SIEMENS

ICs for Consumer Electronics

8-Bit Microcontroller with LCD-Dot Matrix Driver
SDA 20C440

Advanced Data Sheet 01.94

SDA 20C440 Revision History: 01.94	
Previous Releases: 05.93	
Page	Subjects (changes since last revision)

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at T_A = 25 °C and the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description are fulfilled.

For detailed technical information about **"Processing Guidelines"** and **"Quality Assurance"** for ICs, see our **"Short Form Catalog"**.

Edition 01.94
This edition was realized using the software system FrameMaker®

Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation, Balanstraße 73, D-81541 München.

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you - get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Contents **Page**

1 Features . . . 5
1.1 Functional Overview . . . 6
1.2 Pin Configuration . . . 7
1.3 Pin Definition and Functions . . . 8
1.4 Block Diagram . . . 11

2 Functional Description . . . 12
2.1 Architecture . . . 12
2.1.1 CPU Hardware . . . 12
2.2 Memory Organization . . . 13
2.2.1 Internal Program ROM . . . 13
2.2.2 Internal Data RAM . . . 13
2.2.3 Special Function Registers . . . 14
2.3 Interrupt System . . . 15
2.4 Processor Reset and Clock Circuit . . . 17
2.5 Ports and I/O Pins . . . 18
2.5.1 Port 0 . . . 18
2.5.2 Port 1 . . . 18
2.5.3 Port 2 . . . 18
2.5.4 Read-Modify-Write Feature . . . 19
2.6 Timer . . . 19
2.7 Serial Interface . . . 21
2.8 Analog to Digital Converter . . . 27
2.9 LCD-Matrix Controller . . . 28
2.9.1 LCD-Special Function Registers . . . 34
2.10 Instuction Set . . . 39
2.10.1 Notes on Data Addressing Modes . . . 39
2.10.2 Notes on Program Addressing Modes . . . 39
2.10.3 Instruction Set Description . . . 40
2.10.4 Instruction Opcodes in Hexadecimal Order . . . 44

3 Electrical Characteristics . . . 51
3.1 Absolute Maximum Ratings . . . 51
3.2 DC-Characteristics . . . 52
3.3 AC-Characteristics . . . 53

4 Application Example . . . 54

5 Package Outlines . . . 55

VAKATSEITE

.

SIEMENS

8-Bit Microcontroller with LCD-Dot Matrix Driver

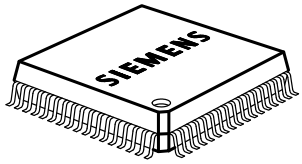
SDA 20C440

Preliminary Data **CMOS IC**

1 Features

Architecture

- CMOS-technology
- Processor core op-code compatible with SAB 8051 series
- On-chip oscillator and clock circuits (quartz oscillator or RC-oscillator, mask option)
- Binary or decimal arithmetic
- Signed-overflow detection and parity computation
- Integrated Boolean processor for control applications
- 83-ns clock cycle / 12-MHz internal operation frequency (4-MHz external frequency reference)
- Hardware multiply and divide
- Full depth stack for subroutine return linkage and data storage
- Three interrupt sources



P-MQFP-80

Type	Ordering Code	Package
SDA 20C440	Customer specific	P-MQFP-80 (SMD)

LCD-Dot Matrix Driver Unit

- 5 ¥ 7 dot character matrix
- 128 user defined characters (mask-programmable)
- SW-definable characters and symbols
- 2 ¥ 40 bytes of display RAM
- Selectable 8 or 16 line multiplexing drive (45 or 40 column drivers / 8 or 16 row drivers)
- Internal LCD-driving voltage divider with input for external reference voltage

On-Chip RAM

- Direct byte and bit addressability
- Four register banks
-
- 128 bytes of data memory (including 128 user-defined software flags)

On-Chip ROM

- Up to 4096 bytes of mask-programmable program memory

I/O-Lines

- One bidirectional 8-bit port (P0) comprising one line for external interrupt
- One 4-bit input port (P1) comprising four lines for digital or analog voltage input
- One bidirectional 2-bit port (P2) comprising two lines for the serial interface

Serial Interface

- Interface for asynchronous serial data communication
- Baudrate generator

Timer

- One 16-bit general purpose timer with 4-bit prescaler

1.1 Functional Overview

Designed for the use in car radio applications, the SDA 20C440 provides high processor performance as well as peripheral functions, specially a dedicated LCD-Dot Matrix Driver unit. The processor is based on the Siemens ECO-51 core, which is op-code compatible to the widely used 8051 controller series. This makes the controller suitable for even more applications using liquid crystal matrix displays.

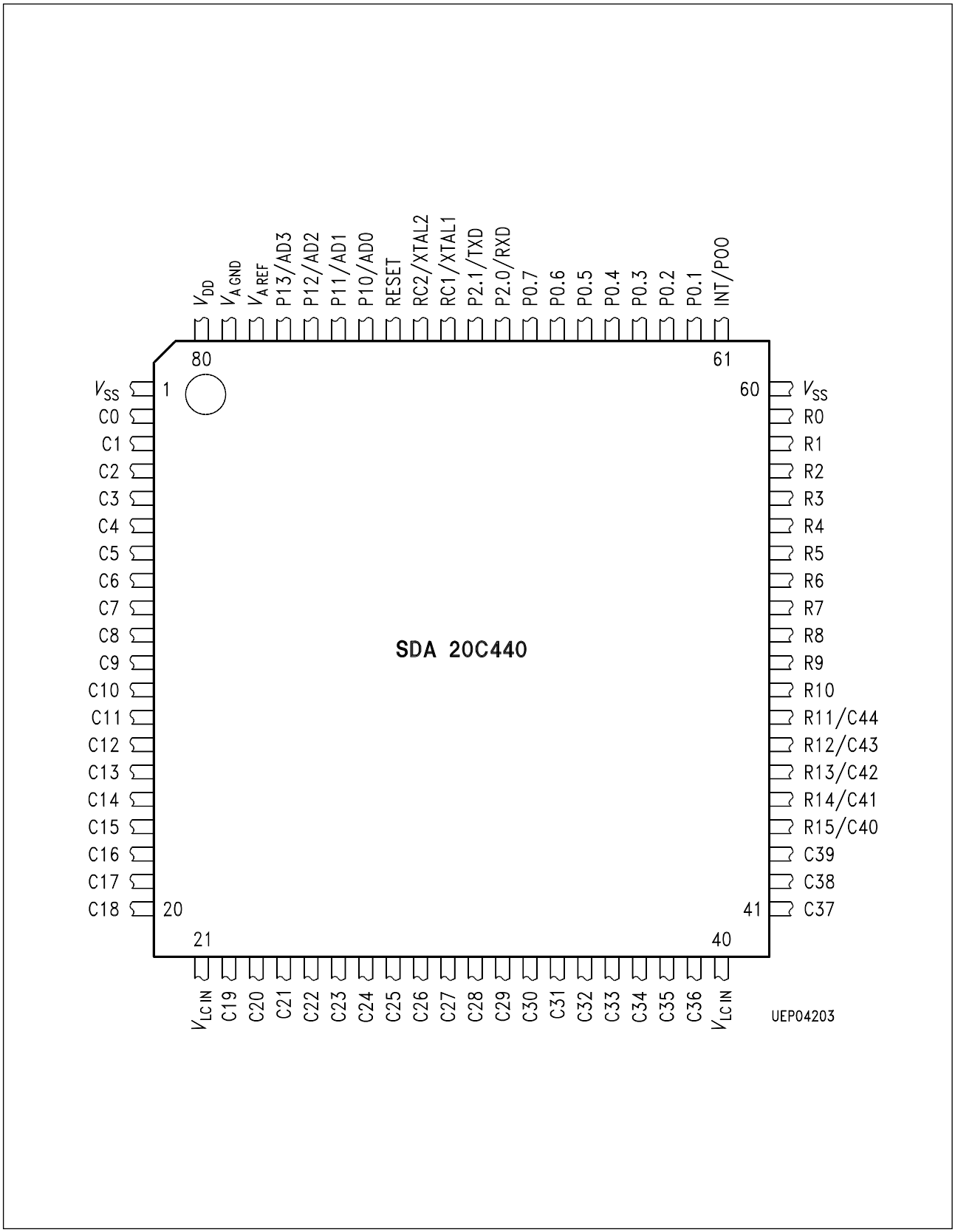
The members of the SDA 20C440 family of microcontrollers are:

– SDA 20C440 with factory-programmable 4-Kbyte on-chip ROM
– SDA 30C440E emulation version of the SDA 20C440
– SDA 20C840 with factory-programmable 8-Kbyte on-chip ROM
– SDA 30C840E emulation version of the SDA 20C840

Whereas the SDA 20C440 executes instructions from its internal ROM, the SDA 30C440E fetches them from the external emulation memory via an additional bus interface.

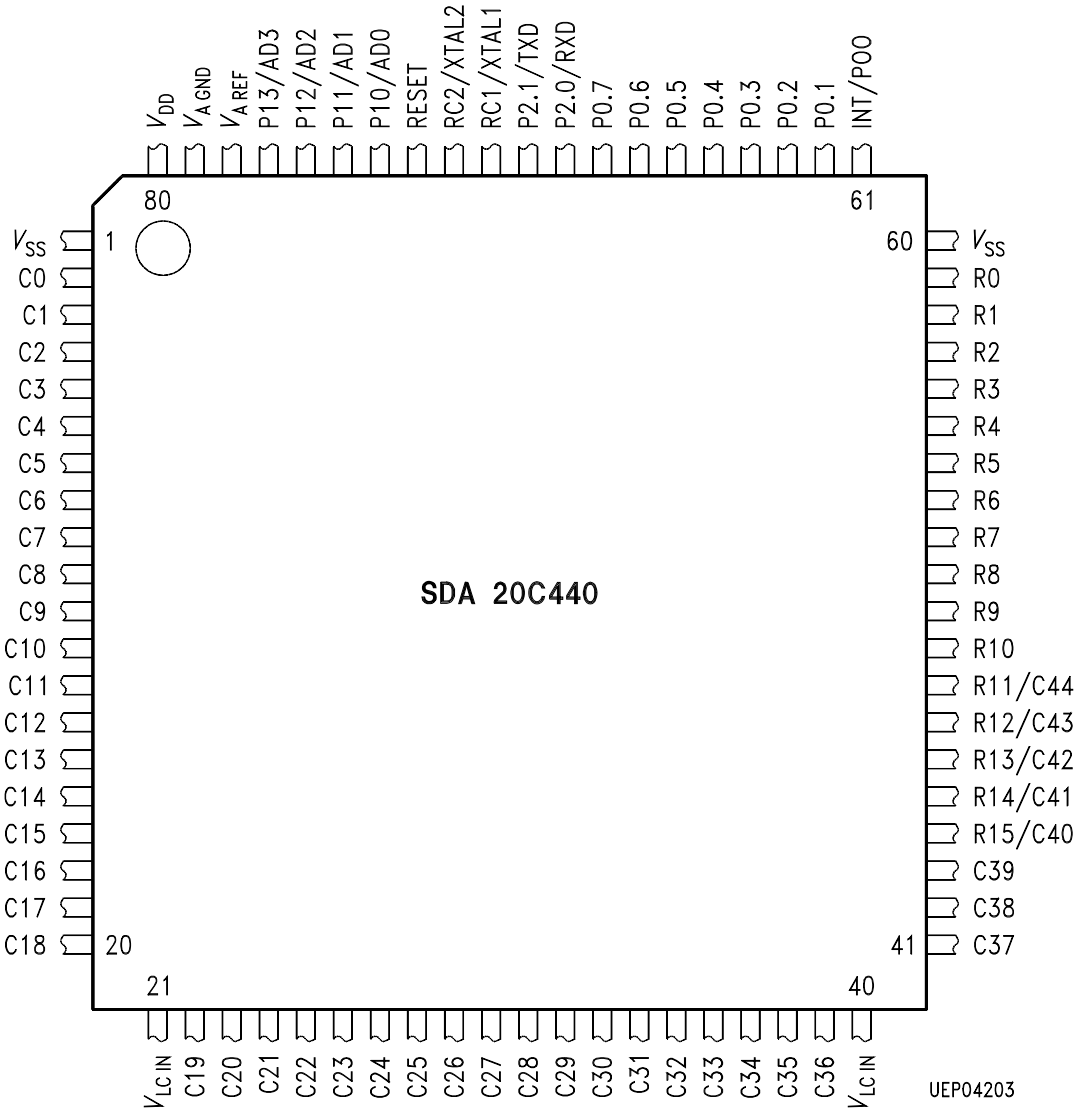
1.2 Pin Configuration

(top view)



1.2 Pin Configuration

(top view)



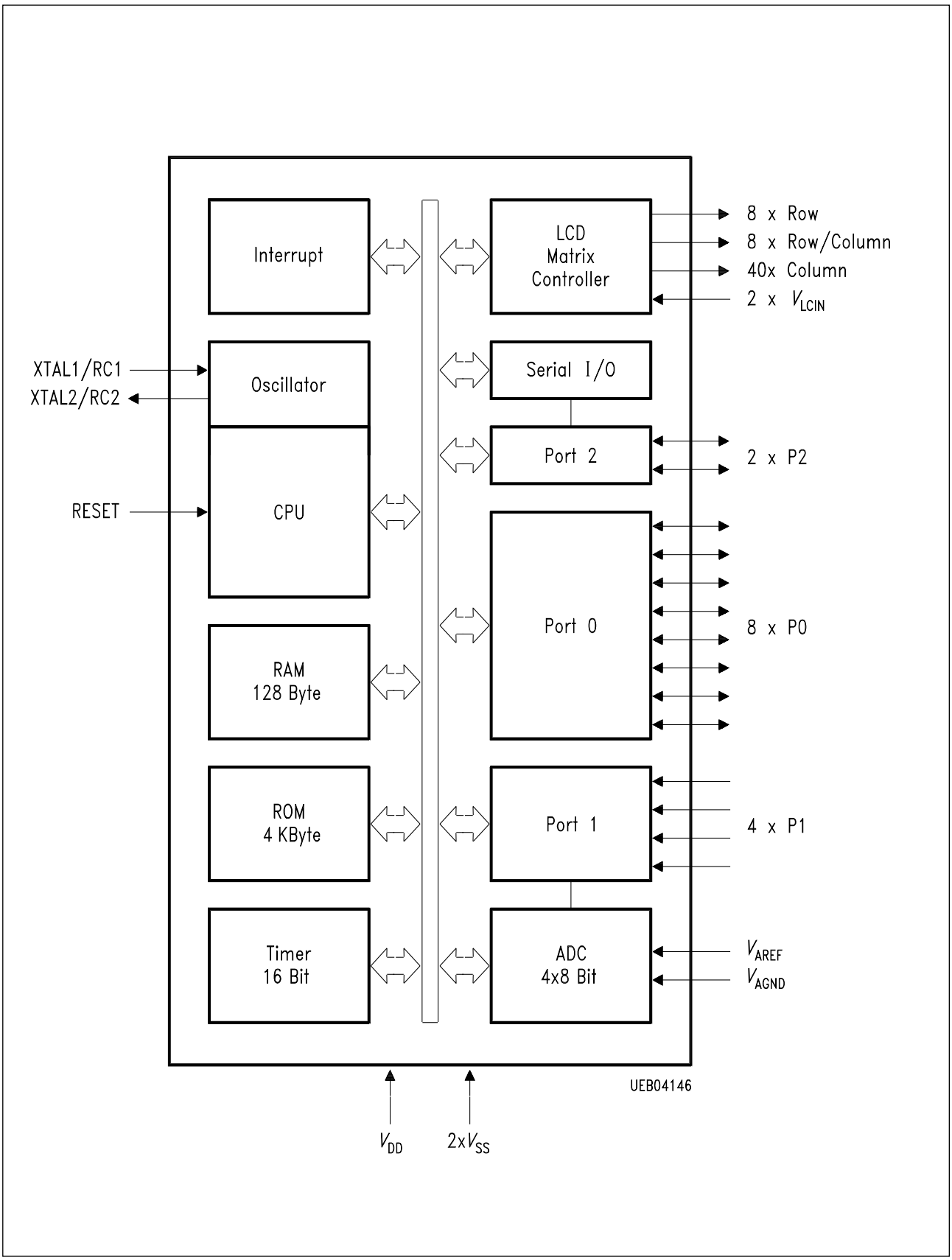
1.3 Pin Definitions and Functions

Pin No.	Symbol	Input (I) Output (O) Supply (S)	Function
61 62 63 64 65 66 67 68	P0.0/INT P0.1 P0.2 P0.3 P0.3 P0.5 P0.6 P0.7	I/O I/O I/O I/O I/O I/O I/O I/O	Port 0 is an 8-bit bidirectional I/O-port, comprising one line for external interrupt.
74 75 76 77	P1.0/AD0 P1.1/AD1 P1.2/AD2 P1.3/AD3	I I I I	Port 1 is a 4-bit input port, comprising four analog inputs for A/D-conversion.
69 70	P2.0/RXD P2.1/TXD	I/O I/O	Port 2 is a 2-bit bidirectional I/O-port, comprising the receive and transmit line for the serial interface.
71 72	XTAL1 XTAL2	I O	Input of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left open. Output of the inverting oscillator amplifier. Only supported, if an external quartz oscillator is used.
71 72	RC1 RC2	I O	Input to a Schmitt trigger inverter for RC-type oscillator (optional) Output of the Schmitt trigger inverter (optional)
73	RESET	I	A low level on this pin for at least 4 µs while the oscillator is running resets the processor.
80 1, 60	V_{DD} V_{SS}	S S	Power supply voltage Ground (0 V)
79 78	V_{AGND} V_{AREF}	S S	ADC-low reference voltage ADC-high reference voltage
21,40	V_{LCIN}	S	LCD-reference voltage inputs

1.3 Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O) Supply (S)	Function
59	R0	O	LCD-row driver output 0-10
58	R1	O	
57	R2	O	
56	R3	O	
55	R4	O	
54	R5	O	
53	R6	O	
52	R7	O	
51	R8	O	
50	R9	O	
49	R10	O	
44	R15/C40	O	LCD-row driver output 11-15
45	R14/C41	O	or LCD-column driver output 40-44
46	R13/C42	O	(depending on the selected multiplex mode)
47	R12/C43	O	
48	R11/C44	O	
2	C0	O	LCD-column driver output 0-39
3	C1	O	
4	C2	O	
5	C3	O	
6	C4	O	
7	C5	O	
8	C6	O	
9	C7	O	
10	C8	O	
11	C9	O	
12	C10	O	
13	C11	O	
14	C12	O	
15	C13	O	
16	C14	O	
17	C15	O	
18	C16	O	
19	C17	O	
20	C18	O	
22	C19	O	
23	C20	O	
24	C21	O	
25	C22	O	
26	C23	O	
27	C24	O	

1.4 Block Diagram



1.3 Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O) Supply (S)	Function
			LCD-column driver output 0-39 (cont'd)
28	C25	O	
29	C26	O	
30	C27	O	
31	C28	O	
32	C29	O	
33	C30	O	
34	C31	O	
35	C32	O	
36	C33	O	
37	C34	O	
38	C35	O	
39	C36	O	
41	C37	O	
42	C38	O	
43	C39	O	

2 Functional Description

2.1 Architecture

The controller uses the Siemens ECO-51 processor core. This core is opcode compatible to the 8051 microcontroller series. Because of a different internal timing, the instruction times differ from the original 8051 behavior.

For detailed information see chapter ‘Instruction Set’.

2.1.1 CPU Hardware

Program Memory

The controller addresses 4 Kbyte of internal mask programmable ROM.

Program Status Word Register (PSW)

The PSW-flags record processor status information and controls the operation of the processor. The carry (CY), auxiliary carry (AC), two user flags (F0 and F1), register bank select (RS0 and RS1), overflow (OV) and parity (P) flags reside in the Program Status Word Register. These flags are bit-memory-mapped within the byte-memory-mapped PSW. The CY-, AC-, and OV-flags generally reflect the status of the latest arithmetic operations. The CY-flag is also the Boolean accumulator for bit operations. The P-flag always reflects the parity of the A-register. F0 and F1 are general purpose flags which are pushed onto the stack as part of a PSW-save. The two register bank select bits (RS1 and RS0) determine which one of the four register banks is selected as follows:

RS1	RS0	Register Bank	Register Location
0	0	0	$00_H – 07_H$
0	1	1	$08_H – 0F_H$
1	0	2	$10_H – 17_H$
1	1	3	$18_H – 1F_H$

MSB	SFR-Address: **D0**$_H$						LSB
PSW: Program Status Word							
CY	AC	F0	RS1	RS0	OV	F1	P

Stack Pointer (SP)

The 8-bit stack pointer contains the address at which the last byte was pushed onto the stack. This is also the address of the next byte that will be popped. The SP is incremented during a push. SP can be read from or written to under software control. The stack may be located anywhere within the internal data RAM-address space and may be as large as 128 bytes.

Figure 1
Memory Map

Data Pointer Register (DPTR)

The 16-bit data pointer register DPTR is the concatenation of registers DPH (high-order byte) and DPL (low-order byte). The DPTR is used in register-indirect addressing to move program memory constants. DPTR may be manipulated as one 16-bit register or as two independent 8-bit registers DPL and DPH.

Port 0, Port 1, Port 2

The three ports P0 ... 2 provide 10 I/O-lines to interface to the external world. Port 0 and Port 1 are both byte and bit addressable. Port 2 is only byte addressable and contains two bits, which can be selected as a serial interface and for normaly I/O. Port 0 is used for binary I/O. Port 1 contains up to four ADC-input channels.

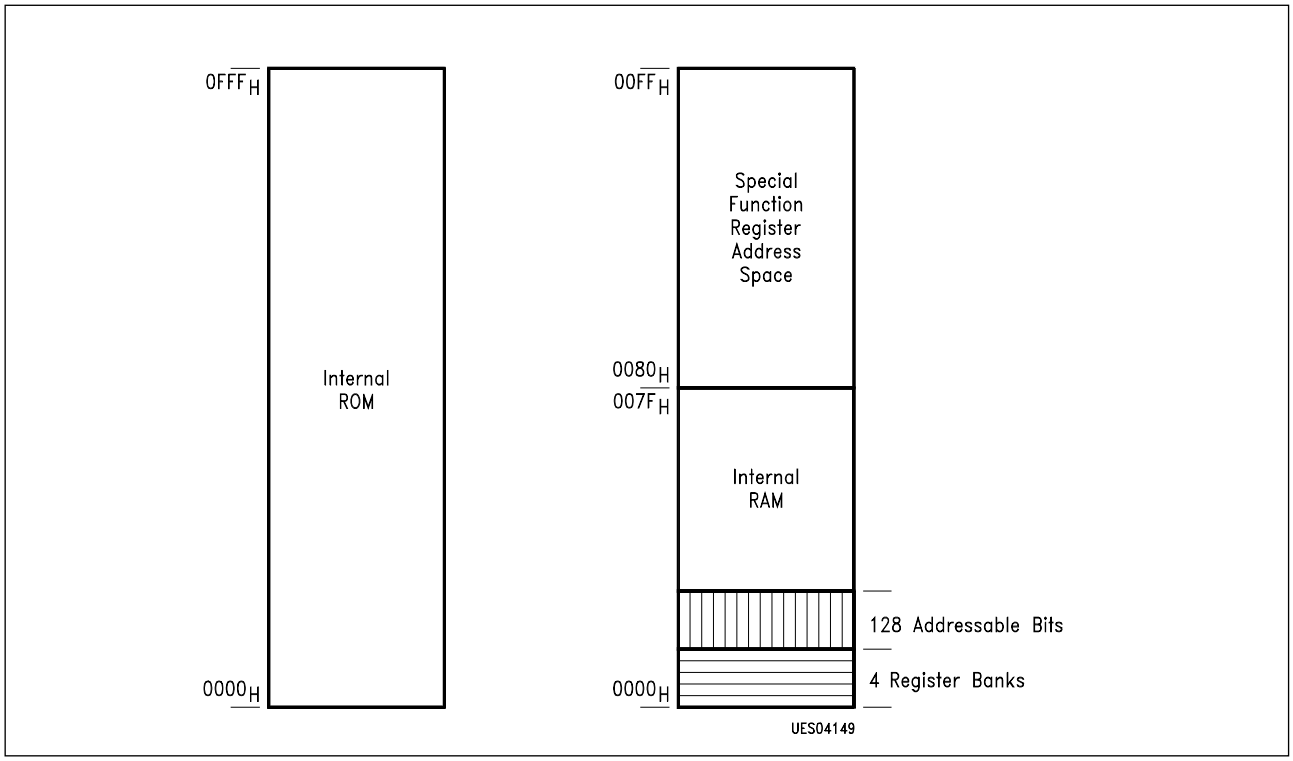
2.2 Memory Organization

2.2.1 Internal Program ROM

An internal program ROM of 4 Kbyte is provided. The ROM is mask-programmable. After reset, the operation starts at address 0000_H.

2.2.2 Internal Data RAM

An internal data RAM of 128 bytes of memory is provided. This internal RAM comprises 128 software flags, which are usable by the implemented bit operation instructions.



2.2.3 Special Function Registers

Accesses to peripheral and CPU functions are done by writing to or reading from special function registers.

Register Function	Symbolic Name	Address (hexadecimal)	Value after Reset
Timer High Byte	TH	$F9_H$	00_H
Timer Low Byte	TL	$F7_H$	00_H
Timer Reload Value High Byte	TRH	$F6_H$	00_H
Timer Reload Value Low Byte	TRL	$F5_H$	00_H
Timer Control Register 0	TCON	DC_H	00_H
ADC-Control Register *	ADCON	90_H	00_H
ADC-Address Register	ADADR	91_H	00_H
ADC-Data Register	ADDAT	92_H	00_H
Data Pointer High	DPH	83_H	00_H
Data Pointer Low	DPL	82_H	00_H
Stack Pointer	SP	81_H	07_H
Accumulator *	A	$E0_H$	00_H
B-Register *	B	$F0_H$	00_H
Program Status Word *	PSW	$D0_H$	00_H
Oscillator Control Register	OSCON	$9A_H$	01_H
Interrupt Enable Register 0	IENA0	$C8_H$	00_H
Interrupt Enable Register 2	IENA2	$C9_H$	00_H
Interrupt Request Register *	IREQ0	$A8_H$	02_H
Serial Port Control Register 0	SACON0	$C1_H$	00_H
Serial Port Control Register 1 *	SACON1	$C0_H$	00_H
Serial Port Transmit/Receive Buffer	SABUF	$C3_H$	00_H
Serial Port Prescaler	SASCAL	$C4_H$	00_H
Serial Port Baud Rate Counter	SABAUD	$C2_H$	00_H
Port 0 Register *	P0	80_H	00_H
Port 0 Direction Register	P0PDR	84_H	00_H
Port 1 Register *	P1	88_H	00_H
Port 2 Register	P2	$8F_H$	00_H
Port 2 Direction Register	P2PDR	94_H	00_H
Port 2 Function Register	P2PFR	95_H	00_H
LCD-Address Register	LCDADR	BC_H	00_H
LCD-Code Register	LCDCOD	$B9_H$	00_H
LCD-Bit Data Register	LCDBIT	BB_H	00_H
LCD-Word Data Register	LCDWRD	BA_H	00_H
LCD-Mode Register *	LCDMOD	$B8_H$	00_H

*) These registers are bit-addressable

2.3 Interrupt System

Three interrupt sources are provided, which are:

- Timer Interrupt
- Serial Interface Interrupt
- External Interrupt

Each of these interrupt vectors to its special interrupt routine location. All interrupts are serviced at the same priority level. Before enabling an interrupt source, the corresponding request bit in IREQ0 has to be set to 0.

Three special function registers are used to control the interrupt functions:

IENA0: The Interrupt Enable Register contains one enable bit for each interrupt source.
IENA2: The Global Interrupt Enable Register contains one global enable bit to enable or disable all individually enabled interrupts.
IREQ0: The Interrupt Request Register contains one request bit for each interrupt; these bits are set and cleared by hardware but may be set or cleared by software as well.

Each interrupt vector to its own interrupt location:

INT0	Serial Port Interrupt	0006_H
INT1	External Interrupt	0009_H
INT2	Timer Interrupt	$000C_H$

Response time: between 0.91 µs and 2.3 µs at an internal operating frequency of 12 MHz.

Interrupt Registers:

MSB	SFR-Address: **C9**$_H$						LSB
IENA2: Global Interrupt Enable Register							
...	...	...	...	...	...	...	IEA

Default after reset: XXXX XXX0

IEA: = 1 enable all Interrupts

IENA2.1 – IENA2.7: reserved, to be set to 0

MSB	SFR-Address: **C8**$_H$						LSB
IENA0: Interrupt Enable Register 0							
...	...	...	...	...	IE2	IE1	IE0

Default after reset: XXXX X000

IE2: = 1 enable Timer Interrupt

IE1: = 1 enable External Interrupt

IE0: = 1 enable Serial Port Interrupt

IENA0.3 – IENA0.7: reserved, to be set to 0

MSB	SFR-Address: **A8**$_H$						LSB
IREQ0: Interrupt Request Register							
...	...	...	...	...	INT2	INT1	INT0

Default after reset: XXXX XXXX

INT2: = 1 Timer Interrupt is requested

INT1: = 1 External Interrupt is requested

INT0: = 1 Serial Port Interrupt is requested

IREQ0.3 – IREQ0.7: reserved, to be set to 0, if writing to IREQ0

2.4 Processor Reset and Clock Circuit

Processor Reset and Initialization

A low level at pin RESET for at least 4 µs resets the processor. All port lines are set to input mode. After reset, most special function registers are set to 00H, except for:

- Stack Pointer Register (set to 07_H)
- Oscillator Control Register (set to 01_H)

(for detail see chapter 'Memory Organization')

Important Note: When entering reset, port lines P0.0 and P0.1 are to be held at high level (or left open). Otherwise the processor test mode is entered.

Processor Clock Circuit

The internal processor clock is derived from an external reference, which may be a quartz/ceramic resonator or an RC-oscillator (selected by mask option). The reference frequency is 4 MHz, the internal operating frequency is 12 MHz. See chapter 'AC-Characteristics' for reference frequency limits.

The operating frequency is generated by a PLL circuit. After reset, the PLL is disabled and the processor runs at the external reference frequency (4 MHz). By setting the oscillator control register (OSCON, see below), the operating frequency is set to 12 MHz. Once being activated, the PLL cannot be switched off any more. The following sequence is to be used for initializing the clock system after reset:

MOV OSCON,#41_H
MOV OSCON,#10_H

This should be done before using one of the peripheral modules (timer, LCD-unit etc.)

MSB	SFR-Address: **$9A_H$**						LSB
OSCON: Oscillator Control Register							
...	OSC	...	OSF	...	...	...	PLLD

PLLD: PLL disable
= 1: PLL is deactivated
= 0: PLL is activated

Note: After activating the PLL once after reset, a deactivation is not possible any more.

OSC, OSF: Oscillator frequency select for 12-MHz system clock
(see initialization sequence)

Default after reset: 01_H

2.5 Ports and I/O Pins

2.5.1 Port 0

Port 0 is an eight-bit bidirectional port for digital I/O. One line of Port 0 (P0.0) is alternatively used for external interrupts. The port register (P0) is used for reading data from and writing data to the port pins. Two hardware modes are provided (I/O-mode and output mode), where I/O-mode is selected by default after reset. During reset, port lines P0.0 and P0.1 have to be held at a logical high level or left open to prevent the processor from entering the processor test mode.

In I/O-mode (which is the default after reset), an instruction that uses a port's bit/byte as a source operand reads a value that is the logical AND of the last value written to the bit/byte and the polarity being applied to the pin/pins by an external device (this assumes that none of the processor's electrical specs are being violated). An instruction that reads a bit/byte, operates on the content, and writes the result back to the bit/byte, reads the last value written to the bit/byte instead of the logical level at the pin/pins. Pins comprising a single port can be made a mixed collection of inputs and outputs by writing a "one" to each pin that is to be an input. Each time an instruction uses a port as the destination, the operation must write "ones" to those bits that correspond to the input pins.

All the port latches have "one"s written to them by the reset function. If a "zero" is subsequently written to a port latch, it can be reconfigured as an input by writing a "one" to it. The instructions that perform a read of, operation on, and write to a port's bit/byte are INC, DEC, IB, JBC, SET, CL, MV P.X., CJD, DJNZ, AND, OR, and XOR. The source read by these operations is the last value that was written to the port, without regard to the levels being applied at the pins. This insures that bits written to a "one" (for use as inputs) are not inadvertently cleared. Ports 0 has "quasi-bidirectional" output drivers which comprise an internal pullup resistor. When configured as inputs they pull high and will source current when externally pulled low.

In Port 0 the output drivers provide source current for a short time if, and only if, software updates the bit in the output latch from a "zero" to an "one". Sourcing current only on "zero to one" transition prevents a pin, programmed as an I/O-line, from sourcing current into the external device that is driving the input pin.

If a bit in port direction register P0PDR is cleared (i.e. set to 0), the corresponding pin high level is constantly driven by a strong output transistor. In this case the pin should not be pulled high or low externally (output mode).

2.5.2 Port 1

Port 1 is a four-bit input port for digital or analog signals (may be used as input to the internal A/D-converter). Only deselected analog lines may be used as digital inputs. The upper four bits of port 1 register are not implemented and may show random values, if read.

2.5.3 Port 2

Port 2 is a two-bit bidirectional port and may be used as the serial interface receive and transmit lines.

Port function of Port 2 is the same as for Port 0. The upper six bits of Port 2 register P2 and Port 2 direction register P2PDR are not implemented and may show random values, if read. If the serial interface is used, the Port 2 function register P2PFR has to be set to 03_H. If the serial interface is not

used, the P2PFR is to be set to 00_H, which is default value after reset. Writing of other values than 00_H or 03_H to P2PFR is not alllowed.

2.5.4 Read-Modify-Write Feature

“Read-modify-write” commands are instructions that read a value, possibly change it, and then rewrite it to the latch. When the destination operand is a port or a port bit, these instructions read the latch rather than the pin. The read-modify-write instructions are listed below:

INC, DEC, JB, JBC, SET, CL, MV P.X., DJNZ, AND, OR, and XOR.

The read-modify-write instructions are directed to the latch rather than the pin in order to avoid a possible misinterpretation of the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a “one” is written to the bit, the transistor is turned on.

If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of “one”.

2.6 Timer

A 16-bit timer with auto-reload is provided. The timer clock is derived from the external oscillator clock rate (4 MHz) by a programmable prescaler in five steps (1 – 1/2 – 1/4 –1/8 – 1/16).

An interrupt request is generated by timer overflow which may be masked by use of the interrupt enable registers IENA0 and IENA2 (see chapter ‘Interrupt System’). The timer is started/stopped by setting/clearing bit RUN in register TCON. The prescale factor is selected by setting a three bit value (bits PS0, PS1 and PS2) in register TCON. If a 16-bit timer value is to be written to the timer register or to the timer reload register, the high byte (TH, TRH) is to be written first. After writing the low byte (TL, TRL), the complete 16-bit word will be transferred from a 16-bit shadow register into the internal pair of registers (timer registers TH/TL, reload registers TRH/TRL). If a 16-bit value is to be read from the timer registers or timer reload registers, the low byte has to be read first to transfer the 16 bit value into a 16-bit shadow register. After this, the high byte can be read. Please note, that for correct operation always two read or write access in the right order are necessary. After reset, the values of the timer registers, the reload registers and the two pairs of shadow registers are undefined and the timer is stopped.

MSB	SFR-Address: **DC_H**						LSB
TCON: Timer Control Register							
...	...	...	...	RUN	PS2	PS1	PS0

PS0..2 prescaler
= 000: factor 1
= 001: factor 1/2
= 010: factor 1/4
= 011: factor 1/8
= 100: factor 1/16

RUN start and stop bit
= 1: start
= 0: stop

TCON.4 – TCON.7: reserved, to be set to 0

Default after reset: XXXX 0000_H

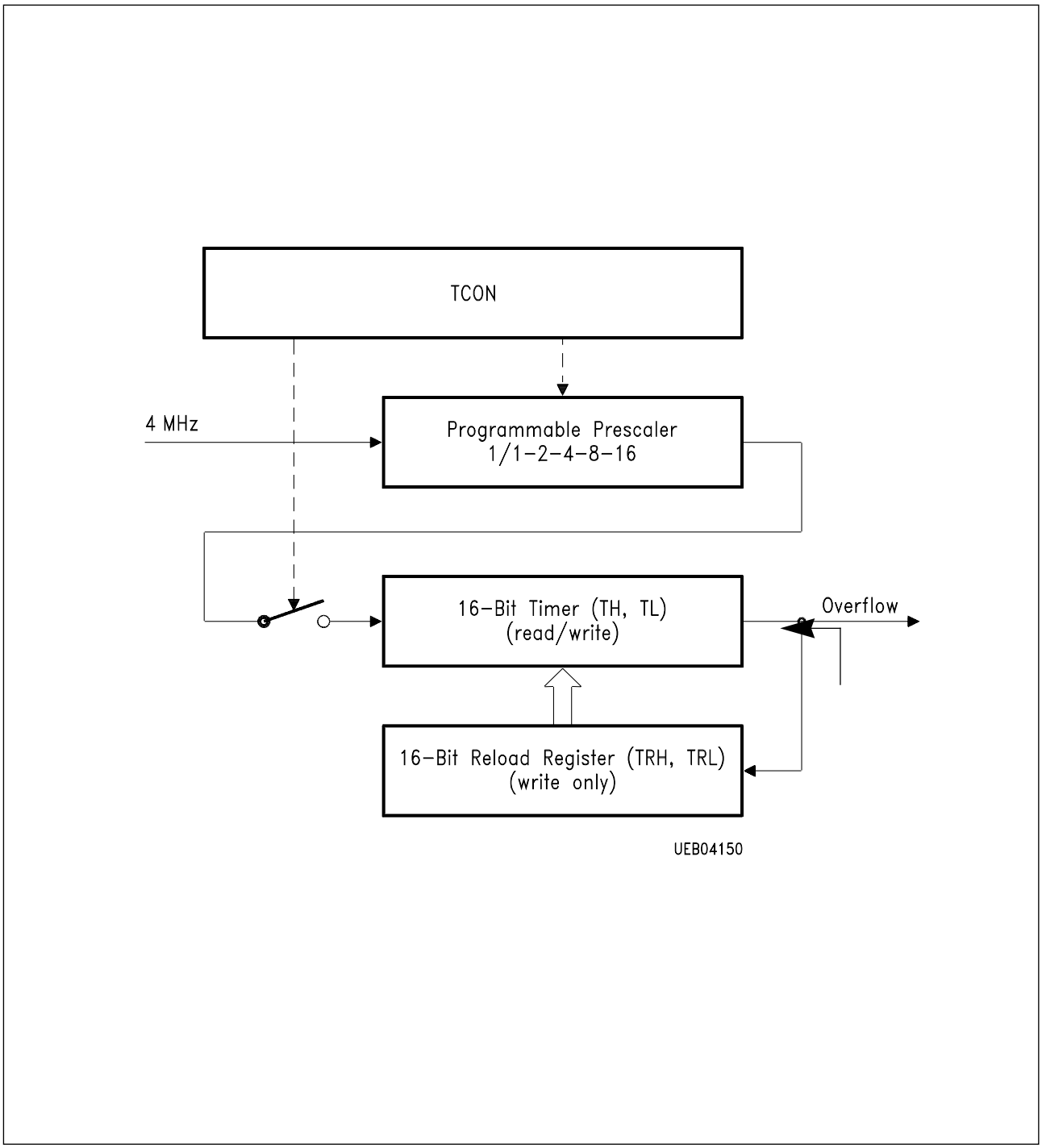


Figure 2
Timer Block Diagram

2.7 Serial Interface

The asynchronous serial interface (SART, Serial Asynchronous Receiver & Transmitter) is used for communication with other controllers or electrical systems. This modul works in full-duplex mode, so it can send and receive data at the same time. Moreover, it comprises a dedicated programmable baud-rate generator.

Port 2.0 and Port 2.1 are used for input and output of the serial data. To allocate these pins to the SART-module, the according bits of the port function register in the special function register must be set.

Three types of data transmission errors are detected: Parity-, Framing- and Overwrite-Error. Three different interrupt-outputs exist for the following events: Transmit-Ready, Receive-Ready and Error-Detected.

The transmit and receive registers are double buffered, so that during the reception of the first byte the second byte can be received, as well as the transmission can run continuously.

Data Sequence

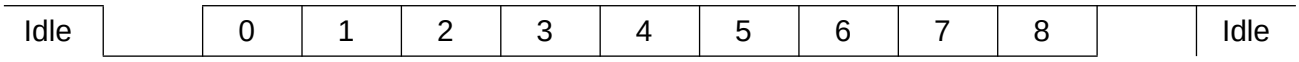
The data words consists of:

- 1 start bit (= ‘0’)
- in mode I : 8 data bits
  in mode II : 8 data bits, 1 parity bit
  in mode III : 8 data bits, 1 address bit
- 1 stop bit (= ‘1’)

During idle state the output is held at ‘1’..

The transfer of the data bits starts with the least significant bit (LSB).

Mode I : no bit
Mode II : 1 parity bit
Mode III : 1 address bit



Start Stop

Operating Modes

The three possible modes are switched with bits SM0 and SM1 of the register SACON0 as shown in table below.

SM0	SM1	Mode	Description
0	0	I	8-bit data
0	1	II	8-bit data, 1 parity bit
1	0	III	8-bit data, 1 address bit
1	1	IV	reserved

Multiprocessor-Communication

Mode III is used for communication in multiprocessor systems. In this mode, 9 data bits are received and transmitted. The contents of the nineth received data bit is written into bit RB8 of register SACON1, and the nineth transmitted data bit is taken from bit TB8 of the register SACON1.

When the mask-address-interrupt bit MAI of register SACON1 is set, the SART-module can be programmed in a way that after a stopbit was received, a transmit-interrupt is released only if bit RB8 is set. If bit MAI is not set, a transmit-interrupt is never caused when nine databits and the stopbit are received. Setting of bit MAI is only to be used in mode III.

Loop Back Mode

The loop back mode is only used for test purpose. It is activated when the bit LBE of the SACON0 register is set. In this case, the transmitted informations do not appear on the output pin but they are switched to the receive part of the SART-module internally. Therefore the transmitted bits are received simultaneously. The disconnecting of the pins is independend of the setting of the port function register.

All the control bits are in function, the test can use all kinds of data transmission, all unmasked interrupt sources also release an interrupt and the receiver must be activated.

Baudrate Generator

The speed of the communication is set by programming the 8-bit-reload-counter and the divider of the integrated baudrate generator. The baudrate is derived from the clock of a central prescaler and based on the external reference frequency (quartz, ceramic resonator or RC-oscillator).

Any value between 0 and FF_H may be written into the baudrate reload register SABAUD. The baudrate is calculated as follows:

$$BR: = \frac{f_{REF}}{2^{(SASCAL + 1)} \times (SABAUD + 1)}$$

Standard baudrates may be set according to the following table:

Baudrate [kBd]	SASCAL	SABAUD	Fault [%]
2.4	3	103	+ 0.16
4.8	3	51	+ 0.16
9.6	3	25	+ 0.16
10.4	3	23	+ 0.16
19.2	3	12	+ 0.16
38.4	0	51	+ 0.16
57.6	0	34	– 0.79
76.8	0	25	+ 0.16
115.2	0	16	+ 2.12
180	0	10	+ 1.01

Security of Data

Following errors are detected by the SART-module:

- **Framing Error**
  A logical ‘0’ is received instead of the expected stopbit (logic ‘1’). A framing-error sets error flag FE.
- **Parity Error**
  The value of the received parity bit is not equal to the parity of the according data byte. Even parity is assumed. A parity error sets error flag PE.
- **Overwrite Error**
  The databyte in the register RXBUF is overwritten by a new databyte before the first databyte was read by the CPU. An overwrite error sets error flag OE.

The occurrence of one of these errors is signaled to the CPU using the control interrupt CNTRINT. The evaluation of the error flags identifies the appropriate cause. The release of the control interrupt CNTRINT can be suppressed for each fault individually by setting the appropriate mask bit to 0 (MFE, MPE and MOE).

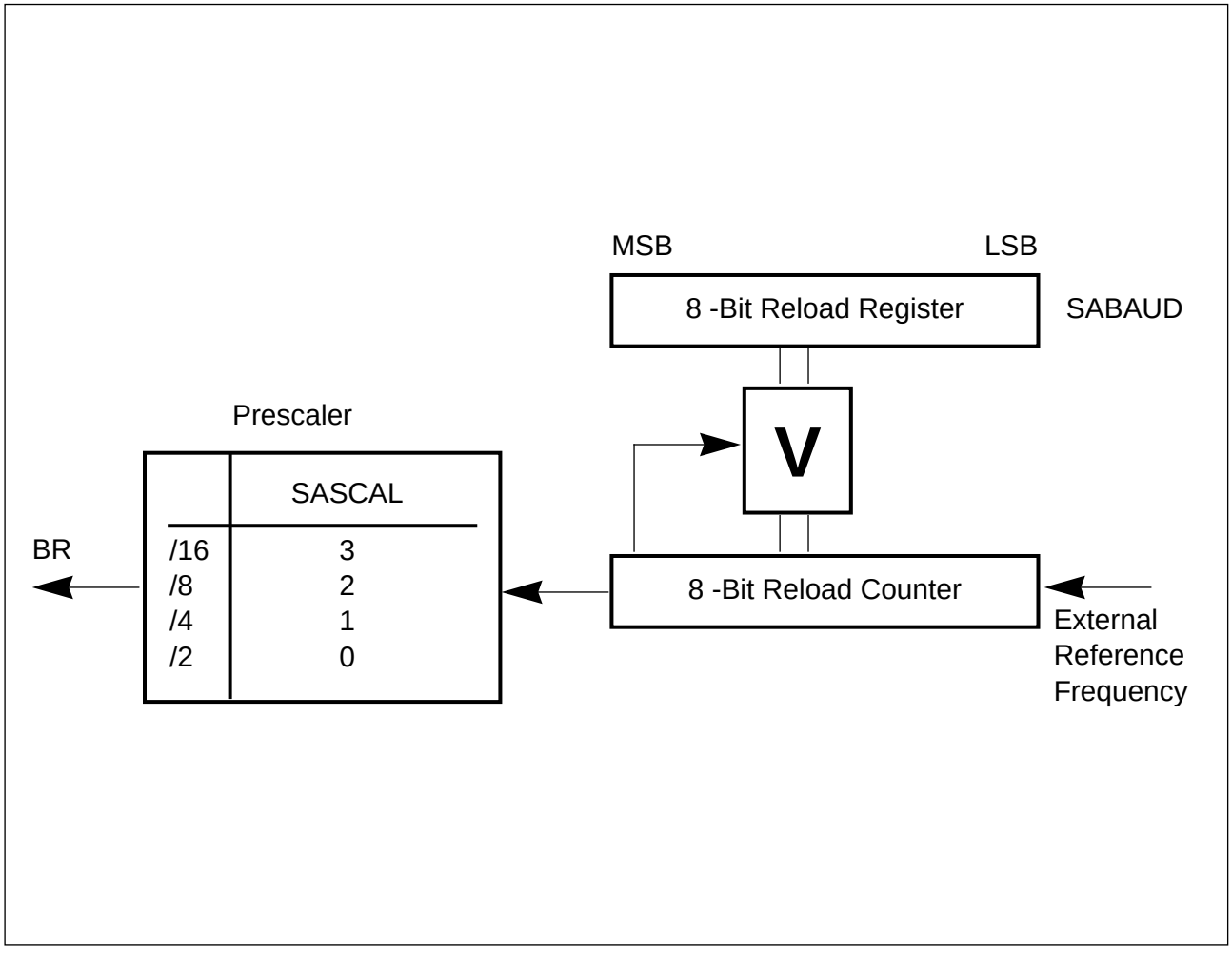


Figure 3
Block Diagram of the Baudrate Generator

Data Reception

In idle state the dataline on pin RXD is held at high level. The beginning of a data byte starts with the first ‘1 – 0’ transition (start bit), which also synchronizes the baudrate generator. After this, eight or nine data bits are read according to the selected mode with the clock of the baudrate generator. In mode II the parity of the data byte is compared with the parity bit automatically. The stop bit is always tested for the value ‘1’. If an error is detected, the appropriate error flag is set and causes an interrupt, if enabled.

In case of no error, the received data byte is written into the register RXBUF and at the same time the reception is signaled to the CPU using interrupt RXINT. If the data byte in the receive register RXBUF is to be overwritten by a new data byte before the CPU has read this register, the overwrite error flag is set and the control interrupt is activated.

Data Transmission

The transmission of a data word is started automatically by writing to the transmit register. The data word of register TXBUF is loaded into the shift register TSREG. With the clock of the baudrate generator the data value is shifted to pin TXD. The data word consists of an automatically generated start bit, eight or nine data bits and one stop bit. In mode II the parity bit is computed and is transmitted as the nineth data bit.

Transmit and Receive Register

The CPU writes the data to be sent into the transmit buffer TXBUF. It is a special function register with write-only access. While reading from this address location, the receive register RXBUF will be available.

MSB	SFR-Address: **$C3_H$**						LSB
SABUF: Serial Port Transmit/Receive Buffer							
D7	D6	D5	D4	D3	D2	D1	D0

Receive-Register

The SART-module writes the received data into the receive register RXBUF. It is a special-function-register with ‘read-only’ access. While writing with this address the transmit register will be written.

Control Register SACON0 and SACON1

MSB	SFR-Address: $\mathbf{C1}_H$						LSB
SACON0: Serial Port Control Register 0							
MOE	MFE	MPE	SM1	SM0	LBE	REN	ENA

MOE: Masking Overwrite Error*
= 0: Overwrite error is masked

MFE: Masking Framing Error*
= 0: Framing error is masked

MPE: Masking Parity Error*
= 0: Parity error is masked

SM0
SM1: These bits select the operating mode

LBE: Loop-Back Enable
= 1 Loop-back is enabled

REN: Receiver Enable
= 1 Receiver is enabled

ENA: SART-Module enable
= 1 The SART-module is enabled

*) The masking of an error suppresses the start of an interrupt. If an error is detected the corresponding error flag is set independent of the masking.

Default after reset: 0000 0000

MSB	SFR-Address: **C0**$_H$						LSB
SACON1: Serial Port Control Register 1							
OE	FE	PE	MAI	TB8	RB8	TI	RI

OE: Overwrite Error Flag
This flag is set by hardware when an overwrite error occurred. It must be cleared by software.

FE: Framing Error Flag
This flag is set by hardware when a framing error occurred. It must be cleared by software.

PE: Parity Error Flag
This flag is set by hardware when a parity error occurred. It must be cleared by software.

MAI: Masking Address Interrupt
= 1 address interrupt is enabled (only in mode III)

TB8: Transmitter Data Bit
In mode III this bit is transmitted as the nineth data bit.

RB8: Receiver Data Bit
In mode III the ninety received bit is written to RB8.

TI: Transmitter Interrupt
This flag is set by hardware when the start bit is transmitted and must be cleared by software.

RI: Receiver Interrupt
This flag is set by hardware when the stop bit is received and must be cleared by software.

Default after reset: 0000 0000

Baudrate Register
With this register the baud rate is adjusted.

MSB	SFR-Address: **C2**$_H$						LSB
SABAUD: Serial Port Baud Rate Counter							
BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0

The contents of this register defines the reload value of the baud rate reload counter.

Baudrate Divider Register (Prescaler)

The divider value for the prescaler is selected with this register.

MSB	SFR-Address: $\mathbf{C4}_H$						LSB
SASCAL: Serial Port Prescaler							
...	...	...	...	...	...	SCL1	SCL0

SCL 1/0	**Division by**
00	2
01	4
10	8
11	16

Default after reset: XXXX XX11

2.8 Analog to Digital Converter

The Analog to Digital Converter unit (ADC) provides the following features:

- 8-bit resolution
- four multiplexed analog inputs
- inputs for external reference voltages
- about 17-µs conversion time, including sample and hold at 12-MHz internal operating frequency

For detailed information see chapter ‘DC-Characteristics’.

The ADC uses the principle of successive approximation by switched capacitance array.
ADC-Special Function Registers:

MSB	SFR-Address: $\mathbf{91}_H$						LSB
ADADR: ADC-Address Register							
...	...	...	...	...	ADR2	ADR1	1

ADADR.0: to be set to 1

ADR1, ADR2: selection of the channel result to be read:
00 – channel 0 (P1.0)
01 – channel 1 (P1.1)
10 – channel 2 (P1.2)
11 – channel 3 (P1.3)

ADADR.3: reserved, to be set to 0
ADADR.4: reserved, to be set to 0
ADADR.5: reserved, to be set to 0
ADADR.6: reserved, to be set to 0

ADADR.7: reserved, to be set to 0

Default after reset: XXXX X00X

MSB	SFR-Address: **92$_H$**						LSB
ADDAT: ADC-Data Register							
DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0

DAT0 … 7: conversion result of the channel addressed by ADADR

Default after reset: XXXX XXXX

MSB	SFR-Address: **90$_H$**						LSB
ADCON: ADC-Control Register							
BSY	ADS	…	EOC	…	MX2	MX1	MX0

MX1, MX0: Selection of the analog input:
00 – channel 0 (P1.0)
01 – channel 1 (P1.1)
10 – channel 2 (P1.2)
11 – channel 3 (P1.3)

MX2: = 1 deselects all channels
Deselected channels may be used as digital inputs (Port 1)

EOC: End of Conversion;
set by ADC after completion of a conversion, to be cleared by software

ADS: ADC Start; setting to 1 starts a conversion, reset by hardware after the end of a conversion

BSY: Busy Bit; a ‘1’ indicates, that a conversion is in progress or that the result has not been read yet

ADCON.3/5: reserved, to be set to 0

Default after reset: 00X0 XX00
This register is bit-addressable.

LCD-Matrix Controller

For the use of multiplexed Liquid Crystal Displays, the processor provides an LCD-control module, which allows to connect a multiplex driven LCD. The module communicates with the CPU via an interface and all settings are done by writing to special function registers. An overview of this module is given in **figure 4.**

The device can be used for 8-line or 16-line multiplexing drive as shown in **figures 5 to 7.**

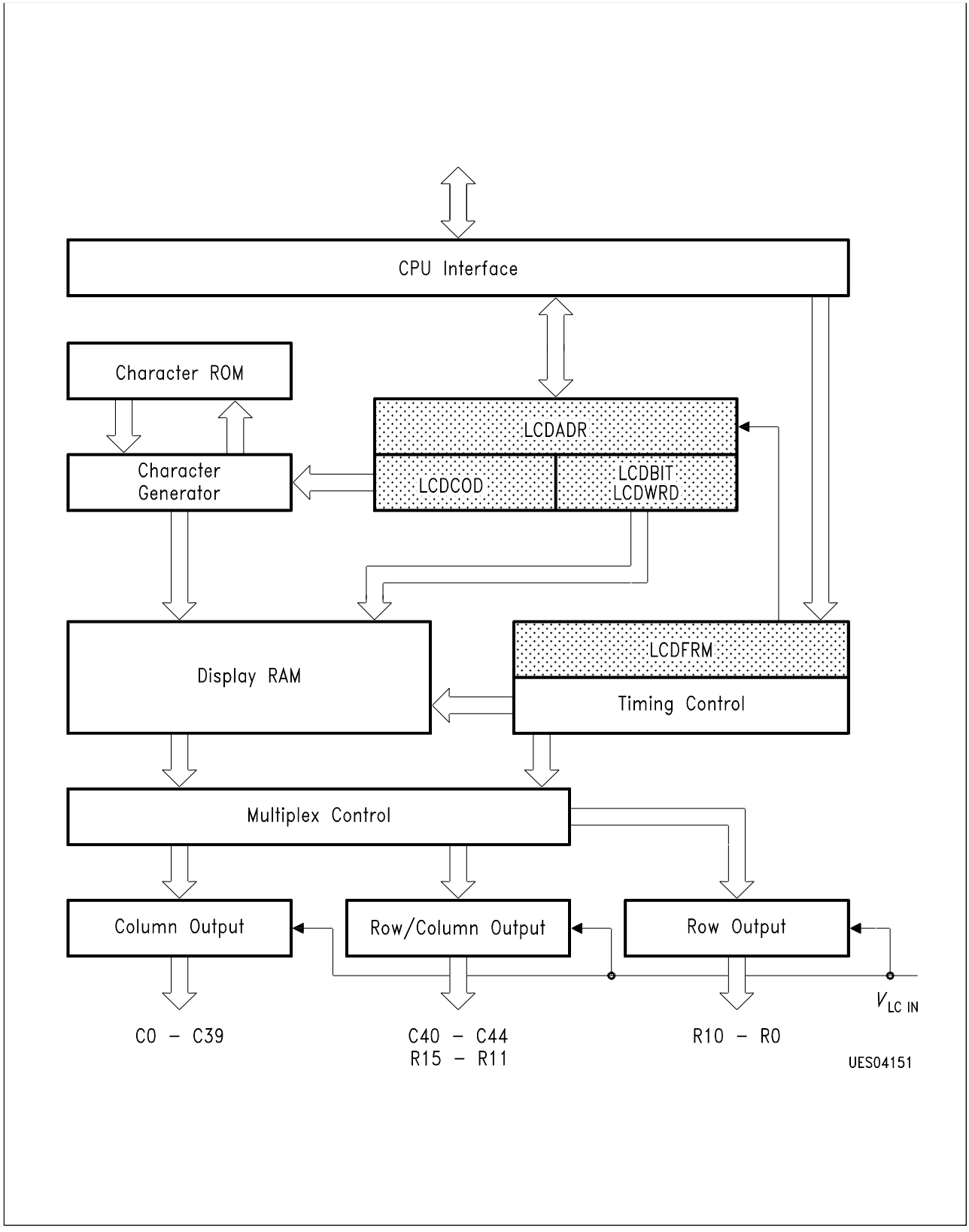


Figure 4
LCD-Matrix Controller Unit

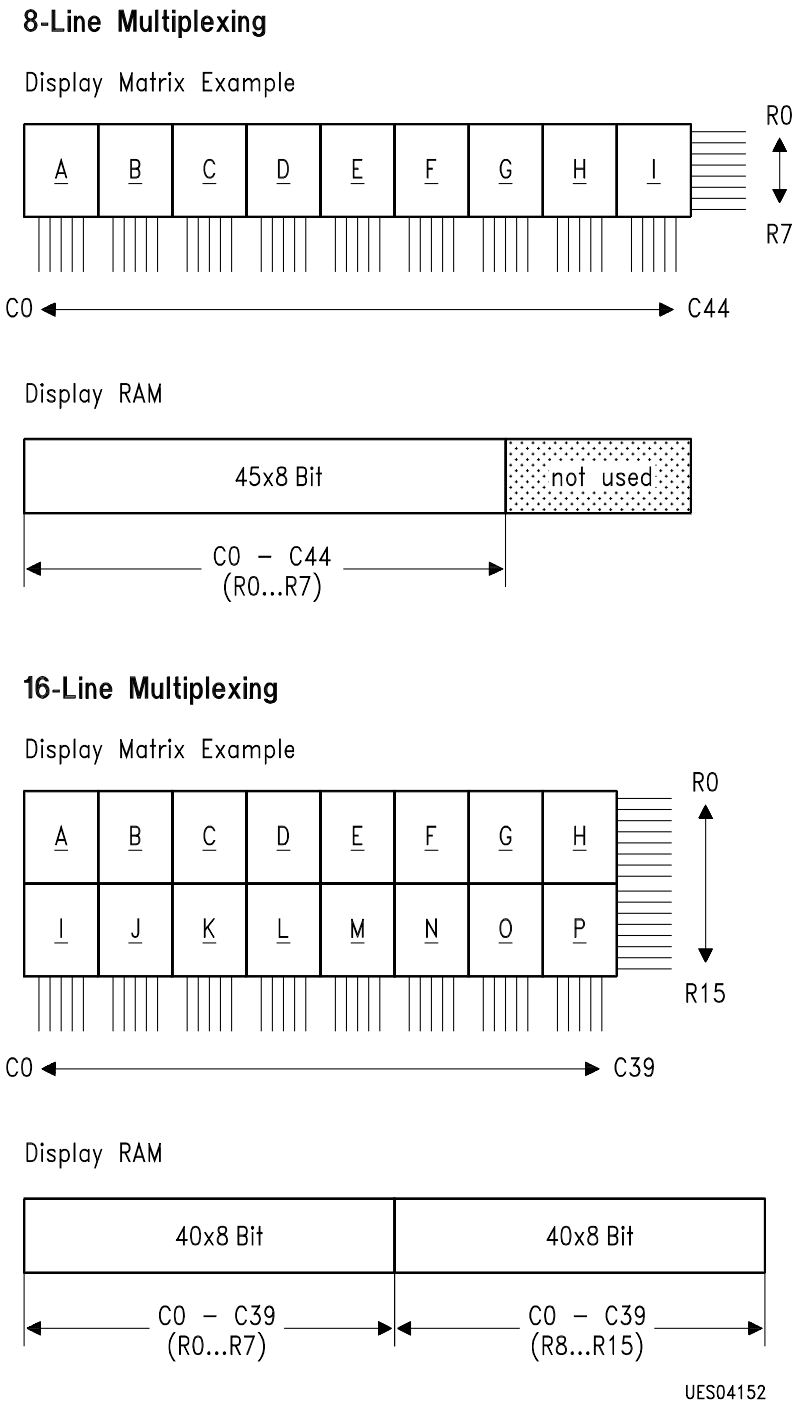


Figure 5
Multiplexing Modes and Display RAM-Configuration

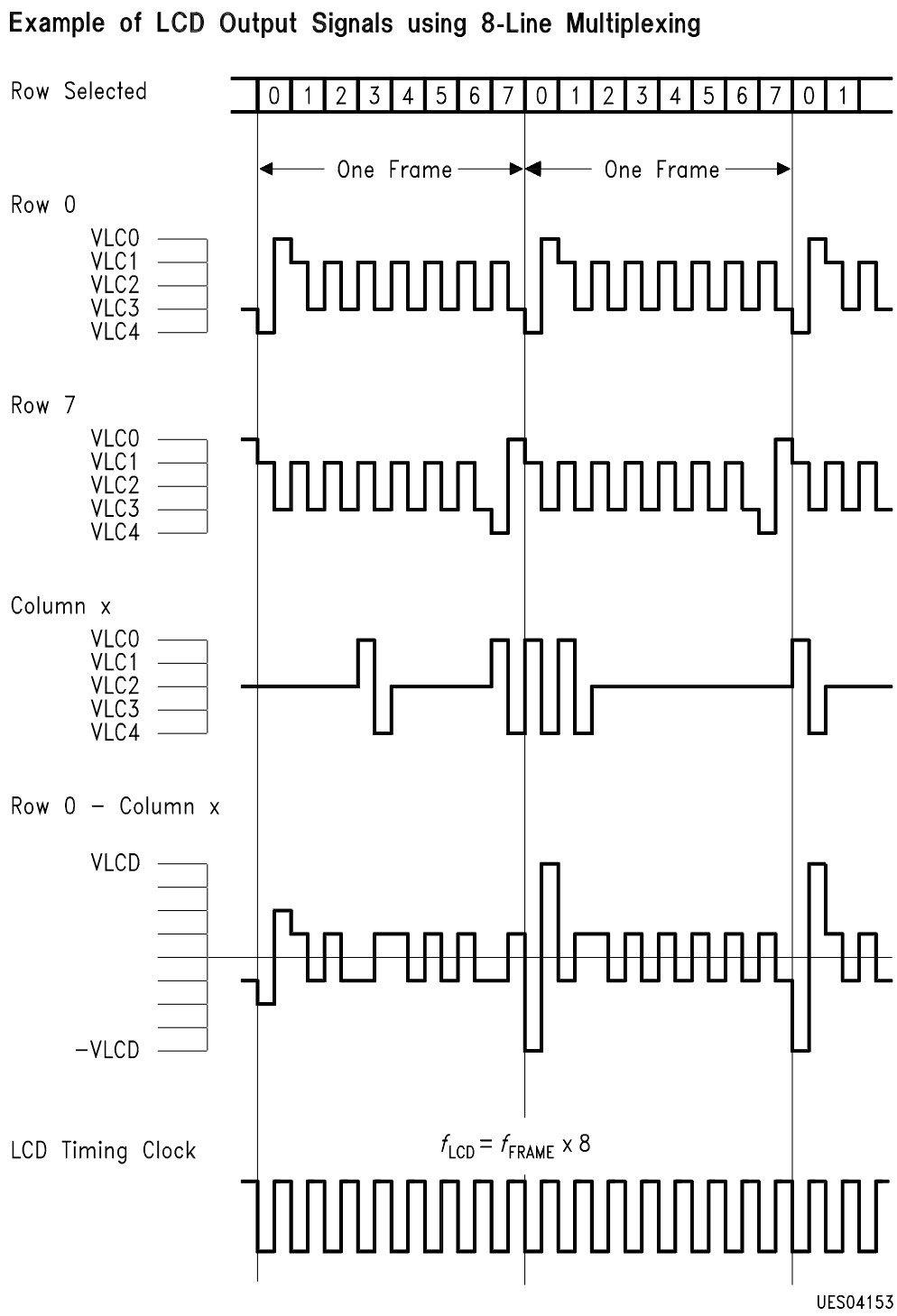


Figure 6
8-Line Multiplexing Modes and Output Signals

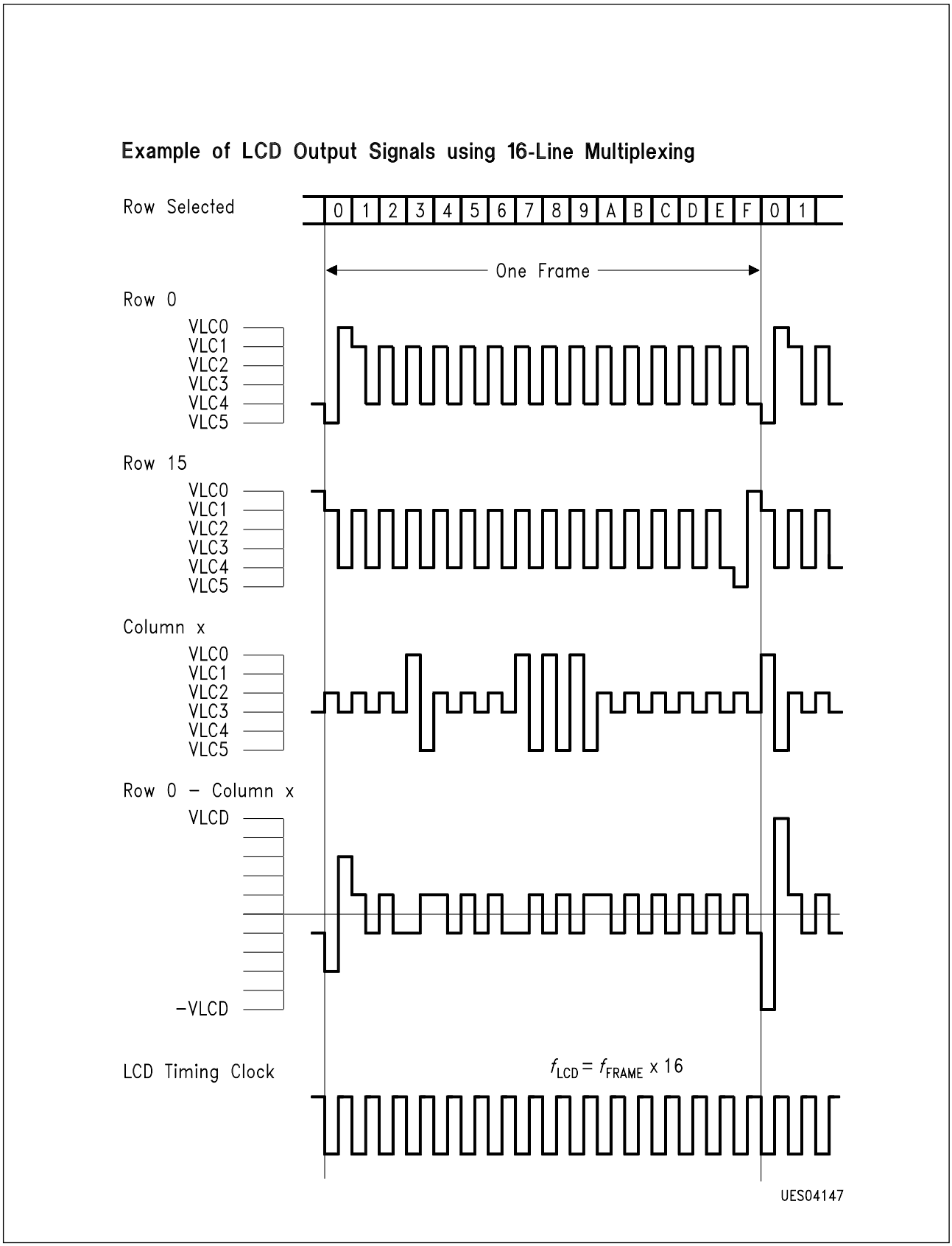


Figure 7
16-Line Multiplexing Modes and Output Signals

Row and Column Driving Voltages

The voltages for selected and not selected row and column signals for both 8-line and 16-line multiplexing are given in the following tables:

Driving Signal (8-Line)	Value
Row Selected/–	V_{LC4}
Row Selected/+	V_{LC0}
Row Not Selected/–	V_{LC1}
Row Not Selected/+	V_{LC3}
Column Selected/–	V_{LC0}
Column Selected/+	V_{LC4}
Column Not Selected/–, +	V_{LC2}

$$V_{LC0} = V_{LCIN}$$
$$V_{LC1} = V_{LCIN} \times 0.8$$
$$V_{LC2} = V_{LCIN} \times 0.5$$
$$V_{LC3} = V_{LCIN} \times 0.2$$
$$V_{LC4} = V_{SS}$$

Driving Signal (16-Line)	Value
Row Selected/–	V_{LC5}
Row Selected/+	V_{LC0}
Row Not Selected/–	V_{LC1}
Row Not Selected/+	V_{LC4}
Column Selected/–	V_{LC0}
Column Selected/+	V_{LC5}
Column Not Selected/–	V_{LC3}
Column Not Selected/+	V_{LC2}

$$V_{LC0} = V_{LCIN}$$
$$V_{LC1} = V_{LCIN} \times 0.8$$
$$V_{LC2} = V_{LCIN} \times 0.6$$
$$V_{LC3} = V_{LCIN} \times 0.4$$
$$V_{LC4} = V_{LCIN} \times 0.2$$
$$V_{LC5} = V_{SS}$$

2.9.1 LCD-Special Function Registers

MSB	SFR-Address: **BC_H**						LSB
LCDADR: Display RAM-Address Register							
...	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

ADR0 ... 6: Display RAM-Address

Valid Range:

8-line mode – 00_H ... $2C_H$
16-line mode – 00_H ... $4F_H$

LCDADR.7: reserved, to be set to 0

Default after reset : X000 0000

The address will be **incremented automatically** after each data transfer to the display RAM (during character generation or single bit / word access).

MSB	SFR-Address: **BA_H**						LSB
LCDWRD: Display RAM-Word Register							
...	WRD6	WRD5	WRD4	WRD3	WRD2	WRD1	WRD0

Default after reset : X000 0000

After the word is transferred to the display RAM, the display RAM-Address Register will be incremented once.

MSB	SFR-Address: **BB_H**						LSB
LCDBIT: Display RAM-Bit Register							
...	...	...	...	...	...	...	SB

SB: Single Bit, used to set or clear a single ‘cursor bit’ or ‘special segment bit’ in the display RAM

LCDDAT.1 ... 7: reserved, to be set to 0

Default after reset: XXXX XXX0

After the bit is transferred to the display RAM, the display RAM-Address Register will be incremented once.

MSB	SFR-Address: **B9**$_H$						LSB
LCDCOD: Display RAM-Code Register							
...	COD6	COD5	COD4	COD3	COD2	COD1	COD0

COD0 ... 6: ASCII-Character Code

LCDCOD.7: reserved, to be set to 0

Default after reset : X000 0000

Writing to this register starts a character generation procedure. This will be completed within three µs, therefore subsequent write accesses to LCDCOD and accesses to LCDADR, LCDBIT and LCDWRD are not allowed within this time.

After each of the five transfer cycles to the display RAM, the display RAM-Address Register will be incremented once.

MSB	SFR-Address: **B8**$_H$						LSB
LCDMOD: LCD-Frame Mode Register							
...	DE	MM	TCLK	...	...	DIV1	DIV0

DIV0 ... 1: Multiplex frequency division factor

DIV1/0	**Frame Frequency (Hz)**
11	61
10	81
01	122
00	244

(at 4-MHz oscillator frequency)

TCLK: used for test purposes, to be set to 0

MM: Multiplex Mode
= 0: 8-line multiplexing
= 1: 16-line multiplexing

DE: Display Enable
= 0: display disabled
(output lines switched to V_{SS})
= 1: display enabled

LCDFRM.2: reserved, to be set to 0
LCDFRM.3: reserved, to be set to 0
LCDFRM.7: reserved, to be set to 0

Default after reset: X000 XX00

This register is bit-addressable.

Character Generation

The character generator interpretes each ASCII-character code written to the LCD-code register (LCDCOD) and generates a 5×7 matrix character in the display RAM. The starting address is given in the LCD-address register (LCDADR). The address is incremented five times during the generation process, which starts after the write operation to special function register LCDCOD.

The character set is user definable and stored in a mask-programmable ROM. This ROM holds 128 character definitions. The character definitions have to be supported as an ASCII-file according to the following syntax:

$CODE = XX_H
00000
00000
00000
00000
00000
00000
00000

$CODE = yy_H
11111
11111

...

$END

Empty lines and lines beginning with '#', '/', '*' are ignored and may be used to insert comments. '$CODE =' and '$END' are reserved keywords. The character code has to be given in hexadecimal notation.

Single Bit Access

In addition to the character matrix produced by the character generator, each display RAM-field holds five bits, which may be used for underlining characters (if 8-line displays are used) or to access special segments of a user specific display (**see figure 8**). Addressed by the contents of LCDADR, each single bit can be set or cleared by setting or clearing bit SB in register LCDBIT. The address register LCDADR is incremented once after each write operation to LCDBIT. The range of valid addresses depends on the selected display mode (**see LCD-Special Function Registers**).

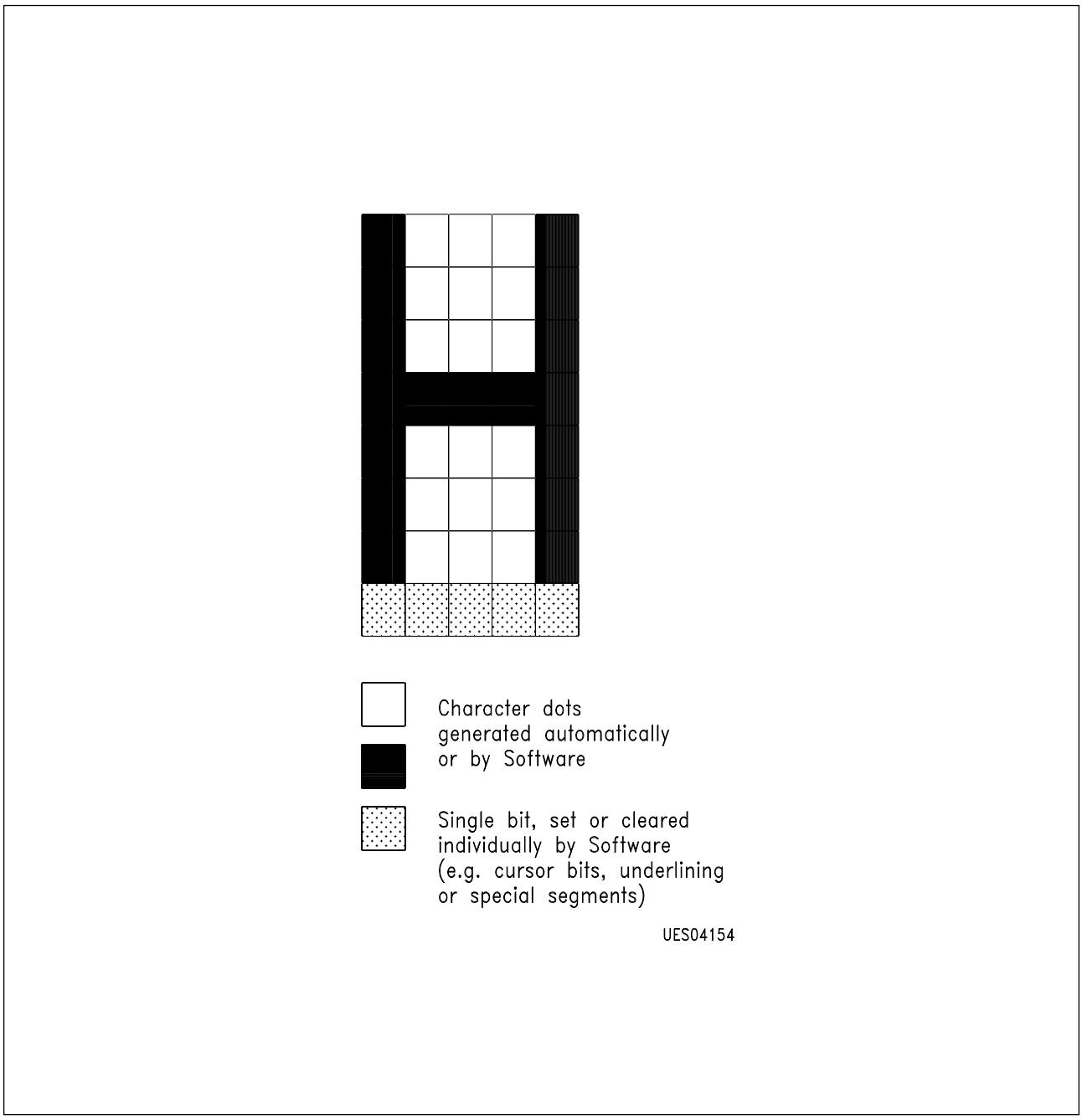


Figure 8
Display RAM-Field

Display RAM

The display RAM contains 80 × 8 bits (80 bytes). Each byte contains 7 character dots generated by the character generator or by software and one single bit for cursor representation or special segments.

The bytes of the display RAM are addressed by the contents of register LCDADR.

The valid addresses are from 00_H to $4F_H$, where locations $2D_H$ to $4F_H$ are unused in 8-line operation mode.

2.10 Instruction Set

The assembly language uses the same instruction set and the same instruction opcodes as the 8051 microcomputer family.

2.10.1 Notes on Data Addressing Modes

A	–	Accumulator
Rn	–	Working register R0–R7.
Ra	–	Working register R0 and R1 used for indirect addressing of the RAM.
adr	–	128 internal RAM-locations, any I/O-port, control or status register.
(Ra)	–	Indirect internal RAM-location addressed by register R0 or R1.
data	–	8-bit constant included in instruction.
wadr	–	16-bit constant included as bytes 2 & 3 of instruction
bit adr	–	128 software flags and bit addressable special function register.

Operations working on external data memory (MOVX ...) are not used.

2.10.2Notes on Program Addressing Modes

wadr	–	Destination address for CALL & JP may be anywhere within the program memory address space.
disp	–	PJP and all conditional jumps include an 8-bit offset byte. Range is + 127/ –128 bytes relative to first byte of the following instruction.

LCD-RAM-Addresses

LCD-fields for 8-line multiplexing:

1	2	3	4	5	6	7	8	9

LCD-fields for 16-line multiplexing:

1A	2A	3A	4A	5A	6A	7A	8A
1B	2B	3B	4B	5B	6B	7B	8B

LCDADR	Addressed LCD-Field Using 8-Line Multiplexing	Addressed LCD-Field Using 16-Line Multiplexing
00_H	1	1A
05_H	2	2A
$0A_H$	3	3A
$0F_H$	4	4A
14_H	5	5A
19_H	6	6A
$1E_H$	7	7A
23_H	8	8A
28_H	9	1B
$2D_H$	unused	2B
32_H	unused	3B
37_H	unused	4B
$3C_H$	unused	5B
41_H	unused	6B
46_H	unused	7B
$4B_H$	unused	8B

2.10.3 Instruction Set Description

Arithmetic Operations

Mnemonic		Description	Byte	Clocks
ADD	A,Rn	Add Register to Accumulator	1	5
ADD	A,adr	Add Direct Byte to Accumulator	2	8
ADD	A,(Ra)	Add Indirect RAM to Accumulator	1	6
ADD	A,data	Add Immediate Data to Accumulator	2	7
ADC	A,Rn	Add Register to Accumulator with Carry Flag	1	5
ADC	A,adr	Add Direct Byte to A with Carry Flag	2	8
ADC	A,(Ra)	Add Indirect RAM to A with Carry Flag	1	6
ADC	A,data	Add Immediate Data to A with Carry Flag	2	7
SUB	A,Rn	Subtract Register from A with Borrow	1	5
SUB	A,adr	Subtract Direct Byte from A with Borrow	2	8
SUB	A,(Ra)	Subtract Indirect RAM from A with Borrow	1	6
SUB	A,data	Subtract Immediate Data from A with Borrow	2	7
INC	A	Increment Accumulator	1	3
INC	Rn	Increment Register	1	5
INC	adr	Increment Direct Byte	2	8
INC	(Ra)	Increment Indirect RAM	1	6
INC	DP	Increment Data Pointer	1	3
DEC	A	Decrement Accumulator	1	3
DEC	Rn	Decrement Register	1	5
DEC	adr	Decrement Direct Byte	2	8
DEC	(Ra)	Decrement Indirect RAM	1	6
MUL		Multiply A&B	1	14
DIV		Divide A&B	1	17
DA		Decimal Adjust Accumulator	1	3

Logical Operations

Mnemonic		Description	Byte	Clocks
AND	A,Rn	AND Register to Accumulator	1	4
AND	A,adr	AND Direct Byte to Accumulator	2	7
AND	A,(Ra)	AND Indirect RAM to Accumulator	1	5
AND	A,data	AND Immediate Data to Accumulator	2	6
AND	adr,A	AND Accumulator to Direct Byte	2	8
AND	adr,data	AND Immediate Data to Direct Byte	3	11
OR	A,Rn	OR Register to Accumulator	1	4
OR	A,adr	OR Direct Byte to Accumulator	2	7
OR	A,(Ra)	OR Indirect RAM to Accumulator	1	5
OR	A,data	OR Immediate Data to Accumulator	2	6
OR	adr,A	OR Accumulator to Direct Byte	2	8
OR	adr,data	OR Immediate Data to Direct Byte	3	11
XOR	A,Rn	Exclusive-OR Register to Accumulator	1	4
XOR	A,adr	Exclusive-OR Direct Byte to Accumulator	2	7
XOR	A,(Ra)	Exclusive-OR Indirect RAM to Accumulator	1	5
XOR	A,data	Exclusive-OR Immediate Data to Accumulator	2	6
XOR	adr,A	Exclusive-OR Accumulator to Direct Byte	2	8
XOR	adr,data	Exclusive-OR Immediate Data to Direct Byte	3	11
CLR		Clear Accumulator	1	3
INV		Complement Accumulator	1	3
RL		Rotate Accumulator Left	1	3
RLC		Rotate Accumulator Left through the Carry Flag	1	3
RR		Rotate Accumulator Right	1	3
RRC		Rotate Accumulator Right through Carry Flag	1	3
SWAP		Swap Nibbles within the Accumulator	1	6

Data Transfer Operations

Mnemonic		Description	Byte	Clocks
MV	A,Rn	Move Register to Accumulator	1	4
MV	A,adr	Move Direct Byte to Accumulator	2	7
MV	A,(Ra)	Move Indirect RAM to Accumulator	1	5
MV	A,data	Move Immediate Data to Accumulator	2	6
MV	Rn,A	Move Accumulator to Register	1	4
MV	Rn,adr	Move Direct Byte to Register	2	9
MV	Rn,data	Move Immediate Data to Register	2	7
MV	adr,A	Move Accumulator to Direct Byte	2	7
MV	adr,Rn	Move Register to Direct Byte	2	8
MV	adr1,adr2	Move Direct Byte Adr2 to Direct Byte Adr1	3	11
MV	adr,(Ra)	Move Indirect RAM to Direct Byte	2	9
MV	adr, data	Move Immediate Data to Direct Byte	3	10
MV	(Ra),A	Move Accumulator to Indirect RAM	1	5
MV	(Ra),adr	Move Direct Byte to Indirect RAM	2	10
MV	(Ra),data	Move Immediate Data to Indirect RAM	2	8
MV	DP,wadr	Load Data Pointer with a 16-Bit Constant	3	9
MV	A,(A + DP)	Move Byte (A + DP) to Accumulator	1	7
MV	A,(A + PC)	Move Byte (A + PC) to Accumulator	1	7
MVX	A,(Ra)	Move External RAM (8-bit addr) to Accumulator	1	8
MVX	A,(DP)	Move External RAM (16-bit addr) to Accumulator	1	7
MVX	(Ra),A	Move A to External RAM (8-bit addr)	1	7
MVX	(DP),A	Move A to External RAM (16-bit addr)	1	6
PUSH	adr	Push Direct Byte onto Stack	2	9
POP	adr	Pop Direct Byte from Stack	2	9
EX	A,Rn	Exchange Register with Accumulator	1	6
EX	A,adr	Exchange Direct Byte with Accumulator	2	9
EX	A,(Ra)	Exchange Indirect RAM with Accumulator	1	7
EXNB	A,(Ra)	Exchange Low-Order Nibble of Indirect RAM with A	1	8

Boolean Variable Manipulation

Mnemonic		Description	Byte	Clocks
CLC		Clear Carry Flag	1	3
CL	bit adr	Clear Direct Bit	2	10
SETC		Set Carry Flag	1	3
SET	bit adr	Set Direct Bit	2	10
IC		Invert Carry Flag	1	4
IB	bit adr	Invert Direct Bit	2	10
AND	C,bit adr	AND Direct Bit to Carry Flag	2	11
ANDI	C,bit adr	AND Complement of Direct Bit to Carry	2	11
OR	C,bit adr	OR Direct Bit to Carry Flag	2	11
ORI	C,bit adr	OR Complement of Direct Bit to Carry	2	11
MV	C,bit adr	Move Direct Bit to Carry Flag	2	11
MV	bit adr,C	Move Carry Flag to Direct Bit	2	14

Program and Machine Control Operations

Mnemonic		Description	Byte	Clocks
PCALL	adr	Absolute Subroutine Call within a 2-Kbyte Page	2	15
CALL	wadr	Long Subroutine Call	3	16
RET		Return from Subroutine	1	9
RETI		Return from Interrupt	1	9
PJP	adr	Absolute Jump within a 2-Kbyte Page	2	11
JP	wadr	Long Jump	3	10
JR	disp	Relative Jump (relative addr)	2	6
JP	(A + DP)	Jump Indirect to (A + DP)	1	6
JZ	disp	Jump if Accumulator is Zero	2	8
JNZ	disp	Jump if Accumulator is Not Zero	2	8
JC	disp	Jump if Carry Flag is Set	2	8
JNC	disp	Jump if Carry Flag is Not Set	2	8
JB	bit adr,disp	Jump if Direct Bit is Set	3	12
JNB	bit adr,disp	Jump if Direct Bit is Not Set	3	12
JBCL	bit adr,disp	Jump if Direct Bit is Set and Clear Bit	3	15
CJD	A,adr,disp	Compare Direct to A and Jump if Not Equal	3	12
CJD	A,data,disp	Compare Immediate to A and Jump if Not Equal	3	11

Program and Machine Control Operations (cont'd)

Mnemonic		Description	Byte	Clocks
CJD	Rn,data,disp	Compare Immediate to Register and Jump if Not Equal	3	12
CJD	(Ra),data,disp	Compare Immediate to Indirect and Jump if Not Equal	3	13
DJNZ	Rn,disp	Decrement Register and Jump if Not Zero	2	9
DJNZ	adr,disp	Decrement Direct and Jump if Not Zero	3	13
NOP		No Operation	1	3

2.10.4 Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
00	1	NOP					
01	2	PJP	adr				
02	3	JP	wadr				
03	1	RR	A				
04	1	INC	A	❙			
05	2	INC	adr				
06	1	INC	(R0)				
07	1	INC	(R1)				
08	1	INC	R0				
09	1	INC	R1				
0A	1	INC	R2				
0B	1	INC	R3				
0C	1	INC	R4				
0D	1	INC	R5				
0E	1	INC	R6				
0F	1	INC	R7				
10	3	JBCL	bit adr,disp				
11	2	PCALL	adr				
12	3	CALL	wadr				
13	1	RRC		❙			❙
14	1	DEC	A	❙			
15	2	DEC	adr				
16	1	DEC	(R0)				
17	1	DEC	(R1)				
18	1	DEC	R0				
19	1	DEC	R1				
1A	1	DEC	R2				
1B	1	DEC	R3				
1C	1	DEC	R4				
1D	1	DEC	R5				
1E	1	DEC	R6				

Instruction Opcodes in Hexadecimal Order (cont'd)

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
1F	1	DEC	R7				
20	3	JB	bit adr,disp				
21	2	PJP	adr				
22	1	RET					
23	1	RL					
24	2	ADD	A,data	❙	❙	❙	❙
25	2	ADD	A,adr	❙	❙	❙	❙
26	1	ADD	A,(R0)	❙	❙	❙	❙
27	1	ADD	A,(R1)	❙	❙	❙	❙
28	1	ADD	A,R0	❙	❙	❙	❙
29	1	ADD	A,R1	❙	❙	❙	❙
2A	1	ADD	A,R2	❙	❙	❙	❙
2B	1	ADD	A,R3	❙	❙	❙	❙
2C	1	ADD	A,R4	❙	❙	❙	❙
2D	1	ADD	A,R5	❙	❙	❙	❙
2E	1	ADD	A,R6	❙	❙	❙	❙
2F	1	ADD	A,R7	❙	❙	❙	❙
30	3	JNB	bit adr,disp				
31	2	PCALL	adr				
32	1	RETI					
33	1	RLC		❙			❙
34	2	ADC	A,data	❙	❙	❙	❙
35	2	ADC	A,data adr	❙	❙	❙	❙
36	1	ADC	A,(R0)	❙	❙	❙	❙
37	1	ADC	A,(R1)	❙	❙	❙	❙
38	1	ADC	A,R0	❙	❙	❙	❙
39	1	ADC	A,R1	❙	❙	❙	❙
3A	1	ADC	A,R2	❙	❙	❙	❙
3B	1	ADC	A,R3	❙	❙	❙	❙
3C	1	ADC	A,R4	❙	❙	❙	❙
3D	1	ADC	A,R5	❙	❙	❙	❙
3E	1	ADC	A,R6	❙	❙	❙	❙
3F	1	ADC	A,R7	❙	❙	❙	❙
40	2	JC	disp				
41	2	PJP	adr				
42	2	OR	adr,A				
43	3	OR	adr,data				
44	2	OR	A,data	❙			
45	2	OR	A,adr	❙			
46	1	OR	A,(R0)	❙			
47	1	OR	A,(R1)	❙			
48	1	OR	A,R0	❙			
49	1	OR	A,R1	❙			

Instruction Opcodes in Hexadecimal Order (cont'd)

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
4A	1	OR	A,R2	▌			
4B	1	OR	A,R3	▌			
4C	1	OR	A,R4	▌			
4D	1	OR	A,R5	▌			
4E	1	OR	A,R6	▌			
4F	1	OR	A,R7	▌			
50	2	JNC	disp				
51	2	PCALL	adr				
52	2	AND	adr,A				
53	3	AND	adr,data				
54	2	AND	A,data	▌			
55	2	AND	A,adr	▌			
56	1	AND	A,(R0)	▌			
57	1	AND	A,(R1)	▌			
58	1	AND	A,R0	▌			
59	1	AND	A,R1	▌			
5A	1	AND	A,R2	▌			
5B	1	AND	A,R3	▌			
5C	1	AND	A,R4	▌			
5D	1	AND	A,R5	▌			
5E	1	AND	A,R6	▌			
5F	1	AND	A,R7	▌			
60	2	JZ	disp				
61	2	PJP	adr				
62	2	XOR	adr,A				
63	3	XOR	adr,data				
64	2	XOR	A,data	▌			
65	2	XOR	A,adr	▌			
66	1	XOR	A,(R0)	▌			
67	1	XOR	A,(R1)	▌			
68	1	XOR	A,R0	▌			
69	1	XOR	A,R1	▌			
6A	1	XOR	A,R2	▌			
6B	1	XOR	A,R3	▌			
6C	1	XOR	A,R4	▌			
6D	1	XOR	A,R5	▌			
6E	1	XOR	A,R6	▌			
6F	1	XOR	A,R7	▌			
70	2	JNZ	disp				
71	2	PCALL	adr				
72	2	OR	C,bit adr				▌
73	1	JP	(A + DP)				
74	2	MV	A,data	▌			

Instruction Opcodes in Hexadecimal Order (cont'd)

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
75	3	MV	adr,data				
76	2	MV	(R0),data				
77	2	MV	(R1),data				
78	2	MV	R0,data				
79	2	MV	R1,data				
7A	2	MV	R2,data				
7B	2	MV	R3,data				
7C	2	MV	R4,data				
7D	2	MV	R5,data				
7E	2	MV	R6,data				
7F	2	MV	R7,data				
80	2	JR	disp				
81	2	PJP	adr				
82	2	AND	C,bit adr				❙
83	1	MV	A,(A + PC)	❙			
84	1	DIV		❙	❙		❙
85	3	MV	adr1,adr2				
86	2	MV	adr,(R0)				
87	2	MV	adr,(R1)				
88	2	MV	adr,R0				
89	2	MV	adr,R1				
8A	2	MV	adr,R2				
8B	2	MV	adr,R3				
8C	2	MV	adr,R4				
8D	2	MV	adr,R5				
8E	2	MV	adr,R6				
8F	2	MV	adr,R7				
90	3	MV	DP,wadr				
91	2	PCALL	adr				
92	2	MV	bit adr,C				
93	1	MV	A,(A + DP)	❙			
94	2	SUB	A,data	❙	❙	❙	❙
95	2	SUB	A,adr	❙	❙	❙	❙
96	1	SUB	A,(R0)	❙	❙	❙	❙
97	1	SUB	A,(R1)	❙	❙	❙	❙
98	1	SUB	A,R0	❙	❙	❙	❙
99	1	SUB	A,R1	❙	❙	❙	❙
9A	1	SUB	A,R2	❙	❙	❙	❙
9B	1	SUB	A,R3	❙	❙	❙	❙
9C	1	SUB	A,R4	❙	❙	❙	❙
9D	1	SUB	A,R5	❙	❙	❙	❙
9E	1	SUB	A,R6	❙	❙	❙	❙
9F	1	SUB	A,R7	❙	❙	❙	❙

Instruction Opcodes in Hexadecimal Order (cont'd)

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
A0	2	ORI	C,bit adr				▌
A1	2	PJP	adr				
A2	2	MV	C,bit adr				▌
A3	1	INC	DP				
A4	1	MUL		▌	▌		▌
A5	1	MVM	A,(DP)	▌			
A6	2	MV	(R0),adr				
A7	2	MV	(R1),adr				
A8	2	MV	R0,adr				
A9	2	MV	R1,adr				
AA	2	MV	R2,adr				
AB	2	MV	R3,adr				
AC	2	MV	R4,adr				
AD	2	MV	R5,adr				
AE	2	MV	R6,adr				
AF	2	MV	R7,adr				
B0	2	ANDI	C,bit adr				▌
B1	2	PCALL	adr				
B2	2	IB	bit adr				
B3	1	IC					▌
B4	3	CJD	A,data,disp				▌
B5	3	CJD	A,adr,disp				▌
B6	3	CJD	(R0),data,disp				▌
B7	3	CJD	(R1),data,disp				▌
B8	3	CJD	R0,data,disp				▌
B9	3	CJD	R1,data,disp				▌
BA	3	CJD	R2,data,disp				▌
BB	3	CJD	R3,data,disp				▌
BC	3	CJD	R4,data,disp				▌
BD	3	CJD	R5,data,disp				▌
BE	3	CJD	R6,data,disp				▌
BF	3	CJD	R7,data,disp				▌
C0	2	PUSH	adr				
C1	2	PJP	adr				
C2	2	CL	bit adr				
C3	1	CLC					▌
C4	1	SWAP					
C5	2	EX	A,adr	▌			
C6	1	EX	A,(R0)	▌			
C7	1	EX	A,(R1)	▌			
C8	1	EX	A,R0	▌			
C9	1	EX	A,R1	▌			
CA	1	EX	A,R2	▌			

Instruction Opcodes in Hexadecimal Order (cont'd)

Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
CB	1	EX	A,R3	▌			
CC	1	EX	A,R4	▌			
CD	1	EX	A,R5	▌			
CE	1	EX	A,R6	▌			
CF	1	EX	A,R7	▌			
D0	2	POP	adr				
D1	2	PCALL	adr				
D2	2	SET	bit adr				
D3	1	SETC					▌
D4	1	DA		▌			▌
D5	3	DJNZ	adr,disp				
D6	1	EXNB	A,(R0)	▌			
D7	1	EXNB	A,(R1)	▌			
D8	2	DJNZ	R0,disp				
D9	2	DJNZ	R1,disp				
DA	2	DJNZ	R2,disp				
DB	2	DJNZ	R3,disp				
DC	2	DJNZ	R4,disp				
DD	2	DJNZ	R5,disp				
DE	2	DJNZ	R6,disp				
DF	2	DJNZ	R7,disp				
E0	1	MVX	A,(DP)	▌			
E1	2	PJP	disp				
E2	1	MVX	A,(R0)	▌			
E3	1	MVX	A,(R1)	▌			
E4	1	CLR		▌			
E5	2	MV	A,adr	▌			
E6	1	MV	A,(R0)	▌			
E7	1	MV	A,(R1)	▌			
E8	1	MV	A,R0	▌			
E9	1	MV	A,R1	▌			
EA	1	MV	A,R2	▌			
EB	1	MV	A,R3	▌			
EC	1	MV	A,R4	▌			
ED	1	MV	A,R5	▌			
EE	1	MV	A,R6	▌			
EF	1	MV	A,R7	▌			
F0	1	MVX	(DP),A				
F1	2	PCALL	adr				
F2	1	MVX	(R0),A				
F3	1	MVX	(R1),A				
F4	1	INV		▌			
F5	2	MV	adr,A				

3 Electrical Characteristics

3.1 Absolute Maximum Ratings

Parameter	Symbol	Limit Values	Unit
Voltage on any pin other than V_{LCIN} with respect to ground (V_{SS})	V	– 0.5 to 6	V
Voltage on pin V_{LCIN} with respect to ground (V_{SS})	V_{LCIN}	11	V*
Power dissipation	P_{tot}	1	W
Ambient temperature under bias	T_A	– 20 to 85	°C
Storage temperature	T_{stg}	– 65 to 125	°C

*) >10 V affects device reliability!

3.2 DC-Characteristics

V_{DD} = 5 V 10 %, V_{SS} = 0 V; T_A = – 20 to 85 °C

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Power supply voltage	V_{DD}	4.5	5.0	5.5	V	
Power supply current	I_{DD}			25	mA	f_{sys} = 12 MHz, all outputs disconnected
LCD-supply voltage	V_{LCIN}	4.0		V_{DD} +5.0	V	
LCD-supply current	I_{LCIN}		15		mA	LCD-unit enabled
L-port input current	I_{IL1}			1	µA	
H-port input current	I_{IH1}	– 1			µA	
L-port input voltage	V_{IL1}	V_{SS}		$0.3 \times V_{DD}$	V	
L-reset input voltage	V_{IL2}	V_{SS}		0.8	V	
H-port input voltage	V_{IH1}	$0.7 \times V_{DD}$			V	
H-reset input voltage	V_{IH2}	2.0		V_{DD}	V	
L-port output voltage	V_{OL1}	V_{SS}		0.4	V	
H-port output voltage	V_{OH1}	V_{DD} – 0.4		V_{DD}	V	
Analog input voltage	V_{AI}	V_{AGND}		V_{AREF}	V	
Analog ground voltage	V_{AGND}	V_{SS}		$0.5 \times V_{DD}$	V	
Analog reference voltage	V_{AREF}	$0.5 \times V_{DD}$		V_{DD}	V	
Analog supply difference volt.	V_{ADELTA}	$0.5 \times V_{DD}$		V_{DD}	V	
Analog ground supply current	I_{AGND}		– 20	– 50	µA	V_{AGND} = 0.0 V
Analog refer. supply current	I_{AREF}		20	50	µA	V_{AREF} = 5.0 V
Analog input current	I_{ANA}	– 100		100	nA	V_{AREF} = 5.0 V V_{AGND} = 0.0 V
DC-offset voltage row/column	V_{RCDC}			100	mV	V_{LCIN} = 10 V
LCD-reference supply current	I_{LCIN}			tbd.	mA	V_{LCIN} = 10 V
ADC-differential non-linearity	$DNLE$			0.5	LSB	$R_S < 1$ kΩ*
ADC-offset error	QE			0.5	LSB	$R_S < 1$ kΩ*
ADC-gain error	GE			0.5	LSB	$R_S < 1$ kΩ*
Analog input capacitance	C_{ANA}			70	pF	

*) R_S = analog voltage source output resistance

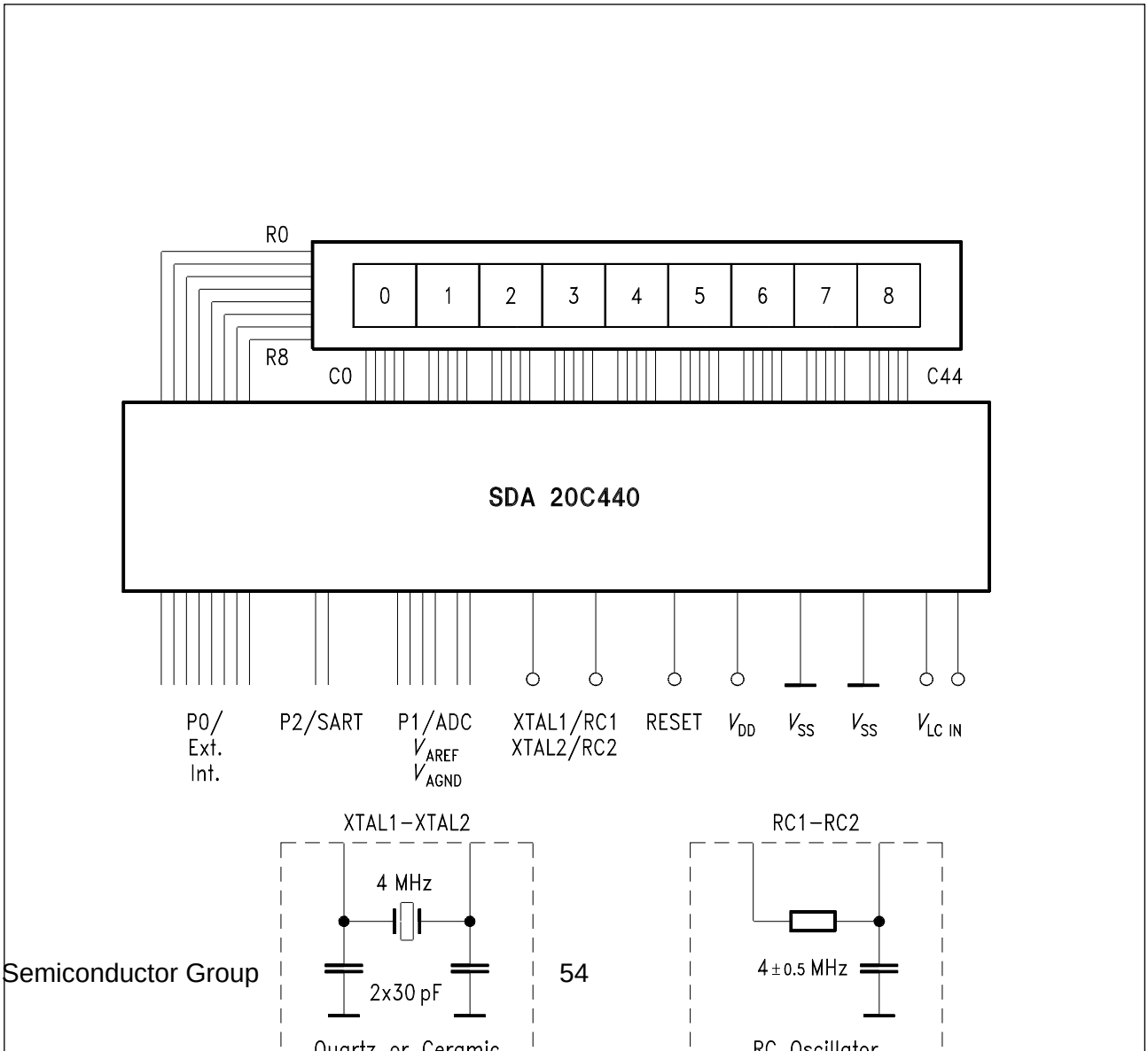
3.3 AC-Characteristics

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Oscillator frequency	f_{OSC}	3.5	4	4.5	MHz	
ADC-conversion time	T_C		17		µs	f_{sys} = 12 MHz
Power-on reset time	t_{POR}	20			ms	
Reset pulse width	t_{RES}	4			µs	

Instruction Opcodes in Hexadecimal Order (cont'd)

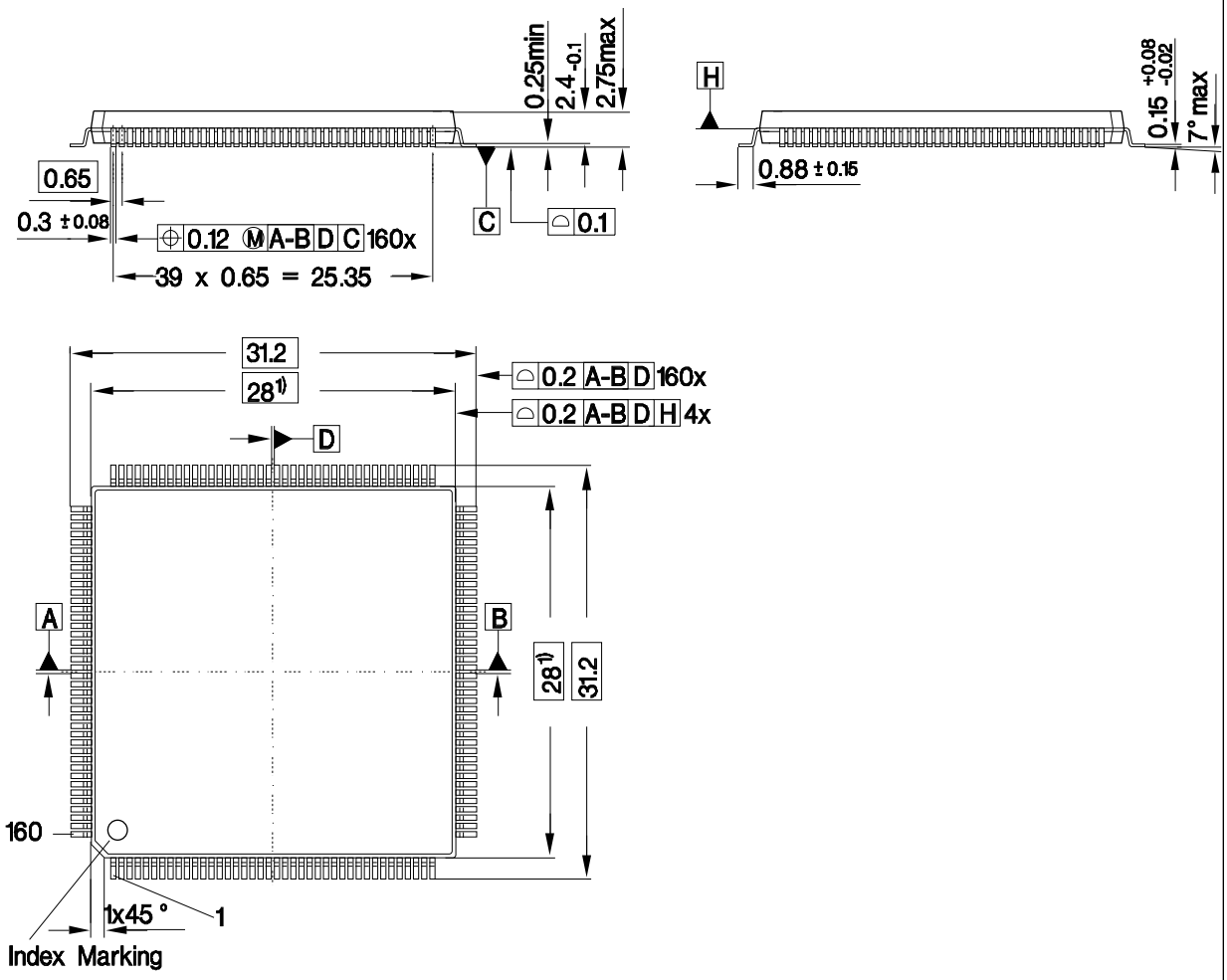
Hex Code	Number of Bytes	Mnemonic	Operands	Flags P	OV	AC	C
F6	1	MV	(R0),A				
F7	1	MV	(R1),A				
F8	1	MV	R0,A				
F9	1	MV	R1,A				
FA	1	MV	R2,A				
FB	1	MV	R3,A				
FC	1	MV	R4,A				
FD	1	MV	R5,A				
FE	1	MV	R6,A				
FF	1	MV	R7,A				

4 Application Example



5 Package Outlines

Plastic Package, M-QFP-80 (SMD)
(Plastic Metric Quad Flat Package))



1) Does not include plastic or metal protrusion of 0.25 max. per side

Sorts of Packing
Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”
SMD = Surface Mounted Device

Dimensions in mm