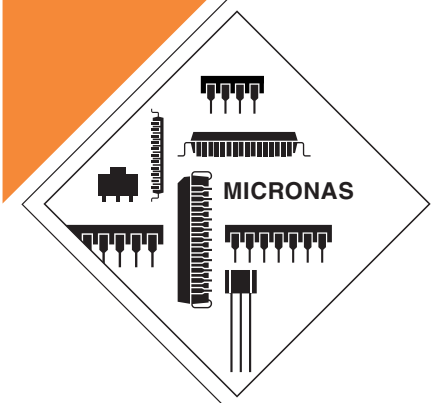


DATA SHEET

SDA 55xx
TVText Pro

Edition Sept. 10, 2004
6251-556-3DS

MICRONAS

Contents

Contents, continued

Contents, continued

Contents, continued

Contents, continued

Contents, continued

TVText Pro

Release Note: Revision bars indicate significant changes to the previous edition.

1. Introduction

The Micronas SDA 55xx TV microcontroller is dedicated to 8 bit applications for TV control and provides dedicated graphic features designed for modern low class to mid range TV sets.

The SDA 55xx is a microcontroller and single chip teletext decoder for decoding World System Teletext data as well as other data services as Video Programming System (VPS), Program Delivery Control (PDC), and Wide Screen Signalling (WSS) data used for PAL plus transmissions (in line 23). The data slicer and display part of the SDA 55xx supports a wide range of TV standards including PAL, NTSC as well as the acquisition of the above mention data services as VPS, WSS, PDC, TTX and Closed Caption data.

The slicer combined with its dedicated hardware stores TTX data in a VBI buffer of 1 kByte. The Microcontroller firmware available from Micronas performs all the acquisition tasks (hamming and parity checks, page search and evaluation of header control bits) once per field. Additionally, the firmware can provide high end teletext features like Packet-26-handling, FLOF, TOP and list page mode. The Application Program Interface (API) to the user software is optimized for a minimum SW overhead.

The on-chip display unit used to display teletext data up to level 1.5 can also be used for customer defined on-screen displays (OSD). The display generator is able to handle parallel display attributes, pixel oriented displays and dynamically re-definable characters (DRCS).

The SDA 55xx provides also an integrated general-purpose, fully 8051-compatible microcontroller with specific hardware features especially suitable in TV sets. The microcontroller core has been enhanced to provide powerful features such as memory banking, data pointers and additional interrupts, etc.

The internal XRAM consists of up to 16 kBytes. The microcontroller provides an internal ROM of up to 128 kBytes. ROMless versions can access up to 1 MByte of external RAM and ROM.

The 8-bit microcontroller runs at 33.33 MHz internal clock. SDA 55xx is realized in 0.25 micron technology with 2.5 V supply voltage for the core and 3.3 V for the I/O port pins to make them TTL compatible.

Based on the SDA 55xx microcontroller the MINTS software package was developed and provides dedicated device drivers for many Micronas video & audio products and includes a full blown TV control SW for the PEPER application chassis. The SDA 55xx is also supported with powerful design tools like emulators from Hitex, Kleinhenz, iSystems, the Keil C51 Compiler and TEDIpro OSD development SW by Tara Systems. This support provided by Micronas leads to:

– Shorter time to market
– Re-usability of the SW also for future Micronas products
– Target independent SW development based on ANSI C.
– Verification and validation of SW before targeting and improved SW test concept
– Graphical interface design requiring a minimum effort for OSD programming and TV controlled know how.
– Complete, modular and open tool chain available and configurable by customer.

1.1. General Features

– 8051 compatible microcontroller with TV related special features and advanced OSD display
– Feature selection via special function register
– Simultaneous processing of TTX, VPS, PDC and WSS (line 23) data
– Supply voltage 2.5 V for core and 3.3 V for ports
– ROM version package PSDIP52-2, PMQFP64-1
– Romless version package PMQFP100-2
– 128 kByte Flash ROM version package PSDIP52-2

1.1.1. External Crystal and Programmable Clock Speed

– Normal mode 33.33 MHz CPU clock, power save mode 8.33 MHz
– CPU clock speed selectable via special function registers.
– Single external 6 MHz crystal, all necessary clock signals are generated internally by means of PLLs

1.1.2. Microcontroller Features

– 8-bit 8051 instruction set compatible CPU
– Two 16-bit timers
– Watchdog timer
– Capture compare timer for infrared remote control decoding
– Pulse width modulation unit (2 channels 14 bit, 6 channels 8 bit)

– ADC (4 channels, 8 bit)
– UART

1.1.3. Memory

– Non-multiplexed 8-bit data and 16…20-bit address bus (ROMless version)
– Memory banking up to 1 MByte (ROMless version)
– Up to 128 kByte on-chip program ROM
– Eight 16-bit data pointer registers (DPTR)
– 256-bytes on-chip processor internal RAM (IRAM)
– 128 bytes extended stack memory
– Display RAM and TXT/VPS/PDC/WSS Data Acquisition Buffer directly accessible via MOVX command
– Up to 16 kByte on-chip extended RAM (XRAM) consisting of
  • 1 kByte on-chip ACQ buffer RAM (access via MOVX)
  • 1 kByte on-chip extended RAM (XRAM, access via MOVX) for user software
  • 3 kByte display memory

1.1.4. Display Features

– ROM character set supports all east and west European languages in a single device
– Mosaic graphic character set
– Parallel display attributes
– Single/double width/height of characters
– Variable flash rate
– Programmable screen size (25 rows × 33 … 64 columns)
– Flexible character matrixes (H x V) 12 x 9 … 16
– Up to 256 dynamically re-definable characters in standard mode; 1024 dynamically re-definable characters in enhanced mode
– CLUT with up to 4096 color combinations
– Up to 16 colors per DRCS character
– One out of eight colors for foreground and background colors for 1-bit DRCS and ROM characters
– Shadowing & contrast reduction
– Pixel by pixel shiftable cursor with up to 4 different colors
– Support of progressive and 100 Hz double scan
– 3 × 4 bits RGB-DACs on chip
– Free programmable pixel clock from 10 MHz to 32 MHz
– Pixel clock independent from CPU clock
– Multinorm H/V-display synchronization in master or slave mode

1.1.5. Acquisition Features

– Multistandard digital data slicer
– Parallel multinorm slicing (TTX, VPS, WSS, CC, G+)
– Four different framing codes available
– Data caption only limited by available memory
– Programmable VBI-buffer
– Full channel data slicing supported
– Fully digital signal processing
– Noise measurement and controlled noise compensation
– Attenuation measurement and compensation
– Group delay measurement and compensation
– Exact decoding of echo disturbed signals

1.1.6. Ports

– One 8-bit I/O-port with open drain output and optional I^2C bus emulation support (Port 0)
– Two 8-bit multifunction I/O-ports (Port 1, Port 3)
– One 4-bit port working as digital or analog inputs for the ADC (Port 2)
– One 2-bit I/O-port with secondary functions (P4.2, 4.3, 4.7)
– One 4-bit I/O-port with secondary function (P4.0, 4.1, 4.4) Not available in PSDIP52-2)

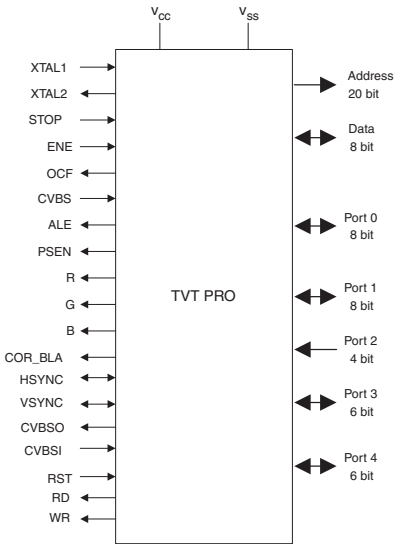


Fig. 1–1: Logic Symbol

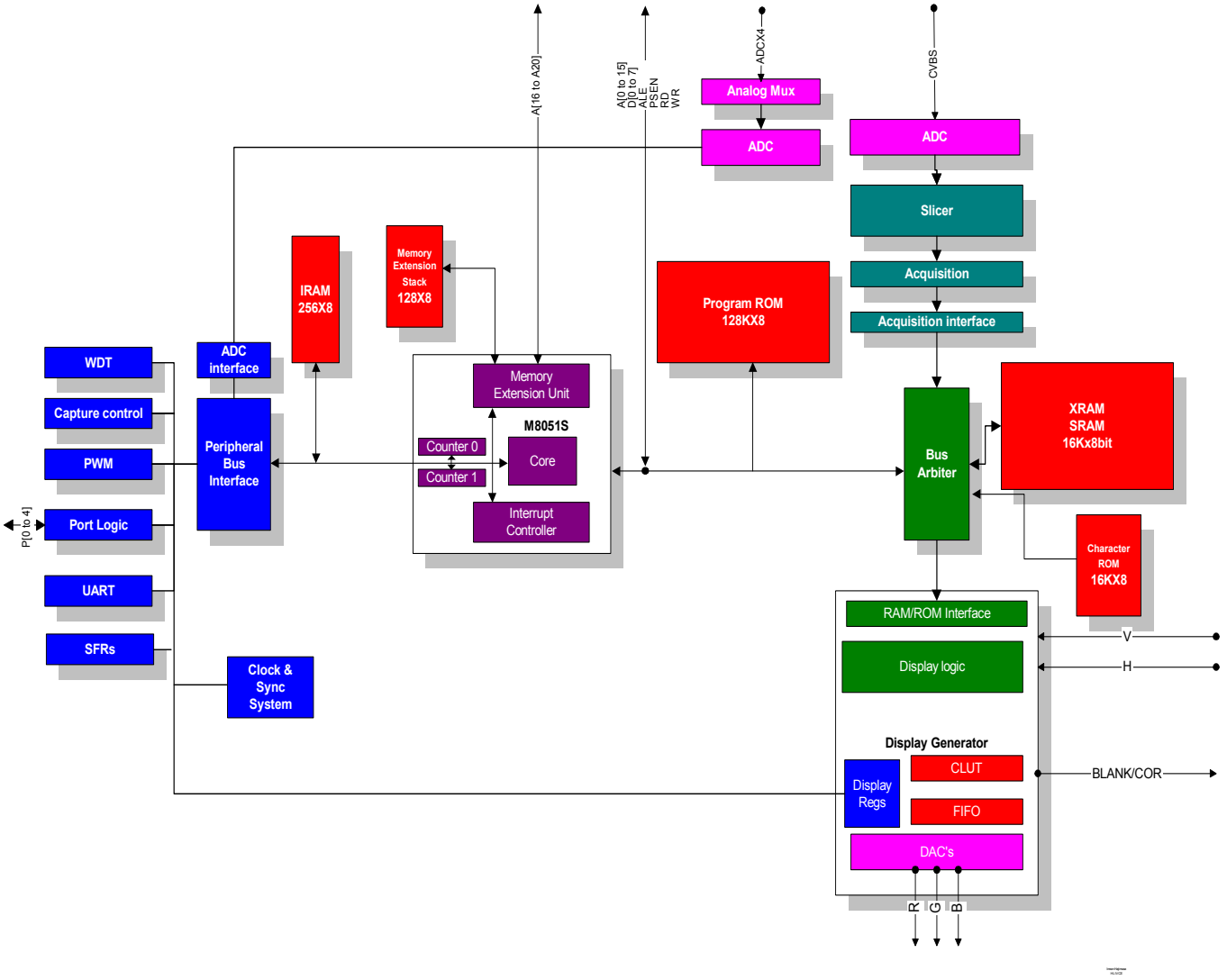


Fig. 1–2: Block Diagram

1.2. Overview of Current Versions and Packages for SDA 55xx

Table 1–1: TVText Pro versions and packages overview

Version	Type	Package
SDA 5550M	– ROMless version – 16 kByte RAM	PMQFP100-1
SDA 5550	– ROMless version – 16 kByte RAM	PLCC84-1
SDA 555xFL	– 128 kByte Flash memory on chip (re-programmable) – 16 kByte RAM	PSDIP52-2
SDA 555x x = 1...5	– 32-128 kByte user ROM – 8-16 kByte RAM	PMQFP64-1, PSDIP52-1, PSDIP52-2 See note
SDA 5521	– OSD-only version – 32 kByte user ROM on chip – 8 kByte RAM	PSDIP52-1, PSDIP52-2 See note
SDA 5522	– OSD-only version – 64 kByte user ROM on chip – 8 kByte RAM	PSDIP52-1, PSDIP52-2 See note
SDA 5523	– OSD-only version – 64 kByte user ROM on chip – 16 kByte RAM	PSDIP52-1, PSDIP52-2 See note
SDA 5525	– OSD only version – 128 kByte user ROM on chip – 16 kByte RAM	PSDIP52-1, PSDIP52-2 See note
SDA 5577	– Standalone co-processor for teletext reception, decoding, and display – 10 pages – ROM fix-programmed with the software P116	PSDIP52-1, PSDIP52-2 See note
Note: Micronas delivers two types of PSDIP52 packages (-1, -2). The packages have slightly different outline dimensions, but are considered identical. See Outline Dimensions for PSDIP52-1 Package on page 144 and Outline Dimensions for PSDIP52-1 Package on page 144. For logistics reasons, the customer **cannot** choose the package to be delivered.		

2. Functional Description

2.1. Clock System

2.1.1. General Function

The on-chip clock generator provides the TVTpro with its basic clock signal. The oscillator runs with an external crystal and the appropriate internal oscillator circuitry (see Fig. on page 174).

For applications with lower timing accuracy requirements (and if the RTC is not used) an external ceramic resonator can be used. The usage of a ceramic resonator is not recommended for Teletext applications as depending on the absolute tolerance of the ceramic resonator the data slicer may not work correctly. Additional this might also require that display timing parameters and the baud rate prescaler have to be adapted. In timing critical applications the horizontal frequency of the incoming CVBS signal can be used to measure the actual timing deviation and to re-program the clock PLL.

The 6 MHz clock signal is used to generate the internal 300 MHz display reference clock by means of an on-chip phase locked loop (PLL). The PLL can be bypassed to reduce the power consumption. If an immediate wake up from power down is not required the PLL can also be switched off in this mode.

From the output frequency of the main clock PLL two clock systems are derived.

2.1.2. System Clock

The 33.33 MHz system clock (f_{CPU}) is provided to the microcontroller core, all microcontroller related peripherals, the sync timing logic, the A/D converters, the slicer, the display generator and the color lookup tables CLUT.

It is possible to use 8.33 MHz (1/4 of 33.33 MHz) for the system clock domain (slow down mode). Setting SFR-bit PLLS = 1 the user is able to send the PLL into a power save mode.

Note: Before the PLL is switched to power save mode (PLLS = 1), the software has to switch the clock source from 200 MHz PLL clock to the 3 MHz oscillator clock (SFR bit CLK_src = 1). In this mode the slicer, acquisition, DAC and display generator are switched off.

To switch back to full frequency operation, the software has to end the PLL power save mode (SFR-bit PLLS = 0), reset the PLL for 10 µs (3 machine cycles, SFR bit PLL_res = '1', then '0' again), wait for 150 µs (38 machine cycles) and switch back to the PLL clock (SFR-bit SCR_src = 0).

If the power down mode is activated, PLL and oscillator are send to sleep mode (SFR bit PDS = 1). Furthermore, there are additional possibilities to disable the clocks for the peripherals - See Section 2.3.17. on page 44.

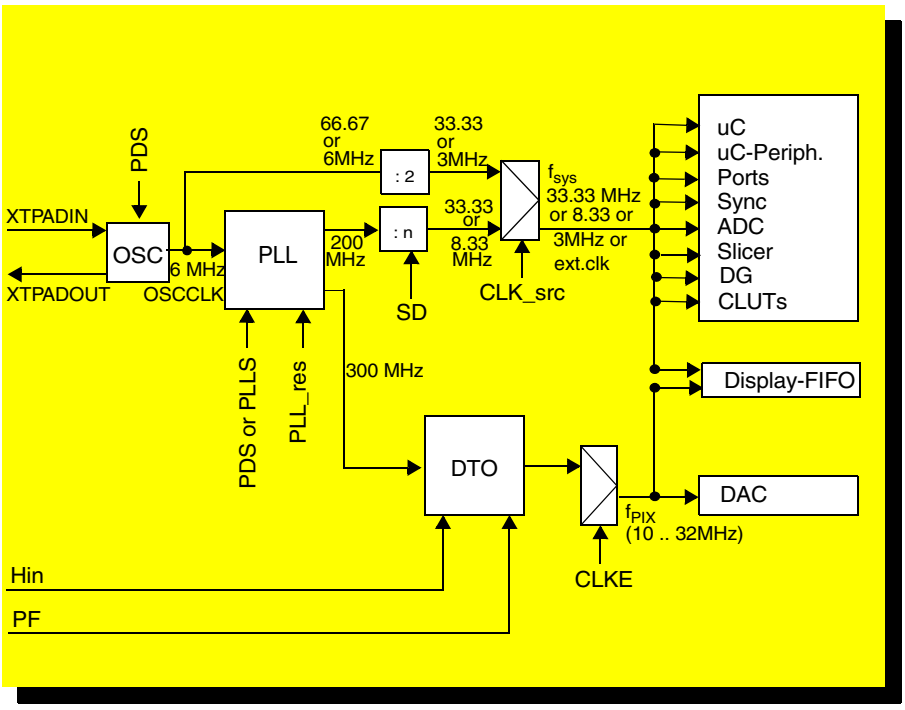


Fig. 2–1: Clock System of TVText Pro

2.1.3. Pixel Clock

The second clock system is the pixel clock (f_{PIX}), which is programmable in a range from 10 … 32 MHz. It serves the output part of the display FIFO and the D/A converters. The pixel clock is derived from the high frequent output of the PLL and line by line phase shifted to the positive edge of the horizontal sync signal (normal polarity). Because the final display clock is derived from a DTO (digital time oscillator) it has no equidistant clock periods although the average frequency is exact. This pixel clock generation system has several advantages:

– The frequency of the pixel clock can be programmed independently from the horizontal line period.

– Because the input of the PLL is already a signal with a relative high frequency, the resulting pixel frequency has an extremely low jitter.

– The resulting pixel clock follows the edge of the H-sync impulse without any delay and has always the same quality than the sync timing of the deflection controller.

2.1.4. Related Registers

Table 2–1: Related registers and bits

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
PCLK1						PF[10:8]		
PCLK0	PF[7:0]							
PCON	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE
PSAVEX						CLK_src	PLL_res	PLLS
See Section 3. on page 110 for detailed register description.								

2.2. Slicer and Data Acquisition

2.2.1. General Function

TVTPro provides a full digital data slicer including digital H- and V-sync separation and digital sync processing. The acquisition interface is capable to process all known data services transmitted in the Vertical Blanking Interval VBI of a CVBS signal (Teletext, VPS, CC, G+, WSS). Four different framing codes (two of them freely programmable for each field) are available for each field. Digital signal processing algorithms are applied to compensate various disturbing influences as there are:

– Noise measurement and compensation.

– Attenuation measurement and compensation.

– Group delay measurement and compensation.

Note: TVTPro is optimized for precise data clock recovery and error free reception of data. Thus, the reception of data services is widely unaffected by noise and the actual transmission channel characteristics.

The CVBS input contains an on chip clamping circuit. The integrated A/D converter has a 7 bit resolution. The sampling frequency is 33.33 MHz.

The sliced data are synchronized to the data clock frequency given by the clock-run-in. The framing code will define the start of the data stream. The resulting valid data will be written to the VBI data buffer. After line 23 is received an interrupt will be issued to the microcontroller. The microcontroller starts processing the buffered data. That means, a SW module will check the data for errors and store them in an assigned memory area.

To improve the data signal quality the slicer control logic generates horizontal and vertical windows during which the reception of the framing code is allowed. The framing code can be programmed individually for each line, so that in each line a different data service can be received. For VPS and WSS the framing code is hardwired. All following acquisition tasks are performed by the internal controller, so in principal the data of any data service can be acquired.

2.2.2. Slicer Architecture

The slicer consists of three main blocks:

– The slicer

– The H/V synchronization for the slicer

– The acquisition interface

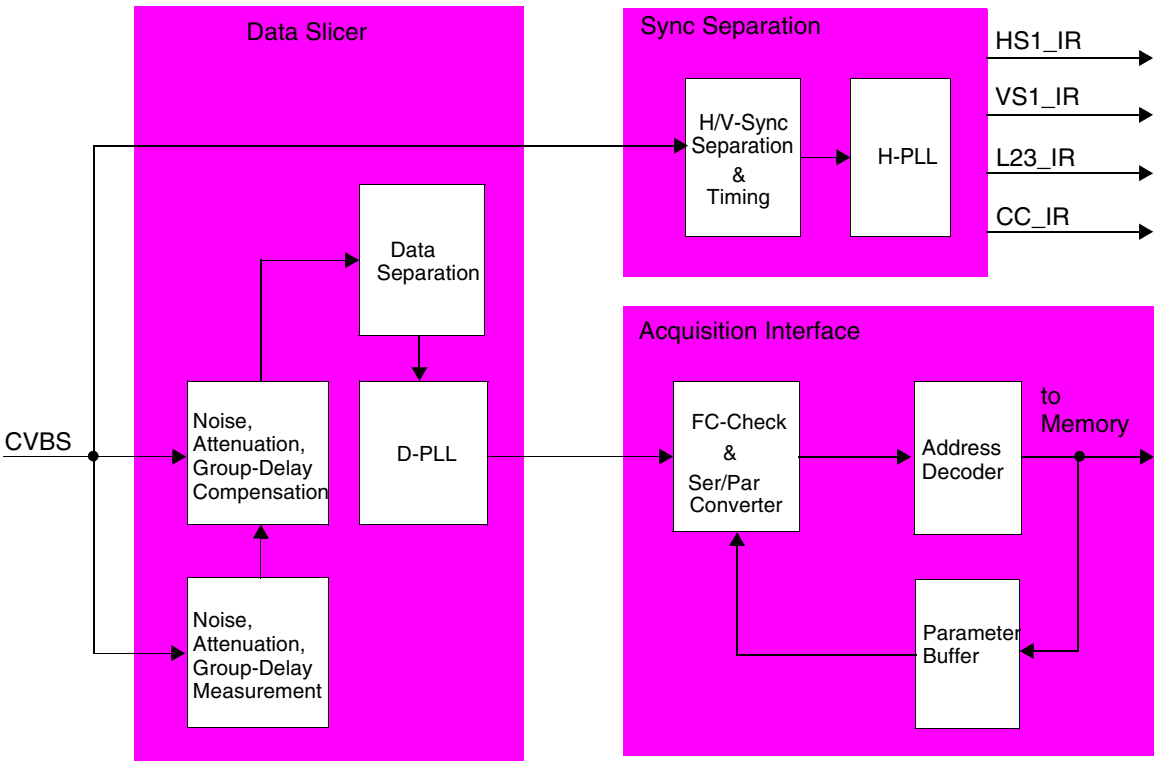


Fig. 2–2: Block Diagram of Digital Slicer and Acquisition Interface

2.2.2.1. Distortion Processing

After A/D conversion the digital CVBS bit stream is applied to internal circuitry which corrects the input signal for distortions created in the transmission channel. In order to apply the right algorithm for the correction, a signal measurement is done in parallel. This measurement unit can detect the following distortions.

2.2.2.1.1. Noise

The noise measurement unit incorporates two different algorithms. Both algorithms use the value between two equalizing pulses, which corresponds to the black level. As the system knows the black level, a window is placed between this two equalizing pulses (located in line 4).

The first algorithm compares successive the amplitude samples inside that window. The difference between these samples is measured and a flag is set as soon as this difference over several TV lines is greater than a specified value. This algorithm is able to detect higher frequency noise (e.g. white noise).

The second algorithm measures the difference between the black value and the actual sampled value inside this window. As soon as this difference over several TV lines is greater than a specified value a second flag is set. This algorithm is sensitive against low frequency noise as it is known from co-channel distortion.

Both flags can be used to optimize the response of the compensation circuits in order to achieve best reception performance.

2.2.2.1.2. Frequency Attenuation

During signal transmission the CVBS signal can severely be attenuated. This attenuation normally is frequency depending. That means that the higher the frequency the stronger the attenuation. As the clock run-in (from now on referred to as CRI) for teletext represents the highest possible frequency (3.5 MHz) it can be used to measure the attenuation. Only strong negative attenuation causes problems during data slicing. A flag is needed to notify highly negative attenuation to the SW. If this flag is set a special peaking filter is switched on in the data-path.

2.2.2.1.3. Group Delay

Quite often the data stream is corrupted because of group delay distortion introduced by the transmission channel. The teletext framing code ($E4_H$) is used as a measurement reference. The delay of the edges inside this code can be used to measure the group delay distortion. The measurement is done during every teletext line and filtered over several lines.

It can be detected whether the signal has positive, negative or no group delay distortions. Two flags are set accordingly. By means of these two flags, an all-pass contained in the compensation circuit is configured to compensate positive or negative group delay.

All of the above mentioned filters can be individually be disabled, set to forced mode, or automatic mode via control registers.

2.2.2.2. Data Separation

Parallel to signal analysis and distortion compensation another filter is calculating the required slicing level. The slicing level is the mean value of the clock run-in CRI. As teletext is coded using the NRZ format, the slicing level can not be calculated outside the CRI timing window and is therefore frozen after CRI. Using the found slicing level the data are sliced from the digitized CVBS signal. The result is a stream of zeros and ones. In order to find the logical zeros and ones which have been transmitted, the data clock needs to be recovered. Therefore during the CRI timing window a digital data PLL (D-PLL) is synchronized to the transitions in the sliced data stream which represent the original data clock. The frequency of the D-PLL is also frozen after the CRI timing window.

Timing information to freeze the slicing level, the D-PLL and to control other actions are generated by the timing circuit. It generates also all control signals which have to be synchronized to the data start.

2.2.3. H/V-Synchronization

Data slicer and acquisition interface require different control signals which have to be synchronized to the incoming CVBS (e.g. line number, field sequence or line start of a TV line). Therefore a slicing level for the sync pulses is calculated and the sync signal is sliced from the filtered digital CVBS signal.

Using a digital integrator vertical and horizontal sync pulses are separated. The horizontal pulses are fed into a digital H-PLL which has flywheel functionality. The H-PLL includes a counter which is used to generate all the necessary horizontal control signals. The vertical sync pulse is used to synchronize the line counter, which generates the required vertical control signals.

The synchronization block includes a watchdog for supervision of the actual lock condition of the H-PLL. The watchdog can produce an interrupt (CC_IR) if synchronization has been lost. It could therefore be an indication for a channel change or missing input signal.

2.2.4. Acquisition Interface

The acquisition interface manages the data transfer from between slicer and memory. From slicer to memory first of all a bit synchronization is performed (Framing Code (FC) check). Following this, the data is serial/parallel converted. 8 bit wide words will be shifted into the memory. The data acquisition supports several features. The FC check is able to handle four different framing codes for one field. Two of this framing codes are programmable and could therefore be changed from field to field. The acquisition can be switched from normal mode (line 6 to 23) to full channel mode (line 6 to the end of a field).

In the other direction parameters are loaded from the memory to the slicer. This parameter down loading takes place after the vertical sync and after the horizontal sync. These parameters are used for the slicer configuration.

2.2.4.1. Framing Code Check

There are four Framing Codes FC implemented which are compared with the FC of the incoming signal.

– The first one is 8-bit wide and is loaded down with the field parameters.

– The second one is 16-bit wide and fixed to the FC of VPS.

– The third one is 16-bit wide as well, but can be loaded with the field parameters. If the third one is used, the user can specify not only the FC but also a “don’t-care” mask.

– The fourth FC is reserved for WSS. The actual FC can be changed line by line.

2.2.4.1.1. Framing Code FC1

This FC should be used for all services with 8-bit framing codes (e.g. for TTX). The actual framing code is loaded down each field. The check can be done without any bit error tolerance or with a tolerance of one bit.

2.2.4.1.2. Framing Code FCVPS

This FC is fixed to that of VPS. Only an error-free signal will enable the reception of the VPS data line.

Note: If VPS should be sliced in field 1 and TTX in field 2, the appropriate line parameters for line 16 have to be changed dynamically from field to field.

2.2.4.1.3. Framing Code FC3

This 16-bit FC is loaded with the field parameters as well as a “don’t care” mask. The incoming signal is compared with both, the framing code and the “don’t care” mask. Further reception is enabled if all bits which are not “don’t care” match the incoming data stream.

2.2.4.1.4. Framing Code FCWSS

This FC is pre-programmed to that of WSS. Only an error-free signal will enable the reception of the WSS data line.

2.2.4.1.5. FC Check Select

There is a two bit line parameter called FCSEL. By means of this parameter the user is able to select which FC check is used for the actual line. If NORM is set to WSS the WSS FC check is used independently of FCSEL.

2.2.4.2. Interrupts

Some events which occur inside the slicer, sync separation or acquisition interface should cause an interrupt. They are summarized in register **CISR0** and **CISR1**. The slicer hardware sets the related interrupt flag which must be reset by the application software before the next interrupt can be accepted.

2.2.4.3. VBI Buffer and Memory Organization

The implemented SW has to provide configuration parameters for the slicer and the acquisition interface. Both circuits will produce status information for the CPU.

Some of these parameters and status bits are constant during the duration of a field. Those parameters are called field parameters. They are downloaded after the vertical sync.

Other parameters and status bits may change from line to line (e.g. data service depending values). Those parameters are called line parameters. They are downloaded after each horizontal sync impulse.

The start address of the VBI buffer can be configured with a special function register ‘**STRVBI**’. 9 Bytes are needed for the field parameter. 47 Bytes should be reserved for every sliced data line. If 18 lines of data (in full channel mode 314) have been send to memory no further data acquisition will take place until the next vertical pulse appears and the H-PLL is still locked.

That means if at least 855 Bytes (14767 Bytes in full channel mode) are reserved for the VBI buffer size in the RAM no VBI overflow will occur. The controller can start or stop the VBI data acquisition using bit ‘ACQON’ of register **STRVBI**. The acquisition is stopped as soon as this bit is changed to ‘0’. If the bit is changed back to ‘1’ the acquisition starts again with the next V-pulse (only if STAB = 1). The start address (Bit 3 … 0 of register **STRVBI**) of the VBI buffer should only be changed if the acquisition is switched off.

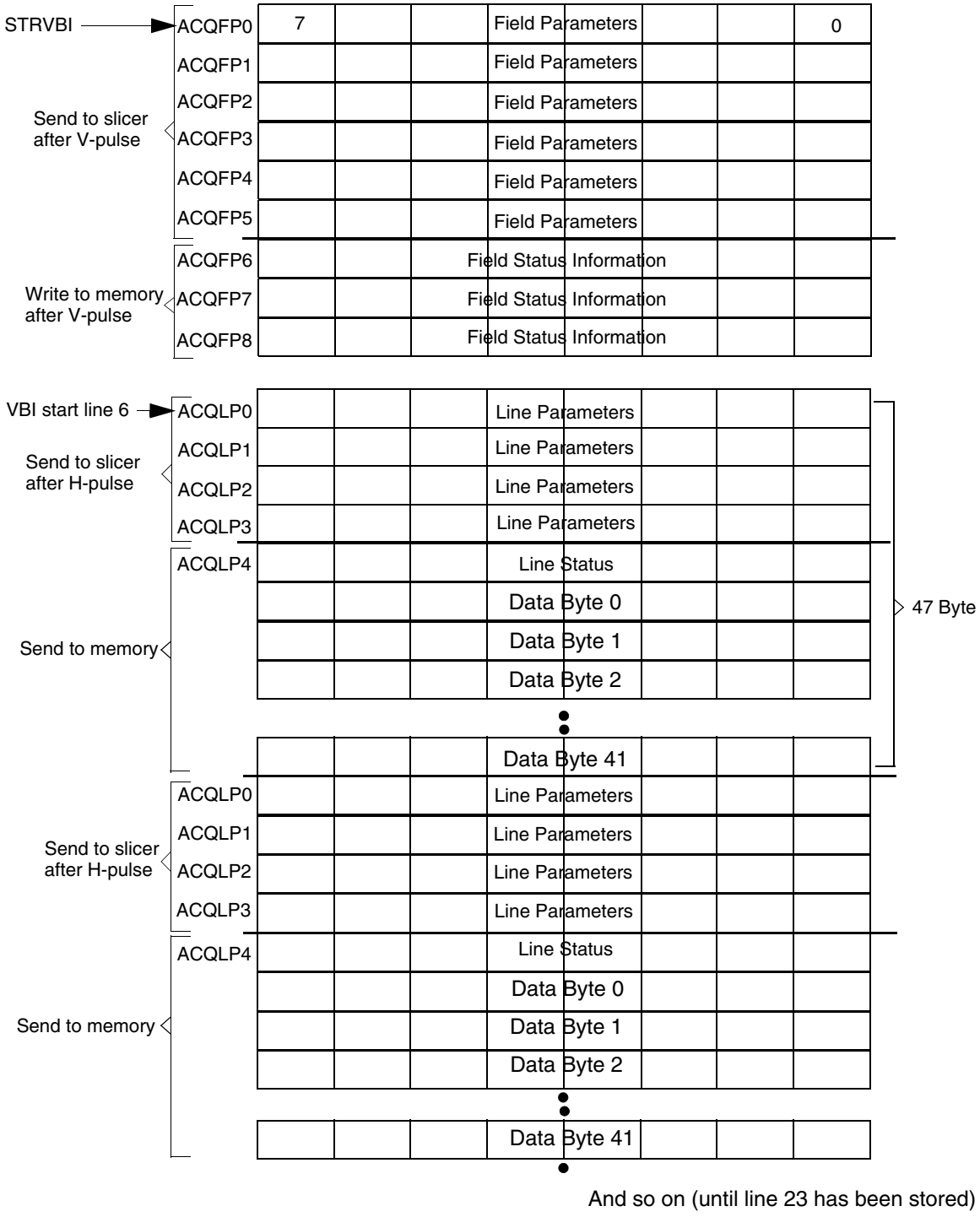


Fig. 2–3: VBI Buffer: General Structure

2.2.5. Related Registers

The acquisition interface has only three SFR Registers. The line and field parameters are stored in the RAM (RAM registers). They have to be initialized by software before starting the data acquisition.

Table 2–2: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STRVBI	ACQON	Reserved	ACQSTA		VBIADR			
CISR0 bit addressable	**L24**	ADC	WTmr	**AVS**	DVS	PWtmr	**AHS**	DHS
CISR1 bit addressable	**CC**	ADW					IEX[1:0]	
See Section 3. on page 110 for detailed register description.								

2.2.5.1. RAM Registers

See Section 3. on page 110 for detailed register description.

2.2.5.1.1. Field Parameters

All field parameters are updated once in a field. That means the status information written from the acquisition interface to the memory represents only a snapshot of the status.

Table 2–3: Field parameters

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ACQFP0	FC3[15:8]							
ACQFP1	FC3[7:0]							
ACQFP2	FC3MASK[15:8]							
ACQFP3	FC3MASK[7:0]							
ACQFP4	FC1[7:0]							
ACQFP5	AGDON	AFRON	ANOON	GDPON	GDNON	FREON	NOION	FULL
ACQFP6	NOISE(0)	FREATTF	STAB	VDOK	FIELD	NOISE(1)	GRDON	GRDSIGN
ACQFP7					LEOFLI[11:8]			
ACQFP8	LEOFLI[7:0]							
See Section 3. on page 110 for detailed register description.								

2.2.5.1.2. Line Parameters

Table 2–4: Line parameters

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
ACQLP0	DINCR[15:8]							
ACQLP1	DINCR[7:0]							
ACQLP2	NORM[2:0]			FCSEL[1:0]		FC1ER	VCR	Reserved
ACQLP3	MLENGTH[2:0]			ALENGTH[1:09		CLKDIV[2:0]		
ACQLP4	PERR[5:0]						TLDE	FCOK
See Section 3. on page 110 for detailed register description.								

2.2.6. Recommended Parameter Settings

Table 2–5: Recommended parameter settings

	TTX	VPS	WSS	CC	G+
AGDON	1	0	0	0	0
AFRON	1	0	0	0	0
ANOON	1	1	1	1	1
GDPON	0	0	0	0	0
GDNON	0	0	0	0	0
FREON	0	0	0	0	0
NOION	0	0	0	0	0
DINCR	54559	39321	39321	7920	7920
FC1E	0	0	0	0	0
MLENGTH	1	2	7	7	7
ALENGTH	2	2	2	2	2
CLKDIV	0	0	2	5	5
NORM	0	2	3	4	5
FCSEL	0	1	2	2	2
VCR	0	0	0	0	0
MATCH	0	0	0	0	0
FC1	228	don’t care	don’t care	don’t care	don’t care
FC3	don’t care	don’t care	don’t care	3	1261
FC3MASK	don’t care	don’t care	don’t care	65472	63488

2.2.7. Microcontroller

2.2.8. Architecture

Every CPU machine cycle consists of 12 internal CPU clock periods.

The CPU manipulates operands in two memory spaces: The program memory space, and the data memory space. The program memory address space is provided to accommodate relocatable code.

The data memory address space is divided into the 256-Byte internal data RAM, XRAM (extended data memory, accessible with MOVX instructions) and the 128-Byte Special Function Register (SFR) address space.

Four register banks (each bank has eight registers), 128 addressable bits, and the stack reside in the internal data RAM. The stack depth is limited only by the available internal data RAM. Its location is determined by the 8-bit stack pointer. All registers except the program counter and the four 8-register banks reside in the special function register address space.

These memory mapped registers include arithmetic registers, pointers, I/O-ports, registers for the interrupt system, timers, pulse width modulator, capture control unit, watchdog timer, UART, display, acquisition control etc. Many locations in the SFR address space are bitwise addressable.

Note: Reading from unused locations within data memory will yield undefined data.

Conditional branches are performed relative to the 16 bit program counter. The register indirect jump permits branching relative to a 16-bit base register with an offset provided by an 8-bit index register. Sixteen-bit jumps and calls permit branching to any location in the memory address space.

The microcontroller has five methods for addressing source operands: Register, direct, register-indirect, immediate, and base register plus index register-indirect addressing.

The first three methods can be used for addressing destination operands. Most instructions have a “destination, source” field that specifies the data type, addressing methods and operands involved. For operations other than moves, the destination operand is also a source operand.

Registers in the four 8-register banks can be accessed through register, direct, or register-indirect addressing. The lower 128 Bytes of internal data RAM can be accessed through direct or register-indirect addressing, the upper 128 Bytes of internal data RAM through register-indirect addressing; and the special function registers through direct addressing. Look-up tables resident in program memory can be accessed through base register plus index register-indirect addressing.

2.2.8.1. CPU Hardware

2.2.8.1.1. Instruction Decoder

Each program instruction is decoded by the instruction decoder. This unit generates the internal signals that control the functions of each unit within the CPU section. These signals control the sources and destination of data, as well as the function of the Arithmetic/Logic Unit (ALU).

2.2.8.1.2. Program Control Section

The program control section controls the sequence in which the instructions stored in the program memory are executed. The conditional branch logic enables conditions internal and external to the microcontroller to cause a change in the sequence of program execution. The 16-bit program counter holds the address of the instruction to be executed. It is manipulated with the control transfer instructions listed in Section 2.2.10..

2.2.8.1.3. Internal Data RAM

The internal data RAM provides a 256-Byte scratch pad memory, which includes four register banks and 128 direct addressable software flags. Each register bank contains registers R0 … R7. The addressable flags are located in the 16-Byte locations starting at Byte address 20_H and ending with Byte location $2F_H$ of the RAM address space.

In addition to this standard internal data RAM the microcontroller contains an extended internal RAM. It can be considered as a part of an external data memory. It is referenced by MOVX instructions (MOVX A, @DPTR), the memory organization is explained in Section 2.5. on page 47.

2.2.8.1.4. Arithmetic/Logic Unit (ALU)

The arithmetic section of the microcontroller performs many data manipulation functions and includes the Arithmetic/Logic Unit (ALU) and the **ACC**, **B**, and **PSW** registers. The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under the control of the instruction decoder. The ALU performs the arithmetic operations of add, subtract, multiply, divide, increment, decrement, BCD-decimal-add adjust and compare, and the logic operations like and, or,

exclusive-or, complement and rotate (right, left, or nibble swap).

The register **ACC** is the accumulator, the register **B** is dedicated during multiply and divide and serves as both source and destination. During all other operations the register **B** is simply another location of the special function register space and may be used for any purpose.

2.2.8.1.5. Boolean Processor

The Boolean processor is an integral part of the microcontroller architecture. It is an independent bit processor with its own instruction set, its own accumulator (the carry flag) and its own bit-addressable RAM and I/O. The bit manipulation instructions allow the direct addressing of 128 bits within the internal data RAM and several bits within the special function registers. The special function registers which have addresses exactly divisible by eight contain directly addressable bits.

The Boolean processor can perform, on any addressable bit, the bit operations of “set”, “clear”, “complement”, “jump-if-set”, “jump-if-not-set”, “jump-if-set then-clear” and “move to/from carry”. Between any addressable bit (or its complement) and the carry flag it can perform the bit operation of logical AND or logical OR with the result returned to the carry flag.

2.2.8.1.6. Program Status Word Register (PSW)

The **PSW** flag bits record microcontroller status information and control the operation of the microcontroller. 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 register **ACC**. F0 and F1 are general purpose flags which are pushed onto the stack as part of a **PSW** save (see Table 2–7).

The two register bank select bits (RS1 and RS0) determine which one of the four register banks is selected as show in Table 2–6.

See Section 3. on page 110 for detailed register description.

2.2.8.1.7. 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 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 256 Bytes.

Note: For memory above 64k, the memory extension stack is used, refer to Section 2.5.3. on page 49.

Table 2–6: Register banks

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 … $1F_H$

Table 2–7: Related register

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
PSW	CY	AC	F0	RS[1:0]		OV	F1	P
See Section 3. on page 110 for detailed register description.								

2.2.8.1.8. Data Pointer Register (DPTR).

Table 2–8: Related registers

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
DPL	DPL[7:0]							
DPH	DPH[7:0]							
DPSEL						DPSEL[2:0]		
See Section 3. on page 110 for detailed register description.								

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 and to access the extended data memory. **DPTR** may be manipulated as one 16-bit register or as two independent 8-bit registers **DPL** and **DPH**.

Eight data pointer registers are available, the active one is selected by a special function register (**DPSEL**)

2.2.8.2. CPU Timing

Timing generation is completely self-contained, except for the frequency reference which can be a crystal or external clock source. The on-board oscillator is a parallel anti-resonant circuit. The XTAL2 pin is the output of a high-gain amplifier, while XTAL1 is its input. A crystal connected between XTAL1 and XTAL2 provides the feedback and phase shift required for oscillation.

In slowdown mode, the microcontroller runs at one fourth the normal frequency. This mode is useful when power consumption needs to be reduced. Slow down mode is entered by setting the bit SD in **PCON** register.

Note: Any slow-down mode should only be used if teletext reception and the display are disabled. Otherwise processing of the incoming text data might be incomplete and the display structure will be corrupted. For disabling acquisition and display generator see Section 2.3.17.

2.2.8.3. Addressing Modes

There are five general addressing modes operating on Bytes. One of these five addressing modes, however, operates on both Bytes and bits:

– Register
– Direct (both Bytes and bits)
– Register indirect
– Immediate
– Indirect, using base register plus index-register

The following list summarizes, which memory spaces may be accessed by each of the addressing modes:

Register Addressing

R0 … R7
ACC, **B**, CY (bit), DPTR

Direct Addressing

RAM (low part)
Special Function Registers

Register-indirect Addressing

RAM (@R1, @R0, **SP**)

Immediate Addressing

Program Memory

Base Register plus Index-Register Indirect Addressing

Program Memory (@DPTR + A, @PC + A)

2.2.8.3.1. Register Addressing

Register addressing accesses the eight working registers (R0 … R7) of the selected register bank. The **PSW** register flags RS1 and RS0 determine which register bank is enabled. The least significant three bits of the instruction opcode indicate which register is to be used. **ACC**, **B**, DPTR and CY, the Boolean processor accumulator, can also be addressed as registers.

2.2.8.3.2. Direct Addressing

Direct Byte addressing specifies an on-chip RAM location (only low part) or a special function register. Direct addressing is the only method of accessing the special function registers. An additional Byte is appended to the instruction opcode to provide the memory location address. The highest order bit of this Byte selects one of two groups of addresses: Values between 00_H … $7F_H$ access internal RAM locations, while values between 80_H … $0FF_H$ access one of the special function registers.

2.2.8.3.3. Register Indirect Addressing

Register indirect addressing uses the contents of either R0 or R1 (in the selected register bank) as a pointer to locations in the 256 Bytes of internal RAM. Note that the special function registers are not accessible by this method.

Execution of PUSH and POP instructions also use register-indirect addressing. The stack pointer may reside anywhere in internal RAM.

2.2.8.3.4. Immediate Addressing

Immediate addressing allows constants to be part of the opcode instruction in program memory.

An additional Byte is appended to the instruction to hold the source variable. In the assembly language and instruction set, a number sign (#) precedes the value to be used, which may refer to a constant, an expression, or a symbolic name.

2.2.8.3.5. Base Register plus Index Register Indirect Addressing

Base register plus index register indirect addressing allows a Byte to be accessed from program memory via an indirect move from the location whose address is the sum of a base register (DPTR or PC) and index register, **ACC**. This mode facilitates accessing to look-up table resident in program memory.

2.2.9. Ports and I/O-Pins

There are 34 Port pins available, out of which 24 are I/O pins configured as three 8-bit wide ports P0, P1, and P3. Port 4 consists of 6 I/O bits, out of which only 3 are available in the PSDIP52-2 package. All 6 port pins are only available in the other packages with higher pin count. Each pin can be individually and independently programmed as input or output and each can be configured dynamically. One 4-bit-port P2 is input only.

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 microcontroller's electrical specifications 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 logic level at the pin/pins.

Pins of a single port can be individually configured as inputs and outputs by writing a ‘one’ to each pin that is to be an input. Each time an instruction uses a complete port as destination, the SW has to make sure that ‘ones’ are written to those bits that correspond to the pins used as inputs. An external input signal to a port pin needs not to be synchronized to the internal clock.

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/Bytes are INC, DEC, CPL, JBC, SETB, CLR, MOV P.X, CJNE, DJNZ, ANL, ORL, and XRL. The data read by these instructions 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.

Port 0 has an open-drain output. Writing a ‘one’ to the bit latch leaves the output transistor off, so the pin floats.

In that condition it can be used as a high-impedance input. Port 0 is considered ‘true bidirectional’, because when configured as an input it floats.

Ports 1, 3 and 4 have ‘quasi-bidirectional’ output drivers.

In ports P1, P3 and P4 the output drivers provide source current for one system clock period 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 input,

from sourcing current into the external device that is driving the input pin.

It is not allowed to drive Port 3.6 to logic low level while reset state changes from the active to inactive state otherwise a special test mode is activated.

Secondary functions can be selected individually and independently for the pins of Port 1 and 3. Further information on Port 1's secondary functions is given in Section 2.9. on page 59. P3 generates the secondary control signals automatically as long as the pin corresponding to the appropriate signal is programmed as an input, i. e. if the corresponding bit latch in the P3 special function register contains a ‘one’.

Table 2–9: Ports and I/O-pins

Port	I/O	Default Function	Alternate Function 2		Alternate Function 3	
			Toggle	Function	Toggle	Function
			Control bit	Function	Control bit	Function
P0(0…7)	I/O	Port pin	–	–	–	–
P1(0)	I/O	Port pin	PWME(E0)	PWM 8 bit channel 0	–	–
P1(1)	I/O	Port pin	PWME(E1)	PWM 8 bit channel 1	–	–
P1(2)	I/O	Port pin	PWME(E2)	PWM 8 bit channel 2	–	–
P1(3)	I/O	Port pin	PWME(E3)	PWM 8 bit channel 3	–	–
P1(4)	I/O	Port pin	PWME(E4)	PWM 8 bit channel 4	–	–
P1(5)	I/O	Port pin	PWME(E5)	PWM 8 bit channel 5	–	–
P1(6)	I/O	Port pin	PWME(E6)	PWM 14 bit channel 0	–	–
P1(7)	I/O	Port pin	PWME(E7)	PWM 14 bit channel 1	–	–
P2(0)	I	Port pin	CADCCO(AD0)	ADC channel 0	–	–
P2(1)	I	Port pin	CADCCO(AD1)	ADC channel 1	–	–
P2(2)	I	Port pin	CADCCO(AD2)	ADC channel 2	–	–
P2(3)	I	Port pin	CADCCO(AD3)	ADC channel 3	–	–
P3(0)	I/O	Port pin	CSCR0(O_E_$\overline{\text{P3_0}}$)	ODD/Even indicator	–	–
P3(1)	I/O	Port pin	Port input mode	External extra Int 0	Port output mode	TXD
P3(2)	I/O	Port pin	Port input mode	External interrupt 0	–	–
P3(3)	I/O	Port pin	Port input mode	External interrupt 1	–	–
P3(4)	I/O	Port pin	Port input mode	Timer/counter 0 input	–	–
P3(5)	I/O	Port pin	Port input mode	Timer/counter 1 input	–	–
P3(6)	I/O	Port pin	–	–	–	–
P3(7)	I/O	Port pin	Port input mode	External extra Int 1	Port input mode	RXD
P4(0)[1)]	I/O	A17	CSCR1($\overline{\text{A17}}$_P4_0)	Port pin	–	–

[1)] Not available in PSDIP52-2

Table 2–9: Ports and I/O-pins, continued

Port	I/O	Default Function	Alternate Function 2		Alternate Function 3	
			Toggle	Function	Toggle	Function
			Control bit	Function	Control bit	Function
P4(1)[1)]	I/O	A18	CSCR1($\overline{A18}$_P4_1)	Port pin	–	–
P4(2)	I/O	Port pin	CSCR1(ENARW)	Read signal	–	–
P4(3)	I/O	Port pin	CSCR1(ENARW)	Write signal	–	–
P4(4)[1)]	I/O	A19	CSCR1($\overline{A19}$_P4_4)	Port pin	–	–
P4(7)	I/O	Port/VS in	CSCR0(VS_$\overline{OE}$, $\overline{P4_7}$_ALT)	VS output	CSCR0 (VS_$\overline{OE}$, $\overline{P4_7}$_ALT)	OddEven output

[1)] Not available in PSDIP52-2

2.2.9.1. Read Modify Write Feature

'Read-modify-write' commands are instructions that read a value, possibly change it, and then rewrite it to the latch. The read-modify-write instructions are listed in Table 2–10.

If the destination operand is a port or a port bit, these instructions read the information stored in the latch rather than the status of the pin. The read-modify-write instructions are directed to the latch rather than to 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. If a 'one' is written to the bit, the transistor is turned on.

If the CPU would read back the status of the same port bit from the pin rather than the latch, it would read the base-emitter voltage of the transistor and interpret it as a logic '0'. Reading the latch rather than the pin will return the correct value of logic '1'.

Table 2–10: Read-modify-write instructions

Mnemonic	Description	Example
ANL	Logical AND	ANL P1, A
ORL	Logical OR	ORL P2, A
XRL	Logical EX – OR	XRL P3, A
JBC	Jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	Complement bit	CPL P3.0
INC	Increment	INC P1
DEC	Decrement	DEC P1
DJNZ	Decrement and jump if not zero	DJNZ P3, LABEL
MOV PX.Y, C[1)]	Move carry bit to bit Y of Port X	MOV P1.7, C
CLR PX.Y[1)]	Clear bit Y of Port X	CLR P2.6
SET PX.Y[1)]	Set bit Y of Port X	SET P3.5

[1)] The instruction reads the port Byte (all 8 bits), modifies the addressed bit, then writes the new Byte back to the latch.

2.2.10. Instruction Set

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

2.2.10.1. Notes on Data Addressing Modes

Rn – Working register R0 - R7.

direct – 128 internal RAM-locations, any I/O-port, control or status register.

@Ri – 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 & 3 of instruction.

bit – 128 software flags, any I/O-pin, control or status bit in special function registers.

Operations working on external data memory (MOVX …) are used to access the extended internal data RAM (XRAM).

2.2.10.2. Notes on Program Addressing Modes

addr 16 – Destination address for LCALL & LJMP may be anywhere within the program memory address space.

addr 11 – Destination address for ACALL & AJMP will be within the same 2 KByte of the following instruction.

rel – SJMP and all conditional jumps include an 8-bit offset Byte. Range is +127/-128 Bytes relative to first Byte of the following instruction.

2.2.10.3. Instruction Set Description

Table 2–11: Arithmetic operations

Mnemonic		Description	Byte
ADD	A, Rn	Add register to Accumulator	1
ADD	A, direct	Add direct Byte to Accumulator	2
ADD	A, @Ri	Add indirect RAM to Accumulator	1
ADD	A, #data	Add immediate data to Accumulator	2
ADDC	A, Rn	Add register to Accumulator with Carry flag	1
ADDC	A, direct	Add direct Byte to A with Carry flag	2
ADDC	A, @Ri	Add indirect RAM to A with Carry flag	1
ADDC	A, #data	Add immediate data to A with Carry flag	2
SUBB	A, Rn	Subtract register from A with Borrow	1
SUBB	A, direct	Subtract direct Byte from A with Borrow	2
SUBB	A, @Ri	Subtract indirect RAM from A with Borrow	1
SUBB	A, #data	Subtract immediate data from A with Borrow	2
INC	A	Increment Accumulator	1
INC	Rn	Increment register	1
INC	direct	Increment direct Byte	2
INC	@Ri	Increment indirect RAM	1
DEC	A	Decrement Accumulator	1
DEC	Rn	Decrement register	1
DEC	direct	Decrement direct Byte	2
DEC	@Ri	Decrement indirect RAM	1
INC	DPTR	Increment Data Pointer	1
MUL	AB	Multiply A & B	1
DIV	AB	Divide A & B	1
DA	A	Decimal Adjust Accumulator	1

Table 2–12: Logical operations

Mnemonic		Description	Byte
ANL	A, Rn	AND register to Accumulator	1
ANL	A, direct	AND direct Byte to Accumulator	2
ANL	A, @Ri	AND indirect RAM to Accumulator	1
ANL	A, #data	AND immediate data to Accumulator	2
ANL	direct, A	AND Accumulator to direct Byte	2
ANL	direct, #data	AND immediate data to direct Byte	3
ORL	A, Rn	OR register to Accumulator	1
ORL	A, direct	OR direct Byte to Accumulator	2
ORL	A, @Ri	OR indirect RAM to Accumulator	1
ORL	A, #data	OR immediate data to Accumulator	2
ORL	direct, A	OR Accumulator to direct Byte	2
ORL	direct, #data	OR immediate data to direct Byte	3
XRL	A, Rn	Exclusive-OR register to Accumulator	1
XRL	A, direct	Exclusive-OR direct Byte to Accumulator	2
XRL	A, @Ri	Exclusive-OR indirect RAM to Accumulator	1
XRL	A, #data	Exclusive-OR immediate data to Accumulator	2
XRL	direct, A	Exclusive-OR Accumulator to direct Byte	2
XRL	direct, #data	Exclusive-OR immediate data to direct	3
CLR	A	Clear Accumulator	1
CPL	A	Complement Accumulator	1
RL	A	Rotate Accumulator left	1
RLC	A	Rotate A left through the Carry flag	1
RR	A	Rotate Accumulator right	1
RRC	A	Rotate A right through Carry flag	1
SWAP	A	Swap nibbles within the Accumulator	1

Table 2–13: Boolean variable manipulation

Mnemonic		Description	Byte
CLR	C	Clear Carry flag	1
CLR	bit	Clear direct bit	2
SETB	C	Set Carry flag	1
SETB	bit	Set direct bit	2
CPL	C	Complement Carry flag	1
CPL	bit	Complement direct bit	2
ANL	C, bit	AND direct bit to Carry flag	2
ANL	C, /bit	AND complement of direct bit to Carry	2
ORL	C, bit	OR direct bit to Carry flag	2
ORL	C, /bit	OR complement of direct bit to Carry	2
MOV	C, bit	Move direct bit to Carry flag	2
MOV	bit, C	Move Carry flag to direct bit	2

Table 2–14: Data transfer operations

Mnemonic		Description	Byte
MOV	A, Rn	Move register to Accumulator	1
MOV	A, direct	Move direct Byte to Accumulator	2
MOV	A, @Ri	Move indirect RAM to Accumulator	1
MOV	A, #data	Move immediate data to Accumulator	2
MOV	Rn, A	Move Accumulator to register	1
MOV	Rn, direct	Move direct Byte to register	2
MOV	Rn, #data	Move immediate data to register	2
MOV	direct, A	Move Accumulator to direct Byte	2
MOV	direct, Rn	Move register to direct Byte	2
MOV	direct, direct	Move direct Byte to direct	3
MOV	direct, @Ri	Move indirect RAM to direct Byte	2
MOV	direct, #data	Move immediate data to direct Byte	3
MOV	@Ri, A	Move Accumulator to indirect RAM	1
MOV	@Ri, direct	Move direct Byte to indirect RAM	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2
MOV	DPTR, #data 16	Load Data Pointer with a 16-bit constant	3
MOVC	A@A + DPTR	Move Code Byte relative to DPTR to Accumulator	1
MOVC	A@A + PC	Move Code Byte relative to PC to Accumulator	1
MOVX	A, @Ri	Move External RAM (8-bit addr) to Accumulator[1)]	1
MOVX	A, @DPTR	Move External RAM (16-bit addr) to Accumulator	1
MOVX	@Ri, A	Move A to External RAM (8-bit addr)[1)]	1
MOVX	@DPTR, A	Move A to External RAM (16-bit addr)	1
PUSH	direct	Push direct Byte onto stack	2
POP	direct	Pop direct Byte from stack	2
XCH	A, Rn	Exchange register with Accumulator	1
XCH	A, direct	Exchange direct Byte with Accumulator	2
XCH	A, @Ri	Exchange indirect RAM with Accumulator	1
XCHD	A, @Ri	Exchange low-order digital indirect RAM with A[1)]	1
1) not applicable			

Table 2–15: Program and machine control operations

Mnemonic		Description	Byte
ACALL	addr 11	Absolute subroutine call	2
LCALL	addr 16	Long subroutine call	3
RET		Return from subroutine	1
RETI		Return from interrupt	1
AJMP	addr 11	Absolute jump	2
LJMP	addr 16	Long jump	3
SJMP	rel	Short jump (relative addr)	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1
JZ	rel	Jump if Accumulator is zero	2
JNZ	rel	Jump if Accumulator is not zero	2
JC	rel	Jump if Carry flag is set	2
JNC	rel	Jump if Carry flag is not set	2
JB	bit, rel	Jump if direct bit set	3
JNB	bit, rel	Jump if direct bit not set	3
JBC	bit, rel	Jump if direct bit is set and clear bit	3
CJNE	A, direct rel	Compare direct to A and jump if not equal	3
CJNE	A, #data, rel	Compare immediate to A and jump if not equal	3
CJNE	Rn, #data, rel	Compare immediate to register and jump if not equal	3
CJNE	@Ri, #data, rel	Compare immediate to indirect and jump if not equal	3
DJNZ	Rn, rel	Decrement register and jump if not zero	2
DJNZ	direct, rel	Decrement direct and jump if not zero	3
NOP		No operation	1

2.2.10.4. Instruction Opcodes in Hexadecimal Order

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	–
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
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	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
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
1F	1	DEC	R7
20	3	JB	bit addr, code addr
21	2	AJMP	code addr
22	1	RET	–
23	1	RL	A
24	2	ADD	A, #data
25	2	ADD	A, data addr
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 addr, code addr
31	2	ACALL	code addr
32	1	RETI	–
33	1	RLC	A
34	2	ADDC	A, #data
35	2	ADDC	A, data addr
36	1	ADDC	A, @R0
37	1	ADDC	A, @R1
38	1	ADDC	A, R0
39	1	ADDC	A, R1

Table 2–16: Instruction opcodes in hexadecimal order

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
3A	1	ADDC	A, R2
3B	1	ADDC	A, R3
3C	1	ADDC	A, R4
3D	1	ADDC	A, R5
3E	1	ADDC	A, R6
3F	1	ADDC	A, R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr., A
43	3	ORL	data addr, #data
44	2	ORL	A, #data
45	2	ORL	A, data addr
46	1	ORL	A, @R0
47	1	ORL	A, @R1
48	1	ORL	A, R0
49	1	ORL	A, R1
4A	1	ORL	A, R2
4B	1	ORL	A, R3
4C	1	ORL	A, R4
4D	1	ORL	A, R5
4E	1	ORL	A, R6
4F	1	ORL	A, R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr, A
53	3	ANL	data addr, #data
54	2	ANL	A, #data
55	2	ANL	A, data addr
56	1	ANL	A, @R0
57	1	ANL	A, @R1

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
58	1	ANL	A, R0
59	1	ANL	A, R1
5A	1	ANL	A, R2
5B	1	ANL	A, R3
5C	1	ANL	A, R4
5D	1	ANL	A, R5
5E	1	ANL	A, R6
5F	1	ANL	A, R7
60	2	JZ	code addr
61	2	AJMP	code addr.
62	2	XRL	data addr, A
63	3	XRL	data addr, #data
64	2	XRL	A, #data
65	2	XRL	A, data addr
66	1	XRL	A, @R0
67	1	XRL	A, @R1
68	1	XRL	A, R0
69	1	XRL	A, R1
6A	1	XRL	A, R2
6B	1	XRL	A, R3
6C	1	XRL	A, R4
6D	1	XRL	A, R5
6E	1	XRL	A, R6
6F	1	XRL	A, R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C, bit addr
73	1	JMP	@A + DPTR
74	2	MOV	A, #data
75	3	MOV	data addr, #data

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
76	2	MOV	@R0, #data
77	2	MOV	@R1, #data
78	2	MOV	R0, #data
79	2	MOV	R1, #data
7A	2	MOV	R2, #data
7B	2	MOV	R3, #data
7C	2	MOV	R4, #data
7D	2	MOV	R5, #data
7E	2	MOV	R6, #data
7F	2	MOV	R7, #data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C, bit addr
83	1	MOVC	A, @A + PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr, @R0
87	2	MOV	data addr, @R1
88	2	MOV	data addr, R0
89	2	MOV	data addr, R1
8A	2	MOV	data addr, R2
8B	2	MOV	data addr, R3
8C	2	MOV	data addr, R4
8D	2	MOV	data addr, R5
8E	2	MOV	data addr, R6
8F	2	MOV	data addr, R7
90	3	MOV	DPTR, #data 16
91	2	ACALL	code addr

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
92	2	MOV	bit addr, C
93	1	MOVC	A, @A + DPTR
94	2	SUBB	A, #data
95	2	SUBB	A, data addr
96	1	SUBB	A, @R0
97	1	SUBB	A, @R1
98	1	SUBB	A, R0
99	1	SUBB	A, R1
9A	1	SUBB	A, R2
9B	1	SUBB	A, R3
9C	1	SUBB	A, R4
9D	1	SUBB	A, R5
9E	1	SUBB	A, R6
9F	1	SUBB	A, R7
A0	2	ORL	C, /bit addr
A1	2	AJMP	code addr
A2	2	MOV	C, bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5	–	reserved	–
A6	2	MOV	@R0, data addr
A7	2	MOV	@R1, data addr
A8	2	MOV	R0, data addr
A9	2	MOV	R1, data addr
AA	2	MOV	R2, data addr
AB	2	MOV	R3, data addr
AC	2	MOV	R4, data addr
AD	2	MOV	R5, data addr
AE	2	MOV	R6, data addr
AF	2	MOV	R7, data addr

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
B0	2	ANL	C, /bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A, #data, code addr
B5	3	CJNE	A, data addr, code addr
B6	3	CJNE	@R0, #data, code addr
B7	3	CJNE	@R1, #data, code addr
B8	3	CJNE	R0, #data, code addr
B9	3	CJNE	R1, #data, code addr
BA	3	CJNE	R2, #data, code addr
BB	3	CJNE	R3, #data, code addr
BC	3	CJNE	R4, #data, code addr
BD	3	CJNE	R5, #data, code addr
BE	3	CJNE	R6, #data, code addr
BF	3	CJNE	R7, #data, code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A, data addr
C6	1	XCH	A, @R0
C7	1	XCH	A, @R1

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
C8	1	XCH	A, R0
C9	1	XCH	A, R1
CA	1	XCH	A, R2
CB	1	XCH	A, R3
CC	1	XCH	A, R4
CD	1	XCH	A, R5
CE	1	XCH	A, R6
CF	1	XCH	A, R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr, code addr
D6	–	not applicable	–
D7	–	not applicable	–
D8	2	DJNZ	R0, code addr
D9	2	DJNZ	R1, code addr
DA	2	DJNZ	R2, code addr
DB	2	DJNZ	R3, code addr
DC	2	DJNZ	R4, code addr
DD	2	DJNZ	R5, code addr
DE	2	DJNZ	R6, code addr
DF	2	DJNZ	R7, code addr
E0	1	MOVX	A, @DPTR
E1	2	AJMP	code addr
E2	–	not applicable	–
E3	–	not applicable	–

Table 2–16: Instruction opcodes in hexadecimal order

Hex Code	Number of Bytes	Mnemonic	Operands
E4	1	CLR	A
E5	2	MOV	A, data addr
E6	1	MOV	A, @R0
E7	1	MOV	A, @R1
E8	1	MOV	A, R0
E9	1	MOV	A, R1
EA	1	MOV	A, R2
EB	1	MOV	A, R3
EC	1	MOV	A, R4
ED	1	MOV	A, R5
EE	1	MOV	A, R6
EF	1	MOV	A, R7
F0	1	MOVX	@DPTR, A
F1	2	ACALL	code addr
F2	–	not applicable	–
F3	–	not applicable	–
F4	1	CPL	A
F5	2	MOV	data addr, A
F6	1	MOV	@R0, A
F7	1	MOV	@R1, A
F8	1	MOV	R0, A
F9	1	MOV	R1, A
FA	1	MOV	R2, A
FB	1	MOV	R3, A
FC	1	MOV	R4, A
FD	1	MOV	R5, A
FE	1	MOV	R6, A
FF	1	MOV	R7, A

2.3. Interrupt

2.3.1. Interrupt System

External events and the real-time operation of on-chip peripherals require CPU service asynchronous to the execution of any particular section of code. To couple the asynchronous activities of these functions to normal program execution, a sophisticated multiple-source, four-priority-level, nested interrupt system is provided.

2.3.2. Interrupt Sources

The TVT microcontroller core is capable of handling up to 24 interrupt sources. In Type<Default ¶ Font> 17 interrupts are implemented. The rest are reserved for future use. The microcontroller acknowledges interrupt requests from these 17 sources. Two external sources via the $\overline{INT0}$ and INT1 pins and two additional external interrupts INTX0 (P3.1) and INTX1 (P3.7) are provided. On-chip peripherals also use interrupts: one from each of the two internal counters, one from the analog digital converter and one from UART. In addition there are four Data Acquisition related interrupts, two display related interrupts and one interrupt indicating change of channel, two interrupts are generated by the WDT and PWM overflow in timer mode.

Timer 0 and Timer 1 overflows are indicated by TCON(TF0) and TCON.(TF1). Interrupts are generated following a rollover in their respective registers (except in Mode 3 when TCON(TH0) controls the Timer 1 interrupt).

The external interrupts $\overline{INT0}$ and $\overline{INT1}$ are either level or edge triggered depending on bits in **TCON** and **IRCON**. Other external interrupts are level sensitive and active high. Any edge triggering will need to be taken care of by individual peripherals.

INTX0 and INTX1 can be programed to be either negative or positive edge triggered.

The analog digital converter interrupt is generated on completion of the analog digital conversion.

2.3.3. Overview

A simple overview of the interrupt handling is shown in Fig. 2–4.

.

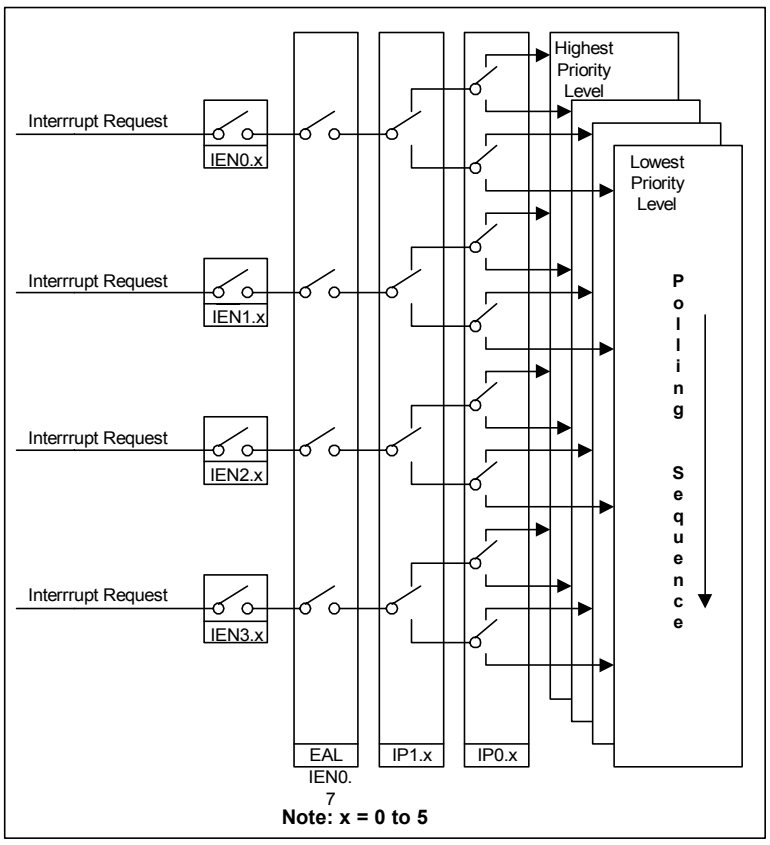


Fig. 2–4: Interrupt Handling Overview

2.3.4. Enabling Interrupts

Interrupts are enabled through a set of Interrupt Enable registers (**IEN0**, **IEN1**, **IEN2**, **IEN3**).

Bits 0 to 5 of the Interrupt Enable registers each individually enable/disable a particular interrupt source. Overall control is provided by bit 7 of **IEN0** (EAL). When EAL is set to '0', all interrupts are disabled: when EAL is set to '1', interrupts are individually enabled or disabled through the other bits of the Interrupt Enable Registers. EAL may however be overridden by the DISINT signal which provides a global disable signal for the interrupt controller.

2.3.4.1. Interrupt Enable Registers (IEN0, IEN1, IEN2, IEN3)

The microcontroller has 4 Interrupt Enable registers. For each bit in these registers, a '1' enables the corresponding interrupt and a '0' disables it.See Table 2–17.

2.3.5. Interrupt Source Registers

All the interrupts except for timer0, timer1, external interrupt0, external interrupt1, external extra interrupt0 and external extra interrupt1 are generated by the respective blocks and are positive edge triggered. They are sampled in a central interrupt source register, the corresponding bit must be cleared by the software after entering the interrupt service routine.

Table 2–17: Interrupt enable registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IEN0	EAL	Reserved	EAD	EU	ET1	EX1	ET0	EX0
IEN1			EDV	EAV	EXX1	EWT	EXX0	EX6
IEN2			EDH	EAH	ECC	EPW	EX13	EX12
IEN3			EADW	E24	EX21	EX20	EX19	EX18
See Section 3. on page 110 for detailed register description.								

Table 2–18: Interrupt source registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CISR0 bit addressable	L24	ADC	WTmr	AVS	DVS	PWtmr	AHS	DHS
CISR1 it addressable	CC	ADW					IEX1	IEX0
See Section 3. on page 110 for detailed register description.								

2.3.6. Interrupt Priority

For the purposes of assigning priority, the 24 possible interrupt sources are divided into groups determined by their bit position in the Interrupt Enable Registers. Their respective requests are scanned in the order as shown in Table 2–19.

Each interrupt group may individually be assigned to one of four priority levels by writing to the **IP0** and **IP1** Interrupt Priority registers at the corresponding bit position.

An interrupt service routine may only be interrupted by an interrupt of higher priority level. If two interrupts of different priority occur at the same time, the higher level interrupt will be serviced first. An interrupt cannot be interrupted by another interrupt of the same or a lower priority level.

If two interrupts of the same priority level occur simultaneously, the order in which the interrupts are serviced is determined by the scan order shown below.

Table 2–19: Interrupt priority

Interrupt Group	Interrupts in Group High Priority				Group Priority
0	External Interrupt 0	External Interrupt 6[1)]	External Interrupt 12[1)]	External Interrupt 18[1)]	High Priority
1	Timer 0	ExternalX Interrupt 0	External Interrupt 13[1)]	External Interrupt 19[1)]	
2	External Interrupt 1	WT Timer	PW Timer	External Interrupt 20[1)]	
3	Timer 1	ExternalX Interrupt 1	Channel Change	External Interrupt 21[1)]	
4	UART	Acquisition V-Sync	Acquisition H-Sync	Line 24 Start	
5	A to D	Display V-Sync	Display H-Sync	A to D Wake up	

[1)] Not implemented

2.3.6.1. Interrupt Priority Registers (IP0 IP1)

Table 2–20: Related registers

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
IP0 bit addressable			G5P0	G4P0	G3P0	G2P0	G1P0	G0P0
IP1			G5P1	G4P1	G3P1	G2P1	G1P1	G0P1

See Section 3. on page 110 for detailed register description.

2.3.7. Interrupt Vectors

When an interrupt is served, a long call instruction is executed to one of the locations listed in Table 2–21.

Table 2–21: Interrupt vectors

Interrupt Sources	Interrupt Enable		Vector Address (hex)	Interrupt Request Flag
	Register	Bit		
External Interrupt 0	IEN0	EX0	0003	IE0 (TCON.1)
Timer 0 Overflow	IEN0	ET0	000B	TF0 (TCON.5)
External Interrupt 1	IEN0	EX1	0013	IE1 (TCON.3)
Timer 1 Overflow	IEN0	ET1	001B	TF1 (TCON.7)
UART	IEN0	EU	0023	R1(SCON.0) and T1(SCON.1)
A to D	IEN0	EAD	002B	ADC(CISR0.6)
External Interrupt 6	IEN1	EX6	0033	Reserved
ExternalX Interrupt 0	IEN1	EXX0	003B	CISR1(IEX0)
Watchdog in timer	IEN1	EWT	0043	WTmr(CISR0.5)
ExternalX Interrupt 1	IEN1	EXX1	004B	CISR1(IEX1)
Acquisition V-Sync	IEN1	EAV	0053	AVS(CISR0.4)
Display V-Sync	IEN1	EDV	005B	DVS(CISR0.3)
External Interrupt 12	IEN2	EX12	0063	Reserved
External Interrupt 13	IEN2	EX13	006B	Reserved
PWM in timer mode	IEN2	EPW	0083	PWtmr(CISR0.2)
Channel Change	IEN2	ECC	008B	CC(CISR1.7)
Acquisition H-Sync	IEN2	EAH	0093	AHS(CISR0.1)
Display H-Sync	IEN2	EDH	009B	DHS(CISR0.0)
External Interrupt 18	IEN3	EX18	00A3	Reserved
External Interrupt 19	IEN3	EX19	00AB	Reserved
External Interrupt 20	IEN3	EX20	00B3	Reserved
External Interrupt 21	IEN3	EX21	00BB	Reserved
Line 24 Start	IEN3	E24	00C3	L24(CISR0.7)
A to D Wake up	IEN3	EADW	00CB	ADW(CISR1.6)

2.3.8. Interrupt and Memory Extension

When an interrupt occurs, the Memory Management Unit (MMU) carries out the following sequence of actions:

1. The **MEX1** register bits are made available on SDATAO[7:0].

2. The **MEXSP** register bits are made available on SADD[7:0].

3. The Stack read and write signals are set for a write operation.

4. A write is performed to External memory.

5. The **MEXSP** Stack Pointer is incremented.

6. The Interrupt Bank bits IB19 - IB16 (MEX2.3 - MEX2.0) are copied to both the NB19 - NB16 and the CB19 - CB16 bits in the **MEX1**.

Then on return from the interrupt service routine:

1. The **MEXSP** Stack Pointer is decremented.

2. The **MEXSP** register bits are made available on SADD [7:0].

3. The Stack read and write signals are set for a read operation.

4. A read is performed on External memory.

5. SDATAI [7:0] is copied to the **MEX1** register.

This action allows the user to place interrupt service routines on specific banks.

2.3.9. Interrupt Handling

External interrupt0, external interrupt1, timer0, timer1 and UART interrupt are handled as follows:

– Interrupts are sampled at Step5 Phase2 in each machine cycle and the sampled interrupt information is polled during the following machine cycle. If an interrupt is active when it is sampled, it will be serviced provided:

– An interrupt of an equal or higher priority is not currently being serviced.

– The polling cycle is not the final cycle of a multi-cycle instruction, and

– The current instruction is neither a RETI nor a write either to one of Interrupt Enable registers or to one of the Interrupt Priority registers.

Note: Active interrupts are only stored for one machine cycle. As a result, if an interrupt was active for one or more polling cycles but not serviced for one of the reasons given above, the interrupt will not be processed.

For all other interrupts the interrupt request is stored as an interrupt flag in registers **CISR0** and **CISR1**. These request bits must be cleared by user software while servicing the interrupt. The interrupts always get serviced once raised regardless of the number of polling cycles required to service them.

2.3.10. Interrupt Latency

The response time in a single interrupt system is between 3 and 9 machine cycles.

2.3.11. Interrupt Flag Clear

In case of external interrupt0 and external interrupt1, if the external interrupts are edge triggered, the interrupt flag is cleared when entering into the interrupt service routine but if they are level triggered, the flag follows the signal applied to the port pin. Timer/counter flags are cleared when entering into the interrupt service routine. All other interrupt flags, including IEX0 and IEX1 are not cleared by hardware. They must be cleared by software.

2.3.12. Interrupt Return

For the proper operation of the interrupt controller it is necessary that all interrupt routines end with a RETI instruction.

2.3.13. Interrupt Nesting

The process whereby a higher-level interrupt request interrupts a lower-level interrupt service routine is called “nesting”.

In this case the address of the next instruction in the lower-priority service routine is pushed onto the stack, the stack pointer is incremented by two and the microcontroller will continue the SW program execution from the memory location of the first instruction of the higher-level service routine. The last instruction of the higher-priority interrupt service program must be a RETI-instruction. This instruction clears the higher ‘priority-level-active’ flip-flop. The RETI command also makes that the microcontroller executes the next instruction of the lower-level interrupt service routine. Since the lower ‘priority-level-active’ flip-flop has remained set, higher priority interrupts are re-enabled while further lower-priority interrupts remain disabled.

2.3.14. External Interrupts

The external interrupt request inputs ($\overline{INT0}$ and $\overline{INT1}$) can be programmed for either transition- activated or level-activated operation. Control of the external interrupts is provided in the **TCON** register.

Table 2–22: Related register

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
See Section 3. on page 110 for detailed register description.								

2.3.15. Extension of Standard 8051 Interrupt Logic

For more flexibility, SDA 55xx family provides a new feature for the status detection of external extra interrupts EX0 and EX1 in an edge-triggered mode. Now there is the possibility to trigger an interrupt on the falling and/or rising edge at the dedicated Port 3 Pin. In order to use this feature respective IT0 and IT1 bits in the **TCON** register must be set to activate edge triggering mode. Table 2–23 shows combination for Interrupt0, however description is also true for interrupt1.

Table 2–23: Interrupt combinations

IT0	EX0R	EX0F	Interrupt
0	0	0	Disabled
0	0	1	Low level
0	1	0	High level
0	1	1	Disabled
1	0	0	Disabled
1	0	1	Negative edge triggered
1	1	0	Positive edge triggered
1	1	1	Positive and negative edge triggered

Table 2–24: Related register

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
IRCON	EXX1R	EXX1F	EXX0R	EXX0F	EX1R	EX1F	EX0R	EX0F
See Section 3. on page 110 for detailed register description.								

Note: If both EXXxR and EXXxF are set then both rising and falling edges would generate an interrupt. Minimum delay between the interrupts should be ensured by the software.
If both the EXXxR and EXXxF are reset to 0, interrupt is disabled.

External extra interrupts EX1 and EX2 are edge triggered interrupts only.

2.3.16. Interrupt Task Function

The microcontroller records the active priority level(s) by setting internal flip-flop(s). Each interrupt level has its own flip-flop. The flip-flop corresponding to the interrupt level being serviced is reset when the microcontroller executes a RETI-instruction.

The sequence of events for an interrupt is:

– A source provokes an interrupt by setting its associated interrupt request bit to let the microcontroller know an interrupt condition has occurred.

– The interrupt request is conditioned by bits in the interrupt enable and interrupt priority registers.

– The microcontroller acknowledges the interrupt by setting one of the four internal 'priority-level active' flip-flops and performing a hardware subroutine call. This call pushes the PC (but not the **PSW**) onto the stack and, for some sources, clears the interrupt request flag.

– The service program is executed.

– Control is returned to the main program when the RETI-instruction is executed. The RETI- instruction also clears one of the internal 'priority-level active' flip-flops.

The interrupt request flags IE0, IE1, TF0 and TF1 are cleared when the microcontroller transfers control to the first instruction of the interrupt service program.

2.3.17. Power Saving Modes

The controller provides four modes in which power consumption can be significantly reduced.

– Idle mode: The CPU is gated off from the oscillator. All peripherals except WDT (in watch dog mode) are still provided with the clock and are able to work.

– Power-down mode: Operation of the controller is turned off. This mode is used to save the contents of internal RAM with a very low standby current.

– Power-save mode: In this mode display generator, Slicer_acq_sync, VADC, CADC, ADC_wakeup, PWM, CRT, WDT, DAC, PLL, and Display (display, pixel clock and D sync) can be turned off.

– Slow-down mode: In this mode the system frequency is reduced by one fourth.

All modes are entered by software. Special function register is used to enter one of these modes.

2.3.18. Power-Save Mode Registers.

The Table 3-25 lists the respective registers which control or reflect the Power-Save Modes. A description is given below.

Table 2–25: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PSAVE				CADC	WAKUP	SLI_ACQ	DISP	PERI
PSAVEX bit addressable						Clk_src	PLL_res	PLLS
PCON	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE
See Section 3. on page 110 for detailed register description.								

2.3.19. Idle Mode

Entering the idle mode is done by two consecutive instructions immediately following each other. The first instruction has to set bit IDLE (PCON.0) and must not set bit IDLS (PCON.5). The following instruction has to set bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). Bits IDLE and IDLS will automatically be cleared after having been set. This double-instruction sequence is implemented to minimize the chance of unintentionally entering the idle mode. The following instruction sequence may serve as an example:

ORL PCON,#00000001_B ;Set bit IDLE, bit IDLS must not be set.

ORL PCON,#00100000_B ;Set bit IDLS, bit IDLE must not be set.

The instruction that sets bit IDLS is the last instruction executed before going into idle mode.

Concurrent setting of the enable and the start bits does **not** set the device into the respective power saving mode.

The idle mode can be terminated by activation of any enabled interrupt (or a hardware reset). The CPU-operation is resumed, the interrupt will be serviced and the next instruction to be executed after RETI-instruction will be the one following the instruction that set the bit IDLS. The port state and the contents of SFRs are held during idle mode.

Entering Idle mode disables, VADC, Acquisition, Slicer, Display, CADC and DAC. However note that CADC Wake up unit is still operational. Leaving idle mode brings them to their original power save configuration (See Section 2.3.21.).

2.3.20. Power-down Mode

Entering the power-down mode is done by two consecutive instructions immediately following each other. The first instruction has to set bit PDE (PCON.1) and must not set bit PDS (PCON.6). The following instruction has to set bit PDS (PCON.6) and must not set bit PDE (PCON.1). Bits PDE and PDS will automatically be cleared after having been set.

This double-instruction sequence is implemented to minimize the chance of unintentionally entering the power-down mode. The following instruction sequence may serve as an example:

ORL PCON,#00000010_B ;Set bit PDE, bit PDS must not be set.

ORL PCON,#01000000_B ;Set bit PDS, bit PDE must not be set.

The instruction that sets bit PDS is the last instruction executed before going into power-down mode.

Concurrent setting of the enable and the start bits does **not** set the device into the respective power saving mode.

If idle mode and power-down mode are invoked simultaneously, the power-down mode takes precedence.

The only exit from power-down mode is a hardware reset. The reset will redefine all SFRs, but will not change the contents of internal RAM.

2.3.21. Power-save Mode

Bits in the **PSAVE** register individually enable and disable different major blocks in the IC. Note that power-save mode is independent of Idle and power-down mode. In case of idle mode, blocks which are in power save mode remain in power-save mode.

Entering the power down mode with power-save mode is possible. However leaving the power down mode (reset) would initialize all the power save register bits.

Note that power-save mode has a higher priority then idle mode.

2.3.22. Slow-Down Mode

SD bit in **PCON** register when sets divides the system frequency by 4. During the normal operation TVT Pro is running with 33.33 MHz and in SD mode TVT Pro runs with 8.33 MHz. In slow-down mode the slicer, Acquisition and display are disabled regardless of power-save mode or other modes. All the pending request to the bus by these blocks are masked off. Leaving slow-down mode restores the original status of these blocks.

2.4. Reset

2.4.1. Reset Sources

TVText Pro can be reset by two sources:

1. Externally by pulling down the reset pin $\overline{RST}$.

2. Internally by Watch dog timer reset.

Please note that both reset signals use the same signal path however a Watchdog reset does not reset the clock PLL.

2.4.2. Reset Filtering

The RST pin uses a filter with a delay element, which suppresses jitter and spikes in the range of 25 ns to 75 ns.

2.4.3. Reset Duration

With the active edge of the $\overline{RST}$ an internal signal resets all the flip flops asynchronously. The internal signal is released synchronously to the internal clock when it is stable as described below.

The minimum duration of the external reset signal depends on the time required for the SDA55xx internal crystal oscillator to reach it's full amplitude swing and is dependent on the crystal used.

During the period when the RST pin is held low, the PLL is initialized and it gets locked. The high going reset pulse then initiates a sequence which requires one machine cycle (12 clock cycles) to initialize the microcontroller and all other registers and peripherals.

2.4.4. Registers

Upon reset, all the registers are initialized to the values as defined in Section 3. on page 110.

2.4.5. Functional Blocks

After reset all the functional blocks will be in a defined known state. Microcontroller, acquisition and display will not have any pending bus requests after reset.

2.4.6. RAMs

The HW reset and its related logic does not initialize any RAMs.

2.4.7. Analog Blocks

After the power up reset the DAC will output a fixed value. ADC and the ADC wake up unit do not generate any interrupts till the 12 cycle long reset sequence is completed.

2.4.8. Microcontroller

After the reset sequence the program counter initializes to 0000_H and starts execution from this location in the ROM. Location 0000H to 0002H are reserved for the a jump instruction to the initialization routine.

2.4.9. Ports

With the reset all ports are set in to the input mode. Exception are Port 4.0, 4.1 and 4.4, which by default after reset are assigned as data outputs for the address lines A17, A18, A19.

2.4.10. Initialization Phase

2.4.10.1. Acquisition

After the reset the Acquisition will not generate any memory accesses to the RAM, due to the fact that the Acq_start bit is initialized to '0'. The microcontroller should then initialize the VBI buffer and set the ACQ_start bit (by software). The Acquisition will not generate any accesses to the RAM if the H / V synchronization is not achieved.

2.4.10.2. Display

After reset the DACs will output a fix value as defined by En_DGOut, which is reset to '0'. COR_BL is reset to a level indicating COR = '0' and BLank = '1'.

The microcontroller should initialize the display memory and set the En_DGOut (OCD_Ctrl) bit.

2.5. Memory Organization

The microcontroller has separate Program and Data memory spaces. Memory spaces can be further classified as:

– Program Memory

– Internal Data Memory of 256 Bytes (CPU RAM)

– Internal Extended Data Memory (XRAM)

A 16-bit program counter and a dedicated banking logic provide the microcontroller with 1 MByte addressing capability (with the ROM-less versions, up to 20 address lines are available).

The program counter allows the user to execute calls and branches to any location within the program memory space.

Data pointers allow to move data to and from Extended Data RAM.

There are no instructions that permit program execution to move from the program memory space to any data memory space.

2.5.1. Program Memory

Program ROM consists of 128 KByte on chip ROM for mask programmed versions.

Locations ‘0000’ through ‘0002’ are reserved for the Long Jump instruction to the initialization routine. Following reset, the CPU always begins execution at location ‘0000’. Locations ‘0003’ through ‘00CB’ are can be reserved for the interrupt-request service routines if required.

Table 2–26: Program memory

Interrupt Source	Vector Address
External Interrupt 0	0003
Timer 0 Overflow	000B
External Interrupt 1	0013
Timer 1 Overflow	001B
UART	0023
ADC	002B
Reserved	0033
ExternalX Interrupt 0	003B
Watchdog timer	0043
External X Interrupt 1	004B
Acquisition V Sync	0053
Display V sync	005B
Reserved	0063
Reserved	006B
Reserved	0073
Reserved	007B
PWM in timer mode	0083
Channel Change	008B
Acq H Sync	0093
Display H Sync	009B
Reserved	00A3
Reserved	00AB
Reserved	00B3
Reserved	00BB
Line 24 start	00C3
A to D wake up	00CB

2.5.2. Internal Data RAM

Internal Data RAM is split into CPU RAM and XRAM

2.5.2.1. CPU RAM

2.5.2.1.1. Address Space

The internal CPU RAM (IRAM) occupies address space 00_H to FF_H. This space is further split into two regions.

The lower 128 Bytes (00_H-$7F_H$) can be accessed using both direct and indirect register addressing method.

The upper half of 128 Bytes (80_H-FF_H) can be accessed using the “register indirect method” only. Register direct method for this address space (80_H-FF_H) is reserved for Special function register access.

2.5.2.1.2. Registers

Controller registers are also located in IRAM. Four banks of eight registers each occupy locations 0 through 31. Only one of these banks may be enabled at a time through a two-bit field in the **PSW**.

2.5.2.1.3. Bit Addressable RAM Area

128-bit locations of the on-chip RAM are accessible through direct addressing.These bits reside in internal data RAM at Byte locations 32 through 47.

2.5.2.1.4. Stack

The stack can be located anywhere in the internal data RAM address space. The stack depth is limited only by the available internal data RAM, thanks to an 8-bit relocatable stack pointer. The stack is used for storing the program counter during subroutine calls and may also be used for passing parameters. Any Byte of internal data RAM or special function registers accessible through direct addressing can be pushed/popped. By default Stack Pointer always has a reset value of 07_H.

2.5.2.2. Extended Data RAM (XRAM)

An additional on-chip RAM space called ‘XRAM’ extends the capacity of the internal RAM. Up to 16 KiloBytes of XRAM are accessed by MOVX @DPTR. The XRAM is located in the upper area of the 64K address space.

1 KByte of the XRAM, called VBI Buffer, is reserved for storing teletext data. 1 KByte of address space can be allocated as CPU work space. Three KiloByte of RAM are reserved as Display RAM. The rest of the RAM can be configured either as Teletext page memory or DRCS (Dynamically Redefinable Character Set) memory.

2.5.2.2.1. Extended Data Memory Address Mapping

The XRAM is mapped in the address space from $C000_H$ to $FFFF_H$. 16 KBytes are implemented as on-chip memory. The address space of the 16K block is decoded starting from $C000_H$. Note that this decoding is done independent of the memory banking. That means that in all 16 available banks of 64K, the upper 16KByte long address space is reserved for internal Extended data memory. This decoding method has the advantage, that when copying data back and forth between on-chip RAM and off-chip RAM, there is no need to switch the memory banks.

2.5.3. Memory Extension

The controller provides four additional address lines A16, A17, A18 and A19. These additional address lines are used to access program and data memory space of up to 1MByte. The extended memory space is split into 16 banks of 64 KByte each.

A16 is available as a dedicated pin, A17, A18 and A19 however work as alternate function to port pins P4.0, P4.1 and P4.4 respectively. Refer to register **CSCR1** ($\overline{A19}$_P4_4, $\overline{A18}$_P4_1,$\overline{A17}$_P4_0).

The functionality for memory extension is provided by a Memory management Unit (MMU) which includes the four SFR registers **MEX1**, **MEX2**, **MEX3** and **MEXSP**.

2.5.3.1. Memory Extension Registers

The following registers are present in the Memory management unit.

Table 2–27: Related registers

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
MEX1	CB[19:16]				NB[19:16]			
MEX2	MM	MB[18:16]			IB[19:16)			
MEX3	MB19	UB3	UB4	MX19	MXM	MX[18:16]		
MEXSP	Reserved	SP[6:0]						
See Section 3. on page 110 for detailed register description.								

.

These registers can be read and written through MOV instructions like any other SFR register. Except for the CB bits in **MEX1** - which are read only - and can be written only by the MMU. During normal operation user must not write to the **MEXSP register**.

2.5.3.2. Reset Value

In order to insure proper 8051 functionality all the bits in SFR **MEX1**, **MEX2**, **MEX3** and **MEXSP** are initialized to ‘0’

2.5.4. Instructions on which Memory Extension would act

The following instruction are used to access the extended memory space:

– LJMP

– MOVC

– MOVX

– LCALL

– ACALL

– RET

– RETI

2.5.4.1. Program Memory Banking (LJMP)

After reset the bits for current bank (CB) and next bank (NB) are set to zero. This insures that the microcontroller starts like a standard 8051 microcontroller at address 00000_H.

When a jump to another bank is required, software changes the bits NB16 … 19 to the appropriate bank address (before LJMP instruction).

When a LJMP is encountered in the code, the MMU copies the NB16 … 19 (next bank) bits to CB16 … 19 (current bank). Note that the NB bits are not destroyed.

Bits related the Extended Memory address would appear at the pins A16 … A19. These address line have the same timing requirements compared to normal address lines A0…A15 and must be stable at the same time.

Only with LJMP above mentioned action is performed, other jmp instructions have no effect.

CB bits are read only.

2.5.4.2. MOVC Handling

There are two modes for MOVC instructions. The mode is selected by MM bit in **MEX2**.

2.5.4.2.1. MOVC with Current Bank

When MM bit = '0', MOVC will access the current bank. The CB16 … CB19 bits would appear as addresses A16 … A19 during MOVC instructions.

2.5.4.2.2. MOVC with Memory Bank

When MM bit = '1', MOVC will access the Memory bank. The MB16 … MB19 bits would appear as addresses A16 … A19 during MOVC instructions.
Note: **MEX1** is not destroyed.

2.5.4.2.3. MOVX Handling

There are two modes for MOVX instructions. The mode is selected by MXM bit in **MEX3**.

2.5.4.2.4. MOVX with Current Bank

When MXM bit = '0', MOVX will access the current bank. The CB16 … CB19 bits would appear as addresses A16 … A19 during MOVX instructions.

2.5.4.2.5. MOVX with Data Memory Bank

When MXM bit = '1', MOVX will access the Data memory bank. The MX16 … MX19 bits would appear as address A16 … A19 during MOVX instructions.
Note: **MEX1** is not destroyed.

2.5.4.3. CALLs and Interrupts

2.5.4.3.1. Memory Extension Stack

For Interrupts and Calls the Memory extension Stack is required. Stack pointer **MEXSP** provides the stack depth of up to 128 Bytes. Stack width is 1 Byte. In TVTPro 128 Bytes stack is implemented.

2.5.4.4. Stack Full

No indication for stack full is provided. The user is responsible to read **MEXSP** SFR to determine the status of the **MEXSP** stack.

2.5.4.5. Timing

The MMU outputs the address bits A19 … A16 at the same time as the normal addresses A15 … A0.

Stack operation signals, SAdd[6:0], SDataI[7:0], SDataO[7:0], SRd and SWr have the same timing as internal RAM signals.

2.5.4.6. Interfacing Extended Memory

The address bits A19, A18, A17, A16 are used to decode extended memory.

2.5.4.7. Application Examples

MOVC

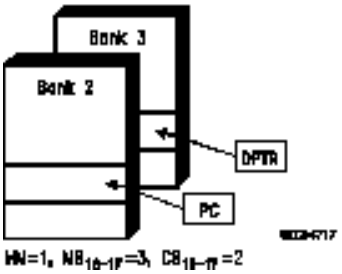


Fig. 2–5: PC and DPTR on different banks

2.5.4.7.1. Sample Code

Fig. 2–6 shows an assembler program run, performing the following actions:

1. Start at bank 0 at 00000.
2. Set ISR-page to bank 2.
3. Jump to bank 1 at address 25.
4. Being interrupted to bank 2 ISR.
5. Call a subprogram at bank 2 address 43.
6. After return read data from bank 2.

2.5.4.8. ROM and ROMless Version

The XROM pin determines whether the on-chip or the off-chip ROM is accessed.

If no internal ROM is to be used, then the XROM pin (in ROMless version) should be driven ‘low’. The controller then accesses the External ROM only. In the ROM version this pin is internally pulled high, indicating that no external ROM is available.

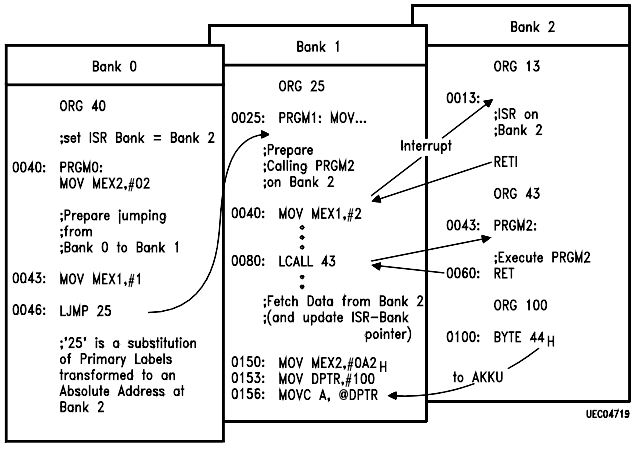


Fig. 2–6: Program Code

2.6. UART

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second Byte before a previously received Byte has been read from the receive register (however, if the first Byte still hasn’t been read by the time the reception of the second Byte is complete, one of the Bytes will be lost). The serial port receive and transmit registers are both accessed at special function register **SBUF**. Writing to **SBUF** loads the transmit register, and reading **SBUF** accesses a physically separate receive register.

The frequencies and baud rates depend on the internal system clock used by the serial interface.

2.6.1. Operation Modes of the UART

The serial port can operate in 4 different modes. In all four modes, transmission is initiated by any instruction that uses **SBUF** as a destination register. Reception is initiated in mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

2.6.1.1. Mode 0

Serial data enter and exit through pin RxD (P3.7). TxD (P3.1) outputs the shift clock.

2.6.1.2. Mode 1

10 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On reception, the stop bit goes into bit RB8 in special function register **SCON**. The baud rate is variable.

2.6.1.3. Mode 2

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9^{th} data bit, and a stop bit (1). On transmission, the 9^{th} data bit (TB8 in **SCON**) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the **PSW**) could be moved into TB8. On reception, the 9^{th} data bit goes into RB8 in the special function register **SCON**, while the stop bit is ignored. The baud rate is programmable via SFR-Bit SMOD.

2.6.1.4. Mode 3

11 bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9^{th} data bit and a stop bit (1). In fact, mode 3 is the same as mode 2 in all respects except the baud rate. The baud rate in mode 3 is variable.

Table 2–28: Select Mode 0-3 for UART

SM0	SM1	Mode	Description	Baud Rate (CDC = 0)
0	0	0	Shift Reg.	$f_{system/12}$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$f_{system/64}$, $f_{system/32}$
1	1	3	9-bit UART	Variable

2.6.2. Multiprocessor Communication

Modes 2 and 3 of the serial interface of the controller have a special provision for multiprocessor communication. In these two modes, 9 data bits are received. The 9th bit goes into register SCON bit RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in **SCON**. A way to use this feature in multiprocessor communications is as follows:

When the master microcontroller wants to transmit a block of data to one of the several slaves, it first sends out an address Byte which identifies the target slave. In an address Byte the 9th bit is a ‘1’, a data Byte is identified with a ‘0’ as 9th. bit.

If the SCON register bit SM2 is set to ‘1’, no slave will be interrupted by a data Byte. An address Byte however, will interrupt all slaves, so that each slave can examine the received Byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data Bytes that will be transmitted by the master. The slaves that were not addressed leave their SM2 bits set to 1 and go on with the execution of the currently running program and ignore the data Byte transmission on the bus.

the bit SM2 has no effect in mode 0, and in mode 1 the SM2 bit can be used to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Table 2–29: Related register

Register Name	Bit Name							
	7	6	5	4	3	2	1	0
SCON	SM0	SM1	SM2	REN	TB8	RB8	Ti	RI
See Section 3. on page 110 for detailed register description.								

2.7. General Purpose Timers/Counters

Two independent general purpose 16-bit timer/ counters are integrated for use to measure time intervals, pulse widths, counting events, and causing periodic (repetitive) interrupts. Both can be configured to operate as timer or event counter.

In the 'timer' function, the registers TLx and/or THx (x = 0, 1) are incremented once every machine cycle. As one machine cycle has a length of 12 cycles of the oscillator the counting frequency is fcrystal /12.

In the 'counter' function, the registers TLx and/or THx (x = 0, 1) are incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled during every machine cycle. If the samples taken show a '1' in one cycle and a '0' in the next cycle, the count is incremented. The new count value appears in the register during the cycle following the one in which the transition was detected. Since it takes 2 machine cycles (24 oscillator periods) to recognize a '1'-to-'0' transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle.

2.7.1. Timer/Counter 0: Mode Selection

Timer/counter 0 can be configured in one of four operating modes, which are selected by bit-pairs (M1, M0) in **TMOD** register. See Section 2.7.1. on page 53.

2.7.1.1. Mode 0

Putting timer/counter 0 into mode 0 makes it looks like an 8048 timer, which is an 8-bit counter with a divide-by-32 prescaler. Table 2–32 shows the mode 0 operation as it applies to timer 0.

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all '1' to all '0', it sets the timer interrupt flag TF0. The timer input is enabled if TR0 = 1 and either GATE = 0 or $\overline{INT0}$ = 1. (Setting GATE = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements.) TR0 is a control bit in the special function register **TCON** (see Section 2.8.5. on page 57). GATE is contained in register **TMOD** (see Section 2.7.4. on page 54).

The 13-bit register consists of all 8 bits of **TH0** and the lower 5 bits of **TL0**. The upper 3 bits of **TL0** are not valid and should be ignored. Setting the run flag TR0 does not clear the registers.

2.7.1.2. Mode 1

Mode 1 is the same as mode 0, except that all 16 bits of the timer/counter 0 register are being used.

2.7.1.3. Mode 2

Mode 2 configures the timer/counter 0 register as an 8-bit counter **TL0** with automatic reload. The High Byte **THo** is used as storage for the Reload value. Overflow from **TL0** not only sets TF0, but also reloads **TL0** with the contents of **TH0**, which is preset by software. The reload leaves **TH0** unchanged.

2.7.1.4. Mode 3

Timer/counter 0 in mode 3 establishes **TL0** and **TH0** as two separate counters. **TL0** uses the timer 0 control bits: C/T, GATE, TR0, $\overline{INT0}$ and TF0. **TH0** is locked into a timer function (counting machine cycles) 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 or counter. With timer 0 in mode 3, the microcontroller can operate as if it has three timers/ counters. 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 can still be used in any application not requiring an interrupt.

2.7.2. Timer/Counter 1: Mode Selection

Timer/counter 1 can also be configured in one of four modes, which are selected by its own bitpairs (M1, M0) in **TMOD** register.

The serial port receives a pulse each time that timer/ counter 1 overflows. This pulse rate is divided to generate the transmission rate of the serial port.

Modes 0 and 1 are the same as for counter 0.

2.7.2.1. Mode 2

The 'reload' mode is reserved to determine the frequency of the serial clock signal (not implemented).

2.7.2.2. Mode 3

When counter 1's mode is reprogrammed to mode 3 (from mode 0, 1 or 2), it disables the increment counter. This mode is provided as an alternative to using the TR1 bit (in **TCON** register) to start and stop timer/counter 1.

2.7.3. Configuring the Timer/Counter Input

The use of the timer/counter is determined by two 8-bit registers, **TMOD** (timer mode) and **TCON** (timer control). The input to the counter circuitry is from an external reference (for use as a counter), or from the on-chip oscillator (for use as a timer). The operation mode depends on whether TMOD's C/T-bit is set or cleared, respectively. When used as a time base, the on-chip oscillator frequency is divided by twelve or six before being used as the counter input. When TMOD's GATE bit is set (1), the external reference input (T1, T0) or the oscillator input is gated to the counter conditional upon a second external input ($\overline{INT0}$) or ($\overline{INT1}$) being high. When the GATE bit is zero (0), the external reference, or oscillator input, is unconditionally enabled. In either case, the normal interrupt function of $\overline{INT0}$ and $\overline{INT1}$ is not affected by the counter's operation. If enabled, an interrupt will occur when the input at $\overline{INT0}$ or $\overline{INT1}$ is low. The counters are enabled for incrementing when **TCON**'s TR1 and TR0 bits are set. When the counters overflow, the TF1 and TF0 bits in **TCON** get set, and interrupt requests are generated.

The counter circuitry counts up to all 1's and then overflows to either 0's or the reload value. Upon overflow, TF1 or TF0 is set. When an instruction changes the timer's mode or alters its control bits, the actual change occurs at the end of the instruction's execution.

2.7.4. Timer/Counter Mode Register

Table 2–30: TMOD

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
	Timer 1				Timer 0			
See Section 3. on page 110 for detailed register description.								

Table 2–31: TCON

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
See Section 3. on page 110 for detailed register description.								

2.8. Capture Reload Timer

The capture control timer is a 16 bit up counter, with special features suited for easier infrared decoding by measuring the time interval between two successive trigger events. Trigger events can be positive, negative or both edges of a digital input signal (Port 3.2 or 3.3). A built in Spike Suppression Unit (SSU) can be used for suppressing pulses with obviously too small or too long time duration at the beginning of an expected telegram, thereby relieving the FW of processing corrupted telegrams. This is especially useful in idle mode.

2.8.1. Input Clock

The input clock is f_{CCT} and is same as the system clock frequency divided by two. In normal mode the system frequency is 33.33 MHz (f_{CCT} = 16.66 MHz) and in slow down mode (SD mode) it is 8.33 MHz (f_{CCT} = 4.16 MHz).

PR prescaler bit: when set the input clock is further divided by 2, setting PR1 divides further by 8.

If the operation is changed to the SD mode the frequency is adjusted accordingly so that maximum time resolution of 15.73 ms or 251.66 ms is achieved depending on Prescaler PR bits.

2.8.2. Reset Values

All the eight 8 bit registers **CRT_relI**, **CRT_relh**, **CRT_capl**, **CRT_caph**, **CRT_mincapl**, **CRT_mincaph**, **CRT_con0** and **CRT_con1** are reset to 00_H.

2.8.3. Functional Description

2.8.3.1. Port Pin

Either Port P3.3 or P3.2 can be selected as capture input via SEL bit. Capture event can be programmed to occur on rising or falling edge or both using the bits RISE and FALL bits.

2.8.3.2. Slow Down Mode

SD bit when set, reduces the system frequency to 8.33 MHz. However the clk to the counter has a fix frequency (for a particular prescaler value). This is achieved by a divide by 4 chain, which divides the incoming frequency by 4 when SD = 0 and feeds the incoming signal directly to the counter when SD = 1.

2.8.3.3. Run

When the counter is started (RUN), a 16 bit reload value is automatically loaded into the 16 bit counter. (Note: REL bit is irrelevant in case of RUN function). Setting run bit resets the First and OV bit.

All the control bits PR, PLG, REL, RUN, RISE, FALL, SEL, Start, Int_Src, SD can be changed anytime during the operation. These changes take immediate effect. There is no protected mode when the counter is running.

2.8.3.4. Overflow

In case no capture event occurs, the counter keeps on counting till it overflows from $FFFF_H$ to 0000_H. At this transition the OV bit is set. After the overflow the counter keeps on counting. Overflow does not reload the reload value. Note that the OV bit is set by the counter and can be reset by software.

2.8.3.5. Modes

There are three different modes in which the counter can be used.

– Normal Capture mode

– Polling mode

– Capture mode with spike suppression at the start of an IR telegram

Table 2–32: Timer/Counter mode selection

Mode	START	PLG
Normal capture mode	0	0
Capture mode with spike suppression	1	0
Polling mode	X	1

For each change in the mode selection it is recommended to reset the RUN bit (if it is not already at 0), set the appropriate mode bit and then re-start the counter by setting the RUN bit again.

For each of the capture modes the event is captured based on the CRTCON0 (bit RISE) and CRTCON0 (bit FALL).

2.8.3.6. Normal Capture Mode

Normal capture mode is started by setting the RUN bit (0 --> 1) and PLG = 0, start = 0. Setting RUN bit will reload the counter with reload value and reset the overflow bit and counter will start to count.

Upon event on the selected port pin, contents of the counter are copied to the capture registers **CRT_caph** and **CRT_capl**.

In capture mode if REL bit is set counter is automatically reloaded upon occurrence of the event with the reload value and starts to count. If however REL bit is not set then the counter continues to count from the current value.

OV bit is not effected by the capture event.

Note: Min_cap register has no functionality in this mode.
Interrupt would be generated from CRT, however it will only be registered in the int source register if Intsrc bits in the **CSCR1** are appropriately set. It is not required to use the CRT generated interrupt in this mode. Direct pin interrupt can be used.

2.8.3.7. Polling Mode

The polling mode is started by setting the PLG bit to '1' (START bit is in don't care for this mode). Setting the RUN bit will reload the counter with the reload value and reset the overflow bit and start the counting.

In the timer polling mode, the capture register mirrors the current timer value, note that in this mode any event at the selected port pin is ignored. Upon overflow the OV bit is set.

Note: Interrupts are not generated as events are not recognized.

2.8.3.8. Capture Mode with Spike Suppression at the Start of an Infrared Telegram

This mode is specially been implemented to prevent false interrupt from being generated specially in idle mode while waiting for a new infrared telegram.

This mode is entered by setting the START bit (PLG = 0). The software sets the Start bit to indicate it is expecting a new telegram. Setting the RUN bit will reload the counter with the reload value and reset the overflow bit and start the counting.

2.8.3.9. First Event

On occurrence of the capture event, the counter value is captured and the comparator then sets the First bit. The Interrupt is suppressed. The OV bit is reset and the counter reloads the reload value (regardless of the status of REL bit) and starts counting again.

2.8.3.10. Second Event

On occurrence of second capture event, the counter value is captured and the interrupt is triggered if the capture value exceeds the value in the Min_Cap register and the OV bit is not set. First bit is reset. The counter will now continue in the normal capture mode. Software may reset the START bit if the capture value is detected as a valid pulse of an IR telegram.

If the pulse was invalid then software must stop the counter and start again (Run bit & First reset and then SET) with start bit set to wait for a new telegram.

If Capture value is less then or equal to min_cap value or OV bit has been set, that is spike has been detected and Interrupt is suppressed. OV bit would be reset counter would be reloaded with reload value (regardless of REL bit).

In this case if either RISE or FALL bit were set then counter will wait for the second event (First = 1), if RISE and FALL both were set then counter will wait for the first event (First = 0).

2.8.3.11. Capture Reload TImer CRT Interrupt

The Capture Reload Timer CRT can generate an interrupt if the Spice Suppression Unit SSU is employed.

The CRT unit uses the same interrupt line as $\overline{\text{INT1}}$ and $\overline{\text{INT0}}$. The interrupt line is selected by the SEL bit.

Note that when using CRT to generate an interrupt, the direct interrupt source from Port 3.2 or 3.3 (which ever is selected) should be switched to CRT (**CSCR1**(IntSrc0), **CSCR1**(IntSrc1)). If the application uses port pins directly to generate interrupts, then these bits should be reset. Note that by default $\overline{\text{INT1}}$ and $\overline{\text{INT0}}$ are mapped to P3.3 and P3.2.

The SSU generates an interrupt signal as a pulse, which is captured in the interrupt source register **TCON** (IE1 or IE0). While using this mode **TCON** (IT0 or IT1) must be set to 1 (edge triggered) and **IRCON** (EX1R or EX0R) must be set to 1 and **IRCON**(EX1F or EX0F) must be set to 0.

For further information on interrupts please refer to Section 2.3. on page 37.

2.8.3.12. Counter Stop

The counter can be stopped any time by resetting the RUN bit. If the counter is stopped and started again (reset and set the RUN bit), the counter reloads with the RELOAD value and reset the OV bit.

2.8.4. Idle and Power-down Mode

In idle mode the Capture Reload Timer CRT continues to function normally, unless it has been explicitly shut down via the **PSAVEX** (PERI) bit.

In power down mode the Capture Reload Timer CRT is shut off.

2.8.5. Registers

The **CRT_rell** and **CRT_relh** are the reload registers (SFR address $B7_H$ and $B9_H$), **CRT_caph** and **CRT_capl** are the corresponding capture registers

(SFR address BA_H and BB_H). **CRT_mincapl** and **CRT_mincaph** (SFR Address BC_H, BD_H) are minimum capture registers. **CRT_con0** ($E5_H$) and **CRT_con1** are the control registers.

Table 2–33: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRT_rell	RelL[7:0]							
CRT_relh	RelH[7:0]							
CRT_capl	CapL[7:0]							
CRT_caph	CapH[7:0]							
CRT_mincapl	MinL[7:0]							
CRT_mincaph	[7:0]							
CRT_con0	OV	PR	PLG	REL	RUN	RISE	FALL	SEL
CRT_con1	Reserved	Reserved	Reserved	Reserved	Reserved	PR1	First	Start
See Section 3. on page 110 for detailed register description.								

Table 2–34: Time resolution

SD	f_{sys}	PR1	PR	f_{ctr}	f_{ctr}	Time Res.	Max Pulse Width
0	33.33 MHz	0	0	f_{sys}/8 MHz	4.17 MHz	240 ns	15.73 ms
		0	1	f_{sys}/16 MHz	2.083 MHz	480 ns	31.46 ms
		1	0	f_{sys}/64 MHz	.5208 MHz	1920 ns	125.83 ms
		1	1	f_{sys}/128 MHz	.2604 MHz	3840 ns	251.66 ms
1	8.33 MHz	0	0	f_{sys}/8 MHz	4.17 MHz	240 ns	15.73 ms
		0	1	f_{sys}/16 MHz	2.083 MHz	480 ns	31.46 ms
		1	0	f_{sys}/64 MHz	.5208 MHz	1920 ns	125.83 ms
		1	1	f_{sys}/128 MHz	.2604 MHz	3840 ns	251.66 ms

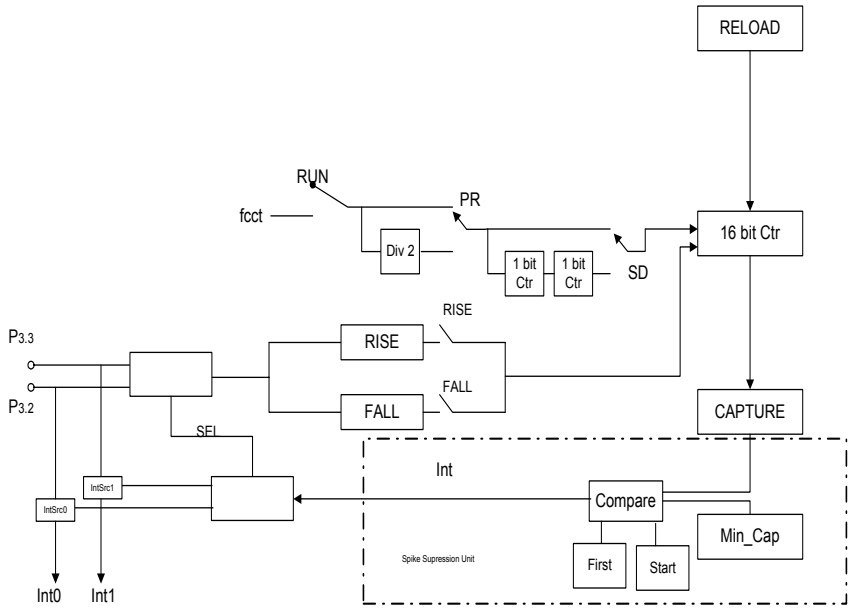


Fig. 2–7: Block Diagram

2.9. Pulse Width Modulation Unit

The Pulse Width Modulation unit consists of 6 channels with 8 bit resolution and 2 channels with 14 bit resolution PWM channels. PWM channels are programmed via special function registers and each channel can be enabled and disabled individually.

2.9.1. Reset Values

All the PWM unit registers as there are: PWME, PWCOMP8 0-5, PWCOMP14 0-1, PWMCOMPEXT14 0-1, PWML and PWMH by default are reset to 00_H.

2.9.2. Input Clock

The input clock fpwm to the PWM unit PWMU is derived from f_{sys}. f_{sys} is 33.33 MHz in normal mode and in slowdown mode 8.33 MHz.

In normal mode f_{sys} is divided by 2 and in slow down mode it is directly fed to the PWMU. Therefore PWM unit is counting at 16.5 MHz in normal mode and 8.25 MHz in slow down mode. If the PR bit PCOMPEXT14 0 (bit 0) is set the then the counting frequency is half of that.

In addition the PWM_direct bit makes it possible to run the PWM counter at system frequency, ignoring the PR bit and the built in divide by 2 prescaler.

To reduce noise radiation, the different PWM-channels are not switched 'on' simultaneously with the same counter value. The channels are switching on with one clock cycle delay to the next channel: Channel 0: 0 clock cycles delayed, Channel 1: 1 clock cycle delayed, …, Channel 5: 5 clock cycles, …, PWM14_0: 6 clock cycles, PWM14_1: 7 clock cycles delayed.

2.9.3. Port Pins

Port 1 is a dual function port. Under normal mode it works as standard Port 1, in the alternate function mode it outputs the PWM channels.

P1.0 … P1.5 corresponds to the six 8 bit resolution PWM channels PWM8_0 … PWM8_5. P1.6 and P1.7 corresponds to the two 14 bit resolution PWM channels PWM14_0 and PWM14_1. PWM channels can be individually enabled by corresponding bits in the PWME register provided the PWM_Tmr bit is not set (timer mode start bit).

2.9.4. Functional Description

2.9.4.1. 8-bit PWM

The base frequency of a 8 bit resolution DA converter channel is derived from the overflow of a six bit counter.

On every counter overflow, the enabled PWM lines would be set to 1. Except in the case it the compare value is set to zero.

In case the comparator bits (7 … 2) are set to 1, the high time of the base cycle is 63 clock cycles. In case all the comparator bits (7 … 0) including the stretching bits are set to 1, the high time of the full cycle (4 base cycles) is 255 clock cycles.

The corresponding PWCOMP8x register determines the duty cycle of the channel. If the counter value is equal to or greater than the compare value then the output channel is set to zero. The duty cycle can be adjusted in steps of fpwm as mentioned in Table 2–36.

In order to achieve the same resolution as 8-bit counter, the high time is stretched periodically by one clock cycle. The stretching cycle is determined based on the two least significant bits in the corresponding PWCOMP8x register.

The relationship for the stretching cycle can be seen in Table 2–35 and the example below.

Table 2–35: 8-bit PWM stretching cycle relationship

PWCOMP8X	Cycle Stretched
Bit 1	1, 3
Bit 0	2

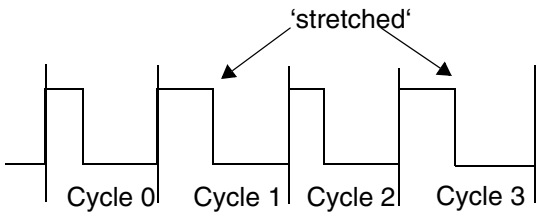


Fig. 2–8: 8-bit PWM and the Stretching Cycles

2.9.4.2. 14-bit PWM

The base frequency of a 14 bit resolution channel is derived from the overflow of a eight bit counter.

On every counter overflow, the enabled PWM lines would be set to 1 - except in the case where the compare value is set to zero.

The corresponding PWCOMP14x register determines the duty cycle of the channel. When the counter value is equal to or greater than the compare value then the output channel is set to zero. The duty cycle can be adjusted in steps of fpwm as mentioned in Table 2–36.

In order to achieve the same resolution as 14bit counter, the high time is stretched periodically by one clock cycle. Stretching cycle is determined based on the bit 7…1 in the corresponding PWCOMPEXT14x register.

Table 2–36: 14-bit PWM stretching cycle relationship

PWCOMPEXT14X	Cycle Stretched
Bit 7	1, 3, 5, 7, …, 59, 61, 63
Bit 6	2, 6, 10, …, 54, 58, 62
Bit 5	4, 12, 20, …, 52, 60
Bit 4	8, 24, 40, 56
Bit 3	16, 48
Bit 2	32

2.9.5. Cycle Time

Table 2–37: Cycle time

PWM Resolution	Slow Down (SD)	PWM_ PR	PWM_ direct	f_{sys} [MHz]	Counting Rate [MHz]	Base Cycle Time [µs]	Full Cycle Time [µs]
8 Bit	**0**	**0**	**0**	**33.33**	**16.66**	**3.84**	**15.37**
	1	0	0	8.33	8.33	7.68	30.73
	0	1	0	33.33	8.33	7.68	30.73
	1	1	0	8.33	4.16	15.37	61.46
	0	X	1	33.33	33.33	1.92	7.68
	1	X	1	8.33	8.33	7.68	30.73
14 Bit	**0**	**0**	**0**	**33.33**	**16.66**	**15.37**	**983.4**
	1	0	0	8.33	8.33	30.7	1967
	0	1	0	33.33	8.33	30.7	1967
	1	1	0	8.33	4.16	61.4	3934
	0	X	1	33.33	33.33	7.68	492
	1	X	1	8.33	8.33	30.7	1967

2.9.6. Power-Down, Idle and Power-save Mode

In idle mode the pulse width modulation unit PWMU continues to function normally, unless it has been explicitly shut down by **PSAVE**(PERI). Note that in **PSAVE** mode all channels are frozen and the pins are switched to port output mode making it possible to use the port lines.

In power-down mode the pulse width modulation unit PWMU is shut off.

2.9.7. Timer

The pulse width modulation unit PWM unit uses a single 14 bit timer to generate signals for all 8 channels. The timer is mapped into the SFR address space and hence is readable by the controller. Timer is enabled (running) if one of the PWM channels is enabled in PWME. If all the channels are disabled counter is stopped. Enabling one of the channels will reset the timer to 0 and start. Note that this reset is done for the first enabled channel. All other channels enabled later will drive the output from the current value of the counter.

If all the channels are disabled then it can be used as a general purpose timer, by enabling it with PWM_Tmr bit in PWCH.

Setting PWM_Tmr bit switches to timer mode and starts the timer. The timer always starts from a reset value of 0 (OV also reset to 0). Timer can be stopped any time by turning off the PWM_Tmr bit.

If the timer overflows it sets an over flow bit OV (bit 6) PWCH and interrupt bit **CISR0** (PWtmr) in the central interrupt register. If the corresponding interrupt enable bit IEN2(EPW) is set the interrupt will be serviced. OV bit and PWtmr bits must be reset by the software.

Note: Before utilizing the timer for PWM channels PWM_Tmr bit must be reset.
On reset the **CISR0** (PWtmr) bit is initialized to 0, however if the counter overflows this bit might be set along with OV bit. However clearing OV bit does not clear the **CISR0** (PWtmr) bit. Therefore the software must clear this bit before enabling the corresponding interrupt.

2.9.8. Control Registers

All control registers for the PWM are mapped in the SFR address space. Their address and bit description are given below.

Note that the controller can write any time into these registers. However registers PWM_COMP8_X, PWM_CPMP14_X and PWM_CPMPEXT14_X, including the bits PWM_direct and PWM_PR are double buffered and values from shadow registers are only loaded into the main register in case timer overflows or timer is stopped (PWME = 00_H) of 8 bit counter.

Overflow for 8 bit PWM occurs at the overflow of 6 bit counter and overflow for 14 bit counter occurs at the overflow.

When any of the PWM channels is not used associated compare register can be used as general purpose registers, except **PWM_En** and PWCOMPEXT14_0 bit 0 and 1.

Table 2–38: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PWM_EN	PE[7:0]							
PWM_comp8_0...	PC80_[7:0]							
PWM_comp8_1	PC81_[7:0]							
PWM_comp8_2	PC82_[7:0]							
PWM_comp8_3	PC83_[7:0]							
PWM_comp8_4	PC84_[7:0]							
PWM_comp8_5	PC85_[7:0]							
PWM_comp14_0	PC140_[7:0]							
PWM_comp14_1	PC141_[7:0]							
PWM_compext14_0	PCX140_[7:0]							
PWM_compext14_1	PCX141_[7:0							
PWM_cl	PWC_[7:0]							
PWM_ch	PWM_Tmr	OV	PWC_[13:8]					
See Section 3. on page 110 for detailed register description.								

2.10. Watchdog Timer

The Watchdog timer is a 16 bit up counter which can be programed to clock by $f_{wdt}/2$ or $f_{wdt/}128$. The current count value of the watchdog timer is contained in the watchdog timer register **WDT_high** and **WDT_low**. which are read-only register. Control and refresh function of the WDT are controlled by **WDT_refresh** and **WDT_ctrl**.

Additionally the counter can be used as a general purpose timer in timer mode. The associated load register can be used either as load register or independently as a scratch pad register by the user.

2.10.1. Input Clock

The input clock f_{wdt} is the same as the CPU clock f_{sys} divided by 12 (i.e. machine cycle). It is fed to the WDT either as divide-by-2 or divide-by-128 clock signal. The divider factor is determined by WDT_in (**WDT_ctrl**) equal 0 and 1 respectively. WDT_in has the same functionality in both watch dog mode and timer mode.

2.10.2. Starting

The watch dog timer WDT can be started if the WDT unit is in the Watch dog mode (WDT_tmr = 0).

WDT is started by setting the bit WDT_start in the **WDT_ctrl** register. Immediately after the start (1 clock cycle) the reload value from the **WDT_rel** register is copied to the **WDT_high**. **WDT_low** is always reset to 0 upon start.

Data can be written to **WDT_rel** any time during normal controller operation. Data are only loaded to the counter upon start, refresh or watchdog reset (if WDT_narst is set).

Note that the counter registers are read only and cannot be directly written to by the controller.

2.10.3. Refresh

Once the watch dog timer WDT is started it cannot be stopped by software. (Note that while the WDT is running any change to WDT_tmr bit would be ignored.) A refresh to the WDT is required before the counter overflows. Refreshing the WDT requires two instruction sequences whereby first instruction sets WDT_ref bit and the next instruction sets the WDT_start bit. (For example if there is a NOP between these two instructions, a refresh would be ignored). This double instruction refresh minimize the chances of an unintentional reset of the watchdog timer. Once set, the WDT_ref bit is reset by the hardware after three machine cycles.

A refresh causes **WDT_low** to reset to 00_H and loads the reload value to from **WDT_rel** to **WDT_high**.

2.10.4. WDT Reset

If the software fails to refresh the WDT before the counter overflows after $FFFF_H$, an internally generated watchdog reset is performed.

The watchdog timer reset differs only from the normal reset in that during normal reset all the WDT relevant bits in the three registers **WDT_rel**, **WDT_refresh**, **WDT_control** are reset to 00_H. The counter gets initialized to 0000_H.

In case of a watchdog reset, WDT_start and WDT_narst are not reset. The bit WDT_rst (read only) is set to indicate the source of the reset. In addition the WDT reset does not reset the PLL and clock generator.

If the WDT_narst bit is set then the values in the **WDT_rel** are retained after the WDT reset. The counter starts with the same pre-scaler (WDT_in) and reload configuration as before reset. If WDT_narst is not set then upon watchdog reset, **WDT_rel** is reset to 00h and WDT_in to 0.

After the WDT reset the counter starts again and must be refreshed by the microcontroller in order to avoid further WDT resets.

Duration of the WDT reset is sufficient to ensure the proper reset sequence.

2.10.5. Power-down Mode

The WDT is shut off during power down mode along with the rest of the peripherals.

In idle mode the WDT (in watchdog mode) is frozen, in timer mode it continues it's operation. In power save mode **PSAVE** (PERI) the watchdog continues it's operation. Any write access to this bit is ignored. If in timer mode the timer can be frozen by setting this bit.

2.10.6. Time Period

The period between refreshing the watchdog timer and the next overflow can be determined by the following formula.

PWDT = [2(1 + (WDT_in) × 6) × (216 - (WDT_rel) × 28)] / [FWDT]

Based on 33.33 MHz system clock minimum time period and maximum time period are as defined below.

Table 2–39: Maximum and minimum WDT overflow time period

	f_{system}	WDT_in	WDT_rel	P_{WDT}
Min.	33.33 MHz	0	FF_H	184.3 µs
Max.	33.33 MHz	1	00_H	3.02 s

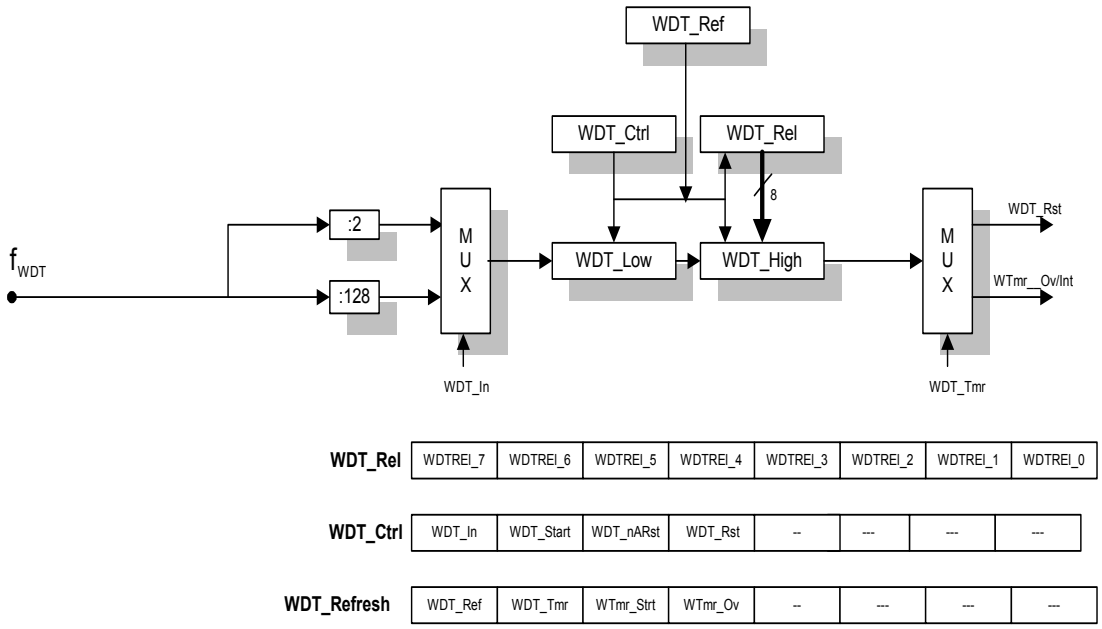


Fig. 2–9: Block Diagram

2.10.7. WDT as General Purpose Timer

The watch dog timer WDT counter can be used as a general purpose timer in timer mode. The associated load register can be used either as load register or independent scratch pad register for the programmer. This is achieved by setting WDT_tmr bit.

WDT_tmr bit can only be set before starting the WDT timer. Once the watchdog timer is started it is not possible to switch to general purpose timer mode.

If WDT_tmr bit is set then timer can be started using WTmr_strt bit.

When the timer is started it

– Resets the WTmr_ov overflow flag.

– Loads the preload value from **WDT_rel** and starts counting up.

Upon overflow the WDT_rst bit is not set neither is internal watchdog reset initiated. Overflow is indicated by the bit WTmr_ov (r/w). Overflow also sets the interrupt source bit **CISR0** (WTmr). Both of these bits are set by hardware and must be cleared by software. If the corresponding watchdog timer interrupt enable IE1 (EWT) bit is set then upon overflow the interrupt is initiated.

After an overflow the timer starts to count from **WDT_rel**. It is possible for the microcontroller to stop the timer by resetting the WTmr_strt bit any time.

While the timer is running, the WDT_tmr bit cannot be toggled. Any write access to this bit is ignored. To reset the WDT_tmr bit, either timer is stopped (WTmr_strt). However it is possible to stop the timer (WTmr_strt) and toggle the bit (WDT_tmr) with the same instruction.

2.11. Analog Digital Converter (CADC)

TVTpro includes a four channel 8-bit ADC for control purposes. By means of these four input signals the controller is able to supervise the status of up to four analog signals and take actions if necessary.

This analog signals can be connected to the Port 2 inputs without any special configuration. If the port pins of Port 2 are used as digital input, make sure that the input high level never exceeds VDDA.

The input range of the ADC is fixed to the analog supply voltage range (2.5 V nominal).

The conversion is done continuously on all four channels the results are stored in the SFRs **CADC0** … **CADC3** and updated automatically every 46 µs. An interrupt can be used to inform the microcontroller about new available results.

2.11.1. Power-down and Wake-up

During idle mode it is required to reduce the power consumption dramatically. In order to do this for the controller ADC a special wake-up unit has been included.

During this mode only the signal on input channel 1 is supervised. As soon as the input signal has fallen below a predefined level an interrupt is triggered and the system wakes up.T wo different levels are available. The first one corresponds to (fullscale-4 LSB) the second one to (fullscale-16 LSB). The actual level can be selected by a control bit (ADWULE).

Nevertheless it is possible to send even this wake-up unit into power-down (for detailed description refer to Section 2.3.20. on page 45).

2.11.2. Registers

Table 2–40: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CADC0	CADCO[7:0]							
CADC1	CADC1[7:0]							
CADC2	CADC2[7:0]							
CADC3	CADC3[7:0]							
CADCCO				ADWULE	AD[3:0]			
CISR0 bit addressable	L24	**ADC**	WTmr	AVS	DVS	PWtmr	AHS	DHS
CISR1 bit addressable	CC	**ADW**					IEX1	IEX0
PSAVE bit addressable				CADC	**WAKUP**	SLI_ACQ	DISP	PERI
PCON	SMOD	**PDS**	**IDLS**	SD	GF1	GF0	PDE	IDLE
See Section 3. on page 110 for detailed register description.								

2.12. Sync System

2.12.1. General Description

The display sync system is completely independent from the acquisition sync system (CVBS timing) and can either work as a sync master or as a sync slave system.

Talking about ‘H/V-Syncs’ in this chapter and in Section 2.13. on page 69 always refers to display related H/V Syncs and never to CVBS related sync timing.

In sync slave mode TVTpro receives the synchronization information from two independent pins which deliver separate horizontal and vertical signals or a sandcastle impulse from which the horizontal and vertical sync signals are separated internally. Due to the not line locked pixel clock generation it can process any possible horizontal and vertical sync frequency.

In sync master mode TVTpro delivers separate horizontal and vertical signals with the same flexibility in the programming of these periods as in sync slave mode.

2.12.1.1. Screen Resolution

The number of displayable pixels on the screen is defined by the pixel frequency (which is independent from horizontal frequency), the line period and number of lines within a field. The screen is divided in three different regions:

2.12.1.1.1. Blacklevel Clamping Area

During horizontal and vertical blacklevel clamping, the black value (RGB = 000) is delivered on output side of TVTPro. Inside this area the BLANK pin and COR pin are set to the same values which are defined as transparency for subCLUT0 (see also Section 2.13.7.5. on page 88). This area is programmable in vertical direction (in terms of lines) and in horizontal direction in terms of 33.33 MHz clock cycles.

2.12.1.1.2. Border Area

The size of this area is defined by the sync delay registers (SDH and SDV) and the size of the character display area. The color and transparency of this area is defined by a color look up vector. See Section 2.13.7. on page 79).

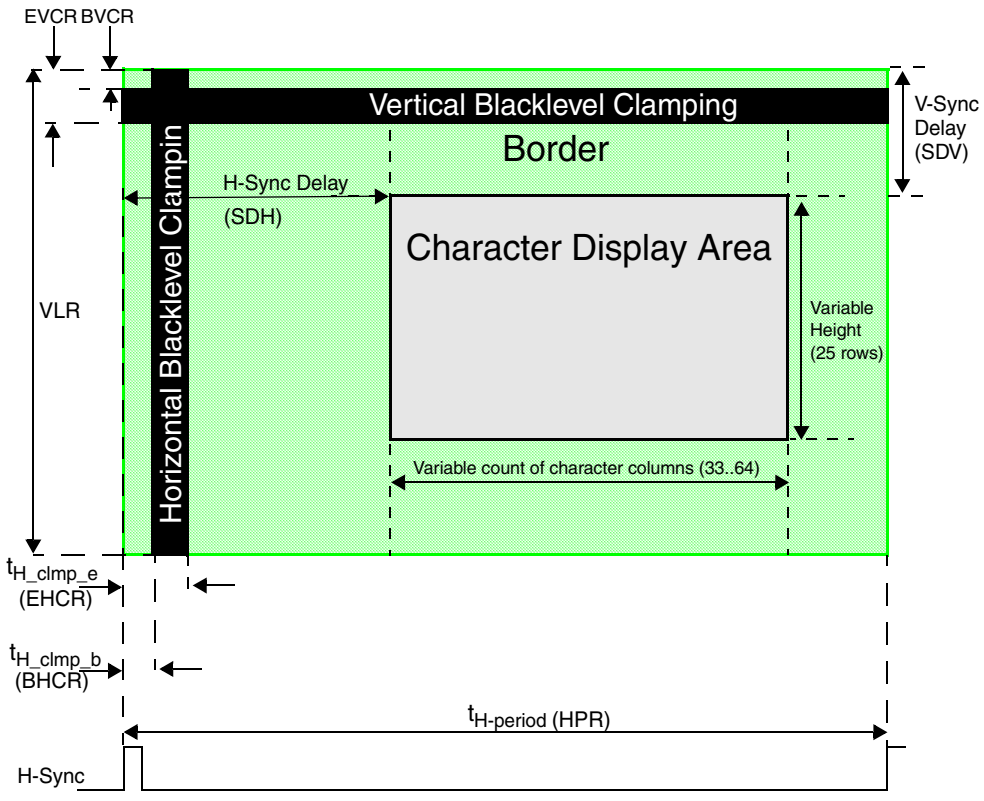


Fig. 2–10: TVTPro’s Display Timing

2.12.1.1.3. Character Display Area

Characters and their attributes which are displayed inside this area are free programmable according to the specifications of the display generator (see also Section 2.13.2. on page 69). The start position of that area can be shifted in horizontal and vertical direction by programming the horizontal and vertical sync delay registers (SDH and SDV). The size of that area is defined by the instruction FSR in the display generator.

Registers which allow to set up the screen and sync parameters are given in Table 2–41.

The user has to take care of setting PFR and SDH so that SDH/PFR is greater than 2 µs.

Table 2–42 lists some of the possible display modes.

Note that the Pixel clock (Pclk) must be appropriately selected to the nearest value in the registers Pclk 0 and Pclk 1.

Table 2–42 serves as an example,. The freely programmable Pixel clock between 10 to 32 MHz makes it possible to adjust and fine tune the display as per application requirement.

2.12.1.2. Sync Interrupts

The sync unit delivers interrupts (Horizontal and vertical interrupt) to the controller to support the recognition of the frequency of an external sync source. These interrupts are related to the positive edge of the non delayed horizontal and vertical impulses which can be seen at pins HSYNC and VSYNC.

Table 2–41: Overview on sync register settings

Parameters	Register	Min. Value	Max. Value	Step	Default
Sync Control Register	SCR	No min/max general setup			
VL - Lines / Field	VLR	1 line	1024 lines	1 line	625 lines
$T_{h\text{-}period}$ - Horizontal Period	HPR	15 µs	122.8 µs	30 ns	64 µs
F_{pixel} - Pixel Frequency	PClk	10 MHz	32 MHz	73.25 kHz	12.01 MHz
T_{vsync_delay} - Sync Delay	SDV	4 lines	1024 lines	1 line	32 lines
T_{hsync_delay} - Sync Delay	SDH	32 pixel	2048 pixel	1 pixel	72 pixel
BVCR - Beginning of Vertical Clamp Phase	BVCR	1 line	1024 lines	1 line	line 0
EVCR - End of Vertical Clamp Phase	EVCR	1 line	1024 lines	1 line	line 4
$T_{h\_clmp\_b}$ - Beginning of Horizontal Clamp Phase	BHCR	0 µs	122.8 µs	480 ns	0 µs
$T_{h\_clmp\_e}$ - End of Horizontal Clamp Phase	EHCR	0 µs	122.8 µs	480 ns	4.8 µs

Table 2–42: Possible display modes

50 Hz/100 Hz	Character Display Mode	Pclk	$T_{Character\ display\ area}$
50 Hz	40 × 25	12 MHz	40 µs
50 Hz	64 × 25	16 MHz	48 µs
100 Hz	40 × 25	24 MHz	20 µs
100 Hz	64 × 25	32 MHz	24 µs

2.12.1.3. Related Registers

Table 2–43: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCR1	Reserved	RGB_G_[1:0]		COR_BL	VSU[3:0]			
SCR0	RGB_D_[1:0]		HP	VP	INT	SNC	VCS	MAST
CISR0 bit addressable	L24	ADC	WTmr	AVS	**DVS**	PWtmr	AHS	**DHS**
VLR1		ODD_Ev	VSU[3:0]				VLR[9:8]	
VLR0	VLR[7:0]							
HPR1					HPR[11:8]			
HPR0	HPR[7:0]							
SDV1							SDV[9:8]	
SDV0	SDV[7:0]							
SDH1					SDH[11:8]			
SDH0	SDH[7:0]							
HCR1	EHCR[7:0]							
HCR0	BHCR[7:0]							
BVCR							BVCR[9:8]	
BVCR0	BVCR[7:0]							
EVCR1							EVCR[9:8]	
EVCR0	EVCR[7:0]							
SNDCSTL		HYS	SND_V[2:0]			SND_H[2:0]		
CSCR0			ENETCLK	ENERCLK	$\overline{\text{PA_7}}$_Alt	VS_$\overline{\text{OE}}$	O_E_$\overline{\text{P3_0}}$	O_E_Pol

See Section 3. on page 110 for detailed register description.

DHS is used as an interface from H input pin to software interrupt routines.
DVS is used as an interface from V input pin to software interrupt routines.
These interrupt routines can be used for detection of the frequency of an external sync source. It is set by the HW and must be reset by the SW. The clamp phase area has higher priority than the screen background area or the character display area and can be shifted independent from any other register.

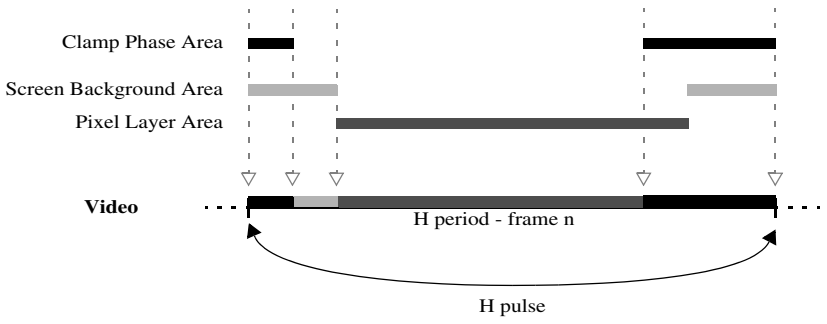


Fig. 2–11: Priority of Clamp Phase, Screen Background and Pixel Layer Area.

2.13. Display

The display is based on the requirements for a Level 1.5 Teletext and powerful additional enhanced OSD features.

The display circuit reads the contents and attribute settings of the display memory and generates the RGB data for a TV back-end signal processing unit.

The display can be synchronized to external H/V sync signals (slave mode) or can generate the synchronization signals by itself (master mode). The display can be synchronized to 50 Hz as well as to 60 Hz systems. Interlaced display is supported for interlaced sync sources and non-interlaced ones.

2.13.1. Display Features

- Teletext Level 1.5 feature set
- ROM Character Set to Support all European Languages in Parallel
- Mosaic Graphic Character Set
- Parallel Display Attributes
- Single/Double Width/Height of Characters
- Variable Flash Rate
- Programmable Screen Size (25 Rows × 33 … 64 Columns)
- Flexible Character Matrixes (HxV) 12 × 9 … 16
- Up to 256 Dynamically Redefinable Characters in standard mode;
  1024 Dynamically Redefinable Characters in Enhanced Mode
- Up to 16 Colors for DRCS Character
- One out of Eight Colors for Foreground and Background Colors for 1-bit DRCS and ROM Characters
- Shadowing
- Contrast Reduction
- Pixel by Pixel Shiftable Cursor With up to 4 Different Colors
- Support of Progressive Scan

Table 2–44: Display memory organization of TVTpro

Row No.	Address	i = 0d	…	i = 39d
0	$DISPOINT_H + 0_H + i \times 3_H$	Character Display Area		
1	$DISPOINT_h + 78_H + i \times 3_H$			
2	$DISPOINT_h + F0_H + i \times 3_H$			
…	…			
23	$DISPOINT_H + AC8_H + i \times 3_H$			
24	$DISPOINT_H + B40_H + i \times 3_H$			

2.13.3. Display Memory

The display memory is located inside the internal XRAM.

The start address of the display memory is at memory address $DISPOINT_H$. This memory address is defined by the user due to a pointer. For each character position three Bytes in the display memory are reserved. These three Bytes are stored in a serial incremental order for each character and used to define the display attributes of each single character position. The complete amount of allocated display memory depends on the display resolution. In vertical direction the character display area is fixed to 25 rows. In horizontal direction the character display area can be adjusted from 33 up to 64 columns. Table 2–44 is an example for a character display area resolution of 25 rows and 40 columns.

2.13.4. Parallel Character Attributes

The character display area content of each character position is defined by a 3 Byte character display word (CDW; see also Section 2.13.9.1. on page 102) in display memory.

Following formula helps to calculate a memory address of a character position (X_{CH}, Y_{CH}) depending on the count of characters in horizontal direction (defined in the binary parameters $(DISALH4 \ldots DISALH0)_H$) and a display start address $DISPOINT_H$:

$$CHARADDRESS_H = DISPOINT_H + (Y_{CH} \times ((DISALH4 \ldots DISALH0)_H + 21_H) + X_{CH} \times 3_H)$$

Table 2–45: Character display word: RAM location: Display memory

Byte Pos.	Bit	Name	Function	Remark
0	0	CHAR0	Used to choose a ROM or DRCS character	DRCS characters are defined by the user. Up to 16 different colors can be used within one DRCS; see also see Section 2.13.4.1.
	1	CHAR1		
	2	CHAR2		
	3	CHAR3		
	4	CHAR4		
	5	CHAR5		
	6	CHAR6		
	7	CHAR7		
1	8	CHAR8		
	9	CHAR9		
	10	FLASH	Control of flash modes	See also Section 2.13.4.4.
	11	UH	Upper half double height	See also Section 2.13.4.5.
	12	DH	Double height	See also Section 2.13.4.5.
	13	DW	Double width	See also Section 2.13.4.6.
	14	BOX	Control for Boxes	See also Section 2.13.7.4.
	15	CLUT0	Bit0/CLUT select	See also Section 2.13.7.5.

Table 2–45: Character display word: RAM location: Display memory, continued

Byte Pos.	Bit	Name	Function	Remark
2	16	CLUT1	Bit1/CLUT select	See also Section 2.13.7.5.
	17	CLUT2	Bit2/CLUT select	See also Section 2.13.7.5.
	18	FG0	Foreground color vector	Only used for ROM characters and 1-bit DRCS characters; Foreground-color is chosen if bit inside ROM-mask/RAM is set to '1' See also Section 2.13.7.5.
	19	FG1		
	20	FG2		
	21	BG0	Background color vector	Used for ROM characters and 1-bit DRCS characters; For 2-bit and 4-bit DRCS characters only used in flash mode; Background color is chosen if bit inside ROM-mask/RAM is set to '0'; See also Section 2.13.7.5.and Section 2.13.4.4.
	22	BG1		
	23	BG2		

2.13.4.1. Access of Characters

The DRCS characters and ROM characters are accessed by a 10-bit character address inside the character display word (CDW; see also Section 2.13.9.1. on page 102).

2.13.4.2. Address Range from 0_d to 767_d

This address range can either be used to access ROM characters or to access 1-bit DRCS characters.

See also Section 2.13.5. on page 75 Global Display Word (GDW).

Table 2–46: Definition of character access mode

CHAAC	Description
0	Normal mode: Address range 0_d - 767_d is used to access ROM characters.
1	Enhanced mode: Address range 0_d - 767_d is used to access 1-bit DRCS characters.

2.13.4.3. Address Range from 768_d to 1023_d

The address range from 768_d to 1023_d is reserved to address the DRCS characters. This range is split into three parts for 1-bit DRCS, 2-bit DRCS and 4-bit DRCS. The boundary between 1-bit DRCS and 2-bit DRCS as well as the boundary between 2-bit DRCS and 4-bit DRCS are defined by two boundary pointers inside the global display word (GDW) (see also Section 2.13.5.)

Table 2–47: Boundary pointer 1

DRCS B1_3	DRCS B1_2	DRCS B1_1	DRCS B1_0	Description
0	0	0	0	Boundary1 set to 768_d
0	0	0	1	Boundary1 set to 784_d
0	0	1	0	Boundary1 set to 800_d
0	0	1	1	Boundary1 set to 816_d
…				…
1	1	1	0	Boundary1 set to 992_d
1	1	1	1	Boundary1 set to 1008_d
See also Section 2.13.5. on page 75 / Global Display Word (GDW)				

Table 2–48: Boundary pointer 2

DRCS B2_3	DRCS B2_2	DRCS B2_1	DRCS B2_0	Description
0	0	0	0	Boundary1 set to 768_d
0	0	0	1	Boundary1 set to 784_d
0	0	1	0	Boundary1 set to 800_d
0	0	1	1	Boundary1 set to 816_d
…				…
1	1	1	0	Boundary1 set to 992_d
1	1	1	1	Boundary1 set to 1008_d
Please notice: DRCSB2_3 … DRCSB2_0 must be set to a greater or a equal value than DRCSB1_3 … DRCSB1_0. See also Section 2.13.5. on page 75 / Global Display Word (GDW)				

Below some examples can be found to show how the character addressing depends on the boundary definitions:

2.13.4.3.1. Example 1

Boundary Pointer 1 set to 848_d
Boundary Pointer 2 set to 928_d

Character Address		Description
From	To	
768_d	847_d	1-bit DRCS characters
848_d	991_d	2-bit DRCS characters
928_d	1023_d	4-bit DRCS characters

2.13.4.3.2. Example 2

Boundary Pointer1 set to 848_d
Boundary Pointer2 set to 848d

Character Address		Description
From	To	
768_d	847_d	1-bit DRCS characters
848_d	1023_d	4-bit DRCS characters

2.13.4.3.3. Example 3

Boundary Pointer 1 set to 768_d
Boundary Pointer 2 set to 928_d

Character Address		Description
From	To	
768_d	927_d	2-bit DRCS characters
928_d	1023_d	4-bit DRCS characters

2.13.4.4. Flash

The bit FLASH inside the character display word (CDW; see also Section 2.13.4.) is used to enable flash for a character.

FLASH	Description
0	Steady (flash disabled)
1	Flash
See also Section 2.13.4. on page 70 / Character Display Word (CDW)	

The meaning of the flash attribute is different for ROM characters and 1-bit DRCS characters in comparison to the meaning of flash for 2-bit and 4-bit DRCS characters.

For flash rate control see also the global attribute "FLRATE1 … FLRATE0" in Section 2.13.7.3..

2.13.4.4.1. Flash for ROM Characters and 1-Bit DRCS Characters

For ROM characters and 1-bit DRCS characters the enabled flash mode causes the foreground pixels to alternate between the foreground and background color vector.

2.13.4.4.2. Flash for 2-Bit and 4-Bit DRCS Characters

For these characters the enabled flash mode causes the DRCS pixels to alternate between the 2-bit/ 4-bit color vector and the background color vector which is defined by the parameters BG2 … BG0 inside character display word (CDW; see also Section 2.13.4. on page 70).

2.13.4.5. Character Individual Double Height

Bit UH (Upper half, double height) marks the upper part of a double height character. It is only active, if the DH bit (Double Height) is set to ‘1’.

Table 2–49 shows the influence of the DH bit and the UH bit on the character ‘A’.

Table 2–49: Character individual double height

DH	UH	Display
0	X	
1	1	
1	0	
See also Section 2.13.4. on page 70 / Character Display Word (CDW)		

2.13.4.6. Character Individual Double Width

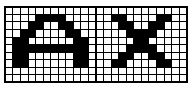
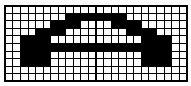
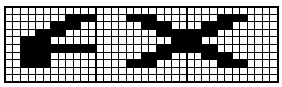
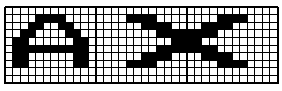
The bit DW (double width) marks the left half of a character with double width.

The character to its right will be overwritten by the right half.

If the DW bit of the following character (here the ‘X’) is also set to ‘1’; the right half of the ‘A’ is overwritten by the left half of the ‘X’.

If a character is displayed in double width mode the attribute settings of the left character position are used to display the whole character.

Table 2–50: Character individual double width

DW Bit		Display
Left Character	Right Character	
0	0	
1	0	
1	1	
0	1	
See also Section 2.13.4. on page 70 / Character Display Word (CDW)		

2.13.5. Global OSD Attributes

Next to the parallel attributes stored inside character display word there are global attributes. The settings of the global attributes affect the full screen.

The settings of the global OSD attributes are stored in the global display word (GDW; see also Section 2.13.5.) within 10 Bytes in the XRAM. The location of the GDW is defined by a programmable pointer (See also Section 2.13.9. on page 100).

Table 2–51: Global OSD attributes

Byte Pos.	Bit	Name	Function	Cross Reference
0	0	DISALH0	Count of display columns in horizontal direction	See also Section 2.13.6. on page 79
	1	DISALH1		
	2	DISALH2		
	3	DISALH3		
	4	DISALH4		
	5	PROGRESS	Used to enable progressive scan mode.	See also Section 2.13.7.8. on page 96
	6	---	Reserved.	---
	7	---	Reserved.	---
1	0	CURSEN	Enables cursor function.	See also Section 2.13.7. on page 79
	1	CURHOR0	Horizontal pixel shift of cursor to character position	
	2	CURHOR1		
	3	CURHOR2		
	4	CURHOR3		
	5	CURVER0	Vertical pixel shift of cursor to character position	
	6	CURVER1		
	7	CURVER2		

Table 2–51: Global OSD attributes, continued

Byte Pos.	Bit	Name	Function	Cross Reference
2	0	CURVER3	Vertical pixel shift of cursor to character position	See also Section 2.13.7. on page 79
	1	POSHOR0	Horizontal character position of cursor	
	2	POSHOR1		
	3	POSHOR2		
	4	POSHOR3		
	5	POSHOR4		
	6	POSHOR5		
	7	POSVER0	Vertical character position of cursor	
3	0	POSVER1		
	1	POSVER2		
	2	POSVER3		
	3	POSVER4		
	4	GLBT0_BOX1	Used to enable transparency of Box1. CLUT transparency of subCLUT0 can be overruled for destined pixels inside Box1.	See also Section 2.13.7.4. on page 86
	5	GLBT1_BOX1		
	6	GLBT2_BOX1		
	7	---	Reserved.	---
4	0	BRDCOL0	Color vector of border	See also Section 2.13.7.1. on page 83
	1	BRDCOL1		
	2	BRDCOL2		
	3	BRDCOL3		
	4	BRDCOL4		
	5	BRDCOL5		
	6	BLA_BOX1	Used to define the overruling transparency levels for Box1.	See also Section 2.13.7.4. on page 86
	7	COR_BOX1		

Table 2–51: Global OSD attributes, continued

Byte Pos.	Bit	Name	Function	Cross Reference
5	0	GDDH0	Double height of the full screen	See also Section 2.13.7.2. on page 83
	1	GDDH1		
	2	GDDH2		
	3	GLBT0_BOX0	Used to enable transparency of Box0. CLUT transparency of subCLUT0 can be overruled for destined pixels inside Box0.	See also Section 2.13.7.4. on page 86
	4	GLBT1_BOX0		
	5	GLBT2_BOX0		
	6	BLA_BOX0	Used to define the overruling transparency levels for Box0.	See also Section 2.13.7.4. on page 86
	7	COR_BOX0		
6	0	CHADRC0	Defines vertical resolution of DRCS characters.	See also Section 2.13.7.6. on page 94
	1	CHADRC1		
	2	CHADRC2		
	3	CHAROM0	Defines vertical resolution of ROM characters.	
	4	CHAROM1		
	5	CHAROM2		
	6	CHAAC	Defines character access mode.	See also Section 2.13.4.1. on page 71
	7	---	Reserved.	---
7	0	DRCSB1_0	Used to define the boundary pointer 1 for DRCS addressing.	See also Section 2.13.4.1. on page 71
	1	DRCSB1_1		
	2	DRCSB1_2		
	3	DRCSB1_3		
	4	DRCSB2_0	Used to define the boundary pointer 2 for DRCS addressing.	See also Section 2.13.4.1. on page 71
	5	DRCSB2_1		
	6	DRCSB2_2		
	7	DRCSB2_3		

Table 2–51: Global OSD attributes, continued

Byte Pos.	Bit	Name	Function	Cross Reference
8	0	SHEN	Enables shadow.	See also Section 2.13.7.7. on page 95
	1	SHEAWE	Defines if east or west shadow is processed.	
	2	SHCOL0	Defines the shadow color vector.	
	3	SHCOL1		
	4	SHCOL2		
	5	SHCOL3		
	6	SHCOL4		
	7	SHCOL5		
9	0	CURCLUT0	Used to choose the foreground vector for the cursor (0 … 63).	See also Section 2.13.7. on page 79
	1	CURCLUT1		
	2	CURCLUT2		
	3	FLRATE0	Defines the flash rate for flashing characters.	See also Section 2.13.7.3. on page 85
	4	FLRATE1		
	5	HDWCLUTCOR	Defines the level of COR for the colors of the hardwired CLUT.	See also Section 2.13.7.5. on page 88
	6	HDWCLUTBLANK	Defines the level of BLANK for the colors of the hardwired CLUT.	See also Section 2.13.7.5. on page 88
	7	---	Reserved.	---

2.13.6. Character Display Area Resolution

Table 2–52: Character display area resolution

DISALH4	DISALH3	DISALH2	DISALH1	DISALH0	Description
0	0	0	0	0	33 columns
0	0	0	0	1	34 columns
0	0	0	1	0	35 columns
…					…
0	1	1	1	1	48 columns
1	0	0	0	0	49 columns
…					…
1	1	1	1	0	63 columns
1	1	1	1	1	64 columns
See also Section 2.13.5. / Global Display Word (GDW)					

The count of rows of the character display area can be adjusted in a range from 33 to 64 columns in horizontal direction. In vertical direction the character display area is fixed to 25 rows. It depends on the settings for synchronization mode, pixel frequency and character matrix if all these columns are visible on the tube.

The programmable parameters DISALH4 to DISALH0 are the binary representation of an offset value. This offset value plus 33_d gives the count of columns:

Table 2–52 shows some examples for the settings.

2.13.7. Cursor

The 2-bit color vector matrix of the cursor is stored in the XRAM. A programmable pointer is used, so that the matrix can be stored at any location inside the XRAM (see also Section 2.13.9.3. on page 102).

The cursor matrix has the same resolution as the character matrix (see also Section 2.13.7.6. on page 94).

If the Global Display Double Height (see also Section 2.13.7.2. on page 83) is set to double height, the rows which are displayed in double height the cursor is also displayed in double height. For rows which are displayed in normal height, the cursor is also displayed in normal height. If cursor is displayed over two rows and one of these rows is displayed in double height, and the other is displayed in normal height, cursor is also partly displayed in double height and partly in normal height. Cursor-Pixels which are shifted to a non-visible row are also not displayed on the screen.

The cursor can be shifted in horizontal and vertical direction pixel by pixel all over the character display area.

Table 2–53: Setting of CURSEN to enable cursor mode

CURSEN	Description
0	Cursor mode disabled
1	Cursor mode enabled
See also Section 2.13.5. on page 75 / Global Display Word (GDW)	

The display position of the cursor is determined by a display column value, a display row value and on pixel level by a pixel shift in horizontal and vertical direction.

The cursor can not be shifted more than one character height and one character width on pixel level. The cursor is clipped at the border of the display area. In full screen double height mode (See also Section 2.13.7.2. on page 83) the cursor is also displayed in double height.

The pixel shift value is always related to a south-east shift. The pixel shift is determined by the parameters shown in Table 2–54 and Table 2–55 on page 80.

Table 2–54: Horizontal cursor pixel offset within character matrix

CURHOR3	CURHOR2	CURHOR1	CURHOR0	Description
0	0	0	0	Horizontal shift of 0
0	0	0	1	Horizontal shift of 1
0	0	1	0	Horizontal shift of 2
0	0	1	1	Horizontal shift of 3
…				…
1	0	1	1	Horizontal shift of 11
1	1	X	X	Not allowed
See also Section 2.13.5. on page 75-Global Display Word (GDW)				

Table 2–55: Vertical cursor pixel offset within character matrix

CURVER3	CURVER2	CURVER1	CURVER0	Description
0	0	0	0	Vertical shift of 0
0	0	0	1	Vertical shift of 1
0	0	1	0	Vertical shift of 2
0	0	1	1	Vertical shift of 3
…				…
1	1	1	0	Vertical shift of 14
1	1	1	1	Vertical shift of 15
See also Section 2.13.5. on page 75-Global Display Word (GDW)				

The character position of the cursor is determined by the parameters shown in Table 2–56 and Table 2–57.

Character position and pixel position have to be changed in parallel. Otherwise it may appear that the character position already has been changed to a new position, but the pixel position is still set to the former value. This may cause a “jumping” cursor.

To avoid this “jumping” cursor there is a EN_LD_GDW (enable load GDW) bit in the SFR bank. If this bit is set to ‘0’ the global display word can be changed without any effect on the screen and in consequence the cursor position can be changed without any effect on the screen. To bring the effect to character display area, the LOAD bit has to be set to 1 for at least one V period (approximately 50 ms).

The cursor is handled as a layer above the character display area. Pixels of the 2-bit cursor bit plane which are set to ‘00’ are transparent to the OSD/Video layer below. So the cursor can be transparent to the OSD (in case of no transparency of OSD) or to video (in case of transparency of OSD).

Table 2–56: Horizontal character position of the cursor within the character matrix

POS HOR5	POS HOR4	POS HOR3	POS HOR2	POS HOR1	POS HOR0	Description
0	0	0	0	0	0	Horizontal character column 0
0	0	0	0	0	1	Horizontal character column 1
…						…
1	1	1	1	1	0	Horizontal character column 62
1	1	1	1	1	1	Horizontal character column 63
See also Section 2.13.5. on page 75-Global Display Word (GDW)						

Table 2–57: Vertical character position of the cursor within the character matrix

POS VER4	POS VER3	POS VER2	POS VER1	POS VER0	Description
0	0	0	0	0	Vertical character row 0
0	0	0	0	1	Vertical character row 1
0	0	0	1	0	Vertical character row 2
0	0	0	1	1	Vertical character row 3
…					…
1	1	1	1	0	Vertical character row 30
1	1	1	1	1	Vertical character row 31
See also Section 2.13.5. on page 75-Global Display Word (GDW)					

Example:

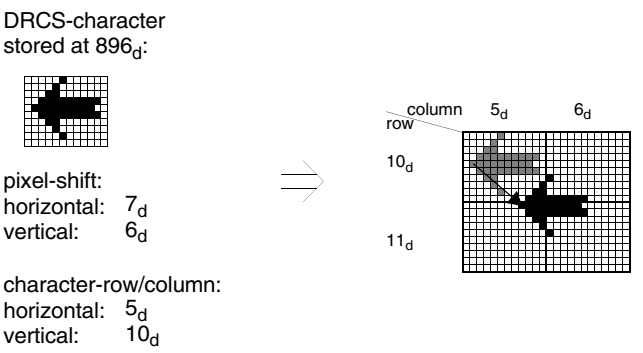


Fig. 2–12: Positioning of HW Cursor

One out of 8 subCLUTs is used to display the cursor. The parameters CURCLUT2 … CURCLUT0 are used to define the subCLUT to be used.

Table 2–58: CURCLUT2, CURCLUT1, CURCLUT0

CUR CLUT2	CUR CLUT1	CUR CLUT0	Description
0	0	0	Used to select the subCLUT which is used for color look up of the cursor (0 … 7)
0	0	1	
0	1	0	
0	1	1	
…			
1	1	0	
1	1	1	
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

For detailed information of CLUT access see Section 2.13.7.5. on page 88

2.13.7.1. Border Color

Table 2–59: Border color settings

BRDCOL5	BRDCOL4	BRDCOL3	BRDCOL2	BRDCOL1	BRDCOL0	Description
0	0	0	0	0	0	Defines a color vector for the border; see also Section 2.13.7.5. on page 88
0	0	0	0	0	1	
0	0	0	0	1	0	
0	0	0	0	1	1	
…						
1	1	1	1	1	0	
1	1	1	1	1	1	
See also Section 2.13.5. on page 75-Global Display Word (GDW)						

Next to the character display area in which the characters are displayed there is an area which is surrounding the character display area. The visibility of this border area depends on the width and height of the character display area. The user is free to define the color vector of this border.

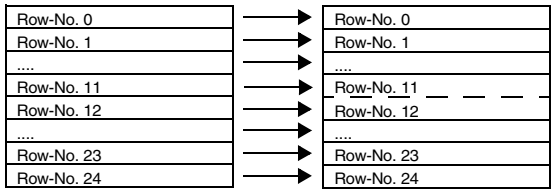
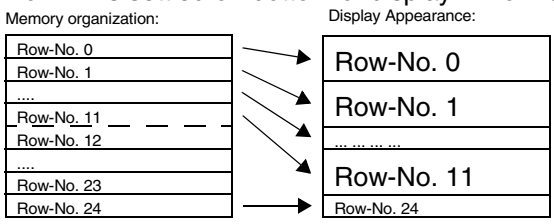
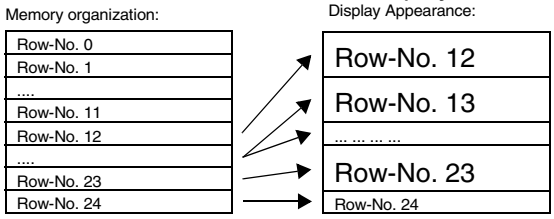
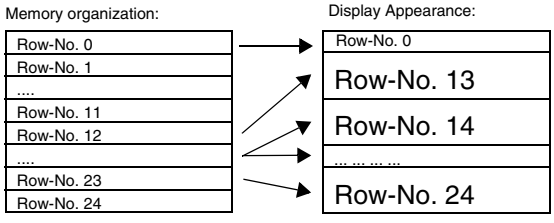
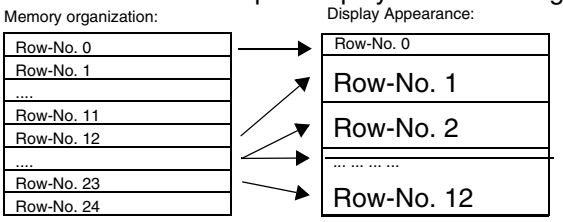
2.13.7.2. Full Screen Double Height

If double height is enabled for the full screen each line of the OSD is repeated twice at the RGB output. As a result, characters which are normally displayed in normal height, are now displayed in double height and characters which are normally displayed in double height are now displayed in quadruple height.

Row 0 and 24 are handled in a special way. If double height is selected for the full screen these two rows can be fixed to normal display (each line of these rows is repeated only once).

In double height mode the user may want to start the processing of the display at row 12 and not at row 0. To decide this, three bits are used as a global attribute.

Table 2–60: Full screen double height

GDDH2	GDDH1	GDDH0	Display Area
0	0	0	Full Screen Normal Height: 
0	0	1	Full Screen Double Height: Rows 0-11 are displayed in double height. Row 24 is settled on bottom of display in normal height. 
0	1	0	Full Screen Double Height: Rows 12-23 are displayed in double height. Row 24 is settled on bottom of display in normal height. 
0	1	1	Full Screen Double Height: Rows 13-24 are displayed in double height. Row 0 is settled on top of display in normal height. 
1	X	X	Full Screen Double Height: Rows 1-12 are displayed in double height. Row 0 is settled on top of display in normal height. 
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

2.13.7.3. Flash Rate Control

This attribute is used to control the flash rate for the full screen. All the characters on the screen for which flash is enabled are flashing with same frequency and in same phase.

Table 2–61: Flash rate control

FLRATE1	FLRATE0	Description
0	0	Slow flash rate. The flash rate is derived from display V pulse. For 50 Hz systems Flash rate is approximately 0.5 Hz. Duty cycle is approximately 50%.
0	1	Medium flash rate. The flash rate is derived from the V pulse. For 50 Hz systems Flash rate is approximately 1.0 Hz. Duty cycle is approximately 50%.
1	X	Fast flash rate. The flash rate is derived from the V pulse. For 50 Hz systems Flash rate is approximately 2.0 Hz. Duty cycle is approximately 50%.
See also Section 2.13.5. on page 75-Global Display Word (GDW)		

2.13.7.4. Transparency of Boxes

For characters which are using subCLUT0 the transparency which is defined for the whole CLUT (see also Section 2.13.7.5. on page 88) can be overruled for foreground or background pixels. There are two different definitions for two box areas to define this overruling. Which of these two box transparencies is used, is selected character individual inside the bit BOX in CDW (character display word; See also Section 2.13.4.)

Transparency definition for characters for BOX0:
The cursor (see also Section 2.13.7. on page 79) is not affected by these bits.

Table 2–62: Transparency mode of BOX0

GLBT2_BOX0	GLBT1_BOX0	GLBT0_BOX0	Description
X	0	0	Box transparency is disabled for BOX0. For all pixels the global defined transparency of subCLUT0 is used.
0	0	1	Box transparency is enabled for BOX0 for following pixels: Foreground pixels of ROM characters
0	1	0	Box transparency is enabled for BOX0 for following pixels: Foreground pixels of 1-bit DRCS characters
0	1	1	Box transparency is enabled for BOX0 for following pixels: Foreground pixels of ROM characters Foreground pixels of 1-bit DRCS characters
1	0	1	Box transparency is enabled for BOX0 for following pixels: Background pixels of ROM characters
1	1	0	Box transparency is enabled for BOX0 for following pixels: Background pixels of 1-bit DRCS characters
1	1	1	Box transparency is enabled for BOX0 for following pixels: Background pixels of ROM characters Background pixels of 1-bit DRCS characters
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

To decide the levels of COR and BLANK for BOX0 two global parameters are used.

Table 2–63: COR/BLANK polarity of BOX0

COR_BOX0	BLA_BOX0	Description
0	0	Box transparency levels of COR and BLANK are overruled by: COR = 0; BLANK = 0
0	1	Box transparency levels of COR and BLANK are overruled by: COR = 0; BLANK = 1
1	0	Box transparency levels of COR and BLANK are overruled by: COR = 1; BLANK = 0
1	1	Box transparency levels of COR and BLANK are overruled by: COR = 1; BLANK = 1
See also Section 2.13.5. on page 75-Global Display Word (GDW)		

For characters which are using subCLUT0 there are two types of transparency which can be defined. Which of these two box transparencies is used is defined character individual inside the bit BOX in CDW (character display word; see also Section 2.13.4. on page 70).

Transparency definition for characters for which BOX is set to 1 and which are using subCLUT0.

Table 2–64: Transparency mode of BOX1

GLBT2_BOX1	GLBT1_BOX1	GLBT0_BOX1	Description
X	0	0	Box transparency is disabled for BOX1.
0	0	1	Box transparency is enabled for BOX1 for following pixels: Foreground pixels of ROM characters
0	1	0	Box transparency is enabled for BOX1 for following pixels: Foreground pixels of 1-bit DRCS characters
0	1	1	Box transparency is enabled for BOX1 for following pixels: Foreground pixels of ROM characters Foreground pixels of 1-bit DRCS characters
1	0	1	Box transparency is enabled for BOX1 for following pixels: Background pixels of ROM characters
1	1	0	Box transparency is enabled for BOX1 for following pixels: Background pixels of 1-bit DRCS characters
1	1	1	Box transparency is enabled for BOX1 for following pixels: Background pixels of ROM characters Background pixels of 1-bit DRCS characters
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

To decide the levels of COR and BLANK for BOX1 two global parameters are used.

Table 2–65: COR/BLANK polarity of BOX1

COR_BOX1	BLA_BOX1	Description
0	0	Box transparency levels of COR and BLANK for BOX1 are overruled by: COR = 0; BLANK = 0
0	1	Box transparency levels of COR and BLANK coming from CLUT0 inside BOX1 are overruled by: COR = 0; BLANK = 1
1	0	Box transparency levels of COR and BLANK coming from CLUT0 inside BOX1 are overruled by: COR = 1; BLANK = 0
1	1	Box transparency levels of COR and BLANK coming from CLUT0 inside BOX1 are overruled by: COR = 1; BLANK = 1
See also Section 2.13.5. on page 75-Global Display Word (GDW)		

2.13.7.5. CLUT

Table 2–66: COR/BLANK polarity setup for hardware CLUT during black clamp phase

HDWCLUTCOR	HDWCLUTBLANK	Description
0	0	Decides the polarity for COR and BLANK output for the hardwired CLUT entries 0-15 and the polarity of COR and BLANK during black clamp phase (See also Section 2.12.1. on page 66): COR = 0 BLANK = 0
0	1	Decides the polarity for COR and BLANK output for the hardwired CLUT entries 0-15 and the polarity of COR and BLANK during black clamp phase (See also Section 2.12.1. on page 66): COR = 0 BLANK = 1
1	0	Decides the polarity for COR and BLANK output for the hardwired CLUT entries 0-15 and the polarity of COR and BLANK during black clamp phase (See also Section 2.12.1. on page 66): COR = 1 BLANK = 0
1	1	Decides the polarity for COR and BLANK output for the hardwired CLUT entries 0-15 and the polarity of COR and BLANK during black clamp phase (See also Section 2.12.1. on page 66): COR = 1 BLANK = 1

The CLUT has a maximum width of 64 entries. The RGB values of the CLUT entries from 0-15 are hardwired and can not be changed by software. The transparency for the hardwired CLUT values are set by a global attribute inside the global display word (GDW; see also Section 2.13.5. on page 75). This global setting can be overruled inside of boxes (see also Section 2.13.7.4. on page 86)

The RGB values of the CLUT entries from 16 to 63 are free programmable. The RGB values of the CLUT are organized in the TVTpro XRAM in a incremental serial order. CLUT locations inside XRAM which are not used for OSD can be used for any other storage purposes.

The CLUT is divided in 8 subCLUTs with 8 entries for 1-bit DRCS and ROM characters. For 2-bit DRCS characters the CLUT is divided in 8 subCLUTs with 4 entries. For 4-bit DRCS characters the CLUT is divided in 4 subCLUTs with 16 different entries.

The subCLUTs can be selected for each character position individual. For this three bits CLUT2, CLUT1 and CLUT0 are reserved inside the character display word (CDW; see also Section 2.13.4. on page 70).

Table 2–67: Selection of used subCLUTX inside CDW for each individual character position

CLUT2	CLUT1	CLUT0	Meaning for ROM Character and 1-bit/2-bit DRCS Characters	Meaning for 4-bit DRCS Characters
0	0	0	subCLUT0 is selected	subCLUT0 is selected
0	0	1	subCLUT1 is selected	subCLUT1 is selected
0	1	0	subCLUT2 is selected	subCLUT2 is selected
0	1	1	subCLUT3 is selected	subCLUT3 is selected
1	0	0	subCLUT4 is selected	subCLUT0 is selected
1	0	1	subCLUT5 is selected	subCLUT1 is selected
1	1	0	subCLUT6 is selected	subCLUT2 is selected
1	1	1	subCLUT7 is selected	subCLUT3 is selected
See also Section 2.13.4. on page 70-Character Display Word (CDW)				

CLUT entries from 0-15 are hardwired and can not be changed by the user.

Each of the 48 RAM programmable CLUT locations have a width of 2 Byte. These 2 Bytes are used to define a 3 × 4-bit RGB value plus the behavior of the BLANK and COR output pins. The following format is used.

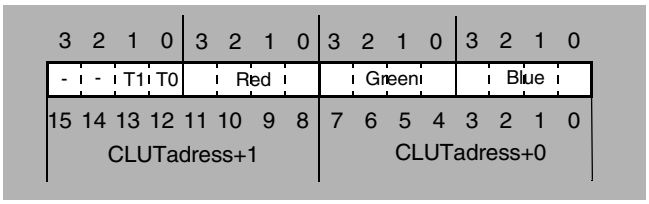


Fig. 2–13: RGB/Transparency Memory Format of CLUT

Fig. 2–13 shows the RBG/Transparency memory format of CLUT:

Bit 3 … 0: 4-bit representation of Blue value

Bit 7 … 4: 4-bit representation of Green value

Bit 11 … 8: 4-bit representation of Red value

Bit 12 Directly fed to BLANK pin

Bit 13 Directly fed to COR pin

Bit 14 Reserved

Bit 15 Reserved

Table 2–68: Organization of CLUT

RAM Address	CLUT Entry	CLUT No for ROM, and 1-bit DRCS Character		CLUT No for Cursor		CLUT No for 2-bit DRCS Character		CLUT No for 4-bit DRCS Character		Hardwired CLUT
		No.	Entry	No.	Entry	No.	Entry	No.	Entry	
	0	0	0	Not available	0	Not available	0	0	0	R G B = 00_d 00_d 00_d
Not available	1		1		1		1		1	R G B = 15_d 00_d 00_d
Not available	2		2		2		2		2	R G B = 00_d 15_d 00_d
Not available	3		3		3		3		3	R G B = 15_d 15_d 00_d
Not available	4		4	Not available	0	Not available	0		4	R G B = 00_d 00_d 15_d
Not available	5		5		1		1		5	R G B = 15_d 00_d 15_d
Not available	6		6		2		2		6	R G B = 00_d 15_d 15_d
Not available	7		7		3		3		7	R G B = 15_d 15_d 15_d
Not available	8	1	0	Not available	0	Not available	0	0	8	R G B = 00_d 00_d 00_d
Not available	9		1		1		1		9	R G B = 07_d 00_d 00_d
Not available	10		2		2		2		10	R G B = 00_d 07_d 00_d
Not available	11		3		3		3		11	R G B = 07_d 07_d 00_d
Not available	12		4	Not available	0	Not available	0		12	R G B = 00_d 00_d 07_d
Not available	13		5		1		1		13	R G B = 07d 00_d 07_d
Not available	14		6		2		2		14	R G B = 00_d 07_d 07_d
Not available	15		7		3		3		15	R G B = 07_d 07_d 07_d

Table 2–68: Organization of CLUT, continued

RAM Address	CLUT Entry	CLUT No for ROM, and 1-bit DRCS Character		CLUT No for Cursor		CLUT No for 2-bit DRCS Character		CLUT No for 4-bit DRCS Character		Hardwired CLUT
		No.	Entry	No.	Entry	No.	Entry	No.	Entry	
$CLUTPOINT_H + 00_H$	16	2	0	0	0	0	0	1	0	Software programmable
$CLUTPOINT_H + 02_H$	17		1		1		1		1	Software programmable
$CLUTPOINT_H + 04_H$	18		2		2		2		2	Software programmable
$CLUTPOINT_H + 06_H$	19		3		3		3		3	Software programmable
$CLUTPOINT_H + 08_H$	20		4	1	0	1	0		4	Software programmable
$CLUTPOINT_H + 0A_H$	21		5		1		1		5	Software programmable
$CLUTPOINT_H + 0C_H$	22		6		2		2		6	Software programmable
$CLUTPOINT_H + 0E_H$	23		7		3		3		7	Software programmable
$CLUTPOINT_H + 10_H$	24	3	0	2	0	2	0		8	Software programmable
$CLUTPOINT_H + 12_H$	25		1		1		1		9	Software programmable
$CLUTPOINT_H + 14_H$	26		2		2		2		10	Software programmable
$CLUTPOINT_H + 16_H$	27		3		3		3		11	Software programmable
$CLUTPOINT_H + 18_H$	28		4	3	0	3	0		12	Software programmable
$CLUTPOINT_H + 1A_H$	29		5		1		1		13	Software programmable
$CLUTPOINT_H + 1C_H$	30		6		2		2		14	Software programmable
$CLUTPOINT_H + 1E_H$	31		7		3		3		15	Software programmable

Table 2–68: Organization of CLUT, continued

RAM Address	CLUT Entry	CLUT No for ROM, and 1-bit DRCS Character		CLUT No for Cursor		CLUT No for 2-bit DRCS Character		CLUT No for 4-bit DRCS Character		Hardwired CLUT
		No.	Entry	No.	Entry	No.	Entry	No.	Entry	
$CLUTPOINT_H + 20_H$	32	4	0	4	0	4	0	2	0	Software programmable
$CLUTPOINT_H + 22_H$	33		1		1		1		1	Software programmable
$CLUTPOINT_H + 24_H$	34		2		2		2		2	Software programmable
$CLUTPOINT_H + 26_H$	35		3		3		3		3	Software programmable
$CLUTPOINT_H + 28_H$	36		4	5	0	5	0		4	Software programmable
$CLUTPOINT_H + 2A_H$	37		5		1		1		5	Software programmable
$CLUTPOINT_H + 2C_H$	38		6		2		2		6	Software programmable
$CLUTPOINT_H + 2E_H$	39		7		3		3		7	Software programmable
$CLUTPOINT_H + 30_H$	40	5	0	6	0	6	0		8	Software programmable
$CLUTPOINT_H + 32_H$	41		1		1		1		9	Software programmable
$CLUTPOINT_H + 34_H$	42		2		2		2		10	Software programmable
$CLUTPOINT_H + 36_H$	43		3		3		3		11	Software programmable
$CLUTPOINT_H + 38_H$	44		4	7	0	7	0		12	Software programmable
$CLUTPOINT_H + 3A_H$	45		5		1		1		13	Software programmable
$CLUTPOINT_H + 3C_H$	46		6		2		2		14	Software programmable
$CLUTPOINT_H + 3E_H$	47		7		3		3		15	Software programmable

Table 2–68: Organization of CLUT, continued

RAM Address	CLUT Entry	CLUT No for ROM, and 1-bit DRCS Character		CLUT No for Cursor		CLUT No for 2-bit DRCS Character		CLUT No for 4-bit DRCS Character		Hardwired CLUT
		No.	Entry	No.	Entry	No.	Entry	No.	Entry	
$CLUTPOINT_H + 40_H$	48	6	0	Not available	0	Not available	0	3	0	Software programmable
$CLUTPOINT_H + 42_H$	49		1		1		1		1	Software programmable
$CLUTPOINT_H + 44_H$	50		2		2		2		2	Software programmable
$CLUTPOINT_H + 46_H$	51		3		3		3		3	Software programmable
$CLUTPOINT_H + 48_H$	52		4	Not available	0	Not available	0		4	Software programmable
$CLUTPOINT_H + 4A_H$	53		5		1		1		5	Software programmable
$CLUTPOINT_H + 4C_H$	54		6		2		2		6	Software programmable
$CLUTPOINT_H + 4E_H$	55		7		3		3		7	Software programmable
$CLUTPOINT_H + 50_H$	56	7	0	Not available	0	Not available	0		8	Software programmable
$CLUTPOINT_H + 52_H$	57		1		1		1		9	Software programmable
$CLUTPOINT_H + 54_H$	58		2		2		2		10	Software programmable
$CLUTPOINT_H + 56_H$	59		3		3		3		11	Software programmable
$CLUTPOINT_H + 58_H$	60		4	Not available	0	Not available	0		12	Software programmable
$CLUTPOINT_H + 5A_H$	61		5		1		1		13	Software programmable
$CLUTPOINT_H + 5C_H$	62		6		2		2		14	Software programmable
$CLUTPOINT_H + 5E_H$	63		7		3		3		15	Software programmable

2.13.7.5.1. CLUT Access for ROM Characters/1-bit DRCS Characters

For each pixel of a character a 1-bit background/foreground information is available. 1 out of 8 sub CLUTs can be selected by character display word (CDW; see also Section 2.13.4.). 1 out of 8 color vectors can be selected as a foreground and background color vector by the character display word (CDW; see also Section 2.13.4. on page 70). Please notice Table 2–68 on page 90.

2.13.7.5.2. CLUT Access for 2-Bit DRCS Characters

2-bit DRCS characters are stored in the RAM. Within a 2-bit DRCS character a 2-bit color vector information is available for each pixel. By this 2-bit information 1 out of 4 color vectors is selected from a subCLUT.

1 out of 8 subCLUTs is selected by character display word (CDW; see also Section 2.13.4. on page 70). Please notice Table 2–68 on page 90.

2.13.7.5.3. CLUT Access for 4-bit DRCS Characters

4-bit DRCS characters are stored in the RAM. Within a 4-bit DRCS character a 4-bit color vector information is available for each pixel. By this 1 out of 16 color vectors is selected from a subCLUT.

One out of 4 subCLUTs are selected by character display word (CDW; see also Section 2.13.4. on page 70).
Please notice Table 2–68 on page 90.

2.13.7.6. Character Resolution

The character matrix of DRCS characters can be adjusted in vertical direction from 9 lines up to 16 lines. In horizontal direction the character matrix is fixed to 12 pixels.

Table 2–69: Character resolution

CHADRC2	CHADRC1	CHADRC0	Description
0	0	0	9 lines
0	0	1	10 lines
0	1	0	11 lines
0	1	1	12 lines
1	0	0	13 lines
1	0	1	14 lines
1	1	0	15 lines
1	1	1	16 lines
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

The character matrix of the ROM characters can also be adjusted in vertical direction from 9 lines up to 16 lines. In horizontal direction the ROM character matrix is fixed to 12 pixels.

Table 2–70: Character matrix settings

CHAROM 2	CHAROM 1	CHAROM 0	Description
0	0	0	9 lines
0	0	1	10 lines
0	1	0	11 lines
0	1	1	12 lines
1	0	0	13 lines
1	0	1	14 lines
1	1	0	15 lines
1	1	1	16 lines
See also Section 2.13.5. on page 75-Global Display Word (GDW)			

The parameter CHAROM is used to characterize the organization of ROM characters. The parameter CHADRC is used to characterize the organization of DRCS characters and the vertical count of lines for a character row on output side. If the count of lines of ROM characters is smaller than the count of DRCS characters the lines of ROM characters are filled up with background colored pixels.

2.13.7.7. Shadowing

If shadowing is enabled the ROM characters and 1-bit DRCS characters of the characters are displayed by west shadow or east shadow. The color vector of the shadow is defined by software. The shadow color vector has a width of 6 bit.

The shadow feature is enabled by the bit SHEN.

Table 2–71: Shadow settings

SHEN	Description
0	Shadow disabled.
1	Shadow for ROM characters and 1-bit DRCS characters.
See also Section 2.13.5. on page 75-Global Display Word (GDW)	

There are two options for shadowing, as shown in Fig. 2–72.

Table 2–72: Shadow options

SHEAWE	Description
0	East shadowing.
1	West shadowing.
See also Section 2.13.5. on page 75 / Global Display Word (GDW)	

Example for a “A” displayed in shadow mode:

Within one character matrix shadowing is only processed for the pixels which are belonging to that character matrix. Pixels of one character matrix can not generate a shadow inside a neighbored character matrix.

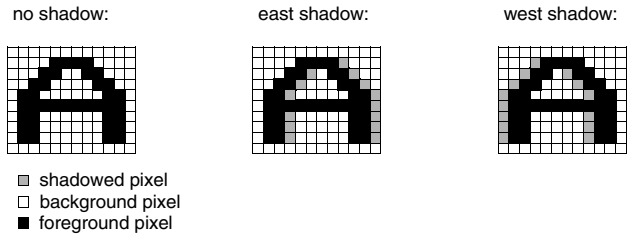


Fig. 2–14: Processing of Shadowing

CLUT entries from 0-63 can be used as a shadow color vector as shown in Table 2–73.

Table 2–73: Shadow color vector settings

SHCOL5	SHCOL4	SHCOL3	SHCOL2	SHCOL1	SHCOL0	Description
0	0	0	0	0	0	Defines a color vector for shadowing See also Section 2.13.7.5. on page 88
0	0	0	0	0	1	
…						
1	1	1	1	1	0	
1	1	1	1	1	1	
See also Section 2.13.5. on page 75-Global Display Word (GDW)						

2.13.7.8. Progressive Scan

This feature is useful for TV-devices in which a frame consists of 1 field with 625 lines instead of 2 fields with 312.5 lines each.

For this TV-fields on RGB-output lines are be repeated twice by enabling the progressive scan feature. This repetition of lines in vertical direction is only processed for lines inside the character display area.

Table 2–74: Progressive scan settings

Progress	Description
0	Progressive scan support is disabled.
1	Progressive scan support is enabled.
See also Section 2.13.5. on page 75-Global Display Word (GDW)	

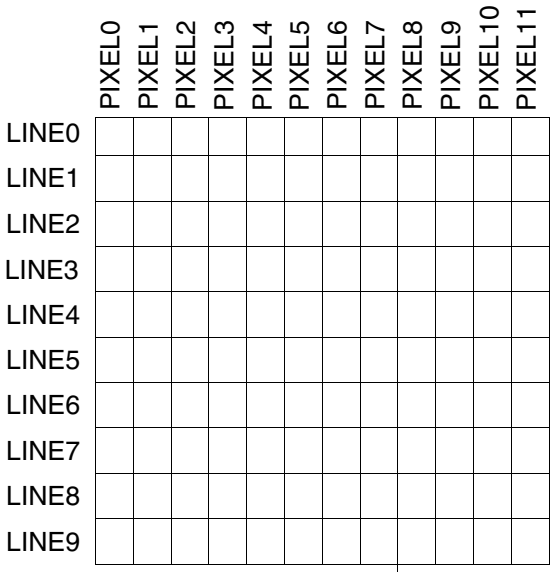


Fig. 2–15: Allocation of Pixels Inside the Character Matrix

2.13.8. DRCS Characters

DRCS characters are available in the XRAM. There are three different DRCS color resolution formats available:

– 1-bit per pixel DRCS characters

– 2-bit per pixel DRCS characters

– 4-bit per pixel DRCS characters

In which way this 1-bit, 2-bit or 4-bit color vector information is used to access the CLUT, see Section 2.13.7.5. on page 88.

2.13.8.1. Memory Organization of DRCS Characters

The following examples are proceeded on the assumption that a height of 11 character lines is selected. The memory organization behaves the same for any other count of lines.

Each character starts at a new Byte address. This causes, that for odd heights nibbles may be left free.

Table 2–75: 1-Bit DRCS characters

Char	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Character 1	$DRC1POINT_H + 00_H$	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 1	PIXEL 2	PIXEL 3	PIXEL 4	PIXEL 5	PIXEL 6	PIXEL 7
		BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0
	$DRC1POINT_H + 01_H$	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 1	LINE 1	LINE 1	LINE 1
		PIXEL 8	PIXEL 9	PIXEL 10	PIXEL 11	PIXEL 0	PIXEL 1	PIXEL 2	PIXEL 3
		BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0
	$DRC1POINT_H + 02_H$	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 1	LINE 1	LINE 1	LINE 1	LINE 1	LINE 1	LINE 1	LINE 1
		PIXEL 4	PIXEL 5	PIXEL 6	PIXEL 7	PIXEL 8	PIXEL 9	PIXEL 10	PIXEL 11
		BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0
	…	…							
	$DRC1POINT_H + 10_H$	CHAR 1	CHAR 1	CHAR 1	CHAR 1	Left free			
		LINE 10	LINE 10	LINE 10	LINE 10				
		PIXEL 8	PIXEL 9	PIXEL 10	PIXEL 11				
		BIT 0	BIT 0	BIT 0	BIT 0				
Character 2	$DRC1POINT_H + 11_H$	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 1	PIXEL 2	PIXEL 3	PIXEL 4	PIXEL 5	PIXEL 6	PIXEL 7
		BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0	BIT 0
…	…	…							

Table 2–76: 2-Bit DRCS characters

Char	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Character 1	$DRC2POINT_H$ + 00_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 0	PIXEL 1	PIXEL 1	PIXEL 2	PIXEL 2	PIXEL 3	PIXEL 3
		BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1
	$DRC2POINT_H$ + 01_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 4	PIXEL 4	PIXEL 5	PIXEL 5	PIXEL 6	PIXEL 6	PIXEL 7	PIXEL 7
		BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1
	$DRC2POINT_H$ + 02_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 8	PIXEL 8	PIXEL 9	PIXEL 9	PIXEL 10	PIXEL 10	PIXEL 11	PIXEL 11
		BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1
	…	…							
	$DRC2POINT_H$ + 20_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10
		PIXEL 8	PIXEL 8	PIXEL 9	PIXEL 9	PIXEL 10	PIXEL 10	PIXEL 11	PIXEL 11
		BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1
Character 2	$DRC2POINT_H$ + 21_H	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 0	PIXEL 1	PIXEL 1	PIXEL 2	PIXEL 2	PIXEL 3	PIXEL 3
		BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1	BIT 0	BIT 1
…	…	…							

Table 2–77: 4-Bit DRCS characters

Char	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Character 1	$DRC4POINT_H$ + 00_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 0	PIXEL 0	PIXEL 0	PIXEL 1	PIXEL 1	PIXEL 1	PIXEL 1
		BIT 0	BIT 1	BIT 2	BIT 3	BIT 0	BIT 1	BIT 2	BIT 3
	$DRC4POINT_H$ + 01_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 2	PIXEL 2	PIXEL 2	PIXEL 2	PIXEL 3	PIXEL 3	PIXEL 3	PIXEL 3
		BIT 0	BIT 1	BIT 2	BIT 3	BIT 0	BIT 1	BIT 2	BIT 3
	$DRC4POINT_H$ + 02_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 4	PIXEL 4	PIXEL 4	PIXEL 4	PIXEL 5	PIXEL 5	PIXEL 5	PIXEL 5
		BIT 0	BIT 1	BIT 2	BIT 3	BIT 0	BIT 1	BIT 2	BIT 3
	…	…							
	$DRC4POINT_H$ + 41_H	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1	CHAR 1
		LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10	LINE 10
		PIXEL 10	PIXEL 10	PIXEL 10	PIXEL 10	PIXEL 11	PIXEL 11	PIXEL 11	PIXEL 11
		BIT 0	BIT 1	BIT 2	BIT 3	BIT 0	BIT 1	BIT 2	BIT 3
Character 2	$DRC4POINT_H$ + 42_H	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2	CHAR 2
		LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0	LINE 0
		PIXEL 0	PIXEL 0	PIXEL 0	PIXEL 0	PIXEL 1	PIXEL 1	PIXEL 1	PIXEL 1
		BIT 0	BIT 1	BIT 2	BIT 3	BIT 0	BIT 1	BIT 2	BIT 3
…	…	…							

2.13.9. Memory Organization

The memory organization concept of the OSD is based on a flexible pointer concept. All display memory registers reside in the internal XRAM only.

There are 4 Bytes of SFR registers which are pointing to two pointer arrays inside the XRAM as shown in Table 2–78

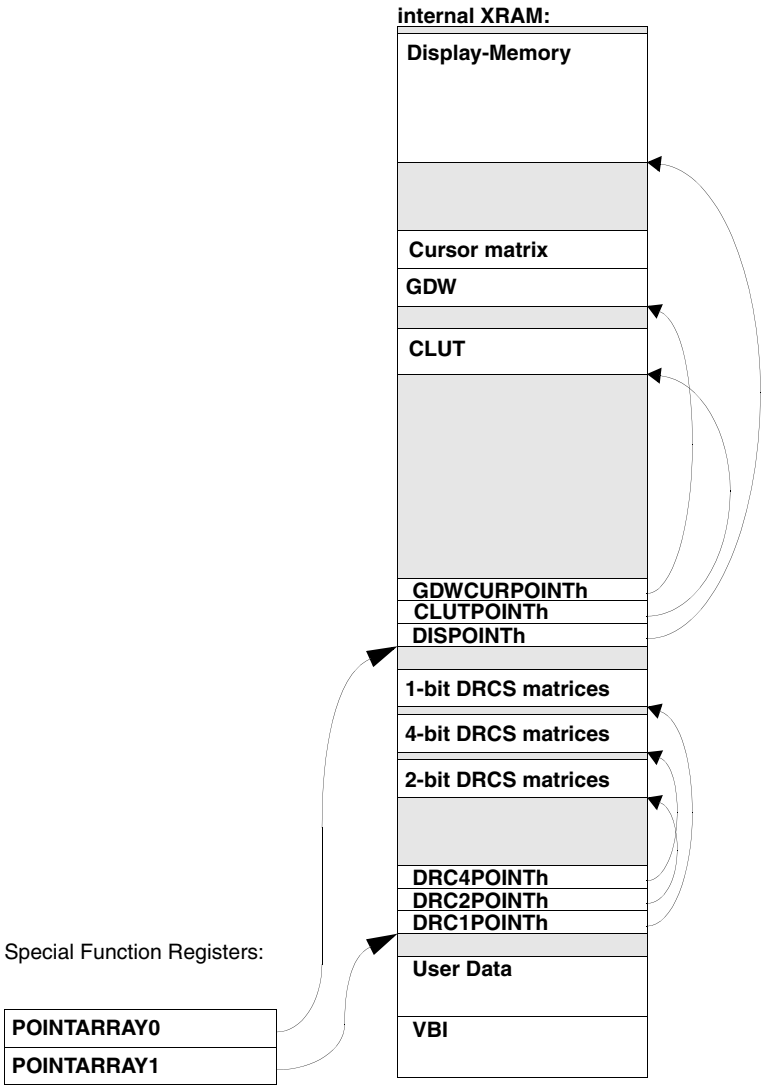


Fig. 2–16: Memory Organization of On Screen Display

Table 2–78: Pointers to start address of on-screen display registers inside XRAM

SFR Address	Name	Function
XX_H	POINTARRAY0	Pointer to pointer array 0
$XX_H + 02_H$	POINTARRAY1	Pointer to pointer array 1

These 2 SFR pointers are used to point to 2 × 3 other pointers. These 6 pointers are pointing to the start address of the following memory areas:

– Start address of character display area memory
– Start address of CLUT
– Start address of 1-bit DRCS characters matrixes
– Start address of 2-bit DRCS characters matrixes
– Start address of 4-bit DRCS characters matrixes
– Start address of global display word / cursor matrix

User has to take care for a pointer definition so that memory areas do not overlap each other on the one hand and that the definition is optimized in a way, so that no memory is wasted on the other hand. The length of the global display word is fixed to 10 Byte and the length of the CLUT is fixed to 2 × 48 Byte. The length of all the other areas depend on the OSD requirements (see also Section 2.13.9.1. to Section 2.13.9.4. on page 102).

Each of the six pointers to the memory areas is stored in an array of pointers. Each pointer in this array has got a width of 16 bits and uses 2 Bytes inside the RAM.

Table 2–79: Memory pointers

Pointer Array	Start address in Array	Name	Function
POINTFIELD0	0_H (LByte) 1_H (HByte)	DISPOINTh	Pointer to display memory
	2_H (LByte) 3_H (HByte)	CLUTPOINT	Pointer to CLUT
	4_H (LByte) 5_H (HByte)	GDWCURPOINTh	Pointer to GDW and cursor matrix
POINTFIELD1	0_H (LByte) 1_H (HByte)	DRC1POINTh	Pointer 1-bit DRCS matrices
	2_H (LByte) 3_H (HByte)	DRC2POINTh	Pointer 2-bit DRCS matrices
	4_H (LByte) 5_H (HByte)	DRC4POINTh	Pointer 4-bit DRCS matrices

2.13.9.1. Character Display Area

The character display area consists of 3 Bytes for each character position of the character display area. These three Bytes are used to store the character display word as it is described in Section 2.13.4. on page 70.

The array is sorted in a incremental serial order coming from the top left character throughout the bottom right character of the character display area. For further information see Section 2.13.2. on page 69.

The length of this display memory area depends on the parameter settings of DISALH0 … DISALH4.

2.13.9.2. CLUT Area

The CLUT area consist of 48 × 2 Byte CLUT contents. The CLUT contents are stored in a serial incremental order.

For further information see Section 2.13.7.5. on page 88.

The length of the CLUT is fixed to 96 Bytes.

2.13.9.3. Global Display Word/Cursor

The area of the global display word is fixed to 10 Byte. All the global display relevant informations are stored inside global display word (GDW; see also Section 2.13.5. on page 75). The cursor matrix for cursor display is stored after the global display word.See also Section 2.13.2. on page 69.

The length of the memory area of global display word is fixed to 10 Byte. The length of the memory area of cursor matrix depends on the settings of CHADRC2 … CHADRC0.

2.13.9.4. 1-bit/2-bit/4-bit DRCS Character

In this area the pixel information of the dynamically reconfigurable characters is stored. For further information on the memory format refer to Section 2.13.8. on page 96.

The length of these areas depends on the settings of DRCSB1_3 … DRCSB1_0 and the settings of DRCSB2_3 … DRCSB2_0.

2.13.9.5. Overview on the SFR Registers

Other than the settings in the XRAM, SFR registers are used for OSD control.

Table 2–80: SFR registers used for OSD control

SFR Address	Name	Bit Programmable	Width	Purpose
$F8_H$	EN_Ld_Cur	Yes	1 bit	Used to avoid the download of the parameter settings of the GDW from the RAM to the local display generator register bank. See also Section : 0: Download disabled. 1: Download enabled. Initial value: 0
$F8_H$	EN_DGOut	Yes	1 bit	Used to disable/enable the output of the display generator. If display generator is disabled the RGB outputs of the IC are set to black and the outputs BLANK and COR are set to. COR = ENABLECOR BLANK = ENABLEBLA If display generator is enabled the display information RGB, COR and BLANK is generated according to the parameter settings in the XRAM. 0: Display generator is disabled. 1: Display generator is enabled. Initial value: 0
$F8_H$	Dis_Cor	No	1 bit	Defines the level of the COR output if display generator is disabled. Initial value: 0
$F8_H$	Dis_Blank	No	1 bit	Defines the level of the BLANK output if display generator is disabled. Initial value: 1
$F3_H$	POINTARRAY 1_1	No	6 bit	Defines a pointer to a pointer array. See also Section 2.13.9. on page 100 Initial value: 0
$F4_H$	POINTARRAY 1_0	No	8 bit	Defines a pointer to a pointer array. See also Section 2.13.9. on page 100 Initial value: 0
$F5_H$	POINTARRAY 0_1	No	6 bit	Defines a pointer to a pointer array. See also Section 2.13.9. on page 100 Initial value: 0
$F6_H$	POINTARRAY 0_0	No	8 bit	Defines a pointer to a pointer array. See also Section 2.13.9. on page 100 Initial value: 0

2.13.10. TVText Pro Characters

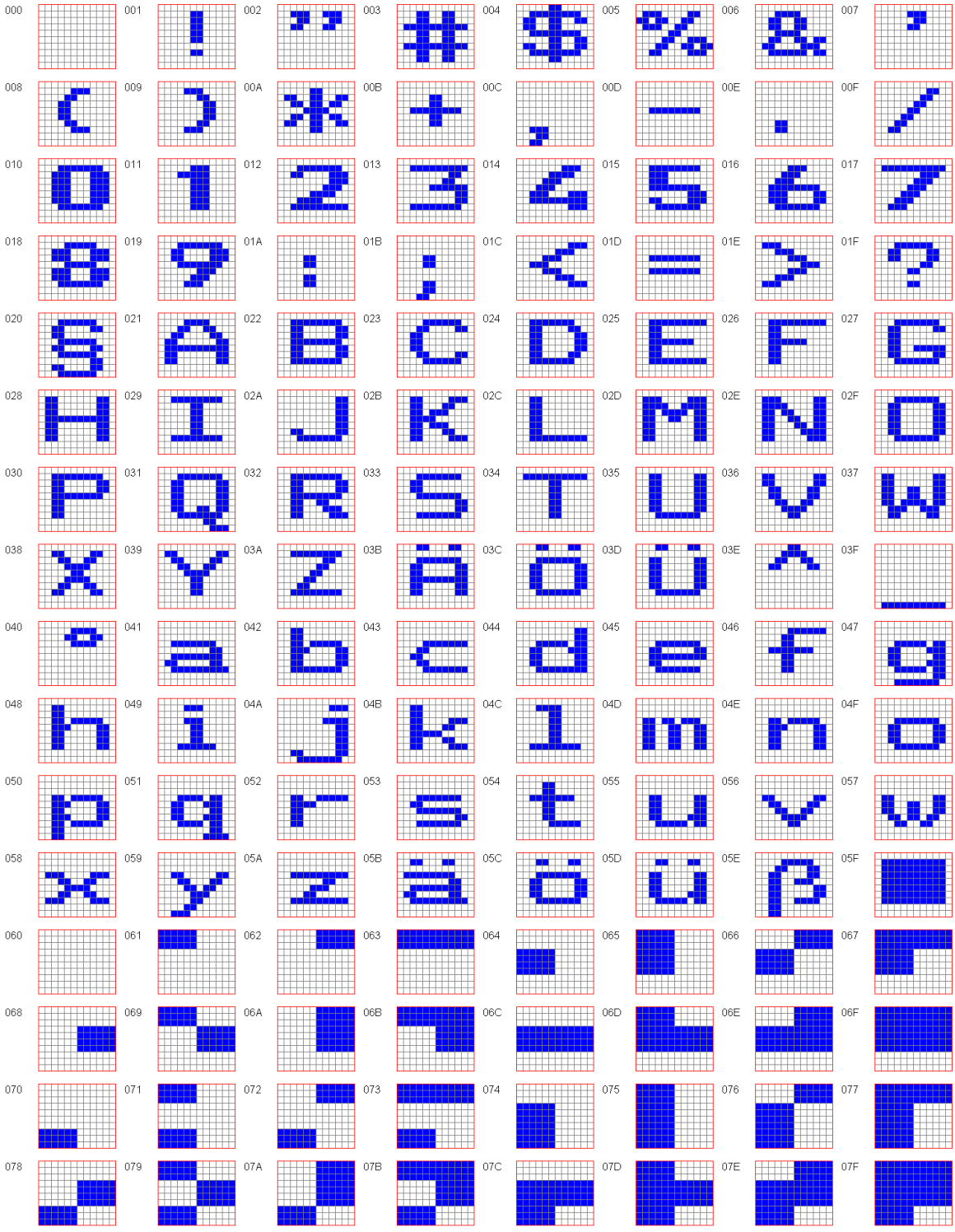


Fig. 2–17: ROM Character Matrices

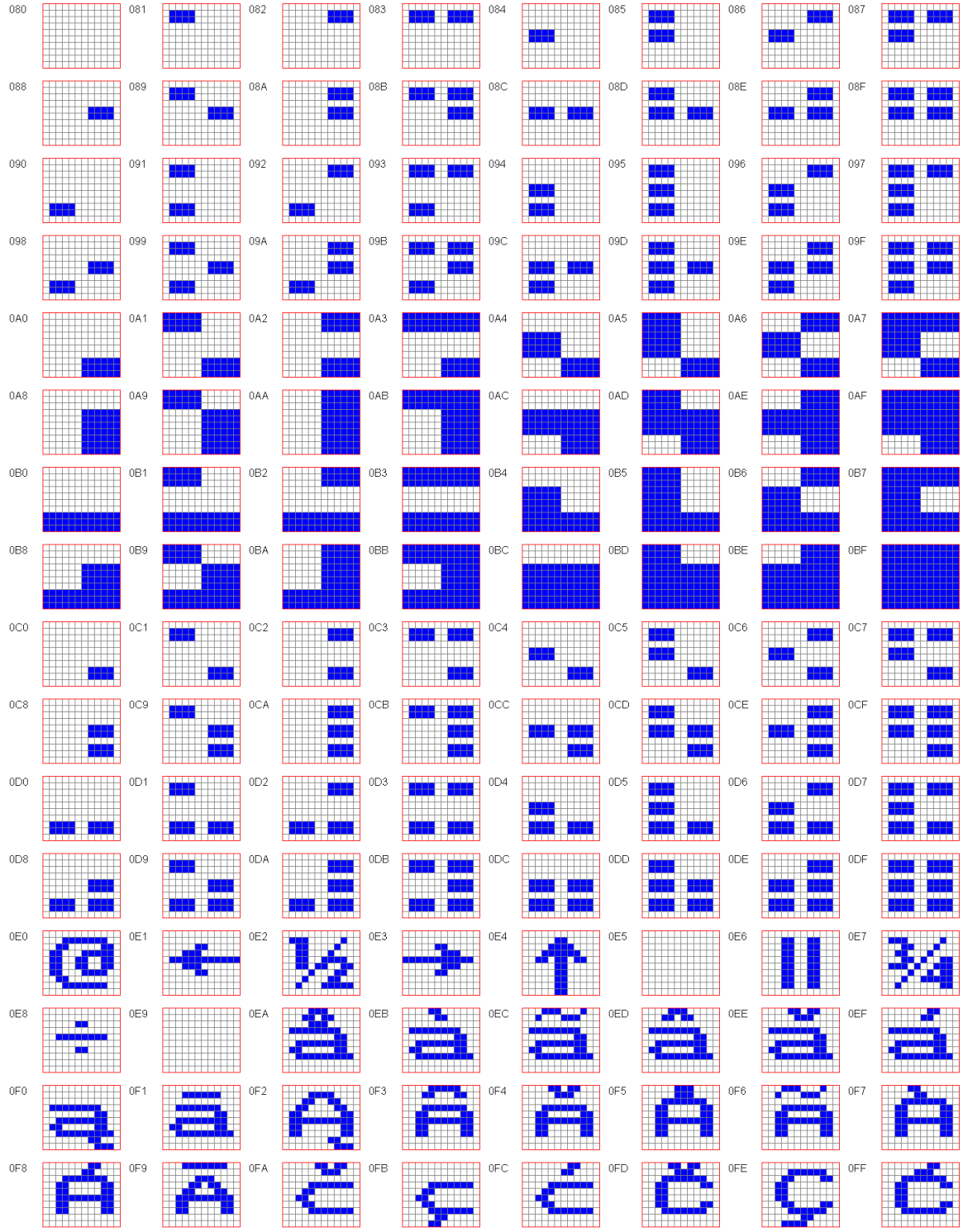


Fig. 2–18: ROM Character Matrices

Fig. 2–19: ROM Character Matrices

Fig. 2–20: ROM Character Matrices

Fig. 2–21: ROM Character Matrices

2.14. D/A Converter

TVTpro uses a 3 × 2-bit voltage D/A converter to generate analog RGB output signals with a nominal amplitude of 0.7 V (also available: 0.5 V, 1.0 V and 1.2 V) peak-to-peak.

2.14.1. Related Registers

Table 2–81: Related registers

Register Name	Bit Name							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCR1	Reserved	**RGB_G[1:0]**		CORBL	VSU[3:0]			
PSAVE bit addressable				CADC	WAKUP	SLI_ACQ	**DISP**	PERI
PCON	SMOD	**PDS**	**IDLS**	**SD**	GF[1:0]		PDE	IDLE
See Section 3. on page 110 for detailed register description.								

3. Special Function Register (SFR)

3.1. SFR Register Block Index

Table 3–1: SFR block index

Name	Page
ADC	128
CRT	126
DISPLAY	135
DSYNC	130
INTERRUPT	121
MICRO	117
PORT	117
PWM	127
SFRIF	129
UART	120
WATCHDOG	125

3.2. SFR Register Index

Table 3–2: SFR register bits index

Name	Page
A[7:0]	120
$\overline{A17}$_P4_0	136
$\overline{A18}$_P4_1	136
$\overline{A19}$_P4_4	136
AC	120
ACQ_STA	130
ACQON	130
AD[3:0]	128
ADC	123
ADW	124
ADWULE	128
AHS	124
AVS	124

Table 3–2: SFR register bits index, continued

Name	Page
B[7:0]	120
BHCR[7:0]	133
BVCR[7:0]	133
BVCR[9:8]	133
C_NT0	119
C_NT1	118
CADC	129
CADC0[7:0]	128
CADC1[7:0]	128
CADC2[7:0]	128
CADC3[7:0]	128
CapH[7:0]	126
CapL[7:0]	126
CB[19:16]	119
CC	124
Clk_src	129
COR_BL	131
CY	120
D[7:0]	120
DHS	124
Dis_Blank	135
Dis_Cor	135
DISP	129
DPH[7:0]	117
DPL[7:0]	117
DPSEL[2:0]	117
DVS	124
E24	121
EAD	121
EADW	121
EAH	121
EAL	121
EAV	121

Table 3–2: SFR register bits index, continued

Name	Page
ECC	121
EDH	121
EDV	121
EHCR[7:0]	132
En_DGOut	135
En_Ld_Cur	135
ENARW	136
ENERCLK	135
ENETCLK	135
EPW	121
ET0	121
ET1	121
EU	121
EVCR[7:0]	133
EVCR[9:8]	133
EWT	121
EX0	121
EX0F	122
EX0R	122
EX1	121
EX12	121
EX13	121
EX18	121
EX19	121
EX1F	122
EX1R	122
EX20	121
EX21	121
EX6	121
EXX0	121
EXX0F	122
EXX0R	122
EXX1	121

Table 3–2: SFR register bits index, continued

Name	Page
EXX1F	122
EXX1R	122
F0	120
F1	120
FALL	126
First	126
FREQSEL(1)	135
FREQSEL(2)	135
G0P0	122
G0P1	123
G1P0	122
G1P1	123
G2P0	122
G2P1	123
G3P0	122
G3P1	123
G4P0	122
G4P1	123
G5P0	122
G5P1	123
GATE0	118
GATE1	118
GF0	118
GF1	118
HP	131
HPR[11:8]	134
HPR[7:0]	134
HYS	124
IB[19:16]	119
IDLE	118
IDLS	118
IE0	118
IE1	118

Table 3–2: SFR register bits index, continued

Name	Page
IEX0	124
IEX1	124
INT	131
IntSrc1	136
IntSrc1	136
IT0	118
IT1	118
L24	123
M0[1:0]	119
M1[1:0]	118
MAST	132
MB[18:16]	119
MB[19]	119
MEXSP[6:0]	119
MinH[7:0]	126
MinL[7:0]	126
MM	119
MSIZ[7:0]	120
MX[19]	119
MX[19]	119
MXM	119
NB[19:16]	119
O_E_$\overline{P3_0}$	136
O_E_Pol	136
Odd_Ev	133
OSCPD	135
OV	126
OV	120
OV	127
P	120
P0[7:0]	117
P1[7:0]	117
P2[7:0]	117

Table 3–2: SFR register bits index, continued

Name	Page
P3[7:0]	117
P4[7:0]	117
$\overline{P4_7}$_Alt	135
PC140[7:0]	127
PC141[7:0]	127
PC80[7:0]	127
PC81[7:0]	127
PC82[7:0]	127
PC83[7:0]	127
PC84[7:0]	127
PC85[7:0]	127
PCX140[7:2]	127
PCX141[7:2]	127
PDE	118
PDS	117
PE[7:0]	128
PERI	129
PF[10:8]	130
PF[7:0]	130
PLG	126
PLL_Res	129
PLLS	129
Point0[13:8]	135
Point0[7:0]	135
Point1[13:8]	135
Point1[7:0]	135
PR	126
PR1	126
PWC[13:8]	127
PWC[7:0]	127
PWM_Tmr	127
PWtmr	124
RB8	120

Table 3–2: SFR register bits index, continued

Name	Page
REL	126
RelH[7:0]	126
RelL[7:0]	126
REN	120
Reserved	130
Reserved	130
RGB_D[1:0]	131
RGB_G[1:0]	130
RI	120
RISE	126
RS[1:0]	120
RUN	126
SD	118
SDH[11:8]	132
SDH[7:0]	132
SDV[7:0]	132
SDV[9:8]	132
SEL	126
SLI_ACQ	129
SM0	120
SM1	120
SM2	120
SMOD	117
SNC	132
SND_H[2:0]	125
SND_V[5:3]	125
SP_[7:0]	117
Start	126
TAP	135
TAP	135
TB8	120
TF0	118
TF1	118

Table 3–2: SFR register bits index, continued

Name	Page
TH0[7:0]	119
TH1[7:0]	119
TI	120
TL0[7:0]	119
TL1[7:0]	119
TR0	118
TR1	118
UB3	119
UB4	119
VBIADR[3:0]	130
VCS	132
VL[7:0]	134
VL[9:8]	134
VP	131
VS_$\overline{OE}$	136
VSU[3:0]	131
VSU2[3:0]	134
WAKUP	129
WDT_in	125
WDT_narst	125
WDT_ref	125
WDT_rst	125
WDT_start	125
WDT_tmr	125
WDThi[7:0]	126
WDTlow[7:0]	125
WDTrel[7:0]	125
WTmr	123
WTmr_ov	125
WTmr_strt	125

3.3. SFR Register Address Index

Table 3–3: SFR subaddress index

Sub	Data Bits								Reset
	7	6	5	4	3	2	1	0	
80 [1]	P0[7:0]								hFF
81	SP_[7:0]								h07
82	DPL[7:0]								h00
83	DPH[7:0]								h00
84	[2]					DPSEL[2:0]			h00
87	SMOD	PDS	IDLS	SD	GF1	GF0	PDE	IDLE	h00
88	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	h00
89	GATE1	C_NT1	M1[1:0]		GATE0	C_NT0	M0[1:0]		h00
8A	TL0[7:0]								h00
8B	TL1[7:0]								h00
8C	TH0[7:0]								h00
8D	TH1[7:0]								h00
90	P1[7:0]								hFF
94	CB[19:16]				NB[19:16]				h00
95	MM	MB[18:16]			IB[19:16]				h00
96	MB[19]	UB3	UB4	MX[19]	MXM	MX[19]			h00
97		MEXSP[6:0]							h00
98	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	h00
99	D[7:0]								h00
A0	P2[7:0]								hFF
A8	EAL		EAD	EU	ET1	EX1	ET0	EX0	h00
A9			EDV	EAV	EXX1	EWT	EXX0		h00
AA			EDH	EAH	ECC	EPW	EX13 [3]	EX12 [3]	h00
AB			EADW	E24	EX21 [3]	EX20 [3]	EX19 [3]	EX18 [3]	h00
AC			G5P0	G4P0	G3P0	G2P0	G1P0	G0P0	h00
AD	EXX1R	EXX1F	EXX0R	EXX0F	EX1R	EX1F	EX0R	EX0F	h05
B0	P3[7:0]								hFF
B1	WDTrel[7:0]								h00
B2	WDT_in	WDT_start	WDT_narst	WDT_rst					h00
B3	WDT_ref	WDT_tmr	WTmr_strt	WTmr_ov					h00
B4	WDTlow[7:0]								h00
B5	WDThi[7:0]								h00
B7	RelL[7:0]								h00
B8			G5P1	G4P1	G3P1	G2P1	G1P1	G0P1	h00
B9	RelH[7:0]								h00
BA	CapL[7:0]								h00
BB	CapH[7:0]								h00
BC	MinL[7:0]								h00
BD	MinH[7:0]								h00
BE	OV	PR	PLG	REL	RUN	RISE	FALL	SEL	h00

Table 3–3: SFR subaddress index, continued

Sub	Data Bits								Reset
	7	6	5	4	3	2	1	0	
BF						PR1	First	Start	h00
C0	L24	ADC	WTmr	AVS	DVS	PWtmr	AHS	DHS	h00
C1	PC80[7:0]								h00
C2	PC81[7:0]								h00
C3	PC82[7:0]								h00
C4	PC83[7:0]								h00
C5	PC84[7:0]								h00
C6	PC85[7:0]								h00
C7	PC140[7:0]								h00
C8	CC	ADW					IEX1	IEX0	h00
C9	PC141[7:0]								h00
CA	PCX140[7:2]								h00
CB	PCX141[7:2]								h00
CC	PWC[7:0]								
CD	PWM_Tmr	OV	PWC[13:8]						
CE	PE[7:0]								h00
D0	CY	AC	F0	RS[1:0]		OV	F1	P	h00
D1	CADC0[7:0]								h00
D2	CADC1[7:0]								h00
D3	CADC2[7:0]								h00
D4	CADC3[7:0]								h00
D5				ADWULE	AD[3:0]				h00
D7						Clk_src	PLL_Res	PLLS	h00
D8				CADC	WAKUP	SLI_ACQ	DISP	PERI	h00
D9	ACQON	Reserved	ACQ_STA		VBIADR[3:0]				h00
DA						PF[10:8]			h01
DB	PF[7:0]								h48
DD			ENETCLK [3]	ENERCLK [3]	P4_7_Alt	VS_OE	O_E_P3_0	O_E_Pol	h00
DE	IntSrc1	IntSrc0			ENARW	A19_P4_4	A18_P4_1	A17_P4_0	h00
DF		HYS	SND_V[2:0]			SND_H[2:0]			h00
E0	A[7:0]								h00
E1	Reserved	RGB_G[1:0]		COR_BL	VSU[3:0]				h00
E2	RGB_D[1:0]		HP	VP	INT	SNC	VCS	MAST	h00
E3							SDV[9:8]		h00
E4	SDV[7:0]								h20
E5					SDH[11:8]				h00
E6	SDH[7:0]								h48
E7	EHCR[7:0]								h0A
E8	P4[7:0]								h00
E9	BHCR[7:0]								h00
EA							BVCR[9:8]		h00
EB	BVCR[7:0]								h00

Table 3–3: SFR subaddress index, continued

Sub	Data Bits								Reset
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
EC							EVCR[9:8]		h00
ED	EVCR[7:0]								h04
EE		Odd_Ev	VSU2[3:0]				VL[9:8]		h02
EF	VL[7:0]								h71
F0	B[7:0]								h00
F1					HPR[11:8]				h08
F2	HPR[7:0]								h55
F3			Point1[13:8]						h00
F4	Point1[7:0]								h06
F5			Point0[13:8]						h00
F6	Point0[7:0]								h00
F8					En_Ld_Cur	En_DGOut	Dis_Cor	Dis_Blank	h00
F9	[4]								-
FD [3]	FREQSEL(1)	FREQSEL(2)	OSCPD						h80
FF	MSIZ[7:0] [3]								-
FF	MSIZ[7:0]								h0F

[1] Addresses in bold are controller fix addresses.
[2] All the empty bits in “grey” are reserved
[3] Reserved.
[4] These registers are for internal use of the device. Do not write in these locations.

As a general rule: Software should only write to the bits which it wants to change. All other bits implemented or not should be masked in order to avoid problems with future versions.

3.4. SFR Register Description

Note: For compatibility reasons every undefined bit in a writeable register should be set to '0'. Undefined bits in a readable register should be treated as "don't care"!

Table 3–4: SFR register description

Name	Sub	Dir	Reset	Range	Function
PORT					
P0	**h80**	**RW**	**hFF**		**Port 0**
P0[7:0]	h80[7:0]	RW	255	0..255	Port 0
P1	**h90**	**RW**	**hFF**		**Port 1**
P1[7:0]	h90[7:0]	RW	255	0..255	Port 1
P2	**hA0**	**RW**	**hFF**		**Port 2**
P2[7:0]	hA0[7:0]	RW	255	0..255	Port 2
P3	**hB0**	**RW**	**hFF**		**Port 3**
P3[7:0]	hB0[7:0]	RW	255	0..255	Port 3
P4	**hE8**	**RW**			**Port 4**
P4[7:0]	hE8[7:0]	RW	255	0..255	Port 4
MICRO					
SP	**h81**	**RW**	**h07**		**Stack Pointer**
SP_[7:0]	h81[7:0]	RW	7	0..255	Stack Pointer
DPL	**h82**	**RW**	**h00**		**Data Pointer Low**
DPL[7:0]	h82[7:0]	RW	0	0..255	Data Pointer low byte
DPH	**h83**	**RW**	**h00**		**Data Pointer High**
DPH[7:0]	h83[7:0]	RW	0	0..255	Data Pointer high byte
DPSEL	**h84**	**RW**	**h00**		**Data Pointer Select**
DPSEL[2:0]	h84[2:0]	RW	0	0..7	Data Pointer Select selects one of eight data pointer
PCON	**h87**	**RW**	**h00**		**Power Control**
SMOD	h87[7]	RW	0	0..1	UART Baud Rate 0: Normal baud rate. 1: Double baud rate.
PDS	h87[6]	RW	0	0..1	Power Down Start Bit 0: Power Down Mode not started. 1: Power Down Mode started. The instruction that sets this bit is the last instruction before entering power down mode. Additionally, this bit is protected by a delay cycle. Power down mode is entered, if and only if bit PDE was set by the previous instruction. Once set, this bit is cleared by hardware and always reads out a 0. DAC, PLL and Oscillator are switched off during Power Down. The CADC is completely switched off (no wake up possible).

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
IDLS	h87[5]	RW	0	0..1	Idle Start Bit 0: Idle Mode not started. 1: Idle Mode started. The instruction that sets this bit is the last instruction before entering idle mode. Additionally, this bit is protected by a delay cycle. Idle mode is entered, if and only if bit IDLE was set by the previous instruction. Once set, this bit is cleared by hardware and always reads out a 0.
SD	h87[4]	RW	0	0..1	Slow-Down Bit 0: Slow-down mode is disabled. 1: Slow-down mode is enabled. This bit is set to indicate the external clock generating circuitry to slow down the frequency. This bit is not protected by a delay cycle.
GF1	h87[3]	RW	0	0..1	Power Control
GF0	h87[2]	RW	0	0..1	General purpose flag bits For user.
PDE	h87[1]	RW	0	0..1	Power-Down Mode Enable Bit When set, a delay cycle is started. The following instruction can then set the device into power down mode. Once set, this bit is cleared by hardware and always reads out a 0.
IDLE	h87[0]	RW	0	0..1	Idle Start Bit 0: Idle Mode not started. 1: Idle Mode started. The instruction that sets this bit is the last instruction before entering idle mode. Additionally, this bit is protected by a delay cycle. Idle mode is entered, if and only if bit IDLE was set by the previous instruction. Once set, this bit is cleared by hardware and always reads out a 0. The CADC is switched off but the CADC-Wake-Up-Unit is active.
TCON	**h88**	**RW**	**h00**		**Timer/Counter Control**
TF1	h88[7]	RW	0	0..1	Timer 1 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	h88[6]	RW	0	0..1	Timer 1 run control bit. Set/cleared by software to turn timer/counter on/off.
TF0	h88[5]	RW	0	0..1	Timer 0 overflow flag. Set by hardware on timer/counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	h88[4]	RW	0	0..1	Timer 0 run control bit. Set/cleared by software to turn timer/counter on/off.
IE1	h88[3]	RW	0	0..1	Interrupt 1 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	h88[2]	RW	0	0..1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts. IT1 = 1 selects transition-activated external interrupts.
IE0	h88[1]	RW	0	0..1	Interrupt 0 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	h88[0]	RW	0	0..1	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts. IT0 = 1 selects transition-activated external interrupts.
TMOD	**h89**	**RW**	**h00**		**Timer/Counter Mode Control**
GATE1	h89[7]	RW	0	0..1	Timer/Ctr Mode
C_NT1	h89[6]	RW	0	0..1	Timer/Ctr Mode
M1[1:0]	h89[5:4]	RW	0	0..3	Timer/Ctr Mode
GATE0	h89[3]	RW	0	0..1	Gating control when set. Timer/counter ëxí is enabled only while ëINTxí pin is high and ëTRxí control pin is set. When cleared, timer ëxí is enabled, whenever ëTRxí control bit is set.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
C_NT0	h89[2]	RW	0	0..1	Timer or counter selector. Cleared for timer operation (input from internal system clock). Set for Counter operation (input from ëTxí input pin).
M0[1:0]	h89[1:0]	RW	0	0..3	Timer Operating Mode 00: 8048 timer: ëTLxí serves as five-bit prescaler. 01: 16-bit timer/counter: ëTHxí and ëTLxí are cascaded, there is no prescaler. 10: 8-bit auto-reload timer/counter: ëTHxí holds a value which is to be reloaded into ëTLxí each time it overflows. 11: (Timer 0) TL0 is an eight-bit timer/counter controlled by the standard timer 0 control bits; TH0 is an eight-bit timer only controlled by timer 1 control bits. (Timer 1) Timer/counter 1 is stopped.
TL0	**h8A**	**RW**	**h00**		**Timer/Counter 0 Low Byte**
TL0[7:0]	h8A[7:0]	RW	0	0..255	Timer/Ctr 0 Low byte
TL1	**h8B**	**RW**	**h00**		**Timer/Counter 0 Low Byte**
TL1[7:0]	h8B[7:0]	RW	0	0..255	Timer/Ctr 1 Low byte
TH0	**h8C**	**RW**	**h00**		**Timer/Counter 0 High Byte**
TH0[7:0]	h8C[7:0]	RW	0	0..255	Timer/Ctr 0 High byte
TH1	**h8D**	**RW**	**h00**		**Timer/Counter 1 High Byte**
TH1[7:0]	h8D[7:0]	RW	0	0..255	Timer/Ctr 1 High byte
MEX1	**h94**	**RW**	**h00**		**Memory Extension Register 1**
CB[19:16]	h94[7:4]	RW	0	0..15	Current Bank; Read Only
NB[19:16]	h94[3:0]	RW	0	0..15	Next Bank; R/W
MEX2	**h95**	**RW**	**h00**		**Memory Extension Register 2**
MM	h95[7]	RW	0	0..1	Memory Mode; R/W; 1 = use MB
MB[18:16]	h95[6:4]	RW	0	0..7	Memory Bank; R/W
IB[19:16]	h95[3:0]	RW	0	0..15	Interrupt Bank; R/W
MEX3	**h96**	**RW**	**h00**		**Memory Extension Register 3**
MB[19]	h96[7]	RW	0	0..1	Memory Bank bit; R/Wbit. See MEX2.
UB3	h96[6]	RW	0	0..1	User bits; available to the user, for MMU they are donít care.
UB4	h96[5]	RW	0	0..1	User bits; available to the user, for MMU they are donít care.
MX[19]	h96[4]	RW	0	0..1	MOVX-Bank R/W If MXM is set, these bits will be used during external data moves into or from an externally connected Data RAM.
MXM	h96[3]	RW	0	0..1	During external Data Memory accesses, the bits MX19 Ö 16 are used as address lines A19 Ö 16 instead of the current bank (CB).
MX[18:16]	h96[2:0]	RW	0	0..7	MOVX-Bank R/W If MXM is set, these bits will be used during external data moves into or from an externally connected Data RAM.
MEXSP	**h97**	**RW**	**h00**		**Memory Extension Stack Pointer**
MEXSP[6:0]	h97[6:0]	RW	0	0..255	Memory Ext Stack Pointer

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
PSW	**hD0**	**RW**	**h00**		**Program Status Word**
CY	hD0[7]	RW	0	0..1	Program Status Word
AC	hD0[6]	RW	0	0..1	Program Status Word
F0	hD0[5]	RW	0	0..1	Program Status Word
RS[1:0]	hD0[4:3]	RW	0	0..3	Program Status Word
OV	hD0[2]	RW	0	0..1	Program Status Word
F1	hD0[1]	RW	0	0..1	Program Status Word
P	hD0[0]	RW	0	0..1	Program Status Word
ACC	**hE0**	**RW**	**h00**		**Accumulator**
A[7:0]	hE0[7:0]	RW	0	0..255	Accumulator
B	**hF0**	**RW**	**h00**		**B Register**
B[7:0]	hF0[7:0]	RW	0	0..255	B register
MSIZ	**hFF**	**RW**	**h0F**		**Scratch Pad Register**
MSIZ[7:0]	hFF[7:0]	RW	15	0..255	Scratch Pad Register
UART					
SCON	**h98**	**RW**	**h00**		**Serial Control**
SM0	h98[7]	RW	0	0..1	Serial Control
SM1	h98[6]	RW	0	0..1	Serial Control
SM2	h98[5]	RW	0	0..1	Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	h98[4]	RW	0	0..1	Enables serial reception. Set by software to enable reception. Cleared by software to disable reception.
TB8	h98[3]	RW	0	0..1	Is the 9th data bit that will be transmitted in modes 2 and 3. Set or cleared by software as desired.
RB8	h98[2]	RW	0	0..1	In modes 2 and 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	h98[1]	RW	0	0..1	Is the transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	h98[0]	RW	0	0..1	Is the receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through stop bit time in the other modes, in any serial reception. Must be cleared by software.
SBUF	**h99**	**RW**	**h00**		
D[7:0]	h99[7:0]	RW	0	0..255	Serial Data Buffer

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
INTERRUPT					
IEN0	**hA8**	**RW**	**h00**		**Interrupt Enable 0**
EAL	hA8[7]	RW	0	0..1	Enable All Interrupts When set to ë0í, all interrupts are disabled. When set to ë1í, interrupts are individually enabled/disabled according to their respective bit selection.
EAD	hA8[5]	RW	0	0..1	Enable or disable Analog to digital convertor Interrupt
EU	hA8[4]	RW	0	0..1	Enable or disable UART Interrupt
ET1	hA8[3]	RW	0	0..1	Enable or disable Timer 1 Overflow Interrupt
EX1	hA8[2]	RW	0	0..1	Enable or disable External Interrupt 1
ET0	hA8[1]	RW	0	0..1	Enable or disable Timer 0 Overflow Interrupt
EX0	hA8[0]	RW	0	0..1	Enable or disable External Interrupt 0
IEN1	**hA9**	**RW**	**h00**		**Interrupt Enable 1**
EDV	hA9[5]	RW	0	0..1	Enable or disable Display V-Sync
EAV	hA9[4]	RW	0	0..1	Enable or disable Acquisition V-Sync
EXX1	hA9[3]	RW	0	0..1	Enable or disable extra external interrupt 1
EWT	hA9[2]	RW	0	0..1	Enable or disable Watchdog in timer mode
EXX0	hA9[1]	RW	0	0..1	Enable or disable extra External Interrupt 0
EX6	hA9[0]				Reserved
IEN2	**hAA**	**RW**	**h00**		**Interrupt Enable 2**
EDH	hAA[5]	RW	0	0..1	Enable or disable Display H-Sync
EAH	hAA[4]	RW	0	0..1	Enable or disable Acquisition H-Sync
ECC	hAA[3]	RW	0	0..1	Enable or disable channel change interrupt
EPW	hAA[2]	RW	0	0..1	Enable or disable PWM in timer mode
EX13	hAA[1]				Reserved
EX12	hAA[0]				Reserved
IEN3	**hAB**	**RW**	**h00**		**Interrupt Enable 3**
EADW	hAB[5]	RW	0	0..1	Enable or disable Analog to digital wake up unit
E24	hAB[4]	RW	0	0..1	Enable or disable line 24 interrupt
EX21	hAB[3]				Reserved
EX20	hAB[2]				Reserved
EX19	hAB[1]				Reserved
EX18	hAB[0]				Reserved

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
IP1	**hAC**	**RW**	**h00**		**Interrupt Priority 1**
G5P0	hAC[5]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G4P0	hAC[4]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G3P0	hAC[3]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G2P0	hAC[2]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G1P0	hAC[1]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G0P0	hAC[0]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
IRCON	**hAD**	**RW**	**h05**		**Interrupt Control Register**
EXX1R	hAD[7]	RW	0	0..1	if set, ExternalX 1-interrupt detection on rising edge at Pin P3.7
EXX1F	hAD[6]	RW	0	0..1	if set, ExternalX 1-interrupt detection on falling edge at Pin P3.7
EXX0R	hAD[5]	RW	0	0..1	if set, ExternalX 0-interrupt detection on rising edge at Pin P3.1
EXX0F	hAD[4]	RW	0	0..1	if set, ExternalX 0-interrupt detection on falling edge at Pin P3.1
EX1R	hAD[3]	RW	0	0..1	if set, External 1-interrupt detection on rising edge at Pin P3.3
EX1F	hAD[2]	RW	1	0..1	if set, External 1-interrupt detection on falling edge at Pin P3.3
EX0R	hAD[1]	RW	0	0..1	if set, External 0-interrupt detection on rising edge at Pin P3.2
EX0F	hAD[0]	RW	1	0..1	if set, External 0-interrupt detection on falling edge at Pin P3.2

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
IP0	**hB8**	**RW**	**h00**		**Interrupt Priority 0**
G5P1	hB8[5]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G4P1	hB8[4]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G3P1	hB8[3]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G2P1	hB8[2]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G1P1	hB8[1]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
G0P1	hB8[0]	RW	0	0..1	Interrupt Group Priority Level as follows: 0 0: Interrupt Group x is set to priority level 0 (lowest). 0 1: Interrupt Group x is set to priority level 1. 1 0: Interrupt Group x is set to priority level 2. 1 1: Interrupt Group x is set to priority level 3 (highest).
CISR0	**hC0**	**RW**	**h00**		**Central Interrupt Service 0**
L24	hC0[7]	RW	0	0..1	1: Line 24 start interrupt occurred, source bit set by hardware, Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
ADC	hC0[6]	RW	0	0..1	1: Analog to digital conversion complete source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
WTmr	hC0[5]	RW	0	0..1	1: Watchdog in timer mode overflow source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred. On reset this bit is initialized to 0, however if timer mode is selected and timer is running, every over flow of timer will set this bit. Therefore software must clear this bit before enabling the corresponding interrupt.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
AVS	hC0[4]	RW	0	0..1	1: Acquisition vertical sync interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
DVS	hC0[3]	RW	0	0..1	1: Display Vertical sync interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
PWtmr	hC0[2]	RW	0	0..1	1: PWM in timer mode overflow interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred. On reset this bit is initialized to 0,however if timer mode is selected and timer is running, every over flow of timer will set this bit. Therefore software must clear this bit before enabling the corresponding interrupt.
AHS	hC0[1]	RW	0	0..1	1: Acquisition horizontal sync interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
DHS	hC0[0]	RW	0	0..1	1: Display horizontal sync interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
CISR1	**hC8**	**RW**	**h00**		**Central Interrupt Service 1**
CC	hC8[7]	RW	0	0..1	1: Channel change interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
ADW	hC8[6]	RW	0	0..1	1: ADC wake up interrupt source bit set by hardware. Source bit must be reset by software after servicing the interrupt. 0: Interrupt has not occurred.
IEX1	hC8[1]	RW	0	0..1	External Extra Interrupt 1 edge flag. Set by hardware when external interrupt edge detected. Must be cleared by software. Port P3.7 must be in input mode to use this interrupt.
IEX0	hC8[0]	RW	0	0..1	External Extra Interrupt 0 edge flag. Set by hardware when external interrupt edge detected. Must be cleared by software. Port P3.1 must be in input mode to use this interrupt.
SNDCSTL	**hDF**	**RW**			**Sandcastle**
HYS	hDF[6]	RW			Definition of Hysteresis (slave mode/sandcastle input) Defines the voltage range for the Hysteresis: 0: Hysteresis set to 0.325 V. 1: Hysteresis set to 0.150 V.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
SND_V[2:0]	hDF[5:3]	RW			Slicing Level Vertical Sync-Pulses (slave mode/sandcastle input) To fit the requirements of various applications the input circuit of the sandcastle decoder is free programmable. The slicing levels for the vertical pulses can be varied in a range from 0.67 V up to 1.83 V in steps of about 0.16 V: 000: Vertical Slicing Level set to 0.67 V. 001: Vertical Slicing Level set to 0.83 V. 010: Vertical Slicing Level set to 1.00 V. 011: Vertical Slicing Level set to 1.17 V. 100: Vertical Slicing Level set to 1.33 V. 101: Vertical Slicing Level set to 1.50 V. 110: Vertical Slicing Level set to 1.67 V. 111: Vertical Slicing Level set to 1.83 V. These are nominal values. They may also differ with supply voltage.
SND_H[2:0]	hDF[2:0]	RW			Slicing Level Horizontal Sync-Pulses (slave mode/sandcastle input) To fit the requirements of various applications the input circuit of the sandcastle decoder is free programmable. The slicing levels for the horizontal pulses can be varied in a range from 1.33 V up to 2.50 V in steps of about 0.16 V: 000: Horizontal Slicing Level set to 1.33 V 001: Horizontal Slicing Level set to 1.50 V 010: Horizontal Slicing Level set to 1.67 V 011: Horizontal Slicing Level set to 1.83 V 100: Horizontal Slicing Level set to 2.00 V 101: Horizontal Slicing Level set to 2.17 V 110: Horizontal Slicing Level set to 2.33 V 111: Horizontal Slicing Level set to 2.50 V These are nominal values. They may also differ with supply voltage.
WATCHDOG					
WDT_rel	**hB1**	**RW**	**h00**		**Watchdog Reload**
WDTrel[7:0]	hB1[7:0]	RW	0	0..255	Reload value of the watchdog timer (also in timer-mode), is loaded in the upper 8 bit of the watchdog counter at WDT-start and ñreload and also at timer start.
WDT_ctrl	**hB2**	**RW**	**h00**		**Watchdog Control**
WDT_in	hB2[7]	RW	0	0..1	Watchdog Control
WDT_start	hB2[6]	RW	0	0..1	Watchdog Control
WDT_narst	hB2[5]	RW	0	0..1	Watchdog Control
WDT_rst	hB2[4]	RW	0	0..1	Watchdog Control
WDT_refresh	**hB3**	**RW**	**h00**		**Watchdog Refresh**
WDT_ref	hB3[7]	RW	0	0..1	Watchdog Refresh
WDT_tmr	hB3[6]	RW	0	0..1	Watchdog Refresh
WTmr_strt	hB3[5]	RW	0	0..1	Watchdog Refresh
WTmr_ov	hB3[4]	RW	0	0..1	Watchdog Refresh
WDT_low	**hB4**	**RW**	**h00**		**WDT Timer Low**
WDTlow[7:0]	hB4[7:0]	RW	0	0..255	Counter value of the watchdog timer; low byte.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
WDT_high	**hB5**	**RW**	**h00**		**WDT Timer High**
WDThi[7:0]	hB5[7:0]	RW	0	0..255	Counter value of the watchdog timer; high byte.
CRT					
CRT_rell	**hB7**	**RW**	**h00**		**CRT Reload Low**
RelL[7:0]	hB7[7:0]	RW	0	0..255	CRT reload low byte
CRT_relh	**hB9**	**RW**	**h00**		**CRT Reload High**
RelH[7:0]	hB9[7:0]	RW	0	0..255	CRT reload high byte
CRT_capl	**hBA**	**RW**	**h00**		**CRT Capture Low**
CapL[7:0]	hBA[7:0]	RW	0	0..255	CRT capture low byte
CRT_caph	**hBB**	**RW**	**h00**		**CRT Capture High**
CapH[7:0]	hBB[7:0]	RW	0	0..255	CRT capture high byte
CRT_mincapl	**hBC**	**RW**	**h00**		**CRT Min Capture Low**
MinL[7:0]	hBC[7:0]	RW	0	0..255	CRT min capture low
CRT_mincaph	**hBD**	**RW**	**h00**		**CRT Min Capture High**
MinH[7:0]	hBD[7:0]	RW	0	0..255	CRT min capture high
CRT_con0	**hBE**	**RW**	**h00**		**CRT Control 0**
OV	hBE[7]	RW	0	0..1	Will be set by hardware, if counter overflow has occurred; must be cleared by software.
PR	hBE[6]	RW	0	0..1	If cleared, 2-bit prescaler; if set, 3-bit prescaler.
PLG	hBE[5]	RW	0	0..1	If set, Timer polling mode selected, capture function is automatically disabled, reading capture registers will now show current timer value.
REL	hBE[4]	RW	0	0..1	If set, counter will be reloaded simultaneously with capture event.
RUN	hBE[3]	RW	0	0..1	Run/stop the CRT counter.
RISE	hBE[2]	RW	0	0..1	Capture (and if REL = ë1í, reload) on rising edge.
FALL	hBE[1]	RW	0	0..1	Capture (and if REL = ë1í, reload) on falling edge.
SEL	hBE[0]	RW	0	0..1	If set, P3.3 is selected for capture input, otherwise P3.2.
CRT_con1	**hBF**	**RW**	**h00**		**CRT Control 1**
PR1	hBF[2]	RW	0	0..1	1: Divides input further by 8. 0: Not divided by 8.
First	hBF[1]	RW	0	0..1	1: Indicates first event. 0: Indicates not first event.
Start	hBF[0]	RW	0	0..1	1: Controller sets this bit enter the SSU mode and to indicate it is expecting a new telegram. When an event occurs CAPUTR unit sets First bit. Upon next event, hardware resets the first bit and interrupt is generated based on MIN_CAP register. 0: Not SSU mode.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
PWM					
PWM_comp8_0	**hC1**	**RW**	**h00**		**PWM 8 bit Compare 0**
PC80[7:0]	hC1[7:0]	RW	0	0..255	PWM 8bit compare 0
PWM_comp8_1	**hC2**	**RW**	**h00**		**PWM 8 bit Compare 1**
PC81[7:0]	hC2[7:0]	RW	0	0..255	PWM 8bit compare 1
PWM_comp8_2	**hC3**	**RW**	**h00**		**PWM 8 bit Compare 2**
PC82[7:0]	hC3[7:0]	RW	0	0..255	PWM 8bit compare 2
PWM_comp8_3	**hC4**	**RW**	**h00**		**PWM 8 bit Compare 3**
PC83[7:0]	hC4[7:0]	RW	0	0..255	PWM 8bit compare 3
PWM_comp8_4	**hC5**	**RW**	**h00**		**PWM 8 bit Compare 4**
PC84[7:0]	hC5[7:0]	RW	0	0..255	PWM 8bit compare 4
PWM_comp8_5	**hC6**	**RW**	**h00**		**PWM 8 bit Compare 5**
PC85[7:0]	hC6[7:0]	RW	0	0..255	PWM 8bit compare 5
PWM_comp14_0	**hC7**	**RW**	**h00**		**PWM 14 bit Compare 0**
PC140[7:0]	hC7[7:0]	RW	0	0..255	PWM 14bit compare 0 This bits define the high time of the output. If all bits are 0, the high time is 0 internal clocks. If all bits are 1, the high time of a base cycle is 255 internal clocks.
PWM_comp14_1	**hC9**	**RW**	**h00**		**PWM 14 bit Compare 1**
PC141[7:0]	hC9[7:0]	RW	0	0..255	PWM 14bit compare 1 This bits define the high time of the output. If all bits are 0, the high time is 0 internal clocks. If all bits are 1, the high time of a base cycle is 255 internal clocks.
PWM_compext14_0	**hCA**	**RW**	**h00**		**PWM 14 bit Compext 0**
PCX140[7:2]	hCA[7:2]	RW	0	0..63	PWM 14bit comp ext 0
PWM_compext14_1	**hCB**	**RW**	**h00**		**PWM 14 bit Compext 1**
PCX141[7:2]	hCB[7:2]	RW	0	0..63	PWM 14bit comp ext 1
PWM_cl	**hCC**	**RW**			**PWM Counter Low Byte**
PWC[7:0]	hCC[7:0]	RW		0..255	PWM counter low byte
PWM_ch	**hCD**	**RW**			**PWM Counter High Byte**
PWM_Tmr	hCD[7]	RW	0	0..1	Start/stop timer when all PWM channels are disabled. If this bit is set, the PWM timer will be reset and starts counting. If this bit is cleared, the PWM timer stops. The PWM_Tmr bit could not be written (set) if one of the PWM channels is enabled (PWM_en not all zero). PWM_en register could not be written (set) if the PWM_Tmr bit is set.
OV	hCD[6]	RW	0	0..1	Overflow bit for the timer mode.
PWC[13:8]	hCD[5:0]	RW		0..63	PWM counter high byte

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
PWM_EN	**hCE**	**RW**	**h00**		**PWM Channel Enable**
PE[7:0]	hCE[7:0]	RW	0	0..255	PWM channel enable 0: The corresponding PWM-channel is disabled. P1.i functions as normal bidirectional I/O-port. 1: The corresponding PWM-channel is enabled. PE0 Ö PE5 are channels with 8-bit resolution, while PE6 and PE7 are channels with 14-bit resolution.
ADC					
CADC0	**hD1**	**RW**	**h00**		**ADC Channel 0 Result**
CADC0[7:0]	hD1[7:0]	RW	0	0..255	ADC result of channel 0 After finishing the A to D conversion the processor is informed by means of an interrupt. The interrupt service routine can now take the conversion result of channel 1 from CADC0. The result will be available for about 46 ms after the interrupt.
CADC1	**hD2**	**RW**	**h00**		**ADC Channel 1 Result**
CADC1[7:0]	hD2[7:0]	RW	0	0..255	ADC result of channel 1 After finishing the A to D conversion the processor is informed by means of an interrupt. The interrupt service routine can now take the conversion result of channel 2 from CADC1. The result will be available for about 46 ms after the interrupt.
CADC2	**hD3**	**RW**	**h00**		**ADC Channel 2 Result**
CADC2[7:0]	hD3[7:0]	RW	0	0..255	ADC result of channel 2 After finishing the A to D conversion the processor is informed by means of an interrupt. The interrupt service routine can now take the conversion result of channel 3 from CADC2. The result will be available for about 46 ms after the interrupt.
CADC3	**hD4**	**RW**	**h00**		**ADC Channel 3 Result**
CADC3[7:0]	hD4[7:0]	RW	0	0..255	ADC result of channel 3 After finishing the A to D conversion the processor is informed by means of an interrupt. The interrupt service routine can now take the conversion result of channel 4 from CADC3. The result will be stable for about 46 ms after the interrupt.
CADCCO	**hD5**	**RW**	**h00**		**ADC Configuration**
ADWULE	hD5[4]	RW	0	0..1	Defines threshold level for wake up. A special wake up unit has been included to allow a system wake up as soon as the analog input signal on pin P2.0 drops below a predefined level. ADWULE defines the threshold level. ADWULE = 0: Threshold level corresponds to fullscale - 4 LSB. This means that if the digital input value drops below 255 - 4 = 251 an interrupt will be triggered. In voltages that is 2.5 V - 0.039 V = 2.461 V. ADWULE = 1: threshold level corresponds to fullscale - 16 LSB. This means that if the digital input value drops below 255 - 16 = 239 an interrupt will be triggered. In voltages that is 2.5 V - 0.156 V = 2.344 V.
AD[3:0]	hD5[3:0]	RW	0	0..15	Defines whether the corresponding port-pin is used as analog input or as digital input. 0: Port pin is digital input (the analog value has less precision). 1: Port pin is analog input (the digital value is always 0).

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
SFRIF					
PSAVEX	**hD7**	**RW**	**h00**		**Power Save Extra Register**
Clk_src	hD7[2]	RW	h00	0 ..1	Clock Source 0:200 MHz PLL (33.33 MHz system clock) selected. 1: PLL is bypassed oscillator clock 6 MHz (3 MHz system clock selected). In this mode slicer, acquisition, DAC and display generator are disabled.
PLL_Res	hD7[1]	RW	h00	0 ..1	PLL Reset 0:PLL not reset. 1:PLL reset. PLL reset sequence requires that PLL_res = 1 for 10 µs then PLL_res = 0, after that 150 µs are required till PLL is locked.
PLLS	hD7[0]	RW	h00	0 ..1	PLL Sleep 0:Power-save mode not started. 1:Power-save mode started. Before the PLL is switched to power-save mode (PLLS = 1), the SW has to switch the clock source from 200 MHz PLL clock to the 6 MHz oscillator clock (CLK_src = 1). To switch back to the normal mode, software has to end the PLL power save mode (PLLS = 0), reset the PLL for 10 µs (3 machine cycles), PLL_res = 1 the back to 0, wait for 150 µs (38 machine cycles) and then switch back to the PLL clock.
PSAVE	**hD8**	**RW**	**hF4**		**Power Save Register**
CADC	hD8[4]	RW	1	0..1	Power Save CADC 0: Power-save mode not started. 1: Power-save mode started. In Power save mode CADC is disabled but the CADC-Wake-Up-Unit is active.
WAKUP	hD8[3]	RW	1	0..1	Power Save CADC-Wake-Up-Unit 0: Power-save mode not started. 1: Power-save mode started. In power-save mode the CADC-Wake-Up-Unit is disabled. Power-save mode of wake up unit is only useful in saving power when CADC bit is set.
SLI_ACQ	hD8[2]	RW	1	0..1	Reset XDFP 0: XDFP running 1: XDFP reset
DISP	hD8[1]	RW	0	0..1	Reset Chip 0: no action 1: reset active (RESQ pin low)
PERI	hD8[0]	RW	0	0..1	Software Reset Enable 0: no software reset possible 1: software reset possible

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
STRVBI	**hD9**	**RW**	**h00**		**Configuration ACQ & Slicer**
ACQON	hD9[7]	RW	0	0..1	Enable Acquisition: 0: The ACQ interface does not access memory (immediately inactive). 1: The ACQ interface is active and writes data to memory (switching on is synchronous to V).
Reserved	hD9[6]	RW	0	0..1	Config ACQ & Slicer
ACQ_STA	hD9[5]	RW	0	0..1	First Framing code after vertical sync: 0: No framing code after vertical sync has been detected. 1: Framing code after vertical sync has been detected. The bit is set by hardware and cleared by software.
VBIADR[3:0]	hD9[3:0]	RW	0	0..31	Defines the 5 MSBs of the start address of the VBI buffer (the LSBs are fixed to 0x000). The VBI buffer location can be aligned to any 1 kByte memory segment.
PCLK1	**hDA**	**RW**	**h01**		**DTO Pixel Frequency Factor 1**
PF[10:8]	hDA[3:0]	RW	1	0..15	Pixel Frequency factor (LSBs) This register defines the relation between the output pixel frequency and the frequency of the crystal. The pixel frequency does not depend on the line frequency. It can be calculated by the following formula: Fpixel = PF * 324 MHz / 8192 The pixel frequency can be adjusted in steps of 36.6 kHz. After power-on this register is set to 328d. So, the default pixel frequency is set to 12.97 MHz. Attention: Register values greater then 983d generate pixel frequencies which are outside of the specified boundaries.
PCLK0	**hDB**	**RW**	**h48**		**DTO Pixel Frequency Factor 0**
PF[7:0]	hDB[7:0]	RW	72	0..255	Pixel Frequency factor (LSBs) This register defines the relation between the output pixel frequency and the frequency of the crystal. The pixel frequency does not depend on the line frequency. It can be calculated by the following formula: Fpixel = PF * 324 MHz / 8192 The pixel frequency can be adjusted in steps of 36.6 kHz. After power-on this register is set to 328d. So, the default pixel frequency is set to 12.97 MHz. Attention: Register values greater then 983d generate pixel frequencies which are outside of the specified boundaries.
DSYNC					
SCR1	**hE1**	**RW**	**h00**		**DSync Control 1**
Reserved	hE1[7]				Reserved for internal use. Must be set to 1 (see Section 2.14. on page 109).
RGB_G[1:0]	hE1[6:5]	RW			Used for DAC setup purpose (see Section 2.14. on page 109)

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
COR_BL	hE1[4]	RW			3-Level Contrast Reduction Output By means of COR_BL the user is able to switch the COR signal to a three level signal providing BLANK and contrast reduction information on pin BLANK/COR. 0: Two level signal for contrast reduction. 1: Three level signal; Three level signal Level0: BLANK off; COR off. Level1: BLANK off; COR on. Level2: BLANK on; COR off. Note: See Section 4.10.3. on page 165 for the detailed specification of these levels.
VSU[3:0]	hE1[3:0]	RW	0	0..15	Vertical Set Up Time The vertical sync signal is internally sampled with the next edge of the horizontal sync edge. The phase relation between V and H differs from application to application. To guarantee (vertical) jitter free processing of external sync signals, the vertical sync impulse can be delayed before it is internally processed. The following formula shows how to delay the external V-sync before it is internally latched and processed. tV_delay = 3.84 us * VSU
SCR0	**hE2**	**RW**	**h00**		**DSync Control 0**
RGB_D[1:0]	hE2[7:6]	RW	0	0..3	RGB/COR Delay Circuitry In some applications of our customers the blanking is fed through other devices before it is used as a signal to control the multiplexing of video/RGB-mix. These other devices may create a delay of the blank signal. If no special effort is taken, this delay would create a vertical band at the beginning and the end of the active blanking zone. To compensate this, the generated RGB and the COR signals can be delayed by TVT in reference to the generated blank signal. This delay is always a multiple of the pixel-frequency from zero delay up to 3 times pixel delay: 00: Zero delay of RGB/COR-output in reference to BLANK-output. 01: One pixel delay of RGB/COR-output in reference to BLANK-output. 10: Two pixel delay of RGB/COR-output in reference to BLANK-output. 11: Three pixel delay of RGB/COR-output in reference to BLANK-output.
HP	hE2[5]	RW	0	0..1	H-Pin Polarity This bit defines the polarity of the H pin (master and slave mode). 0: Normal polarity (active high). 1: Negative polarity.
VP	hE2[4]	RW	0	0..1	V-Pin Polarity This bit defines the polarity of the V pin (master and slave mode). 0: Normal polarity (active high). 1: Negative polarity.
INT	hE2[3]	RW	0	0..1	Interlace / Non-interlace TVT can either generate an interlaced or a non-interlaced timing (master mode only). Interlaced timing can only be created if VLR is an odd number. 0: Interlaced timing is generated. 1: Non-interlaced timing is generated.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
SNC	hE2[2]	RW		0..1	Sandcastle Sync (Slave mode only) To input pins are reserved for synchronisation. These input pins can be used as two separated sync inputs or as one single sync input. If two separated sync inputs is selected horizontal syncs are fed in at H pin and vertical syncs are fed in a V pin. If one single input pin is selected the H pin is used as a sandcastle input pin. 0: H/V-sync input at H/V pins 1: Sancastle input H pin
VCS	hE[1]	RW		0..1	Video Composite Sync VCS defines the sync output at pin V (master mode only) 0: At pin V the vertical sync appears 1: At pin V a composite sync signal (including equalizing pulses, H-Sync and V_Sync) is generated (VCS). The length of the equalizing pulses have fixed values as described in the timing specifications. **Note: Don't forget to set registers VLR and HPR (64 µs) according to your requirements.**
MAST	hE[0]	RW			Master/Slave Mode This bit defines the configuration of the sync system (master or slave mode) and also the direction (input/output) of the V, H pins. 0: Slave mode. H, V pins are configured as inputs 1: Master mode. H, V pins are configured as outputs. **Note: Switching from slave to master mode resets the internal H, V counters in that way, that the phase shift during the switch can be minimized. In slave mode registers VLR and HPR are not used.**
SDV1	**hE3**	**RW**	**h00**		**DSync V Delay 1**
SDV[9:8]	hE3[1:0]	RW	0	0..3	Vertical Sync Delay (master and slave mode) This register defines the delay (in lines) from the vertical sync to the first line of character display area on the screen.
SDV0	**hE4**	**RW**	**h20**		**DSync V Delay 0**
SDV[7:0]	hE4[7:0]	RW	32	0..255	Vertical Sync Delay (master and slave mode) This register defines the delay (in lines) from the vertical sync to the first line of character display area on the screen.
SDH1	**hE5**	**RW**	**h00**		**DSync H Delay 1**
SDH[11:8]	hE5[3:0]	RW	0	0..15	Horizontal Sync Delay (master and slave mode) This register defines the delay (in pixels) from the horizontal sync to the first pixel character display area on the screen.
SDH0	**hE6**	**RW**	**h48**		**DSync H Delay 0**
SDH[7:0]	hE6[7:0]	RW	72	0..255	Horizontal Sync Delay (master and slave mode) This register defines the delay (in pixels) from the horizontal sync to the first pixel character display area on the screen.
HCR1	**hE7**	**RW**	**h0A**		**DSync H Clamp End**
EHCR[7:0]	hE7[7:0]	RW	10	0..255	End of Horizontal Clamp Phase (master and slave mode) This register defines the end of the horizontal clamp phase from the positive edge of the horizontal sync impulse (at normal polarity). The end of clamp phase can be calculated by the following formula: tH_clmp_e = 480 ns * EHCR If EHCR is smaller than BHCR the clamp phase will appear during Hsync.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
HCR0	**hE9**	**RW**	**h00**		**DSync H Clamp Begin**
BHCR[7:0]	hE9[7:0]	RW	0	0..255	Beginning of Horizontal Clamp Phase (master and slave mode) This register defines the delay of the horizontal clamp phase from the positive edge of the horizontal sync impulse (normal polarity is assumed). The beginning of clamp phase can be calculated by the following formula: tH_clmp_b = 480 ns * BHCR If EHCR is smaller than BHCR the clamp phase will appear during Hsync.
BVCR	**hEA**	**RW**	**h00**		**DSync V Clamp Begin 1**
BVCR[9:8]	hEA[1:0]	RW	0	0..3	Beginning of Vertical Clamp Phase (master and slave mode) This register defines the beginning of the vertical clamp phase from the positive edge of the vertical sync impulse (at normal polarity) in count of lines. If EVCR is smaller than BVCR than the clamp phase will appear during Vsync.
BVCR0	**hEB**	**RW**	**h00**		**DSync V Clamp Begin 0**
BVCR[7:0]	hEB[7:0]	RW	0	0..255	Beginning of Vertical Clamp Phase (master and slave mode) This register defines the beginning of the vertical clamp phase from the positive edge of the vertical sync impulse (at normal polarity) in count of lines. If EVCR is smaller than BVCR than the clamp phase will appear during Vsync.
EVCR1	**hEC**	**RW**	**h00**		**DSync V Clamp End 1**
EVCR[9:8]	hEC[1:0]	RW	0	0..3	End of Vertical Clamp Phase (master and slave mode) This register defines the end of the vertical clamp phase from the positive edge of the vertical sync impulse (at normal polarity) in count of lines. If EVCR is set to a value smaller than BVCR than the vertical blanking phase will last over the vertical blanking interval. If EVCR is smaller than BVCR than the clamp phase will appear during Vsync.
EVCR0	**hED**	**RW**	**h04**		**DSync V Clamp End 0**
EVCR[7:0]	hED[7:0]	RW	4	0..255	End of Vertical Clamp Phase (master and slave mode) This register defines the end of the vertical clamp phase from the positive edge of the vertical sync impulse (at normal polarity) in count of lines. If EVCR is set to a value smaller than BVCR than the vertical blanking phase will last over the vertical blanking interval. If EVCR is smaller than BVCR than the clamp phase will appear during Vsync.
VLR1	**hEE**	**RW**	**h02**		**DSync Vertical Line 1**
Odd_Ev	hEE[6]	RW	0	0..1	ODD/EVEN detection (slave mode only) Used as a interface from the hardware odd/even field detection to software. Set to 1 for odd fields and to 0 for even fields.

Table 3–4: SFR register description, continued

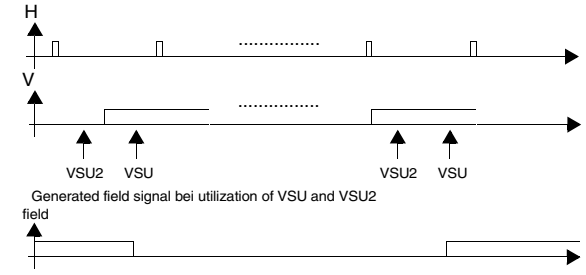
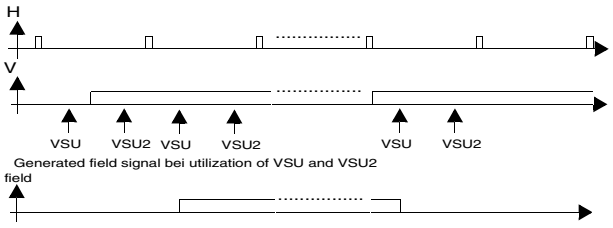
Name	Sub	Dir	Reset	Range	Function
VSU2[3:0]	hEE[5:2]	RW	0	0..15	Vertical Set Up Time 2 (slave mode only) To realize the odd/even detection of a field next to VSU a second vertical setup time VSU2 is defined by the VSU2 register bits. This horizontal delay is used to recognize the VSYNC to another time than it is recognized at VSU. The field detection is realized by detecting if in between these two latching-points the VSync is rising or stable: tV_delay2 = 3.84 us * VSU2 If VSYNC became active for both VSU and VSU2, an odd field is detected. If VSYNC became active only for VSU an even field is detected:  with inverted VSU and VSU2: 
VL[9:8]	hEE[1:0]	RW	2	0..3	DSync Vetical line 1
VLR0	**hEF**	**RW**	**h71**		**DSync Vertical Line 0**
VL[7:0]	hEF[7:0]	RW	113	0..255	Amount of Vertical Lines in a Frame (master mode only) TVT generates in sync master mode vertical sync impulses. If for example a normal PAL timing should be generated, set this register to “625d” and set the interlace bit to “0”. The hardware will generate a vertical impulse periodically after 312.5 lines. If a non-interlace picture with 312 lines should be generated, set this register to “312” and set the interlace bit to “1”. The hardware will generate a vertical impulse every 312 lines. A progressive timing can be generated by setting VLR to “625” and interlace to “0”.
HPR1	**hF1**	**RW**	**h08**		**DSync Horizontal Period 1**
HPR[11:8]	hF1[3:0]	RW	8	0..15	Horizontal Period Factor (master mode only) This register allows to adjust the period of the horizontal sync signal. The horizontal period is independent from the pixel frequency and can be adjusted with the following resolution: tH-period = HP x 30 ns
HPR0	**hF2**	**RW**	**h55**		**DSync Horizontal Period 0**
HPR[7:0]	hF2[7:0]	RW	85	0..255	Horizontal Period Factor (master mode only) This register allows to adjust the period of the horizontal sync signal. The horizontal period is independent from the pixel frequency and can be adjusted with the following resolution: tH-period = HP x 30 ns

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
DISPLAY					
POINTARRAY1_1	**hF3**	**RW**	**h00**		**Display Pointer 1 High Byte**
Point1[13:8]	hF3[7:0]	RW	0	0..255	Display Pointer 1 high byte
POINTARRAY1_0	**hF4**	**RW**	**h06**		**Display Pointer 1 Low Byte**
Point1[7:0]	hF4[7:0]	RW	6	0..255	Display Pointer 1 low byte
POINTARRAY0_1	**hF5**	**RW**	**h00**		**Display Pointer 0 High**
Point0[13:8]	hF5[7:0]	RW	0	0..255	Display Pointer 0 high byte
POINTARRAY0_0	**hF6**	**RW**	**h00**		**Display Pointer 1 High**
Point0[7:0]	hF6[7:0]	RW	0	0..255	Display Pointer 0 low byte
OSD_ctrl	**hF8**	**RW**	**h00**		**Display OSD Control**
En_Ld_Cur	hF8[3]	RW	0	0..1	Used to avoid the download of the parameter settings of the GDW from the RAM to the local display generator register bank. 0: Download disabled. 1: Download enabled.
En_DGOut	hF8[2]	RW	0	0..1	Used to disable/enable the output of the display generator. If display generator is disabled the RGB outputs of the TVT are set to black and the outputs BLANK and COR are set to: COR = ENABLECOR BLANK = ENABLEBLA If display generator is enabled the display information RGB, COR and BLANK is generated according to the parameter settings in the XRAM. 0: Display generator is disabled. 1: Display generator is enabled.
Dis_Cor	hF8[1]	RW	0	0..1	Defines the level of the COR output if display generator is disabled.
Dis_Blank	hF8[0]	RW	0	0..1	Defines the level of the BLANK output if display generator is disabled.
TAP	**hF9**				**Reserved**
TAP	**hFA**				**Reserved**
Optimized OPTI0	**hFD**				**Reserved**
FREQSEL(1)	hFD[7]	RW			
FREQSEL(2)	hFD[6]	RW			
OSCPD	hFD[5]	RW			
Additional Registers					
CSCR0	**hDD**	RW	**h00**		**Central Special Control 0**
ENETCLK	hDD[5]	RW	h00		UART baud rate clk source bits
ENERCLK	hDD[4]	RW	h00		Selects between 6 MHz and system clock.
P4_7_Alt	hDD[3]	RW	h00		Selects the output function of the port 0: Port function is selected 1: Port 4.7 alternate function is selected (see VS_OE) For input port mode or slave mode VS input mode, port must be switched to input mode by writting “1” to the port latch.

Table 3–4: SFR register description, continued

Name	Sub	Dir	Reset	Range	Function
VS_OE	hDD[2]	RW	h00		0: P4.7 alternate output mode, Odd/Even selected 1: P4.7 alternate output mode, Vertical Sync selected See Section 2.13. on page 69 register SCR0, for Vertical Sync details
O_E_P3_0	hDD[1]	RW	h00		0: Port 3.0 port mode selected 1: Port 3.0 works as a Odd/Even output
O_E_Pol	hDD[0]	RW	h00		0: Odd = 1, Even = 0 1: Odd = 0, Even = 1 Note: Polarity is true for both P3.0 and P4.7,
CSCR1	**hDE**	RW	**h00**		**Central Special Control 1**
IntSrc1	hDE[7]	RW	h00		0: Port 3.3 is the source of the interrupt, 1: SSU is the source of interrupt,(Application Note: Use with SEL 0 0),
IntSrc0	hDE[6]	RW	h00		0: Port 3.3 is the source of the interrupt, 1: SSU is the source of interrupt,(Application Note: Use with SEL 0 1),
ENARW	hDE[3]	RW	h00		0: Port P4.2 and P4.3 function as port pins 1: Port 4.2 and P4.3 function as RD and WR signal outputs.
$\overline{A19}$_P4_4	hDE[2]	RW	h00		0: Pin functions as address line 1: Pin function as port
$\overline{A18}$_P4_1	hDE[1]	RW	h00		0: Pin functions as address line 1: Pin function as port
$\overline{A17}$_P4_0	hDE[0]	RW	h00		0: Pin functions as address line 1: Pin function as port

3.5. ACQ Register Block Index

Table 3–5: ACQ block index

Name	Page
FIELD_PARAMETER	139
LINE_PARAMETER	141

3.6. ACQ Register Index

Table 3–6: ACQ register bits index

Name	Page
FC3[15:8]	139
FC3[7:0]	139
FC3MASK[15:8]	139
FC3MASK [7:0]	139
FC1[7:0]	139
AGDON	139
AFRON	139
ANOON	139
GDPON	139
GDNON	139
FREON	139
NOION	139
FULL	140
NOISE(0)	140
FREATTF	140
STAB	140
VDOK	140
FIELD	140
NOISE(1)	140
GRDON	140
GRDSIGN	141
LEOFLI[11:8]	141

Table 3–6: ACQ register bits index, continued

Name	Page
LEOFLI[7:0]	141
DINCR[15:8]	141
DINCR[7:0]	141
NORM[2:0]	141
FCSEL[1:0]	141
FC1ER	142
VCR	142
MLENGTH[7:5]	142
ALENGTH[4:3]	142
CLKDIV[2:0]	142
PERR[7:2]	143
TLDE	143
FCOK	143

3.7. ACQ Register Address Index

Table 3–7: ACQ subaddress index

Sub	Data Bits								Reset
	7	6	5	4	3	2	1	0	
h0000	FC3[15:8]								h0000
h0001	FC3[7:0]								h0000
h0002	FC3MASK[15:8]								h0000
h0003	FC3MASK [7:0]								h0000
h0004	FC1[7:0]								h0000
h0005	AGDON	AFRON	ANOON	GDPON	GDNON	FREON	NOION	FULL	h0000
h0006	NOISE(0)	FREATTF	STAB	VDOK	FIELD	NOISE(1)	GRDON	GRDSIGN	h0000
h0007	LEOFLI[11:8]								h0000
h0008	LEOFLI[7:0]								h0000
h000D	DINCR[15:8]								h0000
h000E	DINCR[7:0]								h0000
h000F	NORM[2:0]			FCSEL[1:0]		FC1ER	VCR	Reserved	h0000
h0010	MLENGTH[7:5]			ALENGTH[4:3]			CLKDIV[2:0]		h0000
h00011	PERR[7:2]						TLDE	FCOK	h0000

3.8. ACQ Register Description

Table 3–8: ACQ register description

Name	Addr	Dir	Sync	Reset	Range	Function
FIELD_PARAMETER						
ACQFP0	**h0000**	**RW**	**VS**	**h0000**		
FC3[15:8]	h0000[7:0]	RW	VS	0	0..255	
ACQFP1	**h0001**	**RW**	**VS**	**h0000**		
FC3[7:0]	h0001[7:0]	RW	VS	0	0..255	Programmable 16-bit Framing code MSB corresponds to first received bit of FC
ACQFP2	**h0002**	**RW**	**VS**	**h0000**		
FC3MASK[15:8]	h0002[7:0]	RW	VS	0	0..255	
ACQFP3	**h0003**	**RW**	**VS**	**h0000**		
FC3MASK [7:0]	h0003[7:0]	RW	VS	0	0..255	Mask for Programmable 16-bit Framing Code MSB corresponds to first received bit of FC 0: this bit is checked 1: this bit is don't care
ACQFP4	**h0004**	**RW**	**VS**	**h0000**		
FC1[7:0]	h0004[7:0]	RW	VS	0	0..255	Programmable 8-bit Framing Code MSB corresponds to first received bit of FC
ACQFP5	**h0005**	**RW**	**VS**	**h0000**		
AGDON	h0005[7]	RW	VS	0	0..1	Automatic group delay compensation 0: Automatic compensation Off 1: Automatic compensation On
AFRON	h0005[6]	RW	VS	0	0..1	Automatic frequency depending attenuation compensation 0: Automatic compensation Off 1: Automatic compensation On
ANOON	h0005[5]	RW	VS	0	0..1	Automatic noise compensation 0: Automatic compensation Off 1: Automatic compensation On
GDPON	h0005[4]	RW	VS	0	0..1	Allpass Filter for positive Group Delay Distortion 0: Group delay compensation depends on AGD_ON 1: Positive group delay compensation is always on
GDNON	h0005[3]	RW	VS	0	0..1	Allpass Filter for negative Group Delay Distortion 0: Group delay compensation depends on AGD_ON 1: Negative group delay compensation is always on
FREON	h0005[2]	RW	VS	0	0..1	Peaking Filter to compensate Frequency Attenuation 0: Frequency depending attenuation compensation depends on AFRE_ON 1: Frequency depending attenuation compensation is always on
NOION	h0005[1]	RW	VS	0	0..1	Noise Detection and Compensation 0: Noise compensation depends on ANOI_ON 1: Noise compensation is always on

Table 3–8: ACQ register description, continued

Name	Addr	Dir	Sync	Reset	Range	Function
FULL	h0005[0]	RW	VS	0	0..1	Full Channel Mode 0: Full channel mode off 1: Full channel mode on Note: Donít forget to reserve enough memory for the VBI buffer and to initialized the appropriate line parameters.
ACQFP6	**h0006**	**RW**	**VS**	**h0000**		
NOISE(0)	h0006[7]	RW	VS	0	0..15	Hsync Window Defines the width of the window for the acceptance of incoming H-sync pulses 0000: +/- 1.92us 0001: +/- 3.84us ... 1111: +/- 30.072us
FREATTF	h0006[6]	RW	VS	0	0..15	Precision Control for WSS-FC-Check The value of wss_pre determines how error values around edges inside the WSS-Framing-Code are accepted. 0: any error acepted 1: 11 errors accepted ... 10: 2 errors accepted 11: 1 error accepted 12: no error accepted
STAB	h0006[5]	RW		0		Horizontal sync watchdog 0: H-PLL is not locked 1: H-PLL is locked (Written to memory by ACQ-interface)
VDOK	h0006[4]	RW		0		Vertical sync watchdog 0: There was no vertical sync during stable horizontal synchronization. 1: There was at least one vertical sync during stable horizontal synchronozation (Written to memory by ACQ-interface)
FIELD	h0006[3]	RW		0		Field detector. 0: Actual field is 1 1: Actual field is 2. (Written to memory by ACQ-interface)
NOISE(1)	h0006[2]	RW		0		Noise and co-channel detector of slicer 1. 00: No noise and no co-channel distortion has been detected. 01: No noise but co-channel-distortion has been detected. 10: Noise but no co-channel-distortion has been detected. 11: Strong noise has been detected. (Written to memory by ACQ-interface)
GRDON	h0006[1]	RW		0		Group delay detector 0: No group delay distortion detected 1: Group delay distortion detected (Written to memory by ACQ_interface)

Table 3–8: ACQ register description, continued

Name	Addr	Dir	Sync	Reset	Range	Function
GRDSIGN	h0006[0]	RW		0		Group delay detector. 0: If group delay distortion has been detected, it was positive. 1: If group delay distortion has been detected, it was negative. (Written to memory by ACQ-interface)
ACQFP7	**h0007**	**RW**	**VS**	**h0000**		
LEOFLI[11:8]	h0007[7:4]	RW	VS	0	0..7	Detection threshold for negative group delay measurement 0000: a small negative group delay activates detection ... 0111: strong negative group delay is needed to activate detection
ACQFP8	**h0008**	**RW**	**VS**	**h0000**		
LEOFLI[7:0]	h0007[3:0]	RW	VS	0	0..7	Detection threshold for positive group delay measurement 0000: a small positive group delay activates detection ... 0111: strong positive group delay is needed to activate detection
LINE_PARAMETER						
ACQLP0	**h000D**	**RW**	**HS**	**h0000**		
DINCR[15:8]	h000D[7:0]	RW	HS	0	0..255	Data PLL Frequency Select Specifies the operating frequency of the D-PLL of the data slicer. DINCR = fdata * 2**18 / 40.5 MHz
ACQLP1	**h000E**	**RW**	**HS**	**h0000**		
DINCR[7:0]	h000E[7:0]	RW	HS	0	0..255	Data PLL Frequency Select (Low Byte) (refer to ACQLP0)
ACQLP2	**h000F**	**RW**	**HS**	**h0000**		
NORM[2:0]	h000F[7:5]	RW	HS	0		Most timing signals are closely related to the actual data service used. Therefore 3 bits are reserved to specify the timing for the service used in the actual line. (corresponds to slicer 1) **NORMService** 000TXT 001reserved 010VPS 011WSS 100CC 101reserved 110reserved 111no data service
FCSEL[1:0]	h000F[4:3]	RW	HS	0		There are three different framing codes which can be used for each field. The framing code used for the actual line is selected with FCSEL (corresponds to slicer 1). **FCSELFC** 00FC1 01FC2 10FC3 11No FC-check

Table 3–8: ACQ register description, continued

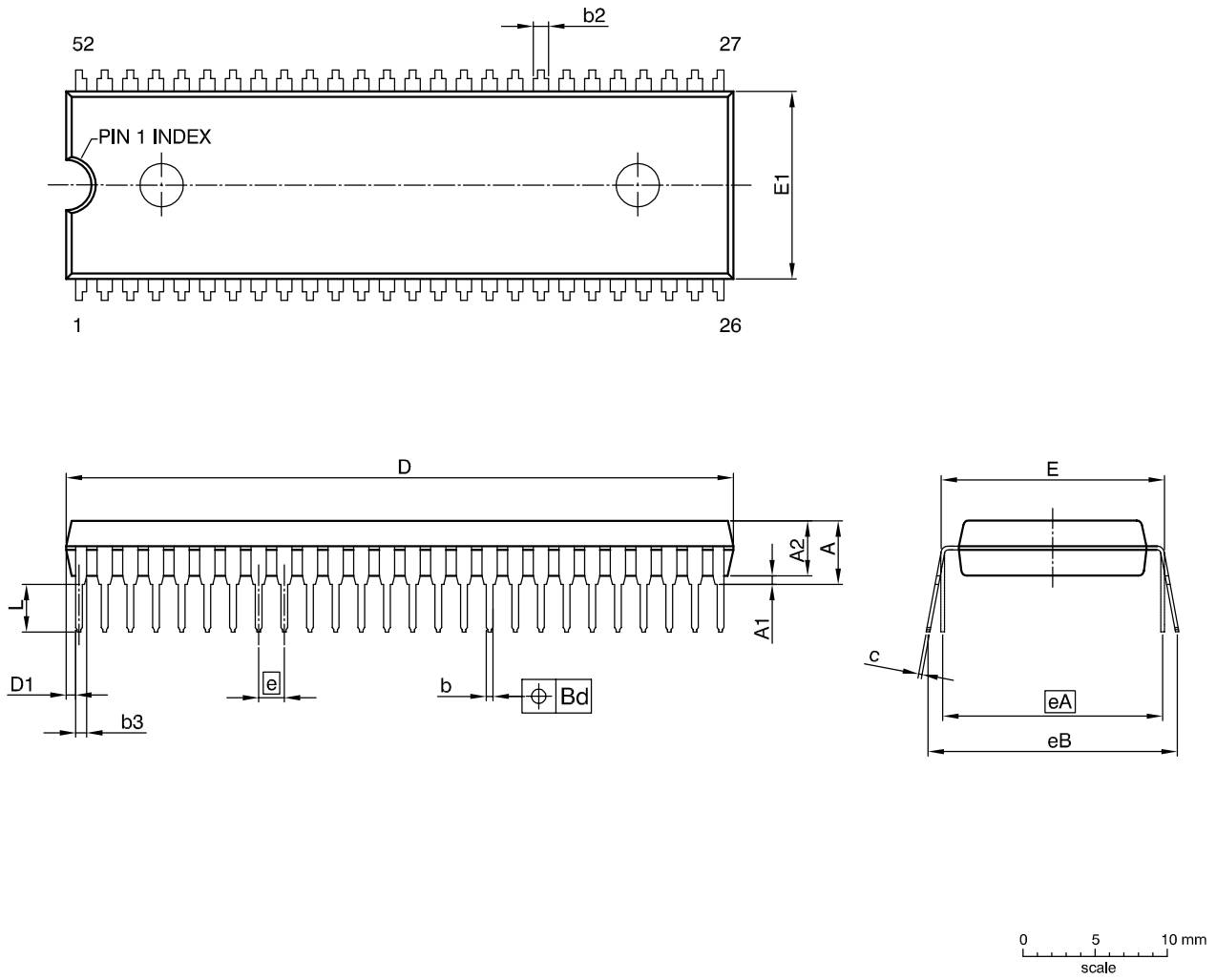
Name	Addr	Dir	Sync	Reset	Range	Function
FC1ER	h000F[2]	RW	HS	0		If this bit is '1' the FC1 check is performed with one bit error tolerance. 0:No error tolerance for FC1-check 1:One bit error tolerance for FC1-check
VCR	h000F[1]	RW	HS	0		This bit is used to change the behavior of the D-PLL. (corresponds to slicer 1) 0:D-PLL tuning is stopped after CRI. 1:D-PLL is tuned throughout the line.
ACQLP3	**h0010**	**RW**	**HS**	**h0000**		
MLENGTH[7:5]	h0010[7:5]	RW	HS	0		For noise suppression reasons a median filter has been introduced after the actual data separation. Because of over sampling successive samples could be averaged. Therefore an odd number of sliced successive samples is taken and if the majority are '1' a '1' is sliced otherwise a '0'. MLENGTH specifies how many samples are taken. (Corresponds to slicer 1) MLENGTHNumber of samples 0001 0013 0105 0117 1009 10111 11013 11115
ALENGTH[4:3]	h0010[4:3]	RW	HS	0		If noise has been detected or if NOISEON = 1 the output of the slicing level filter is further averaged by means of an accumulation (arithmetic averaging). ALENGTH specifies the number of slicing level filter output values used for averaging. The accumulation clock depends on CLKDIV. **ALENGTHNumber of Slicing Level Output Values used for Averaging** 002 014 108 1116
CLKDIV[2:0]	h0010[2:0]	RW	HS	0		The slicing level filter needs to find the DC value of the CVBS during CRI. In order to do this it should suppress at least the CRI frequency. As different services use different data frequencies the CRI frequency will be different as well. Therefore the filter characteristic needs to be shifted. This can be done by using different clocks for the filter. The filter itself shows sufficient suppression for frequencies between $0.0757 \times SL_{CLK}$ and $0.13 \times SL_{CLK}$ (SL_{CLK} is the actual filter clock and corresponds to slicer 1) **CLKDIVSL_{CLK}** $0001 \times f_s$ $0011/2 \times f_s$ $0101/3 \times f_s$ $0111/4 \times f_s$ $1001/5 \times f_s$ $1011/6 \times f_s$ $1101/7 \times f_s$ $1111/8 \times f_s$ **Note: f_s = 33.33 MHz**

Table 3–8: ACQ register description, continued

Name	Addr	Dir	Sync	Reset	Range	Function
ACQLP4	**h0011**	**RW**	**HS**	**h0000**		
PERR[7:2]	h0011[7:2]	RW	HS	0		Phase Error Watch Dog (detection of test line CCIR331a or b) The value shows how often in a line the internal PLL found strong phase deviations between PLL and sliced data. The value can be used to detect test line CCIR331a or b. PERRP < 32? No test line. PERRP > 31? Test line CCIR331a or b detected.
TLDE	h0011[1]	RW	HS	0		Test Line Detected (CCIR17 or CCIR18 or CCIR330) 0: No test line of the above mentioned test lines has been detected. 1: The following data has most likely be sliced from a test line and should therefore be ignored.
FCOK	h0011[0]	RW	HS	0		Framing Code Received 0:No framing code has been detected (no new data has been written to memory). 1:The selected framing code has been detected (new data has been written to memory.

4. Specifications

4.1. Outline Dimensions for PSDIP52-1 Package

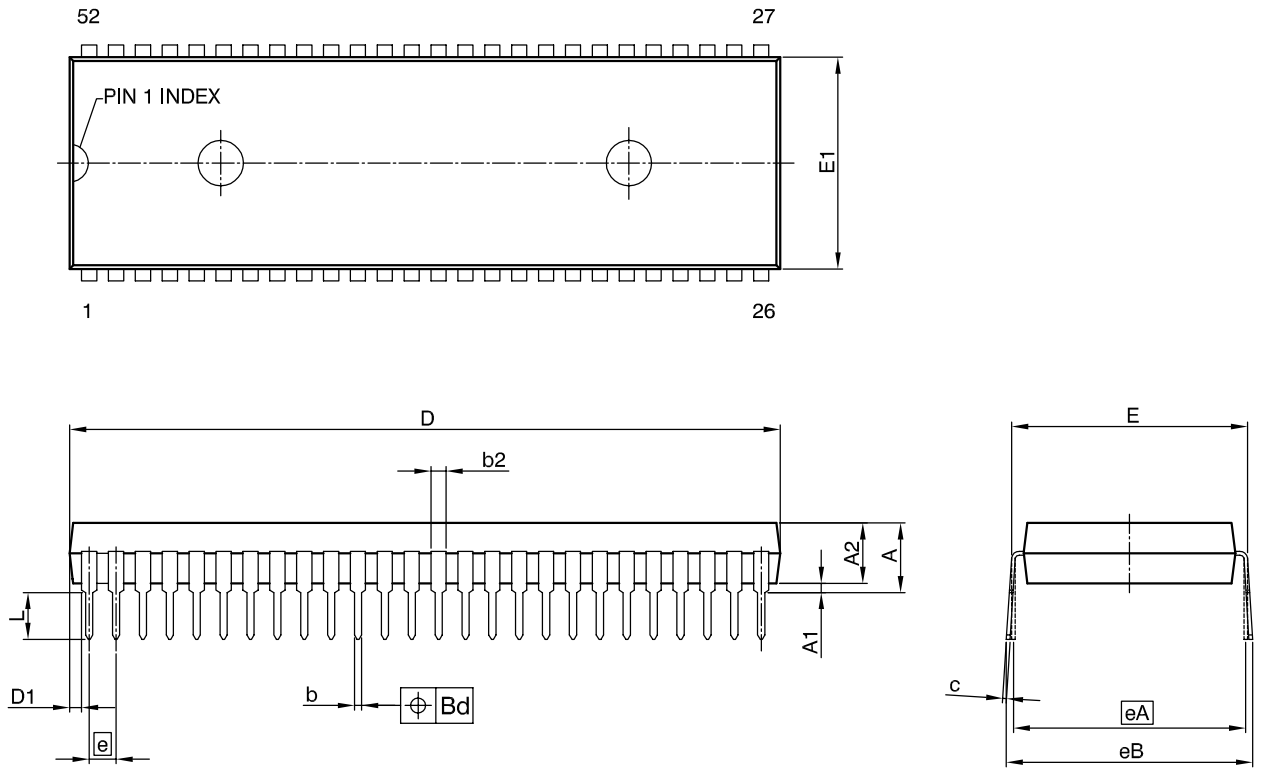


UNIT	A max.	A1 min.	A2 max.	b	b2	b3	Bd	c	D	D1	E	E1	e	eA	eB	L
mm	4.83	0.5	4.33	0.588 0.428	1.55 1.05	1.279 0.779	0.125	0.30 0.20	46.20 45.70	0.676 0.596	15.54 15.44	13.25 12.75	1.778	15.24	17.30 15.30	3.00 min.

JEDEC STANDARD		ANSI	ISSUE DATE YY-MM-DD	DRAWING-NO.	SPZG-NO.
ISSUE	ITEM NO.				
B	MS-020		03-05-15	06586.0001.4	SPZG001000_001_02

Fig. 4–1:
PSDIP52-1: Plastic **S**hrink **D**ual **I**n-line **P**ackage, **52** leads, 600 mil
Ordering code: PO
Weight approximately 5.13 g

4.2. Outline Dimensions for PSDIP52-2 Package

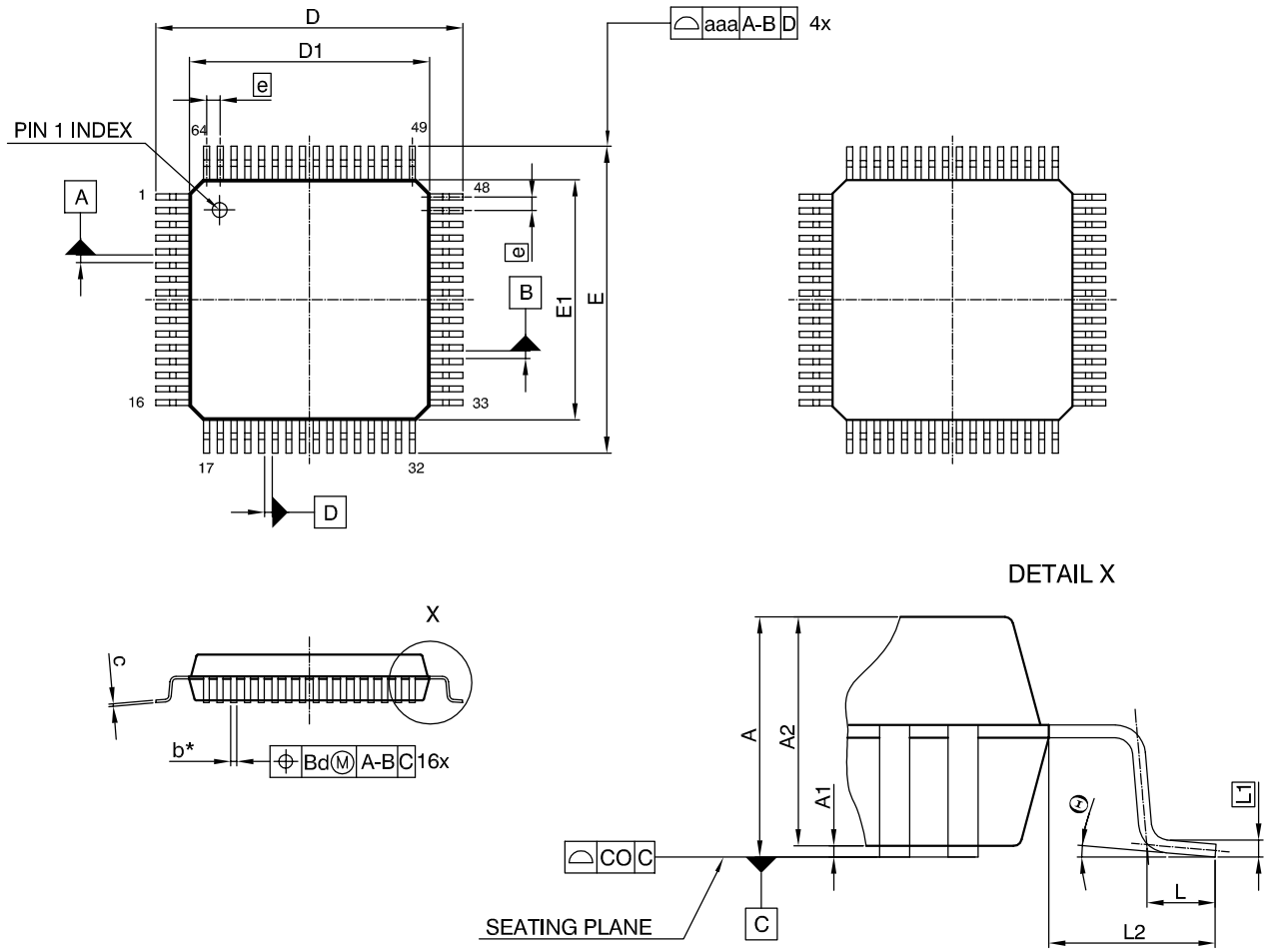


UNIT	A max.	A1 min.	A2 max.	b	b2	Bd	c	D	D1	E	E1	e	eA	eB	L
mm	4.90	0.40	4.10	0.54 0.42	1.05 0.95	0.125	0.34 0.22	47.10 46.90	0.90 0.70	15.70 15.50	14.10 13.90	1.778	15.24	17.30 15.30	3.30 2.90

JEDEC STANDARD		ANSI	ISSUE DATE YY-MM-DD	DRAWING-NO.	SPZG-NO.
ISSUE	ITEM NO.				
B	MS-020		03-05-15	06604.0001.4	SPZG001005_001_02

Fig. 4–2:
PSDIP52-2: Plastic **S**hrink **D**ual **I**n-line **P**ackage, **52** leads, 600 mil
Ordering code: PO
Weight approximately 5.92 g

4.3. Outline Dimensions for PMQFP64-1 Package



UNIT	L2	Θ
mm	1.60	0°-7°

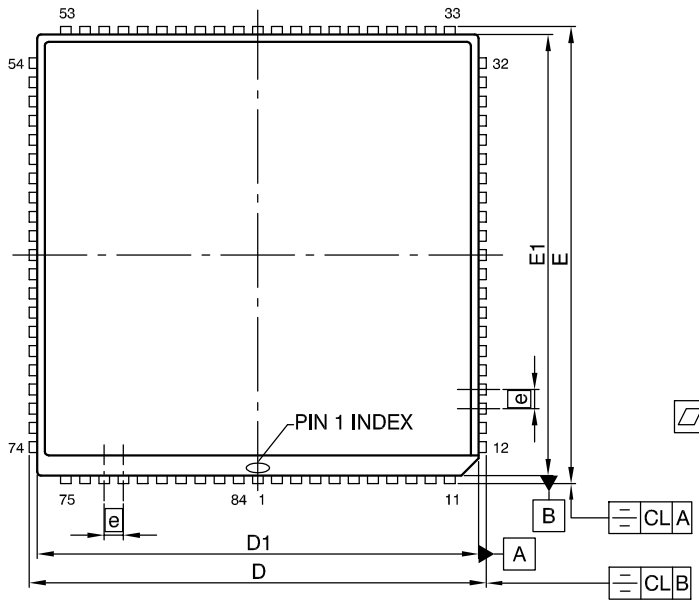
* does not include dambar protrusion of 0.08 max. per side

UNIT	A	A1	A2	aaa	b	Bd	c	CO	D	D1	E	E1	e	L	L1
mm	2.45 2.15	0.25 0.00	2.20 1.80	0.25	0.45 0.29	0.2	0.23 0.11	0.1	17.2	14.0	17.2	14.0	0.8	1.03 0.73	0.25

JEDEC STANDARD		ANSI	ISSUE DATE YY-MM-DD	DRAWING-NO.	SPZG-NO.
ISSUE	ITEM NO.				
B	MS-022		03-01-17	06588.0001.4	SPZG001002_001_02

Fig. 4–3:
PMQFP64-1: Plastic **M**etric **Q**uad **F**lat **P**ackage, **64** leads, 14 × 14 × 2 mm^3
Ordering code: BS
Weight approximately 0.95 g

4.4. Outline Dimensions for PLCC84-1 Package



UNIT	e	L1
mm	1.27	0.5 min

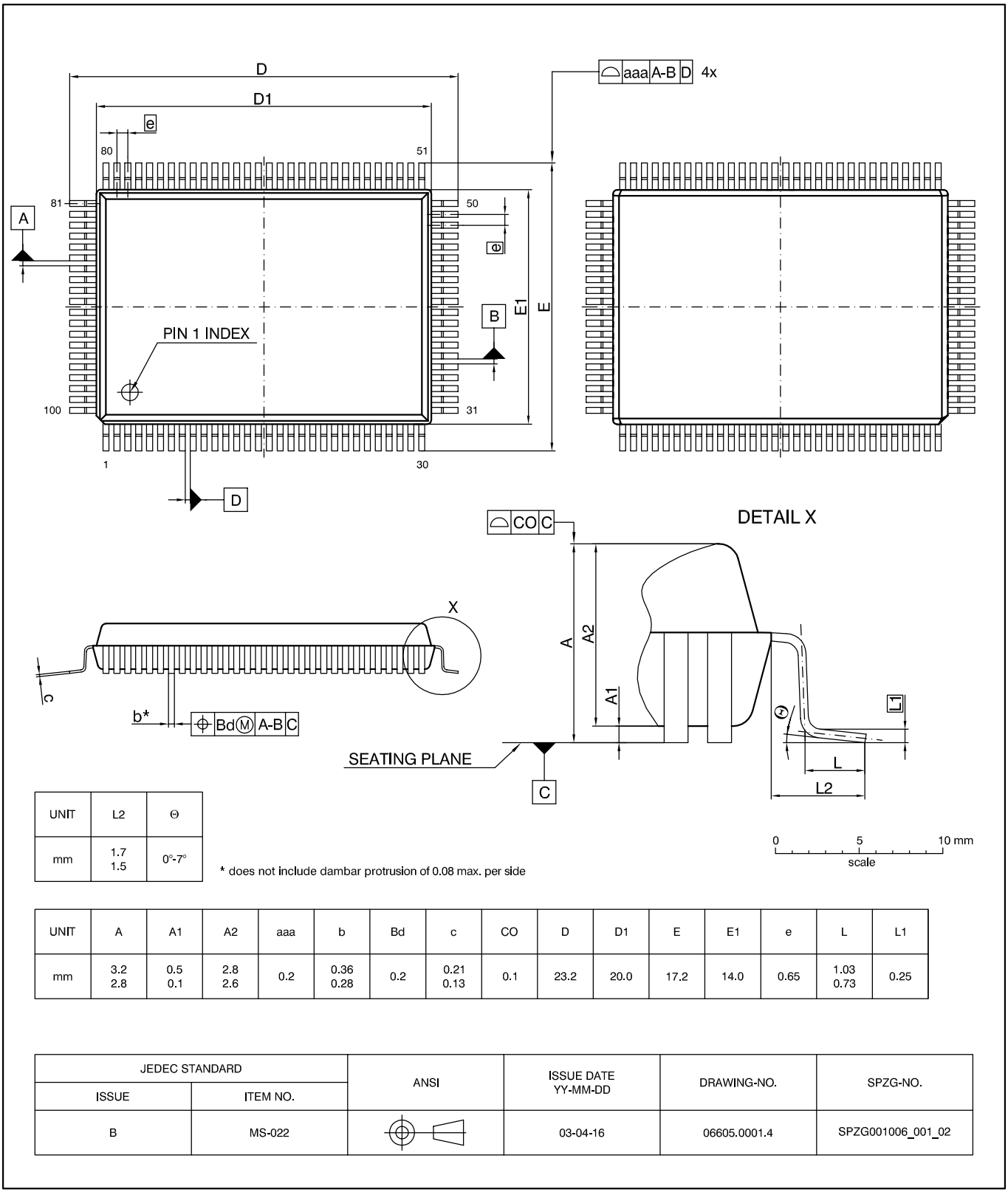
0 5 10 mm
scale

UNIT	A	A1	A3	b	b2	Bd	c	CL	CO	D	D1	D3	E	E1	E3
mm	4.58	0.5	2.64	0.53 0.33	0.812 0.661	0.1	0.20	0.125	0.1	30.36 30.10	29.386 29.234	28.7 27.7	30.36 30.10	29.386 29.234	28.7 27.7

JEDEC STANDARD		ANSI	ISSUE DATE YY-MM-DD	DRAWING-NO.	SPZG-NO.
ISSUE	ITEM NO.				
B	MO-047		03-01-17	06587.0001.4	SPZG001001_001_01

Fig. 4–4:
PLCC84-1: Plastic **L**eaded **C**hip **C**arrier, **84** leads, 29.4 × 29.4 × 3.8 mm^3
Ordering code: WA
Weight approximately 6.72 g

4.5. Outline Dimensions for PMQFP100-1 Package



UNIT	L2	Θ
mm	1.7 1.5	0°-7°

* does not include dambar protrusion of 0.08 max. per side

UNIT	A	A1	A2	aaa	b	Bd	c	CO	D	D1	E	E1	e	L	L1
mm	3.2 2.8	0.5 0.1	2.8 2.6	0.2	0.36 0.28	0.2	0.21 0.13	0.1	23.2	20.0	17.2	14.0	0.65	1.03 0.73	0.25

JEDEC STANDARD		ANSI	ISSUE DATE YY-MM-DD	DRAWING-NO.	SPZG-NO.
ISSUE	ITEM NO.				
B	MS-022		03-04-16	06605.0001.4	SPZG001006_001_02

Fig. 4–5:
PMQFP100-1: **P**lastic **M**etric **Q**uad **F**lat **P**ackage, **100** leads, 14 × 20 × 2.7 mm^3
Ordering code: QB
Weight approximately 1.7 g

4.6. Pin Connections and Short Descriptions

NC = not connected, leave vacant
LV = if not used, leave vacant

Pin No.				Pin Name	Type	Connection (If not used)	Short Description
PMQFP 100-1	PMQFP 64-1	PSDIP 52-1 52-2	PLCC 84-1				
1	–	–	9	D1	I/O		Data bus for external memory or data RAM
2	–	–	10	D4	I/O		Data bus for external memory or data RAM
3	–	–	11	D2	I/O		Data bus for external memory or data RAM
4	–	–	12	D3	I/O		Data bus for external memory or data RAM
5	–	–	13	$\overline{\text{XROM}}$	I/O		This pin must be pulled low to access external ROM
6	3	9	14	VDD 2.5	PS		Supply voltage (2.5 V)
7	4	10	15	VSS2.5	PS		Ground (0 V)
8	5	11	16	VDD 3.3	PS		Input/Output (3.3 V)
9	57	1	17	P0.0	PS		Port 0 is a b-bit open drain bidirectional I/O port. Port 0 pins that have “1” written to them float. In this stage, they can be used as high impedance inputs (e.g. for software driven I²C Bus support).
10	58	2	18	P0.1	I/O		
11	59	3	19	P0.2	I/O		
12	60	4	20	P0.3	I/O		
13	61	5	21	P0.4	I/O		
14	62	6	22	P0.5	I/O		
15	64	7	23	P0.6	I/O		
16	2	8	24	P0.7	I/O		
17	–	–	–	$\overline{\text{ENE}}$	I		Enable Emulation. Only if this pin is set to zero externally, $\overline{\text{STOP}}$ and $\overline{\text{OCF}}$ are operational. $\overline{\text{ENE}}$ has an internal pull-up resistor which switches auotomatically to non-emulation mode if $\overline{\text{ENE}}$ is not connected.
18	–	–	–	$\overline{\text{STOP}}$	I		Emulation control line. Driving a low level during the input phase freezes the real time relevant internal peripherals such as timers and interrupt controller.

Pin No. PMQFP 100-1	Pin No. PMQFP 64-1	Pin No. PSDIP 52-1 52-2	Pin No. PLCC 84-1	Pin Name	Type	Connection (If not used)	Short Description
19	–	–	–	$\overline{OCF}$	I		Opcode Fetch. Emulation control line. A high level driven by the controller during output phase indicates the beginning of a new instruction.
20	–	–	–	$\overline{EXTIF}$	I/O		This pin must be pulled low to enable extended memory interface.
21	8	12	25	CVBS	I/O		CVBS input for the acquisition circuit
22	9	13	26	VDDA 2.5	PS		Supply voltage for analog components.
23	10	14	27	VSSA	PS		Ground for analog components
24	12	15	28	P2.0	I/O		Port 2 is a 4-bit port without pull-up resistors. --- Port 2 has an alternate function. **See Section on page 155**.
25	13	16	29	P2.1	I/O		
26	14	17	30	P2.2	I/O		
27	15	18	31	P2.3	I/O		
28	–	–	–	–	–	NC	Not connected
29	16	19	32	HS/SSC	I/O		In slave mode horizontal sync input or sandcastle input for display synchronization. --- In master mode HS or VCS output
30	17	20	33	VS	I/O		Vertical sync input/ouput for display synchronization. --- It also can be used as digital input P4.7. **See Section on page 155.** --- Furthermore, this pin can be selected as an ODD/EVEN indicator alternatively to P3.0. **See Section on page 155.**

Pin No. PMQFP 100-1	Pin No. PMQFP 64-1	Pin No. PSDIP 52-1 52-2	Pin No. PLCC 84-1	Pin Name	Type	Connection (If not used)	Short Description
31	18	21	34	P3.0	I/O		Port 3 is an 8-bit bidirectional I/O port with internal pull-up resistors. Port 3 pins that have “1” written to them are pulled high by the internal pull-up resistros.In that state the pins can be used as inputs. ---------- To use the alternated functions of Port 3, the corresponding output latch must be programmed to a “1” for that function to operate. **See Section on page 155.**
32	19	22	35	P3.1	I/O		
33	20	23	36	P3.2	I/ O		
34	21	24	37	P3.3	I/ O		
35	22	25	38	P3.4	I/ O		
36	23	26	39	P3.5	I/ O		
37	26	27	40	P3.6	I/ O		
38	27	28	41	P3.7	I/ O		
39	28	29	42	VSS	PS		Ground (0 V)
40	29	30	43	VDD 3.3	PS		Input/Output (3.3 V)
41	47	45	44	P1.0	I/ O		Port 1 is a 8-bit bidirectional multifunction I/O port with internal pull-up resistors. Port 1 pins that have “1” written to them are pulled high by the internal pull-up resistors and in that state can be used as inputs. ---------- Port 1 pins have a alternated function. **See Section on page 155.**
42	49	46	45	P1.1	I/ O		
43	51	47	46	P1.2	I/ O		
44	52	48	47	P1.3	I/ O		
45	53	49	48	P1.4	I/O		
46	54	50	49	P1.5	I/O		
47	55	51	50	P1.6	I/O		
48	30	31	51	P4.2	I/O		Port 4 is a bidirectional I/O port with internal pull-up resistors. Port 4 pins that have 1 written to them are pulled high by the internal pull-up resistors. In that state, the pins can be used as inputs. ---------- Port 4 pins have a alternated function. **See Section on page 155.**
49	31	32	52	P4.3	I/O		
50	32	33	53	$\overline{\text{RST}}$	I/O		A low level on this pin resets the device. An internal pull-up resistor permits power-on reset using only one external capacitor connected to V_{SS}.
51						NC	Not connected.

Pin No. PMQFP 100-1	Pin No. PMQFP 64-1	Pin No. PSDIP 52-1 52-2	Pin No. PLCC 84-1	Pin Name	Type	Connection (If not used)	Short Description
52	34	34	54	XTAL2	I		Input of the inverting oscillator amplifier.
53	35	35	55	XTAL1	O		Output of the inverting oscillator amplifier
54						NC	Not connected.
55	36	14	56	VSSA	PS		Ground for analog components
56	37	13	57	VDDA 2.5	PS		Supply voltage for analog components.
57	38	38	58	R	O		Red
58	39	39	59	G	O		Green
59	40	40	60	B	O		Blue
60	42	41	61	BLANK/COR	O		Blanking and contrast reduction.
61						NC	Not connected.
62	56	52	62	P1.7	I/O		Port 1 is a 8-bit bidirectional multifunction I/O port with internal pull-up resistors. Port 1 pins that have “1” written to them are pulled high by the internal pull-up resistors and in that state can be used as inputs. --- Port 1.7 has an alternated function. **See Section on page 155.**
63						NC	Not connected.
64				$\overline{WR}$	O		Control output. Indicates a write access to the internal XRAM. --- It can be used as a write strobe for writing data into an external data RAM by a MOVX instruction. --- This signal is also available as P4.3 **See Section on page 155.**

Pin No. PMQFP 100-1	Pin No. PMQFP 64-1	Pin No. PSDIP 52-1 52-2	Pin No. PLCC 84-1	Pin Name	Type	Connection (If not used)	Short Description
65				$\overline{RD}$	O		Control output. Indicates a read access to the internal XRAM. --- It can be used for latching data from the data bus into an external data RAM by a MOVX instruction. --- This signal is also available as P4.2 **See Section on page 155.**
66						NC	Not connected.
67			63	A19	I/O		After power-on Port P4.0, P4.1 and P4.4 work as additional address lines A17 ... A19. **See Section on page 155**
68			64	A18	I/O		
69			65	A17	I/O		
70			66	A16	O		Address bus for external program memory or data RAM.
71			67	A15	O		
72				FL_PGM	I		All the pins prefix by FL_ are test pins that must be left open.
73	44	42	68	VDD 2.5	PS		Supply voltage (2.5 V)
74	45	43	69	VSS	PS		Ground (0 V)
75	46	44	70	VDD 3.3	PS		Input/Output (3.3 V)
76			71	A14	O		Address bus for external program memory or data RAM:
77			72	A12	O		
78			73	A13	O		
79			74	A7	O		
80				FL_RST	I		All the pins prefix by FL_ are test pins that must be left open.
81			75	A8	O		Address bus for external program memory or data RAM.
82			76	A6	O		
83			77	A9	O		
84			78	A5	O		
85			79	A11	O		
86			80	A4	O		

Pin No.				Pin Name	Type	Connection (If not used)	Short Description
PMQFP 100-1	PMQFP 64-1	PSDIP 52-1 52-2	PLCC 84-1				
87				ALE	O		Address Latch Enable
88			81	$\overline{\text{PSEN}}$	O		Program Store Enable. It is a control output signal, which is usually connected to OE input line of the external program memory to enable the data output.
89			82	A3	O		Address bus for external program memory or data RAM.
90			83	A10	O		
91			84	VSS	PS		Ground (0 V)
92			1	VDD 3.3	PS		Input/Output (3.3 V)
93			2	A2	O		Address bus for external program memory or data RAM.
94			3	A1	O		
95				FL_CE	I		All the pins prefix by FL_ are test pins that must be left open.
96			4	D7	I/O		Data bus for external memory or data RAM.
97			5	A0	O		Address bus for external program memory or data RAM.
98			6	D6	I/O		Data bus for external memory or data RAM.
99			7	D0	I/O		
100			8	D5	I/O		

4.7. Port Alternate Functions

Pin No. PMQFP 100-1	Pin No. PMQFP 64-1	Pin No. PSDIP 52-1 52-2	Pin No. PLCC 84-1	Pin Name	Type	Connection (If not used)	Short Description
24	12	15	28	CADCCO(AD0)	I		Alternate function of P2.0: ADC.
25	13	16	29	CADCCO(AD1)	I		Alternate function of P2.1: ADC.
26	14	17	30	CADCCO(AD2)	I		Alternate function of P2.2: ADC.
27	15	18	31	CADCCO(AD3)	I		Alternate function of P2.3: ADC.
31	18	21	34	CSCR0(O_E_$\overline{P3_0}$)	I/O		Alternate function of P3.0: ODD/EVEN indicator
32	19	22	35	PORT INPUT MODE	I/O		Alternate function of P3.1: External extra interrupt 0 ($\overline{INTX0}$)/UART(TXD)
				PORT OUTPUT MODE	I/O		Alternate funtion of P3.1: TXD
33	20	23	36	PORT INPUT MODE	I/O		Alternate function of P3.2: Interrupt 0 input/timer 0 gate control input ($\overline{INT0}$)
34	21	24	37	PORT INPUT MODE	I/O		Alternate function of P3.3: Interrupt 1 input/timer 1 gate control input ($\overline{INT1}$)
35	22	25	38	PORT INPUT MODE	I		Alternate function of P3.4: Counter 0 input (T0)
36	23	26	39	PORT INPUT MODE	I/O		Alternate function of P3.5: Counter 1 input (T1) or in master mode HS or VCS output
37	27	28	41	PORT INPUT MODE	O		Alternate function of P3.7: External extra interrupt 1 ($\overline{INTX1}$)/UART(RXD)
38	47	45	44	PWME(E0)	I/O		Alternate function of P1.0: Output 8-bit pulse PWM channel 0
39	49	46	45	PWME(E1)	I/O		Alternate function of P1.1: Output 8-bit pulse PWM channel 1
40	51	47	46	PWME(E2)	I/O		Alternate function of P1.2: Output 8-bit pulse PWM channel 2

Pin No.				Pin Name	Type	Connection (If not used)	Short Description
PMQFP 100-1	PMQFP 64-1	PSDIP 52-1 52-2	PLCC 84-1				
41	52	48	47	PWME(E3)	I/O		Alternate function of P1.3: Output 8-bit pulse PWM channel 3
42	53	49	48	PWME(E4	I/O		Alternate function of P1.4: Output 8-bit pulse PWM channel 4
43	54	50	49	PWME(E5)	I/O		Alternate function of P1.5: Output 8-bit pulse PWM channel 5
44	55	51	50	PWME(E6)	I/O		Alternate function of P1.6: Output 14-bit pulse PWM channel 0
45	56	52	62	PWME(E7)	I/O		Alternate function of P1.7: Output 14-bit pulse PWM channel 1
48	30	31	51	CSCR1(ENARW)	I/O		Alternate function of P4.2: Read signal
49	31	32	52	CSCR1(ENARW)	I/O		Alternate function of P4.3: Write signal
62	56	52	62	CSCR0(VS_$\overline{OE}$, $\overline{P1_7}$_ALT)	O		Alternate function of P1.7: VS output
				CSCR0(VS_$\overline{OE}$, $\overline{P1_7}$_ALT)	O		Alternate function of P1.7: OddEven output
67				CSCR1($\overline{A19}$_P4_1)			Alternate function of P4.4: Port pin
68			64	CSCR1($\overline{A18}$_P4_1)	I/O		Alternate function of P4.1: Port pin
69			65	CSCR1($\overline{A17}$_P4_0)	I/O		Alternate function of P4.0: Port pin

4.8. Pin Descriptions

Pin numbers refer to the PMQFP100-1 package.

Pin 1, 2, 3, 4, **D0, D1, D2, D3** – Data bus for external memory or data RAM.

Pin 5, **$\overline{\text{XROM}}$** – This pin must be pulled low to access external ROM.

Pin 6, 73, **VDD 2.5** – Supply voltage (2.5 V).

Pin 7, 39, 74, 91, **VSS** – Ground (0 V).

Pin 8, 40, 75, 92, **VDD 3.3** – Input/Output (3.3 V).

Pin 9, 10, 11, 12, 13, 14, 15, 16, **P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7** – Port 0 is a 8-bit open drain bidirectional I/O-port. Port 0 pins that have “1” written to them float; in this state they can be used as high impedance inputs (e.g. for software driven I²C Bus support).

Pin 17, **$\overline{\text{ENE}}$** – Enable Emulation. Only if this pin is set to zero externally, $\overline{\text{STOP}}$ and $\overline{\text{OCF}}$ are operational. $\overline{\text{ENE}}$ has an internal pull-up resistor which switches automatically to non-emulation mode if $\overline{\text{ENE}}$ is not connected.

Pin 18, **$\overline{\text{STOP}}$** – Stop. Emulation control line. Driving a low level during the input phase freezes the real time relevant internal peripherals such as timers and interupt controller.

Pin 19, **$\overline{\text{OCF}}$** – Opcode Fetch. Emulation control line. A high level driven by the controller during output phase indicates the beginning of a new instruction.

Pin 20, **$\overline{\text{EXTIF}}$** – This pin must be pulled low to enable extended memory interface.

Pin 21, **CVBS** – CVBS input for the acquisition circuit.

Pin 22, 56, **VDDA 2.5** – Supply voltage for analog components.

Pin 23, 55, **VSSA** – Ground for analog components.

Pin 24, , 25, 26, 27, **P2.0, P2.1, P2.2, P2.3** – Port 2 is a 4-bit input port without puul-up resistors. Port 2 also works as analog input for the 4-channel-ADC. **See Section on page 155.**

Pin 28, **NC** – Pin not connected.

Pin 29, **HS/SSC** – In slave mode horizontal sync input or sandcastle input for display synchronisation. In master mode HS or VCS output.

Pin 30, **VS/P4.7** – Vertical sync input/output for diaplay synchronisation. Can also be used as digital input P4.7 Furthermore this pin can be selected as an ODD/EVEN indicator alternatively to P3.0. **See Section on page 155.**

Pin 31, 32, 33, 34, 35, 36, 37 ,38, **P3.0, P3.1, P3.2, P3.3, P3.4, P3.5, P3.6, P3.7** – Port 3 is an 8-bit bidirectional I/O-port with internal pull-up resistors. Port 3 pins that have “1” written to them are pulled high by the internal pull-up resistors and in that state can be used as inputs. To use the secondary functions of Port 3, the corresponding output latch must be programmed to a “1” for that function to operate.
The secondary functions are as follows:
P3.0: ODD/EVEN indicates output.
P3.1: External extra interrupt 0 ($\overline{\text{INTX0}}$)/UART(TXD)
P3.2: Interrupt 0 input/timer 0 gate control input ($\overline{\text{INT0}}$)
P3.3: Interrupt 1 input/timer 1 gate control input (INT1)
P3.4: Counter 0 input (T0)
P3.5: Counter 1 input (T1) or in master mode HS or VCS output
P3.7: External extra interrupt 1 ($\overline{\text{INTX1}}$)/UART(RXD)
Note: P3.6 mustnot be kept to “0” during reset, otherwise a testmode will be activated.
See Section on page 155.

Pin 41, 42, 43, 44, 45, 46, 47, 62, **P1.0, P1.1, P1.2, P1.3, P1.4, P1.5, P1.6, P1.7** – Port 1 is a 8-bit bidirectional multifunction I/O-port with internal pull-up resistors. Port 1 pins that have “1” written to them are pulled high by the internal pull-up resistors and in that state can be used as inputs.
The secondary functions of Port 1 pins are:
Port bits P1.0 - P1.5 contain the 6 output channels of the 8-bit pulse width modulation unit.
Port bits P1.6 - P1.7 contain the two output channels of the 14-bit pulse width modulation unit.
See Section on page 155.

Pin 48, 49, **P4.2, P4.3** – Port 4 is a bidirectional I/0-port with internal pull-up resistors. Port 4 pins that have an “1” written to them are pulled high by the internal pull-up resistors and in that state can be used as inputs.
The secondary functions are:
P4.2: $\overline{\text{RD}}$, Read line. This signal is the same as the outcoming signal of pin $\overline{\text{RD}}$ available in some packages.
P4.3: $\overline{\text{WR}}$, write line. Thissignal is the same as the outcoming signal of pin $\overline{\text{WR}}$, which is only available in some packages.
See Section on page 155.

Pin 50, **$\overline{\text{RST}}$** – A low level on this pin resets the device. An internal pull-up resistor permits power-on reset using only one extrnal capacitor connected to VSS.

Pin 51, **NC** – Pin is not conected.

Pin 52, **XTAL2** – Output of the inverting oscillator amplifier.

Pin 53, **XTAL1** – Input of the inverting oscillator amplifier.

Pin 54, **NC** – Pin is not connected.

Pin 57, 58, 59, **R, G, B** – Red, Green, Blue.

Pin 60, **BLANK/COR** – Blanking and contrast reduction.

Pin 61, **NC** – Pin is not connected.

Pin 63, **NC** – Pin is not connected.

Pin 64, **$\overline{\text{WR}}$** – Control output. Indicates a write access to the internal XRAM. Can be used as a write strobe for writing data into an external data RAM by a MOVX instruction. The signal is also available as P4.3.
See Section on page 155.

Pin 65, **$\overline{\text{RD}}$** – Control output. Indicates a read access to the internal XRAM. Can be used for latching data from the data bus into an external data RAM by a MOVX instruction. This signal is also available as P4.2
See Section on page 155

Pin 66, **NC** – Pin is not connected.

Pin 67, 68, 69, **A16, A17, A18, A19, P4.0, P4.1, P4.4** – After power-on P4.0, P4.1, P4.4 work as additional address lines A17 ... A19. In port mode, these port lines act as bidirectional I/O-port with internal pul--up resistors. Port pins that have “1” written to them are pulled high by the internal pull-up resistors and in that stage they can be used as inputs.
See Section on page 155.

Pin 70, 71, 76, 77, 78, 79, 81, 82,83, 84, 85, 86, 89, 90, 93, 94, 97, **A15, A14, A13, A7, A8, A9, A5, A11, A4, A3, A10, A2, A1, A0** – Address bus for external program memory od data RAM.

Pin 72, 80, 95, **FL_PGM, FL_RST, FL_CE** – All the pins prefix by FL_ are test pins, which must be left open.

Pin 87, **ALE** – Address Latche Enable.

Pin 88, **$\overline{\text{PSEN}}$** – Program Store Enable. $\overline{\text{PSEN}}$ is a control output signal which is usually connected to OE input line of the external program memory to enable the data output.

Pin 96, 98, 99, 100, **D7, D6, D0, D5** – Data bus for external memory or data RAM.

4.9. Pin Configurations

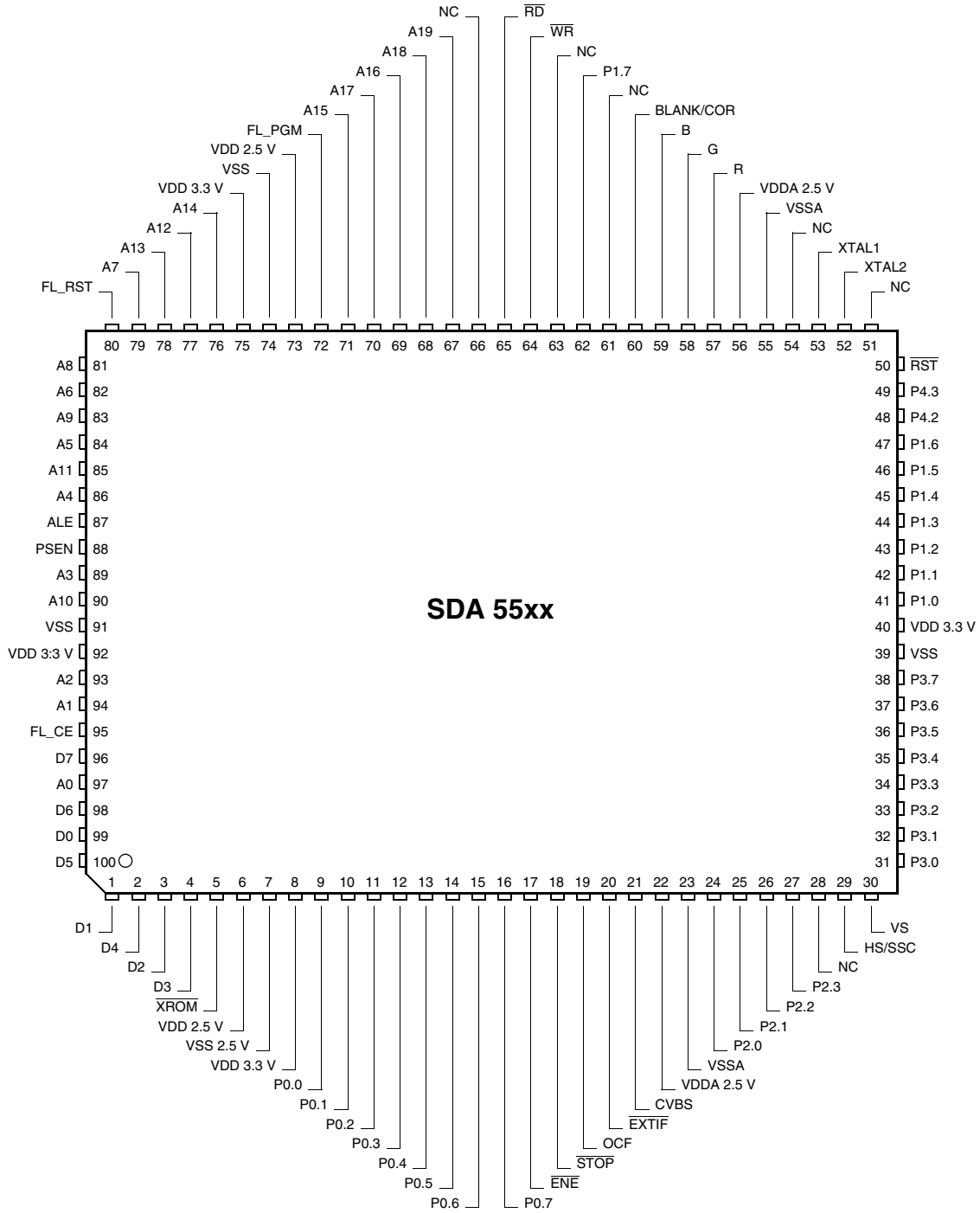


Fig. 4–6: PMQFP100-1 package

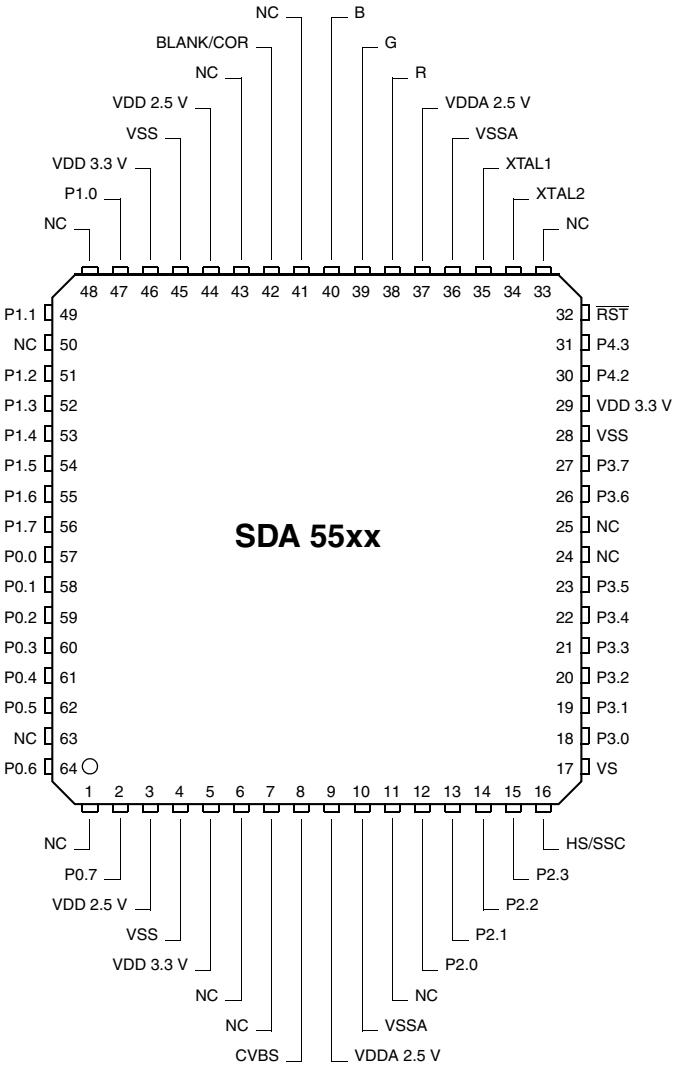


Fig. 4–7: PMQFP64-1 package

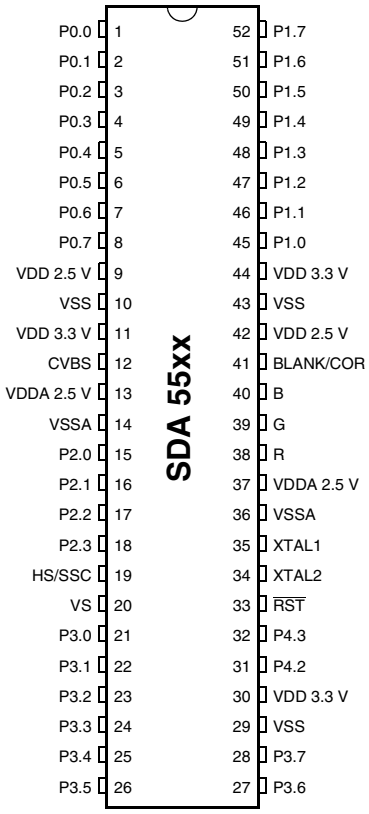


Fig. 4–8: PSDIP52-1 /PSDIP 52-2 package

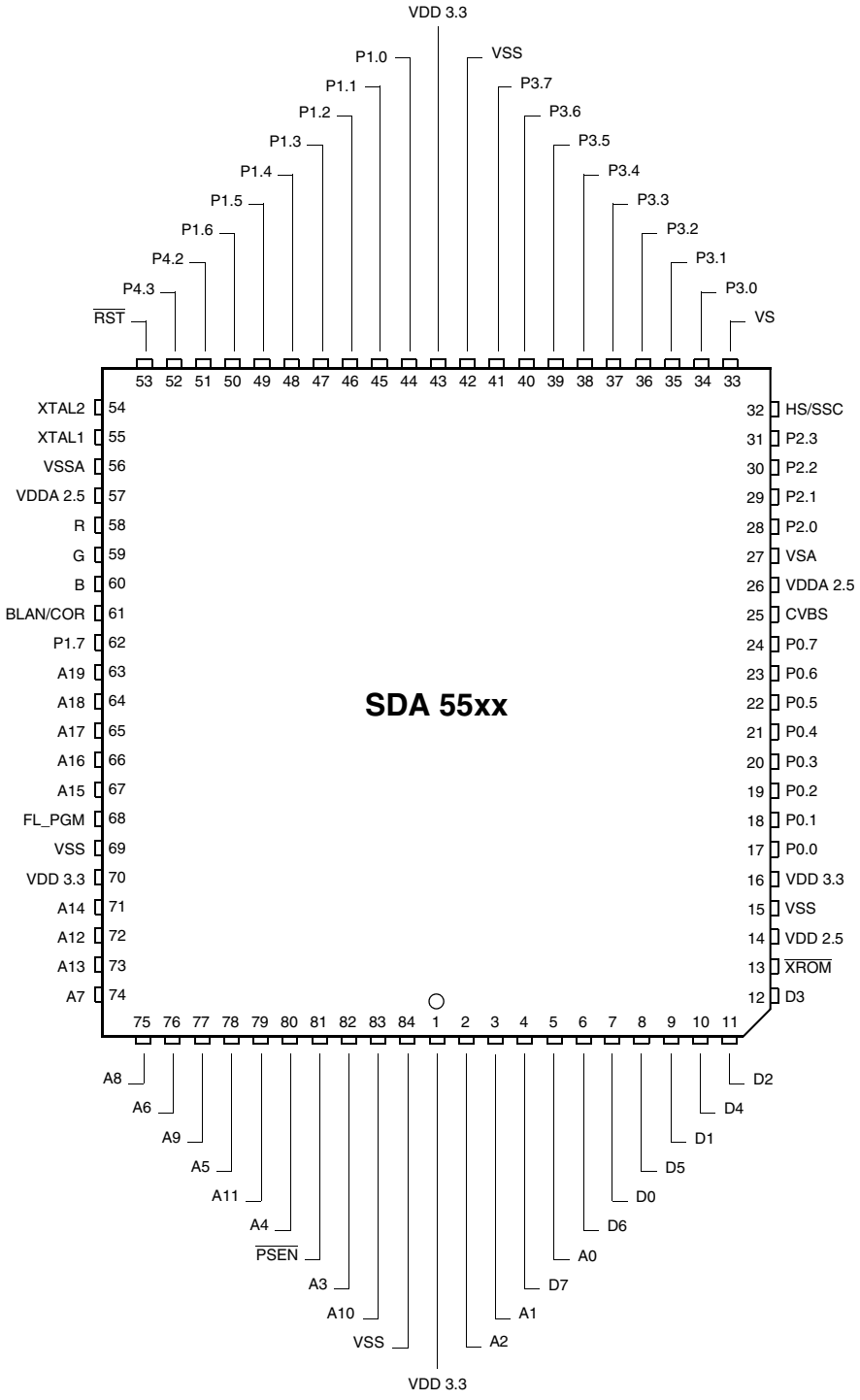


Fig. 4–9: PLCC84-1 package

4.10. Electrical Characteristics

4.10.1. Absolute Maximum Ratings

Stresses beyond those listed in the “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these conditions is not implied. Exposure to absolute maximum ratings conditions for extended periods will affect device reliability.

This device contains circuitry to protect the inputs and outputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than absolute maximum-rated voltages to this high-impedance circuit.

All voltages listed are referenced to ground excepted where noted.
All GND pins must be connected to a low-resistive ground plane close to the IC.

Symbol	Parameter	Pin Name	Limit Values		Unit
			Min	Max	
T_A	Ambient Temperature PSDIP52-1, PSDIP52-2 [1)] PMQFP64-1 PLCC84-1 PMQFP100-1		 –10 –10 –10 –10	 70 70 70 70	 °C °C °C °C
T_C	Case Temperature PSDIP52-1, PSDIP52-2 [1] PMQFP64-1 PLCC84-1 PMQFP100-1		 15 15 15 15	 85 85 85 85	 °C °C °C °C
T_S	Storage Temperature		–20	125	°C
P_{max}	Maximum Power Dissipation PSDIP52-1, PSDIP52-2 [1)] PMQFP64-1 PLCC84-1 PMQFP100-1			 0.6 0.6 0.6 0.6	W
$VDD33_{1..7}$	Supply Voltage 3.3 V		3	3.6	V
$VDD25_{1..2}$	Supply Voltage 2.5 V		2.25	2.75	–
$VDDA_{1..4}$	Analog Supply Voltage		2.25	2.75	–
[1)] Single chip. Not applicable for Flash version (SDA 555xFL)					

4.10.2. Recommended Operating Conditions

Functional operation of the device beyond those indicated in the “Recomended Operating Conditions/Characteristics” is not implied and may result in unpredictable behavior, reduce reliability and lifetime of the device.

All voltages listed are referenced to ground except where noted.
All GND pins must be connected to a low-resistive ground plane close to the IC.

Do not insert the device into a live socket. Instead, apply power by switching on the external power supply.

Symbol	Parameter	Pin No.	Limit Values			Unit
			Min	Typ	Max	
T_A	Ambient Operating Temperature PSDIP52-1, PSDIP52-2 [1)] PMQFP64-1 PLCC84-1 PMQFP100-1		 0 0 0 0		 70 70 70 70	 °C °C °C °C
T_C	Case Operating Temperature PSDIP52-1, PSDIP52-2 [1] PMQFP64-1 PLCC84-1 PMQFP100-1		 15 15 15 15	 40 40 40 40	 85 85 85 85	 °C °C °C °C
P_{max}	Maximum Power Dissipation PSDIP52-1, PSDIP52-2 [1)] PMQFP64-1 PLCC84-1 PMQFP100-1				 0.6 0.6 0.6 0.6	W
$VDD33_{1..7}$	Supply Voltage 3.3 V		3.0	3.3	3.6	V
$VDD25_{1..2}$	Supply Voltage 2.5 V		2.25	2.5	2.75	V
$VDDA_{1..4}$	Analog Supply Voltage		2.25	2.5	2.75	V
V_{IL}	Input Voltage Low	All	–0.4		0.8	V
V_{IH}	Input Voltages High	All	2.0		3.6	V
[1)] Single chip. Not applicable for Flash version (SDA 555xFL)						

4.10.3. Characteristics

Symbol	Parameter	Pin Name	Limit Values			Unit	Test Conditions
			Min.	Typ.	Max.		
Supply							
$I_{3.3\ V}$	Digital Supply Current for 3.3 V Domain		0		1	mA	Digital Pins and BLANK/ COR left open
$I_{2.5\ V}$	Digital Supply Current for 2.5 V Domain		0		30	mA	Min: Power Down Mode Max: Worst Case
I_{ANA}	Analog Power Supply Current		0.2		65	mA	Min: Power Down Mode Max: ADC (20 mA), DAC/ PLL (45 mA) Worse Case
I_{IDLE}	Idle Mode Supply Current (with A/D Wake up, RTC and External Interrupts)		5		10	mA	Max: Digital Core (7 mA) in Idle Mode, PLL (1.5 mA), ADC (1.5 mA)
I_{PD}	Power Down Mode Supply Current		0		1	mA	Max: 1 mA ADC Supply Current
I_{SD}	Slow Down Mode Supply Current		4		8	mA	Max: Digital (5 mA(, PLL (1.5 mA), ADC (1.5 mA)
-	PLL Sleep Mode		-		<4	mA	Digital (< 2 mA), DAC/PLL (< 1 mA), ADC (< 1 ma)
I/O Voltages							
V_{IL}	Input Low Voltage	All	–0.4		0.8	V	-
V_{IH}	Input High Voltage		2.0		3.6	V	-
V_{OL}	Output Low Voltage		-		0.4	V	@ I_{out}= 3.2 mA
V_{OH}	Output High Voltage						@ I_{out}= -1.6 mA during Low-High Transition
I_L	Leakage Current		–1		10	µA	Pins without Pull-up, Input Mode (Port 0, ENE, STOP, VSync)
I_{IL}	Pull-up Low Current		–250		-50	µA	@ V_{ILmax}= 0.8 V
I_{IH}	Pull-up High Current		–170		-25	µA	@ V_{ILmin}= 2.0 V
Crystal Oscillator							
C_{FB}	Crystal Oscillator Frequency	XIN, XOUT	6.0 – 100 ppm		6.0 + 100 ppm	MHz	-

Symbol	Parameter	Pin Name	Limit Values			Unit	Test Conditions
			Min.	Typ.	Max.		
CVBS-Input							
C_P	Pin Capacitance	CVBS	-		-	pF	-
Z_P	Input Impedance		-		-	1/MΩ	-
C_{CPL1}	External Coupling Capacitance		10		100	nF	-
R_Z	Source Impedance		-		< 500	W	-
V_{CVBS}	Overall CVBS Amplitude		0.75		1.3	V	-
V_{SYNC}	CVBS Sync Amplitude		0.18		0.6	V	-
V_{DATA}	TXT Data Amplitude		0.3		0.7	V	-
RGB-Outputs							
C_P	Load Capacitance	R, G, B	-		20	pF	
V_{outpp}	Output Voltage Swing		0.5		1.2	V	Available: 0.5 V, 0.7 V, 1.0 V, 1.2 V
U_{offset}	RGB Offset		175		375	mV	
T_{RF}	Rise/Fall Times		-		12.5	ns	
R_l	Load Resistance		10		-	kΩ	
-	Diff. Non-linearity		-0.5		0.5	LSB	
-	Int. Non-linearity		-0.5		0.5	LSB	
-	RGB Channel Matching		-		3	%	1.2 V Output Voltage Swing
T_{skew}	Skew to COR, Blank		-5		5	ns	
T_{Jit}	Jitter to horizontal Sync Reference		-		4	ns	
Address Bits							
T_r	Output Rise Time	A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, $\overline{A14}$, $\overline{A15}$, $\overline{ALE}$, $\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$	-		15	ns	(10% - 90%)
T_f	Output Fall Time		-		15	ns	(10% - 90%)
C_L	Load Capacitance		-		50	pF	
Alternate Address Control Lines							
T_r	Output Rise Time	P4.0, P4.1, P4.2, P4.3, P4.4	-		15	ns	(10% - 90%)
T_f	Output Fall Time		-		15	ns	(10% - 90%)
C_L	Load Capacitance		-		50	pF	
C_I	Pin Capacitance		-		10	pF	

Symbol	Parameter	Pin Name	Limit Values			Unit	Test Conditions
			Min.	Typ.	Max.		
Data Bits							
T_r	Output Rise Time	D0, D1, D2, D3, D4, D5, D6, D7	-		15	ns	(10% - 90%)
T_f	Output Fall Time		-		15	ns	(10% - 90%)
C_L	Load Capacitance		-		50	pF	
C_I	Pin Capacitance		-		10	pF	
Control Bit CORBL=0, BLANK only							
T_r	Output Rise Time	BLANK, CORBLA	8		15	ns	(10% - 90%)
T_f	Output Fall Time		8		15	ns	(10% - 90%)
$V_{i\text{-}n}$	Output Voltage no Data Insertion (Video)		0		0.4	V	
$V_{i\text{-}y}$	Output Voltage for Data Insertion		2.4		$V_{DD\ 3.3}$	V	
C_L	Load Capacitance		-		50	pF	
Control Bit CORBL=1, BLANK and COR							
T_r	Output Rise Time	BLANK, CORBLA	-		12.5	ns	(10% - 90%)
T_f	Output Fall Time		-		12.5	ns	(10% - 90%)
$V_{ic\text{-}n}$	Output Voltage no Data Insertion no Contrast Reduction		0		0.4	V	-
$V_{c\text{-}y}$	Output Voltage for Contrast Reduction and no Data Insertion		V_{mmin}		V_{mmax}	V	V_{mmin} = 1/3 x $V_{DD\ 3.3}$ - 150 mV V_{mmax} = 1/3 x $V_{DD\ 3.3}$ + 150 mV
$V_{i\text{-}y}$	Output Voltage for Data Insertion		2.4		$V_{DD\ 3.3}$	V	-
C_L	Load Capacitance		-		20	pF	Pure Capacity Load
HSYNC (Slave Mode)							
T_r	Input Rise Time	HSYNC	-		100	ns	(10% - 90%)
T_f	Input Fall Time		-		100	ns	(10% - 90%)
V_{HYST1}	Input Hysteresis 1		200		450	mV	Hys 1 Selected by Software
V_{HYST2}	Input Hysteresis 2		25		275	mV	HYs Selected by Software
T_{IPWH}	Input Pulse Width		100		-	ns	-
C_I	Pin Capacitance		-		10	pF	-
I_I	Leakage Current		-1		1	µA	-
V_{IL}	Input Low Voltage		-0.4		0.8	V	-
V_{IH}	Input High Voltage		2.0		2.6	V	-

Symbol	Parameter	Pin Name	Limit Values			Unit	Test Conditions
			Min.	Typ.	Max.		
VSYNC							
T_r	Input Rise Time	HSYNC	-		200	ns	(10% - 90%)
T_f	Input Fall Time		-		200	ns	(10% - 90%)
T_{IPWV}	Input Pulse Width		2/fh		-	-	-
T_r	Output Rise time		-		15	ns	(10% - 90%)
T_f	Output Fall Time		-		15	ns	(10% - 90%)
C_L	Load Capacitance		-		50	pF	-
C_I	Pin Capacitance		-		10	pF	-
V_{IL}	Input Low Voltage		-0.4		0.8	V	-
V_{IH}	Input High Voltage		2.0		2.6	V	-
Typical VCS Timing (Master Mode)							
T_{HPVCS}	Pulse Width of H-Sync		4.59		4,59	µs	-
T_{DEP}	Distance between Equalizing Impulses		31.98		31.98	µs	-
T_{EP}	Pulse Width of Equalizing Impulses		2.31		2.31	µs	-
T_{FSP}	Pulse Width of Field Sync Impulses		27.39		27.39	µs	-
T_{HPR}	Horizontal Period		-		-	µs	Depends on register HPR
P1.x, P3.x, P4.x							
T_r	Output Rise time	P1.x, P3.x, P4.x	-		15	ns	(10% - 90%)
T_f	Output Fall Time		-		15	ns	(10% - 90%)
C_L	Load Capacitance		-		50	pF	-
C_I	Pin Capacitance		-		10	pF	-

4.10.4. Timings

4.10.4.1. Sync

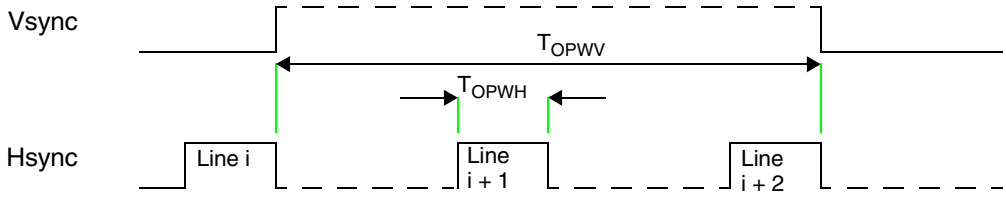


Fig. 4–10: H/V-Sync-Timing (Sync Master Mode)

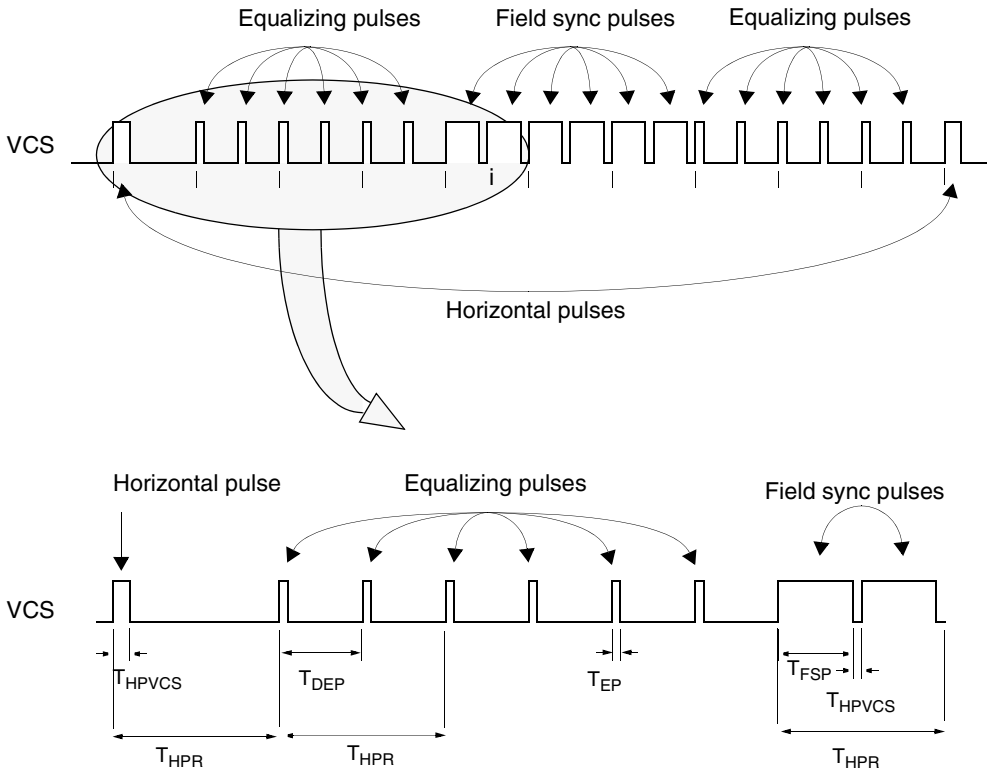
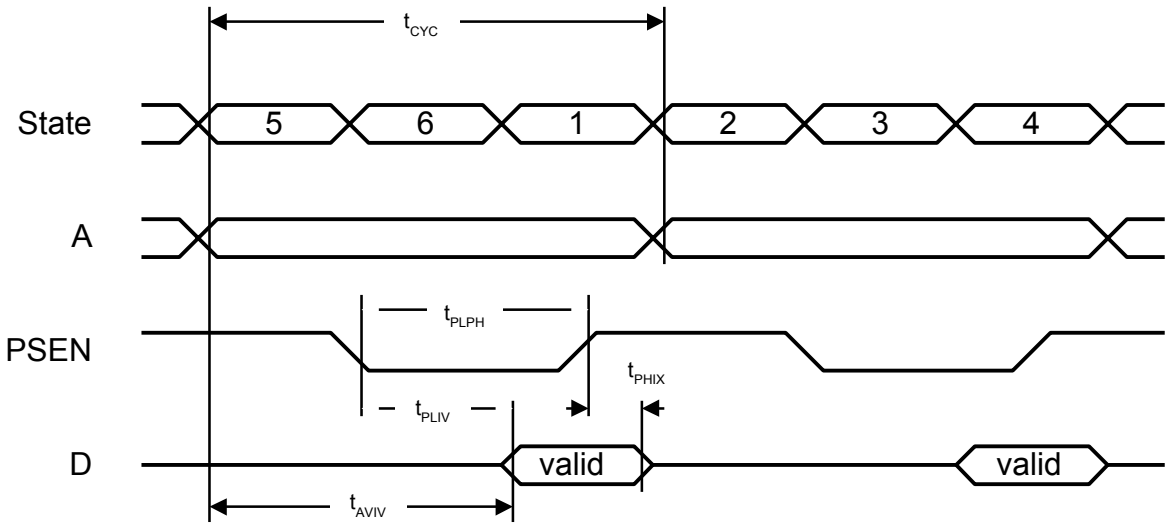


Fig. 4–11: VCS-Tming (Sync Master Mode)

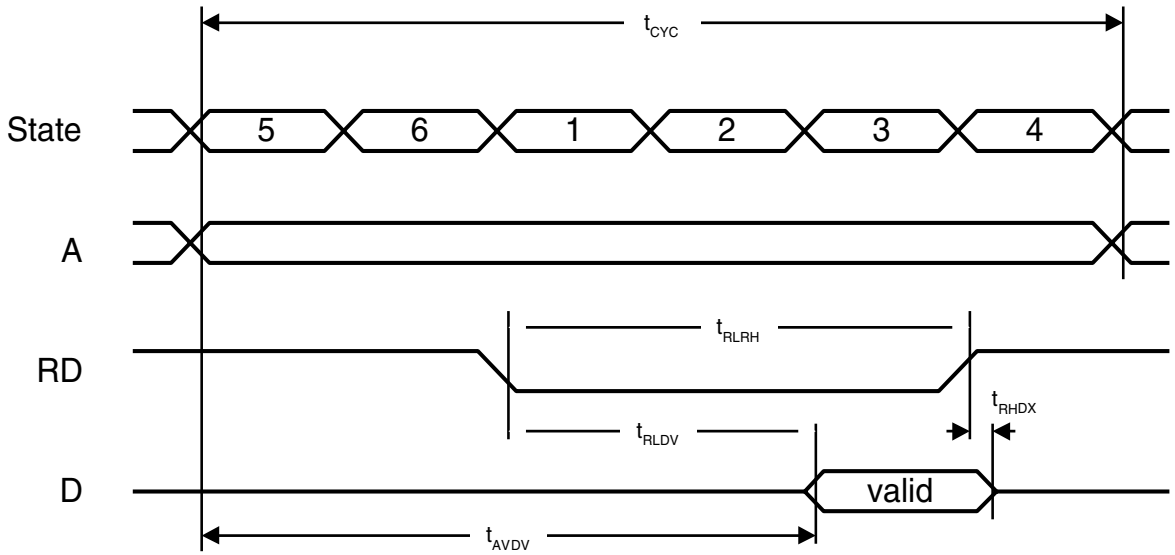
4.10.4.2. Program Memory Read Cycle



Parameter	Symbol	Min	Max
Frequency of internal clock	f_{SYS}		
Instruction read cycle time	t_{CYC}		
PSEN Pulse width	t_{PLPH}	80 ns	
PSEN to valid instruction in	t_{PLIV}	57.5 ns	
Instruction hold after PSEN	t_{PHIX}		0 ns
Address to valid instruction in	t_{AVIV}	115 ns	

Fig. 4–12: Program Memory Read Cycle

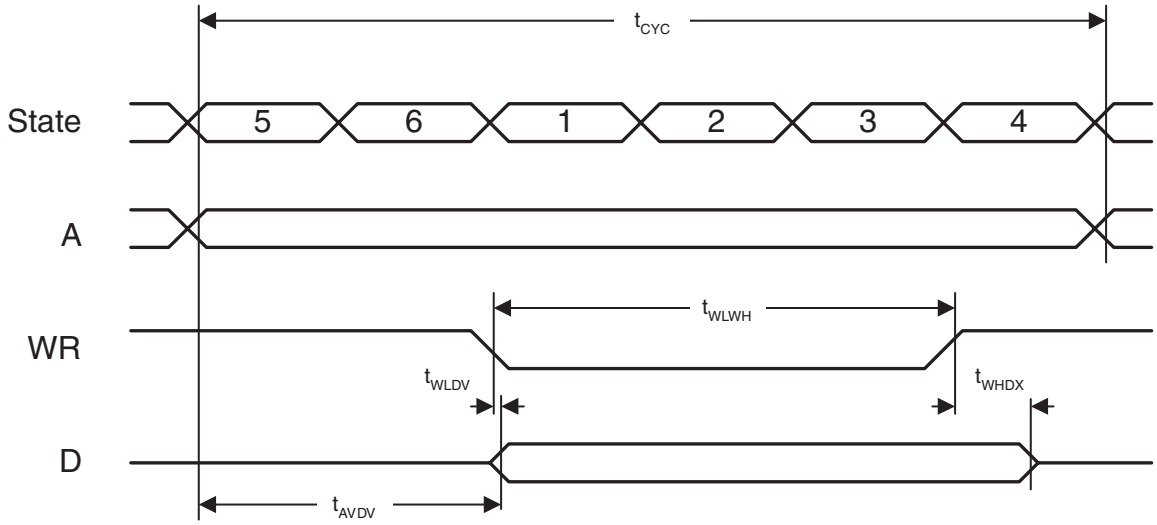
4.10.4.3. Data Memory Read Cycle



Parameter	Symbol	Min	Max
Frequency of internal clock	f_{SYS}		
Data read cycle time	t_{CYC}		
RD Pulse width	t_{RLRH}	170 ns	
RD to valid data in	t_{RLDV}	117.5 ns	
Data hold after RD	t_{RHDX}		0 ns
Address to valid data in	t_{AVDV}	230 ns	

Fig. 4–13: Data Memory Read Cycle

4.10.4.4. Data Memory Write Cycle



Parameter	Symbol	Min	Max
Frequency of internal clock	f_{SYS}		
Data write cycle time	t_{CYC}		
WR Pulse width	t_{WLWH}	170 ns	
WR to data out	t_{WLDV}		15 ns
Data hold after WR	t_{WHDX}	12,5 ns	
Address to valid data out	t_{AVDV}		135

Fig. 4–14: Data Memory Write Cycle

4.10.4.5. Blank/Cor

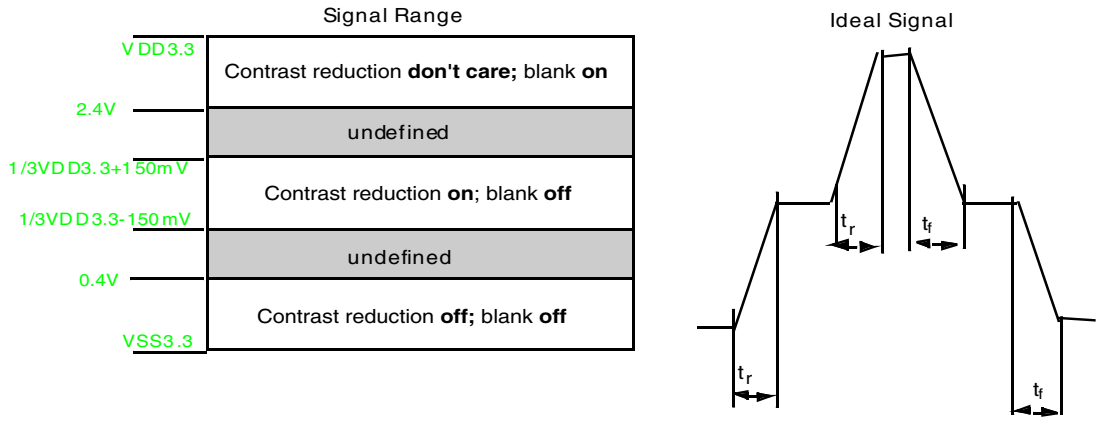


Fig. 4–15: Output Voltage of the Combined BLAN/COR Reduction Signal

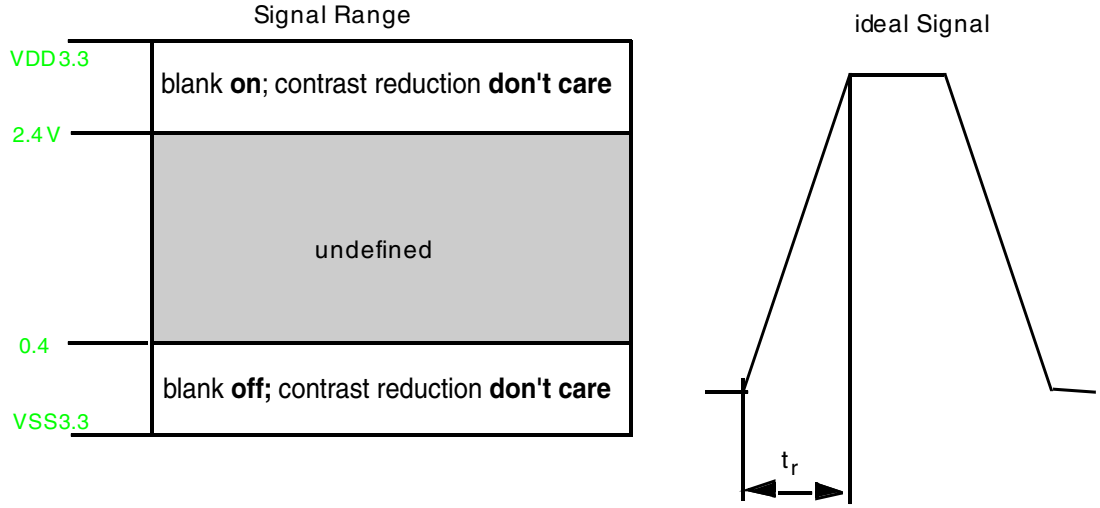


Fig. 4–16: Output Voltage for Blanking Signal

5. Applications

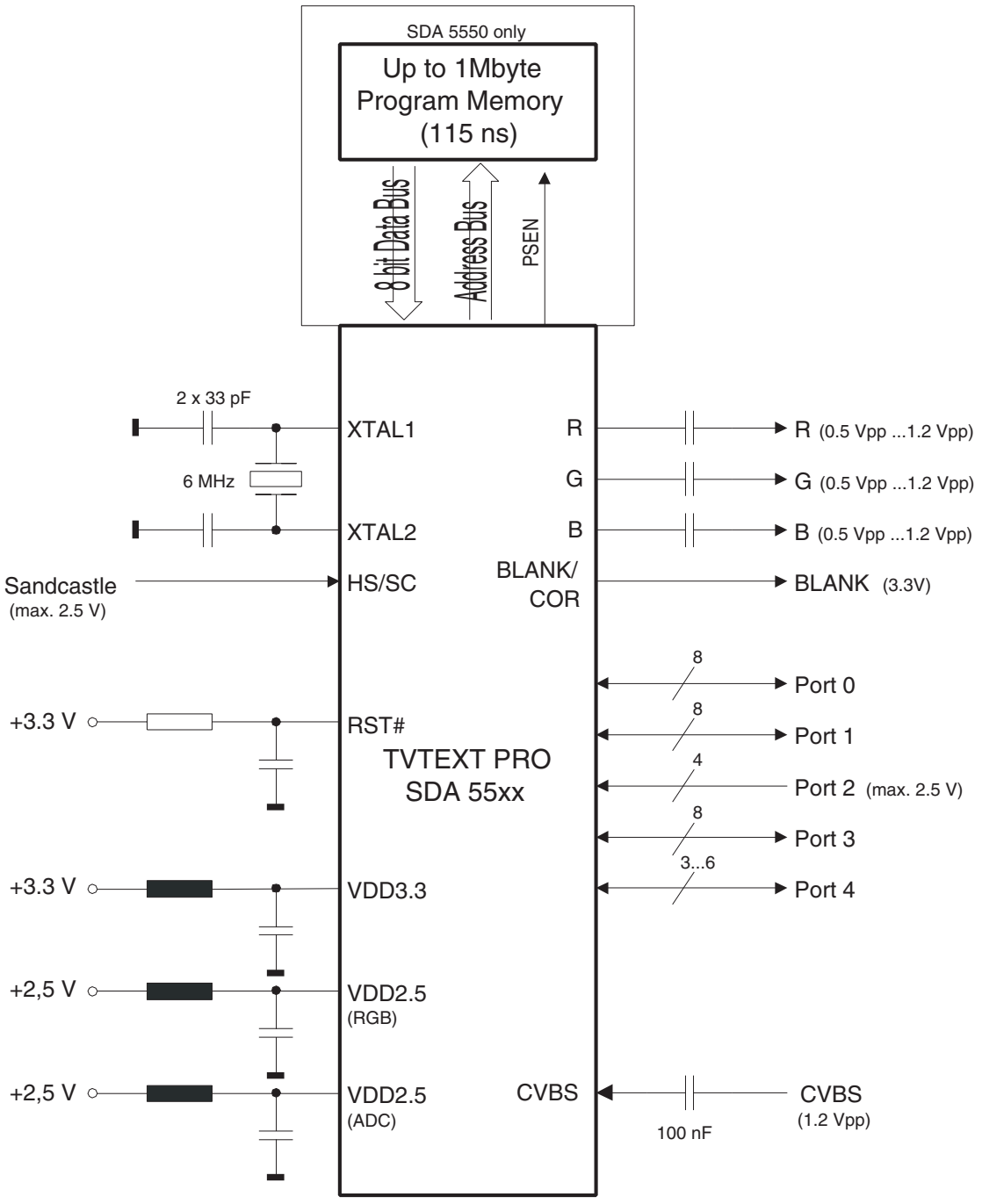


Fig. 5–1: Application Diagram

Intentionally Vacant

6. Data Sheet History

1. Data Sheet: “SDA 55xx TVText Pro”, July 27, 2001, 6251-556-1DS. First release of the data sheet.

2. Data Sheet: “SDA 55xx TVText Pro”, March 23, 2004, 6251-556-2DS. Second release of the data sheet.
   Major changes:

– New revision, completely updated

– Outline Dimensions, new graphics

– Pin configuration, new graphics

– Electrical Characteristics updated

3. Data Sheet: “SDA 55xx TVText Pro”, Sept. 10, 2004, 6251-556-3DS. Third release of the data sheet.
   Major changes:

– New type SDA 5577 added to the SDA 55xx-family

– Absolute maximum Ratings: Ambient temperature limit value min: −10 °C

Micronas GmbH
Hans-Bunte-Strasse 19
D-79108 Freiburg (Germany)
P.O. Box 840
D-79008 Freiburg (Germany)
Tel. +49-761-517-0
Fax +49-761-517-2174
E-mail: docservice@micronas.com
Internet: www.micronas.com

Printed in Germany
Order No. 6251-556-3DS