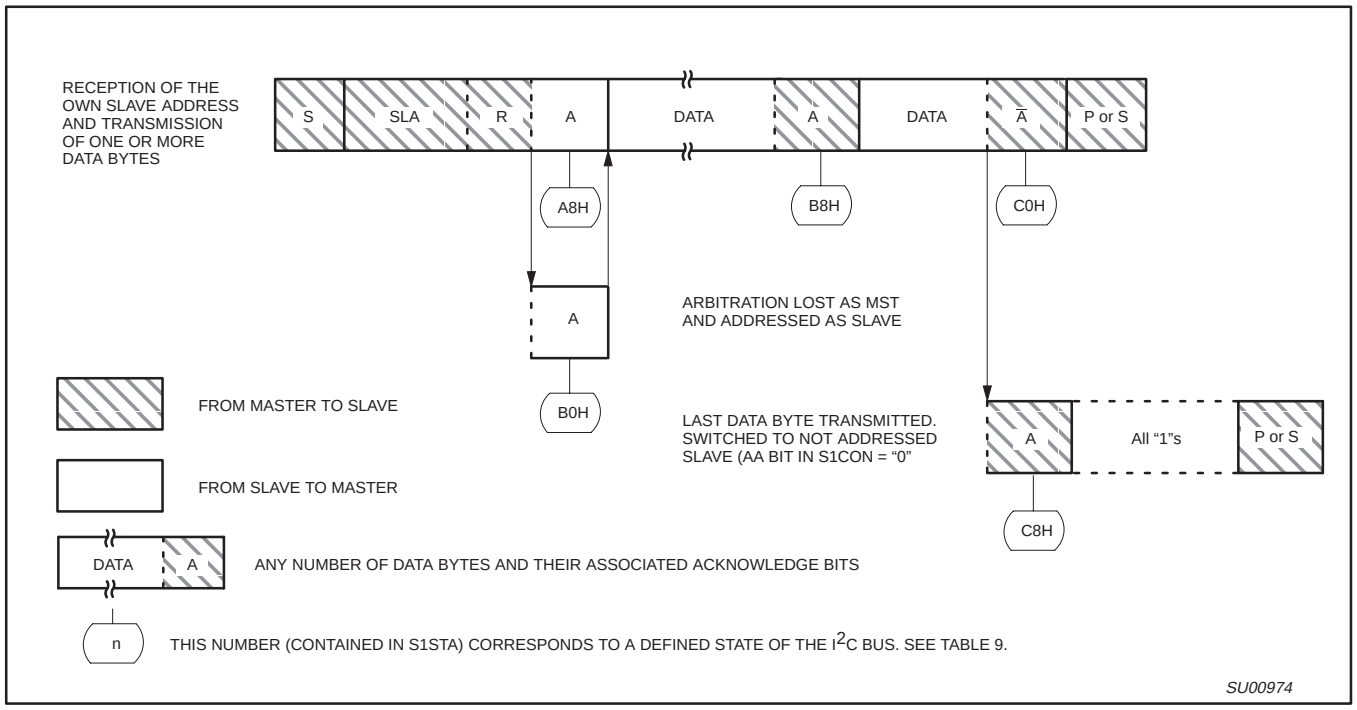


Figure 43. Format and States of the Slave Transmitter Mode

Table 7. Master 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	
08H	A START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or	X	0	0	X	As above
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No S1DAT action or	0	0	0	X	I²C bus will be released; SIO1 will enter a slave mode
		No S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No S1DAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		no S1DAT action	0	0	0	1	Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No S1DAT action or	1	0	0	X	Repeated START condition will be transmitted
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
50H	Data byte has been received; ACK has been returned	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset

Table 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).

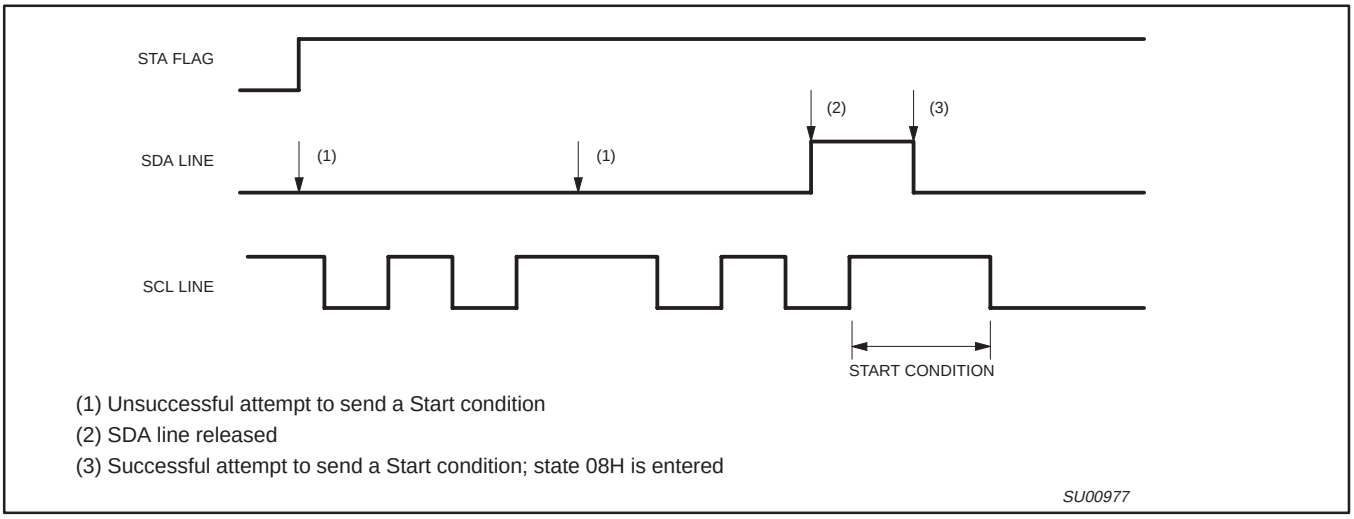


Figure 46. Recovering from a Bus Obstruction Caused by a Low Level on SDA

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    : 20, SLA+W have been transmitted, NOT ACK has been received
                        ! ACTION   : Transmit STOP condition.
                        !-------------------------------------------------------
                        .sect      mts20
                        .base      0x120

0120  75D8D5                                  mov   S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                    ! set STO, clr SI
0123  D0D0                                    pop   psw
0125  32                                      reti

                        !-------------------------------------------------------
                        ! STATE    : 28, DATA of S1DAT have been transmitted, ACK received.
                        ! ACTION   : If Transmitted DATA is last DATA then transmit a STOP condition,
                        !            else transmit next DATA.
                        !-------------------------------------------------------
                        .sect      mts28
                        .base      0x128

0128  D55285                                  djnz  NUMBYTMST,NOTLDAT1      ! JMP if NOT last DATA
012B  75D8D5                                  mov   S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                    ! clr SI, set AA
012E  01B9                                    ajmp  RETmt

                        .sect      mts28sb
                        .base      0x0b0
00B0  75D018    NOTLDAT1:                     mov   psw,#SELRB3
00B3  87DA                                    mov   S1DAT,@r1
00B5  75D8C5    CON:                          mov   S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                    ! clr SI, set  AA
00B8  09                                      inc   r1
00B9  D0D0      RETmt     :                   pop   psw
00BB  32                                      reti
                        !-------------------------------------------------------
                        ! STATE    : 30, DATA of S1DAT have been transmitted, NOT ACK received.
                        ! ACTION   : Transmit a STOP condition.
                        !-------------------------------------------------------
                        .sect      mts30
                        .base      0x130

0130  75D8D5                                  mov   S1CON,#ENS1_NOTSTA_STO_NOTSI_AA_CR0
                                                                    ! set STO, clr SI
0133  D0D0                                    pop   psw
0135  32                                      reti

                        !-------------------------------------------------------
                        ! STATE    : 38, Arbitration lost in SLA+W or DATA.
                        ! ACTION   : Bus is released, not addressed SLV mode is entered.
                        !            A new START condition is transmitted when the IIC bus is free again.
                        !-------------------------------------------------------
                        .sect      mts38
                        .base      0x138

0138  75D8E5                                  mov   S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
013B  855352                                  mov   NUMBYTMST,BACKUP
013E  01B9                                    ajmp  RETmt
```

```
                        !*************************************************************************************************
                        !*************************************************************************************************
                        ! SLAVE RECEIVER STATE SERVICE ROUTINES
                        !*************************************************************************************************
                        !*************************************************************************************************

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

00D0    7840            INITSRD:            mov   r0,#SRD
00D2    7908                                mov   r1,#8
00D4    D0D0                                pop   psw
00D6    32                                  reti

                        !-------------------------------------------------------------
                        ! STATE    : 68, Arbitration lost in SLA and R/W as MST
                        !            Own SLA+W have been received, ACK returned
                        ! ACTION   : DATA will be received and ACK returned.
                        !            STA is set to restart MST mode after the bus is free again.
                        !-------------------------------------------------------------
                        .sect      srs68
                        .base      0x168
0168    75D8E5                              mov   S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
016B    75D018                              mov   psw,#SELRB3
016E    01D0                                ajmp  INITSRD

                        !-------------------------------------------------------------
                        ! STATE    : 70, General call has been received, ACK returned.
                        ! ACTION   : DATA will be received and ACK returned.
                        !-------------------------------------------------------------
                        .sect      srs70
                        .base      0x170
0170    75D8C5                              mov   S1CON,#ENS1_NOTSTA_NOTSTO_NOTSI_AA_CR0
                                                                          ! clr SI, set AA
0173    75D018                              mov   psw,#SELRB3             ! Initialize SRD counter
0176    01D0                                ajmp  initsrd

                        !-------------------------------------------------------------
                        ! STATE    : 78, Arbitration lost in SLA+R/W as MST.
                        !            General call has been received, ACK returned.
                        ! ACTION   : DATA will be received and ACK returned.
                        !            STA is set to restart MST mode after the bus is free again.
                        !-------------------------------------------------------------
                        .sect      srs78
                        .base      0x178
0178    75D8E5                              mov   S1CON,#ENS1_STA_NOTSTO_NOTSI_AA_CR0
017B    75D018                              mov   psw,#SELRB3             ! Initialize SRD counter
017E    01D0                                ajmp  INITSRD
```

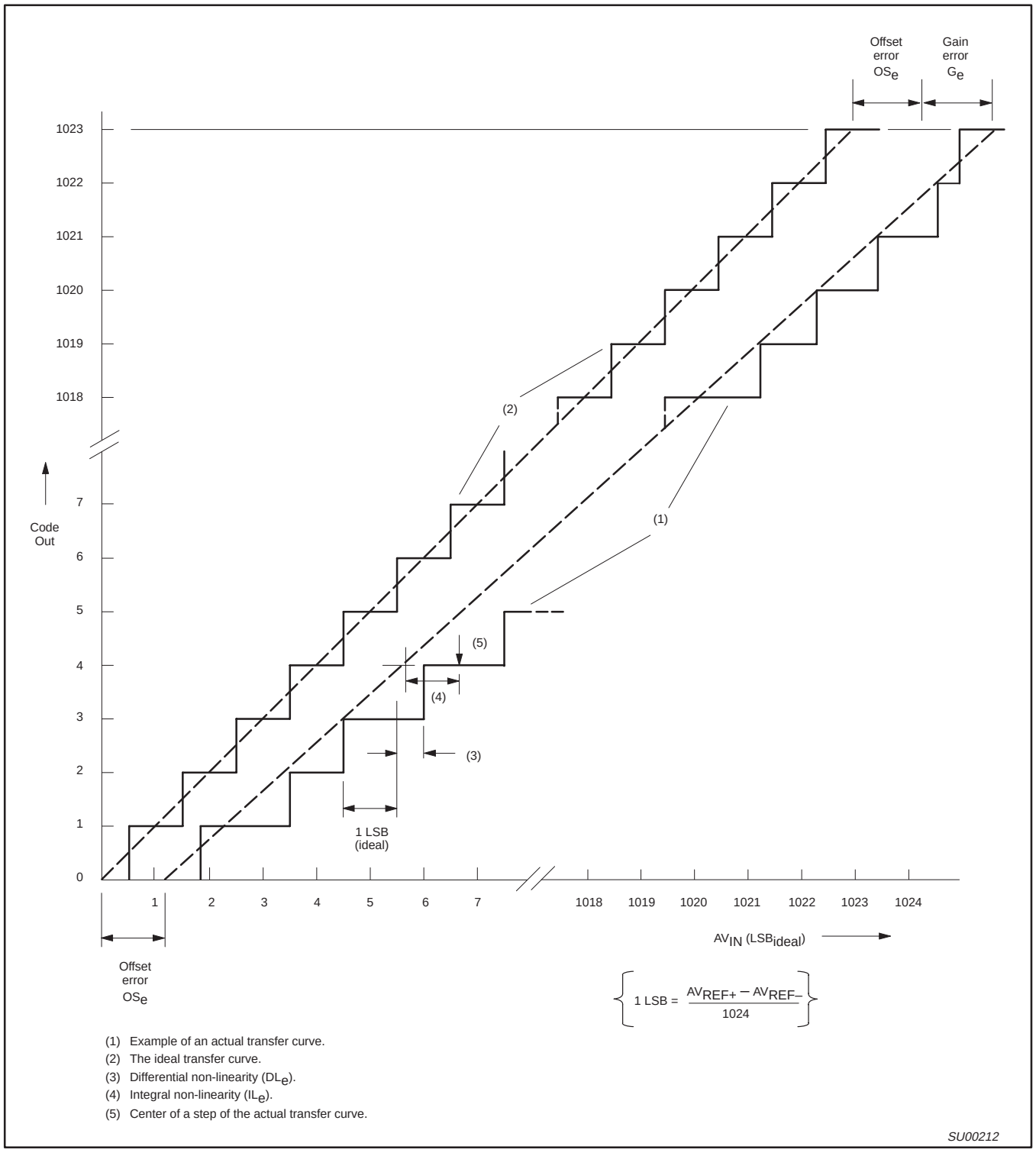


Figure 48. ADC Conversion Characteristic

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.

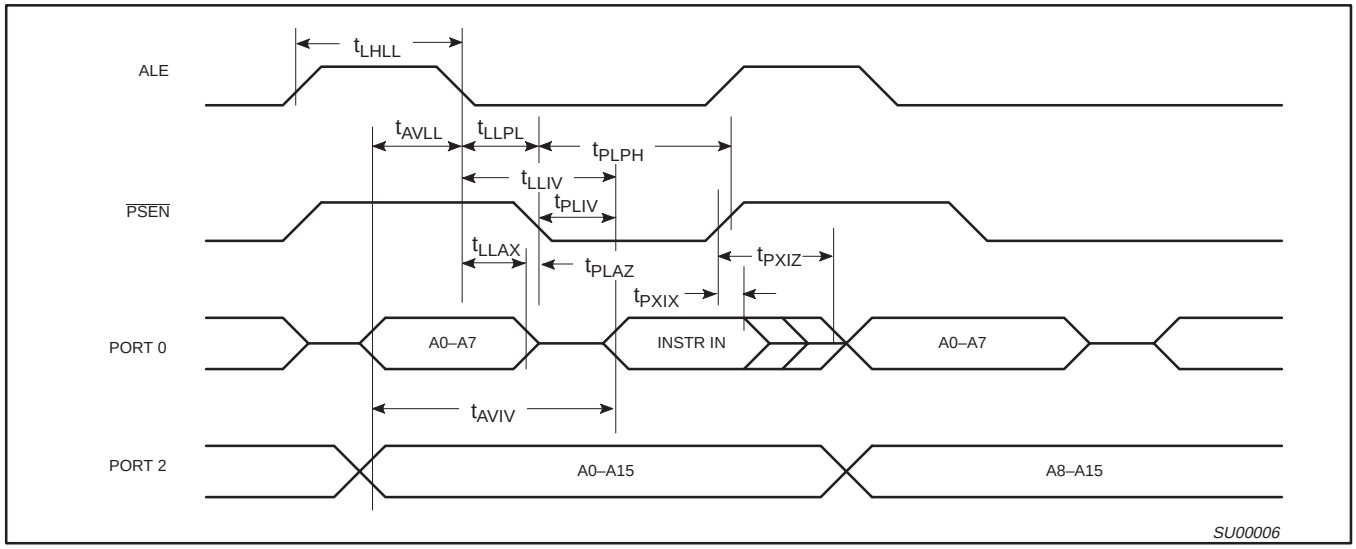


Figure 49. External Program Memory Read Cycle

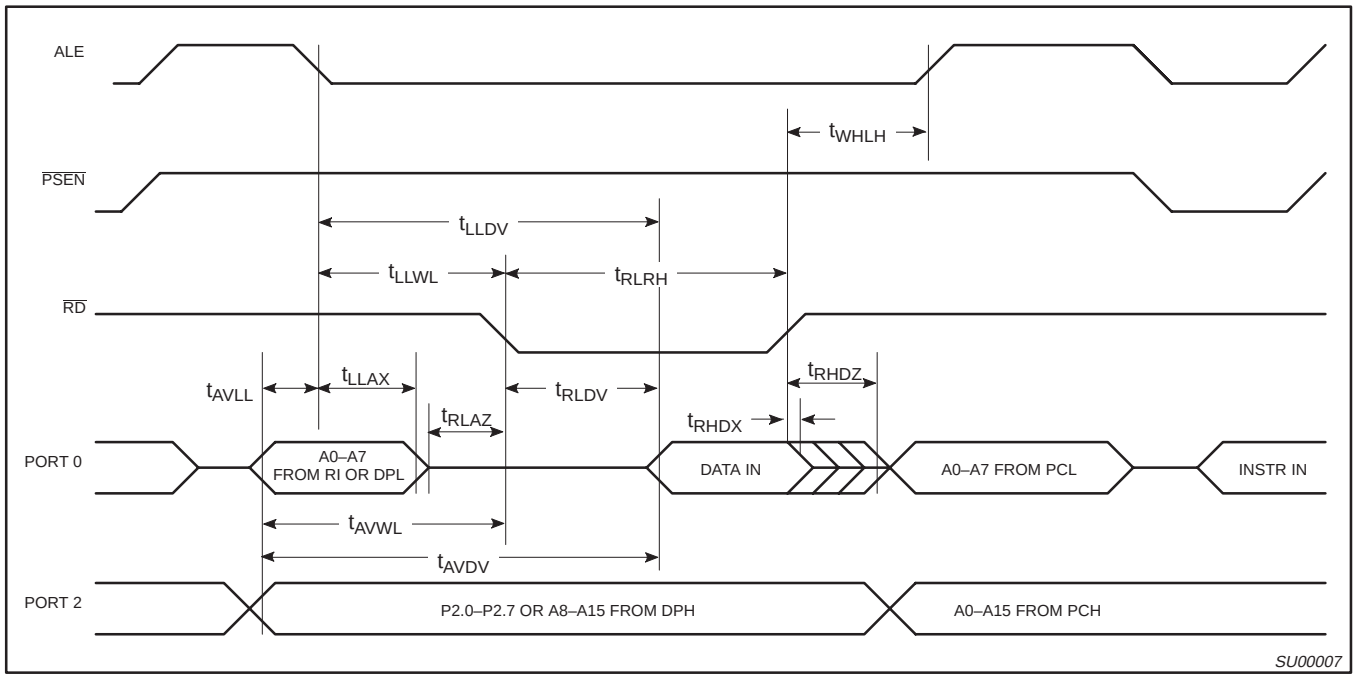


Figure 50. External Data Memory Read Cycle

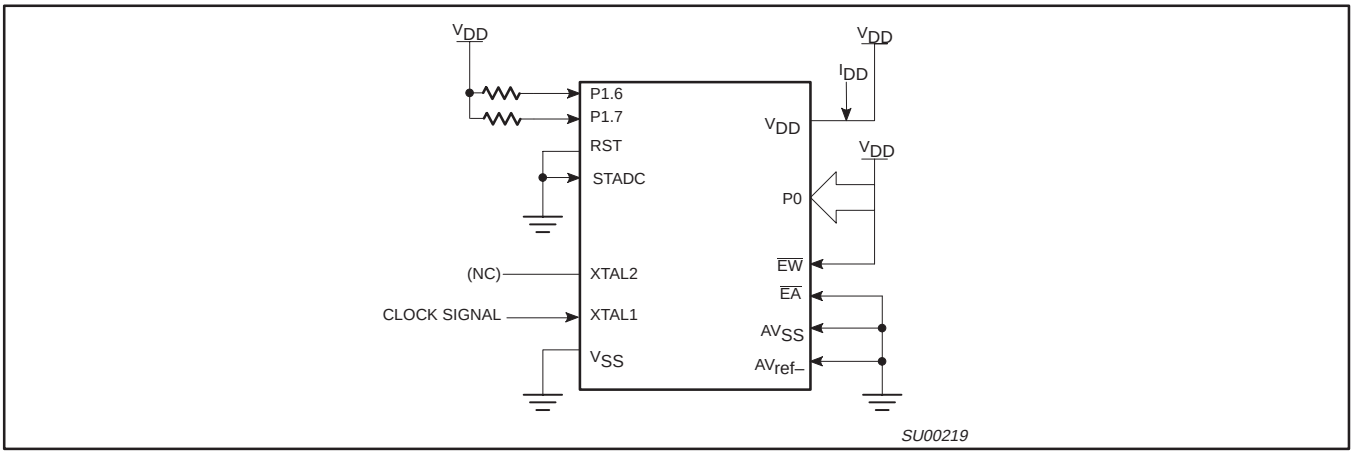


Figure 59. I_{DD} Test Condition, Idle Mode
All other pins are disconnected[2]

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.

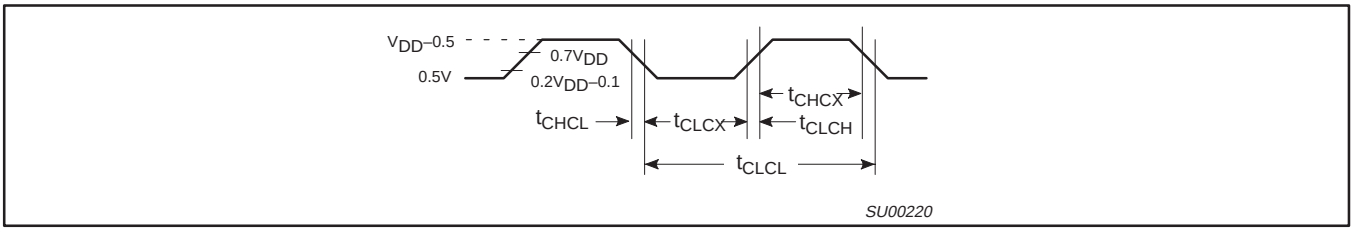


Figure 60. Clock Signal Waveform for I_{DD} Tests in Active and Idle Modes
$t_{CLCH} = t_{CHCL} = 5$ ns

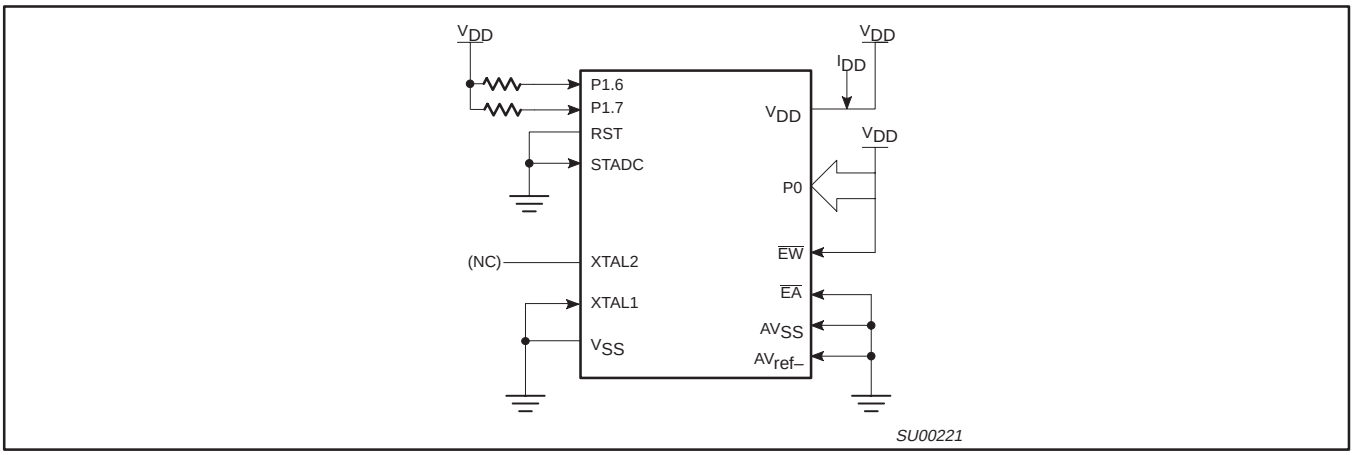


Figure 61. I_{DD} Test Condition, Power Down Mode
All other pins are disconnected. V_{DD} = 2 V to 5.5 V[3]

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.

REVISION HISTORY

Rev	Date	Description
_5	20030128	**Product data (9397 750 11006); ECN 853-2408 29338 of 07 January 2003** Modifications: ● References to ROM (83) devices removed
_4	20001110	**Preliminary data (9397 750 07505); previous release**