DATA SHEET

P87CL888
80C51 Ultra Low Power (ULP) telephony controller

Product specification
Supersedes data of 2000 Sep 25

2002 Apr 24

Philips
Semiconductors

PHILIPS

CONTENTS

1 FEATURES

- Full static asynchronous handshake 80C51 CPU; enhanced 8-bit architecture with:
  - Standard 80C51 instruction set
  - CPU speed independent of clock frequency, average speed: 4.8 Mips at 3.0 V
  - Non-page oriented instructions
  - Direct addressing
  - Four 8-byte RAM register banks
  - Stack depth limited only by available internal RAM (maximum 256 bytes)
  - Multiply, divide, subtract and compare instructions.
- Twenty source, twenty vector interrupt structure with two priority levels, polarity and sensitivity choice
- Key pad interrupt:
  - Edge or level sensitive triggering selectable via software
  - Enabling low-power and low-EMI keyboard control.
- 32 general purpose I/O pins
- Timer 0 and Timer 1: two standard 16-bit timer/event counters
- Timer 2: 16-bit timer/event counter with capture, compare and auto-reload function
- Watchdog Timer
- Wake-up counter
- Idle and Power-down mode
- 8-kbyte blank One Time Programmable (OTP) memory or factory programmed OTP memory
- In-system programming of OTP memory
- Supply voltage: 1.8 to 3.6 V
- 256 bytes RAM
- Internal crystal oscillator
- Power-on reset circuitry with disable pin
- Reset I/O pin for external reset from master or to slave
- MSK modem including Manchester encoder/decoder with 2 digital outputs (by SW) for analog cordless telephones (standards CT0/CT1/CT1+)
- I^2C-bus master/slave (transmitter/receiver, maximum frequency 400 kHz)
- 8 bit digital-to-analog converter with two different outputs
- 6 inputs Comparator with input multiplexer (for successive approximation analog-to-digital conversion).

2 GENERAL DESCRIPTION

The P87CL888 is manufactured in an advanced CMOS technology. The P87CL888 is a member of the VTELX microcontroller family which is a low-power family of low-voltage 80CL51 microcontrollers with advanced features for telecom applications. The Philips exclusive, asynchronous handshaking technology has been used for the CPU implementation which makes the CPU to run at its maximum speed independent of the used crystal frequency.

The P87CL888 is especially suited for low cost analog cordless telephone applications (CT0, CT1 and CT1+ standards) and wired feature phones. For this purpose, functions like ADC/DAC, MSK modem, I^2C-bus and key interrupts are integrated on-chip.

The device is optimized for low power consumption. It has two software selectable modes for power reduction: Idle and Power-down. In addition, the clock to all unused peripheral blocks can be switched off.

The instruction set is based on that of the 80C51. The P87CL888 also functions as an arithmetic processor having facilities for both binary and BCD arithmetic plus bit-handling capabilities. The instruction set consists of over 100 instructions: 49 one-byte, 46 two-byte, and 16 three-byte. Due to the missing Port P2, there is no external data or memory access and the MOVX operations cannot be used.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
P87CL888T/000[1]	TSSOP48	plastic thin shrink small outline package; 48 leads; body width 6.1 mm	SOT362-1
P87CL888T/xxx[2]			

Notes

1. Blank OTP.
2. Factory-programmed OTP.

4 BLOCK DIAGRAM

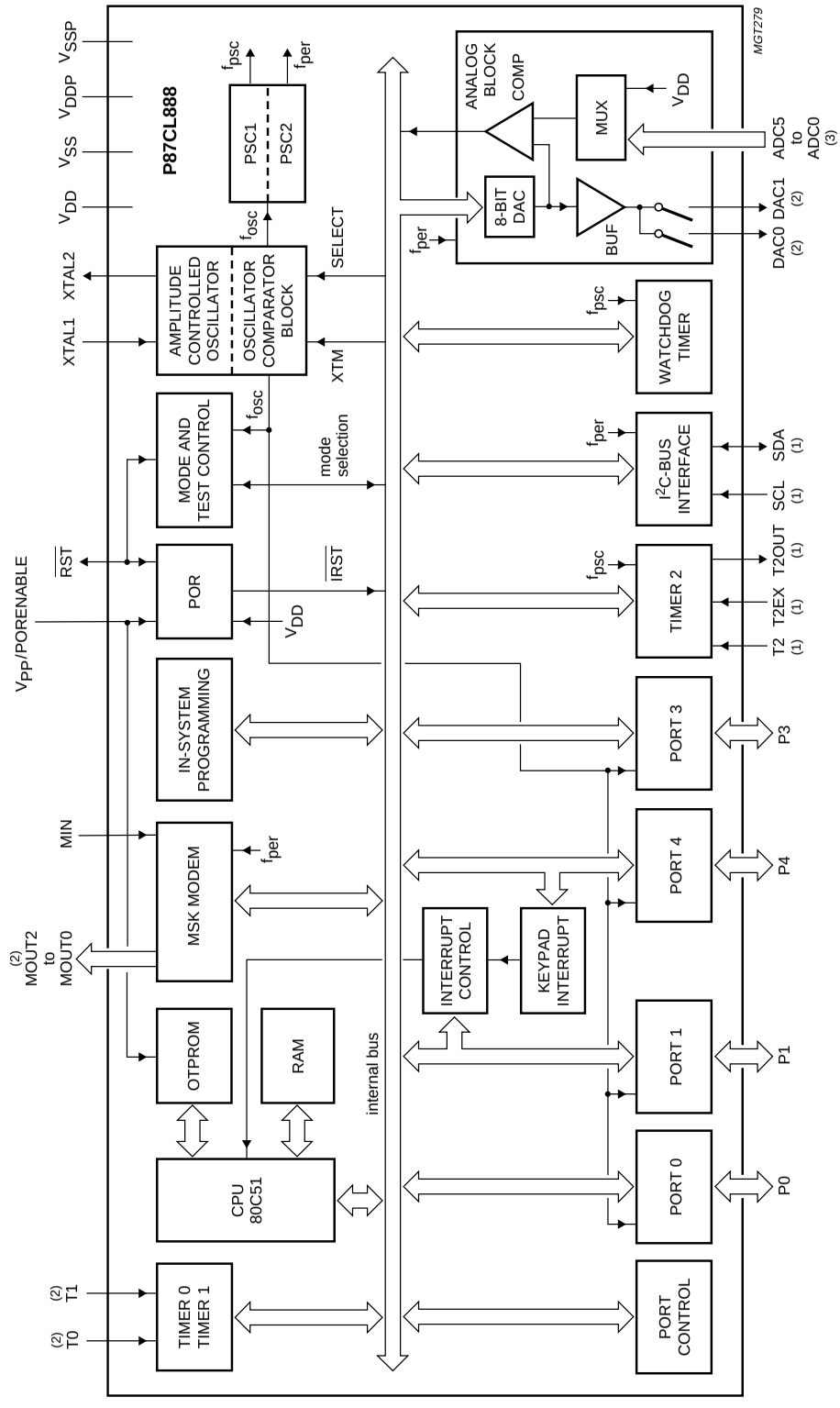


Fig.1 Simplified block diagram.

(1) Alternative function of Port 1.
(2) Alternative function of Port 3.
(3) ADC3 to ADC5: alternative function of Port 1;
ADC0 to ADC2: alternative function of Port 3.

5 PINNING INFORMATION

5.1 Pin configuration

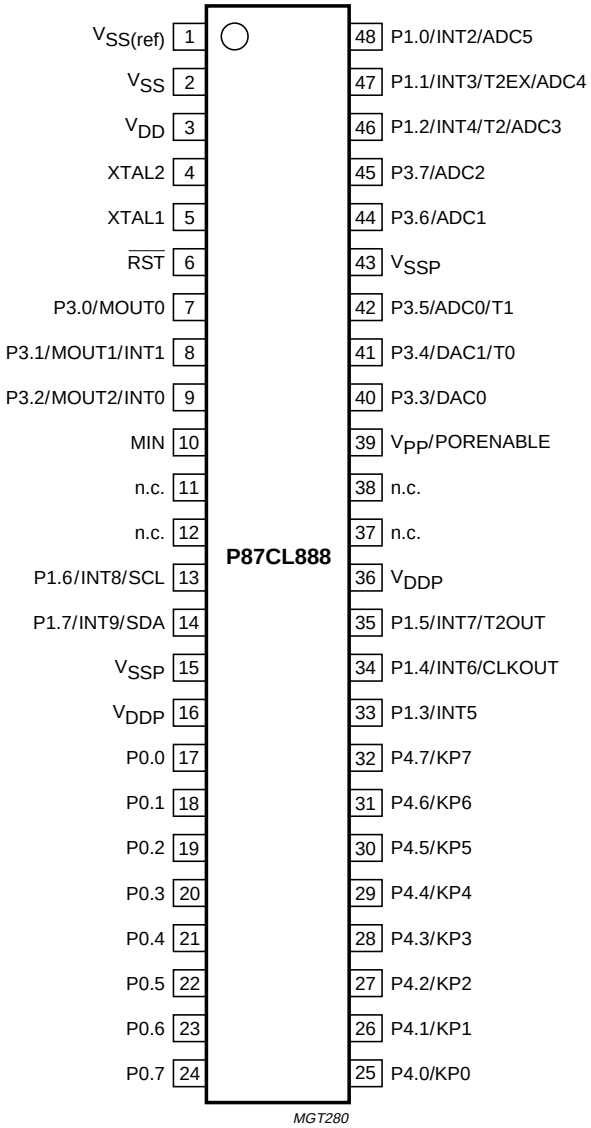


Fig.2 Pin configuration (TSSOP48/SOT362-1).

5.2 Pin description

Table 1 Pin description for TSSOP48 (SOT362-1)

SYMBOL	PIN	TYPE	DESCRIPTION
$V_{SS(ref)}$	1	S	negative reference supply voltage (for bandgap)
V_{SS}(1)	2	S	ground
V_{DD}(1)	3	S	power supply
XTAL2	4	O	crystal output
XTAL1	5	I	crystal input; external clock input
$\overline{RST}$	6	I/O	reset input/output pin; active LOW
P3.0/MOUT0	7	I/O	Port 3: P3.0 to P3.2; bidirectional I/O port with alternative functions MOUT2 to MOUT0: MSK outputs (mapped on the lower 3 bits of Port 3). P3.2 also serves as the external interrupt 0 input (INT1) and P3.1 as the external interrupt 1 input (INT0).
P3.1/MOUT1/INT1	8	I/O	
P3.2/MOUT2/INT0	9	I/O	
MIN	10	I	MSK input
n.c.	11, 12	–	not connected
P1.6/INT8/SCL	13	I/O	Port 1: P1.6 and P1.7; can only be used as open-drain output or high-impedance input. Alternative functions: INT8 and INT9, external interrupt 8 and 9. SCL and SDA I^2C-bus interface clock and data.
P1.7/INT9/SDA	14	I/O	
V_{SSP}(1)	15, 43	S	periphery (I/O) ground
V_{DDP}(1)	16, 36	S	periphery (I/O) positive supply voltage
P0.0 to P0.7	17 to 24	I/O	Port 0: 8-bit bidirectional I/O port. Every port pin can be used as open-drain, standard port, high-impedance input or push-pull output.
P4.0/KP0 to P4.7/KP7	25 to 32	I/O	Port 4: 8-bit bidirectional I/O port with multiplexed alternative functions Key Pad Interrupts: KP0 to KP7. Every port can be used as open-drain, standard port, high-impedance input or push-pull output.
P1.3/INT5	33	I/O	Port 1: P1.3 to P1.5; bidirectional I/O port with alternative functions INT5, INT6 and INT7: external interrupt 5 to 7. P1.4 also serves as auxiliary clock output (CLKOUT). P1.5 also serves as the Timer 2 output (T2OUT).
P1.4/INT6/CLKOUT	34	I/O	
P1.5/INT7/T2OUT	35	I/O	
n.c.	37, 38	–	not connected
V_{PP}/PORENABLE	39	S	high voltage input pin for OTP programming/power-on reset enable signal
P3.3/DAC0	40	I/O	Port 3: P3.3 to P3.5; bidirectional I/O port with alternative functions DAC0, DAC1: DAC outputs and ADC0: ADC input 1. P3.4 also serves as the Timer 0 external count input (T0). P3.5 also serves as the Timer 1 external count input (T1).
P3.4/DAC1/T0	41	I/O	
P3.5/ADC0/T1	42	I/O	
P3.6/ADC1	44	I/O	Port 3: P3.6 and P3.7; bidirectional I/O port with alternative functions ADC1 and ADC2: ADC inputs 1 and 2.
P3.7/ADC2	45	I/O	
P1.2/INT4/T2/ADC3	46	I/O	Port 1: P1.2 to P1.0; bidirectional I/O port with alternative functions INT4, INT3 and INT2: external interrupts 4, 3 and 2. P1.2 also serves as Timer 2 input (T2) or ADC input 3 (ADC3). P1.1 also serves as Timer 2 external input (T2EX) or ADC input 4 (ADC4). P1.0 also serves as ADC input 5 (ADC5).
P1.1/INT3/T2EX/ADC4	47	I/O	
P1.0/INT2/ADC5	48	I/O	

Note

1. For high current drive capability on I/Os all supply pins should be connected.

6 FUNCTIONAL DESCRIPTION

6.1 Oscillator circuitry

The on-chip Amplitude Controlled Oscillator (ACO) circuitry is a single-stage inverting amplifier biased by an internal feedback resistor R_{fb}. The oscillator circuit is shown in Fig.3.

Two comparators with different characteristics can be used with the on-chip crystal oscillator. The first one is an analog comparator built around a differential amplifier and is intended to be selected when an external ceramic or crystal resonator is connected to the chip.

The other comparator has a Schmitt-trigger input with a bigger hysteresis which is especially useful when the P87CL888 is driven from an external clock source.

Two bits in the SYSCON SFR, SELECT and XTM, are used to configure the oscillator. The SELECT bit (SYSCON.1) enables the analog comparator or the hysteresis comparator.

With XTM (SYSCON.0) = 1 (or in Power-down mode; PCON.1 = 1) the oscillator is switched off and the current consumption of the oscillator is reduced to zero.

Table 2 Comparator select bits in SYSCON SFR

SELECT	XTM	DESCRIPTION
0	0	oscillator enabled; analog comparator enabled
0	1	don't use
1	0	oscillator enabled; hysteresis comparator enabled
1	1	oscillator stopped; hysteresis comparator enabled

6.1.1 CLOCK OSCILLATOR CONNECTIONS

No external components are needed, when a quartz crystal is used to drive the oscillator. When an external ceramic resonator is used to drive the oscillator, external components may be required depending upon the used ceramic resonator; refer to the product specification. Two different resonator configurations are shown in Figs 4a and 4b.

To drive the device with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 floating, as shown in Fig.4c. If the amplitude of the input signal is less than V_{DD} to V_{SS} or if a sine wave is applied, capacitive decoupling is needed as shown in Fig.4d.

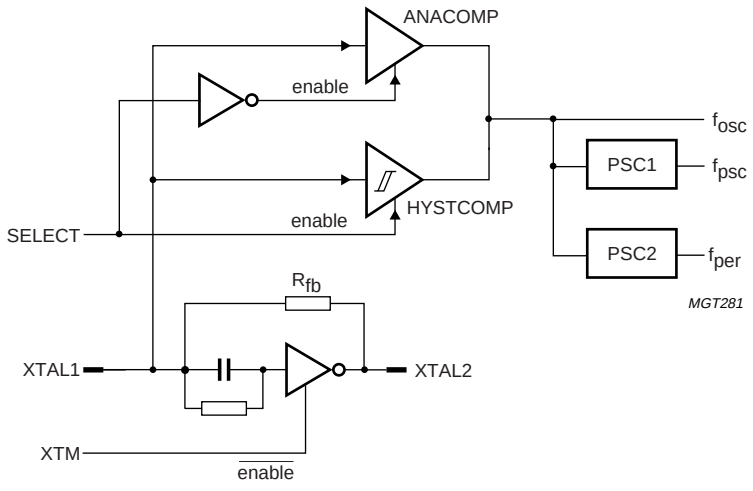


Fig.3 Oscillator.

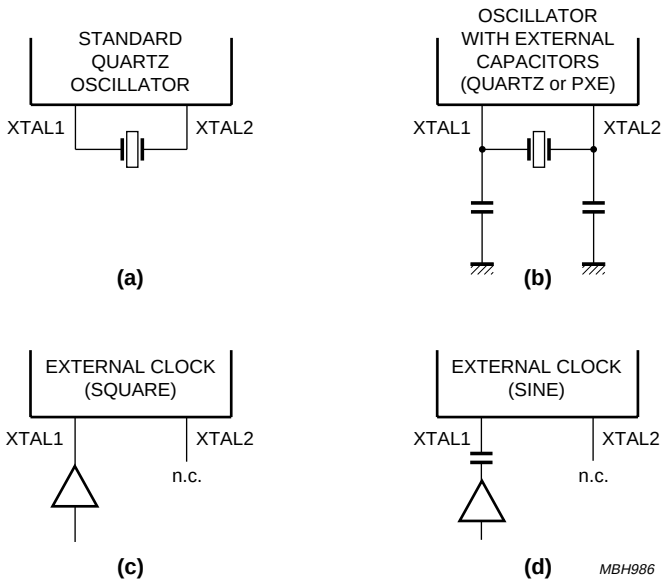
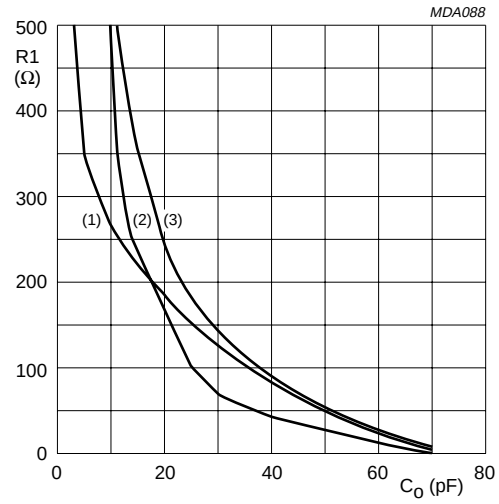


Fig.4 Alternative oscillator configurations.

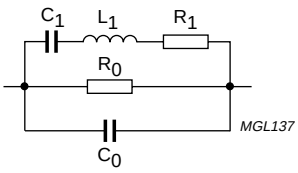
6.1.2 Resonator requirements for the ACO

For correct operation of the oscillator, the values of R_1 and C_0 of the chosen resonator (quartz or PXE) must be below the line shown in Fig.5a. The value of the parallel resistor R_0 must be less than 47 kΩ. The wiring between chip and resonator should be kept as short as possible. Recommended resonator types are:

- CSA 3.58MG (supplier Murata)
- FCR3.58M5 (supplier TDK).



a. Resonator curves.



b. Resonator equivalent circuit.

C_{1e} and C_{2e} are the external load capacitances; normally not needed due to integrated load capacitances of typically 10 pF

(1) $C_{1e} = C_{2e} = 22$ pF.

(2) $C_{1e} = C_{2e} = 0$ pF.

(3) $C_{1e} = C_{2e} = 12$ pF.

Fig.5 Resonator curves and equivalent circuit.

6.1.3 ON-CHIP CLOCKS

The microcontroller does not need a clock signal to run instructions, because the CPU is built using the Philips exclusive handshake technology. The peripheral blocks however are connected to a clock for synchronization with the outside world (e.g. MSK) or for a timed application (e.g. Timer 2). The block related SFRs (peripheral function) are therefore updated/modified with the applied clock. Two prescalers (PSC1 and PSC2) are implemented which allow the generation of two programmable clock signals f_{psc} and f_{per} for internal usage.

Signal f_{psc} from PSC1 is the default input clock of the timer blocks. The complete timer functionality is specified in the Section 6.6. Connected timers are the three 16-bit timers Timer 0, Timer 1 and Timer 2 and the 8-bit Watchdog Timer. The time interval of the connected timers can be adjusted by programming of PSC1. The output frequency f_{psc} can be changed by selecting the division factor with the bits PRESC.2, PRESC.1 and PRESC.0 (see Table 7).

All peripheral blocks, which require a clock signal: MSK, I^2C-bus interface and ADC, are connected to the clock signal f_{per}. PSC2 can be programmed by setting bits PRESC.4 and PRESC.3 (see Table 7). The choice of the division factor must guarantee that all of the peripheral blocks are within their specification, specially if an external clock source of up to 12 MHz is applied.

Additionally Timer 1 and Timer 0 have a multiplexer on the clock input to choose from 4 different clock sources.

The multiplexers are switched by setting user controllable bits in the SYSCON SFR (bits 7 to 4). In the default setting both timers are incrementing on the clock signal f_{psc} coming from PSC1. Timer 1 and Timer 0 can however also run on clock signal f_{per} coming from PSC2. If used in the proper way this flexibility on the timer input sources can substantially contribute to a decrease in power consumption. Ideas and tips to reduce power consumption are given in Chapter 9.

The clock source of Timer 1 and Timer 0 can also be switched to an external clock input signal T1or T0 which are multiplexed with one of the device input pins.

This mode is also functional even when there is no system clock available. This means when a clock source is supplied on a port pin the Timer 1 or 0 can count and generate interrupts even when the chip is in Power-down mode. More details are specified in Section 6.6.

The last multiplexer input to the Timer 1 and Timer 0 is an auxiliary mode which can be used to obtain the operation speed from the handshake CPU. If this mode is activated for the Timer 1 input source, the timer increments on every ROM request. This means the timer increments by three for a three byte instruction and by two for a two byte instruction etc. If the auxiliary mode is activated for Timer 0 the timer increments on every instruction executed by the CPU. This means the timer register holds the number of instructions executed in a certain time frame. More ideas and tips how these clock source modes can be used together with the handshake CPU can be found in Chapter 9.

Table 3 Timer 1 input source select modes
Bits T1SRC[1:0] are defined in SYSCON SFR.

T1SRC1	T1SRC0	DESCRIPTION
0	0	f_{psc} is the Timer 1 clock input
0	1	T1 is the Timer 1 clock input
1	0	the ROMreq signal is the Timer 1 clock input
1	1	f_{per} is the Timer 1 clock input

Table 4 Timer 0 input source select modes
Bits T0SRC[1:0] are defined in SYSCON SFR.

T0SRC1	T0SRC0	DESCRIPTION
0	0	f_{psc} is the Timer 0 clock input
0	1	T0 is the Timer 0 clock input
1	0	the InstrReq signal is the Timer 0 clock input
1	1	f_{per} is the Timer 0 clock input

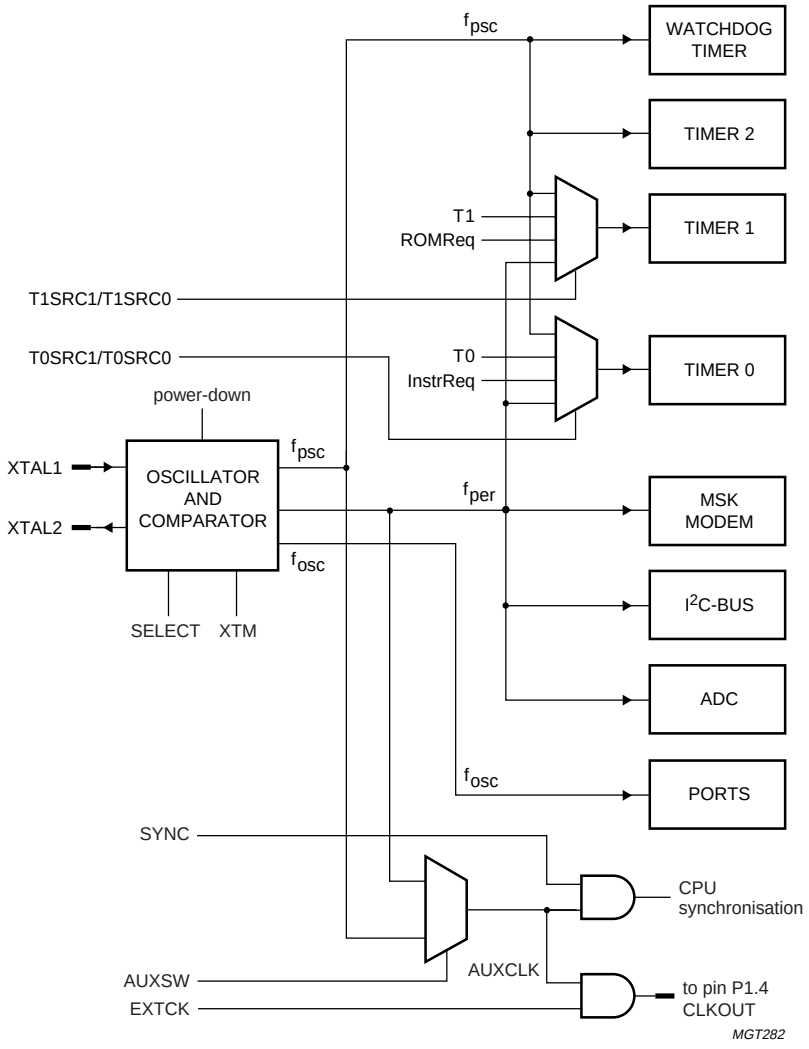


Fig.6 Clock overview.

6.1.3.1 Prescaler Register (PRESC)

Reset value of PRESC SFR is XXX0 0000 (division factor 1 for PSC1 and PSC2).

Table 5 Prescaler Register (SFR address F3H)

7	6	5	4	3	2	1	0
EXTCK	AUXSW	SYNC	PRESC.4	PRESC.3	PRESC.2	PRESC.1	PRESC.0

Table 6 Description of PRESC bits

BIT	SYMBOL	DESCRIPTION
7	EXTCK	switches AUXCLK to device pin P1.4 (CLKOUT)
6	AUXSW	Auxiliary Clock Switch; AUXSW = 0: AUXCLK equals f_{psc}; AUXSW = 1: AUXCLK equals f_{per}
5	SYNC	switches the CPU to synchronous mode
4 to 0	PRESC.[4:0]	these bits define the division factors for PSC1 and PSC2; see Table 7

Table 7 Division factors for PSC1 and PSC2

DIVISION FACTOR		PRESC.4	PRESC.3	PRESC.2	PRESC.1	PRESC.0
PSC2 (f_{osc}/f_{per})	PSC1 (f_{osc}/f_{psc})					
1	–	0	0	X	X	X
2	–	0	1	X	X	X
4	–	1	0	X	X	X
8	–	1	1	X	X	X
–	1	X	X	0	0	0
–	2	X	X	0	0	1
–	4	X	X	0	1	0
–	6	X	X	0	1	1
–	8	X	X	1	0	0
–	10	X	X	1	0	1
–	12	X	X	1	1	0
–	16	X	X	1	1	1

6.1.4 AUXILIARY CLOCK SIGNAL MODES

The 3 most significant bits in the Prescaler Register (PRESC[7:5]; see Table 5 and 6) are used to enable additional clocking options. A multiplexer is implemented (see Fig.6) to choose between f_{psc} and f_{per} as source for AUXCLK. The multiplexer is operated by bit AUXSW (PRESC.6). With bit EXTCK (PRESC.7) the AUXCLK is fed to pin P1.4 (CLKOUT) for external use (initialize the port accordingly). Setting bit SYNC (PRESC.5) connects the AUXCLK to the instruction request input of the CPU. In this way the CPU is synchronised to the clock and an instruction is executed at every clock pulse of AUXCLK. In order to obtain exactly one instruction per clock cycle the period for AUXCLK must always be longer than the length of the slowest instruction.

6.1.5 System Control Register (SYSCON)

Table 8 System Control Register (SFR address B4H)

7	6	5	4	3	2	1	0
T1SRC1	T1SRC0	T0SRC1	T0SRC0	–	–	SELECT	XTM

Table 9 Description of SYSCON bits

BIT	SYMBOL	DESCRIPTION
7	T1SRC1	these 2 bits select the clock source for Timer 1; see Table 3
6	T1SRC0	
5	T0SRC1	these 2 bits select the clock source for Timer 0; see Table 4
4	T0SRC0	
3	–	do not use
2		
1	SELECT	comparator select bit; see Table 2
0	XTM	oscillator disable bit; see Table 2

6.2 The CPU

6.2.1 GENERAL

Ultra Low Power (ULP), points to the special 80C51 CPU architecture used in this device allowing significant power saving.

The CPU of the P87CL888 is realized in the Philips exclusive asynchronous handshaking technology, which is completely different to usual implementations of this core. The processor does not need a clock signal to run instructions. Every function within the CPU is self timed and always runs at the maximum speed that the silicon die under the current operating conditions allows (supply voltage and temperature). The advantage is the combination of a high computing power with reduced average power consumption and low EMC noise generation. Details about speed and energy consumption per instruction can be found in Chapter 8.

Summary of the CPU features:

- No CPU clock is needed
- Only useful bytes are fetched from the program memory; the dummy read cycles which exist in the standard 80C51 have been eliminated to save power
- To further speed up the program execution; there is always a pre-fetch of the next byte of code from memory during the execution of the current instruction; in the case of a jump the pre-fetched byte is discarded
- In Idle mode the CPU power is reduced to leakage; only the enabled peripheral blocks consume power but can be switched off independently
- The only need for a clock is as a timing reference for timers/counters and to generate the timing for the I/O lines to synchronise with the off-chip world.

6.2.2 RESET OPERATION

There are three possibilities to reset the CPU (see Fig.7):

- Power-On Reset (POR)
- Watchdog Timer reset
- External reset via I/O pin $\overline{RST}$.

If an internal reset is executed (POR or Watchdog Timer), the reset pin $\overline{RST}$ will be pulled to ground which can be used as reset signal for other ICs. The reset pin is LOW for at least 1024 clock cycles, and released 16 clock cycles prior to first code fetch (see Figs 9, 10 and 11).

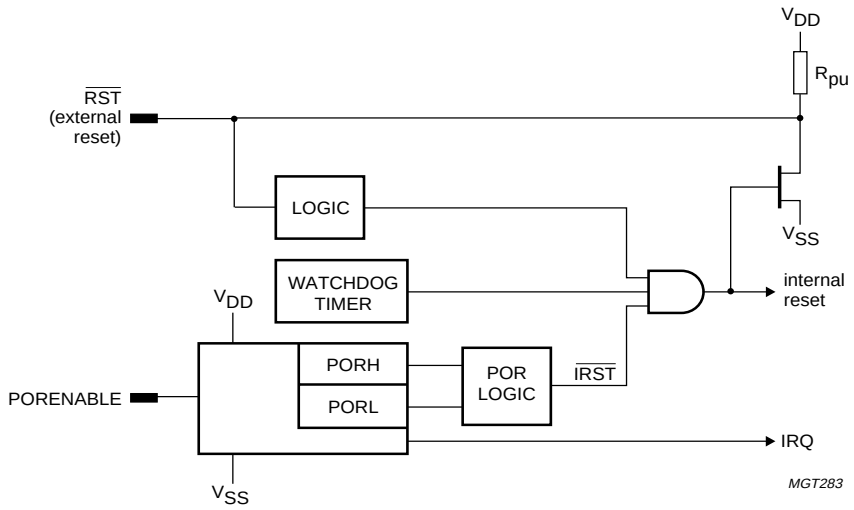


Fig.7 Reset sources.

6.2.2.1 Power-On Reset (POR)

The device has an integrated power-on reset logic with a hysteresis between the two levels PORL and PORH. The value of PORL and PORH is chosen in such a way that a good start-up of the microcontroller is always guaranteed. If the transistor thresholds are high, caused by silicon processing, accordingly the POR levels will also be higher.

The power-on reset circuitry can be disabled by setting the input pin V_{PP}/PORENABLE LOW. The IC must now be reset by applying a reset pulse to the $\overline{RST}$ I/O pin. Note that not all device functionality is guaranteed for V_{DD} voltages below PORL.

When V_{DD} falls below the level of PORH for longer then t_{dip} an 'Early Warning' interrupt signal IRQ is produced, and if V_{DD} falls below PORL for longer then t_{dip} an internal reset signal $\overline{IRST}$ will be generated.

If the level crossings are shorter then t_{dip} the spike will be ignored (see Fig.8). The delay t_{dip} is typical 50 ms long, analog implemented and a compromise between precision and current consumption.

After an Early Warning interrupt the microcontroller should terminate any processing as fast as possible and prepare for the reset. The Early Warning interrupt flag cannot be cleared by software, it will be cleared by hardware when V_{DD} rises above the PORH level before the interrupt is served. A reset will also clear the interrupt flag by hardware.

Only for slow falling V_{DD} there is enough time to serve the interrupt. For fast falling V_{DD} the circuit will go hardware controlled into the reset state.

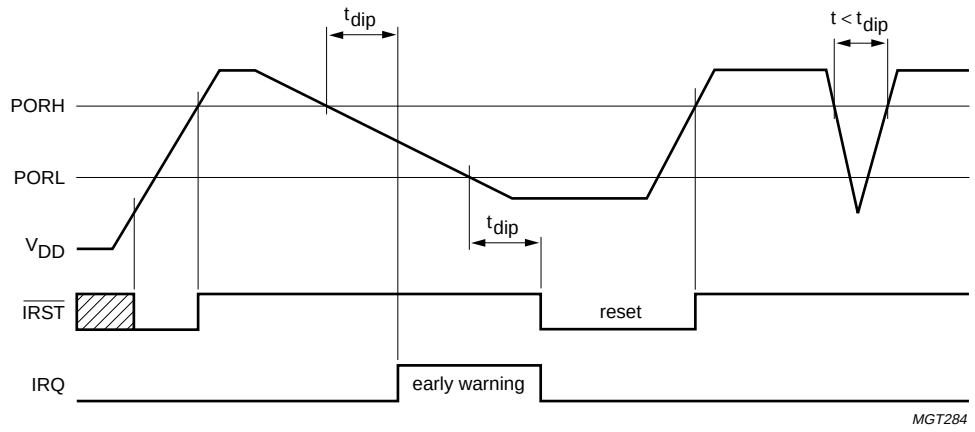


Fig.8 Power-on reset circuit: $\overline{IRST}$ and IRQ generation.

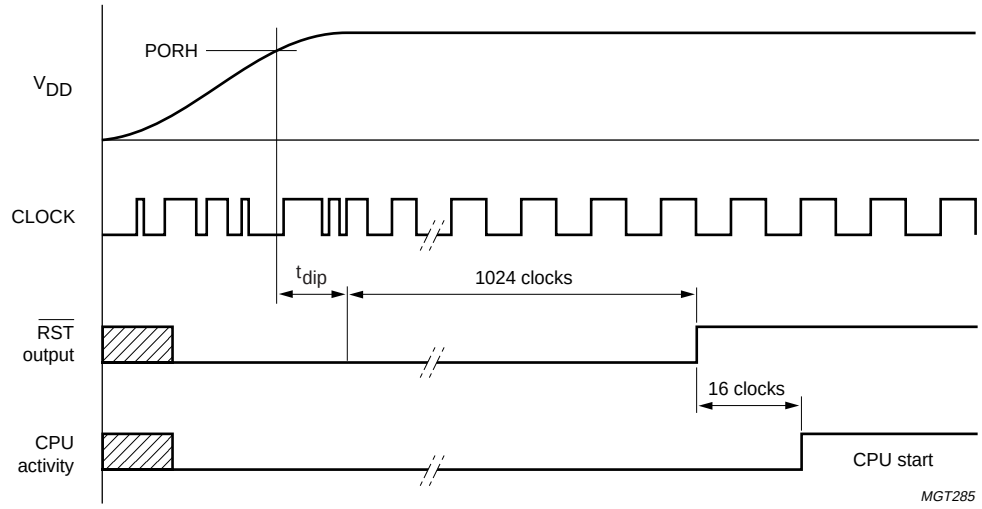


Fig.9 Power-on reset timing.

6.2.2.2 Watchdog Timer reset

If the Watchdog Timer expires, it will trigger a reset.

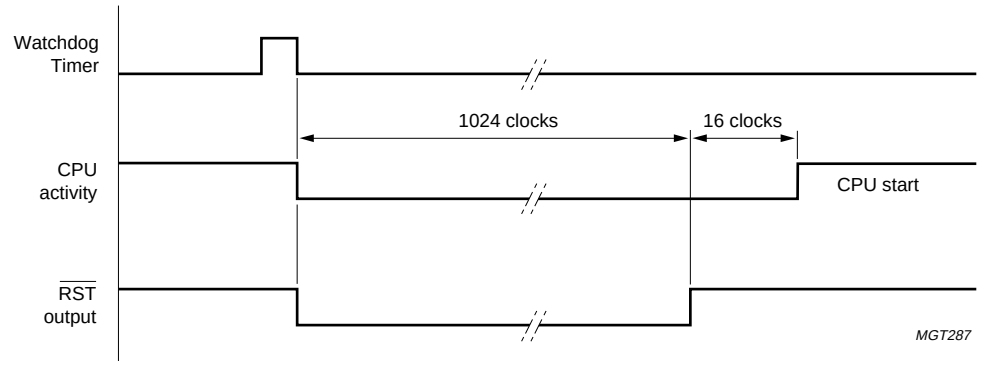
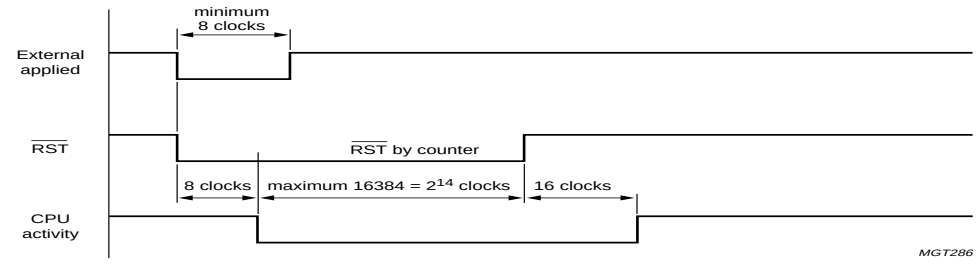


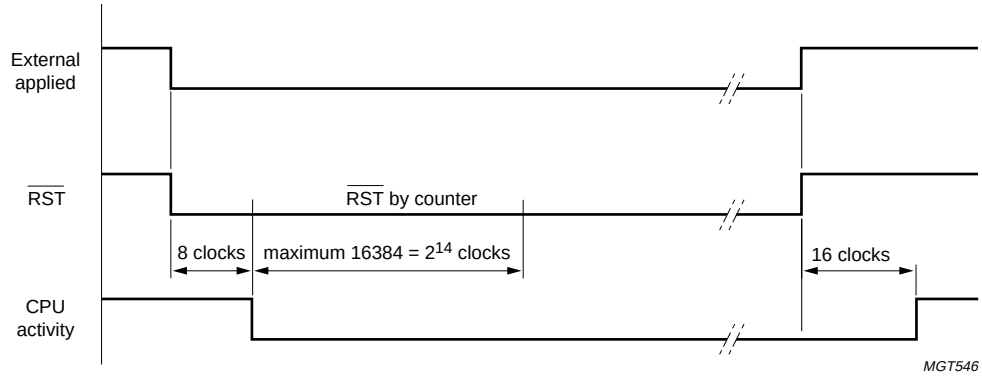
Fig.10 Watchdog Timer reset timing.

6.2.2.3 External reset via $\overline{RST}$

An external device can cause a chip reset, if the reset pin $\overline{RST}$ is pulled to ground.



a. Short external reset.



b. Long external reset.

Fig.11 External reset.

6.2.3 IDLE AND POWER-DOWN OPERATION

Idle and Power-down are power saving modes of the microcontroller that can be activated when no CPU activity is required. These two modes are extremely useful for the asynchronous CPU, because they offer the possibility to profit from the speed of the CPU and to save power as soon as the task is finished. Idle mode stops the code execution of the CPU, but the internal oscillator remains active, and also all peripheral functions connected to the on-chip clock signal. Unused blocks can be switched off independently. However, during Power-down mode the clock oscillator is stopped and therefore also all peripheral blocks will stop their activity.

6.2.3.1 Idle mode

The following functions remain active during Idle mode:

- Timer 0, Timer 1 and Timer 2
- Wake-up counter
- Watchdog Timer counter
- MSK modem
- ADC/DAC
- POR
- I^2C-bus interface
- External interrupt
- Key pad interrupt.

The instruction that sets PCON.0 (PCON SFR) is the last instruction executed in the normal operating mode before the Idle mode is activated. The RAM and all of the registers are preserved and maintain their data during Idle mode: the CPU status, the Stack Pointer, Program Counter, Program Status Word and Accumulator.

There are two ways to terminate the Idle mode:

- Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware thus terminating the Idle mode. The interrupt is serviced, and following the RETI instruction, the next instruction to be executed will be the one following the instruction that put the device in the Idle mode.
- The second way of terminating the Idle mode is with an internal or external hardware reset. Reset redefines all SFRs but does not affect the on-chip RAM. Possible sources of an internal reset in Idle mode are:
  - Watchdog Timer reset if the watchdog had expired
  - POR.

6.2.3.2 Power-down mode

The instruction that sets PCON.1 (PCON SFR) is the last instruction executed in the normal operating mode before the Power-down mode is activated. During Power-down mode, the RAM and all of the registers maintain their data: the CPU status, the Stack Pointer, Program Counter, Program Status Word and Accumulator.

There are two ways to terminate the Power-down mode:

- Activation of any of the below listed interrupts will cause PCON.1 to be cleared by hardware thus terminating the Power-down mode. The interrupt is serviced, and following the RETI instruction, the next instruction to be executed will be the one following the instruction that put the device in the Power-down mode. Interrupts which can generate a wake-up from power-down:
  - External interrupts (INT0 to INT9)
  - Timer 0 and Timer 1: only when pins T0 and T1 are used as the external timer source input (SYSCON SFR bit 7 to 4)
  - Low-voltage detection interrupt
  - Key pad interrupt.
- The second way of terminating the Power-down mode is with an internal or external hardware reset. Reset does not affect the on-chip RAM, but all SFRs are set to the default value. Possible source of an internal reset in power-down is: POR.

6.2.3.3 Power Control Register (PCON)

The reduced power modes are activated by software using this Special Function Register. PCON is not bit addressable.

Table 10 Power Control Register (SFR address 87H; reset value = 0000 0000)
Bits PCON[7:2] are reserved and must be kept to logic 0.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	PD	IDL

Table 11 Reduced power modes selection

PD	IDL	DESCRIPTION
0	0	CPU running
0	1	activates the Idle mode
1	0	activates the Power-down mode
1	1	

6.2.4 CPU START-UP TIMING

6.2.4.1 CPU start-up after reset

Three possibilities how the CPU can start executing code after a reset phase are described below.

When the CPU is triggered to wake-up after a power-on reset (see Fig.9), the clock oscillator usually needs some time to ramp up. To allow the oscillator to stabilize the CPU contains a down counter for a fixed delay of 1024 + 16 clock cycles. After this delay the CPU starts with code execution.

When CPU start-up is initiated from an external reset (see Fig.11), the down counter is not initialized and the time between reset going active and first code execution can be maximum 16400 clock cycles.

When a CPU start-up is after a Watchdog Timer reset (see Fig.10), the $\overline{RST}$ pin will be pulled low for 1024 clock cycles. Another 16 clocks later the CPU will start executing code.

6.2.4.2 CPU start-up after power-down

After wake-up from Power-down mode (see Fig.12) the user has the possibility to shorten the start-up time by programming the Wake-up Counter Register (WKCON). This can be useful when an external clock source is used instead of the on-chip oscillator, or when the accuracy of the time reference is not needed immediately after a restart. This feature enables power saving and fast wake-up in applications where the CPU frequently goes in Power-down mode. The wake-up delay can be calculated as shown in Table 13.

Table 12 Wake-up Counter Register (WKCON; SFR address DDH)

7	6	5	4	3	2	1	0
WKCON.7	WKCON.6	WKCON.5	WKCON.4	WKCON.3	WKCON.2	WKCON.1	WKCON.0

Table 13 Description of WKCON bits

BIT	SYMBOL	DESCRIPTION
7 to 0	WKCON.[7:0]	The wake-up delay can be calculated as follows: Wake-up delay = $1024 - 4 \times$ WKCON Where WKCON is the content of the Wake-up Counter Register. WKCON = 00H: (default) wake-up delay = 1024 clocks WKCON = CCH: wake-up delay = 208 clocks WKCON = FFH: wake-up delay = 4 clocks.

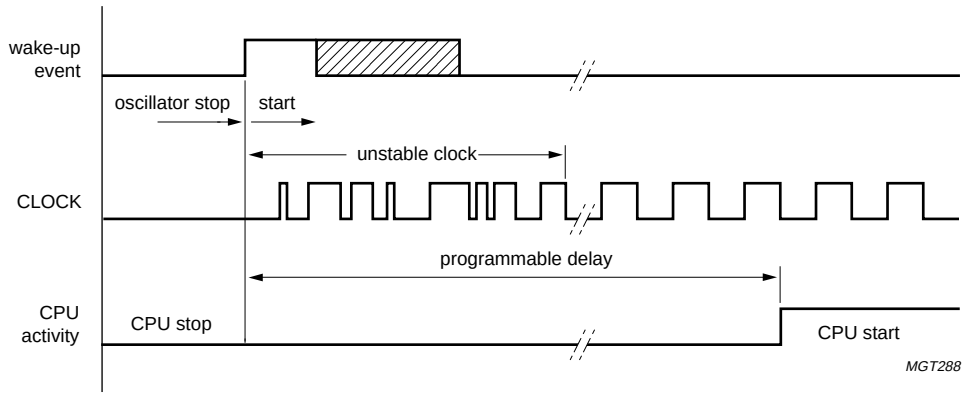


Fig.12 Wake-up timing from power-down.

6.3 Interrupt controller

In order to service interrupt requests coming from external events and from the on-chip peripherals the P87CL888 offers a 20 source, two priority level nested interrupt system. A detailed description of the interrupt process is given in the following sections. Table 14 shows the available interrupts with each vector address and Table 15 shows an overview of all the interrupt related SFRs. The detailed interrupt related SFR description can be found in Sections 6.3.4 to 6.3.10.

6.3.1 GENERAL

Each interrupt vector points to a separate location in program memory for its service routine. Each source can be individually enabled or disabled by its corresponding bit in the Interrupt Enable Registers (IEN0, IEN1 and IEN2). The priority level is selected via the Interrupt Priority Registers (IP0, IP1 and IP2). All available interrupts can be globally disabled or enabled.

The interrupt controller samples all active sources during one instruction cycle. Evaluation of the interrupts is then performed. A priority decoder decides which interrupt is serviced. Each interrupt has its own vector pointing to an 8 bytes long memory segment.

A low priority interrupt can be interrupted by a high priority interrupt, but not by another low priority interrupt i.e. only two interrupt levels are possible.

Between the RETI instruction (Return from Interrupt) and the execution of a next interrupt at least one instruction of the lower program level is executed. The interrupt service with different priorities is shown in Fig.13.

An interrupt is performed with a long subroutine call (LCALL) to a vector address, which is determined by the respective interrupt. During LCALL the Program Counter (PC) is pushed onto the stack. Returning from interrupt with RETI, the PC is popped from the stack.

In the event of several interrupts with the same priority level, the order of sequence in which they will be serviced is determined by the scanning order.

The interrupt highest in the scanning list will always be served first, interrupts lower in the scanning list will be served in the order as shown in Fig.14. No interrupt will be lost.

Table 14 Available interrupts (ordered by vector address) HW = hardware; SW = software.

SOURCE	SYMBOL	VECTOR (HEX)	CLEARED BY
INT 0	X0	0003	HW
Timer 0	T0	000B	HW
INT 1	X1	0013	HW
Timer 1	T1	001B	HW
I^2C-bus	S1	002B	SW
Timer 2	T2	0033	SW
INT2	X2	003B	SW
INT3	X3	0043	SW
INT4	X4	004B	SW
INT5	X5	0053	SW
INT6	X6	005B	SW
INT7	X7	0063	SW
INT8	X8	006B	SW
INT9	X9	0073	SW
MSK modem transmitter	MTI	0083	SW
MSK mode receiver	MRI	008B	SW
Low voltage detector	LVD	0093	HW
Key pad interrupt	KPI	00A3	SW
Analog-to-Digital Converter (ADC)	ADI	00AB	SW
Watchdog Timer	WDI	00B3	SW

Table 15 Interrupt related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
IEN0	Interrupt Enable Register 0	A8H	0000 0000
IEN1	Interrupt Enable Register 1 (INT2 to INT9)	E8H	0000 0000
IEN2	Interrupt Enable Register 2	F1H	0000 0000
IP0	Interrupt Priority Register 0	B8H	0000 0000
IP1	Interrupt Priority Register 1 (INT2 to INT9)	F8H	0000 0000
IP2	Interrupt Priority Register 2	F9H	0000 0000
IX1	External Interrupt Polarity Register 1	E9H	0000 0000
ISE1	External Interrupt Sensitivity Register 1	E1H	0000 0000
IRQ1	External Interrupt Request Flag Register 1	C0H	0000 0000

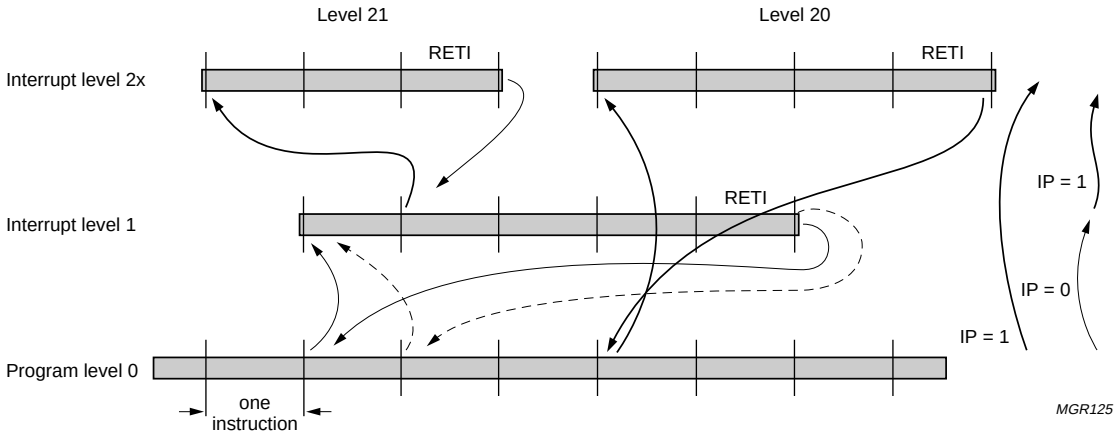


Fig.13 Interrupt hierarchy.

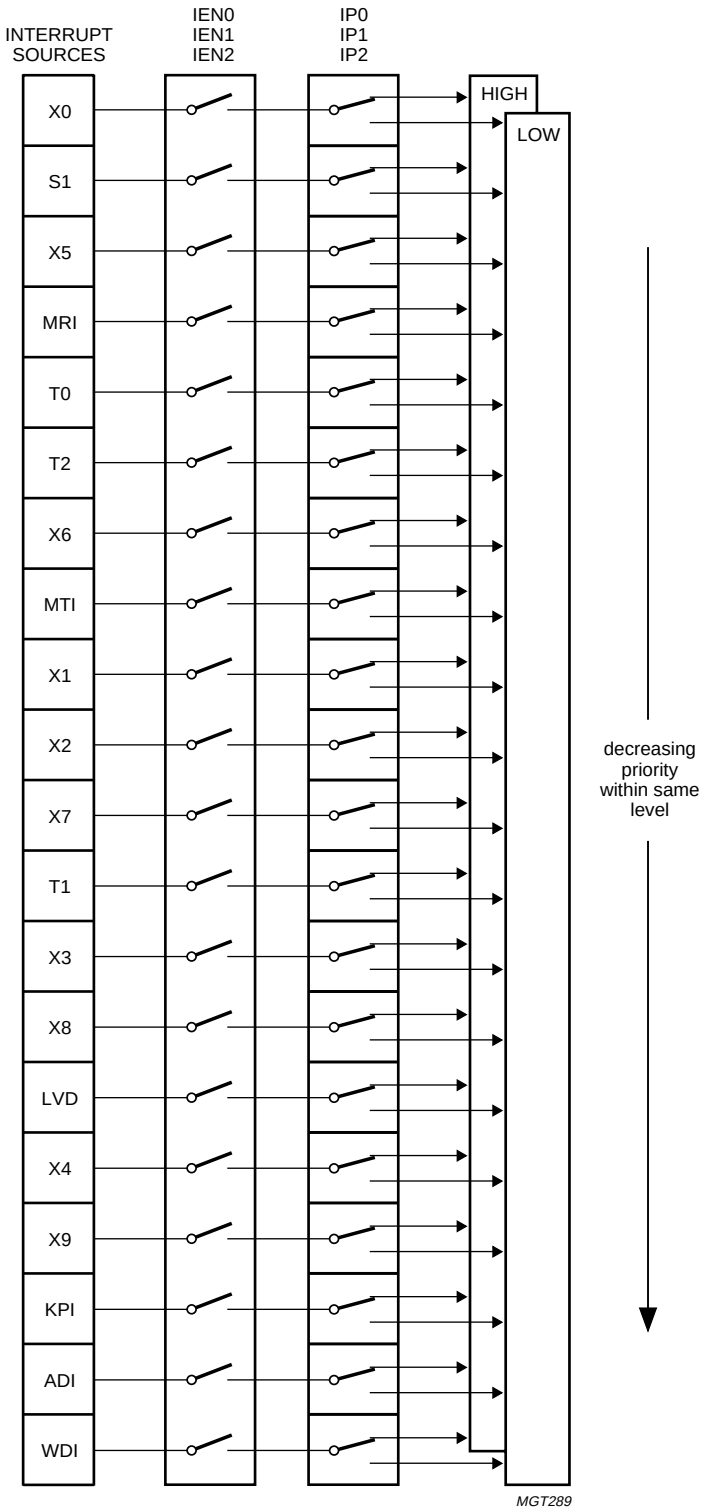


Fig.14 Interrupt assignment and priorities (Listed by scanning order).

6.3.2 INTERRUPT PROCESS

1. **Sample the interrupt lines.** The interrupt lines are latched at the beginning of each instruction cycle.
2. **Analyse the requests.** The sampled interrupt lines will be analysed with respect to the relevant Interrupt Enable Register (IENx) and Interrupt Priority Register (IPx). The process will deliver the vector of the highest interrupt request and the priority information. Depending on the interrupt level and the priority of the interrupt in progress, an interrupt request to the core is performed. The vector address will be passed to the core process.
3. **Interrupt request to core.**
   a) **Level 0:** the interrupt request to the core is performed, when at least one instruction is performed since the RETI from Level 1.
   b) **Level 1:** the interrupt request is performed, when at least one instruction is performed since the RETI from Level 21 and the request has high priority.
   c) **Level 20:** no request is performed.
   d) **Level 21:** no request is performed.
4. **Update the interrupt level.**
   a) **Level 0:** in the event of a high priority interrupt the new level will be Level 20; if it is a low priority interrupt, the new level will be Level 1.
   b) **Level 1:** in the event of a high priority interrupt, the new level will be Level 21; a low priority interrupt is not performed, the level is unchanged; on RETI the new level will be Level 0.
   c) **Level 20:** on RETI; the new level is Level 0.
   d) **Level 21:** on RETI; the new level is Level 1.
   e) **Level 1:** on RETI; the new level is Level 0.
   f) **Level 0:** the new level is Level 0.
5. **Clearing the flags.** During the forced LCALL the interrupt flag of the relevant interrupt is cleared by hardware, if applicable, otherwise by software.
6. **Idle and power-down.** When Idle (PCON.0) or Power-down (PCON.1) is set, the interrupt controller waits for the according wake-up signal. Because the interrupt controller is waiting for wake-up, all activity in the circuit will be stopped, thus no handshake can be completed. The wake-up signal for Idle is the OR of all the interrupt request bits and the reset. For power-down the wake-up signal is built only with the Port 1 external interrupt request flags (X2 to X9) and the reset (external reset or power-on reset).

6.3.3 PORT 1 INTERRUPTS

Eight Port 1 lines can be used as external interrupt inputs (X2 to X9). When enabled by IEN1 SFR, each of these interrupts may wake-up the device from Idle or Power-down. These external interrupts can each independently be programmed to positive and negative polarity and to edge and level sensitivity by setting SFR IX1 and ISE1 (see Table 34). Figure 15 shows programming of polarity and sensitivity of the Port 1 interrupts. When a valid event occurs on an enabled Port 1 interrupt, the corresponding bit in the Interrupt Request Flags Register will be set (IRQ1). The interrupt request flags must be cleared by software.

6.3.4 Interrupt Enable Register 0 (IEN0)

Table 16 Interrupt Enable Register 0 (SFR address A8H)

7	6	5	4	3	2	1	0
EA	ET2	ES1	–	ET1	EX1	ET0	EX0

Table 17 Description of IEN0 bits

Logic 0 = interrupt disabled; logic 1 = interrupt enabled.

BIT	SYMBOL	DESCRIPTION
7	EA	**General enable/disable control**. If EA = 0, no interrupt is enabled; if EA = 1, any individually enabled interrupt will be accepted.
6	ET2	enable T2 interrupt
5	ES1	enable I^2C-bus interrupt
4	–	reserved
3	ET1	enable Timer 1 interrupt (T1)
2	EX1	enable external interrupt 1
1	ET0	enable Timer 0 interrupt (T0)
0	EX0	enable external interrupt 0

6.3.5 Interrupt Enable Register 1 (IEN1)

Table 18 Interrupt Enable Register 1 (SFR address E8H)

7	6	5	4	3	2	1	0
EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2

Table 19 Description of IEN1 bits

Logic 0 = interrupt disabled; logic 1 = interrupt enabled.

BIT	SYMBOL	DESCRIPTION
7	EX9	enable external interrupt 9
6	EX8	enable external interrupt 8
5	EX7	enable external interrupt 7
4	EX6	enable external interrupt 6
3	EX5	enable external interrupt 5
2	EX4	enable external interrupt 4
1	EX3	enable external interrupt 3
0	EX2	enable external interrupt 2

6.3.6 Interrupt Enable Register 2 (IEN2)

Table 20 Interrupt Enable Register 2 (SFR address F1H)

7	6	5	4	3	2	1	0
EWDI	EADI	EKPI	–	ELVD	–	EMTI	EMRI

Table 21 Description of IEN2 bits

Logic 0 = interrupt disabled; logic 1 = interrupt enabled.

BIT	SYMBOL	DESCRIPTION
7	EWDI	enable Watchdog Timer interrupt
6	EADI	enable ADC interrupt
5	EKPI	enable key pad interrupt
4	–	reserved
3	ELVD	enable low-voltage detector interrupts
2	–	reserved
1	EMTI	enable MSK transmitter interrupt
0	EMRI	enable MSK receiver interrupts

6.3.7 Interrupt Priority Register 0 (IP0)

Table 22 Interrupt Priority Register 0 (SFR address B8H)

7	6	5	4	3	2	1	0
–	PT2	PS1	–	PT1	PX1	PT0	PX0

Table 23 Description of IP0 bits

Logic 0 = low priority; logic 1 = high priority.

BIT	SYMBOL	DESCRIPTION
7	–	reserved
6	PT2	Timer 2 interrupt priority level
5	PS1	I^2C-bus interrupt priority level
4	–	reserved
3	PT1	Timer 1 interrupt priority level
2	PX1	external interrupt 1 priority level
1	PT0	Timer 0 interrupt priority level
0	PX0	external interrupt 0 priority level

6.3.8 Interrupt Priority Register 1 (IP1)

Table 24 Interrupt Priority Register 1 (SFR address F8H)

7	6	5	4	3	2	1	0
PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2

Table 25 Description of IP1 bits
Logic 0 = low priority; logic 1 = high priority.

BIT	SYMBOL	DESCRIPTION
7	PX9	external interrupt 9 priority level
6	PX8	external interrupt 8 priority level
5	PX7	external interrupt 7 priority level
4	PX6	external interrupt 6 priority level
3	PX5	external interrupt 5 priority level
2	PX4	external interrupt 4 priority level
1	PX3	external interrupt 3 priority level
0	PX2	external interrupt 2 priority level

6.3.9 Interrupt Priority Register 2 (IP2)

Table 26 Interrupt Priority Register 2 (SFR address F9H)

7	6	5	4	3	2	1	0
PWDI	PADI	PKPI	–	PLVD	–	PMTI	PMRI

Table 27 Description of IP2 bits
Logic 0 = low priority; logic 1 = high priority.

BIT	SYMBOL	DESCRIPTION
7	PWDI	Watchdog Timer interrupt priority level
6	PADI	ADC interrupt priority level
5	PKPI	key pad interrupt priority level
4	–	reserved
3	PLVD	low-voltage detection interrupt priority level
2	–	reserved
1	PMTI	MSK transmitter interrupt priority level
0	PMRI	MSK receiver interrupt priority level

6.3.10 Interrupt Request Flag Register 1 (IRQ1)

Table 28 Interrupt Request Flag Register 1 (SFR address C0H)

7	6	5	4	3	2	1	0
IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2

Table 29 Description of IRQ1 bits

BIT	SYMBOL	DESCRIPTION
7	IQ9	external interrupt 9 request flag
6	IQ8	external interrupt 8 request flag
5	IQ7	external interrupt 7 request flag
4	IQ6	external interrupt 6 request flag
3	IQ5	external interrupt 5 request flag
2	IQ4	external interrupt 4 request flag
1	IQ3	external interrupt 3 request flag
0	IQ2	external interrupt 2 request flag

6.3.11 Interrupt polarity and sensitivity registers

6.3.11.1 Interrupt Polarity Register 1 (IX1)

Writing either a logic 1 or logic 0 to any Interrupt Polarity Register bit sets the polarity of the corresponding external interrupt. If the interrupt sensitivity bit (ISE1 register, Section 6.3.11.2) is set to ‘level’ sensitive then a logic 1 corresponds to active HIGH level and logic 0 to active LOW level. If the ISE1 register is set to ‘edge’ sensitive then a logic 1 corresponds to a rising edge and a logic 0 to a falling edge. See also Table 34 and Fig.15.

Table 30 Interrupt Polarity Register 1 (SFR address E9H)

7	6	5	4	3	2	1	0
IX9	IX8	IX7	IX6	IX5	IX4	IX3	IX2

Table 31 Description of IX1 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	IX9 to IX2	external interrupt 9 to 2 polarity level

6.3.11.2 Interrupt Sensitivity Register 1 (ISE1)

Writing either a logic 1 or logic 0 to an Interrupt Sensitivity Register bit sets the type of the corresponding external interrupt to edge sensitive (logic 1) or level sensitive (logic 0) respectively.

Table 32 Interrupt Sensitivity Register 1 (SFR address E1H)

7	6	5	4	3	2	1	0
ISE9	ISE8	ISE7	ISE6	ISE5	ISE4	ISE3	ISE2

Table 33 Description of ISE1 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	ISE9 to ISE2	external interrupt 9 to 2 sensitivity

6.3.11.3 Interrupt polarity and sensitivity options

Table 34 Interrupt polarity and sensitivity options
'n' denotes the bit position in the SFRs IX1 and ISE1.

IX1.n	ISE1.n	DESCRIPTION
0	0	LOW-level sensitive
1	0	HIGH-level sensitive
0	1	falling edge sensitive
1	1	rising edge sensitive

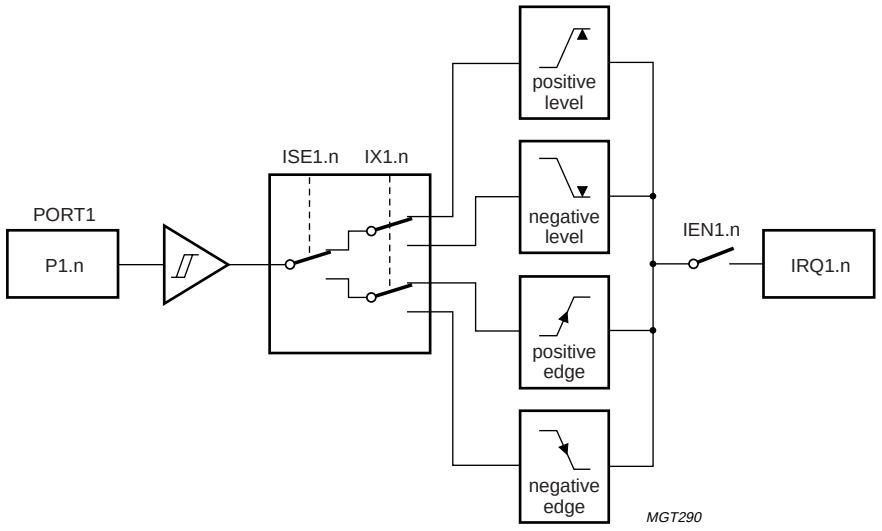


Fig.15 Polarity and sensitivity of Port 1 interrupts.

6.4 Key Pad Interrupt (KPI)

The key pad interrupt is a logical ‘OR’ of 8 inputs from Port 4 (KP0 to KP7). The key pad interrupt generates only 1 interrupt source to the interrupt controller (KPI).

A key pad interrupt is implemented to read out a key board with 8 interrupt lines. An event on one of the inputs KP0 to KP7 generates a KPI interrupt. Each of the key pad interrupt inputs can individually be enabled by setting the corresponding bit in the IEN3 register. The key pad interrupt can be globally disabled by writing to IEN2.5 in the interrupt controller. Each pin can be initialized to both polarities depending on the settings in the Interrupt Polarity Register 3 (IX3). The key pad interrupt can be made edge or level sensitive by writing to the ISE3 register. See Table 44.

The key pad interrupts are set by hardware and must be cleared by software. Table 35 gives an overview of the key pad interrupt related registers. Sections 6.4.3.1 to 6.4.3.2 give detailed description of the SFRs.

Table 35 Key Pad Interrupt related SFRs

ADDR	SFR	DESCRIPTION
F4H	IX3	Key Pad Interrupt Polarity Register 3
F5H	IRQ3	Key Pad Interrupt Request Flag Register 3
F6H	IEN3	Key Pad Interrupt Enable Register 3
F7H	ISE3	Key Pad Interrupt Sensitivity Register 3

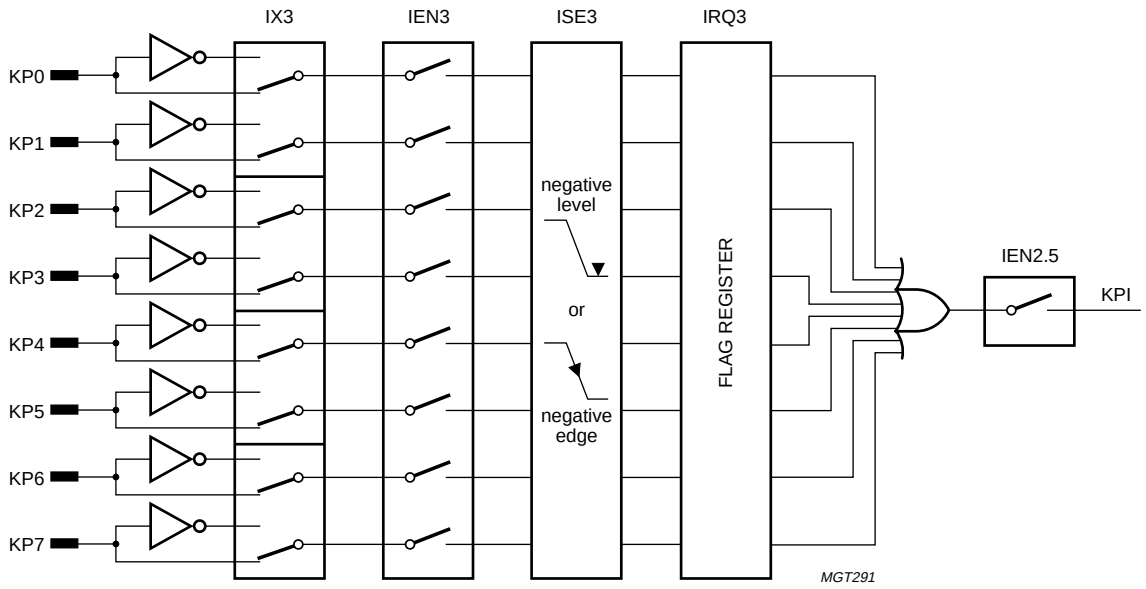


Fig.16 Block diagram of the key pad interrupt.

6.4.1 KEY PAD INTERRUPT REQUEST FLAG REGISTER 3 (IRQ3)

Table 36 Key Pad Interrupt Request Flag Register 3 (SFR address F5H)

7	6	5	4	3	2	1	0
IQKP7	IQKP6	IQKP5	IQKP4	IQKP3	IQKP2	IQKP1	IQKP0

Table 37 Description of IRQ3 bits

BIT	SYMBOL	DESCRIPTION
7	IQKP7	key pad interrupt 7 request flag
6	IQKP6	key pad interrupt 6 request flag
5	IQKP5	key pad interrupt 5 request flag
4	IQKP4	key pad interrupt 4 request flag
3	IQKP3	key pad interrupt 3 request flag
2	IQKP2	key pad interrupt 2 request flag
1	IQKP1	key pad interrupt 1 request flag
0	IQKP0	key pad interrupt 0 request flag

6.4.2 KEY PAD INTERRUPT ENABLE REGISTER 3 (IEN3)

Table 38 Key Pad Interrupt Enable Register 3 (SFR address F6H)

7	6	5	4	3	2	1	0
EKP7	EKP6	EKP5	EKP4	EKP3	EKP2	EKP1	EKP0

Table 39 Description of IEN3 bits

Logic 0 = interrupt disabled; logic 1 = interrupt enabled.

BIT	SYMBOL	DESCRIPTION
7	EKP7	enable key pad interrupt KP7
6	EKP6	enable key pad interrupt KP6
5	EKP5	enable key pad interrupt KP5
4	EKP4	enable key pad interrupt KP4
3	EKP3	enable key pad interrupt KP3
2	EKP2	enable key pad interrupt KP2
1	EKP1	enable key pad interrupt KP1
0	EKP0	enable key pad interrupt KP0

6.4.3 KEY PAD INTERRUPT POLARITY AND SENSITIVITY REGISTERS

6.4.3.1 Key Pad Interrupt Polarity Register 3 (IX3)

Writing either a logic 1 or logic 0 to any Key pad Interrupt Polarity Register bit sets the polarity level of the corresponding external key pad interrupt to an active HIGH (rising edge) or active LOW (falling edge) respectively. See also Table 44 and Fig.16

Table 40 Key Pad Interrupt Polarity Register 3 (SFR address F4H)

7	6	5	4	3	2	1	0
IXKP7	IXKP6	IXKP5	IXKP4	IXKP3	IXKP2	IXKP1	IXKP0

Table 41 Description of IX3 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	IXKP7 to IXKP0	key pad interrupt 7 to 0 polarity level

6.4.3.2 Key Pad Interrupt Sensitivity Register 3 (ISE3)

Writing either a logic 1 or logic 0 to any Key Pad Interrupt Sensitivity Register bit sets the type of the corresponding external interrupt to edge sensitive or level sensitive respectively.

Table 42 Key Pad Interrupt Sensitivity Register (SFR address F7H)

7	6	5	4	3	2	1	0
ISKP7	ISKP6	ISKP5	ISKP4	ISKP3	ISKP2	ISKP1	ISKP0

Table 43 Description of ISE3 bits

BIT	SYMBOL	DESCRIPTION
7 to 0	ISKP7 to ISKP0	key pad interrupt 7 to 0 sensitivity

6.4.3.3 Key pad interrupt level and sensitivity selection

Table 44 Key pad interrupt options
'n' denotes the bit position in the SFRs IX3 and ISE3.

IX3.n	ISE3.n	DESCRIPTION
0	0	LOW-level sensitive
1	0	HIGH-level sensitive
0	1	falling edge sensitive
1	1	rising edge sensitive

6.5 Port control logic

Four 8-bit I/O ports are implemented in the device. Some of these general purpose I/Os are multiplexed with alternative functions. Port 0 is the only port with no multiplexed alternative functions. Port 3 and a part of Port 1 are multiplexed with analog functions. Every port bit can be independently configured in 4 different modes.

6.5.1 PORT FUNCTIONALITY

Port 0 8-bit bidirectional I/O port with no alternative functions. Every port pin can be used as open-drain, standard port, high-impedance input or push-pull output. Port 0 is used during emulation mode.

Port 1 8-bit bidirectional I/O port with alternative functions. Every port, except P1.6 and P1.7 can be used as open-drain, standard port, high-impedance input or push-pull output.

- P1.0 to P1.7 provides the inputs for the external interrupts INT2 to INT9; the interrupts are enabled by selecting the proper bit in the interrupts enable register
- P1.0 to P1.2 provides the analog inputs ADC5 to ADC3[(1)]; the port must be set in the high-impedance configuration to enable the analog input function
- P1.1 and P1.2 provide the Timer 2 external trigger input (T2EX) and the Timer 2 external count input (T2)
- P1.4 provides the clock output CLKOUT (f_{psc} or f_{per})
- P1.5 provide the Timer 2 clock output of the clock-output mode (T2OUT); to enable output the data SFR must contain logic 1s
- P1.6 and P1.7 provide the I^2C-bus clock and data I/O, SCL and SDA. P1.6 and P1.7 can only be configured as open-drain output or high-impedance input; there is no clamp diode to V_{DD}. I^2C-bus signals are connected to the port if bit ENS1 (S1CON SFR) is set to logic 1.

Port 2 Not used.

Port 3 8-bit bidirectional I/O port with alternative functions. Every port can be used as open-drain, standard port, high-impedance input or push-pull output.

- P3.0 to P3.2 provide the MSK output signals MOUT0, MOUT1 and MOUT2
- P3.3 and P3.4 provide the DAC0 and DAC1[(1)] analog output; the port must be set in the high-impedance configuration to enable the analog output function
- P3.4 also provides the Timer 0 external clock input
- P3.5 to P3.7 provide the ADC0 till ADC2[(1)] analog input; the port must be set in the high-impedance configuration to enable the analog input function
- P3.5 also provides the Timer 1 external clock input.

Port 4 8-bit bidirectional I/O port with alternative functions. Every port pin can be used as open-drain, standard port, high-impedance input or push-pull output.

- P4.0 to P4.7 provides the key pad interrupt functionality KP0 to KP7; the keypad interrupt lines are enabled by setting a value in the Key Pad Interrupt Enable Register 3 (IEN3).

(1) The ports which have an analog option (e.g.: ADCn and DACn) multiplexed do not have a hysteresis input.

6.5.2 PORT I/O CONFIGURATION

Each port bit consist of a data latch, two configuration latches, an output driver and an input buffer. The I/O port configurations are determined by the settings in the port configuration SFRs, PnCFGA and PnCFGB, where 'n' indicates the specific port number (0, 1, 3 and 4). The combination of 2 bits in each of the 2 configuration SFRs relates to the output setting for the corresponding port pin, allowing any combination of the 4 I/O modes to be mixed on those port pins. The port I/O configuration types are shown in Fig.17 and described in Sections 6.5.2.1 to 6.5.2.4.

6.5.2.1 Open-drain

Quasi-bidirectional I/O with n-channel open-drain output. Use as an output requires the connection of an external pull-up resistor; all pins have ESD protection diodes against V_{DD} and V_{SS}, except for the I^2C-bus pins P1.6 and P1.7, which have no ESD protection to V_{DD}.

6.5.2.2 Standard port

Quasi-bidirectional I/O with pull-up; the strong pull-up 'p1' is turned on for three clock (f_{osc}) edges after a LOW-to-HIGH transition in the port latch; after these three clock edges the port is only weakly driven through 'p2' and 'very weakly' driven through 'p3' (see Fig.17b).

6.5.2.3 High-impedance input

This mode turns off all output drivers on a port. The pin will not source or sink current and may be used as an input-only pin. (see Fig.17c). In order not to increase the current consumption the high-impedance input should not float.

6.5.2.4 Push-pull

Output with drive capability in both polarities; under this mode, pins can only be used as outputs (see Fig.17d).

Table 45 Port I/O configuration types
'n' indicates the specific port number (0, 1, 3 and 4).

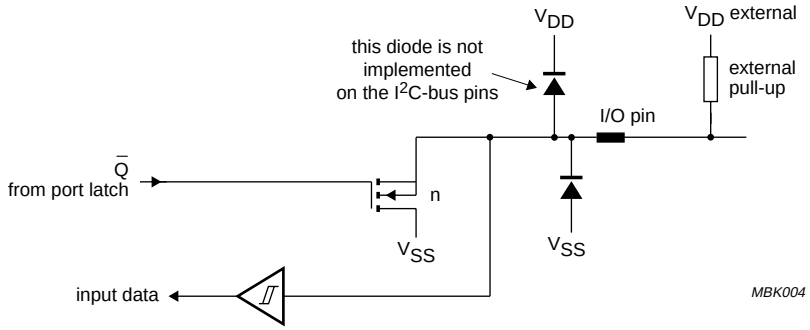
TYPE	PnCFGA	PnCFGB	NORMAL PORTS	I^2C-BUS PORTS
Open-drain	0	0	open-drain	open-drain
Standard port	1	0	quasi-bidirectional	open-drain
High-impedance input	0	1	high-impedance input	high-impedance input
Push-pull	1	1	push-pull	open-drain

Table 46 Reset state of port related SFRs

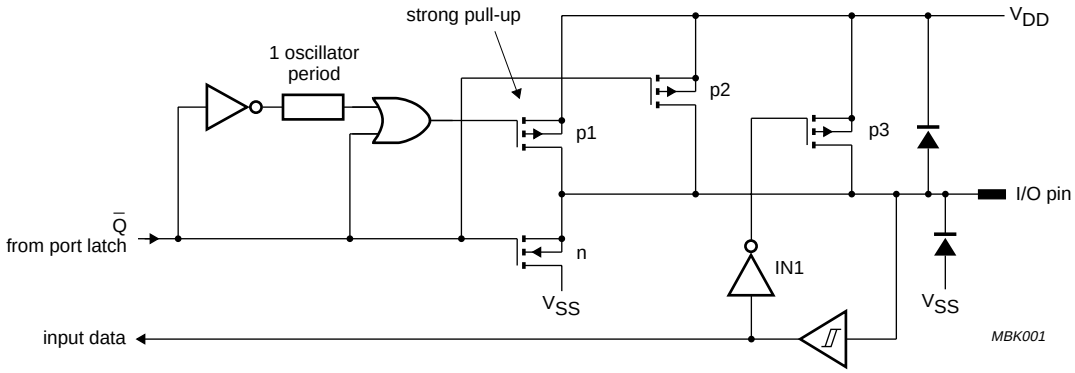
SFR	DESCRIPTION	SFR ADDRESS (HEX)	STATE AFTER RESET(1)
P0	Port 0 output data	80	1110 1111
P0CFGA	Port 0 Configuration A	8E	1111 0011
P0CFGB	Port 0 Configuration B	8F	0000 0000
P1	Port 1 output data	90	1111 1111
P1CFGA	Port 1 Configuration A	9E	1111 1111
P1CFGB	Port 1 Configuration B	9F	0000 0000
P3	Port 3 output data	B0	1111 1111
P3CFGA	Port 3 Configuration A	BE	1111 1111
P3CFGB	Port 3 Configuration B	BF	0000 0000
P4	Port 4 output data	C1	1111 1110
P4CFGA	Port 4 Configuration A	CE	1111 1111
P4CFGB	Port 4 Configuration B	CF	0000 0000

Note

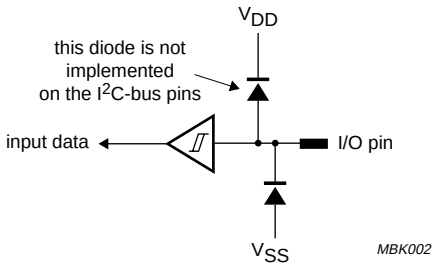
1. This means all ports, except P0.2, P0.3, P0.4, P1.6, P1.7 and P4.0 are initialized in standard port configuration driving a weak logic 1. Port 0.2 and P0.3 are initialised as open-drain outputs, floating. P0.4 is initialised as bidirectional, driving a strong logic 0. I^2C-bus I/Os P1.6 and P1.7 are initialised in open-drain configuration, floating. The configuration registers (P1CFGA.7 to 6 and P1CGB.7 to 6) are however configured as standard port configuration but the connections to the port PMOS transistors are not present. P4.0 is initialised as bidirectional, driving a strong logic 0.



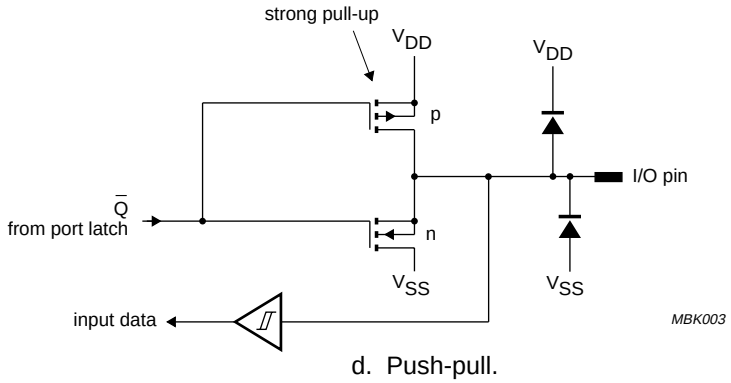
a. Open-drain.



b. Standard/quasi-bidirectional.



c. High-impedance input.



d. Push-pull.

Fig.17 Port configuration options.

6.6 Timer 0 and Timer 1 event counters

Timer 0 and Timer 1 can perform the following functions:

- Measure time intervals and pulse durations
- Count events
- Measure CPU speed
- Generate interrupt requests.

Timer 0 and Timer 1 can be programmed independently to operate in four modes:

Mode 0 8-bit timer or 8-bit counter each with divide-by-32 prescaler.

Mode 1 16-bit time interval or event counter.

Mode 2 8-bit time interval or event counter with automatic reload upon overflow.

Mode 3 Timer 1 stopped and Timer 0 operates as two separate counters.

A block diagram of Timer 0 and Timer 1 with possible clock sources is shown in Fig.18.

Table 47 Timer/Counter 0 and 1 related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
TCON	Timer/counter 0 and 1 Control Register	88H	0000 0000
TMOD	Timer/counter 0 and 1 Mode Control Register	89H	0000 0000
SYSCON	System Control Register	B4H	0000 0000

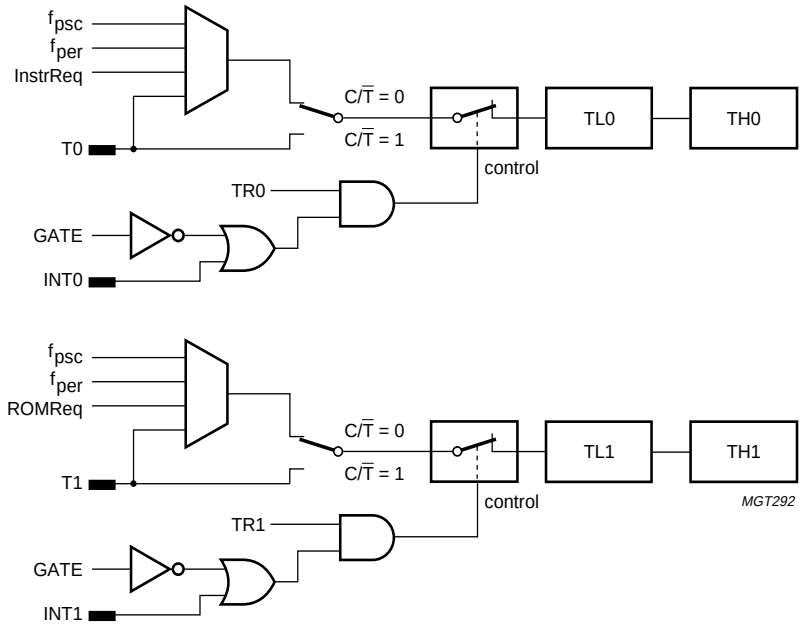


Fig.18 Timer/counter 0 and 1; clock sources and control logic.

6.6.1 Clock source signals of Timer 0 and Timer 1

In all four modes Timer 0 and Timer 1 can be configured to increment from different internal and external clock sources. The TMOD and SYSCON registers must be written to determine the source of the clock signal. After reset the clock source for both timers is connected to the internal clock signal from PSC1 (f_{psc}). The second of four possible clock sources is connected to the other internal clock signal coming from PSC2 (f_{per}).

The clock input on both timers has a multiplexer to choose from 4 different clock sources. If the multiplexers are switched to another input by setting user controllable bits in the SYSCON SFR (bits 7 to 4), the timers can also increment on the other on-chip clock signal coming from PSC2 (f_{per}).

In counter mode the timers are incrementing on transitions on the T0 and T1 input pins. First way to enter this mode is by setting control bits C/$\overline{T}$ (TMOD.6 and 2). Second way is to configure SYSCON to switch the input multiplexer to the clock input signal T1 or T0 while C/$\overline{T}$ is logic 0. The latter is also functional even when there is no system clock available. This means when a clock source is supplied on a port pin the Timer 1 or 0 can count and generate interrupts even when the chip is in Power-down mode. Maximum input signal frequency and duty cycle for the timer in counter mode is given in the electrical specification (see Chapter 12).

The last multiplexer input to the Timer 1 and Timer 0 is an auxiliary mode which can be used to obtain the operation speed from the handshake CPU. If this mode is activated for the Timer 1 input source, the timer increments on every ROM request. This means the timer increments by three for a three byte instruction and by two for a two byte instruction etc. If the auxiliary mode is activated for Timer 0 the timer increments on every instruction executed by the CPU. This means the timer register holds the number of instructions executed in a certain time frame. This can be used to obtain the number of Mips at which the processor is running. The SYSCON register is described in Section 6.6.5.

6.6.2 Operating modes of Timer 0 and Timer 1

The ‘Timer’ or ‘Counter’ function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1 and M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 configures Timer 0 while Timer 1 is disabled.

The four operating modes are:

Mode 0 Putting either Timer 0 or Timer 1 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 the Mode 0 operation as it applies to Timer 1. In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all logic 1s to all logic 0s, it sets the timer interrupt flag TF1. The counted input is enabled to the timer when TR1 = 1 and either GATE = 0 or INT1 = 1. (Setting GATE = 1 allows the timer to be controlled by external input INT1, to facilitate pulse width measurements). TR1 is a control bit in the Special Function Register TCON (see Table 50). GATE is in TMOD. The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers. Mode 0 operation is the same for the Timer 0 as for 1. Substitute TR0, TF0, and INT0 for the corresponding Timer 1 signals in Fig.19. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Mode 1 Is the same as Mode 0, except that the timer register is being run with all 16 bits.

Mode 2 Configures the timer register as an 8-bit counter (TL1) with automatic reload, as shown in Fig.20. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

Mode 3 Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Fig.21. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3 or in any application not requiring an interrupt.

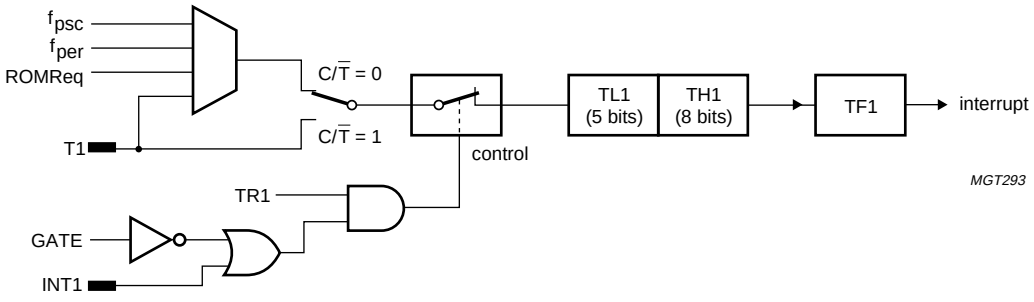


Fig.19 Timer/Counter 0 and Timer/Counter 1; Mode 0: 13-bit counter.

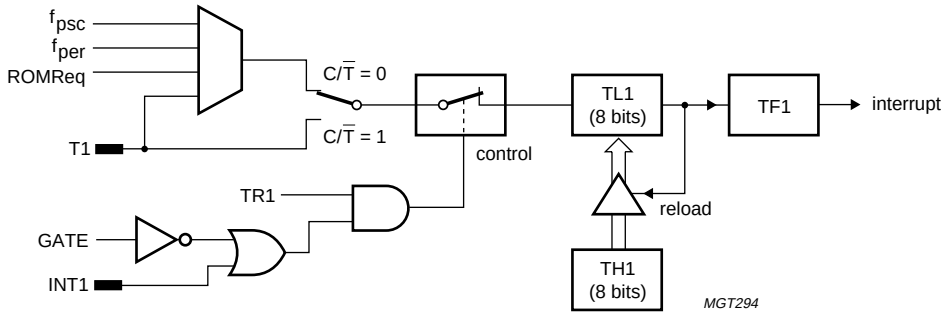


Fig.20 Timer/Counter 0 and Timer/Counter 1; Mode 2: 8-bit auto-reload.

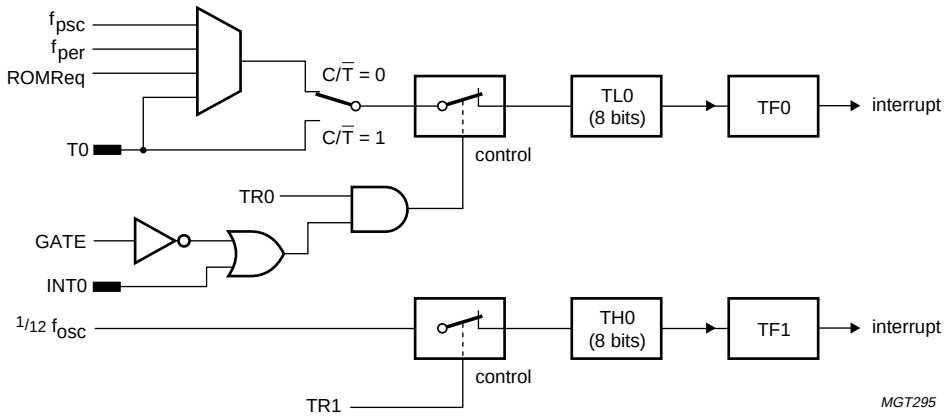


Fig.21 Timer/Counter 0 and Timer/Counter 1; Mode 3: two 8-bit counters.

6.6.3 TIMER/COUNTER 0 AND 1 CONTROL REGISTER (TCON)

Table 48 Timer/Counter 0 and 1 Control Register (SFR address 88H)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table 49 Description of TCON bits

BIT	SYMBOL	DESCRIPTION
7	TF1	**Timer 1 overflow flag**. Set by hardware on timer/counter overflow; cleared by hardware when processor vectors to interrupt routine, or clearing the bit in software.
6	TR1	**Timer 1 run control bit**. Set/cleared by software to turn timer/counter on/off; note 1.
5	TF0	**Timer 0 overflow flag**. Set by hardware on timer/counter overflow. cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
4	TR0	**Timer 0 run control bit**. Set/cleared by software to turn timer/counter on/off; note 1.
3	IE1	**Interrupt 1 edge flag**. Set by hardware when external interrupt edge detected; cleared when interrupt processed.
2	IT1	**Interrupt 1 type control bit**. Set/cleared by software to specify falling edge/low level triggered external interrupts.
1	IE0	**Interrupt 0 edge flag**. Set by hardware when external interrupt edge detected; cleared when interrupt processed.
0	IT0	**Interrupt 0 type control bit**. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Note

1. If the Timer 0 or Timer 1 is not enabled (TR0 or TR1), the clock to Timer 0/Timer 1 is switched off for power saving.

6.6.4 TIMER/COUNTER 0 AND 1 MODE CONTROL REGISTER (TMOD)

Table 50 Timer/Counter 0 and 1 Mode Control Register (SFR address 89H)

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

Table 51 Description of TMOD bits

BIT	SYMBOL	DESCRIPTION
7	GATE	**Gating control**. When set Timer/Counter 1 is enabled only while INT1 pin is HIGH and TR1 control pin is set; when cleared Timer 1 is enabled whenever TR1 control bit is set
6	C/$\overline{T}$	**Timer or counter selector**. Cleared for timer operation (counts on f_{PSC}); set for counter operation (input from T1 input pin)
5	M1	**Timer 1 mode select**. See Table 52
4	M0	
3	GATE	**Gating control**. When set Timer/Counter 0 is enabled only while INT0 pin is HIGH and TR0 control pin is set; when cleared Timer 0 is enabled whenever TR0 control bit is set.
2	C/$\overline{T}$	**Timer or counter selector**. Cleared for timer operation (counts on f_{PSC}); set for counter operation (input from T0 input pin).
1	M1	**Timer 0 mode select**. See Table 52.
0	M0	

Table 52 Timer 0 and Timer 1 mode select
n = 0 or 1.

M1	M0	DESCRIPTION
0	0	**8048-type 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 to be reloaded into TLn each time it overflows.
1	1	**Timer 0**. TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits; TH0 is an 8-bit timer only controlled by Timer 1 control bits.
		Timer 1. Timer/Counter 1 stopped.

6.6.5 System Control Register (SYSCON)

Table 53 System Control Register (SFR address B4H; reset value = 0000 0000)

7	6	5	4	3	2	1	0
T1SRC1	T1SRC0	T0SRC1	T0SRC0	–	–	SELECT	XTM

Table 54 Description of SYSCON bits

BIT	SYMBOL	DESCRIPTION
7	T1SRC1	Timer 1 clock source select bit 1 and 0; see Table 55
6	T1SRC0	
5	T0SRC1	Timer 0 clock source select bit 1 and 0; see Table 56
4	T0SRC0	
3	–	do not use
2	–	
1	SELECT	comparator select bit; see Section 6.1
0	XTM	oscillator disable bit; see Section 6.1

Table 55 Timer 1 input source select modes

T1SRC1	T1SRC0	DESCRIPTION
0	0	f_{psc} is the Timer 1 clock input
0	1	T1 is the Timer 1 clock input
1	0	the ROMreq signal is the Timer 1 clock input
1	1	f_{per} is the Timer 1 clock input

Table 56 Timer 0 input source select modes

T0SRC1	T0SRC0	DESCRIPTION
0	0	f_{psc} is the Timer 0 clock input
0	1	T0 is the Timer 0 clock input
1	0	the instruction request signal is the Timer 0 clock input
1	1	f_{per} is the Timer 0 clock input

6.7 Timer 2

Timer 2 is a 16-bit timer/counter which can operate as either an event timer or an event counter. Timer 2 has three operating modes: capture, auto-reload up/down counting and clock output mode. These modes are selected by bits in the T2CON SFR.

6.7.1 Timer 2 Special Function Registers

Timer 2 has six Special Function Registers (SFRs) that can be read and written by the CPU. These registers are: T2CON, T2MOD, T2H, T2L, T2RCH and T2RCL. Timer 2 register values can change by hardware or by software. When an update by hardware and software occurs in one of the registers T2H, T2L, T2RCH or T2RCL, the update by software has precedence over the update by hardware.

Table 57 Timer 2 related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
T2CON	Timer 2 Control Register	C8H	00XX 0000
T2MOD	Timer 2 Mode Register	C9H	XXXX X000
T2L	Timer 2 Low byte Count Register	CCH	0000 0000
T2H	Timer 2 High byte Count Register	CDH	0000 0000
T2RCL	Timer 2 Low byte Capture/Reload Register	CAH	0000 0000
T2RCH	Timer 2 High byte Capture/Reload Register	CBH	0000 0000

6.7.1.1 Timer 2 Control Register (T2CON)

Table 58 Timer 2 Control Register (SFR address C8H)

7	6	5	4	3	2	1	0
T2F	EXF2	–	–	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Table 59 Description of T2CON bits

BIT	SYMBOL	DESCRIPTION
7	T2F	**Timer 2 overflow flag**. Set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when clock out mode is selected.
6	EXF2	**Timer 2 external flag**. Set on a negative transition on T2EX and bit EXEN2 = 1. In Auto-reload mode it is toggled on an under- or overflow; this bit must be cleared by software.
5	–	Reserved; must be kept to logic 0.
4	–	
3	EXEN2	**Timer 2 external enable flag**. Set by software only; when set, allows a capture or reload to occur, together with an interrupt, as a result of a negative transition on input T2EX if in Capture mode or Auto-reload mode with DCEN reset. If in Auto-reload mode and DCEN is set, the EXEN2 bit has no influence. In the other modes EXF2 is set and an interrupt is generated on a HIGH-to-LOW transition on the T2EX pin. When EXEN2 is reset, Timer 2 ignores events on pin T2EX in all modes.

BIT	SYMBOL	DESCRIPTION
2	TR2	**START/STOP control for Timer 2**. Set by software only; when set, the timer is started; when reset the timer is stopped; note 1.
1	C/$\overline{T2}$	**Timer/counter select for Timer 2**. Set by software only; when set the counter function is selected, when reset the timer function is selected.
0	CP/$\overline{RL2}$	**Capture/Reload flag**. Set by software only; selection of mode capture or reload; when set the capture function is selected, when reset the reload function is selected.

Note

1. If Timer 2 is not enabled (TR2 = 0), the clock to Timer 2 is switched off for power saving.

6.7.1.2 Timer 2 Mode Register (T2MOD)

Table 60 Timer 2 Mode Register (SFR address C9H)

7	6	5	4	3	2	1	0
–	–	–	–	–	T2RD	C/T2OE	CP/DCEN

Table 61 Description of T2MOD bits

BIT	SYMBOL	DESCRIPTION
7 to 3	–	Reserved; must be kept to logic 0.
2	T2RD	**Timer 2 read flag**. Set/reset by hardware only. This bit is set by hardware if a T2L read operation is followed by an increment of T2H before a T2H read operation. This bit is reset on the trailing edge of the next T2L read. This bit is used to indicate that the16-bit Timer 2 register is not read properly since the T2H part was incremented by hardware before it was read.
1	C/T2OE	**Timer 2 output enable bit**. Set by software only. When set and T2CON.TF2 is reset and T2CON.EXF2 is reset, output T2 outputs a clock signal. When this condition is not met, output T2 outputs a logic 1. The clock output is half the overflow frequency of Timer 2.
0	CP/DCEN	**Down count enable flag**. Set by software only. When this bit is set and input T2EX is set Timer 2 can be configured (in Auto-reload mode) as an up counter. When this bit is reset or input T2EX is reset, Timer 2 can be configured (in Auto-reload mode) as a down counter.

6.7.1.3 T2H and T2L Registers

These registers are normal registers in the SFR space. They are the actual timer/counter registers. On the fly reading can give a wrong value since T2H can be changed after T2L is read and before T2H is read. This situation is indicated by flag T2RD in T2MOD SFR. In all cases the two 8 bit registers operate as one 16 bit timer/counter register.

6.7.1.4 T2RCH and T2RCL Registers

These registers are normal registers in the SFR space. They are the capture and reload registers depending on the chosen operation mode. In the Capture mode the T2RCH/T2RCL registers are loaded with the value of the T2H/T2L registers. In the reload mode the T2H/T2L registers are loaded with the value of the T2RCH/T2RCL registers.

6.7.2 Timer 2 modes in general

Timer 2 can operate in three different modes:

- Capture mode
- Auto-reload mode
- Clock output mode.

In these three modes the timer/counter operates on events detected on inputs T2 and T2EX.

Table 62 shows the list of T2CON and T2MOD register bits which set the Timer 2 mode of operation.

Sections 6.7.3 to 6.7.5 describe the Timer 2 modes.

Table 62 Timer 2 modes

CP/$\overline{RL2}$	C/T2OE	C/$\overline{T2}$	OPERATING MODE
0	0	X	16-bit auto-reload
1	0	X	16-bit capture
0	1	0	clock output

6.7.3 Capture mode

In the Capture mode, registers T2RCH/T2RCL are used to capture the T2H/T2L register data.

There are two options selected by the T2CON.EXEN2 bit. This bit enables or disables the events of the external trigger input T2EX.

- T2CON.C/$\overline{T2}$ = 1: Timer 2 is a 16-bit counter. The counter increments at each LOW-to-HIGH transition on input T2 at a maximum rate of one each 12 f_{psc} cycles.
- T2CON.C/$\overline{T2}$ = 0: Timer 2 is a 16 bit timer. The timer increments each 6 f_{PSC} cycles.
- T2CON.EXEN2 = 1: The external trigger input T2EX is enabled. Timer 2 is a 16-bit timer or counter.
  - If T2MOD.DCEN = 0, a HIGH-to-LOW transition at input T2EX causes the current Timer 2 value (T2H/T2L data) to be captured into T2RCH/T2RCL, and bit T2CON.EXF2 becomes set.
  - If T2MOD.DCEN = 1, bit T2CON.EXEN2 has no influence. Overflowing of Timer 2 sets bit T2CON.TF2.
- T2CON.EXEN2 = 0: The external trigger input T2EX is disabled. Timer 2 is a 16-bit timer or counter. The T2EX input is ignored. Overflowing of Timer 2 sets bit T2CON.TF2.

The Capture mode is shown in Fig.22.

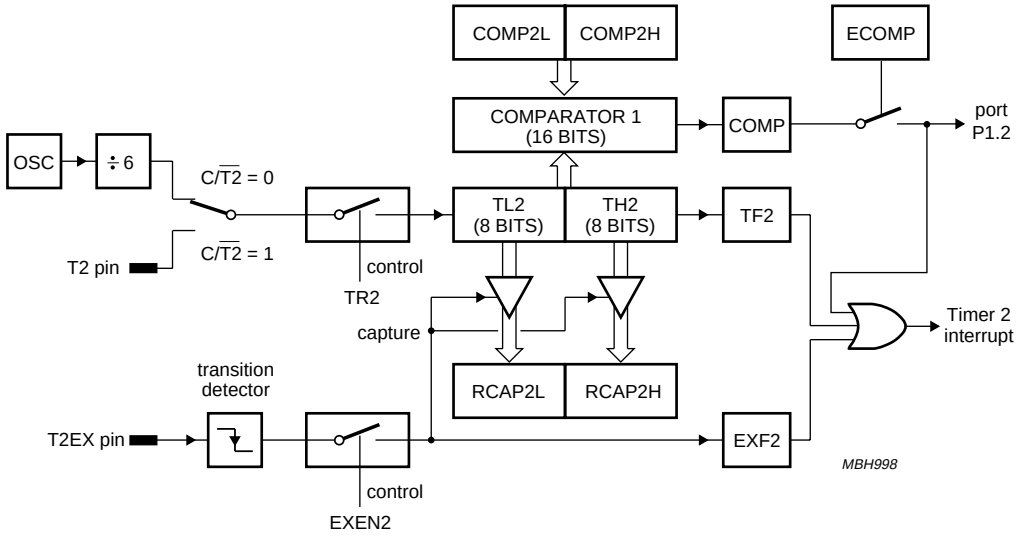


Fig.22 Timer 2 in Capture mode.

6.7.4 AUTO-RELOAD MODE

In the Auto-reload mode, Timer 2 can be configured as a timer or a counter (T2CON.C/$\overline{T2}$ bit) and then programmed to count up or down. The counting direction is determined by bit T2MOD.DCEN (down count enable). When reset is applied, then T2MOD.DCEN is reset which defaults to counting up. If T2MOD.DCEN is set, Timer 2 can count up when T2EX = 1 and count down when T2EX = 0.

- T2CON.C/$\overline{T2}$ = 1: Timer 2 is a 16-bit counter. The counter increments/decrements at each LOW-to-HIGH transition on input T2 at a maximum rate of one each 12 f_{psc} cycles.
- T2CON.C/$\overline{T2}$ = 0: Timer 2 is a 16-bit timer. The timer increments/decrements each 6 f_{psc} cycles.

6.7.4.1 T2MOD.DCEN = 0: counting up

In the Auto-reload mode and counting up, registers T2RCH/T2RCL are used to hold a reload value for T2H/T2L.

By setting bit T2CON.EXEN2 the external trigger input T2EX is enabled. When resetting bit T2CON.EXEN2, the external trigger input T2EX is disabled.

- T2CON.EXEN2 = 1: The external trigger input T2EX is enabled. Timer 2 is a 16-bit timer or counter.
  A HIGH-to-LOW transition at input T2EX causes the value in T2RCH/T2RCL to be reloaded in the Timer 2 T2H/T2L registers, and bit T2CON.EXF2 becomes set. Also overflowing of Timer 2 causes the value in T2RCH/T2RCL to be reloaded in the T2H/T2L registers and sets bit T2CON.TF2.
- T2CON.EXEN2 = 0: The external trigger input T2EX is disabled. Timer 2 is a 16-bit timer or counter. The T2EX input is ignored. Overflowing of Timer 2 causes the value in T2RCH/T2RCL to be reloaded in the T2H/T2L registers and sets bit T2CON.TF2.

Timer 2 interrupt will be set if EXF2 is set or TF2 is set.

The Auto-reload mode (DCEN = 0) is shown in Fig.23.

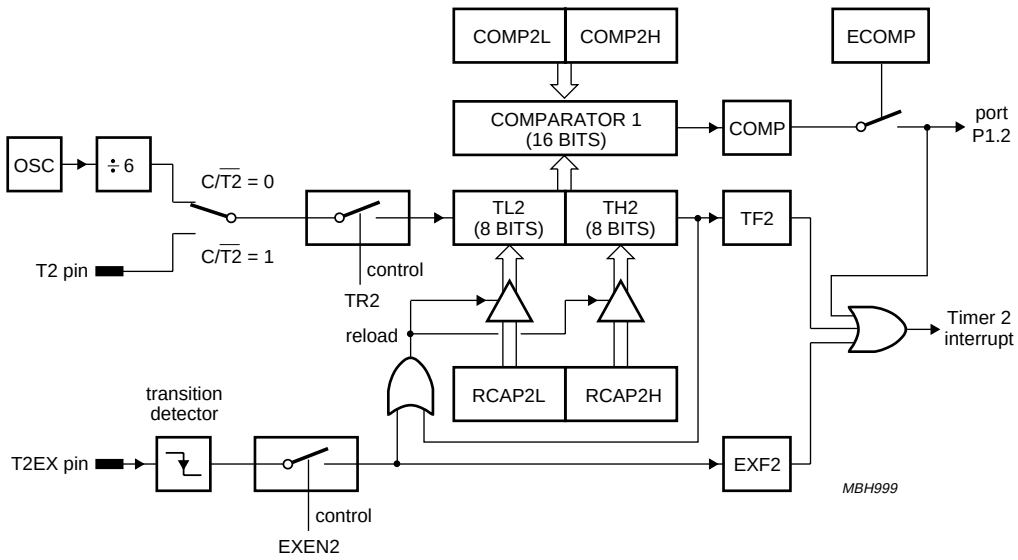


Fig.23 Timer 2 in Auto-reload mode (DCEN = 0).

6.7.4.2 T2MOD.DCEN = 1: T2EX = 1: counting up

The HIGH value of the external trigger input T2EX sets Timer 2 to a count-up mode. In the Auto-reload mode and counting up, registers T2RCH/T2RCL are used to hold a reload value for T2H/T2L. Overflowing of Timer 2 causes the value in T2RCH/T2RCL to be reloaded in the T2H/T2L registers, sets bit T2CON.TF2 and toggles bit T2CON.EXF2 (T2CON.EXF2 can be used as 17th bit if desired). Timer 2 interrupt will be set if TF2 is set.

6.7.4.3 T2MOD.DCEN = 1: T2EX = 0: counting down

The LOW value of the external trigger input T2EX sets Timer 2 to a count down mode.

In the Auto-reload mode and counting down, registers T2RCH/T2RCL are used to hold a value for detecting an underflow of T2H/T2L. Underflow occurs if the contents of T2H/T2L matches the contents of T2RCH/T2RCL. Upon underflow, bit TF2 will be set and registers T2H/T2L will be loaded with FFFFH, bit T2CON.TF2 is set and bit T2CON.EXF2 toggles (T2CON.EXF2 can be used as 17th bit if desired).

Note that a Timer 2 roll over from 0000H to FFFFH is not considered as an underflow (only when T2RCH/T2RCL = 0000H). Timer 2 interrupt will be set if TF2 is set.

The Auto-reload mode (DCEN = 1) is shown in Fig.24.

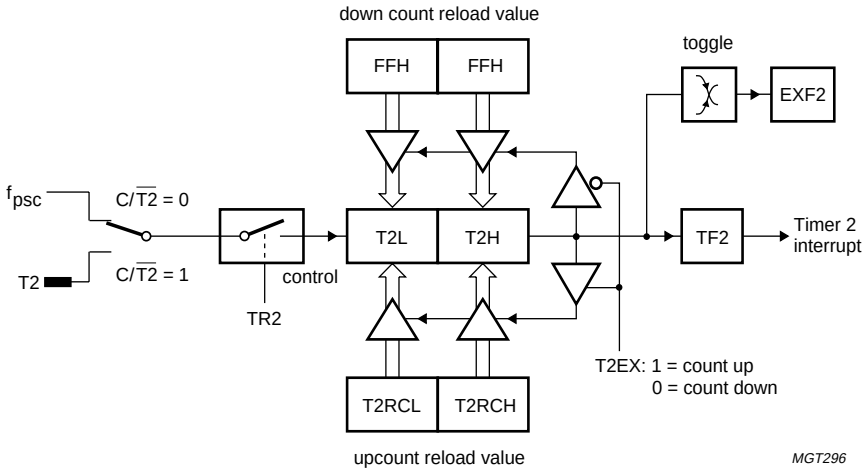


Fig.24 Timer 2 in Auto-reload mode (DCEN = 1).

6.7.5 CLOCK OUTPUT MODE

In the Clock output mode, the output T2OUT is enabled as a clock output. A timer overflow will cause T2H/T2L to be loaded with T2RCH/T2RCL and will toggle output T2OUT. The frequency of pin T2OUT is half the overflow frequency.

Bit T2CON.EXF2 will be set if T2CON.EXEN2 is set and a HIGH-to-LOW transition is detected on T2EX pin.

Timer 2 interrupt will be set only if T2CON.EXF2 is set. This makes an extra external interrupt available.

If Timer 2 does not operate in the Clock output mode, the output T2OUT remains as specified by the I/O SFRs.

The Clock output mode is shown in Fig.25.

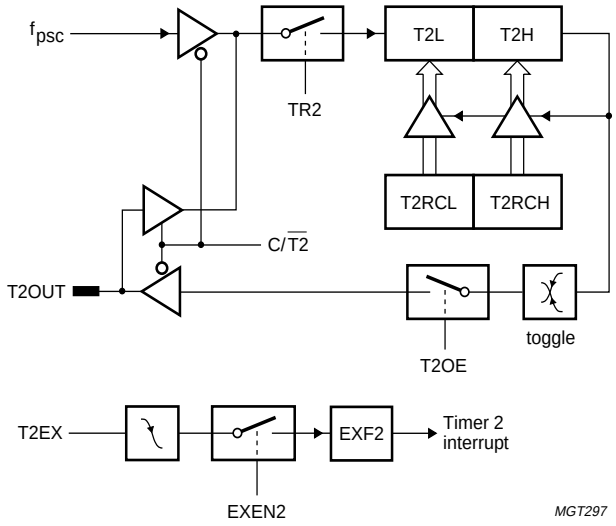


Fig.25 Timer 2 in Clock output mode.

6.8 Watchdog Timer

The Watchdog Timer consists of an 8-bit down counter and a Watchdog prescaler. The binary number defined by bits WD3 to WD0 (WDCON SFR), the Watchdog prescaler and the timer prescaler (f_{psc}) defines the expiration time of the Watchdog Timer. Once enabled this counter runs continuously. Once expired the timer produces firstly an interrupt and finally a reset. The software must reloads the Watchdog Timer at regular intervals to avoid expiration.

A positive edge on bit LD (WDCON SFR) (re)loads the counter with the value of WD3 to WD0, sets the LOW bits to logic 1 and activates this counter if it is not yet running. However, to prepare the (re)loading a positive edge must be applied to the COND bit in WDCON.

In this way at least two locations in software are needed before the counter can be reloaded.

After reset the counter is not running. Only after the first load (LD) it is clocked continuously by a clock pulse.

If the next LD signal is not given within the defined expiration interval an overflow occurs and the processor will be reset (signal WDR). One clock cycle (seen from the Watchdog prescaler output) before the reset is applied an WDI interrupt is issued. This gives the opportunity to avoid the reset if required. The maximum Watchdog Timer expiration time is thus $254/f_{psc}$ to the WD interrupt and $255/f_{psc}$ to the reset.

6.8.1 Watchdog Timer Control Register (WDCON)

The WDCON SFR is used to control the operation of the on-chip Watchdog Timer. If the Watchdog Timer is not loaded after reset, the clock to the Watchdog Timer is switched off for power saving.

Table 63 Watchdog Timer Control Register (SFR address A5H; reset value = 0000 0000)

7	6	5	4	3	2	1	0
COND	WD3	WD2	WD1	WD0	MSKPOL	–	LD

Table 64 Description of WDCON bits

BIT	SYMBOL	DESCRIPTION
7	COND	load condition; control signal from processor
6	WD3	WD0 to WD3 is the preset value for the high nibble of the Watchdog Timer
5	WD2	
4	WD1	
3	WD0	
2	MSKPOL	this bit controls the polarity of the input signal to the MSK modem; MSKPOL = 0: input directly connected to the MSK modem MSKPOL = 1: input inverted and connected to the MSK modem
1	–	reserved, must be kept to logic 0
0	LD	load Watchdog Timer with WD0 to WD3; control signal from CPU

6.8.2 WATCHDOG TIMER PRESCALER REGISTER (WDTIM)

The WDTIM SFR is used to initialize the prescaler of the on-chip Watchdog Timer.

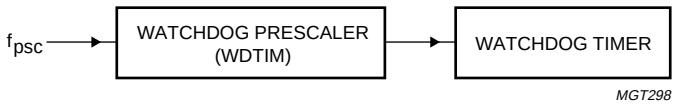


Fig.26 Clocking the Watchdog Timer.

Table 65 Watchdog Timer Prescaler Register (SFR address A6H; reset value is 0000 0000)

7	6	5	4	3	2	1	0
WDTIM.7	WDTIM.6	WDTIM.5	WDTIM.4	WDTIM.3	WDTIM.2	WDTIM.1	WDTIM.0

The expiration time t_{exp} can be calculated as follows:

$$t_{exp} = (\text{prescaler factor}) \times \{2 + 64 \times (\text{WDTIM} + 1)\} \times 16 \times (\text{WDCON} + 1) \times (\text{clock period})$$

Where,

prescaler factor = the dividing factor from prescaler (PSC1 and PSC2); 1, 2, 4, 6, 8, 10, 12 and 16

WDTIM = the 8-bit value (0 to 255) in the Watchdog Timer prescaler

WDCON = the 4-bit value (0 to 15) reloaded in the Watchdog Timer

clock period = the period of the signal applied to pin XTAL1.

From the t_{exp} formulae it follows that, the maximum expiration time

$$t_{exp(max)} = 16 \times (2 + 64 \times 256) \times 16 \times (16) \times (\text{clock period}) = 67\,206\,016 \times (\text{clock period})$$

and the minimum expiration time

$$t_{exp(min)} = 1 \times 66 \times 16 \times (\text{clock period}) = 1056 \times (\text{clock period})$$

6.8.3 EXAMPLE SEQUENCE TO RELOAD THE WATCHDOG TIMER

An example of the reload sequence for the Watchdog Timer:

```
MOV     WDCON,#00h    ;Clear COND and LD bit
ORL     WDCON,#80h    ;Positive edge WDCON.7, Prepare condition
ORL     WDCON,#01h    ;Positive edge WDCON.0, Reload the timer
```

6.9 I²C-bus serial I/O (master/slave interface)

The I²C-bus implements a master/slave I²C-bus interface with integrated shift register, shift timing generation and slave address recognition. I²C-bus standard mode (100 kHz SCLK) and fast mode (400 kHz SCLK) are supported. Low speed mode and extended 10-bit addressing are not supported.

The I²C-bus consists of two lines: a data line (SDA) and a clock line (SCL). These lines also function as the I/O port lines P1.7 and P1.6 respectively. The system is unique because data transport, clock generation, address recognition and bus control arbitration are all controlled by hardware.

The I²C-bus serial I/O has complete autonomy in byte handling and operates in 4 modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver.

These functions are controlled by the S1CON register. S1STA is the Serial Status Register whose contents may also be used as a vector to various service routines. S1DAT is the Data Shift Register and S1ADR the Slave Address Register. Slave address recognition is performed by on-chip hardware. The block diagram of the I²C-bus serial I/O is shown in Fig.27.

The interface between the CPU and the I²C-bus logic, referred to as ‘SIO1’, is accomplished with four Special Function Registers (see Table 66):

The I²C-bus interface is compliant to the specification as described in *“The I²C-bus and how to use it”* (ordering number 9398 393 40011). This paper includes also a detailed description of the I²C-bus protocol.

Table 66 I²C-bus related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
S1CON	Serial Control Register	D8H	0000 0000
S1DAT	Data Shift Register	DAH	0000 0000
S1ADR	Address Register	DBH	0000 0000
S1STA	Serial Status Register	D9H	1111 1000

6.9.1 SERIAL CONTROL REGISTER (S1CON)

The CPU can read from and write to this 8-bit SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^2C-bus. The STO bit is also cleared when ENS1 = 0. Reset initializes S1CON to 00H.

Table 67 Serial Control Register (SFR address D8H)

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

Table 68 Description of S1CON bits

BIT	SYMBOL	DESCRIPTION
7	CR2	This bit along with bits CR1 and CR0 determines the serial clock frequency when SIO is in the Master mode; see Table 69. When CR2 = 0 the I^2C-bus is in fast mode.
6	ENS1	**ENABLE serial I/O.** When ENS1 = 0, the serial I/O is disabled. SDA and SCL outputs are in the high-impedance state; P1.6 and P1.7 function as open-drain ports. When ENS1 = 1, the serial I/O is enabled. Output port latches P1.6 and P1.7 must be set to logic 1; note 1.
5	STA	**START flag.** When this bit is set in Slave mode, the SIO hardware checks the status of the I^2C-bus and generates a START condition if the bus is free or after the bus becomes free. If STA is set while the SIO is in Master mode, SIO will generate a repeated START condition. ENS1 should not be used to temporarily release SIO1 from the I^2C-bus since, when ENS1 is reset, the I^2C-bus status is lost. The AA flag should be used instead.
4	STO	**STOP flag.** When the STO bit is set while SIO1 is in a Master mode, a STOP condition is transmitted to the I^2C-bus. When the STOP condition is detected on the bus, the SIO1 hardware clears the STO flag. In a Slave mode, the STO flag may be set to recover from an error condition. In this case, no STOP condition is transmitted to the I^2C-bus. However, the SIO1 hardware behaves as if a STOP condition has been received and switches to the defined ‘not addressed’ Slave receiver mode. The STO flag is automatically cleared by the hardware. If the STA and STO bits are both set, the STOP condition is transmitted to the I^2C-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. When the STO bit is reset, no STOP condition will be generated.
3	SI	**SIO interrupt flag.** This flag is set, and an interrupt is generated, after any of the following events occur: • A start condition is generated in Master mode • Own slave address has been received during AA = 1 • The general call address has been received while S1ADR0 = 1 and AA = 1 • A data byte has been received or transmitted in Master mode (even if arbitration is lost) • A data byte has been received or transmitted as selected slave • A STOP or START condition is received as selected slave receiver or transmitter. If this flag is set, the I^2C-bus is halted (by pulling down SCL). Received data is only valid until this flag is reset.

BIT	SYMBOL	DESCRIPTION
2	AA	**Assert acknowledge.** When this bit is set, an acknowledge (LOW level to SDA) is returned during the acknowledge clock pulse on the SCL line when: • Own slave address is received • General call address is received (S1ADR.0 = 1) • A data byte is received while the device is programmed to be a master receiver • A data byte is received while the device is a selected slave receiver. When SIO1 is in the addressed Slave transmitter mode, state C8H will be entered after the last serial bit is transmitted. 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 I^2C-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 parts own slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission. When this bit is reset, no acknowledge is returned. Consequently, no interrupt is requested when the own slave address or general call address is received.
1	CR1	These two bits along with the CR2 bit determine the serial clock frequency when SIO is in the Master mode; see Table 69.
0	CR0	

Note

1. If the serial I/O is not enabled (ENS1), the clock to the serial I/O is switched off for power saving.

Table 69 Selection of the serial clock frequency in the Master mode of operation
Bit rates greater than 400 kHz are outside the specified frequency range.

CR2	CR1	CR0	f_{per} DIVISOR	BIT RATE (kHz) AT f_{per}		
				3.58 MHz	4 MHz	6 MHz
0	0	0	10	358	400	(600)
0	0	1	20	179	200	300
0	1	0	30	119.33	133	199.5
0	1	1	40	89.5	100	150
1	0	0	80	44.75	50	75
1	0	1	120	29.83	33	49.5
1	1	0	160	22.38	25	37.5
1	1	1	not valid selection	–	–	–

6.9.2 Data Shift Register (S1DAT)

S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set.

Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7) and after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus.

Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT. Reset initializes S1DAT to 00H.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit.

The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of clock pulses on the SCL line.

When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, the buffer is loaded with the contents of S1DAT.7 which is the first bit to be transmitted to the SDA line. After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

Table 70 Data Shift Register (SFR address DAH)

7	6	5	4	3	2	1	0
S1DAT.7	S1DAT.6	S1DAT.5	S1DAT.4	S1DAT.3	S1DAT.2	S1DAT.1	S1DAT.0

Table 71 Description of S1DAT bits

BIT	SYMBOL	DESCRIPTION
7 to 1	S1DAT.[7:0]	Eight data bits, to be transmitted or just received. A logic 1 in S1DAT corresponds to a HIGH level on the I^2C-bus, and a logic 0 corresponds to a LOW level on the bus. Serial data transmission of S1DAT is MSB first.

6.9.3 Address Register (S1ADR)

The CPU can read from and write to this 8-bit SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a Master mode.

In the Slave modes, the seven most significant bits must be loaded with the microcontrollers own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored. Reset initializes S1ADR to 00H.

Table 72 Address Register (SFR address DBH)

7	6	5	4	3	2	1	0
SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	SLA0	GC

Table 73 Description of S1ADR bits

BIT	SYMBOL	DESCRIPTION
7 to 1	SLA[6:0]	These bits correspond to the 7-bit slave address which will be recognized on the incoming data stream from the I^2C-bus; when the slave address is detected and the interface is enabled, a serial interrupt will be generated to the CPU.
0	GC	This bit is used to determine whether the general CALL address is recognized. When a logic 0, the general CALL address is not recognized; when a logic 1, the general CALL address is recognized.

6.9.4 Serial 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. Reset initializes S1STA to F8H. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = 1).

The status codes for all possible modes of the I^2C-bus interface are given in Table 76.

The contents of this register may be used as a vector to a service routine. This optimizes the response time of the software and consequently that of the I^2C-bus. S1STA is a read-only register.

Table 74 Serial Status Register (SFR address D9H)

7	6	5	4	3	2	1	0
SC4	SC3	SC2	SC1	SC0	0	0	0

Table 75 Description of S1STA bits

BIT	SYMBOL	DESCRIPTION
3 to 7	SC[4:0]	5-bit status code
0 to 2	–	these three bits are held LOW

Table 76 S1STA values in the different modes

S1STA VALUE	DESCRIPTION
MST/TRX mode	
08H	a START condition has been transmitted
10H	a repeated START condition has been transmitted
18H	SLA and W have been transmitted, ACK has been received
20H	SLA and W have been transmitted, $\overline{\text{ACK}}$ received
28H	DATA of S1DAT has been transmitted, ACK received
30H	DATA of S1DAT has been transmitted, $\overline{\text{ACK}}$ received
38H	arbitration lost in SLA, R/W or DATA
MST/REC mode	
38H	arbitration lost while returning $\overline{\text{ACK}}$
40H	SLA and R have been transmitted, ACK received
48H	SLA and R have been transmitted, $\overline{\text{ACK}}$ received
50H	DATA has been received, ACK returned
58H	DATA has been received, $\overline{\text{ACK}}$ returned
SLV/REC mode	
60H	own SLA and W have been received, ACK returned
68H	arbitration lost in SLA, R/W as MST; own SLA and W have been received, ACK returned
70H	general CALL has been received, ACK returned
78H	arbitration lost in SLA, R/W as MST; general CALL has been received
80H	previously addressed with own SLA; DATA byte received, ACK returned
88H	previously addressed with own SLA; DATA byte received, $\overline{\text{ACK}}$ returned
90H	previously addressed with general CALL; DATA byte has been received, ACK has been returned
98H	previously addressed with general CALL; DATA byte has been received, $\overline{\text{ACK}}$ has been returned
A0H	a STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX
SLV/TRX mode	
A8H	own SLA and R have been received, ACK returned
B0H	arbitration lost in SLA, R/W as MST. Own SLA and R have been received, ACK returned
B8H	DATA byte has been transmitted, ACK received
C0H	DATA byte has been transmitted, $\overline{\text{ACK}}$ received
C8H	last DATA byte has been transmitted (AA = 0), ACK received
Miscellaneous	
00H	bus error during MST mode or selected SLV mode, due to an erroneous START or STOP condition
F8H	no information available (reset value). The serial interrupt flag SI, is not yet set

Table 77 Symbols used in Table 76

SYMBOL	DESCRIPTION
SLA	7-bit slave address
R	Read bit
W	Write bit
ACK	acknowledgement (acknowledge bit = logic 0)
$\overline{ACK}$	no acknowledgement (acknowledge bit = logic 1)
DATA	8-bit data byte to or from I^2C-bus
MST	master
SLV	slave
TRX	transmitter
REC	receiver

6.9.5 Modes of Operation

The I^2C-bus logic may operate in any of the following four modes:

- Master transmitter
- Master receiver
- Slave transmitter
- Slave receiver.

As a master, the I^2C-bus logic will generate all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the I^2C-bus will not be released.

Two types of data transfers are possible on the I^2C-bus:

- Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
- Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after each received byte except the last byte. At the end of the last received byte, a ‘not acknowledge’ is returned.

In a given application, SIO1 may operate as a master and as a slave. In the Slave mode, the SIO1 hardware looks for its own slave address and the general call address. If one of these addresses is detected, an interrupt is requested.

When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the Master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the Master mode, SIO1 switches to the Slave mode immediately and can detect its own slave address in the same serial transfer.

6.9.5.1 Master transmitter mode

Serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address (7-bit SLA) of the receiving device and the data direction bit. In this case the data direction bit (R/W) will be a logic 0 (W). Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

In the Master transmitter mode, a number of data bytes can be transmitted to the slave receiver. Before the Master transmitter mode can be entered, S1CON must be initialized with the ENS1 bit set and the STA, STO and SI bits reset. ENS1 must be set to enable the SIO1 interface. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address if they are present on the bus. This will prevent the SIO1 interface from entering a Slave mode.

The Master transmitter mode may now be entered by setting the STA bit. The SIO1 logic will then test the I^2C-bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the Status Register (S1STA) will be 08H.

This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA + W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. The appropriate action to be taken for any of the status codes is detailed in the table. After a repeated start condition (state 10H), SIO1 may switch to the Master receiver mode by loading S1DAT with SLA + R.

6.9.5.2 Master receiver mode

The first byte transmitted contains the slave address of the transmitting device (7-bit SLA) and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (R). Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

In the Master receiver mode, a number of data bytes are received from a slave transmitter. The transfer is initialized as in the Master transmitter mode. When the START condition has been transmitted, the interrupt service routine must load S1DAT with the 7-bit slave address and the data direction bit (SLA + R). The SI bit in S1CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes are possible in S1STA. The appropriate action to be taken for each of the status codes is detailed in the table.

After a repeated start condition (state 10H), SIO1 may switch to the Master transmitter mode by loading S1DAT with SLA + W.

6.9.5.3 Slave receiver mode

Serial data and the serial clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

In the slave receiver mode, a number of data bytes are received from a master transmitter. To initiate the Slave receiver mode, S1ADR must be loaded with the 7-bit slave address to which SIO1 will respond when addressed by a master. Also the least significant bit of S1ADR should be set if the interface should respond to the general call address (00H). The Serial Control Register (S1CON) should be initialized with ENS1 and AA set and STA, STO, and SI reset in order to enter the Slave receiver mode. Setting the AA bit will enable the logic to acknowledge its own slave address or the general call address and ENS1 will enable the interface.

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 logic 0 (W) for SIO1 to operate in the Slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1DAT. This status code should be used to vector to an interrupt service routine, and the appropriate action to be taken for each of the status codes is detailed in Table 76. The Slave receiver mode may also be entered if arbitration is lost while SIO1 is in the Master mode.

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C-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 I^2C-bus.

6.9.5.4 Slave transmitter mode

The first byte is received and handled as in the Slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

In the Slave transmitter mode, a number of data bytes are transmitted to a master receiver. 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 logic 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 the table. The Slave transmitter mode may also be entered if arbitration is lost while SIO1 is in the Master mode.

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 logic 1s as serial data. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C-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 I^2C-bus.

6.9.6 FUNCTIONAL DESCRIPTION I^2C-BUS INTERFACE

6.9.6.1 Input filter

Input signals SDA and SCL from I/O pad cells are synchronized with f_{per}, and spikes shorter than three clock periods are filtered out.

6.9.6.2 Arbitration and control logic

In the Master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I^2C-bus. If another device on the bus overrules a logic 1 and pulls the SDA line LOW, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the Master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a ‘not acknowledge’ (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the ‘mark’ duration is determined by the device that generates the shortest ‘marks,’ and the ‘space’ duration is determined by the device that generates the longest ‘spaces’.

A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

This block also controls all of the signals for serial byte handling. It provides the shift pulses for S1DAT, enables the comparator, generates and detects START and STOP conditions, receives and transmits acknowledge bits, controls the Master and Slave modes, contains interrupt request logic and monitors the I^2C-bus status.

6.9.6.3 Bus clock generator

This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the Master transmitter or Master receiver mode. It is switched off when SIO1 is in a Slave mode. The output frequency is dependent on the CR bits in the control register. The output clock pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

6.9.6.4 Address Register (S1ADR) and comparator

This 8-bit SFR may be loaded with the 7-bit slave address to which SIO1 will respond when programmed as a slave. The least significant bit is used to enable the general call address recognition.

The comparator compares the received 7-bit slave address with its own slave address. It also compares the first received byte with the general call address. If an equality is found, the appropriate status bits are set and an interrupt is requested.

6.9.6.5 Data Shift Register (S1DAT)

This 8-bit SFR contains a byte of serial data to be transmitted or a byte which has just been received. Data in S1DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

6.9.6.6 Serial Control Register (S1CON)

This 8-bit SFR is used by the microcontroller to control the generation of START and STOP conditions, enable the interface, control the generation of ACKs, and to select the clock frequency.

6.9.6.7 Serial Status Register (S1STA)

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I^2C-bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines.

Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the Serial Status Register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code should be sufficient for most of the service routines.

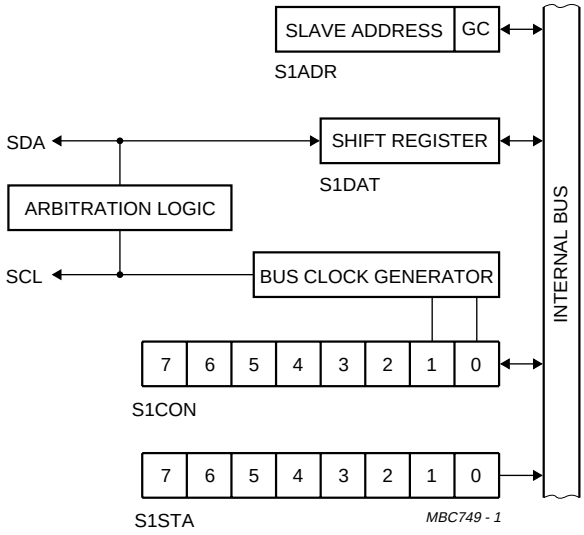


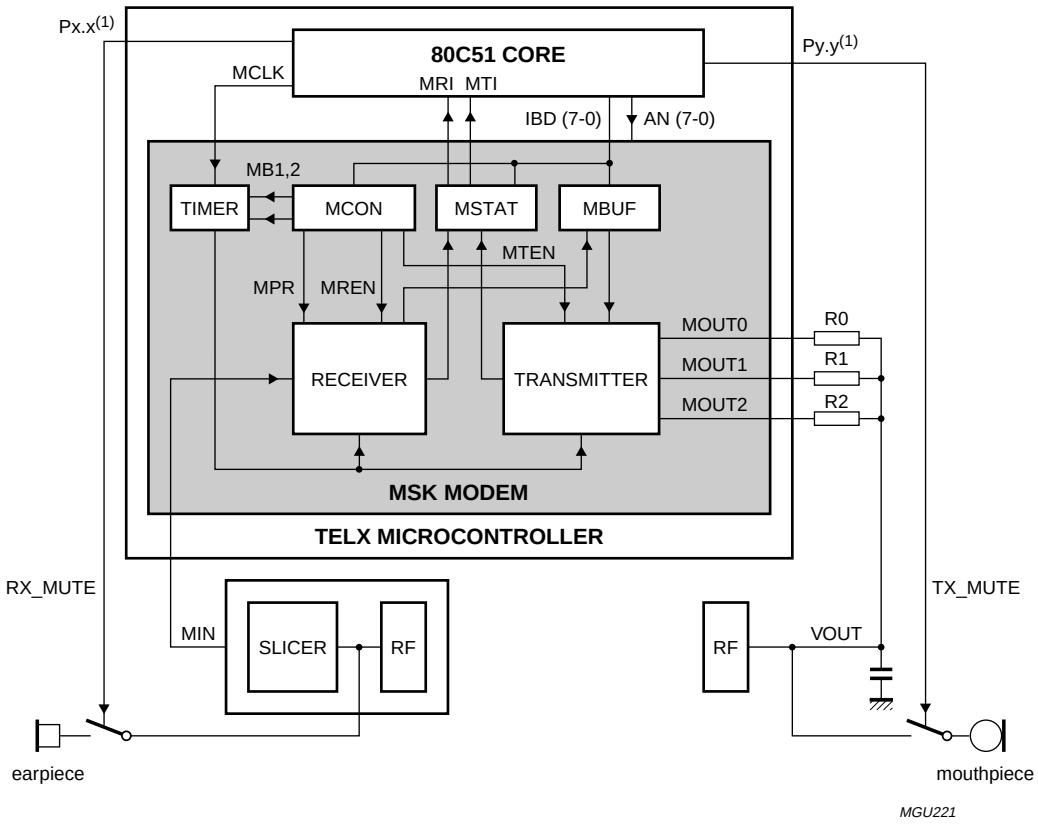
Fig.27 Block diagram of I^2C-bus serial I/O.

6.10 MSK modem

The MSK modem is used for in-band signalling between handset and base in analog cordless telephone systems CT0, CT1 and CT1+. The MSK modem's receiver and transmitter can be enabled separately. Receive and transmit interrupts can wake-up the microcontroller during its power saving Idle mode. Baud rates are programmable. Figure 28 shows the functional diagram of the MSK modem.

The modem has the following features:

- Full duplex operation via 8-bit parallel interface
- The message is fully Manchester coded/decoded
- Automatic detection of 16-bit Manchester preamble pattern
- The last received 4 bits of the preamble pattern are software programmable
- Receiver full, transmitter empty indication bits
- Manchester coding and decoding for clock recovery and early error detection
- Programmable input polarity (see WDCON SFR; Section 6.8.1)
- Baud rate selection of $^{1}/_{2976}f_{per}$, $^{2}/_{2976}f_{per}$, $^{3}/_{2976}f_{per}$ and $^{4}/_{2976}f_{per}$
- Receiver and transmitter off states with no power consumption.



(1) The signals RX_MUTE and TX_MUTE are handled by software. Any available output pin can be used.

Fig.28 MSK modem functional diagram.

6.10.1 80C51 MICROCONTROLLER INTERFACE

The MSK modem block interfaces to the microcontroller via the interrupt signals MRI and MTI and via the control and data SFRs MCON, MSTAT and MBUF. The MSK modem receive and transmit registers are both accessed via the Special Function Register MBUF. Writing to MBUF loads the transmit register and reading MBUF accesses a physically separate receive register.

Table 78 MSK modem related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
MCON	MSK Modem Control Register	D3H	0000 0000
MSTAT	MSK Modem Status Register	D2H	XX00 0000
MBUF	MSK Modem Data Buffer	D1H	0000 0000

6.10.1.1 MSK Modem Control Register (MCON)

Table 79 MSK Modem Control Register (SFR address D3H)

7	6	5	4	3	2	1	0
MPR3	MPR2	MPR1	MPR0	MB1	MB0	MTEN	MREN

Table 80 Description of MCON bits

BIT	SYMBOL	DESCRIPTION
7	MPR3	**Modem preamble pattern.** These 4 bits define the modems preamble pattern.
6	MPR2	
5	MPR1	
4	MPR0	
3	MB1	**Modem transmit/receive frequency.** These 2 bits define the modem transmit/receive frequency; see Table 81.
2	MB0	
1	MTEN	**Modem Transmitter Enable.** If this bit is set the transmitter is active and MOUT[2:0] will get the value ‘100’ if no data is transmitted; if reset, MOUT[2:0] will get the value ‘111’ to zero the currents in the resistive DAC; see note 1.
0	MREN	**Modem Receiver Enable.** If this bit is set the modem receiver is active and scans for Manchester data; see note 1.

Note

1. If both the transmitter and the receiver are disabled (MTEN = 0 and MREN = 0), the clock of the MSK modem is switched off. It is advised to use this state for power saving.

Table 81 Selection of the modems baud rates

MB1	MB0	MODEM BAUD RATE
0	0	$^{1}/_{2976}f_{per}$
0	1	$^{2}/_{2976}f_{per}$
1	0	$^{3}/_{2976}f_{per}$
1	1	$^{4}/_{2976}f_{per}$

6.10.1.2 MSK Modem Status Register (MSTAT)

Table 82 MSK Modem Status Register (SFR address D2H)

7	6	5	4	3	2	1	0
–	–	MRF	MRE	MRP	MRL	MTI	MRI

Table 83 Description of MSTAT bits

BIT	SYMBOL	DESCRIPTION
5	MRF	**Modem Receiver Full flag.** MRF is set when MBUF holds a newly received byte. MRF is reset if the receiver is disabled (MREN = 0) or by clearing MRI. This bit is read-only. Writing to it will have no effect.
4	MRE	**Modem Receiver Error flag.** Indicates the reception of a non-Manchester bit. This bit is set by hardware and is reset by disabling the receiver (MREN = 0) or by clearing MRI. This bit is read-only. Writing to it will have no effect.
3	MRP	**Modem Receiver Preamble flag.** MRP is set by hardware when the modem recognizes the programmed preamble pattern (AAAH) after locking the receiver clock (MRL = 1). MRP is reset by hardware if the receiver is disabled (MREN = 0) or if non-Manchester data is received (MRE = 1). This bit is read-only. Writing to it will have no effect.
2	MRL	**Modem Receiver Clock Locked flag.** This bit is set when the clock of the receiver is locked, i.e. when the receiver has detected three consecutive Manchester bits but has not found the preamble pattern yet. MRL is reset when the receiver detects a non-Manchester bit or when the receiver is disabled. This bit is read-only. Writing to it will have no effect.
1	MTI	**Modem Transmit Interrupt flag.** Indicates MBUF is empty and ready to accept a new byte for transmission. MTI is reset by writing a logic 0 to it. Writing a logic 1 to MTI sets the bit and allows a hardware interrupt to be generated by software.
0	MRI	**Modem Receive Interrupt flag.** Indicates: Modem Receiver Full (MRF = 1) or Modem Receiver Error (MRE = 1) or Modem Receiver Preamble (MRP = 1) or Modem Receiver Clock Locked (MRL = 1). This bit is reset by writing a logic 0 to MRI. A reset of MRI will also reset MRE. Writing a logic 1 to MRI will have no effect.

6.10.1.3 MSK Modem Data Buffer (MBUF)

Table 84 MSK Modem Data Buffer (SFR address D1H)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Table 85 Description of MBUF bits

BIT	SYMBOL	DESCRIPTION
7 to 0	D7 to D0	Writing to MBUF loads the data into the transmit buffer and starts a transmission at MOUT if the transmitter is enabled (MTEN = 1). A new byte can be loaded after MTI is set. If a new byte is loaded before MTI is set the previous byte will be lost. After data has been received at MIN, indicated by MRI, the received byte can be read from MBUF.

6.10.2 MODEM INTERFACE

The modem block has the following modem interface signals,

- **MIN:** MSK Manchester coded input signal from the data slicer
- **MOUT0 to MOUT2:** 3-bit Manchester coded output signal of the modem.

The MSK receiver input can be inverted by programming bit MSKPOL (WDCON.2; see Section 6.8.1):

- MSKPOL = 0: direct connection between MIN pin and MSK receiver
- MSKPOL = 1: inverted connection between MIN and MSK receiver.

The mute signals RX_MUTE and TX_MUTE must be handled by software according to the progress in the data transfer. Any standard I/O port pin can be used for this purpose.

6.10.3 SYNCHRONISATION

When enabled the receiver samples MIN with a frequency $f_{sample} = 8 \times$ baud rate. The sampled values are shifted into an 8-bit shift register. This register is regularly checked to determine whether it contains samples that fulfil the Manchester coding rule, i.e. whether there is a LOW-to-HIGH or a HIGH-to-LOW transition in the middle of the bitcell.

Figure 29a shows a regular, full synchronized bitcell. Figure 29b shows a regular, not synchronized bitcell, this phase shift will be corrected in next received bitcell. Figure 29c (data is faster than internal timebase) and Fig.29d (data is slower than internal timebase) represent a non valid, not synchronized bitcell. In the next received bitcell the data will be re-synchronized but the current data bit does not fulfil the Manchester coding rule and will be lost.

The receiver searches for three consecutive sets of 8 samples that fulfil the Manchester coding rule.

If these three sets have been found the clock is locked (MRL = 1) and the receiver starts looking for the Manchester preamble pattern.

From this point on the receiver uses a Phase-Locked Loop (PLL) to adjust the synchronisation after each received Manchester bit. To detect a sample shift the receiver uses all 8 samples. If the data is maximum one sample out of phase the receiver is able to resynchronize without losing data. If the data is up to three samples out of phase the receiver can still resynchronize but the data is lost. The correction is done by shifting only one sample per bitcell. This means up to three bits cell are needed for full resynchronisation. If the receiver is not able to establish resynchronization within three bitcells the lock bit (MRL) will be reset.

Therefore the MSK modem can receive correct input data with maximum jitter of $1/f_{sample}$.

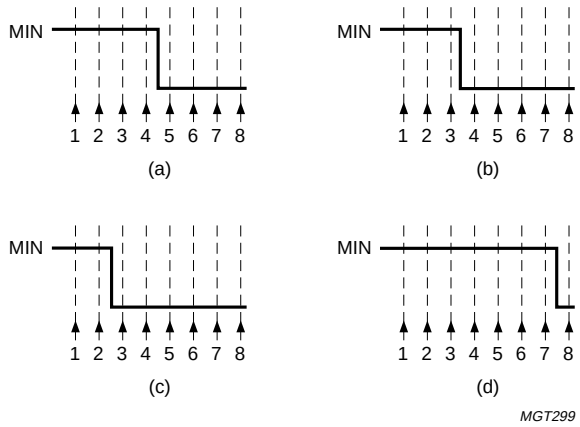


Fig.29 Schematic representation of a bitcell.

6.10.4 DATA RECEPTION

A message is received as a block of one or more data bytes. When enabled, the receiver starts sampling MIN and tries to detect a Manchester pattern. As soon as 3 consecutive Manchester bits are detected the receiver clock is locked (MRL = 1) and the receiver starts scanning the incoming data for the programmed Manchester preamble pattern. When the modem recognizes the preamble pattern, bit MRP is set to a logic 1.

If a non-Manchester bit is detected before finding the preamble pattern then MRL is reset and MRE is set to a logic 1. The synchronisation process has to restart. If the preamble pattern has been detected the receiver starts to Manchester decode the incoming data bits and shifts them into an internal register. After 8 bits the contents of the internal register are copied to MBUF and MRF bit is set to a logic 1. The received byte can be read from MBUF while receiving continues in the internal register. If a non-Manchester bit is received during data reception then MRE is set to a logic 1 and MRL and MRP are reset. The receiver has to resynchronize before receiving new data.

Whenever one of the bits MRF, MRE, MRP and MRL is set the MRI bit is also set and an MSK receive interrupt is generated. This means that when an MSK receive interrupt occurs the 4 status bits have to be polled by software. The bit MRL allows the software to decide very quickly whether an occupied channel contains Manchester coded data or not. The MRP bit is used to find the start of data transmission in a message that is repeated over and over again. MRE is used to detect a Manchester error, which is a violation of the Manchester coding rule that the received level should change in the middle of a bitcell. The MRF bit indicates that the data in MBUF is ready to be read by the software.

During data reception the minimum time between two settings of MRF (each one generating an MRI interrupt) is:

$$t_{min} = \frac{8}{\text{baud rate}}$$

Figure 30 shows an example of the data reception timing diagram.

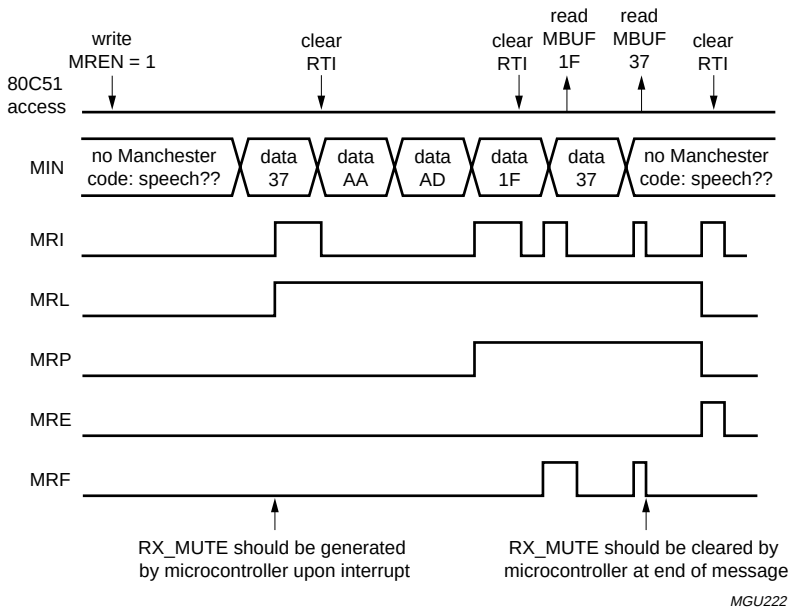


Fig.30 Data reception timing diagram.

6.10.5 DATA TRANSMISSION

Data transmission is enabled if bit MTEN in register MCON is set to a logic 1. If MTEN is a logic 0, data transmission is disabled and MOUT[2:0] is set to '111' to zero the currents in the resistive DAC. Setting MTEN to a logic 1 sets MOUT[2:0] to the idle value '100'. This results in a value close to $\frac{1}{2}V_{DD}$ on the output signal of the external DAC. Transmission is started by loading the first byte into register MBUF. All bytes are transmitted starting with the MSB.

A message is transferred in a block of 3 or more bytes, the first two bytes being the programmed Manchester preamble pattern. In order to insert the preamble pattern, the first two bytes AAH and AxH (with 'x' being the MPR3 to MPR0 value programmed in the receiver MSK modem) have to be written to MBUF by software.

After this, the first byte of the message is written to MBUF. As soon as MBUF is ready to accept new input, signal MTI is set.

The minimum time between two MTI interrupts is:

$$t_{min} = \frac{8}{\text{baud rate}}$$

If no new byte is written to MBUF at the end of a byte transmission, the modem transmitter stops transmission and MOUT[2:0] is set to the idle state '100'.

MTI must be cleared explicitly. If MTEN is reset during transmission, the transmitter will finish the transmission of the current byte and then will set MOUT[2:0] to the off state '111'. No interrupt on MTI will be generated at the end of the transmission.

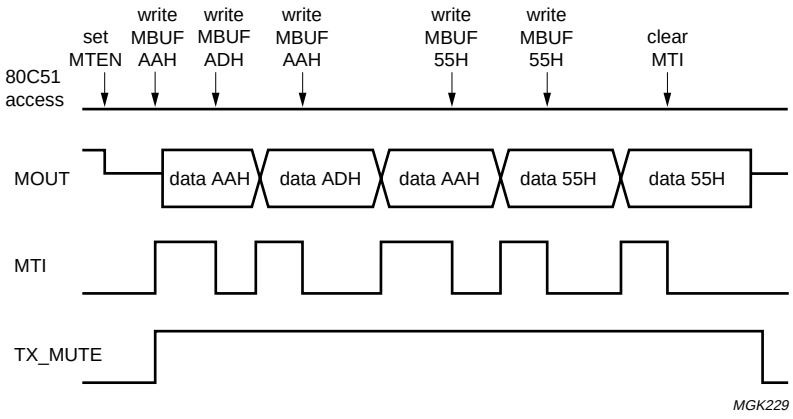


Fig.31 Data transmission timing diagram.

6.10.6 WAVEFORM GENERATION WITH MOUT[2:0]

The 3 digital output pins MOUT0 to MOUT2, should be used as an input to a 3-bit external DAC. The signals can be connected via external resistors R0, R1 and R2 to a summation point and then be filtered with an external capacitor (C1). The 3-bit DAC is shown in Fig.32. Table 86 gives the relationship between the MOUT pins and VOUT.

Figure 33 shows the waveforms that are produced by the waveform generator. The horizontal axis shows the sample counter on which the waveform changes its value. Each bit is built-up out of 2×124 samples.

The vertical axis shows the values of MOUT[2:0], forming the inputs of the resistive DAC. The first half of the waveform is determined by the previous and the current bit, whereas the second half of the waveform is determined by the current and the next bit to be transmitted. The count frequency of the sample counter depends on the programmed baud rate.

If the transmitter is disabled with MTEN set to a logic 0, MOUT[2:0] is '111' to save power in the resistive DAC. If the transmitter is enabled and no data is transmitted, MOUT[2:0] has an idle value of '100', which corresponds to $0.57V_{DD}$.

Table 86 VOUT as a function of MOUT[2:0]

MOUT2	MOUT1	MOUT0	VOUT(1)
0	0	0	0
0	0	1	$0.14V_{DD}$
0	1	0	$0.29V_{DD}$
0	1	1	$0.43V_{DD}$
1	0	0	$0.57V_{DD}$
1	0	1	$0.71V_{DD}$
1	1	0	$0.86V_{DD}$
1	1	1	V_{DD}

Note

1. VOUT with resistor values (see Fig.32):
   R1 = 0.5R0; R2 = 0.25R0

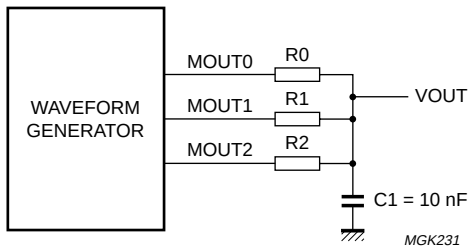


Fig.32 3-bit DAC with MOUT[2:0].

6.10.7 MANCHESTER CODING OF DATA

The bits of the data byte written in MBUF are Manchester encoded as shown in Fig.33. A logic 1 is coded as a LOW-to-HIGH transition in the middle of a bitcell, a logic 0 is coded as a HIGH-to-LOW transition.The Manchester encoded signal contains redundancy for early error detection in received bits. A non-matching HIGH-to-LOW or LOW-to-HIGH pair indicates an error condition.The Manchester encoded signal has a polarity change in each bitcell.

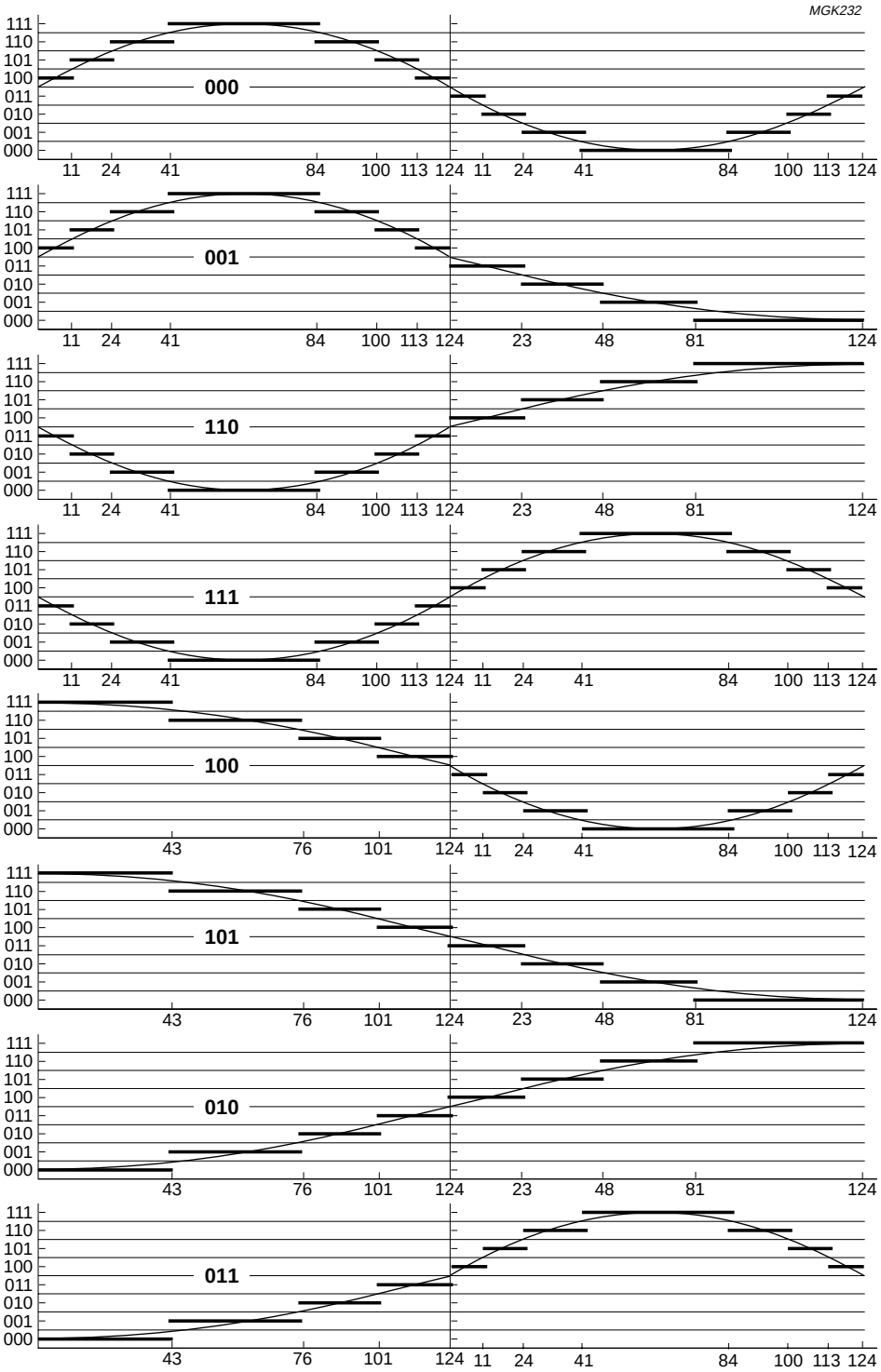


Fig.33 Waveforms with MOUT[2:0] for previous, current and next bits to be transmitted.

6.11 AD/DA converter

6.11.1 GENERAL

The P87CL888 is equipped with a 8-bit Digital-to-Analog Converter (DAC) with a bandgap reference voltage and an analog comparator. The block diagram is shown in Fig.34. The block has an automatic conversion mode, which performs a successive approximation conversion in hardware. There is also a manual mode available where the user can optimize the analog-to-digital conversion for either speed or performance.

Six user selectable ADC channels are present plus one channel, which is connected to V_{DD} for supply voltage measurements. The ADC/DAC has two DAC outputs (DAC1 and DAC0), that are controlled by switches SW1 and SW0. Several power-down modes are present to activate only the parts needed for a conversion.

The ADC/DAC converter related SFRs are shown in Table 87.

Table 87 ADC/DAC converter related SFRs

SFR	DESCRIPTION	SFR ADDRESS	RESET VALUE
ADCON0	ADC Control 0 Register	C4H	0000 0000
ADCON1	ADC Control 1 Register	C5H	0000 0000
ADCDAT	ADC Data Register	C6H	0000 0000
ADCPS	ADC/DAC Port function Select Register	C7H	0000 0000

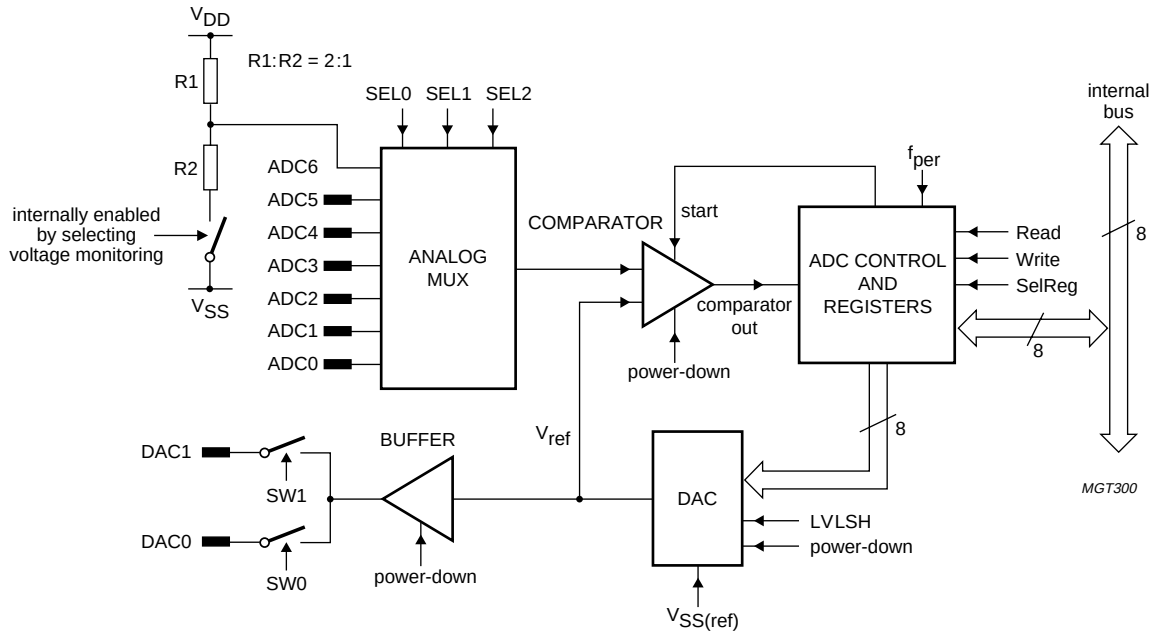


Fig.34 Basic block diagram of the successive approximation ADC.

6.11.2 Analog-to-digital converter automatic conversion mode

To operate the ADC in automatic conversion mode, bit ACM in register ADCON0 must be set. Before starting a conversion the ADC input selector has to be configured by setting bits SEL2 to SEL0 in register ADCON1. To reduce the current consumption in the mixed signal I/O pins the digital path of these pins must be disabled by setting bits ADIS5 to ADIS0 in register ADCPS. Figure 35 shows a diagram with such a mixed signal I/O pin and the disable gate for the digital path.

The conversion time of an analog-to-digital conversion can be optimized by defining the number of clock pulses needed for the DAC to generate a stable output.

The value configured in bits ST3 to ST0 in register ADCON1 define the number of clock pulses of input clock signal f_{per}. For slow clock signals a lower number of clocks can be chosen for the conversion as long as the DAC settling time as specified in Chapter 12 is still fulfilled.

An automatic analog-to-digital conversion is started by setting bit STRT in register ADCON0. After completion of an 8-bit conversion an interrupt is generated and the RDY flag in register ADCON0 is set. At the same time STRT is reset by hardware. RDY flag must be cleared by software.

If automatic conversion mode is selected MM0 and MM1 are set to logic 0 (ADC block switched on) and DAC0 and DAC1 are disabled.

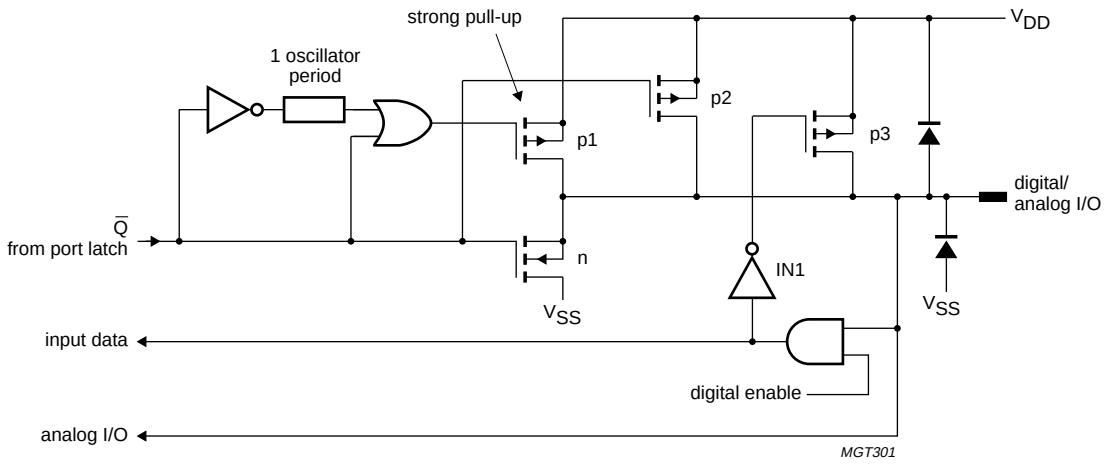


Fig.35 Basic structure of a mixed signal pin with disable for digital path.

6.11.3 Analog-to-digital converter manual mode

Next to the automatic mode one can also perform analog to digital conversion by controlling the DAC and comparator separately.

Via the available SFRs the user program can write an 8-bit value to the DAC and let the comparator compare the result with the analog input. An interrupt alerts the program when the conversion is ready.

For a full 8-bit conversion the program should write 8 times to the DAC register and wait for the result from the comparator.

Of course the user can finish the conversion earlier by making a smaller number of iterations.

When the manual mode conversion is chosen the DAC output buffer can be put into Power-down mode for lower current consumption.

6.11.4 Digital- to-analog converter: DTMF tone generation

The P87CL888 is equipped with an 8-bit digital-to-analog converter with an output voltage which is referenced to a bandgap voltage. The $V_{SS(ref)}$ can be connected to a 'clean' V_{SS} reference. By selecting the switch bits (SW1 and SW0) the analog output of the digital-to-analog converter is fed through a buffer with a low output resistance to device pins DAC0 and DAC1. Thus two pins are available to put out the DAC output signal. This is useful e.g. for the generation of DTMF tones and melodies. The melodies can be sent to one pin, the DTMF tones to the other pins. Port pins used as analog I/Os must be configured as 'high-impedance' by setting the port configuration SFR accordingly.The DAC is connected to the microcontroller via the Special Functions Registers. Both DAC and output buffer can be placed in a Power-down mode where they consume no power.

6.11.5 ADC Control 0 Register (ADCON0)

Table 88 ADC Control 0 Register (SFR address C4H; reset value = 0000 0000)

7	6	5	4	3	2	1	0
RDY	STRT	ACM	MM1	MM0	SW1	SW0	COMP

Table 89 Description of ADCON0 bits

BIT	SYMBOL	DESCRIPTION
7	RDY	ADC ready flag; set when automatic AD conversion is completed. Set by hardware; reset by software. RDY requests the interrupt ADI.
6	STRT	Setting STRT by software starts an automatic AD or DA conversion; this bit is cleared by hardware after a conversion terminated.
5	ACM	0: manual analog-to-digital conversion mode is selected; 1: automatic analog-to-digital conversion mode is selected
4 and 3	MM[1:0]	00: sensing mode, output buffer, DAC and comparator are on; 01: ADC mode, DAC and comparator are on, output buffer is off; 10: DAC mode, DAC and output buffer are on, comparator is off; 11: Power-down mode, output buffer, DAC and comparator are off; note 1
2	SW1	0: DAC1 output is disabled; 1: DAC1 output is enabled
1	SW0	0: DAC0 output is disabled; 1: DAC0 output is enabled
0	COMP	This bit is cleared by hardware, writing to it will have no effect; 0: analog input < DAC reference; 1: analog input > DAC reference

Note

1. If the ADC/DAC is disabled (MM[1:0] = 11), the clock to the DAC is switched off for power saving.

6.11.6 ADC CONTROL 1 REGISTER (ADCON1)

Table 90 ADC Control 1 Register (SFR address C5H; reset value = 0000 0000)

7	6	5	4	3	2	1	0
ST3	ST2	ST1	ST0	LVLSH	SEL2	SEL1	SEL0

Table 91 Description of ADCON1 bits

BIT	SYMBOL	DESCRIPTION
7 to 4	ST[3:0]	DAC settling time in number of clocks of f_{per}: 0000: 1 clock 0001: 2 clocks ..... 1111: 16 clocks
3	LVLSH	0: normal DAC levels 1: shifted DAC levels (can be used for DTMF tone generation)
2 to 0	SEL[2:0]	000: select ADC0 001: select ADC1 010: select ADC2 011: select ADC3 100: select ADC4 101: select ADC5 110: select supply voltage monitor 111: not used

6.11.7 ADC DATA REGISTER (ADCDAT)

DC[7:0] is the input data for the DAC converter, and output data from automatic analog-to-digital conversions.

Table 92 ADC Data Register (SFR address C6H)

7	6	5	4	3	2	1	0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

6.11.8 ADC/DAC PORT FUNCTION SELECT REGISTER (ADCPS)

With the ADCPS register the alternative function of the ADC/DAC ports as digital input can individually be disabled. This reduces the current consumption in the digital input path, when the analog function of the port is used. When a port is used for ADC or DAC analog I/O, it must be configured as a high-impedance port (PnCFGA = 0, PnCFGB = 1; see Section 6.5).

Table 93 ADC/DAC Port function Select Register (SFR address C7H; reset value = 0000 0000)

7	6	5	4	3	2	1	0
DACS1	DACS0	ADIS5	ADIS4	ADIS3	ADIS2	ADIS1	ADIS0

Table 94 Description of ADCPS bits

BIT	SYMBOL	DESCRIPTION
7	DACS1	1: at P3.4 digital input disabled; 0: at P3.4 digital input enabled
6	DACS0	1: at P3.3 digital input disabled; 0: at P3.3 digital input enabled
5	ADIS5	1: at P1.0 digital input disabled; 0: at P1.0 digital input enabled
4	ADIS4	1: at P1.1 digital input disabled; 0: at P1.1 digital input enabled
3	ADIS3	1: at P1.2 digital input disabled; 0: at P1.2 digital input enabled
2	ADIS2	1: at P3.7 digital input disabled; 0: at P3.7 digital input enabled
1	ADIS1	1: at P3.6 digital input disabled; 0: at P3.6 digital input enabled
0	ADIS0	1: at P3.5 digital input disabled; 0: at P3.5 digital input enabled

6.12 Internal Data Memory

Internal Data Memory is mapped in Fig.36. The memory space is divided into three blocks, which are referred to as the lower 128, the upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Fig.36 shows the upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 80C51 devices as mapped in Fig.37. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use.

This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing. The next 16 bytes above the register banks form a block of bit-addressable memory space. The 80C51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH. All of the bytes in the lower 128 can be accessed by either direct or indirect addressing.

The upper address space of 128 bytes is overlaid with the 128-byte SFR address space. When using indirect addressing the internal data memory is accessed but when using direct addressing the SFR memory space is accessed. Figure 36 shows the overlay of internal Data Memory and SFR memory space. SFRs include the Port latches, timers, peripheral controls, etc. Sixteen addresses in SFR space are both byte-and bit-addressable. The bit-addressable SFRs are those whose address ends in 0H or 8H.

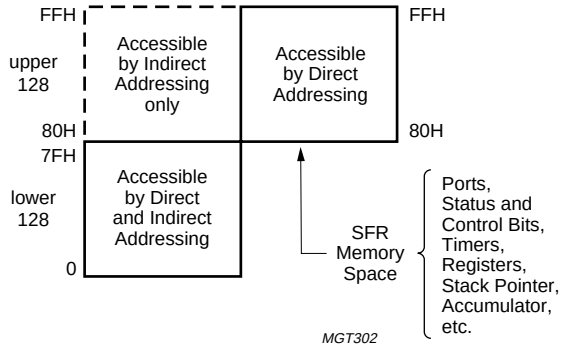


Fig.36 Internal Data Memory.

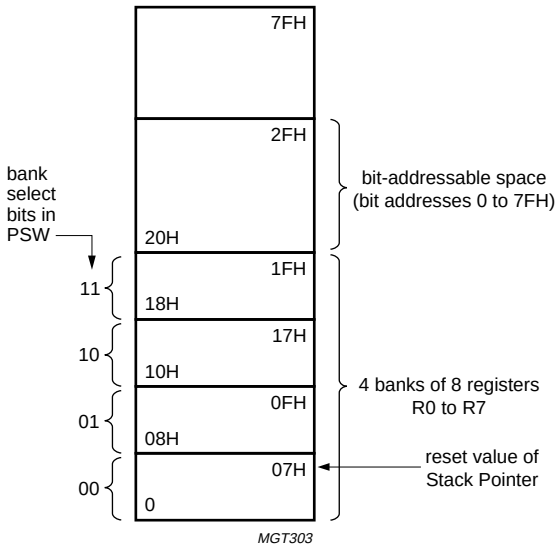


Fig.37 The lower 128 bytes of internal Data Memory.

6.13 Special Function Registers Overview

Table 95 SFRs overview

An empty field (–) in this map indicates a bit that can be read or written to by software.

ADDR (HEX)	R/W	BIT ADDRESSABLE	NAME	7	6	5	4	3	2	1	0	RESET VALUE
80	RW	yes	P0	–	–	–	–	–	–	–	–	EFH
81	RW	–	SP	–	–	–	–	–	–	–	–	07H
82	RW	–	DPL	–	–	–	–	–	–	–	–	00H
83	RW	–	DPH	–	–	–	–	–	–	–	–	00H
87	RW	–	PCON	–	–	–	–	–	–	PD	IDL	00H
88	RW	yes	TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
89	RW	–	TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	00H
8A	RW	–	TL0	–	–	–	–	–	–	–	–	00H
8B	RW	–	TL1	–	–	–	–	–	–	–	–	00H
8C	RW	–	TH0	–	–	–	–	–	–	–	–	00H
8D	RW	–	TH1	–	–	–	–	–	–	–	–	00H
8E	RW	–	P0CFGA	–	–	–	–	–	–	–	–	F3H
8F	RW	–	P0CFGB	–	–	–	–	–	–	–	–	00H
90	RW	yes	P1	–	–	–	–	–	–	–	–	FFH
9E	RW	–	P1CFGA	–	–	–	–	–	–	–	–	FFH
9F	RW	–	P1CFGB	–	–	–	–	–	–	–	–	00H
A5	RW	–	WDCON	COND	WD3	WD2	WD1	WD0	MSKPOL	–	LD	00H
A6	RW	–	WDTIM	–	–	–	–	–	–	–	–	00H
A8	RW	yes	IEN0/IE	EA	ET2	ES1	–	ET1	EX1	ET0	EX0	00H
B0	RW	yes	P3	–	–	–	–	–	–	–	–	FFH
B4	RW	–	SYSCON	T1SRC1	T1SRC0	T0SRC1	T0SRC0	–	–	SELECT	XTM	00H
B8	RW	yes	IP0	–	PT2	PS1	–	PT1	PX1	PT0	PX0	00H
BE	RW	–	P3CFGA	–	–	–	–	–	–	–	–	FFH
BF	RW	–	P3CFGB	–	–	–	–	–	–	–	–	00H
C0	RW	yes	IRQ1	IQ9	IQ8	IQ7	IQ6	IQ5	IQ4	IQ3	IQ2	00H
C1	RW	–	P4	–	–	–	–	–	–	–	–	FEH
C4	RW	–	ADCON0	RDY	STRT	ACM	MM1	MM0	SW1	SW0	COMP	00H
C5	RW	–	ADCON1	ST3	ST2	ST1	ST0	LVLSH	SEL2	SEL1	SEL0	00H

ADDR (HEX)	R/W	BIT ADDRESSABLE	NAME	7	6	5	4	3	2	1	0	RESET VALUE
C6	RW	–	ADCDAT	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0	00H
C7	RW	–	ADCPS	DACS1	DACS0	ADIS5	ADIS4	ADIS3	ADIS2	ADIS1	ADIS0	00H
C8	RW	yes	T2CON	T2F	EXF2	–	–	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$	00H
C9	RW	–	T2MOD	–	–	–	–	–	T2RD	C/T2OE	CP/DCEN	00H
CA	RW	–	T2RCL	–	–	–	–	–	–	–	–	00H
CB	RW	–	T2RCH	–	–	–	–	–	–	–	–	00H
CC	RW	–	T2L	–	–	–	–	–	–	–	–	00H
CD	RW	–	T2H	–	–	–	–	–	–	–	–	00H
CE	RW	–	P4CFGA	–	–	–	–	–	–	–	–	FFH
CF	RW	–	P4CFGB	–	–	–	–	–	–	–	–	00H
D0	RW	yes	PSW	CY	AC	F0	RS1	RS0	OV	–	P(1)	00H
D1	RW	–	MBUF	D7	D6	D5	D4	D3	D2	D1	D0	00H
D2	RW	–	MSTAT	–	–	MRF(1)	MRE(1)	MRP(1)	MRL(1)	MTI	MRI(2)	00H
D3	RW	–	MCON	MPR3	MPR2	MPR1	MPR0	MB1	MB0	MTEN	MREN	00H
D4	RW	–	OAL	A7	A6	A5	A4	A3	A2	A1	A0	00H
D5	RW	–	OAH	A15	A14	A13	A12	A11	A10	A9	A8	00H
D6	RW	–	ODATA	D7	D6	D5	D4	D3	D2	D1	D0	00H
D7	RW	–	OTEST	–	–	–	–	–	–	MODE1	MODE0	00H
D8	RW	yes	S1CON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	00H
D9	R	–	S1STA	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
DA	RW	–	S1DAT	–	–	–	–	–	–	–	–	00H
DB	RW	–	S1ADR	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	SLA0	GC	00H
DC	RW	–	OISYS	–	–	–	–	–	VPON	WE	InSysMode	00H
DD	RW	–	WKCON	–	–	–	–	–	–	–	–	00H
E0	RW	yes	ACC	–	–	–	–	–	–	–	–	00H
E1	RW	–	ISE1	ISE9	ISE8	ISE7	ISE6	ISE5	ISE4	ISE3	ISE2	00H
E8	RW	yes	IEN1	EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2	00H
E9	RW	–	IX1	IX9	IX8	IX7	IX6	IX5	IX4	IX3	IX2	00H
F0	RW	yes	B	–	–	–	–	–	–	–	–	00H
F1	RW	–	IEN2	EWDI	EADI	EKPI	–	ELVD	–	EMTI	EMRI	00H
F3	RW	–	PRESC	EXTCK	AUXSW	SYNC	–	–	–	–	–	00H

ADDR (HEX)	R/W	BIT ADDRESSABLE	NAME	7	6	5	4	3	2	1	0	RESET VALUE
F4	RW	–	IX3	IXKP7	IXKP6	IXKP5	IXKP4	IXKP3	IXKP2	IXKP1	IXKP0	00H
F5	RW	–	IRQ3	IQKP7	IQKP6	IQKP5	IQKP4	IQKP3	IQKP2	IQKP1	IQKP0	00H
F6	RW	–	IEN3	EKP7	EKP6	EKP5	EK4P	EKP3	EKP2	EKP1	EKP0	00H
F7	RW	–	ISE3	ISKP7	ISKP6	ISKP5	ISKP4	ISKP3	ISKP2	ISKP1	ISKP0	00H
F8	RW	yes	IP1	PX7	PX6	PX5	PX6	PX5	PX4	PX3	PX2	00H
F9	RW	–	IP2	PWDI	PADI	PKPI	–	PLVD	–	PMTI	PMRI	00H

Notes

1. This bit is read only.
2. This bit is set to logic 1 by hardware; can only be cleared by software.

7 GENERAL DESCRIPTION OF DEVICE PROGRAMMING

7.1 Introduction

The non-volatile programmable memory is implemented using EPROM technology. The bits are by default set to logic 1 and therefore reading data from a blank One Time Programmable (OTP) memory will result in a value of 'FFH'. During programming, every bit can be altered to logic 0, once.

The interface to the OTP memory has three operating modes:

- **Normal mode** (CPU code fetch). In Normal mode the OTP memory provides the instructions to the CPU. Fetching of code is performed purely self-timed, started with a request from the CPU and finished by an acknowledge from the OTP memory when data is ready.
- **Parallel programming mode**. In Parallel programming mode the OTP can be programmed by an external programmer. In this mode, the CPU is idle and the OTP is externally controlled.
- **In-system programming mode**. With the In-system programming mode the OTP can be programmed under control of the microcontroller. This mode can be used to program data or code while the IC is already built into a system, e.g. to store tuning parameters but also to program the complete customer code.

The Parallel programming and In-system programming modes are described in Sections 7.4 and 7.5.

7.2 SFRs for device programming

The OTP interface uses five SFRs which are listed in Table 96. The ODATA and OISYS registers are exclusively used for In-system programming. The other registers are also used for Parallel programming.

Table 96 Overview of device programming SFRs

ADDRESS	SFR	DESCRIPTION	RESET VALUES	USAGE
D4H	OAL	OTP Address Low Register	00H	Parallel programming, In-system programming
D5H	OAH	OTP Address High Register	00H	
D6H	ODATA	OTP Data Register	00H	In-system programming
D7H	OTEST	OTP Test Register	00H	all modes
DCH	OISYS	OTP In-system Register	00H	In-system programming

Table 97 OTP Address Low Register (OAL; SFR address D4H)

7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Table 98 OTP Address High Register (OAH; SFR address D5H)

7	6	5	4	3	2	1	0
A15	A14	A13	A12	A11	A10	A9	A8

Table 99 OTP Data Register (ODATA; SFR address D6H)

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Table 100 OTP Test Register (OTEST; SFR address D7H)
Bits OTEST[7:2] are not used and must be set to logic 0.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	MODE1	MODE0

Table 101 Mode select defined by bits MODE1 and MODE0

MODE1	MODE0	DESCRIPTION
0	0	normal mode, no test mode selected
0	1	verify mode selected
1	0	don't use
1	1	

Table 102 OTP In-system Register (OISYS; SFR address DCH)

7	6	5	4	3	2	1	0
–	–	–	–	–	VPON	WE	InSysMode

Table 103 Description of OISYS bits

BIT	SYMBOL	DESCRIPTION
7 to 3	–	not used
2	VPON	This bit is read only. VPON = 0: voltage on V_{PP} pin is below high voltage detection level. VPON = 1: voltage on V_{PP} pin is above high voltage detection level.
1	WE	Write enable; set up the programming pulse, when HIGH. An enter Idle mode is required to really start the programming pulse.
0	InSysMode	In-system programming status bit

7.3 Address space mapping

The OTP memory has the user address space and additionally 512 bytes of test memory. The test memory also includes the signature bytes. The OTP interface and control registers are dimensioned for 64-kbyte address space with address lines numbered from A15 down to A0. The test array has been mapped into the user address space with start address at 8000H (32K, A15 = 1). Figure 38 shows the memory map for the user addresses and the test addresses.

For the OTP memory in the test address space no retention nor programmability can be guaranteed, and therefore it should not be used for customer purposes. Also the test OTP memory might already be partially programmed by test routines.

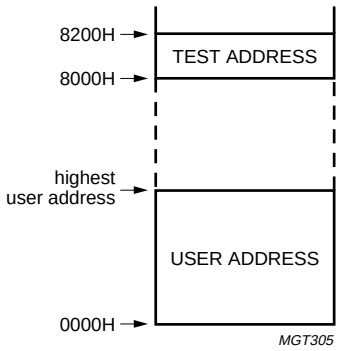


Fig.38 Address mapping.

7.4 The Parallel programming mode

7.4.1 GENERAL

In Parallel programming mode the OTP can be programmed with an EPROM programmer. Programming can be done in single-byte or multiple-byte programming mode. An internal address counter is implemented to accelerate the programming and verify sequence. To program the OTP, multiplexing of address-high/low and data is used.

One bidirectional 8-bit port is used to supply addresses and data and to read back data.

Figure 39 illustrates a coarse block diagram of the OTP interface in Parallel programming mode. In this configuration, the signals that are necessary to program the OTP are shown in Table 104.

The connection between the programmer and the device is shown in Fig.40.

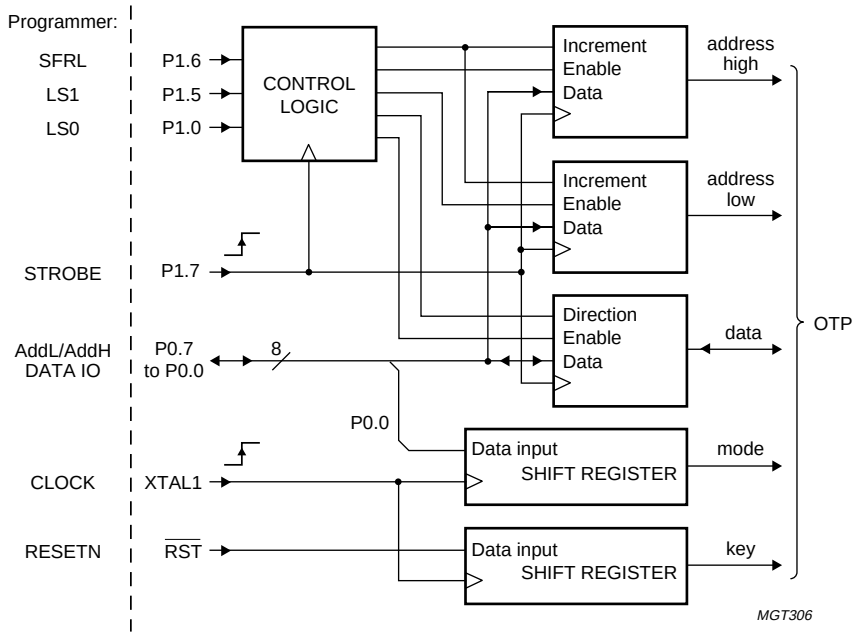


Fig.39 The OTP interface in Parallel programming mode.

Table 104 Signals for Parallel programming mode

SIGNAL	TYPE	DEVICE PIN	FUNCTION DURING PROGRAMMING
VPE	Supply	V_{PP}/PORENABLE	programming voltage
V_{DD}	Supply	V_{DD}	positive supply
GND	Supply	V_{SS}	negative supply
AddL/AddH/Data IO	I/O	Port 0 (P0.[7:0])	address, data in, data out
SFRL(1)	I	P1.6	latch select mode
LS1(1)	I	P1.5	latch select 1
LS0(1)	I	P1.0	latch select 0
STROBE	I	P1.7	strobe
RESETN	I	$\overline{RST}$	reset not signal
CLOCK	I	XTAL1	mode entry clock signal

Note

1. With the signals SFRL, LS0 and LS1 an internal function is selected according to Table 105. The signals are latched with the rising edge of STROBE, and the operation is started. The mode signals need therefore to be stable before a STROBE transition to HIGH.

Table 105 Mode control signals

SFRL	LS1	LS0	PORT 0 DIRECTION	FUNCTION
0	0	0	output	read; with address increment
0	0	1	output	read; counter disabled
0	1	0	input	not used
0	1	1	input	write
1	0	0	input	select test control latch
1	0	1	input	select address low latch
1	1	0	input	select address high latch
1	1	1	input	select internal data latch in multiple byte programming mode

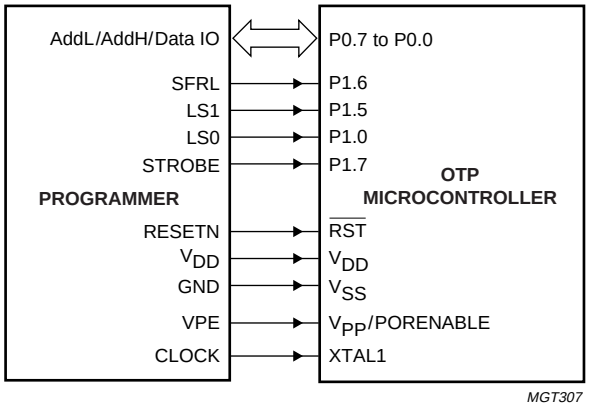


Fig.40 Parallel programming mode.

7.4.2 Entering Parallel programming mode

An external programmer is connected to the IC as illustrated in Fig.40. After power-on the ports P0 and P1 can be output ports, dependent on their reset configuration. Therefore the Parallel programming mode must first be entered by serially shifting in the mode and key data via $\overline{RST}$ and P0.0. The data bits are latched at the rising edge of XTAL1. After this entry the pin functions are as defined in Table 104, and the IC may be controlled by the external programmer for OTP programming.

The IC will return to its Normal mode after a chip reset: taking the $\overline{RST}$ pin LOW for at least 8 clocks, and then HIGH for at least 16 clocks. The CPU will then start executing code from the OTP memory.

The MODE and KEY entry sequence is described in Table 106 and Fig.41. Timing parameters are listed in Chapter 13.

Table 106 OTP Parallel programming mode entry sequence

STEP	V_{DD} (V)	KEY (MSB FIRST)		MODE (MSB FIRST)		XTAL1
		$\overline{RST}$	NAME	P 0.0	NAME	
1	0 to 3.3	HIGH	–	LOW	–	LOW
2	3.3	LOW	–	LOW	–	>1500 clocks
3	3.3	B3H	HOLD_CPU	LOW	–	8 clocks
4	3.3	B2H	LOAD_MODE	05H	OTP_PAR_MODE	8 clocks
5	3.3	B1H	DEVICE_MODE	LOW	–	8 clocks
6	3.3	HIGH	AddL/AddH/Data at P0 with STROBE at P1.7			X

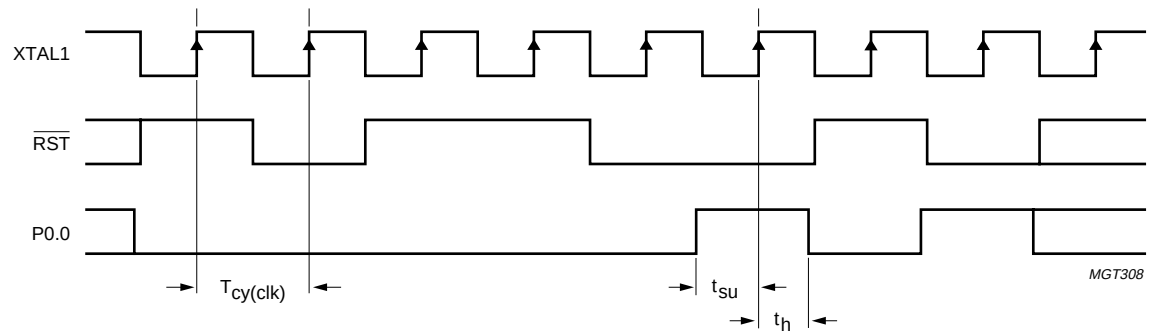


Fig.41 Example of entering LOAD_MODE ($\overline{RST}$ = B2H) and OTP_PAR_MODE (P0.0 = 05H).

7.4.2.1 Entering a KEY

Entering a KEY via $\overline{RST}$ is done by shifting in a byte (MSB first) with the rising edge of CLOCK at pin XTAL1:

```
FOR i IN 7 DOWNTO 0 LOOP
  RST_N = KEY(i)
  XTAL1 = low
  XTAL1 = high
END LOOP.
```

7.4.2.2 Entering MODE and KEY

Entering the MODE (OTP_PAR_MODE via P0.0) is done in parallel with entering the KEY (LOAD_MODE via $\overline{RST}$); see Fig.41:

```
FOR i IN 7 DOWNTO 0 LOOP
  RST_N = KEY(i)
  P0.0 = MODE(i)
   XTAL1 = low
  XTAL1 = high
END LOOP
```

7.4.3 PARALLEL PROGRAMMING MODE INITIALIZATION

Firstly the OTP Parallel programming mode must be initialized. A diagram with the timing of the control signals for the parallel programming initialization is shown in Fig.42.

For the read mode of the memory the register OTEST (address D7H) has to be set to 'Verify mode (01H)'. This mode simulates the most critical read condition which can occur over the whole voltage and temperature range and over the life time of the IC. The SFR OTEST is selected with SFRL = 1, LS1 = 0 and LS0 = 0.

The data '01H' is applied to the AddL/AddH/Data I/O (Port P0) pins. With the rising edge of STROBE the data is loaded to SFR OTEST. Now one dummy read cycle has to be started to check if programming is not blocked by the security settings (see Section 7.4.9). After this initialization the IC is ready for programming.

The dummy read cycle to initialize the security can be replaced by any other read action e.g. reading the security or signature bytes (see Section 7.4.8). Timing parameters are listed in Chapter 13.

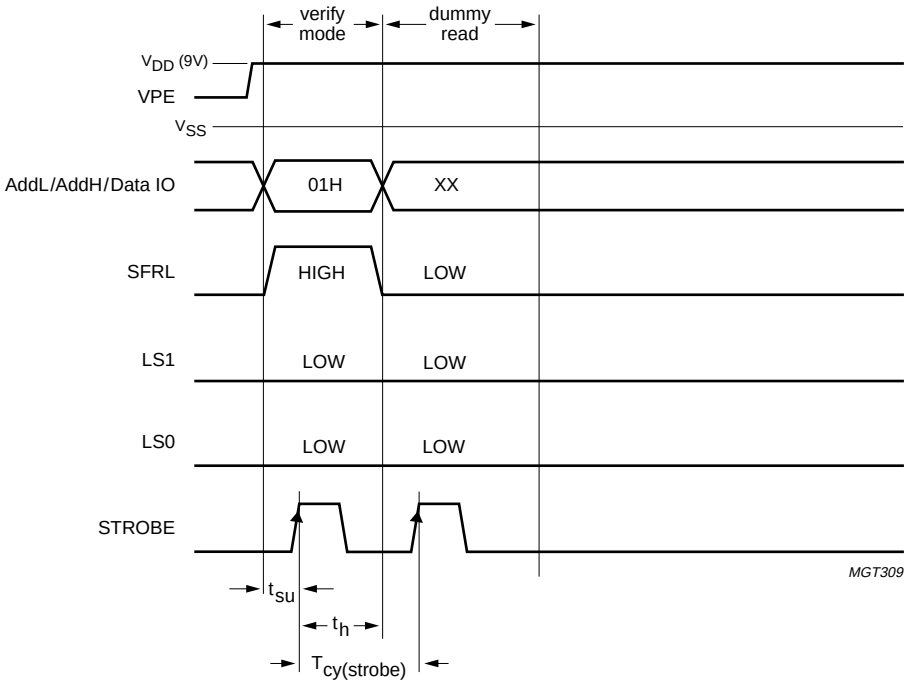


Fig.42 Parallel programming mode initialization.

7.4.4 SINGLE-BYTE PROGRAMMING

Figure 43 illustrates programming and verifying a single byte. The upper and lower address byte are loaded subsequently. With the control signals the proper latch is selected and the positive edge of the STROBE signal clocks in the data. The order of loading the latches is not important. The data to be programmed should be applied to the AddL/AddH/Data I/O pins before a programming pulse is started.

After programming a byte, this byte can be verified without reloading the address. With SFRL, LS1 and LS0 at LOW a read cycle is executed and the address is incremented (no increment with LS0 at HIGH). The next subsequent byte can therefore be programmed without loading its address. Loading a new address re-writes the counter. If the next data byte to be programmed has a different low address but the same high address, a reload of the low address only is sufficient.

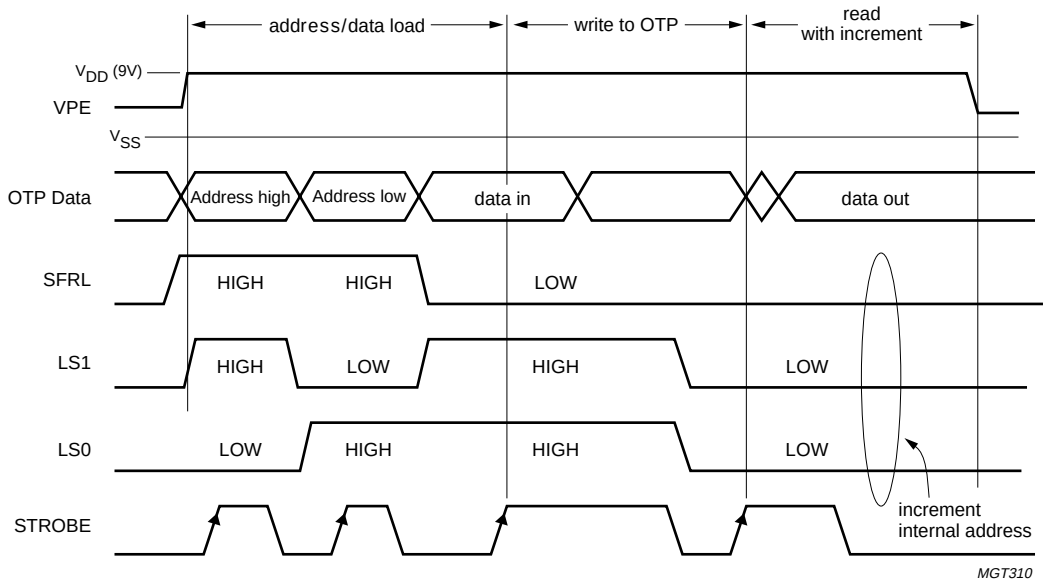


Fig.43 Single-byte programming mode.

7.4.5 MULTIPLE-BYTE PROGRAMMING

Multiple-byte programming mode allows to program four bytes at the same time. Due to the structure of the OTP matrix, the four bytes must have addresses with an increment of 16 (10H). The start address must be at ‘n × 40H + 0XH’ (n = 0 to 3); e.g address 00H, 10H, 20H and 30H can be programmed in parallel. The address AddH (only needed if it has to be changed) and AddL is loaded as usual. The first data byte is loaded, together with SFRL, LS1 and LS0 at HIGH, which selects the multiple-byte programming mode.

For the next bytes, first the lower address register has to be reloaded (address + 16) to select the next latch, then the data byte follows in multiple-byte programming mode. If all bytes are transferred, a write command finishes the cycle.

Remark: The data latches are loaded at a positive level of the STROBE signal and not at the edge. As a consequence the setup time is defined to the positive edge and the hold time to the negative edge. Multiple-byte programming is shown in Fig.44.

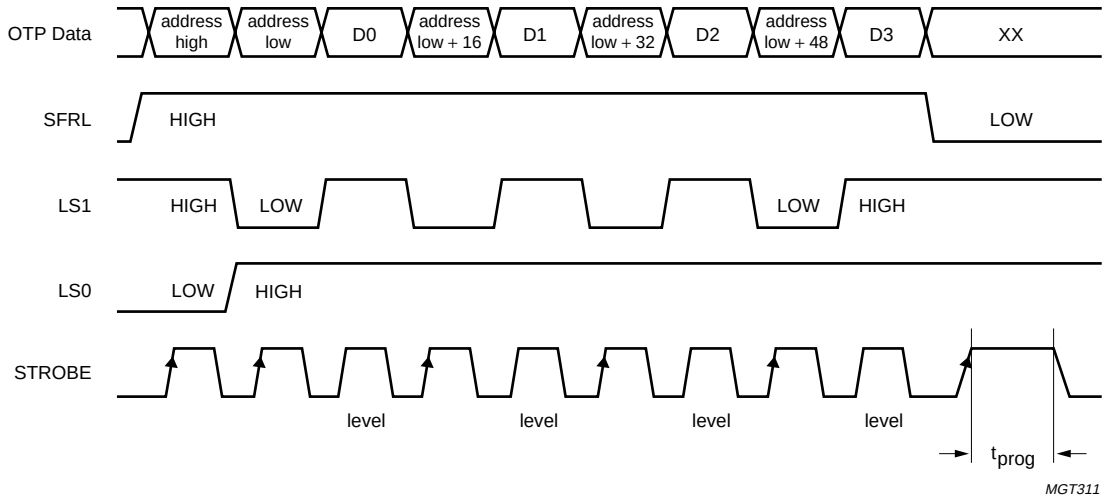


Fig.44 Multiple-byte programming mode.

7.4.6 FAST READ MODE

To speed up verify, read or blank check a fast read mode has been implemented. In this mode the addresses are generated by an internal address counter and only read pulses have to be applied.

The timing of this mode is shown in Fig.45. It is not necessary to load the address latches if the start address is already there. The fast read mode can be used in both verify and normal read mode.

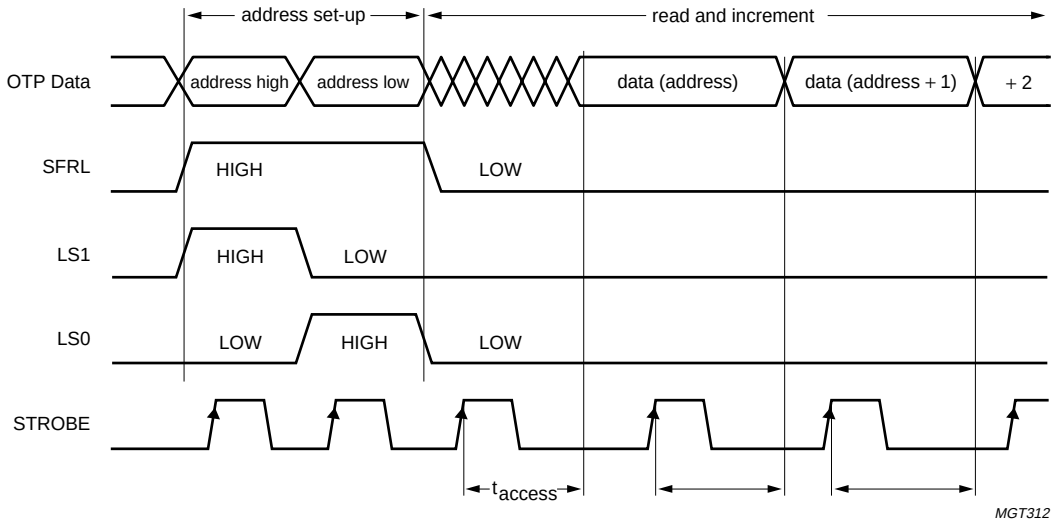


Fig.45 Fast read mode.

7.4.7 HIGH VOLTAGE TIMING

The external program voltage V_{PP} has to be at the specified high voltage level while a program pulse is applied. During verify it can be either at high voltage or equal to V_{DD}. V_{PP} has to be stable for at least 10 µs before a program pulse can be applied.

After applying a program pulse a recovery time of 1 µs is needed to discharge the internal high voltage nodes. During this recover time the memory cannot be accessed for verify.

Due to the above mentioned setup and hold timing for high voltage, the minimum programming duration is obtained when V_{PP} is continuously high during programming and also during verifying.

7.4.8 SIGNATURE BYTES

Three signature bytes are available to identify the device which are located at three addresses above 32 kbytes. The contents of the signature bytes are given in Table 107. They are implemented as read-only bytes and cannot be destroyed by unintentionally writing to them.

Table 107 Addresses and contents of the signature bytes

ADDRESS	CONTENTS	MEANING
8030H	15H	Philips SC
8031H	F1H	P87CL888
8060H	03H	

7.4.9 SECURITY

To prevent programming or reading of the OTP contents by third parties the security can be enabled. For the user these bytes can be accessed by applying address FFFFH before a program or verify cycle. To guarantee optimal security these bits are located outside the normal memory matrix and have input and output lines separated from the normal OTP I/Os. The security bits also have a separate read mechanism which makes them completely independent of the normal memory bits. Three bits facilitate three different levels of security by programming one or more bits. The bits are read in parallel to every read cycle and interpreted as described in Table 108.

After chip reset by default security level 1 is loaded in the OTP interface. One dummy read or verify cycle is needed to update the interface to the actual security level settings.

The security bits can be programmed like normal program bits.

The bits have to be supplied to the three least significant bits of OTP data and address FFFFH must be loaded. If address FFFFH is verified, the security bits are read on the three least significant bits of OTP data. After programming level 2 to the security bits only the security bits and the signature bytes can be verified. No access to the customer code via the OTP interface is possible any more. Verifying a user address while security level 2 or 3 has been programmed will result in reading ‘AAH’. Of course memory access through internal code fetching is possible at every level of security.

Since the contents of the security data is checked at every read or verify cycle, memory access through the OTP interface is disabled immediately after programming one of the security levels.

Note that programming security level 1 or higher disables not only Parallel programming, but also other programming modes (e.g. In-system programming).

Table 108 Definition of security levels

LEVEL	SECURITY DATA	DESCRIPTION
0	XXXX X111	no security, no restrictions
1	XXXX X110	programming disabled
2	XXXX X100	programming and reading disabled
3	XXXX X000	as level 2 and external memory access disabled; only present if external memory access is specified for this device

7.5 In-system programming

7.5.1 INTRODUCTION

In the In-system programming mode the OTP can be programmed under control of the CPU. A boot code to control programming of data has to be available in the OTP. In-system programming can be used when the IC is already mounted in the system for storing some tuning parameters or to program the complete customer code. Any interface to transfer the data from external into the microcontroller can be used to perform In-system programming. This section will give an example where the on-chip I^2C-bus interface is used for data transfer but also a software interface using some I/O pins or even measurements from the ADC can be used to bring in the data.

The complete address space is accessible in the In-system programming mode. The user should take care not to overwrite the boot code. The same timing and voltage parameters apply as for Parallel programming mode.

The OISYS SFR controls the In-system programming mode. OISYS.1 and OISYS.0 are readable and writeable by the CPU. OISYS.2 is a read-only bit. This bit is controlled by the external high voltage status signal VPON. The description of the OISYS SFR is given in Table 103. Other SFRs which are needed for In-system programming are listed in Tables 97, 98 and 99.

7.5.2 MODE ENTRY AND EXIT

The In-system programming mode can be entered in two ways. First by using the CPU which can set the InSysMode bit of the OISYS SFR. Second by applying high voltage to the V_{PP} pin during a reset pulse. The VPON bit (OISYS.2) will be set to logic 1 when the applied voltage is above the high voltage detection level. Note that this does not mean the voltage on V_{PP} is high enough for programming. Switch level for VPON bit is given in Chapter 13.

The In-system programming mode remains active as long as bit InSysMode = 1 (OISYS.0). An exit from this mode is possible by applying a new reset with the voltage on V_{PP} below the high-voltage detection level. Another way to exit the mode is to remove the high voltage and resetting InSysMode (OISYS.0).

7.5.3 EXAMPLE USING I^2C-BUS AND TIMER 2

A possible interface to transfer external data into the microcontroller is the I^2C-bus interface. The interface must be configured as a slave receiver waiting to be contacted by the master, in this case the programmer. Upon an S1 interrupt request the interrupt service routine reads the OISYS SFR to determine the high voltage status. When no high voltage is present the interrupt should be handled as a normal I^2C-bus interrupt caused by the application in normal mode.

When the interrupt service routine detects the presence of high voltage, the software must take care of transferring serial data from the I^2C-bus data register S1DAT to the OTP address and data latches. Special Function Registers are used to store the address and data: OAH and OAL for the high and low byte of the address and ODATA for the data to be programmed. The SFR descriptions are listed in Tables 97, 98 and 99. After loading address and data the program cycle must be configured by setting bit WE (OISYS.1). Setting WE only puts programming to sharp. Entering the Idle mode really starts the programming pulse.

To stop the programming cycle an internal or external interrupt must occur to cause a wake-up from Idle mode. When before starting a programming pulse a timer has been loaded, the overflow interrupt will terminate programming. In this way the value of the timer register determines the length of the programming pulse. After programming of one byte, bit WE must be cleared by software in order to prevent an unintended programming pulse by entering Idle mode.

Verify is done in a similar way as programming. To start a read cycle the circuit should be put into idle mode while bit WE = 0. Verify mode is automatically selected to simulate the worst case read situation. After a wake-up from idle the OTP output data can be read from the ODATA SFR. To be sure that the data in the ODATA SFR is really coming from the OTP memory it is advised to write ‘FFH’ into the ODATA SFR before a verify is started.

During a programming cycle no code can be executed from the OTP memory. Below an example of an assembler routine is listed of how a programming cycle could look like.

7.5.3.1 Example: programming cycle for In-system programming.

```
VPonTest_     IFNOT   OISYS.VPon, JMP NoINSYS_

Initial_      MOV     T2H,ValueH        ;
              MOV     T2L,ValueL        ; Load programming pulse length in T2
              ORL     IEN0,#40h         ; enable Timer 2 interrupt

NextByte_     MOV     OAL, AddL         ; Load address low
              MOV     OAH, AddH         ; Load address high
              MOV     ODATA,Data        ; Load data
              ORL     OISYS,#02h        ; Set bit WE
              ORL     T2CON,#04h        ; Start timer 2
              ORL     PCON,#01          ; set idle mode. Wake up from idle by T2
                                        ; interrupt will automatically stop programming
              ANL     OISYS,#FD         ; After wake-up from idle, clear WE bit
              INC     address
              IFNOT   finished, JMP NextByte_

NoInSYS_      JMP     ProgramStart_
```

7.5.4 SIGNATURE BYTES

The signature is mapped to the address space as already described in Section 7.4.8.

7.5.5 HOW TO CONNECT THE V_{PP} PAD TO ENABLE IN-SYSTEM PROGRAMMING

If the V_{PP} pin is dual-mode (e.g. V_{PP}/PORENABLE), whichever other IC is connected to the signal PORENABLE **must be able to withstand up to 9 V**, i.e. cannot have clamping diodes or low break-down voltages. If the pin is connected to a fixed voltage (V_{DD} or V_{SS}) there must be a way of switching-off this connection on the printed-circuit board (PCB). A possible implementation is presented in Fig.46.

In this example the POR is enabled in Normal mode of operation (PORENABLE = 1 by the pull-up), but the V_{PP} source must supply enough current in R_p in order to guarantee minimum 9.0 V on the V_{PP}/PORENABLE pin.

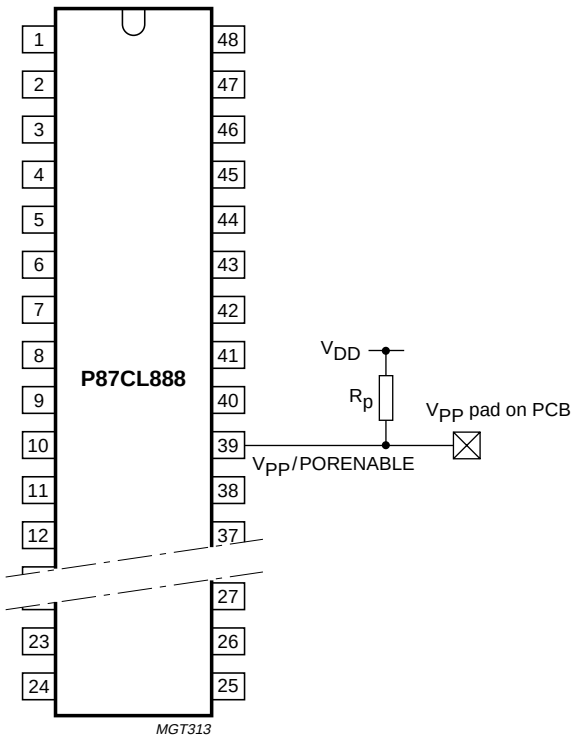


Fig.46 Example of V_{PP}/PORENABLE connection on a printed-circuit board.

8 INSTRUCTION SET

The asynchronous 80C51 family uses a powerful instruction set which permits the expansion of on-chip CPU peripherals and optimizes power consumption in idle and active modes as well as byte efficiency and execution speed. Typical execution times and energy consumption at room temperature (T_{amb} = 25 °C) and V_{DD} = 3.0 V are given in Table 109.

Remark: For most opcodes the numbers for execution speed and energy are also strongly dependent on the data (ADD, SUBB, DEC, INC, MUL, DIV, DA conditional jumps etc.) and the operand address (CPU internal SFRs or SFRs in a peripheral block).

Table 109 Instruction set

MNEMONIC		DESCRIPTION	BYTES	EXEC. TIME(1) (μs)	ENERGY(1) (nJ)	OPCODE (HEX)
Arithmetic operations						
ADD	A, Rn	add register to A	1	0.20	1.13	2*
ADD	A, direct	add direct byte to A	2	0.24	1.68	25
ADD	A, @Ri	add indirect RAM to A	1	0.21	1.36	26, 27
ADD	A, #data	add immediate data to A	2	0.23	1.40	24
ADDC	A, Rn	add register to A with carry flag	1	0.20	1.14	3*
ADDC	A, direct	add direct byte to A with carry flag	2	0.25	1.68	35
ADDC	A, @Ri	add indirect RAM to A with carry flag	1	0.21	1.37	36, 37
ADDC	A, #data	add immediate data to A with carry flag	2	0.23	1.45	34
SUBB	A, Rn	subtract register from A with borrow	1	0.20	1.13	9*
SUBB	A, direct	subtract direct byte from A with borrow	2	0.24	1.69	95
SUBB	A, @Ri	subtract indirect RAM from A with borrow	1	0.21	1.36	96, 97
SUBB	A, #data	subtract immediate data from A with borrow	2	0.23	1.43	94
INC	A	increment A	1	0.17	0.79	04
INC	Rn	increment register	1	0.18	1.16	0*
INC	direct	increment direct byte	2	0.22	1.75	05
INC	@Ri	increment indirect RAM	1	0.19	1.35	06, 07
DEC	A	decrement A	1	0.17	0.81	14
DEC	Rn	decrement register	1	0.18	1.17	1*
DEC	direct	decrement direct byte	2	0.22	1.75	15
DEC	@Ri	decrement indirect RAM	1	0.19	1.38	16, 17
INC	DPTR	increment data pointer	1	0.15	0.78	A3
MUL	AB	multiply A and B	1	0.15	0.70	A4
DIV	AB	divide A by B	1	0.73	3.58	84
DA	A	decimal adjust A	1	0.17	0.74	D4
Logic operations						
ANL	A, Rn	AND register to A	1	0.20	1.24	5*
ANL	A, direct	AND direct byte to A	2	0.30	1.91	55
ANL	A, @Ri	AND indirect RAM to A	1	0.21	1.44	56, 57
ANL	A, #data	AND immediate data to A	2	0.23	1.50	54

MNEMONIC		DESCRIPTION	BYTES	EXEC. TIME(1) (μs)	ENERGY(1) (nJ)	OPCODE (HEX)
ANL	direct, A	AND A to direct byte	2	0.26	1.96	52
ANL	direct, #data	AND immediate data to direct byte	3	0.28	2.41	53
ORL	A, Rn	OR register to A	1	0.29	1.71	4*
ORL	A, direct	OR direct byte to A	2	0.29	1.72	45
ORL	A, @Ri	OR indirect RAM to A	1	0.19	1.27	46, 47
ORL	A, #data	OR immediate data to A	2	0.21	1.23	44
ORL	direct, A	OR A to direct byte	2	0.24	1.78	42
ORL	direct, #data	OR immediate data to direct byte	3	0.27	2.16	43
XRL	A, Rn	exclusive-OR register to A	1	0.29	1.72	6*
XRL	A, direct	exclusive-OR direct byte to A	2	0.29	1.72	65
XRL	A, @Ri	exclusive-OR indirect RAM to A	1	0.19	1.31	66, 67
XRL	A, #data	exclusive-OR immediate data to A	2	0.21	1.33	64
XRL	direct, A	exclusive-OR A to direct byte	2	0.24	1.83	62
XRL	direct, #data	exclusive-OR immediate data to direct byte	3	0.27	2.27	63
CLR	A	clear A	1	0.14	0.71	E4
CPL	A	complement A	1	0.15	0.93	F4
RL	A	rotate A left	1	0.15	0.73	23
RLC	A	rotate A left through the carry flag	1	0.15	0.74	33
RR	A	rotate A right	1	0.17	0.82	03
RRC	A	rotate A right through the carry flag	1	0.15	0.73	13
SWAP	A	swap nibbles within A	1	0.14	0.71	C4
Data transfer						
MOV	A, Rn	move register to A	1	0.15	0.89	E*
MOV	A, direct	move direct byte to A	2	0.19	1.49	E5
MOV	A, @Ri	move indirect RAM to A	1	0.16	1.13	E6, E7
MOV	A, #data	move immediate data to A	2	0.21	1.85	74
MOV	Rn, A	move A to register	1	0.13	0.86	F*
MOV	Rn, direct	move direct byte to register	2	0.23	1.90	A*
MOV	Rn, #data	move immediate data to register	2	0.16	1.28	7*
MOV	direct, A	move A to direct byte	2	0.18	1.47	F5
MOV	direct, Rn	move register to direct byte	2	0.21	1.68	8*
MOV	direct, direct	move direct byte to direct byte	3	0.25	2.22	85
MOV	direct, @Ri	move indirect RAM to direct byte	2	0.22	1.92	86, 87
MOV	direct, #data	move immediate data to direct byte	3	0.21	1.85	75
MOV	@RI, A	move A to indirect RAM	1	0.14	1.01	F6, F7
MOV	@Ri, direct	move direct byte to indirect RAM	2	0.25	2.09	A6, A7
MOV	@Ri, #data	move immediate data to indirect RAM	3	0.11	0.92	76, 77
MOV	DPTR, #data 16	load data pointer with a 16-bit constant	3	0.20	1.58	90

MNEMONIC		DESCRIPTION	BYTES	EXEC. TIME(1) (μs)	ENERGY(1) (nJ)	OPCODE (HEX)
MOVC	A, @A + DPTR	move code byte relative to DPTR to A	1	0.31	2.34	93
MOVC	A, @A + PC	move code byte relative to PC to A	1	0.32	2.47	83
MOVX(2)	A, @Ri	move external RAM (8-bit address) to A	1	–	–	E2, E3
MOVX(2)	A, @DPTR	move external RAM (16-bit address) to A	1	–	–	E0
MOVX(2)	@Ri, A	move A to external RAM (8-bit address)	1	–	–	F2, F3
MOVX(2)	@DPTR,A	move A to external RAM (16-bit address)	1	–	–	F0
PUSH	direct	push direct byte onto stack	2	0.26	1.62	C0
POP	direct	pop direct byte from stack	2	0.26	1.66	D0
XCH	A, Rn	exchange register with A	1	0.20	1.35	C*
XCH	A, direct	exchange direct byte with A	2	0.25	1.98	C5
XCH	A, @Ri	exchange indirect RAM with A	1	0.21	1.42	C6, C7
XCHD	A, @Ri	exchange LOW-order nibble indirect RAM with A	1	0.19	1.38	D6, D7
Boolean variable manipulation						
CLR	C	clear carry flag	1	0.11	0.64	C3
CLR	bit	clear direct bit	2	0.24	1.51	C2
SETB	C	set carry flag	1	0.11	0.65	D3
SETB	bit	set direct bit	2	0.24	1.71	D2
CPL	C	complement carry flag	1	0.12	0.68	B3
CPL	bit	complement direct bit	2	0.23	1.59	B2
ANL	C, bit	AND direct bit to carry flag	2	0.21	1.30	82
ANL	C, /bit	AND complement of direct bit to carry flag	2	0.23	1.55	B0
ORL	C, bit	OR direct bit to carry flag	2	0.21	1.33	72
ORL	C, /bit	OR complement of direct bit to carry flag	2	0.23	1.54	A0
MOV	C, bit	move direct bit to carry flag	2	0.22	1.34	A2
MOV	bit, C	move carry flag to direct bit	2	0.24	1.52	92
Program and machine control						
ACALL	addr11	absolute subroutine call	2	0.40	2.64	•1 addr
LCALL	addr16	long subroutine call	3	0.45	3.09	12
RET		return from subroutine	1	0.20	1.03	22
RETI		return from interrupt	1	0.43	3.01	32
AJMP	addr11	absolute jump	2	0.29	1.76	♦1 addr
LJMP	addr16	long jump	3	0.32	2.14	02
SJMP	rel	short jump (relative address)	2	0.26	1.50	80
JMP	@A+DPTR	jump indirect relative to the DPTR	1	0.46	2.63	73
JZ	rel	jump if A is zero	2	0.29	1.62	60
JNZ	rel	jump if A is not zero	2	0.26	1.34	70
JC	rel	jump if carry flag is set	2	0.24	1.23	40
JNC	rel	jump if carry flag is not set	2	0.29	1.61	50

MNEMONIC		DESCRIPTION	BYTES	EXEC. TIME(1) (μs)	ENERGY(1) (nJ)	OPCODE (HEX)
JB	bit, rel	jump if direct bit is set	3	0.31	1.90	20
JNB	bit, rel	jump if direct bit is not set	3	0.36	2.29	30
JBC	bit, rel	jump if direct bit is set and clear bit	3	0.36	2.25	10
CJNE	A, direct, rel	compare direct to A and jump if not equal	3	0.34	2.27	B5
CJNE	A, #data, rel	compare immediate to A and jump if not equal	3	0.35	2.38	B4
CJNE	Rn, #data, rel	compare immediate to register and jump if not equal	3	0.35	2.59	B*
CJNE	Ri, #data, rel	compare immediate to indirect and jump if not equal	3	0.36	2.82	B6, B7
DJNZ	Rn, rel	decrement register and jump if not zero	2	0.33	2.29	D*
DJNZ	direct, rel	decrement direct and jump if not zero	3	0.39	2.89	D5
NOP		no operation	1	0.11	0.63	00

Notes

1. Verified on sampling base.
2. Only applicable if XRAM is present on-chip (no external access possible).

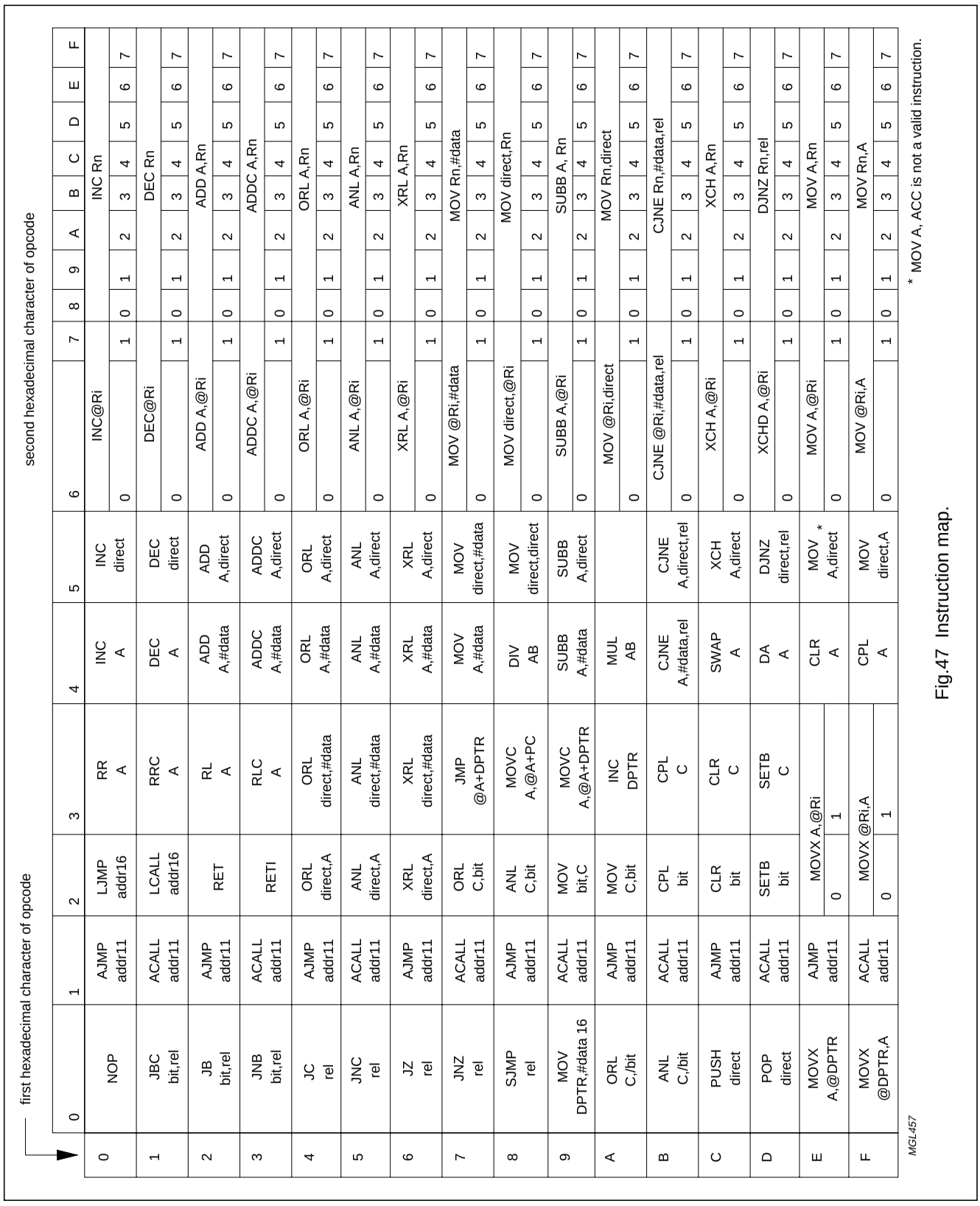
Table 110 Notation for data addressing modes

SYMBOL	DESCRIPTION
R_n	working registers R0 to R7
direct	128 internal RAM locations and any special function register (SFR)
R_i	indirect internal RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction
#data 16	16-bit constant included as bytes 2 and 3 of instruction
bit	direct addressed bit in internal RAM or SFR
addr16	16-bit destination address; used by LCALL and LJMP; the branch will be anywhere within the 64 kbytes program memory address space
addr11	11-bit destination address; used by ACALL and AJMP. The branch will be within the same 2-kbyte page of program memory as the first byte of the following instruction
rel	signed (two's complement) 8-bit offset byte; used by SJMP and all conditional jumps; range is −128 to +127 bytes relative to first byte of the following instruction

Table 111 Hexadecimal opcode cross-reference

SYMBOL	DESCRIPTION
*	8, 9, A, B, C, D, E and F
•	11, 31, 51, 71, 91, B1, D1 and F1
♦	01, 21, 41, 61, 81, A1, C1 and E1

8.1 Instruction map



first hexadecimal character of opcode ↓	second hexadecimal character of opcode →															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	AJMP addr11	LJMP addr16	RR A	INC A	INC direct	INC@Ri		INC Rn							
1	JBC bit,rel	ACALL addr11	LCALL addr16	RRC A	DEC A	DEC direct	DEC@Ri		DEC Rn							
2	JB bit,rel	AJMP addr11	RET	RL A	ADD A,#data	ADD A,direct	ADD A,@Ri		ADD A,Rn							
3	JNB bit,rel	ACALL addr11	RETI	RLC A	ADDC A,#data	ADDC A,direct	ADDC A,@Ri		ADDC A,Rn							
4	JC rel	AJMP addr11	ORL direct,A	ORL direct,#data	ORL A,#data	ORL A,direct	ORL A,@Ri		ORL A,Rn							
5	JNC rel	ACALL addr11	ANL direct,A	ANL direct,#data	ANL A,#data	ANL A,direct	ANL A,@Ri		ANL A,Rn							
6	JZ rel	AJMP addr11	XRL direct,A	XRL direct,#data	XRL A,#data	XRL A,direct	XRL A,@Ri		XRL A,Rn							
7	JNZ rel	ACALL addr11	ORL C,bit	JMP @A+DPTR	MOV A,#data	MOV direct,#data	MOV @Ri,#data		MOV Rn,#data							
8	SJMP rel	AJMP addr11	ANL C,bit	MOVC A,@A+PC	DIV AB	MOV direct,direct	MOV direct,@Ri		MOV direct,Rn							
9	MOV DPTR,#data 16	ACALL addr11	MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data	SUBB A,direct	SUBB A,@Ri		SUBB A, Rn							
A	ORL C,/bit	AJMP addr11	MOV C,bit	INC DPTR	MUL AB		MOV @Ri,direct		MOV Rn,direct							
B	ANL C,/bit	ACALL addr11	CPL bit	CPL C	CJNE A,#data,rel	CJNE A,direct,rel	CJNE @Ri,#data,rel		CJNE Rn,#data,rel							
C	PUSH direct	AJMP addr11	CLR bit	CLR C	SWAP A	XCH A,direct	XCH A,@Ri		XCH A,Rn							
D	POP direct	ACALL addr11	SETB bit	SETB C	DA A	DJNZ direct,rel	XCHD A,@Ri		DJNZ Rn,rel							
E	MOVX A,@DPTR	AJMP addr11	MOVX A,@Ri		CLR A	MOV * A,direct	MOV A,@Ri		MOV A,Rn							
F	MOVX @DPTR,A	ACALL addr11	MOVX @Ri,A		CPL A	MOV direct,A	MOV @Ri,A		MOV Rn,A							

* MOV A, ACC is not a valid instruction.

MGL457

Fig.47 Instruction map.

9 APPLICATION INFORMATION

9.1 Introduction

This chapter presents some information about how to use the P87CL888 in an application. It is not intended to replace the application notes but serves as a quick help when starting to work with the Philips Ultra Low Power handshake microcontrollers. There are some important improvements between the silicon in plastic packages and the Metalink EH emulator system which are described here. Furthermore, some hints on software development and power consumption are given to help the user take advantage of the full benefits of the handshake CPU.

9.2 Differences between P87CL888 and the Metalink EH emulation system

- The SYSCON SFR does not exist on the emulator system.
- On the emulator the oscillator can only be used in normal mode which is the default start-up mode of the P87CL888. The hysteresis input comparator does not exist.
- The clock source of the Timer 0 and Timer 1 is always f_{psc} on the emulator system. The timers can be used as counters, counting from external pin T1 or T0 but this is not possible in Power-down mode.
- The interrupts T0, T1 and LVD can cause on the emulator only a wake-up from idle and not from power-down.
- Prescaler bits PRESC[7:5] are not available on the emulator; therefore the synchronous mode and clock out functionality is not present.
- MSK polarity cannot be inverted on the emulator.
- $\overline{INT1}$ interrupt is on the emulator version present on P3.3 where it is mapped on P3.1 on the P87CL888.
- The clock output on P1.4 does not exist on the emulator.

9.3 The asynchronous handshake CPU

As the CPU of the P87CL888 is built in asynchronous technology (hand-shake mechanism) some properties are singular to it in comparison to standard synchronous 80C51 controllers:

- The CPU it self does not need a clock for code execution
- The performance (MIPs) is not dependent on oscillator frequency but strongly related to V_{DD}, temperature, silicon parameters and type of software. It always runs at the maximum speed determined by the external influences above. Therefore, it operates also with the maximum power consumption in the minimum time. Generally the lower the temperature and the higher the V_{DD} the faster the CPU runs. Details on instruction speed and energy consumption per instruction can be found in Chapter 8.

Because of the above mentioned properties some hints are given for using this controller in any kind of application in an efficient way.

- Due to the high CPU performance, independent of clock frequency, certain functions (e.g. serial or customized interfaces) can be built in software in a very efficient and flexible way
- In classic 80C51 software the user (software engineer) was able to rely on cycle-timing for wait-loops, synchronisation in the system or similar usage (e.g. NOP instruction for waiting one machine cycle). When using the asynchronous CPU wait-loops should be implemented by starting a timer and putting the CPU in Idle mode in order to wait for an interrupt. Significant power reduction and a much more robust software will be obtained. If in an application the instruction counter is needed Timer 0 or Timer 1 can be used with the instruction request signal connected to the clock source input
- One should avoid using ‘wait-until’-loops (SFR polling). This would lead to maximum CPU-load resulting in very high current consumption. The CPU should always be used as an event driven machine waiting for interrupts. After an activity the device must be entered in Idle or Power-down mode as fast as possible, the current is then reduced down to leakage. The device provides flexible means (interrupts, timer, counters) for a recovery from these power reduction modes.

9.4 Telecom applications

Besides the special telephony applications as DTMF and MSK, the controller can also be used to perform the following functions:

- Ring level detect; 10 up to 90 V (RMS) superimposed on 48 up to 72 V (DC voltage); level reject range adjustable between 10 to 50 V (RMS)
- Ring frequency detect: 17 to 66 Hz; reject frequencies <14 and >70 Hz; adjustable between 14 to 70 Hz
- Line in use, set 1 (on-hook) indicates that set 2 is off-hook; this means:
  - $V_{A/B}$ < 16 V: line is in use
  - $V_{A/B}$ > 20 V: line is idle

  this works only in mains supplied systems, or set 1 (off-hook) indicates its own off-hook mode; this last function works also in line powered sets
- Hold release: set 1 is off-hook. Hold button: set 2 goes off-hook; set 1 disconnects
- Parallel set detect; set 1 (for instance an answering machine) is off-hook; set 2 goes off-hook too; set 1 detects this, stops and goes on-hook
- Busy tone detection
- Battery check.

The values of these functions are adjustable by external resistors, so it is easy to adjust to the local requirements.

10 HOW TO ESTIMATE P87CL888 POWER CONSUMPTION

10.1 General

Due to the use of the Philips unique asynchronous technology within the CPU, the power estimation must be done by taking into account several circumstances. To have an accurate power estimation the application must be well known. This especially means that all (or the most significant) application modes (e.g. idle or operation modes) are known and their weight or contribution – what is done when with which occurrence – can be estimated precise enough.

10.2 Modes

10.2.1 Power-down mode

In Power-down mode only the POR circuitry (if enabled; PORENABLE = 1) is drawing current. The CPU, the oscillator, the clock tree and the peripheral functions are switched off except Timer 0 and Timer 1 which can function as counter in Power-down mode. The device can be woken-up by an interrupt.

In this mode the power consumption is dependent on the POR circuitry, outside activity (port toggling, gate-current) and leakage (see Fig.48).

10.2.2 Idle mode

In Idle mode the oscillator (if enabled), the POR circuitry (if enabled; PORENABLE = 1), the clock tree and the enabled peripheral functions are running. The peripheral functions are fixed to the peripheral clock (f_{per} or f_{psc}). In Figs 54, 54 and 58 one can see the behaviour of the idle current with no peripheral functions switched on.

10.2.3 Operating mode

In operating mode the CPU, the oscillator, the POR circuitry (if enabled; PORENABLE = 1), the clock tree and the enabled peripheral functions are running. While the peripheral functions are fixed to the peripheral clock (f_{per} or f_{psc}) the CPU is completely free running. In plain words: it does one instruction after the other without any clocking nor timing scheme. In addition to that and to make code execution faster following instruction is pre-fetched while an instruction is being executed.

10.2.4 CPU only mode

The CPU itself does not need a clock to function so this mode switches off the oscillator. The user must ensure that before starting a peripheral function the oscillator is restarted early enough to be stable. This mode is not implemented on the P87CL888.

10.3 Examples of power consumption estimation

A rough estimation of device power consumption can be made by an add-up of the Power-down mode current, Idle mode current, enabled peripheral function(s) current and estimated CPU processing load times mean value of operating current: $I_{DD(pd)} + I_{DD(id)} + I_{periphery} + \text{CPU load} \times I_{DD(op)}$.

Assume an application part where the device is 50% in idle, during idle for total 40% a peripheral function is running, for 20% the CPU is active and the rest is power-down state. Then the averaged power consumption can roughly be calculated as follows:

$I_{average} = (100\% \times I_{DD(pd)}) + (50\% \times I_{DD(id)}) + (40\% \times I_{periphery}) + (20\% \times I_{DD(op)})$.

When the number of instructions within an application part, its duration and the supply voltage (CPU performance in Mips) is known then the CPU processing load can be estimated as follows:

$$\text{CPU processing load} = 100 \times \frac{\text{instruction number} / \text{duration (seconds)}}{\text{CPU performance (Mips)}}$$

10.3.1 Example of ring-detection

Typically a ring-detection is done by a time measurement using a timer. The device is activated by an external interrupt (e.g. AC on the A/B line), the oscillator runs up and the device does the time measurement with a timer.

In this example one can assume that to read the timer contents, decide whether an external ringer should be enabled or not, to restart the timer and then go to Idle mode are around 30 instruction. Assume a ringing frequency of about 25 Hz and a device supply voltage of 3.0 V which gives us a CPU performance of about 4.8 Mips.

The CPU processing load is then 100×30 instructions $\times$ 25 Hz/4800000 $\approx$ 0.016%.

From the calculation above we can say that the idle current will be dominant in this application part.

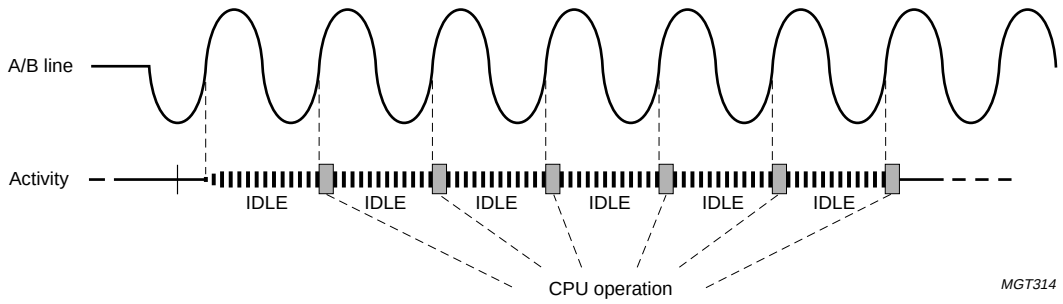


Fig.48 Ring-detection sequence.

10.3.2 Example: DTMF tone generation

With the asynchronous CPU it is possible to generate DTMF dual tones in software. Evaluations have shown, that the produced tones are CEPT compliant with a CPU performance need of about 0.5 Mips. Additional 10 dB in S/N ratio can be obtained with a CPU performance of about 1 Mips. The overall mode is idle since timers are running.
The averaged power consumption at 3.0 V would then be:

$$I_{averaged} = 0.5 \text{ Mips (or 1.0 Mips)} \times \frac{I_{op}(\text{at 3.0 V})}{4.8 \text{ Mips}} + I_{DD(id)}(\text{at 3.0 V})$$

11 LIMITING VALUES

According to the Absolute Maximum Ratings System (IEC 60134); note 1.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{DD}	supply voltage		–0.5	+4.6	V
V_I	input voltage	note 2	–0.5	V_{DD} + 0.5	V
$I_{I/O}$	maximum sink/source current for each input/output pin		–	±10	mA
I_{DD}	maximum supply current for any supply pin		–	50	mA
V_{es}	electrostatic handling voltage	human body model; note 3	–	700	V
		machine model; note 4	–	125	V
P_{tot}	total power dissipation		–	100	mW
T_{stg}	storage temperature		–55	+125	°C
T_{amb}	operating ambient temperature (for all devices)		–25	+70	°C

Notes

1. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise specified.
2. May not exceed the limiting value for V_{DD}.
3. According to SNW-FQ-302A: C = 100 pF; R = 1.5 kW.
4. According to SNW-FQ-302B: C = 200 pF; L = 0.75 mH.

12 CHARACTERISTICS

V_{DD} = 1.8 to 3.6 V; V_{SS} = 0 V; f_{xtal} = 4 MHz; T_{amb} = –25 to +70 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.(1)	UNIT
Supply						
V_{DD}	supply voltage					
	operating		1.8(2)	–	3.6	V
	RAM data retention		1.0	–	3.6	V
I_{DD}	operating supply current	V_{DD} = 3 V; T_{amb} = 25 °C; at 100% CPU load				
		note 3	–	–	3.5	mA
		notes 4 and 5	–	2.25	–	mA
$I_{DD(id)}$	supply current Idle mode	V_{DD} = 3 V; external clock; note 6	–	60	75	µA
		V_{DD} = 3 V; T_{amb} = 25 °C; crystal connected; note 5	–	230	–	µA
$I_{DD(pd)}$	supply current Power-down mode	V_{DD} = 3 V; T_{amb} = 25 °C; note 7				
		POR enabled	–	1	–	µA
		POR disabled	–	0.1	–	µA
		V_{DD} = 3 V; T_{amb} = 70 °C; note 7				
		POR enabled	–	–	5.0	µA
		POR disabled	–	–	4.5	µA
$I_{DD(block)}$	supply current per block:	V_{DD} = 3 V; T_{amb} = 25 °C; note 8				
	MSK modem		–	14	–	µA
	Watchdog Timer		–	2	–	µA
	I^2C-bus		–	30	–	µA
	Timer 2		–	4	–	µA
	Timer 0 or Timer 1		–	10	–	µA
	DAC	including output buffer	–	240	–	µA
	ADC	including DAC and comparator	–	70	–	µA
Performance						
f_{XTAL1}	external clock input frequency	notes 5 and 9	DC	–	12	MHz
CPU_{perf}	CPU performance	T_{amb} = 25 °C; notes 4 and 5				
		V_{DD} = 1.8 V	–	2.8	–	Mips
		V_{DD} = 3.0 V	–	4.8	–	Mips
CPU_{eff}	CPU efficiency	T_{amb} = 25 °C; notes 4 and 5				
		V_{DD} = 1.8 V	–	2200	–	Mips/W
		V_{DD} = 3.0 V	–	740	–	Mips/W

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.(1)	UNIT
Inputs (Ports, MIN, $\overline{RST}$, PORENABLE)						
V_{IL}	LOW-level input voltage	notes 10 and 11	0	–	$0.2V_{DD}$	V
V_{IH}	HIGH-level input voltage	note 10	$0.8V_{DD}$	–	V_{DD}	V
$\lvert I_{IL} \rvert$	LOW-level input current; ports in standard port configuration	V_{IN} = 0.4 V; note 12; see Fig.59	–	20	50	µA
$\lvert I_{IL(T)} \rvert$	LOW-level input current; HIGH-to-LOW transition; ports in standard port configuration	$V_{IN} = 0.5V_{DD}$; note 12; see Fig.59	–	200	1000	µA
$\lvert I_{LI} \rvert$	input leakage current; ports in open-drain or high-impedance input configuration	$V_{SS} \leq V_I \leq V_{DD}$;	–	–	100	nA
Outputs (Ports, $\overline{RST}$)						
I_{OL}	LOW-level output current; except SDA and SCL; note 13	V_{DD} = 3.0 V; V_{OL} = 0.4 V	4	–	–	mA
		V_{DD} = 3.0 V; V_{OL} = 1.5 V	–	10	–	mA
I_{OH}	HIGH -level output current; push-pull configuration only	V_{DD} = 3 V; $V_{OH} = V_{DD} - 0.4$ V;	4	–	–	mA
		V_{DD} = 3 V; $V_{OH} = V_{DD} - 1.5$ V	–	10	–	mA
$I_{\overline{RST}}$	$\overline{RST}$ pull-up resistor current	V_{DD} = 3 V; $V_{OH} = V_{DD} - 0.4$ V	0.05	0.2	–	µA
		V_{DD} = 3 V; $V_{OH} = V_{SS}$	–	0.6	2.5	µA
Power-on reset						
$V_{POR(H)}$	POR HIGH level	note 14	2.0	2.4	2.7	V
$V_{POR(L)}$	POR Low level	note 14	1.6	2.0	2.4	V
V_{hys}	POR hysteresis	note 14	150	400	800	mV
$t_{dip(min)}$	minimum time of dip in V_{DD} voltage that will be detected	T_{amb} = 25 °C	–	50	–	µs
Amplitude Controlled Oscillator (ACO)						
f_{osc}	oscillator frequency	notes 5 and 9	1	–	12	MHz
R_{fb}	feedback resistance	note 5	–	125	–	kΩ
g_m	transconductance	T_{amb} = 25 °C; V_{DD} = 1.8 V	1.5	–	2.5	mS
		T_{amb} = 25 °C; V_{DD} = 3 V	3.5	–	6	mS
$C_{i(L)(XTAL1)}$	capacitive input load on XTAL1		–	500	1000	fF
$V_{XTAL1(p-p)}$	external clock signal amplitude on pin XTAL1 (peak-to-peak value)	in oscillator mode	$0.4V_{DD}$	–	V_{DD}	V
$V_{DC(XTAL1)}$	mean value of external clock signal	in oscillator mode	–	$0.5V_{DD}$	–	V
$V_{IL(XTAL1)}$	LOW-level input voltage pin XTAL1	in external clock mode	0	–	$0.2V_{DD}$	V
$V_{IH(XTAL1)}$	HIGH-level input voltage pin XTAL1	in external clock mode	$0.6V_{DD}$	–	V_{DD}	V

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.(1)	UNIT
C_{1e}, C_{2e}	external required load capacitance on XTAL1 and XTAL2		–	22	–	pF
Analog-to-Digital Converter (ADC)						
$V_{DD(ADC)}$	operating voltage ADC	note 15	2.5	–	3.6	V
RES_{ADC}	ADC resolution		–	8	–	bit
$E_{offset(comp)}$	comparator offset error		–	–	15	mV
$GE_{offset(ADC)}$	ADC gain and offset error		–	–	5%	–
C_i	input capacitance		–	–	1.5	pF
R_i	input resistance		40	–	–	MΩ
Digital-to-Analog Converter (DAC)						
$V_{DD(DAC)}$	operating voltage DAC	note 15	2.5	–	3.6	V
DNL	differential non-linearity	notes 17 and 18				
	normal DAC values	LVLSH = 0	–	0.2	0.5	LSB
	shifted DAC values	LVLSH = 1	–	0.2	0.5	LSB
INL	integral non-linearity	notes 17 and 18				
	normal DAC values	LVLSH = 0	–	–	1	LSB
	shifted DAC values	LVLSH = 1	–	–	1	LSB
$V_{O(DAC)}$	DAC output voltage					
	normal DAC values (LVLSH = 0)	code: 0000 0000	–	0.0	0.005	V
		code: 1111 1111	1.10	1.20	1.30	V
	shifted DAC values (LVLSH = 1)	code: 0000 0000	1.15	0.20	0.25	V
		code: 1111 1111	1.10	1.20	1.30	V
$t_{st(DAC)}$	DAC settling time		–	–	1	µs
DAC output buffer						
$C_{o(L)}$	capacitive output load	note 19	–	–	20	pF
B_{buf}	band width	–3 dB	–	–	–	MHz
Voltage monitor						
VM_{err}	inaccuracy of voltage monitor		–	6.5	–	%
Z_{VM}	impedance when voltage monitor is selected		–	120	–	kΩ

Notes

1. The measurement of the maximum value is done with all output pins disconnected; $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; $\overline{RST} = V_{DD}$; XTAL1 driven with square wave; XTAL2 not connected; all derivative blocks disabled. To see the typical value of each instruction please consult Table 109 “Instruction set”.
2. The minimum operating voltage is the level where V_{DD} is higher than the power-on reset level.
3. For this measurement an instruction was selected which current consumption is around the typical value; the instruction is: LJMP to ADDR + 03H.

4. The typical operating supply current is evaluated as a mean value over all possible instructions (100% CPU load) and with a crystal connected.
5. Verified on sampling basis.
6. The Idle mode supply current is measured with all output pins and $\overline{RST}$ disconnected; $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; XTAL1 driven with square wave; XTAL2 not connected; all derivative blocks disabled.
7. The Power-down mode supply current is measured with all output pins and $\overline{RST}$ disconnected; $V_{IL} = V_{SS}$; $V_{IH} = V_{DD}$; XTAL1 and XTAL2 not connected;
8. The typical currents are only for the specific block. To calculate the typical power consumption of the microcontroller, the current consumption of the CPU weighted with the processing must be added.
   Example: the typical average current consumption of the microcontroller in operating mode with 10% CPU processing load, Watchdog Timer and MSK active can be calculated as: $10\% \times I_{CPU} + I_{DD(id)} + I_{WD} + I_{MSK}$.
9. For some peripheral blocks it could be required to reduce the internal clock frequency with the PSC2 and an additional divider inside the peripherals. Symbol 'f_{XTAL1}' is meant for external device clocking and 'f_{osc}' is meant as on-chip oscillator frequency.
10. The input threshold voltage of P1.6/SCL and P1.7/SDA meet the I^2C-bus specification. Therefore, an input voltage below $0.3V_{DD}$ will be recognized as a logic 0 and an input voltage above $0.7V_{DD}$ will be recognized as a logic 1.
11. For pin PORENABLE the $V_{IL(max)} = 0.1V_{DD}$.
12. Not valid for pins SDA, SCL, RST, MIN and PORENABLE.
13. Due to the maximum allowed current, the number of output pins switching at the same time should be limited to one.
14. $V_{POR(H)}$ and $V_{POR(L)}$ are related to each other. The V_{hys} is defined as $V_{POR(H)} - V_{POR(L)}$ and is granted positive by design.
15. The ADC will run at a lower operating voltage with decreased performance. For V_{DD} = 2.2 to 2.5 V the output of the DAC at full scale in worst case is 95% of the normal full scale output.
16. Conversion time in automatic mode is the time needed for the DAC to generate a stable output and the time needed for the comparator to give a stable output times the number of bits (8). In software mode the conversion speed is dependent on the speed of the CPU.
17. One LSB = $V_{ref(H)}/2^N$.
18. The DNL is specified from 5% to 95% of full range.
19. Maximum capacitive load on DAC output is specified with a typical resistive load of 10 kΩ (AC decoupled).

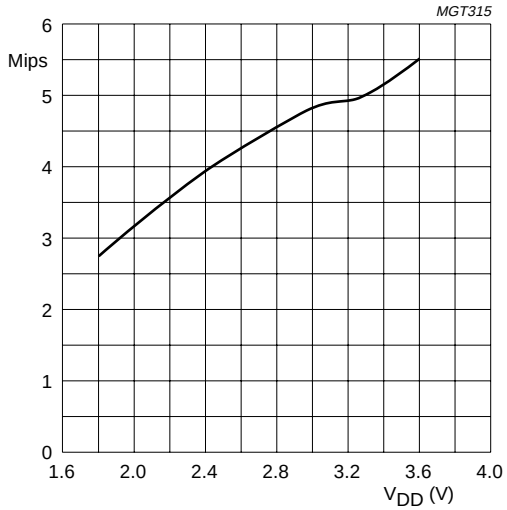


Fig.49 Typical CPU performance as a function of V_{DD}, T_{amb} = 25 °C (mean value over all instructions).

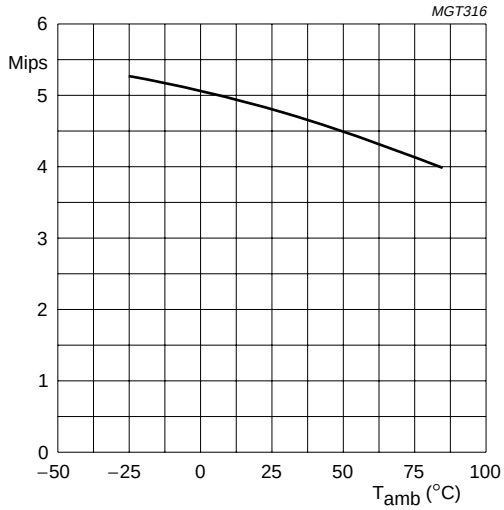


Fig.50 Typical CPU performance as a function of T_{amb}, V_{DD} = 3 V (mean value over all instructions).

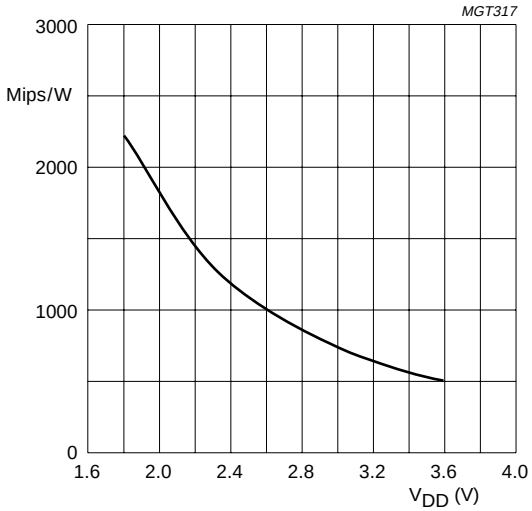


Fig.51 Typical CPU efficiency as a function of V_{DD}, T_{amb} = 25 °C (mean value over all instructions).

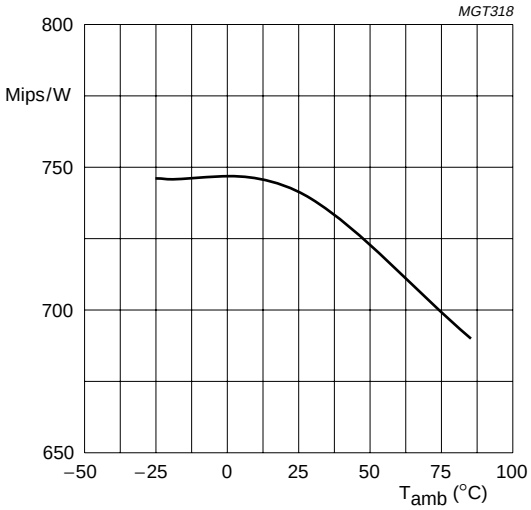


Fig.52 Typical CPU efficiency as a function of T_{amb}, V_{DD} = 3 V (mean value over all instructions).

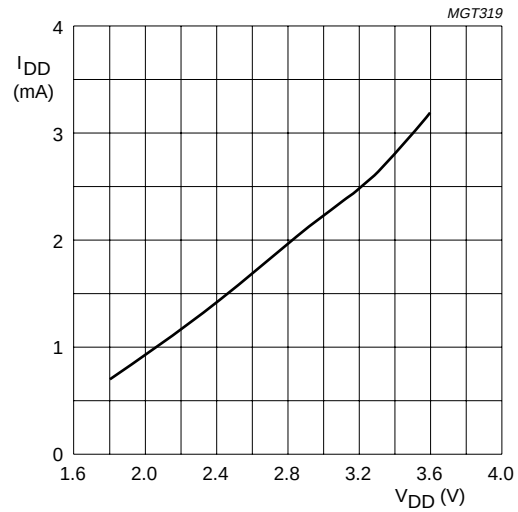


Fig.53 Typical operating current as a function of V_{DD}, T_{amb} = 25 °C; 100% CPU load (mean value over all instructions).

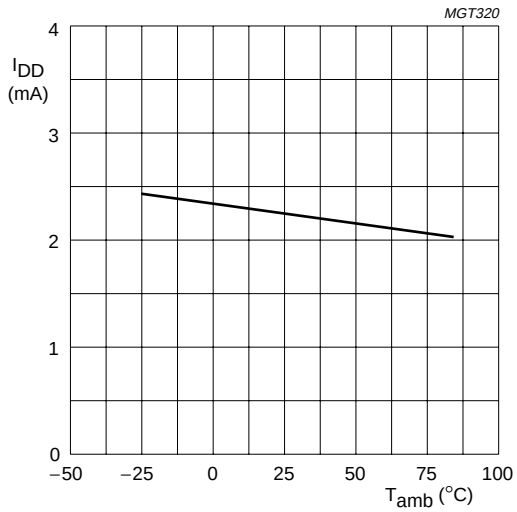


Fig.54 Typical operating current as a function of T_{amb}, V_{DD} = 3.0 V; 100% CPU load (mean value over all instructions).

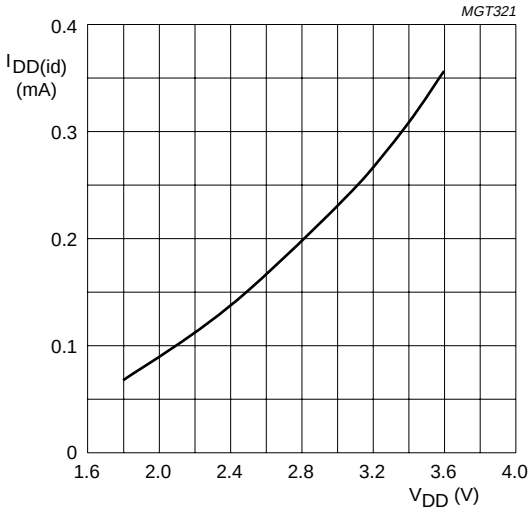


Fig.55 Typical Idle current as a function of V_{DD}, T_{amb} = 25 °C; f_{osc} = 4 MHz (crystal).

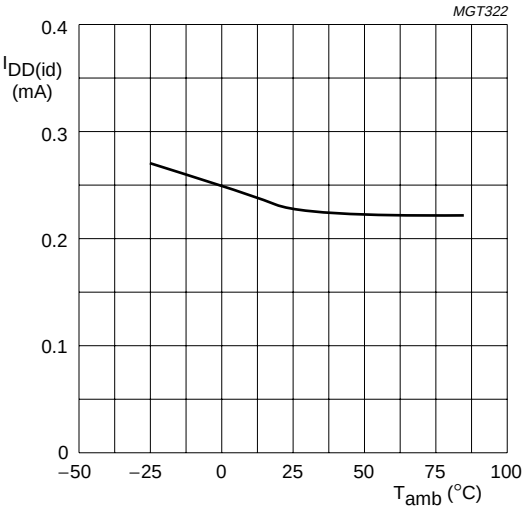


Fig.56 Typical Idle current as a function of T_{amb}, V_{DD} = 3.0 V; f_{osc} = 4 MHz (crystal).

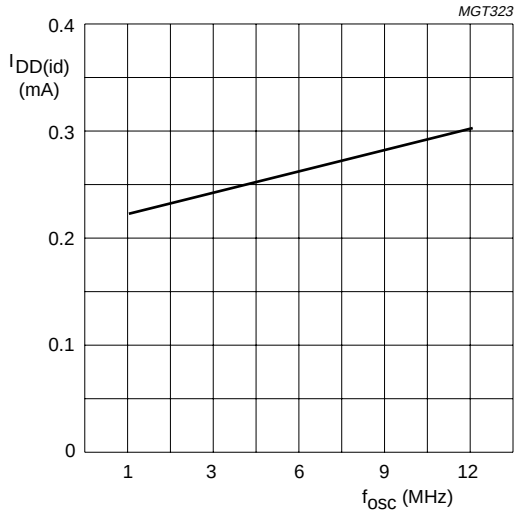
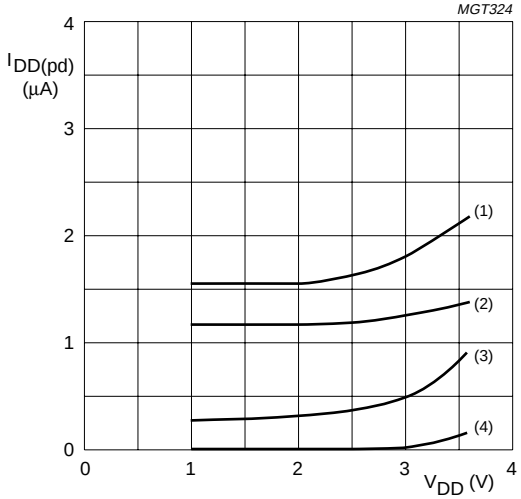


Fig.57 Typical Idle current as a function of oscillator (crystal) frequency, V_{DD} = 3.0 V; T_{amb} = 25 °C.



(1) POR and LVD enabled (T_{amb} = 70 °C).
(2) POR and LVD enabled (T_{amb} = 25 °C).
(3) POR and LVD disabled (T_{amb} = 70 °C).
(4) POR and LVD disabled (T_{amb} = 25 °C).

Fig.58 Typical Power-down current as a function of V_{DD}.

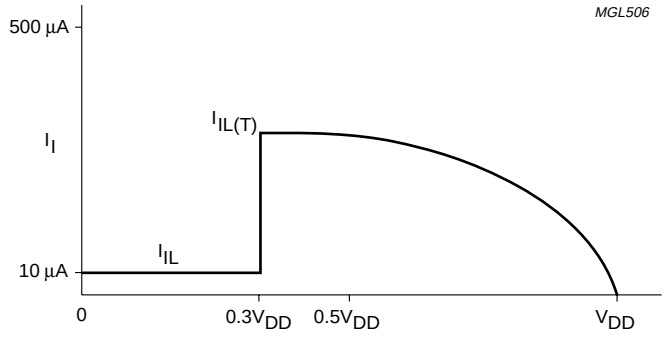


Fig.59 Port input current.

13 ELECTRICAL SPECIFICATION FOR DEVICE PROGRAMMING

Only values which differ from standard values are listed here.

Table 112 DC characteristics

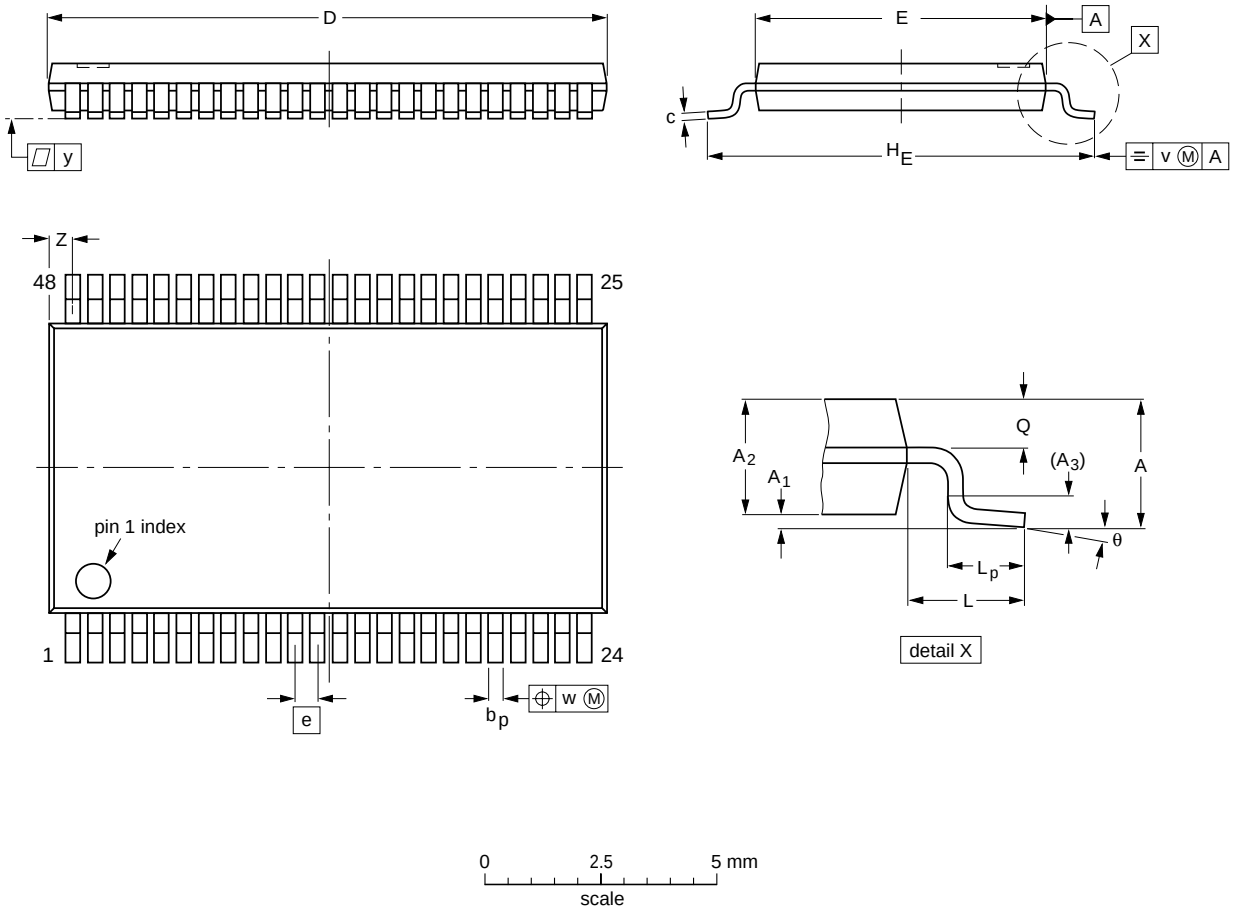
SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	supply voltage		3.0	–	3.6	V
V_{PP}	program voltage		8.75	–	9.25	V
I_{DD}	supply current operating		–	–	50	mA
I_{PP}	program supply current					
	single byte programming mode		–	–	20	mA
	multiple byte programming mode		–	–	50	mA
	mass programming mode		–	–	150	mA
	In-system programming mode		–	–	40	mA
Temperature						
T_{amb}	ambient temperature		21	–	27	°C

Table 113 AC characteristics

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
Parallel programming mode					
t_{su}	address, data, mode setup time	20	–	–	ns
t_h	address, data, mode hold time	20	–	–	ns
$t_{W(R)(strobe)}$	read STROBE pulse width	10	–	–	ns
t_{ACC}	OTP access time (verify mode)	–	–	500	ns
f_{clk}	mode entry clock frequency	–	–	2	MHz
f_{strobe}	STROBE signal clock frequency	–	–	2	MHz
$t_{su(VPE)}$	VPE setup time	10	–	–	µs
$t_{W(prog)}$	program pulse width	90	100	110	µs
$t_{rec(prog)}$	program pulse recover time	1	–	–	µs
$t_{appl(VPP)}$	V_{PP} applied time (over whole product lifetime)	–	–	100	s

14 PACKAGE OUTLINE

TSSOP48: plastic thin shrink small outline package; 48 leads; body width 6.1 mm **SOT362-1**



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	θ
mm	1.2	0.15 0.05	1.05 0.85	0.25	0.28 0.17	**0.2** **0.1**	12.6 12.4	6.2 6.0	0.5	8.3 7.9	1	0.8 0.4	0.50 0.35	0.25	0.08	0.1	0.8 0.4	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	EIAJ				
SOT362-1		MO-153					~~95-02-10~~ 99-12-27

15 SOLDERING

15.1 Introduction to soldering surface mount packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"Data Handbook IC26; Integrated Circuit Packages"* (document order number 9398 652 90011).

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

15.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several methods exist for reflowing; for example, convection or convection/infrared heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferable be kept below 220 °C for thick/large packages, and below 235 °C for small/thin packages.

15.3 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
  - smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

  The footprint must incorporate solder thieves at the downstream end.
- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time is 4 seconds at 250 °C.
A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

15.4 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

15.5 Suitability of surface mount IC packages for wave and reflow soldering methods

PACKAGE	SOLDERING METHOD	
	WAVE	REFLOW(1)
BGA, HBGA, LFBGA, SQFP, TFBGA	not suitable	suitable
HBCC, HLQFP, HSQFP, HSOP, HTQFP, HTSSOP, HVQFN, SMS	not suitable(2)	suitable
PLCC(3), SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended(3)(4)	suitable
SSOP, TSSOP, VSO	not recommended(5)	suitable

Notes

1. All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *"Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods"*.
2. These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.
3. If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
4. Wave soldering is only suitable for LQFP, TQFP and QFP packages with a pitch (e) equal to or larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
5. Wave soldering is only suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

16 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.

Notes

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.

17 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.

18 DISCLAIMERS

Life support applications — 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 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.

19 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.

NOTES

Philips Semiconductors – a worldwide company

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**.

Printed in The Netherlands 403506/02/pp112 Date of release: 2002 Apr 24 Document order number: 9397 750 09661

Let's make things better.

Philips
Semiconductors

PHILIPS