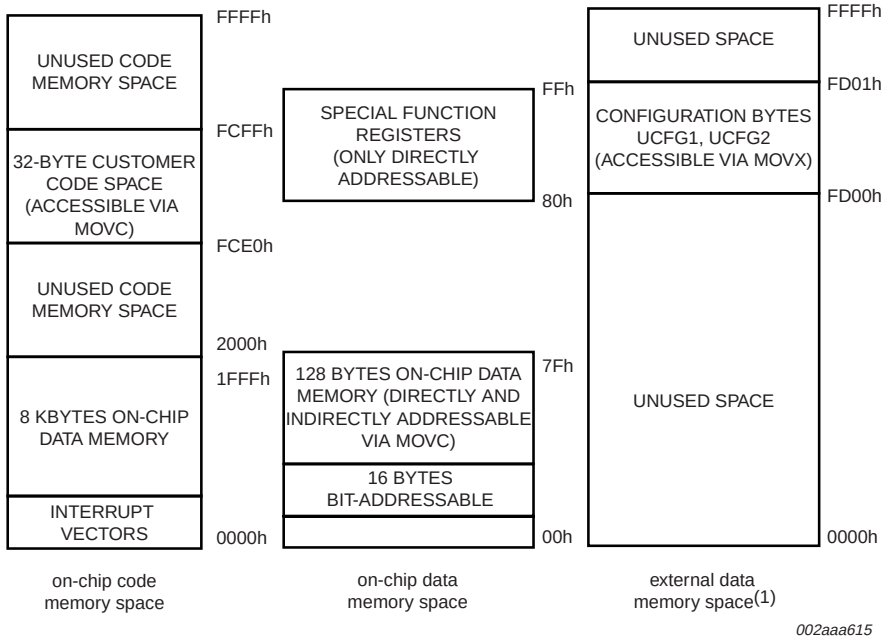
Welcome to E-XFL.COM

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, POR, PWM, WDT
Number of I/O	18
Program Memory Size	8KB (8K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p87lpc778fdh-529



(1) The P87LPC778 does not support access to external data memory. However, the User Configuration Bytes are accessed via the MOVX instruction as if they were in external data memory.

Fig 2. Memory map.

7. Special function registers

Remark: Special Function Registers (SFRs) accesses are restricted in the following ways:

- User must **not** attempt to access any SFR locations not defined.
- Accesses to any defined SFR locations must be strictly for the functions for the SFRs.
- SFR bits labeled '-', '0' or '1' can **only** be written and read as follows:
  - '-' Unless otherwise specified, **must** be written with '0', but can return any value when read (even if it was written with '0'). It is a reserved bit and may be used in future derivatives.
  - '0' **must** be written with '0', and will return a '0' when read.
  - '1' **must** be written with '1', and will return a '1' when read.

Table 3: Special function registers

** indicates SFRs that are bit addressable.*

Name	Description	SFR addr.	Bit functions and addresses								Reset value
			MSB							LSB	Hex
		Bit address	E7	E6	E5	E4	E3	E2	E1	E0	
ACC*	Accumulator	E0H									00H
		Bit address	C7	C6	C5	C4	C3	C2	C1	C0	
ADCON*	A/D Control	C0h	ENADC			ADCI	ADCS	RCCLK	AADR1	AADR0	02h
AUXR1	Auxiliary Function Register	A2h	KBF	BOD	BOI	LPEP	SRST	0	-	DPS	02h
		Bit address	F7	F6	F5	F4	F3	F2	F1	F0	
B*	B register	F0h									00H
CMP1	Comparator 1 control register	ACh	-	-	CE1	CP1	CN1	OE1	CO1	CMF1	00H
CMP2	Comparator 2 control register	ADh	-	-	CE2	CP2	CN2	OE2	CO2	CMF2	00H
CNSW0	PWM Counter shadow register0	D1h	CNSW7	CNSW6	CNSW5	CNSW4	CNSW3	CNSW2	CNSW1	CNSW0	FFh
CNSW1	PWM Counter shadow register1	D2h	-	-	-	-	-	-	CNSW9	CNSW8	FFh
CPSW0	PWM Compare shadow register0	D3h	CPSW07	CPSW06	CPSW05	CPSW04	CPSW03	CPSW02	CPSW01	CPSW00	00H
CPSW1	PWM Compare shadow register0	D4h	CPSW17	CPSW16	CPSW15	CPSW14	CPSW13	CPSW12	CPSW11	CPSW10	00H
CPSW2	PWM Compare shadow register0	D5h	CPSW27	CPSW26	CPSW25	CPSW24	CPSW23	CPSW22	CPSW21	CPSW20	00H
CPSW3	PWM Compare shadow register0	D6h	CPSW37	CPSW36	CPSW35	CPSW34	CPSW33	CPSW32	CPSW31	CPSW30	00H
CPSW4	PWM Compare shadow register0	D7h	CPSW47	CPSW46	CPSW45	CPSW44	CPSW43	CPSW42	CPSW41	CPSW40	00H
DIVM	CPU clock divide-by-M control	95h									00H
DPTR	Data pointer (2 bytes)										
DPH	Data pointer HIGH	83h									00H
DPL	Data pointer LOW	82h									00H
		Bit address	CF	CE	CD	CC	CB	CA	C9	C8	
I2CFG*	I^2C-bus configuration register	C8h/RD	SLAVEN	MASTRQ	0	TIRUN	-	-	CT1	CT0	00H
		C8h/WR	SLAVEN	MASTRQ	CLRTI	TIRUN	-	-	CT1	CT0	
		Bit address	DF	DE	DD	DC	DB	DA	D9	D8	

Table 6: ADCON - ADCI, ADCS A/D status

ADCI, ADCS	A/D status
0 0	A/D not busy, a conversion can be started
0 1	A/D busy, the start of a new conversion is blocked.
1 0	An A/D conversion is complete. ADCI must be cleared prior to starting a new conversion.
1 1	An A/D conversion is complete. ADCI must be cleared prior to starting a new conversion. This state exists for one machine cycle as an A/D conversion is completed.

Table 7: ADCON - AADR1, AADR0 A/D input selection

AADR1, AADR0	A/D input selected
0 0	AD0 (P0.3)
0 1	AD1 (P0.4)
1 0	AD2 (P0.5)
1 1	AD3 (P0.6)

8.4 A/D timing

The A/D may be clocked in one of two ways. The default is to use the CPU clock as the A/D clock source. When used in this manner, the A/D completes a conversion in 31 machine cycles. The A/D may be operated up to the maximum CPU clock rate of 20 MHz, giving a conversion time of 9.3 µs. The formula for calculating A/D conversion time when the CPU clock runs the A/D is: 186 µs / CPU clock rate (in MHz). To obtain accurate A/D conversion results, the CPU clock must be at least 1 MHz.

The A/D may also be clocked by the on-chip RC oscillator, even if the RC oscillator is not used as the CPU clock. This is accomplished by setting the RCCLK bit in ADCON. This arrangement has several advantages. First, the A/D conversion time is faster at lower CPU clock rates. Also, the CPU may be run at speeds below 1 MHz without affecting A/D accuracy. Finally, the Power-down mode may be used to completely shut down the CPU and its oscillator, along with other peripheral functions, in order to obtain the best possible A/D accuracy.

When the A/D is operated from the RCCLK while the CPU is running from another clock source, 3 or 4 machine cycles are used to synchronize A/D operation. The time can range from a minimum of 3 machine cycles (at the CPU clock rate) + 108 RC clocks to a maximum of 4 machine cycles (at the CPU clock rate) + 112 RC clocks.

Example A/D conversion times at various CPU clock rates are shown in Table 8. In the table, maximum times for RCCLK = 1 use an RC clock frequency of 6 MHz. Minimum times for RCCLK = 1 use an RC clock frequency of. Nominal time assume an ideal RC clock frequency of 6 MHz and an average of 3.5 machine cycles at the CPU clock rate.

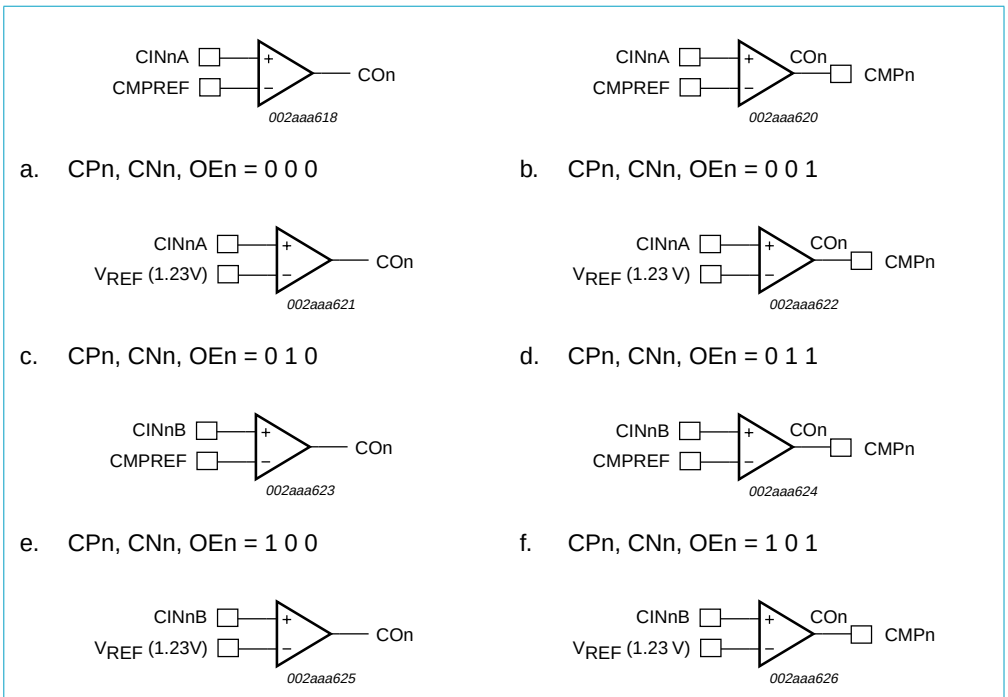


Fig 7. Comparator configurations.

8.5.2 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 Vref, is 1.28 V ±10 %.

8.5.3 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.

8.5.4 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.

8.5.5 Comparator configuration example

The code shown below 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.

```
CmpInit: b
  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.
```

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

8.6 Pulse width modulator

The P87LPC778 contains four Pulse Width Modulated (PWM) channels which can generate pulses of programmable length and interval.

The output for PWM0 is on P0.1, PWM1 on P1.6, PWM2 on P1.7 and PWM3 on P0.0.

After chip reset the output of the each PWM channel is reflect by the setting of UCFG1.5, PRHI, if set to a zero the outputs are low, if set to one the outputs are high.

In this case PRHI is set to zero, before the pin will reflect the state of the internal PWM output a ‘1’ must be written to each port bit that serves as a PWM output.

A block diagram is shown in Figure 8.

The interval between successive outputs is controlled by a 10–bit down counter which uses the internal microcontroller clock as its input.

When bit 3 in the UCFG1 register is a ‘1’ (6-clock mode) the microcontroller clock, and therefore the PWM counter clock, has the same frequency as the clock source:

$$f_{CPWM} = f_{OSC} \quad (2)$$

When bit 3 in the UCFG1 register is a ‘0’ (12-clock mode) the microcontroller and PWM counter clocks operate at half the frequency of clock source:

$$f_{CPWM} = \frac{f_{OSC}}{2} \quad (3)$$

When the counter reaches underflow it is reloaded with a user selectable value. This mechanism allows the user to set the PWM frequency at any integer sub-multiple of the microcontroller clock frequency. The repetition frequency of the PWM is given by:

$$f_{PWM} = \frac{f_{CPWM}}{(CNSW + 1)} \quad (4)$$

Where CNSW is contained in SFRs CNSW0 and CNSW1 as a 10-bit value, described in the following tables.

The word ‘Shadow’ refers to the fact that writes are not into the register that controls the counter; rather they are into a holding register.

As described below the transfer of data from this holding register, into the register which contains the actual reload value, is controlled by the user’s program, by setting the XFER bit and waiting till the next underflow of the counter.

Table 11: CNSW0 - Counter shadow register 0 (address 0D1H) bit allocation

Reset value: FFH

Bit	7	6	5	4	3	2	1	0
Symbol	CNSW7	CNSW6	CNSW5	CNSW4	CNSW3	CNSW2	CNSW1	CNSW0

Table 12: CNSW1 - Counter shadow register 1 (address 0D2H) bit allocation

Reset value: FFH

Bit	7	6	5	4	3	2	1	0
Symbol	Unused	Unused	Unused	Unused	Unused	Unused	CNSW9	CNSW8

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-bus 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-bus 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-bus. 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.

8.7.4 Writing I2CON

Typically, for each bit in an I^2C-bus 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.)

8.7.5 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-bus 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-bus hardware to ignore the I^2C-bus 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.

pulled LOW, it is driven strongly and able to sink a fairly large current. These features are somewhat similar to an open drain output except that there are three pull-up transistors in the quasi-bidirectional output that serve different purposes.

One of these pull-ups, called the ‘very weak’ pull-up, is turned on whenever the port latch for the pin contains a logic 1. The very weak pull-up sources a very small current that will pull the pin HIGH if it is left floating.

A second pull-up, called the ‘weak’ pull-up, is turned on when the port latch for the pin contains a logic 1 and the pin itself is also at a logic 1 level. This pull-up provides the primary source current for a quasi-bidirectional pin that is outputting a ‘1’. If a pin that has a logic 1 on it is pulled LOW by an external device, the weak pull-up turns off, and only the very weak pull-up remains on. In order to pull the pin LOW under these conditions, the external device has to sink enough current to overpower the weak pull-up and take the voltage on the port pin below its input threshold.

The third pull-up is referred to as the ‘strong’ pull-up. This pull-up is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port latch changes from a logic 0 to a logic 1. When this occurs, the strong pull-up turns on for a brief time, two CPU clocks, in order to pull the port pin HIGH quickly. Then it turns off again.

The quasi-bidirectional port configuration is shown in Figure 10.

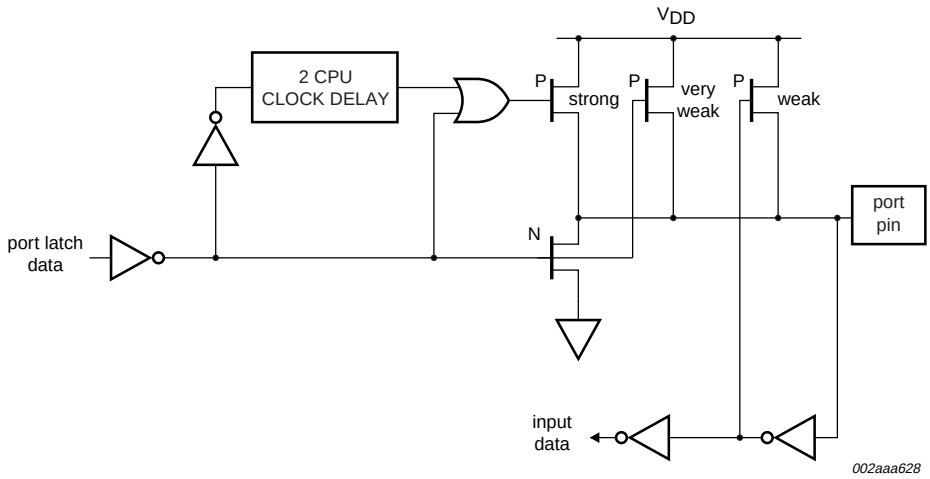


Fig 10. Quasi-bidirectional output.

8.9.2 Open drain output configuration

The open drain output configuration turns off all pull-ups and only drives the pulldown transistor of the port driver when the port latch contains a logic 0. To be used as a logic output, a port configured in this manner must have an external pull-up, typically a resistor tied to V_{DD}. The pulldown for this mode is the same as for the quasi-bidirectional mode.

The open drain port configuration is shown in Figure 11.

8.9.4 Keyboard interrupt (KBI)

The Keyboard Interrupt function is intended primarily to allow a single interrupt to be generated when any key is pressed on a keyboard or keypad connected to specific pins of the P87LPC778, as shown in Figure 13. This interrupt may be used to wake up the CPU from Idle or Power-down modes. This feature is particularly useful in handheld, battery powered systems that need to carefully manage power consumption yet also need to be convenient to use.

The P87LPC778 allows any or all pins of port 0 to be enabled to cause this interrupt. Port pins are enabled by the setting of bits in the KBI register, as shown in Tables 35 and 36. The Keyboard Interrupt Flag (KBF) in the AUXR1 register is set when any enabled pin is pulled LOW while the KBI interrupt function is active. An interrupt will generated if it has been enabled. Note that the KBF bit must be cleared by software.

Due to human time scales and the mechanical delay associated with keyswitch closures, the KBI feature will typically allow the interrupt service routine to poll port 0 in order to determine which key was pressed, even if the processor has to wake up from Power-down mode. Refer to Section 8.12 “Power reduction modes” on page 43 for details.

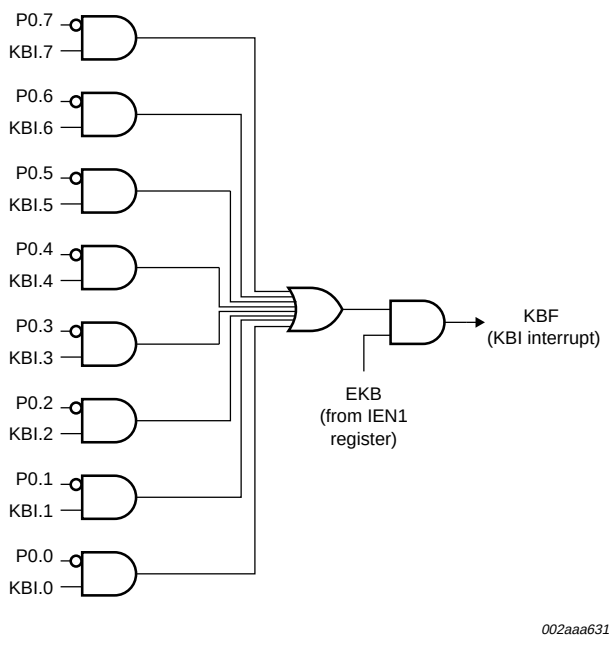
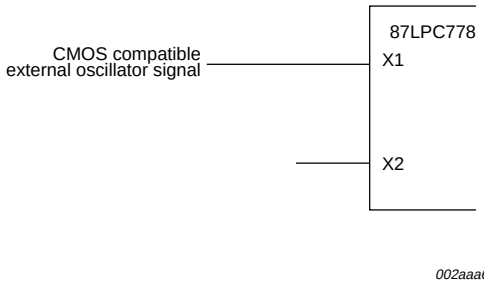


Fig 13. Keyboard interrupt.

Table 35: KBI - Keyboard interrupt register (address 86H) bit allocation

Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	KBI.7	KBI.6	KBI.5	KBI.4	KBI.3	KBI.2	KBI.1	KBI.0



The oscillator must be configured in the External Clock input mode. A clock output may be obtained on the X2 pin by setting the ENCLK bit in the P2M1 register.

Fig 15. Using an external clock input.

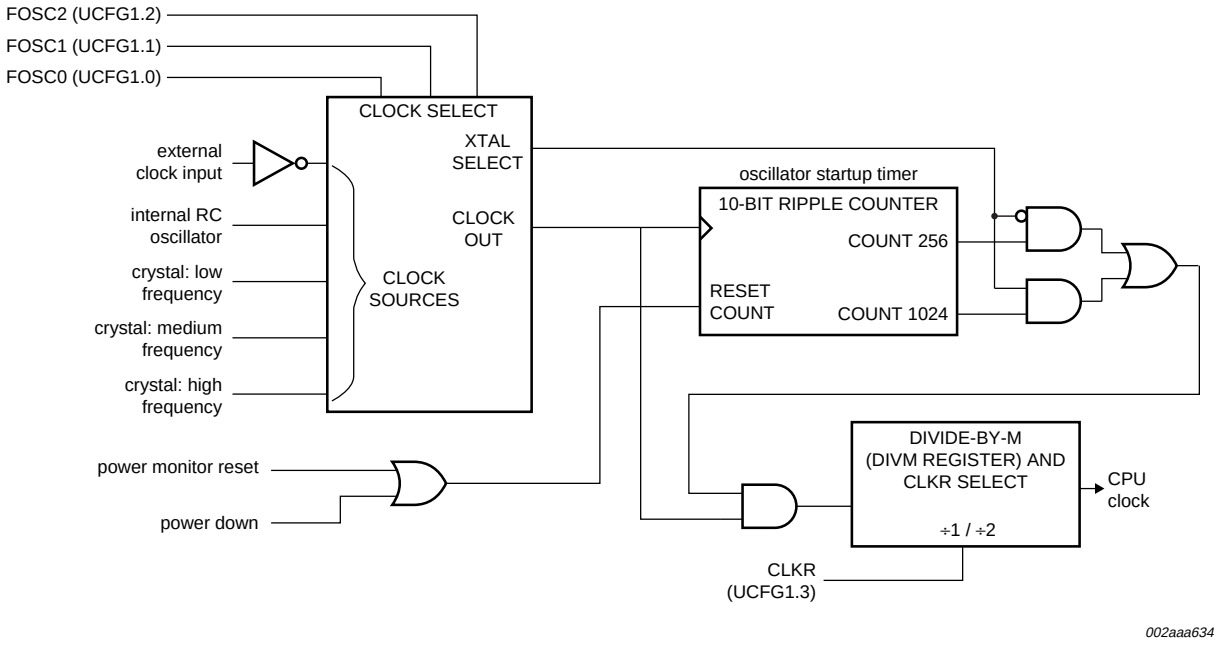


Fig 16. Block diagram of oscillator control

8.10.7 CPU clock modification: CLKR and DIVM

For backward compatibility, the CLKR configuration bit allows setting the P87LPC778 instruction and peripheral timing to match standard 80C51 timing by dividing the CPU clock by two. Default timing for the P87LPC778 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 Section 8.18 “EPROM characteristics” on page 66.

Table 43: TMOD - Timer/counter mode control register (address 89H) bit allocation

Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

Table 44: TMOD - Timer/counter mode control register (address 89H) bit description

Bit	Symbol	Description
7	GATE	Gating control for Timer1. When set, Timer/Counter is enabled only while the INT1 pin is high and the TR1 control pin is set. When cleared, Timer1 is enabled when the TR1 control bit is set.
6	C/$\overline{T}$	Timer or Counter Selector for Timer1. Cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from T1 input pin).
5, 4	M1, M0	Mode select for Timer1 (see Table 45 below).
3	GATE	Gating control for Timer0. When set, Timer/Counter is enabled only while the $\overline{INT0}$ pin is high and the TR0 control pin is set. When cleared, Timer0 is enabled when the TR0 control bit is set.
2	C/$\overline{T}$	Timer or Counter Selector for Timer0. Cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from T0 input pin).
1, 0	M1, M0	Mode Select for Timer0 (see Table 45 below).

Table 45: M1, M0 timer mode

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	Timer0 is a dual 8-bit Timer/Counter in this mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer0 control bits. TH0 is an 8-bit timer only, controlled by the Timer1 control bits (see text). Timer1 in this mode is stopped.

8.14.1 Mode 0

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 19 shows Mode 0 operation.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TFn. The count input is enabled to the Timer when TRn = 1 and either GATE = 0 or $\overline{INTn}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTn}$, to facilitate pulse width measurements). TRn is a control bit in the Special Function Register TCON (Tables 46 and 47). The GATE bit is in the TMOD register.

The 13-bit register consists of all 8 bits of THn and the lower 5 bits of TLn. The upper 3 bits of TLn are indeterminate and should be ignored. Setting the run flag (TRn) does not clear the registers.

Mode 0 operation is the same for Timer0 and Timer1. See Figure 19. There are two different GATE bits, one for Timer1 (TMOD.7) and one for Timer0 (TMOD.3).

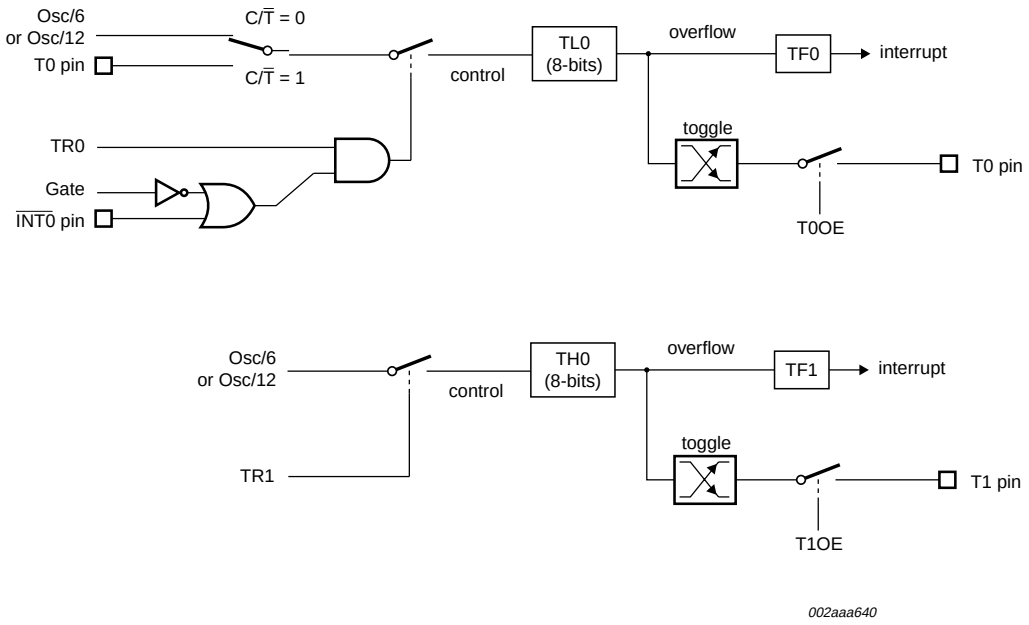


Fig 22. Timer/counter 0 Mode 3 (two 8-bit counters).

8.14.5 Timer overflow toggle output

Timers 0 and 1 can be configured to automatically toggle a port output whenever a timer overflow occurs. The same device pins that are used for the T0 and T1 count inputs are also used for the timer toggle outputs. This function is enabled by control bits ENT0 and ENT1 in the P2M1 register, and apply to Timer0 and Timer1 respectively. The port outputs will be a logic 1 prior to the first timer overflow when this mode is turned on.

8.15 UART

The P87LPC778 includes an enhanced 80C51 UART. The baud rate source for the UART is Timer1 for modes 1 and 3, while the rate is fixed in modes 0 and 2. Because CPU clocking is different on the P87LPC778 than on the standard 80C51, baud rate calculation is somewhat different. Enhancements over the standard 80C51 UART include Framing Error detection and automatic address recognition.

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the SBUF register. However, if the first byte still hasn't been read by the time reception of the second byte is complete, the first byte will be lost. The serial port receive and transmit registers are both accessed through Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can be operated in 4 modes.

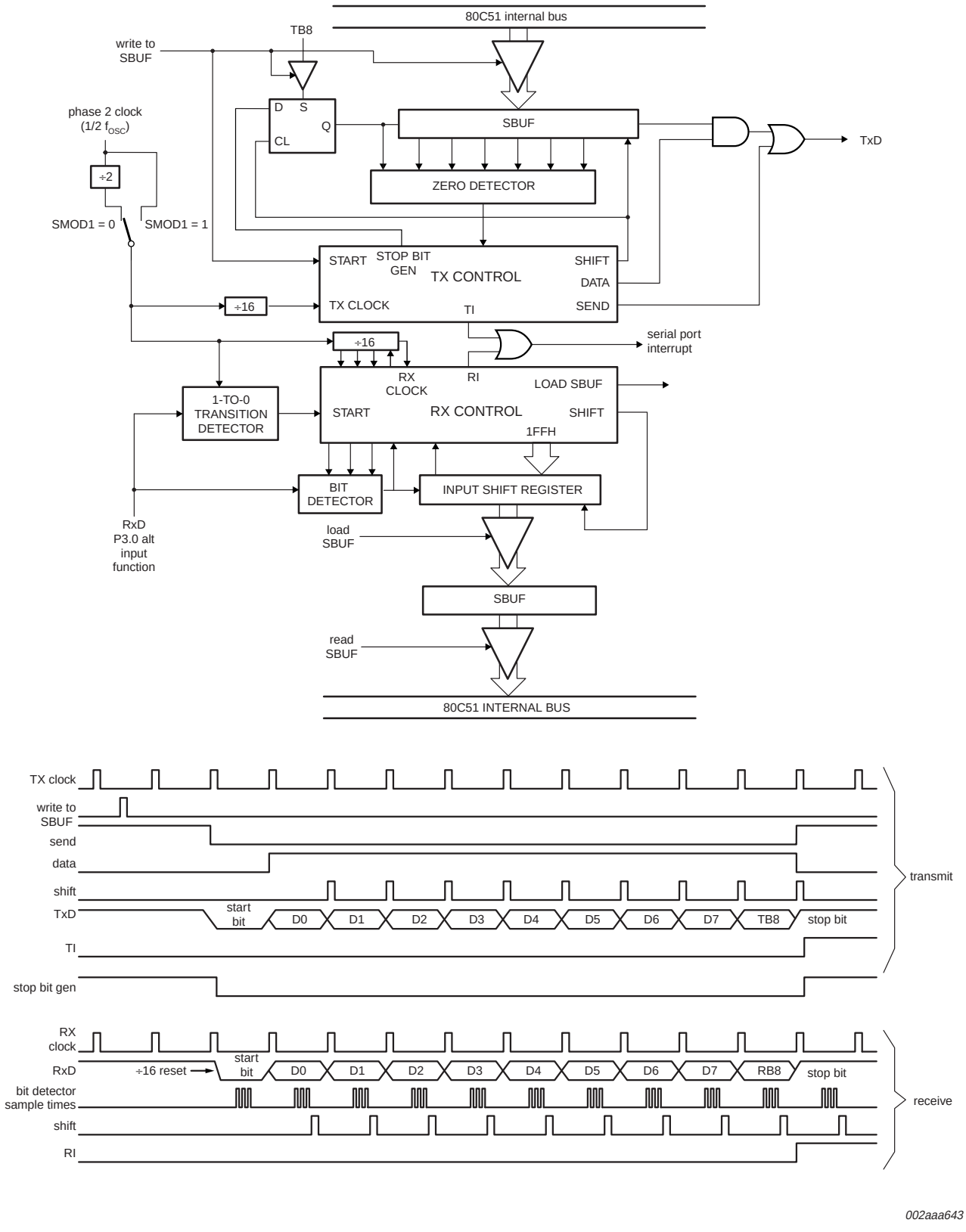


Fig 25. Serial port mode 2.

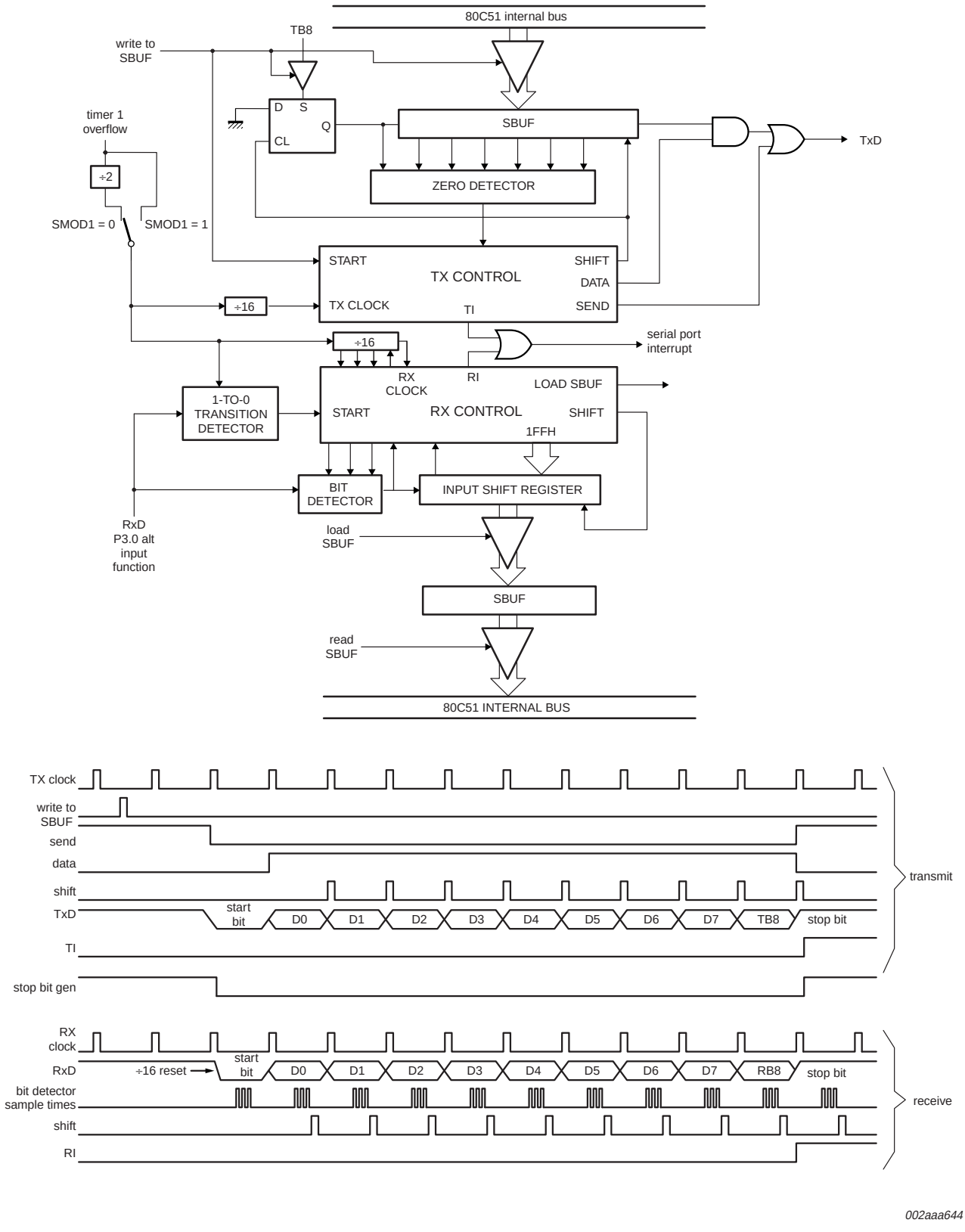


Fig 26. Serial port mode 3.

8.16 Watchdog timer

When enabled via the WDTE configuration bit, the Watchdog timer is operated from an independent, fully on-chip oscillator in order to provide the greatest possible dependability. When the Watchdog feature is enabled, the timer must be fed regularly by software in order to prevent it from resetting the CPU, and it **cannot** be turned off. When disabled as a Watchdog timer (via the WDTE bit in the UCFG1 configuration register), it may be used as an interval timer and may generate an interrupt. The Watchdog timer is shown in Figure 27.

The Watchdog timeout time is selectable from one of eight values, nominal times range from 16 milliseconds to 2.1 seconds. The frequency tolerance of the independent Watchdog RC oscillator is ±37 %. The timeout selections and other control bits are shown in Tables 55 and 56. When the Watchdog function is enabled, the WDCON register may be written **once** during chip initialization in order to set the Watchdog timeout time. The recommended method of initializing the WDCON register is to first feed the Watchdog, then write to WDCON to configure the WDS[2:0] bits. Using this method, the Watchdog initialization may be done any time within 10 milliseconds after start-up without a Watchdog overflow occurring before the initialization can be completed.

Since the Watchdog timer oscillator is fully on-chip and independent of any external oscillator circuit used by the CPU, it intrinsically serves as an oscillator fail detection function. If the Watchdog feature is enabled and the CPU oscillator fails for any reason, the Watchdog timer will time out and reset the CPU.

When the Watchdog function is enabled, the timer is deactivated temporarily when a chip reset occurs from another source, such as a Power-on reset, brownout reset, or external reset.

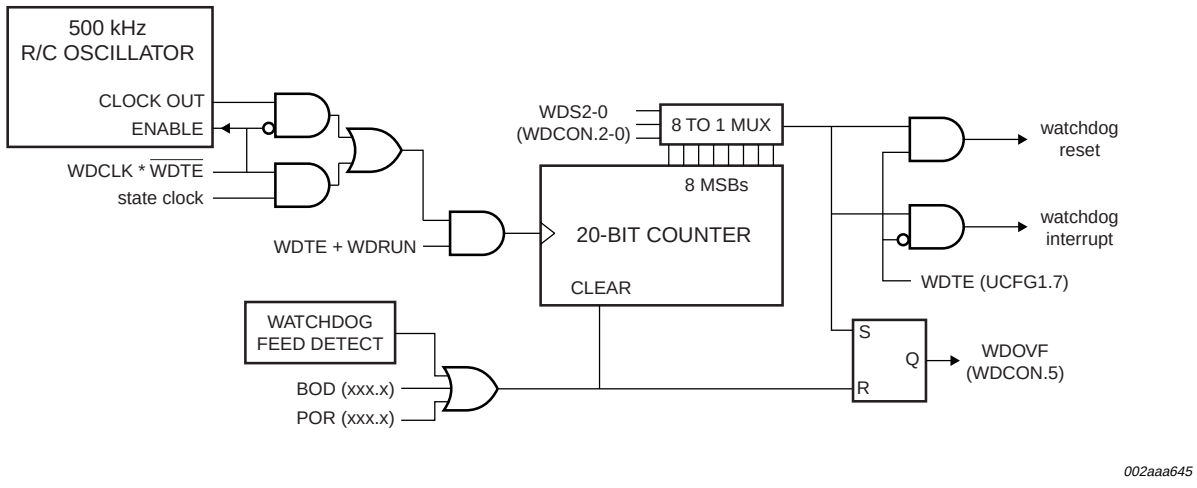


Fig 27. Block diagram of the Watchdog timer.

8.16.1 Watchdog feed sequence

If the Watchdog timer is running, it must be fed before it times out in order to prevent a chip reset from occurring. The Watchdog feed sequence consists of first writing the value 1Eh, then the value E1h to the WDRST register. An example of a Watchdog feed sequence is shown below.

```
WDFeed:
   mov   WDRST,#1eh    ; First part of Watchdog feed sequence.
   mov   WDRST,#0e1h   ; Second part of Watchdog feed sequence.
```

The two writes to WDRST do not have to occur in consecutive instructions. An incorrect Watchdog feed sequence does not cause any immediate response from the Watchdog timer, which will still time out at the originally scheduled time if a correct feed sequence does not occur prior to that time.

After a chip reset, the user program has a limited time in which to either feed the Watchdog timer or change the timeout period. When a low CPU clock frequency is used in the application, the number of instructions that can be executed before the Watchdog overflows may be quite small.

8.16.2 Watchdog reset

If a Watchdog reset occurs, the internal reset is active for approximately one microsecond. If the CPU clock was still running, code execution will begin immediately after that. If the processor was in Power-down mode, the Watchdog reset will start the oscillator and code execution will resume after the oscillator is stable.

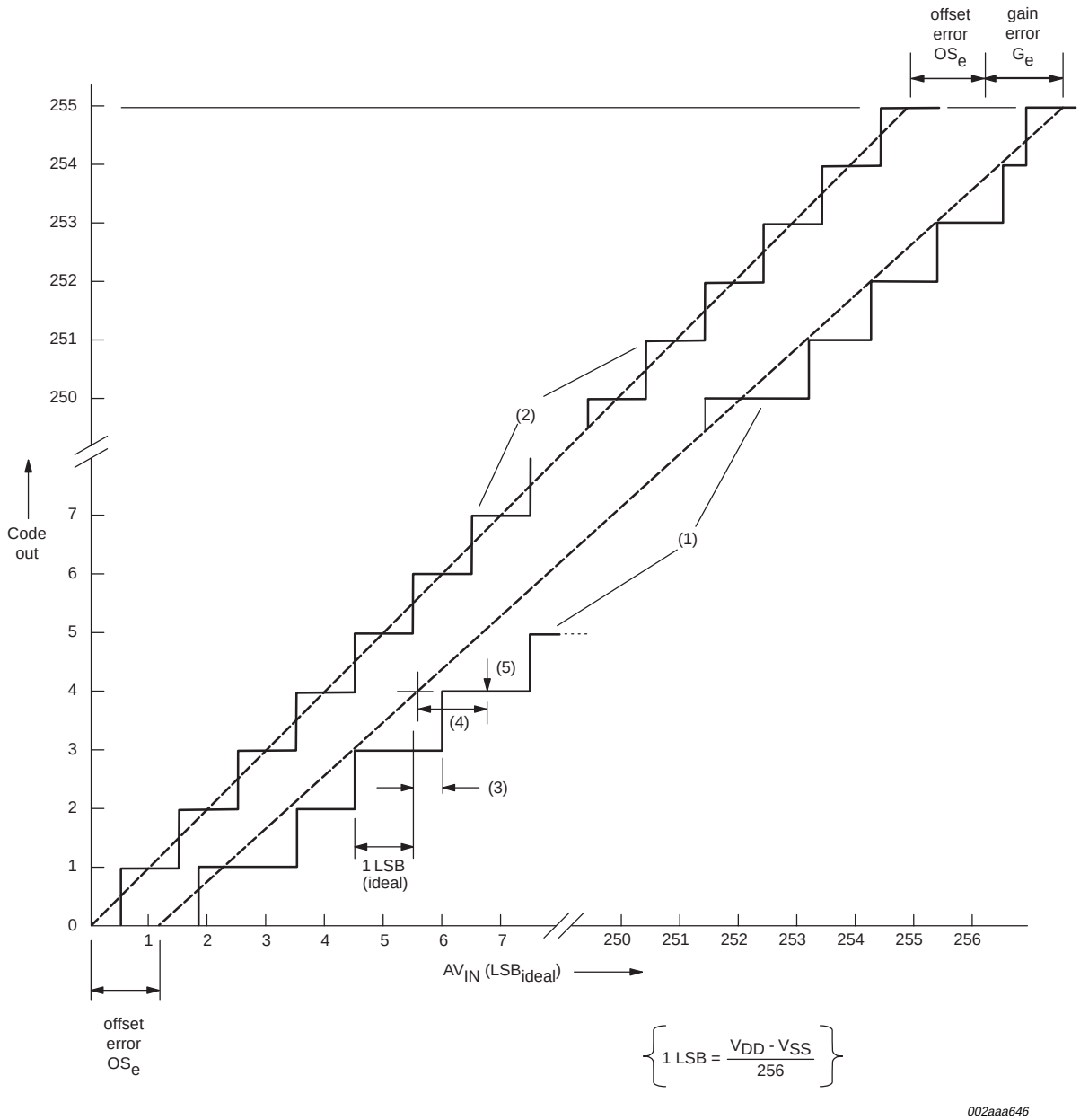
Table 55: WDCON - Watchdog timer control register (address A7H) bit allocation

Not bit addressable; Reset value: 30H for a Watchdog reset; 10H for other reset sources if the Watchdog is enabled via the WDTE configuration bit; 00H for other reset sources if the Watchdog is disabled via the WDTE configuration bit.

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	WDOVF	WDRUN	WDCLK	WDS2	WDS1	WDS0

Table 56: WDCON - Watchdog timer control register (address A7H) bit description

Bit	Symbol	Description
7, 6	-	Reserved for future use. Should not be set to ‘1’ by user programs.
5	WDOVF	Watchdog timer overflow flag. Set when a Watchdog reset or timer overflow occurs. Cleared when the Watchdog is fed.
4	WDRUN	Watchdog run control. The Watchdog timer is started when WDRUN = 1 and stopped when WDRUN = 0. This bit is forced to ‘1’ (Watchdog running) if the WDTE configuration bit = 1.
3	WDCLK	Watchdog clock select. The Watchdog timer is clocked by CPU clock / 6 when WDCLK = 1 and by the Watchdog RC oscillator when WDCLK = 0. This bit is forced to 0 (using the Watchdog RC oscillator) if the WDTE configuration bit = 1.
2 to 0	WDS[2:0]	Watchdog rate select.



(1) Example of an actual transfer curve.

(2) The ideal transfer curve.

(3) Differential non-linearity (DL_e).

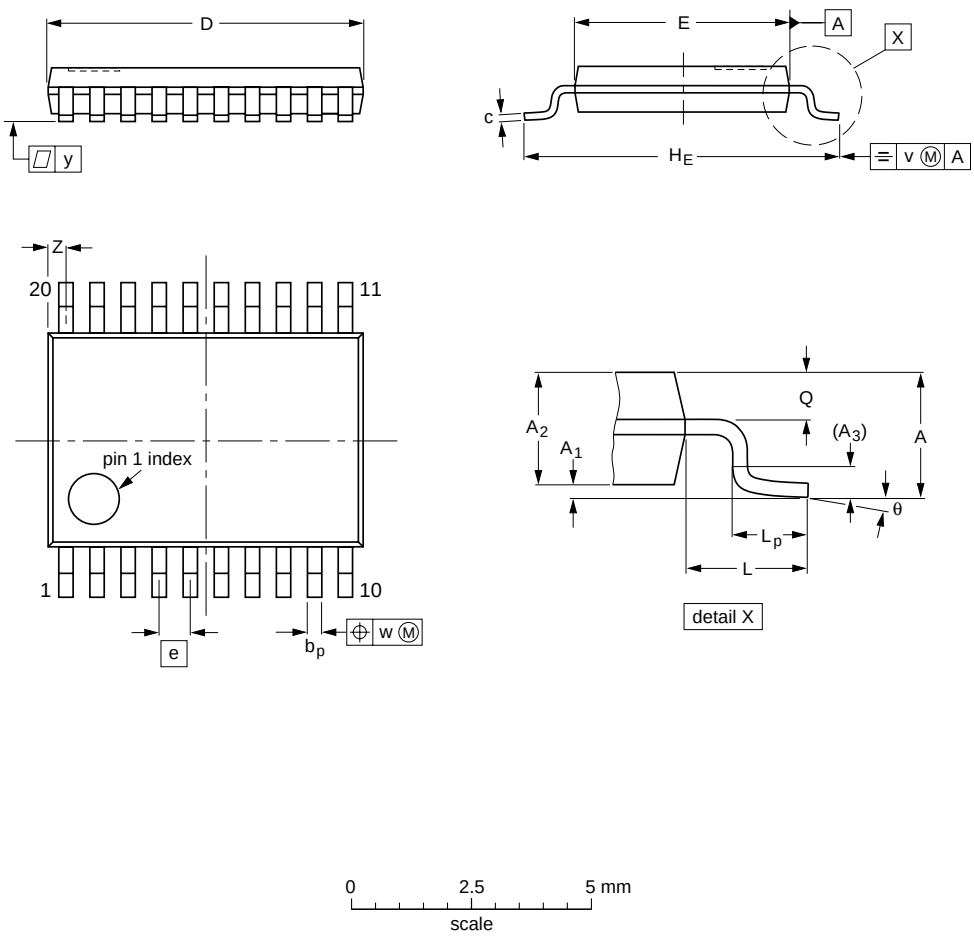
(4) Integral non-linearity (IL_e).

(5) Center of a step of the actual transfer curve.

Fig 28. A/D conversion characteristics.

13. Package outline

TSSOP20: plastic thin shrink small outline package; 20 leads; body width 4.4 mm **SOT360-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D (1)	E (2)	e	H_E	L	L_p	Q	v	w	y	Z (1)	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.5 0.2	8^o 0^o

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT360-1		MO-153				~~99-12-27~~ 03-02-19

Fig 31. SOT360-1

15. Data sheet status

Level	Data sheet status[1]	Product status[2][3]	Definition
I	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.
II	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.
III	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. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

[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.

[3] For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

16. 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.

17. 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 in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence 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.

18. Licenses

Purchase of Philips I²C components

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 specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

Contact information

For additional information, please visit **http://www.semiconductors.philips.com**.
For sales office addresses, send e-mail to: **sales.addresses@www.semiconductors.philips.com**. **Fax: +31 40 27 24825**