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.

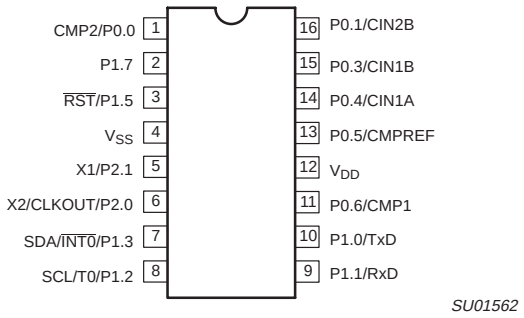
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, UART/USART
Peripherals	Brown-out Detect/Reset, LED, POR, WDT
Number of I/O	14
Program Memory Size	2KB (2K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p87lpc761bdh-112

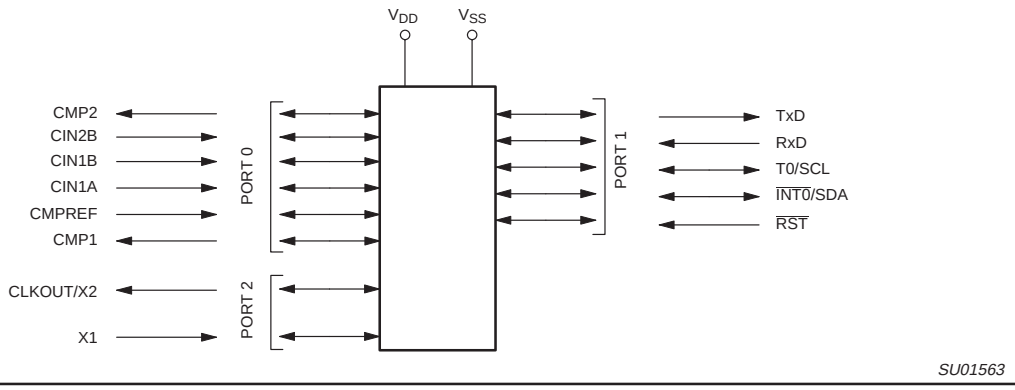
Low power, low price, low pin count (16 pin) microcontroller with 2 kbyte OTP

P87LPC761

PIN CONFIGURATION, 16-PIN DIP AND TSSOP PACKAGES



LOGIC SYMBOL



PIN DESCRIPTIONS

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION		
P0.0–P0.1 P0.3–P0.6	1, 11, 13–16	I/O	**Port 0**: Port 0 is an 6-bit I/O port with a user-configurable output type. Port 0 latches are configured in the quasi-bidirectional mode and have either ones or zeros written to them during reset, as determined by the PRHI bit in the UCFG1 configuration byte. The operation of port 0 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. The Keyboard Interrupt feature operates with port 0 pins. Port 0 also provides various special functions as described below.		
	1	O	**P0.0**	**CMP2**	Comparator 2 output.
	16	I	**P0.1**	**CIN2B**	Comparator 2 positive input B.
	15	I	**P0.3**	**CIN1B**	Comparator 1 positive input B.
	14	I	**P0.4**	**CIN1A**	Comparator 1 positive input A.
	13	I	**P0.5**	**CMPREF**	Comparator reference (negative) input.
	11	O	**P0.6**	**CMP1**	Comparator 1 output.
P1.0–P1.3 P1.5–P1.7	2–3, 7–10	I/O	**Port 1**: Port 1 is an 8-bit I/O port with a user-configurable output type, except for three pins as noted below. Port 1 latches are configured in the quasi-bidirectional mode and have either ones or zeros written to them during reset, as determined by the PRHI bit in the UCFG1 configuration byte. The operation of the configurable port 1 pins as inputs and outputs depends upon the port configuration selected. Each of the configurable port pins are programmed independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. Port 1 also provides various special functions as described below.		
	10	O	**P1.0**	**TxD**	Transmitter output for the serial port.
	9	I	**P1.1**	**RxD**	Receiver input for the serial port.
	8	I/O	**P1.2**	**T0**	Timer/counter 0 external count input or overflow output.
		I/O		**SCL**	I^2C serial clock input/output. When configured as an output, P1.2 is open drain, in order to conform to I^2C specifications.
	7	I	**P1.3**	**$\overline{\text{INT0}}$**	External interrupt 0 input.
		I/O		**SDA**	I^2C serial data input/output. When configured as an output, P1.3 is open drain, in order to conform to I^2C specifications.
	3	I	**P1.5**	**$\overline{\text{RST}}$**	External Reset input (if selected via EPROM configuration). A low on this pin resets the microcontroller, causing I/O ports and peripherals to take on their default states, and the processor begins execution at address 0. When used as a port pin, P1.5 is a Schmitt trigger input only.
P2.0–P2.1	5, 6	I/O	**Port 2**: Port 2 is a 2-bit I/O port with a user-configurable output type. Port 2 latches are configured in the quasi-bidirectional mode and have either ones or zeros written to them during reset, as determined by the PRHI bit in the UCFG1 configuration byte. The operation of port 2 pins as inputs and outputs depends upon the port configuration selected. Each port pin is configured independently. Refer to the section on I/O port configuration and the DC Electrical Characteristics for details. Port 2 also provides various special functions as described below.		
	6	O	**P2.0**	**X2**	Output from the oscillator amplifier (when a crystal oscillator option is selected via the EPROM configuration).
				CLKOUT	CPU clock divided by 6 clock output when enabled via SFR bit and in conjunction with internal RC oscillator or external clock input.
	5	I	**P2.1**	**X1**	Input to the oscillator circuit and internal clock generator circuits (when selected via the EPROM configuration).
V_{SS}	4	I	**Ground**: 0V reference.		
V_{DD}	12	I	**Power Supply**: This is the power supply voltage for normal operation as well as Idle and Power Down modes.		

Name	Description	SFR Address	Bit Functions and Addresses MSB							LSB	Reset Value
KBI#	Keyboard Interrupt	86h									00h
			87	86	85	84	83	82	81	80	
P0*	Port 0	80h	–	CMP1	CMPREF	CIN1A	CIN1B	–	CIN2B	CMP2	Note 2
			97	96	95	94	93	92	91	90	
P1*	Port 1	90h	(P1.7)	–	$\overline{RST}$	–	$\overline{INT0}$	T0	RxD	TxD	Note 2
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0h	–	–	–	–	–	–	X1	X2	Note 2
P0M1#	Port 0 output mode 1	84h	–	(P0M1.6)	(P0M1.5)	(P0M1.4)	(P0M1.3)	–	(P0M1.1)	(P0M1.0)	00h
P0M2#	Port 0 output mode 2	85h	–	(P0M2.6)	(P0M2.5)	(P0M2.4)	(P0M2.3)	–	(P0M2.1)	(P0M2.0)	00H
P1M1#	Port 1 output mode 1	91h	(P1M1.7)	–	–	–	–	–	(P1M1.1)	(P1M1.0)	00h[1]
P1M2#	Port 1 output mode 2	92h	(P1M2.7)	–	–	–	–	–	(P1M2.1)	(P1M2.0)	00h[1]
P2M1#	Port 2 output mode 1	A4h	P2S	P1S	P0S	ENCLK	–	T0OE	(P2M1.1)	(P2M1.0)	00h
P2M2#	Port 2 output mode 2	A5h	–	–	–	–	–	–	(P2M2.1)	(P2M2.0)	00h[1]
PCON	Power control register	87h	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL	Note 3
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0h	CY	AC	F0	RS1	RS0	OV	F1	P	00h
PT0AD#	Port 0 digital input disable	F6h									00h
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial port control	98h	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00h
SBUF	Serial port data buffer register	99h									xxh
SADDR#	Serial port address register	A9h									00h
SADEN#	Serial port address enable	B9h									00h
SP	Stack pointer	81h									07h
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer 0 and 1 control	88h	TF1	TR1	TF0	TR0	–	–	IE0	IT0	00h
TH0	Timer 0 high byte	8Ch									00h
TH1	Timer 1 high byte	8Dh									00h
TL0	Timer 0 low byte	8Ah									00h
TL1	Timer 1 low byte	8Bh									00h

Internal Reference Voltage
An internal reference voltage generator may supply a default reference when a single comparator input pin is used. The value of the internal reference voltage, referred to as V_{ref}, is 1.23 V ±10%.

Comparator Interrupt
Each comparator has an interrupt flag CMFn contained in its configuration register. This flag is set whenever the comparator output changes state. The flag may be polled by software or may be used to generate an interrupt. The interrupt will be generated when the corresponding enable bit ECn in the IEN1 register is set and the interrupt system is enabled via the EA bit in the IEN0 register.

Comparators and Power Reduction Modes
Either or both comparators may remain enabled when Power Down or Idle mode is activated. The comparators will continue to function in the power reduction mode. If a comparator interrupt is enabled, a change of the comparator output state will generate an interrupt and wake up the processor. If the comparator output to a pin is enabled, the pin should be configured in the push-pull mode in order to obtain fast switching times while in power down mode. The reason is that with the oscillator stopped, the temporary strong pull-up that normally occurs during switching on a quasi-bidirectional port pin does not take place.

Comparators consume power in Power Down and Idle modes, as well as in the normal operating mode. This fact should be taken into account when system power consumption is an issue.

Comparator Configuration Example
The code shown in Figure 5 is an example of initializing one comparator. Comparator 1 is configured to use the CIN1A and CMPREF inputs, outputs the comparator result to the CMP1 pin, and generates an interrupt when the comparator output changes.

The interrupt routine used for the comparator must clear the interrupt flag (CMF1 in this case) before returning.

```
CmpInit:
    mov     PT0AD,#30h      ; Disable digital inputs on pins that are used
                            ;   for analog functions: CIN1A, CMPREF.
    anl     P0M2,#0cfh      ; Disable digital outputs on pins that are used
    orl     P0M1,#30h       ;   for analog functions: CIN1A, CMPREF.
    mov     CMP1,#24h       ; Turn on comparator 1 and set up for:
                            ;   – Positive input on CIN1A.
                            ;   – Negative input from CMPREF pin.
                            ;   – Output to CMP1 pin enabled.
    call    delay10us       ; The comparator has to start up for at
                            ;   least 10 microseconds before use.
    anl     CMP1,#0feh      ; Clear comparator 1 interrupt flag.
    setb    EC1             ; Enable the comparator 1 interrupt. The
                            ;   priority is left at the current value.
    setb    EA              ; Enable the interrupt system (if needed).
    ret                     ; Return to caller.
```

SU01189

Figure 5.

I²C Serial Interface

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Bidirectional data transfer between masters and slaves.
- Serial addressing of slaves (no added wiring).
- Acknowledgment after each transferred byte.
- Multimaster bus.
- Arbitration between simultaneously transmitting masters without corruption of serial data on bus.

The I²C subsystem includes hardware to simplify the software required to drive the I²C bus. The hardware is a single bit interface which in addition to including the necessary arbitration and framing error checks, includes clock stretching and a bus timeout timer. The interface is synchronized to software either through polled loops or interrupts.

Refer to the application note AN422, entitled “Using the 8XC751 Microcontroller as an I²C Bus Master” for additional discussion of the 8xC76x I²C interface and sample driver routines.

The P87LPC761 I²C implementation duplicates that of the 87C751 and 87C752 except for the following details:

- The interrupt vector addresses for both the I²C interrupt and the Timer I interrupt.
- The I²C SFR addresses (I2CON, I2CFG, I2DAT).
- The location of the I²C interrupt enable bit and the name of the SFR it is located within (EI2 is Bit 0 in IEN1).
- The location of the Timer I interrupt enable bit and the name of the SFR it is located within (ETI is Bit 7 in IEN1).
- The I²C and Timer I interrupts have a settable priority.

Timer I is used to both control the timing of the I²C bus and also to detect a “bus locked” condition, by causing an interrupt when nothing happens on the I²C bus for an inordinately long period of time while a transmission is in progress. If this interrupt occurs, the program has the opportunity to attempt to correct the fault and resume I²C operation.

Six time spans are important in I²C operation and are insured by timer I:

- The MINIMUM HIGH time for SCL when this device is the master.
- The MINIMUM LOW time for SCL when this device is a master. This is not very important for a single-bit hardware interface like this one, because the SCL low time is stretched until the software responds to the I²C flags. The software response time normally meets or exceeds the MIN LO time. In cases where the software responds within MIN HI + MIN LO) time, timer I will ensure that the minimum time is met.
- The MINIMUM SCL HIGH TO SDA HIGH time in a stop condition.
- The MINIMUM SDA HIGH TO SDA LOW time between I²C stop and start conditions (4.7ms, see I²C specification).
- The MINIMUM SDA LOW TO SCL LOW time in a start condition.
- The MAXIMUM SCL CHANGE time while an I²C frame is in progress. A frame is in progress between a start condition and the following stop condition. This time span serves to detect a lack of software response on this device as well as external I²C problems. SCL “stuck low” indicates a faulty master or slave. SCL “stuck high” may mean a faulty device, or that noise induced onto the I²C bus caused all masters to withdraw from I²C arbitration.

The first five of these times are 4.7ms (see I²C specification) and are covered by the low order three bits of timer I. Timer I is clocked by the P87LPC761 CPU clock. Timer I can be pre-loaded with one of four values to optimize timing for different oscillator frequencies. At lower frequencies, software response time is increased and will degrade maximum performance of the I²C bus. See special function register I2CFG description for prescale values (CT0, CT1).

The MAXIMUM SCL CHANGE time is important, but its exact span is not critical. The complete 10 bits of timer I are used to count out the maximum time. When I²C operation is enabled, this counter is cleared by transitions on the SCL pin. The timer does not run between I²C frames (i.e., whenever reset or stop occurred more recently than the last start). When this counter is running, it will carry out after 1020 to 1023 machine cycles have elapsed since a change on SCL. A carry out causes a hardware reset of the I²C interface and generates an interrupt if the Timer I interrupt is enabled. In cases where the bus hang-up is due to a lack of software response by this device, the reset releases SCL and allows I²C operation among other devices to continue.

Timer I is enabled to run, and will reset the I²C interface upon overflow, if the TIRUN bit in the I2CFG register is set. The Timer I interrupt may be enabled via the ETI bit in IEN1, and its priority set by the PTIH and PTI bits in the IP1H and IP1 registers respectively.

I²C Interrupts

If I²C interrupts are enabled (EA and EI2 are both set to 1), an I²C interrupt will occur whenever the ATN flag is set by a start, stop, arbitration loss, or data ready condition (refer to the description of ATN following). In practice, it is not efficient to operate the I²C interface in this fashion because the I²C interrupt service routine would somehow have to distinguish between hundreds of possible conditions. Also, since I²C can operate at a fairly high rate, the software may execute faster if the code simply waits for the I²C interface.

Typically, the I²C interrupt should only be used to indicate a start condition at an idle slave device, or a stop condition at an idle master device (if it is waiting to use the I²C bus). This is accomplished by enabling the I²C interrupt only during the aforementioned conditions.

Reading I2CON

RDAT — The data from SDA is captured into “Receive DATa” whenever a rising edge occurs on SCL. RDAT is also available (with seven low-order zeros) in the I2DAT register. The difference between reading it here and there is that reading I2DAT clears DRDY, allowing the I²C to proceed on to another bit. Typically, the first seven bits of a received byte are read from I2DAT, while the 8th is read here. Then I2DAT can be written to send the Acknowledge bit and clear DRDY.

ATN — “ATteNtion” is 1 when one or more of DRDY, ARL, STR, or STP is 1. Thus, ATN comprises a single bit that can be tested to release the I²C service routine from a “wait loop.”

DRDY — “Data ReaDY” (and thus ATN) is set when a rising edge occurs on SCL, except at idle slave. DRDY is cleared by writing CDR = 1, or by writing or reading the I2DAT register. The following low period on SCL is stretched until the program responds by clearing DRDY.

Checking ATN and DRDY

When a program detects ATN = 1, it should next check DRDY. If DRDY = 1, then if it receives the last bit, it should capture the data from RDAT (in I2DAT or I2CON). Next, if the next bit is to be sent, it should be written to I2DAT. One way or another, it should clear DRDY and then return to monitoring ATN. Note that if any of ARL, STR, or STP is set, clearing DRDY will not release SCL to high, so that the I^2C will not go on to the next bit. If a program detects ATN = 1, and DRDY = 0, it should go on to examine ARL, STR, and STP.

ARL — "Arbitration Loss" is 1 when transmit Active was set, but this device lost arbitration to another transmitter. Transmit Active is cleared when ARL is 1. There are four separate cases in which ARL is set.

1. If the program sent a 1 or repeated start, but another device sent a 0, or a stop, so that SDA is 0 at the rising edge of SCL. (If the other device sent a stop, the setting of ARL will be followed shortly by STP being set.)

2. If the program sent a 1, but another device sent a repeated start, and it drove SDA low before SCL could be driven low. (This type of ARL is always accompanied by STR = 1.)

3. In master mode, if the program sent a repeated start, but another device sent a 1, and it drove SCL low before this device could drive SDA low.

4. In master mode, if the program sent stop, but it could not be sent because another device sent a 0.

STR — "STaRt" is set to a 1 when an I^2C start condition is detected at a non-idle slave or at a master. (STR is not set when an idle slave becomes active due to a start bit; the slave has nothing useful to do until the rising edge of SCL sets DRDY.)

STP — "SToP" is set to 1 when an I^2C stop condition is detected at a non-idle slave or at a master. (STP is not set for a stop condition at an idle slave.)

MASTER — "MASTER" is 1 if this device is currently a master on the I^2C. MASTER is set when MASTRQ is 1 and the bus is not busy (i.e., if a start bit hasn't been received since reset or a "Timer I" time-out, or if a stop has been received since the last start). MASTER is cleared when ARL is set, or after the software writes MASTRQ = 0 and then XSTP = 1.

Writing I2CON

Typically, for each bit in an I^2C message, a service routine waits for ATN = 1. Based on DRDY, ARL, STR, and STP, and on the current bit position in the message, it may then write I2CON with one or more of the following bits, or it may read or write the I2DAT register.

CXA — Writing a 1 to "Clear Xmit Active" clears the Transmit Active state. (Reading the I2DAT register also does this.)

Regarding Transmit Active

Transmit Active is set by writing the I2DAT register, or by writing I2CON with XSTR = 1 or XSTP = 1. The I^2C interface will only drive the SDA line low when Transmit Active is set, and the ARL bit will only be set to 1 when Transmit Active is set. Transmit Active is cleared by reading the I2DAT register, or by writing I2CON with CXA = 1. Transmit Active is automatically cleared when ARL is 1.

IDLE — Writing 1 to "IDLE" causes a slave's I^2C hardware to ignore the I^2C until the next start condition (but if MASTRQ is 1, then a stop condition will cause this device to become a master).

CDR — Writing a 1 to "Clear Data Ready" clears DRDY. (Reading or writing the I2DAT register also does this.)

CARL — Writing a 1 to "Clear Arbitration Loss" clears the ARL bit.

CSTR — Writing a 1 to "Clear STaRt" clears the STR bit.

CSTP — Writing a 1 to "Clear SToP" clears the STP bit. Note that if one or more of DRDY, ARL, STR, or STP is 1, the low time of SCL is stretched until the service routine responds by clearing them.

XSTR — Writing 1s to "Xmit repeated STaRt" and CDR tells the I^2C hardware to send a repeated start condition. This should only be at a master. Note that XSTR need not and should not be used to send an "initial" (non-repeated) start; it is sent automatically by the I^2C hardware. Writing XSTR = 1 includes the effect of writing I2DAT with XDAT = 1; it sets Transmit Active and releases SDA to high during the SCL low time. After SCL goes high, the I^2C hardware waits for the suitable minimum time and then drives SDA low to make the start condition.

XSTP — Writing 1s to "Xmit SToP" and CDR tells the I^2C hardware to send a stop condition. This should only be done at a master. If there are no more messages to initiate, the service routine should clear the MASTRQ bit in I2CFG to 0 before writing XSTP with 1. Writing XSTP = 1 includes the effect of writing I2DAT with XDAT = 0; it sets Transmit Active and drives SDA low during the SCL low time. After SCL goes high, the I^2C hardware waits for the suitable minimum time and then releases SDA to high to make the stop condition.

Table 1. Interaction of TIRUN with SLAVEN, MASTRQ, and MASTER

SLAVEN, MASTRQ, MASTER	TIRUN	OPERATING MODE
All 0	0	The I^2C interface is disabled. Timer I is cleared and does not run. This is the state assumed after a reset. If an I^2C application wants to ignore the I^2C at certain times, it should write SLAVEN, MASTRQ, and TIRUN all to zero.
All 0	1	The I^2C interface is disabled.
Any or all 1	0	The I^2C interface is enabled. The 3 low-order bits of Timer I run for min-time generation, but the hi-order bits do not, so that there is no checking for I^2C being “hung.” This configuration can be used for very slow I^2C operation.
Any or all 1	1	The I^2C interface is enabled. Timer I runs during frames on the I^2C, and is cleared by transitions on SCL, and by Start and Stop conditions. This is the normal state for I^2C operation.

Table 2. CT1, CT0 Values

CT1, CT0	Min Time Count (Machine Cycles)	CPU Clock Max (for 100 kHz I^2C)	Timeout Period (Machine Cycles)
1 0	7	8.4 MHz	1023
0 1	6	7.2 MHz	1022
0 0	5	6.0 MHz	1021
1 1	4	4.8 MHz	1020

Interrupts

The P87LPC761 uses a four priority level interrupt structure. This allows great flexibility in controlling the handling of the P87LPC761’s many interrupt sources. The P87LPC761 supports up to 11 interrupt sources.

Each interrupt source can be individually enabled or disabled by setting or clearing a bit in registers IEN0 or IEN1. The IEN0 register also contains a global disable bit, EA, which disables all interrupts at once.

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the IP0, IP0H, IP1, and IP1H registers. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. So, if two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve simultaneous requests of the same priority level.

Table 3 summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from Power Down mode.

Table 3. Summary of Interrupts

Description	Interrupt Flag Bit(s)	Vector Address	Interrupt Enable Bit(s)	Interrupt Priority	Arbitration Ranking	Power Down Wakeup
External Interrupt 0	IE0	0003h	EX0 (IEN0.0)	IP0H.0, IP0.0	1 (highest)	Yes
Timer 0 Interrupt	TF0	000Bh	ET0 (IEN0.1)	IP0H.1, IP0.1	4	No
Timer 1 Interrupt	TF1	001Bh	ET1 (IEN0.3)	IP0H.3, IP0.3	8	No
Serial Port Tx and Rx	TI & RI	0023h	ES (IEN0.4)	IP0H.4, IP0.4	10	No
Brownout Detect	BOF	002Bh	EBO (IEN0.5)	IP0H.5, IP0.5	2	Yes
I^2C Interrupt	ATN	0033h	EI2 (IEN1.0)	IP1H.0, IP1.0	5	No
KBI Interrupt	KBF	003Bh	EKB (IEN1.1)	IP1H.1, IP1.1	6	Yes
Comparator 2 interrupt	CMF2	0043h	EC2 (IEN1.2)	IP1H.2, IP1.2	9	Yes
Watchdog Timer	WDOVF	0053h	EWD (IEN0.6)	IP0H.6, IP0.6	3	Yes
Comparator 1 interrupt	CMF1	0063h	EC1 (IEN1.5)	IP1H.5, IP1.5	7	Yes
Timer I interrupt	–	0073h	ETI (IEN1.7)	IP1H.7, IP1.7	11 (lowest)	No

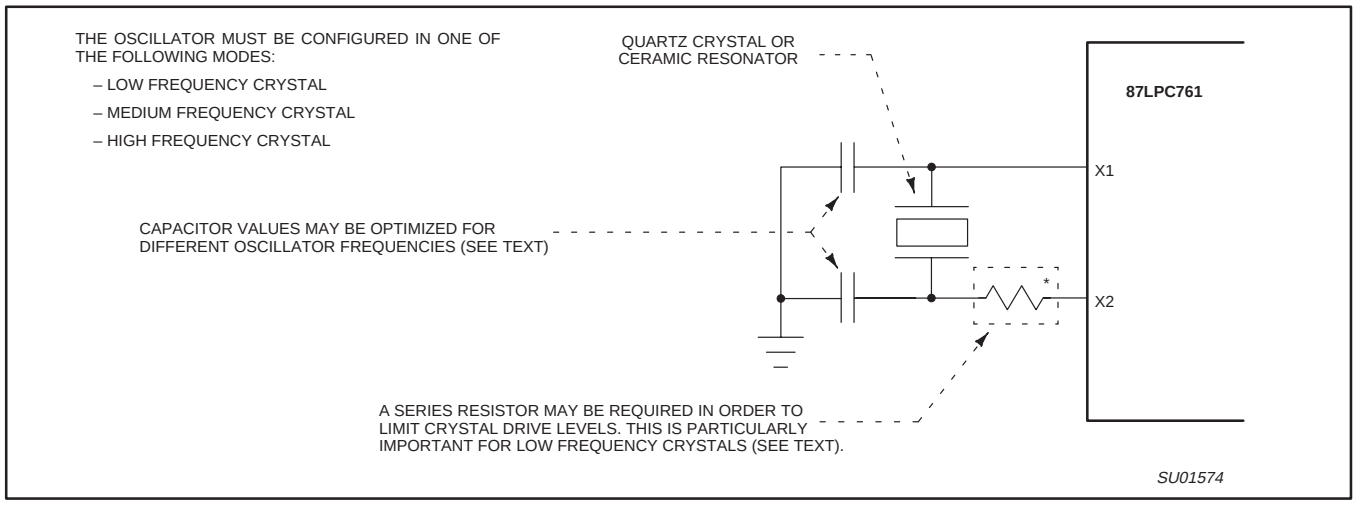


Figure 16. Using the Crystal Oscillator

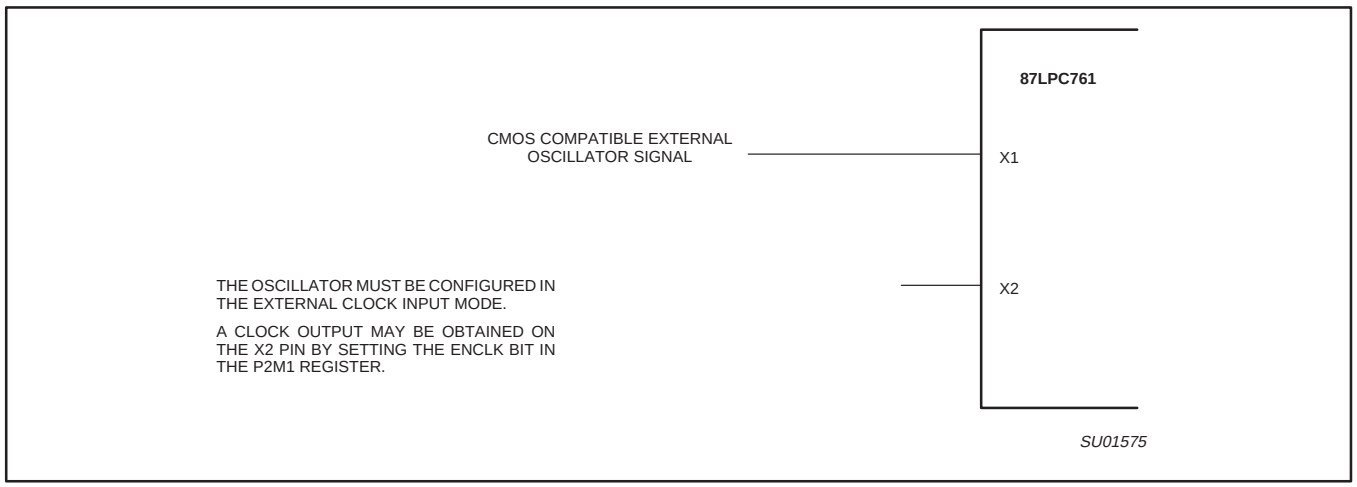


Figure 17. Using an External Clock Input

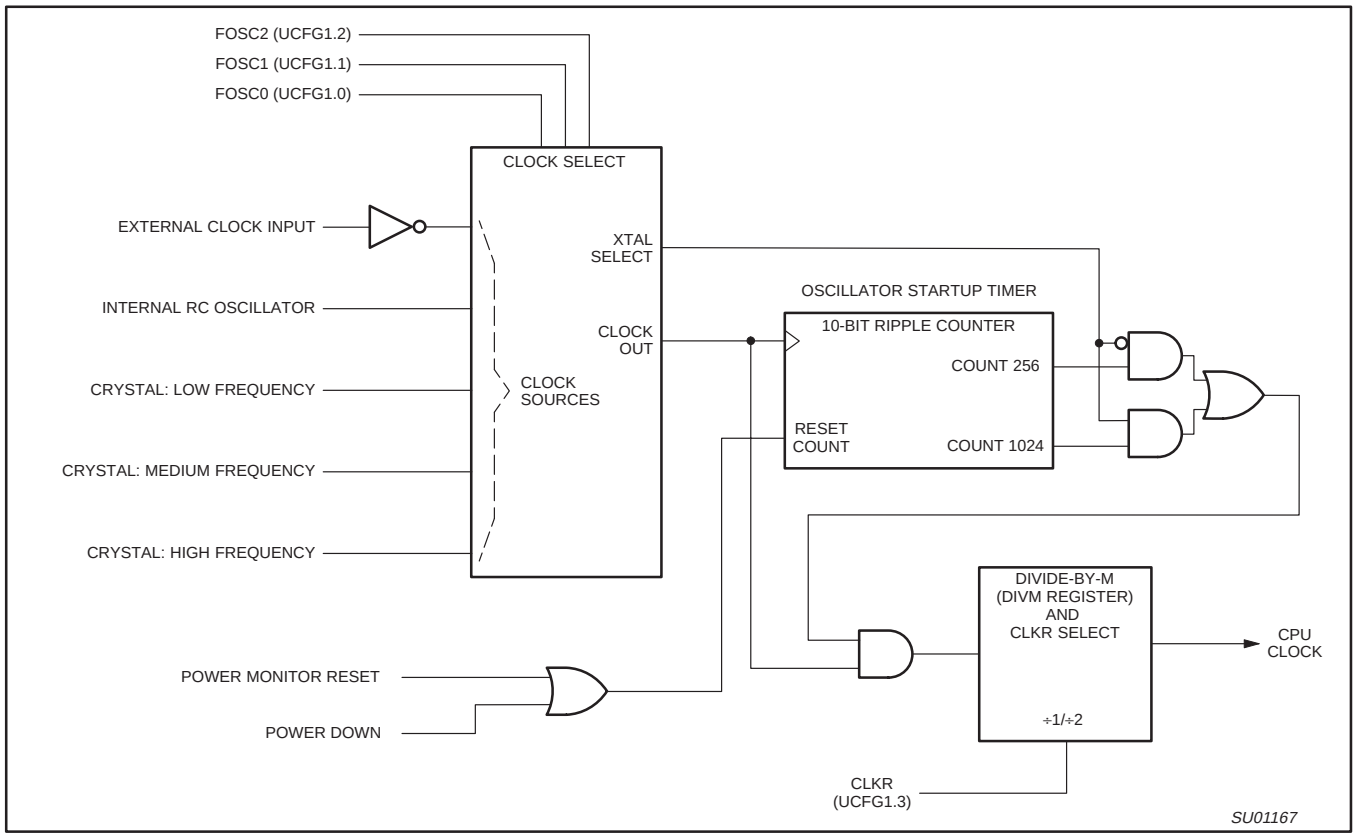


Figure 18. Block Diagram of Oscillator Control

CPU Clock Modification: CLKR and DIVM

For backward compatibility, the CLKR configuration bit allows setting the P87LPC761 instruction and peripheral timing to match standard 80C51 timing by dividing the CPU clock by two. Default timing for the P87LPC761 is 6 CPU clocks per machine cycle while standard 80C51 timing is 12 clocks per machine cycle. This division also applies to peripheral timing, allowing 80C51 code that is oscillator frequency and/or timer rate dependent. The CLKR bit is located in the EPROM configuration register UCFG1, described under EPROM Characteristics

In addition to this, the CPU clock may be divided down from the oscillator rate by a programmable divider, under program control. This function is controlled by the DIVM register. If the DIVM register is set to zero (the default value), the CPU will be clocked by either the unmodified oscillator rate, or that rate divided by two, as determined by the previously described CLKR function.

When the DIVM register is set to some value N (between 1 and 255), the CPU clock is divided by 2 * (N + 1). Clock division values from 4 through 512 are thus possible. This feature makes it possible to temporarily run the CPU at a lower rate, reducing power consumption, in a manner similar to Idle mode. By dividing the clock, the CPU can retain the ability to respond to events other than those that can cause interrupts (i.e. events that allow exiting the Idle mode) by executing its normal program at a lower rate. This can allow bypassing the oscillator startup time in cases where Power Down mode would otherwise be used. The value of DIVM may be changed by the program at any time without interrupting code execution.

Power Monitoring Functions

The P87LPC761 incorporates power monitoring functions designed to prevent incorrect operation during initial power up and power loss or reduction during operation. This is accomplished with two hardware functions: Power-On Detect and Brownout Detect.

Brownout Detection

The Brownout Detect function allows preventing the processor from failing in an unpredictable manner if the power supply voltage drops below a certain level. The default operation is for a brownout detection to cause a processor reset, however it may alternatively be configured to generate an interrupt by setting the BOI bit in the AUXR1 register (AUXR1.5).

The P87LPC761 allows selection of two Brownout levels: 2.5 V or 3.8 V. When V_{DD} drops below the selected voltage, the brownout detector triggers and remains active until V_{DD} is returns to a level above the Brownout Detect voltage. When Brownout Detect causes a processor reset, that reset remains active as long as V_{DD} remains below the Brownout Detect voltage. When Brownout Detect generates an interrupt, that interrupt occurs once as V_{DD} crosses from above to below the Brownout Detect voltage. For the interrupt to be processed, the interrupt system and the BOI interrupt must both be enabled (via the EA and EBO bits in IEN0).

When Brownout Detect is activated, the BOF flag in the PCON register is set so that the cause of processor reset may be determined by software. This flag will remain set until cleared by software.

Timer/Counters

The P87LPC761 has two general purpose counter/timers which are upward compatible with the standard 80C51 Timer 0 and Timer 1. Both can be configured to operate as timers or can be configured to be an event counter (see Figure 22). An option to automatically toggle the T0 pin upon timer overflow has been added.

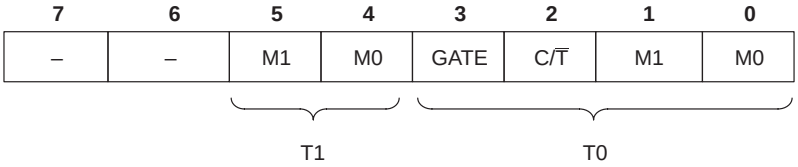
In the “Timer” function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of 6 CPU clock periods, the count rate is 1/6 of the CPU clock frequency. Refer to the section Enhanced CPU for a description of the CPU clock.

In the “Counter” function of Timer 0, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0. In this function, the external input is sampled once during every machine cycle. When the samples of the pin state show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (12 CPU clocks) to recognize a 1-to-0 transition, the maximum count rate is 1/6 of the CPU clock frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

The “Timer” or “Counter” function of Timer 0 is selected by control bit C/$\overline{T}$ in the Special Function Register TMOD. In addition to the “Timer” or “Counter” selection, Timer 0 and Timer 1 have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

TMOD Address: 89h Reset Value: 00h

Not Bit Addressable



BIT	SYMBOL	FUNCTION
TMOD.7, 6	–	Reserved. Must be written with zeros only.
TMOD.5, 4	M1, M0	Mode Select for Timer 1 (see table below).
TMOD.3	GATE	Gating control for Timer 0. When set, Timer/Counter is enabled only while the $\overline{INT0}$ pin is high and the TR0 control pin is set. When cleared, Timer 0 is enabled when the TR0 control bit is set.
TMOD.2	C/$\overline{T}$	Timer or Counter Selector for Timer 0. Cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from T0 input pin).
TMOD.1, 0	M1, M0	Mode Select for Timer 0 (see table below).
	M1, M0	Timer Mode
	0 0	8048 Timer “TLn” serves as 5-bit prescaler.
	0 1	16-bit Timer/Counter “THn” and “TLn” are cascaded; there is no prescaler.
	1 0	8-bit auto-reload Timer/Counter. THn holds a value which is loaded into TLn when it overflows.
	1 1	Timer 0 is a dual 8-bit Timer/Counter in this mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only, controlled by the Timer 1 control bits (see text). Timer 1 in this mode is stopped.

SU01542

Figure 22. Timer/Counter Mode Control Register (TMOD)

Serial Port Control Register (SCON)
The serial port control and status register is the Special Function Register SCON, shown in Figure 31. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

The Framing Error bit (FE) allows detection of missing stop bits in the received data stream. The FE bit shares the bit position SCON.7 with the SM0 bit. Which bit appears in SCON at any particular time is determined by the SMOD0 bit in the PCON register. If SMOD0 = 0, SCON.7 is the SM0 bit. If SMOD0 = 1, SCON.7 is the FE bit. Once set, the FE bit remains set until it is cleared by software. This allows detection of framing errors for a group of characters without the need for monitoring it for every character individually.

SCON Address: 98h Reset Value: 00h

Bit Addressable

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

BIT	SYMBOL	FUNCTION
SCON.7	FE	Framing Error. This bit is set by the UART receiver when an invalid stop bit is detected. Must be cleared by software. The SMOD0 bit in the PCON register must be 1 for this bit to be accessible. See SM0 bit below.
SCON.7	SM0	With SM1, defines the serial port mode. The SMOD0 bit in the PCON register must be 0 for this bit to be accessible. See FE bit above.
SCON. 6	SM1	With SM0, defines the serial port mode (see table below).

SM0, SM1	UART Mode	Baud Rate
0 0	0: shift register	CPU clock/6
0 1	1: 8-bit UART	Variable (see text)
1 0	2: 9-bit UART	CPU clock/32 or CPU clock/16
1 1	3: 9-bit UART	Variable (see text)

SCON.5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2=1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.
SCON.4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
SCON.3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
SCON.2	RB8	In Modes 2 and 3, is the 9th data bit that was received. In Mode 1, it SM2=0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
SCON.1	TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
SCON.0	RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

SU01157

Figure 31. Serial Port Control Register (SCON)

Table 10. Baud Rates, Timer Values, and CPU Clock Frequencies for SMOD1 = 1

Timer Count	Baud Rate						
	2400	4800	9600	19.2 k	38.4 k	57.6 k	115.2 k
–1	0.2304	0.4608	0.9216	* 1.8432	* 3.6864	5.5296	* 11.0592
–2	0.4608	0.9216	* 1.8432	* 3.6864	* 7.3728	* 11.0592	–
–3	0.6912	1.3824	2.7648	5.5296	* 11.0592	16.5888	–
–4	0.9216	* 1.8432	* 3.6864	* 7.3728	* 14.7456	–	–
–5	1.1520	2.3040	4.6080	9.2160	* 18.4320	–	–
–6	1.3824	2.7648	5.5296	* 11.0592	–	–	–
–7	1.6128	3.2256	6.4512	12.9024	–	–	–
–8	* 1.8432	* 3.6864	* 7.3728	* 14.7456	–	–	–
–9	2.0736	4.1472	8.2944	16.5888	–	–	–
–10	2.3040	4.6080	9.2160	* 18.4320	–	–	–
–11	2.5344	5.0688	10.1376	–	–	–	–
–12	2.7648	5.5296	* 11.0592	–	–	–	–
–13	2.9952	5.9904	11.9808	–	–	–	–
–14	3.2256	6.4512	12.9024	–	–	–	–
–15	3.4560	6.9120	13.8240	–	–	–	–
–16	* 3.6864	* 7.3728	* 14.7456	–	–	–	–
–17	3.9168	7.8336	15.6672	–	–	–	–
–18	4.1472	8.2944	16.5888	–	–	–	–
–19	4.3776	8.7552	17.5104	–	–	–	–
–20	4.6080	9.2160	* 18.4320	–	–	–	–
–21	4.8384	9.6768	19.3536	–	–	–	–

NOTES TO TABLES 9 AND 10:

1. Tables 9 and 10 apply to UART modes 1 and 3 (variable rate modes), and show CPU clock rates in MHz for standard baud rates from 2400 to 115.2k baud.
2. Table 9 shows timer settings and CPU clock rates with the SMOD1 bit in the PCON register = 0 (the default after reset), while Table 10 reflects the SMOD1 bit = 1.
3. The tables show all potential CPU clock frequencies up to 20 MHz that may be used for baud rates from 9600 baud to 115.2 k baud. Other CPU clock frequencies that would give only lower baud rates are not shown.
4. Table entries marked with an asterisk (*) indicate standard crystal and ceramic resonator frequencies that may be obtained from many sources without special ordering.

More About UART Mode 0

Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/6 the CPU clock frequency. Figure 32 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between “write to SBUF” and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P1.1 and also enable SHIFT CLOCK to the alternate output function line of P1.0. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1 of the 10th machine cycle after “write to SBUF.” Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 t o the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enable SHIFT CLOCK to the alternate output function line of P1.0. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P1.1 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared as RI is set.

More About UART Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the P87LPC761 the baud rate is determined by the Timer 1 overflow rate. Figure 33 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SBUF” signal.)

The transmission begins with activation of SEND which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after “write to SBUF.”

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated.: 1. R1 = 0, and 2. Either SM2 = 0, or the received stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

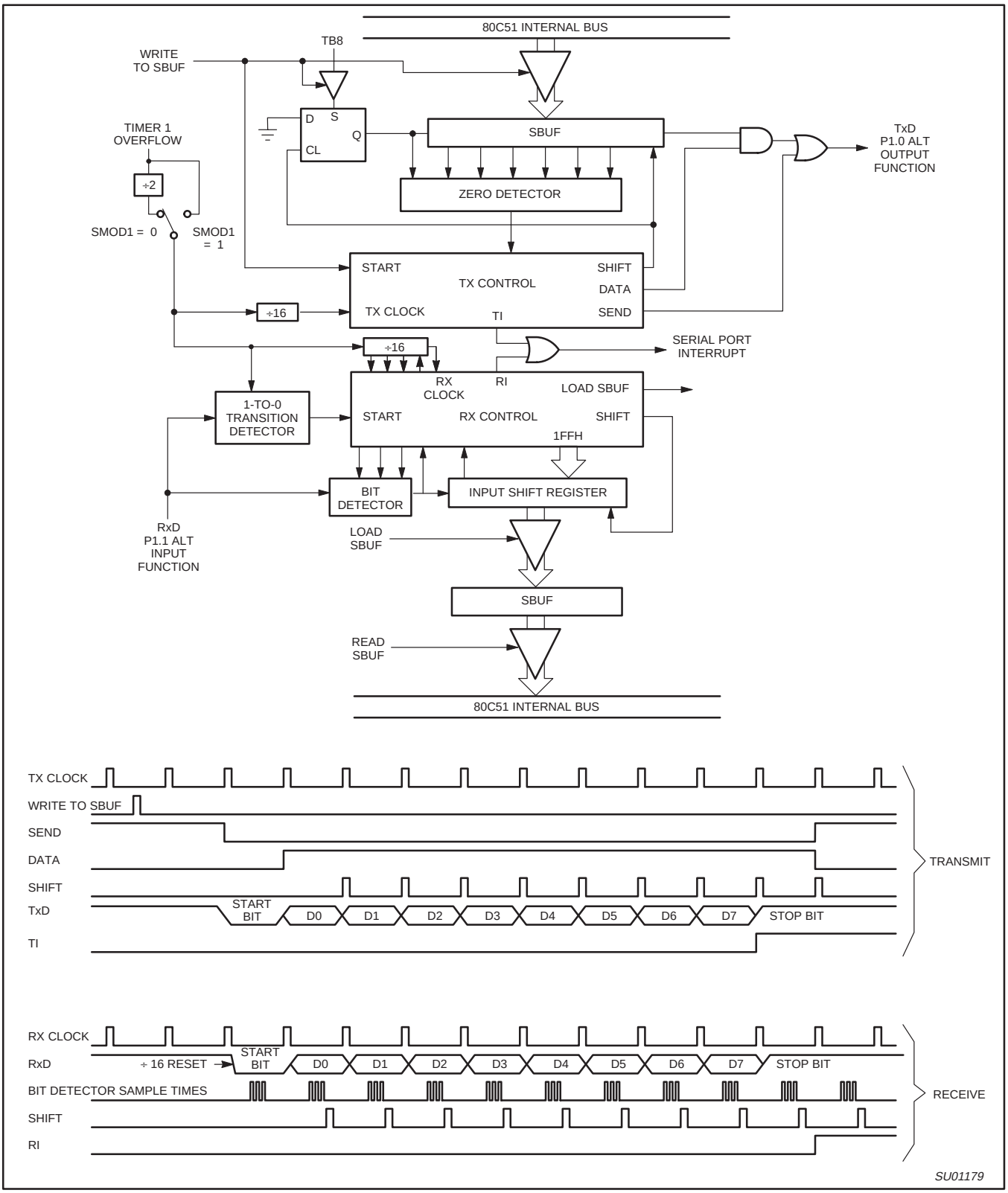


Figure 33. Serial Port Mode 1

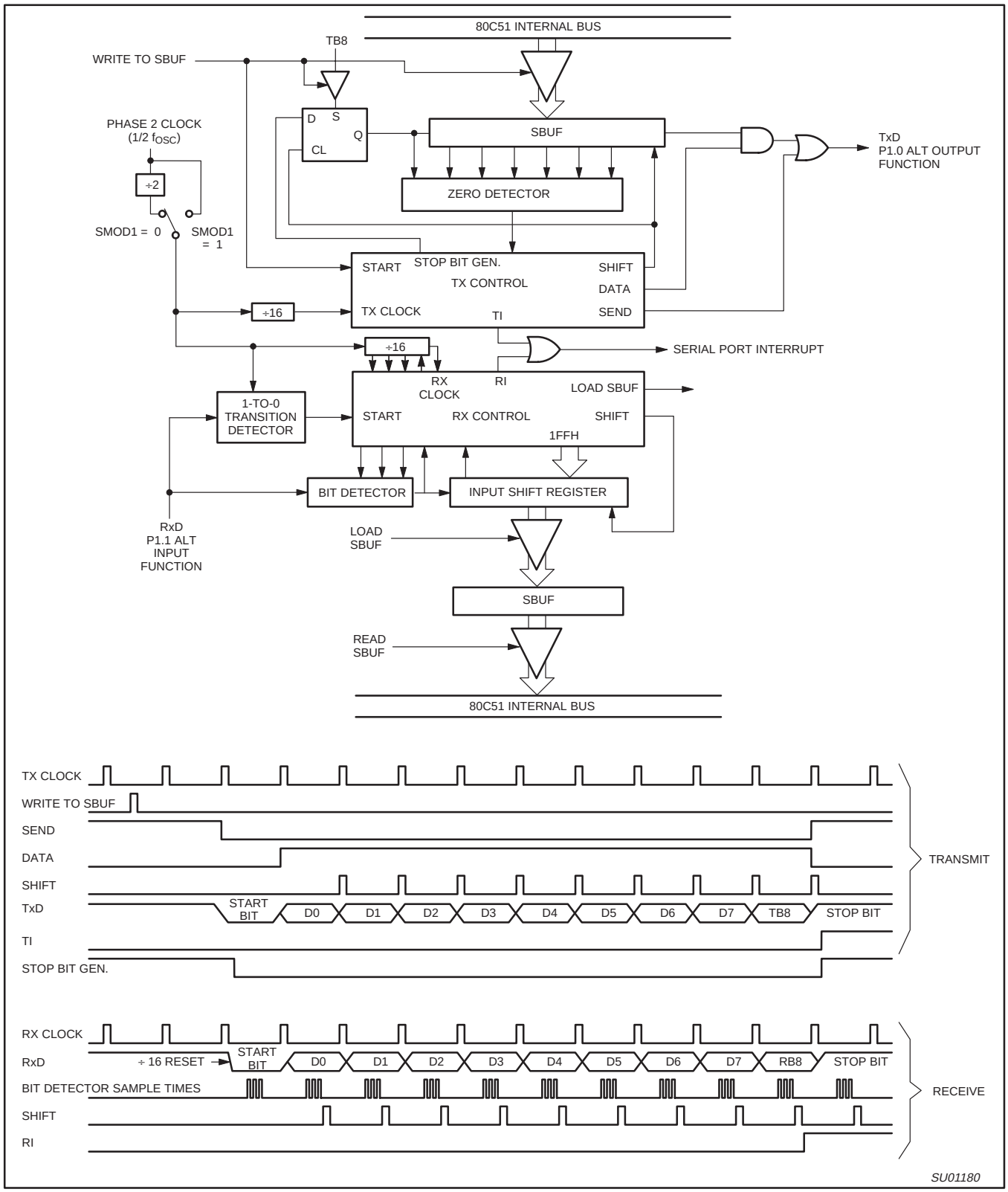


Figure 34. Serial Port Mode 2

EPROM Characteristics

Programming of the EPROM on the P87LPC761 is accomplished with a serial programming method. Commands, addresses, and data are transmitted to and from the device on two pins after programming mode is entered. Serial programming allows easy implementation of in-circuit programming of the P87LPC761 in an application board.

The P87LPC761 contains three signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes designate the device as an P87LPC761 manufactured by Philips. The signature bytes may be read by the user program at addresses FC30h, FC31h and FC60h with the MOVC instruction, using the DPTR register for addressing.

A special user data area is also available for access via the MOVC instruction at addresses FCE0h through FCFFh. This "customer code" space is programmed in the same manner as the main code EPROM and may be used to store a serial number, manufacturing date, or other application information.

32-Byte Customer Code Space

A small supplemental EPROM space is reserved for use by the customer in order to identify code revisions, store checksums, add a serial number to each device, or any other desired use. This area exists in the code memory space from addresses FCE0h through FCFFh. Code execution from this space is not supported, but it may be read as data through the use of the MOVC instruction with the appropriate addresses. The memory may be programmed at the same time as the rest of the code memory and UCFG bytes are programmed.

System Configuration Bytes

A number of user configurable features of the P87LPC761 must be defined at power up and therefore cannot be set by the program after start of execution. Those features are configured through the use of two EPROM bytes that are programmed in the same manner as the EPROM program space. The contents of the two configuration bytes, UCFG1 and UCFG2, are shown in Figures 39 and 40. The values of these bytes may be read by the program through the use of the MOVX instruction at the addresses shown in the figure.

UCFG1 Address: FD00h Unprogrammed Value: FFh

7	6	5	4	3	2	1	0
WDTE	RPD	PRHI	BOV	CLKR	FOSC2	FOSC1	FOSC0

BIT	SYMBOL	FUNCTION
UCFG1.7	WDTE	Watchdog timer enable. When programmed (0), disables the watchdog timer. The timer may still be used to generate an interrupt.
UCFG1.6	RPD	Reset pin disable. When 1 disables the reset function of pin P1.5, allowing it to be used as an input only port pin.
UCFG1.5	PRHI	Port reset high. When 1, ports reset to a high state. When 0, ports reset to a low state.
UCFG1.4	BOV	Brownout voltage select. When 1, the brownout detect voltage is 2.5V. When 0, the brownout detect voltage is 3.8V. This is described in the Power Monitoring Functions section.
UCFG1.3	CLKR	Clock rate select. When 0, the CPU clock rate is divided by 2. This results in machine cycles taking 12 CPU clocks to complete as in the standard 80C51. For full backward compatibility, this division applies to peripheral timing as well.
UCFG1.2–0	FOSC2–FSOC0	CPU oscillator type select. See Oscillator section for additional information. Combinations other than those shown below should not be used. They are reserved for future use.
	FOSC2–FOSC0	Oscillator Configuration
	1 1 1	External clock input on X1 (default setting for an unprogrammed part).
	0 1 1	Internal RC oscillator, 6 MHz. For tolerance, see AC Electrical Characteristics table.
	0 1 0	Low frequency crystal, 20 kHz to 100 kHz.
	0 0 1	Medium frequency crystal or resonator, 100 kHz to 4 MHz.
	0 0 0	High frequency crystal or resonator, 4 MHz to 20 MHz.

SU01477

Figure 39. EPROM System Configuration Byte 1 (UCFG1)

ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Operating temperature under bias	–55 to +125	°C
Storage temperature range	–65 to +150	°C
Voltage on $\overline{RST}/V_{PP}$ pin to V_{SS}	0 to +11.0	V
Voltage on any other pin to V_{SS}	–0.5 to V_{DD}+0.5V	V
Maximum I_{OL} per I/O pin	20	mA
Power dissipation (based on package heat transfer, not device power consumption)	1.5	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 are 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 maximum.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to VSS unless otherwise noted.

COMPARATOR ELECTRICAL CHARACTERISTICS

V_{DD} = 3.0 V to 6.0 V unless otherwise specified; T_{amb} = 0 °C to +70 °C, unless otherwise specified.

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP	MAX	
V_{IO}	Offset voltage comparator inputs[1]		–	–	±10	mV
V_{CR}	Common mode range comparator inputs		0	–	V_{DD}–0.3	V
CMRR	Common mode rejection ratio[1]		–	–	–50	dB
	Response time		–	250	500	ns
	Comparator enable to output valid		–	–	10	µs
I_{IL}	Input leakage current, comparator	$0 < V_{IN} < V_{DD}$	–	–	±10	µA

NOTE:
1. This parameter is guaranteed by characterization, but not tested in production.

AC ELECTRICAL CHARACTERISTICS

T_{amb} = 0 °C to +70 °C, V_{DD} = 2.7 V to 6.0 V unless otherwise specified; V_{SS} = 0 V[1, 2, 3]

SYMBOL	FIGURE	PARAMETER		LIMITS		UNIT
				MIN	MAX	
External Clock						
f_C	42	Oscillator frequency (V_{DD} = 4.0 V to 6.0 V)		0	20	MHz
f_C	42	Oscillator frequency (V_{DD} = 2.7 V to 6.0 V)		0	10	MHz
t_C	42	Clock period and CPU timing cycle		$1/f_C$	–	ns
f_{CLCX}	42	Clock low-time[1]	f_{OSC} = 20 MHz	20	–	ns
f_{CLCX}	42		f_{OSC} = 10 MHz	40	–	ns
f_{CHCX}	42	Clock high-time[1]	f_{OSC} = 20 MHz	20	–	ns
f_{CHCX}	42		f_{OSC} = 10 MHz	40	–	ns
Internal RC Oscillator						
f_{CCAL}		On-chip RC oscillator calibration[2]	f_{RCOSC} = 6 MHz	–1	+1	%
f_{CTOL}		On-chip RC oscillator, 0 °C to +50 °C[3,4] tol.	f_{RCOSC} = 6 MHz	–2.5	+2.5	%
f_{CTOL}		On-chip RC oscillator, 0 °C to +70 °C[3] tol.	f_{RCOSC} = 6 MHz	–5[5]	+2.5	%
Shift Register						
t_{XLXL}	41	Serial port clock cycle time		$6t_C$	–	ns
t_{QVXH}	41	Output data setup to clock rising edge		$5t_C$ – 133	–	ns
t_{XHQX}	41	Output data hold after clock rising edge		$1t_C$ – 80	–	ns
t_{XHDV}	41	Input data setup to clock rising edge		–	$5t_C$ – 133	ns
t_{XHDX}	41	Input data hold after clock rising edge		0	–	ns

NOTES:
1. Applies only to an external clock source, not when a crystal is connected to the X1 and X2 pins.
2. Tested at V_{DD} = 5.0 V and room temperature.
3. These parameters are characterized but not tested.
4. +/– 2.5% accuracy enables serial communication over the UART with the internal Oscillator.
5. Min frequency at hot temperature.

Low power, low price, low pin count (16 pin) microcontroller with 2 kbyte OTP

P87LPC761

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specifications defined by Philips. This specification can be ordered using the code 9398 393 40011.

Data sheet status

Data sheet status[1]	Product status[2]	Definitions
Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Changes will be communicated according to the Customer Product/Process Change Notification (CPCN) procedure SNW-SQ-650A.

[1] Please consult the most recently issued data sheet before initiating or completing a design.

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Contact information

For additional information please visit
http://www.semiconductors.philips.com. Fax: **+31 40 27 24825**

For sales offices addresses send e-mail to:
sales.addresses@www.semiconductors.philips.com.

© Koninklijke Philips Electronics N.V. 2002
All rights reserved. Printed in U.S.A.

Date of release: 03-02

Document order number: 9397 750 09533

Let's make things better.

Philips Semiconductors

PHILIPS